

October 30, 2020

1 Fundamentos de Programación

1.1 PEC 4 - Enunciado

En este Notebook se encontraréis el conjunto de actividades evaluables como PEC de la asignatura. Veréis que cada una de ellas tiene asociada una puntuación, que indica el peso que tiene la actividad sobre la nota final de la PEC. Adicionalmente, hay un ejercicio opcional, que no tiene puntuación dentro de la PEC, pero que se valora al final del semestre de cara a conceder las matrículas de honor y redondear las notas finales. Podréis sacar la máxima nota de la PAC sin necesidad de hacer este ejercicio. El objetivo de este ejercicio es que sirva como pequeño reto para los estudiantes que quieran profundizar en el contenido de la asignatura.

Veréis que todas las actividades de la PEC tienen una etiqueta, que indica los recursos necesarios para llevarla a cabo. Hay tres posibles etiquetas:

- **NM Sólo materiales:** las herramientas necesarias para realizar la actividad se pueden encontrar en los materiales de la asignatura.
- **EG Consulta externa guiada:** la actividad puede requerir hacer uso de herramientas que no se encuentran en los materiales de la asignatura, pero el enunciado contiene indicaciones de dónde o cómo encontrar la información adicional necesaria para resolver la actividad.
- **EI Consulta externa independiente:** la actividad puede requerir hacer uso de herramientas que no se encuentran en los materiales de la asignatura, y el enunciado puede no incluir la descripción de dónde o cómo encontrar esta información adicional. Será necesario que el estudiante busque esta información utilizando los recursos que se han explicado en la asignatura.

Es importante notar que estas etiquetas no indican el nivel de dificultad del ejercicio, sino únicamente la necesidad de consulta de documentación externa para su resolución. Además, recordad que las **etiquetas son informativas**, pero podréis consultar referencias externas en cualquier momento (aunque no se indique explícitamente) o puede ser que podáis hacer una actividad sin consultar ningún tipo de documentación. Por ejemplo, para resolver una actividad que sólo requiera los materiales de la asignatura, podéis consultar referencias externas si queréis, ya sea tanto para ayudaros en la resolución como para ampliar el conocimiento!

En cuanto a la consulta de documentación externa en la resolución de los ejercicios, recordad **citar siempre la bibliografía utilizada** para resolver cada actividad.

1.2 Ejercicios para la PEC

NumPy - Ejercicios

1.2.1 Ejercicio 1

Crea y muestra por pantalla las matrices siguientes: **(0.25 puntos)**

- Una matriz 3x3 que contenga todos los enteros entre 0 y 8.
- La matriz identidad 3x3.
- Una matriz 5x3 con valores reales aleatorios (*Random*).

Considerando las matrices del apartado anterior, muestra por pantalla el resultado de las siguientes operaciones: **(0.25 puntos)**

- Multiplicar las matrices A y B. Explica el resultado obtenido.
- Multiplicar las matrices A y C. Explica brevemente qué método de multiplicación has elegido y el resultado obtenido.

(0.5 puntos) NM

```
[ ]: # Respuesta
```

1.2.2 Ejercicio 2

Crea un array de NumPy 1D de 11 elementos (los enteros de 0 a 10) ordenados ascendentemente. A continuación, pon en negativo todos los elementos que estén entre 3 y 8, y muestra el array resultante.

(0.5 puntos) NM

```
[ ]: # Respuesta
```

1.2.3 Ejercicio 3

Crea una matriz de 7x7 donde los valores de cada posición (i, j) correspondan a $(i + 1) \wedge - j + 2$ para todo i múltiplo de 3 y $(i + 1) \wedge j - 3$ por el resto de posiciones.

(0.5 puntos) NM

```
[ ]: # Respuesta
```

1.2.4 Ejercicio 4

Normaliza una matriz aleatoria de 5x5 mediante su media y desviación estándar tal como se indica en el siguiente [enlace](#). Comprueba que la media y la desviación estándar de la matriz normalizada son bastante cercanas a 0 y 1 respectivamente.

Nota: Para resolver este ejercicio os pueden ser de utilidad las funciones [mean](#) o [std](#) de Numpy.

(0.5 puntos) EG

```
[ ]: # Respuesta
```

1.2.5 Ejercicio 5

Crea una matriz 10x10 con 1 en el borde exterior y 0 en el interior.

