ESMA 6661: Theory of Statistics

Dr. Wolfgang Rolke

January 2, 2021

Contents

1 Introduction to Statistics and R

Statistics is the **Science of Uncertainty**, it let's us learn useful information in situations where there is incomplete information.

1.0.0.1 Example (From the book *An Introduction to the Bootstrap* by Efron and Tibshirani) Below we have the results of a small experiment, in which 7 out of 16 mice were randomly selected to receive a new medical treatment, while the remaining 9 mice were assigned to the control group. The treatment was intended to prolong survival after surgery:

Treatment	Control
94	52
197	104
16	146
38	10
99	50
141	31
23	40
	27
	46

The obvious question is: does the new treatment increase survival times?

How can we answer this question? The first thing we can try is to calculate the mean survival times:

```
round(c(mean(treatment), mean(control)), 1)
```

```
## [1] 86.9 56.2
```

so the mice in the treatment group lived about 30.7 days longer than those in the control group.

But why the mean? Why not the median or some other measure of average?

Is there some theoretical justification for the mean as the best way to calculate an average? Is it always best?

Very good, but we really don't care about these 16 mice, they are dead anyway. These 16 mice were just a **random sample** of the **population** of all mice who might receive this treatment or this control, and what we really want to know is whether the treatment **statistically significantly** increases survival.

Some standard terminology:

Population: all of the entities (people, events, things etc.) that are the focus of a study

Census: If all the entities of a population are included in the study.

Sample: any subset of the population

Random sample: a sample found through some randomization (flip of a coin, random numbers on computer etc.)

Simple Random Sample (SRS): each "entity" in the population has an equal chance of being chosen for the sample.

Stratified Sample: First divide population into subgroups, then do a SRS in each subgroup.

Bias: a systematic difference between a sample and its population

Statistically Significant: not due to random chance.

Parameter: any numerical quantity associated with a population

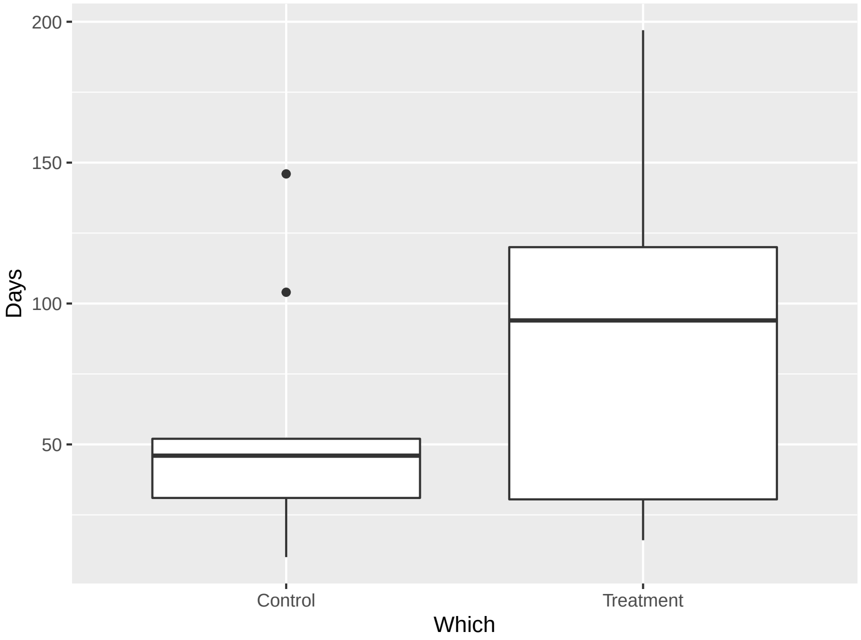
Statistic: any numerical quantity associated with a sample

Here is our question again: from the data we know that the difference of the sample means (a Statistic) is 30.7 days.

What we really want to know is whether the corresponding difference of the population means (a Parameter) is positive.

In other words we want to use the information in the sample to make an **inference** for the corresponding population.

So, how do we find out whether or not the difference of 30.7 days above is statistically significant? Consider the following **boxplot**:



In addition to the **average** of a data set this also gives us an idea of the **variation** in the data.

So, how can we find the variance of the difference of the mean survival times? First we can find the **sample standard deviation**:

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

The idea behind this formula is simple:

- $X_i - \bar{X}$ is the **deviation** (distance) of each individual observation from the mean (these are sometimes called the **residuals** or **errors**)
- squaring the residuals gets rid of minus signs (but so would taking absolute values)
- s would be the square root of the mean of these squared deviations, except we would need to divide by n instead of n-1.

Finding s within each group we get:

```
round(c(sd(treatment), sd(control)), 2)
```

```
## [1] 66.77 42.42
```

- But why the sample standard deviation?

Why not some other measure of "variation"?

This is the standard deviation of the individual observations.

From here we can find the standard errors of the sample means with $s/\sqrt{n}$ (why?)

```
round(c(sd(treatment)/sqrt(length(treatment)),
        sd(control)/sqrt(length(control))), 2)
```

```
## [1] 25.24 14.14
```

Finally we can find the standard error of the difference of the means:

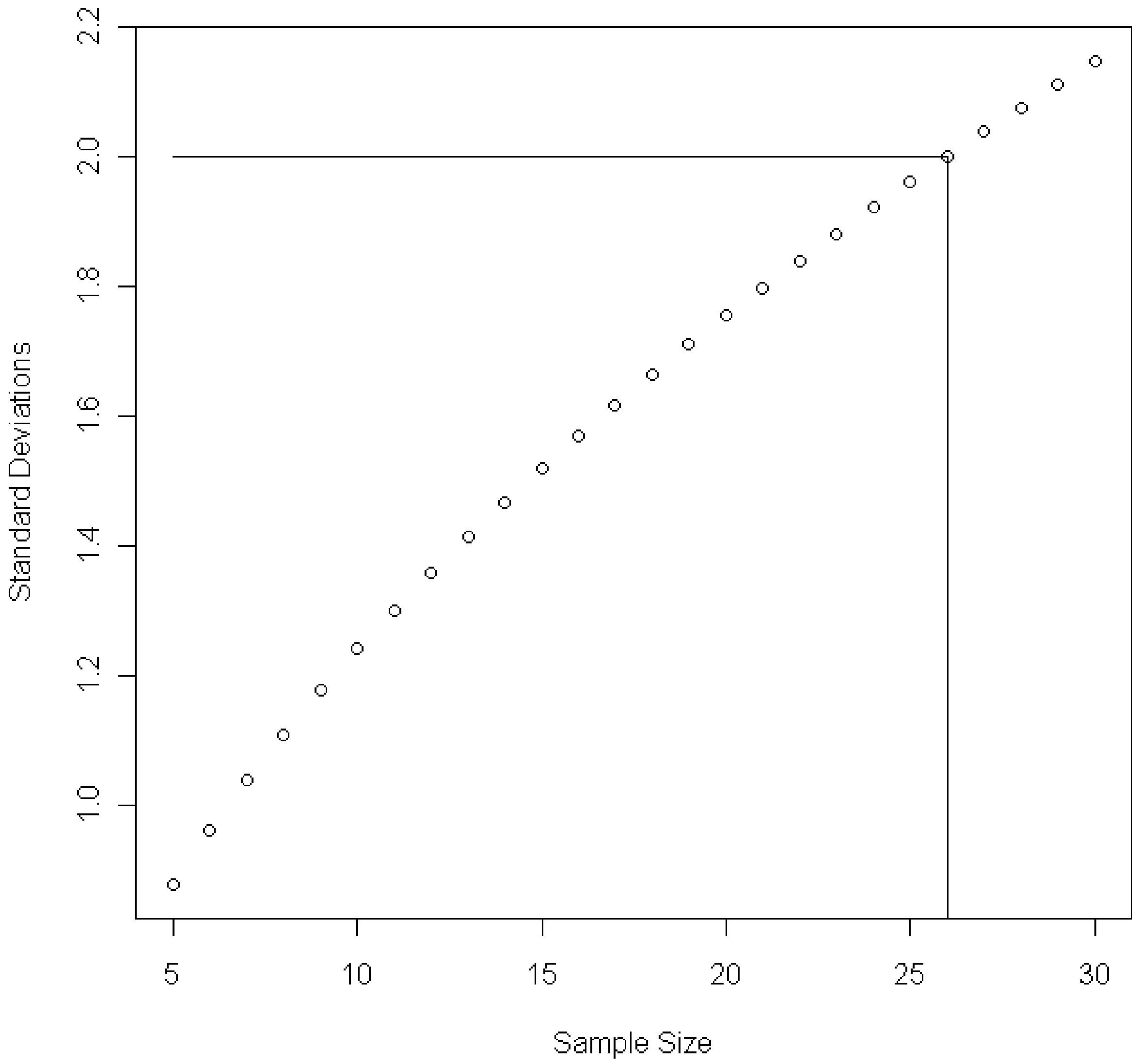
standard error of difference = $\sqrt{25.24^2 + 14.14^2)} = 28.9$.

Why this formula? This is essentially taking the average of the group standard deviations, so why not use (25.24+14.14)/2=19.67?

So we know we have a sample mean difference of 30.7 with a standard error of 28.9, that is the sample mean difference is 30.7/28.9 =1.05 **standard deviations** above 0. From probability theory we know that anything within 2 standard deviations might well be due to random fluctuation.

But why 2 standard deviations? Why not 1 or 3 or 4.55?

It seems we can't say that there is a statistically significant difference between the treatment and the control. Does that mean there is no difference? Actually, no: if we had more data and the difference in means of 30.7 days with standard deviations of about 50 would persist, what sample size would be needed to find a statistically significant difference? The graph shows the standard deviations vs. the sample size (equal for both groups):



so we would need about 26 mice in each group.

Does this mean the treatment is really better than the control, we just didn't use enough mice in our study? Again, not necessarily, maybe the difference in means of 30.7 would decrease if we used more mice, and we would never pass the threshold of 2 standard deviations. We can't know that until we run a larger experiment. The above graph just gives us an idea how large such a new experiment should be.

1.1 R - Installation and Updating

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

1.1.1 RStudio

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

1.1.2 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 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)

1.2.1 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).

1.2.2 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)
```

1.2.3 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
```

```r
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:

```r
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:

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

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

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

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

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

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

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

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

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

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

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

```r
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!

1.2.4 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
```

1.2.5 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
```

1.2.5.1 Example

```
## $Gender
## [1] "M" "M" "F" "F" "F"
##
## $Age
## [1] 23 25 19 22 21 26 34
##
## $f
## function(x) x^2
## <environment: 0x00000148d359d5f8>
##
## [[4]]
## [[4]]$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.

1.2.5.2 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.

```r
dim(upr)
```

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

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

```r
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

```r
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".

1.2.6 Subsetting of Data Frames

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

```
students
```

```
##    Age GPA Gender
## 1   23 3.1 Female
## 2   20 3.2   Male
## 3   21 2.1 Female
## 4   20 2.1   Male
## 5   24 2.3 Female
## 6   18 2.9   Male
## 7   20 2.3   Male
## 8   22 3.9   Male
## 9   20 2.6   Male
## 10  21 3.2   Male
```

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] "Female" "Male"   "Female" "Male"   "Female"
```

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

```
##   Age GPA
## 1  23 3.1
## 2  20 3.2
## 3  21 2.1
## 4  20 2.1
## 5  24 2.3
```

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

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

```
students[1, ]
```

```
##   Age GPA Gender
## 1  23 3.1 Female
```

```
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   23 3.1
## 2   20 3.2
## 3   21 2.1
## 4   20 2.1
## 5   24 2.3
## 6   18 2.9
## 7   20 2.3
## 8   22 3.9
## 9   20 2.6
## 10  21 3.2
```

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

```
students$Gender
```

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

1.2.7 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]]
```

1.2.7.1 Example

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

```r
lst$Gender
```

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

1.2.8 Vector Arithmetic

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

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

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

```r
x^2
```

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

```r
log(x)
```

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

```r
sum(x)
```

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

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

```r
x+y
```

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

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

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

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

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

Let’s try something strange:

```r
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!

1.2.9 *apply*

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

Let's say we have the following matrix:

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

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]   20   23   21   23   27
## [2,]   28   23   29   23   29
## [3,]   27   25   27   20   20
## [4,]   28   24   30   28   28
## [5,]   22   24   26   30   29
```

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

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

```
## [1] 250
```

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

```
## [1] 248
```

. . .

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

```
## [1] 244
```

or much easier

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

```
##  [1] 250 248 261 257 249 259 248 249 244 244
```

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

1.2.9.1 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 
```

1.3 R Basics II - Writing Functions

1.3.1 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

1.3.2 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
```

1.3.3 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
```

```r
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.

1.3.4 Return Values

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

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

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

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

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

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

There is another way to specify what is returned:

```r
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:

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

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

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

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

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

```r
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
```

1.3.5 Basic Programmming Structures in R

R has all the standard programming structures:

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

1.3.5.1 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] -0.5824938
```

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

```
## [1] -0.5463664
```

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

```
## [1] -0.4655241
```

special R construct: *ifelse*

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

```
## [1] 10 10 10  3  8  1  5
```

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

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

1.3.5.2 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] 2
```

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

```r
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] 32
```

1.4 Random Numbers and Simulation

1.4.1 Random Numbers

Everything starts with generating X_1, X_2, .. iid U[0,1]. These are simply called random numbers. There are some ways to get these:

- random number tables
- numbers taken from things like the exact (computer) time
- quantum random number generators
- …

The R package *random* has the routine *randomNumbers* which gets random numbers from a web site which generates them based on (truly random) atmospheric phenomena.

```r
require(random)
randomNumbers(20, 0, 100)
```

```
##      V1 V2 V3 V4 V5
## [1,] 76 19 28 96 20
## [2,] 66 29 96 83 19
## [3,] 76 31  4  7 90
## [4,] 92 74 73 57 76
```

Most of the time we will use *pseudo-random numbers*, that is numbers that are not actually random but are indistinguishable from those. In R this is done with

```r
runif(5)
```

```
## [1] 0.8041365 0.2552747 0.5694376 0.2959418 0.3451290
```

```r
runif(5, 100, 300)
```

```
## [1] 168.1014 221.6839 213.3105 120.4541 165.1059
```

1.4.2 Standard Probability Distributions

Not surprisingly many standard distributions are part of base R. For each the format is

- dname = density
- pname = cumulative distribution function
- rname = random generation
- qname = quantile function

Note we will use the term *density* for both discrete and continuous random variable.

1.4.2.1 Example Poisson distribution

We have $X \sim \text{Pois}(\lambda)$ if

$$P(X=x)=\frac{\lambda^x}{x!}e^{-\lambda},\ x=0,1,...$$

```r
# density
dpois(c(0, 8, 12, 20), lambda=10)
```

```
## [1] 4.539993e-05 1.125990e-01 9.478033e-02 1.866081e-03
```

```r
10^c(0, 8, 12, 20)/factorial(c(0, 8, 12, 20))*exp(-10)
```

```
## [1] 4.539993e-05 1.125990e-01 9.478033e-02 1.866081e-03
```

```r
# cumulative distribution function
ppois(c(0, 8, 12, 20), 10)
```

```
## [1] 4.539993e-05 3.328197e-01 7.915565e-01 9.984117e-01
```

```r
# random generation
rpois(5, 10)
```

```
## [1]  9 12 10 12  8
```

```r
# quantiles
qpois(1:4/5, 10)
```

```
## [1]  7  9 11 13
```

Here is a list of the distributions included with base R:

- beta distribution: dbeta.
- binomial (including Bernoulli) distribution: dbinom.
- Cauchy distribution: dcauchy.
- chi-squared distribution: dchisq.
- exponential distribution: dexp.
- F distribution: df.
- gamma distribution: dgamma.
- geometric distribution: dgeom.
- hypergeometric distribution: dhyper.
- log-normal distribution: dlnorm.
- multinomial distribution: dmultinom.
- negative binomial distribution: dnbinom.
- normal distribution: dnorm.
- Poisson distribution: dpois.
- Student's t distribution: dt.
- uniform distribution: dunif.
- Weibull distribution: dweibull.

With some of these a bit of caution is needed. For example, the usual textbook definition of the geometric random variable is the number of tries in a sequence of independent Bernoulli trials until a success. This means that the density is defined as

$$P(X=x)=p(1-p)^{x-1},\ x=1,2,..$$

R however defines it as the number of failures until the first success, and so it uses

$$P(X^*=x)=\text{dgeom}(x,p)=p(1-p)^x,\ x=0,1,2,..$$

Of course this is easy to fix. If you want to generate the "usual" geometric do

```r
x <- rgeom(10, 0.4) + 1
x
```

```
##  [1] 4 2 4 4 1 3 8 1 1 3
```

if you want to find the probabilities or cdf:

```r
round(dgeom(x-1, 0.4), 4)
```

```
##  [1] 0.0864 0.2400 0.0864 0.0864 0.4000 0.1440 0.0112 0.4000 0.4000 0.1440
```

```r
round(0.4*(1-0.4)^(x-1), 4)
```

```
##  [1] 0.0864 0.2400 0.0864 0.0864 0.4000 0.1440 0.0112 0.4000 0.4000 0.1440
```

Another example is the Gamma random variable. Here most textbooks use the definition

$$f(x;\alpha,\beta)=\frac{1}{\Gamma(\alpha)\beta^\alpha}x^{\alpha-1}e^{-x/\beta}$$

but R uses

$$f^*(x;\alpha,\beta)=\frac{\beta^\alpha}{\Gamma(\alpha)}x^{\alpha-1}e^{-\beta x}$$

```r
dgamma(1.2, 0.5, 2)
```

```
## [1] 0.06607584
```

```r
2^0.5/gamma(0.5)*1.2^(0.5-1)*exp(-2*1.2)
```

```
## [1] 0.06607584
```

Again, it is easy to *re-parametrize*:

```r
dgamma(1.2, 0.5, 1/(1/2))
```

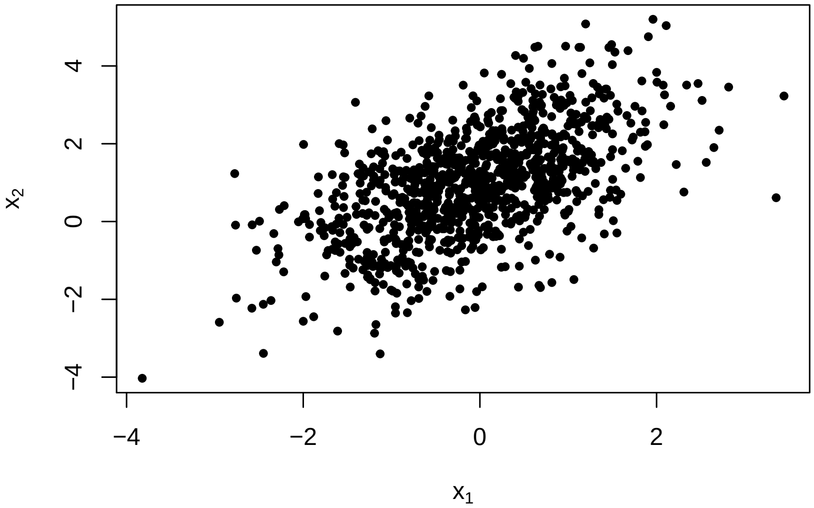
```
## [1] 0.06607584
```

1.4.3 Other Variates

if you need to generate random variates from a distribution that is not part of base R you should first try to find a package that includes it.

1.4.3.1 Example multivariate normal there are actually several packages, the most commonly used one is *mvtnorm*

```
library(mvtnorm)
x <- rmvnorm(1000,
             mean = c(0, 1),
             sigma = matrix(c(1, 0.8, 0.8, 2), 2, 2))
plot(x,
     pch=20,
     xlab = expression(x[1]),
     ylab = expression(x[2]))
```



sigma is the variance-covariance matrix, so in the above we have

$$\rho = Cor(X,Y) = \\ \frac{Cov(X,Y)}{\sqrt{Var(X)\,Var(Y)}} = \\ \frac{0.8}{\sqrt{1*2}} = 0.566$$

```
round(c(var(x[, 1]),
        var(x[, 2]),
        cor(x[, 1], x[, 2])), 3)
```

```
## [1] 0.952 1.921 0.568
```

1.4.4 Simulation

In a *simulation* we attempt to generate data just like what we might see in a real-live experiment, except that we control all the details. The we carry out some calculations on that artificial data, and we repeat this many times. Here are some examples:

1.4.4.1 Example When rolling a fair die 5 times, what is the probability of no sixes? Of no more than one six?

```r
B <- 10000 # number of simulation runs
num.sixes <- rep(0, B) # to store results
for(i in 1:B) {
  x <- sample(1:6, size=5, replace=TRUE) # roll 5 dice
  num.sixes[i] <- length(x[x==6]) # how many sixes?
}
# Probability of no sixes
length(num.sixes[num.sixes==0])/B
```

```
## [1] 0.395
```

```r
# Probability of no more than one sixes
length(num.sixes[num.sixes<=1])/B
```

```
## [1] 0.8008
```

Of course one can do this also analytically:

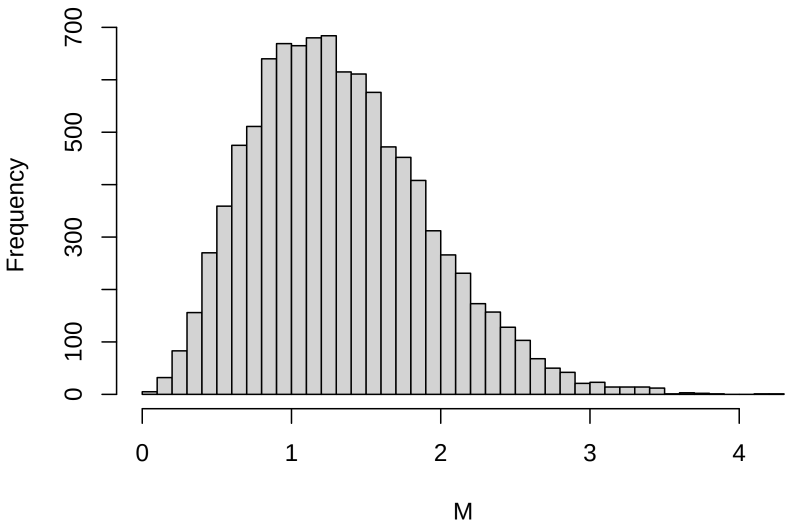
$$\begin{aligned}
&P(\text{no sixes}) = P(\text{no six on any die}) = \\
&P(\text{no six on first die } \cap \text{ .. } \cap \text{ no six on fifth die}) = \\
&\prod_{i=1}^{5} P(\text{no six on } i^{th}\text{die}) = (\frac{5}{6})^5 = 0.402
\end{aligned}$$

but already the second one is a bit harder to do analytically but not via simulation.

One issue we have with a simulation is the *simulation error*, namely that the simulation will always yield a slightly different answer.

1.4.4.2 Example Say we have $X, Y, Z \sim N(0,1)$ and set $M = \max\{|X|, |Y|, |Z|\}$. What is the mean and standard deviation of M?

```r
B <- 10000
x <- matrix(abs(rnorm(3*B)), ncol=3)
M <- apply(x, 1, max)
hist(M, 50, main="")
```



```
round(c(mean(M), sd(M)), 3)
```

```
## [1] 1.331 0.587
```

1.4.4.3 Example Symmetric Random Walk in R Let $P(Z_i = -1) = P(Z_i = 1) = \frac{1}{2}$ and $X_n = \sum_{i=1}^{n} Z_i$. Let A>0 some integer. Let's write a routine that finds the median number of steps the walk takes until it hits either -A or A.

One issue with simulations of *stochastic processes* is that in general they are very slow. Here I will use a little trick: I will generate part of the process, and then check whether the event of interest has already happened.

```
first.hit <- function(A) {
  B <- 10000
  num.steps <- rep(0, B)
  for(i in 1:B) {
    x <- 0
    k <- 0
    repeat {
      z <- sample(c(-1, 1), size=1000, replace=TRUE)
      x <- x + cumsum(z)
      if(max(abs(x))>=A) break
      x <- x[1000]
      k <- k+1000
    }
    k <- k+seq_along(x)[abs(x)>=A][1]
    num.steps[i] <- k
  }
  median(num.steps)
}
first.hit(100)
```

```
## [1] 7480
```

1.4.4.4 Example The following you find in any basic stats course: A $100(1-\alpha)\%$ confidence interval for the success probability in a sequence of n Bernoulli trials is given by

$$\hat{p} \pm z_{\alpha/2}\sqrt{\hat{p}(1-\hat{p})/n}$$

where $\hat{p}$ is the proportion of successes. This method is supposed to work if n is at least 50.

Let's do a simulation to test this method.

```
ci.prop.sim <- function(p, n, conf.level=95, B=1e4) {
  z <- qnorm(1-(1-conf.level/100)/2)
  bad <- 0
  for(i in 1:B) {
    x <- sample(0:1, size=n, replace = TRUE, prob=c(1-p, p))
    phat <- sum(x)/n
    if(phat - z*sqrt(phat*(1-phat)/n)>p) bad<-bad+1
    if(phat + z*sqrt(phat*(1-phat)/n)<p) bad<-bad+1
  }
  bad/B
}
```

```
ci.prop.sim(0.5, 100)
```

```
## [1] 0.0548
```

and that is not so bad.

But

```
ci.prop.sim(0.1, 50)
```

```
## [1] 0.1224
```

and that is very bad indeed!

Soon we will consider a method that is guaranteed to give intervals with correct coverage, no matter what p and n are.

1.4.4.5 Example: Simultaneous Inference There is a famous (infamous?) case of three psychiatrists who studied a sample of schizophrenic persons and a sample of non schizophrenic persons. They measured 77 variables for each subject - religion, family background, childhood experiences etc. Their goal was to discover what distinguishes persons who later become schizophrenic. Using their data they ran 77 hypothesis tests of the significance of the differences between the two groups of subjects, and found 2 significant at the 2% level.They immediately published their findings.

What's wrong here? Remember, if you run a hypothesis test at the 2% level you expect to reject the null hypothesis of no relationship 2% of the time, but 2% of 77 is about 1 or 2, so just by random fluctuations they could (should?) have rejected that many null hypotheses! This is not to say that the variables they found to be different between the two groups were not really different, only that their method did not proof that.

In its general form this is known as the problem of simultaneous inference and is one of the most difficult issues in Statistics today. One general solution of used is called *Bonferroni's method*. The idea is the following:

Let's assume we carry out k hypothesis tests. All tests are done at α significance level and all the tests are all independent. Then the probability that at least one test rejects the null hypothesis although all null are true is given by

$$
\begin{aligned}
&\alpha^* = P(\text{at least one null rejected} \mid \text{all null true}) = \\
&1 - P(\text{none of the nulls rejected} \mid \text{all null true}) = \\
&1 - \prod_{i=1}^{k} P(\text{ith null is not rejected} \mid \text{ith null true}) = \\
&1 - \prod_{i=1}^{k} \left[1 - P(\text{ith null is rejected} \mid \text{ith null true})\right] = \\
&1 - [1-\alpha]^k = \\
&1 - \left[1 - k\alpha + \binom{k}{2}\alpha^2 - +..\right] \approx k\alpha
\end{aligned}
$$

so if each individual test is done with α/k, the *family-wise* error rate is the desired one.

Let's do a simulation to see how that would work in the case of our psychiatrists experiments. There many details we don't know, so we have to make them up a bit:

```
sim.shiz <- function(m, n=50, B=1e3) {
  counter <- matrix(0, B, 2)
  for(i in 1:B) {
    for(j in 1:77) {
      pval <- t.test(rnorm(n), rnorm(n))$p.value
      if(pval<0.02) counter[i, 1]<-1
      if(pval<0.05/m) counter[i, 2]<-1
    }
  }
  apply(counter, 2, sum)/B
}
sim.shiz(77)
```

```
## [1] 0.795 0.042
```

This works fine here. The main problem in real life is that it is rarely true that these test are independent, and then all we can say is that the needed α is between α/k and α.

1.5 Graphics with ggplot2

A large part of this chapter is taken from various works of Hadley Wickham. Among others The layered grammar of graphics and R for Data Science.

1.5.1 Why ggplot2?

Advantages of ggplot2

- consistent underlying grammar of graphics (Wilkinson, 2005)
- plot specification at a high level of abstraction
- very flexible
- theme system for polishing plot appearance
- mature and complete graphics system
- many users, active mailing list

but really, they are just so much nicer than base R graphs!

1.5.2 Grammar of Graphics

In 2005 Wilkinson, Anand, and Grossman published the book "The Grammar of Graphics". In it they laid out a systematic way to describe any graph in terms of basic building blocks. ggplot2 is an implementation of their ideas.

The use of the word *grammar* seems a bit strange here. The general dictionary meaning of the word grammar is:

the fundamental principles or rules of an art or science

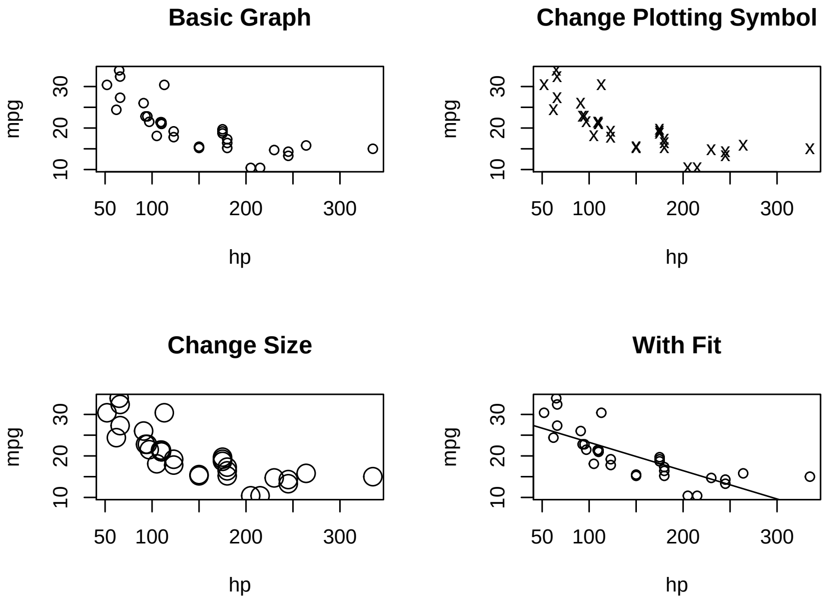
so it is not only about language.

As our running example we will use the *mtcars* data set. It is part of base R and has information on 32 cars:

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

Say we want to study the relationship of hp and mpg. So we have two quantitative variables, and therefore the obvious thing to do is a scatterplot. But there are a number of different ways we can do this:

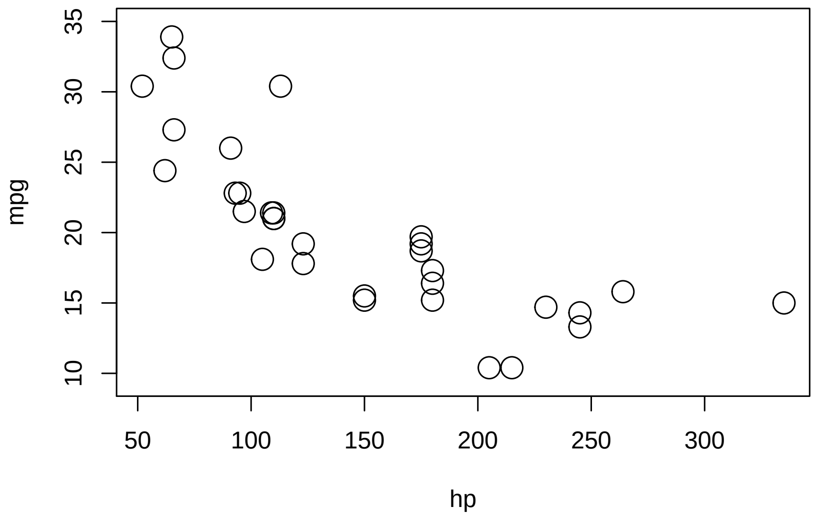
```
attach(mtcars)
par(mfrow=c(2, 2))
plot(hp, mpg, main="Basic Graph")
plot(hp, mpg, pch="x", main="Change Plotting Symbol")
plot(hp, mpg, cex=2, main="Change Size")
plot(hp, mpg, main="With Fit");abline(lm(mpg~hp))
```



The basic idea of the grammar of graphs is to separate out the parts of the graphs: there is the basic layout, there is the data that goes into it, there is the way in which the data is displayed. Finally there are annotations, here the titles, and other things added, such as a fitted line. In ggplot2 you can always change one of these without worrying how that change effects any of the others.

Take the graph on the lower left. Here I made the plotting symbol bigger (with cex=2). But now the graph doesn't look nice any more, the first and the last circle don't fit into the graph. The only way to fix this is to start all over again, by making the margins bigger:

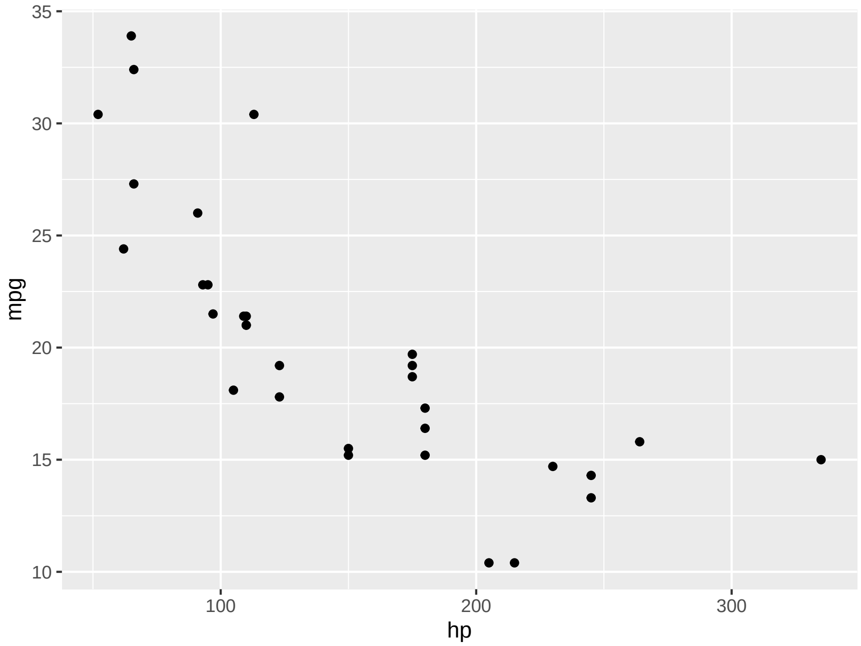
```
plot(hp, mpg, cex=2, ylim=range(mpg)+c(-1, 1))
```



and that is a bit of work because I have to figure out how to change the margins. In ggplot2 that sort of thing is taken care of automatically!

Let's start by recreating the first graph above.

```
ggplot(mtcars, aes(hp, mpg)) +
  geom_point()
```

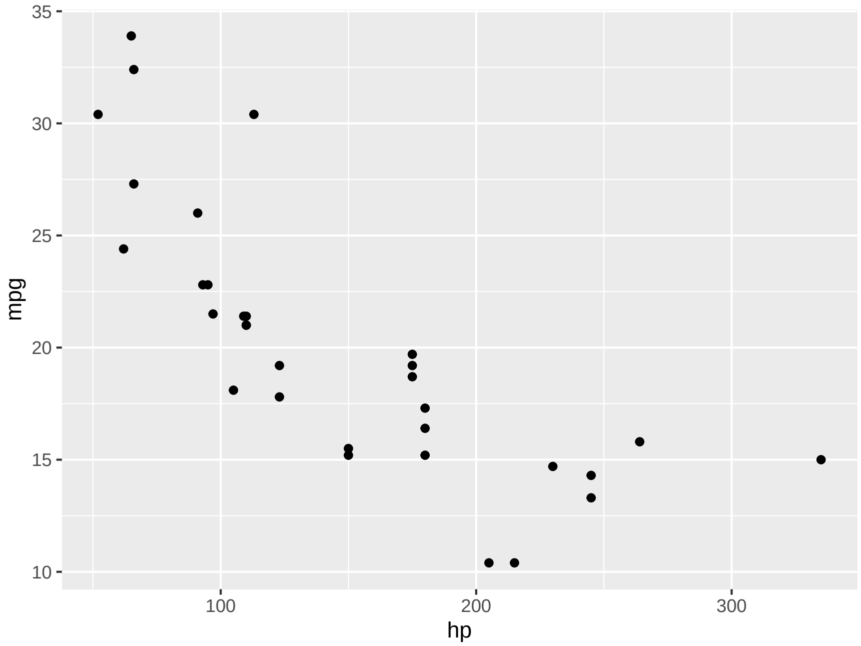


this has the following logic:

- *ggplot* sets up the graph
- it's first argument is the data set (which has to be a dataframe)
- *aes* is the *aestetic mapping*. It connects the data to the graph by specifying which variables go where
- *geom* is the geometric object (circle, square, line) to be used in the graph

Note ggplot2 also has the *qplot* command. This stands for *qick plot*

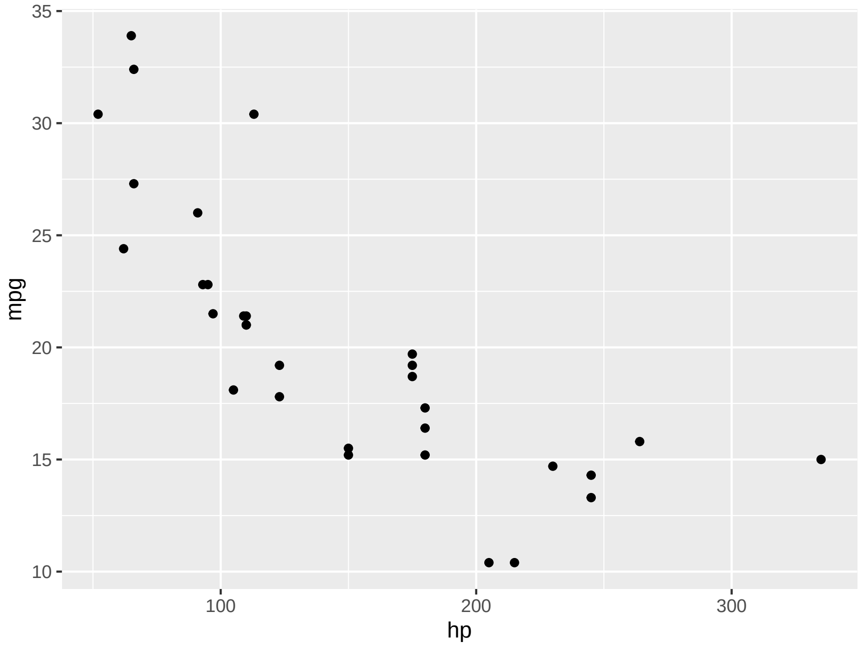
```
qplot(hp, mpg, data=mtcars)
```



This seems much easier at first (and it is) but the qplot command is also quite limited. Very quickly you want to do things that aren't possible with qplot, and so I won't discuss it further here.

Note consider the following variation:

```
ggplot(mtcars) +
  geom_point(aes(hp, mpg))
```



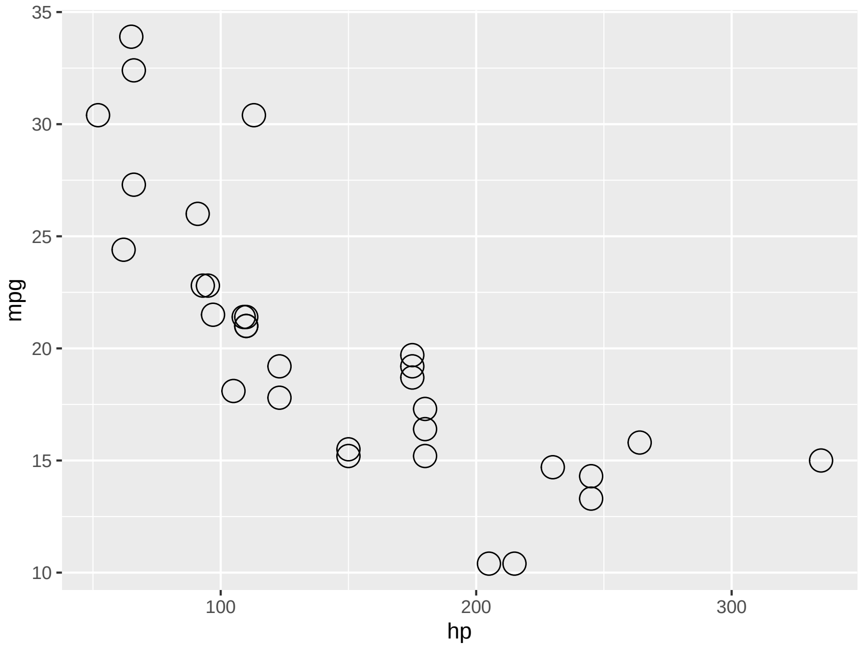
again it seems to do the same thing, but there is a big difference:

- if aes(x, y) is part of ggplot, it applies to all the geom's that come later (unless a different one is specified)

- an aes(x, y) as part of a geom applies only to it.

How about the problem with the graph above, where we had to increase the y margin?

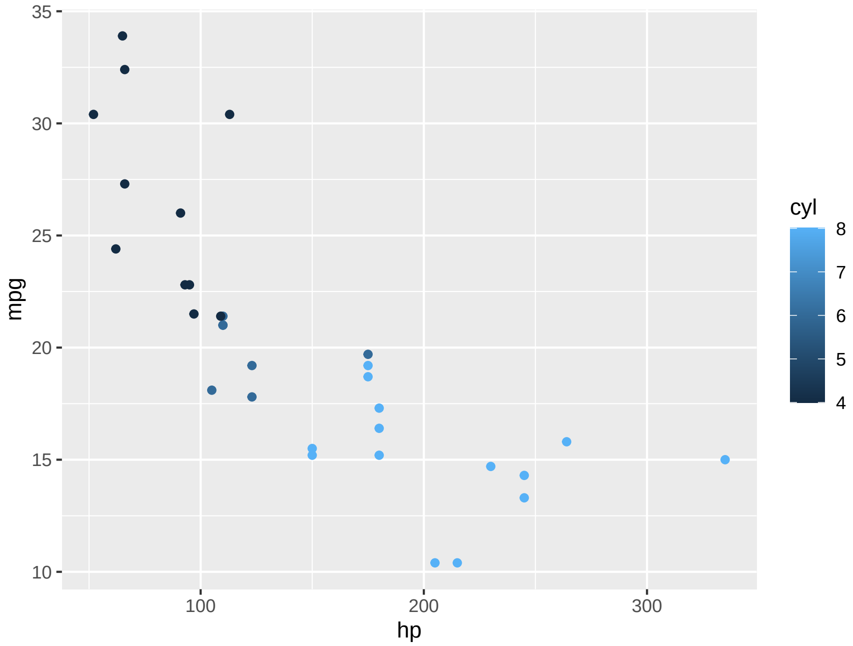
```r
ggplot(mtcars, aes(hp, mpg)) +
  geom_point(shape=1, size=5)
```



so we see that here this is done automatically.

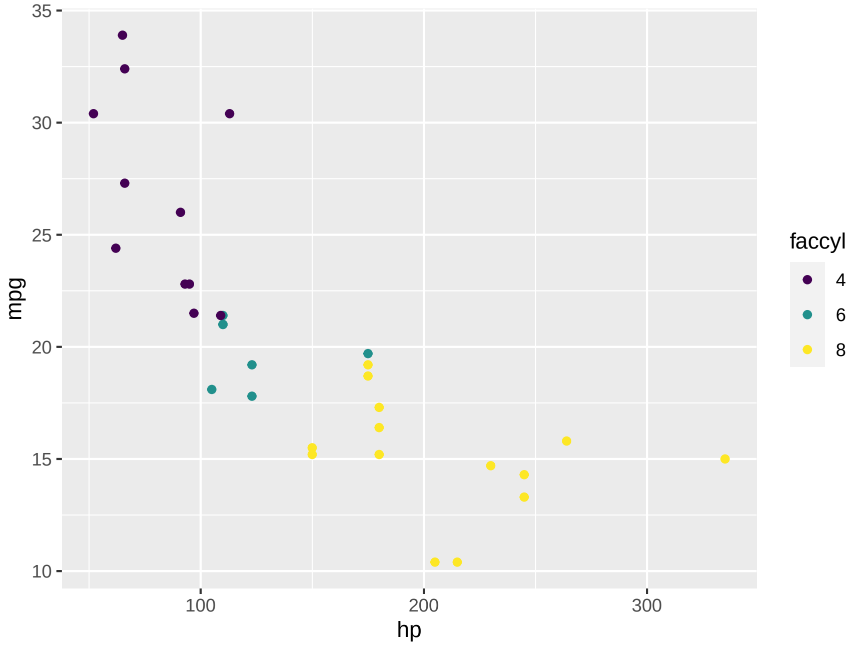
Let's say we want to identify the cars by the number of cylinders:

```r
ggplot(mtcars, aes(hp, mpg, color=cyl)) +
  geom_point()
```



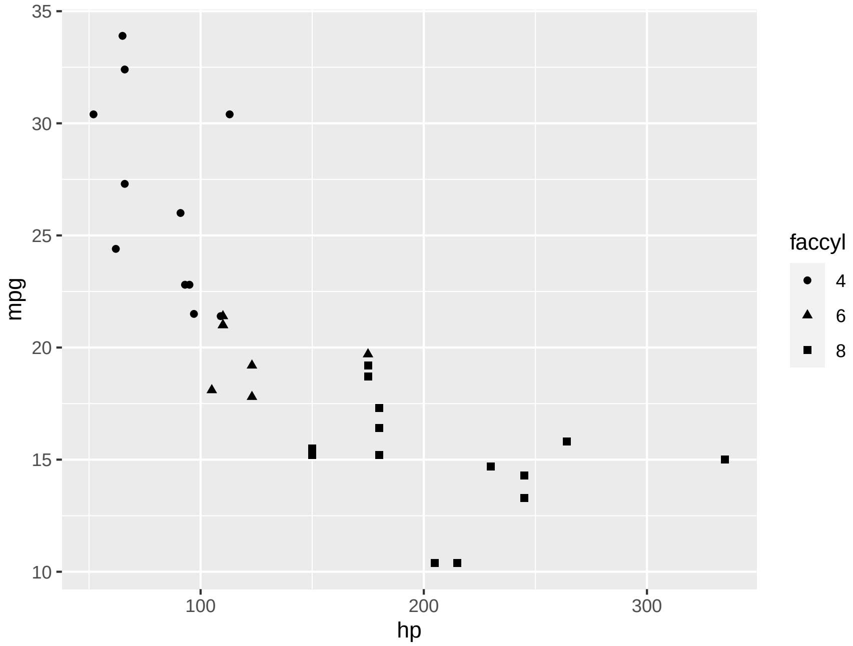
Notice that the legend is a continuous color scale. This is because the variable cyl has values 4, 6, and 8, and so is identified by R as a numeric variable. In reality it is categorical (ever seen a car with 1.7 cylinders?), and so we should change that:

```
mtcars$faccyl <- factor(cyl,
                        levels = c(4, 6, 8),
                        ordered = TRUE)
ggplot(mtcars, aes(hp, mpg, color=faccyl)) +
  geom_point()
```



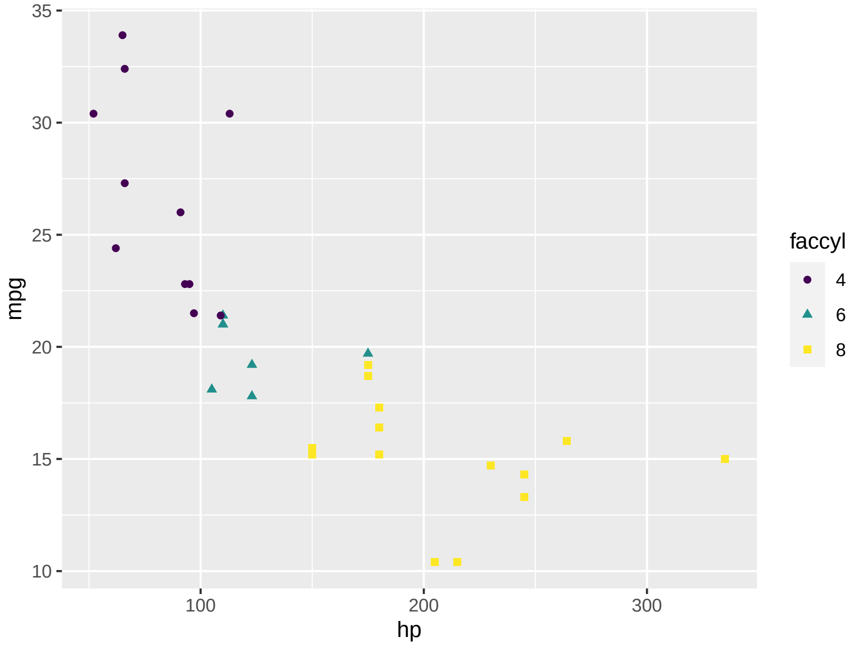
we can also change the shape of the plotting symbols:

```
ggplot(mtcars, aes(hp, mpg, shape=faccyl)) +
  geom_point()
```



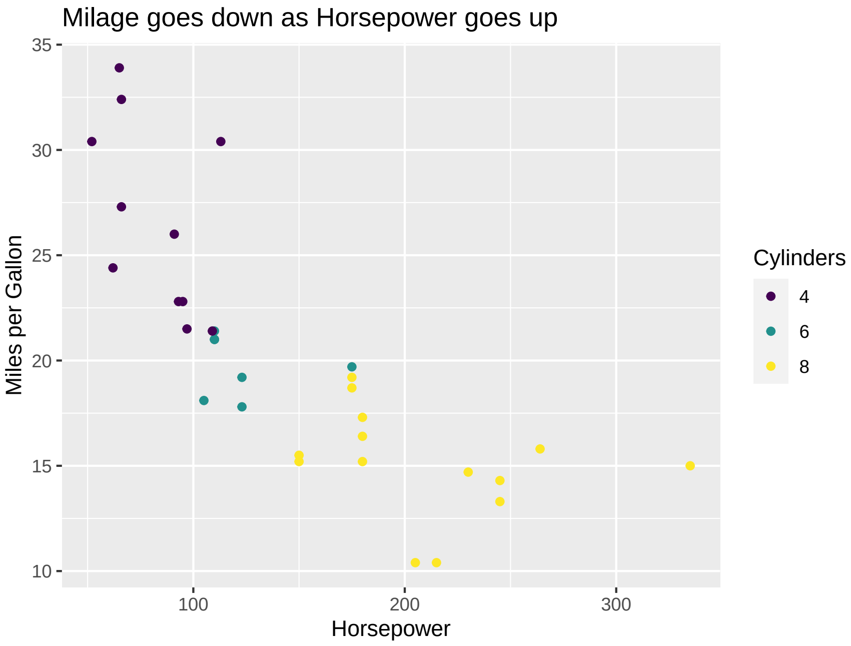
or both:

```
ggplot(mtcars, aes(hp, mpg, shape=faccyl, color=faccyl)) +
  geom_point()
```



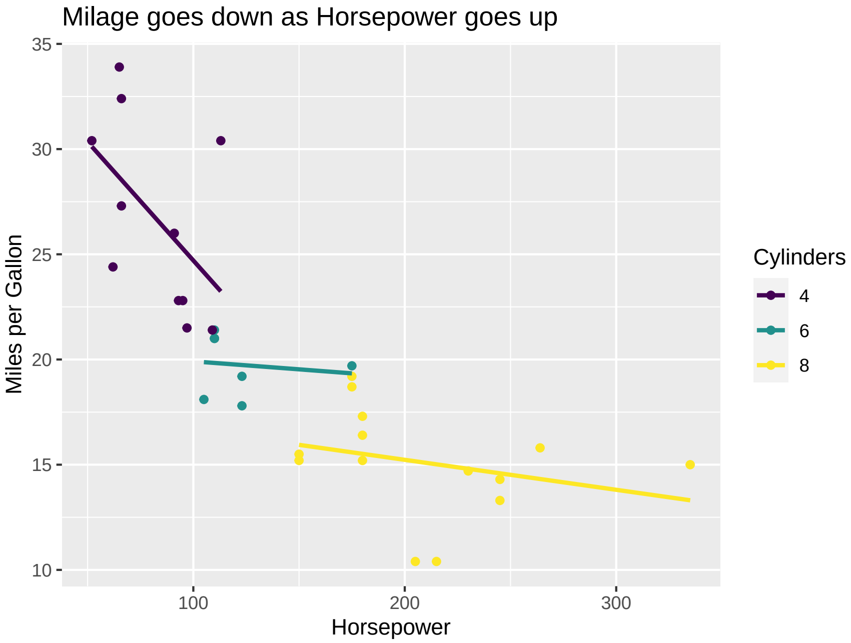
let's pretty up the graph a bit with some labels and a title. We will be playing around with this graph for a while, so I will save some intermediate versions:

```
plt1 <- ggplot(mtcars, aes(hp, mpg, color=faccyl)) +
  geom_point()
plt2 <- plt1 +
  labs(x = "Horsepower",
       y = "Miles per Gallon",
       color = "Cylinders") +
  labs(title = "Milage goes down as Horsepower goes up")
plt2
```



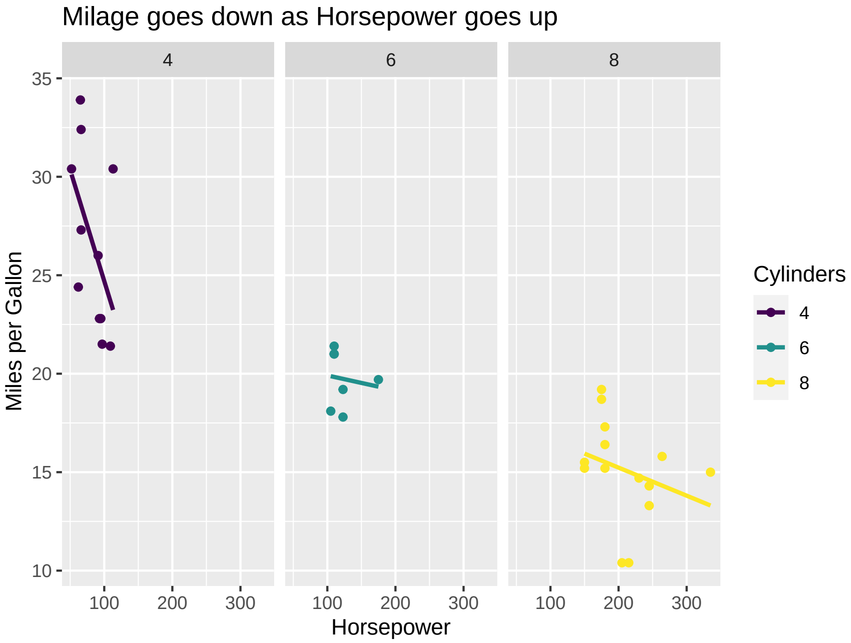
Say we want to add the least squares regression lines for cars with the same number of cylinders:

```
plt3 <- plt2 +
  geom_smooth(method = "lm", se = FALSE)
plt3
```



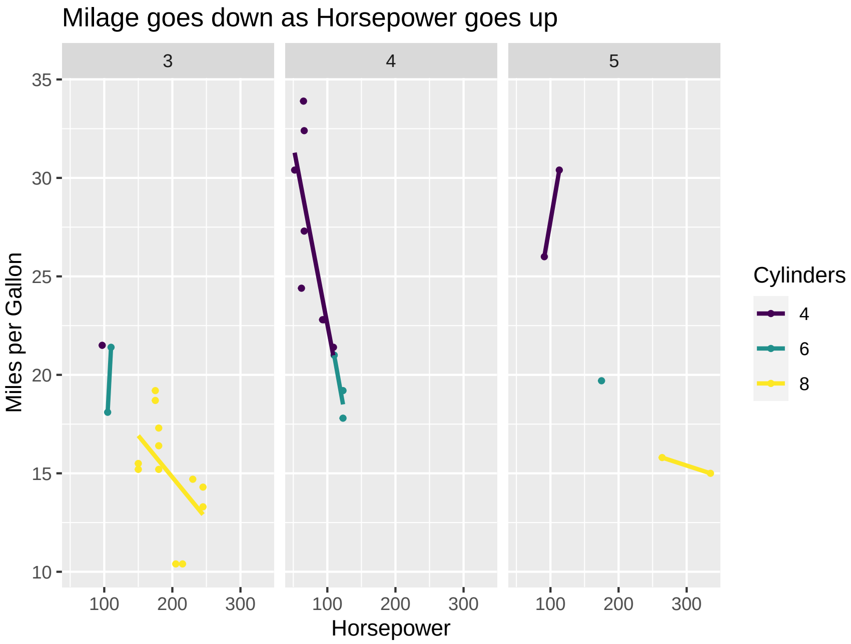
There is another way to include a categorical variable in a scatterplot. The idea is to do several graphs, one for each value of the categorical variable. These are called *facets*:

```
plt3 +
  facet_wrap(~cyl)
```



The use of facets also allows us to include two categorical variables:

```
mtcars$facgear <-
  factor(gear, levels = 3:5, ordered = TRUE)
plt4 <- ggplot(aes(hp, mpg, color=faccyl),
               data = mtcars) +
           geom_point(size = 1)
plt4 <- plt4 +
  facet_wrap(~facgear)
plt4 <- plt4 +
  labs(x = "Horsepower",
       y = "Miles per Gallon",
       color = "Cylinders") +
  labs(title = "Milage goes down as Horsepower goes up")
plt4 <- plt4 +
  geom_smooth(method = "lm", se = FALSE)
plt4
```

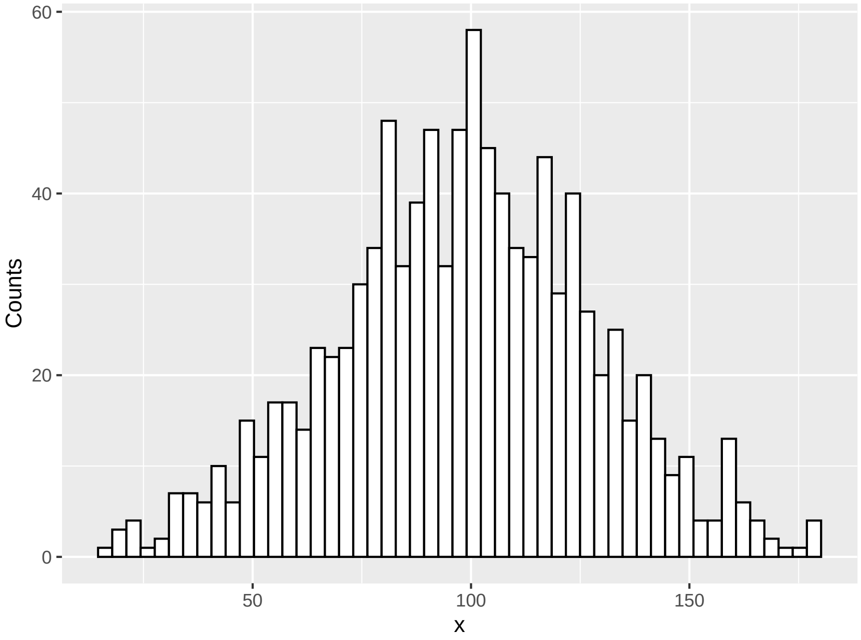


This is almost a bit to much, with just 32 data points there is not really enough for such a split.

Let's see how to use ggplot do a number of basic graphs:

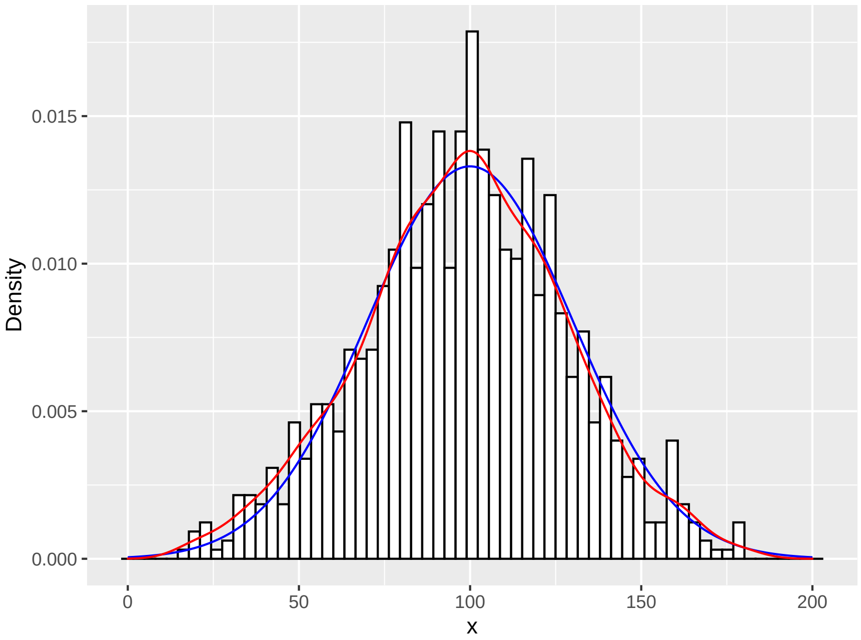
1.5.3 Histograms

```r
x <- rnorm(1000, 100, 30)
df3 <- data.frame(x = x)
bw <- diff(range(x))/50 # use about 50 bins
ggplot(df3, aes(x)) +
  geom_histogram(color = "black",
                 fill = "white",
                 binwidth = bw) +
  labs(x = "x", y = "Counts")
```



Often we do histograms scaled to integrate to one. Then we can add the theoretical density and/or a nonparametric density estimate:

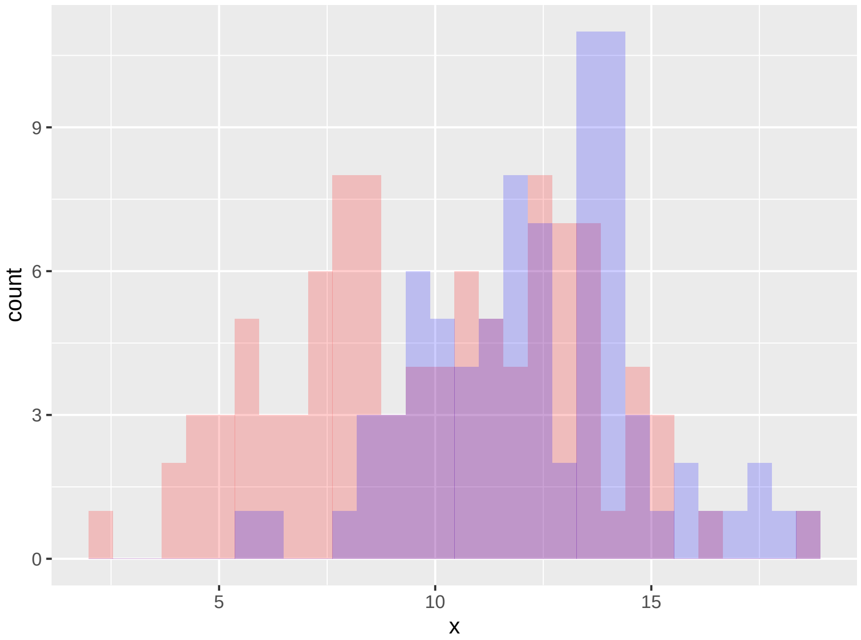
```r
x <- seq(0, 200, length=250)
df4 <- data.frame(x=x, y=dnorm(x, 100, 30))
ggplot(df3, aes(x)) +
  geom_histogram(aes(y = ..density..),
        color = "black",
        fill = "white",
        binwidth = bw) +
  labs(x = "x", y = "Density") +
  geom_line(data = df4, aes(x, y),
            colour = "blue") +
  geom_density(color = "red")
```



Notice the red line on the bottom. This should not be there but seems almost impossible to get rid of!

Here is another interesting case: say we have two data sets and we wish to draw the two histograms, one overlaid on the other:

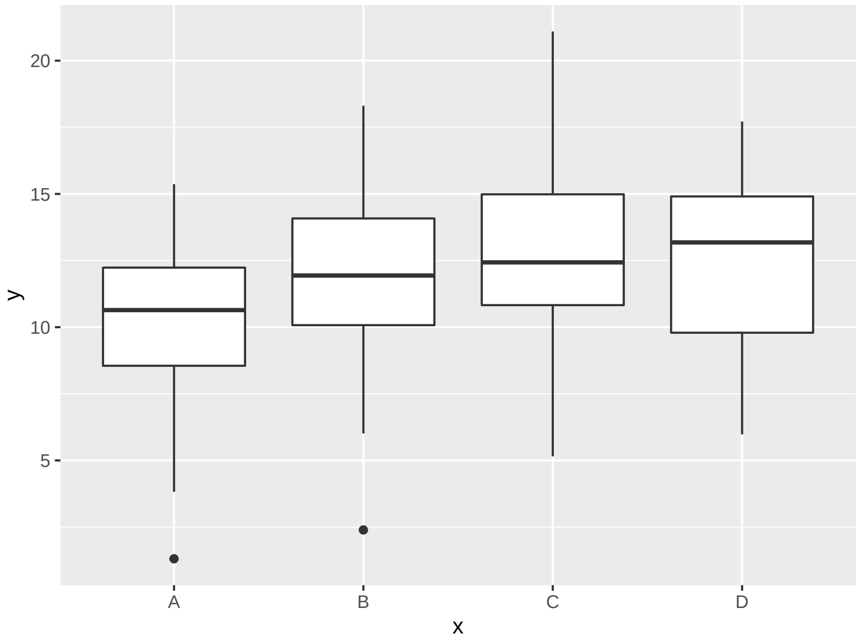
```
df5 <- data.frame(
  x = c(rnorm(100, 10, 3), rnorm(80, 12, 3)),
  y = c(rep(1, 100), rep(2, 80)))
ggplot(df5, aes(x=x)) +
    geom_histogram(data = subset(df5, y == 1),
        fill = "red", alpha = 0.2) +
    geom_histogram(data = subset(df5, y == 2),
        fill = "blue", alpha = 0.2)
```



Notice the use of alpha. In general this “lightens” the color so we can see “behind”.

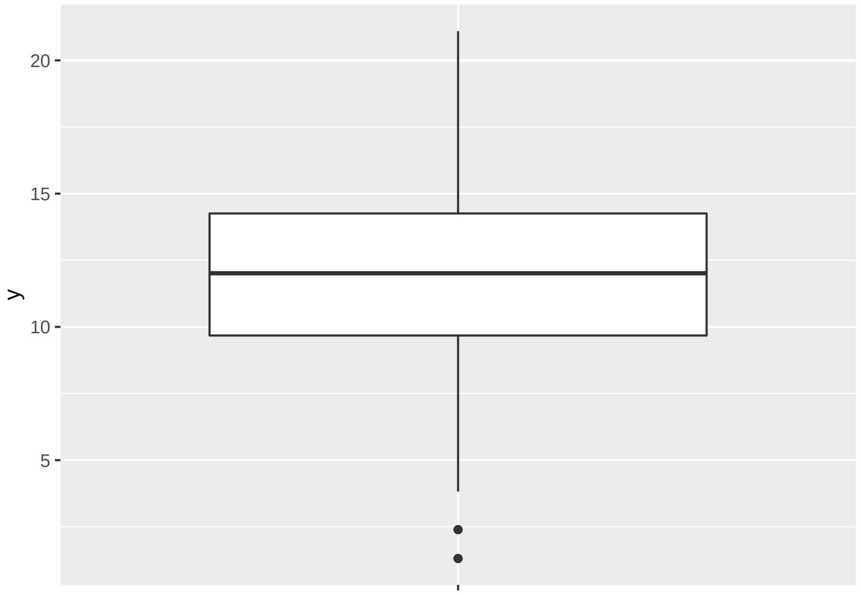
1.5.4 Boxplots

```
y <- rnorm(120, 10, 3)
x <- rep(LETTERS[1:4], each=30)
y[x=="B"] <- y[x=="B"] + rnorm(30, 1)
y[x=="C"] <- y[x=="C"] + rnorm(30, 2)
y[x=="D"] <- y[x=="D"] + rnorm(30, 3)
df6 <- data.frame(x=x, y=y)
ggplot(df6, aes(x, y)) +
  geom_boxplot()
```



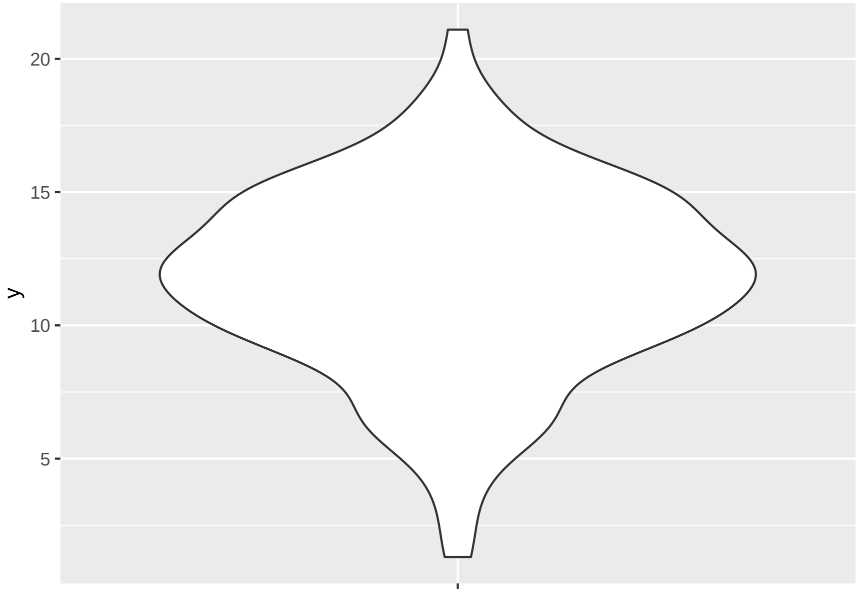
strangely enough doing a boxplot without groups takes a bit of a hack. We have to "invent" a categorical variable:

```
ggplot(df6, aes(x="", y)) +
  geom_boxplot() +
  xlab("")
```



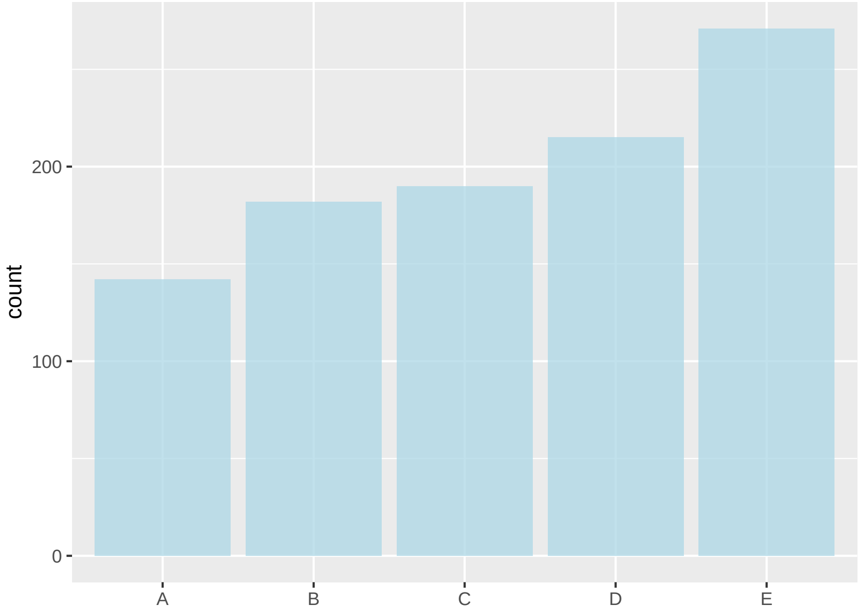
There is a modern version of this graph called a violin plot:

```
ggplot(df6, aes(x="", y)) +
  geom_violin() +
  xlab("")
```



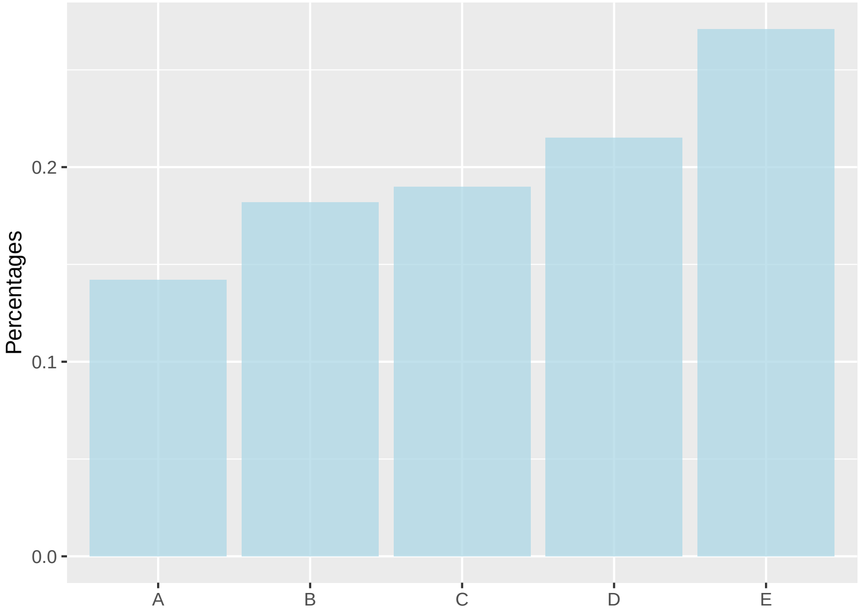
1.5.5 Barcharts

```
x <- sample(LETTERS[1:5],
            size = 1000,
            replace = TRUE,
            prob = 6:10)
df7 <- data.frame(x=x)
ggplot(df7, aes(x)) +
  geom_bar(alpha=0.75, fill="lightblue") +
  xlab("")
```



Say we want to draw the graph based on percentages. Of course we could just calculate them and then do the graph. Here is another way:

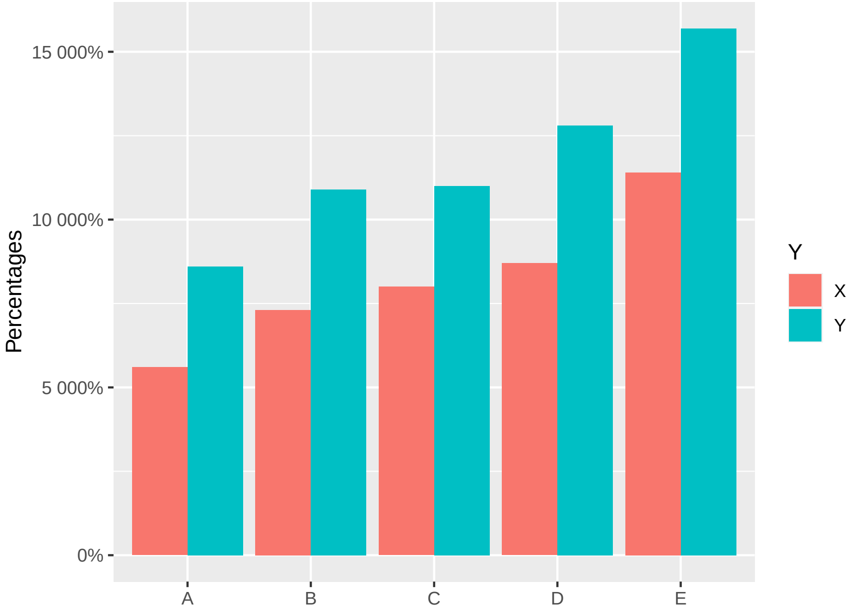
```
ggplot(df7, aes(x=x)) +
  geom_bar(aes(y=(..count..)/sum(..count..)),
      alpha = 0.75,
      fill = "lightblue") +
  labs(x="", y="Percentages")
```



Notice how this works: in geom_bar we use a new aes, but the values in it are calculated from the old data frame.

Finally an example of a contingency table:

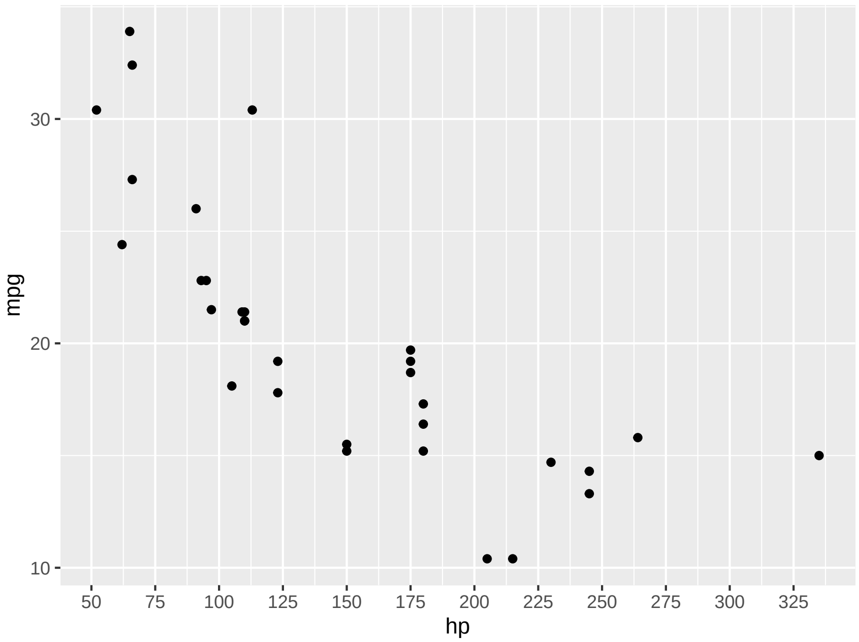
```
df7$y <- sample(c("X", "Y"),
                size = 1000,
                replace = TRUE,
                prob = 2:3)
ggplot(df7, aes(x=x, fill = y)) +
  geom_bar(position = "dodge") +
    scale_y_continuous(labels=scales::percent) +
    labs(x="", y="Percentages", fill="Y")
```



1.5.6 Axis Ticks and Legend Keys

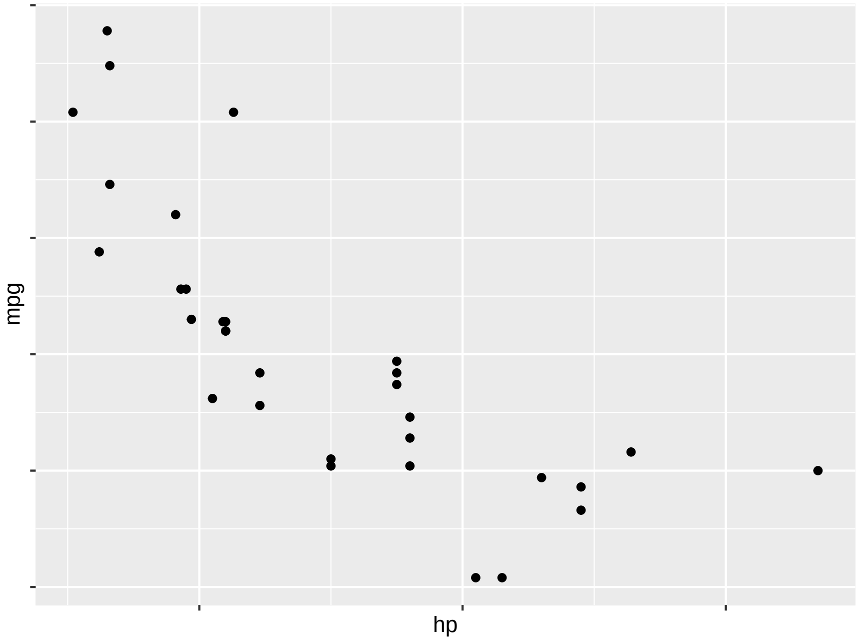
Let's return to the basic plot of mpg by hp. Let's say we want to change the axis tick marks:

```
ggplot(mtcars, aes(hp, mpg)) +
  geom_point() +
  scale_x_continuous(breaks = seq(50, 350, by=25)) +
  scale_y_continuous(breaks = seq(0, 50, by=10))
```



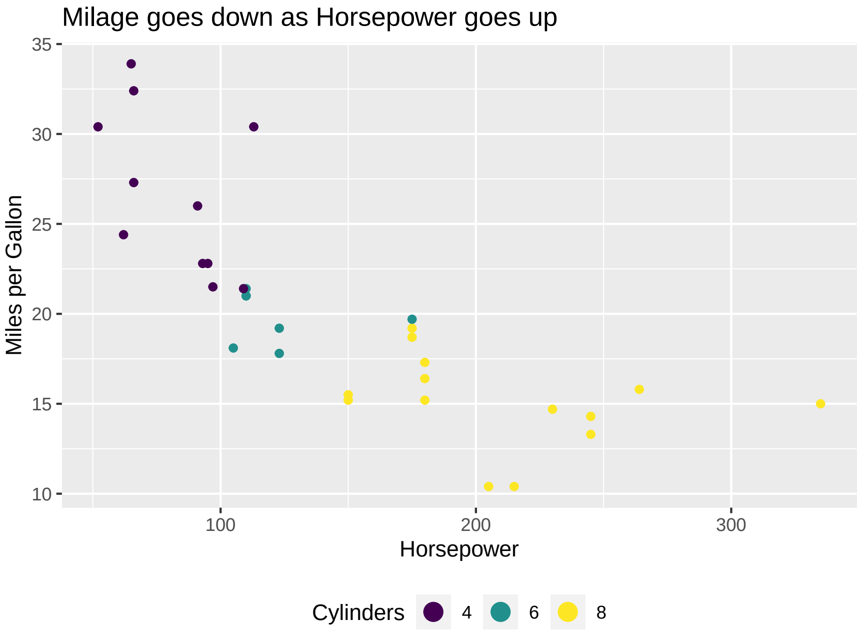
sometimes we want to do graphs without any tick labels. This is useful for example for maps and also for confidential data, so the viewer sees the relationship but can't tell the sizes:

```r
ggplot(mtcars, aes(hp, mpg)) +
  geom_point() +
  scale_x_continuous(labels = NULL) +
  scale_y_continuous(labels = NULL)
```



By default ggplot2 draws the legends on the right. We can however change that. We can also change the appearance of the legend. Recall that the basic graph is in *plt2*. Then

```
plt2 +
  theme(legend.position = "bottom") +
  guides(color=guide_legend(nrow = 1,
                            override.aes = list(size=4)))
```



1.5.7 Saving the graph

It is very easy to save a ggplot2 graph. Simply run

