

PECL1 - Fundamentos de la ciencia de datos

Marcos Barranquero Adrián Montesinos
Eduardo Graván

14 de octubre de 2019

1. Apartado 1 - Ejercicios sobre los datos

1.1. Introducción

En el siguiente documento presentamos los resultados de la Práctica 1 de FCD. En ella, se han realizado un estudio estadístico sobre una variable de un conjunto de variables recogidas en un archivo. Para cada ejercicio, se pide realizar:

1. Frecuencias
2. Media aritmética
3. Varianza
4. Desviación típica
5. Rango
6. Mediana
7. Cuartiles
8. Cuantil 54

Aunque tenemos dos ejercicios claramente diferenciados, ambos estudios se han realizado en el mismo archivo de código R, por lo que comparten la misma estructura inicial:

```
> # Cargamos las librerías:  
> source("../Auxiliar/libreria.r") # Librería con algunas funciones  
> library(foreign) # Librería foreign  
> library(xtable)
```

Además de establecer el directorio de trabajo - cosa que no se puede hacer desde el Code Chunk - cargamos la librería *foreign* y las diferentes funciones de la librería creada en clase *libreria*:

```

rango <-
function(vector)(max(vector)-min(vector))

frecuencia_absoluta <-
function(vector)(table(" "=vector))

frecuencia_absoluta_acumulada <-
function(vector)(cumsum(frecuencia_absoluta(vector)))

frecuencia_relativa <-
function(vector)(round(frecuencia_absoluta(vector)/length(vector), 4))

frecuencia_relativa_acumulada <-
function(vector)(round(frecuencia_absoluta_acumulada(vector)/length(vector), 4))

```

2. Ejercicio 1

En este ejercicio se estudia el **radio** de los distintos satélites del planeta Urano. Utilizamos un archivo *txt* del que cargamos los datos y extraemos la variable.

```

> # Se importa el archivo y se almacena en una variable
> satelites <- read.table("./Datos/satelites.txt")
> # Sacamos la variable con la que queremos trabajar de la tabla:
> radio <- satelites$radio
> # Observamos el vector de datos:
> radio

[1] 13 16 22 33 39 42 27 34 20 30 20 15

```

Una vez con el vector de radios extraído en la variable *radio*, procedemos a hacer las diferentes operaciones solicitadas:

2.1. Frecuencias

Utilizamos las funciones de frecuencia creadas en la librería *libreria* que realizamos en clase.

```

> # Frecuencias absolutas:
> frecuencia_absoluta(radio)

13 15 16 20 22 27 30 33 34 39 42
 1  1  1  2  1  1  1  1  1  1  1

> frecuencia_absoluta_acumulada(radio)

13 15 16 20 22 27 30 33 34 39 42
 1  2  3  5  6  7  8  9 10 11 12

> # Frecuencias relativas:
> frecuencia_relativa(radio)

```

```

      13      15      16      20      22      27      30      33      34      39      42
0.0833 0.0833 0.0833 0.1667 0.0833 0.0833 0.0833 0.0833 0.0833 0.0833 0.0833

```

```
> frecuencia_relativa_acumulada(radio)
```

```

      13      15      16      20      22      27      30      33      34      39      42
0.0833 0.1667 0.2500 0.4167 0.5000 0.5833 0.6667 0.7500 0.8333 0.9167 1.0000

```

Utilizando la librería xtable, podemos disponer la información de forma más visual y adaptada:

2.1.1. Frecuencia absoluta

Elemento	Apariciones
13	1
15	1
16	1
20	2
22	1
27	1
30	1
33	1
34	1
39	1
42	1

2.1.2. Frecuencia absoluta acum.

Elem.	Apariciones acum.
13	1
15	2
16	3
20	5
22	6
27	7
30	8
33	9
34	10
39	11
42	12

2.1.3. Frecuencia relativa

Elemento	Apariciones rel.
13	0.08
15	0.08
16	0.08
20	0.17
22	0.08
27	0.08
30	0.08
33	0.08
34	0.08
39	0.08
42	0.08

2.1.4. Frecuencia relativa acum.

Elem.	Apariciones relativas acum.
13	0.0833
15	0.1667
16	0.2500
20	0.4167
22	0.5000
27	0.5833
30	0.6667
33	0.7500
34	0.8333
39	0.9167
42	1.0000

