ESMA 6616 Linear Models

Dr. Wolfgang Rolke

Contents

1 R

For a detailed introduction to R you can read the material of my course Computing with R

1.1 Installation and Updating

1.1.1 Installing R

You can get a free version of R for your computer from a number of sources. The download is about 70MB and setup is fully automatic. Versions for several operating systems can be found on the R web site

https://cran.r-project.org

Note

- the one item you should change from the defaults is to install R into a folder under the root, aka C:\R
- You might be asked at several times whether you want to do something (allow access, run a program, save a library, ...), always just say yes!
- You will need to connect to a reasonably fast internet for these steps.
- This will take a few minutes, just wait until the > sign appears.

FOR MAC OS USERS ONLY

There are a few things that are different from MacOS and Windows. Here is one thing you should do:

Download XQuartz - XQuartz-2.7.11.dmg
Open XQuartz
Type the letter R (to make XQuartz run R)
Hit enter Open R Run the command .First()
Then, every command should work correctly.

1.1.2 RStudio

We will run R using an interface called **RStudio**. You can download it at RStudio.

1.1.3 Updating

R releases new versions about every three months or so. In general it is not necessary to get the latest version every time. Every now and then a package won't run under the old version, and then it is time to do so. In essence this just means to install the latest version of R from CRAN. More important is to now also update ALL your packages to the latest versions. This is done simply by running

```
update.packages(ask=FALSE, dependencies=TRUE)
```

1.2 R Markdown

R Markdown is a program for making dynamic documents with R. An R Markdown document is written in *markdown*, an easy-to-write plain text format with the file extension .Rmd. It can contain chunks of embedded R code. It has a number of great features:

- easy syntax for a number of basic objects
- code and output are in the same place and so are always synced
- several output formats (html, latex, word)

In recent years I (along with many others) who work a lot with R have made Rmarkdown the basic way to work with R. So when I work on a new project I immediately start a corresponding R markdown document.

1.2.1 Get Started

to start writing an R Markdown document open RStudio, File > New File > R Markdown. You can type in the title and some other things.

The default document starts like this:

---
title: "My first R Markdown Document"
author: "Dr. Wolfgang Rolke"
date: "April 1, 2018"
output: html_document

This follows a syntax called YAML (also used by other programs). Everything between the three dashes (which are needed) is YAML code. There are other things that can be put here as well, or you can erase all of it.

YAML stands for Yet Another Markup Language. It has become a standard for many computer languages to describe different configurations. For details go to yaml.org.

Then there is other stuff you should erase. Next File > Save. Give the document a name with the extension .Rmd

I have a number of things that I need in (almost) all of my Rmd files, and I am to lazy to erase the stuff that the default starting document comes with. So I have a file called blank.Rmd which already has everything as I (usually) want it. All I need to do is rename it and put it in the right folder.

1.2.2 Basic R Markdown Syntax

Markdown has simple keyboard shortcuts for many basic editing features. For example, #, ##, ### are for chapter and section headers. Subscripts are done with beginning and ending ~, so for X_1 you need to type X~1~. For superscripts use X^1^.

For a complete list of the basic syntax go to https://rmarkdown.rstudio.com/articles_intro.html or to https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet

1.2.3 Embedded Code

There are two ways to include code chunks (yes, that's what they are called!) into an R Markdown document:

a. stand alone code

simultaneously enter CTRL-ALT-i and you will see this:

“‘{r}
“‘

Here ‘ is a back tick, on Windows keyboards found on the upper left key, below ~.

you can now enter any R code you like:

“‘{r}
x <- rnorm(10)
mean(x)
“‘

which will appear in the final document as

```
x <- rnorm(10)
mean(x)
```

Actually, it will be like this:

```
x<-rnorm(10)
mean(x)
```

```
## [1] -0.03853522
```

so we can see the result of the R calculation as well. The reason it didn't appear like this before was that I added the argument eval=FALSE:

“‘{r eval=FALSE}

which keeps the code chunk from actually executing (aka *eval*uating). This is useful if the code takes along time to run, or if you want to show code that is actually faulty, or for any number of other reasons.

there are several useful arguments:

- eval=FALSE (shows but doesn't run the code)

- eval=2:5 (shows all the code but only runs lines 2 to 5)
- echo=FALSE (the code chunk is run but does not appear in the document)
- echo=2:5 (shows only code on lines 2 to 5)
- warning=FALSE (warnings are not shown)
- message=FALSE (messages are not shown)
- cache=TRUE (code is run only if there has been a change, useful for lengthy calculations)
- error=TRUE (if there are errors in the code R normal terminates the parsing (executing) of the markdown document. With this argument it will ignore the error, which helps with debugging)
- engine='Rcpp' (to include C++ code)

Many of these options can be set globally, so they are active for the whole document. This is useful so you don't have to type them in every time. I have the following code chunk at the beginning of all my Rmd:

```
library(knitr)
opts_chunk$set(fig.width=6, fig.align = "center",
      out.width = "70%", warning=FALSE, message=FALSE)
```

We have already seen the message and warning options. The other one puts any figure in the middle of the page and sizes it nicely.

If you have to override these defaults just include that in the specific chunk.

b. inline code.

here is a bit of text:

and the mean was -0.0385352.

Now I didn't type in the number, it was done with the chunk

```
## `r mean(x)`
```

1.2.4 Creating Output

To create the output you have to "knit" the document. This is done by clicking on the *knit* button above. If you click on the arrow you can change the output format.

1.2.4.1 HTML, Latex(Pd), Word, PowerPoint etc. One of the great features of Markdown is that its syntax is independent of the eventual document format, so the same markdown file can immediately produce an HTML file of a pdf or or...

In this class we will only use the HTML format, which is the easiest.

In order to knit to pdf you have to install a latex interpreter. My suggestion is to use Miktex, but if you already have one installed it might work as well.

There are several advantages / disadvantages to each output format:

- HTML is much faster
- HTML looks good on a webpage, pdf looks good on paper
- HTML needs an internet connection to display math, pdf does not
- HTML can use both html and latex syntax, pdf works only with latex (and a little bit of html)

I generally use HTML when writing a document, and use pdf only when everything else is done. There is one problem with this, namely that a document might well knit ok to HTML but give an error message when knitting to pdf. Moreover, those error messages are weird! Not even the line numbers are anywhere near right. So it's not a bad idea to also knit to pdf every now and then.

1.2.5 Tables

One of the more complicated things to do in R Markdown is tables. For a nice illustration look at

https://stackoverflow.com/questions/19997242/simple-manual-rmarkdown-tables-that-look-good-in-html-pdf-and-docx

My preference is to generate a data frame and the use the *kable* function:

```
Gender <- c("Male", "Male", "Female")
Age <- c(20, 21, 19)
df <- data.frame(Gender, Age)
knitr::kable(df)
```

Gender	Age
Male	20
Male	21
Female	19

I have written my own kable routine which improves a bit on the basic version:

```
kable.nice <- function (x,
      do.row.names = TRUE,
      col.names = NA, font.size = 15)
{
  library(tidyverse)
  library(kableExtra)
  kable(x, row.names = do.row.names,
        col.names = col.names) %>%
        kable_styling(bootstrap_options = "striped",
```

```
            full_width = FALSE,
            font_size = font.size)
}
kable.nice(df)
```

	Gender	Age
1	Male	20
2	Male	21
3	Female	19

which I am sure you agree is nice! You can use it yourself, just copy paste the function code into an R chunk of your document.

It is also possible to use HTML code to make a table:

```
##  <table border="1">
##  <tr><th>Gender</th><th>Age</th></tr>
##  <tr><td>Male</td><td>20</td></tr>
##  <tr><td>Male</td><td>21</td></tr>
##  <tr><td>Female</td><td>19</td></tr>
##  </table>
```

It will look like this in HTML:

Gender

Age

Male

20

Male

21

Female

19

but won't look like anything in pdf.

The corresponding latex table will look good in pdf but not in HTML!

So what do you do if you don't know yet what the output will be, or if you want your routine to produce nice output either way? The solution is this: the document can check what the output format is at run time, and then insert the corresponding code. This works as follows. Say we want to include some code to print a piece of text in red, say for highlighting it. Now in html we would need the code , then the text and finally to get back to black. In latex however we need \textcolor{red}{our text}. Here is a little routine that will do it:

```
fontcolor <- function (txt) {
  library(knitr)
  output.format <- opts_knit$get("rmarkdown.pandoc.to")
# this figures out what the output format is
  if(output.format == "latex")
    out <- paste0("\\textcolor{red}{", txt, "}")
  else
    out <- paste0("<font color='red'>", txt, "</font>")
  out
}
```

and now if we have

```
## `r fontcolor("this is in red")`
```

it will appear as this is in red in either html or latex.

1.2.6 LATEX

You have not worked with latex (read: latek) before? Here is your chance to learn. It is well worthwhile, latex is the standard document word processor for science. And once you get used to it is WAY better and easier than (say) Word.

Because latex code will generally display correctly in an html document but html will not in a latex document I suggest to stick as much as possible with latex.

Latex has a HUGE list of symbols for just about anything. A nice list of common symbols is found on https://artofproblemsolving.com/wiki/index.php/LaTeX:Symbols. Often when I need one I don't remember I just google it. For example, say I want to use the symbol for the real numbers: $\mathbf{R}$. So I google "latex real numbers symbol", and the first document tells me the code is \mathbf{R}!

Latex code is usually used in two ways: as part of a sentence or as stand-alone. In the first case use a single dollar sign at the beginning and the end. For example the code

We want to integrate the function

```
$f(x)=\exp(-x^2)$
```

will display as

We want to integrate the function

$f(x) = \exp(-x^2)$.

Exercise

Does anyone know how I displayed the code and not the formula the first time?

The other way to display math in latex is via

1.2.6.1 Multiline math

say you want the following in your document:

$$E[X] = \int_{-\infty}^{\infty} xf(x)dx =$$
$$\int_0^1 xdx = \frac{1}{2}x^2|_0^1 = \frac{1}{2}$$

for this to display correctly in HTML and PDF you need to use the format

```
##  $$
##  \begin{aligned}
##  &E[X] = \int_{-\infty}^{\infty} xf(x) dx=\\
##  &\int_{0}^{1} x dx    = \frac12 x^2 |_0^1 = \frac12
##  \end{aligned}
##  $$
```

so a multiline expression starts and ends with double dollar signs.

By default when you knit to pdf the intermediate latex file is deleted. If you want to keep it, maybe so you can change it in a latex editor, use the following in the YAML header:

output:
  pdf_document:
    keep_tex: true

notice the spaces before the text, they are needed!

1.2.7 snippets

A *snippet* is a short piece of code that one uses quite often, and so it would be nice not to have to type it in every time. RStudio has a number of them pre-defined. Go to Tools > Global Options > Code > Edit Snippets.

There are snippets for various languages, including R Markdown. To use a snippet, simply type the code and then Shift+Tab.

You can even wright your own! For example, I have one called *mta* that has all the basics to start a multi-line latex math expression.

1.3 R Basics I

To start run

```
ls()
```

This shows you a "listing"" of the files (data, routines etc.) in the current project. (Likely there is nothing there right now)

Everything in R is either a data set or a function. It is a function if it is supposed to do something (maybe calculate something, show you something like a graph or something else

etc.). If it is a function is ALWAYS NEEDS (). Sometimes the is something in between the parentheses, like in

```
mean(x)
```

```
## [1] 6
```

Sometimes there isn't like in the ls(). But the () has to be there anyway.

If you have worked for a while you might have things you need to save, do that by clicking on

File > Save

RStudio has a nice recall feature, using the up and down arrow keys. Also, clicking on the History tab shows you the recently run commands. Finally, typing the first three letters of a command in the console and then typing CTRL-^ shows you a list of when you ran commands like this the last times.

R is case-sensitive, so a and A are two different things.

Often during a session you create objects that you need only for a short time. When you no longer need them use **rm** to get rid of them:

```
x <- 10
x^2
```

```
## [1] 100
```

```
rm(x)
```

the **<-** is the *assignment* character in R, it assigns what is on the right to the symbol on the left. (Think of an arrow to the left)

1.3.1 Data Entry

For a few numbers the easiest thing is to just type them in:

```
x <-  c(10, 2, 6, 9)
x
```

```
## [1] 10  2  6  9
```

c() is a function that takes the objects inside the () and **c**ombines them into one single object (a vector).

1.3.2 Data Types in R

the most basic type of data in R is a **vector**, simply a list of values.

Say we want the numbers 1.5, 3.6, 5.1 and 4.0 in an R vector called x, then we can type

```
x <- c(1.5, 3.6, 5.1, 4.0)
x
```

```
## [1] 1.5 3.6 5.1 4.0
```

Often the numbers have a structure one can make use of:

```
1:10
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
10:1
```

```
##  [1] 10  9  8  7  6  5  4  3  2  1
```

```
1:20*2
```

```
##  [1]  2  4  6  8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40
```

```
c(1:10, 1:10*2)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10  2  4  6  8 10 12 14 16 18 20
```

Sometimes you need parentheses:

```
n <- 10
1:n-1
```

```
##  [1] 0 1 2 3 4 5 6 7 8 9
```

```
1:(n-1)
```

```
## [1] 1 2 3 4 5 6 7 8 9
```

The *rep* ("repeat") command is very useful:

```
rep(1, 10)
```

```
##  [1] 1 1 1 1 1 1 1 1 1 1
```

```
rep(1:3, 10)
```

```
##  [1] 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3
```

```
rep(1:3, each=3)
```

```
## [1] 1 1 1 2 2 2 3 3 3
```

```
rep(c("A", "B", "C"), c(4,7,3))
```

```
##  [1] "A" "A" "A" "A" "B" "B" "B" "B" "B" "B" "B" "C" "C" "C"
```

what does this do?

```
rep(1:10, 1:10)
```

1.3.3 Commands for Vectors

To find out how many elements a vector has use the *length* command:

```
x <- c(1.4, 5.1, 2.0, 6.8, 3.5, 2.1, 5.6, 3.3, 6.9, 1.1)
length(x)
```

```
## [1] 10
```

The elements of a vector are accessed with the bracket [ ] notation:

```
x[3]
```

```
## [1] 2
```

```
x[1:3]
```

```
## [1] 1.4 5.1 2.0
```

```
x[c(1, 3, 8)]
```

```
## [1] 1.4 2.0 3.3
```

```
x[-3]
```

```
## [1] 1.4 5.1 6.8 3.5 2.1 5.6 3.3 6.9 1.1
```

```
x[-c(1, 2, 5)]
```

```
## [1] 2.0 6.8 2.1 5.6 3.3 6.9 1.1
```

Instead of numbers a vector can also consist of characters (letters, numbers, symbols etc.) These are identified by quotes:

```
c("A", "B", 7, "%")
```

```
## [1] "A" "B" "7" "%"
```

A vector is either numeric or character, but never both (see how the 7 was changed to "7").

You can turn one into the other (if possible) as follows:

```
x <- 1:10
x
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
as.character(x)
```

```
##  [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10"
```

```
x <- c("1", "5", "10", "-3")
x
```

```
## [1] "1"  "5"  "10" "-3"
```

```
as.numeric(x)
```

```
## [1]  1  5 10 -3
```

A third type of data is logical, with values either TRUE or FALSE.

```
x <- 1:10
x
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
x > 4
```

```
##  [1] FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

these are often used as conditions:

```
x[x>4]
```

```
## [1]  5  6  7  8  9 10
```

This, as we will see shortly, is EXTREMELY useful!

1.3.4 Data Frames

data frames are the basic format for data in R. They are essentially vectors of equal length put together as columns.

A data frame can be created as follows:

```
df <- data.frame(
  Gender=c("M", "M", "F", "F", "F"),
  Age=c(23, 25, 19, 22, 21),
  GPA=c(3.5, 3.7, 2.9, 2.8, 3.1)
)
df
```

```
##   Gender Age GPA
## 1      M  23 3.5
## 2      M  25 3.7
## 3      F  19 2.9
## 4      F  22 2.8
## 5      F  21 3.1
```

1.3.5 Lists

The most general data structures are lists. They are simply a collection of objects. There are no restrictions on what those objects are.

```
lst <- list(
  Gender=c("M", "M", "F", "F", "F"),
  Age=c(23, 25, 19, 22, 21, 26, 34),
  f=function(x) x^2,
  list(A=c(1, 1), B=c("X", "X", "Y"))
)
lst
```

1.3.5.1 Example

```
## $Gender
## [1] "M" "M" "F" "F" "F"
##
## $Age
## [1] 23 25 19 22 21 26 34
##
## $f
## function(x) x^2
##
## [[4]]
## [[4]]$A
## [1] 1 1
##
## [[4]]$B
## [1] "X" "X" "Y"
```

A data frame is a list with an additional requirement, namely that the elements of the list be of equal length.

1.3.5.2 Case Study: UPR Admissions

consider the **upr** data set . This is the application data for all the students who applied and were accepted to UPR-Mayaguez between 2003 and 2013.

```
dim(upr)
```

```
## [1] 23666    16
```

tells us that there were 23666 applications and that for each student there are 16 pieces of information.

```
colnames(upr)
```

```
##  [1] "ID.Code"        "Year"           "Gender"         "Program.Code"   "Highschool.
##  [6] "Aptitud.Verbal" "Aptitud.Matem"  "Aprov.Ingles"   "Aprov.Matem"    "Aprov.Espan
## [11] "IGS"            "Freshmen.GPA"   "Graduated"      "Year.Grad."     "Grad..GPA"
## [16] "Class.Facultad"
```

shows us the variables

```
head(upr, 3)
```

```
##      ID.Code Year Gender Program.Code Highschool.GPA Aptitud.Verbal Aptitud.Matem
## 1 00C2B4EF77 2005      M          502           3.97            647           621
## 2 00D66CF1BF 2003      M          502           3.80            597           726
## 3 00AB6118EB 2004      M         1203           4.00            567           691
##   Aprov.Ingles Aprov.Matem Aprov.Espanol IGS Freshmen.GPA Graduated Year.Grad. Grad..
## 1          626         672           551 342         3.67        Si       2012      3
## 2          618         718           575 343         2.75        No         NA
## 3          424         616           609 342         3.62        No         NA
##   Class.Facultad
```

```
## 1           INGE
## 2           INGE
## 3       CIENCIAS
```

shows us the first three cases.

Let's say we want to find the number of males and females. We can use the table command for that:

```
table(Gender)
```

```
## Error: object 'Gender' not found
```

What happened? Right now R does not know what Gender is because it is "hidden" inside the upr data set. Think of **upr** as a box that is currently closed, so R can't look inside and see the column names. We need to open the box first:

```
attach(upr)
table(Gender)
```

```
## Gender
##     F     M
## 11487 12179
```

there is also a detach command to undo an attach, but this is not usually needed because the attach goes away when you close R.

Note: you need to attach a data frame only once in each session working with R.

Note: Say you are working first with a data set "students 2016" which has a column called Gender, and you attached it. Later (but in the same R session) you start working with a data set "students 2017" which also has a column called Gender, and you are attaching this one as well. If you use Gender now it will be from "students 2017".

1.3.6 Subsetting of Data Frames

Consider the following data frame (not a real data set):

```
students
```

```
##    Age GPA Gender
## 1   22 3.1   Male
## 2   23 3.2   Male
## 3   20 2.1   Male
## 4   22 2.1   Male
## 5   21 2.3 Female
## 6   21 2.9   Male
## 7   18 2.3 Female
## 8   22 3.9   Male
## 9   21 2.6 Female
## 10  18 3.2 Female
```

Here each single piece of data is identified by its row number and its column number. So for example in row 2, column 2 we have "3.2", in row 6, column 3 we have "Male".

As with the vectors before we can use the [ ] notation to access pieces of a data frame, but now we need to give it both the row and the column number, separated by a ,:

```
students[6, 3]
```

```
## [1] "Male"
```

As before we can pick more than one piece:

```
students[1:5, 3]
```

```
## [1] "Male"   "Male"   "Male"   "Male"   "Female"
```

```
students[1:5, 1:2]
```

```
##   Age GPA
## 1  22 3.1
## 2  23 3.2
## 3  20 2.1
## 4  22 2.1
## 5  21 2.3
```

```
students[-c(1:5), 3]
```

```
## [1] "Male"   "Female" "Male"   "Female" "Female"
```

```
students[1, ]
```

```
##   Age GPA Gender
## 1  22 3.1   Male
```

```
students[, 2]
```

```
##  [1] 3.1 3.2 2.1 2.1 2.3 2.9 2.3 3.9 2.6 3.2
```

```
students[, -3]
```

```
##    Age GPA
## 1   22 3.1
## 2   23 3.2
## 3   20 2.1
## 4   22 2.1
## 5   21 2.3
## 6   21 2.9
## 7   18 2.3
## 8   22 3.9
## 9   21 2.6
## 10  18 3.2
```

another way of subsetting a data frame is by using the $ notations:

```
students$Gender
```

```
##  [1] "Male"   "Male"   "Male"   "Male"   "Female" "Male"   "Female" "Male"   "Female"
## [10] "Female"
```

1.3.7 Subsetting of Lists

The double bracket and the $ notation also work for lists:

```
lst <- list(
  Gender=c("M", "M", "F", "F", "F"),
  Age=c(23, 25, 19, 22, 21, 26, 34),
  f=function(x) x^2,
  list(A=c(1, 1), B=c("X", "X", "Y"))
)
lst[[4]][[2]]
```

1.3.7.1 Example

```
## [1] "X" "X" "Y"
```

```
lst$Gender
```

```
## [1] "M" "M" "F" "F" "F"
```

1.3.8 Vector Arithmetic

R allows us to apply any mathematical functions to a whole vector:

```
x <- 1:10
2*x
```

```
##  [1]  2  4  6  8 10 12 14 16 18 20
```

```
x^2
```

```
##  [1]   1   4   9  16  25  36  49  64  81 100
```

```
log(x)
```

```
##  [1] 0.0000000 0.6931472 1.0986123 1.3862944 1.6094379 1.7917595 1.9459101 2.0794415
##  [9] 2.1972246 2.3025851
```

```
sum(x)
```

```
## [1] 55
```

```
y <- 21:30
```

```
x+y
```

```
##  [1] 22 24 26 28 30 32 34 36 38 40
```

```
x^2+y^2
```

```
##  [1]  442  488  538  592  650  712  778  848  922 1000
```

```
mean(x+y)
```

```
## [1] 31
```

Let's try something strange:

```
c(1, 2, 3) + c(1, 2, 3, 4)
```

```
## [1] 2 4 6 5
```

so R notices that we are trying to add a vector of length 3 to a vector of length 4. This should not work, but it actually does!

When it runs out of values in the first vector, R simply starts all over again.

In general this is more likely a mistake by you, check that this is what you really wanted to do!

1.3.9 *apply*

A very useful routine in R is *apply*, and its brothers.

Let's say we have the following matrix:

```
Age <- matrix(sample(20:30, size=100, replace=TRUE), 10, 10)
Age[1:5, 1:5]
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]   20   27   27   20   29
## [2,]   25   23   26   25   27
## [3,]   25   26   20   21   24
## [4,]   25   30   23   22   20
## [5,]   23   26   30   30   26
```

and we want to find the sums of the ages in each column. Easy:

```
sum(Age[, 1])
```

```
## [1] 249
```

```
sum(Age[, 2])
```

```
## [1] 263
```

...

```
sum(Age[, 10])
```

```
## [1] 269
```

or much easier

```
apply(Age, 2, sum)
```

```
##  [1] 249 263 252 226 251 248 271 252 271 269
```

There are a number of apply routines for different data formats.

1.3.9.1 Case Study: upr admissions

Let's say we want to find the mean Highschool GPA:

```
mean(Highschool.GPA)
```

```
## [1] 3.65861
```

But what if we want to do this for each year separately? Notice that *apply* doesn't work here because the Years are not in separated columns. Instead we can use

```
tapply(Highschool.GPA, Year, mean)
```

```
##     2003     2004     2005     2006     2007     2008     2009     2010     2011
## 3.646627 3.642484 3.652774 3.654729 3.628072 3.648552 3.642946 3.665298 3.685485 3.69
##     2013
## 3.710843
```

1.4 R Basics II - Writing Functions

1.4.1 General Information

In R/RStudio you have several ways to write your own functions:

- In the R console type

```
myfun <- function(x) {
  out <- x^2
  out
}
```

- RStudio: click on File > New File > R Script. A new empty window pops up. Type fun, hit enter, and the following text appears:

name <- function(variables) {

}

change the name to *myfun*, save the file as myfun.R with File > Save. Now type in the code. When done click the Source button.

- fix: In the R console run

```
fix(myfun)
```

now a window with an editor pops up and you can type in the code. When you are done click on Save. If there is some syntax error DON'T run fix again, instead run

```
myfun <- edit()
```

myfun will exist only until you close R/RStudio unless you save the project file.

- Open any code editor outside of RStudio, type in the code, save it as myfun.R, go to the console and run

```
source('../some.folder/myfun.R')
```

Which of these is best? In large part that depends on your preferences. In my case, if I expect to need that function just for a bit I use the fix option. If I expect to need that function again later I start with the first method, but likely soon open the .R file outside RStudio because most code editors have many useful features not available in RStudio.

If *myfun* is open in RStudio there are some useful keyboard shortcuts. If the curser is on some line in the RStudio editor you can hit

- CTRL-Enter run current line or section
- CTRL-ALT-B run from beginning to line
- CTRL-Shift-Enter run complete chunk
- CTRL-Shift-P rerun previous

1.4.2 Testing

As always you can test whether an object is a function:

```
x <- 1
f <- function(x) x
is.function(x)
```

```
## [1] FALSE
```

```
is.function(f)
```

```
## [1] TRUE
```

1.4.3 Arguments

There are several ways to specify arguments in a function:

```
calc.power <- function(x, y, n=2) x^n + y^n
```

here n has a *default value*, x and y do not.

if the arguments are not named they are matched in order:

```
calc.power(2, 3)
```

```
## [1] 13
```

If an argument does not have a default it can be tested for

```
f <- function(first, second) {
  if(!missing(second))
      out <- first + second
  else out <- first
  out
}
f(1)
```

```
## [1] 1
```

```
f(1, s=3)
```

```
## [1] 4
```

There is a special argument ..., used to pass arguments on to other functions:

```
f <- function(x, which, ...) {
  f1 <- function(x, mult) mult*x
  f2 <- function(x, pow) x^pow
  if(which==1)
    out <- f1(x, ...)
  else
    out <- f2(x, ...)
  out
}
f(1:3, 1, mult=2)
```

```
## [1] 2 4 6
```

```
f(1:3, 2, pow=3)
```

```
## [1]  1  8 27
```

This is one of the most useful programming structures in R!

Note this example also shows that in R functions can call other functions. In many computer programs there are so called *sub-routines*, in R this concept does not exist, functions are just functions.

1.4.4 Return Values

A function can either return nothing or exactly one thing. It will automatically return the last object evaluated:

```
f <- function(x) {
  x^2
```

```
}
f(1:3)
```

```
## [1] 1 4 9
```

however, it is better programming style to have an explicit return object:

```
f <- function(x) {
  out <- x^2
  out
}
f(1:3)
```

```
## [1] 1 4 9
```

There is another way to specify what is returned:

```
f <- function(x) {
  return(x^2)
}
f(1:3)
```

```
## [1] 1 4 9
```

but this is usually used to return something early in the program:

```
f <- function(x) {
  if(!any(is.numeric(x)))
    return("Works only for numeric!")
  out <- sum(x^2)
  out
}
f(1:3)
```

```
## [1] 14
```

```
f(letters[1:3])
```

```
## [1] "Works only for numeric!"
```

If you want to return more than one item use a list:

```
f <- function(x) {
  sq <- x^2
  sm <- sum(x)
  list(sq=sq, sum=sm)
}
f(1:3)
```

```
## $sq
## [1] 1 4 9
##
## $sum
```

```
## [1] 6
```

1.4.5 Basic Programmming Structures in R

R has all the standard programming structures:

```
f <- function(x) {
  if(x>0) y <- log(x)
  else y <- NA
  y
}
f(c(2, -2))
```

1.4.5.1 Conditionals (if-else)

```
## [1] 0.6931472       NaN
```

A useful variation on the *if* statement is *switch*:

```
centre <- function(x, type) {
  switch(type,
         mean = mean(x),
         median = median(x),
         trimmed = mean(x, trim = .1))
}
x <- rcauchy(10)
centre(x, "mean")
```

```
## [1] 6.000776
```

```
centre(x, "median")
```

```
## [1] 1.300273
```

```
centre(x, "trimmed")
```

```
## [1] 1.166
```

special R construct: *ifelse*

```
x <- sample(1:10, size=7, replace = TRUE)
x
```

```
## [1]  5  2 10  9  4  3  5
```

```
ifelse(x<5, "Yes", "No")
```

```
## [1] "No"  "Yes" "No"  "No"  "Yes" "Yes" "No"
```

1.4.5.2 Loops

there are two standard loops in R:

- for loop

```
y <- rep(0, 10)
for(i in 1:10) y[i] <- i*(i+1)/2
y
```

```
##  [1]  1  3  6 10 15 21 28 36 45 55
```

sometimes we don't know the length of y ahead of time, then we can use

```
for(i in seq_along(y)) y[i] <- i*(i+1)/2
y
```

```
##  [1]  1  3  6 10 15 21 28 36 45 55
```

If there is more than one statement inside a loop use curly braces:

```
for(i in seq_along(y)) {
  y[i] <- i*(i+1)/2
  if(y[i]>40) y[i] <- (-1)
}
y
```

```
##  [1]  1  3  6 10 15 21 28 36 -1 -1
```

You can nest loops:

```
A <- matrix(0, 4, 4)
for(i in 1:4) {
  for(j in 1:4)
    A[i, j] <- i*j
}
A
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    2    3    4
## [2,]    2    4    6    8
## [3,]    3    6    9   12
## [4,]    4    8   12   16
```

- repeat loop

```
k <- 0
repeat {
  k <- k+1
  x <- sample(1:6, size=3, replace=TRUE)
  if(length(table(x))==1) break
}
k
```

```
## [1] 92
```

Notice that a repeat loop could in principle run forever. I usually include a counter that ensures the loop will eventually stop:

```
k <- 0
counter <- 0
repeat {
  k <- k+1
  counter <- counter+1
  x <- sample(1:6, size=3, replace=TRUE)
  if(length(table(x))==1 | counter>1000) break
}
k
```

```
## [1] 102
```

1.5 Important Commands

In the section I will list the most important commands in base R. The list is taken in large part from Hadley Wickham's book Advanced R. Most of them we already discussed. Those we have not you can read up on yourself.

1.5.1 The first functions to learn

? str

1.5.2 Important operators and assignment

%in%, match
=, <-, «-
$, [, [[, head, tail, subset
with
assign, get

1.5.3 Comparison

all.equal, identical
!=, ==, >, >=, <, <=
is.na, complete.cases
is.finite

1.5.4 Random variables

(q, p, d, r) * (beta, binom, cauchy, chisq, exp, f, gamma, geom, hyper, lnorm, logis, multinom, nbinom, norm, pois, signrank, t, unif, weibull, wilcox, birthday, tukey)

1.5.5 Matrix algebra

crossprod, tcrossprod
eigen, qr, svd
%*%, %o%, outer
rcond
solve

1.5.6 Workspace

ls, exists, rm
getwd, setwd
q
source
install.packages, library, require

1.5.7 Help

help, ?
help.search
apropos
RSiteSearch
citation
demo
example
vignette

1.5.8 Debugging

traceback
browser
recover
options(error =)
stop, warning, message
tryCatch, try

1.5.9 Output

print, cat
message, warning
dput
format
sink, capture.output

1.5.10 Reading and writing data

data
count.fields
read.csv, write.csv
read.delim, write.delim
read.fwf
readLines, writeLines
readRDS, saveRDS
load, save
library

1.5.11 Files and directories

dir
basename, dirname, tools::file_ext
file.path
path.expand, normalizePath
file.choose
file.copy, file.create, file.remove, file.rename, dir.create
file.exists, file.info
tempdir, tempfile
download.file,

2 Introduction

2.1 Identifying Students at Risk

At UPRM (and all other Universities) a big problem is retaining the students from year to year. That is, many first-year students never return for the second year, and so on. Some years ago our Chancellor put together a group of professors and asked us to find ways to improve the situation. Among other things we tried to see whether it was possible to identify those students that were at a high risk of not returning for the second year and those at risk for not graduating. We asked the Registrars office for some data and received the data set *upr* (now part of RESMA3.RData). Let's see what is in there:

```
dim(upr)
```

```
## [1] 23666    16
```

so the data set has 23666 records and 16 variables. Those are

```
colnames(upr)
```

```
##  [1] "ID.Code"        "Year"           "Gender"         "Program.Code"   "Highschool.
##  [6] "Aptitud.Verbal" "Aptitud.Matem"  "Aprov.Ingles"   "Aprov.Matem"    "Aprov.Espan
## [11] "IGS"            "Freshmen.GPA"   "Graduated"      "Year.Grad."     "Grad..GPA"
```

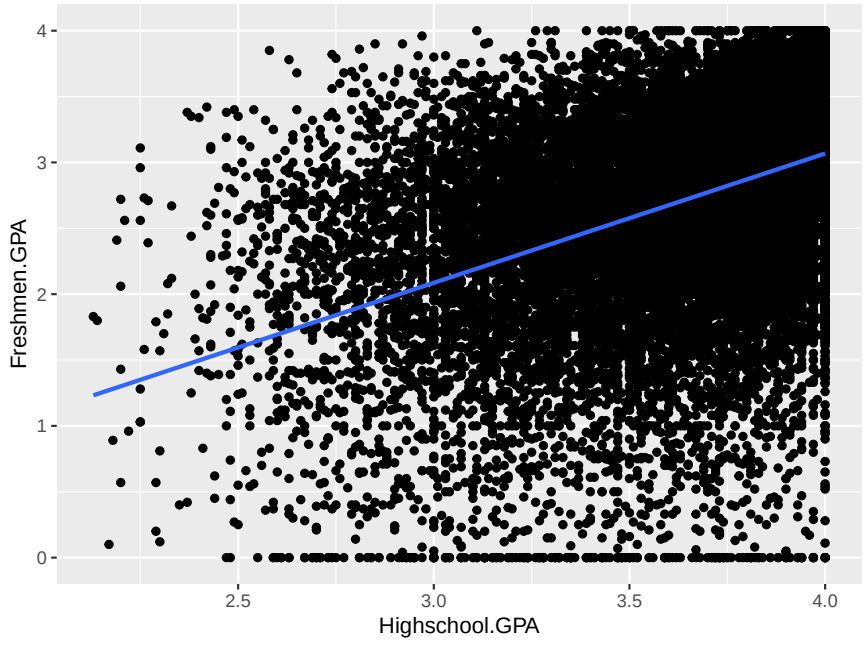
```
## [16] "Class.Facultad"
```

- Id.Code: a random code assigned to each student for purpose of privacy
- Year: the year a student applied (2003-2013)
- Gender: coded as F and M
- Program.Code: a random code assigned to each program for purpose of privacy
- Highschool.GPA
- Results of various aptitude tests
- IGS: the number used for acceptance or rejection
- Freshman.GPA
- Graduated, Year.Grad, Grad..GPA, CLass.Faculty

So, how could this data be used to tell us something about students at risk? Here are some ideas:

a. compare the high school and freshman gpa's

```
ggplot(data=upr, aes(Highschool.GPA, Freshmen.GPA)) +
  geom_point() +
  geom_smooth(method = "lm", se=FALSE)
```



It appears there is a positive correlation between these two. Of course we can find the least squares regression:

```
summary(lm(Freshmen.GPA~Highschool.GPA, data=upr))
```

```
##
## Call:
## lm(formula = Freshmen.GPA ~ Highschool.GPA, data = upr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0675 -0.3826  0.0864  0.4985  2.1763
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.85869    0.04897  -17.54   <2e-16
## Highschool.GPA  0.98155    0.01332   73.67   <2e-16
##
## Residual standard error: 0.7022 on 23449 degrees of freedom
##   (215 observations deleted due to missingness)
## Multiple R-squared:  0.188,  Adjusted R-squared:  0.1879
## F-statistic:  5428 on 1 and 23449 DF,  p-value: < 2.2e-16
```

b. Why just use one predictor? Why not use more? We can do a multiple regression analysis:

```
summary(lm(Freshmen.GPA~Highschool.GPA+Aptitud.Verbal+Aptitud.Matem, data=upr))
```

```
##
## Call:
## lm(formula = Freshmen.GPA ~ Highschool.GPA + Aptitud.Verbal +
##     Aptitud.Matem, data = upr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2938 -0.3700  0.0881  0.4825  2.0421
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1.860e+00  5.663e-02  -32.84   <2e-16
## Highschool.GPA  8.960e-01  1.328e-02   67.48   <2e-16
## Aptitud.Verbal  1.604e-03  6.893e-05   23.26   <2e-16
## Aptitud.Matem   6.023e-04  5.541e-05   10.87   <2e-16
##
## Residual standard error: 0.6862 on 23447 degrees of freedom
##   (215 observations deleted due to missingness)
## Multiple R-squared:  0.2245, Adjusted R-squared:  0.2244
## F-statistic:  2262 on 3 and 23447 DF,  p-value: < 2.2e-16
```

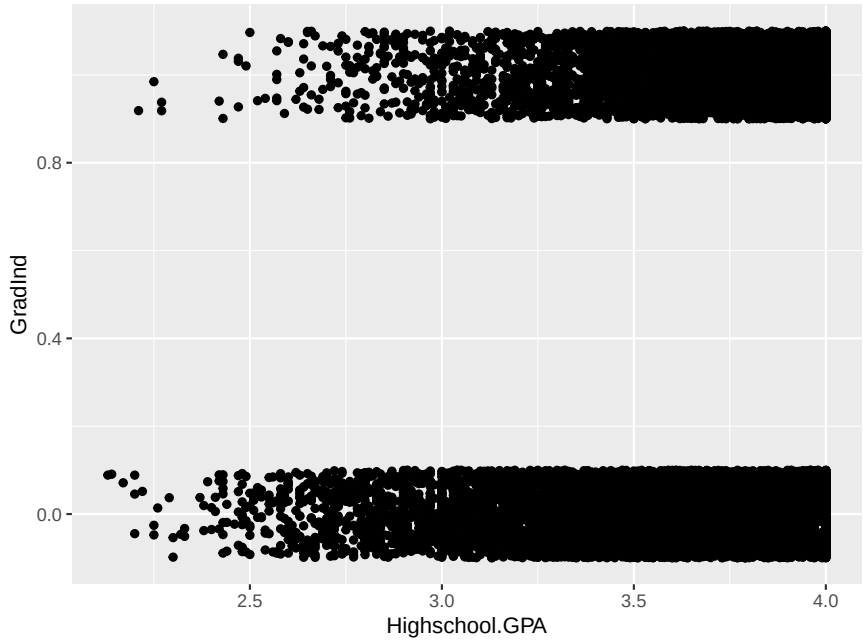
c. It might well be that what puts a student in Arts at risk is not a big problem in the sciences. So one might do an ANOVA analysis

```
summary(aov(Highschool.GPA~Class.Facultad, data = upr))
```

```
##                   Df Sum Sq Mean Sq F value Pr(>F)
## Class.Facultad     4  630.7  157.68    1714 <2e-16
## Residuals      23661 2176.7    0.09
```

d. of course the ultimate success in college is graduating, so one might be interested in the relationship of the high school GPA and whether or not somebody graduated. Of course people do need some time to do that, so we should only include the years 2003-2008 (say):

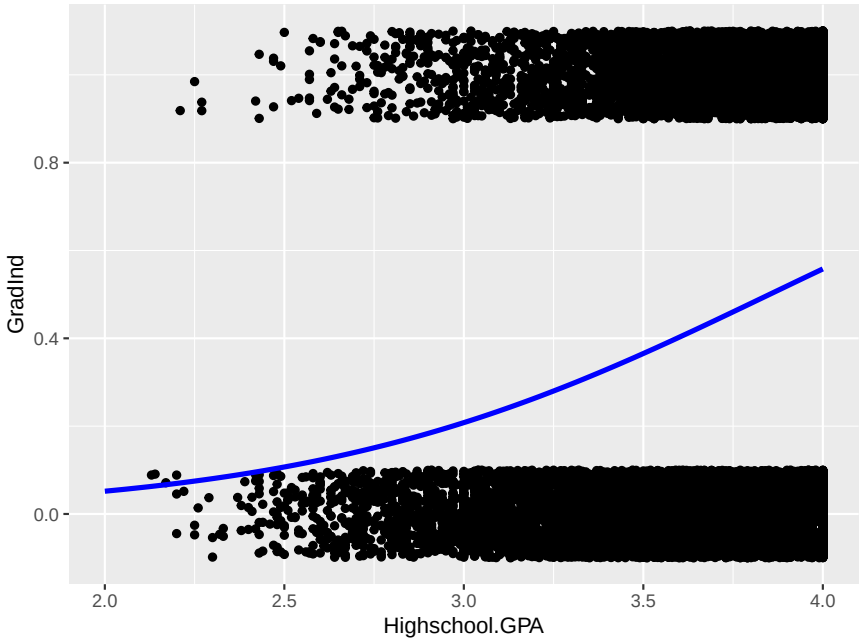
```
dta <- upr[upr$Year<=2008,
           c("Highschool.GPA", "Graduated")]
dta$GradInd <- ifelse(dta$Graduated=="Si", 1, 0)
plt <- ggplot(dta, aes(Highschool.GPA, GradInd)) +
  geom_jitter(width=0, height=0.1)
plt
```



Here the outcome variable is discrete, so a simple regression won't work. Instead one can try to predict the probability of success:

```
fit <- glm(GradInd~Highschool.GPA,
           family=binomial,
           data=dta)
x <- seq(2, 4, length=100)
df <- data.frame(x=x,
         y=predict(fit, data.frame(Highschool.GPA=x),
                                   type="response"))
plt +
  geom_line(data=df, aes(x, y),
```

```
            color="blue", size=1.2)
```



Now it turns out that all of these analyses (and many more) are all special cases of a general approach to statistics called the **Linear Model**!

2.2 A Simple Example

Let's consider the following problem: we have $y_1, .., y_n \sim N(\mu, \sigma^2)$, and we want to find a confidence interval for μ.

Let's first find a point estimate, and for that we will use the method of *least squares*, that is we will find $\hat{\mu}$ that minimizes

$$G(a) = \sum_{i=1}^{n} (y_i - a)^2$$

We find

$$\frac{dG(a)}{da} = 2\sum_{i=1}^{n}(y_i - a) = 2\sum_{i=1}^{n} y_i - 2na = 0$$
$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} y_i = \bar{y}$$

We want to find a confidence interval, and a standard method for that is to first find a hypothesis test and then invert the test. So now we want to test $H_0 : \mu = \mu_0$. Again using the least squares criteria we have $\sum_{i=1}^{n}(y_i - \mu_0)^2$, and a reasonable test statistic is given by

$$\frac{\sum_{i=1}^{n}(y_i-\mu_0)^2}{\sum_{i=1}^{n}(y_i-\hat{\mu})^2}$$

Now

$$\begin{aligned}
&\sum_{i=1}^{n}(y_i-\mu_0)^2 = \\
&\sum_{i=1}^{n}(y_i-\hat{\mu}+\hat{\mu}-\mu_0)^2 = \\
&\sum_{i=1}^{n}\left[(y_i-\hat{\mu})^2+2(y_i-\hat{\mu})(\hat{\mu}-\mu_0)+(\hat{\mu}-\mu_0)^2\right] = \\
&\sum_{i=1}^{n}(y_i-\hat{\mu})^2+2(\hat{\mu}-\mu_0)\sum_{i=1}^{n}(y_i-\hat{\mu})+\sum_{i=1}^{n}(\hat{\mu}-\mu_0)^2 = \\
&\sum_{i=1}^{n}(y_i-\hat{\mu})^2+2(\hat{\mu}-\mu_0)(\sum_{i=1}^{n}y_i-n\hat{\mu})+n(\hat{\mu}-\mu_0)^2 = \\
&\sum_{i=1}^{n}(y_i-\hat{\mu})^2+n(\hat{\mu}-\mu_0)^2
\end{aligned}$$

because $\sum_{i=1}^{n} y_i - n\hat{\mu} = 0$ by the definition of $\hat{\mu}$.

So now we have

$$\frac{\sum_{i=1}^{n}(y_i-\mu_0)^2}{\sum_{i=1}^{n}(y_i-\hat{\mu})^2} = \frac{\sum_{i=1}^{n}(y_i-\hat{\mu})^2+n(\hat{\mu}-\mu_0)^2}{\sum_{i=1}^{n}(y_i-\hat{\mu})^2} = 1+\frac{n(\hat{\mu}-\mu_0)^2}{\sum_{i=1}^{n}(y_i-\hat{\mu})^2}$$

so we can just as well use the test statistic

$$F=\frac{n(\hat{\mu}-\mu_0)^2}{\sum_{i=1}^{n}(y_i-\hat{\mu})^2}$$

Now we need to know the distribution of F. We know

$$\begin{aligned}
&E[\hat{\mu}]=\mu_0 \\
&var(\hat{\mu})=var\left(\frac{1}{n}\sum_{i=1}^{n}y_i\right)=\frac{1}{n^2}\sum_{i=1}^{n}var(y_i)=\frac{\sigma^2}{n}
\end{aligned}$$

so $\sqrt{n}(\hat{\mu}-\mu_0)/\sigma \sim N(0,1)$, and $\sqrt{n}(\hat{\mu}-\mu_0)^2/\sigma^2 \sim \chi^2(1)$

Also if the null hypothesis is true

$$\begin{aligned}
&\frac{y_i-\mu_0}{\sigma}\sim N(0,1) \\
&\frac{1}{\sigma^2}\sum_{i=1}^{n}(y_i-\mu_0)^2=\sum_{i=1}^{n}\left(\frac{y_i-\mu_0}{\sigma}\right)^2\sim\chi^2(n)
\end{aligned}$$

and from above we have

$$\frac{1}{\sigma^2}\sum_{i=1}^{n}\frac{1}{\sigma^2}(y_i-\mu_0)^2=\frac{1}{\sigma^2}\sum_{i=1}^{n}(y_i-\hat{\mu})^2+\frac{n}{\sigma^2}(\hat{\mu}-\mu_0)^2$$

Now the distribution of a sum of independent chi-square random variables is again chi-square, the term on the left is $\chi(n)$, the term on the right is $\chi^2(1)$, so we could conclude that

$$\frac{1}{\sigma^2}\sum_{i=1}^{n}(y_i-\hat{\mu})^2\sim\chi(n-1)$$

IF we knew that $\sum_{i=1}^{n}(y_i-\hat{\mu})^2$ is independent of $\hat{\mu}$. This however is a well known fact from Statistics.

Now the ratio of two independent chi-square random variables has an F distribution, and so we find

$$F=\frac{n(\hat{\mu}-\mu_0)^2}{\sum_{i=1}^{n}(y_i-\hat{\mu})^2}=\frac{n(\hat{\mu}-\mu_0)^2/\sigma^2}{\sum_{i=1}^{n}(y_i-\hat{\mu})^2/\sigma^2}\sim F(1,n-1)$$

Also, if $T\sim t(k)$, then $T^2\sim F(1,k)$, therefore

$$1-\alpha=P\left(\frac{\sqrt{n}|\hat{\mu}-\mu_0|}{\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}<t_{1-\alpha/2,n-1}\right)=$$
$$P\left(\bar{y}-t_{1-\alpha/2,n-1}\frac{s}{\sqrt{n}}\mu_0<\bar{y}+t_{1-\alpha/2,n-1}\frac{s}{\sqrt{n}}\right)$$

where $s^2=\sum_{i=1}^{n}(y_i-\bar{y})^2$, and this is of course the standard confidence interval for a normal mean with unknown standard deviation.

The crucial steps in this derivation where

- the use of the least squares criterion
- the equation

$$\sum_{i=1}^{n}(y_i-\mu_0)^2=\sum_{i=1}^{n}(y_i-\hat{\mu})^2+n(\hat{\mu}-\mu_0)^2$$

which we will write as SST=SSE+SSH (total sum of squares = error sum of squares + hypothesis sum of squares)
- the fact that SSE/σ^2 and SSH/σ^2 have χ^2 distributions
- the fact that SSE and SSH are independent.

In this course we will first show that these facts are true in great generality, and then we will apply that to many different situations.

2.3 Notation, Formulas

2.3.1 Notation

Throughout this course we will use the following notational conventions, unless otherwise indicated:

- small letters a, b, c, ..: numbers (or scalars)
- small letters i,j, k, n, m: integers
- greek letters $\alpha, \beta, \gamma, ..$: parameters
- small letters x, y, z, u, v, ...: variables
- bold face small letters $\boldsymbol{a, b, ..}$: column vectors of numbers
- bold face large letters $\boldsymbol{A, B, ..}$: matrices of numbers
- bold face greek letters $\boldsymbol{\alpha, \beta, ..}$: column vectors of parameters
- bold face large letters $\boldsymbol{X, Y, ..}$: column vectors or matrices of random variables
- exception 1: we will also use bold small letter $\boldsymbol{y}$ for vectors of random variables, rather than just $\boldsymbol{Y}$
- exception 2: bold large $\boldsymbol{X}$ for design matrix, a matrix of constants

2.3.2 Formulas (2.2.1)

Here are some formulas which we will eventually derive that then will be used extensively:

- least squares estimator of $\boldsymbol{\beta}$:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X'X})^{-1}\boldsymbol{X'y}$$

- sums of squares errors

$$\text{SSE} = \sum (y_i - \hat{y})^2 = \sum (y_i - \boldsymbol{x}_i'\hat{\boldsymbol{\beta}})^2$$

3 Probability and Statistics

3.1 Probability - Introduction

3.1.1 Basics

The probability of rain tomorrow is 0.3. What does that mean?

We usually find probabilities in one of three ways:

- empirically through many repetitions of an experiment - relative frequency interpretation
- through reasoning about outcomes etc. - classical interpretation
- by using our intuition and experience - subjective interpretation

3.1.1.1 Example (3.1.1) coin tossing

what is the probability of getting "heads" when tossing a fair coin?

- relative frequency interpretation: take a coin and flip it! the South African mathematician Jon Kerrich, while in a German POW camp during WWII tossed a coin 10000 times. Result 5067 heads, for a probability of 0.5067
- classical interpretation: This experiment has two possible outcomes - heads and tails. Fair means they are equally likely, so p=P("heads")=P("tails")=0.5
- subjective interpretation: I think it's 1/2.

An **experiment** is a well-defined procedure that produces a set of outcomes. For example, "roll a die"; "randomly select a card from a standard 52-card deck"; "flip a coin" and "pick any moment in time between 10am and 12 am" are experiments.

A **sample space** is the set of outcomes from an experiment. Thus, for "flip a coin" the sample space is {H, T}, for "roll a die" the sample space is {1, 2, 3, 4, 5, 6} and for "pick any moment in time between 10am and 12 am" the sample space is [10, 12].

An **event** is a subset, say A, of a sample space S. For the experiment "roll a die", an event is "obtain a number less than 3". Here, the event is {1, 2}.

If all the outcomes of a sample space S are equally likely and if A is an event, then the probability of A is:

$$P(A) = \frac{\#\{\text{events in A}\}}{\#\{\text{events in S}\}}$$

So, the probability of an event, say A, is the *ratio of success to total*.

3.1.1.2 Example (3.1.2) flipping a coin what is the probability of a heads?

The total number of outcomes is 2 and the number of ways to be successful is 1. Thus, P(heads) = 1/2.

3.1.1.3 Example (3.1.3) consider randomly selecting a card from a standard 52-card deck: what is the probability of getting a king?

the total number of outcomes is 52 and of these outcomes 4 would be successful. So, P(king) = 4/52.

3.1.1.4 Example (3.1.4) What is the probability of a sum of 8 when rolling two fair dice?

Solution 1: Sample space is

(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)	(1, 6)
(2, 1)	(2, 2)	(2, 3)	(2, 4)	(2, 5)	(2, 6)
(3, 1)	(3, 2)	(3, 3)	(3, 4)	(3, 5)	(3, 6)
(4, 1)	(4, 2)	(4, 3)	(4, 4)	(4, 5)	(4, 6)
(5, 1)	(5, 2)	(5, 3)	(5, 4)	(5, 5)	(5, 6)
(6, 1)	(6, 2)	(6, 3)	(6, 4)	(6, 5)	(6, 6)

There are 5 pairs that have a sum of 8, so P(sum of 10)=5/36=0.1389

Solution 2: The sum can be any number from 2 to 12, the sample space is {2,3,4,..,11,12}. There are 11 numbers in the sample space, one of them is 8, so P(sum of 10)=1/11=0.091

Which is right, and why?

Let's do a simulation to see which answer is correct. use command "sample" to randomly pick an element from a set

args(sample) shows you the correct syntax of the "sample command

sample(1:6, 2, TRUE) picks two numbers from 1 to 6 **with repetition**

sum(sample(1:6, 2, TRUE)) finds their sum, just what we want

```
z <- rep(0, 10000) #generates a vector of length 10000
for(i in 1:10000)
  z[i] <- sum(sample(1:6, 2, TRUE)) #repeats our experiment 10000 times
length(z[z==8])/10000 #finds the proportion of "8's" in z
```

```
## [1] 0.1381
```

But why is it right?

3.1.2 Fundamentals

The definition above works well as long as S is finite but breaks down if S is infinite. Instead modern probability, like geometry, is built on a small set of basic rules called axioms, derived in the 1930's by Kolmogorov. They are:

$$\text{Axiom 1: } 0 \leq P(A) \leq 1$$
$$\text{Axiom 2: } P(S) = 1$$
$$\text{Axiom 3: } P(\cup_{i=1}^{n} A_i) = \sum_{i=1}^{n} P(A_i)$$

if $A_1, ..., A_n$ are mutually exclusive

3.1.3 Some useful formulas

3.1.3.1 Theorem (3.1.5) **Complement:** $P(A) = 1 - P(A^c)$

3.1.3.2 Example (3.1.6) A fair coin is tossed 5 times. What is the probability of at least one "Heads"?

Sample Space S={(H,H,H,H,H), (H,H,H,H,T), ... , (T,T,T,T,T)}

S has $2^5 = 32$ elements

P(at least one "Heads") =
1 - P("No Heads") =
1 - P({(T,T,T,T,T)}) =
1 - 1/36 = 35/36

3.1.3.3 Theorem (3.1.7) **Addition Formula**: $P(A \cup B) = P(A) + P(B) - P(A \cap B)$

3.1.3.4 Example (3.1.8) We roll two fair dice. What is the probability of a sum of 5 or 8, or highest number on either die is a 3?

Sample Space is above.

Event A = {(1,4), (2,3), (3,2), (4,1), (2,6), (3,5), (4,4), (5,3), (6,2)}, n(A) = 9

Event B = {(1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), (3,3)}, n(B) = 9

Event $A \cap B = \{(2,3),(3,2)\}$, $n(A \cap B) = 2$

$P(A \cup B) = P(A) + P(B) - P(A \cap B) =$
$9/36 + 9/36 - 2/36 = 16/36 = 4/9$

3.2 Conditional Probability and Independence

3.2.1 Conditional Probability

3.2.1.1 Example (3.2.1) Say we pick two cards from a standard deck of 52. What is the probability both are Aces? The answer is

$$\frac{\binom{4}{2}}{\binom{52}{2}} = \frac{4!50!2!}{2!2!52!} = \frac{4 \times 3}{52 \times 51} = 0.0045$$

Now assume we know that one of the cards is an Ace. What is the probability that the other one is also an Ace? Clearly it is $3/51 = 0.0588$.

This kind of probability is called a **conditional** probability.

3.2.1.2 Definition (3.2.2) if P(B)>0 we have

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

3.2.2 Multiplication Rule

A simple manipulation of the equation above yields

$$P(A \cap B) = P(A|B)P(B)$$

3.2.2.1 Example (3.2.3) You draw two cards from a standard 52-card deck. What is the probability to draw 2 Aces?

Solution:

Let A = "First card drawn is an ace"
Let B = "Second card drawn is an ace"

Then

$$P(A \cap B) = P(A)P(B|A) = \frac{4}{52}\frac{3}{51}$$

It's easy to extend this to more than two events: What is the probability of drawing 4 aces when drawing 4 cards?

Let A_i = i^{th} card drawn is an ace"

Then

$$P(A_1 \cap A_2 \cap A_3 \cap A_4) =$$

$$P(A_1)P(A_2|A_1)P(A_3|A_1,A_2)P(A_4|A_1,A_2,A_3) = \frac{4}{52}\frac{3}{51}\frac{2}{50}\frac{1}{49}$$

3.2.3 Law of Total Probability and Bayes Rule

3.2.3.1 Definition (3.2.4) A set of events $\{A_i\}$ is called a **partition** of the sample space if

$$A_i \cap A_j = \emptyset \text{ if } i \neq j$$
$$\bigcup_{i=1}^{n} A_i = S$$

3.2.3.2 Example (3.2.5) a student is selected at random from all the undergraduate students at the Colegio

A_1 = "Student is female", A_2 = "Student is male"

or maybe

A_1 = "Student is freshman", .., A_4 = "Student is senior"

3.2.3.3 Theorem (3.2.6) law of total probability

Let B be any event and $\{A_i\}$ a **partition**, then

$$P(B) = \sum_{i=1}^{n} P(B|A_i)P(A_i)$$

3.2.3.4 Example (3.2.7) A company has 452 employees, 210 men and 242 women. 15% of the men and 10% of the women have a managerial position. What is the probability that a randomly selected person in this company has a managerial position?

Let A_1 = "person is female", A2 = "person is male"

Let B = "person has a managerial position"

Then $P(A_1)=242/452$, $P(A_2)=210/452$, $P(B|A_1)=0.1$ and $P(B|A_2)=0.15$, so

$$P(B) = P(B|A_1)P(A_1) + P(B|A_2)P(A_2) =$$
$$0.1 \times 242/452 + 0.15 \times 210/452 = 0.123$$

3.2.3.5 Theorem (3.2.8) Bayes' Formula

Let B be any event and $\{A_i\}$ a **partition**, then

$$P(A_k|B) = \frac{P(B|A_k)P(A_k)}{\sum_{i=1}^{n} P(B|A_i)P(A_i)}$$

Notice that the denominator is just the law of total probability.

3.2.3.6 Example (3.2.9) In the company above a person is randomly selected, and that person is in a managerial position. What is the probability the person is female?

$$P(A_1|B) = \frac{P(B|A_1)P(A_1)}{P(B|A_1)P(A_1) + P(B|A_2)P(A_2)} =$$
$$\frac{0.1 \times 242/452}{0.123} = 0.434$$

Bayes' Rule plays a very important role in Statistics and in Science in general. It provides a natural method for updating you knowledge based on data.

3.2.4 Independence

Sometimes knowing that one event occurred does not effect the probability of another event. For example if you throw a red and a blue die, knowing that the red die shows a "6" will not change the probability that the blue die shows a "2".

3.2.4.1 Theorem (3.2.10) Say A and B are two events, then A and B are said to be **independent** iff

$$P(A \cap B) = P(A)P(B)$$

3.2.4.2 Example (3.2.11) Say you flip a fair coin 5 times. What is the probability of 5 "heads"?

Let A_i = i^{th} flip is heads

Now it is reasonable to assume that the A_i's are independent and so

$$P(A_1 \cap .. \cap A_5) =$$
$$P(A_1) \times .. \times P(A_5) =$$
$$\frac{1}{2} \times .. \times \frac{1}{2} = \frac{1}{2^5}$$

3.3 Random Variable, Distribution Function, Density and Random Vectors

3.3.1 Random Variables

3.3.2 Random Variable

3.3.2.1 Definition (3.3.1) A **random variable** (r.v.) X is set-valued function from the sample space into the real numbers.

3.3.2.2 Example (3.3.2) We roll a fair die, X is the number shown on the die

3.3.2.3 Example (3.3.3) We roll a fair die, X is 1 if the die shows a six, 0 otherwise.

3.3.2.4 Example (3.3.4) We roll a a fair die until the the first "6", X is the number of rolls needed.

3.3.2.5 Example (3.3.5) We randomly pick a time between 10am and 12 am, X is the minutes that have passed since 10am.

3.3.2.6 Definition (3.3.6)

- If X takes countably many values, X is called a **discrete** r.v.
- If X takes uncountably many values, X is called a **continuous** r.v.

There are also mixtures of these two.

In the first three examples above X is discrete, in the last one X is continuous.

There are some technical difficulties when defining a r.v. on a sample space like $\mathbb{R}$, it turns out to be impossible to define it for every subset of $\mathbb{R}$ without getting logical contradictions. The solution is to define a **σ-algebra** on the sample space and then define X only on that σ-algebra. We will ignore these technical difficulties.

Almost everything to do with r.v.'s has to be done twice, once for discrete and once for continuous r.v.'s. This separation is only artificial, it goes away once a more general definition of "integral" is used (Riemann-Stilties or Lebesgue)

3.3.3 (Commulative) Distribution Function

3.3.3.1 Definition (3.3.7) The distribution function of a r.v. X is defined by

$F(x) = P(X \leq x)\ \forall x \in \mathbb{R}$

3.3.3.2 Example (3.3.8) We roll a fair die, X is the number shown on the die. Say x=2.2, then

$F(2.2) = P(X \leq 2.2) = P(1,2) = 2/6 = 1/3$

3.3.3.3 Example (3.3.9) We randomly pick a time between 10am and 12 am, X is the minutes that have passed since 10am. Say x=67.5, then

$F(67.5) = P(X \leq 67.5) =$ P(we chose a moment between 10am and 11h7.5min am) $= 67.5/120 = 0.5625$

3.3.3.4 Theorem (3.3.10) Some features of cdf's:

1. cdf's are standard functions on $\mathbb{R}$

2. $0 \leq F(x) \leq 1$

3. cdf's are non-decreasing

4. cdf's are right-continuous

5. $$F(x) \to 0 \text{ as } x \to -\infty$$
$$F(x) \to 1 \text{ as } x \to \infty$$

3.3.3.5 Example (3.3.11) We roll a a fair die until the the first "6", X is the number of rolls needed. Let's find the cdf F.

note $X \in \{1, 2, 3, ...\}$

let A_i be the event "a six on the i^{th} roll", i=1,2,3, ... Then

$$P(X = k) = P(A_1^c \cap .. \cap A_{k-1}^c \cap A_k) =$$
$$P(A_1^c) \times .. \times P(A_{k-1}^c)P(A_k) ==$$
$$\frac{5}{6} \times .. \times \frac{5}{6}\frac{1}{6} = (\frac{5}{6})^{k-1}\frac{1}{6}$$
$$P(X \leq k) = \sum_{i=1}^{k} P(X = i) =$$
$$\sum_{i=1}^{k} (\frac{5}{6})^{i-1}\frac{1}{6} =$$
$$\frac{1}{6}\sum_{j=0}^{k-1} (\frac{5}{6})^{j} =$$
$$\frac{1}{6}\frac{1-(5/6)^{k-1+1}}{1-5/6} = 1-(5/6)^k$$

so for $k \leq x < k+1$ we have $F(x) = 1-(5/6)^k$

3.3.4 Density

3.3.4.1 Definition (3.3.12) The probability density function of a discrete r.v. X is defined by $f(x) = P(X = x)$

Note:

$f(x) = P(X = x) = P(X \leq x) - P(X \leq x-1) = F(x) - F(x-1)$

3.3.4.2 Example (3.3.13) the pdf of X in the example above is given by

$f(x) = 1/6 * (5/6)^{x-1}$ if $x \in \{1, 2, ..\}$, 0 otherwise.

Note that it follows from the definition and the axioms that for any density we have

$$f(x) \geq 0$$
$$\sum_x f(x) = 1$$

3.3.4.3 Definition (3.3.14) f is the of a continuous random variable with cdf F if

$F(x) = \int_{-\infty}^{x} f(t)dt$

Again it follows from the definition and the axioms that for any density f we have

$$f(x) \geq 0$$
$$\int_{-\infty}^{\infty} f(x)dx = 1$$

3.3.4.4 Example (3.3.15) Show that $f(x) = \lambda \exp(-\lambda x)$ if x>0, 0 otherwise defines a pdf, where $\lambda > 0$.

clearly $f(x) \geq 0$ for all x.

$$\int_{-\infty}^{\infty} f(x)dx =$$
$$\int_{0}^{\infty} \lambda \exp(-\lambda t)dt =$$
$$-\exp(-\lambda t)|_0^{\infty} = 0 - (-1) = 1$$

This r.v. X is called an *exponential* r.v. with rate λ.

3.3.5 Random Vectors

A random vector is a multi-dimensional random variable.

3.3.5.1 Example (3.3.16) we roll a fair die twice. Let X be the sum of the rolls and let Y be the absolute difference between the two roles. Then (X,Y) is a 2-dimensional random vector. The joint of (X,Y) is given by:

	0	1	2	3	4	5
2	1	0	0	0	0	0
3	0	2	0	0	0	0
4	1	0	2	0	0	0
5	0	2	0	2	0	0
6	1	0	2	0	2	0
7	0	2	0	2	0	2
8	1	0	2	0	2	0
9	0	2	0	2	0	0
10	1	0	2	0	0	0
11	0	2	0	0	0	0
12	1	0	0	0	0	0

where every number is divided by 36.

All definitions are straightforward extensions of the one-dimensional case.

3.3.5.2 Example (3.3.17) for a discrete random vector we have the $f(x, y) = P(X = x, Y = y)$.

Say above

f(4,0) =
P(X=4, Y=0) =
P({(2,2)}) = 1/36

or

f(7,1) =
P(X=7,Y=1) =
P({(3,4),(4,3)}) = 1/18

3.3.5.3 Example (3.3.18) Say $f(x,y) = cxy, 0 \leq x < y \leq 1$ is a pdf. Find c.

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)dxdy =$$
$$\int_0^1 \int_0^y cxydxdy =$$
$$\int_0^1 cy\left[x^2/2|_0^y\right]dy =$$
$$\int_0^1 cy\left[y^2/2\right]dy =$$
$$\int_0^1 cy^3/2dy =$$
$$cy^4/8|_0^1 = c/8 = 1$$

so c=8.

3.3.5.4 Definition (3.3.19) Say (X,Y) is a discrete (continuous) r.v. with joint density f. Then the **marginal** density f_X is given by

$$f_X(x) = \sum_y f(x,y) \text{ if Y is discete}$$
$$f_X(x) = \int_{-\infty}^{\infty} f(x,y)dy \text{ if Y is continuous}$$

3.3.5.5 Example (3.3.20) For the discrete example above we find

$f_X(2) = f(2,0) + f(2,1) + .. + f(2,5) = 1/36$

or

$f_Y(3) = 6/36$

3.3.5.6 Example (3.3.21) Say $f(x,y) = 8xy, 0 \leq x < y \leq 1$, find $f_Y(y)$

$$f_X(x) = \int_{-\infty}^{\infty} f(x,y)dy =$$
$$\int_x^1 8xydy =$$
$$4xy^2|_x^1 = 4x - 4x^3; 0 < x < 1$$

Note that f_X is s proper density.

3.3.6 Conditional R.V.'s

3.3.6.1 Definition (3.3.22) let (X,Y) be a discrete r.v. with joint f(x,y) and marginals f_X and f_Y. For any x such that $f_X(x) > 0$ the conditional $f_{Y|X=x}(y|x)$ is defined by

$$f_{Y|X=x}(y|x) = \frac{f(x,y)}{f_Y(y)}$$

3.3.6.2 Example (3.3.23) For the example above find $f_{X|Y=y}(x|y)$

$$f_{X|Y=y}(x|y) =$$
$$\frac{f(x,y)}{f_Y(y)} =$$
$$\frac{8xy}{4y^3} = \frac{2x}{y^2}$$

for $0 \leq x \leq y$.

Here y is a fixed number!

Note that a conditional density requires a specification for a value of the random variable on which we condition, something like $f_{X|Y=y}$. An expression like $f_{X|Y}$ is not defined!

3.3.7 Independence

3.3.7.1 Definition (3.3.24) Two r.v. X and Y are said to be independent iff

$f_{X,Y}(x,y) = f_X(x)f_Y(y)$

3.4 Expectation

3.4.1 Expectation of a Random Variable

3.4.1.1 Definition (3.4.1) The *expectation* (or expected value) of a random variable g(X) is defined by

$$\sum_x g(x)f(x) \text{ if X discrete}$$
$$\int_{-\infty}^{\infty} g(x)f(x)dx \text{ if X continuous}$$

We use the notation E[g(X)]

3.4.1.2 Example (3.4.2) we roll fair die until the first time we get a six. What is the expected number of rolls?

We saw that f(x) = 1/6*(5/6)$^{x-1}$ if

Here we just have g(x)=x, so

$$E[X] = \sum_{i=1}^{\infty} g(x_i)f(x_i) = \sum_{i=1}^{\infty} i\frac{1}{6}(\frac{5}{6})^{i-1}$$

How do we compute this sum? Here is a "standard" trick:

$$\sum_{k=1}^{\infty} kt^{k-1} =$$
$$\sum_{k=1}^{\infty} \frac{dt^k}{dt} =$$
$$\frac{d}{dt}\sum_{k=1}^{\infty} t^k =$$
$$\frac{d}{dt}\left[\sum_{k=0}^{\infty} t^k - 1\right] =$$
$$\frac{d}{dt}\left[\frac{1}{1-t} - 1\right] =$$
$$\frac{1}{(1-t)^2}$$

and so we find

$$E[X] = \frac{1}{6}\frac{1}{(1-5/6)^2} = 6$$

3.4.1.3 Example (3.4.3) X is said to have a *uniform* [A,B] distribution if f(x)=1/(B-A) for A<x<B, 0 otherwise.

Find E[X^k] (this is called the kth moment of X).

$$E[X^k] = \int_{-\infty}^{\infty} x^k f(x)dx =$$
$$\int_A^B x^k \frac{1}{B-A}dx =$$
$$\frac{1}{B-A}\frac{x^{k+1}}{k+1}|_A^B =$$
$$\frac{B^{k+1}-A^{k+1}}{(k+1)(B-A)} =$$
$$\frac{(B-A)\sum_{i=0}^{k} A^i B^{k-i}}{(k+1)(B-A)} =$$
$$\frac{\sum_{i=0}^{k} A^i B^{k-i}}{k+1}$$

---

some special expectations are the **mean** of X defined by $\mu = EX$ and the **variance** defined by $\sigma^2 = V(X) = E(X-\mu)^2$. Related to the variance is the **standard deviation** σ, the square root of the variance.

3.4.1.4 Theorem (3.4.4)

i. E[aX+b]=aE[X]+b

ii. E[X+Y]=E[X]+E[Y]

iii. var(aX+b)=a^2var(X)

iv. var(X)=E[X^2]-(E[X])^2

the last one is a useful formula for finding the variance and/or the standard deviation.

3.4.1.5 Example (3.4.5)

find the mean and the standard deviation of a uniform [A,B] r.v.

$$E[X] = \frac{\sum_{i=0}^{1} A^i B^{k-i}}{1+1} = \frac{A+B}{2}$$
$$E[X^2] = \frac{\sum_{i=0}^{2} A^i B^{k-i}}{2+1} = \frac{A^2+AB+B^2}{3}$$
$$var(X) = \frac{A^2+AB+B^2}{3} - (\frac{A+B}{2})^2 = \frac{(B-A)^2}{12}$$

and so $\sigma = (B-A)/\sqrt{12}$

3.4.1.6 Example (3.4.6)

Find the mean and the standard deviatiton of an exponential rv with rate λ.

$$E[X^k] = \int_{-\infty}^{\infty} x^k f(x)dx =$$
$$\int_0^{\infty} x^k \lambda e^{-\lambda x} dx =$$
$$-x^k e^{-\lambda x}|_0^{\infty} - \int_0^{\infty} -kx^{k-1} e^{-\lambda x} dx =$$
$$k \int_0^{\infty} x^{k-1} e^{-\lambda x} dx =$$
$$\frac{k}{\lambda} \int_0^{\infty} x^{k-1} \lambda e^{-\lambda x} dx =$$
$$\frac{k}{\lambda} E[X^{k-1}]$$
$$\mu = E[X] = \frac{1}{\lambda} E[X^0] = \frac{1}{\lambda}$$
$$E[X^2] = \frac{2}{\lambda} E[X] = \frac{2}{\lambda^2}$$
$$var(X) = E[X^2] - (E[X])^2 =$$
$$\frac{2}{\lambda^2} - (\frac{1}{\lambda})^2 = \frac{1}{\lambda^2}$$

One way to "link" probabilities and expectations is via the indicator function I_A defined as

$$I_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$$

because with this we have for a continuous r.v. X with f:

$$E[I_A(X)] = \int_{-\infty}^{\infty} I_A(x) f(x) dx = \int_A f(x) dx = P(X \in A)$$

3.4.2 Expectations of Random Vectors

The definition of expectation easily generalizes to random vectors:

3.4.2.1 Example (3.4.7) Let (X,Y) be a discrete random vector with
$f(x, y) = 8xy, 0 \leq x \leq y \leq 1$
Find $E[XY]$

$$E[XY] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} xyf(x,y)dxdy =$$
$$\int_0^1\int_0^y 8xyxydxdy =$$
$$\int_0^1\int_0^y 8x^2y^2dxdy =$$
$$\int_0^1 8y^2\left[x^3/3|_0^y\right]dy =$$
$$\int_0^1 8y^2\left[y^3/3\right]dy =$$
$$\int_0^1 8y^5/3dy =$$
$$4y^6/9|_0^1 = 4/9$$

3.4.3 Covariance and Correlation

3.4.3.1 Definition (3.4.8) The covariance of two r.v. X and Y is defined by

$cov(X,Y) = E[(X-\mu_X)(Y-\mu_Y)]$

The correlation of X and Y is defined by

$cor(X,Y) = \frac{cov(X,Y)}{\sigma_X\sigma_Y}$

Note cov(X,X) = var(X)

As with the variance we have a simpler formula for actual calculations:

3.4.3.2 Theorem (3.4.9) $cov(X,Y) = E(XY) - (EX)(EY)$

3.4.3.3 Example (3.4.10) take the example of the sum and absolute value of the difference of two rolls of a die. What is the covariance of X and Y?

So we have

$\mu_X = E[X] = 2*1/36 + 3*2/36 + ... + 12*1/36 = 7.0$
$\mu_Y = E[Y] = 0*6/36 + 1*12/36 + ... + 5*2/36 = 70/36$
$E[XY] = 0*2*1/36 + 1*2*0/36 + .2*2*0/36.. + 5*12*0/36 = 490/36$

and so

$cov(X,Y) = EXY - EXEY = 490/36 - 7.0*70/36 = 0$

Note that we previously saw that X and Y are **not** independent, so we here have an example that a covariance of 0 does **not** imply independence! It does work the other way around, though:

3.4.3.4 Theorem (3.4.11) If X and Y are independent, then cov(X,Y) = 0 (= cor(X,Y))

proof (in the case of X and Y continuous):

$$E[XY] = \iint_{R^2} xyf(x,y)d(x,y) =$$
$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} xyf(x,y)dxdy =$$
$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} xyf_X(x)f_Y(y)dxdy =$$
$$\int_{-\infty}^{\infty} yf_Y(y)\left(\int_{-\infty}^{\infty} xf_X(x)dx\right)dy =$$
$$\left(\int_{-\infty}^{\infty} xf_X(x)dx\right)\left(\int_{-\infty}^{\infty} yf_Y(y)dy\right) =$$
$$E[X]E[Y]$$

and so cov(X,Y) = EXY-EXEY = EXEY - EXEY = 0

3.4.3.5 Example (3.4.12) we have continuous rv's X and Y with joint
$f(x,y) = 8xy, 0 \le x < y \le 1$
Find the covariance and the correlation of X and Y.

$$E[X^kY^j] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^k y^j f(x,y)dxdy =$$
$$\int_0^1\int_0^y 8x^k y^j xydxdy =$$
$$\int_0^1\int_0^y 8x^{k+1}y^{j+1}dxdy =$$
$$\int_0^1 8y^{j+1}\left[x^{k+2}/(k+2)|_0^y\right]dy =$$
$$\int_0^1 8y^{j+1}\left[y^{k+2}/(k+2)\right]dy =$$
$$\int_0^1 8y^{j+k+3}/(k+2)dy =$$
$$8y^{j+k+4}/(k+2)/(j+k+4)|_0^1 =$$
$$\frac{8}{(k+2)(j+k+4)}$$

therefore

$$E[X] = E[X^1Y^0] = \frac{8}{(1+2)(1+0+4)} = \frac{8}{15}$$
$$E[X^2] = E[X^2Y^0] = \frac{8}{(2+2)(2+0+4)} = \frac{1}{3}$$
$$var(X) = E[X^2] - (E[X])^2 = \frac{1}{3} - (\frac{8}{15})^2 = \frac{11}{225}$$
$$E[Y] = E[X^0Y^1] = \frac{8}{(0+2)(0+1+4)} = \frac{4}{5}$$
$$E[Y^2] = E[X^0Y^2] = \frac{8}{(0+2)(0+2+4)} = \frac{2}{3}$$
$$var(Y) = E[Y^2] - (E[Y])^2 = \frac{2}{3} - (\frac{5}{5})^2 = \frac{12}{75}$$
$$E[XY] = E[X^1Y^1] = \frac{8}{(1+2)(1+1+4)} = \frac{4}{9}$$
$$cov(X,Y) = E[XY] - E[X]E[Y] = \frac{4}{9} - \frac{8}{15}\frac{4}{5} = \frac{12}{675}$$
$$cor(X,Y) = \frac{cov(X,Y)}{\sqrt{var(X)var(Y)}} = 0.492$$

We saw above that E[X+Y] = E[X] + E[Y]. How about var(X+Y)?

3.4.3.6 Theorem (3.4.13)

$$var(X+Y) = var(X) + var(Y) + 2cov(X,Y)$$

if $X \perp Y$ we have var(X+Y) = var(X) + var(Y)

proof

$$var(X+Y) = E[(X+Y)^2] - (E[X+Y])^2 =$$
$$E[X^2 + 2XY + Y^2] - (E[X]^2 + 2E[X]E[Y] + E[Y]^2) =$$
$$E[X^2] + 2E[XY] + E[Y^2] - E[X]^2 - 2E[X]E[Y] - E[Y]^2 =$$
$$(E[X^2] - E[X]^2) + (E[Y^2] - E[Y]^2) + 2E([XY] - 2E[X]E[Y]) =$$
$$var(X) + var(Y) + 2cov(X,Y)$$

3.4.4 Conditional Expectation and Variance

3.4.4.1 Definition (3.4.14) Say X|Y=y is a conditional r.v. with density f. Then the conditional expectation of X|Y=y is defined by

$$E[g(X)|Y=y] = \sum_x g(x) f_{X|Y=y}(x|y) \text{ if X discrete}$$
$$E[g(X)|Y=y] = \int_{-\infty}^{\infty} g(x) f_{X|Y=y}(x|y)dx \text{ if X continuous}$$

Let E[X|Y] denote the function of the random variable Y whose value at Y=y is given by E[X|Y=y]. Note then Z=E[X|Y] is itself a random variable.

3.4.4.2 Example (3.4.15) An urn contains 2 white and 3 black balls. We pick two balls from the urn. Let X be denote the number of white balls chosen. An additional ball is drawn from the remaining three. Let Y equal 1 if the ball is white and 0 otherwise.

For example

$f(0,0) = P(X=0, Y=0) = 3/5 * 2/4 * 1/3 = 1/10.$

The complete is given by:

	x=0	x=1	x=2
y=0	0.1	0.4	0.1
y=1	0.2	0.2	0.0

The marginals are given by

	x	P(X=x)
1	x=0	0.3
2	x=1	0.6
3	x=2	0.1

	y	P(Y=y)
1	y=0	0.6
2	y=1	0.4

The conditional distribution of X|Y=0 is

	x	P(X=x|Y=0)
1	0	1/6
2	1	2/3
3	2	1/6

and so $E[X|Y=0] = 0 * 1/6 + 1 * 2/3 + 2 * 1/6 = 1.0.$

The conditional distribution of X|Y=1 is

	x	P(X=x|Y=1)
1	0	1/2
2	1	1/2
3	2	0

and so $E[X|Y=1] = 0*1/2 + 1*1/2 + 2*0 = 1/2$.

Finally the conditional r.v. Z = E[X|Y] has

	z	P(Z=z)
1	1	3/5
2	1/2	2/5

with this we can find $E[Z] = E[E[X|Y]] = 1*3/5 + 1/2*2/5 = 4/5$.

How about using simulation to do these calculations? - program **urn1**

```
urn1 <- function (n = 2, m = 3, draws = 2, B = 10000) {
   u <- c(rep("w", n), rep("b", m))
   x <- rep(0, B)
   y <- x
   for (i in 1:B) {
       z <- sample(u, draws + 1)
       y[i] <- ifelse(z[draws + 1] == "w", 1, 0)
       for (j in 1:draws)
         x[i] <- x[i] + ifelse(z[j] == "w", 1, 0)
   }
   print("Joint pdf:")
   print(round(table(y, x)/B, 3))
   print("pdf of X:")
   print(round(table(x)/B, 3))
   print("pdf of Y:")
   print(round(table(y)/B, 3))
   print("pdf of X|Y=0:")
   x0 <- table(x[y == 0])/length(y[y == 0])
   print(round(x0, 3))
   print("E[X|Y=0]:")
   print(sum(c(0:draws) * x0))
   print("pdf of X|Y=1:")
   x1 <- table(x[y == 1])/length(y[y == 1])
   print(round(x1, 3))
   print("E[X|Y=1]:")
   print(sum(c(0:1) * x1))

}
```

```
urn1()
```

```
## [1] "Joint pdf:"
##    x
## y       0     1     2
##   0 0.098 0.401 0.103
##   1 0.197 0.202 0.000
## [1] "pdf of X:"
## x
##     0     1     2
## 0.294 0.603 0.103
## [1] "pdf of Y:"
## y
##     0     1
## 0.601 0.399
## [1] "pdf of X|Y=0:"
##
##     0     1     2
## 0.163 0.666 0.171
## [1] "E[X|Y=0]:"
## [1] 1.008314
## [1] "pdf of X|Y=1:"
##
##     0     1
## 0.493 0.507
## [1] "E[X|Y=1]:"
## [1] 0.5067737
```

3.4.4.3 Example (3.4.16) We have continuous rv's X and Y with joint $f(x,y) = 8xy, 0 \leq x < y \leq 1$. We have found $f_Y(y) = 4y^3, 0 < y < 1$, and $f_{X|Y=y}(x|y) = 2x/y^2, 0 \leq x \leq y$. So

$$f_Y(y) = \int_0^y 8xydx = 4x^2y|_0^y = 4y^3; 0 < y < 1$$
$$f_{X|Y=y}(x|y) = \frac{f(x,y)}{f_Y)y} = \frac{8xy}{4y^3} = \frac{2x}{y^2}; 0 < x < y$$
$$E[X|Y=y] = \int_{-\infty}^{\infty} x f_{X|Y=y}(x|y)dx = \int_0^y x\frac{2x}{y^2}dx =$$
$$\frac{2}{y^2}\int_0^y x^2dx = \frac{2}{4y^2}[x^3/3|_0^y = \frac{2y^3}{3y^2} = \frac{2y}{3}$$

Throughout this calculation we treated y as a constant. Now, though, we can change our point of view and consider $E[X|Y=y] = 2y/3$ as a function of y:

$g(y) = E[X|Y=y] = 2y/3$

What are the values of y? Well, they are the observations we might get from the rv. Y, so we can also write

$g(Y) = E[X|Y = Y] = 2Y/3$

but Y is a rv, then so is 2Y/3, and we see that we can define a rv Z=g(Y)=E[X|Y].

Recall that the expression $f_{X|Y}$ does not make sense. Now we see that on the other hand the expression E[X|Y] makes perfectly good sense!

There are very useful formulas for the expectation and variance of conditional r.v.s:

3.4.4.4 Theorem (3.4.17)

We have

i. $E[X] = E\{E[X|Y\}]$

ii. $var(X) = E[var(X|Y)] + var(E[X|Y])$

3.4.4.5 Example (3.4.18)

Say $Y \sim U[0,1]$ and $X|Y = y \sim Exp(y+1)$, then

$$E[X|Y = y] = \frac{1}{y+1}$$

$$E[X] = E\{E[X|Y\}] = E[\frac{1}{Y+1}] =$$

$$\int_0^1 \frac{1}{y+1} dy = \log(y+1)|_0^1 = \log 2$$

$$var(X|Y = y) = \frac{1}{(y+1)^2}$$

$$var(X) = E[var(X|Y)] + var(E[X|Y]) =$$

$$E[\frac{1}{(Y+1)^2}] + var(\frac{1}{Y+1}) =$$

$$E[\frac{1}{(Y+1)^2}] + E[(\frac{1}{Y+1})^2] - (E[\frac{1}{Y+1}])^2 =$$

$$2E[\frac{1}{(Y+1)^2}] - (\log 2)^2 =$$

$$2\int_0^1 \frac{1}{(y+1)^2} dy - (\log 2)^2 =$$

$$2[-\frac{1}{y+1}|_0^1 - (\log 2)^2 =$$

$$2[1 - \frac{1}{2}] - (\log 2)^2 =$$

$$1 - (\log 2)^2 = 0.52$$

let's check:

```
y=runif(1e5)
x=rexp(1e5, y+1)
round(c(log(2), mean(x)), 3)
```

```
## [1] 0.693 0.692
```

```
round(c(1-log(2)^2,var(x)), 3)
```

```
## [1] 0.520 0.514
```

3.5 Inequalities and Limit Theorems

3.5.1 Two very useful inequalities

3.5.1.1 Theorem (3.5.1) Markov's Inequality

If X takes on only nonnegative values, then for any a>0

$$P(X \geq a) \leq \frac{EX}{a}$$

proof omitted

3.5.1.2 Theorem (3.5.2) Chebyshev's Inequality:

If X is a r.v. with mean μ and variance σ^2, then for any k>0:

$$P(|X - \mu| \geq k\sigma) \leq 1/k^2$$

proof

$$P(|X - \mu| \geq k\sigma) =$$
$$P((X - \mu)^2 \geq k^2\sigma^2) \leq$$
$$\frac{E(X - \mu)^2}{k^2\sigma^2} = \frac{\sigma^2}{k^2\sigma^2} = 1/k^2$$

3.5.1.3 Example (3.5.3) Consider the uniform random variable with f(x) = 1 if $0 < x < 1$, 0 otherwise. We already know that $\mu = 0.5$ and $\sigma = 1/\sqrt{12} = 0.2887$. Now Chebyshev says

$P(|X - 0.5| > k0.2887) \leq 1/k^2$

For example

$P(|X - 0.5| > 0.2887) \leq 1$ (rather boring!)

or

$P(|X - 0.5| > 3 \times 0.2887) \leq 1/9$

actually $P(|X - 0.5| > 0.866) = 0$, so this is not a very good upper bound.

3.5.2 Law of Large Numbers, Convergence in Probability

3.5.2.1 Theorem (3.5.4) (Weak) Law of Large Numbers

Let $X_1, X_2, ...$ be a sequence of independent and identically distributed (iid) r.v.'s having mean μ. Then for all $\epsilon > 0$

$$P(|\frac{1}{n}\sum X_i - \mu| > \epsilon) \to 0$$

proof (assuming in addition that $V(X_i) = \sigma^2 < \infty$

$$E[\frac{1}{n}\sum X_i] = \frac{1}{n}\sum E[X_i] = \mu$$
$$V[\frac{1}{n}\sum X_i] = \frac{1}{n^2}\sum V[X_i] = \frac{\sigma^2}{n}$$
$$P(|\frac{1}{n}\sum X_i - \mu| > \epsilon) =$$
$$P(|\frac{1}{n}\sum X_i - \mu| > \frac{\epsilon}{\sigma/\sqrt{n}}\sigma/\sqrt{n}) \leq$$
$$1/(\frac{\epsilon}{\sigma/\sqrt{n}}) = \frac{\sigma}{\epsilon\sqrt{n}} \to 0$$

This theorem forms the bases of (almost) all simulation studies: say we want to find a parameter θ of a population. We can generate data from a random variable X with pdf () $f(x|\theta)$ such that $Eh(X) = \theta$. Then by the law of large numbers

$$\frac{1}{n}\sum h(X_i) \to \theta$$

3.5.2.2 Example (3.5.5)

in a game a player rolls 5 fair dice. He then moves his game piece along k fields on a board, where k is the smallest number on the dice + largest number on the dice. For example if his dice show 2, 2, 3, 5, 5 he moves 2+5 = 7 fields. What is the mean number of fields θ a player will move?

To do this analytically would be quite an exercise. To do it via simulation is easy:

Let X be an independent random vector of length 5, with $X[j] \in 1, .., 6$ and $P(X[j] = k) = 1/6$. Let $h(x) = \min(x) + \max(x)$, then $Eh(X) = \theta$.

Let $X_1, X_2, ..$ be iid copies of X, then by the law of large numbers

```
B <- 1e5
z <- rep(0, B)
for (i in 1:B) {
  x <- sample(1:6, size = 5, replace = TRUE)
  z[i] <- min(x)+max(x)
}
mean(z)
```

```
## [1] 6.98824
```

3.5.3 Central Limit Theorem

This is one of the most famous theorems in all of mathematics / statistics. Without it, Statistics as a science would not have existed until very recently:

We first need the definition of a normal (or Gaussian) r.v.:

A random variable X is said to be normally distributed with mean μ and standard deviation σ if it has :

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

If $\mu = 0$ and $\sigma = 1$ we say X has a standard normal distribution.

We use the symbol Φ for the distribution function of a standard normal r.v.

3.5.3.1 Theorem (3.5.6) Let $X_1, X_2, ..$ be an iid sequence of r.v.'s with mean μ and standard deviation σ. Let $\bar{X} = \frac{1}{n}\sum X$. Then

$$P(\frac{\bar{X}-\mu}{\sigma/\sqrt{n}} \leq z) \to \Phi(z)$$

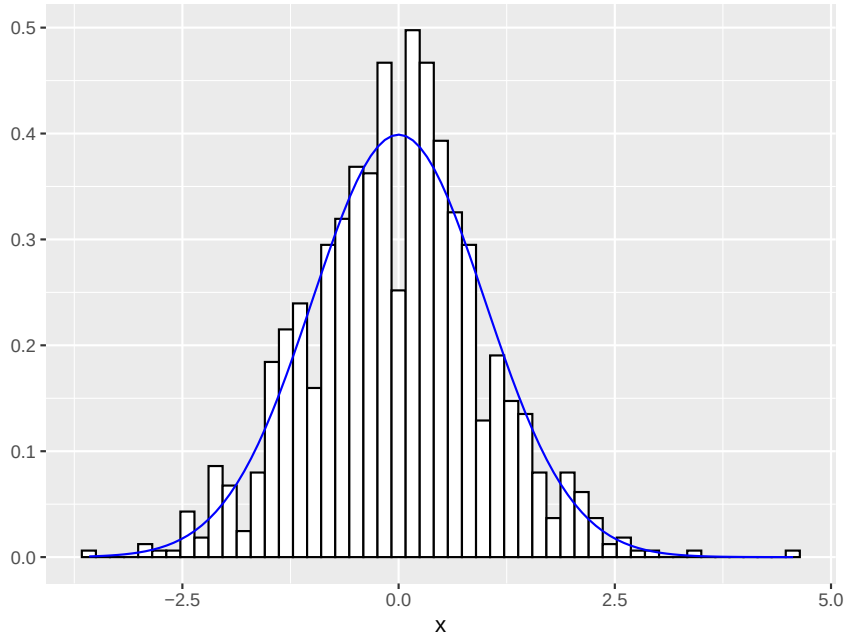
3.5.3.2 Example (3.5.7) Let's do a simulation to illustrate the CLT: we will use the most basic r.v. of all, called a Bernoulli r.v. which has $P(X = 0) = 1 - p$ and $P(X = 1) = p$. (Think indicator function for the coin toss}. So we sample n Bernoulli r.v. with "success parameter p" and find their sample mean. Note that

$$E(X) = p$$
$$V(X) = p(1-p)$$

```
cltexample1 <- function (p, n, B=1000) {
  xbar <- rep(0, n)
  for (i in 1:B) {
    xbar[i] <- mean(sample(c(0, 1), n, TRUE, prob=c(1-p,  p)))
    }
  df <- data.frame(x=sqrt(n)*(xbar-p)/sqrt(p*(1-p)))
  bw <- diff(range(df$x))/50
  ggplot(df, aes(x)) +
    geom_histogram(aes(y = ..density..),
      color = "black",
      fill = "white",
      binwidth = bw) +
      labs(x = "x", y = "") +
```

```
    stat_function(fun = dnorm, colour = "blue",
                  args=list(mean=0, sd=1))