```
ggsave("myplot.density")
```

it will save the last graph to disc.

One issue is figure sizing. You need to do this so that a graph looks "good". Unfortunately this depends on where it ends up. A graph that looks good on a webpage might look ugly in a density. So it is hard to give any general guidelines.

If you use R markdown, a good place to start is with the chunk arguments fig.with=6 and out.width="70%". In fact on top of every R markdown file I have a chunk with

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

so that automatically every graph is sized that way. I also change the default behavior of the chunks to something I like better!

1.6 Important R 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.

1.6.1 The first functions to learn

? str

1.6.2 Important operators and assignment

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

1.6.3 Comparison

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

1.6.4 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)

1.6.5 Matrix algebra

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

1.6.6 Workspace

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

1.6.7 Help

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

1.6.8 Debugging

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

1.6.9 Output

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

1.6.10 Reading and writing data

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

load, save
library

1.6.11 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 Probability in Statistics

2.1 Probability Theory

2.1.1 Introduction

For a detailed discussion of Probability Theory go http://academic.uprm.edu/wrolke/esma6600

We are not going to do a detailed review of the theory of probability. Instead we are going to do through several examples that include the kinds of calculations you should know how to do.

2.1.1.1 Example (2.1.1) say we have a random variable X with density $f(x) = c/x^{a-1}, x > 1, a > 0$

a. find c

$$1 = \int_{-\infty}^{\infty} f(x)dx = \int_{1}^{\infty} \frac{c}{x^{a+1}}dx = \frac{-c}{ax^{a}}|_{1}^{\infty} = \frac{c}{a}$$

so$c = \alpha$.

b) Find E[X] and var(X)

$$
\begin{aligned}
&E[X^k] = \int_{-\infty}^{\infty} x^k f(x)dx = \\
&\int_1^{\infty} x^k \frac{a}{x^{a+1}} dx = \\
&\int_1^{\infty} \frac{a}{x^{a-k+1}} dx = \\
&\frac{-a}{(a-k)x^{a-k}}|_1^{\infty} = \\
&\frac{a}{k-a}
\end{aligned}
$$

if $a > k$ and ∞ otherwise.

So

$$
\begin{aligned}
&E[X] = \begin{cases} \frac{a}{k-a} & a > 1 \\ \infty & a \leq 1 \end{cases} \\
&var(X) = E[X^2] - (E[X])^2 = \\
&\begin{cases} \frac{a}{2-a} - \left(\frac{a}{1-a}\right)^2 & a > 2 \\ \infty & a \leq 2 \end{cases} = \\
&\begin{cases} \frac{a}{(a-2)(a-1)^2} & a > 2 \\ \infty & a \leq 2 \end{cases}
\end{aligned}
$$

c) Let $Y = a \log X$. Find the density of Y.

Notice if $x > 1$, $y = a \log x > 0$, so

$$
\begin{aligned}
&F_Y(y) = P(Y < y) = P(a \log X < y) = \\
&P(X < e^{y/a}) = \int_1^{e^{y/a}} a/x^{a+1} dx = \\
&-x^{-a}|_1^{e^{y/a}} = 1 - e^{-x}
\end{aligned}
$$

so $Y \sim Exp(1)$

2.1.1.2 Example (2.1.2) say we have a discrete random vector (X,Y) with

	0	1	2
0	0.1	0.1	0.2
1	0.0	0.3	0.3

a) Find Cor(X,Y)

Cor(X,Y) = Cov(X,Y)/(sd(X)sd(Y))

Cov(X,Y) = EXY-EX*EY EXY=0*0*0.1+0*1*0.1+. . . +1*2*0.3 = 0.3+2*0.3 = 0.9

	0	1	2	fx
0	0.1	0.1	0.2	0.4
1	0.0	0.3	0.3	0.6
fy	0.1	0.4	0.5	1.0

E[X] = 0*0.4+1*0.6 = 0.6
E[Y] = 0*0.1+1*0.4+2*0.5 = 1.4

cov(X,Y) = E[XY]-E[X]E[Y] = 0.9-0.6*1.4 = 0.06

E[X^2] = 0^2*0.4+1^2*0.6 = 0.6
var(X) = E[X^2]-(E[X])^2 = 0.6-0.6^2 = 0.24
sd(X)=$\sqrt{var(X)} = \sqrt{0.24}$ = 0.489

E[Y^2] = 0^2*0.1+1^2*0.4+2^2*0.5 = 2.4
var(Y) = E[Y^2]-(E[Y])^2 = 2.4-1.4^2 = 0.44
sd(Y)=$\sqrt{var(Y)} = \sqrt{0.44}$ = 0.663

cor(X,Y) = cov(X,Y)/(sd(X)sd(Y)) = 0.06/(0.489*0.663) = 0.185

b) Are X and Y independent?

No, because $cov(X,Y) \neq 0$

or

$f(0,0) = 0.1 \neq f_X(0)f_Y(0) = 0.1 \times 0.4 = 0.04$

c) Find E[X|Y=2]

E[X|Y=2] = $\sum$xf$_{\mathrm{X|Y=1}}$(x|2)
f$_{\mathrm{X|Y=2}}$(x|2) = f(x,2)/f$_{\mathrm{Y}}$(2)
f$_{\mathrm{X|Y=2}}$(0|2) = f(0,2)/f$_{\mathrm{Y}}$(2) = 0.2/0.5 = 0.4
f$_{\mathrm{X|Y=2}}$(1|2) = f(1,2)/f$_{\mathrm{Y}}$(2) = 0.3/0.5 = 0.6

x	P(X=x|Y=1)
0	0.4
1	0.6

E[X|Y=2] = 0*0.4+1*0.6 = 0.6

2.1.1.3 Example (2.1.3)

say we have random variables X,Y~U[0,1] and independent. Find the density of Z=X+Y

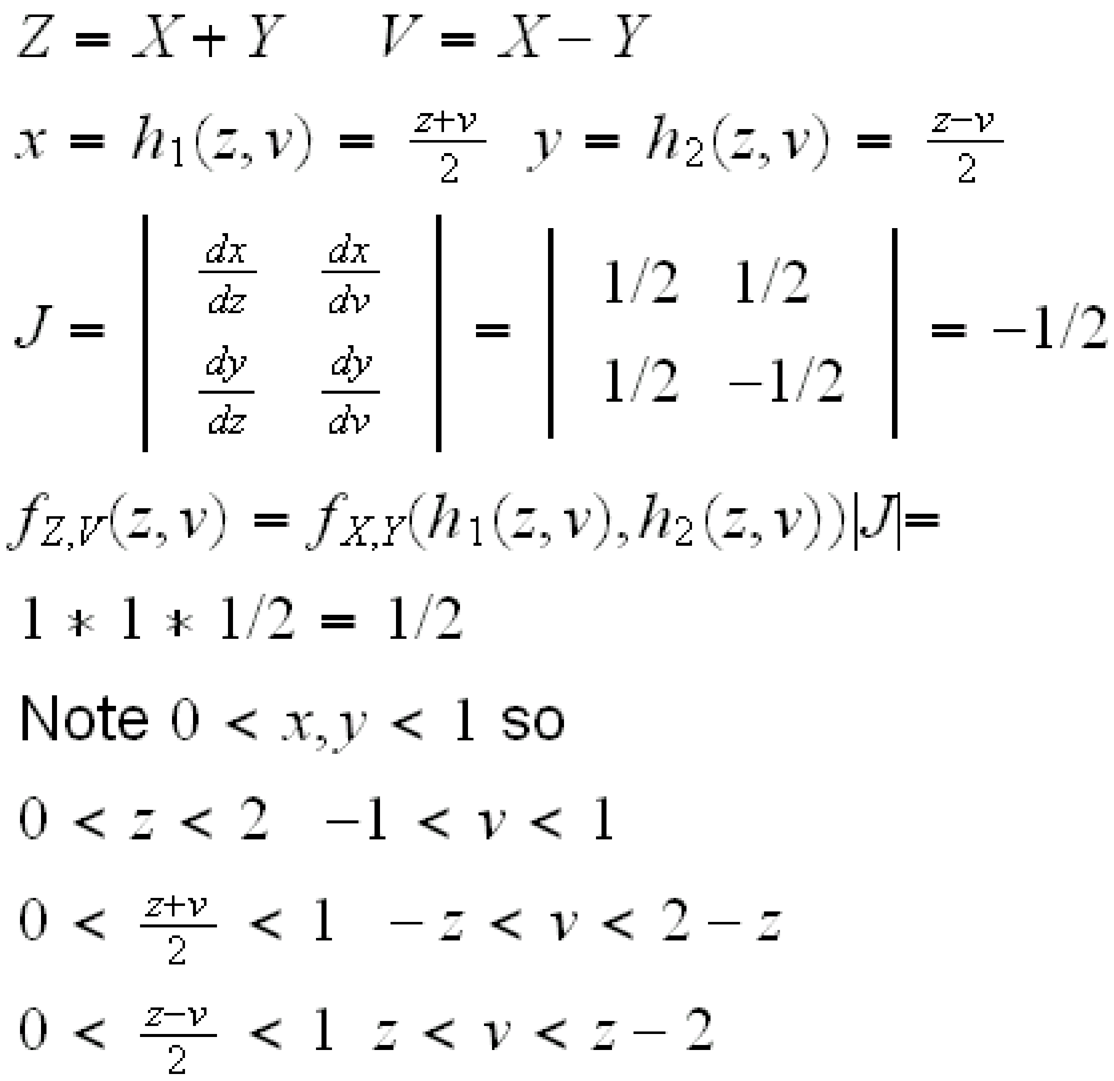
Solution 1:

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

Now $f_x(x) = 0$ if $x < 0$ or $x > 1$, so $f_x(z-y) = 0$ if $z < y$ or $y < z-1$, so

$$\begin{aligned}
&\int_{-\infty}^{\infty} f_X(z-y) f_Y(y)dy = \\
&\begin{cases} \int_0^z 1dy & 0 < z < 1 \\ \int_{z-1}^1 1dy & 1 < z < 2 \end{cases} = \\
&\begin{cases} z & 0 < z < 1 \\ 2-z & 1 < z < 2 \end{cases}
\end{aligned}$$

Solution 2:



$Z = X + Y \qquad V = X - Y$

$x = h_1(z,v) = \frac{z+v}{2} \quad y = h_2(z,v) = \frac{z-v}{2}$

$$J = \begin{vmatrix} \frac{dx}{dz} & \frac{dx}{dv} \\ \frac{dy}{dz} & \frac{dy}{dv} \end{vmatrix} = \begin{vmatrix} 1/2 & 1/2 \\ 1/2 & -1/2 \end{vmatrix} = -1/2$$

$f_{Z,V}(z,v) = f_{X,Y}(h_1(z,v), h_2(z,v))|J| =$

$1 * 1 * 1/2 = 1/2$

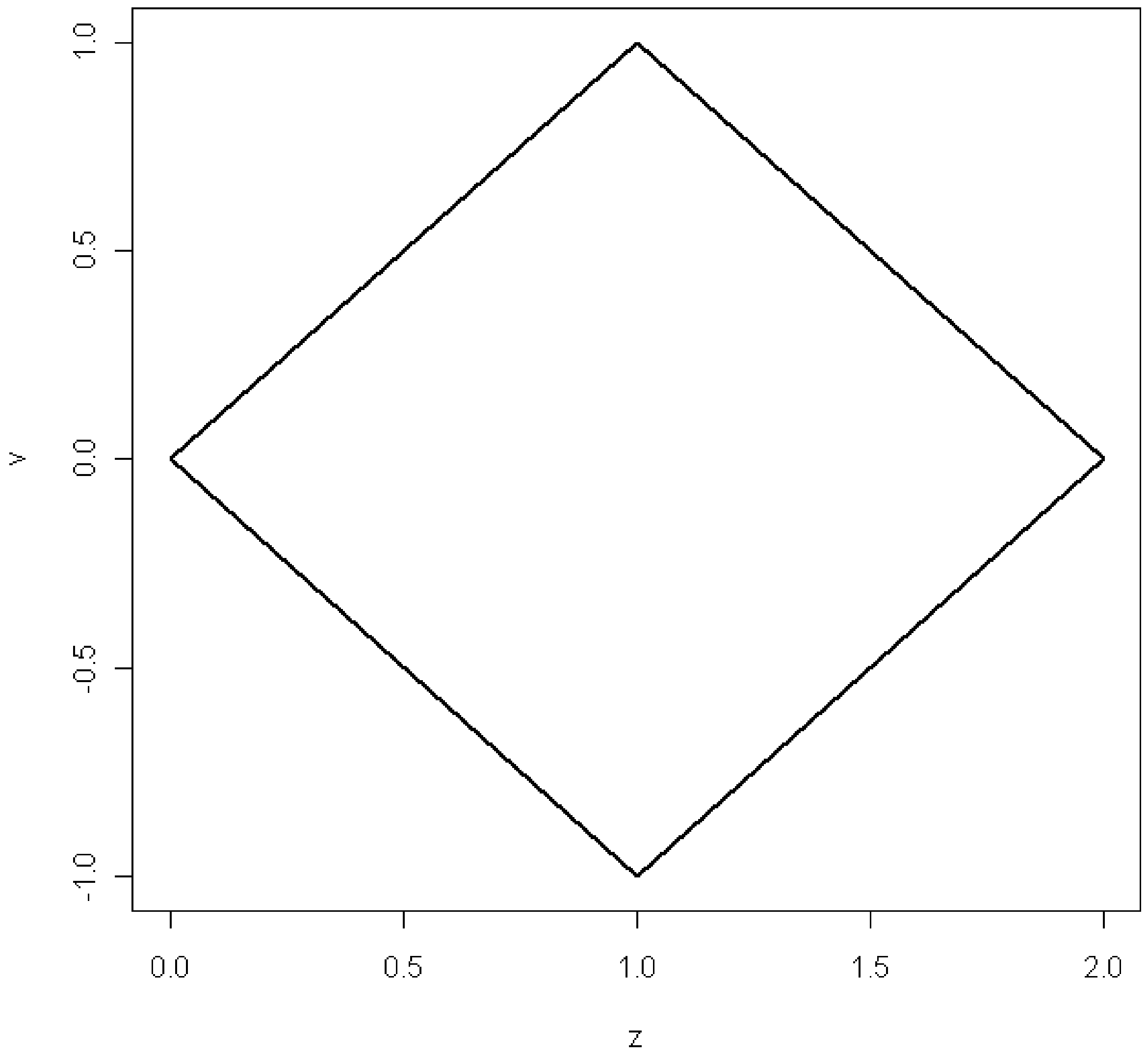
Note $0 < x,y < 1$ so

$0 < z < 2 \quad -1 < v < 1$

$0 < \frac{z+v}{2} < 1 \quad -z < v < 2 - z$

$0 < \frac{z-v}{2} < 1 \quad z < v < z - 2$

here is a figure of this area:



so

$$f_Z(z) = \int_{-\infty}^{\infty} f_{Z,V}(z,v)dv =$$

$$\begin{cases} \int_{-z}^{z} \frac{1}{2}dv & \text{if } 0 < z < 1 \\ \int_{z-2}^{2-z} \frac{1}{2}dv & \text{if } 1 < z < 2 \end{cases} =$$

$$\begin{cases} \frac{1}{2}v|_{-z}^{z} & \text{if } 0 < z < 1 \\ \frac{1}{2}v|_{z-2}^{2-z} & \text{if } 1 < z < 2 \end{cases} =$$

$$\begin{cases} z & \text{if } 0 < z < 1 \\ 2-z & \text{if } 1 < z < 2 \end{cases}$$

2.1.1.4 Example (2.1.4)

Let X_1,..,X_n be a random sample from U{1,..,N} for some N>1. Let $M = \max\{X_i\}$. Show that M→N in probability.

M→N in probability

iff

for all $\epsilon > 0$ $P(|M - N| > \epsilon) \to 0$

Now

$$P(M \le m) = P(X_1 \le m, .., X_n \le m) =$$
$$P(X_1 \le m)^n = (\frac{m}{N})^n, 1 \le m \le N$$
$$P(|M - N| > \epsilon) = 1 - P(N - \epsilon \le M \le N + \epsilon) =$$
$$1 - P(N - \epsilon \le M) = P(M < N - \epsilon) =$$
$$P(M \le \lfloor N - \epsilon \rfloor) =$$
$$(\frac{\lfloor N - \epsilon \rfloor}{N})^n \to 0$$

because $\lfloor N-\epsilon \rfloor < N$.

2.1.1.5 Example (2.1.5)

say (X, Y) has joint density proportional to $g(x,y)=x+y, 0<x,y<1$. Find the distribution function of of Z=E[Y|X].

Here are all the definitions we will need. Note that X and Y are continuous random variables, so

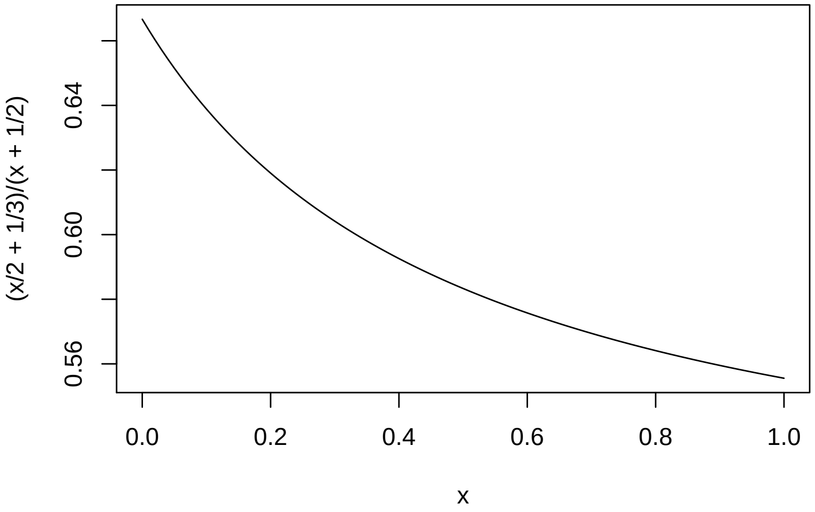
$$
\begin{aligned}
&E[Y|X=x] = \int y f_{Y|X=x}(y|x)dy \\
&f_{Y|X=x}(y|x) = \frac{f(x,y)}{f_X(x)} \\
&f_X(x) = \int f(x,y)dy \\
&f_Y(y) = \int f(x,y)dx
\end{aligned}
$$

Now

$$
\begin{aligned}
&f_X(x) = \int_0^1 c(x+y)dy = c(xy+y^2/2)|_0^1 = c(x+1/2) \\
&1 = \int f_X(x)dx = \int_0^1 c(x+1/2)dx = \\
&c(x^2/2+x/2)|_0^1 = c(1/2+1/2); c=1 \\
&f_X(x) = x+1/2, 0<x<1 \\
&f_Y(y) = y+1/2, 0<y<1 \text{ by symmetry} \\
&f_{Y|X=x} = \frac{f(x,y)}{f_X(x)} = \frac{x+y}{x+1/2} \\
&E[Y|X=x] = \int y f_{Y|X=x}(y|x)dy = \\
&\int_0^1 y\frac{x+y}{x+1/2}dy = \frac{xy^2/2+y^3/3}{x+1/2}|_0^1 = \\
&\frac{x/2+1/3}{x+1/2} \\
&Z = E[Y|X] = \frac{X/2+1/3}{X+1/2}
\end{aligned}
$$

What values does Z take? Let's see:

```
curve((x/2+1/3)/(x+1/2), 0, 1)
```



so $5/9 < z < 2/3$. Now

$$
\begin{aligned}
&F_Z(Z) = P(Z < z) = P(\frac{X/2+1/3}{X+1/2} < z) = \\
&P(X/2+1/3 < zX + z/2) = \\
&P(X(1/2-z) < z/2-1/3) = \\
&P(X > \frac{z-2/3}{1-2z}) = \text{ (because z>1/2)} \\
&1 - \int_0^{\frac{z-2/3}{1-2z}} x + 1/2dx = \\
&1 - (x^2/2 + x/2|_0^{\frac{z-2/3}{1-2z}} = \\
&1 - ((\frac{z-2/3}{1-2z})^2/2 + (\frac{z-2/3}{1-2z})/2) = \\
&1 - ((z-2/3)^2 - (z-2/3)(1-2z))/(2(1-2z)^2) \\
&1 - (3z^2 - 11z/3 + 10/9)/(2(1-2z)^2)
\end{aligned}
$$

2.1.1.6 Example (2.1.6) say $X \sim Pois(\lambda)$, $N = X + 1$ and $Y|N = n \sim Beta(n, 1)$

a) Find E[Y]

$$P(N = n) = P(X + 1 = n) =$$

$$P(X = n - 1) = \frac{\lambda^{n-1}}{(n-1)!} e^{-\lambda}, ..n = 1, 2, \ldots$$

$$f_{Y|N=n}(y|n) = \frac{f(y,n)}{f_N(n)}$$

$$f(y,n) = f_{Y|N=n}(y|n) f_N(n) =$$

$$\frac{\Gamma(n+1)}{\Gamma(n)\Gamma(1)} y^{n-1}(1 - y)^{1-1} \frac{\lambda^{n-1}}{(n-1)!} e^{-\lambda} =$$

$$\frac{n!}{(n-1)!} y^{n-1} \frac{\lambda^{n-1}}{(n-1)!} e^{-\lambda} =$$

$$\frac{n}{(n-1)!} (\lambda y)^{n-1} e^{-\lambda}, ..0 < y < 1, n = 1, 2, \ldots$$

$$f_Y(y) = \sum_{n=1}^{\infty} \frac{n}{(n-1)!} (\lambda y)^{n-1} e^{-\lambda} =$$

$$\sum_{n=1}^{\infty} \frac{(n-1)+1}{(n-1)!} (\lambda y)^{n-1} e^{-\lambda} =$$

$$\sum_{k=0}^{\infty} (k + 1) \frac{(\lambda y)^k}{k!} e^{-\lambda} =$$

$$\left(\sum_{k=0}^{\infty} k \frac{(\lambda y)^k}{k!} e^{-\lambda y} + \sum_{k=0}^{\infty} \frac{(\lambda y)^k}{k!} e^{-\lambda y} \right) e^{\lambda y - \lambda} =$$

$$(EZ + 1) e^{\lambda(y-1)} = (\lambda y + 1) e^{\lambda(y-1)} \quad 0 < y < 1$$

where $Z \sim Pois(\lambda y)$

$$EY = \int_{-\infty}^{\infty} y f_Y(y) = \int_0^1 y(\lambda y + 1) e^{\lambda(y-1)} dy =$$

$$\int_0^1 y^2 \lambda e^{\lambda(y-1)} dy + \int_0^1 y e^{\lambda(y-1)} dy$$

$$y^2 e^{\lambda(y-1)}|_0^1 - \int_0^1 2y e^{\lambda(y-1)} dy + \int_0^1 y e^{\lambda(y-1)} dy$$

$$1 - \int_0^1 y e^{\lambda(y-1)} dy =$$

$$1 - \left(\frac{y}{\lambda} e^{\lambda(y-1)}|_0^1 - \int_0^1 \frac{1}{\lambda} e^{\lambda(y-1)} dy \right) =$$

$$1 - \left(\frac{1}{\lambda} - \frac{1}{\lambda^2} e^{\lambda(y-1)}|_0^1 \right) = 1 - \frac{1}{\lambda} + \frac{1}{\lambda^2}(1 - e^{-\lambda})$$

b) Find E[X|Y=y]

$$f_{N|Y=y}(n|y) = \frac{f(n,y)}{f_Y(y)} = \frac{\frac{n}{(n-1)!}(\lambda y)^{n-1}e^{-\lambda}}{(\lambda y+1)e^{\lambda(y-1)}} =$$

$$\frac{n}{(n-1)!}\frac{(\lambda y)^{n-1}}{\lambda y+1}e^{-\lambda y}, ..n = 1, 2, \ldots$$

$$E[N|Y = y] = \sum_{n=1}^{\infty} n\frac{n}{(n-1)!}\frac{(\lambda y)^{n-1}}{\lambda y+1}e^{-\lambda y} =$$

$$\frac{1}{\lambda y+1}\sum_{n=1}^{\infty} n^2\frac{(\lambda y)^{n-1}}{(n-1)!}e^{-\lambda y} =$$

$$\frac{1}{\lambda y+1}\sum_{n=1}^{\infty}[(n-1)+1]^2\frac{(\lambda y)^{n-1}}{(n-1)!}e^{-\lambda y} =$$

$$\frac{1}{\lambda y+1}\sum_{k=0}^{\infty}(k+1)^2\frac{(\lambda y)^{k}}{k!}e^{-\lambda y} =$$

$$\frac{1}{\lambda y+1}E(Z+1)^2 = \frac{1}{\lambda y+1}E(Z^2+2Z+1) =$$

$$\frac{1}{\lambda y+1}[EZ^2+2EZ+1)] =$$

$$\frac{1}{\lambda y+1}\Big[VarZ+(EZ)^2+2EZ+1)\Big] =$$

$$\frac{1}{\lambda y+1}\Big[\lambda y+(\lambda y)^2+2\lambda y+1)\Big] = \frac{(\lambda y)^2+3\lambda y+1}{\lambda y+1}$$

$$E[X|Y = y] = E[N-1|Y = y] =$$

$$E[X|Y = y]-1 = \frac{(\lambda y)^2+3\lambda y+1}{\lambda y+1}-1 =$$

$$\frac{(\lambda y)^2+2\lambda y}{\lambda y+1} = \lambda y\frac{\lambda y+2}{\lambda y+1}$$

2.1.1.7 Example (2.1.7) Say $X_1, .., X_{10}$ are iid $N(10, 3)$. Find $P(\sum X_i > 110)$.

The sum of independent normal random variables is again normal. Also

$$
\begin{array}{l}
E[\sum X_i] = \sum E[X_i] = 10 \times 10 = 100 \\
Var[\sum X_i] = \sum Var[X_i] = 10 \times 3^2 = 90 \\
\frac{\sum X_i - 100}{\sqrt{90}} \sim N(0,1) \\
P(\sum X_i > 110) = \\
P(\frac{\sum X_i - 100}{\sqrt{90}} > \frac{110-100}{\sqrt{90}}) = \\
P(Z > \frac{10}{\sqrt{90}}) = 1 - \Phi(\frac{\sqrt{10}}{3})
\end{array}
$$

```r
1-pnorm(sqrt(10)/3)
```

```
## [1] 0.1459203
```

2.1.1.8 Example (2.1.8) Say $X_1, .., X_{10}$ are iid with $E[X_1] = 10$ and $sd(X_1) = 3$. Find $P(\sum X_i > 110)$.

If the central limit theorem holds, we again have $\sum X_i \sim N(100, \sqrt{90})$ and so again $P(\sum X_i > 110) = 0.146$.

2.1.1.9 Example (2.1.9) Say $X_1, .., X_n$ are iid Geom(p). Let $T = \sum X_i$.

Are the population mean and median of T the same?

First the population mean of T:

$$X_1,\dots,X_n \sim Geom(p)$$

$$EX_1 = \sum_{i=1}^{\infty} ip(1-p)^{i-1} =$$

$$p\sum_{i=1}^{\infty}\left(\sum_{k=1}^{i} 1\right)(1-p)^{i-1} =$$

$$p\sum_{i=1}^{\infty}\sum_{k=1}^{i}(1-p)^{i-1} =$$

$$p\sum_{k=1}^{\infty}\sum_{i=k}^{\infty}(1-p)^{i-1} =$$

$$p\sum_{k=1}^{\infty}\left[\sum_{i=1}^{\infty}(1-p)^{i-1} - \sum_{i=1}^{k-1}(1-p)^{i-1}\right] =$$

$$p\sum_{k=1}^{\infty}\left[\sum_{i=0}^{\infty}(1-p)^{i} - \sum_{i=0}^{k-2}(1-p)^{i}\right] =$$

$$p\sum_{k=1}^{\infty}\left[\frac{1}{1-(1-p)} - \frac{1-(1-p)^{k-1}}{1-(1-p)}\right] =$$

$$\sum_{k=1}^{\infty}(1-p)^{k-1} = \sum_{k=0}^{\infty}(1-p)^{k} =$$

$$\frac{1}{1-(1-p)} = \frac{1}{p}$$

$$E\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} EX_i = \frac{n}{p}$$

Now the population median is defined as follows: Say T~F, then M such $P(T \leq M) = 0.5$. Let's try first the case n=1:

$$P(X_1 \le M) = \sum_{k=1}^{M} p(1-p)^{k-1} = p\sum_{k=0}^{M-1}(1-p)^k$$

$$p\frac{1-(1-p)^M}{1-(1-p)} = 1-(1-p)^M = 0.5$$

$$M = \frac{\log 0.5}{\log(1-p)}$$

for example if p=0.1 we find $M = 6.57 < 10 = 1/p$, so the median is not equal to the mean.

How about the case n=2? First we need to find the distribution of $X_1 + X_2$:

$$P(X_1+X_2=k) = \sum_{i=1}^{k-1} P(X_1+X_2=k|X_2=i)P(X_2=i) =$$

$$\sum_{i=1}^{k-1} P(X_1=k-i|)P(X_2=i) = \sum_{i=1}^{k-1} p(1-p)^{k-i-1}p(1-p)^{i-1} =$$

$$p^2\sum_{i=1}^{k-1}(1-p)^{k-2} = (k-1)p^2(1-p)^{k-2},..k=2,3,..$$

and we can use R to find M:

```
find.median <- function (p)
{
  dgeom2 <- function(k,p) {(k-1)*p^2*(1-p)^(k-2)}
  M <- 1
  F <- 0
  repeat {
      M <- M+1
      F <- F+dgeom2(M,0.1)
      if(F>=0.5) break
  }
  M
}
find.median(0.1)
```

```
## [1] 17
```

What about n=3?

$$
\begin{aligned}
&P(X_1+X_2+X_3=k)= \\
&\sum_{i=1}^{k-1} P(X_1+X_2+X_3=k|X_3=i)P(X_3=i)= \\
&\sum_{i=1}^{k-1} P(X_1+X_2=k-i)P(X_3=i)= \\
&\sum_{i=1}^{k-1} (k-i-1)p^2(1-p)^{k-i-2}p(1-p)^{i-1}= \\
&p^3\sum_{i=1}^{k-1} (k-i-1)(1-p)^{k-3}=\,?
\end{aligned}
$$

and now we don't even have a function for F, but we can still use another idea: simulation.

```
find.median <- function (n, p, B=10000)
{
    x<-rep(0,B)
    for(i in 1:n) x <- x+rgeom(B,p)+1
    return(quantile(x,0.5))
}
find.median(20, 0.1)
```

```
## 50%
## 198
```

and this works for any reasonably small number n.

Finally we have another solution if n is very large: let's use the Central Limit Theorem:

$$VarX_1 = EX_1^2 - (EX_1)^2$$

$$EX_1^2 = EX_1(X_1 - 1) + EX_1$$

$$EX_1(X_1 - 1) = \sum_{i=1}^{\infty} i(i-1)p(1-p)^{i-1} =$$

$$p(1-p)\sum_{i=2}^{\infty} i(i-1)(1-p)^{i-2}$$

$$\sum_{i=2}^{\infty} n(n-1)x^{n-2} = \sum_{i=2}^{\infty}\left(\frac{d^2}{dx^2}x^n\right) = \frac{d^2}{dx^2}\sum_{i=2}^{\infty} x^n =$$

$$\frac{d^2}{dx^2}\left(\sum_{i=0}^{\infty} x^n - 1 - x\right) = \frac{d^2}{dx^2}\left(\frac{1}{1-x} - 1 - x\right) =$$

$$\frac{d}{dx}\left(\frac{1}{(1-x)^2} - 1\right) = \frac{2}{(1-x)^3}$$

$$EX_1^2 = p(1-p)\frac{2}{(1-(1-p))^3} + \frac{1}{p} = \frac{2(1-p)}{p^2} + \frac{1}{p}$$

$$VarX_1 = EX_1^2 - (EX_1)^2 = \frac{2(1-p)}{p^2} + \frac{1}{p} - \frac{1}{p^2} =$$

$$\frac{2-2p+p-1}{p^2} = \frac{1-p}{p^2}$$

$$P\left(\sum_{i=1}^{n} X_i \le M\right) = P(\overline{X} \le M/n) =$$

$$P\left(\sqrt{n}\,\frac{\overline{X}-\mu}{\sigma} \le \sqrt{n}\,\frac{M/n-\mu}{\sigma}\right) \approx P\left(Z < \sqrt{n}\,\frac{M/n-\mu}{\sigma}\right) =$$

$$pnorm\left(\sqrt{n}\,\frac{M/n-1/p}{(1-p)/p^2}\right) = 0.5$$

$$\sqrt{n}\,\frac{M/n-1/p}{(1-p)/p^2} = 0 \ldots M = n/p$$

and we see that for large n they are indeed the same. Of course that raises the question how large n has to be for this to work.

In this chapter we briefly discuss some distributions that often come up in Statistics.

2.2 Distributions Arising in Statistics

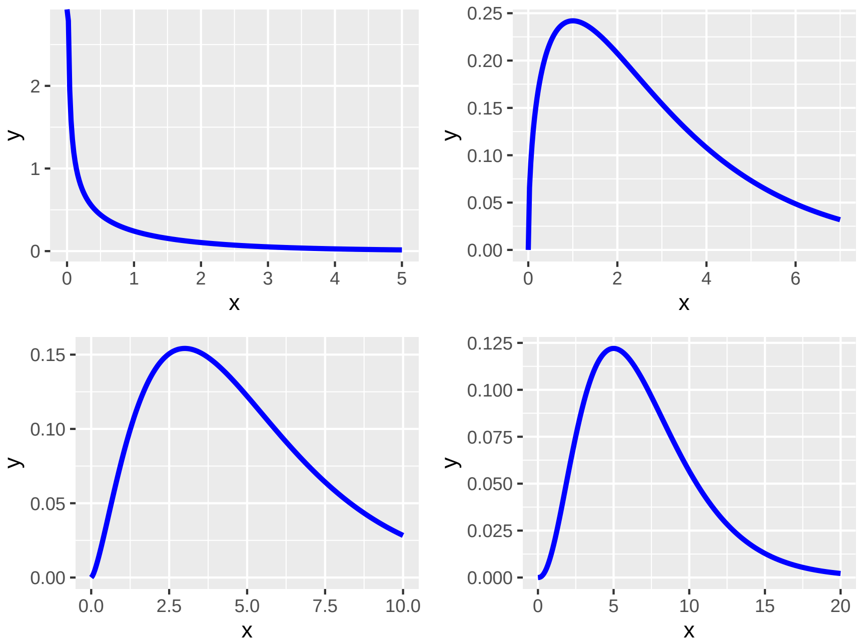
2.2.1 Chisquare Distribution

2.2.1.1 Definition (2.2.1) A random variable X is said to have a chisquare distribution with n degrees of freedom, $X \sim \chi^2(n)$, if it has density

$$f(x|n) = \frac{1}{\Gamma(n/2)2^n} x^{n/2-1} e^{-x/2}; x > 0$$

Of course we have $X \sim \Gamma(n/2, 2)$

```r
pushViewport(viewport(layout = grid.layout(2, 2)))
print(ggcurve(fun=function(x) dchisq(x, 1), A=0, B=5) ,
  vp=viewport(layout.pos.row=1, layout.pos.col=1))
print(ggcurve(fun=function(x) dchisq(x, 3), A=0, B=7) ,
  vp=viewport(layout.pos.row=1, layout.pos.col=2))
print(ggcurve(fun=function(x) dchisq(x, 5), A=0, B=10) ,
  vp=viewport(layout.pos.row=2, layout.pos.col=1))
print(ggcurve(fun=function(x) dchisq(x, 7), A=0, B=20) ,
  vp=viewport(layout.pos.row=2, layout.pos.col=2))
```



Say $Z \sim N(0, 1)$ and let $X = Z^2$, then if x>0

$$
\begin{aligned}
&F_X(x) = P(X < x) = P(Z^2 < x) = \\
&P(-\sqrt{x} < Z < \sqrt{x}) = \\
&\int_{-\sqrt{x}}^{\sqrt{x}} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt \\
&f_X(x) = \frac{dF_x(x)}{dx} = \frac{d}{dx} \int_{-\sqrt{x}}^{\sqrt{x}} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt = \\
&\frac{1}{\sqrt{2\pi}} e^{-(\sqrt{x})^2/2} \frac{1}{2\sqrt{x}} - \frac{1}{\sqrt{2\pi}} e^{-(-\sqrt{x})^2/2} \frac{-1}{2\sqrt{x}} = \\
&\frac{1}{\sqrt{2\pi}} \frac{1}{\sqrt{x}} e^{-x/2} = \\
&\frac{1}{\Gamma(1/2)2^{1/2}} x^{1/2-1} e^{-x/2}
\end{aligned}
$$

so $X \sim \chi^2(1)$

We have the following properties of a chi-square distribution:

2.2.1.2 Theorem (2.2.2) Say $X \sim \chi^2(n)$, $X \sim \chi^2(m)$ and X and Y are independent. Then

- $E[X] = n$
- $var(X) = 2n$
- $X + Y \sim \chi^2(n+m)$

proof

$$
\begin{aligned}
&E[X^k] = \int_0^\infty x^k \frac{1}{\Gamma(n/2)2^{n/2}} x^{n/2-1} e^{-x/2} dx = \\
&\frac{1}{\Gamma(n/2)2^{n/2}} \int_0^\infty x^{k+n/2-1} e^{-x/2} dx = \\
&\frac{\Gamma((2k+n)/2)2^{(2k+n)/2}}{\Gamma(n/2)2^{n/2}} \int_0^\infty \frac{1}{\Gamma((2k+n)/2)2^{(2k+n)/2}} x^{(2k+n)/2-1} e^{-x/2} dx = \\
&\frac{\Gamma(k+n/2)2^{k+n/2}}{\Gamma(n/2)2^{n/2}} = \\
&\frac{(k+n/2-1)(k+n/2-2)..n/2\Gamma(n/2)2^k}{\Gamma(n/2)} = \\
&(k+n/2-1)(k+n/2-2)..(n/2)2^k \\
&E[X] = n/2 \times 2 = n \\
&var(X) = E[X^2] - E[X]^2 = \\
&(n/2+1)(n/2)2^2 - n^2 = n^2 + 2n - n^2 = 2n
\end{aligned}
$$

For the last part we use the convolution formula:

$$
\begin{aligned}
&f_{X+Y}(z) = \int_{-\infty}^{\infty} f_X(t) f_Y(z-t)dt = \\
&\int_0^z f_X(t) f_Y(z-t)dt = (0<t<z) \\
&\int_0^z \frac{1}{\Gamma(n/2)2^{n/2}} t^{n/2-1} e^{-t/2} \frac{1}{\Gamma(m/2)2^{m/2}} (z-t)^{m/2-1} e^{-(z-t)/2} dt = \\
&\frac{1}{\Gamma(n/2)2^{n/2}} \frac{1}{\Gamma(m/2)2^{m/2}} e^{-z/2} \int_0^z t^{n/2} (z-t)^{m/2-1} dt = (u=t/z, du=dt/z) \\
&\frac{1}{\Gamma(n/2)\Gamma(m/2)2^{(n+m)/2}} e^{-z/2} \int_0^1 (zu)^{n/2} (z-zu)^{m/2-1} z du = \\
&\frac{1}{\Gamma(n/2)\Gamma(m/2)2^{(n+m)/2}} e^{-z/2} z^{n/2+m/2-1} \int_0^1 u^{n/2} (1-u)^{m/2-1} du = \\
&\frac{1}{\Gamma((n+m)/2)2^{(n+m)/2}} z^{(n+m)/2-1} e^{-z/2} \int_0^1 \frac{\Gamma((n+m)/2)}{\Gamma(n/2)\Gamma(m/2)} u^{n/2} (1-u)^{m/2-1} du = \\
&\frac{1}{\Gamma((n+m)/2)2^{(n+m)/2}} z^{(n+m)/2-1} e^{-z/2}
\end{aligned}
$$

because the last integrand is a Beta density.

From this theorem it follows that if $Z_1, .., Z_n$ are iid N(0,1), then $\sum Z_i^2 \sim \chi^2(n)$.

2.2.1.3 Definition (2.2.3)

Say $\mathrm{X_1}$, .., $\mathrm{X_n}$ are a sample, then the **sample variance** is defined by

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

2.2.1.4 Theorem (2.2.4)

Say $\mathrm{X_1}$, .., $\mathrm{X_n}$ are iid N(μ,σ), then

$(n-1)S^2/\sigma^2 \sim \chi^2(n-1)$

proof

First note that

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

Now we know that $\frac{X_i-\mu}{\sigma} \sim N(0,1)$, and so $\sum_{i=1}^{n}\left(\frac{X_i-\mu}{\sigma}\right)^2 \sim \chi^2(n)$. Also $\bar{X} \sim N(\mu, \sigma/\sqrt{n})$, and so $\frac{\bar{X}-\mu}{\sigma/\sqrt{n}} \sim N(0,1)$ and $\frac{n(\bar{X}-\mu)^2}{\sigma^2} \sim \chi^2(1)$. Finally

$$(n-1)s^2/\sigma^2 = \sum_{i=1}^{n}\left(\frac{X_i-\mu}{\sigma}\right)^2 - \frac{n(\bar{X}-\mu)^2}{\sigma^2} \sim \chi^2(n-1)$$

Note: we use "n-1" instead of "n" because then s^2 is an unbiased estimator of σ^2, that is $E[s^2]=\sigma^2$.

Note: another important feature here is that $\bar{x} \perp s^2$.

2.2.2 Student's t Distribution (by W.S. Gosset)

2.2.2.1 Definition (2.2.5) Say $X \sim N(0,1), Y \sim \chi^2(n)$ and $X \perp Y$. Then

$$T_n = X/\sqrt{Y/n}$$

has a Student's t distribution with n degrees of freedom, $T_n \sim t(n)$, that is

$$f(t|n) = \frac{\Gamma(\frac{n+1}{2})}{\Gamma(\frac{n}{2})}\frac{1}{\sqrt{\pi n}}\frac{1}{(1+t^2/n)^{(n+1)/2}}$$

Note

$$\frac{1}{(1+t^2/n)^{(n+1)/2}} = \frac{1}{(1+\frac{t^2/2}{n/2})^{n/2}} \frac{1}{(1+\frac{t^2/2}{n/2})^{1/2}} \rightarrow_{n\rightarrow\infty} e^{-t^2/2}$$

so $T_n \rightarrow N(0,1)$ in distribution.

We have $E[T_n] = 0$ if n>1 (and does not exist if n=1) and $var(T_n) = n/(n-2)$ if n>2 (and does not exist if $n \le 2$).

The importance of this distribution in Statistics comes from the following:

2.2.2.2 Theorem (2.2.6)

say $X_1, ..., X_n \sim N(\mu, \sigma)$. Then

$$\sqrt{n}\frac{\bar{X}-\mu}{s} \sim t(n-1)$$

Note: s is of course an estimate of the population standard deviation, so this formula tries to standardize the sample mean without knowing the exact standard deviation.

An important special case is $X \sim t(1)$. This is also called the *Cauchy distribution.* Notice it has no finite mean (and of course then also no finite variance). It has density and cdf:

$$\begin{aligned} f(x) &= \frac{1}{\pi(1+x^2)} \\ F(x) &= \frac{1}{\pi}\arctan(x) \end{aligned}$$

2.2.3 Snedecor's F distribution

2.2.3.1 Definition (2.2.7)

X is said to have an f distribution with n and m degrees of freedom, X~F(n,m) if

$$f(x;n,m) = \frac{\Gamma((n+m)/2)}{\Gamma(n/2)\Gamma(m/2)}(\frac{n}{m})^{n/2}\frac{x^{n/2-1}}{(1+nx/m)^{(n+m)/2}}$$

if x>0

2.2.3.2 Theorem (2.2.8)

Say $X \sim \chi^2(n), Y \sim \chi^2(m)$, independent, then the random variable $F = \frac{X/n}{Y/m} \sim F(n,m)$.

We have $E[F] = m/(m-2)$ (no mention of n!)

2.2.3.3 Theorem (2.2.9) Say $X_1, ..., X_n \sim N(\mu_x, \sigma_x)$ and $Y_1, ..., Y_m \sim N(\mu_x, \sigma_y)$. Furthermore X_i, Y_j are independent for all i and j. Then

$$F = \frac{s_x^2/\sigma_x^2}{s_y^2/\sigma_y^2} \sim F(n-1, m-1)$$

A very nice tool describing these and many other distributions as well as their relationships was created by Lawrance M. Leemis and Raghu Pasupathy and is described in their Chance August 2019 article "The ties that bind" can be found at http://www.math.wm.edu/~leemis/chart/UDR/UDR.html.

2.2.4 Order Statistics

Many statistical methods, for example the median and the range, are based on an ordered data set. In this section we study some of the common distributions of order statistics.

One of the difficulties when dealing with order statistics are ties, that is the same observation appearing more than once. This should only occur for discrete data because for continuous data the probability of a tie is zero. They may happen anyway because of rounding, but we will ignore them in what follows.

Say X_1, .., X_n are iid with density f. Then $X_{(i)}$ is the ith order statistics if $X_{(1)} < \dots < X_{(i)} < \dots < X_{(n)}$

Note $X_{(1)} = \min\{X_i\}$ and $X_{(n)} = \max\{X_i\}$.

Let's find the density of $X_{(i)}$. For this let Y be a r.v. that counts the number of $X_j \leq x$ for some fixed number x. We can think of Y as the number of "successes" of n independent Bernoulli trials with success probability $p = P(X_i \leq x) = F(x)$ for i=1,..,n.

So Y~Bin(n,F(x)). Note also that the event $\{Y \geq i\}$ means that more than i observations are less or equal to x, so the ith largest is less or equal to x. Therefore

$$\begin{aligned} &F_{X_(i)}(x) = P(X_(i) \leq x) = \\ &P(Y \geq i) = \sum_{k=i}^{n} \binom{n}{k} F(x)^k (1-F(x))^{n-k} \end{aligned}$$

with that we find

$$
\begin{aligned}
&\frac{dF_{X_{(i)}}(x)}{dx} = \frac{d}{dx}\sum_{k=i}^{n}\binom{n}{k}F(x)^k(1-F(x))^{n-k}\\
&\sum_{k=i}^{n}\frac{d}{dx}\left[\binom{n}{k}F(x)^k(1-F(x))^{n-k}\right] =\\
&\sum_{k=i}^{n}\binom{n}{k}\left[kF(x)^{k-1}f(x)(1-F(x))^{n-k}+F(x)^k(n-k)(1-F(x))^{n-k-1}(-f(x))\right] =\\
&\sum_{k=i}^{n}\binom{n}{k}\left[kF(x)^{k-1}(1-F(x))^{n-k}-(n-k)F(x)^k(1-F(x))^{n-k-1}\right]f(x)
\end{aligned}
$$

to simplify the notation for a while let's set $t = F(X)$. Also note that the last term $f(x)$ does not depend on k, and so we have the sum

$$
\begin{aligned}
&\sum_{k=i}^{n}\binom{n}{k}\left[kt^{k-1}(1-t)^{n-k}-(n-k)t^k(1-t)^{n-k-1}\right] =\\
&\sum_{k=i}^{n}\binom{n}{k}kt^{k-1}(1-t)^{n-k}-\sum_{k=i}^{n-1}\binom{n}{k}(n-k)t^k(1-t)^{n-k-1} = \{n-n=0\}\\
&\binom{n}{i}it^{i-1}(1-t)^{n-i}+\\
&\sum_{k=i+1}^{n}\binom{n}{k}kt^{k-1}(1-t)^{n-k}-\sum_{k=i}^{n-1}\binom{n}{k}(n-k)t^k(1-t)^{n-k-1} =\\
&\frac{n!}{(n-i)!i!}it^{i-1}(1-t)^{n-i}+\\
&\sum_{k=i}^{n-1}\binom{n}{k+1}(k+1)t^k(1-t)^{n-k-1}-\sum_{k=i}^{n-1}\binom{n}{k}(n-k)t^k(1-t)^{n-k-1}
\end{aligned}
$$

where the last equality follows from a change of summation index.

Note that

$$
\begin{aligned}
\binom{n}{k+1}(k+1) &= \frac{n!(k+1)}{(n-k-1)!(k+1)!} = \frac{n!}{(n-k-1)!k!}\\
\binom{n}{k}(n-k) &= \frac{n!(n-k)}{(n-k)!(n-(n-k))!} = \frac{n!}{(n-k-1)!k!}
\end{aligned}
$$

and so the two sums are actually the same and therefore cancel out. So we find

$$
f_{X_{(i)}}(x) = \frac{n!}{(n-i)!(i-1)!}F(X)^{i-1}(1-F(X))^{n-i}f(x)
$$

2.2.4.1 Example (2.2.10) Say X_1, .., X_n are iid U[0,1]. Then for 0<x<1 we have f(x)=1 and F(x)=x. Therefore

$$f_{X_{(1)}}(x) = \frac{n!}{(i-1)!(n-i)!} x^{i-1}(1-x)^{n-i} =$$

$$\frac{\Gamma(n+1)}{\Gamma(i)\Gamma(n-i+1)} x^{i-1}(1-x)^{(n-i+1)-1}$$

so $X_{(1)} \sim Beta(i, n-i+1)$. Therefore

- $E[X_{(1)}] = \frac{i}{n+1}$
- $var(X_{(1)}) = \frac{i(n-i+1)}{(n+1)^2(n+2)}$

2.2.5 Empirical Distibution Function

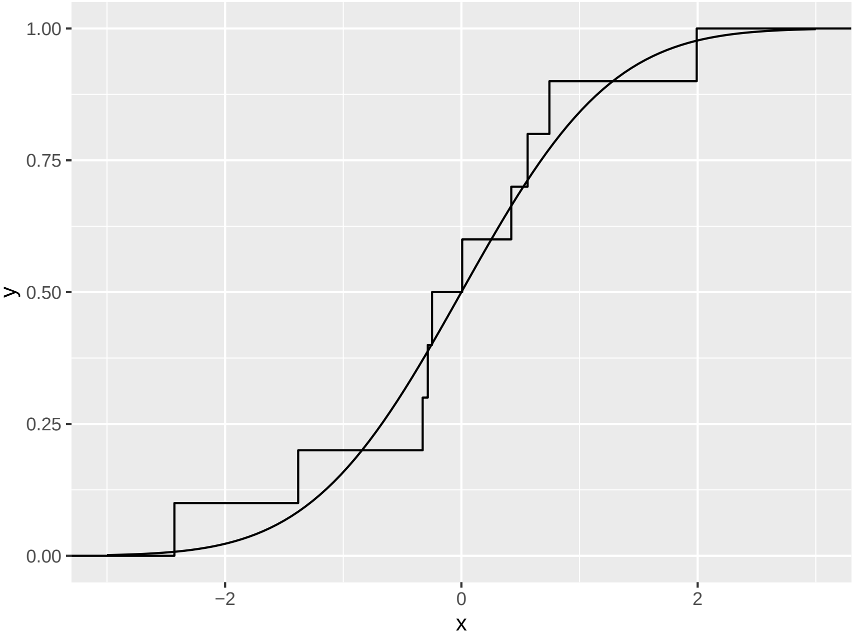
The empirical distribution function of a sample $X_1, .., X_n$ is defined as follows:

$$\hat{F}(x) = \frac{1}{n} \sum_{i=1}^{n} I_{(-\infty,x]}(X_i) = \frac{\#X_i \le x}{n}$$

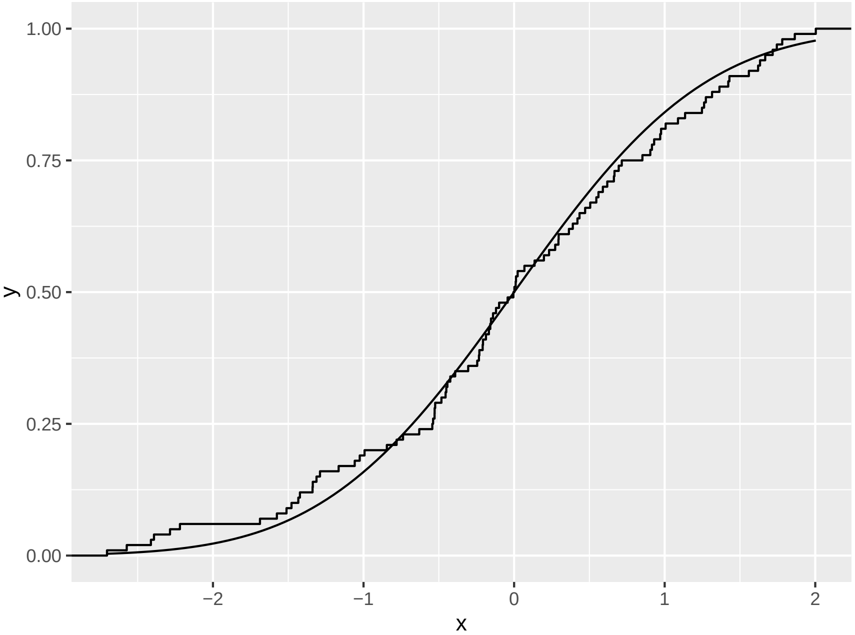
so it is the sample equivalent of the regular distribution function:

- $F(x) = P(X \le x)$ is the probability that the rv $X \le x$.
- $\hat{F}(x)$ is the proportion of $X_1, .., X_n \le x$.

```
df <- data.frame(x = rnorm(10))
ggplot(df, aes(x)) +
  stat_ecdf(geom = "step") +
  stat_function(fun=pnorm) +
  xlim(c(-3, 3))
```



```r
df <- data.frame(x = rnorm(100))
ggplot(df, aes(x)) +
  stat_ecdf(geom = "step") +
  stat_function(fun=pnorm)
```



Let $Z_i = I_{(-\infty,x]}(X_i)$, then $P(Z_i = 1) = P(X_i \le x) = F(x)$, and so $Z_i \sim Ber(F(x))$. $X_1, ..., X_n$ are independent, and therefore $\sum_{i=1}^n Z_i \sim Bin(n, F(x))$. By the weak law of large number

$$\hat{F}(x) = \frac{1}{n}\sum_{i=1}^{n} Z_i \rightarrow E[Z_1] = F(x)$$

in probability. By the central limit theorem

$$\sqrt{n}\frac{\hat{F}(x) - F(x)}{\sqrt{F(x)(1 - F(x))}} \rightarrow N(0, 1)$$

in distribution.

2.2.6 Exponential Family

Definition

A distribution is said to belong to the *exponential family* if its density can be written as

$$f(x; \theta) = h(x) \exp\left\{\theta^T T(x) - A(\theta)\right\}$$

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

$$\begin{aligned}
&\int f(x; \theta)dx = \\
&\int h(x) \exp\left\{\theta^T T(x) - A(\theta)\right\} dx = \\
&\exp\left\{-A(\theta)\right\} \int h(x) \exp\left\{\theta^T T(x)\right\} dx = 1
\end{aligned}$$

so

$$A(\theta) = \log\left[\int h(x) \exp\left\{\theta^T T(x)\right\} dx\right]$$

2.2.6.1 Example (2.2.11)

- Bernoulli

$$
\begin{aligned}
&f(x;p) = p^x(1-p)^{1-x} \\
&\exp\left\{x\log p + (1-x)\log(1-p)\right\} = \\
&\exp\left\{x(\log p - \log(1-p)) + \log(1-p)\right\} = \\
&\exp\left\{x\log\frac{p}{1-p} + \log(1-p)\right\} \\
&\exp\left\{x\theta - \log(1+e^{\theta})\right\}
\end{aligned}
$$

where

$$
\begin{aligned}
&\theta = \log\frac{p}{1-p} \\
&h(x) = 1 \\
&T(x) = x \\
&A(\theta) = -\log(1+e^{\theta})
\end{aligned}
$$

because

$$
\begin{aligned}
&\theta = \log\frac{p}{1-p} \\
&e^{\theta} = \frac{p}{1-p} \\
&p = \frac{e^{\theta}}{1+e^{\theta}} \\
&1-p = \frac{1}{1+e^{\theta}} \\
&\log(1-p) = -\log(1+e^{\theta})
\end{aligned}
$$

- Normal

$$
\begin{aligned}
&\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{1}{2\sigma^2}\left(x-\mu\right)^2\right\} = \\
&\frac{1}{\sqrt{2\pi}}\exp\left\{-\frac{1}{2\sigma^2}\left(x^2-2x\mu+\mu^2\right) - \log\sigma\right\} = \\
&\frac{1}{\sqrt{2\pi}}\exp\left\{-\frac{x^2}{\sigma^2} + \frac{x\mu}{\sigma^2} - \frac{\mu^2}{2\sigma^2} - \log\sigma\right\}
\end{aligned}
$$

so

$$\begin{aligned}
&\theta = (\mu/\sigma^2, -1/(2\sigma^2)^T \\
&h(x) = \frac{1}{\sqrt{2\pi}} \\
&T(x) = (x, x^2)^T \\
&A(\theta) = \frac{\mu^2}{2\sigma^2} + \log\sigma = \\
&-\theta_1^2/(4\theta_2) - \frac{1}{2}\log(-2\theta_2)
\end{aligned}$$

2.2.6.2 Theorem (2.2.12) The product of exponential families is an exponential family

proof

$$\begin{aligned}
&h_1(x)\exp\left\{\theta_1^T T_1(x) - A_1(\theta)\right\} h_2(x)\exp\left\{\theta_2^T T_2(x) - A_2(\theta)\right\} = \\
&h_1(x)h_2(x)\exp\left\{\theta_1^T T_1(x) + \theta_2^T T_2(x) - A_1(\theta_1) - A_2(\theta_2)\right\} = \\
&h(x)\exp\left\{\psi^T S(x) - A(\psi)\right\}
\end{aligned}$$

where

$$\begin{aligned}
&h(x) = h_1(x) + h_2(x) \\
&\psi = (\theta_1, \theta_2)^T \\
&S = (T_1, T_2)^T \\
&A(\psi) = A_1(\theta_1) + A_2(\theta_2)
\end{aligned}$$

The importance of exponential families is that they share many properties and that many theorems can be proven for all of them simultaneously.

2.3 The Likelihood Function and Likelihood Principle

2.3.1 Likelihood Function

One of the most important “objects” in Statistics is the **likelihood function** defined as follows:

Let $\boldsymbol{X} = (X_1, .., X_n)$ be a random vector with joint density f($\boldsymbol{x}|\theta$). then the likelihood function L is defined as

$$L(\theta|\boldsymbol{x}) = f(\boldsymbol{x}|\theta)$$

This must be one of the most deceivingly simple equations in math, actually it seems to be just a change in notation: L instead of f. What really matters and makes a huge difference is that in the density we consider the x’s as variables and the θ as fixed, whereas in the likelihood function we consider the θ as the variable(s) and the x’s as fixed. Essentially we have:

- $f(\boldsymbol{x}|\theta)$ tells what we can expect to happen when we do an experiment.
- $L(\theta|\boldsymbol{x})$ tells us something about the parameter(s) after the experiment is done.

Things simplify a bit if $X_1, .., X_n$ is an iid sample. Then the joint density is given by

$$f(\boldsymbol{x}|\theta) = \prod f(x_i|\theta)$$

2.3.1.1 Example (2.3.1) $X_1, .., X_n \sim Ber(p)$:

$$\begin{aligned}
&f(\boldsymbol{x}|\pi) = \prod_{i=1}^{n} p^{x_i}(1-p)^{1-x_i} = \\
&p^{\sum x_i}(1-p)^{n-\sum x_i} \\
&L(p|\boldsymbol{x}) = p^{\sum x_i}(1-p)^{n-\sum x_i}
\end{aligned}$$

2.3.1.2 Example (2.3.2) $X_1, .., X_n \sim N(\mu, \sigma)$:

$$\begin{aligned}
&L(\mu, \sigma|\boldsymbol{x}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\{-\frac{1}{2}\frac{(x_i-\mu)^2}{\sigma^2}\} = \\
&(2\pi\sigma^2)^{-n/2} \exp\{-\frac{1}{2\sigma^2}\sum(x_i-\mu)^2\}
\end{aligned}$$

2.3.1.3 Example (2.3.3) $X_1, .., X_n \sim \Gamma(\alpha, \beta)$:

$$\begin{aligned}
&L(\alpha, \beta|\boldsymbol{x}) = \prod_{i=1}^{n} \frac{1}{\Gamma(\alpha)\beta^\alpha} x_i^{\alpha-1} \exp\{-x_i/\beta\} = \\
&\frac{1}{\Gamma(\alpha)^n\beta^{n\alpha}} (\prod_{i=1}^{n} x_i)^{\alpha-1} \exp\{-(\sum x_i)/\beta\}
\end{aligned}$$

2.3.1.4 Example (2.3.4) $Y_1 \sim N(\mu_1, \sigma_1), Y_2 \sim N(\mu_2, \sigma_2), Z \sim Ber(p)$ and $X = (1-Z)Y_1 + ZY_2$.

$$L(p, \mu_1, \sigma_1, \mu_2, \sigma_1|\boldsymbol{x}) = \prod_{i=1}^{n} \left\{ \frac{p}{\sqrt{2\pi\sigma_1^2}} \exp\left(-\frac{(x_i-\mu_1)^2}{2\sigma_1^2}\right) + \frac{1-p}{\sqrt{2\pi\sigma_1^2}} \exp\left(-\frac{(x_i-\mu_2)^2}{2\sigma_1^2}\right) \right\}$$

2.3.1.5 Example (2.3.5)

An urn contains N balls. n$_\mathrm{i}$ of these balls have the number "i" on them, i=1,..,k and $\sum n_i = N$. Say we randomly select a ball from the urn, note its number and put it back. We repeat this m times. Let the rv X$_\mathrm{i}$ be the number of balls with the number i that were drawn, and let $\boldsymbol{X} = (X_1, .., X_k)$. Now the density of $\mathbf{X}$ is given

$$f(\boldsymbol{x}|m, n_1, .., n_k) = \frac{m!}{x_1! \cdot .. \cdot x_k!} \prod_{i=1}^{k} \left(\frac{n_i}{N}\right)^{x_i}$$

for any $x_1, .., x_k$ with $x_i \in \{0, 1, .., N\}$ and $\sum x_i = m$

Now let's assume we don't know $n_1, ..n_k$ and want to estimate them. First we can make a slight change in the parameterization: $p_i = n_i/N$, i=1,..,k. The resulting random vector is called the multinomial rv with parameters m, $p_1, .., p_k$.

Note if k=2 $X_1 \sim Bin(m, p_1)$ and $X_2 \sim Bin(m, p_2)$, so the multinomial is a generalization of the binomial.

The likelihood function is given by

$$L(m, p_1, .., p_k|\boldsymbol{x}) = \frac{m!}{x_1! \cdot .. \cdot x_k!} \prod_{i=1}^{k} p_i^{x_i}$$

where $p_1 + .. + p_k = 1$ and $x_1 + .. + x_k = m$.

There is a common misconception about the likelihood function: because it is the same as the density it has the same properties. This is not true because the likelihood function is a function of the parameters, not the variables.

2.3.1.6 Example (2.3.6)

$X \sim Ber(p)$, so $f(x) = p^x(1-p)^{1-x}$, $x = 0, 1; 0 < p < 1$

As a function of x with a fixed p we have $f(x) \geq 0$ for all x and f(0)+f(1)=1 but as a function of p with a fixed x, say x=1, we have

$$\int_{-\infty}^{\infty} L(p|1)dp = \int_{0}^{1} pdp = \frac{1}{2}$$

2.3.2 Log Likelihood

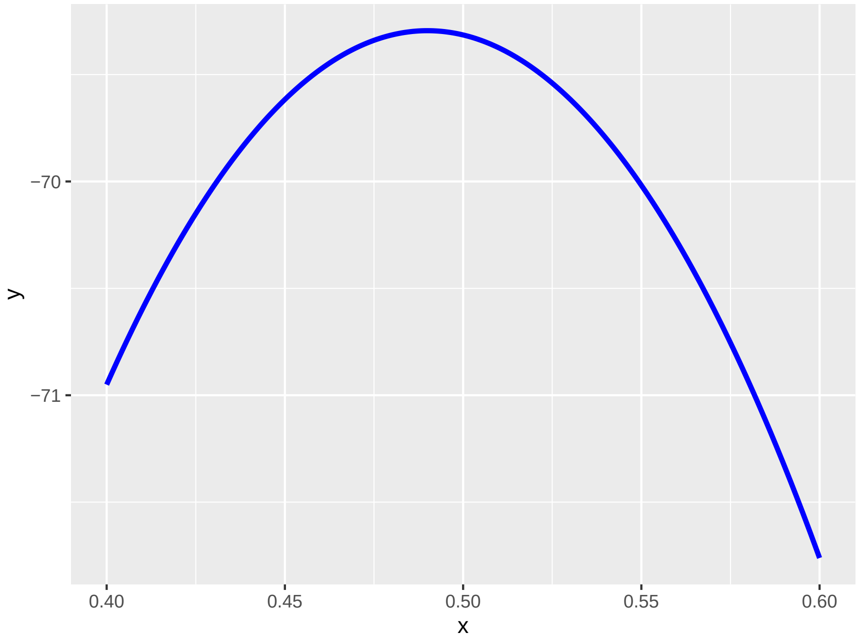
It turns out that for many problems the log of the likelihood function is more manageable entity, mainly because it turns the product into a sum:

2.3.2.1 Example (2.3.7) $X_1, .., X_n \sim Ber(p)$

$$
\begin{aligned}
&l(p|\boldsymbol{x}) = \log L(p|\boldsymbol{x}) = \\
&\log\left(p^{\sum x_i}(1-p)^{n-\sum x_i}\right) = \\
&(\sum x_i)\log p + (n - \sum x_i)\log(1-p)
\end{aligned}
$$

(worry about x_i=0 for all i or x_i=1 for all i yourself)

```
n <- 100; pi <- 0.5
dta <- rbinom(n, 1, pi)
loglike_pi <- function(p) {
  y <- 0*p
  for(i in seq_along(p))
    y[i] <- sum(dta)*log(p[i])+(n-sum(dta))*log(1-p[i])
  y
}
ggcurve(fun=loglike_pi, A=0.4, B=0.6)
```



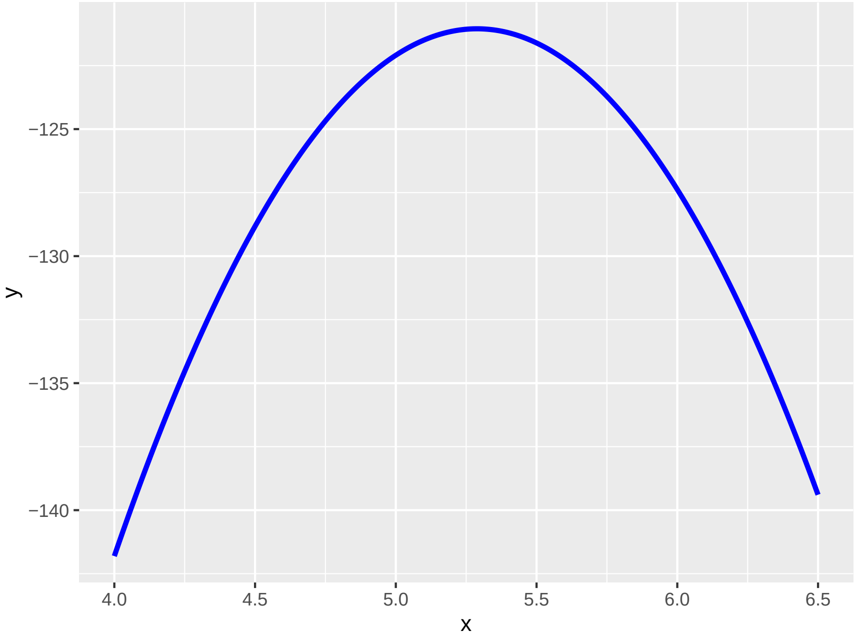
2.3.2.2 Example (2.3.8) $X_1, .., X_n \sim N(\mu, \sigma)$:

$$
\begin{aligned}
&l(\mu, \sigma|\boldsymbol{x}) = \log L(\mu, \sigma|\boldsymbol{x}) = \\
&\log\left((2\pi\sigma^2)^{-n/2}\exp\{-\frac{1}{2\sigma^2}\sum(x_i-\mu)^2\}\right) = \\
&-\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum(x_i-\mu)^2
\end{aligned}
$$

This log-likelihood function is drawn here

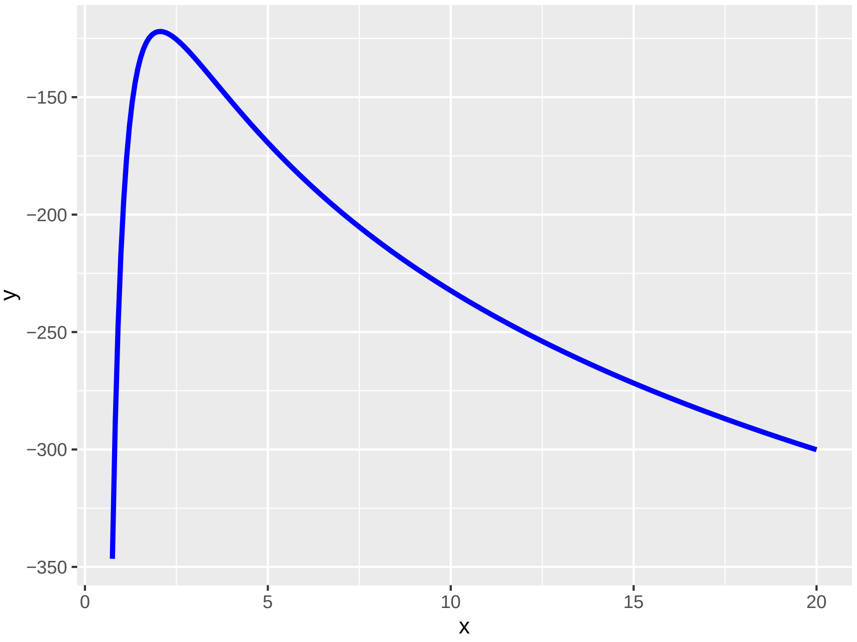
- as a function of μ, with σ fixed:

```
n <- 100; mu <- 5; sig <- 2
dta <- rnorm(n, mu, sig)
loglike_mu <- function(x) {
  y <- x
  for(i in seq_along(x))
    y[i] <- (-n/2*log(2*pi*sig^2)-sum((dta-x[i])^2)/(2*sig^2))
  y
}
ggcurve(fun=loglike_mu, A=4, B=6.5)
```



- as a function of σ, with μ fixed:

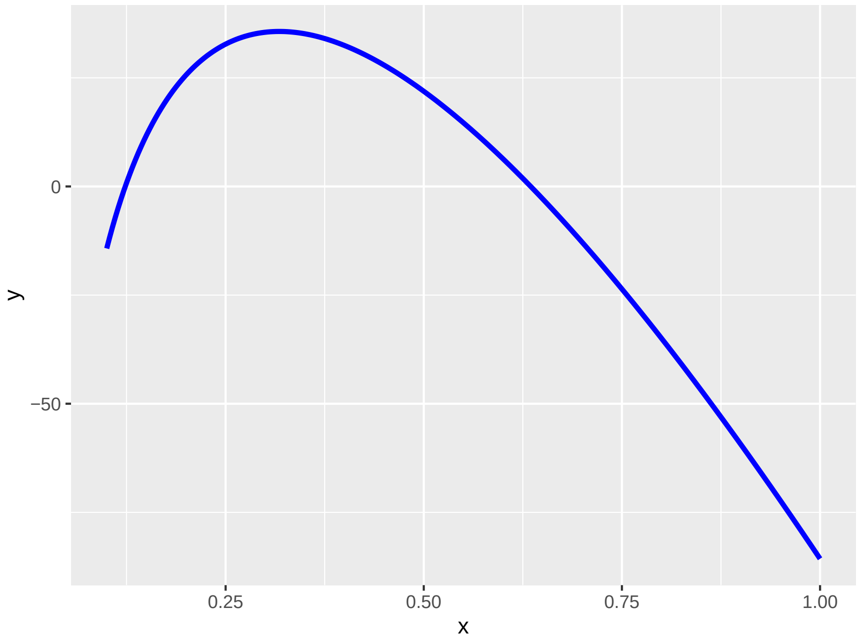
```
loglike_sig <- function(x) {
  y <- x
  for(i in seq_along(x))
    y[i] <-
      (-n/2*log(2*pi*x[i]^2)-sum((dta-mu)^2)/(2*x[i]^2))
  y
}
ggcurve(fun=loglike_sig, A=0.75, B=20)
```



2.3.2.3 Example (2.3.9) $X_1, .., X_n \sim \Gamma(\alpha, \beta)$:

$$
\begin{aligned}
&l(\alpha, \beta) = \log L(\alpha, \beta) = \\
&\log \left[\frac{1}{\Gamma(\alpha)^n \beta^{n\alpha}} (\prod_{i=1}^n x_i)^{\alpha-1} \exp\{-(\sum x_i)/\beta\} \right] = \\
&-\log \Gamma(\alpha) - n\alpha \log \beta + (\alpha - 1) \sum (\log x_i) - (\sum x_i)/\beta
\end{aligned}
$$

```r
alpha <- 0.5; beta <- 2.1
dta <- rgamma(n, alpha, beta)
loglike_alpha <- function(x) {
  y <-  x
  for (i in seq_along(x))
    y[i] <-  sum(log(dgamma(dta, x[i],  1/beta)))
  y
}
ggcurve(fun=loglike_alpha, A=0.1, B=1)
```



2.3.2.4 Example (2.3.10) $Y_1 \sim N(\mu_1, \sigma_1),\ Y_2 \sim N(\mu_2, \sigma_2),\ Z \sim Ber(p)$ and $X = (1 - Z)Y_1 + ZY_2$.

$$l(p, \mu_1, \sigma_1, \mu_2, \sigma_1|\boldsymbol{x}) = \log L(p, \mu_1, \sigma_1, \mu_2, \sigma_1|\boldsymbol{x}) = \sum_{i=1}^{n} \log \left\{ \frac{p}{\sqrt{2\pi\sigma_1^2}} \exp\left(-\frac{(x_i - \mu_1)^2}{2\sigma_1^2} \right) + \frac{1-p}{\sqrt{2\pi\sigma_1^2}} \exp\left(-\frac{(x_i -}{2} \right.\right.$$

2.3.2.5 Example (2.3.11) say X has a multinomial distribution with parameters m, $p_1, .., p_k$, then

$$l(m, p_1, .., p_k|\boldsymbol{x}) = \log L(m, p_1, .., p_k|\boldsymbol{x}) = \frac{m!}{x_1! \cdot .. \cdot x_k!} \prod_{i=1}^{k} p_i^{x_i} = \log \left(\frac{m!}{x_1! \cdot .. \cdot x_k!} \right) + \sum_{i=1}^{k} x_i \log(p_i)$$

2.3.2.6 Example (2.3.12) Say X belongs to an exponential family, then

$$\begin{aligned} &l(\theta; \boldsymbol{x}) = \\ &\log \left[h(x) \exp \left\{ \theta^T T(x) - A(\theta) \right\} \right] = \\ &\log h(x) + \theta^T T(x) - A(\theta) \end{aligned}$$

and in this case the log-likelihood function simplifies nicely.

2.3.3 The Likelihood Principle

2.3.3.1 Definition (2.3.13) Likelihood Principle

If $\boldsymbol{x}$ and $\boldsymbol{x}$ are two sample points such that

$$L(\theta|\boldsymbol{x}) = C(\boldsymbol{x},\boldsymbol{y}) \times L(\theta|\boldsymbol{y})$$

for all θ, then the conclusion drawn from $\boldsymbol{x}$ and $\boldsymbol{y}$ should be identical.

So if two sample points have proportional likelihoods, they contain the same information about the parameter.

2.3.3.2 Example (2.3.14)

say $X_1,..,X_n \sim N(\mu,\sigma)$, $Y_1,..,Y_n \sim N(\mu,\sigma)$ and assume σ is known. Then

$$\begin{aligned}
&\frac{L(\mu|\boldsymbol{x})}{L(\mu|\boldsymbol{y})} = \\
&\frac{(2\pi\sigma^2)^{n/2}\exp\left\{-\frac{1}{2\sigma^2}\sum(x_i-\mu)^2\right\}}{(2\pi\sigma^2)^{n/2}\exp\left\{-\frac{1}{2\sigma^2}\sum(y_i-\mu)^2\right\}} = \\
&(2\pi\sigma^2)^{n/2}\exp\left\{-\frac{1}{2\sigma^2}\left[\sum(x_i-\mu)^2-\sum(y_i-\mu)^2\right]\right\}
\end{aligned}$$

Now

$$\begin{aligned}
&\sum(x_i-\mu)^2 = \\
&\sum(x_i-\bar{x}+\bar{x}-\mu)^2 = \\
&\sum\left[(x_i-\bar{x})^2+2(x_i-\bar{x})(\bar{x}-\mu)+(\bar{x}-\mu)^2\right] = \\
&\sum(x_i-\bar{x})^2+2(\bar{x}-\mu)\sum(x_i-\bar{x})+n(\bar{x}-\mu)^2 = \\
&\sum(x_i-\bar{x})^2+2(\bar{x}-\mu)(\sum x_i-n\bar{x})+n(\bar{x}-\mu)^2 = \\
&\sum(x_i-\bar{x})^2+n(\bar{x}-\mu)^2
\end{aligned}$$

and so

$$\begin{aligned}
&\frac{L(\mu|\boldsymbol{x})}{L(\mu|\boldsymbol{y})} = \\
&(2\pi\sigma^2)^{n/2}\exp\left\{-\frac{1}{2\sigma^2}\left[\sum(x_i-\bar{x})^2+n(\bar{x}-\mu)^2-\sum(y_i-\bar{y})^2-n(\bar{y}-\mu)^2\right]\right\} = \\
&(2\pi\sigma^2)^{n/2}\exp\left\{-\frac{n}{2\sigma^2}\left[(\bar{x}-\mu)^2-(\bar{y}-\mu)^2\right]\right\}\times \\
&\exp\left\{-\frac{1}{2\sigma^2}\left[\sum(x_i-\bar{x})^2-\sum(y_i-\bar{y})^2\right]\right\} = \\
&C(\boldsymbol{x},\boldsymbol{y})
\end{aligned}$$

iff

$$(\bar{x}-\mu)^2-(\bar{y}-\mu)^2=0$$

this has to hold for all μ, which implies $\bar{x}=\bar{y}$.

So according to the likelihood principle if two experiments with the probability model $N(\mu,\sigma)$, σ known observe the same sample means, they should give the same result.

2.3.3.3 Example (2.3.15)

Consider the following problem: we have a Bernoulli trial with success parameter p, and we wish to estimate p.

Experiment 1: in this experiment we repeat the Bernoulli trial 20 times, so the rv X~Bin(20,p). We find x=7. Therefore

$$L_1(p|7)=\binom{20}{7}p^7(1-p)^{13}$$

Experiment 2: in this experiment we repeat the Bernoulli trials until the 7th success, so the rv Y~NegBin(7,p). We find y=20, therefore

$$L_2(p|20)=\binom{19}{6}p^7(1-p)^{13}$$

so now $L_1(p|7)=cL_2(p|20)$ and so according to the likelihood principle both experiments should result in the same estimate of p, regardless of the fact that we performed completely different experiments.

The likelihood principle is a good general principle for a statistical procedure but there are common situations were it is violated. For example, an important task in Statistics is **model checking**. Say for example we have the following probability model: $X_1,..,X_n \sim N(\mu,\sigma)$ and we want to estimate μ. But then we worry whether our data really follows a normal distribution, so we do some checking, for example draw a boxplot. This, though, violates the likelihood principle because for one data set we might decide that the normal assumption is wrong whereas for another we might accept it, even though both have the same sample mean.

3 Bayesian and Frequentist Statistics

3.1 Bayesian Statistics

3.1.1 Bayesian Analysis Basics

In the classical, or frequentist approach to Statistics we consider a parameter θ a fixed although unknown quantity. A random sample X_1, .., X_n is drawn from a population indexed by θ and, based on the observed values in the sample, knowledge about the true value of θ is obtained.

In the Bayesian approach θ is considered a quantity whose variation can be described by a probability distribution (called a prior distribution), which is a subjective distribution describing the experimenters belief and which is formulated before the data is seen. A sample is then taken from a population indexed by θ and the prior distribution is updated with this new information. The updated distribution is called the *posterior* distribution. This updating is done using Bayes' formula, hence the name Bayesian Statistics.

It is not that Bayesians don't think a parameter is a fixed number. Rather it is our lack of knowledge about the parameter that is described by the prior distribution.

Let's denote the prior distribution by $\pi(\theta)$ and the sampling distribution by $f(\mathbf{x}|\theta)$, then the joint density of X and θ is given by

$$f(\mathbf{x},\theta) = \mathbf{f}(\boldsymbol{x}|\theta)\pi(\theta)$$

the marginal of the distribution of $\mathbf{X}$ is

$$m(\boldsymbol{x}) = \int f(\boldsymbol{x}|\theta)\pi(\theta)d\theta$$

and finally the posterior distribution is the conditional distribution of θ given the sample $\boldsymbol{x}$ and is given by

$$\pi(\theta|\boldsymbol{x}) = \frac{f(\boldsymbol{x}|\theta)\pi(\theta)}{m(\boldsymbol{x})}$$

Note the distribution $m(\boldsymbol{x})$ is often called the *prior predictive distribution.*

3.1.1.1 Example (3.1.1)

You want to see whether it is really true that coins come up heads and tails with probability 1/2. You take a coin from your pocket and flip it 10 times. It comes up heads 3 times. As a frequentist we would now use the sample mean as an estimate of the true probability of heads, p and find $\hat{p} = 0.3$.

A Bayesian analysis would proceed as follows: let $X_1, ..., X_n \sim Ber(p)$. Then $Y = X_1 + .. + X_n \sim Bin(n, p)$. Now we need a prior on p. Of course p is a probability, so it has values on [0,1]. One distribution on [0,1] we know is the Beta, so we will use a Beta(α,β) as our prior. Remember, this is a perfectly subjective choice, and anybody can use their own.

We find

$$f(y,p) = \left[\binom{n}{y}p^y(1-p)^{n-y}\right]\left[\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}p^{\alpha-1}(1-p)^{\beta-1}\right] =$$
$$\binom{n}{y}\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}p^{y+\alpha-1}(1-p)^{n-y+\beta-1}$$

for the marginal we find

$$
\begin{aligned}
&m(\boldsymbol{x}) = \int_{-\infty}^{\infty} f(y,p)dp \\
&\int_0^1 \binom{n}{y} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{y+\alpha-1}(1-p)^{n-y+\beta-1} dp = \\
&\binom{n}{y} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \int_0^1 p^{y+\alpha-1}(1-p)^{n-y+\beta-1} dp = \\
&\binom{n}{y} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \frac{\Gamma(y+\alpha)\Gamma(n-y+\beta)}{\Gamma(n+\alpha+\beta)} \times \\
&\int_0^1 \frac{\Gamma(n+\alpha+\beta)}{\Gamma(y+\alpha)\Gamma(n-y+\beta)} p^{y+\alpha-1}(1-p)^{n-y+\beta-1} dp = \\
&\binom{n}{y} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \frac{\Gamma(y+\alpha)\Gamma(n-y+\beta)}{\Gamma(n+\alpha+\beta)}
\end{aligned}
$$

because the integral is over a $Beta(y+\alpha, n-y+\beta)$ density and therefore equal to 1. This is known as the *beta-binomial distribution.*

Finally the posterior distribution of $p|\boldsymbol{X}=\boldsymbol{x}$ is

$$
\begin{aligned}
&\pi(p|\boldsymbol{x}) = \frac{f(\boldsymbol{x}|p)\pi(p)}{m(\boldsymbol{x})} = \\
&\frac{\binom{n}{y}\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{y+\alpha-1}(1-p)^{n-y+\beta-1}}{\binom{n}{y}\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\frac{\Gamma(y+\alpha)\Gamma(n-y+\beta)}{\Gamma(n+\alpha+\beta)}} = \\
&\frac{\Gamma(n+\alpha+\beta)}{\Gamma(y+\alpha)\Gamma(n-y+\beta)} p^{y+\alpha-1}(1-p)^{n-y+\beta-1}
\end{aligned}
$$

and we find that $p|\boldsymbol{X}=\boldsymbol{x} \sim Beta(y+\alpha, n-y+\beta)$

Of course we still need to "extract" some information about the parameter p from the posterior distribution. Once the sampling distribution and the prior are chosen, the posterior distribution is fixed (even though it may not be easy or even possible to find it analytically), but how we proceed now is completely open and there are in general many choices. If we want to estimate p a natural estimator is the mean of the posterior distribution, given here by

$$\hat{p}_B = (y+\alpha)/(\alpha+\beta+n)$$

This can be written as

$$\hat{p}_B = \left(\frac{n}{\alpha+\beta+n}\right)(\frac{y}{n}) + \left(\frac{\alpha+\beta+n}{\alpha+\beta+n}\right)(\frac{\alpha}{\alpha+\beta})$$

and we see that the posterior mean is a linear combination of the prior mean and the sample mean.

How about our problem with the 3 heads in the 10 flips? Well, we have to completely specify the prior distribution, that is we have to choose α and β. The choice depends again on our belief. For example, if we feel strongly that this coin is just like any other coin and therefore really should be a fair coin we should choose them so that the prior puts almost all its weight at around 1/2. For example, with $\alpha = \beta = 100$ we get E[p]=0.5 and var(p)=0.0016. Then

$$\hat{p}_B = (3 + 100)/(100 + 100 + 10) = 0.4905$$

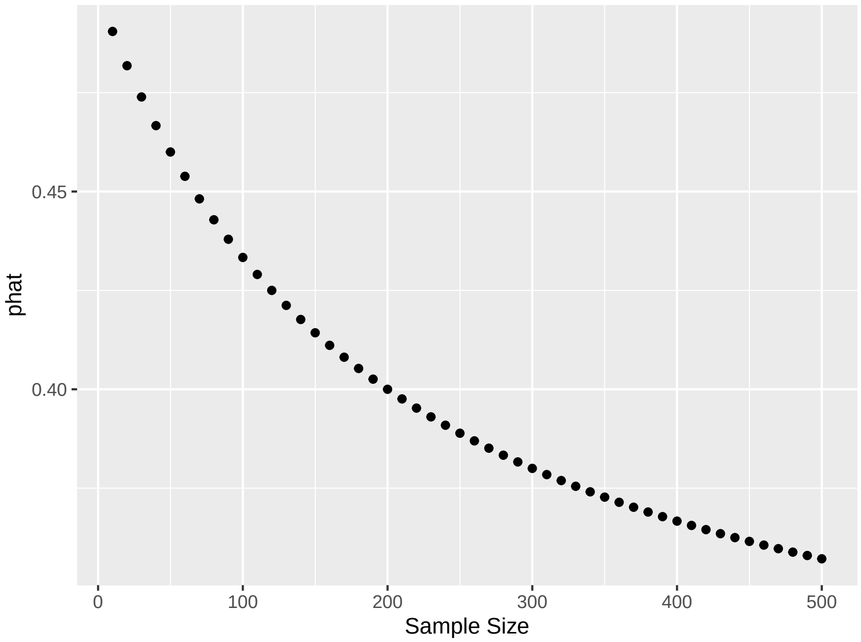
is our estimate for the probability of heads. Clearly for such a strong prior the actual sample almost does not matter, For example for y=0 we would have found $\hat{p}_B = 0.476$ and for y=10 it would be $\hat{p}_B = 0.524$.

Maybe we have never even heard the word "coin" and have no idea what one looks like, let alone what probability of "heads" might be. Then we could choose $\alpha = \beta = 1$, that is the uniform distribution, as our prior. Really this would indicate our complete lack of knowledge regarding p. (this is called an uninformative prior). Now we find $\hat{p}_B = (3+1)/(1+1+10) = 0.3$, which is just the sample mean again.

Let's study the effects of the different parts of our estimator:

- effect of the sample size on the estimate of $\hat{p}$:

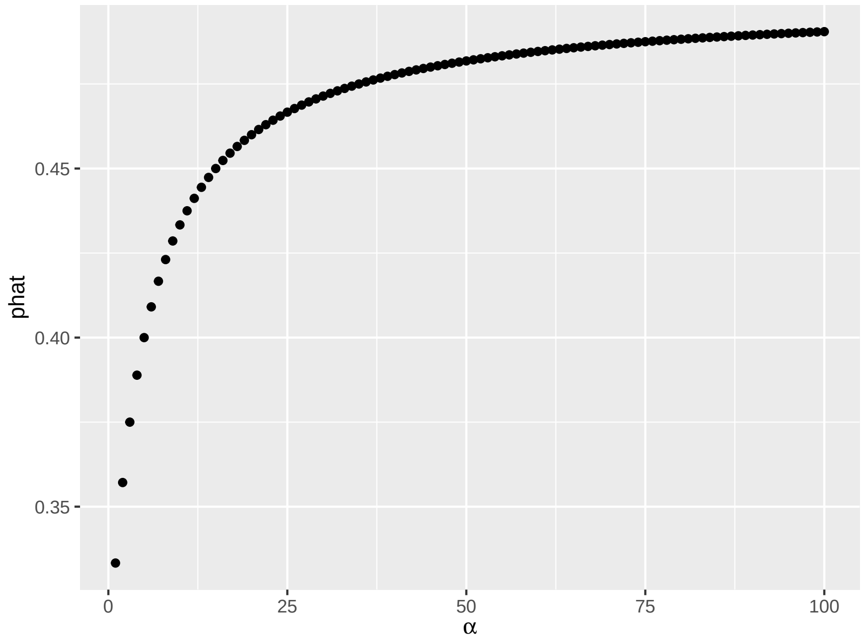
```
df <- data.frame(n=10*1:50,
                 y=(3*1:50 + 100)/(10*1:50 + 200))
ggplot(data=df, aes(n, y)) +
  geom_point() +
  labs(x="Sample Size", y="phat")
```



so as the sample size increases, the estimate moves from close to 0.5 (the prior mean) to 0.3 (the mean of the data). The more data we have the less influence the prior has.

- effect of alpha=beta on the estimate of p:

```
df <- data.frame(alpha=1:100,
                 y=(3+1:100)/(10+2*1:100))
ggplot(data=df, aes(alpha, y)) +
  geom_point() +
  labs(x=expression(alpha), y="phat")
```



A larger alpha means a prior more concentrated around 1/2. The "stronger" the prior, the more it influences the estimate.

3.1.1.2 Example (3.1.2)

say X~Bin(n,p), p known. Again, a Bayesian analysis begins with a prior on n. Now n=1, 2, .. and so a prior is any sequence a_1, a_2, .. with

$a_i \geq 0$ and $\sum a_i = 1$.

Then

$$
\begin{aligned}
f(x,n) &= \left[\binom{n}{x}p^x(1-p)^{n-x}\right]a_n \\
m(\boldsymbol{x}) &= \sum_{n=y}^{\infty}\left[\binom{n}{x}p^x(1-p)^{n-x}\right]a_n \\
\pi(n|\boldsymbol{x}) &= \frac{f(\boldsymbol{x}|p)\pi(p)}{m(\boldsymbol{x})} = \\
\frac{\binom{n}{x}p^x(1-p)^{n-x}a_n}{\sum_{n=y}^{\infty}\binom{n}{x}p^x(1-p)^{n-x}a_n}
\end{aligned}
$$

and this sum can not be found analytically.

If we want to find an estimate for n we can use for example the mode, that is the n which has the largest posterior probability.

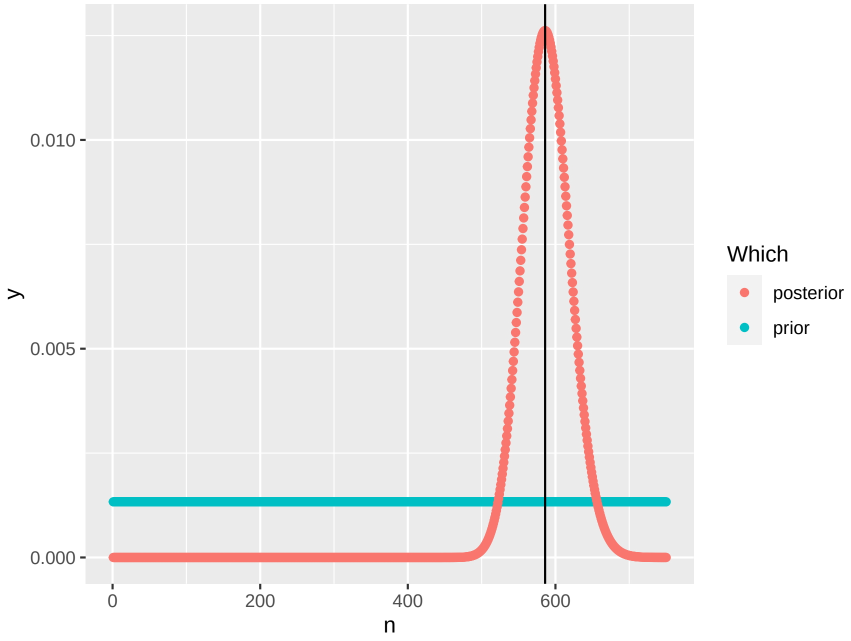
Here are some specific examples: say we observe x=217 and we know p=0.37. Also

- we know only that $n \leq 750$, so we choose

$a_i = 1/750$ if $1 \leq i \leq 750$, 0 otherwise,

The routine *bayes.bin.n* draws the prior and the posterior curve and finds the mode:

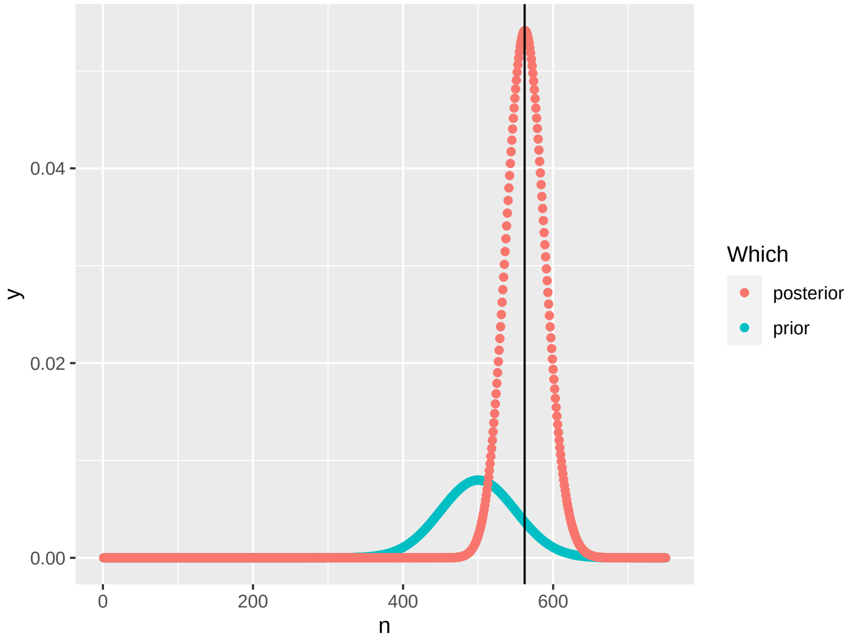
```
 bayes.bin.n <- function (x, p, a)
{
    N <-  length(a)
    q <-  1 - p
    f <-  rep(0, N)
    f[x:N] <- choose(x:N, x) * q^(c(x:N) - x) * a[x:N]
    mx <-  sum(choose(x:N + x, x) * q^c(x:N) * a[x:N])
    f <-  f/mx
    mode <- c(1:N)[which.max(f)]
    df <- data.frame(
            n=c(1:N, 1:N),
            y=c(a, f),
            Which=rep(c("prior", "posterior"), each=N))
    print(ggplot(data=df, aes(n, y, color=Which)) +
      geom_point() +
      geom_vline(xintercept = mode))
    mode
 }
bayes.bin.n(217, 0.37, rep(1/750,750))
```



```
## [1] 586
```

- we know n is most likely 500 with a standard deviation of 50

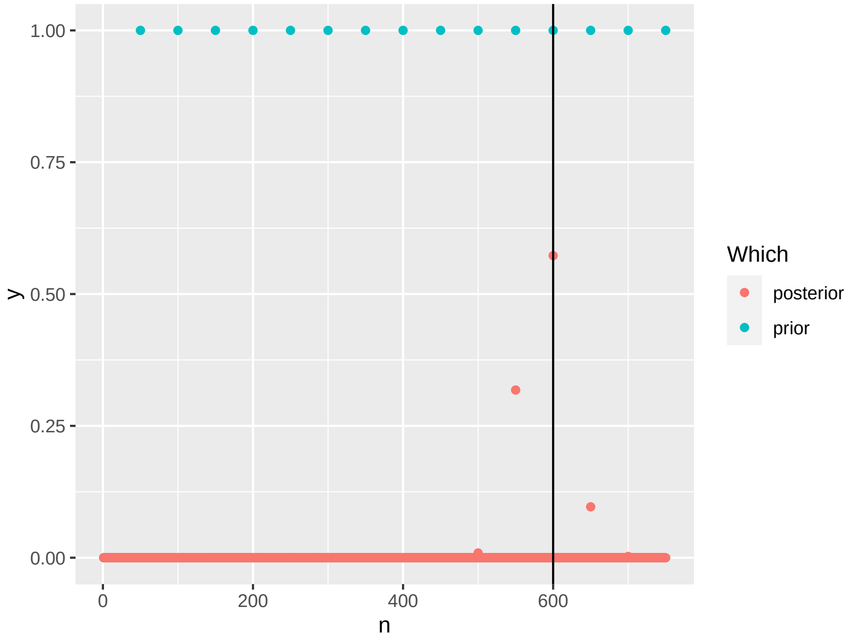
```
bayes.bin.n(217, 0.37, dnorm(1:750,500,50))
```



```
## [1] 562
```

- we know that n ≤ 750 and that n is a multiple of 50,

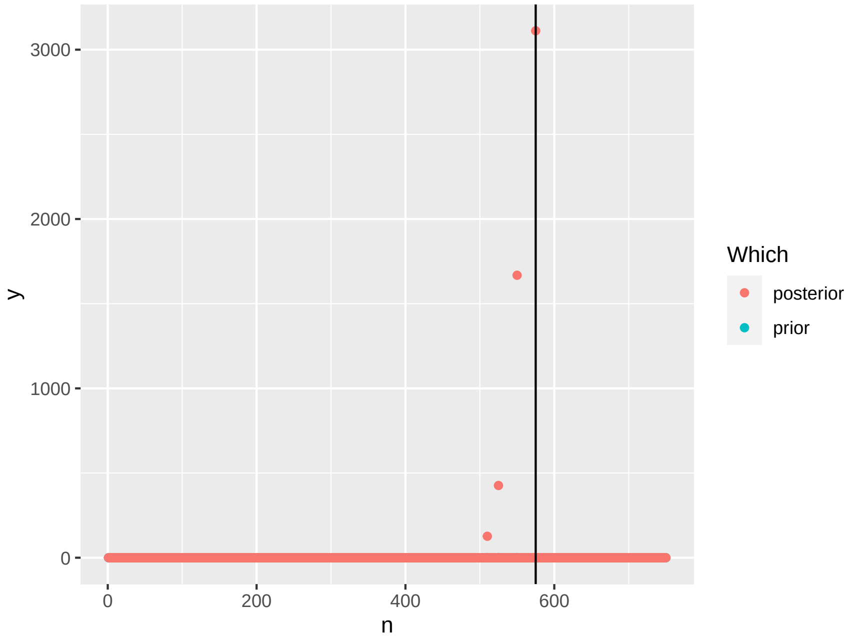
```
bayes.bin.n(217, 0.37, ifelse(c(1:750)%%50,0,1))
```



