Introducción a la Ciencia de Datos

Guía de trabajos prácticos N°8

Tema: *Relational Data*

Esta vez nos toca trabajar con datos de `football_data` que van a encontrar en el campus.

Análisis del conjunto de datasets

Vamos a trabajar con 7 *datasets* de estadísticas de fútbol y apuestas de las 5 mejores ligas europeas entre los años 2014 y 2020. Los datos fueron tomados de acá.

1. Exploren todos los *datasets* para comprender los atributos de cada tabla. La fuente de los datos les puede resultar muy útil para ello (tengan en cuenta que algunos datasets fueron modificados y se quitaron algunos atributos). En este link y este otro van a encontrar información sobre qué son las Xstats. Pueden utilizar comandos como `glimpse()` o `colnames()` para este ejercicio.

Disclaimer 1: Especial atención a la diferencia entre *shots* (tiros) y *shots on target* (tiros al arco). Acá hay una explicación.

Disclaimer 2: Para importar el dataset de *players* (jugadores) van a necesitar agregar el siguiente parámetro al comando `read_csv` para no tener problema con los caracteres especiales:

```
read_csv(ruta_de_acceso, locale = readr::locale(encoding = "latin1"))
```

2. Hagan un esquema a mano o utilizando una Hoja de Cálculo de Google con las variables de cada *dataset* (algo así como hicimos para el *dataset* `vuelos` en la presentación de esta clase).

3. Utilicen el esquema armado para identificar las claves primarias y las claves foráneas (o conjuntos de claves si no fuera suficiente con una sola) en los diferentes *datasets*. Verifiquen que la clave o el conjunto de claves elegidas para cada *dataset* identifiquen de forma única cada observación. Para eso pueden utilizar el comando `count()` con las claves primarias y buscar las entradas con n mayor a uno (ver Cap. 13.3 de R para Ciencia de Datos).

4. ¿Es posible detectar claves primarias en todas las tablas?

5. Piensen cómo podrían conectarse los diferentes *datasets*. Tengan en cuenta que, así como pasaba con los *datasets* de flights, hay columnas que refieren a la misma variable pero se llaman diferente.

Trabajando con comandos de *Relational Data*

A partir de este momento es **fundamental** que inviertan tiempo en pensar cuál de todos los *datasets* es el más adecuado para responder la pregunta que les estamos o se están haciendo. Se pueden encontrar las mismas respuestas por diferentes caminos, pero si no lo piensan con detenimiento pueden tomar rumbos mucho más empantanados de lo necesario.

6. Encuentren los 10 equipos que más goles metieron en todas las Ligas de Europa. Para ello primero deberán identificar el *dataset* que tiene esa información y luego trabajar con él. Para esto último pueden ser de utilidad los comandos `group_by()`, `summarise()`, `slice_max()`, `order()` ó `arrange()` y `head()`. Si no recuerdan cómo funcionan, busquen en la documentación o GOOGLE IT!. Generen un *data frame* que tenga 10 filas que muestren en una columna el `TeamID` y en otra la cantidad de goles totales.

7. Ahora agreguen en una nueva columna del *data frame* anterior el nombre del equipo. Para esto van a necesitar los comandos que aprendimos durante la clase expositiva para unir *datasets*.

8. Hagan un gráfico de barras que muestre la información del ejercicio anterior, donde cada barra tenga el nombre del equipo correspondiente.

9. Repitan los ejercicios 6. a 8. pero en vez de analizar la cantidad de goles totales, analicen los tiros al arco (*shots*).

Sigamos practicando con el dataset

10. Encontremos la Liga en la cual tienen los equipos con mejor relación tiros - goles.

- Utilicen el *dataset* `teamstats` y agreguen una columna que indique a qué Liga pertenece cada tiro realizado. Para eso van a necesitar crear un *dataset* que tenga sólo las variables `GameID` y `LeagueID` y realizar algún tipo de unión entre esta y la tabla `teamstats`. Chequeen que la unión haya sido correcta: no deberían aparecer duplicados en el dataset `teamstats`.
- Armen un *dataset* que indique la cantidad de tiros (*shots*) y de goles (*goals*) **por equipo**, donde además se muestre a qué liga pertenece cada uno.
- Realicen un gráfico de dispersión de la cantidad de goles en función de la cantidad de tiros y coloreen según la Liga. Pueden probar también utilizando el comando `facet_wrap(~Liga)` para separar los gráficos según la Liga.
- Realicen un ajuste lineal del tipo

```
goals ~ a.shoots
```

para cada Liga y con ello determinen qué Liga tiene la mejor relación *shots-goals* por equipo.

- *Uff... qué laburo esto de tener que filtrar cada Liga para poder hacer estos ajustes.*
- No se preocupen, ¡la semana que viene vamos a aprender a hacer esto en una sola línea de código!

11. Elijan una de las 5 ligas europeas. Quédense con los 5 jugadores que más goles han

metido en esa Liga. Analicen la distribución de tiempos (en minutos) en los cuales estos jugadores realizaron los goles **sin importar en qué club lo hicieron**. Para ello:

- Consideren que todos los goles realizados son el resultado de tiros al arco (*shot*). Van a poder entonces encontrar todos los goles realizados a partir del atributo *shotResult* del dataset `shots`.
- Puede serles útil el comando `semi_join()` como se explica en el Cap. 13.5 de R para Ciencia de Datos.
- Pueden usar `ggridges` para mostrar las 5 distribuciones en un mismo gráfico.

Ahora un poquito de modelo lineal para no perder la costumbre

12. Realicen un gráfico de dispersión (*scatter plot*) entre las variables goles totales y tiros al arco totales (*shots*) para **todos los equipos del *dataset***. Analicen y decidan qué variable pondrían en cada eje. ¿Se observa alguna relación entre la cantidad de tiros al arco y la cantidad de goles convertidos? ¿Es la relación que esperaban?

13. Realicen los siguientes ajustes lineales sobre la tendencia observada. Para cada caso realicen un gráfico donde se vean los puntos y la recta/curva ajustada. Revisando las métricas de cada ajuste y **con los conocimientos que poseen hasta ahora**, decidan cuál es el modelo más adecuado para describir la relación.

- goals = a.shots + b
- goals = a.shots + b.shots2 + b
- goals = a.shots + b.shoats2 + c.shots3 + d

* Notarán que hay veces en las cuales no es necesario agregar una ordenada al origen al ajuste, la cual R agrega por defecto cada vez que definimos un modelo lineal. Para poder excluirla (es decir, tener algo como por ejemplo *goals = a.shots + b.shots2*) se debe agregar un -1 al momento de la definición, así:

```
mod <- lm(goals ~ shots + I(shots**2) -1, data = data)
```

Prueben este código con el/los modelo/s que crean necesario.