DIABETES DATABASE DESCRIPTIVE STATISTICS

CONTENTS

- INTRODUCTION
- OBJECTIVE
- SYSTEM ENVIRONMENT
- INDIANS DIABETES DATABASE DESCRIPTIVE STATISTICS PROJECT
- DIABETES DATASET EXPLORATIONS AS PER THE QUESTIONS HAS INSTRUCTED HERE BY
- SCREEN SHOTS OF THE PROJECT
- CONCLUSION

INTRODUCTION

WHAT IS DIABETES?

Conclusdiabetes and pre-diabetes are serious conditions in which people have high levels of sugar or glucose in their blood. The world health organization (WHO) reports that more than 420 million people worldwide live with diabetes. In the US, according to the US centers for disease control and prevention (CDC), over 30 million people have diabetes, and 88 million adults have pre-diabetes (blood sugar levels are higher than normal, but not high enough to be diagnosed with type 2 diabetes). Diabetes is a major cause of blindness, amputation, kidney failure, and cardiovascular disease.

Glucose is a type of sugar that is used as fuel by the body. When you eat, your body converts food into glucose. The glucose then goes into your bloodstream and is carried throughout the body to provide energy to all of your cells. In order for glucose to move from your bloodstream into your cells, you need insulin. Insulin carries the glucose, or sugar, in your bloodstream into your cells. Insulin is a hormone made by the pancreas, an organ in the upper part of your abdomen (belly).

If your body has a problem making or using insulin, the glucose in your bloodstream cannot get into your cells. As a result, glucose stays in the blood (high blood sugar) and the cells do not get enough glucose. A diagnosis of pre-diabetes or diabetes is made when glucose stays at higher-than-normal levels (also called hyperglycemia).

OBJECTIVE

- The data set used in the project is basically from the NIDDK ((National Institute of Diabetes and Digestive and Kidney Disease). The objective of this project to predict whether the person is diabetic or not based on several variables present in the data set.
- The main reason behind the person being diabetic is having high blood sugar or blood glucose. The main source of energy is glucose from the food we eat. Basically, a special cell in the pancreas produces a hormone called insulin. This insulin does a function to move glucose from the blood to cells for energy. If the pancreas not making enough insulin or having from the move to the cells then glucose kept in the blood that causes the diabetes problem.
- *The data consists of predictor variables and the target variable.*
- The Predictor variables are Glucose, Blood Pressure, Insulin, BMI, Pregnancies, Age, Skin thickness, diabetes pedigree function.
- The target variable is theOutcome. It is already in binary values (0 and 1). The value 0 represents for non-diabetic and 1 value represents diabetic

SYSTEM ENVIROMENT

HARDWARE REQUIREMENTS

- **PROCESSOR: INTEL DUAL CORE PROCESSOR**
- **HARD DISK: 1TB**
- **FLOPPY DRIVE: 1.42 MB (14,90,485 BYTES)**
- **MONITOR: LCD COLOR**
- **MOUSE: OPTICAL**
- **RAM: 8GB (8,192MB)**

SOFTWARE REQUIREMENTS

- **OPERATING SYSTEM: WINDOWS 10**
- **LANGUAGE: PYTHON3**
- **DATABASE: MYSQL**

INDIANS DIABETES DATABASE DESCRIPTIVE STATISTICS PROJECT

Predict the onset of diabetes based on diagnostic measures. This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether a patient has diabetes, based on certain diagnostic measurements included in the dataset.

Columns of the Dataset

- Pregnancies: Number of times pregnant
- Glucose: Plasma glucose concentration 2 hours in an oral glucose tolerance test
- Blood Pressure: Diastolic blood pressure (mm Hg)
- Skin Thickness: Triceps skin fold thickness (mm)
- Insulin: 2-Hour serum insulin (mu U/ml)
- BMI: Body mass index (weight in kg/ (height in m) ^2)
- Diabetes Pedigree Function: Diabetes pedigree function
- Age: Age (years)
- Outcome: Class variable (0 or 1) 0 mean non-diabetic and 1 means diabetic

DIABETES DATASET EXPLORATIONS AS PER THE QUESTIONS HAS INSTRUCTED HERE BY:

Q1: *PLEASE DO THE BASIC EXPLORATION OF DATA AND EXPLAIN MISSING VALUES, NUMBER OF ROWS AND COLUMNS AND DATA TYPES IN STATISTICAL TERM.*

Q2: *CALCULATE APPROPRIATE MEASURES OF CENTRAL TENDENCY FOR GLUCOSE AND OUTCOME COLUMN ONLY?*

Q3: *PLEASE PROVIDE 5 POINTS DATA SUMMARIES FOR REQUIRED COLUMNS?*

Q4: *PLEASE CREATE AN APPROPRIATE PLOT TO EXAMINE THE RELATIONSHIP*

BETWEEN AGE AND GLUCOSE.

