UNIVERSIDAD NEBRIJA

MÁSTER DE ESTADÍSTICA APLICADA CON R SOFTWARE

ANÁLISIS DE LOS DATOS DE JUGADORES DE LA LIGA 2019-2020

AUTOR: Óscar Bartolomé Pato

DIRECTOR: Juan Luis López Garrancho

FECHA: 31-08-2020

ENTIDAD COLABORADORA:

MAXIMA formación

Resumen

En la actualidad, análisis de datos deportivos está despegando de forma muy rápida. En el mundo del fútbol en particular, es de sobra conocido que los equipos punteros están apostando por la analítica de datos, ya sea para el estudio del contrario, el diseño de nuevas tácticas o en el scouting de jugadores. Además, se está utilizando en otros ámbitos menos conocidos como el análisis del estado físico de los jugadores y en la prevención de lesiones.

La principal motivación de este trabajo es mezclar los conocimientos adquiridos en este máster con una de mis pasiones, el fútbol. En él, analizaré de diferentes maneras lo que ha sido la temporada 2019-2020 en La Liga.

Mi intención es que este estudio de datos deportivos sea el primero de muchos en mi carrera, ya que mi idea es darle continuidad a esta temática. Espero que os guste.

Agradecimientos

Agradecer a todo el equipo de Máxima Formación toda la ayuda brindada durante el desarrollo del máster y durante la realización de este trabajo de fin de máster, en especial a Juan Luis López por su tiempo, dedicación y rapidez en responder a las dudas presentadas.

Índice

Fuentes de datos

En este apartado se describe la procedencia de los datos que, tras un preprocesamiento, serán la base de nuestro análisis.

estadisticas_estandar_jugadores_liga_19_20.csv y estadisticas_diversas_jugadores_liga_19_20.csv

Datos estadísticos de todos los futbolistas participantes en La Liga Santander de la temporada 2019-2020, recientemente terminada, así como datos personales como sus nombres, edades y nacionalidades.

Proceden de la descarga en formato CSV de las tablas existentes en las siguientes webs:

https://fbref.com/es/comps/12/stats/Estadisticas-de-La-Liga#all_stats_standard

https://fbref.com/es/comps/12/misc/Estadisticas-de-La-Liga

estadísticas_porteros.csv

Datos sobre estadísticas generales de los porteros participantes en La Liga Santander de la temporada 2019-2020. Todos los jugadores que no actúen en la posición de portero tendrán estas estadísticas a NA.

Proceden de la descarga en formato CSV de las tablas existentes en las siguientes webs:

https://fbref.com/es/comps/12/keepers/Estadisticas-de-La-Liga

datos_fisicos_jugadores_liga_19_20.csv

Datos que recogen la altura y el peso de futbolistas participantes en La Liga Santander de la temporada 2019-2020. Hay jugadores que no tienen disponibles estos datos, por lo que se marcarán como NAs.

Proceden de la realización de scraping de la página de perfil de cada jugador en la web:

https://fbref.com/es/

El script utilizado para el proceso de scraping se encuentra en:

scripts/scraping_fbref.R

estadisticas_totales_jugadores_liga_19_20.csv

Aquí se recoge la unión de los datos obtenidos de las fuentes anteriores. Además, para su consecución, se ha realizado un proceso de preprocesamiento para la elección de las variables más relevantes, así como la creación de variables a partir de otras o la corrección de formato de algunas de ellas.

El proceso de preprocesamiento se recoge en:

scripts/preprocesing.R

Descripción de los datos finales

En esta tabla se recoge la descripción de los datos finales que serán objeto de análisis en este trabajo de fin de máster:

Variable	Descripción	Formato	NAs
nombre	Nombre del jugador	CHAR	0
pais	País en el que nació el jugador	FACTOR	0
edad	Edad en años del jugador	NUMERIC	0
altura	Altura en centímetros del jugador	NUMERIC	180
peso	Peso en kilogramos del jugador	NUMERIC	258
equipo	Equipo al que pertenece el jugador	FACTOR	0
posicion	Posición principal del jugador en el campo	FACTOR	0
posicion.alternativa	Posición secundaria del jugador en el campo	FACTOR	428
partidos.jugados	Número de partidos en los que el jugador ha disputado algún minuto	NUMERIC	0
partidos.titular	Número de partidos en los que el jugador ha disputado minutos desde el inicio	NUMERIC	0
minutos	Número de minutos disputados por el jugador	NUMERIC	0
goles.marcados	Número de goles anotados por el jugador	NUMERIC	0
goles.propia.puerta	Número de goles en propia puerta anotados por el jugador	NUMERIC	0
goles.encajados	Número de goles recibidos por el portero	NUMERIC	526
asistencias	Número de asistencias repartidas por el jugador	NUMERIC	0
pases.cruzados	Número de pases cruzados ejecutados por el jugador	NUMERIC	0
recuperaciones	Número de balones sueltos recuperados por el jugador	NUMERIC	0
intercepciones	Número de pases del equipo contrario interceptados por el jugador	NUMERIC	0
tackleos.ganados	Número de entradas defensivas ejecutadas con éxito por el jugador	NUMERIC	0
tiros.puerta.recibidos	Número de tiros a puerta recibidos por el portero	NUMERIC	526
paradas	Número de tiros parados por el portero	NUMERIC	526
duelos.aereos.ganados	Número de duelos aéreos ganados por el jugador	NUMERIC	0
duelos.aereos.perdidos	Número de duelos aéreos perdidos por el jugador	NUMERIC	0

faltas.cometidas	Número de faltas cometidas por el jugador	NUMERIC	0
faltas.recibidas	Número de faltas recibidas por el jugador	NUMERIC	0
tarjetas.amarillas	Número de tarjetas amarillas recibidas por el jugador	NUMERIC	0
tarjetas.rojas	Número de tarjetas rojas recibidas por el jugador	NUMERIC	0
segunda.amarilla	Número de tarjetas amarillas que fueron la segunda tarjeta amarilla para el jugador en un partido	NUMERIC	0
penales.marcados	Número de penales marcados por el jugador	NUMERIC	0
penales.lanzados	Número de penales lanzados por el jugador	NUMERIC	0
penales.concedidos	Número de penales cometidos por el jugador	NUMERIC	0
penales.recibidos	Número de penales recibidos por el portero	NUMERIC	526
penales.parados	Número de penales parados por el portero	NUMERIC	526
fueras.de.juego	Número de fueras de juego cometidos por el jugador	NUMERIC	0

Observaciones

Jugadores repetidos

Un jugador puede haber jugado durante la temporada en dos equipos distintos, por lo que habrá dos observaciones para dicho jugador. Por lo tanto, para los análisis que se realicen a nivel de jugador, se agruparán los datos por nombre de jugador para disponer de sus estadísticas totales.

Niveles de posicion y posicion.alternativa

PO - Porteros
DF - Defensas
CC - Centrocampistas
DL - Delanteros

Análisis descriptivo

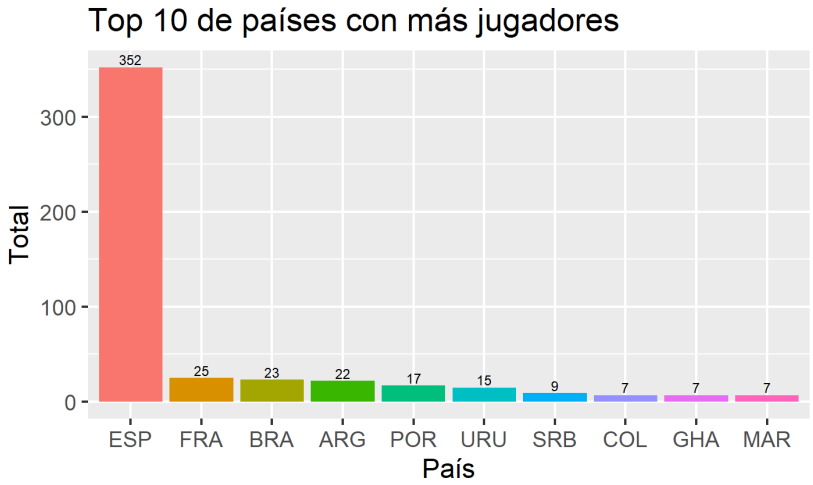
En este análisis se pretende responder a una serie de preguntas con los datos estadísticos obtenidos de los jugadores de La Liga 2019-2020.

El código para la realización de este análisis se recoge en:

scripts/descriptive_analysis.R

¿Cuáles son los países que más jugadores aportan a La Liga?

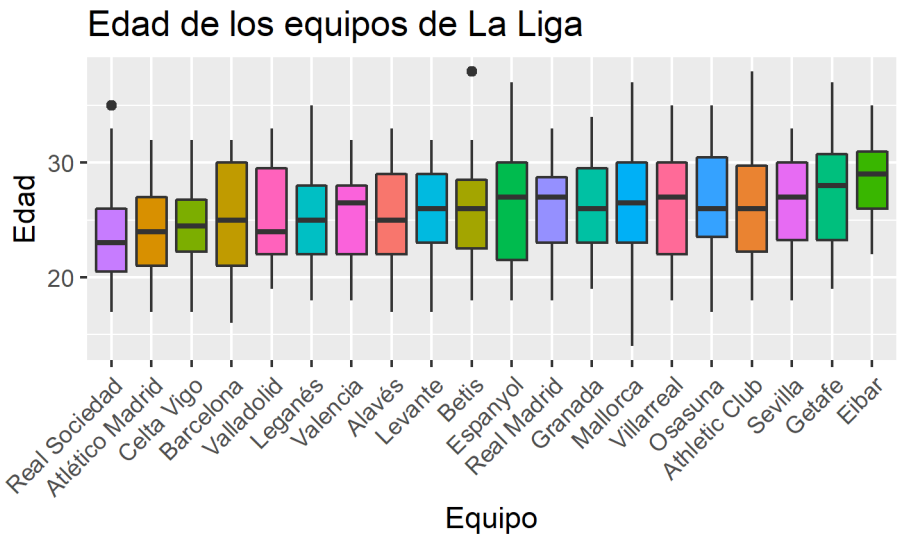
Veamos los 10 países que más jugadores aportan a La Liga:



Evidentemente, el grueso de jugadores los aporta España. Otros países como Brasil o Argentina siempre han aportado muchos jugadores, pero este año se han visto superados por Francia. Además de otros países habituales como Portugal, Uruguay o Serbia, se meten en el Top 10 otros que no lo son tanto como Colombia o Ghana.

¿Cuál es el equipo con menor media de edad en su plantilla?

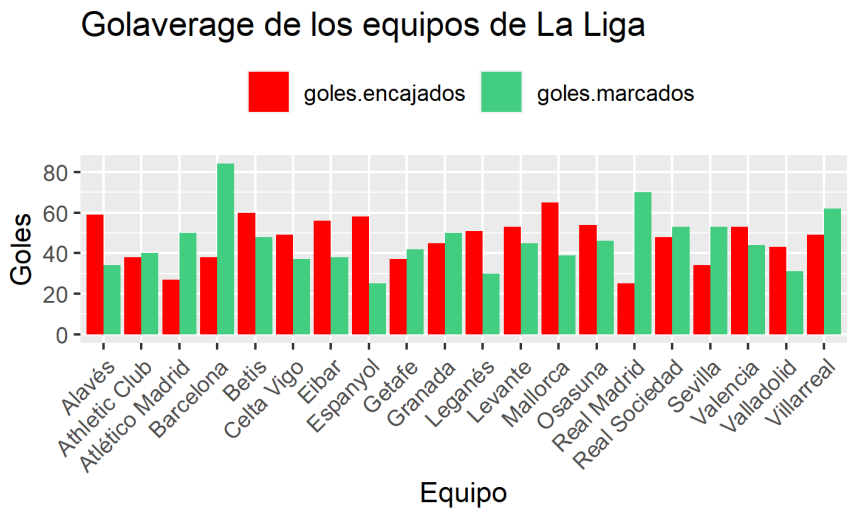
En este gráfico podemos ver todos los equipos ordenados por media de edad:



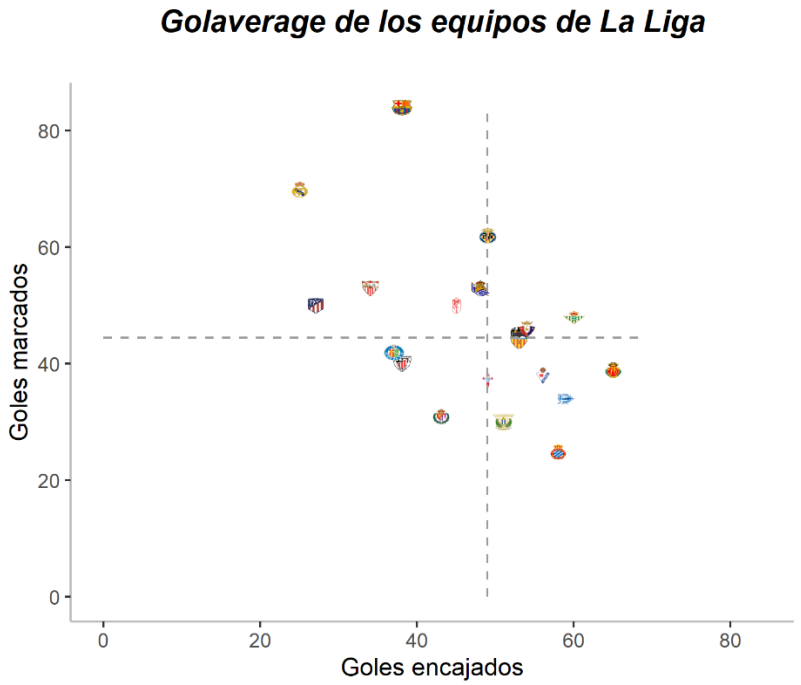
La Real Sociedad es el equipo que tiene una media de edad más joven en su plantilla y eso que, como podemos comprobar, tiene un jugador de 35 años (outlier) que hace que esta media se eleve. Por el contrario, el Éibar es claramente el equipo más 'viejo' de La Liga, donde la mayoría de sus jugadores se mueven entre los 26 y 31 años.
También observamos que los dos equipos de Barcelona (FC Barcelona y Espanyol) son los equipos que más variabilidad muestran.
Como anécdota, en el Mallorca debutó un jugador de 15 años y batió el record de precocidad del campeonato liguero.

¿Qué equipo ha sido el más goleador y cuál el más goleado?

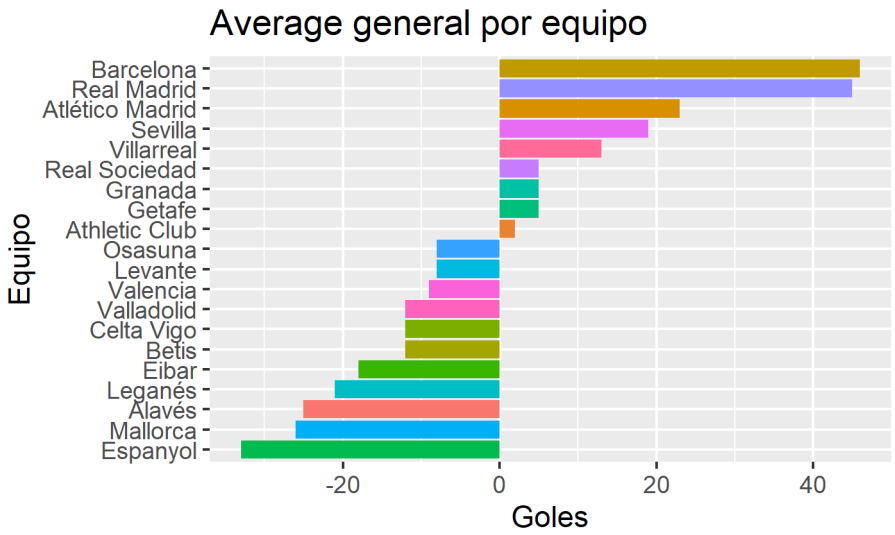
Lo visualizaremos con varios gráficos, unos con los goles marcados y recibidos, y otro con el golaveraje general de cada equipo:



Veámoslo con un scatterplot:



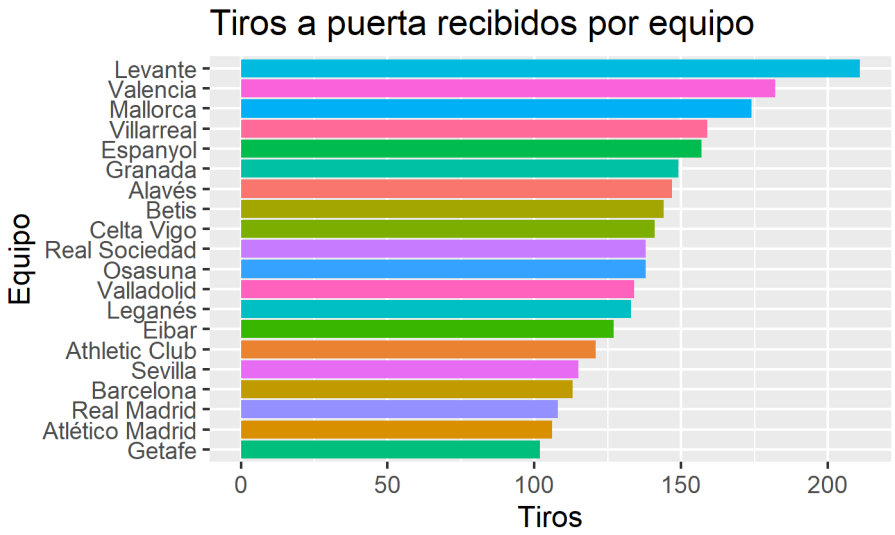
Se puede apreciar que el FC Barcelona ha sido el equipo más goleador y el Mallorca el más goleado. Además podemos observar la similitud entre equipos como Getafe y Athletic o Valencia y Osasuna que confirma que, tanto en juego como en la clasificación, son muy parecidos.



Si ordenamos los equipos por el golaveraje general obtenemos casi la misma clasificación que la de por puntos. Evidentemente, el golaveraje general es un buen indicador de la temporada de un equipo.

¿Qué equipo ha recibido más tiros a puerta?

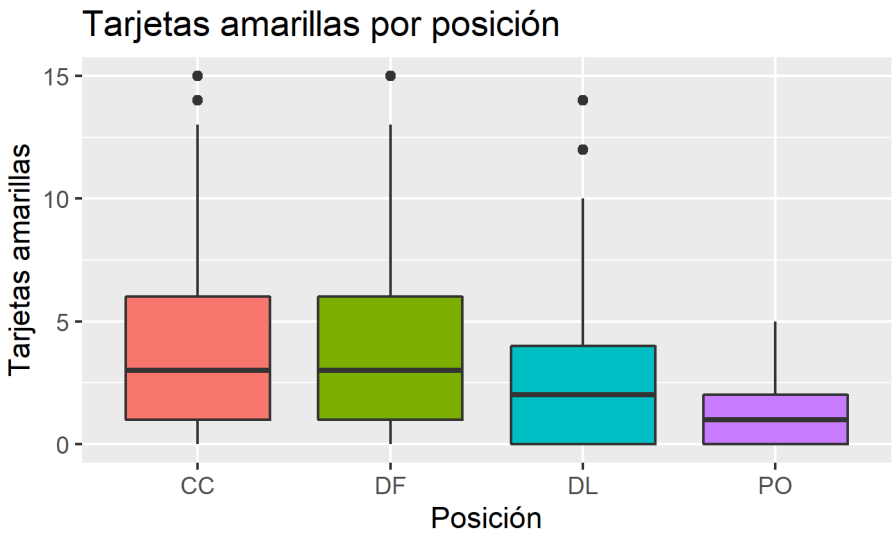
Veámoslo en este gráfico:



El equipo que más deja disparar a sus rivales es el Levante con bastante diferencia. Esto cuadra con la táctica de esperar muy atrás al rival que utiliza el conjunto valenciano. Por contra, el Getafe y el At. Madrid son los que permiten menos disparos a los contrarios. La intensidad defensiva y la presión en todo el campo, que utilizan tanto Bordalás como Simeone, lo hace posible.

