Miguel Ángel Castaño Ibáñez

Práctica de Programación R

Pregunta 1 (20 puntos): Un boleto del sorteo de la ONCE consta de dos partes, la primera es un número de 4 dígitos y la segunda es un número de tres dígitos que forman la serie del boleto.

Aquí consideramos sólo el número, por ejemplo,

| 0| 2 | 0 | 9 |

a) **Genera todos los números que entran en el sorteo de la ONCE y mostrarlos con los cuatro dígitos.**

   En este primer apartado que encontramos en el fichero *practica.R* usaremos la librería *gtools* donde encontraremos la función *permutations()*. Nos permitirá crear una matriz con todas las combinaciones posibles de 4 cifras, con los números de 0 a 9. Todo el calculo necesario se hará en la función *generateNumbersMatrix(initNum, lastNum, rows)*, que es la encargada de devolver la matriz con todas las combinaciones.

   combinations(initNum, lastNum, rows), esta función nos devuelve el numero total de combinaciones posibles, que serán una matriz de 4 x 10000, es decir se pueden dar 10000 combinaciones.

b) **¿Cuál es la suma de los números de un boleto que más se repite?**

Para este apartado hemos ceado 3 funciones:

- *vectorNumRep(matrix):* esta función es la encargada de generar un vector con todas las veces que se repite cada suma coincidiendo la suma con la posición. La suma máxima sería 9 x 4 = 36.

- *maxNumRep(vector):* esta función se usa para recuperar la suma mas repetida.

- *sumRepeated(times, vector):* finalmente con esta función recuperaremos la posicion de la suma mas repetida

La suma mas repetida es 18, mientras que se ha repetido 670 veces. Entre estas combinaciones podemos encontrar: 0099, 1188, 4455, 7722, etc.

Pregunta 2 (20 puntos): En la carpeta covid_19 hay una serie de archivos sobre el covid-19 en España.

a) **Leer los archivos "datos_provincias.csv", "CodProv.txt* y "CodCCAA.dat ". Añade el código de la comunidad autónoma al fichero "datos_provincias.csv" (no manualmente).**

   Este apartado podemos diferenciarlos en dos partes, la primera para leer los ficheros con las funciones *read.csv()* (para los archivos en .CSV) y *read.table()* (para los ficheros en texto plano)

   A continuación, quitamos el la parte *"ES – " de* cada código de comunidad autónoma y provincia y finalmente hacemos un *merge()* de ambos dataframes por el código de la provincia.

b) **Selecciona los datos de la comunidad autónoma que te corresponda. Para saber cuál es tu comunidad autónoma realiza la siguiente operación: DNI o Pasaporte mod 17 por ejemplo (12345678 %% 17 = 6 → Castilla y León) hay que seleccionar las provincias de Castilla y León.**

   En este apartado usamos la función *fliter()*, para tras calcular el numero de CCAA obtener la correspondiente, de esta forma la podemos obtener de forma genérica, sin tener que mirar el resultado del modulo y buscarla a continuación en el dataframe.

   La CCAA obtenida es la 8, Extremadura.

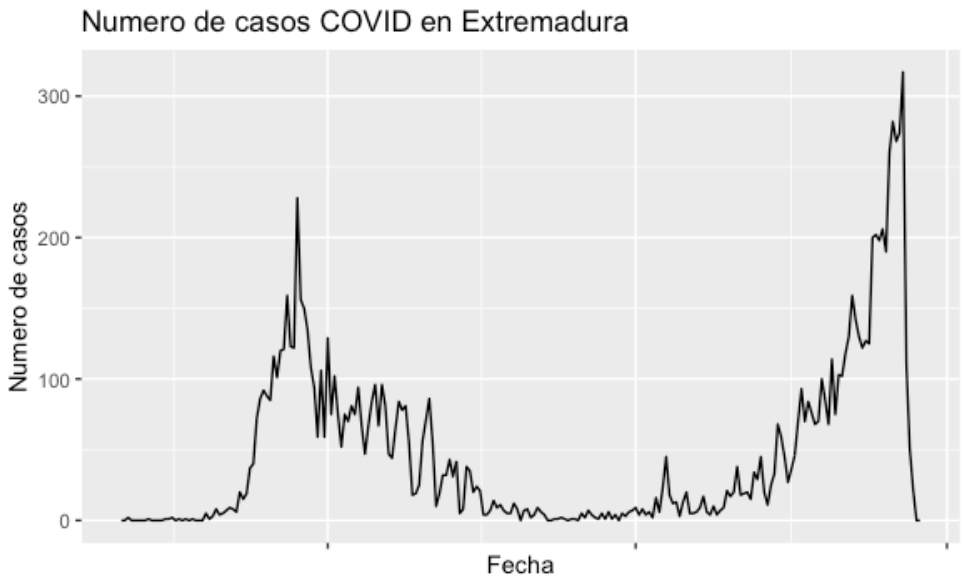
   A continuación, usamos de nuevo *filter(),* para obtener solo las provincias de Extremadura y posteriormente ordenarlos, que aun que no es necesario para hacer un *plot*, si que es conveniente para verla en el dataframe.

