PEC1 Análisis Estadístico

Stwart Porras

2024-10-29

Contents

Preparación del conjunto de datos

1.1 Cargar el archivo de datos

Para leer el archivo de datos, es necesario, utilizar la funcion read_csv, e incluir la ruta del archivo, si éste, está contenido dentro del entorno de trabajo, bastaría con escribir el nombre del archivo. Además hay que especificar una serie de parámetros, los cuales son importantes para que la lectura se realice correctamente. Header = True, es porque el conjunto de datos contiene cabeceros, skip=1, es necesario, ya que la primera fila del conjunto de datos es innecesaria, los encabezados comienzan en la 2 fila. El separador nos indicar el delimitador de las columnas.

```
df <- read.csv("gem01.csv", header = TRUE, skip = 1 ,sep = ";")
```

1.3 Nombres de las columnas

Para cambiar el nombre a las columnas a los especificados en la tarea, es necesario hacer uso de la librería dplyr, de esta manera se pueden cambiar el nombre de las diferentes columnas al mismo tiempo.

```
suppressMessages(library(dplyr))
```

```r
df <- df  %>% dplyr::rename(
    OPP = Perceived.opportunities,
    PC = Perceived.capabilities,
    FAIL = Fear.of.failure.rate..,
    EI = Entrepreneurial.intentions,
    TEA = Total.early.stage.Entrepreneurial.Activity..TEA.,
    OWN = Established.Business.Ownership,
    EMPL = Entrepreneurial.Employee.Activity,
    MOT = Motivational.Index,
    FMTEA = Female.Male.TEA,
    FMOD = Female.Male.Opportunity.Driven.TEA,
    JOB = High.Job.Creation.Expectation,
    INNOV = Innovation,
    BSER = Business.Services.Sector,
    STAT = High.Status.to.Successful.Entrepreneurs,
    CHOI = Entrepreneurship.as.a.Good.Career.Choice)
```

1.2 Descripción de los datos

Para describir los datos, se puede hacer uso del comando str, el cual nos indica los nombres de las columnas, su tipo de datos, y un ejemplo de los primeros valores de cada columna.

```r
print(str(df))
```

```
## 'data.frame':    1128 obs. of  18 variables:
##  $ code   : int  1 27 30 31 33 34 36 39 40 41 ...
##  $ economy: chr  "United States" "South Africa" "Greece" "Netherlands" ...
##  $ year   : int  2023 2023 2023 2023 2023 2023 2023 2023 2023 2023 ...
##  $ OPP    : num  53.8 64.1 45.3 67.4 50.9 ...
##  $ PC     : num  49 69.2 53.8 46 49.5 ...
##  $ FAIL   : num  44.5 59.5 53.2 40.3 40.1 ...
##  $ EI     : num  12.09 7.45 9.1 16.02 13.41 ...
##  $ TEA    : num  14.71 11.11 6.74 13.69 10.75 ...
##  $ OWN    : num  6.74 5.92 14.7 6.92 4.56 6.73 7.38 7.75 5.09 5.83 ...
##  $ EMPL   : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ MOT    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ FMTEA  : num  0.84 0.76 0.79 0.8 0.72 0.81 0.56 0.61 0.71 0.9 ...
##  $ FMOD   : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ JOB    : num  28.8 21.4 12.1 15.1 23.2 ...
##  $ INNOV  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ BSER   : num  20.42 3.57 22.53 23.65 32.58 ...
##  $ STAT   : num  78.9 85.5 69.8 NA 51.8 ...
##  $ CHOI   : num  79.2 78.5 72.9 NA 65.4 ...
## NULL
```

Los nombres de los países deben empezar con letra mayúscula. Para ello se usa la librería stringr, junto con su función to title

```r
suppressMessages(library(stringr))
Capital_countries_by_word <- function(x){

  str_to_title(x)
}
```

```r
df$economy <- Capital_countries_by_word(df$economy)
```

```r
head(df$economy,5)
```

```
## [1] "United States" "South Africa"  "Greece"        "Netherlands"
## [5] "France"
```

2 Normalizacion de variables cualitativas

Revisad los nombres de los países (variable economy) y si hay inconsistencias, corregidlas. A continuación, mostrar una tabla con los nombres de los países, el rango de años (mínimo y máximo año) por cada país, y el número de años en los que se disponen de datos por cada país. Añadid en la tabla un índice desde 1 hasta el número de países. Se pueden utilizar las funciones group_by y summarise de la librería dplyr.

Se usa group by y summarize como menciona el enunciado, se usa la forma %>% para usar el resultado de la primera operación, como argumento de la siguiente operación.

```r
table <- df %>% group_by(economy) %>%
  summarize(
    index = cur_group_id(),
    year_min = min(year),
    year_max = max(year),
    number_of_years = n_distinct(year)
  )
```

```r
print(table)
```

```
## # A tibble: 118 x 5
##    economy    index year_min year_max number_of_years
##    <chr>      <int>    <int>    <int>           <int>
##  1 Algeria        1     2009     2013               4
##  2 Angola         2     2008     2020               7
##  3 Argentin       3     2001     2001               1
##  4 Argentina      4     2002     2018              17
##  5 Armenia        5     2019     2019               1
##  6 Australia      6     2001     2019              13
##  7 Austria        7     2005     2022               8
##  8 Bangladesh     8     2011     2011               1
##  9 Barbados       9     2011     2015               5
## 10 Belarus       10     2019     2021               2
## # i 108 more rows
```

3 Inconsistencias en variables cuantitativas

3.1 Formato

Revisad los valores de las variables cuantitativas y aplicad las transformaciones de formato necesarias. Si hay valores vacíos, se deben sustituir por NA. A continuación, mostrad en una tabla cuántos valores con NAs hay por cada variable numérica. En este apartado no se realiza ninguna imputación. Los valores NA se imputarán más adelante. También mostrad una tabla con los rangos (valores mínimos y máximos) de cada variable numérica. Si existen valores erróneos o atípicos, éstos se tratarán en los siguientes apartados.

