FUNDAMENTOS: LENGUAJES

Práctica 3: Introducción a R

Esta práctica es una introducción a R. Algunos ejercicios que haremos se podrán realizar de forma más sencilla utilizando módulos que veremos más adelante. Sin embargo, es importante que se realicen en esta práctica utilizando los tipos y funciones básicas vistas durante las sesiones.

Estilo de programación

Es conveniente (y se valorará) utilizar un estilo de programación adecuado. Algunas directrices pueden encontrarse en la Guía de estilo de R de Google:
https://google.github.io/styleguide/Rguide.xml

Ejercicios con vectores

1. Los resultados de una encuesta de satisfacción de clientes han sido los siguientes:
   4, 3, -7, 10, 3, 5, 4, 2.5, 2, 3, 1, -1
   Los valores válidos de respuesta eran enteros de 1 a 5, incluidos. Identifica aquellos valores inválidos en el vector, asignarles NA, y calcular la media de satisfacción de los clientes.

2. Ejecuta el siguiente código en R, que genera dos vectores x e y con 250 números enteros aleatorios entre 1 y 1000:

```
n <- 250
x <- sample(1:1000, n, replace=T)
y <- sample(1:1000, n, replace=T)
```

   A partir de los dos vectores anteriores:
   a) Calcula el máximo y el mínimo de los vectores `x` e `y`.
   b) Calcula la media de los vectores `x` e `y`. Antes de calcularla, ¿qué valor esperarías?
   c) Calcula el número de elementos de `x` divisibles por 2. Pista: Recuerda que el operador módulo es `%%`.
   d) Ordena los vectores primero usando la función `order` y luego la función `sort`. ¿Cuál es la diferencia entre estas dos funciones? ¿Cómo podrías obtener el mismo resultado que `sort` usando únicamente `order`?¿ y el resultado de `order` utilizando la función `sort`?
   e) Selecciona los valores de `y` menores que 600.
   f) Crea el vector
   $$(x_1 + 2x_2 - x_3, \quad x_2 + 2x_3 - x_4, \quad \cdots \quad , x_{n-2} + 2x_{n-1} - x_n)$$
   Donde x_i representa el elemento i-ésimo del vector `x`.
   Pista: el vector a generar tiene longitud n-2.
   g) Crea el vector
   $$\left(\frac{\cos y_1}{\sin x_2}, \quad \frac{\cos y_2}{\sin x_3}, \quad \cdots, \quad \frac{\cos y_{n-1}}{\sin x_n},\right)$$
   Pista: tiene longitud n-1.

Ejercicio con matrices

3. Crea la matriz 4x5 siguiente:

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \end{pmatrix}$$

a) Extrae los elementos A[4,3], A[3,4] y A[2,5] utilizando una matriz de índices.
b) Reemplaza dichos elementos con 0.
c) Crea la matriz identidad 5x5,

$$I = \begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{pmatrix}$$

Pista: mira la documentación de la función diag().
d) Convierte la matriz A anterior en una matriz cuadrada B añadiendo al final una fila de unos (función rbind):

$$B = \begin{pmatrix} A \\ 1 \end{pmatrix}$$

e) Calcula la inversa de la matriz B con la función solve.
f) Multiplica B por su inversa B^{-1}.
g) Comprueba si el resultado es exactamente la matriz identidad I.
h) En caso contrario, calcular el "error" o "precisión" de la operación, definido como:

$$Error = \frac{1}{N}\sum_{i,j}\left|(BB^{-1} - I)_{(i,j)}\right|$$

Donde N es el número de elementos de la matriz B. La notación (i,j) representa el elemento de la matriz en la i-ésima fila y la j-ésima columna, donde i toma valores desde 1 al número de filas y j toma valores desde 1 al número de columnas.