   Finalmente, pasamos la variable fecha que es una cadena, a formato timestamp y agrupamos por fecha todas las provincias sumando los casos para poder pintar el grafico de la comunidad en los siguientes apartados.

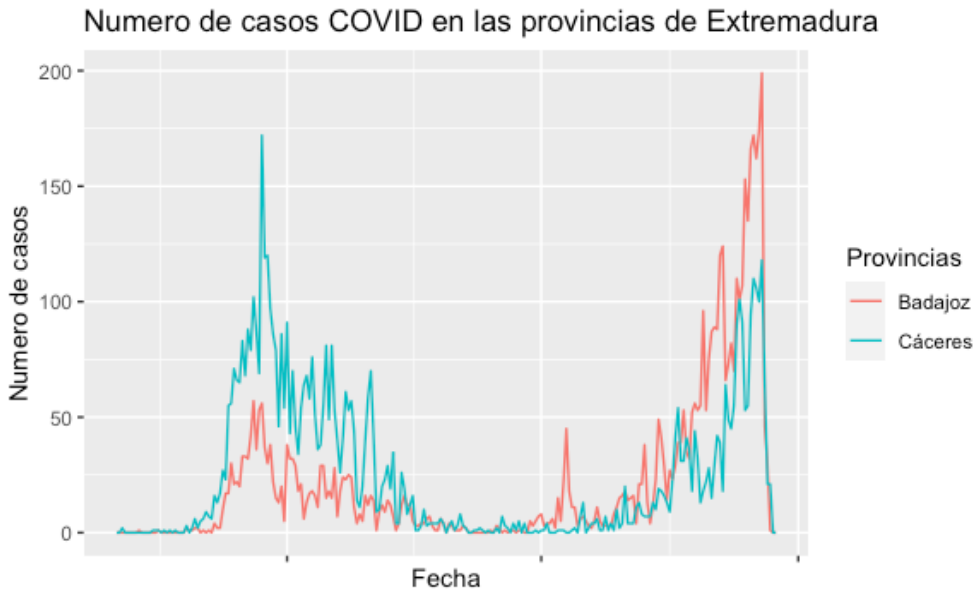
c) **hay que seleccionar las provinci) Realizar un gráfico que muestre adecuadamente la evolución de los casos nuevos. Justifica el gráfico elegido.**

   En este apartado hemos utilizado la librería *ggplot2,* la mas famosa para pintar funciones en R de forma sencilla.

   En el primer gráfico vemos la evolución de los casos en la comunidad autónoma de Extremadura. Podemos apreciar las dos oleadas de coronavirus, en marzo-abril y esta ultima octubre-noviembre.



