PRA2-Tipologia

David Lucas, Francisco Javier Cantero

25/5/2022

Contents

1 Descripción del dataset

Para llevar a cabo el desarrollo de esta práctica, se utilizará el dataset obtenido en la Práctica 1. Este dataset contiene datos de cada uno de los jugadores que hayan interactuado con la NBA durante un periodo de tiempo específico (última temporada) y recoge todas las estadísticas acumuladas. Adicionalmente, el dataset incluye la variable PER (player efficiency rating) que permite evaluar la eficiencia del jugador con respecto a las estadísticas totales de la temporada. Mediante esta variable, se podrá conocer cómo ha sido el rendimiento del jugador y poder comparar con el resto de jugadores de una manera rápida y sencilla. Por lo tanto, el dataset incluye un total de 19 campos:

- Nombre: Indica el nombre del jugador.
- Equipo: Indica el nombre del equipo al que pertenece el jugador.
- MP: (Minutes played) Indica los minutos que ha jugado en total un jugador.
- FG: (Field goals) Indica los tiros de campo anotados (sin contar los tiros libres).
- FGA: (Field goal attempts) Indica los tiros de campo realizados (sin contar los tiros libres).
- FG3: (3-point field goals) Indica los tiros de tres puntos anotados por el jugador.
- FG3A: (3-point field goal attempts) Indica los tiros de tres puntos realizados por el jugador.
- FT: (Free throws) Indica los tiros libres anotados por el jugador.
- FTA: (Free throw attempts) Indica los tiros libres realizados por el jugador.
- ORB: (Offensive rebounds) Indica los rebotes ofensivos capturados por un jugador.
- DRB: (Defensive rebounds) Indica los rebotes defensivos capturados por un jugador.
- AST: (Assists) Indica las asistencias realizadas por un jugador.

- STL: (Steals) Indica los robos realizados por jugador.
- BLK: (Blocks) Indica los tapones realizados por el jugador.
- TOV: (Turnovers) Indica las pérdidas realizadas por el jugador.
- PF: (Personal fouls) Indica las faltas personales realizadas por el jugador.
- PTS: (Points) Indica el total de puntos anotados por el jugador.
- G: (Games) Indica los partidos totales jugados por el jugador.
- PER: (Player efficiency rating) Detalla la eficiencia del jugador haciendo uso de las estadísticas anteriores.

Para finalizar, este conjunto de datos es importanto, ya que representa la volumetría general de diferentes apartado estadísticos durante la temporada 2020-2021 lo que nos permitirá poder conocer en mayor profundidad los jugadores que componen la liga.

2 Integración y selección de los datos de interés a analizar

Para comenzar con el tratamiento de los datos, se realiza la carga del dataset creado en la primera práctica de la asignatura:

```
playerstats <- read.csv("NBAPlayerStatistics_20-21.csv", stringsAsFactors = FALSE)
head(playerstats,3)
```

```
##          Nombre         Equipo   MP  FG FGA FG3 FG3A FT FTA ORB DRB AST STL BLK
## 1  Aaron Gordon Denver Nuggets  650 103 206  17   64 31  44  38  80  56  17  14
## 2  Aaron Gordon  Orlando Magic  737 128 293  42  112 66 105  39 127 105  16  20
## 3 Aaron Holiday Indiana Pacers 1176 170 436  67  182 68  83  15  74 123  46  13
##   TOV PF PTS  G   PER
## 1  30 40 254 25 16.39
## 2  67 49 364 25 17.97
## 3  66 94 475 66 12.15
```

Una vez cargada la información, mostramos la cabecera de este. En los primero registros del conjunto de datos podemos observar que el jugador Aaron Gordon se encuentra dos veces debido a que el jugador fue traspasado de un equipo a otro durante el transcurso de la temporada. Ante esta situación, se ha decidido matener los dos registros ya que la información que refleja cada uno de estos se ve influida por el contexto del equipo en el que jugó el jugador.

Además del conjunto de datos creado en la prácitca anterior, se ha decidido incluir a este información adicional proveniente del conjunto de datos Nba 2020-2021 Season Player Stats del repositorio de Kaggle (https://www.kaggle.com/datasets/umutalpaydn/nba-20202021-season-player-stats?resource=download). Por lo tanto, se pasa a realizar la carga de este conjunto de datos adicional:

```
playerstats2 <- read.csv("nba2021_advanced.csv", stringsAsFactors = FALSE)
summary(playerstats2)
```

```
##     Player              Pos                 Age             Tm
##  Length:497         Length:497         Min.   :19.00   Length:497
##  Class :character   Class :character   1st Qu.:22.00   Class :character
##  Mode  :character   Mode  :character   Median :25.00   Mode  :character
##                                        Mean   :25.62
##                                        3rd Qu.:28.00
##                                        Max.   :37.00
##        G               MP              PER               TS.
##  Min.   : 1.00   Min.   :   2.0   Min.   :-40.90   Min.   :0.0000
##  1st Qu.:12.00   1st Qu.: 130.0   1st Qu.:  9.30   1st Qu.:0.5000
##  Median :20.00   Median : 419.0   Median : 12.60   Median :0.5560
##  Mean   :18.46   Mean   : 417.9   Mean   : 12.65   Mean   :0.5386
##  3rd Qu.:26.00   3rd Qu.: 667.0   3rd Qu.: 16.70   3rd Qu.:0.6070
```

```
##  Max.   :30.00   Max.   :1101.0   Max.   : 38.70   Max.   :1.5000
##      X3PAr             FTr              ORB.             DRB.
##  Min.   :0.0000   Min.   :0.0000   Min.   : 0.000   Min.   : 0.00
##  1st Qu.:0.2530   1st Qu.:0.1420   1st Qu.: 1.800   1st Qu.:10.80
##  Median :0.4180   Median :0.2260   Median : 3.200   Median :14.20
##  Mean   :0.4052   Mean   :0.2743   Mean   : 4.845   Mean   :15.41
##  3rd Qu.:0.5630   3rd Qu.:0.3180   3rd Qu.: 6.600   3rd Qu.:19.10
##  Max.   :1.0000   Max.   :2.6670   Max.   :35.000   Max.   :54.10
##       TRB.             AST.             STL.             BLK.
##  Min.   : 0.00   Min.   : 0.00   Min.   :0.000   Min.   : 0.000
##  1st Qu.: 6.60   1st Qu.: 7.10   1st Qu.:0.900   1st Qu.: 0.600
##  Median : 9.10   Median :10.70   Median :1.400   Median : 1.400
##  Mean   :10.13   Mean   :13.48   Mean   :1.429   Mean   : 2.055
##  3rd Qu.:12.70   3rd Qu.:17.80   3rd Qu.:1.900   3rd Qu.: 2.800
##  Max.   :28.00   Max.   :47.50   Max.   :7.500   Max.   :21.300
##       TOV.             USG.             OWS              DWS
##  Min.   :  0.00   Min.   : 2.00   Min.   :-1.2000   Min.   :-0.100
##  1st Qu.:  9.30   1st Qu.:14.50   1st Qu.: 0.0000   1st Qu.: 0.100
##  Median : 12.00   Median :18.00   Median : 0.2000   Median : 0.300
##  Mean   : 13.47   Mean   :18.75   Mean   : 0.4606   Mean   : 0.426
##  3rd Qu.: 15.70   3rd Qu.:22.30   3rd Qu.: 0.7000   3rd Qu.: 0.700
##  Max.   :100.00   Max.   :41.30   Max.   : 4.9000   Max.   : 2.100
##        WS               WS.48             OBPM              DBPM
##  Min.   :-0.8000   Min.   :-1.1280   Min.   :-40.100   Min.   :-9.2000
##  1st Qu.: 0.1000   1st Qu.: 0.0300   1st Qu.: -3.200   1st Qu.:-1.5000
##  Median : 0.6000   Median : 0.0830   Median : -1.200   Median :-0.5000
##  Mean   : 0.8837   Mean   : 0.0676   Mean   : -1.587   Mean   :-0.5386
##  3rd Qu.: 1.3000   3rd Qu.: 0.1290   3rd Qu.:  0.800   3rd Qu.: 0.4000
##  Max.   : 6.3000   Max.   : 0.3420   Max.   : 14.100   Max.   : 7.1000
##       BPM               VORP
##  Min.   :-47.100   Min.   :-0.7000
##  1st Qu.: -3.800   1st Qu.:-0.1000
##  Median : -1.600   Median : 0.0000
##  Mean   : -2.128   Mean   : 0.1851
##  3rd Qu.:  0.500   3rd Qu.: 0.3000
##  Max.   : 18.000   Max.   : 3.4000
```

