In [5]:

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import os
print(os.listdir("../workspace"))
```

```
['.ipynb_checkpoints', '.metadata', '.recommenders', 'Basic Sales analysi
s.ipynb', 'Basic Sales analysis.py', 'BlackFriday.csv', 'diabetes.csv', 'U
ntitled.ipynb', 'Untitled1.ipynb', 'Untitled2.ipynb']
```

In [14]:

```
data = pd.read_csv('../workspace/diabetes.csv')
data.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
```

In [15]:

```
data.describe()
```

Out[15]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Diat
count	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	
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	
50%	3.000000	117.000000	72.000000	23.000000	30.500000	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	

In [16]:

```
data.columns
```

Out[16]:

```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insuli
n',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

In [17]:

```
data.isnull().sum()
```

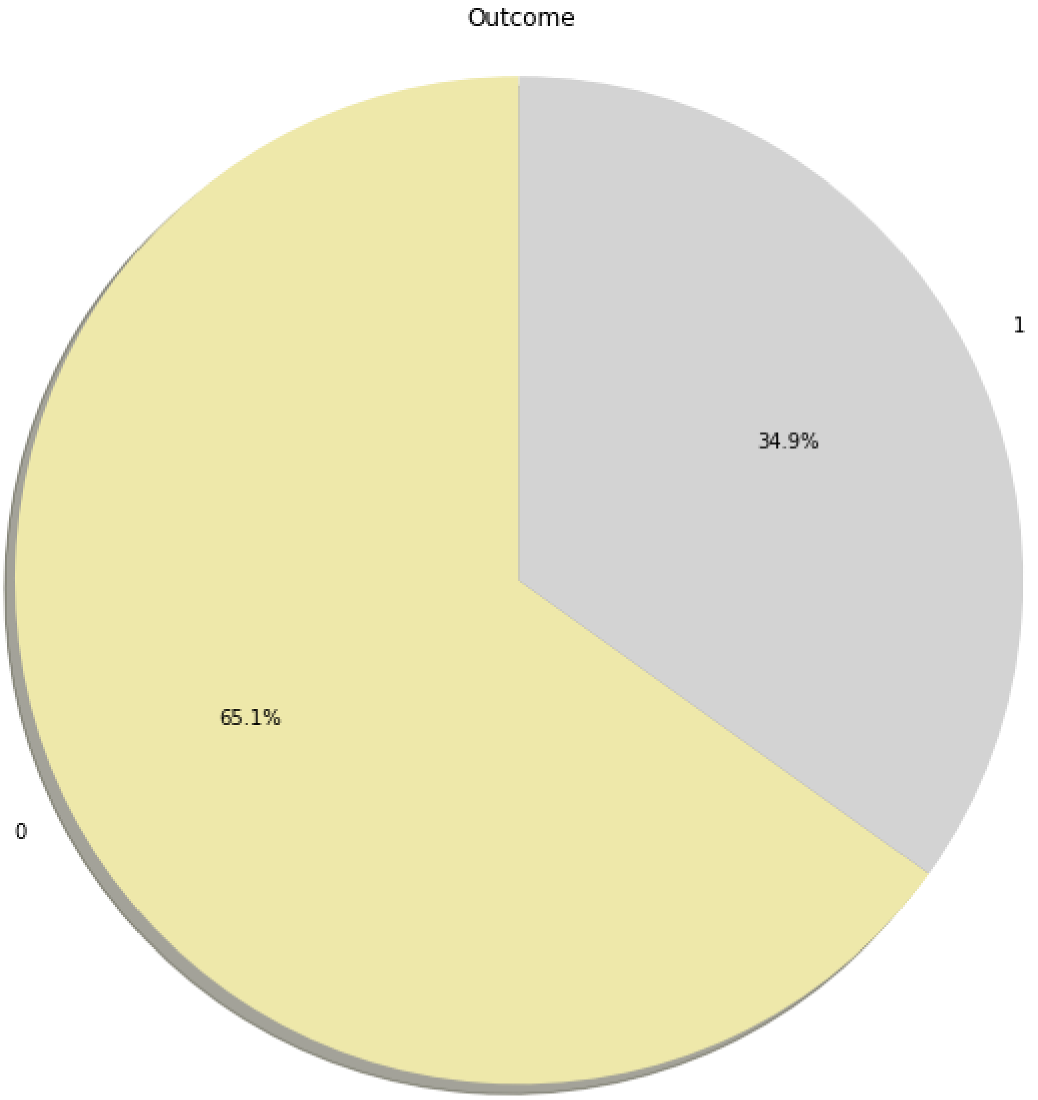
Out[17]:

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

In [18]:

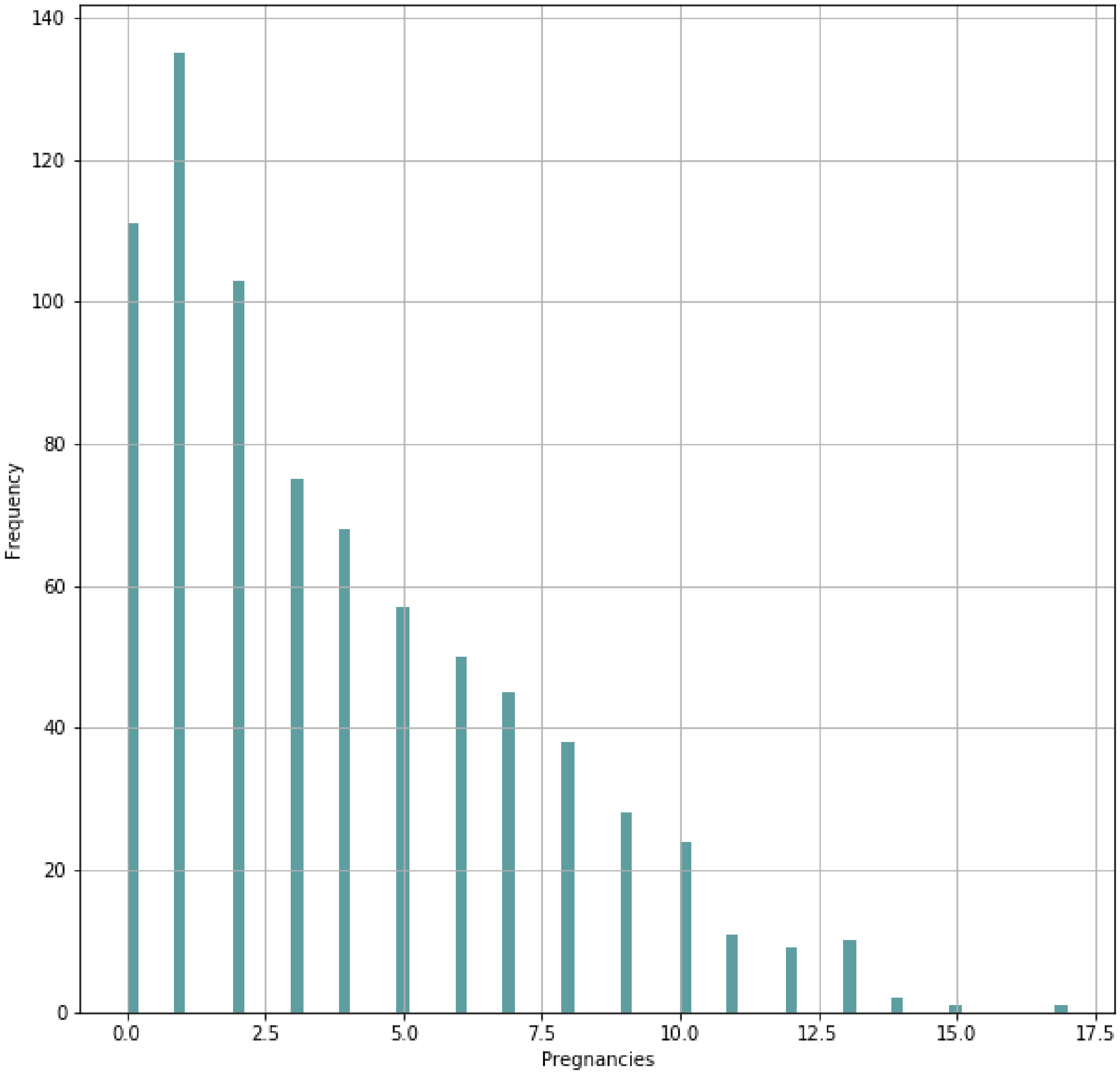
```
print(data.Outcome.value_counts())
labels = '0', '1',
sizes = [500, 268]
colors = ['palegoldenrod','lightgrey']
explode = (0, 0)
fig1, ax1 = plt.subplots(figsize =(10,10))
ax1.pie(sizes,colors = colors ,explode=explode, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax1.axis('equal')
plt.title("Outcome")
plt.show()
```

