Diabetes Analysis Of Women In Pima Tribe

Context

Diabetes is one of the most frequent diseases worldwide and the number of diabetic patients are growing over the years. The main cause of diabetes remains unknown, yet scientists believe that both genetic factors and environmental lifestyle play a major role in diabetes.

A few years ago research was done on the Pima tribe in USA. In this tribe, it was found that the ladies are prone to diabetes very early. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients were females at least 21 years old of Pima Indian heritage.

Objective

Here, we are analyzing different aspects of Diabetes in the Pima Indians tribe by doing Exploratory Data Analysis.

Data Dictionary

The dataset has the following information:

- Pregnancies: Number of times pregnant
- Glucose: Plasma glucose concentration over 2 hours in an oral glucose tolerance test
- BloodPressure: Diastolic blood pressure (mm Hg)
- SkinThickness: Triceps skin fold thickness (mm)
- Insulin: 2-Hour serum insulin (mu U/ml)
- BMI: Body mass index (weight in kg/(height in m)^2)
- DiabetesPedigreeFunction: A function that scores the likelihood of diabetes based on family history.
- Age: Age in years
- Outcome: Class variable (0: a person is not diabetic or 1: a person is diabetic)

Importing the necessary libraries

In [1]:

```
import numpy as np #library used for working with arrays.

import pandas as pd #library used for data manipulation and analysis.

import seaborn as sns #library for visualizations.

import matplotlib.pyplot as plt #library for plots and visualizations

%matplotlib inline
```

Reading the dataset:

In [4]:

```
pima = pd.read_csv("diabetes.csv") #reads csv file
print(pima.head(10)) #tail() gives only last 5 rows. You must use tail(n) for t

print(50*"=")
print("Number of columns:", len(pima.columns)) #find number of columns
print(50*"=")
```

```
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin        BMI  \
0            6      148             72             35       79  33.600000
1            1       85             66             29       79  26.600000
2            8      183             64             20       79  23.300000
3            1       89             66             23       94  28.100000
4            0      137             40             35      168  43.100000
5            5      116             74             20       79  25.600000
6            3       78             50             32       88  31.000000
7           10      115             69             20       79  35.300000
8            2      197             70             45      543  30.500000
9            8      125             96             20       79  31.992578

   DiabetesPedigreeFunction  Age  Outcome
0                     0.627   50        1
1                     0.351   31        0
2                     0.672   32        1
3                     0.167   21        0
4                     2.288   33        1
5                     0.201   30        0
6                     0.248   26        1
7                     0.134   29        0
8                     0.158   53        1
9                     0.232   54        1
==================================================
Number of columns: 9
==================================================
```

In [5]:

```
pima.shape #dimension of the dataframe as (row, column)
```

Out[5]:

```
(768, 9)
```

In [6]:

```
pima.size #total number of elements in the dataset
```

Out[6]:

```
6912
```

Data types of all the variables in the data set

In [7]:

```
print(pima.dtypes) #all data types in the data set.
print(50*"=")
print(pima.info()) #another method to give a more detailed information
```

```
Pregnancies                   int64
Glucose                       int64
BloodPressure                 int64
SkinThickness                 int64
Insulin                       int64
BMI                         float64
DiabetesPedigreeFunction    float64
Age                           int64
Outcome                       int64
dtype: object
==================================================
<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
None
```

- Note that pima.info() gives a more comprehensive summary of the data types than pima.dtypes, and includes the number of non-null counts and the total number of each data type.

In [9]:

```
pima.isnull().values.any() #If there are missing values, output is True, else F
```

Out[9]:

```
False
```

Summary statistics of blood pressure:

In [10]:

```
pima.iloc[: , 0 : 8].describe() #to get the summary statistics of all but the l
```

Out[10]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Diabe
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	
mean	3.845052	121.675781	72.250000	26.447917	118.270833	32.450805	
std	3.369578	30.436252	12.117203	9.733872	93.243829	6.875374	
min	0.000000	44.000000	24.000000	7.000000	14.000000	18.200000	
25%	1.000000	99.750000	64.000000	20.000000	79.000000	27.500000	
50%	3.000000	117.000000	72.000000	23.000000	79.000000	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	

- Summary statistics gives the summary of all the variables on a data frame (count, mean, standard deviation, minimum, maximum, quartiles etc). - The summary of all the variables, except the last variable ('outcome') is shown above.
- The dataframe contains 768 data points for blood pressure (since there are no missing data).
- The mean bood pressure in the sample is 72.250000. The standard deviation is 12.117203. This means that 99.73% of all the data lies within the data points (60.132797, 84.367203).
- The minimum and maximum values of blood pressure in this sample are 24.000000 and 122.000000 respectively.
- Hence the range of this variable is 98.000000 (max - min).
- The median corresponds to 50 percentile, i.e., 50% of the data lies below this point. Hence, the median blood pressure for this sample set is 72.000000. 25% of the data lie below 64.000000. 75% of the data lie below 80.000000.