Q5: *PLEASE CREATE AN APPROPRIATE PLOT TO SEE THE DISTRIBUTION OF OUTCOME VARIABLE?*

Q6: *PLEASE EXAMINE THE DISTRIBUTION OF NUMERICAL DATA AND EXPLAIN*

Q7: *WHICH VARIABLE NORMALLY DISTRIBUTED AND WHICH VARIABLE IS SEEMING TO BE SKEWED. PLEASE ALSO TELL THE DIRECTION OF SKEWNESS.*

Q8: *PLEASE CALCULATE THE SKEWNESS VALUE AND DIVIDE VARIABLES INTO SYMMETRICAL, MODERATELY SKEWED AND HIGHLY SKEWED.*

Q9: *PLEASE CREATE APPROPRIATE PLOT TO EXAMINE THE OUTLIERS OF THESE VARIABLES. PLEASE NAME THE VARIABLES WHICH HAVE OUTLIERS.*

SCREEN SHOTS OF THE PROJECT

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

```
[ ] df=pd.read_csv("C:/Users/Chinmay/Downloads/Diabetes.csv")
    df.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

```
df.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
```

Data describe

+ Code + Text

```
[ ]  df.describe()
```

	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

Findout missing value

```
[ ]  print((df==0).any().sum())
     print((df==0).sum())
```

```
7
Pregnancies                 111
Glucose                       5
BloodPressure                35
SkinThickness               227
Insulin                     374
BMI                          11
DiabetesPedigreeFunction      0
Age                           0
Outcome                     500
dtype: int64
```

There are 111 zero value rows and 7 zero value columns.

Because of zero value of insulin,glucose,Blood pressure,skin thickness,BMI columns are doesn't make sence ,we consider that as null value

```
[ ]  # we need drop insulin column becuse of maximum number of null value
     df1=df.drop('Insulin',axis =1)
     df1.shape
```

```
(768, 8)
```

We need to drop all rows with missing values .

```
[ ]  df=df[-(df[df.columns[1:-1]] == 0).any(axis=1)]
     df.head(20)
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
6	3	78	50	32	88	31.0	0.248	26	1
8	2	197	70	45	543	30.5	0.158	53	1
13	1	189	60	23	846	30.1	0.398	59	1
14	5	166	72	19	175	25.8	0.587	51	1
16	0	118	84	47	230	45.8	0.551	31	1
18	1	103	30	38	83	43.3	0.183	33	0
19	1	115	70	30	96	34.6	0.529	32	1
20	3	126	88	41	235	39.3	0.704	27	0
24	11	143	94	33	146	36.6	0.254	51	1
25	10	125	70	26	115	31.1	0.205	41	1
27	1	97	66	15	140	23.2	0.487	22	0

Appropriate measures of central tendency for Glucose and outcome column

```
[ ]  df.groupby('Outcome').agg(['mean','median'])
```

	Pregnancies		Glucose		BloodPressure		SkinThickness		Insulin		BMI		DiabetesPedigreeFunction		Age	
	mean	median	mean	median	mean	median	mean	median	mean	median	mean	median	mean	median	mean	median
Outcome																
0	2.721374	2	111.431298	107.5	68.969466	70	27.251908	27	130.854962	105.0	31.750763	31.25	0.472168	0.4135	28.347328	25
1	4.469231	3	145.192308	144.5	74.076923	74	32.961538	33	206.846154	169.5	35.777692	34.60	0.625585	0.5460	35.938462	33

```
[ ]  df.groupby('Glucose').agg(['mean','median'])
```

	Pregnancies		BloodPressure		SkinThickness		Insulin		BMI		DiabetesPedigreeFunction		Age		Outcome	
	mean	median	mean	median	mean	median	mean	median	mean	median	mean	median	mean	median	mean	median
Glucose																
56	2.000000	2.0	56.000000	56.0	28.000000	28.0	45.000000	45.0	24.200000	24.2	0.332000	0.3320	22.000000	22.0	0.0	0.0
68	4.666667	2.0	79.333333	70.0	22.666667	23.0	43.333333	49.0	26.866667	25.0	0.243000	0.2570	31.666667	25.0	0.0	0.0
71	1.000000	1.0	63.000000	63.0	34.000000	34.0	60.500000	60.5	26.800000	26.8	0.372500	0.3725	21.500000	21.5	0.0	0.0
74	3.666667	3.0	63.333333	68.0	26.000000	28.0	43.333333	45.0	30.933333	29.7	0.422333	0.2930	28.000000	23.0	0.0	0.0