}
cltexample1(0.5, 500)
```



3.6 Functions of a R.V. - Transformations

3.6.0.1 Example (3.6.1) say $X \sim U[0,1]$ and $\lambda > 0$. What is the pdf of the random variable $Y = -\lambda \log(X)$?

Solution: we first find the cdf and then the pdf as follows:

$$
\begin{aligned}
&F_Y(y) = P(Y < y) = \\
&P(-\lambda \log X < y) = \\
&P(\log X > -y/\lambda) = \\
&P(X > \exp(-y/\lambda)) = \\
&1 - P(X < \exp(-y/\lambda)) = \\
&1 - \exp(-y/\lambda) \\
&f_Y(y) = \frac{dF_Y}{dy} = \lambda \exp(-y/\lambda)
\end{aligned}
$$

if y>0. For y<0 note that $P(-\log X < y) = 0$ because $0 < X < 1$, so $\log X < 0$, so $-\log X > 0$ always.

This is an example of a function (or transformation) of a random variable. These transformations play a major role in probability and statistics. We will see how to find their pdf's on a

few examples.

3.6.0.2 Example (3.6.2) Say X is the number of roles of a fair die until the first six. We have already seen that $P(X = x) = 1/6 * (5/6)^{x-1}$, x=1,2,.. Let Y be 1 if X is even, 0 otherwise. Find the density of Y.

Note: here both X and Y are discrete.

let's do this a little more general, with p instead of 1/6. Also let q=1-p=5/6. Then

$$P(Y = 0) = P(X \in \{1, 3, 5, ..\}) =$$
$$\sum_{k=0}^{\infty} pq^{(2k+1)-1} = p\sum_{k=0}^{\infty} q^{2k} =$$
$$p\sum_{k=0}^{\infty}(q^2)^k = p\frac{1}{1-q^2} =$$
$$p\frac{1}{(1+q)(1-q)} = \frac{1}{1+q}$$

so $P(Y = 0) = 1/(1 + 5/6) = 6/11$ and $P(Y = 1) = 1 - P(Y = 1) = 5/11$.

3.6.0.3 Example (3.6.3) Say we have a fair coin. We flip the coin until the first "Heads". What is the probability this will happen on an even-numbered flip?

Now we have the same as above, with p=0.5, so

P(Y=0)=0.5/(1+0.5)=1/3.

Is there a loaded coin with probability of heads p so that the probability of "first heads on even-numbered flip" is 1/2?

Now P(Y=1)=q/(1+q)=1/2, so 2q=1+q or q=1 or p=0, but if p=0 we never get "heads", so no such coin exists!

3.6.0.4 Example (3.6.4) say X is a continuous r.v with pdf $f_X(x) = 1/2\exp(-|x|)$. This is called a *double exponential* random variable. Let $Y = I_{[-1,1]}(X)$. Find the density of Y.

Note: here X is continuous and Y is discrete.

$$P(Y = 1) = P(I_{[-1,1]}(X) = 1) = P(-1 < X < 1) =$$
$$\int_{-1}^{1} 1/2\exp(-|x|)dx =$$
$$\int_{0}^{1} \exp(-x)dx =$$
$$-e^{-x}|_0^1 = 1 - 1/e$$
$$P(Y = 0) = 1 - P(Y = 1) = 1/e$$

3.6.0.5 Example (3.6.5) again let X have pdf $f_X(x) = 1/2\exp(-|x|)$. Let $Y = X^2$. Then for $y < 0$ we have $P(Y \leq y) = 0$. So let $y > 0$. Then

$$F_Y(y) = P(Y < y) = P(X^2 < y) =$$
$$P(-\sqrt{y} < X < \sqrt{y})$$
$$\int_{-\sqrt{y}}^{\sqrt{y}} 1/2\exp(-|x|)dx =$$
$$\int_0^{\sqrt{y}} \exp(-x)dx =$$
$$-e^{-x}|_0^{\sqrt{y}} = 1 - e^{-\sqrt{y}}$$
$$f_Y(y) = F_Y'(y) = \frac{1}{2\sqrt{y}}e^{-\sqrt{y}}; y > 0$$

Next up some examples of functions of random vectors:

3.6.0.6 Example (3.6.6) say (X,Y) is a bivariate standard normal r.v, that is it has joint given by

$$f(x,y) = \frac{1}{2\pi}\exp\left\{-\frac{1}{2}(x^2+y^2)\right\}$$

Let the r.v. (U,V) be defined by U = X+Y and V = X-Y. Find the joint pdf of (U,V)

To start let's define the functions $g_1(x,y) = x + y$ and $g_2(x,y) = x - y$, so that $U = g_1(X,Y)$ and $V = g_2(X,Y)$.

For what values of u and v is $f_{(U,V)}(u,v)$ positive? Well, for any values for which the system of 2 linear equations in two unknowns u=x+y and u=x-y has a solution.

These solutions are

$$x = h_1(u,v) = (u+v)/2$$
$$y = h_2(u,v) = (u-v)/2$$

From this we find that for any (u,v) there is a unique (x,y) such that u=x+y and v=x-y. So the transformation $(x,y) \to .png)(u,v)$ is one-to-one and therefore has a Jacobian given by

$$\boldsymbol{J} = \begin{vmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \end{vmatrix} = \begin{vmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{vmatrix} = -\frac{1}{2}$$

Now from multivariable calculus we have the following:

$$f_{U,V}(u,v) = f_{X,Y}(h_1(x,y), h_2(x,y))|J| =$$
$$\frac{1}{2\pi}\exp\left\{-\frac{1}{2}\left[(\frac{u+v}{2})^2 + (\frac{u-v}{2})^2\right]\right\}|-\frac{1}{2}| =$$
$$\frac{1}{4\pi}\exp\left\{-\frac{1}{4}\left[u^2+v^2\right]\right\}$$

Note that the density factors into a function of u and a function of v. This is not only a necessary but also a sufficient condition for U and V to be independent.

3.6.0.7 Example (3.6.7) say X and Y are independent standard normal r.v.'s. Let Z = X + Y. Find the pdf of Z.

Note: Z = X + Y = U in the example above, so the pdf of Z is just the marginal of U and we find

$$f_Z(z) = \int_{-\infty}^{\infty}\frac{1}{4\pi}\exp\left\{-\frac{1}{4}\left[z^2+v^2\right]\right\}dv =$$
$$\frac{1}{\sqrt{2\pi}2}e^{-\frac{z^2}{2\times 2}}\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}2}e^{-\frac{v^2}{2\times 2}}dv =$$
$$\frac{1}{\sqrt{2\pi}2}e^{-\frac{z^2}{2\times 2}}$$

3.6.0.8 Theorem (3.6.8) convolution formula

Say X and Y are two continuous independent r.v with pdf f's f_X and f_Y, and let Z = X+Y.

$$f_Z(z) = \int_{-\infty}^{\infty} f_X(t)f_Y(z-t)dt$$

proof

Repeat the above calculations

There is a second method for deriving the convolution formula which is useful. It uses a continuous analog to the law of total probability:

In the setup from above we have

$$F_{X+Y}(z) = P(X+Y \leq z) =$$
$$\int_{-\infty}^{\infty} P(X+Y \leq z | Y = y) f_Y(y) dy =$$
$$\int_{-\infty}^{\infty} (X \leq z - y | Y = y) f_Y(y) dy =$$
$$\int_{-\infty}^{\infty} F_{X|Y=y}(z-y|y) f_Y(y) dy$$
$$f_Z(z) = \frac{d}{dz} F_Z(z) =$$
$$\frac{d}{dz} \int_{-\infty}^{\infty} F_{X|Y=y}(z-y|y) f_Y(y) dy =$$
$$\int_{-\infty}^{\infty} \frac{d}{dz} F_{X|Y=y}(z-y|y) f_Y(y) dy =$$
$$\int_{-\infty}^{\infty} f_{X|Y=y}(z-y|y) f_Y(y) dy =$$
$$\int_{-\infty}^{\infty} f_X(z-y) f_Y(y) dy$$

and here we used the independence only at the very end, the formula above also holds in general.

The tricky part of this is the interchange of the derivative and the integral. Working with densities and cdfs usually means they are ok.

3.6.0.9 Example (3.6.9) Say $X_1, .., X_n$ are iid U[0,1]. Let $M = \max\{X_1, .., X_n\}$. Find f_M.

$$F_M(x) = P(M < m) = P(\max\{X_1, .., X_n\} < x) =$$
$$P(X_1 < x, .., X_n < x) =$$
$$P(X_1 < x) \times .. \times P(X_n < x) =$$
$$P(X_1 < x)^n = x^n$$
$$f_M(x) = nx^{n-1}; 0 < x < 1$$

3.7 Statistics

3.7.1 Basic Concepts of Statistics

Statistics is of course a whole branch of Science all by itself. Here we will just introduce a few basic ideas.

Probability theory and Statistics have in common that both start with a probability model. Typically such models have *parameters*, for example the success probability p in a Bernoulli rv or the rate λ in an exponential distribution. In probability theory we then have problems such as: if p=0.2, what is the mean of the Bernoulli rv? In other words we assume we know the parameters and then ask questions about possible outcomes.

In Statistics it is exactly the other way around: we already have observed outcomes from a rv and we are asking what the parameters might be.

3.7.2 The Likelihood Function

3.7.2.1 Definition (3.7.1) Let $\boldsymbol{X} = (X_1, .., X_n)$ be a random vector with joint density $f(x_1, .., x_n|\boldsymbol{\theta})$ where $\boldsymbol{\theta}$ is a vector of parameters. Then

$$L(\boldsymbol{\theta}|x_1, .., x_n) = f(x_1, .., x_n|\boldsymbol{\theta})$$

is called the *likelihood function*.

If $X_1, .., X_n$ are independent, then

$$L(\boldsymbol{\theta}|\boldsymbol{x}) = \prod_{i=1}^{n} f(x_i|\boldsymbol{\theta})$$

3.7.2.2 Example (3.7.2) $X_i \sim N(\mu, \sigma^2)$, then

$$L(\mu, \sigma^2|\boldsymbol{x}) = \prod_{i=1}^{n} f(x_i|\mu, \sigma^2) =$$
$$\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(x_i - \mu)^2\right\} =$$
$$(2\pi\sigma^2)^{-n/2} \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2\right\}$$

3.7.3 Point Estimation

3.7.3.1 Definition (3.7.3) A function $T(\boldsymbol{x})$ of the data is called a *statistic* or *estimator*.

A statistic $T(\boldsymbol{x})$ is called an *unbiased* estimator of $\boldsymbol{\theta}$ if

$$E[T(\boldsymbol{X})] = \boldsymbol{\theta}$$

3.7.3.2 Example (3.7.4) $X_i \sim N(\mu, \sigma^2)$, then $T(\boldsymbol{x}) = \bar{x}$ is an unbiased estimator of μ because

$$E[T(\boldsymbol{X})] = E[\bar{X}] = E[\frac{1}{n}\sum_{i=1}^{n} X_i] = \frac{1}{n}\sum_{i=1}^{n} E[X_i] = \mu$$

3.7.4 Maximum Likelihood

3.7.4.1 Definition (3.7.5) Let $\hat{\boldsymbol{\theta}} = argmax\,\{L(\boldsymbol{\theta}|\boldsymbol{x});\boldsymbol{\theta}\}$, then $\hat{\boldsymbol{\theta}}$ is called the *maximim likelihood estimator of $\boldsymbol{\theta}$.

3.7.4.2 Example (3.7.6)

$$L(\mu,\sigma^2|\boldsymbol{x}) = (2\pi\sigma^2)^{-n/2}\exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2\right\}$$
$$\log L(\mu,\sigma^2|\boldsymbol{x}) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2$$
$$\frac{\log L(\mu,\sigma^2|\boldsymbol{x})}{d\mu} = \frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i-\mu) = 0$$
$$\hat{\mu} = \bar{x}$$

3.7.5 Confidence Interval

3.7.5.1 Definition (3.7.7) A (random) interval of the form $(L(\boldsymbol{X}), U(\boldsymbol{X}))$ is called a $(1-\alpha)100\%$ confidence interval for θ if

$$P(L(\boldsymbol{X}) < \theta < U(\boldsymbol{X})) \geq 1-\alpha$$

3.7.5.2 Example (3.7.8) $X_i \sim N(\mu,\sigma^2)$, σ^2 known, then

$$L(\boldsymbol{x}) = \bar{x} - z_{\alpha/2}\sigma/\sqrt{n}$$
$$U(\boldsymbol{x}) = \bar{x} + z_{\alpha/2}\sigma/\sqrt{n}$$

is a $(1-\alpha)100\%$ confidence interval for μ. Here z_α is is upper α percentile of a standard normal distribution.

$$P(L(\boldsymbol{X}) < \theta < U(\boldsymbol{X})) =$$
$$P(\bar{X} - z_{\alpha/2}\sigma/\sqrt{n} < \mu < \bar{X} + z_{\alpha/2}\sigma/\sqrt{n}) =$$
$$P(\mu - z_{\alpha/2}\sigma/\sqrt{n} < \bar{X} < \mu + z_{\alpha/2}\sigma/\sqrt{n}) =$$
$$P(-z_{\alpha/2} < \sqrt{n}\frac{\bar{X}-\mu}{\sigma} < z_{\alpha/2}) =$$
$$2\Phi(z_{\alpha/2}) - 1 = 2(1-\alpha/2) - 1 = 1-\alpha$$

where Φ is the cdf of a standard normal distribution.

3.7.6 Hypothesis Test

3.7.6.1 Definition (3.7.9) In testing $H_0 : \theta = \theta_0$

- the error of the first kind is to reject the null hypothesis although it is true. The probability of the error of the first kind is denoted by α.
- the error of the second kind is to fail to reject the null hypothesis although it is false. The probability of the error of the second kind is denoted by β. $1-\beta$ is called the power of the test.
- The p-value of a test is the probability to observe a value of the test statistic as unlike as that just observed or even more so, given that the null hypothesis is true.

3.7.6.2 Example (3.7.10) $X_i \sim N(\mu, \sigma^2)$, σ^2 known, then a test for $H_0 : \mu = \mu_0$ is to reject the null hypothesis if $Z = |\sqrt{n}\frac{\bar{x}-\mu_0}{\sigma}|$ is greater than $z_{\alpha/2}$.

So

$$
\begin{aligned}
&P(\text{reject } H_0 | H_0 \text{ true}) = \\
&P(Z > z_{\alpha/2} | \mu = \mu_0) = \\
&1 - P(|\sqrt{n}\frac{\bar{x}-\mu_0}{\sigma}| < z_{\alpha/2} | \mu = \mu_0) = \\
&1 - P(-z_{\alpha/2} < \sqrt{n}\frac{\bar{x}-\mu_0}{\sigma} < z_{\alpha/2} | \mu = \mu_0) = \\
&1 - (1-\alpha) = \alpha
\end{aligned}
$$

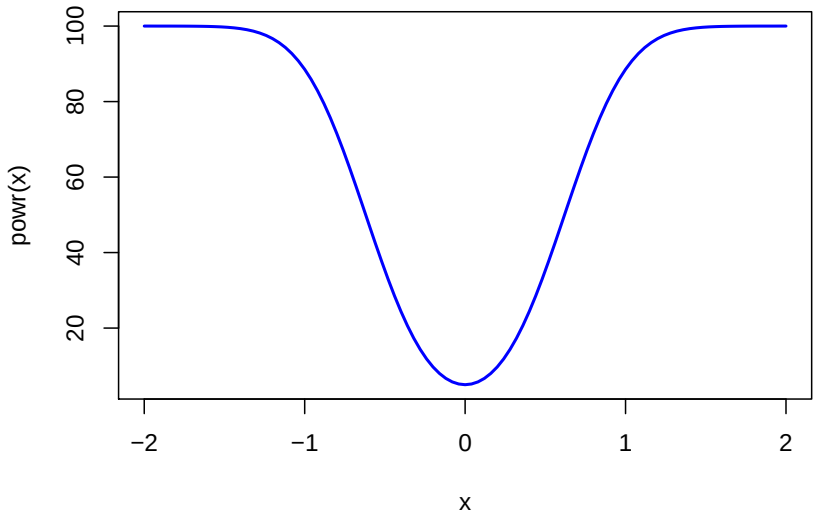
and

$$
\begin{aligned}
&\beta = P(\text{fail to reject } H_0 | H_0 \text{ false}) = \\
&P(Z < z_{\alpha/2} | \mu = \mu_1) = \\
&P(|\sqrt{n}\frac{\bar{x}-\mu_0}{\sigma}| < z_{\alpha/2} | \mu = \mu_1) = \\
&P(-z_{\alpha/2} < \sqrt{n}\frac{\bar{x}-\mu_0}{\sigma} < z_{\alpha/2} | \mu = \mu_1) = \\
&P(-z_{\alpha/2} < \sqrt{n}\frac{\bar{x}-\mu_1+\mu_1-\mu_0}{\sigma} < z_{\alpha/2} | \mu = \mu_1) = \\
&P(-z_{\alpha/2} < \sqrt{n}\frac{\bar{x}-\mu_1}{\sigma} + \sqrt{n}\frac{\mu_1-\mu_0}{\sigma} < z_{\alpha/2} | \mu = \mu_1) = \\
&P(-z_{\alpha/2} - \sqrt{n}\frac{\mu_1-\mu_0}{\sigma} < \sqrt{n}\frac{\bar{x}-\mu_1}{\sigma} < z_{\alpha/2} + \sqrt{n}\frac{\mu_1-\mu_0}{\sigma} | \mu = \mu_1) = \\
&\Phi(z_{\alpha/2} + \sqrt{n}\frac{\mu_1-\mu_0}{\sigma}) - \Phi(-z_{\alpha/2} + \sqrt{n}\frac{\mu_1-\mu_0}{\sigma})
\end{aligned}
$$

and so we see that β (and therefore the power of a test) is a function of the true value of the parameter. It is often shown in the form of a power graph:

3.7.6.3 Example (3.7.11) Say

```
powr=function(mu1, n=10, mu0=0, sigma=1, alpha=0.05) {
   za=qnorm(1-alpha/2)
   beta=pnorm(za-sqrt(n)*(mu1-mu0)/sigma)-pnorm(-za-sqrt(n)*(mu1-mu0)/sigma)
   100*(1-beta)
}
curve(powr, -2, 2, lwd=2,col="blue")
```



3.7.7 Sample Size and Effect Size

An important question during the planning stage of an experiment is how many observations need to be collected. First of all this depends on what analyses is going to be done.

- **Confidence Interval**

If eventually a confidence interval is to be found, one needs to decide what length of the interval is acceptable. More specifically one chooses the *error*, or 1/2 the length. This choice depends on the background of the experiment, and what size interval is small enough to yield an interesting result.

3.7.7.1 Example (3.7.12) $X_i \sim N(\mu, \sigma^2)$, σ^2 known, then we have the interval

$$\left(\bar{x} - z_{\alpha/2}\sigma/\sqrt{n}, \bar{x} + z_{\alpha/2}\sigma/\sqrt{n}\right)$$

so $E = z_{\alpha/2}\sigma/\sqrt{n}$ and so $n = \left(\frac{z_{\alpha/2}\sigma}{E}\right)^2$

Say we want E=0.1 and we know $\sigma = 1$, then

```
round((qnorm(0.975)*1/0.1)^2)
```

```
## [1] 384
```

- **Hypothesis Test**

If we plan on carrying out a hypothesis we need to decide what power the test is supposed to have, one typical choice is at least 80%. From the calculation above it is clear that we also need μ_1. $|\mu_0 - \mu_1|$ is often called the *effect size*, the smallest deviation from the null hypothesis of practical interest.

3.7.7.2 Example (3.7.13)

$X_i \sim N(\mu, \sigma^2)$, σ^2 known, and say we decide to use $\mu_1 = 0.1$. We have the equation

$$\beta = \Phi(z_{\alpha/2} + \sqrt{n}\frac{\mu_1 - \mu_0}{\sigma}) - \Phi(-z_{\alpha/2} + \sqrt{n}\frac{\mu_1 - \mu_0}{\sigma})$$

which we want to solve for n. This has to be done numerically:

```
n=1
repeat {
   n=n+1
   if(powr(mu1=0.1, n=n)>80) break
}
n
```

```
## [1] 785
```

3.7.8 Bayesian Analysis

An entirely different approach to statistics is the Bayesian. Here one treats parameters as random variables with a distribution called a *prior*, and then uses probability theory to combine the likelihood and the prior intothe *posterior* distribution.

3.7.8.1 Example (3.7.14)

$X_i \sim N(\mu, \sigma^2)$, σ^2 known, and we use $\pi(\mu) = 1$ as the prior. Note that this is not a proper density because $\int_{-\infty}^{\infty} 1dx = \infty$. This is ok as long as the posterior is a proper distribution.

First note that

$$\sum_{i=1}^{n}(x_i-\mu)^2 =$$
$$\sum_{i=1}^{n}(x_i-\bar{x}+\bar{x}-\mu)^2 =$$
$$\sum_{i=1}^{n}(x_i-\bar{x})^2+2(x_i-\bar{x})(\bar{x}-\mu)+(\bar{x}-\mu)^2 =$$
$$\sum_{i=1}^{n}(x_i-\bar{x})^2+2(\bar{x}-\mu)\sum_{i=1}^{n}(x_i-\bar{x})(\bar{x}-\mu)+n(\bar{x}-\mu)^2$$
$$\sum_{i=1}^{n}(x_i-\bar{x})^2+2(x_i-\bar{x})(\bar{x}-\mu)+(\bar{x}-\mu)^2 =$$
$$\sum_{i=1}^{n}(x_i-\bar{x})^2+n(\bar{x}-\mu)^2$$

and so

$$f(\boldsymbol{x},\mu)=(2\pi\sigma^2)^{-n/2}\exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2\right\}\pi(\mu) =$$
$$(2\pi\sigma^2)^{-n/2}\exp\left\{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^{n}(x_i-\bar{x})^2+n(\bar{x}-\mu)^2\right]\right\} =$$
$$(2\pi\sigma^2)^{-n/2}\exp\left\{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^{n}(x_i-\bar{x})^2\right]\right\}\exp\left\{-\frac{1}{2\sigma^2}\left[n(\bar{x}-\mu)^2\right]\right\}$$
$$m(\boldsymbol{x})=\int_{-\infty}^{\infty}f(\boldsymbol{x},\mu)d\mu =$$
$$\int_{-\infty}^{\infty}(2\pi\sigma^2)^{-n/2}\exp\left\{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^{n}(x_i-\bar{x})^2\right]\right\}\exp\left\{-\frac{1}{2\sigma^2}\left[n(\bar{x}-\mu)^2\right]\right\}d\mu =$$
$$\exp\left\{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^{n}(x_i-\bar{x})^2\right]\right\}n^{n/2}(2\pi(\sigma^2/n)^{-(n-1)/2}$$
$$\int_{-\infty}^{\infty}(2\pi(\sigma^2/n)^{-1/2}\exp\left\{-\frac{1}{2(\sigma^2/n)}(\mu-\bar{x})^2\right\}d\mu =$$
$$\exp\left\{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^{n}(x_i-\bar{x})^2\right]\right\}n^{n/2}(2\pi(\sigma^2/n)^{-(n-1)/2}$$

because the integral is over a $N(\bar{x},\sigma^2/n)$ density and therefore is 1.

Finally

$$f(\mu|\boldsymbol{x}) = \frac{f(\boldsymbol{x},\mu)}{m(\boldsymbol{x})} =$$
$$\frac{\exp\left\{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^n (x_i-\bar{x})^2\right]\right\} n^{n/2}(2\pi(\sigma^2/n)^{-n/2}\exp\left\{-\frac{1}{2(\sigma^2/n)}(\mu-\bar{x})^2\right\}}{\exp\left\{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^n (x_i-\bar{x})^2\right]\right\} n^{n/2}(2\pi(\sigma^2/n)^{-(n-1)/2}} =$$
$$\frac{1}{\sqrt{2\pi(\sigma^2/n)}}\exp\left\{-\frac{1}{2(\sigma^2/n)}(\mu-\bar{x})^2\right\}$$

and so we find

$$\mu|\boldsymbol{X}=\boldsymbol{x} \sim N(\mu, \sigma^2/n)$$

Say we want to find a $(1-\alpha)100\%$ *credible interval* for μ, that is numbers L and U such that

$$P(L<\mu<U|\boldsymbol{X}=\boldsymbol{x}) = 1-\alpha$$

Note that here L and U are numbers, not random variables as in the frequentist calculation above. Also note that we have one equation in two unknowns, so L and U are not uniquely defined. We can get a solution by imposing an additional condition. For example, we can find

$$P(\mu<L|\boldsymbol{X}=\boldsymbol{x}) = P(\mu>U|\boldsymbol{X}=\boldsymbol{x}) = \alpha/2$$

so

$$\alpha/2 = P(\mu<L|\boldsymbol{X}=\boldsymbol{x}) = \Phi(L;\mu,\sigma^2/n)$$
$$L = \Phi^{-1}(\alpha/2;\mu,\sigma^2/n)$$

where $\Phi(x;\mu,\sigma^2)$ is the cdf of a $N(\mu,\sigma^2)$ and Φ^{-1} is its inverse.

Notice that in this case the frequentist confidence interval and the Bayesian credible interval are the same numerically, but their interpretation is very different.

Linear Algebra

3.8 Matrix and Vector Notation

3.8.1 Matrices

A *matrix* is a rectangular or square array of numbers or variables. We use uppercase boldface letters to represent matrices. All elements of matrices will be real numbers or variables representing real numbers. Here is an example of a 3×2 matrix

$$\boldsymbol{A} = (a_{ij}) = \begin{pmatrix} 0.5 & 12 \\ 0.8 & 9 \\ -0.1 & 14 \end{pmatrix}$$

A *vector* is an $n \times 1$ matrix:

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

A matrix of dimension 1×1 is called a *scalar*.

Two matrices A and B are equal if they have the same dimension and if $a_{ij} = b_{ij}$ for all i and j.

The *transpose* A' of a matrix is the matrix with rows and columns exchanged.

$$\boldsymbol{A}' = (a_{ji}) = \begin{pmatrix} 0.5 & 0.8 & -0.1 \\ 12 & 9 & 14 \end{pmatrix}$$

3.8.1.1 Theorem (4.1.1) For any matrix $\boldsymbol{A}$ we have $\boldsymbol{A} = (\boldsymbol{A}')'$

proof obvious

3.8.1.2 Definition (4.1.2) An $n \times m$ matrix $\boldsymbol{A}$ is called *square* if n=m.

A matrix $\boldsymbol{A}$ is called *symmetric* if $\boldsymbol{A} = \boldsymbol{A}'$.

The *diagonal* of a matrix $\boldsymbol{A}$ are the elements (a_{ii}). A matrix is called *diagonal* if $(a_{ij}) = 0$ for all $i \neq j$.

A matrix $\boldsymbol{A}$ with $(a_{ii}) = 1$ for all i and $(a_{ij}) = 0$ for all $i \neq j$ is called an *identity matrix*.

A matrix is called *upper triangular* if $(a_{ij}) = 0$ for all $i < j$ and *lower triangular* if $(a_{ij}) = 0$ for all $i > j$.

A vector of 1's is denoted by $\boldsymbol{j}$:

$$\boldsymbol{j} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

A square matrix of 1's is denoted by $\boldsymbol{J}$:

$$\boldsymbol{J} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

A vector and a matrix of 0's are denoted by

$$\boldsymbol{0} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

and

$$\boldsymbol{O} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

3.8.1.3 Definition (4.1.3)

Let $\boldsymbol{A}$ be a $n \times m$ matrix and $\boldsymbol{B}$ be a $m \times k$ matrix, then the product $\boldsymbol{C} = \boldsymbol{AB}$ is defined by

$$c_{ij} = \sum_{l=1}^{m} a_{il} b_{lj}$$

3.8.1.4 Corollary (4.1.4)

Let $\boldsymbol{a} = (a_1, .., a_n)'$ and $\boldsymbol{b} = (b_1, .., b_n)'$ then $\boldsymbol{a'a} = \sum_{i=1}^{n} a_i^2$ and $\boldsymbol{a'b} = \sum_{i=1}^{n} a_i b_i$

3.8.1.5 Example (4.1.5)

$$\boldsymbol{j'j} = \sum_{i=1}^{n} 1 = n$$
$$\boldsymbol{jj'} = \boldsymbol{J}$$

3.8.1.6 Example (4.1.6)

$$\begin{pmatrix} 0.5 & 12 \\ 0.8 & 9 \\ -0.1 & 14 \end{pmatrix} \begin{pmatrix} 1 & 2 & 5 & -2.5 & 7 \\ .6 & 0.8 & 6 & 0 & -2.7 \end{pmatrix} = \begin{pmatrix} 7.7 & 10.6 & 74.5 & -1.25 & -28.9 \\ 6.2 & 8.8 & 58.0 & -2.00 & -18.7 \\ 8.3 & 11.0 & 83.5 & 0.25 & -38.5 \end{pmatrix}$$

Matrix multiplication in R is done like this:

```
A=matrix(c(0.5, 0.8,-0.1, 12,9,14), 3,2)
B=matrix(c(1, 0.6, 2, 0.8, 5, 6, -2.5, 0, 7, -2.7), 2,5)
A%*%B
```

```
##      [,1] [,2] [,3]  [,4]  [,5]
## [1,]  7.7 10.6 74.5 -1.25 -28.9
## [2,]  6.2  8.8 58.0 -2.00 -18.7
## [3,]  8.3 11.0 83.5  0.25 -38.5
```

3.8.1.7 Theorem (4.1.7)

In general $\boldsymbol{AB} \neq \boldsymbol{BA}$.

proof

$\boldsymbol{AB}$ and $\boldsymbol{BA}$ can only exist if both matrices are square. Now say

$$\begin{pmatrix} 5 & 12 \\ 8 & 9 \end{pmatrix} \begin{pmatrix} 0 & 2 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 12 & 10 \\ 9 & 16 \end{pmatrix}$$

but

$$\begin{pmatrix} 0 & 2 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 5 & 12 \\ 8 & 9 \end{pmatrix} = \begin{pmatrix} 16 & 18 \\ 5 & 12 \end{pmatrix}$$

3.8.1.8 Definition Let $\boldsymbol{x}$ be a vector, then the *Euclidean distance* or *length* of the vector is defined by

$$\sqrt{\boldsymbol{x'x}} = \sqrt{\sum_{i=1}^{n} x_i^2}$$

Let $\boldsymbol{A}$ be an $n \times m$ matrix and $\boldsymbol{j}$ a vector of 1's, then

$$\boldsymbol{Aj} = \begin{pmatrix} \sum_{i=1}^{n} a_{1i} \\ \sum_{i=1}^{n} a_{2i} \\ \cdots \\ \sum_{i=1}^{n} a_{mi} \end{pmatrix}$$

3.8.1.9 Theorem (4.1.8) $\boldsymbol{(AB)' = A'B'}$

proof

Let $\boldsymbol{C} = \boldsymbol{AB}$. Now

$$(\boldsymbol{AB})'_{ij} = (\boldsymbol{C'})_{ij} = c_{ji} =$$
$$\sum_{k=1}^{n} a_{jk} b_{ki} =$$
$$\sum_{k=1}^{n} b_{ki} a_{jk} =$$
$$\sum_{k=1}^{n} (\boldsymbol{B})_{ki} (\boldsymbol{A})_{jk} =$$
$$\sum_{k=1}^{n} (\boldsymbol{B'})_{ik} (\boldsymbol{A'})_{kj} =$$
$$\boldsymbol{B'A'}$$

3.8.1.10 Definition (4.1.9) Partitioned Matrix

We can partition a matrix as follows

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A_{11}} & \boldsymbol{A_{12}} \\ \boldsymbol{A_{21}} & \boldsymbol{A_{22}} \end{pmatrix}$$

3.8.1.11 Example (4.1.10)

$$\boldsymbol{A} = \begin{pmatrix} 1 & 3 & 1 & 0 \\ 0 & 4 & 5 & 1 \\ 2 & 1 & 3 & 7 \\ 1 & 1 & 5 & 3 \end{pmatrix} = \begin{pmatrix} \boldsymbol{A_{11}} & \boldsymbol{A_{12}} \\ \boldsymbol{A_{21}} & \boldsymbol{A_{22}} \end{pmatrix}$$

where

$$\boldsymbol{A_{11}} = \begin{pmatrix} 1 & 3 \\ 0 & 4 \end{pmatrix} \boldsymbol{A_{12}} = \begin{pmatrix} 1 & 0 \\ 5 & 1 \end{pmatrix} \boldsymbol{A_{21}} = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix} \boldsymbol{A_{22}} = \begin{pmatrix} 3 & 7 \\ 5 & 3 \end{pmatrix}$$

or where

$$\boldsymbol{A_{11}} = \begin{pmatrix} 1 \end{pmatrix} \boldsymbol{A_{12}} = \begin{pmatrix} 3 & 1 & 0 \end{pmatrix} \boldsymbol{A_{21}} = \begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix} \boldsymbol{A_{22}} = \begin{pmatrix} 4 & 5 & 1 \\ 1 & 3 & 7 \\ 1 & 5 & 3 \end{pmatrix}$$

3.8.1.12 Definition (4.1.11) If $\boldsymbol{A}$ is a symmetric matrix and $\boldsymbol{x}, \boldsymbol{y}$ are vectors, then

i.
$$\boldsymbol{Ay} = \sum_{i=1}^{n} a_{ij} y_j$$
is called a *linear form.*

ii.
$$\boldsymbol{y'Ay} = \sum_{i=1}^{n} a_{ii} y_i^2 + \sum_{i \neq j} a_{ij} y_i y_j$$
is called the *quadratic form.*

iii.
$$\boldsymbol{x'Ay} = \sum_{i,j} a_{ij} x_i y_j$$
is called a *bilinear form.*

3.8.1.13 Definition (4.1.12) A set of vectors $\boldsymbol{a}_1, .., \boldsymbol{a}_n$ is called *linearly dependent* if there exist scalars $c_1, .., c_n$ (not all 0) such that

$$c_1 \boldsymbol{a}_1 + .. + c_n \boldsymbol{a}_n = 0$$

If no such coefficients $c_1, .., c_n$ can be found the vectors are called *linearly independent.*

The *rank* of a square matrix $\boldsymbol{A}$ is the number of linearly independent columns of $\boldsymbol{A}$.

An $n \times p$ matrix $\boldsymbol{A}$ with n<p is said to be *full rank* if rank($\boldsymbol{A}$)=n. A full-rank square matrix is called *nonsingular*.

3.8.1.14 Theorem (4.1.13)

i. $\text{rank}(\boldsymbol{AB}) \leq \min\{\text{rank}(\boldsymbol{A}); \text{rank}(\boldsymbol{B})\}$

ii. $\text{rank}(\boldsymbol{AA'}) = \text{rank}(\boldsymbol{A'A}) = \text{rank}(\boldsymbol{A})$

proof omitted

3.9 Matrix Operations

3.9.1 Matrix Inverse

3.9.1.1 Definition (4.2.1)

A nonsingular matrix $\boldsymbol{A}$ has a unique *inverse* $\boldsymbol{A}^{-1}$ such that

$$\boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{I}$$

3.9.1.2 Example (4.2.2)

We can use the R function *solve* to find an inverse:

$$\boldsymbol{A} = \begin{pmatrix} 1 & 5 & -3 \\ -3 & 2 & 7 \\ 2 & 5 & 9 \end{pmatrix}$$

```
A=rbind(c(1, 5, -3), c(-3, 2, 7), c(2, 5, 9))
A
```

```
##      [,1] [,2] [,3]
## [1,]    1    5   -3
## [2,]   -3    2    7
## [3,]    2    5    9
```

```
Ainv=solve(A)
Ainv
```

```
##             [,1]        [,2]        [,3]
## [1,] -0.06938776 -0.24489796 0.167346939
## [2,]  0.16734694  0.06122449 0.008163265
## [3,] -0.07755102  0.02040816 0.069387755
```

```
round(A%*%Ainv, 4)
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    1    0
## [3,]    0    0    1
```

3.9.1.3 Theorem (4.2.3) Let

$$\boldsymbol{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

then

$$\boldsymbol{A}^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

proof easy, just multiply!

This is actually more general than it appears:

3.9.1.4 Theorem (4.2.4) Say $\boldsymbol{A}$ is a symmetric and nonsingular matrix partitioned as

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A_{11}} & \boldsymbol{A_{12}} \\ \boldsymbol{A_{21}} & \boldsymbol{A_{22}} \end{pmatrix}$$

then if $\boldsymbol{B} = \boldsymbol{A_{22}} - \boldsymbol{A_{21}}\boldsymbol{A_{11}^{-1}}\boldsymbol{A_{12}}$ and provided all inverses exist we have

$$\boldsymbol{A^{-1}} = \begin{pmatrix} \boldsymbol{A_{11}^{-1}} + \boldsymbol{A_{11}^{-1}}\boldsymbol{A_{12}}\boldsymbol{B^{-1}}\boldsymbol{A_{21}}\boldsymbol{A_{11}^{-1}} & -\boldsymbol{A_{11}^{-1}}\boldsymbol{A_{12}}\boldsymbol{B^{-1}} \\ -\boldsymbol{B^{-1}}\boldsymbol{A_{21}}\boldsymbol{A_{11}^{-1}} & \boldsymbol{B^{-1}} \end{pmatrix}$$

proof straight-forward multiplication

3.9.1.5 Example (4.2.5)

1. Say $\boldsymbol{A}$ is a square matrix partitioned as

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A_{11}} & \boldsymbol{a}_{12} \\ \boldsymbol{a}_{12} & a_{22} \end{pmatrix}$$

where a_{22} is a scalar, then $b = a_{22} - a'_{21}\boldsymbol{A}_{11}^{-1}a_{12}$ is a scalar and

$$\boldsymbol{A^{-1}} = \frac{1}{b} \begin{pmatrix} b\boldsymbol{A_{11}^{-1}} + \boldsymbol{A_{11}^{-1}}a_{12}a'_{12}\boldsymbol{A_{11}^{-1}} & -\boldsymbol{A_{11}^{-1}}a_{12} \\ -a'_{12}\boldsymbol{A_{11}^{-1}} & 1 \end{pmatrix}$$

Say

$$\boldsymbol{A} = \begin{pmatrix} 1 & -2 & 4 \\ -2 & 3 & 0 \\ 4 & 0 & 1 \end{pmatrix}$$

$\boldsymbol{A}_{11}^{-1} = \begin{pmatrix} -3 & -2 \\ -2 & -1 \end{pmatrix}$ $b = a_{22} - a'_{21}\boldsymbol{A}_{11}^{-1}a_{12} = 1 - \begin{pmatrix} 4 & 0 \end{pmatrix}\begin{pmatrix} -3 & -2 \\ -2 & -1 \end{pmatrix}\begin{pmatrix} 4 \\ 0 \end{pmatrix} = 1 - (-48) = 49$ $b\boldsymbol{A_{11}^{-1}} + \boldsymbol{A_{11}^{-1}}a_{12}a'_{12}\boldsymbol{A_{11}^{-1}}$ =

also

$$-\boldsymbol{A}_{11}^{-1}a_{12} = \begin{pmatrix}12\\8\end{pmatrix} - a_{12}'\boldsymbol{A}_{11}^{-1} = \begin{pmatrix}12 & 8\end{pmatrix}$$

and finally

$$\boldsymbol{A}^{-1} = \frac{1}{49}\begin{pmatrix}-3 & -2 & 12\\-2 & 15 & 8\\12 & 8 & 1\end{pmatrix} =$$

R check:

```
solve(matrix(c(1,-2,4,-2,3,0,4,0,1), 3,  3))*49
```

```
##      [,1] [,2] [,3]
## [1,]   -3   -2   12
## [2,]   -2   15    8
## [3,]   12    8    1
```

2. Say $\boldsymbol{A}$ is a square matrix partitioned as

$$\boldsymbol{A} = \begin{pmatrix}\boldsymbol{A_{11}} & \boldsymbol{O}\\\boldsymbol{O} & \boldsymbol{A_{22}}\end{pmatrix}$$

then

$$\boldsymbol{A^{-1}} = \begin{pmatrix}\boldsymbol{A_{11}^{-1}} & \boldsymbol{O}\\\boldsymbol{O} & \boldsymbol{A_{22}^{-1}}\end{pmatrix}$$

3.9.2 Positive Definite Matrices

3.9.2.1 Definition (4.2.6) Let $\boldsymbol{A}$ be a symmetric matrix and $\boldsymbol{y}$ a vector, then if

i. $\boldsymbol{y'Ay} > 0$ for all $\boldsymbol{y} \neq 0$ A is called *positive definite*.

ii. $\boldsymbol{y'Ay} \geq 0$ for all $\boldsymbol{y} \neq 0$ A is called *positive semi-definite*.

3.9.2.2 Example (4.2.7)

$$\boldsymbol{A} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

$$\boldsymbol{y'Ay} = (y_1 \ y_2) \begin{pmatrix} y_1 - y_2 \\ -y_1 + y_2 \end{pmatrix} =$$

$$\begin{aligned} &y_1(y_1 - y_2) + y_2(-y_1 + y_2) = \\ &y_1^2 - y_1y_2 - y_1y_2 + y_2^2 = \\ &y_1^2 - 2y_1y_2 + y_2^2 = \\ &(y_1 - y_2)^2 > 0 \end{aligned}$$

and so $\boldsymbol{A}$ is positive definite.

3.9.2.3 Theorem (4.2.8)

i. If $\boldsymbol{A}$ is positive definite, then $a_{ii} > 0$ for all i

ii. If $\boldsymbol{A}$ is positive semi-definite, then $a_{ii} \geq 0$ for all i

proof

Let $\boldsymbol{y'} = (0, .., 1, .., 0)$, then $\boldsymbol{y'Ay} = a_{ii} > 0$

3.9.2.4 Theorem (4.2.9)

Let $\boldsymbol{P}$ be a nonsingular matrix, then if $\boldsymbol{A}$ is positive (semi)-definite, so is $\boldsymbol{P'AP}$

proof

$$\boldsymbol{y'P'APy = (Py')A(Py)}$$

and if P is nonsingular $\boldsymbol{Py} = 0$ iff $\boldsymbol{y} = 0$.

3.9.2.5 Theorem (4.2.10)

If $\boldsymbol{A}$ is positive-definite, then $\boldsymbol{A^{-1}}$ is positive-definite.

proof omitted

3.9.2.6 Theorem (4.2.11)

Let $\boldsymbol{A}$ be an $n \times p$ matrix

i. if rank($\boldsymbol{A}$)=p, then $\boldsymbol{A'A}$ is positive definite

ii. if rank($\boldsymbol{A}$)<p, then $\boldsymbol{A'A}$ is positive semi-definite

proof omitted

3.9.3 Systems of Equations

3.9.3.1 Example (4.2.12) We want to solve the system of linear equations

$$\begin{aligned} 2y_1 + 3y_2 - y_3 &= 1 \\ y_1 + 2y_2 + 2y_3 &= 2 \\ 3y_1 + 3y_2 + y_3 &= 3 \end{aligned}$$

this can be done by solving the matrix equation $\boldsymbol{Ay} = \boldsymbol{c}$ where

$$\boldsymbol{A} = \begin{pmatrix} 2 & 3 & -1 \\ 1 & 2 & 2 \\ 3 & 3 & 1 \end{pmatrix}, \boldsymbol{c} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}, \boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

and a solution is given by $\boldsymbol{y} = \boldsymbol{A^{-1}c}$. So

```
A=rbind(c(2, 3, 1), c(1, 2, 2), c(3, 3, 1) )
cc=cbind(c(1, 2, 3))
Ainf=solve(A)
Ainf
```

```
##       [,1]          [,2]  [,3]
## [1,] -1.00 -1.850372e-16  1.00
## [2,]  1.25 -2.500000e-01 -0.75
## [3,] -0.75  7.500000e-01  0.25
```

```
Ainf %*% cc
```

```
##      [,1]
## [1,]  2.0
## [2,] -1.5
## [3,]  1.5
```

or directly

```
solve(A, cc)
```

```
##      [,1]
## [1,]  2.0
## [2,] -1.5
## [3,]  1.5
```

3.9.4 Generalized Inverse

3.9.4.1 Definition (4.2.13) A *generalized inverse* of an $n \times p$ matrix $\boldsymbol{A}$ is any matrix $\boldsymbol{A^-}$ such that

$$\boldsymbol{AA^-A} = \boldsymbol{A}$$

Generalized inverses are not unique, except if $\boldsymbol{A}$ is nonsingular and then $\boldsymbol{A^-} = \boldsymbol{A^{-1}}$. Every matrix has a generalized inverse.

3.9.4.2 Example (4.2.14)

$$\boldsymbol{y} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

then $\boldsymbol{y^-} = (1, 0, 0)$ because

$$\boldsymbol{yy^-y} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} (1, 0, 0) \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} 1 = \boldsymbol{y}$$

3.9.4.3 Theorem (4.2.15) If a system of equations $\boldsymbol{Ax} = \boldsymbol{c}$ is consistent (that is has a solution), then all possible solutions can be found as follows: find $\boldsymbol{A^-}$, then all solutions are of the form

$$\boldsymbol{A^-c} + (\boldsymbol{I} - \boldsymbol{A^-A})\boldsymbol{h}$$

for any arbitrary vector $\boldsymbol{h}$.

proof omitted

3.9.4.4 Example (4.2.16) We want to solve the system

$$\begin{aligned} 2y_1 + 3y_2 - y_3 &= 1 \\ y_1 + 2y_2 + 2y_3 &= 2 \end{aligned}$$

to find a generalized inverse we find the inverse of the matrix

$$\begin{pmatrix} 2 & 3 \\ 1 & 2 \end{pmatrix}$$

which is

$$\begin{pmatrix} 2 & -3 \\ -1 & 2 \end{pmatrix}$$

and so a generalized inverse is given by

$$\boldsymbol{A}^- = \begin{pmatrix} 2 & -3 \\ -1 & 2 \\ 0 & 0 \end{pmatrix}$$

Let's check:

```
A=rbind(c(2, 3, -1), c(1, 2, 2))
A
```

```
##      [,1] [,2] [,3]
## [1,]    2    3   -1
## [2,]    1    2    2
```

```
y= cbind(c(2, -1, 0), c(-3, 2, 0))
y
```

```
##      [,1] [,2]
## [1,]    2   -3
## [2,]   -1    2
## [3,]    0    0
```

```
A %*% y %*% A
```

```
##      [,1] [,2] [,3]
## [1,]    2    3   -1
## [2,]    1    2    2
```

Now all the solutions are given by

$$\boldsymbol{A}^-\boldsymbol{c} + (\boldsymbol{I} - \boldsymbol{A}^-\boldsymbol{A})\boldsymbol{h} =$$
$$\begin{pmatrix} 2 & -3 \\ -1 & 2 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} + \left(\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 2 & -3 \\ -1 & 2 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 2 & 3 & -1 \\ 1 & 2 & 2 \end{pmatrix} \right) \begin{pmatrix} h_1 \\ h_2 \\ h_3 \end{pmatrix}$$
$$\begin{pmatrix} -4 \\ 3 \\ 0 \end{pmatrix} + \left(\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 & -8 \\ 0 & 1 & 5 \\ 0 & 0 & 0 \end{pmatrix} \right) \begin{pmatrix} h_1 \\ h_2 \\ h_3 \end{pmatrix} =$$
$$\begin{pmatrix} -4 \\ 3 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 8 \\ 0 & 0 & -5 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} h_1 \\ h_2 \\ h_3 \end{pmatrix} =$$
$$\begin{pmatrix} -4 + 8h_3 \\ 3 - 5h_3 \\ h_3 \end{pmatrix}$$

Here is a solution using R

```
library(MASS)
A
```

```
##      [,1] [,2] [,3]
```

```
## [1,]    2    3   -1
## [2,]    1    2    2
```

```
gA=ginv(A)
gA
```

```
##             [,1]       [,2]
## [1,]  0.1333333 0.02222222
## [2,]  0.1666667 0.11111111
## [3,] -0.2333333 0.37777778
```

```
A%*%gA%*%A
```

```
##      [,1] [,2] [,3]
## [1,]    2    3   -1
## [2,]    1    2    2
```

```
y=gA%*%cbind(c(1, 2))
A%*%y
```

```
##      [,1]
## [1,]    1
## [2,]    2
```

but of course this yields only one solution.

3.9.5 Determinants

The determinant of an $n \times n$ matrix A is a scalar function of A, denoted by either det(A) or $|A|$, defined as the sum of all n! possible products of n elements such that

1. each product contains one element from every row and every column of A.

2. the factors in each product are written so that the column subscripts appear in order of magnitude and each product is then preceded by a plus or minus sign according to whether the number of inversions in the row subscripts is even or odd. (An inversion occurs whenever a larger number precedes a smaller one.)

3.9.5.1 Theorem (4.2.17)

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$$

proof omitted

3.9.5.2 Definition (4.2.18) The cofactor $\boldsymbol{A}_{ij}$ is the matrix $\boldsymbol{A}$ with the i$^{\text{th}}$ row and j$^{\text{th}}$ column removed.

3.9.5.3 Theorem (4.2.19)

$$|\boldsymbol{A}| = \sum_{i=1}^{n}(-1)^{n+1}a_{ik}|\boldsymbol{A}_{ik}| = \sum_{j=1}^{n}(-1)^{n+1}a_{kj}|\boldsymbol{A}_{kj}|$$

proof omitted

3.9.5.4 Example (4.2.20)

$$\begin{vmatrix} 4 & 3 & 2 \\ 0 & 2 & 3 \\ 2 & 1 & 1 \end{vmatrix} =$$

$$(-1)^{1+1}4\begin{vmatrix} 2 & 3 \\ 1 & 1 \end{vmatrix} + (-1)^{2+1}0\begin{vmatrix} 3 & 2 \\ 1 & 1 \end{vmatrix} + (-1)^{3+1}2\begin{vmatrix} 3 & 2 \\ 2 & 3 \end{vmatrix} =$$

$$4.2.1) + 2(9-4) = -4 + 10 = 6$$

or

```
A=rbind(c(4, 3, 2), c(0, 2, 3), c(2, 1, 1))
A
```

```
##      [,1] [,2] [,3]
## [1,]    4    3    2
## [2,]    0    2    3
## [3,]    2    1    1
```

```
det(A)
```

```
## [1] 6
```

3.9.5.5 Theorem (4.2.21)

i. $|diag(a_1, .., a_n)| = \prod_{i=1}^{n} a_i$

ii. the determinant of a triangular matrix is the product of the diagonal elements.

iii. $\boldsymbol{A}$ is a singular matrix iff det($\boldsymbol{A}$)=0

iv. If $\boldsymbol{A}$ is positive definite $|\boldsymbol{A}| > 0$

v. $|\boldsymbol{A'}| = |\boldsymbol{A}|$

vi. $|\boldsymbol{A^{-1}}| = 1/|\boldsymbol{A}|$

proof

proof of ii: say $\boldsymbol{A}$ is upper tringular, then

$$det(\boldsymbol{A}) = (-1)^{1+1}a_{11}det(\boldsymbol{A}_{11}) = a_{11}det(\boldsymbol{A}_{11}) = .. = \prod a_{ii}$$

proofs of other parts omitted

3.9.5.6 Theorem (4.2.22)

Say $\boldsymbol{A}$ is a square matrix partitioned as

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A_{11}} & \boldsymbol{A_{12}} \\ \boldsymbol{A_{21}} & \boldsymbol{A_{22}} \end{pmatrix}$$

and $\boldsymbol{A}_{11}$ and $\boldsymbol{A}_2$ are square and nonsingular, then

$$|\boldsymbol{A}| = |\boldsymbol{A}_{11}||\boldsymbol{A}_{22} - \boldsymbol{A}_{21}\boldsymbol{A}_{11}^{-1}\boldsymbol{A}_{12}|$$

proof omitted

3.9.5.7 Corollary (4.2.23)

Say $\boldsymbol{A}$ is a square matrix partitioned as

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A_{11}} & \boldsymbol{A_{12}} \\ \boldsymbol{O} & \boldsymbol{A_{22}} \end{pmatrix} \text{or} \boldsymbol{A} = \begin{pmatrix} \boldsymbol{A_{11}} & \boldsymbol{O} \\ \boldsymbol{A_{21}} & \boldsymbol{A_{22}} \end{pmatrix}$$

and $\boldsymbol{A}_{11}$ and $\boldsymbol{A}_2$ are square and nonsingular, then

$$|\boldsymbol{A}| = |\boldsymbol{A}_{11}||\boldsymbol{A}_{22}|$$

proof omitted

3.9.5.8 Theorem (4.2.24)

$$|\boldsymbol{AB}| = |\boldsymbol{A}||\boldsymbol{B}|$$

proof omitted

3.9.5.9 Corollary (4.2.25)

$$|\boldsymbol{A}^n| = |\boldsymbol{A}|^n$$

3.9.6 Orthogonal Vectors and Matrices

3.9.6.1 Definition (4.2.26)

Two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ are said to be *orthogonal* if

$$\boldsymbol{a}'\boldsymbol{b} = \sum_{i=1}^{n} a_i b_i = 0$$

An orthogonal vector is called *orthonormal* if it has length 1.

Geometrically two vectors are orthogonal if they are at right angles (perpendicular) to each other. Let θ be the angle between the two vectors, then

$$\cos\theta = \frac{\boldsymbol{a}'\boldsymbol{a} + \boldsymbol{b}'\boldsymbol{b} - (\boldsymbol{b}-\boldsymbol{a})'(\boldsymbol{b}-\boldsymbol{a})}{2\sqrt{(\boldsymbol{a}'\boldsymbol{a})(\boldsymbol{b}'\boldsymbol{b})}} =$$

$$\frac{\boldsymbol{a}'\boldsymbol{b}}{\sqrt{(\boldsymbol{a}'\boldsymbol{a})(\boldsymbol{b}'\boldsymbol{b})}}$$

so if $\theta = 90^o, \boldsymbol{a}'\boldsymbol{b} = \cos 90^0 = 0$.

A set of vectors where all vectors are mutually orthogonal and normalized is called an *orthonormal set*. A matrix $\boldsymbol{C}$ where all columns form an orthonormal set is called an orthogonal matrix. We have $\boldsymbol{C}'\boldsymbol{C} = \boldsymbol{I}$.

3.9.6.2 Theorem (4.2.27)

Let $\boldsymbol{C}$ be an orthogonal matrix, then

i. $|\boldsymbol{C}| = \pm 1$

ii. $|\boldsymbol{C}'\boldsymbol{A}\boldsymbol{C}| = |\boldsymbol{A}|$

iii. $|c_{ij}| \leq 1$

proof omitted

3.9.7 Trace

3.9.7.1 Definition (4.2.28)

The *trace* of a matrix $\boldsymbol{A}$ is the sum of the diagonal elements of $\boldsymbol{A}$.

3.9.7.2 Example (4.2.29)

$$\boldsymbol{A} = \begin{pmatrix} 1 & 5 & -3 \\ -3 & 2 & 7 \\ 2 & 5 & 9 \end{pmatrix} \operatorname{tr}(\boldsymbol{A}) = 1 + 2 + 9 = 12$$

3.9.7.3 Theorem (4.2.30)

i. $\operatorname{tr}(\boldsymbol{A} \pm \boldsymbol{B}) = \operatorname{tr}(\boldsymbol{A}) \pm \operatorname{tr}(\boldsymbol{B})$

ii. $\operatorname{tr}(\boldsymbol{A}\boldsymbol{B}) = \operatorname{tr}(\boldsymbol{B}\boldsymbol{A})$

iii. $\operatorname{tr}(\boldsymbol{A}'\boldsymbol{A}) = \sum_{i=1}^{n} a_i' a_i$

iv. if $\boldsymbol{P}$ is any nonsingular matrix then

$$\operatorname{tr}(\boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{P}) = \operatorname{tr}(\boldsymbol{A})$$

v. if $\boldsymbol{C}$ is any orthogonal matrix then

$$\text{tr}(\boldsymbol{C}'\boldsymbol{A}\boldsymbol{P}) = \text{tr}(\boldsymbol{A})$$

proof omitted

3.10 Eigenvalues and Eigenvectors, Matrix Calculus

3.10.1 Eigenvalues

3.10.1.1 Definition (4.3.1) For any square matrix $\boldsymbol{A}$ a scalar λ and a vector $\boldsymbol{x}$ can be found such that

$$\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$$

λ is called an *eigenvalue* and $\boldsymbol{x}$ its *eigenvector*.

Note if $\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$ we have $\boldsymbol{A}\boldsymbol{x} - \lambda\boldsymbol{x} = 0$ or $(\boldsymbol{A} - \lambda\boldsymbol{I})\,\boldsymbol{x} = 0$. Therefore $\boldsymbol{A} - \lambda\boldsymbol{I}$ is a singular matrix and $|\boldsymbol{A} - \lambda\boldsymbol{I}| = 0$, which is called the *characteristic equation*.

3.10.1.2 Example (4.3.2)

$$\begin{aligned}
&\boldsymbol{A} = \begin{pmatrix} 1 & 3 \\ -1 & 5 \end{pmatrix} \\
&|\boldsymbol{A} - \lambda\boldsymbol{I}| = \begin{vmatrix} 1-\lambda & 3 \\ -1 & 5-\lambda \end{vmatrix} = \\
&(1-\lambda)(5-\lambda) + 3 = 0 \\
&\lambda^2 - 6\lambda + 8 = (\lambda - 2)(\lambda - 4) = 0
\end{aligned}$$

so we have eigenvalues $\lambda_1 = 2$ and $\lambda_2 = 4$. Now

$$\begin{aligned}
&\boldsymbol{A} - \lambda_1\boldsymbol{I} = 0 \\
&\begin{pmatrix} 1-2 & 3 \\ -1 & 5-2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\
&-x_1 + 3x_2 = 0
\end{aligned}$$

set $x_1 = 1$, then $x_2 = 1/3$. Also

$\sqrt{x_1^2 + x_2^2} = \sqrt{1^2 + (1/3)^2} = \sqrt{10}/3$

so the normalized eigenvector is $(1, 1/3)/(\sqrt{10}/3) = (3, 1)/\sqrt{10}$.

And

$$\boldsymbol{A} - \lambda_2 \boldsymbol{I} = 0$$
$$\begin{pmatrix} 1-4 & 3 \\ -1 & 5-4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$
$$-3x_1 + 3x_2 = 0$$

Again setting $x_1 = 1$, then $x_2 = 1$. Also $\sqrt{x_1^2 + x_2^2} = \sqrt{1^2 + 1^2} = \sqrt{2}$, so the normalized eigenvector is $(1, 1)/\sqrt{2}$.

Let's check:

```
A=rbind(c(1, 3), c(-1, 5))
A
```

```
##      [,1] [,2]
## [1,]    1    3
## [2,]   -1    5
```

```
x=cbind(c(3, 1)/sqrt(10))
cbind(A%*%x, 2*x)
```

```
##           [,1]      [,2]
## [1,] 1.8973666 1.8973666
## [2,] 0.6324555 0.6324555
```

```
x=cbind(c(1, 1)/sqrt(2))
cbind(A%*%x, 4*x)
```

```
##          [,1]     [,2]
## [1,] 2.828427 2.828427
## [2,] 2.828427 2.828427
```

with R we can find eigenvalues and eigenvectors with

```
eigen(A)
```

```
## eigen() decomposition
## $values
## [1] 4 2
##
## $vectors
##            [,1]       [,2]
## [1,] -0.7071068 -0.9486833
## [2,] -0.7071068 -0.3162278
```

3.10.1.3 Theorem (4.3.3) Let $\boldsymbol{A}$ be a matrix with eigenvalue and eigenvector $\lambda, \boldsymbol{x}$. Let c,k be scalars with $c \neq 0, k \neq 0$, Then

i. $c\lambda$, is an eigenvalue of $\boldsymbol{A}$

ii. $c\lambda + k$ is an eigenvalue $c\boldsymbol{A} + k\boldsymbol{I}$

iii. λ^n is an eigenvalue of $\boldsymbol{A}^n$

iv. $1/\lambda$ is an eigenvalue of $\boldsymbol{A}^{-1}$ (if $\boldsymbol{A}^{-1}$ exists)

proof

i.

$$c\boldsymbol{A}\boldsymbol{x} = c\lambda\boldsymbol{x}$$

ii.

$$c(\boldsymbol{A} + k\boldsymbol{I})\boldsymbol{x} = c(\boldsymbol{A}\boldsymbol{x} + k\boldsymbol{I}\boldsymbol{x}) = c\lambda\boldsymbol{x} + k\boldsymbol{x} = (c\lambda + k)\boldsymbol{x}$$

iii.

$$\boldsymbol{A}^2\boldsymbol{x} = \boldsymbol{A}(\boldsymbol{A}\boldsymbol{x}) = \boldsymbol{A}(\lambda\boldsymbol{x}) = \lambda(\boldsymbol{A}\boldsymbol{x}) = \lambda^2\boldsymbol{x}$$

The statement with n follows by repeating this calculation n times.

iv.

$$\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^{-1}\lambda\boldsymbol{x}$$
$$\boldsymbol{A}^{-1}\boldsymbol{x} = \frac{1}{\lambda}\boldsymbol{x}$$

3.10.1.4 Comment Note that if λ, μ are eigenvalues of $\boldsymbol{A}$ and $\boldsymbol{B}$ respectively, in general $\lambda + \mu$ is NOT an eigenvalue of $\boldsymbol{A} + \boldsymbol{B}$.

3.10.1.5 Theorem (4.3.4) Let $\boldsymbol{A}$ be a matrix with eigenvalue and eigenvector $\lambda, \boldsymbol{x}$. Let $p(x) = \sum_{i=0}^{n} a_k x^k$ be a polynomial. Then

$$p(\boldsymbol{A})\boldsymbol{x} = p(\lambda)\boldsymbol{x}$$

proof

$$p(\boldsymbol{A})\boldsymbol{x} = \left(\sum_{i=0}^{n} a_k \boldsymbol{A}^k\right)\boldsymbol{x} =$$
$$\sum_{i=0}^{n} \left(a_k \boldsymbol{A}^k \boldsymbol{x}\right) =$$
$$\sum_{i=0}^{n} a_k \left(\boldsymbol{A}^k \boldsymbol{x}\right) =$$
$$\sum_{i=0}^{n} a_k \left(\lambda^k \boldsymbol{x}\right) =$$
$$\left(\sum_{i=0}^{n} a_k \lambda^k\right)\boldsymbol{x} =$$
$$p(\lambda)\boldsymbol{x}$$

If the respective series are convergent this can sometimes be extended to infinite series:

3.10.1.6 Example (4.3.5) say λ is an eigenvalue of $\boldsymbol{A}$, then $1-\lambda$ is an eigenvalue of $\boldsymbol{I}-\boldsymbol{A}$ by (4.3.3). If $\boldsymbol{I}-\boldsymbol{A}$ is nonsingular $\frac{1}{1-\lambda}$ is an eigenvalue of $(\boldsymbol{I}-\boldsymbol{A})^{-1}$, also by (4.3.3). If $|\lambda| < 1$, then $\frac{1}{1-\lambda} = \sum_{i=0}^{\infty} \lambda^i$ and so

$$(\boldsymbol{I}-\boldsymbol{A})^{-1} = \sum_{i=0}^{\infty} \boldsymbol{A}^i$$

3.10.1.7 Theorem (4.3.6)

i. The eigenvalues of $\boldsymbol{AB}$ are the same as the eigenvalues of $\boldsymbol{BA}$

ii. If P is any nonsingular matrix, then $\boldsymbol{A}$ and $\boldsymbol{P^{-1}AP}$ have the same eigenvalues.

iii. If C is any orthogonal matrix, then $\boldsymbol{A}$ and $\boldsymbol{C'AC}$ have the same eigenvalues.

proof

i. say λ is an eigenvalue of $\boldsymbol{AB}$, then $(\boldsymbol{AB})\boldsymbol{x} = \lambda\boldsymbol{x}$. But then

$$\boldsymbol{BA}(\boldsymbol{Bx}) = \boldsymbol{B}(\boldsymbol{AB})\boldsymbol{x} = \boldsymbol{B}(\lambda\boldsymbol{x}) = \lambda(\boldsymbol{Bx})$$

so λ is an eigenvalue of $\boldsymbol{BA}$ with eigenvector $\boldsymbol{Bx}$.

ii, and iii follow from this.

3.10.2 Symmetric Matrices

3.10.2.1 Theorem (4.3.7) Let $\boldsymbol{A}$ be an $n \times n$ symmetric matrix. Then

i. the eigenvalues of $\boldsymbol{A}$ are real.

ii. the eigenvectors corresponding to distinct eigenvalues are mutually orthogonal.

proof omitted

3.10.2.2 Theorem (4.3.8) Spectral Decomposition

Let $\boldsymbol{A}$ be an $n \times n$ symmetric matrix. Let $\lambda_1, .., \lambda_n$ and $\boldsymbol{x}_1, .., \boldsymbol{x}_n$ be its eigenvalues and eigenvectors. Let $\boldsymbol{D} = diag(\lambda_1, .., \lambda_n)$ and $\boldsymbol{C} = (\boldsymbol{x}_1, .., \boldsymbol{x}_n)$. Then

$$\boldsymbol{A} = \boldsymbol{CDC}' = \sum_{i=1}^{n} \lambda_i \boldsymbol{x}_i \boldsymbol{x}_i'$$

and

$$\boldsymbol{C}'\boldsymbol{AC} = \boldsymbol{D}$$

proof

by its definition $\boldsymbol{C}$ is orthogonal. Therefore $\boldsymbol{I} = \boldsymbol{CC}'$ and so $\boldsymbol{A} = \boldsymbol{ACC}'$ and

$$\begin{aligned}
&\boldsymbol{A} = \\
&\boldsymbol{A}(\boldsymbol{x}_1, .., \boldsymbol{x}_n)\boldsymbol{C}' = \\
&(\boldsymbol{A}\boldsymbol{x}_1, .., \boldsymbol{A}\boldsymbol{x}_n)\boldsymbol{C}' = \\
&(\lambda_1\boldsymbol{x}_1, .., \lambda_n\boldsymbol{x}_n)\boldsymbol{C}' = \\
&\boldsymbol{CDC}'
\end{aligned}$$

3.10.2.3 Theorem (4.3.9) Spectral Decomposition II

Let $\boldsymbol{A}$ be an $n \times n$ nonsingular matrix. Let $\lambda_1, .., \lambda_n$ and $\boldsymbol{x}_1, .., \boldsymbol{x}_n$ be its eigenvalues and eigenvectors. Let $\boldsymbol{D} = diag(\lambda_1, .., \lambda_n)$ and $\boldsymbol{P} = (\boldsymbol{x}_1, .., \boldsymbol{x}_n)$. Then

$$\boldsymbol{A} = \boldsymbol{PDP}^{-1}$$

The spectral decomposition can be used to define functions of matrices: say $\boldsymbol{A} = \boldsymbol{PDP}^{-1}$, then

$$\begin{aligned}
&\boldsymbol{A}^2 = \boldsymbol{AA} = \\
&\boldsymbol{PDP}^{-1}\boldsymbol{PDP}^{-1} = \\
&\boldsymbol{PDDP}^{-1} = \\
&\boldsymbol{PD}^2\boldsymbol{P}^{-1}
\end{aligned}$$

and the square of a diagonal matrix is easy to find. This immediately generalizes to

$$\boldsymbol{A}^n = \boldsymbol{P}\boldsymbol{D}^n\boldsymbol{P}^{-1}$$

Let's say we have a function and we know its power series expansion $f(x) = \sum_{i=0}^{\infty} a_i x^i$, then

$$f(\boldsymbol{A}) = \sum_{i=0}^{\infty} a_i \boldsymbol{A}^i =$$
$$\sum_{i=0}^{\infty} a_i \boldsymbol{P}\boldsymbol{D}^i\boldsymbol{P}^{-1} =$$
$$\boldsymbol{P}\left[\sum_{i=0}^{\infty} a_i \boldsymbol{D}^i\right]\boldsymbol{P}^{-1}$$

3.10.2.4 Example (4.3.10)

$$\boldsymbol{A} = \begin{pmatrix} 1 & 2 \\ 0 & 2 \end{pmatrix}$$

and we want to find $\boldsymbol{B} = \exp(\boldsymbol{A})$.

Let's find its eigenvalues and eigenvectors with R:

```
A=rbind( c(1, 2), c(0, 2) )
EA=eigen(A)
EA
```

```
## eigen() decomposition
## $values
## [1] 2 1
##
## $vectors
##           [,1] [,2]
## [1,] 0.8944272    1
## [2,] 0.4472136    0
```

so $\boldsymbol{A}$ has eigenvalues 2 and 1 and eigenvectors $(0.894, 0.447)'$, $(1, 0)'$. Recall that

$$\exp(x) = \sum_{i=1}^{\infty} \frac{x^i}{i!}$$

and so

$$\boldsymbol{B} = \exp(\boldsymbol{A}) = \boldsymbol{P}\left[\sum_{i=0}^{\infty} \boldsymbol{D}^i/i!\right]\boldsymbol{P}^{-1}$$

now

$$\sum_{i=0}^{\infty} \boldsymbol{D}^i/i! =$$
$$\sum_{i=0}^{\infty} \frac{1}{i!} \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}^i =$$
$$\begin{pmatrix} \sum_{i=0}^{\infty} \frac{2^i}{i!} & 0 \\ 0 & \sum_{i=0}^{\infty} \frac{2^i}{i!} \end{pmatrix} =$$
$$\begin{pmatrix} e^2 & 0 \\ 0 & e \end{pmatrix}$$

and so

```
P=EA$vectors
D1=diag(c(exp(2), exp(1)))
P%*%D1%*%solve(P)
```

```
##          [,1]     [,2]
## [1,] 2.718282 9.341549
## [2,] 0.000000 7.389056
```

3.10.2.5 Theorem (4.3.11) Let $\boldsymbol{A}$ be an $n \times n$ matrix with eigenvalues $\lambda_1, .., \lambda_n$. Then

i. $|\boldsymbol{A}| = \prod_{i=1}^{n} \lambda_i$

ii. $\text{tr}(\boldsymbol{A}) = \sum_{i=1}^{n} \lambda_i$

proof

If $\boldsymbol{A}$ is a symmetric matrix we have

$$|\boldsymbol{A}| = |\boldsymbol{CDC'}| = |\boldsymbol{C}||\boldsymbol{D}||\boldsymbol{C'}| = |\boldsymbol{D}| = \prod_{i=1}^{n} \lambda_i$$

The general case can be found in any linear algebra textbook.

3.10.2.6 Example (4.3.12) Say

$$\boldsymbol{A} = \begin{pmatrix} 1 & 3 \\ -1 & 5 \end{pmatrix}$$

we have previously found the eigenvalues to be 2 and 4. Now

$$det(\boldsymbol{A}) = 1 * 5 - (3 * (-1) = 8 = 2 * 4$$

and

$$tr(\boldsymbol{A}) = 1 + 5 = 6 = 2 + 4$$

3.10.3 Positive Definite Matrices

3.10.3.1 Theorem (4.3.13) Let $\boldsymbol{A}$ be an $n \times n$ matrix with eigenvalues $\lambda_1, .., \lambda_n$. Then

i. if $\boldsymbol{A}$ is positive definite then $\lambda_i > 0$

ii. if $\boldsymbol{A}$ is positive semi-definite then $\lambda_i \geq 0$. The number of eigenvalues > 0 is the rank of $\boldsymbol{A}$.

proof

i.

$$\begin{aligned}
&\boldsymbol{Ax} = \lambda_i \boldsymbol{x} \\
&\boldsymbol{x'Ax} = \boldsymbol{x'}\lambda_i \boldsymbol{x} = \lambda_i \boldsymbol{x'x} \\
&\lambda_i = \frac{\boldsymbol{x'Ax}}{\boldsymbol{x'x}} > 0
\end{aligned}$$

proof of ii omitted

Say $\boldsymbol{A}$ is positive-definite. Let $\boldsymbol{A} = \boldsymbol{CDC'}$ be the spectral decomposition of $\boldsymbol{A}$. Define $\boldsymbol{D}^{1/2} = diag(\sqrt{\lambda_1}, .., \sqrt{\lambda_1})$ (which we can do because all eigenvalues are positive) and define $\boldsymbol{A}^{1/2} = \boldsymbol{CD}^{1/2}\boldsymbol{C'}$. Now

$$\begin{aligned}
&\boldsymbol{A}^{1/2}\boldsymbol{A}^{1/2} = \\
&\boldsymbol{CD}^{1/2}\boldsymbol{C'}\boldsymbol{CD}^{1/2}\boldsymbol{C'} = \\
&\boldsymbol{CD}^{1/2}\boldsymbol{D}^{1/2}\boldsymbol{C'} = \\
&\boldsymbol{CDC'} = \boldsymbol{A}
\end{aligned}$$

and so $\boldsymbol{A}^{1/2}$ is the *square root* of $\boldsymbol{A}$!

3.10.4 Idempotent Matrices

3.10.4.1 Definition (4.3.14) A square matrix $\boldsymbol{A}$ is called *idempotent* if $\boldsymbol{A}^2 = \boldsymbol{A}$

3.10.4.2 Theorem (4.3.15) The only nonsingular idempotent matrix is the identity.

proof

Say $\boldsymbol{A}$ is idempotent and nonsingular, so $\boldsymbol{A}^2 = \boldsymbol{A}$ and $\boldsymbol{A}^{-1}$ exists. Therefore

$$\boldsymbol{A} = (\boldsymbol{A}^{-1}\boldsymbol{A})\boldsymbol{A} = \boldsymbol{A}^{-1}\boldsymbol{A}^2 = \boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{I}$$

3.10.4.3 Theorem (4.3.16) If $\boldsymbol{A}$ is symmetric and idempotent then $\boldsymbol{A}$ is positive semidefinite.

proof

$$\boldsymbol{A} = \boldsymbol{A}^2 = \boldsymbol{A}\boldsymbol{A} = \boldsymbol{A}'\boldsymbol{A}$$

3.10.4.4 Theorem (4.3.17) If $\boldsymbol{A}$ is symmetric and idempotent of rank r, then $\boldsymbol{A}$ has r eigenvalues equal to 1 and n-r eigenvalues equal to 0.

proof

$$\begin{aligned}
&\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x} \\
&\boldsymbol{A}^2\boldsymbol{x} = \lambda^2\boldsymbol{x} \\
&\boldsymbol{A}^2\boldsymbol{x} = \boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x} \\
&\lambda\boldsymbol{x} = \lambda^2\boldsymbol{x} \\
&(\lambda - \lambda^2)\boldsymbol{x} = \lambda(1-\lambda)\boldsymbol{x} = \boldsymbol{0}
\end{aligned}$$

and so all eigenvalues are either 0 or 1.

$\boldsymbol{A}$ is positive semidefinite and therefore the rank of $\boldsymbol{A}$ is equal to the number of positive eigenvalues, that is the number of eigenvalues equal to 1.

3.10.4.5 Corollary (4.3.18) If $\boldsymbol{A}$ is symmetric and idempotent of rank r, then $tr(\boldsymbol{A}) = r$.

3.10.4.6 Theorem (4.3.19) If $\boldsymbol{A}$ is $n \times n$ and idempotent, $\boldsymbol{P}$ is nonsingular, and $\boldsymbol{C}$ is orthogonal we have

i. $\boldsymbol{I} - \boldsymbol{A}$ is idempotent

ii. $\boldsymbol{A}(\boldsymbol{I} - \boldsymbol{A}) = (\boldsymbol{I} - \boldsymbol{A})\boldsymbol{A} = \boldsymbol{O}$

iii. $\boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{P}$ is idempotent

iv. $\boldsymbol{C}'\boldsymbol{A}\boldsymbol{C}$ is idempotent

proof omitted

3.10.5 Vector and Matrix Calculus

There are a number of ways to extend the ideas of calculus to matrices. We will discuss two:

Let $u = f(\boldsymbol{x})$ be a function of variables $\boldsymbol{x} = (x_1, .., x_p)'$ and let $\partial u/\partial x_i$ be the partial derivatives. We define the vector $\partial u/\partial \boldsymbol{x}$ as

$$\frac{\partial u}{\partial \boldsymbol{x}} = \begin{pmatrix} \partial u/\partial x_1 \\ \partial u/\partial x_2 \\ \vdots \\ \partial u/\partial x_p \end{pmatrix}$$

3.10.5.1 Theorem (4.3.20) Let $u = \boldsymbol{a}'\boldsymbol{x}$ where $\boldsymbol{a} = (a_1, .., a_p)'$. Then

$$\frac{\partial u}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{a}'\boldsymbol{x}}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{x}'\boldsymbol{a}}{\partial \boldsymbol{x}} = \boldsymbol{a}$$

proof obvious

3.10.5.2 Theorem (4.3.21) Let $u = \boldsymbol{x}'\boldsymbol{A}\boldsymbol{x}$ where $\boldsymbol{A}$ is a symmetric matrix. Then

$$\frac{\partial u}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{x}'\boldsymbol{A}\boldsymbol{x}}{\partial \boldsymbol{x}} = 2\boldsymbol{A}\boldsymbol{x}$$

proof

$$(\boldsymbol{A}\boldsymbol{x})_k = \sum_{i=1}^{p} a_{ki}x_i$$

$$\boldsymbol{x}'\boldsymbol{A}\boldsymbol{x} = \sum_{j=1}^{p} x_j \left(\sum_{i=1}^{p} a_{ji}x_i \right) = \sum_{i;j=1}^{p} a_{ji}x_i x_j =$$

$$\sum_{i=1}^{p} a_{ii}x_i^2 + \sum_{i \neq j}^{p} a_{ji}x_i x_j$$

$$\frac{\partial \boldsymbol{x}'\boldsymbol{A}\boldsymbol{x}}{\partial x_i} = 2a_{ii}x_i + 2\sum_{j \neq i} a_{ij}x_j = 2\sum_{j} a_{ij}x_j = 2\boldsymbol{a}'\boldsymbol{x}$$

3.10.5.3 Example (4.3.22) Say $\boldsymbol{A} = \begin{pmatrix} 1 & 2 \\ 2 & -1 \end{pmatrix}$, then

$$\boldsymbol{x'Ax} =$$
$$\begin{pmatrix} x_1 & x_2 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 2 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} =$$
$$\begin{pmatrix} x_1 & x_2 \end{pmatrix} \begin{pmatrix} x_1 + 2x_2 \\ 2x_1 - x_2 \end{pmatrix} =$$
$$x_1(x_1 + 2x_2) + x_2(2x_1 - x_2) =$$
$$x_1^2 + 4x_1x_2 - x_2^2$$
$$\frac{\partial \boldsymbol{x'Ax}}{\partial x_1} = 2x_1 + 4x_2$$
$$\frac{\partial \boldsymbol{x'Ax}}{\partial x_2} = -2x_2 + 4x_1$$

but

$$2\boldsymbol{Ax} =$$
$$2 \begin{pmatrix} 1 & 2 \\ 2 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} =$$
$$2 \begin{pmatrix} x_1 + 2x_2 \\ 2x_1 - x_2 \end{pmatrix} =$$
$$\begin{pmatrix} 2x_1 + 4x_2 \\ 4x_1 - 2x_2 \end{pmatrix}$$

Next let $\boldsymbol{X} = (x_{ij})$ be a $p \times p$ matrix of variables and define the function $u = f(\boldsymbol{X})$. Define the matrix of partial derivatives

$$\frac{\partial u}{\partial \boldsymbol{X}} = \begin{pmatrix} \partial u/\partial x_{11} & ... & \partial u/\partial x_{1p} \\ \partial u/\partial x_{21} & ... & \partial u/\partial x_{2p} \\ \vdots & & \vdots \\ \partial u/\partial x_{p1} & ... & \partial u/\partial x_{1p} \end{pmatrix}$$

3.10.5.4 Theorem (4.3.23) Let $u = tr(\boldsymbol{XA})$ where $\boldsymbol{X}$ is a positive definite matrix and $\boldsymbol{A}$ is a matrix of constants. Then

$$\frac{\partial u}{\partial \boldsymbol{X}} = \boldsymbol{A} + \boldsymbol{A'} - diag(\boldsymbol{A})$$

proof

Note that we previously found

$$tr(\boldsymbol{XA}) = \sum_{i,j}^{p} x_{ij}a_{ji} = \sum_{i}^{p} x_{ii}a_{ii} + 2\sum_{i<j}^{p} x_{ij}a_{ji}$$

because $\boldsymbol{X}$ is positive definite and therefore symmetric. So $\partial u/\partial x_{ii} = a_{ii}$ and $\partial u/\partial x_{ij} = a_{ij} + a_{ji}$ if $i \neq j$.

Also

$$\left[\boldsymbol{A} + \boldsymbol{A'} - diag(\boldsymbol{A})\right]_{ii} = a_{ii} + a_{ii} - a_{ii} = a_{ii}$$

and if $i \neq j$

$$\left[\boldsymbol{A} + \boldsymbol{A'} - diag(\boldsymbol{A})\right]_{ij} = a_{ij} + a_{ji}$$

3.10.5.5 Example (4.3.24) Let

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

$\boldsymbol{X}$ is supposed to be positive definite, which implies that it is symmetric and so

$$\boldsymbol{XA} = \begin{pmatrix} x_{11} & x_{12} \\ x_{12} & x_{22} \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} x_{11}a_{11} + x_{12}a_{21} & x_{11}a_{12} + x_{12}a_{22} \\ x_{12}a_{11} + x_{22}a_{21} & x_{12}a_{12} + x_{22}a_{22} \end{pmatrix}$$

and so

$$tr(\boldsymbol{XA}) = x_{11}a_{11} + (a_{21} + a_{12})x_{12} + x_{22}a_{22}$$

and so we find

$$\frac{\partial u}{\partial \boldsymbol{X}} = \begin{pmatrix} a_{11} & a_{12} + a_{21} \\ a_{12} + a_{21} & a_{22} \end{pmatrix}$$

but also

$$\boldsymbol{A} + \boldsymbol{A'} - diag(\boldsymbol{A}) = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} + \begin{pmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{pmatrix} - \begin{pmatrix} a_{11} & 0 \\ 0 & a_{22} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} + a_{21} \\ a_{12} + a_{21} & a_{22} \end{pmatrix}$$

3.10.5.6 Theorem (4.3.25) Let $u = \log|\boldsymbol{X}|$ where $\boldsymbol{X}$ is a positive-definite matrix. Then

$$\frac{\partial u}{\partial \boldsymbol{X}} = 2\boldsymbol{X}^{-1} - diag(\boldsymbol{X}^{-1})$$

proof omitted

3.10.5.7 Example (4.3.26)

$$\boldsymbol{X} = \begin{pmatrix} x_{11} & x \\ x & x_{22} \end{pmatrix} |\boldsymbol{X}| = x_{11}x_{22} - x^2 \log |\boldsymbol{X}| = \log(x_{11}x_{22} - x^2) \frac{\partial u}{\partial \boldsymbol{X}} = \begin{pmatrix} x_{22} & -2x \\ -2x & x_{11} \end{pmatrix} / (x_{11}x_{22} - x^2)$$

also

$$\boldsymbol{X}^{-1} = \frac{1}{x_{11}x_{22} - x^2} \begin{pmatrix} x_{22} & -x \\ -x & x_{11} \end{pmatrix} \quad 2\boldsymbol{X}^{-1} - diag(\boldsymbol{X}^{-1}) = 2\frac{1}{x_{11}x_{22} - x^2} \begin{pmatrix} x_{22} & -x \\ -x & x_{11} \end{pmatrix} - \frac{1}{x_{11}x_{22} - x^2} \begin{pmatrix} x_{22} & 0 \\ 0 & x_{11} \end{pmatrix} =$$

3.10.5.8 Definition (4.3.27) Let A be an $n \times n$ nonsingular matrix with elements (a_{ij}) that are functions of a scalar x. Then we define

$$\frac{\partial \boldsymbol{A}}{\partial x} = \left(\frac{\partial \boldsymbol{a}_{ij}}{\partial x} \right)$$

3.10.5.9 Theorem (4.3.28) Let $\boldsymbol{A}$ be nonsingular of order n with derivative $\frac{\partial \boldsymbol{A}}{\partial x}$. Then

$$\frac{\partial \boldsymbol{A}^{-1}}{\partial x} = -\boldsymbol{A}^{-1} \frac{\partial \boldsymbol{A}}{\partial x} \boldsymbol{A}^{-1}$$

proof

Because $\boldsymbol{A}$ is nonsingular we have

$$\boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{I}$$

Therefore

$$\frac{\partial \boldsymbol{A}^{-1}\boldsymbol{A}}{\partial x} = \frac{\partial \boldsymbol{A}^{-1}}{\partial x} \boldsymbol{A} + \boldsymbol{A}^{-1} \frac{\partial \boldsymbol{A}}{\partial x} = \boldsymbol{O}$$

and so

$$\frac{\partial \boldsymbol{A}^{-1}}{\partial x} \boldsymbol{A} = -\boldsymbol{A}^{-1} \frac{\partial \boldsymbol{A}}{\partial x}$$

and

$$\frac{\partial \boldsymbol{A}^{-1}}{\partial x} = -\boldsymbol{A}^{-1} \frac{\partial \boldsymbol{A}}{\partial x} \boldsymbol{A}^{-1}$$

3.10.5.10 Example (4.3.29) Consider

$$\boldsymbol{A} = \begin{pmatrix} x & x^2 \\ 2x & 1 \end{pmatrix}$$

Let's try and check on this with R:

```
A=function(x) rbind( c(x, x^2), c(2*x, 1) )
A.inf=function(x) solve(A(x))
x=1.5;h=0.001
A.prime = (A(x+h)-A(x))/h
A.inf.prime = (A.inf(x+h)-A.inf(x))/h
round(A.inf.prime, 2)
```

```
##       [,1]  [,2]
## [1,]  0.45 -0.45
## [2,] -0.98  0.49
```

```
round(-solve(A(x))%*%A.prime%*%solve(A(x)), 2)
```

```
##       [,1]  [,2]
## [1,]  0.45 -0.45
## [2,] -0.98  0.49
```

```
A.inf=function(x) solve(A(x))
x=2;h=0.001
A.prime = (A(x+h)-A(x))/h
A.inf.prime = (A.inf(x+h)-A.inf(x))/h
round(A.inf.prime, 2)
```

```
##       [,1]  [,2]
## [1,]  0.12 -0.18
## [2,] -0.33  0.16
```

```
round(-solve(A(x))%*%A.prime%*%solve(A(x)), 2)
```

```
##       [,1]  [,2]
## [1,]  0.12 -0.18
## [2,] -0.33  0.16
```

3.10.5.11 Theorem (4.3.30) Let $\boldsymbol{A}$ be a positive definite matrix. Then

$$\frac{\partial \ln |\boldsymbol{A}|}{\partial x} = tr\left(\boldsymbol{A}^{-1}\frac{\partial \boldsymbol{A}}{\partial x}\right)$$

proof omitted

3.10.5.12 Example (4.3.31) Consider

$$\boldsymbol{A} = \begin{pmatrix} x & x^2 \\ x^2 & 3 \end{pmatrix}$$

$$\frac{\partial \ln |\boldsymbol{A}|}{\partial x} =$$
$$\frac{\partial \ln (3x - x^4)}{\partial x} =$$
$$\frac{3 - 4x^3}{3x - x^4}$$

and

$$\frac{\partial A}{\partial x} = \begin{pmatrix} 1 & 2x \\ 2x & 0 \end{pmatrix}$$
$$\boldsymbol{A}^{-1} = \frac{1}{3x - x^4} \begin{pmatrix} 3 & -x^2 \\ -x^2 & x \end{pmatrix}$$
$$\boldsymbol{A}^{-1} \frac{\partial A}{\partial x} = \frac{1}{3x - x^4} \begin{pmatrix} 3 & -x^2 \\ -x^2 & x \end{pmatrix} \begin{pmatrix} 1 & 2x \\ 2x & 0 \end{pmatrix} =$$
$$\frac{1}{3x - x^4} \begin{pmatrix} 3 - 2x^2 & 6x \\ x^2 & -2x^2 \end{pmatrix}$$
$$tr \left(\boldsymbol{A}^{-1} \frac{\partial A}{\partial x} \right) =$$
$$\frac{1}{3x - x^4} \left(3 - 2x^2 - 2x^2 \right) = \frac{3 - 4x^3}{3x - x^4}$$

3.10.6 Optimization - Lagrange Multipliers

If we have a function $u = f(\boldsymbol{x})$ and we want to find a maxima or minima we may be able to do so by finding solving the equation $\frac{\partial u}{\partial x} = 0$. Often however we need to find a maxima or minima under additional constraints. Let's denote these by $h_i(\boldsymbol{x}) = 0$, i=1,..,q. Using the method of *Lagrange multipliers* means finding an extremum of the function

$$v = u + \sum_{i=1}^{h} \lambda_i h_i(\boldsymbol{x})$$

which means solving the system of equations

$$\frac{\partial u}{\partial x} + \sum_{i=1}^{q} \lambda_i \frac{\partial h_i}{\partial x} = 0$$
$$h_i(\boldsymbol{x}) = 0; i = 1, .., q$$

The $\lambda_i's$ are called the *Lagrange multipliers*

3.10.6.1 Example (4.3.32) Let $f(x,y) = 2x^2 + 3xy + y^2$. Find the minimum of f subject to the constraint $x + 2y = 1$

Here $h(x,y) = x + 2y - 1$, so

$$\frac{\partial f}{\partial x} + \lambda \frac{\partial h}{\partial x} = 4x + 3y + \lambda = 0 \ (I)$$
$$\frac{\partial f}{\partial y} + \lambda \frac{\partial h}{\partial x} = 3x + 2y + 2\lambda = 0 \ (II)$$
$$x + 2y - 1 = 0 \ (III)$$
$$I - II : x + y - \lambda = 0; \lambda = x + y \ (VI)$$
$$VI \to I \ 5x + 4y = 0 \ (V)$$
$$V - 2II \to 3x = -2; x = -2/3$$
$$y = (1 - x)/1 = (1 - (-2/3))/2 = 5/6$$

4 Random Vectors and Distributions

4.1 Distributions of Random Vectors and Random Matrices

4.1.1 Covariance and Correlation

4.1.1.1 Definition (5.1.1) Say we have a random vector $\boldsymbol{X} = (X_1, .., X_p)'$. Let $\sigma_{ij} = cov(X_i, X_j)$, then

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & ... & \sigma_{1p} \\ \sigma_{22} & \sigma_{22} & ... & \sigma_{2p} \\ \vdots & \vdots & & \vdots \\ \sigma_{p1} & \sigma_{p2} & ... & \sigma_{pp} \end{pmatrix}$$

is called the *variance-covariance matrix* or also *covariance matrix*. By definition the covariance matrix is symmetric. If the X_i's are continuous random variables and are linearly independent (that is there do not exist scalars s and r such that $P(sX_i + tX_j = 1) = 1$) then $\boldsymbol{\Sigma}$ is positive definite, otherwise it is positive semi-definite.

Note that

$$var(X_i) = \sigma_{ii} =: \sigma_i^2$$

4.1.1.2 Definition (5.1.2) Let $\boldsymbol{Z}$ be a matrix of random variables, that is

$$\boldsymbol{Z} = \begin{pmatrix} Z_{11} & Z_{12} & ... & Z_{1p} \\ Z_{22} & Z_{22} & ... & Z_{2p} \\ \vdots & \vdots & & \vdots \\ Z_{p1} & Z_{p2} & ... & Z_{pp} \end{pmatrix}$$

then its mean is defined by

$$E[\boldsymbol{Z}] = \begin{pmatrix} E[Z_{11}] & E[Z_{12}] & ... & E[Z_{1p}] \\ E[Z_{22}] & E[Z_{22}] & ... & E[Z_{2p}] \\ \vdots & \vdots & & \vdots \\ E[Z_{p1}] & E[Z_{p2}] & ... & E[Z_{pp}] \end{pmatrix}$$

4.1.1.3 Theorem (5.1.3)

Let $\boldsymbol{X} = (X_1, .., X_p)'$ and let $\boldsymbol{\mu} = (E[X_1], .., E[X_p])'$, then

$$\boldsymbol{\Sigma} = E[(\boldsymbol{X} - \boldsymbol{\mu})(\boldsymbol{X} - \boldsymbol{\mu})'] = E[\boldsymbol{X}\boldsymbol{X}'] - \boldsymbol{\mu}\boldsymbol{\mu}'$$

proof straight-forward

4.1.1.4 Definition (5.1.4)

i. $|\boldsymbol{\Sigma}|$ is called the *generalized variance*.

ii. $(\boldsymbol{X} - \boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{X} - \boldsymbol{\mu})$ is called the *standardized distance* or *Mahalanobis distance*

Note that if $\boldsymbol{Z} = (\boldsymbol{X} - \boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{X} - \boldsymbol{\mu})$, then $E[Z_i] = 0$ and $cov(Z_i, Z_j) = \delta_{ij}$.

4.1.1.5 Definition (5.1.5)

The *correlation matrix* is defined by

$$\boldsymbol{P}_\rho = \begin{pmatrix} 1 & \rho_{12} & ... & \rho_{1p} \\ \rho_{22} & 1 & ... & \rho_{2p} \\ \vdots & \vdots & & \vdots \\ \rho_{p1} & \rho_{p2} & ... & 1 \end{pmatrix}$$

where $\rho_{ij} = \sigma_{ij}/\sigma_{ii}\sigma_{jj}$.

If we define

$$\boldsymbol{D}_\sigma = [diag(\boldsymbol{\Sigma})]^{1/2} = diag(\sigma_1, .., \sigma_p)$$

then

$$\boldsymbol{\Sigma} = \boldsymbol{D}_\sigma \boldsymbol{P}_\rho \boldsymbol{D}_\sigma$$

Suppose a random vector $\boldsymbol{V}$ is partioned as follows:

$$\boldsymbol{V} = \begin{pmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \end{pmatrix} = \begin{pmatrix} X_1 \\ \vdots \\ X_p \\ Y_1 \\ \vdots \\ Y_q \end{pmatrix}$$

then

$$\boldsymbol{\mu} = E[\boldsymbol{V}] = \begin{pmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{pmatrix}$$

and

$$\boldsymbol{\Sigma} = cov[\boldsymbol{V}] = \begin{pmatrix} \boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xy} \\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_{yy} \end{pmatrix}$$

where

$$\boldsymbol{\Sigma}_{yx} = \boldsymbol{\Sigma}'_{xy}$$

4.1.2 Linear Functions of Random Vectors

Let $\boldsymbol{X} = (X_1, .., X_p)'$ be a random vector and $\boldsymbol{a} = (a_1, .., a_p)'$ a vector of scalars. Let $\boldsymbol{Z} = \boldsymbol{a}'\boldsymbol{X}$.

4.1.2.1 Theorem (5.1.6) $\boldsymbol{\mu}_z = E[\boldsymbol{Z}] = E[\boldsymbol{a}'\boldsymbol{X}] = \boldsymbol{a}'\boldsymbol{\mu}$

proof follows from the linearity of expectations.

4.1.2.2 Theorem (5.1.7)

Suppose $\boldsymbol{Y}$ is a random vector, $\boldsymbol{X}$ a random matrix, $\boldsymbol{a}$ and $\boldsymbol{b}$ vectors of constants and $\boldsymbol{A}$ and $\boldsymbol{B}$ matrices of constants. Then

i. $E[\boldsymbol{A}\boldsymbol{Y}] = \boldsymbol{A}E[\boldsymbol{Y}]$

ii. $E[\boldsymbol{a}'\boldsymbol{X}\boldsymbol{b}] = \boldsymbol{a}'E[\boldsymbol{X}]\boldsymbol{b}$

iii. $E[\boldsymbol{A}\boldsymbol{X}\boldsymbol{B}] = \boldsymbol{A}E[\boldsymbol{X}]\boldsymbol{B}$

iv. $E[\boldsymbol{A}\boldsymbol{Y} + \boldsymbol{b}] = \boldsymbol{A}E[\boldsymbol{Y}] + \boldsymbol{b}$

proof follows from linearity of expectations

4.1.2.3 Theorem (5.1.8)

$$var(\boldsymbol{a}'\boldsymbol{Y}) = \boldsymbol{a}'\boldsymbol{\Sigma}\boldsymbol{a}$$

proof

$$\begin{aligned}
&var(\boldsymbol{a}'\boldsymbol{Y}) = \\
&E\left[(\boldsymbol{a}'\boldsymbol{Y} - \boldsymbol{a}'\boldsymbol{\mu})^2\right] = \\
&E\left[(\boldsymbol{a}'(\boldsymbol{Y} - \boldsymbol{\mu}))^2\right] = \\
&E\left[\boldsymbol{a}'(\boldsymbol{Y} - \boldsymbol{\mu})\boldsymbol{a}'(\boldsymbol{Y} - \boldsymbol{\mu})\right] = \\
&E\left[\boldsymbol{a}'(\boldsymbol{Y} - \boldsymbol{\mu})(\boldsymbol{Y} - \boldsymbol{\mu})'\boldsymbol{a}\right] = \\
&\boldsymbol{a}'\boldsymbol{\Sigma}\boldsymbol{a}
\end{aligned}$$

4.1.2.4 Corollary (5.1.9)

$$cov(\boldsymbol{a}'\boldsymbol{Y}, \boldsymbol{b}'\boldsymbol{Y}) = \boldsymbol{a}'\boldsymbol{\Sigma}\boldsymbol{b}$$

4.1.2.5 Theorem (5.1.10)

i.

$$cov(\boldsymbol{A}\boldsymbol{Y}) = \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}'$$

ii.

$$cov(\boldsymbol{A}\boldsymbol{Y}, \boldsymbol{B}\boldsymbol{Y}) = \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{B}'$$

iii.

$$cov(\boldsymbol{A}\boldsymbol{Y} + \boldsymbol{b}) = \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}'$$

iv.

$$cov(\boldsymbol{A}\boldsymbol{Y}, \boldsymbol{B}\boldsymbol{X}) = \boldsymbol{A}\boldsymbol{\Sigma}_{yx}\boldsymbol{B}'$$

proof omitted

4.1.2.6 Theorem (5.1.11)

Let $\boldsymbol{z} = \boldsymbol{A}\boldsymbol{y}$ and $\boldsymbol{w} = \boldsymbol{B}\boldsymbol{y}$, where $\boldsymbol{A}$ is a $k \times p$ matrix of constants and $\boldsymbol{B}$ is a $m \times p$ matrix of constants and $\boldsymbol{y}$ is a $p \times 1$ random vector with covariance matrix $\boldsymbol{\Sigma}$. Then

i.

$$cov(\boldsymbol{A}\boldsymbol{y}) = \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}'$$

ii.

$$cov(\boldsymbol{z}, \boldsymbol{w}) = \boldsymbol{A\Sigma B'}$$

proof follows from the corollary above

4.2 Multivariate Normal Density

4.2.1 Normal (Gaussian) Distribution

Let $\boldsymbol{Z} = (Z_1, .., Z_p)$ be independent standard normal random variables, that is

$$f_{Z_i}(x) = \frac{1}{\sqrt{2\pi}} \exp\{-x^2/2\}$$

Then we have

$$f_{\boldsymbol{Z}}(z_1, .., z_p) = \prod_{i=1}^{p} f_{Z_i}(z_i) =$$
$$\prod_{i=1}^{p} \frac{1}{\sqrt{2\pi}} \exp\{-z_i^2/2\} =$$
$$\frac{1}{(\sqrt{2\pi})^p} \exp\{-\sum_{i=1}^{p} z_i^2/2\} =$$
$$\frac{1}{(\sqrt{2\pi})^p} \exp\{-\boldsymbol{z'z}/2\}$$

A random vector with this density is said to have a *multivariate normal distribution* with mean vector $\boldsymbol{0}$ and covariance matrix $\boldsymbol{I}$.

Notation: $\boldsymbol{Z} \sim N_p(\boldsymbol{0}, \boldsymbol{I})$

4.2.1.1 Theorem (5.2.1) Let $\boldsymbol{\Sigma}^{1/2}$ be a symmetric square root matrix as defined before, and let $\boldsymbol{X} = \boldsymbol{\Sigma}^{1/2}\boldsymbol{Z} + \boldsymbol{\mu}$, then

i. $E[\boldsymbol{X}] = \boldsymbol{\mu}$

ii. $cov(\boldsymbol{X}) = \boldsymbol{\Sigma}$

proof

i.

$$E[\boldsymbol{X}] = E[\boldsymbol{\Sigma}^{1/2}\boldsymbol{Z} + \boldsymbol{\mu}] = \boldsymbol{\Sigma}^{1/2}E[\boldsymbol{Z}] + \boldsymbol{\mu} = \boldsymbol{\Sigma}^{1/2}\boldsymbol{0} + \boldsymbol{\mu} = \boldsymbol{\mu}$$

ii.

$$cov(\boldsymbol{X}) = cov(\boldsymbol{\Sigma}^{1/2}\boldsymbol{Z} + \boldsymbol{\mu}) = \boldsymbol{\Sigma}^{1/2}cov(\boldsymbol{Z})(\Sigma^{1/2})' = \boldsymbol{\Sigma}^{1/2}\boldsymbol{I}(\Sigma^{1/2})' = \boldsymbol{\Sigma}$$

4.2.1.2 Theorem (5.2.2)

The density of $\boldsymbol{X}$ is given by

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{1}{(\sqrt{2\pi})^p |\boldsymbol{\Sigma}|^{1/2}} \exp\{-(\boldsymbol{y}-\boldsymbol{\mu})' |\boldsymbol{\Sigma}|^{-1} (\boldsymbol{y}-\boldsymbol{\mu})/2\}$$

proof

By the change of variable formula from calculus we have the following: say $\boldsymbol{y} = \boldsymbol{\Sigma}^{1/2}\boldsymbol{z} + \boldsymbol{\mu}$

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = f_{\boldsymbol{Z}}(\boldsymbol{z}) \text{abs}(|\boldsymbol{\Sigma}^{-1/2}|)$$

$|\boldsymbol{\Sigma}^{-1/2}|$ is called the *Jacobian* of the transformation.

Now $\boldsymbol{\Sigma}^{-1/2}$ is positive definite, so the Jacobian is also positive and we have

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = f_{\boldsymbol{Z}}(\boldsymbol{z}) |\boldsymbol{\Sigma}^{-1/2}| = f_{\boldsymbol{Z}}(\boldsymbol{z}) |\boldsymbol{\Sigma}|^{-1/2}$$

$\boldsymbol{x} = \boldsymbol{\Sigma}^{1/2}\boldsymbol{z} + \boldsymbol{\mu}$ implies $\boldsymbol{z} = \boldsymbol{\Sigma}^{-1/2}(\boldsymbol{x} - \boldsymbol{\mu})$

Finally

$$\begin{aligned}
&f_{\boldsymbol{X}}(\boldsymbol{x}) = f_{\boldsymbol{Z}}(\boldsymbol{z}) |\boldsymbol{\Sigma}|^{-1/2} = \\
&\frac{1}{(\sqrt{2\pi})^p} \exp\{-\boldsymbol{z}'\boldsymbol{z}/2\} |\boldsymbol{\Sigma}|^{-1/2} = \\
&\frac{1}{(\sqrt{2\pi})^p |\boldsymbol{\Sigma}|^{1/2}} \exp\{-\boldsymbol{z}'\boldsymbol{z}/2\} = \\
&\frac{1}{(\sqrt{2\pi})^p |\boldsymbol{\Sigma}|^{1/2}} \exp\{-\left(\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{x}-\boldsymbol{\mu})\right)' \left(\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{x}-\boldsymbol{\mu})\right)/2\} = \\
&\frac{1}{(\sqrt{2\pi})^p |\boldsymbol{\Sigma}|^{1/2}} \exp\{-(\boldsymbol{x}-\boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1/2}\boldsymbol{\Sigma}^{-1/2} (\boldsymbol{x}-\boldsymbol{\mu})/2\} = \\
&\frac{1}{(\sqrt{2\pi})^p |\boldsymbol{\Sigma}|^{1/2}} \exp\{-(\boldsymbol{x}-\boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})/2\}
\end{aligned}$$

Notation: $\boldsymbol{X} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

4.2.1.3 Corollary (5.2.3)

Let $\boldsymbol{X} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and let $\boldsymbol{Z} = \boldsymbol{\Sigma}^{-1/2}(\boldsymbol{X} - \boldsymbol{\mu})$, then $\boldsymbol{Z} \sim N(\boldsymbol{0}, \boldsymbol{I})$

4.2.1.4 Example (5.2.4)

Let p=1, then

$\Sigma = [a]$, $x'\Sigma x = ax^2 \geq 0$ iff $a \geq 0$

$|\Sigma| = a, \Sigma^{-1} = 1/a$, and

$$f_X(x) = \frac{1}{(\sqrt{2\pi})^p |\boldsymbol{\Sigma}|^{1/2}} \exp\{-(\boldsymbol{x}-\boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})/2\}$$
$$\frac{1}{\sqrt{2\pi} a^{1/2}} \exp\{-(x-\mu)\frac{1}{a}(x-\mu)/2\} =$$
$$\frac{1}{\sqrt{2\pi a}} \exp\{-\frac{(x-\mu)^2}{2a}\}$$

4.2.2 Moment Generating Functions

4.2.2.1 Definition (5.2.5) The *moment generating function* of a random vector $\boldsymbol{X}$ is defined by

$$\psi(\boldsymbol{t}) = E\left[\exp\{\boldsymbol{t}'\boldsymbol{X}\}\right]$$

4.2.2.2 Example (5.2.6) Let $Z \sim N(0,1)$, then

$$\psi(t) = E\left[\exp tZ\right] =$$
$$\int_{-\infty}^{\infty} e^{tz} \frac{1}{\sqrt{2\pi}} \exp\{-\frac{z^2}{2}\} dz =$$
$$\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\{tz - \frac{z^2}{2}\} dz =$$
$$\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\{-\frac{1}{2}(z^2 - 2tz + t^2) + t^2/2\} dz =$$
$$\exp\{t^2/2\} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\{-\frac{1}{2}(z-t)^2\} dz =$$
$$\exp\{t^2/2\}$$

because the integral is over a N(0,t) random variable and therefore equal to 1.
Let $X \sim N(\mu, \sigma)$, then

$$\psi_X(t) = E\left[\exp tX\right] =$$
$$E\left[\exp t(\sigma Z + \mu)\right] =$$
$$e^{\mu t} E\left[\exp(\sigma t) Z\right] =$$
$$e^{\mu t} \psi_Z(\sigma t) =$$
$$e^{\mu t} \exp\{(\sigma t)^2/2\} =$$
$$\exp\{\sigma^2 t^2/2 + \mu t\}$$

4.2.2.3 Theorem (5.2.7) Say $\boldsymbol{X} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then

$$\psi(\boldsymbol{t}) = \exp\{\boldsymbol{t}'\boldsymbol{\mu} + \frac{1}{2}\boldsymbol{t}'\boldsymbol{\Sigma}\boldsymbol{t}\}$$

proof

Let $\boldsymbol{Z} \sim N(\boldsymbol{0}, \boldsymbol{I})$, then

$$\begin{aligned}
&\psi_{\boldsymbol{Z}}(\boldsymbol{t}) = \int .. \int e^{\boldsymbol{t}'\boldsymbol{x}}(2\pi)^{-p/2} \exp\{-\boldsymbol{x}'\boldsymbol{x}/2\}\, d\boldsymbol{x} = \\
&\int .. \int (2\pi)^{-p/2} \exp\{\boldsymbol{t}'\boldsymbol{x} - \boldsymbol{x}'\boldsymbol{x}/2\}\, d\boldsymbol{x} = \\
&\int .. \int (2\pi)^{-p/2} \exp\left\{-\frac{1}{2}\left[\boldsymbol{x}'\boldsymbol{x} - 2\boldsymbol{t}'\boldsymbol{x}\right]\right\} d\boldsymbol{x} = \\
&\int .. \int (2\pi)^{-p/2} \exp\left\{-\frac{1}{2}\left[\boldsymbol{x}'\boldsymbol{x} - 2\boldsymbol{t}'\boldsymbol{x} + \boldsymbol{t}'\boldsymbol{t} - \boldsymbol{t}'\boldsymbol{t}\right]\right\} d\boldsymbol{x} = \\
&\int .. \int (2\pi)^{-p/2} \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{t})'(\boldsymbol{x} - \boldsymbol{t}) + \boldsymbol{t}'\boldsymbol{t}/2\right\} d\boldsymbol{x} = \\
&e^{\boldsymbol{t}'\boldsymbol{t}/2} \int .. \int (2\pi)^{-p/2} \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{t})'(\boldsymbol{x} - \boldsymbol{t})\right\} d\boldsymbol{x} = \\
&e^{\boldsymbol{t}'\boldsymbol{t}/2}
\end{aligned}$$

and the general case follows from the transformation $\boldsymbol{x} = \boldsymbol{\Sigma}^{1/2}\boldsymbol{z} + \boldsymbol{\mu}$.

Recall two properties of moment generating functions:

i. if two random vectors have the same mgf they have the same distribution.

ii. If X and Y are independent then $\psi_{(X,Y)'}((x,y)') = \psi_X(x)\psi_Y(y)$

4.2.2.4 Theorem (5.2.8) Say $\boldsymbol{X} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, $\boldsymbol{a}$ and $\boldsymbol{b}$ are vectors of constants and $\boldsymbol{A}$ a matrix of constants. Then

i. $Y = \boldsymbol{a}'\boldsymbol{X} \sim N(\boldsymbol{a}'\boldsymbol{\mu}, \boldsymbol{a}'\boldsymbol{\Sigma}\boldsymbol{a})$

ii. $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} \sim N(\boldsymbol{A}\boldsymbol{\mu}, \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}')$

iii. $\boldsymbol{T} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{b} \sim N(\boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b}, \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}')$

proof

i.

$$\begin{aligned}
&\psi_Y(t) = E[e^{t\boldsymbol{a}'\boldsymbol{Z}}] = E[e^{(t\boldsymbol{a})'\boldsymbol{Z}}] = \\
&\psi_{\boldsymbol{Z}}(t\boldsymbol{a}) = e^{(t\boldsymbol{a})'(t\boldsymbol{a})/2} = \\
&e^{(\boldsymbol{a}'\boldsymbol{a})t^2/2}
\end{aligned}$$

$$\psi_Y(t) = E[e^{t\boldsymbol{a}'\boldsymbol{X}}] =$$
$$E[e^{(t\boldsymbol{a})'\boldsymbol{X}}] = \psi_{\boldsymbol{X}}(t\boldsymbol{a}) =$$
$$\exp\{(t\boldsymbol{a})'\boldsymbol{\mu} - (t\boldsymbol{a})'\Sigma(t\boldsymbol{a})/2\} =$$
$$\exp\{(\boldsymbol{a}'\boldsymbol{\mu})t - (\boldsymbol{a}'\Sigma\boldsymbol{a})t^2/2\}$$

and this is the mgf of a normal random variable with mean $\boldsymbol{a}'\boldsymbol{\mu}$ and variance $\boldsymbol{a}'\Sigma\boldsymbol{a}$.

ii and iii are done similarly.

Note that one can use this theorem to derive the mgf of a multivariate normal random vector from the mgf of a normal rv: by the theorem $Y = \boldsymbol{t}'\boldsymbol{X}$ has a normal distribution with mean $\boldsymbol{t}'\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{t}'\boldsymbol{\Sigma}\boldsymbol{t}$, so

$$\psi_{\boldsymbol{X}}(\boldsymbol{t}) = E[\exp\left\{\boldsymbol{t}'\boldsymbol{X}\right\}] =$$
$$E[\exp\{Y\}] = \psi_Y(1) =$$
$$\exp\left\{(\boldsymbol{t}'\boldsymbol{\mu})1 + \frac{1}{2}(\boldsymbol{t}'\boldsymbol{\Sigma}\boldsymbol{t})1\right\} =$$
$$\exp\left\{\boldsymbol{t}'\boldsymbol{\mu} + \frac{1}{2}\boldsymbol{t}'\boldsymbol{\Sigma}\boldsymbol{t}\right\}$$

4.2.2.5 Theorem (5.2.9)

i. The marginal distributions of a multivariate normal distribution are also multivariate normal.

ii. $X_i \sim N(\mu_i, \sigma_{ii})$

proof

i. Say we want to find the marginal $\boldsymbol{Y}$. Then there exists a matrix $\boldsymbol{A}$ (of 0's and 1's) such that $\boldsymbol{A}\boldsymbol{X} = \boldsymbol{Y}$ and the result follows from (4.3.9ii).

ii. direct consequence of i.

4.2.2.6 Theorem (5.2.10)

if $\boldsymbol{V} = \begin{pmatrix}\boldsymbol{X}\\ \boldsymbol{Y}\end{pmatrix}$ is $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then $\boldsymbol{X}$ and $\boldsymbol{Y}$ are independent if and only if $\boldsymbol{\Sigma}_{xy} = \boldsymbol{O}$

proof

Suppose $\boldsymbol{\Sigma}_{xy} = \boldsymbol{O}$. Then

$$\boldsymbol{\Sigma} = \begin{pmatrix}\Sigma_{xx} & \boldsymbol{O}\\ \boldsymbol{O} & \Sigma_{yy}\end{pmatrix}$$

and so

$$\boldsymbol{t}'\boldsymbol{\mu} + \frac{1}{2}\boldsymbol{t}'\boldsymbol{\Sigma}\boldsymbol{t} =$$
$$(\boldsymbol{t}_x', \boldsymbol{t}_y') \begin{pmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{pmatrix} + \frac{1}{2}(\boldsymbol{t}_x', \boldsymbol{t}_y') \begin{pmatrix} \Sigma_{xx} & \boldsymbol{O} \\ \boldsymbol{O} & \Sigma_{yy} \end{pmatrix} \begin{pmatrix} \boldsymbol{t}_x \\ \boldsymbol{t}_y \end{pmatrix} =$$
$$\boldsymbol{t}_x'\boldsymbol{\mu}_x + \boldsymbol{t}_y'\boldsymbol{\mu}_y + \frac{1}{2}\boldsymbol{t}_x'\boldsymbol{\Sigma}_{xx}\boldsymbol{t}_x + \frac{1}{2}\boldsymbol{t}_y'\boldsymbol{\Sigma}_{yy}\boldsymbol{t}_y$$
$$\psi_{\boldsymbol{V}}(\boldsymbol{t}) = \exp\{\boldsymbol{t}_x'\boldsymbol{\mu}_x + \frac{1}{2}\boldsymbol{t}_x'\boldsymbol{\Sigma}_{xx}\boldsymbol{t}_x\} \exp\{\boldsymbol{t}_y'\boldsymbol{\mu}_y + \frac{1}{2}\boldsymbol{t}_y'\boldsymbol{\Sigma}_{yy}\boldsymbol{t}_y\}$$

which is the product of two mgf's of multitvariate normals and therefore $\boldsymbol{X}$ and $\boldsymbol{Y}$ are independent.

4.2.2.7 Corollary (5.2.11)

If $\boldsymbol{X} \sim N(\boldsymbol{\mu}, \Sigma)$, then $X_i \perp X_j$ iff $\sigma_{ij} = 0$

4.2.2.8 Corollary (5.2.12)

If $\boldsymbol{X} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and if $cov(\boldsymbol{AX}, \boldsymbol{BX}) = \boldsymbol{A\Sigma B}' = \boldsymbol{O}$ then $\boldsymbol{AX} \perp \boldsymbol{BX}$.

4.2.3 Conditional Distributions

4.2.3.1 Theorem (5.2.13)

if $\boldsymbol{V} = \begin{pmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \end{pmatrix}$ is $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $\boldsymbol{\Sigma}_{xy} \neq \boldsymbol{O}$, then the conditional distribution of $\boldsymbol{Y}|\boldsymbol{X} = \boldsymbol{x}$ is multivariate normal with

$$E[\boldsymbol{Y}|\boldsymbol{X} = \boldsymbol{x}] = \boldsymbol{\mu}_y + \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_x)$$
$$cov(\boldsymbol{Y}|\boldsymbol{X} = \boldsymbol{x}) = \boldsymbol{\Sigma}_{yy} - \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy}$$

proof

We have

$$f_{\boldsymbol{Y}|\boldsymbol{X}=\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{x}) = \frac{f(\boldsymbol{x}, \boldsymbol{y})}{f_{\boldsymbol{X}}(\boldsymbol{x})}$$

and this ratio can be evaluated directly from the definitions.

4.2.3.2 Example (5.2.14)

Say $\begin{pmatrix} X \\ Y \end{pmatrix}$ is bivariate normal with mean vector $(0, 0)'$ and covariance matrix $\Sigma = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$. So

$$det(\Sigma) = 1 - \rho^2$$
$$\Sigma^{-1} = \frac{1}{1-\rho^2}\begin{pmatrix} 1 & -\rho \\ -\rho & 1 \end{pmatrix}$$
$$(x\ y)' \frac{1}{1-\rho^2}\begin{pmatrix} 1 & -\rho \\ -\rho & 1 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} =$$
$$\frac{1}{1-\rho^2}(x\ y)'\begin{pmatrix} x-\rho y) \\ -\rho x + y) \end{pmatrix} =$$
$$\frac{1}{1-\rho^2}\left(x^2 - 2\rho xy + y^2\right)$$

the marginal is given by

$$f_X(x) = (2\pi)^{-1/2}\exp\{-\frac{1}{2}x^2\}$$

$$f_{Y|X=x}(y|x) = \frac{f(x,y)}{f_X(x)} =$$
$$\frac{(2\pi\sqrt{1-\rho^2})^{-1}\exp\{-\frac{1}{2(1-\rho^2)}(x^2 - 2\rho xy + y^2)\}}{(2\pi)^{-1/2}\exp\{-\frac{1}{2}x^2\}} =$$
$$(2\pi(1-\rho^2))^{-1/2}\exp\left\{-\frac{1}{2}\left[\frac{1}{1-\rho^2}(x^2 - 2\rho xy + y^2) - x^2\right]\right\} =$$
$$\frac{1}{\sqrt{2\pi(1-\rho^2)}}\exp\left\{-\frac{1}{2(1-\rho^2)}\left[x^2 - 2\rho xy + y^2 - (1-\rho^2)x^2\right]\right\} =$$
$$\frac{1}{\sqrt{2\pi(1-\rho^2)}}\exp\left\{-\frac{1}{2(1-\rho^2)}\left[y^2 - 2\rho xy + \rho^2 x^2\right]\right\} =$$
$$\frac{1}{\sqrt{2\pi(1-\rho^2)}}\exp\left\{-\frac{1}{2(1-\rho^2)}(y-\rho x)^2\right\}$$

and so $Y|X = x \sim N(\rho x, 1-\rho^2)$.

4.2.3.3 Example (5.2.15)

Say

$$\begin{pmatrix} X \\ Y \end{pmatrix} \sim N\left(\begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 & 0.4 \\ 0.4 & 1 \end{pmatrix}\right)$$

We want to find $E[X|Y = 2.3]$

$$E[\boldsymbol{X}|\boldsymbol{Y}=y]=\mu_x+\boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}(y-\mu_y)$$
$$\mu_y=2$$
$$\boldsymbol{\Sigma}_{xy}=0.4$$
$$\boldsymbol{\Sigma}_{yy}=1$$
$$\boldsymbol{\Sigma}_{yy}^{-1}=1$$
$$E[X|Y=2.3]=$$
$$1+0.4\times 1(2.3-2)=1.12$$

Let's see whether we can verify that with R

