In [1]:

```
#Diabetes Prediction
!pip install mlxtend
!pip install missingno
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
from mlxtend.plotting import plot_decision_regions
import missingno as msno
from pandas.plotting import scatter_matrix
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import confusion_matrix
from sklearn import metrics
from sklearn.metrics import classification_report
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
diabetes_df=pd.read_csv('diabetes.csv')
diabetes_df.head()
```

```
Requirement already satisfied: mlxtend in c:\users\dell\anaconda3\lib\sit
e-packages (0.21.0)
Requirement already satisfied: setuptools in c:\users\dell\anaconda3\lib
\site-packages (from mlxtend) (61.2.0)
Requirement already satisfied: joblib>=0.13.2 in c:\users\dell\anaconda3
\lib\site-packages (from mlxtend) (1.1.0)
Requirement already satisfied: matplotlib>=3.0.0 in c:\users\dell\anacond
a3\lib\site-packages (from mlxtend) (3.5.1)
Requirement already satisfied: scikit-learn>=1.0.2 in c:\users\dell\anaco
nda3\lib\site-packages (from mlxtend) (1.0.2)
Requirement already satisfied: scipy>=1.2.1 in c:\users\dell\anaconda3\li
b\site-packages (from mlxtend) (1.7.3)
Requirement already satisfied: pandas>=0.24.2 in c:\users\dell\anaconda3
\lib\site-packages (from mlxtend) (1.4.2)
Requirement already satisfied: numpy>=1.16.2 in c:\users\dell\anaconda3\l
ib\site-packages (from mlxtend) (1.21.5)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\dell\anacond
a3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (4.25.0)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\dell\anaconda
3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (3.0.4)
Requirement already satisfied: cycler>=0.10 in c:\users\dell\anaconda3\li
b\site-packages (from matplotlib>=3.0.0->mlxtend) (0.11.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\dell\anacond
a3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (1.3.2)
Requirement already satisfied: packaging>=20.0 in c:\users\dell\anaconda3
\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (21.3)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\dell\anac
onda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (2.8.2)
Requirement already satisfied: pillow>=6.2.0 in c:\users\dell\anaconda3\l
ib\site-packages (from matplotlib>=3.0.0->mlxtend) (9.0.1)
Requirement already satisfied: pytz>=2020.1 in c:\users\dell\anaconda3\li
b\site-packages (from pandas>=0.24.2->mlxtend) (2021.3)
Requirement already satisfied: six>=1.5 in c:\users\dell\anaconda3\lib\si
te-packages (from python-dateutil>=2.7->matplotlib>=3.0.0->mlxtend) (1.1
6.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\dell\anac
onda3\lib\site-packages (from scikit-learn>=1.0.2->mlxtend) (2.2.0)
Requirement already satisfied: missingno in c:\users\dell\anaconda3\lib\s
ite-packages (0.5.2)
Requirement already satisfied: seaborn in c:\users\dell\anaconda3\lib\sit
e-packages (from missingno) (0.11.2)
Requirement already satisfied: numpy in c:\users\dell\anaconda3\lib\site-
packages (from missingno) (1.21.5)
Requirement already satisfied: scipy in c:\users\dell\anaconda3\lib\site-
packages (from missingno) (1.7.3)
Requirement already satisfied: matplotlib in c:\users\dell\anaconda3\lib
\site-packages (from missingno) (3.5.1)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\dell\anacond
a3\lib\site-packages (from matplotlib->missingno) (1.3.2)
Requirement already satisfied: packaging>=20.0 in c:\users\dell\anaconda3
\lib\site-packages (from matplotlib->missingno) (21.3)
Requirement already satisfied: cycler>=0.10 in c:\users\dell\anaconda3\li
b\site-packages (from matplotlib->missingno) (0.11.0)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\dell\anaconda
3\lib\site-packages (from matplotlib->missingno) (3.0.4)
Requirement already satisfied: pillow>=6.2.0 in c:\users\dell\anaconda3\l
ib\site-packages (from matplotlib->missingno) (9.0.1)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\dell\anac
onda3\lib\site-packages (from matplotlib->missingno) (2.8.2)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\dell\anacond
a3\lib\site-packages (from matplotlib->missingno) (4.25.0)
```

```
Requirement already satisfied: six>=1.5 in c:\users\dell\anaconda3\lib\si
te-packages (from python-dateutil>=2.7->matplotlib->missingno) (1.16.0)
Requirement already satisfied: pandas>=0.23 in c:\users\dell\anaconda3\li
b\site-packages (from seaborn->missingno) (1.4.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\dell\anaconda3\li
b\site-packages (from pandas>=0.23->seaborn->missingno) (2021.3)
```