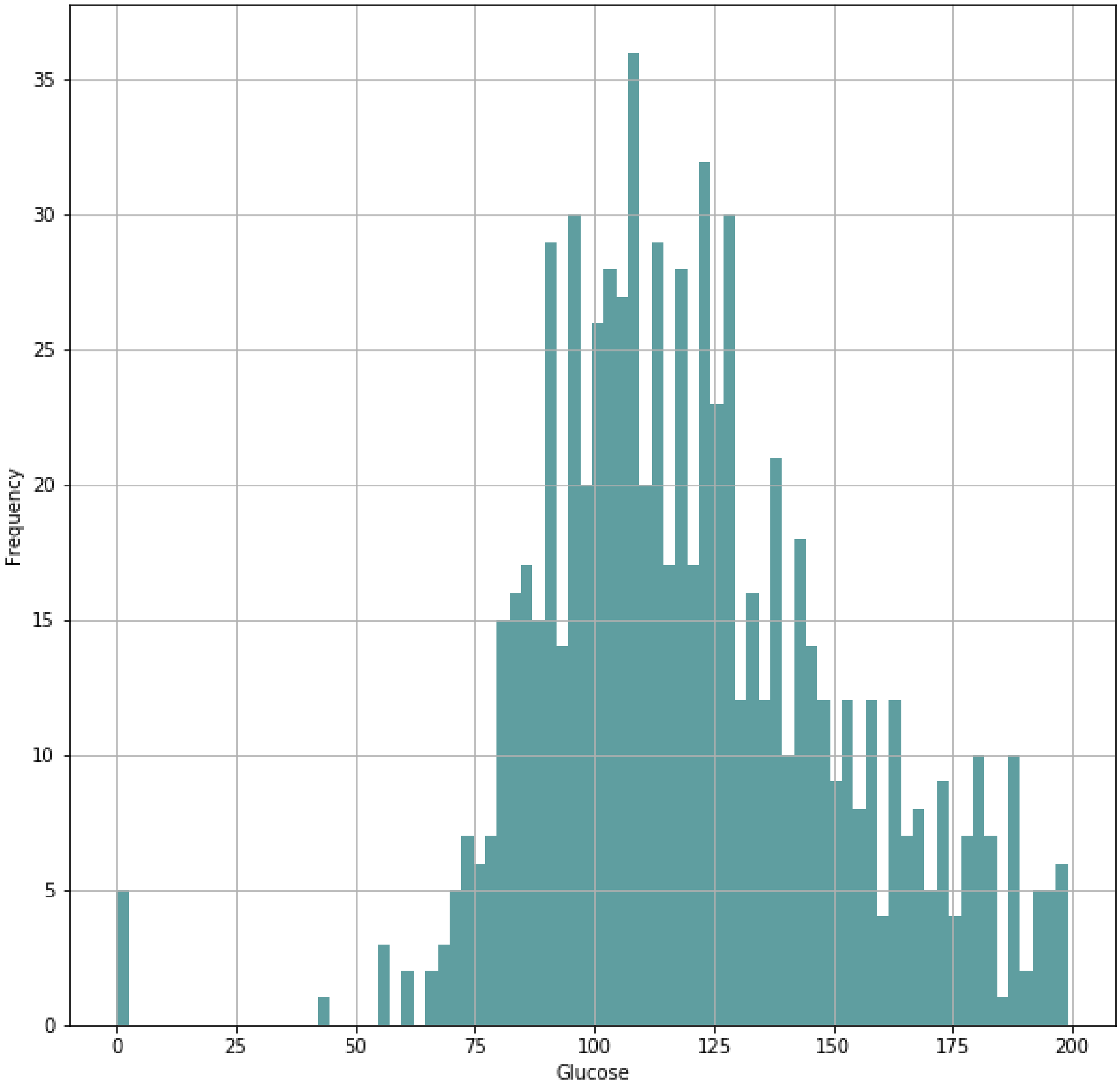
```
0    500
1    268
Name: Outcome, dtype: int64
```

