In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
df=pd.read_csv('after_week1.csv')
df.head()
```

In [2]:
```
df=pd.read_csv('after_week1.csv')
df.head()
```

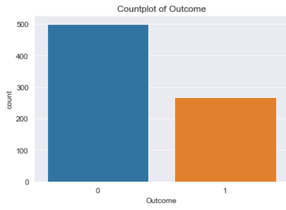
Out[2]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148.0	72.0	35.000000	79.799479	33.6	0.627	50	1
1	1	85.0	66.0	29.000000	79.799479	26.6	0.351	31	0
2	8	183.0	64.0	20.536458	79.799479	23.3	0.672	32	1
3	1	89.0	66.0	23.000000	94.000000	28.1	0.167	21	0
4	0	137.0	40.0	35.000000	168.000000	43.1	2.288	33	1

Countplot

In [8]:
```
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())
```

```
count of class is:
 0    500
1    268
Name: Outcome, dtype: int64
```

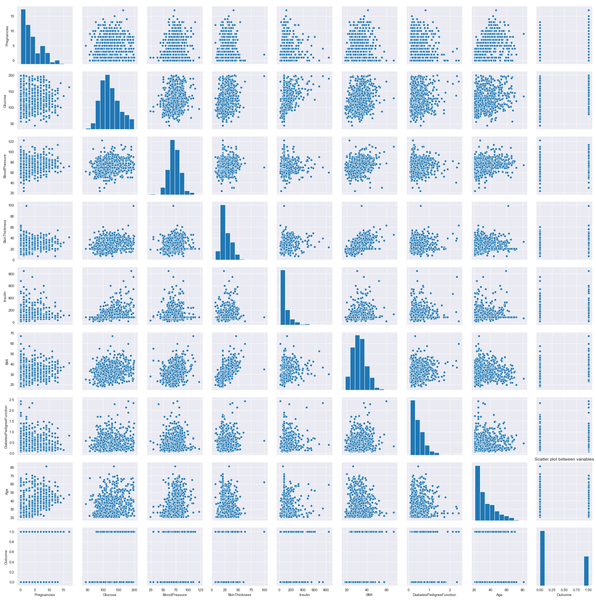


We can see that both class is balanced so we need not to perform any sampling method to maintain the balance between both classes. Therefor i will be directly using this data in training and testing purpose without performing any sampling method.

Scatter Plot

In [9]:
```
sns.pairplot(df)
plt.title('Scatter plot between variables')
```

Out[9]:
```
Text(0.5,1,'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

Correlation Analysis

In [10]:
```
df.corr()
```

Out[10]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunctio
Pregnancies	1.000000	0.127964	0.208984	0.013376	-0.018082	0.021546	-0.03352
Glucose	0.127964	1.000000	0.219666	0.160766	0.396597	0.231478	0.13710
BloodPressure	0.208984	0.219666	1.000000	0.134155	0.010926	0.281231	0.00037
SkinThickness	0.013376	0.160766	0.134155	1.000000	0.240361	0.535703	0.15496
Insulin	-0.018082	0.396597	0.010926	0.240361	1.000000	0.189856	0.15780
BMI	0.021546	0.231478	0.281231	0.535703	0.189856	1.000000	0.15350
DiabetesPedigreeFunction	-0.033523	0.137106	0.000371	0.154961	0.157806	0.153508	1.00000
Age	0.544341	0.266600	0.326740	0.026423	0.038652	0.025748	0.03356
Outcome	0.221898	0.492908	0.162986	0.175026	0.179185	0.312254	0.17384

In [13]:
```
plt.figure(dpi=80)
sns.heatmap(df.corr(),cmap= 'coolwarm')
```

Out[13]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1f87e4e29e8>
```