TOY_1

Paula Moreno Blazquez

Enero 2022

```
library(tidyr)
library(dplyr)
library(chron)                  # Para CHR_to_Time
library(stringr)                # Para str_pad
library(lubridate)
library(reshape)
library(tidyselect)

library(corrplot)
library(RColorBrewer)
library("colorspace")
library(graphics)
library(rpart)

library(ggplot2)

library(car)                    # VIF
```

PARTE 1: Crear Stints

DATA

Se crea una base de datos de juguete random para poder trabajar con un df más pequeño.

```
equipo1 <- c("Juan","Diego","Maria","Andrea","Carla")
equipo2 <- c("Ignasi","Anna","Gerard","Jose","Paula")
equipo3 <- c("Bella","Gus","Alba","Erik","Kevin")
equipo4 <- c("Emma","Mauri","Berta","Judith","Roger")

df <- read.csv(file = './DF_TOY2.csv', header = TRUE, sep = ";")

df_backup <- df

names(df)
```

```
##  [1] "id_play"       "season"        "game_code"     "quarter"
##  [5] "time"          "points_home"   "points_away"   "team_home"
##  [9] "team_away"     "player_home_1" "player_home_2" "player_home_3"
## [13] "player_away_1" "player_away_2" "player_away_3"
```

Modificar variable tiempo

Para facilitar los calculos con el tiempo, se va a pasar los mm:ss a segundos.

- CHR_to_Time: Para pasar la variable tiempo que nos llega como chracter a segundos.
- Print_MS: Que nos devolverá los segundos a formato mm:SS (se hará servir más adelante)

```
CHR_to_Time <- function(x){
  a <- as.POSIXct(x, tz = '', format = "%H:%M:%S", usetz = FALSE)
  tms <- times(format(a, "%H:%M:%S"))
  s <- period_to_seconds(hms(tms))
  return(s)
}

Print_MS <- function(x){
  t <- seconds_to_period(x)
  sprintf('%02d:%02d:%02d', t@hour, minute(t), second(t))
}

df$time <- CHR_to_Time(df$time)
```

Lsita Jugadores-Equipo

```
TP_home <- df %>%
  select(c(contains("team_home"), contains("_home_"))) %>%
  unique()

TP_away <- df %>%
  select(c(contains("team_away"), contains("_away_"))) %>%
  unique()

CBIND_MultipleCol_n <- function(data,col,n){
  d <- unlist(data[col])
  x <- cbind(rep(d, n))
  y <- unlist(data[-1])

  res <- cbind(x, y)
  rownames(res) <- NULL

  res <- unique(res)

  return(res)
}

TeamPlayers_home <- CBIND_MultipleCol_n(TP_home, 1, 3)
TeamPlayers_away <- CBIND_MultipleCol_n(TP_away, 1, 3)

TeamPlayers <- rbind(TeamPlayers_home, TeamPlayers_away) %>%
  as.data.frame() %>%
  `colnames<-`(c("Team", "Player")) %>%
  arrange(Team) %>%
  unique()

Players_Sorted_byTeam <- TeamPlayers$Player
```

Lineups

Se crea variable 'lineups' que recoge los quintetos de ambos equipos en pista.

```r
# Ordenar Lineups para evitar duplicados por DESORDEN
Lineups_PasteSort <- function(x) {
  paste(sort(x), collapse = "-")
}

lineup_home <- df %>% select(contains("_home_"))
lineup_away <- df %>% select(contains("_away_"))
lineups     <- cbind(lineup_home, lineup_away)

lineup_home_sorted <- apply(lineup_home, 1, Lineups_PasteSort)
lineup_away_sorted <- apply(lineup_away, 1, Lineups_PasteSort)
lineups_sorted     <- apply(lineups, 1, Lineups_PasteSort)

#lineups
df_lineups_sorted <- df %>%
  mutate(lineup = lineups_sorted,
    lineup_home = lineup_home_sorted,
    lineup_away = lineup_away_sorted
  ) %>% select(-starts_with("player_"))
```

MERGE Temporada+game_code

Se modifica variable 'game_code' para que quede categorizada con el mismo numero de caracteres. Y unimos 'Season' y 'Game_Code' para tener una variable identificadora del partido.

```r
df_merged <- df_lineups_sorted %>%
  mutate(game_code = paste0("G", str_pad(game_code, 6, pad = "0")),
                     quarter   = paste0("Q", quarter)) %>%
  unite("SeasonGame", c("season", "game_code"))
```