Revision de variables cuantitativas. Se puede filtrar solo las variables del DataFrame que son numéricas usando el comando select_if, e is numeric. Existen más formas, más adelante se usará otra forma, para seleccionar los datos sin necesidad del select_if.

```r
numeric_subset <- df %>% select_if(is.numeric)
```

```r
print(str(numeric_subset))
```

```
## 'data.frame':    1128 obs. of  17 variables:
##  $ code : int  1 27 30 31 33 34 36 39 40 41 ...
##  $ year : int  2023 2023 2023 2023 2023 2023 2023 2023 2023 2023 ...
##  $ OPP  : num  53.8 64.1 45.3 67.4 50.9 ...
##  $ PC   : num  49 69.2 53.8 46 49.5 ...
##  $ FAIL : num  44.5 59.5 53.2 40.3 40.1 ...
##  $ EI   : num  12.09 7.45 9.1 16.02 13.41 ...
##  $ TEA  : num  14.71 11.11 6.74 13.69 10.75 ...
##  $ OWN  : num  6.74 5.92 14.7 6.92 4.56 6.73 7.38 7.75 5.09 5.83 ...
##  $ EMPL : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ MOT  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ FMTEA: num  0.84 0.76 0.79 0.8 0.72 0.81 0.56 0.61 0.71 0.9 ...
##  $ FMOD : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ JOB  : num  28.8 21.4 12.1 15.1 23.2 ...
##  $ INNOV: num  NA NA NA NA NA NA NA NA NA NA ...
##  $ BSER : num  20.42 3.57 22.53 23.65 32.58 ...
##  $ STAT : num  78.9 85.5 69.8 NA 51.8 ...
##  $ CHOI : num  79.2 78.5 72.9 NA 65.4 ...
## NULL
```

Cuántos valores con NAs hay por cada variable numérica.

Para esta tarea, se usa el comando sapply para aplicar una función a cada variable del dataframe, se crea una función ´handler´ similar a la de MATLAB que se hace con una @, o la función lambda en python, para aplicar una función "ánonima". Se suman los valores TRUE=1 de NA y de esta manera se cuenta cuantos valores son NA por cada variable

```r
Count_NA <- sapply(df,function(x) sum(is.na(x)))
```

```r
NA_Table_display <- data.frame(Column = names(Count_NA), Number_of_NA = Count_NA)
```

```r
print(NA_Table_display)
```

```
##          Column Number_of_NA
## code       code            0
## economy economy            0
## year       year            0
## OPP         OPP            0
## PC           PC            0
## FAIL       FAIL            1
## EI           EI           28
## TEA         TEA            0
## OWN         OWN            1
## EMPL       EMPL          580
## MOT         MOT          580
## FMTEA     FMTEA            0
## FMOD       FMOD          761
## JOB         JOB            3
## INNOV     INNOV          639
## BSER       BSER           38
## STAT       STAT          133
## CHOI       CHOI          135
```

Tabla con los rangos (valores mínimos y máximos) de cada variable numérica. De forma similar al apartado anterior se calculan los valores mínimos y máximos y se combinan los vectores min y max con el comando rbind en un dataframe que en este caso se llama table

```
numeric_sample <- df[sapply(df,is.numeric)]
min <- sapply(numeric_sample, function(x) min(x, na.rm = TRUE))
max <- sapply(numeric_sample, function(x) max(x,na.rm = TRUE))
```

```
table <- rbind(Min = min, Max = max)
print(table)
```

```
##     code year   OPP    PC   FAIL    EI   TEA   OWN   EMPL   MOT  FMTEA FMOD
## Min    1 2001  2.85  8.65   7.14  0.75  1.48  0.42   0.05  0.35 -99.00 0.51
## Max  995 2023 95.38 91.90 169.00 90.95 52.11 37.74 299.00 19.50   1.69 1.36
##       JOB INNOV  BSER   STAT  CHOI
## Min -9.00  0.76  0.30  13.06 16.73
## Max 88.73 58.70 59.21 100.00 96.55
```

3.2 Valores erróneos

Revisar los valores de las variables numéricas. En primer lugar, si hay valores erróneos en alguna variable, se deben sustituir por NAs.

```
df[sapply(df,is.numeric)] <- lapply(df[sapply(df,is.numeric)], function(x){
  as.numeric(suppressWarnings(as.numeric(as.character(x))))
})
```

```
head(df,5)
```

```
##   code       economy year   OPP    PC  FAIL    EI   TEA   OWN EMPL MOT FMTEA
## 1    1 United States 2023 53.81 48.99 44.55 12.09 14.71  6.74   NA  NA  0.84
## 2   27  South Africa 2023 64.10 69.21 59.51  7.45 11.11  5.92   NA  NA  0.76
## 3   30        Greece 2023 45.29 53.76 53.16  9.10  6.74 14.70   NA  NA  0.79
## 4   31   Netherlands 2023 67.37 46.01 40.29 16.02 13.69  6.92   NA  NA  0.80
## 5   33        France 2023 50.86 49.51 40.08 13.41 10.75  4.56   NA  NA  0.72
##   FMOD   JOB INNOV  BSER  STAT  CHOI
## 1   NA 28.80    NA 20.42 78.94 79.21
## 2   NA 21.42    NA  3.57 85.51 78.46
## 3   NA 12.12    NA 22.53 69.76 72.90
## 4   NA 15.13    NA 23.65    NA    NA
## 5   NA 23.17    NA 32.58 51.76 65.41
```