Out[1]:

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

In [2]:

```
diabetes_df.columns
```

Out[2]:

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

In [3]:

```
diabetes_df.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 [8]:

```
diabetes_df.describe()
```

Out[8]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Dia
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 [9]:

```
diabetes_df.describe().T
```

Out[9]:

	count	mean	std	min	25%	50%	7
Pregnancies	768.0	3.845052	3.369578	0.000	1.00000	3.0000	6.00
Glucose	768.0	120.894531	31.972618	0.000	99.00000	117.0000	140.25
BloodPressure	768.0	69.105469	19.355807	0.000	62.00000	72.0000	80.00
SkinThickness	768.0	20.536458	15.952218	0.000	0.00000	23.0000	32.00
Insulin	768.0	79.799479	115.244002	0.000	0.00000	30.5000	127.25
BMI	768.0	31.992578	7.884160	0.000	27.30000	32.0000	36.60
DiabetesPedigreeFunction	768.0	0.471876	0.331329	0.078	0.24375	0.3725	0.62
Age	768.0	33.240885	11.760232	21.000	24.00000	29.0000	41.00
Outcome	768.0	0.348958	0.476951	0.000	0.00000	0.0000	1.00

In [10]:

```
diabetes_df.isnull().head(10)
```

Out[10]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFu
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	
5	False	False	False	False	False	False	
6	False	False	False	False	False	False	
7	False	False	False	False	False	False	
8	False	False	False	False	False	False	
9	False	False	False	False	False	False	

In [11]:

```
diabetes_df.isnull().sum()
```

Out[11]:

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

In [12]:

