

Una Introduccion a R

Edgar Acuna

(academic.uprm.edu/eacuna/introRacuna2018.pdf)

Computational and Statistical Learning Group

Departament of Mathematics

University of Puerto Rico at Mayaguez

Agosto, 2018

Contenido

- Introduccion
- Interfaces graficos para R:Rstudio
- Leyendo datos en R
- Operaciones aritmeticas basicas
- Funciones estadisticas basicas
- Graficas :GGplot2 y ggplotgui
- Matrices y DataFrames
- Programacion en R
- Haciendo aplicaciones de R en la web: Shiny
- Haciendo paquetes en R: Dprep

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), estuvo disponible desde los inicios de los 90's hasta el 2010 aprox.

R mayormente usa comandos de linea e incluye un limitado GUI. Hay varias propuestas para GUI's en R, siendo Rstudio, Rcmdr y Deducer los mas usados.

R tiene excelente capacidades de graficas.

R esta disponible para Windows, MacOS. Unix/Linux.

R tiene muy buena documentacion y ayuda disponible.

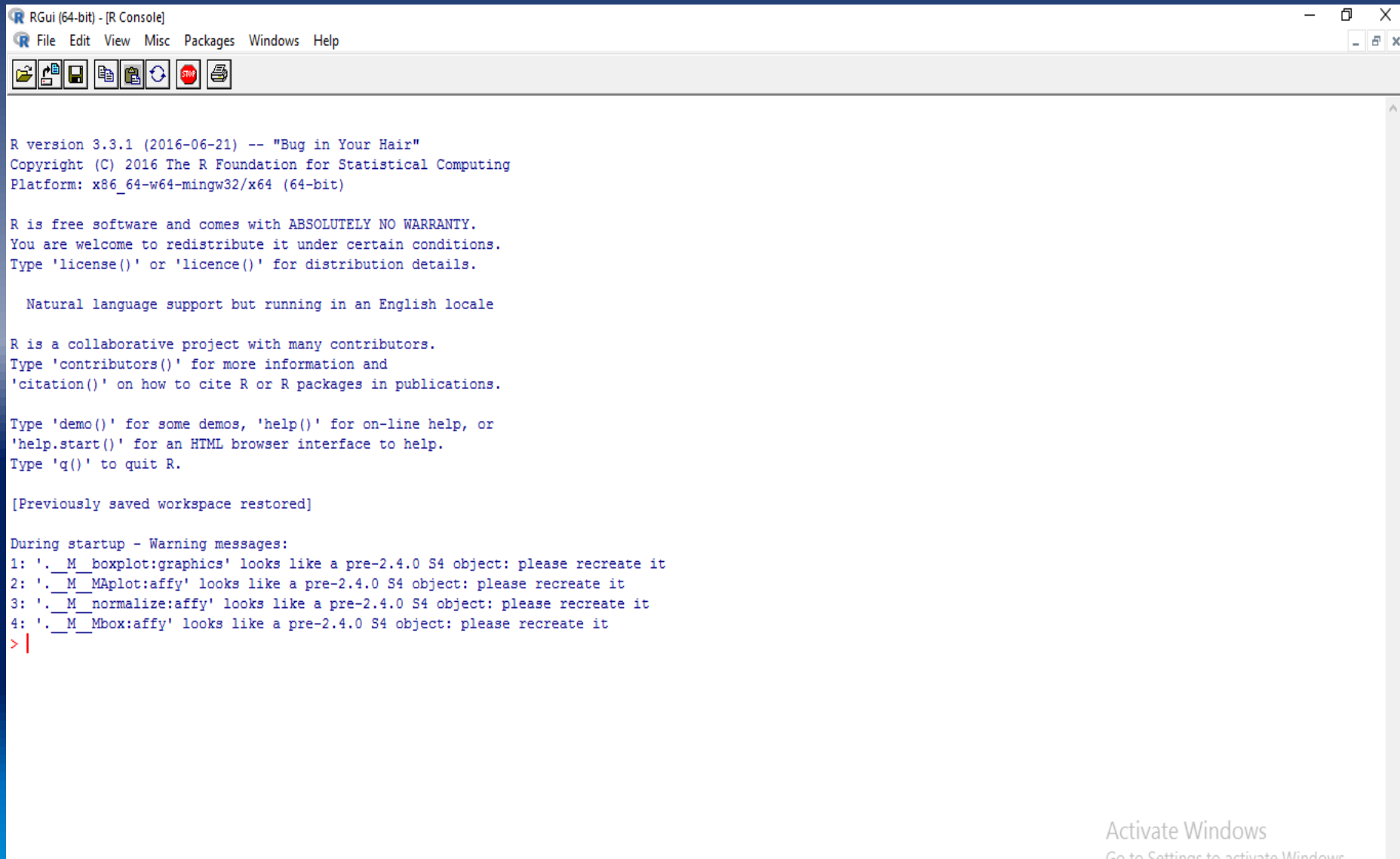
Porque usar R?

- Los metodos estadisticos mas recientes aparecen primero en R.
- Existen muchas librerias disponibles (11,926, Enero 2018) 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) y en diferentes formatos csv, xml, json.
- Puede interactuar con muchos lenguajes de Programacion: Java, C++, Python, etc.
- Es gratis.

Instalando R

- 1-Entrar Website: cran.r-project.org.
- 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. En la pantalla R-3.5.1 for Windows hacer un click a Download R-3.5.1 for Windows para descargar el archivo R-3.5.1-win.exe
5. Localizar el archivo R-3.5.1-win.exe en su computadora y ejecutarlo eso instalara R

El ambiente grafico de R



```
RGui (64-bit) - [R Console]
File Edit View Misc Packages Windows Help

R version 3.3.1 (2016-06-21) -- "Bug in Your Hair"
Copyright (C) 2016 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

During startup - Warning messages:
1: '._M_boxplot:graphics' looks like a pre-2.4.0 S4 object: please recreate it
2: '._M_MApplot:affy' looks like a pre-2.4.0 S4 object: please recreate it
3: '._M_normalize:affy' looks like a pre-2.4.0 S4 object: please recreate it
4: '._M_Mbox:affy' looks like a pre-2.4.0 S4 object: please recreate it
> |
```

Activate Windows
Go to Settings to activate Windows

Ambientes graficos (GUI) de R

1-R commander Rcmdr es un paquete que se instala dentro de R (John Fox, 2007)

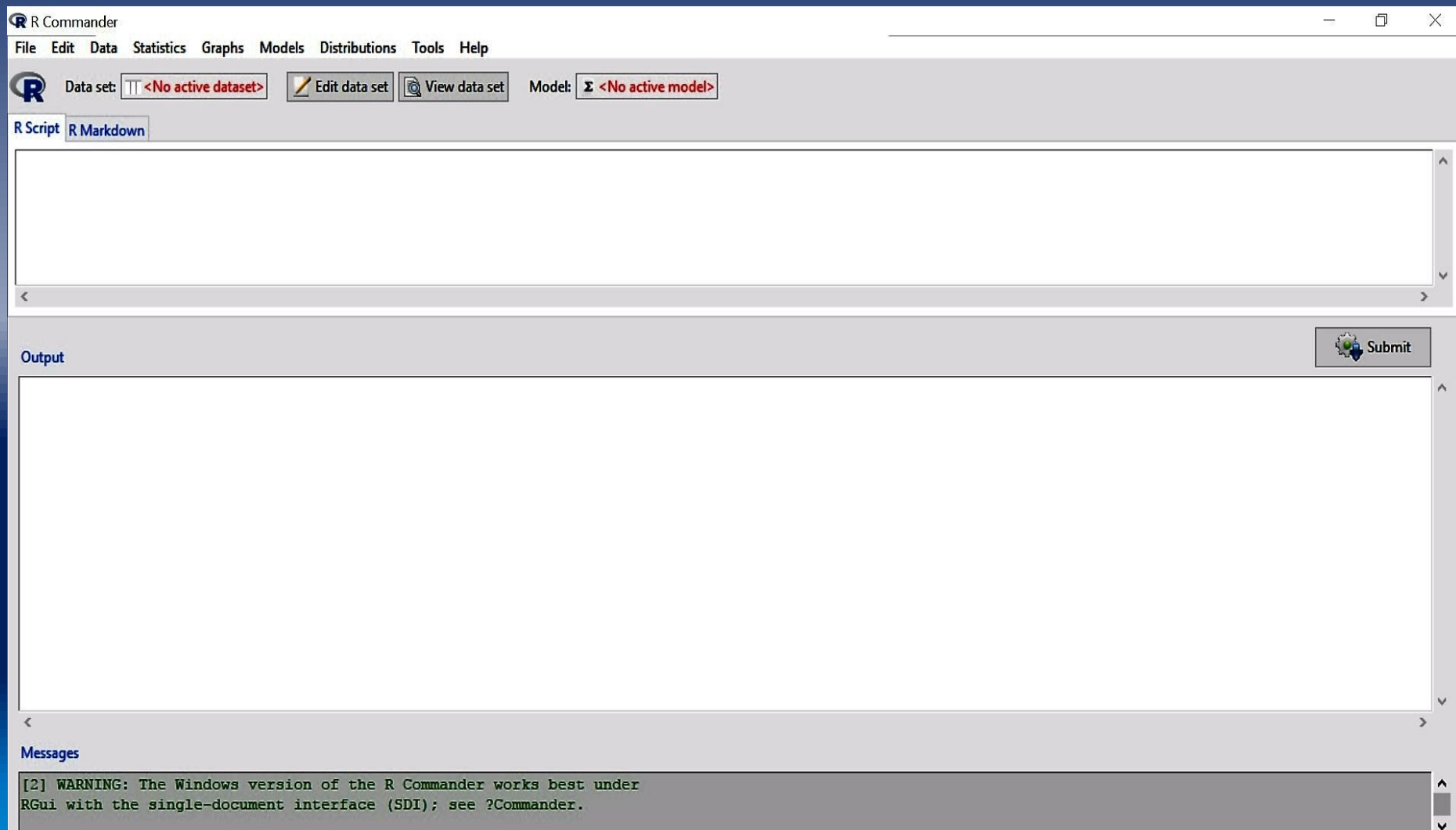
2-Para instalar Rstudio entrar a www.rstudio.com
(Hadley Wickham, 2011)

3. Para instalar Deducer entrar a www.deducer.org. (Ian Fellows, 2011)

4. Para instalar R Analytic Flow entrar a <https://r.analyticflow.com/en/> (Ef-Prtime, Japan 2018)

De todos ellos Rstudio es el que ha ganado mas popularidad

El ambiente grafico Rcmdr de R



Usando el gui R Commander

The screenshot displays the R Commander interface. The top menu bar includes File, Edit, Data, Statistics, Graphs, Models, Distributions, Tools, and Help. Below the menu, the 'Data set:' field shows 'bupa' with buttons for 'Edit data set' and 'View data set'. The 'Model:' field shows '<No active model>'. The main workspace is divided into three panes: 'R Script' (empty), 'Output' (empty), and 'Messages' (containing two messages: [1] NOTE: R Commander Version 2.4-1: Thu Jan 11 19:03:30 2018 and [2] WARNING: The Windows version of the R Commander works best under RGui with the single-document interface (SDI); see ?Commander.). A 'Submit' button is located on the right side of the workspace. A 'bupa' data viewer window is open, showing a table with 30 rows and 7 columns (V1, V2, V3, V4, V5, V6, V7). The data is as follows:

	V1	V2	V3	V4	V5	V6	V7
1	85	92	45	27	31	0.0	1
2	85	64	59	32	23	0.0	2
3	86	54	33	16	54	0.0	2
4	91	78	34	24	36	0.0	2
5	87	70	12	28	10	0.0	2
6	98	55	13	17	17	0.0	2
7	88	62	20	17	9	0.5	1
8	88	67	21	11	11	0.5	1
9	92	54	22	20	7	0.5	1
10	90	60	25	19	5	0.5	1
11	89	52	13	24	15	0.5	1
12	82	62	17	17	15	0.5	1
13	90	64	61	32	13	0.5	1
14	86	77	25	19	18	0.5	1
15	96	67	29	20	11	0.5	1
16	91	78	20	31	18	0.5	1
17	89	67	23	16	10	0.5	1
18	89	79	17	17	16	0.5	1
19	91	107	20	20	56	0.5	1
20	94	116	11	33	11	0.5	1
21	92	59	35	13	19	0.5	1
22	93	23	35	20	20	0.5	1
23	90	60	23	27	5	0.5	1
24	96	68	18	19	19	0.5	1
25	84	80	47	33	97	0.5	1
26	92	70	24	13	26	0.5	1
27	90	47	28	15	18	0.5	1
28	88	66	20	21	10	0.5	1
29	91	102	17	13	19	0.5	1
30	87	41	31	19	16	0.5	1

File: Menu de opciones para cargar y guardar archivos log/script. Guardar las salidas y el espacio de trabajo de R 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 conteniendo opciones para analisis estadistico basico.