4 Valores atípicos en variables cuantitativas

A modo ilustrativo, en este apartado revisaremos si existen valores atípicos en la variable TEA. En caso de observar valores atípicos en esta variable, se deben sustituir por NA. Justificad vuestra elección. En el siguiente apartado, se realizará la imputación de los NAs.

Para la siguiente tarea, utilizaremos el rango intercuartílico, para definir que son valores atípicos o 'outliers'

```
Q1 <- quantile(df$TEA,0.25)
Q3 <- quantile(df$TEA,0.75)
RIQ = IQR(df$TEA)

lower_outliers <- Q1 - 1.5 * RIQ
upper_outliers <- Q3 + 1.5 * RIQ
```

```
lower_outliers_values <- df$TEA[df$TEA < lower_outliers]
upper_outliers_values <- df$TEA[df$TEA > upper_outliers]
```

Valores anómalos por debajo

```
print(lower_outliers_values)
```

```
## numeric(0)
```

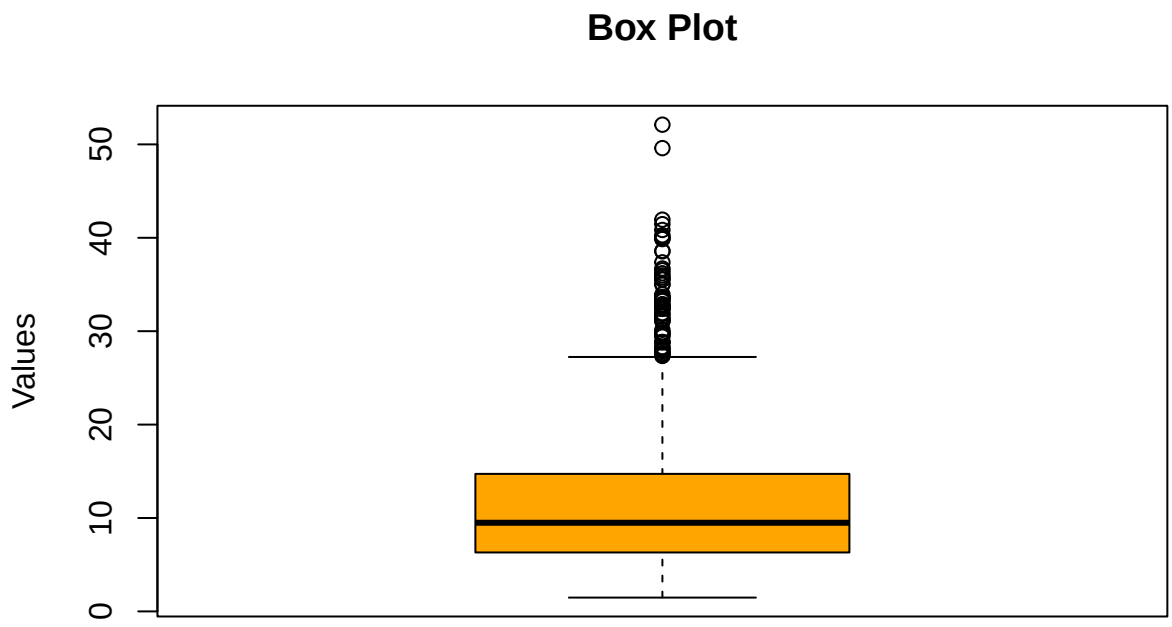
Valores anómalos por encima

```
print(upper_outliers_values)
```

```
##  [1] 31.05 32.39 31.31 32.65 27.96 29.42 27.87 29.89 33.58 28.26 41.94 31.10
## [13] 32.90 49.60 28.30 32.40 36.71 36.20 40.84 27.52 29.62 27.35 33.53 27.56
## [25] 28.83 31.83 38.55 29.75 33.23 33.56 30.15 28.81 37.37 35.53 32.79 27.40
## [37] 32.61 39.86 39.91 33.34 28.11 35.97 36.52 35.04 32.39 35.76 41.46 35.56
## [49] 27.66 34.99 33.95 31.94 31.29 32.63 38.60 52.11 33.67 29.82 40.08 40.27
## [61] 31.60 28.85
```

Además se realiza un diagrama de cajas y bigotes para comprobar que efectivamente, los valores atípicos obtenidos están por encima de Q3.

```
boxplot(df$TEA, main = "Box Plot", ylab = "Values", col = "orange", outline = TRUE)
```