```
diabetes_df_copy=diabetes_df.copy(deep=True)
diabetes_df_copy[['Glucose','BloodPressure','SkinThickness','Insulin','BMI']]=diabetes_
```

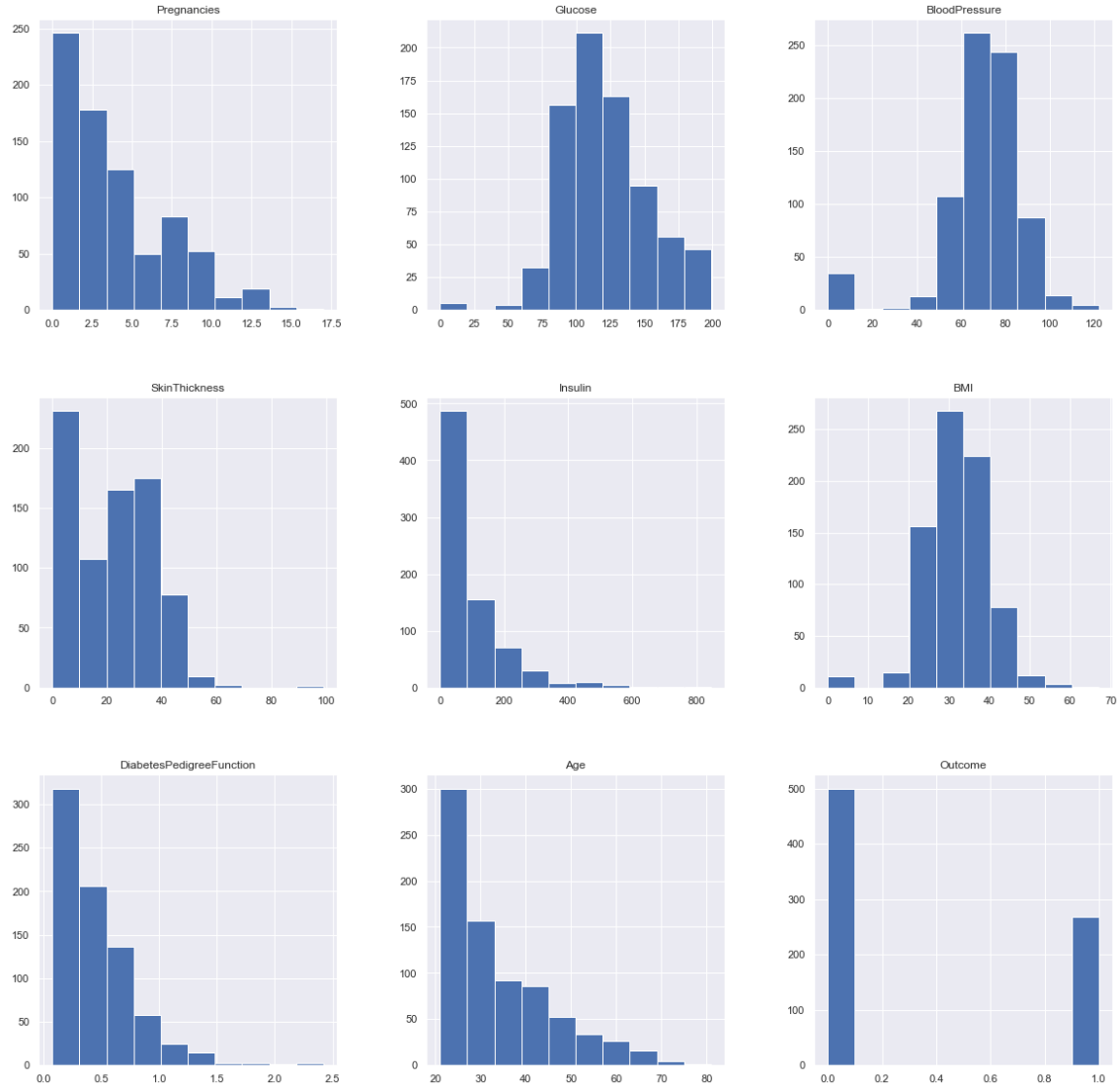
In [13]:

```
print(diabetes_df_copy.isnull().sum())
```

```
Pregnancies                 0
Glucose                     5
BloodPressure              35
SkinThickness             227
Insulin                   374
BMI                        11
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