```
## [1] 600
```

- we know this was one of the four experiments we did, with

```
n <- c(510, 525, 550, 575)
a <- rep(0, 750)
a[n] <- 1
bayes.bin.n(217, 0.37, a)
```



```
## [1] 575
```

As we can see, the Bayesian method lets us include such knowledge in a very simple manner!

3.1.2 The Big Question: Bayesian or Frequentist?

Should you be a Bayesian?

Bayesian Statistics has a lot of good features. To begin with, it answers the right question, P(Hypothesis|Data). There are others as well:

- Decision Theory

There is a branch of mathematics concerned with decision making. It is conceptually a very useful and important one:

- Should you buy a new car, or keep the old one for another year?
- Should you invest your money into the stock market or buy fixed-interest bonds?

- Should the government lower the taxes or instead use the taxes for direct investments?

In decision theory one starts out by choosing a **loss function**, that is a function that assigns a value (maybe in terms of money) to every possible action and every possible outcome.

3.1.2.1 Example (3.1.3) You are offered the following game: you can either take $10 (let's call this action a), or you can flip a coin (action b). If the coin comes up heads you win $50, if it comes up heads you loose $10. So there are two possible actions: take the $10 or flip the coin, and three possible outcomes, you win $10, $50 or loose $10.

We need a value for each combination. One obvious answer is this one:

L(a)=10, L(b,"heads")=50, L(b,"tails")=-10

But there are other possibilities. Say you are in a bar. You already had food and drinks and your tab is $27. Now you notice that you only have $8 in your pocket (and no credit card etc.) Now if you win or loose $10 it doesn't matter, either way you can't pay your bill, and it will be very embarrassing when it comes to paying. But if you win $50, you are fine. Now your loss function might be:

L(a)=0, L(b,"heads")=1000, L(b,"tails")=0

The next piece in decision theory is the **decision function**. The idea is this: let's carry out an experiment, and depending on the outcome of the experiment we chose an action.

- Should you invest your money into the stock market or buy fixed-interest bonds?

Let's do this: we wait until tomorrow. If the Dow Jones goes up, we invest in the stock market, otherwise in bonds.

In decision theory a decision rule is called **inadmissible** if there is another rule that is better no matter what the outcome of the experiment. Obviously it makes no sense to pick an inadmissible rule.

So what's the connection to Bayesian Statistics? First there are **Bayesian decision rules**, which combine prior knowledge with the outcome of the experiment.

- based on the movement of the Dow Jones in the last year, I have a certain probability that it will go up over the next year.

Now there is a famous theorem (the complete class theorem) that says that all admissible rules are Bayesian decision rules for some prior.

Optimality

Obviously when we do something it would be nice to do it in an optimal (best) way. It turns out that in Bayesian statistics it is often possible to show that a certain method is best, better or at least as good as any other.

Should you be a Frequentist?

There are also arguments in favor of Frequentist statistics:

priors are bad

or better to say you don't like the subjectivity introduced by priors. In Bayesian statistics it is entirely possible that two Scientists who have the same data available and use the same method for analysis come to different conclusions, because they have different priors.

Frequentist methods work

For most of the history of Statistics, that is from about 1900 to about 1960, there was (essentially) only Frequentist Statistics. In this time (and since) many methods have been developed that worked very well in practice. Many of those turn out to be also Bayesian methods when the right prior is used, but not all!

3.1.2.2 Example (3.1.4) one of the most useful modern methods, called the Bootstrap, is a purely Frequentist method with no Bayesian theory. (Actually there is something called the Bayesian bootstrap, but it is not the same as the classical bootstrap)

3.1.2.3 Example (3.1.5) A standard technic in regression is to study the **residuals**. This, though, violates the likelihood principle and is therefore not allowed under the Bayesian paradigm. Actually, most Bayesians study the residuals anyway.

Frequentist methods are often fairly simple

Even for the easiest problems ("estimate the mean GPA of students at the Colegio") a Bayesian analysis always seems to be complicated (choose a prior and a loss function, calculate the posterior, extract the estimate from the posterior, try to do all of this optimally) Frequentist solutions are often quick and easy.

So? Be Both!

Often in any a specific problem, one approach just makes more sense than the other.

3.2 Priors

The main issue in a Bayesian analysis is always the choice of priors. Here are some of the common methods:

3.2.1 Subjective Prior

In many ways this is was is required by the Bayesian paradigm, namely to find a prior that "encodes" your subjective belief about the parameter.

3.2.1.1 Example (3.2.1) let's consider again the the coin example from before. Here is a very different, but probably more realistic, prior: Before we flip the coin we reason as follows: either the coin is fair, and we think that is most likely the case, or if it is not, we don't have any idea what it might be. We can "encode" this belief in the following prior:

Let $\delta_{1/2}$ be the point mass at 1/2, that is a random variable which always takes the value 1/2, or $P(\delta_{1/2} = 1/2) = 1$. Let $U \sim U[0, 1]$ and let $Z \sim Ber(\alpha)$. Now $p = Z\delta_{1/2} + (1 - Z)U$.

So with probability α p is just 1/2 (and the coin is fair), and with probability $1-\alpha$ p is uniform on [0,1] (and we don't have any idea what's going on).

This prior is a mixture of a discrete and a continuous r.v. so we will need to be a bit careful with the calculations.

The cdf and the prior density are as follows

$$\begin{aligned}
&F(p) = P(p \leq x) = \\
&P(p \leq x|Z=0)P(Z=0) + P(p \leq x|Z=1)P(Z=1) = \\
&\begin{cases} 0 \times \alpha + x(1-\alpha) & \text{if} \quad x < \frac{1}{2} \\ 1 \times \alpha + x(1-\alpha) & \text{if} \quad x \geq \frac{1}{2} \end{cases} = \\
&\begin{cases} x(1-\alpha) & \text{if} \quad x < \frac{1}{2} \\ \alpha + x(1-\alpha) & \text{if} \quad x \geq \frac{1}{2} \end{cases}
\end{aligned}$$

and so for $x \neq \frac{1}{2}$ $f(p) = 1-\alpha$ because then the cdf is a straight line with slope $1-\alpha$. Also

$$P(p = \frac{1}{2}) = \alpha + \frac{1}{2}(1-\alpha) - \frac{1}{2}(1-\alpha) = \alpha$$

The joint density of p and y is:

$$f(y,p) = f(y|p)\pi(p) = \begin{cases} (1-\alpha)\binom{n}{y}p^y(1-p)^{n-y} & \text{if} \quad p \neq \frac{1}{2} \\ \alpha\binom{n}{y}p^y(1-p)^{n-y} & \text{if} \quad p = \frac{1}{2} \end{cases}$$

and the marginal distribution of y:

$$\begin{aligned}
&m(y) = P(Y=y|Z=0)P(Z=0) + P(Y=y|Z=1)P(Z=1) = \\
&P(Y=y|p=\frac{1}{2})\alpha + P(Y=y|p \sim U[0,1])(1-\alpha) = \\
&\alpha\binom{n}{y}p^y(1-p)^{n-y} + \int_0^1 (1-\alpha)\binom{n}{y}p^y(1-p)^{n-y}dp = \\
&\alpha\binom{n}{y}(\frac{1}{2})^y(1-(\frac{1}{2}))^{n-y} + \\
&(1-\alpha)\binom{n}{y}\frac{\Gamma(y+1)\Gamma(n-y+1)}{\Gamma(n+2)}\int_0^1 \frac{\Gamma(n+2)}{\Gamma(y+1)\Gamma(n-y+1)}p^{(y+1)-1}(1-p)^{(n-y+1)-1}dp = \\
&\alpha\binom{n}{y}(\frac{1}{2})^n + (1-\alpha)\binom{n}{y}\frac{\Gamma(y+1)\Gamma(n-y+1)}{\Gamma(n+2)} = \\
&\alpha\binom{n}{y}(\frac{1}{2})^n + (1-\alpha)\frac{n!}{(n-y)!y!}\frac{y!(n-y)!}{(n+1)!} = \\
&\alpha\binom{n}{y}/2^n + \frac{1-\alpha}{n+1}
\end{aligned}$$

because the uniform is also a Beta(1,1) and we can use the result above.

Next we find the posterior distribution of p given y:

$$f(p|y) = \begin{cases} (1-\alpha)\binom{n}{y}p^y(1-p)^{n-y}/\left[\alpha\binom{n}{y}/2^n + \frac{1-\alpha}{n+1}\right] & \text{if} \quad p \neq \frac{1}{2} \\ \alpha\binom{n}{y}p^y(1-p)^{n-y}/\left[\alpha\binom{n}{y}/2^n + \frac{1-\alpha}{n+1}\right] & \text{if} \quad p = \frac{1}{2} \end{cases}$$

Finally as above we can find the mean of the posterior distribution to get an estimate of p for a given y. To simplify the notation let $K = \alpha\binom{n}{y}/2^n + \frac{1-\alpha}{n+1}$, and then

$$E[p|y] =$$

$$\frac{1}{2}\alpha\binom{n}{y}/2^n/K + \int_0^1 p(1-\alpha)\binom{n}{y}p^y(1-p)^{n-y}/Kdp =$$

$$\alpha\binom{n}{y}/2^{n+1}/K +$$

$$(1-\alpha)\binom{n}{y}\int_0^1 p^{y+1}(1-p)^{n-y}/Kdp =$$

$$\alpha\binom{n}{y}/2^{n+1}/K +$$

$$(1-\alpha)\binom{n}{y}\frac{\Gamma(y+2)\Gamma(n-y+1)}{\Gamma(n+3)}\int_0^1 \frac{\Gamma(n+3)}{\Gamma(y+2)\Gamma(n-y+1)}p^{(y+2)-1}(1-p)^{(n-y+1)-1}dp/K =$$

$$\alpha\binom{n}{y}/2^{n+1}/K + (1-\alpha)\binom{n}{y}\frac{\Gamma(y+2)\Gamma(n-y+1)}{\Gamma(n+3)}/K =$$

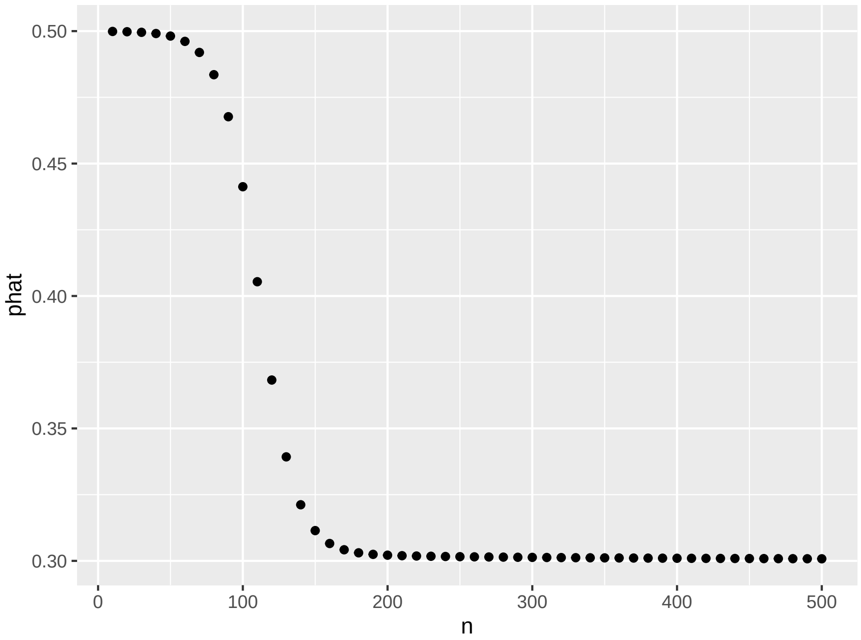
$$\left[\alpha\binom{n}{y}/2^{n+1} + (1-\alpha)\frac{n!}{(n-y)!y!}\frac{(y+1)!(n-y)!}{(n+2)!}\right]/K =$$

$$\left[\alpha\binom{n}{y}/2^{n+1} + (1-\alpha)\frac{y+1}{(n+2)(n+1)}\right] / \left[\alpha\binom{n}{y}/2^n + \frac{1-\alpha}{n+1}\right]$$

Again let's study the effect of the parameters in the posterior distribution:

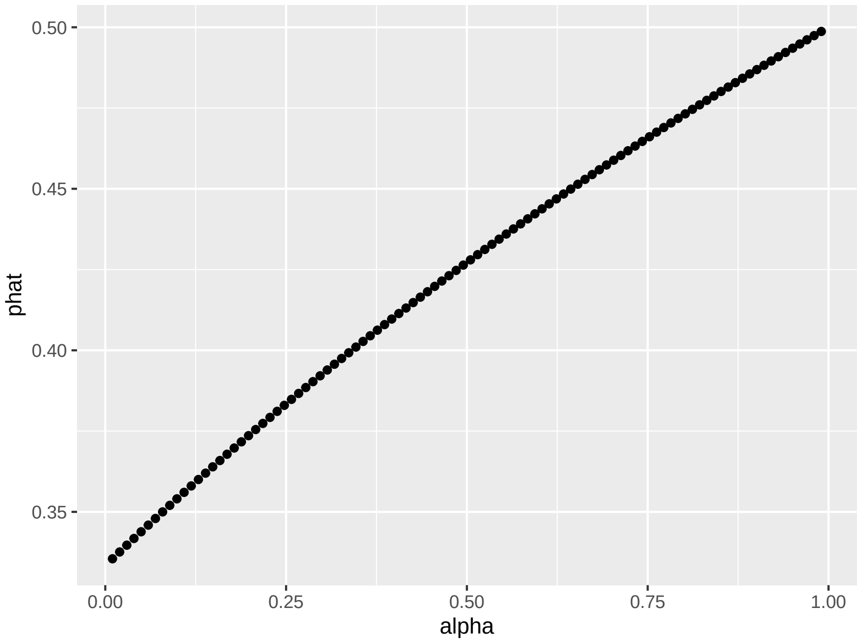
- effect of the sample size on the estimate of p

```
n <- 10*1:50
y <- 3*1:50
phat <- 0*n
alpha <- 0.999
for (i in 1:50)
  phat[i] <- (alpha*choose(n[i], y[i])/2^(n[i]+1) +
    (1-alpha)*(y[i]+1)/(n[i]+2)/(n[i]+1))/(alpha*choose(n[i], y[i])/2^n[i] + (1-alpha)/
ggplot(data.frame(n=n, phat=phat), aes(n, phat)) +
  geom_point()
```



- the effect of alpha on the estimate of p

```
alpha <- seq(0.01, 0.99, length = 100)
phat <- 0 * alpha
for (i in 1:100)
  phat[i] <- (alpha[i]*choose(10, 3)/2^(10 + 1) +
      (1-alpha[i])*(3+1)/(10+2)/(10+1))/(alpha[i]*
      choose(10, 3)/2^10 + (1 - alpha[i])/(10+1))
ggplot(data.frame(alpha=alpha, phat=phat), aes(alpha, phat)) +
  geom_point()
```



A larger alpha means a prior that puts more weight on the coin being fair.

Generally finding a good subjective prior can be done using **utility theory**.

3.2.2 Conjugate Priors

these are priors which together with the sampling distribution lead to a posterior distribution of the same type as the prior. We already saw one example:

3.2.2.1 Example (3.2.2) Binomial-Beta

Say $X_1, ..., X_n \sim Bin(n, p)$ and $p \sim Beta(\alpha, \beta)$, then

$$p|\boldsymbol{X} = \boldsymbol{x} \sim Beta(\alpha + \sum x_i, n - \sum x_i + \beta)$$

Here is another:

3.2.2.2 Example (3.2.3) Poisson-Gamma

Say $X_1, ..., X_n \sim Pois(\lambda)$ and $\lambda \sim Gamma(\alpha, \beta)$, then

$$\lambda|\boldsymbol{X} = \boldsymbol{x} \sim Gamma(\alpha + \sum x_i, \beta + n)$$

and of course we have

3.2.2.3 Example (3.2.4) Normal-Normal

$X_1, ..., X_n \sim N(\mu, \sigma)$ and $\mu \sim N(\tau, \theta)$, where σ, τ and θ are assumed to be known. Then from (2.3.2) we have

$$f(x_1, ..., x_n|\mu) = (2\pi\sigma^2)^{-n/2} \exp\left\{-\frac{1}{2\sigma^2}\sum(x_i-\mu)^2\right\}$$

and so the joint density is given by

$$f(x_1, ..., x_n, \mu) = (2\pi\sigma^2)^{-n/2} \exp\left\{-\frac{1}{2\sigma^2}\sum(x_i-\mu)^2\right\}(2\pi\theta^2)^{-1/2} \exp\left\{-\frac{1}{2\theta^2}\sum(\mu-\tau)^2\right\}$$

Now

$$\begin{aligned}
&\sum(x_i-\mu)^2 = \\
&\sum(x_i-\bar{x}+\bar{x}-\mu)^2 = \\
&\sum\left[(x_i-\bar{x})^2+2(x_i-\bar{x})(\bar{x}-\mu)+(\bar{x}-\mu)^2\right] = \\
&\sum(x_i-\bar{x})^2+2(\bar{x}-\mu)\sum(x_i-\bar{x})+n(\bar{x}-\mu)^2 = \\
&\sum(x_i-\bar{x})^2+2(\bar{x}-\mu)(\sum x_i-n\bar{x})+n(\bar{x}-\mu)^2 = \\
&\sum(x_i-\bar{x})^2+n(\bar{x}-\mu)^2
\end{aligned}$$

therefore

$$\begin{aligned}
&f(x_1, ..., x_n, \mu) = \\
&(2\pi\sigma^2)^{-n/2} \exp\left\{-\frac{1}{2\sigma^2}\left[\sum(x_i-\bar{x})^2+n(\bar{x}-\mu)^2\right]\right\}\frac{1}{\sqrt{2\pi\theta^2}}\exp\left\{-\frac{1}{2\theta^2}(\mu-\tau)^2\right\} = \\
&\exp\left\{-\frac{1}{2\sigma^2}\left[\sum(x_i-\bar{x})^2\right]\right\}(2\pi\sigma^2)^{-n/2}\frac{1}{\sqrt{2\pi\theta^2}}\exp\left\{-\frac{1}{2}\left[\frac{n(\bar{x}-\mu)^2}{\sigma^2}+\frac{(\mu-\tau)^2}{\theta^2}\right]\right\}
\end{aligned}$$

Now

$$T = \frac{n(\bar{x}-\mu)^2}{\sigma^2}+\frac{(\mu-\tau)^2}{\theta^2} = \frac{n\theta^2(\bar{x}-\mu)^2+\sigma^2(\mu-\tau)^2}{\sigma^2\theta^2}$$

Note that

$$\frac{n(\bar{x}-\mu)^2}{\sigma^2}+\frac{(\mu-\tau)^2}{\theta^2} = \frac{n\theta^2(\bar{x}-\mu)^2+\sigma^2(\mu-\tau)^2}{\sigma^2\theta^2}$$

In the numerator we have

$$\begin{aligned}
&n\theta^2(\bar{x}-\mu)^2+\sigma^2(\mu-\tau)^2 = \\
&n\theta^2(\bar{x}^2-2\bar{x}\mu+\mu^2)+\sigma^2(\mu^2-\mu\tau+\tau^2) = \\
&(n\theta^2+\sigma^2)\mu^2-2(n\theta^2\bar{x}^2+\sigma^2\tau)\mu+n\theta^2\bar{x}^2+\sigma^2\tau^2 = \\
&(n\theta^2+\sigma^2)\left\{\mu^2-2\frac{n\theta^2\bar{x}^2+\sigma^2\tau}{n\theta^2+\sigma^2}\mu\right\}+n\theta^2\bar{x}^2+\sigma^2\tau^2 = \\
&(n\theta^2+\sigma^2)\left\{\mu^2-2\frac{n\theta^2\bar{x}^2+\sigma^2\tau}{n\theta^2+\sigma^2}\mu+\left(\frac{n\theta^2\bar{x}^2+\sigma^2\tau}{n\theta^2+\sigma^2}\right)^2-\left(\frac{n\theta^2\bar{x}^2+\sigma^2\tau}{n\theta^2+\sigma^2}\right)^2\right\} \\
&\ +n\theta^2\bar{x}^2+\sigma^2\tau^2 = \\
&(n\theta^2+\sigma^2)\left\{\mu-\frac{n\theta^2\bar{x}^2+\sigma^2\tau}{n\theta^2+\sigma^2}\right\}^2-\frac{(n\theta^2\bar{x}^2+\sigma^2\tau)^2}{n\theta^2+\sigma^2}+n\theta^2\bar{x}^2+\sigma^2\tau^2
\end{aligned}$$

and so

$$\begin{aligned}
&T = \\
&\left\{(n\theta^2+\sigma^2)\left\{\mu-\frac{n\theta^2\bar{x}^2+\sigma^2\tau}{n\theta^2+\sigma^2}\right\}^2-\frac{(n\theta^2\bar{x}^2+\sigma^2\tau)^2}{n\theta^2+\sigma^2}+n\theta^2\bar{x}^2+\sigma^2\tau^2\right\}/(\sigma^2\theta^2) \\
&(n/\sigma^2+1/\theta^2)\left\{\mu-\frac{n\theta^2\bar{x}^2+\sigma^2\tau}{n\theta^2+\sigma^2}\right\}^2-\frac{(n\theta^2\bar{x}^2+\sigma^2\tau)^2}{(n\theta^2+\sigma^2)\sigma^2\theta^2}+n\bar{x}^2/\sigma^2+\tau^2/\theta^2
\end{aligned}$$

Let's define

$$\begin{aligned}
\mu_1 = \frac{n\theta^2\bar{x}^2+\sigma^2\tau}{n\theta^2+\sigma^2} &= \\
\frac{n\bar{x}^2/\sigma^2+\tau/\theta^2}{(n\theta^2+\sigma^2)/(\sigma^2\theta^2)} &= \\
\frac{n\bar{x}^2/\sigma^2+\tau/\theta^2}{\frac{n}{\sigma^2}+\frac{1}{\theta^2}}
\end{aligned}$$

and

$$\sigma_1^2 = \frac{\sigma^2\theta^2}{n\theta^2+\sigma^2} = \frac{1}{\frac{n}{\sigma^2}+\frac{1}{\theta^2}}$$

and so

$$
\begin{aligned}
&T = \\
&(n/\sigma^2+1/\theta^2)\left\{\mu-\frac{n\theta^2\bar{x}^2+\sigma^2\tau}{n\theta^2+\sigma^2}\right\}^2-\frac{(n\theta^2\bar{x}^2+\sigma^2\tau)^2}{(n\theta^2+\sigma^2)\sigma^2\theta^2}+n\bar{x}^2/\sigma^2+\tau^2/\theta^2 = \\
&\left(\frac{\mu-\mu_1}{\sigma_1}\right)^2-\left(\frac{n\theta^2\bar{x}^2+\sigma^2\tau}{n\theta^2+\sigma^2}\right)^2\frac{n\theta^2+\sigma^2}{\sigma^2\theta^2}+n\bar{x}^2/\sigma^2+\tau^2/\theta^2 = \\
&\left(\frac{\mu-\mu_1}{\sigma_1}\right)^2-\mu_1/\sigma_1^2+n\bar{x}^2/\sigma^2+\tau^2/\theta^2
\end{aligned}
$$

Now we put it all together:

$$
\begin{aligned}
&f(x_1,...,x_n,\mu) = \\
&\exp\left\{-\frac{1}{2\sigma^2}\left[\sum(x_i-\bar{x})^2\right]\right\}\times \\
&(2\pi\sigma^2)^{-n/2}\frac{1}{\sqrt{2\pi\theta^2}}\exp\left\{-\frac{1}{2}\left[\frac{n(\bar{x}-\mu)^2}{\sigma^2}+\frac{(\mu-\tau)^2}{\theta^2}\right]\right\} = \\
&\exp\left\{-\frac{1}{2\sigma^2}\left[\sum(x_i-\bar{x})^2\right]\right\}\times \\
&(2\pi\sigma^2)^{-n/2}\frac{1}{\sqrt{2\pi\theta^2}}\exp\left\{-\frac{1}{2}\left[\left(\frac{\mu-\mu_1}{\sigma_1}\right)^2-\mu_1/\sigma_1^2+n\bar{x}^2/\sigma^2+\tau^2/\theta^2\right]\right\} = \\
&\exp\left\{-\frac{1}{2\sigma^2}\left[\sum(x_i-\bar{x})^2-\mu_1\sigma^2/\sigma_1^2+n\bar{x}^2+\sigma^2\tau^2/\theta^2\right]\right\}(2\pi\sigma^2)^{-n/2}\frac{1}{\sqrt{2\pi\theta^2}}\sqrt{2\pi\sigma_1^2} \\
&\frac{1}{\sqrt{2\pi\sigma_1^2}}\exp\left\{-\frac{1}{2}\left(\frac{\mu-\mu_1}{\sigma_1}\right)^2\right\}
\end{aligned}
$$

$$
\begin{aligned}
&m(\boldsymbol{x}) = \\
&\exp\left\{-\frac{1}{2\sigma^2}\left[\sum(x_i-\bar{x})^2-\mu_1\sigma^2/\sigma_1^2+n\bar{x}^2+\sigma^2\tau^2/\theta^2\right]\right\}(2\pi\sigma^2)^{-n/2}\frac{1}{\sqrt{2\pi\theta^2}}\sqrt{2\pi\sigma_1^2} \\
&f(\mu|\boldsymbol{x}) = \frac{f(\boldsymbol{x},\mu)}{m(\boldsymbol{x})} = \frac{1}{\sqrt{2\pi\sigma_1^2}}\exp\left\{-\frac{1}{2}\left(\frac{\mu-\mu_1}{\sigma_1}\right)^2\right\}
\end{aligned}
$$

and so the posterior distribution is a $N(\mu_1,\sigma_1)$

3.2.3 Non-informative Priors

just what it says, a prior that does not contain any "information" on the parameter.

3.2.3.1 Example (3.2.5) X_1, .., X_n~Ber(p), then p~U[0,1] is a non-informative prior.

3.2.3.2 Example (3.2.6)

$X_1, .., X_n$ $N(\mu, \sigma)$, σ known. Now μ can be any real number, so a prior has to be a density on the whole real line. But $\pi(\mu) = c$ is not possible because it integrates out to infinity for any c>0!

There are two solutions to this problem:

- Allow **improper** priors, that is priors with an infinite integral. This is generally ok as long as the posterior is a proper density:

$$
\begin{aligned}
&f(\boldsymbol{x}|\mu) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x_i-\mu)^2}{2\sigma^2}\right\} = \\
&(2\pi\sigma^2)^{-n/2} \exp\left\{-\frac{\sum(x_i-\mu)^2}{2\sigma^2}\right\} = \\
&(2\pi\sigma^2)^{-n/2} \exp\left\{-\frac{\sum(x_i-\bar{x})^2 + n(\mu-\bar{x})^2}{2\sigma^2}\right\} = \\
&(2\pi\sigma^2)^{-(n-1)/2}/\sqrt{n} \exp\left\{-\frac{\sum(x_i-\bar{x})^2}{2\sigma^2}\right\} \frac{1}{\sqrt{2\pi\sigma^2/n}} \exp\left\{-\frac{(\mu-\bar{x})^2}{2\sigma^2/n}\right\}
\end{aligned}
$$

where we used (2.2.4). Next we find the marginal:

$$
\begin{aligned}
&m(\boldsymbol{x}) = \int_{-\infty}^{\infty} f(\boldsymbol{x}|\mu)\pi(\mu)d\mu = \\
&(2\pi\sigma^2)^{-(n-1)/2}/\sqrt{n} \exp\left\{-\frac{\sum(x_i-\bar{x})^2}{2\sigma^2}\right\} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2/n}} \exp\left\{-\frac{(\mu-\bar{x})^2}{2\sigma^2/n}\right\} d\mu = \\
&(2\pi\sigma^2)^{-(n-1)/2}/\sqrt{n} \exp\left\{-\frac{\sum(x_i-\bar{x})^2}{2\sigma^2}\right\}
\end{aligned}
$$

because the integral is over a normal density and therefore equal to 1. Finally we find the posterior distribution:

$$
\begin{aligned}
&f(\mu|\boldsymbol{x}) = \frac{f(\boldsymbol{x}|\mu)\pi(\mu)}{m(\boldsymbol{x})} \\
&\frac{(2\pi\sigma^2)^{-(n-1)/2}/\sqrt{n} \exp\left\{-\frac{\sum(x_i-\bar{x})^2}{2\sigma^2}\right\} \frac{1}{\sqrt{2\pi\sigma^2/n}} \exp\left\{-\frac{(\mu-\bar{x})^2}{2\sigma^2/n}\right\}}{(2\pi\sigma^2)^{-(n-1)/2}/\sqrt{n} \exp\left\{-\frac{\sum(x_i-\bar{x})^2}{2\sigma^2}\right\}} = \\
&\frac{1}{\sqrt{2\pi\sigma^2/n}} \exp\left\{-\frac{(\mu-\bar{x})^2}{2\sigma^2/n}\right\}
\end{aligned}
$$

and so we find $\mu|\boldsymbol{X}=\boldsymbol{x} \sim N(\bar{x}, \sigma/\sqrt{n})$

One justification for this is that we usually can express the improper prior as the limit of a sequence of proper priors: we saw already that if $X \sim N(\mu, \sigma)$ and $\mu \sim N(\tau, \theta)$, then $\mu|\boldsymbol{X}=\boldsymbol{x} \sim N(\mu_1, \sigma_1)$ and

$$\mu_1 = \frac{n\bar{x}/\sigma^2 + \tau/\theta^2}{\frac{n}{\sigma^2} + \frac{1}{\theta^2}} \rightarrow \bar{x}$$

$$\sigma_1^2 = \frac{1}{\frac{n}{\sigma^2} + \frac{1}{\theta^2}} \rightarrow \sigma^2/n$$

- A second solution is to think a little more about what “non-informative” really means.

3.2.3.3 Example (3.2.7) Say we have $X_1, .., X_n \sim N(\mu, \sigma)$ and we want to estimate σ. At first it seems we should use $\pi(\sigma) = 1$, $\sigma > 0$. It turns out, though, that this is not really “completely non-informative” because of the following: say we estimate $\sigma = 2.7$, then there is small interval $(0, 2.7)$ “below” our estimate but a very large interval $(2.7, \infty)$ “above” it.

There is a class of priors that were developed explicitly to express this idea of “complete lack of knowledge” called Jeffrey’s priors:

Let’s say $X_1, .., X_n \sim f(x|\theta)$. Jeffrey considered one-to-one transformations of the parameter $\delta = u(\theta)$. From our discussions in probability we know that the prior density of δ is given by

$$f(\delta) = f(\theta)|u^{-1}(\theta)|'$$

Jeffrey argued that true lack of knowledge means that any transformation of the parameter should yield the equivalent result. Essentially it should not matter whether we measure temperature in Fahrenheit or Centigrade. He showed that this implies that the prior should be proportional to the square root of the **Fisher information**:

$$\pi(\theta) \propto \sqrt{J(\theta)}$$

where

$$J(\theta) = -E\left[\frac{d^2 \log f(x|\theta)}{d\theta^2}\right]$$

here this means

$$
\begin{aligned}
&\log f(x|\sigma) = \log\left[\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}\right] = \\
&-\frac{1}{2}\log(2\pi) - 2\log\sigma - \frac{1}{2\sigma^2}(x-\mu)^2 \\
&\frac{d\log f(x|\theta)}{d\theta} = -\frac{2}{\sigma} + \frac{1}{\sigma^3}(x-\mu)^2 \\
&\frac{d^2\log f(x|\theta)}{d\theta^2} = \frac{2}{\sigma^2} - \frac{3}{\sigma^4}(x-\mu)^2 \\
&E\left[\frac{d^2\log f(x|\theta)}{d\theta^2}\right] = \\
&E\left[\frac{2}{\sigma^2} - \frac{3}{\sigma^4}(X-\mu)^2\right] = \\
&\frac{2}{\sigma^2} - \frac{3}{\sigma^4}E[(X-\mu)^2] = \\
&\frac{2}{\sigma^2} - \frac{3}{\sigma^4}\sigma^2 = -1/\sigma^2 \\
&J(\sigma) = -E\left[\frac{d^2\log f(x|\theta)}{d\sigma^2}\right] = 1/\sigma^2
\end{aligned}
$$

and so Jeffrey's prior for σ is $\sqrt{1/\sigma^2} = 1/\sigma$. Note that this is an improper prior.

3.2.4 Empirical Bayes

3.2.4.1 Example (3.2.8) say $X_1, .., X_n \sim Pois(\lambda)$ and $\lambda \sim Gamma(\alpha, \beta)$, then we know that

$$\lambda|X = x \sim Gamma(\alpha + \sum x_i, \beta + n)$$

But how do we choose α and β? In a subjective Bayesian analysis we would need to use prior knowledge to estimate them. The idea of empirical Bayes is to use the data itself to estimate the "hyper-parameters" α and β. For example, we know that the mean of a Gamma(α,β) is $\alpha\beta$ and the variance is $\alpha\beta^2$. So we choose α and β as the solutions of the system of non-linear equations

$\bar{X} = \alpha\beta$
$s^2 = \alpha\beta^2$

or

$s^2 = \alpha\beta \times \beta = \bar{X} \times \beta$

so

$\beta = s^2/\bar{X}$ and $\alpha = \bar{X}/\beta = \bar{X}^2/s^2$.

One can go even further and use the empirical distribution function as the cdf of the prior itself. In many ways, though, empirical Bayes goes against the spirit of Bayesian analysis. It also violates the likelihood principle!

These are just some of the methods for finding priors, there are many others.

The situation becomes much more difficult in multi-dimensional problems, and there is a lot of disagreement among Bayesians what constitutes a good prior there.

3.2.5 Calculation Issues

In most cases actual calculations have to be done using numerical methods and/or simulation.

Numerical Solutions

3.2.5.1 Example (3.2.9) the following is a data set from a Beta distribution with parameters α and $\beta = 2.5$:

0.62	0.62	0.34	0.58	0.50	0.46	0.64	0.40	0.44	0.29
0.76	0.45	0.49	0.55	0.76	0.75	0.77	0.19	0.58	0.65
0.30	0.50	0.42	0.46	0.36	0.68	0.07	0.67	0.48	0.17
0.03	0.36	0.76	0.30	0.12	0.22	0.29	0.33	0.89	0.78
0.29	0.09	0.33	0.03	0.45	0.30	0.86	0.44	0.49	0.06

We want to find an estimate for α using as a prior $\alpha \sim N(1.5, 1/4)$ and the median of the posterior distribution. The marginal turns out to be

$$m(\boldsymbol{x}) = \int_0^\infty \left(\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\right)^n \left(\prod x_i\right)^{\alpha-1} \left(\prod (1-x_i)\right)^{\beta-1} \frac{1}{\sqrt{\pi/8}} \exp\left\{-8(\alpha-1.5)^2\right\} d\alpha$$

clearly there is no way to find this integral analytically, so we will do it numerically:

```
find.alpha <- function (data)
{
  marginal <- 1
  integrant <- function(t) {
  y <- length(t)
  for(i in 1:length(t))
    y[i] <- prod(dbeta(data,t[i], 2.5))*
            dnorm(t[i], 1.5, 0.25)/marginal
    y
  }
  marginal <- integrate(integrant,0,4)$value
  step <- c(1, 0.1 ,0.01)
  a<-0
  for(i in 1:3) {
      repeat {
          a <- a+step[i]
          I <- integrate(integrant,0,a)$value
          print(c(a,I))
          if(I>0.5) break
      }
      a <- a-step[i]
    }
    a
}
find.alpha(beta.data)
```

```
## [1] 1.000000e+00 1.108583e-07
## [1] 2.0000000 0.9560108
## [1] 1.100000e+00 5.531253e-06
## [1] 1.2000000000 0.0001362987
## [1] 1.300000000 0.001806294
## [1] 1.40000000 0.01382355
## [1] 1.50000000 0.06505595
## [1] 1.6000000 0.1998564
## [1] 1.7000000 0.4262885
## [1] 1.8000000 0.6760933
## [1] 1.7100000 0.4518548
## [1] 1.7200000 0.4775687
## [1] 1.730000 0.503323
```

```
## [1] 1.72
```

Simulation

Because analytic solutions are only possible in simple problems and numerical solutions are difficult in problems with more than 3 or 4 parameters the most common solution today is to simulate data from the posterior distribution.

This is possible because using a methodology called Markov Chain Monte Carlo (MCMC) it is possible to sample from the posterior without having to know any constants, that is without having to find the marginal $\mathrm{m}(\boldsymbol{x})$.

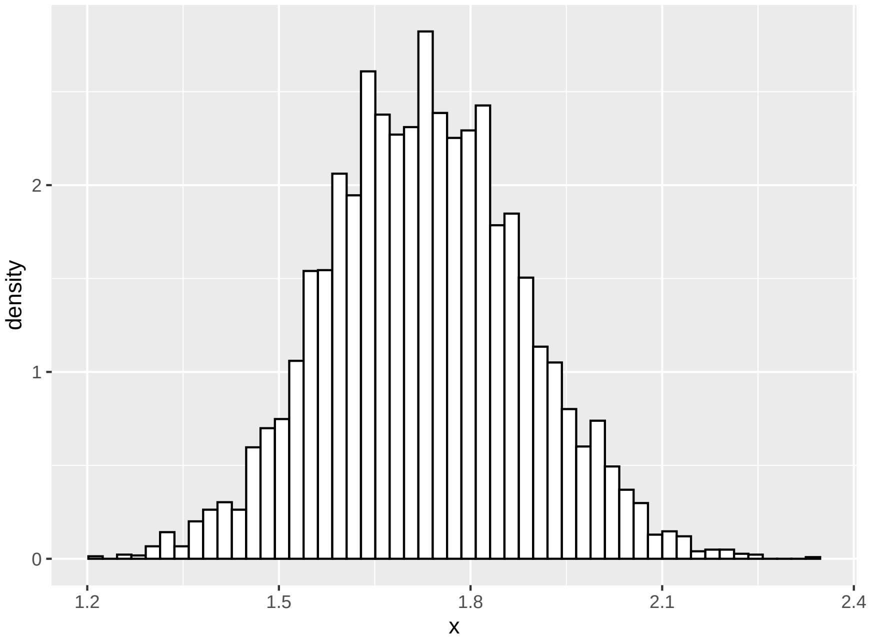
3.2.5.2 Example (3.2.10)

Let's do the example above again. The routine **betaMCMC()** generates data from the posterior and finds the sample median. Notice that in the definition of the posterior we just use the joint density, without the $\mathrm{m}(\boldsymbol{x})$.

```
betaMCMC <- function (x, just.alpha=TRUE, B = 10000) {
  if(just.alpha)
    posterior <- function(a, b)
      prod(dbeta(x, a, b)) * dnorm(a, 1.5, 0.25)
  else
    posterior <- function(a, b)
```

```
      prod(dbeta(x, a, b)) *
      dnorm(a, 1.5, 0.25) * dnorm(b, 2.5, 0.5)

  A <- matrix(0, B, 2)
  A[1, ] <- c(1.7, 2.5)
  for (i in 2:B) {
    u <- runif(1, max(0, A[i-1, 1]-0.5), A[i-1, 1]+0.5)
    if(!just.alpha)
      v <- runif(1, max(0, A[i-1, 2]-0.5), A[i-1, 2]+0.5)
    else v <- 2.5
    if (runif(1) < posterior(u, v)/posterior(A[i-1, 1],
                                   A[i-1, 2]))
            A[i, ] <- c(u, v)
    else A[i, ] <- A[i - 1, ]
  }
  bw <- diff(range(A[, 1]))/50
  plot(ggplot(data.frame(x=A[, 1]), aes(x)) +
      geom_histogram(aes(y = ..density..),
        color = "black",
        fill = "white",
        binwidth = bw))
  if(just.alpha) return(median(A[, 1]))
  bw <- diff(range(A[, 2]))/50
  plot(ggplot(data.frame(x=A[, 2]), aes(x)) +
      geom_histogram(aes(y = ..density..),
        color = "black",
        fill = "white",
        binwidth = bw))
  apply(A, 2, median)
}
```

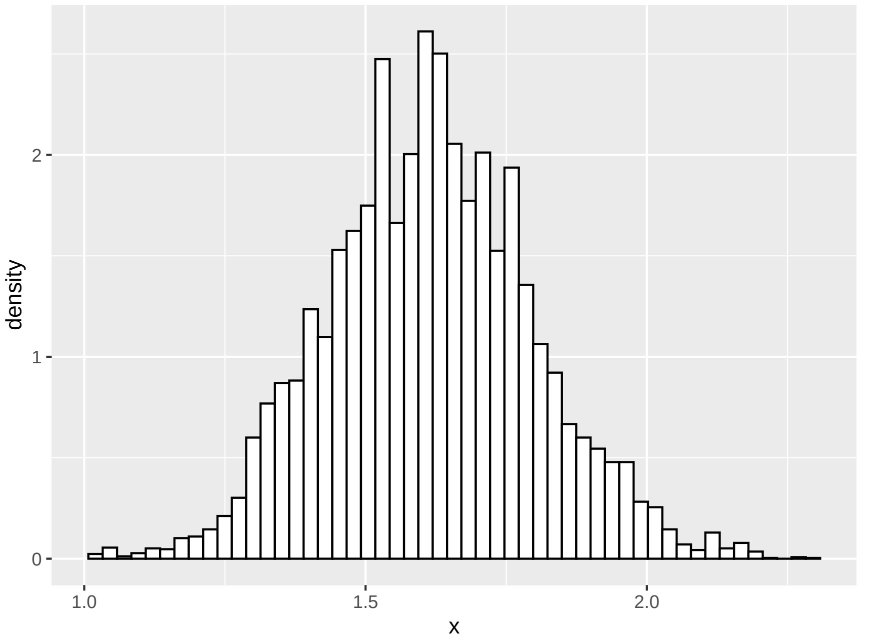
```
betaMCMC(beta.data)
```

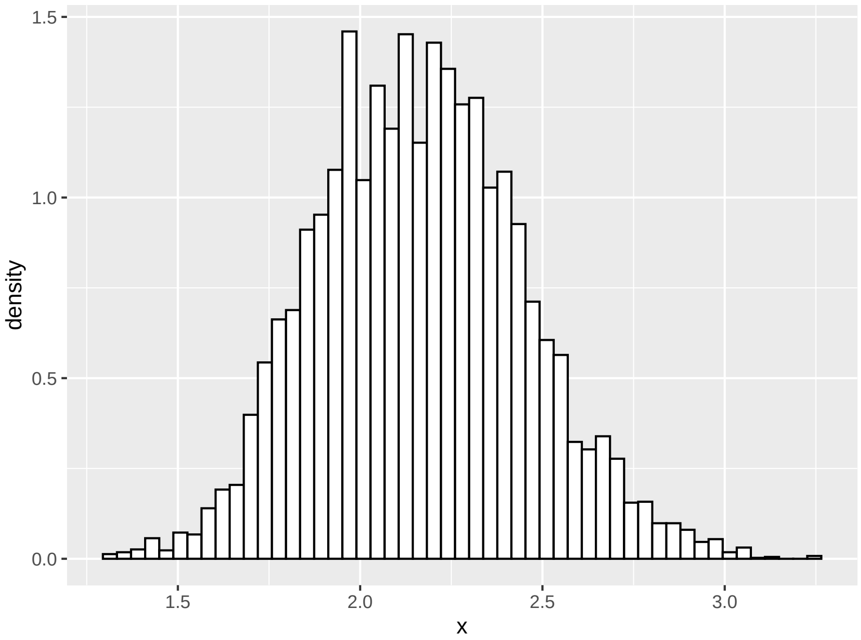


```
## [1] 1.727399
```

It is fairy simple to include β as a free parameter as well. As a prior for β we will use N(2.5,0.5)

```
betaMCMC(beta.data, just.alpha = FALSE)
```





```
## [1] 1.612542 2.159380
```

Want to know why this generates data from the posterior? Come to ESMA 5015 Simulation!

4 Parameter Estimation

4.1 Properties of Estimators

We will use the following notation: X has density $f(x|\theta)$ indicates that the density depends on a parameter θ (which could be a vector). For example $\theta = (\mu, \sigma)$ for the normal.

4.1.0.1 Definition (4.1.1)

Say we have $\boldsymbol{X} = (X_1, ..., X_n)$ with density $f(x|\theta)$. Then any function of the data $T(\boldsymbol{X}) = T(X_1, ..., X_n)$is called a **statistic**. If it is meant to estimate θ it is called an estimator of θ.

We will now discuss a number of properties of estimators. All these properties are equally important for Bayesians and Frequentists.

4.1.1 Unbiased Estimators

An estimator T is called **unbiased** for θ if

$$E[T(X_1, ..., X_n)] = \theta$$

4.1.1.1 Example (4.1.2) say $X_1, ..., X_n \sim U[0, \theta]$. Find an unbiased estimator of θ.

We will consider two possible estimators, one based on the sample mean and another based on the maximum:

1. $E[\bar{X}] = E[X_1] = \theta/2$

Let $T_1(\boldsymbol{x}) = 2\bar{x}$, then $E[T_1] = \theta$, so T_1 is unbiased.

2. Consider $T_2 = \max\{X_i\}$. Recall that $X_i/\theta \sim U[0, 1]$, and so $T_2/\theta \sim Beta(n, 1)$, and so

$$E[T_2] = \theta E[T_2/\theta] = \theta\frac{n}{n+1}$$

and so $\frac{n+1}{n}T_2$ is unbiased.

4.1.2 Bias

the *bias* of an estimator is defined by

$$\text{bias}(T) = E[T] - \theta$$

4.1.2.1 Example (4.1.3) say $X_1, ..., X_n \sim U[0, \theta]$, then

$$\begin{aligned}&\text{bias}(T_2) = ET_2 - \theta = \\ &\theta\frac{n}{n+1} - \theta = -\frac{\theta}{n+1}\end{aligned}$$

4.1.3 Mean Square Error

4.1.3.1 Definition (4.1.4) The *mean square error* of an estimator is defined by

$$\text{MSE}(\theta) = E\left[||T(\boldsymbol{X}) - \theta||^2\right]$$

where $||.||$ is some norm.

4.1.3.2 Example (4.1.5) $X_1, ..X_n \sim N(\mu, 1)$ and $T(\boldsymbol{x}) = \bar{x}$. Then

$$\begin{aligned}&E\left[||T(\boldsymbol{X}) - \theta||^2\right] = \\ &E\left[|\bar{X} - \mu|^2\right] = \\ &E\left[(\bar{X} - \mu)^2\right] = \\ &var(\bar{X}) = 1/n\end{aligned}$$

4.1.3.3 Example (4.1.6) $X_1, ..., X_n \sim U[0,\theta]$

$$
\begin{aligned}
&E\left[||T_2-\theta||^2\right] = \\
&E\left[(T_2-\frac{n\theta}{n+1}+\frac{n\theta}{n+1}-\theta)^2\right] = \\
&E(T_2-\frac{n\theta}{n+1})^2+2(ET_2-\frac{n\theta}{n+1})(\frac{n\theta}{n+1}-\theta)+(\frac{n\theta}{n+1}-\theta)^2 = \\
&var(T_2)+0+(E[T_2]-\theta)^2 = \\
&var(T_2)+\text{bias}(T_2)^2
\end{aligned}
$$

It turns out to be quite true in general that

$$\text{MSE}(T) = var(T) + \text{bias}(T)^2$$

Because the mean square error combines both variance and bias is seems natural that a good estimator should have a small mean square error, and in fact many estimation methods attempt to minimize the mean square error.

From the formula it is clear that a small bias might be acceptable if it also has a small variance, because then we get a small mean square error.

This is often referred to as the **bias-variance trade-off**.

4.1.4 Stein's phenomenon, James-Stein estimator

Say we have $X_1, ..X_n \sim N(\mu, \sigma)$, σ known and $T(\boldsymbol{X}) = \bar{X}$. Then it can be shown that T minimizes the mean square error.

Let's consider the multidimensional version of this problem:

$X_1, ..X_n \sim N(\mu, \sigma\boldsymbol{I})$, where $\mu \in R^d$, σ is known and $\boldsymbol{I}$ is the identity matrix. We want to estimate μ.

Because the variance-covariance matrix is a diagonal matrix the covariances are zero, and so the X_i are independent. It seems therefore natural that the estimator corresponding to $\bar{x}$ should be good estimator. It came therefore as shock to the Statisticians when in 1956 Stain showed that if d>2, this estimator is *inadmissible*, that is no matter what n, μ and σ are, there exists an estimator with lower mean square error.

An example is the *James-Stein estimator*, given by

$$\hat{\mu}_{JS} = \left(1 - \frac{(d-2)\sigma^2}{||x||^2}\right)x$$

Notice that it if $(d-2)\sigma^2 < ||x||^2$, this shrinks the estimator towards 0. For this reason this is an example of a class of estimators called *shrinkage estimators*.

A consequence of the above discussion is the following counter intuitive result: When three or more unrelated parameters are measured, their total mean square error can be reduced by using a combined estimator such as the James–Stein estimator; whereas when each parameter is estimated separately, the least squares (LS) estimator is admissible.

An example would be estimating the age of the Universe, the GPA of the undergraduate students at the Colegio, and the average beer consumption in Germany, all together!

4.1.5 Sufficient Statistics

4.1.5.1 Definition (4.1.7) A statistic T is a **sufficient statistic** for θ if the conditional distribution of the sample X given the value of T(X) does not depend on θ

The meaning of "sufficient statistic" is that all the information about the parameter θ is contained in T, so any inference about θ (such as an estimator, a hypothesis test or a confidence interval) can be based on T.

4.1.5.2 Theorem (4.1.8) If $f(\boldsymbol{x}|\theta)$ is the joint density of $\mathbf{X}$ and $q(t|\theta)$ is the density of T($\mathbf{X}$), then T($\mathbf{X}$) is a sufficient statistic for θ if, for every $\boldsymbol{x}$ in the sample space the ratio

$$f(\boldsymbol{x}|\theta)/q(t|\theta)$$

is constant as a function of θ.

4.1.5.3 Example (4.1.9) say $X_1, ..., X_n \sim Ber(p)$ (Here $\theta = p$). Let $T(\boldsymbol{x}) = \sum x_i$. T is the number of "successes" in n independent Bernoulli trials, and so $T \sim Bin(n, p)$. Now

$$f(\boldsymbol{x}|p) = \prod_{i=1}^{n} \left(p^{x_i}(1-p)^{1-x_i}\right) =$$

$$p^{\sum_{i=1}^{n} x_i}(1-p)^{n-\sum_{i=1}^{n} x_i} = p^{T(\boldsymbol{x})}(1-p)^{n-T(\boldsymbol{x})}$$

and

$$f(\boldsymbol{x}|\theta)/q(t|\theta) = \frac{p^{T(\boldsymbol{x})}(1-p)^{n-T(\boldsymbol{x})}}{\binom{n}{T(\boldsymbol{x})}p^{T(\boldsymbol{x})}(1-p)^{n-T(\boldsymbol{x})}} = \frac{1}{T(\boldsymbol{x})}$$

We see that the ratio is a constant with respect to p and so T is a sufficient statistic for p.

4.1.5.4 Theorem (4.1.10) **Factorization Theorem**

Let $f(\boldsymbol{x}|\theta)$ be the joint density of $\mathbf{X}$. A statistic T($\mathbf{X}$) is a sufficient statistic for θ if and only if there exist functions $g(t|\theta)$ and $h(\boldsymbol{x})$ such that for every $\boldsymbol{x}$ in the sample space and all values of the parameter we have

$$f(\boldsymbol{x}|\theta) = g(t|\theta)h(\boldsymbol{x})$$

4.1.5.5 Example (4.1.11)

say $X_1, ..., X_n \sim N(\mu, \sigma)$ and we assume that σ is known, so $\theta = \mu$. Then

$$
\begin{aligned}
& f(\boldsymbol{x}|\mu) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}\left(x_i-\mu\right)^2\right\} = \\
& (2\pi\sigma^2)^{-n/2} \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^n \left(x_i-\mu\right)^2\right\} = \\
& (2\pi\sigma^2)^{-n/2} \exp\left\{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^n \left(x_i-\bar{x}\right)^2 + n\left(\mu-\bar{x}\right)^2\right]\right\} = \\
& (2\pi\sigma^2)^{-n/2} \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^n \left(x_i-\bar{x}\right)^2\right\} \exp\left\{-\frac{n}{2\sigma^2}\left(\mu-\bar{x}\right)^2\right\} = \\
& h(\boldsymbol{x})g(t|\mu)
\end{aligned}
$$

where

$$
\begin{aligned}
& t = \bar{x} \\
& h(\boldsymbol{x}) = (2\pi\sigma^2)^{-n/2} \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^n \left(x_i-t\right)^2\right\} \\
& g(t|\mu) = \exp\left\{-\frac{n}{2\sigma^2}\left(\mu-t\right)^2\right\} = \\
& h(\boldsymbol{x})g(t|\lambda)
\end{aligned}
$$

and we see that the sample mean is a sufficient statistic for the population mean μ, at least if the variance is known.

4.1.5.6 Example (4.1.12)

say $X_1, ..., X_n \sim Pois(\lambda)$. Then

$$
\begin{aligned}
& f(\boldsymbol{x}|\lambda) = \prod_{i=1}^n \frac{\lambda^{x_i}}{x_i!} e^{-\lambda} = \\
& \frac{\lambda^{\sum_{i=1}^n x_i}}{\prod_{i=1}^n x_i!} e^{-n\lambda} = \\
& =
\end{aligned}
$$

where

$$
\begin{aligned}
& t = \sum_{i=1}^n x_i \\
& h(\boldsymbol{x}) = \frac{1}{\prod_{i=1}^n x_i!} \\
& g(t|\mu) = \lambda^t e^{-n\lambda}
\end{aligned}
$$

so $\sum x_i$ is a sufficient statistic for λ .

4.1.5.7 Example (4.1.13) Say X belongs to an exponential family, then

$$f(x;\theta) = h(x)\exp\left\{\theta^T T(x) - A(\theta)\right\}$$

and clearly T(x) is a sufficient statistic.

4.1.6 Ancillary Statistics

4.1.6.1 Definition (4.1.14) A statistic S(**X**) whose distribution does not depend on θ is called an **ancillary** statistic.

4.1.6.2 Example (4.1.15) say $X_1, ..., X_n \sim U[\theta, \theta+1]$ and let R be the range of the observations, that is $R = \max\{x_i\} - \min\{x_i\}$. It can be shown that $R \sim Beta(n-1, 2)$ for all θ, and so its distribution is independent of θ.

4.1.6.3 Example (4.1.16) say $X_1, ..., X_n \sim N(\mu, \sigma)$ and let s be the sample standard deviation, then

$$(n-1)s^2/\sigma^2 \sim \chi^2(n-1)$$

and is independent of μ.

The usefulness of an ancillary statistics lies in the fact that we need not worry about the value of the unknown parameter when calculating probabilities.

4.1.7 Consistency

4.1.7.1 Definition (4.1.17) A sequence of estimators $T_n = T_n(X_1, .., X_n)$ is a **consistent** sequence of estimators for θ if $T_n \to \theta$ in probability. That is, for every $\epsilon > 0$ and every θ we have

$$\lim P(|T_n - \theta| > \epsilon) \to 0$$

4.1.7.2 Example (4.1.18) By the WLLN if $\mu = EX$ exists the sample mean is a consistent estimator of μ.

4.1.7.3 Example (4.1.19) say $X_1,..,X_n \sim U[0,\theta]$ and $T_1 = 2\bar{x}$. By the WLLN $\bar{X} \rightarrow E[X] = \theta/2$, so $T_1 = 2\bar{X} \rightarrow \theta$ in probability.

Let $T(\boldsymbol{x}) = \frac{n+1}{n}\max\{x_1,..,x_n\}$. Now $X_i/\theta \sim U[0,1]$, and so $M = \max\{X_1,..,X_n\}/\theta \sim Beta(n,1)$. Now

$$\begin{aligned}
&P(|T-\theta| < \epsilon) = \\
&P(\theta-\epsilon < T < \theta+\epsilon) = \\
&P(\frac{n}{n+1}(1-\epsilon/\theta) < M/\theta < \frac{n}{n+1}(1+\epsilon/\theta)) = \\
&P(\frac{n}{n+1}(1-\epsilon/\theta) < M) = \\
&1 - P(M < \frac{n}{n+1}(1-\epsilon/\theta)) = \\
&1 - \left(\frac{n}{n+1}(1-\epsilon/\theta)\right)^n \rightarrow 1 - 0 = 1
\end{aligned}$$

if $1-\epsilon/\theta < 1$ or $\epsilon < \theta$.

4.1.7.4 Example (4.1.20) say $X_1,...,X_n \sim Geom(p)$ and let $T(\boldsymbol{x}) = 1/\bar{x}$ Note $E[X] = 1/p$, and so

$$\begin{aligned}
&P(|T-p| < \epsilon) = \\
&P(p-\epsilon < 1/\bar{X} < p+\epsilon) = \\
&P(\frac{1}{p+\epsilon} < \bar{X} < \frac{1}{p+\epsilon}) = \\
&P(\frac{1}{p+\epsilon} - \frac{1}{p} < \bar{X} - \frac{1}{p} < \frac{1}{p+\epsilon} - \frac{1}{p}) = \\
&P(-\frac{1}{p(p+\epsilon)}\epsilon < \bar{X} - \frac{1}{p} < \frac{1}{p(p-\epsilon)}\epsilon)
\end{aligned}$$

Note that $p-\epsilon < p+\epsilon$, and so $\frac{1}{p+\epsilon} < \frac{1}{p-\epsilon}$ and $\frac{1}{p(p+\epsilon)} < \frac{1}{p(p-\epsilon)}$, therefore

$$\begin{aligned}
&P(|T-p| < \epsilon) \geq \\
&P(-\frac{1}{p(p+\epsilon)}\epsilon < \bar{X} - \frac{1}{p} < \frac{1}{p(p+\epsilon)}\epsilon) \\
&P(|\bar{X} - \frac{1}{p}| < \frac{1}{p(p+\epsilon)}\epsilon) \rightarrow 1
\end{aligned}$$

by the weak law of large numbers, and so T is a consistent estimator of p.

4.1.7.5 Example (4.1.21) say $X_1,..,X_n$ iid $N(\mu,\sigma)$, σ known, and let $M = \text{median}(\boldsymbol{x})$. Assume wlog that n is odd. Let $\phi(x;\mu)$ and $\Phi(x;\mu)$ be the density and cdf of a normal distribution with mean μ. Then from the theory of order statistics we know that

$$
\begin{aligned}
&f_M(x;\mu) = \\
&\frac{n!}{(\frac{n+1}{2}-1)!(n-\frac{n+1}{2})}!\Phi(x;\mu)^{\frac{n+1}{2}-1}(1-\Phi(x;\mu))^{n-\frac{n+1}{2}}\phi(x;\mu) = \\
&\frac{n!}{(\frac{n-1}{2})!^2}\Phi(x;\mu)^{\frac{n-1}{2}}(1-\Phi(x;\mu))^{\frac{n-1}{2}}\phi(x;\mu)
\end{aligned}
$$

now

$$
\begin{aligned}
&P(|M-\mu|>\epsilon) = \\
&1-P(\mu-\epsilon<M<\mu+\epsilon) = \\
&1-\int_{\mu-\epsilon}^{\mu+\epsilon}\frac{n!}{(\frac{n-1}{2})!^2}\Phi(x;\mu)^{\frac{n-1}{2}}(1-\Phi(x;\mu))^{\frac{n-1}{2}}\phi(x;\mu)dx
\end{aligned}
$$

We use the change of variable $t=\Phi(x;\mu)$, so $t=\phi(x;\mu)dx$. Also let Y be a random variable with distribution Beta($\frac{n+1}{2}$, $\frac{n+1}{2}$), then

$$
\begin{aligned}
&P(|M-\mu|>\epsilon) = \\
&1-\frac{n!}{(\frac{n-1}{2})!^2}\int_{\Phi(\mu-\epsilon;\mu)}^{\Phi(\mu+\epsilon;\mu)}t^{\frac{n-1}{2}}(1-t)^{\frac{n-1}{2}}dt = \\
&1-\frac{n!}{(\frac{n-1}{2})!^2}\frac{(\frac{n-1}{2}!)^2}{n!}\int_{\Phi(\mu-\epsilon;\mu)}^{\Phi(\mu+\epsilon;\mu)}\frac{\Gamma(\frac{n+1}{2}+\frac{n+1}{2})}{\Gamma(\frac{n+1}{2})\Gamma(\frac{n+1}{2})}t^{\frac{n-1}{2}}(1-t)^{\frac{n-1}{2}}dt = \\
&1-P(\Phi(\mu-\epsilon;\mu)<Y<\Phi(\mu+\epsilon;\mu))
\end{aligned}
$$

Note that $\Phi(\mu-\epsilon;x)<\frac{1}{2}$, so

$$
\delta=\frac{1}{2}-\Phi(\mu-\epsilon;x)=\Phi(\mu+\epsilon;x)-\frac{1}{2}>0
$$

also note that
$E[Y]=\frac{\frac{n+1}{2}}{\frac{n+1}{2}+\frac{n+1}{2}}=\frac{1}{2}$ and $var(Y)=\frac{\frac{n+1}{2}\frac{n+1}{2}}{(\frac{n+1}{2}+\frac{n+1}{2})^2(\frac{n+1}{2}+\frac{n+1}{2}+1)}=\frac{1}{4(n+2)}$
and so

$$
\begin{aligned}
&P(|M-\mu|>\epsilon) = \\
&1-P(\frac{1}{2}-\delta<Y<\frac{1}{2}+\delta) = \\
&P(|Y-\frac{1}{2}|>\delta) \le \\
&\frac{Var[Y]}{\delta^2}=\frac{1}{4(n+2)\delta^2}\to 0
\end{aligned}
$$

where the inequality follows from Chebyshev's inequality,and so the median is a consistent estimator of the mean

4.1.8 Efficiency

4.1.8.1 Definition (4.1.22) Say we have a sample $X_1, ..., X_n$ with density $f(x|\theta)$, and we have two unbiased estimators T_1 and T_2 of θ. The efficiency of T_1 relative to T_2 is defined by $\text{eff}(\text{T}_1|\text{T}_2) = \text{var}(\text{T}_1)/\text{Var}(\text{T}_2)$ and we say that T_1 is more efficient than T_2 if $\text{eff}(\text{T}_1|\text{T}_2)<1$.

4.1.8.2 Example (4.1.23) let's look again at the example above: we have $X_1, ..., X_n \sim U[0, \theta]$. We found that $T_1 = 2\bar{x}$ and $T_2 = (n+1)/n \max\{x_1, .., x_n\}$ are unbiased estimators of θ. Now

$$var(T_1) = var(2\bar{X}) = 4var(X_1)/n = 4(\theta^2/12)/n = \theta^2/(3n)$$

and

$$var(T_2) = var(\frac{(n+1)\theta}{n}M/\theta) =$$
$$\left(\frac{(n+1)\theta}{n}\right)^2 \frac{n}{(n+1)^2(n+2)} =$$
$$\frac{\theta^2}{n(n+2)}$$

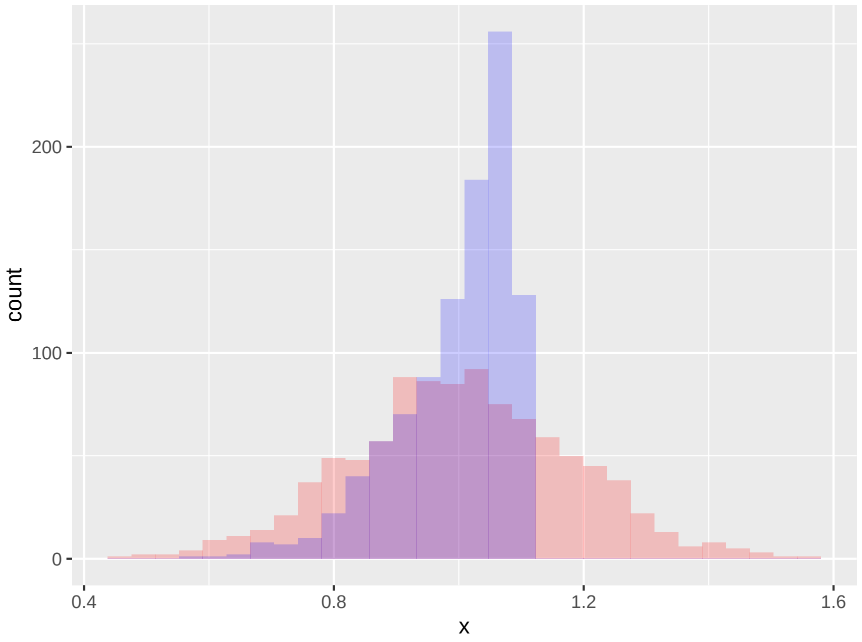
and so

$$eff(T_1|T_2) = var(T_1)/var(T_2) = \frac{\theta^2/(3n)}{\frac{\theta^2}{n(n+2)}} = \frac{n+2}{3} > 1$$

for all $n > 1$. So we find that T_2 is more efficient than T_1 for every value of θ.

Let's do a little simulation:

```r
n <- 10; theta <- 1; B <- 1000
x <-  matrix(runif(n * B, 0, theta), B, n)
T1 <-  2 * apply(x, 1, mean)
T2 <-  (n + 1)/n * apply(x, 1, max)
df <- data.frame(
  x = c(T1, T2),
  y = c(rep(1, 1000), rep(2, 1000)))
ggplot(df, aes(x=x)) +
    geom_histogram(data = subset(df, y == 1),
        fill = "red", alpha = 0.2) +
    geom_histogram(data = subset(df, y == 2),
        fill = "blue", alpha = 0.2)
```



```
print("Means")
```

```
## [1] "Means"
```

```
c(mean(T1), mean(T2))
```

```
## [1] 1.0046276 0.9970209
```

```
print("Relative Efficiency, estimated and true")
```

```
## [1] "Relative Efficiency, estimated and true"
```

```
c(var(T1)/var(T2), (n + 2)/3)
```

```
## [1] 3.64656 4.00000
```

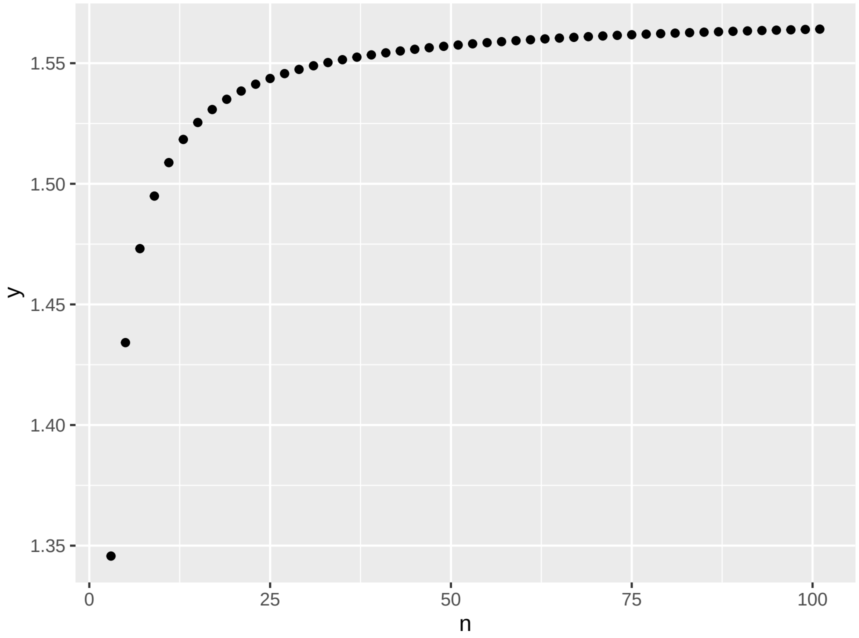
4.1.8.3 Example (4.1.24)

say $X_1,..,X_n \sim N(\mu,1)$, and let $T_1 = \text{median}(\boldsymbol{x})$ and $T_2 = \bar{x}$ Then $var(T_2) = 1/n$. $var(T_1) = E[T_1^2] - (E[T_1])^2$. Now

$$
\begin{aligned}
&f_M(x) = \\
&\frac{n!}{(\frac{n+1}{2}-1)!(n-\frac{n+1}{2})!}\Phi(x)^{\frac{n+1}{2}-1}(1-\Phi(x))^{n-\frac{n+1}{2}}\phi(x) \\
&\frac{n!}{[(n-1)/2]!^2}\left[\Phi(x)(1-\Phi(x))\right]^{(n-1)/2}\phi(x) \\
&E[M^k] = \int_{-\infty}^{\infty} x^k f_M(x)dx
\end{aligned}
$$

and this seems a bit ugly. Let's try and do this numerically:

```
fM <- function(x, n)
    factorial(n)/factorial((n-1)/2)^2*
    (pnorm(x)*(1-pnorm(x)))^((n-1)/2)*dnorm(x)
var.median <- function(n) {
  E1 <- function(x) x*fM(x, n)
  E2 <- function(x) x^2*fM(x, n)
  integrate(E2, -3, 3)$value -
    integrate(E1, -3, 3)$value^2
}
n <-  2*1:50+1
y <-  0*n
for (i in 1:50) y[i] = var.median(n[i])
df <- data.frame(n=n, y=y/(1/n))
ggplot(data=df, aes(n, y)) +
  geom_point()
```



```
round(y[50]/(1/101), 3)
```

```
## [1] 1.564
```

It seems that for large n the relative efficiency is about 1.6.

Generally it is quite possible that one estimator is more efficient than another only for a subset of the parameter space, and the other one is more efficient on the rest. Also, one estimator might be more efficient if n is small but things are reversed if n is large.

An interesting question is whether in a given problem there is an estimator that is more efficient than any other (unbiased) estimator. At least in the sense of minimum variance, such an estimator would be optimal. In order to answer this question we need the following:

4.1.8.4 Definition (4.1.25) Let X be a random variable with density $f(x;\theta)$. Then the *Fisher Information* of θ is defined by

$$I(\theta) = -E\left[\frac{d^2 \log f(X;\theta)}{d\theta^2}\right]$$

4.1.8.5 Theorem (4.1.26) Under suitable conditions

$$I(\theta) = E\left[\left(\frac{d \log f(X;\theta)}{d\theta}\right)^2\right]$$

proof

We will do the proof for a continuous rv. Also we will assume that any interchange of integral and derivative is ok

$$\begin{aligned}
&\frac{\partial^2}{\partial\theta^2}\log f(x;\theta) = \\
&\frac{\partial}{\partial\theta}\frac{\frac{\partial}{\partial\theta}f(x;\theta)}{f(x;\theta)} = \\
&\frac{(\frac{\partial^2}{\partial\theta^2}f(x;\theta))f(x;\theta) - (\frac{\partial}{\partial\theta}f(x;\theta))^2}{f^2(x;\theta)} = \\
&\frac{(\frac{\partial^2}{\partial\theta^2}f(x;\theta))}{f(x;\theta)} - \frac{(\frac{\partial}{\partial\theta}f(x;\theta))^2}{f^2(x;\theta)} = \\
&\frac{(\frac{\partial^2}{\partial\theta^2}f(x;\theta))}{f(x;\theta)} - (\frac{\partial}{\partial\theta}\log f(x;\theta))^2
\end{aligned}$$

also

$$
\begin{aligned}
&E\left[\frac{(\frac{\partial^2}{\partial\theta^2}f(X;\theta))}{f(X;\theta)}\right] = \\
&\int \frac{(\frac{\partial^2}{\partial\theta^2}f(x;\theta))}{f(x;\theta)} f(x;\theta)dx = \\
&\int \frac{\partial^2}{\partial\theta^2} f(x;\theta)dx = \\
&\frac{\partial^2}{\partial\theta^2}\int f(x;\theta)dx = \\
&\frac{\partial^2}{\partial\theta^2} 1 = 0
\end{aligned}
$$

4.1.8.6 Theorem (4.1.27) Rao-Cramer

Let $X_1, ..., X_n$ be a sample from density $f(x|\theta)$, and let T be any estimator satisfying

1. $\frac{d}{d\theta}E[T(X)] = \int \frac{d}{d\theta}T(x)f(x;\theta)dx$
2. $var(T(X)) < \infty$

then

$$
var(T(X)) \geq \frac{(\frac{d}{d\theta}E[T(X)])^2}{nI(\theta)}
$$

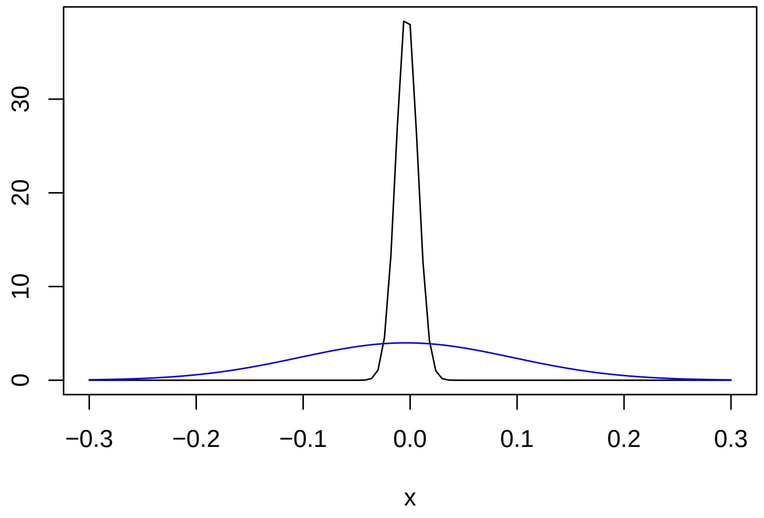
The right hand side of this inequality is called the **Rao-Cramer Lower Bound**.

Note that if T is an unbiased estimator of θ, we have $E[T(\boldsymbol{X})] = \theta$ and the numerator is just 1.

Note that the Fisher Information is calculated for a single random variable. The sample size comes in by multiplying with n.

Does it make sense that the second derivative of the log-likelihood should come into play? Consider the following log-likelihood curves:

```r
xbar <- mean(rnorm(100))
ybar <- mean(rnorm(100, 0, 0.1))
curve(dnorm(ybar, x, 0.1/10), -0.3, 0.3, ylab="")
curve(dnorm(xbar, x, 1/10), -0.3, 0.3,
      add=TRUE, col="blue")
```



so the more peaked the log-likelihood curve is, the smaller the variance of the estimator. But the peakness of a curve is found via the second derivative!

4.1.8.7 Example (4.1.28) Let $X_1, ..., X_n \sim N(\mu, \sigma)$ and consider estimating the standard deviation σ, where μ is unknown. The normal density satisfies the conditions of the theorem, and in (3.2.7) we found

$$I(\sigma) = -E\left[\frac{d^2 \log f(x|\theta)}{d\sigma^2}\right] = 1/\sigma^2$$

and so any unbiased estimator T of σ must satisfy

$$var(T) \geq \frac{1}{nI(\sigma)} = \sigma^2/n$$

4.1.8.8 Example (4.1.29) Let $X_1, ..., X_n \sim Pois(\lambda)$. Now

$$\begin{aligned}
&\log f(x|\lambda) = \log\left(\frac{\lambda^x}{x!}e^{-\lambda}\right) = x\log\lambda - \log x! - \lambda\\
&\frac{d\log(x|\lambda)}{d\lambda} = \frac{x}{\lambda} - 1\\
&\frac{d^2\log(x|\lambda)}{d\lambda^2} = -\frac{x}{\lambda^2}\\
&I(\lambda) = -E\left[\frac{d^2\log(X|\lambda)}{d\lambda^2}\right] =\\
&-E\left[-\frac{X}{\lambda^2}\right] = \frac{E[X]}{\lambda^2} = \frac{1}{\lambda}
\end{aligned}$$

and so for any unbiased estimator T we have $var(T) \ge \lambda/n$.

Note that

$$var(\bar{X}) = var(X_1)/n = \lambda/n$$

and so the sample mean is a *minimum variance unbiased estimator (UMVU)* for λ.

4.1.8.9 Example (4.1.30) Again let's look at the example of $U[0,\theta]$ above. There we have $f(x|\theta) = 1/\theta, 0 < x < \theta$. So

$$E\left[\left(\frac{\partial\log f(X|\theta)}{\partial\theta}\right)^2\right] = E\left[\left(\frac{\partial\log 1/\theta}{\partial\theta}\right)^2\right] = (-1/\theta^2) = 1/\theta^2$$

so it appears that the Rao-Cramer theorem says that for any unbiased estimator T we have $var(T) \ge \theta^2/n$, but we have already seen that $var(T_2) = \theta^2/(n(n+2)) < \theta^2/n$.

So, what goes wrong? Let's check the condition of the theorem for n=1. Then $T(X) = 2X$, $\frac{d}{d\theta}E[T(X)] = \frac{d}{d\theta}\theta = 1$ but

$$\begin{aligned}
&\int_{-\infty}^{\infty}\frac{d}{d\theta}T(x)f(x|\theta)dx =\\
&\int_{-\infty}^{\infty}\frac{d}{d\theta}(2xI_{[0,\theta]}(x)\frac{1}{\theta})dx =\\
&\int_0^{\theta}\frac{d}{d\theta}(2x\frac{1}{\theta})dx =\\
&-\frac{1}{\theta^2}\int_0^{\theta}2xdx =\\
&-\frac{1}{\theta^2}\theta^2 = -1
\end{aligned}$$

So here the first assumption of the theorem is not satisfied, something that happens quite often, especially if the parameter is part of the boundary condition, such as $0 < x < \theta$

How about checking the theorem in a case where it does work?

Say $X_1, .., X_n \sim N(\mu, \sigma)$, with σ known, and $T(\boldsymbol{x}) = \bar{\boldsymbol{x}}$. Now

$$
\begin{aligned}
&\frac{d}{d\mu} \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^n (x_i-\mu)^2\right\} = \\
&\exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^n (x_i-\mu)^2\right\}\left(\frac{1}{\sigma^2}\sum_{i=1}^n (x_i-\mu)\right) \\
&\frac{d}{d\mu} T(\boldsymbol{x}) f(\boldsymbol{x}|\mu) = \\
&\left(\frac{1}{n}\sum_{i=1}^n x_i\right)(2\pi\sigma^2)^{-n/2}\exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^n (x_i-\mu)^2\right\}\left(\frac{1}{\sigma^2}\sum_{i=1}^n (x_i-\mu)\right) = \\
&\frac{1}{n\sigma^2}\left(\sum_{i=1}^n x_i\right)\left(\sum_{i=1}^n (x_i-\mu)\right) f(\boldsymbol{x}|\mu) = \\
&\frac{1}{n\sigma^2}\left(\sum_{i,j=1}^n x_i(x_j-\mu)\right) f(\boldsymbol{x}|\mu) = \\
&\frac{1}{n\sigma^2}\left(\sum_{i=1}^n x_i(x_i-\mu) f(\boldsymbol{x}|\mu) + \sum_{i\neq j}^n x_i(x_j-\mu) f(\boldsymbol{x}|\mu)\right)
\end{aligned}
$$

Now

$$
\begin{aligned}
&\int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} \sum_{i=1}^n x_i(x_i-\mu) f(\boldsymbol{x}|\mu) d\boldsymbol{x} = \\
&\sum_{i=1}^n \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} x_i(x_i-\mu) f(\boldsymbol{x}|\mu) d\boldsymbol{x} = \\
&\sum_{i=1}^n \int_{-\infty}^{\infty} x_i(x_i-\mu)\phi(x_i|\mu,\sigma) dx_i = \\
&\sum_{i=1}^n \left(E[X_i^2] - \mu E[X_i]\right) = \\
&\sum_{i=1}^n \left(var(X_i) + E[X_i]^2 - E[X_i]^2\right) = n\sigma^2
\end{aligned}
$$

where $\phi(x_i|\mu, \sigma)$ is the density of a $N(\mu, \sigma)$.

Also

$$
\begin{aligned}
&\int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} \sum_{i \neq j}^{n} x_i(x_j - \mu) f(\boldsymbol{x}|\mu) d\boldsymbol{x} = \\
&\sum_{i \neq j}^{n} \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} x_i(x_j - \mu) f(\boldsymbol{x}|\mu) d\boldsymbol{x} = \\
&\sum_{i \neq j}^{n} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x_i(x_j - \mu)\phi(x_i|\mu,\sigma)\phi(x_j|\mu,\sigma) dx_i dx_j = \\
&\sum_{i \neq j}^{n} \int_{-\infty}^{\infty} \left\{ \int_{-\infty}^{\infty} (x_j - \mu)\phi(x_i|\mu,\sigma) dx_i \right\} x_i \phi(x_j|\mu,\sigma) dx_j \\
&\sum_{i \neq j}^{n} \int_{-\infty}^{\infty} \left\{ E[X_i] - \mu \right\} x_i \phi(x_j|\mu,\sigma) dx_j = 0
\end{aligned}
$$

and so

$$
\int \frac{d}{d\mu} T(\boldsymbol{x}) f(\boldsymbol{x}|\mu) d\boldsymbol{x} = \frac{1}{n\sigma^2} n\sigma^2 = 1
$$

4.1.9 Robustness

In point estimation we first start by assuming a parametric model for the data, such as $X_1, ..., X_n \sim N(\mu, \sigma)$, and then try to estimate the parameters of the model. But what if our model is wrong, for example if the true model is a t distribution instead of the Normal? A **robust** estimator is one that does not depend to strongly on the assumed model.

4.1.9.1 Example (4.1.31) Let $X_1, ..., X_n \sim N(\mu, \sigma)$. It is known that the sample mean is the best estimator of μ in the sense that it has the smallest variance of all unbiased estimators. But what happens if our assumption of the normal distribution is wrong?

Let's consider instead a model called the δ-contamination model:

$$
X_i \sim \begin{cases} N(\mu, \sigma) & \text{with probability } \delta \\ f(x) & \text{with probability } 1 - \delta \end{cases}
$$

for some other density f. Suppose first we let f be any density with mean θ and variance τ^2. Let $Z_i \sim Ber(\delta)$, then

$$
\begin{aligned}
&E\{X_i|Z_i=0\}=\theta; E\{X_i|Z_i=1\}=\mu\\
&E[X_i]=E\left[E\{X_i|Z_i\}\right]=\\
&E\{X_i|Z_i=0\}P(Z_i=0)+E\{X_i|Z_i=1\}P(Z_i=1)=\\
&\theta\delta+\mu(1-\delta)\\
&E[X_i^2]=E\left[E\{X_i^2|Z_i\}\right]=\\
&E\{X_i^2|Z_i=0\}P(Z_i=0)+E\{X_i^2|Z_i=1\}P(Z_i=1)=\\
&(\tau^2+\theta^2)\delta+(\sigma^2+\mu^2)(1-\delta)\\
&var(X_i)=\\
&(\tau^2+\theta^2)\delta+(\sigma^2+\mu^2)(1-\delta)-(\theta\delta+\mu(1-\delta))^2=\\
&(1-\delta)\sigma^2+\delta\tau^2+(1-\delta)\delta(\mu-\theta)^2
\end{aligned}
$$

and so

$$var(\bar{X})=\left[(1-\delta)\sigma^2+\delta\tau^2+(1-\delta)\delta(\mu-\theta)^2\right]/n$$

Now if f is a Cauchy density we have $\tau=\infty$ and so the variance of the sample mean is infinite as well!

One way to measure the robustness of an estimator is as follows:

4.1.9.2 Definition (4.1.32) Let T_n be a statistic. T_n has a **breakdown value** b if at most b% of the values in the sample can be moved to infinity without T_n becoming infinite.

4.1.9.3 Example (4.1.33) Say T_n is the sample mean. Now

$$T_n=\frac{1}{n}\sum x_i=\frac{1}{n}\sum x_{[i]}\to\infty$$

if $x_{[n]}=x_{[(1-\epsilon)n]}\to\infty$, and so the sample mean has a breakdown value of 0.

4.1.9.4 Example (4.1.34) Say T_n is the sample median. Now T_n has a breakdown value of 1/2.

4.2 Methods for Finding Estimators

4.2.1 Method of Moments

Let $\boldsymbol{x}=(x_1,...,x_n)$ be a sample from a distribution with density $f(x|\theta_1,...,\theta_k)$. Define the i$^{\text{th}}$ sample moment by

$$m_i=(x_1^i+..+x_n^i)/n$$

Analogously define the i$^{\text{th}}$ population moment by

$$\mu_i = E[X^i]$$

Of course μ_i is a function of the $\theta_1, ..., \theta_k$. So we can find estimators of $\theta_1, ..., \theta_k$ by solving the system of k equations in k unknowns

$$m_i = \mu_i, i = 1, .., k$$

4.2.1.1 Example (4.2.1)

say $X_1, ..., X_n \sim N(\mu, \sigma)$. Here $\theta_1 = \mu$ and $\theta_2 = \sigma^2$. Then

$$\begin{aligned}
&\mu_1 = E[X] = \mu = m_1 = \bar{x} \\
&\mu_2 = E[X^2] = \sigma^2 + \mu^2 = m_2 = \overline{x^2} \quad \hat{\mu} = \bar{x} \\
&\widehat{\sigma^2} = \overline{x^2} - \bar{x}^2
\end{aligned}$$

4.2.1.2 Example (4.2.2)

say $X_1, ..., X_n \sim Gamma(\alpha, \beta)$. Then

$$\begin{aligned}
&E[X^k] = \beta^k \prod_{i=1}^{k} (\alpha + i - 1) \\
&E[X] = \alpha\beta = \bar{x} \\
&E[X^2] = (\alpha + 1)\alpha\beta^2 = \overline{x^2} \\
&(\alpha\beta)^2 + (\alpha\beta)\beta = \overline{x^2} \\
&\bar{x}^2 + \bar{x}\beta = \overline{x^2} \\
&\hat{\beta} = \left(\overline{x^2} - \bar{x}^2\right) / \bar{x} \\
&\hat{\alpha} = \bar{x}^2 / \left(\overline{x^2} - \bar{x}^2\right)
\end{aligned}$$

Here is an R calculation:

```r
x <- rgamma(10000, 2.3, 1/5.6)
xbar <-  mean(x)
x2bar <-  mean(x^2)
round(c(xbar^2/(x2bar - xbar^2), x2bar/xbar - xbar), 2)
```

```
## [1] 2.29 5.65
```

4.2.2 Method of Least Squares

4.2.2.1 Example (4.2.3)

say $X_1, .., X_n \sim N(\mu, \sigma)$, σ assumed known. Now μ is the mean of the normal distribution, so any observations should be scattered around μ. If we estimate

μ by say a, then $\epsilon_i = X_i - a$ is called the i$^{\text{th}}$ **residual** or error. Now a measure of the overall error is

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

and an estimator of the parameter can be found as the value $\hat{\mu}$ that minimizes G:

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

Obviously we did not need to use $G(a) = \sum(x_i - a)^2$, other possible choices are

- $G(a) = \sum |x_i - a|$, which leads to a=median($\mathbf{X}$).
- $G(a) = \max\{|x_i - a|\}$, which leads to the mode.

4.2.3 Maximum Likelihood

The idea here is this: the likelihood function gives the likelihood (not the probability!) of a value of the parameter given the observed data, so why not choose the value that "matches" (gives the greatest likelihood) to the observed data.

4.2.3.1 Example (4.2.4) say $X_1, ..., X_n \sim Ber(p)$. First notice that a function f has an extremal point at x iff log(f) does as well because d/dx{log(f(x))}=f'(x)/f(x)=0 iff f'(x)=0.

Let $y = sumx_i$, then

$$L(p|\boldsymbol{x}) = f(\boldsymbol{x}|p) = p^y(1-p)^{n-y}$$
$$l(p) = \log L(p|\boldsymbol{x}) = \log\left(p^y(1-p)^{n-y}\right) = y\log p + (n-y)\log(1-p)$$
$$\frac{dl}{dp} = \frac{y}{p} - \frac{n-y}{1-p} = 0$$
$$\hat{p} = y/n = \bar{x}$$

the second derivative shows that this is indeed a maximum.

4.2.3.2 Example (4.2.5) say $X_1, n.., X_n \sim U[0, \theta]$, $\theta > 0$. Then

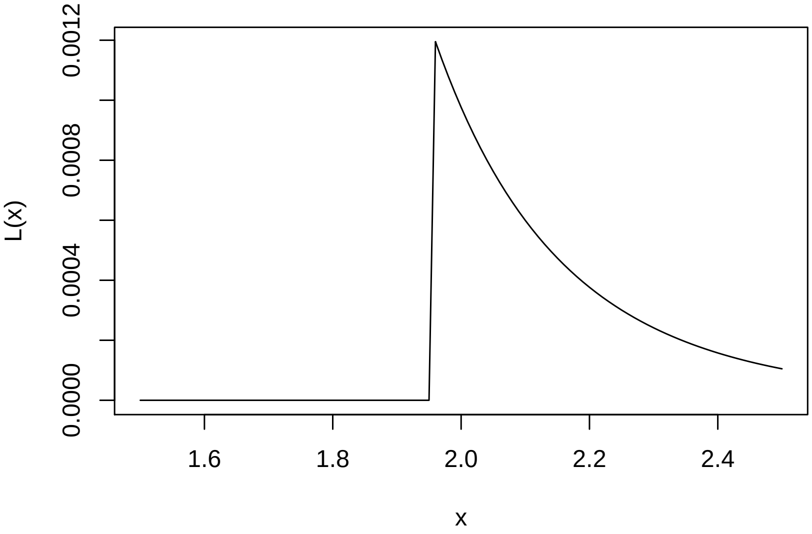
$$L(\theta|\boldsymbol{x}) = f(\boldsymbol{x}|\theta) = \prod_{i=1}^{n} \frac{1}{\theta} I_{[0,\theta]}(x_i) =$$
$$\theta^{-n} I_{[0,\theta]}(\max\{x_i\}) =$$
$$\theta^{-n} I_{[\max\{x_i\},\infty]}(\theta)$$

Here is a graph of this function:

```
n=10
theta0=2
xmax=max(runif(n, 0, theta0))
round(c(xmax, 1/xmax), 3)
```

```
## [1] 1.96 0.51
```

```
L=function(t) ifelse(t>=xmax, 1, 0)/t^n
curve(L, 1.5, 2.5)
```



So $L(\theta|\boldsymbol{x})$ is 0 on $(0, \max(x_i))$, at $\max(x_i)$ it jumps to $1/(\max(x_i))^n$ and then monotonically decreases as θ gets bigger, so the maximum is obtained at $\theta = \max\{x_i\}$, therefore the mle is $\max\{x_i\}$.

Notice that here log f is of no use because f(x)=0 for values of x close to the point were the maximum is obtained.

4.2.3.3 Example (4.2.6) say $X_1, .., X_n \sim Bin(n, p)$, both p and n unknown. We want to find the mle's of p and n. We have

$$
\begin{aligned}
&L(p|\boldsymbol{x}) = f(\boldsymbol{x}|p) = \prod_{i=1}^{n} \binom{n}{x_i} p^{x_i}(1-p)^{n-x_i} \\
&l(p) = \log L(p|\boldsymbol{x}) = \log\left(\prod_{i=1}^{n} \binom{n}{x_i} p^{x_i}(1-p)^{n-x_i}\right) = \\
&\sum_{i=1}^{k} \log\binom{n}{x_i} + \left(\sum_{i=1}^{k} x_i\right)\log p + \left(nk - \sum_{i=1}^{k} x_i\right)\log(1-p)
\end{aligned}
$$

now

$$
\frac{dl}{dp} = \left(\sum_{i=1}^{k} x_i\right)\frac{1}{p} - \left(nk - \sum_{i=1}^{k} x_i\right)\frac{1}{1-p} = 0
$$

and so $\hat{p} = (\sum_{i=1}^{k} x_i)/(nk)$, and so for any fixed value of n we have the mle for p. n has to be an integer, and so we only need to search through the values of n for the overall mle. This is done in *mle.bin.n*. We also need a routine that calculates log(n!) for any n. *logfac* does this using Sterling's formula.

