Project Name: Healthcare PGP

In [2]:

```
import pandas as pd
# ^^^ pyforest auto-imports - don't write above this line
import numpy as np
import pandas as pd


import matplotlib.pyplot as plt
from matplotlib import style
import seaborn as sns

%matplotlib inline
```

In [1]:

```
df=pd.read_csv(r'C:\Users\Anuj Bhalla\Downloads\Project_2\Project 2\Healthcare - Dia
```

Project Task: Week 2

In [25]:

```
'''

Data Exploration:

1. Check the balance of the data by plotting the count of outcomes by their value.
Describe your findings and plan future course of action.

2. Create scatter charts between the pair of variables to understand the relationshi

3. Perform correlation analysis. Visually explore it using a heat map.

'''
```

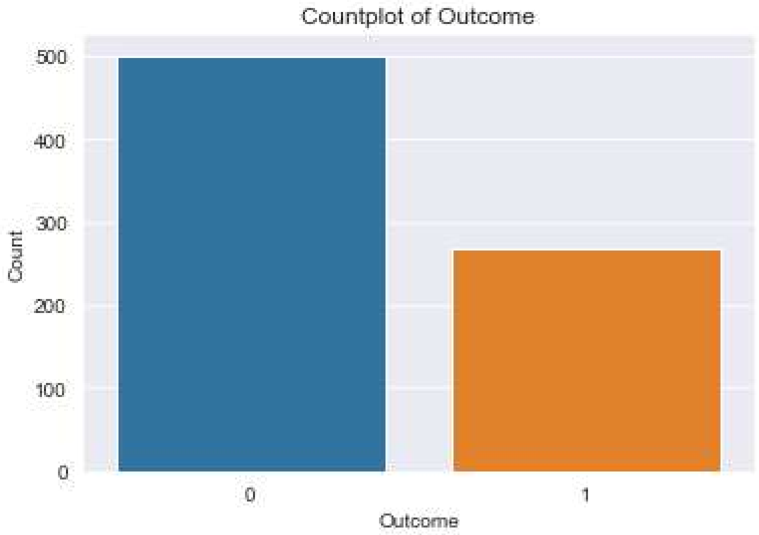
Out[25]:

```
'\n\nData Exploration:\n\n1. Check the balance of the data by plotting the count of
outcomes by their value. \nDescribe your findings and plan future course of actio
n.\n\n2. Create scatter charts between the pair of variables to understand the relat
ionships. Describe your findings.\n\n3. Perform correlation analysis. Visually explo
re it using a heat map.\n\n'
```

In [26]:

```
sns.set_style('darkgrid')
sns.countplot(df['Outcome'])
plt.title("Countplot of Outcome")
plt.xlabel('Outcome')
plt.ylabel("Count")
print("Count of class is:\n",df['Outcome'].value_counts())
```

```
C:\Users\Anuj Bhalla\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWa
rning: Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an exp
licit keyword will result in an error or misinterpretation.
  warnings.warn(
Count of class is:
 0    500
1    268
Name: Outcome, dtype: int64
```



We can see that both class is balanced so we need not to perform any sampling method to maintain the balance between both classes. Therefore I'm directly using this data in training and testing purpose without performing any sampling method. Meanwhile during Model Validation , we also need not worry about ROC Curve because data is not imbalanced, but as this is a health/medical data so I am planning to use ROC curve to make sure TYPE 2 ERROR is not there.

