prog_datasci_8_vis_entrega

December 28, 2020

1 Programación para *Data Science*

1.1 PEC 8 - Visualización de datos en Python

En este Notebook 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.

1.1.1 Ejercicio 1 (SEABORN)

El año 2016 se celebraron los últimos Juegos Olímpicos en Río de Janeiro. Utilizando la página web de Rio2016, hemos descargado el fichero `athletes.csv` donde podemos ver las algunos datos de sus participantes.

a) Carga los datos en un dataframe de pandas y visualiza utilitzando seaborn: * Un histograma de la altura de los participantes. (utilizando `distplot`) * Un histograma del peso de los participantes. (utilizando `distplot`) * Un diagrama de puntos donde veamos la altura en un eje y el peso en el otro. (utilizando `jointplot`)

Explica que información podemos obtener de los tres gráficos realizados.

EG **(0.5 puntos)**

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

En los gráficos podemos ver una distribución aparentemente gaussiana del peso y la altura. Sin embargo, vemos un doble pico que con la información graficada no sabemos explicar. En la tercera gráfica podemos ver una correlación entre peso y altura. A mayor altura, mayor el peso del deportista.

b) Repite el último de los gráficos anteriores, pero ahora visualizando solo los deportistas que hayan logrado una o más medallas de oro (utiliza `scatterplot`). Utiliza colores diferentes para mostrar el número de medallas de oro ganadas (asegúrate que los colores sean diferentes para permitir su identificación).

Sobre los datos mostrados, dibuja la recta de regresión peso/altura de todos los participantes (con y sin medallas medallas) utilitzando regplot. ¿Qué podemos decir después de ver el gráfico?

NM **(0.5 puntos)**

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

Viendo la gráfica anterior, parece que los jugadores que pesan más y no tienen una altura en proporción al peso más cercana a la regresión linear, han ganado menos medallas.

c) Crea una nueva variable en el dataframe que nos enseñe el número de medallas ganadas (independientemente de su tipo), y repite el gráfico anterior, utilitzando como color el número total de medallas. ¿Vemos alguna diferéncia respecto a la gráfica anterior?

NM **(0.5 puntos)**

[3]:
```
# Respuesta
```

d) Seguramente hemos visto cierta correlación entre algunos atributos. Utiliza un mapa de calor pera visualizar las posibles correlaciones entre los atributos numéricos de este último dataframe. ¿Qué información podemos extraer del mapa de calor generado?

NM **(1 punto)**

[4]:
```
# Respuesta
```

(OPCIONAL) En los histogramas realizados inicialmente de peso y altura, vimos un doble pico. Realiza ahora en un mismo gráfico, las gráficas de distribución de densidad de hombres y mujeres por separado. ¿Podemos ahora explicar el motivo del doble pico?

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

1.1.2 Ejercicio 2 (NETWORKX)

El año 1992 coincidiendo com las Olimpadas de Barcelona 1992, se inaguraba la línea de alta velocidad en España que recibia el nombre comercial de AVE, entre Madrid y Sevilla. En esta primera línea, que utilitzaba trenes de la série 100 de Alstom, la velocidad máxima en los trenes en Espanya podía llegar a 300km/h en el tramo creado. Desde su primera inaguración, las lineas, estaciones y velocitats máximes, han ido incrementándose progresivamente.

a) En el fichero `Ave.csv` podemos encontrar una lista con las diferentes estacions de origen y final de trenes de Alta Velocitat Espanyola (AVE). En este ejercicio pedimos crear un grafo de las estaciones, siguiendo los pasos expuestos a continuación:

- Cargar el fichero en un dataframe de pandas.
- Obtener las estaciones, y crear un nodo para cada una de ellas. Visualizar el grafo obtenido.
- Para cada recorrido, crear las aristas correspondientes, y visualizar el grafo.

NM **(0.5 puntos)**

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

b) Generar ahora una nueva visualización del grafo, en el que los nodos tengan una medida proporcional a la población de la ciudad de la estación, y donde las aristas tengan una longitud propocional a la distáncia entre ellas. Para ello, deberemos cargar los ficheros que contienen esta información: `Ave_population.csv` (que contiene la población de cada ciudad) y `Ave_distance.csv`(que contiene las distancias entre estaciones).

