Lab #2

This lab is the first step to get farmilliar with some common Machine Learning libraries, named ***Pandas*** and ***Matlotlib***.

- **Deadline: 23:59, 18/03/2024**

0. Mount Drive

```
from google.colab import drive
drive.mount('/content/gdrive')
%cd '/content/gdrive/MyDrive/Lab2'
```

```
Mounted at /content/gdrive
/content/gdrive/MyDrive/Lab2
```

1. Import libraries

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

2. Load dataset

- Load dataset (named ***student-dataset.csv***) using ***read_csv*** from pandas
- Then, display 10 last examples

```
dataset = pd.read_csv("student-dataset.csv") #DataFrame
dataset.head(10)
```

	id	name	nationality	gender	ethnic.group	age	english.grade	math.grade	sciences.grade	language.grade	portfolio.rating	cove
0	0	Kiana Lor	China	F	NaN	22	3.5	3.7	3.1	1.0	4	
1	1	Joshua Lonaker	United States of America	M	NaN	22	2.9	3.2	3.6	5.0	5	
2	2	Dakota Blanco	United States of America	F	NaN	22	3.9	3.8	3.2	5.0	3	
3	3	Natasha Yarusso	United States of America	F	NaN	20	3.3	2.8	3.2	5.0	5	
4	4	Brooke Cazares	Brazil	F	NaN	21	3.7	2.6	3.4	1.0	4	
5	5	Rochelle Johnson	United States of America	F	NaN	21	3.4	3.1	3.7	5.0	2	
6	6	Joey Abreu	China	M	NaN	22	3.7	3.9	3.6	2.0	5	
7	7	Preston Suarez	Brazil	M	NaN	22	3.8	3.7	3.6	2.0	5	
8	8	Lee Dong	Philippines	F	NaN	24	3.9	3.6	3.2	2.0	4	
9	9	Maa'iz al-Dia	Turkey	M	NaN	22	2.4	2.8	3.8	3.0	5	

Next steps: View recommended plots

3. Show statistics of the given dataset

```
dataset.describe()
```

	id	ethnic.group	age	english.grade	math.grade	sciences.grade	language.grade	portfolio.rating	coverletter.rat
count	307.000000	0.0	307.000000	307.000000	307.000000	307.000000	307.000000	307.000000	307.000
mean	153.000000	NaN	21.964169	3.369707	3.414332	3.446580	4.396417	3.986971	4.110
std	88.767487	NaN	1.248013	0.538724	0.476839	0.509081	0.996474	0.928749	0.823
min	0.000000	NaN	19.000000	1.500000	2.100000	1.400000	1.000000	1.000000	1.000
25%	76.500000	NaN	21.000000	3.100000	3.100000	3.200000	4.000000	3.500000	4.000
50%	153.000000	NaN	22.000000	3.500000	3.500000	3.600000	5.000000	4.000000	4.000
75%	229.500000	NaN	23.000000	3.800000	3.800000	3.800000	5.000000	5.000000	5.000
max	306.000000	NaN	26.000000	4.000000	4.000000	4.000000	5.000000	5.000000	5.000

4. Sort dataset by ***nationality***

```
dataset.sort_values(by="nationality")
```

	id	name	nationality	gender	ethnic.group	age	english.grade	math.grade	sciences.grade	language.grade	portfolio.rating
146	146	Juhaina al-Bilal	Bangladesh	F	NaN	20	3.9	3.9	3.9	5.0	5
285	285	Viridiana Ballesteros	Brazil	F	NaN	22	2.8	3.5	3.9	4.0	4
271	271	Jasmine Lopez	Brazil	F	NaN	21	3.8	3.8	1.5	4.0	4
4	4	Brooke Cazares	Brazil	F	NaN	21	3.7	2.6	3.4	1.0	4
96	96	Mateo Cisneros	Brazil	M	NaN	23	3.0	3.4	2.9	4.0	4
...	...	...	...	...	...	...	...	...	...	...	...
147	147	Siena Ingram	United States of America	F	NaN	21	3.8	3.7	3.8	5.0	5
149	149	Vincent Webster	United States of America	M	NaN	23	2.0	3.7	3.4	5.0	4
151	151	Patrick Carnes	United States of America	M	NaN	22	2.4	3.5	3.6	5.0	3
305	305	Eliana Michelsen	United States of America	F	NaN	23	3.0	2.8	2.9	5.0	4
153	153	Jenna Whitney	United States of America	F	NaN	23	3.7	2.9	3.9	5.0	3