Stints

Queremos obtener un df con los quintetos identificados cada vez que se produce un cambio. Se dejaran aquellos que esten duplicados ya que es necesario diferenciarlos para posteriormente poder hacer el Más/Menos correctamente.

```r
df_stints <- df_merged %>%
  arrange(SeasonGame) %>%
  mutate(StintChanged = (lineup != lag(lineup)),
         StintChanged = replace_na(StintChanged, TRUE),
         stint        = cumsum(StintChanged))

df_reduced <- df_stints %>%
  mutate(StintRemove  = (stint == lead(stint)),
         StintRemove  = replace_na(StintRemove, FALSE)) %>%
  filter(StintRemove != TRUE)
```

Mas/Menos

Obtenemos el Más/Menos segun cada stint. Se tendrá en cuenta el cambio de partido. Además, esta variable estará hecha con HOME como referéncia, pero eso no hace ninguna diferencia estadística importante en nuestro resultado final.

```r
PlusMinus_function <- function(h,a){
  (h-lag(h))-(a-lag(a))
```

```
}

#Home como referencia
df_PlusMinus <- df_reduced %>%
  group_by(SeasonGame) %>%
  mutate(
    stint_time = ifelse(is.na(lag(time)), time, time - lag(time)),
    PlusMinus  = ifelse(is.na(lag(time)), points_home - points_away,
                        PlusMinus_function(points_home, points_away))
  ) %>% ungroup()

df_PlusMinus_reduced <- df_PlusMinus %>%
  select(c(SeasonGame, quarter, lineup, lineup_home, lineup_away, stint_time, PlusMinus))

#Eliminar stints duplicados
df_PlusMinus_reduced_bylineups <- df_PlusMinus_reduced %>%
  group_by(SeasonGame, quarter, lineup, lineup_home, lineup_away) %>%
  summarise(
    stint_time = sum(stint_time),
    PlusMinus  = sum(PlusMinus) ) %>%
  ungroup() %>%
  as.data.frame()
```

PARTE 2: Dummys Jugadores

```
# Vector con todos los nombres de los jugadores:
players <- c(equipo1, equipo2, equipo3, equipo4)
length(players)
```

```
## [1] 20
```

```
df_dummys_H <- fastDummies::dummy_cols(df_PlusMinus_reduced_bylineups,
                                       select_columns = "lineup_home",
                                       split = "-")
df_dummys_A <- fastDummies::dummy_cols(df_PlusMinus_reduced_bylineups,
                                       select_columns = "lineup_away",
                                       split = "-") %>%
                  mutate(across(starts_with("lineup_away_"), function(x) -x))

Remove_firsts_chars_colnames <- function(data, char){
  num_char <- nchar(char)+1
  substring(names(data), num_char)
}

COL_From <- function(data, first_col){
  last_col = ncol(data)
  colnames(data[first_col:last_col])
}

COL_to <- function(data, first_col, char){
  last_col = ncol(data)
  Remove_firsts_chars_colnames(data[first_col:last_col], char)
}
```

