



In [1]:

```
1 import numpy as np  
2 import pandas as pd
```

Import data/ Load data

- Option1 : path + (file name with extension)

In [2]:

```
1 df1 = pd.read_excel("D:\\NIT\\02. Python for Data Science\\liver.xlsx")  
2 df1
```

Out[2]:

	Age	Gender	TB	DB	Alkphos	Sgpt	Sgot	TP	ALB	AG	LiverPatient
0	65	Female	0.7	0.1	187	16	18	6.8	3.3	0.90	Yes
1	62	Male	10.9	5.5	699	64	100	7.5	3.2	0.74	Yes
2	62	Male	7.3	4.1	490	60	68	7.0	3.3	0.89	Yes
3	58	Male	1.0	0.4	182	14	20	6.8	3.4	NaN	Yes
4	72	NaN	3.9	2.0	195	27	59	7.3	2.4	0.40	Yes
5	46	Male	1.8	0.7	208	19	14	7.6	4.4	1.30	Yes
6	26	NaN	0.9	0.2	154	16	12	7.0	3.5	NaN	Yes
7	29	Female	NaN	0.3	202	14	11	6.7	3.6	1.10	Yes
8	17	Male	0.9	0.3	202	22	19	7.4	4.1	1.20	No

In [3]:

```
1 df2 = pd.read_csv("D:\\NIT\\02. Python for Data Science\\liver.csv")  
2 df2
```

Out[3]:

	Age	Gender	TB	DB	Alkphos	Sgpt	Sgot	TP	ALB	AG	LiverPatient
0	65	Female	0.7	0.1	187	16	18	6.8	3.3	0.90	Yes
1	62	Male	10.9	5.5	699	64	100	7.5	3.2	0.74	Yes
2	62	Male	7.3	4.1	490	60	68	7.0	3.3	0.89	Yes
3	58	Male	1.0	0.4	182	14	20	6.8	3.4	NaN	Yes
4	72	NaN	3.9	2.0	195	27	59	7.3	2.4	0.40	Yes
5	46	Male	1.8	0.7	208	19	14	7.6	4.4	1.30	Yes
6	26	NaN	0.9	0.2	154	16	12	7.0	3.5	NaN	Yes
7	29	Female	NaN	0.3	202	14	11	6.7	3.6	1.10	Yes
8	17	Male	0.9	0.3	202	22	19	7.4	4.1	1.20	No

- Option2 : "file name with extension"



Note: this will work, when data & python file both in same folder

In [4]:

```
1 df = pd.read_csv("Liver.csv")
2 df
```

Out[4]:

	Age	Gender	TB	DB	Alkphos	Sgpt	Sgot	TP	ALB	AG	LiverPatient
0	65	Female	0.7	0.1	187	16	18	6.8	3.3	0.90	Yes
1	62	Male	10.9	5.5	699	64	100	7.5	3.2	0.74	Yes
2	62	Male	7.3	4.1	490	60	68	7.0	3.3	0.89	Yes
3	58	Male	1.0	0.4	182	14	20	6.8	3.4	NaN	Yes
4	72	NaN	3.9	2.0	195	27	59	7.3	2.4	0.40	Yes
5	46	Male	1.8	0.7	208	19	14	7.6	4.4	1.30	Yes
6	26	NaN	0.9	0.2	154	16	12	7.0	3.5	NaN	Yes
7	29	Female	NaN	0.3	202	14	11	6.7	3.6	1.10	Yes
8	17	Male	0.9	0.3	202	22	19	7.4	4.1	1.20	No

In [5]:

```
1 #first 5 observations
2 df.head()
```

Out[5]:

	Age	Gender	TB	DB	Alkphos	Sgpt	Sgot	TP	ALB	AG	LiverPatient
0	65	Female	0.7	0.1	187	16	18	6.8	3.3	0.90	Yes
1	62	Male	10.9	5.5	699	64	100	7.5	3.2	0.74	Yes
2	62	Male	7.3	4.1	490	60	68	7.0	3.3	0.89	Yes
3	58	Male	1.0	0.4	182	14	20	6.8	3.4	NaN	Yes
4	72	NaN	3.9	2.0	195	27	59	7.3	2.4	0.40	Yes

In [6]:

```
1 #last 5 observations
2 df.tail()
```

Out[6]:

	Age	Gender	TB	DB	Alkphos	Sgpt	Sgot	TP	ALB	AG	LiverPatient
4	72	NaN	3.9	2.0	195	27	59	7.3	2.4	0.4	Yes
5	46	Male	1.8	0.7	208	19	14	7.6	4.4	1.3	Yes
6	26	NaN	0.9	0.2	154	16	12	7.0	3.5	NaN	Yes
7	29	Female	NaN	0.3	202	14	11	6.7	3.6	1.1	Yes
8	17	Male	0.9	0.3	202	22	19	7.4	4.1	1.2	No



In [7]:

```
1 #dimensions of the data frame
2 df.shape
```

Out[7]:

(9, 11)

In [8]:

```
1 #Return an int representing the number of elements.
2 df.size
```

Out[8]:

99

In [9]:

```
1 #column names/Variable names
2 df.columns
```

Out[9]:

```
Index(['Age', 'Gender', 'TB', 'DB', 'Alkphos', 'Sgpt', 'Sgot', 'TP', 'ALB',
       'AG', 'LiverPatient'],
      dtype='object')
```

In [10]:

```
1 #column names/Variables
2 df.keys() #option-2
```

Out[10]:

```
Index(['Age', 'Gender', 'TB', 'DB', 'Alkphos', 'Sgpt', 'Sgot', 'TP', 'ALB',
       'AG', 'LiverPatient'],
      dtype='object')
```

In [11]:

```
1 #Data type of each column
2 df.dtypes
```

Out[11]:

```
Age          int64
Gender        object
TB           float64
DB           float64
Alkphos      int64
Sgpt          int64
Sgot          int64
TP           float64
ALB          float64
AG           float64
LiverPatient  object
dtype: object
```

concise summary of a DataFrame



In [12]:

```
1 df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9 entries, 0 to 8
Data columns (total 11 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   Age          9 non-null      int64  
 1   Gender       7 non-null      object  
 2   TB           8 non-null      float64 
 3   DB           9 non-null      float64 
 4   Alkphos      9 non-null      int64  
 5   Sgpt          9 non-null      int64  
 6   Sgot          9 non-null      int64  
 7   TP           9 non-null      float64 
 8   ALB          9 non-null      float64 
 9   AG           7 non-null      float64 
 10  LiverPatient 9 non-null      object  
dtypes: float64(5), int64(4), object(2)
memory usage: 920.0+ bytes
```

Check for missing values

In [13]:

```
1 df.isnull().sum()
```

Out[13]:

```
Age          0
Gender       2
TB           1
DB           0
Alkphos      0
Sgpt          0
Sgot          0
TP           0
ALB          0
AG           2
LiverPatient 0
dtype: int64
```



In [14]:

```
1 df.isna().sum()
```

Out[14]:

```
Age          0
Gender       2
TB           1
DB           0
Alkphos      0
Sgpt          0
Sgot          0
TP           0
ALB          0
AG           2
LiverPatient 0
dtype: int64
```

Accessing Single column

In [15]:

```
1 #option-1
2
3 df.TB
```

Out[15]:

```
0    0.7
1   10.9
2    7.3
3    1.0
4    3.9
5    1.8
6    0.9
7    NaN
8    0.9
Name: TB, dtype: float64
```

In [16]:

```
1 #option-2
2
3 df['TB']
```

Out[16]:

```
0    0.7
1   10.9
2    7.3
3    1.0
4    3.9
5    1.8
6    0.9
7    NaN
8    0.9
Name: TB, dtype: float64
```



In [17]:

```
1 #option-3  
2 df['TB'].values #output as array
```

Out[17]:

```
array([ 0.7, 10.9, 7.3, 1. , 3.9, 1.8, 0.9, nan, 0.9])
```

Accessing Multiple Columns

In [18]:

```
1 #we need to pass, list of multiple columns by the column names  
2 df[['TB','DB']]
```

Out[18]:

	TB	DB
0	0.7	0.1
1	10.9	5.5
2	7.3	4.1
3	1.0	0.4
4	3.9	2.0
5	1.8	0.7
6	0.9	0.2
7	NaN	0.3
8	0.9	0.3

Drop a row

- Drop (Need to specify that axis = 0 or 'index', axis=1 or 'columns')