307 rows × 13 columns

5. Group dataset by ***nationality*** and ***gender***

```
dataset.groupby(['nationality', 'gender']).groups
```

```
{('Bangladesh', 'F'): [146], ('Brazil', 'F'): [4, 135, 180, 226, 271, 285], ('Brazil', 'M'): [7, 96, 203], ('Canada', 'F'): [111, 139, 293], ('Canada', 'M'): [18, 121, 306], ('Canada', 'other'): [104], ('Chile', 'M'): [211], ('China', 'F'): [0, 162, 189, 218, 232], ('China', 'M'): [6, 21, 24, 61, 76, 176, 194, 214], ('Colombia', 'F'): [183], ('Colombia', 'M'): [94, 185, 244, 283], ('Cuba', 'F'): [70], ('Dominican Republic', 'F'): [29], ('Egypt', 'M'): [150], ('El Salvador', 'F'): [105], ('Germany', 'F'): [128], ('India', 'F'): [35, 41, 72, 175, 198], ('India', 'M'): [172, 215, 259], ('Japan', 'F'): [15, 37, 79, 95, 164, 200, 264, 280], ('Japan', 'M'): [20, 43, 46, 97, 99], ('Korea (Republic of)', 'F'): [68, 251], ('Korea (Republic of)', 'M'): [213], ('Mexico', 'F'): [39, 49, 82, 117, 125, 140, 145, 148, 169, 188, 237, 265, 296], ('Mexico', 'M'): [13, 50, 87, 112, 141, 178, 191, 210, 236, 268], ('Mexico', 'other'): [17], ('Morocco', 'F'): [187], ('Myanmar', 'F'): [85], ('Netherlands', 'F'): [190], ('Nicaragua', 'F'): [301], ('Pakistan', 'M'): [23, 59, 101], ('Peru', 'M'): [36], ('Philippines', 'F'): [8], ('Poland', 'M'): [65], ('Russian Federation', 'F'): [42, 93, 120, 134, 217], ('Russian Federation', 'M'): [177], ('Spain', 'M'): [52, 129], ('Thailand', 'F'): [234], ('Tunisia', 'M'): [77], ('Turkey', 'F'): [171], ('Turkey', 'M'): [9], ('Ukraine', 'M'): [165], ('United Kingdom', 'F'): [230], ('United Kingdom', 'M'): [173], ('United States
```

```
of America', 'F'): [2, 3, 5, 10, 11, 16, 22, 25, 26, 28, 30, 31, 40, 44, 45, 47, 51, 53, 54, 55, 57, 60, 63, 64, 67, 69, 73, 74, 78,
84, 88, 89, 91, 92, 100, 102, 109, 110, 114, 118, 119, 126, 131, 132, 138, 142, 143, 147, 153, 154, 155, 156, 157, 158, 161, 167, 168,
181, 184, 186, 193, 205, 206, 209, 212, 216, 225, 227, 228, 229, 231, 238, 239, 241, 246, 249, 250, 255, 256, 260, 262, 266, 267, 274,
277, 279, 290, 292, 300, 303, 305], ('United States of America', 'M'): [1, 12, 14, 19, 27, 32, 33, 34, 38, 48, 56, 58, 62, 66, 71, 75,
80, 81, 83, 86, 90, 98, 103, 106, 107, 108, 113, 115, 116, 122, 123, 124, 127, 130, 133, 136, 137, 144, 149, 151, 152, 159, 160, 163,
166, 170, 174, 179, 182, 192, 195, 196, 197, 199, 201, 202, 204, 207, 208, 219, 220, 221, 222, 224, 233, 235, 242, 243, 245, 247, 248,
252, 253, 254, 257, 258, 261, 263, 269, 270, 272, 273, 275, 276, 278, 281, 282, 284, 286, 287, 288, 289, 291, 294, 295, 297, 298, 299,
302, 304], ('United States of America', 'other'): [223, 240]}
```

6. Use scatter plot to represent the relationship between ***age*** and ***refletter.rating***

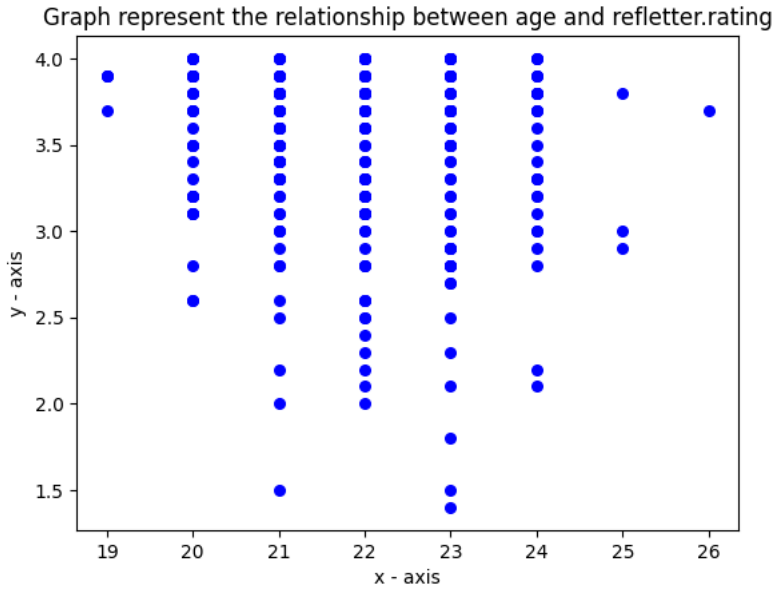
Rememer adding titles, xlabel, ylabel, ... to the plot

```
plt.scatter(dataset["age"], dataset["refletter.rating"], s=60, color="blue")
plt.title('Graph represent the relationship between age and refletter.rating')
plt.xlabel('x - axis')
plt.ylabel('y - axis')
plt.show()
```

