PRA 2. Limpieza y análisis de datos

Contents

Descripción del dataset.

El dataset elegido corresponde al generado en la PRAC1. El dataset contiene los resultados obtenidos por distintos equipos profesionales de fútbol en diversas competiciones a lo largo de la última década.

Nos serviremos de estos datos para anlizar los diferentes clubs de fútbol a nivel global así como para buscar patrones entre ellos.

```
df <- read.csv(file = 'football_competitions.csv')
head(df)
```

```
##                       competition       pl_team pl_pi pl_w pl_d pl_l pl_f
## 1 African Nations - Group A 2008       Ghana *     3    3    0    0    5
## 2 African Nations - Group A 2008      Guinea *     3    1    1    1    5
## 3 African Nations - Group A 2008     Morocco *     3    1    0    2    7
## 4 African Nations - Group A 2008     Namibia *     3    0    1    2    2
## 5 African Nations - Group B 2008 Ivory Coast *     3    3    0    0    8
## 6 African Nations - Group B 2008     Nigeria *     3    1    1    1    2
##   pl_a pl_gd pl_pts
## 1    1     4      9
## 2    5     0      4
## 3    6     1      3
## 4    7    -5      1
## 5    1     7      9
## 6    1     1      4
```

```
str(df)
```

```
## 'data.frame':    6623 obs. of  10 variables:
##  $ competition: Factor w/ 825 levels "Africa Cup of Nations - Group A 2010",..: 19 19 19 19 20 20 20
##  $ pl_team    : Factor w/ 993 levels "1. FC Union Berlin",..: 398 418 616 624 493 643 579 103 298 17
```

```
##  $ pl_pi      : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ pl_w       : int  3 1 1 0 3 1 1 0 2 2 ...
##  $ pl_d       : int  0 1 0 1 0 1 1 0 1 0 ...
##  $ pl_l       : int  0 1 2 2 0 1 1 3 0 1 ...
##  $ pl_f       : int  5 5 7 2 8 2 1 1 8 10 ...
##  $ pl_a       : int  1 5 6 7 1 1 3 7 3 5 ...
##  $ pl_gd      : int  4 0 1 -5 7 1 -2 -6 5 5 ...
##  $ pl_pts     : int  9 4 3 1 9 4 4 0 7 6 ...
```

El fichero contiene una tabla con diez columnas, siendo las dos primeras de tipo cadena y el resto de tipo entero.

Los campos son:

- competition: Nombre de la competición, año y grupo dentro de los cuales el equipo indicado obtuvo los resultados indicados.
- pl_team: Nombre del equipo.
- pl_pi: Número de partidos jugados.
- pl_w: Número de partidos ganados.
- pl_d: Número de partidos empatados.
- pl_l: Número de partidos perdidos.
- pl_f: Número de goles a favor.
- pl_a: Número de goles en contra.
- pl_gd: Diferencia de goles.
- pl_pts: Puntos obtenidos por el equipo.

Integración y selección de los datos de interés a analizar

Algunos de los campos del dataset original son redundantes. Como podemos ver, esto sucede con el campo relativo a la diferencia de goles, que se puede calcular a partir de los goles a favor y en contra.

```
max(df$pl_a - df$pl_f + df$pl_gd)
```

```
## [1] 0
```

Además, el total de partidos jugados se puede calcular mediante la suma de los partidos ganados, perdidos y empatados.

```
max(df$pl_pi - df$pl_w - df$pl_d -df$pl_l)
```

```
## [1] 0
```

Hemos visto mediante una suma de comprobación que estas columnas son consistentes. No obstante, debemos excluir los campos redundantes para evitar la multicolinealidad en el dataset.

```
df_clean <- df[,c('pl_w','pl_d','pl_l','pl_f','pl_a','pl_pts')]
```

Como comprobación adicional en relación al número de partidos jugados, observamos que no necesariamente todos los equipos juegan el mismo número de partidos dentro de una misma competición.

```
comp_variation <- df %>% group_by(df$competition) %>% summarise(variacion_partidos = sd(pl_pi))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
comp_variation <- comp_variation[comp_variation$variacion_partidos > 0,]
comp_variation
```

```
## # A tibble: 65 x 2
```

```
##    `df$competition`                                  variacion_partidos
##    <fct>                                                          <dbl>
##  1 American MLS League - Eastern Conference 2011/12               0.333
##  2 American MLS League - Eastern Conference 2012/13               0.316
##  3 American MLS League - Eastern Conference 2014/15               1.90
##  4 American MLS League - Eastern Conference 2020/21               0.469
##  5 American MLS League - Western Conference 2011/12               0.333
##  6 American MLS League - Western Conference 2014/15               0.943
##  7 American MLS League - Western Conference 2020/21               1.59
##  8 Dutch Eredivisie 2019/20                                       0.428
##  9 Dutch Eredivisie 2020/21                                       0.323
## 10 FA Women's Championship 2019/20                                1.21
## # ... with 55 more rows
```

Dado que el campo competición tiene una naturaleza jerárquica, vamos a descomponerlo en tres campos: - comp: nombre de la competición - group: nombre del grupo dentro de dicha competición - begin: año de inicio de la competición - end: año de finalización de la competición

```
# Separar competicion, grupo y año
intermediate_comp <- stri_match(df$competition, regex="(.*?)[ ]*([^ ]*\\d)$")
intermediate_comp_name <- stri_match(intermediate_comp[,2], regex="(.*?)[ ]+-[ ]+(.*)")
df_clean[is.na(intermediate_comp_name[,1]), 'comp'] <-
  intermediate_comp[is.na(intermediate_comp_name[,1]),2]
df_clean[is.na(intermediate_comp_name[,1]), 'group'] <- NA
df_clean[!is.na(intermediate_comp_name[,1]), 'comp'] <-
  intermediate_comp_name[!is.na(intermediate_comp_name[,1]),2]
df_clean[!is.na(intermediate_comp_name[,1]), 'group'] <-
  intermediate_comp_name[!is.na(intermediate_comp_name[,1]),3]
intermediate_comp_year <-
  stri_match(intermediate_comp[,3], regex="(.*)/(.*)")
# Añadir año de comienzo y año de fin.
df_clean[is.na(intermediate_comp_year[,1]),'begin'] <-
  as.integer(intermediate_comp[is.na(intermediate_comp_year[,1]),3])
df_clean[is.na(intermediate_comp_year[,1]),'end'] <-
  as.integer(intermediate_comp[is.na(intermediate_comp_year[,1]),3])
df_clean[!is.na(intermediate_comp_year[,1]),'begin'] <-
  as.integer(intermediate_comp_year[!is.na(intermediate_comp_year[,1]),2])
df_clean[!is.na(intermediate_comp_year[,1]),'end'] <-
  as.integer(paste("20", intermediate_comp_year[!is.na(intermediate_comp_year[,1]),3], sep=""))
```

Eliminamos el asterisco que aparece al final de algunos nombres de equipos:

```
df_clean$pl_team <- stri_match(df$pl_team, regex="([^\\*]*)( \\*)?")[,2]
df_clean$pl_team <- stri_match(df_clean$pl_team, regex="(.*[^ ])( )*")[,2]
```

Observamos que algunas de las competiciones son femeninas o sub-21, con lo que añadimos dos campos más para identificar dichas competiciones: - female: booleano que indica si la competición es femenina - sub21: booleano que indica si la competición es sub-21

```
df_clean$female <- !is.na(stri_match(df_clean$comp, regex="Women"))
df_clean$sub21 <- !is.na(stri_match(df_clean$comp, regex="U(nder)?( )*-( )*?21")[,1])
```

Basándonos en que el dataset incluye diversas ligas junto con competiciones entre distintas selecciones nacionales, vamos a identificar la manera en la cual se relacionan los equipos entre sí y dentro de cada competición. Para ello vamos a considerar cada registro del dataset como una arista de un grafo bipartito que incluye por un lado los equipos y por otro lado las competiciones.

```
b <- df_clean[,c("comp","pl_team")] %>% distinct()
n <- length(b[,1])
g <- c()
for (i in 1:n ){
g <- c(g, paste( "C:", b[i,"comp"]), paste("T:", b[i,"pl_team"]))
}
graph <- make_graph(g, directed=FALSE)
```

Sobre este grafo, vamos a lanzar un algoritmo de detección de comunidades que nos muestre a que grupos pertenecen los distintos equipos y competiciones. Añadimos al dataset los siguientes campos: - comp_comm: entero que identifica la comunidad a la cual pertenece la competición - team_comm: entero que identifica la comunidad a la cual pertenece el equipo

```
fc <- fastgreedy.community(graph)
df_clean$comp_comm <- membership(fc)[paste("C:",df_clean$comp)]
df_clean$team_comm <- membership(fc)[paste("T:",df_clean$pl_team)]
```

Limpieza de datos

Elementos vacíos y cero.

