Introducción a R

Modelos no paramétricos y de regresión

Enrique Reyes

08 de febrero de 2018

Bases de datos

Internas

```
#Dataframe2
data("iris")
a<-iris
#visualizamos los primeros 6
head(a)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
#verificamos que sea un objeto de base de datos
class(a)
```

```
## [1] "data.frame"
```

```
#explotamos la base de datos
attach(a)
Sepal.Length
```

```
##   [1] 5.1 4.9 4.7 4.6 5.0 5.4 4.6 5.0 4.4 4.9 5.4 4.8 4.8 4.3 5.8 5.7 5.4
##  [18] 5.1 5.7 5.1 5.4 5.1 4.6 5.1 4.8 5.0 5.0 5.2 5.2 4.7 4.8 5.4 5.2 5.5
##  [35] 4.9 5.0 5.5 4.9 4.4 5.1 5.0 4.5 4.4 5.0 5.1 4.8 5.1 4.6 5.3 5.0 7.0
##  [52] 6.4 6.9 5.5 6.5 5.7 6.3 4.9 6.6 5.2 5.0 5.9 6.0 6.1 5.6 6.7 5.6 5.8
##  [69] 6.2 5.6 5.9 6.1 6.3 6.1 6.4 6.6 6.8 6.7 6.0 5.7 5.5 5.5 5.8 6.0 5.4
##  [86] 6.0 6.7 6.3 5.6 5.5 5.5 6.1 5.8 5.0 5.6 5.7 5.7 6.2 5.1 5.7 6.3 5.8
## [103] 7.1 6.3 6.5 7.6 4.9 7.3 6.7 7.2 6.5 6.4 6.8 5.7 5.8 6.4 6.5 7.7 7.7
## [120] 6.0 6.9 5.6 7.7 6.3 6.7 7.2 6.2 6.1 6.4 7.2 7.4 7.9 6.4 6.3 6.1 7.7
## [137] 6.3 6.4 6.0 6.9 6.7 6.9 5.8 6.8 6.7 6.7 6.3 6.5 6.2 5.9
```

```
#reintegrando las variables
detach(a)
a$Sepal.Length
```

```
##   [1] 5.1 4.9 4.7 4.6 5.0 5.4 4.6 5.0 4.4 4.9 5.4 4.8 4.8 4.3 5.8 5.7 5.4
##  [18] 5.1 5.7 5.1 5.4 5.1 4.6 5.1 4.8 5.0 5.0 5.2 5.2 4.7 4.8 5.4 5.2 5.5
##  [35] 4.9 5.0 5.5 4.9 4.4 5.1 5.0 4.5 4.4 5.0 5.1 4.8 5.1 4.6 5.3 5.0 7.0
##  [52] 6.4 6.9 5.5 6.5 5.7 6.3 4.9 6.6 5.2 5.0 5.9 6.0 6.1 5.6 6.7 5.6 5.8
##  [69] 6.2 5.6 5.9 6.1 6.3 6.1 6.4 6.6 6.8 6.7 6.0 5.7 5.5 5.5 5.8 6.0 5.4
##  [86] 6.0 6.7 6.3 5.6 5.5 5.5 6.1 5.8 5.0 5.6 5.7 5.7 6.2 5.1 5.7 6.3 5.8
## [103] 7.1 6.3 6.5 7.6 4.9 7.3 6.7 7.2 6.5 6.4 6.8 5.7 5.8 6.4 6.5 7.7 7.7
## [120] 6.0 6.9 5.6 7.7 6.3 6.7 7.2 6.2 6.1 6.4 7.2 7.4 7.9 6.4 6.3 6.1 7.7
```

```
## [137] 6.3 6.4 6.0 6.9 6.7 6.9 5.8 6.8 6.7 6.7 6.3 6.5 6.2 5.9
```

```
#la extracción de informacion es igual
a[1,]
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
```

```
a[,1]
```

```
##   [1] 5.1 4.9 4.7 4.6 5.0 5.4 4.6 5.0 4.4 4.9 5.4 4.8 4.8 4.3 5.8 5.7 5.4
##  [18] 5.1 5.7 5.1 5.4 5.1 4.6 5.1 4.8 5.0 5.0 5.2 5.2 4.7 4.8 5.4 5.2 5.5
##  [35] 4.9 5.0 5.5 4.9 4.4 5.1 5.0 4.5 4.4 5.0 5.1 4.8 5.1 4.6 5.3 5.0 7.0
##  [52] 6.4 6.9 5.5 6.5 5.7 6.3 4.9 6.6 5.2 5.0 5.9 6.0 6.1 5.6 6.7 5.6 5.8
##  [69] 6.2 5.6 5.9 6.1 6.3 6.1 6.4 6.6 6.8 6.7 6.0 5.7 5.5 5.5 5.8 6.0 5.4
##  [86] 6.0 6.7 6.3 5.6 5.5 5.5 6.1 5.8 5.0 5.6 5.7 5.7 6.2 5.1 5.7 6.3 5.8
## [103] 7.1 6.3 6.5 7.6 4.9 7.3 6.7 7.2 6.5 6.4 6.8 5.7 5.8 6.4 6.5 7.7 7.7
## [120] 6.0 6.9 5.6 7.7 6.3 6.7 7.2 6.2 6.1 6.4 7.2 7.4 7.9 6.4 6.3 6.1 7.7
## [137] 6.3 6.4 6.0 6.9 6.7 6.9 5.8 6.8 6.7 6.7 6.3 6.5 6.2 5.9
```

```
a[2,3]
```

```
## [1] 1.4
```

```
a[4,"Sepal.Width"]
```

```
## [1] 3.1
```

```
a$Sepal.Length[1]
```

```
## [1] 5.1
```