¿Qué posición es la que más tarjetas amarillas recibe?

En este gráfico agrupamos por posición:



Los centrocampistas y defensas son los que más tarjetas reciben, seguidos de lejos por los delanteros y los porteros. Se pueden apreciar varios outliers, que representan a los jugadores denominados como “tarjeteros”, donde destacan Jaime Mata y Roberto Soldado que, siendo delanteros, han recibido 14 amarillas cada uno.

Análisis Inferencial

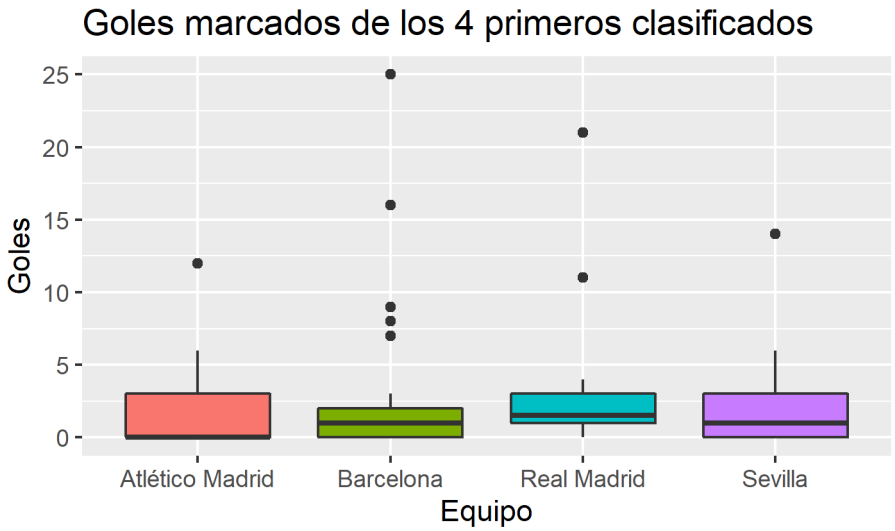
En este análisis realizaremos comparaciones de medias entre distintas muestras en las que en unos casos serán diferenciadas por equipo y en otros por posición.

El código para la realización de este análisis se recoge en:

scripts/inferential_analysis.R

Comparativa de goles de los jugadores de los cuatro primeros clasificados

Queremos comparar los goles marcados por los jugadores de los cuatro primeros clasificados, que para esta temporada de La Liga han sido Real Madrid, FC Barcelona, At. Madrid y Sevilla.



Se puede apreciar que el FC Barcelona presenta una distribución simétrica y el resto bastante asimétrica hacia la derecha, sobretodo el Real Madrid y el At. Madrid. Además, todos los equipos presentan outliers. Atendiendo a estas apreciaciones, todo indica que lo mejor será utilizar pruebas robustas.

Hipótesis

H_o: $\mu_1 = \mu_2 = \mu_3 = \mu_4$
H_1: Alguna μ es distinta

Prueba

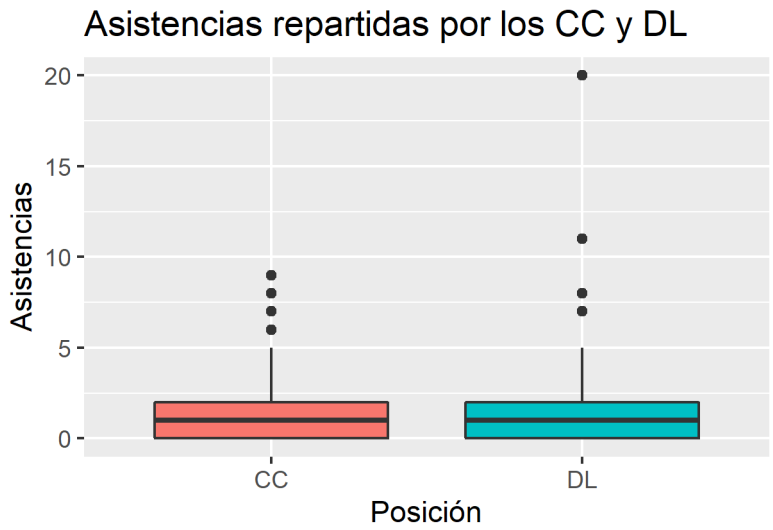
Vamos a utilizar el ANOVA de una vía para comparar las medias recortadas de cada equipo. Utilizaremos la función *t1way* del paquete *WRS2*.

```
Call:
t1way(formula = goles.marcados ~ equipo, data = estad.totales.jug.primeros)

Test statistic: F = 1.4593
Degrees of freedom 1: 3
Degrees of freedom 2: 35.16
p-value: 0.24233

Explanatory measure of effect size: 0.2
```

No se detectan diferencias significativas entre los diferentes equipos (p-value > .05), por lo que aceptamos la hipótesis nula. La media de goles de los jugadores de los cuatro primeros clasificados es similar.

Comparativa de asistencias entre centrocampistas y delanteros

Ahora queremos comparar la cantidad de asistencias repartidas por los centrocampistas con las repartidas por los delanteros.



Se aprecian distribuciones simétricas y muy similares para ambas posiciones. También presentan ambas varios outliers, por lo que nos decantamos por pruebas robustas.

Hipótesis

H_o: $\mu_1 = \mu_2$
H_1: $\mu_1 \neq \mu_2$

Prueba

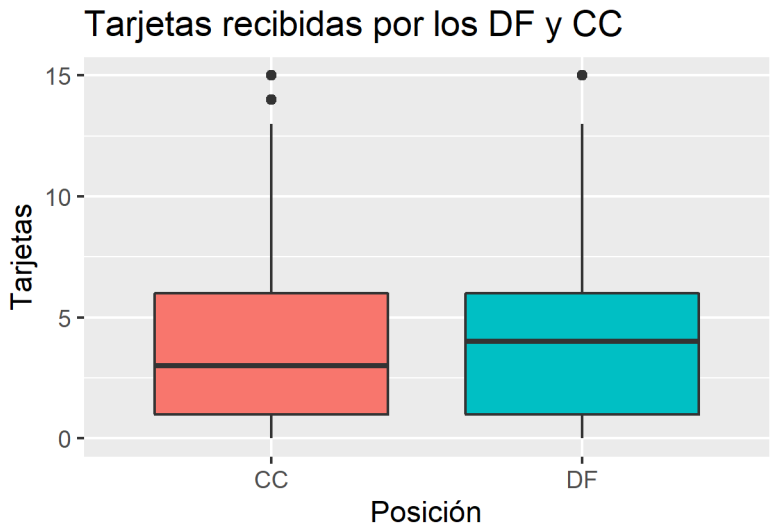
Utilizaremos la prueba de Yuen para la comparación de medias recortadas con la función *yuen* del paquete *WRS2*.

```
Call:
yuen(formula = asistencias ~ posicion, data = estad.totales.jug.cc.dl)

Test statistic: 1.3543 (df = 137.38), p-value = 0.17787

Trimmed mean difference:  -0.2735
95 percent confidence interval:
-0.6728     0.1258

Explanatory measure of effect size: 0.1
```

No se detectan diferencias significativas entre las diferentes posiciones (p-value > .05), por lo que aceptamos la hipótesis nula. La media de asistencias de centrocampistas y delanteros es similar.

Comparativa de tarjetas entre defensas y centrocampistas

Vamos a comparar la cantidad de tarjetas amarillas recibidas por los defensas con las recibidas por los centrocampistas.



Apreciamos ciertas asimetrías, hacia la derecha para los centrocampistas y hacia la izquierda para los defensas. Además existe algún outlier, así que optaremos por realizar pruebas robustas.

Hipótesis

H_o: $\mu_1 = \mu_2$
H_1: $\mu_1 \neq \mu_2$

Prueba

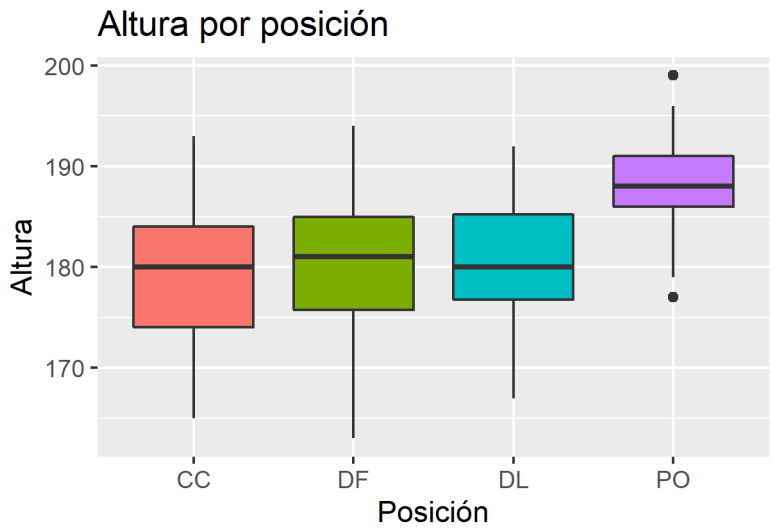
Utilizaremos la prueba de Yuen para la comparación de medias recortadas con la función *yuen* del paquete *WRS2*.

```
Call:
yuen(formula = tarjetas.amarillas ~ posicion, data = estad.totales.jug.df.cc)

Test statistic: 1.2126 (df = 224.69), p-value = 0.22654

Trimmed mean difference:  -0.45502
95 percent confidence interval:
-1.1944     0.2844

Explanatory measure of effect size: 0.09
```

No se detectan diferencias significativas entre las diferentes posiciones (p-value > .05), por lo que aceptamos la hipótesis nula. El número medio de tarjetas amarillas recibidas por defensas y centrocampistas es similar.

Comparativa de alturas por posición

Queremos comparar la altura de los jugadores para las distintas demarcaciones.



Se aprecian pequeñas asimetrías en todas las distribuciones, y destaca la demarcación de portero con alturas bastante más elevadas y algún outlier. Nos decantamos por utilizar pruebas robustas.

Hipótesis

H_0: $\mu_1 = \mu_2 = \mu_3 = \mu_4$
H_1: Alguna μ es distinta

Prueba

Vamos a utilizar el ANOVA de una vía para comparar las medias recortadas de cada posición. Utilizaremos la función *t1way* del paquete *WRS2*.

```
Call:
t1way(formula = altura ~ posicion, data = estad.totales.jug.todas.posiciones)

Test statistic: F = 30.3448
Degrees of freedom 1: 3
Degrees of freedom 2: 86.86
p-value: 0

Explanatory measure of effect size: 0.56
```

Se aprecian diferencias significativas entre las medias ($p < .05$) por lo que rechazamos la hipótesis nula.

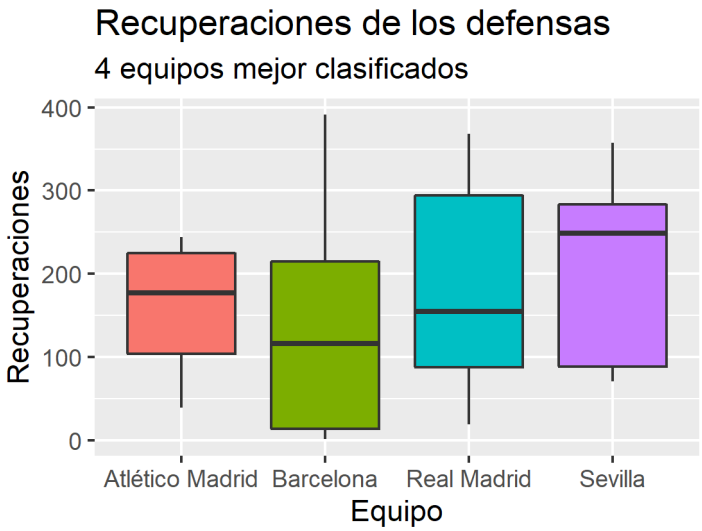
Vamos a realizar pruebas post-hoc para ver más a fondo estas diferencias. Utilizaremos la función *lincon* del paquete *WRS2*.

```
Call:
lincon(formula = altura ~ posicion, data = estad.totales.jug.todas.posiciones)

             psihat  ci.lower ci.upper p.value
CC vs. DF -1.54146  -3.81182  0.72891 0.19799
CC vs. DL -1.62415  -4.22174  0.97344 0.19799
CC vs. PO -9.00693 -11.70784 -6.30602 0.00000
DF vs. DL -0.08269  -2.62725  2.46186 0.93119
DF vs. PO -7.46548 -10.11605 -4.81490 0.00000
DL vs. PO -7.38278 -10.30233 -4.46324 0.00000
```

Claramente, las diferencias mayores son las que tiene la posición de portero con el resto de posiciones. Entre el resto de demarcaciones no hay apenas diferencias.

Comparativa de recuperaciones por defensas de los cuatro primeros equipos

Vamos a comparar el número de recuperaciones de los jugadores de los cuatro equipos mejor clasificados, que para esta temporada de La Liga han sido Real Madrid, FC Barcelona, At. Madrid y Sevilla.



Se pueden apreciar ciertas asimetrías, sobretodo en la distribución del Sevilla, pero no se perciben outliers.

Para ver qué tipo de pruebas utilizamos, comprobaremos la normalidad de estas distribuciones. Utilizaremos la prueba de Shapiro-Wilk con la función *shapiro.test* del paquete *stats*.

```
        Shapiro-Wilk normality test

data:  estad.totales.jug.primeros.df[estad.totales.jug.primeros.df$equipo ==     "Real Madrid", ]$recuperaciones
W = 0.92774, p-value = 0.4957

data:  estad.totales.jug.primeros.df[estad.totales.jug.primeros.df$equipo ==     "Barcelona", ]$recuperaciones
W = 0.90551, p-value = 0.2156

data:  estad.totales.jug.primeros.df[estad.totales.jug.primeros.df$equipo ==     "Atlético Madrid", ]$recuperaciones
W = 0.89758, p-value = 0.2383

data:  estad.totales.jug.primeros.df[estad.totales.jug.primeros.df$equipo ==     "Sevilla", ]$recuperaciones
W = 0.87201, p-value = 0.1933
```

Todas las distribuciones son normales (p-value >.05), por lo que utilizaremos pruebas paramétricas.

Hipótesis

H_o: $\mu_1 = \mu_2 = \mu_3 = \mu_4$
H_1: Alguna μ es distinta

Prueba

Vamos a utilizar el ANOVA de una vía para comparar las medias de cada equipo. Utilizaremos la función *aov* del paquete *stats.*

```
Call:
   aov(formula = recuperaciones ~ equipo, data = estad.totales.jug.primeros.df)

Terms:
                  equipo Residuals
Sum of Squares   23790.1  417243.8
Deg. of Freedom        3        31

Residual standard error: 116.015
Estimated effects may be unbalanced

            Df Sum Sq Mean Sq F value Pr(>F)
equipo       3  23790    7930   0.589  0.627
Residuals   31 417244   13459
```

No se detectan diferencias significativas entre los diferentes equipos (p-value > .05), por lo que aceptamos la hipótesis nula. La media de recuperaciones es similar para los jugadores de los cuatro primeros clasificados.

Análisis de correlación y regresión

En estos análisis trataremos de investigar que variables tienen más correlación entre ellas y además definiremos modelos de regresión lineal múltiple para explicar algunas variables en base al resto.

Antes de realizar ningún análisis, vamos a dividir el conjunto de datos en tres subconjuntos:

- Datos de los porteros con las variables relativas a esta demarcación.
- Datos de defensas y centrocampistas con las variables relativas a la defensa.
- Datos de centrocampistas y delanteros con las variables relativas al ataque.

Las variables *edad*, *altura*, *peso* y *minutos* serán comunes a todos los subconjuntos.

Los NAs de las variables *edad*, *altura*, *peso, recuperaciones, duelos.aereos.ganados, duelos.aereos.perdidos* y *penales.concedidos* se sustituirán por la media respectiva.

Además, se estandarizarán todas las variables por tener diferentes unidades. Para este cometido, utilizaremos la función *decostand* del paquete *vegan*.

El código para la realización de este análisis se recoge en:

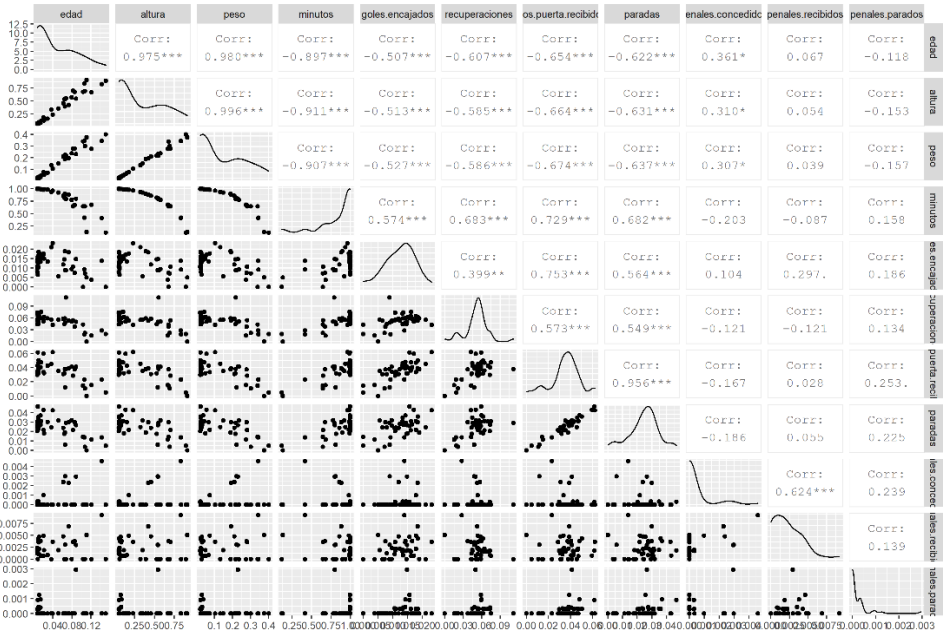
scripts/correlation_regression_analysis.R

Modelo de regresión que explica la variable goles.encajados

En este análisis queremos explicar la variable *goles.encajados* con el resto de variables recopiladas sobre los jugadores con demarcación de portero.

Análisis de correlación

Vemos los valores de correlación y gráficos de dispersión con la función *ggpairs* del paquete *GGally*:



De primeras, se pueden apreciar correlaciones positivas muy altas, bastante esperadas, como el número de tiros a puerta recibidos con los minutos jugados o el número de goles recibidos y de paradas con el número de tiros a puerta recibidos. Hay otras menos entendibles, como la alta correlación negativa entre minutos y las variables de altura y peso.

