---
title: “Dynamic Model”
author: “Antonio Huerta Montellano”
date: “April 1, 2023”
output:
pdf_document: default
html_document: default

Exploración de los paneles

Importemos los paneles donde un pánel corresponde a los bateadores y, el otro, a los fielderos.

```
setwd("~/Documentos/Github/Proyectos/MLB_HN/")
hitters_panel <- read.csv('ETL_Data/Panel/General/Dynamic_model/dynamic_model_hitter_pca.csv')
fielders_panel <- read.csv('ETL_Data/Panel/General/Dynamic_model/dynamic_model_fielder_pca.csv')
```

Por otro lado, se mostrarán las dimensiones de cada pánel

```
print("Bateadores: ")
```

```
[1] "Bateadores: "
```

```
print(dim(hitters_panel))
```

```
[1] 570 207
```

```
print("")
```

```
[1] ""
```

```
print("Fildeadores: ")
```

```
[1] "Fildeadores: "
```

```
print(dim(fielders_panel))
```

```
[1] 542 226
```

Debido a que en las estadísticas descriptivas se observó un shock en el año de la pandemia COVID-19, se obtendrán las estimaciones quitando el año 2020.

Segmentación por grupo

Lo que haremos es dividir los paneles en ciertas categorías. Primero, veamos todas las posiciones en los páneles

```
print("Bateadores:")
```

```
[1] "Bateadores:"
```

```
print(unique(hitters_panel$Posicion_t))
```

```
 [1] SP C  CF RF DH 1B 2B SS 3B LF RP OF
Levels: 1B 2B 3B C CF DH LF OF RF RP SP SS
```

```
print("")
```

```
[1] ""
```

```
print("Fildeadores:")
```

```
[1] "Fildeadores:"
```

```
print(unique(fielders_panel$Posicion_t))
```

```
[1] SP    RP    RP/CL RF    SS
Levels: RF RP RP/CL SP SS
```

Arriba se muestran las posiciones de los jugadores en nuestras bases de datos. A pesar de que en los bateadores aparezcan posiciones defensivas se debe a que estos juegan tanto como ofensivos como defensivos. Estando en la ofensiva se juega en las misma posición que todos por lo que no es necesario especificar que ocupala posición de bateador (**H**). Sin embargo, cuando se dice que es un bateador designado (**DH**) ya que este solo juega en la ofensiva para sustituir a un lanzador/pitcher.

Por otro lado, veamos cuantas observaciones hay por posición.

```
hitters_panel %>% count(Posicion_t, sort = TRUE)
```

```
   Posicion_t   n
1          SP 112
2           C  76
3          LF  60
4          RF  59
5          2B  53
6          RP  47
7          1B  45
8          3B  31
9          DH  31
10         CF  28
11         SS  27
12         OF   1
```

```
fielders_panel %>% count(Posicion_t, sort = TRUE)
```

```
  Posicion_t   n
1         RP 299
2         SP 206
3      RP/CL  22
4         SS  12
5         RF   3
```

Continuemos con la segmentación de acuerdo a categorías. Primero, obtendremos el split de todas las posiciones y luego concatenaremos de acuerdo a los grupos de interés:

Ofensivos:

- **Bateador designado (DH).**
- **No bateador designado (H).**

Debido a la falta de observaciones para los *outfielders* es que se omitirá su estimación. Por otro lado, debido a que la mayoría de los datos para los fildeadores son de los lanzadores, podemos agruparlos de la siguiente manera

Defensivos:

- **Starting pitcher:** Lanzador inicial (SP).
- **Relief pitcher:** Lanzador de relevo (RP) y lanzador de cierre (RP/CL)
- **Campo corto (SS).**

Segundo, crearemos las categorías de acuerdo a la especificación mencionada arriba

Tercero, concatenaremos estas bases de datos de acuerdo a los grupos señalados anteriormente

Veamos las dimensiones de cada una de los paneles sin el shock de la COVID-19:

```
print("Regular hitter: ")
```

```
[1] "Regular hitter: "
```

```
print(dim(hitter_cov_data))
```

```
[1] 501 207
```

```
print("")
```

```
[1] ""
```

```
print("Designated hitter: ")
```

```
[1] "Designated hitter: "
```

```
print(dim(d_hitter_cov_data))
```

```
[1]  30 207
```

```
print("")
```

```
[1] ""
```

```
print("Relief pitchers: ")
```

```
[1] "Relief pitchers: "
```

```
print(dim(relief_pitcher_cov_data))
```

```
[1] 296 226
```

```
print("")
```

```
[1] ""
```

```
print("Starting pitchers: ")
```

```
[1] "Starting pitchers: "
```

```
print(dim(starting_cov_data))
```

```
[1] 185 226
```

```
print("")
```

```
[1] ""
```

```
print("Short stops: ")
```

```
[1] "Short stops: "
```

```
print(dim(shorts_cov_data))
```

```
[1]  12 226
```

Estimaciones y regresiones

Lo que resta hacer es implementar un algoritmo donde se pueda hacer el siguiente modelo para todas las estadísticas deportiva de acuerdo a si el jugador es defensivo u ofensivo:

$$Y_t(\cdot) = \alpha + \beta_0 X_t + \beta_1 \text{Controles}_t + u_t$$

donde

- $Controles_t$:
  - Equipo.
  - Edad.
  - Año.
- α: Heterogeneidad del jugador.

