PRA2: Limpieza y validación de datos

Tipología y ciclo de vida de los datos

Antonio de la Rubia y Enrique Martínez

30/05/2021

Contents

1. Descripción del dataset. ¿Por qué es importante y qué pregunta pretende resolver?

El dataset analizado contiene datos de los partidos de tenis jugados durante la temporada 2020 de la ATP. Está constituido por 49 columnas para 1462 registros. Este dataset es importante porque contiene información tanto de los jugadores como del entorno en el que se desarrolló el partido. Gracias a estos atributos, se pretende clasificar a los jugadores entre vencedores y perdedores en función de las estadísticas obtenidas en los partidos. Además, con este análisis, se pretende determinar la influencia/importancia de algunas de las variables en el resultado del partido.

Este tipo de análisis tiene gran relevancia en varios campos. El primero, evidentemente, serían los propios equipos o jugadores de la ATP que podrían analizar dichos datos y generar estrategias para el partido en función del contrincante, la superficie de juego, etc. Por otro lado, también son interesantes para todas aquellas casas de apuestas que basan sus cuotas de partido en función del histórico de partidos.

- Tourney_id: Año-ID del torneo (2020-8888)
- Tourney_name: Nombre del torneo (Atp Cup, Davis Cup)
- Surface: Superficie de la pista de juego (Hard, Clay, Grass)
- Draw_size: Número de jugadores del torneo (4, 8, 24)
- Tourney_level: Nivel del torneo (A, D, F)
- Tourney_date: año, me sy dia del partido (20200106)
- Match_num: Numero de partido (1, 2, 3)
- Winner_seed: cabeza de serie (1, 2, 3, 4)
- Winner_entry:
- Winner_name: Nombre del jugador vencedor (Novak Djokovic, Roberto Bautista, Rafael Nadal)
- Winner_hand: Mano dominante del jugador vencedor (R, L , U)
- Winner_ht: Altura del jugador vencedor (188, 190, 183)
- Winner_ioc: Nacionalidad del jugador vencedor (SRB, ESP, FRA)
- Winner_age: Edad del jugador vencedor (32'6, 34'7, 27'4)
- Loser_id: Identificador del jugador perdedor (104925, 105138)
- Loser_seed: cabeza de serie (1, 2, 3, 4)
- Loser_entry:
- Loser_name: Nombre del jugador perdedor (Novak Djokovic, Roberto Bautista, Rafael Nadal)
- Loser_hand: Mano dominante del jugador perdedor (R, L , U)
- Loser_ht: Altura del jugador perdedor (188, 190, 183)
- Loser_ioc: Nacionalidad del jugador perdedor (SRB, ESP, FRA)
- Loser_age: Edad del jugador perdedor (32'6, 34'7, 27'4)
- Score: Puntuación final del partido
- Best_of: Número de sets del partido (3, 5)
- Round: Ronda del torneo (F, SF, QF)
- Minutes: Duración del partido en minutos
- l_ace: Número de Ace's del jugador perdedor
- l_df: Número de dobles faltas del jugador perdedor
- l_svpt: Número de puntos de servicio del jugador perdedor
- l_1stIn: Número de primeros servicios dentro de la pista del jugador perdedor
- l_1stWon: Número de puntos por primer servicio del jugador perdedor
- l_2ndWon: Número de puntos por segundo servicio del jugador perdedor
- l_svgms: Número de juegos de servicio del jugador perdedor
- l_bpSaved: Número de puntos de break salvados del jugador perdedor
- l_bpFaced: Número de puntos de break enfrentados del jugador perdedor
- Winner_rank: Ranking del jugador vencedor
- Winner_rank_points: Puntos de ranking del jugador vencedor
- Loser_rank: Ranking del jugador perdedor
- Loser_rank_points: Puntos de ranking del jugador perdedor

2. Integración y selección de los datos de interés a analizar

En este caso, hemos optado por obtener los datos del GitHub del usuario @JeffSackmann. Este, dispone de un conjunto de datasets, de los cuales utilizaremos uno. El dataset seleccionado corresponde a los partidos jugados por los jugadores de la ATP durante el año 2020. En el siguiente enlace podemos encontrar el proyecto GitHub original.

https://github.com/JeffSackmann/tennis_atp/blob/master/atp_matches_2020.csv

En primer lugar, descargamos el fichero csv y lo pasamos a tabla Excel para una primera revisión manual.

Gracias a esta primera revisión, observamos que existen varias columnas con valores null para un gran número de registros. Estas columnas son:

Winner_seed, Winner_entry, Loser_seed y Loser_entry

Tras evaluar el contenido de dichos atributos, consideramos que podemos descartarlas del análisis que deseamos realizar.

Por otro lado, encontramos valores vacíos en algunos de los registros de las columnas winner_ht y looser_ht. En el siguiente apartado consideraremos como proceder con dichos registros.

3. Limpieza de los datos

En primer lugar, debemos realizar la lectura del fichero CSV. En este caso, utilizaremos el IDE RStudio para el análisis, por lo que deberemos hacer uso de la función read.csv que nos permite leer un fichero de texto plano separado por comas. El resultado de la función lo almacenaremos en el objeto dataframe 'df'.

```
#setwd("D:\UOC\Tipología y ciclo de vida de los datos\Prac2\PRACT2")

# Lectura de datos

df <- read.csv("atp_2020.csv", header = TRUE, na.strings=c("","NA"))
head(df)
```

```
##   tourney_id tourney_name surface draw_size tourney_level tourney_date
## 1  2020-8888      Atp Cup    Hard        24             A     20200106
## 2  2020-8888      Atp Cup    Hard        24             A     20200106
## 3  2020-8888      Atp Cup    Hard        24             A     20200106
## 4  2020-8888      Atp Cup    Hard        24             A     20200106
## 5  2020-8888      Atp Cup    Hard        24             A     20200106
## 6  2020-8888      Atp Cup    Hard        24             A     20200106
##   match_num winner_id winner_seed winner_entry           winner_name
## 1       300    104925          NA         <NA>        Novak Djokovic
## 2       299    105138          NA         <NA> Roberto Bautista Agut
## 3       298    104925          NA         <NA>        Novak Djokovic
## 4       297    105583          NA         <NA>         Dusan Lajovic
## 5       296    104745          NA         <NA>          Rafael Nadal
## 6       295    105138          NA         <NA> Roberto Bautista Agut
##   winner_hand winner_ht winner_ioc winner_age loser_id loser_seed loser_entry
## 1           R       188        SRB   32.62697   104745         NA        <NA>
## 2           R       183        ESP   31.72895   105583         NA        <NA>
## 3           R       188        SRB   32.62697   106421         NA        <NA>
## 4           R       180        SRB   29.51951   111575         NA        <NA>
## 5           L       185        ESP   33.59343   200282         NA        <NA>
```

```
## 6           R       183        ESP   31.72895   106401         NA        <NA>
##        loser_name loser_hand loser_ht loser_ioc loser_age       score best_of
## 1    Rafael Nadal          L      185       ESP  33.59343  6-2 7-6(4)       3
## 2   Dusan Lajovic          R      180       SRB  29.51951     7-5 6-1       3
## 3 Daniil Medvedev          R       NA       RUS  23.90144 6-1 5-7 6-4       3
## 4 Karen Khachanov          R       NA       RUS  23.62765  7-5 7-6(1)       3
## 5  Alex De Minaur          R       NA       AUS  20.88433 4-6 7-5 6-1       3
## 6    Nick Kyrgios          R      193       AUS  24.69541     6-1 6-4       3
##   round minutes w_ace w_df w_svpt w_1stIn w_1stWon w_2ndWon w_SvGms w_bpSaved
## 1     F     115    12    2     64      49       41        9      10         5
## 2     F      97     2    1     59      44       29       10      10         3
## 3    SF     167     4    5    111      75       53       16      15         8
## 4    SF     108     1    1     67      48       38       14      12         0
## 5    SF     133     5    3     84      61       48       10      15         1
## 6    SF      81     4    0     57      44       35        7       9         1
##   w_bpFaced l_ace l_df l_svpt l_1stIn l_1stWon l_2ndWon l_SvGms l_bpSaved
## 1         5     5    3     70      51       39        6      10         6
## 2         5     2    1     57      35       21        6       9         5
## 3        11     6    5    108      57       35       25      14         6
## 4         0     9    3     79      54       39       14      12         0
## 5         3     6    1     75      55       37       10      14         1
## 6         1    13    1     55      38       22        8       8         4
##   l_bpFaced winner_rank winner_rank_points loser_rank loser_rank_points
## 1         8           2               9055          1              9985
## 2        10          10               2335         34              1251
## 3        11           2               9055          5              5705
## 4         1          34               1251         17              1840
## 5         5           1               9985         18              1775
## 6         7          10               2335         29              1375
```

Para obtener el tipo de datos que almacena cada variable, utilizaremos la función sapply().

```
# Obtenemos el tipo de dato de cada atributo

sapply(df, function(x) class(x))
```

```
##         tourney_id       tourney_name            surface          draw_size
##        "character"        "character"        "character"          "integer"
##      tourney_level       tourney_date          match_num          winner_id
##        "character"          "integer"          "integer"          "integer"
##        winner_seed       winner_entry        winner_name        winner_hand
##          "integer"        "character"        "character"        "character"
##          winner_ht         winner_ioc         winner_age           loser_id
##          "integer"        "character"          "numeric"          "integer"
##         loser_seed        loser_entry        loser_name         loser_hand
##          "integer"        "character"        "character"        "character"
##           loser_ht          loser_ioc          loser_age              score
##          "integer"        "character"          "numeric"        "character"
##            best_of              round            minutes              w_ace
##          "integer"        "character"          "integer"          "integer"
##               w_df             w_svpt            w_1stIn           w_1stWon
##          "integer"          "integer"          "integer"          "integer"
##           w_2ndWon            w_SvGms          w_bpSaved          w_bpFaced
```

```
##          "integer"          "integer"          "integer"          "integer"
##              l_ace              l_df             l_svpt             l_1stIn
##          "integer"          "integer"          "integer"          "integer"
##           l_1stWon           l_2ndWon             l_SvGms          l_bpSaved
##          "integer"          "integer"          "integer"          "integer"
##          l_bpFaced       winner_rank winner_rank_points        loser_rank
##          "integer"          "integer"          "integer"          "integer"
##  loser_rank_points
##          "integer"
```

3.1 ¿Los datos contienen ceros o elementos vacíos?¿Cómo gestionarías cada uno de estos casos?