7. Use scatter plot to represent the relationship between ***age*** and ***sciences.grade***

Rememer adding titles, xlabel, ylabel, ... to the plot

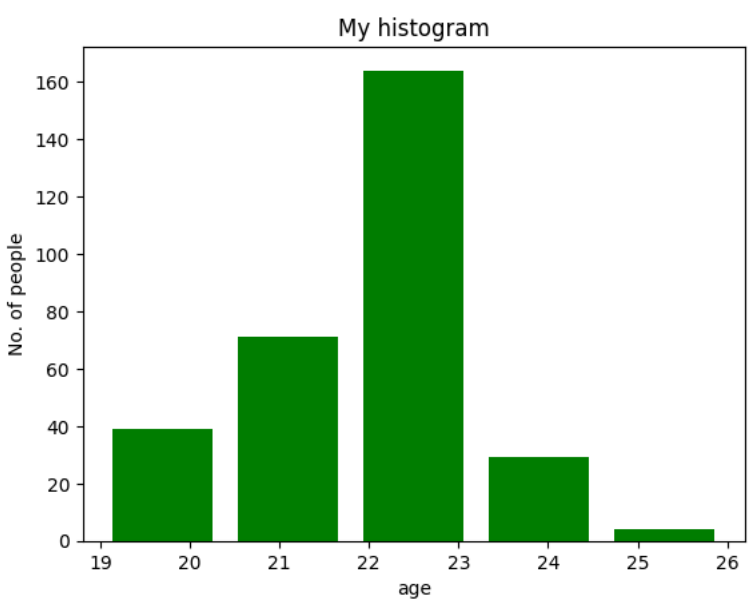
```
plt.scatter(dataset["age"], dataset["sciences.grade"], s=30, color="blue")
plt.title('Graph represent the relationship between age and refletter.rating')
plt.xlabel('x - axis')
plt.ylabel('y - axis')
plt.show()
```



8. Use histogram plot to represent the distribution of ***Age***

using bins=5

```
bins=5
plt.hist(dataset["age"], bins, color='green',histtype='bar',rwidth=0.8)
plt.xlabel('age')
plt.ylabel('No. of people')
plt.title('My histogram')
plt.show()
```



9. Create a data frame that computes the average of ages in each countries?

Hint: Use groupby, select the age column and aggregate using mean.

```
dataset.groupby(['nationality'])['age'].mean()
```

```
nationality
Bangladesh                  20.000000
Brazil                      22.444444
Canada                      21.285714
```

```
Chile                      21.000000
China                      21.692308
Colombia                   21.200000
Cuba                       21.000000
Dominican Republic         22.000000
Egypt                      20.000000
El Salvador                23.000000
Germany                    20.000000
India                      21.625000
Japan                      22.384615
Korea (Republic of)        22.000000
Mexico                     21.833333
Morocco                    21.000000
Myanmar                    22.000000
Netherlands                21.000000
Nicaragua                  24.000000
Pakistan                   22.333333
Peru                       22.000000
Philippines                24.000000
Poland                     21.000000
Russian Federation         22.500000
Spain                      22.500000
Thailand                   22.000000
Tunisia                    23.000000
Turkey                     21.500000
Ukraine                    22.000000
United Kingdom             20.500000
United States of America   22.020725
Name: age, dtype: float64
```

10. Create a dataframe including average of ***english.grade, math.grade, sciences.grade, language.grade, portfolio.rating, coverletter.rating, refletter.rating*** of each country

```
dataset.groupby(['nationality'])['english.grade', 'math.grade',\
              'sciences.grade', 'language.grade', 'portfolio.rating',\
                              'coverletter.rating', 'refletter.rating'].mean()
```

```
<ipython-input-11-903db137743c>:1: FutureWarning: Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecat
  dataset.groupby(['nationality'])['english.grade', 'math.grade',\
```

nationality	english.grade	math.grade	sciences.grade	language.grade	portfolio.rating	coverletter.rating	refletter.rating
Bangladesh	3.900000	3.900000	3.900000	5.000000	5.000000	4.000000	5.000000
Brazil	3.577778	3.377778	3.055556	2.888889	4.000000	4.222222	4.333333
Canada	3.657143	3.014286	3.600000	5.000000	4.571429	3.857143	4.000000
Chile	3.700000	3.700000	4.000000	3.000000	4.000000	3.000000	5.000000
China	3.253846	3.784615	3.376923	3.153846	4.230769	4.230769	4.307692
Colombia	3.540000	3.580000	3.280000	3.200000	3.800000	4.200000	4.200000
Cuba	3.900000	2.800000	3.200000	3.000000	4.000000	4.000000	5.000000
Dominican Republic	3.400000	3.800000	4.000000	4.000000	5.000000	5.000000	5.000000
Egypt	1.500000	3.500000	3.200000	3.000000	3.000000	5.000000	4.000000
El Salvador	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Finally,

Remember renaming the notebook.

Korea (Republic of)	3.533333	3.066667	3.200000	2.666667	3.666667	3.000000	4.000000