Una Introduccion a R

Edgar Acuna

Computational and Statistical Learning Group

Departament of Mathematics

University of Puerto Rico at Mayaguez

Enero 2007

Contenido

I. Introduction
II. Operaciones aritmeticas basicas
III. Funciones estadisticas basicas
IV. Graficas
V. Matrices
VI. Leyendo datos
VII. Usando la ayuda de R
VIII. Programando funciones
IX. Rcmdr: Un GUI mejorado para R.
X. Haciendo paquetes

I. Introduccion

- R (Ihaka and Gentleman, 1994) es una implementacion gratuita del programa de computacion estadistica, S, el cual se origino a principios de los 80's. S-Plus una implementacion comercial de S que incluye un interface grafico(GUI), esta disponible desde los inicios de los 90's.
- R mayormente usa comandos de linea e incluye un limitado GUI. Hay varias propuestas para GUI's en R, siendo Rcmdr el mas usado.
- R tiene excelente capacidades de graficas.
- R esta disponible para Unix/Linux, MacOs y Windows.

Porque usar R?

- Los metodos estadisticos desarrollados mas recientemente son programados primero en R.
- Existen muchas librerias disponibles para aplicar diversos metodos estadisticos.
- Crea excelente graficas con relativa facilidad.
- Es facil de usar.
- Puede leer datos de diferente sistemas de bases de datos (SQL, Oracle, etc).
- Es gratis.

Obteniendo R

1-Entrar al Website:www.r-project.org y escoger CRAN que aparece en el panel izquierdo.

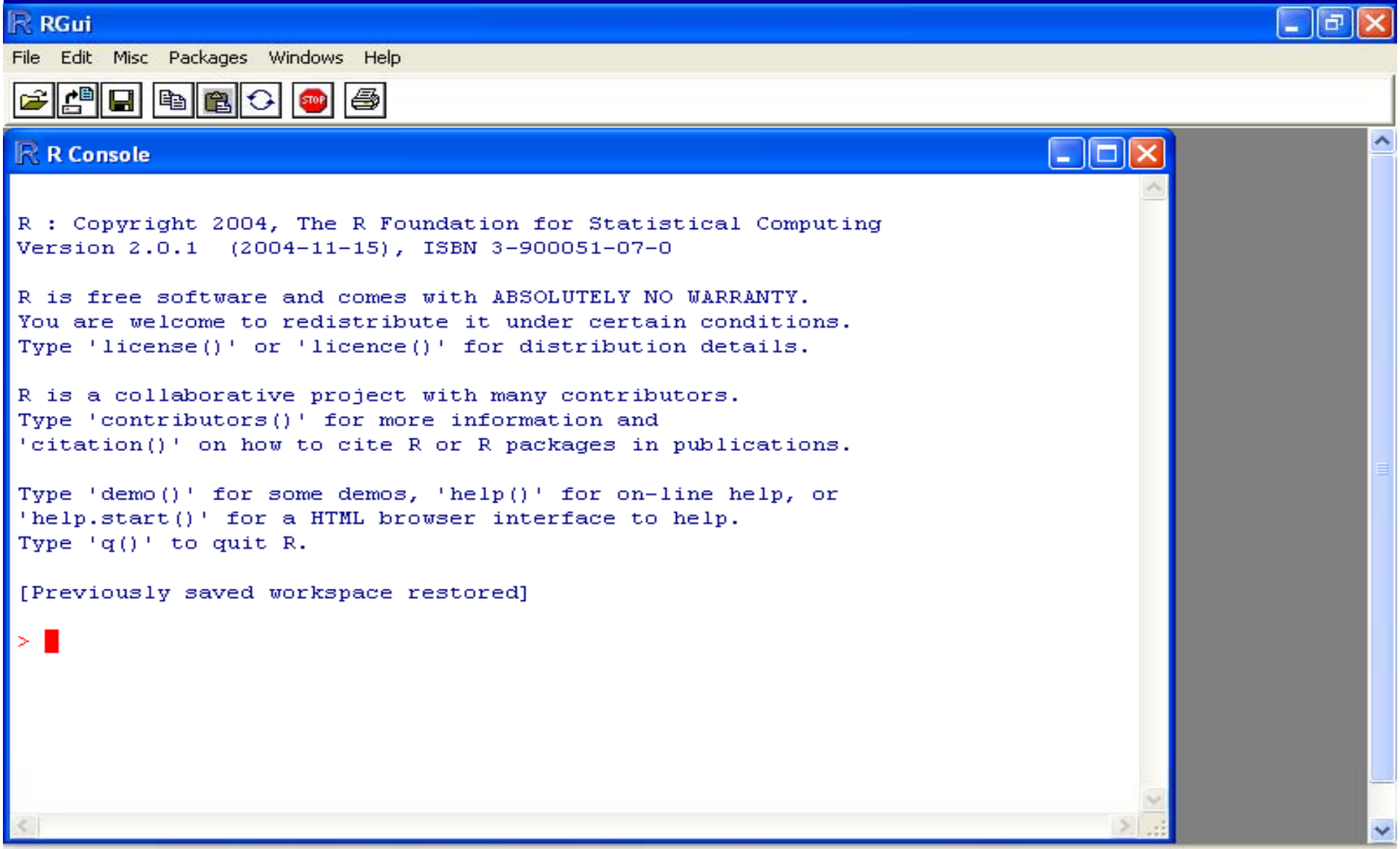
2-Escoger el sistema operativo en donde va a usar R: Linux, MacOs o Windows.

3. En la pantalla R for Windows escoger el subdirectorio base.

4. De la pantalla R-2.9.0 for Windows bajar a su computadora el archivo r-2.9.0-win32.exe.

5. Localizar el archivo r-2.9.0-win32.exe en su computadora y ejecutarlo, eso instalara la version 2.9.0 de R.

El ambiente grafico de R



II. Operaciones aritmeticas basicas

```
2+3 #Suma
2-3  #Resta
2*3  #Producto
2/3  # Division
2^3  #Potencia
2^-3 #Potencia
(4^2) - (3*2) #Operaciones combinadas
```

Funciones matematicas

```
>exp(3)
[1] 20.08554
> cos(pi)
[1] -1
> sin(pi/2)
[1] 1
> tan(pi/4)
[1] 1
log(100)
log(100, base=10)
log(100, b=2)

help(log)
```

vectores

```
x=c(1,2,3,4)  # combinar
x=1:4     # vector formado por una secuencia
x=rep(1,4) # vector de 4 unos
x=seq(2, 8, by=2) # vector:2,4,6,8
x=seq(0, 1, length=11) #vector 0,0.1,0.2……1
x[2]        # el segundo elemento del vector x
x[c(2,4,6)] # vector conteniendo los  elementos 2,4 y 6 de x
x[-c(1,3)]   # vector sin incluir los elementos 1 y 3 de x
x[x < 4]    # vector que contiene los elementos de x t.q x<4.
y=x/2
z=x+y
log(x, 10)
y = sqrt(x)
```

III. Funciones estadísticas basicas

```
x=c(18,24,17,23,23,21,19,18,24,21)

mean(x) #calcula la media

median(x) #calcula la mediana

var(x) #calcula la varianza de x

sd(x) #calcula la desviacion estandar

quantile(x,prob=c(.1,.9)) # percentiles del 10 y 90%
```

