Mastering Pandas Library and EDA (Part-1)

- Create a dedicated pandas environment
- After that, install the important libraries in your created environment (mainly pandas).
- Panda version:
  - to check the pandas version we use the function "__version__" e.g., pd.__version__

This line of code will print our pandas library version.

- Now it's time to read the data:
  - to read the data we use the function **pd.read** for example -> pd.read_csv('path to data')
- Now we want to convert our data into any other type (like CSV, XLS, HTML, JSON etc), for this we use the function **pd.to** for example we have our data in CSV and we want to convert the data into the XLS: data.to_excel('path/to/folder/filename.xlsx')
- to ready data in XLS -> pd.read_excel('path to data')

What if there are different sheets in your excel data? In pandas to read the any particular sheet we add sheet name as well like: pd.read_excel('path to data', sheet_name= 'name_of_sheet')

Some basics function of pandas libraries:

→ to see how our data looks like we use:

- **df.info()** -> this will gives a quick summary of our data (Index Range, Total Columns Count, Column Names, Data Types and Non-Null Count, Memory Usage etc)

Pro tip: if you want to know the number of cells in your dataset it's simple just multiply the number of columns with the number of rows you will get the total number of cells in your data (columns * rows)

→ Now if we want to look the head of our data (top 5 rows) we use:

- **df.head()** -> this will print top 5 rows of our data including all columns

→ Now let's say we want to look up the tail of our data (last 5 rows) we use:

- **df.tail()** -> this will print the last 5 rows of our data

→ what if we want to print the particular number of rows?

- Let's say we want to print the top 10 rows of our data for this we will mention the number of rows as well like e.g., **df.head(10)** -> this will print the top 10 rows of our data and same for bottom rows -> **df.tail(10)**

→ to view the descriptive statistic of numeric data:

- **df.describe()** -> It returns descriptive statistics (count, mean, std, min, 25%, 50%, 75%, max) of the numerical columns in our data by default.