Graphs: Contiene opciones para crear graficas estadisticas.

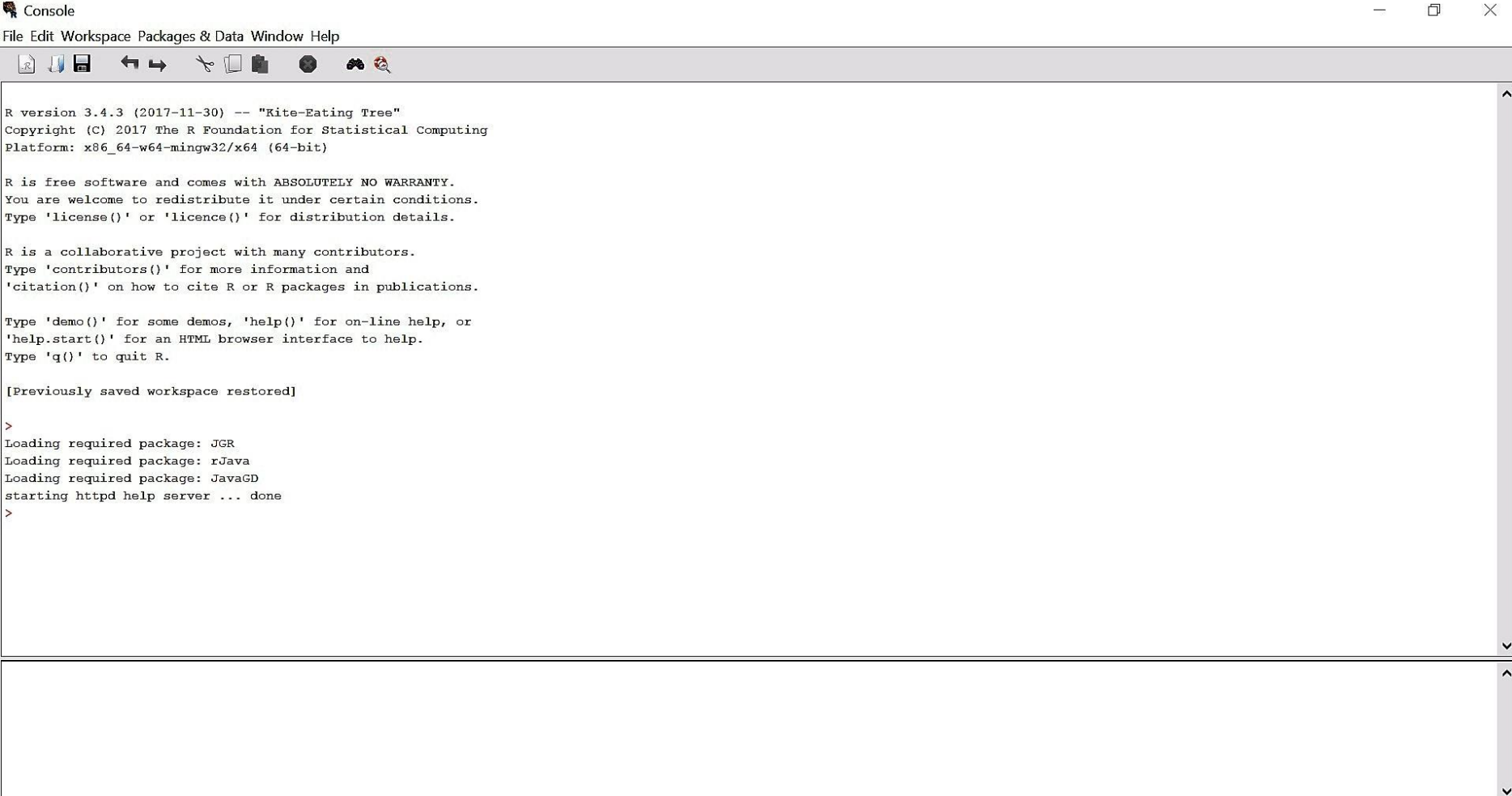
Models: Contiene opciones para obtener resúmenes numéricos, hacer pruebas de hipótesis, intervalos de confianza y modelos de regresión.

Distributions: Contiene opciones para calcular probabilidades, obtener cuantiles, and graficas de distribuciones estadísticas conocidas.

Tools: Menu de opciones para cargar paquetes de R y modificar opciones de Rcmdr

Help: Menu de opciones para obtener informacion acerca del Rcmdr

El ambiente grafico Deducer de R



The screenshot shows the R Console window with the following text:

```
R version 3.4.3 (2017-11-30) -- "Kite-Eating Tree"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

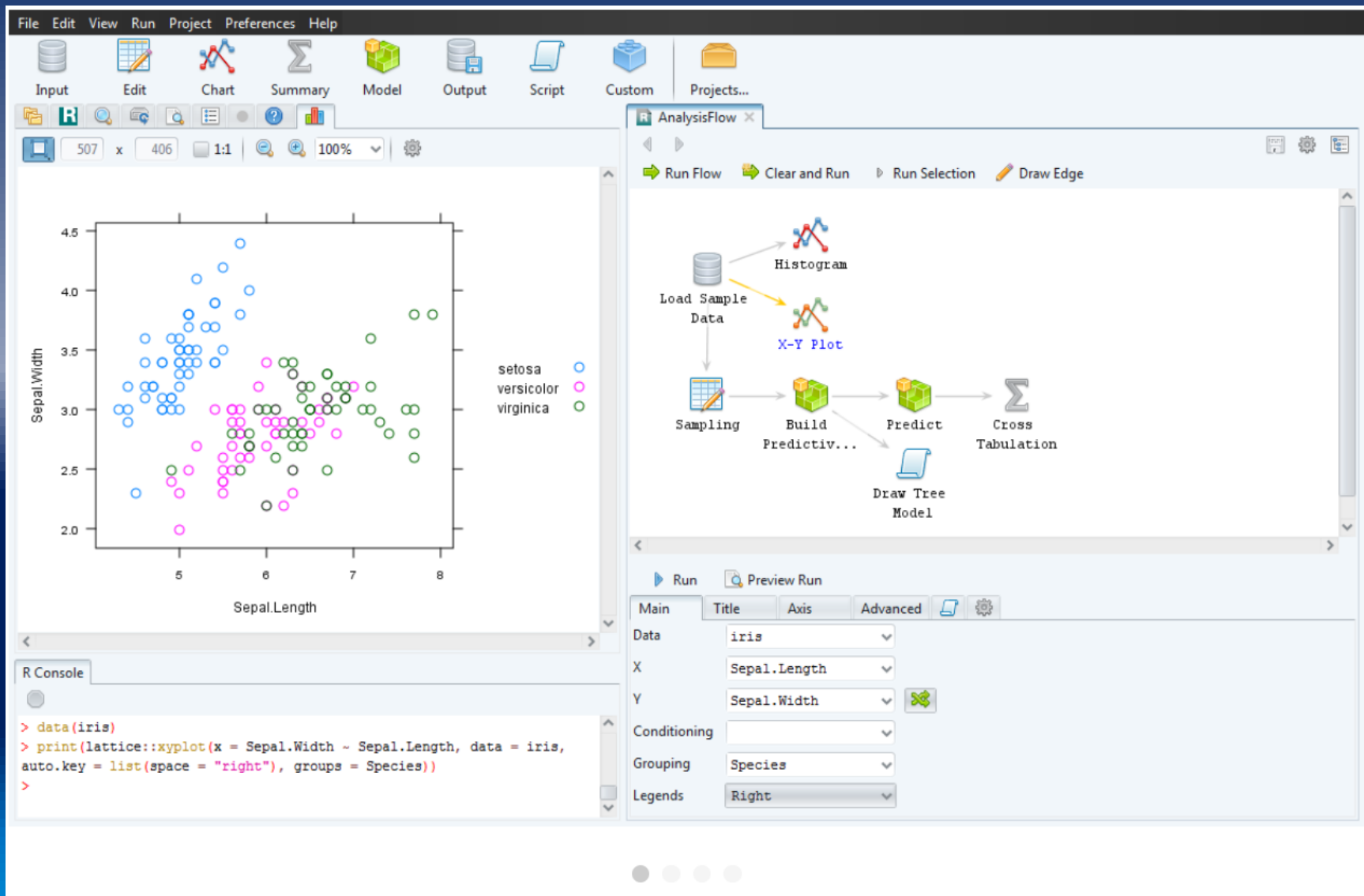
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

>
Loading required package: JGR
Loading required package: rJava
Loading required package: JavaGD
starting httpd help server ... done
>
```

El ambiente grafico de RAnalyticF



El entorno de Desarrollo (IDE) Rstudio

The screenshot displays the RStudio IDE interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu is a toolbar with icons for file operations and a 'Go to file/function' search bar. The main editor area is divided into two panes. The left pane, titled 'R demos', lists various demonstration scripts with brief descriptions: 'crosstalk-highlight-ggplotly' (Linking ggplot2 plots), 'crosstalk-highlight-intro' (An introduction to linked views), 'crosstalk-highlight-leaflet' (Linking to leaflet), 'crosstalk-highlight-pipeline' (Aggregating SharedData), 'crosstalk-highlight-subplot' (Basic scatterplot brushing with subplot()), 'rotate' (Using htmlwidgets::onRender() to rotate the camera of a 3D graph), 'ternary' (A basic ternary plot), and 'Demos in package 'stats':'. Below these are examples of 'glm.vr', 'lm.glm', 'nlm', and 'smooth' functions. The right pane, titled 'Environment', shows a table of objects in the Global Environment:

Object	Size
a1	40552 obs. of 7 variables
a2	3 obs. of 7 variables
a3	13866 obs. of 7 variables
a31	169 obs. of 7 variables
a32	738 obs. of 7 variables
a33	300 obs. of 7 variables
a34	204 obs. of 7 variables
a35	270 obs. of 7 variables
a36	245 obs. of 7 variables

Below the Environment pane is a toolbar with icons for file operations and a 'Zoom' button. The bottom pane is the Console, which shows the R startup message and the RStudio workspace loaded from ~/.RData:

```
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from ~/.RData]

> |
```

Operaciones aritmeticas basicas

$2+3$ #Suma

$2-3$ #Resta

$2*3$ #Producto

$2/3$ # Division

2^3 #Potencia

$(4^2) - (3*2)$ #Operaciones combinadas

2^{-3}

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)
```

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 el 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.

Emtrando datos en R

```
x=c(1,2,3,4) #combinado varios valores
```

```
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 desde 0 ... hasta 1.0
```

#Tambien se puede copiar datos de EXCEL o WORD usando la function scan

```
x=scan()
```

```
> x=scan()
```

```
1: 6
```

```
2: 7
```

```
3: 4
```

```
4:
```

```
Read 3 items
```

Manipulando un vector de datos

`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`.
`x[x != 4]` # vector que contiene los elementos de `x` distintos de 4.
`y=x/2` # divide los elementos del vector `x` por 2
`z=x+y` #suma los vectores `x` y `y`
`log(x, 10)` #logaritmos en base 10
`y = sqrt(x)` #raiz cuadrada

Leyendo datos de un archivo

`read.table("c://esma3016/colon.txt")` #lee los datos que estan en colon.txt localizado en c://esma3016.

`clase=read.table("http://academic.uprm.edu/eacuna/clase97.txt", header=T)` #lee los datos clase97.txt que estan en mi pagina de internet y los guarda en el objeto clase. En la primera fila aparecen los nombres de las variables.

`head(clase)` # muestra las seis primeras filas de clase

	edad	sexo	escuela	programa	creditos	gpa	familia	hestud	htv
1	21	f	publ	biol	119	3.6	3	35	10
2	18	f	priv	mbio	15	3.6	3	30	10
3								

`tail(clase)` #muestra las ultimas seis filas de clase

Leyendo datos de un archivo (cont)

Otra forma de leer datos de la internet es usando la funcion `getURL` de la libreria `RCurl()`.