Como hemos visto anteriormente, encontramos 4 variables con datos nulos para muchos de los registros. Como para el tipo de análisis que vamos a realizar, no nos interesan esas variables, procedemos a descartarlas del dataframe. Para ello, generamos un vector con los nombres de las variables que queremos eliminar y lo pasamos al dataframe. De esta manera, se descartan del dataframe las columnas con nombres coincidentes a los definidos en el vector 'borrar'

```
# Eliminamos atributos con valores null

borrar <- c("winner_seed", "winner_entry", "loser_seed", "loser_entry")
df <- df[ , !names(df) %in% borrar]

head(df)
```

```
##   tourney_id tourney_name surface draw_size tourney_level tourney_date
## 1  2020-8888      Atp Cup    Hard        24             A     20200106
## 2  2020-8888      Atp Cup    Hard        24             A     20200106
## 3  2020-8888      Atp Cup    Hard        24             A     20200106
## 4  2020-8888      Atp Cup    Hard        24             A     20200106
## 5  2020-8888      Atp Cup    Hard        24             A     20200106
## 6  2020-8888      Atp Cup    Hard        24             A     20200106
##   match_num winner_id           winner_name winner_hand winner_ht winner_ioc
## 1       300    104925        Novak Djokovic           R       188        SRB
## 2       299    105138 Roberto Bautista Agut           R       183        ESP
## 3       298    104925        Novak Djokovic           R       188        SRB
## 4       297    105583         Dusan Lajovic           R       180        SRB
## 5       296    104745          Rafael Nadal           L       185        ESP
## 6       295    105138 Roberto Bautista Agut           R       183        ESP
##   winner_age loser_id      loser_name loser_hand loser_ht loser_ioc loser_age
## 1   32.62697   104745    Rafael Nadal          L      185       ESP  33.59343
## 2   31.72895   105583   Dusan Lajovic          R      180       SRB  29.51951
## 3   32.62697   106421 Daniil Medvedev          R       NA       RUS  23.90144
## 4   29.51951   111575 Karen Khachanov          R       NA       RUS  23.62765
## 5   33.59343   200282  Alex De Minaur          R       NA       AUS  20.88433
## 6   31.72895   106401    Nick Kyrgios          R      193       AUS  24.69541
##         score best_of round minutes w_ace w_df w_svpt w_1stIn w_1stWon w_2ndWon
## 1  6-2 7-6(4)       3     F     115    12    2     64      49       41        9
## 2     7-5 6-1       3     F      97     2    1     59      44       29       10
## 3 6-1 5-7 6-4       3    SF     167     4    5    111      75       53       16
## 4  7-5 7-6(1)       3    SF     108     1    1     67      48       38       14
## 5 4-6 7-5 6-1       3    SF     133     5    3     84      61       48       10
```

```
## 6     6-1 6-4       3    SF      81     4    0     57      44       35        7
##   w_SvGms w_bpSaved w_bpFaced l_ace l_df l_svpt l_1stIn l_1stWon l_2ndWon
## 1      10         5         5     5    3     70      51       39        6
## 2      10         3         5     2    1     57      35       21        6
## 3      15         8        11     6    5    108      57       35       25
## 4      12         0         0     9    3     79      54       39       14
## 5      15         1         3     6    1     75      55       37       10
## 6       9         1         1    13    1     55      38       22        8
##   l_SvGms l_bpSaved l_bpFaced winner_rank winner_rank_points loser_rank
## 1      10         6         8           2               9055          1
## 2       9         5        10          10               2335         34
## 3      14         6        11           2               9055          5
## 4      12         0         1          34               1251         17
## 5      14         1         5           1               9985         18
## 6       8         4         7          10               2335         29
##   loser_rank_points
## 1              9985
## 2              1251
## 3              5705
## 4              1840
## 5              1775
## 6              1375
```

En cuanto a los campos de winner_ht y loser_ht que representan la altura de los jugadores, hemos optado por sustituir los valores nulos por la media de altura de los registros.

Para ello, declaramos un vector con los nombres de las columnas que deseamos modificar. A continuación, calculamos la media de dichas columnas e iteramos sobre el dataframe modificando los valores NA de las columnas declaradas por la media de dichas columnas.

Observamos que los valores previamente marcados como NA se ven modificados por la media de la columna.

```
# Remplazos NA

# Declaramos las columnas cualitativas objetivo
columnas_remplazo <- colnames(df)[c(11, 13, 17, 19, 24:41)]

# Calculamos la media de las columnas cualitativas
cols_mean <- rep(NA, ncol(df[,colnames(df)[c(11, 13, 17, 19, 24:41)]]))

# Redondeo de las columnas
cols_mean <- round(cols_mean)

cols_mean[columnas_remplazo] <- colMeans(df[, columnas_remplazo], na.rm = TRUE, dims = 1)

# Sustituimos los valores NA por la media de la columna en las columnas objetivo
for (x in columnas_remplazo){
  df[is.na(df[,x]), x] <- cols_mean[x]
}
head(df)
```

```
##   tourney_id tourney_name surface draw_size tourney_level tourney_date
## 1  2020-8888      Atp Cup    Hard        24             A     20200106
## 2  2020-8888      Atp Cup    Hard        24             A     20200106
## 3  2020-8888      Atp Cup    Hard        24             A     20200106
```

```
## 4  2020-8888      Atp Cup    Hard        24             A     20200106
## 5  2020-8888      Atp Cup    Hard        24             A     20200106
## 6  2020-8888      Atp Cup    Hard        24             A     20200106
##   match_num winner_id           winner_name winner_hand winner_ht winner_ioc
## 1       300    104925        Novak Djokovic           R       188        SRB
## 2       299    105138 Roberto Bautista Agut           R       183        ESP
## 3       298    104925        Novak Djokovic           R       188        SRB
## 4       297    105583         Dusan Lajovic           R       180        SRB
## 5       296    104745          Rafael Nadal           L       185        ESP
## 6       295    105138 Roberto Bautista Agut           R       183        ESP
##   winner_age loser_id      loser_name loser_hand loser_ht loser_ioc loser_age
## 1   32.62697   104745    Rafael Nadal          L 185.0000       ESP  33.59343
## 2   31.72895   105583   Dusan Lajovic          R 180.0000       SRB  29.51951
## 3   32.62697   106421 Daniil Medvedev          R 185.2581       RUS  23.90144
## 4   29.51951   111575 Karen Khachanov          R 185.2581       RUS  23.62765
## 5   33.59343   200282  Alex De Minaur          R 185.2581       AUS  20.88433
## 6   31.72895   106401    Nick Kyrgios          R 193.0000       AUS  24.69541
##         score best_of round minutes w_ace w_df w_svpt w_1stIn w_1stWon w_2ndWon
## 1  6-2 7-6(4)       3     F     115    12    2     64      49       41        9
## 2     7-5 6-1       3     F      97     2    1     59      44       29       10
## 3 6-1 5-7 6-4       3    SF     167     4    5    111      75       53       16
## 4  7-5 7-6(1)       3    SF     108     1    1     67      48       38       14
## 5 4-6 7-5 6-1       3    SF     133     5    3     84      61       48       10
## 6     6-1 6-4       3    SF      81     4    0     57      44       35        7
##   w_SvGms w_bpSaved w_bpFaced l_ace l_df l_svpt l_1stIn l_1stWon l_2ndWon
## 1      10         5         5     5    3     70      51       39        6
## 2      10         3         5     2    1     57      35       21        6
## 3      15         8        11     6    5    108      57       35       25
## 4      12         0         0     9    3     79      54       39       14
## 5      15         1         3     6    1     75      55       37       10
## 6       9         1         1    13    1     55      38       22        8
##   l_SvGms l_bpSaved l_bpFaced winner_rank winner_rank_points loser_rank
## 1      10         6         8           2               9055          1
## 2       9         5        10          10               2335         34
## 3      14         6        11           2               9055          5
## 4      12         0         1          34               1251         17
## 5      14         1         5           1               9985         18
## 6       8         4         7          10               2335         29
##   loser_rank_points
## 1              9985
## 2              1251
## 3              5705
## 4              1840
## 5              1775
## 6              1375
```

En cuanto a la variable "looser_hand", observamos que existen varios valores marcados como NA. En primer lugar, obtenemos la frecuencia de cada uno de los posibles valores tanto para la variable "looser_hand" como para "winner_hand". A continuación, una vez confirmados que los valores NA solo aparecen en la variable que hace referencia a la mano dominante del jugador perdedor, procedemos a sustituir todos aquellos valores marcados como NA por la moda obtenida previamente. En este caso, la moda de la variable "looser_hand" es 'R'.

```r
# Evaluamos la frecuencia de mano dominante sobre los jugadores
table(df$winner_hand, useNA="always")
```

```
##
##    L    R    U <NA>
##  206 1224   32    0
```

```r
table(df$loser_hand, useNA="always")
```

```
##
##    L    R    U <NA>
##  206 1221   31    4
```

```r
# Sustituimos los valores NA por la moda de la columna, en este caso 'R'

getmode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}

df[is.na(df$loser_hand), "loser_hand"] <- getmode(df$loser_hand)

# Comprobamos que se han sustituido los valores correctamente

table(df$loser_hand, useNA="always")
```

```
##
##    L    R    U <NA>
##  206 1225   31    0
```

```r
head(df)
```

```
##   tourney_id tourney_name surface draw_size tourney_level tourney_date
## 1  2020-8888      Atp Cup    Hard        24             A     20200106
## 2  2020-8888      Atp Cup    Hard        24             A     20200106
## 3  2020-8888      Atp Cup    Hard        24             A     20200106
## 4  2020-8888      Atp Cup    Hard        24             A     20200106
## 5  2020-8888      Atp Cup    Hard        24             A     20200106
## 6  2020-8888      Atp Cup    Hard        24             A     20200106
##   match_num winner_id           winner_name winner_hand winner_ht winner_ioc
## 1       300    104925        Novak Djokovic           R       188        SRB
## 2       299    105138 Roberto Bautista Agut           R       183        ESP
## 3       298    104925        Novak Djokovic           R       188        SRB
## 4       297    105583         Dusan Lajovic           R       180        SRB
## 5       296    104745          Rafael Nadal           L       185        ESP
## 6       295    105138 Roberto Bautista Agut           R       183        ESP
##   winner_age loser_id      loser_name loser_hand loser_ht loser_ioc loser_age
## 1   32.62697   104745    Rafael Nadal          L 185.0000       ESP  33.59343
## 2   31.72895   105583   Dusan Lajovic          R 180.0000       SRB  29.51951
## 3   32.62697   106421 Daniil Medvedev          R 185.2581       RUS  23.90144
## 4   29.51951   111575 Karen Khachanov          R 185.2581       RUS  23.62765
```

```
## 5   33.59343   200282  Alex De Minaur          R 185.2581       AUS  20.88433
## 6   31.72895   106401    Nick Kyrgios          R 193.0000       AUS  24.69541
##         score best_of round minutes w_ace w_df w_svpt w_1stIn w_1stWon w_2ndWon
## 1  6-2 7-6(4)       3     F     115    12    2     64      49       41        9
## 2     7-5 6-1       3     F      97     2    1     59      44       29       10
## 3 6-1 5-7 6-4       3    SF     167     4    5    111      75       53       16
## 4  7-5 7-6(1)       3    SF     108     1    1     67      48       38       14
## 5 4-6 7-5 6-1       3    SF     133     5    3     84      61       48       10
## 6     6-1 6-4       3    SF      81     4    0     57      44       35        7
##   w_SvGms w_bpSaved w_bpFaced l_ace l_df l_svpt l_1stIn l_1stWon l_2ndWon
## 1      10         5         5     5    3     70      51       39        6
## 2      10         3         5     2    1     57      35       21        6
## 3      15         8        11     6    5    108      57       35       25
## 4      12         0         0     9    3     79      54       39       14
## 5      15         1         3     6    1     75      55       37       10
## 6       9         1         1    13    1     55      38       22        8
##   l_SvGms l_bpSaved l_bpFaced winner_rank winner_rank_points loser_rank
## 1      10         6         8           2               9055          1
## 2       9         5        10          10               2335         34
## 3      14         6        11           2               9055          5
## 4      12         0         1          34               1251         17
## 5      14         1         5           1               9985         18
## 6       8         4         7          10               2335         29
##   loser_rank_points
## 1              9985
## 2              1251
## 3              5705
## 4              1840
## 5              1775
## 6              1375
```

Para finalizar, redondeamos aquellas columnas que sean necesarias. En nuestro caso, redondeamos las columnas que hacen referencia a la altura y a la edad de los jugadores.

