

# Jorge de Andrés González - Módulo 1

Jorge de Andrés

2/3/2020

## EXAMEN PRÁCTICO MÓDULO 1

Puesto que se trata de una prueba de Introducción a R:

- Se deben emplear las funciones básicas de R para tratar los datos. Así pues, no está permitido usar otros paquetes, como: `dplyr`, `data.table`, etc.
- Para generar los gráficos, se deben emplear exclusivamente las funciones propias del paquete `graphics`. Por lo tanto, no está permitido utilizar otros paquetes, como: `ggplot2`, `lattice`, etc.

---

## Ejercicios

### Vectores

1. Cree el vector A utilizando la función `seq`, de tal modo que se obtenga el siguiente resultado: 10, 8, 6, 4, 2, 0, -2, -4, -6. ¿Qué longitud tiene este vector? ¿Cuál es su suma? ¿Cuál es su valor medio? ¿Cuál es su error típico? El error típico se obtiene dividiendo la desviación típica entre la raíz cuadrada del tamaño de la muestra (número de elementos)

```
A <- seq(10, -6, by = -2)
A
## [1] 10  8  6  4  2  0 -2 -4 -6
print(paste0("La longitud del vector es: ", length(A), ". La suma del
vector es: ", sum(A), ". La media del vector es: ", mean(A), ". El error
típico es: ", sd(A)/sqrt( length(A) ) ))
## [1] "La longitud del vector es: 9. La suma del vector es: 18. La media
del vector es: 2. El error típico es: 1.82574185835055"
```

2. Utilizando las funciones `rep` y `seq`, cree el vector B, con los siguientes datos: 0, 0, 0, 0, 1, 2.5, 4, 5.5. Calcule: la suma de A y B, la diferencia entre A y B,  $\sqrt{A^2 + B^2}$

```
B <- c(rep(0,5), seq(1,5.5,1.5))
B
## [1] 0.0 0.0 0.0 0.0 0.0 1.0 2.5 4.0 5.5
print(A+B)
```

```
## [1] 10.0  8.0  6.0  4.0  2.0  1.0  0.5  0.0 -0.5

print(A-B)

## [1] 10.0  8.0  6.0  4.0  2.0 -1.0 -4.5 -8.0 -11.5

print(sqrt((A^2 + B^2)))

## [1] 10.000000  8.000000  6.000000  4.000000  2.000000  1.000000
3.201562
## [8]  5.656854  8.139410
```

## Matrices

Se dispone de una tabla de datos:

3. Cree la matriz M con estos datos, indicando el nombre de cada columna. Muestre la dimensión de la matriz. Utilizando la función apply, obtenga un promedio de los resultados por columna

```
M <- matrix ( c(74, 85, 58, 80, 1.68, 1.83, 1.58, 1.72, 22, 25, 21, 20),
              nrow = 4,
              ncol= 3)

colnames(M) <- c("PESO", "ESTATURA", "EDAD")

print(dim(M))

## [1] 4 3

apply(M, 2, function(x) mean(x))

##      PESO ESTATURA      EDAD
## 74.2500  1.7025  22.0000
```

## Dataframes

4. Con los datos de la matriz anterior, cree el dataframe datos. Añada dos columnas, con el nombre de los individuos: "Pedro", "José", "María", "Javier"; y el sexo: "Hombre" y "Mujer". Añada un nuevo individuo al conjunto de datos: "Ana", peso=70, estatura=1.72 y edad=24. La variable nombre es de tipo carácter, sexo es un factor y el resto son numéricas. Muestre la estructura del dataframe datos.

```
nombre <- c("Pedro", "José", "María", "Javier")
sexo <- c("Hombre", "Hombre", "Mujer", "Hombre")

datos <- cbind(as.data.frame(M),
              nombre,
              sexo)

individuo <- data.frame(PESO=70, ESTATURA=1.72, EDAD=24, nombre="Ana",
                        sexo="Mujer")
datos <- rbind(datos, individuo)
```

```

datos$PESO <- as.numeric(as.character(datos$PESO))
datos$ESTATURA <- as.numeric(as.character(datos$ESTATURA))
datos$EDAD <- as.numeric(as.character(datos$EDAD))
datos$nombre <- as.character(datos$nombre)
datos$sexo <- as.factor(datos$sexo)

str(datos)

## 'data.frame':    5 obs. of  5 variables:
## $ PESO      : num  74 85 58 80 70
## $ ESTATURA: num  1.68 1.83 1.58 1.72 1.72
## $ EDAD      : num  22 25 21 20 24
## $ nombre    : chr  "Pedro" "José" "María" "Javier" ...
## $ sexo      : Factor w/ 2 levels "Hombre","Mujer": 1 1 2 1 2

