TOY_1

Paula Moreno Blazquez

Enero 2022

PARTE 1: Crear Stints

DATA

```
players_L <- paste0("LP",1:10)
players_A <- paste0("AP",1:10)


toy <- data.frame(
  "id_play" = 1:15,
  "season" = c(rep("S2017",15)),
  "game" = c(rep("G1",10), rep("G2",5)),
  "points_L" = c(0,0,0,2,2,2,5,5,7,7,7,7,10,10,12),
  "points_A" = c(0,2,2,2,4,4,7,7,9,9,10,10,12,12,12),
  "LP1" = c(rep(players_L[1],4),rep(players_L[6],5),rep(players_L[1],6)),
  "LP2" = players_L[2],
  "LP3" = c(rep(players_L[3],7), rep(players_L[7],2),rep(players_L[3],6)),
  "AP1" = players_A[1],
  "AP2" = c(rep(players_A[2],8),players_A[7], rep(players_A[2],6)),
  "AP3" = players_A[3]
  )

toy2 <- data.frame(                          #DF2 para comprovar que funciona
  "id_play" = 1:20,
  "season" = c(rep("S2017",20)),
  "game" = c(rep("G1",10), rep("G2",10)),
  "points_L" = c(0,0,0,2,2,2,5,5,7,7,0,2,2,5,5,7,7,7,9,10),
  "points_A" = c(0,2,2,2,4,4,7,7,9,9,0,0,2,2,4,4,6,6,6,6),
  "LP1" = c(rep(players_L[1],4),rep(players_L[6],5),rep(players_L[1],8), rep(players_L[1],3)),
  "LP2" = players_L[2],
  "LP3" = c(rep(players_L[3],7), rep(players_L[7],2),rep(players_L[3],9), rep(players_L[9],2)),
  "AP1" = players_A[1],
  "AP2" = c(rep(players_A[2],8),players_A[7], rep(players_A[2],9), rep(players_A[5],2)),
  "AP3" = players_A[3]
  )


toy_backup <- toy

df <- toy2

df$PM <- df$points_L-df$points_A
```

```
knitr::kable(df)
```

id_play	season	game	points_L	points_A	LP1	LP2	LP3	AP1	AP2	AP3	PM
1	S2017	G1	0	0	LP1	LP2	LP3	AP1	AP2	AP3	0
2	S2017	G1	0	2	LP1	LP2	LP3	AP1	AP2	AP3	-2
3	S2017	G1	0	2	LP1	LP2	LP3	AP1	AP2	AP3	-2
4	S2017	G1	2	2	LP1	LP2	LP3	AP1	AP2	AP3	0
5	S2017	G1	2	4	LP6	LP2	LP3	AP1	AP2	AP3	-2
6	S2017	G1	2	4	LP6	LP2	LP3	AP1	AP2	AP3	-2
7	S2017	G1	5	7	LP6	LP2	LP3	AP1	AP2	AP3	-2
8	S2017	G1	5	7	LP6	LP2	LP7	AP1	AP2	AP3	-2
9	S2017	G1	7	9	LP6	LP2	LP7	AP1	AP7	AP3	-2
10	S2017	G1	7	9	LP1	LP2	LP3	AP1	AP2	AP3	-2
11	S2017	G2	0	0	LP1	LP2	LP3	AP1	AP2	AP3	0
12	S2017	G2	2	0	LP1	LP2	LP3	AP1	AP2	AP3	2
13	S2017	G2	2	2	LP1	LP2	LP3	AP1	AP2	AP3	0
14	S2017	G2	5	2	LP1	LP2	LP3	AP1	AP2	AP3	3
15	S2017	G2	5	4	LP1	LP2	LP3	AP1	AP2	AP3	1
16	S2017	G2	7	4	LP1	LP2	LP3	AP1	AP2	AP3	3
17	S2017	G2	7	6	LP1	LP2	LP3	AP1	AP2	AP3	1
18	S2017	G2	7	6	LP1	LP2	LP3	AP1	AP2	AP3	1
19	S2017	G2	9	6	LP1	LP2	LP9	AP1	AP5	AP3	3
20	S2017	G2	10	6	LP1	LP2	LP9	AP1	AP5	AP3	4