Redondeamos los valores de las columnas seleccionadas

```
df[,c("winner_ht", "winner_age", "loser_ht", "loser_age")] <-
  round(df[,c("winner_ht", "winner_age", "loser_ht", "loser_age")],0)

head(df)
```

```
##   tourney_id tourney_name surface draw_size tourney_level tourney_date
## 1  2020-8888      Atp Cup    Hard        24             A     20200106
## 2  2020-8888      Atp Cup    Hard        24             A     20200106
## 3  2020-8888      Atp Cup    Hard        24             A     20200106
## 4  2020-8888      Atp Cup    Hard        24             A     20200106
## 5  2020-8888      Atp Cup    Hard        24             A     20200106
## 6  2020-8888      Atp Cup    Hard        24             A     20200106
##   match_num winner_id           winner_name winner_hand winner_ht winner_ioc
## 1       300    104925        Novak Djokovic           R       188        SRB
## 2       299    105138 Roberto Bautista Agut           R       183        ESP
## 3       298    104925        Novak Djokovic           R       188        SRB
## 4       297    105583         Dusan Lajovic           R       180        SRB
## 5       296    104745          Rafael Nadal           L       185        ESP
```

```
## 6       295    105138 Roberto Bautista Agut           R       183        ESP
##   winner_age loser_id      loser_name loser_hand loser_ht loser_ioc loser_age
## 1         33   104745    Rafael Nadal          L      185       ESP        34
## 2         32   105583   Dusan Lajovic          R      180       SRB        30
## 3         33   106421 Daniil Medvedev          R      185       RUS        24
## 4         30   111575 Karen Khachanov          R      185       RUS        24
## 5         34   200282  Alex De Minaur          R      185       AUS        21
## 6         32   106401    Nick Kyrgios          R      193       AUS        25
##         score best_of round minutes w_ace w_df w_svpt w_1stIn w_1stWon w_2ndWon
## 1  6-2 7-6(4)       3     F     115    12    2     64      49       41        9
## 2     7-5 6-1       3     F      97     2    1     59      44       29       10
## 3 6-1 5-7 6-4       3    SF     167     4    5    111      75       53       16
## 4  7-5 7-6(1)       3    SF     108     1    1     67      48       38       14
## 5 4-6 7-5 6-1       3    SF     133     5    3     84      61       48       10
## 6     6-1 6-4       3    SF      81     4    0     57      44       35        7
##   w_SvGms w_bpSaved w_bpFaced l_ace l_df l_svpt l_1stIn l_1stWon l_2ndWon
## 1      10         5         5     5    3     70      51       39        6
## 2      10         3         5     2    1     57      35       21        6
## 3      15         8        11     6    5    108      57       35       25
## 4      12         0         0     9    3     79      54       39       14
## 5      15         1         3     6    1     75      55       37       10
## 6       9         1         1    13    1     55      38       22        8
##   l_SvGms l_bpSaved l_bpFaced winner_rank winner_rank_points loser_rank
## 1      10         6         8           2              9055          1
## 2       9         5        10          10              2335         34
## 3      14         6        11           2              9055          5
## 4      12         0         1          34              1251         17
## 5      14         1         5           1              9985         18
## 6       8         4         7          10              2335         29
##   loser_rank_points
## 1              9985
## 2              1251
## 3              5705
## 4              1840
## 5              1775
## 6              1375
```

3.2 Identificación y tratamiento de valores extremos

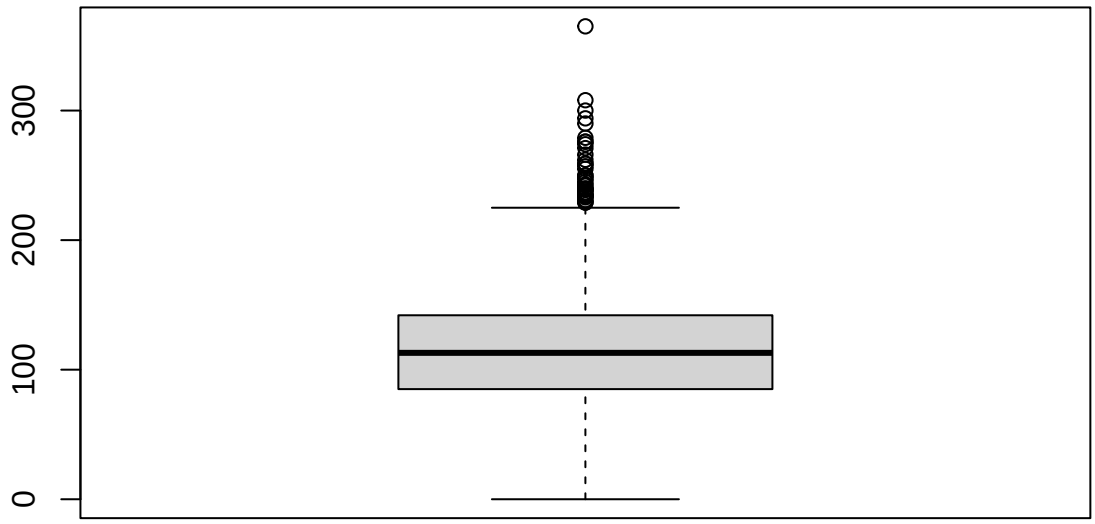
En este apartado se pretenden identificar los valores extremos o outliers. Los outliers son aquellos que parecen ser erróneos al compararlos con el resto de los datos. En este caso, se ha decidido por utilizar la función boxplot.stats() proporcionada por R para realizar el análisis de los valores extremos.

```
# Redondeamos los valores de las columnas seleccionadas

df[,c("winner_ht", "winner_age", "loser_ht", "loser_age")] <-
  round(df[,c("winner_ht", "winner_age", "loser_ht", "loser_age")],0)

head(df)
```

```
##   tourney_id tourney_name surface draw_size tourney_level tourney_date
## 1  2020-8888      Atp Cup    Hard        24             A     20200106
## 2  2020-8888      Atp Cup    Hard        24             A     20200106
```

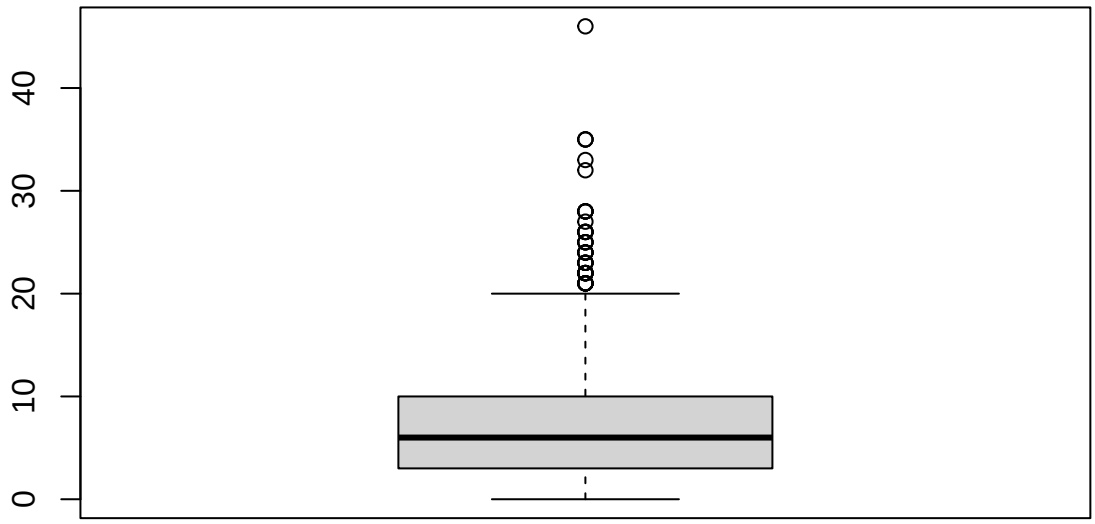
```
## 3  2020-8888      Atp Cup    Hard        24             A     20200106
## 4  2020-8888      Atp Cup    Hard        24             A     20200106
## 5  2020-8888      Atp Cup    Hard        24             A     20200106
## 6  2020-8888      Atp Cup    Hard        24             A     20200106
##   match_num winner_id          winner_name winner_hand winner_ht winner_ioc
## 1       300    104925        Novak Djokovic           R       188        SRB
## 2       299    105138 Roberto Bautista Agut           R       183        ESP
## 3       298    104925        Novak Djokovic           R       188        SRB
## 4       297    105583         Dusan Lajovic           R       180        SRB
## 5       296    104745          Rafael Nadal           L       185        ESP
## 6       295    105138 Roberto Bautista Agut           R       183        ESP
##   winner_age loser_id      loser_name loser_hand loser_ht loser_ioc loser_age
## 1         33   104745    Rafael Nadal          L      185       ESP        34
## 2         32   105583   Dusan Lajovic          R      180       SRB        30
## 3         33   106421 Daniil Medvedev          R      185       RUS        24
## 4         30   111575 Karen Khachanov          R      185       RUS        24
## 5         34   200282  Alex De Minaur          R      185       AUS        21
## 6         32   106401    Nick Kyrgios          R      193       AUS        25
##         score best_of round minutes w_ace w_df w_svpt w_1stIn w_1stWon w_2ndWon
## 1  6-2 7-6(4)       3     F     115    12    2     64      49       41        9
## 2     7-5 6-1       3     F      97     2    1     59      44       29       10
## 3 6-1 5-7 6-4       3    SF     167     4    5    111      75       53       16
## 4  7-5 7-6(1)       3    SF     108     1    1     67      48       38       14
## 5 4-6 7-5 6-1       3    SF     133     5    3     84      61       48       10
## 6     6-1 6-4       3    SF      81     4    0     57      44       35        7
##   w_SvGms w_bpSaved w_bpFaced l_ace l_df l_svpt l_1stIn l_1stWon l_2ndWon
## 1      10         5         5     5    3     70      51       39        6
## 2      10         3         5     2    1     57      35       21        6
## 3      15         8        11     6    5    108      57       35       25
## 4      12         0         0     9    3     79      54       39       14
## 5      15         1         3     6    1     75      55       37       10
## 6       9         1         1    13    1     55      38       22        8
##   l_SvGms l_bpSaved l_bpFaced winner_rank winner_rank_points loser_rank
## 1      10         6         8           2               9055          1
## 2       9         5        10          10               2335         34
## 3      14         6        11           2               9055          5
## 4      12         0         1          34               1251         17
## 5      14         1         5           1               9985         18
## 6       8         4         7          10               2335         29
##   loser_rank_points
## 1              9985
## 2              1251
## 3              5705
## 4              1840
## 5              1775
## 6              1375
```

```
# Gestión de los valores extremos para algunas de las variables

# Minutos jugados
boxplot(df$minutes)$out
```



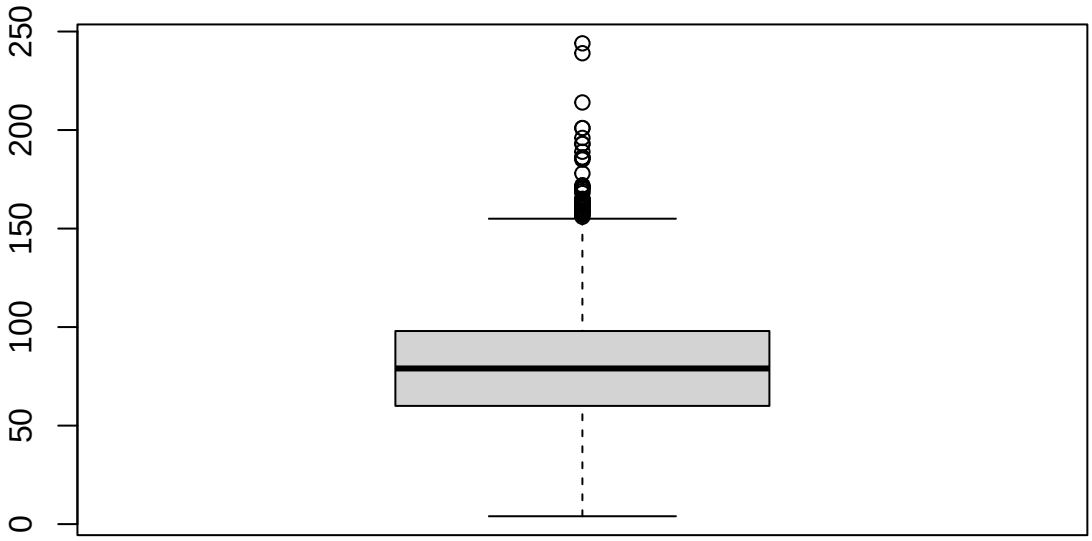

```
##  [1] 236 234 235 234 257 274 245 259 266 243 250 250 239 229 230 239 231 279 259
## [20] 239 290 237 255 276 248 242 271 240 262 247 276 365 294 229 300 239 233 234
## [39] 308 234
```

```
# Numero w_Ace
boxplot(df$w_ace)$out
```