Ejercicios con *dataframes*

4. Con el *dataframe* **mtcars** (viene cargado en R y puedes leer su descripción con `?mtcars`):
   a) Previsualiza el contenido con la función `head`.
   b) Mira el número de filas y columnas con `nrow` y `ncol`.
   c) Crea un nuevo *dataframe* con los modelos de coche que consumen menos de 15 millas/galón (`mpg`).
   d) Ordena el *dataframe* del apartado c) por cilindrada (`disp`.).
   e) Calcula la media del peso (`wt`) de los modelos del *dataframe* del apartado c).
   f) Cambia el nombre de las variables del *dataframe* del apartado c) a `var1, var2, …., var11`. Pista: Mira la documentación de la función `paste` y úsala para generar el vector (`"var1", "var2", …, "varN"`), donde N es el número de variables del *dataframe*.

5. Con el *dataframe* **iris** (viene cargado en R y puedes leer su descripción con `?iris`):
   a) ¿Cómo está estructurado el *dataframe*? Utilizar las funciones `str` y `dim`.
   b) ¿De qué tipo es cada una de las variables del *dataframe*?
   c) Utilizar la función `summary` para obtener un resumen de los estadísticos de las variables.
   d) Comprueba con las funciones `mean` y `range` que se obtienen los mismos valores para cada una de las variables. Pista: usar funciones tipo *apply*.
   e) Cambia los valores de las variables `Sepal.Length` y `Sepal.Width` de las 5 primeras observaciones por `NA`.
   f) ¿Qué pasa si usamos ahora las funciones `mean` y `range` con las variables `Sepal.Length` y `Sepal.Width`? ¿Tiene el mismo problema la función `summary`?
   g) ¿Qué parámetro habría que cambiar para arreglar el problema anterior?
   h) Visto lo anterior, ¿por qué es importante codificar los *missing values* como `NA` y no como 0, por ejemplo?

i) Ordena el *dataframe* por la columna `Sepal.Length` con la función `order` ¿qué sucede con los *missing values*? Pista: Mirar la documentación de la función `order`.
j) Elimina los valores `NA` usando `na.omit`.
k) Calcula la media de la variable `Petal.Length` para cada una de las distintas especies (`Species`) de dos formas diferentes usando el *dataframe* obtenido del apartado j.

Ejercicio 6: Procesado y limpieza de datos con R

El conjunto de datos `titanic` contiene información sobre los pasajeros del barco. Este conjunto de datos se ha utilizado para tratar de predecir la supervivencia de un pasajero en base a otra serie de variables como edad, sexo, o la clase del billete. Ver por ejemplo: https://www.kaggle.com/c/titanic.

El conjunto de datos se proporciona en el fichero **titanic.csv** junto con el enunciado de la práctica[1]. Cada una de las variables del fichero contiene la siguiente información:

- **pclass**: Clase del pasajero. (1 = primera clase; 2 = segunda clase; 3 = tercera clase)
- **survived**: Supervivencia (0 = No; 1 = Sí)
- **name**: Nombre del pasajero
- **sex**: Sexo del pasajero
- **age**: Edad del pasajero. La edad está en años, salvo para edades menores a un año, que contienen un número fraccional correspondiente al número de meses.
- **sibsp**: Número de hermanos/cónyuge a bordo
- **parch**: Número de padres/hijos a bordo
- **ticket**: Número de billete
- **fare**: Precio del billete
- **cabin**: Cabina
- **embarked**: Puerto de embarque (C = Cherbourg; Q = Queenstown; S = Southampton)

En los siguientes ejercicios vamos a realizar un análisis descriptivo de la información contenida en este conjunto de datos.