In [15]:

```
p=diabetes_df.hist(figsize=(20,20))
```

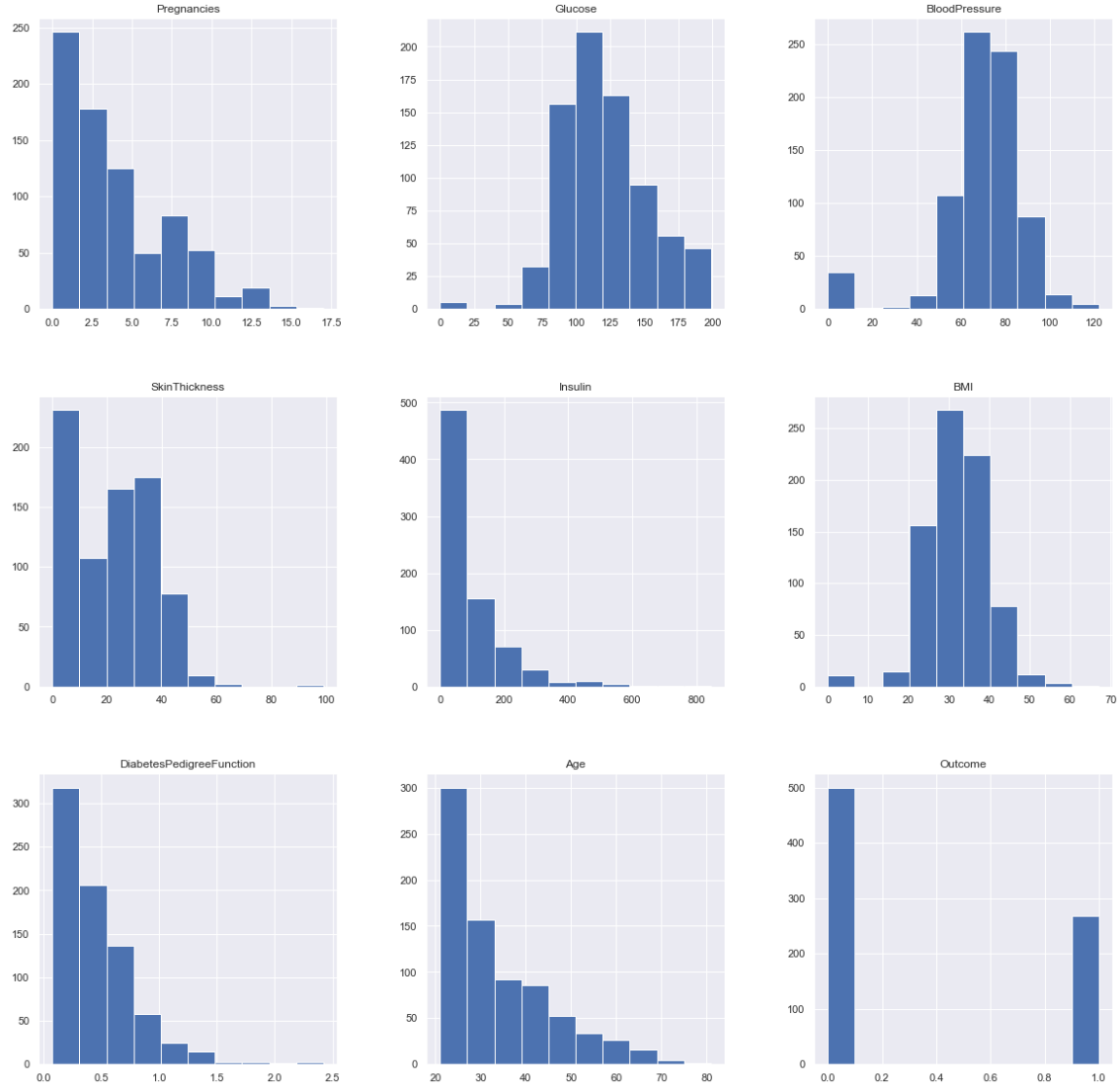


In [17]:

```
diabetes_df_copy['Glucose'].fillna(diabetes_df_copy['Glucose'].mean(), inplace = True)
diabetes_df_copy['BloodPressure'].fillna(diabetes_df_copy['BloodPressure'].mean(), inpla
diabetes_df_copy['SkinThickness'].fillna(diabetes_df_copy['SkinThickness'].median(), in
diabetes_df_copy['Insulin'].fillna(diabetes_df_copy['Insulin'].median(), inplace = True
diabetes_df_copy['BMI'].fillna(diabetes_df_copy['BMI'].median(), inplace = True)
```

