# Muhammad Arham Adeel

## 1st assignemnt

# 01 - How to find the version

- here i can immport pandas library

In [1]:

```
import pandas as pd
pd.__version__
```

Out[1]:

```
'1.3.4'
```

In [2]:

```
# Another way

pd.show_versions()
```

```
INSTALLED VERSIONS
------------------
commit           : 945c9ed766a61c7d2c0a7cbb251b6edebf9cb7d5
python           : 3.9.7.final.0
python-bits      : 64
OS               : Windows
OS-release       : 10
Version          : 10.0.19044
machine          : AMD64
processor        : Intel64 Family 6 Model 61 Stepping 4, GenuineIntel
byteorder        : little
LC_ALL           : None
LANG             : None
LOCALE           : English_United States.1252

pandas           : 1.3.4
numpy            : 1.20.3
pytz             : 2021.3
dateutil         : 2.8.2
pip              : 21.2.4
setuptools       : 58.0.4
Cython           : 0.29.24
pytest           : 6.2.4
hypothesis       : None
sphinx           : 4.2.0
blosc            : None
feather          : None
xlsxwriter       : 3.0.1
lxml.etree       : 4.6.3
html5lib         : 1.1
pymysql          : None
psycopg2         : None
jinja2           : 2.11.3
IPython          : 7.29.0
```

```
pandas_datareader: None
bs4              : 4.10.0
bottleneck       : 1.3.2
fsspec           : 2021.10.1
fastparquet      : None
gcsfs            : None
matplotlib       : 3.4.3
numexpr          : 2.7.3
odfpy            : None
openpyxl         : 3.0.9
pandas_gbq       : None
pyarrow          : None
pyxlsb           : None
s3fs             : None
scipy            : 1.7.1
sqlalchemy       : 1.4.22
tables           : 3.6.1
tabulate         : 0.8.9
xarray           : 2022.3.0
xlrd             : 2.0.1
xlwt             : 1.3.0
numba            : 0.54.1
```

# 02-Make a DataFrame

In [3]:

```
df = pd.DataFrame({"A col" : [1,2,3,7,4,6], "B col" : [4,5,6,9,5,2]})
df
```

Out[3]:

| | A col | B col |
|---|---|---|
| **0** | 1 | 4 |
| **1** | 2 | 5 |
| **2** | 3 | 6 |
| **3** | 7 | 9 |
| **4** | 4 | 5 |
| **5** | 6 | 2 |

## Using Numpy to create Data Frame

In [4]:

```
import numpy as np
arr = np.array([[1,2,3],[4,5,6],[7,8,9]])
df1 = pd.DataFrame(arr)
df1
```

Out[4]:

| | 0 | 1 | 2 |
|---|---|---|---|
| **0** | 1 | 2 | 3 |
| **1** | 4 | 5 | 6 |
| **2** | 7 | 8 | 9 |

#### here i can give random number using numpy array

- Numpy array is used to generate a random number
- we can create columns and instances as much as we need

In [5]:

```
pd.DataFrame(np.random.rand(4,8))
```

Out[5]:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| **0** | 0.737316 | 0.111917 | 0.648699 | 0.454624 | 0.867683 | 0.755936 | 0.439069 | 0.464889 |
| **1** | 0.235865 | 0.165341 | 0.334704 | 0.043644 | 0.195523 | 0.242108 | 0.682831 | 0.595150 |
| **2** | 0.515315 | 0.596346 | 0.649266 | 0.908426 | 0.009808 | 0.022175 | 0.630853 | 0.587509 |
| **3** | 0.587436 | 0.380590 | 0.844358 | 0.953675 | 0.124592 | 0.326889 | 0.096996 | 0.411164 |

In [ ]:

In [6]:

```
pd.DataFrame(np.random.rand(4,8),columns= list("KUCHNAHI"))
```

Out[6]:

| | K | U | C | H | N | A | H | I |
|---|---|---|---|---|---|---|---|---|
| **0** | 0.533359 | 0.226860 | 0.323953 | 0.955066 | 0.623843 | 0.283041 | 0.586427 | 0.873199 |
| **1** | 0.338765 | 0.139070 | 0.316076 | 0.753104 | 0.098088 | 0.330406 | 0.821983 | 0.545780 |
| **2** | 0.620720 | 0.114825 | 0.179843 | 0.180221 | 0.669716 | 0.482552 | 0.015512 | 0.668298 |
| **3** | 0.366212 | 0.833549 | 0.994731 | 0.721742 | 0.045127 | 0.574351 | 0.363345 | 0.757443 |

# 03- How to rename columns

In [7]:

```
df
```

Out[7]:

| | A col | B col |
|---|---|---|
| **0** | 1 | 4 |
| **1** | 2 | 5 |
| **2** | 3 | 6 |
| **3** | 7 | 9 |
| **4** | 4 | 5 |
| **5** | 6 | 2 |

### First method

In [8]:

```
df.rename(columns = {"A col":'col_a' , "B col" : "col b"})
```

Out[8]:

|   | col_a | col b |
|---|---|---|
| **0** | 1 | 4 |
| **1** | 2 | 5 |
| **2** | 3 | 6 |
| **3** | 7 | 9 |
| **4** | 4 | 5 |
| **5** | 6 | 2 |

### Second method to rename the columns

In [9]:

```
## rename the columns another way
df.columns = ["col_aa","col_bb"]
df
```