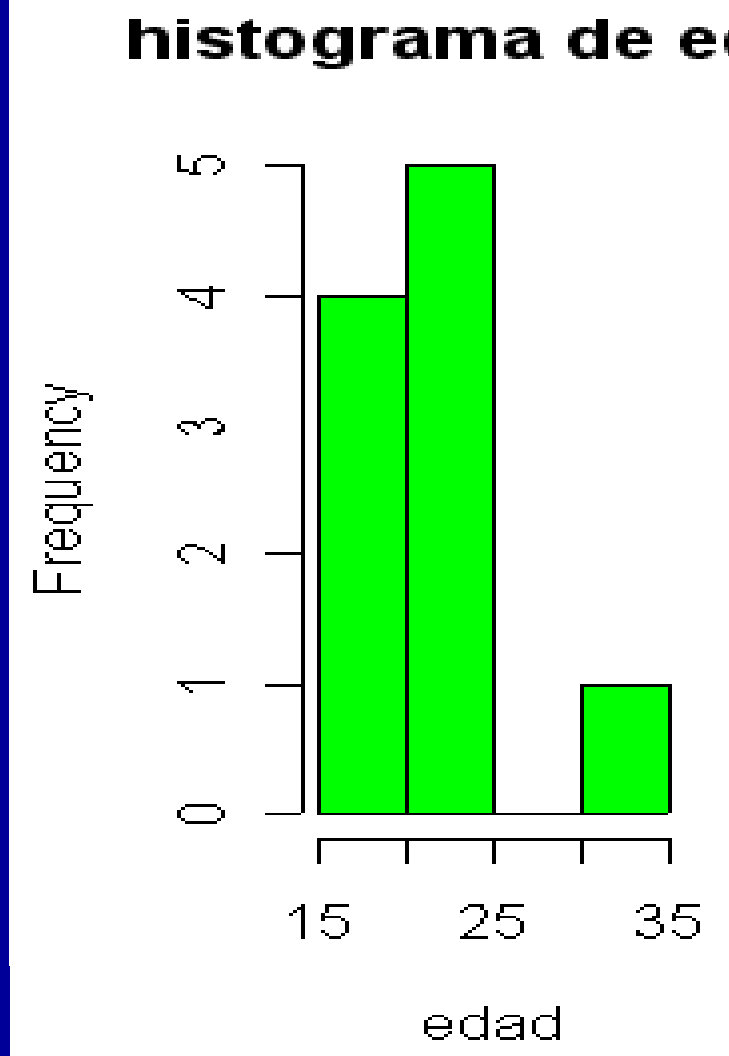
Funciones estadisticas basicas

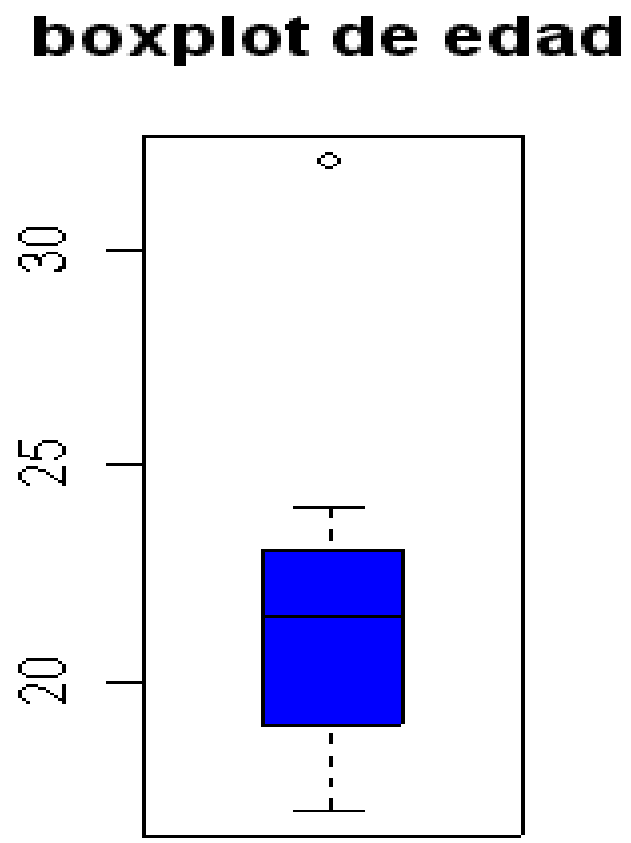
```
summary(x) # calcula varias medidas estadisticas
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  13.00   18.50   23.50   23.00   27.75   32.00

sort(x) #ordena los valores de x en forma creciente

sort(x,decreasing=T) # ordena en forma decreciente

table(x) #muestra las frecuencias absolutas de x

dnorm(x, mean=0, sd=1) #densidad normal
pnorm(15, mean=3, sd=4) #la normal acumulada
qnorm(.3, mean=4, sd=2) #percentiles de una normal
rnorm(20, mean=0, sd=1) # genera n valores de una
normal estandar
```

IV. Graficas estadisticas univariadas

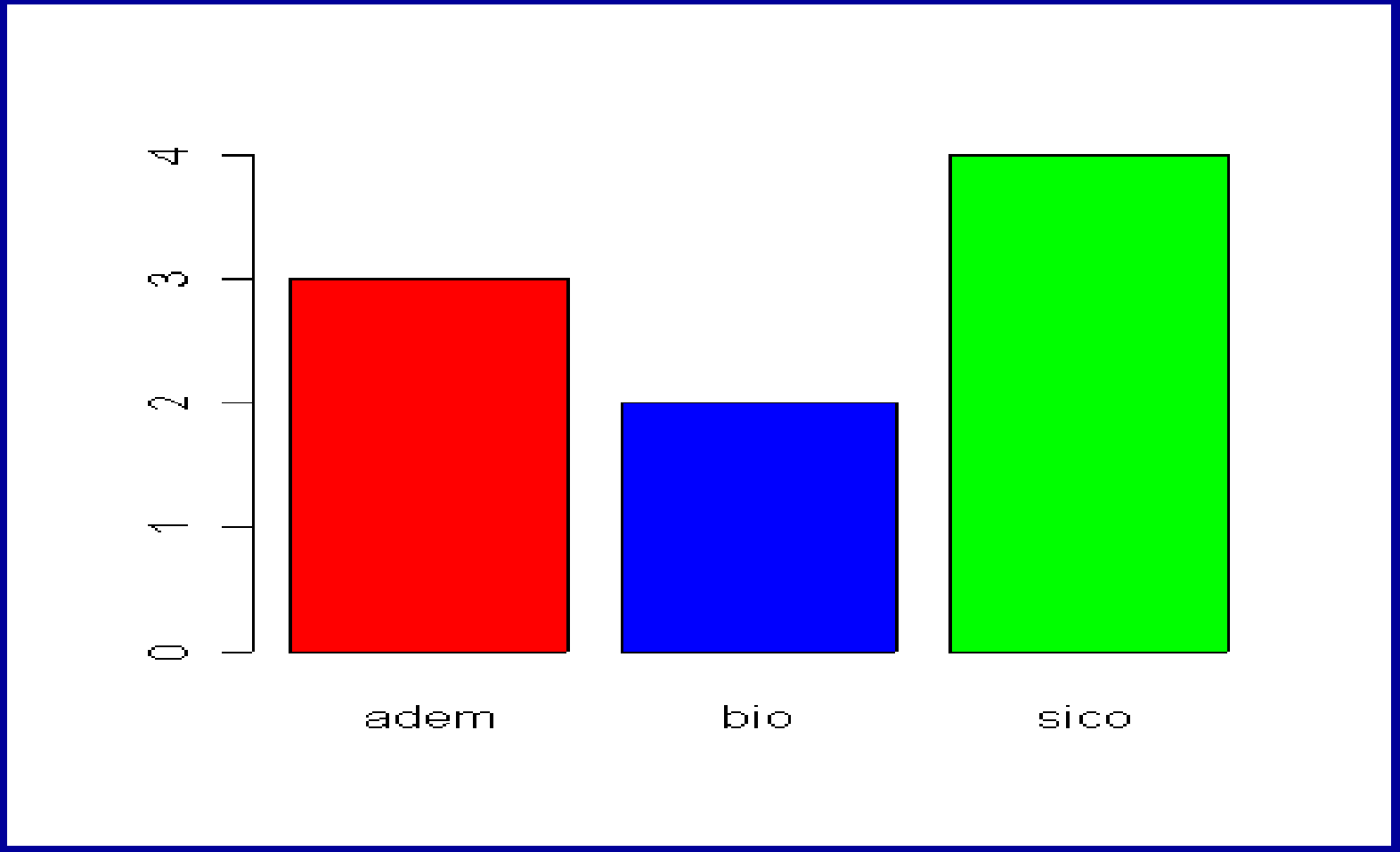
```
edad=c(18,24,19,23,22,32,17,21,23,20)

hist(edad) #hace un histogram

boxplot(edad,horizontal=T) #hace un diagrama de caja
#las dos figuras en la misma pantalla
par(mfrow=c(1,2))
hist(edad,main="histograma de edad",col="green")
boxplot(edad,main="boxplot de edad", col=4)

programas=c("bio","sico","adem","sico","bio","sico","ad
em","adem","sico")
```