Interquartile range for all the variable:

In [29]:

```
Q1 = pima.quantile(0.25)
Q3 = pima.quantile(0.75)
IQR = Q3 - Q1
print(IQR)
```

```
Pregnancies                  5.0000
Glucose                     40.5000
BloodPressure               16.0000
SkinThickness               12.0000
Insulin                     48.2500
BMI                          9.1000
DiabetesPedigreeFunction     0.3825
Age                         17.0000
Outcome                      1.0000
dtype: float64
```

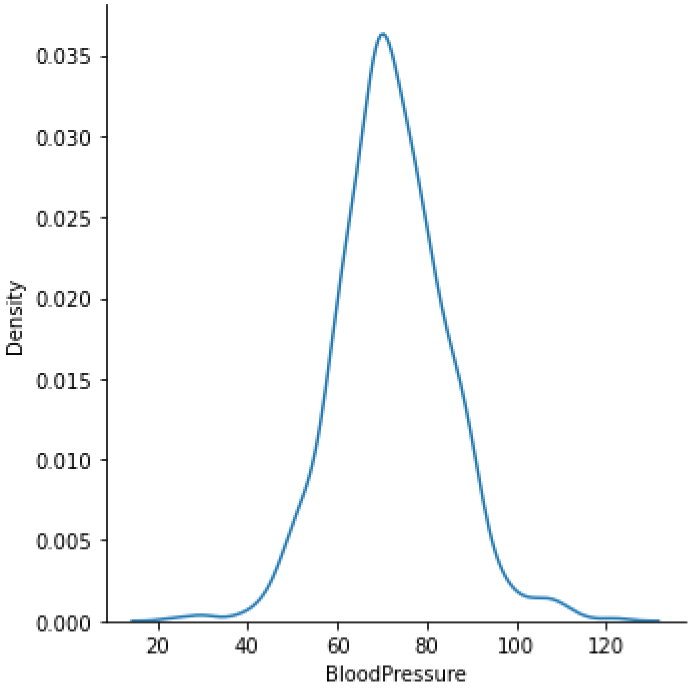
- Interquartile range (IQR) is where the middle 50% data of the variable lie.

- Boxplots use IQR to find out where most of the data lie, and also the outliers.
- Q1 - 1.5*IQR and Q3 + 1.5*IQR are used to determine the lower and upper bounds of the data respectively.
- Data points outside this range are outliers. Whether or not the outliers are important for a given study will depend on the nature of the data and the research question.

Distribution plot for blood pressure:

In [12]:
```
sns.displot(pima['BloodPressure'], kind = 'kde') #kernel distribution estimaito

plt.show()
```



- This distribution plot of the blood pressure uses kernel distribution estimation (kde) to smooth over the histogram of blood pressure.
- kde is a non-parametric estimator of density, and makes it easy to visualize the distribution of the data and reduces the variance in the plot.
- Visualization using histograms on the other hand are heavily affected by the bin size we choose.
- The y-axis is normalized, and hence the total area under the curve is 1.
- The graph peaks around a blood pressure of 70 mm Hg which means that most of the occurances are around that value.
- The distribution tapers off at both ends approximately symmetrically. - Hence the mean and the median should be fairly close in their values.
- In other words, the mean valie divides the data is two approximately equal halves in this sample set.

Analyzing BMI:

In [13]:

```
x = pima[pima['Glucose'] == pima['Glucose'].max()]['BMI'] # find BMI when gluco
x
```

Out[13]:

```
661    42.9
Name: BMI, dtype: float64
```

- The BMI corresponding to the maximum glucose level (199) is 42.9. It occurs in row number 661.
- Note that the row number starts from 0. Hence, it is actually 662nd row.