Primero vamos a ver si hay valores vacíos en el dataset.

```
sapply(df_clean, function(x) sum(is.na(x)))
```

```
##      pl_w      pl_d      pl_l      pl_f      pl_a    pl_pts      comp
##         0         0         0         0         0         0         0
##     group     begin       end   pl_team    female     sub21 comp_comm
##      3747         0         0         0         0         0         0
## team_comm
##         0
```

Unicamente los hay en el campo group, que hemos añadido. Se corresponden a ligas en las cuales no hay una subdivisión por grupos.

Buscamos valores 0.

```
sum(df_clean[,c('pl_w','pl_d','pl_l',
            'pl_f','pl_a','pl_pts',
            'begin','end')]==0)
```

```
## [1] 3610
```

Hay muchos datos que son igual a cero. Debido a la naturaleza del dataframe no es necesario quitarlos, porque puede ser que un equipo tenga 0 puntos, victorias, derrotas, etc. Por el contrario, debemos asegurarnos de que excluimos las entradas del dataset en las cuales no figure ningun partido jugado, para evitar divisiones entre cero a la hora de normalizar los goles y puntos.

```
sum((df_clean[,"pl_w"] + df_clean[,"pl_d"] + df_clean[,"pl_l"]) == 0)
```

```
## [1] 250
```

```
df_clean <- df_clean[(df_clean[,"pl_w"] + df_clean[,"pl_d"] + df_clean[,"pl_l"]) > 0,]
```

```
sum((df_clean[,"pl_w"] + df_clean[,"pl_d"] + df_clean[,"pl_l"]) == 0)
```

```
## [1] 0
```

Valores extremos.

Para la detección de outliers o valores extremos utilizamos la función boxplots.stats(), en concreto el atributo "out" que nos devuelve los valores que distan mucho del rango intercuartílico que se dibujan en los diagramas de caja.

```
boxplot.stats(df_clean$pl_w)$out
```

```
## integer(0)
```

```
boxplot.stats(df_clean$pl_d)$out
```

```
## integer(0)
```

```
boxplot.stats(df_clean$pl_l)$out
```

```
## [1] 35 38
```

```
boxplot.stats(df_clean$pl_f)$out
```

```
## integer(0)
```

```
boxplot.stats(df_clean$pl_a)$out
```

```
## [1] 143 140
```

A penas hay valores outliers y tras comprobarlos, vemos que efectivamente hay equipos que han ganado más de 30 partidos en una competición y lo mismo ocurre con las derrotas y los goles en contra.

Finalmente guardamos la versión limpia y estable del dataset, en la cual nos basaremos para realizar el análisis.

```
#write.csv(df_clean, 'clean.csv')
df_final <- read.csv(file = 'clean.csv')
```

Análisis de los datos.

Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar).

Queremos estudiar diferencias cualitativas entre las comunidades identificadas durante la fase de integración de datos. El algoritmo ha identificado 15 comunidades distintas.

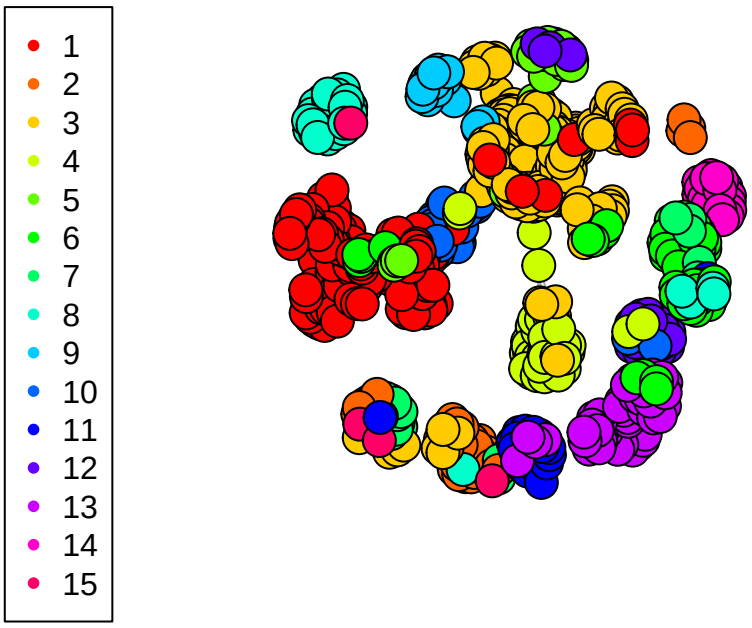
```
max(df_final$comp_comm)
```

```
## [1] 15
```

El grafo formado por los equipos y las competiciones, dentro de sus respectivas comunidades, queda entonces representado de la siguiente manera:

```
colors <- rainbow(max(membership(fc)))

bb <- df_clean[,c("pl_team", "team_comm")] %>% distinct()
bbb <- df_clean[,c("comp", "comp_comm")] %>% distinct()
comm <- list()
n <- length(bb[,1])
for (i in 1:n ){
comm[paste("T:",bb[i,"pl_team"])] <- bb[i,"team_comm"]
}
```

```
n <- length(bbb[,1])
for (i in 1:n ){
comm[paste("C:",bbb[i,"comp"])] <- bbb[i,"comp_comm"]
}
col <- c()
for (name in V(graph)$name){
  col <- c(col, colors[comm[[name]]])

}
plot(graph, vertex.label=NA, vertex.size=15, arrow.width=1,vertex.color=col);legend(
  'topleft',legend=1:max(membership(fc)), pch=20, col=colors)
```



Vemos diferentes componentes conexas, como se espera, debido a que el dataset incluye diferentes categorías y ligas.

Veamos a continuación cuales son las competiciones incluidas en cada una de las comunidades identificadas:

```
unique(as.character(df_final[df_final$comp_comm == 1, "comp"]))
```

```
##  [1] "Leasing.com Trophy"             "Papa John's Trophy"
##  [3] "Blue Square North"              "Skrill North"
##  [5] "Conference North"               "Vanarama National League North"
##  [7] "National League North"          "Blue Square South"
##  [9] "Skrill South"                   "Conference South"
## [11] "Vanarama National League South" "National League South"
## [13] "Blue Square Premier"            "Skrill Premier"
## [15] "Vanarama Conference"            "Vanarama National League"
## [17] "National League"                "Coca-Cola Football League One"
## [19] "npower League 1"                "Sky Bet League 1"
## [21] "Sky Bet League One"             "Coca-Cola Football League Two"
## [23] "npower League 2"                "Sky Bet League 2"
## [25] "Sky Bet League Two"
```

```
unique(as.character(df_final[df_final$comp_comm == 2, "comp"]))
```

```
## [1] "Copa America"
```

```
## [2] "FIFA World Cup CONCACAF Qualifying"
## [3] "FIFA World Cup South American"
## [4] "FIFA World Cup Asian Qualifying"
```

```
unique(as.character(df_final[df_final$comp_comm == 3, "comp"]))
```

```
## [1] "German Bundesliga"     "UEFA Champions League" "Europa League"
## [4] "Spanish Primera Liga"  "Spanish Primera BBVA"  "Spanish La Liga"
## [7] "Italy Serie A"         "Italian Serie A"
```

```
unique(as.character(df_final[df_final$comp_comm == 4, "comp"]))
```

```
##  [1] "The Irn-Bru Scottish Football League Championship First Division"
##  [2] "The Irn-Bru Scottish First Division"
##  [3] "Scottish Championship"
##  [4] "Scottish League Cup"
##  [5] "The Irn-Bru Scottish Football League Championship Second Division"
##  [6] "The Irn-Bru Scottish Second Division"
##  [7] "Scottish League One"
##  [8] "Scottish League 1"
##  [9] "The Irn-Bru Scottish Football League Championship Third Division"
## [10] "The Irn-Bru Scottish Third Division"
## [11] "Scottish League Two"
## [12] "Scottish League 2"
## [13] "Clydesdale Bank Premier League"
## [14] "Scottish Premiership"
```

```
unique(as.character(df_final[df_final$comp_comm == 5, "comp"]))
```

```
## [1] "French Ligue 1"
```

```
unique(as.character(df_final[df_final$comp_comm == 6, "comp"]))
```

```
## [1] "Women's World Cup Qualifying" "SheBelieves Cup"
## [3] "FIFA Women's World Cup"
```

```
unique(as.character(df_final[df_final$comp_comm == 7, "comp"]))
```

```
## [1] "European Championships"
## [2] "World Cup Qualifying"
## [3] "FIFA World Cup European Qualifying"
## [4] "UEFA Nations League"
```

```
unique(as.character(df_final[df_final$comp_comm == 8, "comp"]))
```

```
## [1] "African Nations"                   "Africa Cup of Nations"
## [3] "FIFA World Cup African Qualifying"
```

```
unique(as.character(df_final[df_final$comp_comm == 9, "comp"]))
```

```
## [1] "Dutch Eredivisie"
```

```
unique(as.character(df_final[df_final$comp_comm == 10, "comp"]))
```

```
## [1] "Coca-Cola Football League Championship"
## [2] "npower Championship"
## [3] "Sky Bet Championship"
## [4] "Barclays Premier League"
## [5] "Premier League"
```

```
unique(as.character(df_final[df_final$comp_comm == 11, "comp"]))
```

```
## [1] "European U-21 Champ"            "European Under-21 Championship"
```

```
unique(as.character(df_final[df_final$comp_comm == 12, "comp"]))
```

```
## [1] "American MLS League"
```

```
unique(as.character(df_final[df_final$comp_comm == 13, "comp"]))
```

```
## [1] "Women's Premier League Northern Division"
## [2] "FA Women's National League"
## [3] "Women's Premier League Southern Division"
## [4] "FA Women's Championship"
## [5] "FA Women's Super League"
```

```
unique(as.character(df_final[df_final$comp_comm == 14, "comp"]))
```

```
## [1] "Chinese Super League"
```

```
unique(as.character(df_final[df_final$comp_comm == 15, "comp"]))
```

```
## [1] "FIFA World Cup Oceania Qualifying"
```