```
library(mvtnorm)
mu=c(1, 2)
vc=rbind(c(1, 0.4), c(0.4, 1))
xy=rmvnorm(1e5, mu, vc)
x=xy[ abs(xy[,2]-2.3)<0.1, 1]
length(x)
```

```
## [1] 7710
```

```
mean(x)
```

```
## [1] 1.137246
```

4.2.3.4 Example (5.2.16) Say

$$\begin{pmatrix}X\\Y\\Z\end{pmatrix}\sim N\left(\begin{pmatrix}1\\2\\3\end{pmatrix},\begin{pmatrix}1&0.4&0.6\\0.4&1&-0.2\\0.6&-0.2&1\end{pmatrix}\right)$$

Find $E[X|Y=2.3,Z=2.8]$

$$E[X|Y=y,Z=z] = \mu_x + \mathbf{\Sigma}_{x(yz)}\mathbf{\Sigma}^{-1}_{(yz)(yz)}\begin{pmatrix} y-\mu_y \\ z-\mu_z \end{pmatrix}$$

$$(\mu_y \ \mu_z)' = (2 \ 3)'$$

$$\mathbf{\Sigma}_{x(yz)} = (0.4 \ 0.6)$$

$$\mathbf{\Sigma}_{(yz)(yz)} = \begin{pmatrix} 1 & -0.2 \\ -0.2 & 1 \end{pmatrix}$$

$$\mathbf{\Sigma}^{-1}_{(yz)(yz)} = \frac{1}{0.96}\begin{pmatrix} 1 & 0.2 \\ 0.2 & 1 \end{pmatrix}$$

$$E[X|Y=2.3,Z=2.8] =$$

$$1 + (0.4 \ 0.6)\frac{1}{0.96}\begin{pmatrix} 1 & 0.2 \\ 0.2 & 1 \end{pmatrix}\begin{pmatrix} 2.3-2 \\ 2.8-3 \end{pmatrix} =$$

$$1 + \frac{1}{0.96}(0.4 \ 0.6)\begin{pmatrix} 0.23 \\ -0.14 \end{pmatrix} =$$

$$1 + 0.02/0.96 = 1.021$$

Let's see whether we can verify that with R

```
library(mvtnorm)
mu=c(1, 2, 3)
vc=rbind(c(1, 0.4, 0.6), c(0.4, 1, -0.2), c(0.6, -0.2, 1) )
xyz=rmvnorm(1e5, mu, vc)
round(cor(xyz), 3)
```

```
##       [,1]   [,2]   [,3]
## [1,] 1.000  0.403  0.600
## [2,] 0.403  1.000 -0.197
## [3,] 0.600 -0.197  1.000
```

```
xy=xyz[ abs(xyz[,3]-2.8)<0.075, 1:2]
x=xy[ abs(xy[,2]-2.3)<0.075, 1]
length(x)
```

```
## [1] 351
```

```
mean(x)
```

```
## [1] 1.030432
```

4.2.4 Partial Correlation

Let $\boldsymbol{V} \sim N_{p+q}(\boldsymbol{\mu}, \mathbf{\Sigma})$ and let $\boldsymbol{V}$ be partitioned as follows:

$$\boldsymbol{V} = \begin{pmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \end{pmatrix}$$

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{pmatrix}$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xy} \\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_{yy} \end{pmatrix}$$

Denote the covariance of the conditional distribution of $\boldsymbol{Y}$ given $\boldsymbol{X}$ by

$$\sigma_{ij \cdot rs..q}$$

where X_i, X_j are two of the variables in $\boldsymbol{X}$ and $Y_r, Y_s, .., Y_q$ are variables in $\boldsymbol{Y}$. For example $\sigma_{23 \cdot 124}$ is the covariance between X_2 and X_3 in the conditional distribution of $V_1, .., V_4$ given $V_5, .., V_9$ (say).

The *partial correlation coefficient* $\rho_{ij \cdot rs..q}$ is defined in the usual way:

$$\rho_{ij \cdot rs..q} = \frac{\sigma_{ij \cdot rs..q}}{\sqrt{\sigma_{ii \cdot rs..q} \sigma_{jj \cdot rs..q}}}$$

4.2.4.1 Example (5.2.17) Say $\boldsymbol{V}$ is a multivariate normal random variable with covariance matrix

$$\boldsymbol{\Sigma} = \begin{pmatrix} 10 & 0 & 1 & -2 \\ 0 & 5 & 3 & -2 \\ 1 & 3 & 4 & 1 \\ -2 & -2 & 1 & 6 \end{pmatrix}$$

and we use the partition $\begin{pmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \end{pmatrix}$ with

$$\boldsymbol{\Sigma} = \left(\begin{array}{cc|cc} 10 & 0 & 1 & -2 \\ 0 & 5 & 3 & -2 \\ - & - & - & - \\ 1 & 3 & 4 & 1 \\ -2 & -2 & 1 & 6 \end{array}\right)$$

$$cov(\boldsymbol{X}|\boldsymbol{Y}) = \boldsymbol{\Sigma}_{xx} - \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{yx}$$

$$\boldsymbol{\Sigma}_{xx}^{-1} = \begin{pmatrix} 10 & 0 \\ 0 & 5 \end{pmatrix}^{-1} = \frac{1}{50}\begin{pmatrix} 5 & 0 \\ 0 & 10 \end{pmatrix} = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.2 \end{pmatrix}$$

$$\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{yx} = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.2 \end{pmatrix}\begin{pmatrix} 1 & -2 \\ 3 & -2 \end{pmatrix} = \begin{pmatrix} 0.1 & -0.2 \\ 0.6 & -0.4 \end{pmatrix}$$

$$\boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{yx} = \begin{pmatrix} 1 & 3 \\ -2 & -2 \end{pmatrix}\begin{pmatrix} 0.1 & -0.2 \\ 0.6 & -0.4 \end{pmatrix} = \begin{pmatrix} 1.9 & -1.4 \\ -1.4 & 1.2 \end{pmatrix}$$

$$cov(\boldsymbol{X}|\boldsymbol{Y}) = \begin{pmatrix} 10 & 0 \\ 0 & 5 \end{pmatrix} - \begin{pmatrix} 1.9 & -1.4 \\ -1.4 & 1.2 \end{pmatrix} = \begin{pmatrix} 8.1 & 1.4 \\ 1.4 & 3.8 \end{pmatrix}$$

We can also use R:

```
A=matrix(c(10,0,1,-2,0,5,3,-2,1,3,4,1,-2,-2,1,6), 4, 4)
solve(A[1:2, 1:2])
```

```
##      [,1] [,2]
## [1,]  0.1  0.0
## [2,]  0.0  0.2
```

```
solve(A[1:2, 1:2])%*%A[1:2, 3:4]
```

```
##      [,1] [,2]
## [1,]  0.1 -0.2
## [2,]  0.6 -0.4
```

```
A[3:4, 1:2]%*%solve(A[1:2, 1:2])%*%A[1:2, 3:4]
```

```
##      [,1] [,2]
## [1,]  1.9 -1.4
## [2,] -1.4  1.2
```

```
A[1:2,1:2]-A[3:4, 1:2]%*%solve(A[1:2, 1:2])%*%A[1:2, 3:4]
```

```
##      [,1] [,2]
## [1,]  8.1  1.4
## [2,]  1.4  3.8
```

So now (in terms of $\boldsymbol{V} = (v_1, .., v_4)'$

$$\rho_{12\cdot 34} = \frac{\sigma_{12\cdot 34}}{\sqrt{\sigma_{11\cdot 34}\sigma_{22\cdot 34}}} = \frac{1.4}{\sqrt{8.1 \times 3.8}} = 0.25$$

4.3 Sums of Squares, Mean and Variance of Quadratic Forms

4.3.1 Sums of Squares

4.3.1.1 Example (5.3.1) Let $\boldsymbol{x} = (x_1, .., x_n)$ be a random sample from some population with mean μ and standard deviation σ. Then the *total sum of squares* is given by $\sum_{i=1}^{n} x_i^2$. Let $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ be the *sample mean*, then

$$
\begin{aligned}
&\sum_{i=1}^{n}(x_i - \bar{x})^2 = \\
&\sum_{i=1}^{n}\left(x_i^2 - 2x_i\bar{x} + \bar{x}^2\right) = \\
&\sum_{i=1}^{n}x_i^2 - \sum_{i=1}^{n}2x_i\bar{x} + \sum_{i=1}^{n}\bar{x}^2 = \\
&\sum_{i=1}^{n}x_i^2 - 2\bar{x}\sum_{i=1}^{n}x_i + n\bar{x}^2 = \\
&\sum_{i=1}^{n}x_i^2 - 2\bar{x}n\bar{x} + n\bar{x}^2 = \\
&\sum_{i=1}^{n}x_i^2 - n\bar{x}^2
\end{aligned}
$$

and so we find that the total sum of squares can be partitioned into a sum of squares about the mean and the sum of squares due to the mean:

$$\sum_{i=1}^{n}x_i^2 = \sum_{i=1}^{n}(x_i - \bar{x})^2 + n\bar{x}^2$$

This can also be written as a quadratic form:

$$\sum_{i=1}^{n}x_i^2 = \boldsymbol{x}'\boldsymbol{x} = \boldsymbol{x}'\boldsymbol{I}\boldsymbol{x}$$

Recall that $\boldsymbol{j} = (1, .., 1)'$ and

$$\boldsymbol{J} = \begin{pmatrix} 1 & \dots & 1 \\ \vdots & & \vdots \\ 1 & \dots & 1 \end{pmatrix}$$

then

$$\bar{x} = \frac{1}{n}\boldsymbol{j}'\boldsymbol{x}$$

and

$$n\bar{x}^2 = n(\frac{1}{n}\boldsymbol{j}'\boldsymbol{x})^2 =$$
$$\frac{1}{n}\boldsymbol{x}'\boldsymbol{j}\boldsymbol{j}'\boldsymbol{x} =$$
$$\frac{1}{n}\boldsymbol{x}'\boldsymbol{J}\boldsymbol{x} =$$
$$\boldsymbol{x}'(\frac{1}{n}\boldsymbol{J})\boldsymbol{x}$$

and so

$$\sum_{i=1}^{n} x_i^2 = \boldsymbol{x}'(\boldsymbol{I} - \frac{1}{n}\boldsymbol{J})\boldsymbol{x} + \boldsymbol{x}'(\frac{1}{n}\boldsymbol{J})\boldsymbol{x}$$

4.3.1.2 Theorem (5.3.2)

i. $\boldsymbol{I} = \left(\boldsymbol{I} - \frac{1}{n}\boldsymbol{J}\right) + \frac{1}{n}\boldsymbol{J}$

ii. $\boldsymbol{I}, \boldsymbol{I} - \frac{1}{n}\boldsymbol{J}, \frac{1}{n}\boldsymbol{J}$ are idempotent

iii. $(\boldsymbol{I} - \frac{1}{n}\boldsymbol{J})(\frac{1}{n}\boldsymbol{J}) = \boldsymbol{O}$

proof follows from direct calculation

4.3.2 Mean and Variance of Quadratic Forms

4.3.2.1 Theorem (5.3.3) If $\boldsymbol{X}$ is a random vector with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ and if $\boldsymbol{A}$ is a symmetric matrix of constants, then

$$E[\boldsymbol{X}'\boldsymbol{A}\boldsymbol{X}] = tr(\boldsymbol{A}\boldsymbol{\Sigma}) + \boldsymbol{\mu}'\boldsymbol{A}\boldsymbol{\mu}$$

proof

Note that $\boldsymbol{X}'\boldsymbol{A}\boldsymbol{X}$ is a scalar, so always $\boldsymbol{X}'\boldsymbol{A}\boldsymbol{X} = tr(\boldsymbol{X}'\boldsymbol{A}\boldsymbol{X})$

$$\boldsymbol{\Sigma} = E[\boldsymbol{X}'\boldsymbol{X}] - \boldsymbol{\mu}\boldsymbol{\mu}'$$
$$E[\boldsymbol{X}'\boldsymbol{X}] = \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}'$$
$$E[\boldsymbol{X}'\boldsymbol{A}\boldsymbol{X}] =$$
$$E[tr(\boldsymbol{X}'\boldsymbol{A}\boldsymbol{X})] = \text{ (by 4.2.11)}$$
$$E[tr(\boldsymbol{A}\boldsymbol{X}\boldsymbol{X}')] =$$
$$tr(E[\boldsymbol{A}\boldsymbol{X}\boldsymbol{X}'] =$$
$$tr(\boldsymbol{A}E[\boldsymbol{X}\boldsymbol{X}'] =$$
$$tr(\boldsymbol{A}[\boldsymbol{\Sigma} + \boldsymbol{\mu}'\boldsymbol{\mu}]) =$$
$$tr(\boldsymbol{A}\boldsymbol{\Sigma} + \boldsymbol{A}\boldsymbol{\mu}'\boldsymbol{\mu}) =$$
$$tr(\boldsymbol{A}\boldsymbol{\Sigma}) + tr(\boldsymbol{A}\boldsymbol{\mu}\boldsymbol{\mu}') =$$
$$tr(\boldsymbol{A}\boldsymbol{\Sigma}) + tr(\boldsymbol{\mu}'\boldsymbol{A}\boldsymbol{\mu}) =$$
$$tr(\boldsymbol{A}\boldsymbol{\Sigma}) + \boldsymbol{\mu}'\boldsymbol{A}\boldsymbol{\mu}$$

4.3.2.2 Example (5.3.4) Let $X \sim N(\mu, \sigma^2); \boldsymbol{A} = (a), a > 0$, then $\boldsymbol{\Sigma} = (\sigma^2)$

$$E[x'Ax] = E[aX^2] = aE[X^2] =$$
$$a(var(X) + E[X]^2) = a(\sigma^2 + \mu^2) =$$
$$a\sigma^2 + a\mu^2 =$$
$$tr(\boldsymbol{A}\boldsymbol{\Sigma}) + \boldsymbol{\mu}'\boldsymbol{A}\boldsymbol{\mu}$$

4.3.2.3 Example (5.3.5) Say $\boldsymbol{X} = \begin{pmatrix} X \\ Y \end{pmatrix} \sim N(\begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 & 1 \\ 1 & 2 \end{pmatrix})$ and we want to find $E[X^2 + 2XY + 2Y^2]$.

Direct solution:

$$E[X^2 + 2XY + 2Y^2] =$$
$$E[X^2] + 2E[XY] + 2E[Y^2] =$$
$$var(X) + E[X]^2 + 2(cov(X, Y) + E[X]E[Y]) + 2(var(Y) + E[Y]^2) =$$
$$3 + 1^2 + 2(1 + 1 * 2) + 2(2 + 2^2) = 4 + 6 + 12 = 22$$

or using the formula above: say $\boldsymbol{A} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}$, then

$$\boldsymbol{X}'\boldsymbol{A}\boldsymbol{X} = \begin{pmatrix} X & Y \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} X \\ Y \end{pmatrix} =$$

$$\begin{pmatrix} X & Y \end{pmatrix} \begin{pmatrix} X+Y \\ X+2Y \end{pmatrix} =$$

$$X(X+Y) + Y(X+2Y) = X^2 + 2XY + 2Y^2$$

$$\boldsymbol{A}\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 3 & 1 \\ 1 & 2 \end{pmatrix} = \begin{pmatrix} 4 & 3 \\ 5 & 5 \end{pmatrix}$$

$$tr(\boldsymbol{A}\boldsymbol{\Sigma}) = 4 + 5 = 9$$

$$\boldsymbol{\mu}'\boldsymbol{A}\boldsymbol{\mu} = \begin{pmatrix} 1 & 2 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 1 & 2 \end{pmatrix} \begin{pmatrix} 3 \\ 5 \end{pmatrix} = 13$$

$$E[\boldsymbol{X}'\boldsymbol{A}\boldsymbol{X}] = tr(\boldsymbol{A}\boldsymbol{\Sigma}) + \boldsymbol{\mu}'\boldsymbol{A}\boldsymbol{\mu} = 9 + 13 = 22$$

4.3.2.4 Definition (5.3.6) Let $\boldsymbol{x} = (x_1, .., x_n)$ be a sample from a random vector $\boldsymbol{X}$. The *sample variance* is defined by

$$s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2$$

By (5.3.1) we have

$$(n-1)s^2 = \boldsymbol{x}'(\boldsymbol{I} - \frac{1}{n}\boldsymbol{J})\boldsymbol{x}$$

Say $E[X_1] = \mu$ and $var(X_1) = \sigma^2$. In a random sample the X_i's are independent and identically distributed, so $E[\boldsymbol{X}] = \mu\boldsymbol{j}$ and $cov(\boldsymbol{X}) = \sigma^2\boldsymbol{I}$. Set

$$\boldsymbol{A} = \boldsymbol{I} - \frac{1}{n}\boldsymbol{J}, \boldsymbol{\Sigma} = \sigma^2\boldsymbol{I}$$

and $\boldsymbol{\mu} = \mu\boldsymbol{j}$, therefore

$$E[\sum_{i=1}^{n}(X_i - \bar{X})^2] =$$

$$E[(n-1)s^2] =$$

$$E[\boldsymbol{X}'(\boldsymbol{I} - \frac{1}{n}\boldsymbol{J})\boldsymbol{X}] = \text{(by 5.3.2)}$$

$$tr\left[(\boldsymbol{I} - \frac{1}{n}\boldsymbol{J})\sigma^2\boldsymbol{I}\right] + \mu\boldsymbol{j}'(\boldsymbol{I} - \frac{1}{n}\boldsymbol{J})\mu\boldsymbol{j} =$$

$$\sigma^2 tr\left[\boldsymbol{I} - \frac{1}{n}\boldsymbol{J}\right] + \mu^2\left(\boldsymbol{j}'\boldsymbol{j} - \boldsymbol{j}'\boldsymbol{j}\frac{1}{n}\boldsymbol{j}'\boldsymbol{j}\right) =$$

$$\sigma^2(n - \frac{n}{n}) + \mu^2(n - \frac{1}{n}n^2) =$$

$$\sigma^2(n-1)$$

and so

$$E[s^2] = \frac{1}{n-1} E[\sum_{i=1}^{n} (x_i - \bar{x})^2] = \sigma^2$$

and we see that s^2 is an unbiased estimator of σ^2.

4.3.2.5 Theorem (5.3.7) Let $\boldsymbol{X} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then the moment generating function of $\boldsymbol{X'AX}$ is given by

$$\psi(t) = |\boldsymbol{I} - 2t\boldsymbol{A\Sigma}|^{-1/2} \exp\left\{-\boldsymbol{\mu}' \left[\boldsymbol{I} - (\boldsymbol{I} - 2t\boldsymbol{A\Sigma})^{-1}\right] \boldsymbol{\Sigma}^{-1}\right] \boldsymbol{\mu}/2\}$$

proof

$$\psi(t) = c_1 \int .. \int \exp\{t\boldsymbol{x}'\boldsymbol{A}\boldsymbol{x}\} \exp\{-(\boldsymbol{x} - \boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})/2\} d\boldsymbol{x} =$$
$$\psi(\boldsymbol{t}) = c_1 \int .. \int \exp\{-[(\boldsymbol{x} - \boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}) - t\boldsymbol{x}'\boldsymbol{A}\boldsymbol{x}]/2 d\boldsymbol{x} =$$
$$c_1 \int .. \int \exp\{-[\boldsymbol{x}'(\boldsymbol{I} - 2t\boldsymbol{A\Sigma})\boldsymbol{\Sigma}^{-1}\boldsymbol{x} - 2\boldsymbol{\mu}'\boldsymbol{\Sigma}^{-1}\boldsymbol{x} + \boldsymbol{\mu}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}]/2\} d\boldsymbol{x}$$

where

$$c_1 = 1/\left[(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{-1/2}\right]$$

Now if t is close to 0 $\boldsymbol{I} - 2t\boldsymbol{A\Sigma}$ is nonsingular. Let $\boldsymbol{\theta}' = \boldsymbol{\mu}'(\boldsymbol{I} - 2t\boldsymbol{A\Sigma})^{-1}$ and $\boldsymbol{V}^{-1} = (\boldsymbol{I} - 2t\boldsymbol{A\Sigma})\boldsymbol{\Sigma}^{-1}$, then we have

$$\psi(t) = c_1 c_2 \int .. \int c_3 \exp\{-(\boldsymbol{x} - \boldsymbol{\theta})'\boldsymbol{V}^{-1}(\boldsymbol{x} - \boldsymbol{\theta})/2\} d\boldsymbol{x}$$

where

$$c_2 = 1/\left[(2\pi)^{p/2}|\boldsymbol{V}|^{-1/2}\right] \exp\{-(\boldsymbol{\mu}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} - \boldsymbol{\theta}'\boldsymbol{V}^{-1}\boldsymbol{\theta})/2\}$$

and

$$c_3 = 1/\left[(2\pi)^{p/2}|\boldsymbol{V}|^{-1/2}\right]$$

The integral is one because it is integrating out a multivariate normal, and therefore

$$\psi(\boldsymbol{t}) = c_1 c_2 = 1/\left[(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{-1/2}\right] 1/\left[(2\pi)^{p/2}|\boldsymbol{V}|^{-1/2}\right] \exp\{-(\boldsymbol{\mu}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} - \boldsymbol{\theta}'\boldsymbol{V}^{-1}\boldsymbol{\theta})/2$$

and replacing the terms yields the result.

4.3.2.6 Theorem (5.3.8) Say $\boldsymbol{X} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then

$$var(\boldsymbol{X}'\boldsymbol{A}\boldsymbol{X}) = 2tr[(\boldsymbol{A}\boldsymbol{\Sigma})^2] + 4\boldsymbol{\mu}'\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}\boldsymbol{\mu}$$

proof

From probability theory we know that for any rv Y with mgf ψ we have

$$E[Y^k] = \frac{d^k\psi(t)}{dt^k}|_{t=0}$$

Therefore we have

$$\begin{aligned}
&\frac{d\log\psi(t)}{dt} = \frac{\psi'(t)}{\psi(t)} \\
&\frac{d^2\log\psi(t)}{dt^2} = \frac{\psi''(t)\psi_Y(t) - (\psi'(t))^2}{(\psi(t))^2} \\
&\frac{d^2\log\psi(t)}{dt^2}|_{t=0} = \frac{E[Y^2]E[Y^0] - (E[Y])^2}{(E[Y^0])^2} = \\
&E[Y^2] - (E[Y])^2 = var(Y)
\end{aligned}$$

By (5.3.7) we have

$$m(t) = \log\psi_{\boldsymbol{X}'\boldsymbol{A}\boldsymbol{X}}(t) = -\frac{1}{2}\log|\boldsymbol{C}| - \frac{1}{2}\boldsymbol{\mu}'(\boldsymbol{I} - \boldsymbol{C}^{-1})\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$$

where $\boldsymbol{C} = \boldsymbol{I} - 2t\boldsymbol{A}\boldsymbol{\Sigma}$.

For the first term we find

$$\begin{aligned}
&\frac{d^2\log|C|}{dt^2} = \\
&\frac{d}{dt}\frac{d|C|/dt}{|C|} = \\
&\frac{d^2|C|/dt^2|C| - (d|C|/dt)^2}{|C|^2} = \\
&\frac{1}{|\boldsymbol{C}|}\left[\frac{d^2|\boldsymbol{C}|}{dt^2}\right] - \frac{1}{|\boldsymbol{C}|^2}\left[\frac{d|\boldsymbol{C}|}{dt}\right]^2
\end{aligned}$$

For the second term we have

Now $\frac{\partial\boldsymbol{C}}{\partial t} = -2\boldsymbol{A}\boldsymbol{\Sigma}$. By (4.3.28) we have

$$\frac{\partial\boldsymbol{A}^{-1}}{\partial x} = -\boldsymbol{A}^{-1}\frac{\partial\boldsymbol{A}}{\partial x}\boldsymbol{A}^{-1}$$

and so

$$\frac{\partial \boldsymbol{C}^{-1}}{\partial t} = -\boldsymbol{C}^{-1}\frac{\partial \boldsymbol{C}}{\partial t}\boldsymbol{C}^{-1} = 2\boldsymbol{C}^{-1}\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{C}^{-1}$$

For the second derivative we find

$$\begin{aligned}
&\frac{\partial^2 \boldsymbol{C}^{-1}}{\partial t^2} = \\
&\frac{d}{dt}\left\{2\boldsymbol{C}^{-1}\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{C}^{-1}\right\} = \\
&2\frac{d}{dt}\left\{\boldsymbol{C}^{-1}\right\}\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{C}^{-1} + 2\boldsymbol{C}^{-1}\boldsymbol{A}\boldsymbol{\Sigma}\left\{\boldsymbol{C}^{-1}\right\} = \\
&4\boldsymbol{C}^{-1}\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{C}^{-1}\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{C}^{-1} + 4\boldsymbol{C}^{-1}\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{C}^{-1}\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{C}^{-1} = \\
&\boldsymbol{C}^{-1}\left\{\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{C}^{-1}\boldsymbol{A}\boldsymbol{\Sigma} + \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{C}^{-1}\boldsymbol{A}\boldsymbol{\Sigma}\right\}\boldsymbol{C}^{-1} = \\
&8\boldsymbol{C}^{-1}\left\{\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{C}^{-1}\boldsymbol{A}\boldsymbol{\Sigma}\right\}\boldsymbol{C}^{-1}
\end{aligned}$$

so now we have

$$\begin{aligned}
&m''(t) = \frac{d^2}{dt^2}\left\{-\frac{1}{2}\ln|\boldsymbol{C}| - \frac{1}{2}\boldsymbol{\mu}'(\boldsymbol{I} - \boldsymbol{C}^{-1})\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right\} = \\
&-\frac{1}{2}\frac{1}{|\boldsymbol{C}|}\left[\frac{d^2|\boldsymbol{C}|}{dt^2}\right] + \frac{1}{2}\frac{1}{|\boldsymbol{C}|^2}\left[\frac{d|\boldsymbol{C}|}{dt}\right]^2 - \frac{1}{2}\boldsymbol{\mu}'\left(-8\boldsymbol{C}^{-1}\left\{\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{C}^{-1}\boldsymbol{A}\boldsymbol{\Sigma}\right\}\boldsymbol{C}^{-1}\right)\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} = \\
&-\frac{1}{2}\frac{1}{|\boldsymbol{C}|}\left[\frac{d^2|\boldsymbol{C}|}{dt^2}\right] + \frac{1}{2}\frac{1}{|\boldsymbol{C}|^2}\left[\frac{d|\boldsymbol{C}|}{dt}\right]^2 + 4\boldsymbol{\mu}'\boldsymbol{C}^{-1}\left\{\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{C}^{-1}\boldsymbol{A}\boldsymbol{\Sigma}\right\}\boldsymbol{C}^{-1}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}
\end{aligned}$$

If t=0 we have $\boldsymbol{C} = \boldsymbol{I}$ and the second term becomes $4\boldsymbol{\mu}'\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}\boldsymbol{\mu}$, as required.

For the first term recall that the determinant of a matrix is equal to the product of the eigenvalues, so if the eigenvalues of $\boldsymbol{A}\boldsymbol{\Sigma}$ are $\lambda_1, .., \lambda_p$ we have

$$\begin{aligned}
&|\boldsymbol{C}| = \prod_{i=1}^{p}(1 - t2\lambda_i); |\boldsymbol{C}|_{t=0} = 1 \\
&1 - 2t\sum_{i=1}^{p}\lambda_i + 4t^2\sum_{i\neq j}^{p}\lambda_i\lambda_j - +...(-1)^p 2^p t^p\prod_{i=1}^{p}\lambda_i \\
&\frac{d|\boldsymbol{C}|}{dt} = -2\sum_{i=1}^{p}\lambda_i + 8t\sum_{i\neq j}^{p}\lambda_i\lambda_j + \text{ higher order terms in t} \\
&\frac{d|\boldsymbol{C}|}{dt}|_{t=0} = -2\sum_{i=1}^{p}\lambda_i = -2tr(\boldsymbol{A}\boldsymbol{\Sigma}) \\
&\frac{d^2|\boldsymbol{C}|}{dt^2} = 8\sum_{i\neq j}^{p}\lambda_i\lambda_j + \text{ higher order terms in t} \\
&\frac{d^2|\boldsymbol{C}|}{dt^2}|_{t=0} = 8\sum_{i\neq j}^{p}\lambda_i\lambda_j
\end{aligned}$$

so the first term at t=0 is

$$-\frac{1}{2}\frac{1}{|\boldsymbol{C}|}\left[\frac{d^2|\boldsymbol{C}|}{dt^2}|_{t=0}\right]+\frac{1}{2}\frac{1}{|\boldsymbol{C}|^2}\left[\frac{d|\boldsymbol{C}|}{dt}|_{t=0}\right]^2=$$
$$-\frac{1}{2}8\sum_{i\neq j}^{p}\lambda_i\lambda_j+\frac{1}{2}[-2tr(\boldsymbol{A\Sigma})]^2=$$
$$2[tr(\boldsymbol{A\Sigma})]^2-4\sum_{i\neq j}^{p}\lambda_i\lambda_j$$

and it can be shown that

$$[tr(\boldsymbol{A\Sigma})]^2-4\sum_{i\neq j}^{p}\lambda_i\lambda_j=tr([\boldsymbol{A\Sigma}]^2)$$

4.3.2.7 Example (5.3.9) Let $X\sim N(\mu,\sigma^2);\boldsymbol{A}=(a),a>0$, then $\boldsymbol{\Sigma}=(\sigma^2)$. Let $Z\sim N(0,1)$ and recall that $Z^2\sim\chi^2(1)$, and therefore E[Z]=0, E[Z^2]=1, E[Z^3]=0 and E[Z^4]=3 (=2+1=var + mean^2 of a $\chi^2(1)$). So

$$var(x'Ax)=var(aX^2)=a^2var(X^2)=a^2(E[X^4]-E[X^2]^2)$$
$$E[X^2]=var(X)+E[X]^2=\sigma^2+\mu^2$$
$$E[X^4]=E[(\sigma Z+\mu)^4]=$$
$$\sigma^4E[Z^4]+4\sigma^3E[Z^3]\mu+6\sigma^2E[Z^2]\mu^2+4\sigma E[Z]\mu^3+\mu^4=$$
$$\sigma^4\times3+4\sigma^3\times0\mu+6\sigma^2\times1\mu^2+4\sigma\times0\mu^3+\mu^4=$$
$$3\sigma^4+6\sigma^2\mu^2+\mu^4$$
$$var(x'Ax)=a^2\left(3\sigma^4+6\sigma^2\mu^2+\mu^4-(\sigma^2+\mu^2)^2\right)=$$
$$a^2\left(3\sigma^4+6\sigma^2\mu^2+\mu^4-\sigma^4-2\sigma^2\mu^2-\mu^4\right)=$$
$$a^2\left(2\sigma^4+4\sigma^2\mu^2\right)=2a^2\sigma^2\left(\sigma^2+2\mu^2\right)$$

but also

$$2tr((\boldsymbol{A\Sigma})^2)+4\boldsymbol{\mu}'\boldsymbol{A\Sigma A\mu}=$$
$$2a^2\sigma^4+4\mu a\sigma^2a\mu=2a^2\sigma^2(\sigma^2+2\mu^2)$$

4.3.2.8 Example (5.3.10) Say $\boldsymbol{X}=\begin{pmatrix}X\\Y\end{pmatrix}\sim N(\begin{pmatrix}1\\2\end{pmatrix},\begin{pmatrix}3&1\\1&2\end{pmatrix})$ and we want to find $var(X^2+2XY+2Y^2)$.

Let's try this directly:

$$
\begin{aligned}
&var(X^2 + 2XY + 2Y^2) = \\
&var(X^2) + var(2XY) + var(Y^2) + \\
&2\left[cov(X^2 2XY) + cov(X^2 2Y^2) + cov(2XY 2Y^2)\right] = \\
&var(X^2) + 4var(XY) + var(Y^2) + \\
&2\left[2cov(X^3 Y) + 2cov(X^2 Y^2) + 4cov(XY^3)\right]
\end{aligned}
$$

and many of these terms are not easy to calculate. For example, to find $var(XY)$ we would need the distribution of the product of two correlated normal random variables. But we can use the formula above:

recall that $\boldsymbol{A} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}$ and $\boldsymbol{A\Sigma} = \begin{pmatrix} 4 & 3 \\ 5 & 5 \end{pmatrix}$
, so

$$
\begin{aligned}
&(\boldsymbol{A\Sigma})^2 = \begin{pmatrix} 4 & 3 \\ 5 & 5 \end{pmatrix}\begin{pmatrix} 4 & 3 \\ 5 & 5 \end{pmatrix} = \begin{pmatrix} 31 & 27 \\ 45 & 40 \end{pmatrix} \\
&tr((\boldsymbol{A\Sigma})^2) = 31 + 40 = 71
\end{aligned}
$$

$$
\begin{aligned}
&\boldsymbol{\mu' A\Sigma A\mu} = \\
&\begin{pmatrix} 1 & 2 \end{pmatrix}\begin{pmatrix} 4 & 3 \\ 5 & 5 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}\begin{pmatrix} 1 \\ 2 \end{pmatrix} = \\
&\begin{pmatrix} 1 & 2 \end{pmatrix}\begin{pmatrix} 7 & 10 \\ 10 & 15 \end{pmatrix}\begin{pmatrix} 1 \\ 2 \end{pmatrix} = 107 \\
&var(\boldsymbol{X'AX}) = 2tr[(\boldsymbol{A\Sigma})^2] + 4\boldsymbol{\mu' A\Sigma A\mu} = \\
&2 \times 71 + 4 \times 107 = 570
\end{aligned}
$$

let's check this with R:

```
library(mvtnorm)
x=rmvnorm(2e5, c(1, 2), matrix(c(3, 1, 1, 2), 2, 2))
var(x[,1]^2+2*x[, 1]*x[, 2]+2*x[, 2]^2)
```

```
## [1] 570.5925
```

4.3.2.9 Theorem (5.3.11)

Say $\boldsymbol{X} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then

$$cov(\boldsymbol{X}, \boldsymbol{X'AX}) = 2\boldsymbol{\Sigma A\mu}$$

proof

by definition

$cov(\boldsymbol{X}, \boldsymbol{X}'\boldsymbol{A}\boldsymbol{X}) =$
$E[(\boldsymbol{X} - E[\boldsymbol{X}])(\boldsymbol{X}'\boldsymbol{A}\boldsymbol{X} - E[\boldsymbol{X}'\boldsymbol{A}\boldsymbol{X}])] =$
$E[(\boldsymbol{X} - \boldsymbol{\mu})(\boldsymbol{X}'\boldsymbol{A}\boldsymbol{X} - tr(\boldsymbol{A}\boldsymbol{\Sigma}) - \boldsymbol{\mu}'\boldsymbol{A}\boldsymbol{\mu})] =$
$E[(\boldsymbol{X} - \boldsymbol{\mu})\left((\boldsymbol{X} - \boldsymbol{\mu})'\boldsymbol{A}(\boldsymbol{X} - \boldsymbol{\mu}) - tr(\boldsymbol{A}\boldsymbol{\Sigma}) + 2(\boldsymbol{X} - \boldsymbol{\mu})'\boldsymbol{A}\boldsymbol{\mu}\right)] =$
$E[(\boldsymbol{X} - \boldsymbol{\mu})(\boldsymbol{X} - \boldsymbol{\mu})'\boldsymbol{A}(\boldsymbol{X} - \boldsymbol{\mu})] - E[(\boldsymbol{X} - \boldsymbol{\mu})tr(\boldsymbol{A}\boldsymbol{\Sigma}) + 2E[(\boldsymbol{X} - \boldsymbol{\mu})(\boldsymbol{X} - \boldsymbol{\mu})'\boldsymbol{A}\boldsymbol{\mu}] =$
$E[(\boldsymbol{X} - \boldsymbol{\mu})(\boldsymbol{X} - \boldsymbol{\mu})'\boldsymbol{A}(\boldsymbol{X} - \boldsymbol{\mu})] - E[\boldsymbol{X} - \boldsymbol{\mu}]tr(\boldsymbol{A}\boldsymbol{\Sigma}) + 2E[(\boldsymbol{X} - \boldsymbol{\mu})(\boldsymbol{X} - \boldsymbol{\mu})']\boldsymbol{A}\boldsymbol{\mu} =$
$\boldsymbol{0} - \boldsymbol{0} + 2\boldsymbol{\Sigma}\boldsymbol{A}\boldsymbol{\mu}$

the first term is 0 because it is the third central moment of a multivariate normal.

4.3.2.10 Corollary (5.3.12) Say $\boldsymbol{X} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, and let $\boldsymbol{B}$ be a $n \times p$ matrix of constants, then

$$cov(\boldsymbol{B}\boldsymbol{X}, \boldsymbol{X}'\boldsymbol{A}\boldsymbol{X}) = 2\boldsymbol{B}\boldsymbol{\Sigma}\boldsymbol{A}\boldsymbol{\mu}$$

4.3.2.11 Theorem (5.3.13) Let $\boldsymbol{V} = \begin{pmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \end{pmatrix}$ be a partitioned in the usual way, with $\boldsymbol{\Sigma}_{xy}$ a $p \times q$ matrix. Let $\boldsymbol{A}$ be a $q \times p$ matrix of constants, then
$E[\boldsymbol{x}'\boldsymbol{A}\boldsymbol{y}] = tr(\boldsymbol{A}\boldsymbol{\Sigma}_{xy}) + \boldsymbol{\mu}_y'\boldsymbol{A}\boldsymbol{\mu}_x$
proof similar to proof earlier

4.3.2.12 Definition (5.3.14) We have a random sample $\begin{pmatrix} X \\ Y \end{pmatrix} = \left(\begin{pmatrix} x_1 \\ y_1 \end{pmatrix}, .., \begin{pmatrix} x_n \\ y_n \end{pmatrix}\right)$ from some bivariate distribution with means μ_x and μ_y, variances σ_x^2 and σ_y^2. An estimator of the population covariance $\boldsymbol{\sigma}_{ij}$ is the *sample covariance* defined by

$$s_{xy} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

4.3.2.13 Example (5.3.15) We can write

$s_{xy} =$
$\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) =$
$\frac{1}{n-1}\sum_{i=1}^{n}x_i y_i - n\bar{x}\bar{y} =$
$\frac{1}{n-1}\boldsymbol{x}'[\boldsymbol{I} - (1/n)\boldsymbol{J}]\boldsymbol{y} =$

(x_i, y_i) is independent of (x_j, y_j) if $i \neq j$, we can define a random vector $\boldsymbol{V} = \begin{pmatrix} X \\ Y \end{pmatrix}$ with $E[\boldsymbol{V}] = \begin{pmatrix} \mu_x \boldsymbol{j} \\ \mu_y \boldsymbol{j} \end{pmatrix}$ and covariance matrix

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_x^2 \boldsymbol{I} & \sigma_{xy} \boldsymbol{I} \\ \sigma_{xy} \boldsymbol{I} & \sigma_y^2 \boldsymbol{I} \end{pmatrix}$$

Let $\boldsymbol{A} = \boldsymbol{I} - (1/n)\boldsymbol{J}$ and so

$$\begin{aligned}
&E[\boldsymbol{x}'[\boldsymbol{I} - (1/n)\boldsymbol{J}]\boldsymbol{y}] = \\
&tr\left[(\boldsymbol{I} - (1/n)\boldsymbol{J})\sigma_{xy}\boldsymbol{I}\right] + \mu_y \boldsymbol{j}'(\boldsymbol{I} - (1/n)\boldsymbol{J})\mu_x \boldsymbol{j} = \\
&\sigma_{xy} tr[\boldsymbol{I} - (1/n)\boldsymbol{J}] + \mu_x \mu_y (\boldsymbol{j}'\boldsymbol{j} - (1/n)\boldsymbol{j}'\boldsymbol{j}\boldsymbol{j}'\boldsymbol{j}) = \\
&\sigma_{xy}(n-1) + 0 = (n-1)\sigma_{xy}
\end{aligned}$$

4.4 Noncentral Chi-Square, F and t Distributions

4.4.1 Non-Central Chisquare Distribution

4.4.1.1 Definition (5.4.1) A random variable X is said to have a *chi-square distribution* with n degrees of freedom if it has density

$$\frac{1}{\Gamma(n/2)2^{n/2}} x^{n/2-1} e^{-x/2}$$

Let $Z \sim N(0,1)$ and let $X = Z^2$, then

$$\begin{aligned}
&F_X(x) = P(X < x) = P(Z^2 < x) = \\
&P(-\sqrt{x} < Z < \sqrt{x}) = \\
&\int_{-\sqrt{x}}^{\sqrt{x}} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt \\
&f_X(x) = \frac{dF_x(x)}{dx} = \frac{d}{dx} \int_{-\sqrt{x}}^{\sqrt{x}} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt = \\
&\frac{1}{\sqrt{2\pi}} e^{-(\sqrt{x})^2/2} \frac{1}{2\sqrt{x}} - \frac{1}{\sqrt{2\pi}} e^{-(-\sqrt{x})^2/2} \frac{-1}{2\sqrt{x}} = \\
&\frac{1}{\sqrt{2\pi}} \frac{1}{\sqrt{x}} e^{-x/2} = \\
&\frac{1}{\Gamma(1/2)2^{1/2}} x^{1/2-1} e^{-x/2}
\end{aligned}$$

and so $X \sim \chi^2(1)$.

4.4.1.2 Theorem (5.4.2) Say X has a chi-square distribution with n df, then

i. $E[X] = n$

ii. $var(X) = 2n$

iii. $\psi(t) = (1-2t)^{-n/2}; |t| < 1/2$

proof

$$E[X^k] = \int_0^\infty x^k \frac{1}{\Gamma(n/2)2^{n/2}} x^{n/2-1} e^{-x/2} dx =$$
$$\frac{1}{\Gamma(n/2)2^{n/2}} \int_0^\infty x^{k+n/2-1} e^{-x/2} dx =$$
$$\frac{\Gamma((2k+n)/2)2^{(2k+n)/2}}{\Gamma(n/2)2^{n/2}} \int_0^\infty \frac{1}{\Gamma((2k+n)/2)2^{(2k+n)/2}} x^{(2k+n)/2-1} e^{-x/2} dx =$$
$$\frac{\Gamma(k+n/2)2^{k+n/2}}{\Gamma(n/2)2^{n/2}} =$$
$$\frac{(k+n/2-1)(k+n/2-2)..n/2\Gamma(n/2)2^k}{\Gamma(n/2)} =$$
$$(k+n/2-1)(k+n/2-2)..(n/2)2^k$$
$$E[X] = n/2 \times 2 = n$$
$$var(X) = E[X^2] - E[X]^2 =$$
$$(n/2+1)(n/2)2^2 - n^2 = n^2 + 2n - n^2 = 2n$$

iii follows similarly

Let X, Y have independent chi-square distributions with n and m degrees of freedom, respectively, then

$$\psi_{X+Y}(t) = \psi_X(t)\psi_Y(t) =$$
$$(1-2t)^{-n/2}(1-2t)^{-m/2} = (1-2t)^{-(n+m)/2}$$

and so $X+Y \sim \chi^2(n+m)$. Therefore if $\boldsymbol{Z} = (Z_1, .., Z_n)'$, $Z_i \sim N(0,1)$ and independent, then

$$\boldsymbol{Z'Z} \sim \chi^2(n)$$

Now say $X_i \sim N(\mu_i, 1)$, independent, and let $\boldsymbol{X} = (X_1, .., X_n)'$ and $\boldsymbol{\mu} = (\mu_1, .., \mu_n)'$. Therefore

$$\boldsymbol{(X-\mu)'(X-\mu)} \sim \chi^2(n)$$

Let $Y \sim N(\mu, 1)$ and let $X = Y^2$, then

$$
\begin{aligned}
&F_X(x) = P(X < x) = P(Y^2 < x) = \\
&P(-\sqrt{x} < Z < \sqrt{x}) = \\
&\int_{-\sqrt{x}}^{\sqrt{x}} \frac{1}{\sqrt{2\pi}} e^{-(t-\mu)^2/2} dt \\
&f_X(x) = \frac{dF_x(x)}{dx} = \frac{d}{dx} \int_{-\sqrt{x}}^{\sqrt{x}} \frac{1}{\sqrt{2\pi}} e^{-(t-\mu)^2/2} dt = \\
&\frac{1}{\sqrt{2\pi}} e^{-(\sqrt{x}-\mu)^2/2} \frac{1}{2\sqrt{x}} - \frac{1}{\sqrt{2\pi}} e^{-(-\sqrt{x}-\mu)^2/2} \frac{-1}{2\sqrt{x}} = \\
&\frac{1}{\sqrt{2\pi}} \frac{1}{\sqrt{x}} \left[e^{-(\sqrt{x}-\mu)^2/2} + e^{-(-\sqrt{x}-\mu)^2/2} \right] = \\
&\frac{1}{\sqrt{2\pi}} \frac{1}{\sqrt{x}} \left[e^{-(x-2\sqrt{x}\mu+\mu^2)/2} + e^{-(x+2\sqrt{x}\mu+\mu^2)/2} \right] = \\
&\frac{1}{\Gamma(1/2) 2^{1/2}} x^{1/2-1} e^{-x/2} \left[e^{\sqrt{x}\mu} + e^{-\sqrt{x}\mu} \right] e^{-\mu^2/2} = \\
&g(x, 1) \left[e^{\sqrt{x}\mu} + e^{-\sqrt{x}\mu} \right] e^{-\mu^2/2}
\end{aligned}
$$

where g is the density of a chi-square distribution with 1 degree of freedom.

More generally we have

4.4.1.3 Definition (5.4.3) Let $X_i \sim N(\mu_i, 1)$,i=1,..,n and independent, then the distribution of $\boldsymbol{X'X}$ is called a *non-central chi-square* distribution with n degrees of freedom and non-centrality parameter λ where

$$\lambda = \frac{1}{2} \sum_{i=1}^{n} \mu_i^2 = \boldsymbol{\mu'\mu}/2$$

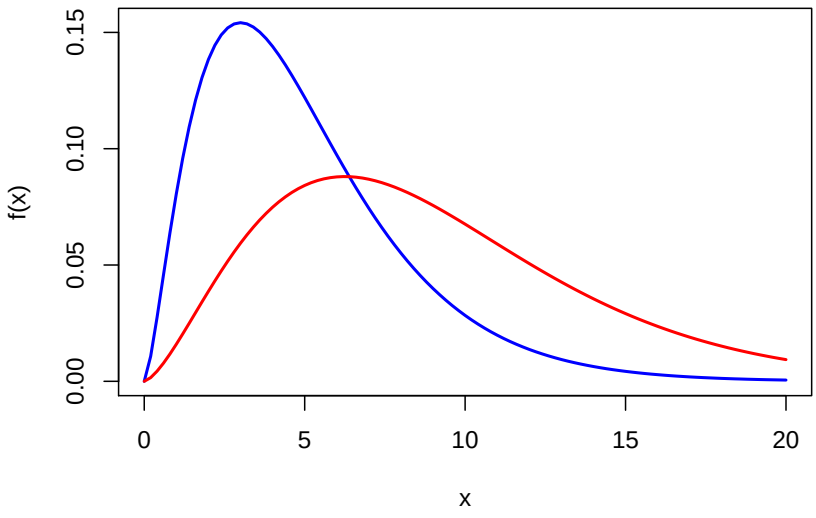
We write $\boldsymbol{X'X} \sim \chi^2(n, \lambda)$

Note

$$E[\sum_{i=1}^{n}(X_i - \mu_i)^2] =$$
$$\sum_{i=1}^{n} E[(X_i - \mu_i)^2] =$$
$$\sum_{i=1}^{n} var(X_i) = n$$
$$E[\sum_{i=1}^{n} X_i^2] =$$
$$\sum_{i=1}^{n} E[X_i^2] =$$
$$\sum_{i=1}^{n}(var(X_i) + E[X_i]^2) =$$
$$\sum_{i=1}^{n}(1 + \mu_i^2) =$$
$$n + \sum_{i=1}^{n} \mu_i^2 = n + 2\lambda$$

We can use R to calculate values for the non-central chisquare:

```
f=function(x) dchisq(x, 5)
f1=function(x) dchisq(x, 5, ncp = 4)
curve(f, 0, 20, lwd=2, col="blue")
curve(f1, 0, 20, lwd=2, col="red", add=TRUE)
```



4.4.1.4 Theorem (5.4.4)

Say $X \sim \chi^2(n, \lambda)$, then

i. $E[X] = n + 2\lambda$

ii. $E[X] = 2n + 8\lambda$

iii. $\psi_X(t) = (1-2t)^{-n/2} e^{-\lambda[1-1/(1-2t)]}$

proof i and ii follow from the calculation above, iii uses the theorem from the last section.

4.4.1.5 Theorem (5.4.5)

If $X_1 \sim \chi^2(n_1, \lambda_1), .., X_k \sim \chi^2(n_k, \lambda_k)$ and independent, then $\sum_{i=1}^k X_i \sim \chi^2(\sum_{i=1}^k n_i, \sum_{i=1}^k \lambda_i)$

proof similar to the corresponding theorem for chi-square distributions above.

4.4.2 Non-Central F Distribution

4.4.2.1 Definition (5.4.6)

Say $X \sim \chi^2(n), Y \sim \chi^2(m)$, independent, then the random variable $F = \frac{X/n}{Y/m}$ is said to have an F distribution with n and m degrees of freedom. It has density

$$f(x; n, m) = \frac{\Gamma((n+m)/2)}{\Gamma(n/2)\Gamma(m/2)} (\frac{n}{m})^{n/2} \frac{x^{n/2-1}}{(1+nx/m)^{(n+m)/2}}$$

4.4.2.2 Theorem (5.4.7)

Say $X \sim F(n, m)$, then

i. $E[X] = \frac{m}{m-2}$

ii. $var(X) = \frac{2m^2(n+m-2)}{n(m-1)^2(m-4)}$

proof omitted

4.4.2.3 Definition (5.4.8)

Say $X \sim \chi^2(n, \lambda), Y \sim \chi^2(m)$, independent, then the random variable $F = \frac{X/n}{Y/m}$ is said to have an *non-central F distribution* with n and m degrees of freedom and non-centrality parameter λ.

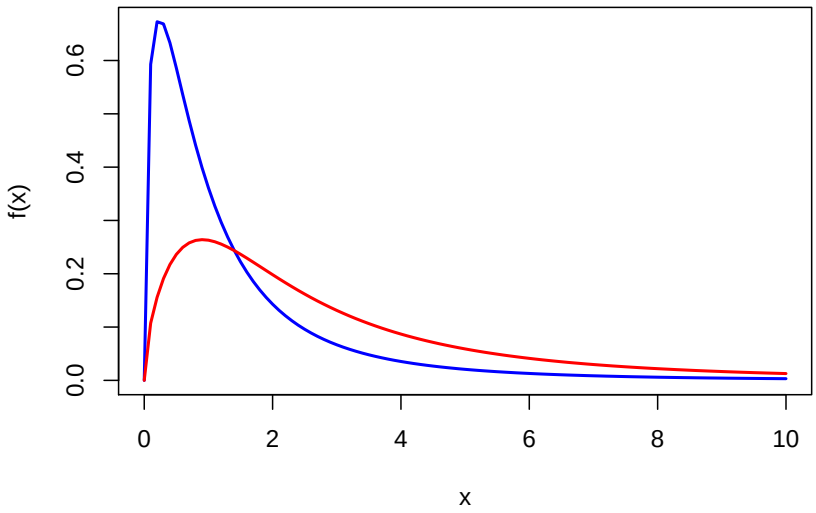
4.4.2.4 Theorem (5.4.9)

Say $X \sim F(n, m, \lambda)$, then $E[X] = \frac{m}{m-2}(1 + 2\lambda/n)$

proof omitted

We can use R to calculate values for the non-central F distribution:

```
f=function(x) df(x, 3, 5)
f1=function(x) df(x, 3, 5, ncp = 4)
curve(f, 0, 10, lwd=2, col="blue")
curve(f1, 0, 10, lwd=2, col="red", add=TRUE)
```



4.4.3 Non-Central t Distribution

4.4.3.1 Definition (5.4.10) Say $Z \sim N(0,1), Y \sim \chi^2(n)$, independent, then the random variable $T = \frac{Z}{\sqrt{Y/m}}$ is said to have a *Student's t distribution* with n degrees of freedom. It has density

$$f(x;n) = \frac{\Gamma((n+1)/2)}{\Gamma(n/2)} \frac{1}{\sqrt{\pi n}} \frac{1}{(1+x^2/n)^{(n+1)/2}}$$

4.4.3.2 Theorem (5.4.11) Say $X \sim t(n)$, then

i. $E[X] = 0$

ii. $var(X) = \frac{n}{n-2}$

proof omitted

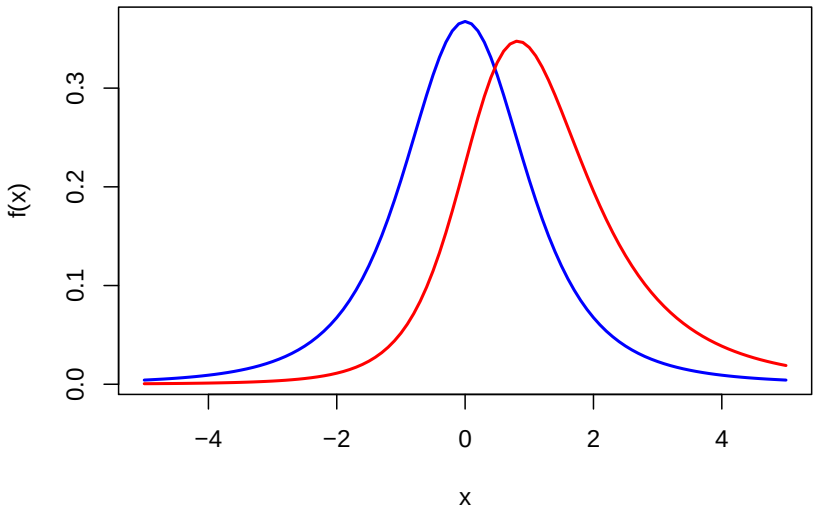
4.4.3.3 Definition (5.4.12) Say $X \sim N(\mu,1), Y \sim \chi^2(n)$, independent, then the random variable $T = \frac{X}{\sqrt{Y/n}}$ is said to have a *non-central t distribution* with n degrees of freedom and non-centrality parameter μ.

Note that if $X \sim N(\mu,\sigma^2), Y \sim \chi^2(n)$, independent, then $T = \frac{X/\sigma}{\sqrt{Y/n}} \sim t(n, \mu/\sigma)$

We can use R to calculate values for the non-central F distribution:

```
f=function(x) dt(x, 3)
f1=function(x) dt(x, 3, ncp = 1)
```

```r
curve(f, -5, 5, lwd=2, col="blue")
curve(f1, -5, 5, lwd=2, col="red", add=TRUE)
```



4.5 Distribution and Independence of Linear and Quadratic Forms

4.5.1 Distribution of Quadratic Forms

4.5.1.1 Theorem (5.5.1)

Let $\boldsymbol{X} \sim N_n(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then

$$(\boldsymbol{X}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{X}-\boldsymbol{\mu}) \sim \chi^2(n)$$

proof

$$\begin{aligned}
&(\boldsymbol{X}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{X}-\boldsymbol{\mu}) = \\
&(\boldsymbol{X}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{X}-\boldsymbol{\mu}) = \\
&\left[\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{X}-\boldsymbol{\mu})\right]'\left[\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{X}-\boldsymbol{\mu})\right] = \\
&\boldsymbol{Z}'\boldsymbol{Z}
\end{aligned}$$

where

$$\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{X}-\boldsymbol{\mu}) = \boldsymbol{Z} \sim N_n(\boldsymbol{0}, \boldsymbol{I})$$

because of (5.2.2)

4.5.1.2 Theorem (5.5.2) Let $\boldsymbol{X} \sim N_n(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, let $\boldsymbol{A}$ be a symmetric matrix of constants with rank r, and let $\lambda = \frac{1}{2}\boldsymbol{\mu}'\boldsymbol{A}\boldsymbol{\mu}$, then

$$\boldsymbol{X}'\boldsymbol{A}\boldsymbol{X} \sim \chi^2(r, \lambda)$$

iff $\boldsymbol{A}\boldsymbol{\Sigma}$ is idempotent.

proof

By (5.3.7) the moment generating function of $\boldsymbol{X}'\boldsymbol{A}\boldsymbol{X}$ is given by

$$\psi(t) = |\boldsymbol{I} - 2t\boldsymbol{A}\boldsymbol{\Sigma}|^{-1/2} \exp\left\{-\boldsymbol{\mu}'\left[\boldsymbol{I} - (\boldsymbol{I} - 2t\boldsymbol{A}\boldsymbol{\Sigma})^{-1}\right]\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}/2\right\}$$

The eigenvalues of $\boldsymbol{I} - 2t\boldsymbol{A}\boldsymbol{\Sigma}$ are $1 - 2t\lambda_i$, i=1,..,p, where λ_i are the eigenvalues of $\boldsymbol{A}\boldsymbol{\Sigma}$. In the proof of (5.3.8) we showed that

$$|\boldsymbol{I} - 2t\boldsymbol{A}\boldsymbol{\Sigma}| = \prod_{i=1}^{p}(1 - 2t\lambda_i)$$

and that

$$(\boldsymbol{I} - 2t\boldsymbol{A}\boldsymbol{\Sigma})^{-1} = \boldsymbol{I} + \sum_{i=1}^{\infty}(2t)^i(\boldsymbol{A}\boldsymbol{\Sigma})^i$$

if $|2t\lambda_i| < 1$ for all i. Therefore

$$\psi(t) = \prod_{i=1}^{p}(1 - 2t\lambda_i)^{-1/2} \exp\left\{-\boldsymbol{\mu}'\left[-\sum_{i=1}^{\infty}(2t)^i(\boldsymbol{A}\boldsymbol{\Sigma})^i\right]\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}/2\right\}$$

Suppose that $\boldsymbol{A}\boldsymbol{\Sigma}$ is idempotent with rank r, then r of the $\lambda_i's$ are equal to 1 and n-r are equal to 0, and $(\boldsymbol{A}\boldsymbol{\Sigma})^i = \boldsymbol{A}\boldsymbol{\Sigma}$, so

$$\psi(t) = (1 - 2t)^{-r/2} \exp\left\{-\boldsymbol{\mu}'\left[-\sum_{i=1}^{\infty}(2t)^i\right]\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}/2\right\} =$$

$$(1 - 2t)^{-r/2} \exp\left\{\boldsymbol{\mu}'\left[\sum_{i=0}^{\infty}(\frac{1}{1/(2t)})^i - 1\right]\boldsymbol{A}\boldsymbol{\mu}/2\right\} =$$

$$(1 - 2t)^{-r/2} \exp\left\{-\frac{1}{2}\boldsymbol{\mu}'\boldsymbol{A}\boldsymbol{\mu}[1 - 1/(1 - 2t)]\right\} =$$

which is the moment generating function of a chi-square with r degrees of freedom and non-centrality parameter $\frac{1}{2}\boldsymbol{\mu}'\boldsymbol{A}\boldsymbol{\mu}$.

The proof of the reverse is omitted.

4.5.1.3 Corollary (5.5.3)

i. If $\boldsymbol{X} \sim N_p(\boldsymbol{0}, \boldsymbol{I})$, then $\boldsymbol{X'AX} \sim \chi^2(r)$ iff $\boldsymbol{A}$ is idempotent of rank r.

ii. If $\boldsymbol{X} \sim N_p(\boldsymbol{\mu}, \sigma^2\boldsymbol{I})$, then $\boldsymbol{X'AX}/\sigma^2 \sim \chi^2(r, \boldsymbol{\mu'A\mu}/(2\sigma^2)$ iff $\boldsymbol{A}$ is idempotent of rank r.

4.5.1.4 Example (5.5.4) Say $\boldsymbol{X} = (X_1, .., X_n)'$ where $X_i \sim N(\mu, \sigma^2)$ and independent. Recall that we can write $\boldsymbol{X} \sim N_n(\mu\boldsymbol{j}, \sigma^2\boldsymbol{I})$ and that $\sum_{i=1}^n (x_i - \bar{x})^2 = \boldsymbol{x'}[\boldsymbol{I} - (1/n)\boldsymbol{J}]\boldsymbol{x}$. Now

$$
\begin{aligned}
&[\boldsymbol{I} - (1/n)\boldsymbol{J}][\boldsymbol{I} - (1/n)\boldsymbol{J}] = \\
&\boldsymbol{I}[\boldsymbol{I} - (1/n)\boldsymbol{J}] - (1/n)\boldsymbol{J}[\boldsymbol{I} - (1/n)\boldsymbol{J}] = \\
&\boldsymbol{I} - (1/n)\boldsymbol{J} - (1/n)\boldsymbol{J} + (1/n)^2\boldsymbol{JJ} = \\
&\boldsymbol{I} - (1/n)\boldsymbol{J} - (1/n)\boldsymbol{J} + (1/n)^2 n\boldsymbol{J} = \\
&\boldsymbol{I} - (1/n)\boldsymbol{J}
\end{aligned}
$$

and so $\boldsymbol{I} - (1/n)\boldsymbol{J}$ is idempotent. Also

$$rank(\boldsymbol{I} - (1/n)\boldsymbol{J}) = tr(\boldsymbol{I} - (1/n)\boldsymbol{J}) = n(1 - 1/n) = n - 1$$

.

Now

$$
\begin{aligned}
&\lambda = \boldsymbol{\mu'A\mu}/(2\sigma^2) = \\
&\mu\boldsymbol{j}'(\boldsymbol{I} - (1/n)\boldsymbol{J})\mu\boldsymbol{j}/(2\sigma^2) = \\
&\mu^2(\boldsymbol{j'j} - (1/n)\boldsymbol{j'Jj}/(2\sigma^2) = \\
&\mu^2(\boldsymbol{j'j} - (1/n)\boldsymbol{j'jj'j}/(2\sigma^2) = \\
&\mu^2(n - (1/n)n^2)/(2\sigma^2) = 0
\end{aligned}
$$

and so

$$(n-1)s^2/\sigma^2 = \boldsymbol{x'}[\boldsymbol{I} - (1/n)\boldsymbol{J}]\boldsymbol{x}/\sigma^2 \sim \chi^2(n)$$

4.5.2 Independence of Linear and Quadratic Forms

4.5.2.1 Theorem (5.5.5) Suppose $\boldsymbol{B}$ is a $k \times p$ matrix of constants, $\boldsymbol{A}$ is a $p \times p$ symmetric matrix of constants and $\boldsymbol{X} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Then $\boldsymbol{BX}$ and $\boldsymbol{X'AX}$ are independent iff $\boldsymbol{B\Sigma A} = \boldsymbol{O}$.

proof

Suppose $\boldsymbol{B\Sigma A} = \boldsymbol{O}$. We will also assume that $\boldsymbol{A}$ is idempotent. The general case can be found in textbooks.

We can write

$$\boldsymbol{X'AX} = \boldsymbol{X'A'AX} = \boldsymbol{(AX)'(AX)}$$

If $\boldsymbol{B\Sigma A} = \boldsymbol{O}$ we have $cov(\boldsymbol{BX}, \boldsymbol{AX}) = 0$. Because they are multivariate normal random vectors, uncorrelated implies independence.

For the reverse assume $\boldsymbol{BX}$ and $\boldsymbol{X'AX}$ are independent. Therefore $cov(\boldsymbol{BX}, \boldsymbol{X'AX}) = \boldsymbol{0}$ and so

$$2\boldsymbol{B\Sigma A\mu} = 0$$

this holds for all $\boldsymbol{\mu}$ and so $\boldsymbol{B\Sigma A} = 0$

4.5.2.2 Corollary (5.5.6)

$\boldsymbol{X} \sim N_p(\boldsymbol{\mu}, \sigma^2\boldsymbol{I})$. Then $\boldsymbol{BX}$ and $\boldsymbol{X'AX}$ are independent iff $\boldsymbol{BA} = \boldsymbol{O}$.

4.5.2.3 Example (5.5.7)

Say $\boldsymbol{X} \sim N_n(\mu\boldsymbol{j}, \sigma^2\boldsymbol{I})$. We can write

$$\bar{X} = (1/n)\boldsymbol{j'X}$$
$$s^2 = \boldsymbol{X'}[\boldsymbol{I} - (1/n)\boldsymbol{J}]\boldsymbol{X}/(n-1)$$

now

$$(1/n)\boldsymbol{j'}[\boldsymbol{I} - (1/n)\boldsymbol{J}] =$$
$$(1/n)\boldsymbol{j'} - (1/n)(1/n)\boldsymbol{j'J} =$$
$$(1/n)\boldsymbol{j'} - (1/n)(1/n)n\boldsymbol{j'} = 0$$

and so $\bar{X}$ and s^2 are independent.

4.5.2.4 Theorem (5.5.8)

Suppose $\boldsymbol{A}$ and $\boldsymbol{B}$ are symmetric matrices of constants and $\boldsymbol{X} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Then $\boldsymbol{X'AX}$ and $\boldsymbol{X'BX}$ are independent iff $\boldsymbol{A\Sigma B} = \boldsymbol{O}$.

proof

Suppose $\boldsymbol{A\Sigma B} = \boldsymbol{O}$. Again we will also assume that $\boldsymbol{A}$ and $\boldsymbol{B}$ are idempotent, so we can write $\boldsymbol{X'AX} = \boldsymbol{(AX)'AX}$ and $\boldsymbol{X'BX} = \boldsymbol{(BX)'BX}$. $\boldsymbol{A\Sigma B} = \boldsymbol{O}$ we have $cov(\boldsymbol{AX}, \boldsymbol{BX}) = \boldsymbol{A\Sigma B} = \boldsymbol{O}$. Because they are multivariate normal random vectors, uncorrelated implies independence and vice versa.

4.5.2.5 Example (5.5.9)

We have previously partitioned

$$\sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n}(x_i - \bar{x})^2 + n\bar{x}^2$$

(see (5.3.1)), which we can write as

$$\boldsymbol{x'x} = \boldsymbol{x'}[\boldsymbol{I} - (1/n)\boldsymbol{J}]\boldsymbol{x} + \boldsymbol{x'}(1/n)\boldsymbol{Jx}$$

now $\boldsymbol{x}'[\boldsymbol{I} - (1/n)\boldsymbol{J}]\boldsymbol{x}$ and $\boldsymbol{x}'(1/n)\boldsymbol{J}\boldsymbol{x}$ are independent iff $[\boldsymbol{I} - (1/n)\boldsymbol{J}][(1/n)\boldsymbol{J}] = \boldsymbol{0}$, which is easy to show.

4.5.2.6 Theorem (5.5.10) (*James' Theorem*)

Let $\boldsymbol{X} \sim N_n(\boldsymbol{\mu}, \sigma^2\boldsymbol{I})$, let $\boldsymbol{A}_i$ be symmetric with rank r_i, i=1,..,k. Set $\boldsymbol{A} = \sum_{i=1}^k \boldsymbol{A}_i$ and assume $\boldsymbol{A}$ is symmetric of rank r. Then

i. $\boldsymbol{X'A_iX}/\sigma^2 \sim \chi^2(r_i, \boldsymbol{\mu'A_i\mu}/(2\sigma^2))$

ii. $\boldsymbol{X'A_iX}$ and $\boldsymbol{X'A_jX}$ are independent for all $i \neq j$

iii. $\boldsymbol{X'AX} \sim \chi^2(r, \boldsymbol{\mu'A\mu}/(2\sigma^2))$

i-iii are true iff at least two of the following are true

a. each $\boldsymbol{A}_i$ is idempotent

b. $\boldsymbol{A}_i\boldsymbol{A}_j = 0$ for all $i \neq j$

c. $\boldsymbol{A}$ is idempotent

or if c and d are true, where

d. $r = \sum_{i=1}^k r_i$

proof omitted

5 Regression

5.1 Simple Linear Regression - The Model

5.1.1 The Model

5.1.1.1 Definition (6.1.1)

Let $\boldsymbol{y} = (y_1, .., y_n)$, $\boldsymbol{x} = (x_1, .., x_n)$, $\boldsymbol{\epsilon} = (\epsilon_1, .., \epsilon_n)$, β_0, β_1 numbers, then a model of the form

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i; i = 1, .., n$$

is called a *simple linear regression model*.

simple refers to the fact that there is only one predictor x.

We assume that $\boldsymbol{y}$ and $\boldsymbol{\epsilon}$ are random vectors whereas $\boldsymbol{x}$ is fixed. We will consider the case where $\boldsymbol{X}$ is random later.

We make the following assumptions:

1. $E[\epsilon_i] = 0$ (model is correct)

2. $var(\epsilon_i) = \sigma^2$ (equal variance, *homoscadasticity*)

3. $cov(\epsilon_i, \epsilon_j) = 0$ (independence)

Notice that so far there is no assumption regarding the distribution of the ϵ_i's.

5.1.2 Estimation

5.1.2.1 Definition (6.1.2) The method of *least squares* estimates parameters by minimizing

$$\hat{\boldsymbol{\epsilon}}'\hat{\boldsymbol{\epsilon}} = \sum_{i=1}^{n}(y_i - \hat{y})^2 = \sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2$$

The $\hat{\boldsymbol{\epsilon}}$ are called the *residuals*.

Now

$$\begin{aligned}
&0 = \frac{d\hat{\boldsymbol{\epsilon}}'\hat{\boldsymbol{\epsilon}}}{d\beta_0} = (-2)\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i) = \\
&(-2)\left(\sum_{i=1}^{n} y_i - n\beta_0 - \sum_{i=1}^{n}\beta_1 x_i\right) = \\
&(-2n)\left(\bar{y} - \beta_0 - \beta_1\bar{x}\right) \\
&0 = \frac{d\hat{\boldsymbol{\epsilon}}'\hat{\boldsymbol{\epsilon}}}{d\beta_1} = (-2)\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)x_i = \\
&(2n)\left(\overline{xy} - \beta_0\bar{x} - \beta_1\overline{x^2}\right) \\
&\beta_0 + \beta_1\bar{x} = \bar{y} \\
&\beta_0\bar{x} + \beta_1\overline{x^2} = \overline{xy} \\
&(\bar{y} - \beta_1\bar{x})\bar{x} + \beta_1\overline{x^2} = \overline{xy} \\
&\beta_1 = \frac{\overline{xy} - \bar{y}\bar{x}}{\overline{x^2} - \bar{x}^2} = \frac{\sum_{i=1}^{n} x_i y_i - n\bar{y}\bar{x}}{\sum_{i=1}^{n} x_i^2 - n\bar{x}^2}
\end{aligned}$$

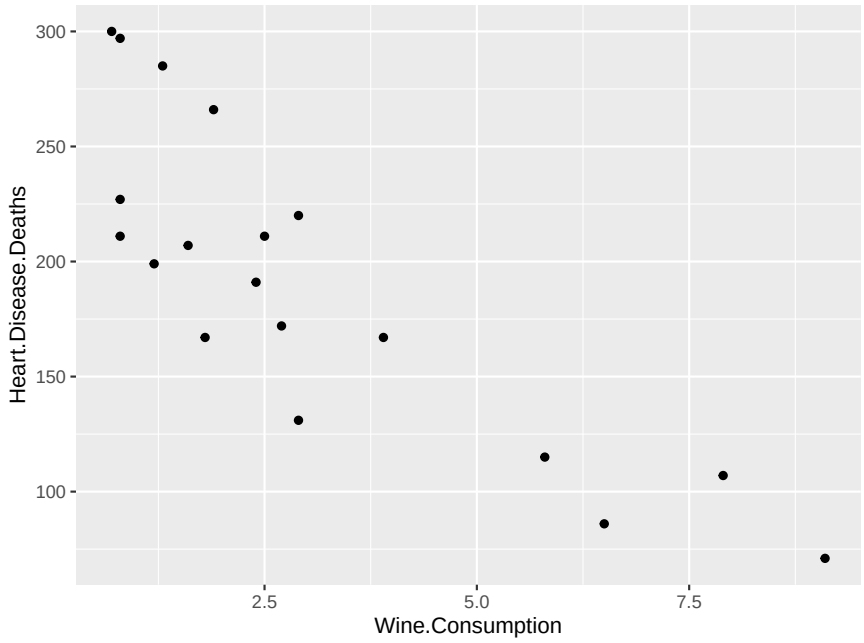
5.1.2.2 Example (6.1.3) We have data from a study for 19 developed countries on wine consumption (liters of wine per person per year) and deaths from heart disease (per 100000 people). (taken from David Moore: The Active Practice of Statistics, data set is part of Resma3.Rdata)

Note that strictly speaking this is **not** an experiment as described above because here $\boldsymbol{X}$ was random and not fixed. It turns out (and we will later study) that most results hold for both cases.

```
kable.nice(wine, do.row.names = FALSE)
```

Country	Wine.Consumption	Heart.Disease.Deaths
Australia	2.5	211
Austria	3.9	167
Belgium	2.9	131
Canada	2.4	191
Denmark	2.9	220
Finland	0.8	297
France	9.1	71
Iceland	0.8	211
Ireland	0.7	300
Italy	7.9	107
Netherlands	1.8	167
New Zealand	1.9	266
Norway	0.8	227
Spain	6.5	86
Sweden	1.6	207
Switzerland	5.8	115
United Kingdom	1.3	285
United States	1.2	199
Germany	2.7	172

```
ggplot(data=wine, aes(Wine.Consumption, Heart.Disease.Deaths)) +
  geom_point()
```

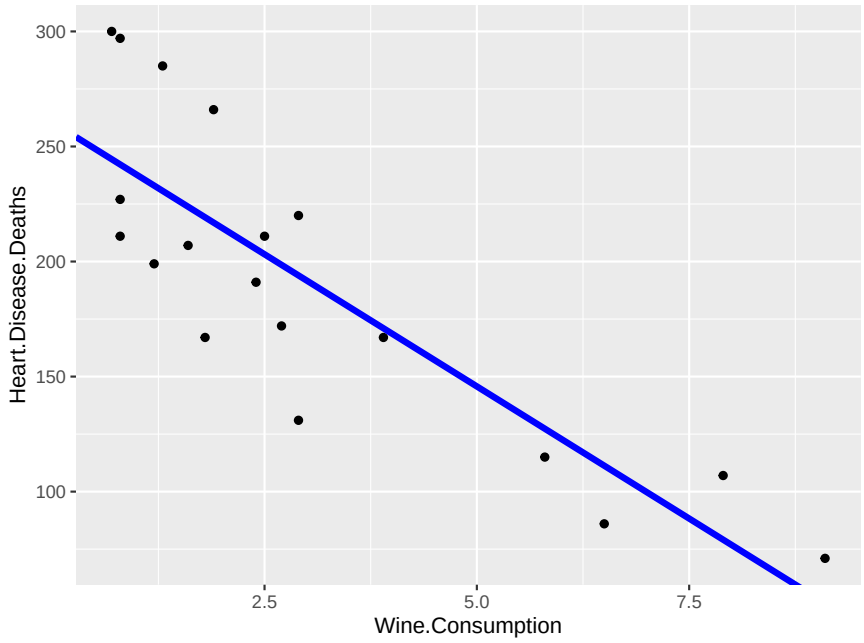


Let's find the least squares regression line:

```
xbar=mean(wine$Wine.Consumption)
x2bar=mean(wine$Wine.Consumption^2)
ybar=mean(wine$Heart.Disease.Deaths)
xybar=mean(wine$Wine.Consumption*wine$Heart.Disease.Deaths)
beta1=(xybar-xbar*ybar)/(x2bar-xbar^2)
beta0=ybar-beta1*xbar
round(c(beta0, beta1), 2)
```

```
## [1] 260.56 -22.97
```

```
ggplot(data=wine, aes(Wine.Consumption, Heart.Disease.Deaths)) +
  geom_point() +
  geom_abline(intercept = beta0, slope = beta1,
              color="blue", size=1.5)
```



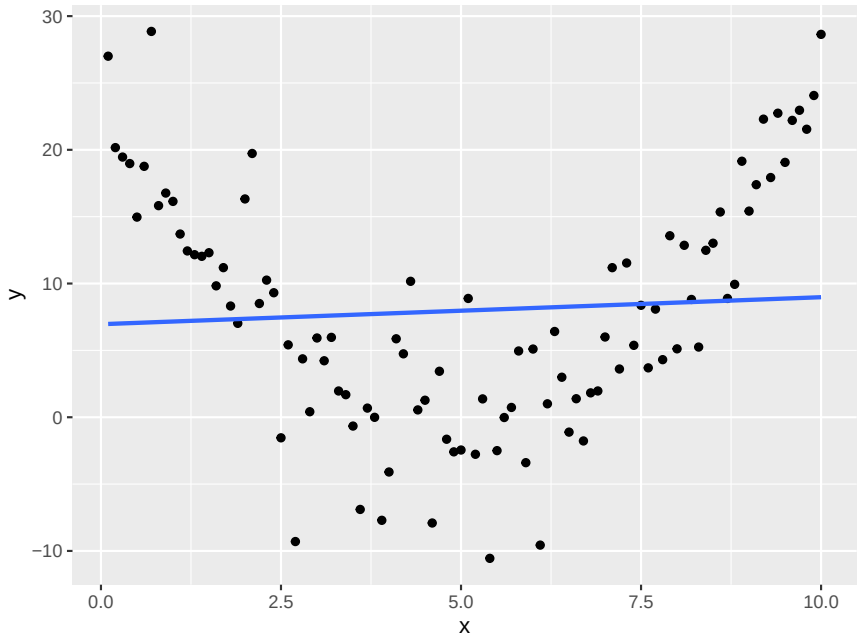
or we can let R do the work:

```
fit=lm(Heart.Disease.Deaths~Wine.Consumption, data=wine)
coef(fit)
```

```
##      (Intercept) Wine.Consumption
##        260.56338        -22.96877
```

Note that the least square regression line can always be found, even if the assumptions are not satisfied and the line is a bad model for the data:

```
x=1:100/10
y=(x-5)^2 + rnorm(100, 0, 5)
df=data.frame(x=x, y=y)
ggplot(data=df, aes(x, y)) +
  geom_point() +
  geom_smooth(method = "lm", se=FALSE)
```



5.1.2.3 Theorem (6.1.4) Under the three assumptions above we have

i. $E[\hat{\beta}_1] = \beta_1$

ii. $E[\hat{\beta}_0] = \beta_0$

iii. $var(\hat{\beta}_1) = \frac{\sigma^2}{\sum(x_i - \bar{x})^2}$

iv. $var(\hat{\beta}_0) = \sigma^2 \left[\frac{1}{n} + \frac{\bar{x}^2}{\sum(x_i - \bar{x})^2}\right]$

proof

i. note that

$$E[\bar{Y}] = E[\frac{1}{n}\sum Y_i] = \frac{1}{n}\sum E[Y_i] =$$
$$\frac{1}{n}\sum(\beta_0 + \beta_1 x_i) = \beta_0 + \beta_1 \bar{x}$$

$$E[\hat{\beta}_1] = E[\frac{\sum_{i=1}^n x_i Y_i - n\bar{Y}\bar{x}}{\sum_{i=1}^n x_i^2 - n\bar{x}^2}] =$$
$$\frac{\sum_{i=1}^n x_i E[Y_i] - nE[\bar{Y}]\bar{x}}{\sum_{i=1}^n x_i^2 - n\bar{x}^2} =$$
$$\frac{\sum_{i=1}^n x_i(\beta_0 + \beta_1 x_i) - n[\beta_0 + \beta_1\bar{x}]\bar{x}}{\sum_{i=1}^n x_i^2 - n\bar{x}^2} =$$
$$\frac{n\beta_0\bar{x} + \beta_1\sum_{i=1}^n x_i^2 - n\beta_0\bar{x} - n\beta_1\bar{x}^2}{\sum_{i=1}^n x_i^2 - n\bar{x}^2} =$$
$$\frac{\beta_1(\sum_{i=1}^n x_i^2 - n\bar{x}^2)}{\sum_{i=1}^n x_i^2 - n\bar{x}^2} = \beta_1$$

the other parts are similar

Until now we assumed that σ is known. If it is not is also has to be estimated from the data. To do so note

$$\sigma^2 = E[\epsilon_i^2] = E[(Y_i - \hat{Y}_i)^2]$$

for i=1,..,n. We can therefore estimate σ^2 as the mean of these deviations, however it turns out to be better to use

$$s^2 = \frac{\sum(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2}{n-2} = \frac{\text{SSE}}{n-2}$$

because then S^2 is an unbiased estimator of σ^2.

We define the *residual sum of squared errors* or *error sum of squares* **SSE** by

$$\text{SSE} = \sum(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2$$

5.1.3 Hypothesis Testing and Confidence Intervals for β_1

Notice that if $\beta_1 = 0$ we have $y_i = \beta_0 + \epsilon_i$ and there are no x's here, so this shows that x and y are independent. Therefore we might be interested in testing to see whether indeed $\beta_1 = 0$.

In order to do a hypothesis test we need to make some assumptions about the distribution of the ϵ_i. The usual one is

$$\epsilon_i \sim N(0, \sigma^2)$$

5.1.3.1 Theorem (6.1.5) If

$$\epsilon_i \sim N(0, \sigma^2)$$

for i=1,..,n, then

i. $\hat{\beta}_1 \sim N(\beta_1, \sigma^2/\sum_{i=1}^n (x_i - \bar{x})^2$

ii. $(n-2)s^2/\sigma^2 \sim \chi^2(n-2)$

iii. $\hat{\beta}_1$ and s^2 are independent

proof we will show these results later in greater generality.

5.1.3.2 Corollary (6.1.6)

$$t = \frac{\hat{\beta}_1}{s/\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2}} \sim t(n-2, \delta)$$

where the non-centrality parameter is

$$\delta = \beta_1/[\sigma/\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2}]$$

.

Therefore under the null hypothesis $H_0 : \beta_1 = 0$ we have $t \sim t(n-2)$ and a two-sided test rejects the null if

$$|t| > t_{\alpha/2, n-2}$$

The p-value of the test is given by

$$p = 2P(T > |t|)$$

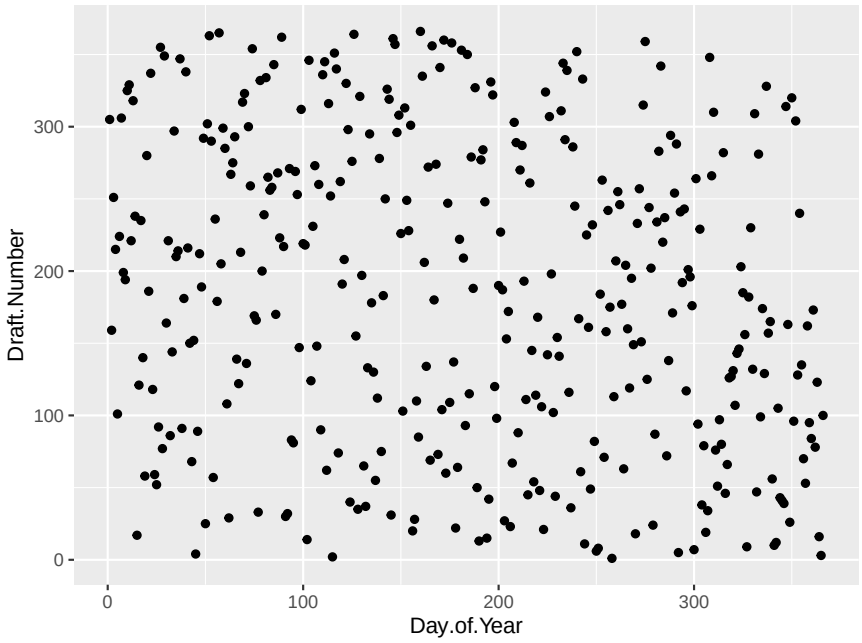
where $T \sim t(n-2)$.

5.1.3.3 Example (6.1.7) The 1970's Military Draft

In 1970, Congress instituted a random selection process for the military draft. All 366 possible birth dates were placed in plastic capsules in a rotating drum and were selected one by one. The first date drawn from the drum received draft number one and eligible men born on that date were drafted first. In a truly random lottery there should be no relationship between the date and the draft number.

Here is a scatterplot of Draft.Number by Day.of.Year:

```
ggplot(data=draft, aes(Day.of.Year, Draft.Number)) +
  geom_point()
```



There is not supposed to be a relationship between Day.of.Year and Draft. Number, so it makes sense to test $H_0 : \beta_1 = 0$.