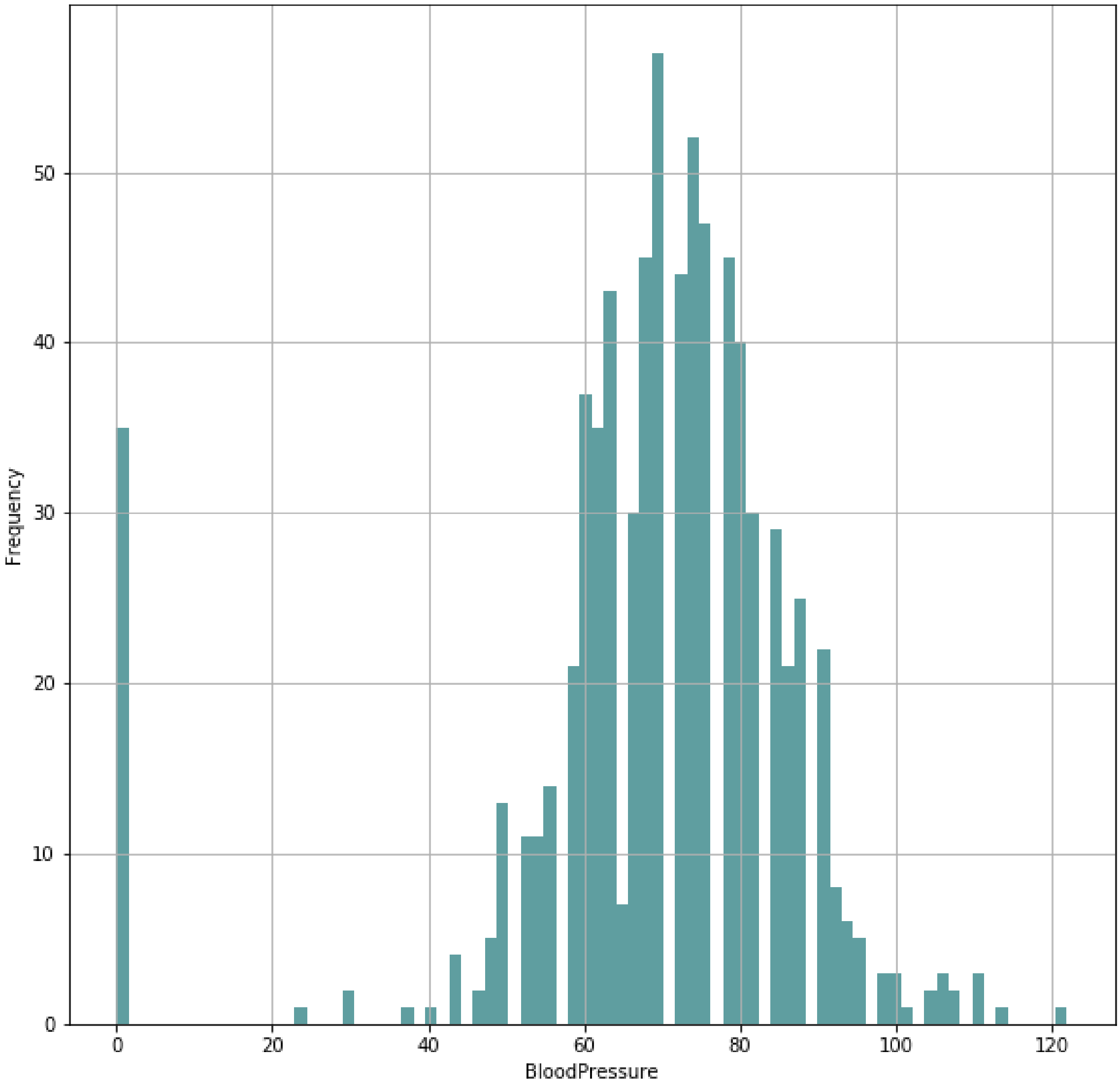


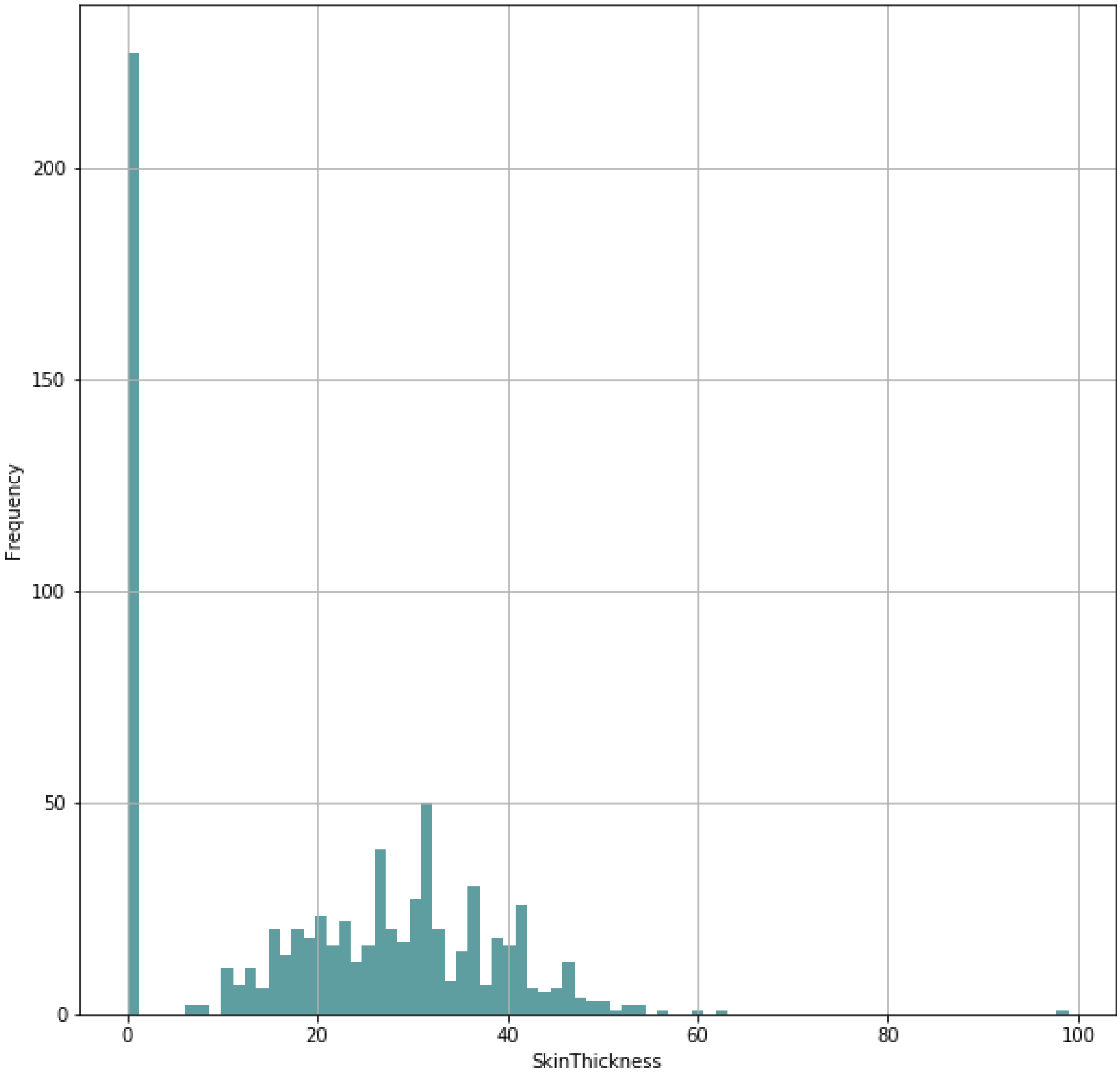
In [19]:

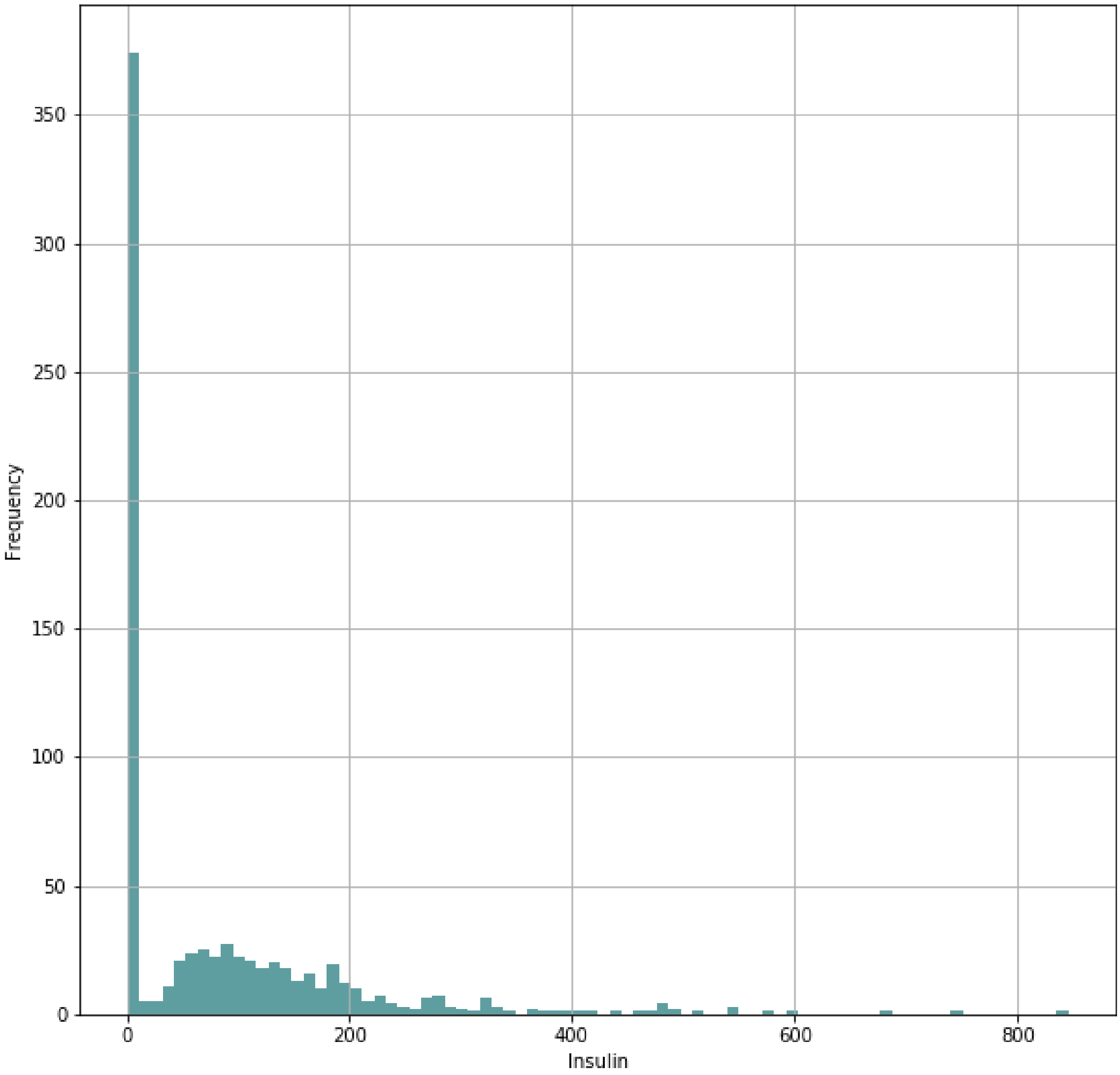
```
data_columns = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age']
for each in data_columns:
    fig1, ax1 = plt.subplots(figsize =(10,10))
    plt.hist(data[each], bins=80,color = "cadetblue")
    plt.xlabel(each)
    plt.ylabel("Frequency")
    plt.grid()
    plt.show()
```