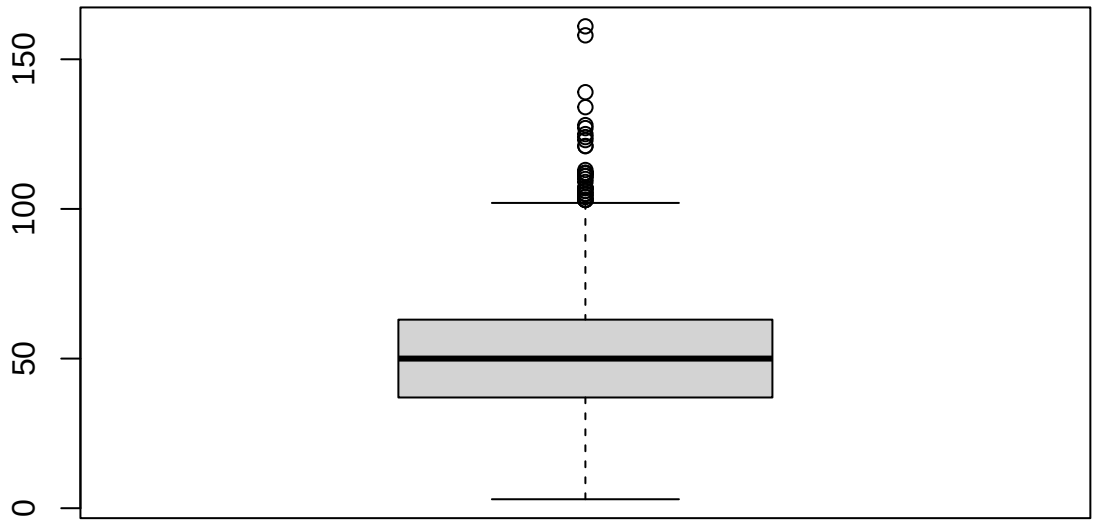

```
##  [1] 25 21 28 25 28 46 21 22 21 22 28 32 22 26 33 24 21 35 26 28 27 22 28 35 22
## [26] 24 35 23 22 23 26 23 26 24 24 22 24 21 23 23 22 26 25 21 25 21 22 21
```

```
# Numero w_svpt
boxplot(df$w_svpt)$out
```



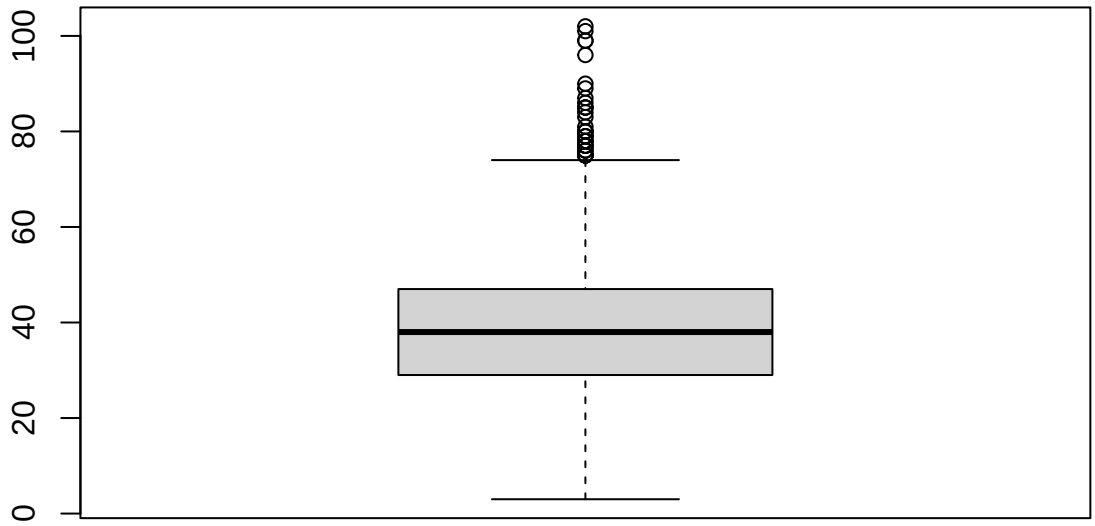

```
##  [1] 157 161 159 156 171 193 159 159 186 165 171 163 164 214 160 186 170 171 185
## [20] 157 171 163 159 193 178 168 172 171 162 165 169 186 158 161 201 158 162 201
## [39] 244 239 196 165 164 189
```

```
# Numero w_1stIn
boxplot(df$w_1stIn)$out
```



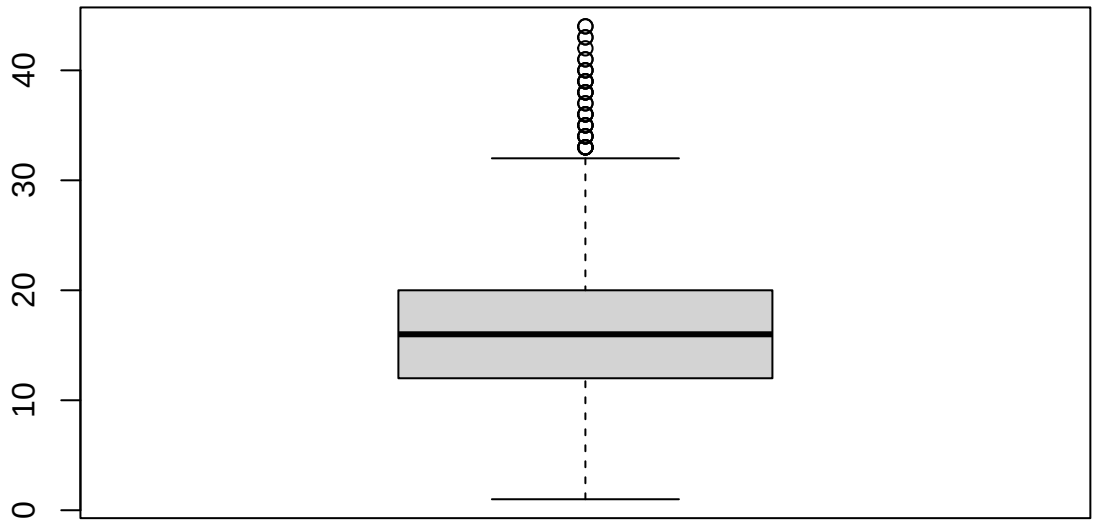

```
##  [1] 105 121 103 109 112 128 121 104 105 123 107 125 111 112 107 104 113 111 107
## [20] 110 103 103 107 112 103 111 112 106 134 105 139 158 161 106 127 112 124
```

```
# Numero w_1stWon
boxplot(df$w_1stWon)$out
```



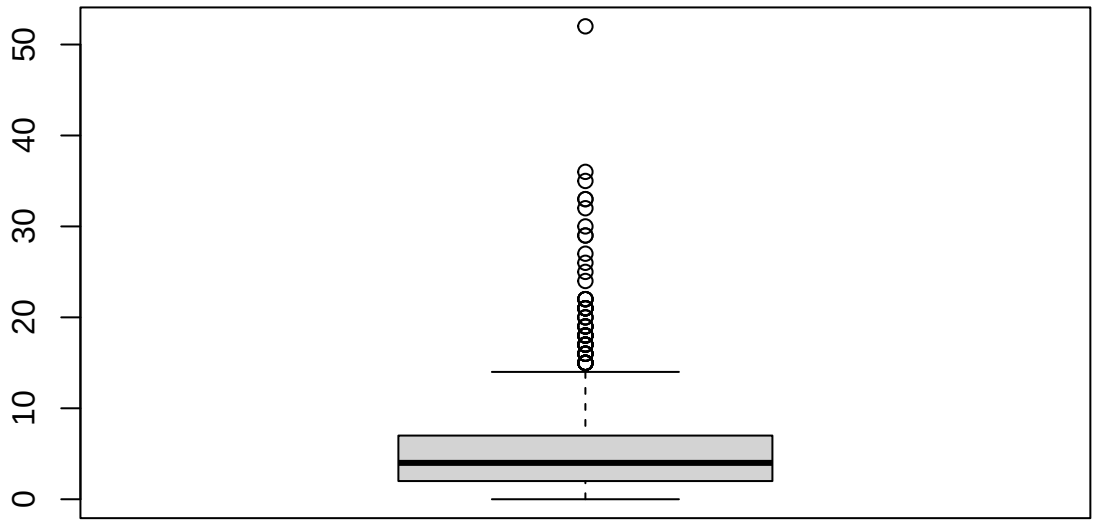

```
##  [1]  80  78  99  80  80  77  89  83  79  76  84  86  85 102  85  75  80  75  96
## [20]  76  78  75  77  79  85  75  75  81  78  77  75  76  75  87  75  90  99 101
## [39]  78  79  77
```

```
# Numero w_2ndWon
boxplot(df$w_2ndWon)$out
```



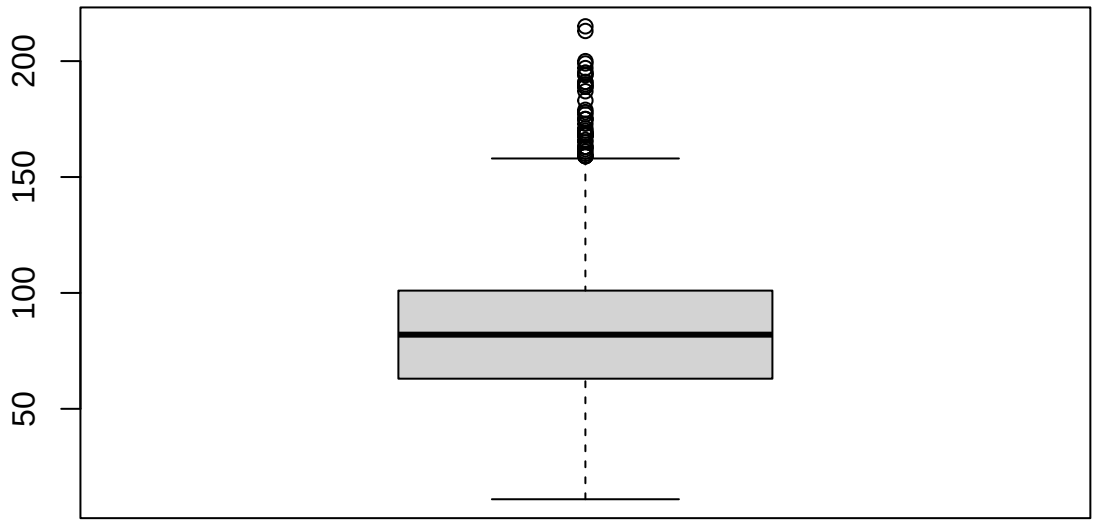

```
##  [1] 40 35 33 36 44 37 39 43 43 37 36 42 44 40 34 35 39 35 39 34 34 38 33 33 35
## [26] 41 38 36 40 36 39 33 41 38 38 33 33 34
```

```
# Numero l_Ace
boxplot(df$l_ace)$out
```