```
getURL("http://academic.uprm.edu/eacuna/clase97.txt")
```

- Tambien hay interfaces que permiten leer datos de otros programas estadisticos como SAS, SPSS y MINITAB.
- Para leer datos guardados Excel es mejor tenerlo en el fomato csv y usar el commando `read.csv("c://datos1.csv", sep="; ")`
- Otra alternativas para leer datos son usar las librerias `data.table`, `foreign`, `xlsx` o `RODBC`.
- Se pueden leer tablas de datos directamente de la internet usando las librerias `rvest` y `XML`.
- Cuando los datos estan en un paquete (libreria de R) se usa simplemente: `data(datospaq)`, donde `datospaq` es un conjunto de datos que viene con el paquete.

Corriendo scripts en R

Se puede escribir varias líneas de comandos y guardarlos en un archivo con la extensión R. Por ejemplo, el archivo comandos1.R contiene las siguientes líneas

```
x=c(1,2,3,4) # combinar
cat("\nEste es el vector x\n")
x
y=1:4      # vector formado por una secuencia
cat("El vector y es:\n")
print(y)
x1=seq(1,4) # vector de 4 unos
cat("\n El vector x1 es\n")
print(x1)
x2=seq(2, 8, by=2) # vector:2,4,6,8
cat("\n El vector x2 es",x2,"\n")
x3=seq(0, 1, length=11)
cat("El vector x3 es",x3,"\n")
```

Corriendo scripts en R(cont)

Para correr el script, abrir R y cuando aparece el prompt escribir el comando

`Source("c://com1.R")` # escribir el path adecuado
o del menu File elegir la opcion Source R code

Otra opcion es abrir Rstudio y en el Menu File elegir Open File y acceder al file com1.R; Luego marcar todas las lineas que quiere ejecutar y darle run or darle CTRL+SHIFT+ENTER para ejecutar todo. Los resultados apareceran en la ventana Console

Funciones estadísticas básicas

```
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 desviación estándar
```

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


Funciones estadísticas básicas

`summary(x)` # calcula varias medidas estadísticas

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

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  
#Muestra las dos figuras en la misma pantalla
```

```
par(mfrow=c(1,2))
```

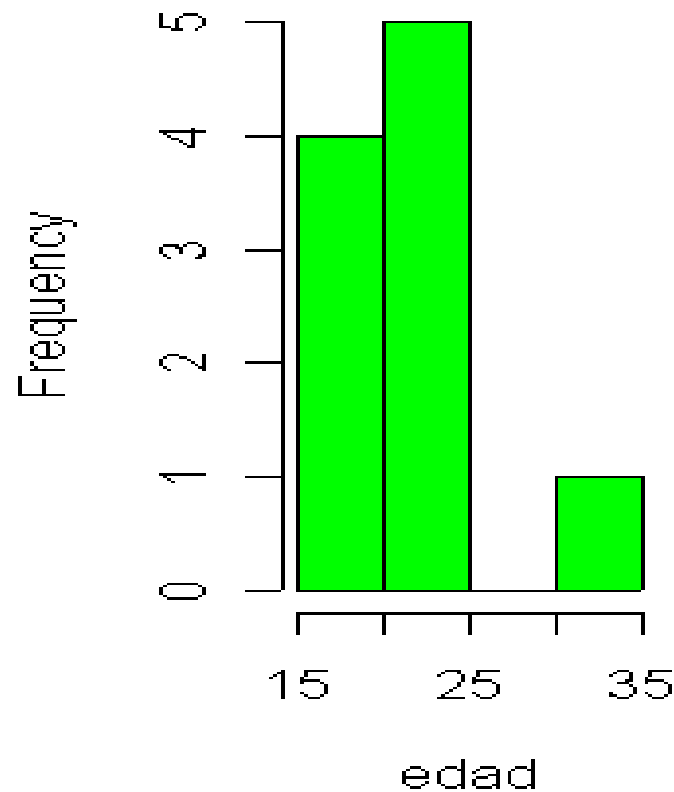
```
hist(edad,main="histograma de edad",col="green")
```

```
boxplot(edad,main="boxplot de edad", col=4)
```

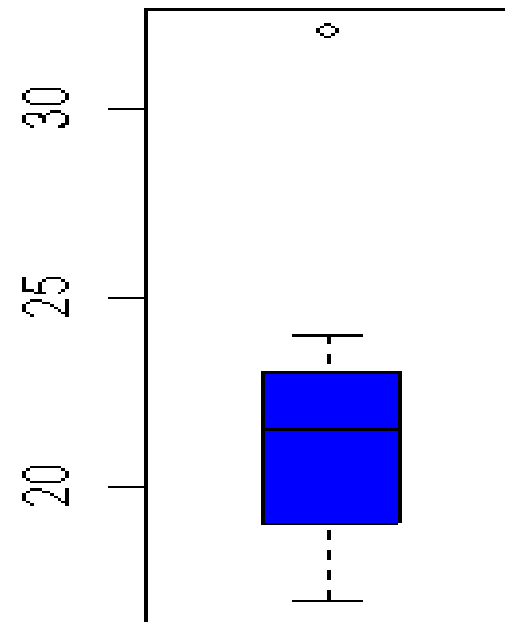
```
programas=c("bio","sico","adem","sico","bio","sico","adem","ade  
m","sico")
```

Graficas estadísticas univariadas

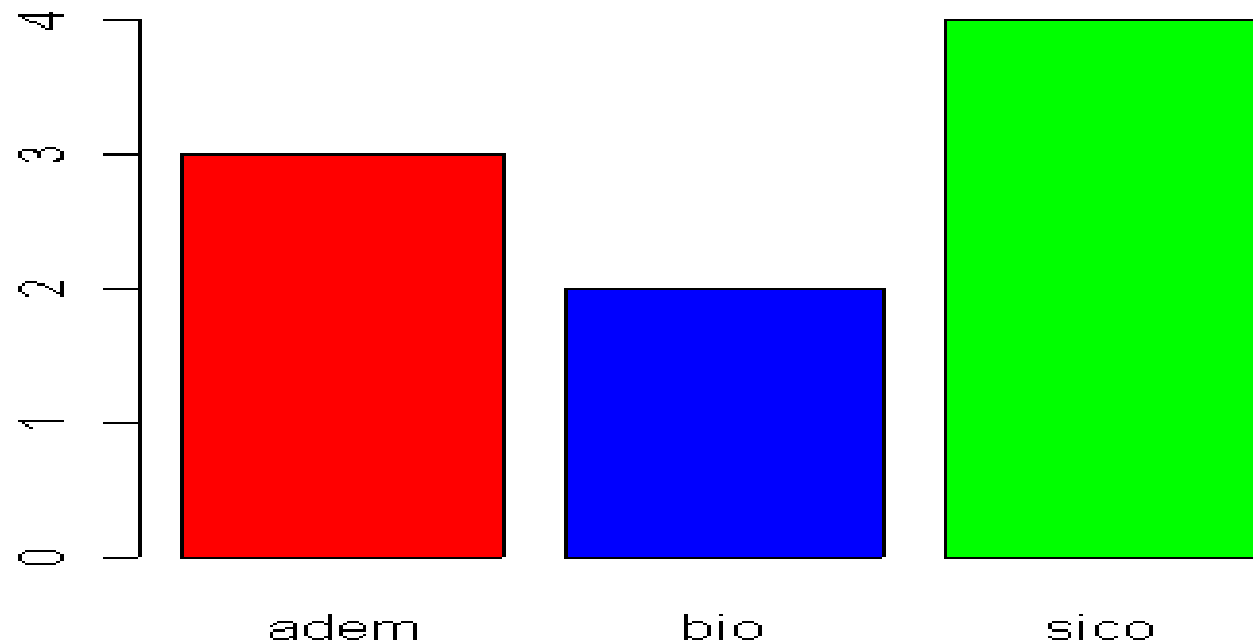
histograma de edad



boxplot de edad



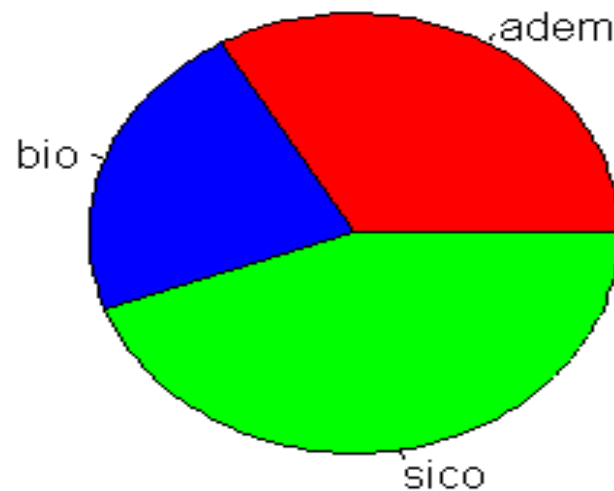
Grafica de Barras



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

Pie-charts

distribucion de estudiantes por programa



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

UPRM, Agosto 2018

Edgar Acuna

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")
```

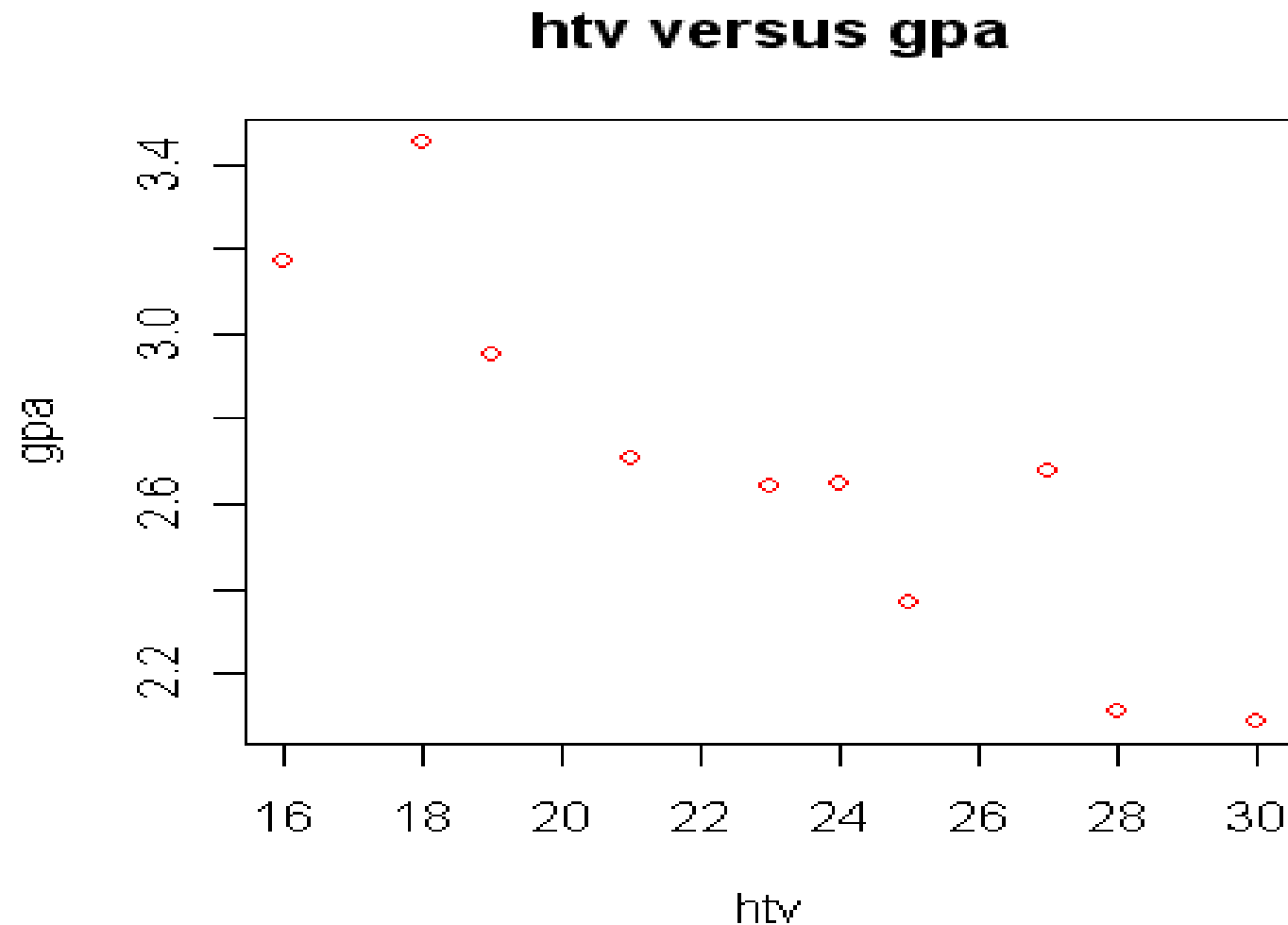
```
plot(htv,gpa,main="htv versus gpa",col="red")
```