In [19]:

```
1 #option-1
2 df1 = df.drop(0, axis=0)
3 df1
```

Out[19]:

	Age	Gender	TB	DB	Alkphos	Sgpt	Sgot	TP	ALB	AG	LiverPatient
1	62	Male	10.9	5.5	699	64	100	7.5	3.2	0.74	Yes
2	62	Male	7.3	4.1	490	60	68	7.0	3.3	0.89	Yes
3	58	Male	1.0	0.4	182	14	20	6.8	3.4	NaN	Yes
4	72	NaN	3.9	2.0	195	27	59	7.3	2.4	0.40	Yes
5	46	Male	1.8	0.7	208	19	14	7.6	4.4	1.30	Yes
6	26	NaN	0.9	0.2	154	16	12	7.0	3.5	NaN	Yes
7	29	Female	NaN	0.3	202	14	11	6.7	3.6	1.10	Yes
8	17	Male	0.9	0.3	202	22	19	7.4	4.1	1.20	No

In [20]:

```
1 #option-2
2 df1 = df.drop(index = 0)
3 df1
```

Out[20]:

	Age	Gender	TB	DB	Alkphos	Sgpt	Sgot	TP	ALB	AG	LiverPatient
1	62	Male	10.9	5.5	699	64	100	7.5	3.2	0.74	Yes
2	62	Male	7.3	4.1	490	60	68	7.0	3.3	0.89	Yes
3	58	Male	1.0	0.4	182	14	20	6.8	3.4	NaN	Yes
4	72	NaN	3.9	2.0	195	27	59	7.3	2.4	0.40	Yes
5	46	Male	1.8	0.7	208	19	14	7.6	4.4	1.30	Yes
6	26	NaN	0.9	0.2	154	16	12	7.0	3.5	NaN	Yes
7	29	Female	NaN	0.3	202	14	11	6.7	3.6	1.10	Yes
8	17	Male	0.9	0.3	202	22	19	7.4	4.1	1.20	No

Drop multiple rows



In [21]:

```
1 #option-1
2 df2 = df.drop([0,1,2],axis=0)
3 df2
```

Out[21]:

	Age	Gender	TB	DB	Alkphos	Sgpt	Sgot	TP	ALB	AG	LiverPatient
3	58	Male	1.0	0.4	182	14	20	6.8	3.4	NaN	Yes
4	72	NaN	3.9	2.0	195	27	59	7.3	2.4	0.4	Yes
5	46	Male	1.8	0.7	208	19	14	7.6	4.4	1.3	Yes
6	26	NaN	0.9	0.2	154	16	12	7.0	3.5	NaN	Yes
7	29	Female	NaN	0.3	202	14	11	6.7	3.6	1.1	Yes
8	17	Male	0.9	0.3	202	22	19	7.4	4.1	1.2	No

In [22]:

```
1 #option-2
2 df2 = df.drop(index=[0,1,2])
3 df2
```

Out[22]:

	Age	Gender	TB	DB	Alkphos	Sgpt	Sgot	TP	ALB	AG	LiverPatient
3	58	Male	1.0	0.4	182	14	20	6.8	3.4	NaN	Yes
4	72	NaN	3.9	2.0	195	27	59	7.3	2.4	0.4	Yes
5	46	Male	1.8	0.7	208	19	14	7.6	4.4	1.3	Yes
6	26	NaN	0.9	0.2	154	16	12	7.0	3.5	NaN	Yes
7	29	Female	NaN	0.3	202	14	11	6.7	3.6	1.1	Yes
8	17	Male	0.9	0.3	202	22	19	7.4	4.1	1.2	No

Drop a Column



In [23]:

```
1 #option-1
2 df1 = df.drop("DB",axis=1)
3 df1
```

Out[23]:

	Age	Gender	TB	Alkphos	Sgpt	Sgot	TP	ALB	AG	LiverPatient
0	65	Female	0.7	187	16	18	6.8	3.3	0.90	Yes
1	62	Male	10.9	699	64	100	7.5	3.2	0.74	Yes
2	62	Male	7.3	490	60	68	7.0	3.3	0.89	Yes
3	58	Male	1.0	182	14	20	6.8	3.4	NaN	Yes
4	72	NaN	3.9	195	27	59	7.3	2.4	0.40	Yes
5	46	Male	1.8	208	19	14	7.6	4.4	1.30	Yes
6	26	NaN	0.9	154	16	12	7.0	3.5	NaN	Yes
7	29	Female	NaN	202	14	11	6.7	3.6	1.10	Yes
8	17	Male	0.9	202	22	19	7.4	4.1	1.20	No