Teniendo en cuenta que cada componente conexa puede albergar más de una comunidad, queremos ser capaces de localizar vínculos entre componentes conexas adyacentes. Recordando que cada entrada en el dataset corresponde a un equipo jugando dentro de una competición, y que las comunidades de ambos pueden no coincidir, podemos observar vínculos entre comunidades mediante la siguiente matriz de confusión:

```
table(df_clean$team_comm, df_clean$comp_comm)
```

```
##
##         1    2    3    4    5    6    7    8    9   10   11   12   13   14
##   1  1648    0    1    0    0    0    0    0    0   65    0    0    0    0
##   2     0  109    0    0    0    0    0    0    0    0    0    0    0    0
##   3     0    0 1546    0   82    0    0    0   59    0    0    0    0    0
##   4     0    0   16  704    0    0    0    0    0    0    0    0    0    0
##   5     0    0   11    0  158    0    0    0    0    0    0    0    0    0
##   6     0    0    0    0    0  140    0    0    0    0    0    0    0    0
##   7     0    0    0    0    0    0  391    0    0    0    0    0    0    0
##   8     0    0    0    0    0    0    0   96    0    0    0    0    0    0
##   9     0    0    4    0    0    0    0    0  157    0    0    0    0    0
##   10   53    0   78    0    0    0    0    0    0  463    0    0    0    0
##   11    0    0    0    0    0    0    0    0    0    0   64    0    0    0
##   12    0    0    0    0    0    0    0    0    0    0    0  218    0    0
##   13    0    0    0    0    0    0    0    0    0    0    0    0  232    0
##   14    0    0    0    0    0    0    0    0    0    0    0    0    0   72
##   15    0    0    0    0    0    0    0    0    0    0    0    0    0    0
##
##        15
##   1     0
##   2     0
##   3     0
##   4     0
##   5     0
##   6     0
##   7     0
##   8     0
```

```
##   9     0
##   10    0
##   11    0
##   12    0
##   13    0
##   14    0
##   15    6
```

Podemos, por ejemplo, observar vínculos entre las comunidades 1 y 10, a las cuales corresponden competiciones inglesas.

Dado que, a la vista del grafo y la matriz, las competiciones inglesas son numerosas y parecen estar vinculadas mediante pocos enlaces con el resto de competiciones europeas, vamos a tratar de comparar los promedios de goles y puntuaciones entre estos dos grupos de competiciones.

De esta manera consideramos un grupo A, formado por las comunidades 1 y 10, frente a un grupo B, formado por las comunidades 3, 4, 5, 9.

```
df_A <- df_final[df_final$comp_comm %in% c(1,10), c("pl_f","pl_a","pl_w","pl_pts")]
df_A$pl_f <- (df_final$pl_f /
           (df_final$pl_w + df_final$pl_d + df_final$pl_l)
         )[df_final$comp_comm %in% c(1,10)]
df_A$pl_a <- (df_final$pl_a /
           (df_final$pl_w + df_final$pl_d + df_final$pl_l)
         )[df_final$comp_comm %in% c(1,10)]
df_A$pl_w <- (df_final$pl_w /
           (df_final$pl_w + df_final$pl_d + df_final$pl_l)
         )[df_final$comp_comm %in% c(1,10)]
df_A$pl_pts <- (df_final$pl_pts /
           (df_final$pl_w + df_final$pl_d + df_final$pl_l)
         )[df_final$comp_comm %in% c(1,10)]
df_A$group <- "A"
df_B <- df_final[df_final$comp_comm %in% c(3,4,5,9), c("pl_f","pl_a","pl_w","pl_pts")]
df_B$pl_f <- (df_final$pl_f /
           (df_final$pl_w + df_final$pl_d + df_final$pl_l)
         )[df_final$comp_comm %in% c(3,4,5,9)]
df_B$pl_a <- (df_final$pl_a /
           (df_final$pl_w + df_final$pl_d + df_final$pl_l)
         )[df_final$comp_comm %in% c(3,4,5,9)]
df_B$pl_w <- (df_final$pl_w /
           (df_final$pl_w + df_final$pl_d + df_final$pl_l)
         )[df_final$comp_comm %in% c(3,4,5,9)]
df_B$pl_pts <- (df_final$pl_pts /
           (df_final$pl_w + df_final$pl_d + df_final$pl_l)
         )[df_final$comp_comm %in% c(3,4,5,9)]
df_B$group <- "B"
df_A_B <- rbind(df_A, df_B)
```

Consideramos tambien grupos Masculino/Femenino:

```
df_masculino <- df_final[!df_final$female, c("pl_f","pl_a","pl_w","pl_pts")]
df_masculino$pl_f <- (df_final$pl_f /
           (df_final$pl_w + df_final$pl_d + df_final$pl_l)
         )[!df_final$female]
df_masculino$pl_a <- (df_final$pl_a /
           (df_final$pl_w + df_final$pl_d + df_final$pl_l)
         )[!df_final$female]
```

```
df_masculino$pl_w <- (df_final$pl_w /
           (df_final$pl_w + df_final$pl_d + df_final$pl_l)
         )[!df_final$female]
df_masculino$pl_pts <- (df_final$pl_pts /
           (df_final$pl_w + df_final$pl_d + df_final$pl_l)
         )[!df_final$female]
df_masculino$group <- "masculino"
df_femenino <- df_final[df_final$female, c("pl_f","pl_a","pl_w","pl_pts")]
df_femenino$pl_f <- (df_final$pl_f /
           (df_final$pl_w + df_final$pl_d + df_final$pl_l)
         )[df_final$female]
df_femenino$pl_a <- (df_final$pl_a /
           (df_final$pl_w + df_final$pl_d + df_final$pl_l)
         )[df_final$female]
df_femenino$pl_w <- (df_final$pl_w /
           (df_final$pl_w + df_final$pl_d + df_final$pl_l)
         )[df_final$female]
df_femenino$pl_pts <- (df_final$pl_pts /
           (df_final$pl_w + df_final$pl_d + df_final$pl_l)
         )[df_final$female]
df_femenino$group <- "femenino"
df_masculino_femenino <- rbind(df_masculino, df_femenino)
```

