Métodos estadísticos aplicados al baloncesto

Paula Moreno Blazquez

Enero 2022

Resumen

Hoy en día, el deporte es un hobby muy popular por todo el mundo. Des de pequeños, los niños practican algún tipo de deporte, especialmente aquellos que son de equipo. Eso nos lleva a querer saber más del deporte, más detalles, más información. Nos entra la curiosidad de "¿quién es el mejor jugador?", "Qué equipo es mejor?", o incluso intentar prevenir qué equipo ganará según sus resultados anteriores. Y gracias a los avances tecnológicos e informáticos, cada vez se nos facilita más poder seguir un deporte des de casa, ver la estadística de los deportistas e incluso hay plataformas o juegos que nos permiten ser, de manera virtual, managers de los clubs y, por lo tanto, nos facilitan mucha información que antes era más difícil de saber.

Eso hace que, de manera progresiva, también mejore el estudio y el análisis de cada deporte, y cada vez sea más específica para cada deporte, implementando nuevos recursos para mejorar los resultados. Pero, ¿son lo suficientemente eficaces los análisis que se realizan actualmente en Europa? ¿O dichos análisis estan anticuados y requieren de una actualización?

Índice

1. Introducción

En este trabajo estudiaremos más a fondo el Baloncesto, el segundo deporte más popular de Europa (solo superado por el futbol), y el cual tengo interés personal, ya que lo practico des de los 4 años.

Esta idea de estudio surgió del constante pensamiento de que los análisis actuales que se hacen en este deporte en Europa son bastante pobres a nivel informativo, puesto que se basan en conceptos muy básicos, y principalmente ofensivos (que vendría a ser el 50 % de un partido). Para que nos hagamos una idea, el estadístico por preferencia es el llamado "Valoraciónz que se originó en 1991 (hace 30 años) y des de entonces nunca se ha modificado.

Es por eso que, considero que actualmente los análisis que se hacen de este deporte necesitan una actualización significativa para llegar a informar de todos aquellos datos que hoy en día si se pueden recoger gracias a los avances tecnológicos, y de los cuales no se analizan por falta de dinero o porque se consideran poco relevantes.

El objetivo principal de este estudio es mejorar los análisis que se elaboran de cada partido, para poder encontrar una variable respuesta que nos diga que aportación al equipo tiene cada jugador personalmente, disminuyendo la diferencia de pesos que hay actualmente entre las aportaciones ofensivas y las aportaciones defensivas.

Este documento se estructura de la siguiente manera: a continuación, se realizará una breve explicación de los recursos informáticos que se han utilizado para realizar este estudio, seguidamente se explicará brevemente los conceptos de baloncesto que son necesarios para entender los tecnicismos del trabajo y se presentaran posibles análisis que se realizan. Finalmente, se describirá la base de datos con la que se ha trabajado y sus variables, y también se explicará en profundidad el análisis que se desarrollará en este trabajo, el Más/Menos Ajustado (*Adjusted Plus/Minus, APM*). EN la sección de resultados presentaremos la resolución del análisis y finalmente discutiremos, en la sección de conclusiones, los resultados obtenidos.

2. ¿Qué es GitHub?

GitHub es una plataforma de alojamiento, propiedad de Microsoft, que ofrece a los desarrolladores la posibilidad de crear repositorios de código y guardarlos en la nube de forma segura, usando un sistema de control de versiones, llamado Git.

Como he comentado, facilita la organización de proyectos y permite la colaboración de varios desarrolladores en tiempo real. Es decir, nos permite centralizar el contenido del repositorio para poder colaborar con los otros miembros de nuestro grupo des de varios dispositivos.

GitHub está basada en el sistema de control de versiones distribuidas de Git, por lo que se puede contar con sus funciones y herramientas, aunque GitHub ofrece varias opciones adicionales y su interfaz es mucho más fácil de manejar, por lo que no es absolutamente necesario que las personas que lo utilizan tengan un gran conocimiento técnico.

2.1. Ventajas

Hay un gran número de razones por las que GitHub es una gran opción para el control y gestión de proyectos de código. Como por ejemplo:

- GitHub permite que alojemos proyectos en repositorios de forma gratuita
- Los repositorios son públicos por defecto. Sin embargo, GitHub te permite también alojar tus proyectos de manera privada
- Puedes crear y compartir páginas web estáticas con GitHub Pages
- Facilita compartir tus proyectos de una forma mucho más fácil y crear un portafolio
- Te permite colaborar para mejorar los proyectos de otros y a otros mejorar o aportar a los tuyos
- Ayuda reducir significativamente los errores humanos y escribir tu código más rápido con GitHub Copilot
- Te da control de versiones, una herramienta muy útil.