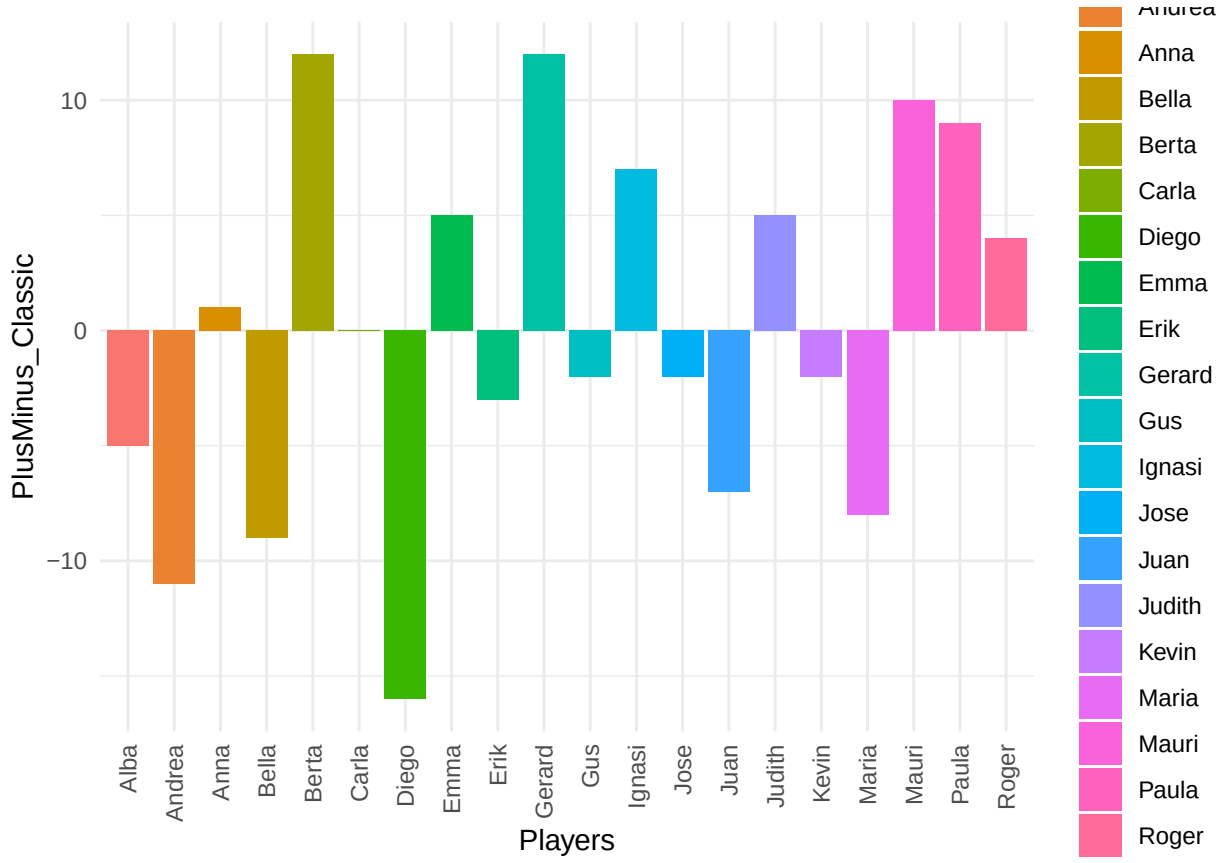
```
#Primera columna con el nombre de un jugador
match <- match(paste0("lineup_away_", players), names(df_dummys_A)) %>%
            na.omit() %>%
            min()

#nombre columnas sin primeras palabras:
col_to_A <- COL_to(df_dummys_A, match, "lineup_away_")
col_to_H <- COL_to(df_dummys_H, match, "lineup_home_")

df_dummys_A <- df_dummys_A %>% rename_at(vars(COL_From(df_dummys_A,match)), function(x) col_to_A)
df_dummys_H <- df_dummys_H %>% rename_at(vars(COL_From(df_dummys_H,match)), function(x) col_to_H)

#Juntar los dos DF:
df_dummys_A[df_dummys_A == 0] <- NA
df_dummys_H[df_dummys_H == 0] <- NA

df_dummys <- coalesce(df_dummys_H,df_dummys_A)
df_dummys[is.na(df_dummys)] <- 0

df_dummys <- df_dummys %>% select(-c(starts_with("lineup_"))) %>% ungroup()
```

Ahora mismo tenemos un DF con el PlusMinus con HOME como referencia (si es positivo, ganaban HOME. Si es Negativo ganaban AWAY). Luego tenemos variables "dummys" con 1 si estaban jugando como HOME, -1 si estaban jugando como AWAY y 0 si no estaban en pista.

PARTE 3: PlusMinus por stint (PlusMinus CLASSIC)

```
df_dummys_PlusMinus <- df_dummys %>%
  group_by(SeasonGame, quarter, lineup, stint_time) %>%
  mutate(across(matches(players), function(x) x*PlusMinus)) %>%
  select(-c(PlusMinus)) %>%
  ungroup()

### MasMenos de los mismos lineups (sin tener en cuenta SeasonGame o Quarter):

df_dummys_PlusMinus_2 <- df_dummys_PlusMinus %>% select(-c(SeasonGame, quarter))

PlusMinus_Lineups <- aggregate(. ~ lineup, df_dummys_PlusMinus_2, sum, na.rm = TRUE) %>%
                     mutate(stint_time = Print_MS(stint_time))

PlusMinus_Classic <- colSums(PlusMinus_Lineups[3:ncol(PlusMinus_Lineups)])

PlusMinus_Classic_df <- as.data.frame(PlusMinus_Classic)

PlusMinus_Classic_df <- PlusMinus_Classic_df %>%
  mutate(Players = rownames(PlusMinus_Classic_df)) %>%
  arrange(Players) %>%
  select(Players, PlusMinus_Classic)

rownames(PlusMinus_Classic_df) <- NULL
```

