Submission

June 1, 2020

1 DATA SCIENCE CAPSTONE - Healthcare Project

Import Libraries

[1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

import warnings
warnings.filterwarnings('ignore')
```

[2]:
```
pwd
```

[2]:
```
'/home/labsuser/ds'
```

[3]:
```
# Importing Dataset

df = pd.read_csv('/home/labsuser/ds/health_care_diabetes.csv')
```

[4]:
```
df.head()
```

[4]:
```
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
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

   DiabetesPedigreeFunction  Age  Outcome
0                     0.627   50        1
1                     0.351   31        0
2                     0.672   32        1
3                     0.167   21        0
4                     2.288   33        1
```

1. Perform descriptive analysis. Understand the variables and their corresponding values. On the columns below, a value of zero does not make sense and thus indicates missing value:

- Glucose
- BloodPressure
- SkinThickness
- Insulin
- BMI

2. Visually explore these variables using histograms. Treat the missing values accordingly.
3. There are integer and float data type variables in this dataset. Create a count (frequency) plot describing the data types and the count of variables.

[5]:
```python
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
```

[6]:
```python
df.shape
```

[6]: (768, 9)

[7]:
```python
df.describe()
```

[7]:
```
       Pregnancies     Glucose  BloodPressure  SkinThickness     Insulin  \
count   768.000000  768.000000     768.000000     768.000000  768.000000
mean      3.845052  120.894531      69.105469      20.536458   79.799479
std       3.369578   31.972618      19.355807      15.952218  115.244002
min       0.000000    0.000000       0.000000       0.000000    0.000000
25%       1.000000   99.000000      62.000000       0.000000    0.000000
50%       3.000000  117.000000      72.000000      23.000000   30.500000
75%       6.000000  140.250000      80.000000      32.000000  127.250000
```

```
max      17.000000  199.000000     122.000000      99.000000  846.000000

              BMI  DiabetesPedigreeFunction         Age     Outcome
count  768.000000                768.000000  768.000000  768.000000
mean    31.992578                  0.471876   33.240885    0.348958
std      7.884160                  0.331329   11.760232    0.476951
min      0.000000                  0.078000   21.000000    0.000000
25%     27.300000                  0.243750   24.000000    0.000000
50%     32.000000                  0.372500   29.000000    0.000000
75%     36.600000                  0.626250   41.000000    1.000000
max     67.100000                  2.420000   81.000000    1.000000
```

- This Datasets have 9 variables and 768 Observations
- The dataset helps to predict the diabetes of various age group of women using the variables of pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin and BMI.
- The Average Age of Patients are 33.24 with minimum being 21 and maximum 81

[8]:
```
df.isnull().any()
```

[8]:
```
Pregnancies                 False
Glucose                     False
BloodPressure               False
SkinThickness               False
Insulin                     False
BMI                         False
DiabetesPedigreeFunction    False
Age                         False
Outcome                     False
dtype: bool
```

[9]:
```
df.isnull().sum()
```

[9]:
```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

[10]:
```
df.columns
```

[10]:
```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
```

```
      dtype='object')
```

[11]:
```
print((df[['Glucose']]==0).sum())
```

```
Glucose    5
dtype: int64
```

[12]:
```
print((df[['BloodPressure']]==0).sum())
```

```
BloodPressure    35
dtype: int64
```

[13]:
```
print((df[['SkinThickness']]==0).sum())
```

```
SkinThickness    227
dtype: int64
```

[14]:
```
print((df[['Insulin']]==0).sum())
```

```
Insulin    374
dtype: int64
```

[15]:
```
print((df[['BMI']]==0).sum())
```

```
BMI    11
dtype: int64
```

[16]:
```
print ((df[['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness',␣
 ↪'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age']] == 0).sum())
```

```
Pregnancies                 111
Glucose                       5
BloodPressure                35
SkinThickness               227
Insulin                     374
BMI                          11
DiabetesPedigreeFunction      0
Age                           0
dtype: int64
```

[17]:
```
print((df[['Glucose']]==0).count())
```

```
Glucose    768
dtype: int64
```

[18]:
```
print ((df[['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness',␣
 ↪'Insulin',
```

```
        'BMI', 'DiabetesPedigreeFunction', 'Age']] == 0).count())
```

```
Pregnancies                 768
Glucose                     768
BloodPressure               768
SkinThickness               768
Insulin                     768
BMI                         768
DiabetesPedigreeFunction    768
Age                         768
dtype: int64
```

[19]:
```
df.head()
```

[19]:
```
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
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

   DiabetesPedigreeFunction  Age  Outcome
0                     0.627   50        1
1                     0.351   31        0
2                     0.672   32        1
3                     0.167   21        0
4                     2.288   33        1
```

[20]:
```
df [['Pregnancies','Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', ]] = df [['Pregnancies','Glucose', 'BloodPressure',␣
 ↪'SkinThickness', 'Insulin',
       'BMI', ]].replace(0,np.NaN)
```

[21]:
```
df.head()
```

[21]:
```
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0          6.0    148.0           72.0           35.0      NaN  33.6
1          1.0     85.0           66.0           29.0      NaN  26.6
2          8.0    183.0           64.0            NaN      NaN  23.3
3          1.0     89.0           66.0           23.0     94.0  28.1
4          NaN    137.0           40.0           35.0    168.0  43.1

   DiabetesPedigreeFunction  Age  Outcome
0                     0.627   50        1
1                     0.351   31        0
2                     0.672   32        1
3                     0.167   21        0
```

```
4                      2.288   33         1
```

[22]:
```
df['Pregnancies'].fillna(df['Pregnancies'].mean(), inplace = True)
```

[23]:
```
print(df['Pregnancies'].isnull().sum())
```

```
0
```

[24]:
```
df.fillna(df.mean(), inplace=True)
```

[25]:
```
df.head()
```

[25]:
```
   Pregnancies  Glucose  BloodPressure  SkinThickness     Insulin   BMI  \
0     6.000000    148.0           72.0       35.00000  155.548223  33.6
1     1.000000     85.0           66.0       29.00000  155.548223  26.6
2     8.000000    183.0           64.0       29.15342  155.548223  23.3
3     1.000000     89.0           66.0       23.00000   94.000000  28.1
4     4.494673    137.0           40.0       35.00000  168.000000  43.1

   DiabetesPedigreeFunction  Age  Outcome
0                     0.627   50        1
1                     0.351   31        0
2                     0.672   32        1
3                     0.167   21        0
4                     2.288   33        1
```

[26]:
```
print (df[['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness',␣
 ↪'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age']].isnull().sum())
```

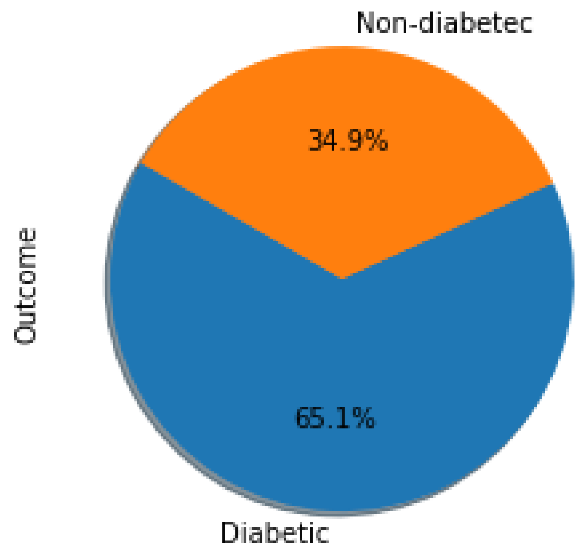
```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
dtype: int64
```

[27]:
```
df.groupby('Outcome').size()
```

[27]:
```
Outcome
0    500
1    268
dtype: int64
```

[28]:
```
labels = 'Diabetic', 'Non-diabetec'
df.Outcome.value_counts().plot.pie(labels=labels, autopct='%1.
↪1f%%',shadow=True, startangle=150)
```

[28]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f999928b450>
```



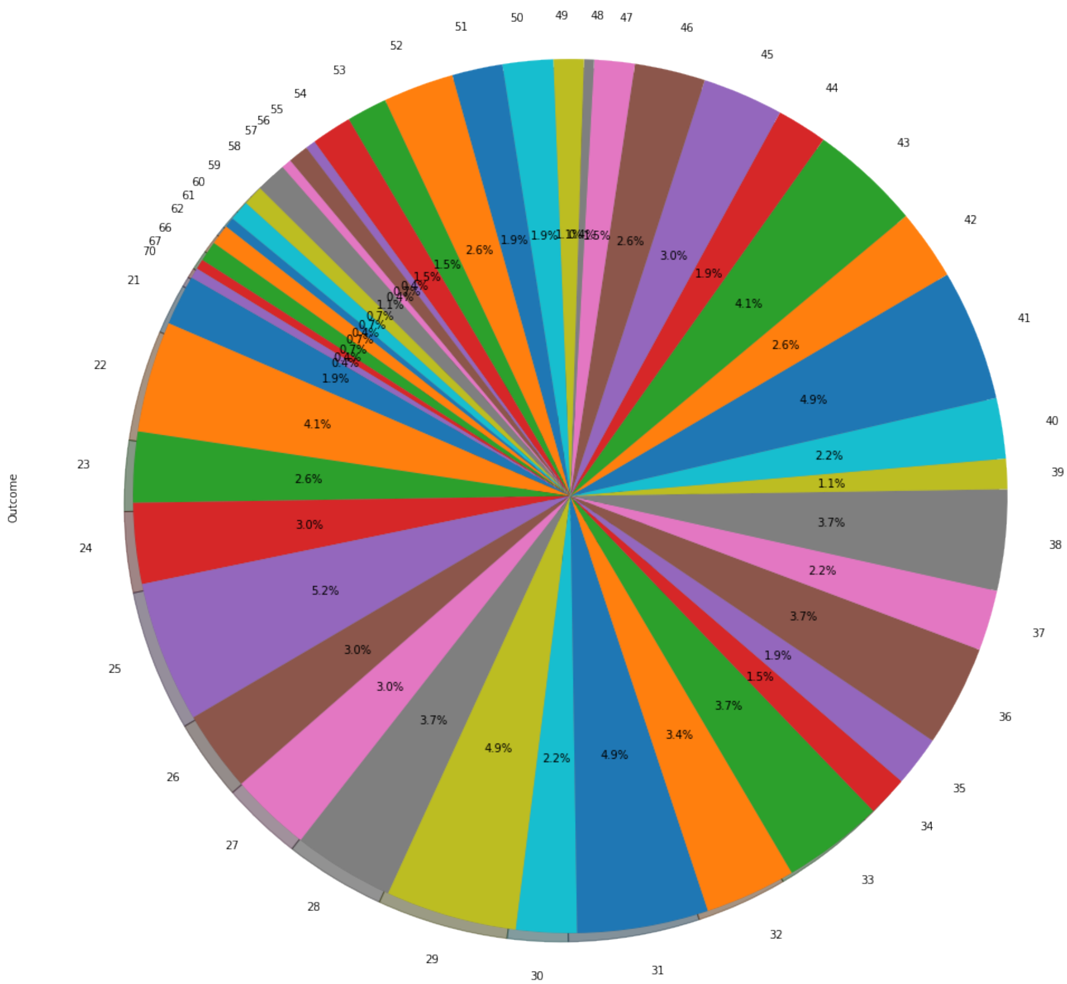
[29]:
```
diabetes_agewise = df[df['Outcome']==1]
diabetes_agewise.groupby('Age')['Outcome'].count()
```

[29]:
```
Age
21     5
22    11
23     7
24     8
25    14
26     8
27     8
28    10
29    13
30     6
31    13
32     9
33    10
34     4
35     5
36    10
37     6
```

```
38    10
39     3
40     6
41    13
42     7
43    11
44     5
45     8
46     7
47     4
48     1
49     3
50     5
51     5
52     7
53     4
54     4
55     1
56     2
57     1
58     3
59     2
60     2
61     1
62     2
66     2
67     1
70     1
Name: Outcome, dtype: int64
```

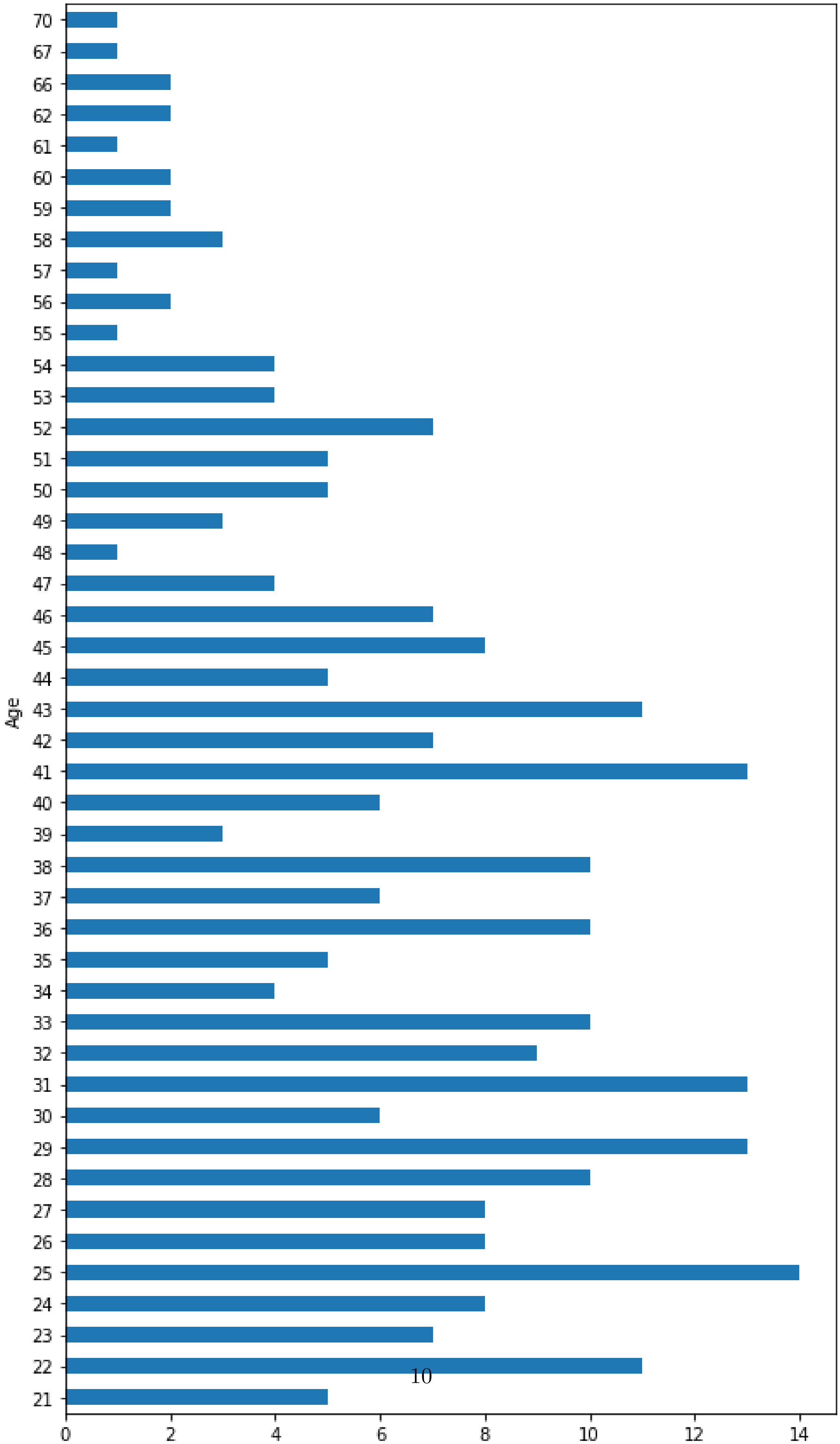
[30]:
```
diabetes_agewise.groupby('Age')['Outcome'].count().plot.pie(autopct='%1.
↪1f%%',shadow=True, startangle=150, figsize=(35,18))
```

[30]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f998fcda610>
```



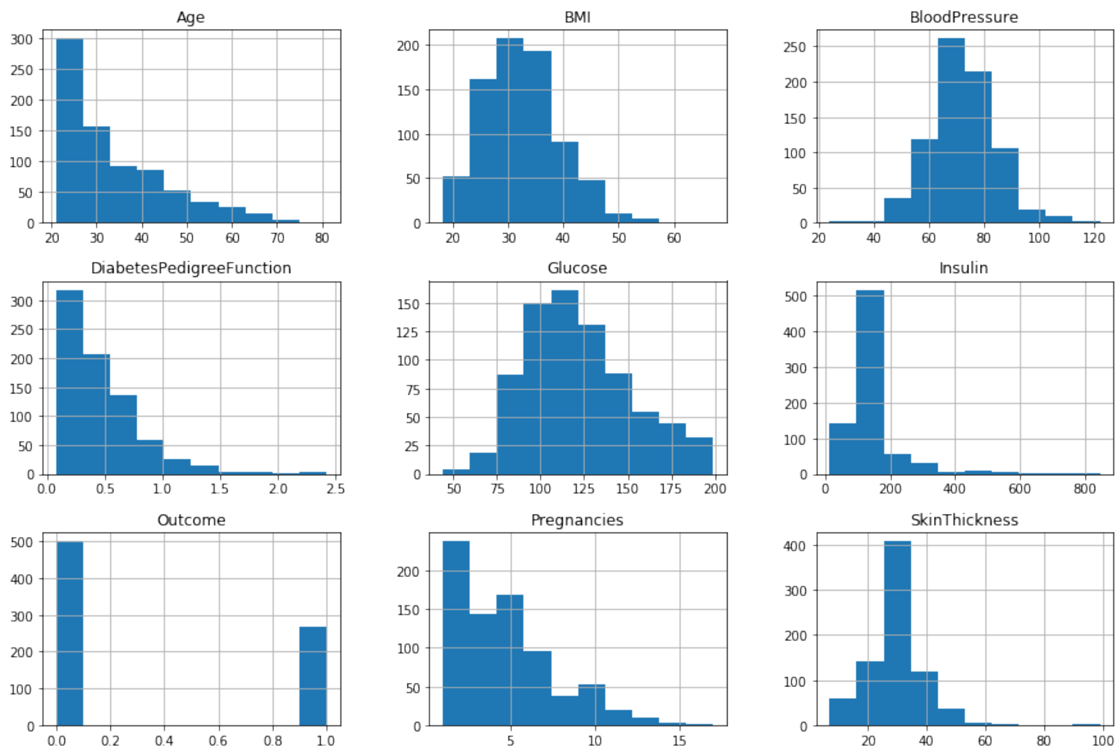
[31]:
```
diabetes_agewise.groupby('Age')['Outcome'].count().plot(kind= 'barh',
↪figsize=(8,15))
```