2.2. ¿Qué es el control de versiones?

Se le llama control de versiones a la administración de los cambios que se realizan sobre los elementos o la configuración de algún proyecto. En otras palabras, el control de versiones sirve para conocer y autorizar los cambios que hagan los colaboradores en tu proyecto, guardando información extra de qué están, incluyendo los cambios y cuándo se hicieron. Este control comienza con una versión básica del documento y luego va guardando los cambios que se hagan a lo largo del proyecto.

El control de versiones es una herramienta muy valiosa, pues con ella puedes tener acceso a las versiones anteriores de tu proyecto si es que en algún momento no llega a funcionar de forma correcta.

2.3. ¿Qué es Git?

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia, la confiabilidad y compatibilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente.

2.3.1. Diferencias Git vs GitHub

Entonces, ¿qué diferencia a Git de GitHub? La principal diferencia es que Git es un sistema que permite establecer un control de versiones, mientras que GitHub es una plataforma que ofrece un grupo de funciones que facilitan el uso de Git y la colaboración en tiempo real, así como el almacenamiento en la nube.

3. El baloncesto

3.1. Historia y reglas básicas

El baloncesto es un deporte de equipo que se originó en 1891, por James Naismith, profesor de educación física en la escuela, que buscaba idear un deporte que sus alumnos pudieran practicar bajo techo, pues los duros inviernos en Nueva Inglaterra dificultaban la realización de ejercicio al aire libre. Con el paso de los años, este deporte, que empezó como actividad de colegio, ha ido evolucionando mucho, añadiendo más reglas, conceptos nuevos, límites de números de jugadores, se ha determinado tiempos de juego, las canastas tienen un valor distinto según la distancia, etc.

Actualmente, las normas más básicas de este deporte son:

- En las ligas superiores, hay un total de 4 cuartos de 10 minutos y pueden estar en pista 5 jugadores por equipo.
- No te puedes desplazar con la pelota en las manos, es obligatorio botar con una mano (si no será una infracción y conllevará la perdida de pelota y saque de banda del equipo rival).
- Cada jugador puede realizar hasta un total de 5 faltas, que será penalizado con un saque de banda o con un tiro libre (dependerá de la situación). El jugador que realiza 5 faltas será expulsado del partido.
- El objetivo es encestar el máximo de puntos posibles, teniendo en cuenta que pueden sumar 1, 2 o 3 puntos, según la distancia.

3.2. Conceptos y definiciones básicas del baloncesto:

Para que podamos entender a que nos referimos en este trabajo, es necesario comprender unos conceptos básicos de vocabulario. Tendremos en cuenta los conceptos que se necesitan para realizar la valoración del jugador y/o del equipo que se utilizan en las estadísticas federadas.

- Puntos: Acumulación de canastas encestadas multiplicadas por su valor, que cada jugador y/o equipo realiza durante el partido
- Minutos: Número de minutos que el jugador está en pista

- Falta: Acción en la que un defensor bloquea el avance de su rival sin tener control de balón o de manera no reglamentaria (empujar, agarrar...)
- Pérdidas de balón: cuando un equipo pierde el control del balón y pasa a ser del equipo rival.
- Rebotes: Recuperación de pelota después de que el tiro sea ejecutado, pero no haya encestado.
- Recuperación de balón: Cuando un equipo consigue robar el balón al equipo rival.
- Asistencia: Es un pase a un jugador que se encuentra en una posición de ventaja o que le ayuda a conseguir una canasta sin hacer ningún bote.
- Tapón: Bloqueo de un tiro en el aire.

4. Posibles análisis realizables en el baloncesto

Viendo la gran cantidad de datos que se pueden extraer de cada partido (y de cada equipo), se han ido creando análisis que recogen estos datos y los analizan para ayudarnos a identificar y desarrollar hipótesis sobre cada jugador y/o equipo.