Creadas

```
#Creamos dos vectores
y <- c("Aguascalientes", "Baja California", "Baja California", "Chihuahua",
       "Zacatecas", "Zacatecas", "Zacatecas", "Baja California", "Chihuahua")
z <- c("H", "M", "M", "M", "H", "M", "M", "M", "M")
x <- c(1, 2, 6, 4, 10, 20, 1, 15, 0)
BD <- data.frame(EDO = y, sexo = z, IDX = x)
names(BD)
```

```
## [1] "EDO"  "sexo" "IDX"
```

```
# Para manejar las variables de una base de datos i.e. columnas
table(BD$EDO)
```

```
##
##  Aguascalientes Baja California       Chihuahua       Zacatecas
##               1               3               2               3
```

```
BD$EDO
```

```
## [1] Aguascalientes  Baja California Baja California Chihuahua
## [5] Zacatecas       Zacatecas       Zacatecas       Baja California
## [9] Chihuahua
## Levels: Aguascalientes Baja California Chihuahua Zacatecas
```

```
BD$sexo
```

```
## [1] H M M M H M M M M
```

```
## Levels: H M
```

```
BD$IDX
```

```
## [1]  1  2  6  4 10 20  1 15  0
```

```
# De esta manera ya podemos manejar las columnas como vectores!
table(BD$EDO)
```

```
##
##  Aguascalientes Baja California       Chihuahua       Zacatecas
##               1               3               2               3
```

```
table(BD$sexo, BD$EDO)
```

```
##
##    Aguascalientes Baja California Chihuahua Zacatecas
##   H              1               0         0         1
##   M              0               3         2         2
```

```
mean(BD$IDX)
```

```
## [1] 6.555556
```

```
sd(BD$IDX)
```

```
## [1] 7.037597
```

```
summary(BD)
```

```
##               EDO    sexo       IDX
##  Aguascalientes :1   H:2   Min.   : 0.000
##  Baja California:3   M:7   1st Qu.: 1.000
##  Chihuahua      :2         Median : 4.000
##  Zacatecas      :3         Mean   : 6.556
##                            3rd Qu.:10.000
##                            Max.   :20.000
```

```
BD$sexo
```

```
## [1] H M M M H M M M M
## Levels: H M
```

```
# Para extraer informacion de una base de datos
BD[1,]
```

```
##              EDO sexo IDX
## 1 Aguascalientes    H   1
```

```
BD[,1] # equivalente a BD$EDO
```

```
## [1] Aguascalientes  Baja California Baja California Chihuahua
## [5] Zacatecas       Zacatecas       Zacatecas       Baja California
## [9] Chihuahua
## Levels: Aguascalientes Baja California Chihuahua Zacatecas
```

```
BD[5,3]
```