```
boxplot(gpa~as.factor(genero),col="green")
```

```
title("GPA por genero")
```

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

Scatterplot



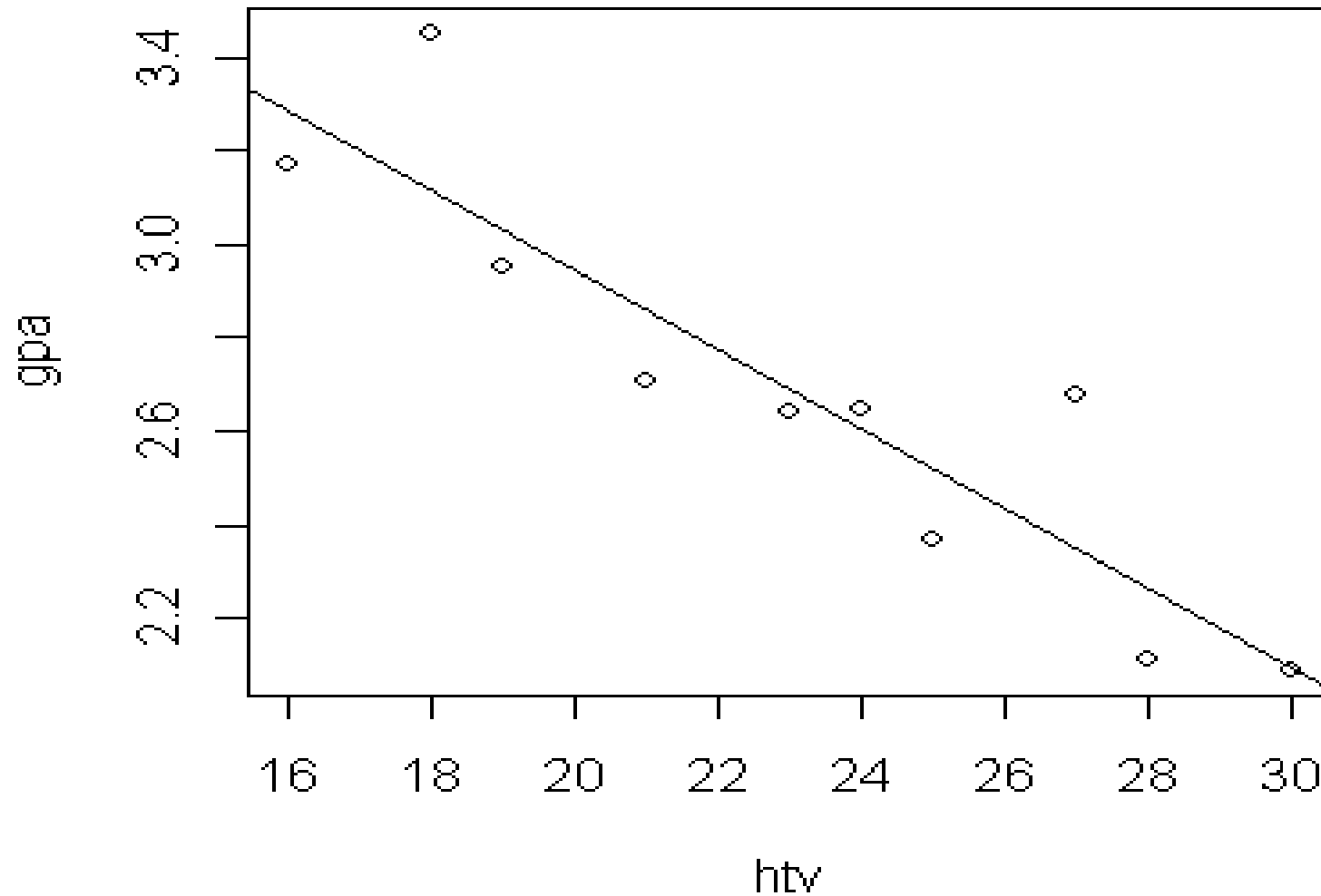
UPRM, Agosto 2018

Edgar Acuna

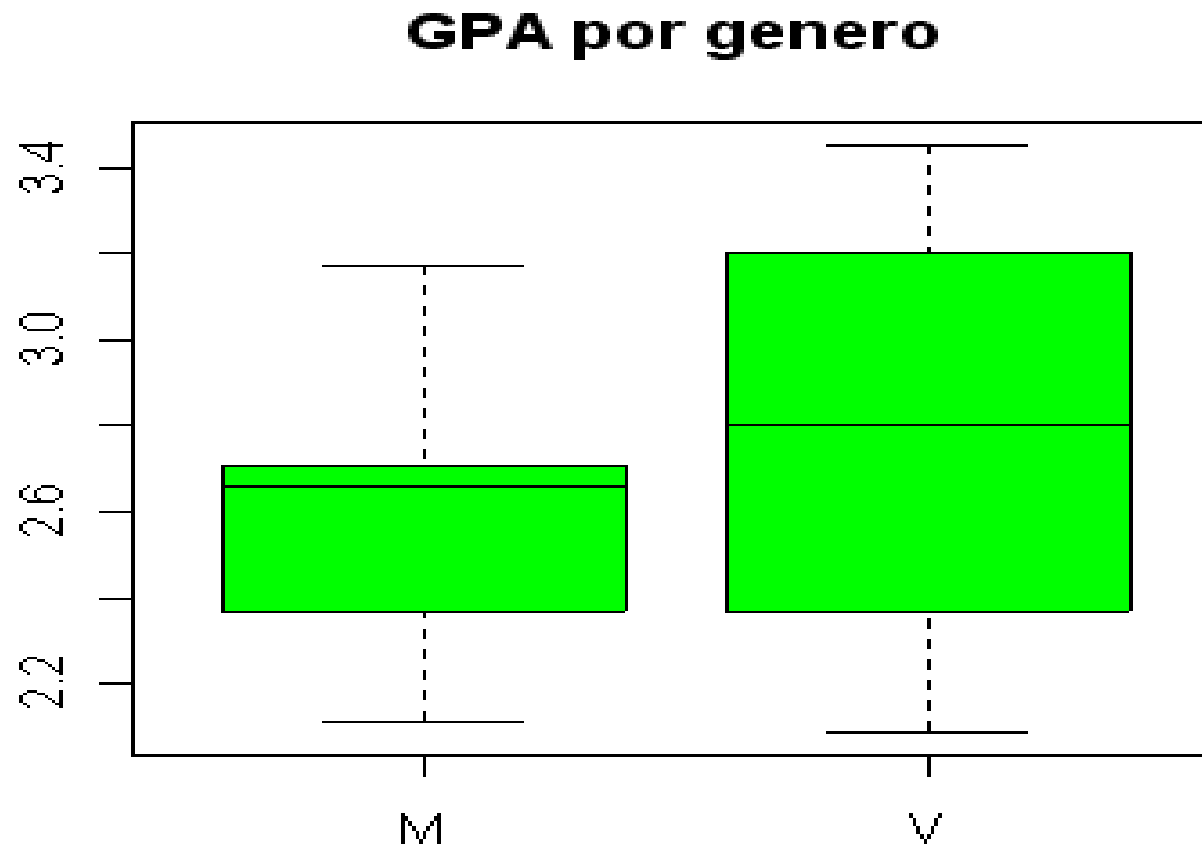
Regresion y correlacion lineal simple

```
> 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



Boxplots para comparar grupos



Comparando dos grupos

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

Welch Two Sample t-test

data: gpa by as.factor(genero)

$t = -0.5372$, $df = 4.592$, $p\text{-value} = 0.6161$

alternative hypothesis: true difference in means is not equal to
0

95 percent confidence interval:

-1.0156057 0.6722724

sample estimates:

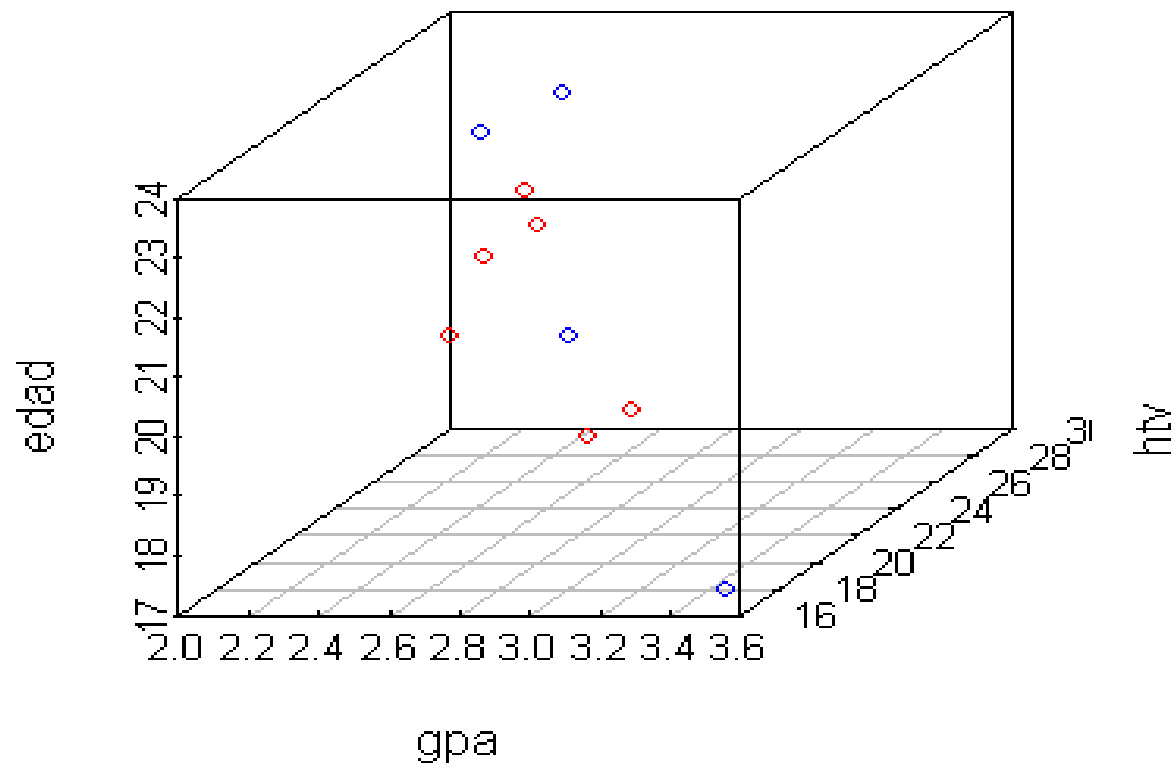
mean in group M mean in group V

2.613333 2.785000

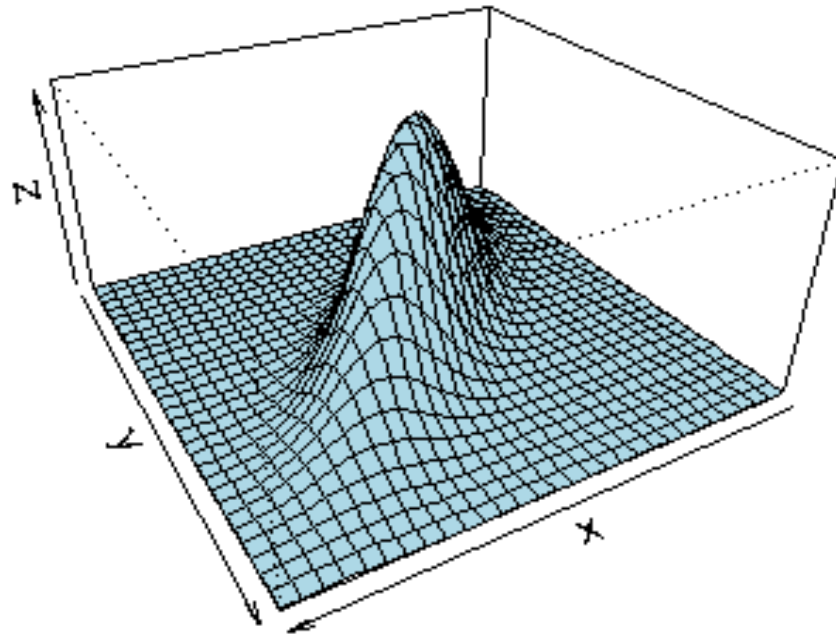
Plot en 3 dimensiones

```
color=c("red","blue")[as.factor(genero)]
> color
[1] "red" "blue" "blue" "red" "red" "blue" "red" "red"
"red" "blue"
>scatterplot3d(gpa,htv,edad,color) #requiere instalar el paquete
scatterplot3d
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 función normal bivariada
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")
```