```r
logfac <- function(n)
  ifelse(n<20, log(factorial(n)),
    0.918938533205+(n+0.5)*log(n)-n+(1/12-1/(360*n^2))/n)
```

```r
mle.bin.n <- function (x, Show = F)
{
    k <-  length(x)
    xbar <-  mean(x)
    f <-  function(n) {
        phat <-  xbar/n
        f <-  k*logfac(n) + n*k*(phat* log(phat) +
              (1-phat)*log(1-phat))
        for (i in 1:k) f <-  f - logfac(n-x[i])
        f
    }
    n <-  max(max(x) + 2, floor(xbar + 2.5 * var(x)))
    l <-  c(0, 0)
    l[1] <-  f(n)
    l[2] <-  f(n + 1)
    if (Show)
        print(c(n, l[1], n+1, l[2]))
    if (l[1] > l[2]) {
        repeat {
            n <-  n-1
            if (n == max(x))
                return(c(n, mean(x)/n))
            l[2] <-  l[1]
            l[1] <-  f(n)
            if (Show)
                print(c(n, l[1], n+1, l[2]))
            if (l[1] < l[2])
                return(c(n + 1, xbar/(n + 1)))
        }
    }
    repeat {
        n <-  n+1
        l[2] <- f(n + 1)
        if (Show)
            print(c(n, l[1], n+1, l[2]))
        if (l[1] > l[2])
            return(c(n, xbar/n))
        l[1] <-  l[2]
    }

}
```

```
x <- rbinom(1000, 67, 0.2)
mle.bin.n(x, Show = TRUE)
```

```
## [1]    39.00 21801.99    40.00 21803.02
## [1]    40.00 21801.99    41.00 21803.89
## [1]    41.00 21803.89    42.00 21804.63
## [1]    42.00 21804.63    43.00 21805.25
## [1]    43.00 21805.25    44.00 21805.77
## [1]    44.00 21805.77    45.00 21806.21
## [1]    45.00 21806.21    46.00 21806.57
## [1]    46.00 21806.57    47.00 21806.87
## [1]    47.00 21806.87    48.00 21807.12
## [1]    48.00 21807.12    49.00 21807.31
## [1]    49.00 21807.31    50.00 21807.47
## [1]    50.00 21807.47    51.00 21807.59
## [1]    51.00 21807.59    52.00 21807.68
## [1]    52.00 21807.68    53.00 21807.74
## [1]    53.00 21807.74    54.00 21807.78
## [1]    54.00 21807.78    55.00 21807.80
## [1]    55.0 21807.8    56.0 21807.8
```

```
## [1] 55.0000000  0.2465273
```

4.2.3.4 Example (4.2.7) $X_1, .., X_n \sim N(\mu, \sigma)$:

$$
\begin{aligned}
&l(\mu, \sigma) = -\frac{n}{2}\log(2\pi) - n\log\sigma - \frac{1}{2\sigma^2}\sum(x_i-\mu)^2\\
&\frac{dl(\mu,\sigma)}{d\mu} = \frac{1}{\sigma^2}\sum(x_i-\mu) = (\sum x_i - n\mu)/\sigma^2 = 0\\
&\hat{\mu} = \bar{x}\\
&\frac{dl(\mu,\sigma)}{d\sigma} = -\frac{n}{\sigma} + \frac{1}{\sigma^3}\sum(x_i-\mu)^2 = 0\\
&\hat{\sigma} = \sqrt{\frac{1}{n}\sum(x_i-\bar{x})^2}
\end{aligned}
$$

4.2.3.5 Example (4.2.8) We have observations $X_1, ..., X_n$ which are independent. We know that our population is made up of two groups (Men - Women, say) and each observation comes from one or the other group but we don't know which. Observations from group i have a $N(\mu_i, \sigma_i)$, i=1,2, distribution. We want to estimate the parameters.

What we have here is called a mixture distribution. Say that proportion of members of group 1 in the population is α. Let's introduce a latent (unobservable) r.v. Z_i, which is 1 if observation X_i comes from group 1, and 2 if it comes from 2. Then

$$
\begin{aligned}
&F(x) = P(X_i \le x) =\\
&P\left(X_i \le x|Z_i = 1\right)P(Z_i = 1) + P\left(X_i \le x|Z_i = 2\right)P(Z_i = 2) =\\
&\Phi(x|\mu_1,\sigma_1)\alpha + \Phi(x|\mu_2,\sigma_2)(1-\alpha)\\
&f(x) = \alpha\phi(x|\mu_1,\sigma_1) + (1-\alpha)\phi(x|\mu_2,\sigma_2)\\
&L(\alpha,\mu_1,\sigma_1,\mu_2,\sigma_2|\boldsymbol{x}) = \prod_{i=1}^{n}\left[\alpha\phi(x|\mu_1,\sigma_1) + (1-\alpha)\phi(x|\mu_2,\sigma_2)\right]\\
&l(\alpha,\mu_1,\sigma_1,\mu_2,\sigma_2|\boldsymbol{x}) = \sum_{i=1}^{n}\log\left[\alpha\phi(x|\mu_1,\sigma_1) + (1-\alpha)\phi(x|\mu_2,\sigma_2)\right]
\end{aligned}
$$

where we use the notation $\Phi(x|\mu,\sigma)$ for the cdf of a $N(\mu,\sigma)$ r.v and $\phi(x|\mu,\sigma)$ for its density.

Unfortunately this expression does not simplify! Also, it is a function in 5 dimensions, so just looking at it with a graph is difficult.

To start let's keep it simple and assume we know μ_1, σ_1, μ_2 and σ_2 and we want to estimate α. Then

$$
\frac{dl}{d\alpha} = \sum_{i=1}^{n}\frac{\phi(x|\mu_1,\sigma_1) - \phi(x|\mu_2,\sigma_2)}{\alpha\phi(x|\mu_1,\sigma_1) + (1-\alpha)\phi(x|\mu_2,\sigma_2)} = 0
$$

This is a non-linear equation, which can not be solved explicitly, so we will have to do it numerically. A standard method in numerical analysis for solving equations of the form h(x)=0 is *Newton's method*:

pick a starting point x_1, find

$$x_{n+1} = x_n - \frac{h(x_n)}{h'(x_n)}$$

If the starting point is close enough to a solution of the equation, the sequence will converge to it.

Let's implement Newton's method for this problem. To simplify the notation set $\phi_i = \phi(x_i, \mu_2, \sigma_2)$ and $\psi_i = \phi(x_i, \mu_2, \sigma_2) - \phi(x_i, \mu_1, \sigma_1)$, then

$$h(a) = \sum_{i=1}^{n} \frac{\psi_i}{\alpha\psi + \phi_i}$$

and

$$h'(a) = -\sum_{i=1}^{n} \frac{\psi_i^2}{(\alpha\psi + \phi_i)^2}$$

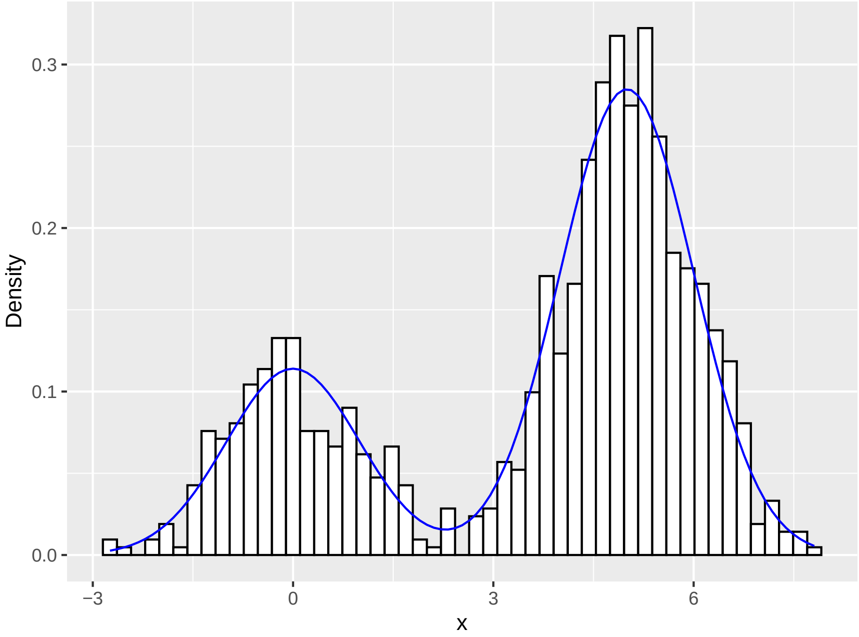
Let's implement this:

```r
mixmle1 <- function(x, mu, sigma) {
    phi <- dnorm(x, mu[2], sigma[2])
    psi <- dnorm(x, mu[1], sigma[1]) - phi
    anew <- 0.5
    repeat {
        aold <- anew
        h <- sum(psi/(aold*psi+phi))
        hprime <- -sum(psi^2/(aold*psi+phi)^2)
        anew <- aold - h/hprime
        if (abs(anew - aold) < 10^-5)
            break
    }
    anew
}
```

```r
n <- 1000; alpha <- 0.3
mu <- c(0, 5); sigma <- c(1, 1)
z <- sample(c(1, 2), size=n, replace=TRUE,
            prob=c(alpha, 1-alpha))
x <- c(rnorm(n, mu[1], sigma[1])[z == 1],
       rnorm(n, mu[2], sigma[2])[z == 2])
alphahat <- mixmle1(x, mu, sigma)
alphahat
```

```
## [1] 0.2857118
```

```
f <- function(x)
  alphahat*dnorm(x, mu[1], sigma[1]) +
  (1-alphahat)*dnorm(x, mu[2], sigma[2])
df <- data.frame(x=x)
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") +
  stat_function(fun = f, colour = "blue")
```



Next let's consider the case where we know α, σ_1 and σ_2 and want to estimate μ_1 and μ_2. For this we need a multivariate extension of Newton's method. Say $h(\boldsymbol{x})$ is a real-valued function in $\mathbb{R}^n$, and we wish to find a maximum (or more generally an extremal point) of h. Let Δ h be the gradient of h, that is

$$\Delta h_i(x) = \frac{\partial h(x)}{\partial x_i}$$

and let H be the Hessian matrix defined by

$$H_{ij}(x) = \frac{\partial^2 h(x)}{\partial x_i \partial x_j}$$

then

$$x_{n+1} = x_n - H^{-1}(x_n)\Delta h(x_n)$$

Here this means:

$$\begin{aligned}
&\frac{d\phi}{d\mu} = \frac{d}{d\mu}\left[\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}\right] = \\
&\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}\frac{x-\mu}{\sigma^2} = \frac{x-\mu}{\sigma^2}\phi(x) \\
&\frac{d^2\phi}{d\mu^2} = \frac{-1}{\sigma^2}\phi(x) + \left(\frac{x-\mu}{\sigma^2}\right)^2\phi(x) = \\
&\left(\frac{(x-\mu)^2-\sigma^2}{\sigma^4}\right)\phi(x)
\end{aligned}$$

Again we use some short-cut notation: set $\phi_i^k = \phi(x_i, \mu_k, \sigma_k)$ and $\psi_i = \phi(x_i, \mu_1, \sigma_1) - \phi(x_i, \mu_2, \sigma_2)$, then

$$\begin{aligned}
h_1(\mu_1, \mu_2) &= \frac{\alpha}{\sigma_1^2}\sum_{i=1}^n \frac{(x_i-\mu_1)\phi_i^1}{\psi_i} \\
h_2(\mu_1, \mu_2) &= \frac{(1-\alpha)}{\sigma_2^2}\sum_{i=1}^n \frac{(x_i-\mu_2)\phi_i^2}{\psi_i} \\
H[1,1] &= \frac{\alpha}{\sigma_1^4}\sum_{i=1}^n \frac{[(x_i-\mu_1)-\sigma_1^2]\phi_i^1\psi_i - \alpha(x_i-\mu_1)^2(\phi_i^1)^2}{\psi_i^2} \\
H[1,2] = H[2,1] &= \frac{\alpha(1-\alpha)}{\sigma_1^2\sigma_2^2}\sum_{i=1}^n \frac{(x_i-\mu_1)(x_i-\mu_2\phi_i^1\phi_i^2}{\psi_i^2} \\
H[2,2] &= \frac{1-\alpha}{\sigma_2^4}\sum_{i=1}^n \frac{[(x_i-\mu_2)-\sigma_2^2]\phi_i^2\psi_i - (1-\alpha)(x_i-\mu_2)^2(\phi_i^2)^2}{\psi_i^2}
\end{aligned}$$

Also note that if

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

then

$$\boldsymbol{A}^{-1} = \frac{1}{ac-b^2}\begin{pmatrix} c & -b \\ -b & a \end{pmatrix}$$

and so we have

```
mixmle2 <- function (x, alpha, sigma) {
  h <- c(0, 0)
  H <- matrix(0, 2, 2)
  munew <- c(1, 4)
  repeat {
    muold <- munew
    phi1 <- dnorm(x, muold[1], sigma[1])
    phi2 <- dnorm(x, muold[2], sigma[2])
    psi <- alpha*phi1 + (1-alpha)*phi2
    print(round(c(munew, sum(log(psi))), 3))
    h[1] <- alpha/sigma[1]^2*sum((x-muold[1])*phi1/psi)
    h[2] <- (1-alpha)/sigma[2]^2*sum((x-muold[2])*phi2/psi)
    H[1, 1] <- alpha/sigma[1]^4*sum(((x-muold[1]-sigma[1]^2) *
                      phi1 * psi - alpha * (x - muold[1])^2 * phi1^2)/psi^2)
    H[1, 2] <- alpha*(1-alpha)/sigma[1]^2 * sigma[2]^2 *
            sum((x - muold[1]) * (x - muold[2]) * phi1 *  phi2/psi^2)
    H[2, 1] <- H[1, 2]
    H[2, 2] <- (1-alpha)/sigma[2]^4 * sum(((x-muold[2] -
            sigma[2]^2)*phi2*psi-(1-alpha) * (x - muold[2])^2 * phi2^2)/psi^2)
    Hinf <- cbind(c(H[2, 2], -H[1, 2]), c(-H[1, 2],
                    H[1, 1]))/(H[1, 1] * H[2, 2] - H[1, 2]^2)
    munew <- muold - Hinf %*% h
    if (sum(abs(munew - muold)) < 10^-5)
        break
  }
  round(c(munew), 3)
}
```

```
mixmle2(x, alpha, sigma)
```

```
## [1]     1.000     4.000 -2522.596
## [1]     0.742     4.518 -2166.417
## [1]     0.510     4.846 -2037.091
## [1]     0.324     4.975 -2001.391
## [1]     0.190     5.026 -1989.504
## [1]     0.102     5.046 -1985.481
## [1]     0.050     5.053 -1984.220
## [1]     0.021     5.056 -1983.852
## [1]     0.006     5.056 -1983.749
## [1]    -0.003     5.056 -1983.721
## [1]    -0.007     5.056 -1983.713
## [1]    -0.009     5.055 -1983.711
## [1]    -0.010     5.055 -1983.710
## [1]    -0.011     5.055 -1983.710
## [1]    -0.011     5.055 -1983.710
## [1]    -0.011     5.055 -1983.710
## [1]    -0.012     5.055 -1983.710
## [1]    -0.012     5.055 -1983.710
## [1]    -0.012     5.055 -1983.710
## [1]    -0.012     5.055 -1983.710
## [1]    -0.012     5.055 -1983.710
```

```
## [1] -0.012  5.055
```

How about the complete problem with 5 parameters? This can be done but is quite an effort.

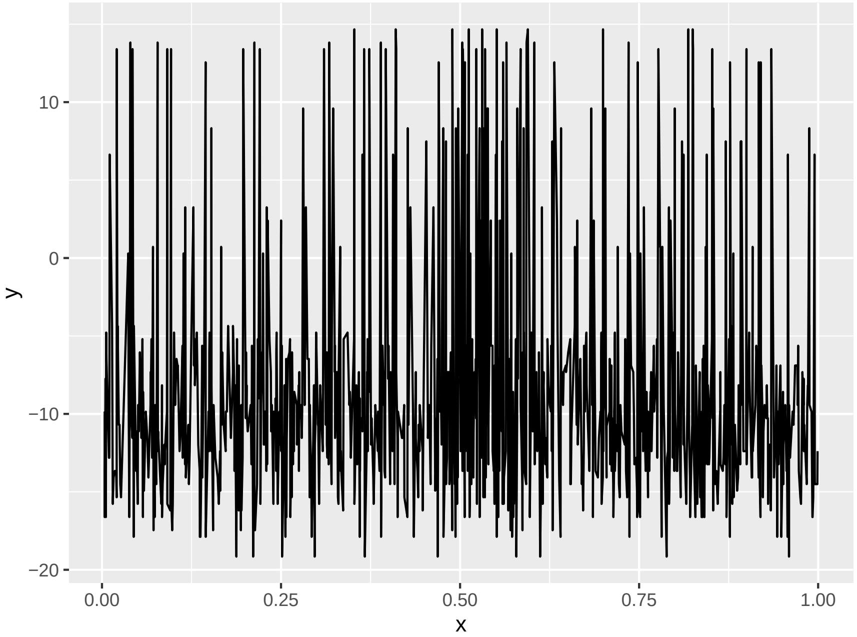
4.2.3.6 Example (4.2.9)

Say $X_1, .., X_n$ are iid with density $h(x|\beta) = 0.95 I_{[0,1]}(x) + 0.5 I_{[\beta,\beta+0.1]}(x)$, so $X \sim U[0,1]$ with probability 0.95 and $X \sim U[\beta, \beta+0.1]$ with probability 0.05. Now

$$l(\beta|\boldsymbol{x}) = \sum \log \left[0.95 I_{[0,1]}(x) + 0.5 I_{[\beta,\beta+0.1]}(x)\right]$$

and here is what this looks like:

```
a <- 0.95; b <- 0.5
x <-  c(runif(1000*a), runif(1000*(1-a), b, b+0.1))
z <-  seq(0, 0.9, length = 100)
loglike <-  function(b)
    sum(log(a+(1-a)*10*ifelse(x>b & x<b+0.1, 1, 0)))
y <-  z
for (i in 1:100) y[i] <-  loglike(z[i])
df <- data.frame(x=x, y=y)
ggplot(data=df, aes(x, y)) +
  geom_line()
```



and we see that it has many local minima. Moreover, it is not differentiable as a function of β. So finding the mle is a very difficult task.

4.2.3.7 Example (4.2.10)

say X has a multinomial distribution with parameters $p_1, .., p_k$ (we assume m is known), then if we simply find the derivatives of the log-likelihood we find

$$\begin{aligned}
&\frac{\partial}{\partial p_i} l(p_1, ..p_k) = \\
&\frac{\partial}{\partial p_i}\left[c + \sum_{j=1}^{k} x_j \log p_j\right] = \\
&\frac{x_i}{p_i} = 0
\end{aligned}$$

and this system has no solution. The problem is that we are ignoring the condition $p_1 + .. + p_k = 1$. So we really have the problem

Minimize $l(p_1, .., p_k)$ subject to $p_1 + .. + p_k = 1$

One way to do this is with the method of **Lagrange multipliers**: minimize

$l(p_1, .., p_k) - \lambda(p_1 + .. + p_k - 1)$

$$\begin{aligned}
&\frac{\partial}{\partial p_i} l(p_1, ..p_k) + \lambda(p_1 + ..p_k - 1) = \\
&\frac{\partial}{\partial p_i}\left[c + \sum_{j=1}^{k} x_j \log p_j + \lambda(p_1 + ..p_k - 1)\right] = \\
&\frac{x_i}{p_i} - \lambda = 0 \\
&x_i = \lambda_p i \\
&m = \sum_{j=1}^{k} x_j = \lambda \sum_{j=1}^{k} p_j = \lambda \\
&\hat{p_i} = \frac{x_i}{m}
\end{aligned}$$

4.2.4 Properties of mle's

Maximum likelihood estimators have a number of nice properties. One of them is their invariance under transformations. That is if $\hat{\theta}$ is the mle of θ, then $g(\hat{\theta})$ is the mle of $g(\theta)$.

4.2.4.1 Example (4.2.11)

say $X_1, .., X_n \sim Ber(p)$, so we know that the mle is $\bar{X}$. Say we are interested in

$$\theta = p - q = p - (1 - p) = 2p - 1$$

the difference in proportions. Therefore $2\bar{X} - 1$ is the mle of θ.

Let's see whether we can verify that. First if $\theta = 2p - 1$ we have $p = (1 + \theta)/2$. Let $y = \sum x_i$, so

$$
\begin{aligned}
&L(p|\boldsymbol{x}) = p^y(1-p)^{n-y}\\
&L(\theta|\boldsymbol{x}) = \left(\frac{1+\theta}{2}\right)^y\left(1-\frac{1+\theta}{2}\right)^{n-y} =\\
&\left(\frac{1+\theta}{2}\right)^y\left(\frac{1-\theta}{2}\right)^{n-y} = (1+\theta)^y(1-\theta)^{n-y}/2^n\\
&l(\theta|\boldsymbol{x}) = y\log(1+\theta)+(n-y)\log(1-\theta)-n\log 2\\
&\frac{dl}{d\theta} = \frac{y}{1+\theta}-\frac{n-y}{1-\theta} = 0\\
&\hat{\theta} = 2(y/n)-1 = 2\bar{x}-1
\end{aligned}
$$

4.2.4.2 Example (4.2.12) say $X_1,..,X_n \sim N(\mu,\sigma)$. We found before that the mle of $\hat{\sigma} = \sqrt{\frac{1}{n}\sum(x_i-\bar{x})^2}$. But then the mle of the variance is

$$
\overline{\sigma^2} = \frac{1}{n}\sum(x_i-\bar{x})^2
$$

4.2.4.3 Theorem (4.2.13) Let $X_1,..,X_n$ be iid $f(x|\theta)$. Let $\hat{\theta}$ denote the mle of θ, and let $g(\theta)$ be a continuous function of θ. Under some regularity conditions on f we have

$$
\sqrt{n}\left[g(\hat{\theta})-g(\theta)\right] \to N\left(0,\sqrt{v(\theta)}\right)
$$

where $v(\theta)$ is the Rao-Cramer lower bound. That is, $g(\hat{\theta})$ is a consistent and asymptotically efficient estimator of $g(\theta)$.

4.2.4.4 Example (4.2.14) say $X_1,..,X_n \sim N(\mu,\sigma)$, σ known. We know that the mle is $\bar{x}$, which is unbiased. Now

$$
\begin{aligned}
&\phi(x;\mu) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}\\
&\log f(\mu) = K-\frac{1}{2\sigma^2}(x-\mu)^2\\
&\frac{d\log f}{d\mu} = \frac{1}{\sigma^2}(x-\mu)\\
&\frac{d^2\log f}{d\mu^2} = -\frac{1}{\sigma^2}\\
&v(\mu) = \frac{1}{n\frac{1}{\sigma^2}} = \frac{\sigma^2}{n}
\end{aligned}
$$

and so

$$\bar{X} - \mu \sim N(0, \frac{\sigma}{\sqrt{n}})$$

4.2.4.5 Example (4.2.15) say $X_1, .., X_n \sim Ber(p)$ and we want to estimate $\theta = 2p-1$. We saw above that the mle is given by $2\bar{X}-1$. We have

$$E[\bar{X}-1] = 2p-1 = \theta$$

and so $2\bar{x}-1$ is an unbiased estimator of θ. Now

$$var(2\bar{X}-1) = 2var(X_1)/n = 4p(1-p)/n =$$
$$4\left(\frac{1+\theta}{2}\right)\left(\frac{1-\theta}{2}\right)/n = (1+\theta)(1-\theta)/n$$

Note that $E[X] = p = \frac{1+\theta}{2}$. Now

$$f(x|\theta) = \left(\frac{1+\theta}{2}\right)^x \left(\frac{1-\theta}{2}\right)^{1-x}$$
$$\log f(x|\theta) = x\log(1+\theta) + (1-x)\log(1-\theta) - \log 2$$
$$\frac{d\log f(x|\theta)}{d\theta} = \frac{x}{1+\theta} - \frac{1-x}{1-\theta}$$
$$\frac{d^2\log f(x|\theta)}{d\theta^2} = -\frac{x}{(1+\theta)^2} - \frac{1-x}{(1-\theta)^2}$$
$$E\left[\frac{d^2\log f(X|\theta)}{d\theta^2}\right] = -\frac{E[X]}{(1+\theta)^2} - \frac{E[1-X]}{(1-\theta)^2} =$$
$$-\frac{\frac{1+\theta}{2}}{(1+\theta)^2} - \frac{\frac{1-\theta}{2}}{(1-\theta)^2} =$$
$$-\frac{1}{(1+\theta)(1-\theta)}$$

so $v(\theta) = (1+\theta)(1-\theta)$ and

$$2\bar{X}-1 \sim N\left(\theta, \sqrt{(1+\theta)(1-\theta)/n}\right)$$

4.2.5 Bayesian Point Estimation

We have already seen how to use a Bayesian approach to do finding point estimators, namely using the mean of the posterior distribution. Of course one could also use the median or any other measure of central tendency. A popular choice for example is the mode of the posterior distribution.

4.2.5.1 Example (4.2.16)

Let's say we have $X_1, .., X_n \sim Ber(p)$ and p~Beta(α, β), then we already know that

$$p|x_1, ..x_n \sim Beta(\alpha + \sum x_i, n - \sum x_i + \beta)$$

and so we can estimate p as follows:

- Mean $\hat{p} = \frac{\alpha + \sum x_i}{\alpha + \beta + n}$
- Median $\hat{p} = qbeta(0.5, \alpha + \sum x_i, n - \sum x_i + \beta)$
- Mode

This is the point where the posterior density has its maximum, and it is easy to verify that $\hat{p} = \frac{\alpha + \sum x_i - 1}{\alpha + \beta + n + 3}$
As $k \to \infty$ the posterior mean and mode clearly approach k/n. In fact so does the median, though that is somewhat more complicated to show.

4.2.5.2 Example (4.2.17)

Let's say we have $X_1, .., X_n \sim N(\mu, \sigma)$ and we want to estimate both μ and σ. So we need priors on both parameters:

- μ: we use the **improper** prior $g(\mu) = 1$
- σ: we use Jeffrey's prior $pi(\sigma) = 1/\sigma$

and we will assume that the priors for μ and σ are independent. We then get the joint prior on (μ, σ) to be proportional to $1/\sigma$. Therefore

$$\begin{aligned}
&f(\boldsymbol{x}|\mu, \sigma) = (2\pi\sigma^2)^{-n/2} \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2\right\} \\
&f(\boldsymbol{x}, \mu, \sigma) = (2\pi\sigma^2)^{-n/2} \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2\right\} \times \frac{1}{\sigma} = \\
&(2\pi)^{-n/2}\sigma^{-n-1} \exp\left\{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^{n}(x_i - \bar{x})^2 + n(\bar{x} - \mu)^2\right]\right\} = \\
&(2\pi)^{-n/2}\sigma^{-n-1} \exp\left\{-\frac{(n-1)s^2}{2\sigma^2}\right\} \exp\left\{-\frac{n(\mu - \bar{x})^2}{2\sigma^2}\right\}
\end{aligned}$$

therefore

$$\begin{aligned}
&f(\mu, \sigma|\boldsymbol{x}) \propto \sigma^{-n-1} \exp\left\{-\frac{(n-1)s^2}{2\sigma^2}\right\} \exp\left\{-\frac{n(\mu - \bar{x})^2}{2\sigma^2}\right\} \\
&f(\mu|\boldsymbol{x}, \sigma) \propto \exp\left\{-\frac{n(\mu - \bar{x})^2}{2\sigma^2}\right\}
\end{aligned}$$

and so $\mu|\boldsymbol{x},\sigma \sim N(\bar{x},\sigma^2/n)$, and so if we use the mean of the posterior distributions we get the sample mean as the estimator of the population mean.

How about σ? The marginal of σ^2 turns out to be a *scaled inverse*-χ^2 distribution, that is the distribution of $1/Z$ where $Z \sim \chi^2$ and its mean is the sample standard deviation s^2.

We see that with these priors Bayesian and Frequentist (maximum likelihood) estimators are the same. If we use these "flat" priors that often turns out to be the case.

4.3 A Longer Example - Estimation

Say $X_1, .., X_n$ are iid F with

$$f(x|a) = ax^{a-1},\, 0 < x < 1,\, a > 0$$

(or simply $X \sim Beta(a,1)$). Note that the cdf is given by $F(x) = x^a; 0 < x < 1$.

First let's find the method of moments estimator and the maximum likelihood estimator of a:

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

so the method of moments estimator is $\hat{a}_1 = \frac{\bar{x}}{1-\bar{x}}$.

$$\begin{aligned}
&f(\boldsymbol{x}|a) = \prod_{i=1}^n (ax_i^{a-1}) = a^n \prod_{i=1}^n (x_i^{a-1}) \\
&l(a|\boldsymbol{x}) = n\log a + (a-1)\sum_{i=1}^n \log x_i \\
&\frac{dl}{da} = \frac{n}{a} + \sum_{i=1}^n \log x_i = 0 \\
&\hat{a}_2 = -\frac{n}{\sum_{i=1}^n \log x_i} = n/T
\end{aligned}$$

where $T = -\sum \log x_i$.

Next we find the Bayes estimator of a if the prior is Exp(1).

$$\begin{aligned}
&f(\boldsymbol{x},a) = a^n \prod_{i=1}^n (x_i^{a-1}) e^{-a} = \\
&a^n \exp\left\{(a-1)\sum_{i=1}^n \log x_i - a\right\} = \\
&a^n \exp\left\{(a-1)(-T) - a\right\} = \\
&a^n \exp\left\{T - (T+1)a\right\} =
\end{aligned}$$

the marginal is

$$
\begin{aligned}
&m(\boldsymbol{x}) = \int_0^\infty a^n \exp\{T-(T+1)a\}\,da = \\
&\frac{e^T}{(T+1)^{n+1}}\int_0^\infty [(T+1)a]^n \exp\{-(T+1)a\}\,[(T+1)da] = \\
&\frac{e^T}{(T+1)^{n+1}}\int_0^\infty x^n \exp\{-x\}\,dx = \\
&\frac{e^T}{(T+1)^{n+1}}\Gamma(n+1)
\end{aligned}
$$

and so the posterior distribution is

$$f(a|\boldsymbol{x}) = \frac{a^n \exp\{T-(T+1)a\}}{\frac{e^T}{(T+1)^{n+1}}\Gamma(n+1)} = \frac{(T+1)^{n+1}}{\Gamma(n+1)}a^n e^{-(T+1)a}$$

so $a|\boldsymbol{x} \sim Gamma(n+1, \frac{1}{T+1})$

Finally we need to “extract” on number from the posterior density. We can again use either

- mean: $\hat{a} = \frac{n+1}{T+1}$
- median: $\hat{a} = qgamma(0.5, n+1, \frac{1}{T+1})$
- mode $\hat{a} = \frac{n}{T+1}$

Note that for this prior the estimators from the Bayesian method are essentially the same as the mle.

Say instead we have some prior knowledge that $a \sim N(1,1)$. Now

$$m(\boldsymbol{x}) = \int_0^\infty a^n \exp\{T-(T+1)a\}\frac{1}{\sqrt{2\pi}}e^{-(a-1)^2/2}da$$

and this integral seems difficult to evaluate. There is a solution that still works, though:

$$
\begin{aligned}
&\frac{d}{da}\left[\log\frac{f(\boldsymbol{x},a)}{m(\boldsymbol{x})}\right] = \frac{d}{da}\log f(\boldsymbol{x},a) = \\
&\frac{d}{da}\log\left[a^n \exp\{T-(T+1)a\}\frac{1}{\sqrt{2\pi}}e^{-(a-1)^2/2}\right] = \\
&\frac{d}{da}\left[n\log a - \log\sqrt{2\pi} + (a-1)T - \frac{1}{2}(a-1)^2\right] = \\
&\frac{n}{a} + T - (a-1) = 0
\end{aligned}
$$

so if we use the mode of the posterior density as our estimator we don’t need to find $m(\boldsymbol{x})$.

What properties do these estimators have?

Unbiasedness

is $\bar{x}/(1-\bar{x})$ unbiased for a? To find out we would need to first find the density of $\bar{x}$, but in this case that is not possible in this generality. Instead we can run a simulation:

```r
n <- 20; a0 <- 5; B <- 10000
x <- matrix(rbeta(n*B, a0, 1), ncol=n)
xbar <- apply(x, 1, mean)
xhat <- xbar/(1-xbar)
round(mean(xhat), 2)
```

```
## [1] 5.24
```

this seems to suggest that the estimator is biased for larger a.

How about the mle? First note that

$$P(-\log X_i \le x) = P(X_i > e^{-x}) = 1 - e^{-ax}$$

so $-\log X_i \sim Exp(a)$. and therefore

$$T = -\sum_{i=1}^n \log X_i \sim Gamma(n, 1/a)$$

which means

$$\begin{array}{l}
E[\hat{a}_2] = E[n/T] = \\
\int_0^\infty \frac{n}{t}\frac{a^n}{\Gamma(n)} t^{n-1} e^{-at} dt = \\
\frac{na}{n+1}\int_0^\infty \frac{a^{n-1}}{\Gamma(n-2)} t^{(n-1)-1} e^{-at} dt = \frac{na}{n+1}
\end{array}$$

so the mle is almost unbiased.

Sufficiency:

$$f(\boldsymbol{x}|a) = a^n e^{-(a-1)T}$$

so the mle is a sufficient statistic for a.

Ancillary Statistic

$$P(T < x) = \int_0^x \frac{a^n}{\Gamma(n)} t^{n-1} e^{-at} dt = \int_0^{ax} \frac{1}{\Gamma(n)} z^{n-1} e^{-z} dz$$

using the change of variables z=at. Therefore

$$P(aT < x) = P(X < x/a) = \int_0^x \frac{1}{\Gamma(n)} z^{n-1} e^{-z} dz$$

So the distribution of aT does not depend on a, it is an ancillary statistic.

Consistency:

From the WLLN we know that $\bar{x} \to E[X_1] = a/(a+1)$, so

$$\begin{aligned}
&P\left(|\frac{\bar{X}}{1-\bar{X}} - a| < \epsilon\right) = \\
&P\left(a - \epsilon < \frac{\bar{X}}{1-\bar{X}} < a + \epsilon\right) = \\
&P\left(\frac{a-\epsilon}{1+a-\epsilon} < \bar{X} < \frac{a+\epsilon}{1+a+\epsilon}\right) = \\
&P\left(\frac{a-\epsilon}{1+a-\epsilon} - \frac{a}{a+1} < \bar{X} - \frac{a}{a+1} < \frac{a+\epsilon}{1+a+\epsilon} - \frac{a}{a+1}\right) = \\
&P\left(\frac{-\epsilon}{(1+a-\epsilon)(a+1)} < \bar{X} - \frac{a}{a+1} < \frac{\epsilon}{(1+a+\epsilon)(a+1)}\right) \ge \\
&P\left(-M_a\epsilon < \bar{X} - \frac{a}{a+1} < M_a\epsilon\right) = \\
&P\left(|\bar{X} - \frac{a}{a+1}| < M_a\epsilon\right) \to 1
\end{aligned}$$

as $n \to \infty$, where

$$M_a = \max\{\frac{1}{(1+a-\epsilon)(a+1)}, \frac{1}{(1+a-\epsilon)(a+1)}\}$$

so we see that the method of moments estimator is a consistent estimator of a.

Note: it is in general not true that if $X_n \to x$ in probability then $g(X_n) \to g(x)$ for any function g.

How about the mle? Again from the WLLN we have

$$-1/n \sum \log X_i \to E[-\log X_1] = 1/a$$

Now

$$
\begin{aligned}
&P\left(|\hat{a}_2 - a| < \epsilon\right) = \\
&P\left(a - \epsilon < \frac{-n}{\sum \log X_i} < a + \epsilon\right) = \\
&P\left(\frac{1}{a+\epsilon} < -\frac{1}{n}\sum \log X_i < \frac{1}{a-\epsilon}\right) = \\
&P\left(\frac{1}{a+\epsilon} - \frac{1}{a} < -\frac{1}{n}\sum \log X_i - \frac{1}{a} < \frac{1}{a-\epsilon} - \frac{1}{a}\right) = \\
&P\left(\frac{-\epsilon}{a(a+\epsilon)} < -\frac{1}{n}\sum \log X_i - \frac{1}{a} < \frac{\epsilon}{a(a-\epsilon)}\right) \ge \\
&P\left(-M_a\epsilon < -\frac{1}{n}\sum \log X_i - \frac{1}{a} < M_a\epsilon\right) = \\
&P\left(| - \frac{1}{n}\sum \log X_i - \frac{1}{a}| < M_a\epsilon\right) \to 1
\end{aligned}
$$

as $n \to \infty$, where

$$M_a = \max\{\frac{1}{a(a-\epsilon)}, \frac{1}{a(a+\epsilon)}\}$$

and so the mle is a consistent estimator of a as well.

Relative Efficiency:

The relative efficiency of the two estimators is the ratio of their variances, unfortunately the variance of the method of moments estimator can not be calculated directly. For any specific case of n and a we could use simulation to find the variance.

As for the variance of the MLE we find

$$E[(\hat{a}_2)^2] = \int_0^\infty (\frac{n}{t})^2 \frac{a^n}{\Gamma(n)} t^{n-1} e^{-at} dt = \frac{n^2 a^2}{(n-1)(n-2)}$$

and so

$$var(\frac{n-1}{n}\hat{a}_2) = \frac{a^2}{n-2}$$

Rao-Cramer lower bound:

$$
\begin{aligned}
&f(x|a) = ax^{a-1} \\
&\log f(x|a) = \log a + (a-1)\log x \\
&\frac{d \log f(x|a)}{da} = 1/a + \log x \\
&\frac{d^2 \log f(x|a)}{da^2} = -1/a^2 \\
&- E\left[\frac{d^2 \log f(X|a)}{da^2}\right] = 1/a^2
\end{aligned}
$$

and so for any unbiased estimator T we have $var(T) \geq a^2/n$, so the MLE does not achieve the lower bound, although the MLE is **asymptotically efficient.**

Robustness: in this case $0 < x < 1$, so robustness is not an issue.

5 Hypothesis Testing

5.1 Hypothesis Testing Basics

5.1.1 Basic Idea

For a more detailed discussions of issues arising in hypothesis testing see my page at academic.uprm.edu/wrolke/esma3101/hyptest.html.

For a talk I gave in the Department seminar on the controversy of hypothesis testing see academic.uprm.edu/wrolke/research/WhatiswrongwithHT.density

A hypothesis is a statement about a population parameter. In its most general form it is as follows: we have data $x_1, .., x_n$ from some density $f(x|\theta)$. We want to test

$$H_0 : \theta \in \Theta_0 \text{ vs } H_a : \theta \notin \Theta_0$$

for some subset of the parameter space Θ_0.

5.1.1.1 Example (5.1.1) X~Ber(p), $\Theta = [0, 1]$, $\Theta_0 = \{0.5\}$, so we are testing whether p=0.5

5.1.1.2 Example (5.1.2) $X \sim N(\mu, \sigma)$, $\Theta = \{(x, y) : x \in R, y > 0\}$, $\Theta_0 = \{(x, y) : x > 100, y > 0\}$, so we are testing whether $\mu > 100$.

In addition to the null hypothesis we usually (but not always) also write down the alternative hypotheses H_a, usually (but not always) the complement of Θ_0. So a hypothesis test makes a choice between H_0 and H_a.

A hypothesis that "fixes" the parameter ($\theta = \theta_0$) is called **simple**, otherwise it is called **composite** (for example $\theta > \theta_0$)

A complete hypothesis test should have all of the following parts:

1) Parameter
2) Method
3) Assumptions

4) Type I error probability α

5) Null hypothesis H_0

6) Alternative hypothesis H_a

7) Test statistic

8) Rejection region

9) Conclusion

5.1.1.3 Example (5.1.3)

Over the last five years the average score in the final exam of a course was 73 points. This semester a class with 27 students used a new textbook, and the mean score in the final was 78.1 points with a standard deviation of 7.1.

Question: did the class using the new text book do (statistically significantly) better?

For this specific example the complete hypothesis test might look as follows:

1) Parameter: mean

2) Method: one-sample t

3) Assumptions: normal data or large sample

4) $\alpha = 0.05$

5) $H_0 : \mu_0 = 73$

6) $H_a : \mu_0 > 73$

7)

$T = \sqrt{n}\frac{\bar{x}-\mu_0}{s} = 3.81$

8) reject H_0 if $T > qt(1-0.05, 26) = 1.706$

9) T = 3.81 > 1.706, so we reject the null hypothesis, it appears that the mean score in the final is really higher.

In the 9 parts of a hypothesis test, the first 6 (at least in theory) should be done **before** looking at the data. The following is not allowed: say we did a study of students at the Colegio. We asked them many questions. Afterwards we computed correlation coefficients for all the pairs of variables and found a high correlation between “Income” and “GPA”. Then we carried out a hypothesis test $H_0 : \rho = 0$ vs $H_a : \rho \neq 0$.

The problem here is that this hypothesis test was suggested to us by the data, but (most standard) hypothesis tests only work as advertised if the hypotheses are formulated without consideration of the data.

Go back to our example of the new textbook. Here we have the following:

Correct: we pick $H_a : \mu > 73$ because we want to proof that the new textbook works better than the old one.

Wrong: we pick $H_a : \mu > 73$ because the sample mean score was 78.1, so if anything the new scores are higher than the old ones.

5.1.2 Type I and Type II errors

When we carry out a hypothesis test in the end we always face one of the following situations:

	State of	Nature
	H_0 is true	H_0 is false
accept H_0	OK	type II error
reject H_0	type I error	OK

In statistics when we do a hypothesis test we decide ahead of time what we are willing to accept as a type I error α, and then accept whatever the type II error β is. Generally, if you make α smaller, thereby reducing the probability of falsely rejecting the null hypothesis you make β larger, that is you increase the probability of falsely accepting a wrong null hypothesis. The only way to make both α and β smaller is by increasing the sample size n.

How do you choose α? This in practice is a very difficult question. What you need to consider is the consequences of the type I and the type II errors.

Many fields such as psychology, biology etc. have developed standards over the years. The most common one is $\alpha = 0.05$, and we will use this if nothing else is said.

5.1.3 p-value

In real live a frequentist hypothesis test is usually done by computing the **p-value**, that is the probability to observe the data or something even more extreme given that the null hypothesis is true.

5.1.3.1 Example (5.1.4)

p=P(mean score on final exam $> 78.2 \mid \mu = 73$)

```
1-pnorm(78.2, 73, 7.1/sqrt(27))
```

```
## [1] 7.072106e-05
```

Then the decision is made as follows:

- $p < \alpha \rightarrow$ reject H_0
- $p > \alpha \rightarrow$ fail to reject H_0

The advantage of the p value approach is that in addition to the decision on whether or not to reject the null hypothesis it also gives us some idea on how close a decision it was. If $p = 0.035 < \alpha = 0.05$ it was a close decision, if $p = 0.0001 < \alpha = 0.05$ it was not.

The p-value depends on the observed sample, which is a random variable, so it in turn is a random variable. What is its distribution?

5.1.3.2 Example (5.1.5)

say $X \sim N(\mu, 1)$ and we want to test

$$H_0 : \mu = 0 \text{ vs } H_1 : \mu > 0$$

Note: this is in fact a general example for data from a normal distribution because if we have a sample $X_1, .., X_n$ and want to do inference for μ, we immediately go to $\bar{X} \sim N(0, \sigma/\sqrt{n})$.

We use the rejection region $\{X > cv\}$ where $cv = qnorm(1 - \alpha)$. Now let $Y \sim N(\mu, 1)$, independent of X, and assume we observe X=x, then

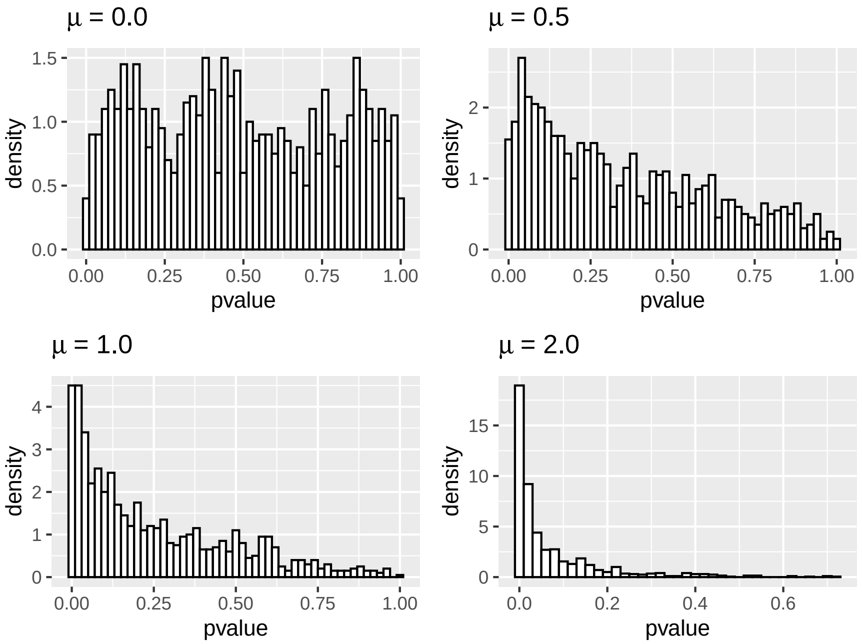
$$\begin{aligned}
&p = \text{p value } = P\left(\text{reject } H_0 | H_0 \text{ true}\right)\\
&P(Y > x|\mu = 0) = 1 - \Phi(x)\\
&F_p(t) = P(p < t) =\\
&P\left(1 - \Phi(X) < t\right) =\\
&P\left(\Phi(X) > 1 - t\right) =\\
&P\left(X > \Phi^{-1}(1 - t)\right) =\\
&1 - P\left(X < \Phi^{-1}(1 - t)\right) =\\
&1 - \Phi\left(\Phi^{-1}(1 - t)\right) =\\
&1 - (1 - t) = t
\end{aligned}$$

and so $p \sim U[0, 1]$.

So if the null hypothesis is true the distribution of the p-value is uniform [0,1]. Notice that in this derivation we made no use of the fact that Φ is a normal cdf, except that Φ^{-1} exists. So this turns out to be true in general for all continuous distributions.

Let's do a simulation to see how the p values look when the null is false:

```r
bw <- 1/50
pushViewport(viewport(layout = grid.layout(2, 2)))
df <- data.frame(pvalue=1-pnorm(rnorm(1000)))
print(ggplot(df, aes(pvalue)) +
         geom_histogram(aes(y = ..density..),
            color = "black", fill = "white", binwidth = bw) +
        labs(title=expression(mu~"= 0.0")),
  vp=viewport(layout.pos.row=1, layout.pos.col=1))
df <- data.frame(pvalue=1-pnorm(rnorm(1000, 0.5)))
print(ggplot(df, aes(pvalue)) +
        geom_histogram(aes(y = ..density..),
            color = "black", fill = "white", binwidth = bw)+
        labs(title=expression(mu~"= 0.5")),
  vp=viewport(layout.pos.row=1, layout.pos.col=2))
df <- data.frame(pvalue=1-pnorm(rnorm(1000, 1)))
print(ggplot(df, aes(pvalue)) +
        geom_histogram(aes(y = ..density..),
            color = "black", fill = "white", binwidth = bw)+
        labs(title=expression(mu~"= 1.0")),
  vp=viewport(layout.pos.row=2, layout.pos.col=1))
df <- data.frame(pvalue=1-pnorm(rnorm(1000, 2)))
print(ggplot(df, aes(pvalue)) +
        geom_histogram(aes(y = ..density..),
            color = "black", fill = "white", binwidth = bw)+
        labs(title=expression(mu~"= 2.0")),
  vp=viewport(layout.pos.row=2, layout.pos.col=2))
```

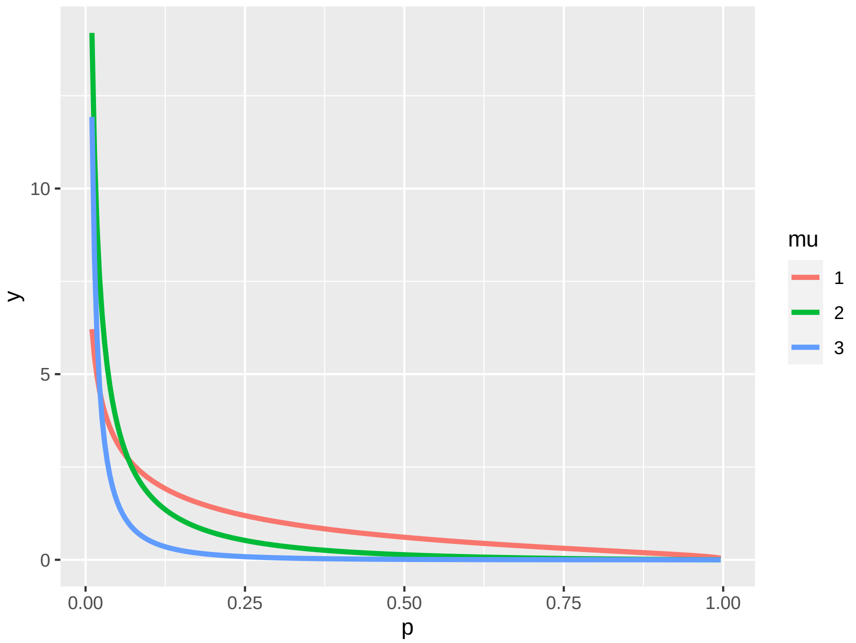


or we can just calculate it. Say the true mean is μ_1, and denote by $\phi(\cdot;\mu)$ the cdf of a normal with mean μ. Then

$$\begin{aligned}
&P(p(Y) < t) = \\
&P(1-\Phi(Y;\mu_0) < t) = \\
&P(\Phi(Y;\mu_0) > 1-t) = \\
&P(Y > \Phi^{-1}(1-t;\mu_0)) = \\
&1-P(Y < \Phi^{-1}(1-t;\mu_0)) = \\
&1-\Phi(\Phi^{-1}(1-t;\mu_0);\mu_1)
\end{aligned}$$

this is the cdf, for the density we would need to differentiate this expression. Or we can use numerical differentiation:

```
ppval <- function(t, mu)
  1-pnorm(qnorm(1-t)-mu)
t <- seq(0.01, 1, length=250)
dpval <- function(t, mu, h=10^-6)
  (ppval(t+h, mu=mu)-ppval(t, mu=mu))/h
df1 <- data.frame(p=c(t, t, t),
        y=c(dpval(t, 1), dpval(t, 2), dpval(t, 3)),
        mu=factor(rep(1:3, each=250)))
ggplot(df1, aes(p, y, color=mu)) +
  geom_line(size=1.2)
```



5.1.4 Bayesian Hypothesis Testing

Strictly speaking hypothesis testing is not a Bayesian concept. To begin with, if we wanted to test the hypothesis $H_0 : \theta = \theta_0$ we would need to start with a prior that puts some probability on the point $\{\theta_0\}$, otherwise the hypothesis will always be rejected. If we do that we can simply compute P(H_0 is true | data), and if this probability is smaller than some thresh-hold (similar to the type I error) we reject the null hypothesis.

Instead of the probability P(H_0 is true | data) we often compute the **Bayes factor**, given as follows: say $X_1, .., X_n \sim f(x|\theta)$ and $\theta \sim g$, then the posterior density is

$$g(\theta|\boldsymbol{x}) \sim L(\theta)g(\theta)$$

The belief about H_0 before the experiment is described by the prior odds ratio

$$\frac{P(\theta \in \Theta_0)}{P(\theta \in \Theta_1)}$$

and belief about H_0 after the experiment is described by the posterior odds ratio

$$\frac{P(\theta \in \Theta_0|\boldsymbol{x})}{P(\theta \in \Theta_1|\boldsymbol{x})}$$

The Bayes factor is then the ratio of the posterior to the prior odds ratios (a ratio of ratios)

5.1.5 (Jeffreys-)Lindley Paradox

Say we have $X_1, .., X_n \sim N(\mu, 1)$ and we want to test $H_0 : \mu = 0$ vs $H_a : \mu \neq 0$. Specifically, say we have n=10 and $\bar{x} = 0.75$, then the p-value is

```
n <- 10; xbar <- 0.75
2*(1-pnorm(xbar, 0, 1/sqrt(n)))
```

```
## [1] 0.01770607
```

and so we would reject the null hypothesis at the 5% level.

Now for a Bayesian analysis. As a prior let's use the following: with probability λ the null is true. Otherwise $\mu \sim N(0, 10)$. We can find the posterior probability that the null is true via simulation:

```
JL <- function(B=1e6, n=10, xbar=0.75, lambda=1/2) {
  mu <- c(rep(0, lambda*B), rnorm((1-lambda)*B, 0, 10))
  x <- rnorm(B, mu, 1/sqrt(n))
  mu <- mu[round(x, 2)==xbar]
  sum(mu==0)/length(mu)
}
JL()
```

```
## [1] 0.6418384
```

and so there is a (slight) preference for the null!

So the answer from a Frequentist and from a Bayesian analysis differ. This is often called the (Jeffrey's-) Lindley paradox.

5.1.5.1 Example (5.1.6)

Here is another example, due to Spanos (2013), from high energy physics that has been cited in the literature: We have a very large number of collisions, n=527135, which are either of type A or type B. We have k=106298 type A collisions. Theory suggests P(A)=0.2. So we want to test

$$H_0 : \pi = 0.2 \text{ vs } H_1 : \pi \neq 0.2$$

- Frequentist: with such large numbers we can use a test based on the central limit theorem:

$$
\begin{aligned}
&p = 2P(\frac{\hat{p}-p_0}{\sqrt{p_0(1-p_0)/n}} > \frac{k/n-0.2}{\sqrt{0.2\times 0.8/n}}) = \\
&P(Z>2.999) = 0.0027
\end{aligned}
$$

and so we have strong evidence that the null is false.

- Bayesian

We use the following priors on: P(H$_0$)=1/2, and the other 1/2 is spread evenly on [0,1]. So we have

$$
\begin{aligned}
&P(k|H_0) = \binom{n}{k} 0.2^k 0.8^{n-k} \\
&P(k|H_1) = \int_0^1 \binom{n}{k} p^k (1-p)^{n-k} dp = \\
&\int_0^1 \frac{n!}{(n-k)!k!} p^k (1-p)^{n-k} dp = \\
&\frac{1}{(n+1)} \int_0^1 \frac{\Gamma(n+2)}{\Gamma(n-k+1)\Gamma(k+1)} p^{k+1-1} (1-p)^{n-k+1-1} dp = \\
&\frac{1}{n+1}
\end{aligned}
$$

so the Bayes factor is

$$
\frac{P(k|H_0)P(H_1)}{P(k|H_1)P(H_0)}
$$

```
n <- 527135; k <- 106298
p0 <- dbinom(k, n, 0.2)
p1 <- 1/(n+1)
(p0*0.5)/(p1*0.5)
```

```
## [1] 8.114854
```

A Bayes factor of 8.1 would be considered some evidence in favor of the null. So again we have a disagreement!

The paradox is often used as an indictment of Frequentist statistics: Bayes and Frequentist disagree, Bayes is right, so Frequentist is wrong!

But who's to say Bayes is right?

To start, Frequentist statistics and Bayesian statistics focus on different ideas, there is no reason that they should agree (although we hope they often do).

Often whether or not there is a paradox depends on the λ:

```
JL(lambda=0.3)
```

```
## [1] 0.413442
```

and the alternative has the higher posterior!

Also is sometimes goes away when the sample size grows:

```
n <- 20
2*(1-pnorm(xbar, 0, 1/sqrt(n)))
```

```
## [1] 0.0007962302
```

```
JL(n=n)
```

```
## [1] 0.1
```

but there are examples where that doesn't happen.

5.2 Evaluating Hypothesis Tests

5.2.1 The Power of a Test

In a hypothesis test the type I error probability α is defined by

$$\alpha = P(\text{reject } H_0 | H_0 \text{ is true})$$

and is chosen by the analyst at the beginning of the test. On the other hand the type II error probability β is defined by

$$\beta = P(\text{accept } H_0 | H_0 \text{ is false})$$

5.2.1.1 Example (5.2.1)

say we have $X_1, .., X_n \sim Ber(p)$ and we want to test $H_0 : p = 0.5$ vs $H_a : p = 0.6$.

Now if the null hypothesis is wrong we would expect more successes than the 50%, so it seems reasonable to reject the null hypothesis if there are many successes, more than should happen if p=0.5. This suggests to use a test with the rejection region

$$\{S > cv\}$$

where S is the number of successes. cv here is some thresh-hold value (57? 58?) and is usually called the **critical value**. It is found as follows:

S is the number of successes in n independent Bernoulli trials with success parameter p, so S~Ber(n,p). Now

$$\begin{aligned}
&\alpha = P(S > cv|p = 0.5) = \\
&1 - P(S \le cv|p = 0.5) = \\
&1 - pbinom(cv, n, 0.5) \\
& \\
&1 - \alpha = pbinom(cv, n, 0.5) \\
&cv = qbinom(1 - \alpha, n, 0.5)
\end{aligned}$$

As a numerical example say $\alpha = 0.05$ and n=100, then

```
qbinom(1-0.05, 100, 0.5)
```

```
## [1] 58
```

Now for the type II error probability β we have to calculate the probability to fail to reject the null hypothesis if the alternative is right, that is if p=0.6:

$$\beta = P(S \le cv|p = 0.6) = pbinom(cv, n, 0.6) = pbinom(qbinom(1 - \alpha, n, 0.5), n, 0.6)$$

```
round(pbinom(qbinom(1-0.05, 100, 0.5), 100, 0.6), 4)
```

```
## [1] 0.3775
```

5.2.1.2 Example (5.2.2)

say we have $X_1, .., X_n \sim Ber(p)$ and now we want to test

$$H_0 : p = 0.5 \text{ vs } H_a : p > 0.5$$

Notice that the alternative hypothesis does not play a role in the calculation of the critical value, so again we have

$$cv = qbinom(1-\alpha, n, 0.5)$$

but when we want to find β we have a problem, we don't know what the p is. What we can do is find β as a function of p:

$$\begin{aligned}
&\beta(p) = P(S \le cv|p) = \\
&pbinom(cv, n, p) = \\
&pbinom(qbinom(1-\alpha, n, 0.5), n, p)
\end{aligned}$$

In real life we usually calculate the **power** of the test, defined by

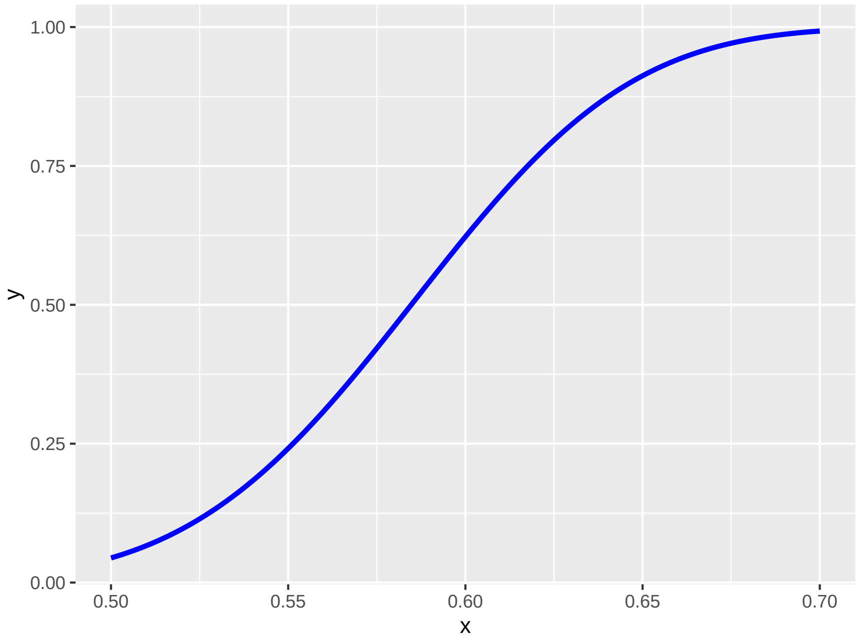
$$\text{Pow}(p) = 1 - \beta(p)$$

It has two advantages:

1) it gives the probability of correctly rejecting a false null hypothesis
2) $\text{Pow}(p_0) = \alpha$

The power curve for this test is drawn here

```r
fun <- function(p)
  1-pbinom(qbinom(1-0.05, 100, 0.5), 100, p)
ggcurve(fun=fun, A=0.5, B=0.7)
```



5.2.1.3 Example (5.2.3) say we have $X_1, ..., X_n \sim N(\mu, \sigma)$, σ known, and we want to test

$$H_0 : \mu = \mu_0 \text{ vs } H_a : \mu \neq \mu_0$$

Again $\bar{x}$ is the mle, and a reasonable test statistic is given by

$$Z = \sqrt{n}\frac{\bar{X} - \mu_0}{\sigma} \sim N(0, 1)$$

so a test might use the rejection region $\{|Z| > cv\}$:

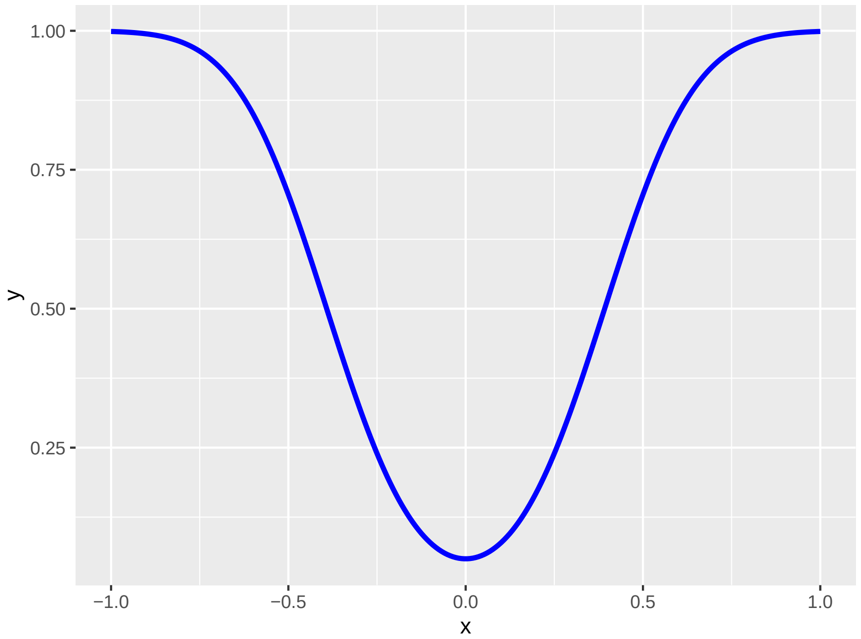
$$\begin{aligned}
&\alpha = P\left(|Z| > cv|\mu_0\right) = \\
&1 - P\left(-cv < Z < cv|\mu_0\right) = \\
&1 - (2\Phi(cv) - 1) = \\
&2(1 - \Phi(cv)) \\
&cv = \Phi^{-1}(1 - \alpha/2)
\end{aligned}$$

and now

$$\begin{aligned}
&\beta(\mu_1) = P\left(|Z| < cv|\mu_1\right) = \\
&P\left(-cv < Z < cv|\mu_1\right) = \\
&P\left(-cv < \sqrt{n}\frac{\bar{X} - \mu_0}{\sigma} < cv|\mu_1\right) = \\
&P\left(-cv < \sqrt{n}\frac{\bar{X} - \mu_1 + \mu_1 - \mu_0}{\sigma} < cv|\mu_1\right) = \\
&P\left(-cv < \sqrt{n}\frac{\bar{X} - \mu_1}{\sigma} + \sqrt{n}\frac{\mu_1 - \mu_0}{\sigma} < cv|\mu_1\right) = \\
&P\left(-cv - \sqrt{n}\frac{\mu_1 - \mu_0}{\sigma} < \sqrt{n}\frac{\bar{X} - \mu_1}{\sigma} < cv - \sqrt{n}\frac{\mu_1 - \mu_0}{\sigma}|\mu_1\right) = \\
&\Phi\left(cv - \sqrt{n}\frac{\mu_1 - \mu_0}{\sigma}\right) - \Phi\left(-cv - \sqrt{n}\frac{\mu_1 - \mu_0}{\sigma}\right)
\end{aligned}$$

The power curve for this test looks like this

```r
power.mean <- function(mu, mu0=0, sd=1, n=25, alpha=0.05) {
  cv <- qnorm(1-alpha/2)
  1-(pnorm(cv-sqrt(n)*abs(mu-mu0)/sd) -
  pnorm(-cv-sqrt(n)*abs(mu-mu0)/sd))
}
ggcurve(fun=power.mean, A=-1, B=1)
```



5.2.1.4 Example (5.2.4) Again we have $X_1, .., X_n \sim N(\mu, \sigma)$, σ known, and now we want to test

$$H_0 : \mu = \mu_0 \text{ vs } H_a : \mu \neq \mu_0$$

but this time we will use the median M as an estimator of μ.

Again a reasonable rejection region is $\{|M - \mu| > cv\}$.

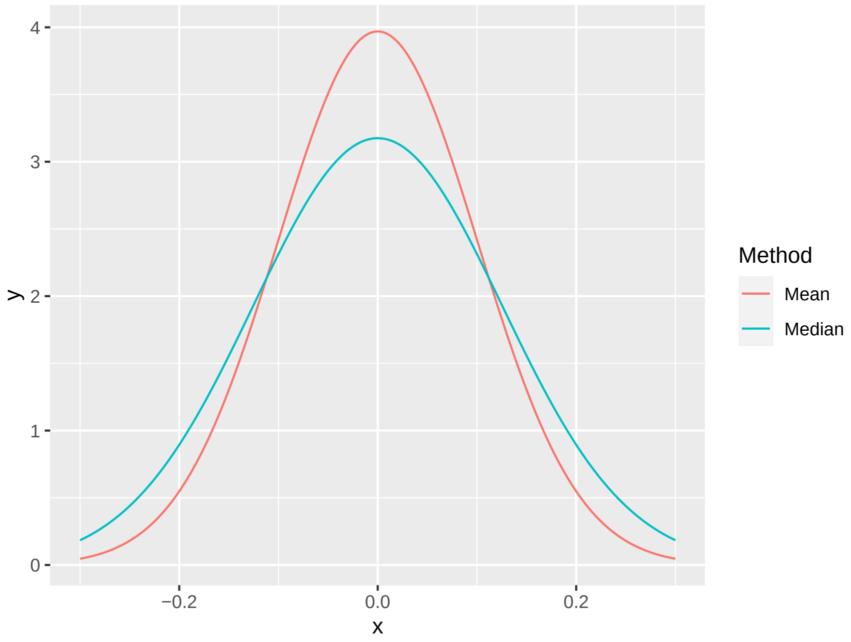
if n is odd we have $M = X_{[(n+1)/2]}$, the (n+1)/2 order statistic of $X_1, ..., X_n$, so

$$f_M(x|\mu) = \frac{n+1}{2}\binom{n}{(n+1)/2}\phi(x|\mu)\Phi(x|\mu)^{(n-1)/2}(1-\Phi(x|\mu))^{(n-1)/2}$$

where $\phi(x|\mu)$ and $\Phi(x|\mu)$ are the density and cdf of normal rv's with mean μ (and sd σ).

Let's see what the density looks like for n=99, and compare it to the one of the mean:

```
dmedian <-  function(x, mu=0, sd=1, n) {
  (n+1)/2*choose(n, (n+1)/2)*dnorm(x, mu, sd) *
    pnorm(x, mu, sd)^((n-1)/2) *
    (1-pnorm(x, mu, sd))^((n-1)/2)
}
x <- seq(-0.3, 0.3, length=250)
df <- data.frame(x=c(x, x),
    y=c(dnorm(x, sd=1/sqrt(99)), dmedian(x, n=99)),
    Method=rep(c("Mean", "Median"), each=250)             )
ggplot(data=df, aes(x, y, color=Method)) +
  geom_line()
```



Now

$$
\begin{aligned}
&\alpha = P\left(|M-\mu_0| > cv|\mu_0\right) = \\
&1 - P\left(-cv < M-\mu_0 < cv|\mu_0\right) = \\
&1 - P\left(-cv+\mu_0 < M < cv+\mu_0|\mu_0\right) = \\
&1 - \int_{cv+\mu_0}^{-cv+\mu_0} f_M(x|\mu_0)dx
\end{aligned}
$$

and cv is the solution of this equation, which of course can not ne found analytically. Instead we can find it numerically using the **integrate** function in R:

0. set cv=0
1. set cv=cv+0.01
2. find a=integrate(f,mu-cv,mu+cv)$value

3. if $a > 1-\alpha$, done, otherwise go back to 1

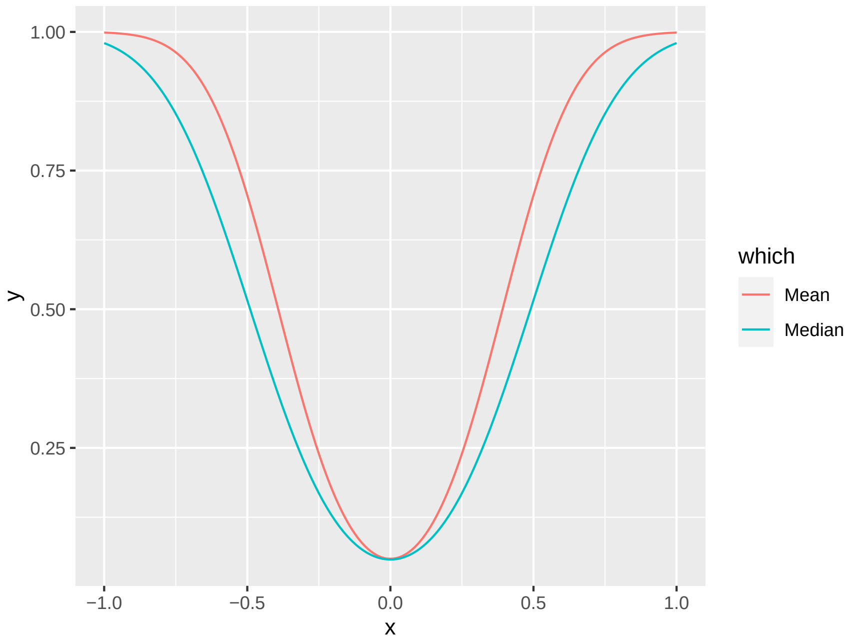
Finally

$$\beta(\mu_1) = P(\mu_0 - cv < M < \mu_0 + cv|\mu_1)$$

which we can again find using the integrate function.

```
power.median <- function(mu, mu0=0, sd=1, n=25, alpha=0.05) {
  cv <- 0
  repeat {
    cv <- cv+0.01
    a <- integrate(dmedian, -cv+mu0, cv+mu0,
                   mu=mu0, sd=sd, n=n)$value
    if(a>1-alpha) break
  }
  y <- 0*mu
  for(i in seq_along(mu))
    y[i] <- 1-integrate(dmedian, mu0-cv, mu0+cv,
               mu=mu[i], sd=sd, n=n)$value
  y
}
```

```
x <- seq(-1, 1, length=250)
y1 <- power.mean(x)
y2 <- power.median(x)
df <- data.frame(x=c(x, x),
                 y=c(y1, y2),
        which=rep(c("Mean", "Median"), each=250))
ggplot(data=df, aes(x, y, color=which)) +
  geom_line()
```



5.2.1.5 Example (5.2.5) Again we have $X_1, .., X_n \sim N(\mu, \sigma)$, σ known, and again we want to test

$$H_0 : \mu = \mu_0 \text{ vs } H_a : \mu \neq \mu_0$$

If we are worried about possible outliers we might decide to use a **trimmed mean** as our estimator: the $100p\%$ trimmed mean is defined by

$$T_p = \frac{1}{n-2np} \sum_{i=\lfloor np \rfloor}^{\lceil n(1-p) \rceil} x_{[i]}$$

In other words, to find the $100p\%$ trimmed mean eliminate the $100p\%$ smallest and largest observations and find the mean of the rest.

Note: mean=0% trimmed mean and median~50% trimmed mean. In R use the *mean(x, trim=p)* function.

Again a reasonable test has rejection region $\{|T_p - \mu| > cv\}$. But what is the distribution of T_p? This can not be done analytically for a general p, so either we do some heavy math every time we want a different p, or we need a different solution. Here is one based on simulation:

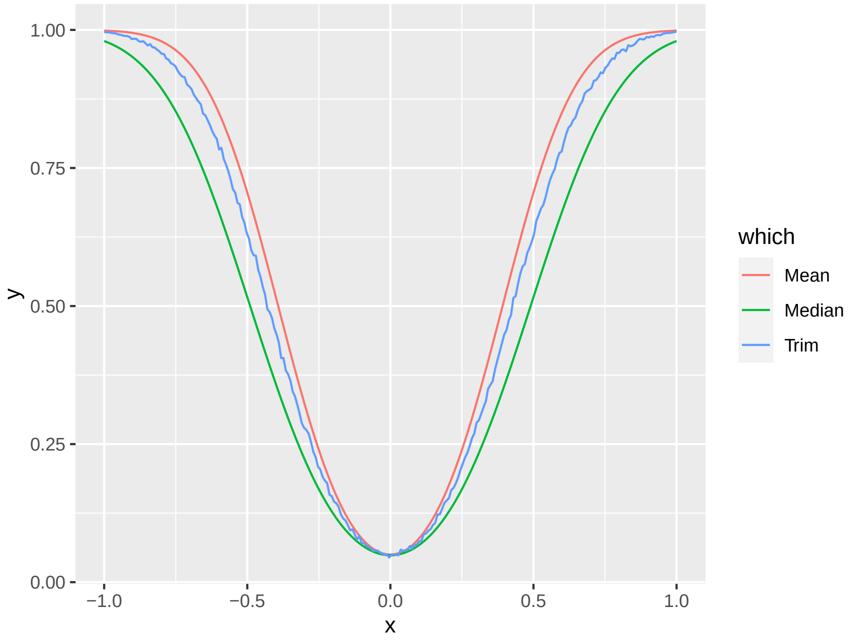
to find cv:

1) generate $Y_1, .., Y_n \sim N(\mu_0, \sigma)$, calculate T_p, call it $T_p(1)$
2) repeat 1. many times, say 10000 times
3) Find cv such that $100\alpha\%$ of the $|T_p - \mu_0|$'s are greater than cv

to find $\beta(\mu_1)$:

1) generate $Y_1, .., Y_n \sim N(\mu_1, \sigma)$, calculate T_p^*, call it $T_p^*(1)$
2) repeat 1) many times, say 10000 times
3) Find $\beta(\mu_1)$ as the proportion of T_p^* such that $|T_p^* - \mu_0| > cv$.

```r
power.trim <- function(mu, p=0.25, mu0=0, sd=1,
                       n=25, alpha=0.05, B=10000) {
  Tp <- matrix(rnorm(B*n, mu0, sd), ncol=n)
  Z <- apply(Tp, 1, mean, trim = p)
  cv <- quantile(abs(Z), 1-alpha)
  out <- 0*mu
  for(i in seq_along(mu)) {
    Tp <- matrix(rnorm(B*n, mu[i], sd), ncol=n)
    Z <- abs(apply(Tp, 1, mean, trim = p)-mu0)
    out[i] <- sum(Z > cv)/B
  }
  out
}
df <- data.frame(x=c(x, x, x),
         y=c(y1, y2, y3=power.trim(x)),
        which=rep(c("Mean", "Median", "Trim"), each=250))
ggplot(data=df, aes(x, y, color=which)) +
  geom_line()
```



as one would expect, the power of the 25% trimmed mean test is between those of the mean and the median.

5.2.1.6 Example (5.2.6) say $X_1, .., X_n \sim Ber(p)$, and we want to test

$$H_0 : p = p_0 \text{ vs. } H_a : p > p_0$$

As above a reasonable test can be based on $\{\bar{x} > cv\}$, which is equivalent to $\{\sum x_i \geq k\}$ for some integer k. Say for example n=10, p$_0$=0.5 and $\alpha = 0.1$. Then

```
sum(dbinom(10, 10, 0.5))
```

```
## [1] 0.0009765625
```

```
sum(dbinom(9:10, 10, 0.5))
```

```
## [1] 0.01074219
```

```r
sum(dbinom(8:10, 10, 0.5))
```

```
## [1] 0.0546875
```

```r
sum(dbinom(7:10, 10, 0.5))
```

```
## [1] 0.171875
```

so for k=8 $P(\text{reject } H_0|H_0 \text{ is true}) < \alpha$ and for k=7 $P(\text{reject } H_0|H_0 \text{ is true}) > \alpha$.

Because of the discreteness of the random variable it is not actually possible to find a cv such that $P(\text{reject } H_0|H_0 \text{ is true}) = \alpha$. In this case we use

$$\min\{k : P(\text{reject } H_0|H_0 \text{ is true}) < \alpha\}$$

or k=8.

There is a way to achieve α exactly: If we get $x > 7$ we reject the null, if we get $x < 7$ we fail to reject the null. If we get $x = 7$ we flip a coin that give success with probability 0.0047, and if we get a success we reject the null, otherwise we fail to reject the null. It is easy to see that now our test has exactly 0.05 as the type I error rate.

Such tests are called *randomized*. They play some role in the theory of statistics, but are not really used in practice.

5.2.2 Neyman Pearson Theory

5.2.2.1 Definition (5.2.7)

Any collection C of tests is called a *class of tests*

5.2.2.2 Example (5.2.8)

let $X_1, .., X_n \sim N(\mu, \sigma)$, σ known, and assume we want to test

$$H_0 : \mu = \mu_0 \text{ vs } H_a : \mu > \mu_0$$

a. C = "reject H_0 if $\{\bar{x} > cv\}$"
b. C = "reject H_0 if $\{\bar{x} > cv\}$ or reject H_0 if $\{\text{Median} > \text{cv}\}$"
c. C= "let T be any unbiased estimator of μ, reject H_0 if {T> cv}"

are all classes of tests.

5.2.2.3 Definition (5.2.9)

a test is called a *level α test* if

$$P(\text{reject } H_0|H_0 \text{ true}) \le \alpha$$

5.2.2.4 Definition (5.2.10) Let C be a class of tests for testing

$$H_0 : \theta \in \Theta_0 \text{ vs } H_a : \theta \in \Theta_0^c$$

A test in C with power function Pow(θ) is a *uniformly most powerful (UMP)* class C test if

$$Pow(\theta) \geq Pow'(\theta)$$

for all $\theta \in \Theta_0^c$ and every power function Pow' for every test in C.

If the class C is the class of all tests with level α, it is called the UMP level α test.

5.2.2.5 Theorem (5.2.11) Neyman-Pearson lemma

Consider testing

$$H_0 : \theta = \theta_0 \text{ vs } H_a : \theta = \theta_1$$

using a test with rejection region R given by

$$x \in R \text{ if } \frac{f(x|\theta_1)}{f(x|\theta_0)} > k$$

and

$$x \in R^c \text{ if } \frac{f(x|\theta_1)}{f(x|\theta_0)} < k$$

for some $k \geq 0$ and $\alpha = P(X \in R|\theta_0)$.

Then

a. (sufficiency) Any test of this form is a UMP level α test.

b. (necessity) If there exists a test of this form with k>0, then every UMP level α test is of this form.

Note: we have written the theorem in terms of the f, but we could of course also have used the likelihood function L.

5.2.2.6 Example (5.2.12) let $X_1, .., X_n \sim N(\mu, \sigma)$, σ known, and assume we want to test

$$H_0 : \mu = \mu_0 \text{ vs } H_a : \mu = \mu_1$$

Then

$$
\begin{aligned}
&f(\boldsymbol{x}|\mu) = \\
&(2\pi\sigma^2)^{-n/2} \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2\right\} = \\
&(2\pi)^{-n/2}\sigma^{-n-1} \exp\left\{-\frac{(n-1)s^2}{2\sigma^2}\right\} \exp\left\{-\frac{n(\mu-\bar{x})^2}{2\sigma^2}\right\}
\end{aligned}
$$

so

$$
\begin{aligned}
&\frac{f(\boldsymbol{x}|\mu_1)}{f(\boldsymbol{x}|\mu_0)} = \\
&\frac{(2\pi)^{-n/2}\sigma^{-n-1}\exp\left\{-\frac{(n-1)s^2}{2\sigma^2}\right\}\exp\left\{-\frac{n(\mu_1-\bar{x})^2}{2\sigma^2}\right\}}{(2\pi)^{-n/2}\sigma^{-n-1}\exp\left\{-\frac{(n-1)s^2}{2\sigma^2}\right\}\exp\left\{-\frac{n(\mu_0-\bar{x})^2}{2\sigma^2}\right\}} = \\
&\exp\left\{\frac{n(\mu_0-\bar{x})^2}{2\sigma^2} - \frac{n(\mu_1-\bar{x})^2}{2\sigma^2}\right\} = \\
&\exp\left\{-\frac{n}{2\sigma^2}\left[(\mu_1-\bar{x})^2-(\mu_0-\bar{x})^2\right]\right\} > k
\end{aligned}
$$

if and only if

$$-\frac{n}{2\sigma^2}\left[(\mu_1-\bar{x})^2-(\mu_0-\bar{x})^2\right] > \log k$$

if and only if

$$(\mu_1-\bar{x})^2-(\mu_0-\bar{x})^2 < -\frac{2\sigma^2}{n}\log k$$

if and only if

$$2\bar{x}(\mu_0-\mu_1)+\mu_1^2-\mu_0^2 < -\frac{2\sigma^2}{n}\log k$$

if and only if

$$
\begin{cases}
\bar{x} < -\frac{\sigma^2/n\log k-\mu_1^2/2+\mu_0^2/2}{\mu_0-\mu_1} & \text{if} \quad \mu_0 > \mu_1 \\
\bar{x} > -\frac{\sigma^2/n\log k-\mu_1^2/2+\mu_0^2/2}{\mu_0-\mu_1} & \text{if} \quad \mu_0 < \mu_1
\end{cases}
$$

here we kept track of the terms on the right, but in fact they don't matter because we don't know k anyway. Therefore a UMP level α test is of the from $\bar{x} < cv$ (if $\mu_0 > \mu_1$), where

$$\alpha = P(\bar{X} < cv)$$

Note

$$\frac{f(x|\theta_1)}{f(x|\theta_0)} = \frac{L(\theta_1|x)}{L(\theta_0|x)}$$

so a Neyman-Pearson type test is based on the ratio of the likelihood functions.

5.2.2.7 Example (5.2.13) $H_0 : \lambda = 1$ vs $H_a : \lambda = 2$

Now

$$\begin{aligned}
&f(x;\lambda) = \frac{\lambda^x}{x!}e^{-\lambda} \\
&\frac{f(x;1)}{f(x;2)} = \frac{\frac{1^x}{x!}e^{-1}}{\frac{2^x}{x!}e^{-2}} = e/2^x \\
&e/2^x > k \text{ iff} \\
&e/k > 2^x \text{ iff} \\
&x < \log(e/k)/\log(2) = [1-\log(k)]/\log(2) \\
&\alpha = P(X < cv|\lambda = 1)
\end{aligned}$$

We have

```
dpois(0, 1)
```

```
## [1] 0.3678794
```

is already larger than 0.05, so the test is to reject the null if we observe x>0.

The Neyman-Pearson lemma only discusses tests of simple vs simple hypotheses. These are very rare. It can also be shown that the theorem fails in any more generality. However, in most cases tests based on the likelihood ratio turn out to be very good, even so they are not necessarily the best.

5.3 Methods for Finding Hypothesis Tests

5.3.1 Ad-hoc Methods

The idea here is to use some estimator of the parameter of interest and then derive a test from there.

5.3.1.1 Example (5.3.1)

Say we have $X_1, ..., X_n \sim Pois(\lambda)$ and we want to test

$$H_0 : \lambda = \lambda_0 \text{ vs. } H_a : \lambda > \lambda_0$$

We know that $\bar{x}$ is the mle of λ , so a test based on $\bar{x}$ seems reasonable.

Clearly large values of $\bar{x}$ will indicate that the alternative is more likely to be true than the null hypothesis, so a reasonable rejection region is $\{\bar{x} > cv\}$. To find cv we need to solve the equation

$$\begin{aligned}
&\alpha = P(\bar{X} > cv|\lambda = \lambda_0) = \\
&1 - P(\sum X_i \le n \times cv|\lambda = \lambda_0) =
\end{aligned}$$

but under the null hypothesis

$$\sum X_i \sim \text{Pois}(n\lambda)$$

so we find cv with

```
cv <- qpois(1-alpha, n*lambda0)/n
```

The p-value of the test is

$$\begin{aligned}
&Y \sim \text{Pois}(n\lambda_0) \\
&p = P(Y \ge \sum x_i)
\end{aligned}$$

```


n <- 10; lambda0 <- 3.0; alpha <- 0.05
x <- rpois(n, 4)
c(n*lambda0, sum(x))
```

5.3.1.2 Example (5.3.2)

```
## [1] 30 43
```

```
cat("Critical value cv: ", qpois(1-alpha, n*lambda0)/n)
```

```
## Critical value cv:  3.9
```

```
cat("p value: ", 1-ppois(sum(x)-1, n*lambda0))
```

```
## p value:  0.01481952
```

Next we want to test

$$H_0 : \lambda = \lambda_0 \text{ vs. } H_a : \lambda \neq \lambda_0$$

$H_0 : \lambda = \lambda_0$ vs. $H_1 : \lambda \neq \lambda_0$

Now the critical region could be

$\bar{x} < cv_1$ or $\bar{x} > cv_2$

It turns out, though, that the choice of cv_1 and cv_2 is not unique. For example,

$cv_1 = 0$ and $cv_3 = qpois(1-\alpha, n\lambda_0)/n$

would work. One popular strategy is use $\alpha/2$ on the left and on the right, so we find

```r
cv1 <- qpois(alpha/2, n*lambda0)/n
cv2 <- qpois(1-alpha/2, n*lambda0)/n
```

The p-value now is found by

$$p = \begin{cases} 2P(Y < \sum x_i|\lambda_0) & \text{if} \quad \sum x_i < n\lambda_0 \\ 2P(Y > \sum x_i|\lambda_0) & \text{if} \quad \sum x_i > n\lambda_0 \end{cases}$$

The factor 2 is needed because the problem is “symmetric”.

The type II error probability for the one and two-sided tests are

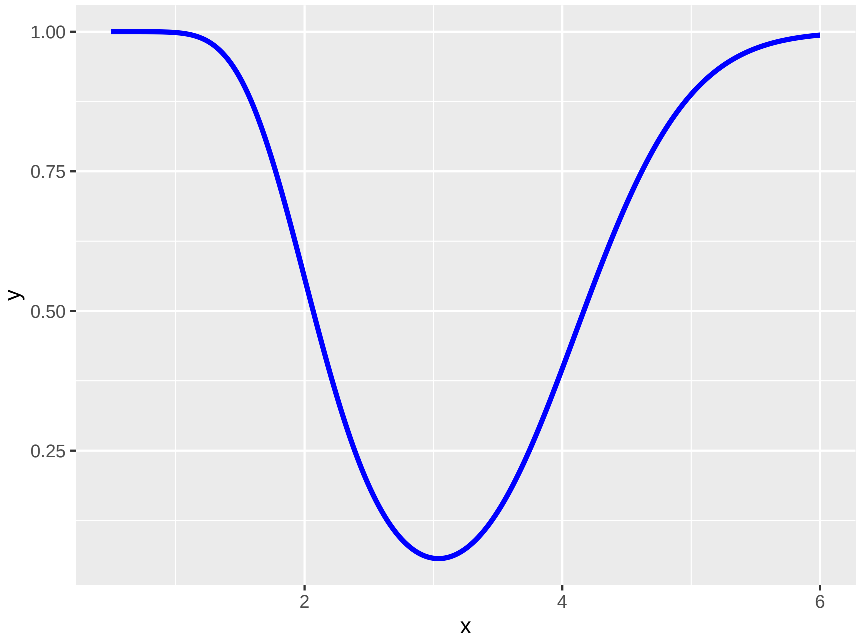
- one-sided

$$\beta(\lambda_1) = P(Y < cv|\lambda_1)$$

- two-sided

$$\beta(\lambda_1) = P(cv_1 < Y < cv_2|\lambda_1)$$