Nota: Podéis comprobar el resultado visualmente con la función [imshow](#) de Matplotlib que nos permite visualizar una imagen 2D.

(1 punto) NM

```
[ ]: # Respuesta
```

Matplotlib - Ejercicios

1.2.6 Ejercicio 1

Representa en un único gráfico las funciones logarítmicas $\log_{10}(x)$, $\log_2(x)$ y $\ln(x)$, evaluadas en el intervalo $[0.1, 5]$ y con paso (resolución) 0.02. Haz que los rangos de valores mostrados sean $[0.1, 5]$ para el eje X y $[-3, 3]$ por el eje Y. Adicionalmente, muestra la gráfica correspondiente a $\log_{10}(x)$ con color verde, la gráfica correspondiente a $\ln(x)$ con color rojo y la gráfica correspondiente a $\log_2(x)$ con color azul.

Nota: Podéis consultar la documentación de [logaritmos](#) de numpy.

(0.5 puntos) NM

```
[8]: # Definimos las funciones
def f1(x):
    return np.log10(x)

def f2(x):
    return np.log(x)

def f3(x):
    return np.log2(x)
```

```
[ ]: # Respuesta
```

1.2.7 Ejercicio 2

Carga los datos del archivo `prog_book.csv`, que puedes encontrar en la carpeta `data`, en forma de dataframe. Este conjunto de datos de la web [Kaggle](#) recoge información de una lista de 270 libros de temas relacionados con la informática y la programación.

Parte A (0.5 puntos) :

Muestra el histograma de las valoraciones (*Rating*).

Consideraciones

- El histograma debe tener 20 bins y ser de color verde.

Parte B (0.5 puntos) :

Muestra un gráfico de puntos (*scatter plot*) de las valoraciones en función del precio (*Price*). Diferencia los puntos en función del tipo de libro (*Type*).

Consideraciones

- Para pintar de un color diferente cada punto en función del tipo de libro, puede que tengas que pasar esta información a numérico (e.g. Libro = 1, Ebook = 2, etc). Una opción para hacerlo es utilizar la función [LabelEncoder](#) de sklearn.
- Incluye la leyenda del *scatter plot*.

Parte C (0.5 puntos) :

Muestra un gráfico circular (*pie chart*) con los tipos de libros en porcentaje.

Consideraciones

- Debes incluir el porcentaje (%) de cada tipo de libro dentro de su respectivo espacio del gráfico.
- Se debe mostrar la leyenda en la parte derecha.

Nota: En el Notebook de teoría hemos visto cómo mostrar gráficos lineales simples. Matplotlib nos ofrece muchos otros tipos de gráficos, siendo los gráficos circulares (*pie charts*) o los *scatter plots* uno de ellos. Para resolver este ejercicio podéis consultar la documentación oficial de matplotlib referente a [pie charts](#) y los [scatter plots](#) para ver cómo se trabaja con estos tipos de gráficos.

(1.5 puntos) EG

```
[ ]: # Respuesta
```

pandas - Ejercicios

1.2.8 Ejercicio 1

Carga los datos del archivo `basket_players_stats.csv`, que puedes encontrar en la carpeta `data`, en un *dataframe*. Este conjunto de datos recoge información sobre jugadores de [baloncesto](#) entre 1999-2020.

Muestra las variables del dataset y el número de entradas (filas). Hay información duplicada?

(0.5 puntos) NM

```
[ ]: # Respuesta
```

1.2.9 Ejercicio 2

La NBA (National Basketball Association) es la liga más famosa de baloncesto, pero en todo el mundo se organizan muchas otras. Contesta y muestra el resultado de las siguientes preguntas:

- Hay información de ligas diferentes a la NBA? En el caso de ser así, de cuántas ligas en total disponemos de información (incluyendo la NBA)? **(0.5 puntos)**
- Agrupa los datos cargados por liga (*League*), y muestra el número total de filas por cada una. De qué liga tenemos más información? Muestra los 10 primeros resultados ordenados de mayor a menor. **(0.5 puntos)**

Nota: En el Notebook de teoría hemos visto cómo calcular la media de una agrupación de datos. Para resolver este ejercicio necesitaréis investigar cómo contar (`count`) todos los valores de una columna en una agrupación. Os puede ser de utilidad la función `aggregate` de pandas, que nos permite aplicar diferentes funciones a una columna en un `groupby`.