De este conjunto de datos, se ha decidido seleccionar las variables de posición (POS) y edad (Age) ya que estas enriquecen el conjunto de datos creado en la primera práctica de la asignatura. Por lo tanto, se pasa a realizar la asignación a cada jugador de su posición y edad durante la temporada 2020-2021:

```r
# playerstats["Age"] <- playerstats2$Age[playerstats2$Player == playerstats$Nombre]
playerstats["Age"] <- NULL
playerstats["Pos"] <- NULL
for (i in 1:length(playerstats$Nombre)){
  for (j in 1:length(playerstats2$Player))
    if(playerstats$Nombre[i] == playerstats2$Player[j]){
      playerstats$Age[i] <- playerstats2$Age[j]
      playerstats$Pos[i] <- playerstats2$Pos[j]
    }
}
playerstats <- playerstats[,c(1,2,20,21,3:19)]
```

Una vez asignadas la posición y la edad a cada jugador, se pasa a realizar la reducción de variables. Analizando en conjunto de datos, podemos observar que en este se almacenan los volumenes de tiro de los jugadores,

como por ejemplo, FG3 y FG3A que muestran el volumen de tiros de tres puntos anotados y lanzados. Es por esto que se ha decido reducir estas dos variables a una sola mediante el cálculo del porcentaje de tiro de tres puntos. Por lo tanto, esta reducción se realizará con respecto a los tiros de tres puntos, tiros de dos puntos y tiros libres:

```
playerstats["FG2p"] <- round((playerstats["FG"] - playerstats["FG3"]) / (playerstats["FGA"] - playersta
playerstats["FG3p"] <- round(playerstats["FG3"] / playerstats["FG3A"], 2)
playerstats["FTp"] <- round(playerstats["FT"] / playerstats["FTA"], 2)
```

Para finalizar con la construcción del conjunto de datos final, se pasa a reordenar las variables para que estas mantengan el orden visual en el conjunto de datos:

```
playerstats <- playerstats[,c(1:5,20,22:24,12:19,21)]
```

Para finalizar con este apartado, se muestra el resumen del conjunto de datos y la cabecera de este:

```
# Resumen del conjunto de datos
summary(playerstats)
```

```
##     Nombre             Equipo               Age           Pos
##  Length:626         Length:626         Min.   :19.0   Length:626
##  Class :character   Class :character   1st Qu.:23.0   Class :character
##  Mode  :character   Mode  :character   Median :25.0   Mode  :character
##                                        Mean   :25.7
##                                        3rd Qu.:28.0
##                                        Max.   :37.0
##
##        MP               G              FG2p             FG3p
##  Min.   :   3.0   Min.   : 1.00   Min.   :0.0000   Min.   :0.0000
##  1st Qu.: 218.2   1st Qu.:18.00   1st Qu.:0.4600   1st Qu.:0.2700
##  Median : 673.5   Median :36.00   Median :0.5100   Median :0.3400
##  Mean   : 833.1   Mean   :36.83   Mean   :0.5066   Mean   :0.3134
##  3rd Qu.:1347.2   3rd Qu.:57.75   3rd Qu.:0.5800   3rd Qu.:0.3900
##  Max.   :2666.0   Max.   :72.00   Max.   :1.0000   Max.   :1.0000
##                                   NA's   :6        NA's   :33
##       FTp              ORB              DRB              AST
##  Min.   :0.0000   Min.   :  0.00   Min.   :  0.00   Min.   :  0.00
##  1st Qu.:0.6800   1st Qu.:  6.00   1st Qu.: 26.25   1st Qu.: 13.00
##  Median :0.7800   Median : 20.00   Median : 81.50   Median : 49.00
##  Mean   :0.7512   Mean   : 33.92   Mean   :118.94   Mean   : 85.59
##  3rd Qu.:0.8500   3rd Qu.: 43.00   3rd Qu.:183.00   3rd Qu.:108.00
##  Max.   :1.0000   Max.   :297.00   Max.   :720.00   Max.   :763.00
##  NA's   :28
##       STL              BLK              TOV              PF
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.00   Min.   :  0.00
##  1st Qu.:  6.00   1st Qu.:  3.00   1st Qu.:  9.00   1st Qu.: 20.00
##  Median : 19.00   Median :  9.00   Median : 31.00   Median : 55.50
##  Mean   : 26.13   Mean   : 16.81   Mean   : 45.64   Mean   : 66.56
##  3rd Qu.: 40.00   3rd Qu.: 22.00   3rd Qu.: 65.00   3rd Qu.:109.00
##  Max.   :128.00   Max.   :190.00   Max.   :312.00   Max.   :237.00
##
##       PTS              PER
##  Min.   :   0.0   Min.   :-44.86
##  1st Qu.:  73.0   1st Qu.: 11.72
##  Median : 253.5   Median : 15.21
##  Mean   : 386.8   Mean   : 15.31
```

```
##  3rd Qu.: 591.5   3rd Qu.: 19.24
##  Max.   :2015.0   Max.   : 62.18
##
```

```
# Cabecera del conjunto de datos
head(playerstats,3)
```

```
##          Nombre         Equipo Age Pos   MP  G FG2p FG3p  FTp ORB DRB AST STL
## 1  Aaron Gordon Denver Nuggets  25  PF  650 25 0.61 0.27 0.70  38  80  56  17
## 2  Aaron Gordon  Orlando Magic  25  PF  737 25 0.48 0.38 0.63  39 127 105  16
## 3 Aaron Holiday Indiana Pacers  24  PG 1176 66 0.41 0.37 0.82  15  74 123  46
##   BLK TOV PF PTS   PER
## 1  14  30 40 254 16.39
## 2  20  67 49 364 17.97
## 3  13  66 94 475 12.15
```

3 Limpieza de los datos

3.1 Ceros y elementos vacíos

Se pasa a continuación a realizar la comprobación de valores nulos y vacíos dentro del conjunto de datos. En primer lugar, se comprueba la existencia de valores NaN dentro de este:

```
colSums(is.na(playerstats))
```

```
## Nombre Equipo    Age    Pos     MP      G   FG2p   FG3p    FTp    ORB    DRB
##      0      0      0      0      0      0      6     33     28      0      0
##    AST    STL    BLK    TOV     PF    PTS    PER
##      0      0      0      0      0      0      0
```

Como podemos observar, en las variables creadas anteriormente, que reflejan los porcentajes de tiro, estas contienen valores NaN producto de los cálculos realizados anteriormente. Para su corrección, se va a asignar a estos valores el valor de 0.00 debido a que es al que hacen referencia:

```
playerstats$`FG2p`[is.na(playerstats$`FG2p`)] <- 0.00
playerstats$`FG3p`[is.na(playerstats$`FG3p`)] <- 0.00
playerstats$`FTp`[is.na(playerstats$`FTp`)] <- 0.00
# Comprobación de que ya no hay valores NaN
colSums(is.na(playerstats))
```

```
## Nombre Equipo    Age    Pos     MP      G   FG2p   FG3p    FTp    ORB    DRB
##      0      0      0      0      0      0      0      0      0      0      0
##    AST    STL    BLK    TOV     PF    PTS    PER
##      0      0      0      0      0      0      0
```

Como se puede observar, estos valores han sido eliminado del conjunto de datos. Por último, se pasa a realizar la comprobación de valores vacíos dentro de las variables:

```
colSums(playerstats==""|playerstats==" ")
```