```
x=draft$Day.of.Year
y=draft$Draft.Number
n=length(x)
beta1hat=(sum(x*y)-n*mean(x)*mean(y))/(sum(x^2)-n*mean(x)^2)
beta0hat=mean(y)-beta1hat*mean(x)
yhat=beta0hat+beta1hat*x
s2=sum((y-yhat)^2)/(n-2)
TS=beta1hat/(sqrt(s2/sum((x-mean(x))^2)))
c(TS, qt(0.95, n-2), 2*(1-pt(abs(TS), n-2)))
```

```
## [1] -4.427181e+00  1.649051e+00  1.263829e-05
```

Again, R can do it for us:

```
summary(lm(Draft.Number~Day.of.Year, data=draft))
```

```
##
## Call:
## lm(formula = Draft.Number ~ Day.of.Year, data = draft)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -210.837  -85.629   -0.519   84.612  196.157
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 225.00922   10.81197  20.811  < 2e-16
```

```
## Day.of.Year  -0.22606    0.05106  -4.427 1.26e-05
##
## Residual standard error: 103.2 on 364 degrees of freedom
## Multiple R-squared:  0.05109,    Adjusted R-squared:  0.04849
## F-statistic:  19.6 on 1 and 364 DF,  p-value: 1.264e-05
```

5.1.3.4 Theorem (6.1.8)

A $100(1-\alpha)\%$ confidence interval for β_1 is given by

$$\hat{\beta}_1 \pm t_{\alpha/2,n-2}\frac{s}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}}$$

proof follows by inverting the hypothesis test in (6.1.6)

5.1.3.5 Example (6.1.9)

A 95% confidence interval for the slope in the wine data set is given by

```
x=wine$Wine.Consumption
y=wine$Heart.Disease.Deaths
n=length(x)
yhat=beta0+beta1*x
sse = sum((y-yhat)^2)
s2hat=sse/(n-2)
denom=sum((x-mean(x))^2)
round(beta1+c(-1, 1)*qt(0.05/2, n-2)*sqrt(s2hat/denom), 2)
```

```
## [1] -15.46 -30.47
```

We will not discuss interval estimated and/or hypothesis tests for β_0. These of course exist but do not play a large role in Statistics. If indeed $H_0 : \beta_0 = 0$ is true the model becomes $\boldsymbol{y = \beta_1 x}$, what is called a *non-intercept model*. Whether such a model is appropriate for an experiment is usually better decided from the context of the experiment and not from some statistical analysis.

5.1.3.6 Example

We have data y_i, the amount of damage done by tropical storms and hurricanes in year i, and x_i, the number of such storms that hit Puerto Rico in year i. Clearly if $x_i = 0$ we immediately have $y_i = 0$, so a no-intercept model is appropriate.

5.1.4 Coefficient of Determination

5.1.4.1 Definition (6.1.10)

i. Residual sum of Squares

$$\text{SSE} = \sum (y_i - \hat{y}_i)^2$$

ii. Regression Sum of Squares

$$\text{SSR} = \sum (\hat{y}_i - \bar{y})^2$$

iii. Total Sum of Squares

$$\text{SST} = \sum (y_i - \bar{y})^2$$

We have **SST=SSR+SSE**

5.1.4.2 Definition (6.1.11)

The *coefficient of determination* r^2 is given by

$$r^2 = \frac{\text{SSR}}{\text{SSE}} = \frac{\sum (\hat{y}_i - \bar{y})^2}{\sum (y_i - \bar{y})^2}$$

An intuitive explanation for the coefficient of determination is as follows: it is the proportion of variation in the data explained by the model.

Comments

1.

$$r^2 = \frac{s_{xy}^2}{s_{xx} s_{yy}}$$

where s_{xy} was defined in (5.3.14). Therefore r is also the absolute value of the sample correlation coefficient.

2. Let t be the t statistic in (6.1.6), then

$$t = \frac{\hat{\beta}_1}{s / \sqrt{\sum_{i=1}^n (x_i - \bar{x})^2}} = \frac{\sqrt{n-2}\, r}{\sqrt{1 - r^2}}$$

If $H_0 : \beta_1 = 0$ is true, then $t \sim t(n-2)$.

5.1.4.3 Example (6.1.12)

For the data in example (6.1.3) we find

```
x=wine$Wine.Consumption
y=wine$Heart.Disease.Deaths
yhat=beta0+beta1*x
ybar=mean(y)
ssr=sum((yhat-ybar)^2)
sst=sum((y-ybar)^2)
r2=ssr/sst
round(r2, 2)
```

```
## [1] 0.71
```

5.2 Multiple Regression

5.2.1 The Model

5.2.1.1 Definition (6.2.1) We have a *response vector* $\boldsymbol{y} = (y_1, .., y_n)'$, a vector of *regression coefficients* $\boldsymbol{\beta} = (\beta_0, \beta_1, .., \beta_n)'$ and a *predictor matrix*

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_{11} & ... & x_{1k} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n1} & & x_{nk} \end{pmatrix}$$

We also have a vector of errors $\boldsymbol{\epsilon} = (\epsilon_0, .., \epsilon_n)'$. Then a model of the form

$$y_i = \beta_0 + \beta_1 x_{i1} + .. + \beta_k x_{ik}; i = 1, .., n$$

or

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

is called a *multiple regression model*. $\boldsymbol{X}$ is called the *design matrix*. As in the simple regression case we assume for now that $\boldsymbol{X}$ is fixed and not random.

Note that this includes models like

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2$$

$$y_i = \beta_0 + \beta_1 \log(x_i)$$

because they are models linear in the coefficients β.

The assumptions are the same as in the simple regression model:

1. $E[\epsilon_i] = 0$ (model is correct)
2. $var(\epsilon_i) = \sigma^2$ (equal variance, *homoscadasticity*)
3. $cov(\epsilon_i, \epsilon_j) = 0$ (independence)

5.2.2 Estimation of $\boldsymbol{\beta}$ and σ^2

Analogous to the simple regression case we can use the method of least squares and estimate the coefficients by minimizing

$$\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_{i1} - .. - \beta_k x_{ik})^2$$

Again one can differentiate this expression and solve the resulting system of equations, however here is a better solution:

5.2.2.1 Theorem (6.2.2) If $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$ where $\boldsymbol{X}$ is a $n \times (k+1)$ matrix of rank k+1<n, then the vector $\hat{\boldsymbol{\beta}}$ that minimizes the least squares criterion is given by

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X'X})^{-1}\boldsymbol{X'y}$$

proof

we have

$$\begin{aligned}
&\hat{\boldsymbol{\epsilon}}'\hat{\boldsymbol{\epsilon}} = (\boldsymbol{y} - (\boldsymbol{X}\hat{\boldsymbol{\beta}})'(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}) = \\
&\boldsymbol{y}'(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}) - (\boldsymbol{X}\hat{\boldsymbol{\beta}})'(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}) = \\
&\boldsymbol{y}'\boldsymbol{y} - \boldsymbol{y}'\boldsymbol{X}\hat{\boldsymbol{\beta}} - (\boldsymbol{y}'\boldsymbol{X}\hat{\boldsymbol{\beta}})' + \hat{\boldsymbol{\beta}}'\boldsymbol{X}'\boldsymbol{X}\hat{\boldsymbol{\beta}} = \\
&\boldsymbol{y}'\boldsymbol{y} - 2\boldsymbol{y}'\boldsymbol{X}\hat{\boldsymbol{\beta}} + \hat{\boldsymbol{\beta}}'\boldsymbol{X}'\boldsymbol{X}\hat{\boldsymbol{\beta}}
\end{aligned}$$

because $\boldsymbol{y}'\boldsymbol{X}\hat{\boldsymbol{\beta}}$ is a scalar.

By (4.3.20) and (4.3.21) we have

$$\begin{aligned}
&\partial\hat{\boldsymbol{\epsilon}}'\hat{\boldsymbol{\epsilon}}/\partial\boldsymbol{\beta} = \\
&\partial(\boldsymbol{y}'\boldsymbol{y} - 2\boldsymbol{y}'\boldsymbol{X}\hat{\boldsymbol{\beta}} + \hat{\boldsymbol{\beta}}'\boldsymbol{X}'\boldsymbol{X}\hat{\boldsymbol{\beta}})/\partial\boldsymbol{\beta} = \\
&\partial(\boldsymbol{y}'\boldsymbol{y})/\partial\boldsymbol{\beta} - \partial(2\boldsymbol{y}'\boldsymbol{X}\hat{\boldsymbol{\beta}})/\partial\boldsymbol{\beta} + \partial(\hat{\boldsymbol{\beta}}'\boldsymbol{X}'\boldsymbol{X}\hat{\boldsymbol{\beta}})/\partial\boldsymbol{\beta} = \\
&\boldsymbol{0} - 2\boldsymbol{X}'\boldsymbol{y} + 2\boldsymbol{X}'\boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{0} \\
&\boldsymbol{X}'\boldsymbol{X}\hat{\boldsymbol{\beta}} = \boldsymbol{X}'\boldsymbol{y}
\end{aligned}$$

$\boldsymbol{X}'\boldsymbol{X}$ is full-rank and therefore has an inverse, and so we have the result.

5.2.2.2 Definition

$$\boldsymbol{X}'\boldsymbol{X}\hat{\boldsymbol{\beta}} = \boldsymbol{X}'\boldsymbol{y}$$

are called the *normal equations*.

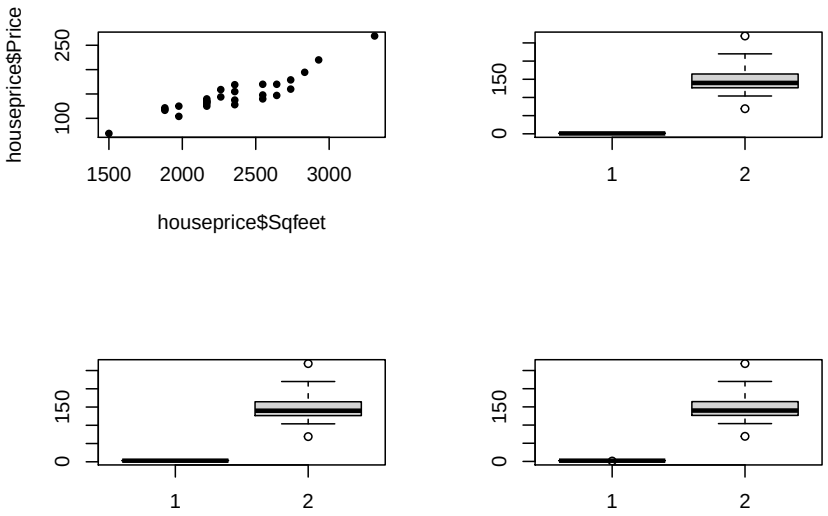
5.2.2.3 Example (6.2.3) Prices of 28 residencies located 30 miles south of a large metropolitan area.

```
kable.nice(houseprice, do.row.names = FALSE)
```

Price	Sqfeet	Floors	Bedrooms	Baths
69.0	1500.000	1	2	1.0
118.5	1880.952	1	2	2.0
104.0	1976.190	1	3	2.0
116.5	1880.952	1	3	2.0
121.5	1880.952	1	3	2.0
125.0	1976.190	1	3	2.0
128.0	2357.143	2	3	2.5
129.9	2166.667	1	3	1.7
133.0	2166.667	2	3	2.5
135.0	2166.667	2	3	2.5
137.5	2357.143	2	3	2.5
139.9	2166.667	1	3	2.0
143.9	2261.905	2	3	2.5
147.9	2547.619	2	3	2.5
154.9	2357.143	2	3	2.5
160.0	2738.095	2	3	2.0
169.0	2357.143	1	3	2.0
169.9	2642.857	1	3	2.0
125.0	2166.667	1	4	2.0
134.9	2166.667	1	4	2.0
139.9	2547.619	1	4	2.0
147.0	2642.857	1	4	2.0
159.0	2261.905	1	4	2.0
169.9	2547.619	2	4	3.0
178.9	2738.095	1	4	2.0
194.5	2833.333	2	4	3.0
219.9	2928.571	1	4	2.5
269.0	3309.524	2	4	3.0

Here we want to predict the Price from the other four variables, so

```
par(mfrow=c(2, 2))
plot(houseprice$Sqfeet, houseprice$Price, pch=20)
boxplot(houseprice$Floors, houseprice$Price)
boxplot(houseprice$Bedrooms, houseprice$Price)
boxplot(houseprice$Baths, houseprice$Price)
```




```
A=as.matrix(houseprice)
y=A[, 1, drop=FALSE]
X=cbind(1, A[, -1])
beta= solve(t(X)%*%X)%*%t(X)%*%y
round(c(beta), 3)
```

```
## [1] -67.620   0.086 -26.493  -9.286  37.381
```

We can write

$$\boldsymbol{X'X} = \begin{pmatrix} n & \sum_i x_{i1} & \sum_i x_{i2} & \dots & \sum_i x_{ik} \\ \sum_i x_{i1} & \sum_i x_{i1}^2 & \sum_i x_{i1}x_{i2} & \dots & \sum_i x_{i1}x_{ik} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ \sum_i x_{ik} & \sum_i x_{i1}x_{ik} & \sum_i x_{i2}x_{i2} & \dots & \sum_i x_{ik}^2 \end{pmatrix}$$

and

$$\boldsymbol{X'y} = \begin{pmatrix} \sum_i y_i \\ \sum_i x_{i1}y_i \\ \vdots \\ \sum_i x_{ik}y_i \end{pmatrix}$$

If $\hat{\boldsymbol{\beta}} = (\boldsymbol{X'X})^{-1}\boldsymbol{X'y}$, then

$\hat{\epsilon} = \boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}} = \boldsymbol{y} - \hat{\boldsymbol{y}}$

is the vector of *residuals*.

5.2.2.4 Example (6.2.4) Let's study a simple regression problem as a special case of a multiple regression problem. Then we have

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}$$

$$\boldsymbol{X'X} = \begin{pmatrix} n & \sum_i x_i \\ \sum_i x_i & \sum_i x_i^2 \end{pmatrix} \boldsymbol{X'y} = \begin{pmatrix} \sum_i y_i \\ \sum_i x_i y_i \end{pmatrix} (\boldsymbol{X'X})^{-1} = \frac{1}{n\sum_i x_i^2 - (\sum_i x_i)^2} \begin{pmatrix} \sum_i x_i^2 & -\sum_i x_i \\ -\sum_i x_i & n \end{pmatrix} (\boldsymbol{X'X})^{-1}\boldsymbol{X'y} =$$

which is the same (6.1.2)

5.2.3 Properties of Least Squares Estimators

We will assume that $\boldsymbol{X}$ is fixed and of full rank, as long as not stated otherwise.

5.2.3.1 Theorem (6.2.5) If $E[\boldsymbol{y}] = \boldsymbol{X\beta}$, then $\hat{\boldsymbol{\beta}}$ is an unbiased estimator of $\boldsymbol{\beta}$.

proof

$$\begin{aligned} &E[\hat{\boldsymbol{\beta}}] = E[(\boldsymbol{X'X})^{-1}\boldsymbol{X'y}] \\ &(\boldsymbol{X'X})^{-1}\boldsymbol{X'}E[\boldsymbol{y}] = \\ &(\boldsymbol{X'X})^{-1}\boldsymbol{X'X\beta} = \boldsymbol{\beta} \end{aligned}$$

5.2.3.2 Theorem (6.2.6) if $cov(\boldsymbol{y}) = \sigma^2\boldsymbol{I}$, then $cov(\hat{\boldsymbol{\beta}}) = \sigma^2(\boldsymbol{X'X})^{-1}$

proof

Using (5.1.11) we have

$$\begin{aligned} &cov(\hat{\boldsymbol{\beta}}) = \\ &cov((\boldsymbol{X'X})^{-1}\boldsymbol{X'y}) = \\ &[(\boldsymbol{X'X})^{-1}\boldsymbol{X'}]cov(\boldsymbol{y})[(\boldsymbol{X'X})^{-1}\boldsymbol{X'}]' = \\ &(\boldsymbol{X'X})^{-1}\boldsymbol{X'}\sigma^2\boldsymbol{X}(\boldsymbol{X'X})^{-1} = \\ &\sigma^2(\boldsymbol{X'X})^{-1}(\boldsymbol{X'X})(\boldsymbol{X'X})^{-1} = \\ &\sigma^2(\boldsymbol{X'X})^{-1} \end{aligned}$$

5.2.3.3 Example (6.2.7) In the case of simple linear regression we have

$$cov(\hat{\boldsymbol{\beta}}) = cov\begin{pmatrix}\hat{\beta}_0\\ \hat{\beta}_1\end{pmatrix} =$$
$$\begin{pmatrix} var(\hat{\beta}_0) & cov(\hat{\beta}_0,\hat{\beta}_1)\\ cov(\hat{\beta}_0,\hat{\beta}_1) & var(\hat{\beta}_1)\end{pmatrix} = \sigma^2(\boldsymbol{X'X})^{-1} =$$
$$\sigma^2\frac{1}{n\sum_i x_i^2 - (\sum_i x_i)^2}\begin{pmatrix}\sum_i x_i^2 & -\sum_i x_i\\ -\sum_i x_i & n\end{pmatrix} =$$
$$\frac{\sigma^2}{\sum_i (x_i - x_i)^2}\begin{pmatrix}\overline{x^2} & -\bar{x}\\ -\bar{x} & 1\end{pmatrix}$$

so we find

$$var(\hat{\beta}_0) = \frac{\sigma^2\overline{x^2}}{\sum_i (x_i - x_i)^2}$$
$$var(\hat{\beta}_1) = \frac{\sigma^2}{\sum_i (x_i - x_i)^2}$$
$$cov(\hat{\beta}_0,\hat{\beta}_1) = \frac{-\sigma^2\overline{x}}{\sum_i (x_i - x_i)^2}$$

Note that if $\bar{x} > 0$, $\hat{\beta}_0$ and $\hat{\beta}_1$ are negatively correlated.

5.2.3.4 Theorem (6.2.8) Gauss-Markov

If $E[\boldsymbol{y}] = \boldsymbol{X\beta}$ and $cov(\boldsymbol{y}) = \sigma^2\boldsymbol{I}$, then the least squares estimators have the smallest variance among all linear unbiased estimators.

proof

Any linear estimator can be written in the form $\boldsymbol{Ay}$. To be unbiased we have to have $E[\boldsymbol{Ay}] = \boldsymbol{\beta}$, and therefore we find

$$E[\boldsymbol{Ay}] = \boldsymbol{A}E[\boldsymbol{y}] = \boldsymbol{AX\beta} = \boldsymbol{\beta}$$

and since this has to hold for all possible $\boldsymbol{\beta}$ we have

$$\boldsymbol{AX} = \boldsymbol{I}$$

The covariance of $\boldsymbol{Ay}$ is given by

$$cov(\boldsymbol{Ay}) = \boldsymbol{A}cov(\boldsymbol{y})\boldsymbol{A'} = \sigma^2\boldsymbol{AA'}$$

The variances of the $\hat{\beta}_j$'s are the diagonal elements of $\sigma^2 \boldsymbol{A}\boldsymbol{A}'$, and so we need to choose $\boldsymbol{A}$, subject to $\boldsymbol{A}\boldsymbol{X} = \boldsymbol{I}$ so that the diagonal elements are minimized.

Let's write

$$\boldsymbol{A}\boldsymbol{A}' = (\boldsymbol{A} - (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}' + (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}')(\boldsymbol{A} - (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}' + (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}') = (a+b)(a+b)$$

where $a = \boldsymbol{A} - (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'$ and $b = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'$

now $(a+b)(a+b) = a^2 + ab + ba + b^2$, and we find

$$\begin{aligned} ab &= \left(\boldsymbol{A} - (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\right)\left((\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\right) = \\ &\boldsymbol{A}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}' - (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}' = \\ &\boldsymbol{A}\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1} - (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1} = \\ &\boldsymbol{I}(\boldsymbol{X}'\boldsymbol{X})^{-1} - (\boldsymbol{X}'\boldsymbol{X})^{-1} = \boldsymbol{0} \end{aligned}$$

also $ba = 0$ and

$$\begin{aligned} b^2 &= (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}' = \\ &(\boldsymbol{X}'\boldsymbol{X})^{-1}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{X} = \\ &(\boldsymbol{X}'\boldsymbol{X})^{-1} \end{aligned}$$

because $\boldsymbol{X}'\boldsymbol{X}$ is symmetric. So we find

$$\boldsymbol{A}\boldsymbol{A}' = \left(\boldsymbol{A} - (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\right)\left(\boldsymbol{A} - (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\right)' + (\boldsymbol{X}'\boldsymbol{X})^{-1}$$

The matrix $(\boldsymbol{A} - (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}')\,(\boldsymbol{A} - (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}')'$ is positive semidefinite by (4.2.10) and so the diagonal elements are greater or equal to 0. They are equal to 0 clearly if $\boldsymbol{A} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'$.

Comment

This theorem is quite remarkable because it has NO requirements on the distribution of the $\boldsymbol{y}$'s!

5.2.3.5 Definition (6.2.9) An estimator is called *BLUE* if it is minimum variance unbiased and linear.

So the Gauss-Markov theorem states that (under its conditions) least squares estimators are BLUE.

5.2.3.6 Corollary (6.2.10) if $E[\boldsymbol{y}] = \boldsymbol{X}\boldsymbol{\beta}$ and $cov(\boldsymbol{y}) = \sigma^2\boldsymbol{I}$, then the BLUE estimator of $\boldsymbol{a}'\boldsymbol{y}$ is $\boldsymbol{a}'\hat{\boldsymbol{\beta}}$.

The theorem also shows that the variance of the estimator depends on $\boldsymbol{X}$. As this is assumed to be fixed before the experiment, it is often under the control of the researcher. In this case it is often a good idea to choose $\boldsymbol{X}$ to be orthogonal so that $\boldsymbol{X'X}$ is diagonal. This often leads to maximizing the power of a hypothesis test.

5.2.3.7 Theorem (6.2.11)

If $\boldsymbol{x} = (1, x_1, .., x_k)'$ and $\boldsymbol{z} = (1, c_k x_1, .., c_k x_k)'$, then

$$\hat{\boldsymbol{\beta}}'\boldsymbol{x} = \hat{\boldsymbol{\beta}}_z'\boldsymbol{z}$$

where $\hat{\boldsymbol{\beta}}_z$ is the least squares estimator of the regression of $\boldsymbol{y}$ on $\boldsymbol{z}$.

proof omitted

This theorem states that least squares estimators are *invariant* under simple scalar multiplication, or changes of scale.

5.2.3.8 Corollary (6.2.12)

$\hat{\boldsymbol{y}}$ is invariant to a full-rank transformation of $\boldsymbol{X}$

5.2.4 Estimation of σ^2

As in simple regression, least squares does not give us an estimate of σ^2. By the assumptions we have $E[y_i] = \boldsymbol{x}_i'\boldsymbol{\beta}$ and so

$$\sigma^2 = E[(y_i - \boldsymbol{x}_i'\boldsymbol{\beta})^2]$$

Again we can estimate σ^2 with the average of these terms, so

$$s^2 = \frac{1}{n-k-1}\sum (y_i - \boldsymbol{x}_i'\hat{\boldsymbol{\beta}})^2$$

where we use n-k-1 so that s^2 is unbiased, see below. Note that by the above corollary $\boldsymbol{x}_i'\hat{\boldsymbol{\beta}}$ is BLUE for $\boldsymbol{x}_i'\boldsymbol{\beta}$.

By the proof of (6.2.2) we can write

$$\begin{aligned} s^2 &= \frac{1}{n-k-1}\sum (y_i - \boldsymbol{x}_i'\hat{\boldsymbol{\beta}})^2 = \\ &\frac{1}{n-k-1}(\boldsymbol{y} - \boldsymbol{X'}\hat{\boldsymbol{\beta}})'(\boldsymbol{y} - \boldsymbol{x'}\hat{\boldsymbol{\beta}}) = \\ &\frac{1}{n-k-1}(\boldsymbol{y'y} - \hat{\boldsymbol{\beta}}'\boldsymbol{X'y}) = \frac{\text{SSE}}{n-k-1} \end{aligned}$$

5.2.4.1 Theorem (6.2.13) If $E[\boldsymbol{y}] = \boldsymbol{X\beta}$ and $cov(\boldsymbol{y}) = \sigma^2\boldsymbol{I}$, then

$$E[s^2] = \sigma^2$$

proof

$$\begin{aligned}
&\text{SSE} = \\
&\boldsymbol{y'y} - \hat{\boldsymbol{\beta}}'\boldsymbol{X'y} = \\
&\boldsymbol{y'y} - [(\boldsymbol{X'X})^{-1}\boldsymbol{X'y}]'\boldsymbol{X'y} = \\
&\boldsymbol{y'y} - \boldsymbol{y'X}(\boldsymbol{X'X})^{-1}\boldsymbol{X'y} = \\
&\boldsymbol{y'}\left[\boldsymbol{I} - \boldsymbol{X}(\boldsymbol{X'X})^{-1}\boldsymbol{X'}\right]\boldsymbol{X'y}
\end{aligned}$$

Using (5.3.3) we find

$$\begin{aligned}
&E[\text{SSE}] = \\
&tr\left\{\left[\boldsymbol{I} - \boldsymbol{X}(\boldsymbol{X'X})^{-1}\boldsymbol{X'}\right]\sigma^2\boldsymbol{I}\right\} + E[\boldsymbol{y'}]\left[\boldsymbol{I} - \boldsymbol{X}(\boldsymbol{X'X})^{-1}\boldsymbol{X'}\right]E[\boldsymbol{y'}] = \\
&\sigma^2 tr\left\{\left[\boldsymbol{I} - \boldsymbol{X}(\boldsymbol{X'X})^{-1}\boldsymbol{X'}\right]\right\} + \boldsymbol{\beta'X'}\left[\boldsymbol{I} - \boldsymbol{X}(\boldsymbol{X'X})^{-1}\boldsymbol{X'}\right]\boldsymbol{X\beta'} = \\
&\sigma^2\left\{n - tr(\boldsymbol{X}(\boldsymbol{X'X})^{-1}\boldsymbol{X'})\right\} + \boldsymbol{\beta'X'X\beta} - \boldsymbol{\beta'X'X\beta} = \\
&\sigma^2\left\{n - tr(\boldsymbol{X}(\boldsymbol{X'X})^{-1}\boldsymbol{X'})\right\} = \\
&\sigma^2\left\{n - tr(\boldsymbol{I}_{k+1})\right\} = (n-k-1)\sigma^2
\end{aligned}$$

because $\boldsymbol{X'X}$ is a $(k+1)\times(k+1)$ matrix.

5.2.4.2 Corollary (6.2.14) $s^2(\boldsymbol{X'X})^{-1}$ is an unbiased estimator of $cov(\hat{\boldsymbol{\beta}})$.

5.2.4.3 Theorem (6.2.15) If $E[\boldsymbol{\epsilon}] = \boldsymbol{0}$, $cov(\boldsymbol{\epsilon}) = \sigma^2\boldsymbol{I}$ and $E[\epsilon_i^4] = 3\sigma^4$, then s^2 is the best (minimum variance) *quadratic* unbiased estimator of σ^2.

proof omitted

5.2.4.4 Example (6.2.16) Say $Z \sim N(0,1)$, then

$$\int_{-\infty}^{\infty} x^4 \exp\left\{-x^2/2\right\} dx =$$
$$\int_{-\infty}^{\infty} x^3[x \exp\left\{-x^2/2\right]\} dx =$$
$$x^3(-\exp\left\{-x^2/2\right\}|_{-\infty}^{\infty} - \int_{-\infty}^{\infty} 3x^2(-\exp\left\{-x^2/2\right\})dx =$$
$$3\int_{\infty}^{\infty} x^2 \exp\left\{-x^2/2\right\})dx$$
$$E[Z^4] = \int_{-\infty}^{\infty} x^4 \frac{1}{\sqrt{2\pi}} \exp\left\{-x^2/2\right\} dx =$$
$$3\int_{-\infty}^{\infty} x^2 \frac{1}{\sqrt{2\pi}} \exp\left\{-x^2/2\right\} dx = 3$$

Let $X \sim N(0, \sigma^2)$, then

$$E[X^4] = E[(\sigma Z))^4] = 3\sigma^4$$

so the condition of theorem (6.2.16) is fulfilled it the residual has a normal distribution.

5.2.4.5 Example (6.2.17)

For the houseprice data we find

```
A=as.matrix(houseprice)
n=nrow(A)
y=A[, 1, drop=FALSE]
X=cbind(1, A[, -1])
betahat= (solve(t(X)%*%X)%*%t(X))%*%y
sse=c(t(y)%*%y-t(betahat)%*%t(X)%*%y)
sse
```

```
## [1] 4321.864
```

```
sse/(n-4-1)
```

```
## [1] 187.9071
```

5.3 Geometric Interpretion, Centered Form

5.3.1 Geometric Interpretation

In this section we will use a purely geometric argument to derive the least squares estimators.

To start we will view the parameters $\boldsymbol{\beta} = (\beta_o, \beta_1, .., \beta_k)'$ as point in (k+1)-dimensional space, called the *parameter space*. Also the vector $\boldsymbol{y}$ can be viewed as a point in n-dimensional space, called the *data space*. The matrix $\boldsymbol{X}$ can be written as a partitioned matrix in k+1 columns as

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{j} & \boldsymbol{x}_1 & \ldots & \boldsymbol{x}_k \end{pmatrix}'$$

all the columns of this matrix are n-dimensional vectors and so are again points in n-dimensional space. Because $\boldsymbol{X}$ is assumed to be of rank k+1, the vectors are linearly independent. The set of linear combinations of of these vectors form a subspace of data space. The elements of this space are of the form

$$\boldsymbol{Xb} = b_0\boldsymbol{j} + b_1\boldsymbol{x_1} + .. + b_k\boldsymbol{x_k}$$

where $\boldsymbol{b}$ is a k+1 vector of scalars, that is any vector in parameter space. It can be shown that this is indeed a subspace, that is closed under addition and multiplication. It is said to be the subspace *spanned* by the columns of $\boldsymbol{X}$ and it is called the *prediction space*.

The columns of $\boldsymbol{X}$ form a *basis* of the prediction space.

Under the multiple regression model $\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\epsilon}$ we see that $\boldsymbol{y}$ is a vector in prediction space, $E[\boldsymbol{y}] = \boldsymbol{X\beta}$, plus a vector of random errors $\boldsymbol{\epsilon}$. Here neither $\boldsymbol{\beta}$ nor $\boldsymbol{\epsilon}$ is known.

In this setup multiple regression is the problem of finding a reasonable estimate of $E[\boldsymbol{y}]$ in prediction space and then finding the corresponding vector in parameter space.

What do we mean by "reasonable"? An obvious answer is in terms of distance, namely the point closest to $\boldsymbol{y}$. Again, one needs to say how a distance is defined, an in general there are many choices. Here we will use the basic Euclidean distance. Then from geometry we find that the point $\hat{\boldsymbol{\epsilon}} = \boldsymbol{y} - \hat{\boldsymbol{y}}$ has to be *perpendicular* to the prediction space. Because the prediction space is spanned by the columns of $\boldsymbol{X}$, the point $\hat{\boldsymbol{y}}$ must be such that $\hat{\boldsymbol{\epsilon}}$ is orthogonal to the columns of $\boldsymbol{X}$. From (4.1.12) we know that this means

$$\boldsymbol{X'\hat{\epsilon}} = \boldsymbol{0}$$

so

$$\begin{aligned}
&\boldsymbol{0} = \boldsymbol{X'\hat{\epsilon}} = \\
&\boldsymbol{X'}(\boldsymbol{y} - \hat{\boldsymbol{y}}) = \\
&\boldsymbol{X'}(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}) = \\
&\boldsymbol{X'y} - \boldsymbol{X'X}\hat{\boldsymbol{\beta}}
\end{aligned}$$

and so again we arrive at the *normal equations* $\boldsymbol{X'X}\hat{\boldsymbol{\beta}} = \boldsymbol{X'y}$!

5.3.2 The Model in Centered Form

5.3.2.1 Definition (6.3.1)

$$y_i = \beta_0 + \sum_{j=1}^{k} \beta_j x_{ij} + \epsilon_i =$$

$$\alpha + \sum_{j=1}^{n} \beta_j (x_{ij} - \bar{x}_j) + \epsilon_i$$

where

$$\alpha = \beta_0 + \sum_{j=1}^{k} \beta_j \bar{x}_j$$

and $\bar{x}_j = \frac{1}{n}\sum_{j=1}^{k} x_{ij}$ are the sample means of the variables. This is called the model in *centered form.*

In matrix notation we have

$$\boldsymbol{y} = \begin{pmatrix} \boldsymbol{j} & \boldsymbol{X}_c \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta}_1 \end{pmatrix} + \boldsymbol{\epsilon}$$

where $\boldsymbol{\beta} = \begin{pmatrix} \beta_1 & ... & \beta_k \end{pmatrix}'$,

$$\boldsymbol{X}_c = \left(\boldsymbol{I} - \frac{1}{n}\boldsymbol{J} \right) \boldsymbol{X}_1 = \begin{pmatrix} x_{11} - \bar{x}_1 & x_{12} - \bar{x}_2 & ... & x_{1k} - \bar{x}_k \\ x_{21} - \bar{x}_1 & x_{22} - \bar{x}_2 & ... & x_{2k} - \bar{x}_k \\ \vdots & \vdots & & \vdots \\ x_{n1} - \bar{x}_1 & x_{n2} - \bar{x}_2 & ... & x_{nk} - \bar{x}_k \end{pmatrix}$$

and $\boldsymbol{X}_1$ is $\boldsymbol{X}$ without the column of 1's.

In this form the normal equations become

$$\begin{pmatrix} \boldsymbol{j} & \boldsymbol{X}_c \end{pmatrix}' \begin{pmatrix} \boldsymbol{j} & \boldsymbol{X}_c \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta}_1 \end{pmatrix} = \begin{pmatrix} \boldsymbol{j} & \boldsymbol{X}_c \end{pmatrix} \boldsymbol{y}$$

but

$$\begin{pmatrix} \boldsymbol{j} & \boldsymbol{X}_c \end{pmatrix}' \begin{pmatrix} \boldsymbol{j} & \boldsymbol{X}_c \end{pmatrix} = \begin{pmatrix} n & \boldsymbol{0}' \\ \boldsymbol{0} & \boldsymbol{X}_c'\boldsymbol{X}_c \end{pmatrix}$$

and

$$\begin{pmatrix} \boldsymbol{j} & \boldsymbol{X}_c \end{pmatrix} \boldsymbol{y} = \begin{pmatrix} n\bar{y} \\ \boldsymbol{X}_c'\boldsymbol{y} \end{pmatrix}$$

and so the least squares estimates are given by

$$\hat{\alpha} = \bar{y}$$
$$\hat{\boldsymbol{\beta}}_1 = (\boldsymbol{X}_c'\boldsymbol{X}_c)^{-1}\boldsymbol{X_c}'\boldsymbol{y}$$

these are the same estimators as in the original model, and $\hat{\beta}_0 = \hat{\alpha} - \sum \hat{\beta}_j \bar{x}_j$

5.3.2.2 Example (6.3.2)

For the houseprice data we find

Standard form:

```
A=as.matrix(houseprice)
n=nrow(A)
y=A[, 1, drop=FALSE]
X=cbind(1, A[, -1])
betahat= (solve(t(X)%*%X)%*%t(X))%*%y
round(c(betahat), 4)
```

```
## [1] -67.6198   0.0857 -26.4931  -9.2862  37.3807
```

centered form:

```
A=as.matrix(houseprice)
n=nrow(A)
y=A[, 1, drop=FALSE]
Xc=A[, -1]
xbar=apply(Xc, 2, mean)
for(j in 1:4) Xc[ ,j]=Xc[ ,j]-xbar[j]
beta1hat= (solve(t(Xc)%*%Xc)%*%t(Xc))%*%y
round(c(beta1hat), 4)
```

```
## [1]   0.0857 -26.4931  -9.2862  37.3807
```

```
round(mean(y)-sum(beta1hat*xbar), 4)
```

```
## [1] -67.6198
```

Let S_{xx} be the matrix of sums of squares for the centered form, that is

$$(S_{xx})_{ij} = \frac{1}{n-1}\sum_{k=1}^{n}(x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j)$$

and s_{yx} be the vector with

$$(s_{yx})_i = \frac{1}{n-1}\sum_{k=1}^{n}(x_{ki} - \bar{x}_i)(y_k - \bar{y})$$

then we find $S_{xx} = \boldsymbol{X}_c'\boldsymbol{X}_c/(n-1)$ and $s_{yx} = \boldsymbol{X}_c'\boldsymbol{y}/(n-1)$, and using this we we can write

$$\hat{\boldsymbol{\beta}}_1 = S_{xx}^{-1}s_{xy}$$

$$\hat{\beta}_0 = \bar{y} - s_{yx}'S_{xx}^{-1}\bar{\boldsymbol{x}}$$

5.3.2.3 Example (6.3.3)

For the houseprice data we find

```
A=as.matrix(houseprice)
n=nrow(A)
y=A[, 1, drop=FALSE]
Xc=A[, -1]
xbar=apply(Xc, 2, mean)
for(j in 1:4) Xc[ ,j]=Xc[ ,j]-xbar[j]
Sxx=t(Xc)%*%Xc/(n-1)
syx=t(Xc)%*%y/(n-1)
beta1hat=solve(Sxx)%*%syx
beta0hat=mean(y)-t(syx)%*%solve(Sxx)%*%cbind(xbar)
round(c(beta0hat, beta1hat), 4)
```

```
## [1] -67.6198   0.0857 -26.4931  -9.2862  37.3807
```

5.3.3 Orthogonalization

Say we have a model of the form $\boldsymbol{y} = \boldsymbol{X_1}\boldsymbol{\beta_1} + \boldsymbol{X_2}\boldsymbol{\beta_2} + \boldsymbol{\epsilon}$ and a reduced model $\boldsymbol{y} = \boldsymbol{X_1}\boldsymbol{\beta_1^*} + \boldsymbol{\epsilon^*}$. Let $\hat{\boldsymbol{\beta}}_\mathbf{1}$ be the estimator of $\boldsymbol{\beta_1}$ and $\hat{\boldsymbol{\beta}}_\mathbf{1}^*$ the estimator of $\boldsymbol{\beta_1^*}$. Then it is generally not true that $\hat{\boldsymbol{\beta}}_\mathbf{1}$=$\hat{\boldsymbol{\beta}}_\mathbf{1}^*$.

5.3.3.1 Example

Let's use Sqfeet and Floors for $\boldsymbol{X}_1$ and bathrooms and baths for $\boldsymbol{X}_2$:

```
colnames(houseprice)
```

```
## [1] "Price"    "Sqfeet"   "Floors"   "Bedrooms" "Baths"
```

```
fit=lm(Price~., data=houseprice)
fit1=lm(Price~Sqfeet+Floors, data=houseprice)
coef(fit)
```

```
##  (Intercept)       Sqfeet       Floors     Bedrooms        Baths
## -67.61983705   0.08570823 -26.49305703  -9.28622097  37.38067201
```

```
coef(fit1)
```

```
## (Intercept)      Sqfeet      Floors
##  -60.876065    0.091688   -4.149169
```

and we see that the estimators for Sqfeet and Floors differ.

However:

5.3.3.2 Theorem (6.3.4)

If $\boldsymbol{X_1'X_2} = \boldsymbol{O}$ we have $\hat{\boldsymbol{\beta}}_\mathbf{1} = \hat{\boldsymbol{\beta}}_\mathbf{1}^*$

proof

We have $\hat{\boldsymbol{\beta}}_\mathbf{1}^* = (\boldsymbol{X_1'X_1})^{-1}\boldsymbol{X_1'y}$. For $\hat{\boldsymbol{\beta}}_\mathbf{1}$ we partition $\hat{\boldsymbol{\beta}}_\mathbf{1}^* = (\boldsymbol{X_1'X_1})^{-1}\boldsymbol{X_1'y}$:

$$\begin{pmatrix}\hat{\beta}_1\\ \hat{\beta}_2\end{pmatrix} = \begin{pmatrix}X_1'X_1 & X_1'X_2\\ X_2'X_1 & X_2'X_2\end{pmatrix}^{-1}\begin{pmatrix}X_1' & X_2'\end{pmatrix}\begin{pmatrix}y\\ y\end{pmatrix} = \begin{pmatrix}X_1'X_1 & O\\ O & X_2'X_2\end{pmatrix}^{-1}\begin{pmatrix}X_1'y\\ X_2'y\end{pmatrix} = \begin{pmatrix}(X_1'X_1)^{-1} & O\\ O & (X_2'X_2)^{-1}\end{pmatrix}\begin{pmatrix}X_1'y\\ X_2'y\end{pmatrix}$$

We can use this theorem to obtain estimators of $\boldsymbol{\beta_2}$ as follows:

1. Regress $\boldsymbol{y}$ on $\boldsymbol{X_1}$ and calculate residuals $\boldsymbol{y - \hat{y}(X_1)}$, where $\boldsymbol{\hat{y}(X_1) = X_1\hat{\beta}_1 = X_1(X_1'X_1)^{-1}X_1'y}$.
2. Regress the columns of $\boldsymbol{X_2}$ on $\boldsymbol{X_1}$ and obtain residuals $\boldsymbol{X_{2.1} = X_2 - \hat{X}_2(X_1)}$. If X_2 is written in terms of its columns as $\boldsymbol{X_2} = \begin{pmatrix}x_{21} & .. & x_{2j} & .. & x_{2p}\end{pmatrix}$, then the regression coefficient vector for $\boldsymbol{x_{2j}}$ is $\boldsymbol{b_j = (X_1'X_1)^{-1}X_1'x_{2j}}$, and so

$$\boldsymbol{\hat{x}_{2j} = X_1(X_1'X_1)^{-1}X_1'x_{2j}}$$

Taking all columns of $\boldsymbol{X}_2$ together we get

$$\boldsymbol{\hat{X}_2(X_1) = X_1(X_1'X_1)^{-1}X_1'X_2 = X_1A}$$

where $\boldsymbol{A = (X_1'X_1)^{-1}X_1'X_2}$ is called the *alias matrix*. Note that $\boldsymbol{X_{2.1} = X_2 - \hat{X}_2(X_1)}$ is orthogonal to $\boldsymbol{X}_1$:

$$\boldsymbol{X_1'X_{2.1} = X_1'X_2 - X_1'X_1(X_1'X_1)^{-1}X_1'X_2 = O}$$

3. Regress $\boldsymbol{y - \hat{y}(X_1)}$ on $\boldsymbol{X_{2.1}}$. Since $\boldsymbol{X_{2.1}}$ is orthogonal to $\boldsymbol{X}_1$, we obtain the same $\hat{\beta}_2$ as in the full model.

```
A=as.matrix(houseprice)
y=A[, 1, drop=FALSE]
X1=cbind(1, A[, 2:3])
X2=A[, 4:5, drop=FALSE]
yhatX1 = X1%*%solve(t(X1)%*%X1)%*%t(X1)%*%y
y1=y-yhatX1
X2hat.X1=X1%*%solve(t(X1)%*%X1)%*%t(X1)%*%X2
X2.1 = X2-X2hat.X1
round(t(X1)%*%X2.1, 5)
```

5.3.3.3 Example (6.3.5)

```
##         Bedrooms Baths
##                0     0
## Sqfeet         0     0
## Floors         0     0
```

```
round(c(solve(t(X2.1)%*%X2.1)%*%t(X2.1)%*%y1), 2)
```

```
## [1] -9.29 37.38
```

```r
round(coef(lm(Price~., data=houseprice)), 2)
```

```
## (Intercept)      Sqfeet      Floors    Bedrooms       Baths
##      -67.62        0.09      -26.49       -9.29       37.38
```

5.4 Normal Model and Coefficient of Determination R^2

5.4.1 Normal Model

We now add the assumption

$$\boldsymbol{y} \sim N_n(\boldsymbol{X}\boldsymbol{\beta}, \sigma^2\boldsymbol{I})$$

Under this assumption it is now possible to use maximum likelihood for estimation. We denote the likelihood function by $L(\boldsymbol{\beta}, \sigma^2)$ and the log likelihood function by $l(\boldsymbol{\beta}, \sigma^2) = \log L(\boldsymbol{\beta}, \sigma^2)$.

5.4.1.1 Theorem (6.4.1) If $\boldsymbol{y} \sim N_n(\boldsymbol{X}\boldsymbol{\beta}, \sigma^2\boldsymbol{I})$ where $\boldsymbol{X}$ is $n \times k+1$ with full rank k+1<n, the maximum likelihood estimators are

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y} =$$
$$\hat{\sigma}^2 = \frac{1}{n}(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})'(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})$$

so the mle of $\boldsymbol{\beta}$ is the least squares estimator.

proof (sketch)

$$l(\boldsymbol{\beta}, \sigma^2) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log\sigma^2 - \frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})'(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})$$

taking partial derivatives and setting them equal to 0 yields the desired result.

5.4.1.2 Theorem (6.4.2) Under the assumptions of theorem (6.4.1) we find

i. $\hat{\boldsymbol{\beta}} \sim N_{k+1}(\boldsymbol{\beta}, \sigma^2(\boldsymbol{X}'\boldsymbol{X})^{-1})$

ii. $n\hat{\sigma}^2/\sigma^2 \sim \chi^2(n-k-1)$

iii. $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}^2$ are independent

proof

i follows from (5.2.8) and ii and iii have been shown before for the least squares estimators

5.4.1.3 Theorem (6.4.3) Under the assumptions of theorem (6.4.1) $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}^2$ are jointly sufficient statistics

proof omitted

5.4.1.4 Theorem (6.4.4)

Under the assumptions of theorem (6.4.1) $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}^2$ have minimum variance among all unbiased estimators.

proof omitted

Note that this is a much stronger result than the Gauss-Markov theorem, which was about **linear** unbiased estimators.

5.4.2 R^2 in fixed-x regression

We can partition SST as follows: SST=SSR+SSE, where

$$\text{SSR} = \hat{\boldsymbol{\beta}}_1' \boldsymbol{X}_c' \boldsymbol{y}$$

is called the *regression sum of squares*.

5.4.2.1 Definition (6.4.5)

$$R^2 = \frac{\text{SSR}}{\text{SSE}} = \frac{\hat{\boldsymbol{\beta}}_1' \boldsymbol{X}_c' \boldsymbol{X}_c \hat{\boldsymbol{\beta}}_1}{\sum (y_i - \bar{y})^2} = \frac{\hat{\boldsymbol{\beta}}' \boldsymbol{X}' \boldsymbol{y} - n\bar{y}^2}{\boldsymbol{y}' \boldsymbol{y} - n\bar{y}^2}$$

is called the *coefficient of determination*. It provides a measure of how well the model fits the data.

5.4.2.2 Example (6.4.6)

For the houseprice data we find

```
A=as.matrix(houseprice)
n=nrow(A)
y=A[, 1, drop=FALSE]
X=cbind(1, A[, -1])
betahat= (solve(t(X)%*%X)%*%t(X))%*%y
(t(betahat)%*%t(X)%*%y-n*mean(y)^2)/(t(y)%*%y-n*mean(y)^2)
```

```
##           Price
## Price 0.8862443
```

so this model has an R^2 of 88.6%.

Of course we can also use R:

```
summary(lm(Price~., data=houseprice))
```

```
##
## Call:
## lm(formula = Price ~ ., data = houseprice)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -23.018  -5.943   1.860   5.947  30.955
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -67.61984   17.70818  -3.819 0.000882
## Sqfeet        0.08571    0.01076   7.966 4.62e-08
## Floors      -26.49306    9.48952  -2.792 0.010363
## Bedrooms     -9.28622    6.82985  -1.360 0.187121
## Baths        37.38067   12.26436   3.048 0.005709
##
## Residual standard error: 13.71 on 23 degrees of freedom
## Multiple R-squared:  0.8862, Adjusted R-squared:  0.8665
## F-statistic:  44.8 on 4 and 23 DF,  p-value: 1.558e-10
```

Here are some properties of R^2:

- $0 < R^2 < 1$
- if $\boldsymbol{\beta} = 0$ $R^2 = 0$
- $R = cor(\boldsymbol{y}, \hat{\boldsymbol{y}})^2$, that is R^2 is the square of the correlation between the observed and the predicted y's.

```
fit=lm(Price~., data=houseprice)
summary(fit)
```

5.4.2.3 Example (6.4.7)

```
##
## Call:
## lm(formula = Price ~ ., data = houseprice)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -23.018  -5.943   1.860   5.947  30.955
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -67.61984   17.70818  -3.819 0.000882
## Sqfeet        0.08571    0.01076   7.966 4.62e-08
## Floors      -26.49306    9.48952  -2.792 0.010363
## Bedrooms     -9.28622    6.82985  -1.360 0.187121
## Baths        37.38067   12.26436   3.048 0.005709
##
## Residual standard error: 13.71 on 23 degrees of freedom
## Multiple R-squared:  0.8862, Adjusted R-squared:  0.8665
## F-statistic:  44.8 on 4 and 23 DF,  p-value: 1.558e-10
```

```r
yhat=predict(fit)
cor(houseprice$Price, yhat)^2
```

```
## [1] 0.8862443
```

- Adding a variable to the predictors never decreases R^2.
- if $\boldsymbol{\beta} = 0$ $E[R^2] = k/(n-1)$
- R^2 is invariant under a full-rank linear transformation of the x's and under a scale transformation of the y's, but not a linear transformation of the x's and the y's simultaneously.
- $R^2 = \boldsymbol{s}'_{yx}\boldsymbol{S}_{xx}^{-1}\boldsymbol{s}_{yx}/s_y^2$

5.5 Generalized Least Squares

Until now we had the assumption that the y variables were independent. We will now study the case where $cov(\boldsymbol{y}) = \sigma^2\boldsymbol{V}$. One common case is were the variance of the y's increases (or decreases) as the x's increase (or decrease). Another is if the x's are time points, and one would expect responses close together in time to have some correlation.

So the model now is

$$\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\epsilon},\ E[\boldsymbol{y}] = \boldsymbol{X\beta},\ cov(\boldsymbol{y}) = \sigma^2\boldsymbol{V}$$

where $\boldsymbol{X}$ is full-rank and $\boldsymbol{V}$ is a known positive definite matrix.

Notice that $\boldsymbol{V}$ is an $n \times n$ matrix, and we have n observations, so estimation of $\boldsymbol{V}$ is not possible. Sometimes additional information on $\boldsymbol{V}$ is available and estimation is possible.

5.5.1 Estimation

5.5.1.1 Theorem (6.5.1) Under the model above we have

i. The BLUE estimator of $\boldsymbol{\beta}$ is

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X'V^{-1}X})^{-1}\boldsymbol{X'V^{-1}y}$$

ii.

$$cov(\hat{\boldsymbol{\beta}}) = \sigma^2(\boldsymbol{X'V^{-1}X})^{-1}$$

iii. an unbiased estimator of σ^2 is

$$\hat{s}^2 = (\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})'\boldsymbol{V}^{-1}(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})/(n-k-1)$$

proof omitted

5.5.1.2 Theorem (6.5.2) say $\boldsymbol{y} \sim N_n(\boldsymbol{X\beta}, \sigma^2\boldsymbol{V})$, then the maximum likelihood estimators are

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X'V^{-1}X})^{-1}\boldsymbol{X'V^{-1}y}$$

$$\hat{s}^2 = (\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})'\boldsymbol{V}^{-1}(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})/n$$

proof omitted

5.5.1.3 Example (6.5.3) Recall the centered model

$$\boldsymbol{y} = \begin{pmatrix}\boldsymbol{j} & \boldsymbol{X}_c\end{pmatrix}\begin{pmatrix}\boldsymbol{\alpha} \\ \boldsymbol{\beta}_1\end{pmatrix} + \boldsymbol{\epsilon}$$

with covariance pattern

$$\boldsymbol{\Sigma} = \begin{pmatrix}1 & \rho & \ldots & \rho \\ \rho & 1 & \ldots & \rho \\ \vdots & \vdots & & \vdots \\ \rho & \rho & \ldots & 1\end{pmatrix} = \sigma^2\left[(1-\rho)\boldsymbol{I} + \rho\boldsymbol{J}\right] = \sigma^2\boldsymbol{V}$$

so all variables have equal variance and any pair has the same correlation.

Note

$$\boldsymbol{X'V^{-1}X} = \begin{pmatrix}\boldsymbol{j}' \\ \boldsymbol{X}_c'\end{pmatrix}\boldsymbol{V}^{-1}\begin{pmatrix}\boldsymbol{j} & \boldsymbol{X}_c\end{pmatrix} = \begin{pmatrix}\boldsymbol{j'V^{-1}j} & \boldsymbol{j'V^{-1}X}_c \\ \boldsymbol{X}_c'\boldsymbol{V^{-1}j} & \boldsymbol{X}_c'\boldsymbol{V^{-1}X}_c\end{pmatrix}$$

We can find

$$\boldsymbol{V}^{-1} = a(\boldsymbol{I} - b\rho\boldsymbol{J})$$

where $a = 1/(1-\rho)$ and $b = 1/[(1+(n-1)\rho)]$. Now

$$\boldsymbol{X'V^{-1}X} =$$

$$\begin{pmatrix} \boldsymbol{j}'a(\boldsymbol{I}-b\rho\boldsymbol{J})\boldsymbol{j} & \boldsymbol{j}'a(\boldsymbol{I}-b\rho\boldsymbol{J})\boldsymbol{X}_c \\ \boldsymbol{X}'_c a(\boldsymbol{I}-b\rho\boldsymbol{J})\boldsymbol{j} & \boldsymbol{X}'_c a(\boldsymbol{I}-b\rho\boldsymbol{J})\boldsymbol{X}_c \end{pmatrix} =$$

$$\begin{pmatrix} a\boldsymbol{j}'\boldsymbol{j}-ab\rho\boldsymbol{j}'\boldsymbol{J}\boldsymbol{j} & a\boldsymbol{j}'\boldsymbol{X}_c-ab\rho\boldsymbol{j}'\boldsymbol{J}\boldsymbol{X}_c \\ a\boldsymbol{X}'_c\boldsymbol{j}-ab\rho\boldsymbol{X}'_c\boldsymbol{J}\boldsymbol{j} & a\boldsymbol{X}'_c\boldsymbol{X}_c-ab\rho\boldsymbol{X}'_c\boldsymbol{J}\boldsymbol{X}_c \end{pmatrix} =$$

$$\begin{pmatrix} an-ab\rho n^2 & a\boldsymbol{j}'\boldsymbol{X}_c-ab\rho n\boldsymbol{j}'\boldsymbol{X}_c \\ a\boldsymbol{X}'_c\boldsymbol{j}-ab\rho\boldsymbol{X}_c n\boldsymbol{j} & a\boldsymbol{X}'_c\boldsymbol{X}_c-ab\rho\boldsymbol{X}'_c\boldsymbol{J}\boldsymbol{X}_c \end{pmatrix} =$$

$$\begin{pmatrix} an(1-bn\rho) & a(1-bn\rho)\boldsymbol{j}'\boldsymbol{X}_c \\ a(1-bn\rho)\boldsymbol{X}'_c\boldsymbol{j} & \boldsymbol{X}'_c a(1-b\rho\boldsymbol{J})\boldsymbol{X}_c \end{pmatrix} =$$

$$\begin{pmatrix} bn & \boldsymbol{0}' \\ \boldsymbol{0} & a\boldsymbol{X}'_c\boldsymbol{X}_c \end{pmatrix}$$

because $\boldsymbol{X}_c$ is the centered matrix and so $\boldsymbol{j'X}_c = \boldsymbol{0}'$. Also we have

$$\boldsymbol{X'V^{-1}y} = \boldsymbol{X} = \begin{pmatrix} bn\bar{y} \\ a\boldsymbol{X}'_c\boldsymbol{y} \end{pmatrix}$$

and so

$$\begin{pmatrix} \hat{\alpha} \\ \hat{\beta}_1 \end{pmatrix} = \begin{pmatrix} \bar{y} \\ (\boldsymbol{X}'_c\boldsymbol{X}_c)^{-1}\boldsymbol{X}_c\boldsymbol{y} \end{pmatrix}$$

5.5.2 Weighted Regression

5.5.2.1 Example (6.5.4) Suppose we have a simple regression problem of the form

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

where $var(y_i) = \sigma^2 x_i$ and $cov(y_i, y_j) = 0$ for all $i \neq j$. (this is an example of a *weighted regression model*). So we have

$$\boldsymbol{V} = \sigma^2 \begin{pmatrix} x_1 & 0 & ... & 0 \\ 0 & x_2 & ... & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & .. & x_n \end{pmatrix}$$

and

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}$$

and

$$\boldsymbol{V}^{-1} = 1/\sigma^2 \begin{pmatrix} 1/x_1 & 0 & ... & 0 \\ 0 & 1/x_2 & ... & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & .. & 1/x_n \end{pmatrix}$$

$\boldsymbol{X'V^{-1}X} =$

$$\begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}' 1/\sigma^2 \begin{pmatrix} 1/x_1 & 0 & ... & 0 \\ 0 & 1/x_2 & ... & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & .. & 1/x_n \end{pmatrix} \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix} =$$

$$1/\sigma^2 \begin{pmatrix} 1/x_1 & 1/x_2 & ... & 1/x_n \\ 1 & 1 & ... & 1 \end{pmatrix} \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix} =$$

$$\begin{pmatrix} \sum 1/x_i & n \\ n & \sum x_i \end{pmatrix}$$

$(\boldsymbol{X'V^{-1}X})^{-1} =$

$$\frac{\sigma^2}{(\sum 1/x_i)(\sum x_i) - n^2} \begin{pmatrix} \sum x_i & -n \\ -n & \sum 1/x_i \end{pmatrix}$$

$\boldsymbol{X'V^{-1}y} =$

$$1/\sigma^2 \begin{pmatrix} 1/x_1 & 1/x_2 & ... & 1/x_n \\ 1 & 1 & ... & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} =$$

$$1/\sigma^2 \begin{pmatrix} \sum y_i/x_i \\ \sum y_i \end{pmatrix}$$

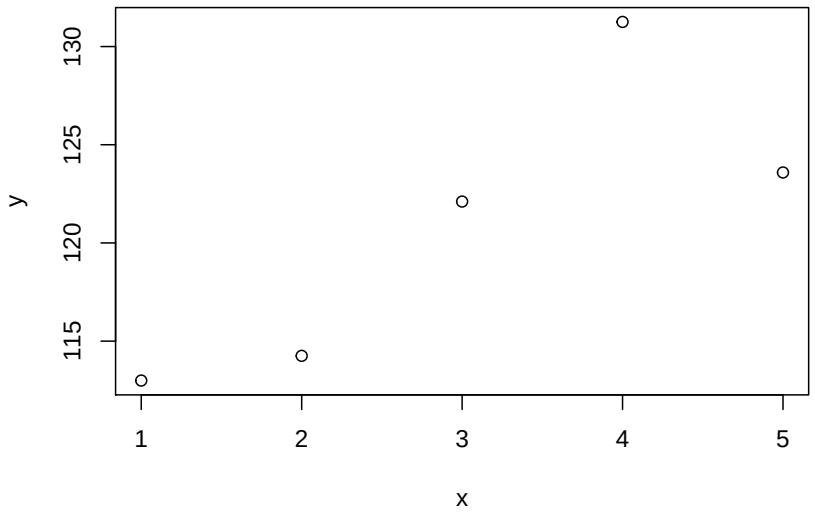
and so

$$\hat{\beta} = (\boldsymbol{X'V^{-1}X})^{-1}\boldsymbol{X'V^{-1}y} =$$

$$\frac{\sigma^2}{(\sum 1/x_i)(\sum x_i) - n^2}\begin{pmatrix}\sum x_i & -n \\ -n & \sum 1/x_i\end{pmatrix} 1/\sigma^2 \begin{pmatrix}\sum y_i/x_i \\ \sum y_i\end{pmatrix} =$$
$$\frac{\sigma^2}{(\sum 1/x_i)(\sum x_i) - n^2}\begin{pmatrix}(\sum y_i/x_i)(\sum x_i) - n\sum y_i \\ (\sum 1/x_i)(\sum y_i) - n(\sum y_i/x_i)\end{pmatrix}$$

Let's do a numerical example using R: say x=1,..,5 and y=100+10x+N(0, 5x), then

```
x=1:5
y=cbind(100+10*x+rnorm(5, 0, 5*x))
plot(x,y)
```




```
V=diag(x)
V
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    0    0    0    0
## [2,]    0    2    0    0    0
## [3,]    0    0    3    0    0
## [4,]    0    0    0    4    0
## [5,]    0    0    0    0    5
```

```
X=cbind(1,x)
X
```

```
##        x
## [1,] 1 1
## [2,] 1 2
```

```
## [3,] 1 3
## [4,] 1 4
## [5,] 1 5
```

```
Vinf=diag(1/x)
Xp.Vinf.X = t(X)%*%Vinf%*%X
c(sum(1/x), sum(x))
```

```
## [1]  2.283333 15.000000
```

```
Xp.Vinf.X
```

```
##              x
##   2.283333  5
## x 5.000000 15
```

```
A=solve(Xp.Vinf.X)
c(sum(x), -5, sum(1/x))/((sum(1/x)*sum(x)-5^2))
```

```
## [1]  1.6216216 -0.5405405  0.2468468
```

```
A
```

```
##                       x
##    1.6216216 -0.5405405
## x -0.5405405  0.2468468
```

```
B=t(X)%*%Vinf%*%y
c(sum(y/x), sum(y))
```

```
## [1] 268.3518 604.1924
```

```
B
```

```
##       [,1]
##   268.3518
## x 604.1924
```

```
c(sum(y/x)*sum(x)-5*sum(y),
  sum(1/x)*sum(y)-5*sum(y/x))/((sum(1/x)*sum(x)-5^2))
```

```
## [1] 108.574538   4.087978
```

```
A%*%B
```

```
##         [,1]
##   108.574538
## x   4.087978
```

so this works very well!

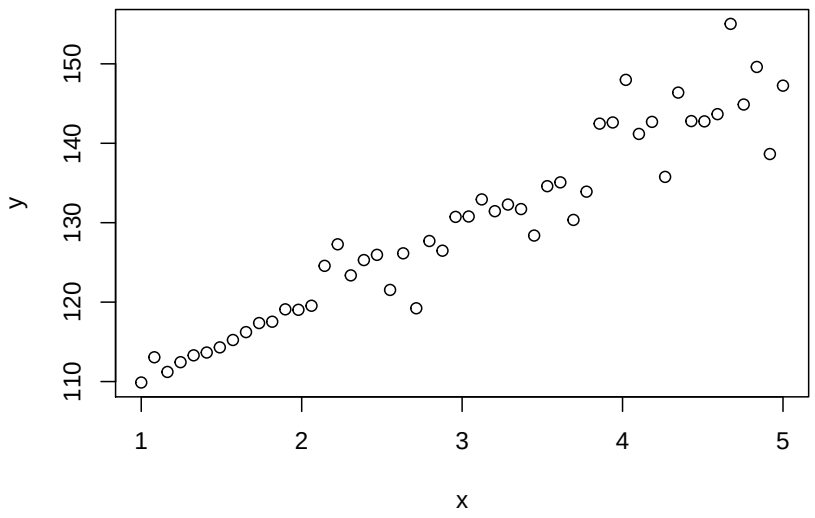
We can fit a weighted regression also with the R routine *lm* and the argument *weights*. These are the inverses of the variances:

```
summary(lm(y~x, weights=1/x))
```

```
##
## Call:
## lm(formula = y ~ x, weights = 1/x)
##
## Weighted Residuals:
##       1       2       3       4       5
##  0.3303 -1.7673  0.7326  3.1637 -2.4271
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  108.575      3.261  33.298 5.95e-05
## x              4.088      1.272   3.213   0.0488
##
## Residual standard error: 2.561 on 3 degrees of freedom
## Multiple R-squared:  0.7749, Adjusted R-squared:  0.6998
## F-statistic: 10.33 on 1 and 3 DF,  p-value: 0.04883
```

5.5.2.2 Example (6.5.5) Let's do a simple simulation to see the difference between ordinary and weighted regression. We use the model similar to the last example:

```
gen.data=function(n=50, beta0=100, beta1=10, sigma2=1) {
   x=seq(1, 5, length=n)
   y=beta0+beta1*x+rnorm(n, 0, sigma2*x)
   cbind(x, y)
}
plot(gen.data())
```

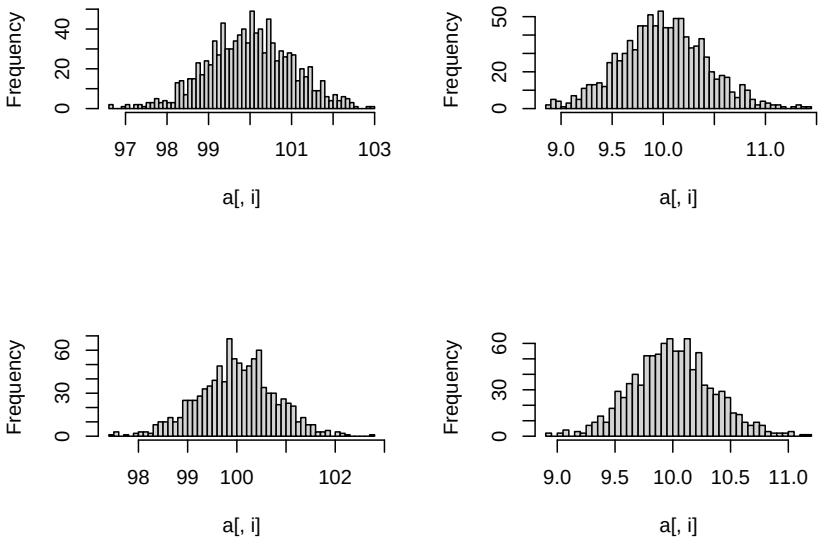


Now we do the following: we generate a data set, estimate the parameters with both ordinary and weighted least squares, and repeat this 1000 times:

```
B=1000
a=matrix(0, B, 4)
for(i in 1:B) {
   x=gen.data()
   a[i, 1:2]=coef(lm(x[ ,2]~x[ ,1]))
   a[i, 3:4]=coef(lm(x[ ,2]~x[ ,1], weights=1/x[ ,1]))
}
```

Let's see:

```
par(mfrow=c(2,2))
for(i in 1:4)
   hist(a[, i], 50, main="")
```




```
apply(a, 2, mean)
```

```
## [1] 99.999575  9.999502 99.998931  9.999717
```

```
apply(a, 2, var)
```

```
## [1] 1.0577596 0.1791154 0.6136294 0.1233135
```

and so we see that the weighted least squares estimators have smaller variance

5.6 Hypothesis Tests in Regression

5.6.1 Test for Overall Regression

We start with a test to see whether any of the predictor variables is useful. So let $\boldsymbol{\beta}_1 = (\beta_1 \ ... \ \beta_k)'$, then we wish to test

$$H_0 : \boldsymbol{\beta}_1 = \mathbf{0}$$

To find a test we will use the centered model

$$\boldsymbol{y} = \begin{pmatrix} \boldsymbol{j} & \boldsymbol{X}_c \end{pmatrix} \begin{pmatrix} \alpha \\ \boldsymbol{\beta}_1 \end{pmatrix} + \boldsymbol{\epsilon}$$

where $\boldsymbol{X}_c = [\boldsymbol{I} - (1/n)\boldsymbol{J}]\boldsymbol{X}_1$ is the centered matrix and $\boldsymbol{X}_1$ is $\boldsymbol{X}$ without the column of 1's. The corrected total sum of squares is given by

$$\begin{aligned} &\textstyle\sum (y_i - \bar{y})^2 = \\ &\hat{\boldsymbol{\beta}}_1' \boldsymbol{X}_c' \boldsymbol{y} + \left[\textstyle\sum (y_i - \bar{y})^2 - \hat{\boldsymbol{\beta}}_1' \boldsymbol{X}_c' \boldsymbol{y} \right] = \\ &\hat{\boldsymbol{\beta}}_1' \boldsymbol{X}_c' \boldsymbol{X}_c \hat{\boldsymbol{\beta}}_1 + \text{SSE} = \\ &\text{SSR} + \text{SSE} \end{aligned}$$

using (6.4.5).

Recall the following formulas:

$$\begin{aligned} &\textstyle\sum (y_i - \bar{y})^2 = \boldsymbol{y}'[\boldsymbol{I} - (1/n)\boldsymbol{J}]\boldsymbol{y} \\ &\hat{\boldsymbol{\beta}}_1 = (\boldsymbol{X}_c' \boldsymbol{X}_c)^{-1} \boldsymbol{X}_c' \boldsymbol{y} \\ &\text{SSE} = \textstyle\sum (y_i - \bar{y})^2 - \hat{\boldsymbol{\beta}}_1 \boldsymbol{X}_c' \boldsymbol{y} \end{aligned}$$

so we have

$$\begin{aligned} &\text{SSR} + \text{SSE} = \textstyle\sum (y_i - \bar{y})^2 = \boldsymbol{y}'[\boldsymbol{I} - (1/n)\boldsymbol{J}]\boldsymbol{y} = \\ &\boldsymbol{y}'\boldsymbol{X}_c(\boldsymbol{X}_c'\boldsymbol{X}_c)^{-1}\boldsymbol{X}_c'\boldsymbol{y} + \boldsymbol{y}'[\boldsymbol{I} - (1/n)\boldsymbol{J}]\boldsymbol{y} - \boldsymbol{y}'\boldsymbol{X}_c(\boldsymbol{X}_c'\boldsymbol{X}_c)^{-1}\boldsymbol{X}_c'\boldsymbol{y} = \\ &\boldsymbol{y}'\boldsymbol{H}_c\boldsymbol{y} + \boldsymbol{y}'[\boldsymbol{I} - (1/n)\boldsymbol{J} - \boldsymbol{H}_c]\boldsymbol{y} = \end{aligned}$$

where $\boldsymbol{H}_c = \boldsymbol{X}_c(\boldsymbol{X}_c'\boldsymbol{X}_c)^{-1}\boldsymbol{X}_c'$.

5.6.1.1 Theorem (6.6.1) The matrices above have the following properties:

i. $\boldsymbol{H}_c[\boldsymbol{I} - (1/n)\boldsymbol{J}] = \boldsymbol{H}_c$

ii. $\boldsymbol{H}_c$ is idempotent of rank k

iii. $\boldsymbol{I} - (1/n)\boldsymbol{J} - \boldsymbol{H}_c$ is idempotent of rank n-k-1

iv. $\boldsymbol{H}_c[\boldsymbol{I} - (1/n)\boldsymbol{J} - \boldsymbol{H}_c] = \boldsymbol{O}$

proof follow from direct calculation

5.6.1.2 Theorem (6.6.2)

If $\boldsymbol{y} \sim N_n(\boldsymbol{X}\boldsymbol{\beta}, \sigma^2\boldsymbol{I})$ then

i. $\text{SSR}/\sigma^2 = \hat{\boldsymbol{\beta}}_1'\boldsymbol{X}_c'\boldsymbol{X}_c\hat{\boldsymbol{\beta}}_1/\sigma^2 \sim \chi^2(k, \lambda_1)$ where $\lambda_1 = \boldsymbol{\beta}_1'\boldsymbol{X}_c'\boldsymbol{X}_c\boldsymbol{\beta}_1/(2\sigma^2)$

ii. $\text{SSE}/\sigma^2 = [\sum(y_i - \bar{y})^2 - \hat{\boldsymbol{\beta}}_1'\boldsymbol{X}_c'\boldsymbol{X}_c\hat{\boldsymbol{\beta}}_1]/\sigma^2 \sim \chi^2(n-k-1)$

iii. SSR and SSE are independent

proof i and ii follow from the calculation above and (5.2.2). The proof of iii is omitted.

5.6.1.3 Theorem (6.6.3)

Under the conditions of theorem (6.6.2) let

$$F = \frac{\text{SSR}/k}{\text{SSE}/(n-k-1)}$$

then

i. if $H_0 : \boldsymbol{\beta}_1 = \boldsymbol{0}$ is false $F \sim F(k, n-k-1, \lambda_1)$, where $\lambda_1 = \boldsymbol{\beta}_1'\boldsymbol{X}_c'\boldsymbol{X}_c\boldsymbol{\beta}_1/(2\sigma^2)$

ii. if $H_0 : \boldsymbol{\beta}_1 = \boldsymbol{0}$ is true $F \sim F(k, n-k-1)$

proof

see (5.4.6) and (6.6.1)

The test $H_0 : \boldsymbol{\beta}_1 = \boldsymbol{0}$ is done as follows: Reject H_0 if $F \geq F_{\alpha,k,n-k-1}$, where $F_{\alpha,k,n-k-1}$ is the upper α percentile of a (central) F distribution with k and n-k-1 degrees of freedom.

The p value of the test is given by $P(X > F)$, where $X \sim F(k, n-k-1)$.

The result of such a test is usually presented in the form of an ANOVA table, which looks as follows:

Source	df	SS	F	TS
Due to β_1	k	SSR	SSR/k	F
Error	$n-k-1$	SSE	SSE/$(n-k-1)$	
Total	$n-1$	SST		

5.6.1.4 Example (6.6.4)

We run this the test for the houseprice data

```
A=as.matrix(houseprice)
n=nrow(A);k=ncol(A)-1
c(n, k, n-k-1)
```

```
## [1] 28  4 23
```

```
y=A[, 1, drop=FALSE]
Xc=A[, -1]
xbar=apply(Xc, 2, mean)
for(j in 1:4) Xc[ ,j]=Xc[ ,j]-xbar[j]
beta1hat= (solve(t(Xc)%*%Xc)%*%t(Xc))%*%y
round(c(beta1hat), 4)
```

```
## [1]   0.0857 -26.4931  -9.2862  37.3807
```

$$SSR = \hat{\boldsymbol{\beta}}_1' \boldsymbol{X}_c' \boldsymbol{y}$$

```
ssr=t(beta1hat)%*%t(Xc)%*%y
ssr/c(1, k)
```

```
## [1] 33670.654  8417.663
```

$$SSE = \sum (y_i - \bar{y})^2 - SSR$$

```
sst=sum((y-mean(y))^2)
sst
```

```
## [1] 37992.52
```

```
sse=sst-ssr
sse/c(1, n-k-1)
```

```
## [1] 4321.8636  187.9071
```

```
FTS = (ssr/k)/(sse/(n-k-1))
FTS
```

```
##          Price
## Price 44.79694
```

Source	df	SS	F	TS
Due to β_1	4	33670	8418	44.8
Error	23	4322	187.9	
Total	27	37992		

If we test at the 5% level the critical value is

```
qf(0.95, k, n-k-1)
```

```
## [1] 2.795539
```

$F = 44.8 > 2.8$, and so we reject the null hypothesis, at least some of the variables are useful for predicting the house prices.

The p value of the test is

```
1-pf(FTS, k, n-k-1)
```

```
##              Price
## Price 1.55808e-10
```

5.6.2 Test on Subsets of $\boldsymbol{\beta}$

Say we have split $\boldsymbol{\beta} = (\boldsymbol{\beta}_1\ \boldsymbol{\beta}_2)'$ and we wish to test $H_0 : \boldsymbol{\beta}_2 = 0$. Then we can partition $\boldsymbol{X}$ accordingly, so the model becomes

$$\boldsymbol{y} = \boldsymbol{X_1\beta_1} + \boldsymbol{X_2\beta_2} + \boldsymbol{\epsilon}$$

where the intercept β_0 is included in $\boldsymbol{\beta}_1$.

We define matrices

$$\boldsymbol{H} = \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'$$
$$\boldsymbol{H}_1 = \boldsymbol{X}_1(\boldsymbol{X}_1'\boldsymbol{X}_1)^{-1}\boldsymbol{X}_1'$$

so that $\boldsymbol{Hy}$ is the least squares estimator of $\boldsymbol{\beta}$ under the full model and $\boldsymbol{H_1y}$ is the least squares estimator of $\boldsymbol{\beta}_1$ under reduced model assuming the null hypothesis is true.

5.6.2.1 Theorem (6.6.5)

$\boldsymbol{H} - \boldsymbol{H}_1$ is idempotent with rank h, where h is the number of elements in $\boldsymbol{\beta}_2$

proof omitted

5.6.2.2 Theorem (6.6.6)

If $\boldsymbol{y} \sim N_n(\boldsymbol{X\beta}, \sigma^2\boldsymbol{I})$ then

i. $\boldsymbol{y'(I - H)y}/\sigma^2 \sim \chi^2(n - k - 1)$

ii. $\boldsymbol{y'(H - H_1)y}/\sigma^2 \sim \chi^2(h, \lambda_1)$, where

$$\lambda_1 = \boldsymbol{\beta}_2'[\boldsymbol{X}_2'\boldsymbol{X}_2 - \boldsymbol{X}_2'\boldsymbol{X}_1(\boldsymbol{X}_1'\boldsymbol{X}_1)^{-1}\boldsymbol{X}_1'\boldsymbol{X}_2]\boldsymbol{\beta}_2/(2\sigma^2)$$

iii. $\boldsymbol{y'(I - H)y}$ and $\boldsymbol{y'(H - H_1)y}$ are independent

proof omitted

5.6.2.3 Theorem (6.6.7) If $\boldsymbol{y} \sim N_n(\boldsymbol{X\beta}, \sigma^2\boldsymbol{I})$ and define

$$F = \frac{\boldsymbol{y'(H - H_1)y}/h}{\boldsymbol{y'(I - H)y}/(n-k-1)} =$$

then

i. if $H_0 : \boldsymbol{\beta_2} = 0$ is false $F \sim F(h, n-k-1, \lambda_1)$

ii. if $H_0 : \boldsymbol{\beta_2} = 0$ is true $F \sim F(h, n-k-1)$

proof same as above

so the test rejects the null hypothesis if $F \geq F_{\alpha,h,n-k-1}$

5.6.2.4 Example (6.6.8) Let's consider the houseprice data. First note

```
round(cor(houseprice)[1, -1], 3)
```

```
##   Sqfeet   Floors Bedrooms    Baths
##    0.915    0.291    0.605    0.653
```

that the variable with the smallest correlation with Price is Floors, so one might want to test whether Floors is a useful predictor variable. Let's see:

```
A=as.matrix(houseprice)
n=nrow(A)
k=ncol(A)-1
X=cbind(1, A[, -1])
y=A[, 1, drop=FALSE]
X1=X[, -3]
H=X%*%solve(t(X)%*%X)%*%t(X)
H1=X1%*%solve(t(X1)%*%X1)%*%t(X1)
num=t(y)%*%(H-H1)%*%y
denom=t(y)%*%(diag(nrow(H))-H1)%*%y/(n-k-1)
FTS=num/denom
c(FTS, qf(0.95, 1, n-k-1))
```

```
## [1] 5.821483 4.279344
```

$5.82 > 4.28$, so we reject the null hypothesis, Floors is (borderline) useful.

The p-value is

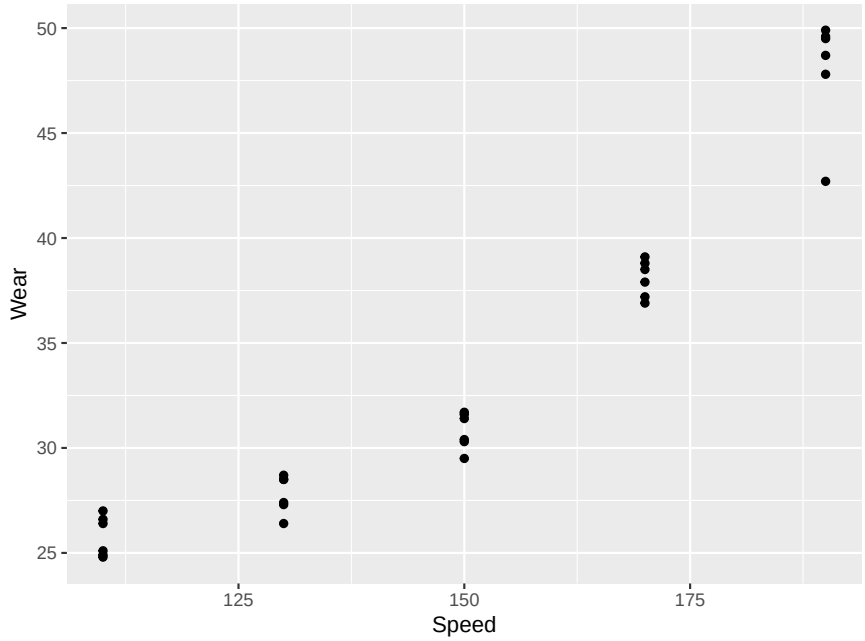
```
1-pf(FTS, 1, n-k-1)
```

```
##             Price
## Price 0.02420375
```

Comment using correlations with the response to see whether a variable is useful in a multiple regression problem turns out to be a bad idea. We will return to this important topic soon.

5.6.2.5 Example (6.6.9) We have the results from an experiment designed to determine how much the speed of a washing machine effects the wear on a new fabric. The machine was run at 5 different speeds (measured in rpm) and with six pieces of fabric each.

```
ggplot(data=fabricwear, aes(Speed, Wear)) +
  geom_point()
```



The scatterplot makes it clear that a linear model is not going to work, so we will try a polynomial model. Because the predictor variable has large values we first standardize it:

```
n=length(fabricwear$Speed)
x=(fabricwear$Speed-110)/(190-110)
X=cbind(1, x, x^2)
y=cbind(fabricwear$Wear)
k=ncol(X)
X1=X[, -k]
H=X%*%solve(t(X)%*%X)%*%t(X)
H1=X1%*%solve(t(X1)%*%X1)%*%t(X1)
num=t(y)%*%(H-H1)%*%y
denom=t(y)%*%(diag(nrow(H))-H1)%*%y/(n-k-1)
round(c(num/denom, qf(0.95, 1, n-k-1)), 2)
```

```
## [1] 19.56  4.23
```

$19.6 \geq 4.23$, and so we reject the null hypothesis, the quadratic term improves the fit significantly.

How about a cubic term?

```
X=cbind(1, x, x^2,x^3)
k=ncol(X)
```

```
X1=X[, -k]
H=X%*%solve(t(X)%*%X)%*%t(X)
H1=X1%*%solve(t(X1)%*%X1)%*%t(X1)
num=t(y)%*%(H-H1)%*%y
denom=t(y)%*%(diag(nrow(H))-H1)%*%y/(n-k-1)
round(c(num/denom, qf(0.95, 1, n-k-1)), 2)
```

```
## [1] 0.75 4.24
```

$0.75 < 4.24$, and so a cubic term is not needed.

Recall the definition of R^2:

$$R^2 = \frac{\hat{\boldsymbol{\beta}}'\boldsymbol{X}'\boldsymbol{y} - n\bar{y}^2}{\boldsymbol{y}'\boldsymbol{y} - n\bar{y}^2}$$

Now

5.6.2.6 Theorem (6.6.10)

i. when testing $H_0 : \boldsymbol{\beta}_1 = 0$ we have

$$F = \frac{R^2/k}{(1-R^2)/(n-k-1)}$$

ii. when testing $H_0 : \boldsymbol{\beta}_2 = 0$ we have

$$F = \frac{(R^2 - R_r^2)/h}{(1-R^2)/(n-k-1)}$$

where R_r^2 is is R^2 for the reduced model

proof direct calculation

5.6.3 Testing the General Linear Hypothesis

5.6.3.1 Definition (6.6.11) Let $\boldsymbol{C}$ be a $q \times (k+1)$ matrix of constants, then a test of a hypothesis of the form

$$H_0 : \boldsymbol{C}\boldsymbol{\beta} = 0$$

is called the *general linear hypothesis*

5.6.3.2 Example (6.6.12) The test for the full model corresponds to $\boldsymbol{C} = \begin{pmatrix} \mathbf{0} & \boldsymbol{I}_k \end{pmatrix}$.

The test for the a set of predictors corresponds to $\boldsymbol{C} = \begin{pmatrix} \mathbf{0} & \boldsymbol{I}_h \end{pmatrix}$.

We can also use this to test many other hypothesis:

Say have $\boldsymbol{\beta} = (\beta_0, \beta_1, .., \beta_4)'$ and we want to test

$$H_0 : \beta_2 - 2\beta_3 = \beta_2 + \beta_4 = 0$$

then we can write this as follows

$$H_0 : \begin{pmatrix} 0 & 0 & 1 & -2 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{pmatrix} = \mathbf{0}$$

5.6.3.3 Theorem (6.6.13) If $\boldsymbol{y} \sim N_n(\boldsymbol{X\beta}, \sigma^2\boldsymbol{I})$ and $\boldsymbol{C}$ be a $q \times (k+1)$ matrix of constants, then

i. $\boldsymbol{C\hat{\beta}} \sim N_q\left(\boldsymbol{C\beta}, \sigma^2\boldsymbol{C(X'X)^{-1}C'}\right)$

ii. $\text{SSH}/\sigma^2 = (\boldsymbol{C\hat{\beta}})'[\boldsymbol{C(X'X)^{-1}C'}]^{-1}\boldsymbol{C\hat{\beta}} \sim \chi^2(q, \lambda)$

where $\lambda = (\boldsymbol{C\beta})'[\boldsymbol{C(X'X)^{-1}C'}]^{-1}\boldsymbol{C\beta}/(2\sigma^2)$

iii. $\text{SSE}/\sigma^2 = \boldsymbol{y'[I - X(X'X)^{-1}X']y}/\sigma^2 \sim \chi^2(n-k-1$

iv. SSH and SSE are independent

proof omitted

5.6.3.4 Theorem (6.6.14) Say $\boldsymbol{y} \sim N_n(\boldsymbol{X\beta}, \sigma^2\boldsymbol{I})$ and $\boldsymbol{C}$ be a $q \times (k+1)$ matrix of constants. Let

$$F = \frac{\text{SSH}/q}{\text{SSE}/(n-k-1)}$$

i. If $H_0 : \boldsymbol{C\beta} = 0$ is false, then $F \sim F(q, n-k-1, \lambda)$, where $\lambda = (\boldsymbol{C\beta})'[\boldsymbol{C(X'X)^{-1}C'}]^{-1}\boldsymbol{C\beta}/(2\sigma^2)$

ii. If $H_0 : \boldsymbol{C\beta} = 0$ is true, then $F \sim F(q, n-k-1)$,

proof same as theorems above

If we want to test a single condition, for example $H_0 : \beta_1 = \beta_2$, we can use the test above but the calculations simplify. In this case $\boldsymbol{C}$ is a matrix of one row, so we can write the null hypothesis as $H_0 : \boldsymbol{a'\beta} = 0$, and the test statistic becomes

$$F = \frac{(\boldsymbol{a'}\hat{\boldsymbol{\beta}})^2}{s^2\boldsymbol{a'}(\boldsymbol{X'X})^{-1}\boldsymbol{a}}$$

where s^2=SSE/(n-k-1). Under the null hypothesis F now has an F distribution with 1 and n-k-1 degrees of freedom.

Also consider a test for a single variable, that is the test $H_0 : \beta_j = 0$. This can be done with the above and $\boldsymbol{a} = (0, .., 1, .., 0)'$, which leads to

$$F = \frac{\hat{\beta}_j^2}{s^2 g_{jj}}$$

where g_{jj} is the j^{th} diagonal element of $(\boldsymbol{X'X})^{-1}$.

Since an F distribution with 1 and n degrees of freedom is equal to a t distribution with n degrees of freedom we also have the following test: Let

$$T = \frac{\hat{\beta}_j}{s\sqrt{g_{jj}}}$$

and reject the null hypothesis if $|T| \geq t(\alpha/2, n-k-1)$

5.6.3.5 Example (6.6.15)

Let's repeat the test for Floors in the houseprice data.

```
A=as.matrix(houseprice)
n=nrow(A)
k=ncol(A)-1
X=cbind(1, A[, -1])
y=A[, 1, drop=FALSE]
G=solve(t(X)%*%X)
betahat= (solve(t(X)%*%X)%*%t(X))%*%y
betahat
```

```
##                 Price
##          -67.61983705
## Sqfeet     0.08570823
## Floors   -26.49305703
## Bedrooms  -9.28622097
## Baths     37.38067201
```

```
sse = t(y)%*%(diag(n)-X%*%G%*%t(X))%*%y/(n-k-1)
round(c(abs(betahat[3]/sqrt(sse*G[3, 3])), qt(0.975, n-k-1)),3)
```

```
## [1] 2.792 2.069
```

5.7 Simultaneous Inference, Multiple Testing

We will discuss this topic in the context of regression analysis, it is however much more general and arises in almost all areas of Statistics. The solutions discussed here will generally also work in other areas.

5.7.1 Bonferroni's Method

In example (6.5.15) we tested for the predictor Floors of the houseprice data. But why just Floors, let's test for all predictors. In fact, let's write an R routine that does this in general:

```
test.all.predictors=function(y, X, alpha=0.05) {
  n=nrow(X)
  k=ncol(X)
  X=cbind(1, X)
  y=cbind(y)
  G=solve(t(X)%*%X)
  betahat= (solve(t(X)%*%X)%*%t(X))%*%y
  sse = t(y)%*%(diag(n)-X%*%G%*%t(X))%*%y/(n-k-1)
  TS=c(abs(betahat)/sqrt(sse*diag(G)))[-1]
  names(TS)=colnames(X)[-1]
  list(TS=TS, crit=qt(1-alpha/2, n-k-1))
}
test.all.predictors(houseprice[, 1], as.matrix(houseprice[, -1]))
```

```
## $TS
##   Sqfeet   Floors Bedrooms    Baths
## 7.965553 2.791823 1.359653 3.047909
##
## $crit
## [1] 2.068658
```

and so we reject all the null hypotheses except the one for Bedrooms, all other predictors are useful.

There is however a problem. Let's consider a situation where in fact none of the predictors is useful, so all the null hypotheses are true. Let's do a little simulation to see what would happen:

```
B=1000
n=20
TS=matrix(0, B, 4)
x=seq(0, 1, length=n)
X=matrix(runif(4*n), n, 4)
for(i in 1:B) {
   y=rnorm(n) # y does not depend on any of the predictors
```

```
   TS[i, ]=test.all.predictors(y, X)$TS
}
```

```
crit=qt(0.975, n-4-1)
for(i in 1:4) print(sum(abs(TS[, i])> crit)/B)
```

5.7.1.1 Example (6.7.1)

```
## [1] 0.055
## [1] 0.053
## [1] 0.05
## [1] 0.046
```

and so this seems to work fine. However consider the following question: how often do we reject *at least* on of the four hypotheses? Well

```
TSmax=apply(TS, 1, max)
sum((TSmax>crit))/B
```

```
## [1] 0.172
```

In other words, if we applied this idea in practice, we would declare at least one predictor to be useful 17.2% of the time, even though none of them is.

The problem is a simple one: if you pick a card from a standard deck, the probability to pick the Ace of hearts is small (1/52) but if you pick a card, look at it, put the card back, shuffle, pick another card, look at it and so on, sooner or later you will find the Ace of Hearts!

5.7.1.2 Definition (6.7.2)

Say we carry out k hypothesis tests, each at the α level of significance. Then the probability to reject at least one null although they are all true is called the *familywise error rate*, denoted by α_f. α is then called the *comparisonwise error rate* α_c.

What can be said about α_f? Here is one case

5.7.1.3 Theorem (6.7.3)

If the k tests are independent we have $\alpha_f = 1 - (1 - \alpha_c)^k$

proof

Let the event *test i rejects null hypothesis* be denoted by T_i, then

$$
\begin{aligned}
&\alpha_f = P(\text{at least one test rejects null} \mid \text{all nulls are true}) = \\
&1 - P(\text{none of the tests rejects null} \mid \text{all nulls are true}) = \\
&1 - P(\cap_{i=1}^{k}\{T_i^c\}) = \\
&1 - \prod_{i=1}^{k} P(T_i^c) = \ \{\text{by independence}\} \\
&1 - P(T_1^c)^k = \\
&1 - (1 - P(T_1))^k = \\
&1 - (1 - \alpha_c)^k
\end{aligned}
$$

So in this case the solution would be easy: if we want a familywise error rate of α_f we should to each test at $1 - (1 - \alpha_c)^{1/k}$.

The problem is that in our case (and in almost all real live cases) the tests are not independent! In fact they are all using the same data set. But we do have:

5.7.1.4 Theorem (6.7.4)

$$\alpha_f \leq k\alpha_c$$

proof

$$
\begin{aligned}
&\alpha_f = P(\text{at least one test rejects null} \mid \text{all nulls are true}) = \\
&P(\cup_{i=1}^{k}\{T_i\}) \leq \{\text{Bonferroni Inequality}\} \\
&\sum_{i=1}^{k} P(T_i) = k\alpha_c
\end{aligned}
$$

So even if the tests are not independent, doing each test at an α/k level insures that α_f is at most α. This is called the *Bonferroni approach.*

On the other hand, if the tests are fairly dependent this method is quite *conservative*, that is the true α_f can be much smaller than α. This in turn will lead to a test with low power.

There are a number of modifications to the Bonferroni approach that have been proposed that are not as conservative, The best known of these is the method known as Holm-Bonferroni.

5.7.2 Scheffe's Method

A second approach to the multiple testing problem is due to Scheffe. Say we want to test

$$H_0 : \boldsymbol{a'\beta} = 0$$

Again the test is based on the test statistic

$$F = \frac{(\boldsymbol{a'}\hat{\boldsymbol{\beta}})^2}{s^2\boldsymbol{a'}(\boldsymbol{X'X})^{-1}\boldsymbol{a}}$$

The test rejects the null hypothesis if F is greater than some critical value. The idea now is to find critical values that work for all possible vectors $\boldsymbol{a}$. It is therefore necessary to find the distribution of $\max_{\boldsymbol{a}} F$.

5.7.2.1 Theorem (6.7.5)

i.

$$F_M = \max_{\boldsymbol{a}} \left[\frac{(\boldsymbol{a'}\hat{\boldsymbol{\beta}})^2}{s^2\boldsymbol{a'}(\boldsymbol{X'X})^{-1}\boldsymbol{a}} \right] = \frac{\hat{\boldsymbol{\beta}}\boldsymbol{X'X}\hat{\boldsymbol{\beta}}}{s^2}$$

ii. if $y \sim N_n(\boldsymbol{X\beta}, \sigma^2\boldsymbol{I})$, then $F_m/(k+1) \sim F(k+1, n-k-1)$

proof

$$\frac{\partial}{\partial a}\frac{(\boldsymbol{a'}\hat{\boldsymbol{\beta}})^2}{\boldsymbol{a'}(\boldsymbol{X'X})^{-1}\boldsymbol{a}} =$$
$$\frac{\frac{\partial}{\partial a}(\boldsymbol{a'}\hat{\boldsymbol{\beta}})^2(\boldsymbol{a'}(\boldsymbol{X'X})^{-1}\boldsymbol{a}) - (\boldsymbol{a'}\hat{\boldsymbol{\beta}})^2\frac{\partial}{\partial a}(\boldsymbol{a'}(\boldsymbol{X'X})^{-1}\boldsymbol{a})}{(\boldsymbol{a'}(\boldsymbol{X'X})^{-1}\boldsymbol{a})^2} = \boldsymbol{0}$$

now from (4.3.20) we have

$$\frac{\partial}{\partial a}(\boldsymbol{a'}\hat{\boldsymbol{\beta}})^2 = 2(\boldsymbol{a'}\hat{\boldsymbol{\beta}})\frac{\partial}{\partial a}(\boldsymbol{a'}\hat{\boldsymbol{\beta}}) = 2(\boldsymbol{a'}\hat{\boldsymbol{\beta}})\hat{\boldsymbol{\beta}}$$

and from (4.3.21) we have

$$\frac{\partial}{\partial a}(\boldsymbol{a'}(\boldsymbol{X'X})^{-1}\boldsymbol{a}) = 2(\boldsymbol{X'X})^{-1}\boldsymbol{a}$$

and so we have

$$(\boldsymbol{a'}\hat{\boldsymbol{\beta}})\hat{\boldsymbol{\beta}}(\boldsymbol{a'}(\boldsymbol{X'X})^{-1}\boldsymbol{a}) - (\boldsymbol{a'}\hat{\boldsymbol{\beta}})^2(\boldsymbol{X'X})^{-1}\boldsymbol{a} = \boldsymbol{0}$$

$\boldsymbol{a'}\hat{\boldsymbol{\beta}}$ is a number, so this is equivalent to

$$[\boldsymbol{a'}(\boldsymbol{X'X})^{-1}\boldsymbol{a}]\hat{\boldsymbol{\beta}} - \boldsymbol{a'}\hat{\boldsymbol{\beta}}(\boldsymbol{X'X})^{-1}\boldsymbol{a} = \boldsymbol{0}$$

$$\boldsymbol{a'}\hat{\boldsymbol{\beta}}(\boldsymbol{X'X})^{-1}\boldsymbol{a} = [\boldsymbol{a'}(\boldsymbol{X'X})^{-1}\boldsymbol{a}]\hat{\boldsymbol{\beta}}$$

$$(\boldsymbol{X'X})^{-1}\boldsymbol{a} = \frac{\boldsymbol{a}'(\boldsymbol{X'X})^{-1}\boldsymbol{a}}{\boldsymbol{a}'\hat{\boldsymbol{\beta}}}\hat{\boldsymbol{\beta}}$$

and finally

$$\boldsymbol{a} = \frac{\boldsymbol{a}'(\boldsymbol{X'X})^{-1}\boldsymbol{a}}{\boldsymbol{a}'\hat{\boldsymbol{\beta}}}\boldsymbol{X'X}\hat{\boldsymbol{\beta}} =: c\boldsymbol{X'X}\hat{\boldsymbol{\beta}}$$

substituting this back we get the result

ii. follows from (6.6.14) with $\boldsymbol{C} = \boldsymbol{I}_{k+1}$.

So, to test $H_0 : \boldsymbol{a'\beta} = 0$ for any $\boldsymbol{a}$ with $\alpha_f \leq \alpha$ find

$$F = \frac{(\boldsymbol{a'}\hat{\boldsymbol{\beta}})^2}{s^2\boldsymbol{a}'(\boldsymbol{X'X})^{-1}\boldsymbol{a}}$$

and reject the null hypothesis if $F \geq (k+1)F_{\alpha,k+1,n-k-1}$.

To test for individual $\beta's$ using Scheffe's method we use $\boldsymbol{a} = (0,..,1,0,..0)'$. Then as before the test reduces to the t test

$$t = \frac{\hat{\beta}_j}{s\sqrt{g_{jj}}}$$

and we reject the null hypothesis if $|t| \geq \sqrt{(k+1)F_{\alpha,k+1,n-k-1}}$.

If k is small when compared to n we have

$$t_{\alpha/2k,n-k-1} < \sqrt{(k+1)F_{\alpha,k+1,n-k-1}}$$

and so the Bonferroni test for individual predictors is more powerful than Scheffe's test.

5.7.2.2 Example (6.7.6)

Let's apply both methods to the houseprice data. We already found the test statistics:

```
TS.house=test.all.predictors(houseprice[, 1], as.matrix(houseprice[, -1]))$TS
```

Now for Bonferroni the critical value would be

```
qt(1-0.05/(2*4), 28-4-1)
```

```
## [1] 2.70972
```

```
TS.house>qt(1-0.05/(2*4), 28-4-1)
```

```
##   Sqfeet   Floors Bedrooms    Baths
##     TRUE     TRUE    FALSE     TRUE
```

and so we find Bedrooms to not be statistically significant

With Scheffe's method:

```
sqrt((4+1)*qf(1-0.05, 4+1, 28-4-1))
```

```
## [1] 3.63318
```

```
TS.house>sqrt((4+1)*qf(1-0.05, 4+1, 28-4-1))
```

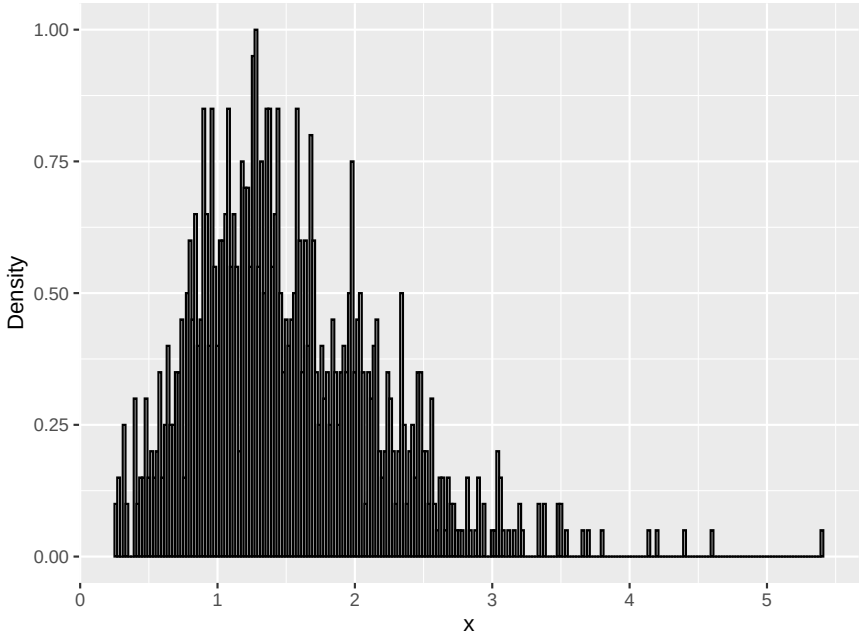
```
##   Sqfeet   Floors Bedrooms    Baths
##     TRUE    FALSE    FALSE    FALSE
```

and only SqFeet is found significant.

5.7.3 MC: Simulation based Methods

5.7.3.1 Example (6.7.7) let's return for a moment to example (6.7.1). There we ran a simulation and collected all the test statistics. Then we found the largest of each (in absolute value), that is the test statistic most likely to be larger than the critical value and leading to a rejection of a null. This is what they looked like:

```
df=data.frame(x=TSmax)
bw <- diff(range(x))/50
ggplot(df, aes(x)) +
  geom_histogram(aes(y = ..density..),
    color = "black",
    fill = "white",
    binwidth = bw) +
    labs(x = "x", y = "Density")
```



But if these were values of test statistics from real experiments, and knowing that in fact all null hypotheses were true, we could just find critical values from it:

```
crit=quantile(TSmax, 0.95)
TS.house>crit
```

```
##   Sqfeet   Floors Bedrooms    Baths
##     TRUE     TRUE    FALSE     TRUE
```

and using these critical values the test would have correct familywise error rate by design.

As discussed before, both Bonferroni and Scheffe are conservative methods, that is they usually lead to a α_f smaller than desired, and therefore also to a smaller power. The MC method however achieves α_f exactly.

This approach has another advantage: neither the Bonferroni method (as described above) nor Scheffe's method give us a p-value, which is what is generally quoted when doing a hypothesis test. The MC method can be used to do just that.

To see how let's return to the case were all the tests are independent. Let's denote by P_i the p-value of the i$^{\text{th}}$ test and by $P_{min} = \min\{P_i\}$, the smallest p-value which would correspond to the test most likely to be rejected. As always p values are random variables, and therefore so is P_{min}. What is its distribution? First recall that for any correct hypothesis test (for a continuous parameter) the distribution of the p-value has to be U[0,1], we have $P(P_i < x) = x$ for $0 < x < 1$.

$$
\begin{aligned}
&F_{P_{min}}(x) = P(P_{min} < x | \text{all nulls are true}) = \\
&1 - P(P_{min} > x) = \\
&1 - P(\cap_{i=1}^{k}\{P_i > x\}) = \\
&1 - \prod_{i=1}^{k} P(P_i > x) = \ \{\text{by independence}\} \\
&1 - P(P_1 > x)^k = \\
&1 - (1 - P(P_1 < x))^k = \\
&1 - (1 - x)^k
\end{aligned}
$$

Next we can make use of the following theorem from probability theory

5.7.3.2 Theorem (6.7.8) (Probability Integral Transform)

Let $X \sim F$ be a continuous random variable, then $F(X) \sim U[0,1]$.

proof see any probability theory textbook

So we find that

$$P^*_{min} = 1 - (1 - P_{min})^k \sim U[0,1]$$

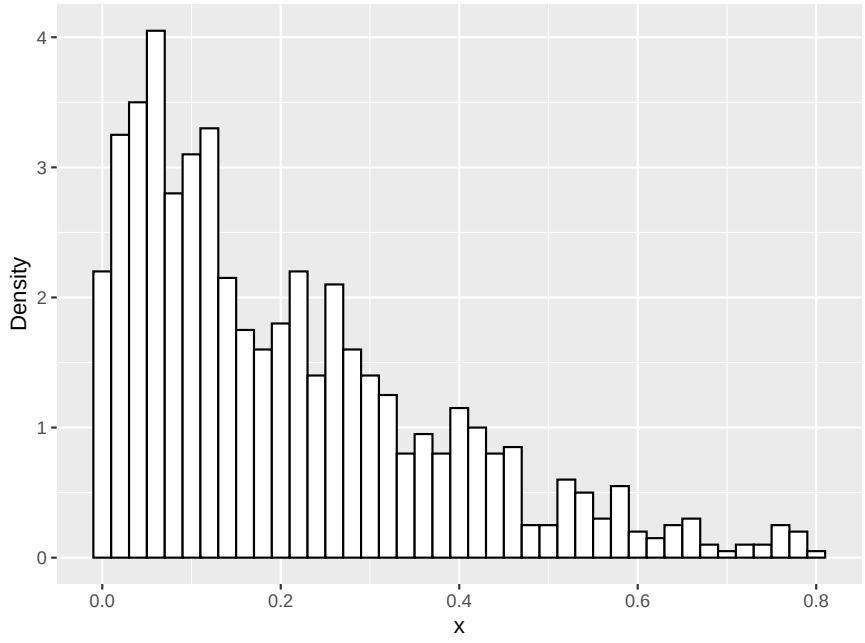
But in our case we do not have independence! So we do not know the distribution of P_{min}. However, we do have data from our simulation, and we can use that to find first observations from P_{min} and then its distribution:

- find values from P_{min}

```
pmin=rep(0, B)
tmp=rep(0, 4)
for(i in 1:B) {
   xsim=TS[sample(1:B, 1), ]
   for(k in 1:4) tmp[k] <- sum(xsim[k] < TS[ ,k])/B
    pmin[i] <- min(tmp)
}
```

Here is a histogram of the simulated P_{min} values:

```
df=data.frame(x=pmin)
bw <- 1/50
ggplot(df, aes(x)) +
  geom_histogram(aes(y = ..density..),
    color = "black",
    fill = "white",
    binwidth = bw) +
    labs(x = "x", y = "Density")
```



- find the distribution

This we can do by finding the *empirical distribution function*, defined by

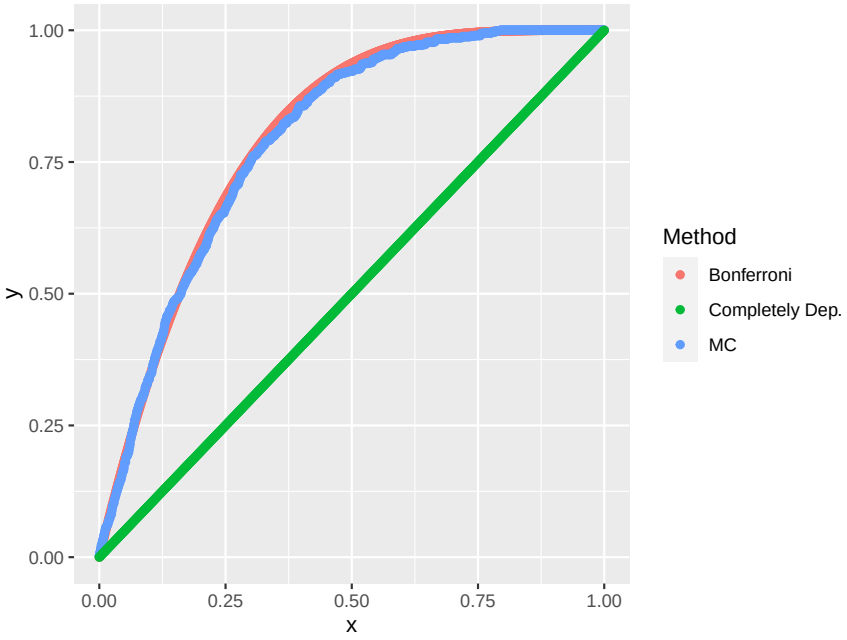
$$\hat{F}(x) = \frac{\#X's \leq x}{\#of X's}$$

.