```

- Del dataframe creado, extraiga: el peso de Ana, todos los nombres y los datos de Javier. Añada una columna al dataframe datos, denominada ALTURA. De tal modo que, si el individuo mide más 1.78, tendrá el valor “Alto”; y de lo contrario, tendrá el valor “Normal”.

```

datos [datos$nombre == "Ana", "PESO"]

## [1] 70

datos$nombre

## [1] "Pedro" "José" "María" "Javier" "Ana"

datos [datos$nombre == "Javier", ]

##  PESO ESTATURA EDAD nombre  sexo
## 4   80      1.72   20 Javier Hombre

datos$ALTURA <- ifelse( datos$ESTATURA > 1.78, "Alto", "Normal" )
str(datos)

## 'data.frame':    5 obs. of  6 variables:
## $ PESO      : num  74 85 58 80 70
## $ ESTATURA: num  1.68 1.83 1.58 1.72 1.72
## $ EDAD      : num  22 25 21 20 24
## $ nombre    : chr  "Pedro" "José" "María" "Javier" ...
## $ sexo      : Factor w/ 2 levels "Hombre","Mujer": 1 1 2 1 2
## $ ALTURA   : chr  "Normal" "Alto" "Normal" "Normal" ...

```

## Listas

- Cree la lista L, con todos los objetos que ha creado. Encuentre en la lista, los valores de A que cumplen la condición  $A+2 < 0$ , el valor [3,2] de la matriz M, y los datos de Ana del dataframe datos.

```

L <- list (A, B, M, datos)
L

```

```
## [[1]]
## [1] 10  8  6  4  2  0 -2 -4 -6
##
## [[2]]
## [1] 0.0 0.0 0.0 0.0 0.0 1.0 2.5 4.0 5.5
##
## [[3]]
##      PESO  ESTATURA  EDAD
## [1,]   74      1.68   22
## [2,]   85      1.83   25
## [3,]   58      1.58   21
## [4,]   80      1.72   20
##
## [[4]]
##      PESO  ESTATURA  EDAD  nombre  sexo  ALTURA
## 1    74      1.68   22   Pedro  Hombre  Normal
## 2    85      1.83   25    José  Hombre   Alto
## 3    58      1.58   21   María  Mujer  Normal
## 4    80      1.72   20   Javier  Hombre  Normal
## 5    70      1.72   24     Ana  Mujer  Normal

sapply( L[[1]], function(x) x + 2 < 0 )

## [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE

L[[1]][A + 2 < 0]

## [1] -4 -6

print(paste0("El valor es: ", L[[3]][3,2], " corresponde a la columna",
",names(L[[3]][3,2]))

## [1] "El valor es: 1.58 corresponde a la columna ESTATURA"

L[[4]][ L[[4]]$nombre == "Ana", ]

##      PESO  ESTATURA  EDAD  nombre  sexo  ALTURA
## 5    70      1.72   24     Ana  Mujer  Normal
```

## Funciones y Bucles

7. Cree la función normalizar, que realice la normalización estándar (restar la media y dividir por el error típico) y la normalización min-max (restar el mínimo y dividir por la diferencia entre el máximo y el mínimo). Dicha función devolverá una lista. Utilice esta función empleando como argumento el PESO del dataframe datos, y después empleando la ESTATURA.

```
normalizador <- function(x)
{
  min <- min(x)
  max <- max(x)
  num <- length(x)
  media <- mean(x)
```

```

error <- sd(x) / sqrt(num)
norm_estandar <- ( (x-media) )/error
norm_mmax <- ( (x-min)/(max-min) )

list_norm <- list(norm_estandar, norm_mmax)
names(list_norm) <- c("Normalización Estándar", "Normalización Min-Max")
return(list_norm)
}

normalizador(datos$PESO)

## $`Normalización Estándar`
## [1]  0.1298227  2.5099055 -3.3321159  1.4280497 -0.7356619
##
## $`Normalización Min-Max`
## [1] 0.5925926 1.0000000 0.0000000 0.8148148 0.4444444

normalizador(datos$ESTATURA)

## $`Normalización Estándar`
## [1] -0.6467742  3.0846153 -3.1343671  0.3482630  0.3482630
##
## $`Normalización Min-Max`
## [1] 0.40 1.00 0.00 0.56 0.56