Gráficas estadísticas univariadas



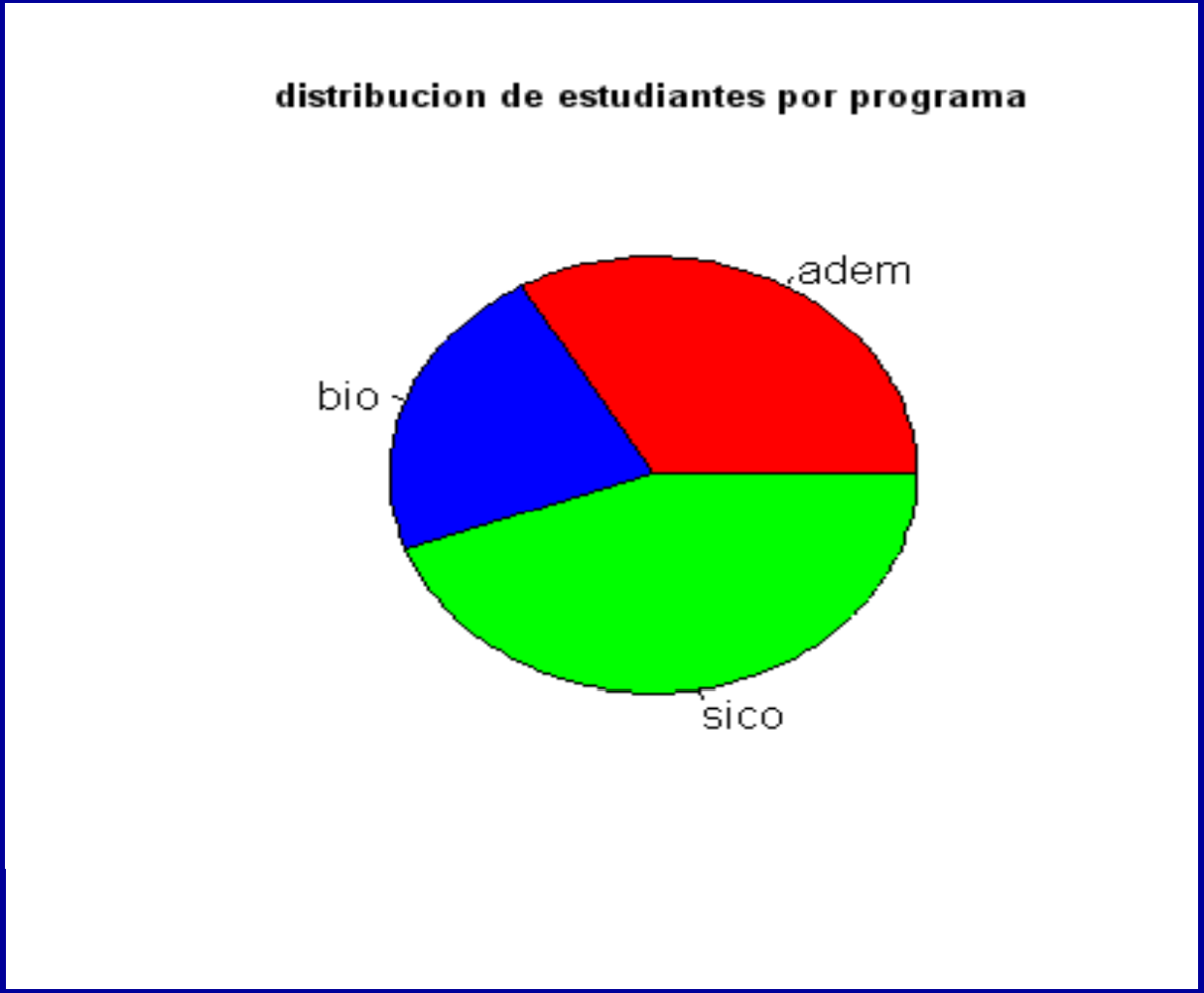


Grafica de Barras

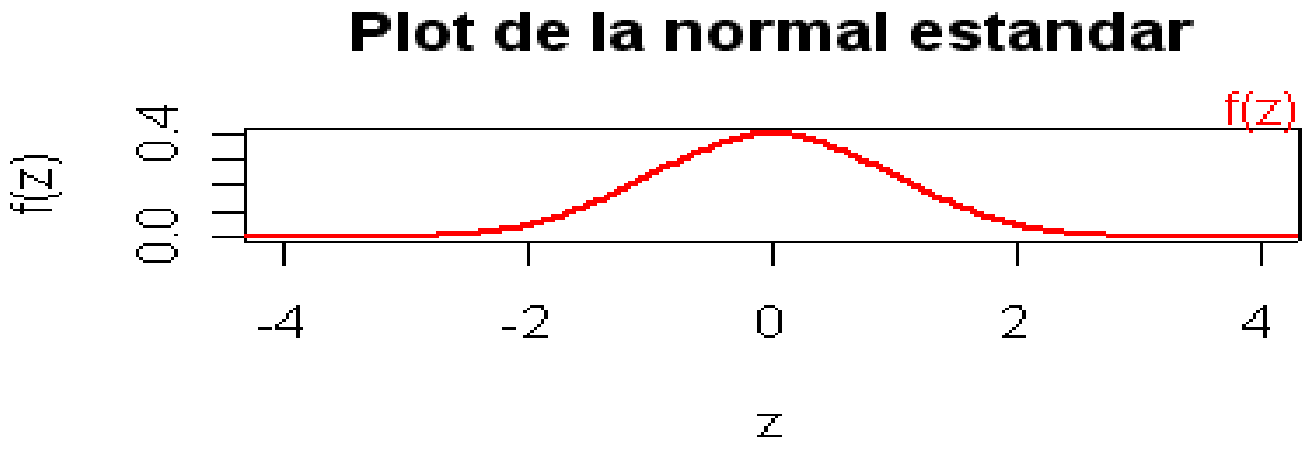


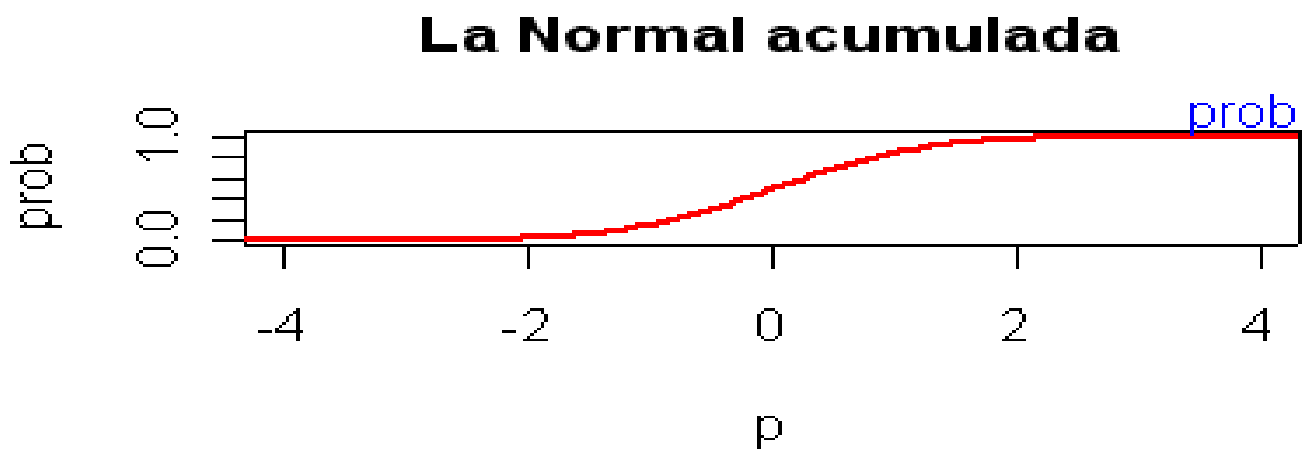

```
barplot(table(programas),col=c("red","blue","green"))
```

Pie-charts




```
pie(table(programas),col=c("red","blue","green"),main="distribucion de
estudiantes por programa",cex.main=.8)
```



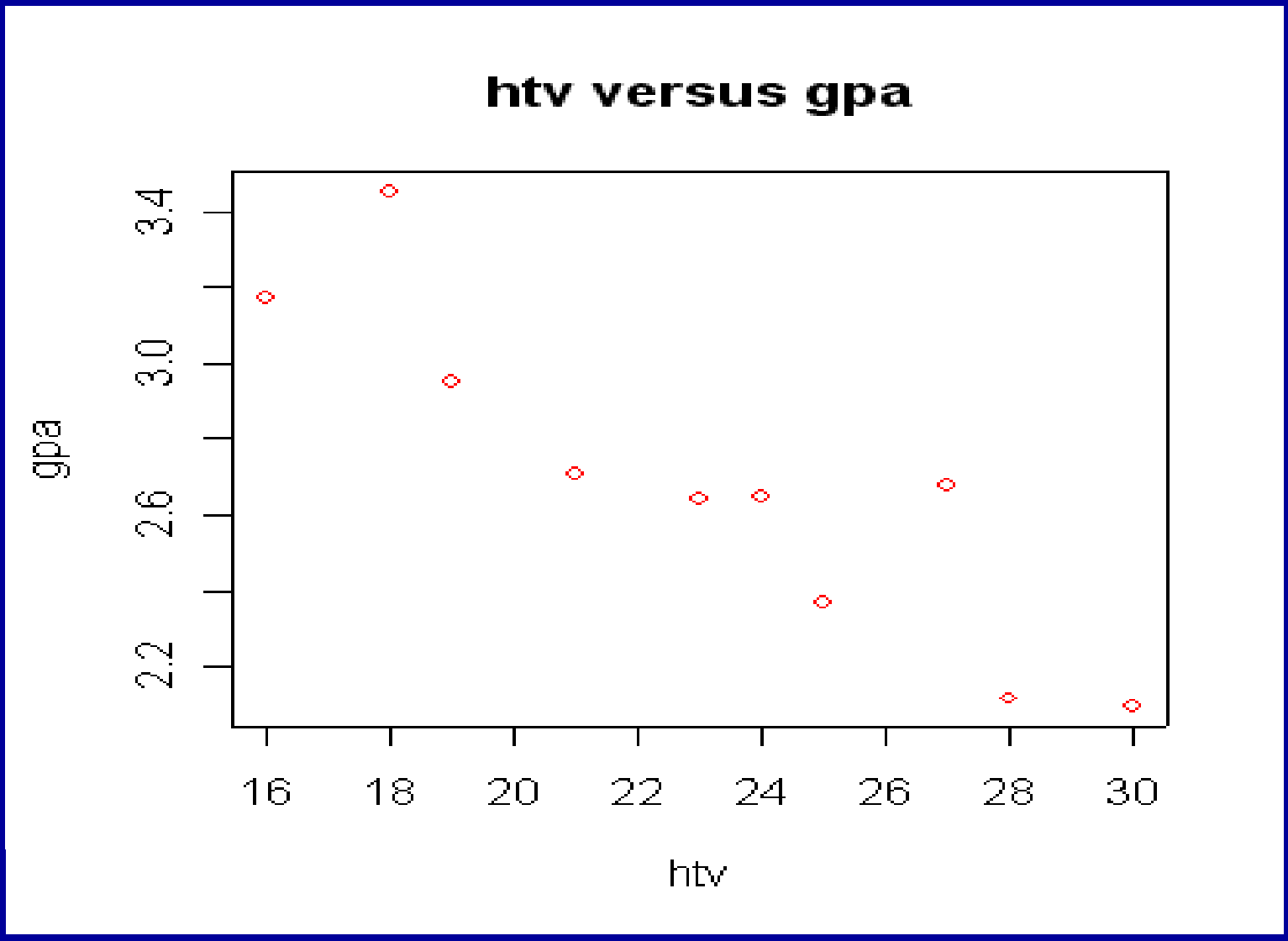



```
curve(dnorm(x),-4,4,col=2,lwd=4)
curve(pnorm(x),-4,4,col=2,lwd=4)
```

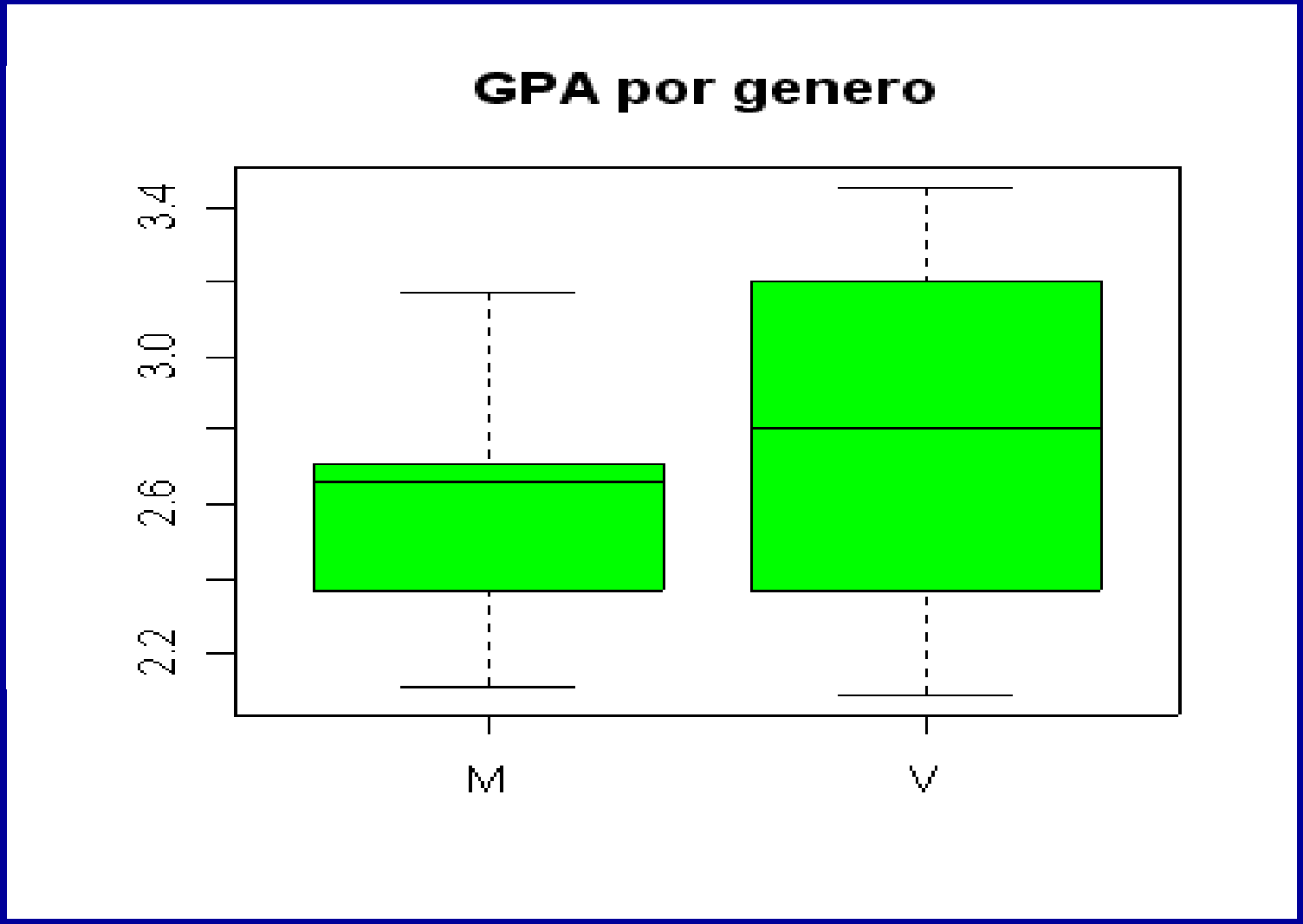
Grafica en dos dimensiones

```
htv=c(16,18,19,21,23,24,25,27,28,30)

gpa=c(3.17,3.45,2.95,2.71,2.64,2.65,2.37,2.68,2.11,2.09)

genero=c("M", "V", "V", "M", "M", "V", "M", "M", "M", "V")
plot(htv,gpa,main="htv versus gpa",col="red")
boxplot(gpa~as.factor(genero),col=c("red","green"))
title("GPA por genero")
```