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

```
BD[5:7,]
```

```
##         EDO sexo IDX
```

```
## 5 Zacatecas    H  10
## 6 Zacatecas    M  20
## 7 Zacatecas    M   1
```

```
# Filtrando informacion
id <- BD$EDO == "Zacatecas"
BD_ZAC <- BD[id,]
mean(BD_ZAC$IDX)
```

```
## [1] 10.33333
```

```
tapply(BD$IDX, BD$EDO, median)
```

```
##  Aguascalientes Baja California       Chihuahua       Zacatecas
##               1               6               2              10
```

```
id2 <- BD$sexo == "H"
BD_H <- BD[id2,]
id3 <- BD$IDX <= 5
BD_ID_5 <- BD[id3,] ; BD_ID_5
```

```
##               EDO sexo IDX
## 1  Aguascalientes    H   1
## 2 Baja California    M   2
## 4       Chihuahua    M   4
## 7       Zacatecas    M   1
## 9       Chihuahua    M   0
```

```
tb <- table(BD_ID_5$EDO, BD_ID_5$sexo) ; tb
```

```
##
##                   H M
##   Aguascalientes  1 0
##   Baja California 0 1
##   Chihuahua       0 2
##   Zacatecas       0 1
```

Paqueterías en *R*

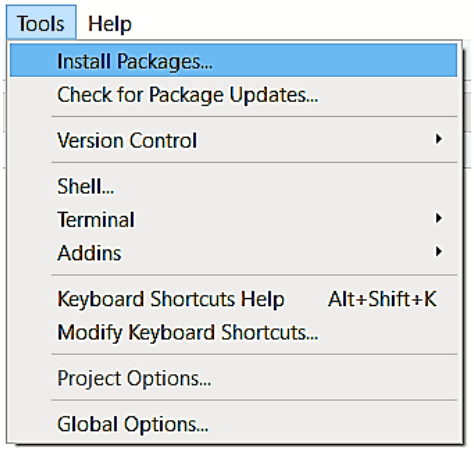
Las paqueterías son de gran utilidad porque están conformadas por una serie de funciones que nos facilitarán la obtención de resultados de manera rápida y efectiva, además si llegasemos a tener dudas sobre su creación, podemos revisar su documentación desde la sección de ayuda.

Algunas paqueterías recomendadas para el curso y su forma de instalación en *R* son:

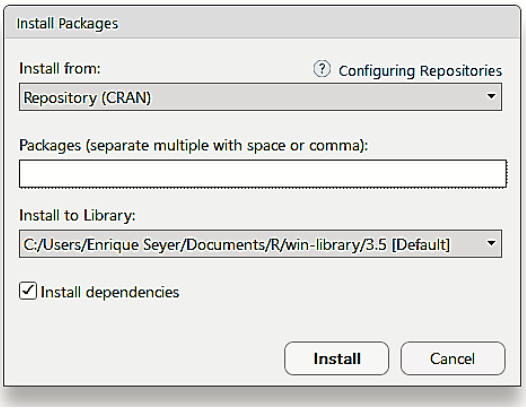
```
install.packages("sqldf") #Carga base de datos y se realizan consultas
install.packages("sqlite") #Carga base de datos y se realizan consultas
install.packages("RODBC") #Carga base de datos y se realizan consultas
install.packages("RPostgresSQL") #Carga base de datos y se realizan consultas
install.packages("foreing") #Carga archivos de SPSS, SAS, Stata, DBF, Epi info, Minitab
install.packages("plyr") #Extracción de datos y aplicación de funciones a grupos
install.packages("dplyr") #Trabaja con fechas
install.packages("reshape2") #Transformación de datos
install.packages("ggplot2") #Genera graficos
install.packages("rgl") #Gráficos en 3D
install.packages("forecats") #Formateo de datos y creación modelos
install.packages("knitr") #Genera codigos en Latex y Html
install.packages("xtable") #Exporta datos en html y Latex
```

```r
install.packages("actuar") #Varias distribuciones
install.packages("MASS") #Análisis multivariado
install.packages("xlsx") #Lectura de archivos de Excel
#Esta última paquetería necesita tener Java instalado en el equipo
install.packages("alr4") #Conjunto de datos del libro Applied Linear Regression
install.packages("lmtest") #Pruebas de validación de supuestos
```