2.2. Media Aritmética

Contamos con la función integrada de R que permite calcular la media aritmética sobre un vector, aproximando a 2 decimales.

```
> round(mean(radius), 2)
```

```
[1] 25.92
```

2.3. Varianza

Contamos con la función integrada de R que permite calcular la varianza sobre un vector, aproximando a 2 decimales.

```
> round(var(radius), 2)
```

```
[1] 93.9
```

2.4. Desviación típica

Contamos con la función integrada de R que permite hacer la desviación típica sobre un vector, aproximando a 2 decimales.

```
> round(sd(radio), 2)
```

```
[1] 9.69
```

2.5. Rango

Llamamos a la función *rango* definida en la librería *libreria*.

```
> rango(radio)
```

```
[1] 29
```

2.6. Mediana

Contamos con la función integrada de R que permite calcular la media sobre un vector.

```
> median(radio)
```

```
[1] 24.5
```

2.7. Cuartiles

Contamos con la función integrada de R que permite calcular el cuantil especificado sobre un vector.

```
> # Cuartil 1
```

```
> quantile(radio, 0.25)
```

```
25%
```

```
19
```

```
> # Cuartil 2
```

```
> quantile(radio, 0.50)
```

```
50%
```

```
24.5
```

```
> # Cuartil 3
```

```
> quantile(radio, 0.75)
```

```
75%
```

```
33.25
```

```
> # Cuartil 4
```

```
> quantile(radio, 1.00)
```

```
100%
```

```
42
```

2.8. Cuantil 54

```
> quantile(radio, 0.54)
```

```
54%  
26.7
```

3. Ejercicio 2

En este ejercicio se estudia la variable **mpg** de un archivo *spss* que contiene datos relativos a coches. En primer lugar volvemos a extraer la variable del archivo:

```
> # Se importa el archivo y se almacena en una variable:  
> automoviles <- read.spss("./Datos/cardata.sav")  
> # Sacamos la variable con la que queremos trabajar de la tabla:  
> mpg <- automoviles$mpg  
> # Eliminamos los valores NA  
> mpg <- mpg[!is.na(mpg)]  
> # Observamos el vector de datos  
> mpg
```

```
[1] 36.1 19.9 19.4 20.2 19.2 20.5 20.2 25.1 20.5 19.4 20.6 20.8 18.6 18.1 19.2  
[16] 17.7 18.1 17.5 30.0 30.9 23.2 23.8 21.5 19.8 22.3 20.2 20.6 17.0 17.6 16.5  
[31] 18.2 16.9 15.5 19.2 18.5 35.7 27.4 23.0 23.9 34.2 34.5 28.4 28.8 26.8 33.5  
[46] 32.1 28.0 26.4 24.3 19.1 27.9 23.6 27.2 26.6 25.8 23.5 30.0 39.0 34.7 34.4  
[61] 29.9 22.4 26.6 20.2 17.6 28.0 27.0 34.0 31.0 29.0 27.0 24.0 23.0 38.0 36.0  
[76] 25.0 38.0 26.0 22.0 36.0 27.0 27.0 32.0 28.0 31.0 43.1 20.3 17.0 21.6 16.2  
[91] 31.5 31.9 25.4 27.2 37.3 41.5 34.3 44.3 43.4 36.4 30.4 40.9 29.8 35.0 33.0  
[106] 34.5 28.1 30.7 36.0 44.0 32.8 39.4 36.1 27.5 27.2 21.1 23.9 29.5 34.1 31.8  
[121] 38.1 37.2 29.8 31.3 37.0 32.2 46.6 40.8 44.6 33.8 32.7 23.7 32.4 39.1 35.1  
[136] 32.3 37.0 37.7 34.1 33.7 32.4 32.9 31.6 25.4 24.2 37.0 31.0 36.0 36.0 34.0  
[151] 38.0 32.0 38.0 32.0
```

3.1. Frecuencias

Aunque en el script de R se han calculado las frecuencias, se ha decidido no incluirlas en el documento ya que son tablas de gran tamaño que realmente no aportan información extra.

