

Programación para *Data Science*

Unidad 8: Visualización de datos en Python - Ejercicios y preguntas

Ejercicio 0

Cargad el conjunto de datos `pulitzer-circulation-data.csv` en un *dataframe* de pandas. La fuente original de este conjunto de datos es el repositorio de datos de `FiveThirtyEight` (<https://github.com/fivethirtyeight/data>).

```
In [1]: # Respuesta
```

Pregunta 1

¿Qué es la regresión lineal simple?

Respuesta:

Ejercicio 1

Generad un diagrama de dispersión que permita visualizar las muestras del conjunto de datos de Pulitzer según las variables 'Daily Circulation 2004' y 'Daily Circulation 2013'. Incluid una recta con el ajuste lineal entre ambas variables.

Pista: podéis usar la función `jointplot` que hemos visto en el Notebook de explicación. Pensad en qué tipo de gráfica, de entre las que ofrece `jointplot` (<http://seaborn.pydata.org/generated/seaborn.jointplot.html>), se ajusta a los requerimientos del enunciado.

```
In [2]: # Respuesta
```

Pregunta 2

Observando la gráfica generada en el ejercicio anterior, ¿creéis que existe una relación lineal entre la circulación diaria de periódicos en 2004 y en 2013?

Respuesta:

Ejercicio 2

La policía tiene identificados 20 miembros de una banda de atracadores. Ya hace tiempo que están detrás de ellos, por lo que han podido rastrear las comunicaciones entre la banda y tienen muy claro qué canales de comunicación existen entre ellos. Además, la policía también ha podido deducir el *modus operandi* de la banda: cada atraco es realizado únicamente por tres miembros (un conductor, un pistolero y un mochilero) que se encuentran a la hora acordada en un parque de la ciudad. La noche antes del atraco los tres miembros se comunican (ya sea directamente o indirectamente a partir de los canales de comunicación existentes entre los miembros de la banda), para acordar el lugar y la hora del encuentro.

La policía ha recibido un chivatazo de que mañana habrá un nuevo atraco de la banda. Ya son las 7 de la tarde, por lo que solo pueden realizar una única detención antes de que se produzcan las llamadas que acordarán la hora y el lugar de encuentro. ¿A qué miembro de la banda detendríais para evitar el atraco con certeza? Justificad vuestra decisión al jefe de policía.

Para tomar la decisión, tenéis acceso a los datos sobre la banda que ha ido recopilando la policía:

- Un grafo con los canales de comunicación de la banda (dos miembros pueden comunicarse directamente si hay una arista entre ellos en el grafo),
- Tres listas que indican los miembros que pueden actuar como pistoleros, conductores y mochileros, respectivamente

Nota: el jefe de policía puede ser convencido fácilmente mostrándole una visualización del grafo que demuestre por qué vuestra decisión evitaría el atraco.

```
In [3]: # Datos disponibles para el ejercicio.

import networkx as nx

# Grafo G, que representa las comunicaciones directas entre miembros de la banda.
G = nx.Graph()

G.add_edges_from([(1,2), (1,3), (3, 5), (16, 18), (3, 4), (15, 16), (1, 4), (6, 2), (6, 15), (7, 15), \
                  (7, 8), (8, 9), (4, 6), (9, 7), (9, 10), (7, 10), (4, 5), (2, 5), (10, 11), (11, 12), \
                  (11, 7), (12, 10), (13, 8), (14, 8), (3, 6), (16, 17), (17, 18), (18, 19), (17, 19), \
                  (0, 1), (0, 3)])

# Lista de miembros pistoleros
pistoleros = [15, 16, 17, 18, 19]

# Lista de miembros mochileros
mochileros = [0, 1, 2, 3, 4, 5, 6]

# Lista de miembros conductores
conductores = [7, 8, 9, 10, 11, 12, 13, 14]
```

In [4]: # Respuesta

Respuesta:

Ejercicio 3

Cread un mapa con la librería geoplolib que muestre 5 lugares donde os gustaría ir de vacaciones.

Nota 1: podéis guardar las localizaciones en un fichero .csv como el del ejemplo de las paradas de autobuses del notebook de explicación o bien cargar directamente los datos (<https://github.com/andrea-cuttone/geoplolib/wiki/User-Guide#loading-data>) a partir de un diccionario o un *dataframe*.

Nota 2: podéis obtener las coordenadas geográficas de un lugar con el servicio LangLot.net (<http://www.latlong.net/>) o usando Google Maps (<https://www.google.es/maps>) (en este último caso, una vez hayáis buscado el lugar, podéis extraer las coordenadas de la URL).

In [5]: # Respuesta