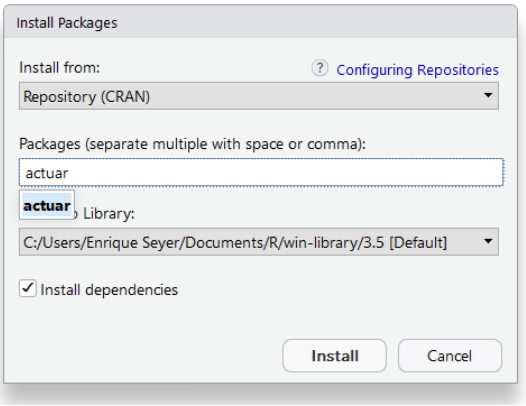
Otro método para instalas una paquetería es irse a la pestaña *Tools*, desplegarla y seleccionar la primera opción *Install packages*



Eso les desplegará la siguiente ventana



Sólo se tiene que colocar el nombre del paquete en la sección *Packages*



Como se puede observar se autocompleta el nombre de la paquetería, así que una vez seleccionado, sólo debemos presionar el botón *Install* y listo, se instalará el paquete.

La forma de activar las funciones y bases de datos asociadas a cada paquetería es:

```
library(alr4) #Con este comando se activan todas las funciones y bases
#No basta con instalar la paquetería, se tiene que habilitar
```

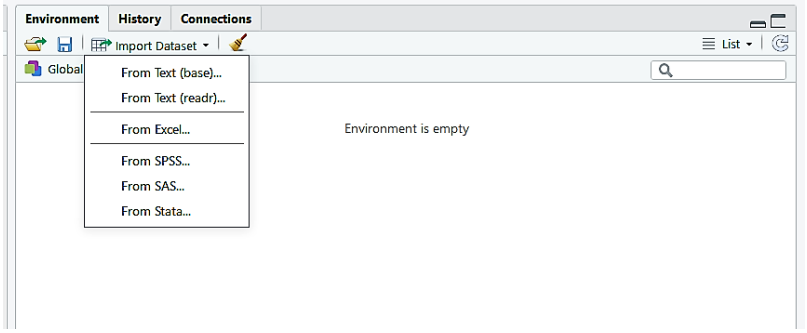
además verificamos que la paqueteria se ha activado

Importando datos

R soporta un gran cantidad de datos, tanto estructurados como no estructurados, puede leer bases guardadas del equipo y datos cargados desde un servidor de internet, a continuación motratemos la forma de llamar las bases de datos, para posteriormente trabajar con ellos.

```
#CSV
library(foreign)
datos1<-read.csv("ejemplo1.csv",header=TRUE)
datos11<-read.table("ejemplo1.csv",sep=",",header=TRUE)
#SPSS
datos2<-read.spss("ejrl.sav",to.data.frame=TRUE)
library(haven)
datos21<- read_sav("ejrl.sav")
#txt
datos3<-read.table("Basetarea.txt",sep="\t",header=TRUE)
#Excel
library(xlsx)
datos4<-read.xlsx("base.xlsx")
library(readxl)
datos41<- read_excel("base.xlsx")
datos42<-read_xlsx("base.xlsx")
#DBF
read.dbf("file")
#Stata
read.dta("file")
read_dta("file")
#Minitab
read.mtp("file")
#SAS
read.ssd("file")
#JSON
library("rjson")
fromJSON("data1.json")
```

Otra forma de cargar una base de datos es colocarse en el tercer cuadrante, y seleccionar la opcón *Import Dataset*



Aquí basta con elegir el tipo de archivo que tengamos:

- From Text (base)... (Aquí se subirá un archivo de extensión *.txt*)
- From Text (readr)... (Aquí se cargará un archivo de extensión *.csv*)
- From Excel... (Aquí se importará un archivo de extensión *.xlsx* o *.xls*)
- From SPSS... (Aquí se cargará un archivo de extensión *.sav*)
- From SAS... (Aquí se cargará un archivo de extensión *.sas7bdat* o *.sd7*)
- From Stata... (Aquí se cargará un archivo de extensión *.dta*)

Dos observaciones importantes: 1. Las opciones aquí mostradas dependerán del sistema operativo y la versión de *R* 2. Son limitadaslas opciones para cargar los datos, por lo cual se recomienda hacer uso de las funciones *read...*

Como mencionamos, una vez cargada la base de datos, la forma de oprar será igual al manejo de datos con matrices, pero la gran ventaja de trabajar con esta estructura de datos, es que cada variable puede ser de un tipo diferente, es decir, podemos tener en la base de datos una variable de tipo caracter como *Nombre* y otra variable de tipo numérica como *Calificación*, pero cada variable sí tiene que ser de un sólo tipo.