Análisis de regresión

Vamos a ajustar un modelo de regresión lineal múltiple para explicar la variable *goles.encajados* con el resto de variables y evaluaremos su bondad de ajuste.

Modelo

Utilizaremos la función *lm* del paquete *stats*.

```
Call:
lm(formula = goles.encajados ~ ., data = estad.totales.por.est)

Residuals:
       Min         1Q     Median         3Q        Max
-0.0032017 -0.0002149  0.0000969  0.0003382  0.0030168

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)             -0.001097   0.002096  -0.524   0.6041
edad                    -0.018710   0.020375  -0.918   0.3651
altura                  -0.006751   0.006195  -1.090   0.2838
peso                     0.022814   0.016023   1.424   0.1639
minutos                  0.001753   0.002155   0.813   0.4218
recuperaciones          -0.001928   0.012157  -0.159   0.8749
tiros.puerta.recibidos   0.985708   0.043224  22.805  < 2e-16 ***
paradas                 -0.996983   0.054406 -18.325  < 2e-16 ***
penales.concedidos      -0.112702   0.260253  -0.433   0.6678
penales.recibidos        0.924761   0.111951   8.260 1.54e-09 ***
penales.parados         -0.767733   0.334149  -2.298   0.0281 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.001014 on 33 degrees of freedom
Multiple R-squared:  0.9701,    Adjusted R-squared:  0.961
F-statistic:   107 on 10 and 33 DF,  p-value: < 2.2e-16
```

Vemos que el RSE es .001 y obtenemos una tasa de error del 8,19%, que está muy bien.

El R^2 ajustado es .961, por lo que con estas variables podemos explicar el 96,1% de la variabilidad de *goles.encajados*. Veamos ahora los R^2 parciales:

```
$adjustment
[1] FALSE

$variable
 [1] "edad"                 "altura"               "peso"                 "minutos"              "recuperaciones"
 [6] "tiros.puerta.recibidos" "paradas"            "penales.concedidos"   "penales.recibidos"    "penales.parados"

$partial.rsq
 [1] 0.0249171590 0.0347267481 0.0578758365 0.0196585102 0.0007618508 0.9403320716 0.9105206822 0.0056505983 0.6740230905 0.1379055606
```

Podemos apreciar que *tiros.puerta.recibidos*, *paradas* y *penales.recibidos* son las variables que mejor explican de forma independiente la variabilidad de *goles.encajados*.

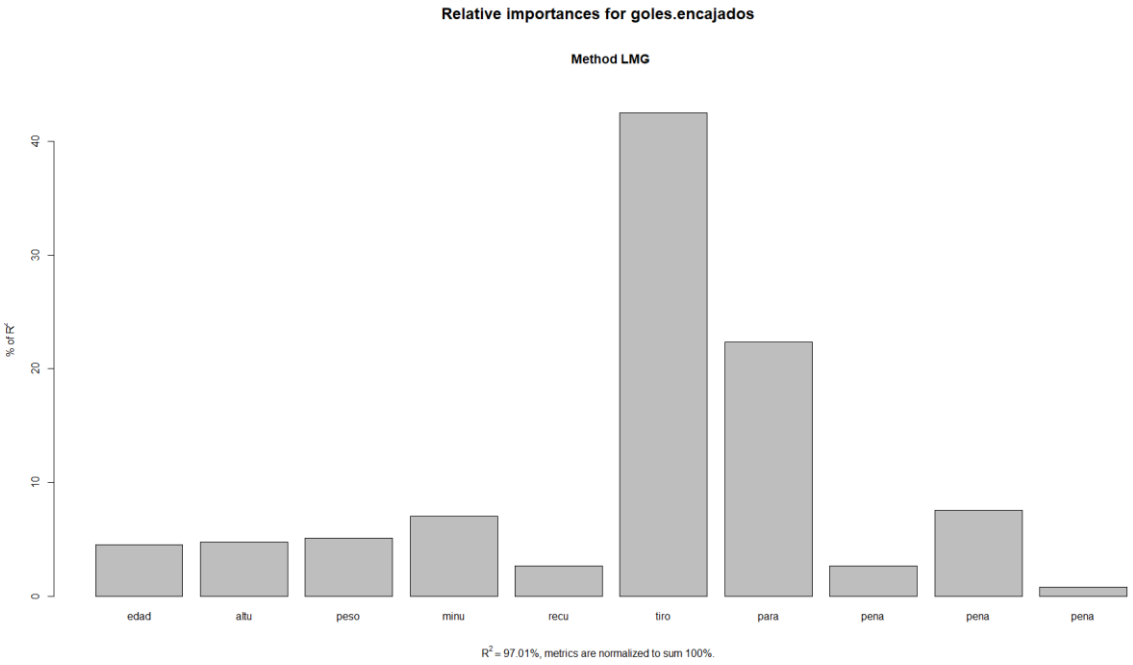
Vamos a interpretar los coeficientes del modelo (pruebas t):

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-0.001097093	0.002095510	-0.5235447	6.040936e-01
edad	-0.018710199	0.020374780	-0.9183019	3.651243e-01
altura	-0.006750515	0.006195452	-1.0895920	2.837859e-01
peso	0.022813589	0.016022935	1.4238084	1.638892e-01
minutos	0.001753120	0.002155103	0.8134738	4.217773e-01
recuperaciones	-0.001928270	0.012156551	-0.1586198	8.749352e-01
tiros.puerta.recibidos	0.985707866	0.043223617	22.8048445	8.947355e-22
paradas	-0.996982798	0.054406040	-18.3248549	7.277145e-19
penales.concedidos	-0.112701678	0.260252976	-0.4330466	6.677983e-01
penales.recibidos	0.924760525	0.111951095	8.2603973	1.537520e-09
penales.parados	-0.767732595	0.334148520	-2.2975789	2.806326e-02

Vemos que sólo las variables *tiros.puerta.recibidos*, *paradas*, *penales.recibidos* y *penales.parados* afectarían significativamente a la variable *goles.encajados* para una cantidad fija en el resto de variables. Tendríamos que comprobar si podemos eliminar el resto de variables mediante técnicas de comparación de modelos.

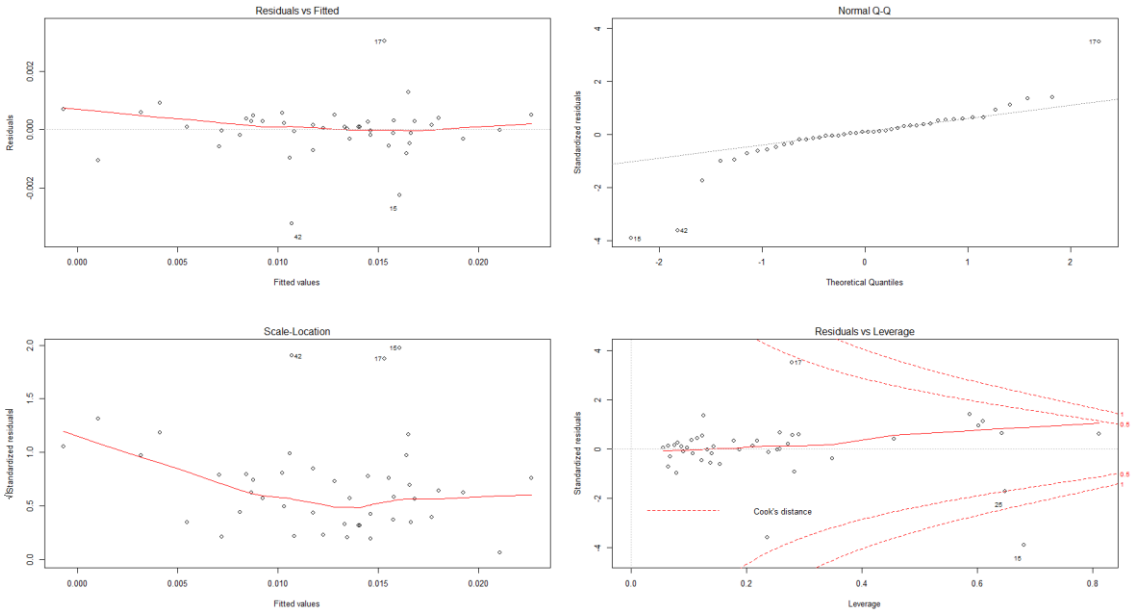
Ahora vamos a estimar la importancia relativa de los predictores:

```
Response variable: goles.encajados
Total response variance: 2.636059e-05
Analysis based on 44 observations

10 Regressors:
edad altura peso minutos recuperaciones tiros.puerta.recibidos paradas penales.concedidos penales.recibidos penales.parados
Proportion of variance explained by model: 97.01%
Metrics are normalized to sum to 100% (rela=TRUE).

Relative importance metrics:

                            lmg
edad                0.045228056
altura              0.047545393
peso                0.051161675
minutos             0.070338877
recuperaciones      0.026688033
tiros.puerta.recibidos 0.424792673
paradas             0.223686553
penales.concedidos  0.026604524
penales.recibidos   0.075732236
penales.parados     0.008221979

Average coefficients for different model sizes:

                              1x          2xs          3xs          4xs          5xs          6xs          7xs          8xs
edad                -0.061254860 -0.029940755 -0.01559037 -0.010673082 -0.010917393 -0.013603573 -0.016750487 -0.018976749
altura              -0.009137514 -0.002609554  0.00224427  0.005645334  0.007634311  0.008138517  0.007008508  0.004104358
peso                -0.021950686 -0.023439021 -0.02451432 -0.024655978 -0.023347128 -0.019985575 -0.013966025 -0.004838401
minutos              0.013206719  0.011612182  0.01000708  0.008514000  0.007128077  0.005843490  0.004674636  0.003616827
recuperaciones       0.110693572  0.054317555  0.02493762  0.009727242  0.001135956 -0.004103928 -0.006958144 -0.007582790
tiros.puerta.recibidos  0.285325590  0.356944826  0.42666590  0.497985354  0.572327986  0.649930011  0.730447406  0.813517115
paradas              0.285973213  0.106499931 -0.04994649 -0.190481246 -0.323299203 -0.454415866 -0.586881932 -0.721657288
penales.concedidos   0.523847204  0.942874237  1.07583338  1.042573529  0.935926888  0.795987075  0.627852140  0.425014812
penales.recibidos    0.737763980  0.794075335  0.76563005  0.738678875  0.727185908  0.729505017  0.747175505  0.783864827
penales.parados      1.878436010  1.071131705  0.53890959  0.188141245 -0.077413345 -0.296055843 -0.474558807 -0.613695279
                              9Xs         10Xs
edad                -0.0196482188 -0.018710199
altura              -0.0005671308 -0.006750515
peso                 0.0075221586  0.022813589
minutos              0.0026298333  0.001753120
recuperaciones      -0.0058171044 -0.001928270
tiros.puerta.recibidos  0.8988855419  0.985707866
paradas             -0.8587646516 -0.996982798
penales.concedidos   0.1792354035 -0.112701678
penales.recibidos    0.8430642276  0.924760525
penales.parados     -0.7121723139 -0.767732595
```



Observamos que las variables *tiros.puerta.recibidos*, *paradas*, *penales.recibidos* y *minutos* son las que más importancia relativa tienen en el modelo.

Vemos ahora si se cumplen los supuestos del modelo y la existencia de datos atípicos:



En los gráficos *Residuals vs Fitted* y *Scale-Location*, podemos ver que los residuos siguen un patrón al azar. Teniendo en cuenta esto, se cumplen los supuestos de linealidad y homocedasticidad.

En el gráfico *Normal Q-Q*, los puntos se separan de la diagonal en las colas, por lo que no cumplimos el supuesto de normalidad.

En el gráfico *Residuals vs Leverage* se aprecia que el punto 15 sobrepasa la distancia de Cook, por lo que habrá que estudiarlo.

Por último, para comprobar el supuesto de independencia, realizaremos el test de Durbin-Watson mediante la función *durbinWatsonTest* del paquete *car*:

```
lag Autocorrelation D-W Statistic p-value
  1      -0.03725336       2.05766   0.455
Alternative hypothesis: rho > 0
```

No existe correlación, por lo que cumplimos el supuesto de independencia.

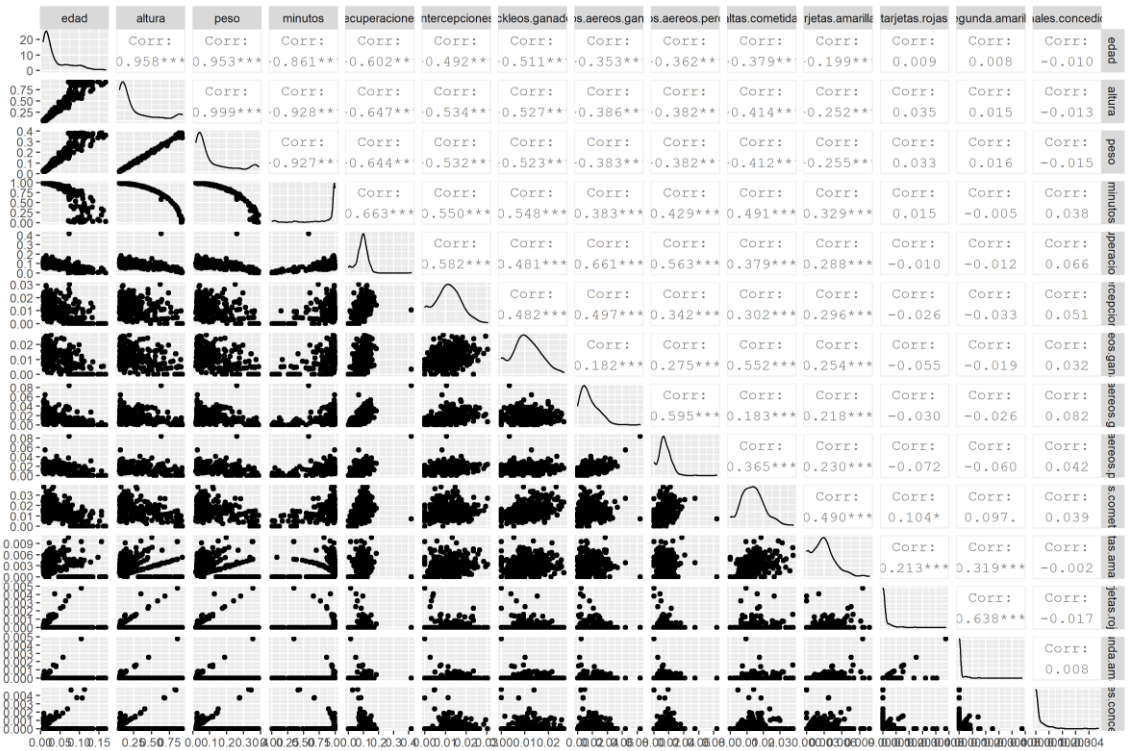
Con estos datos, podemos concluir que no disponemos de un modelo del todo fiable para explicar la variable *goles.encajados*. Tenemos que solucionar el problema de no normalidad en los residuos y vigilar el punto 15 que se considera outlier.

Modelo de regresión que explica la variable tarjetas.amarillas

En este análisis queremos explicar la variable *tarjetas.amarillas* con el resto de variables defensivas recopiladas sobre los jugadores con demarcación de defensa y centrocampista.

Análisis de correlación

Vemos los valores de correlación y gráficos de dispersión:



Podemos apreciar correlaciones esperadas como las de la variable *recuperaciones* con minutos y *duelos.aereos.ganados*. Destacar también la baja correlación entre la variable *altura* y *duelos.aereos.ganados*, que a priori podría imaginarse más alta.

Análisis de regresión

Vamos a ajustar un modelo de regresión lineal múltiple para explicar la variable *tarjetas.amarillas* con el resto de variables y evaluaremos su bondad de ajuste.

Modelo

Utilizaremos la función *lm* del paquete *stats*.

```
Call:
lm(formula = tarjetas.amarillas ~ ., data = estad.totales.resto.def.est)

Residuals:
       Min         1Q     Median         3Q        Max
-0.0042822 -0.0009667 -0.0001623  0.0006824  0.0067901

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)             -0.0018509  0.0010006  -1.850  0.06513 .
edad                     0.0115855  0.0085876   1.349  0.17811
altura                   0.0136504  0.0068323   1.998  0.04644 *
peso                    -0.0323431  0.0154832  -2.089  0.03739 *
minutos                  0.0021659  0.0009658   2.243  0.02550 *
recuperaciones           0.0012253  0.0036763   0.333  0.73910
intercepciones           0.0479978  0.0175018   2.742  0.00639 **
tackleos.ganados        -0.0234952  0.0191344  -1.228  0.22025
duelos.aereos.ganados    0.0133374  0.0104931   1.271  0.20449
duelos.aereos.perdidos  -0.0041932  0.0128469  -0.326  0.74430
faltas.cometidas         0.1133930  0.0147855   7.669 1.49e-13 ***
tarjetas.rojas          -0.1541889  0.2189674  -0.704  0.48177
segunda.amarilla         1.8870968  0.3230214   5.842 1.12e-08 ***
penales.concedidos      -0.1607667  0.1460222  -1.101  0.27161
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.001608 on 377 degrees of freedom
Multiple R-squared:  0.3813,    Adjusted R-squared:  0.3599
F-statistic: 17.87 on 13 and 377 DF,  p-value: < 2.2e-16
```

Vemos que el RSE es .001 y obtenemos una tasa de error del 59,28%, muy alto.

El R^2 ajustado es .3599, por lo que con estas variables podemos explicar el 36 % de la variabilidad de *tarjetas.amarillas*. Veamos ahora los R^2 parciales:

```
$adjustment
[1] FALSE

$variable
 [1] "edad"                "altura"              "peso"                "minutos"             "recuperaciones"
 [6] "intercepciones"      "tackleos.ganados"    "duelos.aereos.ganados" "duelos.aereos.perdidos" "faltas.cometidas"
[11] "tarjetas.rojas"      "segunda.amarilla"    "penales.concedidos"

$partial.rsq
 [1] 0.0048045795 0.0104770147 0.0114420623 0.0131646754 0.0002945682 0.0195594135 0.0039833819 0.0042671082 0.0002825111
[10] 0.1349574306 0.0013135150 0.0830132476 0.0032049332
```

Vemos que *faltas.cometidas* es la variable que mejor explica de forma independiente la variabilidad de *tarjetas.amarillas*.