[31]: <matplotlib.axes._subplots.AxesSubplot at 0x7f998e9a2e50>


Age
70
67
66
62
61
60
59
58
57
56
55
54
53
52
51
50
49
48
47
46
45
44
43
42
41
40
39
38
37
36
35
34
33
32
31
30
29
28
27
26
25
24
23
22
21
0
2
4
6
8
10
12
14

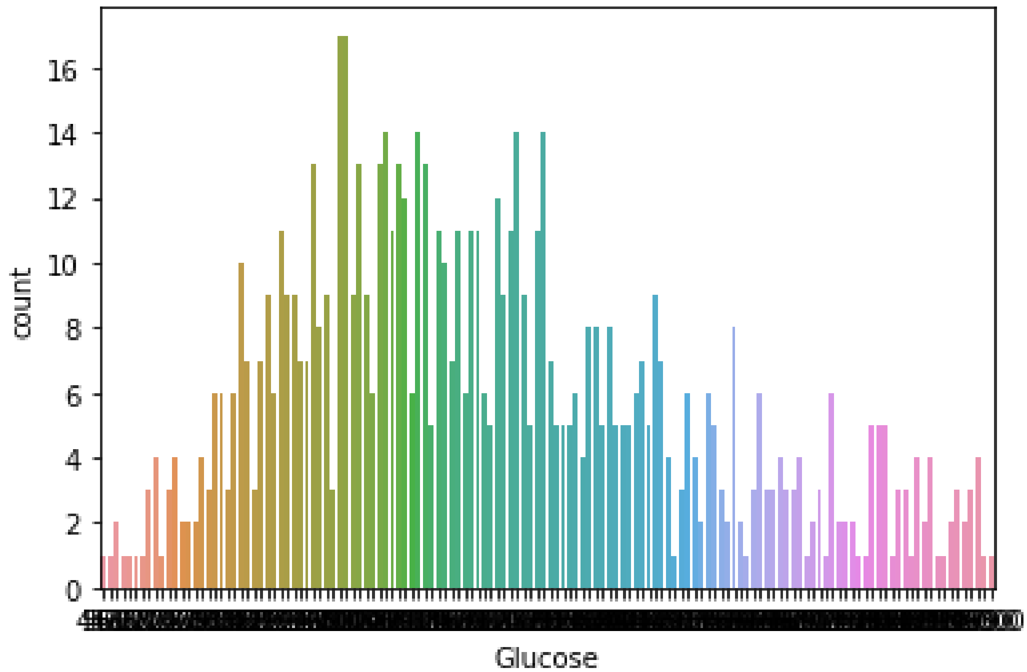
[32]:
```
df.hist(figsize=(15,10))
```

[32]:
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f998ed60510>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f998ed86bd0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f998ed48bd0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f998ecff850>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f998ecbbbd0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f998ec73850>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f998edb4cd0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f998ec45510>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f998ec4f090>]],
      dtype=object)
```



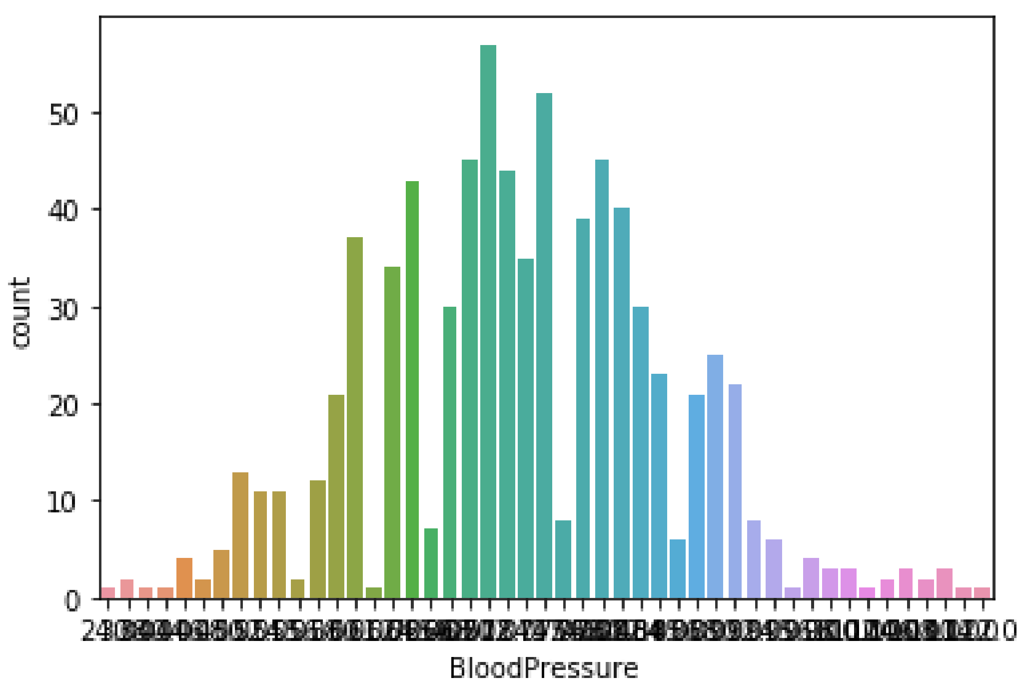
[33]:
```
sns.countplot(df['Glucose'])
```

[33]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f998d427fd0>
```



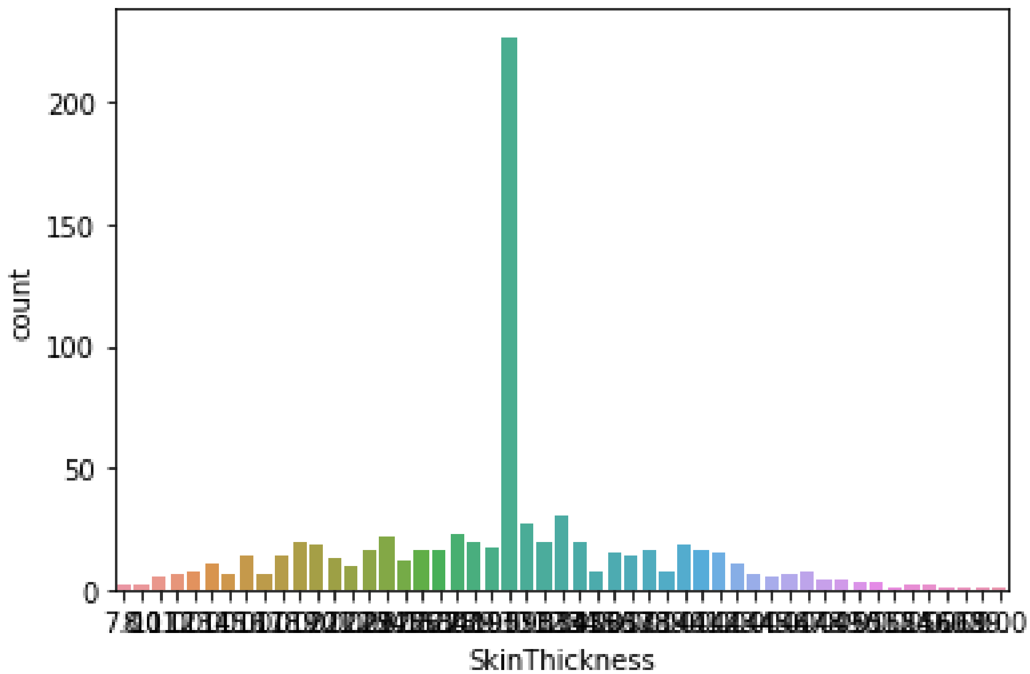
[34]:
```
sns.countplot(df['BloodPressure'])
```

[34]: <matplotlib.axes._subplots.AxesSubplot at 0x7f9998b25450>



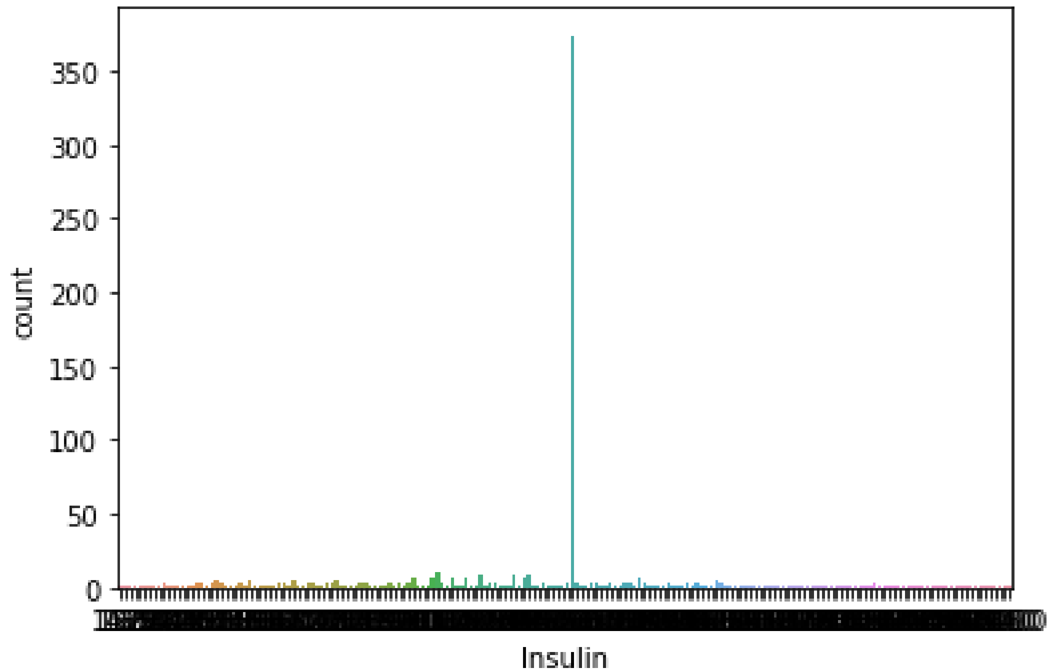
[35]:
```
sns.countplot(df['SkinThickness'])
```

[35]: <matplotlib.axes._subplots.AxesSubplot at 0x7f998cf61350>



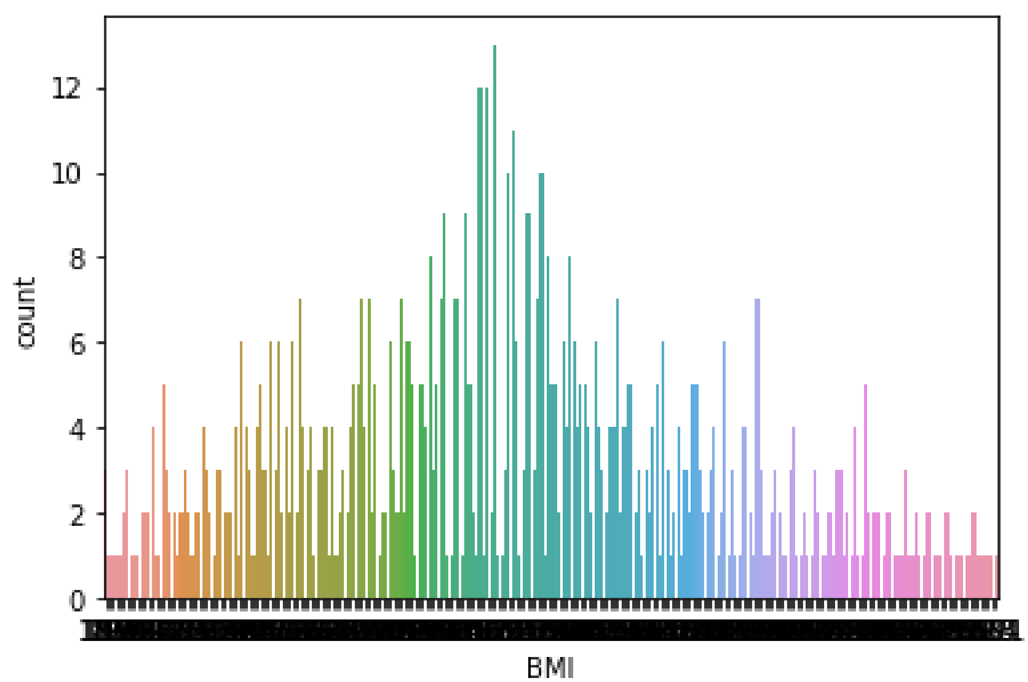
[36]:
```
sns.countplot(df['Insulin'])
```

[36]: <matplotlib.axes._subplots.AxesSubplot at 0x7f998cdb2090>



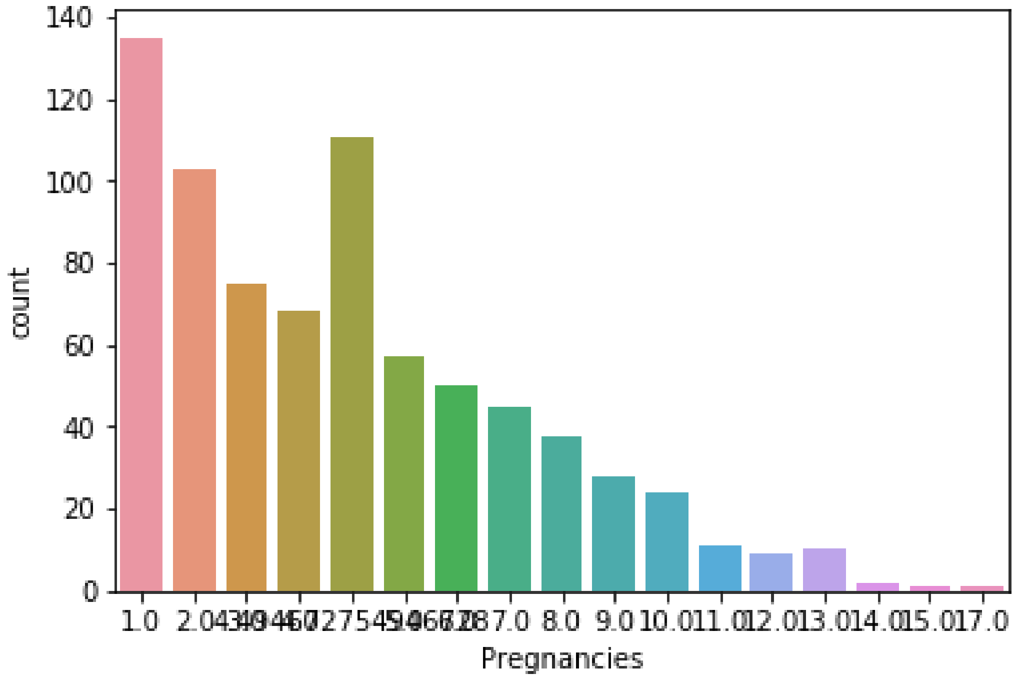
[37]:
```
sns.countplot(df['BMI'])
```

[37]: <matplotlib.axes._subplots.AxesSubplot at 0x7f998c906cd0>



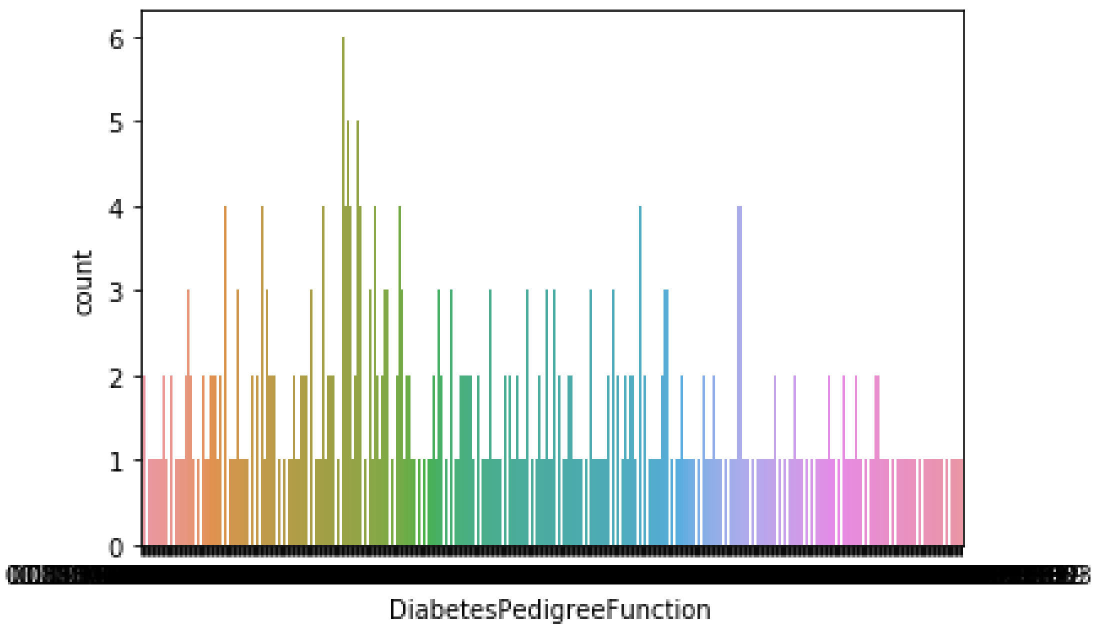
```
[38]: sns.countplot(df['Pregnancies'])
```

```
[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f998cdac250>
```



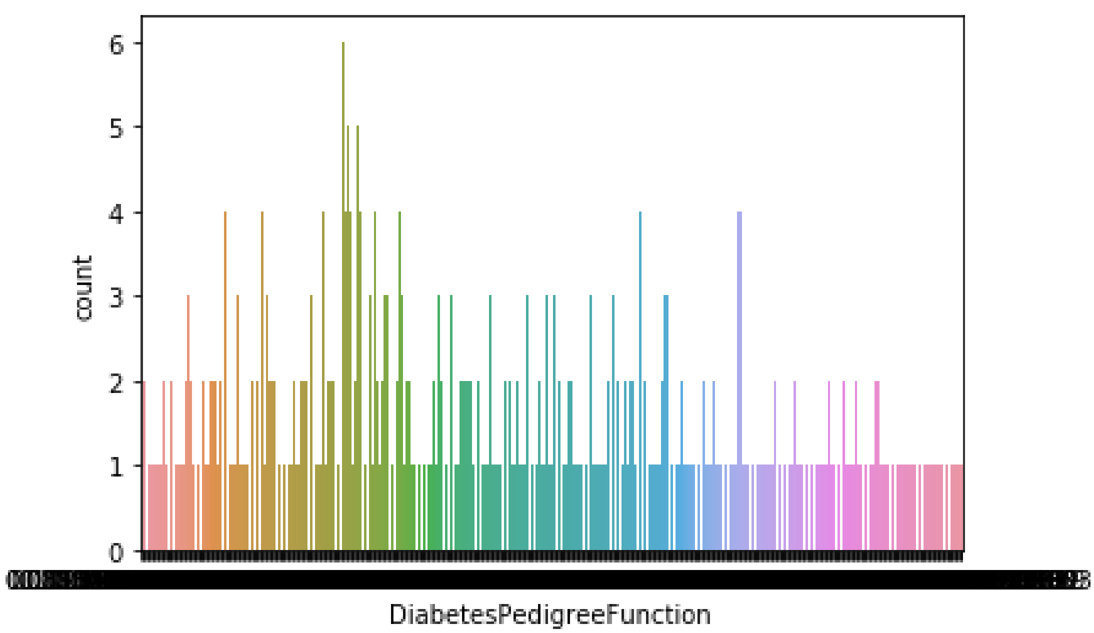

```
[39]: sns.countplot(df['DiabetesPedigreeFunction'])
```

```
[39]: <matplotlib.axes._subplots.AxesSubplot at 0x7f998c115410>
```