Estadística descriptiva

En la introducción mencionamos el gran potencial de *R* a nivel estadístico, evidentemente la parte descriptiva tiene varias funciones que nos permitirán analizar su comportamiento, de forma numérica y gráfica.

Numéricamente

```
#Ejemplo. Consideremos la siguiente muestra que representa las
#calificaciones de alumnos en el primer parcial de Estadística II
x=c(0,1,2,2,3,3,3,4,4,4,4,4,5,5,5,5,5,5,5,5,5,5,6,6,6,6,6,6,6,6,6,6,6,7,7,7,7,7,7,7,
    8,8,8,8,9,9,10)
#con estos datos haremos un análisis descriptivo

################################
# Medidas de tendencia central #
################################

#media, mediana
mean(x); median(x); quantile(x)[3]
```

```
## [1] 5.48
```

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

```
## 50%
##   6
```

Podemos ver que estas dos medidas son muy similares, la media es un promedio de todos los valores presentes en muestra, al considerar todos los valores esta medida se vuelve muy sencible a datos extremos, además el resultado de operación no necesariamente tomará algún valor en muestra; en cambio la mediana siempre tomará algún valor en muestra y no es sencible a valores extremos, la medida es estable.

```
#Moda
#Esta medida se puede obtener de dos maneras, la primera sería manualmente, es decir
#extraer la observación con más repeticiones
moda=c(table(x))
moda[which(moda == max(moda))[1]]
```

```
##  6
## 12
```

Este método es eficiente si y sólo si hay un valor único, *R* ya tiene una función que calcula la moda para valores discretos, esta se encuentra en la paquetería *modeest*:

```
#cargamos la paqueteria
library(modeest)
##############################################################################
#si marca error por la falta de genefilter, debe de instalar otro paquete
#install.packages("BiocManager")
#library("BiocManager")
#BiocManager::install("genefilter")
##############################################################################
#Aplicamos la función
mfv(x)
```

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

Esta función es eficiente y devuelve múltiples modas en caso de que exista más de una.

```
#Percentiles
quantile(x,0.1); quantile(x,0.75); quantile(x,0.99)
```

```
## 10%
##   3
```

```
## 75%
##   7
```

```
##  99%
## 9.51
```

R genera de forma automática un resumen descriptivo con la función *summary*

```
#Resumen: mínimo, máximo, cuartiles y media
summary(x)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    4.25    6.00    5.48    7.00   10.00
```

```
###################################
#     Medidas de dispersion     #
###################################
#Varianza y Desviacion estandar
var(x); s=sd(x)
```

```
## [1] 4.050612
```

```r
#Rango
range(x); R=max(x)-min(x); R
```

```
## [1]  0 10
```

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

```r
#rango intercuartilico
RIC<-quantile(x,0.75)-quantile(x,0.25); RIC
```

```
##  75%
## 2.75
```

```r
#coeficiente de variación
cv=s/mean(x)*100; cv
```

```
## [1] 36.72652
```

```r
############################
#     Medidas de forma     #
############################
#coeficiente de asimetria
ca=sum((x-mean(x))^3)/(length(x)*sd(x)^3); ca
```

```
## [1] -0.3320303
```

```r
#con R necesitamos una paquetería extra
library(moments)
```

```
##
## Attaching package: 'moments'
```

```
## The following object is masked from 'package:modeest':
##
##     skewness
```

```r
skewness(x)
```

```
## [1] -0.3422462
```

Como el coeficiente de asimetria es negativo, se dice el sesgo de la distribución esta a la izquierda de la mediana y hay una "más" acumulación a la derecha de la mediana.

```r
#coeficiente de curtosis
curtosis=(sum((x-mean(x))^4)/(length(x)*sd(x)^4)); curtosis
```

```
## [1] 3.248305
```

```r
#con R necesitamos la paquetería moments
kurtosis(x)
```