El código que calcula dichas frecuencias es el siguiente:

```
# Frecuencias absolutas:  
frecuencia_absoluta(mpg)  
frecuencia_absoluta_acumulada(mpg)  
# Frecuencias relativas:  
frecuencia_relativa(mpg)  
frecuencia_relativa_acumulada(mpg)
```

3.2. Media Aritmética

Contamos con la función integrada de R que permite calcular la media aritmética sobre un vector, aproximando a 2 decimales.

```
> round(mean(mpg), 2)
```

```
[1] 28.79
```

3.3. Varianza

Contamos con la función integrada de R que permite calcular la varianza sobre un vector, aproximando a 2 decimales.

```
> round(var(mpg), 2)
```

```
[1] 54.42
```

3.4. Desviación típica

Contamos con la función integrada de R que permite hacer la desviación típica sobre un vector, aproximando a 2 decimales.

```
> round(sd(mpg), 2)
```

```
[1] 7.38
```

3.5. Rango

Llamamos a la función *rango* definida en la librería *libreria*.

```
> rango(mpg)
```

```
[1] 31.1
```

3.6. Mediana

Contamos con la función integrada de R que permite calcular la media sobre un vector.

```
> median(mpg)
```

```
[1] 28.9
```

3.7. Cuartiles

Contamos con la función integrada de R que permite calcular el cuantil especificado sobre un vector.

```
> # Cuartil 1  
> quantile(mpg, 0.25)
```

```
25%  
22.55
```

```
> # Cuartil 2
> quantile(mpg, 0.50)
```

```
50%
28.9
```

```
> # Cuartil 3
> quantile(mpg, 0.75)
```

```
75%
34.275
```

```
> # Cuartil 4
> quantile(mpg, 1.00)
```

```
100%
46.6
```

3.8. Cuantil 54

```
> quantile(mpg, 0.54)
```

```
54%
30
```

4. Apartado 2 - Ampliación de ejercicio

En este apartado se propone ampliar los ejercicios propuestos, introduciendo modificaciones y haciendo uso de otras funciones y librerías. Se han instalado los paquetes de *xlsx* y *pastecs* mediante el siguiente comando:

```
install.packages("pastecs")
install.packages("xlsx")
```

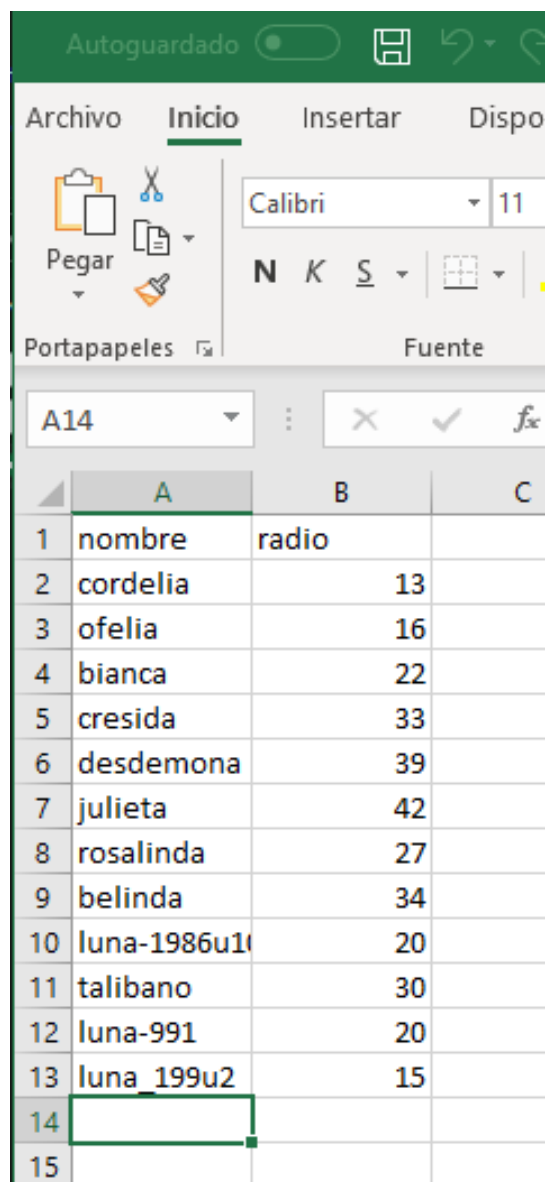
Después, se han vertido los datos en un archivo Excel (Figura 1):
Tras esto, se ha procedido a leer los datos:

```
> library(xlsx)
> satelites <- read.xlsx("./Datos/satelites.xlsx", 1)
> radio <- satelites$radio
```