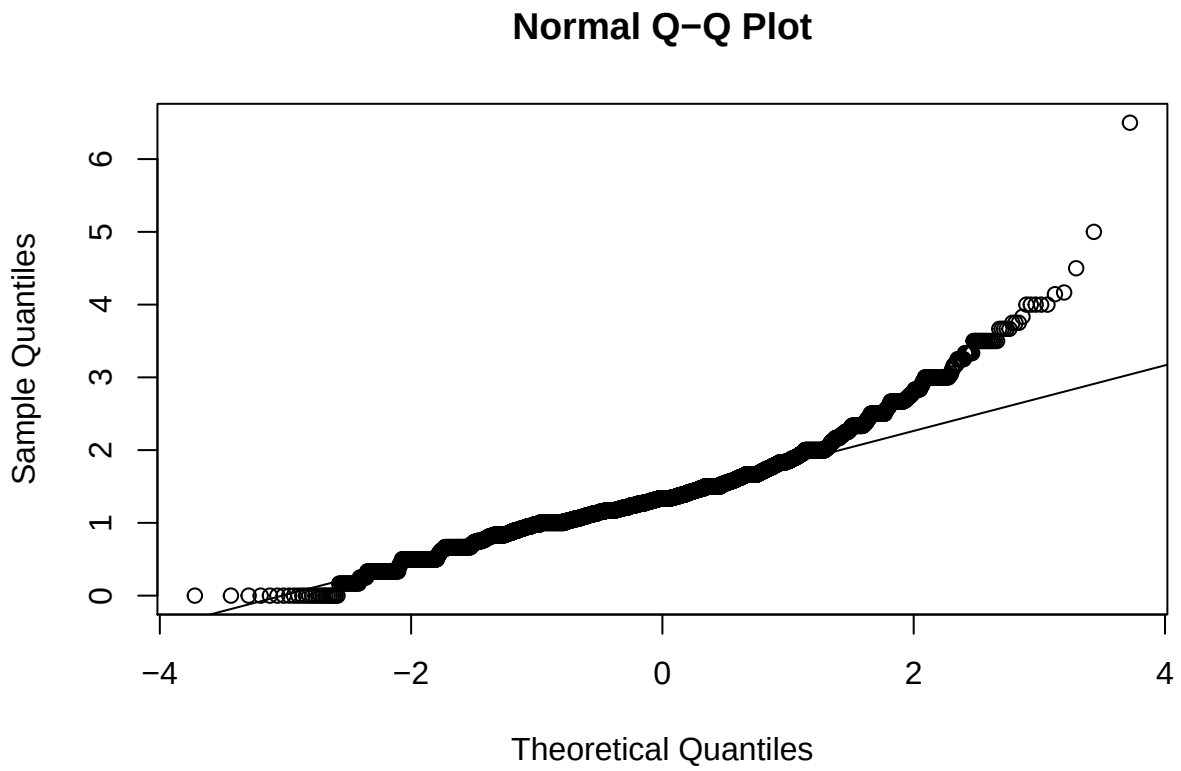
Por último, tenemos en cuenta tambien los grupos Absoluta/Sub21:

```
df_absoluta <- df_final[!df_final$sub21, c("pl_f","pl_a","pl_w","pl_pts")]
df_absoluta$pl_f <- (df_final$pl_f /
           (df_final$pl_w + df_final$pl_d + df_final$pl_l)
         )[!df_final$sub21]
df_absoluta$pl_a <- (df_final$pl_a /
           (df_final$pl_w + df_final$pl_d + df_final$pl_l)
         )[!df_final$sub21]
df_absoluta$pl_w <- (df_final$pl_w /
           (df_final$pl_w + df_final$pl_d + df_final$pl_l)
         )[!df_final$sub21]
df_absoluta$pl_pts <- (df_final$pl_pts /
           (df_final$pl_w + df_final$pl_d + df_final$pl_l)
         )[!df_final$sub21]
df_absoluta$group <- "absoluta"
df_sub21 <- df_final[df_final$sub21, c("pl_f","pl_a","pl_w","pl_pts")]
df_sub21$pl_f <- (df_final$pl_f /
           (df_final$pl_w + df_final$pl_d + df_final$pl_l)
         )[df_final$sub21]
df_sub21$pl_a <- (df_final$pl_a /
           (df_final$pl_w + df_final$pl_d + df_final$pl_l)
         )[df_final$sub21]
df_sub21$pl_w <- (df_final$pl_w /
           (df_final$pl_w + df_final$pl_d + df_final$pl_l)
         )[df_final$sub21]
df_sub21$pl_pts <- (df_final$pl_pts /
           (df_final$pl_w + df_final$pl_d + df_final$pl_l)
         )[df_final$sub21]
df_sub21$group <- "sub21"
df_absoluta_sub21 <- rbind(df_absoluta, df_sub21)
```

Comprobación de la normalidad y homogeneidad de la varianza.

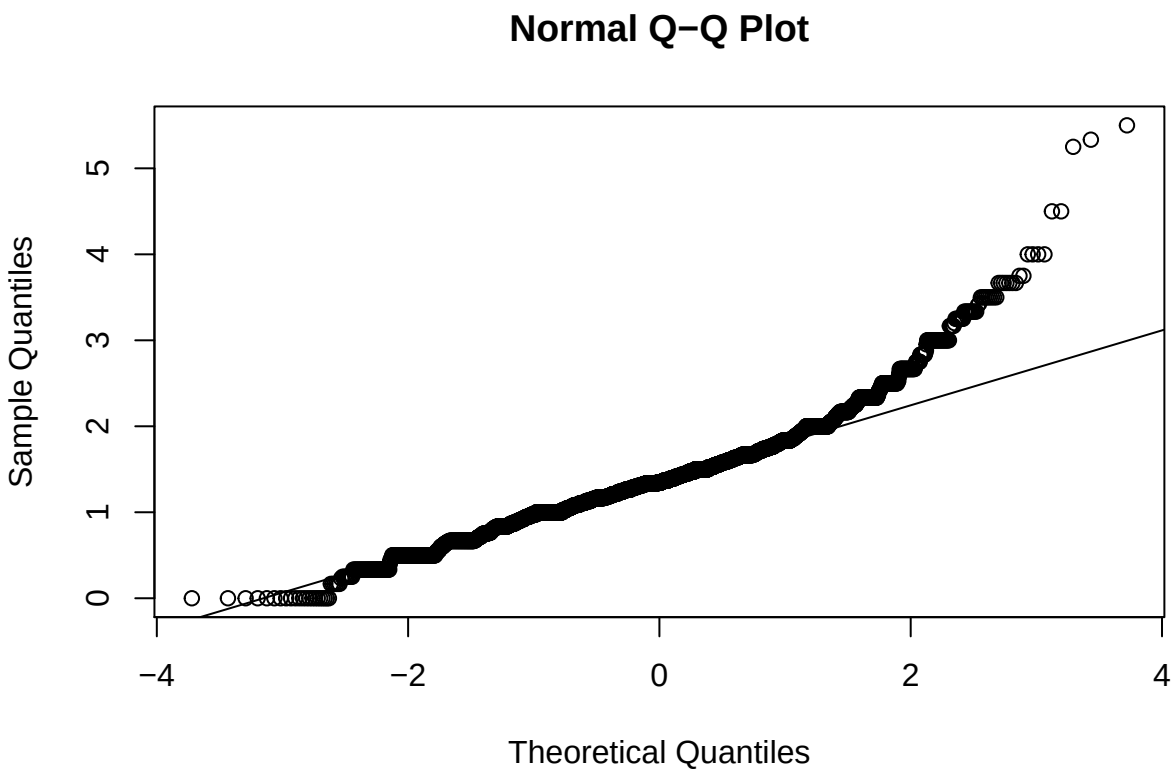
En primer lugar comprobamos la normalidad de la población mediante un gráfico Q-Q.

```
qqnorm(df_A_B$pl_f); qqline(df_A_B$pl_f)
```