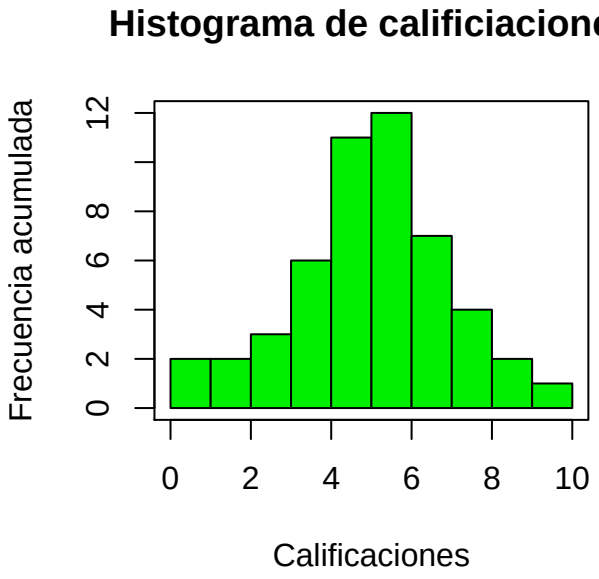
```
## [1] 3.382241
```

Como el coeficiente de curtosis es mayor a cero, podemos decir que nuestros datos tienen una distribución leptocúrtica, es decir los datos están untanto concentrados en la media, siendo una curva apuntada.

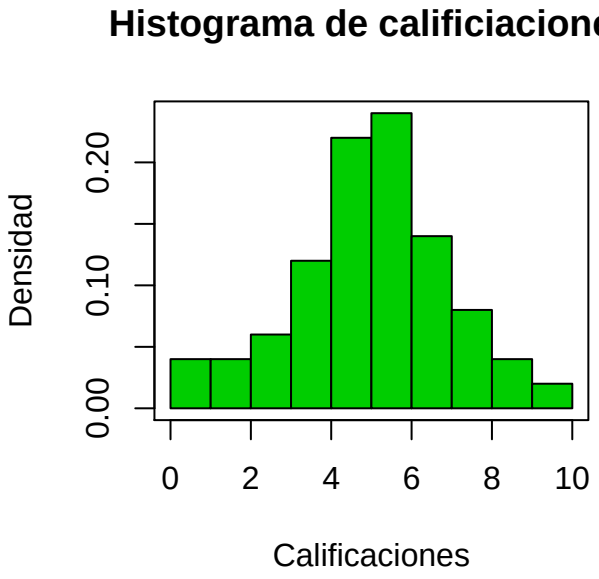
Gráficamente

```r
#histograma de frecuencia acumulada
hist(x,breaks=10,col="green2",main="Histograma de calificiaciones",xlab="Calificaciones",
```

```
     ylab="Frecuencia acumulada");
box()
```

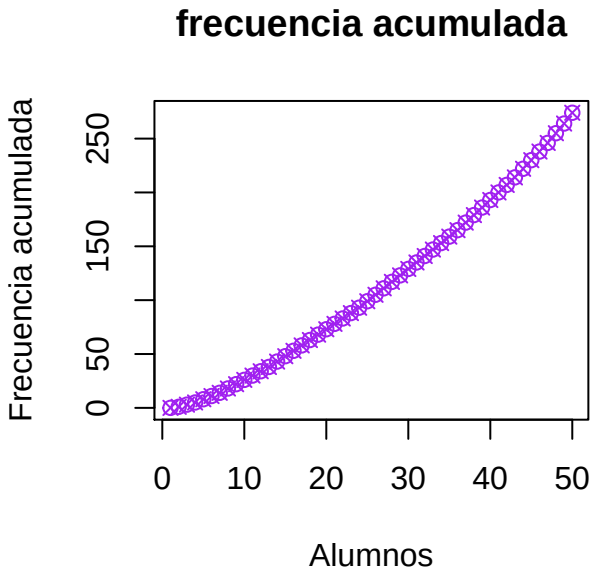



```
#histograma de frecuencia relativa
hist(x,freq=F,breaks=10,col="green3",main="Histograma de calificiaciones",
     xlab="Calificaciones", ylab="Densidad");
box()
```

