data manipulation

In [1]:
```
#Name : Vaibhav Laxman Karale
#Roll no. 58
#sub:E.T.1
#Section :3A
#Date:24/08/2024
```

In [2]:
```
# Aim: to perform data manipulation
```

In [3]:
```
import pandas as pd
```

In [4]:
```
import os
```

In [5]:
```
os.getcwd()
```

Out[5]:
```
'C:\\Users\\DELL'
```

In [6]:
```
os.chdir("C:\\Users\\DELL\\OneDrive\\Desktop")
```

In [7]:
```
data=pd.read_csv("diabetes.csv")
```

In [8]:
```
data.head(18)
```

Out[8]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunctio
0	6	148	72	35	0	33.6	0.62
1	1	85	66	29	0	26.6	0.35
2	8	183	64	0	0	23.3	0.67
3	1	89	66	23	94	28.1	0.16
4	0	137	40	35	168	43.1	2.28
5	5	116	74	0	0	25.6	0.20
6	3	78	50	32	88	31.0	0.24
7	10	115	0	0	0	35.3	0.13
8	2	197	70	45	543	30.5	0.15
9	8	125	96	0	0	0.0	0.23
10	4	110	92	0	0	37.6	0.19
11	10	168	74	0	0	38.0	0.53
12	10	139	80	0	0	27.1	1.44
13	1	189	60	23	846	30.1	0.39
14	5	166	72	19	175	25.8	0.58
15	7	100	0	0	0	30.0	0.48
16	0	118	84	47	230	45.8	0.55
17	7	107	74	0	0	29.6	0.25

In [9]:
```
data.tail(12)
```

Out[9]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunctio
756	7	137	90	41	0	32.0	0.3
757	0	123	72	0	0	36.3	0.2
758	1	106	76	0	0	37.5	0.1
759	6	190	92	0	0	35.5	0.2
760	2	88	58	26	16	28.4	0.7
761	9	170	74	31	0	44.0	0.4
762	9	89	62	0	0	22.5	0.1
763	10	101	76	48	180	32.9	0.1
764	2	122	70	27	0	36.8	0.3
765	5	121	72	23	112	26.2	0.2
766	1	126	60	0	0	30.1	0.3
767	1	93	70	31	0	30.4	0.3

In [10]:
```
data.describe()
```

Out[10]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Diabetes
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	

In [11]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Pregnancies               768 non-null    int64  
 1   Glucose                   768 non-null    int64  
 2   BloodPressure             768 non-null    int64  
 3   SkinThickness             768 non-null    int64  
 4   Insulin                   768 non-null    int64  
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64  
 8   Outcome                   768 non-null    int64  
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

In [12]:
```
data.isna()
```

Out[12]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunct
0	False	False	False	False	False	False	Fa
1	False	False	False	False	False	False	Fa
2	False	False	False	False	False	False	Fa
3	False	False	False	False	False	False	Fa
4	False	False	False	False	False	False	Fa
...	...	...	...	...	...	...	
763	False	False	False	False	False	False	Fa
764	False	False	False	False	False	False	Fa
765	False	False	False	False	False	False	Fa
766	False	False	False	False	False	False	Fa
767	False	False	False	False	False	False	Fa

768 rows × 9 columns

In [13]:
```
data.shape
```

Out[13]: (768, 9)

In [13]:
```
data.size
```

Out[13]: 6912

In [14]:
```
data.columns
```

Out[14]:
```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

In [15]:
```
data.drop(labels="Age",axis=1)
```

Out[15]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunctio
0	6	148	72	35	0	33.6	0.6
1	1	85	66	29	0	26.6	0.3
2	8	183	64	0	0	23.3	0.6
3	1	89	66	23	94	28.1	0.1
4	0	137	40	35	168	43.1	2.2
...	...	...	...	...	...	...	
763	10	101	76	48	180	32.9	0.1
764	2	122	70	27	0	36.8	0.3
765	5	121	72	23	112	26.2	0.2
766	1	126	60	0	0	30.1	0.3
767	1	93	70	31	0	30.4	0.3

768 rows × 8 columns

In [16]:
```
data.drop(labels=["Age","Glucose"],axis=1)
```

Out[16]:

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

768 rows × 7 columns

In [17]:
```
data.drop(labels=2,axis=0)
```

Out[17]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunctio
0	6	148	72	35	0	33.6	0.6
1	1	85	66	29	0	26.6	0.3
3	1	89	66	23	94	28.1	0.1
4	0	137	40	35	168	43.1	2.2
5	5	116	74	0	0	25.6	0.2
...	...	...	...	...	...	...	
763	10	101	76	48	180	32.9	0.1
764	2	122	70	27	0	36.8	0.3
765	5	121	72	23	112	26.2	0.2
766	1	126	60	0	0	30.1	0.3
767	1	93	70	31	0	30.4	0.3

767 rows × 9 columns

In [18]:
```
data.drop(labels=[2,4],axis=0)
```

Out[18]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunctio
0	6	148	72	35	0	33.6	0.6
1	1	85	66	29	0	26.6	0.3
3	1	89	66	23	94	28.1	0.1
5	5	116	74	0	0	25.6	0.2
6	3	78	50	32	88	31.0	0.2
...	...	...	...	...	...	...	
763	10	101	76	48	180	32.9	0.1
764	2	122	70	27	0	36.8	0.3
765	5	121	72	23	112	26.2	0.2
766	1	126	60	0	0	30.1	0.3
767	1	93	70	31	0	30.4	0.3

766 rows × 9 columns

In [ ]:
```

```