```
x=seq(0, 1, length=1000)
y=x
for(i in 1:1000 ) {
  y[i]=sum(pmin<=x[i])/B
}
```

Here is a graph that shows this distribution function as well as the Bonferroni one and the distribution for completely dependent test, which is just F(x)=x:

```
df=data.frame(x=c(x, x, x),
              y=c(1-(1-x)^4, y, x),
              Method=rep(c("Bonferroni", "MC", "Completely Dep."), each=B))
ggplot(data=df, aes(x, y, color=Method)) +
  geom_point()
```



In this example the Bonferroni curve is the same as the MC curve because we did the simulation under the assumption of independence. In general the MC curve will be between the other two.

More details of this method can be found in Buja and Rolke, Calibration for Simultaneity: (Re) Sampling Methods for Simultaneous Inference with Applications to Function Estimation and Functional Data.

5.8 Confidence and Prediction Intervals

5.8.1 Confidence Intervals for β_j's

In (6.6.14) we found a test for $H_0 : \beta_j = 0$. Subtracting β_j and inverting this test we find

5.8.1.1 Theorem (6.8.1)

A $(1-\alpha)100\%$ confidence interval for β_j is given by

$$\hat{\beta}_j \pm t_{\alpha/2,n-k-1} s\sqrt{g_{jj}}$$

proof From In (6.6.14) we have

$$P\left[-t_{\alpha/2,n-k-1} < \frac{\hat{\beta}_j - \beta_j}{s\sqrt{g_{jj}}} < t_{\alpha/2,n-k-1}\right] = 1-\alpha$$

Solving the corresponding equations yields the result.

5.8.1.2 Example (6.8.2)

Let's find 90% confidence intervals for the houseprice data:

```
A=as.matrix(houseprice)
n=nrow(A)
k=ncol(A)-1
X=cbind(1, A[ ,-1])
y=A[, 1, drop=FALSE]
G=solve(t(X)%*%X)
betahat= (solve(t(X)%*%X)%*%t(X))%*%y
sse = t(y)%*%(diag(n)-X%*%G%*%t(X))%*%y/(n-k-1)
crit=qt(1-(1-0.9)/2, n-k-1)
Low=c(betahat-crit*sqrt(sse*diag(G)))
High=c(betahat+crit*sqrt(sse*diag(G)))
CI= round(cbind(Low, High), 3)
rownames(CI) = c("Intercept", colnames(A)[-1])
CI
```

```
##                Low    High
## Intercept -97.969 -37.270
## Sqfeet      0.067   0.104
## Floors    -42.757 -10.229
## Bedrooms  -20.992   2.419
## Baths      16.361  58.400
```

Warning again we have issue of simultaneous inference: the 90% coverage applies to each interval individually, not to the collection of intervals.

5.8.1.3 Theorem (6.8.3)

A $(1-\alpha)100\%$ confidence interval for $\boldsymbol{a'\beta} \neq 0$ is given by

$$\boldsymbol{a'\hat{\beta}} \pm t_{\alpha/2,n-k-1} s\sqrt{\boldsymbol{a'(X'X)^{-1}a}}$$

proof similar to proof of (6.8.1)

5.8.2 Confidence Intervals for $E[\boldsymbol{y}]$

Let $\boldsymbol{x}_0 = (1\ x_{01}\ ...\ x_{0k})'$ be some point, not necessarily a point in $\boldsymbol{X}$. A prediction of the response at that point is given by

$$\hat{y}_0 = \boldsymbol{x_0}\hat{\boldsymbol{\beta}}$$

5.8.2.1 Corollary (6.8.4)

A $(1-\alpha)100\%$ confidence interval for $E[\hat{y}]$ is given by

$$\boldsymbol{x_0'}\hat{\boldsymbol{\beta}} \pm t_{\alpha/2,n-k-1} s \sqrt{\boldsymbol{x_0'(X'X)^{-1}x_0}}$$

5.8.2.2 Example (6.8.5)

Let's find a 95% confidence interval for the average price of a two-story house with 2500 sqfeet, 3 bedrooms and 2 baths.

```
x0=rbind(1, 2500, 2, 3, 2)
crit=qt(1-(1-0.95)/2, n-k-1)
tmp=crit*sqrt(sse*t(x0)%*%solve(t(X)%*%X)%*%x0)
round(t(x0)%*%betahat+c(-1, 1)*tmp, 1)
```

```
## [1] 124.4 156.8
```

we can also use R:

```
fit=lm(Price~., data=houseprice)
newx=data.frame(Sqfeet=2500, Floors=2, Bedrooms=3, Baths=2)
predict(fit, newdata=newx, interval="confidence")
```

```
##        fit      lwr      upr
## 1 140.5673 124.3619 156.7727
```

5.8.2.3 Example (6.8.6)

Let's consider the case of simple regression. Here we have previously found

$$(\boldsymbol{X'X})^{-1} = \frac{1}{n\sum_i (x_i - \bar{x})^2} \begin{pmatrix} \sum_i x_i^2 & -\sum_i x_i \\ -\sum_i x_i & n \end{pmatrix}$$

and so if $\boldsymbol{x}_0 = (1\ x_0)'$

$$
\begin{aligned}
&\boldsymbol{x_0'}(\boldsymbol{X'X})^{-1}\boldsymbol{x_0} = \\
&\begin{pmatrix}1 & x_0\end{pmatrix}\frac{1}{n\sum_i(x_i-\bar{x})^2}\begin{pmatrix}\sum_i x_i^2 & -\sum_i x_i \\ -\sum_i x_i & n\end{pmatrix}\begin{pmatrix}1 \\ x_0\end{pmatrix} = \\
&\frac{1}{n\sum_i(x_i-\bar{x})^2}\begin{pmatrix}1 & x_0\end{pmatrix}\begin{pmatrix}\sum_i x_i^2 - x_0\sum_i x_i \\ -\sum_i x_i + nx_0\end{pmatrix} = \\
&\frac{1}{n\sum_i(x_i-\bar{x})^2}\left[\sum_i x_i^2 - x_0\sum_i x_i + nx_0^2 - x_0\sum_i x_i\right] = \\
&\frac{1}{n\sum_i x_i^2 - (\sum_i x_i)^2}\left[\sum_i x_i^2 - 2x_0\sum_i x_i + nx_0^2\right] = \\
&\frac{1}{n\sum_i(x_i-\bar{x})^2}\sum_i\left[x_i^2 - 2x_0x_i + x_0^2\right] = \\
&\frac{1}{n\sum_i(x_i-\bar{x})^2}\sum_i(x_i-x_0)^2 = \\
&\frac{1}{n\sum_i(x_i-\bar{x})^2}\sum_i(x_0-\bar{x}+\bar{x}-x_i)^2 = \\
&\frac{1}{n\sum_i(x_i-\bar{x})^2}\left(n(x_0-\bar{x})^2 + \sum_i(\bar{x}-x_i)^2\right) = \\
&\frac{1}{n} + \frac{(x_0-\bar{x})^2}{\sum_i(x_i-\bar{x})^2}
\end{aligned}
$$

This shows that the width of the interval increases with the distance of x_0 to $\bar{x}$.

5.8.3 Prediction Intervals for Future Observations

Another type of problem is to predict a future observation, not part of the current data set. This is called a *prediction problem*, and we use the term *prediction interval*. Because the future observation is independent from the data we find

$$
\begin{aligned}
&var(y_0 - \hat{y}_0) = \\
&var(\boldsymbol{x_0'}\boldsymbol{\beta} + \boldsymbol{\epsilon}_0 - \boldsymbol{x_0'}\hat{\boldsymbol{\beta}}_0) = \\
&var(\boldsymbol{\epsilon}_0 - \boldsymbol{x_0'}\hat{\boldsymbol{\beta}}_0) = \\
&var(\boldsymbol{\epsilon}_0) + var(\boldsymbol{x_0'}\hat{\boldsymbol{\beta}}_0) = \\
&\sigma^2 + \sigma^2\boldsymbol{x_0'}(\boldsymbol{X'X})^{-1}\boldsymbol{x_0} = \\
&\sigma^2\left[1 + \boldsymbol{x_0'}(\boldsymbol{X'X})^{-1}\boldsymbol{x_0}\right]
\end{aligned}
$$

5.8.3.1 Theorem (6.8.7) A $(1-\alpha)100\%$ prediction interval of y at the point $\boldsymbol{x}_0$ is given by

$$
\boldsymbol{x_0'}\hat{\boldsymbol{\beta}} \pm t_{\alpha/2,n-k-1}s\sqrt{1 + \boldsymbol{x_0'}(\boldsymbol{X'X})^{-1}\boldsymbol{x_0}}
$$

proof omitted

5.8.3.2 Example (6.8.8)

Let's find a 95% prediction interval for the of a two-story house with 2500 sqfeet, 3 bedrooms and 2 baths.

```
x0=rbind(1, 2500, 2, 3, 2)
crit=qt(1-(1-0.95)/2, n-k-1)
tmp=crit*sqrt(sse*(1+t(x0)%*%solve(t(X)%*%X)%*%x0))
round(t(x0)%*%betahat+c(-1, 1)*tmp, 1)
```

```
## [1] 107.9 173.2
```

or

```
fit=lm(Price~., data=houseprice)
newx=data.frame(Sqfeet=2500, Floors=2, Bedrooms=3, Baths=2)
predict(fit, newdata=newx, interval="predict")
```

```
##        fit      lwr      upr
## 1 140.5673 107.9064 173.2282
```

5.9 Regression Diagnostics

Recall that all our calculations have been based on a number of assumptions, namely

1. $E[\boldsymbol{y}] = \boldsymbol{X'\beta}$ (model is correct)
2. $var(\epsilon_i) = \sigma^2$ (equal variance, *homoscadasticity*)
3. $cov(\epsilon_i, \epsilon_j) = 0$ (independence)
4. $\boldsymbol{y} \sim N_n(\boldsymbol{X'\beta}, \sigma^2\boldsymbol{V})$ (normal residuals)

some or all of these assumptions will have to hold for our analysis, so the question arises, how does one check them? This is known as *regression diagnostics*.

5.9.1 Residuals

5.9.1.1 Definition (6.9.1)

- The *residuals* are defined as

$$\hat{\epsilon}_i = y_i - \hat{y}_i$$

- the *hat matrix* is defined by

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}} = \boldsymbol{X}(\boldsymbol{X'X})^{-1}\boldsymbol{X'y} = \boldsymbol{Hy}$$

so

$$\boldsymbol{H} = \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'$$

Note

$$\boldsymbol{H}\boldsymbol{X} = \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{X} = \boldsymbol{X}$$

and

$$\hat{\boldsymbol{\epsilon}} = (\boldsymbol{I} - \boldsymbol{H})\boldsymbol{y} = (\boldsymbol{I} - \boldsymbol{H})\boldsymbol{\epsilon}$$

5.9.1.2 Theorem (6.9.2)

If $E[\boldsymbol{y}] = \boldsymbol{X}'\boldsymbol{\beta}$ and $cov(\boldsymbol{y}) = \sigma^2\boldsymbol{I}$, then

i. $E[\hat{\boldsymbol{\epsilon}}] = \boldsymbol{0}$

ii. $cov(\hat{\boldsymbol{\epsilon}}) = \sigma^2(\boldsymbol{I} - \boldsymbol{H})$

iii. $cov(\hat{\boldsymbol{\epsilon}}, \boldsymbol{y}) = \sigma^2(\boldsymbol{I} - \boldsymbol{H})$

iv. $cov(\hat{\boldsymbol{\epsilon}}, \hat{\boldsymbol{y}}) = \boldsymbol{O}$

v. $\bar{\hat{\epsilon}} = 0$

vi. $\hat{\boldsymbol{\epsilon}}'\boldsymbol{y} = \text{SSE} = \boldsymbol{y}'(\boldsymbol{I} - \boldsymbol{H}\boldsymbol{y}$

vii. $\hat{\boldsymbol{\epsilon}}'\hat{\boldsymbol{y}} = \boldsymbol{0}$

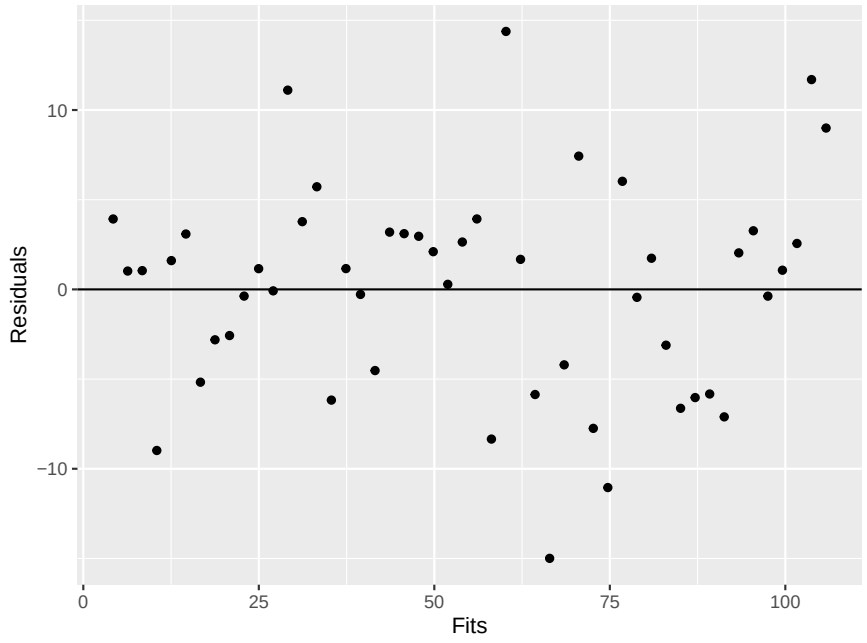
viii. $\hat{\boldsymbol{\epsilon}}'\boldsymbol{X} = \boldsymbol{0}'$

proof straightforward calculations

From iv we see that the residuals and the fitted values should be uncorrelated. This allows a check of the model with the *residual vs fits* plot:

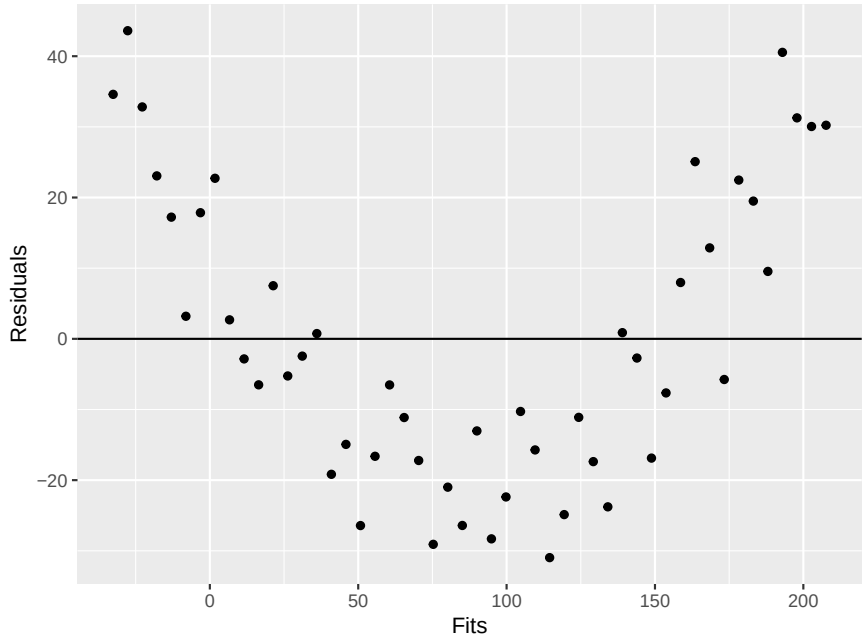
5.9.1.3 Example (6.9.3) Linear model is good:

```
x <- 1:50
y <- 5 + 2*x + rnorm(50, 0, 5)
fit <- lm(y~x)
df <- data.frame(Fits=fitted(fit),
                 Residuals=residuals(fit))
ggplot(data=df, aes(Fits, Residuals)) +
            geom_point() +
            geom_hline(yintercept = 0)
```



Linear model is bad:

```
x <- 1:50
y <- 0.1*x^2+rnorm(50, 0, 10)
fit <- lm(y~x)
df <- data.frame(Fits=fitted(fit),
                 Residuals=residuals(fit))
ggplot(data=df, aes(Fits, Residuals)) +
            geom_point() +
            geom_hline(yintercept = 0)
```



The U shaped pattern in the residual vs. fits plot is a very common one if the linear model is bad.

5.9.2 The Hat Matrix

5.9.2.1 Theorem (6.9.4) The hat matrix $\boldsymbol{H}$ is symmetric and idempotent

proof

$$\begin{aligned}&\boldsymbol{H}' = \left[\{\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\}\boldsymbol{X}'\right]' = \\ &\boldsymbol{X}\{\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\}' = \\ &\boldsymbol{X}'[(\boldsymbol{X}'\boldsymbol{X})^{-1}]'\boldsymbol{X}' = \\ &\boldsymbol{X}'(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}' = \boldsymbol{H}\end{aligned}$$

$$\begin{aligned}&\boldsymbol{H}\boldsymbol{H} = \\ &\left[\{\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\}\boldsymbol{X}'\right]\left[\{\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\}\boldsymbol{X}'\right]' = \\ &\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}(\boldsymbol{X}'\boldsymbol{X})(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}' = \\ &\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}' = \boldsymbol{H}\end{aligned}$$

5.9.2.2 Theorem (6.9.5) for the centered model we have

$$\boldsymbol{H} = \frac{1}{n}\boldsymbol{J} + \boldsymbol{H}_c = \frac{1}{n}\boldsymbol{J} + \boldsymbol{X_c}(\boldsymbol{X_c}'\boldsymbol{X_c})^{-1}\}\boldsymbol{X_c}'$$

proof straightforward

5.9.2.3 Theorem (6.9.6)

i. $\frac{1}{n} \leq h_{ii} \leq 1$, i=1,2,..,n

ii. $-\frac{1}{2} \leq h_{ij} \leq \frac{1}{2}; i \neq j$

iii. $tr(\boldsymbol{H}) = \sum h_{ii} = k + 1$

proof

i. $\boldsymbol{X_c}'\boldsymbol{X_c}$ is positive definite, so it's diagonal elements are positive. Therefore

$$h_{ii} = [\frac{1}{n}\boldsymbol{J} + \boldsymbol{X_c}(\boldsymbol{X_c}'\boldsymbol{X_c})^{-1}\}\boldsymbol{X_c}']_{ii} > \frac{1}{n}\boldsymbol{J}_{ii} = \frac{1}{n}$$

$\boldsymbol{H}$ is symmetric and idempotent, so

$$h_{ii} = \boldsymbol{h'h} = \sum_j h_{ij}^2 = h_{ii}^2 \sum_j h_{j\neq i}^2$$
$$1 = h_{ii} + \sum_{j\neq i} \frac{h_{ij}^2}{h_{ii}}$$

but $h_{ii} > 0$, so $h_{ii} < 1$

ii. using the above we can write

$$h_{ii} = h_{ii}^2 + h_{ik}^2 + \sum_{j\neq ik} h_{ij}^2$$
$$h_{ii} - h_{ii}^2 = h_{ik}^2 + \sum_{j\neq ik} h_{ij}^2$$
$$h_{ik}^2 \leq h_{ii} - h_{ii}^2 \leq \frac{1}{4}$$

iii. omitted

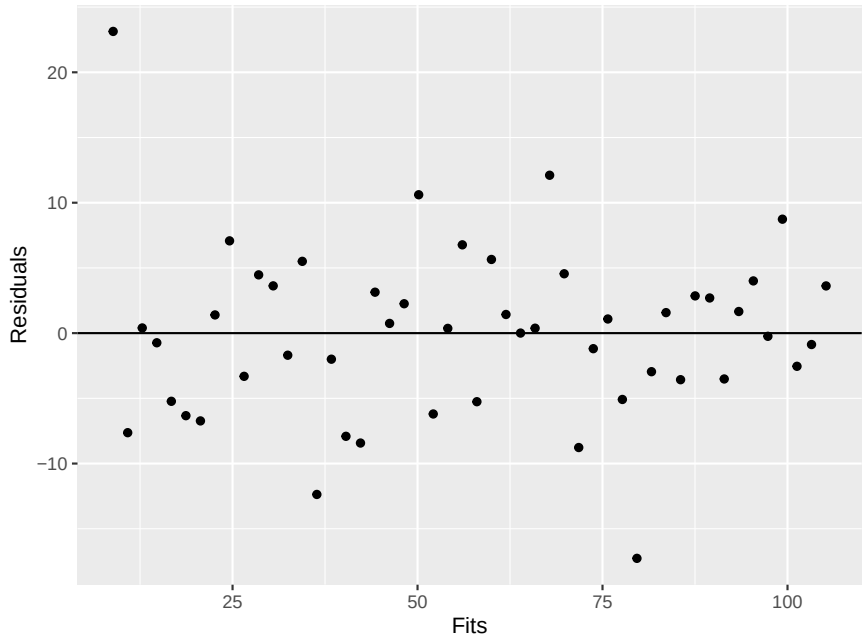
5.9.3 Outliers

5.9.3.1 Definition (6.9.7) An *outlier* is any observation that is unusual given the model.

5.9.3.2 Example (6.9.8) say $X \sim N(0, 1)$, then any observations outside [-3, 3] is an outlier.

A graphical check for outliers is again the residual vs fits plot:

```
x <- 1:50
y <- 5 + 2*x + rnorm(50, 0, 5)
y[1] = 5 + 2*x[1] + 25
fit <- lm(y~x)
df <- data.frame(Fits=fitted(fit),
                 Residuals=residuals(fit))
ggplot(data=df, aes(Fits, Residuals)) +
            geom_point() +
            geom_hline(yintercept = 0)
```



Note that by (6.9.2ii) $var(\hat{\epsilon}_i) = \sigma^2(1 - h_{ii})$ and by (6.9.6i) $h_{ii}) \leq 1$, therefore $var(\hat{\epsilon}_i)$ will be small if $h_{ii})$ is near 1. Moreover it can be shown that $h_{ii})$ will be large for observations far from $\bar{x}_i$. These issues lead to the

5.9.3.3 Definition (6.9.9)

The *studentized residuals* are defined by

$$r_i = \frac{\hat{\epsilon}_i}{s\sqrt{1 - h_{ii}}}$$

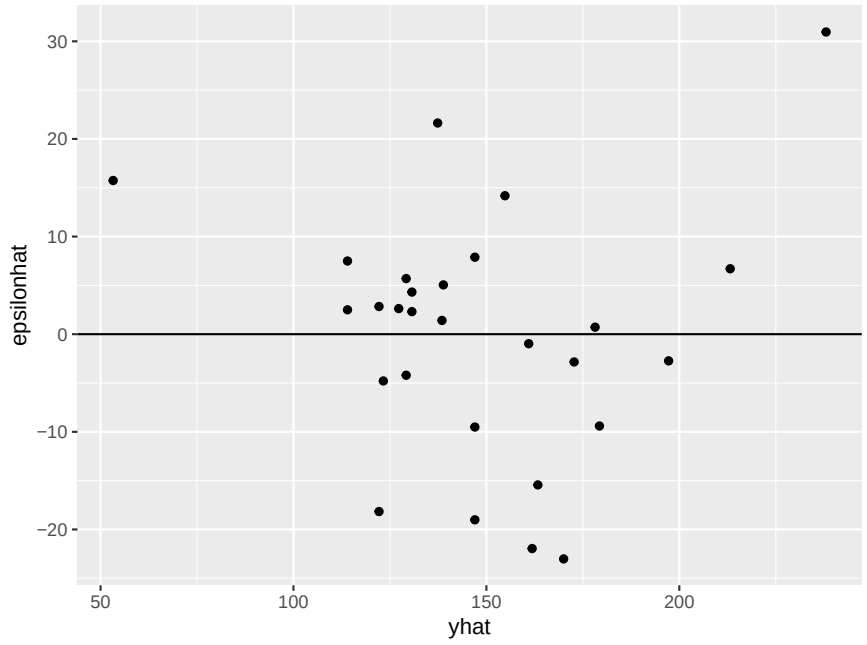
The idea is to eliminate the effect of observations far from their means to have large variances.

5.9.3.4 Example (6.9.10)

Let's consider the houseprice data:

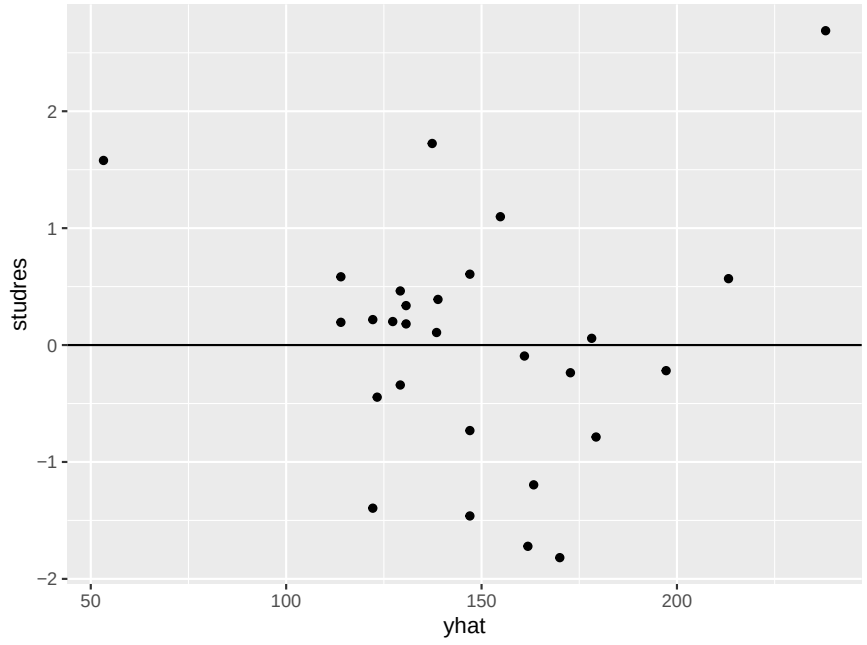
```
A=as.matrix(houseprice)
n=nrow(A)
k=ncol(A)-1
X=cbind(1, A[, -1])
y=A[, 1, drop=FALSE]
G=solve(t(X)%*%X)
betahat= G%*%t(X)%*%y
yhat=X%*%betahat
sse = t(y)%*%(diag(n)-X%*%G%*%t(X))%*%y/(n-k-1)
epsilonhat=y-yhat
H=X%*%G%*%t(X)
residual = epsilonhat-yhat
df=data.frame(residuals=residual, yhat=yhat)
```

```
ggplot(data=df, aes(yhat, epsilonhat)) +
  geom_point() +
  geom_abline(slope=0,intercept=0)
```



there are two potential outlier, one on the left and one in the upper right corner.

```
df$studres = epsilonhat/sqrt(sse*(1-diag(H)))
ggplot(data=df, aes(yhat, studres)) +
  geom_point() +
  geom_abline(slope=0,intercept=0)
```



which looks the same here.

5.9.4 Leverage or Influential Observations

5.9.4.1 Definition (6.9.11) The *Cook's distance* is defined by

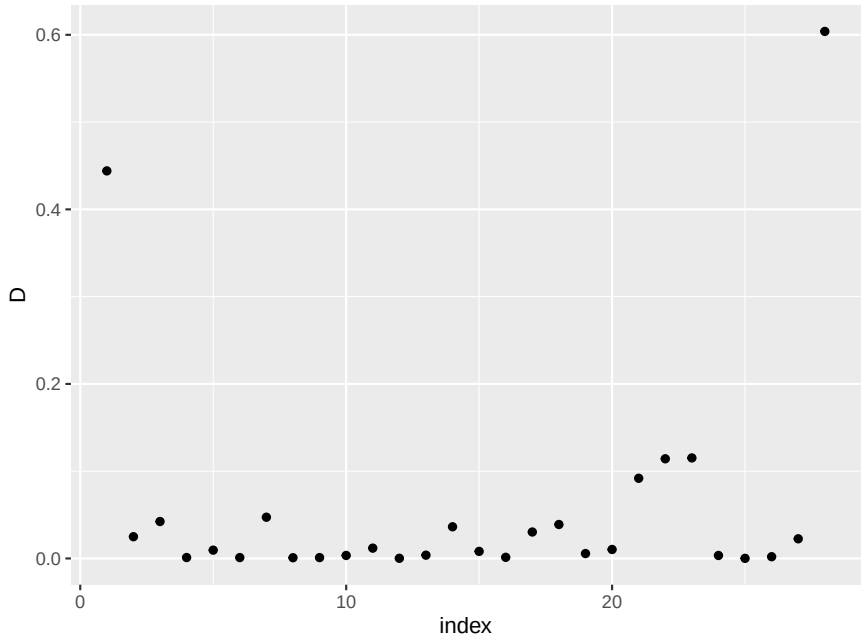
$$D_i = \frac{(\hat{\boldsymbol{\beta}}_{(i)} - \hat{\boldsymbol{\beta}})' \boldsymbol{X}' \boldsymbol{X} (\hat{\boldsymbol{\beta}}_{(i)} - \hat{\boldsymbol{\beta}})}{(k+1)s^2}$$

where $\hat{\boldsymbol{\beta}}_{(i)}$ is the least squares estimator with the i$^{\text{th}}$ observation.

The distribution of D_i is known, but as a simple rule-of-thumb any observation with $D_i > 1$ should be considered influential.

5.9.4.2 Example (6.9.12) Let's find and plot the Cook distances for the houseprice data:

```
A=as.matrix(houseprice)
n=nrow(A)
k=ncol(A)-1
X=cbind(1, A[, -1])
y=A[, 1, drop=FALSE]
G=solve(t(X)%*%X)
betaihat= G%*%t(X)%*%y
sse = t(y)%*%(diag(n)-X%*%G%*%t(X))%*%y/(n-k-1)
D=rep(n, 0)
for(i in 1:n) {
  Xtmp=X[-i, ]
  ytmp=y[-i, ]
  betaihat= solve(t(Xtmp)%*%Xtmp)%*%t(Xtmp)%*%ytmp
  D[i]=t(betaihat-betahat)%*%t(X)%*%X%*%(betaihat-betahat)/(k+1)/sse
}
df=data.frame(index=1:n, D=D)
ggplot(data=df, aes(index, D)) +
  geom_point() +
  geom_abline(slope=0, intercept = 1)
```

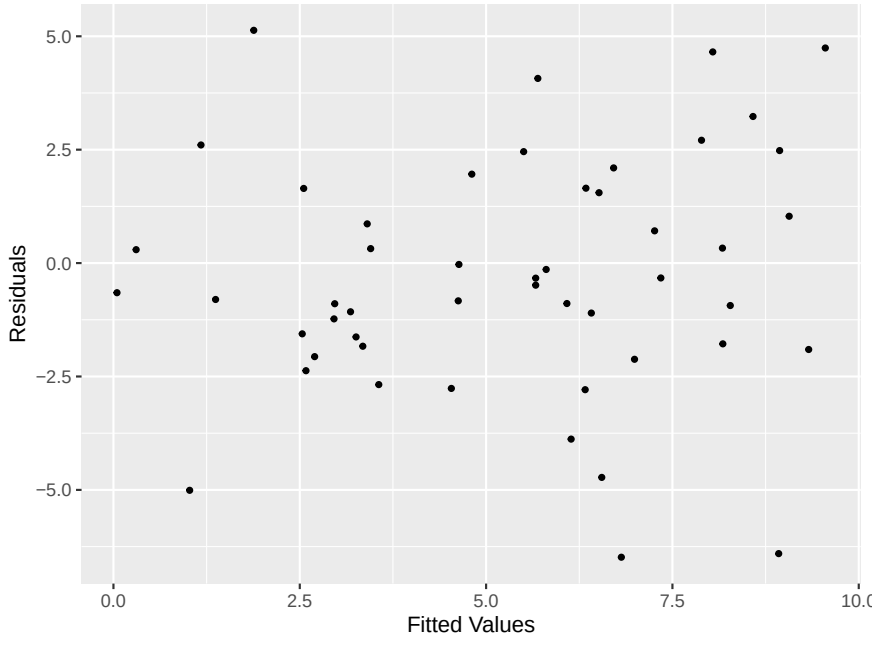


so the two observations noted earlier as potential outliers do not have Cook's distance over 1, and therefore are ok.

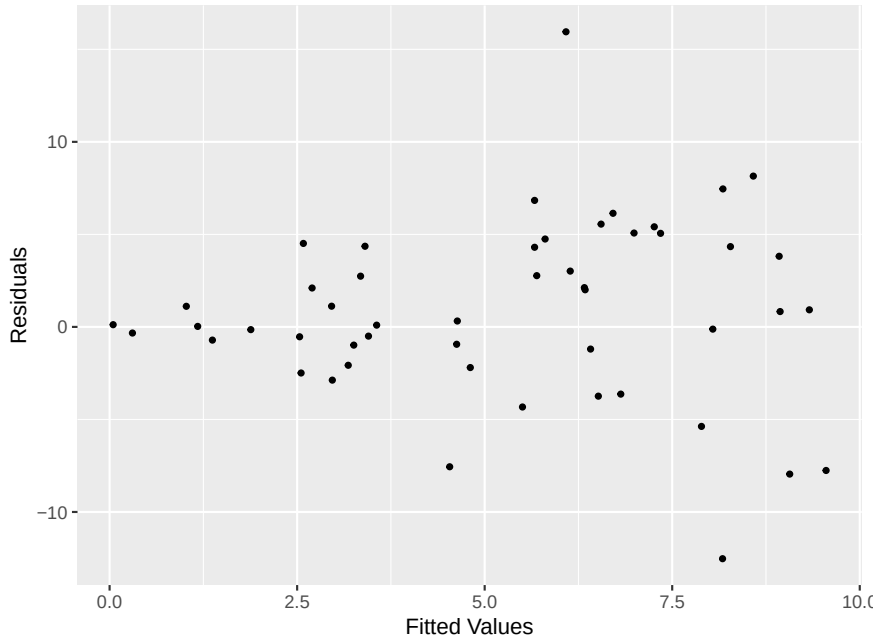
5.9.5 Equal Variance

We will again draw the Residuals vs. Fits plot and check whether the *variance (or spread) of the dots changes as you go along the x axis.*

5.9.5.1 Example (6.9.13) Equal Variance ok:



Equal Variance not ok:



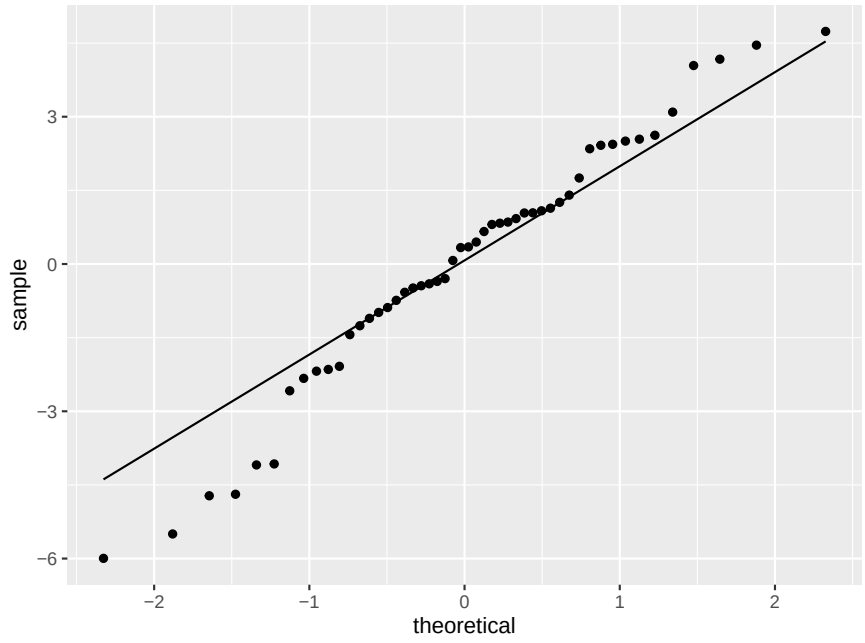
This can be a tricky one to decide, especially if there are few observations.

5.9.6 Normal Distribution

To check the normal assumption we can draw the *normal plot of residuals*. If the assumption is ok the dot's will follow along a straight line.

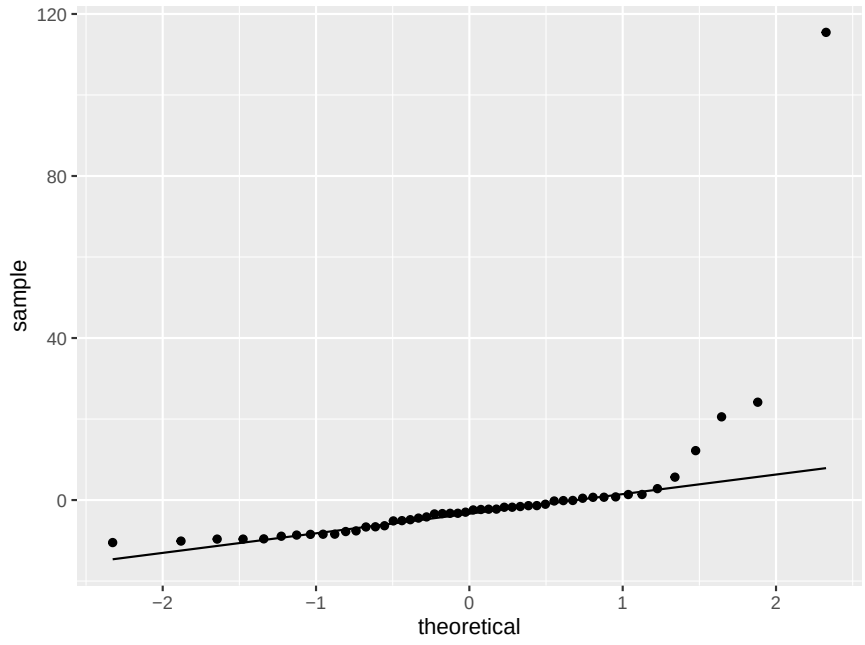
Normal assumption OK:

```
x <- runif(50, 0, 10)
y1 <- 1+2*x+rnorm(50, 0, 3)
fit1=lm(y1~x)
df <- data.frame(Residuals=resid(fit1),
            Fits = fitted(fit1))
ggplot(data=df, aes(sample=Residuals)) +
           geom_qq() + geom_qq_line()
```



Normal assumption not OK:

```
y2 <- 1+2*x+rt(50, 1)
fit2=lm(y2~x)
df <- data.frame(Residuals=resid(fit2),
            Fits = fitted(fit2))
ggplot(data=df, aes(sample=Residuals)) +
          geom_qq() + geom_qq_line()
```



In addition one can do formal hypothesis tests, so-called goodness-of-fit tests. A good test for normality is the *Shapiro-Wilks test*:

```
shapiro.test(resid(fit1))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(fit1)
## W = 0.97152, p-value = 0.2666
```

```
shapiro.test(resid(fit2))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(fit2)
## W = 0.40013, p-value = 5.303e-13
```

5.10 Random Predictors

Up to now we have assumed that the design matrix $\boldsymbol{X}$ was fixed. Often however the values of the predictor variables are themselves random. In fact, that was the case in the wine and houseprice examples. It turns out that treating a random X case as if the X's were fixed is acceptable in many cases, in fact we will see that most of the results we have obtained so far still hold. Moreover, it often makes sense to analyze a regression problem conditional on the the X's, in which case the predictors are treated as fixed although they originated in some random fashion.

If the x's are to be treated as random, we need to consider

$$cov \begin{pmatrix} y \\ x_1 \\ \vdots \\ x_k \end{pmatrix} = \boldsymbol{\Sigma}$$

5.10.1 Multivariate Normal Regression Model

In this section we will assume that $\boldsymbol{y}$ and $\boldsymbol{X}$ have a joint multivariate normal distribution with mean vector

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_y \\ \boldsymbol{\mu}_x \end{pmatrix}$$

and covariance matrix

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_{yy} & \boldsymbol{\sigma}'_{yx} \\ \boldsymbol{\sigma}_{yx} & \boldsymbol{\sigma}_{xx} \end{pmatrix}$$

By (5.2.13) we have

$$E[y|\boldsymbol{x}] = \mu_y + \boldsymbol{\sigma}'_{yx}\boldsymbol{\Sigma}_{xx}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_x) = \beta_0 + \boldsymbol{\beta}'_{\mathbf{1}}\boldsymbol{x}$$

where

$$\beta_0 = \mu_y - \boldsymbol{\sigma}'_{yx}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_x$$

$$\boldsymbol{\beta}'_{\mathbf{1}} = \boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\sigma}_{yx}$$

Also from (5.2.13) we have

$$var(y|\boldsymbol{x}) = \boldsymbol{\sigma}_{yy} - \boldsymbol{\sigma}'_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\sigma}_{yx} = \sigma^2$$

Note that under this model y is not only linear in $\boldsymbol{\beta}$ but also linear in the x's, so this does not allow for a model (say) quadratic in x.

5.10.2 Estimation and Testing

5.10.2.1 Theorem (6.10.1) Under the multivariate normal model the maximum likelihood estimators are given by

$$\hat{\boldsymbol{\mu}} = \begin{pmatrix} \bar{y} \\ \bar{\boldsymbol{x}} \end{pmatrix}$$

and

$$\hat{\boldsymbol{\Sigma}} = \frac{n-1}{n}\begin{pmatrix} s_{yy} & \boldsymbol{s}'_{yx} \\ \boldsymbol{s}_{yx} & \boldsymbol{s}_{xx} \end{pmatrix}$$

proof omitted

5.10.2.2 Theorem (6.10.2) (Invariance of MLEs)

If $\hat{\boldsymbol{\theta}}$ is the mle of $\boldsymbol{\theta}$, then $g(\hat{\boldsymbol{\theta}})$ is the mle of $g(\boldsymbol{\theta})$

proof any book on theory of statistics

5.10.2.3 Theorem (6.10.3) The mle's of β_0, $\boldsymbol{\beta}_1$ and σ^2 are given by

$$\hat{\beta}_0 = \bar{y} - \boldsymbol{s}'_{yx}\boldsymbol{S}^{-1}\bar{\boldsymbol{x}}$$

$$\hat{\boldsymbol{\beta}}'_{\mathbf{1}} = \boldsymbol{S}_{xx}^{-1}\boldsymbol{s}_{yx}$$

and

$$\hat{\sigma}^2 = \frac{n-1}{n}\left(s_{yy} - \boldsymbol{s}'_{yx}\boldsymbol{S}_{xx}^{-1}\boldsymbol{s}_{yx}\right)$$

proof follows from the invariance of mle's

Notice that these estimators are the same as the least-squares estimators in the fixed x case. However, their distributions are no longer multivariate normal but multivariate t.

The F tests discussed in section 6.6 work equally well in the random x case since they are based on the conditional distributions.

5.10.3 Standardized Regression Coefficients

The sample correlation matrix can be written as

$$\boldsymbol{R} = \begin{pmatrix} 1 & r_{y1} & \ldots & r_{yk} \\ r_{1y} & 1 & \ldots & r_{1k} \\ \vdots & \vdots & & \ldots \quad \vdots \\ r_{ky} & r_{k1} & \ldots & 1 \end{pmatrix} = \begin{pmatrix} 1 & \boldsymbol{r}'_{yx} \\ \boldsymbol{r}_{yx} & \boldsymbol{R}_{xx} \end{pmatrix}$$

here (for example)

$$r_{y1} = \frac{s_{y2}}{\sqrt{s_y^2 s_2^2}} = \frac{\sum(y_i - \bar{y})(x_{i2} - \bar{x}_2)}{\sqrt{\sum(y_i - \bar{y})^2 \sum(x_{i2} - \bar{x}_2)^2}}$$

and

$$r_{12} = \frac{s_{12}}{\sqrt{s_1^2 s_2^2}} = \frac{\sum(x_{i1} - \bar{x}_1)(x_{i2} - \bar{x}_2)}{\sqrt{\sum(x_{i1} - \bar{x}_1)^2 \sum(x_{i2} - \bar{x}_2)^2}}$$

we have $\boldsymbol{S = DRD}$, where $\boldsymbol{D} = [diag(S)]^{1/2}$

$$\boldsymbol{D} = \begin{pmatrix} s_y & 0 & \ldots & 0 \\ 0 & \sqrt{s_{11}} & \ldots & 0 \\ \vdots & \vdots & & \ldots \quad \vdots \\ 0 & 0 & \ldots & \sqrt{s_{kk}} \end{pmatrix} = \begin{pmatrix} s_y & \mathbf{0}' \\ \mathbf{0} & \boldsymbol{D}_{xx} \end{pmatrix}$$

and we can write

$$\hat{\boldsymbol{\beta}}_1 = s_y \boldsymbol{D}^{-1} \boldsymbol{R}_{xx}^{-1} \boldsymbol{r}_{yx}$$

5.10.3.1 Definition (6.10.4) Let $\boldsymbol{x}$ be sample. The the *z scores* are defined as

$$\boldsymbol{z} = \frac{\boldsymbol{x} - \bar{x}}{s_x}$$

Recall that the model in centered form is

$$\hat{y}_i = \bar{y} + \sum_i \hat{\beta}_j (x_{ij} - \bar{x}_j)$$

and so

$$\frac{\hat{y}_i - \bar{y}}{s_y} = \sum_i \frac{s_j}{s_y} \hat{\beta}_j \left(\frac{x_{ij} - \bar{x}_j}{s_j} \right)$$

5.10.3.2 Definition (6.10.5) The coefficients $\hat{\beta}_j^* = \frac{s_j}{s_y} \hat{\beta}_j$ are called the *beta weights* or *beta coefficients*. They can also be found as

$$\hat{\boldsymbol{\beta}}_1^* = \frac{1}{s_y} \boldsymbol{D}_x \hat{\boldsymbol{\beta}}_1 = \boldsymbol{R}_{xx}^{-1} \boldsymbol{r}_{yx}$$

5.10.3.3 Example (6.10.6) For the houseprice data we find

```
A=as.matrix(houseprice)
y=A[, 1, drop=FALSE]
ybar=mean(y)
X=cbind(1, A[, -1])
xbar=apply(X[, -1], 2, mean)
sxx=cov(A[, -1])
syx=cov(A)[-1, 1]
betahat= solve(t(X)%*%X)%*%t(X)%*%y
round(c(betahat), 3)
```

```
## [1] -67.620   0.086 -26.493  -9.286  37.381
```

```
round(c(solve(sxx)%*%cbind(syx)), 3)
```

```
## [1]   0.086 -26.493  -9.286  37.381
```

```
round(ybar-rbind(syx)%*%solve(sxx)%*%xbar, 3)
```

```
##       [,1]
## syx -67.62
```

5.10.4 R^2

5.10.4.1 Definition (6.10.7) The *sample coefficient of determination* is defined by

$$R^2 = \frac{\boldsymbol{s}'_{yx}\boldsymbol{S}_{xx}^{-1}\boldsymbol{s}_{yx}}{s_{yy}}$$

5.10.4.2 Example (6.10.8) For the houseprice data we find

```
A=as.matrix(houseprice)
y=A[, 1, drop=FALSE]
sxx=cov(A[, -1])
tmp=cov(A)[, 1]
syy=tmp[1]
syx=tmp[-1]
round(rbind(syx)%*%solve(sxx)%*%cbind(syx)/syy, 3)
```

```
##       syx
## syx 0.886
```

```
summary(lm(Price~., data=houseprice))
```

```
##
## Call:
## lm(formula = Price ~ ., data = houseprice)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -23.018  -5.943   1.860   5.947  30.955
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -67.61984   17.70818  -3.819 0.000882
## Sqfeet        0.08571    0.01076   7.966 4.62e-08
## Floors      -26.49306    9.48952  -2.792 0.010363
## Bedrooms     -9.28622    6.82985  -1.360 0.187121
## Baths        37.38067   12.26436   3.048 0.005709
##
## Residual standard error: 13.71 on 23 degrees of freedom
## Multiple R-squared:  0.8862, Adjusted R-squared:  0.8665
## F-statistic:  44.8 on 4 and 23 DF,  p-value: 1.558e-10
```

5.11 Bayesian Inference for Regression

We will now analyze a multiple regression problem using Bayesian analysis. The models are often expressed in terms of the *precision* τ instead of the variance, where

$$\tau = \frac{1}{\sigma^2}$$

A multiple regression model with k predictor variables has k+2 parameters, all of whom need a prior distribution. As an example it might look like this:

$$\begin{aligned}
&\boldsymbol{y}|\boldsymbol{\beta},\boldsymbol{\tau} \sim N_n(\boldsymbol{X\beta}, \frac{1}{\tau}\boldsymbol{I}) \\
&\boldsymbol{\beta}|\boldsymbol{\tau} \sim N_{k+1}(\boldsymbol{\phi}, \frac{1}{\tau}\boldsymbol{V}) \\
&\tau \sim Gamma(\alpha, \delta)
\end{aligned}$$

We will assume that $\boldsymbol{\phi}, \boldsymbol{V}, \boldsymbol{\alpha}, \boldsymbol{\delta}$ are known, but in real live they are often treated as *hyperparameters* with their own distributions in what are called *hierarchical models*. $\boldsymbol{\phi}, \boldsymbol{V}, \boldsymbol{\alpha}, \boldsymbol{\delta}$ are chosen so as to express some prior knowledge of the experiment.

5.11.0.1 Theorem (6.11.1) The priors in the model above are *conjugate priors*, that is the posterior distribution is again a normal distribution.

proof

Ignoring all constants the joint density can be written as

$$\begin{aligned}
&f(\boldsymbol{y}, \boldsymbol{\beta}, \tau) = \\
&K \exp\left\{-\frac{\tau}{2}(\boldsymbol{y}-\boldsymbol{X\beta})'(\boldsymbol{y}-\boldsymbol{X\beta})\right\} \exp\left\{-\frac{\tau}{2}(\boldsymbol{\beta}-\boldsymbol{\theta})'\boldsymbol{V}^{-1}(\boldsymbol{\beta}-\boldsymbol{\theta})\right\} \tau^{\alpha-1} \exp\{-\delta\tau\} = \\
&K \exp\left\{-\frac{\tau}{2}\left[(\boldsymbol{y}-\boldsymbol{X\beta})'(\boldsymbol{y}-\boldsymbol{X\beta}) + (\boldsymbol{\beta}-\boldsymbol{\theta})'\boldsymbol{V}^{-1}(\boldsymbol{\beta}-\boldsymbol{\theta}) + 2\delta\right]\right\} \tau^{\alpha-1}
\end{aligned}$$

$$g(\boldsymbol{\beta}, \tau|\boldsymbol{y}) = K_1 \exp\left\{-\frac{\tau}{2}(\boldsymbol{\beta}-\boldsymbol{\theta_1})'\boldsymbol{V}_1^{-1}(\boldsymbol{\beta}-\boldsymbol{\theta_1}) + \delta_1\right\}$$

where

$$\begin{aligned}
&\boldsymbol{V}_1 = (\boldsymbol{V}^{-1} + \boldsymbol{X'X})^{-1} \\
&\boldsymbol{\phi}_1 = \boldsymbol{V}_1(\boldsymbol{V}^{-1}\boldsymbol{\phi} + \boldsymbol{X'y}) \\
&\delta_1 = -\boldsymbol{\phi_1'}\boldsymbol{V}_1\boldsymbol{\phi}_1 + \boldsymbol{\phi'}\boldsymbol{V}^{-1}\boldsymbol{\phi} + \boldsymbol{y'y} + 2\delta
\end{aligned}$$

and so the posterior distribution is again a normal distribution.

5.11.0.2 Theorem (6.11.2) With the Bayesian model above we have

$$\boldsymbol{\beta}|\boldsymbol{y} \sim t(n + 2\alpha, \boldsymbol{\phi}_1, \boldsymbol{W}_1)$$

where t is a multivariate t distribution and

$$\boldsymbol{\phi}_1 = (\boldsymbol{V}^{-1} + \boldsymbol{X'X})^{-1}(\boldsymbol{V}^{-1}\boldsymbol{\phi} + \boldsymbol{X'y})$$
$$\boldsymbol{W}_1 = \left((\boldsymbol{y} - \boldsymbol{X\phi})'(\boldsymbol{I} + \boldsymbol{XVX'})^{-1}(\boldsymbol{y} - \boldsymbol{X\phi}) + 2\delta\right)(\boldsymbol{V}^{-1} + \boldsymbol{X'X})^{-1}/(n + 2\alpha)$$

proof omitted

5.11.1 Bayesian Inference for β

5.11.1.1 Theorem (6.11.3)

i. Based on the above model a reasonable estimator is the mean of the posterior distribution:

$$\hat{\boldsymbol{\beta}} = \boldsymbol{\phi}_1 = (\boldsymbol{V}^{-1} + \boldsymbol{X'X})^{-1}(\boldsymbol{V}^{-1}\boldsymbol{\phi} + \boldsymbol{X'y})$$

ii. for any vector $\boldsymbol{a}$ we have

$$\frac{\boldsymbol{a'}\hat{\boldsymbol{\beta}} - \boldsymbol{a'\phi}_1}{\boldsymbol{a'W_1a}} \sim t(n + 2\alpha)$$

iii.

$$\frac{\hat{\beta}_i - \theta_{1i}}{W_{1ii}} \sim t(n + 2\alpha)$$

iv. A $(1 - w)100\%$ credible interval is given by

$$\theta_{1i} \pm t_{w/2,n+2\alpha} W_{1ii}$$

proof omitted

5.11.1.2 Example (6.11.4) Let's apply this method to our houseprice data. To do so we need values for $\boldsymbol{\phi}, \boldsymbol{V}, \alpha, \delta$. We will use non-informative priors, so sensible choices are

$$\boldsymbol{\phi} = (0, 0, 0, 0)'$$
$$\boldsymbol{V} = 1000\boldsymbol{I}$$
$$\alpha = 0.001; \delta = 0.001$$

then

```
phi=cbind(rep(0, 5))
V=1000*diag(5)
alp=0.001;del=0.001
A=as.matrix(houseprice)
y=A[, 1, drop=FALSE]
n=length(y)
X=cbind(1, A[, -1])
```

```
phi1 = solve(solve(V)+t(X)%*%X)%*%(solve(V)%*%phi+t(X)%*%y)
round(cbind(phi1, solve(t(X)%*%X)%*%t(X)%*%y), 3)
```

```
##             Price   Price
##          -67.503 -67.620
## Sqfeet     0.086   0.086
## Floors   -26.463 -26.493
## Bedrooms  -9.284  -9.286
## Baths     37.326  37.381
```

and we see that the estimators are quite similar to the frequentist ones, a typical result when non-informative priors are used.

5.11.2 Modern Bayesian Analysis

Calculating the posterior distribution analytically as we did above is generally not possible. Instead one uses so called *Markov Chain Monte Carlo* simulation to generate data from the posterior distribution. For more on this take my course ESMA 5015 Simulation.

6 Analysis of Variance (ANOVA)

6.1 Non-Full Rank Models

6.1.1 Introduction

In many experiments one of a (relatively few) *treatments* is applied to each subject, and the interest is in the effects of the treatments. For example, in an experiment in agriculture one might treat each plant with one of four different fertilizers. In a medical trial a new vaccince might be compared to an old vaccince as well as to a placebo and maybe even no treatment.

In an *Analysis of Variance* (ANOVA) the focus is on the mean responses per treatment.

6.1.2 One-Way Model

6.1.2.1 Example (7.1.1) Chasnoff and others obtained several measures and responses for newborn babies whose mothers were classified by degree of cocaine use.

The study was conducted in the Perinatal Center for Chemical Dependence at Northwestern University Medical School. The measurement given here is the length of the newborn.

Source: Cocaine abuse during pregnancy: correlation between prenatal care and perinatal outcome Authors: SN MacGregor, LG Keith, JA Bachicha, and IJ Chasnoff Obstetrics and Gynecology 1989;74:882-885

Here is a look at the data

```
kable.nice(mothers[1:10, ], do.row.names = FALSE)
```

Status	Length
Drug Free	44.3
Drug Free	45.3
Drug Free	46.9
Drug Free	47.0
Drug Free	47.2
Drug Free	47.8
Drug Free	47.8
Drug Free	48.5
Drug Free	48.8
Drug Free	49.6

```
nrow(mothers)
```

```
## [1] 94
```

```
table(mothers$Status)
```

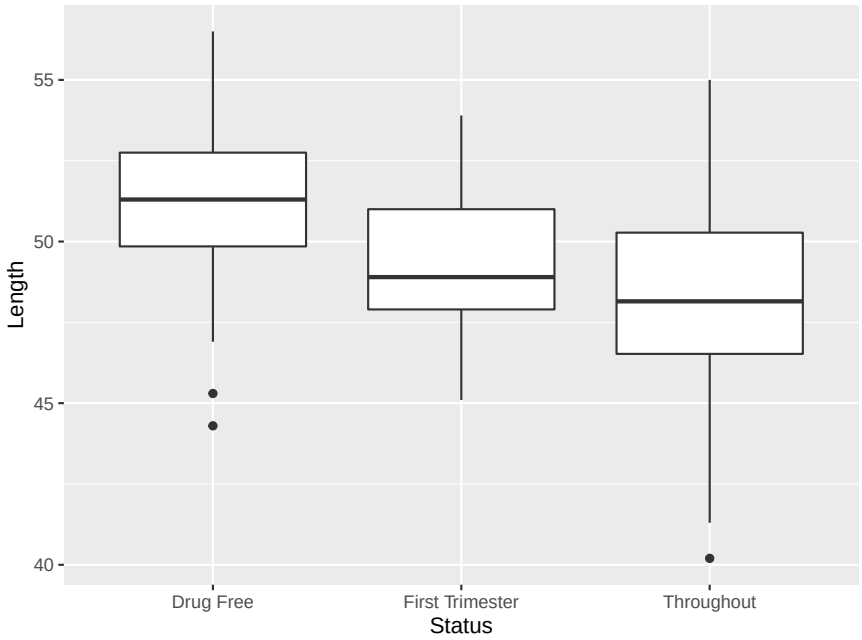
```
##
##       Drug Free First Trimester      Throughout
##              39              19              36
```

so there were a total of 94 babies. 39 had mothers with no drug use, 19 had mothers who stopped the cocaine use early during pregnancy and 36 had mothers who continued to take cocaine until birth.

Basic Question: does the cocaine use of the mother influence the health of the baby?

A good way to start is with a *multiple boxplot*:

```
ggplot(data=mothers , aes(Status, Length)) +
  geom_boxplot()
```



which does look like that more drug use leads to smaller (and likely less healthy) babies.

Here is another way to summarize the data:

```
out <- matrix(0, 3, 3)
colnames(out) <- c("Size", "Mean", "SD")
rownames(out) <- unique(mothers$Status)
out[, 1] <- tapply(mothers$Length,
                   mothers$Status, length)
out[, 2] <- round(tapply(mothers$Length,
                         mothers$Status, mean), 2)
out[, 3] <- round(tapply(mothers$Length,
                         mothers$Status, sd), 2)
kable.nice(out)
```

	Size	Mean	SD
Drug Free	39	51.1	2.9
First Trimester	19	49.3	2.5
Throughout	36	48.0	3.6

and again it seems there is an effect.

How can we write this as a model? Typically this is done with

$$y_{ij} = \mu + \alpha_i + \epsilon_{ij}$$

with i=1,2,3 and j_1=39, j_2=19, j_3=36

Here μ is the overall mean (disregarding the treatments), α_i is the mean effect due to treatment i and ϵ_{ij} is the random effect.

We can again use matrix notation to write down this model: Let

$$\boldsymbol{y} = \begin{pmatrix} y_{11} \\ \vdots \\ y_{336} \end{pmatrix}$$

$$\boldsymbol{X} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

$$\boldsymbol{\beta} = \begin{pmatrix} \mu \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix}$$

$$\boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_{11} \\ \vdots \\ \epsilon_{336} \end{pmatrix}$$

then the model is

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

and we see a model of the same form as in the regression problems before. The main difference is that now the design matrix $\boldsymbol{X}$ is always of the form as above, that is with 0's and 1's.

Here $\boldsymbol{X}$ is 96×4 matrix of rank 3 because the first column is equal to sum of the others, which are clearly linearly independent. Because $\boldsymbol{X}$ is not of full rank $\boldsymbol{X'X}$ is singular, $(\boldsymbol{X'X})^{-1}$ does not exist and therefore $(\boldsymbol{X'X})^{-1}\boldsymbol{X'y}$ can not be used to estimate $\boldsymbol{\beta}$.

One reason for this is that the model as written does not determine the parameters uniquely. For example,

$$y_{11} = 10 + 2 + \epsilon_{11} = 8 + 4 + \epsilon_{11}$$

We say the model is *overparametrized*. Note that this is not an issue of the sample size, it will not go away if n is increased.

There are a number of ways to overcome this problem:

1. reparametrize the problem with fewer, now unique, parameters
2. place constraints on the parameters
3. work with unique linear combinations of the parameters

Here are examples of these technics:

1. use the model

$$y_{ij} = \mu_i + \epsilon_{ij}$$

2. Add the condition $\alpha_1 + \alpha_2 + \alpha_3 = 0$ to the model.
3. rewrite the model in terms of $\alpha_1 - \alpha_2$ and $\alpha_2 - \alpha_3$, that is the changes from one treatment to the next. This makes most sense if (as is the case for our data) there is an ordering of the treatments.

6.1.3 Two-Way Model

6.1.3.1 Example (7.1.2) Reference: Loven, Faith. (1981). A Study of the Interlist Equivalency of the CID W-22 Word List Presented in Quiet and in Noise. Unpublished MS Thesis, University of Iowa.

Description: Percent of a Standard 50-word list heard correctly in the presence of background noise. 24 subjects with normal hearing listened to standard audiology tapes of English words at low volume with a noisy background. They repeated the words and were scored correct or incorrect in their perception of the words. The order of list presentation was randomized.

The word lists are standard audiology tools for assessing hearing. They are calibrated to be equally difficult to perceive. However, the original calibration was performed with normal-hearing subjects and no noise background. The experimenter wished to determine whether the lists were still equally difficult to understand in the presence of a noisy background.

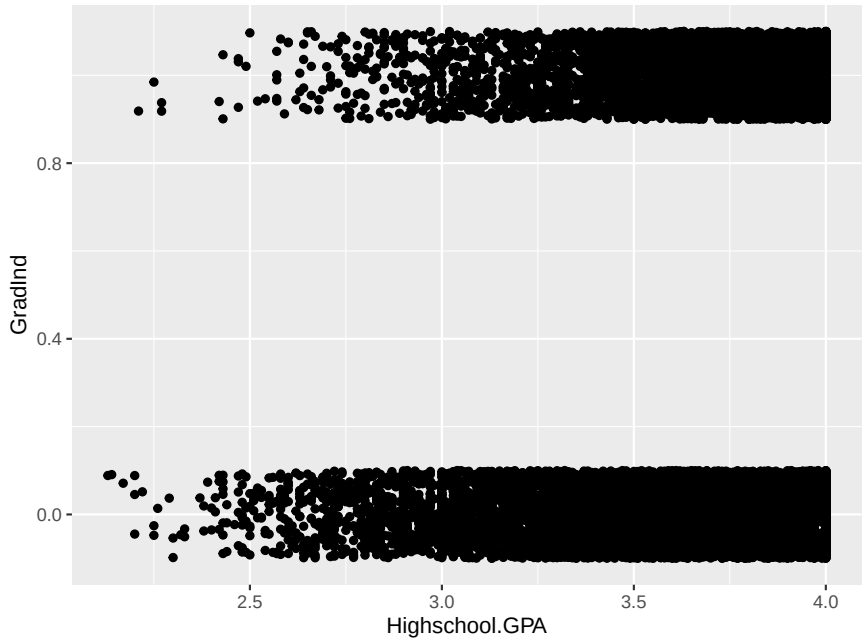
```
kable.nice(hearingaid[1:10, ], do.row.names = FALSE)
```

Subject	List	Score
1	1	28
2	1	24
3	1	32
4	1	30
5	1	34
6	1	30
7	1	36
8	1	32
9	1	48
10	1	32

so here the response variable is Score and there are two predictor variables, Subject and List. The interest is specifically in List.

Both Subject and List contain numbers, but these are really labels, so in R we need to turn them into *factors*:

```
tmp <- tapply(hearingaid$Score, hearingaid$List, mean)
hearingaid$List <- factor(hearingaid$List,
                          levels=order(tmp),
                          ordered = TRUE)
tmp <- tapply(hearingaid$Score, hearingaid$Subject, mean)
hearingaid$Subject <- factor(hearingaid$Subject,
                             levels=order(tmp),
                             ordered = TRUE)
ggplot(data=hearingaid, aes(List, Score)) +
         geom_boxplot()
```



Also the summary table for List:

```
sum.tbl <-
  data.frame(
    List=c(3, 4, 2, 1),
    n=as.integer(tapply(hearingaid$Score,
                hearingaid$List,length)),
    Mean=round(tapply(hearingaid$Score,
                      hearingaid$List, mean), 1),
    Sd=round(tapply(hearingaid$Score,
                    hearingaid$List, sd), 2)
)
rownames(sum.tbl) <- NULL
kable.nice(sum.tbl)
```

	List	n	Mean	Sd
1	3	24	25.2	8.32
2	4	24	25.6	7.78
3	2	24	29.7	8.06
4	1	24	32.8	7.41

Notice the ordering by mean response.

Analogously to the previous discussion we can write the model for this data as

$$y_{ij} = \mu + \alpha_i + \beta_j + \epsilon_{ij}$$

for i=1,..,4 and j=1,..,24

In matrix form we have

$$\boldsymbol{y} = \begin{pmatrix} y_{11} \\ \vdots \\ y_{424} \end{pmatrix}$$

$$\boldsymbol{X} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & \dots & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & & & & & \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & \dots & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & & & & & \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & \dots & 1 \end{pmatrix}$$

$$\boldsymbol{\beta} = \begin{pmatrix} \mu & \alpha_1 & \dots & \alpha_4 & \beta_1 & \dots & \beta_{24} \end{pmatrix}'$$

$$\boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_{11} \\ \vdots \\ \epsilon_{424} \end{pmatrix}$$

then the model is

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

How can we create such a matrix $\boldsymbol{X}$ in R? Note for any block with 1's in columns 2 to 5 the following column form a diagonal matrix. Let's illustrate this with i=3,j=4 instead of 24, so it is easier to see:

```
make.X=function(I, J) {
  X=NULL
  for(i in 1:I) {
    tmp=matrix(0, J, I)
    tmp[, i]=1
    X=rbind(X, cbind(tmp, diag(J)))
  }
  X=cbind(1, X)
  X
}
make.X(3,4)
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
##  [1,]    1    1    0    0    1    0    0    0
##  [2,]    1    1    0    0    0    1    0    0
##  [3,]    1    1    0    0    0    0    1    0
##  [4,]    1    1    0    0    0    0    0    1
```

```
##  [5,]    1    0    1    0    1    0    0    0
##  [6,]    1    0    1    0    0    1    0    0
##  [7,]    1    0    1    0    0    0    1    0
##  [8,]    1    0    1    0    0    0    0    1
##  [9,]    1    0    0    1    1    0    0    0
## [10,]    1    0    0    1    0    1    0    0
## [11,]    1    0    0    1    0    0    1    0
## [12,]    1    0    0    1    0    0    0    1
```

The rank of our matrix $\boldsymbol{X}$ is

```
qr(make.X(4,24))$rank
```

```
## [1] 27
```

but there are 97 parameters, so again not all can be estimated.

6.2 Estimation in ANOVA

6.2.1 Estimation of β

We consider the model

$$\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\epsilon}$$

where $\boldsymbol{X}$ is $n \times p$ with rank $k < p \leq n$. We also have the assumptions $E[\boldsymbol{y}] = \boldsymbol{X\beta}$, $cov(\boldsymbol{y}) = \sigma^2\boldsymbol{I}$.

Using least squares we need to find $\hat{\boldsymbol{\beta}}$ that minimizes

$$\boldsymbol{\epsilon'\epsilon} = (\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})'(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})$$

as before we can expand this and differentiate and arrive at *normal equations*

$$\boldsymbol{X'X}\hat{\boldsymbol{\beta}} = \boldsymbol{X'y}$$

but now $\boldsymbol{X'X}$ is singular and has no inverse. In fact this system of equations has infinitely many solutions:

6.2.1.1 Theorem (7.2.1)

If $\boldsymbol{X}$ is $n \times p$ with rank $k < p \leq n$, the system of equations

$$\boldsymbol{X'X}\hat{\boldsymbol{\beta}} = \boldsymbol{X'y}$$

is consistent.

proof omitted

Because the system of equations is consistent, by (4.2.14) a solution is given by

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X'X})^{-}\boldsymbol{X'y}$$

where $(\boldsymbol{X'X})^{-}$ is a generalized inverse of $\boldsymbol{X'X}$.

For any generalized inverse we have

$$E[\hat{\boldsymbol{\beta}}] = (\boldsymbol{X'X})^{-}\boldsymbol{X'}E[\boldsymbol{y}] = (\boldsymbol{X'X})^{-}\boldsymbol{X'X\beta} = \boldsymbol{\beta}$$

and so $\hat{\boldsymbol{\beta}}$ is an unbiased estimator of $(\boldsymbol{X'X})^{-}\boldsymbol{X'X\beta} = \boldsymbol{\beta}$. However, since $(\boldsymbol{X'X})^{-}\boldsymbol{X'X} \neq \boldsymbol{I}$, $\hat{\boldsymbol{\beta}}$ is not an unbiased estimator of $\boldsymbol{\beta}$. In fact, $E[\hat{\boldsymbol{\beta}}]$ depends on the particular choice of $(\boldsymbol{X'X})^{-}$.

Is there a matrix $\boldsymbol{A}$ such that $E[\boldsymbol{Ay}] = \boldsymbol{\beta}$? If so, then

$$\boldsymbol{\beta} = E[\boldsymbol{Ay}] = E[\boldsymbol{A}(\boldsymbol{X\beta} + \boldsymbol{\epsilon})] = E[\boldsymbol{AX\beta}] + \boldsymbol{A}E[\boldsymbol{\epsilon}] = \boldsymbol{AX\beta}$$

This must hold for all $\boldsymbol{\beta}$, and so we must have $\boldsymbol{AX} = \boldsymbol{I}$. But $rank(\boldsymbol{AX}) < p$, and so no such matrix $\boldsymbol{A}$ can exist.

6.2.1.2 Example (7.2.2) Let's consider a simple oneway model with two groups and three repeated measurements:

$$y_{ij} = \mu + \alpha_i + \epsilon_{ij}$$

with i=1,2 and j=1,2,3. So

$$\boldsymbol{\beta} = \begin{pmatrix} \mu \\ \alpha_1 \\ \alpha_2 \end{pmatrix}$$

$$\boldsymbol{X} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

$$\boldsymbol{X'X} = \begin{pmatrix} 6 & 3 & 3 \\ 3 & 3 & 0 \\ 3 & 0 & 3 \end{pmatrix}$$

A generalized inverse is given by

$$(\boldsymbol{X'X})^{-} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1/3 & 0 \\ 0 & 0 & 1/3 \end{pmatrix}$$

let's check:

```
X=cbind(1, c(1,1,1,0,0,0), c(0,0,0,1,1,1))
X
```

```
##      [,1] [,2] [,3]
## [1,]    1    1    0
## [2,]    1    1    0
## [3,]    1    1    0
## [4,]    1    0    1
## [5,]    1    0    1
## [6,]    1    0    1
```

```
X.X=t(X)%*%X
X.X
```

```
##      [,1] [,2] [,3]
## [1,]    6    3    3
## [2,]    3    3    0
## [3,]    3    0    3
```

```
g.X=diag(c(0,1,1)/3)
g.X
```

```
##      [,1]      [,2]      [,3]
## [1,]    0 0.0000000 0.0000000
## [2,]    0 0.3333333 0.0000000
## [3,]    0 0.0000000 0.3333333
```

```
X.X%*%g.X%*%X.X
```

```
##      [,1] [,2] [,3]
## [1,]    6    3    3
## [2,]    3    3    0
## [3,]    3    0    3
```

so this is indeed a generalized inverse. So now

$$\boldsymbol{X'y} = \begin{pmatrix} \sum_{i,j} y_{ij} \\ \sum_j y_{1j} \\ \sum_j y_{2j} \end{pmatrix} = \begin{pmatrix} y_{..} \\ y_{1.} \\ y_{2.} \end{pmatrix} \quad \hat{\boldsymbol{\beta}} = (\boldsymbol{X'X})^{-}\boldsymbol{X'y} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1/3 & 0 \\ 0 & 0 & 1/3 \end{pmatrix} \begin{pmatrix} \sum_{i,j} y_{ij} \\ \sum_j y_{1j} \\ \sum_j y_{2j} \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{1}{3}\sum_j y_{1j} \\ \frac{1}{3}\sum_j y_{2j} \end{pmatrix} = \begin{pmatrix} 0 \\ \bar{y}_{1.} \\ \bar{y}_{2.} \end{pmatrix}$$

Note

$$
\begin{aligned}
&E[\bar{y}_{i.}] = \\
&E[\frac{1}{3}\sum_j y_{ij}] = \\
&\frac{1}{3}\sum_j E[y_{ij}] = \\
&\frac{1}{3}\sum_j E[\mu + \alpha_i + \epsilon_{ij}] = \\
&\frac{1}{3}\left[3\mu + 3\alpha_i + 0\right] = \\
&\mu + \alpha_i
\end{aligned}
$$

6.2.1.3 Example (7.2.3) Let's return to the hearing aid data set, see (7.1.2). Here we have a model of the form

$$y_i = \mu + \alpha_i + \beta_j + \epsilon_{ij}$$

i=1,..,4 and j=1,..,24

```
make.X=function(I, J) {
  X=NULL
  for(i in 1:I) {
    tmp=matrix(0, J, I)
    tmp[, i]=1
    X=rbind(X, cbind(tmp, diag(J)))
  }
  X=cbind(1, X)
  X
}
X=make.X(4, 24)
y=as.matrix(hearingaid[, 1, drop=FALSE])
X.X=t(X)%*%X
library(MASS)
gX=ginv(X.X)
betahat=gX%*%t(X)%*%y
round(c(betahat), 3)
```