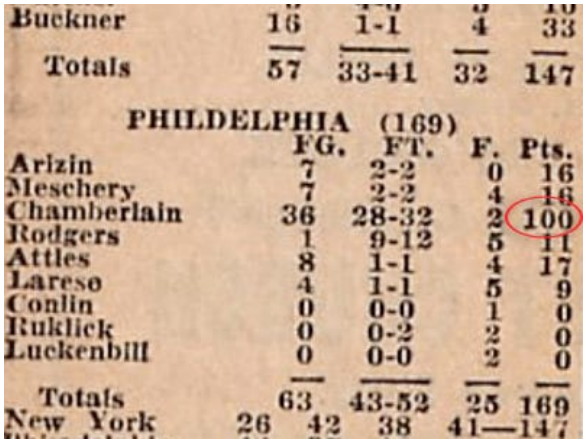
4.1. Boxscore

El primer análisis que se hizo fue un *Box Score* (Caja de puntuación) donde se recopilaba únicamente los puntos de cada jugador según el valor de esta y las faltas realizadas. Posteriormente, se fue mejorando añadiendo conceptos como rebotes, tapones, perdidas de balón, recuperaciones de balón... Y se añadió el estadístico (que acutalmente es por defecto) que se realiza a partir de todos estos datos: "Valoración"(en inglés PIR, *Performance Index Rating*) que engloba todo lo básico que pasa en el partido de manera individual y que, cuanto más positivo, mejor. Este estadístico se calcula utilizando la siguiente fórmula:

$$\begin{aligned} PIR = &(Puntos + Rebotes + Asistencias + Robos + Tapones + FaltasRecibidas) - \\ &(TirosdeCampoFallados + TirosLibresFallados + TaponesRecibidos + Prdidas + FaltasRealizadas) \end{aligned} \tag{1}$$

(en el Annexo 1 encontraréis la descripción de cada variable)

Posteriormente, se añadió la variable "Más/Menos"(*P/M, Plus/Minus*) que tiene que ver con la diferencia de puntos en el marcador durante el tiempo que el jugador está en pista. Esta variable sirve para ver la contribución de los jugadores cuando están en pista. Todos los jugadores parten inicialmente con un 0, y según van entrando y saliendo de la pista, esta variable se va actualizando. Por ejemplo, los jugadores que son del quinteto inicial, empiezan con el marcador 0 - 0, y un $P/M = 0$. Si en el minuto 5, se substituye un jugador en cancha del equipo local (J1) por otro que está descansando (J2), y el marcador va 12 - 7, el P/M del J1 pasará a ser +5. Y si al cabo de 3 minutos, se sustituye el J2 por otro (J3) y el marcador ahora va 20 - 9, el P/M del J2 será +6 ($(20 - 12) - (9 - 7) = 8 - 2 = +6$).



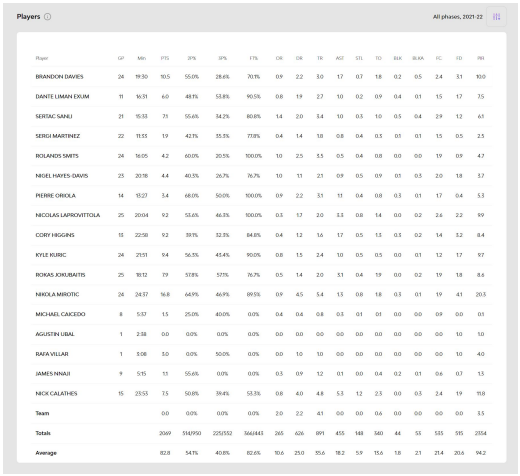
Buckner	16	1-1	4	33
Totals	57	33-41	32	147

PHILDELPHIA (169)

	FG.	FT.	F.	Pts.
Arizin	7	2-2	0	16
Meschery	7	2-2	4	16
Chamberlain	36	28-32	2	100
Rodgers	1	9-12	5	11
Attles	8	1-1	4	17
Larese	4	1-1	5	9
Conlin	0	0-0	1	0
Ruklick	0	0-2	2	0
Luckenbill	0	0-0	2	0
Totals	63	43-52	25	169

New York 26 42 38 41—147

Figura 1: *Boxscore del partido de la NBA de Philadelphia Warriors contra New York Knicks, del 2 de Marzo de 1962*