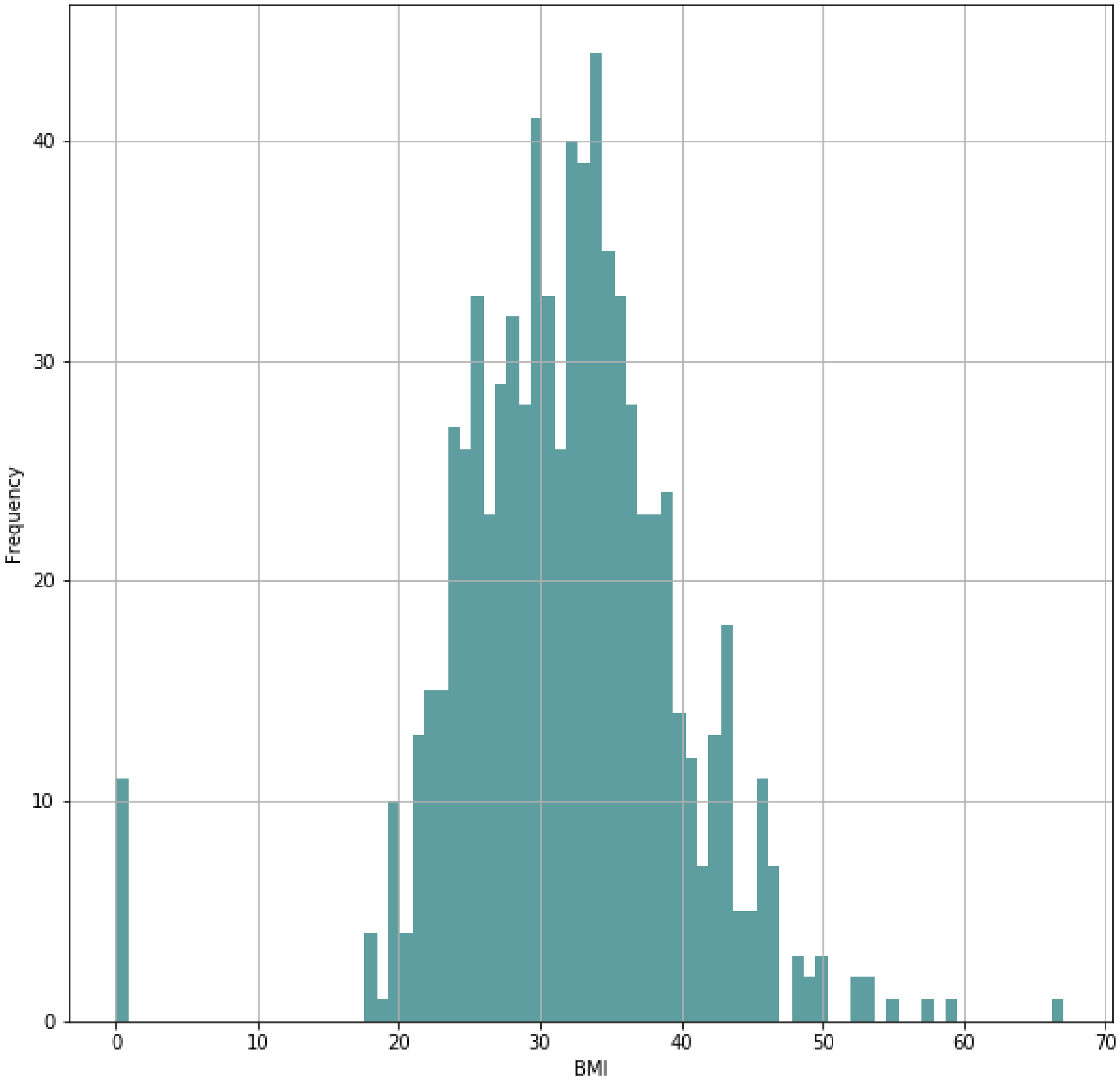

140
120
100
80
60
40
20
0
Frequency
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Pregnancies


