Análisis exploratorio y de componentes principales

Índice

Análisis del código

1. Librerías

```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)
```

Primeramente, instalé las librerías necesarias y tras ello las importo. Las he dejado comentadas para que no se vuelvan a instalar tras ejecutar el código.

2. Lectura del fichero

```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)
```

En primer lugar, seteamos ese path como nuestro directorio actual. Ahí, contenemos todos los archivos correspondientes al módulo 2 del Premáster Curso en Estadística y Matemáticas.

A continuación, leemos los archivos del directorio previamente establecido y leemos el archivo .csv que contiene diversas métricas de los jugadores de las 5 grandes ligas.

Selecciono los equipos de la liga española y después creo el dataframe df_defensas que filtra por la posición de defensa.

3. Selección de métricas

```
# ---------------------------------------
# 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()

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

Elegimos las métricas de intercepciones, bloqueos, recuperaciones, duelos aéreos ganados, pases completados, pases clave y pases al área por 90 minutos.

4. Análisis Exploratorio de Datos (EDA)

```
# ---------------------------------------
# 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")

```

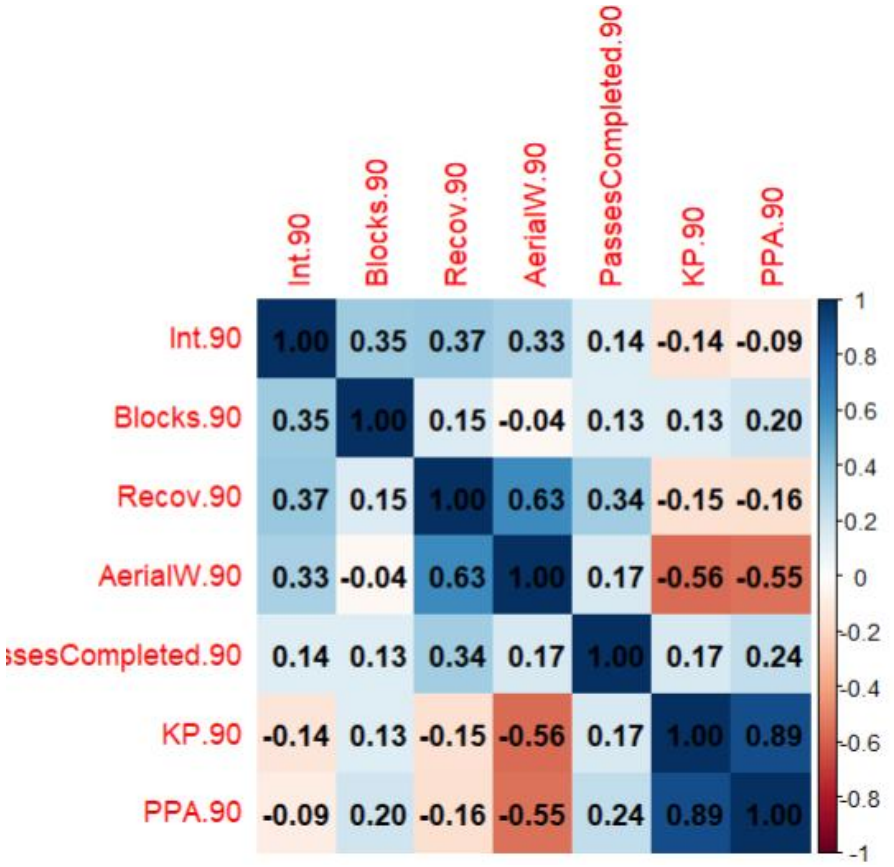
Realizamos un análisis básico con la función summary() para ver la dispersión de cada variable. A continuación, usamos ggpairs() para ver las relaciones entre métricas y corrplot() para mostrar la matriz de correlaciones.

5. Análisis de Componentes Principales (ACP)

```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")])
```

Normalizamos las variables utilizando Min-Max para evitar que las escalas distintas influyeran en los resultados. Después, aplicamos prcomp() para obtener las componentes principales. Observamos que los dos primeros explican aproximadamente el 65% de la varianza total del dataset. Representamos gráficamente las variables y los jugadores en el nuevo espacio reducido.



Distribución de puntuaciones sobre las CPs