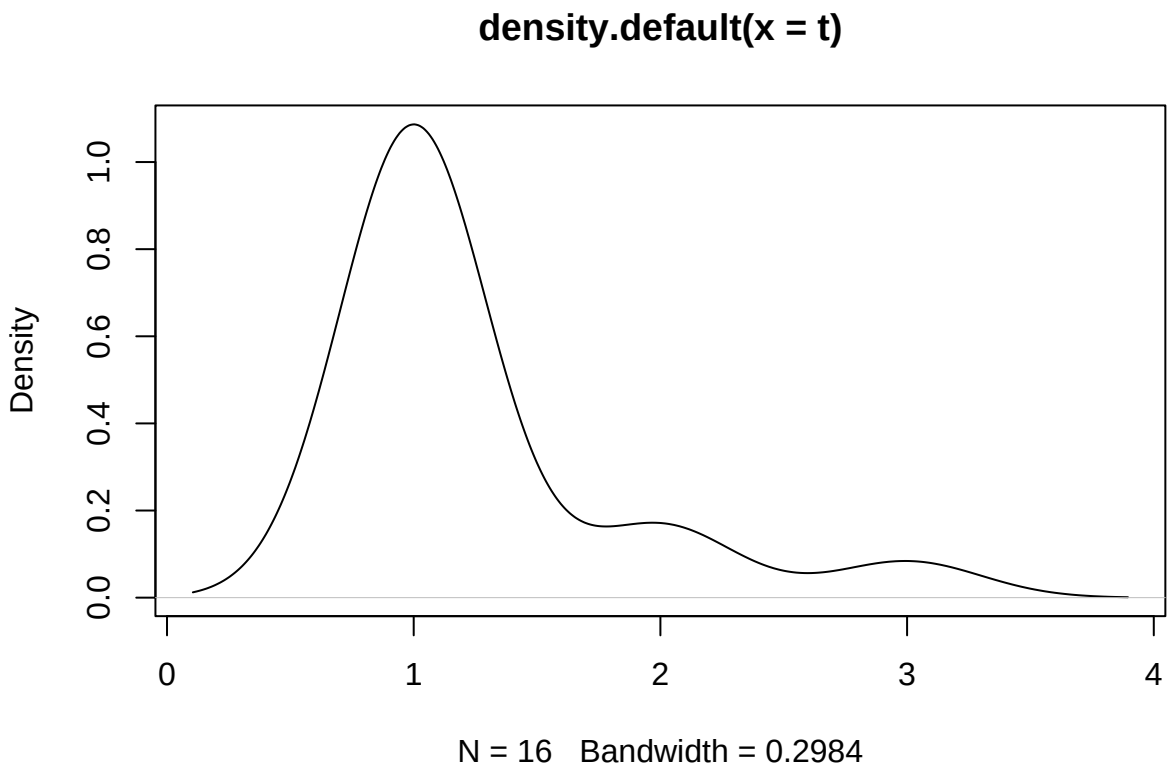
```
ggplot(PlusMinus_Classic_df, aes(x=Players, y=PlusMinus_Classic, fill=Players)) +
  geom_bar(stat="identity") + theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



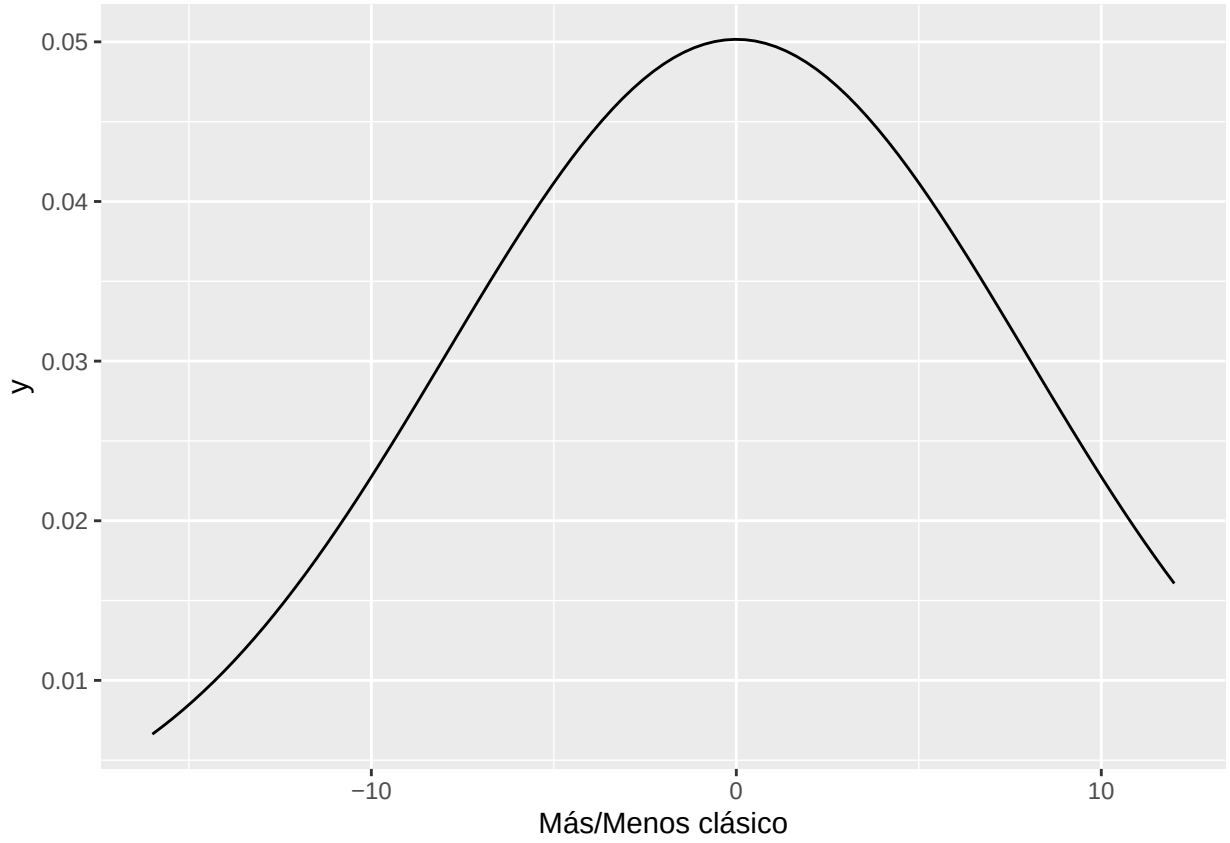