Corelation between all variables

```
[ ]  df.corr()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1.000000	0.198291	0.213355	0.093209	0.078984	-0.025347	0.007562	0.679608	0.256566
Glucose	0.198291	1.000000	0.210027	0.198856	0.581223	0.209516	0.140180	0.343641	0.515703
BloodPressure	0.213355	0.210027	1.000000	0.232571	0.098512	0.304403	-0.015971	0.300039	0.192673
SkinThickness	0.093209	0.198856	0.232571	1.000000	0.182199	0.664355	0.160499	0.167761	0.255936
Insulin	0.078984	0.581223	0.098512	0.182199	1.000000	0.226397	0.135906	0.217082	0.301429
BMI	-0.025347	0.209516	0.304403	0.664355	0.226397	1.000000	0.158771	0.069814	0.270118
DiabetesPedigreeFunction	0.007562	0.140180	-0.015971	0.160499	0.135906	0.158771	1.000000	0.085029	0.209330
Age	0.679608	0.343641	0.300039	0.167761	0.217082	0.069814	0.085029	1.000000	0.350804
Outcome	0.256566	0.515703	0.192673	0.255936	0.301429	0.270118	0.209330	0.350804	1.000000

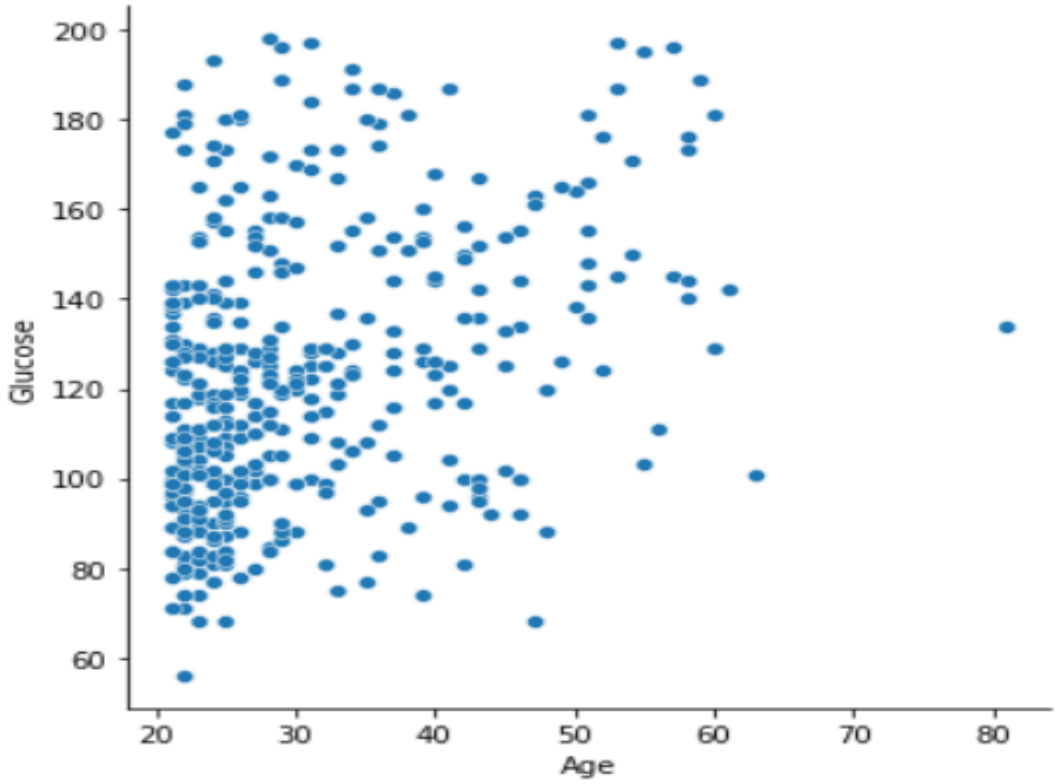
```
plt.figure(figsize=(9,9))
sns.heatmap(np.abs(df.corr()),annot=True,cmap="viridis",fmt="0.2f")
```

```
<AxesSubplot:>
```

Examine the relationship between Age and Glucose.