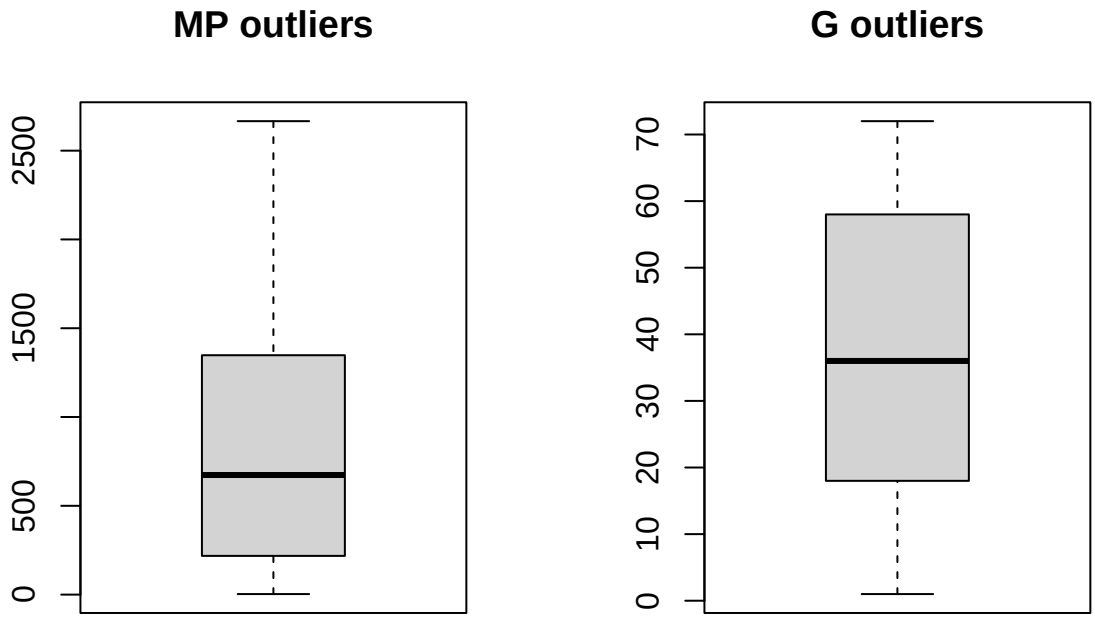
```
## Nombre Equipo    Age    Pos     MP      G   FG2p   FG3p    FTp    ORB    DRB
##      0      0      0      0      0      0      0      0      0      0      0
##    AST    STL    BLK    TOV     PF    PTS    PER
##      0      0      0      0      0      0      0
```

Con respecto a los valores vacíos, se puede observar que no existen. Por lo tanto, no es necesario realizar ninguna acción para su corrección.

3.2 Identificación y gestión de valores extremos

Se pasa a continuación a realizar la detección de outliers dentro del conjunto de datos. Para ello, únicamente se va a realizar esta detección sobre las variables que representan los minutos jugados (MP) y los partidos (G). El motivo de detectar únicamente en estas dos variables si existen outliers se debe a que por un lado podremos analizar si existen jugadores que han jugados pocos minutos o partidos en la liga con respecto al resto de jugadores, y en caso de existir, podremos eliminarlos para que no afecten a los análisis que se realizaran posteriormente. Dicho esto, se pasa a visualizar los diagramas de cajas de las variables anteriormente mencionadas:

```
# Diagramas de caja
par(mfrow = c(1, 2))
box_MP <- boxplot(playerstats$MP,main="MP outliers")
box_G <- boxplot(playerstats$G,main="G outliers")
```



Como se puede observar, en ambos diagramas no existen outliers. Sin embargo, estos diagramas son útiles ya que a traves del primer percentil se puede realizar el filtro de los jugadores que han jugado pocos minutos o partidos. Por lo tanto, a partir de este se pasa a realizar la eliminación de los jugadores que en la variables MP y G, se encuentran por debajo de este percentil:

```
# Percential 25 de ambas variables
box_MP$stats[2]
```

```
## [1] 218
```

```
box_G$stats[2]
```

```
## [1] 18
```

```
# Eliminación de registros por debajo de este percentil
playerstats <- playerstats[playerstats$MP > box_MP$stats[2],]
playerstats <- playerstats[playerstats$G > box_G$stats[2],]
```

Una vez hecho esto, el conjunto de datos obtenido y con el que se va a pasar a realizar los análisis contiene la siguiente información:

```
# Resumen del conjunto de datos
summary(playerstats)
```

```
##     Nombre             Equipo               Age            Pos
##  Length:437         Length:437         Min.   :19.00   Length:437
##  Class :character   Class :character   1st Qu.:23.00   Class :character
##  Mode  :character   Mode  :character   Median :25.00   Mode  :character
##                                        Mean   :25.99
##                                        3rd Qu.:29.00
##                                        Max.   :37.00
##        MP              G              FG2p             FG3p
##  Min.   : 228   Min.   :19.00   Min.   :0.1900   Min.   :0.0000
##  1st Qu.: 624   1st Qu.:33.00   1st Qu.:0.4700   1st Qu.:0.3000
##  Median :1066   Median :50.00   Median :0.5100   Median :0.3500
##  Mean   :1140   Mean   :47.82   Mean   :0.5222   Mean   :0.3264
##  3rd Qu.:1606   3rd Qu.:62.00   3rd Qu.:0.5700   3rd Qu.:0.3900
##  Max.   :2666   Max.   :72.00   Max.   :0.9000   Max.   :1.0000
##       FTp              ORB              DRB             AST
##  Min.   :0.4400   Min.   :  3.00   Min.   : 13   Min.   :  4.0
##  1st Qu.:0.6900   1st Qu.: 17.00   1st Qu.: 76   1st Qu.: 44.0
##  Median :0.7800   Median : 31.00   Median :139   Median : 81.0
##  Mean   :0.7627   Mean   : 46.24   Mean   :163   Mean   :118.1
##  3rd Qu.:0.8500   3rd Qu.: 58.00   3rd Qu.:219   3rd Qu.:148.0
##  Max.   :1.0000   Max.   :297.00   Max.   :720   Max.   :763.0
##       STL              BLK              TOV              PF
##  Min.   :  2.00   Min.   :  0.00   Min.   :  5.0   Min.   : 11.00
##  1st Qu.: 17.00   1st Qu.:  8.00   1st Qu.: 28.0   1st Qu.: 51.00
##  Median : 31.00   Median : 15.00   Median : 48.0   Median : 87.00
##  Mean   : 35.66   Mean   : 22.73   Mean   : 62.6   Mean   : 90.11
##  3rd Qu.: 48.00   3rd Qu.: 30.00   3rd Qu.: 80.0   3rd Qu.:121.00
##  Max.   :128.00   Max.   :190.00   Max.   :312.0   Max.   :237.00
##       PTS              PER
##  Min.   :  43.0   Min.   : 2.76
##  1st Qu.: 236.0   1st Qu.:13.46
##  Median : 416.0   Median :16.24
##  Mean   : 534.1   Mean   :17.03
##  3rd Qu.: 754.0   3rd Qu.:19.98
##  Max.   :2015.0   Max.   :36.03
```

```
# Cabecera del conjunto de datos
head(playerstats)
```

```
##           Nombre                Equipo Age Pos   MP  G FG2p FG3p  FTp ORB DRB
## 1  Aaron Gordon       Denver Nuggets  25  PF  650 25 0.61 0.27 0.70  38  80
## 2  Aaron Gordon        Orlando Magic  25  PF  737 25 0.48 0.38 0.63  39 127
## 3 Aaron Holiday       Indiana Pacers  24  PG 1176 66 0.41 0.37 0.82  15  74
## 4 Aaron Nesmith       Boston Celtics  21  SF  665 46 0.54 0.37 0.79  28  99
## 5   Abdel Nader         Phoenix Suns  27  SF  353 24 0.53 0.42 0.76   7  55
```

```
## 7    Al Horford Oklahoma City Thunder  34   C  782 28 0.51 0.37 0.82  29 159
##   AST STL BLK TOV PF PTS   PER
## 1  56  17  14  30 40 254 16.39
## 2 105  16  20  67 49 364 17.97
## 3 123  46  13  66 94 475 12.15
## 4  23  15   9  23 87 218 11.34
## 5  19  10   9  19 34 160 16.07
## 7  94  25  26  29 48 398 21.14
```