In [24]:

```
1 #option-2
2 df1 = df.drop(columns="DB")
3 df1
```

Out[24]:

	Age	Gender	TB	Alkphos	Sgpt	Sgot	TP	ALB	AG	LiverPatient
0	65	Female	0.7	187	16	18	6.8	3.3	0.90	Yes
1	62	Male	10.9	699	64	100	7.5	3.2	0.74	Yes
2	62	Male	7.3	490	60	68	7.0	3.3	0.89	Yes
3	58	Male	1.0	182	14	20	6.8	3.4	NaN	Yes
4	72	NaN	3.9	195	27	59	7.3	2.4	0.40	Yes
5	46	Male	1.8	208	19	14	7.6	4.4	1.30	Yes
6	26	NaN	0.9	154	16	12	7.0	3.5	NaN	Yes
7	29	Female	NaN	202	14	11	6.7	3.6	1.10	Yes
8	17	Male	0.9	202	22	19	7.4	4.1	1.20	No

Deleting multiple columns



In [25]:

```
1 #option-1
2 df2 = df.drop(["TB", "DB"], axis=1)
3 df2
```

Out[25]:

	Age	Gender	Alkphos	Sgpt	Sgot	TP	ALB	AG	LiverPatient
0	65	Female	187	16	18	6.8	3.3	0.90	Yes
1	62	Male	699	64	100	7.5	3.2	0.74	Yes
2	62	Male	490	60	68	7.0	3.3	0.89	Yes
3	58	Male	182	14	20	6.8	3.4	NaN	Yes
4	72	NaN	195	27	59	7.3	2.4	0.40	Yes
5	46	Male	208	19	14	7.6	4.4	1.30	Yes
6	26	NaN	154	16	12	7.0	3.5	NaN	Yes
7	29	Female	202	14	11	6.7	3.6	1.10	Yes
8	17	Male	202	22	19	7.4	4.1	1.20	No

In [26]:

```
1 #option-2
2 df2 = df.drop(columns=["TB", "DB"])
3 df2
```

Out[26]:

	Age	Gender	Alkphos	Sgpt	Sgot	TP	ALB	AG	LiverPatient
0	65	Female	187	16	18	6.8	3.3	0.90	Yes
1	62	Male	699	64	100	7.5	3.2	0.74	Yes
2	62	Male	490	60	68	7.0	3.3	0.89	Yes
3	58	Male	182	14	20	6.8	3.4	NaN	Yes
4	72	NaN	195	27	59	7.3	2.4	0.40	Yes
5	46	Male	208	19	14	7.6	4.4	1.30	Yes
6	26	NaN	154	16	12	7.0	3.5	NaN	Yes
7	29	Female	202	14	11	6.7	3.6	1.10	Yes
8	17	Male	202	22	19	7.4	4.1	1.20	No

Add a new Column



In [27]:

```
1 df[ 'IB' ] = df[ "TB"]-df[ "DB"]
2 df
```

Out[27]:

	Age	Gender	TB	DB	Alkphos	Sgpt	Sgot	TP	ALB	AG	LiverPatient	IB
0	65	Female	0.7	0.1	187	16	18	6.8	3.3	0.90	Yes	0.6
1	62	Male	10.9	5.5	699	64	100	7.5	3.2	0.74	Yes	5.4
2	62	Male	7.3	4.1	490	60	68	7.0	3.3	0.89	Yes	3.2
3	58	Male	1.0	0.4	182	14	20	6.8	3.4	NaN	Yes	0.6
4	72	NaN	3.9	2.0	195	27	59	7.3	2.4	0.40	Yes	1.9
5	46	Male	1.8	0.7	208	19	14	7.6	4.4	1.30	Yes	1.1
6	26	NaN	0.9	0.2	154	16	12	7.0	3.5	NaN	Yes	0.7
7	29	Female	NaN	0.3	202	14	11	6.7	3.6	1.10	Yes	NaN
8	17	Male	0.9	0.3	202	22	19	7.4	4.1	1.20	No	0.6

