

# Fundamentos de Programación

## Unidad 8 - Visualitzación de datos en Python

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 PEC 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.

### Ejercicio 1 (Seaborn)

A partir de una web de referencia de datos de sanidad (a [https://github.com/datadista/datasets/tree/master/COVID 19](https://github.com/datadista/datasets/tree/master/COVID%2019)) hemos bajado los datos históricos del proceso de vacunación en España, disponible en el archivo `ccaa_vacunas.csv`, ddonde verás entre otros datos, informes actualizados de forma diaria sobre el estado de:

- Dosis entregadas Pfizer, Moderna, AstraZeneca y totales
- Dosis administradas
- Porcentaje de dosis administradas por 100 habitantes
- Porcentaje sobre entregadas
- Personas con pauta completa
- Porcentaje con pauta completa

Verás que ello se publica de forma diaria por cada comunidad autónoma y el agregado para el estado.

a) Carga los datos en un dataframe de pandas. Muestra el número de filas del dataset, los nombres de las variables, y las 5 primeras filas.

**NM (0.5 puntos)**

In [1]: `# Respuesta`

b) Usando los métodos de Seaborn, calcula y representa en dos gráficos de barras: 1) el total de vacunas administradas por CCAA en este periodo en escala logarítmica, mediante una `bar plot` de forma que muestre la vacunaciones para cada CAA y los 4 tipos de vacunas en diferentes barras y, 2) el porcentaje de dosis administradas por cada 100 habitantes por CCAA. Ten en cuenta que debes eliminar los valores del estado en este caso, y que se deben leer bien los nombres de las diferentes comunidades en la figura.

Según cómo soluciones este ejercicio, te puede ser útil el método `reset_index()` de Pandas.

**EG (1.5 puntos)**

In [2]: `# Respuesta`

c) Representa el Porcentaje de Dosis administradas por cada Comunidad Autónoma a lo largo del tiempo en una misma gráfica mediante Seaborn.

Haz que el eje temporal sea legible. Recuerda en eliminar los valores del estado.

Nota: puedes usar este [link](#) para ayudarte

**EG (1 puntos)**

In [3]: `# Respuesta`

d) Utilizando Seaborn, representa, para los datos publicados a fecha de 2021-10-08:

- Un histograma con el porcentaje sobre entregadas.
- Un histograma con el porcentaje con pauta completa.
- Un diagrama de puntos cruzando las dos variables anteriores (con `jointplot`)

Recuerda en eliminar los valores del estado.

**NM (1 punto)**

In [4]: `# Respuesta`

e) Para la misma fecha que en el ejercicio anterior, utiliza `pairplot` para visualizar un plot de parejas, de las siguientes variables,

- Porcentaje de dosis administradas por 100 habitantes
- Porcentaje con pauta completa
- Porcentaje sobre entregadas

Utiliza un color para cada comunidad autónoma y representa el color en la leyenda. Configura el plot para que en las diagonales se represente un histograma de los datos. Recuerda en eliminar los valores del estado.

**NM**

**(0.5 puntos)**

In [5]: `# Respuesta`

### Ejercicio 2 (NETWORKX)

Considera un grafo no dirigido con NetworkX con las siguientes propiedades.

- El grafo debe tener los siguientes nodos: A, B, C, D, E, F, G, H, I, J, K, L.
- El grafo debe tener las siguientes aristas: (A,B), (A,C), (C,D), (D,F),(D,G), (G,H),(H,I), (H,J), (H,K), (L,E).

a) Crea y visualiza el grafo tal y como se ha descrito.

**EI (1 punto)**

In [6]: `# Respuesta`

b) Imprime el número de nodos, el número de aristas, y el número de componentes conexas del grafo.

**NM (0.5 puntos)**

In [7]: `# Respuesta`

c) Calcula la ruta más corta de `K` a `B`.

**EI (0.5 puntos)**

In [8]: `# Respuesta`

d) Elimina el nodo `D`. Calcula el tamaño de todos los componentes conexos del grafo. Encuentra programáticamente el componente conexo más grande del grafo (esto es, el subgrafo más grande). Representa sólo el subgrafo más grande de forma que los nodos tengan un color proporcional al la centralidad de cada nodo.

Nota: Usar eigenvector centrality como en [aquí](#)

**EG (1.5 puntos)**

In [9]: `# Respuesta`

### Ejercicio 3 (GEOPLOTLIB)

En este ejercicio intentaremos mostrar las antenas de telefonía en España mediante la librería `geoplotlib`.

Nos faltará una tabla con las coordenadas de longitud y latitud asociadas torre de telefonía. Encontrarás un fichero con todas las torres de telefonía en el directorio de datos (`214.csv`)

a) Visualiza un mapa con un punto en el lugar de cada torre de telefonía indicado en el fichero. Ajusta los parámetros del plot para que se visualizen correctamente los datos.

**NM (0.5 puntos)**

In [10]: `# Respuesta`

a) Visualiza la densidad de torres en el mapa mediante la función `kde()`, utiliza un *kernel bandwidth* de `[1,1]`.

**EG (0.5 puntos)**

In [11]: `# Respuesta`

### Ejercicio 4 (matplotlib)

Consiera la siguiente curva:

$$\begin{aligned}r &= 0.5 + \cos(t) + \sin(t) \\ x &= r \cdot \cos^2(t) \\ y &= r \cdot \sin^3(t)\end{aligned}$$

Representa la curva anterior mediante `matplotlib`, con 400 valores de `t` equiespaciados dentro del dominio de  $t = 0$  hasta  $t = 2\pi$ .

Os recordamos que podéis usar la función `linspace()` de Numpy.

**NM (1 punto)**

In [12]: `# Respuesta`

### Ejercicio Opcional

Dada la serie

$$Z_n = z_{n-1}^2 + c$$

- Escribe una función  $f(c, z_0, A, N_{max})$ , que devuelva el número iteraciones  $N$  de la serie anterior necesarias para que el módulo  $|z| > A$ , siendo  $A \in \mathbb{R}$ .
- Necesaria y representa una imagen 2D representando  $N$  sobre el dominio de  $z_0$  comprendido por  $Re(z_0) \in [-2, 2]$ ,  $Im(z_0) \in [-1.2, 1.2]$  con los parámetros ( $A = 2$ ,  $n_{max} = 100$ ,  $c = 0.7885\cos(2\pi/3) + 0.7885\sin(2\pi/3)j$ , Resolución de la imagen = (800.1200) pixels).

Nota: Te puede ser útil la función `meshgrid` de `numpy`.

**EI**

In [13]: `# Respuesta`