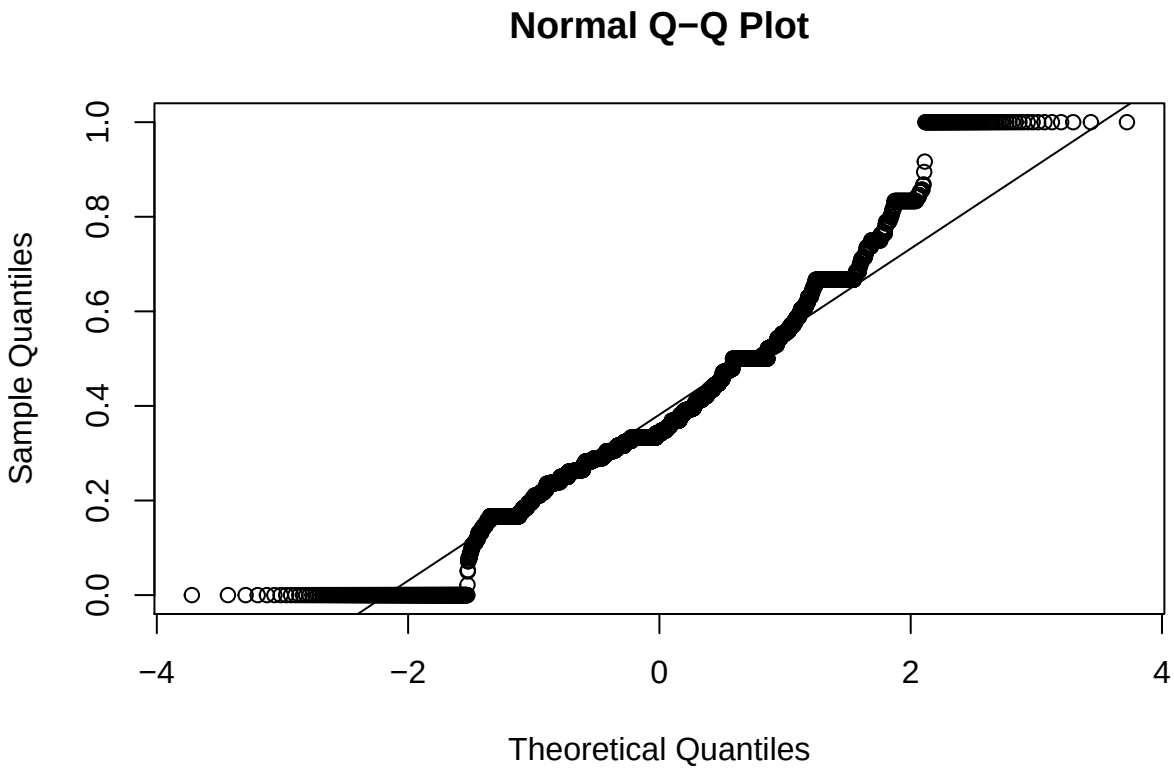


Vemos que el gráfico obtenido se aleja bastante de la recta que caracterizaría una distribución normal. Debemos, por lo tanto, aplicar métodos de inferencia no paramétrica para comparar promedios de goles a favor. Comprobemos el resto de métricas:

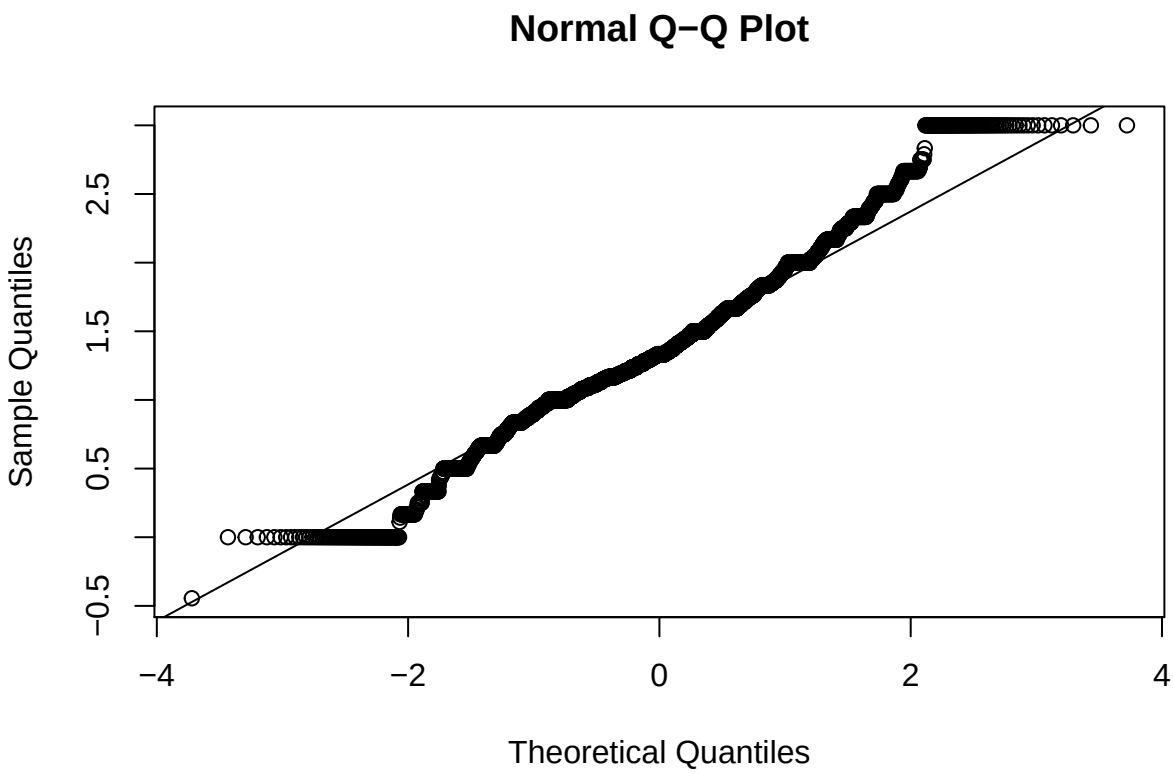
```
qqnorm(df_A_B$pl_a); qqline(df_A_B$pl_a)
```




```
qqnorm(df_A_B$pl_w); qqline(df_A_B$pl_w)
```




```
qqnorm(df_A_B$pl_pts); qqline(df_A_B$pl_pts)
```



Comprobamos que en ninguna de ellas aplica la hípotesis de normalidad.

Comprobamos la homocedasticidad mediante el test de Fligner-Killeen para las metricas de los grupos A y B:

```
fligner.test(pl_f ~ group, data = df_A_B)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  pl_f by group
## Fligner-Killeen:med chi-squared = 140.18, df = 1, p-value <
## 2.2e-16
```

```
fligner.test(pl_a ~ group, data = df_A_B)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  pl_a by group
## Fligner-Killeen:med chi-squared = 109.43, df = 1, p-value <
## 2.2e-16
```

```
fligner.test(pl_w ~ group, data = df_A_B)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  pl_w by group
## Fligner-Killeen:med chi-squared = 113.62, df = 1, p-value <
## 2.2e-16
```

```
fligner.test(pl_pts ~ group, data = df_A_B)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  pl_pts by group
## Fligner-Killeen:med chi-squared = 101.89, df = 1, p-value <
## 2.2e-16
```

Dado que todos los p-valores son ínfimos, estamos ante una situacion de heterocedasticidad.

Comprobemos el mismo test para los grupos Masculino/Femenino y Absoluta/Sub21:

```
fligner.test(pl_f ~ group, data = df_masculino_femenino)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  pl_f by group
## Fligner-Killeen:med chi-squared = 272.97, df = 1, p-value <
## 2.2e-16
```

```
fligner.test(pl_a ~ group, data = df_masculino_femenino)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  pl_a by group
## Fligner-Killeen:med chi-squared = 234.63, df = 1, p-value <
## 2.2e-16
```

```
fligner.test(pl_w ~ group, data = df_masculino_femenino)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  pl_w by group
## Fligner-Killeen:med chi-squared = 126.2, df = 1, p-value < 2.2e-16
```

```
fligner.test(pl_pts ~ group, data = df_masculino_femenino)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  pl_pts by group
## Fligner-Killeen:med chi-squared = 126.04, df = 1, p-value <
## 2.2e-16
```

```
fligner.test(pl_f ~ group, data = df_absoluta_sub21)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  pl_f by group
## Fligner-Killeen:med chi-squared = 10.8, df = 1, p-value = 0.001015
```

```
fligner.test(pl_a ~ group, data = df_absoluta_sub21)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
```

```
## data:  pl_a by group
## Fligner-Killeen:med chi-squared = 25.058, df = 1, p-value =
## 5.563e-07
```

```
fligner.test(pl_w ~ group, data = df_absoluta_sub21)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  pl_w by group
## Fligner-Killeen:med chi-squared = 36.657, df = 1, p-value =
## 1.408e-09
```

```
fligner.test(pl_pts ~ group, data = df_absoluta_sub21)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  pl_pts by group
## Fligner-Killeen:med chi-squared = 45.198, df = 1, p-value =
## 1.781e-11
```

Comprobamos que en todos los casos estamos ante una situación de heterocedasticidad.