```
[ ]  sns.relplot(x='Age', y= 'Glucose', data=df)
```

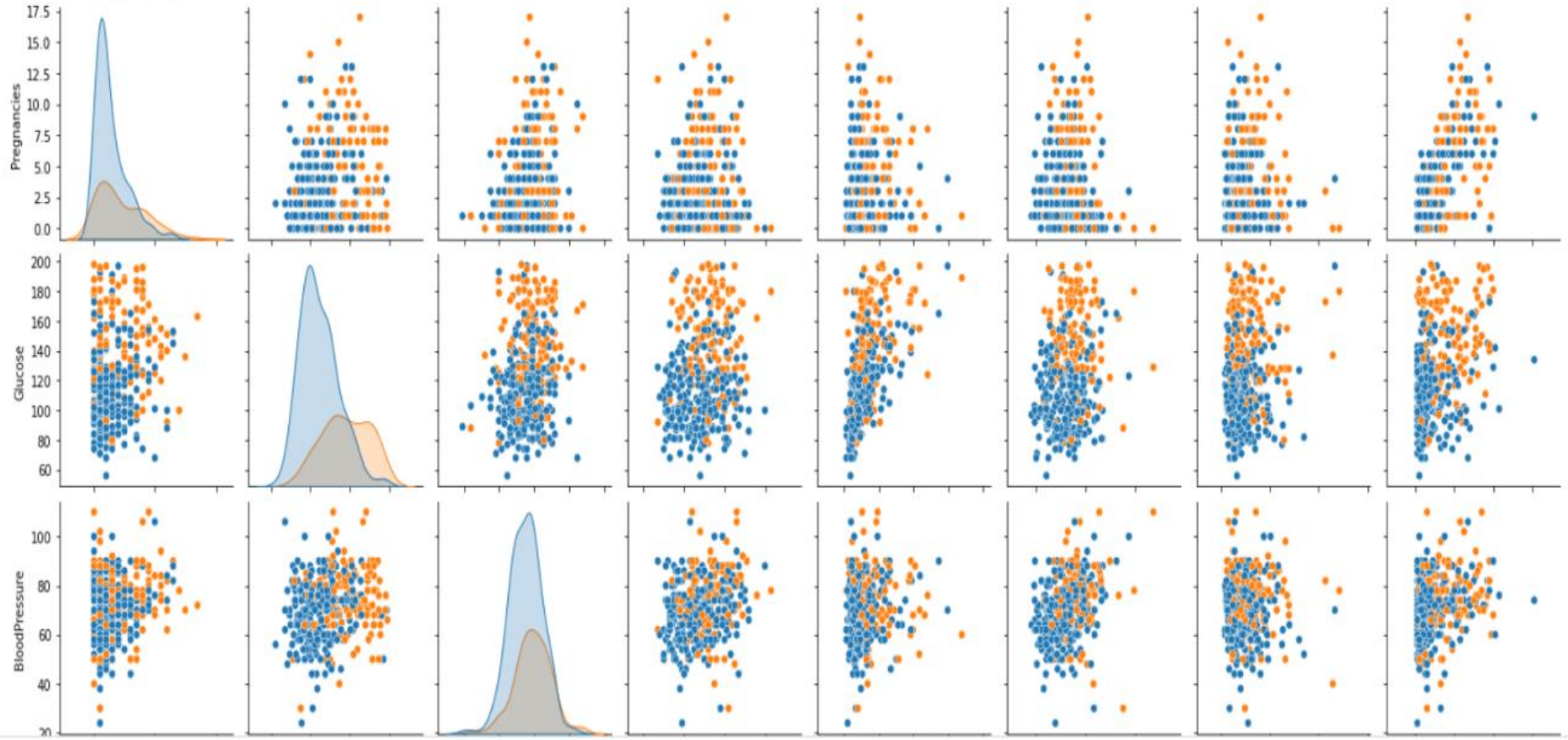
```
<seaborn.axisgrid.FacetGrid at 0x257c40bd970>
```



Distribution of Outcome variable

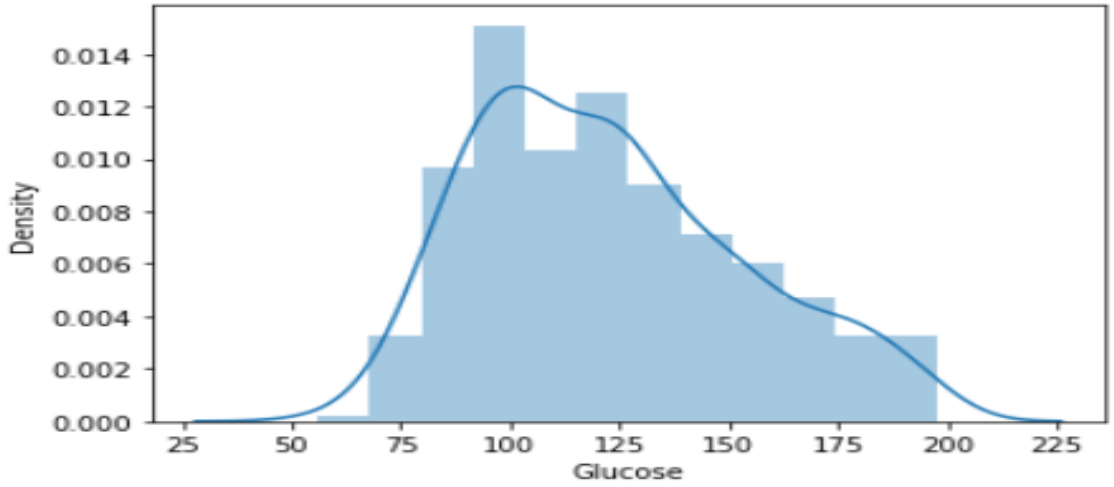
```
sns.pairplot(df,hue='Outcome')
```

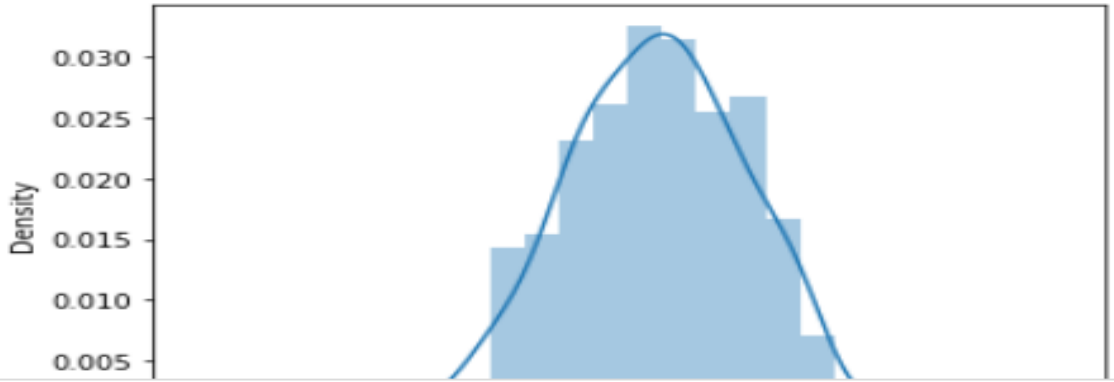
```
<seaborn.axisgrid.PairGrid at 0x257c40cfac0>
```

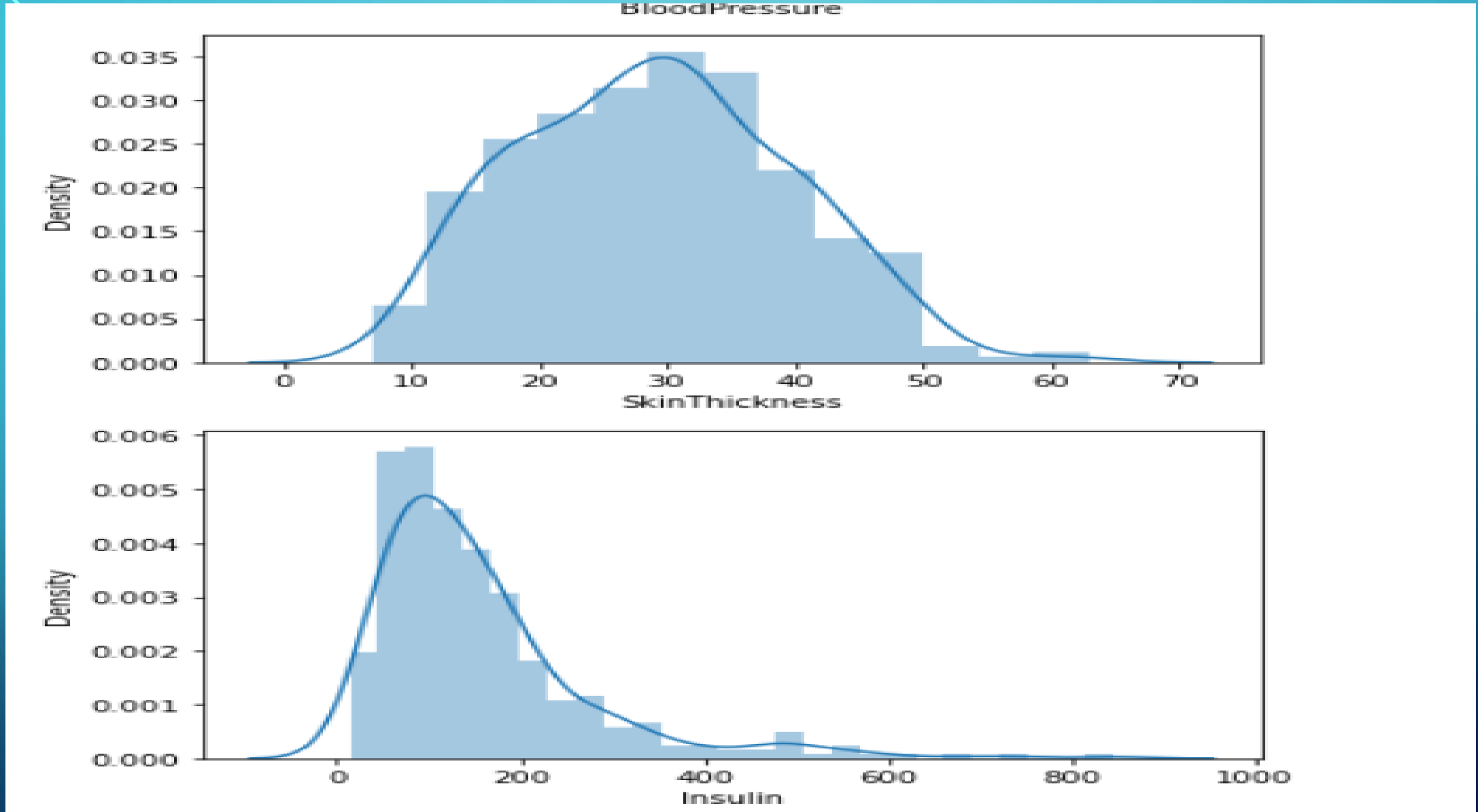