4 Análisis de los datos

4.1 Selección de los grupos de datos que se quieren analizar/comparar

Con respecto a la selección de los grupos de datos, esta se va a realizar en cada análisis para no perder el hilo durante la realización de estos. Sin embargo, se pasa a comentar que conjuntos de datos se van a tener en cuenta y que análisis se van a realizar sobre estos:

- *Estudio de correlaciones lineales.* Este estudio permite asegurarse de la dependencia/independecia entre las variables que componen el conjunto de datos creado en los apartados anteriores. El objetivo de este análisis es conocer que variables tienen una mayor relevancia a la hora de calcular el PER y nos ayudará a descartar aquellas variables que no lo definen. Para ello, el conjunto de datos que se utilizará estará compuesto por la variables numéricas estadísticas que componen el conjunto de datos construido durante esta práctica.
- *Regresión lineal.* Con respecto al análisis de regresión lineal se va a realizar la creación de dos modelos que permitan calcular el valor del PER a partir de esto. El primero de estos modelos, hará uso de las variables numéricas estadísticas y en el segundo se añadirá la edad del jugador para conocer si esta influye en el valor del PER. Por lo tanto, el conjunto de datos que se utilizará para este análisis será el compuesto por las variables numéricas del conjunto de datos contruido durante la práctica a excepción de las variables FP y PTS ya que no intervienen en el PER.
- *Regresión logistica.* Mediante la creación de un modelo de regresión logistica, se va a proceder a saber si un jugador puede ser categorizado como grande o pequeño en función de sus apartados estadísticos. Por lo tanto, el conjunto de datos que se utilizará para este análisis será el compuesto por las variables numéricas del conjunto de datos contruido durante la práctica a excepción de las variables FP y PTS ya que no intervienen en el PER.
- *Contraste de hipótesis.* A partir del contraste de hipótesis se va a proceder a comprobar si es verdad la creencia generalizada en el mundo del baloncesto de que los jugadores que juegan en la posición de CENTER tienen un mayor PER que los que juegan en el resto de posiciones. Para ello, se va a proceder a la creación de dos conjuntos de datos en los que en el primero solo se incluyan los jugadores que juegan en la posición de CENTER y las variables estadísticas numéricas que hayan obtenido, y en el segundo, la de los jugadores del resto de posiciones.

Estos seran los conjuntos de datos y los análisis que se procederan a realizar a continuación.