ORDENAR DF

```
players_pbp <- df[6:11] #subset de solo jugadores pbp
resto_pbp <- cbind(df[1:5], df[12])

## Ordenar alfabeticamente jugadores
order_players_pbp <- lapply(1:nrow(players_pbp), function(row)
                     players_pbp[row, order(players_pbp[row, ], decreasing = TRUE)])
df_order_players_pbp <- data.frame(matrix(unlist(order_players_pbp),
                        nrow=length(order_players_pbp), byrow=TRUE))
names(df_order_players_pbp) <- c(paste0("LP",1:3), paste0("AP",1:3))

df_new <- cbind(resto_pbp, df_order_players_pbp)

## Ordernar Columnas
col_order <- c("season", "game", "id_play", "points_L", "points_A", "PM",
               paste0("LP",1:3), paste0("AP",1:3))
df_order <- df_new[, col_order]
knitr::kable(df_order)
```

season	game	id_play	points_L	points_A	PM	LP1	LP2	LP3	AP1	AP2	AP3
S2017	G1	1	0	0	0	LP3	LP2	LP1	AP3	AP2	AP1
S2017	G1	2	0	2	-2	LP3	LP2	LP1	AP3	AP2	AP1
S2017	G1	3	0	2	-2	LP3	LP2	LP1	AP3	AP2	AP1
S2017	G1	4	2	2	0	LP3	LP2	LP1	AP3	AP2	AP1
S2017	G1	5	2	4	-2	LP6	LP3	LP2	AP3	AP2	AP1
S2017	G1	6	2	4	-2	LP6	LP3	LP2	AP3	AP2	AP1

season	game	id_play	points_L	points_A	PM	LP1	LP2	LP3	AP1	AP2	AP3
S2017	G1	7	5	7	-2	LP6	LP3	LP2	AP3	AP2	AP1
S2017	G1	8	5	7	-2	LP7	LP6	LP2	AP3	AP2	AP1
S2017	G1	9	7	9	-2	LP7	LP6	LP2	AP7	AP3	AP1
S2017	G1	10	7	9	-2	LP3	LP2	LP1	AP3	AP2	AP1
S2017	G2	11	0	0	0	LP3	LP2	LP1	AP3	AP2	AP1
S2017	G2	12	2	0	2	LP3	LP2	LP1	AP3	AP2	AP1
S2017	G2	13	2	2	0	LP3	LP2	LP1	AP3	AP2	AP1
S2017	G2	14	5	2	3	LP3	LP2	LP1	AP3	AP2	AP1
S2017	G2	15	5	4	1	LP3	LP2	LP1	AP3	AP2	AP1
S2017	G2	16	7	4	3	LP3	LP2	LP1	AP3	AP2	AP1
S2017	G2	17	7	6	1	LP3	LP2	LP1	AP3	AP2	AP1
S2017	G2	18	7	6	1	LP3	LP2	LP1	AP3	AP2	AP1
S2017	G2	19	9	6	3	LP9	LP2	LP1	AP5	AP3	AP1
S2017	G2	20	10	6	4	LP9	LP2	LP1	AP5	AP3	AP1

MERGE Jugadores y Temporada+Game

```
library(tidyr)
df_order <- df_order %>%
  unite("Merged_Players", LP1:AP3, remove = TRUE) %>%
  unite("Merged_SG", c("season", "game"))

knitr::kable(df_order)
```