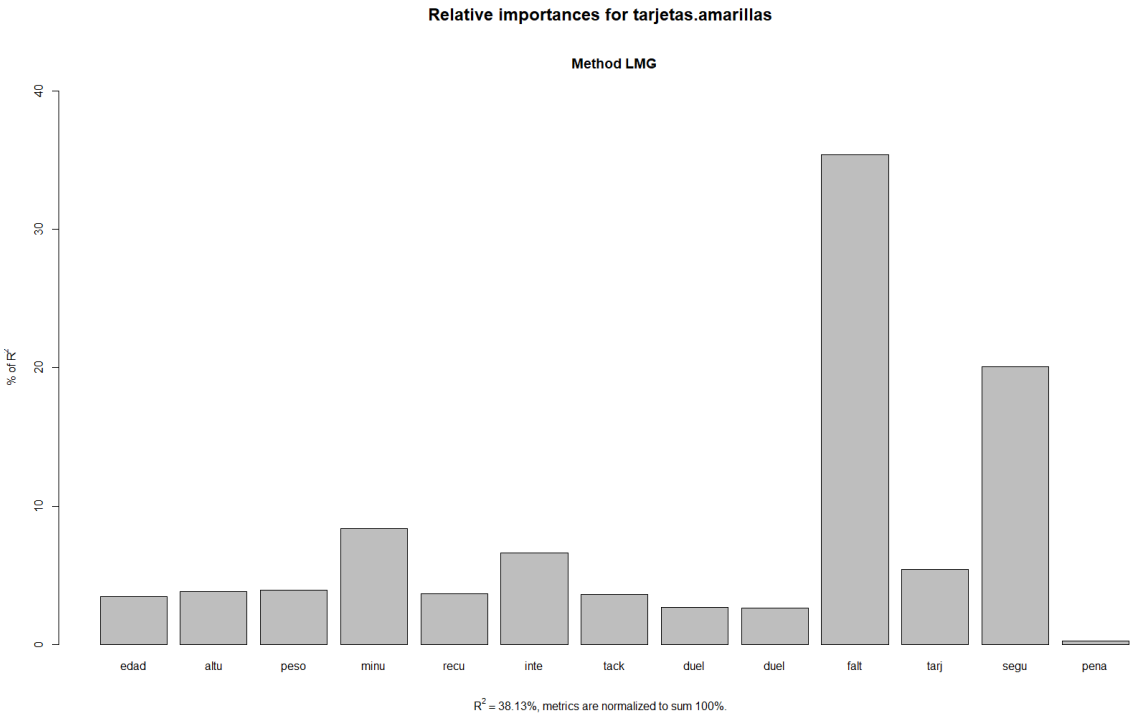
Vamos a interpretar los coeficientes del modelo (pruebas t):

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-0.001850897	0.0010005994	-1.8497883	6.512647e-02
edad	0.011585506	0.0085875803	1.3491001	1.781145e-01
altura	0.013650374	0.0068323178	1.9979127	4.644412e-02
peso	-0.032343095	0.0154831675	-2.0889198	3.738511e-02
minutos	0.002165942	0.0009658138	2.2426082	2.550266e-02
recuperaciones	0.001225303	0.0036763422	0.3332940	7.390976e-01
intercepciones	0.047997752	0.0175018092	2.7424451	6.389227e-03
tackleos.ganados	-0.023495203	0.0191344441	-1.2279010	2.202502e-01
duelos.aereos.ganados	0.013337411	0.0104931395	1.2710601	2.044910e-01
duelos.aereos.perdidos	-0.004193217	0.0128468775	-0.3263997	7.443031e-01
faltas.cometidas	0.113392976	0.0147854932	7.6692048	1.488980e-13
tarjetas.rojas	-0.154188878	0.2189673931	-0.7041636	4.817656e-01
segunda.amarilla	1.887096848	0.3230214354	5.8420174	1.115564e-08
penales.concedidos	-0.160766721	0.1460222138	-1.1009744	2.716102e-01

Vemos que sólo las variables *faltas.cometidas, altura, peso, minutos, intercepciones y segunda.amarilla* afectarían significativamente a la variable *tarjetas.amarillas* para una cantidad fija en el resto de variables. Tendríamos que comprobar si podemos eliminar el resto de variables mediante técnicas de comparación de modelos.

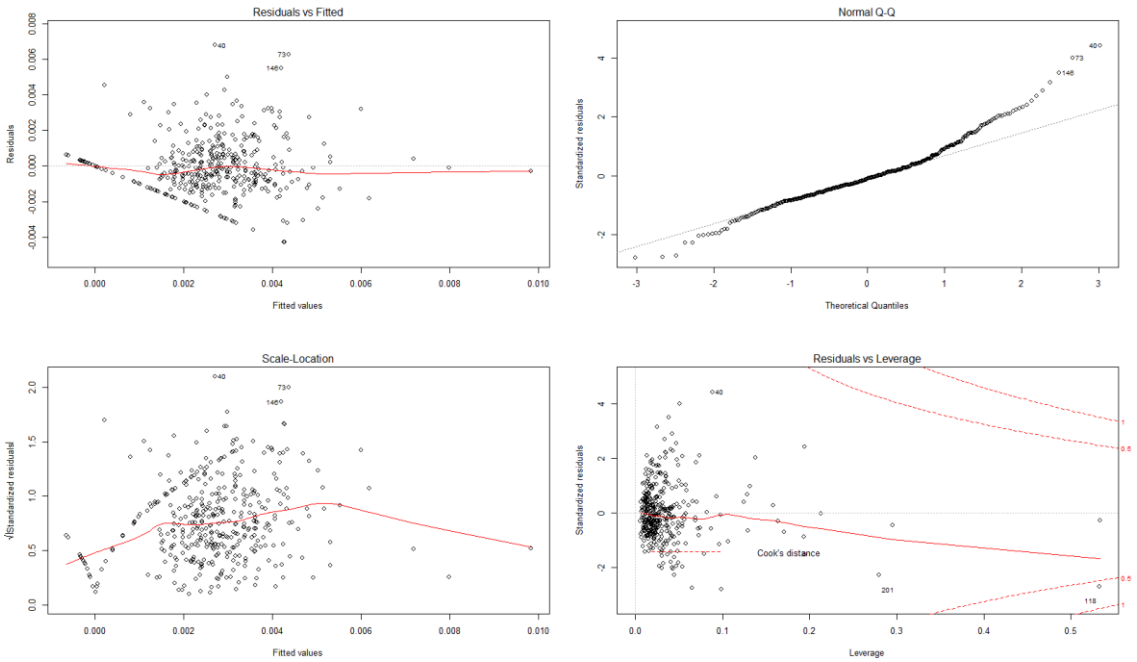
Ahora vamos a estimar la importancia relativa de los predictores:

```
Response variable: tarjetas.amarillas
Total response variance: 4.041418e-06
Analysis based on 391 observations

13 Regressors:
edad altura peso minutos recuperaciones intercepciones tackleos.ganados duelos.aereos.ganados duelos.aereos.perdidos faltas.cometidas ta
rjetas.rojas segunda.amarilla penales.concedidos
Proportion of variance explained by model: 38.13%
Metrics are normalized to sum to 100% (rela=TRUE).

Relative importance metrics:

                            lmg
edad                0.034847094
altura              0.038057720
peso                0.039322415
minutos             0.083721867
recuperaciones      0.036498971
intercepciones      0.066394038
tackleos.ganados    0.036144604
duelos.aereos.ganados  0.027082313
duelos.aereos.perdidos 0.026527233
faltas.cometidas    0.353953125
tarjetas.rojas      0.054374802
segunda.amarilla    0.200599775
penales.concedidos  0.002476041

Average coefficients for different model sizes:

                              1Xs          2Xs          3Xs          4Xs          5Xs          6Xs          7Xs          8Xs
edad                -0.011159386  0.0010214432  0.008830385  0.013786360  0.016832862  0.018553867  0.019310441  0.019324457
altura              -0.001846219 -0.0003506749  0.001040598  0.002340702  0.003564992  0.004731616  0.005861857  0.006980727
peso                -0.004571054 -0.0054891087 -0.007012749 -0.008941561 -0.011128930 -0.013471983 -0.015903588 -0.018387389
minutos              0.002563430  0.0028419810  0.003009914  0.003081692  0.003073449  0.003002336  0.002886020  0.002742191
recuperaciones       0.014946943  0.0116082969  0.009190131  0.007379839  0.005972251  0.004839273  0.003906049  0.003132510
intercepciones       0.093554164  0.0753532964  0.065012294  0.058885058  0.055026381  0.052441897  0.050639831  0.049379119
tackleos.ganados     0.084965335  0.0593448658  0.043179727  0.031949756  0.023328016  0.016139268  0.009780240  0.003916790
duelos.aereos.ganados   0.036470799  0.0252266561  0.019696806  0.016870277  0.015329647  0.014439472  0.013912722  0.013604867
duelos.aereos.perdidos  0.053047363  0.0377405968  0.029069041  0.023432204  0.019193964  0.015643560  0.012477453  0.009553772
faltas.cometidas     0.136112943  0.1295174090  0.125288792  0.122355541  0.120102749  0.118256057  0.116718075  0.115463748
tarjetas.rojas       0.851526237  0.7909001022  0.717305484  0.638435990  0.556988880  0.473956866  0.389736408  0.304401279
segunda.amarilla     1.935304503  1.9334487828  1.917952789  1.900545212  1.884698110  1.871579114  1.861601969  1.854928053
penales.concedidos  -0.007690912 -0.0402665018 -0.059008429 -0.073197992 -0.085566049 -0.096757137 -0.106951085 -0.116309780
                              9Xs         10Xs         11Xs         12Xs         13Xs
edad                 0.018730903  0.017612428  0.016022858  0.014002305  0.011585506
altura               0.008117846  0.009308459  0.010594234  0.012023455  0.013650374
peso                -0.020915840 -0.023510302 -0.026221740 -0.029130325 -0.032343095
minutos              0.002588047  0.002439759  0.002311958  0.002217280  0.002165942
recuperaciones       0.002499379  0.001997925  0.001622957  0.001368505  0.001225303
intercepciones       0.048535739  0.048034925  0.047817963  0.047826927  0.047997752
tackleos.ganados    -0.001660817 -0.007099905 -0.012508580 -0.017961047 -0.023495203
duelos.aereos.ganados   0.013430113  0.013334039  0.013287119  0.013283183  0.013337411
duelos.aereos.perdidos  0.006785490  0.004099433  0.001422968 -0.001317573 -0.004193217
faltas.cometidas     0.114491203  0.113802483  0.113396581  0.113266020  0.113392976
tarjetas.rojas       0.217750507  0.129324726  0.038438969 -0.055756927 -0.154188878
segunda.amarilla     1.851747939  1.852470292  1.857830133  1.868913848  1.887096848
penales.concedidos  -0.125059174 -0.133494893 -0.141975959 -0.150916068 -0.160766721
```



Observamos que las variables *faltas.cometidas*, *segunda.amarilla, minutos, intercepciones* y *tarjetas.rojas* son las que más importancia relativa tienen en el modelo.

Vemos ahora si se cumplen los supuestos del modelo y la existencia de datos atípicos:



En los gráficos Residuals vs Fitted y Scale-Location podemos apreciar un patrón de embudo. Se cumple el supuesto de linealidad, pero no el de homocedasticidad.
En el gráfico Normal Q-Q se aprecia que los residuos se apartan de la diagonal en las colas, sobre todo en la derecha, por lo que no cumple el supuesto de normalidad.

En el gráfico Residuals vs Leverage se aprecia que ningún punto sobrepasa la distancia de Cook, aunque el punto 118 está en el límite, por lo que no tenemos datos atípicos.

Ahora realizaremos el test de Durbin-Watson mediante la función durbinWatsonTest del paquete car, para comprobar el supuesto de independencia:

```
lag Autocorrelation D-W Statistic p-value
  1      -0.06958727       2.138769   0.926
Alternative hypothesis: rho > 0
```

No hay correlación en los residuos, por lo que cumplimos el supuesto de independencia.

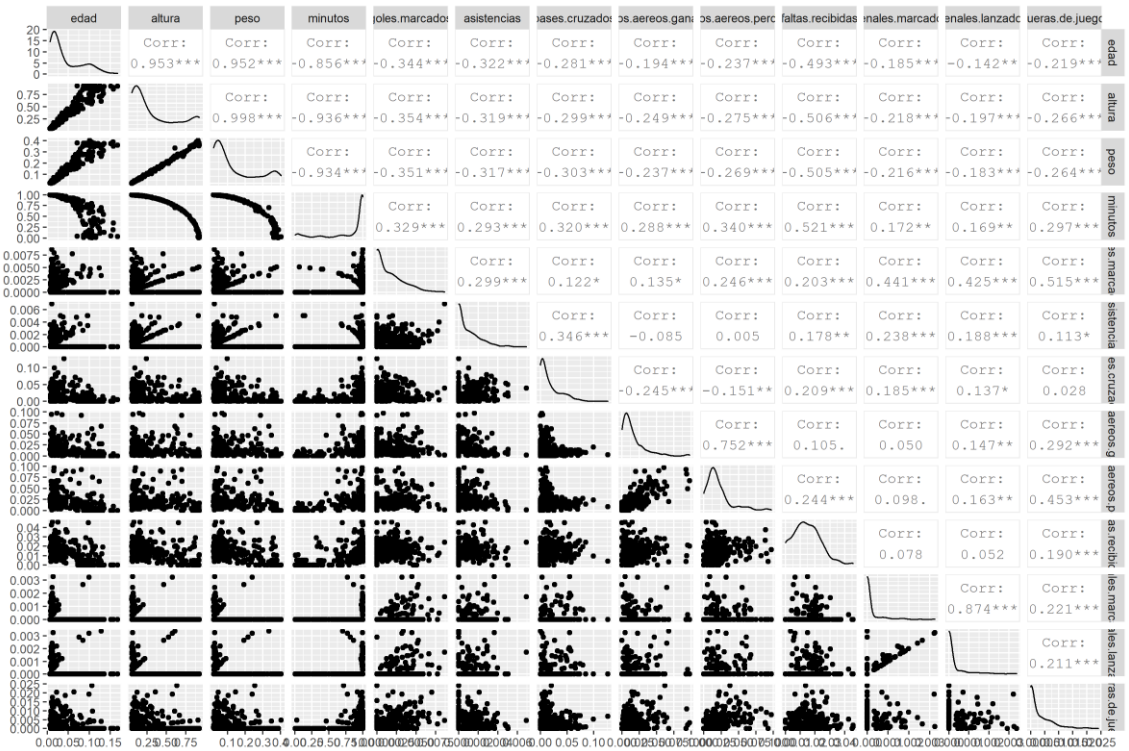
Viendo el resultado sobre el cumplimiento de los supuestos y el valor de la bondad de ajuste (R^2 ajustado), podemos concluir que nuestro modelo no es fiable. Tendríamos que buscar una solución al incumplimiento de los supuestos de homocedasticidad y normalidad.

Modelo de regresión que explica la variable goles.marcados

En este análisis queremos explicar la variable *goles.marcados* con el resto de variables atacantes recopiladas sobre los jugadores con demarcación de centrocampista y delantero.

Análisis de correlación

Vemos los valores de correlación y gráficos de dispersión:



No se perciben correlaciones significativas.

Análisis de regresión

Vamos a ajustar un modelo de regresión lineal múltiple para explicar la variable *goles.marcados* con el resto de variables y evaluaremos su bondad de ajuste.

Modelo

Utilizaremos la función *lm* del paquete *stats*.

```
Call:
lm(formula = goles.marcados ~ ., data = estad.totales.resto.ataq.est,
    na.action = NULL)

Residuals:
       Min         1Q     Median         3Q        Max
-0.0038899 -0.0007734 -0.0002726  0.0003840  0.0062242

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)             9.724e-04  8.747e-04   1.112   0.2671
edad                   -1.427e-02  7.077e-03  -2.016   0.0446 *
altura                  3.066e-03  5.007e-03   0.612   0.5407
peso                   -5.268e-03  1.164e-02  -0.452   0.6513
minutos                 5.209e-05  8.069e-04   0.065   0.9486
asistencias             2.168e-01  7.171e-02   3.024   0.0027 **
pases.cruzados         -4.260e-03  4.354e-03  -0.978   0.3286
duelos.aereos.ganados  -5.464e-03  7.566e-03  -0.722   0.4707
duelos.aereos.perdidos  9.282e-04  7.247e-03   0.128   0.8982
faltas.recibidas        2.465e-03  9.656e-03   0.255   0.7987
penales.marcados        4.359e-01  3.596e-01   1.212   0.2263
penales.lanzados        6.641e-01  3.060e-01   2.170   0.0307 *
fueras.de.juego         1.726e-01  2.065e-02   8.358    2e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.001363 on 319 degrees of freedom
Multiple R-squared:  0.4409,    Adjusted R-squared:  0.4198
F-statistic: 20.96 on 12 and 319 DF,  p-value: < 2.2e-16
```

Se puede apreciar que el RSE es .001 y obtenemos una tasa de error del 89,72%, muy alto.

El R^2 ajustado es .4198, por lo que con estas variables podemos explicar el 42% de la variabilidad de *goles.marcados*. Veamos ahora los R^2 parciales:

```
$adjustment
[1] FALSE

$variable
 [1] "edad"                 "altura"               "peso"                 "minutos"              "asistencias"
 [6] "pases.cruzados"       "duelos.aereos.ganados" "duelos.aereos.perdidos" "faltas.recibidas"     "penales.marcados"
[11] "penales.lanzados"     "fueras.de.juego"

$partial.rsq
 [1] 1.258494e-02 1.174164e-03 6.411667e-04 1.306551e-05 2.785954e-02 2.991872e-03 1.632432e-03 5.142093e-05 2.042630e-04
[10] 4.584904e-03 1.454690e-02 1.796381e-01
```

Vemos que *fueras.de.juego*, *asistencias*, *penales.lanzados* y *edad* son las variables que mejor explican de forma independiente la variabilidad de *goles.marcados*.

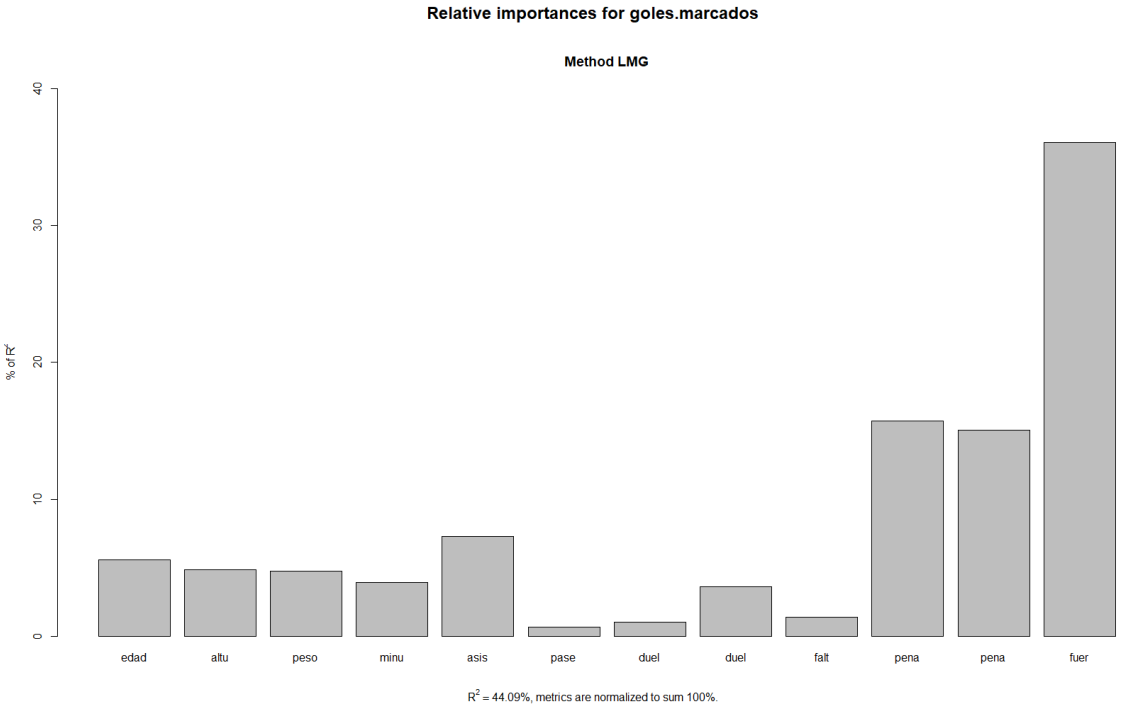
Vamos a interpretar los coeficientes del modelo (pruebas t):

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	9.723736e-04	0.0008747317	1.11162491	2.671365e-01
edad	-1.427064e-02	0.0070773793	-2.01637373	4.459869e-02
altura	3.066112e-03	0.0050069493	0.61237121	5.407284e-01
peso	-5.267825e-03	0.0116442432	-0.45239740	6.512902e-01
minutos	5.209487e-05	0.0008069258	0.06455967	9.485650e-01
asistencias	2.168113e-01	0.0717074419	3.02355432	2.700898e-03
pases.cruzados	-4.260258e-03	0.0043543007	-0.97840238	3.286168e-01
duelos.aereos.ganados	-5.464149e-03	0.0075658004	-0.72221686	4.706902e-01
duelos.aereos.perdidos	9.281598e-04	0.0072468002	0.12807857	8.981675e-01
faltas.recibidas	2.465163e-03	0.0096563105	0.25529041	7.986633e-01
penales.marcados	4.359469e-01	0.3596460065	1.21215548	2.263499e-01
penales.lanzados	6.640569e-01	0.3060149990	2.17001422	3.074344e-02
fueras.de.juego	1.725667e-01	0.0206473926	8.35779800	1.997114e-15

Vemos que sólo las variables *fueras.de.juego*, *asistencias*, *penales.lanzados* y *edad* afectarían significativamente a la variable *goles.marcados* para una cantidad fija en el resto de variables. Tendríamos que comprobar si podemos eliminar el resto de variables mediante técnicas de comparación de modelos.

