
GESTIÓN DE DATOS

TAREA N°2 (PAREJAS)

Por medio de la utilización de las librerías vistas hasta la fecha, en especial Pandas, resuelva de manera individual los siguientes ejercicios. Subir resolución completa antes del 26 de septiembre de 2024.

1. Genere un script que pregunte al usuario por montos de ventas entre un rango de años determinado para luego mostrar por pantalla una **Serie** con los datos de las ventas indexadas por año, tanto antes como después de aplicarles un descuento de 10% (ventas con y sin descuento).
2. Genere un script que reciba un **Diccionario de datos** (generado por ud.) con las notas de un certamen de 5 alumnos diferentes, luego devuelva una **Serie** con la nota mínima, la máxima y la desviación típica.
3. Genere una función que reciba por parámetro un **Diccionario de datos** con las notas de 5 alumnos en un certamen y que devuelva una **Serie** con las notas de los alumnos aprobados ordenadas de mayor a menor.
4. Genere un script que muestre por pantalla un **DataFrame** con los datos de la siguiente tabla:

Mes	Ventas	Gastos
Enero	30500	22000
Febrero	35600	23400
Marzo	28300	18100
Abril	33900	20700

5. Genere un script que reciba el **DataFrame** del ejercicio anterior y pregunte al usuario un intervalo de meses (desde mes x, hasta mes y) en los cuales se hará un balance total (ventas – gastos).
6. El fichero *cotización.csv* contiene las cotizaciones de empresas 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 la bolsa), Efectivo (capitalización al cierre en miles de euros). Se pide generar una función que construya un **DataFrame** a partir de un fichero con el formato anterior y que devuelva otro **DataFrame** con el mínimo, máximo y la media de cada una de las columnas.
7. El fichero *titanic.csv* contiene información sobre los pasajeros del Titanic. Se pide generar un script con los siguientes requisitos:
 - a) Generar un **DataFrame** con los datos del fichero.
 - b) 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
 - c) Mostrar por pantalla los datos del pasajero con identificador 148.
 - d) Mostrar por pantalla las filas pares del **DataFrame**.

- e) Mostrar por pantalla los nombres de las personas que iban en primera clase ordenadas alfabéticamente.
 - f) Mostrar por pantalla el porcentaje de personas que sobrevivieron y murieron.
 - g) Mostrar por pantalla el porcentaje de personas que sobrevivieron en cada clase.
 - h) Eliminar del DataFrame los pasajeros con edad desconocida.
 - i) Mostrar por pantalla la edad media de las mujeres que viajaban en cada clase.
 - j) Añadir una nueva columna booleana para ver si el pasajero era menor de edad o no.
 - k) Mostrar por pantalla el porcentaje de menores y mayores de edad que sobrevivieron en cada clase.
8. 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:
- a) Generar un DataFrame con los datos de los cuatro ficheros.
 - b) Filtrar las columnas del DataFrame para quedarse con las columnas ESTACION, MAGNITUD, AÑO, MES y las correspondientes a los días D01, D02, etc.
 - c) Reestructurar el DataFrame para que los valores de los contaminantes de las columnas de los días aparezcan en una única columna.
 - d) Añadir una columna con la fecha a partir de la concatenación del año, el mes y el día (usar el módulo `datetime`).
 - e) Eliminar las filas con fechas no válidas (utilizar la función `isnat` del módulo `numpy`) y ordenar el DataFrame por estaciones, contaminantes y fecha.
 - f) Mostrar por pantalla las estaciones y los contaminantes disponibles en el DataFrame.
 - g) 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.
 - h) Mostrar un resumen descriptivo (mínimo, máximo, media, etc.) para cada contaminante.
 - i) Mostrar un resumen descriptivo para cada contaminante por distritos.
 - j) 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.
 - k) Crear una función que devuelva las emisiones medias mensuales de un contaminante y un año dados para todas las estaciones.
 - l) Crear una función que reciba una estación de medición y devuelva un DataFrame con las medias mensuales de los distintos tipos de contaminantes.