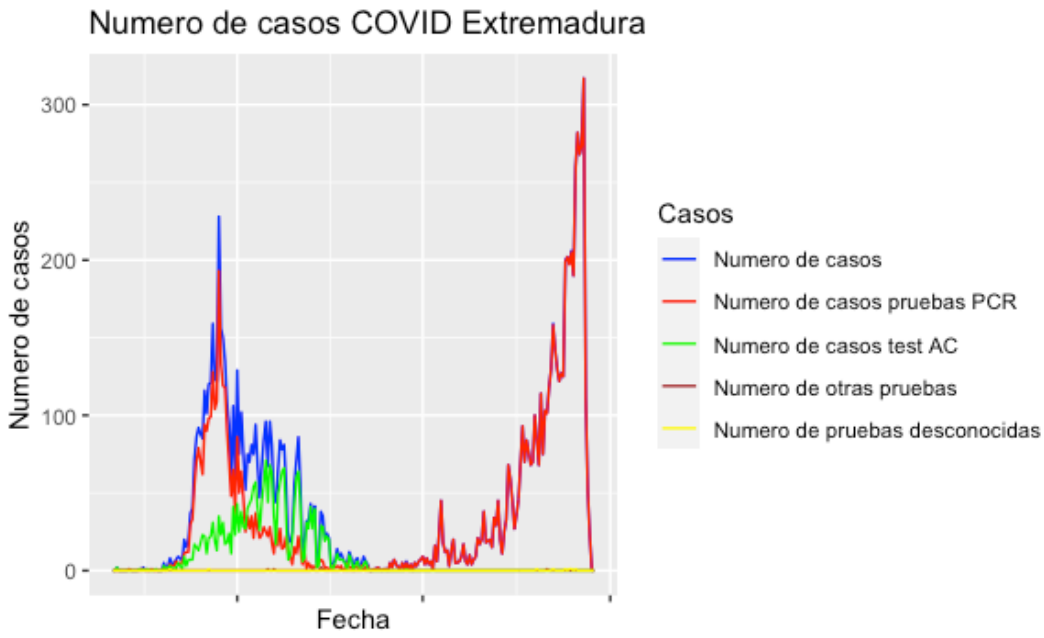
Seguidamente ya que Extremadura tiene solo dos provincias he querido pintar la evolución de los casos, separándolos en provincias, donde podemos apreciar que mientras Badajoz fue la menos afectada en la primera ola, en esta segunda esta siendo la mas afectada, aun que actualmente vemos como han bajado los casos, debido a que estos últimos días puedan ser datos provisionales, todavía sin confirmar.



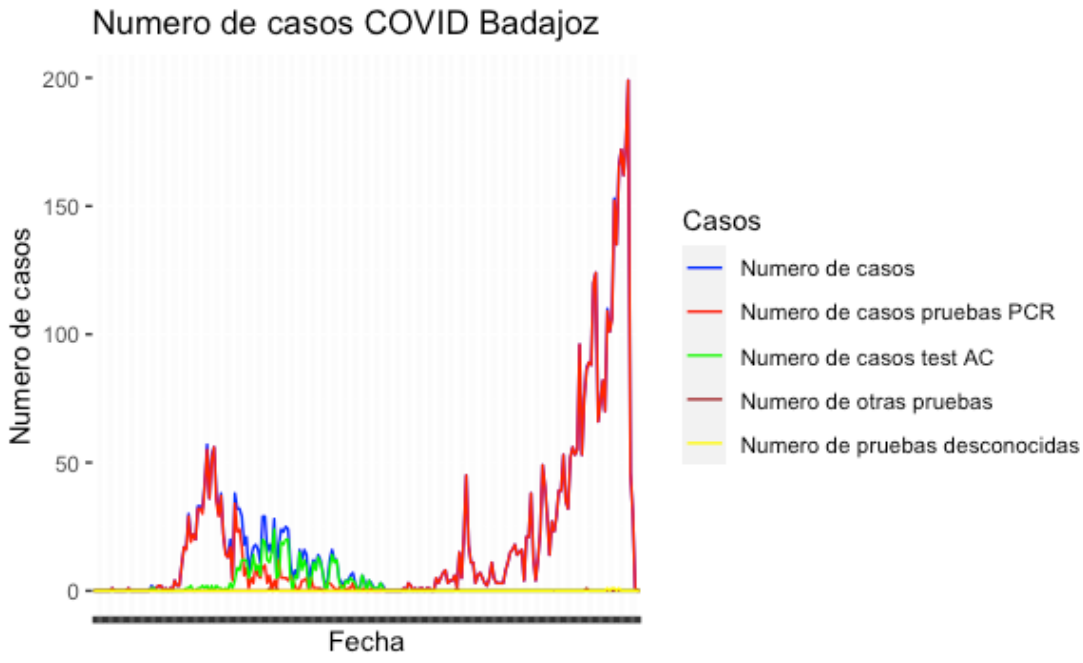
d) Presenta en único gráfico la evolución de las distintas variables (columnas) por medio de un gráfico de líneas múltiples. Utiliza diferentes colores y añade una leyenda muestre el origen de cada línea.