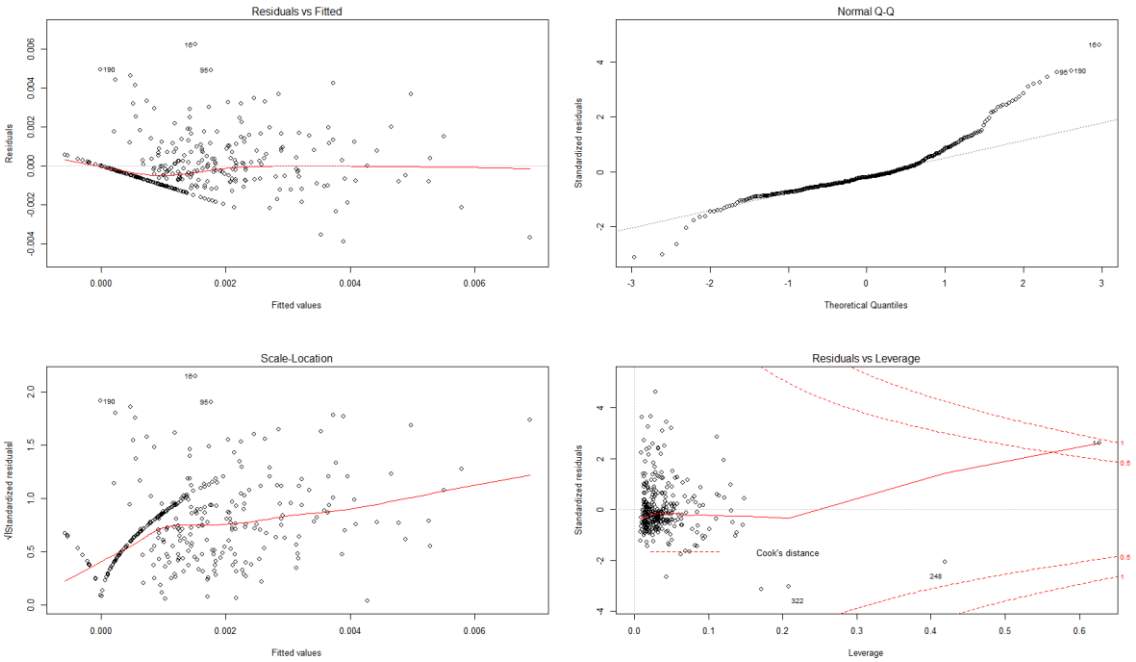
Ahora vamos a estimar la importancia relativa de los predictores:

```
Response variable: goles.marcados
Total response variance: 3.203195e-06
Analysis based on 332 observations

12 Regressors:
edad altura peso minutos asistencias pases.cruzados duelos.aereos.ganados duelos.aereos.perdidos faltas.recibidas penales.marcados penal
es.lanzados fueras.de.juego
Proportion of variance explained by model: 44.09%
Metrics are normalized to sum to 100% (rela=TRUE).

Relative importance metrics:

                          lmg
edad                0.055670624
altura              0.048843481
peso                0.047803530
minutos             0.039231008
asistencias         0.072790918
pases.cruzados      0.006853308
duelos.aereos.ganados  0.010531520
duelos.aereos.perdidos 0.036098533
faltas.recibidas    0.013846365
penales.marcados    0.157231882
penales.lanzados    0.150358418
fueras.de.juego     0.360740413

Average coefficients for different model sizes:

                            1x           2xs           3xs           4xs           5xs           6xs           7xs
edad                -0.016018466 -0.011779892 -0.0099133236 -0.0094591459 -9.751016e-03 -1.038445e-02 -0.0111360429
altura              -0.002104647 -0.002171847 -0.0019779710 -0.0016154461 -1.158806e-03 -6.589552e-04 -0.0001426276
peso                -0.005052298 -0.003107315 -0.0019460042 -0.0013462376 -1.119768e-03 -1.131077e-03 -0.0013031568
minutos              0.001962377  0.001268913  0.0007355338  0.0003499297  9.210475e-05 -6.094719e-05 -0.0001323576
asistencias          0.454051898  0.372307601  0.3246275949  0.2941007418  2.727191e-01  2.567027e-01  0.2442549903
pases.cruzados       0.010738912  0.005830071  0.0029485482  0.0010998309 -1.690112e-04 -1.101362e-03 -0.0018370255
duelos.aereos.ganados   0.014853669  0.007274968  0.0026486285 -0.0005421589 -2.850481e-03 -4.523955e-03 -0.0056874286
duelos.aereos.perdidos  0.024929469  0.019436292  0.0160411961  0.0138027518  1.215087e-02  1.074444e-02  0.0093809818
faltas.recibidas     0.038717918  0.022438536  0.0136047178  0.0088439230  6.240578e-03  4.775141e-03  0.0039101523
penales.marcados     1.656740785  1.468484417  1.3373756444  1.2337784397  1.141330e+00  1.051462e+00  0.9601146253
penales.lanzados     1.331425273  1.147855388  1.0233202919  0.9307245427  8.572063e-01  7.976687e-01  0.7499376656
fueras.de.juego      0.217651257  0.200983024  0.1914690944  0.1857260298  1.819418e-01  1.792089e-01  0.1771026790
                            8xs           9xs          10xs          11xs          12xs
edad                -0.0118910730 -0.0125944372 -1.322334e-02 -1.377594e-02 -1.427064e-02
altura               0.0003850493  0.0009368485  1.539986e-03  2.233487e-03  3.066112e-03
peso                -0.0016141749 -0.0020893647 -2.791682e-03 -3.813241e-03 -5.267825e-03
minutos             -0.0001437812 -0.0001147237 -6.245622e-05 -2.260908e-06  5.209487e-05
asistencias          0.2345070098  0.2270284133  2.216160e-01  2.182047e-01  2.168113e-01
pases.cruzados      -0.0024555936 -0.0029987982 -3.483289e-03 -3.908285e-03 -4.260258e-03
duelos.aereos.ganados  -0.0064096027 -0.0067316666 -6.678280e-03 -6.258820e-03 -5.464149e-03
duelos.aereos.perdidos  0.0079488081  0.0063967710  4.709946e-03  2.888319e-03  9.281598e-04
faltas.recibidas     0.0033628213  0.0029867843  2.714183e-03  2.530755e-03  2.465163e-03
penales.marcados     0.8655860835  0.7671408759  6.640180e-01  5.546602e-01  4.359469e-01
penales.lanzados     0.7127867342  0.6854116220  6.675334e-01  6.597485e-01  6.640569e-01
fueras.de.juego      0.1754437905  0.1741669503  1.732549e-01  1.727128e-01  1.725667e-01
```



Observamos que las variables *fueras.de.juego*, penales.marcados, *penales.lanzados* y *asistencias* son las que más importancia relativa tienen en el modelo.

Vemos ahora si se cumplen los supuestos del modelo y la existencia de datos atípicos:



En el gráfico Residuals vs Fitted podemos apreciar un patrón de embudo. Se cumple el supuesto de linealidad, pero no el de homocedasticidad.
En el gráfico Normal Q-Q se aprecia que los residuos se apartan de la diagonal en las colas, sobre todo en la derecha, por lo que no cumple el supuesto de normalidad.

En el gráfico *Residuals vs Leverage* se aprecia que ningún punto sobrepasa la distancia de Cook, aunque el punto 1 está en el límite, por lo que no tenemos datos atípicos.

Ahora realizaremos el test de Durbin-Watson mediante la función *durbinWatsonTest* del paquete *car*, para comprobar el supuesto de independencia:

```
lag Autocorrelation D-W Statistic p-value
  1    -0.004568233      2.000971   0.484
Alternative hypothesis: rho > 0
```

No hay correlación en los residuos, por lo que cumplimos el supuesto de independencia.

Viendo el resultado sobre el cumplimiento de los supuestos y el valor de la bondad de ajuste (R^2 ajustado), podemos concluir que nuestro modelo no es fiable. Tendríamos que buscar una solución al incumplimiento de los supuestos de homocedasticidad y normalidad.

Conclusiones

Después de la realización de los tres modelos de regresión, podemos concluir que debemos buscar una forma de mejorarlos, ya que ninguno ha conseguido un buen resultado. Otra opción sería probar otros modelos más complejos o buscar (o crear) más variables sobre las estadísticas de estos jugadores, que ayuden a mejorar los modelos.

Análisis de clúster

En este análisis pretendemos agrupar los jugadores en distintos clústeres para intentar encontrar similitudes entre ellos, en base a ciertas características.

Para ello, vamos a dividir los datos por demarcación, de modo que vamos a tener cuatro sets de datos distintos sobre los que realizaremos el análisis de clúster.

Para cada demarcación utilizaremos variables distintas, en base a las que creemos que son representativas de cada demarcación. Las variables *edad*, *altura*, *peso* y *minutos* serán comunes a los cuatro sets de datos.

Los NAs de las variables *edad*, *altura*, *peso, recuperaciones, duelos.aereos.ganados, duelos.aereos.perdidos* y *penales.concedidos* se sustituirán por la media respectiva.

Además, se estandarizarán todas las variables por tener diferentes unidades. Para este cometido, utilizaremos la función *decostand* del paquete *vegan*.

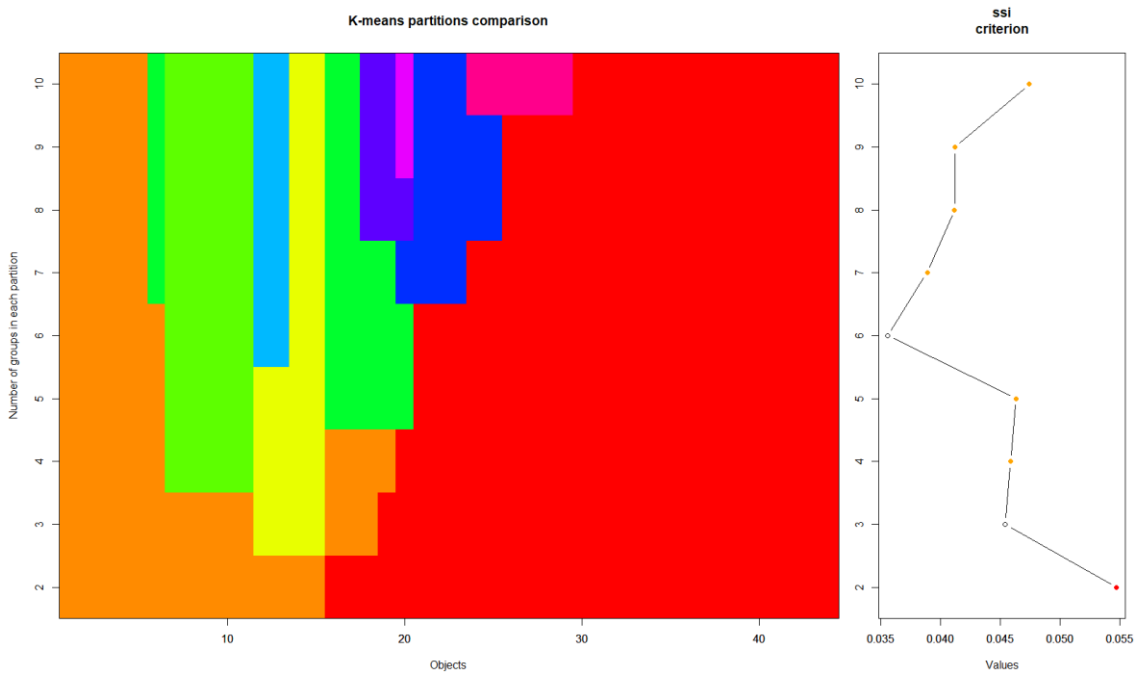
El método de clustering elegido será K-Means y analizaremos soluciones de entre 2 y 10 grupos.

El código para la realización de este análisis se recoge en:

scripts/clustering_analysis.R

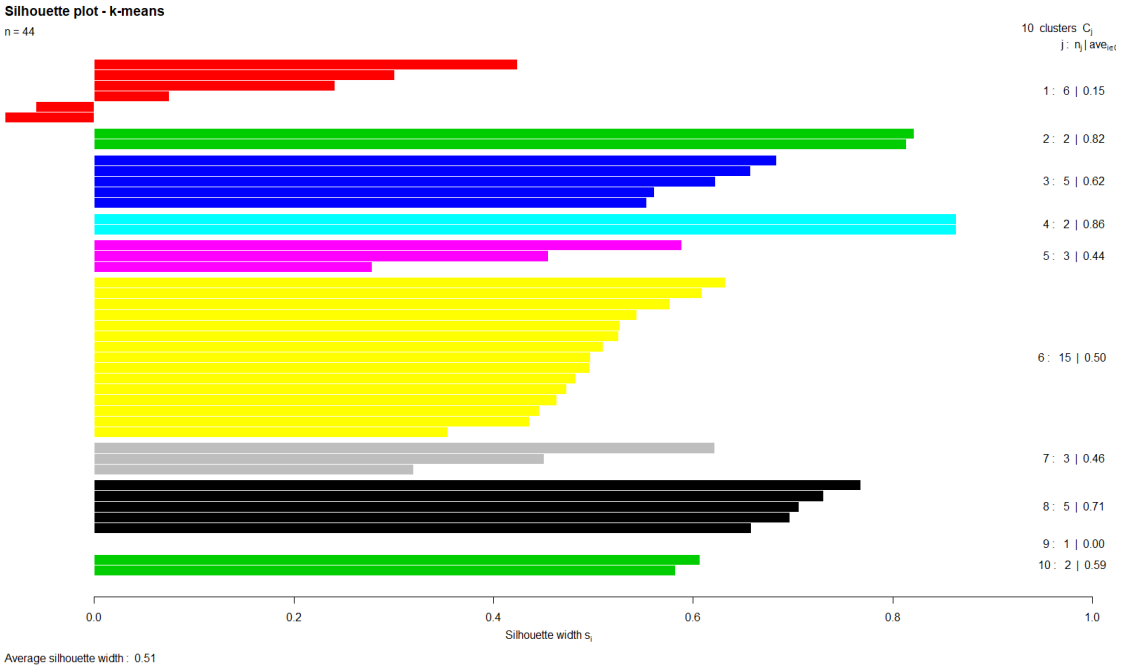
Grupos de porteros

En primer lugar vamos a ver qué número de grupos sería el recomendable. Para ello, utilizaremos la función *cascadeKM* del paquete *vegan*:



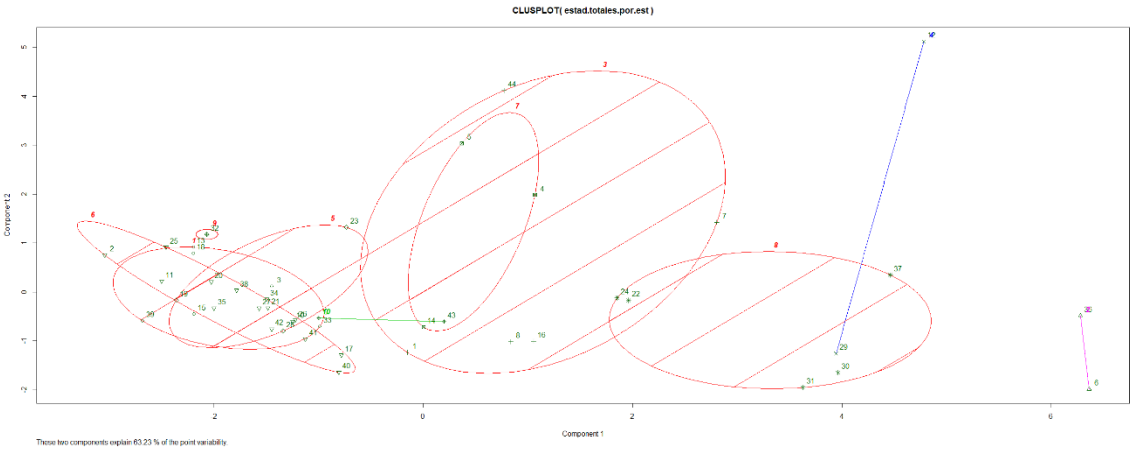
Aunque el resultado nos recomienda 2 grupos, vamos a elegir la segunda mejor opción para obtener una diferenciación más amplia, es decir, 10 grupos.

Una vez aplicado el método K-Means para 10 grupos con la función *kmeans* del paquete *stats*, analizamos su silueta con la función *silhouette* del paquete *cluster*:



Se pueden apreciar los 10 grupos bien formados, aunque en el grupo 1 hay dos casos que no agrupa correctamente.