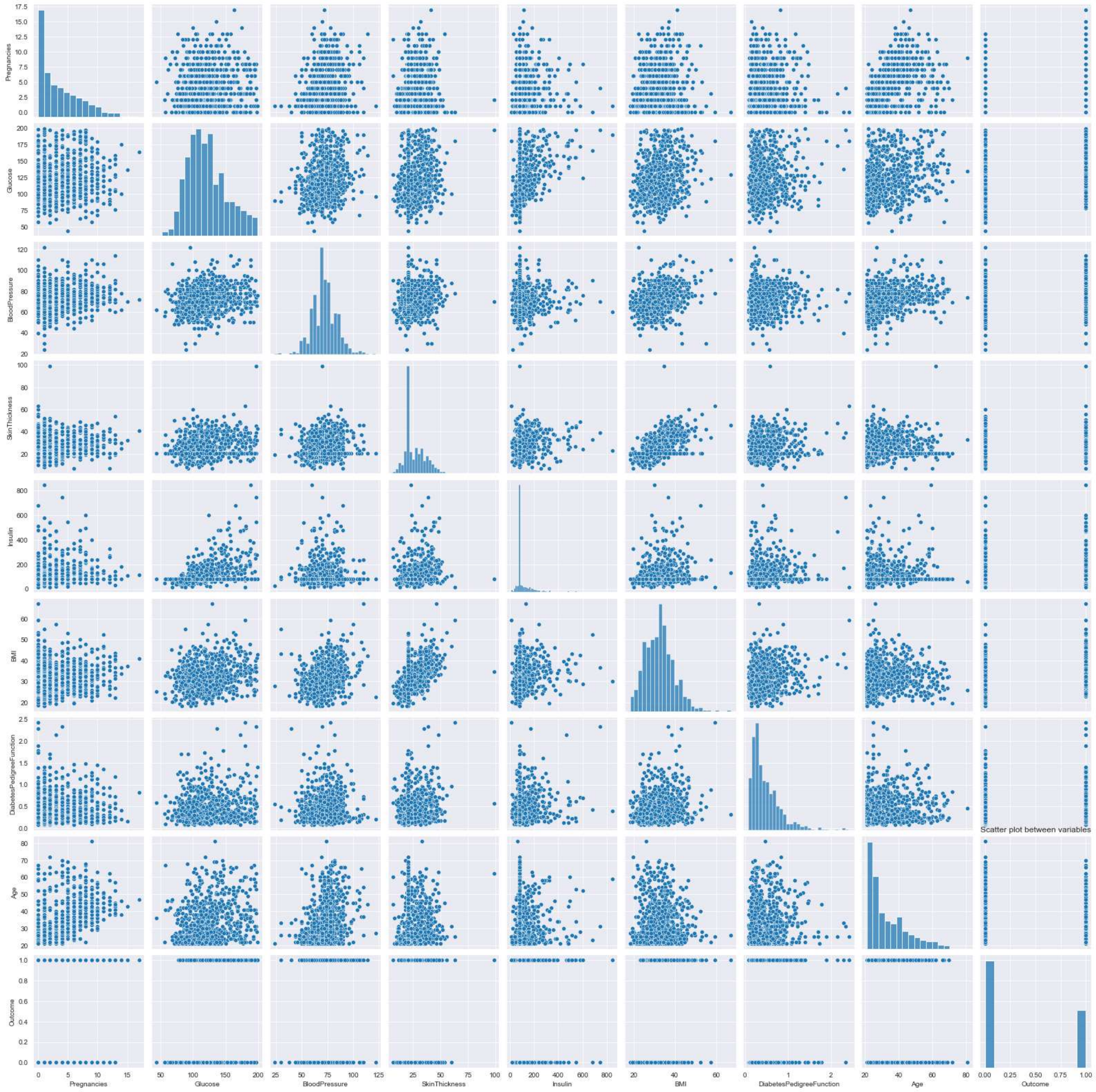
Create scatter charts between the pair of variables to understand the relationships. Describe your findings.

In [27]:

```
sns.pairplot(df)
plt.title('Scatter plot between variables')
```

Out[27]:

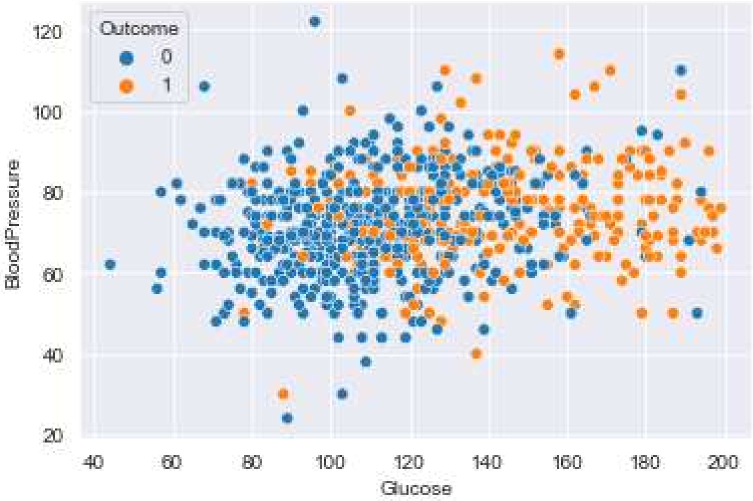
```
Text(0.5, 1.0, 'Scatter plot between variables')
```