```
#util para grupos pequeños de jugadores, por ejemplo equipos o maximos

t<-table(PlusMinus_Classic); t
```

```
## PlusMinus_Classic
## -16 -11  -9  -8  -7  -5  -3  -2   0   1   4   5   7   9  10  12
##   1   1   1   1   1   1   1   3   1   1   1   2   1   1   1   2
```

```
plot(density(t))
```




```
ggplot(PlusMinus_Classic_df, aes(x = PlusMinus_Classic)) +
  stat_function(
    fun = dnorm,
    args = with(PlusMinus_Classic_df, c(mean = mean(PlusMinus_Classic),
                                        sd = sd(PlusMinus_Classic)))
  ) + scale_x_continuous("Más/Menos clásico")
```



PARTE 4: Modelar

Variable outcome: PlusMinus

```
summary(df_dummys)
```

```
##   SeasonGame          quarter             lineup            stint_time
##  Length:21          Length:21          Length:21          Min.   : 51.0
##  Class :character   Class :character   Class :character   1st Qu.:128.0
##  Mode  :character   Mode  :character   Mode  :character   Median :239.0
##                                                           Mean   :328.4
##                                                           3rd Qu.:411.0
##                                                           Max.   :960.0
##    PlusMinus           Andrea           Carla             Juan
##  Min.   :-4.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:-2.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median : 0.0000   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :-0.3333   Mean   :0.4762   Mean   :0.2857   Mean   :0.3333
##  3rd Qu.: 1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   : 3.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##      Maria            Diego            Anna        Gerard
##  Min.   :0.0000   Min.   :0.0000   Min.   :-1   Min.   :-1.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 0   1st Qu.: 0.0000
##  Median :0.0000   Median :0.0000   Median : 0   Median : 0.0000
##  Mean   :0.4286   Mean   :0.3333   Mean   : 0   Mean   : 0.1429
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.: 0   3rd Qu.: 1.0000
```

```
##  Max.   :1.0000   Max.   :1.0000   Max.   : 1   Max.   : 1.0000
##       Jose              Paula            Ignasi              Alba
##  Min.   :-1.00000   Min.   :0.0000   Min.   :-1.00000   Min.   :-1.0000
##  1st Qu.: 0.00000   1st Qu.:0.0000   1st Qu.: 0.00000   1st Qu.:-1.0000
##  Median : 0.00000   Median :0.0000   Median : 0.00000   Median : 0.0000
##  Mean   : 0.09524   Mean   :0.2381   Mean   : 0.09524   Mean   :-0.2857
##  3rd Qu.: 0.00000   3rd Qu.:0.0000   3rd Qu.: 1.00000   3rd Qu.: 0.0000
##  Max.   : 1.00000   Max.   :1.0000   Max.   : 1.00000   Max.   : 0.0000
##      Bella              Erik              Gus              Kevin
##  Min.   :-1.0000   Min.   :-1.0000   Min.   :-1.0000   Min.   :-1.0000
##  1st Qu.:-1.0000   1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: 0.0000
##  Median : 0.0000   Median : 0.0000   Median : 0.0000   Median : 0.0000
##  Mean   :-0.2857   Mean   :-0.1905   Mean   :-0.2381   Mean   :-0.1429
##  3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000
##  Max.   : 0.0000   Max.   : 0.0000   Max.   : 0.0000   Max.   : 0.0000
##      Berta              Emma              Judith            Mauri
##  Min.   :-1.0000   Min.   :-1.0000   Min.   :-1.0000   Min.   :-1.0000
##  1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.: 0.0000
##  Median : 0.0000   Median : 0.0000   Median : 0.0000   Median : 0.0000
##  Mean   :-0.3333   Mean   :-0.2857   Mean   :-0.2857   Mean   :-0.2381
##  3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000
##  Max.   : 0.0000   Max.   : 0.0000   Max.   : 0.0000   Max.   : 0.0000
##      Roger
##  Min.   :-1.0000
##  1st Qu.: 0.0000
##  Median : 0.0000
##  Mean   :-0.1429
##  3rd Qu.: 0.0000
##  Max.   : 0.0000
```