```
##  [1]   9.677   2.419   2.419   2.419   2.419 -11.097 -10.097  -9.097  -8.097  -7.097
## [12]  -5.097  -4.097  -3.097  -2.097  -1.097  -0.097   0.903   1.903   2.903   3.903
## [23]   5.903   6.903   7.903   8.903   9.903  10.903  11.903
```

6.2.2 Estimable Functions

If we can not estimate $\boldsymbol{\beta}$, can we instead estimate a linear function of $\boldsymbol{\beta}$, say $\boldsymbol{\lambda}'\boldsymbol{\beta}$?

6.2.2.1 Definition (7.2.4) A linear function of parameters $\boldsymbol{\lambda'\beta}$ is said to be *estimable* if there exists a vector $\boldsymbol{a}$ such that $E[\boldsymbol{a'y}] = \boldsymbol{\lambda'\beta}$.

6.2.2.2 Theorem (7.2.5) Say $\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\epsilon}$ with $E[\boldsymbol{y}] = \boldsymbol{X\beta}$ and $\boldsymbol{X}$ is $n \times p$ of rank $k < p \leq n$. A linear function $\boldsymbol{\lambda'\beta}$ is estimable if and only if one of the following conditions holds

i. $\boldsymbol{\lambda'}$ is a linear combination of the rows of $\boldsymbol{X}$.

ii. $\boldsymbol{\lambda'}$ is a linear combination of the rows of $\boldsymbol{X'X}$ or $\boldsymbol{\lambda}$ is a linear combination of the columns of $\boldsymbol{X'X}$.

iii. Either $\boldsymbol{X'X(X'X)^{-}\lambda = \lambda}$ or $\boldsymbol{\lambda'(X'X)^{-}X'X = \lambda'}$

proof

i. say $\boldsymbol{\lambda'}$ is a linear combination of the rows of $\boldsymbol{X}$, then there exists $\boldsymbol{a}$ such that $\boldsymbol{\lambda' = a'X}$ and so

$$E[\boldsymbol{a'y}] = \boldsymbol{a'}E[\boldsymbol{y}] = \boldsymbol{a'X\beta} = \boldsymbol{\lambda'\beta}$$

Conversely, if $\boldsymbol{\lambda'\beta}$ is estimable there exists $\boldsymbol{a}$ such that $E[\boldsymbol{a'y}] = \boldsymbol{\lambda'\beta}$. Therefore $\boldsymbol{a'X\beta} = \boldsymbol{\lambda'\beta}$ for all $\boldsymbol{\beta}$, and therefore $\boldsymbol{a'X} = \boldsymbol{\lambda'}$, and so $\boldsymbol{\lambda'}$ is a linear combination of the rows of $\boldsymbol{X}$.

proofs of ii and iii omitted.

6.2.2.3 Example (7.2.6) Consider again the simple oneway model with two groups and three repeated measurements from (7.2.2). That is $\boldsymbol{\beta} = \begin{pmatrix}\mu & \alpha_1 & \alpha_2\end{pmatrix}'$ and

$$\boldsymbol{X} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

We want to show that $\alpha_1 - \alpha_2$ is estimable. Note that

$$\alpha_1 - \alpha_2 = \begin{pmatrix}0 & 1 & -1\end{pmatrix} \begin{pmatrix}\mu \\ \alpha_1 \\ \alpha_2\end{pmatrix} = \boldsymbol{\lambda'\beta}$$

we see that $\boldsymbol{\lambda'} = \begin{pmatrix}0 & 1 & -1\end{pmatrix}$.

Using (7.2.4i):

We need to find $\boldsymbol{a}$ such that $\boldsymbol{a'X} = \boldsymbol{\lambda'}$. In fact, if $\boldsymbol{a'} = \begin{pmatrix}0 & 0 & 1 & -1 & 0 & 0\end{pmatrix}$, then we have $\boldsymbol{a'X} = \boldsymbol{\lambda'}$.

$\boldsymbol{a}$ here is not unique, there are many other choices.

Using (7.2.4ii):

We have

$$\boldsymbol{X'X} = \begin{pmatrix} 6 & 3 & 3 \\ 3 & 3 & 0 \\ 3 & 0 & 3 \end{pmatrix}$$

Now we need a vector $\boldsymbol{a}$ such that $\boldsymbol{X'Xa} = \begin{pmatrix} 0 & 1 & -1 \end{pmatrix}'$. One such vector is $\boldsymbol{a} = \begin{pmatrix} 0 & 1/3 & -1/3 \end{pmatrix}'$. Again there are other possibilities.

Using (7.2.4iii):

We saw before that a generalized inverse is given by

$$(\boldsymbol{X'X})^- = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1/3 & 0 \\ 0 & 0 & 1/3 \end{pmatrix}$$

and we see easily that for $\boldsymbol{\lambda}' = \begin{pmatrix} 0 & 1 & -1 \end{pmatrix}$ we have $\boldsymbol{X'X}(\boldsymbol{X'X})^-\boldsymbol{\lambda} = \boldsymbol{\lambda}$.

6.2.2.4 Theorem (7.2.7) The number of estimable linear functions of $\boldsymbol{\beta}$ is equal to the rank of $\boldsymbol{X}$.

proof omitted

From the two theorems above it is clear that we can check the rows of $\boldsymbol{X}$ or of $\boldsymbol{X'X}$ to see which functions are estimable.

6.2.2.5 Example (7.2.8) Consider a twoway model with two groups and no repeated measurements. That is $\boldsymbol{\beta} = \begin{pmatrix} \mu & \alpha_1 & \alpha_2 & \beta_1 & \beta_2 \end{pmatrix}'$ and

$$\boldsymbol{X} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

To get to a matrix with only linearly independent rows we can proceed as follows:

- subtract the first row from all others:

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & -1 & 1 \end{pmatrix}$$

- subtract second and third from fourth:

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

taking the first three rows as $\boldsymbol{\lambda}_1'$,$\boldsymbol{\lambda}_2'$ and $\boldsymbol{\lambda}_3'$, we find

$$\begin{aligned} \boldsymbol{\lambda}_1'\boldsymbol{\beta} &= \mu + \alpha_1 + \beta_1 \\ \boldsymbol{\lambda}_2'\boldsymbol{\beta} &= \beta_2 - \beta_1 \\ \boldsymbol{\lambda}_3'\boldsymbol{\beta} &= \alpha_2 - \alpha_1 \end{aligned}$$

6.2.2.6 Definition (7.2.9) Let $\boldsymbol{a}$ be a vector such $\sum a_i = 0$. The $\boldsymbol{a'\beta}$ is called a *contrast.*

6.2.2.7 Example (7.2.10) In the example above the second and third linear combination are contrasts.

6.2.2.8 Theorem (7.2.11) Let $\boldsymbol{\lambda'\beta}$ be an estimable function. Let $\hat{\boldsymbol{\beta}}$ be any solution to the normal equations $\boldsymbol{X'X\beta} = \boldsymbol{X'y}$ and let $\boldsymbol{a}$ be any solution to $\boldsymbol{X'Xa} = \boldsymbol{\lambda}$. Then the two estimators $\boldsymbol{\lambda'\hat{\beta}}$ and $\boldsymbol{a'X'y}$ have the following properties:

i. $E[\boldsymbol{\lambda'\hat{\beta}}] = E[\boldsymbol{a'X'y}] = \beta$

ii. $\boldsymbol{\lambda'\hat{\beta}} = \boldsymbol{a'X'y}$

iii. $\boldsymbol{\lambda'\hat{\beta}}$ and $\boldsymbol{a'X'y}$ are invariant to the choice of $\hat{\boldsymbol{\beta}}$ or $\boldsymbol{a}$

proof

i. By (7.2.1) we have

$$E[\boldsymbol{\lambda'\hat{\beta}}] = \boldsymbol{\lambda'}(\boldsymbol{X'X})^{-}\boldsymbol{X'X\beta}$$

by (7.2.3)iii $\lambda'(\boldsymbol{X'X})^{-}\boldsymbol{X'X\beta} = \boldsymbol{\lambda'}$ and so

$$E[\boldsymbol{\lambda'\hat{\beta}}] = \boldsymbol{\lambda'\beta}$$

by (7.2.3)ii

$$E[\boldsymbol{a'X'y}] = \boldsymbol{a'X'}E[\boldsymbol{y}]] = \boldsymbol{a'X'X\beta} = \boldsymbol{\lambda'\beta}$$

6.2.2.9 Example (7.2.12) In example (7.2.5) we saw that the linear function $\boldsymbol{\lambda'\beta} = \alpha_1 - \alpha_2$ was estimable with $\boldsymbol{a} = \begin{pmatrix} 0 & 1/3 & -1/3 \end{pmatrix}'$, so now

$$\boldsymbol{a'X'y} =$$

$$\begin{pmatrix} 0 & 1/3 & -1/3 \end{pmatrix}' \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} y_{11} \\ y_{12} \\ y_{13} \\ y_{21} \\ y_{22} \\ y_{23} \end{pmatrix} =$$

$$\begin{pmatrix} 0 & 1/3 & -1/3 \end{pmatrix}' \begin{pmatrix} y_{..} \\ y_{1.} \\ y_{2.} \end{pmatrix} =$$

$$y_{1.}/3 - y_{2.}/3 = \bar{y}_{1.} - \bar{y}_{2.}$$

or using the other solution we need a solution to the normal equation $\boldsymbol{X'X\beta} = \boldsymbol{X'y}$.

$$\begin{pmatrix} 6 & 3 & 3 \\ 3 & 3 & 0 \\ 3 & 0 & 3 \end{pmatrix} \begin{pmatrix} \hat{\mu} \\ \hat{\alpha}_1 \\ \hat{\alpha}_2 \end{pmatrix} = \begin{pmatrix} y_{..} \\ y_{1.} \\ y_{2.} \end{pmatrix}$$

$$6\hat{\mu} + 3\hat{\alpha}_1 + 3\hat{\alpha}_2 = y_{..}$$

$$3\hat{\mu} + 3\hat{\alpha}_1 = y_{1.}$$

$$3\hat{\mu} + \hat{\alpha}_2 = y_{2.}$$

Now we have two equations in three unknowns, so we can set $\hat{\mu}$ equal to some constant and obtain

$$\hat{\boldsymbol{\beta}} = \begin{pmatrix} \hat{\mu} \\ \hat{\alpha}_1 \\ \hat{\alpha}_2 \end{pmatrix} = \begin{pmatrix} 0 \\ \bar{y}_{1.} \\ \bar{y}_{2.} \end{pmatrix} + \hat{\mu} \begin{pmatrix} 0 \\ -1 \\ -1 \end{pmatrix}$$

Finally

$$\boldsymbol{\lambda'\hat{\beta}} = \begin{pmatrix} 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ \bar{y}_{1.} - \hat{\mu} \\ \bar{y}_{2.} - \hat{\mu} \end{pmatrix} = \bar{y}_{1.} - \bar{y}_{2.}$$

6.2.2.10 Theorem (7.2.13) Let $\boldsymbol{\lambda'\beta}$ be an estimable function. Let $\hat{\boldsymbol{\beta}}$ be any solution to the normal equations $\boldsymbol{X'X\beta} = \boldsymbol{X'y}$ and let $\boldsymbol{a}$ be any solution to $\boldsymbol{X'Xa} = \boldsymbol{\lambda}$. Then the variances of the two estimators $\boldsymbol{\lambda'\hat{\beta}}$ and $\boldsymbol{a'X'y}$ have the following properties:

i. $var(\boldsymbol{a'X'y}) = \sigma^2 \boldsymbol{a'X'Xa} = \sigma^2 \boldsymbol{a'\lambda}$

ii. $var(\boldsymbol{\lambda'\hat{\beta}}) = \sigma^2 \boldsymbol{\lambda}'(\boldsymbol{X'X})^{-}\boldsymbol{\lambda}$

iii. $var(\boldsymbol{\lambda'}\hat{\boldsymbol{\beta}})$ is unique, hat is invariant under the choice of $\boldsymbol{a}$ or $(\boldsymbol{X'X})^-$

proof

i.

$$
\begin{aligned}
&var(\boldsymbol{a'X'y}) = \\
&\boldsymbol{a'X'}cov(\boldsymbol{y})\boldsymbol{Xa} = \\
&\boldsymbol{a'X'}(\sigma^2\boldsymbol{I})\boldsymbol{Xa} = \\
&\sigma^2\boldsymbol{a'X'Xa} = \\
&\sigma^2\boldsymbol{a'\lambda}
\end{aligned}
$$

ii and iii omitted

6.2.2.11 Theorem (7.2.14) Let $\boldsymbol{\lambda_1'\beta}$ and $\boldsymbol{\lambda_2'\beta}$ be two estimable function, then

$$cov(\boldsymbol{\lambda_1'}\hat{\boldsymbol{\beta}}, \boldsymbol{\lambda_2'}\hat{\boldsymbol{\beta}}) = \sigma^2\boldsymbol{\lambda}_1'(\boldsymbol{X'X})^-\boldsymbol{\lambda}_2$$

proof similar to proof of theorem above

6.2.2.12 Theorem (7.2.15) Let $\boldsymbol{\lambda'\beta}$ be an estimable function. Then the two estimators $\boldsymbol{\lambda'}\hat{\boldsymbol{\beta}}$ and $\boldsymbol{a'X'y}$ are BLUE.

proof omitted

6.2.3 Estimation of σ^2

Again we define

$$\text{SSE} = (\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})'(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})$$

where $\hat{\boldsymbol{\beta}}$ is any solution of the normal equations. As before we have alternatively

$$\text{SSE} = \boldsymbol{y'y} - \hat{\boldsymbol{\beta}}'\boldsymbol{X'y} = \boldsymbol{y'}\left[\boldsymbol{I} - \boldsymbol{X}(\boldsymbol{X'X})^-\boldsymbol{X'}\right]\boldsymbol{y}$$

and we define

$$s^2 = \text{SSE}/(n-k)$$

6.2.3.1 Theorem (7.2.16)

i. $E[s^2] = \sigma^2$

ii. s^2 is invariant under the choice of $\hat{\boldsymbol{\beta}}$ or the choice of $\boldsymbol{(X'X)^-}$.

proof

6.2.4 Normal Model

6.2.4.1 Theorem (7.2.17)

If $\boldsymbol{y} \sim N(\boldsymbol{X\beta}, \sigma^2\boldsymbol{I})$, the maximum likelihood estimators are

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X'X})^-\boldsymbol{X'y}$$
$$\hat{\sigma}^2 = \frac{1}{n}(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})'(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})$$

proof omitted

6.2.4.2 Theorem (7.2.18)

Under the normal model

i. $\hat{\boldsymbol{\beta}} \sim N_p\left[(\boldsymbol{X'X})^-\boldsymbol{X'X\beta}, \sigma^2(\boldsymbol{X'X})^-\boldsymbol{X'X}(\boldsymbol{X'X})^-\right]$

ii. $(n-k)s^2/\sigma^2 \sim \chi^2(n-k)$

iii. $\hat{\boldsymbol{\beta}}$ and s^2 are independent.

proof omitted

6.2.5 Reparametrization

We discussed before that one often can change the parameters in order to make the problem solvable. Here is a formal discussion of this issue.

A reparametrization is a transformation of the non-full rank model $\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\epsilon}$ to a full-rank model $\boldsymbol{y} = \boldsymbol{Z\gamma} + \boldsymbol{\epsilon}$, where $\boldsymbol{\gamma} = \boldsymbol{U\beta}$ is a set of k linearly independent functions of $\boldsymbol{\beta}$. So we can write

$$\boldsymbol{Z\gamma} = \boldsymbol{ZU\beta} = \boldsymbol{X\beta}$$

This holds for all $\boldsymbol{\beta}$, and so we have $\boldsymbol{ZU} = \boldsymbol{X}$. Since $\boldsymbol{U}$ is $k \times p$ of rank $k < p$, the matrix $\boldsymbol{UU'}$ is nonsingular and we find $\boldsymbol{ZUU'} = \boldsymbol{XU'}$ or

$$\boldsymbol{Z} = \boldsymbol{XU'}(\boldsymbol{UU'})^{-1}$$

It can be seen that $\boldsymbol{Z}$ is full-rank and that therefore the normal equations have the solution

$$\hat{\boldsymbol{\gamma}} = (\boldsymbol{Z'Z})^{-1}\boldsymbol{Z'y}$$

Since $\boldsymbol{Z\gamma} = \boldsymbol{X\beta}$, the estimators $\boldsymbol{Z\hat{\gamma}}$ and $\boldsymbol{X\hat{\beta}}$ are also equal

$$\boldsymbol{Z\hat{\gamma}} = \boldsymbol{X\hat{\beta}}$$

6.2.5.1 Theorem (7.2.19)

$$s^2 = \frac{1}{n-k}(\boldsymbol{y} - \boldsymbol{Z\hat{\gamma}})(\boldsymbol{y} - \boldsymbol{Z\hat{\gamma}})$$

$$\text{SSE} = (\boldsymbol{y} - \boldsymbol{X\hat{\beta}})(\boldsymbol{y} - \boldsymbol{X\hat{\beta}}) = (\boldsymbol{y} - \boldsymbol{Z\hat{\gamma}})(\boldsymbol{y} - \boldsymbol{Z\hat{\gamma}})$$

proof omitted

6.2.5.2 Example (7.2.20)

Consider the model

$$\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\epsilon} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mu \\ \alpha_1 \\ \alpha_2 \end{pmatrix} + \begin{pmatrix} \epsilon_{11} \\ \epsilon_{12} \\ \epsilon_{21} \\ \epsilon_{22} \end{pmatrix}$$

$\boldsymbol{X}$ has rank 2, so there are two linearly independent estimable functions. These can be chosen in any number of ways, for example $\mu + \alpha_1$ and $\mu + \alpha_2$. With this choice we have

$$\boldsymbol{\gamma} = \begin{pmatrix} \mu + \alpha_1 \\ \mu + \alpha_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mu \\ \alpha_1 \\ \alpha_2 \end{pmatrix} = \boldsymbol{U\beta}$$

Let

$$\boldsymbol{Z} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$$

then $\boldsymbol{Z\gamma} = \boldsymbol{X\beta}$

6.2.6 Side Conditions

6.2.6.1 Definition (7.2.21)

A *side condition* is an $(p-k) \times k$ matrix $\boldsymbol{T}$ of rank p-k such that $\boldsymbol{T\beta} = 0$ and $\boldsymbol{T\beta}$ are nonestimable functions.

Note that if one of the $\boldsymbol{T\beta}$ were an estimable function it would be a linear combination of $\boldsymbol{X'X\beta}$ and would therefore not add to the rank.

6.2.6.2 Theorem (7.2.22) If $\boldsymbol{y = X\beta + \epsilon}$ and $\boldsymbol{T}$ is a side condition, then

$$\hat{\boldsymbol{\beta}} = \left(\boldsymbol{X'X + T'T}\right)^{-1}\boldsymbol{X'y}$$

is the unique vector $\hat{\boldsymbol{\beta}}$ such that $\boldsymbol{X'X\hat{\beta} = X'y}$ and $\boldsymbol{T\hat{\beta}} = 0$

proof the two equation can be combined into

$$\begin{pmatrix}\boldsymbol{y}\\ \boldsymbol{0}\end{pmatrix} = \begin{pmatrix}\boldsymbol{X}\\ \boldsymbol{T}\end{pmatrix}\boldsymbol{\beta} + \begin{pmatrix}\boldsymbol{\epsilon}\\ \boldsymbol{0}\end{pmatrix}$$

and by the conditions of the theorem the matrix $\begin{pmatrix}\boldsymbol{X}\\ \boldsymbol{T}\end{pmatrix}$ is full-rank. Therefore $\begin{pmatrix}\boldsymbol{X}\\ \boldsymbol{T}\end{pmatrix}'\begin{pmatrix}\boldsymbol{X}\\ \boldsymbol{T}\end{pmatrix}$ has an inverse, and we find

$$\begin{pmatrix}\boldsymbol{X}\\ \boldsymbol{T}\end{pmatrix}'\begin{pmatrix}\boldsymbol{X}\\ \boldsymbol{T}\end{pmatrix}\hat{\boldsymbol{\beta}} = \begin{pmatrix}\boldsymbol{X}\\ \boldsymbol{T}\end{pmatrix}'\begin{pmatrix}\boldsymbol{y}\\ \boldsymbol{0}\end{pmatrix}$$

$$\hat{\boldsymbol{\beta}} = \left(\begin{pmatrix}\boldsymbol{X}\\ \boldsymbol{T}\end{pmatrix}'\begin{pmatrix}\boldsymbol{X}\\ \boldsymbol{T}\end{pmatrix}\right)^{-1}\begin{pmatrix}\boldsymbol{X}\\ \boldsymbol{T}\end{pmatrix}'\begin{pmatrix}\boldsymbol{y}\\ \boldsymbol{0}\end{pmatrix}$$

$$\hat{\boldsymbol{\beta}} = \left(\begin{pmatrix}\boldsymbol{X'} & \boldsymbol{T'}\end{pmatrix}\begin{pmatrix}\boldsymbol{X}\\ \boldsymbol{T}\end{pmatrix}\right)^{-1}\begin{pmatrix}\boldsymbol{X'} & \boldsymbol{T'}\end{pmatrix}'\begin{pmatrix}\boldsymbol{y}\\ \boldsymbol{0}\end{pmatrix}$$

$$\hat{\boldsymbol{\beta}} = \left(\boldsymbol{X'X + T'T}\right)^{-1}\boldsymbol{X'y}$$

6.2.6.3 Example (7.2.23) Let's return to example (7.2.18), where we used the model

$$y_{ij} = \mu + \alpha_i + \epsilon_{ij}$$

with i=1, 2.

Using theorem (7.2.4) we can easily see that $\alpha_1 + \alpha_2$ is not an estimable function. This can be written as $(0\ 1\ 1)\boldsymbol{\beta} = 0$ and so $\boldsymbol{T} = (0\ 1\ 1)$.

Now

$$\boldsymbol{X'X + T'T} = \begin{pmatrix} 4 & 2 & 2 \\ 2 & 2 & 0 \\ 2 & 0 & 2 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} (0\ 1\ 1) =$$

$$\begin{pmatrix} 4 & 2 & 2 \\ 2 & 2 & 0 \\ 2 & 0 & 2 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 4 & 2 & 2 \\ 2 & 3 & 1 \\ 2 & 1 & 3 \end{pmatrix}$$

$$(\boldsymbol{X'X + T'T})^{-1} = \frac{1}{4} \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & 0 \\ -1 & 0 & 2 \end{pmatrix}$$

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X'X + T'T})^{-1}\boldsymbol{X'y} =$$

$$\frac{1}{4} \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & 0 \\ -1 & 0 & 2 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} y_{11} \\ y_{12} \\ y_{21} \\ y_{22} \end{pmatrix} =$$

$$\frac{1}{4} \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & 0 \\ -1 & 0 & 2 \end{pmatrix} \begin{pmatrix} y_{..} \\ y_{1.} \\ y_{2.} \end{pmatrix} =$$

$$\frac{1}{4} \begin{pmatrix} 2y_{..} - y_{1.} - y_{2.} \\ 2y_{1.} - y_{..} \\ 2y_{2.} - y_{..} \end{pmatrix} = \begin{pmatrix} \bar{y}_{..} \\ \bar{y}_{1.} - \bar{y}_{..} \\ \bar{y}_{2.} - \bar{y}_{..} \end{pmatrix}$$

because $\bar{y}_{1.} + \bar{y}_{.} = \bar{y}_{..}$

6.3 Hypothesis Tests in ANOVA

While in a regression problem the focus is usually on estimation, in the case of ANOVA it is generally on hypothesis testing.

In this section we will consider hypotheses about the β's in the model $\boldsymbol{y = X\beta + \epsilon}$, where $\boldsymbol{X}$ is $n \times p$ of rank $k < p \leq n$, and $\boldsymbol{y} \sim N_n(\boldsymbol{X\beta}, \sigma^2\boldsymbol{I})$.

6.3.1 Testable Hypotheses

There is a close connection between estimable functions and whether or not a hypothesis can be tested:

6.3.1.1 Definition (7.3.1) A hypothesis such as $H_0 : \beta_1 = \beta_2$ is called *testable* if there exists linearly independent estimable functions $\boldsymbol{\lambda_1'\beta}$, .., $\boldsymbol{\lambda_k'\beta}$ such that H_0 is true iff

$$\boldsymbol{\lambda_1'\beta} = ... = \boldsymbol{\lambda_k'\beta} = 0$$

6.3.1.2 Example (7.3.2) say we have the model

$$y_i = \mu + \alpha_i + \beta_j + \epsilon_{ij}$$

with i=1,2,3;j=1,2,3.

and we want to test $H_0 : \alpha_1 = \alpha_2 = \alpha_3$. We find

$$\boldsymbol{X\beta} = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mu \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \beta_1 \\ \beta_2 \\ \beta 3 \end{pmatrix} = \begin{pmatrix} \mu + \alpha_1 + \beta_1 \\ \mu + \alpha_1 + \beta_2 \\ \mu + \alpha_1 + \beta_3 \\ \mu + \alpha_2 + \beta_1 \\ \mu + \alpha_2 + \beta_2 \\ \mu + \alpha_2 + \beta_3 \\ \mu + \alpha_1 + \beta_1 \\ \mu + \alpha_2 + \beta_2 \\ \mu + \alpha_3 + \beta_3 \end{pmatrix}$$

Now

$$(\mu + \alpha_1 + \beta_1) - (\mu + \alpha_2 + \beta_1) = \alpha_1 - \alpha_2$$

so $\alpha_1 - \alpha_2$ is an estimable function. Also

$$(\mu + \alpha_1 + \beta_1) + (\mu + \alpha_2 + \beta_1) - 2(\mu + \alpha_3 + \beta_1) = \alpha_1 + \alpha_2 - 2\alpha_3$$

and so $\alpha_1 + \alpha_2 - 2\alpha_3$ is an estimable function. But

$$\begin{aligned} \alpha_1 - \alpha_2 &= 0 \\ \alpha_1 + \alpha_2 - 2\alpha_3 &= 0 \end{aligned}$$

iff $\alpha_1 = \alpha_2 = \alpha_3$. Therefore H_0 is testable and is equivalent to

$$H_0 : \begin{pmatrix} \alpha_1 - \alpha_2 \\ \alpha_1 + \alpha_2 - 2\alpha_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

So if we want to test $H_0 : \beta_1 = .. = \beta_q$ we can find a set of linearly independent estimable functions such that $H_0 : \beta_1 = .. = \beta_q$ is equivalent to

$$H_0 : \boldsymbol{\gamma}_1 = \begin{pmatrix} \boldsymbol{\lambda_1'\beta} \\ \boldsymbol{\lambda_2'\beta} \\ \vdots \\ \boldsymbol{\lambda_l'\beta} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

It is also possible to find

$$\boldsymbol{\gamma}_2 = \begin{pmatrix} \boldsymbol{\lambda}'_{\boldsymbol{l+1}}\boldsymbol{\beta} \\ \vdots \\ \boldsymbol{\lambda}'_{\boldsymbol{k}}\boldsymbol{\beta} \end{pmatrix}$$

such that the functions $\boldsymbol{\lambda}'_{\boldsymbol{i}}\boldsymbol{\beta}$,i=1,..,k are linearly independent and estimable. Let $\boldsymbol{\gamma} = \begin{pmatrix} \boldsymbol{\gamma}_1 \\ \boldsymbol{\gamma}_2 \end{pmatrix}$, then we can reparametrize from the non-full rank model $\boldsymbol{y = X\beta + \epsilon}$ to the full rank model

$$\boldsymbol{y = Z\gamma + \epsilon = Z_1\gamma_1 + Z_2\gamma_2 + \epsilon}$$

Now under the null hypothesis we have the reduced model $\boldsymbol{y = Z_2\gamma_2^* + \epsilon^*}$. Also

6.3.1.3 Theorem (7.3.3) Consider the partitioned model $\boldsymbol{y = X\beta + \epsilon = X_1\beta_1 + X_2\beta_2 + \epsilon}$. If $\boldsymbol{X'_2X_1 = 0}$, any estimate of $\boldsymbol{\beta_2^*}$ in the reduced model is also an estimate of $\boldsymbol{\beta_2}$ in the full model.

proof follows from (6.3.4)

So we know that the estimate of γ_2^* is the same as the estimate of γ_2 if the columns of $\boldsymbol{Z}_2$ are orthogonal to the columns of $\boldsymbol{Z}_1$, that is if $\boldsymbol{Z'_2Z_1 = 0}$. This is typically true for the balanced models considered here.

Since $\boldsymbol{y = Z\gamma + \epsilon}$ is a full-rank model, we can use the theorems of section 6.6. The details of the test are here:

Source	df	SS	F
Due to γ_1 adjusted for γ_1	l	$\text{SS}(\hat{\boldsymbol{\gamma}}_{\boldsymbol{1}}\|\hat{\boldsymbol{\gamma}}_{\boldsymbol{2}}) = \hat{\boldsymbol{\gamma}}'\boldsymbol{Z'y} - \hat{\boldsymbol{\gamma}}'_{\boldsymbol{2}}\boldsymbol{Z'_2y}$	$\frac{\text{SS}(\hat{\boldsymbol{\gamma}}_{\boldsymbol{1}}\|\hat{\boldsymbol{\gamma}}_{\boldsymbol{2}})/l}{\text{SSE}/(n-k)}$
Error	$n-k$	$\text{SSE} = \boldsymbol{y'y} - \hat{\boldsymbol{\gamma}}'\boldsymbol{Z'y}$	
Total	$n-1$	$\text{SST} = \boldsymbol{y'y} - n\bar{y}^2$	

The difficulty with this approach is finding the matrix $\boldsymbol{Z}$. However, we also have $\text{SSE} = \boldsymbol{y'y} - \hat{\boldsymbol{\beta}}'\boldsymbol{X'y}$ and (7.2.18), so

$$\boldsymbol{y'y} - \hat{\boldsymbol{\beta}}'\boldsymbol{X'y} = \boldsymbol{y'y} - \hat{\boldsymbol{\gamma}}'\boldsymbol{Z'y}$$

or

$$\hat{\boldsymbol{\beta}}'\boldsymbol{X'y} = \hat{\boldsymbol{\gamma}}'\boldsymbol{Z'y}$$

The same is true for the reduced model obtained by setting $\beta_1 = .. = \beta_l$, and so we have the ANOVA table for testing $H_0 : \beta_1 = .. = \beta_l$ in balanced non-full rank models:

Source	df	SS	F
Due to β_1 adjusted for β_1	l	$\hat{\boldsymbol{\beta}}'\boldsymbol{X}'\boldsymbol{y} - \hat{\boldsymbol{\beta}}_2'\boldsymbol{X}_2'\boldsymbol{y}$	$\frac{\text{SS}(\hat{\boldsymbol{\beta}}_1 \mid \hat{\boldsymbol{\beta}}_2)/l}{\text{SSE}/(n-k)}$
Error	$n-k$	$\text{SSE} = \boldsymbol{y}'\boldsymbol{y} - \hat{\boldsymbol{\beta}}'\boldsymbol{X}'\boldsymbol{y}$	
Total	$n-1$	$\text{SST} = \boldsymbol{y}'\boldsymbol{y} - n\bar{y}^2$	

6.3.2 General Linear Hypothesis

6.3.2.1 Theorem (7.3.4) If $\boldsymbol{y} \sim N_n(\boldsymbol{X}\boldsymbol{\beta}, \sigma^2\boldsymbol{I})$, where $\boldsymbol{X}$ is $n \times p$ pf rank $k < p \leq n$, if $\boldsymbol{C}$ is $m \times p$ of rank $m \leq k$ such that $\boldsymbol{C}\boldsymbol{\beta}$ is a set of m linearly independent estimable functions, and if $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^-\boldsymbol{X}'\boldsymbol{y}$, then

i. $\boldsymbol{C}(\boldsymbol{X}'\boldsymbol{X})^-\boldsymbol{C}'$ is nonsingular

ii. $\boldsymbol{C}\hat{\boldsymbol{\beta}} \sim N_m[\boldsymbol{C}\boldsymbol{\beta}, \sigma^2\boldsymbol{C}(\boldsymbol{X}'\boldsymbol{X})^-\boldsymbol{C}']$

iii. $\text{SSH}/\sigma^2 = (\boldsymbol{C}\hat{\boldsymbol{\beta}})'\boldsymbol{C}(\boldsymbol{X}'\boldsymbol{X})^-\boldsymbol{C}']^{-1}\boldsymbol{C}\hat{\boldsymbol{\beta}}'/\sigma^2 \sim \chi^2(m, \lambda)$ where $\lambda = (\boldsymbol{C}\boldsymbol{\beta})'\boldsymbol{C}(\boldsymbol{X}'\boldsymbol{X})^-\boldsymbol{C}']^{-1}\boldsymbol{C}\boldsymbol{\beta}'/(2\sigma^2)$

iv. $\text{SSE}/\sigma^2 = \boldsymbol{y}[\boldsymbol{I} - \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^-\boldsymbol{X}']\boldsymbol{y}/\sigma^2 \sim \chi^2(n-k)$

v. SSH and SSE are independent

proof omitted

6.3.2.2 Theorem (7.3.5) Under the conditions of the above theorem if $H_0 : \boldsymbol{C}\boldsymbol{\beta} = \boldsymbol{0}$ is true

$$F = \frac{\text{SSH}/m}{\text{SSE}/(n-k)} \sim F(m, n-k)$$

proof see (5.4.6)

6.3.3 A Simple Model

As an example of all of the above consider an additive two-way model without interactions and without repetitions:

$$y_{ij} = \mu + \alpha_i + \beta_j + \epsilon_{ij}$$

i=1,2,3 and j=1,2

Let's use R for some of the work:

```
make.X=function(I, J) {
  X=NULL
  for(i in 1:I) {
    tmp=matrix(0, J, I)
    tmp[, i]=1
    X=rbind(X, cbind(tmp, diag(J)))
  }
  X=cbind(1, X)
  X
}
X=make.X(3, 2)
qr(X)$rank
```

```
## [1] 4
```

```
t(X)%*%X
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    6    2    2    2    3    3
## [2,]    2    2    0    0    1    1
## [3,]    2    0    2    0    1    1
## [4,]    2    0    0    2    1    1
## [5,]    3    1    1    1    3    0
## [6,]    3    1    1    1    0    3
```

say we want to test $H_0 : \alpha_1 = \alpha_2 = \alpha_3$. This can be written as

$$H_0 : \alpha_1 - \alpha_2 = 0 \text{ and } \alpha_1 - \alpha_3 = 0$$

so H_0 is testable if $\alpha_1 - \alpha_2 = 0$ and $\alpha_1 - \alpha_3 = 0$ are estimable.

Let's write

$$\alpha_1 - \alpha_2 = \begin{pmatrix} 0 & 1 & -1 & 0 & 0 & 0 \end{pmatrix} \boldsymbol{\beta} = \boldsymbol{\lambda_1 \beta}$$

then

$$\begin{pmatrix} 0 & 1/2 & -1/2 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 6 & 2 & 2 & 2 & 3 & 3 \\ 2 & 2 & 0 & 0 & 1 & 1 \\ 2 & 0 & 2 & 0 & 1 & 1 \\ 2 & 0 & 0 & 2 & 1 & 1 \\ 3 & 1 & 1 & 1 & 3 & 0 \\ 3 & 1 & 1 & 1 & 0 & 3 \end{pmatrix} = \begin{pmatrix} 0 & 1 & -1 & 0 & 0 & 0 \end{pmatrix}$$

Similarly we have

$$\alpha_1 - \alpha_3 = \begin{pmatrix} 0 & 1 & 0 & -1 & 0 & 0 \end{pmatrix} \boldsymbol{\beta} = \boldsymbol{\lambda_2 \beta}$$

and

$$\begin{pmatrix} 0 & 1/2 & 0 & -1/2 & 0 & 0 \end{pmatrix} \boldsymbol{X'X} = \begin{pmatrix} 0 & 1 & 0 & -1 & 0 & 0 \end{pmatrix}$$

If we need a complete set of linearly independent estimable functions we can subtract rows of $\boldsymbol{X}$. One solution is

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mu \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} \mu + \alpha_1 + \beta_1 \\ \beta_1 - \beta_2 \\ \alpha_1 - \alpha_2 \\ 0 \\ \alpha_1 - \alpha_3 \\ 0 \end{pmatrix} = \mathbf{0}$$

As we needed two estimable functions to express the null hypothesis, the sum of squares will have two degrees of freedom.

The normal equations are given by

$$\begin{pmatrix} 6 & 2 & 2 & 2 & 3 & 3 \\ 2 & 2 & 0 & 0 & 1 & 1 \\ 2 & 0 & 2 & 0 & 1 & 1 \\ 2 & 0 & 0 & 2 & 1 & 1 \\ 3 & 1 & 1 & 1 & 3 & 0 \\ 3 & 1 & 1 & 1 & 0 & 3 \end{pmatrix} \begin{pmatrix} \hat{\mu} \\ \hat{\alpha_1} \\ \hat{\alpha_2} \\ \hat{\alpha_3} \\ \hat{\beta_1} \\ \hat{\beta_2} \end{pmatrix} = \begin{pmatrix} y_{..} \\ y_{1.} \\ y_{2.} \\ y_{3.} \\ y_{.1} \\ y_{.2} \end{pmatrix}$$

adding the side conditions $\hat{\alpha}_1 + \hat{\alpha}_2 + \hat{\alpha}_3 = 0$ and $\hat{\beta}_1 + \hat{\beta}_2 = 0$ we find the solutions

$$\begin{aligned} \hat{\mu} &= \bar{y}_{..} \\ \hat{\alpha}_1 &= \bar{y}_{1.} - \bar{y}_{..} \\ \hat{\alpha}_2 &= \bar{y}_{2.} - \bar{y}_{..} \\ \hat{\alpha}_3 &= \bar{y}_{3.} - \bar{y}_{..} \\ \hat{\beta}_1 &= \bar{y}_{.1} - \bar{y}_{..} \\ \hat{\beta}_2 &= \bar{y}_{.2} - \bar{y}_{..} \end{aligned}$$

For the test $H_0 : \alpha_1 = \alpha_2 = \alpha_3$ we need $\hat{\boldsymbol{\beta}}\boldsymbol{X'y}$, which we denote by

$$\text{SS}(\mu, \alpha, \beta) = \hat{\boldsymbol{\beta}}\boldsymbol{X'y} =$$

$$\begin{pmatrix} \hat{\mu} & \hat{\alpha_1} & \hat{\alpha_2} & \hat{\alpha_3} & \hat{\beta_1} & \hat{\beta_2} \end{pmatrix} = \begin{pmatrix} y_{..} \\ y_{1.} \\ y_{2.} \\ y_{3.} \\ y_{.1} \\ y_{.2} \end{pmatrix} =$$

$$\bar{y}_{..}y_{..} + \sum_{i=1}^{3}(\bar{y}_{i.} - \bar{y}_{..})y_{i.} + \sum_{j=1}^{2}(\bar{y}_{.j} - \bar{y}_{..})y_{.j} =$$

$$y_{..}^2/6 + \sum_{i=1}^{3}(y_{i.}/2 - y_{..}/6)y_{i.} + \sum_{j=1}^{2}(y_{.j}/3 - y_{..}/6)y_{.j} =$$

$$y_{..}^2/6 + \left(\sum_{i=1}^{3} y_{i.}^2/2 - y_{..}^2/6\right) + \left(\sum_{j=1}^{2} y_{.j}^2/3 - y_{..}^2/6\right)$$

where we used the fact that $\sum_i y_{i.} = \sum_j y_{.j} = y_{..}$.

For the error sum of squares we find

$$\boldsymbol{y'y} - \hat{\boldsymbol{\beta}}\boldsymbol{X'y} =$$

$$\sum_{i,j} y_{ij}^2 - y_{..}^2/6 + \left(\sum_{i=1}^{3} y_{i.}^2/2 - y_{..}^2/6\right) + \left(\sum_{j=1}^{2} y_{.j}^2/3 - y_{..}^2/6\right)$$

Next we need $\hat{\boldsymbol{\beta}}\boldsymbol{X_2'y}$ for the reduced model

$$y_{ij} = \mu + \beta_j + \epsilon_{ij}$$

where μ is $\mu + \alpha$.

Now the normal equations are found as follows

```
X2=X[, -c(2:4)]
X2
```

```
##      [,1] [,2] [,3]
## [1,]    1    1    0
## [2,]    1    0    1
## [3,]    1    1    0
## [4,]    1    0    1
## [5,]    1    1    0
## [6,]    1    0    1
```

```
t(X2)%*%X2
```

```
##      [,1] [,2] [,3]
## [1,]    6    3    3
```

```
## [2,]    3    3    0
## [3,]    3    0    3
```

and adding the side condition $\hat{\beta}_1 + \hat{\beta}_2 = 0$ we have the system of equations

$$\begin{aligned} 6\hat{\mu} + 3\hat{\beta}_1 + 3\hat{\beta}_2 &= y_{..} \\ 3\hat{\mu} + 3\hat{\beta}_1 &= y_{.1} \\ \hat{\beta}_1 + \hat{\beta}_2 &= 0 \end{aligned}$$

which we solve:

$$\begin{aligned} & 6\hat{\mu} + 3(\hat{\beta}_1 + \hat{\beta}_2) = 6\hat{\mu} = y_{..} \\ & \hat{\mu} = \bar{y}_{..} \\ & \text{I-2*II} : -3\hat{\beta}_1 + 3\hat{\beta}_2 = y_{..} - 2y_{.1} \\ & \quad -3\hat{\beta}_1 - 3\hat{\beta}_1 = y_{..} - 2y_{.1} \\ & \hat{\beta}_1 = (2y_{.1} - y_{..})/6 = \bar{y}_{.1} - \bar{y}_{..} \\ & \hat{\beta}_2 = -(\bar{y}_{.1} - \bar{y}_{..}) = \bar{y}_{.2} - \bar{y}_{..} \end{aligned}$$

$$\begin{aligned} & \text{SS}(\mu, \beta) = \hat{\boldsymbol{\beta}}\boldsymbol{X_2'y} = \\ & \begin{pmatrix} \hat{\mu} & \hat{\beta}_1 & \hat{\beta}_2 \end{pmatrix} \begin{pmatrix} y_{..} \\ y_{.1} \\ y_{.2} \end{pmatrix} = \\ & \bar{y}_{..}y_{..} + \sum_{j=1}^{2}(\bar{y}_{.j} - \bar{y}_{..})y_{.j} = \\ & y_{..}^2/6 + \sum_{i=1}^{3}(y_{i.}/2 - y_{..}/6)y_{i.} + \sum_{j=1}^{2}(y_{.j}/3 - y_{..}/6)y_{.j} = \\ & y_{..}^2/6 + \left(\sum_{j=1}^{2} y_{.j}^2/3 - y_{..}^2/6\right) = \end{aligned}$$

Finally

$$\text{SS}(\alpha|\mu, \beta) = \hat{\boldsymbol{\beta}}\boldsymbol{X'y} - \hat{\boldsymbol{\beta}}_{\boldsymbol{2}}\boldsymbol{X_2'y} = \sum_{i=1}^{3} y_{i.}^2/2 - y_{..}^2/6$$

and we summarize the test for $H_0 : \alpha_1 = \alpha_2 = \alpha_3$ in the ANOVA table

Source	df	SS	F
Due to α adjusted for β, μ	2	$\text{SS}(\alpha\|\mu,\beta) = \sum_i y_{i.}^2/2 - y_{..}^2/6$	$\frac{\text{SS}(\alpha\|\mu,\beta)/2}{\text{SSE}/2}$
Error	2	$\text{SSE} = \boldsymbol{y'y} - \hat{\boldsymbol{\beta}}\boldsymbol{'X'y}$	
Total	5	$\text{SST} = \boldsymbol{y'y} - n\bar{y}^2$	

6.3.3.1 Example (7.3.6) Let's generate some data and see how this works. We do this so that the null hypothesis is true in y_1 and false in y_2:

```
make.X=function(I, J) {
  X=NULL
  for(i in 1:I) {
    tmp=matrix(0, J, I)
    tmp[, i]=1
    X=rbind(X, cbind(tmp, diag(J)))
  }
  X=cbind(1, X)
  X
}
X=make.X(3, 2)
X
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    1    1    0    0    1    0
## [2,]    1    1    0    0    0    1
## [3,]    1    0    1    0    1    0
## [4,]    1    0    1    0    0    1
## [5,]    1    0    0    1    1    0
## [6,]    1    0    0    1    0    1
```

```
n=nrow(X)
k=qr(X)$rank
beta1=rbind(5, 0, 0, 0, -2, 2)
beta2=rbind(5, -3, -1, 4, -2, 2)
epsilon=rnorm(6, 0, 1)
y1=cbind(round(X%*%beta1+epsilon, 3))
y2=cbind(round(X%*%beta2+epsilon, 3))

# Find ydots and betahats
I=3;J=2
ydots1=c(t(X)%*%y1)
tmp=ydots1/apply(X, 2, sum)
tmp[-1]=tmp[-1]-tmp[1]
betahat1=tmp
```

```
ydots2=c(t(X)%*%y2)
tmp=ydots2/apply(X, 2, sum)
tmp[-1]=tmp[-1]-tmp[1]
betahat2=tmp
rbind(round(betahat1, 3), round(betahat2, 3))
```

```
##       [,1]   [,2]   [,3]   [,4]   [,5]  [,6]
## [1,] 4.648  0.365 -0.129 -0.236 -1.975 1.976
## [2,] 4.648 -2.636 -1.129  3.764 -1.975 1.976
```

```
# Find sse
sse1=t(y1)%*%y1-rbind(betahat1)%*%t(X)%*%y1
sse2=t(y2)%*%y2-rbind(betahat2)%*%t(X)%*%y2
round(c(sse1, sse2), 5)
```

```
## [1] 1.12942 1.12942
```

```
# Find salpha given mu, beta
ssalpha1=sum(ydots1[1+1:I]^2/J)-ydots1[1]^2/(I*J)
ssalpha2=sum(ydots2[1+1:I]^2/J)-ydots2[1]^2/(I*J)
round(c(ssalpha1, ssalpha2), 3)
```

```
## [1]  0.410 44.784
```

```
# Find F statistics
F1=(ssalpha1/(I-1))/(sse1/(n-k))
F2=(ssalpha2/(I-1))/(sse2/(n-k))
# Find p  values
round(c(F1, F2), 1)
```

```
## [1]  0.4 39.7
```

```
c(1-pf(F1, I-1, n-k), 1-pf(F2, I-1, n-k))
```

```
## [1] 0.73370245 0.02459893
```

and so we correctly reject the second case but not the first.

6.3.3.2 Example (7.3.7) Let's apply this to the hearing aid data. Instead of deriving the equations all over we do this by analogy:

```
I=4;J=24
X=make.X(I, J)
n=nrow(X)
k=qr(X)$rank
k
```

```
## [1] 27
```

```
y=as.matrix(hearingaid[, 3, drop=FALSE])
ydots=c(t(X)%*%y)
```

```
tmp=ydots/apply(X, 2, sum)
tmp[-1]=tmp[-1]-tmp[1]
betahat=tmp
round(betahat, 3)
```

```
##  [1]  28.312   4.438   1.354  -3.062  -2.729  -3.812  -4.312  -0.312  -7.812   2.688
## [12]  -1.312   0.188   8.188   2.188  -0.312   2.688   3.688   7.688   5.188  -1.312
## [23]  -8.812   9.688  -6.812 -11.312  -6.312  -1.312  11.688
```

```
# Find sse
sse=t(y)%*%y-rbind(betahat)%*%t(X)%*%y
round(sse/c(1, n-k), 3)
```

```
## [1] 2506.542   36.327
```

```
# Find salpha given mu, beta
ssalpha=sum(ydots[1+1:I]^2/J)-ydots[1]^2/(I*J)
round(ssalpha/c(1, I-1), 3)
```

```
## [1] 920.458 306.819
```

```
# Find F statistics
FTS=(ssalpha/(I-1))/(sse/(n-k))
round(FTS, 3)
```

```
##       Score
## Score 8.446
```

```
# Find p  value
1-pf(FTS, I-1, n-k)
```

```
##              Score
## Score 7.412012e-05
```

and so we reject the null hypothesis, the lists are not all equal.

How about letting R do all the work?

```
summary(aov(Score~as.factor(Subject)+as.factor(List), data=hearingaid))
```

```
##                    Df Sum Sq Mean Sq F value   Pr(>F)
## as.factor(Subject) 23   3232  140.51   3.868 6.96e-06
## as.factor(List)     3    920  306.82   8.446 7.41e-05
## Residuals          69   2507   36.33
```

and so we have found all the relevant numbers!

6.4 One-way ANOVA

6.4.1 The One-Way Model

We have a model of the form

$$y_{ij} = \mu + \alpha_i + \epsilon_{ij}$$

i=1,..,k and j=1,..,n_i

Also let $n = \sum_i n_i$ be the sample size.

We have the assumptions

i. $E[\epsilon_{ij}] = 0$ for all i,j

ii. $var(\epsilon_{ij}) = \sigma^2$ for all i,j

iii. $cov(\epsilon_{ij}, \epsilon_{rs}) = 0$ for all $(i,j) \neq (r,s)$

Often we also have

iv. $\epsilon_{ij} \sim N_n(0, \sigma^2)$
and

v. $\sum_i \alpha_i = 0$

The one-way layout is sometimes also called a *completely randomized design*.

6.4.2 Estimable Functions

For illustration let's study the case k=3, n_i=2. Then

$$\boldsymbol{y} = \begin{pmatrix} y_{11} \\ y_{12} \\ y_{21} \\ y_{22} \\ y_{31} \\ y_{32} \end{pmatrix} = \begin{pmatrix} \mu + \alpha_1 \\ \mu + \alpha_1 \\ \mu + \alpha_2 \\ \mu + \alpha_2 \\ \mu + \alpha_3 \\ \mu + \alpha_3 \end{pmatrix} + \begin{pmatrix} \epsilon_{11} \\ \epsilon_{12} \\ \epsilon_{21} \\ \epsilon_{22} \\ \epsilon_{31} \\ \epsilon_{32} \end{pmatrix} =$$

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mu \\ \alpha_1 \\ \alpha_2 \\ \alpha_2 \end{pmatrix} + \begin{pmatrix} \epsilon_{11} \\ \epsilon_{12} \\ \epsilon_{21} \\ \epsilon_{22} \\ \epsilon_{31} \\ \epsilon_{32} \end{pmatrix}$$

and $\boldsymbol{X}$ is 6×4 of rank 3. Again the vectors $\boldsymbol{\beta} = \begin{pmatrix} \mu & \alpha_1 & \alpha_2 & \alpha_2 \end{pmatrix}'$. In general, $\boldsymbol{X}$ for a one-way balanced design is $n \times (k+1)$ of rank k.

We have seen before that in general contrasts are estimable, that is $\sum c_i \alpha_i$ is estimable if $\sum c_i = 0$.

If we add some side conditions and denote the constrained parameters as $\mu^*, \alpha_1^*, ..\alpha_k^*$, then these are estimable.

6.4.2.1 Theorem (7.4.1) Under the side condition $\sum \alpha_i^* = 0$ the estimators are $\mu^* = \bar{\mu}_.$ and $\alpha_i^* = \mu_i - \bar{\mu}_.$, where $\mu_i = \mu + \alpha_i$ and $\bar{\mu}_. = \frac{1}{k}\sum_i \mu_i$.

proof

$$y_{ij} = \mu + \alpha_i + \epsilon_{ij} = \mu_i + \epsilon_{ij}$$
$$\bar{y}_. = \frac{1}{k}\sum_i \mu_i = \frac{1}{k}\sum_i (\mu^* + \alpha_i^*) =$$
$$\mu^* + \frac{1}{k}\sum_i \alpha_i^* = \mu^*$$
$$\alpha_i^* = \mu_i - \bar{\mu}_.$$

6.4.3 Parameter Estimation

The general one-way model has the design matrix

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{j}_{n_1} & \boldsymbol{j}_{n_1} & \mathbf{0} & ... & \mathbf{0} \\ \boldsymbol{j}_{n_2} & \mathbf{0} & \boldsymbol{j}_{n_3} & ... & \mathbf{0} \\ \vdots & \vdots & \vdots & & \vdots \\ \boldsymbol{j}_{n_k} & \mathbf{0} & \mathbf{0} & ... & \boldsymbol{j}_{n_k} \end{pmatrix}$$

where $\boldsymbol{j}_{n_i}$ is the vector of 1's of length n_i.

The normal equations are

$$\boldsymbol{X'X\hat{\beta} = X'y} \begin{pmatrix} n & n_1 & 0 & 0 & ... & 0 \\ n_1 & n_2 & 0 & 0 & ... & 0 \\ n_2 & 0 & n_3 & 0 & ... & 0 \\ \vdots & \vdots & \vdots & \vdots & ... & \vdots \\ n_k & 0 & 0 & 0 & ... & n_k \end{pmatrix} \begin{pmatrix} \hat{\mu} \\ \hat{\alpha}_1 \\ \vdots \\ \hat{\alpha}_k \end{pmatrix} = \begin{pmatrix} y_{..} \\ y_{1.} \\ \vdots \\ y_{k.} \end{pmatrix}$$

Adding the usual side condition we have the system of equations

$$n\hat{\mu} + n_1\hat{\alpha}_1 + .. + n_k\hat{\alpha}_k = y_{..}$$
$$n\hat{\mu} + n_i\hat{\alpha}_i = y_{i.}; \; i = 1, .., k$$
$$\sum n_i\hat{\alpha}_i = 0$$

and so

$$y_{..} = \sum_{i=1}^{k} y_{i.} = \sum_{i=1}^{k}(n_i\hat{\mu} + n_i\hat{\alpha}_i) = n\hat{\mu} + \sum_{i=1}^{k} n_i\hat{\alpha}_i = n\hat{\mu}$$
$$\hat{\mu} = y_{..}/n = \bar{y}_{..}$$
$$\hat{\alpha}_i = y_{i.}/n_i - \hat{\mu} = \bar{y}_{i.} - \bar{y}_{..}$$

6.4.3.1 Example (7.4.2)

In the mothers and cocaine use data we find:

```
y=mothers$Length
x=mothers$Status
ni=table(x)
n=sum(ni)
y.. = sum(y)
yi. = tapply(y, x, sum)
round(c(y../n, yi./ni-y../n), 3)
```

```
##                         Drug Free First Trimester      Throughout
##          49.549           1.551          -0.249          -1.549
```

6.4.3.2 Example (7.4.3)

The flammability of fabric used in children's sleepwear is tested by placing a flame under a piece of fabric until it ignites. The flame is then removed, and the fabric stops burning. The length of the charred portion of the fabric is measured. In the data set pieces of the same cloth were sent to five different laboratories, which then carried out this experiment eleven times.

Research Question: Are there differences between the labs?

```
kable.nice(head(flammability), do.row.names = FALSE)
```

Labs	Length
Lab 1	2.9
Lab 1	3.1
Lab 1	3.1
Lab 1	3.7
Lab 1	3.1
Lab 1	4.2

```
table(flammability$Labs)
```

```
##
## Lab 1 Lab 2 Lab 3 Lab 4 Lab 5
##    11    11    11    11    11
```

so here we have a balanced one-way design with k=5 and n_i=11, i=1,..,k. Now

```
y=flammability$Length
x=flammability$Labs
ni=rep(11, 5)
n=sum(ni)
k=5
y..=sum(y)
yi.=tapply(y, x, sum)
round(c(y../n, yi./ni-y../n), 3)
```

```
##          Lab 1  Lab 2  Lab 3  Lab 4  Lab 5
##  3.376 -0.040  0.224 -0.076 -0.376  0.269
```

As previously discussed another approach to estimation is to use the generalized inverse. We can easily verify that a generalized inverse of $\boldsymbol{X'X}$ is given by

$$(\boldsymbol{X'X})^- = \begin{pmatrix} 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1/n_1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1/n_2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1/n_k \end{pmatrix}$$

and so a solution to the normal equations is given by

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X'X})^-\boldsymbol{X'y} = \begin{pmatrix} 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1/n_1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1/n_2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1/n_k \end{pmatrix} \begin{pmatrix} \boldsymbol{j}_{n_1} & \boldsymbol{j}_{n_1} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{j}_{n_2} & \boldsymbol{0} & \boldsymbol{j}_{n_3} & \dots & \boldsymbol{0} \\ \vdots & \vdots & \vdots & & \vdots \\ \boldsymbol{j}_{n_k} & \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{j}_{n_k} \end{pmatrix} \begin{pmatrix} y_{11} \\ \vdots \\ y_{kn} \end{pmatrix} = \begin{pmatrix} 0 \\ \bar{y}_{1\cdot} \\ \vdots \\ \bar{y}_{k\cdot} \end{pmatrix}$$

6.4.4 An estimator of σ^2

6.4.4.1 Definition (7.4.4) An unbiased estimator for σ^2 is given by $s^2 = \text{SSE}/[k(n-1])$, where

$$\text{SSE} = \boldsymbol{y'y} - \hat{\boldsymbol{\beta}}\boldsymbol{X'y} = \boldsymbol{y'}[\boldsymbol{I} - \boldsymbol{X}(\boldsymbol{X'X})^-\boldsymbol{X'}]\boldsymbol{y}$$

Alternative formulas are

$$s^2 = \frac{\sum_{ij} y_{ij}^2 - \sum_i y_{i\cdot}^2/n_i}{n-k} = \frac{\sum_{ij}(y_{ij} - \bar{y}_{i\cdot})^2}{n-k}$$

6.4.4.2 Example (7.4.5) For the children's ware data we find

```
n=55;k=5
ybar = tapply(flammability$Length, flammability$Labs, mean)
sse=sum( (flammability$Length-rep(ybar,each=11))^2 )
sse/(n-k)
```

```
## [1] 0.1646545
```

6.4.5 Hypothesis Testing

Say we want to test $H_0 : \mu_1 = .. = \mu_k$. As always for testing we now also assume

$$\boldsymbol{y} \sim N_n(\boldsymbol{X\beta}, \sigma^2 \boldsymbol{I})$$

Adding the usual side condition the null hypothesis becomes

$$H_0 : \alpha_1 = .. = \alpha_k = 0$$

Under the null hypothesis the reduced model is $\boldsymbol{y} = \mu \boldsymbol{j} + \boldsymbol{\epsilon}$.
For the full model we have

$$\mathrm{SS}(\mu, \alpha) = \hat{\boldsymbol{\beta}} \boldsymbol{X'y} = \sum_i y_{i.}^2 / n_i$$

and for the reduced model we find

$$\hat{\mu} = (\boldsymbol{j_n' j_n})^{-1} \boldsymbol{j_n' y} = \bar{y}_{..}$$

$$\mathrm{SS}(\mu) = \hat{\mu} \boldsymbol{j_n' y} = y_{..}^2 / n$$

By the ANOVA table just before theorem (9.3.3) the sum of squares for the α's adjusted for μ is given by

$$\mathrm{SS}(\alpha|\mu) = \mathrm{SS}(\alpha, \mu) - \mathrm{SS}(\mu) = \sum_i y_{i.}^2 / n_i - y_{..}^2 / n$$

all this is summarized in the

ANOVA table for one-way design

Source	df	SS	F
Treatments	$k-1$	$\mathrm{SS}(\alpha\|\mu) = \sum_i y_{i.}^2/n_i - y_{..}^2/n$	$\frac{\mathrm{SS}(\alpha\|\mu)/(k-1)}{\mathrm{SSE}/[k(n-1)]}$
Error	$n-k$	$\mathrm{SSE} = \sum_{ij}(y_{ij} - \bar{y}_{i.})^2$	
Total	$n-1$	$\mathrm{SST} = \sum_{ij} y_{ij}^2 - y_{..}^2/n$	

6.4.5.1 Example (7.4.6)

In the mothers and cocaine use data we find:

```
y=mothers$Length
x=mothers$Status
k=3
ni=table(x)
n=sum(ni)
y..=sum(y)
yi.=tapply(y, x, sum)
sse=sum(y^2)-sum(yi.^2/ni)
ssalpha=sum(yi.^2/ni)-y..^2/n
FTS=(ssalpha/(k-1))/(sse/(n-k))
round(c(ssalpha/c(1, k-1), sse/c(1, n-k), FTS, 1-pf(FTS, k-1, k*(n-1))), 3)
```

```
## [1] 181.375  90.687 885.580   9.732   9.319   0.000
```

or of course

```
summary(aov(y~x))
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## x            2  181.4   90.69   9.319 0.000208
## Residuals   91  885.6    9.73
```

and so we find that the babies do not have equal lengths.

6.4.5.2 Example (7.4.7)

For the children's sleep ware data:

```
y=flammability$Length
x=flammability$Labs
ni=rep(11, 5)
n=sum(ni)
k=5
y..=sum(y)
yi.=tapply(y, x, sum)
sse=sum(y^2)-sum(yi.^2/ni)
ssalpha=sum(yi.^2/ni)-y..^2/n
FTS=(ssalpha/(k-1))/(sse/(n-k))
round(c(ssalpha/c(1, k-1), sse/c(1, n-k), FTS, 1-pf(FTS, k-1, k*(n-1))), 3)
```

```
## [1] 2.987 0.747 8.233 0.165 4.535 0.001
```

or of course

```
summary(aov(y~x))
```

```
##             Df Sum Sq Mean Sq F value  Pr(>F)
## x            4  2.987  0.7466   4.535 0.00334
## Residuals   50  8.233  0.1647
```

and so we find that the Labs do not have equal means.

6.4.6 Contrasts

In the case of a one-way model a contrast can be written in terms of the α_i's or the μ_i's:

$$\sum_i c_i\mu_i = \sum_i c_i(\mu + \alpha_i) = \mu\sum_i c_i + \sum_i c_i\alpha_i = \sum_i c_i\alpha_i$$

Tests of contrasts are essentially comparisons of means:

$H_0 : 2\mu 1 - \mu_2 - mu_3 = 0 \leftrightarrow H_0 : \mu_1 = \frac{\mu_2+\mu_3}{2}$

To test these hypotheses we can use theorem (9.3.5). Here m=1 and the test statistic becomes

$$F = \frac{(\boldsymbol{c}'\hat{\boldsymbol{\beta}})[\boldsymbol{c}'(\boldsymbol{X}'\boldsymbol{X})^{-}\boldsymbol{c}]^{-1}\boldsymbol{c}'\hat{\boldsymbol{\beta}}}{\text{SSE}/(n-k)} = \frac{(\sum_i c_i\bar{y_i}.)^2}{s^2\sum_i c_i^2/n_i}$$

where $s^2 = \text{SSE}/(n-k)$

6.4.6.1 Example (7.4.8)

Let's have a look at the children's sleep ware data:

```
y=flammability$Length
x=factor(flammability$Labs)
ni=rep(11, 5)
n=sum(ni)
k=5
ybar=tapply(y, x, mean)
ybar
```

```
##    Lab 1    Lab 2    Lab 3    Lab 4    Lab 5
## 3.336364 3.600000 3.300000 3.000000 3.645455
```

so it seems that mean of Lab 1 is equal to the mean of Lab 3 but Lab 4 is different from Lab 5.

Let's test whether the mean of Lab 1 is equal to the mean of Lab 3. So we have $H_0 : \mu_1 = \mu_3$ or $H_0 : \mu_1 - \mu_3 = 0$ or

$$H_0 : \begin{pmatrix} 1 & 0 & -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \end{pmatrix} = 0$$

```
sse=sum( (y-rep(ybar,each=11))^2 )
cc=c(1, 0, -1, 0, 0)
num=sum(cc*ybar)^2
s2=sse/(n-k)
denom=s2*sum(cc^2/ni)
```

```
FTS=num/denom
round(c(FTS, 1-pf(FTS, 1, n-k)), 4)
```

```
## [1] 0.0442 0.8344
```

and we fail to reject the null hypothesis, the means of Lab 1 and Lab 3 are the same.

We can also use R to test for contrasts:

```
library(multcomp)
fit=aov(y~x)
fit.gh=glht(fit, linfct = mcp(x = cc))
summary(fit.gh)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: User-defined Contrasts
##
##
## Fit: aov(formula = y ~ x)
##
## Linear Hypotheses:
##        Estimate Std. Error t value Pr(>|t|)
## 1 == 0  0.03636    0.17302    0.21    0.834
## (Adjusted p values reported -- single-step method)
```

Note that because the F distribution has 1 degree of freedom the F test is also a t-test, and

```
(0.21)^2
```

```
## [1] 0.0441
```

similarly testing Lab 4 vs. Lab 5 we find

```
cc=c(0, 0, 0, 1, -1)
num=sum(cc*ybar)^2
s2=sse/(n-k)
denom=s2*sum(cc^2/ni)
FTS=num/denom
round(c(FTS, 1-pf(FTS, 1, n-k)), 4)
```

```
## [1] 13.9162  0.0005
```

and now we reject the null hypothesis.

and again with Rs:

```
fit.gh=glht(fit, linfct = mcp(x = cc))
summary(fit.gh)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
```

```
##
## Multiple Comparisons of Means: User-defined Contrasts
##
##
## Fit: aov(formula = y ~ x)
##
## Linear Hypotheses:
##        Estimate Std. Error t value Pr(>|t|)
## 1 == 0  -0.6455     0.1730   -3.73 0.000489
## (Adjusted p values reported -- single-step method)
```

Note that because the F distribution has 1 degree of freedom the F test is also a t-test, and

```
(-3.73)^2
```

```
## [1] 13.9129
```

6.4.7 Orthogonal Contrasts

6.4.7.1 Definition (7.4.9) Two contrasts $\boldsymbol{c_i'\hat{\beta}}$ and $\boldsymbol{c_j'\hat{\beta}}$ are *orthogonal* iff $\boldsymbol{c_i'c_j} = 0$.

Note that

$$cov(\boldsymbol{c_i'\hat{\beta}}, \boldsymbol{c_j'\hat{\beta}}) = \boldsymbol{c_i'(X'X)^-c_j} = \boldsymbol{c_i'}diag(0, 1/n, .., 1/n)\boldsymbol{c_j} = \boldsymbol{c_i'c_j}/n = 0$$

iff $\boldsymbol{c_i'c_j} = 0$. But we also have the normal assumption, and so two contrasts are orthogonal iff $\boldsymbol{c_i'c_j} = 0$.

(Here we added a 0 to all c's)

A similar argument shows that the sums of squares $(\boldsymbol{c_i'\hat{\beta}})^2/\boldsymbol{c_i'}(X'X)^-c_i$ and $(\boldsymbol{c_j'\hat{\beta}})^2/\boldsymbol{c_i'}(X'X)^-c_j$ are also independent.

6.4.7.2 Theorem (7.4.10) In the balanced one-way model, if $\boldsymbol{y} \sim N_n(\boldsymbol{X\beta}, \sigma^2\boldsymbol{I})$ and if $H_0 : \alpha_1 = .. = \alpha_k$ is written as $\boldsymbol{C\beta} = 0$ where the rows of $\boldsymbol{C} = (c_i')$, i=1,..k-1, are mutually orthogonal contrasts, then

$$\text{SSH} = \boldsymbol{(C\hat{\beta})'[C(X'X)^-C']C\hat{\beta}} = \sum_{i=1}^{k-1} \frac{(c_i'\hat{\beta})^2}{c_i'(X'X)^-c_i}$$

and the sums of squares $c_i'(X'X)^-c_i$ are independent.

proof

By the calculation above $C(X'X)^-C'$ is a diagonal matrix with $c_i'(X'X)^-c_i$ on the diagonal, which shows the second equality. The first follows from the independence of the orthogonal contrasts.

Notice a consequence of this theorem

6.4.7.3 Corollary (7.4.11)

Let F be the test statistic of the overall test, then

$$F = \frac{\text{SSH}/(k-1)}{s^2} = \frac{1}{k-1}\sum_{i=1}^{k-1}\frac{(c_i'\hat{\beta})^2}{c_i'(X'X)^{-}c_i} = \frac{1}{k-1}\sum_{i=1}^{k-1}F_i$$

where F_i is the test statistic for the i$^{\text{th}}$ contrast.

6.4.7.4 Example (7.4.12)

For the children's sleep ware data:

There are a number of standard orthogonal contrasts. One of them is

```
C=t(contr.helmert(5))
C
```

```
##      1  2  3  4 5
## [1,] -1  1  0  0 0
## [2,] -1 -1  2  0 0
## [3,] -1 -1 -1  3 0
## [4,] -1 -1 -1 -1 4
```

this compares 1-2, the means of 1,2 with 3, and so on.

```
C%*%t(C)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    2    0    0    0
## [2,]    0    6    0    0
## [3,]    0    0   12    0
## [4,]    0    0    0   20
```

shows that these are indeed orthogonal contrasts.

Let's apply this to our data, but let's order the the groups first by their means

```
x=factor(flammability$Labs, levels = levels(x)[order(ybar)],
         ordered = TRUE)
ybar=sort(ybar)
FTS=rep(0, 4)
for(i in 1:4) {
   num=n*sum(C[i, ]*ybar)^2
   denom=s2*sum(C[i, ]^2)
   FTS[i]=num/denom
}
FTS
```

```
## [1] 15.031471  7.734283 37.691402 30.234099
```

```
pvalue=1-pf(FTS, 1, k*(n-1))
round(cbind(FTS, pvalue), 3)
```

```
##         FTS pvalue
## [1,] 15.031  0.000
```

```
## [2,]  7.734  0.006
## [3,] 37.691  0.000
## [4,] 30.234  0.000
```

```
FTS.all=sum(FTS)/(k-1)
round(c(FTS.all, 1-pf(FTS.all, k-1, k*(n-1))), 3)
```

```
## [1] 22.673  0.000
```

and we see that the overall test is the same as before, in (9.2.5)

```
sqrt(FTS)
```

```
## [1] 3.877044 2.781058 6.139332 5.498554
```

```
fit=aov(y~x)
fit.gh=glht(fit, linfct = mcp(x = C))
summary(fit.gh)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: User-defined Contrasts
##
##
## Fit: aov(formula = y ~ x)
##
## Linear Hypotheses:
##        Estimate Std. Error t value Pr(>|t|)
## 1 == 0   0.3000     0.1730   1.734   0.3065
## 2 == 0   0.3727     0.2997   1.244   0.6210
## 3 == 0   1.1636     0.4238   2.746   0.0328
## 4 == 0   1.3455     0.5471   2.459   0.0672
## (Adjusted p values reported -- single-step method)
```

Notice the last line of the printout: *Adjusted p values reported – single-step method)*: because these tests are independent on can use the Bonfferoni adjustment, so what is reported here is the *family-wise* error rate. However

```
round((k-1)*pvalue,4)
```

```
## [1] 0.0005 0.0232 0.0000 0.0000
```

shows that the method used by R is slightly different.

6.5 Pairwise Comparisons

After the basic ANOVA test and rejecting the null hypothesis of no differences one generally wants to go a step further and investigate what the differences are.

6.5.0.1 Example We rejected the null hypothesis of equal lengths of the babies in the mother and cocaine use experiment. But

- is there a stat. signif. difference between drug free and 1st trimester?
- is there a stat. signif. difference between 1st trimester and throughout?

Notice that in this case we have a natural ordering of the levels, so there are two pairs of interest. In general if a factor has k levels there are $\binom{n}{k}$ pairs.

Comparing two groups could be done with the two-sample t test, but doing $\binom{n}{k}$ tests we again have the issue of simultaneous inference, as discussed in section 6.7. The same general solutions proposed there (Bonferroni's method, Scheffe's method and p-value adjustment via simulation) can be applied here as well. In addition we have some methods specifically designed for pairwise comparisons in ANOVA.

Note an often made suggestion is to do pairwise comparisons at the $\alpha = 10\%$ level rather than 5%.

6.5.1 Bonferroni's Method

This method uses the basic 2-sample t test but with the adjusted significance level α/k if k comparisons are done.

6.5.2 Sheffe's Method

We already discussed this method in the context of regression in (6.5.2). There we saw that all hypotheses of the form $H_0 : \boldsymbol{a'\beta} = 0$ can be tested simultaneously with

$$F_M = \frac{(\boldsymbol{a'\hat{\beta}})^2}{s^2 \boldsymbol{a'}(\boldsymbol{X'X})^{-1}\boldsymbol{a}}$$

using $F/(k+1) \sim F_M(k+1, n-k-1)$ if k tests are done.

A pairwise comparison is a contrast of the form $\boldsymbol{a} = (0 \,..\, 0\ 1\ 0 \,..\, 0\ -1\ 0 \,..\, 0)'$, so $\boldsymbol{a'\hat{\beta} = \bar{y}_i - \bar{y}_j}$ and

$$\boldsymbol{a}'(\boldsymbol{X}'\boldsymbol{X})^{-}\boldsymbol{a} =$$

$$\boldsymbol{a}'\begin{pmatrix} 0 & 0 & 0 & 0 & ... & 0 \\ 0 & 1/n_1 & 0 & 0 & ... & 0 \\ 0 & 0 & 1/n_2 & 0 & ... & 0 \\ \vdots & \vdots & \vdots & \vdots & ... & \vdots \\ 0 & 0 & 0 & 0 & ... & 1/n_k \end{pmatrix}\begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ -1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} =$$

$$(0\ ..\ 0\ 1\ 0\ ..\ 0\ \ -1\ 0\ ..\ 0)'\begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1/n_1 \\ 0 \\ \vdots \\ 0 \\ -1/n_j \\ 0 \\ \vdots \\ 0 \end{pmatrix} =$$

$$1/n_i + 1/n_j$$

and so

$$F_M = \frac{(\bar{y}_i - \bar{y}_j)^2}{s^2\,(1/n_i + 1/n_j)}$$

6.5.2.1 Example In the mothers and cocaine use data we find:

```
y=mothers$Length
x=mothers$Status
k=3
ni=table(x)
n=sum(ni)
y..=sum(y)
yi.=tapply(y, x, sum)
sse=sum(y^2)-sum(yi.^2/ni)
s2=sse/(n-k)
F12= (yi.[2]/ni[2]-yi.[1]/ni[1])^2/s2/((1/ni[2]+1/ni[1]))
F23= (yi.[3]/ni[3]-yi.[2]/ni[2])^2/s2/((1/ni[3]+1/ni[2]))
```

```
FTS=c(F12, F23)
names(FTS)=c("Drug Free - First Trimester", "First Trimester - Throughout")
round(1-pf(FTS, 1, n-k), 3)
```

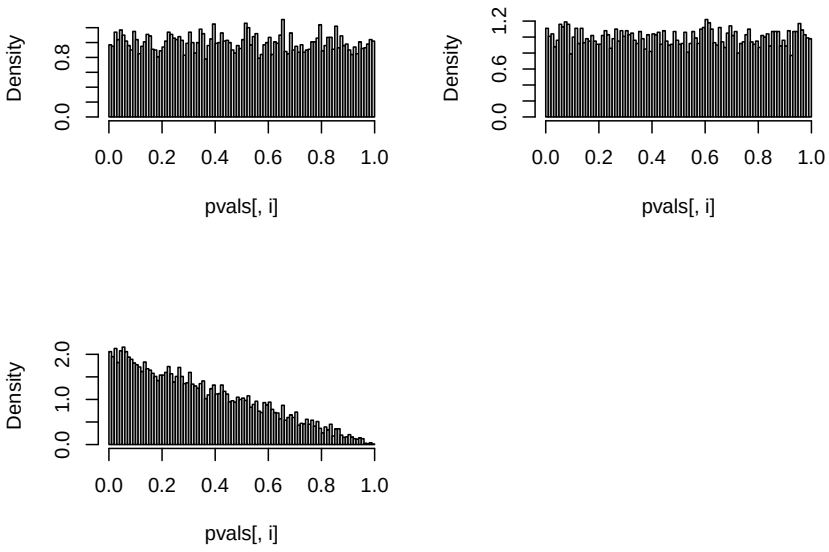
```
##  Drug Free - First Trimester First Trimester - Throughout
##                        0.042                        0.145
```

and so we find a difference between Drug Free and First Trimester but not between First Trimester and Throughout, although this is almost certainly due to the small sample sizes.

6.5.3 Adjusted p value

The idea is to find the distribution of the smallest p value via simulation and then adjust the actual p values accordingly:

```
B=10000
mu=mean(y)
sig=sd(y)
TS=function(x,y) t.test(x,y)$p.value
pvals = matrix(0, B, 3)
for(i in 1:B) {
   pvals[i, 1]=TS(rnorm(ni[1], mu, sig),rnorm(ni[2], mu, sig))
   pvals[i, 2]=TS(rnorm(ni[2], mu, sig),rnorm(ni[3], mu, sig))
}
pvals[, 3]=apply(pvals[, 1:2], 1, min)
par(mfrow=c(2, 2))
for(i in 1:3) hist(pvals[,i], 100, freq=FALSE, main="")
```



so we see that while the p values of the individual tests have a uniform [0,1] distribution, the

minimum p-value does not. We can now use the empirical distribution function to adjust the p-values:

```
data.pvals=c(TS(y[x=="Drug Free"], y[x=="First Trimester"]),
             TS(y[x=="Throughout"], y[x=="First Trimester"]))
data.pvals
```

```
## [1] 0.01917369 0.12397785
```

```
adj.pvals=ecdf(pvals[, 3])(data.pvals)
names(adj.pvals)=c("Drug Free - First Trimester", "First Trimester - Throughout")
adj.pvals
```

```
##  Drug Free - First Trimester First Trimester - Throughout
##                       0.0388                       0.2407
```

and so again we find a difference between Drug Free and First Trimester but not between First Trimester and Throughout

6.5.4 Fisher's LSD Method

LSD stands for *least significant difference*. Say we wish to compare groups i and j. Now $\bar{y}_{i.}$ and $\bar{y}_{j.}$ are the respective group means and $s^2 = \text{SSE}/(n_1 + n_2 - 2)$, then the test statistic

$$T = \frac{|\bar{y}_{i.} - \bar{y}_{j.}|}{s\sqrt{1/n_1 + 1/n_2}} \sim t(1 - \alpha/2, n_1 + n_2 - 2)$$

this of course is just the standard 2-sample t test for the difference in means, assuming equal variance. Fisher's LSD method therefore does not provide an experiment-wise error rate, but is guards against to many false positives because it is run only after the F test rejects the null hypothesis. It is sometimes referred to as the *protected LSD* method.

6.5.4.1 Example We did the F test for the mothers and cocaine use, so now we can do Fisher's LSD:

```
round(t.test(mothers$Length[mothers$Status=="Drug Free"],
       mothers$Length[mothers$Status=="First Trimester"])$p.value, 3)
```

```
## [1] 0.019
```

```
round(t.test(mothers$Length[mothers$Status=="First Trimester"],
       mothers$Length[mothers$Status=="Throughout"])$p.value, 3)
```

```
## [1] 0.124
```

so again we find a difference between Drug Free and First Trimester but not between First Trimester and Throughout.

```
pairwise.t.test(mothers$Length, mothers$Status, p.adjust.method="none")
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  mothers$Length and mothers$Status
##
##                 Drug Free First Trimester
## First Trimester 0.042     -
## Throughout      4.3e-05   0.145
##
## P value adjustment method: none
```

One disadvantage of using the pairwise.t.test command is that it finds all pairwise comparisons, whereas because our levels have an ordering only two are of interest.

6.5.5 Holm's Method

Holm, S. (1979). *A simple sequentially rejective multiple test procedure.* Scandinavian Journal of Statistics. 6 (2): 65–70. describes a method to adjust p values specifically for the case of pairwise comparisons. It is implemented in R in

```
pairwise.t.test(mothers$Length, mothers$Status)
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  mothers$Length and mothers$Status
##
##                 Drug Free First Trimester
## First Trimester 0.08404   -
## Throughout      0.00013   0.14512
##
## P value adjustment method: holm
```

6.5.6 Tukey's HSD Method

Tukey's HSD (*honest significant difference*) test is based on the *studentized range*:

$$q = \frac{\max_{i,j}\{|\bar{y}_{i.} - \bar{y}_{j.}|\}}{s/\sqrt{n}}$$

Under the normal assumption it is possible to derive the distribution of q. This is implemented in

```
round(TukeyHSD(aov(Length~Status, data=mothers))$Status[,4], 3)
```

```
##  First Trimester-Drug Free       Throughout-Drug Free Throughout-First Trimester
##                      0.103                      0.000                      0.310
```

and unlike with the other methods neither difference between Drug Free and First Trimester nor between First Trimester and Throughout is statistically significant, although Drug Free and First Trimester is close.

6.5.7 Other Methods

There are many other methods that have been developed such as Newman-Keuls multiple range test, Duncan's range test and others

6.5.8 Suggested Method

Many studies have been done to investigate these methods. The general conclusion is that Tukey's HSD test provides a good performance in most cases, and is therefore the recommended method.

6.6 Power and Sample Size

6.6.1 Power of a One-Way ANOVA Test.

Say we are considering a one-way ANOVA model with k groups and the basic test

$$H_0 : \alpha_1 = .. = \alpha_k$$

By the discussion in (6.6.3) we have that if the null hypothesis is wrong the test statistic

$$F = \frac{\text{SSR}/k}{\text{SSE}/(n-k-1)}$$

has a non-central F distribution with k and n-k-1 degrees of freedom and non-centrality parameter

$$\lambda = \boldsymbol{\beta_1 X_c' X_c \beta_1}/(2\sigma^2)$$

where $\boldsymbol{X}_c$ is the centered matrix of $\boldsymbol{X}$, the design matrix without the column of 1's, and $\boldsymbol{\beta}_1$ is $\boldsymbol{\beta}$ without μ.

6.6.1.1 Example Let's illustrate this in the case of the mothers and cocaine use data:

```
y=mothers$Length
x=mothers$Status
k=3
ni=table(x)
n=sum(ni)
y..=sum(y)
```

```
yi.=tapply(y, x, sum)
sse=sum(y^2)-sum(yi.^2/ni)
s2=sse/(n-k-1)
```

```
j1=rep(1, ni[1])
j2=rep(1, ni[2])
j3=rep(1, ni[3])
X1=cbind(c(j1, rep(0, ni[2]+ni[3])),
         c(rep(0, ni[1]), j2, rep(0, ni[3])),
         c(rep(0, ni[1]+ni[2]), j3))
Xc=(diag(n)-matrix(1, n ,n )/n)%*%X1
Xcc=t(Xc)%*%Xc
```

Finally we need β_1, that is a vector of true values. Let's check the data

```
yi./ni-y../n
```

```
##       Drug Free First Trimester      Throughout
##       1.5510638      -0.2489362      -1.5489362
```

Let's find the power if $\boldsymbol{\beta}_1 = \begin{pmatrix} 1.5 & 0 & -1.5 \end{pmatrix}'$:

```
beta=rbind(1.5, 0, -1.5)
lambda=t(beta)%*%Xcc%*%beta/2/s2
crit=qf(0.95, k-1, n-k-1)
round(1-pf(crit, k-1, n-k-1, lambda), 3)
```

```
## [1] 0.733
```

so in this case our test had a 73% chance of correctly rejecting the null hypothesis!

6.6.2 Sample Size

Say we are planning on carrying out the following experiment: we will collect n observations each from 4 groups. Then we will do the one-way ANOVA test. We want the test to have a power of 80%. What should n be?

To answer the question we need two things:

- σ^2: we do need some idea of the population standard deviation. Often one uses some number from similar experiments, or one does a small scale pilot study. Say for our purpose we know $\sigma^2 = 0.15$.
- $\boldsymbol{\beta}_1$: here we need to decide what the smallest *effect size* of practical interest is. This is the smallest difference that would be important enough to justify the experiment. Let's say that it is δ. Then one often uses

$$\boldsymbol{\beta}_1 = \begin{pmatrix} \delta/2 & -\delta/2 & 0 & 0 \end{pmatrix}'$$

so that the difference between groups 1 and 2 is δ, and the other groups are between those. Say in our experiment $\delta = 2.3$.

So now

```
pwr <- function(n, k, delta, sigma2, alpha=0.05) {
  X1=matrix(0, k*n, k)
  for(i in 1:k) X1[((i-1)*n+1):(i*n), i]=1
  Xc=(diag(k*n)-matrix(1, k*n , k*n)/(k*n))%*%X1
  Xcc=t(Xc)%*%Xc
  beta=c(delta/2, -delta/2, rep(0, k-2))
  lambda=t(beta)%*%Xcc%*%beta/2/s2
  crit=qf(1-alpha, k-1, n*k-k-1)
  round(1-pf(crit, k-1, n*k-k-1, lambda), 4)
}
pwr(20, 4, 2.3, 0.15)
```

```
## [1] 0.2374
```

so with 20 observations per group the power is 23%, to low.

```
pwr(50, 4, 2.3, 0.15)
```

```
## [1] 0.5609
```

```
pwr(100, 4, 2.3, 0.15)
```

```
## [1] 0.8791
```

```
pwr(90, 4, 2.3, 0.15)
```

```
## [1] 0.839
```

```
pwr(80, 4, 2.3, 0.15)
```

```
## [1] 0.7884
```

```
pwr(83, 4, 2.3, 0.15)
```

```
## [1] 0.8048
```

```
pwr(82, 4, 2.3, 0.15)
```

```
## [1] 0.7994
```

so we need 83 observations per group, for a total of 332!

6.7 Balanced Two-Way ANOVA

6.7.1 The Model

In this section we will study the case of a *balanced two-way model with interaction*. Such a model can be written as

$$y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_{ij} + \epsilon_{ijk}$$

where i=1,..,I, j=1,..,J and k=1,..,n.

Here the new term γ_{ij} is called the *interaction term* and is meant to encode a dependency between the factors A and B. If these are independent we would expect $\gamma_{ij} = 0$ for all i and j.

The assumptions are the same as in a one-way ANOVA, extended to this case.

6.7.1.1 Example (7.7.1) The hearingaid data discussed in example (7.1.2) is a balanced two-way design, however by design one would not expect an interaction (dependency) between subjects and lists.

6.7.1.2 Example (7.7.2) In an experiment to study gas mileage four different blends of gasoline are tested in each of three makes of automobiles. The cars are driven a fixed distance to determine the mpg (miles per gallon) The experiment is repeated three times for each blend-automobile combination. (Taken from Lyman Ott)

Note that the interest here is indifferent gasoline blends, automobile is a blocking variable, so this is a *randomized block design*.

```
head(gasoline)
```

```
##    MPG Gasoline Automobile
## 1 22.7        1          A
## 2 22.4        1          A
## 3 22.9        1          A
## 4 21.5        2          A
## 5 21.8        2          A
## 6 21.6        2          A
```

Note that Gasoline is coded as 1, 2, 3 and 4, but these are simple codes without any meaning. So it is better to change them into factors.

```
gasoline$Gasoline = factor(gasoline$Gasoline)
```

Now

```
table(gasoline$Gasoline, gasoline$Automobile)
```

```
##
##     A B C
##   1 3 3 3
##   2 3 3 3
##   3 3 3 3
##   4 3 3 3
```

shows that each *factor-level combination* was tested three times, so here we have

I=4, J=3 and n=3

6.7.2 Interaction

Usually the first step in the analysis of a two-way design is to check for interaction. If we found $\gamma_{ij} = 0$ for all i and j, the model would simplify to an *additive model* of the form

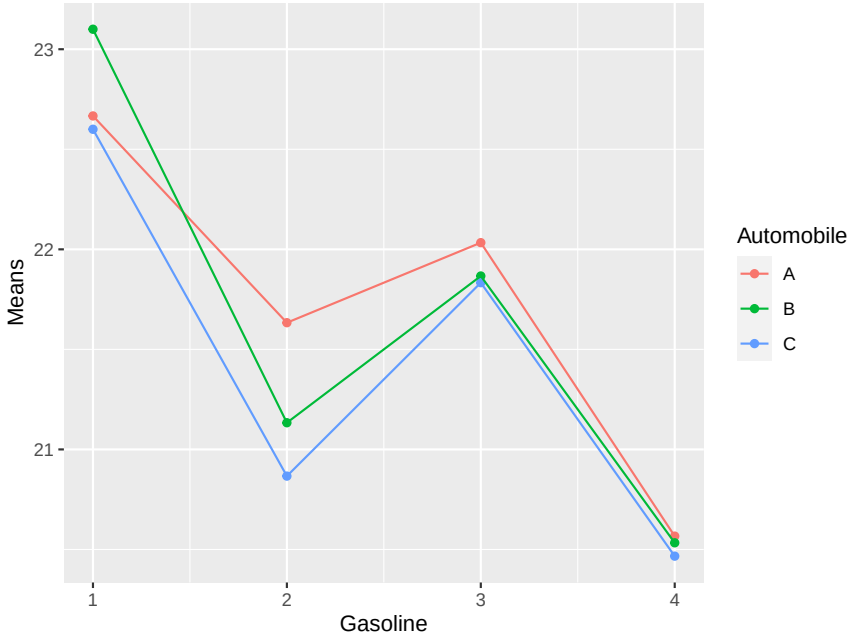
$$y_{ijk} = \mu + \alpha_i + \beta_j + \epsilon_{ijk}$$

A graphical check is to draw the *interaction plot*: here we find the mean responses for each factor-level combination $\bar{y}_{ij.}$, choose one of the factors to use on the x-axis, plot the means on the y-axis and connect the dots corresponding to the other factor.

6.7.2.1 Example (7.7.3)

For the gasoline data:

```
mns = tapply(gasoline$MPG, gasoline[, -1], mean)
df=data.frame(Gasoline=rep(1:4, 3),
              Means=c(mns),
              Automobile = rep(c("A", "B", "C"), each=4))
ggplot(data=df, aes(Gasoline, Means, color=Automobile)) +
  geom_point() +
  geom_line()
```



Now in an additive model, going from level 1 of factor A to level 2, all responses get added the same (namely $\beta_2 - \beta_1$), so the line segments should be parallel.

The problem with this approach is that each of the means is based on just three numbers, so the variance is quite high and the graphs can easily be far from parallel even if no interaction is present. It is therefore preferable to do a formal test.

6.7.3 Test for Interaction

It can be shown that estimable contrasts in γ_{ij}'s have the form

$$\gamma_{ij} - \gamma_{it} + \gamma_{sj} - \gamma_{st}$$

We illustrate the ideas in the case of I=3, J=2. Then the cell means are

$$\begin{pmatrix} \mu_{11} & \mu_{21} \\ \mu_{12} & \mu_{22} \\ \mu_{13} & \mu_{23} \end{pmatrix}$$

so if factor 1 is at level 1, the effect of factor 2 is $\mu_{11} - \mu_{12}$. At level 2 it is $\mu_{21} - \mu_{22}$ and at level 3 $\mu_{31} - \mu_{32}$. If all of these effects were 0, there would be no interaction. Notice the similarity to the discussion above on the interaction plot.

So the hypothesis of no interaction can be written as

$$H_0 : \mu_{11} - \mu_{12} = \mu_{21} - \mu_{22} = \mu_{31} - \mu_{32}$$

Now

$$\begin{aligned} &\mu_{11} - \mu_{12} = \mu + \alpha_1 + \beta_1 + \gamma_{11} - (\mu + \alpha_1 + \beta_2 + \gamma_{12}) = \\ &\beta_1 - \beta_2 + \gamma_{11} - \gamma_{12} = \\ &\mu_{21} - \mu_{22} = \beta_1 - \beta_2 + \gamma_{21} - \gamma_{22} \end{aligned}$$

and so $\mu_{11} - \mu_{12} = \mu_{21} - \mu_{22}$ implies $\beta_1 - \beta_2 + \gamma_{11} - \gamma_{12} = \beta_1 - \beta_2 + \gamma_{21} - \gamma_{22}$ or

$$\gamma_{11} - \gamma_{12} - \gamma_{21} + \gamma_{22} = 0$$

It can be shown that this is an estimable contrast. Similarly we find

$$\gamma_{21} - \gamma_{22} - \gamma_{31} + \gamma_{32} = 0$$

and again this is an estimable contrast. Therefore the interaction hypothesis is testable.

To get a reduced model we reparametrize the model as follows: let

$$\gamma^*_{ij} = \mu_{ij} - \bar{\mu}_{.j} - \bar{\mu}_{i.} + \bar{\mu}_{..}$$

which is estimable.

6.7.3.1 Theorem (7.7.4) Consider the model

$$y_{ijk} = \mu^* + \alpha_i^* + \beta_j^* + \gamma_{ij}^* + \epsilon_{ijk}$$

where

$$\begin{aligned} \mu^* &= \bar{y}_{..} \\ \alpha_i^* &= \bar{y}_{i.} - \bar{y}_{..} \\ \beta^* &= \bar{y}_{.j} - \bar{y}_{..} \end{aligned}$$

In this model the no interaction hypothesis $H_0 : \gamma_{ij}^* = 0$ for all i,j is equivalent to

$$H_0 : \mu_{11} - \mu_{12} = \mu_{21} - \mu_{22} = \mu_{31} - \mu_{32}$$

which in turn is equivalent to

$$H_0 : \begin{pmatrix} \gamma_{11} - \gamma_{12} - \gamma_{21} + \gamma_{22} \\ \gamma_{21} - \gamma_{22} - \gamma_{31} + \gamma_{32} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

proof

We will show this in the case of I=3, J=2. Consider for example $\gamma_{12}^* = \mu_{12} - \bar{\mu}_{.2} - \bar{\mu}_{1.} + \bar{\mu}_{..}$. Under $H_0 : \gamma_{12}^* = 0$, and so

$$0 = \mu_{12} - \bar{\mu}_{.2} - \bar{\mu}_{1.} + \bar{\mu}_{..} =$$

$$\mu_{12} - \frac{1}{I}\sum_i \mu_{i2} - \frac{1}{J}\sum_j \mu_{1j} + \frac{1}{IJ}\sum_{i,j} \mu_{ij} =$$
$$\mu + \alpha_1 + \beta_2 + \gamma_{12} -$$
$$\frac{1}{I}\sum_i (\mu + \alpha_i + \beta_2 + \gamma_{ij}) -$$
$$\frac{1}{J}\sum_j (\mu + \alpha_1 + \beta_j + \gamma_{ij}) +$$
$$\frac{1}{IJ}\sum_{ij} (\mu + \alpha_i + \beta_j + \gamma_{ij}) =$$

$$\mu + \alpha_1 + \beta_2 + \gamma_{12} -$$
$$\frac{1}{I}\left(I\mu + \sum_i \alpha_i - I\beta_2 + \sum_i \gamma_{ij}\right) -$$
$$\frac{1}{J}\left(J\mu + J\alpha_1 + \sum_j \beta_j + \sum_j \gamma_{ij}\right) +$$
$$\frac{1}{IJ}\left(IJ\mu + J\sum_i \alpha_i + I\sum_j \beta_j + \sum_{ij} \gamma_{ij}\right) =$$
$$\frac{1}{IJ}\left(IJ\mu + IJ\alpha_1 + IJ\beta_2 + IJ\gamma_{12}\right.$$
$$- IJ\mu - J\sum_i \alpha_i - IJ\beta_2 - J\sum_i \gamma_{ij}$$
$$- IJ\mu - IJ\alpha_1 - I\sum_j \beta_j - I\sum_j \gamma_{ij} +$$
$$\left. IJ\mu + J\sum_i \alpha_i + I\sum_j \beta_j + \sum_{ij} \gamma_{ij}\right) =$$

$$\frac{1}{IJ}\left(IJ\alpha_1 - J\sum_i \alpha_i - IJ\alpha_1 + J\sum_i \alpha_i\right.$$
$$+ IJ\beta_2 - IJ\beta_2 - I\sum_j \beta_j + I\sum_j \beta_j$$
$$\left. + IJ\gamma_{12} - J\sum_i \gamma_{i2} - I\sum_j \gamma_{1j} + \sum_{ij} \gamma_{ij}\right) =$$

$$\gamma_{12} - \frac{1}{I}\sum_i \gamma_{i2} - \frac{1}{J}\sum_j \gamma_{1j} + \frac{1}{IJ}\sum_{ij} \gamma_{ij} =$$
$$\gamma_{12} - \bar{\gamma}_{1.} - \bar{\gamma}_{.2} + \bar{\gamma}_{..}$$

and so $\gamma_{12} = \bar{\gamma}_{1.} + \bar{\gamma}_{.2} - \bar{\gamma}_{..}$

In the same way we can show that

$$\begin{aligned}\gamma_{11} &= \bar{\gamma}_{1.} + \bar{\gamma}_{.1} - \bar{\gamma}_{..} \\ \gamma_{21} &= \bar{\gamma}_{2.} + \bar{\gamma}_{.1} - \bar{\gamma}_{..} \\ \gamma_{12} &= \bar{\gamma}_{2.} + \bar{\gamma}_{.2} - \bar{\gamma}_{..}\end{aligned}$$

and so

$$\gamma_{11} - \gamma_{12} - \gamma_{21} + \gamma_{22} = \bar{\gamma}_{1.} + \bar{\gamma}_{.1} - \bar{\gamma}_{..} - \bar{\gamma}_{1.} - \bar{\gamma}_{.2} + \bar{\gamma}_{..} - \bar{\gamma}_{2.} - \bar{\gamma}_{.1} + \bar{\gamma}_{..} + \bar{\gamma}_{2.} + \bar{\gamma}_{.2} - \bar{\gamma}_{..} = 0$$

The sum of squares for the test $H_0 : \gamma^*_{ij} = 0$ is given by

$$\mathrm{SS}(\gamma|\mu, \alpha, \beta) = \mathrm{SS}(\mu, \alpha, \beta, \gamma) - \mathrm{SS}(\mu, \alpha, \beta)$$

Using the usual side conditions $\sum_i \hat{\alpha}_i = 0$, $\sum_j \hat{\beta}_j = 0$, $\sum_i \hat{\gamma}_{ij} = 0$ and $\sum_j \hat{\gamma}_{ij} = 0$, a solution to the normal equation is given by

$$\begin{aligned}\hat{\mu} &= \bar{y}_{...} \\ \hat{\alpha}_i &= \bar{y}_{i..} - \bar{y}_{...} \\ \hat{\beta}_j &= \bar{y}_{.j.} - \bar{y}_{...} \\ \hat{\gamma}_{ij} &= \bar{y}_{ij.} - \bar{y}_{i..} - \bar{y}_{.j.} + -\bar{y}_{...}\end{aligned}$$

and so

$$\begin{aligned}&\mathrm{SS}(\mu, \alpha, \beta, \gamma) = \hat{\boldsymbol{\beta}}' \boldsymbol{X}' \boldsymbol{y} = \\ &\hat{\mu} y_{...} + \sum_{i=1}^{I} \hat{\alpha}_i y_{i..} + \sum_{i=j}^{J} \hat{\beta}_j y_{.j.} + \sum_{i=1}^{I} \sum_{j=1}^{J} \hat{\gamma}_{ij} y_{ij.} = \\ &\bar{y}_{...} y_{...} + \sum_{i=1}^{I} (\bar{y}_{i..} - \bar{y}_{...})\, y_{i..} + \\ &\sum_{j=1}^{J} (\bar{y}_{.j.} - \bar{y}_{...})\, y_{.j.} + \sum_{i=1}^{I} \sum_{j=1}^{J} (\bar{y}_{ij.} - \bar{y}_{i..} - \bar{y}_{.j.} + -\bar{y}_{...})\, y_{ij.} = \\ &\frac{1}{n} \sum_{i,j} y^2_{ij.}\end{aligned}$$

a similar calculation for the reduced model yields

$$\mathrm{SS}(\mu, \alpha, \beta) = \frac{y^2_{...}}{nIJ} + \left(\sum_i \frac{y^2_{i..}}{nJ} - \frac{y^2_{...}}{nIJ} \right) + \left(\sum_j \frac{y^2_{.j.}}{nI} - \frac{y^2_{...}}{nIJ} \right)$$

and so

$$\text{SS}(\gamma|\mu,\alpha,\beta) = \sum_{ij}\frac{y_{ij.}^2}{n} - \sum_i \frac{y_{i..}^2}{nJ} - \sum_j \frac{y_{.j.}^2}{nI} + \frac{y_{...}^2}{nIJ}$$

As always

$$\text{SSE} = \boldsymbol{y'y} - \boldsymbol{\hat{\beta}X'y} = \sum_{ijk} y_{ijk}^2 - \sum_{ij}\frac{y_{ij.}^2}{n}$$

There are IJ parameters involved in the null hypothesis. However, the side conditions impose I+J-1 restrictions. So the degrees of freedom for $\text{SS}(\gamma|\mu,\alpha,\beta)$ are (I-1)(J-1).

6.7.3.2 Theorem (7.7.5)

To test $H_0 : \gamma_{ij}^* = 0$ use

$$F = \frac{\text{SS}(\gamma|\mu,\alpha,\beta)/[(I-1)(J-1)]}{\text{SSE}/[IJ(n-1)]} \sim F((I-1)(J-1), IJ(n-1))$$

proof all of the above

Notice the degrees of freedom of SSE: IJ(n-1). This shows that the test is only possible if we have repeated measurements.

6.7.3.3 Example (7.7.6)

Let's do the interaction test for the gasoline data:

```
y=gasoline$MPG
I=4;J=3;n=3
yij. = tapply(gasoline$MPG, gasoline[, -1], sum)
yi.. = tapply(gasoline$MPG, gasoline[, 2], sum)
y.j. = tapply(gasoline$MPG, gasoline[, 3], sum)
y...= sum(y)
sse=sum(y^2)-sum(yij.^2)/n
ssgamma=sum(yij.^2)/n-sum(yi..^2)/(n*J)-sum(y.j.^2)/(n*I)+y...^2/(n*I*J)
FTS=(ssgamma/((I-1)*(J-1)))/(sse/(I*J*(n-1)))
round(c(sse/c(1, I*J*(n-1)), ssgamma/c(1, (I-1)*(J-1)), FTS, 1-pf(FTS, (I-1)*(J-1), I*J
```

```
## [1] 2.247 0.094 0.909 0.151 1.618 0.185
```

or

```
gasoline$Gasoline=factor(gasoline$Gasoline)
fit=aov(MPG~.^2 , data=gasoline)
summary(fit)
```

```
##                    Df Sum Sq Mean Sq F value   Pr(>F)
## Gasoline            3 25.405   8.468  90.464 3.21e-13
## Automobile          2  0.527   0.263   2.813   0.0799
```

```
## Gasoline:Automobile  6  0.909   0.151   1.618   0.1854
## Residuals           24  2.247   0.094
```

6.7.4 Tests for Main Effects

While it is possible to test for individual factors in the presence of interaction, it is usually better practice to first test for interaction and if such is found to be statistically significant, not to test for individual factors.

As before we consider the reparametrized model:

$$\begin{aligned}
&\alpha_i^* = \bar{\mu}_{i.} - \bar{\mu}_{..} = \\
&\frac{1}{J}\sum_j \mu_{ij} - \frac{1}{IJ}\sum_{ij} \mu_{ij} = \\
&\frac{1}{J}\sum_j \left(\mu_{ij} - \frac{1}{I}\sum_i \mu_{ij}\right) = \\
&\frac{1}{J}\sum_j (\mu_{ij} - \bar{\mu}_{.j})
\end{aligned}$$

Now $\mu_{ij} - \bar{\mu}_{.j}$ is the effect of the i$^{\text{th}}$ level of factor A at the j$^{\text{th}}$ level of factor B. So α_i^* is the mean effect of the i$^{\text{th}}$ level of A. We therefore have the side condition $\sum_i \alpha_i^* = 0$. So now

$$H_0 : \alpha_1 = ... = \alpha_I$$

is equivalent to

$$H_0 : \alpha_1^* = ... = \alpha_I^* = 0$$

and this hypothesis is testable.

By their definition this is equal to

$$H_0 : \bar{\mu}_{1.} - \bar{\mu}_{..} = ... = \bar{\mu}_{I.} - \bar{\mu}_{..}$$

which is obviously equal to

$$H_0 : \bar{\mu}_{1.} = ... = \bar{\mu}_{I.}$$

Under the null hypothesis $H_0 : \alpha_1^* = ... = \alpha_I^* = 0$ the reduced model is

$$y_{ijk} = \mu^* + \beta_j^* + \gamma_{ij}^* + \epsilon_{ijk}$$

In the balanced case this model is orthogonal, so the estimators $\hat{\mu}^*$, $\hat{\beta}_j^*$ and $\hat{\gamma}_{ij}^*$, and so we find

$$\mathrm{SS}(\mu,\beta,\gamma) = \hat{\boldsymbol{\beta}}'\boldsymbol{X}_1'\boldsymbol{y} =$$
$$\hat{\mu}y_{\cdot\cdot\cdot} + \sum_{i=j}^{J}\hat{\beta}_j y_{\cdot j\cdot} + \sum_{i=1}^{I}\sum_{j=1}^{J}\hat{\gamma}_{ij}y_{ij\cdot} =$$
$$\bar{y}_{\cdot\cdot\cdot}y_{\cdot\cdot\cdot} + \sum_{j=1}^{J}\left(\bar{y}_{\cdot j\cdot} - \bar{y}_{\cdot\cdot\cdot}\right)y_{\cdot j\cdot} + \sum_{i=1}^{I}\sum_{j=1}^{J}\left(\bar{y}_{ij\cdot} - \bar{y}_{i\cdot\cdot} - \bar{y}_{\cdot j\cdot} + \bar{y}_{\cdot\cdot\cdot}\right)y_{ij\cdot} =$$
$$y_{\cdot\cdot\cdot}^2/nIJ + \sum_{j=1}^{J}\left(y_{\cdot j\cdot}^2/nI - y_{\cdot\cdot\cdot}^2/nIJ\right) +$$
$$\left(\sum_{ij} y_{ij\cdot}^2/n - \sum_{i=1} y_{i\cdot\cdot}^2/nJ - \sum_{j=1} y_{\cdot j\cdot}^2/nI + y_{\cdot\cdot\cdot}^2/nIJ\right)$$

and

$$\mathrm{SS}(\alpha|\mu,\beta,\gamma) = \mathrm{SS}(\mu,\alpha,\beta,\gamma) - \mathrm{SS}(\mu,\beta,\gamma) = \sum_{i=1} y_{i\cdot\cdot}^2/nJ - y_{\cdot\cdot\cdot}^2/nIJ$$

Putting it all together we have the

ANOVA table for balanced two-way design

Source	df	SS
Factor A	$I-1$	$\sum_{i=1} y_{i\cdot\cdot}^2/nJ - y_{\cdot\cdot\cdot}^2/nIJ$
Factor B	$J-1$	$\sum_{j=1} y_{\cdot j\cdot}^2/nI - y_{\cdot\cdot\cdot}^2/nIJ$
Interaction	$(I-1)(J-1)$	$\sum_{ij}\frac{y_{ij\cdot}^2}{n} - \sum_i \frac{y_{i\cdot\cdot}^2}{nJ} - \sum_j \frac{y_{\cdot j\cdot}^2}{nI} + \frac{y_{\cdot\cdot\cdot}^2}{nIJ}$
Error	$IJ(n-1)$	$\sum_{ijk} y_{ijk}^2 - \sum_{ij} y_{ij\cdot}^2/n$
Total	$nIJ-1$	$\sum_{ijk} y_{ijk}^2 - y_{\cdot\cdot\cdot}^2/(nIJ)$

6.7.4.1 Example (7.7.7) Let's do the test for the gasoline:

```
y=gasoline$MPG
I=4;J=3;n=3
yij. = tapply(gasoline$MPG, gasoline[, -1], sum)
yi.. = tapply(gasoline$MPG, gasoline[, 2], sum)
y.j. = tapply(gasoline$MPG, gasoline[, 3], sum)
y...= sum(y)
sse=sum(y^2)-sum(yij.^2)/n
ssalpha=sum(yi..^2)/(n*J)-y...^2/(n*I*J)
FTS=(ssalpha/(I-1))/(sse/(I*J*(n-1)))
round(c(sse/c(1, I*J*(n-1)), ssalpha/c(1, (I-1)), FTS, 1-pf(FTS, (I-1), I*J*(n-1))), 3)
```

```
## [1]  2.247  0.094 25.405  8.468 90.464  0.000
```

or again

```
gasoline$Gasoline=factor(gasoline$Gasoline)
fit=aov(MPG~.^2 , data=gasoline)
summary(fit)
```

```
##                      Df Sum Sq Mean Sq F value   Pr(>F)
## Gasoline              3 25.405   8.468  90.464 3.21e-13
## Automobile            2  0.527   0.263   2.813   0.0799
## Gasoline:Automobile   6  0.909   0.151   1.618   0.1854
## Residuals            24  2.247   0.094
```

If we previously tested for interaction and failed to reject the null hypothesis of no interaction we can (and should) instead fit an *additive model*

$$y_{ijk} = \mu + \alpha_i + \beta_j$$

and then we find

ANOVA table for balanced additive two-way design

Source	df	SS
Factor A	$I-1$	$\sum_{i=1} y_{i..}^2/nJ - y_{...}^2/nIJ$
Factor B	$J-1$	$\sum_{i=1} y_{.j.}^2/nI - y_{...}^2/nIJ$
Error	$nIJ-I-J+1$	$\sum_{ijk} y_{ijk}^2 - \sum_{i=1} y_{i..}^2/nJ - \sum_{j=1} y_{.j.}^2/nI + y_{...}^2/nIJ$
Total	$nIJ-1$	$\sum_{ijk} y_{ijk}^2 - y_{...}^2/(nIJ)$

6.7.4.2 Example (7.7.8) Let's do the test for the gasoline in the additive model:

```
ssalpha=sum(yi..^2)/(n*J) - y...^2/(n*I*J)
ssalpha/c(1, I-1)
```

```
## [1] 25.405278  8.468426
```

```
ssbeta=sum(y.j.^2)/(n*I) - y...^2/(n*I*J)
ssbeta/c(1, J-1)
```

```
## [1] 0.5266667 0.2633333
```

```
sse= sum(y^2)-sum(yi..^2)/(n*J)-sum(y.j.^2)/(n*I)+y...^2/(n*I*J)
FTS=(ssalpha/(I-1))/(sse/(n*I*J-I-J+1))
round(c(sse/c(1, n*I*J-I-J+1), ssalpha/c(1, (I-1)), FTS, 1-pf(FTS, (I-1), n*I*J-I-J+1)),
```

```
## [1]  3.156  0.105 25.405  8.468 80.510  0.000
```

```r
fit=aov(MPG~. , data=gasoline)
summary(fit)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## Gasoline     3 25.405   8.468  80.510 1.89e-14
## Automobile   2  0.527   0.263   2.504   0.0987
## Residuals   30  3.156   0.105
```

6.8 Unbalanced Two-Way ANOVA

Things begin to be rather difficult when we consider two (or more) factors with an unequal number of replications, maybe even some factor-level combinations with zero counts. In this situation there turn out to be several ways to parametrize the same experiment which can actually lead to different results, and there is no mathematical way to determine which is correct. In practice one needs to use subject matter knowledge to decide which model to use. In general when planning an experiment it is highly recommended to attempt an at least somewhat balanced design to avoid those difficulties.

In this section we will discuss one approach called the *cell means model*:

$$y_{ijk} = \mu + \alpha_i + \beta_j + \epsilon_{ijk} = \mu_{ij} + \epsilon_{ijk}$$

i=1„.,I;j=1,..J,K;k=1,..,n_{ij}

6.8.1 Unconstrained Model

Throughout this section we will use the following for purposes of illustration: I=2,J=3, n_{12}=n_{21}=1, n_{11}=n_{13}=n_{23}=2 and n_{22}=3, so $N = \sum n_{ij} = 11$. Therefore we have the equations

$$\begin{aligned}
y_{111} &= \mu_{11} + \epsilon_{111} \\
y_{112} &= \mu_{11} + \epsilon_{112} \\
y_{121} &= \mu_{12} + \epsilon_{121} \\
y_{131} &= \mu_{13} + \epsilon_{131} \\
y_{132} &= \mu_{13} + \epsilon_{132} \\
y_{211} &= \mu_{21} + \epsilon_{211} \\
y_{221} &= \mu_{22} + \epsilon_{221} \\
y_{222} &= \mu_{22} + \epsilon_{222} \\
y_{223} &= \mu_{22} + \epsilon_{223} \\
y_{231} &= \mu_{23} + \epsilon_{231} \\
y_{232} &= \mu_{23} + \epsilon_{232}
\end{aligned}$$

In matrix form the model is $\boldsymbol{y} = \boldsymbol{W}\boldsymbol{\mu} + \boldsymbol{\epsilon}$ or

$$\begin{pmatrix} y_{111} \\ y_{112} \\ y_{121} \\ y_{131} \\ y_{132} \\ y_{211} \\ y_{221} \\ y_{222} \\ y_{223} \\ y_{231} \\ y_{232} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mu_{11} \\ \mu_{12} \\ \mu_{13} \\ \mu_{21} \\ \mu_{22} \\ \mu_{23} \end{pmatrix} = \begin{pmatrix} \epsilon_{111} \\ \epsilon_{112} \\ \epsilon_{121} \\ \epsilon_{131} \\ \epsilon_{132} \\ \epsilon_{211} \\ \epsilon_{221} \\ \epsilon_{222} \\ \epsilon_{223} \\ \epsilon_{231} \\ \epsilon_{232} \end{pmatrix}$$

here $\boldsymbol{y}$ and $\boldsymbol{\epsilon}$ are 11×1 and $\boldsymbol{W}$ is 11×6, in general they are $N \times 1$ and $\boldsymbol{W}$ is $N \times IJ$.

Since $\boldsymbol{W}$ is full rank we have

$$\hat{\mu} = (\boldsymbol{W'W})^{-1}\boldsymbol{W'y}$$

Let's use R to find $\hat{\mu}$:

```
W=rbind(c(1 , 0 , 0 , 0 , 0 , 0),
c(1 , 0 , 0 , 0 , 0 , 0),
c(0 , 1 , 0 , 0 , 0 , 0),
c(0 , 0 , 1 , 0 , 0 , 0),
c(0 , 0 , 1 , 0 , 0 , 0),
c(0 , 0 , 0 , 1 , 0 , 0),
c(0 , 0 , 0 , 0 , 1 , 0),
c(0 , 0 , 0 , 0 , 1 , 0),
c(0 , 0 , 0 , 0 , 1 , 0),
c(0 , 0 , 0 , 0 , 0 , 1),
c(0 , 0 , 0 , 0 , 0 , 1))
t(W)%*%W
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    2    0    0    0    0    0
## [2,]    0    1    0    0    0    0
## [3,]    0    0    2    0    0    0
## [4,]    0    0    0    1    0    0
## [5,]    0    0    0    0    3    0
## [6,]    0    0    0    0    0    2
```

shows that $\boldsymbol{W'W}$ is a diagonal matrix with the respective sample sizes on the diagonal, so in general $(\boldsymbol{W'W})^{-1} = diag(\frac{1}{n_{11}}, .., \frac{1}{n_{IJ}})$. Of course $\boldsymbol{W'y} = (y_{11.}, .., y_{IJ.})'$, and so

$$\hat{\boldsymbol{\mu}} = \begin{pmatrix} \bar{y}_{11.} \\ \bar{y}_{12.} \\ \vdots \\ \bar{y}_{IJ.} \end{pmatrix} = \bar{\boldsymbol{y}}$$

Also

$$cov(\hat{\boldsymbol{\mu}}) = \sigma^2(\boldsymbol{W'W})^{-1} =$$
$$diag(\frac{\sigma^2}{n_{11}}, .., \frac{\sigma^2}{n_{IJ}})$$

An unbiased estimator of σ^2 is given by

$$s^2 = \frac{\text{SSE}}{\nu_E} = \frac{(\boldsymbol{y - W\hat{\mu}})'(\boldsymbol{y - W\hat{\mu}})}{N - IJ}$$

Alternatively we have

$$\text{SSE} = \boldsymbol{y'[I - W(W'W)^{-1}W']y} = \sum_{ijk}(y_{ijk} - \bar{y}_{ij})^2$$

If $n_{ij} \geq 2$ for all cells we can find the within-cell variance $s^2_{ij} = \frac{1}{n_{ij}-1}\sum_{k=1}^{n_{ij}}(y_{ijk} - \bar{y}_{ij})^2$ and the s^2 can be written as the *pooled estimator*.

$$s^2 = \frac{\sum_{ij}(n_{ij} - 1)s^2_{ij}}{N - IJ}$$

Notice that the cell means model does not have any explicit terms for main effects and/or interactions. These are "hard-coded" in to the μ_{ij}'s. If we want to test for them we have to express them as contrasts.

Let's consider the main effect of factor A. In the vector

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_{11} & \mu_{12} & \mu_{13} & \mu_{21} & \mu_{22} & \mu_{23} \end{pmatrix}'$$

the first three elements correspond to the first level of factor A, the last three to the second level. So for the main affect of A we can compare the mean of $\mu_{11}, \mu_{12}, \mu_{13}$ to the mean of $\mu_{21}, \mu_{22}, \mu_{23}$. The difference can be expressed as a contrast:

$$\boldsymbol{a'\mu} = \mu_{11} + \mu_{12} + \mu_{13} - \mu_{21} - \mu_{22} - \mu_{23}$$
$$\begin{pmatrix} 1 & 1 & 1 & -1 & -1 & -1 \end{pmatrix}\boldsymbol{\mu} =$$
$$(\mu_{11} - \mu_{21}) + (\mu_{12} - \mu_{22}) + (\mu_{13} - \mu_{23})$$

So we have $H_0 : \boldsymbol{a'\mu} = 0$ is equivalent to $H_0 : (\mu_{11} - \mu_{21}) + (\mu_{12} - \mu_{22}) + (\mu_{13} - \mu_{23}) = 0$.

Factor B has three levels, so we will need two contrasts:

$$\begin{aligned}
\boldsymbol{b_1'\mu} &= 2(\mu_{11}+\mu_{21}) - (\mu_{12}+\mu_{22}) - (\mu_{13}+\mu_{23}) = \\
&\begin{pmatrix} 2 & -1 & -1 & 2 & -1 & -1 \end{pmatrix}\boldsymbol{\mu} \\
\boldsymbol{b_2'\mu} &= (\mu_{12}+\mu_{22}) - (\mu_{13}+\mu_{23}) = \\
&\begin{pmatrix} 0 & 1 & -1 & 0 & 1 & -1 \end{pmatrix}\boldsymbol{\mu}
\end{aligned}$$

Setting $\boldsymbol{B} = \begin{pmatrix} \boldsymbol{b}_1' \\ \boldsymbol{b}_2' \end{pmatrix}$ we have $H_0 : \boldsymbol{B\mu} = 0$ is equivalent to $H_0 : \mu_{11}+\mu_{21} = \mu_{12}+\mu_{22} = \mu_{13}+\mu_{23}$.

Finally for the interaction we find the test $H_0 : \mu_{11} - \mu_{21} = \mu_{12} - \mu_{22} = \mu_{13} - \mu_{23}$

which also can be written in terms of contrasts:

$$\boldsymbol{C} = \begin{pmatrix} 2 & -1 & -1 & -2 & 1 & 1 \\ 0 & 1 & -1 & 0 & -1 & 1 \end{pmatrix}$$

Here are the corresponding tests:

- $H_0 : \boldsymbol{a'\mu} = 0$

$$F = \frac{(\boldsymbol{a'\hat{\mu}})'[\boldsymbol{a'(W'W)^{-1}a}]^{-1}(\boldsymbol{a'\hat{\mu}})}{s^2} = \frac{(\sum_{ij} a_{ij}\bar{y}_{ij.})^2}{s^2 \sum_{ij} a_{ij}^2/n_{ij}} \sim F(1, N-ab)$$

- $H_0 : \boldsymbol{B'\mu} = 0$

$$F = \frac{(\boldsymbol{B'\hat{\mu}})'[\boldsymbol{B'(W'W)^{-1}B}]^{-1}(\boldsymbol{B'\hat{\mu}})}{s^2} =\sim F(\nu_B, N-ab)$$

where ν_B is the number of rows of $\boldsymbol{B}$.

- Interaction test $H_0 : \boldsymbol{C'\mu} = 0$

$$F = \frac{(\boldsymbol{C'\hat{\mu}})'[\boldsymbol{C'(W'W)^{-1}C}]^{-1}(\boldsymbol{C'\hat{\mu}})}{s^2} =\sim F(\nu_C, N-ab)$$

where ν_C is the number of rows of $\boldsymbol{C}$.

6.8.1.1 Example Remington and Schork carried out an experiment to investigate the effects of smoking on physical activity. 21 people were classified as to their smoking habit and then randomly assigned to one of three physical activities. The time until maximum oxygen uptake in minutes was recorded. The data is

```
df=data.frame(
  Time=c(12.8,13.5,11.2,16.2,17.8,22.6,19.3,18.9,9.2,7.5,13.2,8.1,16.2,16.1,17.8),
  Smoking=factor(rep(c("None", "Heavy"), c(8, 7)),
                 levels = c("None", "Heavy"),
                 ordered = TRUE),
```

```
  Activity=factor(c(rep(c("Bycicle", "Treatmill", "Step"), c(3, 2, 3)),
                    rep(c("Bycicle", "Treatmill", "Step"), c(2,2,3))),
                  levels = c("Bycicle", "Treatmill", "Step"),
                  ordered=TRUE)
)
kable.nice(df, do.row.names = FALSE)
```

Time	Smoking	Activity
12.8	None	Bycicle
13.5	None	Bycicle
11.2	None	Bycicle
16.2	None	Treatmill
17.8	None	Treatmill
22.6	None	Step
19.3	None	Step
18.9	None	Step
9.2	Heavy	Bycicle
7.5	Heavy	Bycicle
13.2	Heavy	Treatmill
8.1	Heavy	Treatmill
16.2	Heavy	Step
16.1	Heavy	Step
17.8	Heavy	Step

The sample sizes and within cell means are

```
n=tapply(df[, 1], df[, 2:3], length)
n
```

```
##         Activity
## Smoking Bycicle Treatmill Step
##   None        3         2    3
##   Heavy       2         2    3
```

```
N=sum(n)
```

We need

```
I=2;J=3
baryij.=c(t(tapply(df[, 1], df[, 2:3], mean)))
n=c(t(n))
sse=sum( (df$Time-rep(baryij., n))^2)
s2=sse/(N-I*J)
```

- Test for Smoking:

```
a=c(1,1,1,-1,-1,-1)
FA=sum(a*baryij.)^2/s2/sum(a^2/n)
round(c(sse/c(1,N-I*J), FA, 1-pf(FA, 1, N-I*J)), 4)
```

```
## [1] 28.5767  3.1752 24.9272  0.0007
```

- Test for Activity:

```
W.Winv=diag(1/n)
B=rbind(c(2, -1, -1, 2, -1, -1),
        c(0, 1, -1, 0, 1, -1))
muhat=cbind(baryij.)
ssB=t(B%*%muhat)%*%solve(B%*%W.Winv%*%t(B))%*%(B%*%muhat)
FB=(ssB/2)/(sse/(N-I*J))
round(c(FB, 1-pf(FA, 2, N-I*J)), 4)
```

```
## [1] 27.8014  0.0002
```

- Test for Interaction:

```
C=rbind(c(2, -1, -1, -2, 1, 1),
        c(0, 1, -1, 0, -1, 1))
ssC=t(C%*%muhat)%*%solve(C%*%W.Winv%*%t(C))%*%(C%*%muhat)
FC=(ssC/2)/(sse/(N-I*J))
round(c(FC, 1-pf(FC, 2, N-I*J)), 4)
```

```
## [1] 0.7678 0.4922
```

Let's compare this with

```
summary(aov(Time~Smoking*Activity, data=df))
```

```
##                  Df Sum Sq Mean Sq F value   Pr(>F)
## Smoking           1  58.30   58.30  18.362 0.002035
## Activity          2 180.33   90.17  28.398 0.000129
## Smoking:Activity  2   4.88    2.44   0.768 0.492173
## Residuals         9  28.58    3.18
```

and we see while the error sum of squares is the same and the results are as well (both factors are significant whereas the interaction is not), the values of the F statistics are somewhat different. This is because *aov* uses a somewhat different parametrization.

6.8.2 Additive Model

If it is clear that there is no interaction between the factors (maybe because of the design of the experiment) we might wish to test an additive model. Using the cell means approach this means we want to fit the constraint model

$$\boldsymbol{y} = \boldsymbol{W}\boldsymbol{\mu} + \boldsymbol{\epsilon} \text{ subject to } \boldsymbol{C}\boldsymbol{\mu} = \boldsymbol{0}$$

such a model can be fit using Lagrange multipliers.

6.8.3 Missing Values

Sometimes it happens that $n_{ij} = 0$ for one (or a few) cells. It turns out that the cell means model is still viable, with the following change: the matrix $\boldsymbol{W}$ now has rows with all 0's, as many as there are cells with 0 counts. Therefore $\boldsymbol{W}$ is no-full rank and no longer has an inverse. However, if $(W'W)^{-1}$ is replaced by the generalized inverse $(W'W)^{-}$ the above derivation still holds.

7 Other Linear Models

7.1 Analysis of Covariance - ANCOVA

7.1.0.1 Example (8.1.1) We have data on the number of shoes sold by year and type:

```
kable.nice(head(shoesales), do.row.names = FALSE)
```

Sales	Year	Type
1539	1	Mens
12984	1	Kids
25809	1	Ladies
5742	2	Mens
30058	2	Kids
34764	2	Ladies

There are two ways to look at this problem:

- as a regression problem with response Sales and predictors Year and Type, where Type is a categorical variable, usually called a *dummy variable*.
- as a one-way ANOVA problem with some additional information, called a *covariate*. This is then called *Analysis of Covariance ANCOVA*.

From the above it is clear that we have here a blend of regression and ANOVA.

7.1.1 Regression Analysis

In order to study this as a regression problem we first need to code the categorical variable. The obvious way to do this is to assign numbers, for example Mens=1, Kids=2 and Ladies=3. This however is usually a bad idea because it does two things:

- it introduces order (Mens=1<2=Kids)
- it introduces a scale (Kid-Mens = 2-1 = 3-2 = Ladies-Kids)

The better way to do this is to introduce two dummy variables:

- $d_1 = 1$ if Mens, 0 otherwise
- $d_2 = 1$ if Kids, 0 otherwise

so if d_1+d_2=0 it has to be Ladies.

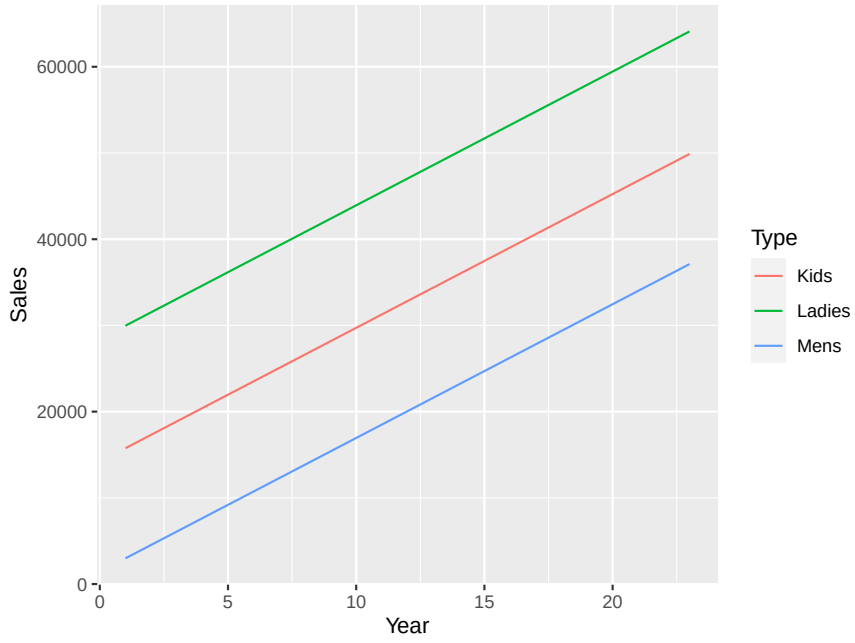
```
d1=ifelse(shoesales$Type=="Mens", 1, 0)
d2=ifelse(shoesales$Type=="Kids", 1, 0)
y=shoesales$Sales
X=data.frame(Sales=y, Year=shoesales$Year, d1=d1, d2=d2)
fit.shoe1=lm(Sales~., data=X)
summary(fit.shoe1)
```

7.1.1.1 Example (8.1.2)

```
##
## Call:
## lm(formula = Sales ~ ., data = X)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12963.7  -3433.5   -469.7   3349.1  22146.6
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  28416.0     1917.8  14.817  < 2e-16
## Year          1551.6      115.4  13.440  < 2e-16
## d1          -26986.9     1875.7 -14.388  < 2e-16
## d2          -14212.4     1875.7  -7.577 1.65e-10
##
## Residual standard error: 6361 on 65 degrees of freedom
## Multiple R-squared:  0.8565, Adjusted R-squared:  0.8498
## F-statistic: 129.3 on 3 and 65 DF,  p-value: < 2.2e-16
```

However, notice the following

```
x=seq(min(shoesales$Year), max(shoesales$Year), length=100)
z1=coef(fit.shoe1)[1]+coef(fit.shoe1)[2]*x+coef(fit.shoe1)[3]
z2=coef(fit.shoe1)[1]+coef(fit.shoe1)[2]*x+coef(fit.shoe1)[4]
z3=coef(fit.shoe1)[1]+coef(fit.shoe1)[2]*x
df=data.frame(Year=rep(x, 3),
              Sales=c(z1, z2, z3),
              Type=rep(c("Mens", "Kids", "Ladies"), each=100))
ggplot(data=df, aes(Year, Sales, color=Type)) +
  geom_line()
```



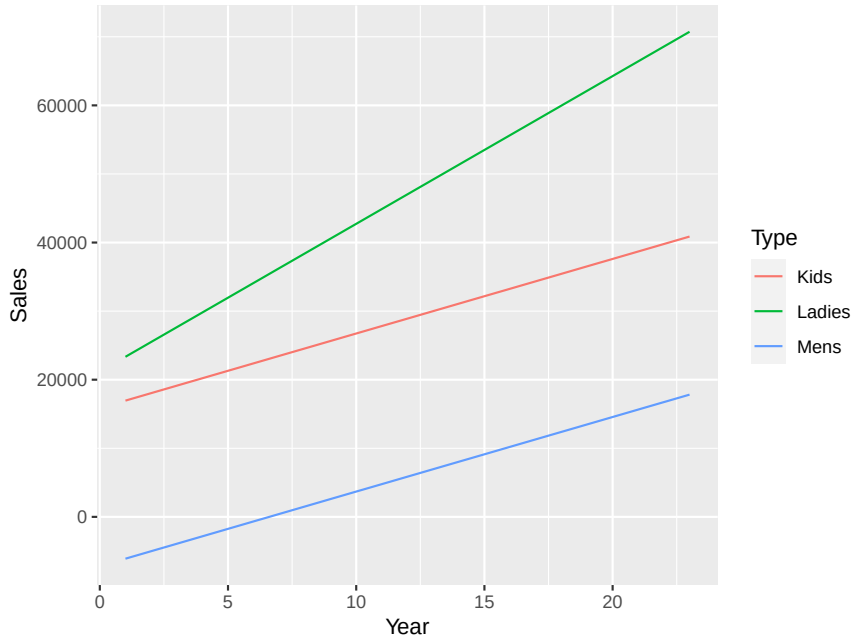
and so we see that this fits parallel lines!

Here is how to fit separate lines for each group:

```
fit.shoe2=lm(Sales~(.)^2, data=X)
summary(fit.shoe2)
```

```
##
## Call:
## lm(formula = Sales ~ (.)^2, data = X)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11588.7  -3433.0   -256.7   2947.3  16121.3
##
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  21185.6     2443.6   8.670 2.42e-12
## Year          2154.1      178.2  12.087  < 2e-16
## d1          -14185.1     3455.7  -4.105 0.000119
## d2           -5322.9     3455.7  -1.540 0.128490
## Year:d1      -1066.8      252.0  -4.233 7.64e-05
## Year:d2       -740.8      252.0  -2.939 0.004594
## d1:d2             NA         NA      NA       NA
##
## Residual standard error: 5669 on 63 degrees of freedom
## Multiple R-squared:  0.8895, Adjusted R-squared:  0.8807
## F-statistic: 101.4 on 5 and 63 DF,  p-value: < 2.2e-16
```

```
z1=coef(fit.shoe2)[1]+coef(fit.shoe2)[2]*x+coef(fit.shoe2)[3]+coef(fit.shoe2)[3]+coef(fi
z2=coef(fit.shoe2)[1]+coef(fit.shoe2)[2]*x+coef(fit.shoe2)[4]+coef(fit.shoe2)[5]*x
z3=coef(fit.shoe2)[1]+coef(fit.shoe2)[2]*x
df=data.frame(Year=rep(x, 3),
              Sales=c(z1, z2, z3),
              Type=rep(c("Mens", "Kids", "Ladies"), each=100))
ggplot(data=df, aes(Year, Sales, color=Type)) +
  geom_line()
```



7.1.2 ANCOVA Model

The model can be written as

$$\boldsymbol{y} = \boldsymbol{Z\alpha} + \boldsymbol{X\beta} + \boldsymbol{\epsilon}$$

where $\boldsymbol{Z}$ contains 0's and 1's, $\boldsymbol{\alpha}$ contains μ and parameters such as α_i, β_j and γ_{ij} representing factors and interactions; $\boldsymbol{X}$ a matrix of covariate values and $\boldsymbol{\beta}$ their coefficients.

7.1.2.1 Example (8.1.3)

For the shoesales data we have

$$\boldsymbol{y} = \boldsymbol{Z\alpha} + \boldsymbol{X\beta} + \boldsymbol{\epsilon} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \\ 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mu \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} + \begin{pmatrix} x_{11} \\ x_{12} \\ \vdots \\ x_{1n_1} \\ x_{21} \\ \vdots \\ x_{3n_3} \end{pmatrix} \beta$$

7.1.3 Estimation

We will assume that $\boldsymbol{Z}$ is less than full-rank and that $\boldsymbol{X}$ is full-rank. We can withe the model in the form

$$\begin{aligned} &\boldsymbol{y} = \boldsymbol{Z\alpha} + \boldsymbol{X\beta} + \boldsymbol{\epsilon} = \\ &\begin{pmatrix} \boldsymbol{Z} & \boldsymbol{X} \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{pmatrix} + \boldsymbol{\epsilon} = \\ &\boldsymbol{U\theta} + \boldsymbol{\epsilon} \end{aligned}$$

The normal equations are

$$\boldsymbol{U'U\hat{\theta}} = \boldsymbol{U'y}$$

which is equal to

$$\begin{pmatrix} \boldsymbol{Z'Z} & \boldsymbol{Z'X} \\ \boldsymbol{X'Z} & \boldsymbol{X'X} \end{pmatrix} \begin{pmatrix} \hat{\boldsymbol{\alpha}} \\ \hat{\boldsymbol{\beta}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{Z'y} \\ \boldsymbol{X'y} \end{pmatrix}$$

which is the two equation

$$\begin{aligned} \boldsymbol{Z'Z}\hat{\boldsymbol{\alpha}} + \boldsymbol{Z'X}\hat{\boldsymbol{\beta}} &= \boldsymbol{Z'y} \\ \boldsymbol{X'Z}\hat{\boldsymbol{\alpha}} + \boldsymbol{X'X}\hat{\boldsymbol{\beta}} &= \boldsymbol{X'y} \end{aligned}$$

using a generalized inverse for $\boldsymbol{Z'Z}$ we find

$$\hat{\boldsymbol{\alpha}} = (\boldsymbol{Z'Z})^{-}\boldsymbol{Z'y} - (\boldsymbol{Z'Z})^{-}\boldsymbol{Z'X}\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\alpha}}_0 - (\boldsymbol{Z'Z})^{-}\boldsymbol{Z'X}\hat{\boldsymbol{\beta}}$$

where $\hat{\boldsymbol{\alpha}}_0$ is a solution of the normal equations without the covariates.

To find $\hat{\boldsymbol{\beta}}$ we substitute this into the second equation:

$$\boldsymbol{X'Z}\left[(\boldsymbol{Z'Z})^{-}\boldsymbol{Z'y}-(\boldsymbol{Z'Z})^{-}\boldsymbol{Z'X}\hat{\boldsymbol{\beta}}\right]+\boldsymbol{X'X}\hat{\boldsymbol{\beta}}=\boldsymbol{X'y}$$
$$\boldsymbol{X'}\left[\boldsymbol{I}-\boldsymbol{Z}(\boldsymbol{Z'Z})^{-}\boldsymbol{Z'}\right]\boldsymbol{X}\hat{\boldsymbol{\beta}}=\boldsymbol{X'y}-\boldsymbol{X'Z}(\boldsymbol{Z'Z})^{-}\boldsymbol{Z'y}$$
$$\boldsymbol{X'}\left[\boldsymbol{I}-\boldsymbol{P}\right]\boldsymbol{X}\hat{\boldsymbol{\beta}}=\boldsymbol{X'}(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{y}$$

where $\boldsymbol{P}=\boldsymbol{Z}(\boldsymbol{Z'Z})^{-}\boldsymbol{Z'}$. Because $\boldsymbol{X}$ is a set of covariates it is (almost always) independent of $\boldsymbol{Z}$, and so $\boldsymbol{X'}\left[\boldsymbol{I}-\boldsymbol{P}\right]\boldsymbol{X}$ is non-singular. So we find

$$\hat{\boldsymbol{\beta}}=\left[\boldsymbol{X'}\left(\boldsymbol{I}-\boldsymbol{P}\right)\boldsymbol{X}\right]^{-1}\boldsymbol{X'}(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{y}=\boldsymbol{E}_{xx}^{-1}\boldsymbol{e}_{xy}$$

7.1.3.1 Example (8.1.4)

Let's find the estimates for the shoesales data. For that we need to reorder the data as well

```
shoesales=shoesales[order(shoesales$Type), ]
X=matrix(as.numeric(shoesales[ ,"Year"]),ncol=1)
y=matrix(as.numeric(shoesales[ ,"Sales"]),ncol=1)
ni=table(shoesales$Type)
ni
```

```
##
##   Kids Ladies   Mens
##     23     23     23
```

```
n=sum(ni)
Z=matrix(0, n, 4)
Z[, 1]=1
Z[1:ni[1], 2]=1
Z[(1+ni[1]):(ni[1]+ni[2]), 3]=1
Z[(1+ni[1]+ni[2]):n, 4]=1
library(MASS)
gZZ=ginv(t(Z)%*%Z)
P=Z%*%gZZ%*%t(Z)
betahat=solve(t(X)%*%(diag(n)-P)%*%X)%*%t(X)%*%(diag(n)-P)%*%y
alpha0hat=gZZ%*%t(Z)%*%y
c(alpha0hat)
```

```
## [1] 24976.239  7846.109 22058.500 -4928.370
```

```
alphahat=alpha0hat-gZZ%*%t(Z)%*%X%*%betahat
c(alphahat, betahat)
```

```
## [1] 11012.182  3191.423 17403.814 -9583.055  1551.562
```

Now for the sum of squares we find

$$
\begin{aligned}
&\text{SSE} = \text{SSE}_{y.x} = \boldsymbol{y'y - \hat{\theta} U' y} = \\
&\boldsymbol{y'y} - \begin{pmatrix} \hat{\boldsymbol{\alpha}}' & \hat{\boldsymbol{\beta}}' \end{pmatrix} \begin{pmatrix} \boldsymbol{Z'y} \\ \boldsymbol{X'y} \end{pmatrix} = \\
&\boldsymbol{y'y} - \hat{\boldsymbol{\alpha}}' \boldsymbol{Z'y} - \hat{\boldsymbol{\beta}}' \boldsymbol{X'y} = \\
&\boldsymbol{y'y} - \left[\hat{\boldsymbol{\alpha}}_0' - \hat{\boldsymbol{\beta}}' \boldsymbol{X'Z} (\boldsymbol{Z'Z})^{-} \right] \boldsymbol{Z'y} - \hat{\boldsymbol{\beta}}' \boldsymbol{X'y} = \\
&\boldsymbol{y'y} - \hat{\boldsymbol{\alpha}}_0' \boldsymbol{Z'y} + \hat{\boldsymbol{\beta}}' \boldsymbol{X'Z} (\boldsymbol{Z'Z})^{-} \boldsymbol{Z'y} - \hat{\boldsymbol{\beta}}' \boldsymbol{X'y} = \\
&\boldsymbol{y'y} - \hat{\boldsymbol{\alpha}}_0' \boldsymbol{Z'y} - \hat{\boldsymbol{\beta}}' \boldsymbol{X'} \left[\boldsymbol{I} - \boldsymbol{Z} (\boldsymbol{Z'Z})^{-} \boldsymbol{Z'} \right] \boldsymbol{y} = \\
&\text{SSE}_y - \hat{\boldsymbol{\beta}}' \boldsymbol{X'} \left[\boldsymbol{I} - \boldsymbol{P} \right] \boldsymbol{y}
\end{aligned}
$$

where SSE_y is the error sum of squares for the ANOVA model without covariates.

Setting $e_{yy} = \text{SSE}_y = \boldsymbol{y'(I - P)y}$ we can also write

$$\text{SSE}_{y.x} = e_{yy} - e_{xy}' \boldsymbol{E}_{xx}^{-1} \boldsymbol{e}_{xy}$$

It can be shown that $\boldsymbol{E}_{xx}$ is positive definite, so that $e_{xy}' \boldsymbol{E}_{xx}^{-1} \boldsymbol{e}_{xy} > 0$, and so the use of covariates reduces the sum of squares, thereby leading to smaller variances.

7.1.3.2 Example (8.1.5)

Continuing the analysis of the shoesales data:

```
Exx=t(X)%*%(diag(n)-P)%*%X
exy=t(X)%*%(diag(n)-P)%*%y
eyy=t(y)%*%(diag(n)-P)%*%y
sse=eyy-t(exy)%*%solve(Exx)%*%exy
c(sse, eyy)
```

```
## [1] 2629827256 9938524862
```

7.1.4 Testing Hypotheses

As always we will now assume that $\boldsymbol{\epsilon} = N_n(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$. The hypothesis $H_0 : \alpha_1 = .. = \alpha_k = 0$ can be written as

$$H_0 : \begin{pmatrix} \boldsymbol{C}_1 & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{pmatrix} = \boldsymbol{0}$$

and we can use a general linear hypothesis test.

A test that is often of interest is $H_0 : \boldsymbol{\beta} = 0$, that is a test whether any of the covariates is useful. For a general linear hypothesis test of $H_0 : \boldsymbol{\beta} = 0$ we need $cov(\hat{\boldsymbol{\beta}})$, where $\hat{\boldsymbol{\beta}} = \left[\boldsymbol{X'} \left(\boldsymbol{I} - \boldsymbol{P} \right) \boldsymbol{X} \right]^{-1} \boldsymbol{X'(I - P)y}$. By (4.3.19) $\boldsymbol{I - P}$ is idempotent we have

$$
\begin{aligned}
&cov(\hat{\boldsymbol{\beta}}) = \\
&[\boldsymbol{X'(I-P)X}]^{-1}\boldsymbol{X'(I-P)}\sigma^2\boldsymbol{I(I-P)X}[\boldsymbol{X'(I-P)X}]^{-1} = \\
&\sigma^2[\boldsymbol{X'(I-P)X}]^{-1}\boldsymbol{X'(I-P)(I-P)X}[\boldsymbol{X'(I-P)X}]^{-1} = \\
&\sigma^2[\boldsymbol{X'(I-P)X}]^{-1}[\boldsymbol{X'(I-P)X}][\boldsymbol{X'(I-P)X}]^{-1} = \\
&\sigma^2[\boldsymbol{X'(I-P)X}]^{-1}
\end{aligned}
$$

So now

$$\text{SSH} = \hat{\boldsymbol{\beta}}\boldsymbol{X'(I-P)X} = \boldsymbol{e'_{xy}E^{-1}_{xx}e_{exy}}$$

ANOVA table for test of covariates in ANCOVA design

Source	df	SS	F
Covariates	k	$\text{SSH} = \boldsymbol{e'_{xy}E^{-1}_{xx}e_{exy}}$	$\frac{\text{SSH}/k}{\text{SSE}/(n-k-1)}$
Error	$n-k-1$	$\text{SSE} = \boldsymbol{e_{yy} - e'_{xy}E^{-1}_{xx}e_{xy}}$	
Total	$n-1$	$\text{SST} = \boldsymbol{e_{yy}}$	

7.1.4.1 Example (8.1.6)

Let's test for Year in shoesales:

```
k=1
ssh=t(exy)%*%solve(Exx)%*%exy
FTS = (ssh/1)/(sse/(n-k-1))
round(c(ssh, sse/c(1, n-k-1), FTS, 1-pf(FTS, 1, n-k-1)))
```

```
## [1] 7308697607 2629827256   39251153        186          0
```

or with R:

```
summary(aov(Sales~., data=shoesales))
```

```
##              Df    Sum Sq   Mean Sq F value Pr(>F)
## Year          1 7.309e+09 7.309e+09   180.6 <2e-16
## Type          2 8.383e+09 4.192e+09   103.6 <2e-16
## Residuals    65 2.630e+09 4.046e+07
```

In the above we have assumed that there was no interaction between the predictors. If we suspect that there is we should run

```
summary(aov(Sales~(.)^2, data=shoesales))
```

```
##              Df    Sum Sq   Mean Sq F value   Pr(>F)
## Year          1 7.309e+09 7.309e+09  227.39  < 2e-16
## Type          2 8.383e+09 4.192e+09  130.41  < 2e-16
## Year:Type     2 6.049e+08 3.024e+08    9.41 0.000266
## Residuals    63 2.025e+09 3.214e+07
```

and indeed there such an interaction.

7.1.5 Balanced One-Way Model with One Covariate

In the case where we have two predictors, one categorical and one continuous, and a balanced design the formulas simplify:

$$\hat{\boldsymbol{\alpha}} = \hat{\boldsymbol{\alpha}}_0 - (\boldsymbol{Z'Z})^{-}\boldsymbol{Z'X}\hat{\boldsymbol{\beta}} = \begin{pmatrix} 0 \\ \bar{y}_{1.} - \hat{\beta}\bar{x}_{1.} \\ \bar{y}_{2.} - \hat{\beta}\bar{x}_{2.} \\ \vdots \\ \bar{y}_{k.} - \hat{\beta}\bar{x}_{k.} \end{pmatrix}$$

$$\begin{aligned} \boldsymbol{E}_{xx} &= e_{xx} = \sum_{ij}(x_{ij} - \bar{x}_{i.})^2 \\ \boldsymbol{e}_{xy} &= e_{xy} = \sum_{ij}(x_{ij} - \bar{x}_{i.})(y_{ij} - \bar{y}_{i.}) \\ \boldsymbol{e}_{yy} &= e_{xy} = \sum_{ij}(y_{ij} - \bar{y}_{i.})^2 \\ \hat{\beta} &= \frac{e_{xy}}{e_{yy}} \\ \text{SSE}_{y.x} &= e_{yy} - e_{xy}^2/e_{xx} \end{aligned}$$

and $\text{SSE}_{y.x}$ has k(n-1)-1 degrees of freedom, where k is the number of groups and n is the number of observations per group.

ANOVA table for test of one covariate in One-Way ANOVA design

Source	df	SS	F
Covariate	1	$\text{SSH} = e_{xy}^2/e_{xx}$	$\frac{\text{SSH}/k}{\text{SSE}/(n-k-1)}$
Error	$k(n-1)-1$	$\text{SSE} = e_{yy} - e_{xy}^2/e_{xx}$	
Total	$k(n-1)-2$	$\text{SST} = e_{yy}$	

7.2 Generalized Least Squares

7.2.1 Correlated Response

In section 6.5 we discussed the case of correlated responses in the context of regression. We now return to this case in the general linear model.

In all the discussion so far (except section 6.5) we always assumed a model of the form $\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\epsilon}$ with $cov(\boldsymbol{\epsilon}) = \sigma^2\boldsymbol{I}$. We now consider the case where $cov(\boldsymbol{\epsilon}) = \sigma^2\boldsymbol{D}$, where $\boldsymbol{D}$ is a known positive-definite matrix.

Because D is positive-definite there exists a non-singular matrix $\boldsymbol{D}^{1/2}$ such that $\boldsymbol{D}^{1/2}\boldsymbol{D}^{1/2} = \boldsymbol{D}$, see (4.3.13). Now consider the transformed model

$$\boldsymbol{y}^* = \boldsymbol{D}^{-1/2}\boldsymbol{y} = \boldsymbol{D}^{-1/2}\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{D}^{-1/2}\boldsymbol{\epsilon}$$

$$cov(\boldsymbol{D}^{1/2}\boldsymbol{\epsilon}) = \boldsymbol{D}^{-1/2}\sigma^2\boldsymbol{D}\boldsymbol{D}^{-1/2} = \sigma^2\boldsymbol{I}$$

and so this transformed model has uncorrelated errors with equal variance. Therefore we can apply the methods previously discussed.

Notice

$$\begin{aligned}
&(\boldsymbol{y}^* - \boldsymbol{X}^*\hat{\boldsymbol{\beta}})'(\boldsymbol{y}^* - \boldsymbol{X}^*\hat{\boldsymbol{\beta}}) = \\
&(\boldsymbol{D}^{-1/2}\boldsymbol{y} - \boldsymbol{D}^{-1/2}\boldsymbol{X}\hat{\boldsymbol{\beta}})'(\boldsymbol{D}^{-1/2}\boldsymbol{y} - \boldsymbol{D}^{-1/2}\boldsymbol{X}\hat{\boldsymbol{\beta}}) = \\
&(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})'\boldsymbol{D}^{-1}(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})
\end{aligned}$$

and so the method of generalized least squares amounts to minimizing a quadratic form

$$(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})'\boldsymbol{D}^{-1}(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})$$

The normal equations of the transformed model are

$$\boldsymbol{X}^{*'}\boldsymbol{X}^*\hat{\boldsymbol{\beta}} = \boldsymbol{X}^{*'}\boldsymbol{y}^*$$

and so

$$\begin{aligned}
&(\boldsymbol{D}^{-1/2}\boldsymbol{X})'(\boldsymbol{D}^{-1/2}\boldsymbol{X})\hat{\boldsymbol{\beta}} = (\boldsymbol{D}^{-1/2}\boldsymbol{X})'(\boldsymbol{D}^{-1/2}\boldsymbol{y}) \\
&\boldsymbol{X}'\boldsymbol{D}^{-1}\boldsymbol{X}\hat{\boldsymbol{\beta}} = \boldsymbol{X}'\boldsymbol{D}^{-1}\boldsymbol{y} \\
&\hat{\boldsymbol{\beta}} = [\boldsymbol{X}'\boldsymbol{D}^{-1}\boldsymbol{X}]^-\boldsymbol{X}'\boldsymbol{D}^{-1}\boldsymbol{y}
\end{aligned}$$

7.2.1.1 Theorem (8.2.1) In a model of the form $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$ with $cov(\boldsymbol{\epsilon}) = \sigma^2\boldsymbol{D}$, the BLUE of a linear function $\boldsymbol{a}'\boldsymbol{\beta}$ is $\boldsymbol{a}'\hat{\boldsymbol{\beta}}$ where $\hat{\boldsymbol{\beta}} = [\boldsymbol{X}'\boldsymbol{D}^{-1}\boldsymbol{X}]^-\boldsymbol{X}'\boldsymbol{D}^{-1}\boldsymbol{y}$.

Furthermore

$$var(\boldsymbol{a}'\hat{\boldsymbol{\beta}}) = \sigma^2\boldsymbol{a}'[\boldsymbol{X}'\boldsymbol{D}^{-1}\boldsymbol{X}]^-\boldsymbol{a}$$

Also

$$\text{SSE} = \boldsymbol{y}'\boldsymbol{D}^{-1}\boldsymbol{y} - (\boldsymbol{y}'\boldsymbol{D}^{-1}\boldsymbol{X})(\boldsymbol{X}'\boldsymbol{D}^{-1}\boldsymbol{X})^-(\boldsymbol{X}'\boldsymbol{D}^{-1}\boldsymbol{y})$$

and $\text{SSE}/\sigma^2 \sim \chi^2(n-r)$ and is independent of the distribution of the BLUES.

proof follows from the discussion above.

7.2.1.2 Theorem (8.2.2) For testing a hypothesis of the form $H_0 : \boldsymbol{K\beta} = d$ where K is $m \times p$ of rank m the sum of squares is

$$\text{SSH} = \boldsymbol{(K\hat{\beta} - d)'[K(X'D^{-1}X)^{-}K']^{-1}(K\hat{\beta} - d)}$$

which has m degrees of freedom. The F-test is

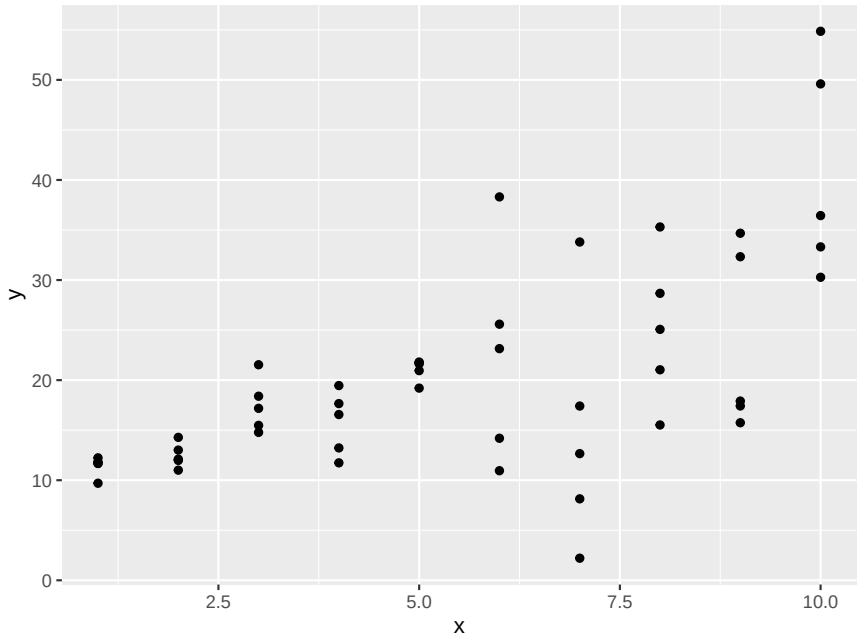
$$F = \frac{\text{SSH}/m}{\text{SSE}/(n-r)} \sim F(m, n-r)$$

proof omitted, but easy.

7.2.2 Weighted Least Squares

7.2.2.1 Example (8.2.3) We have following (artificial) data:

```
ggplot(data=df, aes(x, y)) +
  geom_point()
```



so here we have a positive linear relationship, but as x increases so does the variance of y|x. Assuming that the observations are still uncorrelated we have a model of the form

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

and $cov(\epsilon_i, \epsilon_j) = \sigma_i^2 \delta_{ij}$. In other words the matrix $\boldsymbol{D}$ is

$$\boldsymbol{D} = diag(\sigma_1^2, .., \sigma_n^2)$$

so then

$$\boldsymbol{D}^{-1} = diag(1/\sigma_1^2, .., 1/\sigma_n^2)$$

Therefore the least squares criterion becomes

$$\begin{aligned}
&(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})'\boldsymbol{D}^{-1}(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}) = \\
&(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})'diag(1/\sigma_1^2, .., 1/\sigma_n^2)(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}) = \\
&\sum_{i=1}^{n} \left(y_i - (\boldsymbol{X}\hat{\boldsymbol{\beta}})_{\boldsymbol{i}}\right)^2 / \sigma_i^2 = \\
&\sum_{i=1}^{n} w_i \left(y_i - (\boldsymbol{X}\hat{\boldsymbol{\beta}})_{\boldsymbol{i}}\right)^2
\end{aligned}$$

where $w_i = 1/\sigma_i^2$ are called the *weights*, hence the name weighted least squares.

7.2.2.2 Example (8.2.4) So, what are the weights in the example above? Ideally we would know these, but in practice we usually do not. Notice though that our data has multiple measurements at each x value, so we can estimate these:

```
w=1/tapply(y, x, var)
round(w, 4)
```

```
##      1      2      3      4      5      6      7      8      9     10
## 1.0172 0.6565 0.1395 0.0987 0.8321 0.0087 0.0070 0.0178 0.0121 0.0087
```

and so the weighted least squares estimators of the coefficients are

```
X=cbind(1, x)
Dinf=diag(rep(w, each=5))
tmp=solve(t(X)%*%Dinf%*%X)
betahat= c(tmp%*%t(X)%*%Dinf%*%cbind(y))
betahat
```

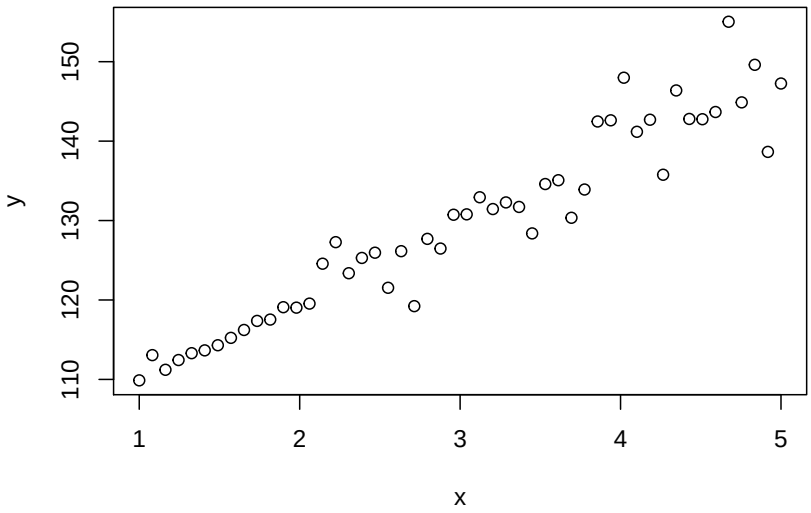
```
## [1] 8.690910 2.375936
```

or

```
fit=lm(y~x, data=df, weights=rep(w, each=5))
coef(fit)
```

```
## (Intercept)           x
##    8.690910    2.375936
```

```
ggplot(data=df, aes(x, y)) +
  geom_point() +
  geom_abline(xintercept=betahat[1],slope=betahat[2], size=2, col="blue")
```



If there is one predictor we can also do this directly:

$$0 = \frac{d}{d\beta_0}\sum_{i=1}^{n} w_i(y_i - \beta_0 - \beta_1 x_i)^2 =$$
$$(-2)\sum_{i=1}^{n} w_i(y_i - \beta_0 - \beta_1 x_i) =$$
$$(-2)\left(\sum_{i=1}^{n} w_i y_i - n\beta_0 \sum w_i - \sum_{i=1}^{n} \beta_1 w_i x_i\right) =$$
$$(-2n)\left(\overline{wy} - \beta_0 \bar{w} - \beta_1 \overline{wx}\right)$$
$$0 = \frac{d}{d\beta_1}\sum_{i=1}^{n} w_i(y_i - \beta_0 - \beta_1 x_i)^2 =$$
$$(-2)\sum_{i=1}^{n} w_i(y_i - \beta_0 - \beta_1 x_i)x_i =$$
$$(2n)\left(\overline{wxy} - \beta_0 \bar{w}\bar{x} - \beta_1 \overline{wx^2}\right)$$
$$\beta_0 \bar{w} + \beta_1 \overline{wx} = \overline{wy}$$
$$\beta_0 \overline{wx} + \beta_1 \overline{wx^2} = \overline{wxy}$$
$$\beta_0 = (\overline{wy} - \beta_1 \overline{wx})/\bar{w}$$
$$\beta_1 = \frac{\overline{wy} \times \overline{wx} - \bar{w} \times \overline{wxy}}{\overline{wx}^2 - \bar{w} \times \overline{wx^2}}$$

—

Notice that in the above did not actually know $\boldsymbol{D}$ but estimated it from the data. This is quite commonly done even so it violated the assumptions. This turns out to be generally ok.

```
x=df$x
y=df$y
w=rep(w, each=5)
bx=mean(x)
bw=mean(w)
bwx=mean(w*x)
bwy=mean(w*y)
bwxy=mean(w*x*y)
bwx2=mean(w*x^2)
betahat1=(bwx*bwy-bw*bwxy)/(bwx^2-bw*bwx2)
betahat0=(bwy-betahat1*bwx)/bw
round(c(betahat0, betahat1), 3)
```

7.2.2.3 Example (8.2.5)

```
## [1] 8.691 2.376
```

7.2.3 Sampling from a Small Population

Say we are selecting without replacement from a population that is so small that the probability of selecting the same object is not negligible. Let's investigate the effect of this.

Notice that so far there is no linear model, this is a general problem.

Say the population is $y_1, .., y_N$, then the population means is $\mu = \frac{1}{N}\sum y_i$ and the population variance is

$$\sigma^2 = \frac{1}{N}\sum(y_i - \mu)^2 = \frac{(N-1)S^2}{N}$$

where $S^2 = \frac{1}{N-1}\sum(y_i - \bar{y})^2$.

Now say we draw a sample of size $\{X_1, .., X_n\}$ from the population such that x_i is equally likely any of the y_j's. This is called a *simple random sample* (SRS). So

$$
\begin{aligned}
&E[X_i] = \sum_{i=1}^{N} y_i \frac{1}{N} = \bar{y} \\
&E[X_i^2] = \sum_{i=1}^{N} y_i^2 \frac{1}{N} \\
&E[X_i X_j] = E\{E[X_i X_j | X_j\} = \\
&E\{X_j E[X_i | X_j\} = \\
&E\{X_j \frac{\sum_{i \neq j} y_i}{N-1}\} = \\
&\frac{1}{N-1} E\{X_j [\sum_i y_i - X_j]\} = \\
&\frac{1}{N-1} E\{X_j [N\mu - X_j]\} = \\
&\frac{N\mu}{N-1} E\{X_j\} - \frac{1}{N-1} E\{X_j^2\} = \\
&\frac{N\mu}{N-1}\mu - \frac{1}{N-1}\frac{1}{N}\sum_{i=1}^{N} y_i^2 = \\
&\frac{N\mu^2}{N-1} - \frac{1}{N(N-1)}\sum_{i=1}^{N} y_i^2 \\
&cov(X_i, X_j) = E[X_i X_j] - E[X_i]E[X_j] = \\
&\frac{N\mu^2}{N-1} - \frac{1}{N(N-1)}\sum_{i=1}^{N} y_i^2 - \mu^2 = \\
&\frac{\mu^2}{N-1} - \frac{1}{N(N-1)}\sum_{i=1}^{N} y_i^2 = \\
&- \frac{1}{N(N-1)}\left(\sum_{i=1}^{N} y_i^2 - N\mu^2\right) = \\
&- S^2/N
\end{aligned}
$$

and also

$$var(X_i) = (N-1)S^2/N$$

So we can set up a model

$$X_i = \mu + \epsilon_i$$

where

$$cov(\boldsymbol{\epsilon}) = S^2(\boldsymbol{I_n} - \frac{1}{N}\boldsymbol{J}_{nn})$$

and so we have a generalized least squares model with $\boldsymbol{D} = \boldsymbol{I_n} - \frac{1}{N}\boldsymbol{J}_{nn}$ and $\sigma^2 = S^2$.

Note that $\boldsymbol{D}^{-1} = (\boldsymbol{I} - \frac{1}{N}\boldsymbol{J})^{-1} = \boldsymbol{I} + \frac{1}{N-n}\boldsymbol{J}$, so

$$\begin{aligned}
&\hat{\mu} = [\boldsymbol{X}'\boldsymbol{D}^{-1}\boldsymbol{X}]^{-}\boldsymbol{X}'\boldsymbol{D}^{-1}\boldsymbol{y} = \\
&[\boldsymbol{j}'(\boldsymbol{I} + \frac{1}{N-n}\boldsymbol{J})\boldsymbol{j}]^{-}\boldsymbol{j}'(\boldsymbol{I} + \frac{1}{N-n}\boldsymbol{J})\boldsymbol{y} = \\
&[\boldsymbol{j}'\boldsymbol{j} + \frac{1}{N-n}\boldsymbol{j}'\boldsymbol{J}\boldsymbol{j}]^{-}(\boldsymbol{j}'\boldsymbol{y} + \frac{1}{N-n}\boldsymbol{j}'\boldsymbol{J}\boldsymbol{y}) = \\
&[n + \frac{1}{N-n}n^2]^{-}(n\bar{y} + \frac{1}{N-n}n^2\bar{y}) = \\
&[n + \frac{1}{N-n}n^2]^{-}(n + \frac{1}{N-n}n^2)\bar{y} = \bar{y}
\end{aligned}$$

so the BLUE is $\bar{y}$!

Also

$$\begin{aligned}
&var(\hat{\mu}) = \sigma^2[\boldsymbol{X}'\boldsymbol{D}^{-1}\boldsymbol{X}]^{-}\boldsymbol{X}'\boldsymbol{D}^{-1} = \\
&\sigma^2[\boldsymbol{X}'\boldsymbol{D}^{-1}\boldsymbol{X}]^{-}\boldsymbol{X}'\boldsymbol{D}^{-1} = \frac{N-n}{Nn}S^2
\end{aligned}$$

$\frac{N-n}{Nn}$ is called the *finite population correction*.

7.2.4 Combining Experiments - Meta Analysis

Say we have k experiments, each taking measurements for the same quantity θ. Experiment k has the unbiased estimator $\mathrm{T_k}$. Is it possible to combine these experiments in to *super-experiment*? This is part of a branch of statistics called *meta analysis*.

We can write the model

$$T_k = \theta + \epsilon_i$$

where $cov(\boldsymbol{\epsilon}) = \sigma^2\boldsymbol{D}$. The BLUE of the combined experiment is therefore given by

$$\hat{\boldsymbol{\theta}} = [\boldsymbol{X}'\boldsymbol{D}^{-1}\boldsymbol{X}]^{-}\boldsymbol{X}'\boldsymbol{D}^{-1}\boldsymbol{T} = [\boldsymbol{j}'\boldsymbol{D}^{-1}\boldsymbol{j}]^{-}\boldsymbol{j}'\boldsymbol{D}^{-1}\boldsymbol{T}$$

with variance

$$var(\hat{\boldsymbol{\theta}}) = [\boldsymbol{j}'\boldsymbol{D}^{-1}\boldsymbol{j}]^{-}$$

- independent experiments

If it can be assumed that the experiments are independent we have $\boldsymbol{D} = diag(\sigma_1^2, .., \sigma_k^2)$, so

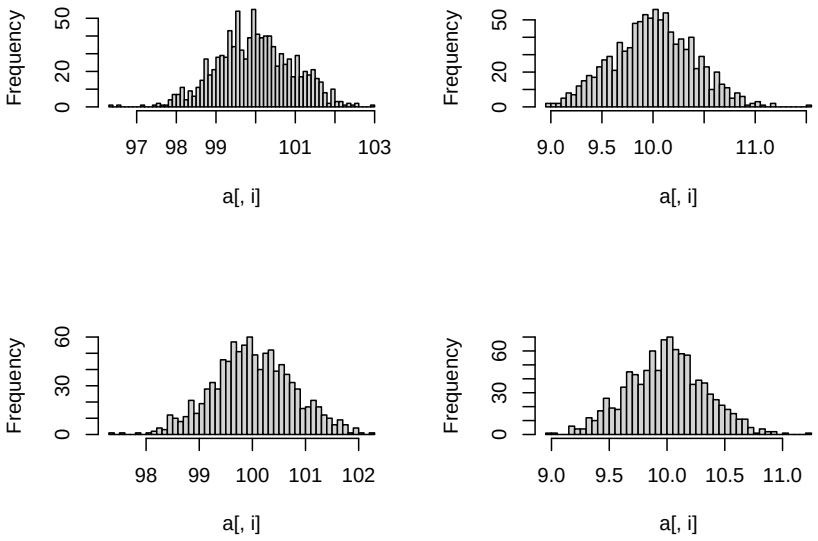
$$\hat{\boldsymbol{\theta}} = [\boldsymbol{j}' diag(1/\sigma_1^2, .., 1/\sigma_k^2)\boldsymbol{j}]^{-1}\boldsymbol{j}' diag(1/\sigma_1^2, .., 1/\sigma_k^2)\boldsymbol{T} =$$
$$[\sum_i 1/\sigma_i^2]^{-1} \sum_i T_i/\sigma_i^2 =$$
$$\frac{\sum_i T_i/\sigma_i^2}{\sum_i 1/\sigma_i^2} = \frac{\sum_i w_i T_i}{\sum_i w_i}$$

where the weights w are defined as before. The variance is

$$var(\hat{\boldsymbol{\theta}}) = \frac{1}{\sum_i 1/\sigma_i^2}$$

Say we have two such experiment, on with $\sigma^2 = 1$, then the variance of the combined experiment is

```
curve(1/(1+1/x), 0, 10)
```



and so we always gain something but we gain the most if the second experiment has a small variance.

In the special case where all experiments has the same variance σ^2 but different sample sizes we have $w_i = 1/\sigma_i^2 = 1/(\sigma^2/n_i) = n_i/\sigma^2$, and so

$$\hat{\boldsymbol{\theta}} = \frac{\sum_i n_i/\sigma^2 T_i}{\sum_i n_i/\sigma^2} = \frac{\sum_i n_i T_i}{\sum_i n_i}$$

and finally if all the sample sizes were the same the BLUE is simply the mean of the T_i's.

- two correlated experiments.

Say we have T_1 and T_2 with

$$\boldsymbol{D} = \begin{pmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho \\ \sigma_1\sigma_2\rho & \sigma_2^2 \end{pmatrix}$$

then

$$\boldsymbol{D}^{-1} = \frac{1}{\sigma_1^2\sigma_2^2(1-\rho^2)} \begin{pmatrix} \sigma_2^2 & -\sigma_1\sigma_2\rho \\ -\sigma_1\sigma_2\rho & \sigma_1^2 \end{pmatrix}$$

$$\boldsymbol{j}'\boldsymbol{D}^{-1}\boldsymbol{j} = \frac{1}{\sigma_1^2\sigma_2^2(1-\rho^2)} \begin{pmatrix} 1 & 1 \end{pmatrix} \begin{pmatrix} \sigma_2^2 - \sigma_1\sigma_2\rho \\ \sigma_1^2 - \sigma_1\sigma_2\rho \end{pmatrix} =$$

$$\frac{1}{\sigma_1^2\sigma_2^2(1-\rho^2)} \left[\sigma_2^2 - \sigma_1\sigma_2\rho + \sigma_1^2 - \sigma_1\sigma_2\rho\right] =$$

$$\frac{\sigma_1^2 + \sigma_2^2 - 2\sigma_1\sigma_2\rho}{\sigma_1\sigma_2(1-\rho^2)}$$

and so

$$\hat{\boldsymbol{\theta}} = [\boldsymbol{j}'\boldsymbol{D}^{-1}\boldsymbol{j}]^{-}\boldsymbol{j}'\boldsymbol{D}^{-1}\boldsymbol{T} =$$

$$[\frac{\sigma_1^2 + \sigma_2^2 - 2\sigma_1\sigma_2\rho}{\sigma_1\sigma_2(1-\rho^2)}]^{-1}\boldsymbol{j}' \frac{1}{\sigma_1^2\sigma_2^2(1-\rho^2)} \begin{pmatrix} \sigma_2^2 & -\sigma_1\sigma_2\rho \\ -\sigma_1\sigma_2\rho & \sigma_1^2 \end{pmatrix} \boldsymbol{T} =$$

$$\frac{\sigma_1\sigma_2(1-\rho^2)}{\sigma_1^2 + \sigma_2^2 - 2\sigma_1\sigma_2\rho} \frac{1}{\sigma_1^2\sigma_2^2(1-\rho^2)} \boldsymbol{j}' \begin{pmatrix} \sigma_2^2 T_1 - \sigma_1\sigma_2\rho T_2 \\ -\sigma_1\sigma_2\rho T_1 + \sigma_1^2 T_2 \end{pmatrix} =$$

$$\frac{(\sigma_2^2 - \sigma_1\sigma_2\rho)T_1 + (\sigma_1^2 - \sigma_1\sigma_2\rho)T_2}{\sigma_1^2 + \sigma_2^2 - 2\sigma_1\sigma_2\rho} =$$

$$\frac{a_2 T_1 + a_1 T_2}{a_1 + a_2}$$

where $a_i = \sigma_i^2 - \sigma_1\sigma_2\rho$

The variance of the combined estimator is

$$var(\hat{\boldsymbol{\beta}}) = [\boldsymbol{X}'\boldsymbol{D}^{-1}\boldsymbol{X}]^{-} = \frac{\sigma_1\sigma_2(1-\rho^2)}{\sigma_1^2 + \sigma_2^2 - 2\sigma_1\sigma_2\rho}$$

Let's investigate this a bit. To do so we fix $\sigma_1 = \sigma_2 = 1$, so we have

$$var(\hat{\boldsymbol{\beta}}) = \frac{1-\rho^2}{2-2\rho} = \frac{(1-\rho)(1+\rho)}{2(1-\rho)} = \frac{1+\rho}{2}$$

Not that $-1 \leq \rho \leq 1$, so $0 \leq \frac{1+\rho}{2} \leq 1 = var(T_i)$, so there is always an improvement but the best case is two highly negatively correlated experiments! If the two experiments are independent $\rho = 0$ and the variance is $1/2$.

7.3 Linear Mixed Models

In the last section we considered models with correlated responses of the form $\boldsymbol{y = X\beta + \epsilon}$, $E[\boldsymbol{\epsilon}] = 0$ and $cov(\boldsymbol{\epsilon}) = \sigma^2\boldsymbol{D}$, where $\boldsymbol{D}$ was assumed to be a *known* covariance matrix. In practice we often encounter cases where $\boldsymbol{D}$ is not known but has to be estimated. Unfortunately $\boldsymbol{D}$ has $\binom{n}{2}$ elements, and so there nowhere near enough equations to estimate it. However, in some cases one knows something about the *structure* of $\boldsymbol{D}$, which lowers the number of unknowns sufficiently to allow estimation.

Dependencies between observations can arise in many ways. For example, if observations are taken on meteorological phenomena (temperature, pressure etc) each day, measurements from one day to the next are likely correlated. In surveys of people those living in the same geographical area (maybe even in the same household) are likely correlated.

A model for experiments of this kind can often be written in the form

$$\boldsymbol{y = X\beta + Z_1a_1 + .. + Z_ma_m + \epsilon}$$

where $E[\boldsymbol{\epsilon}] = 0$ and $cov(\boldsymbol{\epsilon}) = \sigma^2\boldsymbol{I}$. Here $\boldsymbol{X}$ is a (possibly less than full-rank) $n \times p$ known matrix of fixed predictors, just as before. The $\boldsymbol{Z}_i$'s are known $n \times r_i$ full-rank matrices. The $\boldsymbol{a}_i$'s are vectors of *unknown* quantities, similar to $\boldsymbol{\epsilon}$. We assume $E[\boldsymbol{a}_i] = 0$, $cov(\boldsymbol{a}_i) = \sigma_i^2\boldsymbol{I}$ and $cov(\boldsymbol{a}_i, \boldsymbol{a}_j) = \boldsymbol{O}$ for $i \neq j$.

Compare this model to the random X model discussed in section 6.10. There it was the matrix $\boldsymbol{X}$ that was assumed to be random and the vector $\boldsymbol{\beta}$ was fixed. This model however is fairly close to the Bayesian model, where $\boldsymbol{\beta}$ is also assumed to be random.

Models of this kind are (for obvious reasons) called *mixed linear models*. If $\boldsymbol{X = j}$ is sometimes called a *random model*. The σ_i^2's are called *variance components*.

7.3.0.1 Theorem (8.3.1) Consider the model $\boldsymbol{y = X\beta + Z_1a_1 + .. + Z_ma_m + \epsilon}$, with elements as described above. Then $E[\boldsymbol{y}] = \boldsymbol{X\beta}$ and

$$cov(\boldsymbol{y}) = \boldsymbol{\Sigma} = \sum_{i=1}^{m} \sigma_i^2 \boldsymbol{Z_i'Z_i} + \sigma^2\boldsymbol{I}$$

proof

$$\begin{aligned}
&E[\boldsymbol{y}] = E[\boldsymbol{X\beta + Z_1a_1 + .. + Z_ma_m + \epsilon}] = \\
&E[\boldsymbol{X\beta}] + \boldsymbol{Z_1}E[\boldsymbol{a_1}] + .. + \boldsymbol{Z_m}E[\boldsymbol{a_m}] + E[\boldsymbol{\epsilon}] = \boldsymbol{X\beta} \\
&cov(\boldsymbol{y}) = cov(\boldsymbol{X\beta + Z_1a_2 + .. + Z_ma_m + \epsilon}) = \\
&cov(\boldsymbol{Z_1a_1}) + .. + cov(\boldsymbol{Z_ma_m}) + cov(\boldsymbol{\epsilon}) = \\
&\sigma_1^2\boldsymbol{Z_1'Z_1} + .. + \sigma_m^2\boldsymbol{Z_m'Z_m} + \sigma^2
\end{aligned}$$

because $cov(\boldsymbol{Z_ia_i}, \boldsymbol{Z_ja_j}) = \boldsymbol{Z_i}'cov(\boldsymbol{a_i, a_j})\boldsymbol{Z_j} = \boldsymbol{Z_i}'\boldsymbol{Z_i}$ if i=j and 0 otherwise.

7.3.1 Examples

7.3.1.1 Example (8.3.2) *Randomized Block design*

We carry out a study on the effectiveness of a new drug. We have three treatments: no drug, placebo and new drug. We randomly choose 4 hospitals (out of a much larger list of possible hospitals) where the study will take place. If we average over the patients for each factor level combination a model would be

$$y_{ij} = \mu + \tau_i + \alpha_j + \epsilon_{ij}$$

with i=1,2,3;j=1,2,3,4; $a_j \sim N(0, \sigma_j^2)$; $\epsilon_{ij} \sim N(0, \sigma^2)$ and $cov(a_j, \epsilon_{ij}) = 0$. where μ is the overall effect, τ_i is the effect of the three treatments, which are fixed, and α_i is the effect of the choice of hospital, which is random. Note that we did not include an interaction term, which is assumed not to exist.

So we have

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{j}_3 & \boldsymbol{I}_3 \\ \boldsymbol{j}_3 & \boldsymbol{I}_3 \\ \boldsymbol{j}_3 & \boldsymbol{I}_3 \\ \boldsymbol{j}_3 & \boldsymbol{I}_3 \end{pmatrix}$$

$$\boldsymbol{Z}_1 = \begin{pmatrix} \boldsymbol{j}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{j}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{j}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{j}_3 \end{pmatrix}$$

Let's use R to find $\boldsymbol{\sigma} = \sigma^2 \boldsymbol{I} + \sigma_1^2 \boldsymbol{Z_1' Z_1}$:

```
I=3;J=4
sig=2.5;sig1=1.9
Z=matrix(0, I*J, J)
for(j in 1:J) Z[(j-1)*I+1:I, j]=1
sig*diag(I*J)+sig1*Z%*%t(Z)
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
##  [1,]  4.4  1.9  1.9  0.0  0.0  0.0  0.0  0.0  0.0   0.0   0.0   0.0
##  [2,]  1.9  4.4  1.9  0.0  0.0  0.0  0.0  0.0  0.0   0.0   0.0   0.0
##  [3,]  1.9  1.9  4.4  0.0  0.0  0.0  0.0  0.0  0.0   0.0   0.0   0.0
##  [4,]  0.0  0.0  0.0  4.4  1.9  1.9  0.0  0.0  0.0   0.0   0.0   0.0
##  [5,]  0.0  0.0  0.0  1.9  4.4  1.9  0.0  0.0  0.0   0.0   0.0   0.0
##  [6,]  0.0  0.0  0.0  1.9  1.9  4.4  0.0  0.0  0.0   0.0   0.0   0.0
##  [7,]  0.0  0.0  0.0  0.0  0.0  0.0  4.4  1.9  1.9   0.0   0.0   0.0
##  [8,]  0.0  0.0  0.0  0.0  0.0  0.0  1.9  4.4  1.9   0.0   0.0   0.0
##  [9,]  0.0  0.0  0.0  0.0  0.0  0.0  1.9  1.9  4.4   0.0   0.0   0.0
## [10,]  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0   4.4   1.9   1.9
## [11,]  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0   1.9   4.4   1.9
## [12,]  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0   1.9   1.9   4.4
```

and so

$$\boldsymbol{\sigma} = \sigma^2\boldsymbol{I} + \sigma_1^2\boldsymbol{Z_1'Z_1} = \begin{pmatrix} \boldsymbol{\Sigma_1} & 0 & 0 & 0 \\ 0 & \boldsymbol{\Sigma_1} & 0 & 0 \\ 0 & 0 & \boldsymbol{\Sigma_1} & 0 \\ 0 & 0 & 0 & \boldsymbol{\Sigma_1} \end{pmatrix}$$

where

$$\boldsymbol{\Sigma}_1 = \begin{pmatrix} \sigma_1^2 + \sigma^2 & \sigma_1^2 & \sigma_1^2 \\ \sigma_1^2 & \sigma_2^2 + \sigma^2 & \sigma_1^2 \\ \sigma_1^2 & \sigma_1^2 & \sigma_1^2 + \sigma^2 \end{pmatrix}$$

7.3.1.2 Example (8.3.3) Subsampling

Five batches were produced using each of two processes. Two samples were obtained and measured from each of the batches. Constraining the process effects to sum to zero, the model is

$$y_{ijk} = \mu + \tau_i + \alpha_{ij} + \epsilon_{ijk}$$

with i=1,2;j=1,..,5;k=1,2; $\tau_1 + \tau_2 = 0$, $a_{ij} \sim N(0, \sigma_1^2)$; $\epsilon_{ijk} \sim N(0, \sigma^2)$ and $cov(a_{ij}, \epsilon_{ij}) = 0$.

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{j}_{10} & \boldsymbol{j}_{10} \\ \boldsymbol{j}_{10} & \boldsymbol{j}_{10} \end{pmatrix}$$

$$\boldsymbol{Z}_1 = \begin{pmatrix} \boldsymbol{j}_2 & \boldsymbol{0}_2 & ... & \boldsymbol{0}_2 \\ \boldsymbol{0}_2 & \boldsymbol{j}_2 & ... & \boldsymbol{0}_2 \\ \vdots & \vdots & & \vdots \\ \boldsymbol{0}_2 & \boldsymbol{0}_2 & ... & \boldsymbol{j}_2 \end{pmatrix}$$

and again we can find

$$\boldsymbol{\sigma} = \sigma^2\boldsymbol{I} + \sigma_1^2\boldsymbol{Z_1'Z_1} = \begin{pmatrix} \boldsymbol{\Sigma_1} & 0 & ... & 0 \\ 0 & \boldsymbol{\Sigma_1} & ... & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & ... & \boldsymbol{\Sigma_1} \end{pmatrix}$$

where

$$\boldsymbol{\Sigma}_1 = \begin{pmatrix} \sigma_1^2 + \sigma^2 & \sigma_1^2 \\ \sigma_1^2 & \sigma_1^2 + \sigma^2 \end{pmatrix}$$

7.3.1.3 Example (8.3.4) Split-Plot Studies

A 3×2 factorial experiment (with factors A and B, respectively) was carried out using six main units, each of which was subdivided into two subunits. The levels of A were each randomly assigned to two of the main units, and the levels of B were randomly assigned to subunits within main units.

An appropriate model is

$$y_{ijk} = \mu + \tau_i + \delta_j + \theta_{ij} + \alpha_{ik} + \epsilon_{ijk}$$

with i=1,2,3;j=1,2;k=1,2; $a_{ij} \sim N(0, \sigma_1^2)$; $\epsilon_{ijk} \sim N(0, \sigma^2)$ and $cov(a_{ij}, \epsilon_{ij}) = 0$.

Here are the matrices, created with R:

```
X=matrix(0, I*J, J)
for(j in 1:J) X[(j-1)*I+1:I, j]=1
tmp=rbind(cbind(diag(2),diag(2), 0*diag(2), 0*diag(2)),
      cbind(diag(2),diag(2), 0*diag(2), 0*diag(2)),
      cbind(diag(2),0*diag(2), diag(2), 0*diag(2)),
      cbind(diag(2),0*diag(2), diag(2), 0*diag(2)),
      cbind(diag(2),0*diag(2), 0*diag(2), diag(2)),
      cbind(diag(2),0*diag(2), 0*diag(2), diag(2)))
X=cbind(1, X, tmp)
X
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
##  [1,]    1    1    0    0    0    1    0    1    0     0     0     0     0
##  [2,]    1    1    0    0    0    0    1    0    1     0     0     0     0
##  [3,]    1    1    0    0    0    1    0    1    0     0     0     0     0
##  [4,]    1    0    1    0    0    0    1    0    1     0     0     0     0
##  [5,]    1    0    1    0    0    1    0    0    0     1     0     0     0
##  [6,]    1    0    1    0    0    0    1    0    0     0     1     0     0
##  [7,]    1    0    0    1    0    1    0    0    0     1     0     0     0
##  [8,]    1    0    0    1    0    0    1    0    0     0     1     0     0
##  [9,]    1    0    0    1    0    1    0    0    0     0     0     1     0
## [10,]    1    0    0    0    1    0    1    0    0     0     0     0     1
## [11,]    1    0    0    0    1    1    0    0    0     0     0     1     0
## [12,]    1    0    0    0    1    0    1    0    0     0     0     0     1
```

```
I=3;J=2;K=2
Z=matrix(0, I*J*K, I*J)
j=matrix(1, 2, 1)
o=matrix(0, 2, 1)
Z=rbind(cbind(j, o, o, o , o, o),
        cbind(o, j, o, o , o, o),
        cbind(o, o, j, o , o, o),
        cbind(o, o, o, j , o, o),
        cbind(o, o, o, o , j, o),
```

```
        cbind(o, o, o, o , o, j))
Z
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6]
##  [1,]    1    0    0    0    0    0
##  [2,]    1    0    0    0    0    0
##  [3,]    0    1    0    0    0    0
##  [4,]    0    1    0    0    0    0
##  [5,]    0    0    1    0    0    0
##  [6,]    0    0    1    0    0    0
##  [7,]    0    0    0    1    0    0
##  [8,]    0    0    0    1    0    0
##  [9,]    0    0    0    0    1    0
## [10,]    0    0    0    0    1    0
## [11,]    0    0    0    0    0    1
## [12,]    0    0    0    0    0    1
```

```
sig=2.5;sig1=1.9
sig*diag(I*J*K)+sig1*Z%*%t(Z)
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
##  [1,]  4.4  1.9  0.0  0.0  0.0  0.0  0.0  0.0  0.0   0.0   0.0   0.0
##  [2,]  1.9  4.4  0.0  0.0  0.0  0.0  0.0  0.0  0.0   0.0   0.0   0.0
##  [3,]  0.0  0.0  4.4  1.9  0.0  0.0  0.0  0.0  0.0   0.0   0.0   0.0
##  [4,]  0.0  0.0  1.9  4.4  0.0  0.0  0.0  0.0  0.0   0.0   0.0   0.0
##  [5,]  0.0  0.0  0.0  0.0  4.4  1.9  0.0  0.0  0.0   0.0   0.0   0.0
##  [6,]  0.0  0.0  0.0  0.0  1.9  4.4  0.0  0.0  0.0   0.0   0.0   0.0
##  [7,]  0.0  0.0  0.0  0.0  0.0  0.0  4.4  1.9  0.0   0.0   0.0   0.0
##  [8,]  0.0  0.0  0.0  0.0  0.0  0.0  1.9  4.4  0.0   0.0   0.0   0.0
##  [9,]  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  4.4   1.9   0.0   0.0
## [10,]  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.9   4.4   0.0   0.0
## [11,]  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0   0.0   4.4   1.9
## [12,]  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0   0.0   1.9   4.4
```

and so we find $\boldsymbol{\Sigma}$ to be same as in the previous example.

7.3.1.4 Example (8.3.5) One-Way Random Effects Model

Let's analyze a one-way model where the factor is random. So we have $\boldsymbol{y = \mu + \alpha_i + \epsilon_{ij}}$, $\alpha_i \sim N(0, \sigma_1^2)$, $\sigma_{ij} \sim N(0, \sigma^2)$, $cov(\alpha_i, \sigma_{ij}) = 0$.

For the purpose of illustration we will use I=3,J=4, then we have m=1, $\boldsymbol{X = j}$ (the vector for μ). Recall from the example above

$$\boldsymbol{Z}_1 = \begin{pmatrix} \boldsymbol{j}_4 & \boldsymbol{0}_4 & \boldsymbol{0}_4 \\ \boldsymbol{0}_4 & \boldsymbol{j}_4 & \boldsymbol{0}_4 \\ \boldsymbol{0}_4 & \boldsymbol{0}_4 & \boldsymbol{j}_4 \end{pmatrix}$$

and

$$\boldsymbol{\Sigma} = \sigma_1^2 \boldsymbol{Z_1 Z_1}' + \sigma^1 \boldsymbol{I} = \begin{pmatrix} \boldsymbol{\Sigma}_1 & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{\Sigma}_1 & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{\Sigma}_1 \end{pmatrix}$$

where

$$\boldsymbol{\Sigma}_1 = \begin{pmatrix} \sigma_1^2 + \sigma^2 & \sigma_1^2 & \sigma_1^2 & \sigma_1^2 \\ \sigma_1^2 & \sigma_1^2 + \sigma_1^2 & \sigma_1^2 & \sigma_1^2 \\ \sigma_1^2 & \sigma_1^2 & \sigma_1^2 + \sigma_1^2 & \sigma_1^2 \\ \sigma_1^2 & \sigma_1^2 & \sigma_1^2 & \sigma_1^2 + \sigma_1^2 \end{pmatrix}$$

7.3.2 Estimation of Variance Components

There are a number of different methods known for estimating the variance components in a mixed linear model. We will discuss a method known as *residual maximum likelihood (REML)*. This is also the estimator s^2 in the usual linear models.

We now add the normal assumption. So we have

$$\boldsymbol{y} \sim N_n(\boldsymbol{X\beta}, \boldsymbol{\Sigma})$$

where $\boldsymbol{\Sigma} = \sum_{i=1}^m \sigma_i^2 \boldsymbol{Z_i Z_i}' + \sigma^2 \boldsymbol{I}$. To simplify set $\sigma_0^2 = \sigma^2$ and $\boldsymbol{Z}_0 = \boldsymbol{I}$, so that $\boldsymbol{\Sigma} = \sum_{i=0}^m \sigma_i^2 \boldsymbol{Z_i Z_i}'$.

The idea of REML is to use maximum likelihood estimation on data $\boldsymbol{Ky}$, where $\boldsymbol{K}$ is chosen so that the distribution of $\boldsymbol{Ky}$ involves only the variance components, not $\boldsymbol{\beta}$. Therefore we need a matrix with $\boldsymbol{KX} = \boldsymbol{0}$. So we need $E[\boldsymbol{Ky}] = 0$.

7.3.2.1 Theorem (8.3.6) $\boldsymbol{K}$ must be of the form $\boldsymbol{K} = \boldsymbol{C}(\boldsymbol{I} - \boldsymbol{H}) = \boldsymbol{C}[\boldsymbol{I} - \boldsymbol{X}(\boldsymbol{X'X})^- \boldsymbol{X'}]$, where $\boldsymbol{C}$ is some full-rank transformation of the rows of $\boldsymbol{I} - \boldsymbol{H}$

proof The rows k_i of $\boldsymbol{K}$ must satisfy the equations $\boldsymbol{k_i' X} = \boldsymbol{0}$ or equivalently $\boldsymbol{X' k_i} = \boldsymbol{0}$. By (4.2.14) solutions to this system of equations are given by $\boldsymbol{k_i} = (\boldsymbol{I} - \boldsymbol{X}^- \boldsymbol{X})\boldsymbol{c}$ for all possible $p \times 1$ vectors $\boldsymbol{C}$. In other words, the solutions include all possible linear combinations of the columns of $\boldsymbol{I} - \boldsymbol{X}^- \boldsymbol{X}$.

Now $rank(\boldsymbol{X}^- \boldsymbol{X}) = r$. Also $\boldsymbol{I} - \boldsymbol{X}^- \boldsymbol{X}$ is idempotent. Because of this idempotency, $rank(\boldsymbol{I} - \boldsymbol{X}^- \boldsymbol{X}) = rank(\boldsymbol{I}) - rank(\boldsymbol{X}^- \boldsymbol{X}) = n - r$. Hence by the definition of rank n-r linearly independent vectors k_i that satisfy $\boldsymbol{X' k_i} = \boldsymbol{0}$ and thus the maximal number of rows in K is n-r.

Since $\boldsymbol{k_i} = (\boldsymbol{I} - \boldsymbol{X}^- \boldsymbol{X})\boldsymbol{c}$, $\boldsymbol{K} = \boldsymbol{C}(\boldsymbol{I} - \boldsymbol{X}^- \boldsymbol{X})$ for some full-rank $(n - r) \times n$ matrix $\boldsymbol{C}$ that specifies n-r linearly independent linear combinations of the rows of the symmetric matrix $\boldsymbol{I} - \boldsymbol{X}^- \boldsymbol{X}$.

7.3.2.2 Theorem (8.3.7)

$$\boldsymbol{Ky} \sim N_{n-r}(\mathbf{0}, \boldsymbol{K}(\sum_{\boldsymbol{i=0}}^{\boldsymbol{m}} \boldsymbol{\sigma_i^2 Z_i Z_i'})\boldsymbol{K'})$$

proof follows from (5.2.8)

So as desired the distribution of $\boldsymbol{Ky}$ depends only on the variance components. Therefore

7.3.2.3 Theorem (8.3.8)

A set of m+1 estimating equations is given by

$$tr[\boldsymbol{K'(K\Sigma K')^{-1}KZ_iZ_i'}] = \boldsymbol{y'K'(K\Sigma K')^{-1}KZ_iZ_i'K'(K\Sigma K')^{-1}Ky}$$

where $\Sigma = \sum_{i=0}^{m} \sigma_i^2 \boldsymbol{Z_i Z_i'}$

proof since $E[\boldsymbol{Ky}] = 0$ the log likelihood function is given by

$$\begin{aligned}
&\log L(\sigma_0^2, .., \sigma_m^2) = \\
&\frac{n-r}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{K\Sigma K'}| - \frac{1}{2}\boldsymbol{y'(K\Sigma K')^{-1}Ky} = \\
&\frac{n-r}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{K}\sum_{\boldsymbol{i=0}}^{\boldsymbol{m}}\boldsymbol{\sigma_i^2 Z_i Z_i' K'}| - \frac{1}{2}\boldsymbol{y'}(\boldsymbol{K}\sum_{\boldsymbol{i=0}}^{\boldsymbol{m}}\boldsymbol{\sigma_i^2 Z_i Z_i' K'})^{-1}\boldsymbol{Ky}
\end{aligned}$$

and so by (4.3.28) and (4.3.28)

$$\begin{aligned}
&\frac{\partial}{\partial\sigma_i^2}\log L(\sigma_0^2, .., \sigma_m^2) = \\
&-\frac{1}{2}tr\left(\boldsymbol{(K\Sigma K')}^{-1}\left[\frac{\partial}{\partial\sigma_i^2}\boldsymbol{(K\Sigma K')}^{-1}\right]\right) + \\
&\frac{1}{2}\boldsymbol{y'(K\Sigma K')^{-1}}\left[\frac{\partial}{\partial\sigma_i^2}\boldsymbol{(K\Sigma K')}^{-1}\right]\boldsymbol{(K\Sigma K')^{-1}Ky}
\end{aligned}$$

In most cases the these equations have to be solved numerically.

7.3.2.4 Example (8.3.9) One-Way Random Effects Model

Let's analyze a one-way model where the factor is random. SO we have $\boldsymbol{y = \mu + \alpha_i + \epsilon_{ij}}$, $\alpha_i \sim N(0, \sigma_1^2)$, $\sigma_{ij} \sim N(0, \sigma^2)$, $cov(\alpha_i, \sigma_{ij}) = 0$.

We saw before that

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_1 & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{\Sigma}_1 & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{\Sigma}_1 \end{pmatrix}$$

where

$$\boldsymbol{\Sigma}_1 = \begin{pmatrix} \sigma_1^2+\sigma^2 & \sigma_1^2 & \sigma_1^2 & \sigma_1^2 \\ \sigma_1^2 & \sigma_1^2+\sigma_1^2 & \sigma_1^2 & \sigma_1^2 \\ \sigma_1^2 & \sigma_1^2 & \sigma_1^2+\sigma_1^2 & \sigma_1^2 \\ \sigma_1^2 & \sigma_1^2 & \sigma_1^2 & \sigma_1^2+\sigma_1^2 \end{pmatrix}$$

Now $\boldsymbol{X'X} = 12$, so $\boldsymbol{X}(\boldsymbol{X'X})^{-}\boldsymbol{X'} = \frac{1}{12}\boldsymbol{XX'} = \frac{1}{12}\boldsymbol{J}$ and $\boldsymbol{I - X}(\boldsymbol{X'X})^{-}\boldsymbol{X'} = \boldsymbol{I} - \frac{1}{12}\boldsymbol{J}$.

As $\boldsymbol{C}$ we can choose $\boldsymbol{C} = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \end{pmatrix}$, then

$$\boldsymbol{K = C(I - H)} = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \end{pmatrix} (\boldsymbol{I} - \frac{1}{12}\boldsymbol{J}) = \boldsymbol{I}$$
$$\boldsymbol{K\Sigma K'} = \boldsymbol{I\Sigma_1 I'} = \boldsymbol{\Sigma}$$

Now we need $\boldsymbol{\Sigma}^{-1}$, which is quite some calculation. In the end we find the equations

$$9\sigma_0^2 = \boldsymbol{y'(I - \frac{1}{4}Z_1 Z_1')y}$$
$$2(4\sigma_1^2 + \sigma_0^2) = \boldsymbol{y'(\frac{1}{4}Z_1 Z_1' - \frac{1}{12}J)}y$$

which has the solution

$$\widehat{\sigma_0^2} = \boldsymbol{y'(I - \frac{1}{4}Z_1 Z_1')y}/9$$
$$\widehat{\sigma_1^2} = \boldsymbol{y'(\frac{1}{4}Z_1 Z_1' - \frac{1}{12}J)}y/4 - \widehat{\sigma_0^2}/8$$

It is possible that $\widehat{\sigma_1^2} < 0$. In this case one usually sets $\widehat{\sigma_1^2} = 0$.

Rather than using this approach directly there are a number of iterative method known that find the solutions.

7.3.2.5 Example (8.3.10) Let's return to the hearing aid data. Here List is a fixed effect (those are all the lists of interest) but Subject is a random effect, a sample from all possible people with good hearing. So this is a randomized block design with Subject the blocking variable.

```
library(lme4)
fit=lmer(Score~List+(1|Subject), data=hearingaid)
summary(fit)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Score ~ List + (1 | Subject)
##    Data: hearingaid
##
## REML criterion at convergence: 646.1
##
## Scaled residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -2.1393 -0.6430 -0.0658  0.6349  2.7597
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Subject  (Intercept) 25.90    5.089
##  Residual             36.92    6.076
## Number of obs: 96, groups:  Subject, 24
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  34.7917     1.8402  18.907
## List         -2.5917     0.5546  -4.673
##
## Correlation of Fixed Effects:
##      (Intr)
## List -0.754
```

```
library(car)
Anova(fit)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: Score
##       Chisq Df Pr(>Chisq)
## List 21.834  1  2.973e-06
```

so we see that the REML method is used, and the routine finds $\sigma_1^2 = 25.9$.

Notice that the estimates could not have been found using *aov*:

```
summary(aov(Score~List+Subject, data=hearingaid))
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## List         1    806   806.0  12.837 0.000543
## Subject      1     13    13.5   0.215 0.644213
## Residuals   93   5839    62.8
```

7.3.3 Inference for β

Estimates of the variance components can be plugged into $\mathbf{\Sigma}$ to obtain

$$\hat{\mathbf{\Sigma}} = \sum \hat{\sigma}_i^2 \boldsymbol{Z_1 Z_i'}$$

Using this we find an estimator of $\boldsymbol{\beta}$ to be

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X'\hat{\Sigma}^{-1}X})^{-}\boldsymbol{X'\hat{\Sigma}^{-1}y}$$

This is called the *generalized least-squares (EGLS)* estimator. It is a non-linear estimator because $\hat{\Sigma}$ is a non-linear function of $\boldsymbol{y}$. $\hat{\boldsymbol{\beta}}$ is not MVUE but it can be shown to be asymptotically MVUE.

Similarly, a sensible estimator of the covariance matrix of $\hat{\boldsymbol{\beta}}$ is given by

$$cov(\hat{\boldsymbol{\beta}}) = (\boldsymbol{X'}\hat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{X})^{-}\boldsymbol{X'}\hat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{X}(\boldsymbol{X'}\hat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{X})^{-}$$

which, if $\boldsymbol{X}$ is full-rank, simplifies to

$$cov(\hat{\boldsymbol{\beta}}) = (\boldsymbol{X'}\hat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{X})^{-1}$$

7.3.4 Inference for Estimable Functions of β

Exact methods for inference for estimable functions of $\boldsymbol{\beta}$ in linear mixed models do generally not exits. However, a number of approximate methods are known.

7.3.4.1 Example (8.3.11)

7.3.4.2 Example (8.3.12) Let's return to the hearing aid data. Notice that the output of the lmer routine does not yield p values:

```
library(lme4)
fit=lmer(Score~List+(1|Subject), data=hearingaid)
summary(fit)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Score ~ List + (1 | Subject)
##    Data: hearingaid
##
## REML criterion at convergence: 646.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.1393 -0.6430 -0.0658  0.6349  2.7597
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Subject  (Intercept) 25.90    5.089
##  Residual             36.92    6.076
## Number of obs: 96, groups:  Subject, 24
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  34.7917     1.8402  18.907
```

```
## List         -2.5917     0.5546  -4.673
##
## Correlation of Fixed Effects:
##      (Intr)
## List -0.754
```

we can get those as follows:

```
library(car)
Anova(fit)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: Score
##       Chisq Df Pr(>Chisq)
## List 21.834  1  2.973e-06
```

7.4 Nonlinear Regression

We have previously discussed polynomial regression, which is one way beyond linear regression. In this section we discuss two others.

7.4.1 Transformations

It is often possible to turn a non-linear model into a linear one via a *transformation*. Say for example that we want to fit an exponential model of the form $y = ae^{bx}$, then

$$\log y = \log ae^{bx} = \log a + bx$$

so this is equivalent to a linear model in $\log y$.

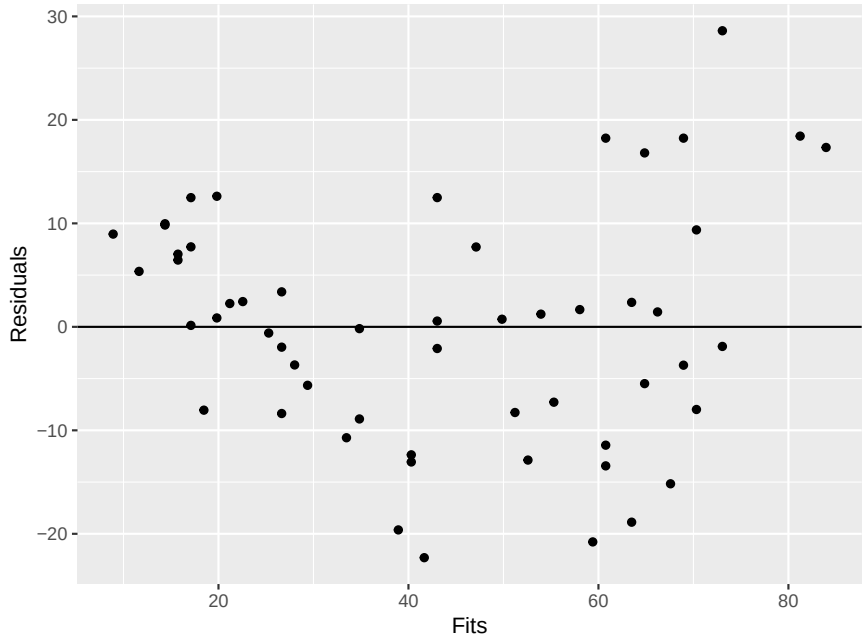
7.4.1.1 Example (8.4.1) In Westchester County, north of New York City, Consolidated Edison bills residential customers for electricity on a monthly basis. The company wants to predict residential usage, in order to plan purchases of fuel and budget revenue flow. The data includes information on usage (in kilowatt-hours per day) and average monthly temperature for 55 consecutive months for an all-electric home. Data on consumption of electricity and the temperature in Westchester County, NY.

```
attach(elusage)
head(elusage)
```

```
##   Month Year  Usage Temperature
## 1     8 1989 24.828          73
## 2     9 1989 24.688          67
## 3    10 1989 19.310          57
## 4    11 1989 59.706          43
## 5    12 1989 99.667          26
```

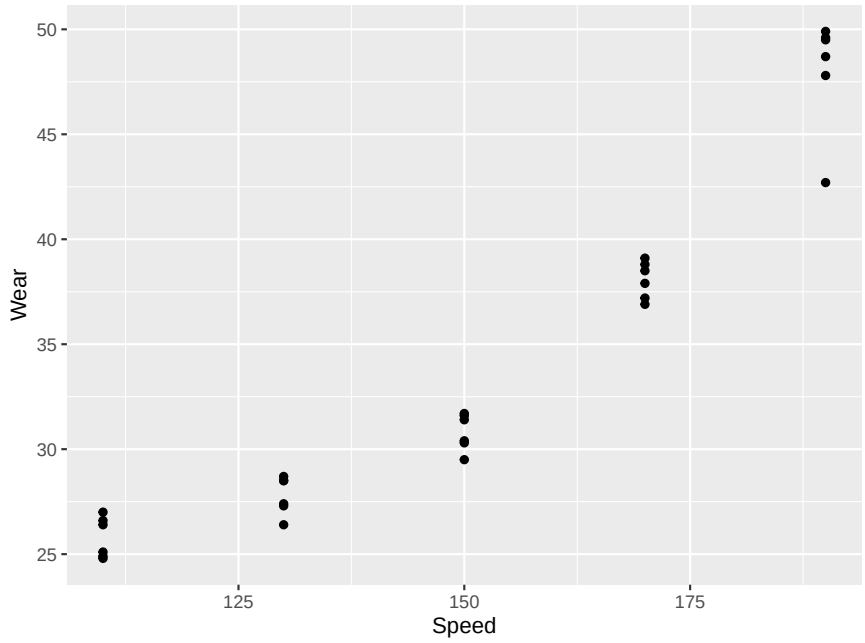
```
## 6     1 1990 49.333          41
```

```
fitlin=lm(Usage~Temperature, data=elusage)
df <- data.frame(Residuals=resid(fitlin),
            Fits = fitted(fitlin))
ggplot(data=df, aes(Fits, Residuals)) +
            geom_point() +
            geom_hline(yintercept = 0)
```



so the residual vs fits plot shows a clear pattern, so a linear model is not good. Now

```
fitlogy=lm(log(Usage)~Temperature, data=elusage)
df <- data.frame(Residuals=resid(fitlogy),
            Fits = fitted(fitlogy))
ggplot(data=df, aes(Fits, Residuals)) +
            geom_point() +
            geom_hline(yintercept = 0)
```



is much better

```
exp(coef(fitlogy)[1])
```

```
## (Intercept)
##     207.513
```

```
coef(fitlogy)[2]
```

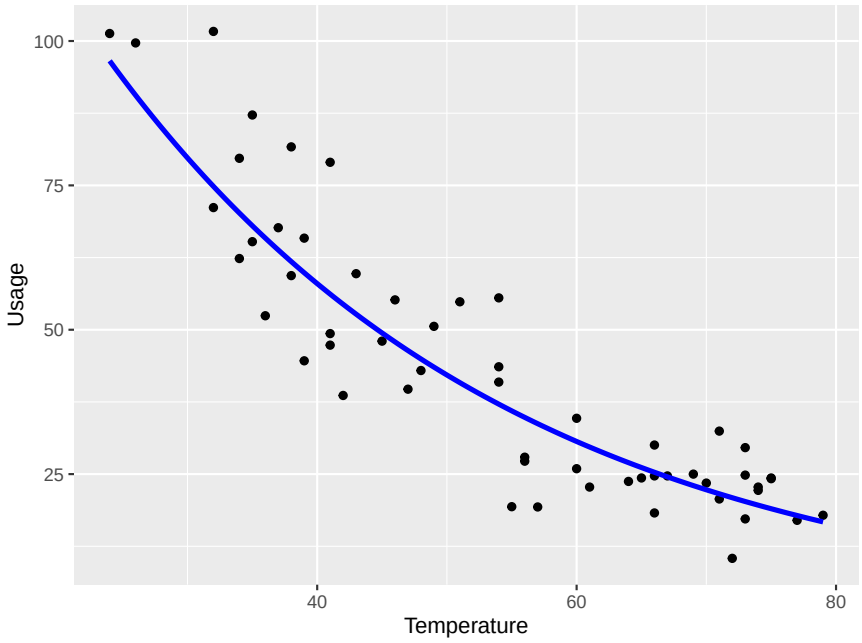
```
## Temperature
## -0.03187381
```

shows that the model is

$$y = 207.5e^{-0.0319x}$$

which looks like this:

```
x=seq(min(elusage$Temperature), max(elusage$Temperature), length=100)
y=exp(coef(fitlogy)[1])*exp(coef(fitlogy)[2]*x)
df=data.frame(x=x,y=y)
ggplot(data=elusage, aes(Temperature, Usage)) +
  geom_point() +
  geom_line(aes(x, y), data=df, col="blue", size=1.2)
```



There are issues when finding confidence intervals and doing hypothesis tests, but we won't discuss these here.

7.4.2 Nonlinear Parametric Models

Sometimes the model we wish to fit is known, up to parameters. Generally that is the case if there is a scientific theory that predicts the shape of the relationship. For example, radioactive decay is known to be exponential: $y = \alpha e^{-\beta t}$

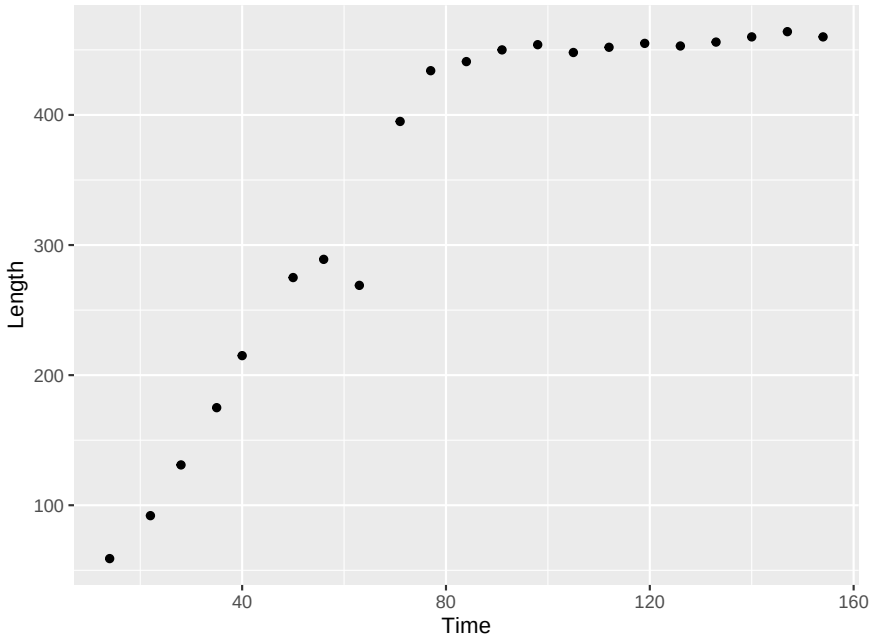
7.4.2.1 Example (8.4.2) Growth of Lobsters

Data from an experiment to raise Florida lobster in a controlled environment. The data shows the overall length and the age of a certain species of lobster.

