Lab02 - Descriptive Analysis

Descriptive analysis

▼ Introducción de una base de datos simple

Crear una base de datos (interna a RStudio)

```
DATA <- datasets::mtcars #load the data frame
DATA <- rbind(DATA,rep(NA,ncol(DATA))) #adding a couple of er
```

Instalar los packages necesarios

```
install.packages("dplyr")
install.packages("knitr")
```

Se verá algo así

```
DATA
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	
1											
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0		
3	2										
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0		

```
names(DATA)
#"mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"
```

Ayuda para entender la base de datos

```
?datasets::mtcars
```

▼ Factors, creación y cambios en variables

Variable “am” with 1 for automatic and 0 for manual

```
DATA$SHIFT <- factor(DATA$am, levels = c(0:1), labels = c("Ma
```

Ver valores

```
DATA$SHIFT
```

Ver categorías

```
levels(DATA$SHIFT)
```

Ver si se correlaciona bien

```
DATA[,c("am","SHIFT")]
```

▼ Características de las bbdd (mean, median, frequencies...)

- Summary

Tenemos que fijarnos en que trabajamos con valores factors para evitar errores

```
summary(DATA$mpg)
#   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
#  10.40   15.43   19.20   20.09   22.80   33.90       1
```

Si lo hacemos con la variable am que creamos antes (solo 0 y 1) obtendremos esto innecesario

```
summary(DATA$am)
#   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
# 0.0000  0.0000  0.0000  0.4062  1.0000  1.0000       1
```

Mejor deberíamos usar algo así

```
summary(DATA$SHIFT)
# Manual Automatic NA's
# 19        13       2
```

- Frequency (veces que se repite)

```
table(DATA$SHIFT)
#    Manual Automatic
#      19        13
```

```
table(DATA$SHIFT, useNA = "ifany")
#    Manual Automatic      <NA>
#      19        13         1
```

- Frequency (con %)

```
## Sin NA ya que esos no sirven para hacer un buen cómputo
prop.table(table(DATA$SHIFT))
#   Manual Automatic
#  0.59375   0.40625
```

Si lo queremos en porcentaje deberemos multiplicarlo por 100

```
prop.table(table(DATA$SHIFT))*100
#   Manual Automatic
```

```
#   59.375    40.625
```

▼ Variables cuantitativas(Ordenar, normalidad, rango Sd, cuartiles...)

→Normality

If n>3 and n<5000 → Shapiro-Wilks
else Lilliefor's correction for Kolmogorov-Smirnoff

```
install.packages("nortest")
```

Para ver n

```
length(DATA$mpg)
```

Aquí valoramos n, vemos que está en el rango de Shapiro-Wilks

```
shapiro.test(DATA$mpg)
#Shapiro-Wilk normality test
#data:  DATA$mpg
#W = 0.94756, p-value = 0.1229
```

En el caso de Lilliefor's (aquí no)

```
nortest::lillie.test(DATA$mpg)
#Lilliefors (Kolmogorov-Smirnov) normality test
#data:  DATA$mpg
#D = 0.1263, p-value = 0.2171
```

p-value under 0.05 means that the variable is not normal

→ Skewness and kurtosis

Necesitamos instalar OTRO paquete

```
install.packages("moments")
```

```
moments::skewness(DATA$mpg, na.rm = TRUE)
# 0.6404399
```

```
moments::kurtosis(DATA$mpg, na.rm = TRUE)
# 2.799467
```

Para interpretarlo podemos tomar estos valores del Kurtosis

if >0 means that is “leptokurtic” (higher than wider) #en pico
if =0 means that is “mesokurtic” (gauss-like)
if <0 means that is “platykurtic” (wider than higher) #aplanada

→ Central tendency

```
mean(DATA$mpg, na.rm=TRUE)
#Media sin valores NA, si estuvieran daría error
#20.09062
median(DATA$mpg, na.rm = TRUE)
#19.2
```

→Mode

```
table(DATA$mpg)
```

Ordenar la tabla

```
sort(table(DATA$mpg))
#13.3 14.3 14.7   15 15.5 15.8 16.4 17.3 17.8 18.1 18.7 19.7
#1    1    1    1    1    1    1    1    1    1    1    1
```

Para coger el que más se repite

```
sort(table(DATA$mpg), decreasing = T)
#10.4 15.2 19.2   21 21.4 22.8 30.4 13.3 14.3 14.7   15 15.5
#2    2    2      2    2    2    2    1    1    1    1    1
```

Para coger el mayor concretamente de uno

```
max(sort(table(DATA$mpg), decreasing = T))
#2
```

Para ver de dónde a dónde van los valores

```
range(DATA$mpg, na.rm = TRUE)
#10.4 33.9
```

Para ver cuál es el autentico rango (diferencia de extremos)

```
diff(range(DATA$mpg, na.rm = TRUE))
#23.5
```

→Standard desviation

```
sd(DATA$mpg, na.rm = TRUE)
#6.026948
```

→Variance

```
var(DATA$mpg, na.rm = TRUE) // (sd(DATA$mpg, na.rm = TRUE))^2
#36.3241
```

→Quantiles

```
quantile(DATA$mpg, na.rm = TRUE)
#    0%    25%    50%    75%   100%
#10.400 15.425 19.200 22.800 33.900
```

Si queremos un cuantil exacto (o varios concretos)

```
quantile(DATA$mpg, na.rm = TRUE, probs = 0.9)
#  90%
#30.09
```

```
quantile(DATA$mpg, na.rm = TRUE, probs = c(.25,.5,.75))
#   25%    50%    75%
#15.425 19.200 22.800
```

O una cadena (con los deciles)

```
quantile(DATA$mpg, na.rm = TRUE, probs = seq(0,1,1/10))
#   0%   10%   20%   30%   40%   50%   60%   70%   80%   90%
#10.40 14.34 15.20 15.98 17.92 19.20 21.00 21.47 24.08 30.09
```

Interquartil

```
IQR(DATA$mpg, na.rm=TRUE) // diff(quantile(DATA$mpg, na.rm =
#7.375         //     75%
#                  7.375
```

Coeficiente de desviación

```
data_values <- DATA$mpg

cv <- sd(data_values, na.rm = TRUE) / mean(data_values, na.r
cv
#29.99881
```

▼ Gráficos

Para usar esto teneis que abrir Rstudio, ir a Tools → Install Packages → Escribir ggplot2 en el único sitio vacío y esperar que se instale

Metemos los datos en otra variable y cargamos la librería instalada

```
gg.data <- datasets::mtcars
library(ggplot2)
```

Creamos un gráfico vacío

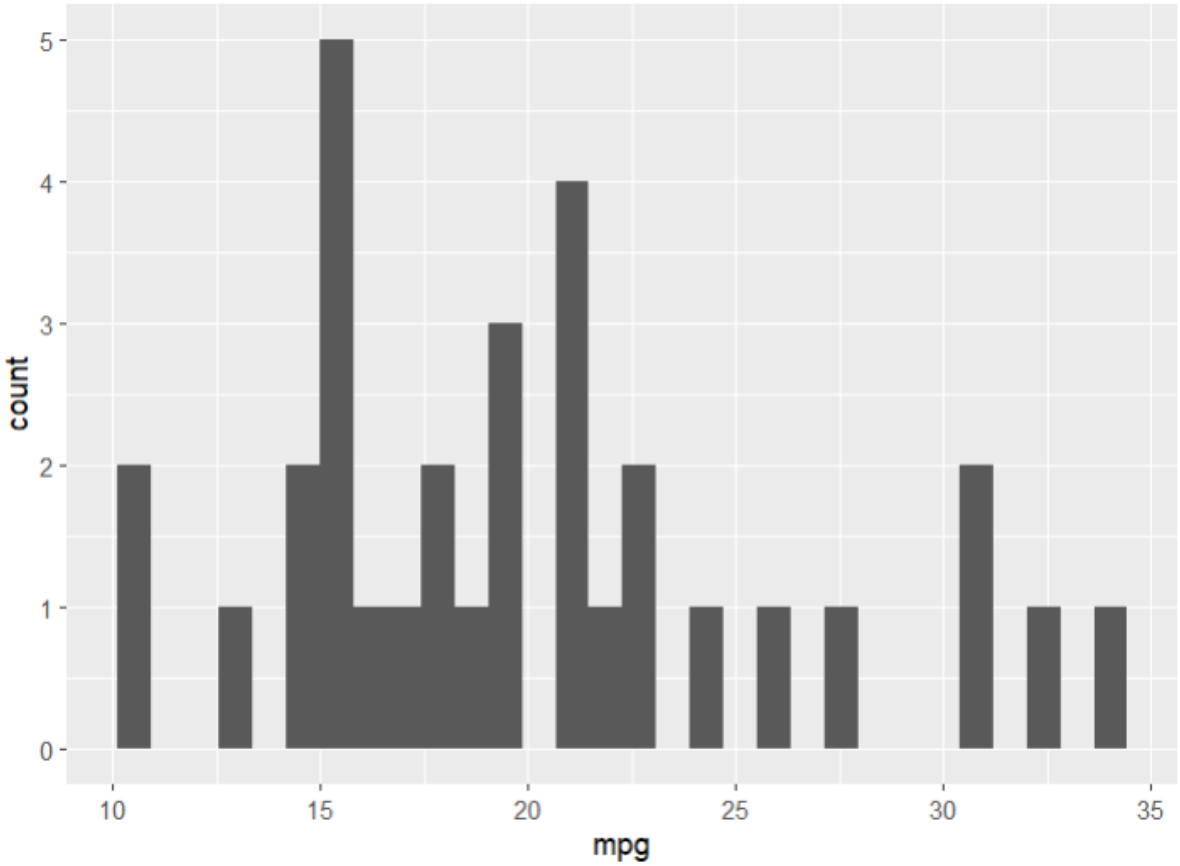
```
g <- ggplot(gg.data, aes(x=mpg))
g
```

Creamos una boxplot(dos barras solas)

```
g + geom_boxplot()
```

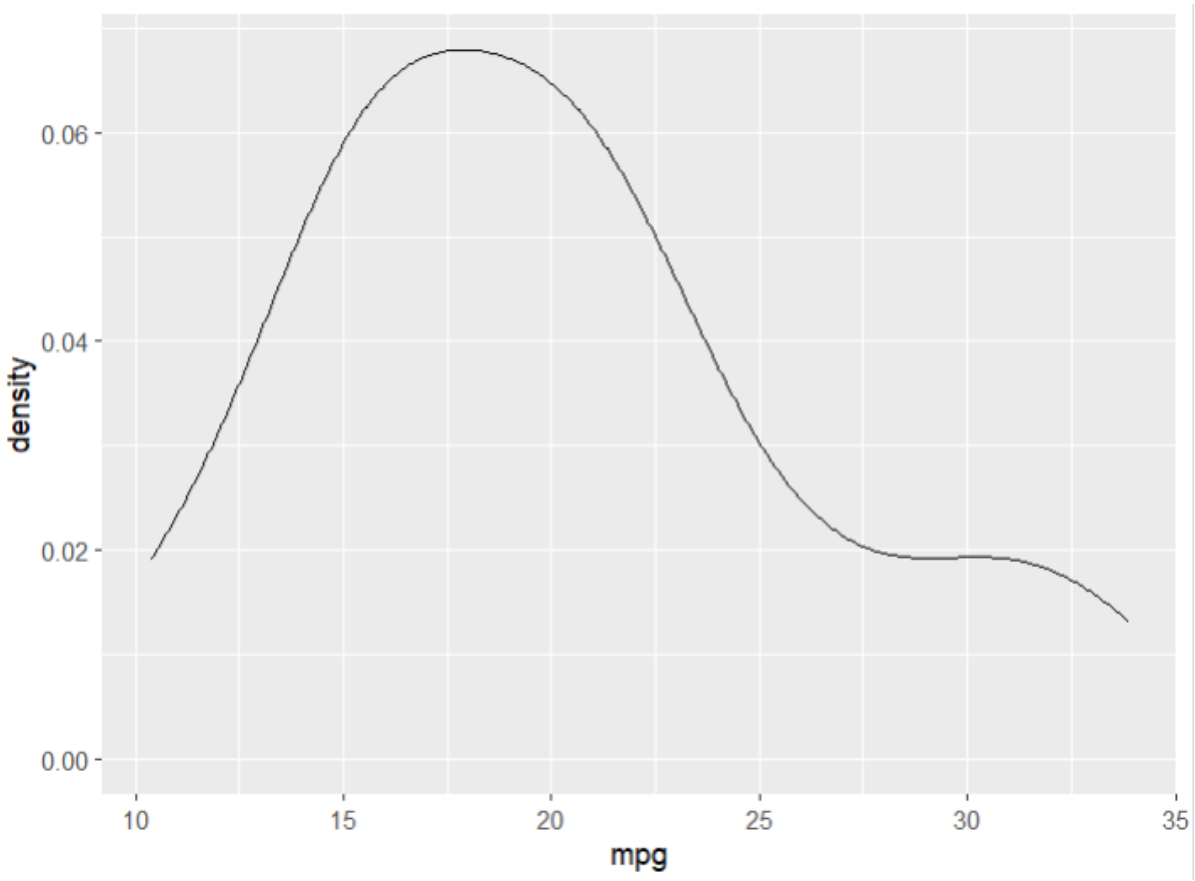
Creamos un histograma(mostrar las barritas con los valores de la BBDD y las veces repetidas)

```
g + geom_histogram()
#Para hacer las barras más gordas usamos geom_histogram(binw
```



Función de densidad

```
g + geom_density()
```

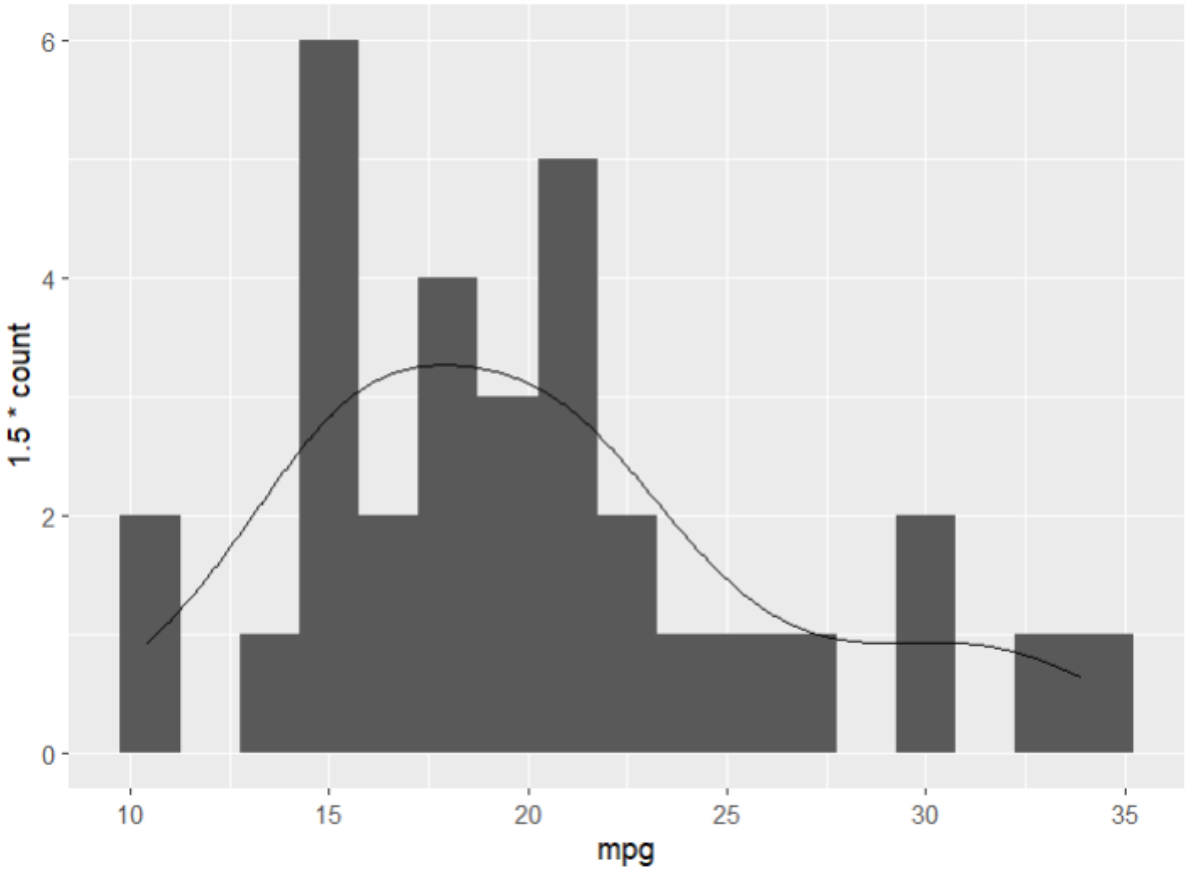


Asignando valores a los ejes X e Y

```
ggplot(gg.data, aes(x=mpg)) +
geom_density(aes(y=..count..))
```

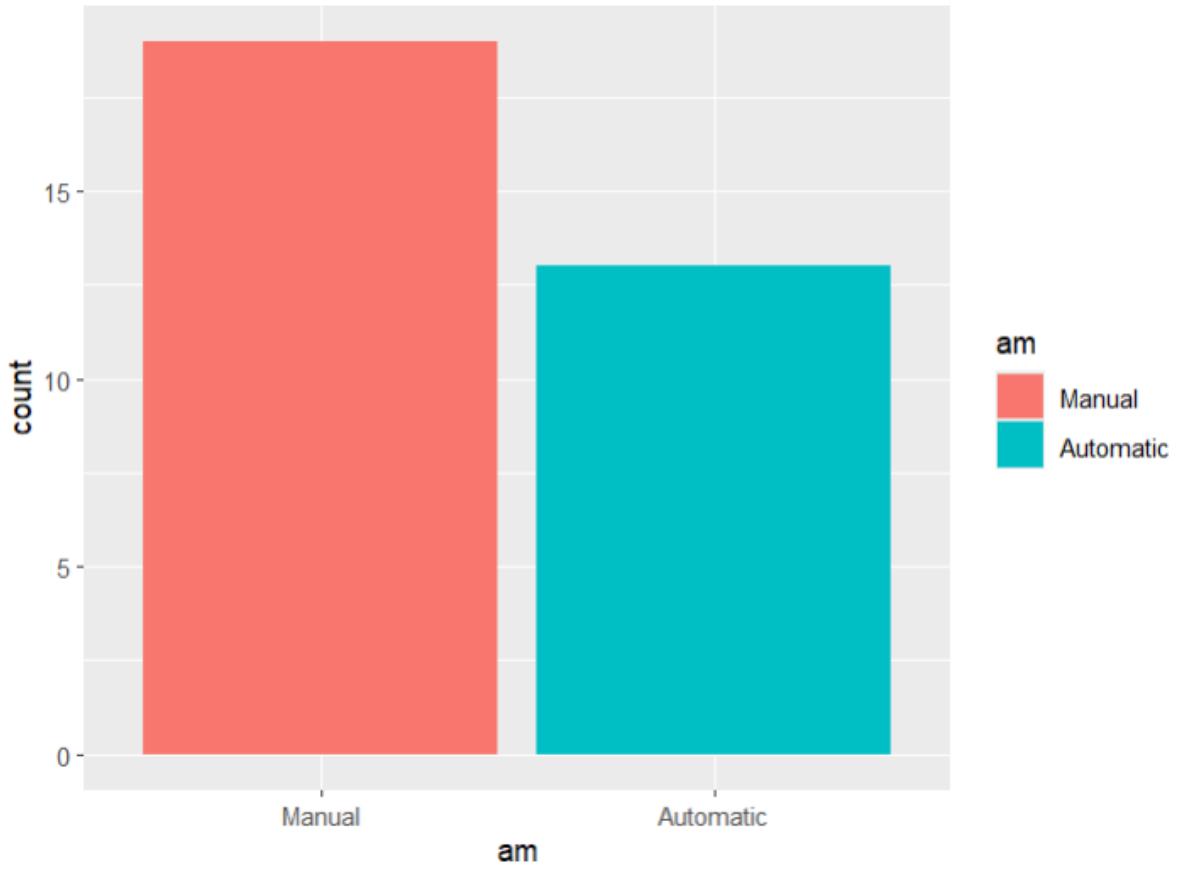
Tambien podemos unir gráficos en Rstudio

```
ggplot(gg.data, aes(x=mpg)) +
geom_histogram(binwidth = 1.5) +
geom_density(aes(y=1.5*..count..))
```



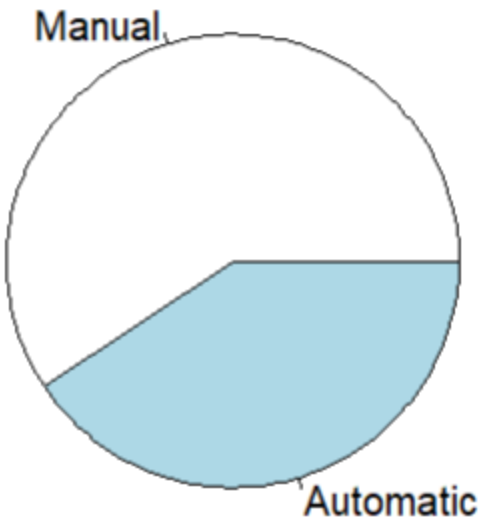
→Barplots

```
gg.data$am <- factor(gg.data$am, levels = 0:1, labels=c("Manu
"Automatic"))
ggplot(gg.data, aes(x=am, fill=am)) +
geom_bar()
```



El siguiente NO se recomienda usarlo

```
pie(prop.table(table(gg.data$am)))
```



EJERCICIOS 2