(1 punto) EG

```
[ ]: # Respuesta
```

1.2.10 Ejercicio 3

A continuación nos centraremos en la temporada 2018-2019 de la liga de la NBA.

- Crea un nuevo dataset con sólo estos datos. **(0.5 puntos)**
- ¿Qué jugador ha jugado más minutos (*MIN*) de la temporada regular (*Regular_Season*). Y de la temporada de *PlayOffs*? **(0.5 puntos)**

(1 punto) NM

```
[ ]: # Respuesta
```

1.2.11 Ejercicio 4

Los triples (tiros de 3 puntos) son un elemento clave en un partido de baloncesto. Los equipos que tienen buenos tiradores tienen una gran ventaja respecto a los equipos que no.

- Cuál es el promedio de triples (*3PM*) de los jugadores de la temporada Regular 2018-2019? **(0.5 puntos)**
- Qué equipos tienen la media de triples por encima del valor calculado en el apartado anterior? **(0.5 puntos)**

Nota: Además del contenido de los notebooks de teoría, podéis utilizar lo que habéis aprendido haciendo el ejercicio 3.

(1 punto) NM

```
[ ]: # Respuesta
```

1.2.12 Ejercicio 5

Considere el dataset de la temporada 2018-2019 de la liga de la NBA regular (no PlayOffs).

- Crea una nueva columna al dataset que nos indicará por cada fila si el jugador es de un equipo que ha participado en los PlayOffs (0 si no ha participado, 1 si ha participado). **(0.5 puntos)**
- Muestra los histogramas de las características `height_cm` y `weight_kg` separados en función de la nueva columna de PlayOffs. Los jugadores de la NBA que han pasado a PlayOffs tienen características diferentes a los jugadores de los equipos que no han participado en los PlayOffs? **(0.25 puntos)**
- Hay alguna relación entre la altura y el número de rebotes (*REB*) de un jugador? ¿Hay alguna diferencia entre los jugadores que llegan a PlayOffs y los que no? Responde gráficamente esta pregunta. **(0.25 puntos)**

(1 punto) EG

```
[ ]: # Respuesta
```

SciPy - Ejercicio

1.2.13 Ejercicio 1

Python es un lenguaje de programación muy utilizado en el procesamiento de imágenes. En este ejercicio trabajaremos con una imagen de un mapache (*raccoon*), en color y en blanco y negro.

- Carga y muestra las imágenes `raccoon.png` y `raccoonBW.png` que se encuentran en la carpeta `data`. Qué tamaño tiene cada una? Razona la respuesta. **(0.25 puntos)**
- Aplica un filtro gaussiano con tamaño de kernel 3 y 5 a la imagen en blanco y negro. ¿Qué diferencia ves? Y si aplicas un filtro de mediana? Razona la respuesta. **(0.25 puntos)**

Nota 1: Explorad la opción `cmap` de la función `imshow` () para ver la imagen en blanco y negro.

Nota 2: En el notebook de teoría hemos visto el filtro gaussiano (*gaussian_filter*). En este ejercicio también debemos aplicar el [filtro de mediana](#) de la librería `scipy`.

(0.5 puntos) EG

```
[ ]: # Respuesta
```

1.2.14 Ejercicio Opcional

En este ejercicio queremos comparar la liga regular de NBA y la ACB en la temporada 2018-2019. Genera y muestra un gráfico de barras con el número de jugadores de cada nacionalidad de ambas ligas. El

Consideraciones:

- Hay casos de jugadores con doble nacionalidad. Para facilitar la visualización, considera sólo la **primera nacionalidad**.

- Muestra los dos grupos (NBA regular y ACB) en gráficos separados.
- Debes mostrar los datos de manera horizontal (barras horizontales).
- El eje de las Y debe incluir las etiquetas de las nacionalidades.
- Debes incluir un texto con el número de jugadores cada categoría del gráfico. El texto debería estar situado a la derecha de cada barra.

Nota 1: Para considerar sólo una nacionalidad debéis hacer alguna modificación del texto. Consultad la siguiente página para más [información](#)

Nota 2: En el Notebook de teoría hemos visto cómo mostrar gráficos lineares simples con el matplotlib. Para este ejercicio explorad las funcionalidades que presenta la librería pandas para generar gráficos a partir de dataframes. También pueden ser de utilidad las funciones [text](#) o [annotate](#) para mostrar el número de nacionalidades.

```
[ ]: # Respuesta
```