Players — All phases, 2021-22

Player	GP	Min	PTS	2P%	3P%	FT%	OR	DR	TR	AST	STL	TO	BLK	BLKA	FC	FD	PIR
BRANDON DAVIES	24	19:30	10.5	55.0%	28.6%	70.1%	0.9	2.2	3.0	1.7	0.7	1.8	0.2	0.5	2.4	3.1	10.0
DANTE LIMAN EXUM	11	16:31	6.0	48.1%	53.8%	90.5%	0.8	1.9	2.7	1.0	0.2	0.9	0.4	0.1	1.5	1.7	7.5
SERTAC SANLI	21	15:33	7.1	55.6%	34.2%	80.8%	1.4	2.0	3.4	1.0	0.3	1.0	0.5	0.4	2.9	1.2	6.1
SERGI MARTINEZ	22	11:55	1.9	42.1%	55.3%	77.8%	0.4	1.4	1.8	0.8	0.4	0.5	0.1	0.1	1.5	0.5	2.5
ROLANDS SMITS	24	16:05	4.2	60.0%	20.5%	100.0%	1.0	2.5	3.5	0.5	0.4	0.8	0.0	0.0	1.9	0.9	4.7
NIGEL HAYES-DAVIS	23	20:18	4.4	40.5%	26.7%	76.7%	1.0	1.1	2.1	0.9	0.5	0.9	0.1	0.3	2.0	1.8	3.7
PIERRE ORIOLA	14	13:27	3.4	68.0%	50.0%	100.0%	0.9	2.2	3.1	1.1	0.4	0.8	0.3	0.1	1.7	0.4	5.3
NICOLAS LAPROVITTOLA	25	20:04	9.2	53.6%	46.3%	100.0%	0.3	1.7	2.0	3.3	0.8	1.4	0.0	0.2	2.6	2.2	9.9
CORY HIGGINS	15	22:50	9.2	39.1%	32.3%	84.8%	0.4	1.2	1.6	1.7	0.5	1.5	0.3	0.2	1.4	3.2	8.4
KYLE KURIC	24	21:51	9.4	56.5%	45.4%	90.0%	0.8	1.5	2.4	1.0	0.5	0.5	0.0	0.1	1.2	1.7	9.7
ROKAS JOKUBAITIS	25	18:12	7.9	57.8%	57.1%	76.7%	0.5	1.4	2.0	3.1	0.4	1.9	0.0	0.2	1.9	1.8	8.6
NIKOLA MIROTIC	24	24:37	16.8	64.9%	46.9%	89.5%	0.9	4.5	5.4	1.3	0.8	1.8	0.3	0.1	1.9	4.1	20.3
MICHAEL CACEDO	8	5:37	1.5	25.0%	40.0%	0.0%	0.4	0.4	0.8	0.3	0.1	0.1	0.0	0.0	0.9	0.0	0.1
AGUSTIN UBAL	1	2:38	0.0	0.0%	0.0%	0.0%	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
RAFA VILLAR	1	3:08	3.0	0.0%	50.0%	0.0%	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	4.0
JAMES NNAJI	9	5:15	1.1	55.6%	0.0%	0.0%	0.3	0.9	1.2	0.1	0.0	0.4	0.2	0.1	0.6	0.7	1.3
NICK CALATHES	15	23:53	7.5	50.8%	39.4%	53.3%	0.8	4.0	4.8	5.3	1.2	2.3	0.0	0.3	2.4	1.9	19.8
Team			0.0	0.0%	0.0%	0.0%	2.0	2.2	4.1	0.0	0.0	0.6	0.0	0.0	0.0	0.0	3.5
Totals			2069	1141/950	225/552	366/443	265	626	891	455	148	360	44	53	535	515	2354
Average			82.8	54.1%	40.8%	82.6%	10.6	25.0	35.6	18.2	5.9	15.6	1.8	2.1	21.4	20.6	94.2

Figura 2: *Boxscore del partido de la Euroliga de Real Madrid contra FC Barcelona, del 11 de Febrero del 2022*

Aunque un *Box Score* es muy útil para realizar análisis básicos, ya que es muy visual y cualquier persona sin la necesidad de muchos recursos puede analizar y predecir ciertos valores, pero estadísticamente perdemos una parte importante de la información de los datos, puesto que no nos los muestra progresivamente, sino que nos da los valores acumulados al final del tiempo establecido, y muchas veces contiene información engañosa, especialmente en las estadísticas defensivas.

Por lo que, para el desarrollo temporal del partido y para conocer cierta información de equipo que piden entrenadores y clubs, no nos sirve (como por ejemplo la eficacia de los quintetos, el desarrollo del marcador o de cualquier otra variable del equipo entero durante un tiempo determinado del partido, etc.)

4.2. Play-by-play

Este tipo de recogida de información se creo para solucionar el problema que teniamos con el *Box Score*. Los datos de *Play-by-Play (PBP)* han sido la fuente principal de muchas estadísticas avanzadas, como el más-menos ajustado, que se desarrollará en este trabajo.

Play-by-Play proporciona una transcripción del juego en un formato de eventos individuales. Los datos típicos de jugada por jugada deben tener la siguiente información:

- El tiempo de la posesión
- El jugador que inició la posesión (en caso de robo o rebote defensivo)
- El jugador contrario que inició la posesión (en caso de un tiro fallado o pérdida de balón), incluida la ubicación en el piso desde donde se realizó el tiro y algunos otros identificadores únicos que usamos para clasificar el tipo de posesión.