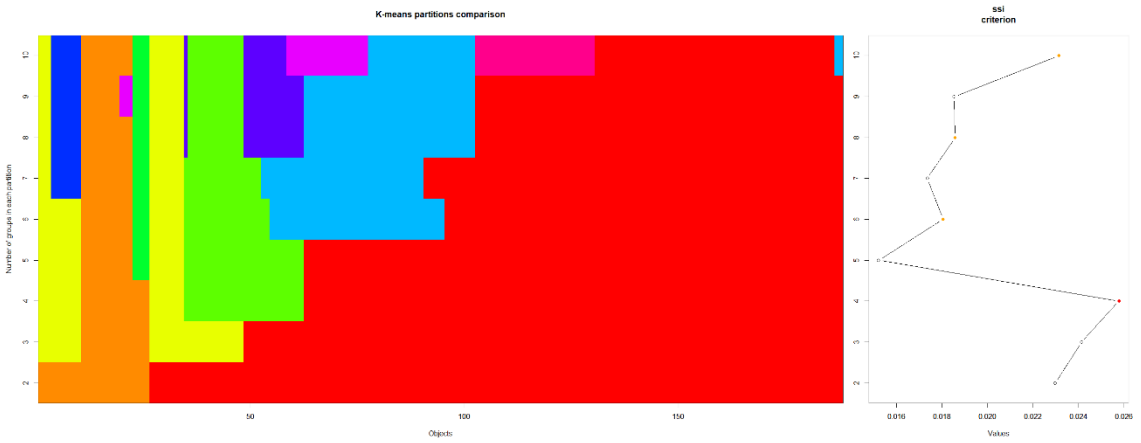
Veamos cómo se reparten los diferentes porteros por los grupos. Lo haremos a través de un gráfico con la función *cusplot* del paquete *cluster*:



En el primer grupo están porteros como Remiro de la Real Sociedad, Pacheco del Alavés o Cillessen del Valencia, que son porteros titulares de gran nivel. En el grupo 6 podemos ver a los porteros que se encuentran en equipos top como Oblak del At. Madrid, Ter Stegen del FC Barcelona o Courtois del Real Madrid. En el resto de grupos se reparten los porteros no titulares, donde destaca el grupo 9 con sólo un integrante, el portero Roberto. El meta suplente del Alavés jugó varios partidos después del parón cuajando buenas actuaciones, pero encajando muchos goles en pocos partidos. Quizás por ello no se encuentre entre los otros grupos de porteros con menos minutos.

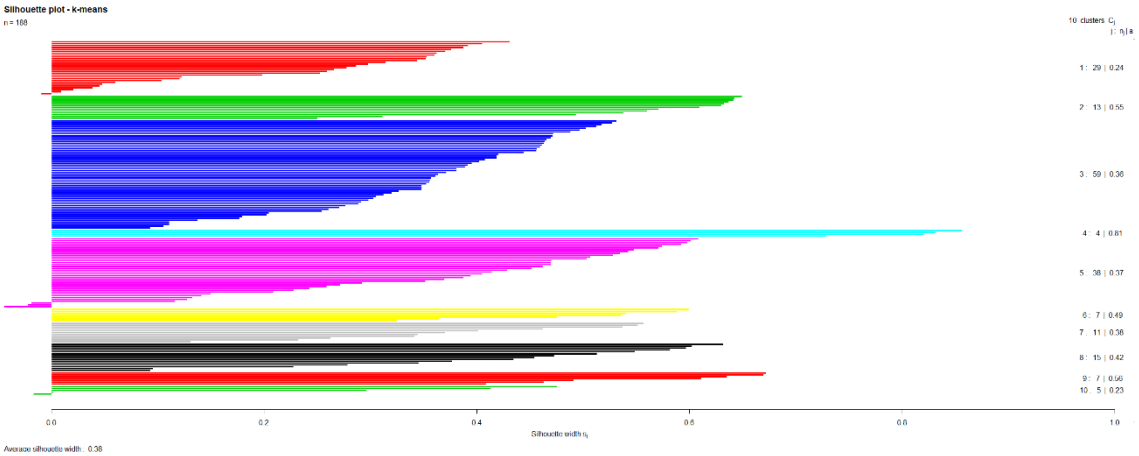
Grupos de defensas

Veamos ahora qué número de grupos sería el recomendable para la demarcación de defensa:



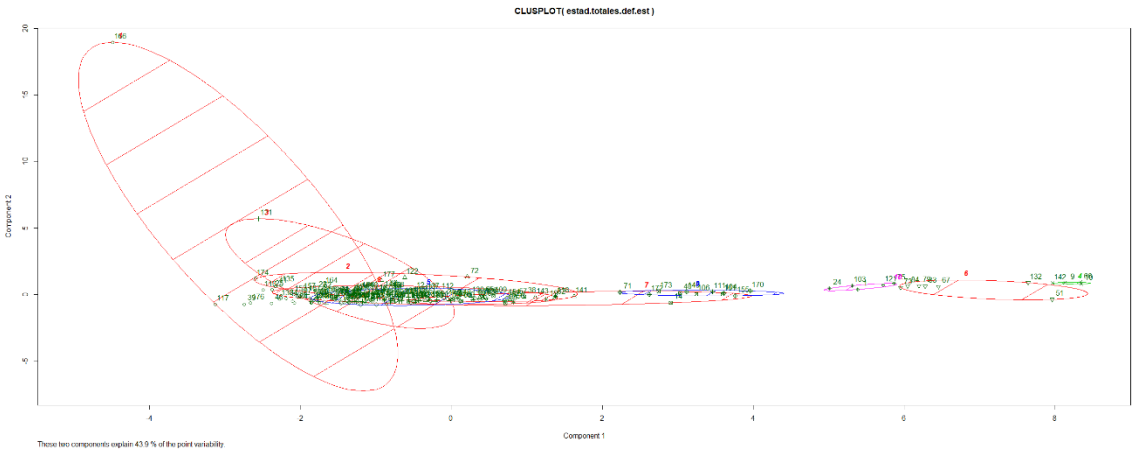
Aunque el resultado nos recomienda 4 grupos, vamos a elegir la segunda mejor opción para obtener una diferenciación más amplia, es decir, 10 grupos.

Aplicamos el método K-Means y analizamos su silueta:



Vemos los 10 grupos con los defensas bastante repartidos por todos ellos. También se aprecia algún caso que no termina de clasificar bien.

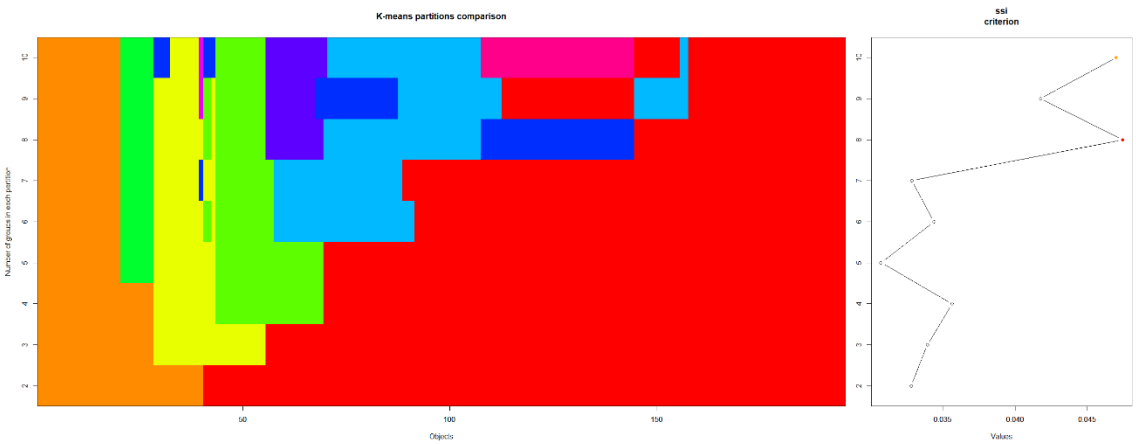
Ahora, representamos gráficamente el reparto de defensas entre los grupos:



En el primer grupo se aprecian defensas centrales que han jugado de titular este año, como son Piqué del FC Barcelona, Ramos del Real Madrid o Koundé del Sevilla o Djené del Getafe. El grupo 3 está poblado sobretodo de laterales titulares como Navas del Sevilla, Carvajal del Real Madrid o Gayá del Valencia. Entre el resto de grupos se reparten los defensas menos habituales en el once titular de sus respectivos equipos.

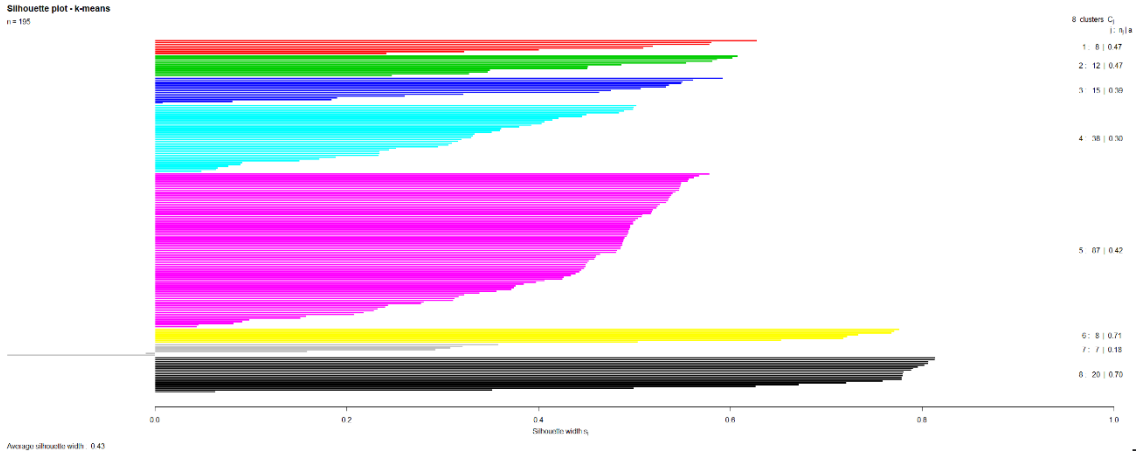
Grupos de centrocampistas

Buscamos el número de grupos recomendado para esta demarcación:



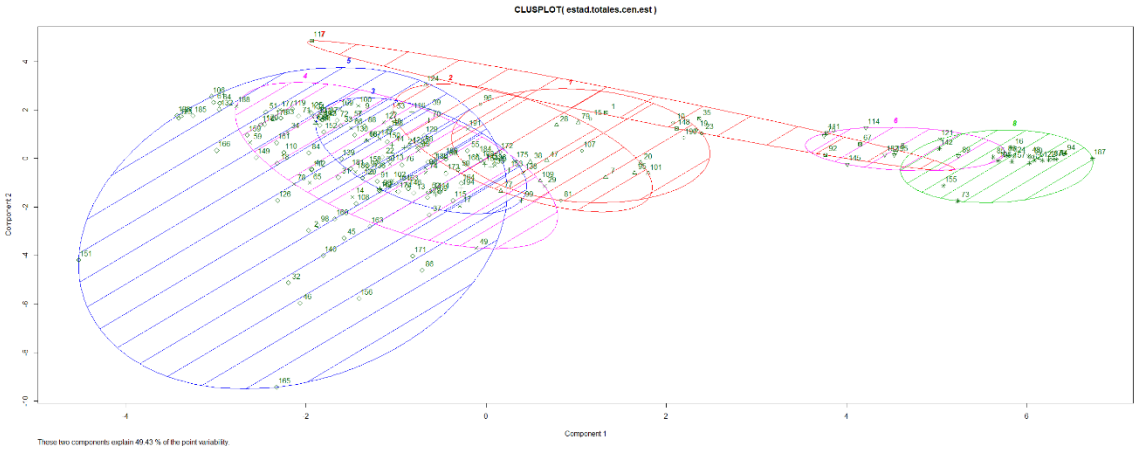
Se puede apreciar que el número óptimo de grupos es 8.

Después de aplicar el método K-Means para 8 grupos, analizamos su silueta:



Los 8 grupos están bien diferenciados y sólo un par de casos parecen no estar bien clasificados.

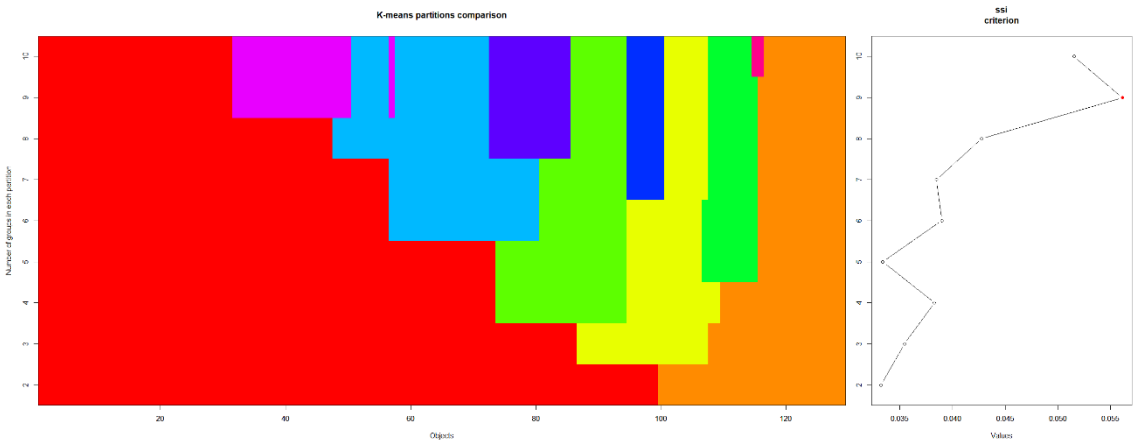
Veamos gráficamente el reparto de centrocampistas entre los grupos:



En el grupo 5 podemos ver a los mejores centrocampistas de La Liga como Casemiro del Real Madrid, Odegaard de la Real Sociedad o Saúl del At. Madrid. En estas agrupaciones no están tan diferenciados las distintas subposiciones dentro de la demarcación de centrocampista, como si pudimos ver en la agrupación de defensas. Parece que lo que más ha primado a la hora de agrupar ha sido el número de minutos jugados.

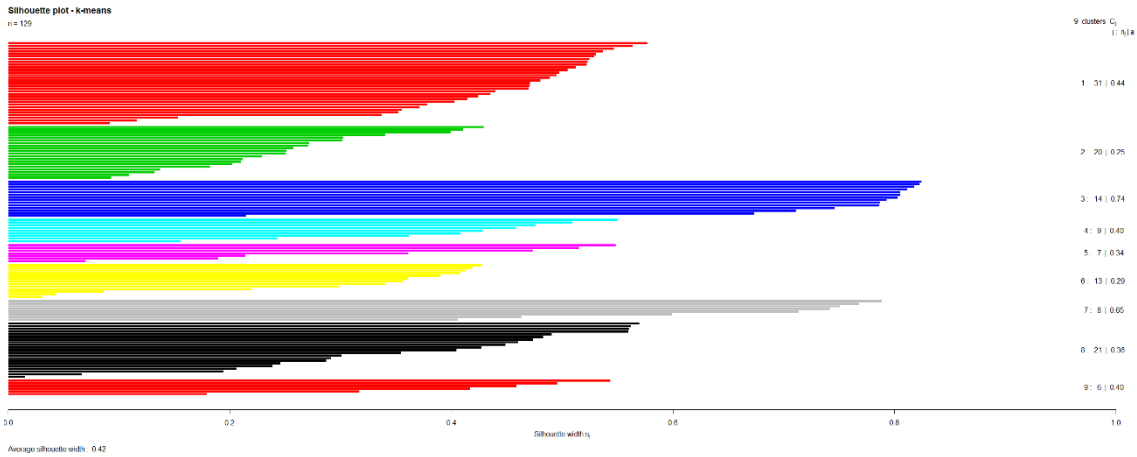
Grupos de delanteros

Por último, vamos con los delanteros. Vamos a ver cuál es el número óptimo de grupos:



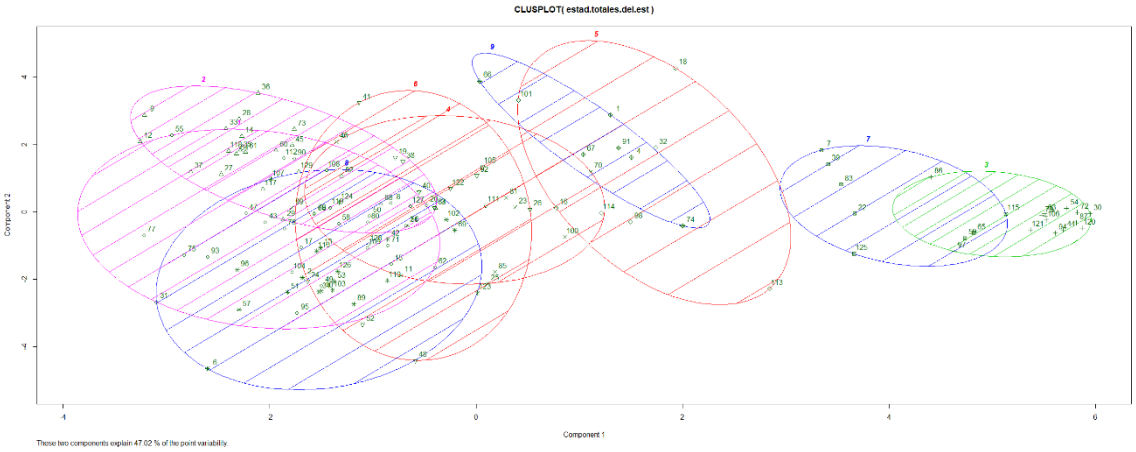
En este caso, 9 grupos sería lo más óptimo.

Tras aplicar el método K-Means sobre los datos de delanteros, analizamos su silueta:



Podemos ver los 9 grupos bien diferenciados y de forma correcta.

Veamos la distribución de los delanteros entre los diferentes grupos:



En el primer grupo podemos apreciar a los delanteros titulares más goleadores de La Liga, como Messi del FC Barcelona, Benzema del Real Madrid o Moreno del Villarreal. En el grupo 2 también tenemos delanteros titulares que han sido protagonistas en sus equipos, pero que quizás no son tan goleadores como los del primero. Por ejemplo, Carlos Fernández del Granada o Adrián López de Osasuna. En el resto de grupos se distribuyen los delanteros que no han tenido tantos minutos en sus equipos.

Conclusiones

En este análisis se han podido hacer agrupaciones bastante buenas de los jugadores por demarcación. El mayor 'pero' que podemos poner es que sobretodo ha primado la cantidad de minutos disputados por los jugadores. Esto podría solucionarse si calculásemos cada estadística de los jugadores por cada 90 minutos.

También hemos podido ver que en alguna demarcación, como en los centrocampistas, no diferenciaba demasiado bien los tipos de subposiciones dentro de la propia demarcación. Para esto nos podría ayudar un mayor número de variables con otras características de los jugadores.

Conclusiones finales

Al margen de las conclusiones que detallo en cada análisis, las principales que extraigo de la realización de este trabajo son:

- La estadística nos proporciona grandes técnicas para encontrar valor en los datos mediante diferentes tipos de análisis, y el lenguaje R nos ofrece grandes medios para realizarlos.

- En el mundo del fútbol se genera una cantidad ingente de datos cada día y su estudio puede ser de gran ayuda para directivas, cuerpos técnicos y los propios jugadores.

- Tener una cantidad muy grande de datos no te garantiza nada si no sabemos buscar en ellos. Por ello, con los análisis adecuados, podremos obtener información que nos otorgue ventaja sobre el contrario.

- A veces, no es necesario realizar grandes operaciones o utilizar los modelos más complejos. Con un buen gráfico o una simple regresión lineal se puede conseguir el mejor resultado.

- También tenemos que entender que, por muchos datos que tengamos, a veces no es posible llegar a una conclusión o conseguir un modelo fiable.

- Por último, comentar lo mucho que he disfrutado realizando este máster y, en particular, este trabajo.