```r
power.pois <- function(n, lambda0, lrange, alpha=0.05) {
  fun <- function(lambda) {
    1-(ppois(qpois(1-alpha/2, n*lambda0), n*lambda) -
      ppois(qpois(alpha/2, n*lambda0), n*lambda))
  }
  ggcurve(fun=fun, A=lrange[1], B=lrange[2])
}
power.pois(10, 3.0, c(0.5, 6))
```



5.3.2 Likelihood Ratio Tests

Say we want to test

$$H_0 : \theta \in \Theta_0 \text{ vs } H_0 : \theta \in \Theta_0^c$$

Then the *likelihood ratio test statistic* is defined by

$$\lambda(\boldsymbol{x}) = \frac{\sup_{\Theta_0} L(\theta|\boldsymbol{x})}{\sup L(\theta|\boldsymbol{x})}$$

A *likelihood ratio test* (LRT) is any test that has a rejection region of the form

$$\{x : \lambda(x) \leq c\}$$

The constant c here is not important, it will be found once we decide on the type I error probability α. It may be better to think of this as

“reject H_0 if $\lambda(\boldsymbol{x})$ is small”

Note that the supremum in the denominator is found over the whole parameter space, so this is just like finding the mle, and then finding the corresponding value of the likelihood function.

Note that in the numerator we find the supremum over a subset of the one used in the denominator, so we always have

$$0 \leq \lambda(\boldsymbol{x}) \leq 1$$

The logic of the LRT is this:

- In the denominator we have the likelihood of observing the data we did observe, given the most favorable parameters (the mle) possible.
- In the numerator we have the likelihood of observing the data we did observe, given the most favorable parameters allowed under the null hypothesis.
- if their ratio is much smaller than 1, then there are parameters outside the null hypothesis which are much more likely than any in the null hypothesis, and we would reject the null hypothesis.

Notice the connection to the Neyman-Pearson theory: again a test is based on the likelihood ratio. While we no longer have a theorem that guarantees optimality if the hypotheses are composite, it is still reasonable to expect this type of test to be quite good.

5.3.2.1 Example (5.3.3)

Let $X_1, ..., X_n \sim N(\mu, \sigma)$, σ known. Consider testing

$$H_0 : \mu = \mu_0 \text{ vs. } H_1 : \mu \neq \mu_0$$

Here $\Theta_0 = \{\mu_0\}$ and so the numerator of $\lambda(\boldsymbol{x})$ is $L(\mu_0|\boldsymbol{x})$. For the denominator we have to find the mle, which we already know is $\bar{x}$. Therefore using (5.2.12) we find

$$L(\mu_0|\boldsymbol{x}) = (2\pi)^{-n/2}\sigma^{-n-1}\exp\left\{-\frac{(n-1)s^2}{2\sigma^2}\right\}\exp\left\{-\frac{n(\mu_0-\bar{x})^2}{2\sigma^2}\right\}$$
$$L(\bar{x}|\boldsymbol{x}) = (2\pi)^{-n/2}\sigma^{-n-1}\exp\left\{-\frac{(n-1)s^2}{2\sigma^2}\right\}\exp\left\{-\frac{n(\bar{x}-\bar{x})^2}{2\sigma^2}\right\} =$$
$$(2\pi)^{-n/2}\sigma^{-n-1}\exp\left\{-\frac{(n-1)s^2}{2\sigma^2}\right\}$$
$$\frac{L(\mu_0|\boldsymbol{x})}{L(\bar{x}|\boldsymbol{x})} = \exp\left\{-\frac{n(\mu_0-\bar{x})^2}{2\sigma^2}\right\}$$

Now an LRT test rejects the null hypothesis if $\lambda(\boldsymbol{x}) < c$ for some constant c. c depends on the choice of α. Again it is best to think of the test as rejecting H_0 if " $\lambda(\boldsymbol{x})$ is small". But

$$\lambda(\boldsymbol{x}) = \exp\left\{-\frac{n(\mu_0-\bar{x})^2}{2\sigma^2}\right\}$$

is small iff

$$-\frac{n(\mu_0-\bar{x})^2}{2\sigma^2}$$

is small iff

$$(\mu_0-\bar{x})^2$$

is large iff

$$|\bar{x} - \mu_0|$$

is large, say

$$|\bar{x} - \mu_0| > cv$$

In other words the LRT test rejects the null hypothesis if $\lambda(\boldsymbol{x}$ is small, which is equivalent to $|\bar{x} - \mu_0|$ being large.

What is the constant cv? It depends on α, namely

$$\begin{aligned}
&\alpha = P(\text{reject } H_0|H_0 \text{ true}) = \\
&P(|\bar{X} - \mu_0| > cv) = \\
&P(|\sqrt{n}\frac{\bar{X} - \mu_0}{\sigma}| > \frac{\sqrt{n} \cdot cv}{\sigma}) = \\
&2\left(1 - \Phi(\frac{\sqrt{n} \cdot cv}{\sigma})\right) \\
&cv = \Phi^{-1}(1 - \alpha/2)\frac{\sigma}{\sqrt{n}} = z_{\alpha/2}\frac{\sigma}{\sqrt{n}}
\end{aligned}$$

for example

```r
n <- 10; sigma=1; alpha <- 0.05
qnorm(1-alpha/2)*sigma/sqrt(n)
```

```
## [1] 0.619795
```

5.3.2.2 Example (5.3.4)

Let $X_1, ..., X_n$ be a sample from a population with density

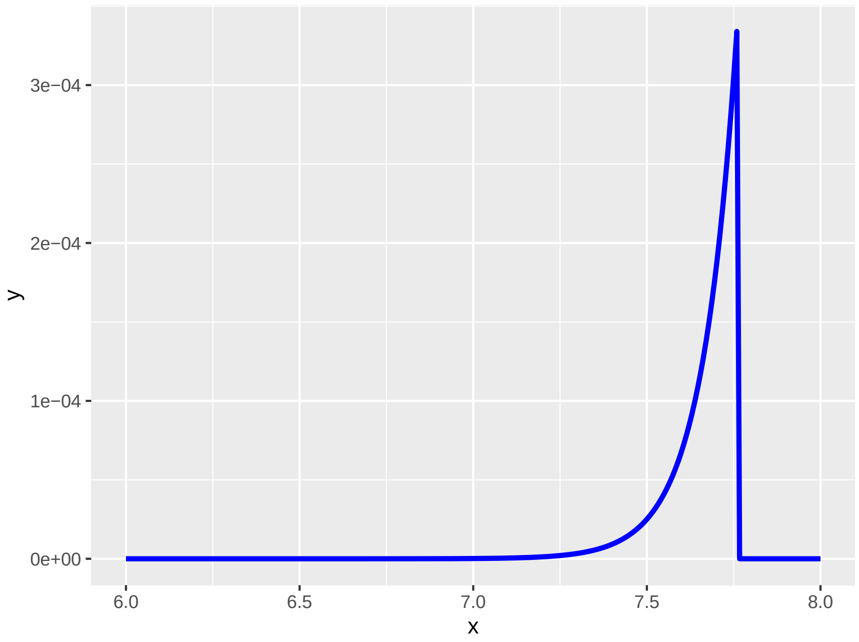
$$f(x|\theta) = e^{\theta - x}$$

if $x > \theta$ and 0 otherwise. (This is an exponential r.v with rate 1, shifted by θ). The likelihood function is given by

$$\begin{aligned}
&L(\theta|\boldsymbol{x}) = \prod \exp(\theta - x_i)I_{(\theta,\infty))}(x_i) = \\
&\exp(n\theta - \sum x_i)I_{(-\infty,x_{(1)})}(\theta)
\end{aligned}$$

Here is an example of what this function looks like:

```r
n <- 10; theta0 <- 5
x <- rexp(10, 1) + 7.5
fun <- function(t)
  exp(n*t-sum(x))*ifelse(t<min(x), 1, 0)
ggcurve(fun=fun, A=6, B=8)
```



so it is positive and increasing on $-\infty < \theta < x_{(1)}$, and then drops to 0. So clearly the mle of θ is $x_{(1)}$, the minimum. In this example it can not be found by differentiating the log-likelihood!

Let's say we want to test

$$H_0 : \theta \le \theta_0 \text{ vs. } H_1 : \theta > \theta_0$$

For the maximum of the likelihood function under the null hypothesis we have to consider two cases:

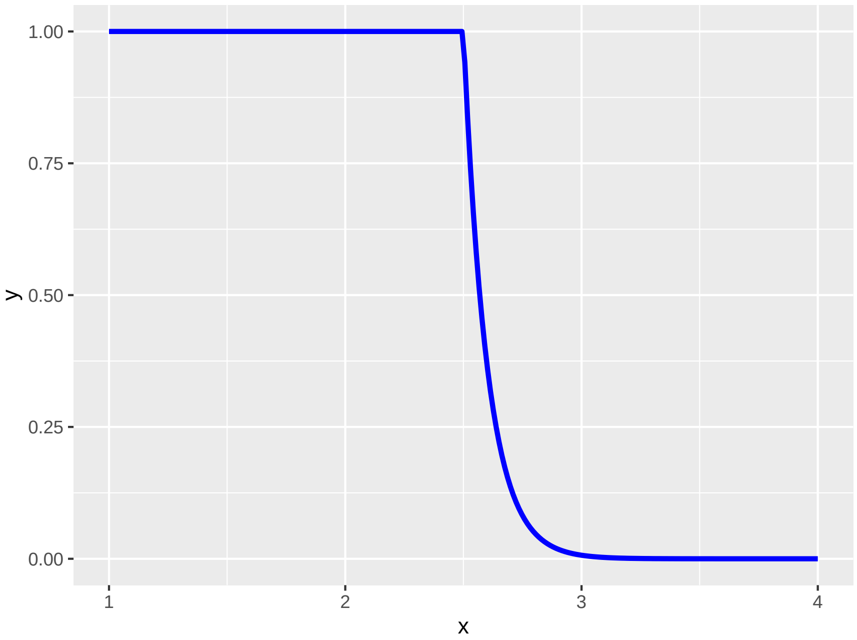
- if $x_{(1)} < \theta_0$ the maximum is at $x_{(1)}$
- if $x_{(1)} > \theta_0$ the maximum is at θ_0

Therefore the likelihood ratio statistic is given by

$$\lambda(\boldsymbol{x}) = 1 \text{ if } x_{(1)} \le \theta_0$$
$$\lambda(\boldsymbol{x}) = \exp\{n\theta_0 - nx_{(1)}\} \text{ if } x_{(1)} \le \theta_0$$

Here is an example of what this function looks like:

```
n <- 10; theta0 <- 2.5
x <- rexp(10, 1) + 7.5
fun <- function(t)
  ifelse(t<theta0, 1, exp(-n*(t-theta0)))
ggcurve(fun=fun, A=1, B=4)
```



An LRT rejects the null hypothesis if

$$\lambda(\boldsymbol{x}) \le c$$

which is clearly equivalent to a test which rejects the null hypothesis if

$$X_{(1)} \ge c$$

To determine the value of c for some specific n and α we need the distribution of $X_{(1)}$:

$$\begin{aligned}
&f(x|\theta) = \exp\{\theta - x\} I_{[\theta,\infty]}(x) \\
&F(x|\theta) = \int_\theta^x \exp\{\theta - t\} dt = \\
&-\exp\{\theta - t\} |_\theta^x = 1 - \exp\{\theta - x\}
\end{aligned}$$

and so

$$\begin{aligned}
&F_{X_{(1)}}(x) = P(X_{(1)} \le x) = 1 - P(X_{(1)} > x) = \\
&1 - P(X_1 > x, ..., X_n > x) = \\
&1 - P(X_1 > x) \cdot ... \cdot P(X_n > x) = \\
&1 - [1 - F_{X_1}(x)]^n = \\
&1 - [\exp\{\theta - x\}]^n = \\
&1 - \exp\{n(\theta - x)\} \\
&f_{X_{(1)}}(x) = n \exp\{n(\theta - x)\}; x > \theta
\end{aligned}$$

so now

$$
\begin{aligned}
&\alpha = P(X_{(1)} > cv|\theta_0) = 1 - F_{X_{(1)}}(cv) = \exp\left\{n(\theta_0 - x)\right\} \\
&cv = \theta_0 - \log(\alpha)/n
\end{aligned}
$$

The p value is

$$p = P(Y > X_{(1)}|\theta_0) = \exp\left\{n(\theta_0 - X_{(1)})\right\}$$

```
n <- 10; theta0 <- 2.5
cat("cv: ", theta0-log(alpha)/n)
```

```
## cv:  2.799573
```

```
cat("p value: ", min(1, exp(n*(theta0-min(x)))))
```

```
## p value:  2.702004e-23
```

The power of the test is given by

$$Pow(\theta_1) = P(X_{(1)} > cv|\theta_1) = \exp\left\{n(\theta_1 - \theta_0 + \log(\alpha)/n)\right\}$$

5.3.3 Asymptotic Distribution of the LRT, Wilk's Theorem

5.3.3.1 Theorem (5.3.5) Wilk

Suppose $X_1, ..., X_n$ are iid $f(x|\theta)$ and we wish to test

$$H_0 : \theta \in \Theta_0 \text{ vs. } H_1 : \theta \notin \Theta_0$$

Then under some regularity conditions the distribution of $-2\log\lambda(\boldsymbol{X})$ converges to the distribution of a $\chi^2(p)$. Here p is difference of the number of free parameters in Θ and the number of free parameters in Θ_0.

Note Let $\hat{\theta}$ be the mle of θ, and denote by $\hat{\hat{\theta}}$ the maximum under the null. Then

$$\lambda(\boldsymbol{x}) = (-2)\log\frac{L(\hat{\hat{\theta}})}{L(\hat{\theta})} = 2\left(l(\hat{\theta}) - l(\hat{\hat{\theta}})\right)$$

where l is the log-likelihood function.

5.3.3.2 Example (5.3.6) We flip a coin 1000 times and find 545 heads. Test at the 5% whether this is a fair coin.

In general we have $X_1, .., X_n \sim Ber(p)$ and we want to test

$$H_0 : p = p_0 \text{ vs } H_1 : p \neq p_0$$

Let's find the LRT test for this problem. First we have

$$L(p|\boldsymbol{x}) = p^k(1-p)^{n-k}$$

where $k = sumx_i$.

Therefore

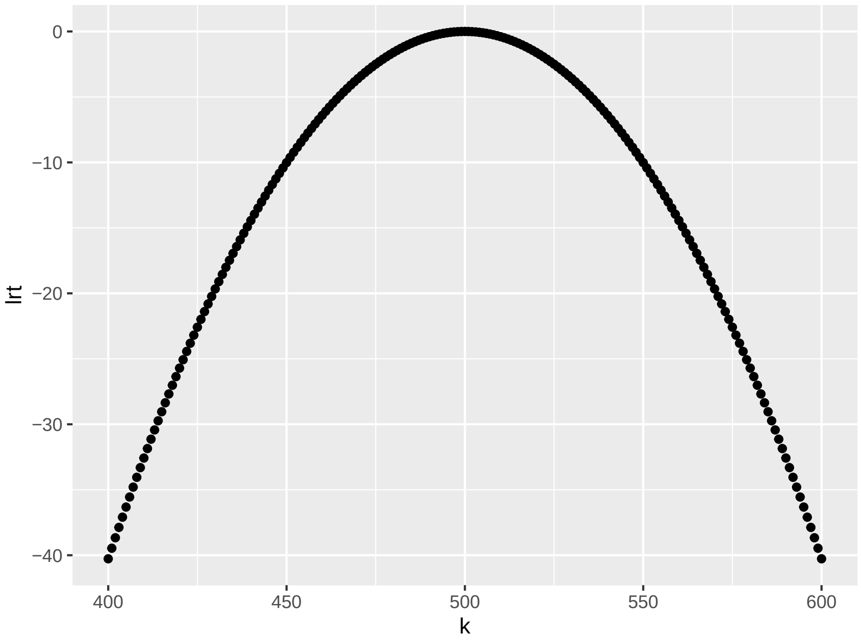
$$l(p|\boldsymbol{x}) = \log L(p|\boldsymbol{x}) = k \log p + (n-k)\log(1-p)$$

We already know that the mle is $\hat{p} = \bar{x}$ (see (4.2.4)), and so

$$\lambda(\boldsymbol{x}) = 2\left(k \log \hat{p} + (n-k)\log(1-\hat{p}) - k \log p_0 - (n-k)\log(1-p_0)\right) =$$
$$2\left(k \log \frac{\hat{p}}{p_0} + (n-k)\log \frac{1-\hat{p}}{1-p_0}\right)$$

The lrt looks like this:

```r
n <- 1000; p0 <- 0.5
k <- 400:600
lrt <- 2*(k*log(n*p0/k)+(n-k)*log(n*(1-p0)/(n-k)))
df <- data.frame(k=k, lrt=lrt)
ggplot(df, aes(k, lrt)) + geom_point()
```



it is clear that

$$\lambda(\boldsymbol{x})$$

is small iff k much smaller or much larger than np_0 iff

$$|k - np_0|$$

is large

Now let $Y = \sum X_i$, then $Y \sim Bin(n, p_0)$ and

$$\begin{aligned}
&\alpha = P(|Y - np_0| > \text{cv}) = \\
&1 - P(|Y - np_0| \le \text{cv}|p = p_0) = \\
&1 - P(-\text{cv} \le Y - np_0 \le \text{cv}) = \\
&1 - P(np_0 - \text{cv} \le Y \le \text{cv} + np_0) =
\end{aligned}$$

For our test we have n=1000, $p_0 = 0.5$, so $np_0 = 500$. Let's say we want $\alpha = 0.05$, then

```
1-diff(pbinom(500+c(-1,1)*20, 1000, 0.5))
```

```
## [1] 0.2061073
```

```
1-diff(pbinom(500+c(-1,1)*30, 1000, 0.5))
```

```
## [1] 0.05785052
```

```r
1-diff(pbinom(500+c(-1,1)*31, 1000, 0.5))
```

```
## [1] 0.04998452
```

and we find $cv = 31$, and so we reject the null hypothesis because

$$|k - np_0| = |545 - 1000 \cdot 0.5| = 45 > 31$$

We conclude that the coin is not fair.

How about using the chisquare approximation? In that case

$$T = -2 \log \lambda(\boldsymbol{X}) \sim \chi^2(1)$$

so

$$\begin{aligned}
&T = 2\left(k \log \frac{\hat{p}}{p_0} + (n-k) \log \frac{1-\hat{p}}{1-p_0}\right) = \\
&2\left[545 \log(\frac{545}{500} + (455) \log(\frac{455}{500})\right] = 8.11 \\
&\alpha = P(T > \text{cv})
\end{aligned}$$

```r
qchisq(1-0.05, 1)
```

```
## [1] 3.841459
```

and again we reject H_0, now because T=8.11>cv=3.84

Why do we have this approximation? If H_0 is true $X \sim Bin(n, p_0)$, so $E[X] = np_0$, so $X \approx np_0$.

Recall that the Taylor series expansion of log(x+1) at x=0 is

$$\log(x+1) \approx x - x^2/2$$

and so (using k instead of x) we find

$$
\begin{aligned}
&-2\log\lambda(\boldsymbol{x}) = \\
&2\left(k\log\frac{\hat{p}}{p_0} + (n-k)\log\frac{1-\hat{p}}{1-p_0}\right) = \\
&2\left(k\log\frac{k/n}{p_0} + (n-k)\log\frac{1-k/n}{1-p_0}\right) = \\
&2\left(k\log\frac{k}{np_0} + (n-k)\log\frac{n-k}{n(1-p_0)}\right) = \\
&2\left(k\log\left[(\frac{k}{np_0}-1)+1\right] + (n-k)\log\left[(\frac{n-k}{n(1-p_0)}-1)+1\right]\right) \approx \\
&2k\left[(\frac{k}{np_0}-1) - (\frac{k}{np_0}-1)^2/2\right] + \\
&2(n-k)\left[(\frac{n-k}{n(1-p_0)}-1) - (\frac{n-k}{n(1-p_0)}-1)^2/2\right] =
\end{aligned}
$$

$$
\begin{aligned}
&2k(\frac{k-np_0}{np_0}) - k(\frac{k-np_0}{np_0})^2 + \\
&2(n-k)(\frac{n-k-n(1-p_0)}{n(1-p_0)}) - (n-k)(\frac{n-k-n(1-p_0)}{n(1-p_0)})^2 = \\
&2k(\frac{k-np_0}{np_0}) - k(\frac{k-np_0}{np_0})^2 + \\
&2(n-k)(\frac{np_0-k)}{n(1-p_0)}) - (n-k)(\frac{np_0-k}{n(1-p_0)})^2 = \\
&\frac{k-np_0}{np_0(1-p_0)}\left[2k(1-p_0) - k(1-p_0)\frac{k-np_0}{np_0} + 2(n-k)p_0 - (n-k)p_0\frac{k-np_0}{n(1-p_0)}\right] = (*)
\end{aligned}
$$

Now

$$
\begin{aligned}
&2k(1-p_0) - k(1-p_0)\frac{k-np_0}{np_0} - 2(n-k)p_0 - (n-k)p_0\frac{k-np_0}{n(1-p_0)} = \\
&2k - 2kp_0 - k\frac{k-np_0}{np_0} + kp_0\frac{k-np_0}{np_0} - 2np_0 + 2kp_0 - np_0\frac{k-np_0}{n(1-p_0)} + kp_0\frac{k-np_0}{n(1-p_0)} = \\
&2(k-n_0) - k\frac{k-np_0}{np_0} + k\frac{k-np_0}{n} - p_0\frac{k-np_0}{1-p_0} + kp_0\frac{k-np_0}{n(1-p_0)} = \\
&(k-np_0)\left[2 - k\frac{1}{np_0} + k\frac{1}{n} - p_0\frac{1}{1-p_0} + kp_0\frac{1}{n(1-p_0)}\right]
\end{aligned}
$$

Noting that under the null hypothesis $k \approx np_0$, or $\frac{k}{np_0} \approx 1$, we have

$$(*) = \frac{(k-np_0)^2}{np_0(1-p_0)}\left[2 - \frac{k}{np_0} + \frac{k}{n} - \frac{p_0}{1-p_0} + \frac{kp_0}{n(1-p_0)}\right] =$$
$$\frac{(k-np_0)^2}{np_0(1-p_0)}\left[2 - \frac{k}{np_0} + \frac{k}{np_0}p_0 - \frac{p_0}{1-p_0} + \frac{k}{np_0}\frac{p_0^2}{1-p_0}\right] \approx$$
$$\frac{(k-np_0)^2}{np_0(1-p_0)}\left[2 - 1 + p_0 - \frac{p_0}{1-p_0} + \frac{p_0^2}{1-p_0}\right] =$$
$$\frac{(k-np_0)^2}{np_0(1-p_0)}\frac{1-p_0+p_0(1-p_0)-p_0+p_0^2}{1-p_0} =$$
$$\frac{(k-np_0)^2}{np_0(1-p_0)} = \left(\frac{k-np_0}{\sqrt{np_0(1-p_0)}}\right)^2 \sim \chi(1)$$

5.3.3.3 Example (5.3.7) say $X_1, .., X_n \sim Beta(\alpha, \beta)$ and we want to test

$$H_0 : \alpha = \beta \text{ vs. } H_1 : \alpha \neq \beta$$

Now

$$f(\boldsymbol{x}|\alpha,\beta) = \prod_{i=1}^n \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)})^n x_i^{\alpha-1}(1-x_i)^{\beta-1} =$$
$$\left(\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\right)^n [\prod_{i=1}^n x_i]^{\alpha-1}[\prod_{i=1}^n(1-x_i)^]\beta - 1 =$$
$$l\alpha,\beta|\boldsymbol{x}) =$$
$$n\log\Gamma(\alpha+\beta) - n\Gamma(\alpha) - n\Gamma(\beta) +$$
$$(\alpha-1)\sum_{i=1}^n \log x_i + (\beta-1)\sum_{i=1}^n \log(1-x_i)$$

For the numerator of the LRT statistic we assume the null hypothesis is true: $\alpha = \beta$, and estimate α using Newton's method. We also make use of some functions built into R: lgamma, digamma and trigamma compute the log of the gamma function and it's first and second derivative. So let g(x)=log(gamma(x)), Let $A = \sum_{i=1}^n \log x_i$ and $B = \sum_{i=1}^n \log(1-x_i)$, then

$$h(t) = ng(2t) - 2ng(t) + (t-1)(A+B)$$
$$h'(t) = 2ng'(2t) - 2ng'(t) + A + B$$
$$h''(t) = 4ng''(2t) - 2ng''(t)$$

for the denominator we have

$$\begin{aligned}
h(t,s) &= ng(t+s) - ng(t) - ng(s) + (t-1)A + (s-1)B \\
h_t(t,s) &= ng'(t+s) - ng'(t) + A \\
h_s(t,s) &= ng'(t+s) - ng'(s) + B \\
H[1,1] &= ng''(t+s) - ng''(t) \\
H[1,2] &= H[2,1] = ng''(t+s) \\
H[2,2] &= ng''(t+s) - ng''(s)
\end{aligned}$$

After computing the suprema we can find the test statistic. All of this is implemented here:

```
lrt.beta <- function (x, n=100, alpha=1, beta=1,
                      Show=TRUE) {
    l <- function(alpha, beta) {
        n*(lgamma(alpha + beta) - lgamma(alpha) -
          lgamma(beta)) + (alpha-1)*A+(beta-1)*B
    }
    if (missing(x))
        x <- rbeta(n, alpha, beta)
    else n <- length(x)
    A <- sum(log(x))
    B <- sum(log(1 - x))
    real.alpha <- alpha
    k <- 0
    repeat {
        k <- k + 1
        alphaold <- alpha
        h <- 2*n*(digamma(2*alpha)-digamma(alpha))+A+B
        hprime <- 2*n*(2*trigamma(2*alpha) -
                         trigamma(alpha))
        alpha <- alpha - h/hprime
        if (abs(alpha - alphaold) < 10^(-5))
            break
        if (k > 50)
            return(2)
    }
    if (Show) cat("Supremum under Null Hypothesis is at",
                  round(alpha, 3), "\n")
    num <- l(alpha, alpha)
    alpha <- real.alpha
    h <- c(0, 0)
    H <- matrix(0, 2, 2)
    k <- 0
    repeat {
        k <- k + 1
        xold <- c(alpha, beta)
        h[1] <- n*(digamma(alpha+beta) -
                     digamma(alpha)) + A
        h[2] <- n*(digamma(alpha + beta) -
                     digamma(beta)) + B
        H[1, 1] <- n*(trigamma(alpha + beta) -
                      trigamma(alpha))
        H[1, 2] <- n*trigamma(alpha+beta)
        H[2, 1] <- H[1, 2]
        H[2, 2] <- n*(trigamma(alpha+beta) -
                      trigamma(beta))
        xnew <- xold - solve(H) %*% cbind(h)
        if (sum(abs(xnew - xold)) < 10^(-5))
            break
        if (k > 50)
            return(2)
        alpha <- xnew[1]
        beta <- xnew[2]
    }
    if (Show) {
        cat("MLE's are: ", round(c(alpha, beta), 3), "\n")
    }
    denom <- l(alpha, beta)
    T <- -2 * (num - denom)
    if (Show)
        cat("-2log(LRT)=", round(T, 3), " crit val=",
            round(qchisq(0.95, 1), 3), "\n")
    ifelse(T > qchisq(0.95, 1), 1, 0)

}
lrt.beta()
```

```
## Supremum under Null Hypothesis is at 1.148
## MLE's are:  1.084 1.26
## -2log(LRT)= 2.013  crit val= 3.841
```

```
## [1] 0
```

```
lrt.beta(alpha=1.0, beta=1.5)
```

```
## Supremum under Null Hypothesis is at 1.272
## MLE's are:  1.185 1.687
## -2log(LRT)= 11.804  crit val= 3.841
```

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

5.3.3.4 Example (5.3.8) Let $X_1, ..., X_n$ be a sample from a population with density

$$f(x|\theta) = e^{\theta - x}$$

if $x > \theta$ and 0 otherwise. We saw before in (5.3.4) that

$$\lambda(\boldsymbol{x}) = \begin{cases} 1 & \text{f} \quad x_{(1)} \le \theta_0 \\ n \exp\left\{-n(x_{(1)} - \theta_0)\right\} & \text{f} \quad x_{(1)} > \theta_0 \end{cases}$$

therefore

$$-2\log\lambda(\boldsymbol{x}) = \begin{cases} 0 & \text{f} \quad x_{(1)} \le \theta_0 \\ 2n(x_{(1)} - \theta_0) & \text{f} \quad x_{(1)} > \theta_0 \end{cases}$$

and so

$$\begin{aligned} &P(-2\log\lambda(\boldsymbol{X}) < x) = \\ &P(2n(X_{(1)} - \theta_0) < x) = \\ &P(X_{(1)} < x/2n + \theta_0) = \\ &1 - \exp\left\{n[\theta - (x/2n + \theta_0)]\right\} = \\ &1 - e^{-x/2} \end{aligned}$$

or $-2\log\lambda(\boldsymbol{X}) \sim Exp(1/2) = \chi^2(2) \ne \chi^2(1)$, and so here we have a case where Wilk's theorem fails.

Let's consider the following general problem: we have data $\mathrm{X_1}$, .., $\mathrm{X_n}$ are iid $f(x|\theta)$ where θ is a one-dimensional parameter. We wish to use the LRT for testing

$$H_0 : \theta = \theta_0 \text{ vs } H_1 : \theta \ne \theta_0$$

Now for the denominator we need to find the mle of θ, that is we have to find the maximum of

$$\prod f(x_i;\theta)$$

or as we usually do, the maximum of

$$\sum \log\{f(x_i;\theta)\}$$

Usually we find the mle analytically, but if the density f is nice enough we can do this completely automatically:

```
lrtfun <- function (x, df, theta0, Int) {
    g <- function(theta) sum(log(df(x,theta)))
    thetahat <- optimize(g, Int, maximum =TRUE)$maximum
    chi <- 2*(g(thetahat)-g(theta0))
    1-pchisq(chi,1)
}
```

5.3.3.5 Example (5.3.9) $X_1, ..., X_n \sim N(\mu, \sigma)$, σ known

$$H_0 : \mu = \mu_0 \text{ vs } H_1 : \mu \neq \mu_0$$

```
x <- rnorm(50)
lrtfun(x, df=dnorm, theta0=0, Int=c(-5,5))
```

```
## [1] 0.1691898
```

```
lrtfun(x, df=dnorm, theta0=0.25, Int=c(-5,5))
```

```
## [1] 0.001674667
```

5.3.3.6 Example (5.3.10) $X_1, ..., X_n \sim N(\mu, \sigma)$, μ known,

$$H_0 : \sigma = \sigma_0 \text{ vs } H_1 : \sigma \neq \sigma_0$$

```
x <- rnorm(50, 0, 4)
lrtfun(x,
       df=function(x, sig) {dnorm(x, 0, sig)},
       theta0=4, Int=c(0, 10))
```

```
## [1] 0.3967211
```

```
lrtfun(x,
       df=function(x, sig) {dnorm(x, 0, sig)},
       theta0=5, Int=c(0, 10))
```

```
## [1] 0.004906238
```

5.3.3.7 Example (5.3.11) $X_1, ..., X_n \sim Gamma(\alpha, \beta)$, α known,

$$H_0 : \beta = \beta_0 \text{ vs } H_1 : \beta \neq \beta_0$$

```
x <- rgamma(25, 1, 2)
lrtfun(x,
       df=function(x, b) {dgamma(x, 1, b)},
       theta0=2, Int=c(0, 5))
```

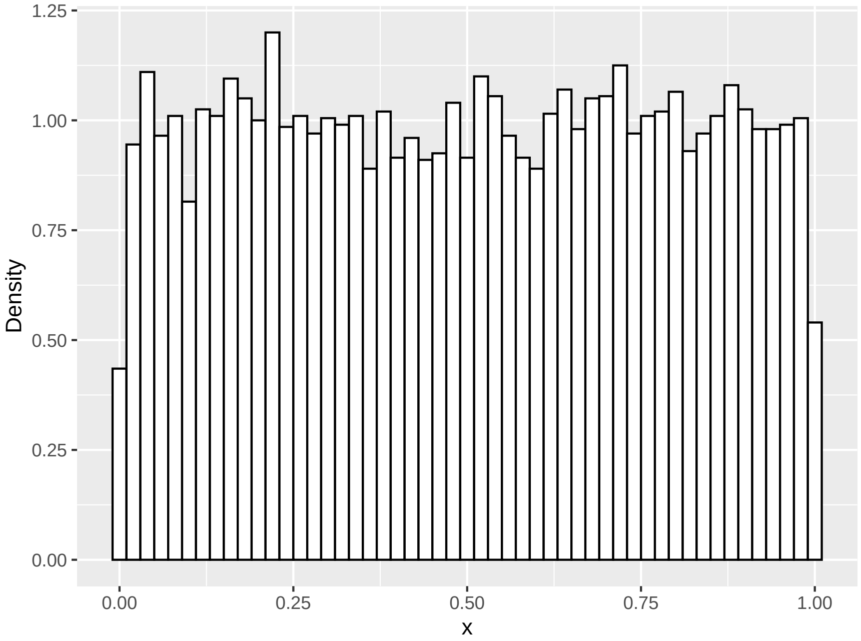
```
## [1] 0.2536346
```

```
lrtfun(x,
       df=function(x, b) {dgamma(x, 1, b)},
       theta0=2.5, Int=c(0, 5))
```

```
## [1] 0.0167621
```

Does this actually work? Let's do a small simulation, testing for the standard deviation of a normal distribution:

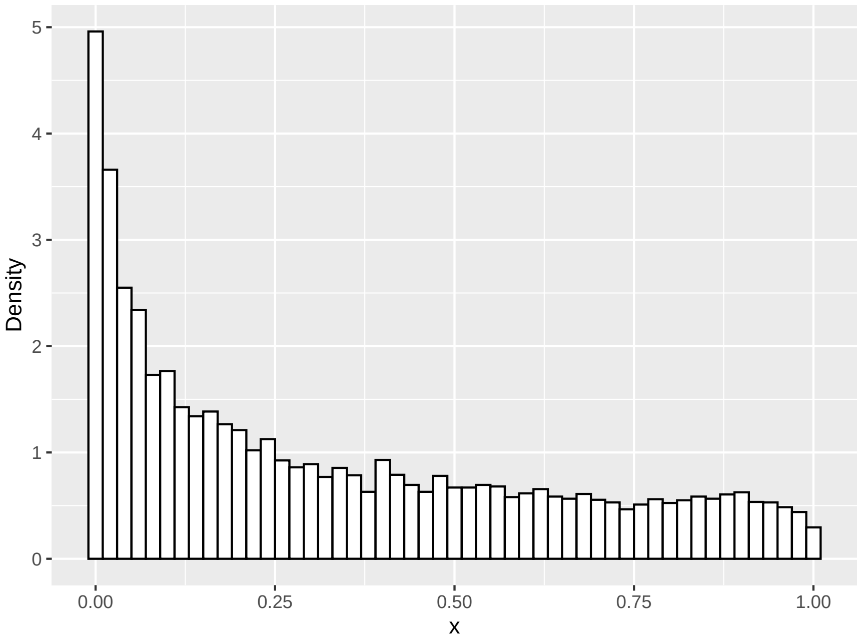
```
pvals <- rep(0, 10000)
for(i in 1:10000)
  pvals[i] <- lrtfun(x=rnorm(50, 0, 4),
                df=function(x, sig) {dnorm(x, 0, sig)},
                theta0=4, Int=c(0, 10))
df <- data.frame(x=pvals)
bw <- 1/50
ggplot(df, aes(x)) +
  geom_histogram(aes(y = ..density..),
    color = "black",
    fill = "white",
    binwidth = bw) +
    labs(x = "x", y = "Density")
```



Histogram appears flat (uniform), so the test achieves the nominal type I error probability.

And a case where the null is false:

```
pvals <- rep(0, 10000)
for(i in 1:10000)
  pvals[i] <- lrtfun(x=rnorm(50, 0, 4.5),
                df=function(x, sig) {dnorm(x, 0, sig)},
                theta0=4, Int=c(0, 10))
df <- data.frame(x=pvals)
bw <- 1/50
ggplot(df, aes(x)) +
  geom_histogram(aes(y = ..density..),
    color = "black",
    fill = "white",
    binwidth = bw) +
    labs(x = "x", y = "Density")
```



5.3.4 Large Sample Tests based on the CLT

Suppose we wish to test a hypothesis about a real-valued parameter θ, and T_n is a point estimator of θ. Say σ_n is the standard deviation of T_n. Now if some form of the CLT shows that

$$(T_n - \theta)/\sigma_n$$

converges in distribution to N(0,1) we can use this as a basis for a test.

Sometimes σ_n also depends on unknown parameters. In that case we can use an estimate of σ_n such as S_n instead.

A test based on

$$Z_n = (T_n - \theta)/S_n$$

is often called a *Wald test*.

5.3.4.1 Example (5.3.12) Let $X_1, ..., X_n \sim Ber(p)$.

Consider testing

$$H_0 : p = p_0 \text{ vs } H_1 : p \neq p_0$$

The MLE of p is $\hat{p} = \bar{x}$,so the CLT applies and states that for any p

$$\sqrt{n}\frac{\bar{x}-p}{\sqrt{p(1-p)}} \rightarrow N(0,1)$$

Of course we don't know p, but again we can estimate the p's in the denominator by $\hat{p}$ and so we get a test with the test statistic

$$Z_1 = \sqrt{n}\frac{\hat{p}-p_0}{\sqrt{\hat{p}(1-\hat{p})}}$$

and we reject the null hypothesis is $|Z_1| > z_{\alpha/2}$.

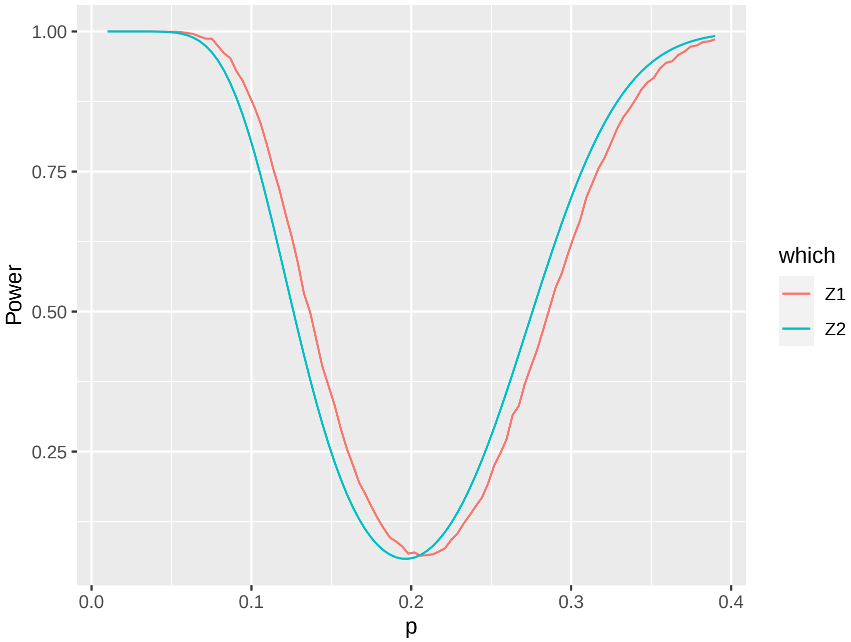
Instead of replacing the p's in the denominator by $\hat{p}$ we could also have used its value under the null hypothesis, p_0. Then another test is based on

$$Z_2 = \sqrt{n}\frac{\hat{p}-p_0}{\sqrt{p_0(1-p_0)}}$$

which rejects the null hypothesis if $|Z_2| > z_{\alpha/2}$.

Which of these tests is better? Well, that depends on the power function.

```
bernoulli.power <- function(n=100, p0=0.2,
                            B=10000, alpha=0.05) {
  crit <- qnorm(1-alpha/2)
  p <- seq(0.01, 0.39, length = 100)
  Pow1 <- 0*p
  for (i in 1:100) {
    x <- rbinom(B, size = n, prob = p[i])
    z <- sqrt(n)*(x/n-p0)/sqrt(x/n*(1-x/n))
    Pow1[i] <- mean(ifelse(abs(z)>crit, 1, 0))
  }
  Pow2 <- 1-(pbinom(n*p0+ crit*sqrt(n*p0*(1-p0)), n, p) -
             pbinom(n*p0-crit*sqrt(n*p0*(1-p0)), n, p))
  df <- data.frame(p=c(p, p),
                   Power=c(Pow1, Pow2),
                   which=rep(c("Z1", "Z2"), each=100))
  df
}
df <- bernoulli.power()
ggplot(data=df, aes(p, Power, color=which)) +
  geom_line()
```



Here the Z_1 curve is found via simulation, whereas the Z_2 power curve can be calculated directly.

As we see the power curves cross, so it depends on the true value of p which test is better.

If we suspect that p>0.2 we might prefer the Z_2 test, otherwise the Z test.

5.3.5 Bayesian Hypothesis Testing

In the Bayesian framework hypothesis tests are based on

$$P(H_0 \text{ is true } |\boldsymbol{x})$$

and

$$P(H_1 \text{ is true } |\boldsymbol{x})$$

In many ways these probabilities are exactly what a researcher desires to know, but they can only be found at the price of a prior distribution.

A Bayesian hypothesis test might then reject the null hypothesis if

$$P(H_1 \text{ is true}|\boldsymbol{x}) > 1-\alpha$$

5.3.5.1 Example (5.3.13) Let $X_1, ..., X_n \sim N(\mu, \sigma)$ and let the prior distribution on μ be $N(\mu_0, \tau)$, where σ, μ_0 and τ are known.

Say we wish to test

$$H_0 : \mu = \mu_0 \text{ vs } H_1 : \mu \neq \mu_0$$

It can be shown that then the posterior distribution $\pi(\mu|\boldsymbol{x})$ is

$$N(n\tau^2\bar{x} + \sigma^2\mu)/(n\tau^2 + \sigma^2), \sigma\tau\sqrt{(}n\tau^2 + \sigma^2))$$

Say we decide to accept

H_0 if

$$P(H_1 \text{ is true}|\boldsymbol{x}) > P(H_1 \text{ is true}|\boldsymbol{x})$$

and reject H_0 otherwise. So we reject H_0 if

$$P(\mu \leq \mu_0|\boldsymbol{x}) \geq 1/2$$

Since $\pi(\mu|\boldsymbol{x})$ is symmetric, this is true iff the mean of $\pi(\mu|\boldsymbol{x})$ is less than or equal to μ_0. Therefore H_0 is accepted if

$$\bar{x} \leq \mu_0 + \sigma^2(\mu_0 - \mu)/(n\tau^2)$$

5.3.6 Tests based on Simulation

The idea here is very simple: generate lots of simulated data from the same distribution as the real data, assuming the null hypothesis is true. For each run compute the corresponding test statistic, and then compare these values to the one from the data.

5.3.6.1 Example (5.3.14) Say X$_1$, .., X$_\text{n}$ are iid Pois(λ). We wish to test

$$H_0 : \lambda = \lambda_0 \text{ vs } H_1 : \lambda \neq \lambda_0$$

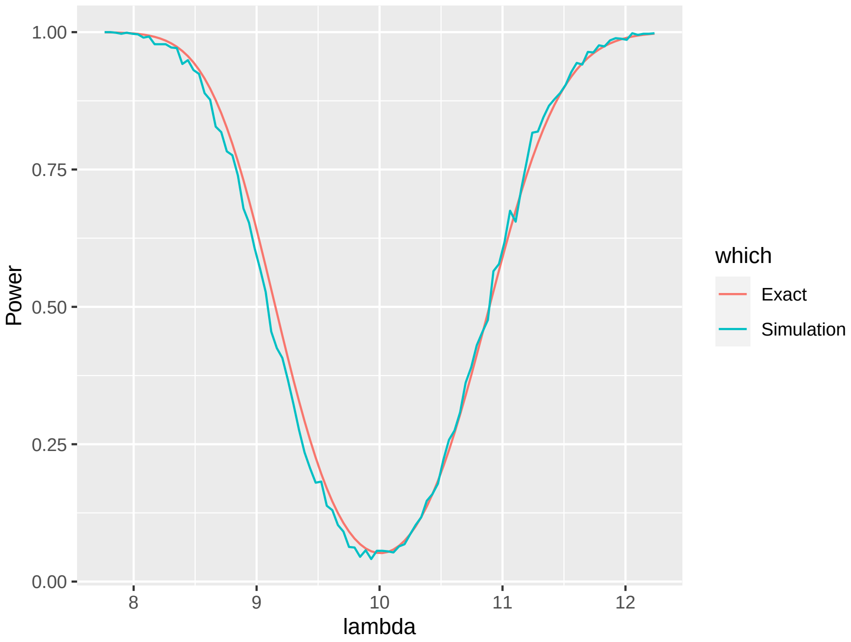
We know that the sample mean $\bar{x}$ is the mle of λ, and so we can base a test on $\bar{x}$ and we reject H_0 if $\bar{x}$ is to far from λ.

For this we compute B simulated data sets $X_1', ..., X_n' \sim Pois(\lambda_0)$, compute their sample means and find the $\alpha/2$ and the $1 - \alpha/2$ quantiles.

This will give us the critical values cv$_1$ and cv$_2$, and we reject H_0 if either

$\bar{x} < cv_1$ or $\bar{x} > cv_2$

```
poissim.power <- function(lambda0=10, n=50,
                          B=1000, alpha=0.05) {
  xbar <- apply(matrix(rpois(n*B, lambda0), B, n), 1, mean)
  cv <- as.numeric(quantile(xbar, c(alpha/2, 1-alpha/2)))
  lambda <- seq(max(0, lambda0-5*sqrt(lambda0/n)),
              lambda0+5*sqrt(lambda0/n), length = 100)
  Pow1 <- 0*lambda
  for(i in 1:100) {
    xbar <- apply(matrix(rpois(n*B, lambda[i]), B, n),
                1, mean)
    Pow1[i] <- sum(ifelse(xbar<cv[1] | xbar>cv[2], 1, 0))/B
  }
  Pow2 <- 1-(ppois(qpois(1-alpha/2, n*lambda0), n*lambda) -
        ppois(qpois(alpha/2, n*lambda0), n*lambda))
  df <- data.frame(lambda=c(lambda, lambda),
        Power=c(Pow1, Pow2),
        which=rep(c("Simulation", "Exact"), each=100))
  df
}
df <- poissim.power()
ggplot(data=df, aes(lambda, Power, color=which)) +
  geom_line()
```



The blue curve is the exact test we discussed earlier, and we see that the simulation test is just about as good.

There are also some tests already built on simulation. Say we have X_1, .., X_n from some distribution f(x) with mean μ_1 and Y_1, .., Y_m from the same distribution f(x) with mean μ_2.

We wish to test

$$H_0 : \mu_1 = \mu_2 \text{ vs } H_1 : \mu_1 \neq \mu_2$$

Now under the null hypothesis the two samples come from the exact same distribution with the same mean. So their order is completely random, and if we find the test statistic

$$T = \bar{x} - \bar{y}$$

it should have a mean of 0. We could now proceed as above, generating data from f etc., but here is another idea:

- put the x's and the y's in one vector of length n+m
- find a random permutation of this vector
- split it up into the first n and the last m numbers, calling them x' and y'

Because under H_0 any rearrangement of the data is just as likely as any other, this new data set is just as good as the original one. So we can now find T' from these observations. Because we have completely mixed the X's and the Y's we have ET'=0 for sure. Generating many T's we get an idea of "likely" values of T, and can compare the one in the real data set to them.

This is called a *permutation test*.

5.3.6.2 Example (5.3.15)

Say we have $X_1, ..., X_n \sim N(\mu_1, \sigma)$ and $Y_1, ..., Y_m \sim N(\mu_2, \sigma)$.

Now the LRT test for this is based on

$$T = \frac{\bar{x} - \bar{y}}{s\sqrt{1/n + 1/m}}$$

where

$$s^2 = \frac{(n-1)s_x^2 + (m-1)s_y^2}{n+m-2}$$

which has a t distribution with n+m-2 degrees of freedom, and we reject H_0 if $|T| > qt(1 - \alpha/2, n+m-2)$.

Let's compare this test (which is known to be optimal) to a permutation test.

The 2-sample t test is already implemented in R in **t.test**, and we can find its power either via simulation or analytically. The permutation test is implemented in

```
perm.test <- function (x, y, alpha=0.05, B = 1000) {
  n <- length(x)
  m <- length(y)
  Tdata <- mean(x) - mean(y)
  Tsim <- rep(0, B)
  for (i in 1:B) {
    xy <- sample(c(x, y), size = n + m)
    Tsim[i] <- mean(xy[1:n]) - mean(xy[(n + 1):(n + m)])
  }
  length(Tsim[abs(Tsim) > abs(Tdata)])/B
}
```

as an example:

```
x <- rnorm(50, 10, 2)
y <- rnorm(40, 10, 2)
t.test(x, y)$p.value
```

```
## [1] 0.6053234
```

```
perm.test(x, y)
```

```
## [1] 0.614
```

```
y <- rnorm(40, 11, 2)
t.test(x, y)$p.value
```

```
## [1] 0.006282209
```

```
perm.test(x, y)
```

```
## [1] 0.004
```

```
perm.power <- function(mu=c(0, 0), sigma=c(1, 1),
                       n=c(50, 50), alpha=0.05, B=1000) {
  pvals <- matrix(0, B, 2)
  for(i in 1:B) {
    x <- rnorm(n[1], mu[1], sigma[1])
    y <- rnorm(n[2], mu[2], sigma[2])
    pvals[i, 1] <- t.test(x, y)$p.value
    pvals[i, 2] <- perm.test(x, y)
  }
  c(sum(pvals[, 1]<alpha), sum(pvals[, 2]<alpha))/B
```

```r
}
mu <- seq(0, 0.8, length=10)
power <- matrix(0, 10, 3)
colnames(power) <- c("mu2", "t-test", "permutation")
power[, 1] <- mu
for(i in 1:10)
  power[i, 2:3] <- perm.power(mu=c(0, mu[i]))
power <- round(power, 3)
```

```r
kable.nice(power, do.row.names = FALSE)
```

mu2	t-test	permutation
0.000	0.056	0.057
0.089	0.065	0.069
0.178	0.154	0.149
0.267	0.259	0.259
0.356	0.443	0.441
0.444	0.616	0.617
0.533	0.760	0.760
0.622	0.878	0.879
0.711	0.936	0.938
0.800	0.972	0.971

and we see that the permutation test has a power quite similar to the t test.

5.4 A Simple Example (Exponential)

say $X_1, ..., X_n$ iid with

$$f(x) = \beta \exp(-\beta x), x > 0$$

and we want to test

$$H_0 : \beta = \beta_0 \text{ vs } H_a : \beta \neq \beta_0$$

5.4.1 Wald test

we can often find such a test by first finding the method of moments estimator: $E[X] = 1/\beta$, so $\hat{\beta}_1 = 1/\bar{x}$

so if $1/\bar{x}$ is close to β_0 we should accept the null hypothesis, otherwise we should reject it. This is of course equivalent to $\bar{x}$ is close to $1/\beta_0$. Now $var(X) = 1/\beta^2$ and so

$$\sqrt{n}\frac{\bar{X} - 1/\beta_0}{1/\beta_0} = \sqrt{n}(\beta_0\bar{X} - 1) \sim N(0,1)$$

and we reject H_0 if

$$\sqrt{n}|\beta_0\bar{x} - 1| > z_{\alpha/2}$$

5.4.2 LRT test (5.4.1)

$$\begin{aligned}
&f(\boldsymbol{x}|\beta) = \prod_{i=1}^{n} \beta e^{-\beta x_i} = \beta^n e^{-\beta \sum x_i} \\
&l(\beta|\boldsymbol{x}) = n\log\beta - \beta\sum x_i \\
&\frac{dl(\beta|\boldsymbol{x})}{d\beta} = \frac{n}{\beta} - \sum x_i = 0 \\
&\hat{\beta} = 1/\bar{x}
\end{aligned}$$

for the likelihood ratio test statistic we find

$$\begin{aligned}
&\lambda(\boldsymbol{x}) = \frac{\beta_0^n e^{-\beta_0 \sum x_i}}{(1/\bar{x})^n e^{-(1/\bar{x})\sum x_i}} = (\beta_0\bar{x})^n \, e^{n(1-\beta_0\bar{x})} \\
&-2\log\lambda(\boldsymbol{x}) = -2\left[n\log\left(\beta_0\bar{x}\right) + n(1-\beta_0\bar{x}\right] \\
&=
\end{aligned}$$

and we reject H_0 if

$$(-2)\left[n\log(\beta_0\bar{x}) + n - \beta_0\bar{x}\right] > \text{qchisq}(1-\alpha, 1)$$

How about using the LRT without the chisquare approximation? As always we have

“reject H_0 if $\lambda(\boldsymbol{x})$ is small”

iff

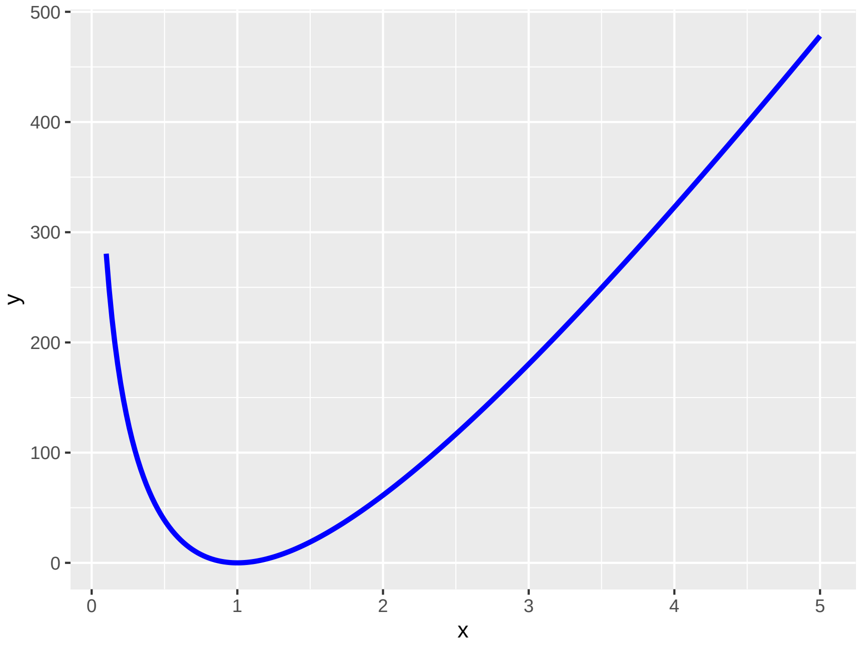
“reject H_0 if $-2\log\lambda(\boldsymbol{x})$ is large”"

iff

“reject H_0 if $\bar{x}$ is small or large”"

this can be seen by drawing the graph of the lrt as a function of $\bar{x}$:

```
n <- 100; beta0 <- 1
fun <- function(xbar)
  (-2)*(n*log(xbar*beta0)+n-beta0*n*xbar)
ggcurve(fun=fun, A=0.1, B=5)
```



so we reject H_0 if $\bar{x} < cv_1$ or $\bar{x} > cv_2$.

Now under the null hypothesis $X_i \sim Exp(\beta_0)$, so $\sum_{i=1}^n X_i \sim Gamma(n, 1/\beta_0)$ and we have

$$\alpha/2 = P(\bar{X} < cv_1) = P(\sum_{i=1}^n X_i < n \times cv_1) = \text{pgamma}(n \times cv_1, n, \beta_0)$$

and so $cv_1 = \text{qgamma}(\alpha/2, n, \beta_0)/n$. And also clearly we have $cv_2 = \text{qgamma}(1 - \alpha/2, n, \beta_0)/n$.

This solution only works because we know the distribution of a sum of exponential variables (gamma), in another example we might be stuck with using the chisquare approximation.

5.4.3 Which test is better?

for that we need to find the power of the tests:

- Wald test:

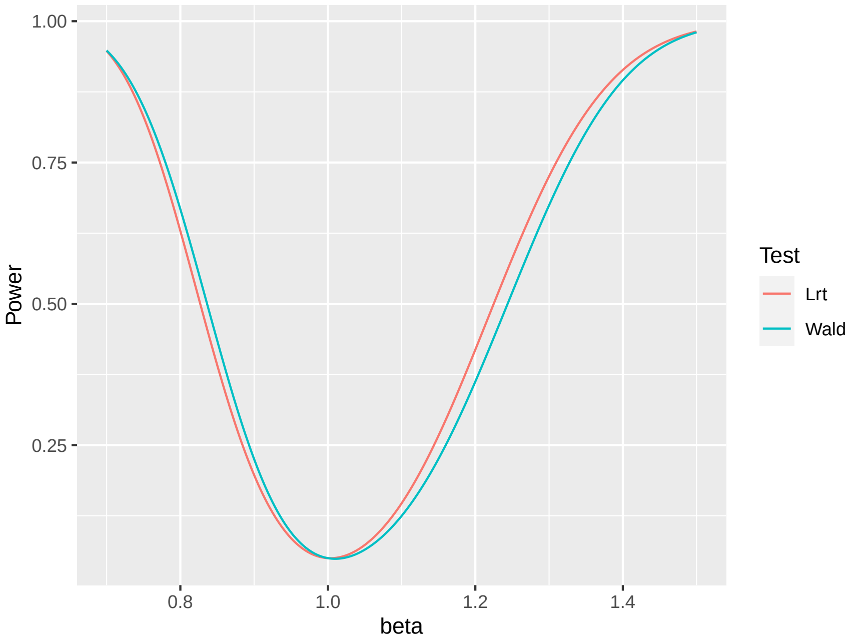
$$
\begin{aligned}
&Pow_A(\beta_1) = P\left(\sqrt{n}|\beta_0\bar{X} - 1| > z_{\alpha/2}|\beta_1\right) = \\
&1 - P\left(-z_{\alpha/2} < \sqrt{n}(\beta_0\bar{X} - 1) < z_{\alpha/2}|\beta_1\right) = \\
&1 - P\left(-z_{\alpha/2} + \sqrt{n} < \sqrt{n}\beta_0\bar{X}) < z_{\alpha/2}\beta_0|\beta_1\right) = \\
&1 - P\left([-z_{\alpha/2} + \sqrt{n}]\frac{\beta_0}{\beta_1} < \sqrt{n}\beta_0\bar{X}) < [z_{\alpha/2} + \sqrt{n}]\frac{\beta_0}{\beta_1}|\beta_1\right) = \\
=&1 - P\left([-z_{\alpha/2} + \sqrt{n}]\frac{\beta_0}{\beta_1} - \sqrt{n} < \sqrt{n}(\beta_0\bar{X} - 1) < [z_{\alpha/2} + \sqrt{n}]\frac{\beta_0}{\beta_1} - \sqrt{n}|\beta_1\right) = \\
&\Phi([z_{\alpha/2} + \sqrt{n}]\frac{\beta_0}{\beta_1} - \sqrt{n}) - \Phi([-z_{\alpha/2} + \sqrt{n}]\frac{\beta_0}{\beta_1} - \sqrt{n})
\end{aligned}
$$

- likelihood ratio test:

$$
\begin{aligned}
&Pow_B(\beta_1) = P(cv_1 < \bar{X} < cv_2|\beta_1) = \\
&P(n \cdot cv_1 < \sum X_i < n \cdot cv_2|\beta_1) = \\
&pgamma(n \cdot cv_2; n; \beta_1) - pgamma(n \cdot cv_1; n; \beta_1)
\end{aligned}
$$

here are the power graphs:

```
alpha <- 0.05
beta1 <- seq(0.7, 1.5, length = 250)
Pow1 = 1-(pnorm((qnorm(1-alpha/2) + sqrt(n)) * beta1/beta0 -
          sqrt(n)) - pnorm((-qnorm(1 - alpha/2) + sqrt(n)) *
          beta1/beta0 - sqrt(n)))
cv <- qgamma(c(alpha/2, 1 - alpha/2), n, beta0)/n
Pow2 <- 1-(pgamma(n*cv[2], n, beta1) -
             pgamma(n*cv[1], n, beta1))
df <- data.frame(beta=c(beta1, beta1),
          Power=c(Pow1, Pow2),
          Test=rep(c("Wald", "Lrt"), each=250))
ggplot(data=df, aes(beta, Power, color=Test)) +
  geom_line()
```



and so we see that it depends on the true value of β. Of course the big advantage of the LRT test here is that we were able to turn it into an exact test and so don't have to worry about the sample size.

5.5 A Longer Example - Testing

Say $X_1, .., X_n \sim Beta(a, 1)$

We want to test

$$H_0 : a = a_0 \text{ vs } H_1 : a \neq a_0$$

5.5.1 Case n=1

We already know that E[X]=a/(a+1), so $x \approx a/(a+1)$ or $a \approx x/(1-x)$. So a test could be based on the rejection region

$$x/(1-x) < c_1 \text{ or } x/(1-x) > c_2$$

But the function x/(1-x) is monotonically increasing on [0, 1], so this rejection region is equivalent to one with

$$x < c_1 \text{ or } x > c_2$$

Now

$$\begin{aligned}
&\alpha/2 = P(X < c_1) = c_1^{a_0}\\
&c_1 = (\alpha/2)^{1/a_0}\\
&\alpha/2 = P(X > c_2) = c_2^{a_0}\\
&c_2 = (1-\alpha/2)^{1/a_0}
\end{aligned}$$

For example if we wish to test a_0=0.5 at the 5% level we reject the null hypothesis if

```
a0=0.5;alpha=0.05
round(c(alpha/2,1-alpha/2)^(1/a0), 5)
```

```
## [1] 0.00063 0.95062
```

5.5.2 Case n=2

say x and y. Again we might try to use the fact that E[(X+Y)/2] = a/(1+a) But eventually we would need to find the density of X+Y. Using the convolution formula if $0 < t < 1$ this would mean finding the integral

$$\int_{t-1}^{t}[x(1-x)]^{a-1}dx$$

and this integral does not exist analytically. What can we do? One idea is to find the integral numerically:

```
dbeta2 <-function (x, a=1) {
  f <- function(x, t) {(x*(t-x))^(a-1)}
  y <- 0*x
  for(i in 1:length(x)) {
    if(x[i] <1) y[i] <- integrate(f, 0, x[i], t=x[i])$value
    else y[i] <- integrate(f,x[i]-1, 1, t=x[i])$value
  }
  a^2*y
}
pbeta2  <-  function(x, a=1) {
  y <- x
  for(i in 1:length(x))
    y[i] <- integrate(dbeta2, 0, x[i], a=a)$value
  y
}
qbeta2  <- function (p, a=1)   {
  pbeta2 <- function(x) {integrate(dbeta2, 0, x, a=a)$value}
  low <- 0
  high <- 2
  repeat {
    mid <- (low+high)/2
    fmid <- pbeta2(mid)
    if(abs(fmid-p) <0.001) break
    if(fmid<p) low <- mid
    else high <- mid
  }
  mid
}
round(c(qbeta2(0.025, 1/2), qbeta2(0.975, 1/2)), 3)
```

```
## [1] 0.031 1.594
```

Alternatively we can use simulation to find the null distribution:

```
B <- 1e4
xy <- matrix(rbeta(2*B, 0.5, 1), ncol=2)
z <- apply(xy, 1, sum)
round(quantile(z,c(0.025,0.975)), 3)
```

```
##  2.5% 97.5%
## 0.035 1.628
```

This of course works equally well for n=3, 4, … whereas the numerical solution gets much harder quickly.

5.5.3 Case large n

From the CLT we know that

$$\sqrt{n}\frac{\bar{x}-\mu_0}{\sigma_0} \rightarrow N(0,1)$$

where

$$\mu_0 = \frac{a_0}{a_0+1}$$

and

$$\sigma_0^2 = \frac{a_0}{(a_0+1)^2(a_0+2)}$$

therefore

$$cv = \mu_0 \pm z_{\alpha/2}\sigma_0/\sqrt{n} = \frac{1}{a_0+1}\left(a_0 \pm z_{\alpha/2}\sqrt{\frac{a_0}{n(a_0+2)}}\right)$$

So we have the following test: reject H_0 if $\bar{x}$ <cv_1 or $\bar{x}$ >cv_2.

5.5.4 One-sided tests

above we tested

$$H_0 : a = a_0 \text{ vs } H_1 : a \neq a_0$$

Often we want to test instead alternatives of the form

$$H_0 : a = a_0 \text{ vs } H_1 : a < a_0$$

or

$$H_0 : a = a_0 \text{ vs } H_1 : a > a_0$$

In that case choose the appropriate rejection region, with α instead of $\alpha/2$

5.5.5 p-value

if we want to quote the p-value of this test we calculate it as follows: say we observed $\bar{x} = t$ in our experiment. The p-value is the probability of repeating the experiment and observing a value of the test statistic as "unlikely" (given the null hypothesis) as that seen in the original experiment.

Say we observed $t > a_0$ and let $\bar{y}$ be the sample mean of the new experiment, then

$$p = 2P(\bar{Y} > t) = 2 - 2\Phi(\sqrt{n}\frac{t - \mu_1}{\sigma_0})$$

the "2" is because we do a two-sided test.

5.5.6 Likelihood ratio test

we saw before that the mle was given by

$$\hat{a}_2 = n/T$$

where $T = -\sum \log x_i$. So using the results of section 4.3 we have

$$
\begin{aligned}
&-2\log\lambda(\boldsymbol{x}) = \\
&2\left[l(\hat{a}_2) - l(a_0)\right] = \\
&2\left[n\log\hat{a}_2 + (\hat{a}_2 - 1)\sum_{i=1}^n \log x_i - n\log a_0 - (a_0 - 1)\sum_{i=1}^n \log x_i\right] = \\
&2\left[n\log\frac{\hat{a}_2}{a_0} - (\hat{a}_2 - a_0)\sum_{i=1}^n \log x_i\right] = \\
&2\left[n\log\frac{\hat{a}_2}{a_0} - (\hat{a}_2 - a_0)(-n/\hat{a}_2)\right] = \\
&2n\left[\log\frac{\hat{a}_2}{a_0} + \frac{a_0}{\hat{a}_2} - 1\right]
\end{aligned}
$$

and we reject H_0 if

$$-2\log\lambda(\boldsymbol{x}) > qchisq(1-\alpha, 1)$$

One problem with both these methods is that they are **large sample** methods, they rely on the CLT. Can we derive a method that also works for small samples? The basic idea of the LRT is to reject the null hypothesis if $\lambda(\boldsymbol{x})$ is small, which is the same as $(-2)\log\lambda(\boldsymbol{x})$is large. Now consider this:

$$
\begin{aligned}
&h(a) = \log\frac{a}{a_0} + \frac{a_0}{a} - 1 \\
&\frac{dh}{da} = \frac{1}{a} - \frac{a_0}{a^2} = \frac{a - a_0}{a^2} > 0
\end{aligned}
$$

if $a > a_0$. So h is decreasing on $(0, a_0)$ and increasing on (a_0, ∞), and so h is large for small or large values of a.

Now $\hat{a}_2 = n/T$, and so

$\hat{a}_2$ is small or large

iff

T is small or large

If H_0 is true $T \sim \Gamma(n, 1/a_0)$ and if we use $\alpha/2$ on the left and the right we find

$$\alpha/2 = P(T < x) = pgamma(x, n, a_0)$$

$$x = qgamma(1 - \alpha/2, n, a_0)$$

Similarly $\alpha/2 = P(T > y)$ yields $T > qgamma(1 - \alpha/2, n, a_0)$.

Note

$$\begin{aligned}
&pgamma(x, n, a) = \\
&\int_0^x \frac{a^n}{(n-1)!} t^{n-1} e^{-at} dt = \\
&\int_0^{ax} \frac{1}{(n-1)!} y^{n-1} e^{y} dy = \\
&pgamma(ax, n, 1)
\end{aligned}$$

and so

$$\begin{aligned}
&qamma(y, n, a) = x \\
&y = pamma(x, n, a) = gamma(ax, n, 1) \\
&qamma(y, n, 1) = ax \\
&qamma(y, n, a) = qamma(y, n, 1)/a
\end{aligned}$$

so we reject H_0 if

$a_0 T < qgamma(\alpha/2, n, 1)$ or $a_0 T > qgamma(1 - \alpha/2, n, 1)$

5.5.7 One-sided tests

Say we want to test $H_a : a < a_0$. Then we have the rejection region $a_0 T < qgamma(\alpha, n, 1)$

5.5.8 p-value

$$p = 2P(T > t|a = a_0) = 2(1 - pgamma(t, n, a_0))$$

if $t > a_0$.

5.5.9 Bayesian analysis

Let's use again the prior Exp(1), then we know that $a|x \sim \Gamma(n + 1, T + 1)$. A test could be designed as follows: reject H_0 if

$a_0 < qgamma(\alpha/2, n + 1, T + 1)$ or $a_0 > qgamma(1 - \alpha/2, n + 1, T + 1)$

But from the above we know that

$$a_0 < qgamma(alpha/2, n + 1, T + 1) = qgamma(\alpha/2, n + 1, 1)/(T + 1)$$

iff

$$a_0(T + 1) < qgamma(\alpha/2, n + 1, 1)$$

and we see that this is essentially the same as the test based on the mle. (except with n+1 instead of n and T+1 instead of T)

5.5.10 Power

Let's go back to the two tests based on the sample mean and the likelihood ratio. Which of these is best? That depends on the power of the test. First we have

- Wald test

$$
\begin{aligned}
&x = \mu + \frac{\sigma}{\sqrt{n}}\Phi^{-1}(\alpha/2)\\
&y = \mu + \frac{\sigma}{\sqrt{n}}\Phi^{-1}(1-\alpha/2)\\
&\mu_1 = \frac{a_1}{a_1+1}\\
&\sigma_1^2 = \frac{a_1}{(a_1+1)^2(a_1+2)}\\
&1 - Pow(a_1, n) = P(x < \bar{X} < y|a_1) =\\
&P\left(\sqrt{n}\frac{x-\mu_1}{\sigma_1} < \sqrt{n}\frac{\bar{X}-\mu_1}{\sigma_1} < \sqrt{n}\frac{y-\mu_1}{\sigma_1}\right) =\\
&\Phi(\sqrt{n}\frac{y-\mu_1}{\sigma_1}) - \Phi(\sqrt{n}\frac{x-\mu_1}{\sigma_1})
\end{aligned}
$$

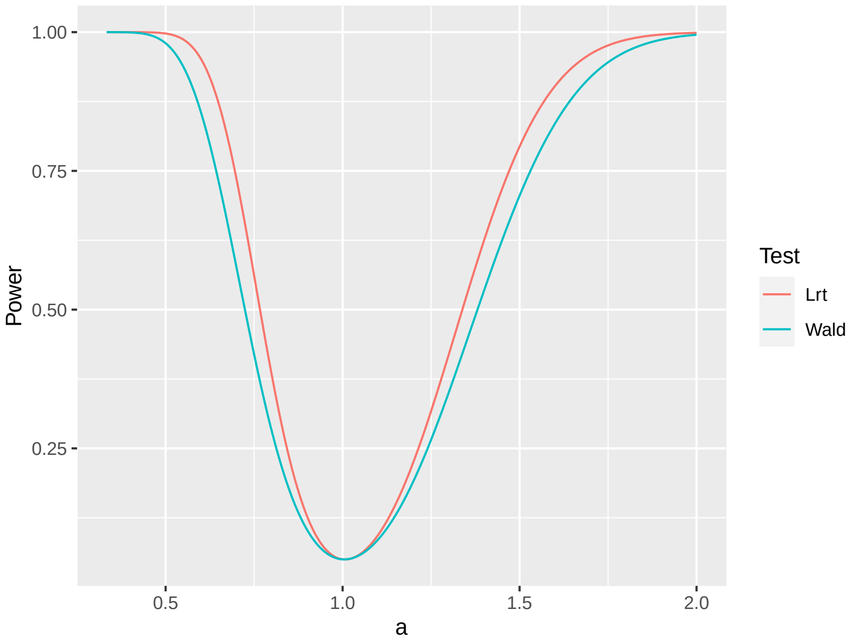
- lrt test

$$
\begin{aligned}
&P\left(qgamma(\alpha/2, n, 1) < a_0 T < qgamma(1-\alpha/2, n, 1)|a_1\right) =\\
&P\left(qgamma(\alpha/2, n, 1)\frac{a_1}{a_0} < a_1 T < qgamma(1-\alpha/2, n, 1)\frac{a_1}{a_0}|a_1\right) =\\
&pgamma(qgamma(\alpha/2, n, 1)\frac{a_1}{a_0}, n, 1) - pgamma(qgamma(\alpha/2, n, 1)\frac{a_1}{a_0}, n, 1)
\end{aligned}
$$

so

```
a0 <- 1; n <- 50; alpha <- 0.05
a <- seq(a0/3, 2*a0, length=250)
mu <- a0/(a0+1)
sigma <- sqrt(a0/(a0+1)^2/(a0+2))
mu1 <- a/(a+1)
sigma1 <- sqrt(a/(a+1)^2/(a+2))
xy <- mu+sigma/sqrt(n)*qnorm(c(alpha/2, 1-alpha/2))
Pow1 <- 1-(pnorm(sqrt(n)/sigma1*(xy[2]-mu1)) -
           pnorm(sqrt(n)/sigma1*(xy[1]-mu1)))
Pow2 <- 1-(pgamma(a/a0*qgamma(1-alpha/2, n, 1), n, 1) -
          pgamma(a/a0*qgamma(alpha/2, n, 1), n, 1))
df <- data.frame(a=c(a, a),
          Power=c(Pow1, Pow2),
```

```r
          Test=rep(c("Wald", "Lrt"), each=250))
ggplot(data=df, aes(a, Power, color=Test)) +
  geom_line()
```



Of course the Wald type test only works for large samples. A difficult question is how large n needs to be. Simulation can help to decide that.

We also have two tests for the case n=1. How do they relate to each other? For the test based on the likelihood ratio statistic we reject H_0 if

$T < qgamma(\alpha/2, n, a_0)$ or $T > qgamma(1-\alpha/2, n, a_0)$

but now $T = -\log X$ so

$-\log x < qgamma(\alpha/2, n, a_0)$ or $-\log x > qgamma(1-\alpha/2, n, a_0)$

iff

$x < \exp[-qgamma(1-\alpha/2, n, a_0)]$ or $x > \exp[-qgamma(\alpha/2, n, a_0)]$

for the direct test we reject H_0 if

$x < (\alpha/2)^{1/a0}$ or $x > (1-\alpha/2)^{1/a0}$

Let’s find some of these critical values

```r
a0 <- seq(0.1, 10, length=10)
cv1 <- (alpha/2)^(1/a0)
cv2 <- (1-alpha/2)^(1/a0)
cv1a <- exp(-qgamma(1-alpha/2, 1, a0))
cv2a <- exp(-qgamma(alpha/2, 1, a0))
kable.nice(round(cbind(a0,cv1,cv1a,cv2,cv2a), 4),
           do.row.names = FALSE)
```

a0	cv1	cv1a	cv2	cv2a
0.1	0.0000	0.0000	0.7763	0.7763
1.2	0.0462	0.0462	0.9791	0.9791
2.3	0.2011	0.2011	0.9891	0.9891
3.4	0.3379	0.3379	0.9926	0.9926
4.5	0.4405	0.4405	0.9944	0.9944
5.6	0.5175	0.5175	0.9955	0.9955
6.7	0.5766	0.5766	0.9962	0.9962
7.8	0.6232	0.6232	0.9968	0.9968
8.9	0.6607	0.6607	0.9972	0.9972
10.0	0.6915	0.6915	0.9975	0.9975

and we see they are the same! Here is why:

$qgamma(y, 1, a)$ is the solution to the equation

$$\begin{aligned}
&y = \int_0^x \frac{a^1}{(1-1)!} t^{1-1} e^{-at} dt = \\
&\int_0^x a e^{-ay} dy = 1 - e^{-ax} \\
&e^{-x} = (1-y)^{1/a}
\end{aligned}$$

How about n=2? This is much trickier because one test is based on x+y, the other one on log(x)+log(y).

Here is a little simulation:

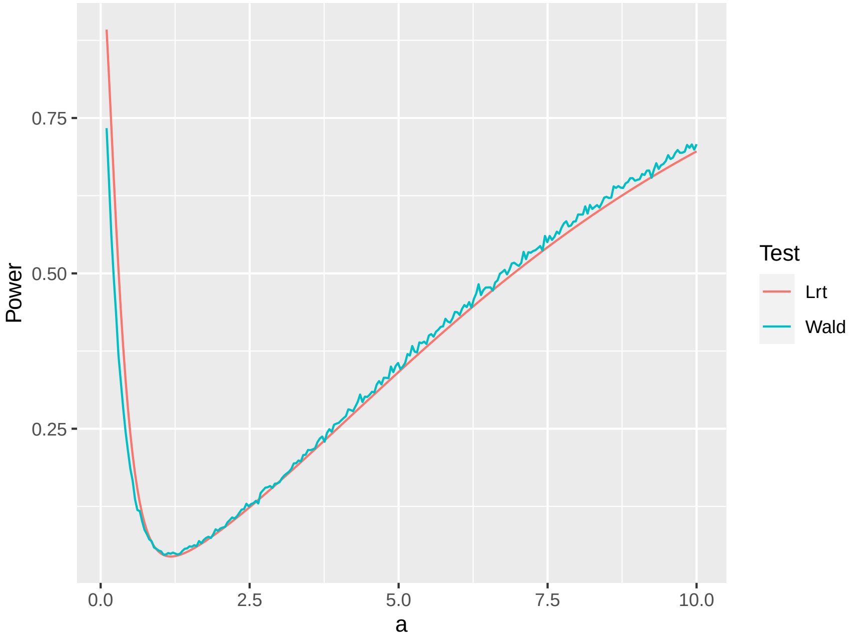
```r
B <- 1e4
r <- matrix(0, B, 2)
for(i in 1:B) {
  xy <- rbeta(2, 1/2, 1)
  if(sum(xy)<0.03125 | sum(xy)>1.593) r[i,1] <- 1
  if(-sum(log(xy))<0.484 | -sum(log(xy))>11.14) r[i,2] <- 1
  if(r[i, 1]!=r[i, 2]) break
}
xy
```

```
## [1] 0.026905593 0.002043601
```

which shows that sometime one test rejects H_0 while the other does not.

So then which is better? We do have a formula for the power of the likelihood ratio test, the other one needs to be done via simulation. Here is the result for $a_0 = 0.5$:

```
a0 <- 1; B <- 1e4
xy <- matrix(rbeta(2*B, a0, 1), ncol=2)
z <- apply(xy, 1, sum)
cv <- quantile(z, c(0.025,0.975))
a <- seq(0.1, 10,length=250)
Pow1 <- rep(0, 250)
for(i in 1:250) {
  xy <- matrix(rbeta(2*B, a[i],1), ncol=2)
  z <- apply(xy, 1, sum)
  z1 <- z[z<cv[1]]
  z2 <- z[z>cv[2]]
  Pow1[i] <- length(c(z1, z2))/B
}
Pow2 <- 1-(pgamma(a/a0*qgamma(0.975,2,1),2,1)-
             pgamma(a/a0*qgamma(0.025,2,1),2,1))
df <- data.frame(a=c(a, a),
          Power=c(Pow1, Pow2),
          Test=rep(c("Wald", "Lrt"), each=250))
ggplot(data=df, aes(a, Power, color=Test)) +
  geom_line()
```



and we see that the two tests have very similar power.

6 Interval Estimation

6.1 Interval Estimation - Basics

In real life point estimates are rarely enough, usually we also need some estimate of the **error** in our estimate.

6.1.0.1 Example (6.1.1) A census of all the students at the Colegio 10 years ago showed a mean GPA of 2.75. In our survey of 150 students we find today a mean GPA of 2.53. How much (if at all) has the GPA changed?

The problem of course is that the sample mean GPA depends on the sample, if we repeated our survey tomorrow with a different sample of 150 students, their mean GPA will not again be 2.53. But how far away from 2.53 might it be? Could it actually be higher than 2.75?

One way to answer such questions is to find an **interval estimate** rather than a point estimate.

6.1.1 Frequentist Solution: Confidence Intervals

6.1.1.1 Definition (6.1.2) Say we have $X_1, ..., X_n$ iid $f(x|\theta)$. Then $(L(\boldsymbol{X}), U(\boldsymbol{X}))$ is a $100(1-\alpha)\%$ confidence interval for θ iff

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

for all θ.

Note: in a confidence interval it is the endpoints that are random variables!

6.1.1.2 Example (6.1.3) say $X_1, .., X_n \sim N(\mu, \sigma)$, then a $100(1-\alpha)\%$ confidence interval for the population mean is given by

$$\bar{x} \pm t_{n-1,\alpha/2}\frac{s}{\sqrt{n}}$$

Here $t_{n-1,\alpha}$ is the $1-\alpha$ critical value of a t distribution with n degrees of freedom.

Note that this interval is given in the form **point estimate ± error**, which is quite often true in Statistics, although not always.

We know that $\sqrt{n}\frac{\bar{X}-\mu}{s} \sim t(n-1)$. Denote the cdf of a t(n) by F, then

$$
\begin{aligned}
&P\left(\bar{X}-t_{n-1,\alpha/2}\frac{s}{\sqrt{n}}<\mu<\bar{X}+t_{n-1,\alpha/2}\frac{s}{\sqrt{n}}\right)=\\
&P\left(\mu-t_{n-1,\alpha/2}\frac{s}{\sqrt{n}}<\bar{X}<\mu+t_{n-1,\alpha/2}\frac{s}{\sqrt{n}}\right)=\\
&P\left(-t_{n-1,\alpha/2}<\sqrt{n}\frac{\bar{X}-\mu}{s}<t_{n-1,\alpha/2}\right)=\\
&2P\left(\sqrt{n}\frac{\bar{X}-\mu}{s}<t_{n-1,\alpha/2}\right)-1=\\
&2F(t_{n-1,\alpha/2})-1=\\
&2F(F^{-1}(1-\alpha/2))-1=\\
&2(1-\alpha/2)-1=1-\alpha
\end{aligned}
$$

As a numerical example consider the case of the new text book:

```r
n <- 150; xbar <- 2.53; s <- 0.65; alpha <- 0.05
round(xbar+c(-1,1)*qt(1-alpha/2, n-1)*s/sqrt(n), 3)
```

```
## [1] 2.425 2.635
```

What does that mean: a 90% confidence interval for the mean is (2.425, 2.635)? The interpretation is this: suppose that over the next year statisticians (and other people using statistics) all over the world compute 100,000 90% confidence intervals, many for the mean, others maybe for medians or standard deviations or ..., than about 90% or about 90,000 of those intervals will actually contain the parameter that is supposed to be estimated, the other 10,000 or so will not.

It is tempting to interpret the confidence interval as follows: having found our 90% confidence interval of (2.425, 2.635), we are now 90% sure that the true mean GPA (the one for all the students at the Colegio) is somewhere between 2.425 and 2.635.

Strictly speaking this interpretation is not correct because once we have computed the interval (2.425, 2.635) the true mean GPA is either in it or not. There is now no longer a frequentist interpretation of probability.

6.1.2 Coverage

The main property of confidence intervals is their **coverage**, that is just the equation above.

6.1.2.1 Example (6.1.4) say $X_1, .., X_n \sim Ber(p)$, then by the CLT

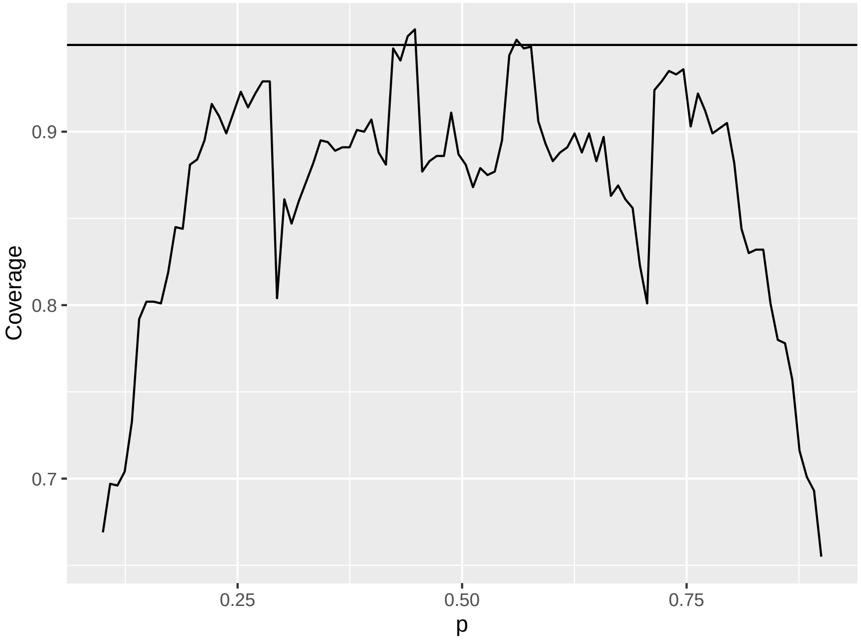
$$\bar{X} \sim N\left(p, \sqrt{p(1-p)/n}\right)$$

so we have a candidate for a $100(1-\alpha)\%$ CI:

$$\bar{x} \pm z_{1-\alpha/2}\sqrt{\bar{x}(1-\bar{x})/n}$$

But this is based on the CLT, so there is a question how large n needs to be for this to work. Let's see:

```
n <- 10; alpha <- 0.05; B <- 1000
p <- seq(0.1, 0.9, length = 100)
cov <- 0*p
crit <- qnorm(1 - alpha/2)
for (i in 1:100) {
  xbar <- rbinom(B, size = n, prob = p[i])/n
  L <- xbar-crit*sqrt(xbar*(1-xbar)/n)
  U <- xbar+crit*sqrt(xbar*(1-xbar)/n)
  cov[i] <- sum(L < p[i] & U > p[i])/B
}
ggplot(data.frame(p=p, Coverage=cov), aes(p, Coverage)) +
  geom_line() +
  geom_hline(yintercept=0.95)
```



As we can see, it does not work very well at all.

Notice the ragged appearance of the coverage graph. This is typical for discrete rv's like the Bernoulli.

Here we find the coverage using simulation, which is fine and even uses the spirit of confidence intervals.

Because this is a discrete random variable we can however also calculate the coverage exactly, using R : say we want to find the coverage for the case n=10, p=0.43. In this case x is one of 0, 1, 2.., 10, so $\bar{x}$ can only have values 0/10, 1/10, . . . , 10/10, and so we can find all the possible intervals:

```r
n <- 10; xbar <- 0:n/n
LU <- matrix(0, n+1, 2)
LU[, 1] <- xbar-crit*sqrt(xbar*(1-xbar)/n)
LU[, 2] <- xbar+crit*sqrt(xbar*(1-xbar)/n)
df <- round(data.frame(x=0:n, xbar=xbar, L=LU[, 1], U=LU[, 2]), 3)
kable.nice(df, do.row.names = FALSE)
```

x	xbar	L	U
0	0.0	0.000	0.000
1	0.1	-0.086	0.286
2	0.2	-0.048	0.448
3	0.3	0.016	0.584
4	0.4	0.096	0.704
5	0.5	0.190	0.810
6	0.6	0.296	0.904
7	0.7	0.416	0.984
8	0.8	0.552	1.048
9	0.9	0.714	1.086
10	1.0	1.000	1.000

Notice that for x=2 to x=7 we have intervals that contain p=0.43, so the true coverage is

$$\begin{aligned}
&\text{Coverage} = P(L(X) < p < U(X)|p = 0.43) = \\
&P(2 \le X \le 7|p = 0.43) = \\
&\sum_{i=2}^{7} \binom{10}{i} 0.43^i (1 - 0.43)^{10-i}
\end{aligned}$$

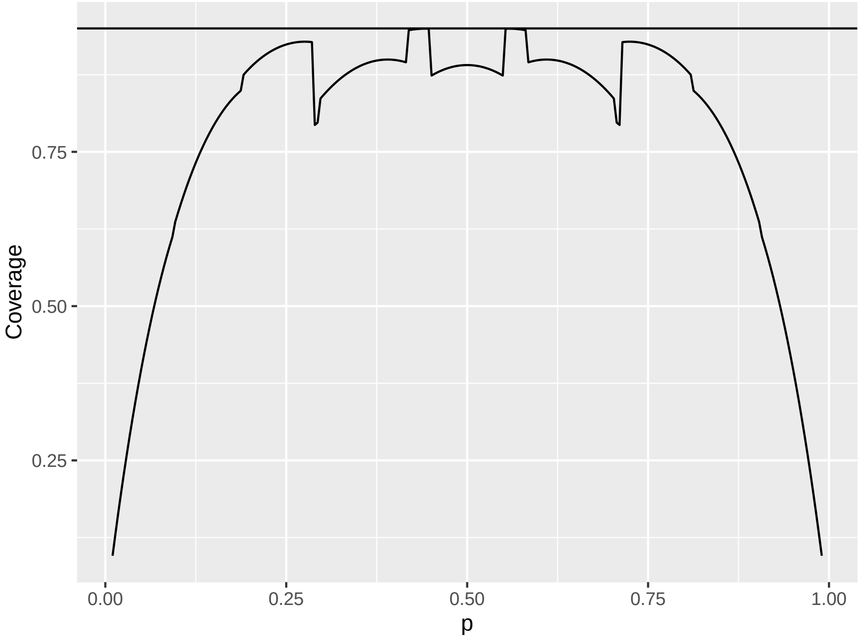
```r
round(sum(dbinom(2:7, n, 0.43)), 4)
```

```
## [1] 0.9489
```

Let's redo the coverage graph above, now using exact calculations:

```r
n <- 10; alpha <- 0.05
xbar <- 0:n/n
p <- seq(0.01, 0.99, length = 250)
```

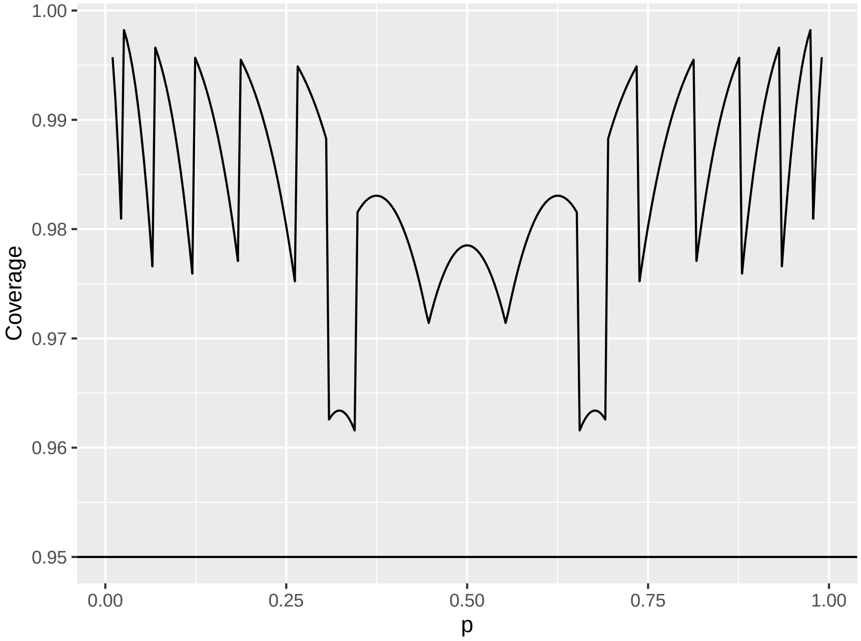
```
cov <- 0*p
crit <- qnorm(1 - alpha/2)
LU <- matrix(0, n+1, 2)
LU[, 1] <- xbar-crit*sqrt(xbar*(1-xbar)/n)
LU[, 2] <- xbar+crit*sqrt(xbar*(1-xbar)/n)
for (i in 1:250) {
  m <- (0:n)[LU[, 1]<p[i] & p[i]<LU[, 2]]
  cov[i] <- sum(dbinom(m, n, p[i]))
}
ggplot(data.frame(p=p, Coverage=cov), aes(p, Coverage)) +
  geom_line()+
  geom_hline(yintercept=0.95)
```



A much better method was invented by Clopper and Pearson in 1934. It is implemented in R in the *binom.test* command. Let's check its coverage:

```
n <- 10; alpha <- 0.05
p <- seq(0.01, 0.99, length = 250)
cov <- 0*p
LU <- matrix(0, n+1, 2)
for(x in 0:n) LU[x+1, ] <- binom.test(x, n)$conf.int
for (i in 1:250) {
  m <- (0:n)[LU[, 1]<p[i] & p[i]<LU[, 2]]
  cov[i] <- sum(dbinom(m, n, p[i]))
}
```

```
ggplot(data.frame(p=p, Coverage=cov), aes(p, Coverage)) +
  geom_line()+
  geom_hline(yintercept=0.95)
```



Now this method has some *over-coverage*. It is said to be *conservative*, that is its intervals are a bit larger than they need to be. This is not nice but often acceptable, whereas *under-coverage* is not.

6.1.2.2 Example (6.1.5) say $X \sim N(\mu, 1)$ and we are told that the routine *norm1* calculates 95% confidence intervals for μ. Let's check its coverage.

Again we can use simulation:

```
mu <- 0.76; B <- 1e4
x <- rnorm(B, mu)
LU <- norm1(x)
head(round(cbind(x, LU), 3))
```

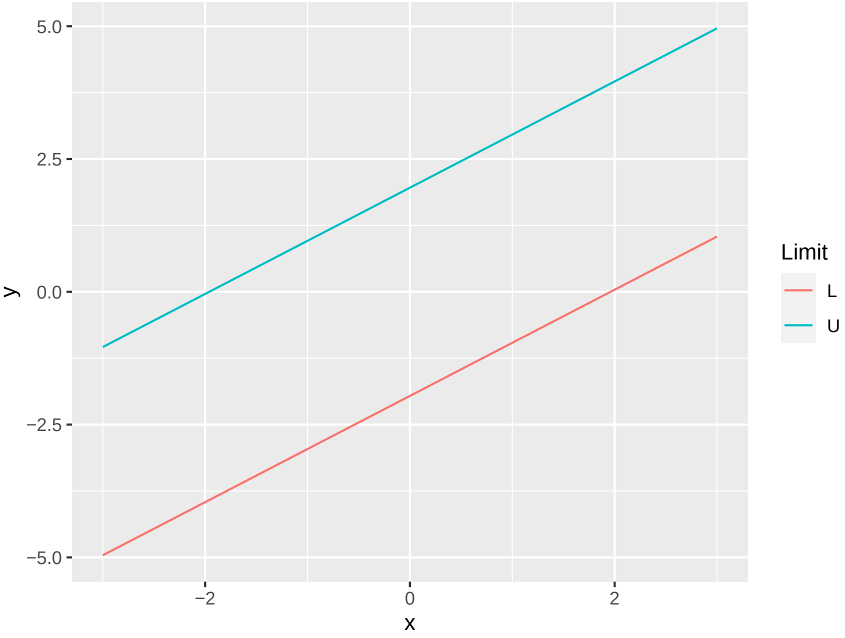
```
##            x
## [1,]  1.607 -0.353 3.567
## [2,]  0.350 -1.610 2.310
## [3,] -0.102 -2.062 1.858
## [4,]  0.142 -1.818 2.102
## [5,]  2.112  0.152 4.072
## [6,]  0.283 -1.677 2.243
```

```
sum(LU[, 1]<mu & mu<LU[, 2])/B
```

```
## [1] 0.9468
```

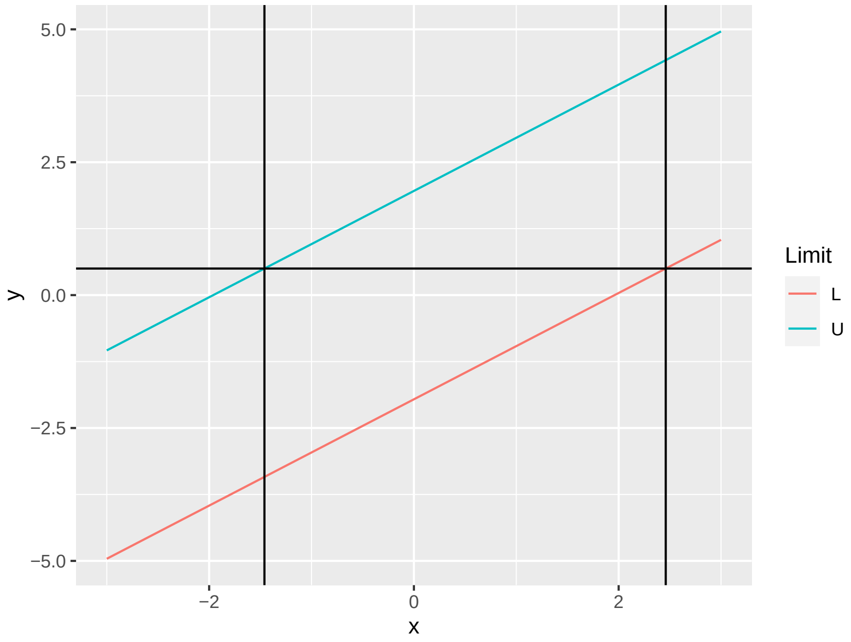
Can we do this here as well without simulation?