Aplicación de pruebas estadísticas para comparar los grupos de datos. En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc. Aplicar al menos tres métodos de análisis diferentes.

A continuación, vamos a comparar las medias de las métricas para los distintos grupos mediante un test de Wilcoxon:

```
wilcox.test(df_A_B$pl_f~df_A_B$group)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  df_A_B$pl_f by df_A_B$group
## W = 3215424, p-value = 0.1339
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(df_A_B$pl_a~df_A_B$group)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  df_A_B$pl_a by df_A_B$group
## W = 3109552, p-value = 0.574
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(df_A_B$pl_w~df_A_B$group)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  df_A_B$pl_w by df_A_B$group
## W = 3166958, p-value = 0.5784
```

```
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(df_A_B$pl_pts~df_A_B$group)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  df_A_B$pl_pts by df_A_B$group
## W = 3175947, p-value = 0.4652
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(df_masculino_femenino$pl_f~df_masculino_femenino$group)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  df_masculino_femenino$pl_f by df_masculino_femenino$group
## W = 1228030, p-value = 8.021e-05
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(df_masculino_femenino$pl_a~df_masculino_femenino$group)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  df_masculino_femenino$pl_a by df_masculino_femenino$group
## W = 1179636, p-value = 0.01161
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(df_masculino_femenino$pl_w~df_masculino_femenino$group)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  df_masculino_femenino$pl_w by df_masculino_femenino$group
## W = 1178608, p-value = 0.01258
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(df_masculino_femenino$pl_pts~df_masculino_femenino$group)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  df_masculino_femenino$pl_pts by df_masculino_femenino$group
## W = 1116386, p-value = 0.5045
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(df_absoluta_sub21$pl_f~df_absoluta_sub21$group)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  df_absoluta_sub21$pl_f by df_absoluta_sub21$group
## W = 213666, p-value = 0.4212
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(df_absoluta_sub21$pl_a~df_absoluta_sub21$group)
```

```
##
```

```
##  Wilcoxon rank sum test with continuity correction
##
## data:  df_absoluta_sub21$pl_a by df_absoluta_sub21$group
## W = 214485, p-value = 0.3896
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(df_absoluta_sub21$pl_w~df_absoluta_sub21$group)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  df_absoluta_sub21$pl_w by df_absoluta_sub21$group
## W = 199038, p-value = 0.8456
## alternative hypothesis: true location shift is not equal to 0
```

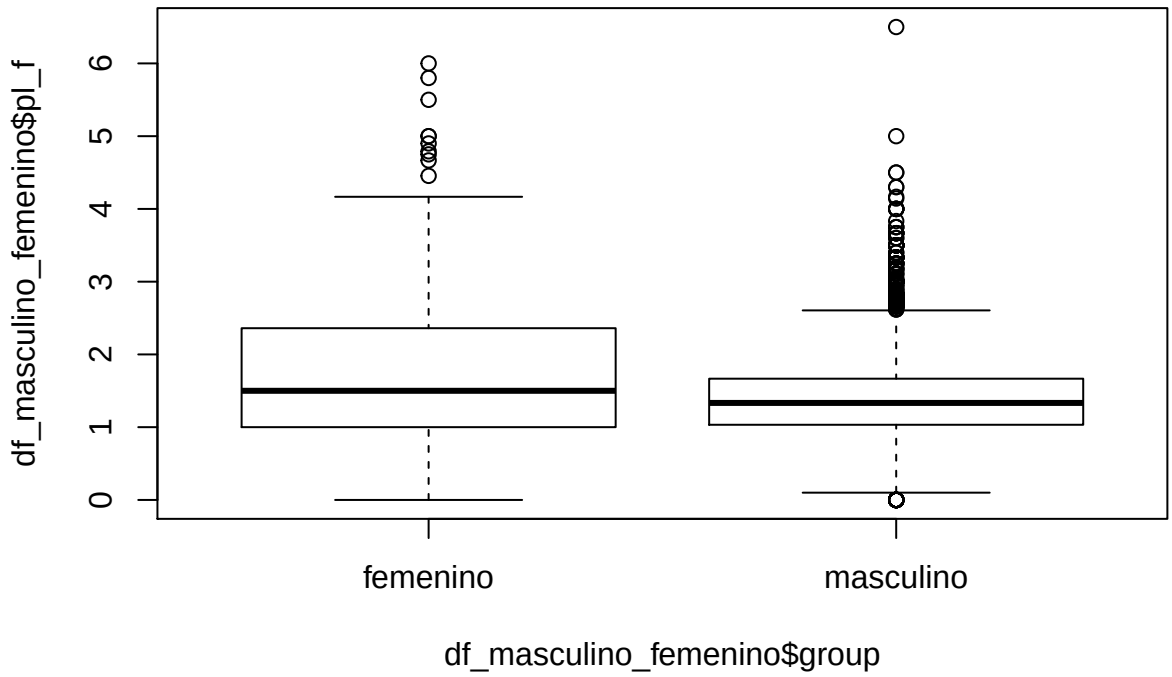
```
wilcox.test(df_absoluta_sub21$pl_pts~df_absoluta_sub21$group)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  df_absoluta_sub21$pl_pts by df_absoluta_sub21$group
## W = 200176, p-value = 0.907
## alternative hypothesis: true location shift is not equal to 0
```

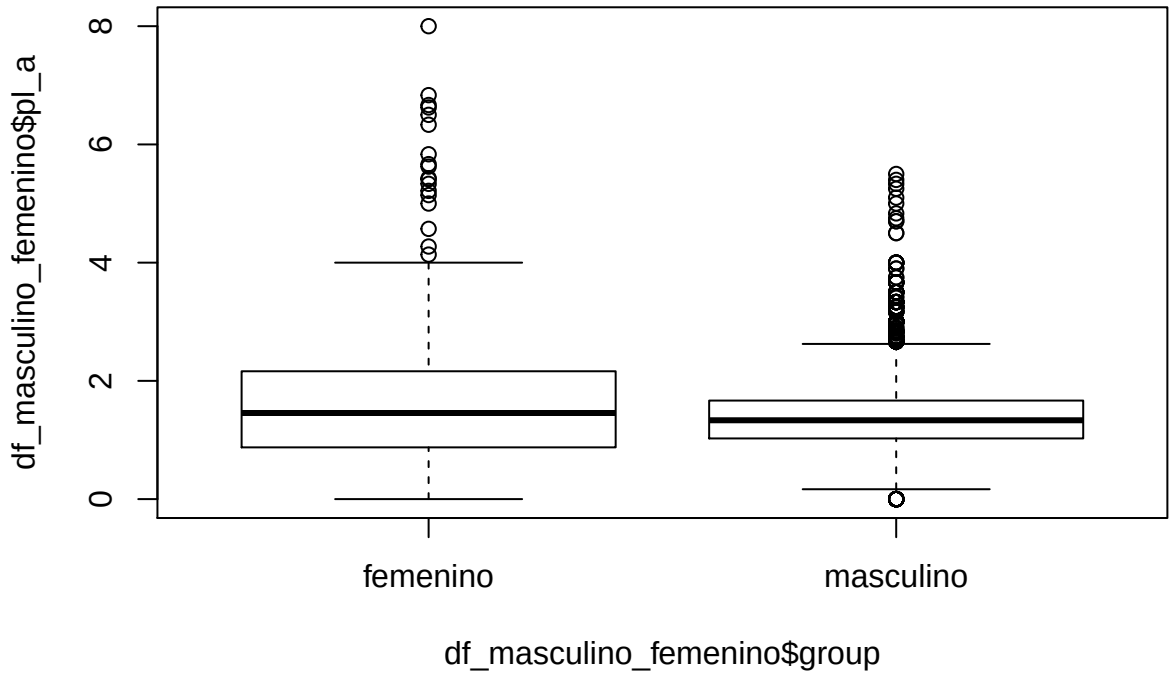
Encontramos resultados significativos, con p-valor menor que 0.05, al comparar goles y victorias entre equipos masculinos y femeninos.