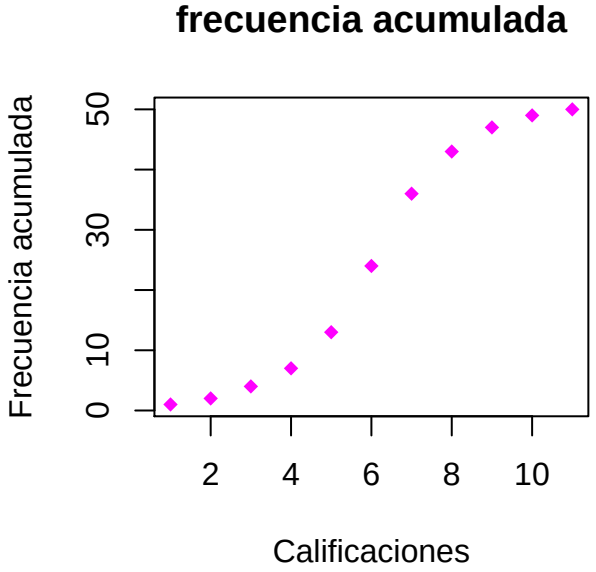



```
#frecuencia acumulada
fa=cumsum(x);
plot(fa,main="frecuencia acumulada",xlab="Alumnos",ylab="Frecuencia acumulada",
```

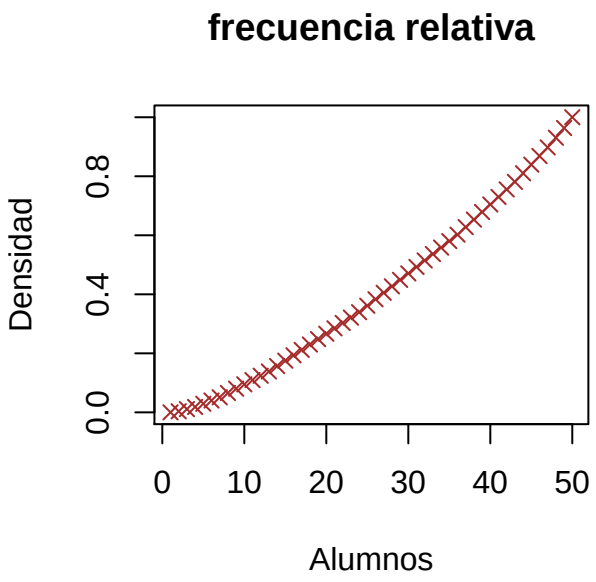
```
     col="purple",pch=13)
```



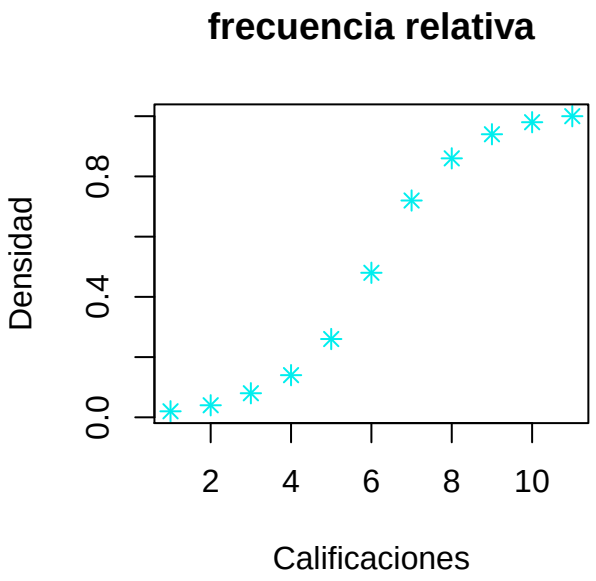

```
fa1=cumsum(table(x));
plot(fa1, main="frecuencia acumulada",xlab="Calificaciones",ylab="Frecuencia acumulada",
     col="magenta",pch=18)
```



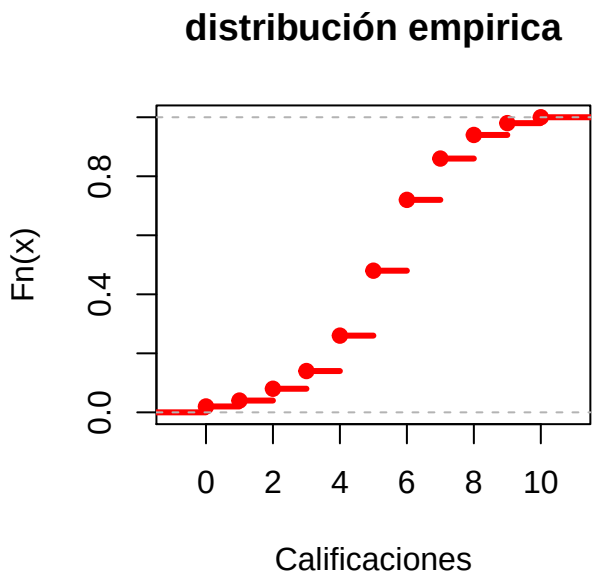

```
#frecuencia relativa
fr=cumsum(x)/sum(x);
plot(fr,main="frecuencia relativa",xlab="Alumnos",ylab="Densidad",col="brown",pch=4)
```



