prog_datasci_4_python_entrega

March 27, 2022

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, puedéis 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

A continuación encontraréis los ejercicios que se deben completar en esta PEC y que forman parte de la evaluación de esta unidad.

NumPy - Ejercicios

1.2.1 Ejercicio 1

a) Crea y muestra por pantalla las matrices siguientes:

- Una matriz A 3x3 que tenga valores del 1 al 9 ordenados por fila.
- Una matriz B 3x3 llena de 1.
- Una matriz C 3x2 con valores aleatorios (*random*).

(0.5 puntos) NM

[ ]:
```
# Respuesta
```

b) A continuación, hemos intentado ejecutar el código siguiente con las matrices del apartado anterior, pero nos da error:

- Suma de A y C
- Multiplicación matricial entre C y A

¿Por qué nos encontramos con esos errores? Razona la respuesta.**(0.5 puntos)** NM

[3]:
```
# Suma de A y C

matSuma = A + C
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-3-d312efa0379c> in <module>
      1 # Suma de A y C
      2
----> 3 matSuma = A + C

ValueError: operands could not be broadcast together with shapes (3,3) (3,2)
```

[4]:
```
# Multiplicación matricial entre C y A

matMul = C.dot(A)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-4-0dff880f9d20> in <module>
      1 # Multiplicación matricial entre C y A
      2
```

```
----> 3 matMul = C.dot(A)

ValueError: shapes (3,2) and (3,3) not aligned: 2 (dim 1) != 3 (dim 0)
```

[5]:
```
# Respuesta
```

1.2.2 Ejercicio 2

a) Crea un array de 10 elementos, que vaya de 0 a 20 (ambos incluidos), con un paso de 2 (es decir, 0, 2, 4, etc.). ¿Qué tamaño tiene ese array? Muestra el contenido del array y su tamaño por pantalla. **(0.5 puntos)** NM

[6]:
```
# Respuesta
```

b) ¿Es posible reorganizar este array en una matriz 5x2? En caso de dar error, ¿qué podrías hacer para solucionarlo? Razona la respuesta. **(0.5 puntos)** NM

[8]:
```
# Respuesta
```

1.2.3 Ejercicio 3

Crea una matriz 10 x 10 con todos los elementos como 0. A continuación, pon los elementos de las diagonales con valor 1 (es decir, formando una cruz). Muestra gráficamente la matriz resultante mediante la función imshow() de la llibrería Matplotlib. **(1 punto)** EG

Nota: Para resolver este ejercicio puede resultar útil explorar la función fliplr de numpy.

[10]:
```
# Respuesta
```

1.2.4 Ejercicio 4

a) Crea una matriz 4x5 y llena con muestras aleatorias a partir de una distribución normal estándar (es decir, con media 0 y desviación estándar 1). **(0.25 puntos)** EG

Nota: Para resolver este ejercicio, puede ser útil la función random.normal de Numpy.

[12]:
```
# Respuesta
```

b) Modifica los valores de la matriz M, siguiendo las siguientes indicaciones:

- b.1) Multiplica todos los elementos de la matriz por 10 y conviértelos a integers.
- b.2) Considerando la matriz modificada en el apartado anterior (b.1), M[i,j], on `i`, `j` corresponden a los índicss de las filas y columnas, los valores de cada posición `(i, j)` serán $i^5 + 3j$ para todo `i` par, y $7i - j^2$ para todo `i` impar.

Muestra por pantalla la matriz original y las matrices modificadas (apartados b.1 y b.2). **(0.75 puntos)** NM

[14]:
```
# Respuesta
```

Matplotlib - Ejercicios

1.2.5 Ejercicio 5

En el código que tenemos a continuación, queríamos graficar una Lemniscata de Bernoulli, pero se nos han borrado algunos pedazos. Completa las partes de código que faltan para poder mostrar la gráfica teniendo en cuenta los siguientes elementos: - El valor de alfa es 1. - t debe ser un vector de 1000 elementos, con valores de 0 a 2pi. - La gráfica debe ser de color verde.

(0.5 puntos) NM

[ ]:
```
## Código a completar

import numpy as np
import matplotlib.pyplot as plt

# Para completar
alfa =
t = np.linspace()

# Definimos las ecuaciones
x = alpha * np.sqrt(2) * np.cos(t) / (np.sin(t)**2 + 1)
y = alpha * np.sqrt(2) * np.cos(t) * np.sin(t) / (np.sin(t)**2 + 1)

# Para completar
plt.plot()

plt.show()
```

[ ]:
```
# Respuesta
```

1.2.6 Ejercicio 6

Carga los datos del archivo `fruits.csv`, que puedes encontrar en la carpeta data, en forma de dataframe. Este conjunto de datos recopila información sobre las ventas mensuales de una empresa de frutas.

a) Muestra un gráfico de líneas de puntos de las ventas totales (*Total*) por cada mes (los meses están numerados de 1 a 12). **(0.5 puntos)** EG

Consideraciones:

- La línea debe ser verde con puntos (*dotted line*) de color rojo.