```
mod1 <- lm(PlusMinus ~ . -SeasonGame -quarter -lineup -stint_time, data=df_dummys)
summary(mod1)
```

```
##
## Call:
## lm(formula = PlusMinus ~ . - SeasonGame - quarter - lineup -
##     stint_time, data = df_dummys)
##
## Residuals:
##          1          2          3          4          5          6          7
## -1.022e+00  4.783e-01  5.435e-01 -1.082e-15 -8.604e-16  2.692e-15 -8.327e-17
##          8          9         10         11         12         13         14
## -1.943e-16 -1.943e-16 -2.304e-15  1.138e-15 -1.082e-15 -5.218e-15  1.087e+00
##         15         16         17         18         19         20         21
## -5.435e-01 -5.435e-01 -1.630e+00  5.435e-01  6.522e-02 -5.435e-01  1.565e+00
##
## Coefficients: (3 not defined because of singularities)
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.565e+00  8.577e+00   0.532    0.631
## Andrea      -2.870e+00  4.815e+00  -0.596    0.593
## Carla       -9.348e-01  5.324e+00  -0.176    0.872
## Juan        -1.761e+00  5.067e+00  -0.347    0.751
## Maria       -9.348e-01  4.506e+00  -0.207    0.849
## Diego       -5.261e+00  4.318e+00  -1.218    0.310
## Anna        -1.674e+00  2.420e+00  -0.692    0.539
```

```
## Gerard       3.326e+00  3.728e+00   0.892    0.438
## Jose        -2.217e+00  3.422e+00  -0.648    0.563
## Paula       -2.174e-01  4.638e+00  -0.047    0.966
## Ignasi      -3.674e+00  3.103e+00  -1.184    0.322
## Alba                NA         NA      NA       NA
## Bella        1.000e+00  3.362e+00   0.297    0.786
## Erik         3.000e+00  2.297e+00   1.306    0.283
## Gus          1.945e-15  2.297e+00   0.000    1.000
## Kevin               NA         NA      NA       NA
## Berta       -3.696e-01  2.804e+00  -0.132    0.903
## Emma        -8.696e-01  3.727e+00  -0.233    0.831
## Judith      -1.761e+00  2.709e+00  -0.650    0.562
## Mauri        1.696e+00  1.619e+00   1.047    0.372
## Roger               NA         NA      NA       NA
##
## Residual standard error: 1.736 on 3 degrees of freedom
## Multiple R-squared:  0.9152, Adjusted R-squared:  0.4348
## F-statistic: 1.905 on 17 and 3 DF,  p-value: 0.329
```

```
mod2 <- lm(PlusMinus/stint_time ~ . -SeasonGame -quarter -lineup, data=df_dummys)
summary(mod2)
```

```
##
## Call:
## lm(formula = PlusMinus/stint_time ~ . - SeasonGame - quarter -
##     lineup, data = df_dummys)
##
## Residuals:
##          1          2          3          4          5          6          7
## -6.944e-03  5.757e-03  1.187e-03 -1.106e-17  6.200e-03  2.017e-17 -6.200e-03
##          8          9         10         11         12         13         14
##  6.200e-03  1.084e-18 -6.200e-03  1.084e-18 -9.324e-18 -3.665e-17  2.374e-03
##         15         16         17         18         19         20         21
## -1.187e-03 -1.187e-03 -3.560e-03  1.187e-03 -4.570e-03 -1.187e-03  8.131e-03
##
## Coefficients: (3 not defined because of singularities)
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0505262  0.0531637   0.950    0.412
## Andrea      -0.0470915  0.0298484  -1.578    0.213
## Carla       -0.0404379  0.0330001  -1.225    0.308
## Juan        -0.0365595  0.0314103  -1.164    0.329
## Maria       -0.0095904  0.0279310  -0.343    0.754
## Diego       -0.0513439  0.0267647  -1.918    0.151
## Anna        -0.0148069  0.0150025  -0.987    0.396
## Gerard       0.0088423  0.0231072   0.383    0.727
## Jose        -0.0159937  0.0212134  -0.754    0.506
## Paula        0.0072254  0.0287479   0.251    0.818
## Ignasi      -0.0436946  0.0192316  -2.272    0.108
## Alba                NA         NA      NA       NA
## Bella        0.0052489  0.0208409   0.252    0.817
## Erik         0.0176504  0.0142370   1.240    0.303
## Gus         -0.0005314  0.0142370  -0.037    0.973
## Kevin               NA         NA      NA       NA
## Berta       -0.0278989  0.0173825  -1.605    0.207
## Emma        -0.0241961  0.0231041  -1.047    0.372
```