In [21]:

```
m1 = pima['BMI'].mean()  # mean
print(f"The mean of BMI is {m1:.2f}.")

m2 = pima['BMI'].median()  # median
print(f"The median of BMI is {m2:.2f}.")

m3 = pima['BMI'].mode()[0]  # mode
print(f"The mode of BMI is {m3}.")
```

```
The mean of BMI is 32.45.
The median of BMI is 32.00.
The mode of BMI is 32.0.
```

In [22]:

```
x2 = pima[pima['Glucose'] > pima['Glucose'].mean()].shape[0] #number of rows fo
print(f"Glucose level of {x2} women are above the mean level of glucose in wome
```

```
Glucose level of 343 women are above the mean level of glucose in women above
the age of 21 in Pima tribe.
```

In [23]:

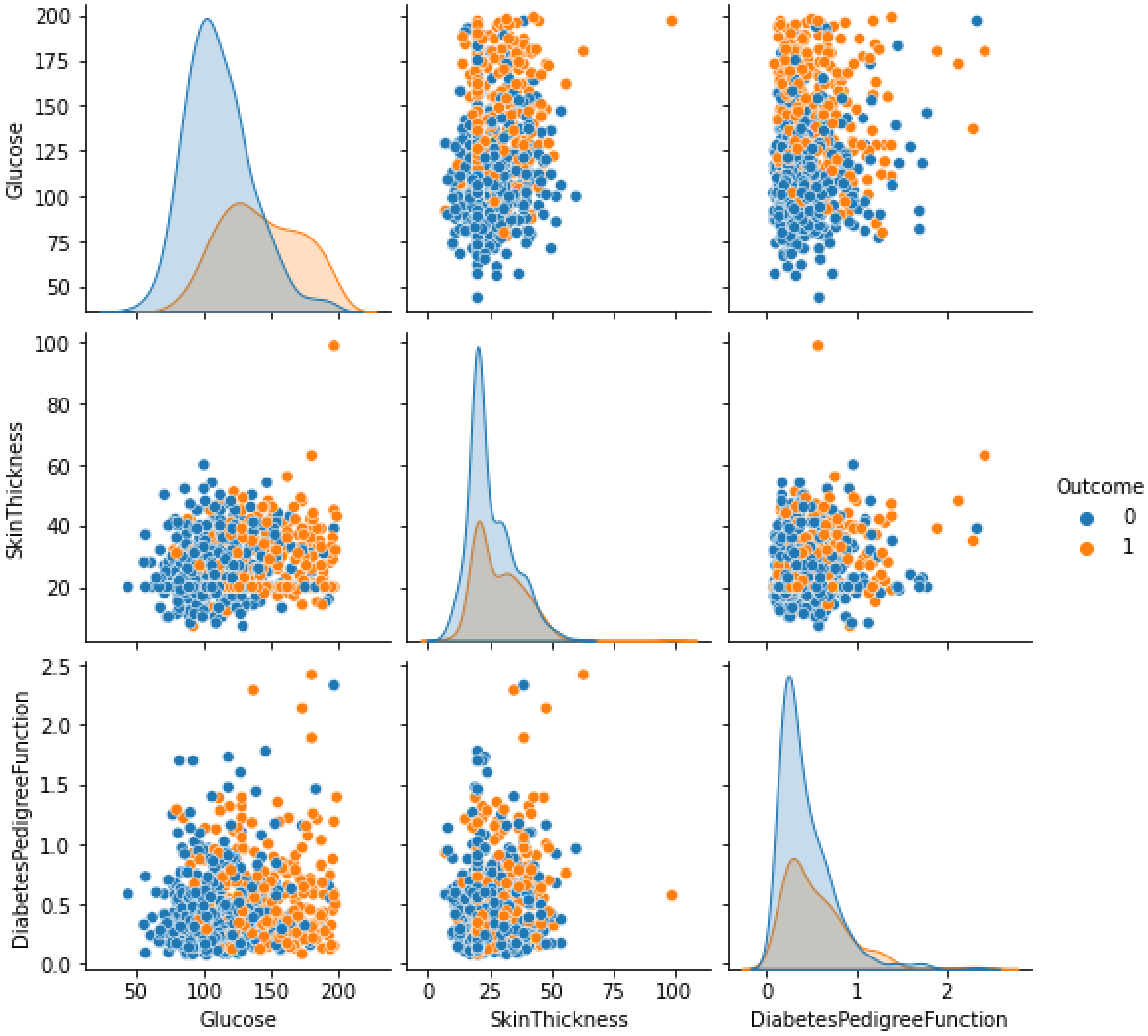
```
x3 = pima[(pima['BloodPressure'] == pima['BloodPressure'].median()) & (pima['BM
print(f"{x3} women have their BP equal to the median of BP and their BMI less t
```

```
22 women have their BP equal to the median of BP and their BMI less than the m
edian of BMI in women above the age of 21 in Pima tribe.
```

