

# Práctica de Programación en R

Fecha de entrega: 21 de noviembre

## Ejercicio 1 (4 puntos)

Junto con la práctica se entrega un fichero con un *dataset* (**F1\_results.csv**) que contiene información sobre los resultados históricos en las carreras de Fórmula 1. Las variables del *dataset* son las siguientes:

- **resultId**: id del resultado
- **grid\_quali**: posición en clasificación
- **positionText**: posición final
- **positionOrder**: posición orden
- **points**: puntos obtenidos
- **laps**: vueltas realizadas
- **milliseconds**: tiempo en finalizar la carrera
- **fastestLap**: vuelta rápida
- **fastestLapTime**: tiempo de la vuelta rápida
- **fastestLapSpeed**: velocidad máxima de la vuelta rápida
- **statusId**: id del estatus
- **status**: estatus del resultado
- **raceId**: id de la carrera
- **season**: temporada
- **race\_round**: ronda del campeonato
- **race\_name**: nombre de la carrera
- **race\_Date**: fecha de la carrera
- **circuitId**: id del circuito
- **circuitRef**: nombre del circuito
- **circuit\_location**: lugar del circuito
- **circuit\_country**: país del circuito
- **driverId**: id del piloto
- **driverRef**: referencia del piloto
- **driver\_number**: número del piloto
- **driver\_name**: nombre del piloto
- **driver\_surname**: apellido del piloto
- **driver\_nationality**: nacionalidad del piloto
- **constructorId**: id del constructor (equipo)
- **constructor\_name**: nombre del equipo
- **constructor\_nationality**: nacionalidad del equipo

1. Lee el fichero y asígnalo a una variable. **Pista:** cuidado con el encoding...
2. ¿De qué clase es el objeto?
3. ¿Cómo se pueden ver el tipo de cada columna y una muestra de ejemplos?
4. Muestra los primeros 23 registros del *dataset*.
5. Muestra los últimos 19 registros del *dataset*.
6. ¿Cuáles son las dimensiones del *dataset*?
7. ¿Cuáles son los nombres de las variables del *dataset*?
8. Comprueba si alguna de las variables contiene Nas. **Pista:** ¿R siempre los detecta?
9. Las variables *positionText*, *status*, *circuit\_location*, *circuit\_country*, *driver\_nationality*, *constructor\_nationality* y *driver\_number* deberían ser categóricas. Crea un factor con etiquetas para dichas columnas y asígnalo a la columna correspondiente de nuevo. Es decir, el factor deberá tener tantas categorías como valores tiene la variable y cada una de esas categorías debe tener una etiqueta. Recuerda pasar los parámetros correctamente. **Pista:** cuidado con los Nas.
10. Crea una variable booleana, *end\_race*, que indique si el piloto finalizó la carrera con la variable *status* como "Finished"
11. Calcula la media de la columna *laps*.
12. Guarda en un vector (media) la media de las columnas *laps*, *fastestLapSpeed* y *points*. Es decir, el vector "media" deberá tener 3 elementos: el primero conteniendo la media de la columna *laps*, el segundo con la media de la columna *fastestLapSpeed* y el tercero con la media de la columna *points*.
13. ¿Qué variables son numéricas? **PISTA:** utiliza *apply* junto con la función *is.numeric*. Guardarlo en una variable. Cuidado con los ids...
14. Utilizando la variable con el resultado anterior, selecciona aquellas columnas numéricas y calcula la media de aquellas en las que tenga sentido.
15. Selecciona las 31 primeras filas y todas las columnas menos las tres últimas (**emplea índices positivos tanto en filas como en columnas**).
16. Selecciona las 31 primeras filas y todas las columnas menos las tres últimas (**emplea índices negativos tanto en filas como en columnas**).
17. Obtén los cuartiles de la variable *laps*.
18. Obtén los deciles de la variable *laps*.
19. Obtén los estadísticos básicos de todas las variables en un solo comando.
20. ¿En cuántos circuitos se han disputado carreras de Fórmula 1?
21. ¿Cuántas veces un piloto ha conseguido más de 6 puntos?
22. Ordena de mayor a menor los 100 primeros elementos de la variable *fastestLapSpeed*.
23. Ordena el *dataset* por la variable *points* de manera ascendente. Inspecciona los primeros resultados para comprobar que se ha ordenado como se pide.
24. Obtén los índices de los registros para los que el valor de la variable *laps* es superior a la mediana.
25. ¿Cuántas veces ha disputado un piloto español una carrera?
26. ¿Qué nacionalidad es la que más participaciones tiene en una carrera de Fórmula 1? ¿Y las que menos?
27. ¿Qué pilotos no han finalizado una carrera?
28. Comprueba utilizando el *boxplot* si la variable *laps* tiene *outliers*.
29. Pinta un histograma de la variable *fastestLap*.
30. Crea una función (*get\_podium*) que reciba dos parámetros:
  - a. *positionOrder*: posición final en la carrera.