```
## Judith      -0.0285099  0.0167931  -1.698    0.188
## Mauri        0.0165612  0.0100358   1.650    0.197
## Roger               NA         NA      NA       NA
##
## Residual standard error: 0.01076 on 3 degrees of freedom
## Multiple R-squared:   0.95,  Adjusted R-squared:  0.6669
## F-statistic: 3.356 on 17 and 3 DF,  p-value: 0.1736
```

```
mod3 <- lm(PlusMinus*stint_time ~ . -SeasonGame -quarter -lineup, data=df_dummys)
summary(mod3)
```

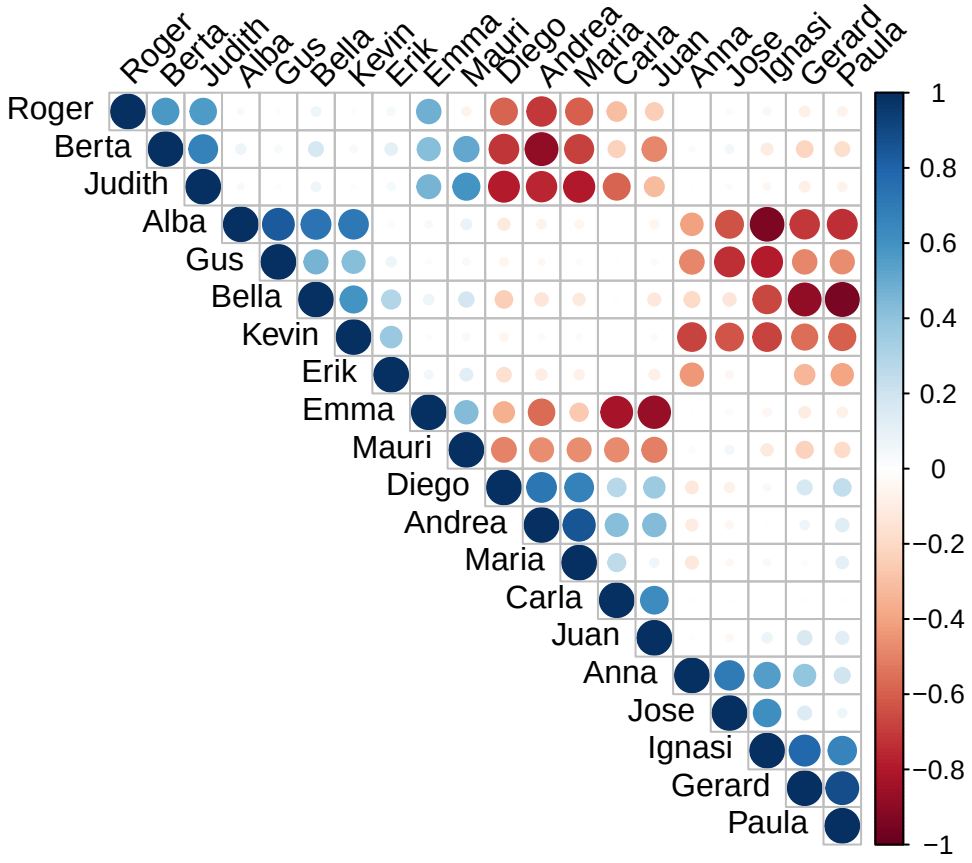
```
##
## Call:
## lm(formula = PlusMinus * stint_time ~ . - SeasonGame - quarter -
##     lineup, data = df_dummys)
##
## Residuals:
##          1          2          3          4          5          6          7
## -6.370e+00 -5.114e+02  5.177e+02  2.898e-13  4.700e+01 -6.766e-13 -4.700e+01
##          8          9         10         11         12         13         14
##  4.700e+01 -1.089e-12 -4.700e+01 -3.924e-13  3.182e-13  3.466e-13  1.035e+03
##         15         16         17         18         19         20         21
## -5.177e+02 -5.177e+02 -1.553e+03  5.177e+02  1.029e+03 -5.177e+02  5.241e+02
##
## Coefficients: (3 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2623.1     7231.5   0.363    0.741
## Andrea       -1422.4     4060.1  -0.350    0.749
## Carla         -531.6     4488.8  -0.118    0.913
## Juan          -730.1     4272.5  -0.171    0.875
## Maria         -907.6     3799.3  -0.239    0.827
## Diego        -2145.1     3640.6  -0.589    0.597
## Anna          -694.8     2040.7  -0.340    0.756
## Gerard        1013.2     3143.1   0.322    0.768
## Jose         -1212.5     2885.5  -0.420    0.703
