```
#univariate Analysis of diabetes dataset
print(‘MEAN:\n’,df.mean())
print(‘MEDIAN:\n:’,df.median())
print(‘MODE:\n:’,df.mode())
print(‘STANDAR DEVIATION:\n:’,df.std())
print(‘VARIANCE:\n:’,df.var())
print(‘SKEWNESS:\n:’,df.skew())
print(‘KURTOSIS:\n:’,df.kurtosis())
df.describe()
```

OUTPUT

```
    MEAN:
     Pregnancies                      3.845052
    Glucose                         120.894531
    BloodPressure                    69.105469
    SkinThickness                    20.536458
    Insulin                          79.799479
    BMI                              31.992578
    DiabetesPedigreeFunction          0.471876
    Age                              33.240885
    Outcome                           0.348958
    dtype: float64
    MEDIAN:
    : Pregnancies                     3.0000
    Glucose                         117.0000
    BloodPressure                    72.0000
    SkinThickness                    23.0000
    Insulin                          30.5000
    BMI                              32.0000
    DiabetesPedigreeFunction          0.3725
    Age                              29.0000
    Outcome                           0.0000
```

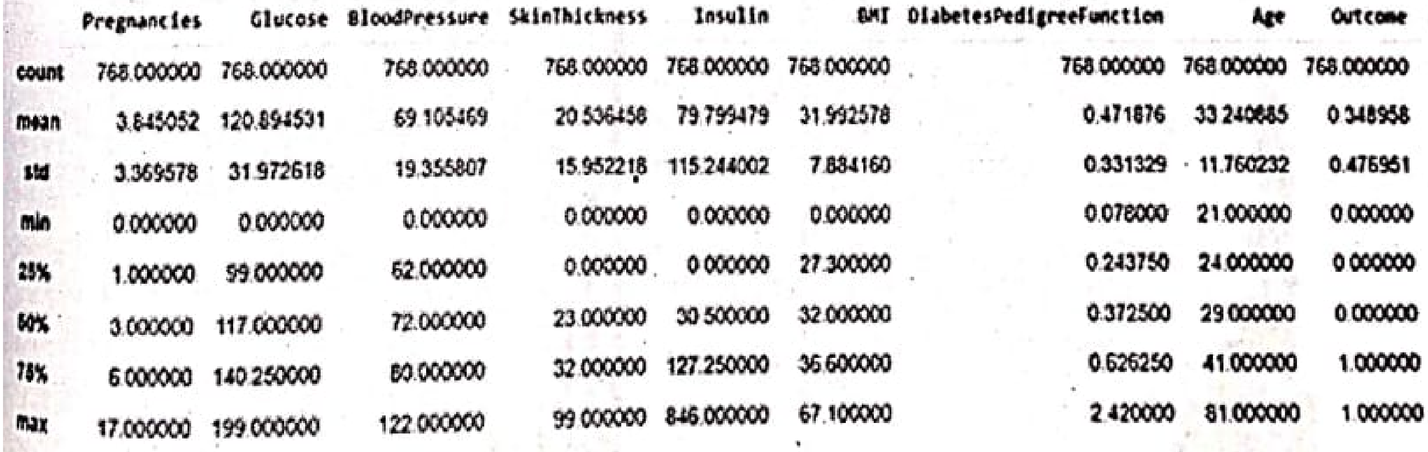
4. Use the diabetes data set from UCI Univariate analysis: Frequency, Mean, Median, Mode, Variance, Standard Deviation, Skewness and Kurtosis.

```
#Diabetes data set from UCI
#Univariate analysis: Frequency, Mean, Median, Mode, Variance, Standard
# Deviation, Skewness and Kurtosis.
import pandas as pd
import numpy as np
import statistics as st
# Load the data
df = pd.read_csv("diabetes.csv")
print(df.shape)
print(df.info())
```

OUTPUT:

```
    (768, 9)
    <class 'pandas.core.frame.DataFrame'>
    RangeIndex: 768 entries, 0 to 767
    Data columns (totai 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
None
```

```
Glucose                         0.173754
BloodPressure                  -1.843608
SkinThickness                   0.109372
Insulin                         2.272251
BMI                            -0.428982
DiabetesPedigreeFunction        1.919911
Age                             1.129597
Outcome                         0.635017
dtype: float64
KURTOSIS:
: Pregnancies                   0.159220
Glucose                         0.640780
BloodPressure                   5.180157
SkinThickness                  -0.520072
Insulin                         7.214260
BMI                             3.290443
DiabetesPedigreeFunction        5.594954
Age                             0.643159
Outcome                        -1.600930
dtype: float64
```



	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	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	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

```
liabetes dataset in histogram
ata_X = df.copy(deep =True)
ata_X = Data_X.drop(['Outcome'], axis = 1)
nport matplotlib.pyplot as plt
lt.rcParams['figure.figsize'] = [40, 40]
```

```
# Plotting Histogram of Data
Data_X.hist(bins = 40)
```