Frequency
0
5
10
15
20
25
30
35
0
25
50
75
100
125
150
175
200
Glucose

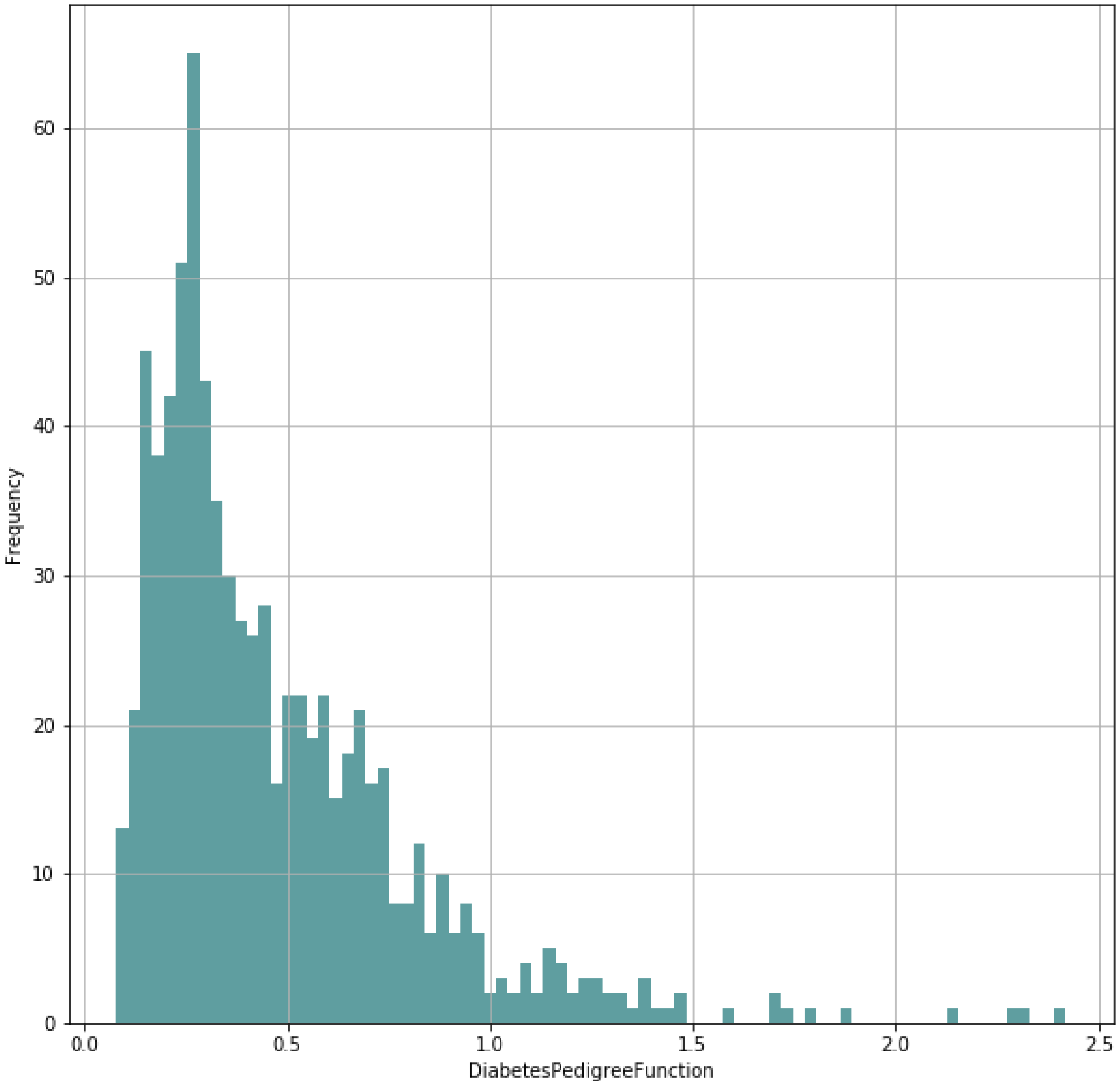

Frequency
BloodPressure
0
10
20
30
40
50
0
20
40
60
80
100
120

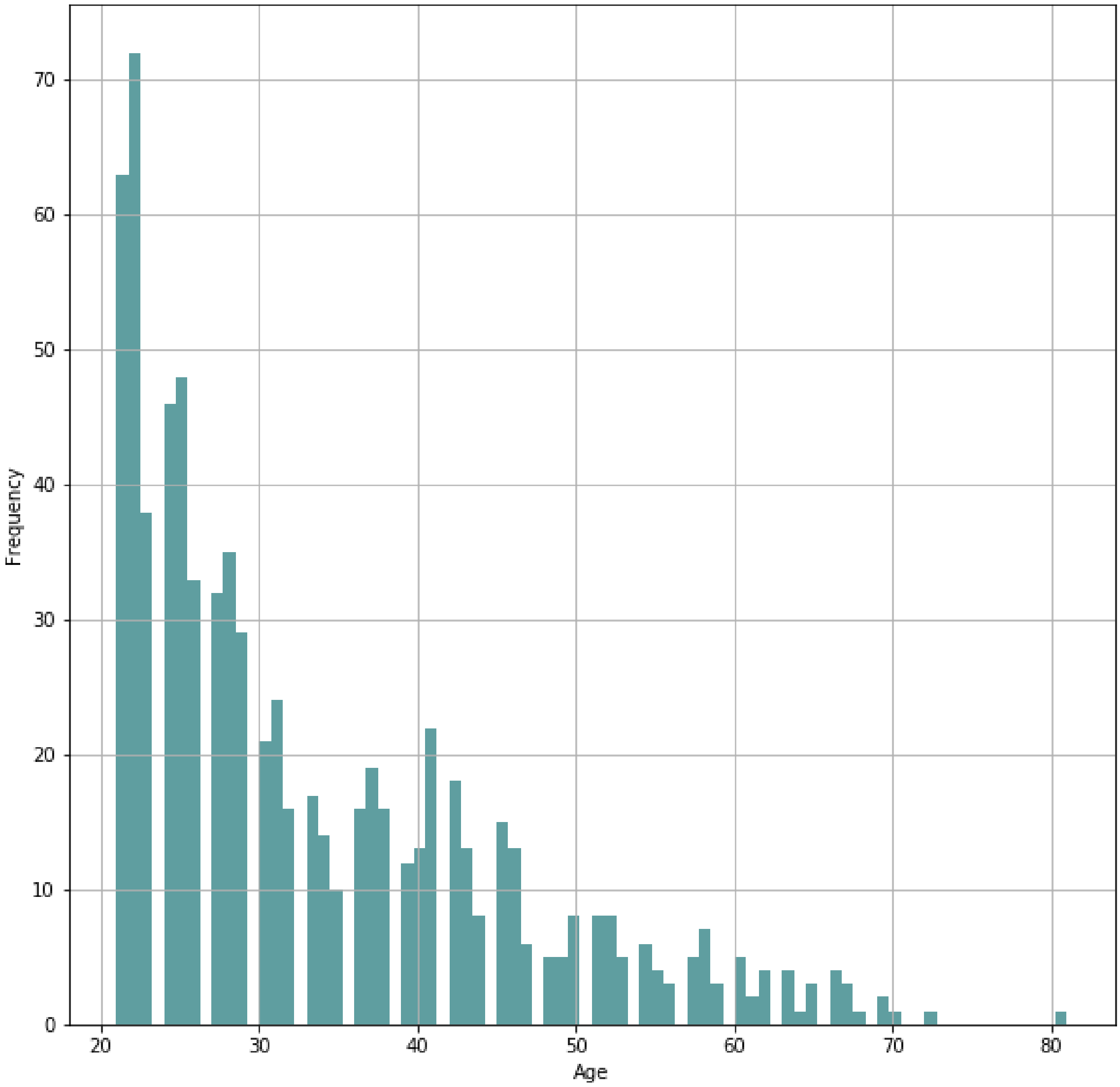

Frequency
200
150
100
50
0
0
20
40
60
80
100
SkinThickness


Frequency
350
300
250
200
150
100
50
0
0
200
400
600
800
Insulin



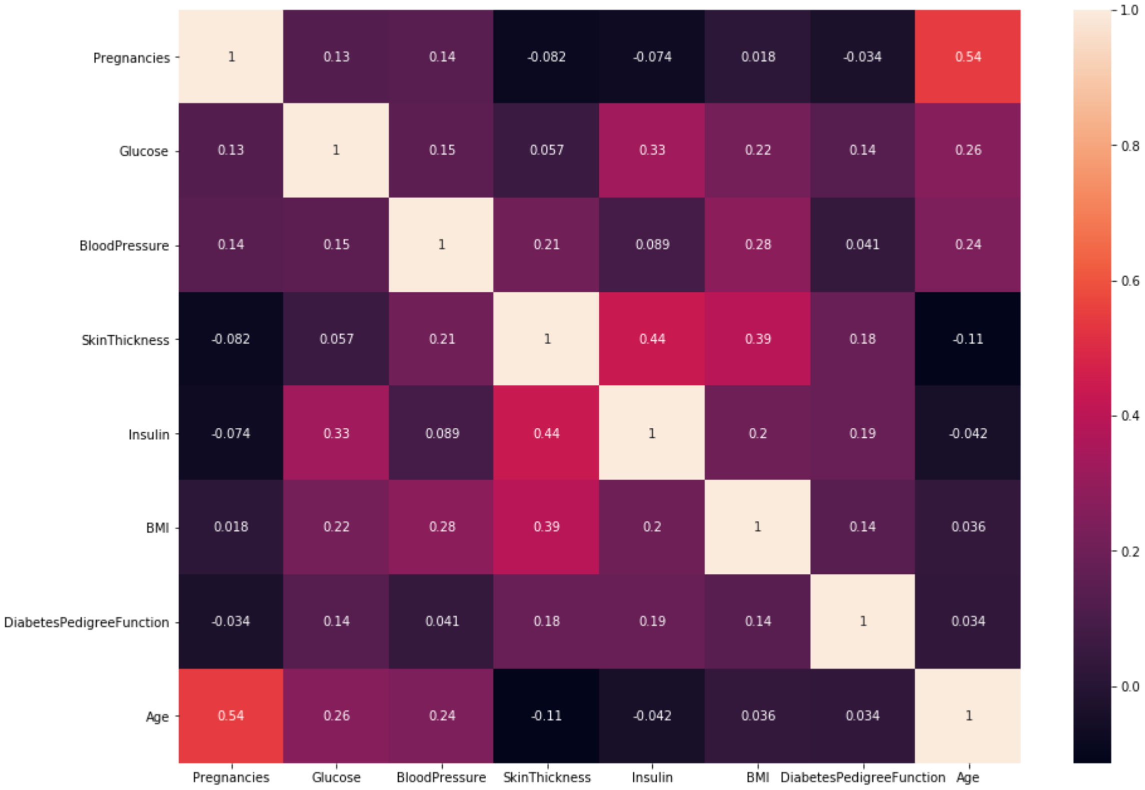
Frequency
40
30
20
10
0
0
10
20
30
40
50
60
70
BMI