4.2 Comprobación de la normalidad y homogeneidad de la varianza

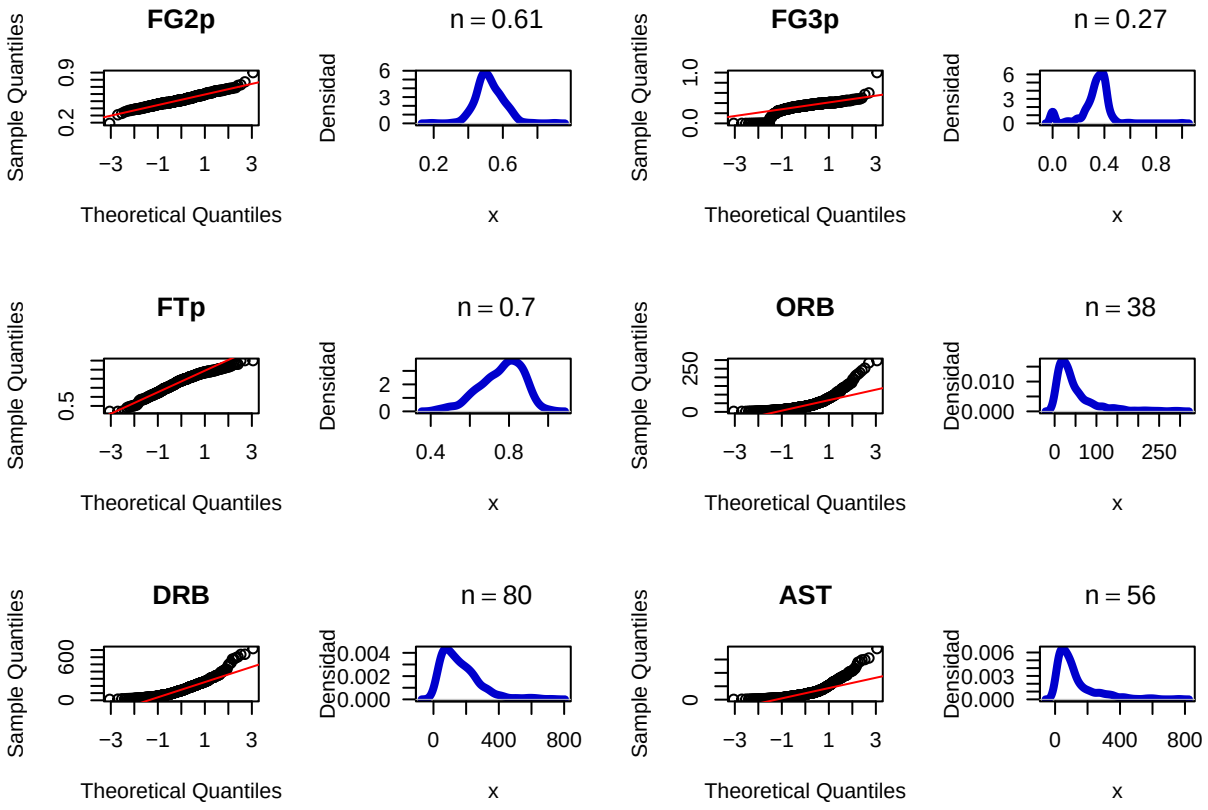
4.2.1 Comprobación de la normalidad

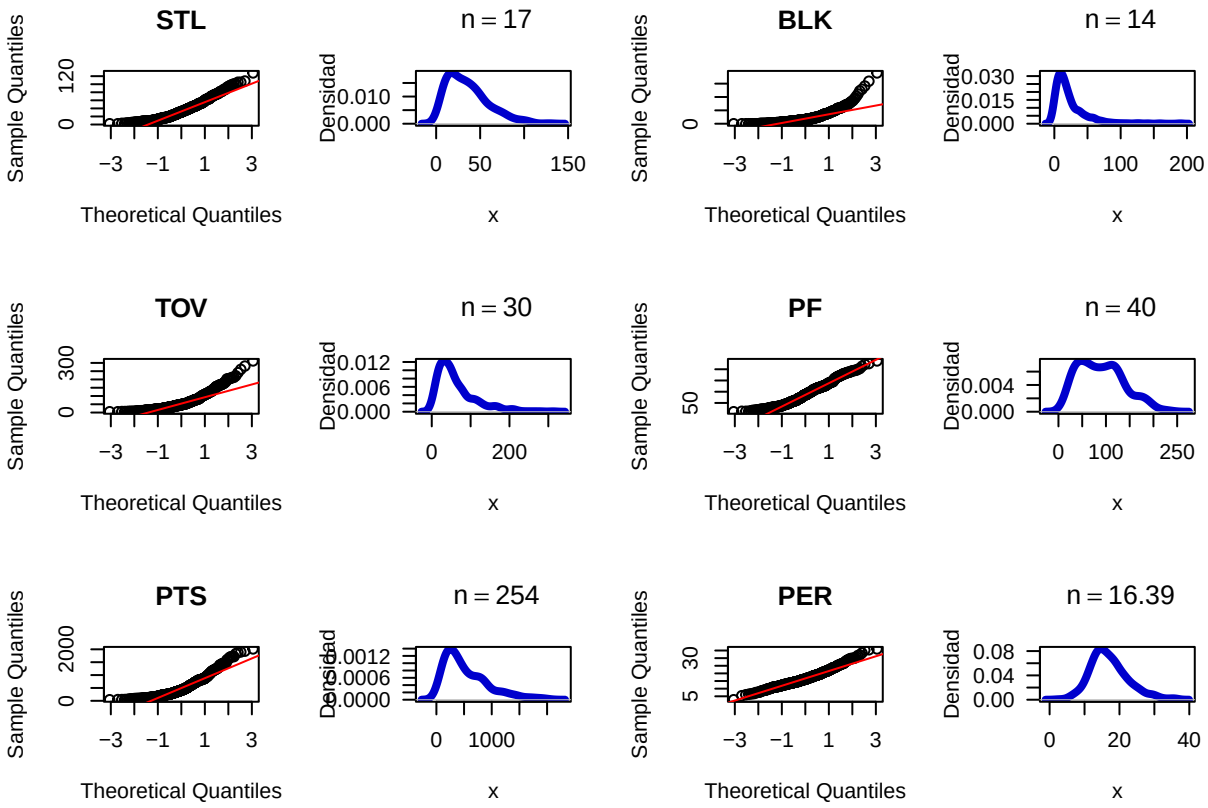
Para realizar la comprobación de la normalidad, se va a realizar el estudio sobre las variables numéricas que se utilizarán posteriormente en los modelos. Es decir, la variables sobre las cuales se procede a estudiar la normalidad y la homogeneidad de la varianza que son *FG2p*, *FG3p*, *FTp*, *ORB*, *DRB*, *AST*, *STL*, *BLK*, *TOV*, *PF*, *PTS*, *PER*:

```
par(mfrow=c(3,4))
for(i in 7:ncol(playerstats)) {
  qqnorm(playerstats[,i],main = paste(colnames(playerstats)[i]))
  qqline(playerstats[,i],col="red")
```

```
  x <-playerstats[,i]
  plot(density(x), main=bquote(~ n == .(playerstats[,i])),
       ylab='Densidad', col='blue3', xlab='x', las=1, lwd=4)


}
```





Como se puede observar, tenemos variables que, mediante los gráficos Q-Q y de densidad, podemos concluir que siguen una distribución bastante normal. Estas variables serían FG2p, FTp, PF y PER. El resto, se alejan un poco de la simetría que se buscaría en este tipo de distribuciones aunque tampoco se podría descartar al 100% la normalidad. Para confirmar esto, se podrían realizar tests de normalidad como el de Shapiro, pero al tener una cantidad de muestras muy grande (superior a 30), no nos podemos fiar demasiado de estos resultados.

```
for(i in 7:ncol(playerstats)) {
  print(paste("Tests de ", colnames(playerstats)[i]))
  print(shapiro.test(x)$p.value)
}
```

```
## [1] "Tests de  FG2p"
## [1] 2.762569e-07
## [1] "Tests de  FG3p"
## [1] 2.762569e-07
## [1] "Tests de  FTp"
## [1] 2.762569e-07
## [1] "Tests de  ORB"
## [1] 2.762569e-07
## [1] "Tests de  DRB"
## [1] 2.762569e-07
## [1] "Tests de  AST"
## [1] 2.762569e-07
## [1] "Tests de  STL"
## [1] 2.762569e-07
## [1] "Tests de  BLK"
```

```
## [1] 2.762569e-07
## [1] "Tests de  TOV"
## [1] 2.762569e-07
## [1] "Tests de  PF"
## [1] 2.762569e-07
## [1] "Tests de  PTS"
## [1] 2.762569e-07
## [1] "Tests de  PER"
## [1] 2.762569e-07
```

Se aprecia que según estos tests ningúna variable sigue una distribución normal, ya que nos indica en todas que el p-value es menor que el coeficiente 0.05 y nos indicaría que se puede rechazar la hipótesis nula, lo que en resumen significaría que no siguen una distribución normal. Sin embargo, teniendo en cuenta que tenemos 437 muestras, se puede aplicar el Teorema del Límite Central, que establece que el contraste de hipótesis sobre la media de una muestra se aproxima a una distribución normal aunque la población original no siga una distribución normal, siempre que el tamaño de la muestra sea suficientemente grande.

4.2.2 Homogeneidad de la varianza

Para la homogeneidad de la varianza, se va a hacer uso del test de Fligner-Killeen para su estudio. En este estudio, se va a comparar todas las variables con la variable del PER ya que esta se extrae de las anteriores. Por lo tanto, suponiendo que la hipótesis nula consiste en que ambas varianzas son iguales, se procede a aplicar el test:

```
for(i in 7:(ncol(playerstats)-1)) {
print(paste('Test sobre la homogeneidad',colnames(playerstats)[i],' - PER'))
flitest <- fligner.test(playerstats[,i] ~ PER, data = playerstats)
print(flitest$p.value)
}
```

```
## [1] "Test sobre la homogeneidad FG2p  - PER"
## [1] 0.0975209
## [1] "Test sobre la homogeneidad FG3p  - PER"
## [1] 0.09876332
## [1] "Test sobre la homogeneidad FTp  - PER"
## [1] 0.1073745
## [1] "Test sobre la homogeneidad ORB  - PER"
## [1] 0.1093564
## [1] "Test sobre la homogeneidad DRB  - PER"
## [1] 0.06738393
## [1] "Test sobre la homogeneidad AST  - PER"
## [1] 0.05829403
## [1] "Test sobre la homogeneidad STL  - PER"
## [1] 0.05925969
## [1] "Test sobre la homogeneidad BLK  - PER"
## [1] 0.08415959
## [1] "Test sobre la homogeneidad TOV  - PER"
## [1] 0.04914693
## [1] "Test sobre la homogeneidad PF  - PER"
## [1] 0.08565374
## [1] "Test sobre la homogeneidad PTS  - PER"
## [1] 0.04771163
```

Como se puede observar, practicamente todas las variables son homogeneas con la variable PER ya que el p-valor es superior a 0.05. Las variables que no son homogeneas son PTS y TOV que a pesar de estar por debajo se acercan mucho. Los resultados obtenidos, nos seran útiles para los test posteriores sobre el

contraste de hipótesis donde tendremos que saber si las varianzas son igual o no.

4.3 Aplicación de pruebas estadísticas para comparar los grupos de datos

4.3.1 Estudio de correlaciones lineales

El estudio de correlaciones pretende determinar si dos variables están relacionadas[2]. El resultado del análisis es un coeficiente de correlación que tomará valores entre -1 y +1. Si el signo es positivo, indica que si una variable aumenta, la otra también, ya que existe una relación positiva entre las dos variables. Si el signo es negativo indica que la relación es negativa y mientras los valores de una variable incrementan, los de la otra disminuyen. En cambio, Si las variables son independientes, el coeficiente de correlación será 0. La fuerza de la relación lineal incrementa a medida que el coeficiente de correlación se aproxima a -1 o a +1 y disminuye cuanto más se acerque al 0.