- Se debe mostrar la leyenda en la parte inferior derecha.

Nota: Para resolver este ejercicio puede ser útil explorar detalladamente los argumentos de la función plot de Matplotlib.

[17]:
```
# Respuesta
```

b) Muestra un gráfico circular (*pie chart*) con el número de frutas vendidas este año por producto en porcentaje. **(1 punto)** EG

Consideraciones:

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

Nota: Para realizar este apartado, puedes consultar la documentación de la función pie de matplotlib.

[ ]:
```
# Respuesta
```

Pandas - Ejercicios

1.2.7 Ejercicio 7

Carga los datos del archivo `michelin`, que puedes encontrar en la carpeta data, en un dataframe. Este conjunto de datos recoge a las estrellas Michelin de restaurantes de todo el mundo. Los datos con los que trabajaremos se han adaptado del dataset original que puedes encontrar en la web Kaggle.

Muestra por pantalla las variables del dataset (nombre de las columnas) y el número de filas y columnas.

(0.5 puntos) NM

[ ]:
```
# Respuesta
```

1.2.8 Ejercicio 8

a) Somos unos aficionados de la cocina y estamos planeando un viaje culinario. Si consideramos restaurantes con cualquier tipo de estrella (1, 2 y 3), ¿cuál es la ciudad con más restaurantes Michelin? **(0.5 puntos)** EG

Nota: Para resolver este ejercicio, y en los siguientes, puede ser de utilidad la función aggregate de pandas. Esta función nos permite aplicar diferentes funciones en una columna en un groupby.

[22]:
```
# Respuesta
```

b) Una vez encontrada esta ciudad, ¿cuál es el restaurante de 3 estrellas Michelin más caro de esta ciudad (*AvgPrice* mayor)? ¿Y el más barato (*AvgPrice* menor)?**(0.5 puntos)** NM

[24]:
```
# Respuesta
```

1.2.9 Ejercicio 9

a) Selecciona los restaurantes de Londres y responde a las siguientes preguntas:

- ¿Cuántos restaurantes tienen 1, 2 y 3 estrellas?**(0.5 puntos)** NM
- Si nos fijamos sólo en los restaurantes de 2 estrellas, ¿qué tipo de cocina (columna *Cuisine*) predomina? **(0.5 puntos)** EG

(1 punto)

[26]:
```
# Respuesta
```

b) Utiliza la función plot() de Pandas para mostrar un gráfico de barras con la información de los tipos de cocina más frecuentes en los restaurantes de 2 estrellas Michelin en Londres.

Opcional

[28]:
```
# Respuesta
```

1.2.10 Ejercicio 10

a) Queremos estudiar los precios de los restaurantes **europeos**. Para ello, sigue los siguientes pasos:

- Selecciona los restaurantes que tienen el Euro como moneda. **(0.25 puntos)** NM
- ¿Cuál es el precio medio de los restaurantes según las estrellas Michelin? **(0.25 puntos)** EG

(0.5 puntos)

[30]:
```
# Respuesta
```

b) Para que nos sea más fácil organizar los restaurantes según el precio, queremos separarlos en **3 categorias**:

- medium: si el precio es inferior a 150 euros.
- expensive: si el precio es de entre 150 y 300 (ambos incluidos).
- very expensive: si el precio es superior a 300.

Crea una copia del dataframe original añadiendo una nueva columna que nos indicará por cada fila su categoría según el precio.

Muestra por pantalla los nombres de los restaurantes que tienen la categoría `very expensive` de **París**.

(1 punto) EG

[32]:
```
# Respuesta
```

1.2.11 Ejercicio Opcional

En álgebra lineal, una matriz de Toeplitz, es una matriz en la que cada diagonal descendente de izquierda a derecha es constante.

Construye, mediante la función toeplitz de Scipy, una matriz de Toeplitz que tenga la siguiente forma:

$$\begin{bmatrix} 1 & 4 & 6 & 8 \\ 3 & 1 & 4 & 6 \\ 5 & 3 & 1 & 4 \end{bmatrix}$$

EI

[35]:
```
# Respuesta
```