- b. `podiumPlaces`: posición hasta la que se considera que el piloto ha conseguido un podium. El valor por defecto es 3.
- La función debe devolver "PODIUM" si ha finalizado entre las plazas de podium y "NO PODIUM" en caso contrario. Ejemplifica en el código que funciona correctamente con varios casos.
31. Mediante una llamada a una de las funciones `apply` aplica la función anterior a toda la columna `positionOrder`. El resultado obtenido debe ser almacenado en una nueva variable del *data frame* llamada `podium`.
32. Repite el ejercicio anterior usando el paquete `PURRR`.

## Ejercicio 2 (4 puntos)

### Preparación (*tidyr, reshape2*)

1. A partir de las variables `driver_id` y `constructor_id`, crea una nueva `carId` que sea la concatenación de ambas mediante un guion bajo. No borres las columnas `id` y `title` en el proceso.
2. Sobreescribe el valor de las columnas `race_Date` por una columna de tipo fecha.
3. Crea un *long dataset* que contenga las siguientes variables:
  - **resultId**: identificador único de cada carrera.
  - **VARIABLE**: que contiene la lista de variables: `grid_quali`, `positionOrder` y `points`.
  - **VALUE**: los diferentes valores.
4. Realiza el proceso inverso desde el *long dataset* anterior para obtener el *dataset* original con las siguientes columnas:
  - **resultId**
  - **result\_name**
  - **grid\_quali**
  - **points**
  - **positionOrder**

Resuelve el ejercicio 2.3 y 2.4 empleando los paquetes vistos en clase: *tidyr* y *reshape2*.

### Manipulación (*dplyr, data.table*)

1. Calcula el número de victorias totales para cada piloto y ordena el resultado según esta nueva variable de forma descendente. Utiliza `driverRef`.
2. Calcula el número de carreras disputadas y los puntos medios obtenidos por cada piloto en cada circuito y ordena el resultado según esta nueva variable de forma descendente. Utiliza `driverRef` y `circuitRef`.
3. Calcula el número de vueltas totales completadas en cada mes.
4. ¿Cuál es el constructor con mayor número de puntos medios?
5. Restringiéndonos al siglo XXI, ¿cuál es la nacionalidad (de pilotos) que más veces ha logrado finalizar entre los 5 primeros?
6. Calcula las vueltas de la carrera más larga, más corta y la vuelta media en la que se obtiene la vuelta rápida de las carreras posteriores a 2014 y sólo si el piloto ha finalizado la carrera (que `status` sea *Finished*).

Resuelve el ejercicio empleando los paquetes vistos en clase: *dplyr* y *data.table*.

### Ejercicio 3 (2 puntos)

Busca un *dataset* de tu interés, descárgalo y utilizando lo aprendido en clase haz un análisis a tu elección que te permita descubrir **5 insights**.

Trata de que el procesamiento y limpieza de los datos tenga cierta **dificultad**. Se valorará positivamente tanto el procesamiento como el interés de éstos.

El resultado del ejercicio debe ser reproducible mediante **R Markdown**, ocupando un máximo de dos páginas. **Debes entregar tanto el código Markdown como la salida de éste.**

**Reparto de puntos: 0,5 ptos por uso de Markdown, 0,5 ptos por procesamiento, 1 pto por los insights.**

---

**La entrega de la práctica será individual.** Para cada ejercicio entrega un fichero R con el código para realizar cada una de las operaciones. Ten en cuenta que pueden existir diversas formas de realizar los ejercicios. Siempre que se consiga el objetivo se considerará que la respuesta es válida. Ambos ficheros se comprimirán dentro de un fichero zip con tu nombre. Por ejemplo: **jose\_manuel\_sanchez\_madrid.zip**