In [28]:

```
1 df = df.drop(columns="IB")
2 df
```

Out[28]:

	Age	Gender	TB	DB	Alkphos	Sgpt	Sgot	TP	ALB	AG	LiverPatient
0	65	Female	0.7	0.1	187	16	18	6.8	3.3	0.90	Yes
1	62	Male	10.9	5.5	699	64	100	7.5	3.2	0.74	Yes
2	62	Male	7.3	4.1	490	60	68	7.0	3.3	0.89	Yes
3	58	Male	1.0	0.4	182	14	20	6.8	3.4	NaN	Yes
4	72	NaN	3.9	2.0	195	27	59	7.3	2.4	0.40	Yes
5	46	Male	1.8	0.7	208	19	14	7.6	4.4	1.30	Yes
6	26	NaN	0.9	0.2	154	16	12	7.0	3.5	NaN	Yes
7	29	Female	NaN	0.3	202	14	11	6.7	3.6	1.10	Yes
8	17	Male	0.9	0.3	202	22	19	7.4	4.1	1.20	No



In [29]:

```
1 df.drop(columns=["TB"], inplace=True)
2 df
```

Out[29]:

	Age	Gender	DB	Alkphos	Sgpt	Sgot	TP	ALB	AG	LiverPatient
0	65	Female	0.1	187	16	18	6.8	3.3	0.90	Yes
1	62	Male	5.5	699	64	100	7.5	3.2	0.74	Yes
2	62	Male	4.1	490	60	68	7.0	3.3	0.89	Yes
3	58	Male	0.4	182	14	20	6.8	3.4	NaN	Yes
4	72	NaN	2.0	195	27	59	7.3	2.4	0.40	Yes
5	46	Male	0.7	208	19	14	7.6	4.4	1.30	Yes
6	26	NaN	0.2	154	16	12	7.0	3.5	NaN	Yes
7	29	Female	0.3	202	14	11	6.7	3.6	1.10	Yes
8	17	Male	0.3	202	22	19	7.4	4.1	1.20	No

In [30]:

```
1 df = pd.read_csv("Liver.csv")
```

Sort Function

In [31]:

```
1 #Sorting in ascending order.
2 df.sort_values(by='DB', ascending=True)
```

Out[31]:

	Age	Gender	TB	DB	Alkphos	Sgpt	Sgot	TP	ALB	AG	LiverPatient
0	65	Female	0.7	0.1	187	16	18	6.8	3.3	0.90	Yes
6	26	NaN	0.9	0.2	154	16	12	7.0	3.5	NaN	Yes
7	29	Female	NaN	0.3	202	14	11	6.7	3.6	1.10	Yes
8	17	Male	0.9	0.3	202	22	19	7.4	4.1	1.20	No
3	58	Male	1.0	0.4	182	14	20	6.8	3.4	NaN	Yes
5	46	Male	1.8	0.7	208	19	14	7.6	4.4	1.30	Yes
4	72	NaN	3.9	2.0	195	27	59	7.3	2.4	0.40	Yes
2	62	Male	7.3	4.1	490	60	68	7.0	3.3	0.89	Yes
1	62	Male	10.9	5.5	699	64	100	7.5	3.2	0.74	Yes



In [32]:

```
1 # Sorting in descending order.  
2 df.sort_values(by='TP', ascending=False)
```

Out[32]:

	Age	Gender	TB	DB	Alkphos	Sgpt	Sgot	TP	ALB	AG	LiverPatient
5	46	Male	1.8	0.7	208	19	14	7.6	4.4	1.30	Yes
1	62	Male	10.9	5.5	699	64	100	7.5	3.2	0.74	Yes
8	17	Male	0.9	0.3	202	22	19	7.4	4.1	1.20	No
4	72	NaN	3.9	2.0	195	27	59	7.3	2.4	0.40	Yes
2	62	Male	7.3	4.1	490	60	68	7.0	3.3	0.89	Yes
6	26	NaN	0.9	0.2	154	16	12	7.0	3.5	NaN	Yes
0	65	Female	0.7	0.1	187	16	18	6.8	3.3	0.90	Yes
3	58	Male	1.0	0.4	182	14	20	6.8	3.4	NaN	Yes
7	29	Female	NaN	0.3	202	14	11	6.7	3.6	1.10	Yes