4.1. Uso de data, frame, y table

Una vez leídos, se ha utilizado *table()* para calcular las frecuencias absolutas y *prop.table()* para el cálculo de las frecuencias relativas. Usamos *data.frame()* para manipular las tablas usando nombres de columna y para mostrarlo en una sola tabla con nombres.

```
> # Calculamos frecuencias
> frecAbs <- data.frame(table(radio))
> frecAbsAcum <- cumsum(frecAbs["Freq"])
> frecRel <- prop.table(frecAbs["Freq"])
```

	A	B	C
1	nombre	radio	
2	cordelia	13	
3	ofelia	16	
4	bianca	22	
5	cresida	33	
6	desdemona	39	
7	julieta	42	
8	rosalinda	27	
9	belinda	34	
10	luna-1986u10	20	
11	talibano	30	
12	luna-991	20	
13	luna_199u2	15	
14			
15			

Figura 1: Datos en el archivo Excel

```

> frecRelAcum <- cumsum(frecRel["Freq"])
> # Utilizamos data.frame() para unirlo todo en una tabla.
> frecuencias <- data.frame(
+   frecAbs["radio"],
+   frecAbs["Freq"],
+   frecAbsAcum["Freq"],
+   frecRel["Freq"],
+   frecRelAcum["Freq"]
+ )
> # Asignamos nombres:
> names(frecuencias) <- c("Radio", "Absoluta", "Abs. acumulada",
+   "Relativa", "Rel. acumulada")
> # Imprimimos tabla:
> frecuencias

```

	Radio	Absoluta	Abs. acumulada	Relativa	Rel. acumulada
1	13	1	1	0.08333333	0.08333333
2	15	1	2	0.08333333	0.16666667
3	16	1	3	0.08333333	0.25000000
4	20	2	5	0.16666667	0.41666667
5	22	1	6	0.08333333	0.50000000
6	27	1	7	0.08333333	0.58333333
7	30	1	8	0.08333333	0.66666667
8	33	1	9	0.08333333	0.75000000
9	34	1	10	0.08333333	0.83333333
10	39	1	11	0.08333333	0.91666667
11	42	1	12	0.08333333	1.00000000

4.2. Uso de pastecs

Alternativamente, empleamos el paquete *pastecs* para el cálculo de la media, mediana, mínimo, máximo y medidas de dispersión.

```

> # Cargamos la librería
> library(pastecs)
> # cargamos funciones sobre la tabla
> dispersion <- stat.desc(radio)[c("mean", "var", "std.dev", "range",
+   "min", "max", "median")]
> # Establecemos nombres para la tabla
> names(dispersion) <- c("Media", "Varianza", "Desv. tipica", "Rango",
+   "Minimo", "Maximo", "Mediana")
> # Mostramos por pantalla
> dispersion

```

	Media	Varianza	Desv. tipica	Rango	Minimo	Maximo
	25.916667	93.901515	9.690279	29.000000	13.000000	42.000000
Mediana	24.500000					

```

>

```

Finalmente empleamos `summary` para generar máximo, mínimo, cuartiles y media ordenador según aparecen. Añadimos el cuantil 54

```
> cuantil54 <- quantile(radio, 0.54)
> cuantiles <- c(summary(radio), cuantil54)
> cuantiles <- cuantiles[order(unlist(cuantiles))] # Ordenamos el cuantil 54%
> print(cuantiles)
```

	Min.	1st Qu.	Median	Mean	54%	3rd Qu.	Max.
	13.00000	19.00000	24.50000	25.91667	26.70000	33.25000	42.00000

5. Conclusiones

Mediante esta práctica nos hemos familiarizado con el uso del lenguaje de programación de R, así como su IDE *Rgui*, *Rstudio* y el empleo de *Latex* para la producción de documentos científicos relacionados con el estudio de datos, empleando la herramienta *Sweavy*.

Vemos que es un lenguaje potente, que en pocos segundos es capaz de analizar grandes cantidades de datos y con un gran soporte de la comunidad a la hora de crear nuevos paquetes, algoritmos y funcionalidades.