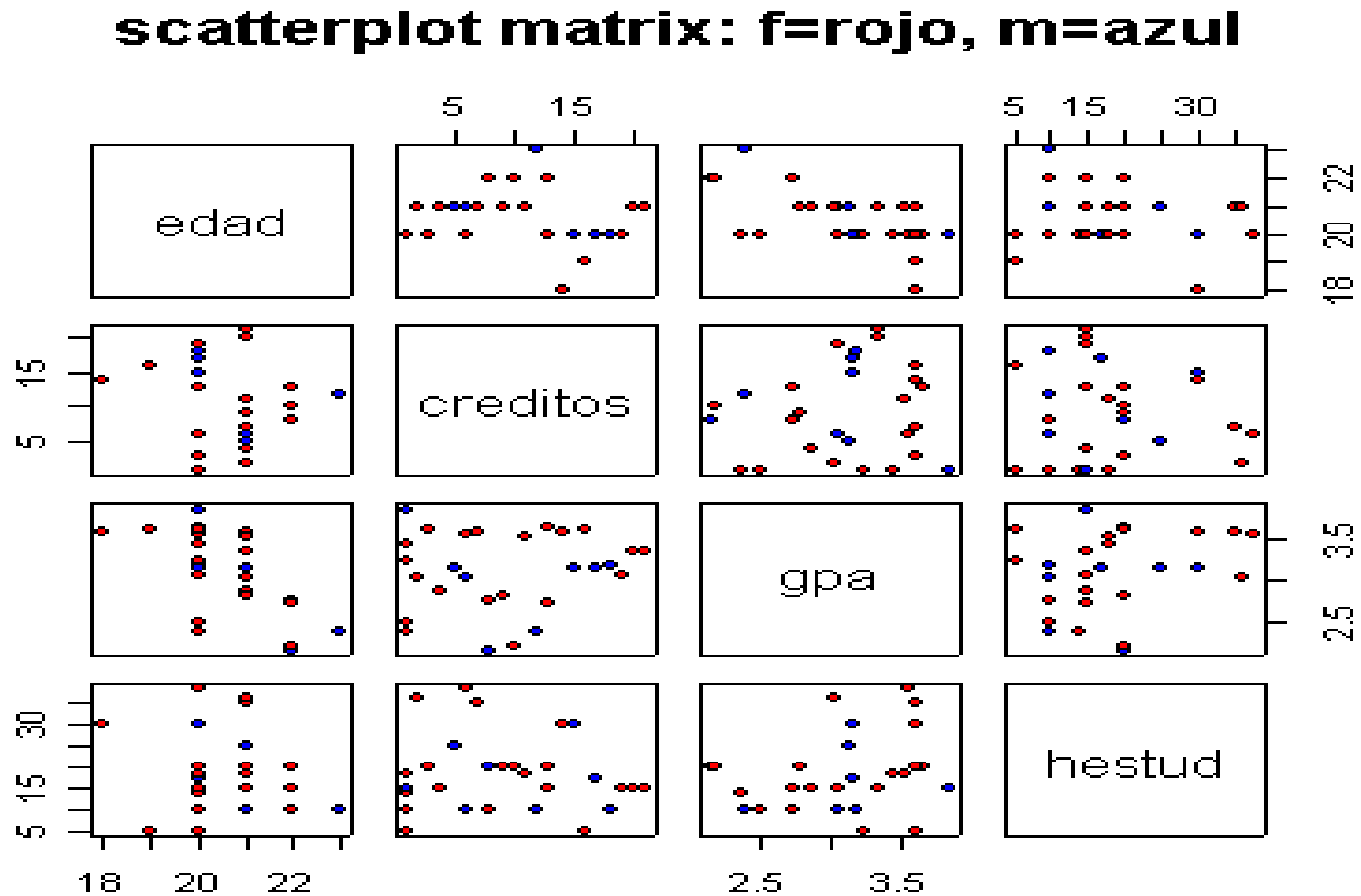
Plot en 3 dimensiones



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



Grafica en mas de tres dimensiones



```
pairs(clase[,c(1,5,6,8)], pch=21, bg=c("red","blue")[unclass(clase$sexo)])
```

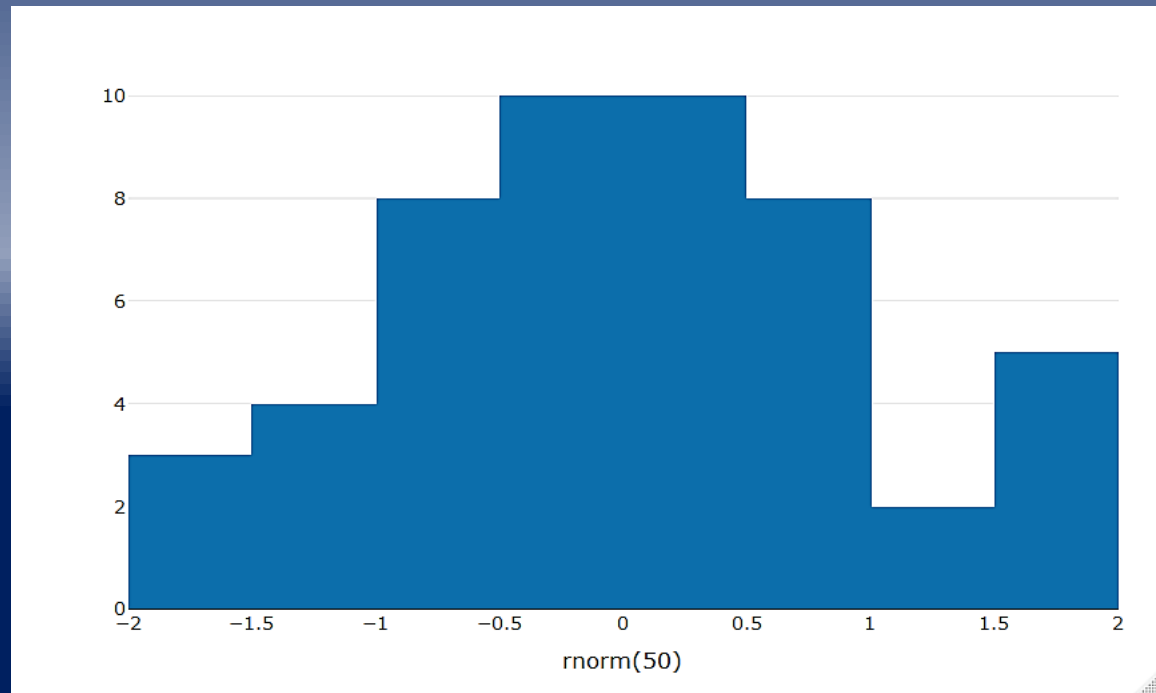
Librerías para gráficas en R

Ggplot2 (Hadley Wickham, 2009)

Plotly (2014)

Ggvis (Hadley Wickham , 2014)

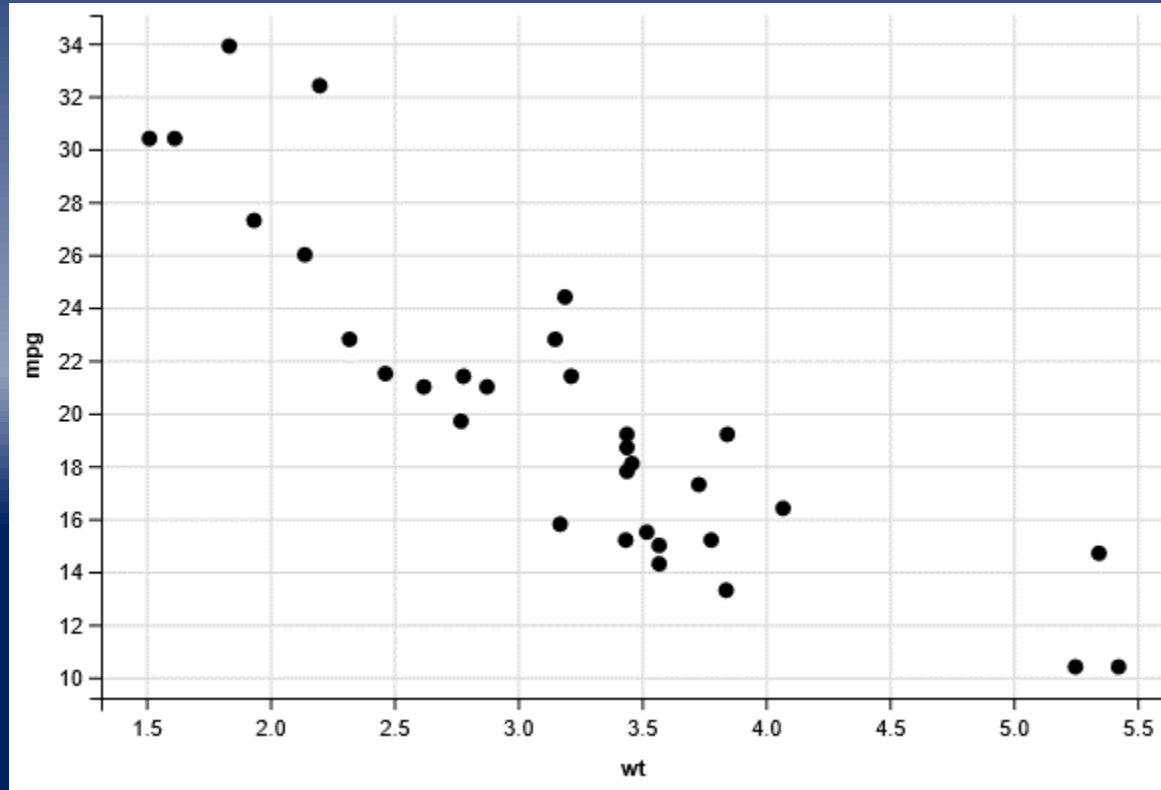
Histograma en Plotly



```
library(plotly)
p <- plot_ly(x = ~rnorm(50), type = "histogram")
p
```

UPRM, Agosto 2018
Edgar Acuna

Scatterplot en ggvis



```
library(ggvis)  
data(mtcars)  
ggvis(mtcars, ~wt, ~mpg)
```