Visualicemos estas distribuciones:

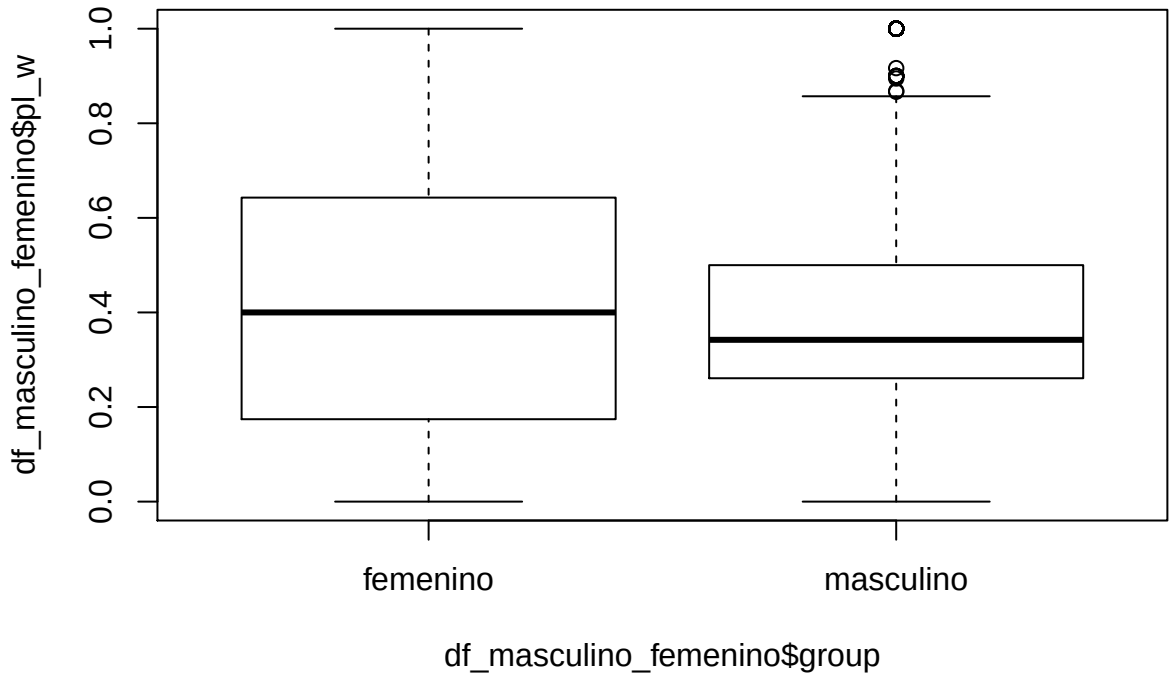
```
boxplot(df_masculino_femenino$pl_f~df_masculino_femenino$group)
```



```r
boxplot(df_masculino_femenino$pl_a~df_masculino_femenino$group)
```




```r
boxplot(df_masculino_femenino$pl_w~df_masculino_femenino$group)
```



En vista de los resultados del contraste entre medias, podemos concluir que el promedio de goles, tanto a favor como en contra, y de victorias es superior en las competiciones femeninas, frente a las competiciones masculinas.

Veamos ahora mediante un analisis de correlación en qué medida la cantidad de goles a favor va de la mano con el total de partidos ganados.

```r
cor.test(df_masculino$pl_f, df_masculino$pl_w, method="spearman")
```

```
## Warning in cor.test.default(df_masculino$pl_f, df_masculino$pl_w, method =
## "spearman"): Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  df_masculino$pl_f and df_masculino$pl_w
## S = 8873125946, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.7546301
```

```r
cor.test(df_masculino$pl_a, df_masculino$pl_w, method="spearman")
```

```
## Warning in cor.test.default(df_masculino$pl_a, df_masculino$pl_w, method =
## "spearman"): Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  df_masculino$pl_a and df_masculino$pl_w
## S = 6.0086e+10, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## -0.661568
```

```r
cor.test(df_femenino$pl_f, df_femenino$pl_w, method="spearman")
```

```
## Warning in cor.test.default(df_femenino$pl_f, df_femenino$pl_w, method =
## "spearman"): Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  df_femenino$pl_f and df_femenino$pl_w
## S = 1508666, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.812309
```

```r
cor.test(df_femenino$pl_a, df_femenino$pl_w, method="spearman")
```

```
## Warning in cor.test.default(df_femenino$pl_a, df_femenino$pl_w, method =
## "spearman"): Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  df_femenino$pl_a and df_femenino$pl_w
## S = 14223572, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
```

```
## -0.7695346
```

Vemos que las correlaciones entre goles y victorias son mucho mayores en las competiciones femeninas, lo que sugiere un menor número de empates.

Para valorar esta hipótesis, vamos a modelar el total de puntos obtenidos por un equipo mediante un modelo lineal:

```
summary(lm(df_masculino_femenino$pl_pts ~
             df_masculino_femenino$group + df_masculino_femenino$pl_f +
             df_masculino_femenino$pl_a + df_masculino_femenino$pl_w ))
```

```
##
## Call:
## lm(formula = df_masculino_femenino$pl_pts ~ df_masculino_femenino$group +
##     df_masculino_femenino$pl_f + df_masculino_femenino$pl_a +
##     df_masculino_femenino$pl_w)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.44789 -0.05808 -0.00461  0.04850  1.46149
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         0.635902   0.011885   53.51  < 2e-16
## df_masculino_femenino$groupmasculino  0.034877   0.007484    4.66 3.22e-06
## df_masculino_femenino$pl_f           0.121861   0.004318   28.22  < 2e-16
## df_masculino_femenino$pl_a          -0.187995   0.003627  -51.83  < 2e-16
## df_masculino_femenino$pl_w           2.132304   0.014891  143.19  < 2e-16
##
## (Intercept)                         ***
## df_masculino_femenino$groupmasculino ***
## df_masculino_femenino$pl_f          ***
## df_masculino_femenino$pl_a          ***
## df_masculino_femenino$pl_w          ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1356 on 6368 degrees of freedom
## Multiple R-squared:  0.9495, Adjusted R-squared:  0.9495
## F-statistic: 2.992e+04 on 4 and 6368 DF,  p-value: < 2.2e-16
```

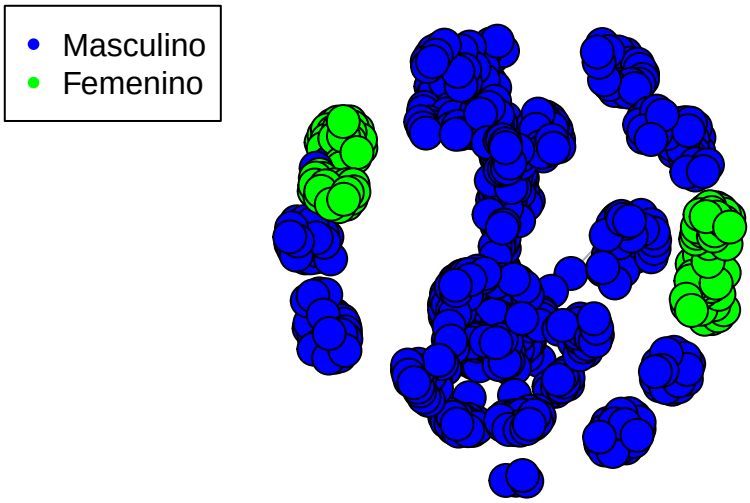
Podemos comprobar como, siendo todos los parámetros significativos, el más relevante es el número de victorias, como es de esperar. A igualdad de victorias, el modelo da mas importancia a evitar goles en contra que a conseguir goles a favor. Además, a igualdad de circunstancias, los equipos masculinos tienden a conseguir un mayor número de puntos, lo cual se explica por la mayor prevalencia de empates dentro de las categorías masculinas.