```

8. Mediante un bucle for, cree un vector que contenga los primeros 100 elementos de la siguiente sucesión numérica:  $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \dots$ . Para resolver el ejercicio, tenga en cuenta lo siguiente:  $\frac{1}{2^1}, \frac{1}{2^2}, \frac{1}{2^3}, \frac{1}{2^4}, \frac{1}{2^5}, \dots$

```

vector <- c()

# Me permito hacer un bucle ya que es pequeño (R no trabaja bien con bucles grandes)

for(i in 1:100)
{
  vector[i] <- 1/2 ^ i
}

sum(vector)

## [1] 1

```

## Dataframes y gráficos

En los siguientes ejercicios trabajará con el dataset mtcars, que consta de 32 coches y 11 variables. Descripción de las variables que se van a utilizar:

- mpg: Millas recorridas por galón de combustible.
- cyl: Número de cilindros.

- hp: Caballos de fuerza.
  - wt: Peso (1000 libras).
  - qsec: Tiempo empleado en recorrer 1/4 de milla (segundos)
9. Cuáles son los 5 coches más pesados? ¿Qué coches tienen 8 cilindros y 175 o más caballos de fuerza? (ordene el resultado por caballos de fuerza) ¿Cuál es el coche más rápido? ¿Cuál es el coche que menos combustible consume? ¿Cuál es el consumo medio de los coches de 4 cilindros?

```
mtcars <- mtcars
str(mtcars)

## 'data.frame': 32 obs. of 11 variables:
## $ mpg : num 21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
## $ cyl : num 6 6 4 6 8 6 8 4 4 6 ...
## $ disp: num 160 160 108 258 360 ...
## $ hp : num 110 110 93 110 175 105 245 62 95 123 ...
## $ drat: num 3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
## $ wt : num 2.62 2.88 2.32 3.21 3.44 ...
## $ qsec: num 16.5 17 18.6 19.4 17 ...
## $ vs : num 0 0 1 1 0 1 0 1 1 1 ...
## $ am : num 1 1 1 0 0 0 0 0 0 0 ...
## $ gear: num 4 4 4 3 3 3 3 4 4 4 ...
## $ carb: num 4 4 1 1 2 1 4 2 2 4 ...

mtcars[ order(-mtcars$wt)[1:5], ]

##           mpg cyl  disp  hp drat   wt  qsec vs am gear carb
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0   3
## Chrysler Imperial  14.7   8 440.0 230 3.23 5.345 17.42  0  0   3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0   3
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0   3
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0   3

filtro <- mtcars$cyl == 8 & mtcars$hp >= 175
filtered <- mtcars[filtro,]
filtered[ order(filtered$hp), ]

##           mpg cyl  disp  hp drat   wt  qsec vs am gear carb
## Hornet Sportabout  18.7   8 360.0 175 3.15 3.440 17.02  0  0   3
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0   3
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0   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   3
4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0   3
4
## Chrysler Imperial 14.7   8 440.0 230 3.23 5.345 17.42  0  0   3
4
## Duster 360      14.3   8 360.0 245 3.21 3.570 15.84  0  0   3
4
## Camaro Z28       13.3   8 350.0 245 3.73 3.840 15.41  0  0   3
4
## Ford Pantera L   15.8   8 351.0 264 4.22 3.170 14.50  0  1   5
4
## Maserati Bora     15.0   8 301.0 335 3.54 3.570 14.60  0  1   5
8
```

```
mtcars[mtcars$qsec == max(mtcars$qsec),]
```

```
##           mpg cyl  disp hp drat   wt  qsec vs am gear carb
## Merc 230 22.8   4 140.8 95 3.92 3.15 22.9  1  0    4    2
```

```
mtcars[mtcars$mpg == max(mtcars$mpg),]
```

```
##           mpg cyl  disp hp drat   wt  qsec vs am gear carb
## Toyota Corolla 33.9   4 71.1 65 4.22 1.835 19.9  1  1    4    1
```

```
print(paste0("El consumo medio de los de 4 cilindros es de: ",
mean(mtcars[mtcars$cyl==4,]$mpg) ))
```

