

PROGRAMA DE ATRACCIÓN DE TALENTO TECNOLÓGICO PARA EL SECTOR TURÍSTICO ANDALUZ

Visualización de datos con R

20
20

Visualización de datos con R.

F
T
T

Instrucciones previas.

```
#Carga de datos básica con los Datasets disponibles en R  
data(airquality)  
#Establecer el Working Directory.  
setwd("/home/adastra/Escritorio/CursoR/IntroBasicVisualization/")  
#Carga del fichero en un dataset.  
airqualitydataset = read.csv('airquality.csv',header=TRUE, sep=",")
```

Instrucciones previas.

#Functions para explorar los datos cargados.

```
str(airqualitydataset)
```

```
head(airqualitydataset, n=3)
```

```
summary(airqualitydataset)
```

Visualización de datos.

#Funciones de visualización.

#Plot con una de las columnas del dataset.

```
plot(airqualitydataset$Ozone)
```

#Plot con la relación de 2 columnas x, y.

```
plot(airquality$Ozone, airquality$Wind)
```

#Plot con el dataset completo. Se obtiene una matriz de correlación con todas las columnas.

```
plot(airqualitydataset)
```

#Plot con una columna y tipo "b". Los tipos más comunes son:

#p: points, l: lines, b: both, h: high density. Ver la documentación ?plot

```
plot(airqualitydataset$Ozone, type= "b")
```

#Plot con etiquetas en eje x e y.

```
plot(airqualitydataset$Ozone,  
     xlab = 'ozone Concentration',  
     ylab = 'No of Instances',  
     main = 'Ozone levels in NY city',  
     col = 'green')
```

Visualización de datos.

```
#Generación de BarPlots: Representación de datos en forma de barras rectangulares.  
#La longitud de las barras es proporcional al valor de la variable o columna en el dataset.  
# Horizontal bar-plot  
barplot(airqualitydataset$Ozone,  
        main = 'Ozone Concentration in air',  
        xlab = 'ozone levels',  
        col= 'green',  
        horiz = TRUE)  
  
barplot(airqualitydataset$Ozone,  
        main = 'Ozone Concentration in air',  
        xlab = 'ozone levels',  
        col= 'green',  
        horiz = F)  
  
#Generación de histogramas. Los grupos de barras se agrupan en rangos contiguos.  
hist(airqualitydataset$Solar.R)  
  
#El histograma también puede tener un color.  
hist(airqualitydataset$Solar.R,  
     main = 'Solar Radiation values in air',  
     xlab = 'Solar rad.',  
     col='red')  
  
#Boxplot: Gráfico que permite enseñar estadísticas descriptivas para cada una de las variables del dataset.  
#Representa la ubicación de los valores medios del dataset.  
boxplot(airqualitydataset$Solar.R)  
boxplot(airqualitydataset)
```

Visualización de datos.

#Malla de gráficos. Es posible ubicar múltiples plots en uno solo para posteriormente visualizarlos en la misma ventana.

#Se utiliza la función "par" y a continuación se crean objetos plot.

#Para que funcione correctamente se deben incluir como mínimo los siguientes parámetros:

#mar: Margen de la malla.

#mfrow: número de filas y columnas.

#bty: Incluir un borde o no.

#las: Posición de las etiquetas. 1 para horizontal y 0 para vertical.

```
par(mfrow=c(3,3), mar=c(2,5,2,1), las=0, bty="n")
```

```
plot(airquality$Ozone)
```

```
plot(airquality$Ozone, airquality$Wind)
```

```
plot(airquality$Ozone, type= "c")
```

```
plot(airquality$Ozone, type= "s")
```

```
plot(airquality$Ozone, type= "h")
```

```
barplot(airquality$Ozone, main = 'Ozone Concentration in air', xlab = 'ozone levels', col='green', horiz = TRUE)
```

```
hist(airquality$Solar.R)
```

```
boxplot(airquality$Solar.R)
```

```
boxplot(airquality[,0:4], main='Multiple Box plots')
```


Librerías para Visualización de datos.

Existen algunas librerías que no están disponibles en el core de R y que incluyen elementos que extienden las funciones gráficas del lenguaje.

- ***lattice*** es un paquete con gráficos para datos multivariados.
- ***GGPlot2*** es una de las librerías más utilizadas para visualización de datos y permite crear gráficos del tipo "Grammar of Graphics" o "GG". Se trata de un esquema general para la visualización de datos que permite indicar componentes semánticos tales como escalas y capas.
- ***Plotly*** dispone de funciones que permiten generar plots interactivos utilizando la librería en JavaScript *plotly.js*. Una de las ventajas de este enfoque es que se puede exportar en formato web y conservar sus características interactivas.
- ***maps*** Librería core para enseñar mapas geográficos. Es necesario instalar paquetes extra para cargar mapas de otros países y tipos: Más información:

<https://geoinquietosmadrid.github.io/datavis-with-r/secciones/maps/index.html>

Librería: LATTICE

```
install.packages("lattice")  
library(lattice) #Loading the dataset  
attach(mtcars)  
head(mtcars)  
gear_factor<-factor(gear,levels=c(3,4,5),  
                    labels=c("3gears","4gears","5gears"))  
  
cyl_factor <-factor(cyl,levels=c(4,6,8),  
                  labels=c("4cyl","6cyl","8cyl"))  
  
densityplot(~mpg, main="Density Plot", xlab="Miles per Gallon")  
  
splom(mtcars[c(1,3,4,5,6)], main="MTCARS Data")  
  
xyplot(mpg~wt|cyl_factor*gear_factor,  
       main="Scatterplots : Cylinders and Gears",  
       ylab="Miles/Gallon", xlab="Weight of Car")
```

Librería: GGPlot2

```
install.packages("ggplot2")
library(ggplot2)

#Loading the dataset
attach(mtcars)
# create factors with value labels

mtcars$gear <- factor(mtcars$gear,levels=c(3,4,5),
                     labels=c("3gears", "4gears", "5gears"))
mtcars$am <- factor(mtcars$am,levels=c(0,1),
                   labels=c("Automatic","Manual"))
mtcars$cyl <- factor(mtcars$cyl,levels=c(4,6,8),
                    labels=c("4cyl","6cyl","8cyl"))

#Plot con el dataset completo, indicando cuáles serán los puntos X y Y además de agregando las figuras "punto".
ggplot(data = mtcars, mapping = aes(x = wt, y = mpg)) + geom_point()

#Plot para generar una representación gráfica de los factores con colores diferenciados.
ggplot(data = mtcars, mapping = aes(x = mtcars, y = mtcars, color = as.factor(mtcars) )) + geom_point()

#Plot que representa puntos en función al tamaño de los atributos.
ggplot(data = mtcars, mapping = aes(x = wt, y = mpg, size = qsec)) + geom_point()

#Plot que permite asignar formas y color a cada factor.
ggplot(mtcars,aes(mpg, wt, shape = factor(cyl))) +
  geom_point(aes(colour = factor(cyl)), size = 4) +
  geom_point(colour = "grey90", size = 1.5)
```

Librería: Plotly

```
install.packages("plotly")
library(plotly)

#Plot básico con plotly.
plot_ly(data = mtcars, x = hp, y = wt)

#Plot con un tamaño fijo de las primeras 10 observaciones y colores establecidos.
plot_ly(data = mtcars, x = hp, y = wt,
        marker = list(size = 10, color = 'rgba(255, 182, 193, .9)',
                      line = list(color = 'rgba(152, 0, 0, .8)',
                                width = 2)))

#Plot con 3 datasets aleatorios en Y y un vector de 1 a 100 en X

data1 <- rnorm(100, mean = 10)
data2 <- rnorm(100, mean = 0)
data3 <- rnorm(100, mean = -10)
x <- c(1:100)
data <- data.frame(x, data1, data2, data3)
plot_ly(data, x = ~x)%>%
  add_trace(y = ~data1, name = 'data1', mode = 'lines')%>%
  add_trace(y = ~data2, name = 'data2', mode = 'lines+markers')%>%
  add_trace(y = ~data3, name = 'data3', mode = 'markers')
```

Librería: maps

```
#Gráficos para mapas.
install.packages("maps", dependencies=TRUE)
install.packages("rgdal")
library(sp)
library(maps)
library(mapdata)
#library(rgdal)
library(mapttools)
library(RColorBrewer)
# library(knitr)
# library(rmarkdown)

map(database="state")
data <- read.csv('ABC_locations.csv', sep=",")
symbols(data$Longitude, data$Latitude, squares =rep(1, length(data$Longitude)), inches=0.03, add=TRUE)

map('worldHires')
map('worldHires','Spain')
points(c(-3.682746,2.1734066),c(40.4893538,41.4850595),pch=19, col="red",cex=0.5)
text(x=c(-3.682746,2.1734066),y=c(40.4893538,41.4850595),labels = c("Madrid", "Barcelona"), pos = 3)
#Más información https://geoinquietosmadrid.github.io/datavis-with-r/secciones/maps/index.html
```