

May 29, 2021

## 1 Fonaments de Programació

### 1.1 PAC 8 - Visualització de dades en Python

En aquest Notebook hi trobareu el conjunt d'activitats avaluable com a PAC de l'assignatura. Veureu que cadascuna d'elles té associada una puntuació, que indica el pes que té l'activitat sobre la nota de la PAC. Addicionalment, trobareu un exercici opcional, que no té puntuació dins de la PAC, però que es valora al final del semestre de cara a concedir les matrícules d'honor i arrodonir les notes finals. Podeu treure la màxima nota de la PAC sense necessitat de fer aquest exercici! L'objectiu d'aquest exercici és que serveixi com a petit repte per als estudiants que vulguin aprofundir en el contingut de l'assignatura.

Veureu que totes les activitats de la PAC tenen una etiqueta, que indica els recursos necessaris per tal de dur-la a terme. Hi ha tres possibles etiquetes:

- **NM Només materials:** les eines necessàries per a realitzar l'activitat es poden trobar als materials de l'assignatura.
- **EG Consulta externa guiada:** l'activitat pot requerir fer ús d'eines que no es troben als materials de l'assignatura, però l'enunciat conté indicacions d'on o com trobar la informació addicional necessària per resoldre l'activitat.
- **EI Consulta externa independent:** l'activitat pot requerir fer ús d'eines que no es troben als materials de l'assignatura, i l'enunciat pot no incloure la descripció d'on o com trobar aquesta informació addicional. Caldrà que l'estudiant busqui aquesta informació fent servir els recursos que s'han explicat a l'assignatura.

És important notar que aquestes etiquetes no indiquen el nivell de dificultat de l'exercici, sinó únicament la necessitat de consulta de documentació externa per a la seva resolució. A més, recordeu que les **etiquetes són informatives**, però podeu consultar referències externes sempre que vulgueu (encara que no s'indiqui explícitament) o pot ser que pugueu fer una activitat sense consultar cap mena de documentació. Per exemple, per resoldre una activitat que només requereixi els materials de l'assignatura, podeu consultar referències externes si voleu, ja sigui tant per ajudar-vos en la seva resolució com per ampliar el coneixement!

Pel que fa a la consulta de documentació externa en la resolució dels exercicis, recordeu **citar sempre la bibliografia utilitzada** per a resoldre cada activitat.

### 1.1.1 Exercici 1 (SEABORN)

A partir d'una web de referència de dades de sanitat (a <https://github.com/datadista/datasets/tree/master/COVID%2019>) hem baixat dades històriques de casos d'infecció del virus sars-cov-2 i la seva severitat, que teniu al fitxer el fitxer `provincias_covid19_datos_sanidad_nueva_serie.csv`, on veureu els casos, hospitalitzacions, ingressos a UCI i defuncions, per província i data.

- a) Carrega les dades en un dataframe de pandas. Agrega (suma) casos, hospitalitzacions, ingressos a UCI per cada província, i tot utilitzant seaborn, representa:
- Un histograma de les defuncions (amb `displot`)
  - Un histograma de les hospitalitzacions. (amb `displot`)
  - Un diagrama de punts on amb defuncions i hospitalitzacions. (amb `jointplot`)

Què representa un punt en l'última gràfica?

EG (0.5 punts)

```
[1]: # Resposta
```

- b) Defineix una nova variable anomenada `sever` que contingui una string d'entre `low,midlow,midhigh,high`, en funció del *quartil* de cada província respecte el nombre de funcions.

Torna a generar el darrer gràfic de l'apartat anterior, mostrant ara una relació entre els casos de UCI i les hospitalitzacions (tot usant `scatterplot`). Utilitza diferents colors per mostrar el grau de severitat definit per la variable `sever`.

Sobre les dades mostrades, mostra una recta de regressió de totes les dades utilitzant `regplot`. Què podem dir després de veure la gràfica?

EG (1 punt)

```
[2]: # Resposta
```

- c) Existeix una clara correlació entre els quatre índexs que conté la base de dades. Utilitza un mapa de calor per visualitzar les correlacions entre els atributs numèrics d'aquest darrer dataframe.

NM (0.5 punt)

```
[3]: # Resposta
```

- d) Representa un plot de parelles mitjançant la funció `pairplot`, marca el color de cada província en funció de la severitat (variable `sever`).

NM (0.5 punt)

```
[4]: # Resposta
```

### 1.1.2 Exercici 2 (NETWORKX)

Amb el mateix fitxer de l'exercici 1 podem intentar calcular la correlació de casos entre províncies, per això:

- a) Calcula la correlació de casos entre províncies al llarg del temps. Calcula aquesta correlació entre totes les províncies i visualitza la matriu de correlació de forma clusteritzada amb la funció `clustermap` de seaborn. Visualment, quines províncies apareixen menys correlades amb la resta?