```
x <- seq(-3, 3, 0.01)
LU <- norm1(x)
df <- data.frame(x=c(x, x),
    y=c(LU[, 1], LU[, 2]),
    Limit=rep(c("L", "U"), each=length(x)))
ggplot(data=df, aes(x, y, color=Limit)) +
  geom_line()
```



shows that the limits are strictly increasing. This need not be the case in general! So say we want to check coverage for μ. Then there exists x_1 such that $L(x_1) = \mu$ and x_2 such that $L(x_2) = \mu$. For example, if $\mu = 0.5$ we find

```
x1 <- x[abs(LU[, 1]-0.5)==min(abs(LU[, 1]-0.5))]
x2 <- x[abs(LU[, 2]-0.5)==min(abs(LU[, 2]-0.5))]
ggplot(data=df, aes(x, y, color=Limit)) +
  geom_line() +
  geom_hline(yintercept = 0.5) +
  geom_vline(xintercept = c(x1, x2))
```



and clearly

$$\text{cov}(\mu) = \int_{x_1}^{x_2} f(x;\mu)dx$$

```
round(diff(pnorm(c(x2, x1), 0.5)), 4)
```

```
## [1] 0.95
```

6.1.3 Bayesian Solution: Credible Intervals

6.1.3.1 Definition (6.1.6) Say we have $X_1,..,X_n$ iid $f(x|\theta)$. Then $(L(\boldsymbol{x}), U(\boldsymbol{x}))$ is a $100(1-\alpha)\%$ credible interval for θ iff

$$P\left(L(\boldsymbol{x}) < \theta < U(\boldsymbol{x})|\boldsymbol{x}\right) = 1-\alpha$$

Notice now the data appears in the conditional part, so this is a probability based on the posterior distribution of $\theta|\boldsymbol{x}$.

Also note that now the upper and lower limits are not random variables.

6.1.3.2 Example (6.1.7) say $X_1,..,X_n \sim N(\mu,\sigma)$

To keep things simple we will assume that σ is known, so we just need a prior on μ. Let's say we use $\mu \sim N(\mu_0, \tau)$, then we have previously seen that $\mu|\boldsymbol{x} \sim N(a,b)$ where

$$
\begin{aligned}
c^2 &= \frac{n}{\sigma^2} + \frac{1}{\tau^2} \\
a &= \frac{(\sum x_i)/\sigma^2 + \mu_0^2/\tau^2}{c^2} \\
b &= 1/c
\end{aligned}
$$

is again a normal.

How can we get a credible interval from this? The definition above does not determine a unique interval, essentially we have one equation for two unknowns, so we need an additional condition. Here are some standard solutions:

- equal tail probability intervals: choose L and U such that

$$P\left(\theta < L(\boldsymbol{x})|\boldsymbol{x}\right) = \alpha/2 P\left(\theta > U(\boldsymbol{x})|\boldsymbol{x}\right) = \alpha/2$$

- *highest posterior density intervals* (HPD). Here we choose the limits in such a way that the density has the same value. That is, we have the solution to the system of equations

$$
\begin{aligned}
&P\left(L(\boldsymbol{x}) < \theta < U(\boldsymbol{x})|\boldsymbol{x}\right) = 1 - \alpha \\
&f(L(\boldsymbol{x})|\boldsymbol{x}) = f(U(\boldsymbol{x})|\boldsymbol{x})
\end{aligned}
$$

- quantiles from simulated data

Let's consider as a numerical example the text book data:

First we need to choose σ, μ_0 and τ. Let's use $\sigma = 0.65, \mu_0 = 3.0, \tau = 1.0$, then

- equal tail probability intervals

```r
n <- 150; mu0 <- 3.0; sigma <- 0.65; tau <- 1.0
c2 <- n/sigma^2+1/tau^2
a <- (150*2.53/sigma^2+mu0^2/tau^2)/c2
b <- 1/sqrt(c2)
round(qnorm(c(0.025, 0.975), a, b), 2)
```

```
## [1] 2.44 2.65
```

- highest posterior density interval. In our case this yields the same interval as above because the normal density is symmetric around the mean.
- quantiles from simulated data.

```r
round(quantile(rnorm(1e4, a, b), c(0.025, 0.975) ), 2)
```

```
##  2.5% 97.5%
##  2.44  2.65
```

The main property of credible intervals is just the equation that defines them.

6.1.3.3 Example (6.1.8) say $X_1, .., X_n \sim Ber(p)$, and let's use as a prior on p the U[0,1]. Then we saw before that the posterior distribution is Beta($\sum x_i + 1, n + 1 - \sum x_i$).

- with the equal tail probabilities method we find

$$L(x) = qbeta(\alpha/2, 1 + \sum x, n + 1 - \sum x)$$

$$U(x) = qbeta(1 - \alpha/2, \sum x, n + 1 - \sum x)$$

- HPD

Let $y = \sum x_i$, then we need to solve the system of equations

$$\int_a^b \frac{(n+1)!}{y!(n-y)!} t^y (1-t)^{n-y} dt = 1 - \alpha$$
$$a^y (1-a)^{n-y} = b^y (1-b)^{n-y}$$

and this has to be done numerically.

6.1.4 Multidimensional Estimates

6.1.4.1 Example (6.1.9) say $X_1, .., X_n \sim N(\mu, \sigma)$ and we are interested in estimating μ and σ simultaneously. So we want to find a region $A(\boldsymbol{x}) \in \mathbb{R} \times \mathbb{R}$ such that

$$P(A(\mathbf{X})) = 1 - \alpha$$

We already know that $\bar{x}$ and s are good point estimators of μ and σ. Moreover it can be shown that $\bar{x}$ and s are independent, so

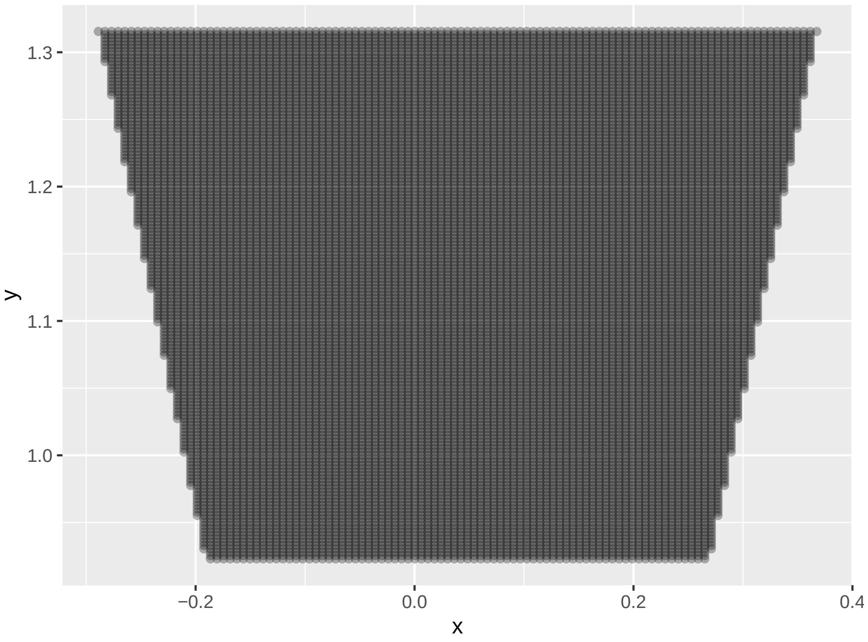
$$
\begin{aligned}
&P\left(\bar{X} - t_{n-1,w/2}\frac{s}{\sqrt{n}} < \mu < \bar{X} + t_{n-1,w/2}\frac{\sigma}{\sqrt{n}}\ ,\right.\\
&\left.\frac{(n-1)s^2}{q\chi^2(1-w/2)} < \sigma^2 < \frac{(n-1)s^2}{q\chi^2(w/2)}\right) =\\
&P\left(\bar{X} - t_{n-1,w/2}\frac{s}{\sqrt{n}} < \mu < \bar{X} + t_{n-1,w/2}\frac{\sigma}{\sqrt{n}}\right) \cdot\\
&P\left(\frac{(n-1)s^2}{q\chi^2(1-w/2)} < \sigma^2 <= \frac{(n-1)s^2}{q\chi^2(w/2)}\right) =\\
&(1-w/2)^2
\end{aligned}
$$

so if we want a $(1-\alpha)100\%$ confidence region we need to use

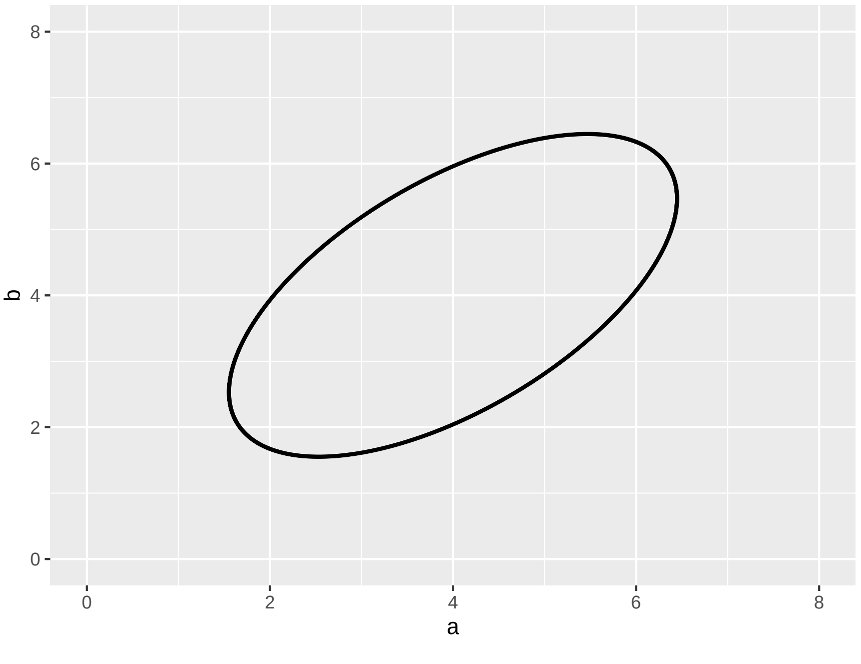
$$w = 1 - \sqrt{1-\alpha}$$

What does this region look like?

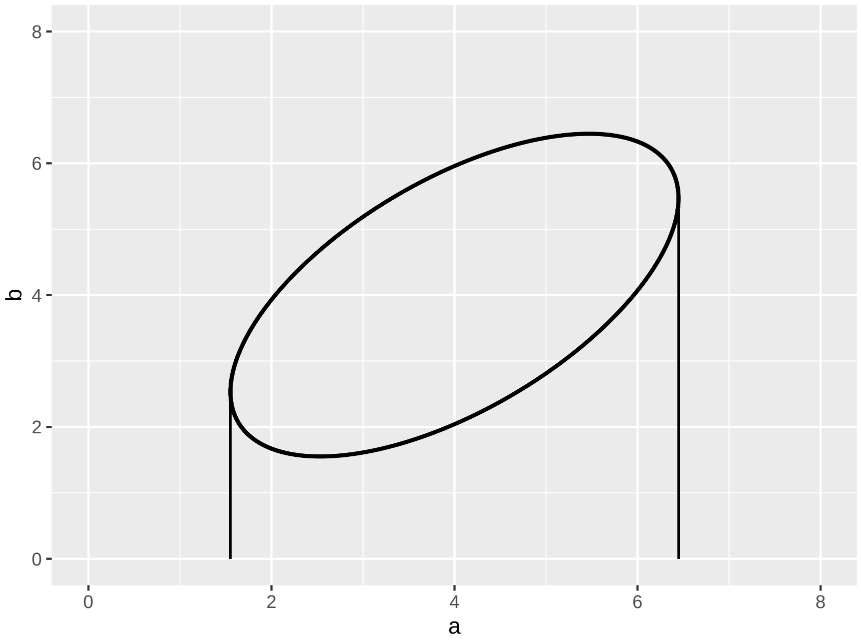
```
n <- 100; mu0 <- 0; sigma <- 1
x <- rnorm(100, mu0, sigma)
xbar <- mean(x)
s <- sd(x)
w <- 1 - sqrt(1 - 0.05/2)
a <- qnorm(1 - w/2)/sqrt(n)
cv1 <- xbar + c(-1, 1)*qnorm(1-w/2)*sigma/sqrt(n)
cv2  <-  c((n-1)*s^2/qchisq(1-w/2, n-1),
           (n-1)*s^2/qchisq(w/2, n-1))
mu <- seq(cv1[1]-0.5, cv1[2]+0.5, length = 250)
sigma <- seq(0.9*cv2[1]^0.5, 1.1*cv2[2]^0.5, length = 250)
xy <- expand.grid(mu, sigma)
I <- rep(FALSE, dim(xy)[1])
for(i in 1:dim(xy)[1]) {
  if(xbar-a*xy[i, 2] < xy[i, 1] &
     xy[i, 1] < xbar+a*xy[i, 2] &
     cv2[1] < xy[i, 2]^2 &
     xy[i, 2]^2 < cv2[2]) I[i] <- TRUE

}
df <- data.frame(x=xy[I,1], y=xy[I, 2])
ggplot(data=df, aes(x, y)) +
  geom_point(alpha=0.3)
```



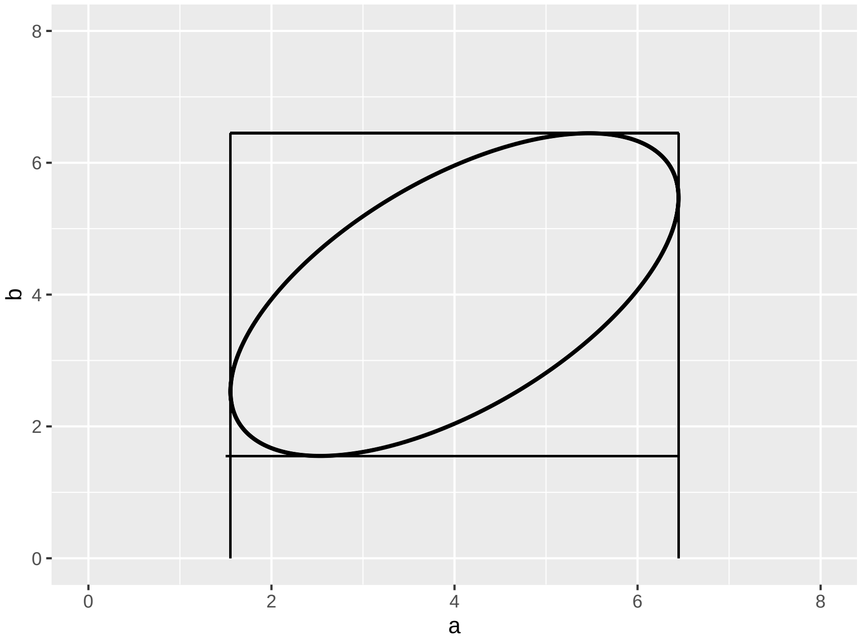
Say we have some density f(.;a,b) and a 95% confidence region for (a, b) that looks like this:



But actually, we want a 95% confidence interval for a. So how about



about (1.8, 6.3). But this will clearly overcover badly because it would be the same one-dimensional interval as we would get if the confidence regions were a square:



but of course the region of the square is much larger. There is in fact no known way to get confidence intervals from confidence regions!

6.2 Methods for Finding Interval Estimates

6.2.1 Inverting a Hypothesis Test

Hypothesis testing and confidence intervals are closely related, in fact, a hypothesis test can always be turned into a confidence interval and vice versa. Say we have a hypothesis test with type I error probability α and we define the acceptance region of the test $A(\theta_0)$ as the complement of the critical region when testing $H_0 : \theta = \theta_0$. That is A consists of all those points in $\mathbf{R}^n$ that would have lead to a failure to reject the null hypothesis.

Define the set $C(\boldsymbol{x})$ in the parameter space by

$$C(\boldsymbol{x}) = \{\theta : \boldsymbol{x} \in A(\theta)\}$$

In other words, the confidence interval is the set of all parameters for which the hypothesis test would have accepted H_0.

6.2.1.1 Example (6.2.1) Let $X_1, .., X_n \sim N(\mu, \sigma)$. Say we want to find a confidence interval for μ. To do this we first need a hypothesis test for

$$H_0 : \mu = \mu_0 \text{ vs } H_1 : \mu \neq \mu_0$$

Of course we have the 1-sample t-test with the test statistic

$$T = \sqrt{n}\frac{\bar{x} - \mu_0}{s}$$

which rejects the null if $|T| > t_{\alpha/2,n-1}$. So it accepts H_0 if $|T| \le t_{\alpha/2,n-1}$ and we have the acceptance region

$$A(\mu_0) = \left\{\boldsymbol{x} | |T| \le t_{\alpha/2,n-1}\right\}$$

Now

$$\begin{aligned}
&1 - \alpha = P\left(|T| < t_{n-1,\alpha/2}\right) = \\
&P\left(|\sqrt{n}\frac{\bar{X} - \mu}{s}| < t_{n-1,\alpha/2}\right) = \\
&P\left(-t_{n-1,\alpha/2} < \sqrt{n}\frac{\bar{X} - \mu}{s} < t_{n-1,\alpha/2}\right) = \\
&P\left(\mu - t_{n-1,\alpha/2}\frac{s}{\sqrt{n}} < \bar{X} < \mu + t_{n-1,\alpha/2}\frac{s}{\sqrt{n}}\right) = \\
&P\left(\bar{X} - t_{n-1,\alpha/2}\frac{s}{\sqrt{n}} < \mu < \bar{X} + t_{n-1,\alpha/2}\frac{s}{\sqrt{n}}\right)
\end{aligned}$$

and so a $100(1 - \alpha)\%$ confidence interval for μ is given by $\bar{x} \pm t_{n-1,\alpha/2}\frac{s}{\sqrt{n}}$

6.2.1.2 Example (6.2.2)

Let's look at another example to illustrate the important point here. Let's say we have $X_1, .., X_n \sim Ber(p)$ and we want to find a $100(1-\alpha)\%$ CI for p. To do so we first need to find a test for

$$H_0: p = p_0 \text{ vs } H_a: p \neq p_0$$

Let $x = \sum x_i$ be the number of successes. We have previously found the likelihood ratio test to be: reject H_0 if $|x - np_0| > c$.

What is c? We have

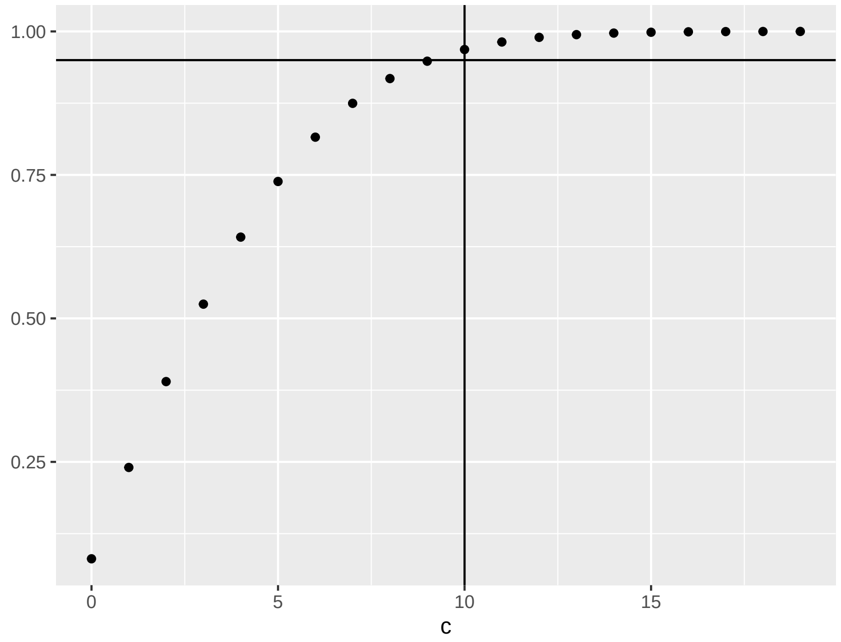
$$\begin{aligned}
&\alpha = P_{p=p_0}(\text{reject H}_0) = \\
&P_{p=p_0}(|X - np_0| > c) = \\
&1 - P_{p=p_0}(|X - np_0| \leq c) = \\
&1 - P_{p=p_0}(np_0 - c \leq X \leq np_0 + c)
\end{aligned}$$

so

$$1 - \alpha = pbinom(np_0 + c, n, p_0) - pbinom(np_0 - c - 1, n, p_0)$$

Let's use R and a simple search to find c:

```r
find.c <- function(p, n=100, alpha=0.05) {
  k <- 0
  repeat {
    k <- k + 1
    if(pbinom(n*p+k, n, p) -
       pbinom(n*p-k-1, n, p) > 1 - alpha)
            break
  }
  k
  cv <- k
  x <- 0:n
  y <- pbinom(n*p+x, n, p) - pbinom(n*p-x-1, n, p)
  reject <- ifelse(x<n*p-cv | x>n*p+cv, TRUE, FALSE)
  data.frame(x, y, p=rep(p, n+1), reject)
}
df <- find.c(0.4)
ggplot(data=df[1:20, ], aes(x, y)) +
  geom_point() +
  geom_vline(xintercept = 10) +
  geom_hline(yintercept = 0.95) +
  xlab("c") + ylab("")
```

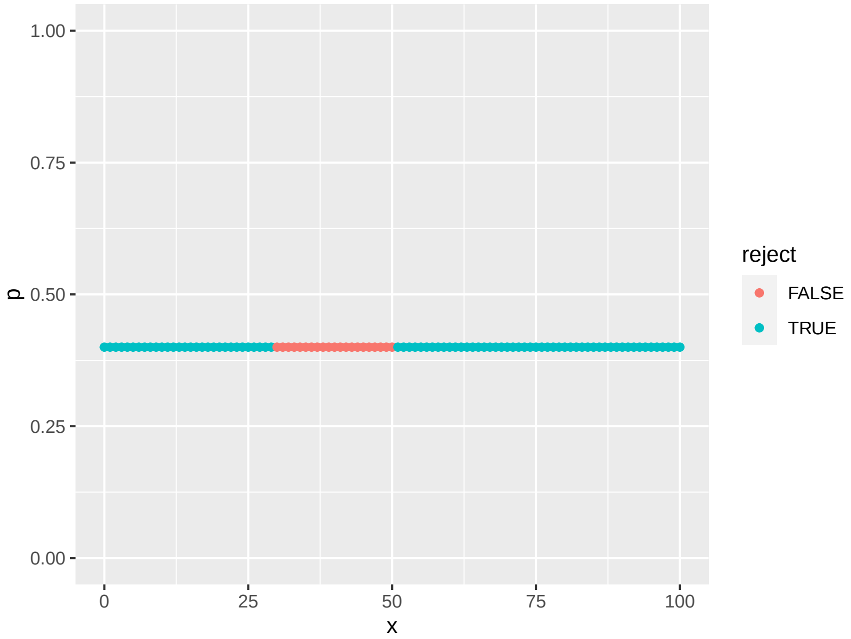


Notice that the actual probability is a bit higher than 0.95. This is again because we have a discrete random variable.

So for $p_0 = 0.4$ and n=100 we find c=10. Therefore the test is as follows: reject the null hypothesis if $|x - 40| > 10$. This is the same as $x < 30$ or $x > 50$.

We can illustrate this acceptance region as follows:

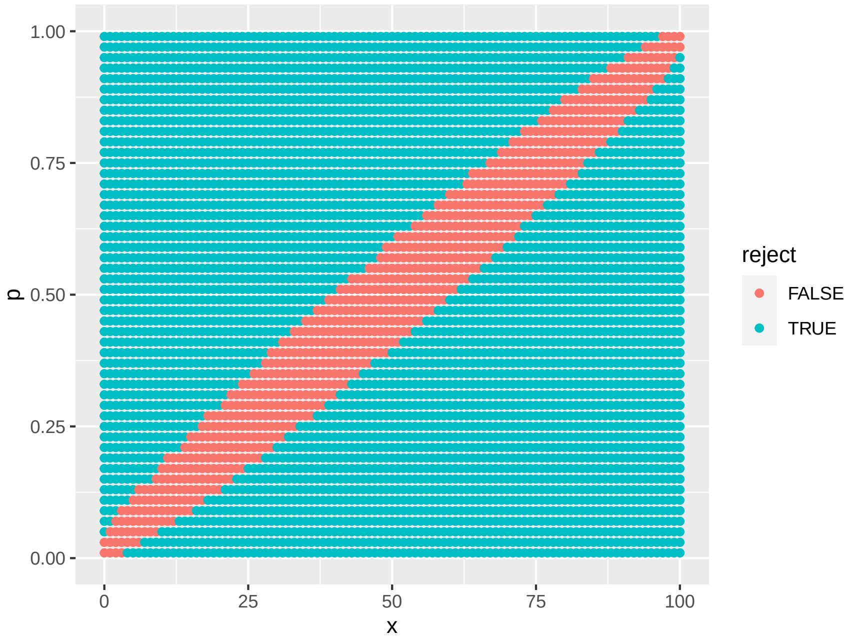
```r
ggplot(data=df, aes(x, p, color=reject)) +
  geom_point() + ylim(c(0,1))
```



This plots a dot for each possible observation x (0-n), in red if observing this value leads to accepting the null hypothesis, in blue if it means rejecting H_0.

Let's do this now for other values of p as well:

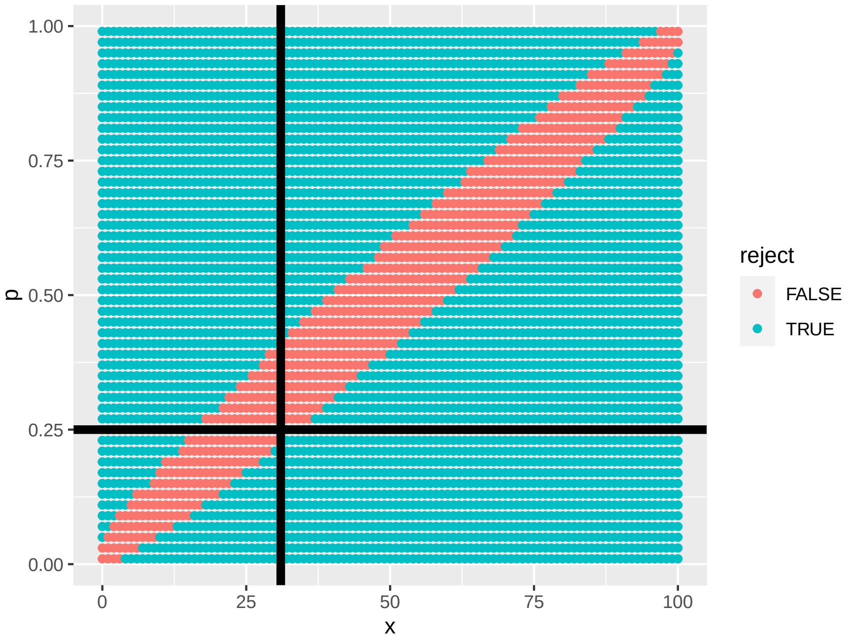
```
p <- seq(0.01, 0.99, length=50)
df <- find.c(p[1])
for(i in 2:50)
  df <- rbind(df, find.c(p[i]))
ggplot(data=df, aes(x, p, color=reject)) +
  geom_point() + ylim(c(0,1))
```



This graph tells us everything about the test (for a fixed n).

For example, say we wish to test $H_0 : p = 0.25$ and we observe $x = 31$, then

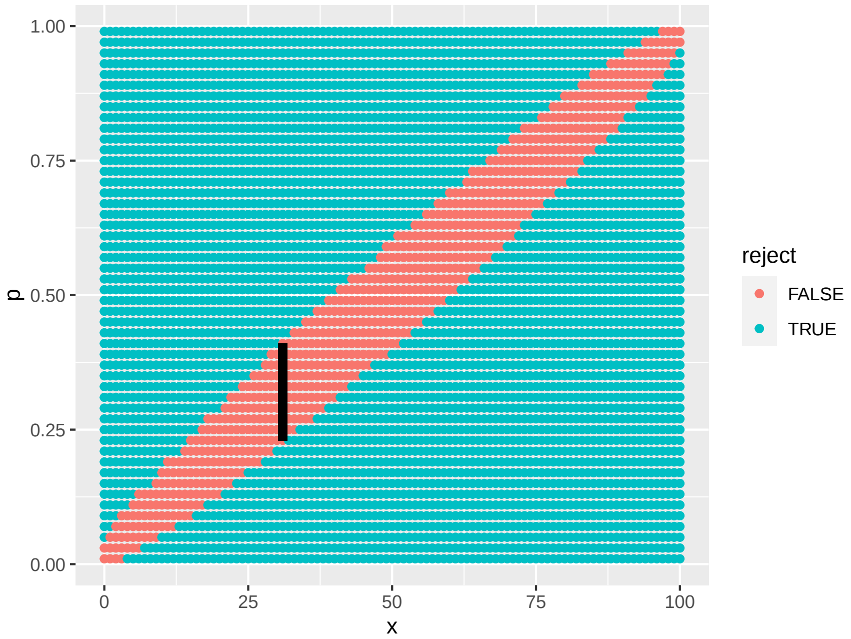
```
ggplot(data=df, aes(x, p, color=reject)) +
  geom_point() +
  geom_hline(yintercept = 0.25, size=2) +
  geom_vline(xintercept = 31, size=2)
```



shows that we should accept the null hypothesis because the intersection of the two lines is in the red (acceptance) region.

The idea of inverting the test is now very simple: for a fixed (observed) x_0, what values of p lead to accepting the null hypothesis? That is, for a given vertical line which p's are in the red region?

```
x0 <- 31
z <- rep(FALSE, dim(df)[1])
z[df$x==x0 & !df$reject] <- TRUE
p0 <- range(df$p[z])
ggplot(data=df, aes(x, p, color=reject)) +
  geom_point() +
  geom_segment(x=x0, y=p0[1],xend=x0, yend=p0[2],
               size=2, color="black")
```



```
p0
```

```
## [1] 0.23 0.41
```

here is the same graph as before, but now finding those values of p which lead to accepting the null hypothesis. This gives us the 95% confidence interval of (0.23, 0.41).

In essence we have the following:

- to do a hypothesis test fix p_0 on the y axis and scan across the graph horizontally.
- to find a confidence interval fix x_0 on the x axis and scan across the graph vertically.

6.2.1.3 Example (6.2.3)

Suppose we have $X_1, .., X_n \sim Exp(\beta)$ and we want a confidence interval for β. Again we start by considering a hypothesis test:

$$H_0 : \beta = \beta_0 \text{ vs. } H_1 : \beta \neq \beta_0$$

In (5.4.1) we found the LRT for this problem to be

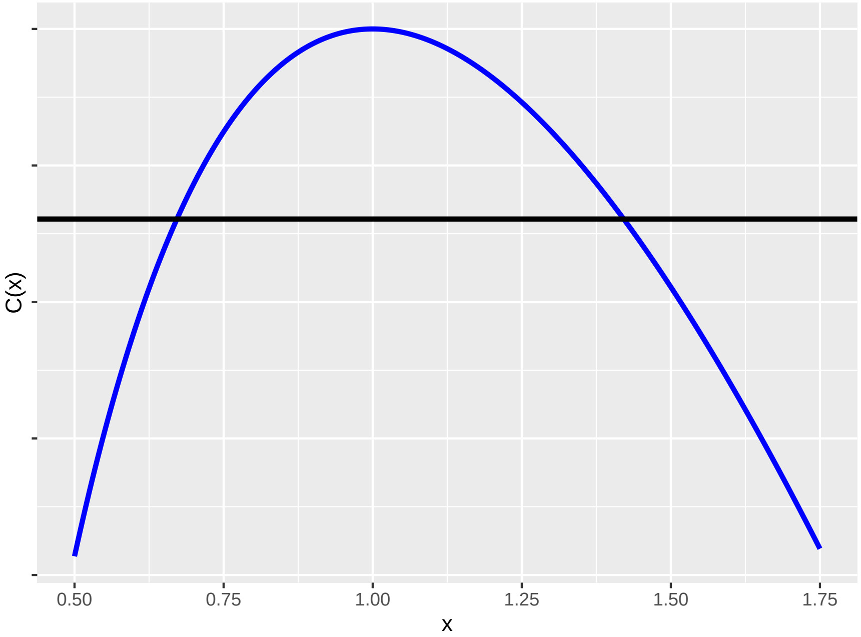
$$(-2) \log \lambda(\boldsymbol{x}) = (-2) \left[n \log(\beta_0 \bar{x}) + n - \beta_0 \bar{x} \right]$$

and so the acceptance region is given by

$$A(\beta) = \{\boldsymbol{x} : n \log(\beta_0 \bar{x}) + n - \beta_0 \bar{x} \geq c\}$$

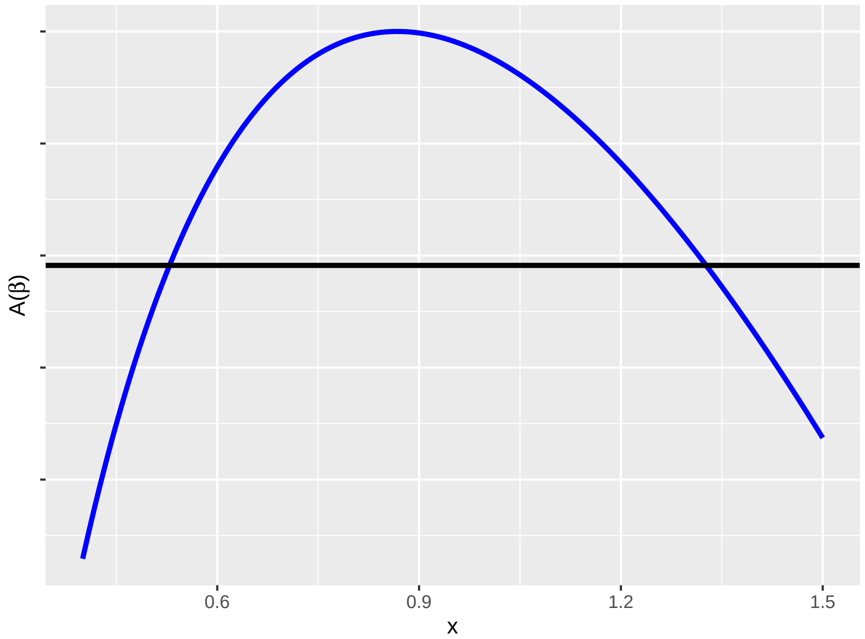
Let's draw the graph:

```
n <- 50; beta <- 1
xbar <- mean(rexp(n, 1/beta))
lrt <- function(x, b=1) n*log(x*b)-n*b*x
ggcurve(fun=lrt, A=0.5, B=1.75) +
  scale_y_continuous(labels = NULL) +
  geom_hline(yintercept = lrt(1/xbar)-3, size=1.2) +
  ylab("C(x)")
```



and so we reject the null if $\bar{x}$ is either small or large. To invert the test we can do the graph again, but now as a function of beta with $\bar{x}$ fixed:

```
lrt <- function(b, x=xbar) n*log(x*b)-n*b*x
ggcurve(fun=lrt, A=0.4, B=1.5) +
  scale_y_continuous(labels = NULL) +
  geom_hline(yintercept = lrt(xbar)-3, size=1.2) +
  ylab(expression("A("*beta*")"))
```



The expression defining the confidence interval depends on $\boldsymbol{x}$ only through $\bar{x}$. So we can write it in the form

$$C(\mathbf{x}) = \{\beta : L(\bar{x}) \leq \beta \leq U(\bar{x})\}$$

for some functions L and U which are determined so that the interval has probability $1-\alpha$.

Also note that the height of the curve at the left and the right confidence interval limit is the same, so

$$(L(\bar{x})\bar{x})^n \exp\left(-nL(\bar{x})\bar{x}\right) = (U(\bar{x})\bar{x})^n \exp\left(-nU(\bar{x})\bar{x}\right)$$

Let's denote $a = L(\bar{x})\bar{x}$ and $b = U(\bar{x})\bar{x}$, then we have the equation

$$a^n e^{-a} = b^n e^{-b}$$

Now note that

$$\sum X_i \sim \Gamma(n, 1/\beta)$$

and it is easy to show that then

$$\beta \sum X_i \sim \Gamma(n, 1)$$

and so

$$\begin{aligned}&1-\alpha = P\left(a/\bar{X} < \beta < b/\bar{X}\right) = \\ &P\left(a < \beta\bar{X} < b\right) = \\ &P\left(na < \beta\sum X < nb\right) = \\ &pgamma(nb,n,1) - pgamma(na,n,1)\end{aligned}$$

So the confidence interval becomes

$$\{\beta : a/\bar{x} \le \beta \le b/\bar{x}\}$$

where a and b satisfy

$$\begin{aligned}&a^n e^{-a} = b^n e^{-b} \\ &1-\alpha = pgamma(nb,n,1) - pgamma(na,n,1)\end{aligned}$$

This system of nonlinear equations will of course have to be solved numerically.

6.2.1.4 Example (6.2.4) Say $X_1,..,X_n \sim Pois(\lambda)$. Let $Y = \sum X_i$, than $Y \sim Pois(n\lambda)$. Say we observe $Y = y_0$. We have previously found a hypothesis test for

$$H_0 : \lambda = \lambda_0 \text{ vs } H_a : \lambda \neq \lambda_0$$

It had the acceptance region

$$A(\lambda_0) = \{\text{qpois}(\alpha/2, n\lambda_0)/n \le \bar{x} \le \text{qpois}(1-\alpha/2, n\lambda_0)/n\}$$

Inverting this test means solving the equations

$$\begin{aligned}\sum_{i=0}^{y_0} \frac{(nL)^i}{i!} e^{-nL} &= \frac{\alpha}{2} \\ \sum_{i=y_0}^{\infty} \frac{(nU)^i}{i!} e^{-nU} &= \frac{\alpha}{2}\end{aligned}$$

In general this might have to be done numerically. Here, though, we can take advantage of the equation linking the Poisson and the Gamma distributions:

If $X \sim Gamma(n, \beta)$ and $Y \sim Pois(x/\beta)$ then $P(X \le x) = P(Y \ge n)$.

Using $\beta = 2, n = y_0 + 1, x = 2n\lambda$ we have

$$
\begin{aligned}
&\frac{\alpha}{2} = P(Y \le y_0) = \\
&P(Y < y_0 + 1) = \\
&1 - P(Y \ge y_0 + 1) = \\
&1 - P(Y \ge \frac{2(y_0 + 1)}{2}) = \\
&P(X \le 2n\lambda) = \\
&P(X > 2n\lambda)
\end{aligned}
$$

and so

$$\lambda = qchisq(\alpha/2, 2(y_0 + 1)/(2n)$$

Using a similar calculation for the lower bound we find

$$qchisq(1 - \alpha/2, 2(y_0 + 1)/(2n) < \lambda = qchisq(\alpha/2, 2(y_0 + 1)/(2n)$$

where if $y_0 = 0$ we have $\chi^2(1 - \alpha/2, 0) = 0$

Let's implement this method:

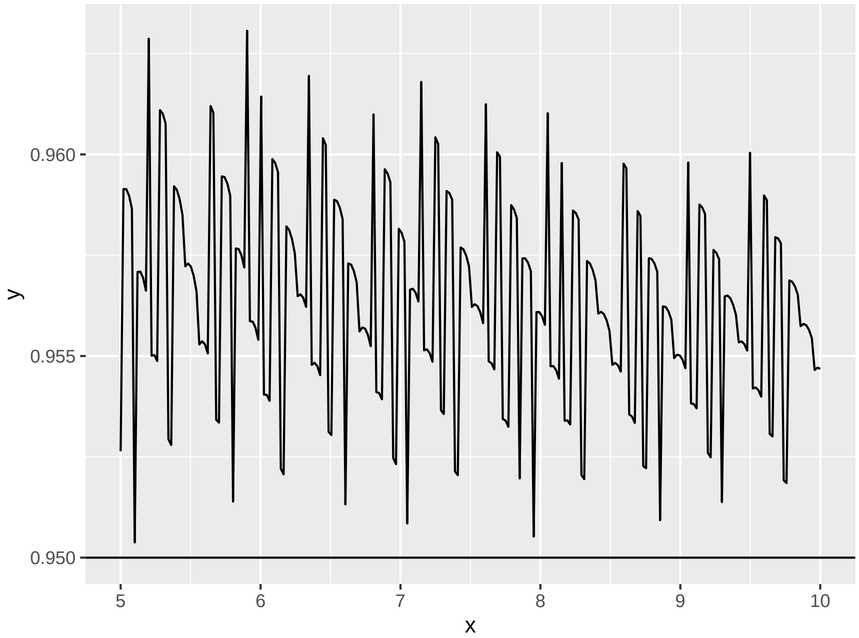
```
poisci <- function(x, alpha = 0.05) {
    c(qchisq(alpha/2, 2*sum(x)),
      qchisq(1-alpha/2, 2*(sum(x)+1))
      )/2/length(x)
}
round(poisci(rpois(30, 5.8)), 3)
```

```
## [1] 5.217 7.015
```

This method was first invented by **Garwood** in 1932.

This confidence interval has correct coverage by construction, so we don't need to worry about that as we would if our method used some approximation, say. Nevertheless, let's do a coverage study of our method. As before we can do this without simulation:

```
n <- 10; alpha = 0.05
X <- matrix(0, 30*n+1, 3)
X[, 1] <- 0:(30*n)
X[, 2] <- qchisq(alpha/2, 2*X[, 1])/2/n
X[, 3] <- qchisq(1-alpha/2, 2*(X[, 1]+1))/2/n
lambda <- seq(5, 10, length=250)
Coverage = 0*lambda
for(i in 1:250) {
  tmp <- X[X[, 2]<lambda[i] & lambda[i]<X[, 3], ]
  Coverage[i] = sum(dpois(tmp, n*lambda[i]))
}
df <- data.frame(x=lambda, y=Coverage)
ggplot(df, aes(x, y)) +
  geom_line() + geom_hline(yintercept = 0.95)
```



so again we see this ragged appearance typical for coverage graphs of discrete random variables.

6.2.2 Using the Large Sample Theory of Maximum Likelihood Estimators

From our previous discussion we know (under some regularity conditions) that the mle has an (approximate) normal distribution. This can be used to derive confidence intervals:

6.2.2.1 Example (6.2.5)

Let's say we have $X_1, .., X_n \sim N(\mu, \sigma)$

first we consider the case where σ is fixed but unknown. Let's assume we have not done anything yet with this model, and we want to estimate μ using the maximum likelihood estimator. Moreover we will use Newton's method for finding it. So we need the first two derivatives of the log likelihood function:

$$
\begin{aligned}
&f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\} \\
&l(\mu,\sigma|\boldsymbol{x}) = -\frac{1}{2}\log(2\pi) - n\log\sigma - \frac{1}{2\sigma^2}\sum(x_i-\mu)^2 \\
&\frac{dl}{d\mu} = \frac{1}{\sigma^2}\sum(x_i-\mu) \\
&\frac{d^2l}{d\mu^2} = -\frac{n}{\sigma^2}
\end{aligned}
$$

and so Newton's method yields

$$
\mu_{n+1} = \mu_n - \left(\frac{1}{\sigma^2}\sum(x_i-\mu)\right) / \left(-\frac{n}{\sigma^2}\right) = \mu_n + \sum(x_i-\mu)/n
$$

and this will converge to the sample mean.

Now what can we say about the mle? Because of the large sample theorem for mle's we know that

$$
\begin{aligned}
E\left[\frac{d^2\log f(X|\mu,\sigma)}{d\mu^2}\right] &= E[-\frac{1}{\sigma^2}] = -\frac{1}{\sigma^2} \\
v(\mu) &= -\frac{1}{nE\left[\frac{d^2\log f(X|\mu,\sigma)}{d\mu^2}\right]} = \frac{\sigma^2}{n} \\
\hat{\mu} &\sim N\left(\mu, \sqrt{v(\mu)}\right) = N(\mu, \sigma/\sqrt{n})
\end{aligned}
$$

one problem: we don't know σ, so what can we do to find the variance of the mle? But notice that:

$$
v(\mu) = \sigma^2/n = -\left(\frac{d^2 f}{d\mu^2}\right)^{-1}
$$

and we already have $\frac{d^2l}{d\mu^2}$ from when we ran Newton's method! So all we have to do is use the last value from the iterations and we immediately have an estimate of the variance!

Now we have

$$
\begin{aligned}
f_{\hat{\mu}}(x) &= \frac{\sqrt{n}}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{n}{2\sigma^2}(x-\mu)^2\right\} = L(\mu|x) \\
\log L(\mu|x) &= K - \frac{n}{2\sigma^2}(\mu-x)^2 \\
2\log L(\mu|\hat{\mu}) &= K - \frac{n}{\sigma^2}(\mu-\hat{\mu})^2
\end{aligned}
$$

so if we use the mle for x, 2 log-likelihood as a function of μ is a quadratic. Let's call this function ψ:

$$\psi(\mu) = K - \frac{n(\mu - \hat{\mu})^2}{\sigma^2}$$

Let's say we want to find a $(1-\alpha)100\%$ confidence interval. We already know one solution, from our previous discussion:

$$\left(\hat{\mu} - z_{\alpha/2} s/\sqrt{n}, \hat{\mu} + z_{\alpha/2} s/\sqrt{n}\right)$$

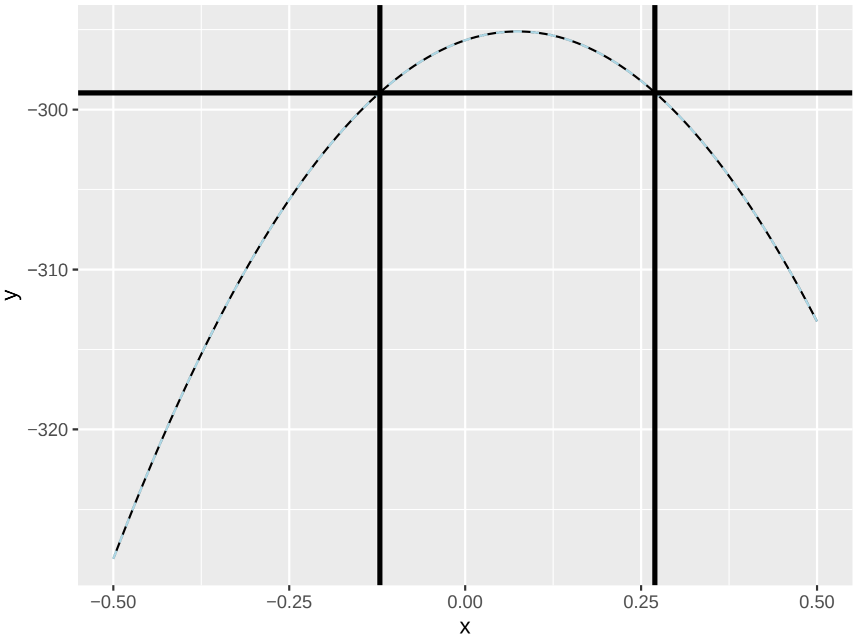
What can we say about those point(s) on the 2 log-likelihood curve? We find

$$\begin{aligned}
&\psi(\hat{\mu}) - \psi(\hat{\mu} - z_{\alpha/2} s/\sqrt{n}) = \\
&(K - \frac{n(\hat{\mu} - \hat{\mu})^2}{\sigma^2}) - (K - \frac{n(\hat{\mu} - z_{\alpha/2} s/\sqrt{n} - \hat{\mu})^2}{\sigma^2}) = z_{\alpha/2}^2
\end{aligned}$$

and so an (at this point admittedly very weird!) way to find the confidence interval is to find the points where the 2log-likelihood curve drops down by $z_{\alpha/2}^2$ from its maximum!

Let's do the graph:

```r
alpha <- 0.95; n <- 100; mu0 <- 0; sigma0 <- 1
crit <- qnorm(1-(1-alpha)/2)
x <- rnorm(n, mu0, sigma0)
muhat <- mean(x)
shat <- sqrt(sum((x-muhat)^2)/n)
K <- 2*sum(log(dnorm(x, muhat, sigma0)))
mu <- seq(-5*sigma0/sqrt(n), 5*sigma0/sqrt(n), length=500)
y <- 0*mu
for(i in 1:500)
  y[i] <- 2*sum(log(dnorm(x, mu[i], sigma0)))
Hess <- (-n)/sigma0^2
v <- 1/sqrt(-Hess)
L <- (mu[1:250])[which.min(abs(y[1:250]-K+crit^2))]
U <- (mu[250:500])[which.min(abs(y[250:500]-K+crit^2))]
df <- data.frame(x=mu, y=y, yquad=K-(mu-muhat)^2/v^2)
ggplot(df) +
  geom_line(aes(x, y), linetype="solid") +
  geom_line(aes(x, yquad),
            linetype = "dashed", color="lightblue") +
  geom_hline(yintercept = K-crit^2, size=1.2) +
  geom_vline(xintercept = c(L, U), size=1.2)
```



```
c(L, U)
```

```
## [1] -0.1212425  0.2695391
```

of course the 2 log-likelihood curve and the quadratic are the same.

Let's turn next to the case where we know μ but want to find a confidence interval for σ. Following along the same arguments as for the mean we find

$$
\begin{aligned}
&f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}\\
&l(\mu,\sigma|\boldsymbol{x}) = -\frac{1}{2}\log(2\pi) - n\log\sigma - \frac{1}{2\sigma^2}\sum(x_i-\mu)^2\\
&\frac{dl}{d\sigma} = -\frac{n}{\sigma} + \frac{1}{\sigma^3}\sum(x_i-\mu)^2\\
&\frac{dl}{d\sigma} = \frac{n}{\sigma^2} - \frac{3}{\sigma^4}\sum(x_i-\mu)^2
\end{aligned}
$$

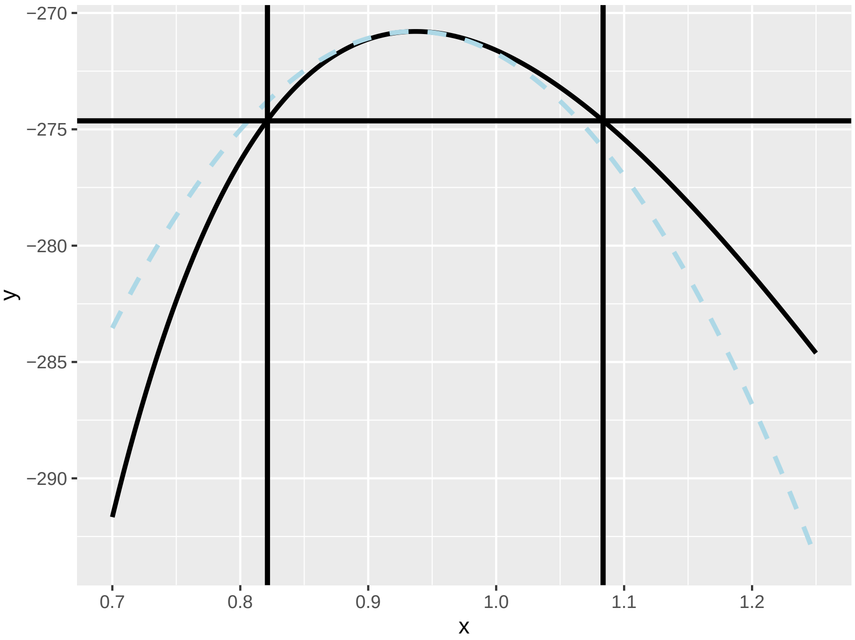
$$
\begin{aligned}
&E\left[\frac{d^2\log f(X|\mu,\sigma)}{d\sigma^2}\right]=\\
&E\left[\frac{1}{\sigma^2}-\frac{3}{\sigma^4}(X-\mu)^2\right]=\\
&\frac{1}{\sigma^2}-\frac{3}{\sigma^4}E(X-\mu)^2=\\
&\frac{1}{\sigma^2}-\frac{3}{\sigma^4}\sigma^2=-\frac{2}{\sigma^2}\\
&v(\sigma)=-\frac{1}{nE\left[\frac{d^2\log f(X|\mu,\sigma)}{d\sigma^2}\right]}=\frac{\sigma^2}{2n}\\
&\hat\sigma\sim N\left(\sigma,\sqrt{v(\sigma)}\right)=N(\sigma,\sigma/\sqrt{2n})
\end{aligned}
$$

and so we find the (approximate) confidence interval

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

Let's illustrate this:

```
alpha <- 0.95; n <- 100; mu0 <- 0; sigma0 <- 1
crit <- qnorm(1-(1-alpha)/2)
x <- rnorm(n, mu0, sigma0)
muhat <- mean(x)
shat <- sqrt(sum((x-muhat)^2)/n)
K <- 2*sum(log(dnorm(x, mu0, shat)))
sig <- seq(sigma0*0.7, sigma0*1.25, length=500)
y <- 0*mu
for(i in 1:500)
  y[i] <- 2*sum(log(dnorm(x, mu0, sig[i])))
Hess <- n/shat^2-3/shat^4*sum((x-mu0)^2)
v <- 1/sqrt(-Hess)
L <- (sig[1:150])[which.min(abs(y[1:150]-K+crit^2))]
U <- (sig[250:500])[which.min(abs(y[250:500]-K+crit^2))]
df <- data.frame(x=sig, y=y, yquad=K-(sig-shat)^2/v^2)
ggplot(df) +
  geom_line(aes(x, y), linetype="solid", size=1.1) +
  geom_line(aes(x, yquad),
            linetype = "dashed", color="lightblue",
            size=1.1) +
  geom_hline(yintercept = K-crit^2, size=1.2) +
  geom_vline(xintercept = c(L, U), size=1.2)
```



```
c(L, U)
```

```
## [1] 0.8212425 1.0835671
```

Now the parabola and the 2 log-likelihood curve are not exactly the same, because the true distribution of the mle is not (for this sample size) a normal. So now we could use the approximating parabola to find intervals (black line) or the exact 2 log-likelihood curve. Which one is better? It depends which one has correct coverage, and the only way to tell is by running a simulation study.

One advantage of the parabola solution is that we can find the interval points explicitly:

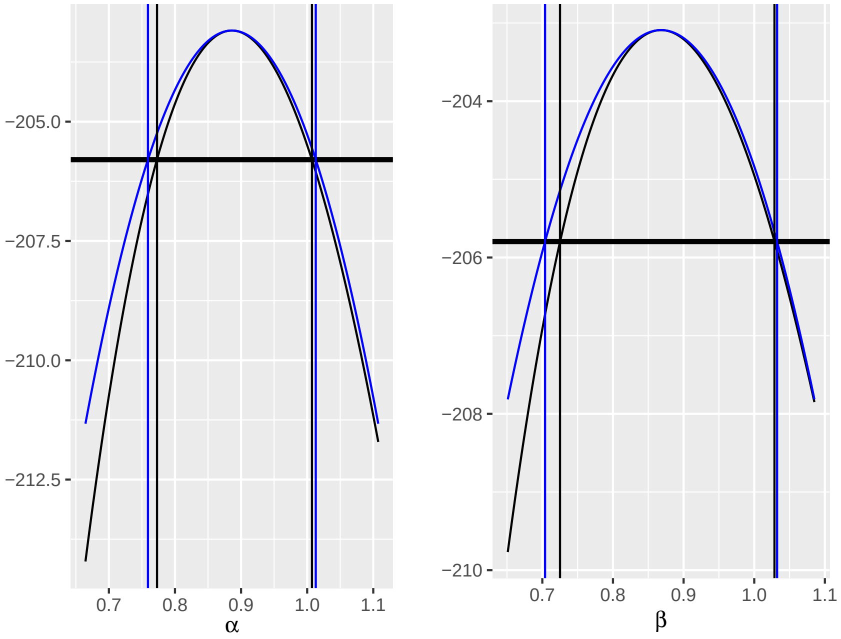
$$t = \hat{s} \pm z_{\alpha/2} v$$

6.2.2.2 Example (6.2.6) Let's say we have $X_1, .., X_n \sim \Gamma(\alpha, \beta)$, both α,β unknown. We want to find confidence intervals for both.

Now we could do the math, but let's try to use R as much as possible. To find the mle we need a routine that maximizes a function of a vector and allows for restricting the range of the variables (because $\alpha, \beta > 0$). One such routine is nlminb.

```
mle.gamma <- function(dens="gamma", alpha=1, beta=1, n=100) {
  library(numDeriv)
  rdens <- get(paste0("r", dens))
  ddens <- get(paste0("d", dens))
  crit <- qnorm(1-0.05)
  dta <- rdens(n, alpha, beta)
```

```
loglike <- function(x, dta)
   (-2)*sum(log(ddens(dta, x[1], x[2])))
z <- nlminb(c(alpha, beta), loglike,
            lower=c(0, 0), dta=dta)
mle <- z$par
K <- (-z$objective)
Hess <- hessian(loglike, x=mle, dta=dta)
a <- seq(mle[1]*0.75, mle[1]*1.25, length=500)
y <- 0*a
for(i in 1:500)
  y[i] <- (-loglike(c(a[i], mle[2]), dta))
y1 <- y[a<mle[1]]
alphaL <- a[a<mle[1]]
alphaL <- alphaL[abs(y1-K+crit^2)==min(abs(y1-K+crit^2))]
y1 <- y[a>mle[1]]
alphaR <- a[a>mle[1]]
alphaR <- alphaR[abs(y1-K+crit^2)==min(abs(y1-K+crit^2))]
v <- sqrt(solve(Hess)[1,1])
alpha2 <- mle[1]+c(-1,1)*v*crit
df <- data.frame(x=a, y=y, yquad=K-(a-mle[1])^2/v^2)
plt1 <- ggplot(df) +
        geom_line(aes(x, y)) +
        geom_line(aes(x, yquad), color="blue") +
        geom_hline(yintercept=K-crit^2, size=1.2) +
        geom_vline(xintercept = c(alphaL, alphaR)) +
        geom_vline(xintercept = alpha2, color="blue") +
        labs(x=expression(alpha), y="")
b <- seq(mle[2]*0.75, mle[2]*1.25, length=500)
y <- 0*b
for(i in 1:500) y[i] <- (-loglike(c(mle[1], b[i]), dta))
v <- sqrt(solve(Hess)[2,2])
beta2 <- mle[2] + c(-1, 1)*v*crit
y1 <- y[b<mle[2]]
betaL <- b[b<mle[2]]
betaL <- betaL[abs(y1-K+crit^2)==min(abs(y1-K+crit^2))]
y1 <- y[b>mle[2]]
betaR <- b[b>mle[2]]
betaR <- betaR[abs(y1-K+crit^2)==min(abs(y1-K+crit^2))]
df <- data.frame(x=b, y=y, yquad=K-(b-mle[2])^2/v^2)
plt2 <- ggplot(df) +
        geom_line(aes(x, y)) +
        geom_line(aes(x, yquad), color="blue") +
        geom_hline(yintercept=K-crit^2, size=1.2) +
        geom_vline(xintercept = c(betaL, betaR)) +
        geom_vline(xintercept = beta2, color="blue") +
        labs(x=expression(beta), y="")
  pushViewport(viewport(layout = grid.layout(1, 2)))
  print( plt1,
    vp=viewport(layout.pos.row=1, layout.pos.col=1))
  print( plt2,
    vp=viewport(layout.pos.row=1, layout.pos.col=2))

}
mle.gamma()
```



How about the parabola solution? For this we need the Hessian matrix, which we can find also using numerical methods. This can be done using the hessian function in the numDeriv library.

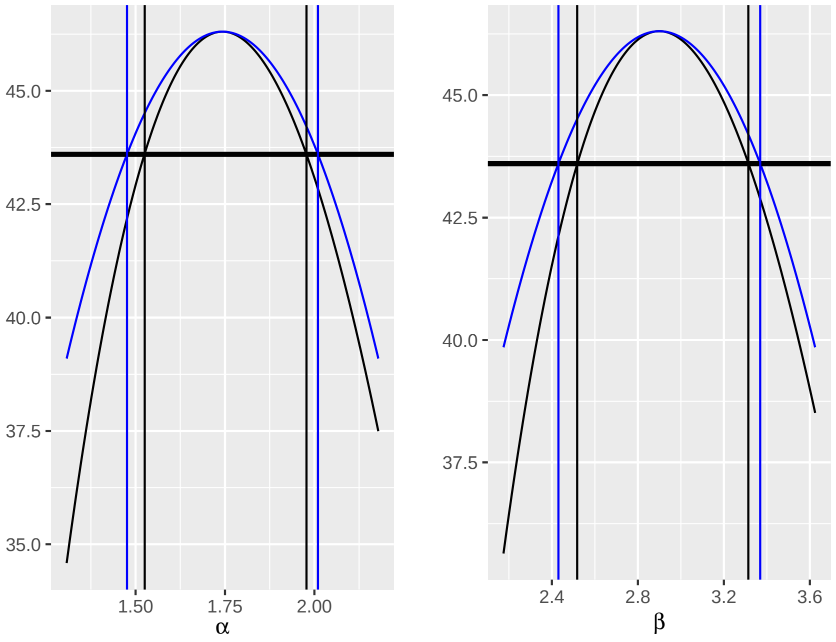
Also new: now we have a two-dimensional problem , so

$$v_i = \sqrt{H^{-1}_{i,i}}$$

6.2.2.3 Example (6.2.7) Let's say we have $X_1, .., X_n \sim Beta(\alpha, \beta)$, both α, β unknown. We want to find confidence intervals for both.

actually, the routine **mle.gamma** works just as is, we only need to call it with

```
mle.gamma("beta", alpha=2, beta=3)
```



In some way what we have here is a fully automatic confidence interval calculator! Of course it is based on a large sample theorem, so to what degree it works (aka the resulting intervals have coverage) needs to be checked in each case.

6.2.3 One-sided Confidence Intervals

Sometimes one is interested in an upper or a lower bound for a parameter, so what we need (for an upper bound) is a function U($\boldsymbol{x}$) such that $P(\theta < U(X)) = 1-\alpha$. We can again derive such an interval by inverting a hypothesis test, this time a test with an alternative of $>$ or $<$.

6.2.3.1 Example (6.2.8) Say we have observations $X_1, .., X_n \sim N(\mu, 1)$. We want a 90% upper bound for μ.

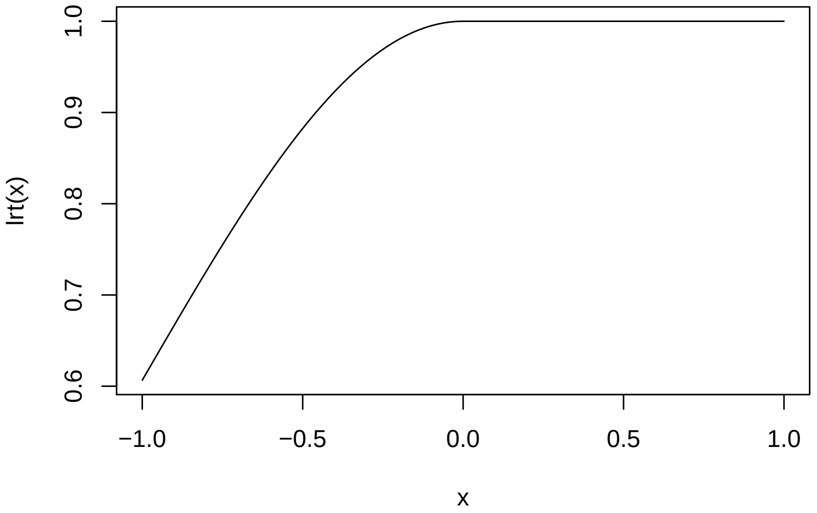
We will derive the LRT test for

$$H_0 : \mu \ge \mu_0 \text{ vs } H_a : \mu < \mu_0$$

We already know that the mle is $\bar{x}$. Now under the null we find the maximum to also $\bar{x}$ if $\bar{x}$ is allowed, that is if $\bar{x} \ge \mu_0$, or the maximum is at μ_0. of course $L(\bar{x})/L(\bar{x}) = 1$, So

$$LRT(\boldsymbol{x}) = \frac{L(\hat{\hat{\mu}})}{L(\hat{\mu})} = \begin{cases} \frac{L(\mu_0)}{L(\bar{x})} & \text{if } \bar{x} < \mu_0 \\ 1 & \text{if } \bar{x} > \mu_0 \end{cases}$$

Here is an example what this might look like:



so clearly “LRT is small” means “$\bar{x}$ is small”. So

$$\begin{aligned}
&\alpha = P(\bar{X} < c) = \\
&P(\sqrt{n}(\bar{X} - \mu_0) < \sqrt{n}(c - \mu_0)) = \Phi(\sqrt{n}(c - \mu_0)) \\
&\sqrt{n}(c - \mu_0) = z_\alpha \\
&c = \mu_0 + \frac{1}{\sqrt{n}} z_\alpha \\
&1 - \alpha = P(\bar{X} > \mu_0 + \frac{1}{\sqrt{n}} z_\alpha) = \\
&P(\mu_0 < \bar{X} - \frac{1}{\sqrt{n}} z_\alpha) = \\
&P(\mu_0 < \bar{X} + \frac{z_{1-\alpha}}{\sqrt{n}})
\end{aligned}$$

6.3 Inference for Binomial p

We have data $Z_1, .., Z_n \sim Ber(p)$ and we want to develop a confidence interval for p.

First we need a corresponding hypothesis test:

$$H_0 : p = p_0 \text{ vs } H_a : p \neq p_0$$

If half-open intervals (“upper or lower limits”) are desired use the corresponding alternative hypotheses.

Let $X = \sum_{i=1}^n Z_i$, then $X \sim Bin(n, p)$

Note: in this example X is the obvious choice to base a test on because $\bar{z} = x/n$ is an unbiased, sufficient and consistent estimator of p.

Next we need a rejection region, that is we need to decide what values of S will lead us to reject the null hypothesis.

And we get to the first fork in the road:

6.3.1 Testing Idea 1:

Let $Y \sim Bin(n, p_0)$, reject H_0 if

$$P(Y < x|p_0) < \alpha/2$$

or if

$$P(Y > x|p_0) < \alpha/2$$

```r
n <- 20; p0 <- 0.3; alpha=0.05
x <- 3
pbinom(x-1, n, p0)
```

6.3.1.1 Example (6.4.1)

```
## [1] 0.03548313
```

$0.0354 > 0.025$ and so we fail to reject the null hypothesis.

```r
x <- 2
pbinom(x-1, n, p0)
```

```
## [1] 0.00763726
```

$0.0076 < 0.025$ and so we reject the null hypothesis.

6.3.2 Testing Idea 2:

Let $TS = \frac{x-np_0}{\sqrt{np_0(1-np_0)}}$ and reject H_0 if $|T| > z_{\alpha/2}$. This idea is obviously based on the central limit theorem.

```
x <- 0:20
TS <- round((x-n*p0)/sqrt(n*p0*(1-p0)), 3)
df <- data.frame(x=x, a=abs(TS),
        b=ifelse(abs(TS)>qnorm(1-alpha/2), "Yes", "No"),
        Decision=ifelse(abs(TS)>qnorm(1-alpha/2),
                        "Reject", "Fail to reject"))
colnames(df)[2:3] <- c(’|TS|’, ’|TS|>crit’)
# critical value
qnorm(1-0.05/2)
```

6.3.2.1 Example (6.4.2)

```
## [1] 1.959964
```

```
kable.nice(df, do.row.names = FALSE)
```

x	|TS|	|TS|>crit	Decision
0	2.928	Yes	Reject
1	2.440	Yes	Reject
2	1.952	No	Fail to reject
3	1.464	No	Fail to reject
4	0.976	No	Fail to reject
5	0.488	No	Fail to reject
6	0.000	No	Fail to reject
7	0.488	No	Fail to reject
8	0.976	No	Fail to reject
9	1.464	No	Fail to reject
10	1.952	No	Fail to reject
11	2.440	Yes	Reject
12	2.928	Yes	Reject
13	3.416	Yes	Reject
14	3.904	Yes	Reject
15	4.392	Yes	Reject
16	4.880	Yes	Reject
17	5.367	Yes	Reject
18	5.855	Yes	Reject
19	6.343	Yes	Reject
20	6.831	Yes	Reject

and so we reject H_0 if $x < 2$ or $x > 10$.

Now we have some tests. To get to a confidence interval we have to "invert" these tests.

The interval will contain all the parameter values that would have lead to accepting the null hypothesis given the observed data.

6.3.3 Interval Idea 1:

For a fixed x find p_1 so that

$$P(Y < x|p_1) = 1 - \alpha/2$$

and p_2 so that

$$P(Y > x|p_2) = 1 - \alpha/2$$

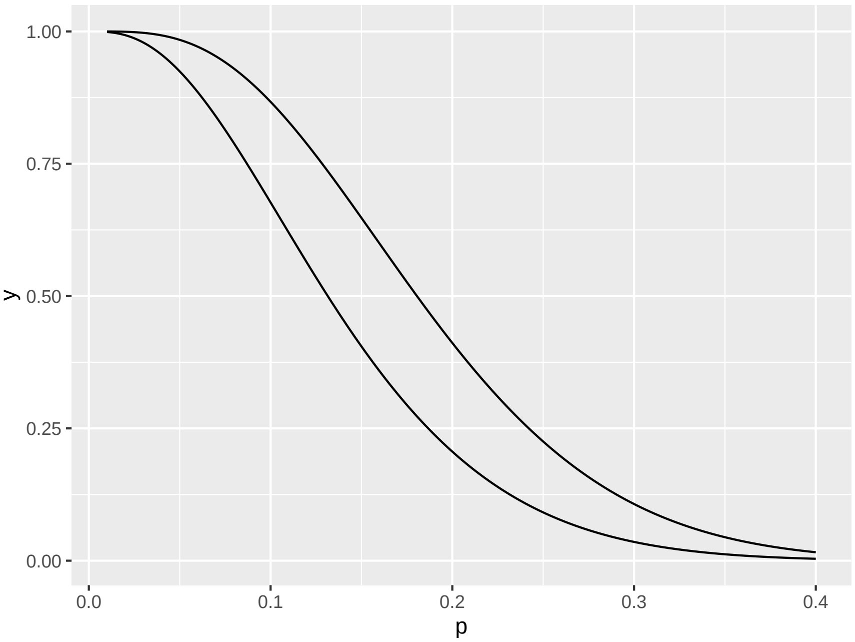
or

$$P(Y \leq x|p_2) = \alpha/2$$

then (p_1, p_2) is the confidence interval.

```
x <- 3; n <- 20; alpha <- 0.05
p <- seq(0.01, 0.4, length=1000)
y1 <- pbinom(x-1, n, p)
y2 <- pbinom(x, n, p)
df <- data.frame(p=c(p, p), y=c(y1, y2),
              which=rep(c("Lower", "Upper"), each=1000))
ggplot(df, aes(p, y, coloer=which)) +
  geom_line()
```

6.3.3.1 Example (6.4.3)



we saw that the curves are strictly decreasing, so we can find the solutions with

```
round(c(p[y1<1-alpha/2][1], p[y2<alpha/2][1]), 4)
```

```
## [1] 0.0323 0.3793
```

This method is called the **Clopper-Pearson** method. It was invented in 1934.

Above we found p_1 and p_2 via a grid search. One could also write a numerical routine to do that.

Or we can use a theorem from probability theory: if $Z \sim Beta(x, n-x+1)$ and $Y \sim Bin(n,p)$, then

$$P(Z<p) = P(Y<x)$$

so with n=x and m=n-x+1 we have

$$\begin{aligned}&1-\alpha/2 = P(Y<x|p) = \\ &P(Z<p|x,n-x+1) = \\ &p = qbeta(1-\alpha/2|x,n-x+1)\end{aligned}$$

```
round(c(qbeta(0.025, 3, 18), qbeta(0.975,4,17)), 4)
```

```
## [1] 0.0321 0.3789
```

This is already implemented in the base R routine

```
round(c(binom.test(3, 20)$conf.int), 4)
```

```
## [1] 0.0321 0.3789
```

6.3.4 Innterval Idea 2:

the acceptance region is

$$-z_{\alpha/2} < \frac{x-np_0}{\sqrt{np_0(1-np_0)}} < z_{\alpha/2}$$

so we need to solve these equations for p_0. Again we have choices:

Option 1:

x/n is an estimator of p_0 so let's replace the p_0 in the denominator with x/n:

$$p = x/n \pm z_{\alpha/2}\sqrt{x/n(1-x/n)}$$

```
x <- 3; n <- 20; alpha <- 0.05
round(x/n + c(-1, 1)*qnorm(1-alpha/2)*sqrt(x/n*(1-x/n)), 4)
```

6.3.4.1 Example (6.4.4)

```
## [1] -0.5498  0.8498
```

This is called a Wald type interval (because we replace var(X) with an estimate). It was invented in 1948.

We already see one obvious "bad" feature: the lower bound is negative. First of all p can never be negative. More than that because we observed x=3 we know p must be positive.

Option 2

Work a little harder

$$
\begin{aligned}
&\frac{x-np}{\sqrt{np(1-p)}} = \pm z_{\alpha/2} \\
&\frac{(x-np)^2}{np(1-p)} = \pm z_{\alpha/2}^2 =: z \\
&(x-np)^2 = znp(1-p) \\
&n(n+z)p^2 - n(2x+z)p + x^2 = 0 \\
&p_{1,2} = \frac{1}{2n(n+z)}\left(n(2x+z) \pm \sqrt{[n(2x+z)]^2 - 4n(n+z)x^2}\right) = \\
&\frac{1}{n+z}\left(x + z/2 \pm \sqrt{zx + z^2/4 - zx^2/n}\right)
\end{aligned}
$$

```
x <- 3; n <- 20; alpha <- 0.05
z <- qnorm(1-alpha/2)^2
round( (x+z/2+c(-1, 1)*sqrt(z*x+z^2/4-z*x^2/n))/(n+z), 4)
```

6.3.4.2 Example (6.4.5)

```
## [1] 0.0524 0.3604
```

This is called a *Wilson interval*. It was invented in 1927.

6.3.5 Ad-Hoc Adjustments

There have been a number of adjustments suggested for a variety of these. Here is one example:

we are approximating a discrete rv by a continuous one, so maybe it is a good idea to correct for that a bit by subtracting (for lower bound) and adding (for upper bound) 1/2 to x.

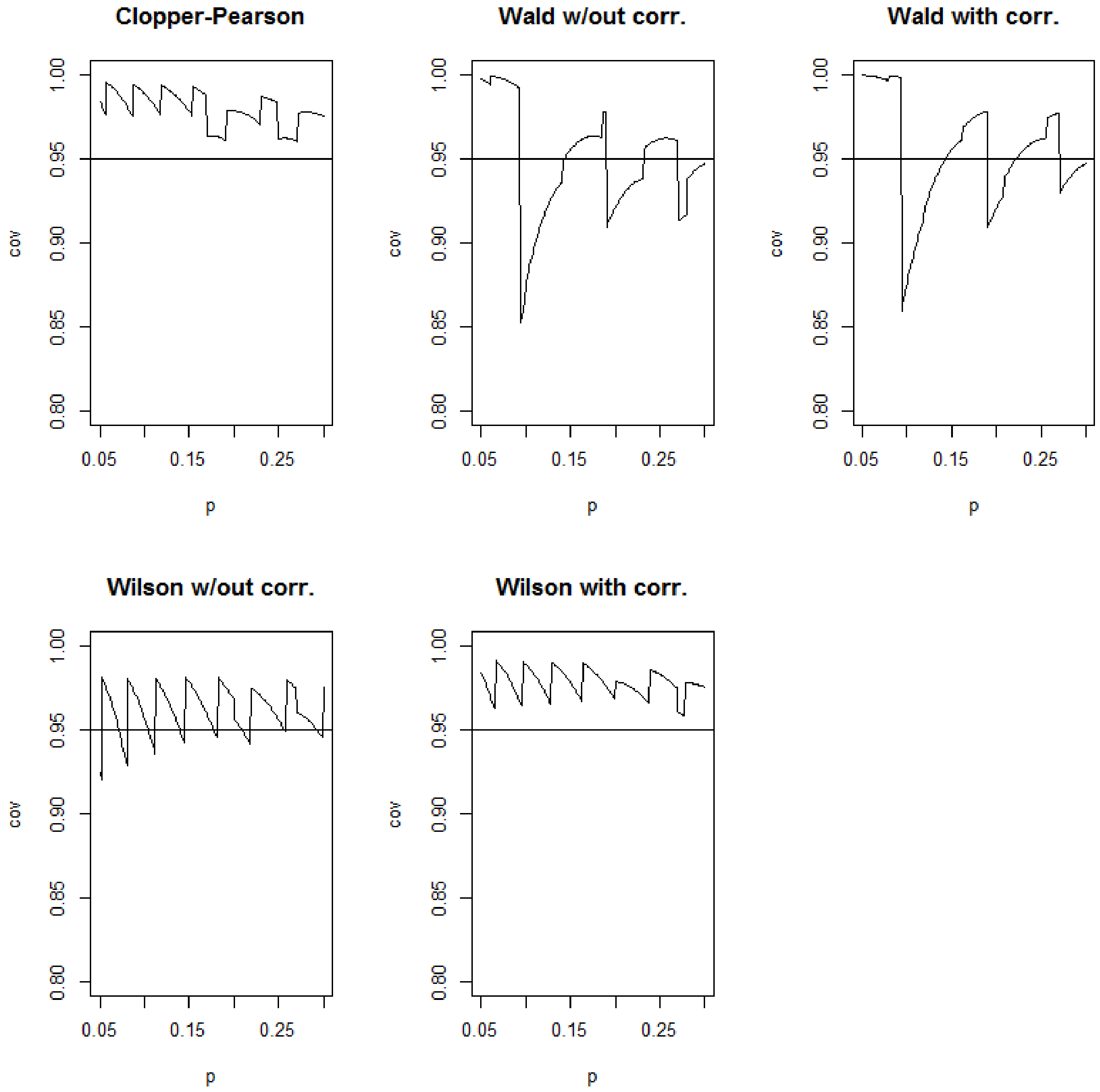
This can be applied to both the Wald and the Wilson type intervals. For the Wald test this already done in the built-in function **prop.test**.

Note there is a similar adjustment for Clopper-Pearson intervals called mid-p intervals, which we won't have time to discuss.

6.3.6 How to choose:

How do we pick between these? First we need to make sure that the methods have coverage

6.3.6.1 Example (6.4.6) say we have n=20 and p=3, so x/n=0.15, so the true p is probably between 0.05 and 0.3 . Let's do the coverage graphs:

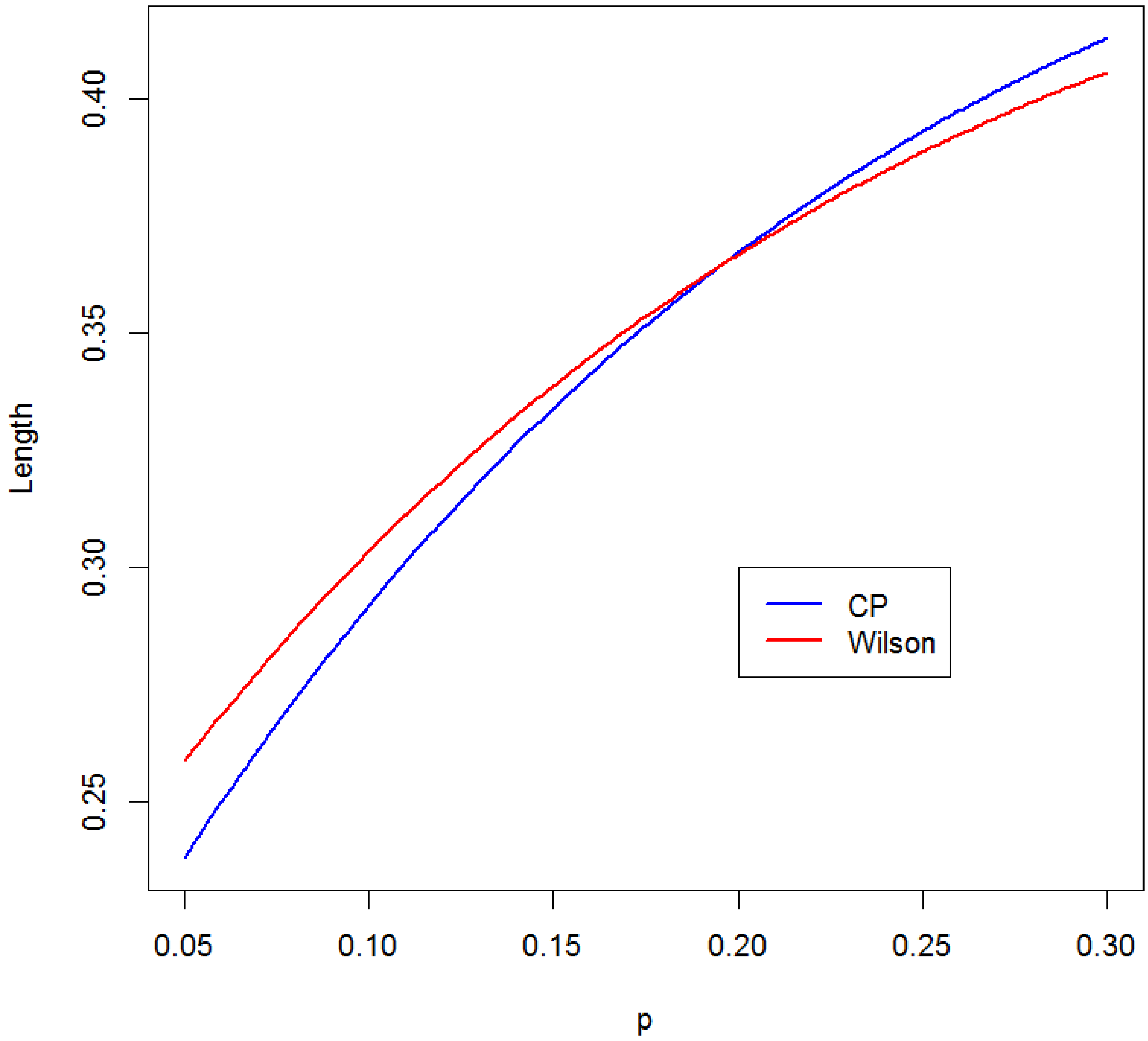


so the only two that "work" are Clopper-Pearson and Wilson with continuity correction.

How do we choose between these two? One possible criterion is the expected length of the interval:

$$e(p) = E\left[U(X) - L(X)\right] = \sum_{i=0}^{n} (u(i) - l(i)) \binom{n}{i} p^i (1-p)^{n-i}$$

drawn here:



so for p around 0.15 they are pretty much the same.

6.4 A Longer Example - Intervals

$X_1, .., X_n \sim Beta(a, 1)$, and we want to find a $(1 - \alpha)100\%$ confidence interval for a.

6.4.1 Frequentist Intervals

Let's again first consider the case n=1. Previously we had the following hypothesis test for

$$H_0 : a = a_0 \text{ vs } H_a : a \neq a_0$$

reject the null hypothesis if

$x < (\alpha/2)^{1/a0}$ or $x > (1-\alpha/2)^{1/a0}$

therefore we find

$$x = (1-\alpha/2)^{1/a} \log x = \frac{1}{a}\log(1-\alpha/2) a = \frac{\log(1-\alpha/2)}{\log x}$$

and so we have the interval

$$\left(\frac{\log(1-\alpha/2)}{\log x}, \frac{\log(\alpha/2)}{\log x}\right)$$

```
x <- 0.45;alpha <- 0.05
round(c(log(1-alpha/2), log(alpha/2))/log(x), 3)
```

```
## [1] 0.032 4.620
```

Here we split α 50-50 on the left and the the right. Is this optimal? Let's put $\lambda\alpha$ on the left and $(1-\lambda)\alpha$ on the right for some $0 \le \lambda \le 1$. Then

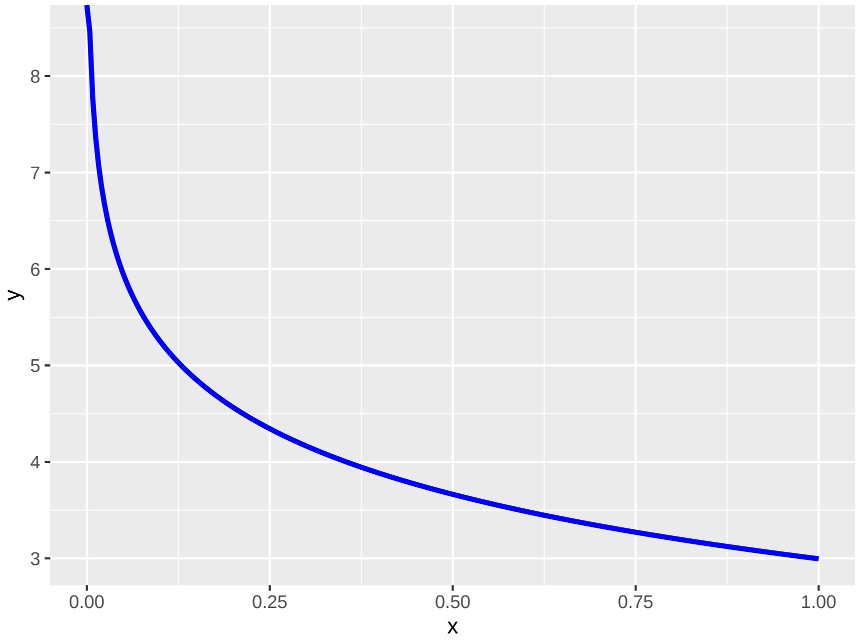
$$\begin{aligned}
&\lambda\alpha = P(X < c_1) = c_1^a \text{, so } c_1 = (\lambda\alpha)^{1/a} \\
&a = \log(\lambda\alpha)/\log x \\
&(1-\lambda)\alpha = P(X < c_2) = 1 - P(X < c_2) = 1 - c_2^a \\
&c2 = [1-(1-\lambda)\alpha)]^{1/a} \\
&a = \log(1-(1-\lambda)\alpha)/\log x
\end{aligned}$$

so the interval is of the form

$$\left(\log(\lambda\alpha), \log(1-(1-\lambda)\alpha))\right) / \log x$$

say we want to find a shortest interval. Let's draw the length of the interval as a function of λ:

```
fun <- function(lambda, alpha=0.05)
  log(1-(1-lambda)*alpha)-log(lambda*alpha)
ggcurve(fun=fun, A=0, B=1)
```



so this is strictly decreasing, so it is minimized at $\lambda = 1$, and we find the optimal interval to be

$$(0, \log(1-\alpha)) \,/\, \log x$$

and now if we observe X=0.45 a 95% CI for a is

```
round(c(0, log(alpha))/log(x), 3)
```

```
## [1] 0.000 3.752
```

Note that these intervals always start at 0, which might not be a good idea for some experiments if it is known that $a > 0$.

How about the case n=2?

We have qbeta2($\alpha/2, a$)=x which is equivalent to pbeta2(x,a)=$\alpha/2$, and we need to solve this for a. Again this needs to be done numerically:

```
invbeta2 <- function (x, alpha=0.05)  {
  x <- sum(x)
  a <- 0
  repeat {
    a <- a+0.001
    if(pbeta2(x, a)<1-alpha/2)
      break
  }
  L <- a
  repeat {
    a <- a+0.001
    if(pbeta2(x, a)<alpha/2)
      break
  }
  c(L, a)
}
invbeta2(c(0.31, 0.59))
```

```
## [1] 0.078 3.043
```

For example, say we observe (x,y) = (0.31, 0.59) note that (x+y)/2 = 0.45, same as above with n=1. Then we find a 95% CI for a to be (0.078, 3.043).

Previously we also used simulation to estimate pbeta2(x,a). If we could not calculate it numerically we could use this as well, but clearly we are going to get routines that take take rather a long time to run.

Finally for the case of large n we had the Wald test: reject H_0 if

$$\bar{x} < \frac{1}{a_0+1}\left(a_0 - z_{\alpha/2}\sqrt{\frac{a_0}{n(a_0+2)}}\right)$$

or

$$\bar{x} > \frac{1}{a_0+1}\left(a_0 + z_{\alpha/2}\sqrt{\frac{a_0}{n(a_0+2)}}\right)$$

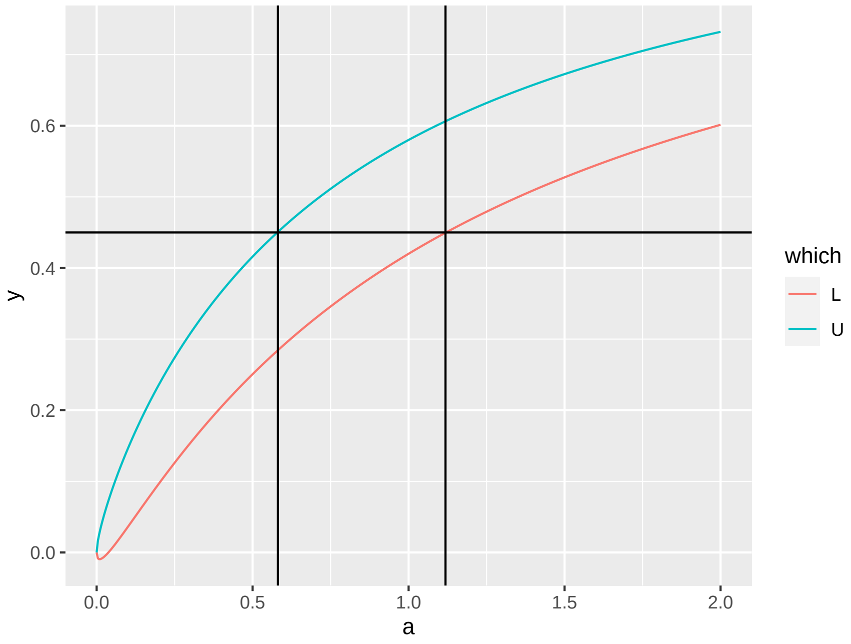
so the acceptance region of the test is

$$\frac{1}{a_0+1}\left(a_0 - z_{\alpha/2}\sqrt{\frac{a_0}{n(a_0+2)}}\right) < \bar{x} < \frac{1}{a_0+1}\left(a_0 + z_{\alpha/2}\sqrt{\frac{a_0}{n(a_0+2)}}\right)$$

to get a confidence interval we need to "invert the test". This means solving the double-inequality in the acceptance region for a (which now replaces a_0). The next graph shows the left and the right side of the double-inequality as a function of a:

```
n <- 50; xbar <- 0.45; alpha <- 0.05
crit <- qnorm(1-alpha/2)
a <- seq(0, 2, length=500)
y1 <- (a-sqrt(a/(n*(a+2)))*crit)/(a+1)
```

```
y2 <- (a+sqrt(a/(n*(a+2)))*crit)/(a+1)
R <- a[which.min(abs(y1-xbar))]
L <- a[which.min(abs(y2-xbar))]
df <- data.frame(a=c(a, a), y=c(y1, y2),
        which=rep(c("L", "U"), each=500))
ggplot(df, aes(a, y, color=which)) +
  geom_line() +
  geom_hline(yintercept = xbar) +
  geom_vline(xintercept = c(L, R))
```



```
round(c(L, R), 3)
```

```
## [1] 0.581 1.118
```

Formally we need to solve the equation

$$x = \frac{1}{a+1}\left(a \pm z\sqrt{\frac{a}{n(a+2)}}\right)$$

doing the arithmetic yields the cubic equation

$$n(x-1)^2a^3 + 2n(2x-1)(x-1)a^2 + [nx(5x-4) - z^2]a + 2nx^4 = 0$$

so now we have a cubic equation, which we can solve. In R this is done by the routine **polyroot**:

```
z <- qnorm(alpha/2)
cf <- c(n*(xbar-1)^2, 2*n*(2*xbar-1)*(xbar-1),
        n*xbar*(5*xbar-4)-z^2, 2*n*xbar^2)
round(Re(polyroot(cf[4:1])), 3)
```

```
## [1]  0.579 -2.063  1.120
```

Of course there are generally three roots, but one of them is negative and the other two are our solutions.

How about the mle? This one is easy: recall that $T = -\sum \log x_i$, and

$$\begin{aligned}
&1-\alpha = \\
&P((qgamma(\alpha/2,n,1) < aT < qgamma(1-\alpha/2,n,1))) = \\
&P((qgamma(\alpha/2,n,1)/T < a < qgamma(1-\alpha/2,n,1)/T))
\end{aligned}$$

6.4.2 Bayesian Interval

Previously we saw that if we use a prior Exp(1) we get a posterior distribution

$$a|\boldsymbol{x} \sim \Gamma(n+1, 1/(T+1))$$

using this we can find the equal-tail probability interval by solving

$$\alpha/2 = P(a < l|\boldsymbol{x}) = pgamma(Tl, n+1, 1/(T+1))$$

so we have

$$l = qgamma(\alpha/2, n+1, 1/(T+1))u = qgamma(1-\alpha/2, n+1, 1/(T+1))$$

which is (almost) the same as the confidence interval based on the LRT test.

6.4.3 Which Interval is Better?

First of frequentist confidence intervals and Bayesian credible intervals can not really be compared directly. We will just compare the two confidence intervals.

First we need to check that the two methods yield proper confidence intervals, that is that they have **coverage**. So if we calculate a 90% interval it really contains the true parameter 90% of the time. This is true for the LRT interval by construction because we could find the exact distribution of the LRT statistic and invert the interval analytically. The other test is a large sample test and needs to be checked via simulation. Here are some results (red=MM, blue=LRT)

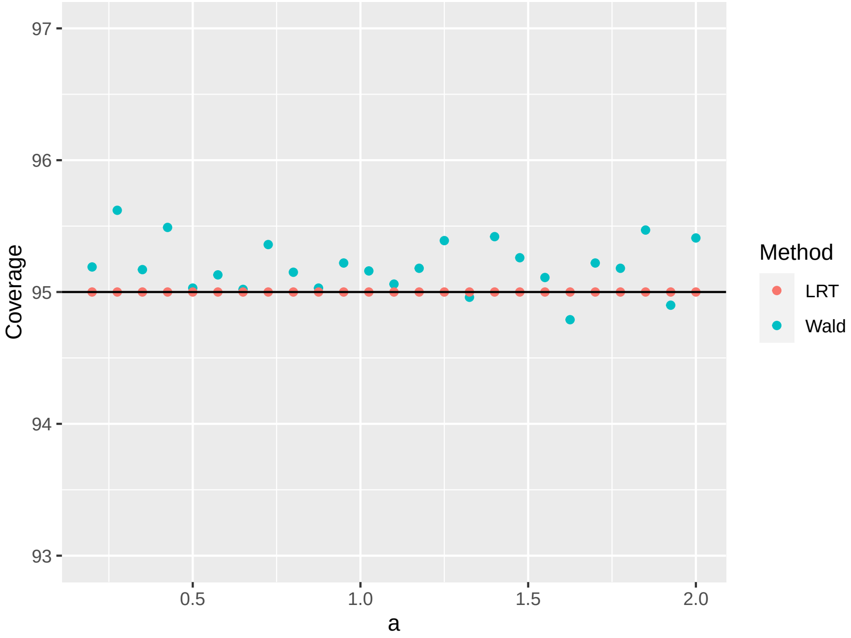
```r
wald.test <- function(x,  alpha=0.05 ) {
  Ts=-sum(log(x))
  n=length(x)
  tmp=qgamma(1-alpha/2, n, 1)/Ts
  crit <- qnorm(1-alpha/2)
  a <- seq(0, 2*tmp, length=500)
  n <- length(x)
  xbar <- mean(x)
  y1 <- (a-sqrt(a/(n*(a+2)))*crit)/(a+1)
  y2 <- (a+sqrt(a/(n*(a+2)))*crit)/(a+1)
  R <- a[which.min(abs(y1-xbar))]
  L <- a[which.min(abs(y2-xbar))]
  round(c(L, R), 3)
}
lrt.test <- function(x,  alpha=0.05 ) {
  Ts=-sum(log(x))
  n=length(x)
  round(c(qgamma(alpha/2, n, 1)/Ts, qgamma(1-alpha/2, n, 1)/Ts), 3)
}
```

```r
coverage <- function(n, a, B=1e4) {
  A=rep(0,B)
  for(i in 1:B) {
    x=rbeta(n, a, 1)
    tmp=wald.test(x)
    if(tmp[1]<a & a<tmp[2]) A[i]=1
  }
  sum(A)/B
}
a=seq(0.2, 2, length=25)
out=a
for(i in seq_along(a))
  out[i]=coverage(10, a[i])
```

```r
rbind(a,out)
```

```
##       [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8]   [,9]  [,10]  [,11]  [,1
## a   0.2000 0.2750 0.3500 0.4250 0.5000 0.5750 0.6500 0.7250 0.8000 0.8750 0.9500 1.02
## out 0.9519 0.9562 0.9517 0.9549 0.9503 0.9513 0.9502 0.9536 0.9515 0.9503 0.9522 0.95
##      [,13]  [,14]  [,15]  [,16]  [,17]  [,18]  [,19]  [,20]  [,21]  [,22]  [,23] [,24
## a   1.1000 1.1750 1.2500 1.3250 1.4000 1.4750 1.5500 1.6250 1.7000 1.7750 1.8500 1.92
## out 0.9506 0.9518 0.9539 0.9496 0.9542 0.9526 0.9511 0.9479 0.9522 0.9518 0.9547 0.94
##      [,25]
## a   2.0000
## out 0.9541
```

```r
df=data.frame(a=c(a,a),
              Coverage=c(100*out, rep(95,length(out))),
              Method=rep(c("Wald", "LRT"), each=length(out)))
ggplot(data=df, aes(a, Coverage, color=Method)) +
  geom_point() +
  geom_hline(yintercept = 95) +
  lims(y=c(93,97))
```



so we see that the Wald method has some over-coverage. Over-coverage is acceptable but not really desirable because it generally means larger intervals.

How can we choose between the two? We need a measure of performance for a confidence interval. Again we can consider the mean length of the interval E[U(X)-L(X)]. In the case of the MM method this will need to be found via simulation. As for the LRT method, we have

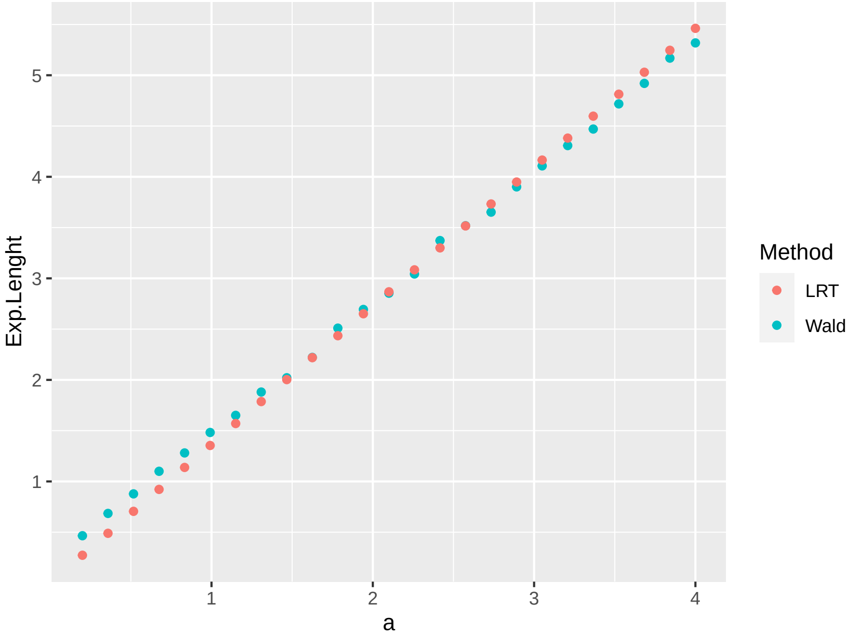
$$
\begin{aligned}
&E\left[U(\boldsymbol{x})-L(\boldsymbol{x})\right]= \\
&[qgamma(1-\alpha/2,n,1)-qgamma(\alpha/2,n,1)]\,E[1/T]= \\
&[qgamma(1-\alpha/2,n,1)-qgamma(\alpha/2,n,1)]\,\frac{1}{n}E[n/T]= \\
&[qgamma(1-\alpha/2,n,1)-qgamma(\alpha/2,n,1)]\,\frac{1}{n}\frac{na}{n-1}= \\
&[qgamma(1-\alpha/2,n,1)-qgamma(\alpha/2,n,1)]\,\frac{a}{n-1}
\end{aligned}
$$

Here is a graph of the mean lengths:

```
expected.length <- function(n, a, B=1e3) {
  A=rep(0,B)
  for(i in 1:B) {
    x=rbeta(n, a, 1)
    A[i]=diff(wald.test(x))
  }
  mean(A)
}
a=seq(0.2, 4, length=25)
out=a
for(i in seq_along(a))
  out[i]=expected.length(10, a[i])
```

```
el.lrt=(qgamma(0.975,10,1)-qgamma(0.025,10,1))*a/9
```

```
df=data.frame(a=c(a,a),
              Exp.Lenght=c(out, el.lrt),
              Method=rep(c("Wald", "LRT"), each=length(out)))
ggplot(data=df, aes(a, Exp.Lenght, color=Method)) +
  geom_point()
```



and the Wald test is a bit better if $a < 2$, otherwise the LRT test is better.

Above we used the fact that aT has distribution that does not depend on the parameter a. We then found the interval by dividing the error probability α equally on the left and the right. But is that the best option? If we are interested in shortest-length intervals, can we find the intervals that are optimal? That is can we find L(T) and U(T) such that minimize $E[U(T)-L(T)]$ subject to $P(L(T)<a<U(T))=1-\alpha$

Let's set $q(z)=qgamma(z,n,1)$ and let's consider intervals of the form $L(T)=q(z_1)/T$ and $U(T)=q(1-z_2)/T$.

Clearly we need $z_1+z_2=1-\alpha$, so we have $z_2=1-\alpha$-z_1$ or we just write

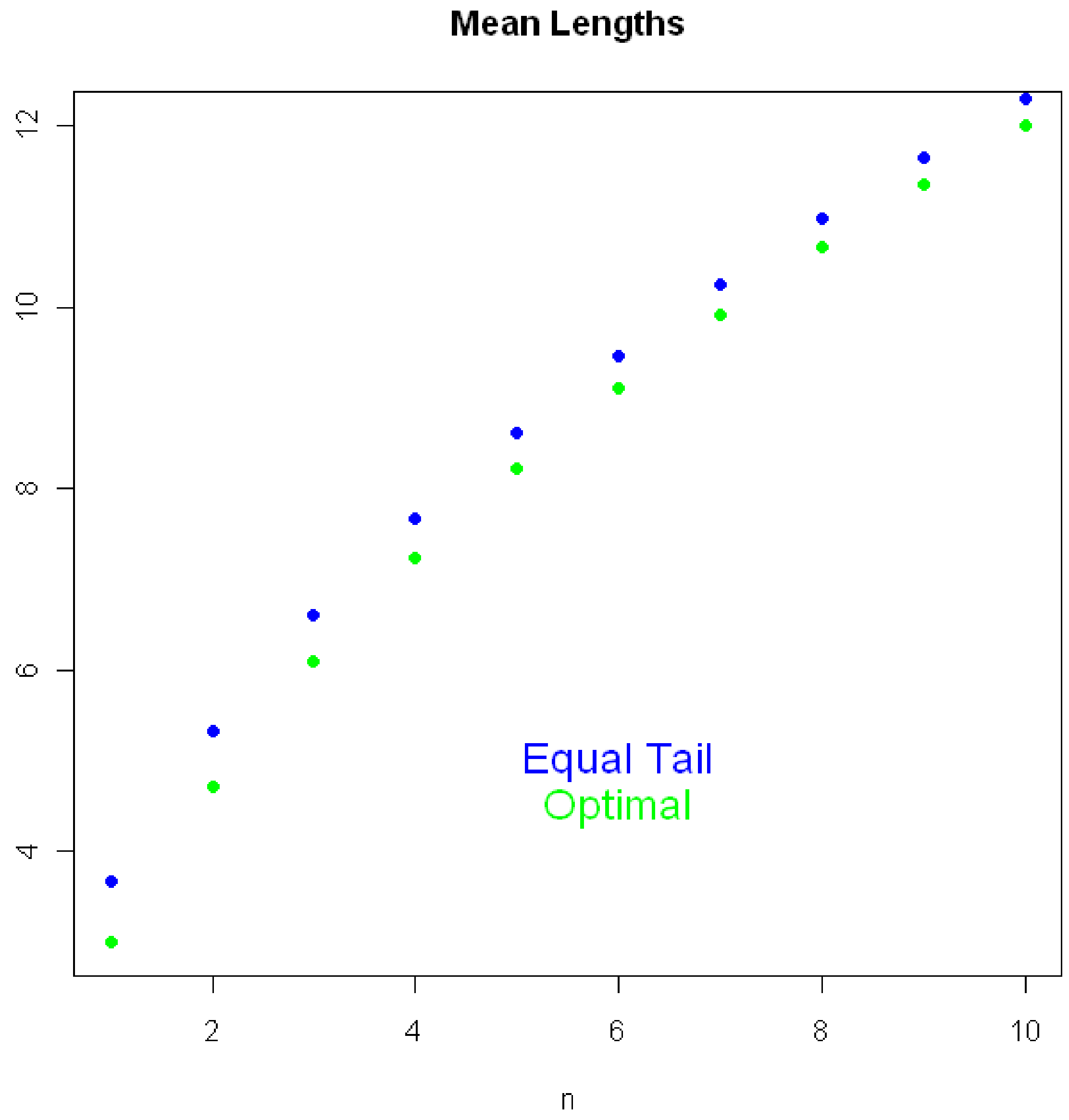
$L(T)=q(z)/T$ and $U(T)=q(1-\alpha-z)/T$

This has length

$$E\left[U(\boldsymbol{x})-L(\boldsymbol{x})\right]=\left[q(1-\alpha-1,n,1)-q(z,n,1)\right]\frac{a}{n-1}$$

This can not be done analytically, and so again we need to resort to a numerical solution. It turns out that assigning a smaller part of α to the left side leads to shorter intervals.

How much better are these intervals? Here are the mean lengths (for a=1)



For example if n=2 (and a=1) the mean length of the equal-tail intervals is 5.33 but for the "optimal" length intervals it is 4.72, an improvement of about 10%.

7 Some Standard Problems

7.1 Sample Size

Among the more important issues in Statistics are questions concerning the sample size required to achieve an answer of a certain quality. In this section we will study these issues.

7.1.1 Sample Size for Confidence Intervals

When we find an interval estimate of a parameter the width of the interval is an indicator of how well we are estimating the parameter. Specifically one consider 1/2 of the width, which is then called the error E.

7.1.1.1 Example (7.1.1) *Normal Mean*

A $(1-\alpha)100\%$ confidence interval for the normal mean with unknown standard deviation is given by $\bar{x} \pm t_{1-\alpha/2,n-1}s/\sqrt{n}$, so we have $E = t_{1-\alpha/2,n-1}s/\sqrt{n}$.

Notice that E here is a random variable as it depends on the sample standard deviation s. It might therefore be better to consider $E[E] = E[t_{1-\alpha/2,n-1}s/\sqrt{n}] = t_{1-\alpha/2,n-1}\sigma/\sqrt{n}$

7.1.1.2 Example (7.1.2) *Binomial p*

We previously found the the Clopper-Pearson intervals for a Binomial p to be

$$\begin{aligned} L(x) &= qbeta(\alpha/2, x, n-x+1) \\ U(x) &= qbeta(1-\alpha/2, x, n-x+1) \end{aligned}$$

so

$$E = \left(qbeta(1-\alpha/2, x, n-x+1) - qbeta(\alpha/2, x, n-x+1)\right)/2$$

Notice that one of the items in the calculation of the estimation error is the sample size n. So one can turn this around ask the question: what sample size is needed to achieve a specified error E?

7.1.1.3 Example (7.1.3) *Normal Mean*

We have $E = t_{1-\alpha/2,n-1}s/\sqrt{n}$, so we have $n = \frac{t^2_{1-\alpha/2,n-1}s^2}{E^2}$. However, we can not find the right side because it includes n twice:

- in $t_{1-\alpha/2,n-1}$. Here one generally assumes that n will be sufficiently large so that $t_{1-\alpha/2,n-1} \approx z_{1-\alpha/2}$, which means $n > 30$ or so.
- in s^2. Here we will need some idea of the population standard deviation. Generally one needs to have some information from similar experiments, or one has to do a pilot study.

What should E be? This depends on the experiment and what kind of error is acceptable to still have a useful.

**Numerical example*: say we know $\sigma^2 = 12.7$, we want $E = 0.5$ and find a 90% confidence interval, the the required sample size is

```r
round(qnorm(1-0.1/2)^2*12.7/0.5^2)
```

```
## [1] 137
```

7.1.1.4 Example (7.1.4) *Binomial p*

We wish to check the parts in a shipment for faulty ones. Typically about 1 in 20 will be faulty. We want to find a 95% confidence interval with an error no more than 0.03.

```r
E=function(n) {
  x=round(n/20)
  (qbeta(1-0.05/2, x, n-x+1)-qbeta(0.05/2, x, n-x+1))/2
}
n=20
repeat {
  n=n+1
  if(n%%10==0) cat(n,"  ",round(E(n), 4),"\n")
  if(E(n)<0.03) break
}
```

```
## 30    0.082
## 40    0.0627
## 50    0.0508
## 60    0.0524
## 70    0.0522
## 80    0.046
## 90    0.041
## 100    0.0414
## 110    0.0413
## 120    0.038
## 130    0.0352
## 140    0.0353
## 150    0.0352
## 160    0.0331
## 170    0.0312
## 180    0.0313
```

```r
n
```

```
## [1] 189
```

7.1.2 Sample Size when Testing

In this section we will consider the question of sample size if we plan on doing a hypothesis test. In this context the role of the estimation error is played by the power of the test. In other words we want a sufficiently large sample size so that the power of the test is reasonably large. This often means a power of at least 80%.

Of course the power of a test also depends on how wrong the null hypothesis is, that is the difference between the value of the parameter under the null hypothesis θ_0 and the true value θ_1. However, we do not know the true value θ_1! In this case the expert in the subject matter needs to decide on an *effect size*, that is the smallest difference between θ_0 and θ_1 that is of practical importance. For example, say we test a new medication for a disease that using current treatments is cured in 10 day. If our new treatment cures the disease in 8 days, that seems quite important, but if it does it in 9.8 days, maybe not. So maybe the doctors tell us that anything less than a 1 day improvement is not enough.

7.1.2.1 Example (7.1.5) *Normal Mean*

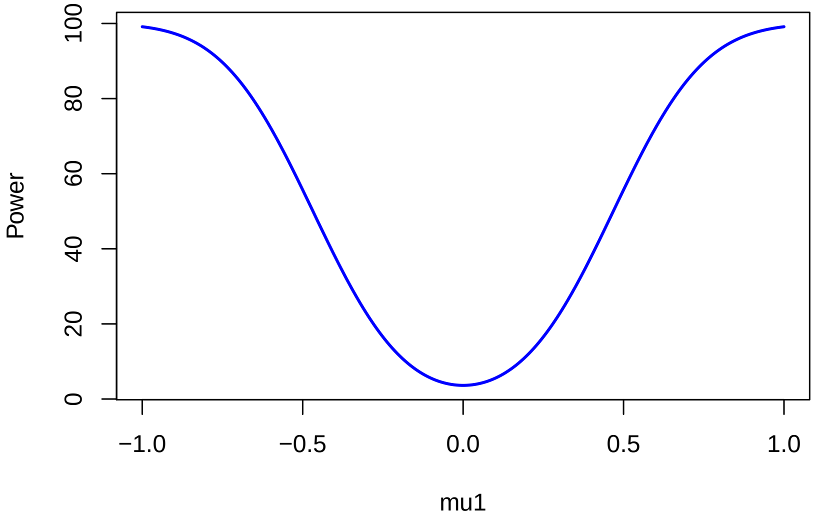
We previously found a test based on the test statistic $T = \sqrt{n}(\bar{x} - \mu_0)/s$ and a rejection region $|T| > t_{1-\alpha/2,n-1}$. So we know

$$\begin{aligned}
&P\left(\text{reject } H_0|H_0 \text{ is true}\right) = \\
&P\left(|\sqrt{n}(\bar{X} - \mu_0)/s| > t_{1-\alpha/2,n-1}\Big|\, \mu = \mu_0\right) = 1-\alpha
\end{aligned}$$

now

$$\begin{aligned}
&\text{Power} = P\left(\text{reject } H_0|H_0 \text{ is false}\right) = \\
&P_{\mu_1}\left(|\sqrt{n}(\bar{X} - \mu_0)/s| > t_{1-\alpha/2,n-1}\right) = \\
&1 - P_{\mu_1}\left(|\sqrt{n}(\bar{X} - \mu_0)/s| < t_{1-\alpha/2,n-1}\right) = \\
&1 - P_{\mu_1}\left(-t_{1-\alpha/2,n-1} < \sqrt{n}(\bar{X} - \mu_0)/s < t_{1-\alpha/2,n-1}\right) = \\
&1 - P_{\mu_1}\left(-t_{1-\alpha/2,n-1} < \sqrt{n}(\bar{X} - \mu_1 + \mu_1 - \mu_0)/s < t_{1-\alpha/2,n-1}\right) = \\
&1 - P_{\mu_1}\left(-t_{1-\alpha/2,n-1} < \sqrt{n}(\bar{X} - \mu_1)/s + \sqrt{n}(\mu_1 - \mu_0)/s < t_{1-\alpha/2,n-1}\right) = \\
&1 - P_{\mu_1}\left(-t_{1-\alpha/2,n-1} - \sqrt{n}(\mu_1 - \mu_0)/s < \sqrt{n}(\bar{X} - \mu_1)/s < t_{1-\alpha/2,n-1} - \sqrt{n}(\mu_1 - \mu_0)/s\right) = \\
&1 - \left(\Phi(t_{1-\alpha/2,n-1} - \sqrt{n}(\mu_1 - \mu_0)/s) - \Phi(-t_{1-\alpha/2,n-1} - \sqrt{n}(\mu_1 - \mu_0)/s)\right)
\end{aligned}$$

```r
pwr.normal=function(n, mu0=0,mu1=1,sigma=1,alpha=0.05) {
  crit=qt(1-alpha/2, n-1)
  tmp=sqrt(n)*(mu1-mu0)/sigma
  100*(1-(pnorm(crit-tmp)-pnorm(-crit-tmp)))
}
mu1=seq(-1, 1, length=250)
plot(mu1, pwr.normal(20, mu1=mu1), type="l", lwd=2,col="blue", ylab="Power")
```



If we now fix mu1 we can also find n:

```
n=2
repeat {
  n=n+1
  if(pwr.normal(n, mu1=1)>80) break
}
cat(n, " ", round(pwr.normal(n, mu1=1)),"\n")
```

```
## 10   82
```

and so here a sample of size 10 is required for a power of 80%.

7.1.2.2 Example (7.1.6) Binomial p

We saw before that a test can be based on $Z = \frac{X-np_0}{\sqrt{np_0(1-p_0)}} \sim N(0,1)$ and we reject the null hypothesis if $|Z| > z_{1-\alpha/2}$. Let's use this to find the sample size required if $p_0 = 0.25, p_1 = 0.3$ and $\alpha = 0.05$.

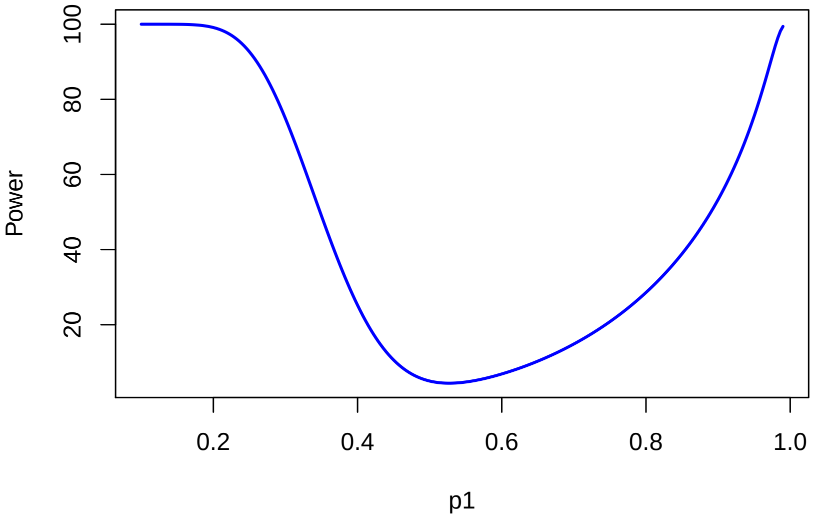
$$\begin{aligned}
&\text{Power} = P\left(\text{reject } H_0|H_0 \text{ is false}\right) = \\
&P\left(|\frac{X-np_0}{\sqrt{np_0(1-p_0)}}| > z_{1-\alpha/2}|p=p_1\right) = \\
&1-P\left(-z_{1-\alpha/2} < \frac{X-np_0}{\sqrt{np_0(1-p_0)}} < z_{1-\alpha/2}|p=p_1\right)
\end{aligned}$$

Now

$$
\begin{aligned}
&P\left(\frac{X-np_0}{\sqrt{np_0(1-p_0)}}<z\right)=\\
&P\left(\frac{X-np_0}{\sqrt{np_1(1-p_1))}}<z\sqrt{\frac{p_1(1-p_1))}{p_0(1-p_0))}}\right)=\\
&P\left(\frac{X-np_1}{\sqrt{np_1(1-p_1)}}<z\sqrt{\frac{p_1(1-p_1))}{p_0(1-p_0)}}+\frac{n(p_0-p_1)}{\sqrt{np_1(1-p_1)}}\right)=\\
&\Phi(z\sqrt{\frac{p_1(1-p_1))}{p_0(1-p_0)}}+\frac{n(p_0-p_1)}{\sqrt{np_1(1-p_1)}})
\end{aligned}
$$

and so

```
pwr.binomial=function(n, p0=0.5,p1=0.6,alpha=0.05) {
  tmp1=-qnorm(1-alpha/2)*sqrt(p1*(1-p0)/p0/(1-p1))+
    n*(p0-p1)/sqrt(n*p1*(1-p1))
  tmp2=qnorm(1-alpha/2)*sqrt(p1*(1-p0)/p0/(1-p1))+
    n*(p0-p1)/sqrt(n*p1*(1-p1))
  100*(1-(pnorm(tmp2)-pnorm(tmp1)))
}
p1=seq(0.1, 0.99, length=250)
plot(p1, pwr.binomial(20, p1=p1), type="l", lwd=2,col="blue", ylab="Power")
```



Also, if $p1 = 0.6$ the sample size required is

```
n=10
repeat{
  n=n+1
  if(pwr.binomial(n, p1=0.6)>80) break
}
cat(n, " ", round(pwr.binomial(n, p1=0.6)),"\n")
```

```
## 253   80
```

Notice that this problem is not symmetric in $p_0 - p_1$:

```
n=10
repeat{
  n=n+1
  if(pwr.binomial(n, p1=0.4)>80) break
}
cat(n, " ", round(pwr.binomial(n, p1=0.4)),"\n")
```

```
## 144   80
```

7.2 Difference in Proportions

7.2.0.1 Example (7.2.1) In a survey of 1000 likely voters 523 said they will vote for party A, the other 477 for party B. Find a 95% CI for the **lead** of one party over the other.

First we need a probability model for this experiment. Here this is clearly as follows: let $X_i = 1$ if vote is for A, 0 if it is for B, then $X_i \sim Ber(p)$. We can assume that $X_1, .., X_n$ are independent. The parameter of interest is the difference in proportions $\theta = p-(1-p) = 2p-1$.

7.2.1 Frequentist Analysis

we already know that the mle of p is $\bar{x}$, so the mle of θ is $2\bar{x} - 1$, here $2 \times 0.523 - 1 = 0.046$.

Of course if $\theta = 2p - 1$ we have $p = [1 + \theta]/2$. Let $y = \sum x_i$, then

$$
\begin{aligned}
&L(p|\boldsymbol{x})f(\boldsymbol{x}|p) = p^y(1-p)^{n-y}\\
&L(\theta|\boldsymbol{x}) = \left(\frac{1+\theta}{2}\right)^y \left(1 - \frac{1+\theta}{2}\right)^y = (1+\theta)^y(1-\theta)^{n-y}/2^n\\
&\lambda(\boldsymbol{x}) = \frac{L(\theta_0|\boldsymbol{x})}{L(\hat{\theta}|\boldsymbol{x})} = \frac{(1+\theta_0)^y(1-\theta_0)^{n-y}}{(1+[2\bar{x}-1]))^y(1-[2\bar{x}-1])^{n-y}}\\
&\frac{(1+\theta_0)^y(1-\theta_0)^{n-y}}{2^n(y/n)^y(1-y/n)^{n-y}}
\end{aligned}
$$

It is easy to show that $\lambda(\boldsymbol{x})$ is large if and only if $|y/n||$.

Now

$$
\begin{aligned}
&\frac{\bar{X}-p}{\sqrt{p(1-p)/n}} \sim N(0,1)\\
&\frac{1}{2}\frac{\hat{\theta}-\theta_0}{\sqrt{p(1-p)/n}} = \frac{1}{2}\frac{(2\bar{X}-1)-(2p-1)}{\sqrt{p(1-p)/n}} \sim N(0,1)\\
&=
\end{aligned}
$$

and so a $(1-\alpha)100\%$ confidence interval fot θ is given by

$$\left(2\bar{x}-1-2z_{\alpha/2}\sqrt{\bar{x}(1-\bar{x})/n}, 2\bar{x}-1+2z_{\alpha/2}\sqrt{\bar{x}(1-\bar{x})/n}\right)$$

Notice that the estimation error is twice the one for a Binomial p.

For our numbers we get the interval

```
x <- 523; n <- 1000; alpha <- 0.05
round(2*x/n-1 +
        c(-1, 1)*2*qnorm(1-alpha/2)*sqrt(x/n*(1-x/n)/n), 3)
```

```
## [1] -0.016  0.108
```

7.2.2 Bayesian Analysis

Our parameter is θ with values in [-1,1], so we need a prior with values in this interval. For p we usually use $Beta(\alpha, \beta)$, and then we have

$$
\begin{aligned}
&p|\boldsymbol{x} \sim Beta(\alpha+y, n-y+\beta)\\
&F_{\theta|\boldsymbol{x}}(t) = P(\theta < t|\boldsymbol{x}) =\\
&P(2p-1 < t|\boldsymbol{x}) =\\
&P\left(p < \frac{t+1}{2}|\boldsymbol{x}\right)
\end{aligned}
$$

and so we find the posterior distribution to be

$$
\begin{aligned}
&f_{\theta|\boldsymbol{x}}(t) = f_{p|\boldsymbol{x}}(\frac{t+1}{2})\frac{1}{2} =\\
&\frac{\Gamma(\alpha+\beta+n)}{\Gamma(\alpha+y)\Gamma(\beta-y+n)}\left(\frac{t+1}{2}\right)^{\alpha+y-1}\left(1-\frac{t+1}{2}\right)^{\beta-y+n-1}\frac{1}{2} =\\
&\frac{\Gamma(\alpha+\beta+n)}{\Gamma(\alpha+y)\Gamma(\beta-y+n)}(1+t)^{\alpha+y-1}(1-t)^{\beta-y+n-1}2^{\alpha+\beta+n+3}
\end{aligned}
$$

for $-1 < t < 1$.

For a credible interval we can use $\alpha/2$ on the left and right, and so we find

$$\alpha/2 = P(\theta < t|\boldsymbol{x}) = P\left(p < \frac{t+1}{2}|\boldsymbol{x}\right)$$

and so $l = 2qbeta(\alpha/2, \alpha+y, \beta-y+n)-1$ and $u = 2qbeta(1-\alpha/2, \alpha+y, \beta-y+n)-1$

```
round(2*qbeta(c(alpha/2, 1-alpha/2), 1+x, n-x+1)-1, 3)
```

```
## [1] -0.016  0.108
```

7.3 Equal Variance

7.3.0.1 Example (7.3.1) say we have samples $X_1,..,X_n$ iid $N(\mu_x, \sigma_x)$ and $Y_1,..,Y_m$ iid $N(\mu_y, \sigma_y)$ and we want to test

$$H_0 : \sigma_x = \sigma_y \text{ vs } H_1 : \sigma_x \neq \sigma_y$$

We will assume μ_x and μ_y are known, in which case we can assume $\mu_x = \mu_y = 0$.

Let's derive the likelihood ratio test. We will do this in terms of the variances $v_x = \sigma_x^2$ and $v_y = \sigma_y^2$, which is of course the same test.

First we find the joint density, using (2.3.14)

$$\begin{aligned}
&f(\boldsymbol{x}, \boldsymbol{y}|v_x, v_y) = \\
&(2\pi v_x)^{-n/2} \exp\left\{-\frac{1}{2v_x}\sum x_i^2\right\} (2\pi v_y)^{-m/2} \exp\left\{-\frac{1}{2v_y}\sum y_i^2\right\} = \\
&(2\pi v_x)^{-n/2} \exp\left\{-\frac{nt_x}{2v_x}\right\} (2\pi v_y)^{-m/2} \exp\left\{-\frac{mt_y}{2v_y}\right\}
\end{aligned}$$

where we define $t_x = \frac{1}{n}\sum x_i^2$ and $t_y = \frac{1}{m}\sum y_i^2$.

Now

$$\begin{aligned}
&l(v_x, v_y) = -\frac{n}{2}\log(2\pi v_x) - \frac{nt_x}{2v_x} - \frac{m}{2}\log(2\pi v_y) - \frac{nt_y}{2v_y} \\
&\frac{dl}{dv_x} = -\frac{n}{2v_x} + \frac{t_x}{2v_x^2} = 0
\end{aligned}$$

yields $\hat{v}_x = t_x$, and similarly $\hat{v}_y = t_y$

Under H_0 we have $v_x = v_y =: v$, and so

$$L(v, v) = (2\pi v)^{-n/2} \exp\left\{-\frac{nt_x}{2v}\right\} (2\pi v)^{-m/2} \exp\left\{-\frac{mt_y}{2v}\right\} = (2\pi v)^{-(n+m)/2} \exp\left\{-\frac{nt_x + mt_y}{2v}\right\}$$

$$l(v, v) = -\frac{n+m}{2}\log(2\pi v) - \frac{nt_x + mt_y}{2v}$$

and so $\hat{v} = \frac{nt_x+mt_y}{n+m}$.

$$\lambda(\boldsymbol{x},\boldsymbol{y}) = \frac{L(\hat{v},\hat{v})}{L(\hat{v_x},\hat{v_y})} =$$

$$\frac{(2\pi\hat{v})^{-(n+m)/2}\exp\left\{-\frac{nt_x+mt_y}{2\hat{v}}\right\}}{(2\pi\hat{v}_x)^{-n/2}\exp\left\{-\frac{nt_x}{2\hat{v}_x}\right\}(2\pi\hat{v}_y)^{-m/2}\exp\left\{-\frac{mt_y}{2\hat{v}_y}\right\}} =$$

$$\frac{(2\pi\frac{nt_x+mt_y}{n+m})^{-(n+m)/2}\exp\left\{-\frac{nt_x+mt_y}{2\frac{nt_x+mt_y}{n+m}}\right\}}{(2\pi t_x)^{-n/2}\exp\left\{-\frac{nt_x}{2t_x}\right\}(2\pi t_y)^{-m/2}\exp\left\{-\frac{mt_y}{2t_y}\right\}} =$$

$$\frac{(\frac{nt_x+mt_y}{n+m})^{-(n+m)/2}\exp\left\{-\frac{n+m}{2}\right\}}{(t_x)^{-n/2}\exp\left\{-\frac{n}{2}\right\}(t_y)^{-m/2}}\exp\left\{-\frac{m}{2}\right\} =$$

$$(n+m)^{(n+m)/2}\frac{(nt_x+mt_y)^{-(n+m)/2}}{(t_x)^{-n/2}(t_y)^{-m/2}} =$$

$$(n+m)^{(n+m)/2}\left(\frac{nt_x+mt_y}{t_x}\right)^{-n/2}\left(\frac{nt_x+mt_y}{t_y}\right)^{-m/2} =$$

$$(n+m)^{(n+m)/2}\left(n+m(t_y/t_x)\right)^{-n/2}\left(n(t_x/t_y)+m\right)^{-m/2} =$$

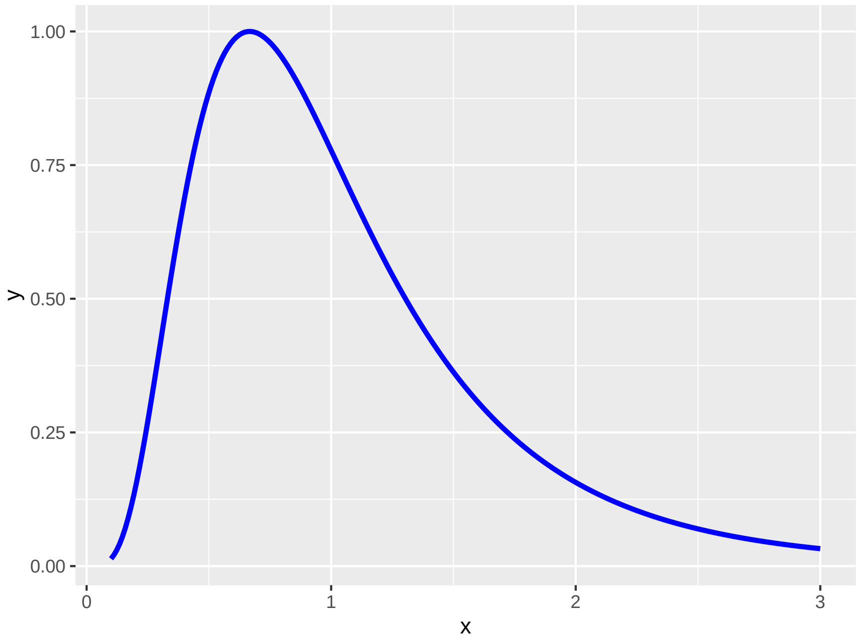
$$(n+m)^{(n+m)/2}\left(n+n(mt_y/nt_x)\right)^{-n/2}\left(m(nt_x/mt_y)+m\right)^{-m/2} =$$

$$\frac{(n+m)^{(n+m)/2}}{n^{n/2}m^{m/2}}\left(1+1/F)\right)^{-n/2}\left(1+F\right)^{-m/2}$$

where $F = (nt_x)/(mt_y)$

Now *LRT is small* is equivalent to *F is small or large*, as we can see here:

```r
n<-10; m<-15
fun <- function(x)
  (n+m)^((n+m)/2)/n^(n/2)/m^(m/2)*
   (1+1/x)^(-n/2)*(1+x)^(-m/2)
ggcurve(fun=fun, A=0.1, B=3)
```



and so under the null hypothesis

$$
\begin{aligned}
&X_i \sim N(0, \sigma)\\
&X_i/\sigma \sim N(0, 1)\\
&X_i^2/v \sim \chi^2(1) =\\
&nt_x/v = \sum_{i=1}^n X_i^2/v \sim \chi^2(n)\\
&F = (nt_x/v)/(mty/v) \sim F(n, m)
\end{aligned}
$$

and we reject the null if $F < qf(\alpha/2, n, m)$ or $F > qf(1-\alpha/2, n, m)$.

```
n <- 10; m <- 14
x <- rnorm(n, 0, 1)
y <- rnorm(m, 0, 1)
tx <- mean(x^2)
ty <- mean(y^2)
(n*tx)/(m*ty)
```

```
## [1] 1.131456
```

```
qf(c(0.025, 0.975), n, m)
```

```
## [1] 0.2816576 3.1468612
```

```
n <- 10; m <- 14
x <- rnorm(n, 0, 1)
y <- rnorm(m, 0, 3)
tx <- mean(x^2)
ty <- mean(y^2)
(n*tx)/(m*ty)
```

```
## [1] 0.2153921
```

7.4 Analysis of Variance (ANOVA)

7.4.0.1 Example (7.4.1) 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. Each baby was classified by the cocaine use of the mother: Free-no drugs of any kind, Trimester-mothers used cocain but stopped during the first trimester (three month of pregnancy and Throughout-mother used cocaine until birth.

Is there a statistically significant difference between the groups?

Source: Cocaine abuse during pregnancy: correlation between prenatal care and perinatal outcome Authors: SN MacGregor, LG Keith, JA Bachicha, and IJ Chasnoff Obstetrics & Gynecology 1989;74:882-885

```
kable.nice(mothers[c(1:2, 40:41, 80:81), ])
```

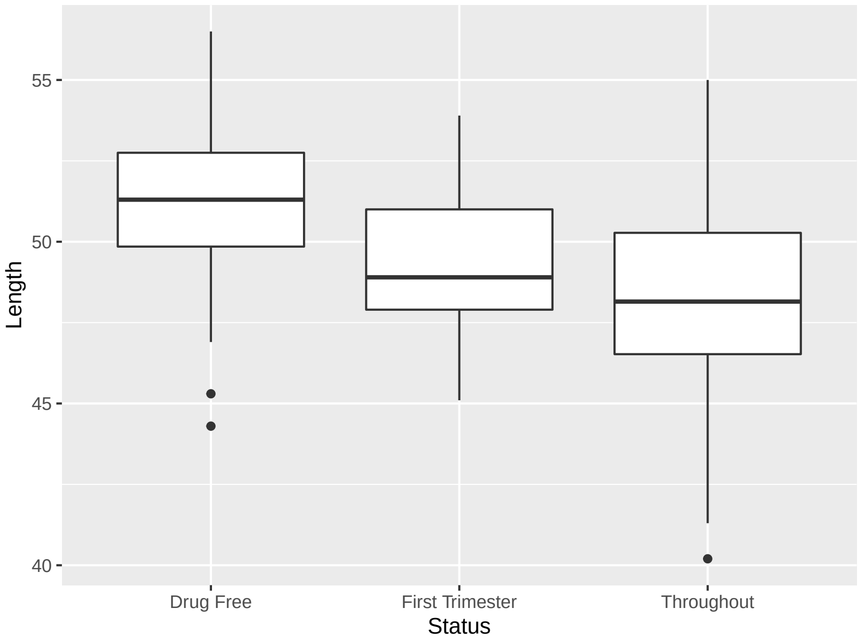
	Status	Length
1	Drug Free	44.3
2	Drug Free	45.3
40	First Trimester	45.1
41	First Trimester	45.7
80	Throughout	48.5
81	Throughout	49.0

What is a probability model here? In it's most general form it is as follows: we have observations

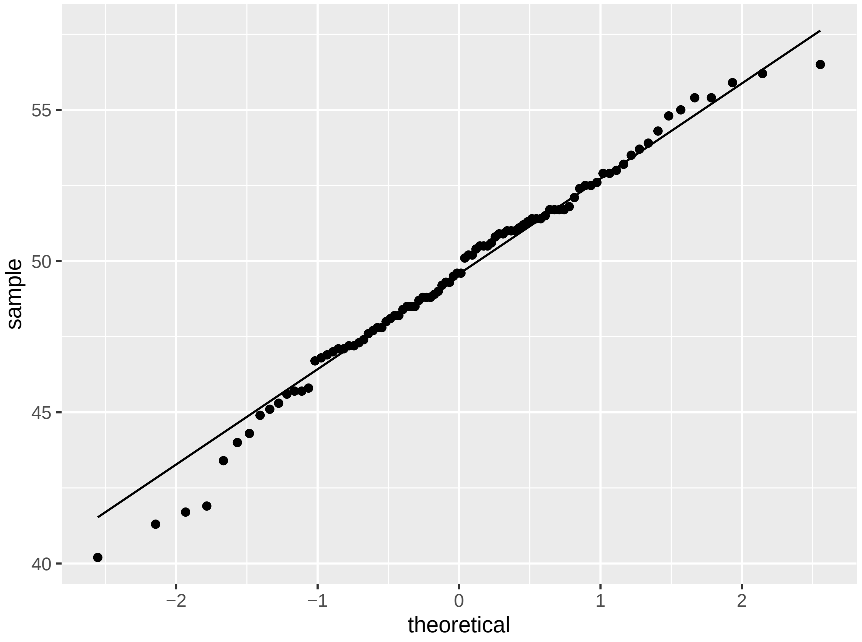
$$X_{ij} \sim F_i,\ i=1,..,k,\ j=1,..,n_i$$

that is each group has it's own distribution. A look at the boxplot and the normal probability plots makes it appear, though, as if each of the distributions were actually normal:

```
ggplot(mothers, aes(Status, Length)) +
  geom_boxplot()
```



```r
ggplot(data=mothers, aes(sample=Length)) +
            geom_qq() + geom_qq_line()
```



So we can write the probability model:

$$X_{ij} \sim N(\mu_i, \sigma_i),\ i = 1, .., k,\ j = 1, .., n_i$$

Especially the boxplot strongly suggests a further simplification of the model, namely that the standard deviations are the same, so we have

$$X_{ij} \sim N(\mu_i, \sigma),\ i = 1, .., k,\ j = 1, .., n_i$$

Standard ANOVA terminology would write this model as follows:

$$X_{ij} = \mu_i + \epsilon_{ij} \epsilon_{ij} \sim N(0, \sigma)$$

where the ϵ_{ij} are called the **residuals**.

The basic ANOVA test is then

$H_0 : \mu_1 = \mu_2 = \mu_3$ vs $H_a : \mu_i \neq \mu_j$ for some i and j.

7.4.1 Frequentist Solution

Let's derive the likelihood ratio test for this problem:

$$f(x_{11}, .., x_{kn_k}|\boldsymbol{\mu}) = \prod_{i=1}^{k} \prod_{j=1}^{n_i} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2} \left(x_{ij} - \mu_i\right)^2\right\} =$$
$$(2\pi\sigma^2)^{-n/2} \exp\left\{-\frac{1}{2\sigma^2} \sum_{i=1}^{k} \sum_{j=1}^{n_i} \left(x_{ij} - \mu_i\right)^2\right\}$$
$$l(\boldsymbol{\mu}|\boldsymbol{x}) = \frac{n}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{k} \sum_{j=1}^{n_i} \left(x_{ij} - \mu_i\right)^2$$
$$\frac{dl(\boldsymbol{\mu}|\boldsymbol{x})}{d\mu_i} = \frac{1}{\sigma^2} \sum_{j=1}^{n_i} (x_{ij} - \mu_i) = 0$$
$$\hat{\mu_i} = \frac{1}{n_i} \sum_{j=1}^{n_i} x_{ij} =: x_i.$$
$$\frac{dl(\boldsymbol{\mu}|\boldsymbol{x})}{d(\sigma^2)} = -\frac{n}{\sigma^2} + \frac{1}{2(\sigma^2)^2} \sum_{i=1}^{k} \sum_{j=1}^{n_i} \left(x_{ij} - \mu_i\right)^2 = 0$$
$$\widehat{\sigma^2} = \frac{1}{n} \sum_{i=1}^{k} \sum_{j=1}^{n_i} \left(x_{ij} - x_{i.}\right)^2$$

under the null hypothesis

$$H_0 : \mu_1 = ... = \mu_k = \mu$$

and so

$$
\begin{aligned}
&f(x_{11},..,x_{kn_k}|\mu) = L(\boldsymbol{\mu},\sigma^2|\boldsymbol{x})\\
&(2\pi\sigma^2)^{-n/2} = \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{k}\sum_{j=1}^{n_i}(x_{ij}-\mu)^2\right\}\\
&l(\mu|\boldsymbol{x}) = \frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{k}\sum_{j=1}^{n_i}(x_{ij}-\mu)^2\\
&\frac{dl(\mu|\boldsymbol{x})}{d\mu} = \frac{1}{\sigma^2}\sum_{i=1}^{k}\sum_{j=1}^{n_i}(x_{ij}-\mu) = 0\\
&\hat{\hat{\mu}} = \frac{1}{n}\sum_{i=1}^{k}\sum_{j=1}^{n_i}x_{ij} =: x_{..}\\
&\frac{dl(\boldsymbol{\mu}|\boldsymbol{x})}{d(\sigma^2)} = -\frac{n}{\sigma^2} + \frac{1}{2(\sigma^2)^2}\sum_{i=1}^{k}\sum_{j=1}^{n_i}(x_{ij}-\mu)^2 = 0\\
&\widehat{\widehat{\sigma^2}} = \frac{1}{n}\sum_{i=1}^{k}\sum_{j=1}^{n_i}(x_{ij}-x_{..})^2
\end{aligned}
$$

Now we find the likelihood ratio test statistic:

$$
\begin{aligned}
&\lambda(\boldsymbol{x}) = \frac{L(\widehat{\widehat{\mu}},\widehat{\widehat{\sigma^2}}|\boldsymbol{x})}{L(\hat{\boldsymbol{\mu}},\widehat{\sigma^2}|\boldsymbol{x})} =\\
&\frac{(2\pi\widehat{\widehat{\sigma^2}})^{-n/2}\exp\left\{-\frac{1}{2\widehat{\widehat{\sigma^2}}}\sum_{i=1}^{k}\sum_{j=1}^{n_i}(x_{ij}-x_{..})^2\right\}}{(2\pi\widehat{\sigma^2})^{-n/2}\exp\left\{-\frac{1}{2\widehat{\sigma^2}}\sum_{i=1}^{k}\sum_{j=1}^{n_i}(x_{ij}-x_{i.})^2\right\}} =\\
&\frac{(\widehat{\widehat{\sigma^2}})^{-n/2}\exp\left\{-\frac{1}{2\widehat{\widehat{\sigma^2}}}n\widehat{\widehat{\sigma^2}}\right\}}{(\widehat{\sigma^2})^{-n/2}\exp\left\{-\frac{1}{2\widehat{\sigma^2}}n\widehat{\sigma^2}\right\}} =\\
&\left(\frac{\widehat{\sigma^2}}{\widehat{\widehat{\sigma^2}}}\right)^{n/2}
\end{aligned}
$$

you can now see why this is called the analysis of variance although it really is a method concerned with means. Now as always

$$
\begin{aligned}
&\sum_{j=1}^{n_i} \left(x_{ij} - x_{i.}\right)^2 = \\
&\sum_{j=1}^{n_i} \left(x_{ij} - x_{..} + x_{..} - x_{i.}\right)^2 = \\
&\sum_{j=1}^{n_i} \left[\left(x_{ij} - x_{..}\right)^2 + 2\left(x_{ij} - x_{..}\right)\left(x_{..} - x_{i.}\right) + \left(x_{..} - x_{i.}\right)^2\right] = \\
&\sum_{j=1}^{n_i} \left(x_{ij} - x_{..}\right)^2 + 2\left(x_{..} - x_{i.}\right)\sum_{j=1}^{n_i}\left(x_{ij} - x_{..}\right) + n_i\left(x_{..} - x_{i.}\right)^2 = \\
&\sum_{j=1}^{n_i} \left(x_{ij} - x_{..}\right)^2 + 2\left(x_{..} - x_{i.}\right)\left(\sum_{j=1}^{n_i} x_{ij} - n_i x_{..}\right) + n_i\left(x_{..} - x_{i.}\right)^2 = \\
&\sum_{j=1}^{n_i} \left(x_{ij} - x_{..}\right)^2 + 2\left(x_{..} - x_{i.}\right)\left(n_i x_{i.} - n_i x_{..}\right) + n_i\left(x_{..} - x_{i.}\right)^2 = \\
&\sum_{j=1}^{n_i} \left(x_{ij} - x_{..}\right)^2 - 2n_i\left(x_{..} - x_{i.}\right)^2 + n_i\left(x_{..} - x_{i.}\right)^2 = \\
&\sum_{j=1}^{n_i} \left(x_{ij} - x_{..}\right)^2 - n_i\left(x_{..} - x_{i.}\right)^2
\end{aligned}
$$

summing over k yields

$$
\widehat{\sigma^2} = \widehat{\widehat{\sigma^2}} - \frac{1}{n}\sum_{i=1}^{k} n_i\left(x_{i.} - x_{..}\right)^2
$$

and so

$$
\begin{aligned}
\lambda(\boldsymbol{x}) = \left(\frac{\widehat{\sigma^2}}{\widehat{\widehat{\sigma^2}}}\right)^{n/2} &= \\
\left(\frac{\widehat{\sigma^2}}{\widehat{\sigma^2} + \frac{1}{n}\sum_{i=1}^{k} n_i\left(x_{i.} - x_{..}\right)^2}\right)^{n/2} &= \\
\left(\frac{1}{1 + \frac{\frac{1}{n}\sum_{i=1}^{k} n_i\left(x_{i.} - x_{..}\right)^2}{\widehat{\sigma^2}}}\right)^{n/2} &= \\
\left(\frac{1}{1 + \frac{\sum_{i=1}^{k} n_i\left(x_{i.} - x_{..}\right)^2}{\sum_{i=1}^{k}\sum_{j=1}^{n_i}\left(x_{ij} - x_{i.}\right)^2}}\right)^{n/2}
\end{aligned}
$$

and it can be shown that $\lambda(\boldsymbol{x})$ is large if and only if

$$\frac{\sum_{i=1}^{k} n_i \left(x_{i.} - x_{..}\right)^2}{\sum_{i=1}^{k} \sum_{j=1}^{n_i} \left(x_{ij} - x_{i.}\right)^2}$$

is large.

In the numerator we have an estimate of the variance **between** the group means and the overall mean, and in the denominator an estimate of the variance **within** the groups. It is easy to show that this test statistic (when properly scaled) has an F distribution with k-1 and n-k degrees of freedom.

Let's find all the relevant numbers for our data set:

```r
x.. <- mean(mothers$Length)
sum((mothers$Length-x..)^2)
```

```
## [1] 1066.955
```

```r
round(x.., 2)
```

```
## [1] 49.55
```

```r
n <- tapply(mothers$Length, mothers$Status, length)
n
```

```
##       Drug Free First Trimester      Throughout
##              39              19              36
```

```r
x. <- tapply(mothers$Length, mothers$Status, mean)
round(x., 2)
```

```
##       Drug Free First Trimester      Throughout
##            51.1            49.3            48.0
```

```r
num <- sum(n*(x.-x..)^2)
num
```

```
## [1] 181.3749
```

```r
residuals <- 0*mothers$Length
st <- unique(mothers$Status)
for(i in 1:3)
  residuals[mothers$Status==st[i]] <-
  mothers$Length[mothers$Status==st[i]]-x.[i]
denom <- sum(residuals^2)
denom
```

```
## [1] 885.58
```

```r
mean.square <- c(num/2, denom/91)
round(mean.square, 2)
```

```
## [1] 90.69  9.73
```

```r
Fs <- mean.square[1]/mean.square[2]
round(Fs, 3)
```

```
## [1] 9.319
```

```r
1-pf(Fs, 2, 91)
```

```
## [1] 0.0002080747
```

now the information is usually summarized in an **ANOVA** table:

```r
df <- data.frame(DF=c(2, 9),
       SS=c(181.375, 885.58),
       Mean=c(90.69, 9.73),
       F=c(9.318, ""),
       p.value=c(0.002, ""))
rownames(df) <- c("Status", "Residuals")
kable.nice(df)
```

	DF	SS	Mean	F	p.value
Status	2	181.375	90.69	9.318	0.002
Residuals	9	885.580	9.73		

or of course we can use R:

```r
summary(aov(Length~Status, data=mothers))
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## Status       2  181.4   90.69   9.319 0.000208
## Residuals   91  885.6    9.73
```

The basic F test is rarely of great interest, ANOVA becomes more interesting when we test more specific hypotheses.

A **contrast** is an expression of the form

$\sum a_i\mu_i$

where $(a_1, .., a_k)$ is such that $\sum a_i = 0$.

With this notation we can write many interesting hypotheses:

- basic F test:

$H_0 : \sum a_i\mu_i = 0$

for all $(a_1, .., a_k)$ is such that $\sum a_i = 0$

- multiple comparison test:

$H_0 : \sum a_i\mu_i = 0$ for (1,-1,0) (1,0,-1) and (0,1,-1)

- a test of interest in a specific situation:

$H_0 : \sum a_i\mu_i = 0$ for (1/2,1/2,1)

tests whether

$(\mu_1 + \mu_2)/2 = \mu_3$

To test $H_0 : \sum a_i\mu_i = 0$ vs $H_a : \sum a_i\mu_i \neq 0$ use the test statistic

$$T = \frac{|\sum_{i=1}^k a_ix_i}{s_p\sqrt{\sum_{i=1}^k a_i^2/n_i}}$$

where

$$s_p^2 = \frac{1}{n-k}\sum_i\sum_j(x_{ij} - x_{i.})^2$$

then $T \sim t(n-k)$

7.4.1.1 Example (7.4.2) We have

$$H_0 : \sum a_i\mu_i = 0 \text{ vs } H_a : \sum a_i\mu_i \neq 0$$

for a=(0,1,-1)

so

```
a <- c(0, 1, -1)
Ts <- abs(sum(x.*a))/sqrt(mean.square[2]*(1/n[2]+1/n[3]))
out <- round(c(Ts, 1-pt(Ts, sum(n)-2)), 3)
names(out) <- c("T", "p-value")
out
```

```
##       T p-value
##   1.470   0.073
```

One important point to remember here is that the classical ANOVA method is just a straightforward application of the likelihood ratio test.

7.4.2 Bayesian Inference

This will of course always depend on the priors on $(\mu_1, .., \mu_k, \sigma)$. If we choose noninfomative priors, for example proportional to $1/\sigma^2$, then we recover essentially the classical ANOVA above.

7.5 Two-way Tables

7.5.1 Categorical Data

7.5.1.1 Example (7.5.1) Psychological and social factors can influence the survival of patients with serious diseases. One study examined the relationship between survival of patients with coronary heart disease and pet ownership. Each of 92 patients was classified as having a pet or not, and whether they survived one year.

Here is the data, from Erika Friedmann et al., "Animal companions and one-year survival of patients after discharge from a coronary care unit.":

	Status	Alive	Dead
1	Owns a Pet	50	3
2	Does not own a Pet	28	11

Question: is there a statistically significant relationship (association) between Ownership and Survival?

What is an appropriate probability model here? For each patient in the population there are four possibilities: owns a pet-alive, owns a pet-dead, does not own a pet-alive, does not own a pet-dead. We can model this using a multinomial distribution: (X,Y) takes values (1,1), (1,2), (2,1), (2,2) with $P((X,Y) = (i,j)) = p_{ij}$. Of course we have

$0 \le p_{ij} \le 1$ and $\sum_{ij} p_{ij} = 1$.

The math that follows get's a little easier if we reparametrize the problem as follows: a discrete random vector with finitely many values is always equivalent to a multinomial distribution. So let Z be a rv with values 1-4 and probabilities $p_1, .., p_4$.

7.5.2 Frequentist Analysis

Let's begin by finding the mle's of the p_i's. Let $z_i = \sum I[Z = i]$, then

$$f(\boldsymbol{z}|\boldsymbol{p}) = \prod p_i^{z_i}$$
$$l(\boldsymbol{p}) = \sum_i z_i \log p_i$$

Now we need to be careful because we need to maximize this function with the additional condition $\sum p_i = 1$ (otherwise the maxima is at infinity anyway), so we need to use **Lagrange multipliers:**

$$h(\boldsymbol{p}) = \sum_i z_i \log p_i - \lambda(\sum_i p_i - 1)$$
$$\frac{dh}{dp_i} = \frac{z_i}{p_i} - \lambda = 0$$
$$z_i = \lambda p_i$$
$$n = \sum z_i = (\sum p_i)\lambda = \lambda$$

and so we find $\hat{p}_i = \frac{z_i}{p_i}$.

What does our question mean in terms of the p_i's? If there is no relationship between Ownership and Survival then X and Y are independent and we should have

$$P((X,Y) = (i,j)) = P(X = i)P(Y = j)$$

for i,j=1,2, Or

$p_1 = (p_1 + p_2)(p_1 + p_3)$

$p_2 = (p_1 + p_2)(p_2 + p_4)$

$p_3 = (p_1 + p_3)(p_3 + p_4)$

$p_4 = (p_2 + p_4)(p_3 + p_4)$

It's easy to see why if you think in terms of marginals:

	Status	Alive	Dead	Total
1	Owns a Pet	p1	p3	p1+p3
2	Does not own a Pet	p2	p4	p2+p4
3	Total	p1+p2	p3+p4	

So let's do the LRT test for this problem:

H_0: X independent of Y equivalent to H_0: above equations hold

we already found the mle's, so now we need to find the numerator. First note that:

$$p_1 = (p_1 + p_2)(p_1 + p_3) =$$
$$p_1^2 + p_1(p_2 + p_3) + p_2 p_3 =$$
$$p_1^2 + p_1(1 - p_1 + p_4) + p_2 p_3 =$$
$$p_1 - p_1 p_4 + p_2 p_3$$

so we find

$$p_1p_4 - p_2p_3 = 0$$

In the same way we can verify that the other equations also lead to this one. So we need to maximize

$$\max\left\{\sum z_i \log(p_i)|p_i \geq 0; \sum p_i = 1; p_1p_4 - p_2p_3 = 0\right\}$$

Again we use Lagrange multipliers:

$$\begin{aligned}
h(\boldsymbol{p}) &= \sum_i z_i \log p_i - \lambda_1(\sum_i p_i - 1) + \lambda_2(p_1p_4 - p_2p_3)\\
\frac{dh}{dp_1} &= \frac{z_i}{p_i} - \lambda_1 + \lambda_2 p_4 = 0\\
\frac{dh}{dp_2} &= \frac{z_i}{p_i} - \lambda_1 - \lambda_2 p_3 = 0\\
\frac{dh}{dp_3} &= \frac{z_i}{p_i} - \lambda_1 - \lambda_2 p_2 = 0\\
\frac{dh}{dp_4} &= \frac{z_i}{p_i} - \lambda_1 + \lambda_2 p_1 = 0
\end{aligned}$$

which has the solution

$$\begin{aligned}
\hat{\hat{p}}_1 &= \frac{z_1+z_2}{n}\frac{z_1+z_3}{n}\\
\hat{\hat{p}}_2 &= \frac{z_2+z_1}{n}\frac{z_2+z_4}{n}\\
\hat{\hat{p}}_3 &= \frac{z_3+z_1}{n}\frac{z_3+z_4}{n}\\
\hat{\hat{p}}_4 &= \frac{z_4+z_2}{n}\frac{z_4+z_3}{n}
\end{aligned}$$

Now let $E_1 = (z_1 + z_2)(z_1 + z_3)/n$, and so on, then

$$
\begin{aligned}
&\lambda(\boldsymbol{x}) = \prod \frac{(E_i/n)^{z_i}}{(z_i/n)^{z_i}} \\
&-2\log\lambda(\boldsymbol{x}) = -2\sum z_i \log\frac{E_i}{z_i} = \\
&-2\sum z_i \left(\log 1 + (\frac{E_i}{z_i} - 1)\right) = \\
=& \\
&-2\sum z_i \left[(\frac{E_i}{z_i} - 1) - \frac{1}{2}(\frac{E_i}{z_i} - 1)^2\right] = \\
&-2\sum \left[E_i - z_i - \frac{1}{2}\frac{(E_i - z_i)^2}{z_i}\right] \approx \\
&\sum \frac{(z_i - E_i)^2}{E_i}
\end{aligned}
$$

because $E_i \approx z_i$.

which shows how one eventually ends up with the famous chisquare statistic:

$$X^2 = \sum \frac{(O-E)^2}{E}$$

for our data we have

```
O <- c(50, 28, 3, 11)
n <- sum(O)
E <- c((O[1]+O[2])*(O[1]+O[3]),
       (O[1]+O[2])*(O[2]+O[4]),
       (O[3]+O[4])*(O[1]+O[3]),
       (O[3]+O[4])*(O[2]+O[4]))/n
```

	Status	Alive	Dead
1	Owns a Pet	50 (44.9)	3 (5.9)
2	Does not own a Pet	28 (33.1)	11 (8.1)

```
chi2 <- sum((O-E)^2/E)
round(c(chi2, 1-pchisq(chi2, 1)), 3)
```

```
## [1] 8.851 0.003
```

this has a chisquare distribution with 1(=(r-1)(c-1)) df.

So again we see that one of the famous methods in Statistics can be derived from the likelihood ratio test (plus some extra approximations).

We have done this for a 2x2 table, but the generalization to an RxC table is straight forward.

7.5.3 Bayesian Analysis

as always this starts with a prior. If we again use the parametrization $(p_1, .., p_4)$ then $Z = (z_1, .., z_4)$ has a multinomial distribution $(n, p_1, .., p_4)$ where n is assumed to be known.

A conjugate prior for the multinomial is the *Dirichlet distribution* with density

$$\pi(p) \propto \prod p_i^{\alpha_i - 1}$$

and then $\pi(p|z) \sim D(n, \alpha_1 + z_1, .., \alpha_4 + z_4)$.

The choice of $\alpha_1 = .. = \alpha_4 = 1$ is a non-informative prior ($p_i = 1/k$). The null hypothesis of independence then means independence of the posterior distribution, same as above. Indeed, under the non-informative prior we could again recover the chisquare test.

7.6 Ordinary Linear Regression

7.6.0.1 Example (7.6.1) Consider the the data set **hubble**. In 1929 Edwin Hubble published a paper showing a relationship between the distance and radial velocity away from Earth of “extra-galactic nebulae” (galaxies). His findings revolutionized astronomy. The “Hubble constant,” the slope of the regression of velocity (Y) on distance (X), is still a subject of research and debate. The data here are those Hubble published in his original paper.

Question: If it is true there is a linear relationship between Velocity (Y) and Distance (X), what is the slope of the line?

If there is a linear relationship, there exist β_0 and β_1 such that

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

i=1,..,n

where the ϵ_i are (again) called the **residuals**.

In the problem above the main task is to find a interval estimate for β_1. In other problems it might be to estimate Y for a specific value of x, to estimate E[Y] for some x, to see whether β_0 or β_1 are zero (or some other value) etc.

Another version of the regression problem is

$$Y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

i=1,..,n

that is the x’s are not random but fixed. For example the income (y) and the number of years of service (x) of randomly selected employees in a company. In practice these two versions are usually treated the same way.

First we need a probability model. Again this will depend on the problem, but one often used is to assume that (X,Y) are bivariate normal with parameters

$(\mu_x,\mu_y,\sigma_x,\sigma_y,\rho)$

If we think in terms of predicting Y from a fixed value of X=x we need the conditional distribution of Y|X=x, which is

$$Y|X = x \sim N(\mu_y + \rho\sigma_y/\sigma_x(x - \mu_x), \sigma_y\sqrt{1-\rho^2})$$

Therefore we have

$$E[Y|X = x] = \mu_y + \rho\sigma_y/\sigma_x(x - \mu_x) = \mu_y - \mu_x\rho\sigma_y/\sigma_x + \rho\sigma_y/\sigma_x x$$

so we find that under this probability model we have a natural linear relationship between X and Y with

$$\beta_0 = \mu_y - \mu_x\rho\sigma_y/\sigma_x$$

and

$$\beta_1 = \rho\sigma_y/\sigma_x$$

Generally in a regression context the analysis is carried out using the conditional distribution of $(Y_1, .., Y_n)$ given $X_1 = x_1, .., X_n = x_n$, in which case we can consider the x's as fixed and known. The probability model then becomes

$$Y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

$\epsilon_i \sim N(0, \sigma), i = 1, .., n$

Notice that we are assuming equal variance. If this is not reasonable, the analysis is still possible but somewhat more difficult.

In linear regression it is common to use the method of *least squares* for estimation, that is to find $\hat{\beta_0}$ and $\hat{\beta_1}$ that minimize

$$\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2$$

Instead let's find the mle's of β_0, β_1 and σ:

$$
\begin{aligned}
&f(\boldsymbol{x}|\beta_0\beta_1,\sigma) = (2\pi\sigma^2)^{-n/2}\exp\left\{-\frac{1}{2\sigma^2}\sum]\left[y_i-\beta_0-\beta_1x_i\right]^2\right\}\\
&l(\beta_0\beta_1,\sigma) = \frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum\left[y_i-\beta_0-\beta_1x_i\right]^2\\
&\frac{dl}{d\beta_0} = -\frac{1}{\sigma^2}\sum\left[y_i-\beta_0-\beta_1x_i\right] = -\frac{1}{\sigma^2}\left[\sum y_i - n\beta_0 - \beta_1\sum x_i\right] = 0\\
&n\hat\beta_0 = \sum y_i - \beta_1\sum x_i
\end{aligned}
$$

$$
\begin{aligned}
&\frac{dl}{d\beta_1} = -\frac{1}{\sigma^2}\sum\left[y_i-\beta_0-\beta_1x_i\right]x_i = -\frac{1}{\sigma^2}\left[\sum x_iy_i - \beta_0\sum x_i - \beta_1\sum x_i^2\right] = 0\\
&\beta_0\sum x_i = \sum x_iy_i - \beta_1\sum x_i^2
\end{aligned}
$$

and this system of equations has the solution

$$
\hat\beta_1 = \frac{\sum x_iy_i - (\sum x_i)(\sum y_i)/n}{\sum x_i^2 - (\sum x_i)/n}\hat\beta_0 = \bar y - \hat\beta_1\bar x
$$

which are of course the standard least squares regression estimates!

For σ^2 we find

$$
\begin{aligned}
&\frac{dl}{d\sigma^2} = -\frac{n}{}\sigma^2 + \frac{1}{2(\sigma^2)^2}\sum\left[y_i-\beta_0-\beta_1x_i\right]^2 = 0\\
&\widehat{\sigma^2} = \frac{1}{n}\sum\left[y_i-\hat\beta_0-\hat\beta_1x_i\right]^2
\end{aligned}
$$

What are the sampling distributions? First we can write

$$
\hat\beta_0 = \sum\left[\frac{1}{n} - \frac{x_i-\bar x}{\sum x_i^2 - (\sum x_i)/n}\right]y_i
$$

and so β_0 is a linear combination of normal rv's, and therefore normal itself. Moreover

$$
\begin{aligned}
&E[\hat\beta_0] = \sum\left[\frac{1}{n} - \frac{x_i-\bar x}{\sum x_i^2 - (\sum x_i)/n}\right]E[y_i] =\\
&\sum\left[\frac{1}{n} - \frac{x_i-\bar x}{\sum x_i^2 - (\sum x_i)/n}\right](\beta_0+\beta_1x_i) = \beta_0\\
&var(\hat\beta_0) = \sigma^2\frac{\sum x_i^2}{n(\sum x_i^2 - (\sum x_i)/n)}
\end{aligned}
$$

similarly we can show that

$$
\hat\beta_1 \sim N(\beta_1, \sigma/\sqrt{\sum x_i^2 - (\sum x_i)/n})(n-2)\widehat{s^2}/\sigma^2 \sim \chi^2(n-2)
$$

A confidence interval for the slope can be found as follows:

$$\hat{\beta}_1 \pm qt(1-\alpha/2, n-1)\sqrt{\frac{\widehat{\sigma^2}}{\sum x_i^2 - (\sum x_i)/n}}$$

```
alpha <- 0.05
x <- hubble$Distance
y <- hubble$Velocity
n <- length(x)
xbar <- mean(x)
ybar <- mean(y)
Sxx <- sum(x^2) - sum(x)^2/n
Sxy <- sum(x*y) - sum(x)*sum(y)/n
beta1 <- Sxy/Sxx
beta0 <- ybar - beta1*xbar
fits <- beta0 + beta1*x
e <- y - fits
sse <- sum(e^2)/(n-2)
round(beta1 + c(-1, 1)*qt(1-alpha/2, n-2)*sqrt(sse/Sxx), 1)
```

```
## [1] 298.1 610.2
```

or with R function *lm*:

```
fit <- lm(Velocity~Distance, data=hubble)
summary(fit)
```

```
##
## Call:
## lm(formula = Velocity ~ Distance, data = hubble)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -397.96 -158.10  -13.16  148.09  506.63
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -40.78      83.44  -0.489     0.63
## Distance      454.16      75.24   6.036 4.48e-06
##
## Residual standard error: 232.9 on 22 degrees of freedom
## Multiple R-squared:  0.6235, Adjusted R-squared:  0.6064
## F-statistic: 36.44 on 1 and 22 DF,  p-value: 4.477e-06
```

```r
round(confint(fit)[2, ], 1)
```

```
##  2.5 % 97.5 %
##  298.1  610.2
```

7.6.0.2 Example (7.6.2) Say $x_1, .., x_n$ are fixed numbers and $Y_i = \alpha + \beta x_i + \epsilon_i$, where $\epsilon_i \sim N(0, \sigma)$. We want to predict $y = \alpha + \beta x$ for some x. Specifically we want to find a $(1-\alpha)100\%$ confidence interval for $Y = \alpha + \beta x + \epsilon$.

Using least squares we find the estimators of α and β

$$
\begin{aligned}
&S_{xy} = \sum (x - \bar{x})(y - \bar{y}) \\
&\hat{\alpha} = S_{xy}/S_{xx} \\
&\hat{\beta} = \bar{y} - \hat{\alpha}\bar{x}
\end{aligned}
$$

From here one can show that for a fixed x (not necessarily one of the x_i's) a $(1-\alpha)100\%$ confidence the interval for $\mu_x = E[Y] = E[\alpha + \beta x]$ is given by

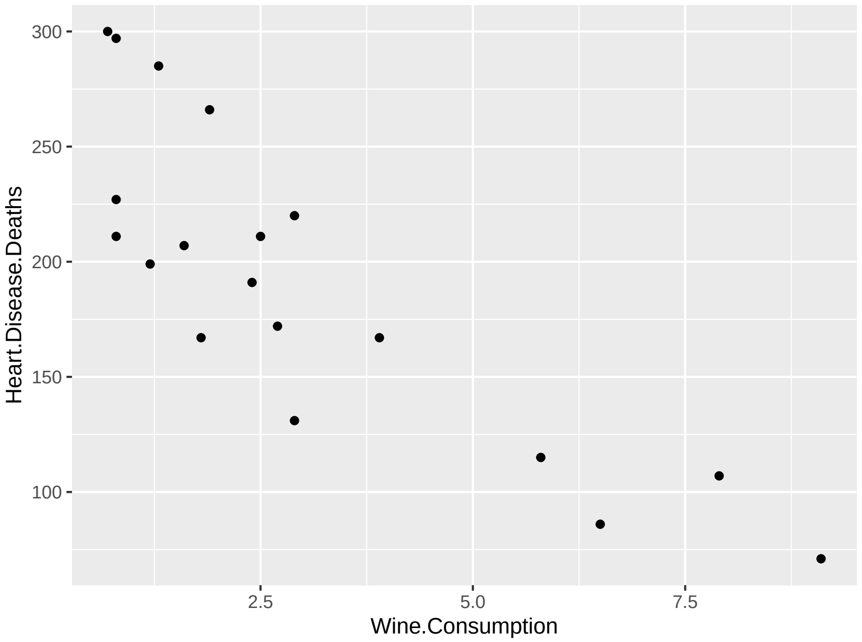
$$\hat{\alpha} + \hat{\beta}x \pm t_{n-1,\alpha/2}\hat{\sigma}\sqrt{\frac{1}{n} + \frac{(x-\bar{x})^2}{S_{xx}}}$$

where $\hat{\sigma}^2 = \frac{1}{n-2}\sum\left(y_i - \hat{\alpha} - \hat{\beta}x\right)$. So this is a confidence interval for the *mean response* at some given x. If we want a confidence interval for an *individual response* we have to use

$$\hat{\alpha} + \hat{\beta}x \pm t_{n-1,\alpha/2}\hat{\sigma}\sqrt{1 + \frac{1}{n} + \frac{(x-\bar{x})^2}{S_{xx}}}$$

7.6.0.3 Example (7.6.3) Consider the *wine* data set from Resma3:

```r
ggplot(data=wine, aes(Wine.Consumption, Heart.Disease.Deaths)) +
  geom_point()
```



Say we want a 90% confidence interval for the heart disease rate of a country with a wine consumption of 5 liters.

```
xnew=5
x=wine$Wine.Consumption
y=wine$Heart.Disease.Deaths
n=length(x)
sxx=sum((x-mean(x))^2)
syy=sum((y-mean(y))^2)
sxy=sum((x-mean(x))*(y-mean(y)))
alphahat=sxy/sxx
betahat=mean(y)-alphahat*mean(x)
yhat=betahat+alphahat*x
sigmahat=sqrt(sum((y-yhat)^2)/(n-2))
round(c(n, betahat, alphahat, sigmahat), 2)
```

```
## [1]  19.00 260.56 -22.97  37.88
```

```
round(betahat+alphahat*xnew+c(-1,1)*qt(1-alpha/2,n-1)*sigmahat*sqrt(1+1/n+(xnew-mean(x)
```

```
## [1]  62.8 228.7
```

or using R:

```
fit=lm(y~x)
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -62.95 -25.91 -12.35  26.97  55.52
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  260.563     13.835  18.833 7.97e-13
## x            -22.969      3.557  -6.457 5.91e-06
##
## Residual standard error: 37.88 on 17 degrees of freedom
## Multiple R-squared:  0.7103, Adjusted R-squared:  0.6933
## F-statistic: 41.69 on 1 and 17 DF,  p-value: 5.913e-06
```

```
predict(fit,newdata=data.frame(x=5), se.fit=TRUE, interval="prediction")
```

```
## $fit
##        fit      lwr      upr
## 1 145.7195 62.39922 229.0399
##
## $se.fit
## [1] 11.17192
##
## $df
## [1] 17
##
## $residual.scale
## [1] 37.87858
```

7.7 Goodness of Fit Tests

7.7.1 Chisquare Goodness-of-fit Test

7.7.1.1 Example (7.7.1) Experiments in Plant Hybridization (1865) by Gregor Mendel is one of the most famous papers in all of Science. His theory of genetics predicted that the number of smooth yellow, wrinkled yellow, smooth green and wrinkled green peas would be in the proportions 9:3:3:1. In one of his experiments he observed 315, 101, 108 and 32. Does this agree with his theory?

Again we use the likelihood ratio test. As with the test for independence we have a multinomial distribution Z with parameters (n, p_1,..,p_k) and we assume n is known. The mles are the same as before, $z_{i/n}$.

Under the null hypothesis the proportions should be 9:3:3:1, so

$$H_0 : p = (9/16, 3/16, 3/16, 1/16)$$

in this application the null hypothesis fixes the probabilities completely. Again we can do the Taylor approximation to -2log(LRT) and we find the chisquare statistic

$$X^2 = \sum_{i=1}^k \frac{(O-E)^2}{E}$$

which has a chisquare distribution with 3 (4-1) degrees of freedom (-1 because of $\sum p_i = 1$)

For Mendels' data we find

```
O <- c(315, 101, 108, 32)
E <- sum(O)*c(9/16, 3/16, 3/16, 1/16)
chi2 <- sum((O-E)^2/E)
round(c(chi2, 1-pchisq(chi2, 3)), 2)
```

```
## [1] 0.47 0.93
```

The chisquare statistic was already known in the mid 19th century but its distribution was derived by Karl Pearson in 1900. His argument was as follows:

- O is the sum of indicator random variables (X_i is of type i or not), so O has a binomial distribution
- if n is large enough $(O-E)/\sqrt{E} \approx N(0,1)$
- therefore $(O-E)^2/E \approx \chi^2(1)$
- finally $\sum_1^n (O-E)^2/E \sim \chi^2(n-1)$ because there is one restriction, namely $\sum O = n$.

We have also seen in the section on the large sample theory of LRT's that the chisquare statistic is asymptotically equivalent to the likelihood ratio test statistic.

Often in Statistics we assume that the data was generated by a specific distribution, for example the normal. If we are not sure that such an assumption is justified we would like to test for this.

7.7.1.2 Example (7.7.2) Say we have X_1, .., X_n iid F, and we wish to test

$$H_0 : F = N(0,1)$$

First notice that here the alternative hypothesis is

$$H_0 : F \neq N(0,1)$$

or even simply left out. Either way it is a HUGE set, made up of all possible distributions other than N(0,1). This makes assessing the power of a test very difficult.

7.7.1.3 Example (7.7.3)

Another famous data set in statistics is the number of deaths from horsekicks in the Prussian army from 1875-1894:

```
kable.nice(head(horsekicks))
```

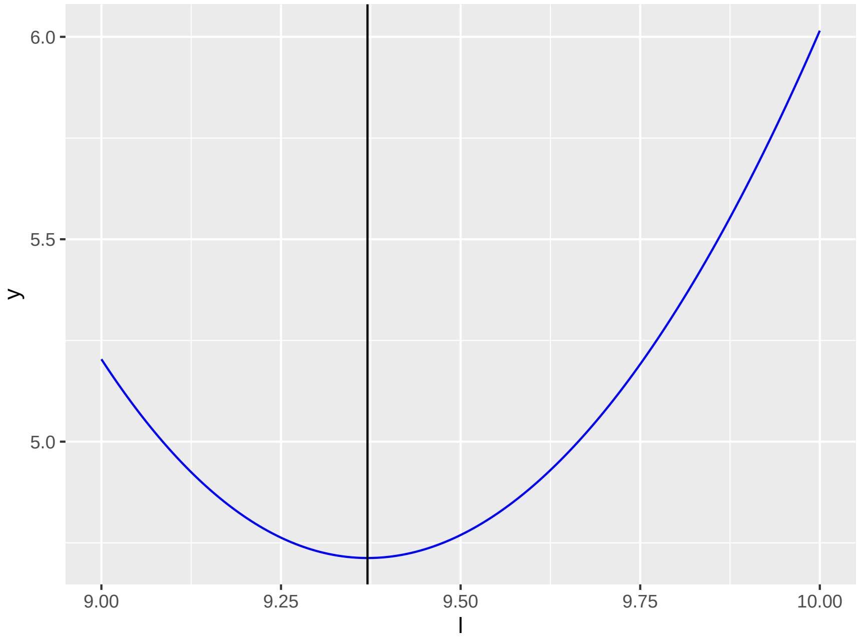
	Year	Deaths
1	1875	3
2	1876	5
3	1877	7
4	1878	9
5	1879	10
6	1880	18

It has been hypothesized that this data follows a Poisson distribution. Let's carry out a hypothesis test for this.

First of a Poisson distribution has a parameter, λ . Clearly even if the assumption of a Poisson distribution is correct it will be correct only for some values of λ . We reject the null if the the chi-square statistic is small, so if we reject it for the value of λ that minimizes the chi-square statistic, we would also reject it for any other value of λ .

The chisquare goodness-of-fit test is a large-sample test, it has the assumption that none of the expected numbers be to small. We deal with this by combining some categories. We will consider the cases 0-6, 7-9, 10-12 and Over 12. Then

```
cells <- c(0:6, 7:9, 10:12, 13:100)
O <- c(6, 4, 5, 5)
chi2 <- function(l) {
  y <- 0*l
  p <- rep(0, 4)
  for(i in seq_along(l)) {
    p[1] <- sum(dpois(0:6, l[i]))
    p[2] <- sum(dpois(7:9, l[i]))
    p[3] <- sum(dpois(10:12, l[i]))
    p[4] <- sum(dpois(13:100, l[i]))
    E <- sum(O)*p
    y[i] <- sum((O-E)^2/E)
  }
  y
}
l <- seq(9, 10, length=1000)
y <- chi2(l)
lhat <- l[which.min(y)]
ggplot(data.frame(l=l, y=y), aes(l, y)) +
  geom_line(color="blue") +
  geom_vline(xintercept = lhat)
```



```
lhat
```

```
## [1] 9.37037
```

```
round(c(chi2(lhat), 1-pchisq(chi2(lhat), 2)), 3)
```

```
## [1] 4.713 0.095
```

Under the null hypothesis the χ^2 statistic has a χ^2 distribution with m-k-1 degrees of freedom, where m is the number of classes and k is the number of parameters estimated from the data. So here we have m-k-1 = 4-1-1 = 2 d.f.

In the binning we have used, some E are a bit small. We could of course bin even further, but then we also loose even more information.

Notice that here we used an unusual estimation method, called *minimum chi-square*. Often in practice people use maximum likelihood, this however is wrong!

The adjustment of the degrees of freedom for the number of estimated parameters has an interesting history. It does not appear in Pearson's original derivation. In fact, following Pearson's logic there should be no need for this adjustment, because if the sample size is large enough any parameter should be estimated with sufficiently high precision. The need for the adjustment was recognized only 20 years after the original publication of Pearson by none other than Karl Fisher and is now sometimes called the Fisher-Pearson statistic.

7.7.1.4 Example (7.7.4)

Let's study this question for a bit. Say we want to test

$$H_0 : F = Bin(m, p)$$

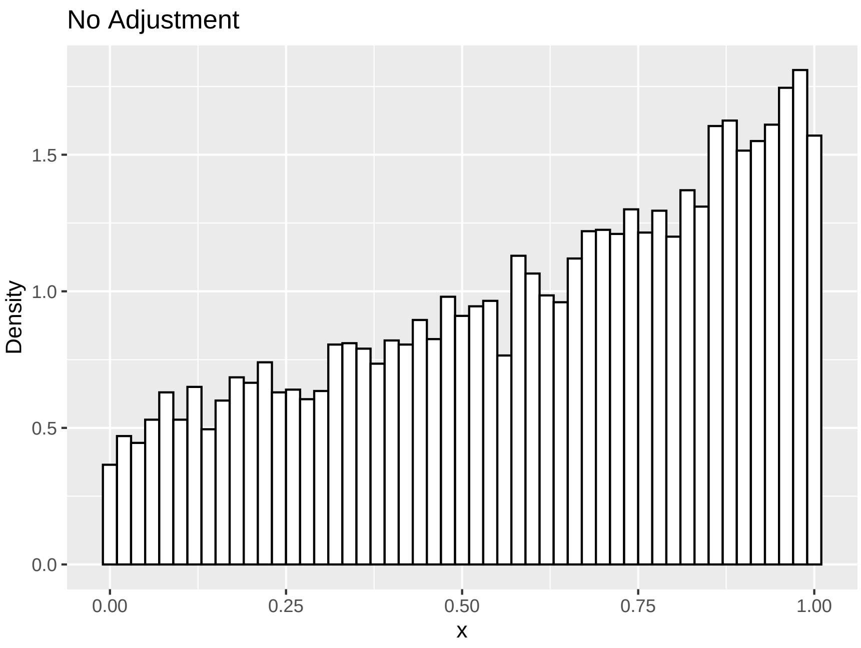
The following graphs show the histograms of the p values of 10000 simulated experiments. Clearly the test without the adjustment is wrong, even for large sample size.