Plot

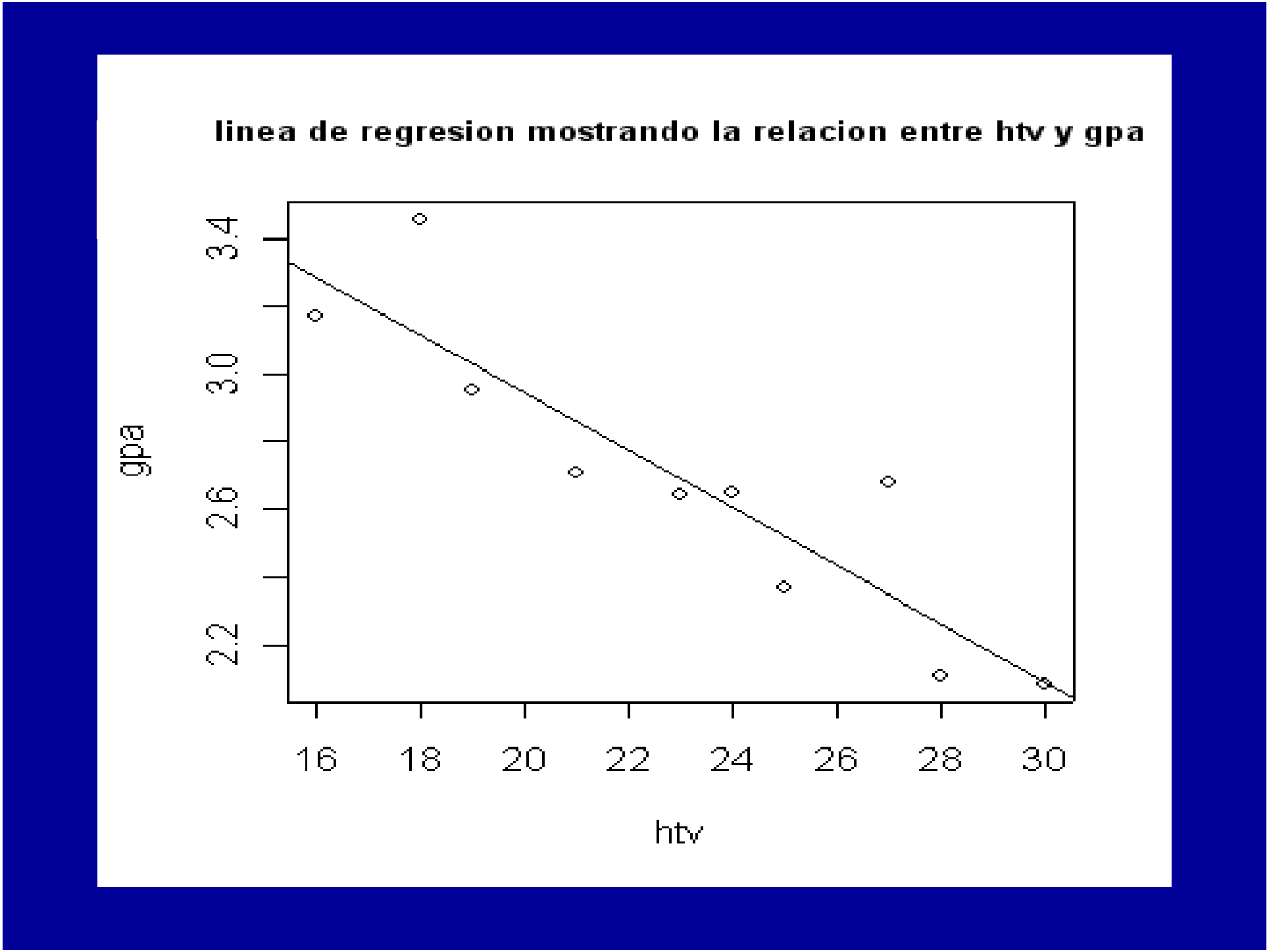


Boxplots por grupos



Regresion y correlacion lineal simple

```
> cor(htv,gpa)
[1] -0.9015101
> rl=lm(gpa~htv)
> summary(rl)
Call:
lm(formula = gpa ~ htv)
Residuals:
     Min       1Q   Median       3Q      Max
-0.15562 -0.14167 -0.06545  0.03194  0.33462
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.64495    0.33899  13.702 7.76e-07 ***
htv         -0.08498    0.01442  -5.892 0.000365 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1
Residual standard error: 0.1982 on 8 degrees of freedom
Multiple R-Squared: 0.8127,     Adjusted R-squared: 0.7893
F-statistic: 34.72 on 1 and 8 DF,  p-value: 0.000365
```


linea de regresion mostrando la relacion entre htv y gpa
gpa
2.2
2.6
3.0
3.4
16
18
20
22
24
26
28
30
htv

Comparando dos grupos

```
t.test(gpa~as.factor(genero))

        Welch Two Sample t-test

data:  gpa by as.factor(genero)
t = -0.5372, df = 4.592, p-value = 0.6161
alternative hypothesis: true difference in means is not
    equal to 0
95 percent confidence interval:
 -1.0156057  0.6722724
sample estimates:
mean in group M mean in group V
       2.613333        2.785000
```

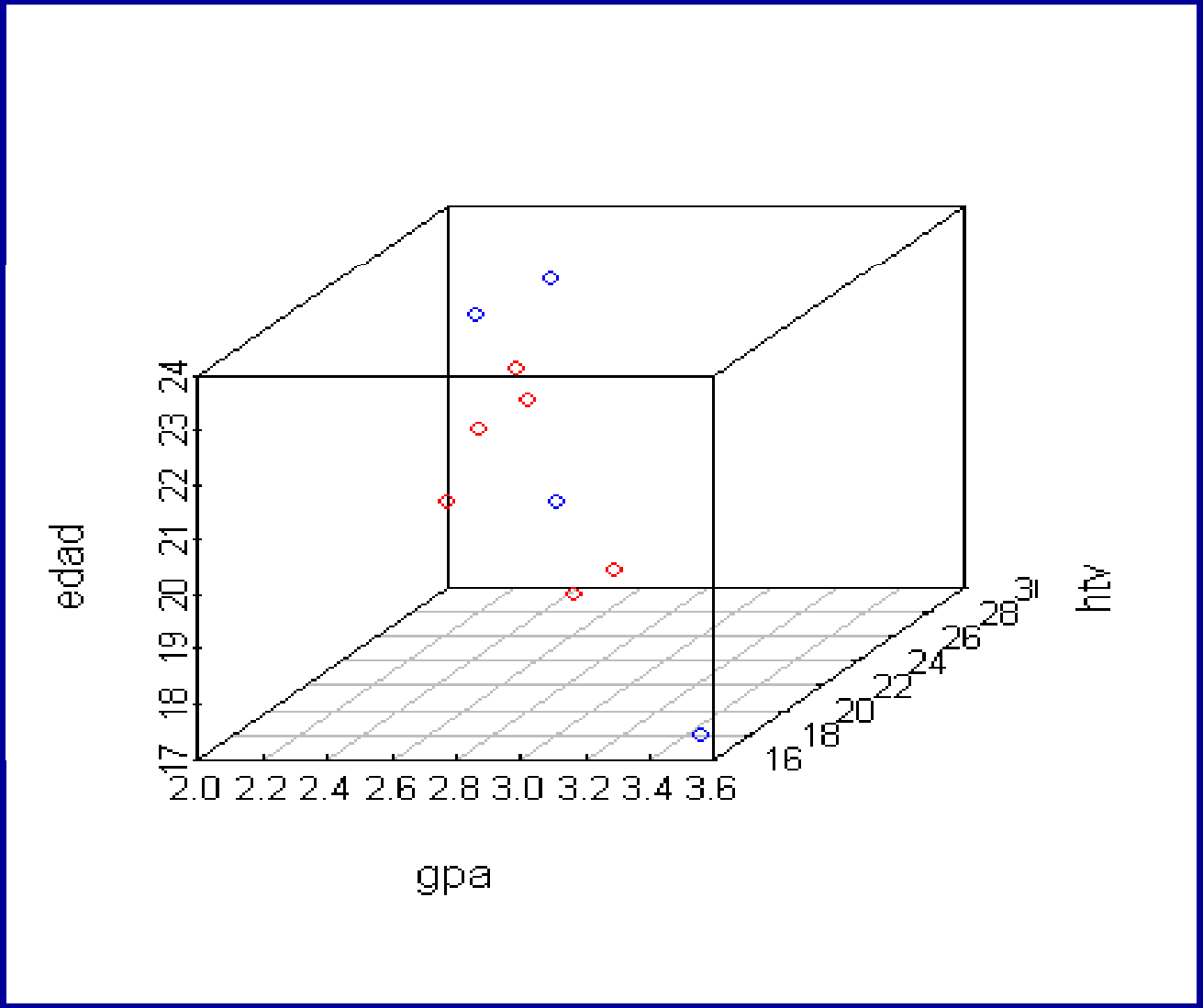
Plot en 3 dimensiones

```
 color=c("red","blue")[as.factor(genero)]
> color
 [1] "red"  "blue" "blue" "red"  "red"  "blue" "red"
"red"  "red"  "blue"
```