[40]:
```
sns.countplot(df['DiabetesPedigreeFunction'])
```

[40]: <matplotlib.axes._subplots.AxesSubplot at 0x7f9987504ed0>

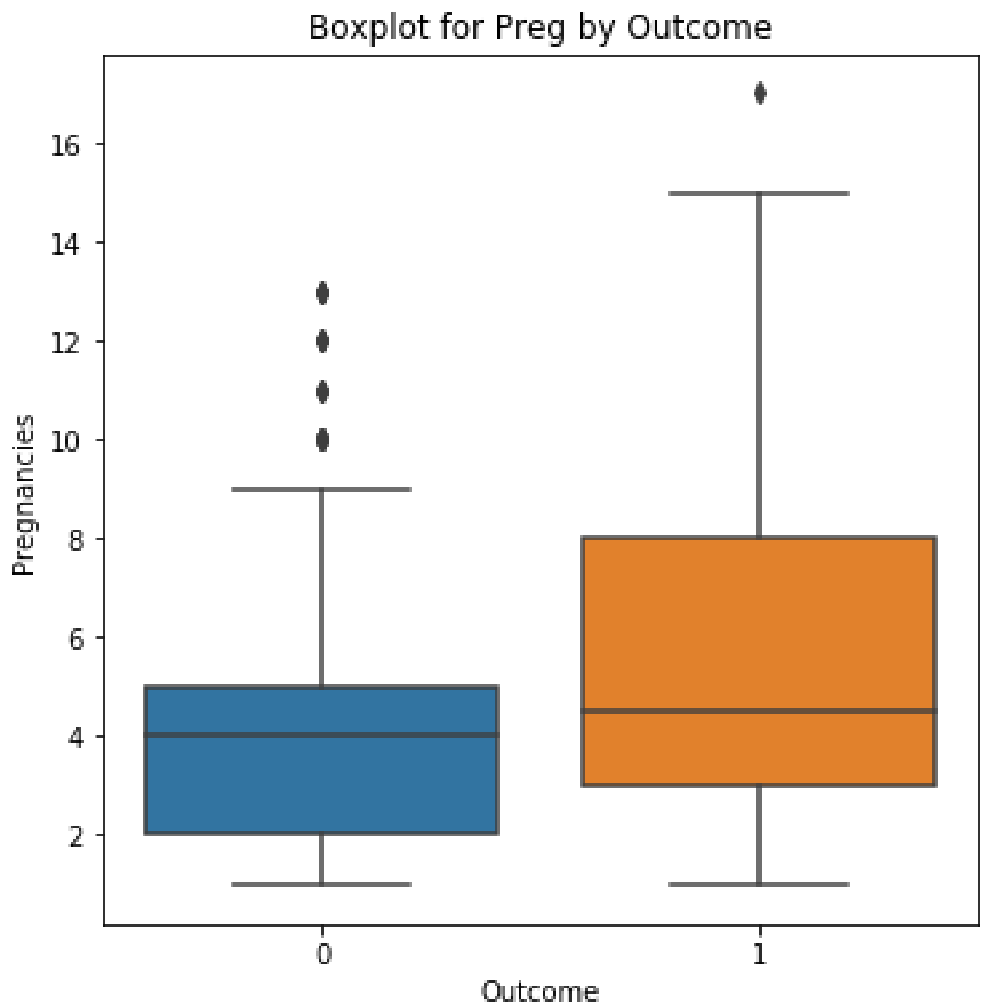


1. Check the balance of the data by plotting the count of outcomes by their value. Describe your findings and plan future course of action.

2. Create scatter charts between the pair of variables to understand the relationships. Describe your findings.
3. Perform correlation analysis. Visually explore it using a heat map.M

[41]:
```
# Plots for count of outcome by values
plt.figure(figsize=(20, 6))
plt.subplot(1,3,3)
sns.boxplot(x=df.Outcome,y=df.Pregnancies)
plt.title("Boxplot for Preg by Outcome")
```

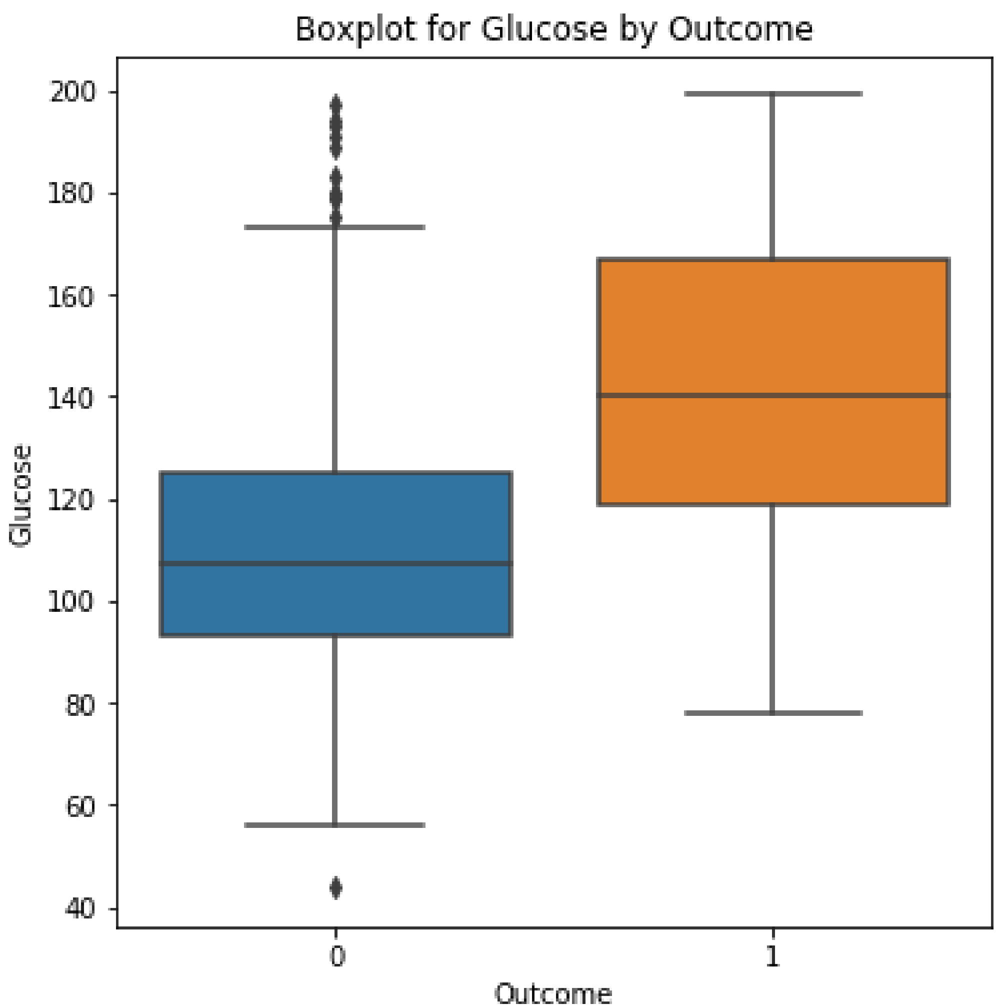
[41]:
```
Text(0.5, 1.0, 'Boxplot for Preg by Outcome')
```



[42]:
```
# Plot for glucose
plt.figure(figsize=(20, 6))
plt.subplot(1,3,3)
```

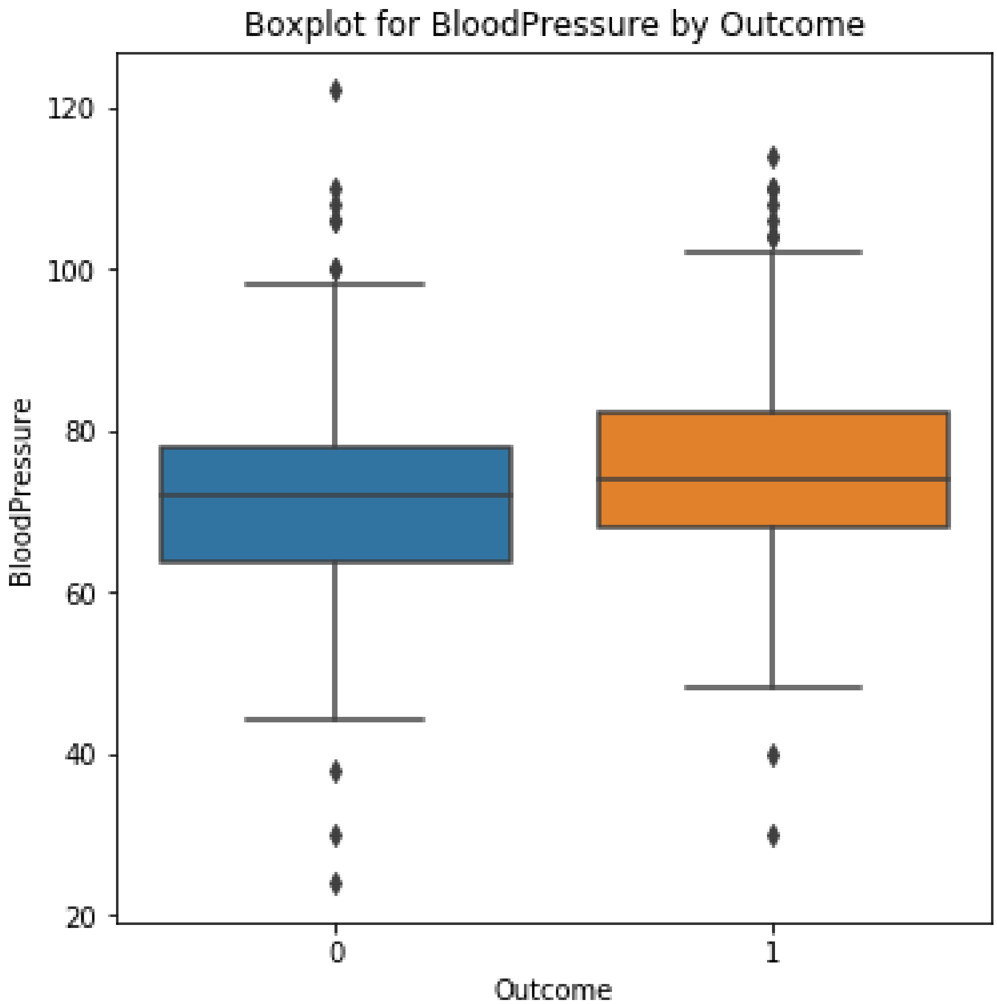
```
sns.boxplot(x=df.Outcome,y=df.Glucose)
plt.title("Boxplot for Glucose by Outcome")
```

[42]: Text(0.5, 1.0, 'Boxplot for Glucose by Outcome')



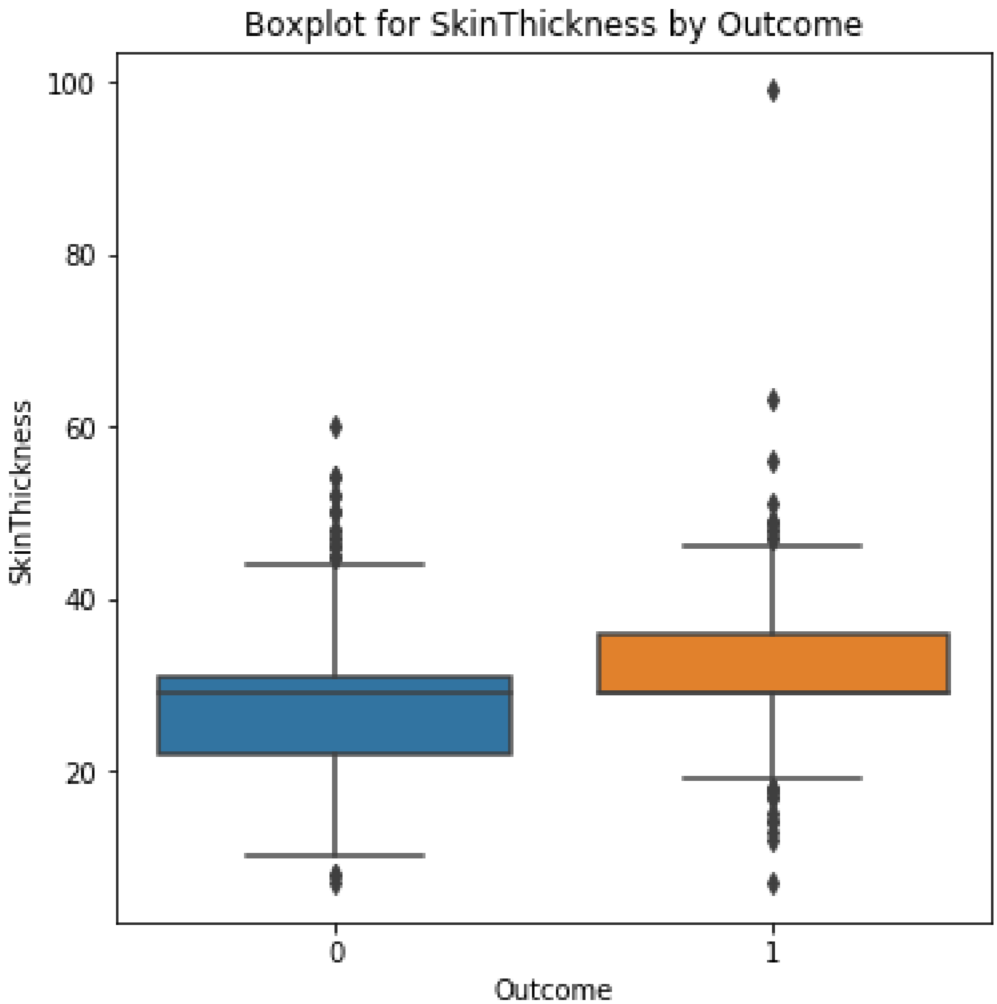
[43]:
```
# Plot for BloodPressure
plt.figure(figsize=(20, 6))
plt.subplot(1,3,3)
sns.boxplot(x=df.Outcome,y=df.BloodPressure)
plt.title("Boxplot for BloodPressure by Outcome")
```

[43]: Text(0.5, 1.0, 'Boxplot for BloodPressure by Outcome')



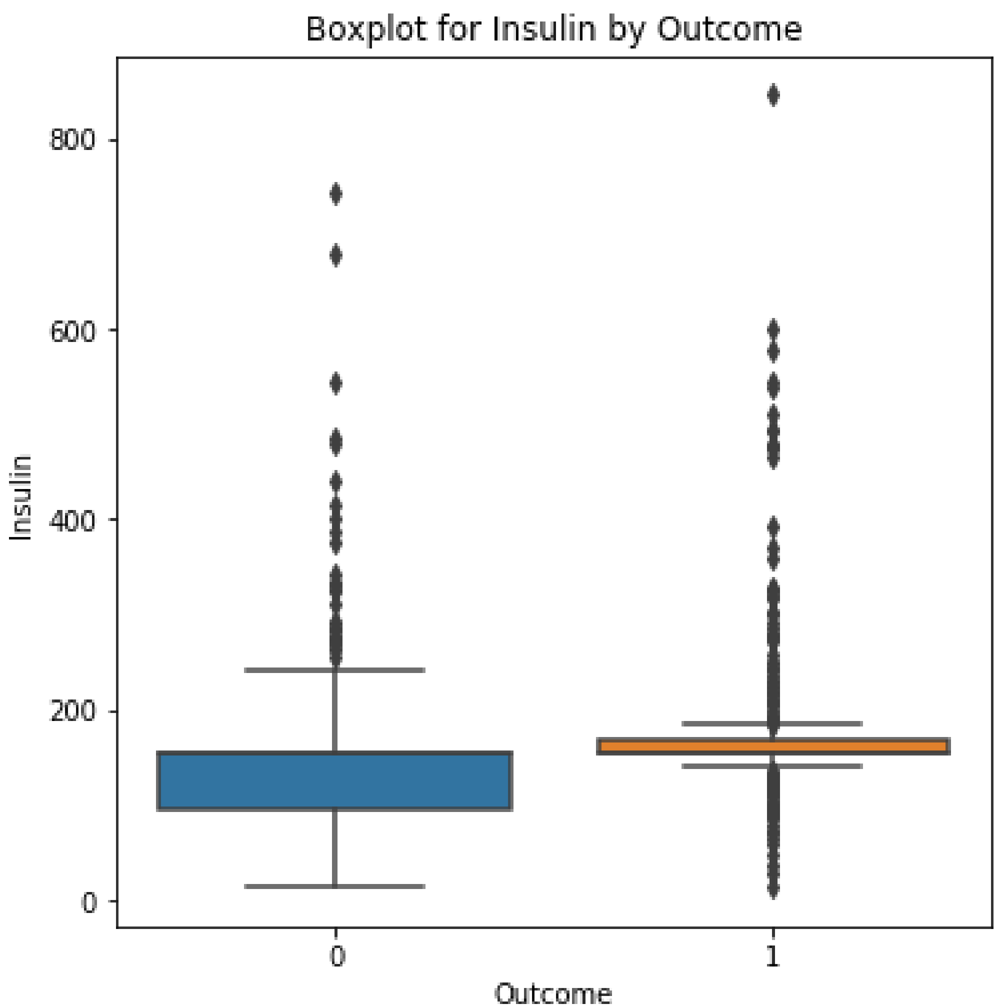
[44]:
```python
# Plot for SkinThickness
plt.figure(figsize=(20, 6))
plt.subplot(1,3,3)
sns.boxplot(x=df.Outcome,y=df.SkinThickness)
plt.title("Boxplot for SkinThickness by Outcome")
```

[44]: Text(0.5, 1.0, 'Boxplot for SkinThickness by Outcome')