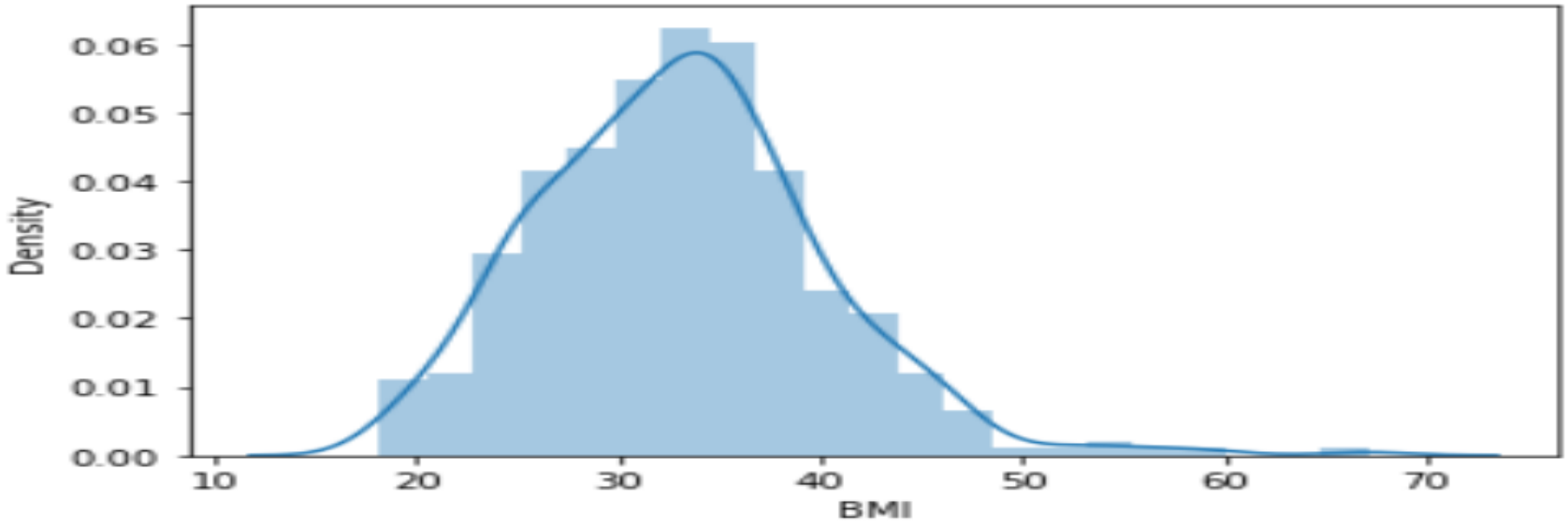
skewness of all variable

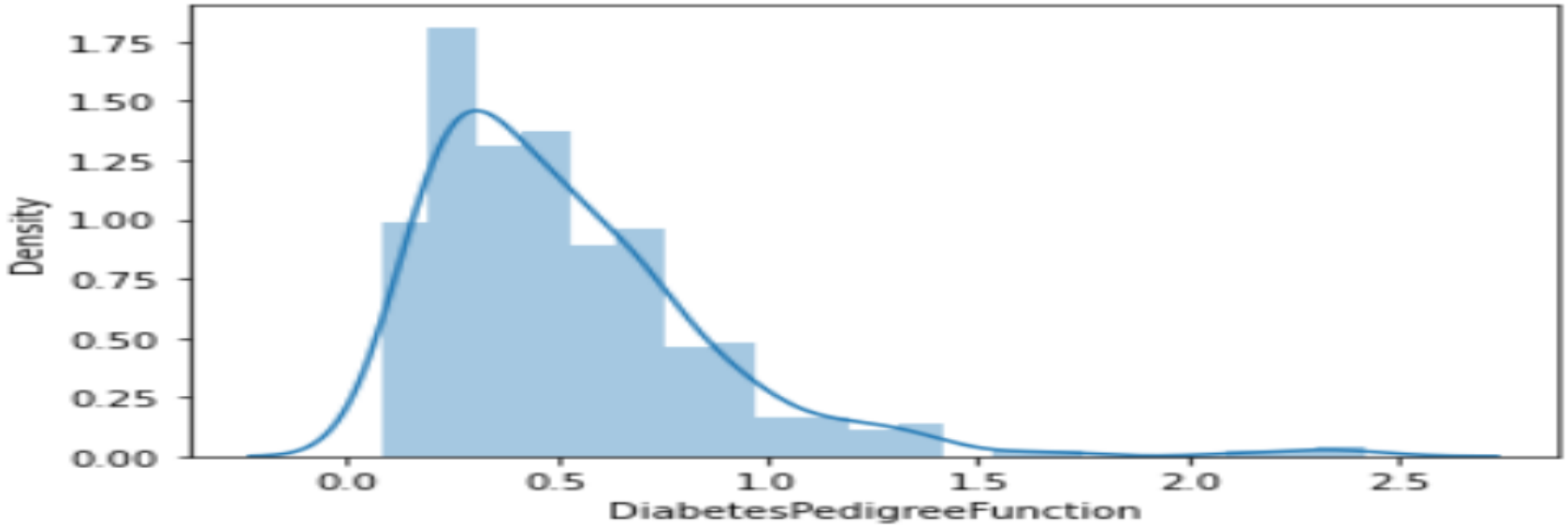
```
[ ]  for i in df.columns[1:-1]:
         plt.subplots()
         sns.distplot(df[i])
         print('\n')
         print(f"the skewness of {i} is {df[i].skew()}")
         print('\n')
```

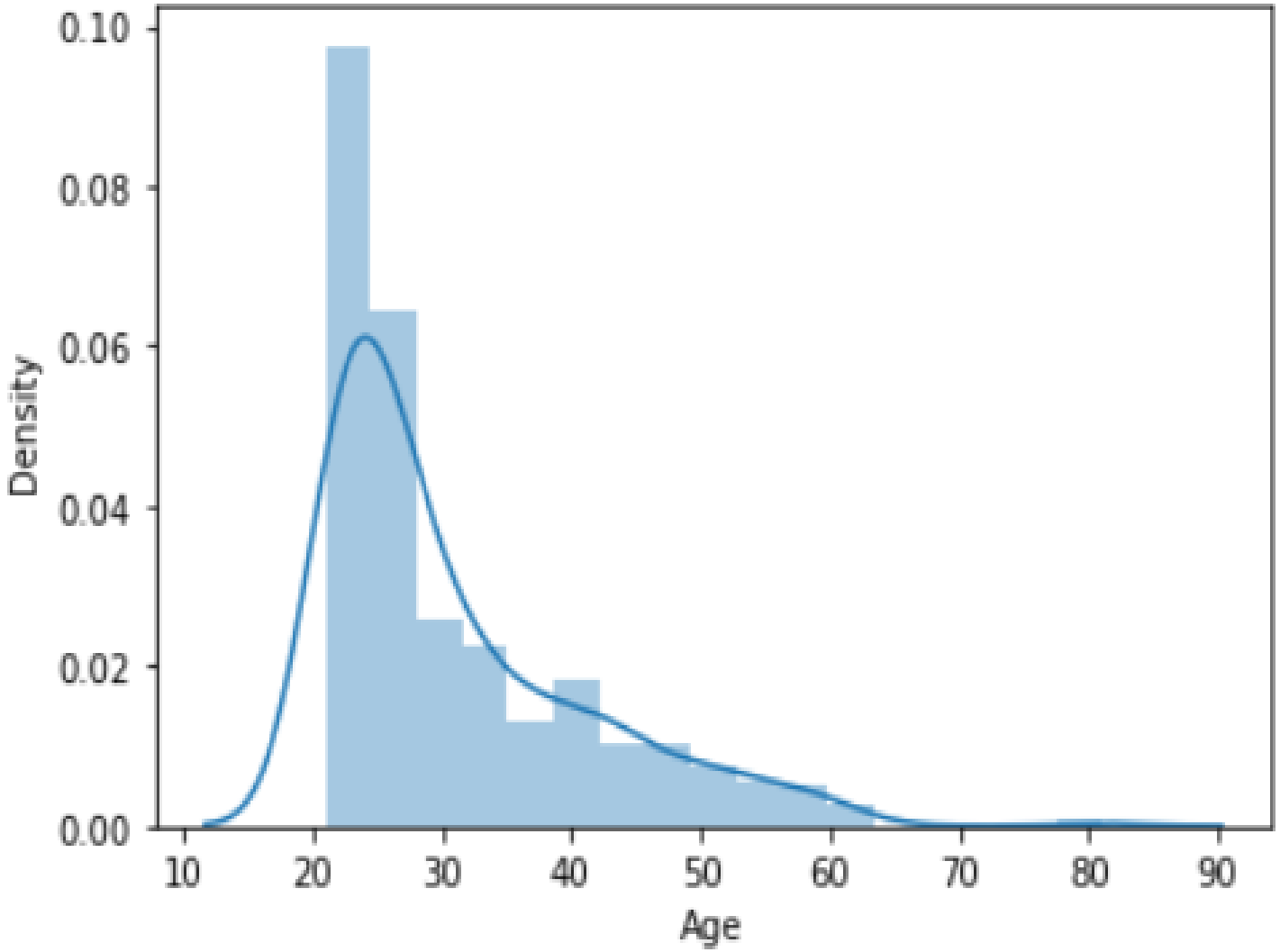





BloodPressure
Density