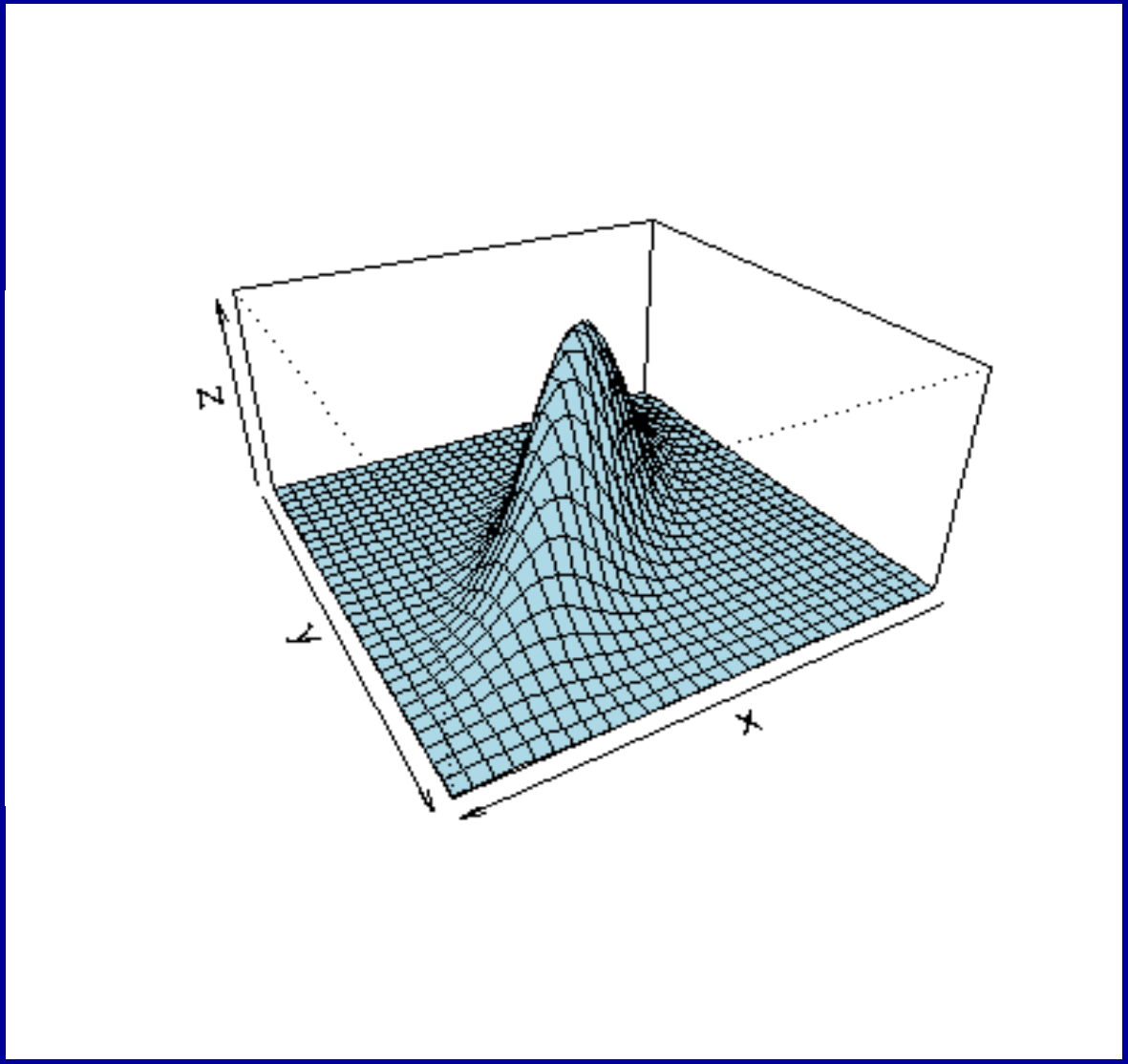
- scatterplot3d(gpa,htv,edad,color)

```
x =seq(-3, 3, length= 30)
y=x
f=function(x,y) { (1/(2*pi*.6))*exp(-(1/.72)*(x^2-
1.6*x*y+y^2) )} # definiendo la funcion
z = outer(x, y, f) # calculando la funcion en cada par
(x,y)
persp(x, y, z, theta = 30, phi = 30, expand = 0.5, col =
"lightblue")
```

Plot en 3 dimensiones



Grafica de una densidad normal bivariada (0,0,1,1,.8)



V. Matrices

```
x=c(1,4,3,7,5,8)
xmat=matrix(x,nrow=2,ncol=3) #convirtiendo el vector en
matriz
xmat
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    4    7    8
xmat[1,2] # el elemento de la matriz en la posicion (1,2)
[1] 3
 xmat[,3] # la tercera columna de la matriz
[1] 5 8

xmat[2,] # la segunda fila de la matriz

[1] 4  7  8
```

Añadiendo y quitando columns

```
> addcol=c(9,2)
>
> newmat=cbind(xmat,addcol)
> newmat
          addcol
[1,] 1 5 7      9
[2,] 3 4 8      2
> mat2=newxmat[,-c(3,4]) #elimina columnas 3 y 4
> mat2

[1,] 1 5
[2,] 3 4
```

Haciendo operaciones con matrices

m1+m2 #suma de matrices

dim(m1) #La dimension de una matriz

t(m1) # transpuesta de una matriz

t(m1)%*%m1 # producto de matrices t(m1) y m1

det(m1) # determinante de una matriz

solve(m1) #inversa de una matriz

eigen(m1) # produce los valores y vectorios propios de la matriz m1

Hallando estadísticas de filas y columnas

colSums(m1) # suma de columnas

rowSums(m1) # suma de filas

apply(m1,2,sum) # suma de columnas

apply(m1,1,sum) # suma de filas

apply(m1,1,max) # maximo de las filas

apply(m1,2,min) # minimo de las columnas

apply(m1,2,mean) # media de las columnas

VI. Leyendo datos

•read.table(“c://datos1.txt”,header=T) # lee los datos del archivo datos1.txt localizado en el drive c:. En la primer linea aparecen los nombres de las variables.

•read.table(“http://math.uprm.edu/~edgar/clase97.dat”, header=T) #lee los datos del archivo clase197.dat de la pagina de internet del profesor.

•data(“datos1”) carga el archivo de datos cuando este esta disponible en un paquete.

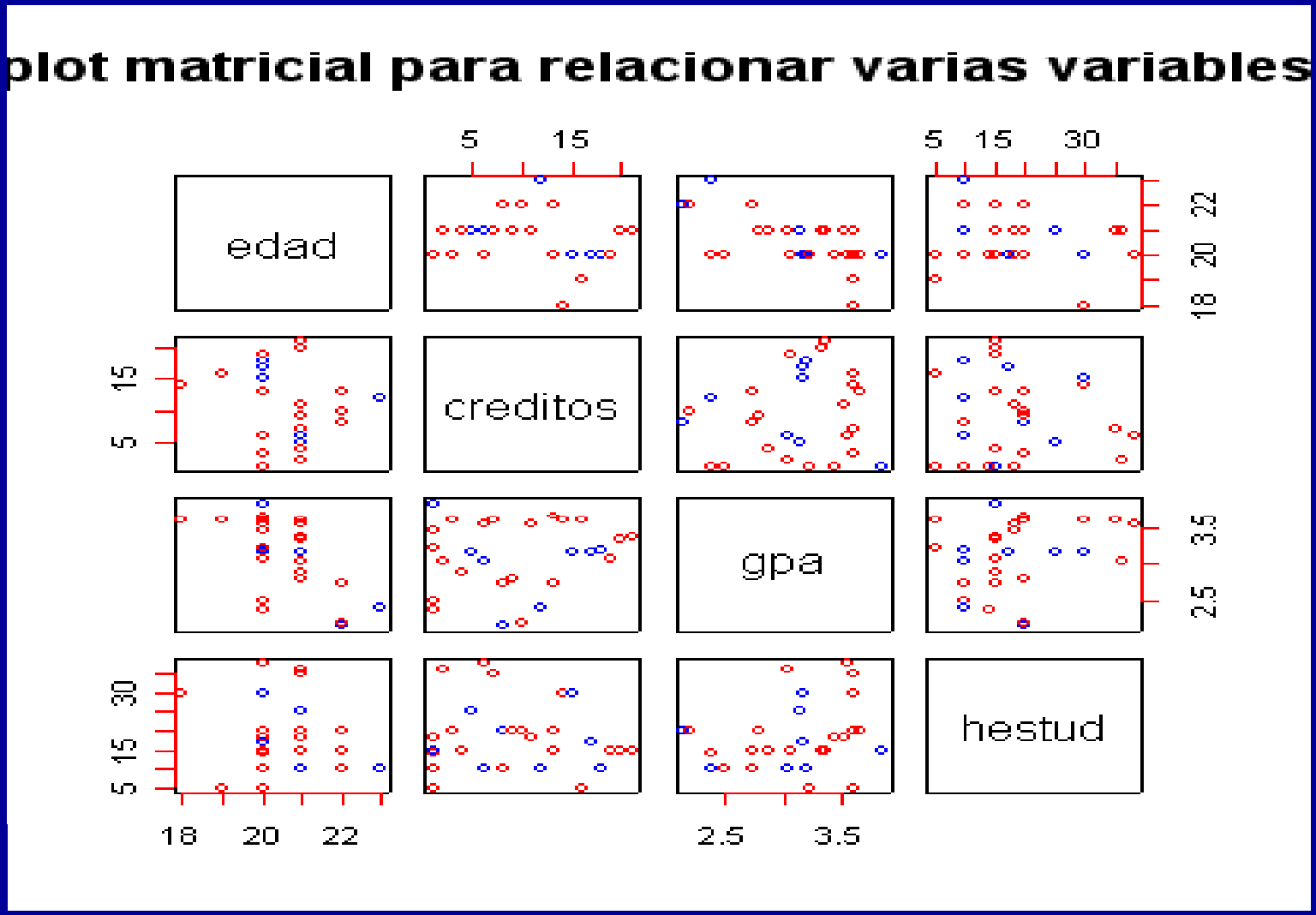
•Tambien hay interfaces que permiten leer datos de otros programas estadisticos como SAS, SPPS y MINITAB

•write(t(clase),file="c://clase.txt",ncolumns=9) #escribe los datos del objeto clase en el archivo clase.txt de c:

Leyendo datos en Excel

```
#metodo1: Pasando a formato csv
a=read.csv("c://datos1.csv“,sep="; ")
a
# Metodo 2: Usando la interface ODBC para bases de datos
#incluye bases de datos en ACCESS, Excel y Dbase
require(RODBC)
z = odbcConnectExcel2007("c://datoscluster.xlsx")
dd = sqlFetch(z,"datoscluster")
close(z)
dd
z = odbcConnectAccess2007("c://census.accdb")
dd = sqlFetch(z,"Datosprueba")
close(z)
dd
```

Grafica en mas de tres dimensiones



VII. Usando la ayuda de R

help(plot)
?plot

help.search(“plot”) #lista todas las funciones que tiene el string “plot”. Un comando similar es

apropos(“plot”)

Tambien hay un menu de help, en donde hay manuales en formatos pdf e informacion acerca de comandos y paquetes en formato html.

help(package=Rcmdr) # da ayuda acerca del uso del paquete Rcmdr

La ventana de ayuda para plot (vista parcial)

plot package:graphics R Documentation

Generic X-Y Plotting

Description:

Generic function for plotting of R objects. For more details about the graphical parameter arguments, see 'par'.