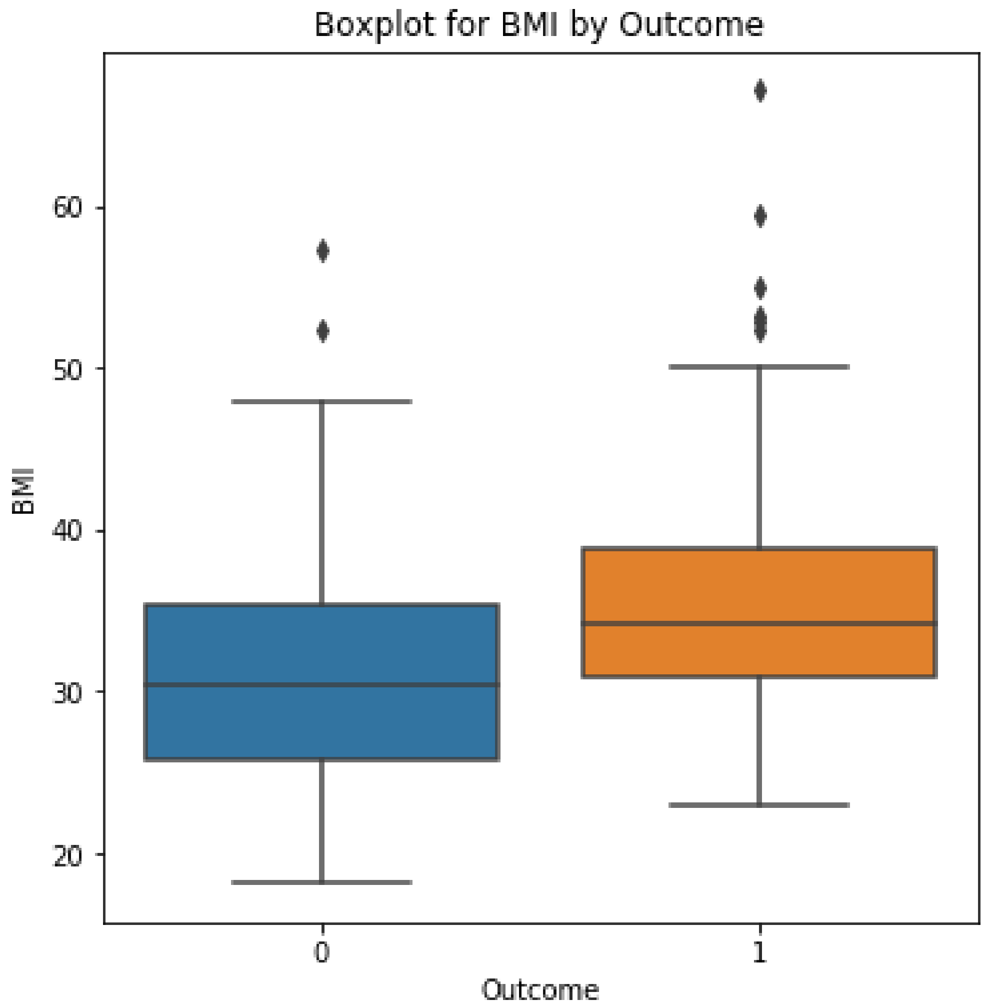

```
[45]: # plot for Insulin
      plt.figure(figsize=(20, 6))
      plt.subplot(1,3,3)
      sns.boxplot(x=df.Outcome,y=df.Insulin)
      plt.title("Boxplot for Insulin by Outcome")
```

```
[45]: Text(0.5, 1.0, 'Boxplot for Insulin by Outcome')
```



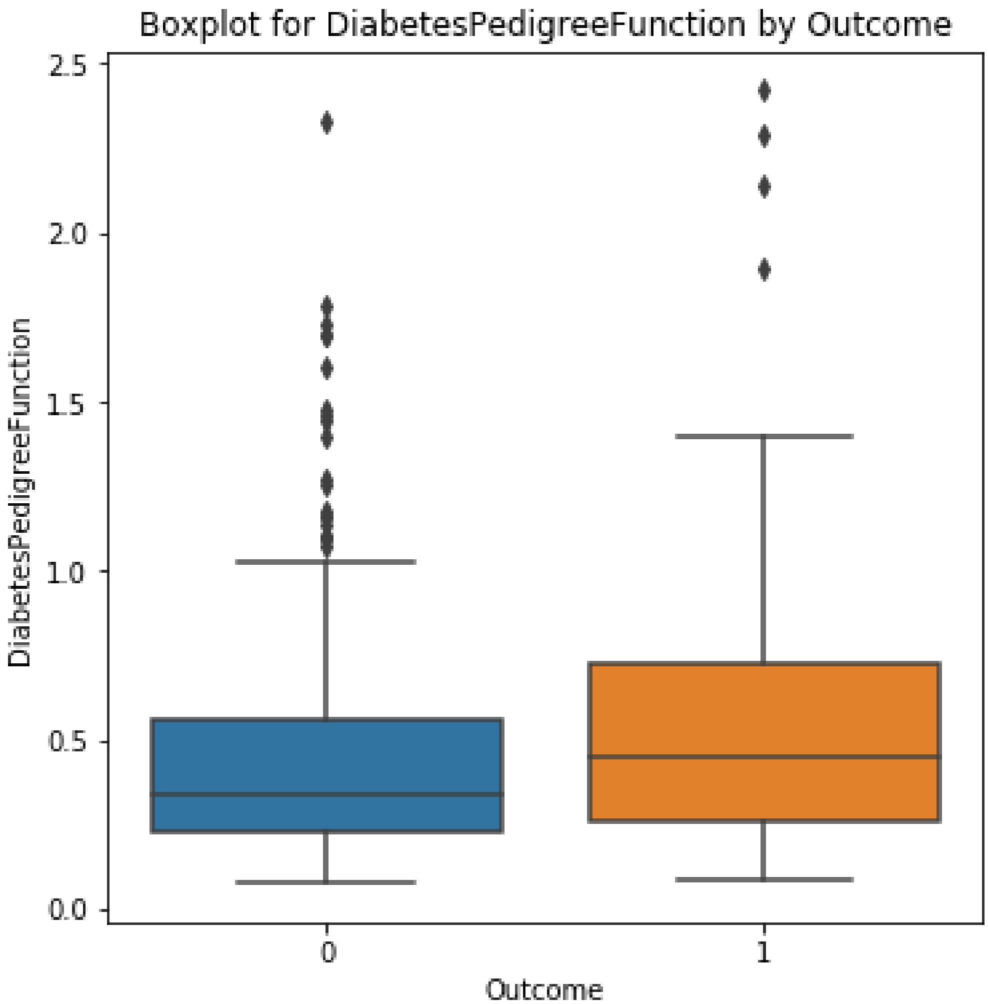
[46]:
```
# Plot for BMI
plt.figure(figsize=(20, 6))
plt.subplot(1,3,3)
sns.boxplot(x=df.Outcome,y=df.BMI)
plt.title("Boxplot for BMI by Outcome")
```

[46]: Text(0.5, 1.0, 'Boxplot for BMI by Outcome')



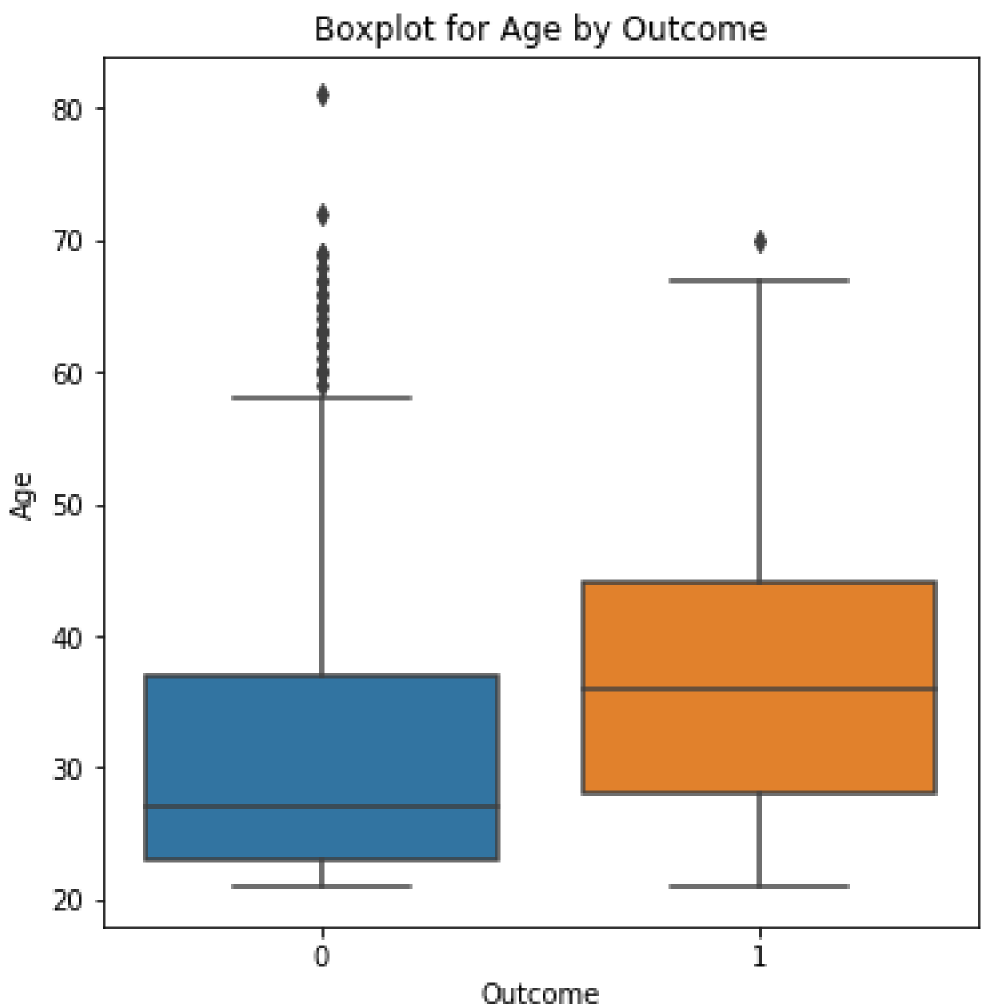
[47]:
```
# Plot for Diabetes Pedigree Function
plt.figure(figsize=(20, 6))
plt.subplot(1,3,3)
sns.boxplot(x=df.Outcome,y=df.DiabetesPedigreeFunction)
plt.title("Boxplot for DiabetesPedigreeFunction by Outcome")
```

[47]:
```
Text(0.5, 1.0, 'Boxplot for DiabetesPedigreeFunction by Outcome')
```



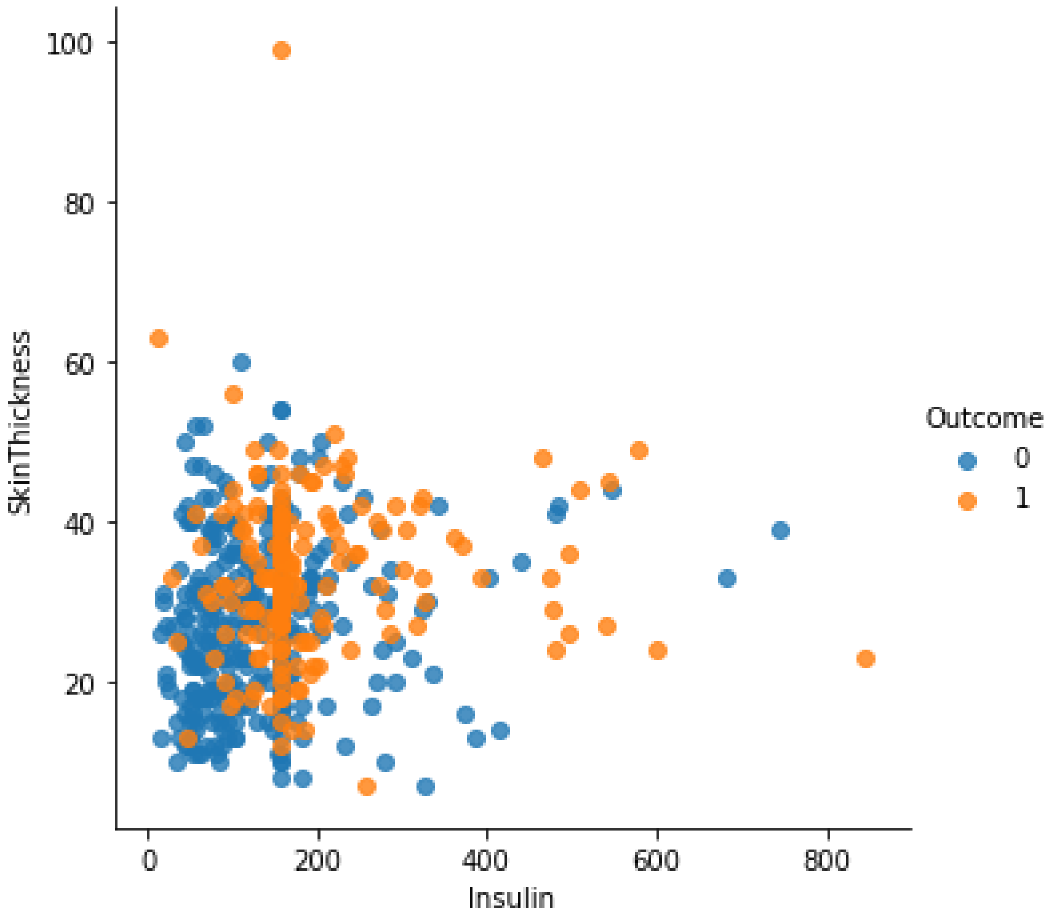
[48]:
```
# Plot for Age
plt.figure(figsize=(20, 6))
plt.subplot(1,3,3)
sns.boxplot(x=df.Outcome,y=df.Age)
plt.title("Boxplot for Age by Outcome")
```

[48]: Text(0.5, 1.0, 'Boxplot for Age by Outcome')



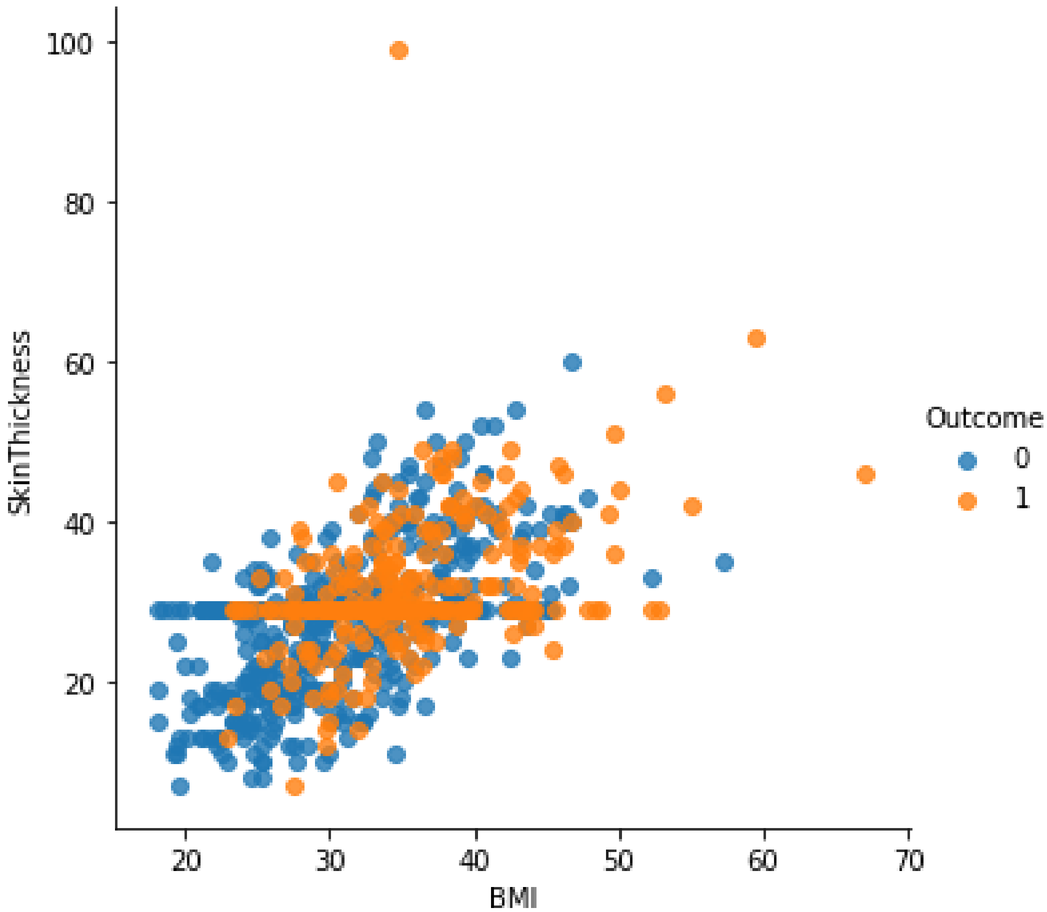
[49]:
```
# Plot with outcome and variables
sns.lmplot(x='Insulin',y='SkinThickness',data=df,fit_reg=False,hue='Outcome')
```

[49]: <seaborn.axisgrid.FacetGrid at 0x7f9986690990>



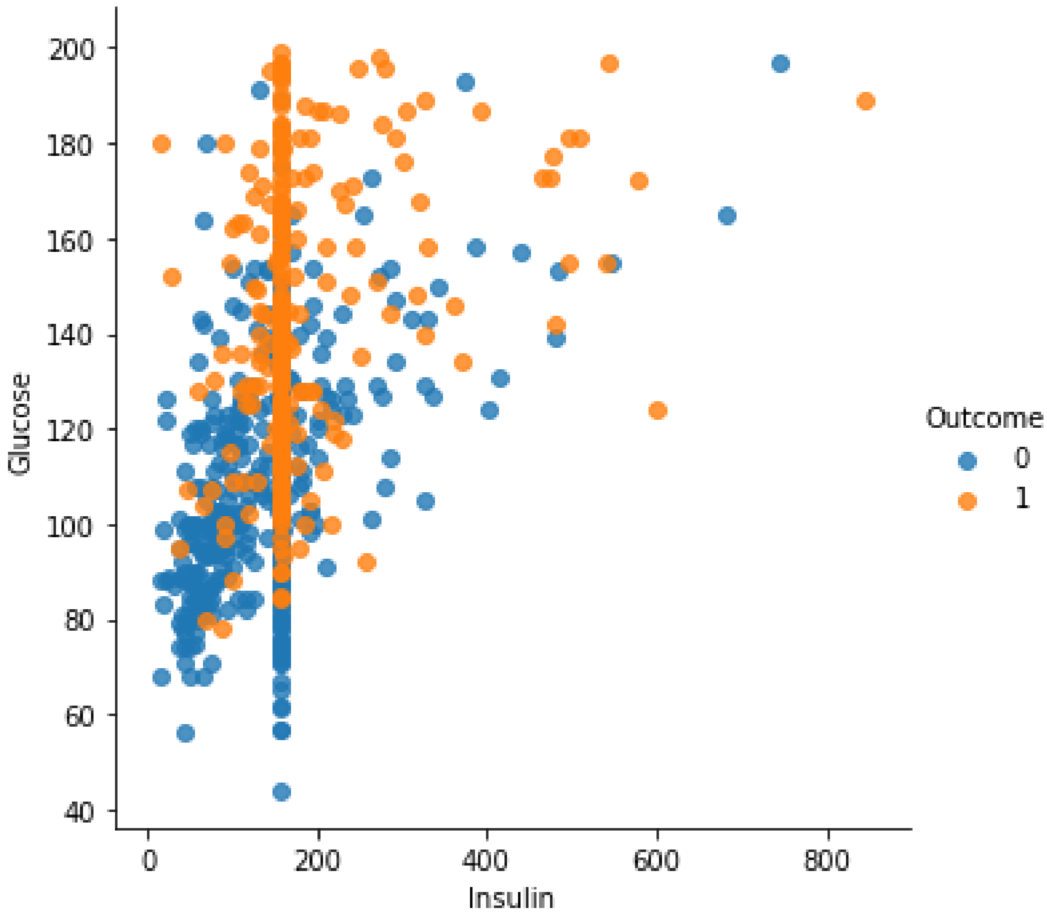

```
[50]: sns.lmplot(x='BMI',y='SkinThickness',data=df,fit_reg=False,hue='Outcome')
```

```
[50]: <seaborn.axisgrid.FacetGrid at 0x7f9986642c50>
```