Ninguna de las variables explicativas puede ser combinación lineal de las otras, ya que en este caso no tendríamos un modelo de k variables, sino de k-1 variables (queremos que las variables Xi sean independientes). Debido a la naturaleza de los datos de nuestro conjunto, no sera necesario comprobar si existe una dependencia lineal entre las variables predictoras:

```
x <- playerstats[7:18]
y <- playerstats[18]
cor_mat <- cor(y, x, use = "complete.obs")
cor_mat
```

```
##           FG2p        FG3p       FTp       ORB       DRB       AST       STL
## PER 0.4073303 -0.01475301 0.1507436 0.473949 0.5751889 0.500181 0.3701477
##           BLK       TOV        PF       PTS PER
## PER 0.4007462 0.5740526 0.3344931 0.6709724   1
```

Tras ejecutar la función *cor()*, podemos ver las correlaciones que existen entre la variable PER y las demás variables numéricas. Primeramente vemos que la variable más independiente, es decir, donde el coeficiente de correlación es 0 o lo más próximo, es en el porcentaje de tiros de 3 puntos (FG3p), donde el resultado es -0.0147. Sin embargo, el resto de variables tienen relaciones positivas, lo que significa que si aumentan, hacen aumentar el PER, lo cual tiene sentido teniendo en cuenta lo que representan. La variable que parece tener una mayor relación con el PER es la de los puntos anotados, que aunque no se utilicen para calcularlo, también acaban representando el rendimiento de un jugador. Otras variables que muestran una gran relación con el PER son las de TOV, DRB y AST, que se mantienen por encima de 0.5 en positivo. Tiene sentido que estas variables tengan relaciones altas debido a que es más probable que cualquier jugador de cualquier posición pueda destacar por ello, mientras que las variables de anotar puntos o realizar faltas pierden importancia a la hora de describir el PER de todas las posiciones.

Gracias a este análisis podríamos llegar a descartar la variable FG3p ya que parece tener poca importancia, pero se acabará de comprobar si este resultado también se refleja en los análisis predictivos que se realizarán a continuación.

4.3.2 Regresión lineal

Debido a que la variable de PER, se calcula a partir de la variables ORB, DRB, AST, STL, BLK, TOV y los volumenes de tiro. A continuación, se va a crear un modelo de regresión lineal el cual nos permita conocer que variables de las que tenemos en el conjunto de datos tienen una mayor importancia a la hora de realizar predicción del PER haciendo uso de este tipo de regresión. Es por esto que en primer lugar, se construye el modelo con las variables anteriormente comentadas y los porcentajes de efectivias en el tiro de los jugadores:

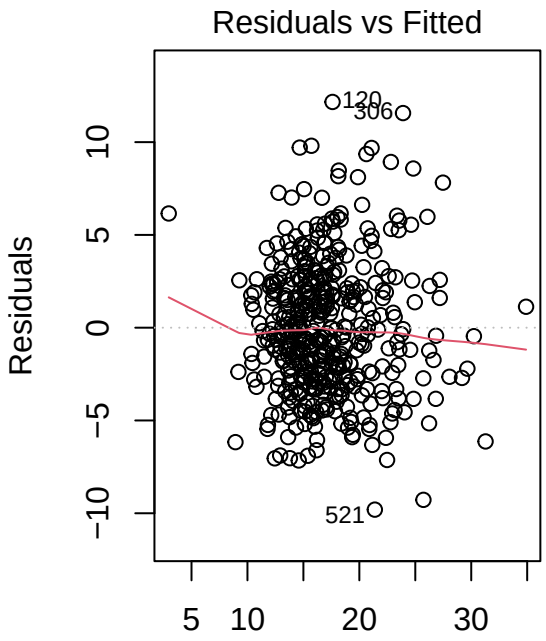
```
regresion_1 <- lm(PER ~ FG2p+FG3p+FTp+ORB+DRB+AST+STL+BLK+TOV, data = playerstats)
summary(regresion_1)
```

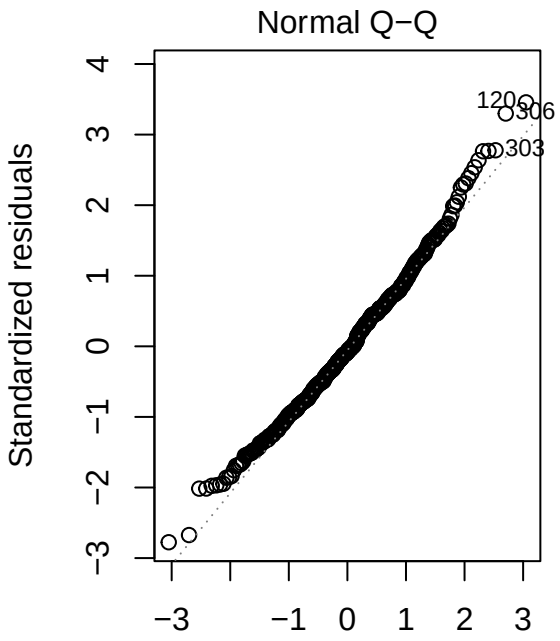
```
##
## Call:
```

```
## lm(formula = PER ~ FG2p + FG3p + FTp + ORB + DRB + AST + STL +
##     BLK + TOV, data = playerstats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.8047 -2.6133 -0.2728  2.2345 12.1650
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.9301040  2.0016471  -3.462  0.00059 ***
## FG2p        22.4762942  2.6162923   8.591  < 2e-16 ***
## FG3p         2.7065903  1.8647465   1.451  0.14739
## FTp         10.0536564  1.8581824   5.410 1.05e-07 ***
## ORB          0.0312166  0.0074715   4.178 3.57e-05 ***
## DRB         -0.0009365  0.0033345  -0.281  0.77897
## AST          0.0175997  0.0039438   4.463 1.04e-05 ***
## STL         -0.0554290  0.0121959  -4.545 7.17e-06 ***
## BLK          0.0280043  0.0113764   2.462  0.01423 *
## TOV          0.0262912  0.0089334   2.943  0.00343 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.58 on 427 degrees of freedom
## Multiple R-squared:  0.5619, Adjusted R-squared:  0.5527
## F-statistic: 60.86 on 9 and 427 DF,  p-value: < 2.2e-16
```

Como se puede observar las variables que tienen una mayor relevancia a la hora de obtener el PER según el modelo creado son las variables FG2p, FTp, ORB, AST y STL. Adicionalmente, podemos observar que el modelo creado tiene un coeficiente de determinación de 0.56 lo que nos permite suponer que el modelo ajusta a más de la mitad de los datos. Para poder confirmar esta suposición se pasa a realizar la visualización del grafico de los residuos y QQ con el que podremos entender de forma visual como se ajusta el modelo a los datos:

```
par(mfrow = c(1, 2))
plot(regresion_1,which = 1)
plot(regresion_1,which = 2)
```





Como se puede observar, la suposición anterior es cierta y el modelo representa más de la mitad de los datos de conjunto de datos. Adicionalmente, se va a crear un modelo que a las variables anteriormente utilizadas al que se le incluya la variable de Age que representa la edad del jugador con el fin de conocer si la edad de un jugador es tiene relevancia a la hora de que un jugador tenga un mayor PER o no:

```
regresion_l_age <- lm(PER ~ FG2p+FG3p+FTp+ORB+DRB+AST+STL+BLK+TOV+Age, data = playerstats)
summary(regresion_l_age)
```

```
##
## Call:
## lm(formula = PER ~ FG2p + FG3p + FTp + ORB + DRB + AST + STL +
##     BLK + TOV + Age, data = playerstats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.7521 -2.5585 -0.3002  2.3209 12.3678
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.042550   2.190186  -3.672 0.000271 ***
## FG2p        22.362831   2.616174   8.548 2.26e-16 ***
## FG3p         2.700236   1.863541   1.449 0.148079
## FTp          9.740066   1.873918   5.198 3.14e-07 ***
## ORB          0.031503   0.007470   4.217 3.02e-05 ***
## DRB         -0.001277   0.003344  -0.382 0.702600
## AST          0.016594   0.004023   4.125 4.46e-05 ***
## STL         -0.054558   0.012208  -4.469 1.01e-05 ***
```

```
## BLK          0.027178   0.011388   2.386 0.017446 *
## TOV          0.028645   0.009125   3.139 0.001812 **
## Age          0.054427   0.043638   1.247 0.212994
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.578 on 426 degrees of freedom
## Multiple R-squared:  0.5635, Adjusted R-squared:  0.5533
## F-statistic:    55 on 10 and 426 DF,  p-value: < 2.2e-16
```