Merged_SG	id_play	points_L	points_A	PM	Merged_Players
S2017_G1	1	0	0	0	LP3_LP2_LP1_AP3_AP2_AP1
S2017_G1	2	0	2	-2	LP3_LP2_LP1_AP3_AP2_AP1
S2017_G1	3	0	2	-2	LP3_LP2_LP1_AP3_AP2_AP1
S2017_G1	4	2	2	0	LP3_LP2_LP1_AP3_AP2_AP1
S2017_G1	5	2	4	-2	LP6_LP3_LP2_AP3_AP2_AP1
S2017_G1	6	2	4	-2	LP6_LP3_LP2_AP3_AP2_AP1
S2017_G1	7	5	7	-2	LP6_LP3_LP2_AP3_AP2_AP1
S2017_G1	8	5	7	-2	LP7_LP6_LP2_AP3_AP2_AP1
S2017_G1	9	7	9	-2	LP7_LP6_LP2_AP7_AP3_AP1
S2017_G1	10	7	9	-2	LP3_LP2_LP1_AP3_AP2_AP1
S2017_G2	11	0	0	0	LP3_LP2_LP1_AP3_AP2_AP1
S2017_G2	12	2	0	2	LP3_LP2_LP1_AP3_AP2_AP1
S2017_G2	13	2	2	0	LP3_LP2_LP1_AP3_AP2_AP1
S2017_G2	14	5	2	3	LP3_LP2_LP1_AP3_AP2_AP1
S2017_G2	15	5	4	1	LP3_LP2_LP1_AP3_AP2_AP1
S2017_G2	16	7	4	3	LP3_LP2_LP1_AP3_AP2_AP1
S2017_G2	17	7	6	1	LP3_LP2_LP1_AP3_AP2_AP1
S2017_G2	18	7	6	1	LP3_LP2_LP1_AP3_AP2_AP1
S2017_G2	19	9	6	3	LP9_LP2_LP1_AP5_AP3_AP1
S2017_G2	20	10	6	4	LP9_LP2_LP1_AP5_AP3_AP1

UNIQUE quintetos

```
library(dplyr)
players_Merged <- df_order$Merged_Players == lag(df_order$Merged_Players)
```

```
                                          ## col_TrueFalse si son igual que lag row
SG_DIF <- df_order$Merged_SG == lag(df_order$Merged_SG)

df_order$TF_Cambios <- ifelse(((players_Merged == FALSE)|(SG_DIF == FALSE )), "C", "NC")

df_order
```

```
##    Merged_SG id_play points_L points_A PM          Merged_Players TF_Cambios
## 1   S2017_G1       1        0        0  0 LP3_LP2_LP1_AP3_AP2_AP1       <NA>
## 2   S2017_G1       2        0        2 -2 LP3_LP2_LP1_AP3_AP2_AP1         NC
## 3   S2017_G1       3        0        2 -2 LP3_LP2_LP1_AP3_AP2_AP1         NC
## 4   S2017_G1       4        2        2  0 LP3_LP2_LP1_AP3_AP2_AP1         NC
## 5   S2017_G1       5        2        4 -2 LP6_LP3_LP2_AP3_AP2_AP1          C
## 6   S2017_G1       6        2        4 -2 LP6_LP3_LP2_AP3_AP2_AP1         NC
## 7   S2017_G1       7        5        7 -2 LP6_LP3_LP2_AP3_AP2_AP1         NC
## 8   S2017_G1       8        5        7 -2 LP7_LP6_LP2_AP3_AP2_AP1          C
## 9   S2017_G1       9        7        9 -2 LP7_LP6_LP2_AP7_AP3_AP1          C
## 10  S2017_G1      10        7        9 -2 LP3_LP2_LP1_AP3_AP2_AP1          C
## 11  S2017_G2      11        0        0  0 LP3_LP2_LP1_AP3_AP2_AP1          C
## 12  S2017_G2      12        2        0  2 LP3_LP2_LP1_AP3_AP2_AP1         NC
## 13  S2017_G2      13        2        2  0 LP3_LP2_LP1_AP3_AP2_AP1         NC
## 14  S2017_G2      14        5        2  3 LP3_LP2_LP1_AP3_AP2_AP1         NC
## 15  S2017_G2      15        5        4  1 LP3_LP2_LP1_AP3_AP2_AP1         NC
## 16  S2017_G2      16        7        4  3 LP3_LP2_LP1_AP3_AP2_AP1         NC
## 17  S2017_G2      17        7        6  1 LP3_LP2_LP1_AP3_AP2_AP1         NC
## 18  S2017_G2      18        7        6  1 LP3_LP2_LP1_AP3_AP2_AP1         NC
## 19  S2017_G2      19        9        6  3 LP9_LP2_LP1_AP5_AP3_AP1          C
## 20  S2017_G2      20       10        6  4 LP9_LP2_LP1_AP5_AP3_AP1         NC
```