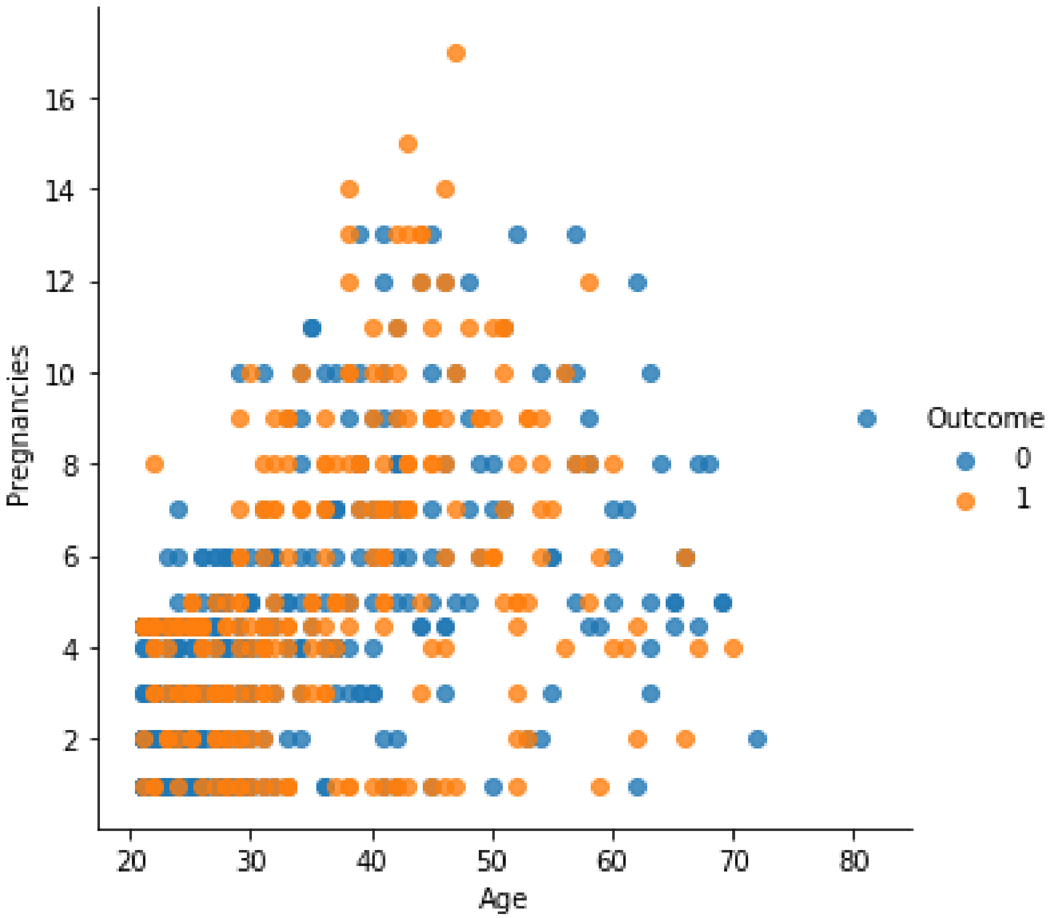
[51]:
```
sns.lmplot(x='Insulin',y='Glucose',data=df,fit_reg=False,hue='Outcome')
```

[51]: <seaborn.axisgrid.FacetGrid at 0x7f998659a990>



[52]:
```
sns.lmplot(x='Age',y='Pregnancies',data=df,fit_reg=False,hue='Outcome')
```

[52]: <seaborn.axisgrid.FacetGrid at 0x7f9986561810>




```
[53]: sns.pairplot(df, vars=["Pregnancies",␣
       ↪"Glucose","BloodPressure","SkinThickness","Insulin",␣
       ↪"BMI","DiabetesPedigreeFunction", "Age"],hue="Outcome")
      plt.title("Pairplot of Variables by Outcome")
```

```
[53]: Text(0.5, 1, 'Pairplot of Variables by Outcome')
```

[54]:
```
cor = df.corr()
cor
```

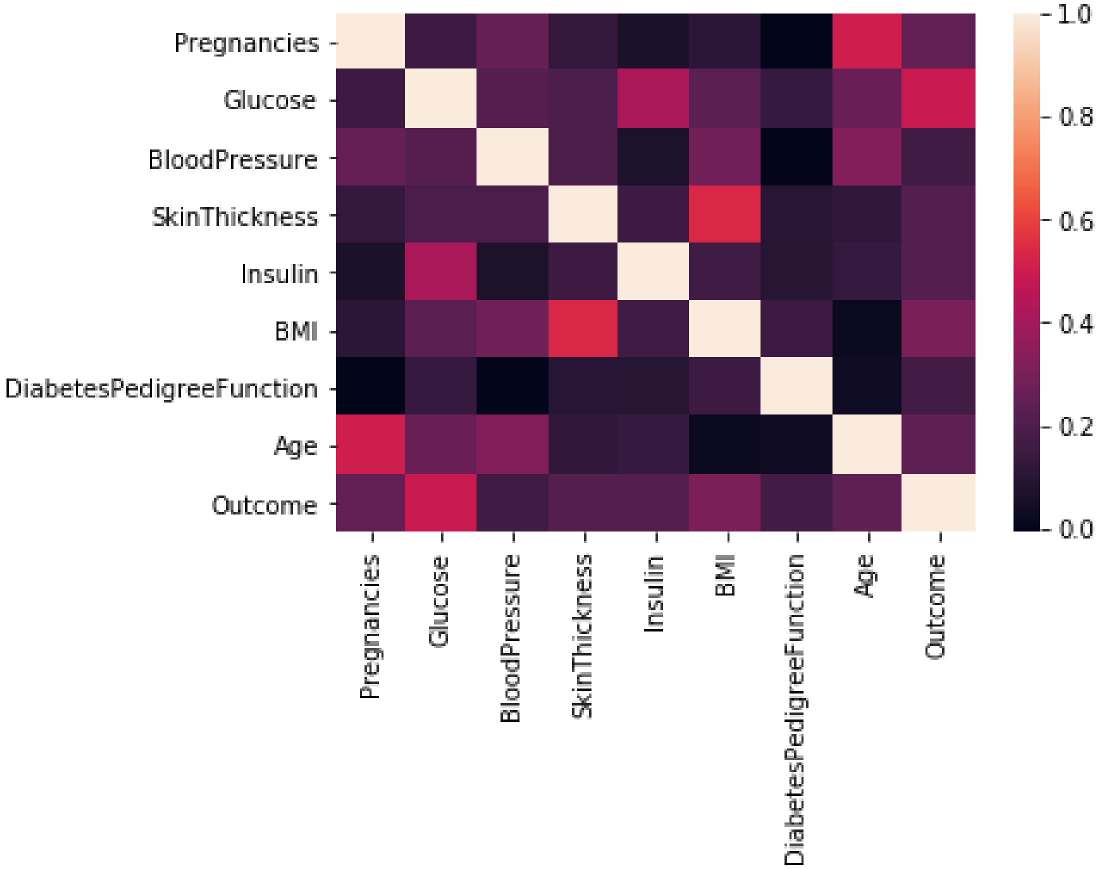
[54]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	\
Pregnancies	1.000000	0.154290	0.259117	0.131819	
Glucose	0.154290	1.000000	0.218367	0.192991	
BloodPressure	0.259117	0.218367	1.000000	0.192816	
SkinThickness	0.131819	0.192991	0.192816	1.000000	
Insulin	0.068077	0.420157	0.072517	0.158139	
BMI	0.110590	0.230941	0.281268	0.542398	
DiabetesPedigreeFunction	-0.005658	0.137060	-0.002763	0.100966	
Age	0.511662	0.266534	0.324595	0.127872	
Outcome	0.248263	0.492928	0.166074	0.215299	

```
                             Insulin       BMI  DiabetesPedigreeFunction  \
Pregnancies                 0.068077  0.110590                 -0.005658
Glucose                     0.420157  0.230941                  0.137060
BloodPressure               0.072517  0.281268                 -0.002763
SkinThickness               0.158139  0.542398                  0.100966
Insulin                     1.000000  0.166586                  0.098634
BMI                         0.166586  1.000000                  0.153400
DiabetesPedigreeFunction    0.098634  0.153400                  1.000000
Age                         0.136734  0.025519                  0.033561
Outcome                     0.214411  0.311924                  0.173844

                                 Age   Outcome
Pregnancies                 0.511662  0.248263
Glucose                     0.266534  0.492928
BloodPressure               0.324595  0.166074
SkinThickness               0.127872  0.215299
Insulin                     0.136734  0.214411
BMI                         0.025519  0.311924
DiabetesPedigreeFunction    0.033561  0.173844
Age                         1.000000  0.238356
Outcome                     0.238356  1.000000
```

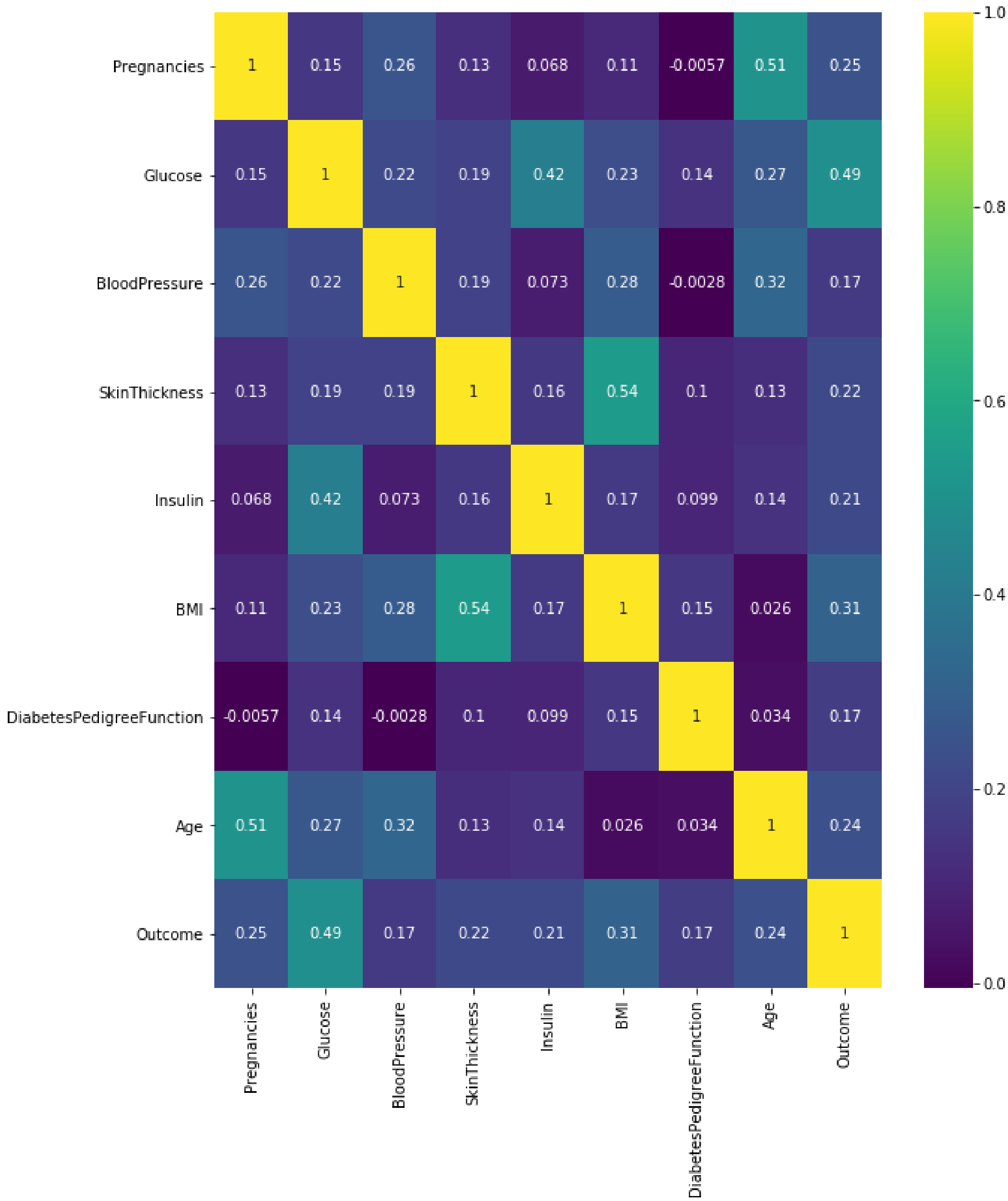
[55]:
```
sns.heatmap(cor)
```

[55]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f99849e9ad0>
```



[56]:
```
plt.subplots(figsize=(10,12))
sns.heatmap(cor,annot=True,cmap='viridis')
```

[56]: <matplotlib.axes._subplots.AxesSubplot at 0x7f9984951950>



1. Devise strategies for model building. It is important to decide the right validation framework. Express your thought process.
2. Apply an appropriate classification algorithm to build a model. Compare various models with the results from KNN algorithm.

[58]:
```
features = df.iloc[:,[0,1,2,3,4,5,6,7]].values
label = df.iloc[:,8].values
```

[59]:
```
#Train test split
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(features,
                                                 label,
                                                 test_size=0.2,
                                                 random_state =10)
```

[60]:
```
#Create model
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(X_train,y_train)
```

[60]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

[61]:
```
print(model.score(X_train,y_train))
print(model.score(X_test,y_test))
```

```
0.7833876221498371
0.7337662337662337
```

[62]:
```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(label,model.predict(features))
cm
```

[62]:
```
array([[448,  52],
       [122, 146]])
```

[63]:
```
from sklearn.metrics import classification_report
print(classification_report(label,model.predict(features)))
```

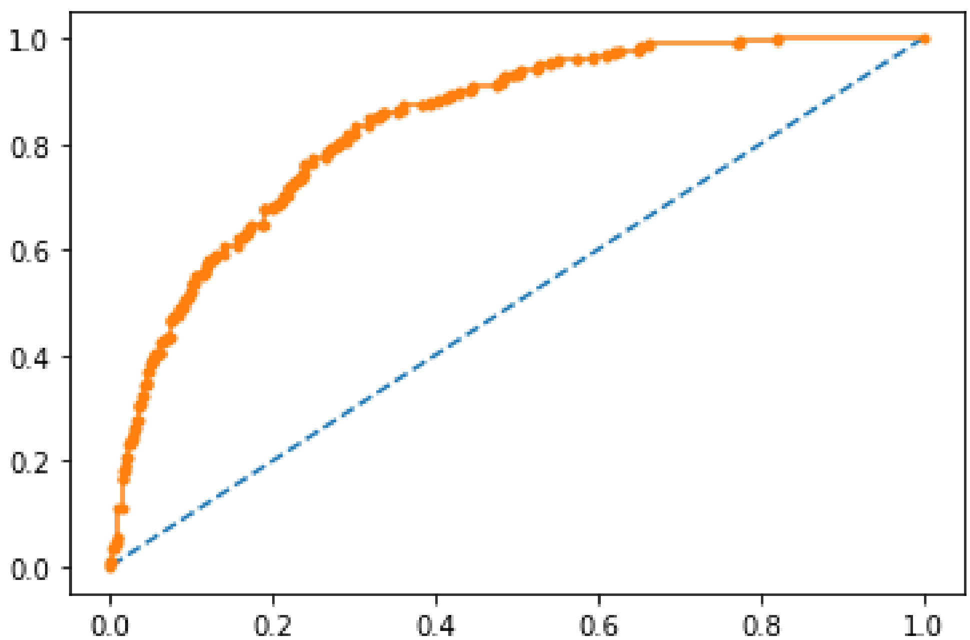
```
              precision    recall  f1-score   support

           0       0.79      0.90      0.84       500
           1       0.74      0.54      0.63       268

    accuracy                           0.77       768
   macro avg       0.76      0.72      0.73       768
weighted avg       0.77      0.77      0.76       768
```

[64]:
```
#Preparing ROC Curve (Receiver Operating Characteristics Curve)
from sklearn.metrics import roc_curve
from sklearn.metrics import roc_auc_score
```

```
# predict probabilities
probs = model.predict_proba(features)
# keep probabilities for the positive outcome only
probs = probs[:, 1]
# calculate AUC
auc = roc_auc_score(label, probs)
print('AUC: %.3f' % auc)
# calculate roc curve
fpr, tpr, thresholds = roc_curve(label, probs)
# plot no skill
plt.plot([0, 1], [0, 1], linestyle='--')
# plot the roc curve for the model
plt.plot(fpr, tpr, marker='.')
```

```
AUC: 0.839
```

[64]: [<matplotlib.lines.Line2D at 0x7f998206a690>]



[65]:
```
#Applying Decission Tree Classifier
from sklearn.tree import DecisionTreeClassifier
model3 = DecisionTreeClassifier(max_depth=5)
model3.fit(X_train,y_train)
```

[65]:
```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                       max_depth=5, max_features=None, max_leaf_nodes=None,
```