Como se puede observar, la inclusión de la variable Age en el modelo no tiene una gran relevancia ya que este mejora muy levemente con un coeficiente de determinación de 0.56. Por lo tanto, para la siguiente parte del análisis, se va a hacer uso del primer modelo creado ya que no se aprecia casi mejoría. Por lo tanto, la recta de regresión del primer modelo creado es la siguiente:

$$y = -6.9301 + 22.4763x_1 + 2.7066x_2 + 10.0536x_3 + 0.0312x_4 - 0.0009x_5 + 0.0175x_6 - 0.0554x_7 + 0.0280x_8 + 0.0262x_9$$

Para finalizar, se va a proceder a realizar una predicción haciendo uso del primer modelo. Para ello, se va a selecciónar el primero y el séptimo de los registros y se va a comprobar si el PER del conjunto de datos es similar al PER calculado por el modelo:

```
seleccion_1 <- playerstats[1,]
pred <- predict(regresion_1, seleccion_1[7:15], type ="response")
print(paste('PER al calculado del jugador 1:',round(head(pred),2)))
```

```
## [1] "PER al calculado del jugador 1: 16.88"
```

```
print(paste('PER original del jugador 1:',seleccion_1[1,18]))
```

```
## [1] "PER original del jugador 1: 16.39"
```

```
seleccion_2 <- playerstats[7,]
pred <- predict(regresion_1, seleccion_2[7:15], type ="response")
print(paste('PER calculado del jugador 2:',round(head(pred),2)))
```

```
## [1] "PER calculado del jugador 2: 14.8"
```

```
print(paste('PER original del jugador 2:',seleccion_2[1,18]))
```

```
## [1] "PER original del jugador 2: 17.36"
```

Como podemos observar, para el primer jugador, se obtiene un PER casi idéntico pero para el segundo esta diferencia se amplía a 2.5. Por lo tanto, y como se había visualizado, el modelo no se ajusta a todos los datos del conjunto de datos.

4.3.3 Regresión logística

Como segundo análisis, sobre el conjunto de datos, se ha decidido realizar un modelo de regresión logística el cual nos permita conocer que según sus estadísticas el jugador en cuestión es pequeño (Guard - G, Point Guard - PG, Small Guard - SG o Small Forward - SF) o grande (Paint Forward- PF o Center - C). Para ello, en primer lugar, se va a realizar la creación de una nueva variable llamada *tamanyo* la cual identifique si un jugador por su posición es pequeño o grande:

```
playerstats$tamanyo<-"Small"
playerstats$tamanyo[playerstats$Pos=="C" | playerstats$Pos=="PF"] <- "Big"
playerstats$tamanyo <- as.factor(playerstats$tamanyo)
```

Una vez creada esta nueva variable y factorizada, se pasa a crear el modelo de regresión logística el cual nos permita conocer si un jugador es pequeño o grande según sus estadísticas:

```
regresion_log <- glm(formula=tamanyo~FG2p+FG3p+FTp+ORB+DRB+AST+STL+BLK+TOV, data = playerstats, family=
summary(regresion_log)
```

```
##
## Call:
## glm(formula = tamanyo ~ FG2p + FG3p + FTp + ORB + DRB + AST +
##     STL + BLK + TOV, family = binomial(link = logit), data = playerstats)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6008  -0.4430   0.1876   0.6276   2.9282
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.445200   1.517074  -0.953 0.340781
## FG2p        -1.071422   1.930399  -0.555 0.578877
## FG3p         4.517624   1.610913   2.804 0.005041 **
## FTp          1.499683   1.421367   1.055 0.291380
## ORB         -0.023848   0.008013  -2.976 0.002919 **
## DRB         -0.014870   0.003207  -4.637 3.54e-06 ***
## AST          0.001376   0.004201   0.327 0.743327
## STL          0.080592   0.013627   5.914 3.34e-09 ***
## BLK         -0.049606   0.014119  -3.513 0.000442 ***
## TOV          0.018467   0.009483   1.947 0.051494 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 595.50  on 436  degrees of freedom
## Residual deviance: 331.42  on 427  degrees of freedom
## AIC: 351.42
##
## Number of Fisher Scoring iterations: 6
```

Tras la creación del modelo, se va a pasar a predecir con este, el tamaño de los jugadores según sus estadísticas. Para ello, se hará uso de la matriz de confusión para la obtención de resultados:

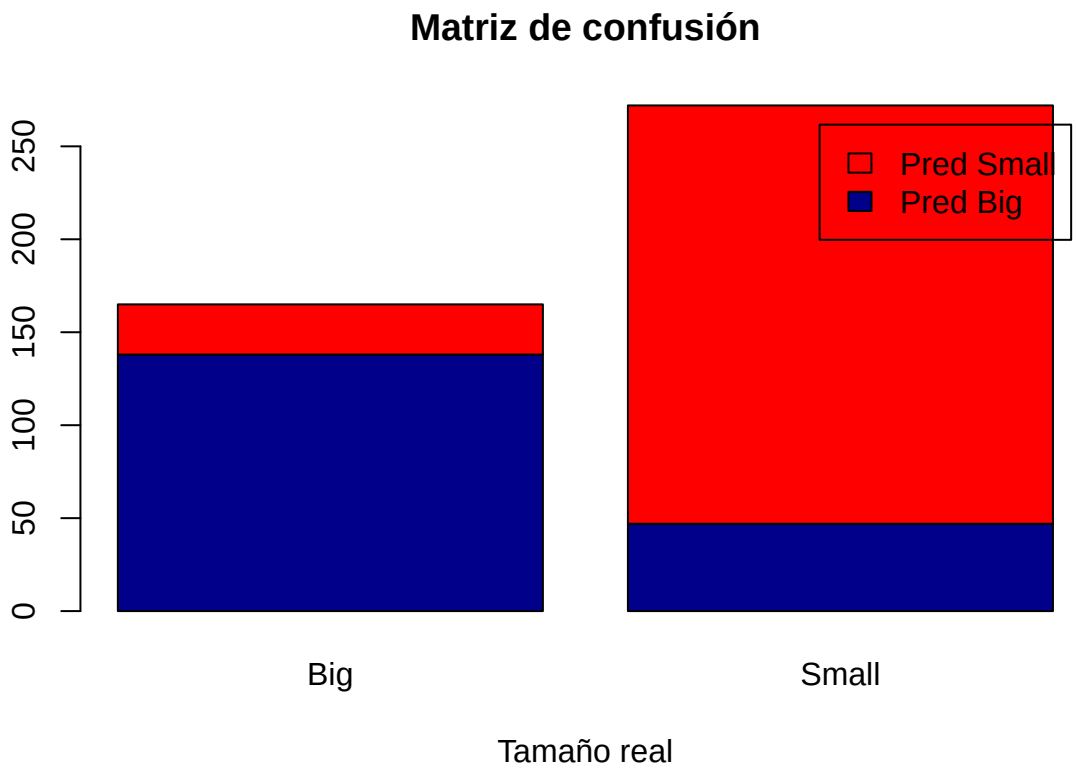
```
playerstats$PROB_PRED <- round(predict(regresion_log, newdata = playerstats, "response"),4)
playerstats$TAM_PRED <- ifelse(playerstats$PROB_PRED > 0.5, "Small", "Big")
y_pred <- as.factor(playerstats$TAM_PRED)
y_obs <- as.factor(playerstats$tamanyo)
confusionMatrix(data=y_pred, reference = y_obs,  positive="Small")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Big Small
##      Big   138    27
##      Small  47   225
##
##                Accuracy : 0.8307
##                  95% CI : (0.7921, 0.8646)
```

```
##     No Information Rate : 0.5767
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6481
##
##  Mcnemar's Test P-Value : 0.0272
##
##             Sensitivity : 0.8929
##             Specificity : 0.7459
##          Pos Pred Value : 0.8272
##          Neg Pred Value : 0.8364
##              Prevalence : 0.5767
##          Detection Rate : 0.5149
##    Detection Prevalence : 0.6224
##       Balanced Accuracy : 0.8194
##
##        'Positive' Class : Small
##
```

