Práctica 2: Limpieza y validación de los datos

Autores: Mikel Laburu Haro, Unai Mateos Corral

Curso 2018/2019

Índice

1. Introducción

1.1. Descripción

En esta actividad se lleva a cabo el tratamiento del dataset "Fifa 18 More Complete Player Dataset" extraido de Kaggle (https://www.kaggle.com/kevinmh/fifa-18-more-complete-player-dataset/home), con el que se elabora un caso práctico en el que se emplean herramientas de integración, limpieza, validación y análisis.

1.2. Objetivos

Los objetivos que se pretenden lograr mediante la elaboración de esta actividad son los siguientes:

- Aprender a aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios o multidisciplinares.
- Saber identificar los datos relevantes y los tratamientos necesarios (integración, limpieza y validación) para llevar a cabo un proyecto analítico.
- Aprender a analizar los datos adecuadamente para abordar la información contenida en los datos.
- Identificar la mejor representación de los resultados para aportar conclusiones sobre el problema planteado en el proceso analítico.
- Actuar con los principios éticos y legales relacionados con la manipulación de datos en función del ámbito de aplicación.

- Desarrollar las habilidades de aprendizaje que les permitan continuar estudiando de un modo que tendrá que ser en gran medida autodirigido o autónomo.
- Desarrollar la capacidad de búsqueda, gestión y uso de información y recursos en el ámbito de la ciencia de datos.

1.3. Competencias

Desarrollando a su vez las siguientes competencias del máster de Ciencia de Datos:

- Capacidad de analizar un problema en el nivel de abstraccn adecuado a cada situación y aplicar las habilidades y conocimientos adquiridos para abordarlo y resolverlo.
- Capacidad para aplicar las técnicas específicas de tratamiento de datos (integración, transformación, limpieza y validación) para su posterior análisis.

2. Dataset

2.1. Descripción del dataset

A continuación se realiza la carga del conjuto de datos al completo, y posteriormente se hace la selección de los atributos que se consideran de mayor relevancia, llegando a simplificar el datataset considerablemente, reduciendo el número de columnas de 185 a 46. Además, a modo de ejemplo, se muestra el valor de cinco atributos de las diez primeras instancias.

```
# Cargamos el Dataset
myData <- read.csv("FIFA.csv", header = TRUE, sep = ",")
# Seleccion de los atributos a utilizar
myDataAux <- myData[, c("name", "club", "age", "height_cm",
    "weight_kg", "nationality", "eur_value", "eur_wage",
    "eur_release_clause", "overall", "potential", "pac",
    "sho", "pas", "dri", "def", "phy", "crossing", "finishing",
    "heading_accuracy", "short_passing", "volleys", "dribbling",
    "curve", "free_kick_accuracy", "long_passing", "ball_control",
    "acceleration", "sprint_speed", "agility", "reactions",
    "balance", "shot_power", "jumping", "stamina", "strength",
    "long_shots", "aggression", "interceptions", "positioning",
    "vision", "penalties", "composure", "marking", "standing_tackle",
    "sliding_tackle")]

# Mostrar las primeras 10 lineas del dataset y el
# contenido de 5 variables
head(myDataAux[, 1:5], 10)
```

```
##                 name               club age height_cm weight_kg
## 1  Cristiano Ronaldo      Real Madrid CF  32       185        80
## 2           L. Messi        FC Barcelona  30       170        72
## 3             Neymar Paris Saint-Germain  25       175        68
## 4         L. SuÃ¡rez        FC Barcelona  30       182        86
## 5           M. Neuer    FC Bayern Munich  31       193        92
## 6     R. Lewandowski    FC Bayern Munich  28       185        79
## 7              De Gea   Manchester United  26       193        76
## 8          E. Hazard             Chelsea  26       173        76
## 9           T. Kroos      Real Madrid CF  27       182        78
## 10       G. HiguaÃn            Juventus  29       184        87
```

```
# Tipo de los datos de cada variable
sapply(myDataAux, class)
```

```
##               name               club                age
##           "factor"           "factor"          "integer"
##          height_cm          weight_kg        nationality
##          "numeric"          "numeric"           "factor"
##          eur_value           eur_wage eur_release_clause
##          "numeric"          "numeric"          "numeric"
##            overall          potential                pac
##          "integer"          "integer"          "integer"
##                sho                pas                dri
##          "integer"          "integer"          "integer"
##                def                phy           crossing
##          "integer"          "integer"          "integer"
##          finishing   heading_accuracy      short_passing
##          "integer"          "integer"          "integer"
##            volleys          dribbling              curve
##          "integer"          "integer"          "integer"
## free_kick_accuracy       long_passing       ball_control
##          "integer"          "integer"          "integer"
##       acceleration       sprint_speed            agility
##          "integer"          "integer"          "integer"
##          reactions            balance         shot_power
##          "integer"          "integer"          "integer"
##            jumping            stamina           strength
##          "integer"          "integer"          "integer"
##         long_shots         aggression      interceptions
##          "integer"          "integer"          "integer"
##        positioning             vision          penalties
##          "integer"          "integer"          "integer"
##          composure            marking    standing_tackle
##          "integer"          "integer"          "integer"
##     sliding_tackle
##          "integer"
```

Este conjunto de datos consta de un total de 17994 registros de jugadores, y como ya se ha comentado, cada una de ellos queda finalmente caracterizado por 46 atributos, entre los que se puede observar su nombre, su salarioo sus habilidades. A continuación se realiza una breve descripción de alguno de estos atributos:

- **Name**: Nombre completo del jugador.
- **Club**: Equipo al que pertenece el jugador.
- **Age**: Edad del jugador.
- **Eur_value**: Valor en euros que el juego estima para el jugador.
- **Eur_wage**: Salario en euros del jugador.
- **Eur_release_clause**: Cláusula de rescisión del jugador.
- **Overall**: Puntuación media del jugador.
- **PAC**: (Ritmo) Atributo que representa la media de los atributos de veloricadad (aceleración, velocidad...).
- **Sho**: (Disparo) Atributo que representa la media de los atributos de disparo (finalización, remate de cabeza...).
- **Pas**: (Pase) Atributo que representa la media de los atributos de pase (pase en corto, pase en largo...).
- **Dri**: (Regate) Atributo que representa la media de los atributos de regate (control del balón, dribring...).
- **Def**: (Defensa) Atributo que representa la media de los atributos defensivos (anticipación, entrada...).
- **Phy**: (Físico) Atributo que representa la media de los atributos físicos (resistencia, fuerza...).

3. Limpieza de los datos

3.1. Eliminación de ceros, vacios y nulos

En este apartado se pretende limpiar el conjunto de datos de tal forma que no disponga ni de ceros, ni de valores vacíos, ni de valores nulos.

En primer lugar se realizará el tratamiento de valores nulos, como se aprecia a continuación el conjunto de datos dispone de valores nulos en los campos de la columna "eur_release_clause". Para no disponer de estos valores existen distintas estrategias como eliminar los registros en los que aparece este valor o asignar la media de "eur_release_clause" a estos campos, pero en este caso se ha considerado más oportuno la imputación de valores basada en k vecinos más próximos, o más comúnmente conocida como $kNN - imputation$, debido principalmente a que los registros de este conjunto de datos guardan relación entre sí.

```
library(VIM)

colSums(is.na(myDataAux))
```

```
##               name               club                age
##                  0                  0                  0
##          height_cm          weight_kg        nationality
##                  0                  0                  0
##          eur_value           eur_wage eur_release_clause
##                  0                  0               1494
##            overall          potential                pac
##                  0                  0                  0
##                sho                pas                dri
##                  0                  0                  0
##                def                phy           crossing
##                  0                  0                  0
##          finishing   heading_accuracy      short_passing
##                  0                  0                  0
##            volleys          dribbling              curve
##                  0                  0                  0
## free_kick_accuracy       long_passing       ball_control
##                  0                  0                  0
##       acceleration       sprint_speed            agility
##                  0                  0                  0
##          reactions            balance        shot_power
##                  0                  0                  0
##            jumping            stamina           strength
##                  0                  0                  0
##         long_shots         aggression      interceptions
##                  0                  0                  0
##        positioning             vision          penalties
##                  0                  0                  0
##          composure            marking    standing_tackle
##                  0                  0                  0
##     sliding_tackle
##                  0
```

```
# Tratamiento de valores nulos
myDataAux$eur_release_clause <- kNN(myDataAux)$eur_release_clause
```

Tras comprobar que no quedan valores nulos, es momento de tratar los campos en los que debería haber ceros

y sí los hay. Para esto, al igual que antes, primero se muetran a ver qué columnas pueden poseer algún cero, y tal y como se aprecia los campos "eur_value" y "eur_wage" disponen de ceros. Con lo que para tratarlos se opta por emplear la misma técnica que para los nulos, por la misma razón.

```
# Tratamiento de 0s
colSums(myDataAux == 0)
```

```
##               name               club                age
##                  0                  0                  0
##          height_cm          weight_kg        nationality
##                  0                  0                  0
##          eur_value           eur_wage eur_release_clause
##                259                253                  0
##            overall          potential                pac
##                  0                  0                  0
##                sho                pas                dri
##                  0                  0                  0
##                def                phy           crossing
##                  0                  0                  0
##          finishing   heading_accuracy      short_passing
##                  0                  0                  0
##            volleys          dribbling              curve
##                  0                  0                  0
## free_kick_accuracy       long_passing       ball_control
##                  0                  0                  0
##       acceleration       sprint_speed            agility
##                  0                  0                  0
##          reactions            balance         shot_power
##                  0                  0                  0
##            jumping            stamina           strength
##                  0                  0                  0
##         long_shots         aggression      interceptions
##                  0                  0                  0
##        positioning             vision          penalties
##                  0                  0                  0
##          composure            marking    standing_tackle
##                  0                  0                  0
##     sliding_tackle
##                  0
```

```
myDataAux$eur_value[myDataAux$eur_value == 0] <- NA
myDataAux$eur_wage[myDataAux$eur_wage == 0] <- NA
myDataAux$eur_value <- kNN(myDataAux)$eur_value
myDataAux$eur_wage <- kNN(myDataAux)$eur_wage
```

Por úlitmo, queda procesar los registros en los que hay valores vacíos. Con lo que en primer lugar, tal y como se ha estado haciendo hasta ahora, se listan todos los atributos observando si contienen valores vacios o no, se observa que la columna que hace referencia al equipo de cada jugador contiene valores vacíos, se da esta problemática cuando los jugadores están libres y actualmente no pertenecen a ningún equipo, con lo que se ha optado por sustituir el campo vacio por el valor de "Libre".

```
# Tratamiento de valores vacios
colSums(myDataAux == "")
```

```
##               name               club                age
##                  0                253                  0
```

```
##          height_cm          weight_kg        nationality
##                  0                  0                  0
##          eur_value          eur_wage eur_release_clause
##                  0                  0                  0
##            overall          potential                pac
##                  0                  0                  0
##                sho                pas                dri
##                  0                  0                  0
##                def                phy           crossing
##                  0                  0                  0
##          finishing   heading_accuracy      short_passing
##                  0                  0                  0
##            volleys          dribbling              curve
##                  0                  0                  0
## free_kick_accuracy       long_passing       ball_control
##                  0                  0                  0
##       acceleration       sprint_speed            agility
##                  0                  0                  0
##          reactions            balance         shot_power
##                  0                  0                  0
##            jumping            stamina           strength
##                  0                  0                  0
##         long_shots         aggression      interceptions
##                  0                  0                  0
##        positioning             vision          penalties
##                  0                  0                  0
##          composure            marking    standing_tackle
##                  0                  0                  0
##     sliding_tackle
##                  0
```

```
myDataAux$club <- as.character(myDataAux$club)
myDataAux$club[myDataAux$eur_release_clause == ""] <- "Libre"
myDataAux$club <- as.factor(myDataAux$club)
```

Como punto extra en este apartado, se ha considerado necesario tratar como Numeric los valores que el programa había recogido como Integers, debido a que son variables cuantitativas cuntinuas y han de tener este formato. Esto se realiza con el bucle que se aprecia a en el siguiente bloque, donde recorrerá cada una de las columnas comprobando el tipo de la misma y si detecta que es de tipo Integer será convertida a Numeric. Finalmente se muestra el tipo de cada uno de los atributos de este conjunto de datos junto con alguno de los registros que pueden llegar a tomar.

```
# Transformación de integer a numerico
col.names = colnames(myDataAux)
for (i in 1:ncol(myDataAux)) {
    if (is.integer(myDataAux[, i])) {
        myDataAux[, i] <- as.numeric(myDataAux[, i])
    }
}

str(myDataAux)
```

```
## 'data.frame':    17994 obs. of  46 variables:
##  $ name              : Factor w/ 17022 levels "A. Ã-hman","A. Ã-mÃ¼r",..: 3219 9671 12459 9873 11203
##  $ club              : Factor w/ 648 levels "","" SSV Jahn Regensburg",..: 472 226 437 226 229 229 38
```

```
##  $ age               : num  32 30 25 30 31 28 26 26 27 29 ...
##  $ height_cm         : num  185 170 175 182 193 185 193 173 182 184 ...
##  $ weight_kg         : num  80 72 68 86 92 79 76 76 78 87 ...
##  $ nationality       : Factor w/ 164 levels "Afghanistan",..: 121 6 19 158 58 120 138 13 58 6 ...
##  $ eur_value         : num  9.55e+07 1.05e+08 1.23e+08 9.70e+07 6.10e+07 9.20e+07 6.45e+07 9.05e+07 
##  $ eur_wage          : num  565000 565000 280000 510000 230000 355000 215000 295000 340000 275000 ..
##  $ eur_release_clause: num  1.96e+08 2.15e+08 2.37e+08 1.99e+08 1.01e+08 ...
##  $ overall           : num  94 93 92 92 92 91 90 90 90 90 ...
##  $ potential         : num  94 93 94 92 92 91 92 91 90 90 ...
##  $ pac               : num  90 89 92 82 91 81 90 90 56 79 ...
##  $ sho               : num  93 90 84 90 90 88 85 82 81 87 ...
##  $ pas               : num  82 86 79 79 95 75 87 84 89 70 ...
##  $ dri               : num  90 96 95 87 89 86 90 92 81 83 ...
##  $ def               : num  33 26 30 42 60 38 58 32 73 25 ...
##  $ phy               : num  80 61 60 81 91 82 86 66 70 74 ...
##  $ crossing          : num  85 77 75 77 15 62 17 80 85 68 ...
##  $ finishing         : num  94 95 89 94 13 91 13 83 76 91 ...
##  $ heading_accuracy  : num  88 71 62 77 25 85 21 57 54 86 ...
##  $ short_passing     : num  83 88 81 83 55 83 50 86 90 75 ...
##  $ volleys           : num  88 85 83 88 11 87 13 79 82 88 ...
##  $ dribbling         : num  91 97 96 86 30 85 18 93 79 84 ...
##  $ curve             : num  81 89 81 86 14 77 21 82 85 74 ...
##  $ free_kick_accuracy: num  76 90 84 84 11 84 19 79 84 62 ...
##  $ long_passing      : num  77 87 75 64 59 65 51 81 93 59 ...
##  $ ball_control      : num  93 95 95 91 48 89 42 92 89 85 ...
##  $ acceleration      : num  89 92 94 88 58 79 57 93 60 78 ...
##  $ sprint_speed      : num  91 87 90 77 61 83 58 87 52 80 ...
##  $ agility           : num  89 90 96 86 52 78 60 93 71 75 ...
##  $ reactions         : num  96 95 88 93 85 91 88 85 86 88 ...
##  $ balance           : num  63 95 82 60 35 80 43 91 69 69 ...
##  $ shot_power        : num  94 85 80 87 25 88 31 79 87 88 ...
##  $ jumping           : num  95 68 61 69 78 84 67 59 32 79 ...
##  $ stamina           : num  92 73 78 89 44 79 40 79 77 72 ...
##  $ strength          : num  80 59 53 80 83 84 64 65 74 85 ...
##  $ long_shots        : num  92 88 77 86 16 83 12 82 90 82 ...
##  $ aggression        : num  63 48 56 78 29 80 38 54 60 50 ...
##  $ interceptions     : num  29 22 36 41 30 39 30 41 85 20 ...
##  $ positioning       : num  95 93 90 92 12 91 12 85 79 92 ...
##  $ vision            : num  85 90 80 84 70 78 68 86 88 70 ...
##  $ penalties         : num  85 78 81 85 47 84 40 86 73 70 ...
##  $ composure         : num  95 96 92 83 70 87 64 87 85 86 ...
##  $ marking           : num  22 13 21 30 10 25 13 25 63 12 ...
##  $ standing_tackle   : num  31 28 24 45 10 42 21 27 82 22 ...
##  $ sliding_tackle    : num  23 26 33 38 11 19 13 22 69 18 ...
```

3.2. Identificación y tratamiento de valores extremos

En este apartado se detectarán los valores extremos que dispone en conjunto de datos en los campos de "overall", "eur_value", "eur_wage" y "eur_release_clause". Los valores extremos o *outliers* son aquellos que se encuantran fuera de los esperados, ya sea porque son muy altos o muy bajos. A continuación se muestra un diagrama *boxplot* para cada uno de los atributos.