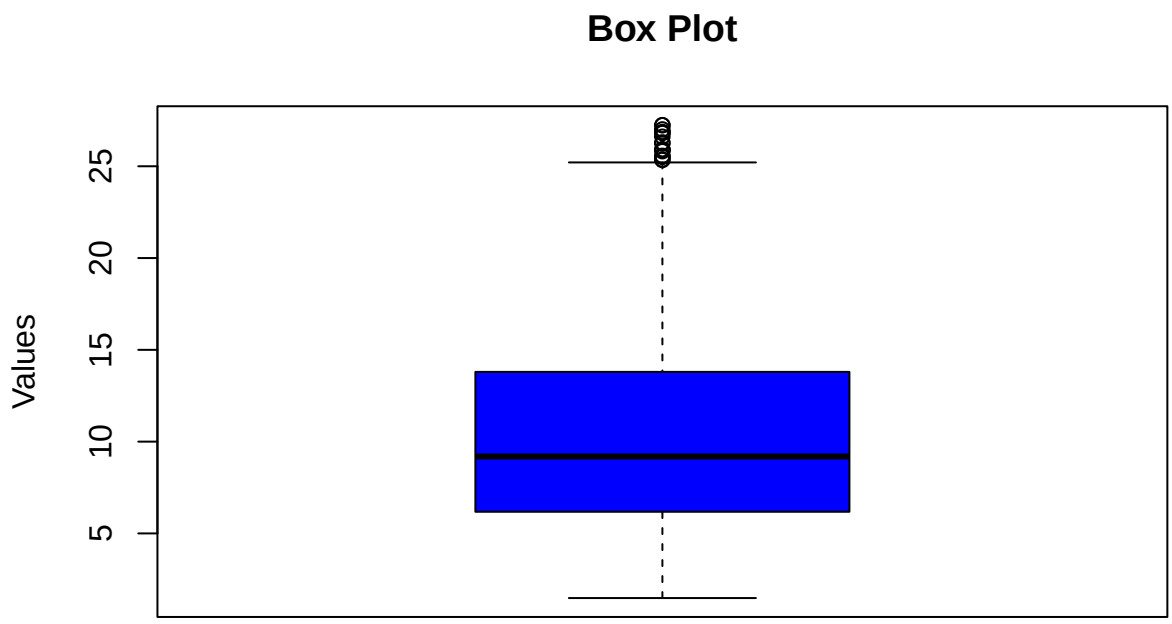
Para sustituir los valores atípicos por NA, realizamos una comparación entre los valores contenidos en la columna y en este caso solo los valores por encima de Q3

```
df$TEA[df$TEA > upper_outliers] <- NA
```

Esto no significa que ahora no hayan valores atípicos. Porque al haber sustituido los valores atípicos originalmente encontrados por NA, ahora los valores para los limites para el rango intercuartílico, cambian.

4.1 Diagrama de cajas y bigotes nuevo

```
boxplot(df$TEA, main = "Box Plot", ylab = "Values", col = "blue", outline = TRUE)
```



Estos nuevos valores atípicos están suficientemente cerca de Q3, así se pueden considerar como válidos. Para "eliminarlos", habría que igualar el parámetro de outline a FALSE. Si utilizamos de nuevo el algoritmo, se obtiene:

```
Q1_n <- quantile(df$TEA,0.25, na.rm=TRUE)
Q3_n <- quantile(df$TEA,0.75, na.rm=TRUE)
RIQ_n = IQR(df$TEA, na.rm = TRUE)

lower_outliers_n <- Q1 - 1.5 * RIQ
upper_outliers_n <- Q3 + 1.5 * RIQ

lower_outliers_values_n <- df$TEA[df$TEA < lower_outliers]
upper_outliers_values_n <- df$TEA[df$TEA > upper_outliers]
```

```
print(lower_outliers_values_n)
```

```
##  [1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
## [26] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
## [51] NA NA NA NA NA NA NA NA NA NA NA NA
```

```
print(upper_outliers_values_n)
```

```
##  [1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
## [26] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
## [51] NA NA NA NA NA NA NA NA NA NA NA NA
```

5 Imputacion

Realizad la imputación por vecinos más cercanos de la variable EI, usando el método de imputación basado en los k vecinos más cercanos. Utilizad k=5. Se deben elegir las variables que pueden ser más adecuadas para realizar esta imputación. Justificad vuestra elección. Al terminar la imputación, mostrad en una tabla los valores imputados.

Para seleccionar las variables, es necesario escoger aquellas que tengan una relación alta entre ellas con EI, para ello se realizará un análsis de correlación con la función cor

```
correlation_table <- cor(df[sapply(df,is.numeric)],use = "complete.obs")["EI",]

print(correlation_table)
```

```
##        code        year         OPP          PC        FAIL          EI
##  0.16518135  0.05252725  0.34510768  0.57678228 -0.15740559  1.00000000
##         TEA         OWN        EMPL         MOT       FMTEA        FMOD
##  0.63693176  0.12534847 -0.15371552 -0.22282327  0.03685573 -0.05287416
##         JOB       INNOV        BSER        STAT        CHOI
##  0.24937034 -0.05376341 -0.44035927  0.27027951  0.49468259
```