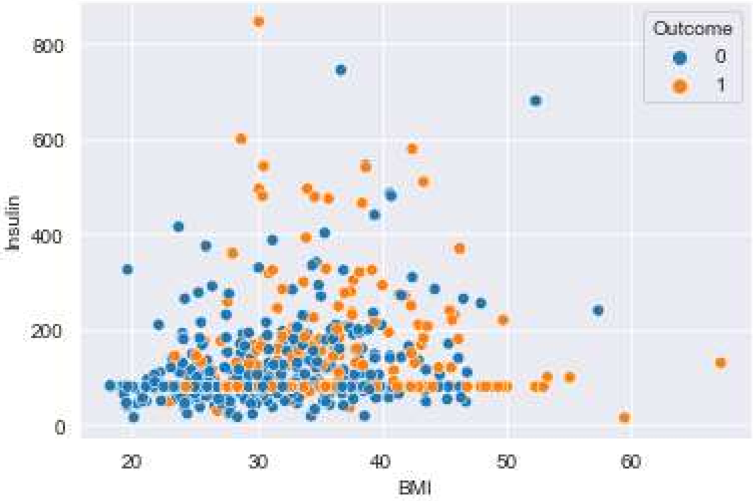
We can see from scatter plot that there is no strong multicolinearity among features, but between skin thickness and BMI, Pregnancies and age it looks like there is small chance of positive correlation..i will explore more when analyzing correlation

In [28]:
```
Glucose = df['Glucose']
BloodPressure=df['BloodPressure']
Outcome=df['Outcome']
BMI=df['BMI']
Insulin=df['Insulin']
SkinThickness=df['SkinThickness']
```

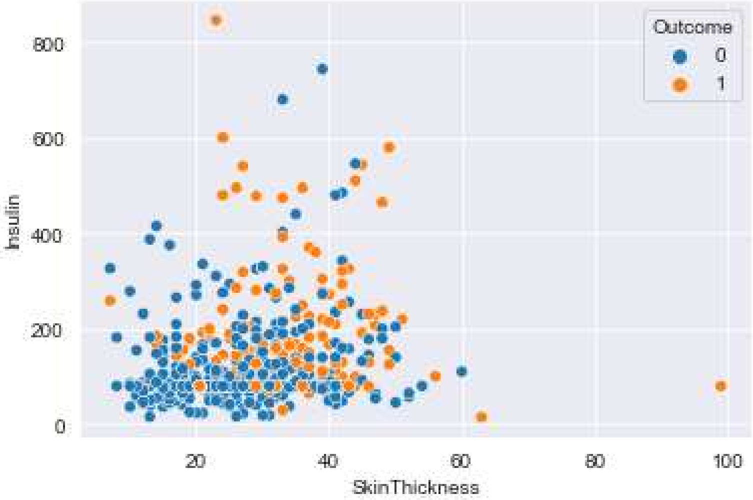
In [29]:
```
g = sns.scatterplot(x= "Glucose" ,y= "BloodPressure",
              hue="Outcome",
              data=df);
```



In [30]:
```
B =sns.scatterplot(x= "BMI" ,y= "Insulin",
                   hue="Outcome",
                   data=df);
```



In [31]:
```
S =sns.scatterplot(x= "SkinThickness" ,y= "Insulin",
                   hue="Outcome",
                   data=df);
```



from the scatter plot patterns between variables we can see that BMI and Blood Pressure is most important feature in prediction followed by Glucose and Age.

Perform correlation analysis. Visually explore it using a heat map.

