

11.4-Llibreria-Pandas

1) El fitxer `titanic.csv` conté informació sobre els passatgers del Titanic. Escriure un programa amb els requisits següents:

1. Generar un `DataFrame` amb les dades del fitxer.
2. Mostrar per pantalla les dimensions del `DataFrame`, el nombre de dades que conté, els noms de les columnes i files, els tipus de dades de les columnes, les 10 primeres files i les 10 últimes files
3. Mostrar per pantalla les dades del passatger amb identificador 148.
4. Mostrar per pantalla les files parells del `DataFrame`.
5. Mostrar per pantalla els noms de les persones que anaven a primera classe ordenades alfabèticament.
6. Mostrar per pantalla el percentatge de persones que van sobreviure i morir.
7. Mostrar per pantalla el percentatge de persones que van sobreviure a cada classe.
8. Eliminar del `DataFrame` els passatgers amb edat desconeguda.
9. Mostrar per pantalla l'edat mitjana de les dones que viatjaven a cada classe.
10. Afegir una nova columna booleana per veure si el passatger era menor o no.
11. Mostrar per pantalla el percentatge de menors i majors que van sobreviure a cada classe.

```
import pandas as pd
```

```
# Generar un DataFrame con los datos del fichero.
```

```
titanic = pd.read_csv('https://raw.githubusercontent.com/asalber/asalber.github.io/master/python/ejercicios/soluciones/pandas/titanic.csv', index_col=0)
```

2) Els fitxers *emissions-2016.csv* , *emissions-2017.csv* , *emissions-2018.csv* i *emissions-2019.csv* , contenen dades sobre les emissions contaminats a la ciutat de Madrid els anys 2016, 2017, 2018 i 2019 respectivament. Escriure un programa amb els requisits següents:

1. Generar un DataFrame amb les dades dels quatre fitxers.
2. Filtrar les columnes del DataFrame per quedar-se amb les columnes ESTACIÓ, MAGNITUD, ANY, MES i les corresponents als dies D01, D02, etc.
3. Reestructurar el DataFrame perquè els valors dels contaminants de les columnes dels dies apareguin en una única columna.
4. Afegir una columna amb la data a partir de la concatenació de l'any, el mes i el dia (usar el mòdul `datetime`).
5. Eliminar les files amb dates no vàlides (utilitzar la funció `isnatdel` mòdul `numpy`) i ordenar el DataFrame per estacions contaminants i data.
6. Mostrar per pantalla les estacions i els contaminants disponibles al DataFrame.
7. Crear una funció que rebí una estació, un contaminant i un rang de dates i torneu una sèrie amb les emissions del contaminant donat a l'estació i rang de dates donat.
8. Mostrar un resum descriptiu (mínim, màxim, mitjana, etc.) per a cada contaminant.
9. Mostrar un resum descriptiu per a cada contaminant per districtes.
10. Crear una funció que rebí una estació i un contaminant i torni un resum descriptiu de les emissions del contaminant indicat a l'estació indicada.
11. Crear una funció que retorni les emissions mitges mensuals d'un contaminant i un any donats per a totes les estacions.
12. Crear una funció que rebí una estació de mesura i torni un DataFrame amb les mitges mensuals dels diferents tipus de contaminants.

```
import pandas as pd
import numpy as np
import datetime as dt
```

```
# Generar un DataFrame con los datos de los cuatro ficheros
import pandas as pd
```

```
emisiones_2016 = pd.read_csv('emisiones-2016.csv', sep = ';')
emisiones_2017 = pd.read_csv('emisiones-2017.csv', sep = ';')
emisiones_2018 = pd.read_csv('emisiones-2018.csv', sep = ';')
emisiones_2019 = pd.read_csv('emisiones-2019.csv', sep = ';')
emisiones = pd.concat([emisiones_2016, emisiones_2017, emisiones_2018, emisiones_2019])
emisiones
```

Pràctica:

Dissenya un programa diferent, enunciat inclòs, que combini els exemples anteriors, prova'l, mostra'l i enganxa'l aquí.