```
                              min_impurity_decrease=0.0, min_impurity_split=None,
                              min_samples_leaf=1, min_samples_split=2,
                              min_weight_fraction_leaf=0.0, presort='deprecated',
                              random_state=None, splitter='best')
```

[66]:
```
model3.score(X_train,y_train)
```

[66]:
```
0.8192182410423453
```

[67]:
```
model3.score(X_test,y_test)
```

[67]:
```
0.7532467532467533
```

[68]:
```
#Applying Random Forest
from sklearn.ensemble import RandomForestClassifier
model4 = RandomForestClassifier(n_estimators=11)
model4.fit(X_train,y_train)
```

[68]:
```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=11,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

[69]:
```
model4.score(X_train,y_train)
```

[69]:
```
0.99185667752443
```

[70]:
```
model4.score(X_test,y_test)
```

[70]:
```
0.7662337662337663
```

[71]:
```
#Support Vector Classifier

from sklearn.svm import SVC
model5 = SVC(kernel='rbf',
           gamma='auto')
model5.fit(X_train,y_train)
```

[71]:
```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

[76]:
```
model5.score(X_test,y_test)
```

[76]: 0.6168831168831169

[77]:
```
model5.score(X_test,y_test)
```

[77]: 0.6168831168831169

[78]:
```
#Applying K-NN
from sklearn.neighbors import KNeighborsClassifier
model2 = KNeighborsClassifier(n_neighbors=7,
                             metric='minkowski',
                             p = 2)
model2.fit(X_train,y_train)
```

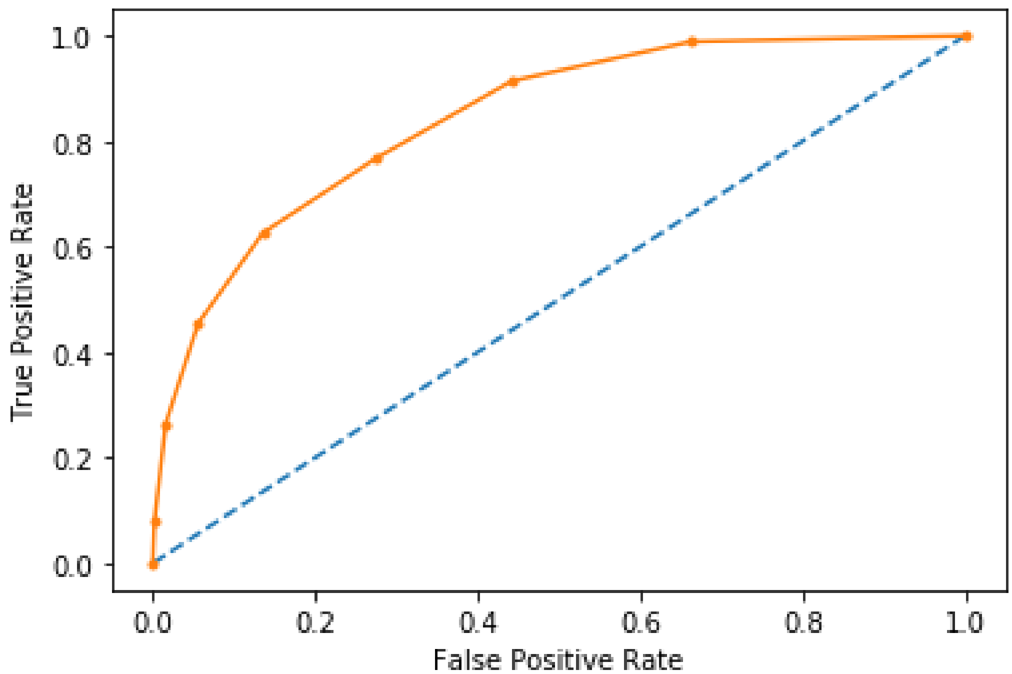
[78]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=7, p=2,
                     weights='uniform')
```

[79]:
```
#Preparing ROC Curve (Receiver Operating Characteristics Curve)
from sklearn.metrics import roc_curve
from sklearn.metrics import roc_auc_score

# predict probabilities
probs = model2.predict_proba(features)
# keep probabilities for the positive outcome only
probs = probs[:, 1]
# calculate AUC
auc = roc_auc_score(label, probs)
print('AUC: %.3f' % auc)
# calculate roc curve
fpr, tpr, thresholds = roc_curve(label, probs)
print("True Positive Rate - {}, False Positive Rate - {} Thresholds - {}".
 ↪format(tpr,fpr,thresholds))
# plot no skill
plt.plot([0, 1], [0, 1], linestyle='--')
# plot the roc curve for the model
plt.plot(fpr, tpr, marker='.')
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
```

```
AUC: 0.843
True Positive Rate - [0.         0.07835821 0.26492537 0.45522388 0.62686567
0.76865672
 0.9141791  0.98880597 1.        ], False Positive Rate - [0.    0.002 0.016
0.056 0.136 0.276 0.442 0.662 1.   ] Thresholds - [2.         1.
0.85714286 0.71428571 0.57142857 0.42857143
```

```
0.28571429 0.14285714 0.        ]
```

[79]: Text(0, 0.5, 'True Positive Rate')



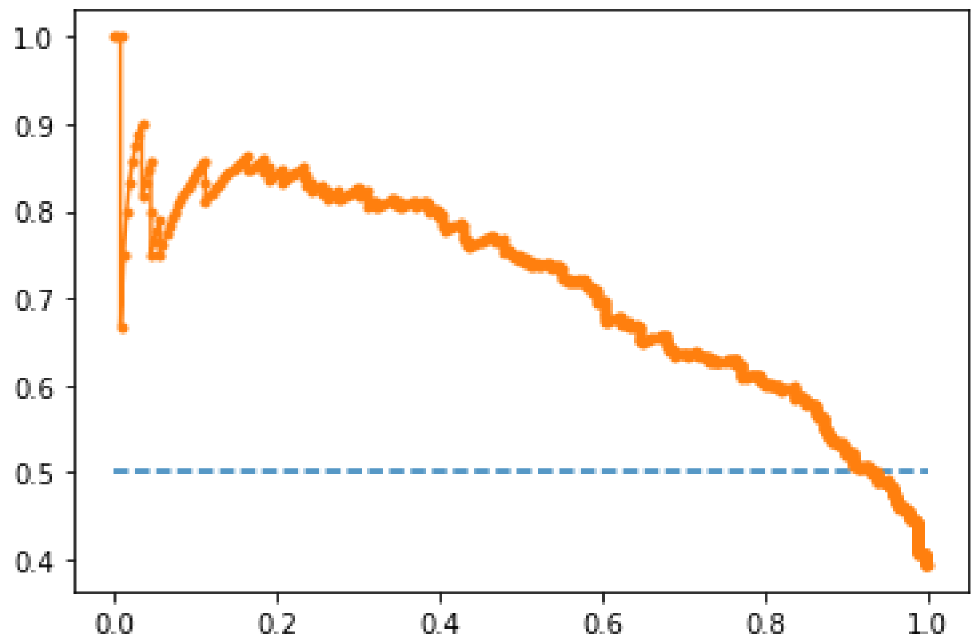
[80]:
```
#Precision Recall Curve for Logistic Regression

from sklearn.metrics import precision_recall_curve
from sklearn.metrics import f1_score
from sklearn.metrics import auc
from sklearn.metrics import average_precision_score
# predict probabilities
probs = model.predict_proba(features)
# keep probabilities for the positive outcome only
probs = probs[:, 1]
# predict class values
yhat = model.predict(features)
# calculate precision-recall curve
precision, recall, thresholds = precision_recall_curve(label, probs)
# calculate F1 score
f1 = f1_score(label, yhat)
# calculate precision-recall AUC
auc = auc(recall, precision)
# calculate average precision score
ap = average_precision_score(label, probs)
print('f1=%.3f auc=%.3f ap=%.3f' % (f1, auc, ap))
```

```
# plot no skill
plt.plot([0, 1], [0.5, 0.5], linestyle='--')
# plot the precision-recall curve for the model
plt.plot(recall, precision, marker='.')
```

```
f1=0.627 auc=0.714 ap=0.715
```

[80]:

```
[<matplotlib.lines.Line2D at 0x7f997fbd8ed0>]
```



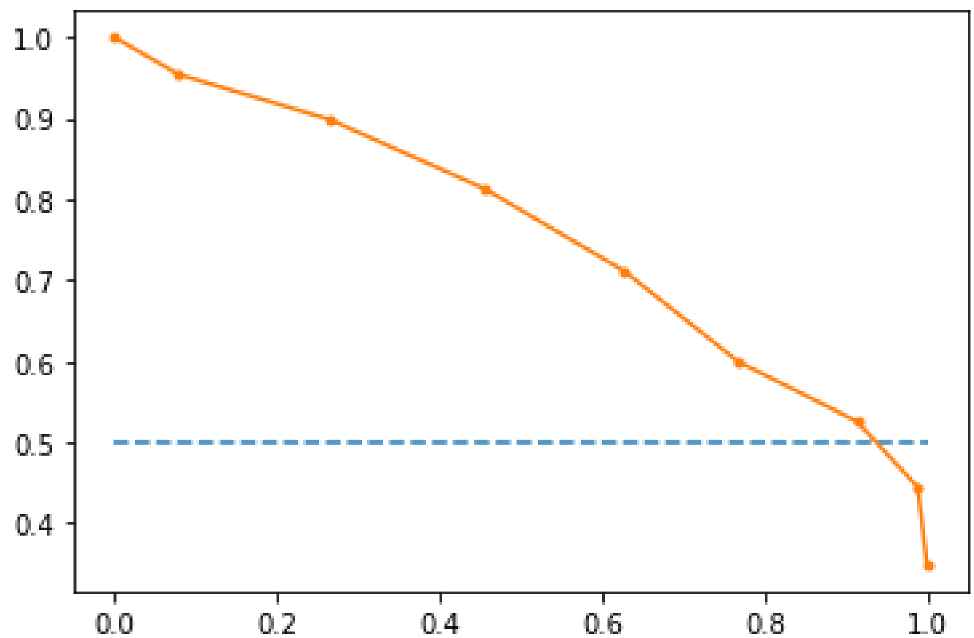
[81]:

```
#Precision Recall Curve for KNN

from sklearn.metrics import precision_recall_curve
from sklearn.metrics import f1_score
from sklearn.metrics import auc
from sklearn.metrics import average_precision_score
# predict probabilities
probs = model2.predict_proba(features)
# keep probabilities for the positive outcome only
probs = probs[:, 1]
# predict class values
yhat = model2.predict(features)
# calculate precision-recall curve
precision, recall, thresholds = precision_recall_curve(label, probs)
# calculate F1 score
f1 = f1_score(label, yhat)
# calculate precision-recall AUC
```

```
auc = auc(recall, precision)
# calculate average precision score
ap = average_precision_score(label, probs)
print('f1=%.3f auc=%.3f ap=%.3f' % (f1, auc, ap))
# plot no skill
plt.plot([0, 1], [0.5, 0.5], linestyle='--')
# plot the precision-recall curve for the model
plt.plot(recall, precision, marker='.')
```

```
f1=0.667 auc=0.759 ap=0.718
```

[81]: [<matplotlib.lines.Line2D at 0x7f997fbb56d0>]

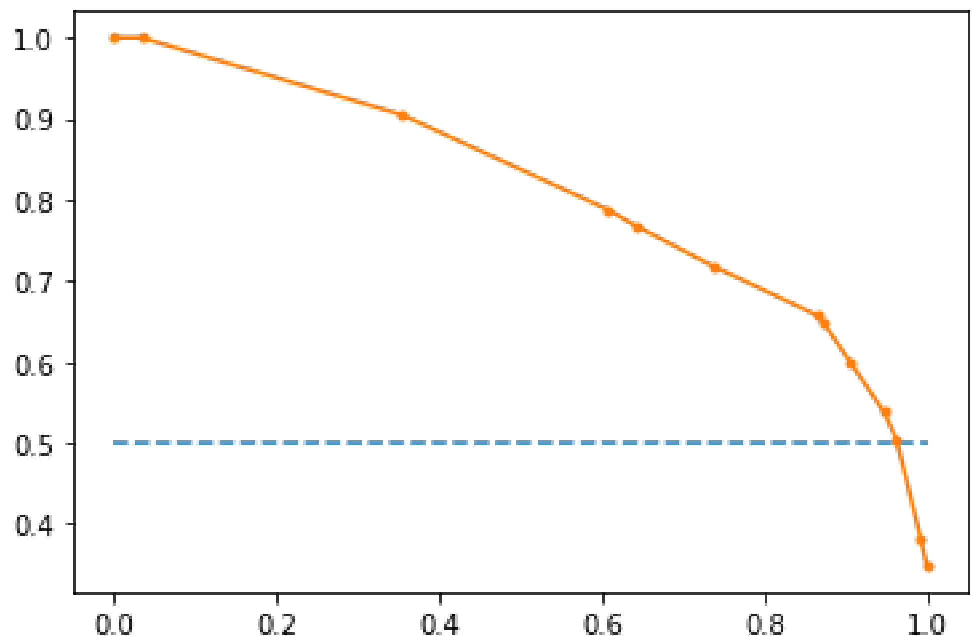


[82]:
```
#Precision Recall Curve for Decission Tree Classifier

from sklearn.metrics import precision_recall_curve
from sklearn.metrics import f1_score
from sklearn.metrics import auc
from sklearn.metrics import average_precision_score
# predict probabilities
probs = model3.predict_proba(features)
# keep probabilities for the positive outcome only
probs = probs[:, 1]
# predict class values
yhat = model3.predict(features)
# calculate precision-recall curve
```

```
precision, recall, thresholds = precision_recall_curve(label, probs)
# calculate F1 score
f1 = f1_score(label, yhat)
# calculate precision-recall AUC
auc = auc(recall, precision)
# calculate average precision score
ap = average_precision_score(label, probs)
print('f1=%.3f auc=%.3f ap=%.3f' % (f1, auc, ap))
# plot no skill
plt.plot([0, 1], [0.5, 0.5], linestyle='--')
# plot the precision-recall curve for the model
plt.plot(recall, precision, marker='.')
```

```
f1=0.686 auc=0.812 ap=0.771
```

[82]: [<matplotlib.lines.Line2D at 0x7f997fb20f50>]

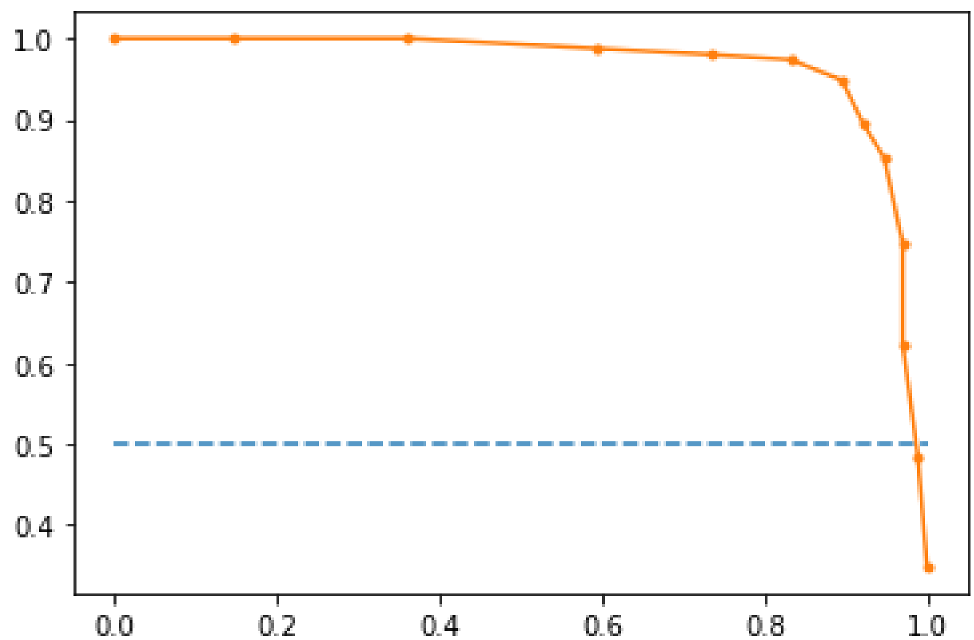


[83]:
```
#Precision Recall Curve for Random Forest

from sklearn.metrics import precision_recall_curve
from sklearn.metrics import f1_score
from sklearn.metrics import auc
from sklearn.metrics import average_precision_score
# predict probabilities
probs = model4.predict_proba(features)
# keep probabilities for the positive outcome only
```

```
probs = probs[:, 1]
# predict class values
yhat = model4.predict(features)
# calculate precision-recall curve
precision, recall, thresholds = precision_recall_curve(label, probs)
# calculate F1 score
f1 = f1_score(label, yhat)
# calculate precision-recall AUC
auc = auc(recall, precision)
# calculate average precision score
ap = average_precision_score(label, probs)
print('f1=%.3f auc=%.3f ap=%.3f' % (f1, auc, ap))
# plot no skill
plt.plot([0, 1], [0.5, 0.5], linestyle='--')
# plot the precision-recall curve for the model
plt.plot(recall, precision, marker='.')
```