## Paula        -1530.5     3910.4  -0.391    0.722
## Ignasi        -761.8     2616.0  -0.291    0.790
## Alba              NA         NA      NA       NA
## Bella          511.0     2834.8   0.180    0.868
## Erik           833.0     1936.6   0.430    0.696
## Gus            338.0     1936.6   0.175    0.873
## Kevin             NA         NA      NA       NA
## Berta          324.7     2364.4   0.137    0.899
## Emma          -663.8     3142.7  -0.211    0.846
## Judith        -289.9     2284.3  -0.127    0.907
## Mauri         -137.7     1365.1  -0.101    0.926
## Roger             NA         NA      NA       NA
##
## Residual standard error: 1464 on 3 degrees of freedom
## Multiple R-squared:  0.6722, Adjusted R-squared:  -1.185
## F-statistic: 0.3619 on 17 and 3 DF,  p-value: 0.9261
```

PARTE 5: HeatMap Correlaciones Jugadores

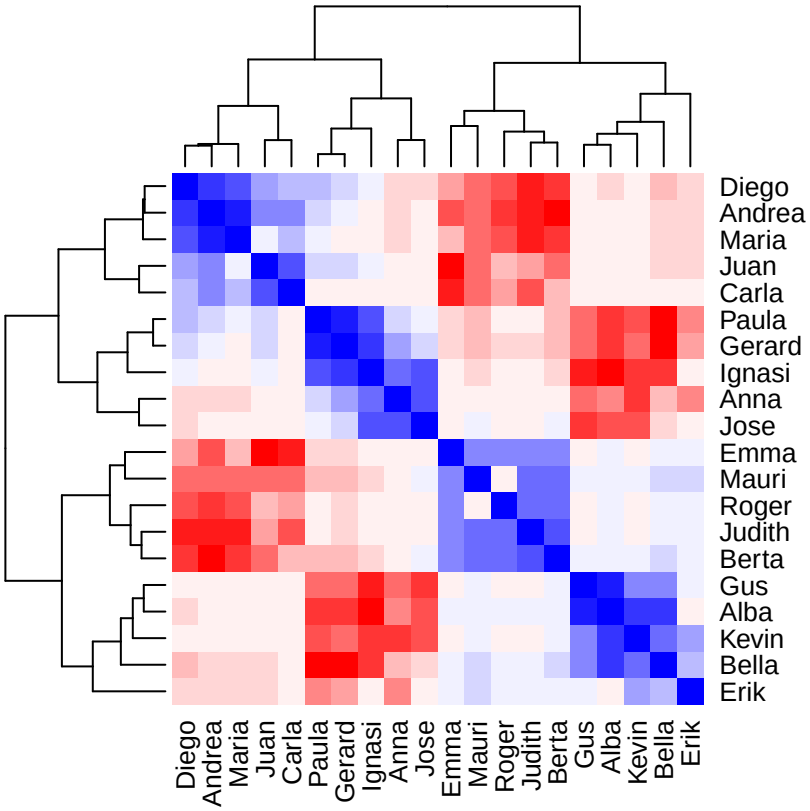
Primero vamos a preparar los datos con la estructura que necesitamos para crear el heatmap:

```
match_firstplayer <- match(players, names(df_dummys_PlusMinus)) %>%
            na.omit() %>%
            min()
n <- dim(df_dummys_PlusMinus)[2]

correlations <- cor(model.matrix(~.-1, data=df_dummys_PlusMinus[,match_firstplayer:n]))
```

```
corrplot(correlations, type = "upper", order = "hclust",
         tl.col = "black", tl.srt = 45)
```




```
col<- colorRampPalette(c("red", "white", "blue"))(20)
heatmap(x = correlations, col = col, symm = TRUE)
```



PARTE 6: Variance Inflation Factor (VIF)

```
vif()
```

Tenemos pocos casos. Y por eso, tenemos dos o más variables predictoras en el modelo que están altamente (o perfectamente) correlacionadas.