Páginas web utilizadas

- Fuente de datos de los jugadores de La Liga 2019-2020: https://fbref.com/es/
- Guía para hacer scraping con el paquete *rvest* de R: https://sportsdatachallenge.wordpress.com/2016/09/21/a-total-beginners-guide-to-web-scraping-football-data-part-1/
- Paquete *FootballBadges* de Jesús Lagos, para dibujar los escudos de los equipos en scatterplots: https://github.com/Jelagmil/FootballBadges

ANEXO 1: Código en R para el desarrollo de este trabajo

A continuación se adjunta todo el código en R que se ha utilizado para el desarrollo de este trabajo de fin de máster.

scraping_fbref.R

```
###### SCRIPT PARA HACER SCRAPING DEL PESO Y ALTURA DE LOS JUGADORES
EN FBREF ######

# Instalamos y cargamos los paquetes que hagan falta
packages = c("tidyverse", "rvest")
inst <- packages %in% installed.packages()
if(length(packages[!inst]) > 0) install.packages(packages[!inst])
lapply(packages, require, character.only=T)


# Hacemos scraping de FBRef de las URL del área personal de todos los
jugadores
pagina <- read_html("https://widgets.sports-
reference.com/wg.fcgi?css=1&site=fb&url=%2Fes%2Fcomps%2F12%2Fstats%2FE
stadisticas-de-La-Liga&div=div_stats_standard")
urls <- pagina %>%
  html_nodes(".right+ .left a") %>%
  html_attr("href") %>%
  data.frame() %>%
  filter(row_number() %% 2 == 1)


# Añadimos el campo RL que nos servirá para hacer join con otras
tablas
url.jugadores <- cbind(data.frame(RL = 1:nrow(urls)), url.jugadores =
urls$.)
url.jugadores$url.jugadores <-
as.character(url.jugadores$url.jugadores)


# Función para hacer scraping del peso y altura de los jugadores
scrap.height.weight <- function(players){
  data <- data.frame()
  for(i in 1:nrow(players)){
    page <- tryCatch(read_html(players[i,2]),
                     error = function(e) e)
    label <- page %>%
      html_node("p:nth-child(4)") %>%
      html_text()
    data <- rbind(data, as.data.frame(label))
  }
  return(data)
}


# Hacemos scraping de FBRef de la altura y peso del área personal de
todos los jugadores
datos.fisicos.jugadores <- cbind(url.jugadores,
scrap.height.weight(url.jugadores))
```

```
# Ponemos a NA los que no tienen estos datos registrados
datos.fisicos.jugadores$label <-
as.character(datos.fisicos.jugadores$label)
datos.fisicos.jugadores[is.na(as.numeric(substring(datos.fisicos.jugad
ores$label, 1, 1))),]$label <- NA


# Limpiamos los datos y separamos en altura y peso
datos.fisicos.jugadores <- datos.fisicos.jugadores %>%
  rowwise() %>%
  mutate(altura = as.numeric(strsplit(label, "[cm]")[[1]][1]), weight
= strsplit(label, "[, ]")[[1]][2]) %>%
  mutate(peso = as.numeric(str_squish(ifelse(!is.na(weight),
                                             ifelse(grepl("kg",
weight, fixed = T),
                                                    strsplit(weight,
"[kg]")[[1]][1],
                                                    NA),
                                             NA)))) %>%
  select(RL, altura, peso)

# Guardamos los datos en formato CSV
write.table(datos.fisicos.jugadores,
"data/datos_fisicos__jugadores_liga_19_20.csv", sep = ";", row.names =
F)
```

preprocesing.R

```
###### PREPROCESAMIENTO DE DATOS ######

# Instalamos y cargamos los paquetes que hagan falta
packages = c("tidyverse")
inst <- packages %in% installed.packages()
if(length(packages[!inst]) > 0) install.packages(packages[!inst])
lapply(packages, require, character.only=T)


# Cargamos los datos
estad.estandar.jug <-
read.csv("data/estadisticas_estandar_jugadores_liga_19_20.csv",
stringsAsFactors = F, skip = 1, fileEncoding = "UTF-8")
estad.diversas.jug <-
read.csv("data/estadisticas_diversas_jugadores_liga_19_20.csv",
stringsAsFactors = F, skip = 1, fileEncoding = "UTF-8")
estad.porteros.jug <- read.csv("data/estadísticas_porteros.csv",
stringsAsFactors = F, skip = 1, fileEncoding = "UTF-8")
datos.fisicos.jug <-
read.csv("data/datos_fisicos_jugadores_liga_19_20.csv",
stringsAsFactors = F, sep = ";")

estad.totales.jug <- estad.estandar.jug %>%
  inner_join(estad.diversas.jug, by = "RL") %>%
  inner_join(datos.fisicos.jug, by = "RL" ) %>%
  left_join(estad.porteros.jug, by = c("Jugador.x" = "Jugador",
"Posc.x" = "Posc"))
```

```r
# Seleccionamos los campos de los datos estadísticos a analizar
estad.totales.jug <- estad.totales.jug %>%
  rowwise() %>%
  mutate(posicion = ifelse(nchar(Posc.x) == 4,
                           substr(Posc.x, 1, 2),
                           Posc.x),
         posicion.alternativa = ifelse(nchar(Posc.x) == 4,
                                       substr(Posc.x, 3, 4),
                                       NA)) %>%
  select(nombre = Jugador.x,
         pais = País.x,
         edad = Edad.x,
         altura,
         peso,
         equipo = Equipo.x,
         posicion,
         posicion.alternativa,
         partidos.jugados = PJ.x,
         partidos.titular = Titular.x,
         minutos = Mín.x,
         goles.marcados = Gls.,
         goles.propia.puerta = GC.x,
         goles.encajados = GC.y,
         asistencias = Ass,
         pases.cruzados = Pcz,
         recuperaciones = Recup.,
         intercepciones = Int,
         tackleos.ganados = TklG,
         tiros.puerta.recibidos = DaPC,
         paradas = Salvadas,
         duelos.aereos.ganados = Ganados,
         duelos.aereos.perdidos = Perdidos,
         faltas.cometidas = Fls,
         faltas.recibidas = FR,
         tarjetas.amarillas = TA.x,
         tarjetas.rojas = TR.x,
         segunda.amarilla = X2a.amarilla,
         penales.marcados = TP,
         penales.lanzados = TPint.x,
         penales.concedidos = Penal.concedido,
         penales.recibidos = TPint.y,
         penales.parados = PD.1,
         fueras.de.juego = PA)


# Limpiamos los datos y les damos el formato adecuado
estad.totales.jug$nombre <- gsub("[\\].*", "",
estad.totales.jug$nombre) # Nos quedamos con el primer nombre
estad.totales.jug$pais <- gsub(".* ", "", estad.totales.jug$pais)
# Nos quedamos con la abreviatura en mayúsculas


# Guardamos los datos en formato CSV
write.table(estad.totales.jug,
"data/estadisticas_totales_jugadores_liga_19_20.csv", sep = ";",
row.names = F)
```

descriptive_analysis.R

```
###### ANÁLISIS DESCRIPTIVO ######


# Instalamos y cargamos los paquetes que hagan falta
packages = c("tidyverse")
inst <- packages %in% installed.packages()
if(length(packages[!inst]) > 0) install.packages(packages[!inst])
lapply(packages, require, character.only=T)

## Librería para hacer scatterplots con los escudos de La Liga
# devtools::install_github('jelagmil/FootballBadges', build_opts =
c("--no-resave-data", "--no-manual"))
library(FootballBadges)


# Cargamos los datos y hacemos una primera observación
estad.totales.jug <-
read.csv("data/estadisticas_totales_jugadores_liga_19_20.csv",
stringsAsFactors = F, sep = ";")
str(estad.totales.jug)
summary(estad.totales.jug)


# Convertimos a factor las variables de equipo, país y posiciones
estad.totales.jug <- estad.totales.jug %>%
  mutate(pais = as.factor(pais),
         equipo = as.factor(equipo),
         posicion = as.factor(posicion),
         posicion.alternativa = as.factor(posicion.alternativa))
str(estad.totales.jug)
summary(estad.totales.jug)


# Cantidad de jugadores por país
(num.por.pais.jug <- sort(table(estad.totales.jug$pais), decreasing =
T))
(per.por.pais.jug <- sort(round(prop.table(num.por.pais.jug) * 100,
2), decreasing = T))

## 10 países que más jugadores aportan a La Liga
(country.plot <- num.por.pais.jug %>%
  as.data.frame() %>%
  top_n(10) %>%
  transmute(Country = Var1, Freq) %>%
  ggplot(aes(Country, Freq, fill = Country)) +
    geom_bar(stat = "identity") +
    theme(legend.position = "None") +
    labs(title = "Top 10 de países con más jugadores",
         x = "País",
         y = "Total") +
    geom_text(aes(label=Freq), vjust=-0.25, size = 2))

ggsave("plots/country_plot.png",
       country.plot)


# Edad de los equipos
(age.plot <- estad.totales.jug %>%
```

```
  ggplot(aes(reorder(equipo, edad, mean), edad, fill = equipo)) +
    geom_boxplot() +
    theme(legend.position = "None",
        axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
+
    labs(title = "Edad de los equipos de La Liga",
         x = "Equipo",
         y = "Edad"))

ggsave("plots/age_plot.png",
       age.plot)


# Goles por equipo
goles.equipo <- estad.totales.jug %>%
  group_by(equipo) %>%
  summarise(goles.marcados = sum(goles.marcados, na.rm = T),
            goles.encajados = sum(goles.encajados, na.rm = T)) %>%
  mutate(average.general = goles.marcados - goles.encajados) %>%
  arrange(-average.general)

## Goles marcados y encajados por equipo (barras)
(goals.plot <- goles.equipo %>%
  gather("Type", "Value",-c(equipo,average.general)) %>%
  ggplot(aes(equipo, Value, fill = Type)) +
  geom_bar(position = "dodge", stat = "identity") +
  theme(legend.title = element_blank(),
        legend.position="top",
        axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
+
  labs(title = "Golaverage de los equipos de La Liga",
       x = "Equipo",
       y = "Goles") +
  scale_fill_manual(values=c("red1", "seagreen3")))

ggsave("plots/goals_plot.png",
       goals.plot)

## Goles marcados y encajados por equipo (scatterplot)
equipos <- as.data.frame(goles.equipo$equipo)
codigo.equipo <-  c("BCN", "RMD", "ATM", "SEV", "VIL", "GET", "GRA",
"RSO", "ATH", "LEV", "OSA", "VCF", "BET",
                    "CDV", "VLL", "EIB", "LEG", "ALA", "MAL", "ESP")
codigo.equipo.dt <- as.data.frame(cbind(equipos, codigo.equipo))
names(codigo.equipo.dt) <- c("equipo", "codigo")

goles.equipo.cod <- goles.equipo %>%
  inner_join(codigo.equipo.dt, by = c("equipo" = "equipo"))

(goals.badges.plot <- PlotXY_Badges(goles.equipo.cod[,
c(5,3,2)],"ESP"))

ggsave("plots/goals_badges_plot.png",
       goals.badges.plot)

## Average general por equipo
(average.plot <- ggplot(goles.equipo, aes(reorder(equipo,
average.general), average.general, fill = equipo)) +
  geom_bar(stat = "identity") +
  theme(legend.position = "None") +
```

```
  coord_flip() +
  labs(title = "Average general por equipo",
       x = "Equipo",
       y = "Goles"))

ggsave("plots/average_plot.png",
       average.plot)


# Porteros que más penales paran
(penalties.plot <- estad.totales.jug %>%
  filter(posicion == "PO") %>%
  group_by(nombre) %>% # Agrupamos por nombre del portero, por si hay
porteros que han jugado en 2 equipos
  summarise(penales.parados = sum(penales.parados, na.rm = T)) %>%
  filter(penales.parados > 0) %>%
  arrange(-penales.parados) %>%
  ggplot(aes(reorder(nombre, penales.parados), penales.parados, fill =
nombre)) +
    geom_bar(stat = "identity") +
    theme(legend.position = "None") +
    coord_flip() +
    labs(title = "Penales parados por portero",
         x = "Portero",
         y = "Penales"))

ggsave("plots/penalties_plot.png",
       penalties.plot)


# Equipos que más tiros a puerta reciben
(shots.plot <- estad.totales.jug %>%
  filter(posicion == "PO") %>%
  group_by(equipo) %>%
  summarise(tiros.recibidos = sum(tiros.puerta.recibidos, na.rm = T))
%>%
  arrange(-tiros.recibidos) %>%
  ggplot(aes(reorder(equipo, tiros.recibidos), tiros.recibidos, fill =
equipo)) +
    geom_bar(stat = "identity") +
    theme(legend.position = "None") +
    coord_flip() +
    labs(title = "Tiros a puerta recibidos por equipo",
         x = "Equipo",
         y = "Tiros"))

ggsave("plots/shots_plot.png",
       shots.plot)

# Tarjetas amarillas por posición
(cards.plot <- estad.totales.jug %>%
  ggplot(aes(posicion, tarjetas.amarillas, fill = posicion)) +
  geom_boxplot() +
  theme(legend.position = "None") +
  labs(title = "Tarjetas amarillas por posición",
       x = "Posición",
       y = "Tarjetas amarillas"))

ggsave("plots/cards_plot.png",
       cards.plot)
```

```r
# Altura por posición
(height.plot <- estad.totales.jug %>%
  ggplot(aes(posicion, altura, fill = posicion)) +
  geom_boxplot() +
  theme(legend.position = "None") +
  labs(title = "Altura por posición",
       x = "Posición",
       y = "Altura"))

ggsave("plots/height_plot.png",
       height.plot)
```

inferential_analysis.R

```r
###### ANÁLISIS INFERENCIAL ######


# Instalamos y cargamos los paquetes que hagan falta
packages = c("tidyverse","WRS2")
inst <- packages %in% installed.packages()
if(length(packages[!inst]) > 0) install.packages(packages[!inst])
lapply(packages, require, character.only=T)


# Cargamos los datos
estad.totales.jug <-
read.csv("data/estadisticas_totales_jugadores_liga_19_20.csv",
stringsAsFactors = F, sep = ";")


# Preparamos los datos

## Convertimos a factor las variables de equipo, país y posiciones
estad.totales.jug <- estad.totales.jug %>%
  mutate(pais = as.factor(pais),
         equipo = as.factor(equipo),
         posicion = as.factor(posicion),
         posicion.alternativa = as.factor(posicion.alternativa))


# Comparación de goles marcados de los cuatro primeros

## Filtramos los datos por los cuatro primeros
estad.totales.jug.primeros <- estad.totales.jug %>%
  filter(equipo %in% c("Real Madrid", "Barcelona", "Atlético Madrid",
"Sevilla")) %>%
  droplevels()

## Observamos las distribuciones de los goles marcados para cada
equipo
(goles.marcados.plot <- ggplot(estad.totales.jug.primeros, aes(equipo,
goles.marcados, fill = equipo)) +
  geom_boxplot() +
  theme(legend.position = "None") +
  labs(title = "Goles marcados de los 4 primeros clasificados",
```

```
        x = "Equipo",
        y = "Goles"))

ggsave("plots/goals scored first four plot.png",
       goles.marcados.plot)

## Realizamos el ANOVA
t1way(goles.marcados ~ equipo, estad.totales.jug.primeros)


# Comparativa del número de asistencias de CC y DL

## Agrupamos por jugador para que no haya jugadores repetidos y
filtramos CC y DL
estad.totales.jug.cc.dl <- estad.totales.jug %>%
  group_by(nombre, pais, edad, altura, peso) %>%
  summarise(equipo = last(equipo),
            posicion = last(posicion),
            posicion.alternativa = last(posicion.alternativa),
            across(partidos.jugados:fueras.de.juego, sum)) %>%
  filter(posicion %in% c("CC", "DL")) %>%
  droplevels()

## Observamos las distribuciones de las asistencias repartidas para
cada posición
(asistencias.cc.dl.plot <- ggplot(estad.totales.jug.cc.dl,
aes(posicion, asistencias, fill = posicion)) +
    geom_boxplot() +
    theme(legend.position = "None") +
    labs(title = "Asistencias repartidas por los CC y DL",
         x = "Posición",
         y = "Asistencias"))

ggsave("plots/assists_plot.png",
       asistencias.cc.dl.plot)

## Realizamos la prueba de Yuen
yuen(asistencias ~ posicion, estad.totales.jug.cc.dl)


# Comparación de tarjetas entre DF y CC

## Agrupamos por jugador para que no haya jugadores repetidos y
filtramos DF y CC
estad.totales.jug.df.cc <- estad.totales.jug %>%
  group_by(nombre, pais, edad, altura, peso) %>%
  summarise(equipo = last(equipo),
            posicion = last(posicion),
            posicion.alternativa = last(posicion.alternativa),
            across(partidos.jugados:fueras.de.juego, sum)) %>%
  filter(posicion %in% c("DF", "CC")) %>%
  droplevels()

## Observamos las distribuciones de las tarjetas recibidas para cada
posición
(tarjetas.df.cc.plot <- ggplot(estad.totales.jug.df.cc, aes(posicion,
tarjetas.amarillas, fill = posicion)) +
    geom_boxplot() +
    theme(legend.position = "none") +
    labs(title = "Tarjetas recibidas por los DF y CC",
```

```
         x = "Posición",
         y = "Tarjetas"))

ggsave("plots/yellow cards plot.png",
       tarjetas.df.cc.plot)

## Realizamos la prueba de Yuen
yuen(tarjetas.amarillas ~ posicion, estad.totales.jug.df.cc)


# Comparación de alturas para las diferentes demarcaciones

## Agrupamos por jugador para que no haya jugadores repetidos
estad.totales.jug.todas.posiciones <- estad.totales.jug %>%
  group_by(nombre, pais, edad, altura, peso) %>%
  summarise(equipo = last(equipo),
            posicion = last(posicion),
            posicion.alternativa = last(posicion.alternativa),
            across(partidos.jugados:fueras.de.juego, sum))

## Observamos las distribuciones de las alturas para cada posción
(alturas.plot <- ggplot(estad.totales.jug.todas.posiciones,
aes(posicion, altura, fill = posicion)) +
    geom_boxplot() +
    theme(legend.position = "none") +
    labs(title = "Altura por posición",
         x = "Posición",
         y = "Altura"))

ggsave("plots/heights_plot.png",
       alturas.plot)

## Realizamos el ANOVA
t1way(altura ~ posicion, estad.totales.jug.todas.posiciones)

## Pruebas post-hoc
lincon(altura ~ posicion, estad.totales.jug.todas.posiciones)


# Comparación de recuperaciones de los DF de los cuatro primeros

## Filtramos los datos por los defensas de los cuatro primeros
estad.totales.jug.primeros.df <- estad.totales.jug %>%
  filter(posicion %in% c("DF")) %>%
  filter(equipo %in% c("Real Madrid", "Barcelona", "Atlético Madrid",
"Sevilla")) %>%
  droplevels()

## Observamos las distribuciones de las recuperaciones de los DF para
cada equipo
(recuperaciones.plot <- ggplot(estad.totales.jug.primeros.df,
aes(equipo, recuperaciones, fill = equipo)) +
    geom_boxplot() +
    theme(legend.position = "None") +
    labs(title = "Recuperaciones de los defensas",
         subtitle = "4 equipos mejor clasificados",
         x = "Equipo",
         y = "Recuperaciones"))

ggsave("plots/recovers_plot.png",
```

```
       recuperaciones.plot)

## Prueba de normalidad
shapiro.test(estad.totales.jug.primeros.df[estad.totales.jug.primeros.
df$equipo == "Real Madrid",]$recuperaciones)
shapiro.test(estad.totales.jug.primeros.df[estad.totales.jug.primeros.
df$equipo == "Barcelona",]$recuperaciones)
shapiro.test(estad.totales.jug.primeros.df[estad.totales.jug.primeros.
df$equipo == "Atlético Madrid",]$recuperaciones)
shapiro.test(estad.totales.jug.primeros.df[estad.totales.jug.primeros.
df$equipo == "Sevilla",]$recuperaciones)

## Realizamos el ANOVA
anova.recuperaciones <- aov(recuperaciones ~ equipo,
estad.totales.jug.primeros.df)
summary(anova.recuperaciones)
```