En este apartado utilizamos la función *melt()*, para conseguir una matriz de solo 3 columnas, y poder agrupar los números de casos mas fácilmente. Esta función se encuentra en la librería *reshape2*.

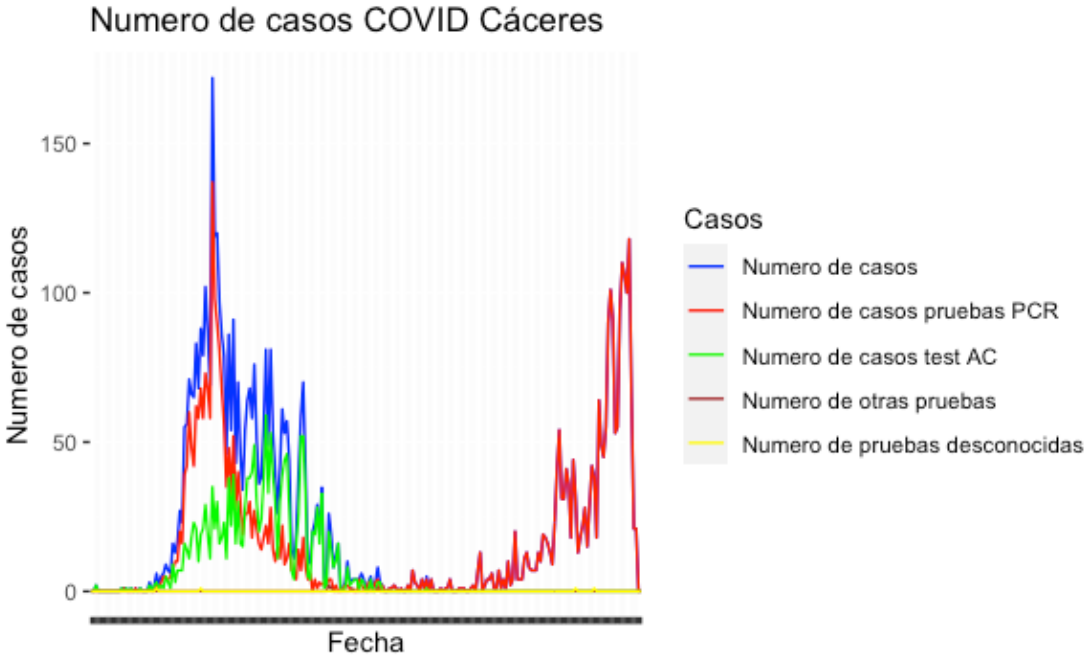
Aquí podemos ver a continuación, el numero de casos en Extremadura, las pruebas pcr, los test de anticuerpos, otro tipo de pruebas, y pruebas desconocidas. Podemos apreciar como en esta segunda oleada solo las pruebas pcr han sido las utilizadas par diagnosticar COVID



Aquí podemos ver a continuación, el numero de casos en solo en Badajoz, una vez más, las pruebas pcr, los test de anticuerpos, otro tipo de pruebas, y pruebas desconocidas. Obtenemos las mismas conclusiones que para toda la CCAA



Aquí podemos ver a continuación, el numero de casos en solo en Cáceres, una vez más, las pruebas pcr, los test de anticuerpos, otro tipo de pruebas, y pruebas desconocidas. Obtenemos las mismas conclusiones que para toda la CCAA



Pregunta 3 (20 puntos): Consideramos un fichero de datos en formato SAS de nombre "punt.sas7bdat" que contiene datos sobre alumnos matriculados en diversos cursos. Las variables son:

a) Importa el fichero de datos y guárdalo en un objeto de nombre punt. Comprueba la estructura del objeto punt. Si es necesario conviértelo en un data frame.

Gracias a la librería *sas7bdat* podemos leer mas fácilmente las extensiones .sas7bdat que directamente la almacena como un dataframe.

b) Obtener una nueva variable overall que de la puntuación media de los cuatro test para cada estudiante suponiendo que el último test se pondera el doble.