Pairplot for the variables 'Glucose', 'SkinThickness', and 'DiabetesPedigreeFunction':

In [24]:

```
sns.pairplot(data = pima, vars = ['Glucose', 'SkinThickness', 'DiabetesPedigree
plt.show()
```



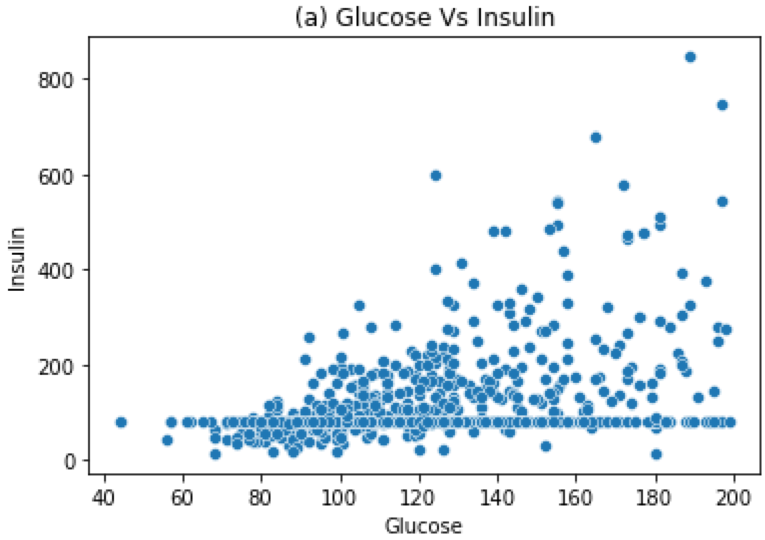
- The pairplot shows the relationships between Glucose level, skin thickness and diabetes pedigree function in the sample set. The graph shows the data for the nondiabetic and diabetic patients in blue (0) and orange (1) respectively.
- The distribution plot of glucose for diabetic vs non-diabetic women show that the glucose level for the diabetic patients is higher (orange distribution plot is shifted to higher value of gluciose level than blue).
- However, whether this difference is significant or not, needs to be tested. The distribution for skin thickness and diabetespredigreefunction (family history) shows no significant difference in diabetic and non diabetic patients.
- The scatter plots of skin thickness vs glucose, diabetespedigreefunction vs glucose and skin thickness vs diabetespedigreefunction, do not show any correlation between these pairs.
- However, diabetic patients appear to have a higher glucose level in average for the same diabetespedigreefunction and skin thickness as non-dabetic patients, as evident from the slightly rightward shifted (higher glucose level) orange clusters in the scatterplots.
- Such a decoupling of the diabetic and non-diabetic patients is not clear in the distribution graph of skin thickness Vs the iabetespedigreefunction.

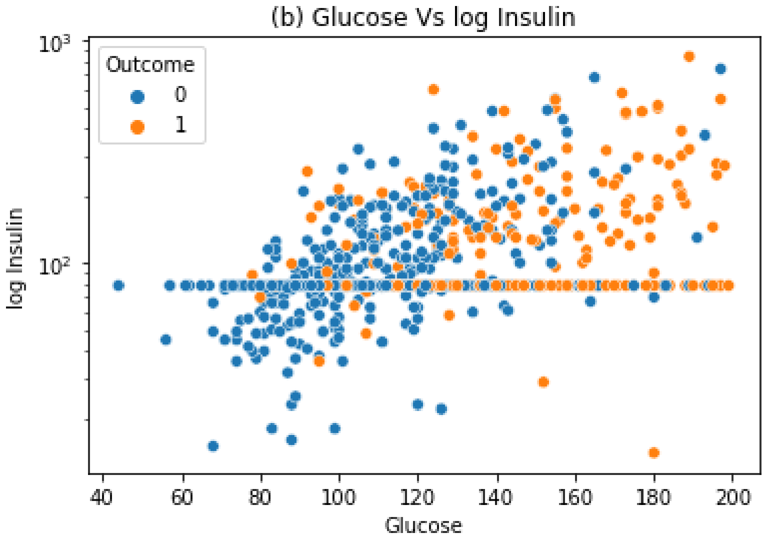
Relationship between glucose and insulin level:

```
In [25]: xplot = sns.scatterplot(x = 'Glucose', y = 'Insulin', data = pima).set(title='(
         plt.show()


         xplot = sns.scatterplot(x = 'Glucose', y = 'Insulin', data = pima, hue = 'Outco
         xplot.set(yscale="log") #set y-axis to log scale
         plt.ylabel('log Insulin') #set y label
         xplot.set_title('(b) Glucose Vs log Insulin') #set plot title

         plt.show()
```





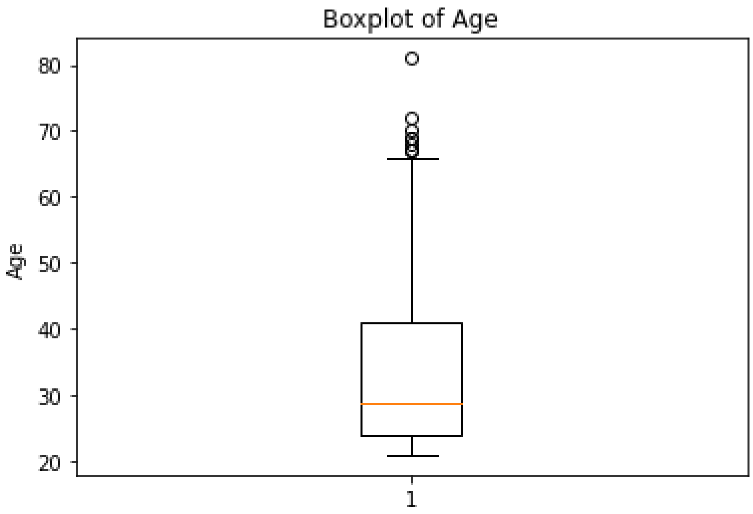
- The scatterplot of glucose level vs insulin have been plotted in two different ways. Plot (a) shows that a large number of data lies along a staight line. However there is a large scatter than increases with the glucose level. A large chunk of data seems to be clumped at glucose level < 122. In order to be able to see the trend in an expanded scale at the lower values of insulin levels (<300), we have recreated the plot (see plot (b)). Insulin, which is in y-axis is shown as a logarithm. This let's us see the trend for the

large amount of data clustered at < 300 of insulin level. We have also decouples the diabeltic (in orange) and non-diabetic (in blue) patients.
- It seems that most of the non-diabetic patients cluster below a glucose level of ~120 (note that average glucose level of the sample is 121.675781). Whereas, the diabetic patients tend to have higher than average glucose level. This is expected.
- There seems to be a large upward scatter in the insulin level for higher glucose levels in the patients. This could be due to insulin resistance. Insulin resistance is when the cells in liver, fat, uscles etc. do not respond well to insulin. Hence they cannot effectively use glucose from the blood for energy. To balance this, the pancreas makes more insulin, causing the blood sugar levels to go up. (Reference: https://www.webmd.com/diabetes/insulin-resistance-syndrome).
- There are also a significant number of data points along a straight line, which shows an almost constant insulin level even as glucose level increases. It appears that the diabetic patients whose data points lie along the straight line, may have higher glucose level not due to insulin resistance (Reference: https://academic.oup.com/jcem/article/85/6/2113/2850735).

Finding outliers in the age of the subjects:

In [26]:
```
plt.boxplot(pima['Age'])

plt.title('Boxplot of Age')
plt.ylabel('Age')
plt.show()
```

