

Carga de datos desde ficheros

Índice

Tarea 1 Carga de fichero .csv .xls	3
Tarea 2 Carga de json	4
Tarea 3 Carga de xml	5
Tarea 4 Carga desde ficheros shape	6
Tarea 5 Carga desde ficheros geojson	

Descripción

- Los ficheros que se muestran se han obtenido sin modificación alguna de su origen. Estudiarlos antes de comenzar las prácticas.
- A continuación se solicita realizar la carga de datos desde distintos formatos de ficheros.
- Debes cargarlos en un **dataframe de Pandas** y realizar las operaciones que se indican en un Notebook de Jupyter con el código correspondiente.
- Utiliza los **markdowns**, creando una estructura que permita seguir correctamente la ejecución, así como los comentarios pertinentes-
- Carga únicamente las librerías que vayas a utilizar.
- Asegúrate de ejecutar todas las celdas. Una opción para verificar que no te olvidas de ninguna es ir a *Kernel* → *Restart Kernel and Run All Cells...*
- **Entrega** tanto el notebook en **formato .ipynb** como exportado a **formato .pdf**.

CONCEPTOS BÁSICOS DEL CONTENIDO DE LOS FICHEROS FAO¹

- **Índice de precios de los alimentos.** es el promedio aproximado de los índices de los productos básicos que se mencionan: carne, lácteos, cereales, aceites vegetales y azúcar.
- **Índice de precios de la carne:** tomado de los valores de exportación para cuatro tipos de carne, bovina, porcina, ave y ovina.
- **Índice de precios de los productos lácteos:** tomado de los valores de exportación de los valores de mantequilla, queso, leche desnatada en polvo y entera en polvo.
- **Índice de precios de los cereales:** es un promedio de los valores presentes en los mercados de exportación del maíz, arroz, cebada, sorgo y trigo, entre otros.
- **Índice de precios de aceites vegetales:** tomados de los valores de mercado de exportación de varios tipos de aceites de origen vegetal.
- **Índice de precios del azúcar:** se basan en los precios establecidos por el Convenio Internacional del Azúcar

¹ DATOS Y DEFINICIONES OBTENIDAS DE LA FAO (*Food and Agriculture Organization of ONU*)

1. Carga de fichero .csv .xls

El fichero *fao.csv* y el fichero *fao.xls* incluyen un conjunto de registros correspondientes a los precios mensuales de los diferentes productos mencionados desde el mes de enero del año 1990 hasta julio del año 2023

Los decimales en el fichero .csv usan el “.” como separador, mientras en el fichero .xls usan “,”.

Debes cargar los registros correspondientes a los precios en un dataframe, de modo que:

- Se cargan todas las filas que tienen datos. La cabecera del fichero recoge el nombre de cada columna del dataframe.
- El tipo de datos de cada columna debe ser el más correcto, acorde a los datos.
- Debe calcular la media anual del precio de cada producto en un nuevo dataframe añadiendo un ID adicional de la forma ID_YYYY, siendo YYYY el año para cada producto.

En el notebook incluye el código para realizar la carga correcta.

¿observas algún tipo de correlación entre los datos?

2. Carga de json

El fichero *fao.json* tiene información similar al que existía en el fichero csv. Revisa el fichero para entender cómo está construido.

Debes cargarlo de modo que el dataframe resultante tenga un índice formado por el ID_YYYY_MM.

Deberá tener la información de las restantes 12 columnas con los tipos de datos más adecuados.

Incluye el código para cargar correctamente muestra los tipos de cada columna y las filas correspondiente entre el año 1992 a 1995 en un nuevo dataframe.

Tarea 3 Carga de xml

Dispones del fichero *fao.xml*, que tiene exactamente la misma información que el .csv anterior.

Debes parsearlo y cargar la información en un dataframe:

- Los tipos de datos de cada columna del dataframe deben ser los adecuados para los valores almacenados.
- El índice del dataframe será un multiíndice formado por *ID_YYYY_MM*
- Revisa los valores vacíos, deben cargarse como NaN si los hubiera.

El resultado debe ser un dataframe igual al de la Tarea 1.

Indica el código para la carga y muestra tanto las diferentes columnas con sus tipos de datos asociados sólo al año 2023.

Tarea 4 Carga desde ficheros shape

Nota.- La ejecución de los scripts en ficheros shape a veces no es instantánea puede tardar tranquilamente, dependiendo del fichero de datos y la potencia del equipo incluso hasta algún minuto. En mi caso tarda entre 1 y 2 minutos. Dependiendo de la prueba.

Dispones de ficheros PORTAL.shp, PORTAL.shx y PORTAL.dbf que contiene los datos de los portales de las calles de la provincia de Pontevedra

- muestra la información y los primeras filas del fichero shape de las columnas que van desde la 4 a la 10.
- muestra aquellos los portales de la calle pasada por teclado:

```
# Solicitar al usuario que ingrese el valor para el filtro  
  
valor_filtro = int(input("Ingrese el valor para el filtro: "))
```

- utilizando la función *describe()* muestra un resumen estadístico de **todos los datos obtenidos** no solo de la calle que anterior.
- con **gplot()** visualiza todos los datos de la provincia
¿qué crees que serán las manchas blancas que aparecen?

Nota.- si quieres visualizar los datos como dataframe de PANDAS usa

```
# Convertir el GeoDataFrame filtrado a un DataFrame de Pandas  
df_resultado = gdf_filtrado.drop(columns='geometry').copy()
```

Nota 2.- Si las ejecuciones son excesivamente lentas, se puede utilizar los ficheros MANZANAS.xxx siendo la búsqueda por Código Postal. Estos ficheros contienen la información de posiciones geoespaciales de las calles de la ciudad de Pontevedra. La búsqueda sería por Código Postal ()

Tarea 5 Carga desde ficheros geojson

Fuente de datos: *starbucks.geojson*

- Utiliza *pandas* con ***json*** para cargar el archivo GeoJSON en un DataFrame
- Realiza una exploración inicial del conjunto de datos:
 - Imprime las primeras filas de la ciudad de París, pasando el parámetro por teclado para comprender la estructura de los datos.
 - Verifica el **tipo de las columnas** presentes en el conjunto de datos
- Calcula el número de stores de Starbucks de los 10 primeros países con más tiendas.

[Acrónimos de países](#)

- Utiliza un histograma para comparar los valores de esos países.

```
tiendas_por_pais = gdf.groupby('Country')['Store Name'].count().reset_index()
```