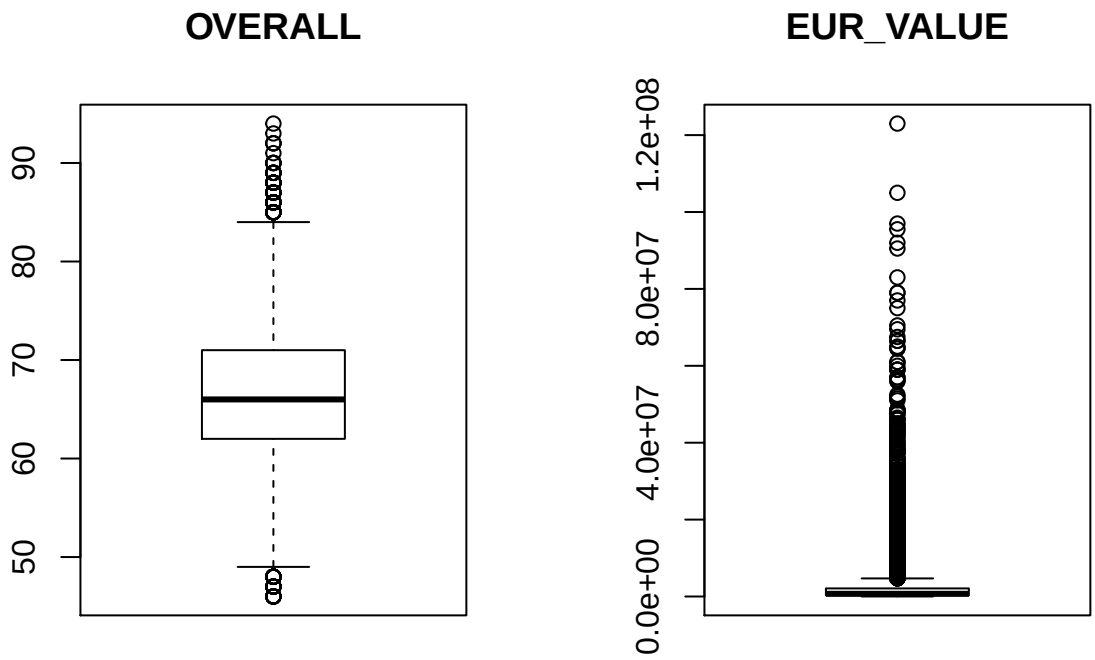
```r
# Comentario
par(mfrow = c(1, 2))

boxplot(myDataAux$overall, main = "OVERALL")
overallAti <- boxplot.stats(myDataAux$overall)$out
summary(myDataAux$overall)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   46.00   62.00   66.00   66.25   71.00   94.00
```

```r
length(overallAti)
```

```
## [1] 142
```

```r
boxplot(myDataAux$eur_value, main = "EUR_VALUE")
```




```r
eur_valueAti <- boxplot.stats(myDataAux$eur_value)$out
summary(myDataAux$eur_value)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##     10000    325000    700000   2393358   2100000 123000000
```

```r
length(eur_valueAti)
```

```
## [1] 2411
```

```r
par(mfrow = c(1, 2))
boxplot(myDataAux$eur_wage, main = "EUR_WAGE")
eur_wageAti <- boxplot.stats(myDataAux$eur_wage)$out
```

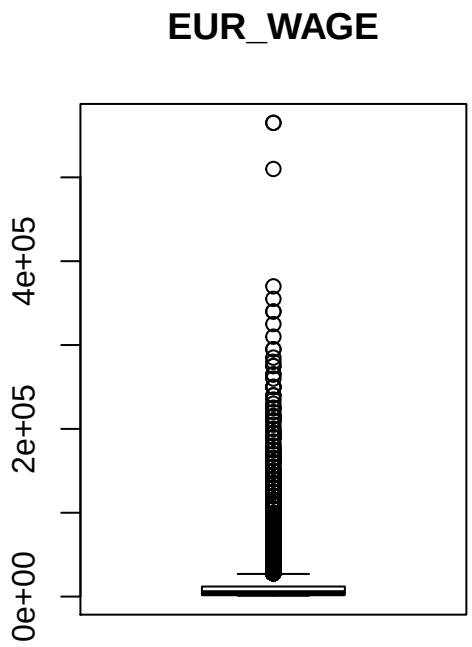
```
summary(myDataAux$eur_wage)
```

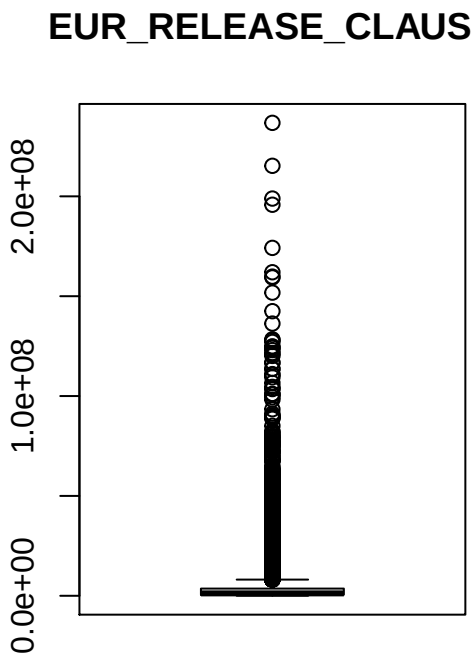
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1000    2000    4000   11610   12000  565000
```

```
length(eur_wageAti)
```

```
## [1] 1899
```

```
boxplot(myDataAux$eur_release_clause, main = "EUR_RELEASE_CLAUSE")
```






```
eur_release_clauseAti <- boxplot.stats(myDataAux$eur_release_clause)$out
summary(myDataAux$eur_release_clause)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##     13000    546000   1200000   4362880   3600000 236800000
```

```
length(eur_release_clauseAti)
```

```
## [1] 2460
```

En primer lugar se analizarán los valores extremos para el atributo "overall". Esta columna consta de 142 **outliers** y se corresponde con los jugadores que tienen una puntiación media superior a 845 ($Q3 + 15 \times IQR$) e inferior a 485 ($Q1 - 15 \times IQR$). En segundo lugar, se encuentra el atributo "eur_value", que por su parte tiene un mayor número de valores extremos, 2411, debido a las grandes diferencias que hay entre el coste de los jugadores. En este caso los valores extremos se encuentran cuando toman un valor superior a 4762500 e inferior a 0, con lo que no habrá ningun **outlier** por abajo. En tercer lugar, se analizan los valores extremos del atributo "eur_wage", que consta de un total de 1899 valores, dándose cuando el salario del jugador es supeior a 30000 e inferior a 0, por lo que sucederá lo mismo que para el atributo "eur_value" que no podía

haber valores extremos por abajo. Por úlitmo, queda analizar los **outliers** del atributo "eur_release_clause", que consta de 2460 valores extremos, apareciendo cuando la cláusula de rescisión del jugador es superior a 9000000 e inferior a 0, tal y como sucedía con los atributos anteriores.

4. Análisis de los datos

4.1. Selección de los grupos de datos a analizar

En este apartado se dividirán los registros del conjunto de datos en distintas agrupaciones en función de los valores que pueden llegar a tomar.

La primera selección de grupos de datos será en función de la puntuación media que el juego otorga a cada jugador, separando así a los mejores jugadores ($overall \geq 80$), a los jugadores standard ($80 > overall \geq 65$) y a los peores jugadores ($overall < 65$).

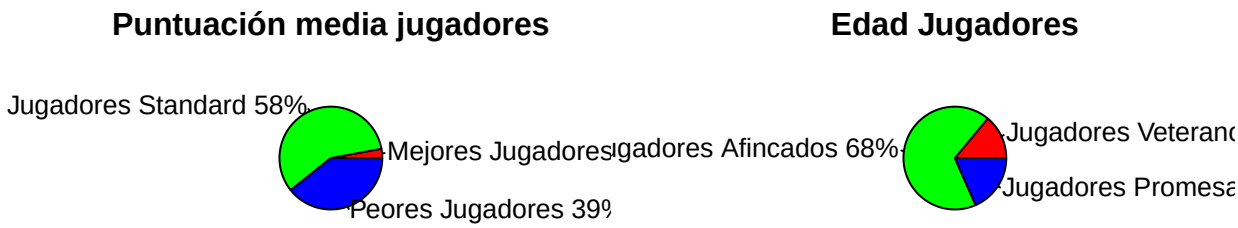
La segunda segmentación se llevará a cabo en función de la edad de los jugadores, creando para este caso también tres rangos, el de los jugadores de mayor edad ($age \geq 31$), el de los jugadores de edad media ($31 > age \geq 21$) y el de los jugadores más jóvenes ($age < 21$).

En tercer lugar, se hará una selección de grupos en base a las cláusulas de rescisión de los jugadores, separando los que tiene cláusulas más altas ($eur\_release\_clause \geq 30000000$), cláusulas medias ($30000000 > eur\_release\_clause \geq 5000000$) y cláusulas bajas ($eur_release_clause < 5000000$).

Por último, se hacen dos selecicones de grupos que son un tanto distintas al resto debido a que no emplean todo el conjunto de datos, y serán empleadas más adelante en la práctica. La primera de ellas diferencia a los jugadores en función de su nacionalidad, pero simplemente entre argentinos y brasileños. La segunda por su parte, es similar a la primera solo que la agrupación se realiza en base al club al que pertenece el jugador, seleccionando únicamente los jugadores del FC Barcelona y del Real Madrid CF.

```
par(mfrow = c(2, 2))
# Agrupación en función de la media de cada jugador
myDataAux.Mejores <- myDataAux[myDataAux$overall >= 80,
    ]
myDataAux.Standars <- myDataAux[myDataAux$overall < 80 &
    myDataAux$overall >= 65, ]
myDataAux.Peores <- myDataAux[myDataAux$overall < 65, ]
# Representación gáfica
slices <- c(nrow(myDataAux.Mejores), nrow(myDataAux.Standars),
    nrow(myDataAux.Peores))
lbls <- c("Mejores Jugadores", "Jugadores Standard", "Peores Jugadores")
pct <- round(slices/sum(slices) * 100)
lbls <- paste(lbls, pct)
lbls <- paste(lbls, "%", sep = "")
pie(slices, labels = lbls, col = rainbow(length(lbls)),
    main = "Puntuación media jugadores")

# Agrupación en funcion la edad de cada jugador
myDataAux.Veteranos <- myDataAux[myDataAux$age >= 31, ]
myDataAux.Afincados <- myDataAux[myDataAux$age < 31 & myDataAux$age >=
    21, ]
myDataAux.Promesas <- myDataAux[myDataAux$age < 21, ]
# Representación gáfica
slices <- c(nrow(myDataAux.Veteranos), nrow(myDataAux.Afincados),
    nrow(myDataAux.Promesas))
lbls <- c("Jugadores Veteranos", "Jugadores Afincados",
```

```
    "Jugadores Promesas")
pct <- round(slices/sum(slices) * 100)
lbls <- paste(lbls, pct)
lbls <- paste(lbls, "%", sep = "")
pie(slices, labels = lbls, col = rainbow(length(lbls)),
    main = "Edad Jugadores")

# Agrupación en función de la clausula de rescisión de
# cada jugador
myDataAux.Alta <- myDataAux[myDataAux$eur_release_clause >=
    3e+07, ]
myDataAux.Media <- myDataAux[myDataAux$eur_release_clause <
    3e+07 & myDataAux$eur_release_clause >= 5e+06, ]
myDataAux.Baja <- myDataAux[myDataAux$eur_release_clause <
    5e+06, ]
# Representación gáfica
slices <- c(nrow(myDataAux.Alta), nrow(myDataAux.Media),
    nrow(myDataAux.Baja))
lbls <- c("Alta", "Media", "Baja")
pct <- round(slices/sum(slices) * 100)
lbls <- paste(lbls, pct)
lbls <- paste(lbls, "%", sep = "")
pie(slices, labels = lbls, col = rainbow(length(lbls)),
    main = "Cláusula rescición jugadores")

# Agrupación en función de la nacionalidad
myDataAux.Argentinos <- myDataAux[myDataAux$nationality ==
    "Argentina", ]
myDataAux.Brasilenos <- myDataAux[myDataAux$nationality ==
    "Brazil", ]

# Agrupación en función del club
myDataAux.Barcelona <- myDataAux[myDataAux$club == "FC Barcelona",
    ]
myDataAux.Madrid <- myDataAux[myDataAux$club == "Real Madrid CF",
    ]
```



Tras realizar una representación gráfica de las agrupaciones que representan la puntuación media, la edad y la cláusula de rescisión de los jugadores se puede observar el porcentaje de jugadores que tiene cada uno de los grupos. Donde en las dos primeras agrupaciones de datos el mayor número de jugadores se encuentra en el rango medio, tal y como se podría esperar. No obstante, en la tercer agrupación, la de la cláusula de rescisión, el 80 % de los jugadores tiene lo que se ha considerado una cláusula de rescisión baja para lo que es la realidad.

4.2. Comprobación de la normalidad y homogeneidad de la varianza

La normalidad ha de calcularse solo con las variables numéricas, con lo que en este caso podrá ser calculada con todos los atributos del conjunto de datos excepto con los atributos “name”, “club” y “nationality”. Dicho esto, porcedemos a calcular la normalidad empleando la técnica de Anderson-Darlinng, con la que habrá que comparar el p-valor obtenido por la aplicación de esta técnica con un porcentaje de significación del 5 %, $\alpha = 0,05$, teniendo que ser el p-valor obtenido menor que α para que esta variable pueda ser considerada normal. Aunque existen diferentes test para determinar si un conjunto de datos sigue una distribución normal, se ha optado por el test comentado debido a que es el más adecuado cuando el dataset esta constituido por un gran número de instancias.

```
# Normalidad
library(nortest)
alpha = 0.05
col.names = colnames(myDataAux)
for (i in 1:ncol(myDataAux)) {
    if (i == 1)
        cat("Variables que no siguen una distribución normal:\n")
    if (is.numeric(myDataAux[, i])) {
```

```
        p_val = ad.test(myDataAux[, i])$p.value
        if (p_val < alpha) {
            cat(col.names[i], "| con p-valor:", p_val, "\n")
        }
    }
}
```

```
## Variables que no siguen una distribución normal:
## age | con p-valor: 3.7e-24
## height_cm | con p-valor: 3.7e-24
## weight_kg | con p-valor: 3.7e-24
## eur_value | con p-valor: 3.7e-24
## eur_wage | con p-valor: 3.7e-24
## eur_release_clause | con p-valor: 3.7e-24
## overall | con p-valor: 3.7e-24
## potential | con p-valor: 3.7e-24
## pac | con p-valor: 3.7e-24
## sho | con p-valor: 3.7e-24
## pas | con p-valor: 3.7e-24
## dri | con p-valor: 3.7e-24
## def | con p-valor: 3.7e-24
## phy | con p-valor: 3.7e-24
## crossing | con p-valor: 3.7e-24
## finishing | con p-valor: 3.7e-24
## heading_accuracy | con p-valor: 3.7e-24
## short_passing | con p-valor: 3.7e-24
## volleys | con p-valor: 3.7e-24
## dribbling | con p-valor: 3.7e-24
## curve | con p-valor: 3.7e-24
## free_kick_accuracy | con p-valor: 3.7e-24
## long_passing | con p-valor: 3.7e-24
## ball_control | con p-valor: 3.7e-24
## acceleration | con p-valor: 3.7e-24
## sprint_speed | con p-valor: 3.7e-24
## agility | con p-valor: 3.7e-24
## reactions | con p-valor: 3.7e-24
## balance | con p-valor: 3.7e-24
## shot_power | con p-valor: 3.7e-24
## jumping | con p-valor: 3.7e-24
## stamina | con p-valor: 3.7e-24
## strength | con p-valor: 3.7e-24
## long_shots | con p-valor: 3.7e-24
## aggression | con p-valor: 3.7e-24
## interceptions | con p-valor: 3.7e-24
## positioning | con p-valor: 3.7e-24
## vision | con p-valor: 3.7e-24
## penalties | con p-valor: 3.7e-24
## composure | con p-valor: 3.7e-24
## marking | con p-valor: 3.7e-24
## standing_tackle | con p-valor: 3.7e-24
## sliding_tackle | con p-valor: 3.7e-24
```

Tras lograr el p-valor correspondiente de todas las variables numéricas, se puede determinar que ninguna de ellas sigue una distribución normal, puesto que sus p-valor son todos inferiores a $\alpha = 0,05$.

La homogeneidad será calculada entre mútiples pares de atributos, el primero de ellos formado por los atributos que representan la puntuación media de cada jugador junto con el el valor económico que el juego otorga a cada jugador, y el segundo, por su parte, con los atributos que representan el salario y club de cada jugador. Para ello, se empleará el F-test, tal y como se aprecia a continuación:

```
# Homogeneidad
fligner.test(overall ~ eur_value, data = myDataAux)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  overall by eur_value
## Fligner-Killeen:med chi-squared = 1734.6, df = 206, p-value <
## 2.2e-16
```

```
fligner.test(eur_wage ~ club, data = myDataAux)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  eur_wage by club
## Fligner-Killeen:med chi-squared = 10084, df = 647, p-value <
## 2.2e-16
```

En el caso en el que se mide la homegeniedad entre la puntuación media y valor económico el p-valor obtenido es de $22e^{-16}$, siendo mucho menor que 005, con lo que se considera que hay diferencias significativcas entre las varianzas de ambos grupos. Por otra parte, cuando la homegeneidad es medida entre el salario y el club de cada jugador, el p-valor resultante también es de $22e^{-16}$, por lo que como sucedía anteriormente si hay diferencias entre las varianzas de estos dos grupos.

4.3. Aplicación de pruebas estadísticas

A continuación, se va a proceder a realizar un análisis estadístico, en el que se efecuarán:

- Contraste de hipótesis.
- Regreseión lineal Múltiple.
- Regresión logística.

4.3.1. Contraste de hipótesis

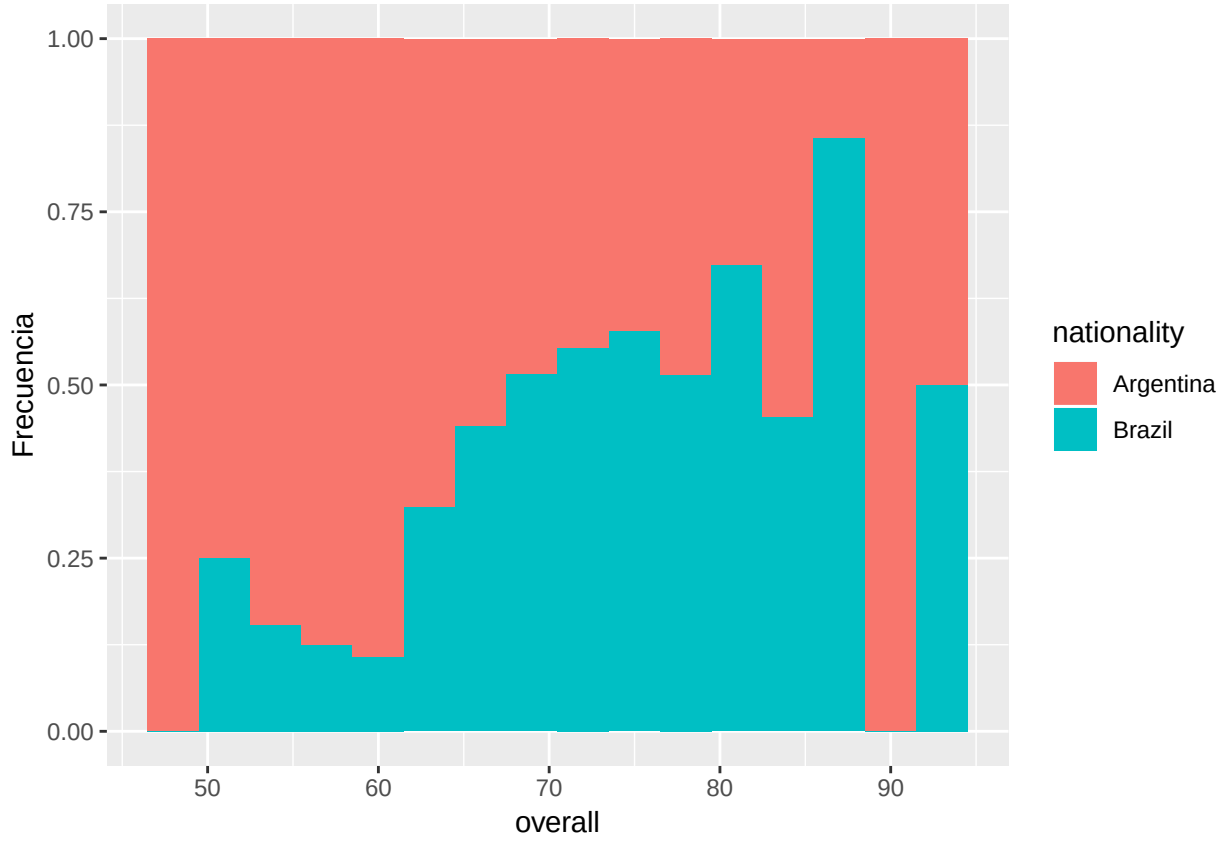
En este estudio, se van a plantear dos contrastes de hipótesis.

1) Se pretende analizar, si los jugadores de nacionalidad argentina tienen mayor puntuación media (overall) que los brasileños. Para ello se emplearán los grupos de datos "myDataAux.Argentinos" y "myDataAux.Brasilenos" creados anteriormente. En esta ocasión, se procede a comparar estas dos nacionalidades por el interés que puede generar la rivalidad Argentina vs Brasileña, pero se podria realizar con diferentes nacionalidades de la misma manera, en lo que variará el resultado que se obtenga.

Para tener un conocimiento previo al contraste de hipótesis los datos de estas dos nacionalides, se muestra en el siguiente gráfico, la relación entre las diferentes puntuaciones de los jugadores de ambas nacionalidades:

```
#Representación gáfica de las puntuaciones medias de argentinos y brasileños
library(ggplot2)
aux <- myDataAux[myDataAux$nationality=="Argentina" | myDataAux$nationality=="Brazil",]
aux$nationality <- as.character(aux$nationality)
```

```r
aux$nationality <- as.factor(aux$nationality)
ggplot(data = aux[!is.na(aux[1:nrow(aux),]$overall),],aes(x=overall,fill=nationality))+geom_histogram(b
```



Se procede con el contraste de hipótesis de dos muestras sobre la diferencia de medias:

Se establece la hipótesis nula y alternativa:

Hipótesis nula: $H_0 : \mu_1 - \mu_2 = 0$

Hipótesis Alternativa: $H_1 : \mu_1 - \mu_2 < 0$

Como se observa, las hipótesis son unilaterales, donde μ_1 representa la media de la población de la que se extráe la primera muestra (nacionalidad argentina), y μ_2, por su parte, la media de la población de la segunda muestra (nacionalidad brasileña). La hipótesis nula, representa que los jugadores de nacionalidad argentina y brasileña, tienen una puntución media (overall) similar. En cambio, la hipótesis alternativa, representa, que los jugadores argentinos, tienen una puntuación inferior.

Se fija el valor de significación $\alpha = 005$ y se procede a relizar el test, haciendo uso de la función **t.test** que facilita R.

Se aplica el test

```r
t.test(myDataAux.Argentinos$overall, myDataAux.Brasilenos$overall,
    alternative = "less")
```