Out[9]:

|   | col_aa | col_bb |
|---|---|---|
| **0** | 1 | 4 |
| **1** | 2 | 5 |
| **2** | 3 | 6 |
| **3** | 7 | 9 |
| **4** | 4 | 5 |
| **5** | 6 | 2 |

### here i can replace any strig,values from columns

- to replace any character, string

In [10]:

```
df.columns = df.columns.str.replace("_"," ")
df
```

Out[10]:

|   | col aa | col bb |
|---|---|---|
| **0** | 1 | 4 |
| **1** | 2 | 5 |
| **2** | 3 | 6 |
| **3** | 7 | 9 |
| **4** | 4 | 5 |
| **5** | 6 | 2 |

# Add prefix

- Prefix can be used to add any words,character, numbers space, on the strat of the given character/number

In [11]:

```
df = df.add_prefix("hello_")
df
```

Out[11]:

| | hello_col aa | hello_col bb |
|---|---|---|
| **0** | 1 | 4 |
| **1** | 2 | 5 |
| **2** | 3 | 6 |
| **3** | 7 | 9 |
| **4** | 4 | 5 |
| **5** | 6 | 2 |

# Add Suffix

- Prefix can be used to add any words,character, numbers on the last of the given character/number

In [12]:

```
df = df.add_suffix("_Bye")
df
```

Out[12]:

| | hello_col aa_Bye | hello_col bb_Bye |
|---|---|---|
| **0** | 1 | 4 |
| **1** | 2 | 5 |
| **2** | 3 | 6 |
| **3** | 7 | 9 |
| **4** | 4 | 5 |
| **5** | 6 | 2 |

### here i can create again rename the columns

In [13]:

```
df.columns = ["col_a" , "col_b"]
df
```

Out[13]:

| | col_a | col_b |
|---|---|---|
| **0** | 1 | 4 |
| **1** | 2 | 5 |
| **2** | 3 | 6 |
| **3** | 7 | 9 |

| | col_a | col_b |
|---|---|---|
| **4** | 4 | 5 |
| **5** | 6 | 2 |

# 04-Using template data

In [14]:

```
import pandas as pd
import numpy as np
import seaborn as sns

df = sns.load_dataset("tips")
df
```

Out[14]:

| | total_bill | tip | sex | smoker | day | time | size |
|---|---|---|---|---|---|---|---|
| **0** | 16.99 | 1.01 | Female | No | Sun | Dinner | 2 |
| **1** | 10.34 | 1.66 | Male | No | Sun | Dinner | 3 |
| **2** | 21.01 | 3.50 | Male | No | Sun | Dinner | 3 |
| **3** | 23.68 | 3.31 | Male | No | Sun | Dinner | 2 |
| **4** | 24.59 | 3.61 | Female | No | Sun | Dinner | 4 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **239** | 29.03 | 5.92 | Male | No | Sat | Dinner | 3 |
| **240** | 27.18 | 2.00 | Female | Yes | Sat | Dinner | 2 |
| **241** | 22.67 | 2.00 | Male | Yes | Sat | Dinner | 2 |
| **242** | 17.82 | 1.75 | Male | No | Sat | Dinner | 2 |
| **243** | 18.78 | 3.00 | Female | No | Thur | Dinner | 2 |

244 rows × 7 columns

### here i can see the summary

In [15]:

```
df.describe()
```

Out[15]:

| | total_bill | tip | size |
|---|---|---|---|
| **count** | 244.000000 | 244.000000 | 244.000000 |
| **mean** | 19.785943 | 2.998279 | 2.569672 |
| **std** | 8.902412 | 1.383638 | 0.951100 |
| **min** | 3.070000 | 1.000000 | 1.000000 |
| **25%** | 13.347500 | 2.000000 | 2.000000 |
| **50%** | 17.795000 | 2.900000 | 2.000000 |

| | total_bill | tip | size |
|---|---|---|---|
| **75%** | 24.127500 | 3.562500 | 3.000000 |
| **max** | 50.810000 | 10.000000 | 6.000000 |

### here i can see the columns

In [16]:

```
df.columns
```

Out[16]:

```
Index(['total_bill', 'tip', 'sex', 'smoker', 'day', 'time', 'size'], dtype='object')
```

### Saving a data set into csv

In [17]:

```
df.to_csv("tips_save.csv",index=False)
```

### Saving data into excel

In [18]:

```
df.to_excel("tips_save.xlsx")
```

# 05- Using your own data

In [19]:

```
df = pd.read_csv("tips_save.csv")
df
```

Out[19]:

| | total_bill | tip | sex | smoker | day | time | size |
|---|---|---|---|---|---|---|---|
| **0** | 16.99 | 1.01 | Female | No | Sun | Dinner | 2 |
| **1** | 10.34 | 1.66 | Male | No | Sun | Dinner | 3 |
| **2** | 21.01 | 3.50 | Male | No | Sun | Dinner | 3 |
| **3** | 23.68 | 3.31 | Male | No | Sun | Dinner | 2 |
| **4** | 24.59 | 3.61 | Female | No | Sun | Dinner | 4 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **239** | 29.03 | 5.92 | Male | No | Sat | Dinner | 3 |
| **240** | 27.18 | 2.00 | Female | Yes | Sat | Dinner | 2 |
| **241** | 22.67 | 2.00 | Male | Yes | Sat | Dinner | 2 |
| **242** | 17.82 | 1.75 | Male | No | Sat | Dinner | 2 |
| **243** | 18.78 | 3.00 | Female | No | Thur | Dinner | 2 |

244 rows × 7 columns

In [ ]: