

Fundamentos de la programación estadística y Data Mining en R

Unidad 1. Visualización básica de datos en R

Dr. Germán Rosati (Digital House - UNTREF - UNSAM)

03 septiembre, 2017

Visualizaciones básicas de datos en R

- Existen muchos comandos para hacer visualizaciones en R. Vamos avanzar sobre el paquete de gráficos “base” del lenguaje. Hay otras herramientas muy poderosas para generar gráficos en R, como por ejemplo el paquete ggplot2 (<http://ggplot2.org/>)
- La función básica para generar gráficos en R es plot.

```
> str(plot)
## function (x, y, ...)
```

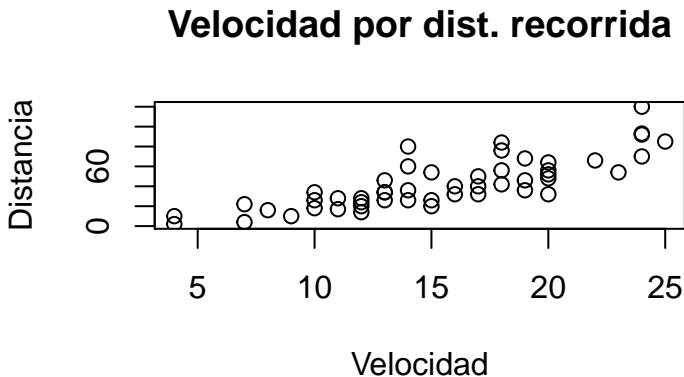
- Dos argumentos importantes:
 - ▶ x: vector que va en el eje x
 - ▶ y: vector que va en el eje y

Visualizaciones básicas de datos en R

- Luego, hay otros parámetros opcionales para setear el gráfico:
 - ▶ `type`: el tipo de gráfico
 - ▶ `main`: título para el gráfico
 - ▶ `xlab`, `yylab`: títulos para eje X e Y
 - ▶ `*x`: la variable a plotear
 - ▶ `*breaks`: cantidad de intervalos a plotear
 - ▶ ... muchos otros

Visualizaciones básicas de datos en R

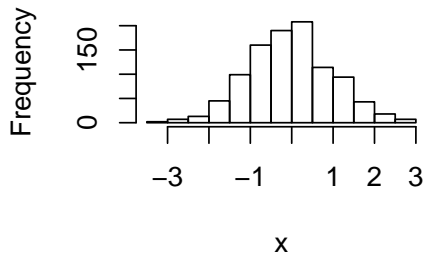
```
plot(cars$speed,cars$dist  
  ,main="Velocidad por dist. recorrida"  
  ,xlab="Velocidad",ylab="Distancia")
```



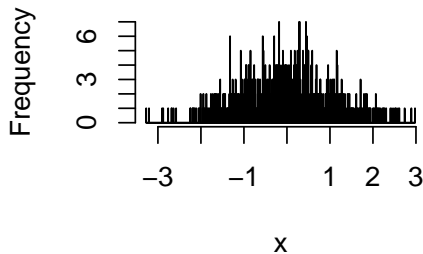
Visualizaciones básicas de datos en R: hist

```
x<-rnorm(1000,0,1)
par(mfrow=c(1,2))
hist(x)
hist(x,breaks=1000)
```

Histogram of x



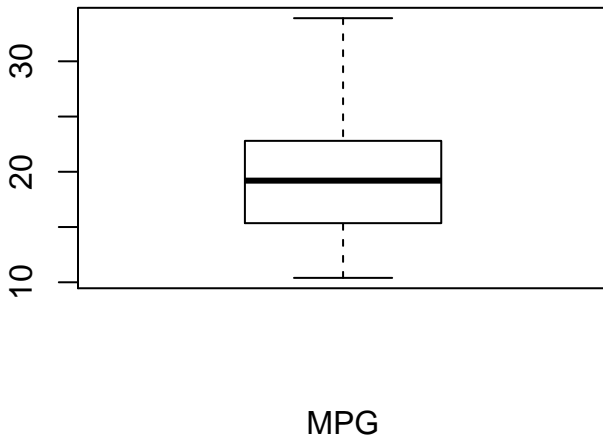
Histogram of x



Visualizaciones básicas de datos en R: boxplot

```
boxplot(mtcars$mpg, main='Car Mileage Data',xlab='MPG')
```

Car Mileage Data



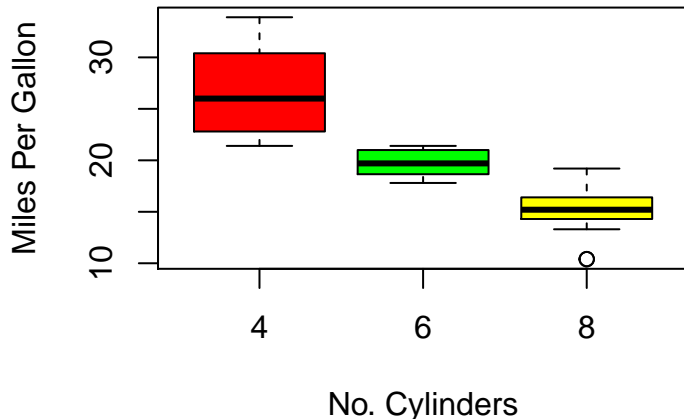
Visualizaciones básicas de datos en R: boxplot

- Se puede generar un boxplot de mpg para cada categoría de la variable (cualitativa) cyl y variar los colores.
- Ver mpg~cyl y col=c("red","green","yellow")

```
boxplot(mpg~cyl,data=mtcars, main="Car Milage Data"  
  , xlab="No. Cylinders", ylab="Miles Per Gallon"  
  , col=c("red","green","yellow"))
```

Visualizaciones básicas de datos en R: boxplot

Car Milage Data



Visualizaciones básicas de datos en R: boxplot

- `x`: la variable a plotear
- `formula`: si se desea hacer un boxplot para cada categoría de otra variable
- `data`: el dataframe a plotear
- ... muchos otros

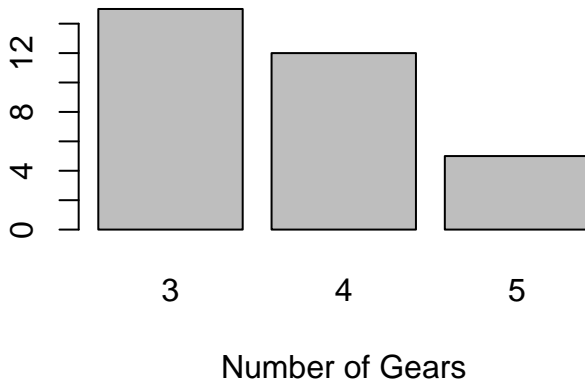
Visualizaciones básicas de datos en R: barplot

- Para generar gráficos de barras, puede usarse la función `barplot`.
- La misma requiere como input una tabla de frecuencias de valores (a diferencia del caso de histogramas que toma el vector de distribución de la función).

```
counts <- table(mtcars$gear)
barplot(counts, main="Car Distribution",
        xlab="Number of Gears")
```

Visualizaciones básicas de datos en R: barplot

Car Distribution

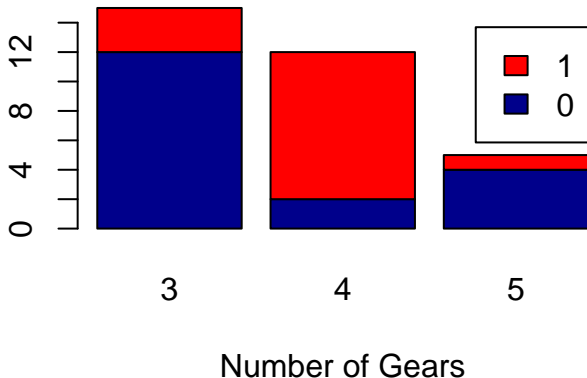


Visualizaciones básicas de datos en R: barplot

- También se pueden realizar barplots apilados. Para ello, generamos una tabla de contingencia con dos variables (`mtcars$vs` cruzada por `mtcars$gear`) y la usamos como input al gráfico.
- Luego generamos
 - ▶ dos colores diferentes `col=c("darkblue","red")`
 - ▶ una legenda con los nombres de las categorías `legend = rownames(counts)`

```
counts <- table(mtcars$vs, mtcars$gear)
barplot(counts, main="Car Distribution by Gears and VS",
        xlab="Number of Gears", col=c("darkblue","red"),
        legend = rownames(counts))
```

Car Distribution by Gears and VS



Visualizaciones básicas de datos en R: barplot

```
str(barplot)
```

```
## function (height, ...)
```

- height: vector o matriz
 - ▶ Si es vector, determina la altura de la matriz
 - ▶ Si es una matriz y besides=FALSE cada barra del plot corresponde a una columna con los valores “apilados”
 - ▶ Si es una matriz y besides=TRUE cada barra del plot corresponde a una columna con los valores “yuxtapuestos”.

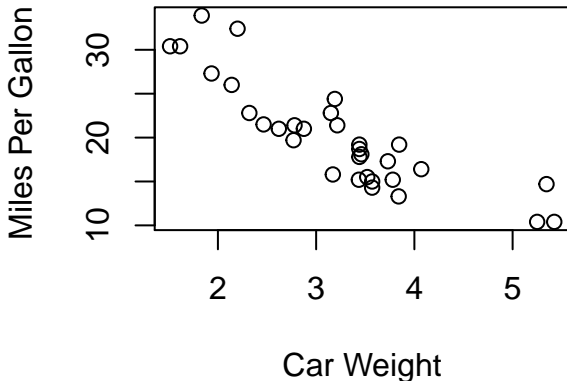
Visualizaciones básicas de datos en R: scatterplots

- Muchas formas de hacer un scatter plot en R.
- La más simple: con `plot()`
 - ▶ x: variable en eje X
 - ▶ y: variable en eje Y

Visualizaciones básicas de datos en R: scatterplots

```
plot(mtcars$wt, mtcars$mpg, main="Ejemplo",  
      xlab="Car Weight", ylab="Miles Per Gallon ")
```

Ejemplo



Visualizaciones básicas de datos en R: scatterplots

- Finalmente, podemos agregar líneas de tendencia al scatter.
- Generamos dos:
 - ▶ una basada en regresión lineal: `abline(lm(mtcars$mpg~mtcars$wt), col="red")`
 - ▶ otra basada en una regresión no lineal
`lines(lowess(mtcars$wt,mtcars$mpg), col="blue")`

```
plot(mtcars$wt, mtcars$mpg, main="Ejemplo",  
     xlab="Car Weight ", ylab="Miles Per Gallon ")  
abline(lm(mtcars$mpg~mtcars$wt), col="red")  
lines(lowess(mtcars$wt,mtcars$mpg), col="blue")
```

Scatterplot Example