EG (0.5 punts)

[5]: `# Resposta`

Apareixen les illes amb menys correlació, especialment Las Palmas i Santa Cruz de Tenerife

- b) Construeix un graf amb els elements de la matriu de correlació anterior de forma que dues províncies estiguin connectades quan presentin una correlació major de 0.8. Genereu una nova visualització del graf. Mostra les etiquetes de les províncies en cada node.

EI (1 punt)

[6]: `# Resposta`

- c) Trobeu el component connex més gran del graf (això és, el subgraf més gran). Representeu només el subgraf més gran de forma que els nodes tinguin una mida proporcional al màxim de casos de cada província i el color indiqui de forma proporcional la mortalitat (tot usant un mapa de color seqüencial, de forma que el color estigui donant informació sobre el valor de la variable en qüestió, veure [aquí](#)).

EI (1 punt)

[7]: `# Resposta`

- d) Calcula utilitzant les eines de `networkx` la ruta de propagació del virus més probable a partir del graf anterior de correlació de casos entre Zaragoza i Balears, i mostra-la.

**NOTA** Pots consultar a [https://networkx.org/documentation/networkx-1.10/reference/generated/networkx.algorithms.simple\\_paths.shortest\\_simple\\_paths.html](https://networkx.org/documentation/networkx-1.10/reference/generated/networkx.algorithms.simple_paths.shortest_simple_paths.html) com buscar camins entre nodes.

EG (0.5 punts)

[8]: `# Resposta`

### 1.1.3 Exercici 3 (GEOPLOTLIB)

Intentarem mostrar les províncies amb un comportament rellevant en un mapa mitjançant la llibreria `geoplotlib`.

- a) Ens farà falta una taula amb les coordenades de longitud i latitud associades a cada província.

- Utilitza `geocode` de `geopy` per trobar la latitud i longitud associada a cada província, i afegeix aquesta informació en un nou dataframe que tingui variables `provincia`, `lon` i `lat` (per `longitude`, `latitude`).
- Crea un tercer dataframe, amb les següents columnes per cadascuna de les dades d'evolució de les que disposem: `[Fecha, cod_ine, provincia, Casos, Fallecidos, Hospitalizados, UCI, lat, lon]`

NM (1.5 punts)

[9]: `# Resposta`

- b) Marca en un mapa les províncies que van presentar més de 10 casos a data 2020-05-01, utilitzant el dataframe generat anteriorment i la llibreria `geoplotlib`.

NM (1 punt)

[10]: `# Resposta`

#### 1.1.4 Exercici 4 (MATPLOTLIB)

A l'últim exercici representarem l'evolució al llarg del temps de les variables del conjunt de dades tot usant `matplotlib`.

- a) Construeix un plot amb 4 subplots diferents (2x2, és a dir, dos files i dues columnes) tot mostrant l'evolució temporal de cada índex pels casos de Madrid i Barcelona, tot indicant les dates a l'eix x:

```
----- | Casos                | Defuncions |
-----| | Hospitalitzacions | UCI        |
-----
```

Recorda etiquetar els plots, eixos i incloure llegendes.

Nota: Pots usar el mòdul `datetime` per processar la columna de dates.

NM (1 punt)

[11]: `# Resposta`

- b) Veiem que les sèries temporals tenen forces oscil·lacions:
- Calcula la mitjana en finestra lliscant a 14 dies per cadascun dels índexs a fi de reduir l'oscil·lació de cada índex.
  - Representa en dos plots:
    1. Les dades de Defuncions i UCI de Barcelona (en l'eix y) contra les de Madrid (en l'eix x), tot representant una evolució respecte de l'altra.
    2. Les dades de casos i Hospitalitzacions de Barcelona (en l'eix y) contra les de Madrid (en l'eix x).

Recorda etiquetar els plots, eixos i incloure llegendes.

Nota: pots usar la funció `pandas.rolling` per calcular les mitjanes en finestres lliscants.

EG

(1 punts)

[12]: *# Resposta*

### Exercici Opcional

Donada la sèrie

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

i donada un element inicial de la sèrie  $z_0 = 0 + i0$

1. Escribe una funció  $f(c, A, N_{max})$ , que retorni el nombre iteracions  $N$  de la sèrie anterior necessàries perquè el mòdul  $|z| > A$ , essent  $A \in \mathbb{R}$ .
2. Calcula i representa una imatge 2D representant  $N$  sobre el domini de  $c$  compres per  $Re(c) \in [-2, 1]$ ,  $Im(c) \in [-1, 1]$  amb els paràmetres ( $A = 2$ ,  $n_{max} = 100$ ,  $z_0 = i0$ , Resolució de la imatge = (800,1200) pixels).

Nota: Us pot ser útil la funció `meshgrid` de `numpy`.

EI

[13]: *# Resposta*