Hasta aquí tenemos detectados cuando hay cambios

STINTS

```
df_order$stint <- ifelse(lead(df_order$TF)=="C", df_order$TF, NA)
                                                #Filas que tenemos que conservar

last_row <- tail(df_order, n =1)              #Tendremos que añadirla al final.
last_row$stint <- "LastRow"

knitr::kable(df_order)
```

Merged_SG	id_play	points_L	points_A	PM	Merged_Players	TF_Cambios	stint
S2017_G1	1	0	0	0	LP3_LP2_LP1_AP3_AP2_AP1	NA	NA
S2017_G1	2	0	2	-2	LP3_LP2_LP1_AP3_AP2_AP1	NC	NA
S2017_G1	3	0	2	-2	LP3_LP2_LP1_AP3_AP2_AP1	NC	NA
S2017_G1	4	2	2	0	LP3_LP2_LP1_AP3_AP2_AP1	NC	NC
S2017_G1	5	2	4	-2	LP6_LP3_LP2_AP3_AP2_AP1	C	NA
S2017_G1	6	2	4	-2	LP6_LP3_LP2_AP3_AP2_AP1	NC	NA
S2017_G1	7	5	7	-2	LP6_LP3_LP2_AP3_AP2_AP1	NC	NC
S2017_G1	8	5	7	-2	LP7_LP6_LP2_AP3_AP2_AP1	C	C
S2017_G1	9	7	9	-2	LP7_LP6_LP2_AP7_AP3_AP1	C	C
S2017_G1	10	7	9	-2	LP3_LP2_LP1_AP3_AP2_AP1	C	C
S2017_G2	11	0	0	0	LP3_LP2_LP1_AP3_AP2_AP1	C	NA

Merged_SG	id_play	points_L	points_A	PM	Merged_Players	TF_Cambios	stint
S2017_G2	12	2	0	2	LP3_LP2_LP1_AP3_AP2_AP1	NC	NA
S2017_G2	13	2	2	0	LP3_LP2_LP1_AP3_AP2_AP1	NC	NA
S2017_G2	14	5	2	3	LP3_LP2_LP1_AP3_AP2_AP1	NC	NA
S2017_G2	15	5	4	1	LP3_LP2_LP1_AP3_AP2_AP1	NC	NA
S2017_G2	16	7	4	3	LP3_LP2_LP1_AP3_AP2_AP1	NC	NA
S2017_G2	17	7	6	1	LP3_LP2_LP1_AP3_AP2_AP1	NC	NA
S2017_G2	18	7	6	1	LP3_LP2_LP1_AP3_AP2_AP1	NC	NC
S2017_G2	19	9	6	3	LP9_LP2_LP1_AP5_AP3_AP1	C	NA
S2017_G2	20	10	6	4	LP9_LP2_LP1_AP5_AP3_AP1	NC	NA

```
library(tidyr)
STINTS <- df_order %>% drop_na(stint)
STINTS <- rbind(STINTS, last_row)
STINTS$ID_Stint <- 1:nrow(STINTS)
STINTS <- STINTS[c(9, 1:8)]


STINTS_PM <- STINTS %>% select(c(ID_Stint, Merged_SG, Merged_Players, PM))


knitr::kable(STINTS_PM)
```

	ID_Stint	Merged_SG	Merged_Players	PM
1	1	S2017_G1	LP3_LP2_LP1_AP3_AP2_AP1	0
2	2	S2017_G1	LP6_LP3_LP2_AP3_AP2_AP1	-2
3	3	S2017_G1	LP7_LP6_LP2_AP3_AP2_AP1	-2
4	4	S2017_G1	LP7_LP6_LP2_AP7_AP3_AP1	-2
5	5	S2017_G1	LP3_LP2_LP1_AP3_AP2_AP1	-2
6	6	S2017_G2	LP3_LP2_LP1_AP3_AP2_AP1	1
20	7	S2017_G2	LP9_LP2_LP1_AP5_AP3_AP1	4

```
dim(STINTS_PM)
```

```
## [1] 7 4
```

PARTE 2: Dummys Jugadores

```
vec_players <- unique(c(players_L, players_A))             #Vector con los nombres de los jugadores

length(vec_players)
```

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

```
NA_players <- matrix(data=NA, nrow = dim(STINTS_PM)[1], ncol = length(vec_players)) #Solo jugadores

df_DummyPlayers <- cbind(STINTS_PM, NA_players)                    #DF con jugadores como columnas
names(df_DummyPlayers) <- c(names(STINTS_PM), vec_players)

## TRUE/FALSE si aparecen en la alineacion
df_DummyPlayers[vec_players] <- grepl(vec_players, df_DummyPlayers$Merged_Players, fixed=TRUE)
```

```r
#Me las rellena todas igual al primer jugador que aparece en vec_players (LP1). Como hacer
#para que rellene con todos los jugadores??

#Como deberian ser:
d1 <- grepl(vec_players[1], df_DummyPlayers$Merged_Players, fixed=TRUE) #LP1
d2 <- grepl(vec_players[2], df_DummyPlayers$Merged_Players, fixed=TRUE) #LP2
d3 <- grepl(vec_players[3], df_DummyPlayers$Merged_Players, fixed=TRUE) #LP3
d4 <- grepl(vec_players[4], df_DummyPlayers$Merged_Players, fixed=TRUE) #LP4

df_Dum <- data.frame(
  "LP1" <- d1,
  "LP2" <- d2,
  "LP3" <- d3,
  "LP4" <- d4
)

names(df_Dum) <- paste0("LP", 1:4)
df_Dum
```

```
##     LP1  LP2   LP3   LP4
## 1  TRUE TRUE  TRUE FALSE
## 2 FALSE TRUE  TRUE FALSE
## 3 FALSE TRUE FALSE FALSE
## 4 FALSE TRUE FALSE FALSE
## 5  TRUE TRUE  TRUE FALSE
## 6  TRUE TRUE  TRUE FALSE
## 7  TRUE TRUE FALSE FALSE
```

```
df_DummyPlayers
```

```
##    ID_Stint Merged_SG          Merged_Players PM   LP1   LP2   LP3   LP4   LP5
## 1         1  S2017_G1 LP3_LP2_LP1_AP3_AP2_AP1  0  TRUE  TRUE  TRUE  TRUE  TRUE
## 2         2  S2017_G1 LP6_LP3_LP2_AP3_AP2_AP1 -2 FALSE FALSE FALSE FALSE FALSE
## 3         3  S2017_G1 LP7_LP6_LP2_AP3_AP2_AP1 -2 FALSE FALSE FALSE FALSE FALSE
## 4         4  S2017_G1 LP7_LP6_LP2_AP7_AP3_AP1 -2 FALSE FALSE FALSE FALSE FALSE
## 5         5  S2017_G1 LP3_LP2_LP1_AP3_AP2_AP1 -2  TRUE  TRUE  TRUE  TRUE  TRUE
## 6         6  S2017_G2 LP3_LP2_LP1_AP3_AP2_AP1  1  TRUE  TRUE  TRUE  TRUE  TRUE
## 20        7  S2017_G2 LP9_LP2_LP1_AP5_AP3_AP1  4  TRUE  TRUE  TRUE  TRUE  TRUE
##      LP6   LP7   LP8   LP9  LP10   AP1   AP2   AP3   AP4   AP5   AP6   AP7
## 1   TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## 2  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 3  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 4  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 5   TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## 6   TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## 20  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##      AP8   AP9  AP10
## 1   TRUE  TRUE  TRUE
## 2  FALSE FALSE FALSE
## 3  FALSE FALSE FALSE
## 4  FALSE FALSE FALSE
## 5   TRUE  TRUE  TRUE
## 6   TRUE  TRUE  TRUE
## 20  TRUE  TRUE  TRUE
```