Data Extraction

This can be done by 3 ways

- 1. by using conditions
- 2. by using iloc() & loc()
- 3. combination of condition & loc()

using 1 condition

In [33]:

```
1 #condition  
2  
3 df["Gender"] == "Male"
```

Out[33]:

```
0    False  
1    True  
2    True  
3    True  
4   False  
5    True  
6   False  
7   False  
8    True  
Name: Gender, dtype: bool
```



In [34]:

```
1 #Syntax: df[condition]
2
3 df[df[ "Gender" ]=="Male"]
```

Out[34]:

	Age	Gender	TB	DB	Alkphos	Sgpt	Sgot	TP	ALB	AG	LiverPatient
1	62	Male	10.9	5.5	699	64	100	7.5	3.2	0.74	Yes
2	62	Male	7.3	4.1	490	60	68	7.0	3.3	0.89	Yes
3	58	Male	1.0	0.4	182	14	20	6.8	3.4	NaN	Yes
5	46	Male	1.8	0.7	208	19	14	7.6	4.4	1.30	Yes
8	17	Male	0.9	0.3	202	22	19	7.4	4.1	1.20	No

In [35]:

```
1 male_data=df[df[ "Gender" ]=="Male"]
2 male_data
```

Out[35]:

	Age	Gender	TB	DB	Alkphos	Sgpt	Sgot	TP	ALB	AG	LiverPatient
1	62	Male	10.9	5.5	699	64	100	7.5	3.2	0.74	Yes
2	62	Male	7.3	4.1	490	60	68	7.0	3.3	0.89	Yes
3	58	Male	1.0	0.4	182	14	20	6.8	3.4	NaN	Yes
5	46	Male	1.8	0.7	208	19	14	7.6	4.4	1.30	Yes
8	17	Male	0.9	0.3	202	22	19	7.4	4.1	1.20	No

In [36]:

```
1 df[df[ 'Age' ]>=70]
```

Out[36]:

	Age	Gender	TB	DB	Alkphos	Sgpt	Sgot	TP	ALB	AG	LiverPatient
4	72	NaN	3.9	2.0	195	27	59	7.3	2.4	0.4	Yes

Multiple Conditions



In [37]:

```
1 filter1=df[(df['Age']==65) | (df['Gender']=="Female")]
2 filter1
```

Out[37]:

	Age	Gender	TB	DB	Alkphos	Sgpt	Sgot	TP	ALB	AG	LiverPatient
0	65	Female	0.7	0.1	187	16	18	6.8	3.3	0.9	Yes
7	29	Female	NaN	0.3	202	14	11	6.7	3.6	1.1	Yes

In [38]:

```
1 filter2=df[(df['LiverPatient']=="Yes") & (df.Age>=70)]
2 filter2
```

Out[38]:

	Age	Gender	TB	DB	Alkphos	Sgpt	Sgot	TP	ALB	AG	LiverPatient
4	72	NaN	3.9	2.0	195	27	59	7.3	2.4	0.4	Yes

In [39]:

```
1 filter3=df[(df['Gender']=="Female") | (df['Age']>=35) & (df['DB']<=6)]
2 filter3
```

Out[39]:

	Age	Gender	TB	DB	Alkphos	Sgpt	Sgot	TP	ALB	AG	LiverPatient
0	65	Female	0.7	0.1	187	16	18	6.8	3.3	0.90	Yes
1	62	Male	10.9	5.5	699	64	100	7.5	3.2	0.74	Yes
2	62	Male	7.3	4.1	490	60	68	7.0	3.3	0.89	Yes
3	58	Male	1.0	0.4	182	14	20	6.8	3.4	NaN	Yes
4	72	NaN	3.9	2.0	195	27	59	7.3	2.4	0.40	Yes
5	46	Male	1.8	0.7	208	19	14	7.6	4.4	1.30	Yes
7	29	Female	NaN	0.3	202	14	11	6.7	3.6	1.10	Yes

iloc

-- works on the row index number & column index number



In [40]:

```
1 # get the first row
2 df.iloc[0]
```

Out[40]:

```
Age           65
Gender        Female
TB            0.7
DB            0.1
Alkphos       187
Sgpt           16
Sgot          18
TP             6.8
ALB            3.3
AG             0.9
LiverPatient  Yes
Name: 0, dtype: object
```

