SPORTS DATA CAMPUS

MÁSTER EN BIG DATA APLICADO AL SCOUTING EN FÚTBOL
Análisis exploratorio y de componentes principales

SPORTS DATA CAMPUS
UCAM UNIVERSIDAD CATÓLICA DE MURCIA
ENIT INNOVA IT BUSINESS SCHOOL
BIG DATA International Campus

Índice

1. Análisis del código

```r
# ----------------------------------------
# 1. Librerías
# ----------------------------------------
# install.packages("FactoMineR")
# install.packages("GGally")
# install.packages("corrplot")
# install.packages("factoextra")
# install.packages("caret")

library(dplyr)
library(ggplot2)
library(GGally)
library(corrplot)
library(FactoMineR)
library(factoextra)
library(caret)
```

En primer lugar, instalé las librerías necesarias e importé los paquetes requeridos para el análisis. Dejé las líneas de instalación comentadas con el fin de evitar su reinstalación ejecuciones posteriores.

```r
# ----------------------------------------
# 2. Lectura y limpieza del dataset
# ----------------------------------------

getwd()

setwd("C:/Users/pepec/Documents/Master/Premaster/PM-Estadística/Modulo2")


# Leer el CSV (ajustar el path si es necesario)
df <- read.csv("FBREF_players.csv", sep = ";")

# Filtro de defensas en La Liga con mínimo 684 minutos (20% de 38 partidos)
equipos_laliga <- c("Alavés", "Athletic Club", "Atlético Madrid", "Barcelona", "Betis",
                    "Cádiz", "Celta Vigo", "Eibar", "Elche", "Getafe", "Granada",
                    "Huesca", "Levante", "Osasuna", "Real Madrid", "Real Sociedad",
                    "Sevilla", "Valencia", "Valladolid", "Villarreal")

df_defensas <- df %>%
  filter(grepl("DF", Pos),
         Squad %in% equipos_laliga,
         Min >= 684)
```

A continuación, establecí el directorio de trabajo y procedí a leer los ficheros disponibles. Entre ellos, seleccioné el archivo .csv que contenía diversas métricas de rendimiento de jugadores pertenecientes a las cinco grandes ligas europeas.

Posteriormente, filtré los datos para quedarme únicamente con los equipos de LaLiga, y creé un nuevo *dataframe* denominado df_defensas, que contenía exclusivamente a los jugadores que actuaban en posiciones defensivas.

```r
# ----------------------------------------
# 3. Selección de métricas
# ----------------------------------------

metricas <- c("Int.90", "Blocks.90", "Recov.90", "AerialW.90",
              "PassesCompleted.90", "KP.90", "PPA.90")

df_metricas <- df_defensas %>%
  select(all_of(metricas)) %>%
  na.omit()

# ----------------------------------------
```

Seleccioné las métricas de intercepciones, bloqueos, recuperaciones, duelos aéreos ganados, pases completados, pases clave y pases al área por 90 minutos, al considerarlas variables representativas del rendimiento defensivo.

```r

# ----------------------------------------
# 4. Análisis exploratorio de los datos
# ----------------------------------------

# Estadísticas descriptivas
summary(df_metricas)

# Pairplot para ver relaciones
ggpairs(df_metricas)

# Matriz de correlación
cor_matrix <- cor(df_metricas)
corrplot(cor_matrix, method = "color", addCoef.col = "black")

```

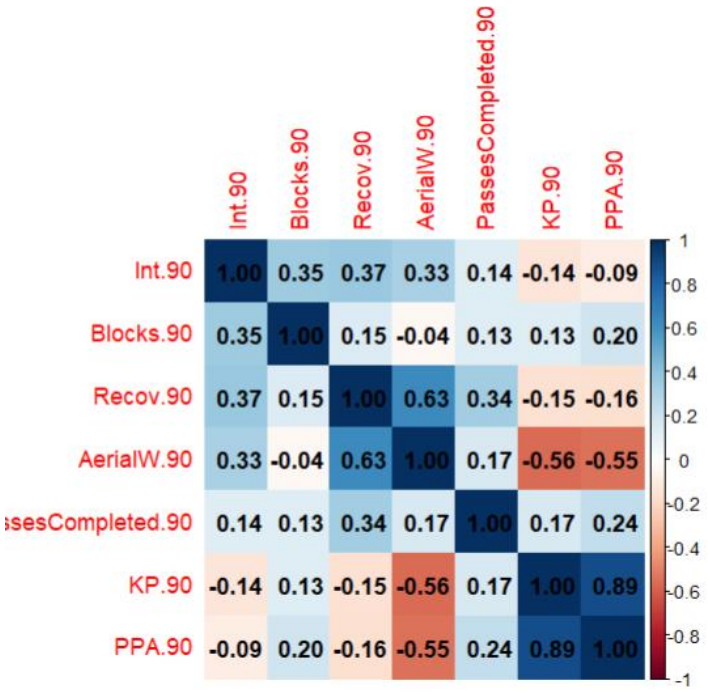
Realicé un análisis exploratorio de datos (EDA) utilizando la función *summary()* para examinar la dispersión de cada variable. Complementé el análisis con la función *ggpairs()* para visualizar las relaciones entre métricas y *corrplot()* para representar la matriz de correlaciones y detectar posibles redundancias entre variables.