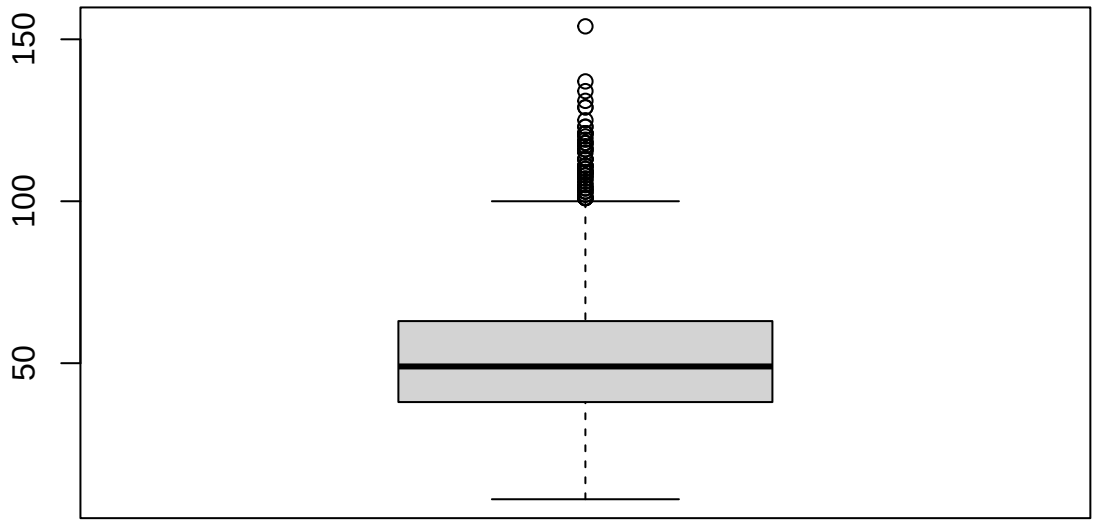

```
##  [1] 15 17 15 18 15 33 15 18 19 36 16 18 22 21 18 15 35 32 17 21 16 21 30 21 20
## [26] 15 25 19 27 18 16 15 22 15 17 24 22 33 17 16 15 17 15 18 52 18 20 21 17 22
## [51] 20 19 21 17 21 29 18 16 26 15 17 15 29 16 18 16 15 22 19 21 17 15 19
```

```
# Numero l_svpt
boxplot(df$l_svpt)$out
```



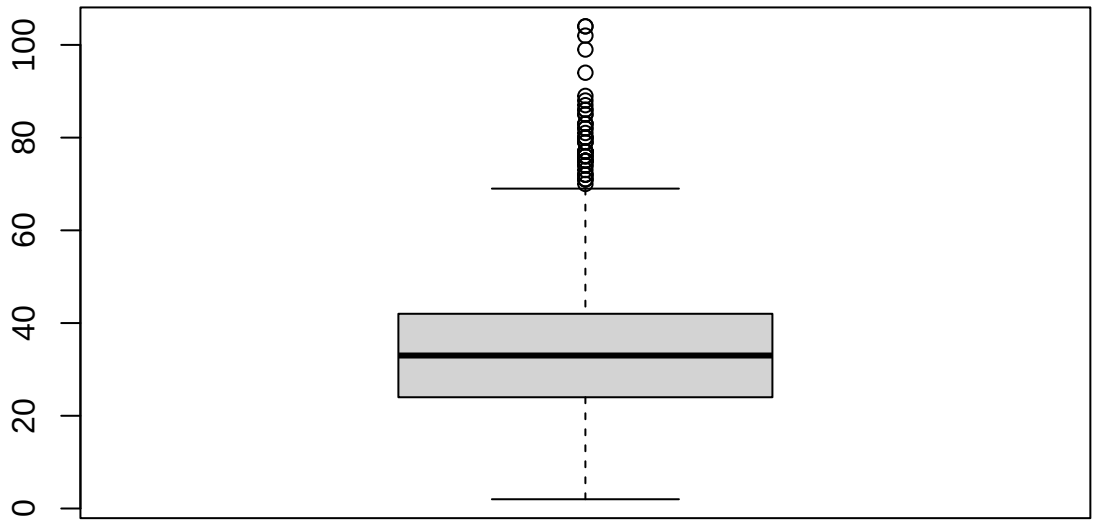

```
##  [1] 191 165 168 173 168 177 168 183 175 189 163 166 169 189 190 189 168 170 159
## [20] 171 161 159 175 169 191 162 168 213 163 179 169 200 197 194 175 160 195 195
## [39] 215 199 178 159 194 163 168 159 187 168
```

```
# Numero l_1stIn
boxplot(df$l_1stIn)$out
```



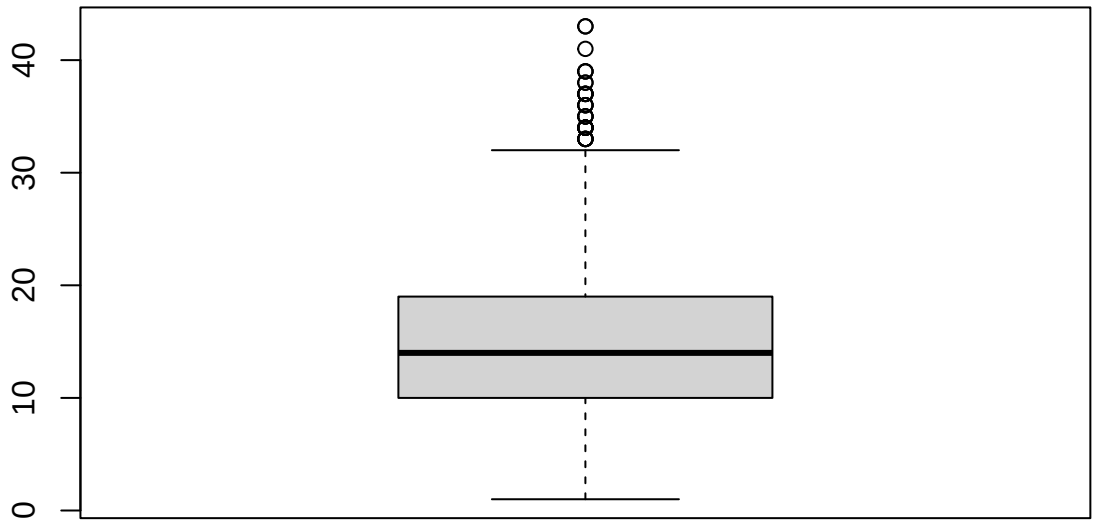

```
##  [1] 101 103 111 120 121 111 116 121 115 118 118 110 109 123 109 110 101 125 129
## [20] 118 116 113 105 103 108 104 101 134 107 119 101 108 113 109 104 102 137 121
## [39] 104 106 101 123 111 117 105 109 107 108 131 116 154 129 113 109 121 101 103
## [58] 111 113
```

```
# Numero l_1stWon
boxplot(df$l_1stWon)$out
```




```
##  [1]  77  75  80  76  76 102  87  77  75  86  75  88  94  71  71  83  82  74  85
## [20]  73 104  80  76  83  72  77  70  75  79  99  76  72  80  85  77  77  72  89
## [39]  75  81  82  71  86  75  79  79  77  77 104  83  74  80
```

```
# Numero l_2ndWon
boxplot(df$l_2ndWon)$out
```




```
##  [1] 36 41 33 34 33 34 34 43 33 34 38 34 38 39 37 34 34 39 37 33 34 34 35 35 37
## [26] 39 37 43 36 35 33 34 36 37 35 33 37 36
```

Tras el análisis de los resultados obtenidos con la función boxplot.stats(), comparando entre los valores obtenidos para los vencedores y perdedores así como para otras temporadas, concluimos que los valores son posibles.

En el caso de la duración del partido, el reglamento en tenis no establece una duración máxima del encuentro. De hecho, se han dado situaciones en las que el partido se ha disputado en varios días por lo que entendemos que los valores contenidos en el dataset son correctos.

En relación a las métricas del partido, se han analizado los casos en cuestión y se ha determinado que los datos representados son plausibles.

En conclusión, podemos considerar que los valores marcados como outliers no son tal y que por tanto no necesitan de tratamiento alguno y serán contabilizados en los análisis posteriores.

4. Análisis de los datos.

En el presente apartado, llevaremos a cabo el análisis de los datos contenidos en el dataset, ajustado por los cambios mencionados en los anteriores apartados.

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

En este apartado vamos a crear grupos de datos que posteriormente utilizaremos en nuestro análisis.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# Creamos un dataframe con todos los jugadores que
# han disputado algún partido de los torneos de 2020.
jugadores <- df[,8:13]
colnames(jugadores) <- c("id", "name", "hand", "height", "country", "age")
todos_jugadores <- jugadores
jugadores <- df[,14:19]
colnames(jugadores) <- c("id", "name", "hand", "height", "country", "age")
todos_jugadores <- rbind(todos_jugadores, jugadores)
rm(jugadores)
todos_jugadores <- distinct(todos_jugadores, id, .keep_all = TRUE)

# Adicionalmente, creamos nuevos campos con las principales
# estadisticas de juego.
win <- rep(0, nrow(todos_jugadores))
lose <- win
aces <- win
d_faults <- win
svpt <- win
first_in <- win
first_won <- win
second_won <- win
svgms <- win
bp_faced <- win

for (i in 1:nrow(todos_jugadores)){
  for (j in 1:nrow(df)){
    if (todos_jugadores$id[i]==df$winner_id[j]){
      win[i] <- win[i]+1
      aces[i] <- aces[i] + df$w_ace[j]
      d_faults[i] <- d_faults[i] + df$w_df[j]
      svpt[i] <- svpt[i] + df$w_svpt[j]
      first_in[i] <- first_in[i] + df$w_1stIn[j]
      first_won[i] <- first_won[i] + df$w_1stWon[j]
      second_won[i] <- second_won[i] + df$w_2ndWon[j]
      svgms[i] <- svgms[i] + df$w_SvGms[j]
      bp_faced[i] <- bp_faced[i] + df$w_bpFaced[j]
    } else if (todos_jugadores$id[i]==df$loser_id[j]){
      lose[i] <- lose[i]+1
      aces[i] <- aces[i] + df$l_ace[j]
```

```
      d_faults[i] <- d_faults[i] + df$l_df[j]
      svpt[i] <- svpt[i] + df$l_svpt[j]
      first_in[i] <- first_in[i] + df$l_1stIn[j]
      first_won[i] <- first_won[i] + df$l_1stWon[j]
      second_won[i] <- second_won[i] + df$l_2ndWon[j]
      svgms[i] <- svgms[i] + df$l_SvGms[j]
      bp_faced[i] <- bp_faced[i] + df$l_bpFaced[j]
    }
  }
}
# Creamos la variables victorias netas por partido jugado.
net_wins <- (win-lose)/(win+lose)

# Calculamos todas las variables en función de los partidos
# disputados por cada jugador.
aces <- aces/(win+lose)
d_faults <- d_faults/(win+lose)
svpt <- svpt/(win+lose)
first_in <- first_in/(win+lose)
first_won <- first_won/(win+lose)
second_won <- second_won/(win+lose)
svgms <- svgms/(win+lose)
bp_faced <- bp_faced/(win+lose)

# Integramos las nuevas variables a nuestro dataframe.
nuevos_campos <- cbind(win, lose, net_wins,
                       aces, d_faults, svpt,
                       first_in, first_won,
                       second_won, svgms,
                       bp_faced)

todos_jugadores <- cbind(todos_jugadores, nuevos_campos)

# NOTA: Omitimos la variable de break points salvados ya que vemos que está incluida
# su relación con las victorias a través de los break point enfrentados.