```
X2 <- function(B=1e4,n=100,m=5,p=0.4) {
   A=rep(0,B)
   for(i in 1:B) {
     x=rbinom(n,m,p)
     E=n*dbinom(0:m, m, mean(x)/m)
     O=table(x)
     if(length(O)<6) O=c(O,0)
     A[i]=sum((O-E)^2/E)
   }
   A
}
A=X2()
df=data.frame(p1=1-pchisq(A, 6-1),
              p2=1-pchisq(A, 6-1-1))
```

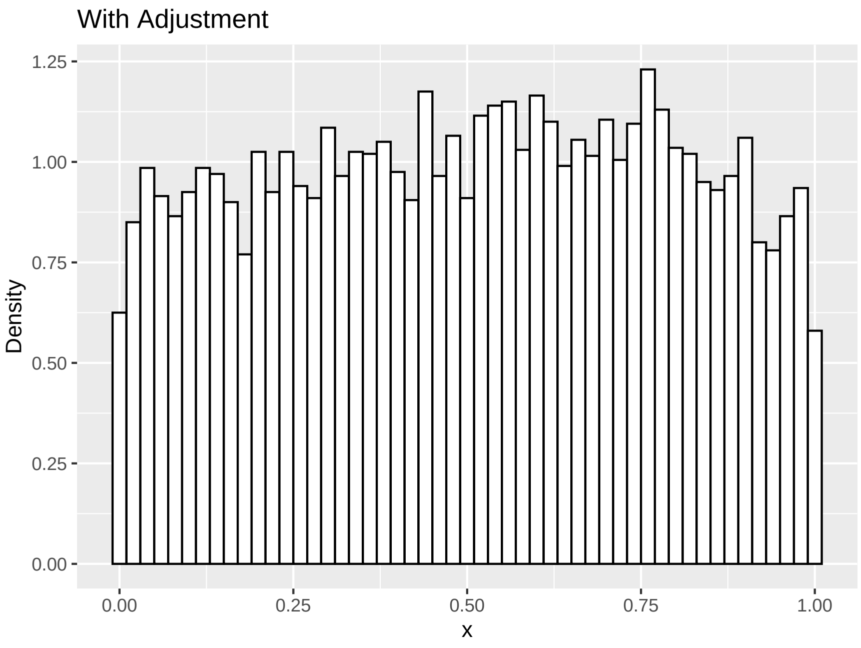
```
A[1:5]
```

```
## [1] 1.880161 2.587287 7.847604 4.226324 3.489151
```

```
bw <- 1/50
ggplot(df, aes(p1)) +
  geom_histogram(aes(y = ..density..),
    color = "black",
    fill = "white",
    binwidth = bw) +
    labs(x = "x", y = "Density") +
   labs(title = "No Adjustment")
```



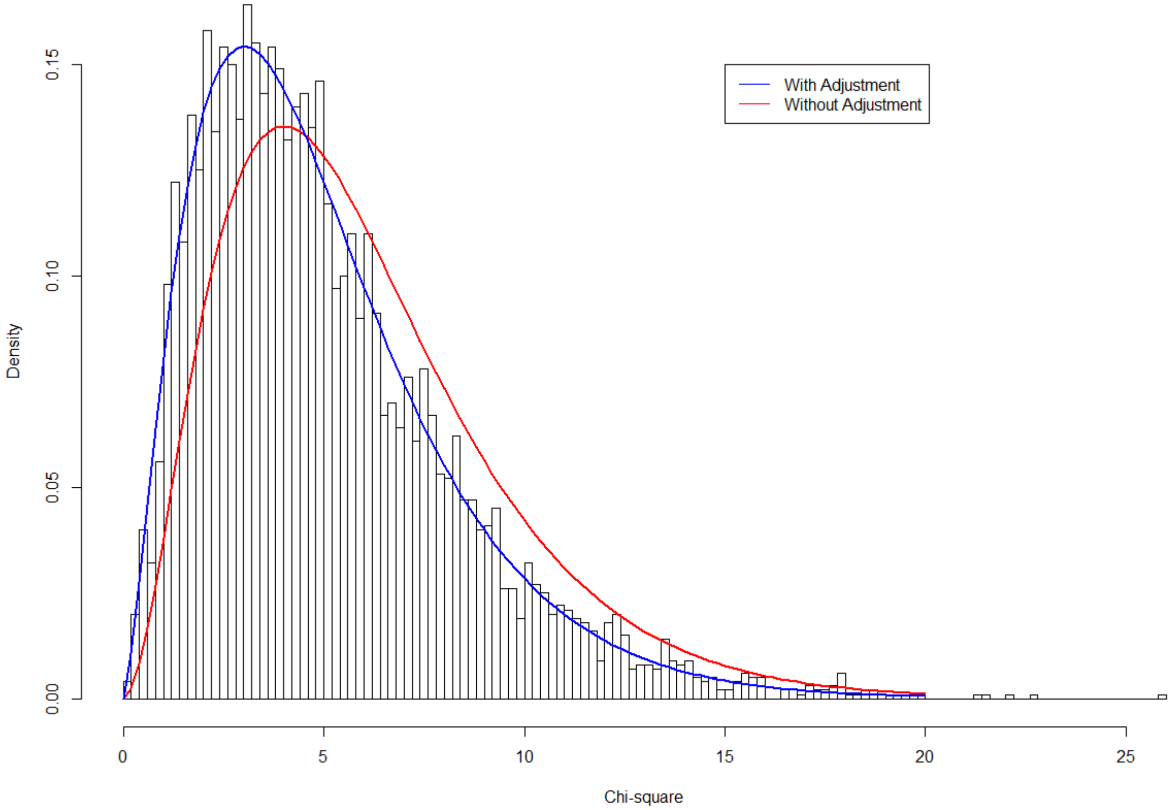
```
bw <- 1/50
ggplot(df, aes(p2)) +
  geom_histogram(aes(y = ..density..),
    color = "black",
    fill = "white",
    binwidth = bw) +
    labs(x = "x", y = "Density") +
   labs(title = "With Adjustment")
```



7.7.1.5 Example (7.7.5) Say we have $X_1, ..X_n \sim Pois(\lambda)$.

The following is the histogram of 5000 simulated data sets with n=10000, $\lambda = 1.0$, and bins x=0, .., x=5, x>5. In each run λ is estimated by minimum chi-square.

The red curve is dchisq(x,7-1) and the blue one is dchisq(7-1-1)



7.7.1.6 Example (7.7.6) Say we have a data set and we want to test whether is comes from a normal distribution. In order to use the χ^2 test we first need to bin the data. There are two basic strategies:

- use equal size bins (with the exception of the first and the last)
- use adaptive bins chosen so that each bin has roughly the same number of observations.

Testing for normality is a very important problem, although because of simulation not quite as important today as it used to be. There are a number of test available for this problem, most of them much better (that is with higher power) than the chisquare test. Look for example for the Shapiro-Wilks test and the Anderson-Darling test.

A very good way to assess the distribution of a sample (such as normality) is to draw a graph specifically designed for this purpose, the probability plot. It plots the sample quantiles vs. the quantiles of the hypothesized distribution. If the data follows that distribution the resulting plot should be linear.

7.7.2 Kolmogorov-Smirnov Test

Say we have X_1, .., X_n which are continuous and independent r.v. and we wish to test

$$H_0 : X_i \sim F$$

for all i

Above we talked about graphs that give us some idea whether the data really comes from a certain distribution. Now let's use the empirical distribution function. If the null hypothesis is true, than the empirical cdf should be close to the true one, that is the "distance" between the two curves should be small.

How can we define this "distance"? in mathematics there are a number of possible definitions:

- L^1 norm

$$\int_{-\infty}^{\infty} |F(x - \hat{F}(x)| dx$$

- L^2 norm

$$\int_{-\infty}^{\infty} \left(F(x - \hat{F}(x)\right)^2 dx$$

- L^∞ norm

$$\max \left\{ |F(x - \hat{F}(x)| : x \in R \right\}$$

We are going to consider the L^∞ norm here, so we have the test statistic

$$D = \max \left\{ |F(x) - \hat{F}(x)| : x \in R \right\}$$

This is called the *Kolmogorov-Smirnov statistic.*

At first glance it appears that computing D is hard: it requires finding a maximum of a function which is not differentiable. But inspection of the graphs (and a little calculation) shows that the maximum has to occur at one of the jump points, which in turn happen at the observations. So all we need to do is find $\mathrm{F}(\mathrm{X}_\mathrm{i})$-$\mathrm{Fhat}(\mathrm{X}_\mathrm{i})$ for all i.

Next we need the null distribution, that is the distribution of D if the null hypothesis is true. The full derivation is rather lengthy and won't be done here, but see for example J.D

Gibbons, *Nonparametric Statistical Inference*. The main result is that if F is continuous and X ~ F, then F(X) ~ U[0,1], and therefore D does not depend on F, it is called a distribution-free statistic. It's distribution can be found by simply assuming that F is U[0,1].

The method is implemented in R in the routine **ks.test** where x is the data set and y specifies the null hypothesis, For example y="pnorm" tests for the normal distribution. Parameters can be given as well. For example ks.test(x,"pnorm",5,2) tests whether X~N(5,2).

Note that this implementation does **not** allow us to estimate parameters from the data. Versions of this test which allow such estimation for some of the standard distributions are known, but not part of R. We can of course use simulation to implement such tests.

It is generally recognized that the Kolmogorov-Smirnov test is much better than the Chisquare test.

For our general discussion this test is interesting because it does not derive from any specific principle such as the likelihood principle. It is simply an idea (let's compare the cdf under H_0 with the empirical cdf) and a lot of heavy probability theory. Such methods are quite common in Statistics.

7.8 Nuisance Parameters

7.8.0.1 Example (7.8.1) say $X \sim Pois(\mu + b)$. We have previously studied this problem under the assumption that b is known. Now let's consider what we can do if b is unknown.

In order to do anything we need another measurement, so say we also have Y~Pois(b), X and Y independent

Let's say we want to find interval estimates μ.

First note that the problem does not say anything about b. In such a situation b is called a **nuisance parameter**.

Whether (or which) parameter is a nuisance parameter is entirely dependent on the interest of the investigator, it could just as well have been μ, or none of the two.

7.8.1 Frequentist Inference

Let's derive and then invert the likelihood ratio test for this problem. First we have the joint density of X and Y:

$$f(x, y|\mu.b) = \frac{(\mu + b)^x}{x!} e^{-\mu - b} \frac{b^y}{y!} e^{-b}$$

so we find the mle's as:

$$
\begin{aligned}
&l(\mu, b|x, y) = x\log(\mu+b) - \log x! - \mu - b + y\log b - \log y! - b\\
&\frac{dl}{d\mu} = \frac{x}{\mu+b} - 1 = 0\\
&\hat{\mu} = x - b\\
&\frac{dl}{db} = \frac{x}{\mu+b} - 1 - \frac{y}{b} - 1 = 0\\
&\frac{x}{x-b+b} - \frac{y}{b} = 2\\
&\hat{b} = y
\end{aligned}
$$

Under the null hypothesis we have $H_0 : \mu = \mu_0$, and so

$$
\begin{aligned}
&\frac{dl(\mu_0, b|x, y)}{db} = \frac{x}{\mu_0+b} - \frac{y}{b} - 2 = 0\\
&xb - y(\mu_0+b) - 2(\mu_0+b)b = 0\\
&-2b^2 + (x+y-2\mu_0)b + y\mu_0 = 0\\
&b_{1,2} = \left(-(x+y-2\mu_0)b \pm \sqrt{(x+y-2\mu_0)^2 - 4(-2)y\mu_0}\right)/[2(-2)]\\
&\hat{\hat{b}} = \left(x+y-2\mu_0 \pm \sqrt{(x+y-2\mu_0)^2 + 8y\mu_0}\right)/4
\end{aligned}
$$

$$
\begin{aligned}
\lambda(x, y) = {} & \frac{\frac{(\mu_0+\hat{\hat{b}})^x}{x!}e^{-\mu_0-\hat{\hat{b}}}\frac{\hat{\hat{b}}^y}{y!}e^{-\hat{\hat{b}}}}{\frac{(\hat{\mu}+\hat{b})^x}{x!}e^{-\hat{\mu}-\hat{b}}\frac{\hat{b}^y}{y!}e^{-\hat{b}}} =\\
& \frac{2\mu_0 - x - y \pm \sqrt{(x+y-2\mu_0)^2 + 8y\mu_0}}{4x^x y^y}\cdot\\
& \exp\left\{\left(x+y-2\mu_0 \pm \sqrt{(x+y-2\mu_0)^2 + 8y\mu_0}\right)/4\right\} =\\
& =
\end{aligned}
$$

This is called the method of **profile likelihood**.

Now intervals can be found using the chisquare approximation to the likelihood ratio statistic. The resulting limits are called the *Rolke-Lopez-Conrad* limits

7.8.2 Bayesian Solution

Here we need priors for μ and b. Let's use $\mu \sim 1$ and $b \sim 1$. Then

$$m(x,y) = \int_0^\infty \int_0^\infty f(x,y|\mu,b)\pi(\mu)\pi(b)d\mu db =$$

$$\frac{1}{x!y!}\int_0^\infty b^y e^{-2b}\left[\int_0^\infty (\mu+b)^x e^{-\mu} d\mu\right] db$$

$$\int_0^\infty (t+b)^x e^{-t} dt =$$

$$\int_0^\infty \sum_{n=0}^x \binom{x}{n} b^x t^{x-n} e^{-t} dt =$$

$$\sum_{n=0}^x \binom{x}{n} b^x \int_0^\infty t^{x-n} e^{-t} dt =$$

$$\sum_{n=0}^x \binom{x}{n} b^x \int_0^\infty t^{(x-n+1)-1} e^{-t} dt =$$

$$\sum_{n=0}^x \binom{x}{n} b^x \Gamma(x-n+1)$$

$$m(x,y) =$$

$$\frac{1}{x!y!}\int_0^\infty b^y e^{-2b}\left[\sum_{n=0}^x \binom{x}{n} b^x \Gamma(x-n+1)\right] db =$$

$$\frac{1}{x!y!2^{y+n+1}} \sum_{n=0}^x \binom{x}{n} \Gamma(x-n+1) \int_0^\infty (2b)^{(y+n+1)-1} e^{-2b} (2db) =$$

$$\frac{1}{x!y!2^{y+n+1}} \sum_{n=0}^x \binom{x}{n} \Gamma(x-n+1)\Gamma(y+n+1)$$

and so the posterior distribution is given by

$$f(\mu,b|x,y) = \frac{f(x,y,\mu,b)}{m(x,y)}$$

and for the parameter of interest we can find the marginal

$$f(\mu|x,y) = \int f(\mu,b|x,y)db =$$

$$\frac{1}{x!y!2^{x+y-n+1}m(x,y)} \sum_{n=0}^x \binom{x}{n} (x+y-n)!\mu^n e^{-\mu}$$

and now intervals can be derived from the posterior distribution, for example via the highest posterior density method.

In a paper in 2015 I could show that intervals derived in this way but then treated as frequentist confidence intervals actually have good coverage properties. If one uses the prior μ~1/ μ, though, they do not.

Notice an important distinction between the way nuisance parameters are treated by frequentists and by Bayesians: in one case we use differentiation (to find the profile likelihood), in the other integration (to find the marginal distribution).

7.9 Inference for Stochastic Processes

Note Here I discuss topics like Poisson process and Markov chains. If you are not familiar with these don't worry!

Most of the basic methods from Statistics apply to stochastic processes as well.

7.9.0.1 Example (7.9.1) The data set *ex1* has 1000 observations from some discrete-time discrete-state space Markov chain:

```
ex1[1:20]
```

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

```
ex1[981:1000]
```

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

We want to estimate the transition matrix. Let's use maximum likelihood. Let $p_{ij} = P(X_2 = j|X_1 = i)$ be the transition probabilities, and let n_{ij} be the number of times the chain went from i to j, then the likelihood function is given by

$$L(p_{11}, .., p_{55}) = \prod_{1\le i,j\le 5} p_{ij}^{n_{ij}}$$

and we need to maximize this subject to the conditions $\sum_{j=1}^{5} p_{ij} = 1$, i=1,..,5.

As always we use the log-likelihood, and then using Lagrange multipliers we have

$$\sum_{1\le i,j\le 5} n_{ij}\log p_{ij} + \sum_{i=1}^{5}\lambda_i\left(\sum_{j=1}^{5} p_{ij} - 1\right)$$

.

Now

$$\begin{aligned}
&\frac{d}{dp_{ij}} = \frac{n_{ij}}{p_{ij}} - \sum_{i=1}^{5}\lambda_i = 0\\
&p_{ij} = n_{ij}/\sum_{i=1}^{5}\lambda_i\\
&1 = (\sum_{j=1}^{5} n_{ij})/\sum_{i=1}^{5}\lambda_i\\
&\sum_{i=1}^{5}\lambda_i = n_{i.}\\
&\hat{p}_{ij} = n_{ij}/n_{i.}
\end{aligned}$$

```
n <- matrix(0, 5, 5)
for(i in 2:1000) {
  n[ex1[i-1], ex1[i]] <- n[ex1[i-1], ex1[i]]+1
}
n. <- apply(n, 1, sum)
phat <- round(n/n., 2)
dimnames(phat) <- list(1:5, 1:5)
kable.nice(phat)
```

	1	2	3	4	5
1	0.11	0.34	0.00	0.36	0.19
2	0.00	0.36	0.19	0.45	0.00
3	0.00	0.36	0.17	0.46	0.00
4	0.33	0.00	0.09	0.11	0.46
5	0.10	0.28	0.24	0.00	0.37

Above we started with the assumption that this sequence is from a Markov chain. Can we test this? To do so we have to carry out the following hypothesis test. Let

$$p_{(i,j)k} = P(X_3 = k|X_1 = i, X_2 = j)$$

then the Markov property (plus stationarity) imply

$$p_{(i,j)k} = p_{jk}$$

Applying the likelihood ratio test and the usual Taylor approximation to the logarithm yields the chi-square test statistic

$$X^2 = \sum_{i,j,k} \left(n_{ijk} - e_{ijk}\right)^2 / e_{ijk}$$

where n_{ijk} is the number of transitions from i to j to k and $e_{ijk} = n_{ij.} n_{jk.} / n_{j..}$. Under the null hypothesis of a Markov chain we X^2 will have a chi-square distribution with c^3 degrees of freedom.

The test is implemented in the *markovchain* library:

```
library(markovchain)
verifyMarkovProperty(ex1)
```

```
## Testing markovianity property on given data sequence
## Chi - square statistic is: 37.57234
## Degrees of freedom are: 50
## And corresponding p-value is: 0.9024476
```

Could this sequence actually come from independent observations? We can of course do the basic test for independence, but notice that we never saw a transition from 3 to 1, which is impossible under independence!

7.9.0.2 Example (7.9.2)

A store wants to find out about the times when customers entered the store. They open at 8am and close at 6pm. For 20 working days they record the times and find

```
##  [1]  483.1900  485.7712  504.1513  511.2431  533.9519  556.3276  564.5901  600.1509
##  [9]  601.2124  613.2977  615.0961  656.5382  672.0500  678.0181  691.2101  738.8729
## [17]  742.7525  751.9882  767.3979  782.9098  816.6775  832.7054  840.4809  861.4346
## [25]  869.5839  894.2626  925.1999  997.7035  998.6705  999.6550 1011.9642 1028.8511
## [33] 1049.1714
```

```
Day[[1]]
```

```
 [1] "8-03-11"  "8-05-46"  "8-24-09"  "8-31-14"  "8-53-57"  "9-16-20"  "9-24-35"  "10-00
 [9] "10-01-13" "10-13-18" "10-15-06" "10-56-32" "11-12-03" "11-18-01" "11-31-13" "12-18
[17] "12-22-45" "12-31-59" "12-47-24" "13-02-55" "13-36-41" "13-52-43" "14-00-29" "14-21
[25] "14-29-35" "14-54-16" "15-25-12" "16-37-42" "16-38-40" "16-39-39" "16-51-58" "17-08
[33] "17-29-10"
```

so on day 1 the first customer came in at 8h03m11s and the last one at 17h29m10s.

Do the arrivals follow a Poisson distribution?

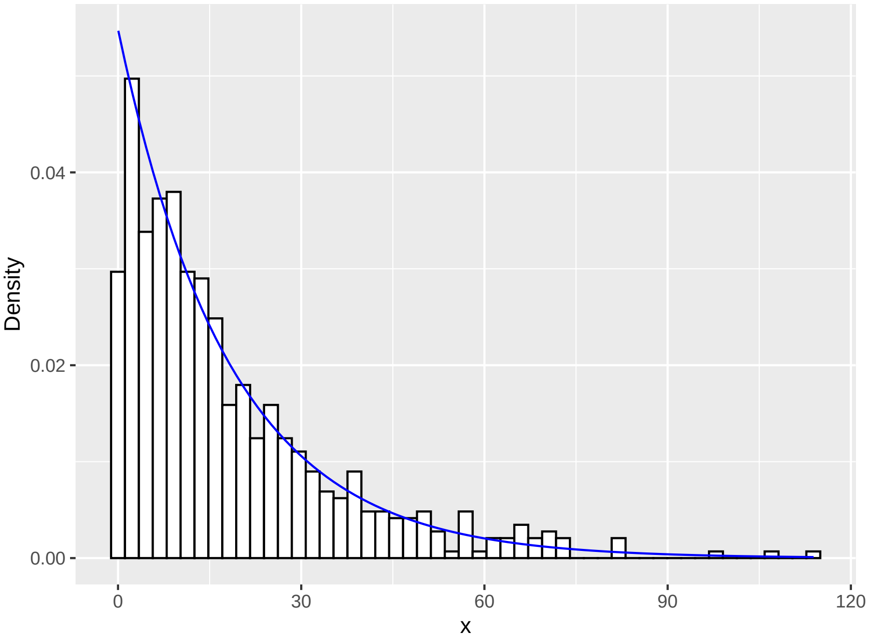
We will assume that from the details of this store we know that the arrivals of customers have independent and stationary increments, so what we need to do is test whether the interarrival times come from an exponential distribution. For this we can use the *Lilliefors test*. It is implemented in the package *KScorrect*.

First, though we need the interarrival times:

```
inter.arrival.times <- NULL
for(i in 1:20) {
  tmp <- strsplit(Day[[i]], "-")
  hours <- as.numeric(unlist(tmp)[ c(TRUE, FALSE, FALSE) ])
  minutes <- as.numeric(unlist(tmp)[ c(FALSE, TRUE, FALSE) ])
  seconds <- as.numeric(unlist(tmp)[ c(FALSE, FALSE, TRUE) ])
  tmp <- c(8*60, hours*60+minutes+seconds/100)
  inter.arrival.times <- c(inter.arrival.times,
                           diff(tmp))
}
```

Let's look at a graph:

```
df <- data.frame(x=inter.arrival.times)
bw <- diff(range(inter.arrival.times))/50
ggplot(df, aes(x)) +
  geom_histogram(aes(y = ..density..),
    color = "black",
    fill = "white",
    binwidth = bw) +
    labs(x = "x", y = "Density") +
  stat_function(fun = dexp, colour = "blue", args=list(rate=1/mean(df$x)))
```



This looks ok. Now for the test:

```
library(KScorrect)
LcKS(inter.arrival.times, "pexp")$p.value
```

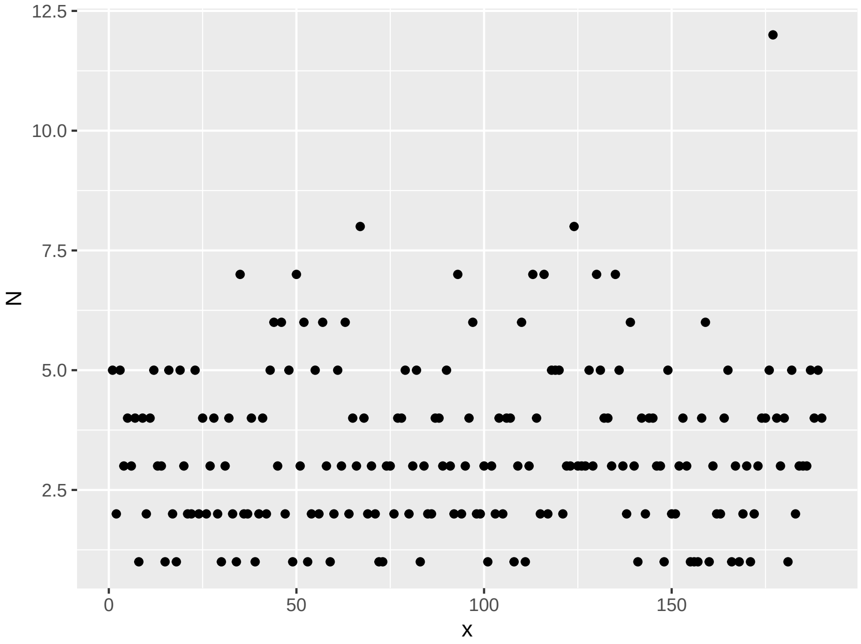
```
## [1] 0.9096
```

and so it seems the interarrival times do come from an exponential distribution.

Can we also test for the other conditions of a Poisson process, for example stationary increments? This implies that the number of arrivals over equal length time periods has the same distribution. Let's check that for time periods of on hour:

```
N <- NULL
for(i in 1:20) {
  tmp <- strsplit(Day[[i]], "-")
  hours <- as.numeric(unlist(tmp)[ c(TRUE, FALSE, FALSE) ])
  N <- c(N, as.numeric(table(hours)))
}
```

```
df <- data.frame(x=1:length(N), N=N)
ggplot(data=df, aes(x, N)) +
  geom_point()
```



and it does not appear that there is a change in the distribution over time.

7.10 Approximation Methods

7.10.1 Taylor Approximations

Say we have a r.v. X with density f, a function h and we want to know var(h(X)). Of course by definition we have

$$var(X) = \int_{-\infty}^{\infty} x^2 f(x)dx - \left(\int_{-\infty}^{\infty} xf(x)dx\right)^2$$

but sometimes these integrals (sums) are very difficult to evaluate. In this section we discuss some methods for approximating the variance.

Recall: If a function h(x) has derivatives of order r, that is if $g^{(r)}$(x) exists, then for any constant a the *Taylor polynomial* of order r is defined by

$$T_r(x) = \sum_{n=0}^{r} \frac{h^{(n)}(a)}{n!}(x-a)^n$$

One of the most famous theorems in mathematics called Taylor's theorem states that the remainder of the approximation h(x)-T$_{\mathrm{r}}$(x) goes to 0 faster than the highest order term:

7.10.1.1 Theorem (8.10.1) Taylor's theorem

$$\lim_{x \to a} \frac{h(x) - T_r(x)}{(x-a)^r} = 0$$

There are various formulas for the remainder term, but we won't need them here.

7.10.1.2 Example (8.10.2) say $h(x) = \log(x+1)$ and we want to approximate h at x=0. Then we have

$$\begin{aligned}
&h(0) = \log(1) = 0 \\
&\frac{dh}{dx}|_{x=0} = \frac{1}{x+1}|_{x=0} = 1 \\
&\frac{d^2h}{dx^2}|_{x=0} = -\frac{1}{(x+1)^2}|_{x=0} = -1 \\
&\frac{d^3h}{dx^3}|_{x=0} = \frac{2}{(x+1)^3}|_{x=0} = 2 \\
&\frac{d^{(r)}h}{dx^r}|_{x=0} = \frac{(-1)^{r-1}(r-1)!}{(x+1)^r}|_{x=0} = (-1)^{r-1}(r-1)!
\end{aligned}$$

and so

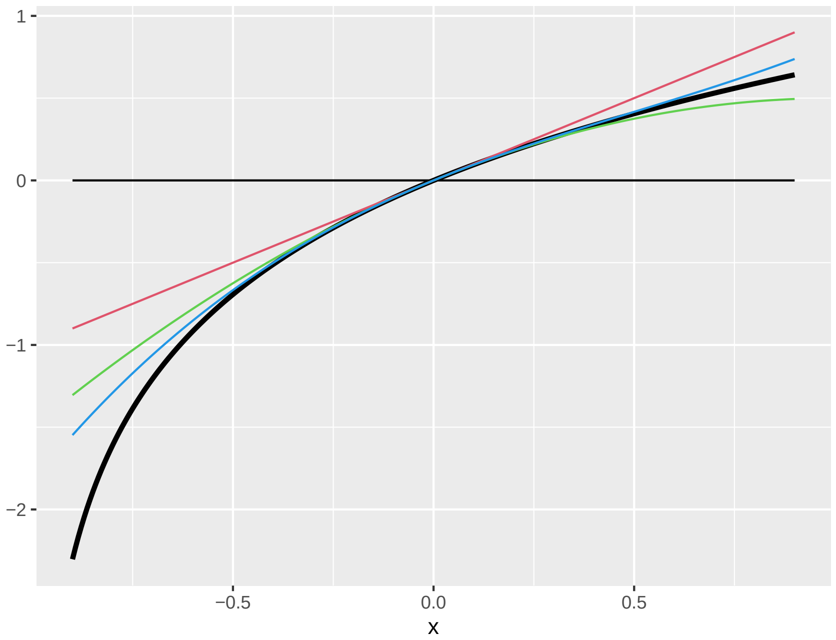
$$\begin{aligned}
&T_0(x) = h(0) = 0 \\
&T_1(x) = T_0(x) + \frac{dh}{dx}|_{x=0} \cdot (x-0) = x \\
&T_2(x) = T_1(x) + \frac{d^2h}{dx^2}|_{x=0} \cdot (x-0)^2 = x - x^2/2 \\
&T_3(x) = T_2(x) + \frac{d^3h}{dx^3}|_{x=0} \cdot (x-0)^3 = x - x^2/2 + x^3/2
\end{aligned}$$

$$T_r(x) = \sum_{n=0}^{r} \frac{(-1)^{n-1}(n-1)!}{n!} x^n = \sum_{n=0}^{r} \frac{(-1)^{n-1}x^n}{n}$$

The approximation is illustrated here:

```r
a <- 0; r <- 3
x <- seq(-0.9, 0.9, length = 250)
h <- rep(0, r+1)
h[1] <- log(a+1)
for (n in 1:r) h[n+1] <- (-1)^(n+1)/n/(a+1)^n
y <- matrix(0, 250, r+1)
```

```r
y[, 1] <- rep(log(a+1), 250)
for (k in 1:r) y[ , k+1] <- y[, k] + h[k+1]*(x-a)^k
df <- data.frame(x=x, ly=log(x+1))
plt <- ggplot(df, aes(x, ly)) +
        geom_line(size=1.2) + ylab("")

for (k in 1:(r + 1))
  plt <- plt +
    geom_line(data=data.frame(x=x, y=y[, k]), aes(x, y), color=k)
plt
```



One application of this is the

7.10.2 Delta Method

7.10.2.1 Theorem (8.10.3) Let Y_n be a sequence of rv's that satisfies

$$\sqrt{n}(Y_n - \theta) \to N(0, \sigma)$$

in distribution. For a given function g and a specific value of θ, suppose that $g'(\theta)$ exist and and is not 0. Then

$$\sqrt{n}\left(g(Y_n) - g(\theta)\right) \to N(0, \sigma g'(\theta))$$

proof

the Taylor expansion of $g(Y_n)$ around $Y_n = \theta$ is

$g(Y_n) = g(\theta) + g'(\theta)(Y_n - \theta) + R$

where $R \to 0$ as $Y_n \to \theta$. Now

$\sqrt{n}[g(Y_n) - g(\theta)] =$

$\sqrt{n}[g(\theta) + g'(\theta)(Y_n - \theta) + R - g(\theta)] =$

$g'(\theta)\sqrt{n}(Y_n - \theta) + \sqrt{n}R \to g'(\theta)X$

where $X \sim N(0, \sigma)$

7.10.2.2 Example (8.10.4) say $X_1, .., X_n$ iid Exp(1), so E[X]=var(X)=1, then by the CLT

$\sqrt{n}(\bar{X} - 1) \to N(0, 1)$

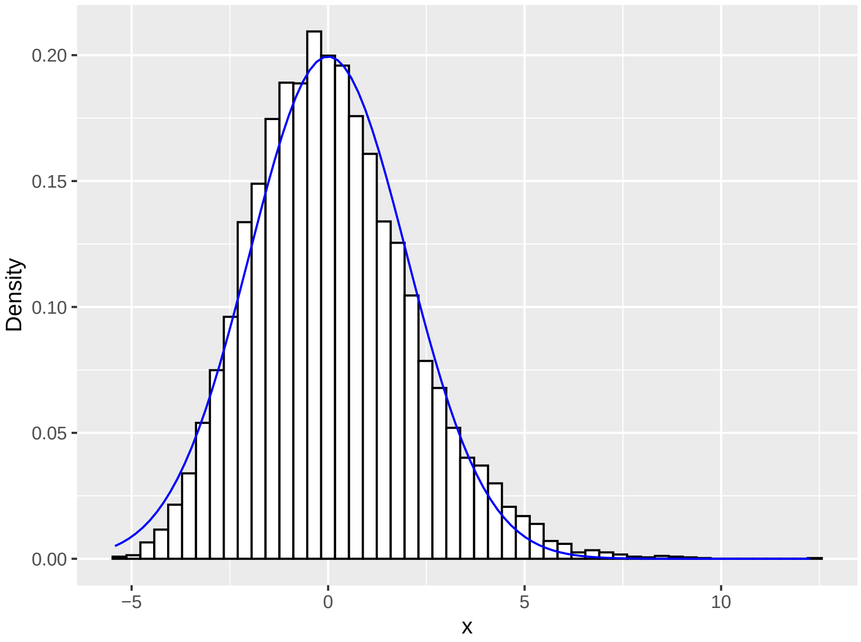
Let g(x)=x^{p}, so g’(x)=px^{p-1} and by the delta method

$\sqrt{n}(\bar{X}^p - 1) \to N(0, p)$

```
n <- 100; p <- 2; B <- 10000
x <-  matrix(rexp(n*B, 1), ncol=n)
xbar <- apply(x, 1, mean)
print(mean(xbar))
```

```
## [1] 0.9999839
```

```
y <- sqrt(n)*(xbar^p-1)
bw <- diff(range(y))/50
df <- data.frame(x=y)
ggplot(df, aes(x)) +
  geom_histogram(aes(y = ..density..),
    color = "black",
    fill = "white",
    binwidth = bw) +
    labs(x = "x", y = "Density") +
  stat_function(fun = dnorm,
                colour = "blue",
                args=list(mean=0, sd=p))
```



say we have a sequence of iid rv's $X_1, .., X_n$, each with mean $\mu \neq 0$ and standard deviation σ. We know from the law of large numbers that

$\sqrt{n}(\bar{X} - \mu) \rightarrow N(0, \sigma)$

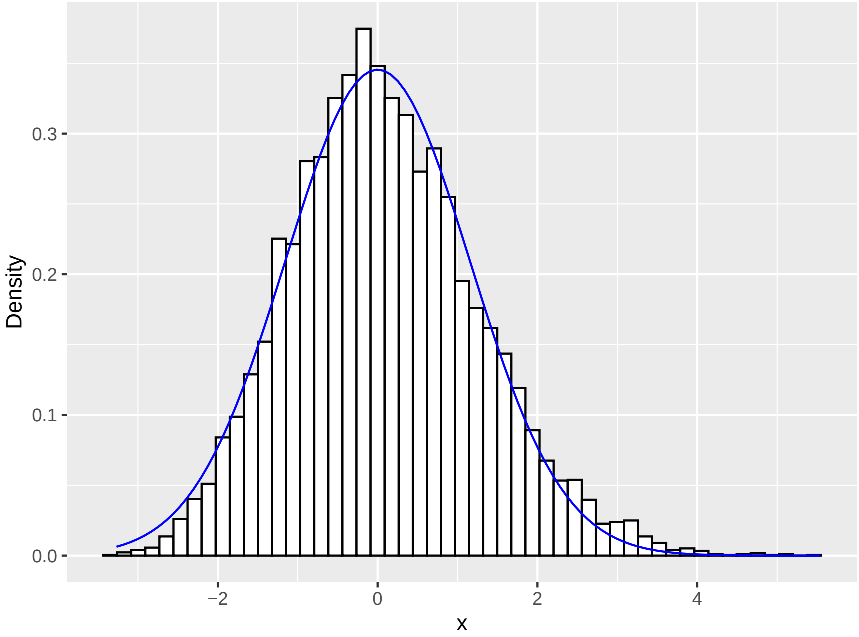
Now let g(x)=1/x, then g'(x)=-1/x^2 and we get

$\sqrt{n}(1/\bar{X} - 1/\mu) \rightarrow N(0, \sigma/\mu^2)$

say for example $X_i \sim U[0,1]$, then $\mu = 1/2$ and $\sigma = 1/\sqrt{12}$, so according to the delta method

$\sqrt{n}(1/\bar{X} - 1/\mu) \sim N(0, \sigma/\mu^2) = N(0, 2/\sqrt{3})$

```
n <- 100; B <- 10000
x <-  matrix(runif(n*B), ncol=n)
xbar <- apply(x, 1, mean)
y <- sqrt(n)*(1/xbar - 1/0.5)
bw <- diff(range(y))/50
df <- data.frame(x=y)
ggplot(df, aes(x)) +
  geom_histogram(aes(y = ..density..),
    color = "black",
    fill = "white",
    binwidth = bw) +
    labs(x = "x", y = "Density") +
  stat_function(fun = dnorm,
                colour = "blue",
                args=list(mean=0, sd=2/sqrt(3)))
```



7.10.3 Variance Approximations based on Taylor's Theorem

For our purposes we will need only first-order approximations (that is using the first derivative) but we will need a multivariate extension as follows: say X_1, ..,X_n are r.v. with means $\mu_1, .., \mu_n$ and define $\mathbf{X}$=(X_1, ..,X_n) and $\boldsymbol{\mu} = (\mu_1, .., \mu_n)$. Suppose there is a differentiable function h($\mathbf{X}$) for which we want an approximate estimate of the variance. Define

$$h_i'(\boldsymbol{\mu}) = \frac{\partial h(\boldsymbol{t})}{\partial t_i}|_{\boldsymbol{t}=\boldsymbol{\mu}}$$

Then first order Taylor expansion of h about $\boldsymbol{\mu}$ is

$$h(\boldsymbol{t}) = h(\boldsymbol{\mu}) + \sum_{i=1}^{n} h_i'(\boldsymbol{\mu})(t_i - \mu_i) + \text{Remaindeer}$$

Forgetting about the remainder we have

$$E[h(\boldsymbol{X})] \approx E[h(\boldsymbol{\mu}) + \sum_{i=1}^{n} h_i'(\boldsymbol{\mu})(X_i - \mu_i)] =$$

$$h(\boldsymbol{\mu}) + \sum_{i=1}^{n} h_i'(\boldsymbol{\mu})(E[X_i] - \mu_i) = h(\boldsymbol{\mu})$$

$$
\begin{aligned}
&var(h(\boldsymbol{X})) \approx E[(h(\boldsymbol{X}) - h(\boldsymbol{\mu})^2] \approx \\
&E\left[\left(\sum_{i=1}^n h_i'(\boldsymbol{\mu})(X_i-\mu_i)\right)^2\right] = \\
&E\left[\sum_{i,j=1}^n h_i'(\boldsymbol{\mu})(X_i-\mu_i)h_j'(\boldsymbol{\mu})(X_j-\mu_j)\right] = \\
&E\left[\sum_{i=1}^n (h_i'(\boldsymbol{\mu}))^2(X_i-\mu_i)^2\right] + \\
&2E\left[\sum_{i<j=1}^n h_i'(\boldsymbol{\mu})h_j'(\boldsymbol{\mu})(X_i-\mu_i)(X_j-\mu_j)\right] = \\
&\sum_{i=1}^n (h_i'(\boldsymbol{\mu}))^2 E\left[(X_i-\mu_i)^2\right] + \\
&2\sum_{i<j=1}^n h_i'(\boldsymbol{\mu})h_j'(\boldsymbol{\mu})E\left[(X_i-\mu_i)(X_j-\mu_j)\right] = \\
&\sum_{i=1}^n (h_i'(\boldsymbol{\mu}))^2 var(X_i) + 2\sum_{i<j=1}^n h_i'(\boldsymbol{\mu})h_j'(\boldsymbol{\mu})cov\left(X_i, X_j\right)
\end{aligned}
$$

7.10.3.1 Example (8.10.5) Say we have just one rv X, then the formula simplifies to

$$var(h(X)) \approx (h'(\mu))^2 var(X)$$

say $X \sim N(\mu, 1)$ with μ large enough so that P(X>0)=1. We want to find $var(\log(X))$. Set h(x)=log(x), then h'(x)=1/x and

$$var(\log(X)) \approx (\frac{1}{\mu})^2 \times 1 = \frac{1}{\mu^2}$$

check with

```
var(log(rnorm(10000, 10)))
```

```
## [1] 0.0103564
```

7.10.3.2 Example (8.10.6) say we have a sample X_1, ..,X_n from a Bernoulli r.v. with success parameter p, that is P(X=1)=p=1-P(X=0). One popular measure of the probability of winning a game is the odds p/(1-p). For example when you roll a fair die the odds of getting a six are (1/6)/(1-(1/6) = 1:5.

An obvious estimator for p is $\hat{p}$, the sample mean, or here the proportion of "successes" in the n trials. Then an obvious estimator for the odds is $\hat{p}/(1-\hat{p})$. The question is, what is the variance of this estimator?

First note that

$$
\begin{aligned}
&var(\hat{p}) = \\
&var(1/n \sum X_i) = \\
&\frac{1}{n^2} \sum var(X_i) = \\
&\frac{1}{n} var(X_1) = \\
&p(1-p)/n
\end{aligned}
$$

Using the above approximation we get the following: let h(p)=p/(1-p), so h'(p)=1/(1-p)^2 and

$$
\begin{aligned}
&var(\frac{\hat{p}}{1-\hat{p}}) \approx (h'(p))^2 var(\hat{p}) = \\
&\left[\frac{1}{(1-p)^2}\right]^2 \frac{p(1-p)}{n} = \frac{p}{n(1-p)^3}
\end{aligned}
$$

```r
p <- 0.25; n <- 25; B <- 10000
x <- matrix(rbinom(B, 1, p), ncol=n)
phat <- apply(x, 1, mean)
odds <- phat/(1 - phat)
round(c(var(odds), p/n/(1 - p)^3), 4)
```

```
## [1] 0.0279 0.0237
```

Say we have two rv's X and Y and $X \perp Y$, then the formula simplifies to

$$var(h(X,Y)) \approx h_x^2(\mu_x, \mu_y) var(X) + h_y^2(\mu_x, \mu_y) var(Y)$$

7.10.3.3 Example (8.10.7) say X and Y have a geometric distribution with parameters p and r, respectively. We want to approximate the variance of $\sqrt{X^2+Y^2}$

Now $\mu_X = 1/p, var(X) = (1-p)/p^2, \mu_Y = 1/r, var(Y) = (1-r)/r^2$

let $h(x,y) = \sqrt{x^2+y^2}$, then

$$
\begin{aligned}
&\frac{dh}{dx} = \frac{x}{x^2+y^2}\\
&\frac{dh}{dy} = \frac{y}{x^2+y^2}\\
&var(\sqrt{X^2+Y^2}) =\\
&\left(\frac{\mu_x}{\mu_x^2+\mu_y^2}\right)^2 var(X) + \left(\frac{\mu_y}{\mu_x^2+\mu_y^2}\right)^2 var(Y) =\\
&\left(\frac{1/p}{1/^2+1/r^2}\right)^2 \frac{1-p}{p^2} + \left(\frac{1/r}{1/^2+1/r^2}\right)^2 \frac{1-r}{r^2} =\\
&\frac{1-p}{p^2(1+(p/r)^2)} + \frac{1-r}{r^2(1+(r/p)^2)}
\end{aligned}
$$

```r
p <- 0.2; r <- 0.3; B <- 10000
x <- rgeom(B, p)+1
y <- rgeom(B, r)+1
round(c(var(sqrt(x^2+y^2)),
  (1-p)/(p^2*(1+(p/r)^2))+(1-r)/(r^2*(1+(r/p)^2))), 3)
```

```
## [1] 19.882 16.239
```

7.10.3.4 Example (8.10.8) let's consider the random vector with joint density $f(x,y)=1$, $0<x,y<1$.

Say we want to find var(X/Y). Of course $X,Y \sim U[0,1]$ and independent, so we know E[X]=E[Y]=1/2 and var(X)=var(Y)=1/12.

If we consider the function h(x,y) = x/y we have $h_x(x,y)=1/y$ and $h_y(x,y)=-x/y^2$ and so

$$
\begin{aligned}
&var(X/Y) \approx h_x^2(\mu_x,\mu_y)var(X) + h_y^2(\mu_x,\mu_y)var(Y) =\\
&(\frac{1}{1/2})^2\frac{1}{12} + (-\frac{1/2}{(1/2)^2})^2\frac{1}{12} = \frac{2}{3}
\end{aligned}
$$

How good is this approximation?

```r
var(runif(10000)/runif(10000))
```

```
## [1] 4782.344
```

shows that it is actually very bad! The reason is that occasionally the denominator is very small, so the ratio is very big.

Let's change the problem a little: now $f(x,y)=1$, $1<x,y<2$.

that is $X,Y \sim U[1,2]$, so E[X]=E[Y]=3/2, var(X)=var(Y)=1/12. Now

$$
\begin{aligned}
&var(X/Y) \approx h_x^2(\mu_x,\mu_y)var(X) + h_y^2(\mu_x,\mu_y)var(Y) = \\
&(\frac{1}{3/2})^2\frac{1}{12} + (-\frac{3/2}{(3/2)^2})^2\frac{1}{12} = \frac{2}{27}
\end{aligned}
$$

and this is actually quite good:

```r
round(c(2/27, var(runif(10000,1,2)/runif(10000,1,2))), 4)
```

```
## [1] 0.0741 0.0849
```

Generally ratios are often trouble!

7.10.3.5 Example (8.10.9) let's consider the random vector with joint density $f(x,y) = 6x$, $0 < x < y < 1$

Say we want to find car(X/Y)

First we have

$$
\begin{aligned}
&f_x(x) = \int_x^1 6xdy = 6x(1-x); 0 < x < 1 \\
&X \sim Beta(2,2) \\
&E[X] = 1/2 \\
&var(x) = \frac{2\cdot 2}{(2+2)^2(2+2+1)} = \frac{1}{20}
\end{aligned}
$$

$$
\begin{aligned}
&f_y(y) = \int_0^y 6xdx = 3y^2; 0 < y < 1 \\
&X \sim Beta(3,1) \\
&E[X] = 3/4 \\
&var(x) = \frac{3\cdot 1}{(3+1)^2(3+1+1)} = \frac{3}{80}
\end{aligned}
$$

$$
\begin{aligned}
&E[XY] = \int_0^1\int_0^y xy6xdydxdy\frac{2}{5} \\
&cov(X,Y) = E[XY] - E[X]E[Y] = \frac{1}{40}
\end{aligned}
$$

so

$$
\begin{aligned}
&var(X/Y) \approx \\
&h_x^2(\mu_x,\mu_y)var(X) + h_y^2(\mu_x,\mu_y)var(Y) + 2h_x(\mu_x,\mu_y)h_y(\mu_x,\mu_y)cov(X,Y) = \\
&(\frac{1}{3/4})^2\frac{1}{20} + (-\frac{1/2}{(3/4)^2})^2\frac{3}{80} + 2(\frac{1}{3/4})(-\frac{1/2}{(3/4)^2})\frac{1}{40} = 0.058
\end{aligned}
$$

and this is quite good:

```r
x <- rbeta(10000, 2, 2)
y <- runif(10000, x, 1)
round(var(x/y), 4)
```

```
## [1] 0.0561
```

7.10.3.6 Example (8.10.10) Let (X, Y, Z) be a multivariate normal with mean vector (1, 1, 1) and variance-covariance matrix

$$\begin{bmatrix} 1 & 0.5 & 0.8 \\ 0.5 & 1 & -0.2 \\ 0.8 & -0.2 & 1 \end{bmatrix}$$

Find an approximation to $var(\sqrt{X^2+Y^2+Z^2})$

$$h(x,y,z) = \sqrt{x^2+y^2+z^2}$$
$$\frac{\partial h}{\partial x}(1,1,1) = \frac{x}{\sqrt{x^2+y^2+z^2}} = \frac{1}{\sqrt{3}}$$

and by symmetry the same holds for the other derivatives. So

$$var(\sqrt{X^2+Y^2+Z^2}) \approx$$
$$(\frac{1}{\sqrt{3}})^2 \left(\sum var(X_i) + 2\sum_{i<j} cov(X_i, X_j) \right) =$$
$$\frac{1}{3}(1+1+1+2(0.5+0.8-0.2)) = 1.7$$

```r
library(mvtnorm)
mu <- rep(1, 3)
vc <- cbind(c(1, 0.5, 0.8),
            c(0.5, 1, -0.2),
            c(0.8, -0.2, 1))
x <- rmvnorm(1e4, mu, vc)
var(sqrt(x[, 1]^2+x[, 2]^2+x[, 3]^2))
```

```
## [1] 1.098052
```

8 The Bootstrap

8.1 The Bootstrap - Introduction

8.1.0.1 Example (8.1.1) say we have rv's X where $X \sim N(\mu, \sigma)$ (σ known), and we are interested in estimating μ. We can use the sample mean $\bar{x}$. What is the standard error of this estimate? Of course it is $\sigma/\sqrt{n}$.

Let's instead say we want to use the median. Now what is the standard error? To find it we would first have to find the distribution of the sample median and then its variance.

Instead we can of course use simulation:

```r
B <- 10000; n <- 50; mu <- 2.5; sigma <- 1
x <- matrix(rnorm(B*n, mu, sigma), ncol=n)
xbar <- apply(x, 1, mean)
M <- apply(x, 1, median)
round(c(sigma/sqrt(n), sd(xbar), sd(M)), 3)
```

```
## [1] 0.141 0.144 0.177
```

Now let's say that the x_i's come from some unknown distribution $F(.;\theta)$. We can no longer do the simulation because we do not know what to simulate from. Instead we can use a method called the *bootstrap*.

It starts of very strangely: instead of sampling from a distribution as in a standard Monte Carlo study, we will now **resample** the data itself, that is if the data is n observations $\mathrm{x_1}$, .., $\mathrm{x_n}$, then the bootstrap sample is also n numbers **with replacement** from $\mathrm{x_1}$, .., $\mathrm{x_n}$, that is x_1^* is any of the original $\mathrm{x_1}$, .., $\mathrm{x_n}$ with probability 1/n.

In any one **bootstrap sample** an original observation, say $\mathrm{x_1}$, may appear once, several times, or not at all.

8.1.0.2 Example (8.1.2) say the data is (5.1, 2.3, 6.4, 7.8, 4.6), then one possible bootstrap sample is (6.4, 4.6, 2.3, 6.4, 5.1).

Say we have a sample $\mathrm{x_1}$, .., $\mathrm{x_n}$ from some unknown distribution F and we wish to estimate some parameter $\theta = t(F)$. For this we find some estimate

$\hat{\theta} = s(\mathbf{x})$

How accurate is $\hat{\theta}$?

8.1.0.3 Example (8.1.3) $X_1, ..., X_n \sim F; \theta = E(X_1)$ so

$t(F) = \int xf(x)dx$

and

$s(\mathbf{x}) = \bar{X}$

8.1.0.4 Example (8.1.4) $X_1, ..., X_n \sim F; \theta = var(X_1)$, so

$t(F) = \int (x-\mu)^2 f(x)dx$

and

$s(\boldsymbol{x}) = 1/(n-1)\sum(x_i - \bar{x})^2$

Here is the algorithm to find the bootstrap estimate of the standard error in $\hat{\theta}$:

1) Select B independent bootstrap samples $\mathbf{x}_1^*, .., \mathbf{x}_B^*$, each consisting of n data values drawn with replacement from $\mathbf{x}$. Here B is usually on the order 2000.
2) Evaluate the bootstrap replication corresponding to each bootstrap sample, $\hat{\theta}_b^* = s(\mathbf{x}_b^*)$, b=1,..,B
3) Estimate the standard error $se_f(\hat{\theta})$ by the sample standard deviation of the bootstrap replications.

8.1.0.5 Example (8.1.5) say the data is (5.1, 2.3, 6.4, 7.8, 4.6) and we want to estimate the mean μ, then

```
x <- c(5.1, 2.3, 6.4, 7.8, 4.6)
B <- 500
thetastar <- rep(0, B)
for(i in 1:B) {
  xstar <- sample(x, size=length(x), replace = TRUE)
                              # with replacement!
  thetastar[i] <- mean(xstar)
}
sd(thetastar)
```

```
## [1] 0.8179605
```

R has a library to do most of the work for us:

```
library(bootstrap)
sd(bootstrap(x, 2000, mean)$thetastar)
```

```
## [1] 0.8308119
```

8.1.0.6 Example (8.1.6) say the following is a sample from some distribution F:

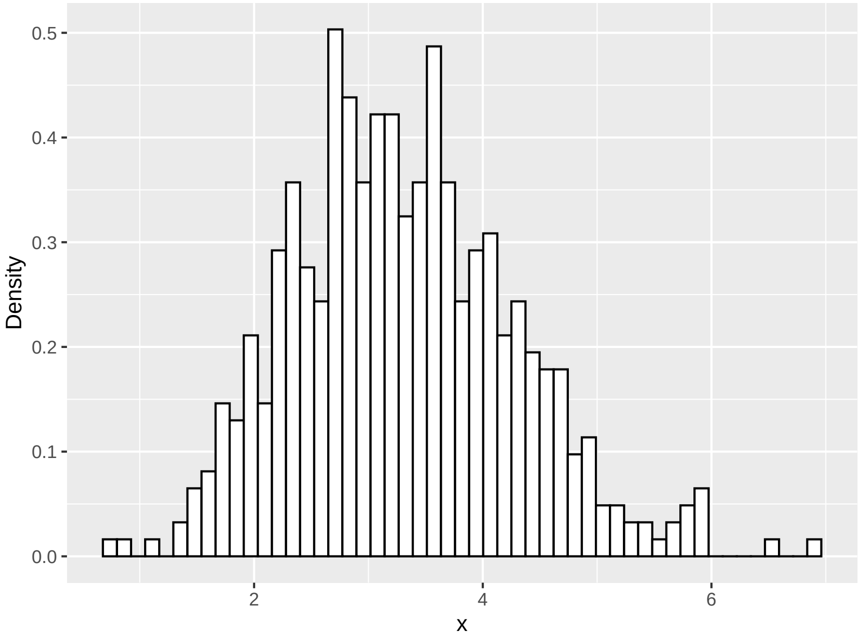
```
cat(x[c(1:10, 491:500)])
```

```
0.76 0.92 1.11 1.34 1.35 1.44 1.51 1.52 1.52 1.58 5.72 5.79 5.81 5.85 5.89 5.91 5.94 5.9
```

and we want to find 95% confidence intervals for $\theta = E[X]$.

Now this distribution is not a normal:

```
bw <- diff(range(x))/50
ggplot(data.frame(x=x), aes(x)) +
  geom_histogram(aes(y = ..density..),
    color = "black",
    fill = "white",
    binwidth = bw) +
    labs(x = "x", y = "Density")
```

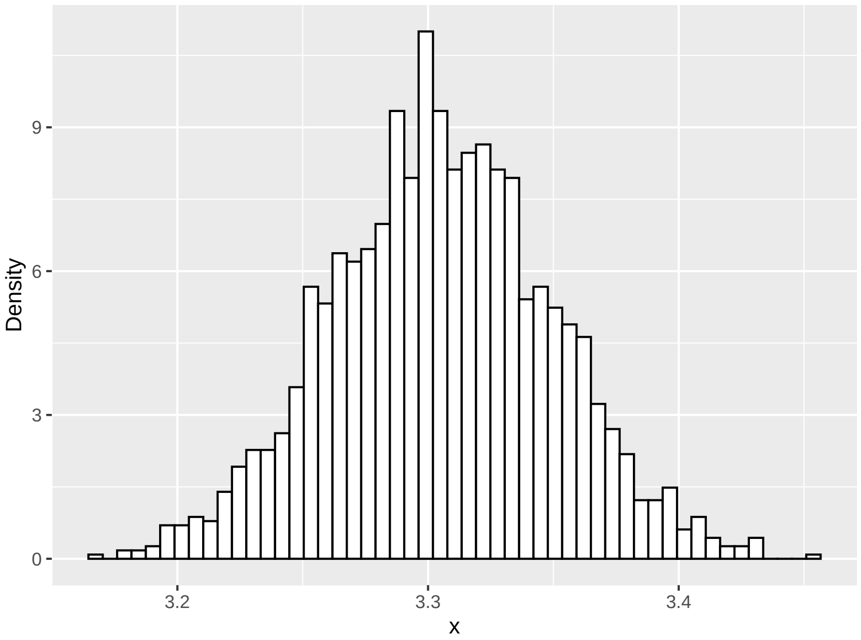


but we can use the bootstrap instead:

```
thetastar <- bootstrap(x, 2000, mean)$thetastar
```

Notice that in fact the thetastar's do have a normal distribution:

```
bw <- diff(range(thetastar))/50
ggplot(data.frame(x=thetastar), aes(x)) +
  geom_histogram(aes(y = ..density..),
    color = "black",
    fill = "white",
    binwidth = bw) +
    labs(x = "x", y = "Density")
```



and so we can find the confidence interval:

```
round(mean(x) +c(-1, 1)*qnorm(0.975)*sd(thetastar), 2)
```

```
## [1] 3.22 3.39
```

Note there is no $\sqrt{n}$ because sd(thetastar) is already the standard error of the estimate, not of the original data.

Here the bootstrap estimates are normal, which is quite often the case. If they are not we could use a CI based on the percentiles:

```
round(quantile(thetastar, c(0.025, 0.975)), 2)
```

```
##  2.5% 97.5%
##  3.22  3.39
```

This idea of the bootstrap is very strange: at first it seems we are getting more out of the data than we should. It is also a fairly new idea, invented by Bradley Efron in the 1980's.

Here is some justification why it works:

Let's say we have $X_1, ..., X_n \sim F$ for some cdf F, and we want to investigate the properties of some parameter θ of F, for example its mean or its median. We have an estimator of θ, say $s(x_1, .., x_n)$, for example $\bar{x}$ in the case of the mean.

What is the error in $s(x_1, .., x_n)$? In the case of the mean this is very easy and we already know that the answer is $sd(x_1)/\sqrt{n}$.

But what if we don't know it and we want to use Monte Carlo simulation to find out? Formally what this means is the following:

1) generate $X_1', ..., X_n' \sim F$
2) find the $\theta' = s(x_1', .., x_n')$

3) repeat 1 and 2 many times (say 1000 times)

4) Study the MC estimates of θ, for example find their standard deviation.

But what do we do if we don't know that our sample came from F?

A simple idea then is to replace sampling from the actual distribution function by sampling from the next best thing, the empirical distribution function. So the idea of the bootstrap is simple: replace F in the simulation above with Fhat:

1) generate $X_1', ..., X_n' \sim \hat{F}$

2) find the $\theta' = s(x_1', .., x_n')$

3) repeat 1 and 2 many times (say 1000 times)

4) Study the MC estimates of θ, for example find their standard deviation.

What does it mean, generate $X_1', .., X_n'$ from the empirical distribution function of $X_1, .., X_n$? Actually it means finding a bootstrap sample as described above.

8.1.0.7 Example (8.1.7) Let's return to the experiment on mice we discussed at the beginning of the class. Below we have the results of a small experiment, in which 7 out of 16 mice were randomly selected to receive a new medical treatment, while the remaining 9 mice were assigned to the control group. The treatment was intended to prolong survival after surgery:

	Treatment	Control
1	94	52
2	197	104
3	16	146
4	38	10
5	99	50
6	141	31
7	23	40
8		27
9		46

How can we answer the question on whether this new treatment is effective? First of course we can find the within group means and standard deviations:

```
round(unlist(lapply(mice, mean)), 1)
```

```
## treatment   control
##      86.9      56.2
```

```
round(unlist(lapply(mice, sd))/sqrt(unlist(lapply(mice, length))), 2)
```

```
## treatment   control
##     25.24     14.14
```

so we see that the mice who received the treatment lived on average 30.63 days longer. But unfortunately the standard error of the difference is $28.93 = \sqrt{(25.24^2 + 14.14^2)}$, so we see that the observed difference 30.63 is only $30.63/28.93 = 1.05$ standard deviations above 0.

Let's say next that instead of using the mean we wish to use the median to measure average survival. We find the following:

```
round(unlist(lapply(mice, median)), 1)
```

```
## treatment   control
##        94        46
```

Now we get a difference in median survival time of 48 days, but what is the standard error of this estimate? Of course there is a formula for the standard error of the median, but it is not simple and just finding it in a textbook would be some work. On the other hand we can use the bootstrap method to find it very easily:

```
treatstar <- bootstrap(mice$treatment, 1000, median)$thetastar
contstar <- bootstrap(mice$control, 1000, median)$thetastar
sds <- round(c(sd(treatstar), sd(contstar)), 2)
sds
```

```
## [1] 35.61 12.44
```

```
sqrt(sum(sds^2))
```

```
## [1] 37.72036
```

```
round(48/sqrt(sum(sds^2)), 3)
```

```
## [1] 1.273
```

So the difference is 1.15 standard deviations larger than 0.

This is larger than the one for the mean, but still not statistically significant.

8.2 Some Applications of the Bootstrap

```
library(bootstrap)
```

Let's revisit some of the examples we discussed earlier, and analyze them using the bootstrap:

8.2.0.1 Example (8.2.1)

In a survey of 1000 likely voters 523 said they will vote for party A, the other 477 for party B. Find a 95% CI for the **lead** of one party over the other.

```
B <- 10000
x <- c(rep(1, 523), rep(0, 477))
lead <- abs(sum(x) - sum(1 - x))/1000
lead.boot <- rep(0, B)
for (i in 1:B) {
  xstar <- sample(x, size = 1000, replace = T)
  lead.boot[i] <- abs(sum(xstar) - sum(1 - xstar))/1000
}
cat("Bootstrap Estimate")
```

```
## Bootstrap Estimate
```

```
as.numeric(quantile(lead.boot, c(0.025, 0.975)))
```

```
## [1] 0.002 0.106
```

```
x <- 523; n <- 1000; alpha <- 0.05
cat("Likelihood Ratio")
```

```
## Likelihood Ratio
```

```
round(2*x/n-1 +
        c(-1, 1)*2*qnorm(1-alpha/2)*sqrt(x/n*(1-x/n)/n), 3)
```

```
## [1] -0.016  0.108
```

8.2.0.2 Example (8.2.2)

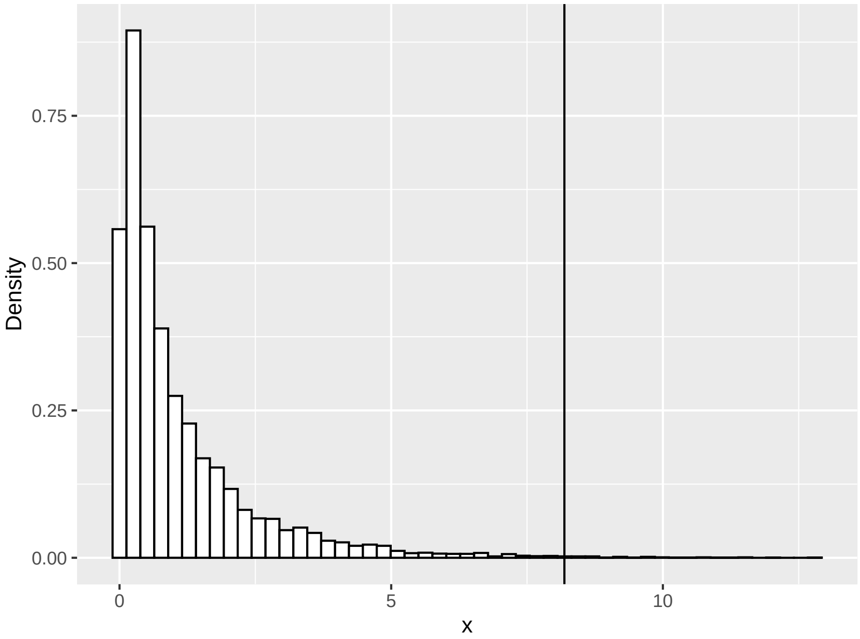
Recall the pet ownership and survival data:

	Status	Alive	Dead
1	Owns a Pet	50	3
2	Does not own a Pet	28	11

Question: is there a statistically significant association between Ownership and Survival?

First we need a *measure of association*, that is a number we can calculate from the data that tells us something about the relationship (or lack thereof) between our variables. Previously we used the chisquare statistic, so let's use it again. Only now we don't need the chisquare approximation, we can just study the distribution of the chisquare values for the bootstrap sample.

```
z <- c(50, 28, 3, 11)
A <- matrix(0, 92, 2)
A[79:92, 1] <- 1
A[c(51:78, 82:92), 2] <- 1
O <- c(table(A[, 1], A[, 2]))
E <- c((O[1]+O[2])*(O[1]+O[3]),
       (O[1]+O[2])*(O[2]+O[4]),
       (O[1]+O[3])*(O[3]+O[2]),
       (O[2]+O[4])*(O[3]+O[4]))/92
chi <- sum((O-E)^2/E)
chistar <- rep(0, B)
for (i in 1:B) {
  a1 <- A[sample(1:92, size=92, replace=TRUE), 1]
  a2 <- A[sample(1:92, size=92, replace=TRUE), 2]
  O <- c(table(a1, a2))
  E <- c((O[1]+O[2])*(O[1]+O[3]),
       (O[1]+O[2])*(O[2]+O[4]),
       (O[1]+O[3])*(O[3]+O[2]),
       (O[2]+O[4])*(O[3]+O[4]))/92
  chistar[i] <- sum((O-E)^2/E)
}
bw <- diff(range(chistar))/50
ggplot(data.frame(x=chistar), aes(x)) +
  geom_histogram(aes(y = ..density..),
    color = "black",
    fill = "white",
    binwidth = bw) +
    labs(x = "x", y = "Density") +
  geom_vline(xintercept = chi)
```



```
length(chistar[chistar > chi])/B
```

```
## [1] 0.0036
```

Here we take bootstrap samples of the *owns* and of *survives* **independently**, calculate the corresponding chisquare statistics and repeat that B times. Finally we find the percentage of bootstrap runs larger than the observed chisquare, which is essentially the p-value of our test.

8.2.0.3 Example (8.2.3) the effect of the mother's cocaine use on the length of the newborn.

Let's use the bootstrap to find 95% confidence intervals for the means and medians of the lengths.

```
A <- matrix(0, 3, 4)
Status <- c("Drug Free", "First Trimester", "Throughout")
dimnames(A) = list(Status,
      c("Low Mean", "High Mean", "Low Median", "High Median"))
for (i in 1:3) {
  x <- mothers$Length[mothers$Status==Status[i]]
  thetastar <- bootstrap(x, 1000, mean)$thetastar
  A[i, 1:2] = mean(x) + c(-1, 1)*qnorm(0.975)*sd(thetastar)
  thetastar <- bootstrap(x, 1000, median)$thetastar
  A[i, 3:4] = median(x) + c(-1, 1)*qnorm(0.975)*sd(thetastar)
}
A <- round(A, 1)
kable.nice(A)
```

	Low Mean	High Mean	Low Median	High Median
Drug Free	50.2	52.0	50.6	52.0
First Trimester	48.2	50.4	47.6	50.2
Throughout	46.8	49.2	47.2	49.1

```
head(hubble)
```

8.2.0.4 Example (8.2.4)

```
##   Velocity Distance Galaxy.NGC. velocity.km.s Distance.Mpc.
## 1      170    0.032         925           553          9.70
## 2      290    0.034       1326A          1831         15.81
## 3     -130    0.214        1365          1636         18.48
## 4      -70    0.263        1425          1510         20.83
## 5     -185    0.275        2090           921         11.57
## 6     -220    0.275        2541           548         12.06
```

```
B <- 1000; alpha = 0.05
coef <- lm(Velocity~Distance, data=hubble)$coef
print("coefficients for data")
```

```
## [1] "coefficients for data"
```

```
print(as.numeric(coef))
```

```
## [1] -40.78365 454.15844
```

```
coef.boot <- matrix(0, B, 2)
for (i in 1:B) {
  Index <- sample(1:24, size=24, replace = TRUE)
  coef.boot[i, ] <- lm(Velocity~Distance,
    data=hubble[Index, ])$coef
}
print("CI for constant:")
```

```
## [1] "CI for constant:"
```

```
print(as.numeric(quantile(coef.boot[, 1], c(alpha/2, 1 -
        alpha/2))))
```

```
## [1] -206.9136  118.0921
```

```
print("CI for slope:")
```

```
## [1] "CI for slope:"
```

```
print(as.numeric(quantile(coef.boot[, 2], c(alpha/2, 1 -
        alpha/2))))
```

```
## [1] 325.9313 610.3808
```

8.2.0.5 Example (8.2.5)

Consider Gregor Mendel's pea experiment:

```
O <- c(315, 101, 108, 32)
B <- 1000
E <- sum(O)*c(9, 3, 3, 1)/16
chi <- sum((O-E)^2/E)
print("Chisquare Statistic of Data")
```

```
## [1] "Chisquare Statistic of Data"
```

```
print(chi)
```

```
## [1] 0.470024
```

```
chi.boot <- rep(0, B)
x <- rep(1:4, E)
for (i in 1:B) {
  xstar <- sample(x, size = 556, replace = TRUE)
  chi.boot[i] = sum((table(xstar) - E)^2/E)
}
print("% of Bootstrap runs > Data")
```

```
## [1] "% of Bootstrap runs > Data"
```

```
length(chi.boot[chi.boot > chi])/B
```

```
## [1] 0.927
```

8.2.0.6 Example (8.2.6) Hidalgo stamps

A well known data set in statistics has the thicknesses (espesor) in millimeters of 485 Mexican stamps (sello) printed in 1872-1874, from the 1872 Hidalgo issue.

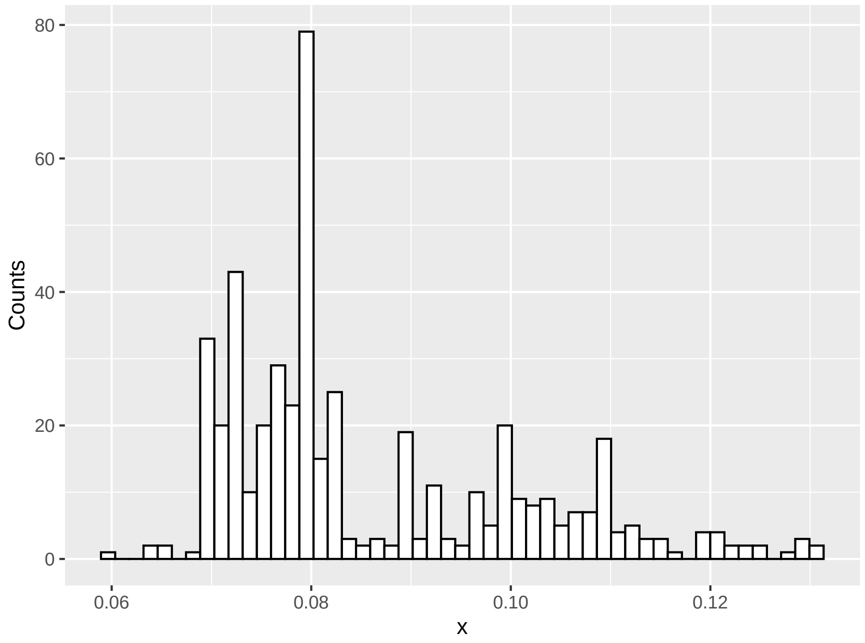
It is thought that the stamps from this issue are a "mixture" of different types of paper, of different thicknesses. Can we determine from the data how many different types of paper were used?

```
kable.nice(matrix(stamps[1:50], nrow=10))
```

0.060	0.069	0.07	0.070	0.071
0.064	0.069	0.07	0.070	0.071
0.064	0.069	0.07	0.070	0.071
0.065	0.070	0.07	0.070	0.071
0.066	0.070	0.07	0.070	0.071
0.068	0.070	0.07	0.070	0.071
0.069	0.070	0.07	0.070	0.071
0.069	0.070	0.07	0.070	0.071
0.069	0.070	0.07	0.070	0.071
0.069	0.070	0.07	0.071	0.071

Let's start with

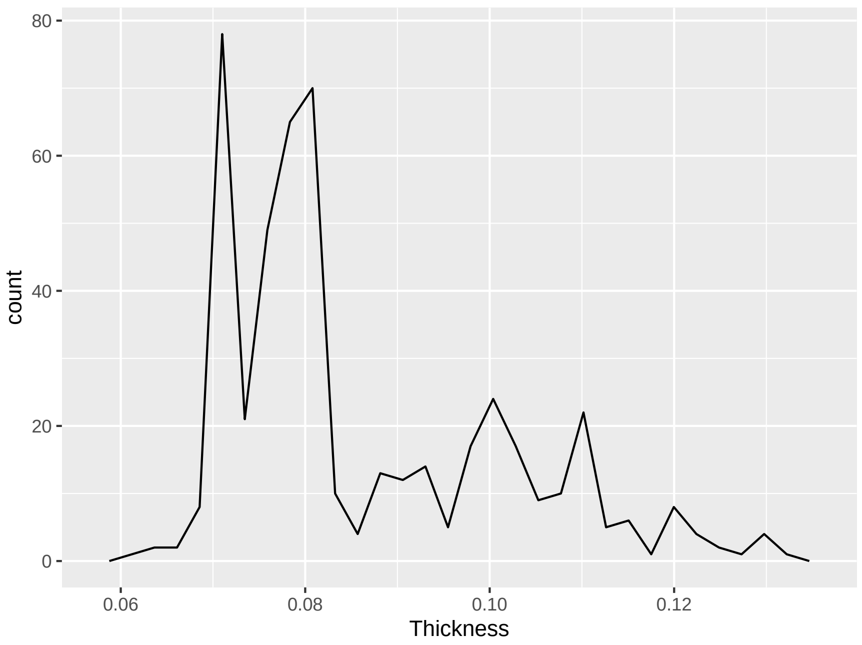
```
bw <- diff(range(stamps))/50
df <- data.frame(Thickness=stamps)
ggplot(df, aes(Thickness)) +
geom_histogram(color = "black",
    fill = "white",
    binwidth = bw) +
    labs(x = "x", y = "Counts")
```



which seems to have at least two modes. This judgment however is tricky because it depends on the number of bins we use.

An alternative is to use a frequency polygon

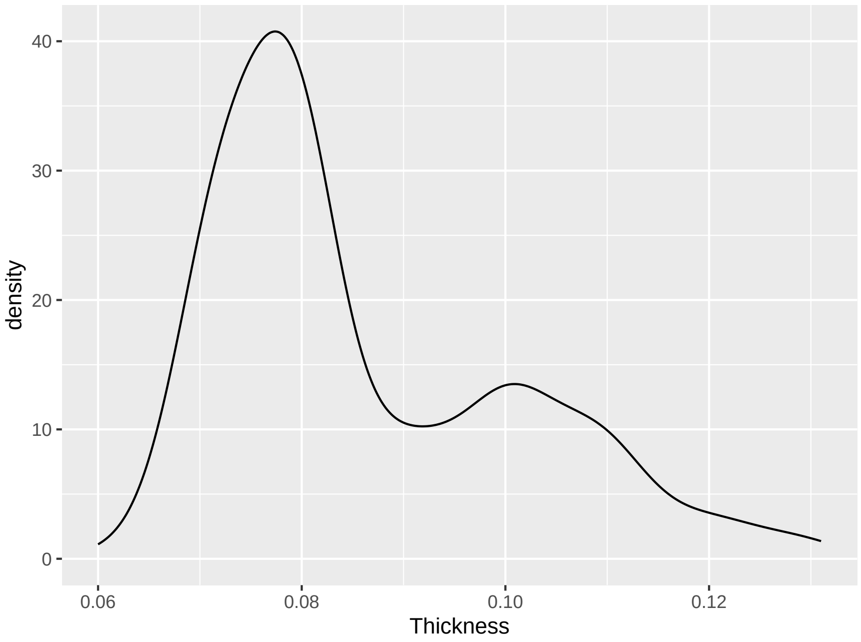
```
ggplot(df, aes(Thickness)) +
  geom_freqpoly()
```



which seems to suggest a much larger number of modes.

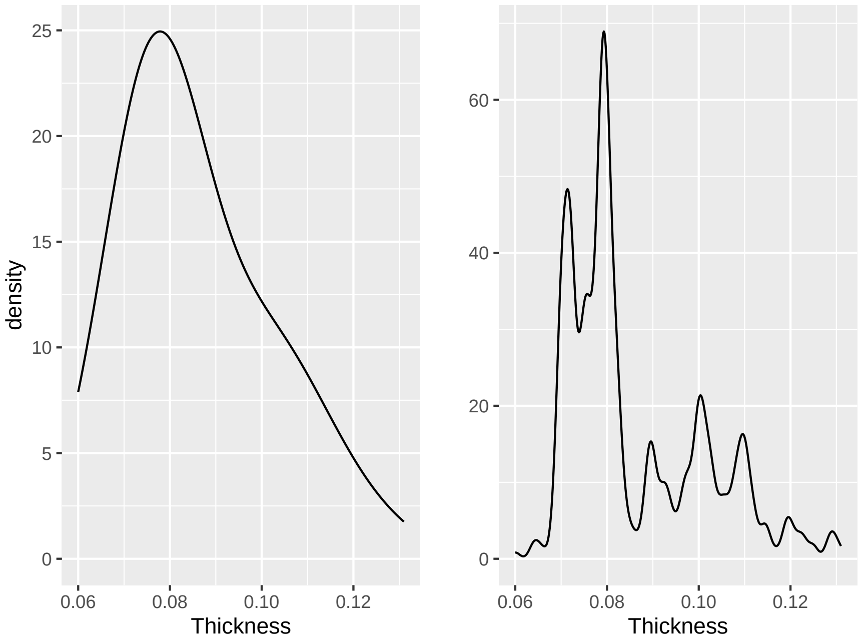
Let's instead draw the graph using a nonparametric density estimate:

```
ggplot(df, aes(Thickness)) +
  stat_density(geom="line")
```



here it seems again like there are two modes, but this depends largely on the chosen bandwidth:

```
pushViewport(viewport(layout = grid.layout(1, 2)))
print(ggplot(df, aes(Thickness)) +
   stat_density(geom="line", bw=0.01)   ,
  vp=viewport(layout.pos.row=1, layout.pos.col=1))
print(ggplot(df, aes(Thickness)) +
   stat_density(geom="line", bw=0.001) +ylab("")   ,
  vp=viewport(layout.pos.row=1, layout.pos.col=2))
```



stat_density implements a *kernel density estimator*. In what follows we will need to explicitly calculate these estimates and use the *density* routine.

From the above it is clear that the number of modes depends on the choice of h. It is possible to show that the number of modes is a non-increasing function of h. At the extremes we would have a simple normal distribution with one mode (h large) and on the other a sharply peaked mode at each observation (h tiny).

Let's say we want to test

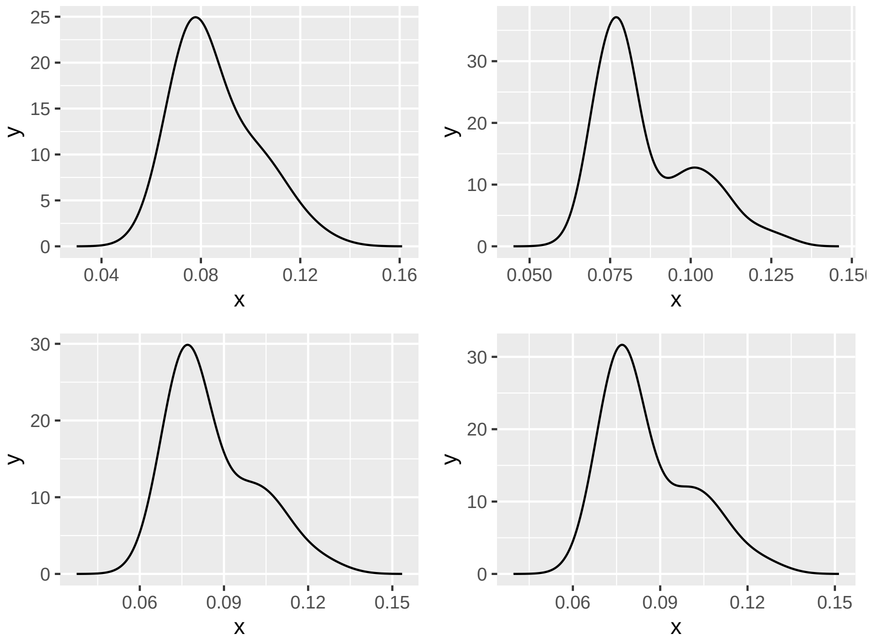
$$H_0 : \text{number of modes} = 1 \text{ vs. } H_1 : \text{number of modes} > 1$$

Because the number of modes is a non-increasing function of h there exists an h_1 such that the density estimator has one mode for $h < h_1$ and two or more modes for $h > h_1$. Playing around with

```
fhat <- function(h, t, dta=stamps) {
  tmp <- density(dta, bw=h)
  df <- data.frame(x=tmp$x, y=tmp$y)
  if(missing(t)) return(df)
  out <- approx(df, xout=t)$y
  out[!is.na(out)]
}
draw.fhat <- function(h)
  ggplot(fhat(h), aes(x, y)) + geom_line()
```

```
pushViewport(viewport(layout = grid.layout(2, 2)))
print(draw.fhat(0.01) ,
  vp=viewport(layout.pos.row=1, layout.pos.col=1))
```

```
print(draw.fhat(0.005) ,
  vp=viewport(layout.pos.row=1, layout.pos.col=2))
print(draw.fhat(0.0075) ,
  vp=viewport(layout.pos.row=2, layout.pos.col=1))
print(draw.fhat(0.0068) ,
  vp=viewport(layout.pos.row=2, layout.pos.col=2))
```



we find $h_1 \sim 0.0068$.

Is there a way to calculate the number of modes for a given h? here is one:

- calculate $y_i = \hat{f}(t_i; h)$ on a grid $t_1, ..t_k$
- calculate $z_i = y_i - 1 - y_i$ and note that at a mode z will change from positive to negative
- number of modes $= \sum I[z_i > 0 \text{ and } z_{i+1} < 0]$

Let's write a simple routine that automates the process. It uses a bisection algorithm.

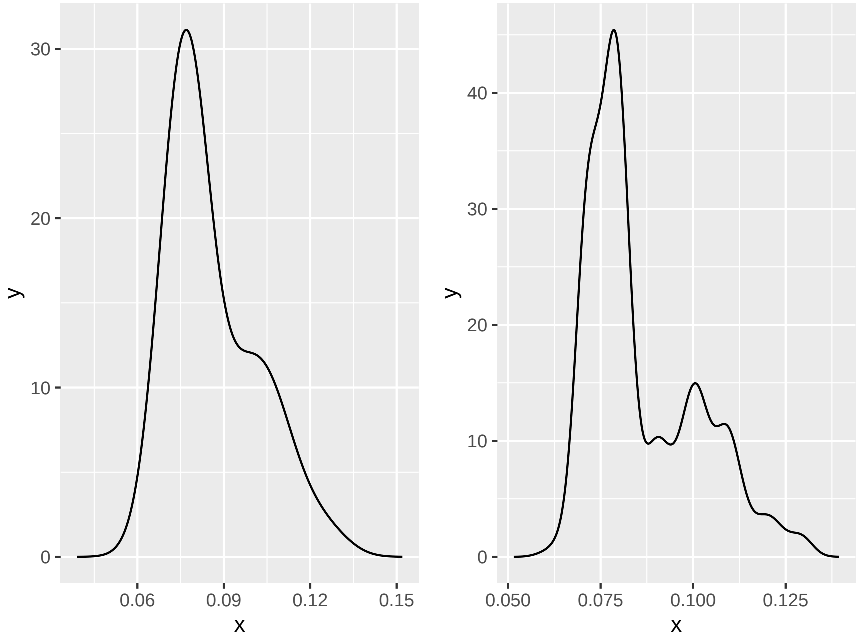
```
x.points <- seq(min(stamps), max(stamps), length = 250)
calc.num.mode = function(y) {
  m <- length(y) - 1
  z <- diff(y)
  sum(ifelse(z[-m] >= 0 & z[-1] < 0, 1, 0))
}
find.h <- function(num.modes, h=0.007, Show=FALSE) {
  repeat {
    h <- h-0.001
    if(Show)
      cat("h =", h, " modes=",
          calc.num.mode(fhat(h, x.points)), "\n")
    if(calc.num.mode(fhat(h, x.points)) >= num.modes)               break
  }
  low <- h
  high <- h + 0.001
  repeat {
    h <- (low+high)/2
    if(Show)
      cat("h =", h, " modes=",
          calc.num.mode(fhat(h, x.points)), "\n")
    if(calc.num.mode(fhat(h, x.points)) < num.modes)
      high <- h
    else
      low <- h
    if(high-low<10^-7)
      break
  }
  h
}
```

```
h1 <- find.h(1, Show = TRUE)
```

```
## h = 0.006  modes= 2
## h = 0.0065  modes= 2
## h = 0.00675  modes= 1
## h = 0.006875  modes= 1
## h = 0.0069375  modes= 1
## h = 0.00696875  modes= 1
## h = 0.006984375  modes= 1
## h = 0.006992188  modes= 1
## h = 0.006996094  modes= 1
## h = 0.006998047  modes= 1
## h = 0.006999023  modes= 1
## h = 0.006999512  modes= 1
## h = 0.006999756  modes= 1
## h = 0.006999878  modes= 1
## h = 0.006999939  modes= 1
```

```
h5 <- find.h(5)
```

```
pushViewport(viewport(layout = grid.layout(1, 2)))
print(draw.fhat(h1) ,
  vp=viewport(layout.pos.row=1, layout.pos.col=1))
print(draw.fhat(h5) ,
  vp=viewport(layout.pos.row=1, layout.pos.col=2))
```



So, how we can test

$$H_0 : \text{number of modes} = 1 \text{ vs. } H_a : \text{number of modes} > 1$$

Here it is:

- draw B bootstrap samples of size n from fhat(h1)
- for each find h_1^*, the smallest h for which this bootstrap sample has just 1 mode
- approximate p-value of test is the proportion of $h_1^* > h_1$.

the idea is this; if there is indeed just one mode, then in the bootstrap samples h_1^* should be around h_1 and so this proportions shouldn't be to small.

Notice we don't actually need h_1^*, we just need to check if $h_1^* > h_1$, which is the case if $\hat{f}(x^*; h_1^*)$ has at least two modes.

Note that we are not drawing bootstrap samples from "stamps" but from a density estimate, $\hat{f}$. So this is an example of the *smooth bootstrap* mentioned above.

How do we draw from fhat? It can be shown that if $y_1^*, .., y_n^*$ is a bootstrap sample from the data, then a smooth bootstrap sample is given by

$$x_i^* = \bar{y}^* + (1 + h_1^*/s^2)^{-1/2}(y_i^* - \bar{y}^* + h_1^* \epsilon_i)$$

where $\epsilon_i \sim N(0, 1)$

```
test.modes <- function(k) {
  h <- find.h(k+1)
  q <- 1/sqrt((1 + h^2/var(stamps)))
  B <- 1000
  hstar <- rep(0, B)
  for (i in 1:B) {
    ystar <- sample(stamps, size = 485, replace = TRUE)
    xstar <- mean(ystar) + q*(ystar-mean(ystar) +
                h*rnorm(485))
    y <- fhat(h, x.points, dta=xstar)
    if (calc.num.mode(y) > k)
        hstar[i] <- 1
  }
  length(hstar[hstar > h])/B
}
test.modes(1)
```

```
## [1] 0.001
```

and so we find strong evidence against the null, there are more than one modes.

The same method works for testing

$$H_0 : \text{number of modes} = k \text{ vs. } H_a : \text{number of modes} > k$$

and we find

```
for(k in 2:9)
  cat("k =", k, ", p =", test.modes(k),"\n")
```

```
## k = 2 , p = 0.313
## k = 3 , p = 0.058
## k = 4 , p = 0.006
## k = 5 , p = 0.001
## k = 6 , p = 0
## k = 7 , p = 0.332
## k = 8 , p = 0.781
## k = 9 , p = 0.543
```

So there are certainly more than one mode, with a chance for as many 7.