60
50
40
30
20
10
0
Frequency
0.0
0.5
1.0
1.5
2.0
2.5
DiabetesPedigreeFunction


70
60
50
40
30
20
10
0
Frequency
20
30
40
50
60
70
80
Age

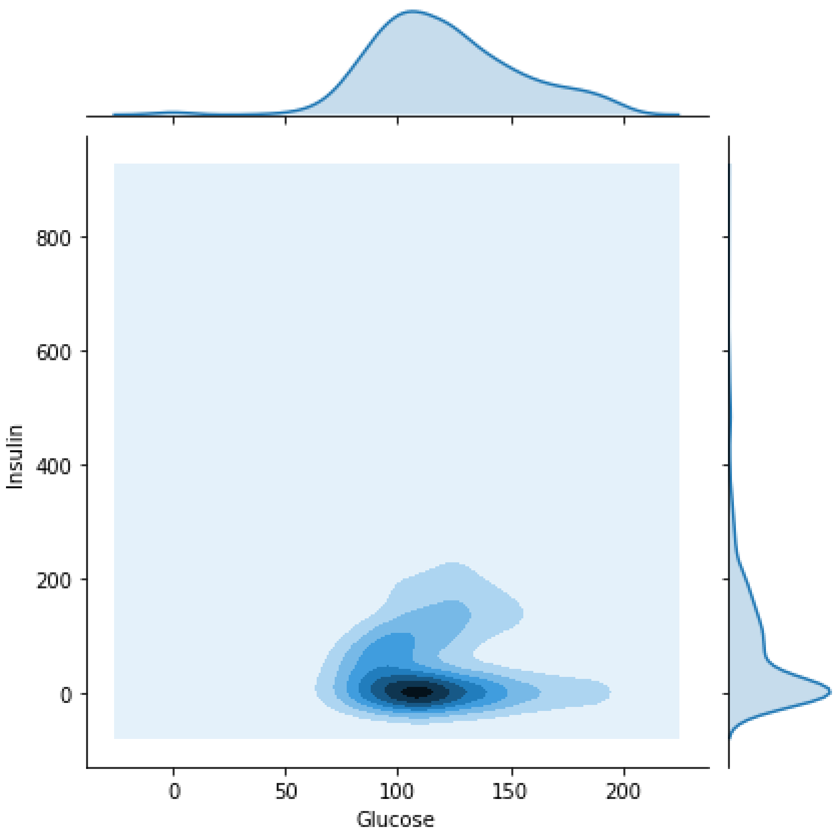
In [20]:

```
df = pd.DataFrame(data,columns=data_columns)
f, ax = plt.subplots(figsize =(15,11))
corrMatrix = df.corr()
sns.heatmap(corrMatrix, annot=True)
plt.show()
```



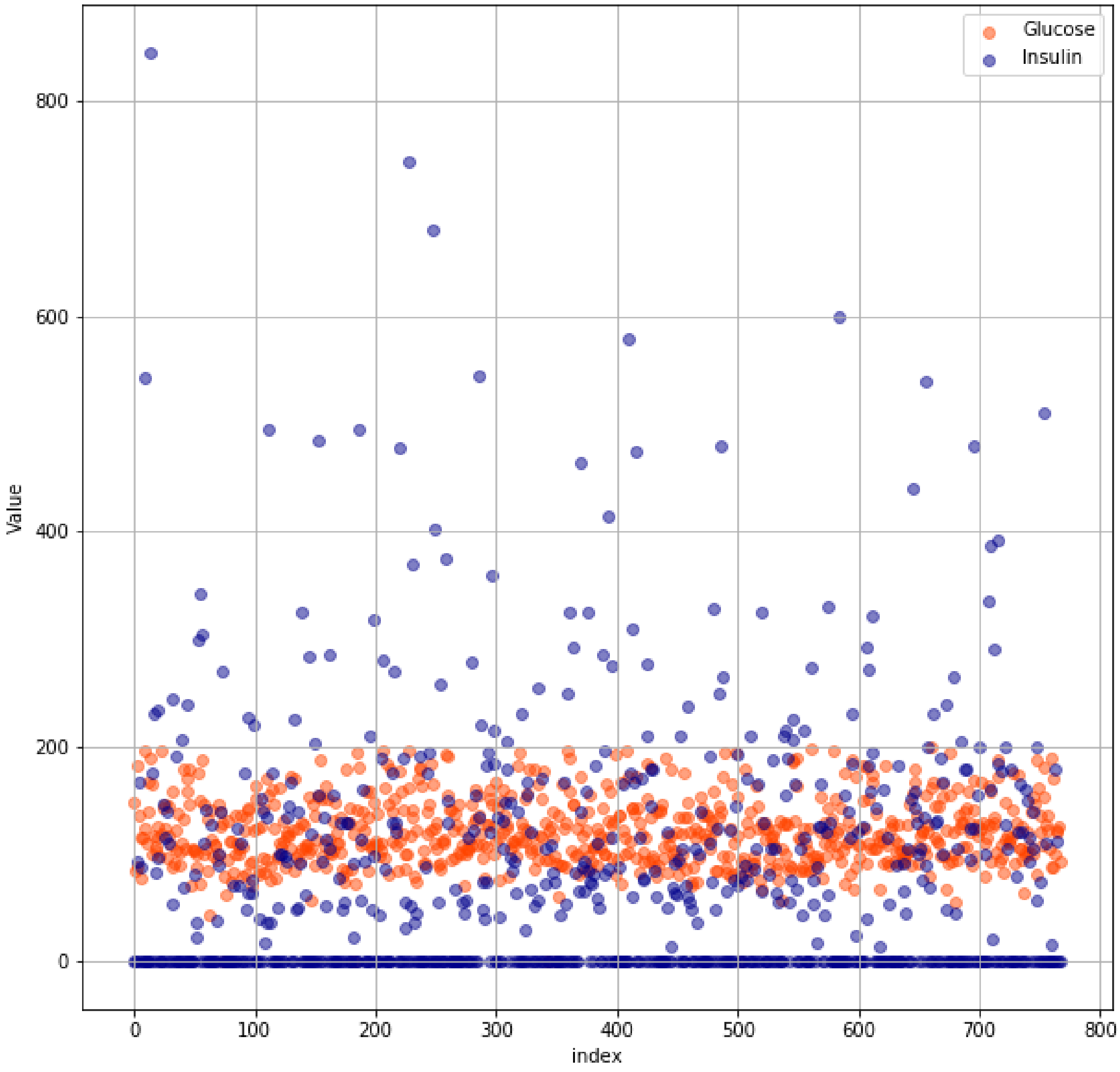
In [21]:

```
g = sns.jointplot(
    data=data,
    x="Glucose", y="Insulin",
    kind="kde",
)
plt.show()
```



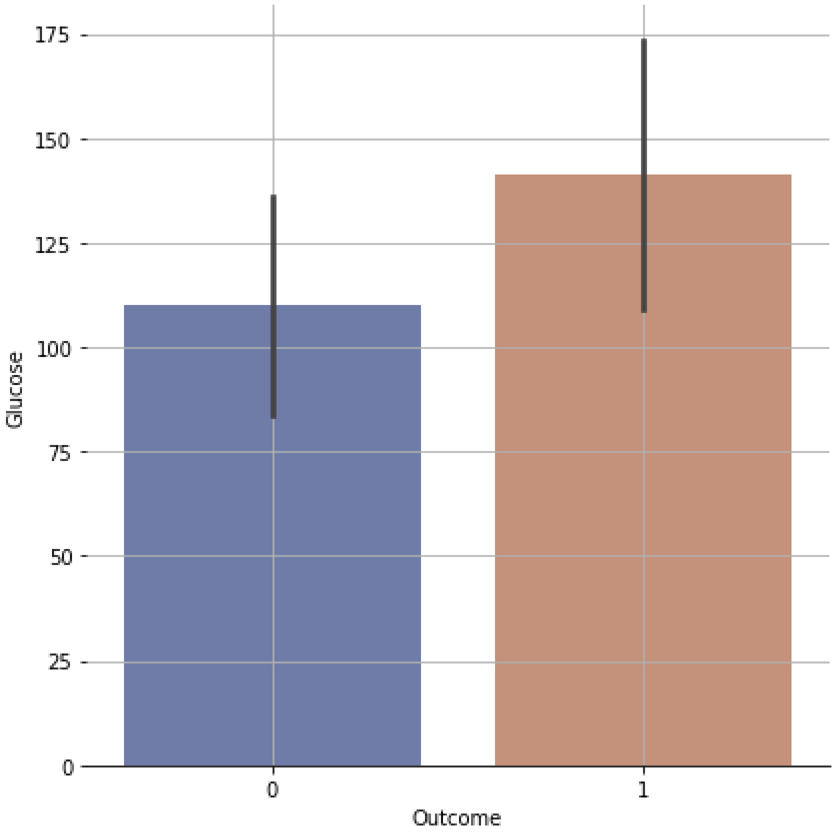
In [22]:

```
fig1, ax1 = plt.subplots(figsize =(10,10))
plt.scatter(data.index , data.Glucose,label =  "Glucose",alpha = 0.5,color = "orangered")
plt.scatter(data.index , data.Insulin,label =  "Insulin",alpha = 0.5,color = "darkblue")
plt.legend(loc ="best")
plt.xlabel("index")
plt.ylabel("Value")
plt.grid()
plt.show()
```



