Data Manipulation

In [29]:
```
#Aim : To perform operation of Data Manipulation on Data set.
```

In [30]:
```
# Name : Shruti A. Dhote
# Roll no. : 72
# Sec : C
# Subject : ET1
#date: 02/08/2024
```

In [31]:
```
import pandas as pd
```

In [32]:
```
import os
```

In [33]:
```
os.getcwd()
```

Out[33]:
```
'C:\\Users\\SURUTI DHOTE\\Desktop'
```

In [34]:
```
os.chdir("C:\\Users\\SURUTI DHOTE\\Desktop")
```

In [35]:
```
data  = pd.read_csv("diabetes.csv")
```

In [36]:
```
data.head(10)
```

Out[36]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	A
0	6	148	72	35	0	33.6	0.627	
1	1	85	66	29	0	26.6	0.351	
2	8	183	64	0	0	23.3	0.672	
3	1	89	66	23	94	28.1	0.167	
4	0	137	40	35	168	43.1	2.288	
5	5	116	74	0	0	25.6	0.201	
6	3	78	50	32	88	31.0	0.248	
7	10	115	0	0	0	35.3	0.134	
8	2	197	70	45	543	30.5	0.158	
9	8	125	96	0	0	0.0	0.232	

In [37]:
```
data.tail()
```

Out[37]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction
763	10	101	76	48	180	32.9	0.171
764	2	122	70	27	0	36.8	0.340
765	5	121	72	23	112	26.2	0.245
766	1	126	60	0	0	30.1	0.349
767	1	93	70	31	0	30.4	0.315

Pandas dimension of dataframes/series which is equivalent to total number of elements.

Returns tuple of shape (Rows, columns) of data.

In [38]:
```
data.shape
```

Out[38]:
```
(768, 9)
```

Returns size of dataframe/series which is equivalent to total number of elements.

That is rows x columns.

In [39]:
```
data.size
```

Out[39]:
```
6912
```

Returns size of dataframe/series. 1 is for one dimension. 2 is for two dimension.

In [40]:
```
data.ndim
```

Out[40]:
```
2
```

In [41]:
```
data.columns
```

Out[41]:
```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

In [42]:
```
data.head()
```

Out[42]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	A
0	6	148	72	35	0	33.6	0.627	
1	1	85	66	29	0	26.6	0.351	
2	8	183	64	0	0	23.3	0.672	
3	1	89	66	23	94	28.1	0.167	
4	0	137	40	35	168	43.1	2.288	

Drop is used to drop one or more than one column from a Data

axis = 1 i.e Column

In [43]:
```
data.drop(labels ="Age",axis = 1)
```

Out[43]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction
0	6	148	72	35	0	33.6	0.627
1	1	85	66	29	0	26.6	0.351
2	8	183	64	0	0	23.3	0.672
3	1	89	66	23	94	28.1	0.167
4	0	137	40	35	168	43.1	2.288
...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171
764	2	122	70	27	0	36.8	0.340
765	5	121	72	23	112	26.2	0.245
766	1	126	60	0	0	30.1	0.349
767	1	93	70	31	0	30.4	0.315

768 rows × 8 columns

In [44]:
```
data.drop(labels = ["Age","Glucose"], axis =1)
```

Out[44]:

	Pregnancies	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Outcome
0	6	72	35	0	33.6	0.627	1
1	1	66	29	0	26.6	0.351	0
2	8	64	0	0	23.3	0.672	1
3	1	66	23	94	28.1	0.167	0
4	0	40	35	168	43.1	2.288	1
...	...	...	...	...	...	...	...
763	10	76	48	180	32.9	0.171	0
764	2	70	27	0	36.8	0.340	0
765	5	72	23	112	26.2	0.245	0
766	1	60	0	0	30.1	0.349	1
767	1	70	31	0	30.4	0.315	0

768 rows × 7 columns

In [17]:
```
data.drop(labels = 1,axis =0)
```

Out[17]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction
0	6	148	72	35	0	33.6	0.627
2	8	183	64	0	0	23.3	0.672
3	1	89	66	23	94	28.1	0.167
4	0	137	40	35	168	43.1	2.288
5	5	116	74	0	0	25.6	0.201
...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171
764	2	122	70	27	0	36.8	0.340
765	5	121	72	23	112	26.2	0.245
766	1	126	60	0	0	30.1	0.349
767	1	93	70	31	0	30.4	0.315

767 rows × 9 columns

In [18]:
```
data.head(10)
```

Out[18]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	A
0	6	148	72	35	0	33.6	0.627	
1	1	85	66	29	0	26.6	0.351	
2	8	183	64	0	0	23.3	0.672	
3	1	89	66	23	94	28.1	0.167	
4	0	137	40	35	168	43.1	2.288	
5	5	116	74	0	0	25.6	0.201	
6	3	78	50	32	88	31.0	0.248	
7	10	115	0	0	0	35.3	0.134	
8	2	197	70	45	543	30.5	0.158	
9	8	125	96	0	0	0.0	0.232	

Drop is use to drop one or more than one column from a Data axis = 0 i.e Row

In [22]:
```
data.drop(labels = [2,3], axis = 0)
```

Out[22]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction
0	6	148	72	35	0	33.6	0.627
1	1	85	66	29	0	26.6	0.351
4	0	137	40	35	168	43.1	2.288
5	5	116	74	0	0	25.6	0.201
6	3	78	50	32	88	31.0	0.248
...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171
764	2	122	70	27	0	36.8	0.340
765	5	121	72	23	112	26.2	0.245
766	1	126	60	0	0	30.1	0.349
767	1	93	70	31	0	30.4	0.315

766 rows × 9 columns