Como se puede observar, el modelo ha tenido una precisión del 83% aproximadamente y ha clasificado correctamente 138 jugadores como hombres grandes y 225 como pequeños según su posición. Para finalizar, se muestra la distribución de los casos clasificados correctamente e incorrectamente por el modelo creado:

```
Tabla <-  table(y_obs, y_pred)
barplot(Tabla, main="Matriz de confusión",
  xlab="Tamaño real", col=c("darkblue","red"),
  legend.text=c("Pred Big","Pred Small")
)
```



4.3.4 Contraste de hipótesis

Como último análisis sobre el conjunto de datos, se pasa a realizar un constraste de hipótesis en el que se intentará dar una respuesta sobre el conjunto de datos a la afirmación de que los jugadores que juegan en la posición de Center obtenienen un mayor ratio de eficiencia(PER) que el resto de jugadores que juegan en el resto de posiciónes. Por lo tanto, para este análisis plantearemos que la hipótesis nula es que los jugadores del resto de las posiciones tienen un mayor o igual PER que los centers y como hipótesis alternativa la contraria:

$$\begin{aligned} H_0: &\quad PER_{center} =< PER_{others} \\ H_1: &\quad PER_{center} > PER_{others} \end{aligned}$$

Por lo tanto, una vez planteada las hipótesis se pasa a realizar el contraste mediante la función *t.test* en el que se define como parámetros var.equal a TRUE ya que como se ha podido comprobar antes las varianzas son iguales y el parámetro conf.level con un nivel de confianza del 95%:

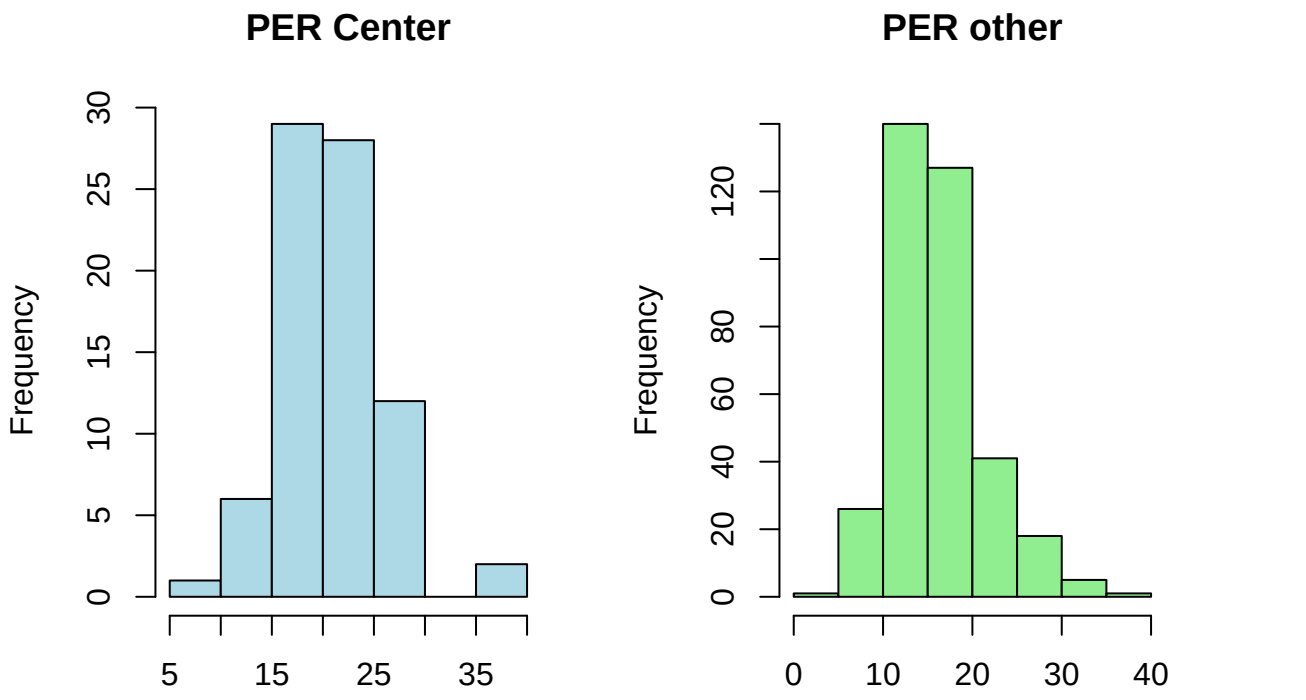
```
# REFERENCIA:
t.test(
  x           = playerstats$PER[playerstats$Pos=='C'],
  y           = playerstats$PER[playerstats$Pos!='C'],
  alternative = "greater",
  var.equal   = TRUE,
  conf.level  = 0.95
)
```

```
##
##  Two Sample t-test
##
```

```
## data:  playerstats$PER[playerstats$Pos == "C"] and playerstats$PER[playerstats$Pos != "C"]
## t = 7.6007, df = 435, p-value = 9.139e-14
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  3.744521      Inf
## sample estimates:
## mean of x mean of y
##  20.96090  16.17942
```

Tras realizar el test, podemos ver que el p-value es muy próximo a 0, por lo tanto es menor que alpha y podríamos rechazar la hipótesis nula que dice que el PER de los jugadores de la posición de center tienen un PER menor. Si visualizamos la distribución de ambos conjuntos de datos podremos observar que los jugadores en la posición de center tienen un mayor PER que el resto de jugadores:

```
mean_center <- mean(playerstats$PER[playerstats$Pos=='C'])
mean_noCenter <- mean(playerstats$PER[playerstats$Pos!='C'])
par(mfrow=c(1,2))
hist(playerstats$PER[playerstats$Pos=='C'], main="PER Center", xlab = NULL, col = "lightblue")
hist(playerstats$PER[playerstats$Pos!='C'], main="PER other", xlab = NULL, col = "lightgreen")
```



5 Resolución del problema

Para resolver el problema lo primero que se ha hecho es un estudio sobre la correlación de las variables con el PER, es decir, la variable que buscamos predecir. De este análisis hemos obtenido conclusiones positivas donde se confirma que todas las variables están bastante relacionadas con esta nota que se atribuye a los jugadores, algo que tiene lógica, pero hemos observado que los tiros de 3 puntos no eran demasiado necesarios.

Tras la realización de los análisis empleados en el apartado anterior, se ha podido demostrar que, para el cálculo de la variable PER en un modelo de regresión lineal, las variables que tienen una mayor importancia son FG2p, FTp, ORB, AST y STL, descartando también los tiros de 3 puntos. Sin embargo, se ha demostrado que este modelo no se ajusta a la totalidad de los datos ya que solo se ajusta a un 56%. Adicionalmente, se ha comprobado que la inclusión en el modelo de la variable Age, que describe la edad de los jugadores, no mejora la precisión de este, demostrando que la edad no importa a la hora de obtener un mejor o peor PER.

Con respecto al modelo de regresión logística, se ha comprobado que se puede conocer el tamaño de los jugadores a partir de sus estadísticas con un 83% de precisión. Esto nos ha permitido afirmar que los jugadores tendrán una estadísticas y otras según su tamaño y que, apartir de estas, se puede conocer si son jugadores grandes o pequeños.

Por último, mediante el constraste de hipótesis, se ha visto que por normal general los jugadores que juegan en la posición de Center obtienen un mayor PER que el resto de posiciones, y por lo tanto, se ha confirmado la suposición del mundo del baloncesto en la que los jugadores de esta posición tienen un mayor PER.

6 Referencias

[1] https://www.kaggle.com/datasets/umutalpaydn/nba-20202021-season-player-stats?resource=download [2] José Alquicira. (2017). Análisis de correlación. 2022, Junio 2, Conogasi.org Sitio web: https://conogasi.org/articulos/analisis-de-correlacion-2/ [3] https://rpubs.com/Joaquin_AR/218467

7 Firma de apartados

Contribuciones	Firma
Invetigación previa	DLT y FJCZ
Redacción de las respuestas	DLT y FJCZ
Desarrollo código	DLT y FJCZ