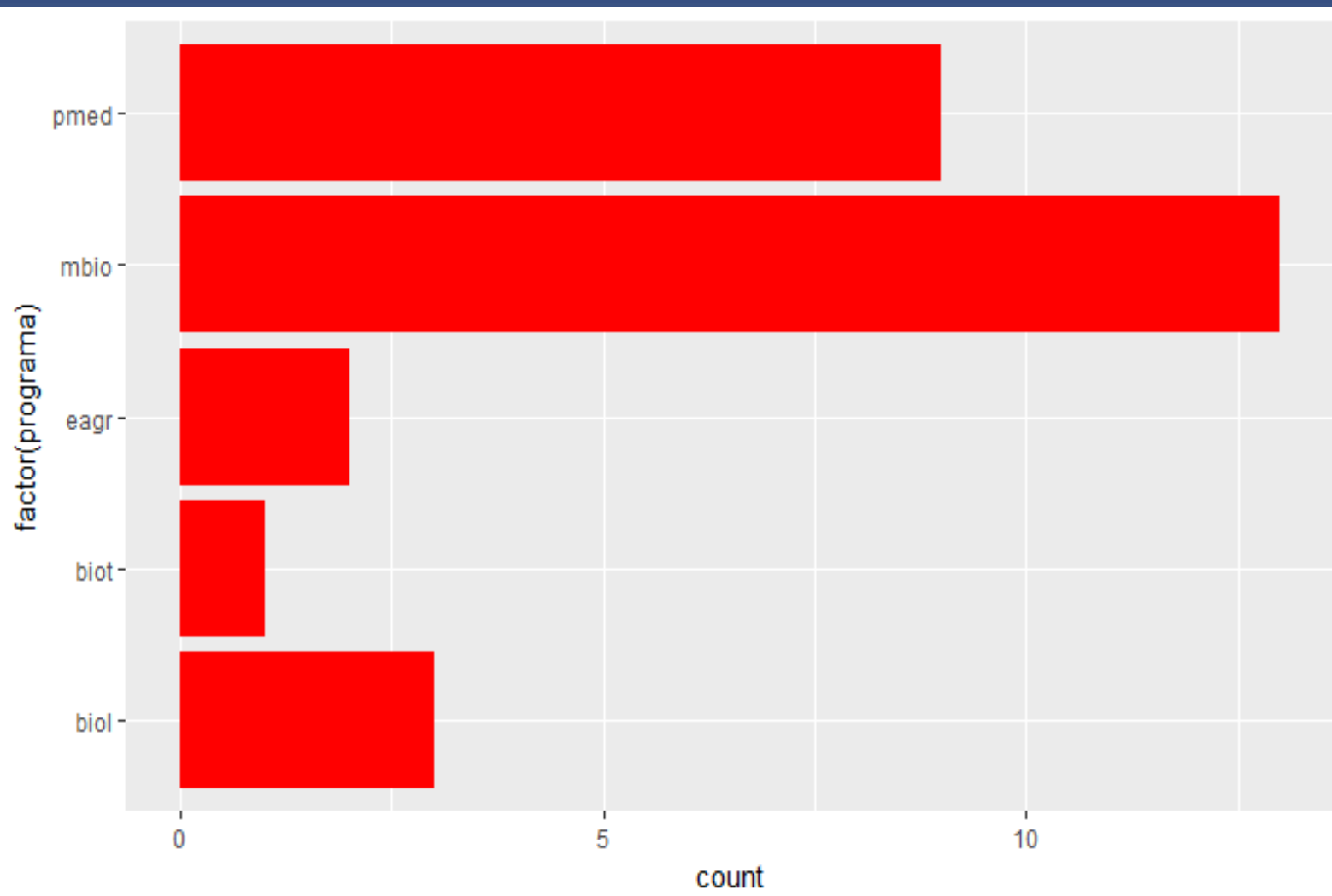
UPRM, Agosto 2018

Edgar Acuna

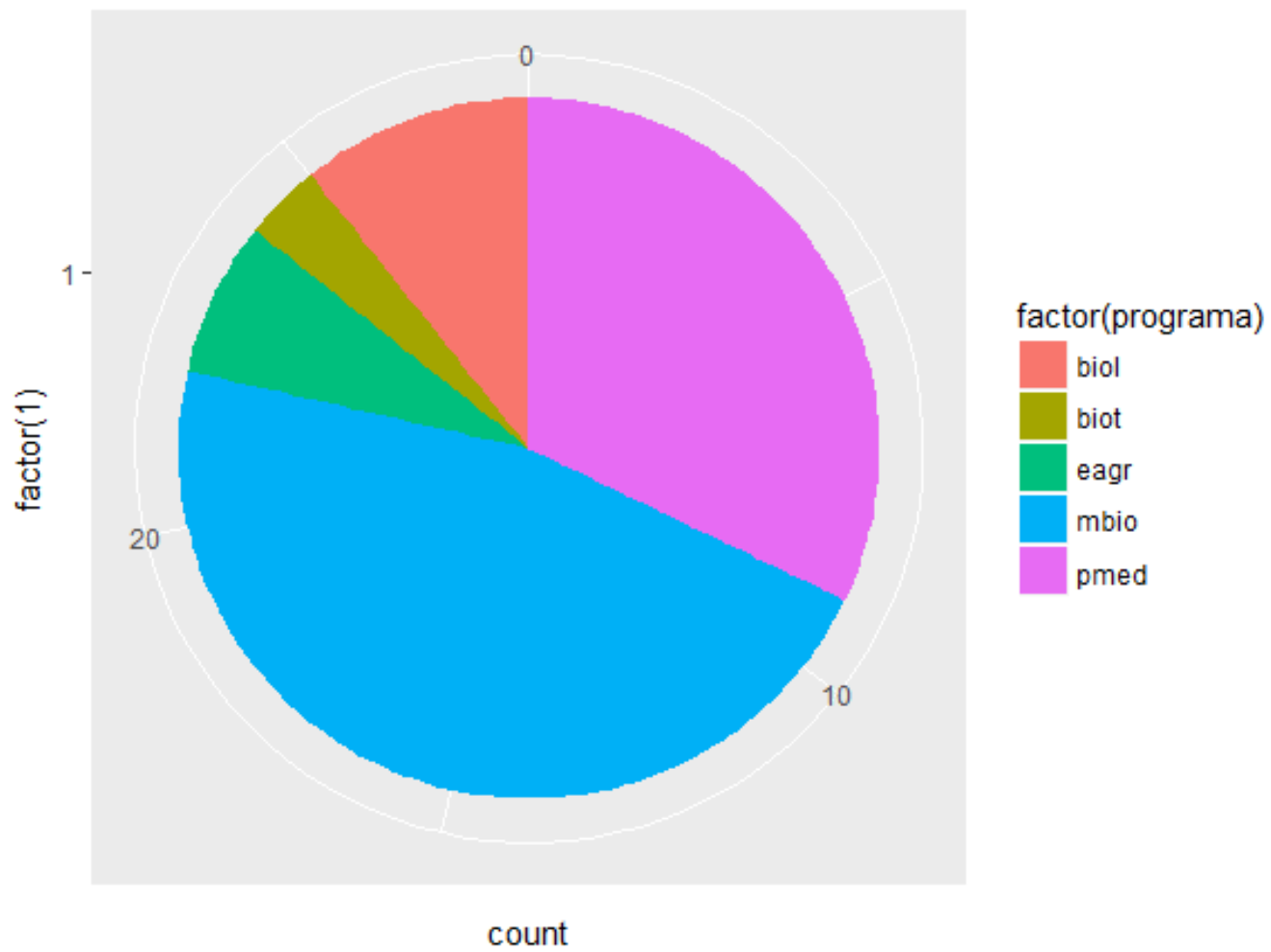
ggplot2

Graficas de barras y pie-chart

```
datos=read.table("http://academic.uprm.edu/eacuna/clase97.txt",
na.strings="*",header=T)
attach(datos)
#Usando la libreria ggplot2
library(ggplot2)
#grafica de barras verticales
ggplot(datos, aes(factor(programa)))+geom_bar(fill="green")
#Grafica de barras horizontales
ggplot(datos, aes(factor(programa)))+geom_bar(fill="red")+coord_flip()
#Grafica de pie-chart
barras=ggplot(datos,aes(x=factor(1),fill=factor(programa)))+geom_bar(
width=1)
barras+ coord_polar("y")
```



UPRM, Agosto 2018
Edgar Acuna



Boxplots usando ggplot

#Boxplot elegante

```
(datos,aes(x=1,y=edad))+geom_boxplot()+theme(axis.text.x=element_blank(),
```

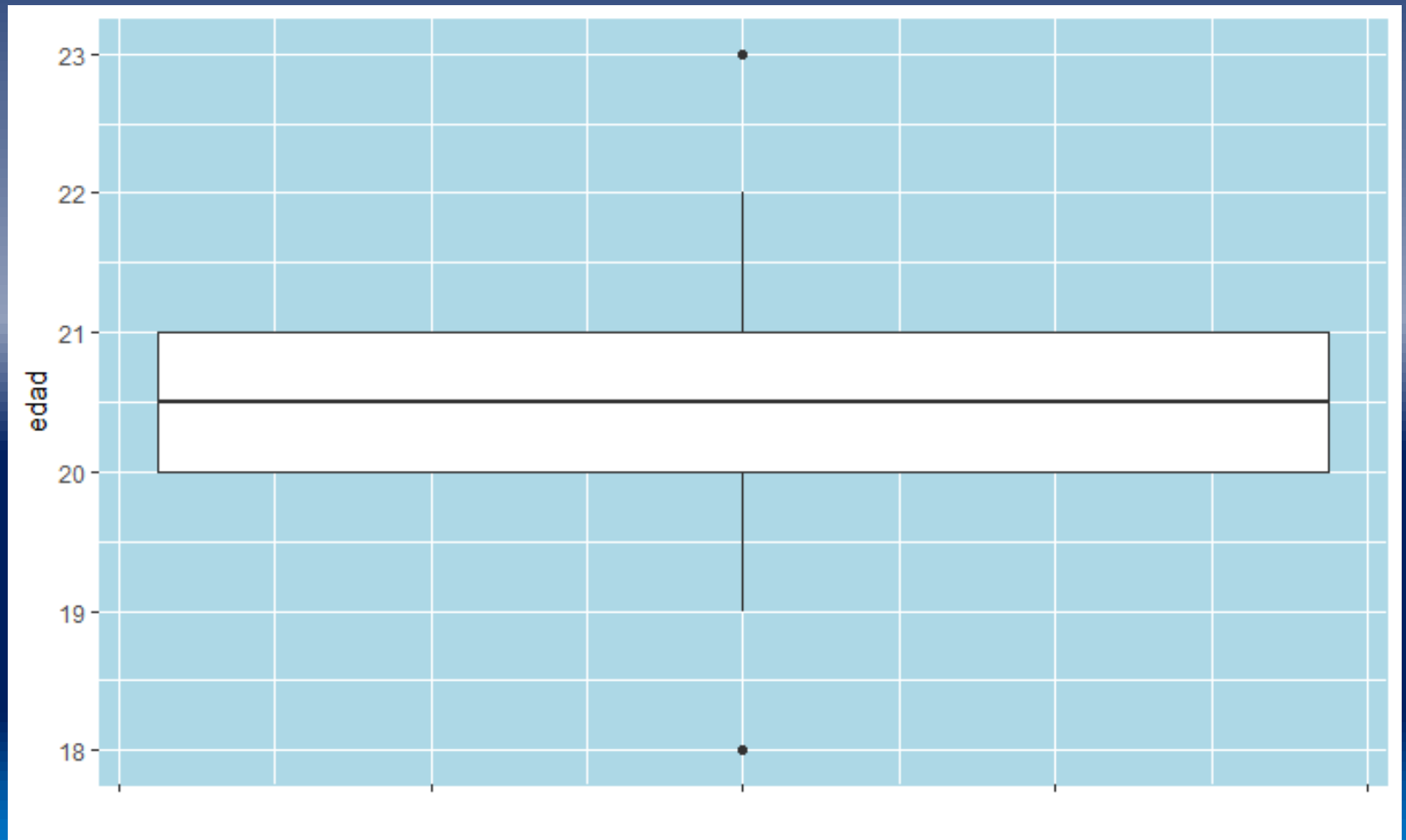
```
panel.background = element_rect(fill = "lightblue",colour = "lightblue",size = 0.5, linetype = "solid"),
```

```
panel.grid.major = element_line(size = 0.5, linetype = 'solid', colour = "white"),
```