EI **(1.5 puntos)**

[7]:
```
# Respuesta
```

c) Calcula utilitzando las herramientas de NetworkX la ruta entre Girona y Sevilla, y muéstrala.

NOTA Puedes consultar en https://networkx.org/documentation/networkx-1.10/reference/generated/networkx.algorithms.simple_paths.shortest_simple_paths.html como generar caminos entre nodos.

EG **(0.5 puntos)**

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

(OPCIONAL) ¿Puedes mostrar el nombre de las estaciones que se situan al final de línea (aquellas que solo tienen un vecino) utilitzando las funciones que nos proporcionan NetworkX?

[9]:
```
# Respuesta
```

(OPCIONAL) Añade una nueva arista entre "Tarragona" y "Castellón" de 186Km, muestra el grafo y calcula las posibles rutas entre "Barcelona" y "Castellón", indicando la distancia de cada una de ellas.

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

1.1.3 Ejercicio 3 (GEOPLOTLIB)

Al largo de la história de los Juegos Olímpicos estos se han realizado en diferentes ciudades del mundo. Vamos a ver su distribución sobre un mapa en este ejercicio.

a) Para realizar una visualización básica de la localización geográfica de las sedes de los Juegos Olímpicos, pedímos:

- Carga los datos con la localización de las principales ciudades del planeta del fichero `worldcities.csv` en un dataframe
- En un segundo dataframe, carga el fichero con los datos de las sedes de los Juegos Olímpicos del fichero `olimpics.csv`.
- Crea un tercer dataframe, con las siguientes columnas para cada uno de los diferentes eventos olímpicos [year,city,country,population,continent,lat,lon]

NM **(0.5 puntos)**

[11]:
```
# Respuesta
```

b) Visualiza un mapa del mundo con las posiciones en las que se han celebrado alguna vez algunos Juegos Olímpicos, utilitzando el dataframe generado anteriormente y la librería geoplotlib. NM **(1 punto)**

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

c) Hay algunas ciudades que han repetido celebraciones. Para estos casos deberemos:

- Cambiar el color de los puntos utilizados para ubicar cada ciudad, y utilizar un color diferente en función de las repeticiones .
- Añadir una etiqueta, con el número de repeticiones, al lado de cada sede olímpica.

EI **(1 punto)**

[13]:
```
# Respuesta
```

1.1.4 Ejercicio 4 (MATPLOTLIB)

Debido al COVID-19, los Juegos Olímpicos del 2020 fueron históricamente aplazados.

Una de las pruebas reinas de los Juegos es el Maratón, que se disputará en la ciudad de Sapporo. Pero quizás, dados los efectos del calentamiento global, las condiciones puede que sean un poco diferentes a las del año 2020. Vamos a ver, utilizando el fichero `Sapporo_weather.csv`, si esto puede ser así.

a) Carga los datos en un dataframe y muestra un histograma de las temperaturas del los últimos 100 años, visualizando en el eje de abcisas 50 subdivisiones entre la temperatura mínima y la máxima, y en el eje de ordenadas, la frecuéncia de cada una de estas temperaturas.

Recuerda etiquetar los ejes. ¿Qué información nos aporta? ¿Es útil?

NM **(0.5 puntos)**

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

b) Muestra ahora un gráfico en el que tengamos el mes del año en el eje de abcisas, y la temperatura en el de ordenadas. Crea una línea por año utilizando un color diferente. ¿Qué es lo que podemos ver ahora?

NM **(1 punto)**

[15]:
```
# Respuesta
```

c) Los Juegos Olímpicos se celebran en verano. ¿Podrías mostrar la evolución de la temperatura en Sapporo en el mes de Julio al largo de los últimos 100 años?

Muestra utilizando una primera línea de regresión la progresión de la temperatura en los últimos 100 años, y con una segunda línea, la de los últimos 50.¿Qué es lo que vemos?

Extrae los datos del dataframe que nos interesan, y pon estos datos en una lista antes de generar el gráfico. Recuerda que es necesario poner el título, y etiquetar correctamente cada eje.

NOTA Podéis consultar en https://stackoverflow.com/questions/6148207/linear-regression-with-matplotlib-numpy como realizar la regresión.

EG **(1 puntos)**

[16]:
```
# Respuesta
```