Se seleccionan las variables que tengan una correlación alta, en este caso se han escogido las variables PC, TEA, BSER y CHOI. Ahora se realiza la imputación de la variable EI, con el comando knn de la libreria VIM

```
suppressMessages(library(VIM))
df <- VIM::kNN(df, variable = "EI", k=5, dist_var = c("PC","TEA","BSER","CHOI"))
head(df,5)
```

```
##   code       economy year   OPP    PC  FAIL    EI   TEA   OWN EMPL MOT FMTEA
## 1    1 United States 2023 53.81 48.99 44.55 12.09 14.71  6.74   NA  NA  0.84
## 2   27  South Africa 2023 64.10 69.21 59.51  7.45 11.11  5.92   NA  NA  0.76
## 3   30        Greece 2023 45.29 53.76 53.16  9.10  6.74 14.70   NA  NA  0.79
## 4   31   Netherlands 2023 67.37 46.01 40.29 16.02 13.69  6.92   NA  NA  0.80
## 5   33        France 2023 50.86 49.51 40.08 13.41 10.75  4.56   NA  NA  0.72
##   FMOD   JOB INNOV  BSER  STAT  CHOI EI_imp
## 1   NA 28.80    NA 20.42 78.94 79.21  FALSE
## 2   NA 21.42    NA  3.57 85.51 78.46  FALSE
## 3   NA 12.12    NA 22.53 69.76 72.90  FALSE
## 4   NA 15.13    NA 23.65    NA    NA  FALSE
## 5   NA 23.17    NA 32.58 51.76 65.41  FALSE
```

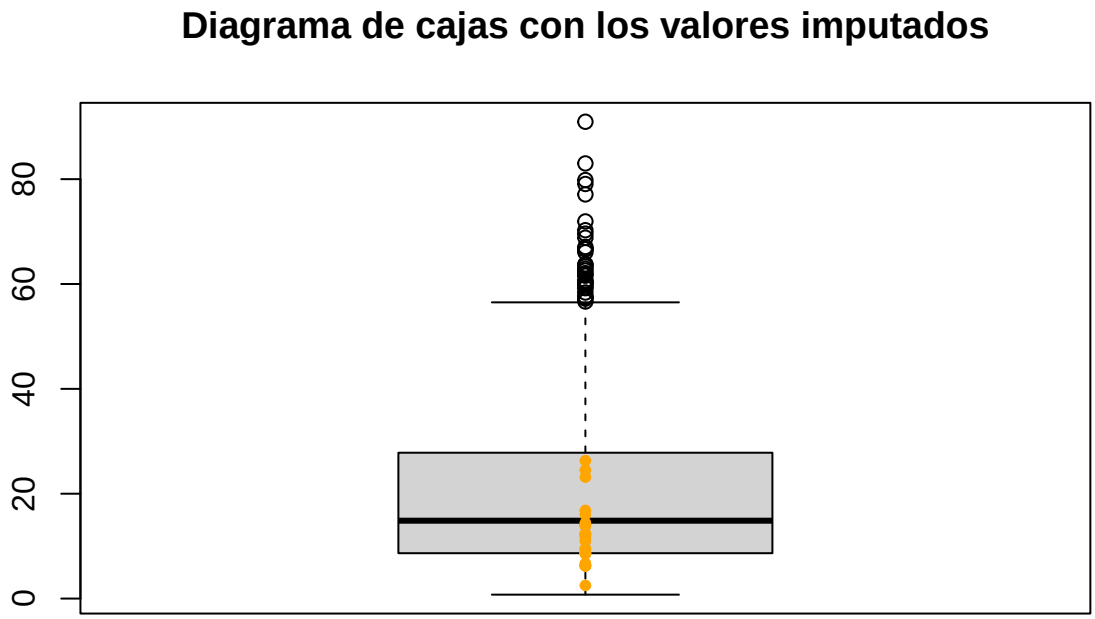
Se ha creado una nueva columna llamada EI_imp, con valores lógicos. Los valores TRUE, en esta columna quiere decir que su respectivo valor en la columna EI, ha sido imputado por el método.

```
imputed_values <- df[df$EI_imp == TRUE, c("EI")]

print(imputed_values)
```

```
##  [1] 12.35  6.24  6.24  8.55  8.55 12.17 11.64 13.93  9.52 10.96  8.73  6.72
## [13] 13.93  6.24 24.50 14.43 23.17 12.35 26.29  6.37  2.51 16.81 13.94 12.35
## [25]  9.40 16.02  8.73  6.24
```

```r
boxplot(df$EI, main =  "Diagrama de cajas con los valores imputados")
stripchart(imputed_values, vertical = TRUE,col ="orange", pch = 20, add = TRUE)
```



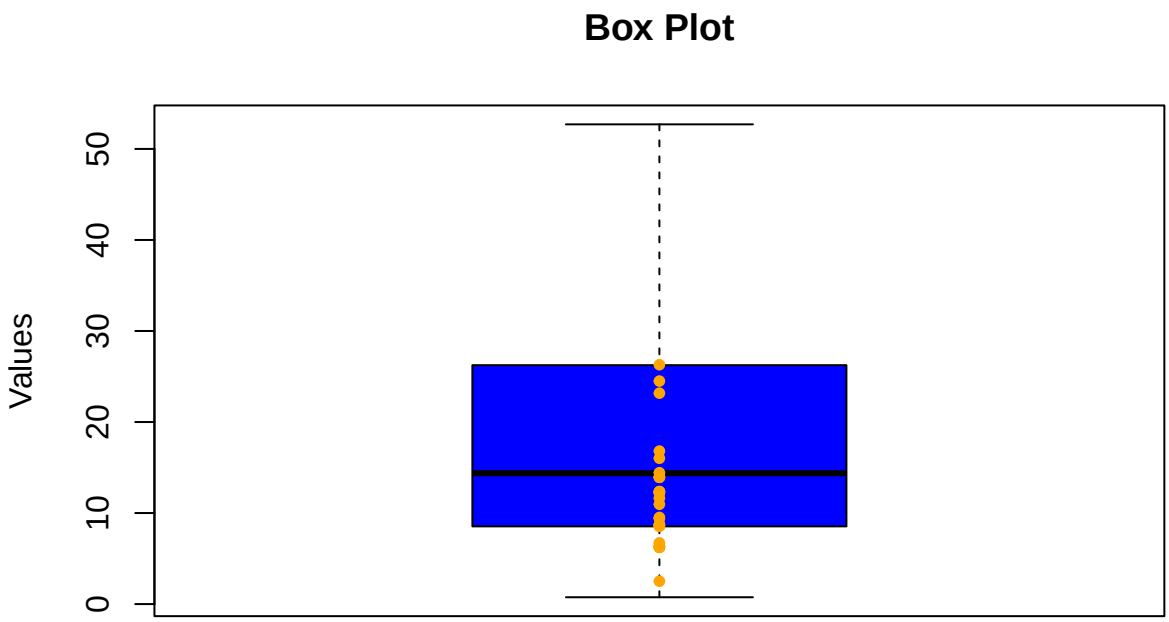
Para poder observar los valores imputados con mejor claridad, se va a proceder a aplicar el mismo algortimo que en el apartado anterior para eliminar los outliers

```r
Q1_i <- quantile(df$EI,0.25)
Q3_i <- quantile(df$EI,0.75)
RIQ_i <- IQR(df$EI)
```

```r
lower_outliers_i <- Q1_i - 1.5 * RIQ_i
upper_outliers_i <- Q3_i + 1.5 * RIQ_i
```

```r
df$EI[df$EI > upper_outliers_i] <- NA
df$EI[df$EI < lower_outliers_i] <- NA
```

```r
boxplot(df$EI, main = "Box Plot", ylab = "Values", col = "blue", outline = FALSE)
stripchart(imputed_values, vertical = TRUE,col ="orange", pch = 20, add = TRUE)
```



Ahora se puede apreciar mejor que los valores imputados están dentro del rango original

6 Análisis de correlaciones

En este apartado evaluaremos los datos que tienen relación con la percepción sobre el emprendimiento de la población no emprendedora (variables OPP, PC, FAIL, EI), la actividad emprendedora (TEA), el valor otorgado al emprendimiento (STAT) y el emprendimiento como opción de carrera (CHOI). En primer lugar, vamos a estudiar la correlación entre estas variables. Mostrad la matriz de correlación entre estas variables e interpretad el resultado.

```
correlation_table2 <- cor(df[,c("OPP", "PC", "FAIL", "EI", "TEA", "STAT",
                                "CHOI")], use="complete.obs")

print(correlation_table2)
```

```
##              OPP         PC         FAIL          EI        TEA       STAT
## OPP  1.000000000  0.5904418  0.009822408  0.41385949  0.4580931 0.42214633
## PC   0.590441773  1.0000000 -0.134342362  0.59045860  0.6093081 0.33524642
## FAIL 0.009822408 -0.1343424  1.000000000 -0.09359321 -0.0593581 0.05554004
## EI   0.413859491  0.5904586 -0.093593215  1.00000000  0.6761525 0.29050094
## TEA  0.458093125  0.6093081 -0.059358095  0.67615252  1.0000000 0.21224450
## STAT 0.422146333  0.3352464  0.055540042  0.29050094  0.2122445 1.00000000
## CHOI 0.414621633  0.5482364  0.036334273  0.51625669  0.4354789 0.44187276
##            CHOI
## OPP  0.41462163
## PC   0.54823645
## FAIL 0.03633427
```

```
## EI   0.51625669
## TEA  0.43547888
## STAT 0.44187276
## CHOI 1.00000000
```

La interpretación de los resultados depende del valor absoluto de los resultados, de manera que cuando mayor sean éstos, mas correlacionados estarán el signo '-', hace referencia a que los valores son inversamente proporcionales, si los valores son positivos, esto quiere decir al contrario, que los valores son proporcionales. Con esta explicación, se puede asegurar, que por ejemplo la variable OPP, está considerablemente correlacionado con las variables PC, EI, TEA, STAT y CHOI. Obviamente su correlación consigo misma es máxima, por eso el 1.

7 Análisis de componentes principales

Usando las mismas variables que en la sección anterior, aplicad análisis de componentes principales. Mostrad el resultado e interpretadlo. ¿Cuántas dimensiones se podrían utilizar para reducir el conjunto de datos? ¿Qué variables son más relevantes en estos componentes?

Para realizar el análisis de componentes principales primero es necesario saber que matriz, se va a escoger, en este caso, ya que la matriz de datos está formada por variables con la misma magnitud se va a usar la matriz de covarianza

Primero se limpia el subset de las variables OPP PC FAIL EI TEA STAT y CHOI, de valores ausentes, o NA.

Se escoge center = TRUE y scale.=FALSE para diagonalizar la matriz de covarianzas. Con ACP.cov$sdev se presenta la desviación estándar de cada componente principal que corresponde a la raíz cuadrada de los valores propios.

```
df_subset_clean = na.omit(df[,c("OPP", "PC", "FAIL", "EI", "TEA", "STAT", "CHOI")])
```

```
df.cov <- prcomp(x = df_subset_clean, center = TRUE, scale. = FALSE)

summary(df.cov)
```

```
## Importance of components:
##                             PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     24.1850 12.1821 11.7079 9.84191 8.56268 8.04093 3.69201
## Proportion of Variance  0.5228  0.1326  0.1225 0.08657 0.06553 0.05779 0.01218
## Cumulative Proportion   0.5228  0.6554  0.7779 0.86450 0.93003 0.98782 1.00000
```

La varianza de cada componente principal es el valor propio y se calcula elevando al cuadrado ACP.cov$sdev.

```
df.cov$sdev
```

```
## [1] 24.185027 12.182141 11.707941  9.841907  8.562679  8.040927  3.692005
```

```
var.exp.cov <-(df.cov$sdev^2 / sum(df.cov$sdev^2))*100
print("%  de varianza explicada en cada CP (debajo)")
```

```
## [1] "%  de varianza explicada en cada CP (debajo)"
```

```
print(var.exp.cov)
```

```
## [1] 52.277533 13.263836 12.251323  8.657260  6.553016  5.778754  1.218278
```

Los vectores propios están en el objeto df.cov$rotation.

```
df.cov$rotation
```

```
##              PC1         PC2         PC3         PC4         PC5        PC6
## OPP  -0.56449071 -0.68594025  0.30823743 -0.01970116 -0.07744652 -0.3285612
```

```
## PC   -0.50824233  0.22397898  0.24773374 -0.21008040 -0.24065607  0.7193945
## FAIL  0.02353955 -0.39673448 -0.64458495 -0.61546279  0.07443465  0.2046449
## EI   -0.37427564  0.41284179  0.02260685 -0.31895579  0.66116189 -0.3065684
## TEA  -0.15446082  0.09868011  0.05052981 -0.14586854  0.11297362 -0.0242677
## STAT -0.26454397 -0.18337303 -0.35653487  0.65239881  0.46977476  0.3474283
## CHOI -0.43423720  0.32881508 -0.54584474  0.16747838 -0.50986249 -0.3425785
##               PC7
## OPP   0.038565875
## PC    0.102827809
## FAIL  0.018530884
## EI    0.236888878
## TEA  -0.963961222
## STAT -0.047484093
## CHOI -0.001845024
```

El objeto df.cov$x contiene la proyección de todos los valores.

```
df.cov$x[1,]
```

```
##        PC1        PC2        PC3        PC4        PC5        PC6        PC7
## -13.541508  -9.947440 -12.610860   5.075378  -7.360975  -3.596125  -5.478035
```

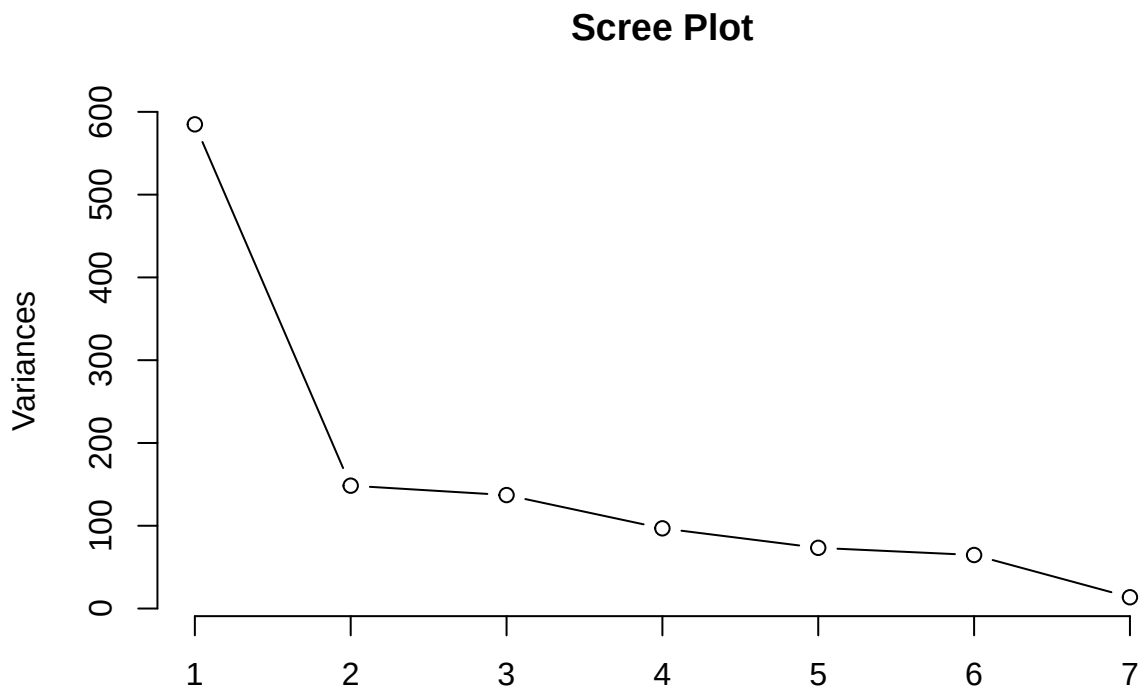
En el caso de realizar el ACP con la matriz de covarianzas, los valores propios son

```
df.cov$sdev^2
```

```
## [1] 584.91552 148.40455 137.07588  96.86313  73.31947  64.65651  13.63090
```

Aplicando el criterio de Kaiser-Gutman es suficiente con seleccionar la primera componente principal. Si se observa el gráfico el codo se forma con el primer valor propio y después es plano. Los criterios coinciden en el número de componentes principales a escoger. Es suficiente con seleccionar una dimensión.

```
screeplot(df.cov, type="lines", main="Scree Plot")
```

