

1.- El fichero `cotizacion.csv` contiene las cotizaciones de las empresas del IBEX35 con las siguientes columnas: `nombre` (nombre de la empresa), `Final` (precio de la acción al cierre de bolsa), `Máximo` (precio máximo de la acción durante la jornada), `Mínimo` (precio mínimo de la acción durante la jornada), `volumen` (Volumen al cierre de bolsa), `Efectivo` (capitalización al cierre en miles de euros). Construir una función que construya un DataFrame a partir del un fichero con el formato anterior y devuelva otro DataFrame con el mínimo, el máximo y la media de cada columna.

2.- Los ficheros `emisiones-2016.csv`, `emisiones-2017.csv`, `emisiones-2018.csv` y `emisiones-2019.csv`, contienen datos sobre las emisiones contaminantes en la ciudad de Madrid en los años 2016, 2017, 2018 y 2019 respectivamente. Escribir un programa con los siguientes requisitos:

1. Generar un DataFrame con los datos de los cuatro ficheros.
2. Filtrar las columnas del DataFrame para quedarse con las columnas ESTACION, MAGNITUD, AÑO, MES y las correspondientes a los días D01, D02, etc.
3. Reestructurar el DataFrame para que los valores de los contaminantes de las columnas de los días aparezcan en una única columna.
4. Añadir una columna con la fecha a partir de la concatenación del año, el mes y el día (usar el módulo `datetime`).
5. Eliminar las filas con fechas no válidas (utilizar la función `isnat` del módulo `numpy`) y ordenar el DataFrame por estaciones contaminantes y fecha.
6. Mostrar por pantalla las estaciones y los contaminantes disponibles en el DataFrame.

7. Crear una función que reciba una estación, un contaminante y un rango de fechas y devuelva una serie con las emisiones del contaminante dado en la estación y rango de fechas dado.
8. Mostrar un resumen descriptivo (mínimo, máximo, media, etc.) para cada contaminante.
9. Mostrar un resumen descriptivo para cada contaminante por distritos.
10. Crear una función que reciba una estación y un contaminante y devuelva un resumen descriptivo de las emisiones del contaminante indicado en la estación indicada.
11. Crear una función que devuelva las emisiones medias mensuales de un contaminante y un año dados para todas las estaciones.
12. Crear una función que reciba una estación de medición y devuelva un DataFrame con las medias mensuales de los distintos tipos de contaminantes.

3.- El fichero `bancos.csv` contiene las cotizaciones de los principales bancos de España con : `Empresa` (nombre de la empresa), `Apertura` (precio de la acción a la apertura de bolsa), `Máximo` (precio máximo de la acción durante la jornada), `Mínimo` (precio mínimo de la acción durante la jornada), `Cierre` (precio de la acción al cierre de bolsa), `Volumen` (volumen al cierre de bolsa). Construir una función que reciba el fichero `bancos.csv` y cree un diagrama de líneas con las series temporales de las cotizaciones de cierre de cada banco.

4.- El fichero [titanic.csv](#) contiene información sobre los pasajeros del Titanic. Crear un dataframe con Pandas y a partir de él generar los siguientes diagramas.

1. Diagrama de sectores con los fallecidos y supervivientes.
2. Histograma con las edades.
3. Diagrama de barras con el número de personas en cada clase.
4. Diagrama de barras con el número de personas fallecidas y supervivientes en cada clase.
5. Diagrama de barras con el número de personas fallecidas y supervivientes acumuladas en cada clase.

5.- El fichero [titanic.csv](#) contiene información sobre los pasajeros del Titanic. Escribir un programa con los siguientes requisitos:

1. Generar un DataFrame con los datos del fichero.
2. Mostrar por pantalla las dimensiones del DataFrame, el número de datos que contiene, los nombres de sus columnas y filas, los tipos de datos de las columnas, las 10 primeras filas y las 10 últimas filas
3. Mostrar por pantalla los datos del pasajero con identificador 148.
4. Mostrar por pantalla las filas pares del DataFrame.
5. Mostrar por pantalla los nombres de las personas que iban en primera clase ordenadas alfabéticamente.
6. Mostrar por pantalla el porcentaje de personas que sobrevivieron y murieron.
7. Mostrar por pantalla el porcentaje de personas que sobrevivieron en cada clase.
8. Eliminar del DataFrame los pasajeros con edad desconocida.
9. Mostrar por pantalla la edad media de las mujeres que viajaban en cada clase.

10.      Añadir una nueva columna booleana para ver si el pasajero era menor de edad o no.
11.      Mostrar por pantalla el porcentaje de menores y mayores de edad que sobrevivieron en cada clase.