# Agrupamos entre jugadores diestros, zurdos y ambidiestro.
todos_jugadores_dcha <- todos_jugadores[todos_jugadores$hand=="R",]
todos_jugadores_izq <- todos_jugadores[todos_jugadores$hand=="L",]
todos_jugadores_amb <- todos_jugadores[todos_jugadores$hand=="U",]
```

4.2. Comprobación de la normalidad y homogeneidad de la varianza.

Con el fin de comprobar que los valores de las variables cuantitativas de nuestro dataset provienen de una población distribuida de manera normal, vamos a aplicar a nuestro caso la prueba de normalidad de **Anderson-Darling**.

De esta forma, comprobamos que para cada una de las pruebas que ralizamos se obtiene un p-valor superior al nivel de significación prefijado en $\alpha = 0.05$. Si esto se cumple, entonces consideraremos que la variable cuantitativa en cuestión sigue una distribución normal.

```
library(nortest)
alpha = 0.05
nombre_col = colnames(todos_jugadores)
for (i in 1:ncol(todos_jugadores)) {
  if (i == 1) cat("Las variables que NO siguen una distribución normal son las siguientes:\n")
  if (is.integer(todos_jugadores[,i]) | is.numeric(todos_jugadores[,i])) {
    p_val = ad.test(todos_jugadores[,i])$p.value
    if (p_val < alpha) {
      cat(nombre_col[i])
      # Format output
      if (i <= ncol(todos_jugadores) - 1) cat(", ")
    }
  }
}
```

```
## Las variables que NO siguen una distribución normal son las siguientes:
## id, height, age, win, lose, net_wins, aces, d_faults, svpt, first_in, first_won, second_won, svgms, b
```

Vemos que ninguna de las variables cuantitativas sigue esta distribución.

A continuación, vamos a comprobar la homogeneidad de varianzas a través de la aplicación del test de **Fligner-Killeen**. En este caso, estudiaremos esta homogeneidad en cuanto a los grupos conformados por jugadores diestros y zurdos. En este test, la hipótesis nula se basa en afirmar que ambas varianzas son iguales.

```
fligner.test(net_wins ~ hand, data = todos_jugadores)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  net_wins by hand
## Fligner-Killeen:med chi-squared = 33.124, df = 2, p-value = 6.414e-08
```

En conclusión, dado que obtenemos un p-valor superior a 0.05, debemos aceptamos la hipótesis de que las varianzas de ambas muestras son homogéneas. En el caso siguiente (nacionalidad), dado que el p-valor es inferior (aunque próximo), rechazaríamos la hipótesis de que las varianzas son homogéneas.

```
fligner.test(net_wins ~ country, data = todos_jugadores)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  net_wins by country
## Fligner-Killeen:med chi-squared = 109.85, df = 79, p-value = 0.01242
```

4.3. 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.

4.3.1. Correlación.

La primera prueba estadística que vamos a realizar es la comprobación de la **correlación** que existe entre las caracteristicas de los jugadores con el número de victorias netas para saber cuales de ellas pueden afectan más a esta variable. Dado que hemos visto que nuestras variables no son normales, utilizamos el test de Spearman para obtener las correlaciones.

Como matriz de correlaciones podemos ejecutar la siguiente función:

```
cor(todos_jugadores[,c(4,6,9:ncol(todos_jugadores))], method = "spearman")
```

```
##                  height          age    net_wins       aces    d_faults
## height      1.000000000 -0.023551815  0.46710036  0.2257899 -0.03686477
## age        -0.023551815  1.000000000  0.04905146  0.0698271 -0.04029752
## net_wins    0.467100364  0.049051462  1.00000000  0.2324350 -0.12833086
## aces        0.225789897  0.069827104  0.23243496  1.0000000  0.18829964
## d_faults   -0.036864767 -0.040297516 -0.12833086  0.1882996  1.00000000
## svpt       -0.007054795 -0.006575512 -0.02728415  0.3936472  0.28407703
## first_in    0.039883871 -0.034830728  0.03031453  0.3320862  0.16150140
## first_won   0.211568869  0.029755305  0.29642134  0.5880776  0.16024037
## second_won  0.027844830  0.044084021  0.16731375  0.4239546  0.20200539
## svgms       0.108953869 -0.004630439  0.15297291  0.4810926  0.22895032
## bp_faced   -0.305771319 -0.114462316 -0.52121332 -0.2007664  0.23621188
##                   svpt    first_in  first_won second_won        svgms
## height     -0.007054795  0.03988387 0.21156887 0.02784483  0.108953869
## age        -0.006575512 -0.03483073 0.02975531 0.04408402 -0.004630439
## net_wins   -0.027284150  0.03031453 0.29642134 0.16731375  0.152972911
## aces        0.393647196  0.33208624 0.58807756 0.42395463  0.481092609
## d_faults    0.284077031  0.16150140 0.16024037 0.20200539  0.228950320
## svpt        1.000000000  0.90252380 0.80676951 0.69074938  0.891110042
## first_in    0.902523801  1.00000000 0.86160723 0.47309247  0.824593428
## first_won   0.806769510  0.86160723 1.00000000 0.57073160  0.878474287
## second_won  0.690749381  0.47309247 0.57073160 1.00000000  0.721071324
## svgms       0.891110042  0.82459343 0.87847429 0.72107132  1.000000000
## bp_faced    0.439903988  0.39372039 0.06164665 0.06931743  0.190695659
##               bp_faced
## height     -0.30577132
## age        -0.11446232
## net_wins   -0.52121332
## aces       -0.20076643
## d_faults    0.23621188
## svpt        0.43990399
## first_in    0.39372039
## first_won   0.06164665
## second_won  0.06931743
## svgms       0.19069566
## bp_faced    1.00000000
```

Podemos ver la relación entre las principales variables que afectan a las victorias netas y su p-valor (puesto que este puede dar información acerca del peso estadístico de la correlación obtenida):

```
cor.test(todos_jugadores$net_wins, todos_jugadores$bp_faced,
         method = "spearman", exact = FALSE)
```

```
##
##  Spearman's rank correlation rho
##
## data:  todos_jugadores$net_wins and todos_jugadores$bp_faced
## S = 10411001, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.5212133
```

```
cor.test(todos_jugadores$net_wins, todos_jugadores$height,
         method = "spearman", exact = FALSE)
```

```
##
##  Spearman's rank correlation rho
##
## data:  todos_jugadores$net_wins and todos_jugadores$height
## S = 3647101, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.4671004
```

```
cor.test(todos_jugadores$net_wins, todos_jugadores$first_won,
         method = "spearman", exact = FALSE)
```

```
##
##  Spearman's rank correlation rho
##
## data:  todos_jugadores$net_wins and todos_jugadores$first_won
## S = 4815208, p-value = 1.995e-08
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.2964213
```

```
cor.test(todos_jugadores$net_wins, todos_jugadores$aces,
         method = "spearman", exact = FALSE)
```

```
##
##  Spearman's rank correlation rho
##
## data:  todos_jugadores$net_wins and todos_jugadores$aces
## S = 5253123, p-value = 1.292e-05
## alternative hypothesis: true rho is not equal to 0
```

```
## sample estimates:
##      rho
## 0.232435
```

Aunque no podemos decir que exista una correlación muy fuerte entre las variables analizadas y el número de victorias, sí que destacamos cinco que afectan por encima de las demás: el número de break points en contra (relación inversa), la altura, el número de puntos ganados con el primer servicio y el número de saques directos.

4.3.2. Regresión lineal.

A través de un modelo de regresión lineal se puede comprobar el efecto que tiene sobre las victorias de un jugador cada una de las variables que afectan a su juego: altura, edad, aces... De esta forma, se podría incluso prever las victorias de un tenista dada una serie de caracteristicas. Para desarrollar este apartado, plantearemos varios modelos de regresión lineal y posteriormente analizaremos su bondad del ajuste para quedarnos con el que mayor bondad tenga (R cuadrado).

Para definir los modelos solamente utilizaremos las variables cuantitativas que más correlación presentan y todas las cualitativas.

```
modelo1 <- lm(net_wins ~ height + age + aces
              + first_won + second_won + bp_faced,
              data = todos_jugadores)

modelo2 <- lm(net_wins ~ height + age + aces
              + hand + country + first_won + second_won + bp_faced,
              data = todos_jugadores)

modelo3 <- lm(net_wins ~ height + age + aces
              + first_won + bp_faced,
              data = todos_jugadores)

modelo4 <- lm(net_wins ~ height + age + hand
              + country + first_won + second_won,
              data = todos_jugadores)

modelo5 <- lm(net_wins ~ height + age + aces
              + hand + country,
              data = todos_jugadores)

modelo6 <- lm(net_wins ~ height + age + aces
              + first_won + second_won,
              data = todos_jugadores)
```

Desarrollados los anteriores modelos, vamos a analizar la bondad del ajuste de los mismos para así compararlos.

```
tabla.coeficientes <- matrix(c(1, summary(modelo1)$r.squared,
                               2, summary(modelo2)$r.squared,
                               3, summary(modelo3)$r.squared,
                               4, summary(modelo4)$r.squared,
                               5, summary(modelo5)$r.squared,
                               6, summary(modelo6)$r.squared),
```

```
                              ncol = 2, byrow = TRUE)

colnames(tabla.coeficientes) <- c("Modelo", "R^2")
tabla.coeficientes
```

```
##      Modelo        R^2
## [1,]      1 0.31161165
## [2,]      2 0.55070470
## [3,]      3 0.31157595
## [4,]      4 0.41593974
## [5,]      5 0.41867836
## [6,]      6 0.05091288
```

Podemos concluir que el segundo modelo es el más conveniente dado que tiene una mayor bondad del ajuste, un mayor coeficiente de determinación (R cuadrado). A continuación, vamos a aplicar este modelo para comprobar como nos serviría para predecir las victorias de un jugador, dadas unas determinadas características.

```
prediccion1 <- data.frame(height = 188,
                          age =  33,
                          aces = 6,
                          hand = "R",
                          country = "SRB",
                          first_won = 37,
                          second_won = 14,
                          bp_faced = 5
)

prediccion2 <- data.frame(height = 199,
                          age = 22,
                          aces = 15,
                          hand = "L",
                          country = "ESP",
                          first_won = 50,
                          second_won = 30,
                          bp_faced = 5
)

predict(modelo2, prediccion1)
```

```
##         1
## 0.3831386
```

```
predict(modelo2, prediccion2)
```

```
##         1
## 0.4619409
```

4.3.3. Contraste de hipótesis.

Como tercera prueba estadística, vamos a llevar a cabo un contraste de hipótesis para responder a la pregunta de si ser zurdo aumenta el número de victorias netas de un jugador (recordemos que se suele decir que es más dificil, siendo diestro jugar contra un jugador zurdo, siendo mayoria los primeros).

Para llevar a cabo el contraste, definiremos dos muestras: en la primera incluiremos las victorias netas de los jugadores zurdos y en la segunda las victorias netas de los jugadores diestros

Debemos aclarar que para este test paramétrico, dado que las muestras son superiores a 30, no es necesario que los datos sean normales. Por esa razón, podemos confirmar que el siguiente contraste de hipótesis es correcto.

```
vic_jug_zurdos <- todos_jugadores$net_wins[todos_jugadores$hand=="L"]
vic_jug_diestros <- todos_jugadores$net_wins[todos_jugadores$hand=="R"]
```

Realizada la anteriore agrupación, planteamos el siguiente contraste de hipótesis, nula y alternativa, de dos muestras sobre la diferencia de medias. Este contrate es unilateral dada la formulación de la hipótesis alternativa:

$$\begin{cases} H_0: & \mu_1 - \mu_2 = 0 \\ H_1: & \mu_1 - \mu_2 < 0 \end{cases}$$

; donde μ_1 es la media de la población de la que se extrae la primera muestra y μ_2 es la media de la población de la que extrae la segunda. Así, tomaremos $\alpha = 0.05$.

