In [9]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
import missingno as msno
from pandas.plotting import scatter_matrix
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

In [10]:

```
path = 'E:\power bi\MeriSKILL-Intern-Pinaki-DIABETES-PATIENTS-TASK-2-main\Pinaki_Di
df_diabetics = pd.read_csv(path)
df_diabetics_info = df_diabetics.info()
display(df_diabetics.head())
```

```
<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
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFuncti
0	6	148	72	35	0	33.6	0.6
1	1	85	66	29	0	26.6	0.3
2	8	183	64	0	0	23.3	0.6
3	1	89	66	23	94	28.1	0.1
4	0	137	40	35	168	43.1	2.2

In [11]:

```
df_diabetics.describe()
```

Out[11]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Dial
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 [12]:

```
df_diabetics.describe().T
```

Out[12]:

	count	mean	std	min	25%	50%	
Pregnancies	768.0	3.845052	3.369578	0.000	1.00000	3.0000	6.0
Glucose	768.0	120.894531	31.972618	0.000	99.00000	117.0000	140.2
BloodPressure	768.0	69.105469	19.355807	0.000	62.00000	72.0000	80.0
SkinThickness	768.0	20.536458	15.952218	0.000	0.00000	23.0000	32.0
Insulin	768.0	79.799479	115.244002	0.000	0.00000	30.5000	127.2
BMI	768.0	31.992578	7.884160	0.000	27.30000	32.0000	36.6