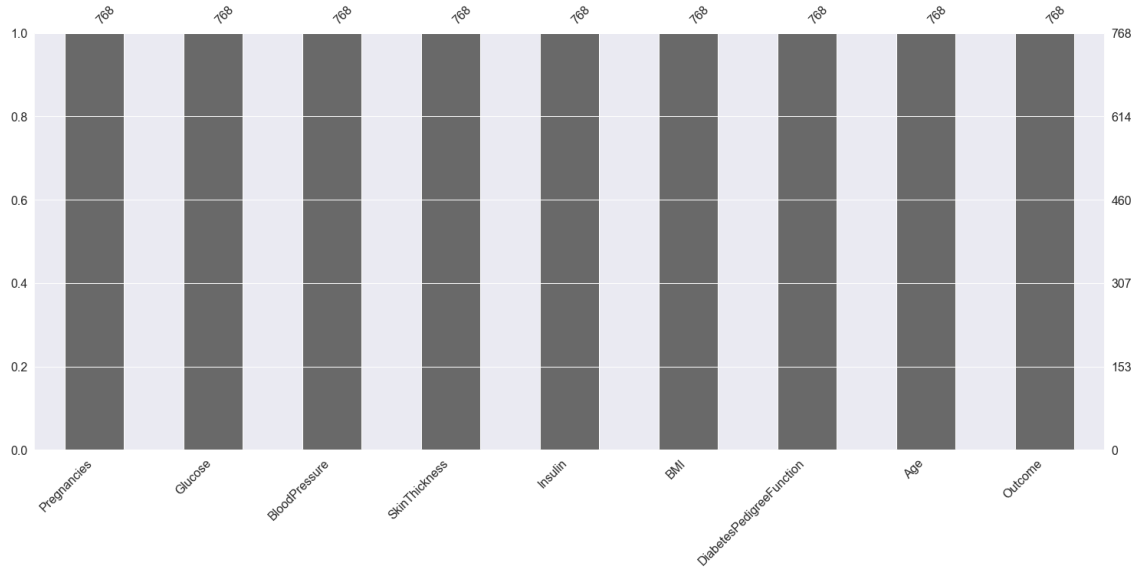
In [18]:

```
p=diabetes_df.hist(figsize=(20,20))
```



In [19]:

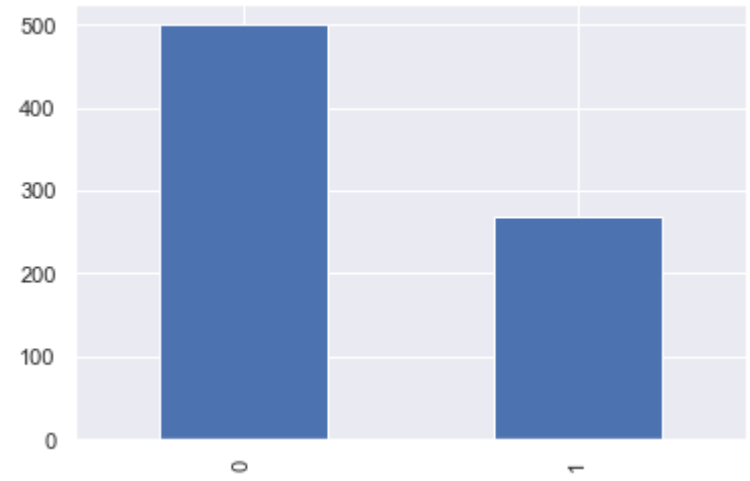
```
p = msno.bar(diabetes_df)
```



In [20]:

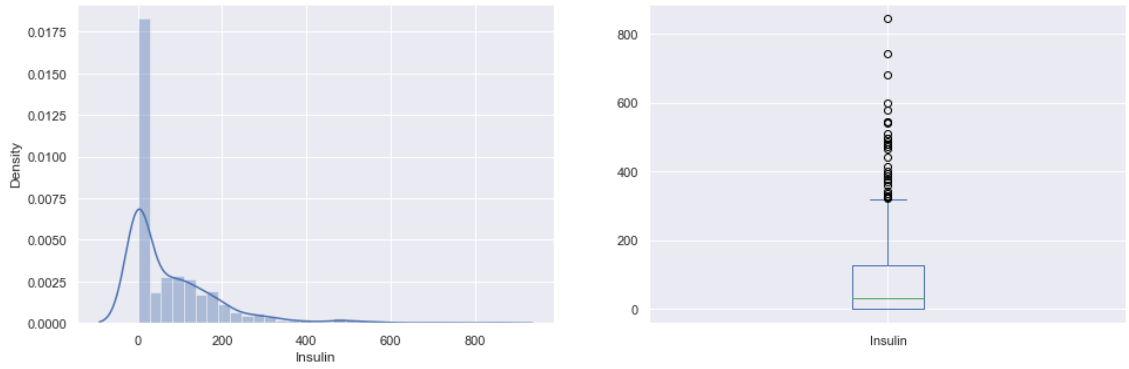
```
color_wheel = {1: "#0392cf", 2: "#7bc043"}
colors = diabetes_df["Outcome"].map(lambda x: color_wheel.get(x + 1))
print(diabetes_df.Outcome.value_counts())
p=diabetes_df.Outcome.value_counts().plot(kind="bar")
```

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



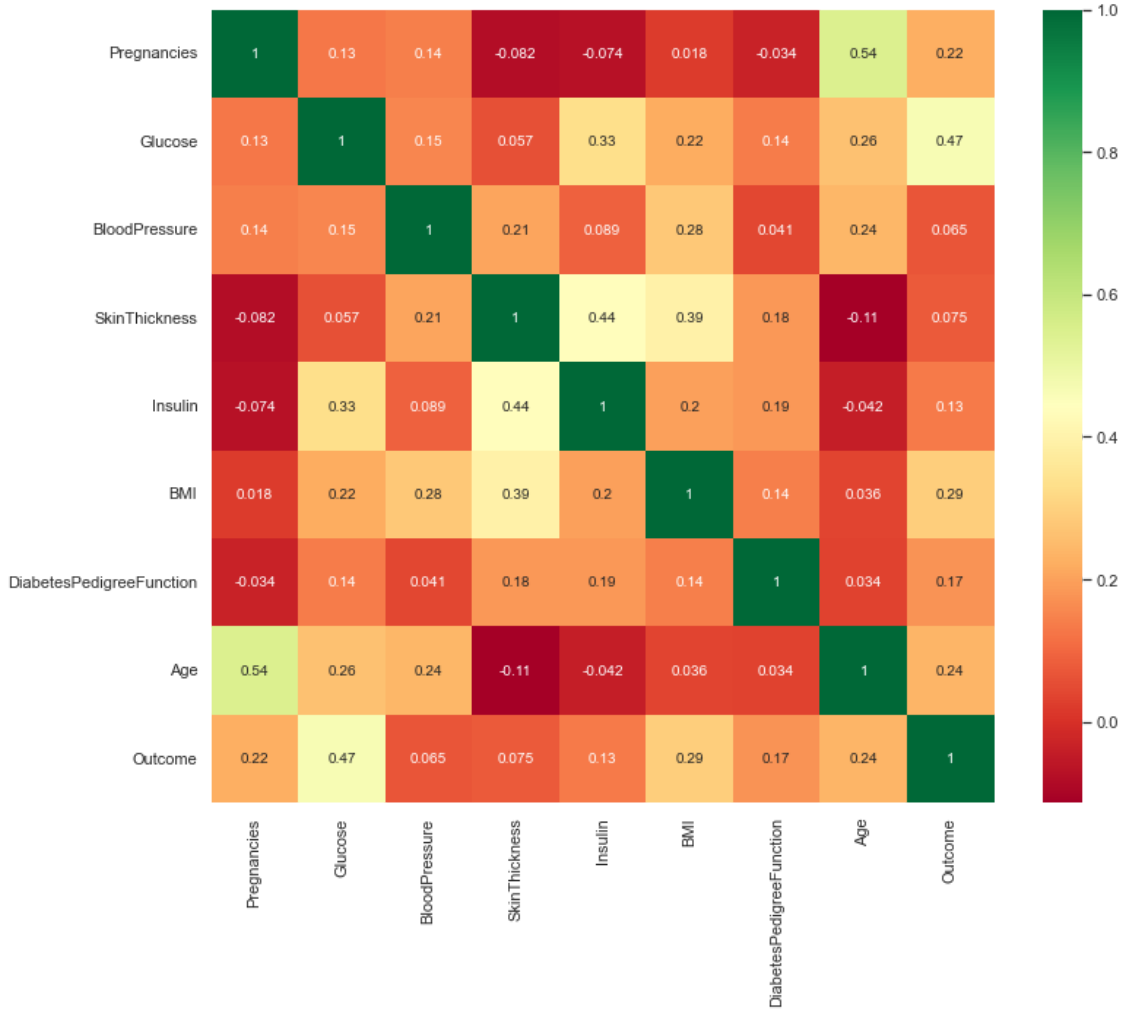
In [21]:

```
plt.subplot(121), sns.distplot(diabetes_df['Insulin'])
plt.subplot(122), diabetes_df['Insulin'].plot.box(figsize=(16,5))
plt.show()
```