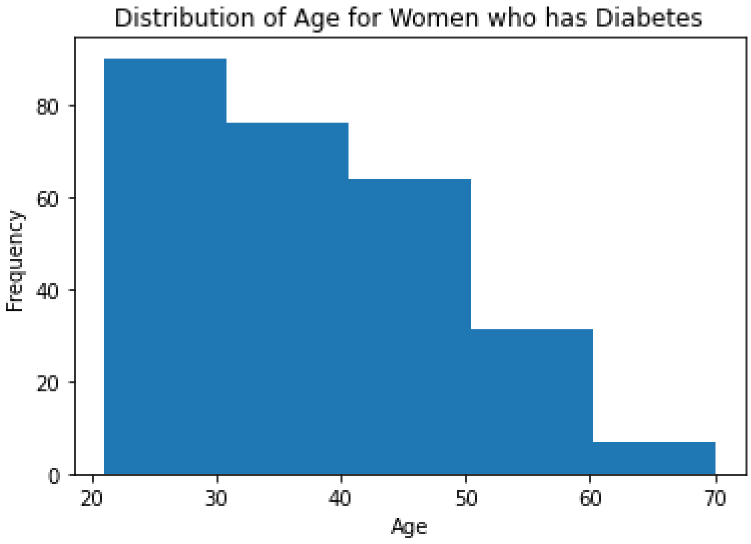


- The boxplot appears to be right skewed. It has a median at 29 years and a range of (21, 81).
- The outliers are shown as circles and lie above approximately greater than 65 years.
- These are the points outside the whiskers which range from (Q1 - 1.5*IQR) on the lower end, to (Q3 + 1.5*IQR) on the upper end of the boxplot above.

How do the number of diabetic/non-diabetic women in different age groups compare?

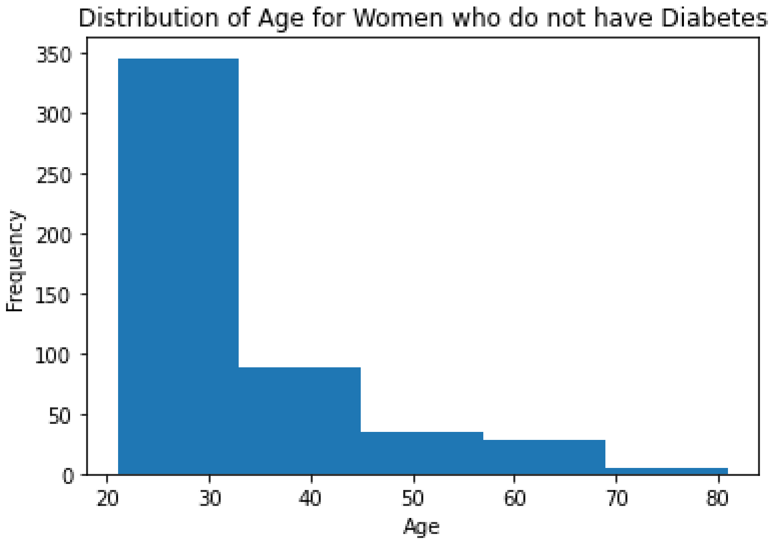
In [27]:

```
plt.hist(pima[pima['Outcome'] == 1]['Age'], bins = 5) #histogram for diabetic p
plt.title('Distribution of Age for Women who has Diabetes')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```



In [28]:

```
plt.hist(pima[pima['Outcome'] == 0]['Age'], bins = 5) #histogram for non-diabet
plt.title('Distribution of Age for Women who do not have Diabetes')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```



- For the non-diabetic patients, most of the patients are in the lower age group (<30) and their number tapers off sharply as the age increases (>30).
- For the diabetic patients, the number of diabetic patients are also higher for younger age groups as compared to the higher age group.

- However when we compare the histograms of diabetic vs non-diabetic patients, we see that the frequency of diabetic patients is significant in the higher age group (> 30) as well.
- For diabetic patients, the number of patients decrease much more slowly with age.

Correlation matrix of the observations:

In [30]:

```
corr_matrix = pima.iloc[ : ,0 : 8].corr()

corr_matrix
```

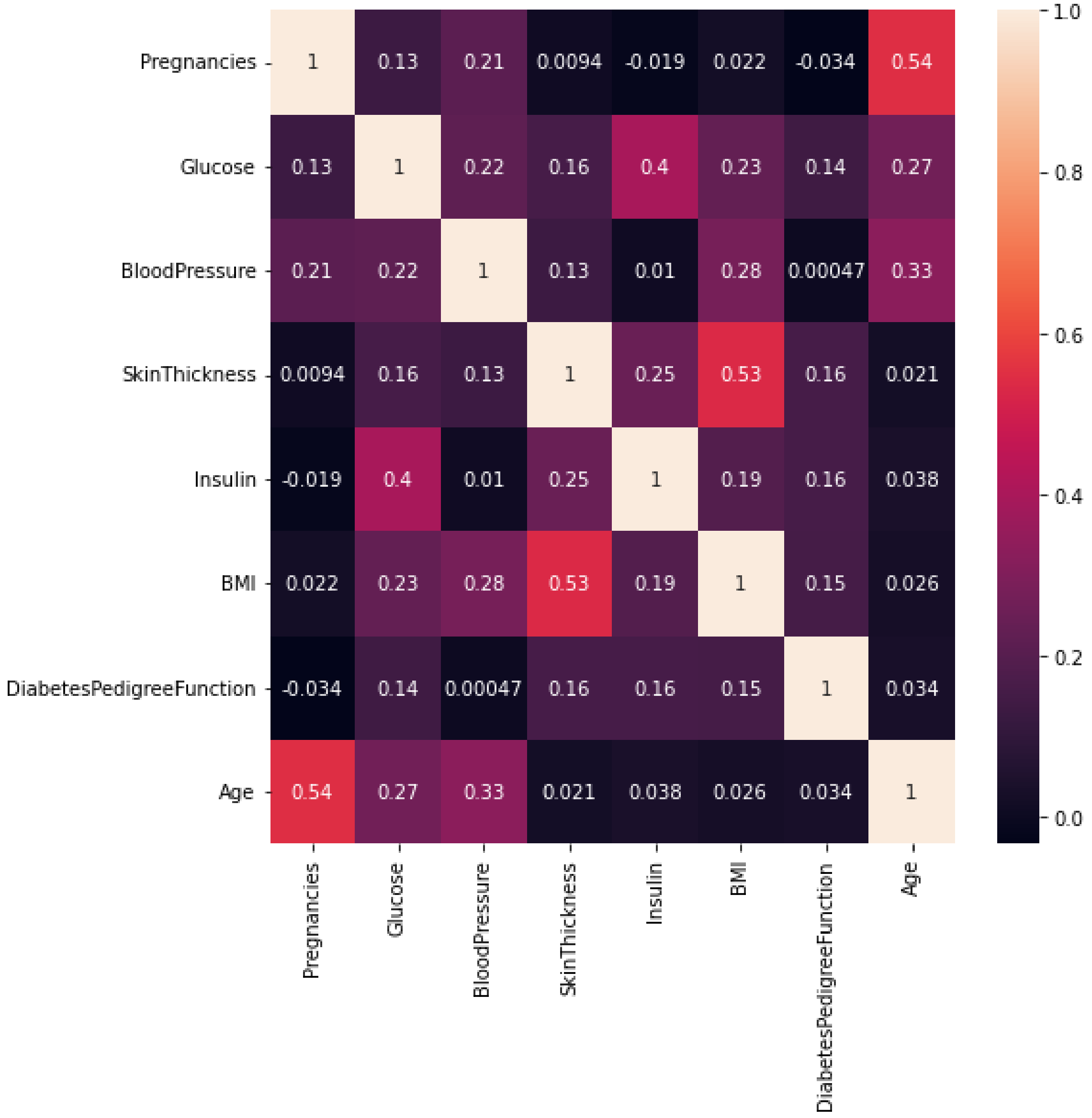
Out[30]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	
Pregnancies	1.000000	0.128022	0.208987	0.009393	-0.018780	0.0
Glucose	0.128022	1.000000	0.219765	0.158060	0.396137	0.2
BloodPressure	0.208987	0.219765	1.000000	0.130403	0.010492	0.2
SkinThickness	0.009393	0.158060	0.130403	1.000000	0.245410	0.5
Insulin	-0.018780	0.396137	0.010492	0.245410	1.000000	0.1
BMI	0.021546	0.231464	0.281222	0.532552	0.189919	1.0
DiabetesPedigreeFunction	-0.033523	0.137158	0.000471	0.157196	0.158243	0.1
Age	0.544341	0.266673	0.326791	0.020582	0.037676	0.0

In [31]:

```
plt.figure(figsize = (8, 8))
sns.heatmap(corr_matrix, annot = True)

# Display the plot
plt.show()
```



- There seems to be a moderate positive correlation between skin thickness and BMI (r = 0.53), moderate posiitve correlation between insulin and glucose (r = 0.4) and moderate pritive correlation between age and pregnancies (r = 0.54).
- Rest of the correlations are weak (0.2 < r < 0.39) to very week (r < 0.19).