Práctica 2: Limpieza y análisis de datos

UOC - Tipología y ciclo de vida de los datos

Miguel Santos Pérez y Alejandro Arzola García

17 de junio de 2020

Índice

1 Descripción del dataset

En la Práctica 1 de la asginatura de *Tipología y ciclo de vida de los datos* se creó un proceso *web scraping* en el que se obtuvieron los datos relacionados a todos los partidos y las estadísticas individuales de todos los jugadores de la actual temporada (2019-2020) de la Euroliga, la máxima competición de clubes de baloncesto de Europa.

El objetivo final era contribuir al aumento de explotación de los datos para los equipos europeos y tratar de igualar lo que se realiza en América con las poderosas franquicias de NBA.

Mediante el proceso de obtención de datos realizado en *Python* que comentamos anteriormente, conseguimos crear dos datasets: el primero contiene los datos generales de un partido (nombre de los equipos, puntuación de cada equipo, fecha y hora del partido) y el segundo contiene más de quince estadísticas individuales para cada jugador por cada partido (puntos, minutos, rebotes, asistencias, ...). Este trabajo está disponible en un repositorio de *GitHub* al que se puede acceder haciendo click aquí.

Para esta segunda práctica queremos seguir progresando para lograr este objetivo y por ello vamos a utilizar los datasets que hemos mencionado para realizar un análisis estadístico.

Los datasets y el código utilizado para generar el análisis estadístico que se desarrolla a continuación está disponible en el siguiente repositorio de *GitHub*:

- https://github.com/aarzola-uoc/practica2-tycvd

---

2 Integración y selección de los datos de interés a analizar

Tras realizar un análisis en profundidad de los dos datasets de los que disponemos, hemos tomado la decisión de utilizar únicamente el que contiene las estadísticas individuales por partido de cada jugador.

Los datos relevantes del primer dataset son exclusivamente los nombres de los equipos y el marcador. Sin embargo, estos datos son posibles calcularlos a partir del segundo dataset, incluso aumentar los datos de equipo ya que son la suma de las estadísticas de todos los jugadores para cada partido.

En base a esto, hemos decidido que se creará un nuevo dataset utilizando código R, a partir del dataset de estadísticas de los jugadores, que tendrá las estadísticas de cada equipo por partido (suma de cada jugador es el total del equipo).

Además vamos a calcular una serie de estadísticos muy interesantes y que permiten un análisis mucho más técnico y útli para los equipos. Estos estadísticos son..... **FALTA POR HACER (Miguel): definición de estadísticos**

2.1 Carga de datos originales

El primer paso que tenemos que realizar es cargar el fichero csv *euroleague_stats_per_game.csv* que contiene el dataset con los datos individuales por partido:

```
data <- read.csv2('../csv/euroleague_stats_per_game.csv')
```

Hemos utilizado la función `read.csv2` ya que el fichero a cargar está en formato *csv* español, es decir, los separadores son el caracter `;`.

Mostramos los primeros registros del dataset para verificar que los datos se han cargado correctamente:

```
head(data)
```

```
##   MatchId                Team PlayerNumber          PlayerName   Min Pts X2FG
## 1       1 Khimki Moscow Region            1       SHVED, ALEXEY 30:19  22  4/7
## 2       1 Khimki Moscow Region            5       BOOKER, DEVIN 19:21   4  1/4
```

```
## 3       1 Khimki Moscow Region            6        TIMMA, JANIS 33:53  17  1/3
## 4       1 Khimki Moscow Region            8 ZAYTSEV, VYACHESLAV 10:23   0    0
## 5       1 Khimki Moscow Region            9      VIALTSEV, EGOR  4:58   0    0
## 6       1 Khimki Moscow Region           10 DESIATNIKOV, ANDREI   DNP   -    -
##   X3FG   FT O D T As St To Fv Ag Cm Rv PIR
## 1 2/10 8/10 0 2 2  6  0  3  1  0  1  5  19
## 2    0  2/2 3 3 6  0  2  1  0  0  2  1   7
## 3 5/12    0 1 3 4  1  4  1  0  0  5  2  13
## 4    0    0 1 1 2  2  1  1  0  0  2  1   3
## 5    0    0 0 0 0  1  0  0  0  0  2  0  -1
## 6    -    - - - -  -  -  -  -  -  -  -   -
```

Mostramos el número de registros transmitidos:

```
nrow(data)
```

```
## [1] 6505
```

---

3 Limpieza de los datos

El dataset dispone de 6505 registros y 20 variables. Estas variables son:

- **MatchId:** identificador único de partido.
- **Team:** equipo al que pertenece el jugador.
- **PlayerNumber:** número del jugador.
- **PlayerName:** nombre del jugador.
- **Min:** minutos jugados.
- **Pts:** puntos anotados.
- **2FG:** tiros de dos (metidos-anotados).
- **3FG:** tiros de tres (metidos-anotados).
- **FT:** tiros libres (metidos-anotados).
- **O:** rebotes ofensivos.
- **D:** rebotes defensivos.
- **T:** rebotes totales.
- **As:** asistencias.
- **St:** recuperaciones.
- **To:** pérdidas.
- **Fv:** tapones a favor.
- **Ag:** tapones en contra.
- **Cm:** faltas cometidas.
- **Rv:** faltas recibidas.
- **PIR:** valoración del jugador.

En total disponemos de estadísticas de 252 partidos y de 306 jugadores diferentes.

3.1 Tratamiento de datos

3.2 Elementos vacíos

3.3 Identificación y tratamiento de valores extremos

---

4 Análisis de los datos

FALTA POR HACER

4.1 Selección de los grupos de datos que se quieren analizar/comparar

4.2 Comprobación de la normalidad y homogeneidad de la varianza

4.3 Pruebas estadísticas para comparar los grupos de datos

4.3.1 Contraste de hipótesis

4.3.2 Correlaciones

4.3.3 Regresiones

---

5 Exportación de datos finales

A continuación vamos a exportar nuestro *dataframe* final a un archivo csv. Este archivo se llamará *euroleague_stats_per_game_clean.csv*. Utilizamos la función `write.csv2()` para exportar el fichero en formato *csv* español:

```
write.csv2(data, row.names = TRUE, file = "../csv/euroleague_stats_per_game_clean.csv")
```

Este nuevo dataset también está disponible en el repositorio de *GitHub* mencionado en el primer apartado de este documento.

---

6 Representación de los resultados a partir de tablas y gráficas

FALTA POR HACER

---

7 Resolución del problema

FALTA POR HACER

---

8 Contribuciones al trabajo

Contribuciones	Firma
Selección del dataset	MSP, AAG
Creación del repositorio GitHub	MSP, AAG
Desarrollo código en R	MSP, AAG
Redacción de las respuestas	MSP, AAG