In [22]:

```
plt.figure(figsize=(12,10))
# seaborn has an easy method to showcase heatmap
p = sns.heatmap(diabetes_df.corr(), annot=True,cmap ='RdYlGn')
```



In [23]:

```
diabetes_df_copy.head()
```

Out[23]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFun
0	6	148.0	72.0	35.0	125.0	33.6	
1	1	85.0	66.0	29.0	125.0	26.6	
2	8	183.0	64.0	29.0	125.0	23.3	
3	1	89.0	66.0	23.0	94.0	28.1	
4	0	137.0	40.0	35.0	168.0	43.1	

In [24]:

```
sc_X = StandardScaler()
X =  pd.DataFrame(sc_X.fit_transform(diabetes_df_copy.drop(["Outcome"],axis = 1),), colu
'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunctio
X.head()
```

Out[24]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPed
0	0.639947	0.865108	-0.033518	0.670643	-0.181541	0.166619	
1	-0.844885	-1.206162	-0.529859	-0.012301	-0.181541	-0.852200	
2	1.233880	2.015813	-0.695306	-0.012301	-0.181541	-1.332500	
3	-0.844885	-1.074652	-0.529859	-0.695245	-0.540642	-0.633881	
4	-1.141852	0.503458	-2.680669	0.670643	0.316566	1.549303	

In [29]:

```
X = diabetes_df.drop('Outcome', axis=1)
y = diabetes_df['Outcome']
```

In [30]:

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.33,
                                                    random_state=7)
```

In [31]:

```
from sklearn.ensemble import RandomForestClassifier

rfc = RandomForestClassifier(n_estimators=200)
rfc.fit(X_train, y_train)
rfc_train = rfc.predict(X_train)
from sklearn import metrics

print("Accuracy_Score =", format(metrics.accuracy_score(y_train, rfc_train)))
```

```
Accuracy_Score = 1.0
```

In [32]:

```
from sklearn import metrics

predictions = rfc.predict(X_test)
print("Accuracy_Score =", format(metrics.accuracy_score(y_test, predictions)))
```

```
Accuracy_Score = 0.7716535433070866
```

In [40]:

```
y_predict = rfc.predict([[0,118,84,47,230,45.8,0.551,31]])
print(y_predict)
if y_predict==1:
    print("Diabetic")
else:
    print("Non Diabetic")
```

```
[1]
Diabetic
```

In [ ]:

```
import pdfkit
config=pdfkit.configuration(wkhtmltopdf=r"C:\Users\DELL\Downloads\Diabetes_prediction (1
```