```
##
##  Welch Two Sample t-test
##
## data:  myDataAux.Argentinos$overall and myDataAux.Brasilenos$overall
## t = -11.169, df = 1764.8, p-value < 2.2e-16
```

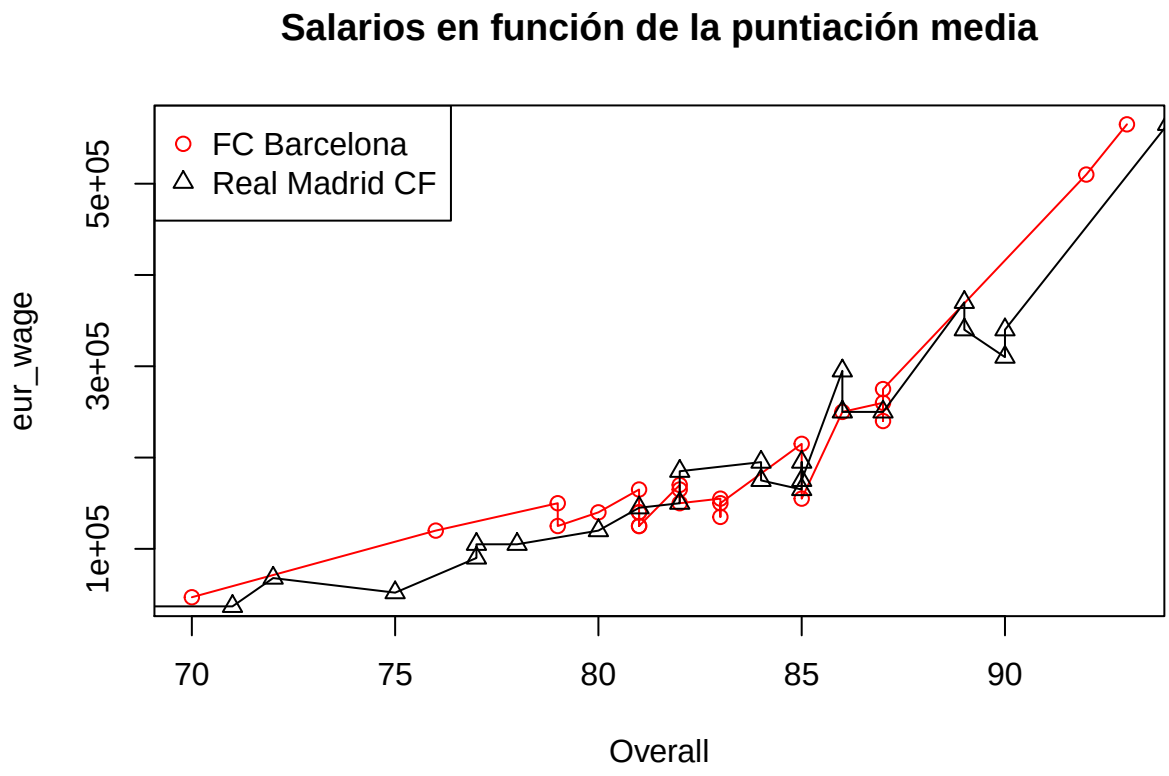
```
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##       -Inf -2.675422
## sample estimates:
## mean of x mean of y
##  67.71830  70.85608
```

En base a los resultados, se rechaza la hipótesis nula, puesto que el p-valor obtenido es significativamente inferior a $\alpha = 005$. Además, se aprecia que la puntuación media de los jugadores argentinos es de 6772 , mientras que los brasileños poseen una media de 7086

2) En el sigueinte contraste de hipótesis, se pretende analizar si los jugadores del F.C. Barcelona tienen salarios superiores que los jugadores del Real Madrid. En este caso se emplearán las agrupaciones de datos "myDataAux.Barcelona" y "myDataAux.Madrid" generadas en apartados anteriores. Al igual que sucedía en el apartado anterior, este analisis puede realizarse con diferentes clubes.

Antes de comenzar con el contraste de hipótesis, se procede a visualizar la relación de los salarios segun la puntuación de los jugadores y el club.

```r
# Representación gáfica de los salarios de cada
# equipo en función de la puntuación media de cada
# jugador

overallsBarc <- c(myDataAux.Barcelona$overall)
overallsMad <- c(myDataAux.Madrid$overall)

salarioBarc <- c(myDataAux.Barcelona$eur_wage)
salarioMad <- c(myDataAux.Madrid$eur_wage)

plot(overallsBarc, salarioBarc, type = "overplotted",
    pch = 1, col = "red", xlab = "Overall", ylab = "eur_wage",
    main = "Salarios en función de la puntiación media")

lines(overallsMad, salarioMad, type = "overplotted",
    pch = 2, col = "black")

legend("topleft", legend = c("FC Barcelona", "Real Madrid CF"),
    pch = c(1, 2), col = c("red", "black"))
```



El planteamiento del contraste de hipótesis sigue el mismo formato que el anterior:

Se define la hipótesis nula y alternativa:

Hipótesis nula: $H_0 : \mu_1 - \mu_2 = 0$

Hipótesis Alternativa: $H_1 : \mu_1 - \mu_2 < 0$

En este caso, μ_1 representa el salario de los jugadores del FC Barcelona y μ_2 los salarios de los jugadores del Real Madrid CF, el contraste también es unilateral atendiendo a la formulación de la hipótesis, y se fija un nivel de confianza del 95 %, es decir, un nivel de significación de $\alpha = 005$.

Se aplica el test

```
t.test(myDataAux.Barcelona$eur_wage, myDataAux.Madrid$eur_wage,
    alternative = "less")
```

```
##
##  Welch Two Sample t-test
##
## data:  myDataAux.Barcelona$eur_wage and myDataAux.Madrid$eur_wage
## t = 0.68615, df = 49.933, p-value = 0.7521
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf 82087.85
## sample estimates:
## mean of x mean of y
##  194666.7  170821.4
```

Con lo que tras haber llevado a cabo este contraste de hipótesis se acepta la hipótesis nula, debido a que el

p-valor obtenido, $0,7521$ es mayor que $\alpha = 005$.

4.3.2. Regresión lineal

En este apartado, se procede a analizar la relación existente entre diferentes variables del conjunto de datos. Como se ha mencionado en el apartado de descripción de los datos, el conjunto tiene una gran antidad de variables, y por esta razón, para realizar la regresión lineal, se ha decidido utilizar las siguientes variables, según el criterio de los desarrolladores de la práctica, dado que, a modo de ver son variables que han de tener relación entre sí:

Regresión Lineal Variable Overall

Se va a estudiar el valor de la variable explicada ($Y = overal$) en función de:

- Primer Modelo:
  - Variable explicada $Y = overall$.
  - variables explicativas $X = \{pac, sho, pas, dri, def, phy\}$.
- Segundo Modelo:
  - Variable explicada $Y = overall$.
  - variables explicativas $X = \{pas, phy\}$.
- Tercer Modelo:
  - Variable explicada $Y = overall$.
  - variables explicativas $X = \{acceleration, finishing, sprint\_speed, stamina, long\_shots, vision, positioning\}$.

Para generar los distintos modelos, se hace uso de la función **lm** que proporciona R.

```
# Creación de modelos de regresión lineal de overall

modelOverall_1 <- lm(overall ~ pac + sho + pas + dri + def +
    phy, data = myDataAux)
modelOverall_2 <- lm(overall ~ pas + phy, data = myDataAux)
modelOverall_3 <- lm(overall ~ acceleration + finishing +
    sprint_speed + stamina + long_shots + vision + positioning,
    data = myDataAux)
```

A continuación, se compara el resultado obtenido de los modelos, para escoger entre ellos, el que mejor resultado obtenga segun el coeficiente de determinacion "R2".

```
# Resultados de los modelos de regresión lineal de
# overall
resultados <- matrix(c(1, summary(modelOverall_1)$r.squared,
    2, summary(modelOverall_2)$r.squared, 3, summary(modelOverall_3)$r.squared),
    ncol = 2, byrow = TRUE)

colnames(resultados) <- c("Modelo", "R^2")

resultados
```

```
##      Modelo       R^2
## [1,]      1 0.7106128
## [2,]      2 0.6342957
## [3,]      3 0.2760295
```

Como se aprecia en los resultados, el coeficiente de determinación (R2) que obtiene el mayor valor, corresponde al modelo de regresión:

- Primer Modelo:
  - Variable explicada $Y = overall$.

- variables explicativas $X = \{pac, sho, pas, dri, def, phy\}$.

A continuación, se observan los resultados de este modelo:

```
# Resumen modelo1
summary(modelOverall_1)
```

```
##
## Call:
## lm(formula = overall ~ pac + sho + pas + dri + def + phy, data = myDataAux)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.7518  -2.6600  -0.3657   2.3599  16.2912
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 14.781571   0.280079  52.776  < 2e-16 ***
## pac          0.025274   0.003321   7.609 2.89e-14 ***
## sho          0.103896   0.004180  24.858  < 2e-16 ***
## pas          0.069194   0.005561  12.443  < 2e-16 ***
## dri          0.259437   0.006553  39.593  < 2e-16 ***
## def          0.116802   0.002832  41.250  < 2e-16 ***
## phy          0.281202   0.003764  74.699  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.738 on 17987 degrees of freedom
## Multiple R-squared:  0.7106, Adjusted R-squared:  0.7105
## F-statistic:  7361 on 6 and 17987 DF,  p-value: < 2.2e-16
```

Con los resultados del modelo se identifica lo siguiente:

- Todos los coeficientes estimados son significativos.
- Los coeficientes estimados, son significativos con un nivel de significancia del 01 % ('***'0.001)
- El coeficiente de determinación R2 tiene un valor de: 07106 y el ajustado de 07105. Como se sabe que el valor de R2 está: $0 < R2 < 1$, cuanto mas cercano sea a 1, mayor es la proporción de variabilidad de la variable explicada (Y) por el modelo, y por tanto, mayor será la bondad del ajuste. En este caso, se puede comentar que el modelo de regresión múltiple generado explica el 7106 % de la variabilidad de la variable overall de cada jugador. El valor de R2-ajustado es alto, y similar al de R2, lo que nos indica que el modelo tiene predictores útiles, aun así, cuanto mayor sea este valor mejor seria el modelo.

Regresión Lineal Variable sho

Se va a estudiar el valor de la variable explicada $(Y = sho)$ en función de:

- Primer Modelo:
  - Variable explicada $Y = sho$.
  - variables explicativas $X = \{finishing, heading\_accuracy, free\_kick\_accuracy, shot\_power, long\_shots, penalti$
- Segundo Modelo:
  - Variable explicada $Y = sho$.
  - variables explicativas $X = \{finishing, shot\_power\}$.
- Tercer Modelo:
  - Variable explicada $Y = sho$.
  - variables explicativas $X = \{aggression, strength, interceptions\}$

Para generar los distintos modelos, se hace uso de la función **lm** que proporciona R.

```r
# Creación de modelos de regresión lineal de sho

modelShotting_1 <- lm(sho ~ finishing + heading_accuracy +
    free_kick_accuracy + shot_power + long_shots + penalties,
    data = myDataAux)
modelShotting_2 <- lm(sho ~ finishing + shot_power + long_shots,
    data = myDataAux)
modelShotting_3 <- lm(sho ~ aggression + strength + interceptions,
    data = myDataAux)
```

A continuación, se compara el resultado obtenido de los modelos, para escoger entre ellos, el que mejor resultado obtenga según el coeficiente de determinación "R2".

```r
# Resultados de los modelos de regresión lineal de sho
resultados <- matrix(c(1, summary(modelShotting_1)$r.squared,
    2, summary(modelShotting_2)$r.squared, 3, summary(modelShotting_3)$r.squared),
    ncol = 2, byrow = TRUE)

colnames(resultados) <- c("Modelo", "R^2")

resultados
```

```
##      Modelo       R^2
## [1,]      1 0.6065040
## [2,]      2 0.3583834
## [3,]      3 0.2139681
```

Como se observa en el resultado, el coeficiente de determinación (R2) que obtiene el mayor valor, corresponde al modelo de regresión:

- Primer Modelo:
  - Variable explicada $Y = sho$.
  - variables explicativas $X = \{finishing, heading\_accuracy, free\_kick\_accuracy, shot\_power, long\_shots, penalti$

A continuación, observamos los resultados de este modelo:

```r
# Resumen modelo1
summary(modelShotting_1)
```

```
##
## Call:
## lm(formula = sho ~ finishing + heading_accuracy + free_kick_accuracy +
##     shot_power + long_shots + penalties, data = myDataAux)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -23.179  -6.099  -0.713   5.257  43.049
##
## Coefficients:
##                     Estimate Std. Error  t value Pr(>|t|)
## (Intercept)        47.606527   0.250547  190.011   <2e-16 ***
## finishing           0.400602   0.007619   52.582   <2e-16 ***
## heading_accuracy   -0.500677   0.004862 -102.980   <2e-16 ***
## free_kick_accuracy -0.130836   0.006505  -20.113   <2e-16 ***
## shot_power          0.158871   0.008848   17.955   <2e-16 ***
## long_shots          0.211117   0.009422   22.407   <2e-16 ***
## penalties           0.015807   0.008512    1.857   0.0633 .
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.679 on 17987 degrees of freedom
## Multiple R-squared:  0.6065, Adjusted R-squared:  0.6064
## F-statistic:  4621 on 6 and 17987 DF,  p-value: < 2.2e-16
```

Con los resultados del modelo se identificar lo siguiente:

- Todos los coeficientes estimados son significativos.
- Los coeficientes estimados, son significativos con un nivel de significancia del 01 % ('***'0.001) y el coeficiente estimado de la variable penalties, tiene un nivel de significancia del 10 % ('.'0.1)
- El coeficiente de determinación R2 tiene un valor de: 06065 y el ajustado de 06064. Como se sabe que el valor de R2 está: $0 < R2 < 1$, cuanto mas cercano sea a 1, mayor es la proporción de variabilidad de la variable explicada (Y) por el modelo, y por tanto, mayor será la bondad del ajuste. En este caso, se puede comentar que el modelo de regresión múltiple generado explica el 6065 % de la variabilidad de la variable sho de cada jugador. El valor de R2-ajustado es superior a 05 , y similar al de R2, lo que nos indica que el modelo tiene predictores útiles.

4.3.3. Regresión logística

Para generar un modelo de regresión logística, se establece una variable dependiente binaria, es decir, que tome el valor 0 o 1 , en función del valor que pueda llegar a tener la propia variable. En el caso del dataset que se dispone, no nos econtramos con ninguna variable que nos permita realizar un estudio correcto a traves de una regresión logística. Por ello, para aplicar este tipo de regresión, basándonos en el valor de la varaible overall (puntuacion media del jugador), se va a generar una variable binaria que indique si un jugador es de calidad superior o inferior. Es decir, si la puntuacion media del jugador supera los 70 puntos, se esablece como que es un jugador de calidad "1", en cambio, si el jugador tiene una puntuación inferior a 70 se le asignará un "0".

Generamos la nueva varaible binaria y la introducimos en el dataset.

```
# Generación de variable binaria
calidad <- ifelse(myDataAux$overall > 70, 1, 0)
myDataAux <- cbind(myDataAux, calidad)
myDataAux$calidad <- as.factor(myDataAux$calidad)
table(myDataAux$calidad)
```

```
##
##     0     1
## 13169  4825
```

Como se puede observar, se obtienen 13169 jugadores de baja calidad y 4825 de calidad alta. A continuación se procede con la generación de los modelos de regresión logística:

Como en el caso de la regresión lineal múltiple, se escogen diferentes variables explicativas con las que estudiar la variable explicada.

- modelo1:
  - Variable Explicada $Y = calidad$.
  - Varaibles explicativas $X = \{pas, phy\}$.
- modelo2:
  - Variable Explicada $Y = calidad$.
  - Varaibles explicativas $X = \{pas, phy, eur\_release\_clause\}$.
- modelo3:
  - Variable Explicada $Y = calidad$.

- Varaibles explicativas $X = \{pas, phy, eur\_release\_clause, club\}$.

Para generar los distintos modelos, se hace uso de la funcion **glm** que proporciona R, y se indica la "family" a la funcion, en este caso **binomial**.

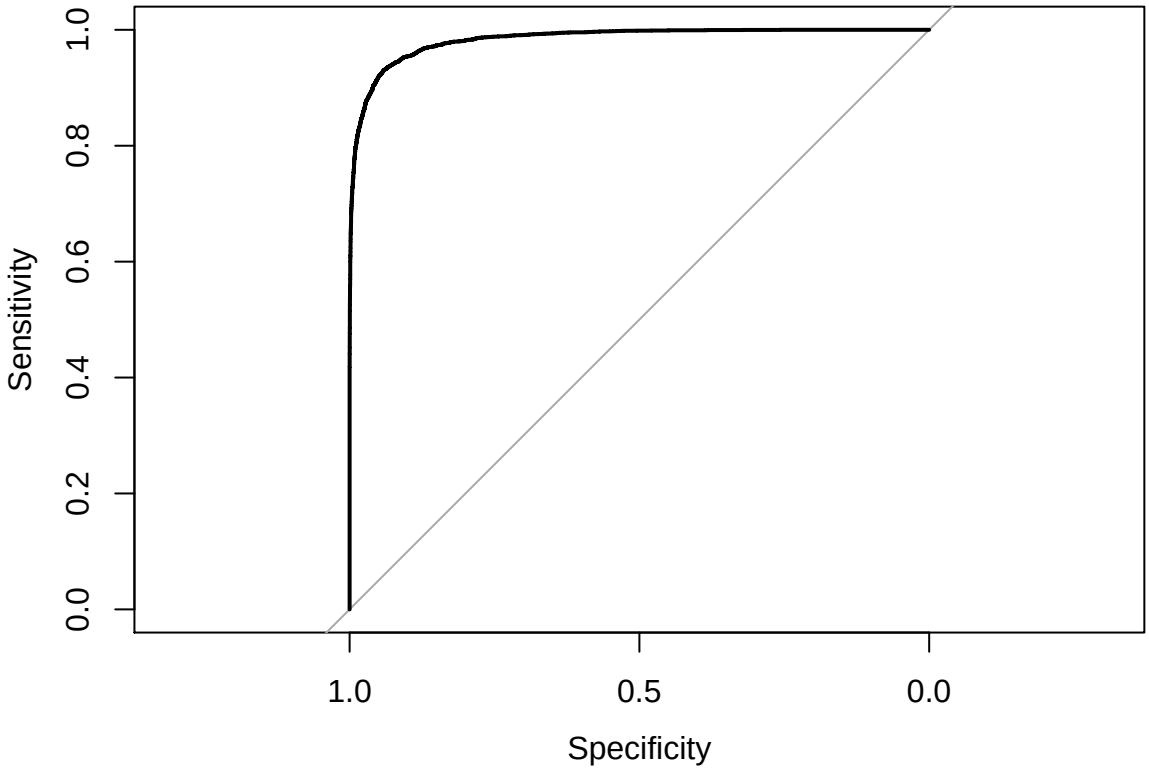
```
# Creación de modelos
modeloLog1 <- glm(calidad ~ pas + phy, data = myDataAux,
    family = "binomial")
modeloLog2 <- glm(calidad ~ pas + phy + eur_release_clause,
    data = myDataAux, family = "binomial")
modeloLog3 <- glm(calidad ~ pas + phy + eur_release_clause +
    club, data = myDataAux, family = "binomial")

resultados <- matrix(c(1, summary(modeloLog1)$aic, 2, summary(modeloLog2)$aic,
    3, summary(modeloLog3)$aic), ncol = 2, byrow = TRUE)
colnames(resultados) <- c("Modelo", "AIC")
```

A partir de los valores AIC obtenidos en los distintos modelos de regresión logística, se procede a seleccionar el que mejor resultados nos aporta, para a continuación, mostrar la curva roc correspondiente. Esta selección, se realiza en base al criterio del menor valor posible obtenido en el campo AIC.

Como se puede observar, el modelo2, que contiene las variables: Variable Explicada ($Y = calidad$) y varaibles explicativas $X = \{pas, phy, eur\_release\_clause\}$, es el que menor AIC, obtiene.

Por tanto, se visualiza la curva ROC correspondiente a este modelo y su valor AUC (area under curve) correspondiente. Para ello, se obtienen las predicciones según el modelo generado, es decir, con el mejor de los modelos logísticos se calcularán las probabilidades de que un registro determinado pueda tomar el valor 0 (puntuación media baja) o 1 (puntuación media alta):

```
# Obtenemos las predicciones
predicciones <- predict(modeloLog2, type = "response")
library(pROC)
# Se introducen los resultados de las probabilidades en
# una nueva columna del dataset.
myDataAux$prob = predicciones
# Se genera la curva Roc
g <- roc(calidad ~ prob, data = myDataAux)
# Visualizar la curva
plot(g)
```




```
# Funcion que ofrece el area bajo la curva
auc(g)
```

```
## Area under the curve: 0.9837
```

A través de la curva ROC, se evalúa la bondad de ajuste del modelo generado. Cada punto de la curva corresponde a un nivel de umbral de discriminación en la matriz de confusión. Es decir, se construyen todas las matrices desde un umbral del 1 % al 99 %.

Observando la gráfica que se ha obtenido, y teniendo en cuenta el valor AUC (**Area under the curve**), 09837, obtenido gracias al modelo generado, se puede deducir que la bondad de este es casi "perfecta", es decir, que dispone de una bondad alta. Por tanto, el modelo en cuestión es bastante bueno cuando se pretende conocer si un jugador es o no de calidad, haciendo uso de los datos $\{pas, phy, eur\_release\_clause\}$.

A continuación, se realiza una prueba de predicción empleando el modelo que ha ofrecido mejores resultados:

```
# Predicción
newData1 <- data.frame(pas = 79, phy = 50, eur_release_clause = 1e+06)
# pred2 <- predict(modeloLog2,newData1,type =
# 'response')
newData2 <- data.frame(pas = 80, phy = 80, eur_release_clause = 1.5e+07)
# pred1 <- predict(modeloLog2,newData2,type =
# 'response')


resultados <- matrix(c(1, predict(modeloLog2, newData1,
    type = "response"), 2, predict(modeloLog2, newData2,
    type = "response")), ncol = 2, byrow = TRUE)
```

```r
colnames(resultados) <- c("Num prueba", "Probabilidad de calidad alta")
resultados
```

```
##      Num prueba Probabilidad de calidad alta
## [1,]          1                   0.02294008
## [2,]          2                   0.99999996
```

Se han realizado dos predicciones, una primera en la cual los valores otorgados a las variables explicativas son relativamente bajos, y una segunda en la que estas variables toman valores más elevados. Con lo que tras generar estos datos de test, se considera que el primer caso tiene un 2 % de probabilidades de que se trate de un jugador de calidad, mientras que el segundo se puede garantizar con casi toda seguridad de que será un jugador de calidad alta, debido a que ha obtenido un 99 % de probabilidades de serlo.

5. Conclusiones

Tras realizar esta práctica, se han llevado a cabo diferentes pruebas estadísticas con el conjunto de datos "Fifa 18 More Complete Player Dataset", ya mencionado. Realizando análisis de las variables más intuitivas por las que está compuesto este dataset, con la intención de cumplir todos los objetivos propuestos en la actividad práctica.

Apoyándonos en la visualización de gráficos y tablas los resultados se han podido interpretar de forma más sencilla, debido a que los resultados que llegan a ofrecer ciertas herramientas pueden resultar un tanto confusos.

Mediante los contrastes de hipótesis sobre dos muestras realizados se ha conseguido determinar la diferencia que estas muestras tienen entre sí en base a unos atributos determinados. En este caso, se han analizado por un lado el nivel de los jugadores en función de su nacionalidad, y por otro lado, la diferencia de salarios en base al club al que pertence cada jugador.

La regresión lineal, ha permitido conocer qué atributos del conjunto de datos guardan mayor relación con las variables objetivo, de esta forma dando valor a estos atributos se podría llegar a conocer la variación en la variable estudiada en cuestión. En este caso se han estudiado dos variables, la puntución media del jugador y la puntuación media de disparo del jugador.

Como último análisis estadístico, se ha propuesto una regresión logística la cual permite conocer según ciertas características si un jugador es considerado de calidad alta o baja. Mediante los modelos conseguidos con esta regresión pueden llegar a realizarse predicciones, donde otorgando un valor a los atributos explicativos puede obtenerse una estimación de si este jugador es de calidad baja o alta.

Previos a estos análisis, ha sido necesario preprocesar el conjunto de datos. En primer lugar se ha realizado una eliminación de atributos que se consideraban innecesarios para el estudio. Posteriormente, se han tratado los registros que disponían de ceros, valores vacíos o valores nulos en alguno de sus atributos. Además, también se han estudiado los valores extremos de las principales columnas del conjunto de datos. Por último, se han realizado grupos de datos en función de los valores que toman determindos atributos.