Creemos la lista de variables sobre las cuáles se va a iterar el clico

Variables para los fildeadores

Las variables base para ambos tipos de jugadores son los controles

```
# Constroles:
vars  <- 'Y_Sueldo_regular_norm_t ~ Edad_t + Anios_de_contrato_t'
```

Estimaciones directas

Pooling

Bateadores

Se obtendrán las estimaciones de las variables referentes a estadísticas deportivas sin controles

```
# Create a model to store the results
hitter_simple_pooling <- list()
hitter_results_simple_pooling_frt <- list()
hitter_results_simple_pooling_sec <- list()
hitter_results_simple_pooling_thr <- list()

# Loop over the variables in var_hitter_list
for (i in 1:9){
  # Run linear regression with grouped errors by country and robust errors
  base_vars_h  <- paste(vars, stat_hitter_t[[i]],
                        sep = '+')
  formula <- paste(base_vars_h,
                   stat_hitter_t_1[[i]],
                   sep = " + ")

  hitter_simple_pooling[[2*i-1]] <- plm(formula, data = hitter_data,
                                        model = "pooling",
                                        index = c("id", "Anio_ref"))

  hitter_results_simple_pooling_frt[[i]] <- coeftest(hitter_simple_pooling[[2*i-1]],
                                                     vcov = vcovHC(hitter_simple_pooling[[2*i-1]],
                                                                   type = "HC1",
                                                                   cluster = "group"))
  if (i <= 5){
    base_vars_h  <- paste(vars, stat_hitter_t[[i+9]],
                          sep = '+')
    formula <- paste(base_vars_h,
                     stat_hitter_t_1[[i+9]],
                     sep = " + ")
```

```
    hitter_simple_pooling[[2*i]] <- plm(formula, data = hitter_data,
                                          model = "pooling",
                                          index = c("id", "Anio_ref"))

    hitter_results_simple_pooling_sec[[i]] <- coeftest(hitter_simple_pooling[[2*i]],
                                                         vcov = vcovHC(hitter_simple_pooling[[2*i]],
                                                                       type = "HC1",
                                                                       cluster = "group"))
  }
}

# Loop over the variables in var_hitter_list
for (i in 19:length(stat_hitter_t_1)){
  # Run linear regression with grouped errors by country and robust errors
  base_vars_h  <- paste(vars, stat_hitter_t[[i]],
                      sep = '+')
  formula <- paste(base_vars_h,
                   stat_hitter_t_1[[i]],
                   sep = " + ")

  hitter_simple_pooling[[i]] <- plm(formula, data = hitter_data,
                                     model = "pooling",
                                     index = c("id", "Anio_ref"))

  hitter_results_simple_pooling_thr[[i - 18]] <- coeftest(hitter_simple_pooling[[i]],
                                                           vcov = vcovHC(hitter_simple_pooling[[i]],
                                                                         type = "HC1",
                                                                         cluster = "group"))
}

# Print the first block of results
stargazer(hitter_results_simple_pooling_frt,
          type = "text",
          title = "Bateadores: Modelo Pooling")
```

```
Bateadores: Modelo Pooling
=================================================================================================
                                                        Dependent variable:
                        -------------------------------------------------------------------------

                          (1)       (2)      (3)      (4)      (5)      (6)      (7)      (8)      (9)
-------------------------------------------------------------------------------------------------
Edad_t                  -0.006** -0.006**  -0.006** -0.006** -0.006** -0.006** -0.006** -0.006** -0.006
                        (0.003)   (0.003)  (0.002)  (0.003)  (0.003)  (0.003)  (0.002)  (0.003)  (0.003

Anios_de_contrato_t      0.001    -0.001    0.001    -0.001  -0.0004   -0.001   0.001    -0.001   -0.00
                        (0.004)   (0.004)  (0.004)  (0.004)  (0.003)  (0.003)  (0.004)  (0.004)  (0.003

X_At_bats_t              -0.001
                        (0.001)

X_At_bats_t_1            -0.001
                        (0.001)
```