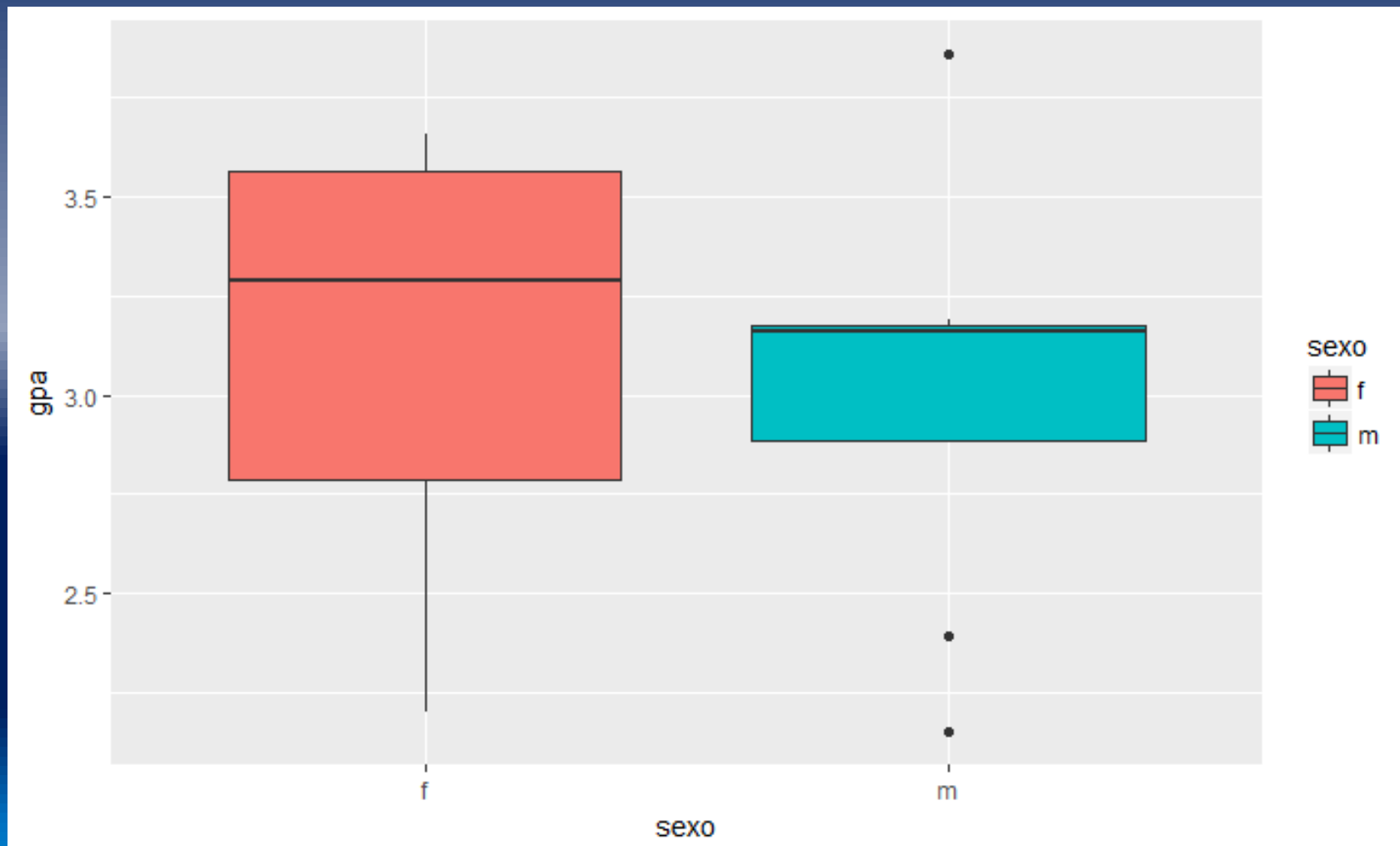
```
panel.grid.minor = element_line(size = 0.25, linetype = 'solid',colour = "white"))+labs(x=" ")
```

#Boxplot para comparar dos grupos

```
ggplot(datos,aes(x=sexo,y=gpa,fill=sexo))+geom_boxplot()
```



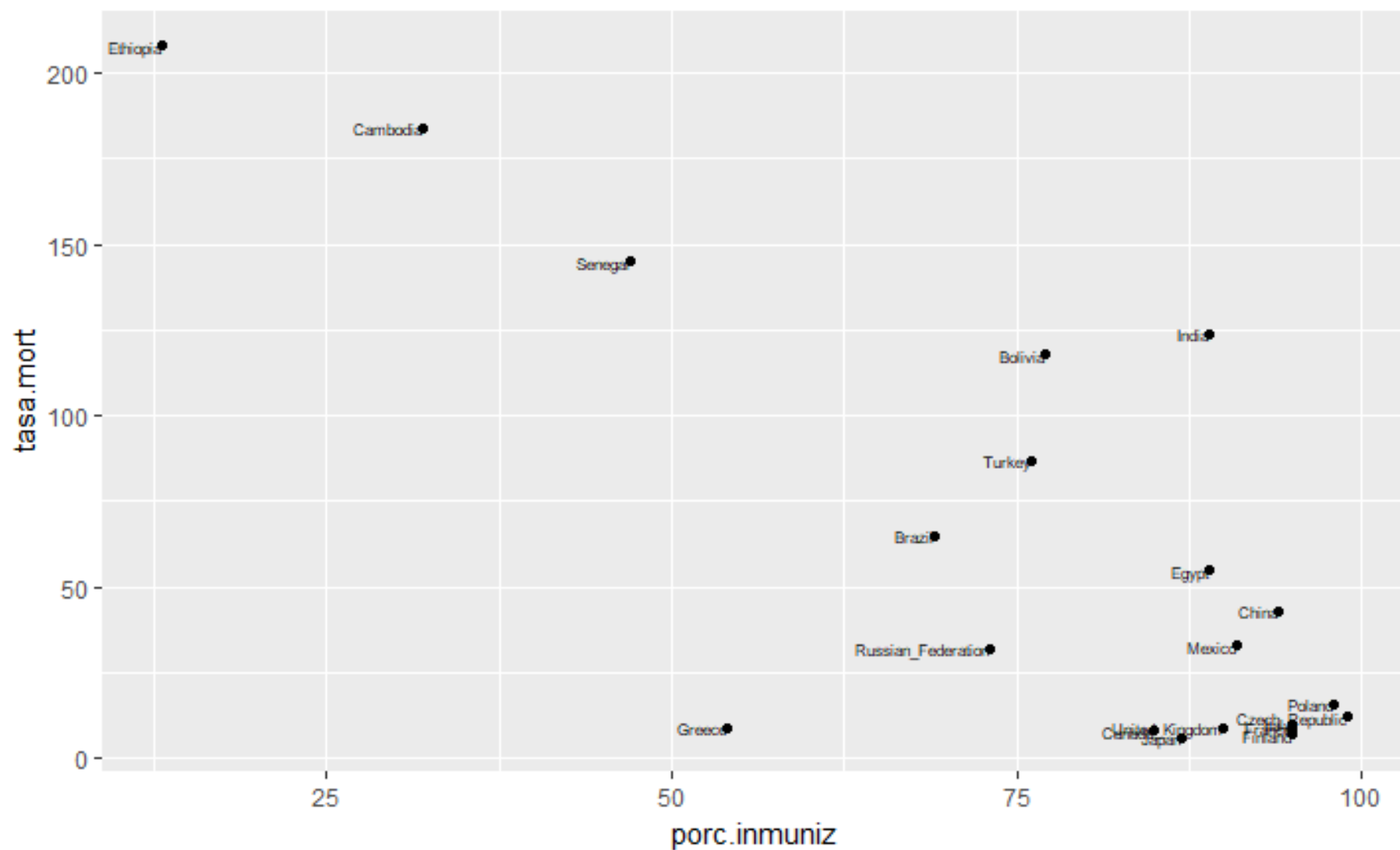
UPRM, Agosto 2018
Edgar Acuna

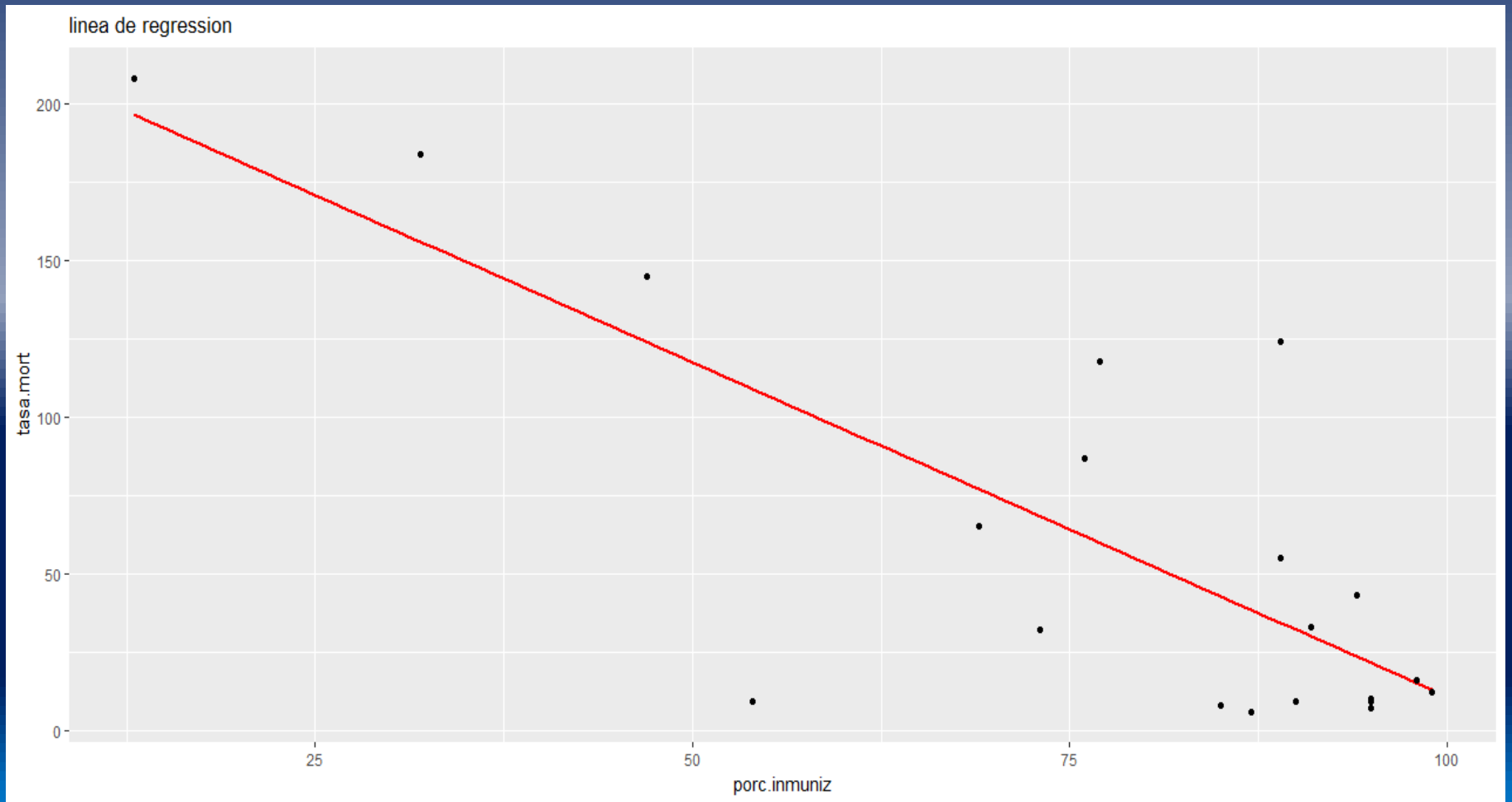


Scatterplot y regression lineal

```
datos1=read.table("http://academic.uprm.edu/eacuna/mortalidad.txt",header=T)
attach(datos1)
#Usando la libreria ggplot2
ggplot(datos1,aes(x=porc.inmuniz,y=tasa.mort,label=nacion))+
  geom_point()+geom_text(hjust=1,size=2)

#scatter plot incluyendo la linea de regresion
p=ggplot(datos1,aes(x=porc.inmuniz,y=tasa.mort))+geom_smooth(method="lm",se=FALSE,color="red",formula=y~x)+geom_point()
p +ggtitle("linea de regression")
```





ggplotGUI

Introducida en Julio el 2017.