DiabetesPedigreeFunction	768.0	0.471876	0.331329	0.078	0.24375	0.3725	0.6
Age	768.0	33.240885	11.760232	21.000	24.00000	29.0000	41.0
Outcome	768.0	0.348958	0.476951	0.000	0.00000	0.0000	1.0

In [13]:

```
df_diabetics.isnull()
```

Out[13]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeF
0	False	False	False	False	False	False	
1	False	False	False	False	False	False	
2	False	False	False	False	False	False	
3	False	False	False	False	False	False	
4	False	False	False	False	False	False	
...	...	...	...	...	...	...	
763	False	False	False	False	False	False	
764	False	False	False	False	False	False	
765	False	False	False	False	False	False	
766	False	False	False	False	False	False	
767	False	False	False	False	False	False	

768 rows × 9 columns

In [14]:

```
df_diabetics.isnull().sum()
```

Out[14]:

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

In [30]:

```
df_diabetics_copy = df_diabetics.copy
df_diabetics_copy=df_diabetics_copy(deep = True)
df_diabetics_copy[['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI']]=
```

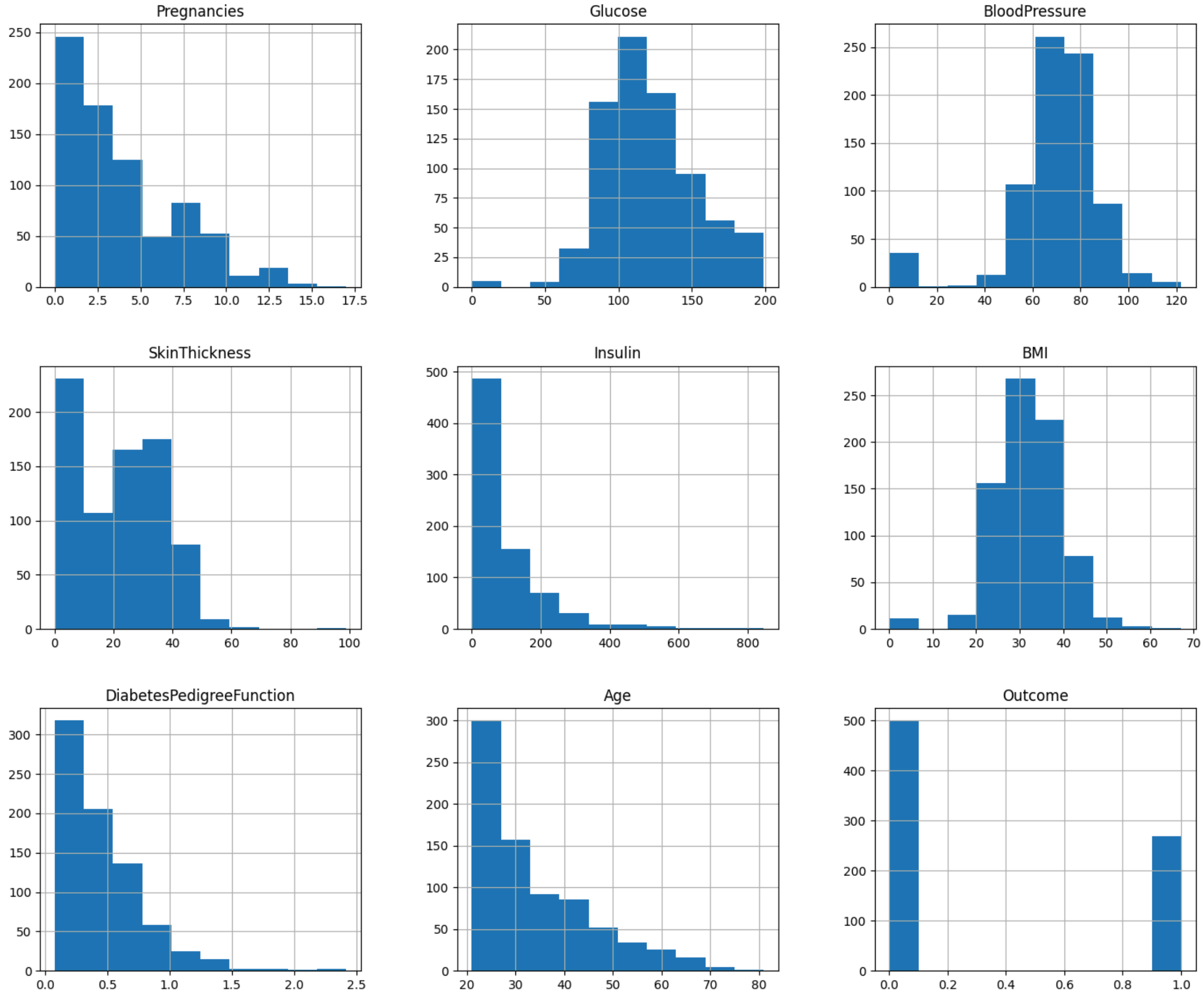
In [31]:

```
print (df_diabetics_copy.isnull().sum())
```

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

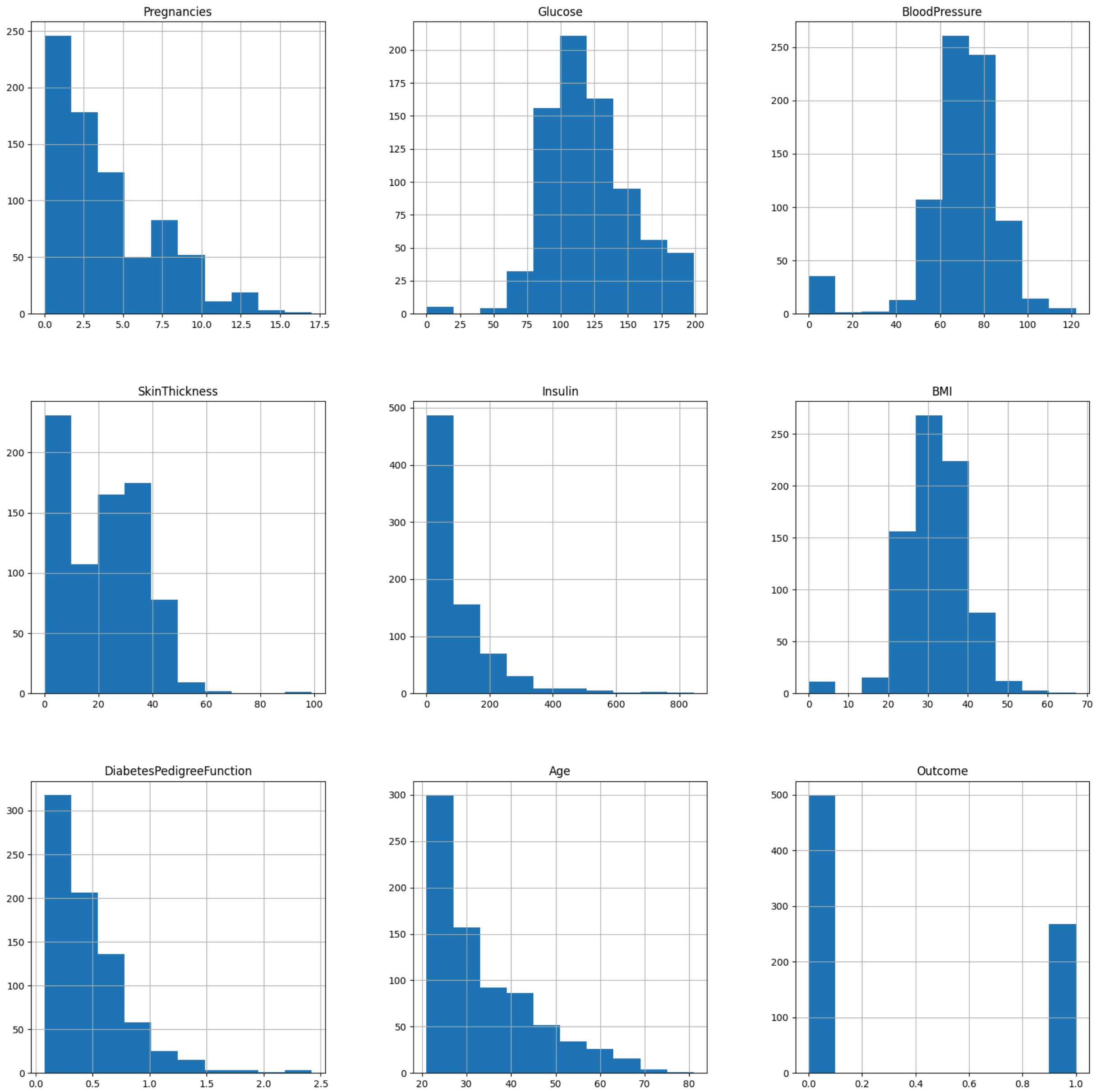
In [17]:
```
df_diabetics.hist(figsize=(17,14))
plt.show
```