4.3. Shot-charts

5. Bibliografia

6. Annexo

6.1. Descripción de las variables

- MIN (*Minutes*): Minutos totales jugados
- PTS (*Points*): Puntos totales realizados
- 2FGA (*2-point Field Goals Attempted*): Número de canastas de 2 puntos intentadas
- 2FGM (*2-point Field Goals Made*): Número de canastas de 2 puntos anotadas
- 3FGA (*3-point Field Goals Attempted*): Número de canastas de 3 puntos ("triples") intentadas
- 3FGM (*3-point Field Goals Made*): Número de canastas de 3 puntos ("triples") anotadas
- FTA (*Free Throws Attempted*): Número de tiros libres intentados
- FTM (*Free Throws Made*): Número de tiros libres anotados

6.2. Código R

```
> library(readr)
> library(dplyr)
> library(ggplot2)
> library(here)
> library(tidyverse)

> pbp2018 <- read.csv(file="pbp2018.csv", head=TRUE, sep=",")
> names(pbp2018)

 [1] "season"         "game_code"      "play_number"    "team_code"
 [5] "player_name"    "play_type"      "time_remaining" "quarter"
 [9] "points_home"    "points_away"    "play_info"      "seconds"
[13] "home_team"      "away_team"      "home"           "team_name"
[17] "last_ft"        "and1"           "home_player1"   "home_player2"
[21] "home_player3"   "home_player4"   "home_player5"   "away_player1"
[25] "away_player2"   "away_player3"   "away_player4"   "away_player5"
[29] "lineups"
```

Se encontraron errores en la extracción de las alineaciones. Corrección de Sergio:

```
> pbp_2018 <- read_csv("pbp2018.csv")
> ## Check how many rows are affected by this
> bad_lineups <- pbp_2018 %>%
+   select(matches("_player[1-5]")) %>%
+   apply(1, function(x) max(table(x)) > 1)
> pbp_bad <- pbp_2018 %>%
+   filter(bad_lineups)
> pbp_bad %>%
+   select(season, game_code, play_number, play_type, away_player4, away_player5)

# A tibble: 10,260 x 6
   season game_code play_number play_type away_player4 away_player5
    <dbl>     <dbl>       <dbl> <chr>     <chr>        <chr>
 1   2018         2          93 IN        TOMIC, ANTE  TOMIC, ANTE
 2   2018         2          94 OUT       TOMIC, ANTE  TOMIC, ANTE
 3   2018         2          95 OUT       TOMIC, ANTE  TOMIC, ANTE
 4   2018         2          96 IN        TOMIC, ANTE  TOMIC, ANTE
 5   2018         2          97 OUT       TOMIC, ANTE  TOMIC, ANTE
 6   2018         2          98 IN        TOMIC, ANTE  TOMIC, ANTE
 7   2018         2          99 OUT       TOMIC, ANTE  TOMIC, ANTE
 8   2018         2         100 FTM       TOMIC, ANTE  TOMIC, ANTE
 9   2018         2         101 FTM       TOMIC, ANTE  TOMIC, ANTE
10   2018         2         116 AST       KURIC, KYLE  KURIC, KYLE
# ... with 10,250 more rows

> ## Solución
> source(here("R", "fix-lineups.R"))
> ## Function fix_lineups() only takes data from a single game,
> ## so I split the data and apply the function to each splitted data frame.
> pbp_2018_fixed <- split(pbp_2018, factor(pbp_2018$game_code)) %>%
+   map_df(fix_lineups)
> ## Check that this has been fixed
> pbp_2018_fixed %>%
+   select(matches("_player[1-5]")) %>%
+   apply(1, function(x) max(table(x)) > 1) %>%
+   sum()

[1] 0
```

6.2.1. MAS/MENOS Clásico

6.2.2. MAS/MENOS AJUSTADO, APM

```
> # Lista de jugadores:
> players_rep <- pbp_2018_fixed$player_name
> players <- unique(players_rep)
> length(players) #Numero de jugadores

[1] 253

> unique(pbp_2018_fixed$play_type) #Variables finales BBDD

 [1] "BP"      "TPOFF"   "2FGM"    "2FGA"    "DRB"     "3FGA"    "ORB"
 [8] "CPF"     "RPF"     "FTM"     "TOV"     "STL"     "3FGM"    "AST"
[15] "OF"      "TOUT"    "IN"      "OUT"     "FTA"     "EP"      "RBLK"
[22] "BLK"     "EG"      "TOUT_TV" "CMU"     "C"       "CMT"     "B"
[29] "AS"      "RV"      "CMD"     "D"
```