X_At_bats_2_t		-0.00002							
		(0.00004)							
X_At_bats_2_t_1		-0.00000							
		(0.00003)							
X_Bateos_t			-0.002*						
			(0.001)						
X_Bateos_t_1			0.0003						
			(0.001)						
X_Bateos_2_t				-0.0001					
				(0.0001)					
X_Bateos_2_t_1				0.0001					
				(0.0001)					
X_Bateos_promedio_t					-0.030				
					(0.019)				
X_Bateos_promedio_t_1					0.018				
					(0.017)				
X_Bateos_promedio_2_t						-0.047			
						(0.030)			
X_Bateos_promedio_2_t_1						0.004			
						(0.017)			
X_Dobles_t							-0.004		
							(0.003)		
X_Dobles_t_1							-0.001		
							(0.003)		
X_Dobles_2_t								-0.0004	
								(0.001)	
X_Dobles_2_t_1								0.001	
								(0.001)	
X_Home_runs_t									-0.00
									(0.004)
X_Home_runs_t_1									0.003
									(0.002)
Constant	0.175**	0.170**	0.163**	0.165**	0.167**	0.162**	0.163**	0.168**	0.171**
	(0.082)	(0.078)	(0.077)	(0.083)	(0.081)	(0.081)	(0.076)	(0.080)	(0.079)

Note: *p<0.1; **p<0.05; ***p<0.

```
# Print the second block of results
stargazer(hitter_results_simple_pooling_sec,
          type = "text",
          title = "Bateadores: Modelo Pooling")
```

Bateadores: Modelo Pooling

	Dependent variable:				
	(1)	(2)	(3)	(4)	(5)
Edad_t	-0.006**	-0.006**	-0.006**	-0.006**	-0.005**
	(0.003)	(0.003)	(0.003)	(0.003)	(0.003)
Anios_de_contrato_t	-0.001	0.001	-0.001	-0.0005	0.00003
	(0.003)	(0.004)	(0.004)	(0.003)	(0.004)
X_Home_runs_2_t	-0.001				
	(0.001)				
X_Home_runs_2_t_1	-0.0001				
	(0.0004)				
X_Juegos_iniciados_t		-0.002			
		(0.001)			
X_Juegos_iniciados_t_1		-0.001			
		(0.001)			
X_Juegos_iniciados_2_t			-0.0001		
			(0.0002)		
X_Juegos_iniciados_2_t_1			0.00005		
			(0.0001)		
X_Porcentaje_On_base_plus_slugging_t				-0.020	
				(0.014)	
X_Porcentaje_On_base_plus_slugging_t_1				-0.001	
				(0.013)	
X_Porcentaje_On_base_plus_slugging_2_t					-0.026**
					(0.013)
X_Porcentaje_On_base_plus_slugging_2_t_1					0.008
					(0.011)
Constant	0.173**	0.174**	0.171**	0.174**	0.155*
	(0.080)	(0.080)	(0.078)	(0.081)	(0.083)

```
===========================================================================================
Note:                                                           *p<0.1; **p<0.05; ***p<0.01
```

```r
# Print the third block of results
stargazer(hitter_results_simple_pooling_thr,
          type = "text",
          title = "Bateadores: Modelo Pooling")
```

```
Bateadores: Modelo Pooling
==============================================================================
                                           Dependent variable:
                       -------------------------------------------------------

                          (1)      (2)      (3)      (4)       (5)       (6)
------------------------------------------------------------------------------
Edad_t                 -0.006** -0.006** -0.006** -0.006** -0.007*** -0.006***
                        (0.003)  (0.003)  (0.003)  (0.003)  (0.002)   (0.002)

Anios_de_contrato_t     0.0002   -0.002   -0.002   -0.001   -0.006    -0.002
                        (0.004)  (0.004)  (0.003)  (0.003)  (0.004)   (0.003)

X_Runs_batted_in_t     -0.003*
                        (0.001)

X_Runs_batted_in_t_1    0.001
                        (0.002)

X_Runs_batted_in_2_t             0.0001
                                (0.0002)

X_Runs_batted_in_2_t_1           0.0001
                                (0.0002)

X_Triples_t                               -0.011
                                          (0.008)

X_Triples_t_1                            0.012**
                                          (0.005)

X_Triples_2_t                                      -0.002
                                                   (0.004)

X_Triples_2_t_1                                     0.001
                                                   (0.001)

X_WAR_t                                                     0.016**
                                                            (0.007)

X_WAR_t_1                                                   0.013**
                                                            (0.006)

X_WAR_2_t                                                              0.006
                                                                      (0.004)
```

X_WAR_2_t_1						0.005**
						(0.002)
Constant	0.163**	0.177**	0.166**	0.168**	0.219***	0.192**
	(0.079)	(0.081)	(0.080)	(0.081)	(0.078)	(0.076)
Note:						*p<0.1; **p<0.05; ***p<0.01