Usage:

```
plot(x, y, ...)
```

Arguments:

x: the coordinates of points in the plot. Alternatively, a single plotting structure, function or _any R object with a 'plot' method_ can be provided.

y: the y coordinates of points in the plot, _optional_ if 'x' is an appropriate structure.

VIII. Programando funciones

```
moda=function(x)
{
#Funcion que encuentra la moda de un vector x
  m1=sort(table(x),decreasing=T)
  moda=names(m1[m1==m1[1]])
  moda=as.numeric(moda)
  return(moda)
}
> x1=c(2,3,4,4,5,2,3,3,8)
> moda(x1)
[1] 3
> x
[1] 1 3 4 5 3 2 4 5 7
> moda(x)
[1] 5 4 3
```

Programando funciones

```
tablaq=function (x)
{# Esta funcion hace una tabla de frequencias para
datos discretos
freq.abs=table(x)
freq.rel.perc=freq.abs*100/length(x)
freq.abs.acum=cumsum(freq.abs)
freq.rel.perc.acum=cumsum(freq.rel.perc)
cat("\n","Tabla de frecuencias para datos
discretos","\n")
tabla=cbind(freq.abs,freq.rel.perc,freq.abs.acum,freq.
rel.perc.acum)
cat("\n","variable")
colnames(tabla)=c("frec. absoluta","frec.
relativa","frec. abs. acum","frec. rel. acum")
print(tabla)
}
```

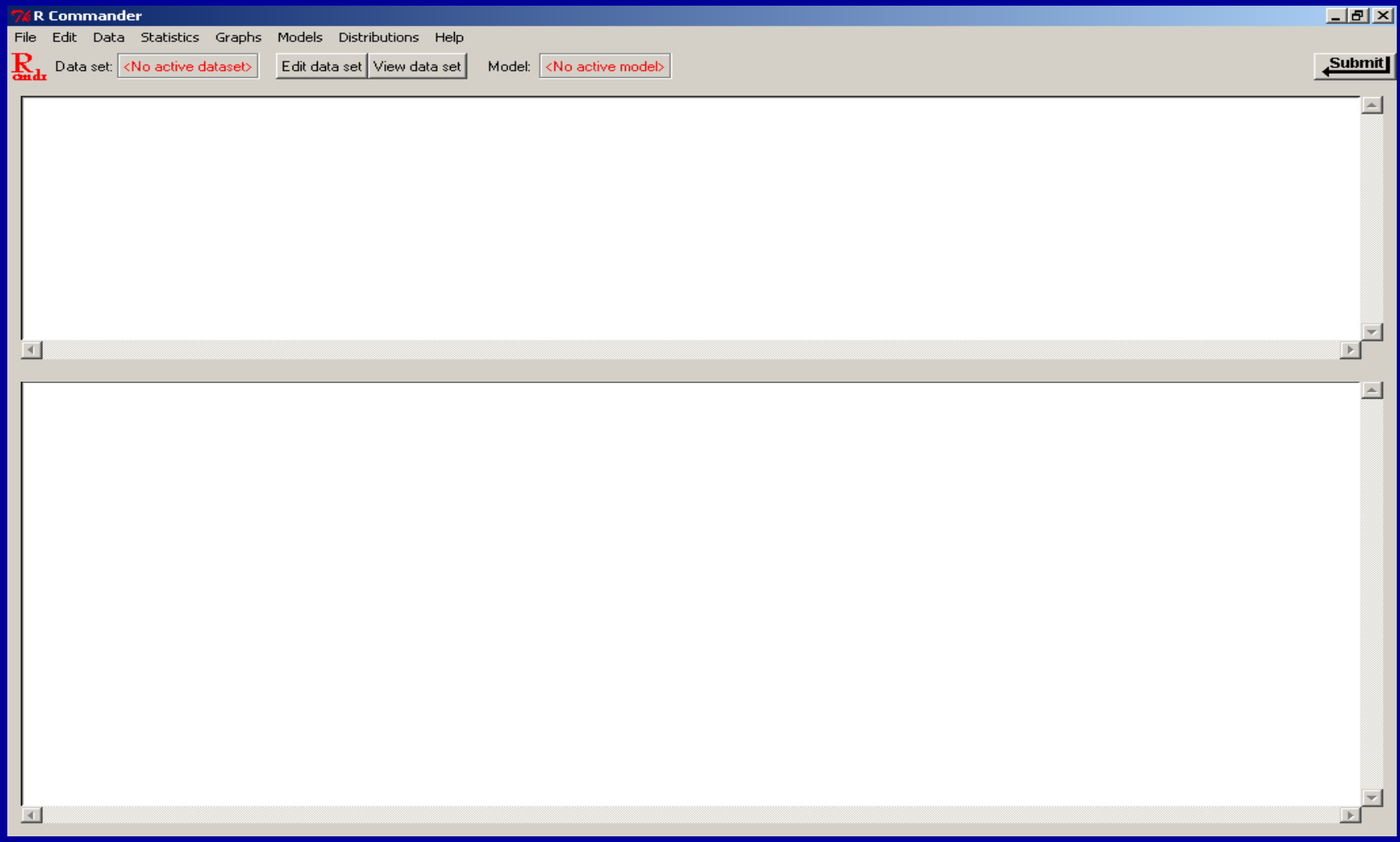
Paquetes(librerias)

- Un Paquete es una coleccion de funciones que realizan ciertas tareas especificas y que han sido construidas por diversos usuarios de R.
- Hay alrededor de 500 paquetes disponibles en el website de R.
- La mayoria de ellos se instalan eligiendo primero el menu Packages y luego la opcion Install Packages from CRAN.
- La calidad y la cantidad de funciones incluidas en los paquetes varia bastante.
- library("Rcmdr") # carga las funciones del paquete Rc,dr

Algunos paquetes disponibles

fBasics	Financial Software Collection - fBasics
foreign	Read Data Stored by Minitab, S, SAS, SPSS, Stata, Systat, dBase,
lattice	Lattice Graphics
latticeExtra	Extra Graphical Displays based on lattice
mimR	mimR: A package for graphical modelling in R
nortest	Tests for Normality
plotrix	Useful plotting functions
Rcmdr	R Commander
Rlab	Functions and Datasets Required for ST370 class
vcd	Visualizing Categorical Data

IX. Usando el gui R Commander



Menu del Rcmdr

File: Menu de opciones para cargar y salvar archivos log/script, setar opciones y salir.

Edit: Opciones para editar el contenido de las ventanas output y log/script..

Data: contiene opciones para leer y manipular datos.

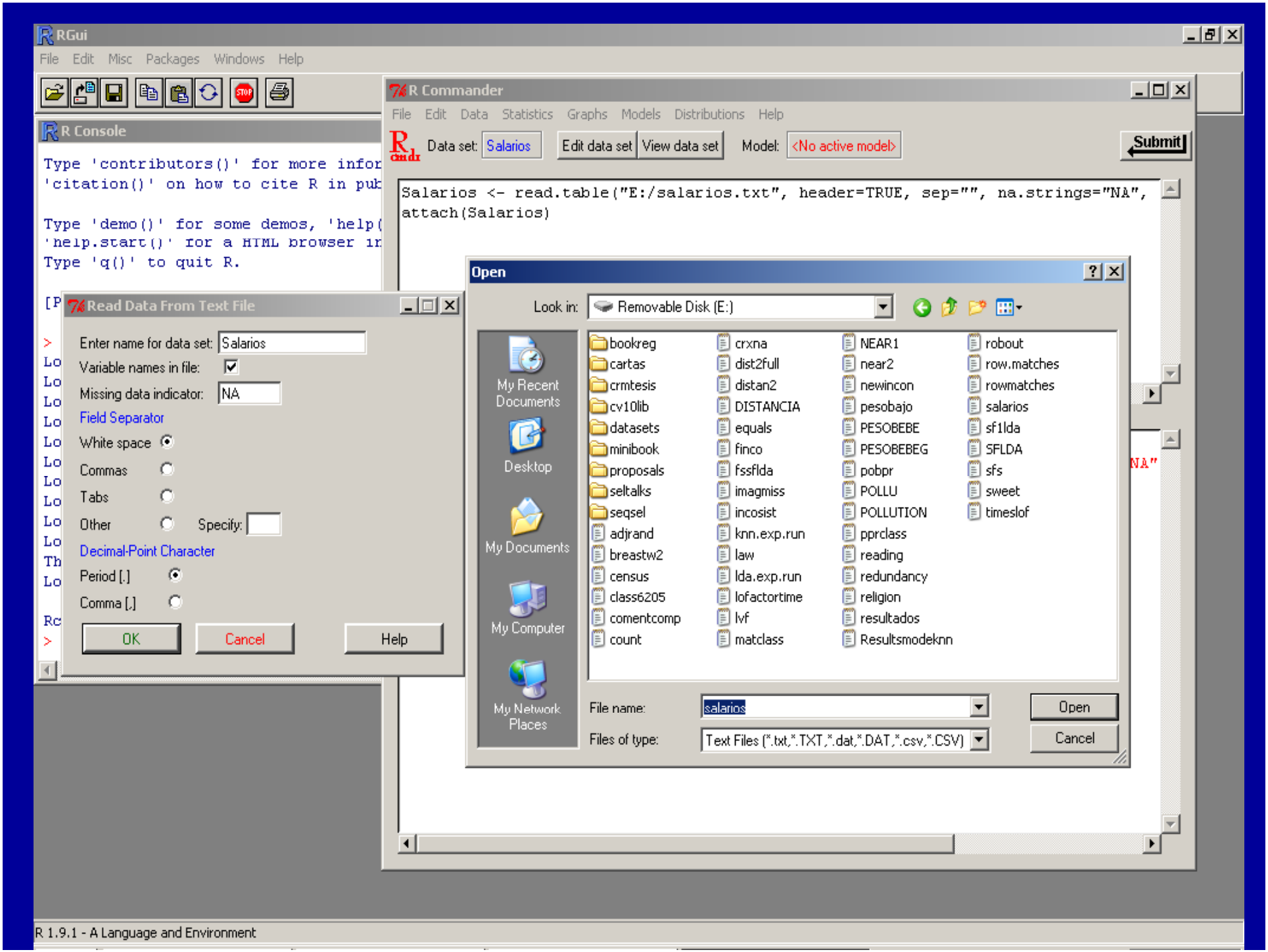
Statistics: Submenus contaniendo opciones para analisis estadistico basico.