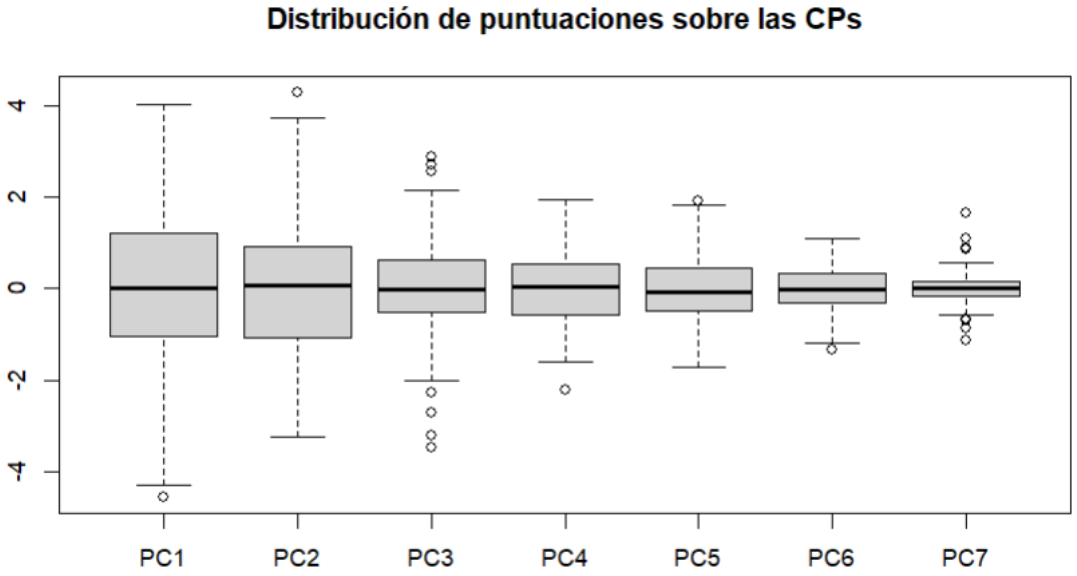
```r
# ---------------------------------------
# 5. Análisis de componentes principales
# ---------------------------------------

# Escalar las variables (normalización Min-Max)
preproc <- preProcess(df_metricas, method = c("range"))
df_normalizado <- predict(preproc, df_metricas)

# Aplicar PCA
acp <- prcomp(df_normalizado, center = TRUE, scale. = TRUE)

# Resumen de varianza explicada
summary(acp)

# Scree plot (varianza por componente)
fviz_eig(acp, addlabels = TRUE, ylim = c(0, 60))

# Correlación entre variables originales y componentes
fviz_pca_var(acp, col.var = "contrib", repel = TRUE)

# Puntuaciones de los jugadores sobre los dos primeros componentes
fviz_pca_ind(acp,
             geom.ind = "point",
             pointshape = 21,
             col.ind = "cos2",
             palette = "viridis",
             addEllipses = FALSE,
             repel = TRUE)

# Boxplot de las puntuaciones
scores <- as.data.frame(acp$x)
boxplot(scores, main = "Distribución de puntuaciones sobre las CPs")
```

```r
# ---------------------------------------
# 6. Rating
# ---------------------------------------

df_resultado <- df_defensas %>%
  filter(complete.cases(select(., all_of(metricas)))) %>%
  mutate(PC1 = scores$PC1,
         PC2 = scores$PC2)

head(df_resultado[, c("Player", "Squad", "PC1", "PC2")])
```

Finalmente, normalicé todas las variables mediante la técnica Min-Max Scaling, con el objetivo de eliminar el efecto de las diferentes escalas y asegurar comparaciones consistentes entre métricas. Sobre este conjunto de datos estandarizados apliqué el Análisis de Componentes Principales (PCA) a través de la función *prcomp()*.

El resultado mostró que las dos primeras componentes principales explicaban aproximadamente el 65 % de la varianza total del dataset, lo que permitió reducir la dimensionalidad sin pérdida significativa de información. Por último, representé de forma gráfica tanto las variables como los jugadores en el nuevo espacio reducido, facilitando la interpretación visual de los patrones y relaciones existentes entre ellos.


Int.90
Blocks.90
Recov.90
AerialW.90
PassesCompleted.90
KP.90
PPA.90
1.00 0.35 0.37 0.33 0.14 -0.14 -0.09
0.35 1.00 0.15 -0.04 0.13 0.13 0.20
0.37 0.15 1.00 0.63 0.34 -0.15 -0.16
0.33 -0.04 0.63 1.00 0.17 -0.56 -0.55
0.14 0.13 0.34 0.17 1.00 0.17 0.24
-0.14 0.13 -0.15 -0.56 0.17 1.00 0.89
-0.09 0.20 -0.16 -0.55 0.24 0.89 1.00
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1


Distribución de puntuaciones sobre las CPs
4
2
0
-2
-4
PC1
PC2
PC3
PC4
PC5
PC6
PC7