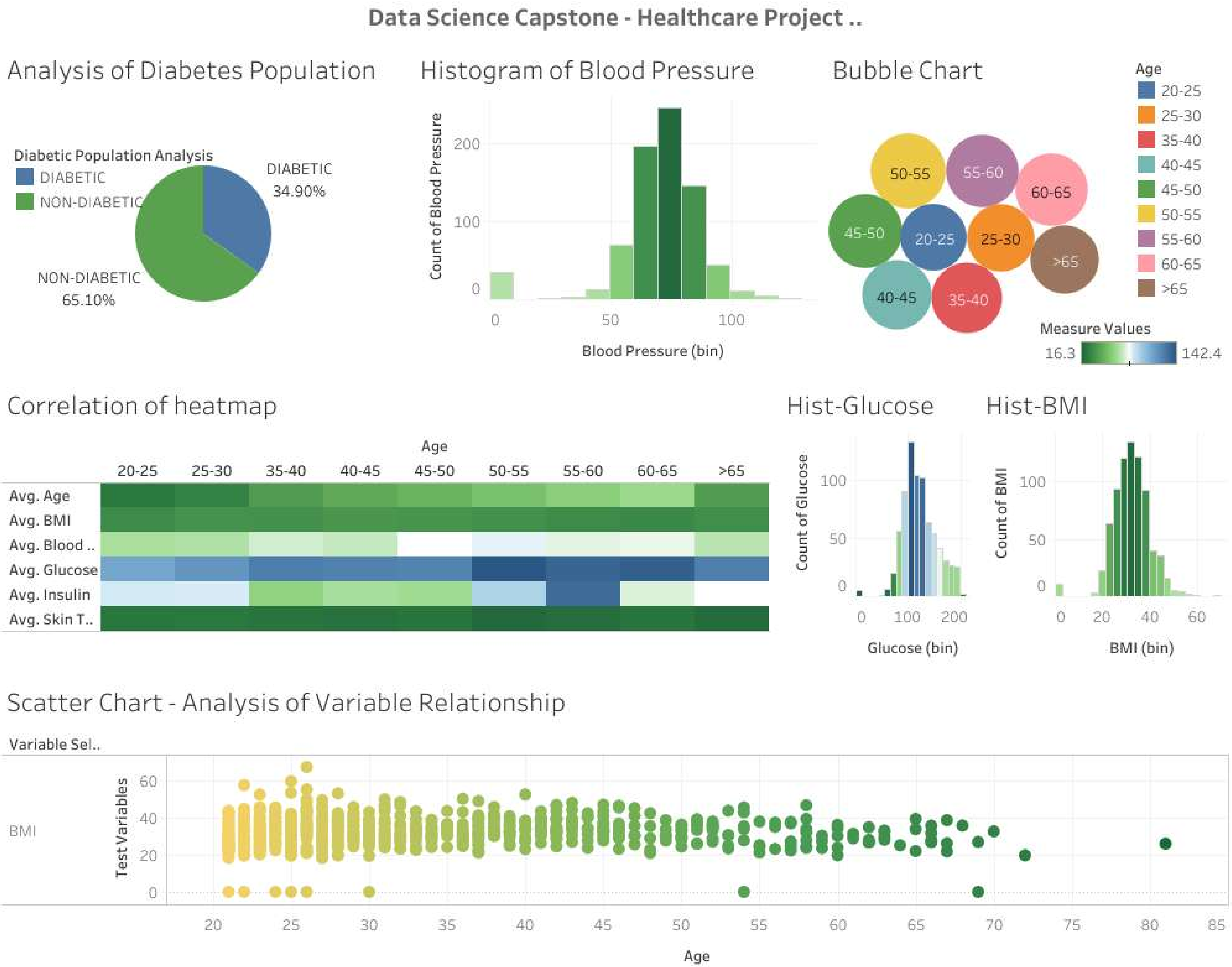
```
f1=0.921 auc=0.967 ap=0.959
```

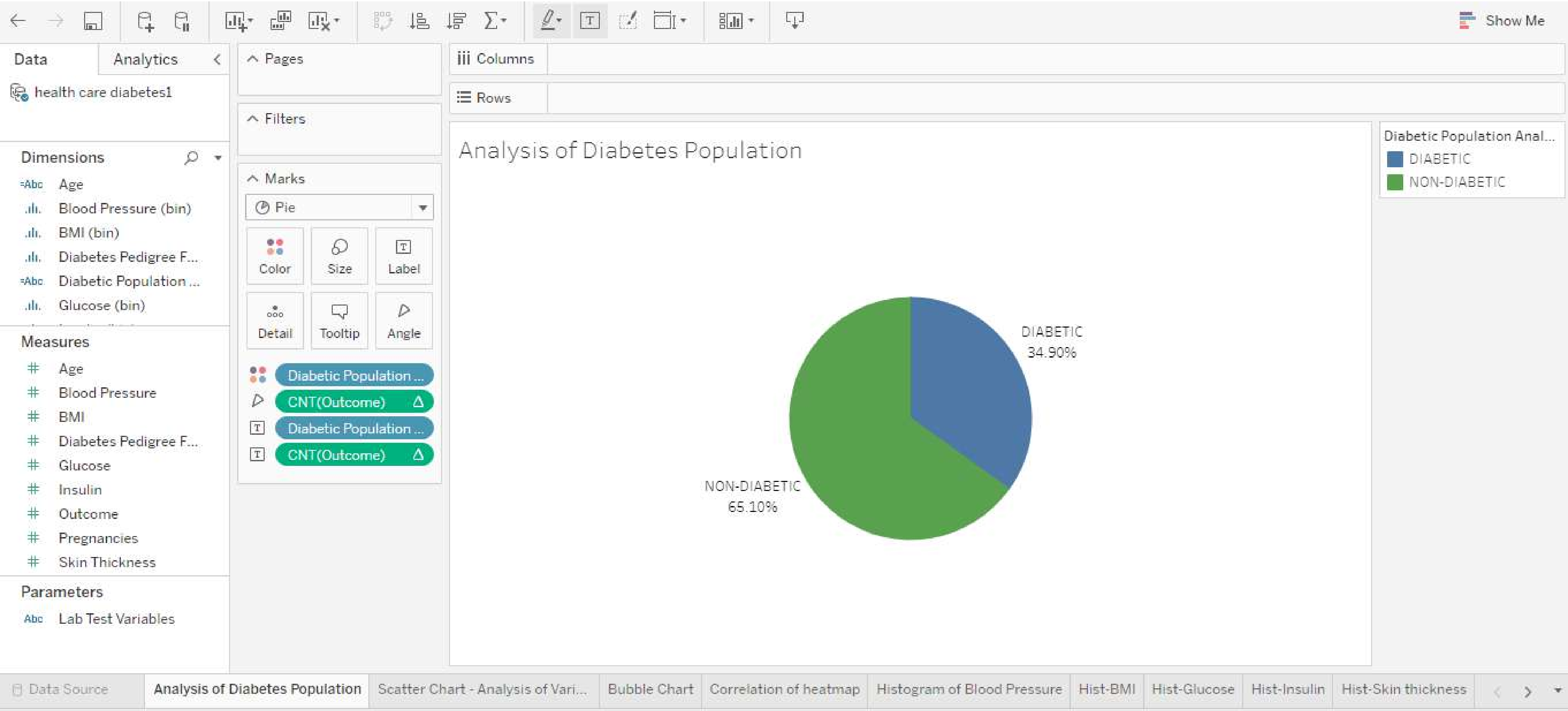
[83]: [<matplotlib.lines.Line2D at 0x7f997fb0a4d0>]

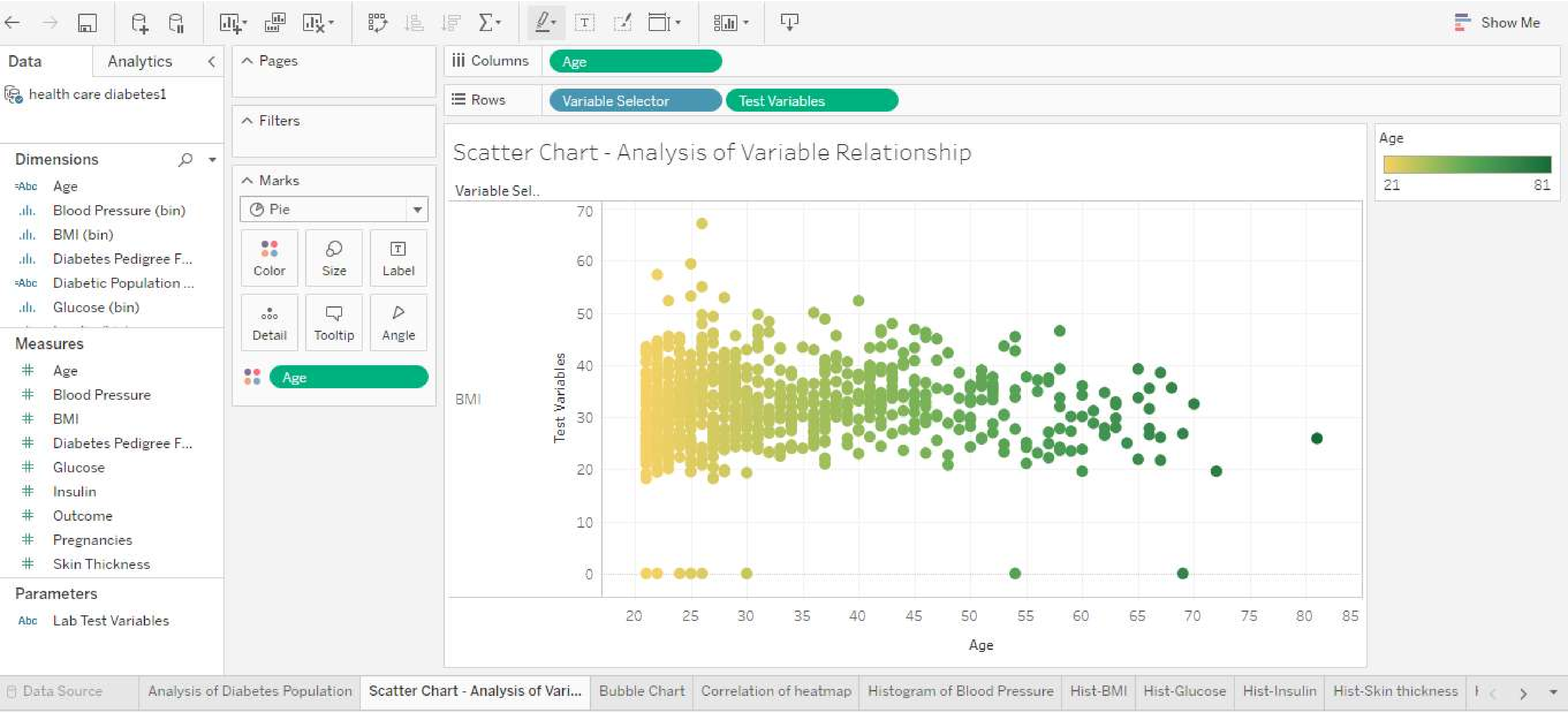


[ ]:

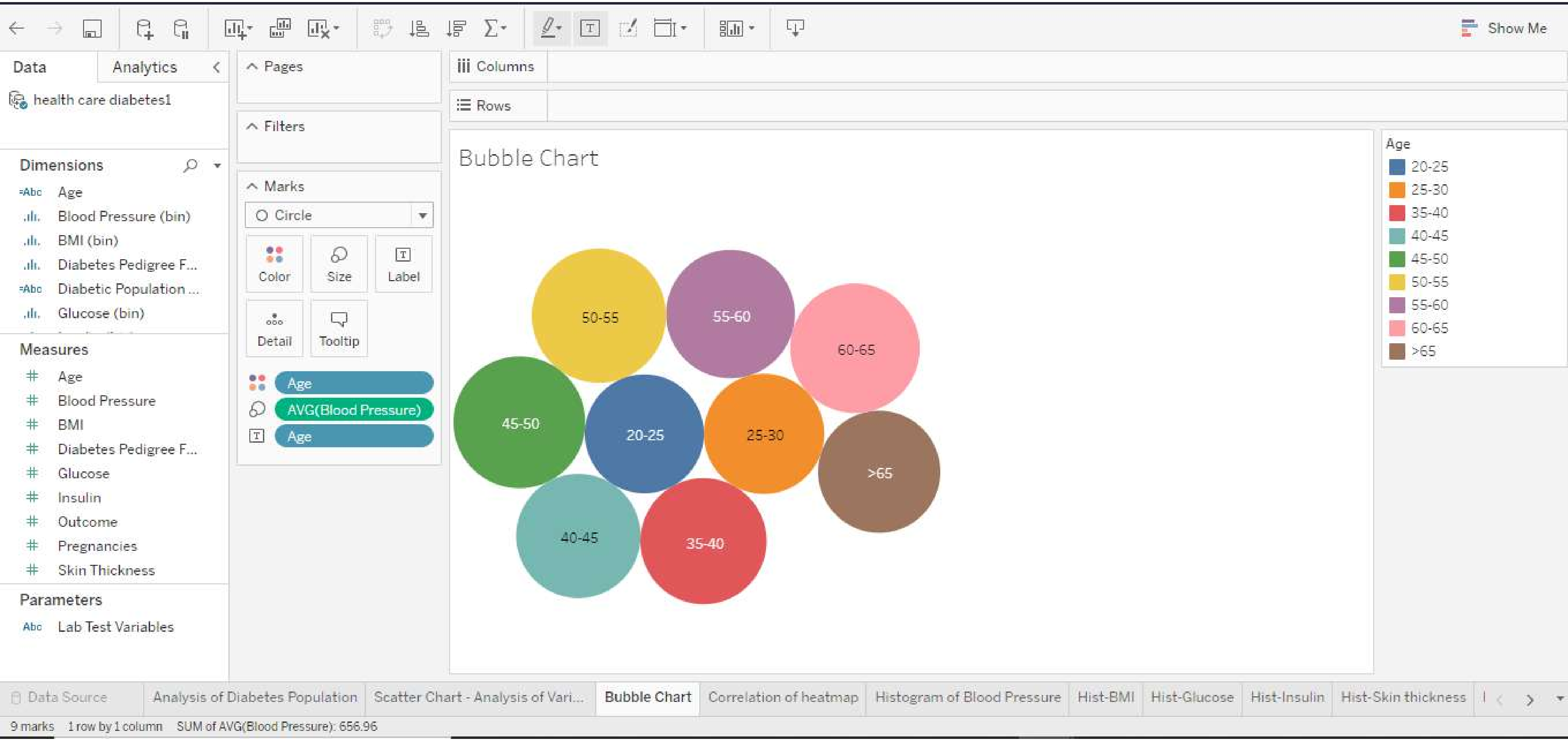
[ ]:


Data Science Capstone - Healthcare Project ..
Analysis of Diabetes Population
Diabetic Population Analysis
DIABETIC
NON-DIABETIC
DIABETIC 34.90%
NON-DIABETIC 65.10%
Histogram of Blood Pressure
Count of Blood Pressure
Blood Pressure (bin)
Bubble Chart
Age
20-25
25-30
35-40
40-45
45-50
50-55
55-60
60-65
>65
Measure Values
16.3
142.4
Correlation of heatmap
Avg. Age
Avg. BMI
Avg. Blood ..
Avg. Glucose
Avg. Insulin
Avg. Skin T..
Hist-Glucose
Count of Glucose
Glucose (bin)
Hist-BMI
Count of BMI
BMI (bin)
Scatter Chart - Analysis of Variable Relationship
Variable Sel..
BMI
Test Variables
Age

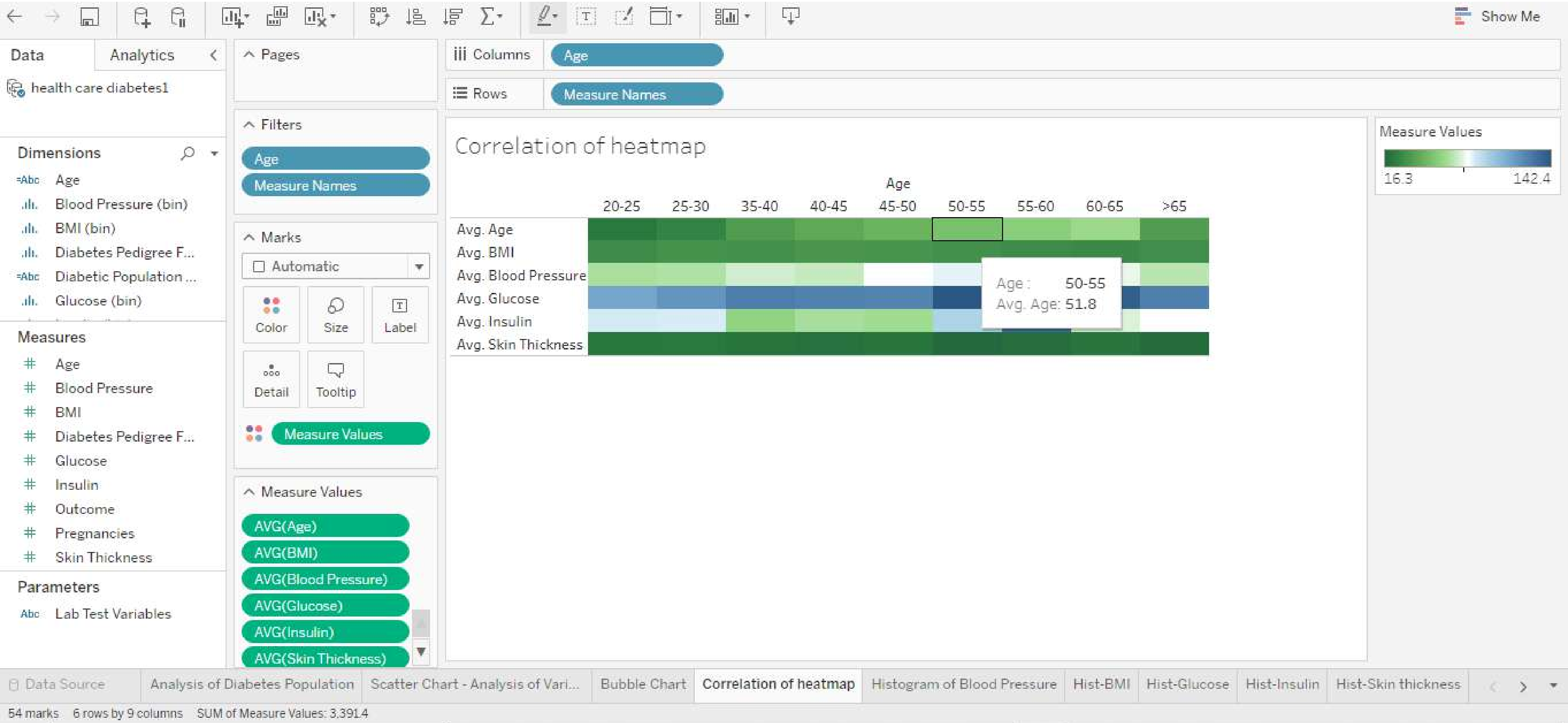

Show Me
Data
Analytics
health care diabetes1
Dimensions
Age
Blood Pressure (bin)
BMI (bin)
Diabetes Pedigree F...
Diabetic Population ...
Glucose (bin)
Measures
Age
Blood Pressure
BMI
Diabetes Pedigree F...
Glucose
Insulin
Outcome
Pregnancies
Skin Thickness
Parameters
Lab Test Variables
Pages
Filters
Marks
Pie
Color
Size
Label
Detail
Tooltip
Angle
Diabetic Population ...
CNT(Outcome)
Diabetic Population ...
CNT(Outcome)
Columns
Rows
Analysis of Diabetes Population
DIABETIC
34.90%
NON-DIABETIC
65.10%
Diabetic Population Anal...
DIABETIC
NON-DIABETIC
Data Source
Analysis of Diabetes Population
Scatter Chart - Analysis of Vari...
Bubble Chart
Correlation of heatmap
Histogram of Blood Pressure
Hist-BMI
Hist-Glucose
Hist-Insulin
Hist-Skin thickness