Out[17]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```

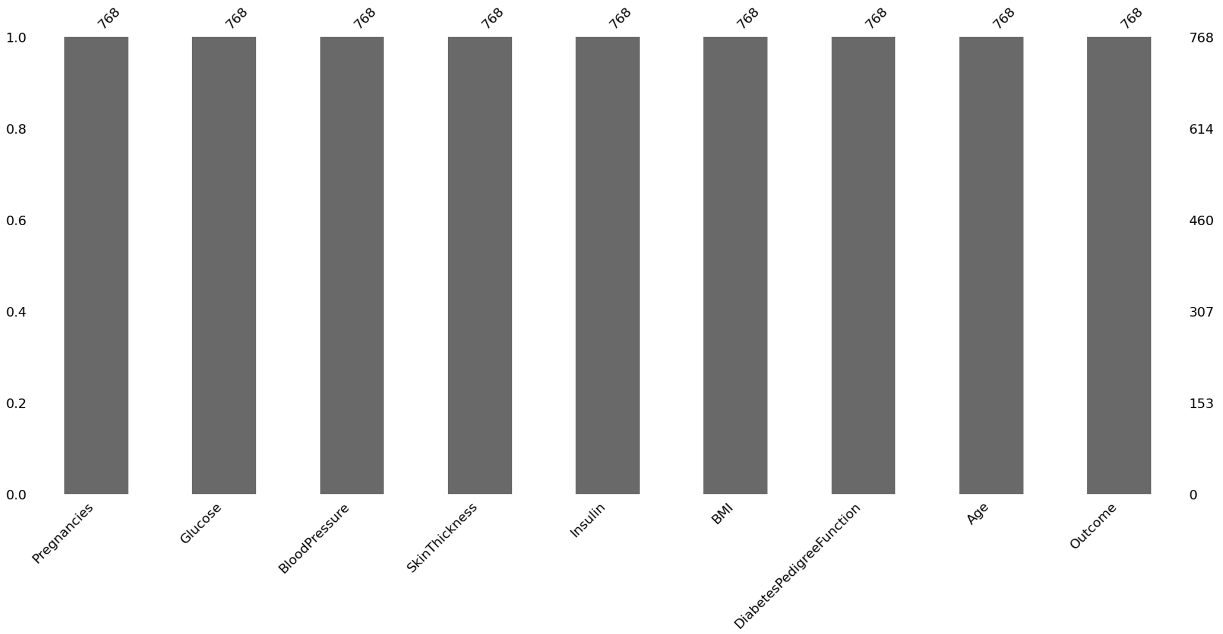


In [18]:
```
df_diabetics_copy['Glucose'].fillna(df_diabetics_copy['Glucose'].mean(), inplace =
df_diabetics_copy['BloodPressure'].fillna(df_diabetics_copy['BloodPressure'].mean()
df_diabetics_copy['SkinThickness'].fillna(df_diabetics_copy['SkinThickness'].median
df_diabetics_copy['Insulin'].fillna(df_diabetics_copy['Insulin'].median(), inplace
df_diabetics_copy['BMI'].fillna(df_diabetics_copy['BMI'].median(), inplace = True)
```

In [19]:
```
p= df_diabetics_copy.hist(figsize=(20,20))
```

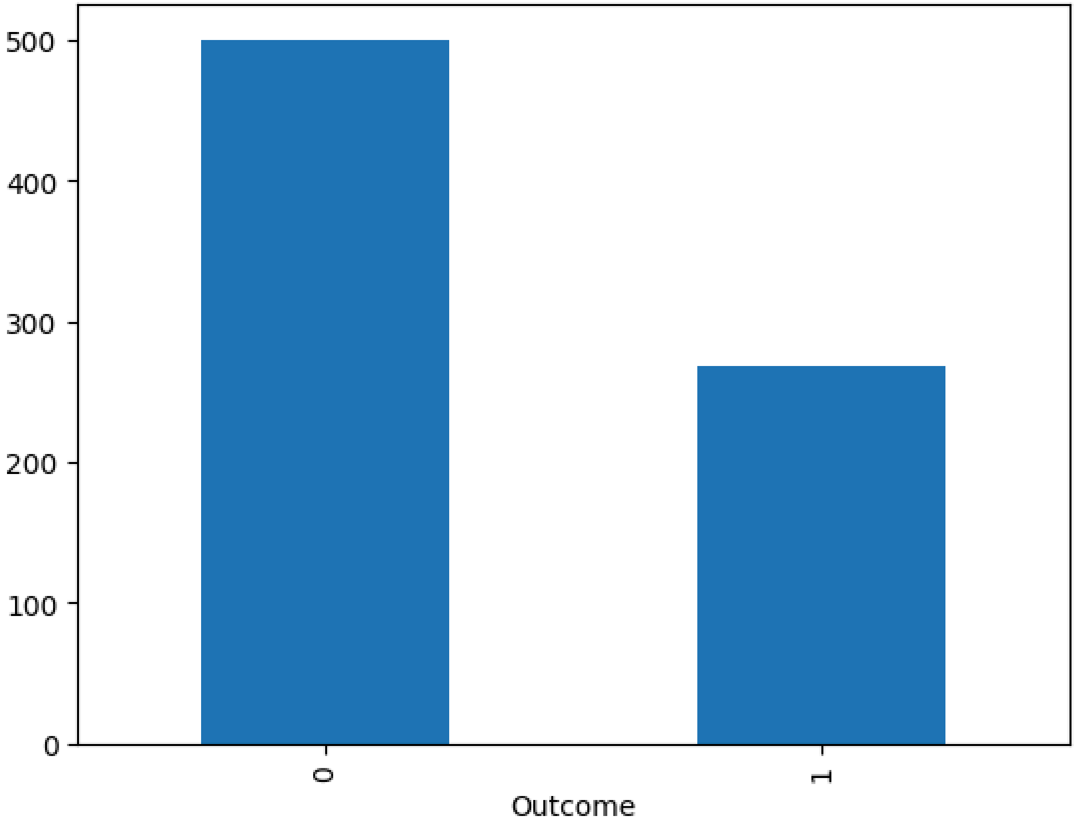


In [20]:
```
p = msno.bar(df_diabetics)
```

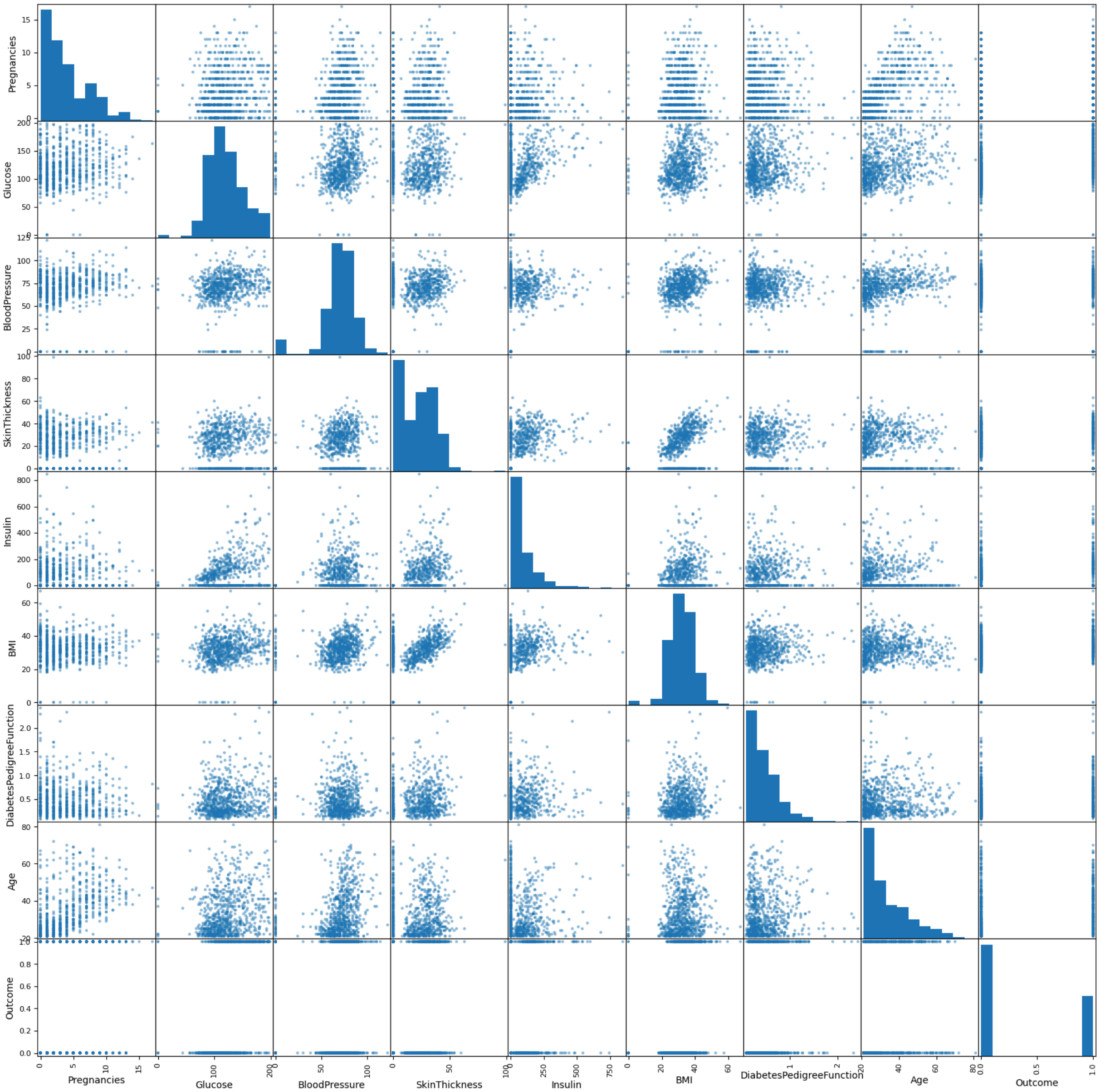


In [21]:
```
color_wheel={1: "#0392cf", 2: "#7bc043"}
colors=df_diabetics["Outcome"].map(lambda x:color_wheel.get(x+1))
print(df_diabetics.Outcome.value_counts())
p = df_diabetics.Outcome.value_counts().plot(kind="bar")
```

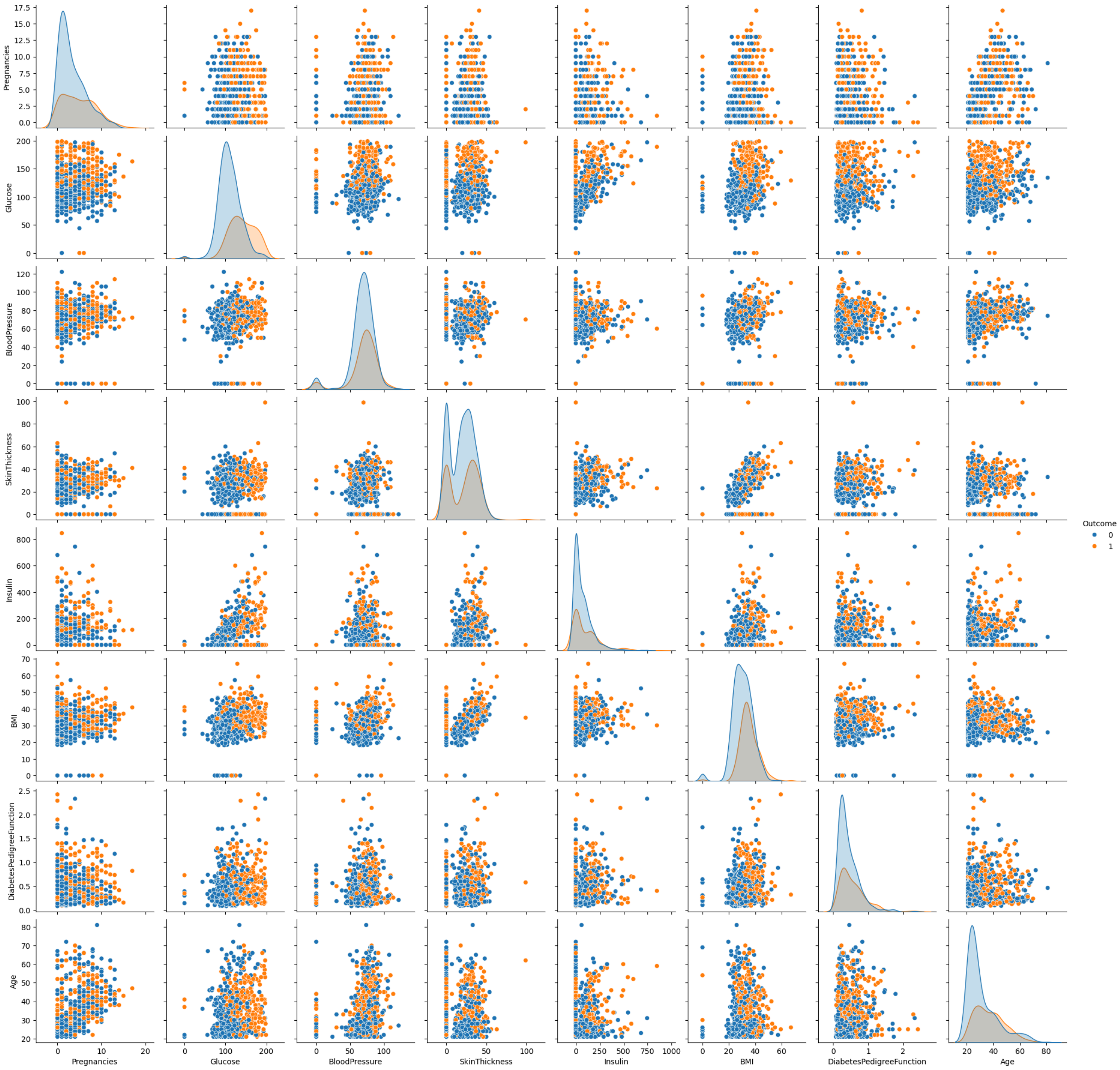
```
Outcome
0    500
1    268
Name: count, dtype: int64
```



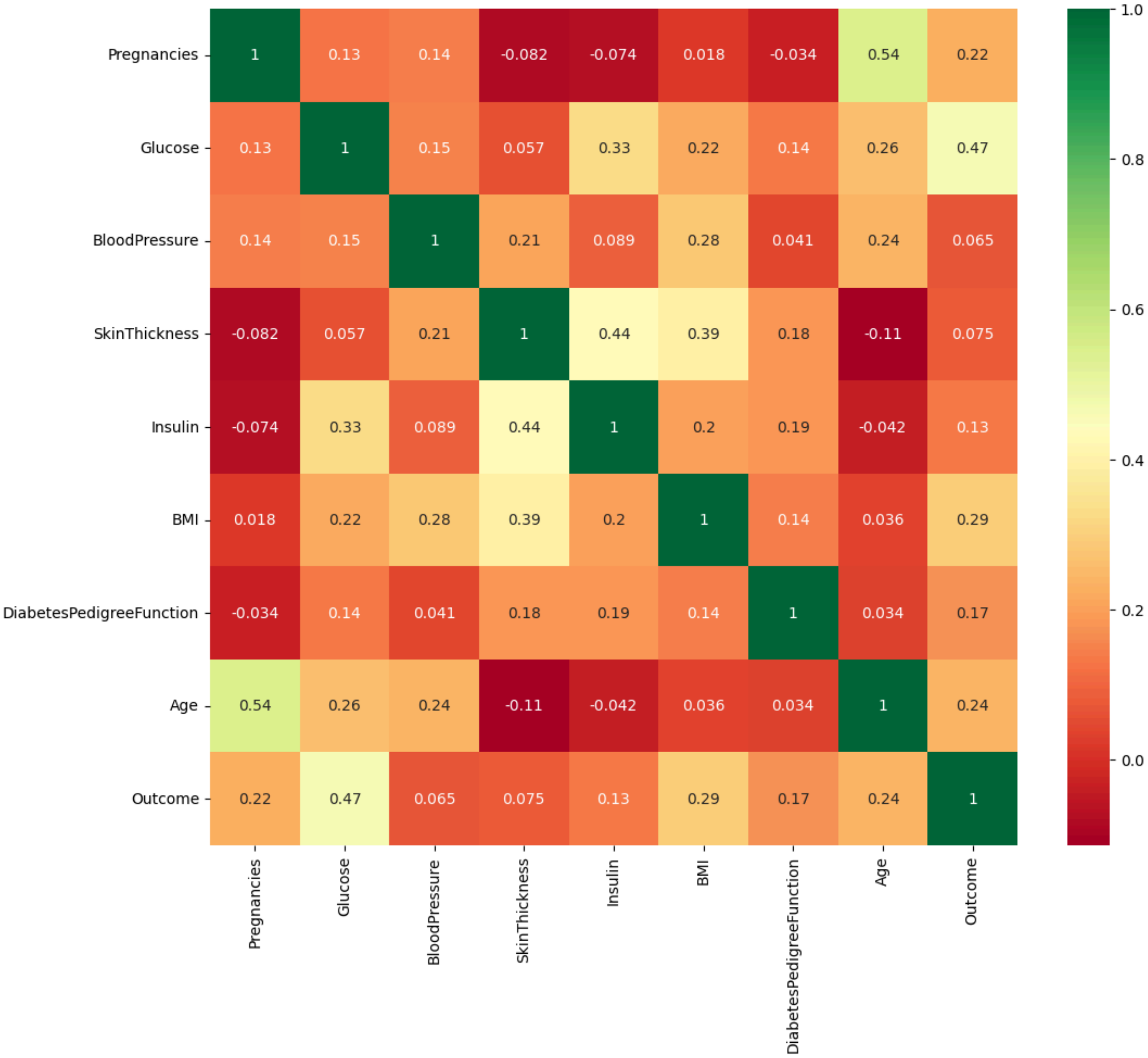
In [22]:
```
p = scatter_matrix(df_diabetics, figsize=(20,20))
```



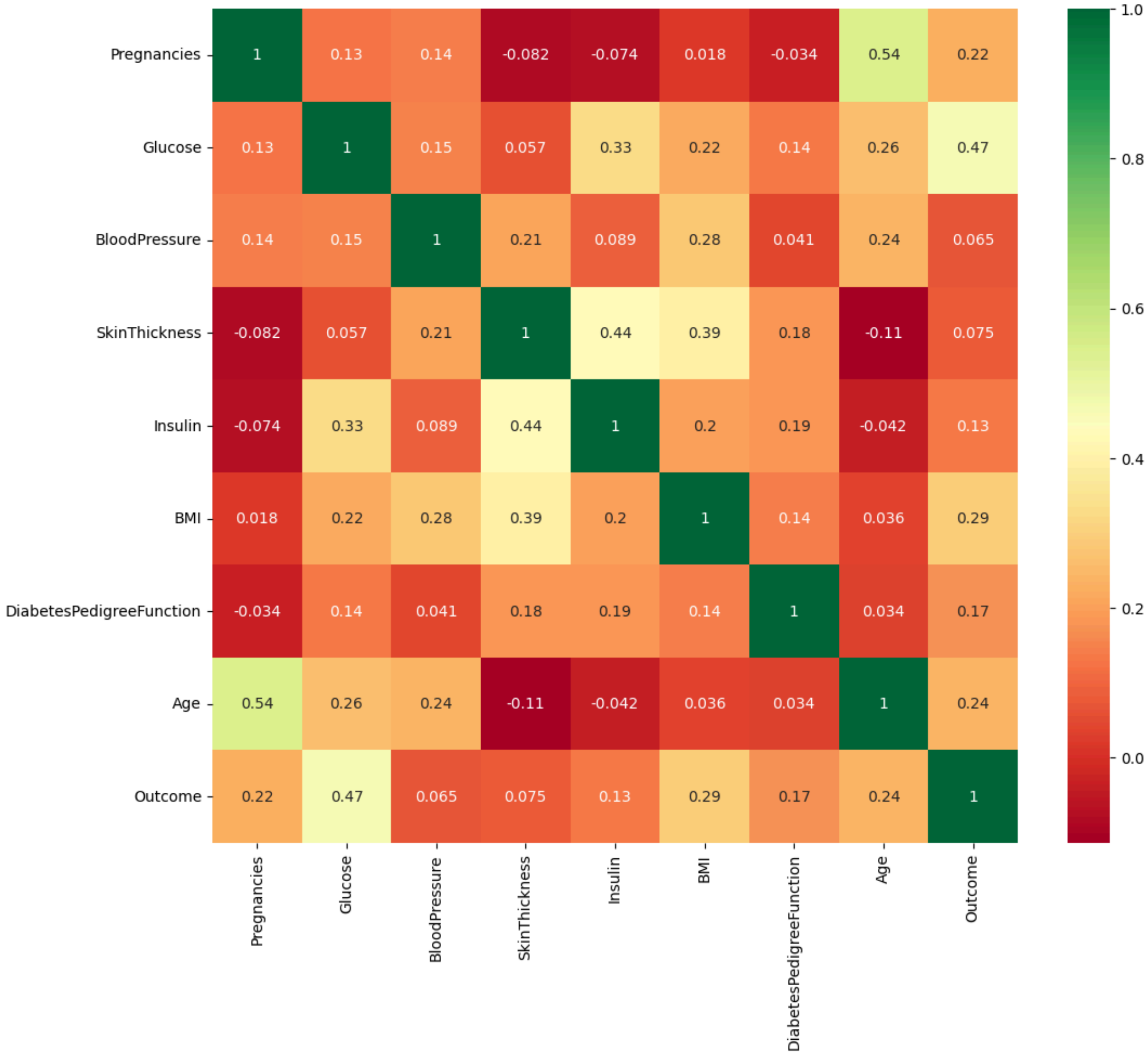
In [23]:
```
sns.pairplot(df_diabetics_copy, hue='Outcome')
plt.show()
```



In [24]:
```
plt.figure(figsize=(12,10))
p=sns.heatmap(df_diabetics.corr(), annot = True, cmap = "RdYlGn")
```



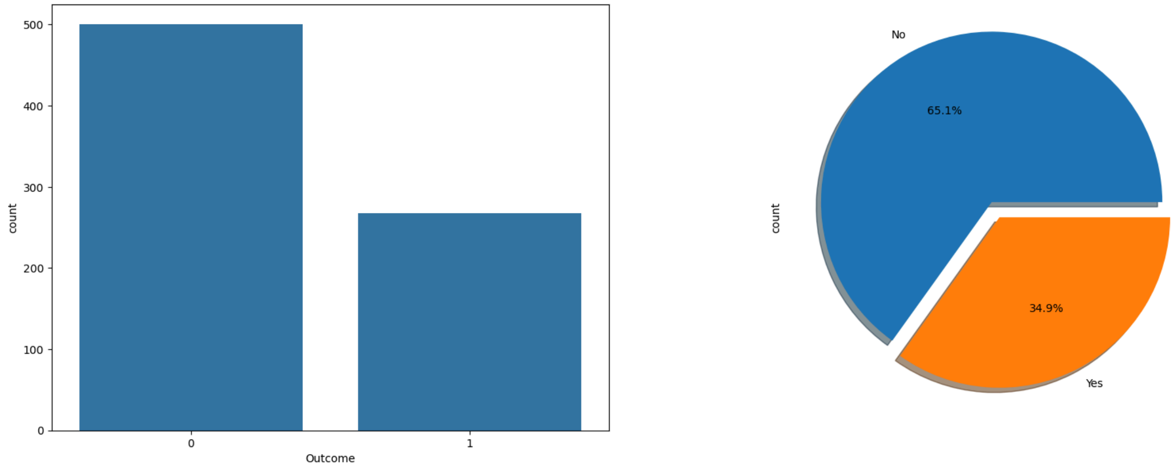
In [25]:
```
plt.figure(figsize=(12,10))
p=sns.heatmap(df_diabetics_copy.corr(), annot = True, cmap = "RdYlGn")
```



In [26]:
```
fig, ax = plt.subplots(1,2, figsize=(20,7))

sns.countplot(data = df_diabetics, x = "Outcome", ax = ax[0])
df_diabetics["Outcome"].value_counts().plot.pie(explode=[0.1,0], autopct = "%1.1f%%
labels= ["No", "Yes"], shadow= True, ax = ax[1])
```

Out[26]:
```
<Axes: ylabel='count'>
```



In [32]:

```
print("we observe from the above plot that:")
print("65.1% patients in the dataset do not have diabetes")
print("34.9% patients in the dataset has diabetes")
```

```
we observe from the above plot that:
65.1% patients in the dataset do not have diabetes
34.9% patients in the dataset has diabetes
```

In [ ]:

```
1. It has a decent level of precision, indicating that when it predicts positie cas
It's correct about 65% of the time.
2. Out of the 768 patients, 268 hae been diagnosed with diabetes.
3. patients with high blood pressure has greater chances of diabetes.
4. An increase in blood pressure BMI and sin Thickness also increases.
5. Increasing level of glucose and insulin increases chances of diabetes.
```