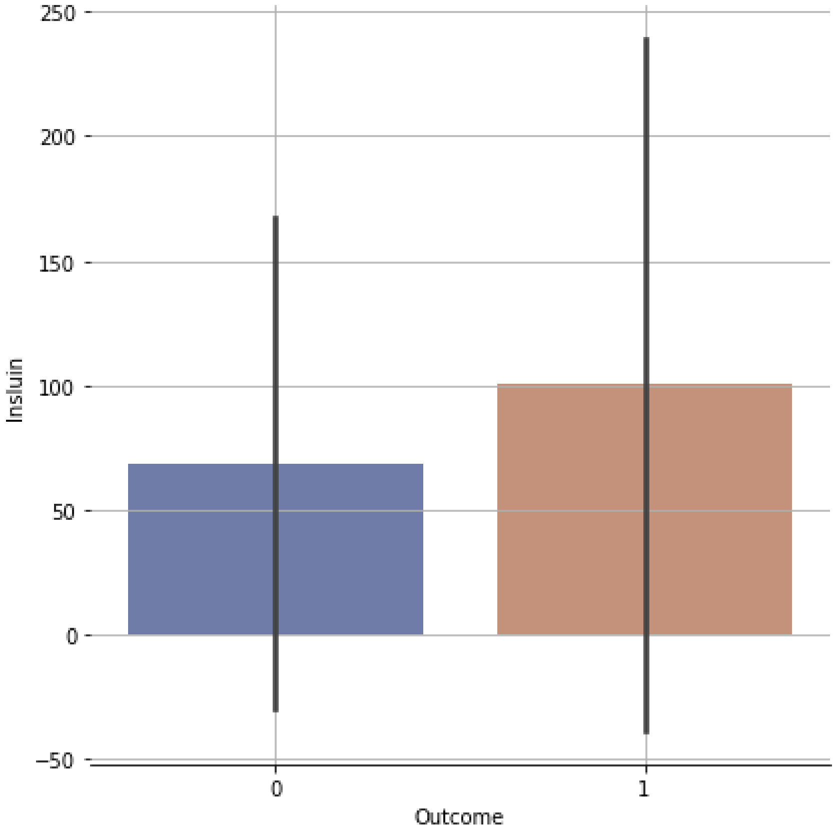
In [23]:

```
g = sns.catplot(
    data=data, kind="bar",
    x="Outcome", y="Glucose",
    ci="sd", palette="dark", alpha=.6, height=6,
)
g.despine(left=True)
g.set_axis_labels("Outcome", "Glucose")
plt.grid()
plt.show()
```



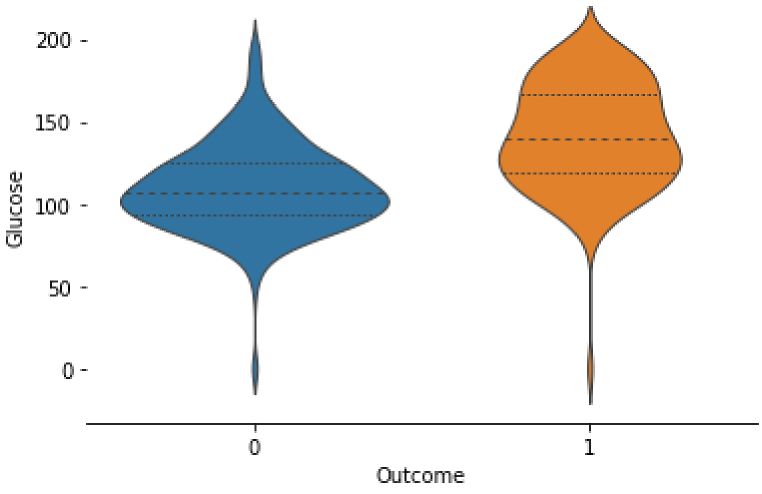
In [24]:

```
g = sns.catplot(
    data=data, kind="bar",
    x="Outcome", y="Insulin",
    ci="sd", palette="dark", alpha=.6, height=6,
)
g.despine(left=True)
g.set_axis_labels("Outcome", "Insluin")
plt.grid()
plt.show()
```



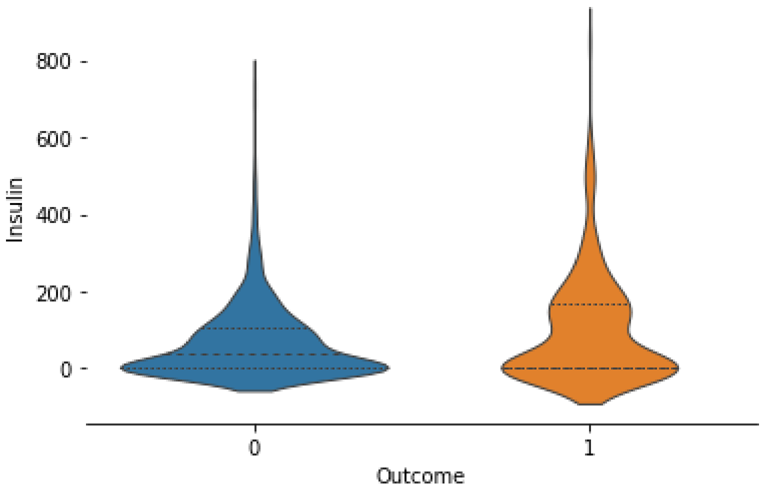
In [25]:

```
sns.violinplot(data=data, x="Outcome", y="Glucose",
               split=True, inner="quart", linewidth=1,)
sns.despine(left=True)
plt.show()
```



In [26]:

```
sns.violinplot(data=data, x="Outcome", y="Insulin",
               split=True, inner="quart", linewidth=1,)
sns.despine(left=True)
plt.show()
```



In [ ]:

```
```