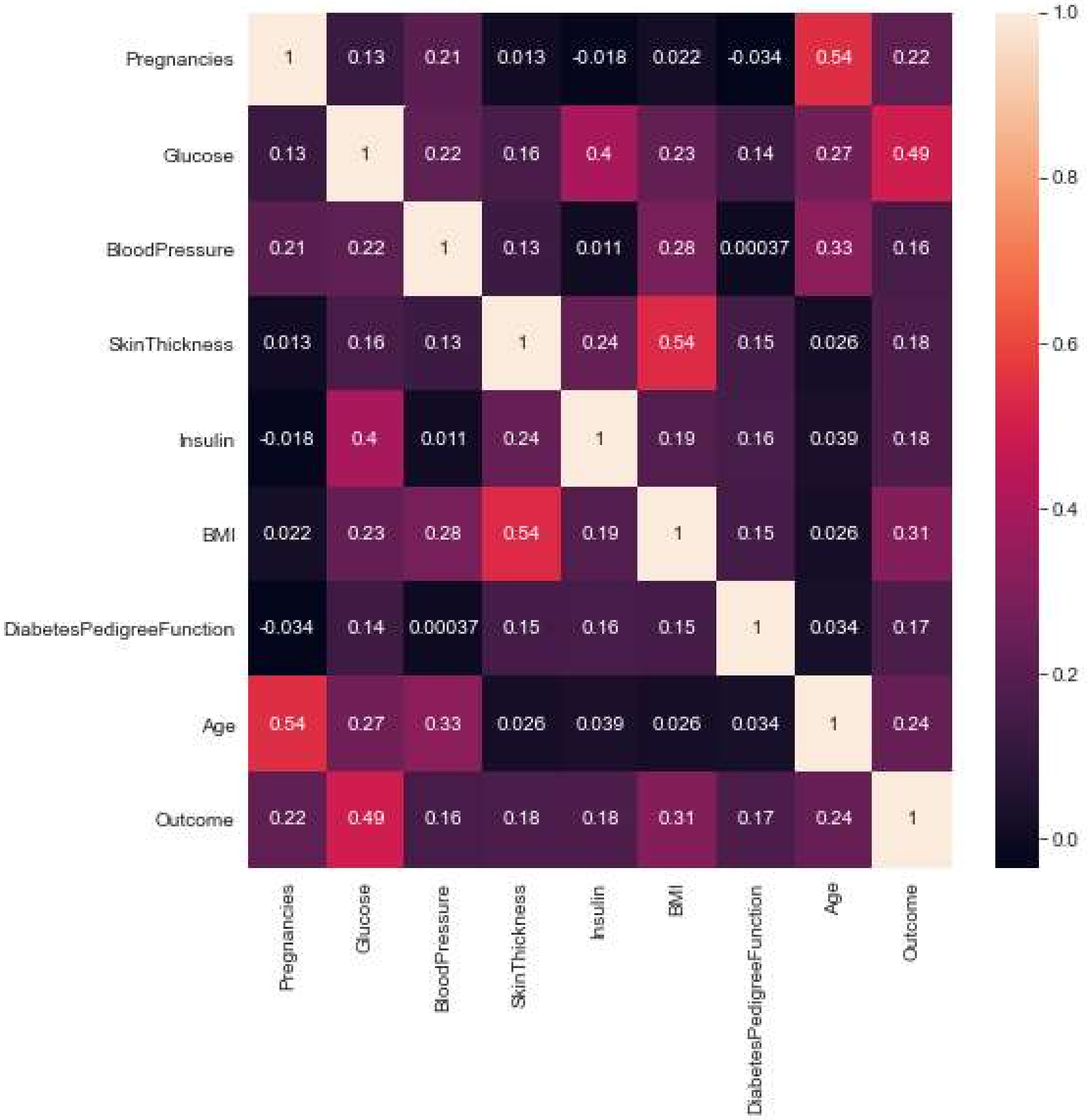
In [32]:
```
df.corr()
```

Out[32]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI
Pregnancies	1.000000	0.127964	0.208984	0.013376	-0.018082	0.021546
Glucose	0.127964	1.000000	0.219666	0.160766	0.396597	0.231478
BloodPressure	0.208984	0.219666	1.000000	0.134155	0.010926	0.281231
SkinThickness	0.013376	0.160766	0.134155	1.000000	0.240361	0.535703
Insulin	-0.018082	0.396597	0.010926	0.240361	1.000000	0.189856
BMI	0.021546	0.231478	0.281231	0.535703	0.189856	1.000000
DiabetesPedigreeFunction	-0.033523	0.137106	0.000371	0.154961	0.157806	0.153508
Age	0.544341	0.266600	0.326740	0.026423	0.038652	0.025748
Outcome	0.221898	0.492908	0.162986	0.175026	0.179185	0.312254

In [34]:
```
plt.subplots(figsize=(8,8))
sns.heatmap(df.corr(),annot=True)  ### gives correlation value
```

Out[34]:
```
<AxesSubplot:>
```



''' From the HeatMap, we can see that Majority of the correlations are "Positive", but weak. Strongest correlated pairs are

"BMI : Skin Thickness",

"Age : Pregnancies",

"Glucose : Outcome(Target Variable)",

"Insulin : Glucose"'''

```
In [ ]:
```