In [41]:

```
1 df.iloc[-1]
```

Out[41]:

```
Age           17
Gender        Male
TB            0.9
DB            0.3
Alkphos      202
Sgpt          22
Sgot          19
TP             7.4
ALB            4.1
AG             1.2
LiverPatient  No
Name: 8, dtype: object
```

In [42]:

```
1 # select the first, 100th, and 500th row
2 df.iloc[[3,7,8]]
```

Out[42]:

	Age	Gender	TB	DB	Alkphos	Sgpt	Sgot	TP	ALB	AG	LiverPatient
3	58	Male	1.0	0.4	182	14	20	6.8	3.4	NaN	Yes
7	29	Female	Nan	0.3	202	14	11	6.7	3.6	1.1	Yes
8	17	Male	0.9	0.3	202	22	19	7.4	4.1	1.2	No



In [43]:

```
1 #Displaying multiple specified columns and rows.  
2 df.iloc[[3,7,8],[1,4]]
```

Out[43]:

	Gender	Alkphos
3	Male	182
7	Female	202
8	Male	202

In [44]:

```
1 # select the 4th row & 3rd Column  
2 df.iloc[3,2]
```

Out[44]:

1.0

In [45]:

```
1 #Displaying specific column of range of rows.  
2 df.iloc[0:8,8]
```

Out[45]:

```
0    3.3  
1    3.2  
2    3.3  
3    3.4  
4    2.4  
5    4.4  
6    3.5  
7    3.6  
Name: ALB, dtype: float64
```

loc

In [46]:



```
1 # get the first row
2 print(df.loc[0])
```

```
Age          65
Gender      Female
TB           0.7
DB           0.1
Alkphos     187
Sgpt          16
Sgot         18
TP            6.8
ALB           3.3
AG            0.9
LiverPatient Yes
Name: 0, dtype: object
```

"Data Science & AI"
"Data Science by
Siva Rama Krishna"



In [47]:

```
1 # get the 100th row
2 # recall that values start with 0
3 print(df.loc[-1])
```

```
-----  
ValueError                                     Traceback (most recent call last)
File ~\anaconda3\lib\site-packages\pandas\core\indexes\range.py:385, in RangeIndex.get_loc(self, key, method, tolerance)
    384     try:
--> 385         return self._range.index(new_key)
    386     except ValueError as err:
```

ValueError: -1 is not in range

The above exception was the direct cause of the following exception:

```
KeyError                                     Traceback (most recent call last)
```

```
Input In [47], in <cell line: 3>()
    1 # get the 100th row
    2 # recall that values start with 0
----> 3 print(df.loc[-1])
```

```
File ~\anaconda3\lib\site-packages\pandas\core\indexing.py:967, in _LocationIndexer.__getitem__(self, key)
    964     axis = self.axis or 0
    965     maybe_callable = com.apply_if_callable(key, self.obj)
--> 967     return self._getitem_axis(maybe_callable, axis=axis)
```

```
File ~\anaconda3\lib\site-packages\pandas\core\indexing.py:1202, in _LocIndexer._getitem_axis(self, key, axis)
   1200     # fall thru to straight lookup
   1201     self._validate_key(key, axis)
-> 1202     return self._get_label(key, axis=axis)
```

```
File ~\anaconda3\lib\site-packages\pandas\core\indexing.py:1153, in _LocIndexer._get_label(self, label, axis)
   1151     def _get_label(self, label, axis: int):
   1152         # GH#5667 this will fail if the label is not present in the axis.
-> 1153         return self.obj.xs(label, axis=axis)
```

```
File ~\anaconda3\lib\site-packages\pandas\core\generic.py:3864, in NDFrame.xs(self, key, axis, level, drop_level)
   3862             new_index = index[loc]
   3863     else:
-> 3864         loc = index.get_loc(key)
   3865         if isinstance(loc, np.ndarray):
   3866             if loc.dtype == np.bool_:
```

```
File ~\anaconda3\lib\site-packages\pandas\core\indexes\range.py:387, in RangeIndex.get_loc(self, key, method, tolerance)
    385         return self._range.index(new_key)
    386     except ValueError as err:
--> 387         raise KeyError(key) from err
    388     self._check_indexing_error(key)
    389 raise KeyError(key)
```