```
t.test(vic_jug_diestros, vic_jug_zurdos, alternative = "less")
```

```
##
##  Welch Two Sample t-test
##
## data:  vic_jug_diestros and vic_jug_zurdos
## t = -0.46865, df = 54.503, p-value = 0.3206
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##       -Inf 0.1209041
## sample estimates:
##  mean of x  mean of y
## -0.3094847 -0.2624489
```

Como vemos, obtenemos un p-valor superior al valor de significación fijado. Por esa razón, no podemos rechazar la hipótesis nula ni concluir que el hecho de ser zurdo, te haga ganar más partidos.

5. Representación de los resultados a partir de tablas y gráficas.

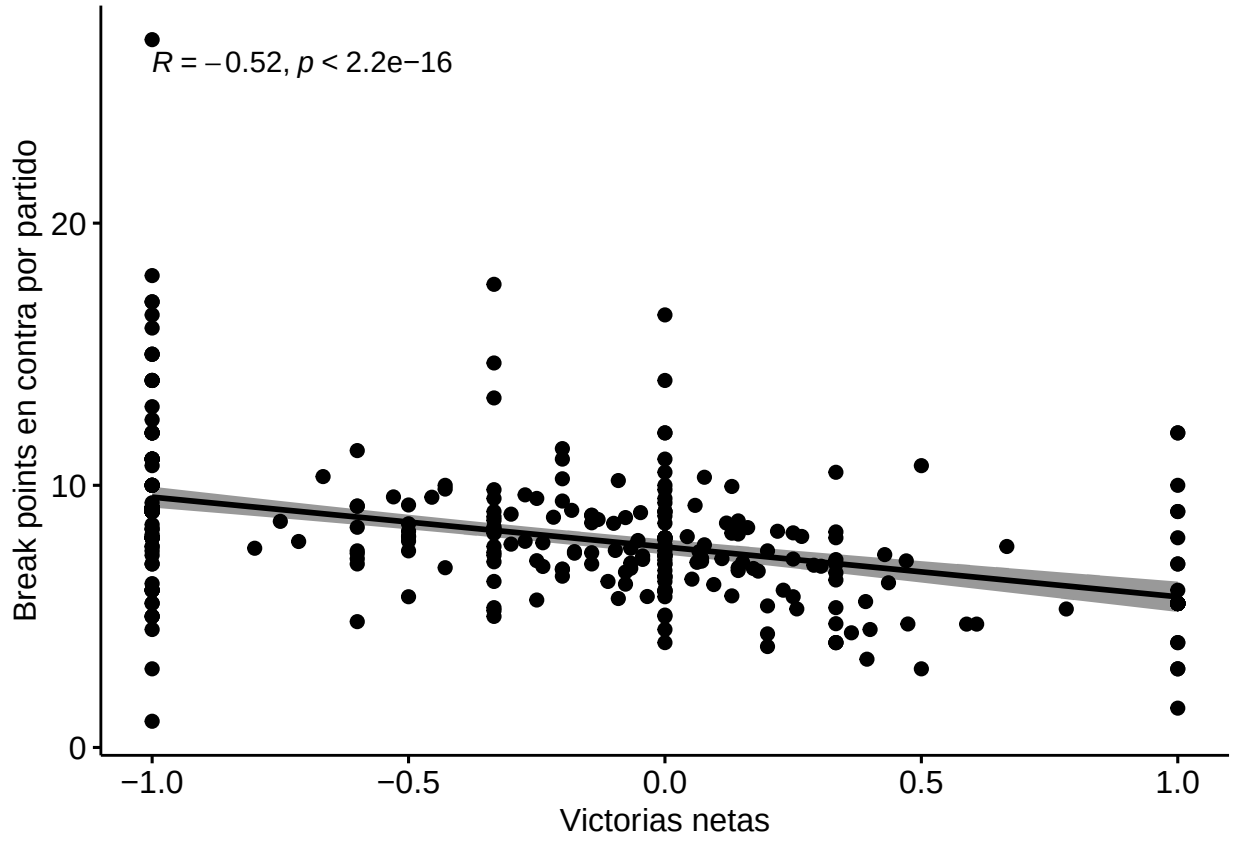
A continuación, a modo de resumen, se presentan una serie de gráficas con información sobre la relación entre las principales variables y las victorias netas:

```
library("ggpubr")
```

```
## Loading required package: ggplot2
```

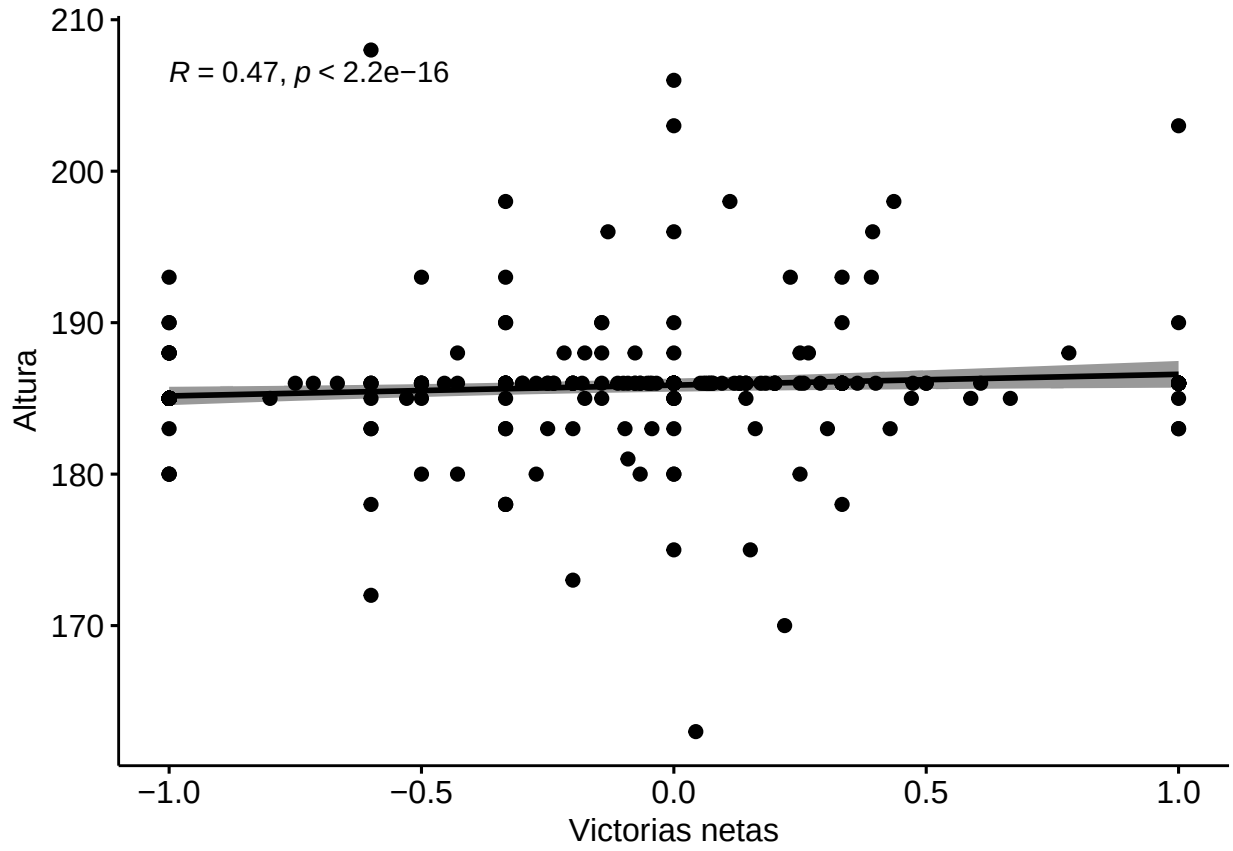
```
# Gráfico de la relación entre los break points en contra
# y el número de victorias netas en el año.
ggscatter(todos_jugadores, x = "net_wins", y = "bp_faced",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "spearman",
          xlab = "Victorias netas",
          ylab = "Break points en contra por partido")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



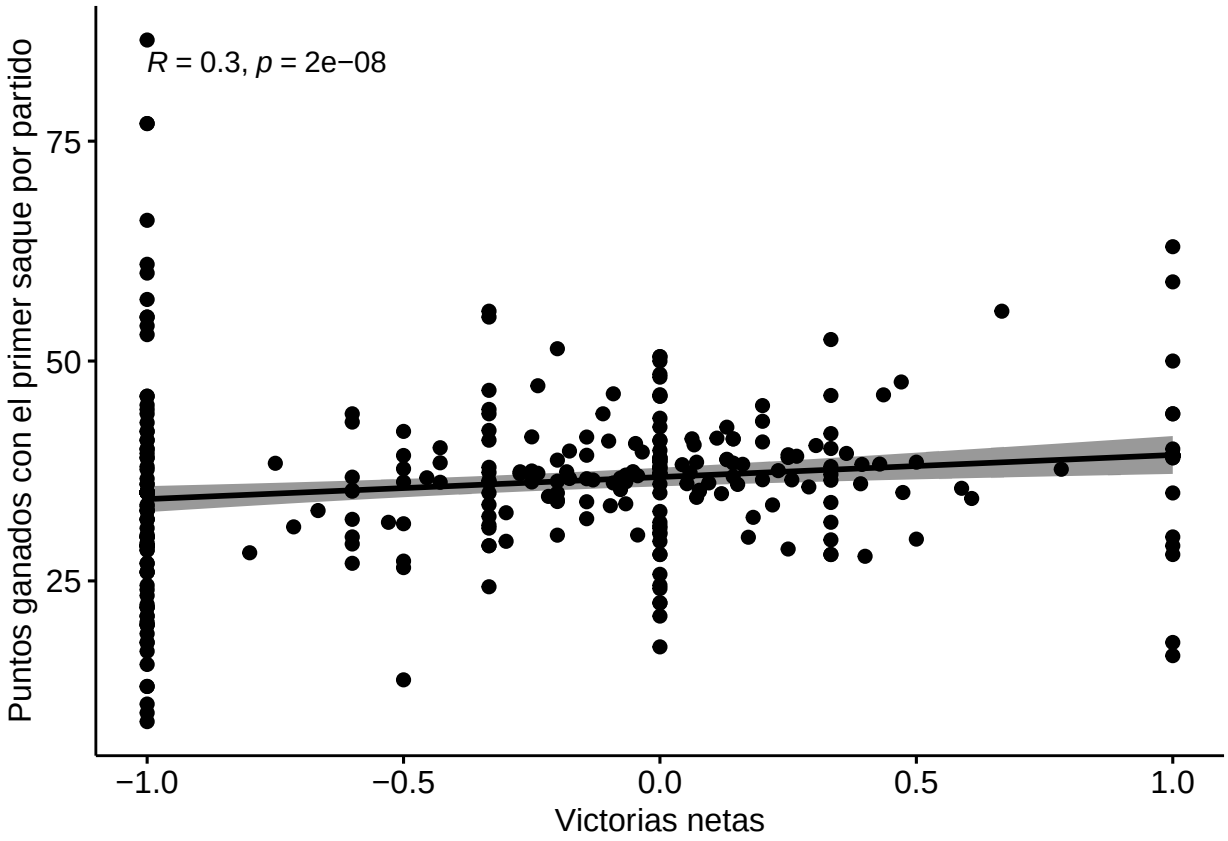

```
# Gráfico de la relación entre la altura
# y el número de victorias netas en el año.
ggscatter(todos_jugadores, x = "net_wins", y = "height",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "spearman",
          xlab = "Victorias netas", ylab = "Altura")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



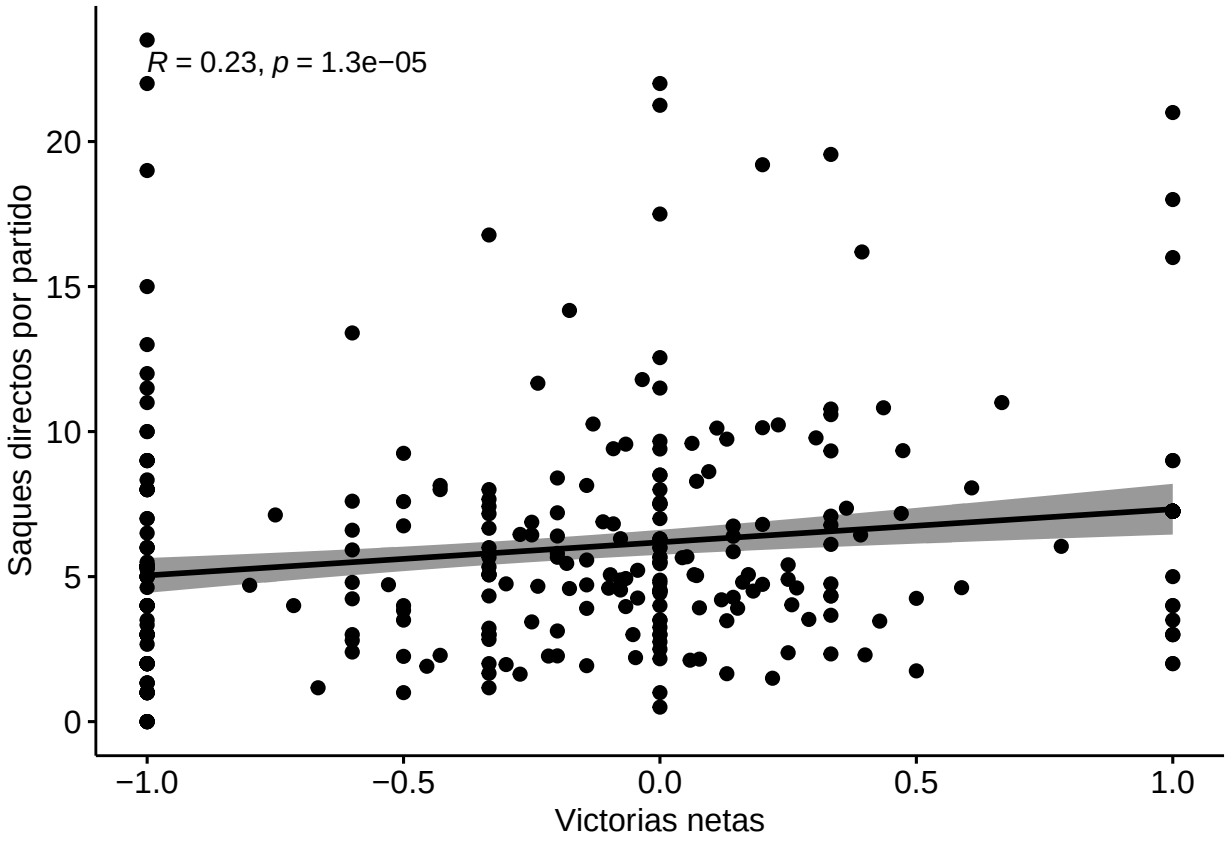

```
# Gráfico de la relación entre los puntos ganados con el primer saque
# y el número de victorias netas en el año.
ggscatter(todos_jugadores, x = "net_wins", y = "first_won",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "spearman",
          xlab = "Victorias netas",
          ylab = "Puntos ganados con el primer saque por partido")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



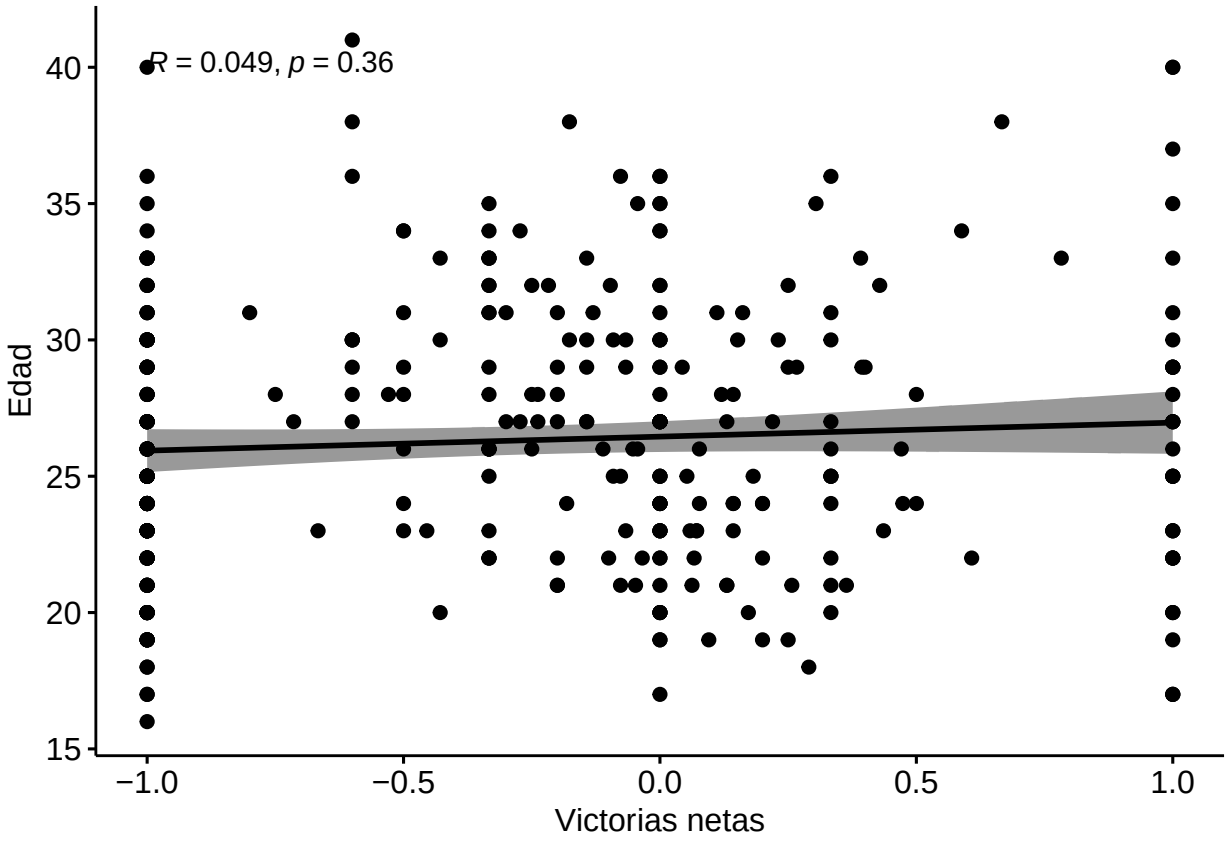

```
# Gráfico de la relación entre el número de saques directos
# y el número de victorias netas en el año.
ggscatter(todos_jugadores, x = "net_wins", y = "aces",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "spearman",
          xlab = "Victorias netas", ylab = "Saques directos por partido")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

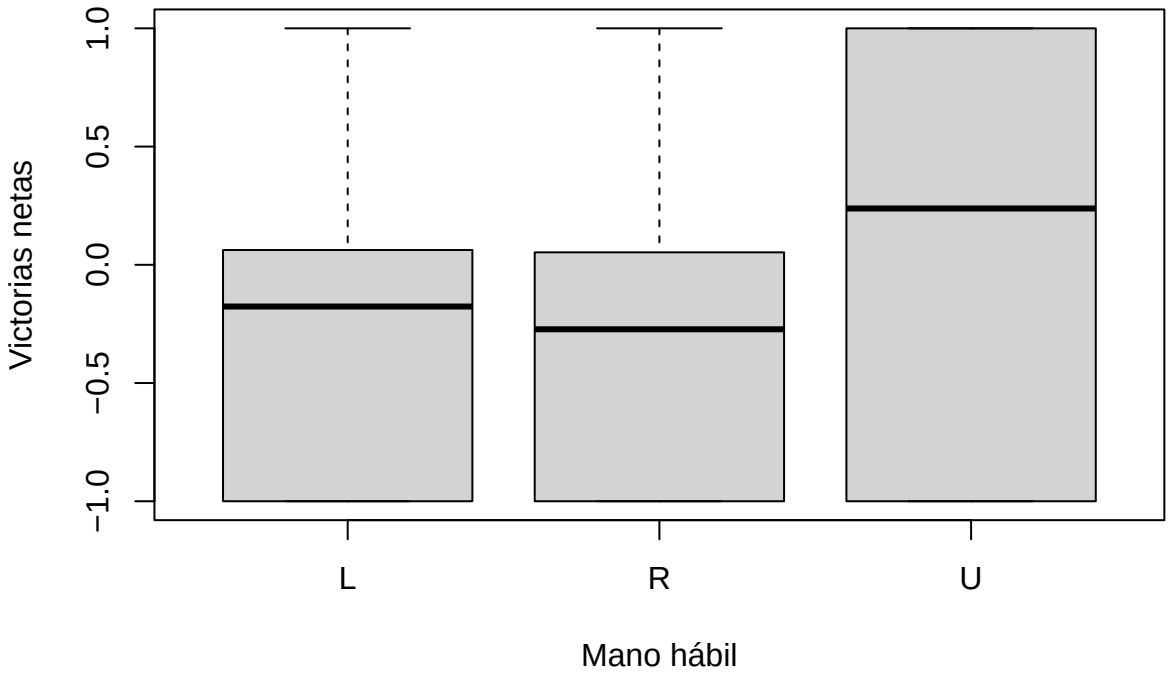



```
# Gráfico de la relación entre la edad
# y el número de victorias netas en el año.
ggscatter(todos_jugadores, x = "net_wins", y = "age",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "spearman",
          xlab = "Victorias netas", ylab = "Edad")
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
# Gráfico de la relación entre la mano hábil
#y el número de victorias netas en el año.
boxplot(todos_jugadores$net_wins ~ todos_jugadores$hand,
        xlab = "Mano hábil", ylab = "Victorias netas")
```



6. Resolución del problema. A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema?

El objetivo de nuestro análisis era intentar responder a la pregunta de qué características de un jugador hacen que este gane más o menos partidos y, a la larga, consiga más títulos. A partir de datos de la temporada 2020, que incluyen estadísticas propias de los jugadores como su altura, nacionalidad o edad, así como otras puramente técnicas como el número de aces, el número de puntos ganados con el primer servicio o los break points a favor y en contra, hemos desarrollado una serie de pruebas para ver como estas variables se relacionan con el número de victorias netas de un jugador (entendido como victorias menos derrotas).

En primer lugar, se ha llevado a cabo una limpieza de los datos a través de la eliminación de valores nulos y la sustitutición de los valores NA. Seguidamente, se ha llevado a cabo un análisis y gestión de los valores extremos de las principales variables cuantitativas.

A continuación, se ha procedido a la creación de un nuevo dataframe, a partir del original, en el que se han desplegado las principales variables por cada uno de los tenistas que han jugado algún partido en la temporada 2020. Por último, se han llevado a cabo una serie de análisis estadísticos para intentar explicar las causas que explican el número de victorias de un tenista.

De todas las variables a nuestro alcance, han destacado cinco por encima de las demás: los puntos de break en contra por partido (inversamente), la altura, el número de puntos ganados con el primer saque por partido y los saques directos por partido. Pordemos ver como algunos componentes como la edad (inversamente) afecta menos de lo que comunmente se piensa. Esto seguramente tiene que ver con varios "viejos" outliers en el mundo del tenis (Djokovic, Federer, Nadal...).

La principal conclusión que podemos obtener del análisis es que, si bien estas variables en su conjunto afectan de una u otra manera al rendimiento de un tenista, lo que es innegable es que hay un componente llamado talento que no puede recoger ninguna prueba estadística.