correlation_regression_analysis.R

```
###### ANÁLISIS DE CORRELACIÓN Y REGRESIÓN ######

# Instalamos y cargamos los paquetes que hagan falta
packages = c("tidyverse", "GGally", "rsq", "relaimpo", "car", "vegan")
inst <- packages %in% installed.packages()
if(length(packages[!inst]) > 0) install.packages(packages[!inst])
lapply(packages, require, character.only=T)


# Cargamos los datos
estad.totales.jug <-
read.csv("data/estadisticas_totales_jugadores_liga_19_20.csv",
stringsAsFactors = F, sep = ";")


# Preparamos los datos

## Convertimos a factor las variables de equipo, país y posiciones
estad.totales.jug <- estad.totales.jug %>%
  mutate(pais = as.factor(pais),
         equipo = as.factor(equipo),
         posicion = as.factor(posicion),
         posicion.alternativa = as.factor(posicion.alternativa))

## Sustituimos los NAs por la media
estad.totales.jug <- estad.totales.jug %>%
  mutate(edad = as.double(edad),
         altura = as.double(altura),
         peso = as.double(peso),
         recuperaciones = as.double(recuperaciones),
         duelos.aereos.ganados = as.double(duelos.aereos.ganados),
         duelos.aereos.perdidos = as.double(duelos.aereos.perdidos),
         penales.concedidos = as.double(penales.concedidos)) %>%
  mutate(edad = if_else(is.na(edad), mean(edad, na.rm=TRUE), edad),
         altura = if_else(is.na(altura), mean(altura, na.rm=TRUE),
altura),
         peso = if_else(is.na(peso), mean(peso, na.rm=TRUE), peso),
```

```
         recuperaciones = if_else(is.na(recuperaciones),
round(mean(recuperaciones, na.rm=TRUE),0), recuperaciones),
         duelos.aereos.ganados = if_else(is.na(duelos.aereos.ganados),
round(mean(duelos.aereos.ganados, na.rm=TRUE)),
duelos.aereos.ganados),
         duelos.aereos.perdidos =
if_else(is.na(duelos.aereos.perdidos),
round(mean(duelos.aereos.perdidos, na.rm=TRUE)),
duelos.aereos.perdidos),
         penales.concedidos = if_else(is.na(penales.concedidos),
round(mean(penales.concedidos, na.rm=TRUE)), penales.concedidos))


## Vamos a separar los datos de porteros y el resto de demarcaciones
estad.totales.por <- estad.totales.jug %>%
  filter(posicion == "PO") %>%
  dplyr::select(-c(nombre,
            pais,
            equipo,
            posicion,
            posicion.alternativa,
            partidos.jugados,
            partidos.titular,
            goles.marcados,
            goles.propia.puerta,
            asistencias,
            pases.cruzados,
            intercepciones,
            tackleos.ganados,
            duelos.aereos.ganados,
            duelos.aereos.perdidos,
            faltas.cometidas,
            faltas.recibidas,
            tarjetas.amarillas,
            segunda.amarilla,
            tarjetas.rojas,
            penales.marcados,
            penales.lanzados,
            fueras.de.juego))

estad.totales.resto <- estad.totales.jug %>%
  filter(posicion != "PO") %>%
  dplyr::select(-c(nombre,
            pais,
            equipo,
            posicion.alternativa,
            partidos.jugados,
            partidos.titular,
            goles.encajados,
            tiros.puerta.recibidos,
            paradas,
            penales.recibidos,
            penales.parados))

## Vamos a dividir entre características defensivas y atacantes
estad.totales.resto.def <- estad.totales.resto %>%
  filter(posicion %in% c("DF", "CC")) %>%
  dplyr::select(-c(posicion,
            goles.marcados,
            goles.propia.puerta,
```

```
            asistencias,
            pases.cruzados,
            faltas.recibidas,
            penales.marcados,
            penales.lanzados,
            fueras.de.juego))

estad.totales.resto.ataq <- estad.totales.resto %>%
  filter(posicion %in% c("DL", "CC")) %>%
  dplyr::select(-c(posicion,
            goles.propia.puerta,
            recuperaciones,
            intercepciones,
            tackleos.ganados,
            faltas.cometidas,
            tarjetas.amarillas,
            segunda.amarilla,
            tarjetas.rojas,
            penales.concedidos))

## Estandarizamos las variables para que queden en el mismo rango
estad.totales.por.est <- estad.totales.por %>%
  decostand("normalize")

estad.totales.resto.def.est <- estad.totales.resto.def %>%
  decostand("normalize")

estad.totales.resto.ataq.est <- estad.totales.resto.ataq %>%
  decostand("normalize")


# Análisis de correlación

## Vemos los gráficos de correlación y dispersión con la función
ggpairs
(cor.por <- ggpairs(estad.totales.por.est, cardinality_threshold =
20))
ggsave("plots/correlation_por.png", cor.por, width = 30, height = 20,
units = "cm")
(cor.resto.def <- ggpairs(estad.totales.resto.def.est,
cardinality_threshold = 20))
ggsave("plots/correlation_def.png", cor.resto.def, width = 30, height
= 20, units = "cm")
(cor.resto.ataq <- ggpairs(estad.totales.resto.ataq.est,
cardinality_threshold = 20))
ggsave("plots/correlation_ataq.png", cor.resto.ataq, width = 30,
height = 20, units = "cm")


# Análisis de regresión para explicar los goles recibidos por los
porteros

## Modelo
por.lm <- lm(goles.encajados ~ ., estad.totales.por.est)
summary(por.lm)

## Tasa de error
sigma(por.lm) * 100 / mean(estad.totales.por.est$goles.encajados)
rsq.partial(por.lm)
```

```
## Coeficientes
summary(por.lm)$coefficients

## Importancia relativa de los predictores
(crlm.por <- calc.relimp(por.lm,
                         type = "lmg",
                         rela = T))
par(mfrow = c(1,1))
plot(crlm.por)

## Evaluamos los residuos
par(mfrow=c(2,2))
plot(por.lm)

durbinWatsonTest(por.lm, alternative = "positive")


# Análisis de regresión para explicar las tarjetas amarillas recibidas
por los DF y CC

## Modelo
def.lm <- lm(tarjetas.amarillas ~ ., estad.totales.resto.def.est)
summary(def.lm)

## Tasa de error
sigma(def.lm) * 100 /
mean(estad.totales.resto.def.est$tarjetas.amarillas)
rsq.partial(def.lm)

## Coeficientes
summary(def.lm)$coefficients

## Importancia relativa de los predictores
(crlm.def <- calc.relimp(def.lm,
                         type = "lmg",
                         rela = T))
par(mfrow = c(1,1))
plot(crlm.def)

## Evaluamos los residuos
par(mfrow=c(2,2))
plot(def.lm)

durbinWatsonTest(def.lm, alternative = "positive")


# Análisis de regresión para explicar los goles marcados por los CC y
DL

## Modelo
ataq.lm <- lm(goles.marcados ~ ., estad.totales.resto.ataq.est,
na.action = NULL)
summary(ataq.lm)

## Tasa de error
sigma(ataq.lm) * 100 /
mean(estad.totales.resto.ataq.est$goles.marcados)
rsq.partial(ataq.lm)
```

```
## Coeficientes
summary(ataq.lm)$coefficients

## Importancia relativa de los predictores
(crlm.ataq <- calc.relimp(ataq.lm,
                          type = "lmg",
                          rela = T))
par(mfrow = c(1,1))
plot(crlm.ataq)

## Evaluamos los residuos
par(mfrow=c(2,2))
plot(ataq.lm)

durbinWatsonTest(ataq.lm, alternative = "positive")
```

clustering_analysis.R

```
###### ANÁLISIS DE CLÚSTER ######

# Instalamos y cargamos los paquetes que hagan falta
packages = c("tidyverse", "vegan", "cluster")
inst <- packages %in% installed.packages()
if(length(packages[!inst]) > 0) install.packages(packages[!inst])
lapply(packages, require, character.only=T)


# Cargamos los datos
estad.totales.jug <-
read.csv("data/estadisticas_totales_jugadores_liga_19_20.csv",
stringsAsFactors = F, sep = ";")


# Preparamos los datos

## Convertimos a factor las variables de equipo, país y posiciones
estad.totales.jug <- estad.totales.jug %>%
  mutate(pais = as.factor(pais),
         equipo = as.factor(equipo),
         posicion = as.factor(posicion),
         posicion.alternativa = as.factor(posicion.alternativa))

## Sustituimos los NAs por la media
estad.totales.jug <- estad.totales.jug %>%
  mutate(edad = as.double(edad),
         altura = as.double(altura),
         peso = as.double(peso),
         recuperaciones = as.double(recuperaciones),
         duelos.aereos.ganados = as.double(duelos.aereos.ganados),
         duelos.aereos.perdidos = as.double(duelos.aereos.perdidos),
         penales.concedidos = as.double(penales.concedidos)) %>%
  mutate(edad = if_else(is.na(edad), mean(edad, na.rm=TRUE), edad),
         altura = if_else(is.na(altura), mean(altura, na.rm=TRUE),
altura),
         peso = if_else(is.na(peso), mean(peso, na.rm=TRUE), peso),
         recuperaciones = if_else(is.na(recuperaciones),
round(mean(recuperaciones, na.rm=TRUE),0), recuperaciones),
```

```
         duelos.aereos.ganados = if_else(is.na(duelos.aereos.ganados),
round(mean(duelos.aereos.ganados, na.rm=TRUE)),
duelos.aereos.ganados),
         duelos.aereos.perdidos =
if_else(is.na(duelos.aereos.perdidos),
round(mean(duelos.aereos.perdidos, na.rm=TRUE)),
duelos.aereos.perdidos),
         penales.concedidos = if_else(is.na(penales.concedidos),
round(mean(penales.concedidos, na.rm=TRUE)), penales.concedidos))

## Agrupamos por jugador para que no haya jugadores repetidos
estad.totales.jug <- estad.totales.jug %>%
  group_by(nombre, pais, edad, altura, peso) %>%
  summarise(equipo = last(equipo),
            posicion = last(posicion),
            posicion.alternativa = last(posicion.alternativa),
            across(partidos.jugados:fueras.de.juego, sum))

## Dividimos los datos por demarcación y nos quedamos con las
variables que nos interesen de cada demarcación
estad.totales.por <- estad.totales.jug %>%
  filter(posicion == "PO") %>%
  ungroup() %>%
  dplyr::select(-c(pais,
                   equipo,
                   posicion,
                   posicion.alternativa,
                   partidos.jugados,
                   partidos.titular,
                   goles.marcados,
                   goles.propia.puerta,
                   asistencias,
                   pases.cruzados,
                   intercepciones,
                   tackleos.ganados,
                   duelos.aereos.ganados,
                   duelos.aereos.perdidos,
                   faltas.cometidas,
                   faltas.recibidas,
                   segunda.amarilla,
                   penales.marcados,
                   penales.lanzados,
                   fueras.de.juego))

estad.totales.def <- estad.totales.jug %>%
  filter(posicion == "DF") %>%
  ungroup() %>%
  dplyr::select(-c(pais,
                   equipo,
                   posicion,
                   posicion.alternativa,
                   partidos.jugados,
                   partidos.titular,
                   goles.encajados,
                   tiros.puerta.recibidos,
                   paradas,
                   penales.recibidos,
                   penales.parados,
                   posicion,
                   pases.cruzados,
```

```
                    faltas.recibidas,
                    fueras.de.juego))

estad.totales.cen <- estad.totales.jug %>%
  filter(posicion == "CC") %>%
  ungroup() %>%
  dplyr::select(-c(pais,
                    equipo,
                    posicion,
                    posicion.alternativa,
                    partidos.jugados,
                    partidos.titular,
                    goles.encajados,
                    tiros.puerta.recibidos,
                    paradas,
                    penales.recibidos,
                    penales.parados,
                    goles.propia.puerta,
                    fueras.de.juego))

estad.totales.del <- estad.totales.jug %>%
  filter(posicion == "DL") %>%
  ungroup() %>%
  dplyr::select(-c(pais,
                    equipo,
                    posicion,
                    posicion.alternativa,
                    partidos.jugados,
                    partidos.titular,
                    goles.encajados,
                    tiros.puerta.recibidos,
                    paradas,
                    penales.recibidos,
                    penales.parados,
                    goles.propia.puerta,
                    penales.concedidos))

## Estandarizamos las variables para que queden en el mismo rango
estad.totales.por.est <- estad.totales.por %>%
  dplyr::select(-nombre) %>%
  decostand("normalize")

estad.totales.def.est <- estad.totales.def %>%
  dplyr::select(-nombre) %>%
  decostand("normalize")

estad.totales.cen.est <- estad.totales.cen %>%
  dplyr::select(-nombre) %>%
  decostand("normalize")

estad.totales.del.est <- estad.totales.del %>%
  dplyr::select(-nombre) %>%
  decostand("normalize")


# Análisis de clúster para porteros

## Vemos el número correcto de clústeres a retener
estad.totales.por.kmcascade <- cascadeKM(estad.totales.por.est,
inf.gr=2, sup.gr=10, iter=100, criterion="ssi")
```

```
summary(estad.totales.por.kmcascade)
estad.totales.por.kmcascade$results
plot(estad.totales.por.kmcascade, sortg=TRUE)

## Matriz de distancias
estad.totales.por.euc <- vegdist(estad.totales.por.est, "euc")

## K-Means para 10 grupos
set.seed(1)
estad.totales.por.kmeans <- kmeans(estad.totales.por.est, centers=10,
nstart=100)

## Analizamos su silueta
k <- 10
sil <- silhouette(estad.totales.por.kmeans$cluster,
estad.totales.por.euc)
rownames(sil) <- row.names(estad.totales.por.est)
plot(sil, main="Silhouette plot - k-means",
     cex.names=0.8, col=2:(k+1))

## Vemos el gráfico de clústeres
clusplot(estad.totales.por.est, estad.totales.por.kmeans$cluster,
color=TRUE, shade=TRUE, labels=2, lines=0)

## Vemos en que clúster está cada portero
estad.totales.por.clu <- cbind(estad.totales.por$nombre,
as.data.frame(estad.totales.por.kmeans$cluster))
names(estad.totales.por.clu) <- c("nombre", "cluster")


# Análisis de clúster para defensas

## Vemos el número correcto de clústeres a retener
estad.totales.def.kmcascade <- cascadeKM(estad.totales.def.est,
inf.gr=2, sup.gr=10, iter=100, criterion="ssi")
summary(estad.totales.def.kmcascade)
estad.totales.def.kmcascade$results
plot(estad.totales.def.kmcascade, sortg=TRUE)

## Matriz de distancias
estad.totales.def.euc <- vegdist(estad.totales.def.est, "euc")

## K-Means para 10 grupos
set.seed(1)
estad.totales.def.kmeans <- kmeans(estad.totales.def.est, centers=10,
nstart=100)

## Analizamos su silueta
k <- 10
sil <- silhouette(estad.totales.def.kmeans$cluster,
estad.totales.def.euc)
rownames(sil) <- row.names(estad.totales.def.est)
plot(sil, main="Silhouette plot - k-means",
     cex.names=0.8, col=2:(k+1))

## Vemos el gráfico de clústeres
clusplot(estad.totales.def.est, estad.totales.def.kmeans$cluster,
color=TRUE, shade=TRUE, labels=2, lines=0)
```

```
## Vemos en que clúster está cada defensa
estad.totales.def.clu <- cbind(estad.totales.def$nombre,
as.data.frame(estad.totales.def.kmeans$cluster))
names(estad.totales.def.clu) <- c("nombre", "cluster")


# Análisis de clúster para centrocampistas

## Vemos el número correcto de clústeres a retener
estad.totales.cen.kmcascade <- cascadeKM(estad.totales.cen.est,
inf.gr=2, sup.gr=10, iter=100, criterion="ssi")
summary(estad.totales.cen.kmcascade)
estad.totales.cen.kmcascade$results
plot(estad.totales.cen.kmcascade, sortg=TRUE)

## Matriz de distancias
estad.totales.cen.euc <- vegdist(estad.totales.cen.est, "euc")

## K-Means para 8 grupos
set.seed(1)
estad.totales.cen.kmeans <- kmeans(estad.totales.cen.est, centers=8,
nstart=100)

## Analizamos su silueta
k <- 8
sil <- silhouette(estad.totales.cen.kmeans$cluster,
estad.totales.cen.euc)
rownames(sil) <- row.names(estad.totales.cen.est)
plot(sil, main="Silhouette plot - k-means",
     cex.names=0.8, col=2:(k+1))

## Vemos el gráfico de clústeres
clusplot(estad.totales.cen.est, estad.totales.cen.kmeans$cluster,
color=TRUE, shade=TRUE, labels=2, lines=0)

## Vemos en que clúster está cada centrocampista
estad.totales.cen.clu <- cbind(estad.totales.cen$nombre,
as.data.frame(estad.totales.cen.kmeans$cluster))
names(estad.totales.cen.clu) <- c("nombre", "cluster")


# Análisis de clúster para delanteros

## Vemos el número correcto de clústeres a retener
estad.totales.del.kmcascade <- cascadeKM(estad.totales.del.est,
inf.gr=2, sup.gr=10, iter=100, criterion="ssi")
summary(estad.totales.del.kmcascade)
estad.totales.del.kmcascade$results
plot(estad.totales.del.kmcascade, sortg=TRUE)

## Matriz de distancias
estad.totales.del.euc <- vegdist(estad.totales.del.est, "euc")

## K-Means para 9 grupos
set.seed(1)
estad.totales.del.kmeans <- kmeans(estad.totales.del.est, centers=9,
nstart=100)

## Analizamos su silueta
k <- 9
```

```
sil <- silhouette(estad.totales.del.kmeans$cluster,
estad.totales.del.euc)
rownames(sil) <- row.names(estad.totales.del.est)
plot(sil, main="Silhouette plot - k-means",
     cex.names=0.8, col=2:(k+1))

## Vemos el gráfico de clústeres
clusplot(estad.totales.del.est, estad.totales.del.kmeans$cluster,
color=TRUE, shade=TRUE, labels=2, lines=0)

## Vemos en que clúster está cada delantero
estad.totales.del.clu <- cbind(estad.totales.del$nombre,
as.data.frame(estad.totales.del.kmeans$cluster))
names(estad.totales.del.clu) <- c("nombre", "cluster")
```