In [17]:
```
#EXP -2
```

In [18]:
```
#Aim: To perform of getting basic statastical description of data
```

In [3]:
```
# Name: Payal Devanand Manwar
# Roll no.: 37
# Sec:A
# Subject:ET1
# Date: 25/08/2025
```

In [4]:
```
import pandas as pd
```

In [5]:
```
import os
```

In [6]:
```
os.getcwd()
```

Out[6]:
```
'C:\\Users\\USER'
```

In [7]:
```
os.chdir("C:\\Users\\USER\\Desktop")
```

In [8]:
```
data=pd.read_csv("diabetes - diabetes.csv.csv")
```

In [9]:
```
data.head(20)
```

Out[9]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Ag
0	6	148	72	35	0	33.6	0.627	5
1	1	85	66	29	0	26.6	0.351	3
2	8	183	64	0	0	23.3	0.672	3
3	1	89	66	23	94	28.1	0.167	2
4	0	137	40	35	168	43.1	2.288	3
5	5	116	74	0	0	25.6	0.201	3
6	3	78	50	32	88	31.0	0.248	2
7	10	115	0	0	0	35.3	0.134	2
8	2	197	70	45	543	30.5	0.158	5
9	8	125	96	0	0	0.0	0.232	5
10	4	110	92	0	0	37.6	0.191	3
11	10	168	74	0	0	38.0	0.537	3
12	10	139	80	0	0	27.1	1.441	5
13	1	189	60	23	846	30.1	0.398	5
14	5	166	72	19	175	25.8	0.587	5

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Ag
15	7	100	0	0	0	30.0	0.484	3
16	0	118	84	47	230	45.8	0.551	3
17	7	107	74	0	0	29.6	0.254	3
18	1	103	30	38	83	43.3	0.183	3
19	1	115	70	30	96	34.6	0.529	3

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

Out[10]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Ag
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	

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

Out[11]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedig
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 [12]:
```
data.shape
```

Out[12]:
```
(768, 9)
```

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

Out[13]:
```
6912
```

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

Out[14]:
```
2
```

In [15]:
```
data.columns
```

Out[15]:
```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

In [16]:
```
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 [ ]:
```

```