Gráficos, tablas y resolución del problema

Los resultados mas significativos que hemos encontrado se ubican en torno al género de los participantes:

```
bb <- df_clean[,c("pl_team", "female")] %>% distinct()
bbb <- df_clean[,c("comp", "female")] %>% distinct()
comm <- list()
```

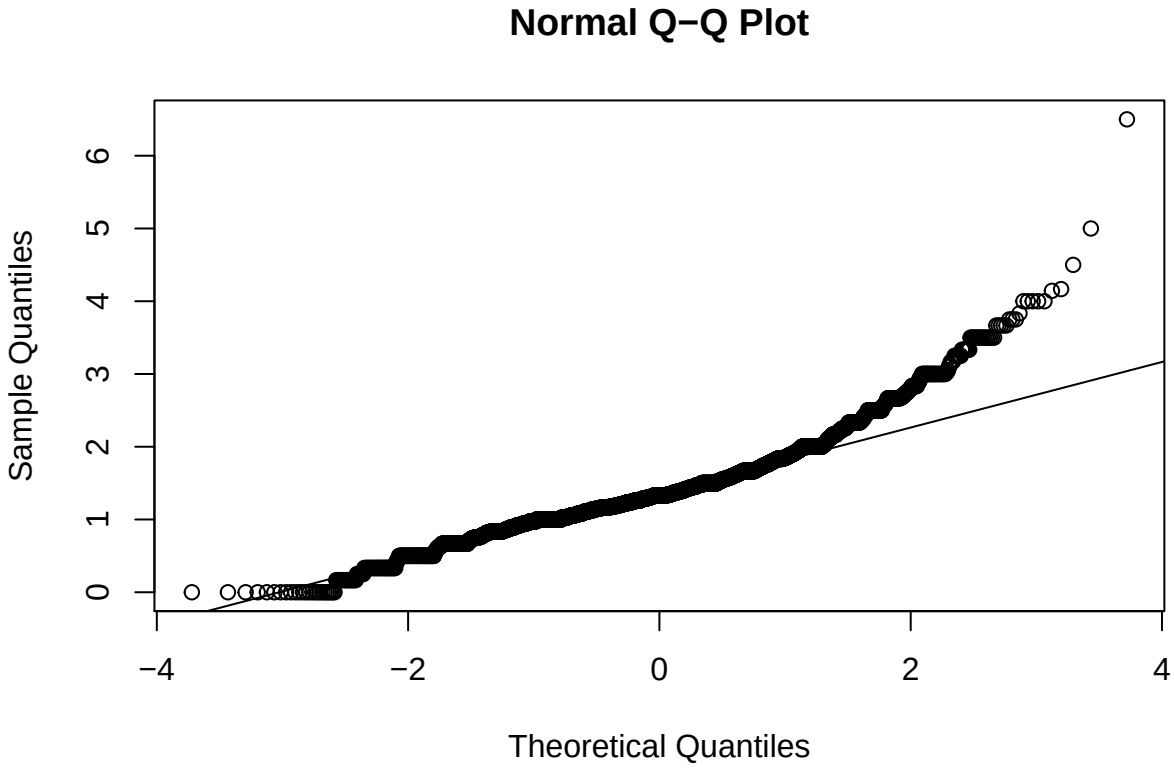
```
n <- length(bb[,1])
for (i in 1:n ){
comm[paste("T:",bb[i,"pl_team"])] <- bb[i,"female"]
}
n <- length(bbb[,1])
for (i in 1:n ){
comm[paste("C:",bbb[i,"comp"])] <- bbb[i,"female"]
}
col <- c()
for ( i in 1:length(V(graph)$name)){
  if (length(comm[[(V(graph)$name[i])]]) < 1) {
      col <- c(col, "#0000FF")
  } else {
  if (comm[[(V(graph)$name[i])]]) {
      col <- c(col, "#00FF00")
  } else {
      col <- c(col, "#0000FF")
  }}

}
plot(graph, vertex.label=NA, vertex.size=15, arrow.width=1,vertex.color=col);legend(
  'topleft',legend=c("Masculino", "Femenino"), pch=20, col=c("#0000FF","#00FF00"))
```



En este grafo podemos ver como las competiciones femeninas representan cúmulos alejados del nucleo central de la competición. Mientras que en la componente conexa de mayor tamaño no hemos encontrado diferencias estadísticas significativas, este aislamiento de las categorías femeninas hace más probable encontrar situaciones y estilos de juego que se alejen de la tendencia dominante.

Nos hemos encontrado con la dificultad de unas observaciones que, lejos de aproximarse a una distribución normal, se acumulan en gran medida en torno al mínimo a la vez mantienen una gran prevalencia de valores elevados:

```
qqnorm(df_A_B$pl_f); qqline(df_A_B$pl_f)
```



Esta situación es característica de escenarios competitivos, donde la mayoria de individuos lucha por mantenerse relevante, mientras que unos pocos muestran una holgada superioridad respecto al resto.

Observando los contrastes de hipótesis sobre las medias del futbol masculino y femenino, nos ha sorprendido ver que todas son significativamente diferentes exceptuando el total de puntos obtenidos:

```
a <- as.numeric(wilcox.test(df_masculino_femenino$pl_f~df_masculino_femenino$group)[3])
b <- as.numeric(wilcox.test(df_masculino_femenino$pl_a~df_masculino_femenino$group)[3])
c <- as.numeric(wilcox.test(df_masculino_femenino$pl_w~df_masculino_femenino$group)[3])
d <- as.numeric(wilcox.test(df_masculino_femenino$pl_pts~df_masculino_femenino$group)[3])

data.frame(
metrica=c("Goles a favor","Goles en contra","Victorias","Puntos"),
"p-valor"=c(a,b,c,d)
)
```

```
##           metrica      p.valor
## 1   Goles a favor 8.021329e-05
## 2 Goles en contra 1.161334e-02
## 3       Victorias 1.258042e-02
## 4          Puntos 5.044603e-01
```

Para ahondar en esto, hemos recurrido a una correlación de Spearman para entender mejor las similitudes y diferencias entre estos grupos de jugadores:

```
a <- as.numeric(cor.test(df_masculino$pl_f, df_masculino$pl_w, method="spearman")[4])
```

```
## Warning in cor.test.default(df_masculino$pl_f, df_masculino$pl_w, method =
## "spearman"): Cannot compute exact p-value with ties
```

```
b <- as.numeric(cor.test(df_masculino$pl_a, df_masculino$pl_w, method="spearman")[4])
```

```
## Warning in cor.test.default(df_masculino$pl_a, df_masculino$pl_w, method =
```

```
## "spearman"): Cannot compute exact p-value with ties
```

```
c <- as.numeric(cor.test(df_femenino$pl_f, df_femenino$pl_w, method="spearman")[4])
```

```
## Warning in cor.test.default(df_femenino$pl_f, df_femenino$pl_w, method =
## "spearman"): Cannot compute exact p-value with ties
```

```
d <- as.numeric(cor.test(df_femenino$pl_a, df_femenino$pl_w, method="spearman")[4])
```

```
## Warning in cor.test.default(df_femenino$pl_a, df_femenino$pl_w, method =
## "spearman"): Cannot compute exact p-value with ties
```

```
data.frame(
  Goles=c("A favor", "En contra"),
  Masculino=c(a,b),
  Femenino=c(c,d)
)
```

```
##       Goles  Masculino   Femenino
## 1   A favor  0.7546301  0.8123090
## 2 En contra -0.6615680 -0.7695346
```

Observamos por el coeficiente de correlacción de Spearman entre goles y victorias que, al tener mas impacto ambos tipos de goles en los partidos femeninos, en los partidos masculinos los goles conducen al empate con mayor frecuencia.

Por último, la tabla obtenida mediante el modelo lineal confirma nuestra conclusión sobre los empates, dado que otorga mayor puntuación a los hombres en circunstancias de ausencia de victorias:

```
summary(lm(df_masculino_femenino$pl_pts ~
             df_masculino_femenino$group + df_masculino_femenino$pl_f
           + df_masculino_femenino$pl_a + df_masculino_femenino$pl_w ))[4]
```

```
## $coefficients
##                                       Estimate  Std. Error    t value
## (Intercept)                          0.6359017 0.011884777  53.505562
## df_masculino_femenino$groupmasculino  0.0348769 0.007483572   4.660461
## df_masculino_femenino$pl_f            0.1218606 0.004317589  28.224218
## df_masculino_femenino$pl_a           -0.1879947 0.003627491 -51.824990
## df_masculino_femenino$pl_w            2.1323045 0.014891089 143.193318
##                                          Pr(>|t|)
## (Intercept)                          0.000000e+00
## df_masculino_femenino$groupmasculino 3.219248e-06
## df_masculino_femenino$pl_f           3.087866e-165
## df_masculino_femenino$pl_a           0.000000e+00
## df_masculino_femenino$pl_w           0.000000e+00
```

Contribuciones	Firma
Investigación previa	ICR, SDC
Redacción de las respuestas	ICR, SDC
Desarrollo código	ICR, SDC