Facilita el analisis grafico de bases de datos Requiere que tenga instalado la libreria shiny

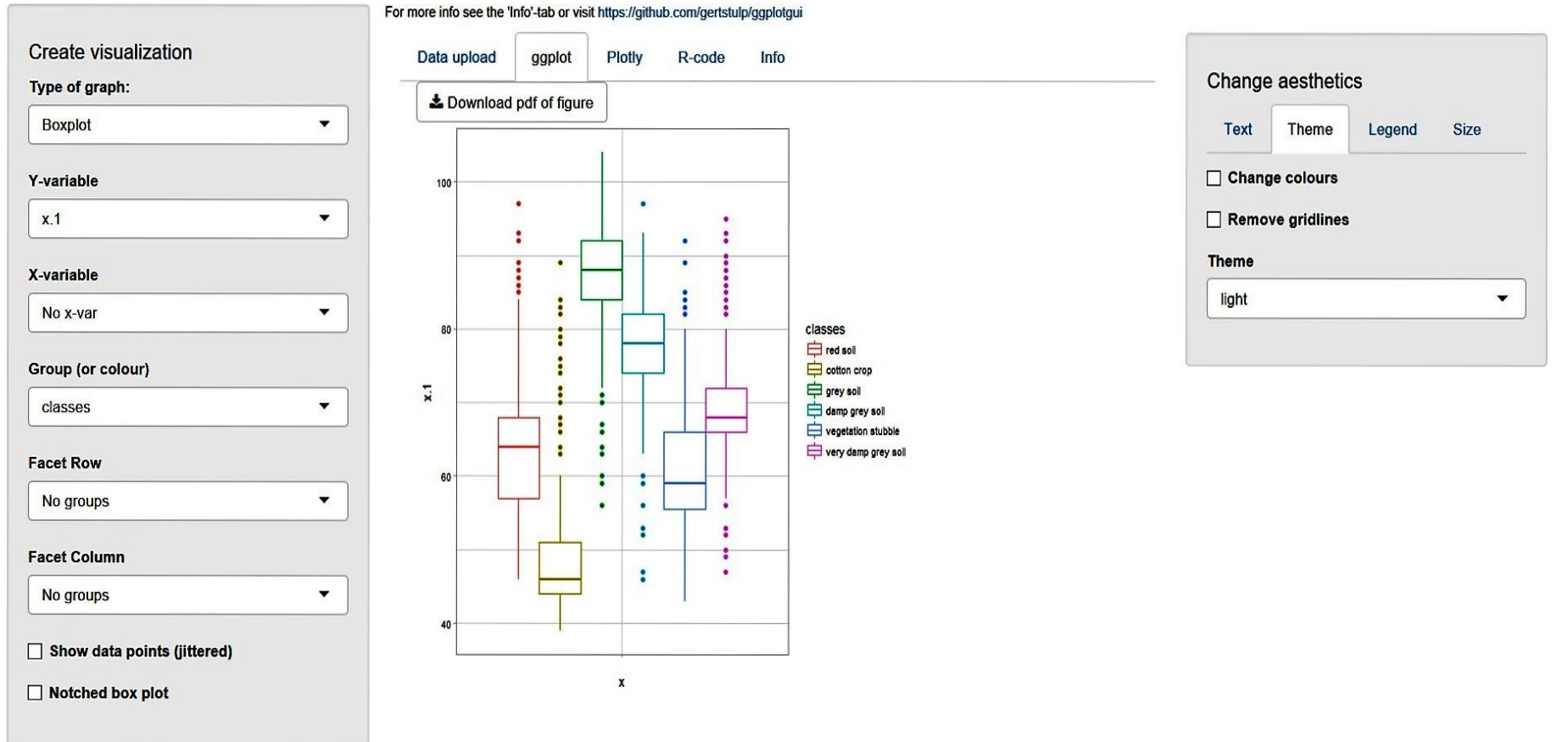
Despues de dar `library(shiny)` y `library(ggplotgui)`

Se activa usando el commando

`ggplot_shiny("filename")`. Se puede usar desde R o Rstudio.

Usar la opcion Data Upload para cargar su propios datos

ggplot GUI



Matrices y Dataframes

```
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

```
dim(xmat) #Tamano de la matriz(filas y columnas)
```

```
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
```

Anadiendo y quitando columns

```
> addcol=c(9,2)
>
> newmat=cbind(xmat,addcol)
> newmat
```

				addcol
[1,]	1	5	7	9
[2,]	3	4	8	2

```
> mat2=newmat[,-c(3,4)] #elimina columnas 3 y 4
> mat2
```

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

Anadiendo y quitando filas

```
> addrow=c(9,2,7)
> newmat=rbind(xmat,addrow)
> newmat
  V1 V2 V3
1  1  3  5
2  4  7  8
3  9  2  7
>mat2=newmat[-c(1,2),] #elimina  filas 1 y 2
> mat2
  V1 V2 V3
3  9  2  7
```

Haciendo operaciones con matrices

`m1+m2` #suma de matrices

`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 vectores 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

Convirtiendo una matriz en dataframe

```
xmat=as.data.frame(xmat)
```

```
xmat
```

```
  V1 V2 V3
```

```
1 85 86 87
```

```
2 85 91 98
```

```
> rownames(xmat) #nombres de las filas
```

```
[1] "1" "2"
```

```
> colnames(xmat) #nombres de las columnas
```

```
[1] "V1" "V2" "V3"
```

En un dataframe las columnas y las filas tienen nombres y se pueden hacer las mismas operaciones que con una matriz. Pero una matriz permite solo numeros mientras que en un dataframe se pueden incluir variables categoricas y missing values.

Subsetting un dataframe

```
clase=read.table("http://academic.uprm.edu/eacuna/clase97.txt",header=T)
```

```
clase.pub=subset(clase,escuela=="pub") #solo los estudiantes de publica
```

```
clase.pubyf= subset(clase,escuela=="pub" & sexo=="f") #solo los estudiantes de publica y mujeres
```

```
clase.biologpa= subset(clase,programa=="biol" | gpa>3.25) #solo los estudiantes de biol o con gpa mayor de 3.25
```

Subsetting un dataframe

#Extrae todos los estudiantes que son mujeres y que tienen promedio mas de 3.20

```
clase1=clase[which(clase$sexo=='f' & clase$gpa > 3.20),]
```

#Extrae todos los estudiantes que son mujeres o que tienen promedio

```
clase2=clase[which(clase$sexo=='f' | clase$gpa > 3.20),]
```

#divide al conjunto clase en dos los datos de las mujeres y los varones

```
split(clase,as.factor(clase$sexo))
```

Uso de For

```
for(x in 1:4){  
cat( x,"al cuadrado es", x^2, "\n") }
```

Output:

```
1 al cuadrado es 1  
2 al cuadrado es 4  
3 al cuadrado es 9  
4 al cuadrado es 16
```

```
x=c(3,7,12)  
for( i in x){  
cat( i,"al cuadrado es", i^2, "\n") }
```

```
3 al cuadrado es 9  
7 al cuadrado es 49  
12 al cuadrado es 144
```

Uso del condicional If

```
gpa = 1.4
if(gpa >= 2.5)
{cat("Bienvenido al Colegio! ")}else
{cat("su solicitud fue denegada")}
```

Notar que el else tiene que estar en la misma linea del bracket que Cierra el if

```
Ana=3
```

```
Rosa=25
```

```
if (Ana <= 5 && Rosa >= 10 || Rosa == 500 && Ana != 5)
{print ("Ana and Rosa")}
```


Haciendo operaciones repetidas con while

```
number = 1
while(number < 200){
    print(number)
    number = number * 2}
```

```
1] 1
[1] 2
[1] 4
[1] 8
[1] 16
[1] 32
[1] 64
[1] 128
```

Construyendo funciones I

```
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
```

Construyendo funciones II

```
tablafreq=function (x)
{#Tabla de frecuencias para datos discretos
n=length(x)
frec.abs=table(x)
frec.rel.porc=table(x)*100/n
frec.abs.acum=cumsum(frec.abs)
frec.rel.acum=cumsum(frec.rel.porc)
tabla=cbind(frec.abs,frec.rel.porc,frec.abs.acum,frec.rel.acum)
return(tabla)
}
```

Paquetes(librerias)

- Un Paquete es un conjunto de funciones que realizan ciertas tareas específicas y que han sido construidas por diversos usuarios de R.
- Hay mas de 11,000 paquetes disponibles en el website de R.
- La mayoría 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 el paquete Rcmdr
```

```
>help(package="paquete") # da ayuda de como usar el paquete
```

Algunos paquetes disponibles

<u>fBasics</u>	Financial Software Collection - fBasics
<u>foreign</u>	Read Data Stored by Minitab, S, SAS, SPSS, Stata, Systat, dBase,
<u>cluster</u>	Funciones para hacer clustering
<u>dplyr</u>	Funciones para subsetting, summarizing, y juntar datasets
<u>ggplot2</u>	Funciones para hacer graficas de alta calidad
<u>lubridate</u>	Funciones para trabajar facilnente con datos que tienen fecha y hora
<u>manipulater</u>	Funciones para plots interactivos en Rstudio
<u>Rcmdr</u>	R Commander
<u>Rcpp</u>	Funciones para llamar en R programs escritos en c++
<u>vcd</u>	Visualizing Categorical Data

Construyendo librerías

Aquí solo explicaremos como hacer una librería local. Hacer una librería para ponerlo en el cran es mas tedioso

1. Descargar Rtools del cran de R
2. Modificar su path. Abrir Control Panel, luego ir a System and Security y luego a System > Advanced System Settings > Environment Variables. Hallar la variable "Path" y añadir C:\Program Files\R\R\ - 3.3\bin\x64;C:\Rtools\bin

2. 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 ensamblar y ponerlas todas a la vez en una librería llamada "mipaquete", la cual va a estar localizada en el directorio ("c:\Rpaquetes". Supongamos además que se usan 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 librerías

Por ejemplo

```
package.skeleton(name="mipaquete",  
list=c("moda","tablafreq"), path="c://Rpaquetes").
```

En el subdirectorío mipaquete van a crearse dos subdirectoríos R y man, en uno están los códigos de las funciones y en otra la ayuda

3- Finalmente se construye el paquete dando en la Ventana de terminal el comando

```
Rcmd build --binary mipaquete
```

Seguido del comando

```
Rcmd INSTALL mynewpackage_1.0.tar.gz
```

4- Después de esto se puede entrar a R y dar el comando

```
library(mipaquete)
```

La Libreria Dprep

Package: dprep

Type: Package

Title: Data Pre-Processing and Visualization Functions for Classification Version: 3.0.2

Date: 2015-11-14

Author: Edgar Acuna and the CASTLE research group at The University of Puerto Rico-Mayaguez

Maintainer: Edgar Acuna edgar.acuna@upr.edu

Description: Data preprocessing techniques for classification. Functions for normalization, handling of missing values, discretization, outlier detection, feature selection, and data visualization are included.

Depends: R ($\geq 3.1.0$), graphics, stats Imports: MASS, e1071, class, nnet, rpart, FNN, StatMatch, rgl, methods

License: GPL

LazyLoad: yes

NeedsCompilation: yes

Packaged: 2015-11-24 00:51:58 UTC; Edgar

Repository: CRAN

Date/Publication: 2015-11-24 07:46:38

Built: R 3.4.0; x86_64-w64-mingw32; 2017-06-21 03:40:56 UTC; windows

Archs: i386, x64

UPRM, Agosto 2018

Edgar Acuna

Aplicaciones Web Shiny

Crear un archivo server.R y otro ui.R dentro de un directorio

Ver varios ejemplos en la pagina de shiny

Aqui les muestro uno de mi proyecto

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.