```
kable.nice(lobster[1:10, ])
```

	Time	Length
1	14	59
2	22	92
3	28	131
4	35	175
5	40	215
6	50	275
7	56	289
8	63	269
9	71	395
10	77	434

```
ggplot(data=lobster, aes(Time, Length)) +
  geom_point()
```



Now biology suggests that the relationship should be of the form

$$y = \frac{\beta_2}{1 + (\beta_2 - \beta_0)/\beta_0 \exp(\beta_1 t)} + \epsilon$$

where

- β_0 is the expected value of y at time t=0
- β_1 is a measure of the growth rate

- β_2 is the expected limit of y as $t \to \infty$

This is often called the **logistic** or **autocatalytic** model

Previously we have always used the method of least squares to estimate the parameters in our models, that is we minimized the "figure of merit"

$$\text{RSS} = \sum (y_i - \beta_0 - \beta_1 x_i)^2$$

the natural extension of this is to use

$$\text{RSS} = \sum (y_i - f(x_i; \boldsymbol{\beta}))^2$$

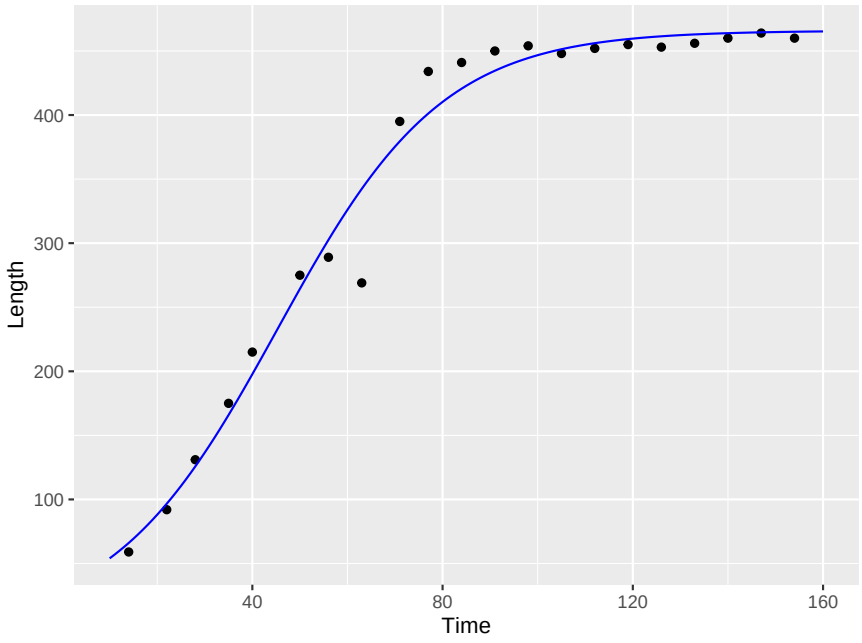
now for a linear model minimizing this expression could be done with *lm*. This however is still a minimization problem, and we can do it with

```
fit <- nls(Length ~ beta[3]/(1 + ((beta[3] -
          beta[1])/beta[1]) *  exp(beta[2] * Time)),
          start = list(beta = c(10, -0.1, 500)),
          data = lobster)
summary(fit)
```

```
##
## Formula: Length ~ beta[3]/(1 + ((beta[3] - beta[1])/beta[1]) * exp(beta[2] *
##     Time))
##
## Parameters:
##         Estimate Std. Error t value Pr(>|t|)
## beta1  32.008757   6.755720   4.738 0.000164
## beta2  -0.057557   0.004957 -11.612 8.55e-10
## beta3 465.884778   8.340739  55.857  < 2e-16
##
## Residual standard error: 21.63 on 18 degrees of freedom
##
## Number of iterations to convergence: 7
## Achieved convergence tolerance: 7.722e-06
```

```
x <- seq(10, 160, 1)
df <- data.frame(x=x,
                 y = predict(fit,
                  newdata = data.frame(Time = x)))
```

```
ggplot(data=lobster, aes(Time, Length)) +
 geom_point() +
 geom_line(data=df, aes(x, y), color="blue")
```



7.4.2.2 Example (8.4.3) Prime Number Theorem

That there were infinitely many prime numbers was first proven by the Greek mathematician Euclid at around 300BC. A serious study of how fast they grow was first begun by Adrienne-Marie Legendre. He studied the function N(k), which gives the number of primes less or equal to k. We can do the same. The primes up to 1,000,000 are available at

```
primes <- scan("C://Users//Wolfgang//dropbox//teaching//Computing-with-R//primes.txt")
```

```
## Error in file(file, "r"): cannot open the connection
```

```
primes <- as.integer(primes)
```

```
## Error in eval(expr, envir, enclos): object 'primes' not found
```

```
kable.nice(matrix(primes[1:100], ncol=10, byrow = TRUE))
```

```
## Error in matrix(primes[1:100], ncol = 10, byrow = TRUE): object 'primes' not found
```

A detailed study of these primes led Legendre in 1798 to propose the function

$$N(k) = k/(\log k - \alpha)$$

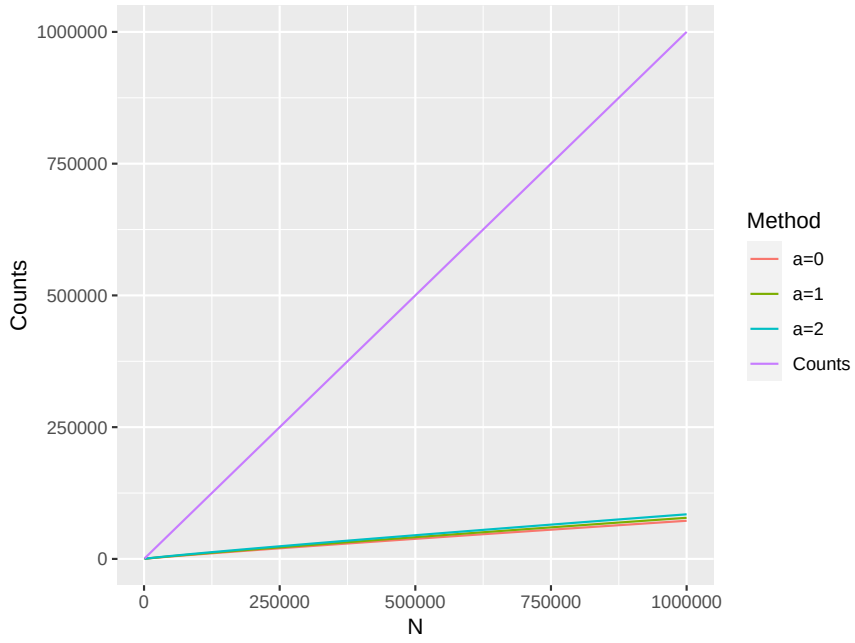
Here is what that looks like for several values of α:

```
N <- function(k, alpha) {
  k/(log(k)-alpha)
}
k <- seq(1000, 1e6, length=250)
exact.counts <- k
```

```
for(i in 1:250)
  exact.counts[i] <- sum(primes<k[i])
```

```
## Error in eval(expr, envir, enclos): object 'primes' not found
```

```
df <- data.frame(N=c(k, k, k, k),
      Counts=c(exact.counts, N(k, 0), N(k, 1), N(k, 2)),
      Method=rep(c("Counts", "a=0", "a=1", "a=2"),
                 each=250))
ggplot(df, aes(N, Counts, color=Method)) +
  geom_line()
```



and so it seems a value of $\alpha = 1$ is good.

Legendre however was not satisfied with that, he wanted to find the optimal answer. So he found the least squares solution!

```
fit <- nls(exact.counts ~ k/(log(k) - alpha),
           start = list(alpha = 0))
coef(fit)
```

```
##    alpha
## 10.47122
```

and so he claimed that

$$N(k) = k/(\log k - 1.08)$$

Around the same time German mathematician Carl Friedrich Gauss also looked at this problem, and he made a different conjecture. He said

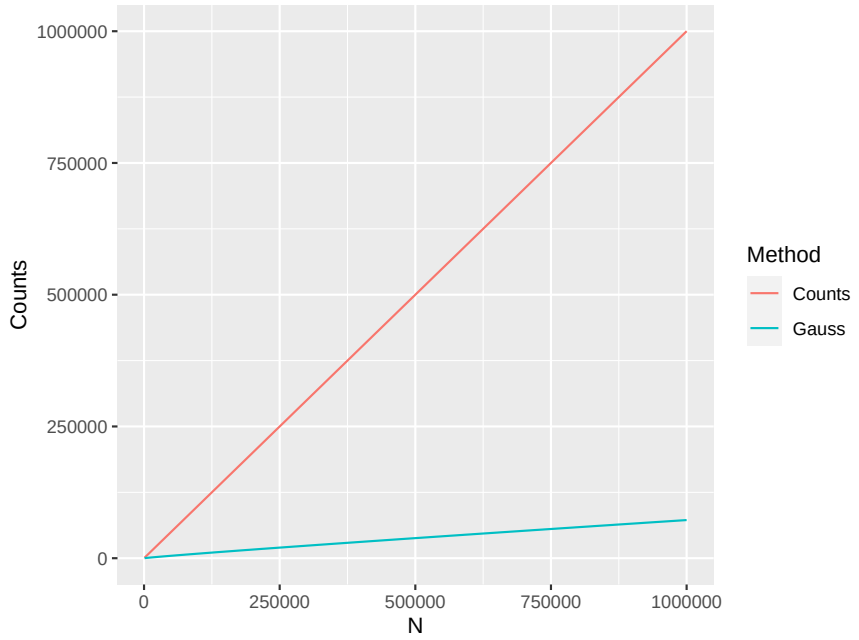
$$N(k) = k/\log k$$

That was a rather strange guess, because it looks like this:

```
N <- function(k, alpha) {
  k/(log(k)-alpha)
}
k <- seq(1000, 1e6, length=250)
exact.counts <- k
for(i in 1:250)
  exact.counts[i] <- sum(primes<k[i])
```

```
## Error in eval(expr, envir, enclos): object 'primes' not found
```

```
df <- data.frame(N=c(k, k),
      Counts=c(exact.counts, N(k, 0)),
      Method=rep(c("Counts", "Gauss"),   each=250))
ggplot(df, aes(N, Counts, color=Method)) +
  geom_line()
```



and it surely looks like the two curves are growing further apart. However, almost 100 years later in 1896 the French mathematicians Jacques-Salomon Hadamard and Charles de la Valée Poussin independently showed that Gauss was right!

From our modern point of view we might say Legendre was guilty of **over-fitting**!

7.5 Logistic and Poisson Regression, Generalized Linear Models

7.5.1 Logistic Regression

7.5.1.1 Example (8.5.1) We begin with a very famous data set from the Challenger shuttle disaster. On Jan 28, 1986, at 11.38 am EST, the space shuttle challenger was launched from Cape Canaveral, Florida. The mission ended 73 seconds later when the Challenger exploded. All 7 crew members were killed.

What happened?

Hot propellant gases flew past the aft joint of the right solid rocket booster, burning through two rubber O-rings. An investigation ensued into the reliability of the shuttle's propulsion system. The explosion was eventually traced to the failure of one of the three field joints on one of the two solid booster rockets. Each of these six field joints includes two O-rings, designated as primary and secondary, which fail when phenomena called erosion and blowby both occur.

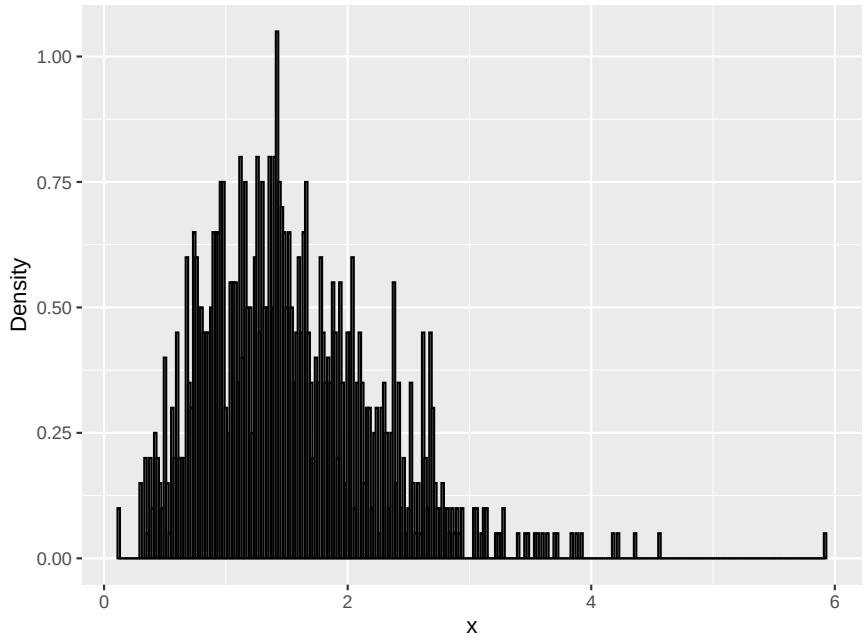
The night before the launch a decision had to be made regarding launch safety. The discussion among engineers and managers leading to this decision included concern that the probability of failure of the O-rings depended on the temperature t at launch, which was forecast to be 31 degrees F. There are strong engineering reasons based on the composition of O-rings to support the judgment that failure probability may rise monotonically as temperature drops.

The discussion centered on the following data from the previous 23 shuttle launches:

```
kable.nice(head(shuttle))
```

	Temp	NumFail	Failure
1	66	0	0
2	70	1	1
3	69	0	0
4	68	0	0
5	67	0	0
6	72	0	0

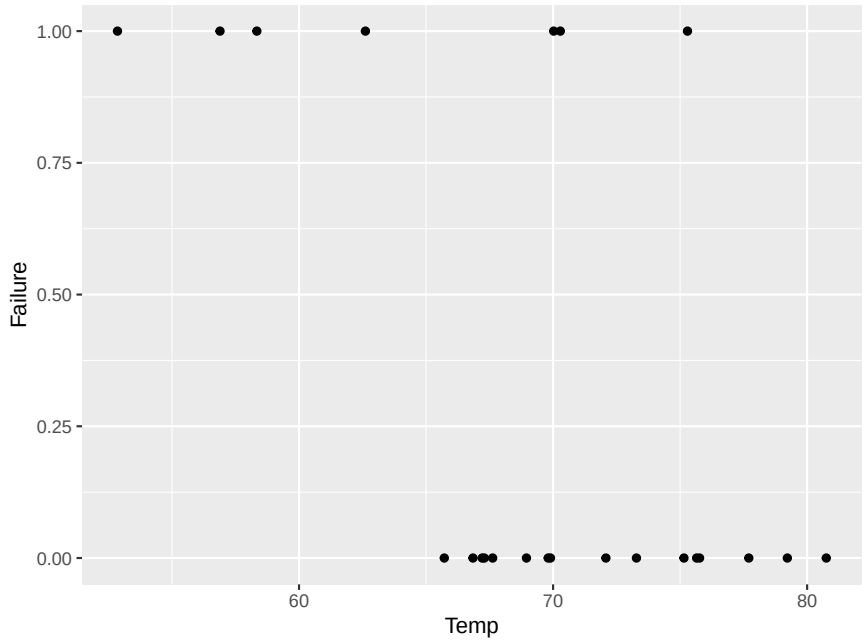
```
ggplot(data=shuttle, aes(Temp, NumFail)) +
  geom_point()
```



there seems to be a tendency for failures at lower temperatures.

The variable Failure is an indicator of failure or not:

```
plt <- ggplot(data=shuttle, aes(Temp, Failure)) +
  geom_jitter(height = 0)
plt
```



So again we have a model of the form $y_i = \beta_0 + \beta_1 x_i + \epsilon_i$, but now $y_1, .., y_n$ are Bernoulli rv's. Therefore

$$E[y_i] = P(y_i = 1) = p_i = 1 - P(y_i = 0) = 1 - E[y_i]$$
$$var(y_i) = p_i(1 - p_i)$$

Note that therefore

$$var(y_i) = p_i(1 - p_i) = E[y_i](1 - E[y_i]) = (\beta_0 + \beta_1 x_i)(1 - \beta_0 - \beta_1 x_i)$$

and we also have unequal variances. This means that the usual least squares estimators are no longer optimal.

Another issue is that as always $0 \leq p_i \leq 1$ but using least squares $\hat{p}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ need not be. What is needed is a model that always yields values in [0,1], that is a model of the form

$$l(p_i) = \beta_0 + \beta_1 x_i + \epsilon_i$$

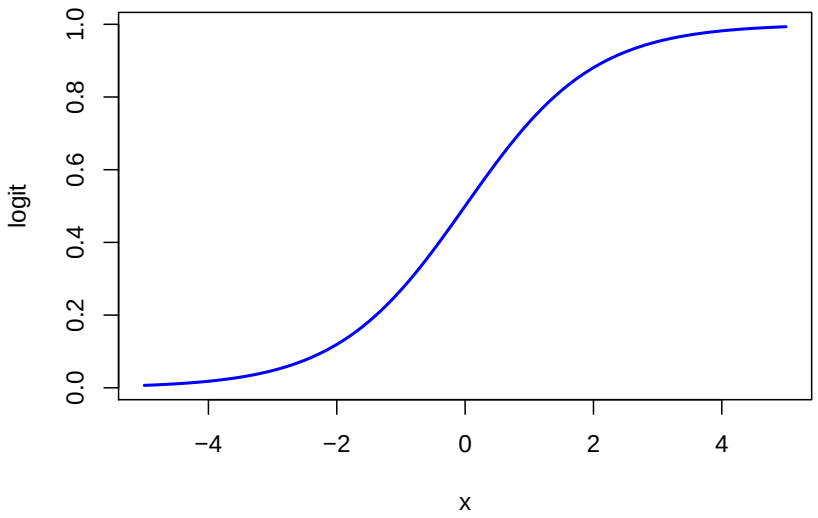
where l is a *link* function. One popular choice is the *logit transform* $l(x) = \log(\frac{x}{1-x})$, which leads to *logistic regression*.

Note

$$\log(\frac{p_i}{1 - p_i}) = \beta_0 + \beta_1 x_i$$
$$\frac{p_i}{1 - p_i} = \exp\{\beta_0 + \beta_1 x_i\}$$
$$p_i = \frac{\exp\{\beta_0 + \beta_1 x_i\}}{1 + \exp\{\beta_0 + \beta_1 x_i\}} = \frac{1}{1 + \exp\{-\beta_0 - \beta_1 x_i\}}$$

Here is what this looks like:

```
curve(exp(x)/(1+exp(x)),-5, 5,ylab="logit", col="blue", lwd=2)
```



Because the observed data is discrete and the predicted values are continuous $y_i - \hat{y}_i$ no longer is sensible, and so least square can not be used for estimation. Instead one generally uses maximum likelihood:

$$L(\beta_0, \beta_1) = \prod_{i=1}^{n} f(x_i; \beta_0, \beta_1) = \prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{1-y_i}$$

$$\log L(\beta_0, \beta_1) = \sum_i [y_i \log p_i + (1-y_i)\log(1-p_i)] =$$

$$\sum_i \left[y_i \log \frac{\exp\{\beta_0 + \beta_1 x_i\}}{1+\exp\{\beta_0+\beta_1 x_i\}} + (1-y_i)\log(1 - \frac{\exp\{\beta_0+\beta_1 x_i\}}{1+\exp\{\beta_0+\beta_1 x_i\}}) \right] =$$

$$\sum_i [y_i(\beta_0+\beta_1 x_i) - y_i \log(1+\exp\{\beta_0+\beta_1 x_i\} -$$

$$(1-y_i)\log(1+\exp\{\beta_0+\beta_1 x_i\})] =$$

$$\sum_i y_i(\beta_0+\beta_1 x_i) - \sum_i \log(1+\exp\{\beta_0+\beta_1 x_i\}$$

$$d\log L(\beta_0,\beta_1)/d\beta_0 = \sum_i y_i - \sum_i \frac{\exp\{\beta_0+\beta_1 x_i\}}{1+\exp\{\beta_0+\beta_1 x_i\}} = 0$$

$$d\log L(\beta_0,\beta_1)/d\beta_1 = \sum_i y_i x_i - \sum_i \frac{\exp\{\beta_0+\beta_1 x_i\}\, x_i}{1+\exp\{\beta_0+\beta_1 x_i\}} = 0$$

and these equations have to be solved numerically.

Clearly if there are more than one predictor the model becomes

$$\log(\frac{p_i}{1-p_i}) = \boldsymbol{X'\beta}$$

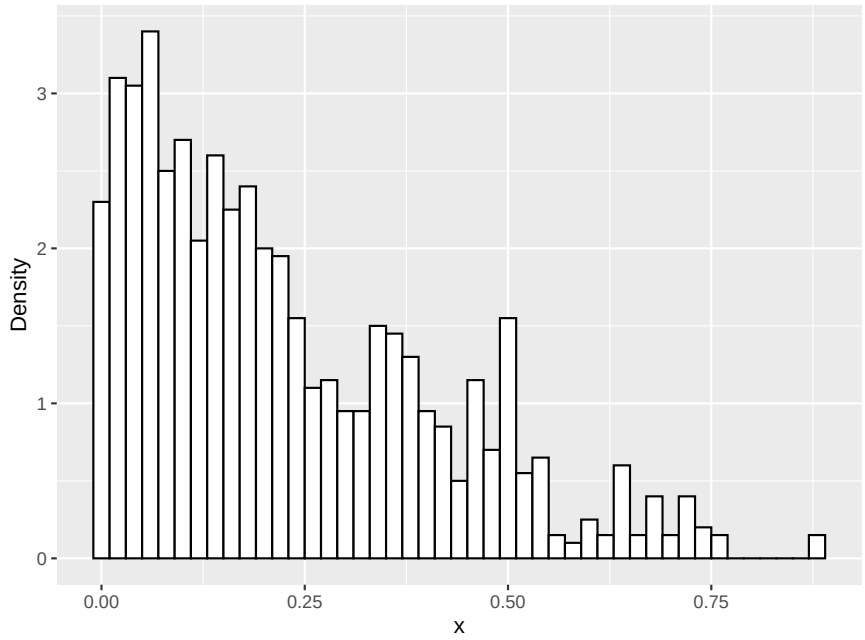
Example (8.5.2)

for the Challenger disaster data we find

```
ll=function(beta)
  -(sum(shuttle$Failure*(beta[1]+beta[2]*shuttle$Temp))-
   sum(log(1+exp(beta[1]+beta[2]*shuttle$Temp))))
fit=optim(c(0, 0), ll)
fit
```

```
## $par
## [1] 15.0453570 -0.2321977
##
## $value
## [1] 10.1576
##
## $counts
## function gradient
##       99       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```

```
x=seq(30, 90, length=100)
y=exp(fit$par[1]+fit$par[2]*x)/(1+exp(fit$par[1]+fit$par[2]*x))
df=data.frame(x=x, y=y)
plt +
  geom_line(aes(x,y), data=df, col="blue", size=1.2)
```



so at the expected launch temperature of 38 degrees F the estimated probability of a failure was 1.

Using R we can fit such a logistic regression model with

```
fit <- glm(Failure~Temp,
           family=binomial,
           data=shuttle)
fit
```

```
##
## Call:  glm(formula = Failure ~ Temp, family = binomial, data = shuttle)
##
## Coefficients:
## (Intercept)         Temp
##     15.0429      -0.2322
##
## Degrees of Freedom: 22 Total (i.e. Null);  21 Residual
## Null Deviance:       28.27
## Residual Deviance: 20.32     AIC: 24.32
```

What would be a 95% confidence interval for the probability at 32F?

```
tmp <- predict(fit, data.frame(Temp=32),
     type="response", se.fit=TRUE)
round(tmp$fit +c(-1, 1)*qnorm(0.975)*tmp$se.fit, 3)
```

```
## [1] 0.996 1.003
```

but there is something silly about this interval: it goes beyond 1! This is a consequence of using normal theory intervals. Here is a better solution:

```
tmp <- predict(fit, data.frame(Temp=32),
     type="link", se.fit=TRUE)
e <- tmp$fit
r <- tmp$se.fit
cr <- qnorm(0.975)
round(c(exp(e-cr*r)/(1+exp(e-cr*r)),
        exp(e+cr*r)/(1+exp(e+cr*r))), 3)
```

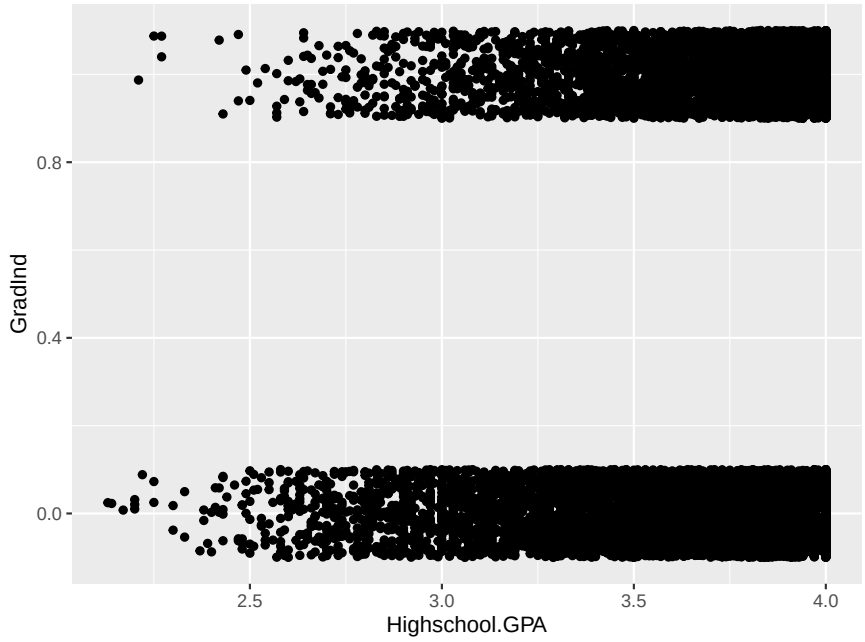
```
##     1     1
## 0.476 1.000
```

but this has a much lower (likely to low) lower limit.

7.5.1.2 Example (8.5.3) UPR Admissions data

Let's say we want to predict whether or not somebody will graduate from UPR as a function of their Highschool GPA. We will give them six years to do so, so we will consider all applicants from 2003-2007:

```
dta <- upr[upr$Year<=2007,
           c("Highschool.GPA", "Graduated")]
dta$GradInd <- ifelse(dta$Graduated=="Si", 1, 0)
plt <- ggplot(dta, aes(Highschool.GPA, GradInd)) +
  geom_jitter(width=0, height=0.1)
plt
```

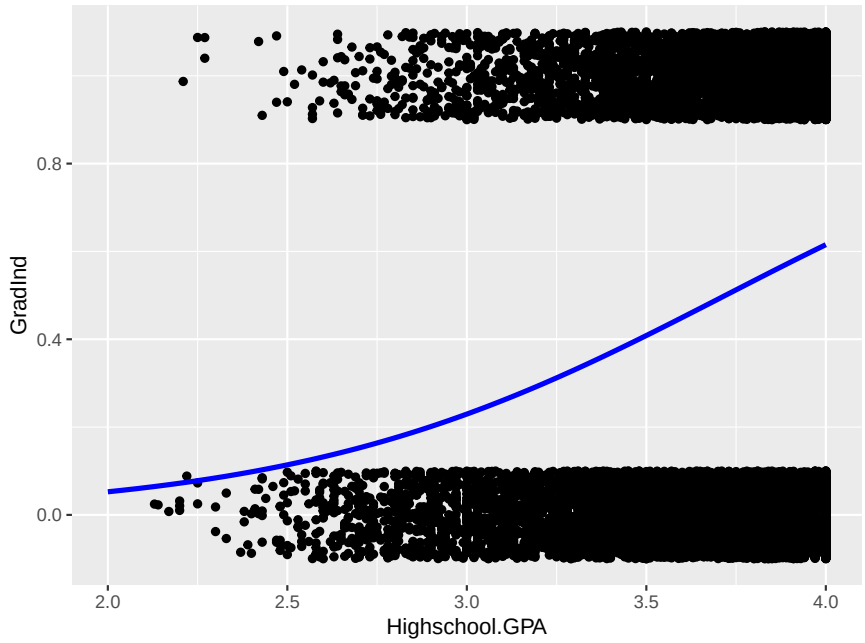



```
fit <- glm(GradInd~Highschool.GPA,
           family=binomial,
           data=dta)
```

```
fit
```

```
##
## Call:  glm(formula = GradInd ~ Highschool.GPA, family = binomial, data = dta)
##
## Coefficients:
##    (Intercept)  Highschool.GPA
##         -6.252           1.680
##
## Degrees of Freedom: 11409 Total (i.e. Null);  11408 Residual
## Null Deviance:       15790
## Residual Deviance: 14960     AIC: 14960
```

```
x <- seq(2, 4, length=100)
df <- data.frame(x=x,
         y=predict(fit, data.frame(Highschool.GPA=x),
                                   type="response"))
plt +
  geom_line(data=df, aes(x, y),
            color="blue", size=1.2)
```

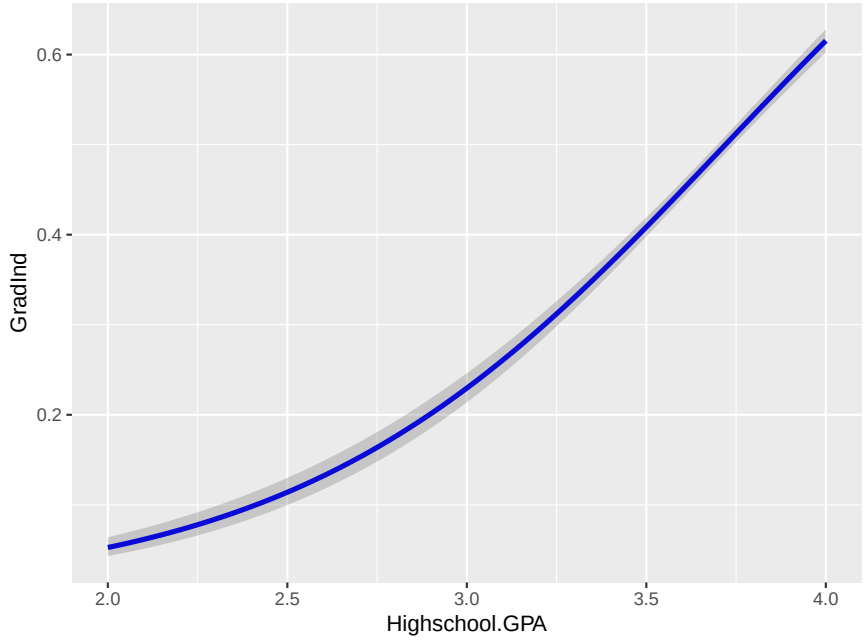


so the probability of a successful graduation does increase, but only to about 62%.

Let's find a pointwise confidence band for the success probability:

```
tmp <- predict(fit, data.frame(Highschool.GPA=x),
     type="link", se.fit=TRUE)
e <- tmp$fit
r <- tmp$se.fit
cr <- qnorm(0.975)
```

```
ymin <- exp(e-cr*r)/(1+exp(e-cr*r))
ymax <- exp(e+cr*r)/(1+exp(e+cr*r))
df1 <- data.frame(x=x, ymin=ymin, ymax=ymax)
ggplot(dta, aes(Highschool.GPA, GradInd)) +
 geom_line(data=df, aes(x, y),
           color="blue", size=1.2) +
 geom_ribbon(data=df1,
             aes(x=x, ymin=ymin, ymax=ymax),
             alpha=0.2, inherit.aes = FALSE)
```



7.5.2 Poisson Regression

7.5.2.1 Example (8.5.4) Warp Breaks: the data set gives the results of an experiment to determine the effect of wool type (A or B) and tension (low, medium or high) on the number of warp breaks per loom. Data was collected for nine looms for each combination of settings.

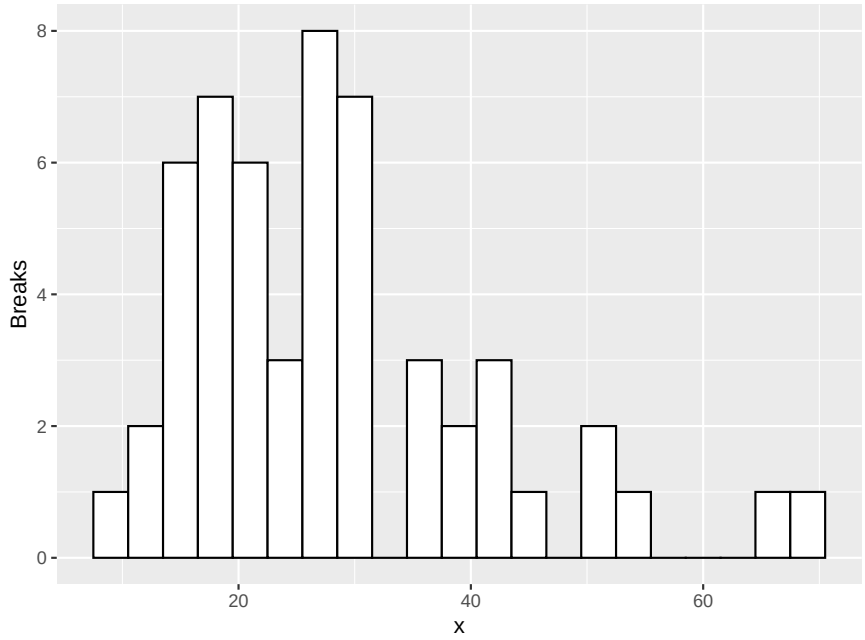
```
kable.nice(head(warpbreaks))
```

	breaks	wool	tension
1	26	A	L
2	30	A	L
3	54	A	L
4	25	A	L
5	70	A	L
6	52	A	L

we want to build a model relating the wool type and tension to the number of breaks.

What distribution might be appropriate for *breaks*? First, let's have a look at them:

```
bw <- diff(range(warpbreaks$breaks))/20
ggplot(warpbreaks, aes(x=breaks)) +
  geom_histogram(color = "black",
    fill = "white",
    binwidth = bw) +
    labs(x = "x", y = "Breaks")
```



our data is counts with a bit of skew to the right. This is typical for data from a *Poisson* distribution.

Here is another argument in favor of a Poisson: Each loom could be considered as a series of small intervals. We then would have a large number of such intervals, each of which has a small probability of a break. The total number of breaks would be the sum of the breaks in each interval, and therefore would be Binomial. But in this case the Poisson approximation to the Binomial would be very good.

Again we want to use regression to relate type and tension to breaks. In the case of a Poisson response variable the link function is given by the logarithm.

```
fit <- glm(breaks~wool*tension,
           data=warpbreaks,
           family=poisson)
summary(fit)
```

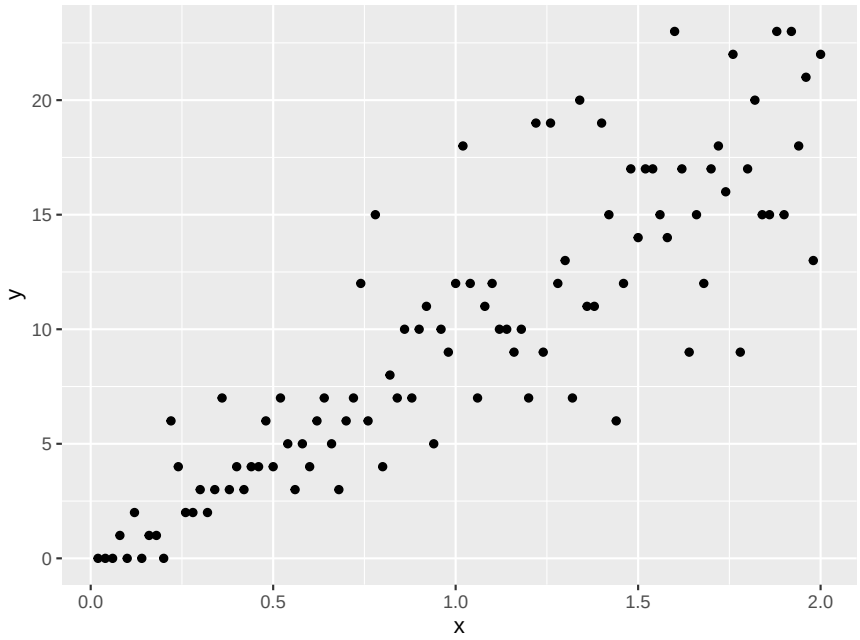
```
##
## Call:
## glm(formula = breaks ~ wool * tension, family = poisson, data = warpbreaks)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3383  -1.4844  -0.1291   1.1725   3.5153
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     3.79674    0.04994  76.030  < 2e-16
## woolB          -0.45663    0.08019  -5.694 1.24e-08
## tensionM       -0.61868    0.08440  -7.330 2.30e-13
## tensionH       -0.59580    0.08378  -7.112 1.15e-12
## woolB:tensionM  0.63818    0.12215   5.224 1.75e-07
## woolB:tensionH  0.18836    0.12990   1.450    0.147
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 297.37  on 53  degrees of freedom
## Residual deviance: 182.31  on 48  degrees of freedom
## AIC: 468.97
##
## Number of Fisher Scoring iterations: 4
```

and we see that all terms except one interaction term are stat. significant.

Let's do our own little study of Poisson regression. First we generate some data:

```
x <- 1:100/50
df <- data.frame(x=x, y=rpois(100, 10*x))
plt <- ggplot(data=df, aes(x, y)) +
  geom_point()
plt
```

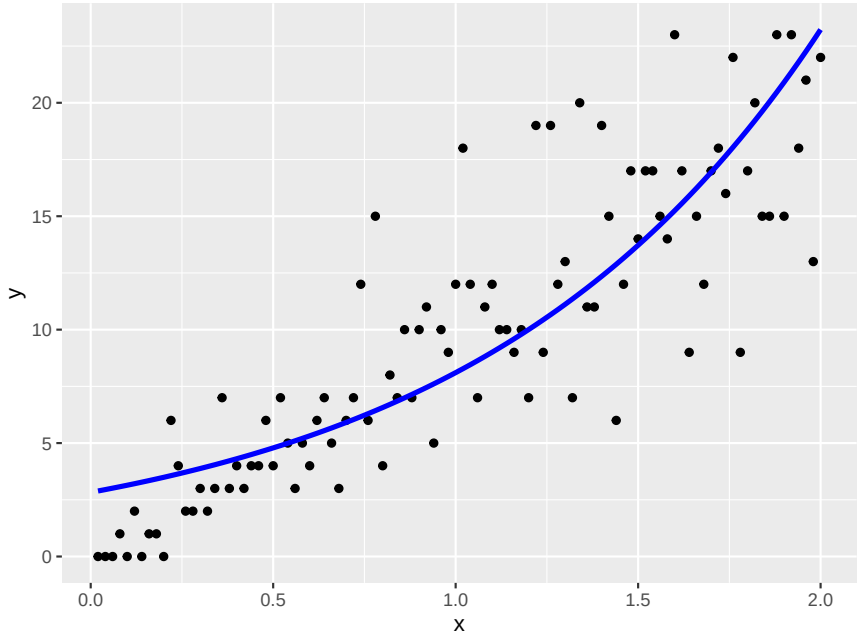



```
fit <- glm(y~x,
           data=df,
           family=poisson)
summary(fit)
```

```
##
## Call:
## glm(formula = y ~ x, family = poisson, data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6425  -0.9848  -0.1348   0.5833   2.9190
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.03973    0.08816   11.79   <2e-16
## x            1.05278    0.06143   17.14   <2e-16
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 472.36  on 99  degrees of freedom
## Residual deviance: 146.59  on 98  degrees of freedom
## AIC: 524.89
##
## Number of Fisher Scoring iterations: 5
```

```
df1 <- df
df1$y <- predict(fit, type="response")
plt +
```

```
  geom_line(data=df1, aes(x, y), color="blue", size=1.2)
```



and that looks quite good!

7.5.3 Generalized Linear Models

Logistic and Poisson regression are examples of *Generalized Linear Models*, which can be characterized as follows: we have

1. Independent random variables $y_1, .., y_n$ with $E[y_i] = \mu_i$ and density function from an exponential family.
2. A linear predictor $\boldsymbol{x_i'\beta}$
3. A link function $g(\mu_i) = \boldsymbol{x_i'\beta}$

7.5.3.1 Definition (8.5.5) A distribution is said to belong to the *exponential family* if its density can be written as

$$f(\boldsymbol{x};\boldsymbol{\theta}) = h(\boldsymbol{x}) \exp\left\{\boldsymbol{\theta}' T(\boldsymbol{x}) - A(\boldsymbol{\theta})\right\}$$

where

- θ is a vector of parameters
- $T(x)$ is a vector of sufficient statistics
- A is a function of θ alone and h is a function of x alone

we have

$$\int f(x;\theta)dx =$$
$$\int h(x)\exp\left\{\theta^T T(x) - A(\theta)\right\}dx =$$
$$\exp\left\{-A(\theta)\right\}\int h(x)\exp\left\{\theta^T T(x)\right\}dx = 1$$

so

$$A(\theta) = \log\left[\int h(x)\exp\left\{\theta^T T(x)\right\}dx\right]$$

7.5.3.2 Examples

- Bernoulli

$$f(x;p) = p^x(1-p)^{1-x}$$
$$\exp\left\{x\log p + (1-x)\log(1-p)\right\} =$$
$$\exp\left\{x(\log p - \log(1-p)) + \log(1-p)\right\} =$$
$$\exp\left\{x\log\frac{p}{1-p} + \log(1-p)\right\}$$
$$\exp\left\{x\theta - \log(1+e^\theta)\right\}$$
$$\theta = \log\frac{p}{1-p}$$
$$h(x) = 1$$
$$T(x) = x$$
$$A(\theta) = -\log(1+e^\theta)$$

because

$$\theta = \log\frac{p}{1-p}$$
$$e^\theta = \frac{p}{1-p}$$
$$p = \frac{e^\theta}{1+e^\theta}$$
$$1-p = \frac{1}{1+e^\theta}$$
$$\log(1-p) = -\log(1+e^\theta)$$

- Normal

$$\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{1}{2\sigma^2}\left(x-\mu\right)^2\right\} =$$
$$\frac{1}{\sqrt{2\pi}}\exp\left\{-\frac{1}{2\sigma^2}\left(x^2-2x\mu+\mu^2\right)-\log\sigma\right\} =$$
$$\frac{1}{\sqrt{2\pi}}\exp\left\{-\frac{x^2}{\sigma^2}+\frac{x\mu}{\sigma^2}-\frac{\mu^2}{2\sigma^2}-\log\sigma\right\}$$

so

$$\theta = (\mu/\sigma^2, -1/(2\sigma^2)^T$$
$$h(x) = \frac{1}{\sqrt{2\pi}}$$
$$T(x) = (x, x^2)^T$$
$$A(\theta) = \frac{\mu^2}{2\sigma^2} + \log\sigma =$$
$$-\theta_1^2/(4\theta_2) - \frac{1}{2}\log(-2\theta_2)$$

—

The likelihood function is given by

$$L(\boldsymbol{y};\boldsymbol{\theta}) = \prod_i h(\boldsymbol{y_i})\exp\left\{\boldsymbol{\theta}' T(\boldsymbol{x}) - A(\boldsymbol{\theta})\right\} =$$
$$[\prod_i h(\boldsymbol{y_i})]\exp\left\{\sum_i \boldsymbol{\theta}' T(\boldsymbol{x}) - nA(\boldsymbol{\theta})\right\}$$

and so

$$\log L(\boldsymbol{y};\boldsymbol{\theta}) = K\left\{\sum_i \boldsymbol{\theta}' T(\boldsymbol{x}) - nA(\boldsymbol{\theta})\right\}$$

For the exponential family we have

$$E[y_i] = \mu_i = A'(\theta)$$

7.6 Classification

7.6.0.1 Example (8.6.1) One of the most famous data sets in Statistics was first studied by Fisher, his iris data. For each of three types of iris flowers (Iris setosa, Iris virginica and Iris versicolor) we have four measurements: the lengths and the widths of the Petal and the Sepal. The goal is to determine from these measurements the type of flower.

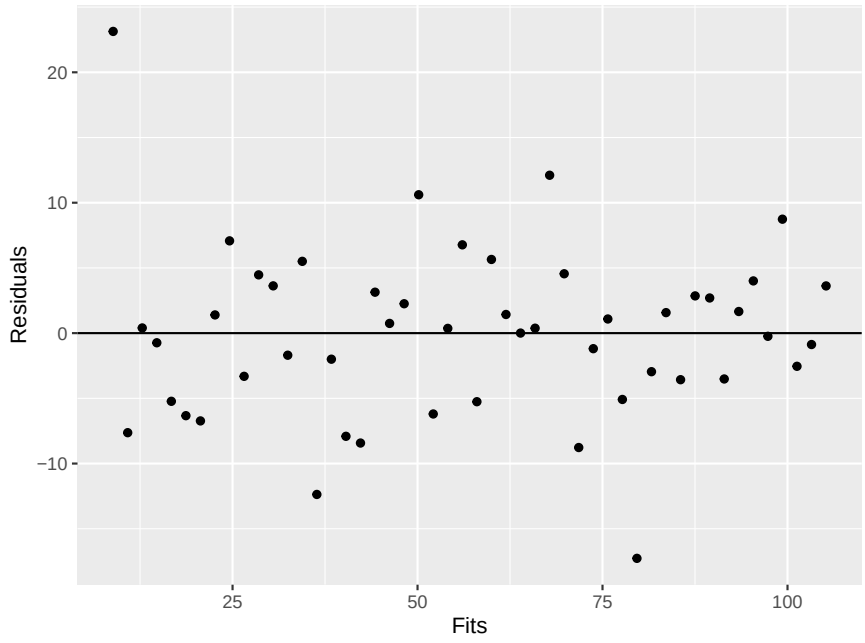
```
kable.nice(head(iris), do.row.names = FALSE)
```

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa

```
table(iris$Species)
```

```
##
##     setosa versicolor  virginica
##         50         50         50
```

```
pushViewport(viewport(layout = grid.layout(3, 3)))
print(ggplot(data=iris,
          aes(Sepal.Length, Sepal.Width, color=Species)) +
      geom_point() + theme(legend.position="none"),
  vp=viewport(layout.pos.row=1, layout.pos.col=1))
print(ggplot(data=iris,
         aes(Sepal.Length, Petal.Length, color=Species)) +
      geom_point() + theme(legend.position="none"),
  vp=viewport(layout.pos.row=1, layout.pos.col=2))
print(ggplot(data=iris,
             aes(Sepal.Length, Petal.Width, color=Species)) +
      geom_point() + theme(legend.position="none"),
  vp=viewport(layout.pos.row=1, layout.pos.col=3))
print(ggplot(data=iris,
             aes(Sepal.Width, Petal.Length, color=Species)) +
      geom_point() + theme(legend.position="none"),
  vp=viewport(layout.pos.row=2, layout.pos.col=2))
print(ggplot(data=iris,
             aes(Sepal.Width, Petal.Width, color=Species)) +
      geom_point() + theme(legend.position="none"),
  vp=viewport(layout.pos.row=2, layout.pos.col=3))
print(ggplot(data=iris,
             aes(Petal.Length, Petal.Width, color=Species)) +
      geom_point() + theme(legend.position="none"),
  vp=viewport(layout.pos.row=3, layout.pos.col=3))
```



It is clear that we the different flowers have different values of the predictors, but with some overlap.

If we code the response variable Species (say setosa=0, versicolor=1 and virginica=2) we again have a regression problem with a discrete response variable like in the last section. However, there are two differences:

1. We do not assume some probability distribution (Binomial, Poisson) for y

2. The goal is to find a model that let's us assign a new case to one of the groups, not to find the probability of group membership. Of course those two are closely related.

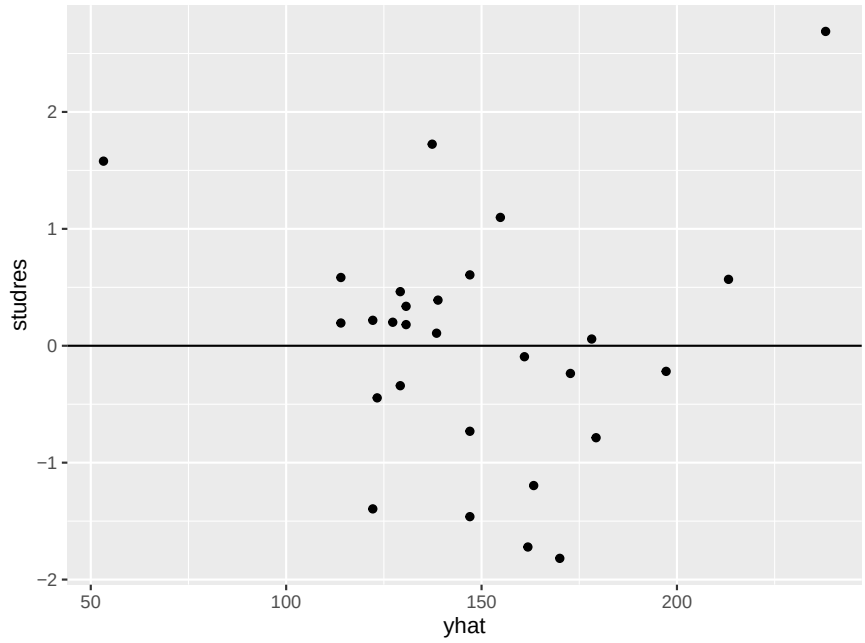
7.6.0.2 Example (8.6.2) We will consider the three artificial examples. Here is a routine that generates some data from each of them:

```
gen.ex <- function(which, n=50) {
  library(mvtnorm)
  ex1 <- function(mu=2, n=50) {
    x1 <- rmvnorm(n, mean=c(0,0), sigma=diag(2))
    x2 <- rmvnorm(n, mean=c(mu,mu), sigma=diag(2))
    data.frame(x=c(x1[, 1], x2[, 1]),
               y=c(x1[, 2], x2[, 2]),
               group=rep(c("A", "B"), each=n))
  }
  ex2 <- function(mu=2, n=50) {
    x <- cbind(runif(10000), runif(10000, -1, 1))
    x <- x[x[, 1]^2 + x[, 2]^2<1, ]
```

```
    x <- x[1:n, ]
    y <- cbind(runif(10000, 0, 2), runif(10000, -2, 2))
    y <- y[y[, 1]^2 + y[, 2]^2>0.9, ]
    y <- y[1:n, ]
    data.frame(x=c(x[, 1], y[, 1]),
               y=c(x[, 2], y[, 2]),
               group=rep(c("A", "B"), each=n))
  }
  ex3 <- function(mu=2, n=33) {
   x1 <- rmvnorm(n, mean=c(0, 0), sigma=diag(2))
   x2 <- rmvnorm(n, mean=c(mu, mu), sigma=diag(2))
   x3 <- rmvnorm(n, mean=2*c(mu, mu), sigma=diag(2))
   data.frame(x=c(x1[, 1], x2[, 1], x3[, 1]),
               y=c(x1[, 2], x2[, 2], x3[, 2]),
               group=rep(c("A", "B", "C"), each=n))
  }
  if(which==1)
      df <- ex1(n=n)
  if(which==2)
      df <- ex2(n=n)
  if(which==3)
      df <- ex3(n=n)
  df$Code <- ifelse(df$group=="A", 0, 1)
  if(which==3)
    df$Code[df$group=="C"] <- 2
  df
}
```

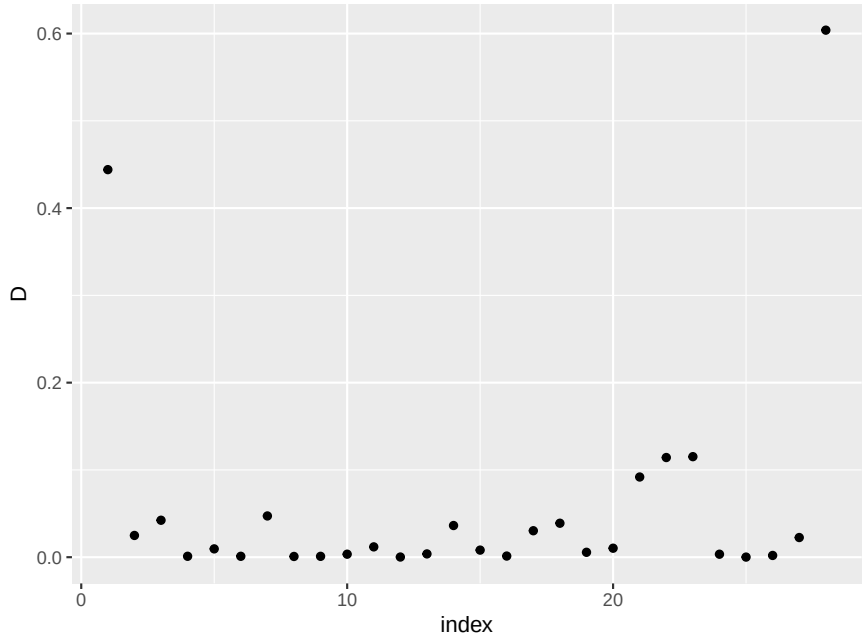
- two types, fairly simple separation

```
ggplot(data=gen.ex(1, n=150), aes(x, y,color=group)) +
  geom_point() +
   labs(x="x1",y="x2") +
  theme(legend.position="none")
```



- two types, more complicated separation

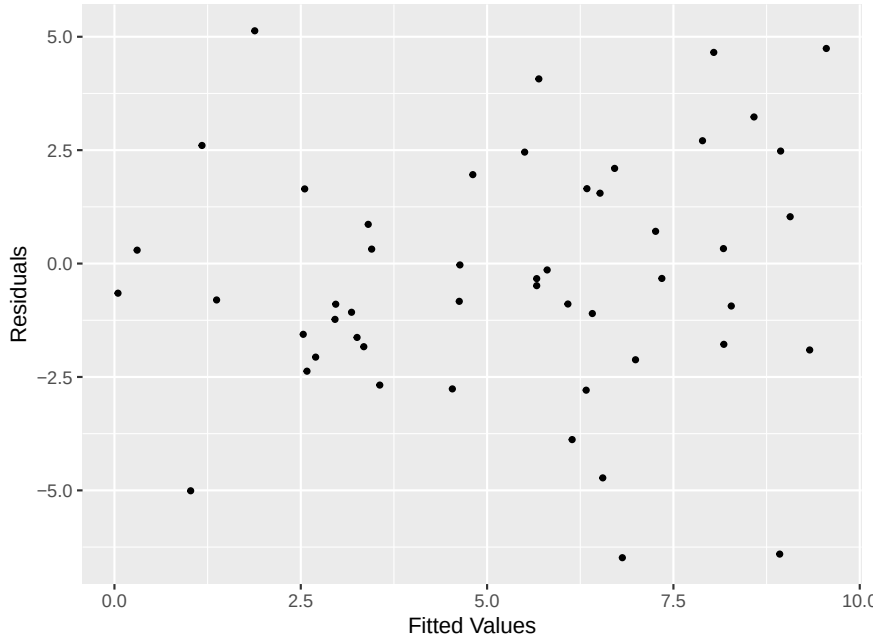
```
ggplot(data=gen.ex(2, n=150), aes(x, y,color=group)) +
  geom_point() +
   labs(x="x1",y="x2") +
  theme(legend.position="none")
```



- three types

```
ggplot(data=gen.ex(3, n=150), aes(x, y,color=group)) +
  geom_point() +
```

```
  labs(x="x1",y="x2") +
theme(legend.position="none")
```



Let's say we code a response variable y as 0 if "green" and 1 if "red" if there are two groups (models 1 and 2) or with 0, 1 and 2 if there are three groups (model 3). Then we run the linear regression of y on x1 and x2.

Now the model is the same as always: $\boldsymbol{y = X\beta + \epsilon}$, and so the estimator is as always $\boldsymbol{\hat{\beta} = (X'X)^{-1}X'y}$. Say we have two groups ordered so that all cases with y=0 come first, then

$$\boldsymbol{y} = \begin{pmatrix} 0 & 0 & \dots & 0 & 1 & \dots & 1 \end{pmatrix}'$$

Let's say there are n observations from group 0 and m from group 1, so

$$\boldsymbol{X'y} = \begin{pmatrix} \boldsymbol{X'_n} & \boldsymbol{X'_m} \end{pmatrix} \boldsymbol{y} = \boldsymbol{X'_m j} = \begin{pmatrix} \sum_{i=m+1}^{m+n} x_{i1} \\ \vdots \\ \sum_{i=m+1}^{m+n} x_{k1} \end{pmatrix}$$

Finally we assign a point (x1, x2) to "green" if its predicted response is <0.5, and to "red" otherwise for models 1 and 2, and depending on whether its predicted response is <2/3 or >4/3 for model 3.

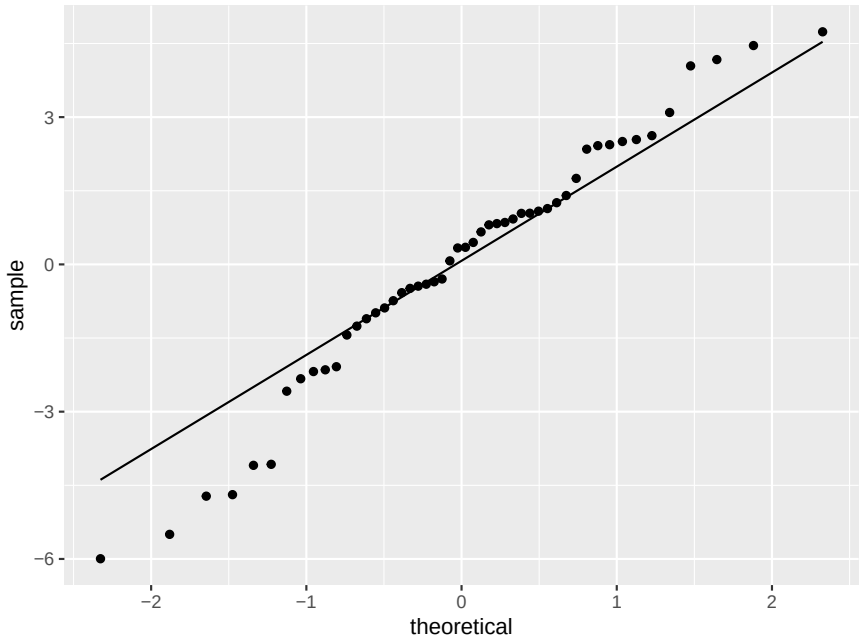
Of course in the case of two groups we could also use *logistic regression*, but we won't pursue this idea here.

To see what this looks like we find an even spaced grid and predict the color for each point. Then we overlay that grid onto the graph. This is done in

```
make.grid <- function(df) {
  x <- seq(min(df$x), max(df$x), length=100)
  y <- seq(min(df$y), max(df$y), length=100)
  expand.grid(x=x, y=y)
}
do.graph <- function(df, df1) {
  print(ggplot(data=df, aes(x, y, color=group)) +
   geom_point(size=2) +
   labs(x="x1",y="x2") +
   theme(legend.position="none") +
   geom_point(data=df1,
              aes(x,y, color=group, alpha=0.1),
              inherits.aes=FALSE))
}
```

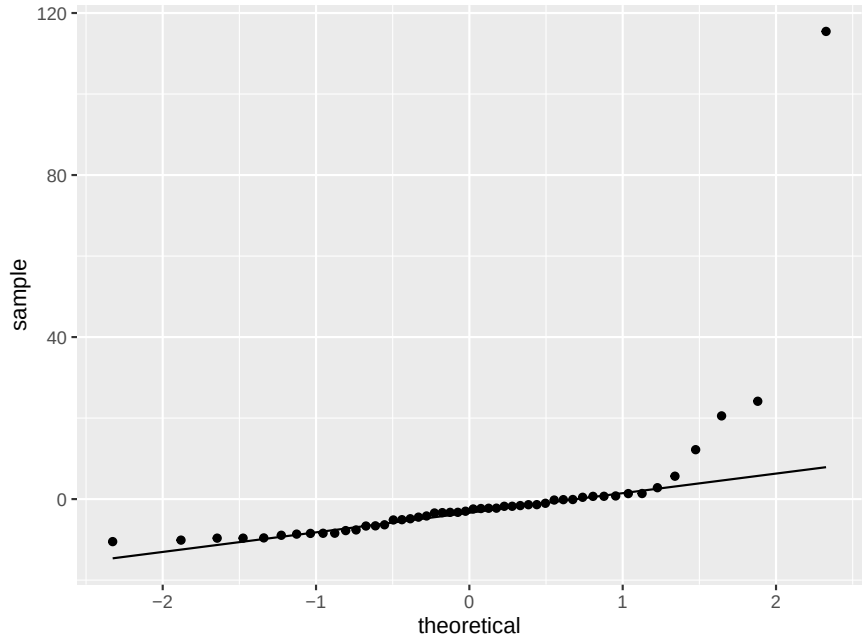
Here our three examples:

```
df <- gen.ex(1)
df$Code <- ifelse(df$group=="A", 0, 1)
fit <- lm(Code~x+y, data=df)
df1 <- make.grid(df)
df1$group <- ifelse(predict(fit, df1)<0.5, "A", "B")
do.graph(df, df1)
```

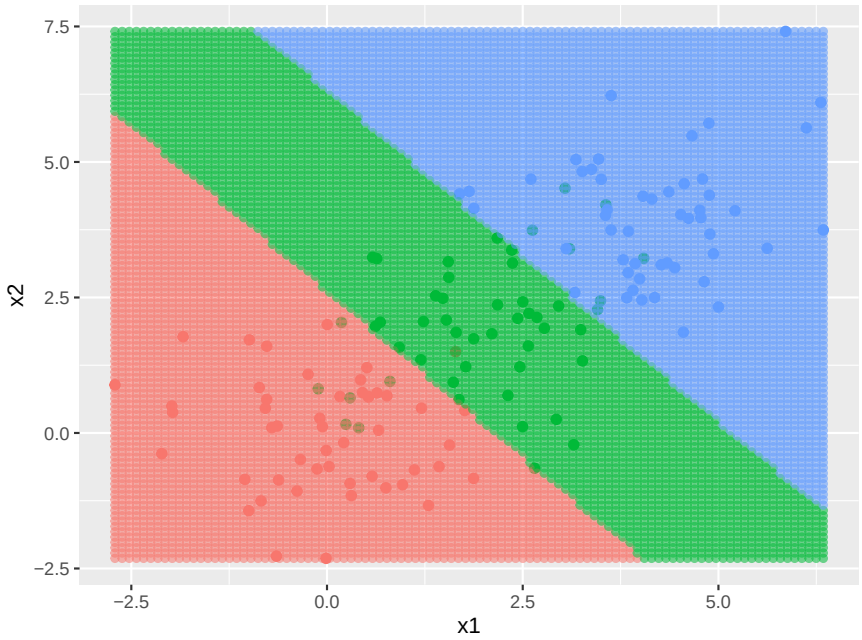



```
df <- gen.ex(2)
df$Code <- ifelse(df$group=="A", 0, 1)
fit <- lm(Code~x+y, data=df)
df1 <- make.grid(df)
df1$group <- ifelse(predict(fit, df1)<0.5, "A", "B")
```

```
do.graph(df, df1)
```




```
df <- gen.ex(3)
df$Code <- ifelse(df$group=="A", 0, 1)
df$Code[df$group=="C"] <- 2
fit <- lm(Code~x+y, data=df)
df1 <- make.grid(df)
tmp <- predict(fit, df1)
df1$group <- ifelse(tmp<2/3, "A", "B")
df1$group[tmp>4/3] <-"C"
do.graph(df, df1)
```



this seems to work ok for examples 1 and 3, not so much for 2.

Let's have a closer look at example 1:

```
df <- gen.ex(1)
fit <- lm(Code~x+y, data=df)
coef(fit)
```

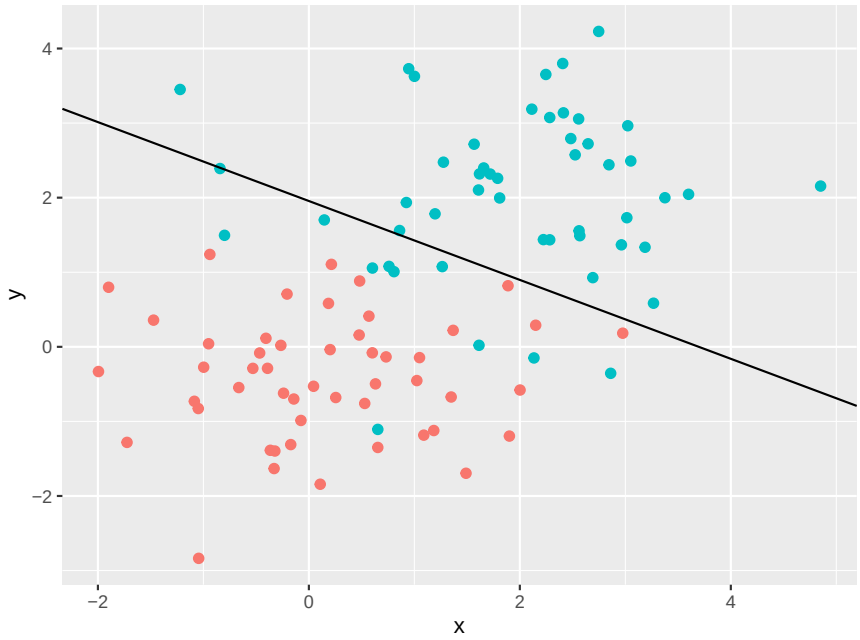
```
## (Intercept)           x           y
##   0.2279637   0.1064668   0.2012716
```

we assign the group depending if the fitted value is $<$ or $>$ than 0.5. What do we get if it is equal to 0.5?

$$
\begin{aligned}
0.5 &= \beta_0 + \beta_1 x_1 + \beta_2 x_2 \\
x_2 &= (0.5 - \beta_0 - \beta_1 x_1)/\beta_2 \\
x_2 &= \frac{0.5 - \beta_1}{\beta_2} - \frac{\beta_1}{\beta_2} x_1
\end{aligned}
$$

Let's add that line to the graph:

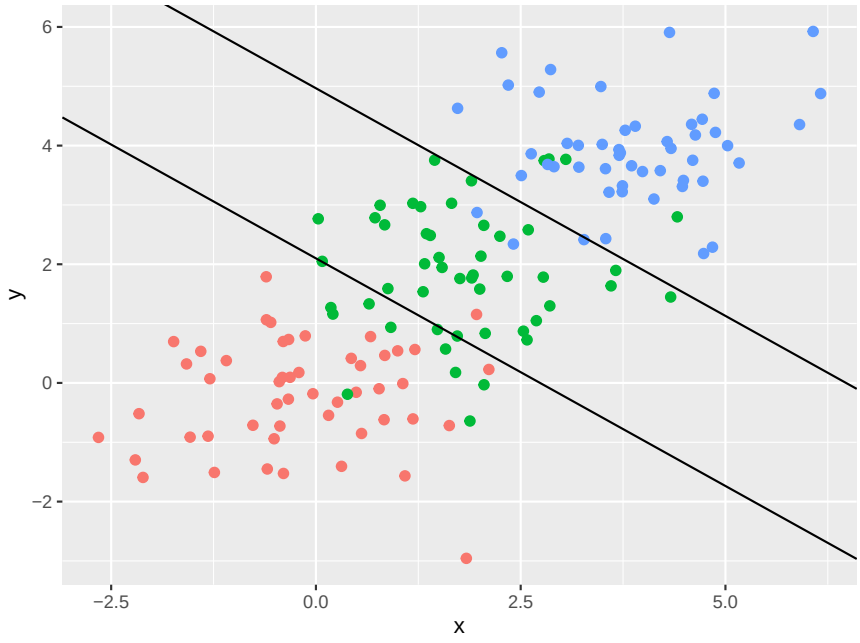
```
ggplot(data=df, aes(x, y, color=group)) +
  geom_point(size=2) +
  theme(legend.position="none") +
  geom_abline(intercept = (0.5-coef(fit)[2])/coef(fit)[3],
              slope=-coef(fit)[2]/coef(fit)[3])
```



and this is called the *decision boundary*.

It is easy to see that in example 3 it works like this:

```
df <- gen.ex(3)
fit <- lm(Code~x+y, data=df)
ggplot(data=df, aes(x, y, color=group)) +
  geom_point(size=2) +
  theme(legend.position="none") +
  geom_abline(intercept = (2/3-coef(fit)[2])/coef(fit)[3],
              slope=-coef(fit)[2]/coef(fit)[3]) +
  geom_abline(intercept = (4/3-coef(fit)[2])/coef(fit)[3],
              slope=-coef(fit)[2]/coef(fit)[3])
```



7.6.1 Misclassification Rate

One thing that sets a classification problem apart from regression is that here we have an obvious way to judge how good a method is, namely the **miss-classification rate**: What percentage of the observations are given the wrong label?

Let's see:

```
msr <- function(x, y) {
  z <- table(x, y)
  round((sum(z)-sum(diag(z)))/sum(z)*100, 1)
}
```

```
df <- gen.ex(1, n=1000)
fit <- lm(Code~x+y, data=df)
pred <- ifelse(predict(fit)<0.5, "A", "B")
table(df$group, pred)
```

```
##    pred
##       A   B
##   A 919  81
##   B  83 917
```

```
msr(df$group, pred)
```

```
## [1] 8.2
```

```
df <- gen.ex(2, n=1000)
fit <- lm(Code~x+y, data=df)
```

```
pred <- ifelse(predict(fit)<0.5, "A", "B")
msr(df$group, pred)
```

```
## [1] 19.5
```

```
df <- gen.ex(3, n=1000)
fit <- lm(Code~x+y, data=df)
tmp <- predict(fit)
pred <- ifelse(tmp<2/3, "A", "B")
pred[tmp>4/3] <- "C"
msr(df$group, pred)
```

```
## [1] 11.8
```

7.6.2 Overfitting and Cross-validation

Of course these misclassification rates are to optimistic: we calculated it on the same data set that we fit on. We should always *train* and *test* on different data sets, maybe using cross-validation:

```
df <- gen.ex(1, n=1000)
print(dim(df))
```

```
## [1] 2000    4
```

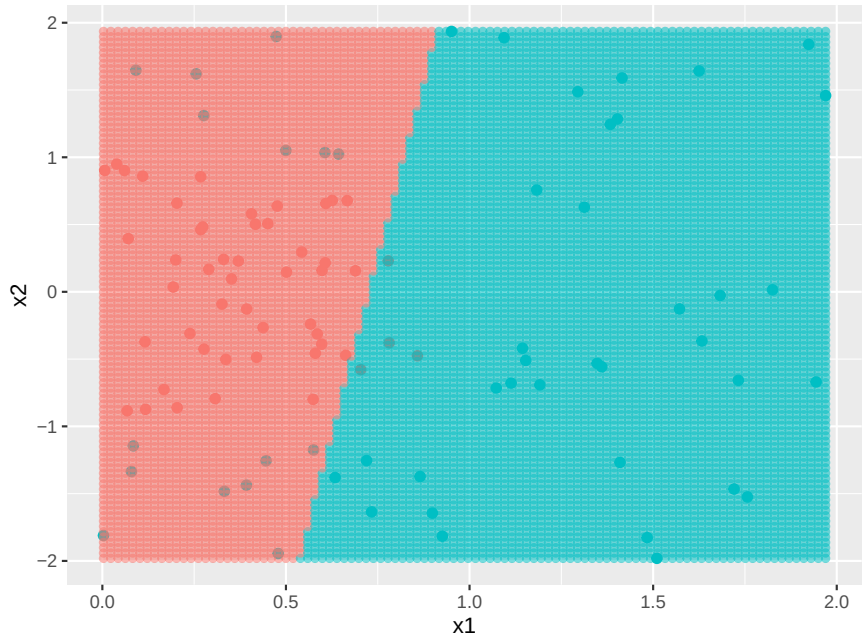
```
out <- rep(0, 10)
for(i in 1:10) {
  I <- sample(1:2000, size=400)
  fit <- lm(Code~x+y, data=df[-I, ])
  pred <- ifelse(predict(fit, df[I, 1:2])<0.5, "A", "B")
  out[i] <- msr(df$group[I], pred)
}
mean(out)
```

```
## [1] 7.42
```

Here we split the data into 80% for training and 20% for evaluation. Is this a good split? Actually, nobody knows!

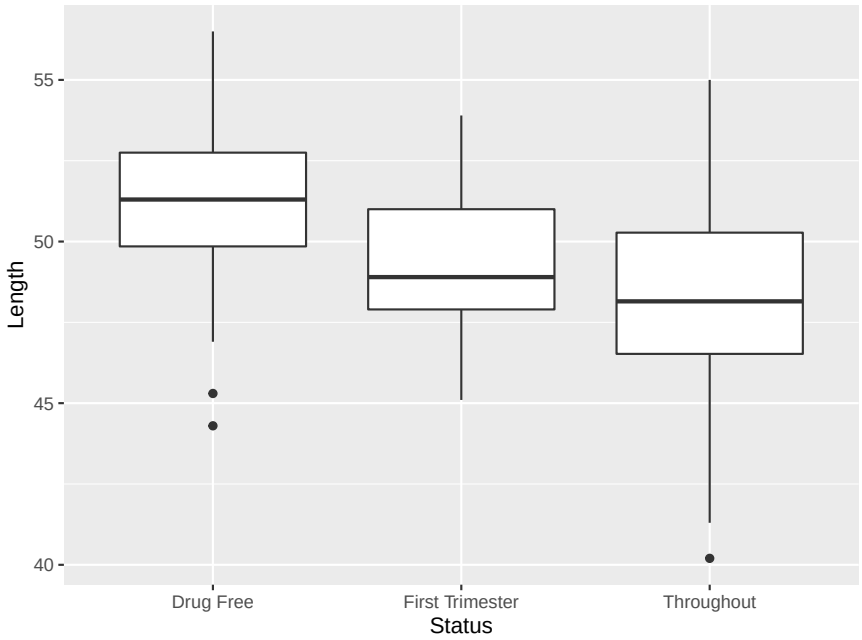
Our method works quite well for examples 1 and 3, but not so much for example 2.

```
df <- gen.ex(2)
df$Code <- ifelse(df$group=="A", 0, 1)
fit <- lm(Code~x+y, data=df)
df1 <- make.grid(df)
df1$group <- ifelse(predict(fit, df1)<0.5, "A", "B")
do.graph(df, df1)
```



shows us why: here a linear decision boundary clearly won't work. So how about a quadratic one?

```
df$x2 <- df$x^2
df$y2 <- df$y^2
df$xy <- df$x*df$y
fit <- lm(Code~x+y+x2+y2+xy, data=df)
df1 <- make.grid(df)
df1$x2 <- df1$x^2
df1$y2 <- df1$y^2
df1$xy <- df1$x*df1$y
df1$group <- ifelse(predict(fit, df1)<0.5, "A", "B")
do.graph(df, df1)
```



and that looks much better!

Here is the mcr based on cross-validation:

```
df <- df[, c(4, 1:2, 5:7)]
out <- rep(0, 10)
for(i in 1:10) {
  I <- sample(1:2000, size=400)
  fit <- lm(Code~x+y+x2+y2+xy, data=df[-I, ])
  pred <- ifelse(predict(fit, df[I, -1])<0.5, "A", "B")
  out[i] <- msr(df$Code[I], pred)
}
mean(out)
```

```
## [1] 7.67
```

The two solutions we have discussed above, linear and quadratic regression, are (slight variations of) what Fisher came up with back when he introduced the Iris data set. They are now called

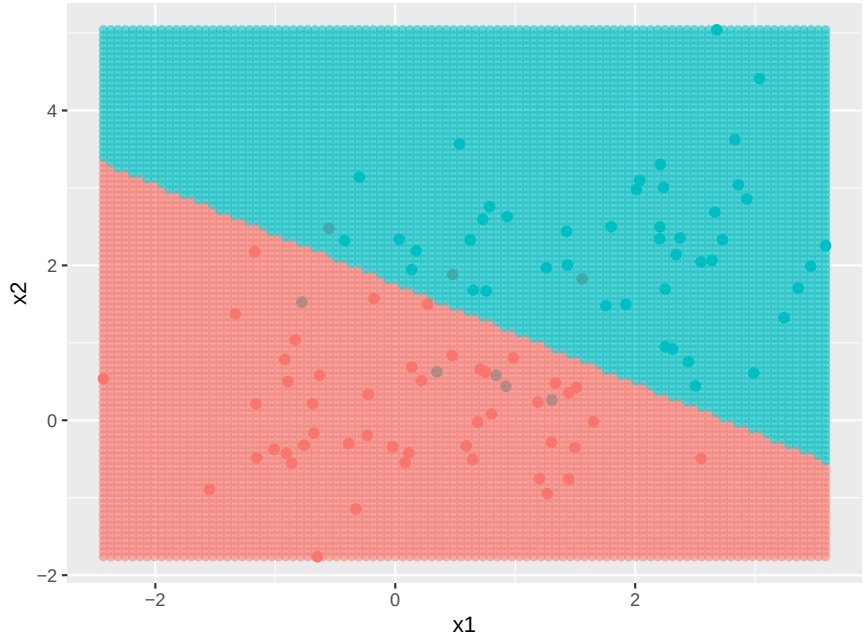
7.6.3 Linear and Quadratic discriminants

and are implemented in R with

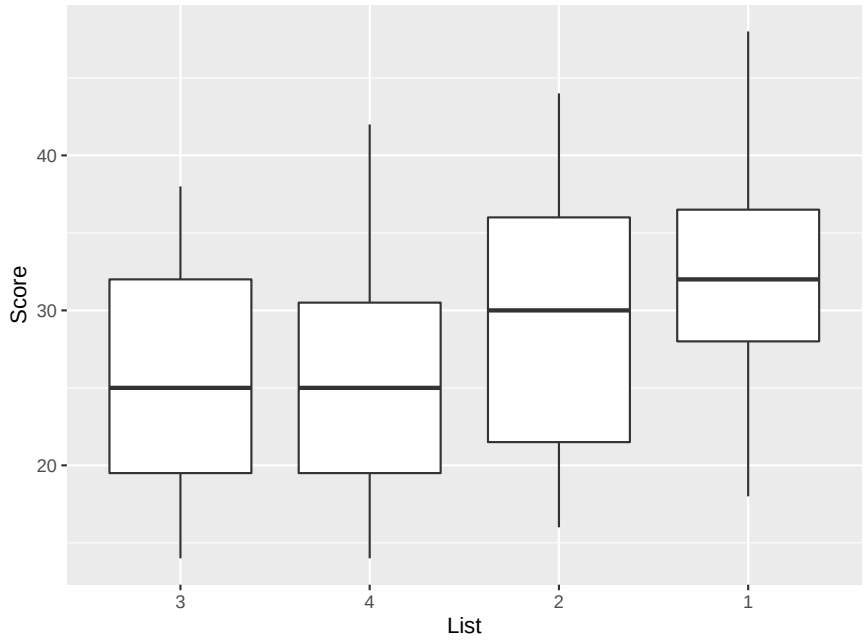
```
library(MASS)
df <- gen.ex(1)
fit <- lda(df$group~x+y, data=df)
df1 <- make.grid(df)
df1$group <- predict(fit, df1)$class
head(df1)
```

```
##           x         y group
## 1 -2.432248 -1.761245     A
## 2 -2.371442 -1.761245     A
## 3 -2.310637 -1.761245     A
## 4 -2.249831 -1.761245     A
## 5 -2.189026 -1.761245     A
## 6 -2.128220 -1.761245     A
```

```
do.graph(df, df1)
```

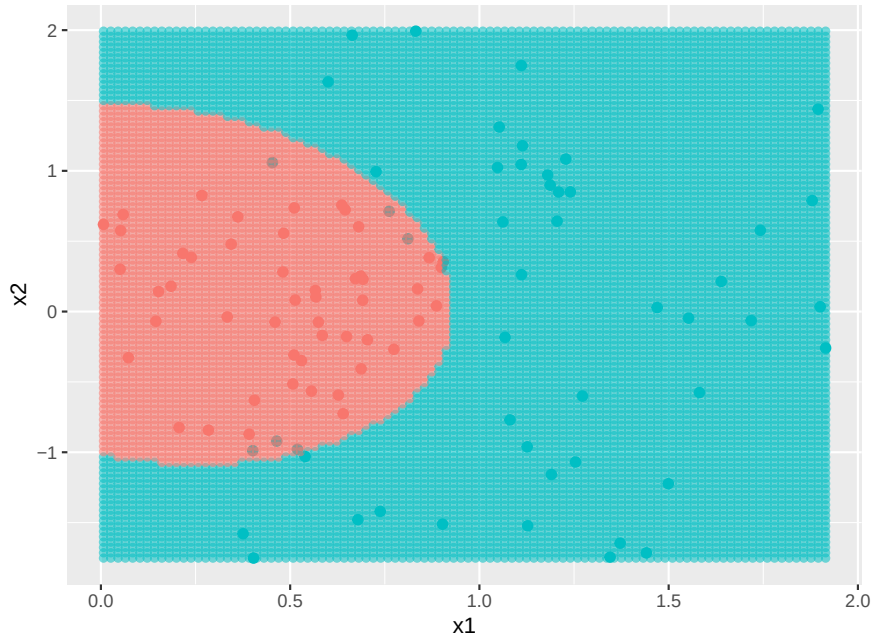



```
df <- gen.ex(3)
fit <- lda(group~x+y, data=df)
df1 <- make.grid(df)
df1$group <- predict(fit, df1)$class
do.graph(df, df1)
```



for example 2 we should use

```
df <- gen.ex(2)
fit <- qda(group~x+y, data=df)
df1 <- make.grid(df)
df1$group <- predict(fit, df1)$class
do.graph(df, df1)
```



Notice a couple of differences between the lm and lda/qda solutions:

- in lda/qda we don't have to do any coding, they accept categorical variables as response.

- there is a difference between the lm and the lda/qda solutions of examples 2 and 3. Do you see what it is, and why?

7.7 Nonparametric Regression

In this last section we will discuss fitting more general functions to data. We will now allow infinite dimensional models. We will still use least squares, so the problem will become as follows: find a function f that minimizes

$$\Phi(\boldsymbol{y}, \boldsymbol{X}, f) = \sum_{i=1}^{n} (y_i - f(\boldsymbol{x}_i))^2$$

Of course in this generality we can always find a function f so that $\Phi(\boldsymbol{y}, \boldsymbol{x}, f) = 0$. For example, if $\boldsymbol{X}$ is $n \times 1$ one can always find a polynomial of degree n such $p(x_i) = y_i$.

One way to make this work is to restrict the set of functions over which to minimize:

1. Allow only polynomials up to some (fixed) degree

2. Allow only cubic splines, that is functions that are piecewise cubic such that at the points where the cubic functions change (called knots) the function still has a continuous derivative.

A more general approach is to use instead a criterion of the form

$$\Phi(\boldsymbol{y}, \boldsymbol{X}, f) = \sum_{i=1}^{n} (y_i - f(\boldsymbol{x}_i))^2 - \lambda \Psi(f)$$

where Ψ is some penalty function and λ a parameter that allows us to determine the amount of penalty. This idea is called *regularization*.

7.7.0.1 Example (8.7.1) The idea of regularization is even useful in standard linear regression. Sometimes is a good idea to make sure the coefficients do not get to large, and so we might minimize

$$\Phi(\boldsymbol{y}, \boldsymbol{X}, f) = \sum_{i} (y_i - (\boldsymbol{X'\beta})_i))^2 - \lambda \sum_{i} |\beta_i|$$

This is known as *ridge regression*.

7.7.0.2 Example (8.7.2) Let's define the space of functions $f : [0, 1] \to R$ which have a continuous second derivative and who have $\int_0^1 (f''(x))^2 dx < R$ for some R>0. Then we can minimize

$$\Phi(\boldsymbol{y}, \boldsymbol{X}, f) = \sum_{i=1}^{n} (y_i - f(\boldsymbol{x}_i))^2 - \lambda \int_0^1 (f''(x))^2 dx$$

It may seem at first very hard and maybe impossible to actually find such a minimum because we are here optimizing over an infinite dimensional space, but in some cases this can actually be done. For example, in this specific case it can be shown that the solution is always a cubic spline!

7.7.1 Reproducing Kernel Hilbert Space (RKHS)

In order to be able to find such a minimum it is necessary to use a function space with sufficient "structure". A good choice are so called *Reproducing Kernel Hilbert Spaces (RKHS)*, which we will now define:

7.7.1.1 Definition (8.7.3)

A *vector space* is a set of elements with operations "addition" and "multiplication" that follow the usual rules.

7.7.1.2 Example (8.7.4)

1. R^d with vector addition and multiplication
2. C[0,1], the set of continuous functions on [0,1] with the usual addition and multiplication, for example (f*g)(x)=f(x)g(x)

7.7.1.3 Definition (8.7.5)

Let V be some vector space V. A mapping $\langle ., . \rangle : V^2 \to \mathbb{R}$ is an *inner product* on V if for $x, y, z \in V$ and $a \in \mathbb{R}$

1. $\langle x, y \rangle \geq 0$
2. $\langle x, y \rangle = \overline{\langle y, x \rangle}$
3. $\langle ax + y, z \rangle = a\langle x, z \rangle + \langle y, z \rangle$

A vector space with an inner product is called an *inner product space*.

Often we also write $\langle x, x \rangle = ||x||^2$ and then ||x|| is called the *norm*.

Such a norm induces a *metric* on the space via $d(x, y) = ||x - y||$.

7.7.1.4 Example (8.7.6)

1. $\mathrm{R^n}$ with $\langle x, y \rangle = \sum x_i y_i$
2. the space of continuous functions C with

$\langle f, g \rangle = \int f(x)g(x)dx$

Note that in an inner product space we have a version of the Pythagorean theorem: if x and y are such that

$\langle x, y \rangle = 0$

they are said to be orthogonal, and then we have

$$||x+y||^2 =$$
$$\langle x+y, x+y\rangle =$$
$$\langle x,x\rangle + \langle x,y\rangle + \langle y,x\rangle + \langle y,y\rangle =$$
$$||x||^2 + ||y||^2$$

7.7.1.5 Definition (8.7.7) A sequence is called a "Cauchy sequence" if for all $\epsilon > 0$ there exists N such that for all n, m>N such $d(x_n, x_m) < \epsilon$.

This definition of convergence has the advantage that one can check it without knowing what the limit of the sequence is

7.7.1.6 Example $x_n = \frac{1}{n}$, $d(x_n, x_m) = |x_n - x_m|$ then if $n < m$

$$|\frac{1}{n} - \frac{1}{m}| = \frac{m-n}{nm} < \frac{m}{mn} = \frac{1}{m} < \epsilon$$

if $n, m > N = 1/\epsilon$

7.7.1.7 Definition (8.7.8) A space is called *complete* if each Cauchy sequence converges.

It may seem at first that this is always true, but it is not:

7.7.1.8 Example Consider the space of functions $L^1[0,1]$ on [0,1] that are continuous and have norm $||f|| = \int_0^1 |f(x)|dx$. Now consider the sequence $f_n(x) = x^n$, so

$$||f_n|| = \int_0^1 x^n dx = \frac{1}{n+1}$$

and so we see that f_n is a Cauchy sequence. But

$$\lim_{n\to\infty} f_n(x) = I_0(x)$$

which is not continuous and therefore is not in $L^1[0,1]$. Therefore we have the definition

7.7.1.9 Definition (8.7.9) A complete inner product space is called a *Hilbert Space*

The main idea here is that Hilbert spaces can be infinite dimensional but retain (almost) all the nice properties of Euclidean space R_d, such as the convergence of Cauchy sequences.

7.7.1.10 Example (8.7.10)

1. $l^2 = \{(a_1, a_2, ..) : \sum a_i^2 < \infty\}$ is a Hilbert space with $\langle \boldsymbol{a}, \boldsymbol{b} \rangle = \sum a_i b_i$

Notice that d-dimensional Euclidean space is a subspace of l^2.

2. $L^2[0,1] = \left\{ f : \int_0^1 f^2(x)dx < \infty \right\}$ is a Hilbert space with $\langle f, g \rangle = \int_0^1 f(x)g(x)dx$

Let $f \in L^2[0,1]$, and let $\{\phi_i\}$ be an orthonormal basis of $L^2[0,1]$. Then we have

$$f(x) = \sum_{i=0}^{\infty} a_i \phi_i(x)$$

where $a_j = \langle f, \phi_j \rangle$. By Parseval's theorem we have

$$||\boldsymbol{a}||_{l^2} = ||f||_{L^2[0,1]}$$

therefore we have an isomorphism from l^2 to $L^2[0,1]$, and in a certain sense the two spaces are the same!

7.7.1.11 Definition (8.7.11) A *linear functional* is a L mapping from some vector space V into the real numbers such that for any $a \in R$

$$L(ax + y) = aL(x) + L(y)$$

A linear functional is called *bounded* if there exists M>0 such that

$$|L(f)| \leq M||f||$$

The mathematical theory that studies such objects is called *functional analysis*.

7.7.1.12 Theorem (8.7.12) Let H be a Hilbert space and $g \in H$ some element of H, then $L(f) = \langle f, g \rangle$ defines a bounded linear functional.

proof

The linearity follows from the definition of the inner product. The boundedness is a consequence of the Cauchy-Schwartz inequality:

$$|L(f)| = \langle f, g \rangle \leq ||f|| \times ||g|| = M||f||$$

and there is a famous theorem that says that

7.7.1.13 Theorem (8.7.13) Riesz representation theorem

Let L be a bounded linear function on a Hilbert space H. Then there exists $g \in H$ such that $L(f) = \langle f, g \rangle$.

In other words, on a Hilbert space all bounded linear functionals are generated by elements of the space and the inner product.

7.7.1.14 Definition (8.7.14) A symmetric bivariate function $K : H \times H \to R$ is called *positive (semi)-definite* if for all n>0 and elements $x_1, , , ., x_n \in H$ the matrix with elements $K(x_i, x_j)$ is positive (semi)-definite.

7.7.1.15 Example (8.7.15)

1. R_d: let $K(\boldsymbol{x}, \boldsymbol{x'}) = \langle \boldsymbol{x}, \boldsymbol{x'} \rangle$, then

$$\boldsymbol{a'Ka} = \sum_{i,j=1}^{n} a_i a_j x_i x_j = ||\sum a_i x_i||^2 \geq 0$$

This is of course the usual case of a positive semi-definite matrix.

2. (Gaussian kernel)

for any $x, y \in R^d$ let

$$K(x, y) = \exp\{-(||x - y||^2/2\sigma^2\}$$

7.7.1.16 Definition Let $\mathcal{X}$ be some space, and $\mathcal{H}$ a Hilbert space of functions from the Cartesian product $\mathcal{X} \times \mathcal{X}$ into the Cartesian product $\mathcal{R}$. Let K be a semi-definite kernel function on $\mathcal{X} \times \mathcal{X}$. $\mathcal{H}$ is called a *reproducing kernel Hilbert space (RKHA)* if for any $x \in \mathcal{X}$ the function $K(\cdot, x) \in \mathcal{H}$ and we have the relation

$$\langle f, K(\cdot, x) \rangle = f(x)$$

for all $f \in \mathcal{H}$.

In other words finding inner products on such a space is easy: one only needs to evaluate the functions. So finding an f that maximizes the inner product means maximizing a function on $\mathcal{R}$. This is often referred to as the *kernel trick*.

How can we find such RKHS's? Let's say we have a space $\mathcal{X}$ and a semi-definite kernel K on the Cartesian product $\mathcal{X} \times \mathcal{X}$. Now consider the space $\mathcal{H}$ of functions on $\mathcal{X}$ defined by

$$f(\cdot) = \sum_{j=1}^{n} a_j K(\cdot, x_j)$$

for some integer $n \geq 1$, points $x_1, .., x_n \in \mathcal{X}$ and weights $a_1, ., , a_n \in \mathcal{R}$. It is easy to check that $\mathcal{H}$ is a vector space under the usual function addition and scalar multiplication.

For any given pair $f, g \in \mathcal{H}$ we define the inner product

$$\langle f, g \rangle_{\mathcal{H}} = \sum_{j,k=1}^{n} a_j b_k K(x_j, \bar{x}_k)$$

and with this definition we have

$$\langle f, K(\cdot, x) \rangle = \sum_{j=1}^{n} a_j K(x_j, x) = f(x)$$

and one can verify that this indeed defines a proper inner product. Finally taking limits as $n \to \infty$ and using the completeness of Hilbert spaces we have

7.7.1.17 Theorem Given any positive semi-definite kernel function K, there is a unique Hilbert space $\mathcal{H}$ in which the kernel satisfies the reproducing property. It is known as the RKHS associated with K.

proof omitted

7.7.1.18 Example $L^2[0,1]$ is not an RKHS. If it were there would have to be a function R_x such that

$$\int_0^1 f(y) R_x(y) dy = f(x)$$

for all $f \in L^2[0,1]$. It can be shown that the only function to do so is $R_x(y) = \infty I_x(y)$, but that function is not in $L^2[0,1]$.

In a certain sense $L^2[0,1]$ is to large a space to be a RHKS, so in order for a space of functions to be a RKHS we need to put further restrictions on them. A common one is

7.7.1.19 Definition For some fixed $n \geq 1$ define the space $\mathcal{H}^n[0,1]$ as the space of all real-valued functions that are n-times continuously differentiable with the n^{th} derivative being Lebesgue integrable and

$$f(0) = f'(0) = ..f^{(n-1)}(0) = 0$$

If we define the inner product

$$\langle f, g \rangle_{\mathcal{H}^n[0,1]} = \int_0^1 f^{(n)}(x) g^{(n)}(x) dx$$

It can be shown that this defines a RKHS. It is called the *Sobolev space of order n.*

7.7.1.20 Theorem Let $\mathcal{H}^n[0,1]$ be the Sobolev space of order n, then

$$K(x,y) = \int_0^1 \frac{(x-z)_+^{n-1}(y-z)_+^{n-1}}{(n-1)!(n-1)!}dz$$

proof (for n=1)

In this case the kernel becomes $K(x,y) = \int_0^1 I_{[0,x]}(z)I_{[0,y]}(z)dz$. Now say $x < y$, then $K(x,y) = \int_0^1 I_{[0,x]}(z)I_{[0,y]}(z)dz = \int_0^x dz = x$, so in general $K(x,y) = \min\{x,y\}$. With this kernel one can now verify all the properties of a RKHS.

7.7.2 Function Interpolation

Say we observe n samples of some unknown function $y_i = f^*(x_i)$ with known design matrix $x_1, .., x_n$. There are two questions of interest:

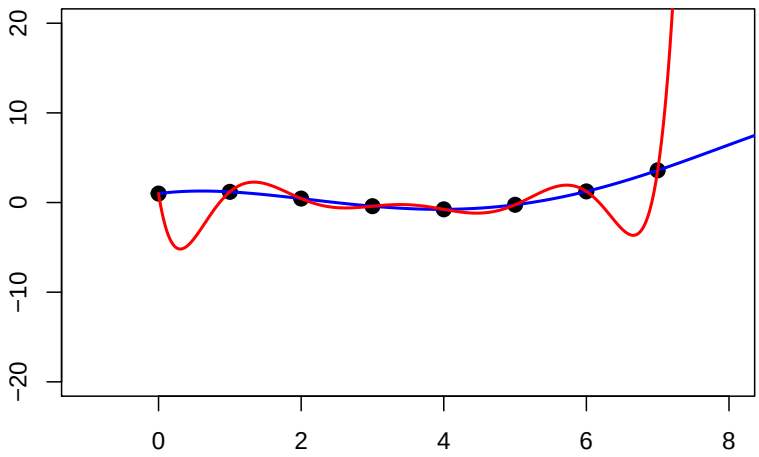
- is there a function f from some specified set that fits the points exactly, that is we have $y_i = f(x_i)$ for all i.?
- if there are many such functions, is the a best one?

The first question can be answered affirmatively by writing down such a function, but the second one is a bit vague: what does best mean? If the set is a RKHS we can use the norm to "order" them and we can write the following optimization problem:

choose $\hat{f} \in \text{argmin}\{||f||_{\mathcal{H}}\}$ such that $y_i = f(x_i)$.

An optimization problem of this type is called a *convex program*.

7.7.2.1 Example



Clearly in the example above the blue curve is "better".

So, how can one solve such an optimization problem? Here is one answer:

7.7.2.2 Theorem Let $\boldsymbol{K}$ be the kernel matrix defined by the design points $\{x_i\}$. The convex program above is feasible if and only if $y \in range(\boldsymbol{K})$, in which case the optimal solution can be written as

$$\hat{f}(\cdot) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \hat{\alpha}_i K(\cdot, x_i)$$

where $\boldsymbol{K}\hat{\alpha} = y/\sqrt{n}$

proof omitted

7.7.3 Fitting via Ridge Regression

In statistics we usually have observations that also include some random noise, so our model is

$$y_i = f^*(x_i) + \epsilon_i$$

In this case we no longer want a function that connects all the points but a function that smooths out the random noise. This means we now want to minimize some trade-off between the fit to the data and the Hilbert space norm.For example we might want to minimize the mean-squared difference between the observed data and the fitted values, which leads to the optimiziation problem

$\min \{||f||_{\mathcal{H}} : f \in \mathcal{H}\}$ such that $\frac{1}{2n} \sum_{i=1}^{n} (y_i - f(x_i))^2 < \delta^2$

where $\delta > 0$ is some type of tolerance parameter. It is possible to reformulate such a problem to

$$\hat{f} = \arg\min \left\{ \frac{1}{2n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda_n ||f||_{\mathcal{H}} \right\}$$

Notice that we are now back to least squares!

This method is known as kernel ridge regression.

7.7.3.1 Theorem For all $\lambda_n > 0$ the kernel ridge regression estimate can be written as

$$\hat{f}(\cdot) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \hat{\alpha}_i K(\cdot, x_i)$$

where

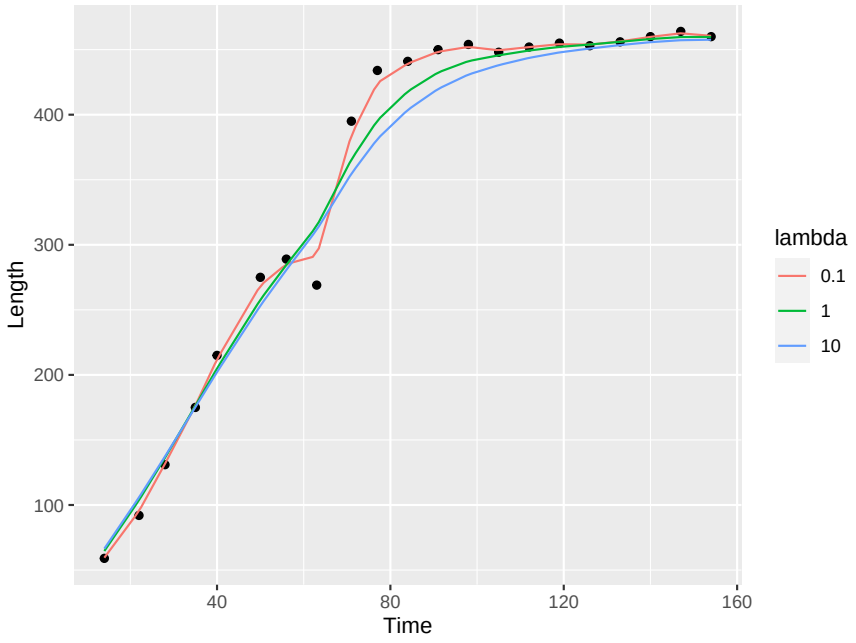
$$\hat{\boldsymbol{\alpha}} = (\boldsymbol{K} + \lambda_n \boldsymbol{I})^{-1} \boldsymbol{y}/\sqrt{n}$$

proof omitted

7.7.3.2 Example Let's use the first order Sobolev space and ridge regression to fit to the Lobster data.

```
K=function(x, y) {
  if(length(x)==1 & length(y)==1) return(1+min(x, y))
  apply(cbind(x, y), 1, min)
}
x=lobster$Time
y=lobster$Length
K1=outer(x, x, K)/length(x)
fhat=function(t, l) {
  n=nrow(K1)
  alphahat=c(solve(K1+l*diag(n))%*%cbind(y))/sqrt(n)
  s=t
  for(i in seq_along(t)) s[i]=sum(alphahat*K(t[i], x))
  s/sqrt(n)
}
t=seq(x[1], max(x), length=100)
df=data.frame(x=c(t, t, t),
              y=c(fhat(t, 0.1), fhat(t, 1), fhat(t, 2)),
              lambda=rep(c("0.1", 1, 10),each=100))
```

```
ggplot(data=lobster, aes(Time, Length)) +
  geom_point() +
  geom_line(aes(x, y,color=lambda), data=df)
```


Length
400
300
200
100
40
80
120
160
Time
lambda
0.1
1
10