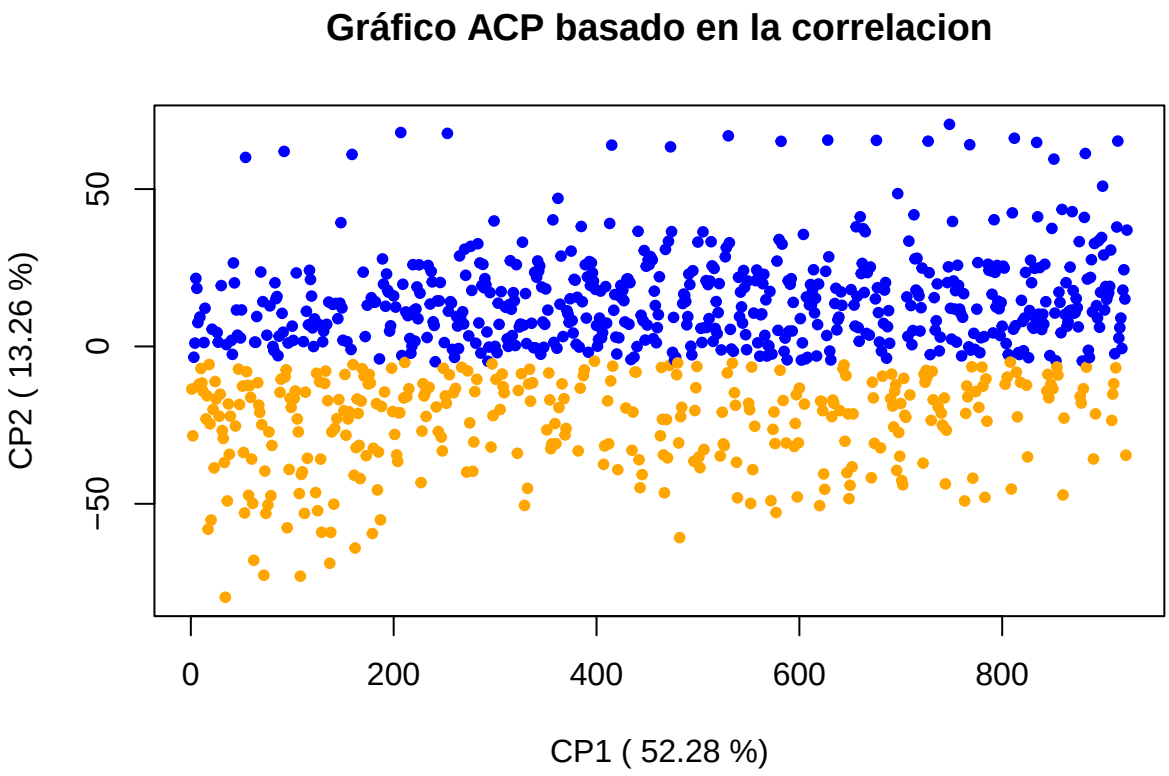


Se va aproceder a utilizar al menos 2 ya que la utilización de los primeros dos componentes capturaría la mayor parte de la estructura mientras se mantiene baja la dimensionalidad.

```
kmeans <- kmeans(df.cov$x, centers=2)
col_factor <- as.factor(kmeans$cluster)
```

Así que se pasa del espacio original de 7 dimensiones a una proyección de solo dos dimensiones con un porcentaje de varianza explicada mayor del 50%

```
plot(df.cov$x[,1],
     xlab = paste("CP1 (", round(var.exp.cov[1],2),"%)"),
     ylab = paste("CP2 (", round(var.exp.cov[2],2),"%)"),
     main = "Gráfico ACP basado en la correlacion",
     col = c("blue","orange")[col_factor],
     pch=20)
```



¿Cuántas dimensiones se podrían utilizar para reducir el conjunto de datos? Según la explicación anterior 2. ¿Qué variables son más relevantes en estos componentes?

```
df.cov$rotation[,1:2]
```

```
##              PC1         PC2
## OPP  -0.56449071 -0.68594025
## PC   -0.50824233  0.22397898
## FAIL  0.02353955 -0.39673448
## EI   -0.37427564  0.41284179
## TEA  -0.15446082  0.09868011
## STAT -0.26454397 -0.18337303
## CHOI -0.43423720  0.32881508
```

Las variables con las cargas más altas en valor absoluto son OPP PC CHOI y EI para PC1, ya que son las más influyentes para explicar la mayoría de la varianza. En el caso de PC2 OPP, EI, CHOI y FAIL

Para las dos a la vez, las variables OPP, CHOI, EI y PC que juntas explican aproximadamente +60% de la varianza

8. Archivo final

Una vez realizado el preprocesamiento sobre el archivo, guardad el resultado de los datos en un archivo csv llamado gem02.csv.

NOTA: EL DATAFRAME EXPORTADO, CONTIENE LAS MODIFICACIONES ESPECIFICADOS EN LOS APARTADOS, QUIERE DECIR, QUE POR EJEMPLO LAS IMPUTACIONES SOLO SE HAN REALIZADO PARA LA VARIABLE EI, QUE ES LA QUE ESPECIFICA EL ENUNCIADO, DE MODO,

QUE PARA LAS DEMÁS VARIABLES DICHO EJERCICIO NO SE HA EJECUTADO.

```
ruta <- file.path(
  "C:/Users/alexs/Documents/PCbp/Master-DataScience",
  "Analisis estadístico/PEC1",
  "gem02.csv"
)
```

```
write.csv(df, file = ruta , row.names = FALSE)
```