Graphs: Contiene opciones para crear grafucas estadisticas elementales.

Models: Contiene opciones para obtener resumenes numericos, hacer pruebas de hipotesis y modelos de regresion.

Distributions: Contiene opciones para calcular probabilidades, obtener quantiles, and graficas de distribuciones estadisticas conocidas.

Help: Menu de opciones para obtener informacion acerca del R Commander


RGui
File Edit Misc Packages Windows Help
R Console
Type 'contributors()' for more infor
'citation()' on how to cite R in pub
Type 'demo()' for some demos, 'help(
'help.start()' for a HTML browser in
Type 'q()' to quit R.
R Commander
File Edit Data Statistics Graphs Models Distributions Help
Data set: Salarios
Edit data set
View data set
Model: <No active model>
Submit
Salarios <- read.table("E:/salarios.txt", header=TRUE, sep="", na.strings="NA",
attach(Salarios)
Read Data From Text File
Enter name for data set: Salarios
Variable names in file:
Missing data indicator: NA
Field Separator
White space
Commas
Tabs
Other
Specify:
Decimal-Point Character
Period [.]
Comma [,]
OK
Cancel
Help
Open
Look in: Removable Disk (E:)
My Recent Documents
Desktop
My Documents
My Computer
My Network Places
bookreg
cartas
crmtesis
cv10lib
datasets
minibook
proposals
seltalks
seqsel
adjrand
breastw2
census
class6205
comentcomp
count
crxna
dist2full
distan2
DISTANCIA
equals
finco
fssflda
imagmiss
incosist
knn.exp.run
law
lda.exp.run
lofactortime
lvf
matclass
NEAR1
near2
newincon
pesobajo
PESOBEBE
PESOBEBEG
pobpr
POLLU
POLLUTION
pprclass
reading
redundancy
religion
resultados
Resultsmodeknn
robout
row.matches
rowmatches
salarios
sf1lda
SFLDA
sfs
sweet
timeslof
File name: salarios
Files of type: Text Files (*.txt,*.TXT,*.dat,*.DAT,*.csv,*.CSV)
Open
Cancel
NA"
R 1.9.1 - A Language and Environment

RGui - [Data Editor]

File Windows Edit Help

	Salario	Ahorro	var3	var4	var5	var6	var7	var8	var9	var10	var11
1	800	150									
2	850	100									
3	900	280									
4	1200	400									
5	1500	350									
6	1700	500									
7	1900	635									
8	2000	600									
9	2300	750									
10	2500	680									
11	2700	900									
12	3000	800									
13	3200	300									
14	3500	1200									
15	5000	1000									
16											
17											
18											
19											
20											
21											
22											
23											
24											
25											
26											
27											
28											
29											
30											

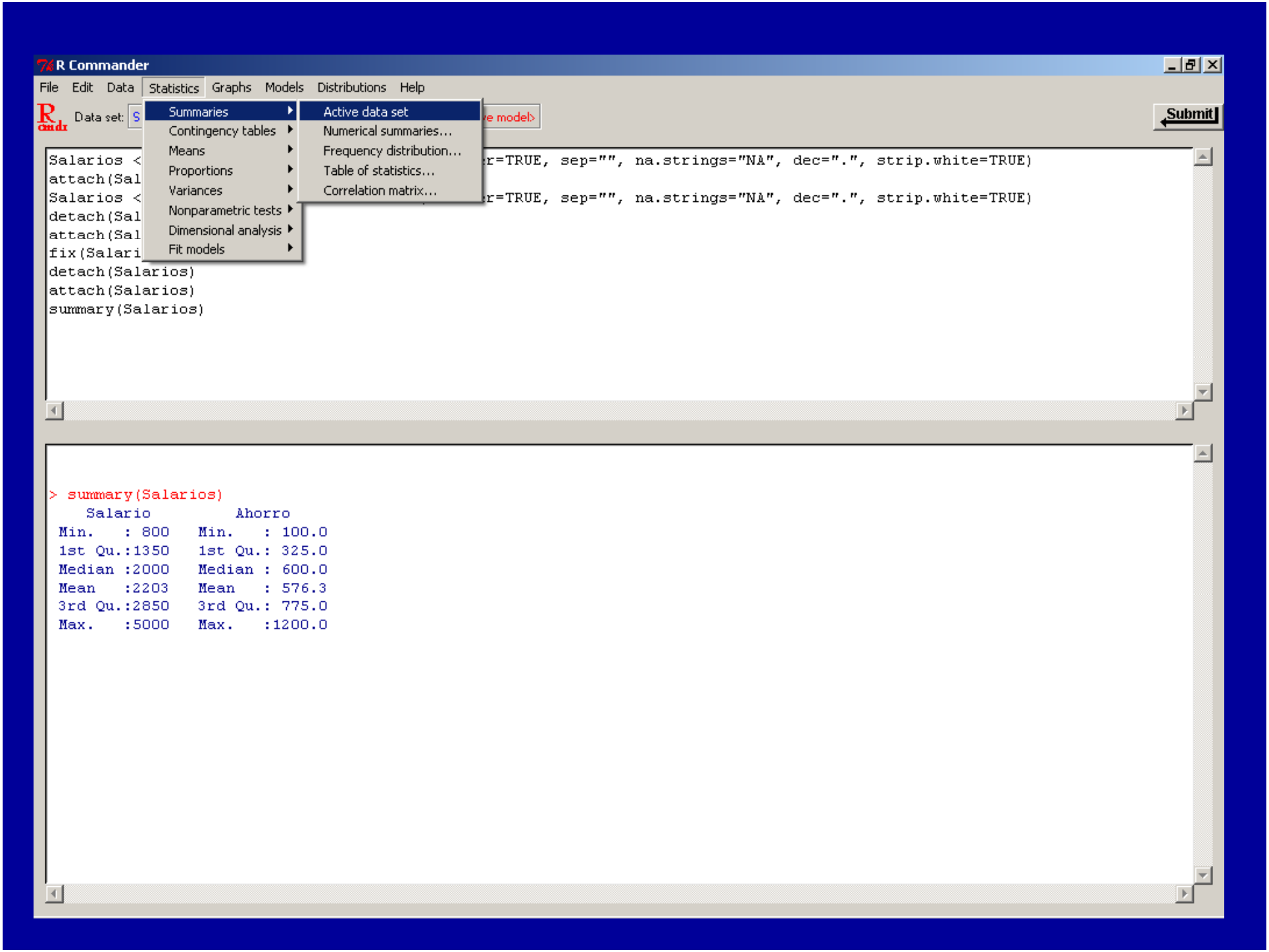
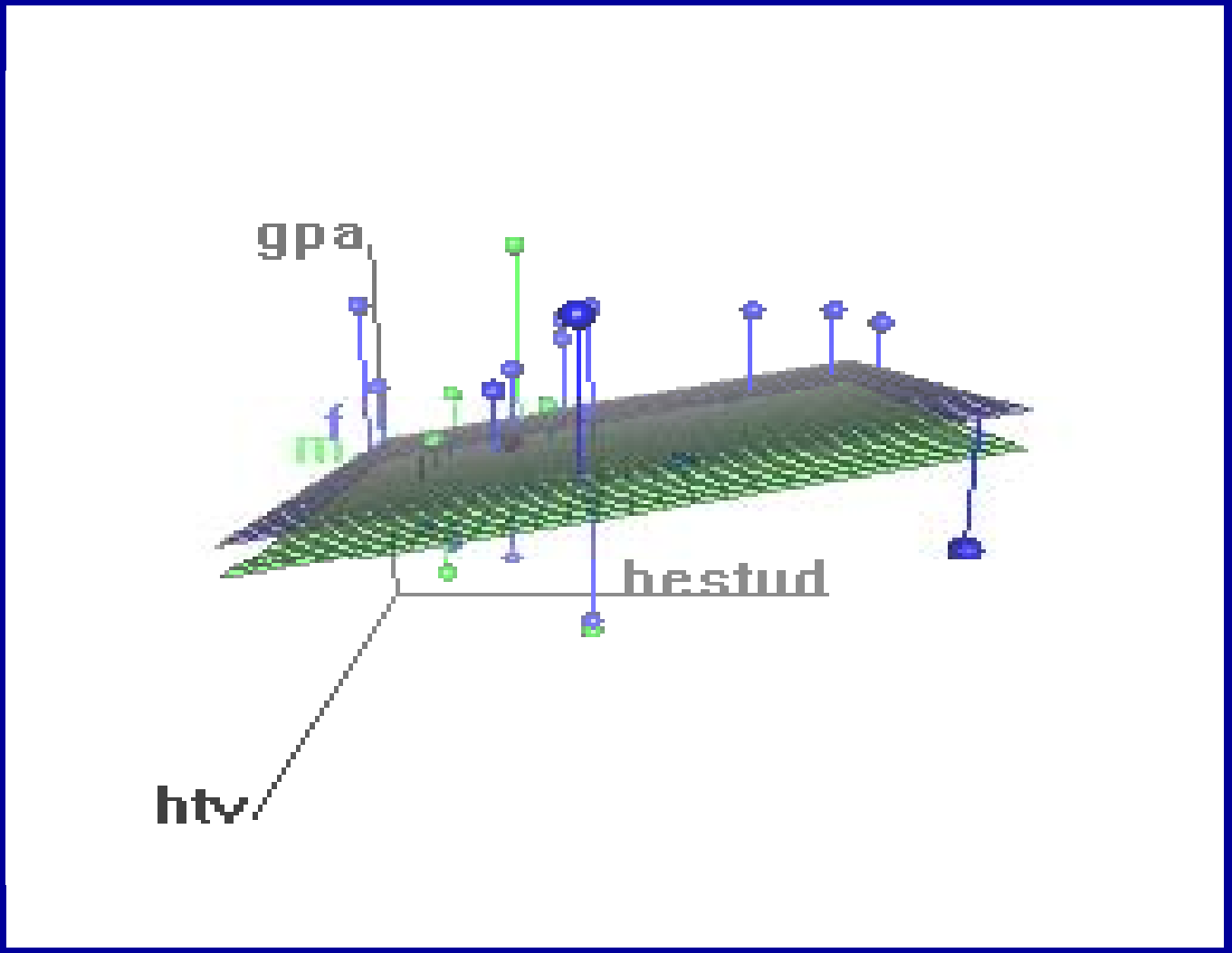