KeyError: -1



In [48]:

```
1 #Retrieving selected rows with range of columns between 'TB' and 'TP'
2 df.loc[[5,6], 'TB':'TP']
```

Out[48]:

	TB	DB	Alkphos	Sgpt	Sgot	TP
5	1.8	0.7	208	19	14	7.6
6	0.9	0.2	154	16	12	7.0

In [49]:

```
1 df.loc[:, "TB"]
```

Out[49]:

```
0      0.7
1     10.9
2      7.3
3      1.0
4      3.9
5      1.8
6      0.9
7      NaN
8      0.9
Name: TB, dtype: float64
```

Just make sure you don't confuse the differences between loc and iloc

In [50]:

```
1 df.loc[2, "Age"]
```

Out[50]:

62

In [51]:

```
1 df.iloc[2,0]
```

Out[51]:

62

In [52]:

```
1 df.iloc[[4,5], 0:4]
```

Out[52]:

	Age	Gender	TB	DB
4	72	NaN	3.9	2.0
5	46	Male	1.8	0.7



In [53]:

```
1 df.loc[[4,5],"Age":"DB"]
```

Out[53]:

	Age	Gender	TB	DB
4	72	NaN	3.9	2.0
5	46	Male	1.8	0.7

SetIndex

In [54]:

```
1 l1= df.set_index("TB")
2 l1
```

Out[54]:

	Age	Gender	DB	Alkphos	Sgpt	Sgot	TP	ALB	AG	LiverPatient
TB										
0.7	65	Female	0.1	187	16	18	6.8	3.3	0.90	Yes
10.9	62	Male	5.5	699	64	100	7.5	3.2	0.74	Yes
7.3	62	Male	4.1	490	60	68	7.0	3.3	0.89	Yes
1.0	58	Male	0.4	182	14	20	6.8	3.4	NaN	Yes
3.9	72	NaN	2.0	195	27	59	7.3	2.4	0.40	Yes
1.8	46	Male	0.7	208	19	14	7.6	4.4	1.30	Yes
0.9	26	NaN	0.2	154	16	12	7.0	3.5	NaN	Yes
NaN	29	Female	0.3	202	14	11	6.7	3.6	1.10	Yes
0.9	17	Male	0.3	202	22	19	7.4	4.1	1.20	No

In [55]:

```
1 l1.loc[0.9]
```

Out[55]:

	Age	Gender	DB	Alkphos	Sgpt	Sgot	TP	ALB	AG	LiverPatient
TB										
0.9	26	NaN	0.2	154	16	12	7.0	3.5	NaN	Yes
0.9	17	Male	0.3	202	22	19	7.4	4.1	1.2	No



In [56]:

```
1 l1.loc[1.0, 'DB':'ALB']
```

Out[56]:

```
DB      0.4
Alkphos 182
Sgpt    14
Sgot    20
TP      6.8
ALB     3.4
Name: 1.0, dtype: object
```

In [57]:

```
1 l1.reset_index()
```

Out[57]:

	TB	Age	Gender	DB	Alkphos	Sgpt	Sgot	TP	ALB	AG	LiverPatient
0	0.7	65	Female	0.1	187	16	18	6.8	3.3	0.90	Yes
1	10.9	62	Male	5.5	699	64	100	7.5	3.2	0.74	Yes
2	7.3	62	Male	4.1	490	60	68	7.0	3.3	0.89	Yes
3	1.0	58	Male	0.4	182	14	20	6.8	3.4	NaN	Yes
4	3.9	72	NaN	2.0	195	27	59	7.3	2.4	0.40	Yes
5	1.8	46	Male	0.7	208	19	14	7.6	4.4	1.30	Yes
6	0.9	26	NaN	0.2	154	16	12	7.0	3.5	NaN	Yes
7	NaN	29	Female	0.3	202	14	11	6.7	3.6	1.10	Yes
8	0.9	17	Male	0.3	202	22	19	7.4	4.1	1.20	No

In [58]:

```
1 #Using= condition for selected columns.
2 df1= df.loc[df['DB']>=3.7,['Age','DB','LiverPatient']]
3 df1
```

Out[58]:

	Age	DB	LiverPatient
1	62	5.5	Yes
2	62	4.1	Yes