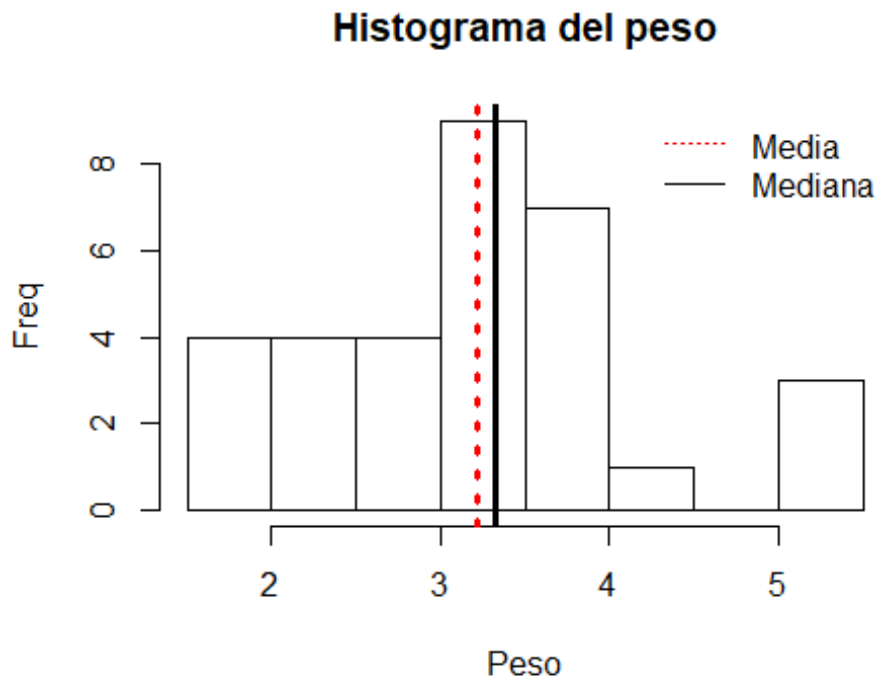
```
## [1] "El consumo medio de los de 4 cilindros es de: 26.6636363636364"
```

10. Cree un histograma del peso de los coches, y mediante la función `abline` dibuje: una línea vertical (roja) en la media de la distribución y otra línea vertical (verde) en la mediana de la distribución.

```
hist (mtcars$wt, main = "Histograma del peso",
      xlab = "Peso",
      ylab = "Freq")
```

```
abline(v=mean(mtcars$wt),col=c("red"),lwd=3, lty=3)
abline(v=median(mtcars$wt),col=c("#010101"),lwd=3, lty=1)
```

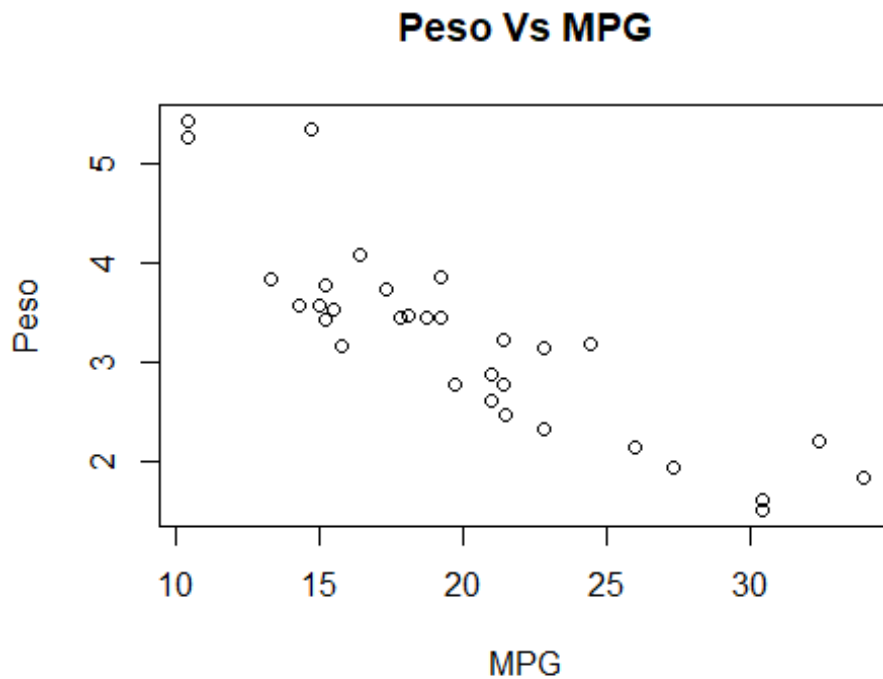
```
legend("topright", # No cabe en otro sitio
      c("Media",
        "Mediana"),
      lty=c(3,1),
      col=c("red",
            "#010101"),
      bty = "n")
```