R Commander
File Edit Data Statistics Graphs Models Distributions Help
Data set:
Submit
Summaries
Contingency tables
Means
Proportions
Variances
Nonparametric tests
Dimensional analysis
Fit models
Active data set
Numerical summaries...
Frequency distribution...
Table of statistics...
Correlation matrix...
Salarios <
attach(Sal
Salarios <
detach(Sal
attach(Sal
fix(Salari
detach(Salarios)
attach(Salarios)
summary(Salarios)
r=TRUE, sep="", na.strings="NA", dec=".", strip.white=TRUE)
r=TRUE, sep="", na.strings="NA", dec=".", strip.white=TRUE)
> summary(Salarios)
Salario Ahorro
Min. : 800 Min. : 100.0
1st Qu.:1350 1st Qu.: 325.0
Median :2000 Median : 600.0
Mean :2203 Mean : 576.3
3rd Qu.:2850 3rd Qu.: 775.0
Max. :5000 Max. :1200.0

Grafico 2-D en Rcmdr



X. Construyendo librerias

1. Instalación de los programas necesarios para la construcción de la librería

a. Bajar el paquete de utilidades de unix para windows tools.zip y
desempaquetar todos los archivos excepto **cygwin1.dll** en el directorio c:\cygwin\bin.

b. Mover el archivo cywin1.dll al directorio c:\cygwin. Si su librería contiene llamadas a funciones en C o Fortran talvez deba bajar todo el ambiente cygwin en c:\cygwin.

c. Bajar e instalar Active Perl en el folder c:\ActivePerl

d. Bajar e instalar HTMLHelp Workshop en el folder c:\HTML Help Workshop

e. Bajar e instalar minGW en el folder c:\minGW.

Construyendo librerias

2. Hacer las modificaciones en su path

Para que los programas instalados en el paso 1 puedan correr hay que hacer previamente lo siguiente.

3. Construir el esqueleto (versión básica) del paquete

Supongamos que tiene varias funciones en su medio ambiente (“environment”) de R, digamos “fun1”, “fun2”,…,”funN “ las cuales han sido corridas individualmente y que se las quiere ensamblarlas y ponerlas todas a la vez en una librería llamada “Mipaquete”, la cual va a estar localizada en el directorio (fólder) “c:\Rpaquetes”. Supongamos además que estas funciones han sido probadas usando los conjuntos de datos “dat1”,….”datN”.

a. Entrar al medio ambiente de R

b. Escribir el siguiente comando de línea dentro de R

package.skeleton(name="mipaquete",
list=c(“fun1”,”fun2”,…..”funN”,”dat1,…”datN”),
path=”c://Rpaquetes”).

Construyendo librerias

- c. Al hacer esto se va a crear un directorio “mipaquete” dentro del directorio“Rpaquetes”. El directorio “mipaquete” contiene varios subdirectorios: “R”,“data”, ”man” , “src” y los archivos de texto DESCRIPTION y README. Dentro delsubdirectorio de “R” estarán las funciones que se crearon y en el subdirectorio “data”estarán los conjuntos de datos.
- d. Salirse del medio ambiente de R y moverse al directorio donde está el archivo DESCRIPTION y editarlo (puede usar NOTEPAD).

Construyendo librerias

e. En el subdirectorio “man” hay un archivo de texto por cada función incluida en la librería. Estos son los archivos de ayuda para explicar el uso de cada una de las funciones. Editar cada uno de los campos que aparece en el archivo. Puede usar Notepad o WordPad.

4. Construyendo la librería

a) Abrir la ventana de comando DOS, para esto elija la opción “Run” del menú Start y escriba “cmd” y luego oprima OK.

b) Moverse al directorio c:\Rpaquetes

c) Escriba el commando

RCMD build –force –binary mipaquete

Esto crea el archivo mipaquete.zip que es una versión comprimida de la librería “Mipaquete”.

Un ejemplo basico

El siguiente comando crea una libreria basico que consiste de las funciones moda, tablaq y del conjunto de datos clase

```
>package.skeleton(name="basico",list=c("moda","tablaq","cla
se"),path="c://Rpaquetes")
Creating directories ...
Creating DESCRIPTION ...
Creating READMEs ...
Saving functions and data ...
Making help files ...
Done.
Further steps are described in
c://Rpaquetes/basico/README
>
```

La libreria dprep

```
                Information on Package 'dprep'

Description:

Package:        dprep
Title:          Data preprocessing and visualization functions for
                classification
Version:        1.3
Author:         Caroline Rodriguez<caroline@math.uprm.edu> and Edgar
                Acuna<edgar@cs.uprm.edu>
Description:    Functions for: imputation, normalization,feature
                selection, outlier detection, and visualization
Maintainer:     Caroline Rodriguez <caroline@math.uprm.edu>
License:        GPL
Built:          R 2.0.1; i386-pc-mingw32; 2005-02-04 08:16:45; windows

Index:

baysout                 Function that determines the outlyingness
                        measure for a requested number of
                        observations.
bupa                    Bupa
ce.knn.imp              Function that calls ec.knnimp to perform knn
                        imputation
ce.mimp                 Mean or median imputation
census                  The Census dataset
```

Exploring the dataset using *imagmiss()*

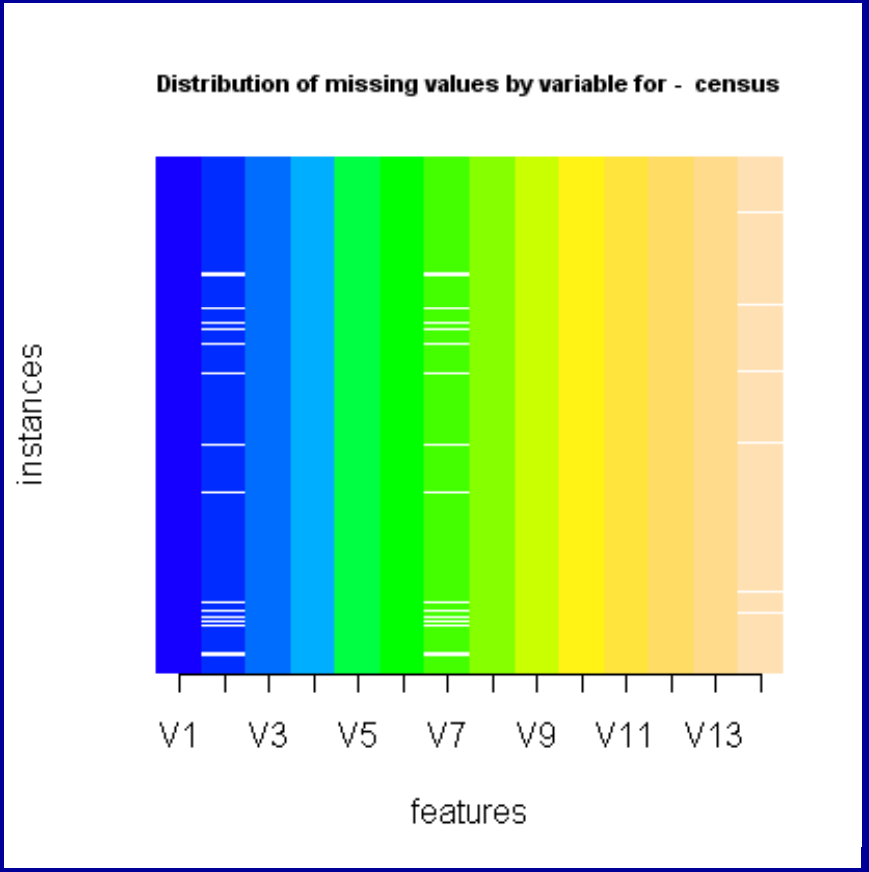
> imagmiss(data, name="dataname")

Report on missing values for Census :

Number of missing values overall: 4262
Percent of missing values overall: 0.9349485
Features with missing values (percent):

V2	V7	V14
5.638647	5.660146	1.790486

Percent of features with missing values: 21.42857
Number of instances with missing values: 2399
Percent of instances with missing values: 7.36771



The clean function

- This function deletes columns and rows with a large number of missing values
- census.cl=clean(censusn,tol.col=.5,tol.row=.3,name="cl.census")
  - **Variables Percent.of.missing**
  - **1 V2 5.6386474616873**
  - **2 V7 5.66014557292466**
  - **3 V14 1.79048555019809**
  - **Maximum number of values to be imputed: 4262**

Treating Missing Values

- census.mimp=ce.mimp(censusn,"mean",1:14,“”)
- census.mdimp=ce.mimp(censusn,"median",atr=1:14, nomatr=c(2,4:10,14),"")
  - census.knn=ec.knnimp(censusn,k=10)
  - census.knn=ec.knnimp(censusn,nomatr=c(2,4:10,14),k=10)

Conclusion

R es flexible y poderoso

- Facil de leer datos.
- Bastante capacidad de manipular datos.
- Enorme capacidad para hacer graficas.
- Un rango bien amplio de funciones estadisticas.
- Un gran numero de paquetes disponibles.
- Se puede guardar todo el trabajo que se hace en una sesion.