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
10
20
30
40
50
60
70
SkinThickness
Density
0.006
0.005
0.004
0.003
0.002
0.001
0.000
0
200
400
600
800
1000
Insulin


0.06
0.05
0.04
0.03
0.02
0.01
0.00
Density
10
20
30
40
50
60
70
BMI


1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Density
0.0
0.5
1.0
1.5
2.0
2.5
DiabetesPedigreeFunction

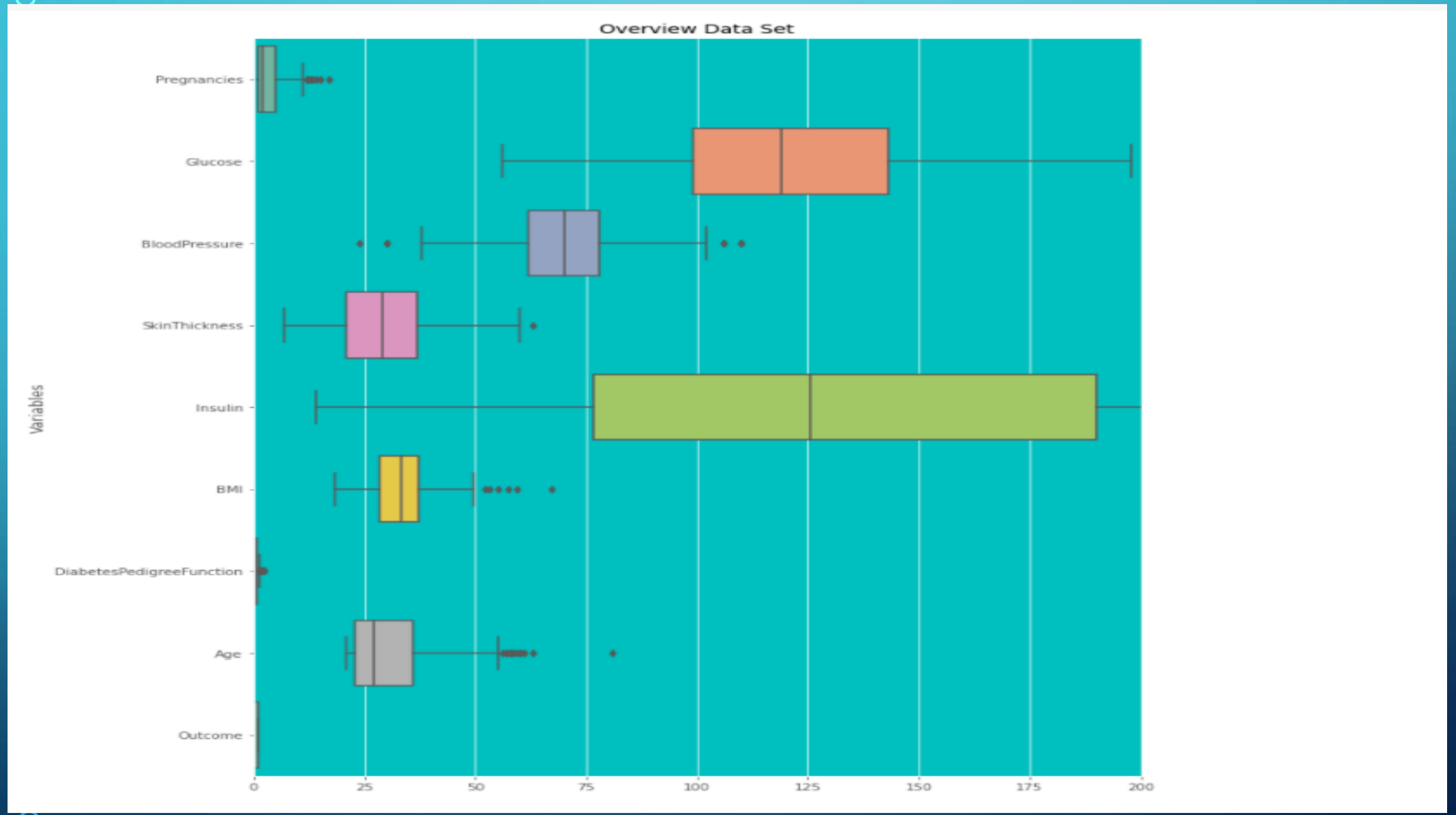

0.10
0.08
0.06
0.04
0.02
0.00
Density
10
20
30
40
50
60
70
80
90
Age

examine the outliers of these variables.

```
[ ]  plt.style.use('ggplot')

     f, ax = plt.subplots(figsize=(11, 15))

     ax.set_facecolor('c')
     ax.set(xlim=(-.05, 200))
     plt.ylabel('Variables')
     plt.title("Overview Data Set")
     ax = sns.boxplot(data = df,
       orient = 'h',
       palette = 'Set2')
```


Overview Data Set
Pregnancies
Glucose
BloodPressure
SkinThickness
Insulin
BMI
DiabetesPedigreeFunction
Age
Outcome
Variables
0
25
50
75
100
125
150
175
200

CONCLUSION

Diabetes is a killer with no known curable treatments. However, its complications can be reduced

Through proper awareness and timely treatment. Three major complications are related to blindness, kidney damage and heart attack.