11. Cree un gráfico que muestre la relación entre el peso y las millas por galón: mpg en función de wt. Interprete el gráfico.

```
plot(mtcars$mpg,  
      mtcars$wt,  
      xlab = "MPG",  
      ylab = "Peso",  
      main = "Peso Vs MPG",  
      col = "black")
```





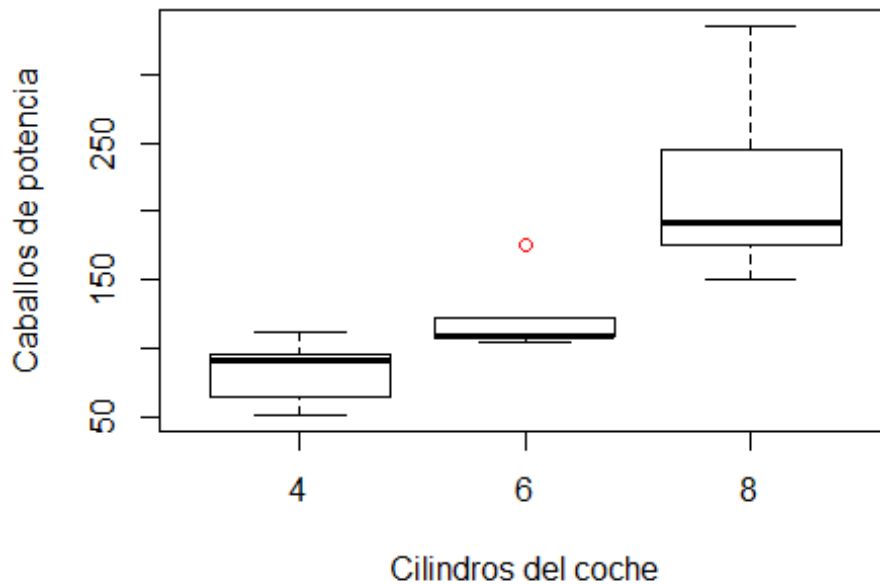
La gráfica del apartado 11 permite la visualización de la relación entre el consumo de combustible (millas por galón) y el peso de los coches. Como se puede ver, la relación conlleva que una disminución en el peso suele conllevar un mayor número de millas recorridas por galón de combustible. La relación, a falta de comprobación estadística con modelo (lm), parece ser ciertamente lineal.

12. Convierta la variable cyl en un factor. A continuación, cree un diagrama de caja que represente los caballos de fuerza en función del número de cilindros. Interprete el gráfico.

```
mtcars_ <- mtcars
mtcars_$cyl <- as.factor(mtcars_$cyl)

boxplot(mtcars_$hp ~ mtcars_$cyl,
        xlab = "Cilindros del coche",
        ylab = "Caballos de potencia",
        main = "Relación entre los caballos y los cilindros",
        outcol="red"
        )
```

## Relación entre los caballos y los cilindros



Como se puede observar, el gráfico está dividido en 3 boxplots: Uno para 4 cilindros, uno para 6 y uno para 8 cilindros. Se puede observar la evolución de las medianas como elemento más destacado (la mediana no se desvirtúa demasiado con outliers, mientras que la media sí). Las medianas indican un aumento de potencia conforme los coches tienen mayor cilindrada.

Posteriormente, se puede observar el IQR (Rango intercuartílico) de los boxplots. Se observa como los coches con 6 cilindros apenas tienen variabilidad, mientras que los coches con 4 u 8 cilindros presentan una mayor (en especial los de 8, con los bigotes bastante separados).

Entrando al dato, los coches con 4 cilindros tienen una potencia mediana de unos 95 CV, los de 6 de unos 110 CV y los de 8 cilindros rondan los 170 CV de mediana.

Finalmente, comentar el tema del outlier. Sólo existe un valor outlier en todo el gráfico, y se encuentra en el boxplot de los 6 cilindros. Este trata de un coche con potencia cercana a los 200 Cv, cosa que destaca mucho en su grupo y que, por supuesto, se sale de los bigotes. Por ello, está marcado como rojo gracias a "outcol".