a) Leer el fichero `titanic.csv` como un *dataframe*. Los campos de este fichero se encuentran separados por punto y coma (;), los *missing values* se codifican como cadena vacía, y los separadores decimales vienen indicador con coma. Para la lectura del fichero, también es necesario indicar que no existe un delimitador especial para cadenas de caracteres. Esto se hace con la opción `quote=""`. No codificar las cadenas de caracteres como factores, dado que esto no tiene sentido para algunas variables como `name`.
b) Calcular el porcentaje de pasajeros que sobrevivió.
c) Calcular el porcentaje de missing values en cada uno de los atributos. Pista: averiguar qué devuelve la función is.na cuando se aplica a un *dataframe*.
d) Eliminar la variable `cabin` del dataframe.
e) Completar los *missing values* del atributo `age` con la mediana del resto de datos. Pista: función median.
f) Calcular la probabilidad de supervivencia en base al género. ¿Qué conclusión(es) obtienes del resultado? Pista: dado que la variable `survived` toma el valor 1 cuando el pasajero sobrevivió, es sencillo calcular la probabilidad de supervivencia con una función estadística.
g) Calcular la probabilidad de supervivencia en base a la edad. ¿Te parecen fácilmente interpretables estos resultados?
h) Crea una nueva variable `decade` en el *dataframe* que contenga la década de la edad de los pasajeros y repite el análisis del apartado g) sobre esta nueva variable. ¿Qué conclusión(es) obtienes del resultado?
i) Calcula la probabilidad de supervivencia en base a la clase del billete del pasajero (`pclass`). ¿Qué conclusión(es) obtienes del resultado?
j) Combina en una tabla el análisis de la probabilidad de supervivencia en base al sexo y clase del billete del pasajero. ¿Qué conclusiones se obtienen?
k) Crea dos nuevas variables en el *dataframe* con la siguiente información:
    - **familysize**: número total de parientes incluyendo al propio pasajero.
    - **sigleton**: valor lógico indicando con valor `TRUE` si el pasajero viaja solo y `FALSE` en caso contrario.

[1] Datos procedentes de: biostat.mc.vanderbilt.edu/wiki/pub/Main/DataSets/**titanic**3.xls

l) Calcula la probabilidad de supervivencia en base a si el pasajero viajaba solo o no. ¿Qué conclusión se puede obtener?
m) Cuenta el número de pasajeros por tamaño de familia y clase. Por ejemplo, cuántos pasajeros de primera clase pertenecen a una familia de tamaño 4. El resultado debe ser una matriz con la información para todas las posibles combinaciones de clase del billete y tamaño de familia.
n) El fichero **titanic2.csv** contiene información adicional sobre los pasajeros del barco:

- **boat**: identificador del bote salvavidas
- **body**: identificador del cuerpo
- **home.dest**: Origen/destino

Leer este fichero (con el mismo tipo de formato que titanic.csv).
Para unificar estos dos *datraframes*, parecería buena opción utilizar la variable `name` como clave. Determina si esta variable es única por pasajero utilizando la función `duplicated` y mostrando el número de nombres diferentes repetidos. En caso de existir varios pasajeros con el mismo nombre, listar aquellas filas del *dataframe* inicial en las que el nombre del pasajero esté repetido (la función `%in%` puede resultar útil para ello). De acuerdo a los resultados, ¿sería la combinación del nombre del pasajero y la clase una buena clave para combinar los *dataframes*?
o) Combina ambos *dataframes* utilizando la combinación del nombre y el número de billete respetando el orden de los *dataframes* de partida.
p) ¿Qué porcentaje de los pasajeros que sobrevivió tiene asociado un identificador del bote salvavidas?
q) Guarda la información de las variables `name`, `age`, `sex`, `ticket`, `pclass`, `boat` por este orden para los pasajeros supervivientes en el fichero **titanic_all.csv** usando como separador de columnas el tabulador (`\t`), indicando las cadenas de caracteres con dobles comillas, usando el punto como separador decimal, y mostrando el nombre de las columnas y las filas.

Entrega

Se deberá entregar la práctica de acuerdo al calendario de entregas del Máster. La entrega se realizará a través de la página del curso y será un fichero zip que contenga 6 ficheros .R, uno con cada ejercicio. El nombre del fichero .zip debe ser `P3_<apellidos>.zip`.

Se valorará tanto el correcto funcionamiento del código como su generalidad y estilo. Incluir comentarios en el código siempre que se considere necesario. Las respuestas planteadas a las preguntas deben responderse como comentarios en el fichero .R después de la línea de código correspondiente.