En este apartado añadimos la columna overall con una ponderación de 0.2 cada examen y 0.4 el ultimo.

c) Formar una nueva variable denominada start compuesta por el mes y día de ENROLLED y por el año corriente y presenta en pantalla las variables SEGSOC, COURSE y star.

Utilizamos la función un *substr()* para coger el año, ya que conocemos el formato, pero creo que seria mas conveniente utilizar una librería que recupere el año, independientemente del formato. Pero he usado esta, ya que en este caso solo tenemos el año actual "2020". A continuación, con la función paste añadimos el año a *ENROLLED*

y lo colocamos en otra columna de nuestro dataframe: *start*. Finalmente, "printeamos" el nuevo dataframe con las tres columnas exigidas: *SEGSOC, COURSE* y *statr.*

d) Formar un nuevo dataframe de nombre level500 que contenga los estudiantes cuyo curso acaba en 500. Crear dos nuevas variables carácter, una de nombre subject con el código de curso (parte literal) y otra de nombre level con el número del curso (parte numérica).

En este apartado usamos la librería *dplyr* para usar la función *filter(),* la cual usamos para poder flitrar todo los elementos donde el curso acaba en 500.
Primero la funciones *substrLit()* separa la parte literal cogiendo el subString de los tres primeros caracteres de *COURSE.*

En segundo lugar la función *substrNum()* tomara la parte numérica de *COURSE*, es decir a partir del tercer carácter, no inclusive.

Finalmente, *filter()* filtrará las los elementos donde *COURSE* sea igual a 500

e) Escribe la información de level500 en fichero ASCII de nombre "level500.dat".

En este apartado guardamos el dataframe resultante de lapartadoanterior en el directorio de trabajo de mi ordenador:
setwd("/Users/macasib/Desktop/UCM/Programación\ R/TareaEvaluacion")

Usamos la librería *haven* para utilizar la funcion *write_dta()* la cual escribe en un fichero de texto plano ASCII el dataframe, con la extensión .dta, en el directorio de trabajo.

Pregunta 4 (40 puntos): La siguiente tabla representa puntuaciones de sensación de ardor para 16 sujetos en un estudio para probar un nuevo hidrogel. La primera columna da el número del sujeto. Las siguientes columnas dan la puntuación de sensación de ardor (en una escala de 1 a 4) para semanas 1 (S1) a 7 (S7). (La matriz de datos se encuentra en "matriz.R")

1- Para la semana S_7, calcule el vector (f_1, 1 − f_1, f_2, 1 − f_2, f_3, 1 − f_3, f_4, 1 − f_4) donde f_i es la frecuencia de la modalidad $i \in \{1,2,3,4\}$ observada en la semana S_7 sobre los 16 sujetos. (Sugerencia: use las funciones `tabulate()`, `cbind()`, `t()` y `as.vector()`)

En este apartado creamos la función *frequencyVector()* donde a partir de la matriz y el máximo nivel de ardor nos devolverá el vector de la semana 7 en el formato indicado. Esta máxima sensación de ardor la obtenemos con la funcion max() aplicándosela a la matriz en todos los casos de ardor, podríamos calcularlo internamente dentro de la función, pero para los siguiente apartados, nos interesa saber la sensación máxima de ardor de toda la matriz.

