

Ejercicios de la Librería Pandas

Ejercicio 1

Escribir un programa que pregunte al usuario por las ventas de un rango de años y muestre por pantalla una serie con los datos de las ventas indexada por los años, antes y después de aplicarles un descuento del 10%.

Ejercicio 2

Escribir una función que reciba un diccionario con las notas de los alumnos de un curso y devuelva una serie con la nota mínima, la máxima, media y la desviación típica.

Ejercicio 3

Escribir una función que reciba un diccionario con las notas de los alumnos de un curso y devuelva una serie con las notas de los alumnos aprobados ordenadas de mayor a menor.

Ejercicio 4

Escribir programa que genere y muestre por pantalla un DataFrame con los datos de la tabla siguiente:

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

Ejercicio 5

Escribir una función que reciba un DataFrame con el formato del ejercicio anterior, una lista de meses, y devuelva el balance (ventas - gastos) total en los meses indicados.

Ejercicio 6

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 de un fichero con el formato anterior y devuelva otro DataFrame con el mínimo, el máximo y la media de cada columna.

Ejercicio 7

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.

Ejercicio 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:

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 todos 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.