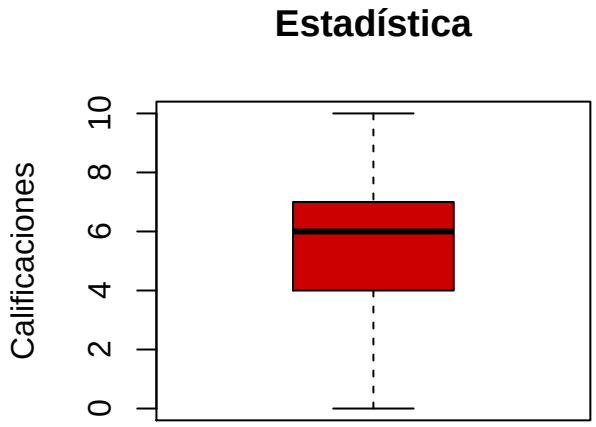

```
#frecuencia relativa
fr1=cumsum(moda)/sum(moda);
plot(fr1, main="frecuencia relativa",xlab="Calificaciones",ylab="Densidad",col="cyan2",
     pch=8)
```




```
#grafica de distribucion empirica
ECDF1=ecdf(x);
plot(ECDF1, col="red",lwd=3,xldo="datos",yldo=" ",
     main="distribución empirica", xlab="Calificaciones")
```




```
#diagrama de caja
boxplot(x,col="red3",main="Estadística",ylab="Calificaciones")
```

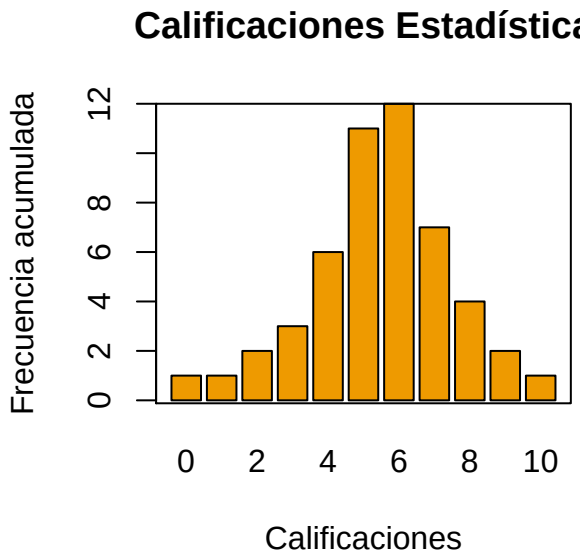



```
#tallo de hoja
stem(x)
```

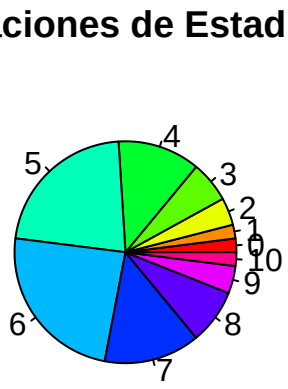
```
##
##   The decimal point is at the |
##
##    0 | 00
##    2 | 00000
```

```
##     4 | 00000000000000000
##     6 | 000000000000000000
##     8 | 000000
##    10 | 0
```

```
#gráfico de barra
barplot(moda,col="orange2",main="Calificaciones Estadística",
        ylab="Frecuencia acumulada",xlab="Calificaciones");
box()
```




```
#diagrama de pie
pie(moda,main="Calificaciones de Estadistica",col=rainbow(11))
```



```r
#gráfico de dispersión
plot(moda,col="pink3",pch=20, cex=2,main="Calificaciones de Estadistica",
     ylab="Frecuencia acumulada",xlab ="Calificaciones",ylim=c(0,13),
     xlim=c(0,13),bcolor="red")
```