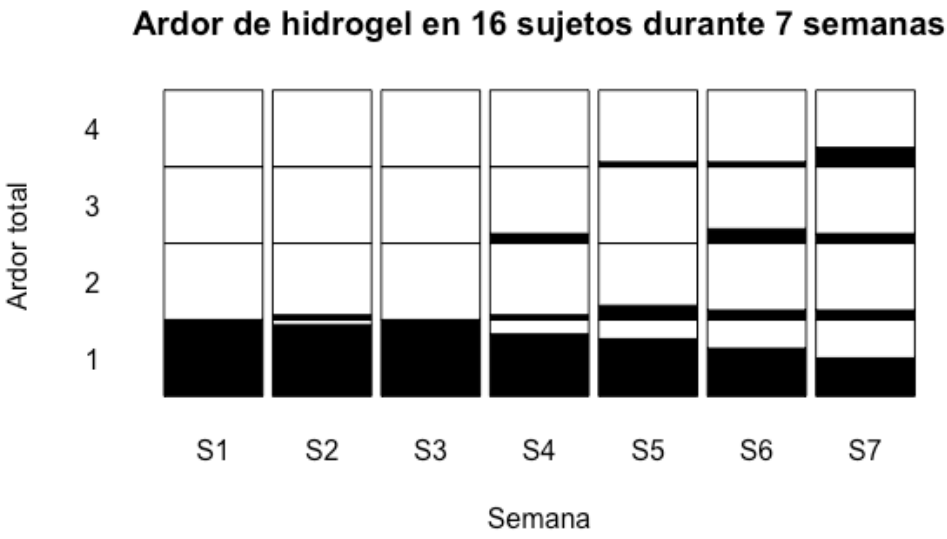
(f_1, 1 − f_1, f_2, 1 − f_2, f_3, 1 − f_3, f_4, 1 − f_4) = (8, 8, 2, 14, 2, 14, 4, 12)

2- Ahora, use la función apply() para hacer el mismo cálculo para todas las demás semanas. Almacene el resultado en una matriz.

En este apartado, aplicaremos la función *frequencyVector()* por columnas haciendo uso de la con la función *aply().*

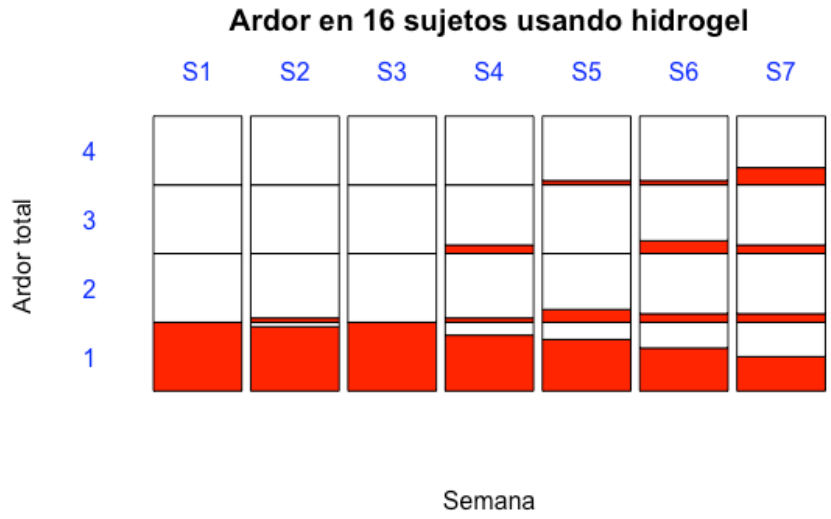
3- Utilice la función barplot() y el argumento col = c ("black", "white") en esta matriz. El gráfico que se obtiene ofrece una descripción general de la evolución de la Sensación de ardor con el tiempo.

Esta primera gráfica ha sido pintada gracias a la función *barplor()* de R base, con la secuencia calculada gracias a la función *seq(8,16*4, by =16)* calcularemos la posición de las etiquetas en el eje y, añadida posteriormente en la función Axis.



4- Cambie el gráfico anterior para que las barras que representan las frecuencias estén en rojo. Los números de las semanas deben estar en azul y en la parte superior del gráfico en lugar del fondo. Los números de modalidad deben estar a la izquierda, en azul. Agrega un título al gráfico

En esta segunda gráfica ha sido pintada de la misma manera, a excepción que para poder pintar las semanas arriba hemos tenido que calcular la posición con el algoritmo implementado en la función *calculateSpaceH()* que en base a las anchura y espacio entre barras definida en la función barplot(width = 1, space = 0.1).



Conclusiones

Tras esta practica he podido aprender y fortificar los conocimientos vistos de R en mi grado de ingeniería informática. Considero este, un lenguaje bastante cómodo y perfecto para el análisis de datos. Creo quizás que es más intuitivo que Python, e incluso su alto nivel hace que sea mas fácil de aprender. A pesar de ello, prefiero un lenguaje como Python para programar algoritmos ya que estoy mas acostumbrado a su uso, aunque considero que quizás este lenguaje sea mas idóneo para el manejo de dataframes.