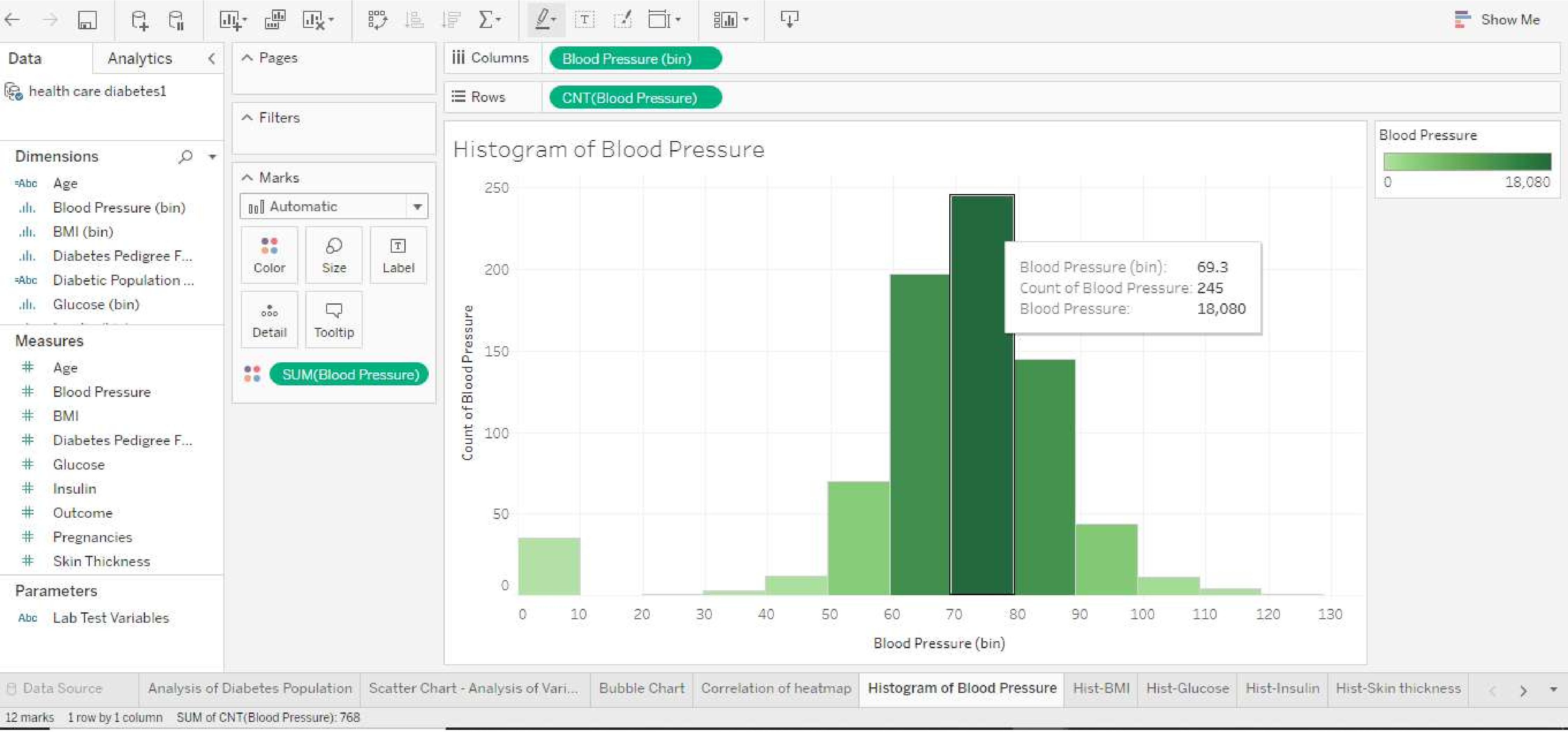


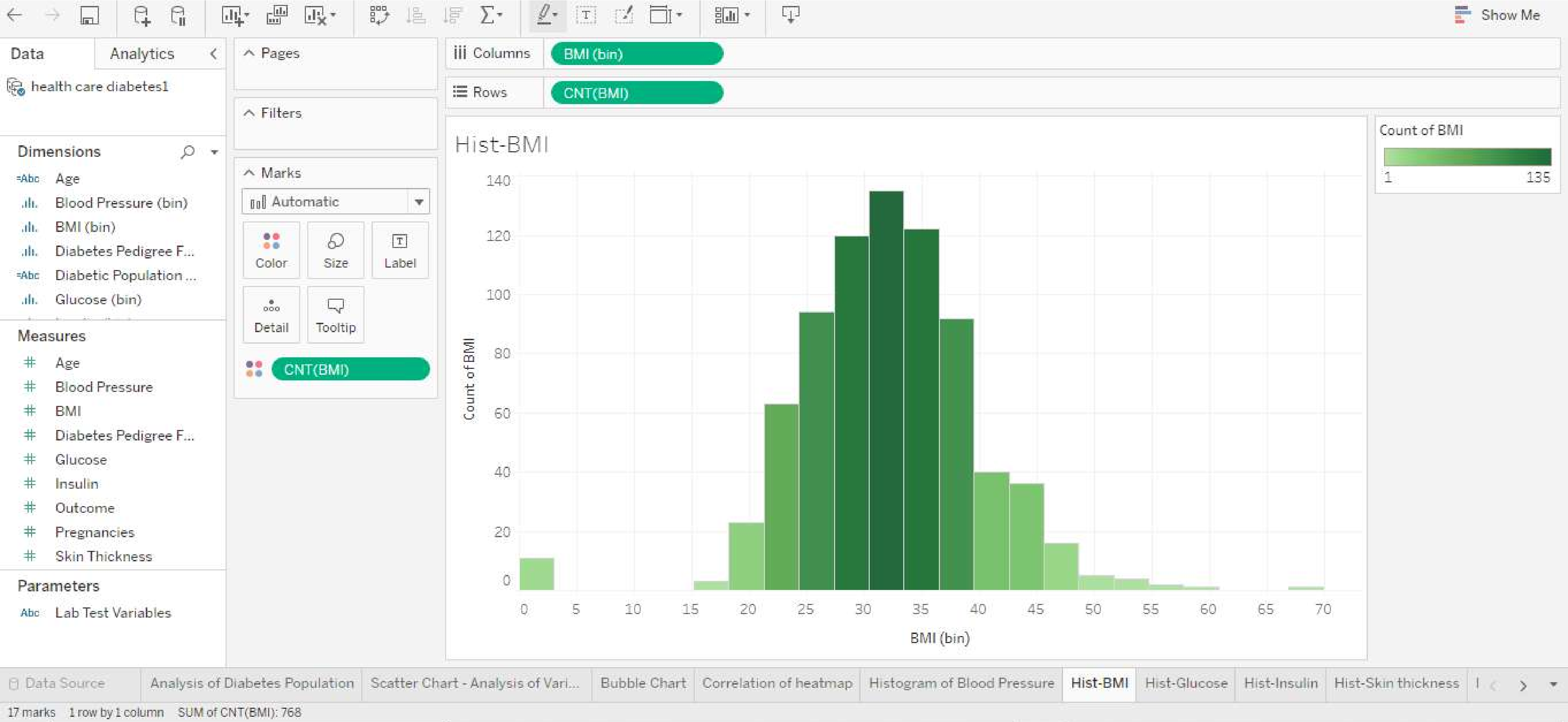
Show Me
Data
Analytics
health care diabetes1
Dimensions
Age
Blood Pressure (bin)
BMI (bin)
Diabetes Pedigree F...
Diabetic Population ...
Glucose (bin)
Measures
Age
Blood Pressure
BMI
Diabetes Pedigree F...
Glucose
Insulin
Outcome
Pregnancies
Skin Thickness
Parameters
Lab Test Variables
Pages
Filters
Marks
Pie
Color
Size
Label
Detail
Tooltip
Angle
Age
Columns
Age
Rows
Variable Selector
Test Variables
Scatter Chart - Analysis of Variable Relationship
Variable Sel..
BMI
Test Variables
Age
Age
21
81
Data Source
Analysis of Diabetes Population
Scatter Chart - Analysis of Vari...
Bubble Chart
Correlation of heatmap
Histogram of Blood Pressure
Hist-BMI
Hist-Glucose
Hist-Insulin
Hist-Skin thickness



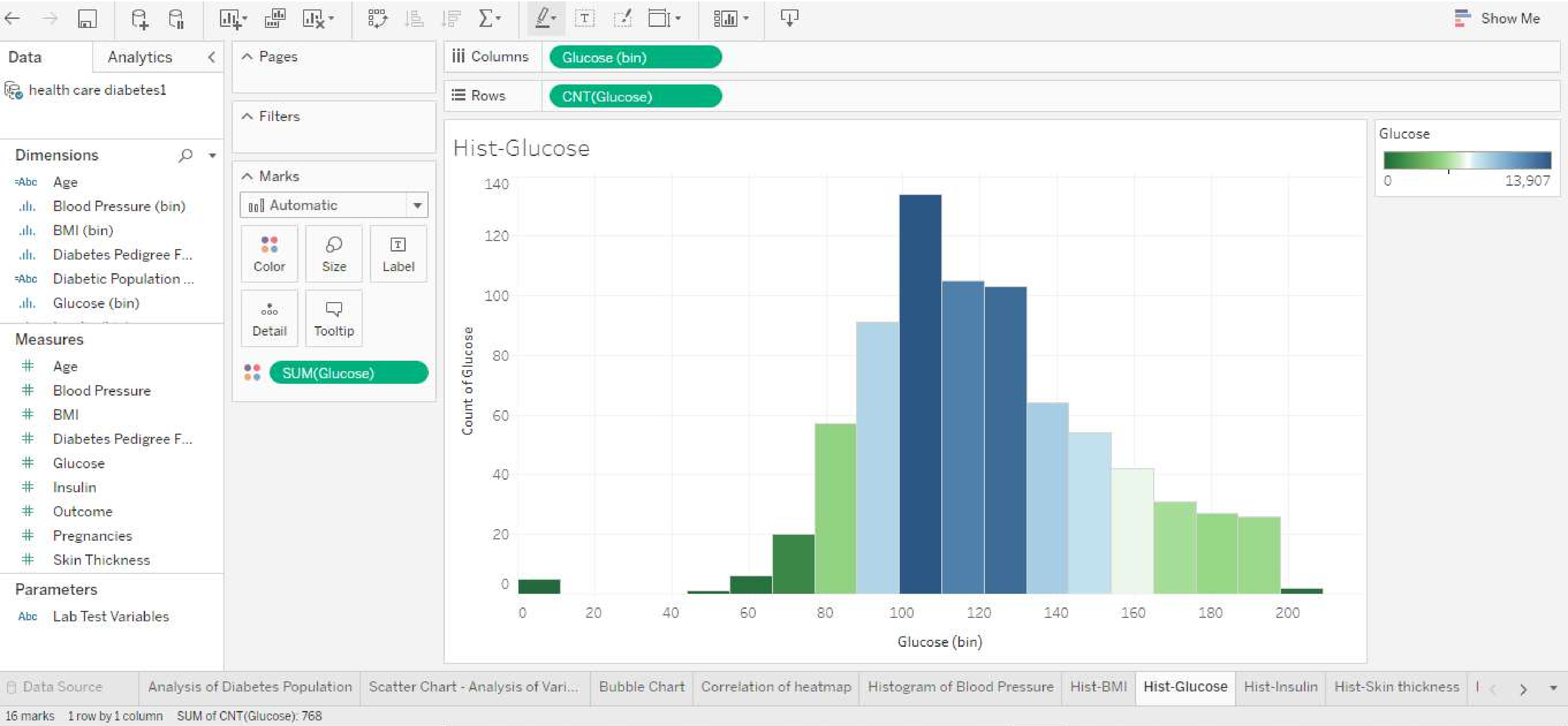
Show Me
Data
Analytics
health care diabetes1
Dimensions
Age
Blood Pressure (bin)
BMI (bin)
Diabetes Pedigree F...
Diabetic Population ...
Glucose (bin)
Measures
Age
Blood Pressure
BMI
Diabetes Pedigree F...
Glucose
Insulin
Outcome
Pregnancies
Skin Thickness
Parameters
Lab Test Variables
Pages
Filters
Marks
Circle
Color
Size
Label
Detail
Tooltip
Age
AVG(Blood Pressure)
Age
Columns
Rows
Bubble Chart
50-55
55-60
60-65
45-50
20-25
25-30
>65
40-45
35-40
Age
20-25
25-30
35-40
40-45
45-50
50-55
55-60
60-65
>65
Data Source
Analysis of Diabetes Population
Scatter Chart - Analysis of Vari...
Bubble Chart
Correlation of heatmap
Histogram of Blood Pressure
Hist-BMI
Hist-Glucose
Hist-Insulin
Hist-Skin thickness
9 marks
1 row by 1 column
SUM of AVG(Blood Pressure): 656.96

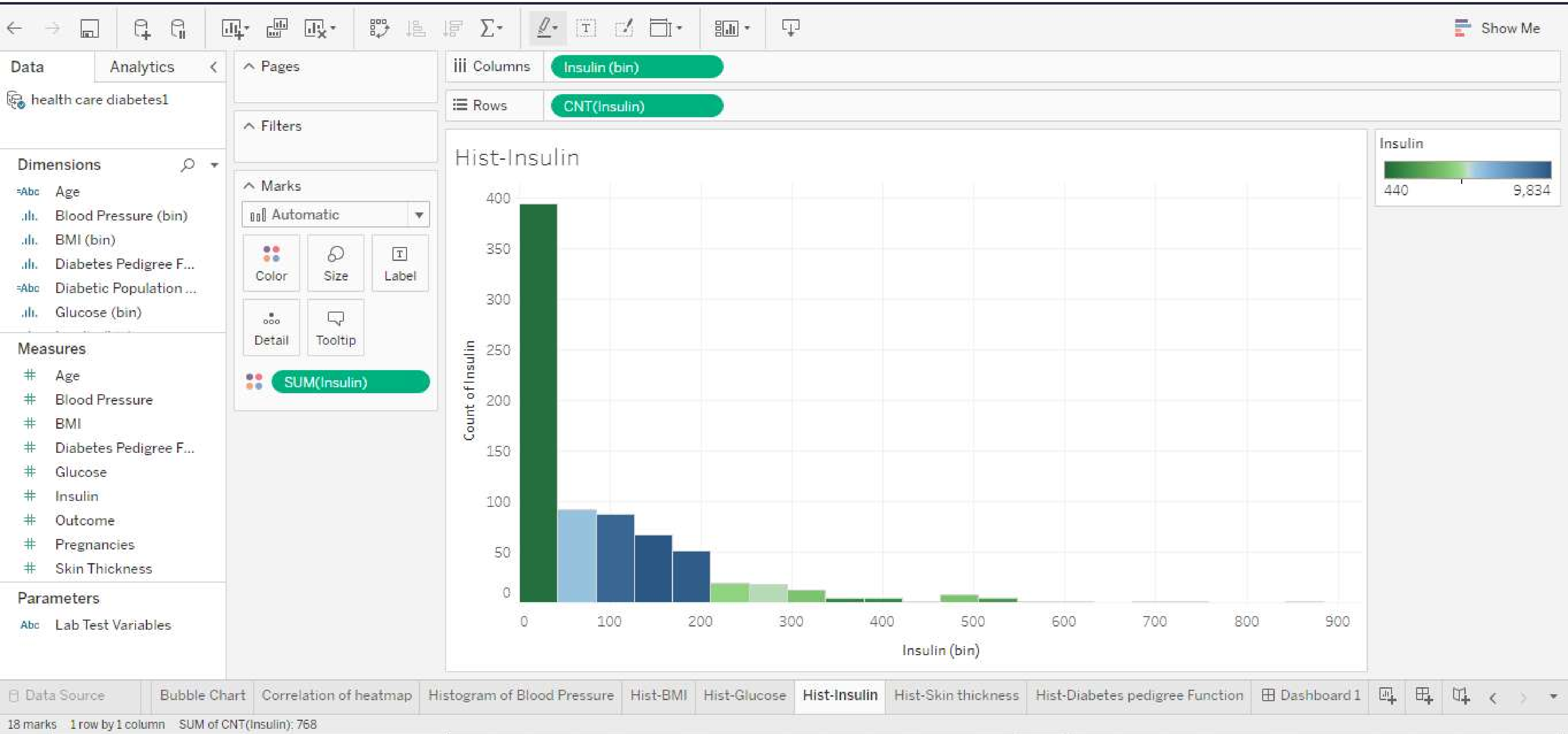

Show Me
Data
Analytics
health care diabetes1
Dimensions
Age
Blood Pressure (bin)
BMI (bin)
Diabetes Pedigree F...
Diabetic Population ...
Glucose (bin)
Measures
Age
Blood Pressure
BMI
Diabetes Pedigree F...
Glucose
Insulin
Outcome
Pregnancies
Skin Thickness
Parameters
Lab Test Variables
Pages
Filters
Age
Measure Names
Marks
Automatic
Color
Size
Label
Detail
Tooltip
Measure Values
Measure Values
AVG(Age)
AVG(BMI)
AVG(Blood Pressure)
AVG(Glucose)
AVG(Insulin)
AVG(Skin Thickness)
Columns
Age
Rows
Measure Names
Correlation of heatmap
Age
20-25
25-30
35-40
40-45
45-50
50-55
55-60
60-65
>65
Avg. Age
Avg. BMI
Avg. Blood Pressure
Avg. Glucose
Avg. Insulin
Avg. Skin Thickness
Age : 50-55
Avg. Age: 51.8
Measure Values
16.3
142.4
Data Source
Analysis of Diabetes Population
Scatter Chart - Analysis of Vari...
Bubble Chart
Correlation of heatmap
Histogram of Blood Pressure
Hist-BMI
Hist-Glucose
Hist-Insulin
Hist-Skin thickness
54 marks
6 rows by 9 columns
SUM of Measure Values: 3,391.4


Show Me
Data
Analytics
Pages
Columns
Blood Pressure (bin)
Rows
CNT(Blood Pressure)
health care diabetes1
Filters
Dimensions
Age
Blood Pressure (bin)
BMI (bin)
Diabetes Pedigree F...
Diabetic Population ...
Glucose (bin)
Marks
Automatic
Color
Size
Label
Detail
Tooltip
SUM(Blood Pressure)
Measures
Age
Blood Pressure
BMI
Diabetes Pedigree F...
Glucose
Insulin
Outcome
Pregnancies
Skin Thickness
Parameters
Lab Test Variables
Histogram of Blood Pressure
Blood Pressure
0
18,080
Blood Pressure (bin): 69.3
Count of Blood Pressure: 245
Blood Pressure: 18,080
Count of Blood Pressure
250
200
150
100
50
0
0 10 20 30 40 50 60 70 80 90 100 110 120 130
Blood Pressure (bin)
Data Source
Analysis of Diabetes Population
Scatter Chart - Analysis of Vari...
Bubble Chart
Correlation of heatmap
Histogram of Blood Pressure
Hist-BMI
Hist-Glucose
Hist-Insulin
Hist-Skin thickness
12 marks 1 row by 1 column SUM of CNT(Blood Pressure): 768

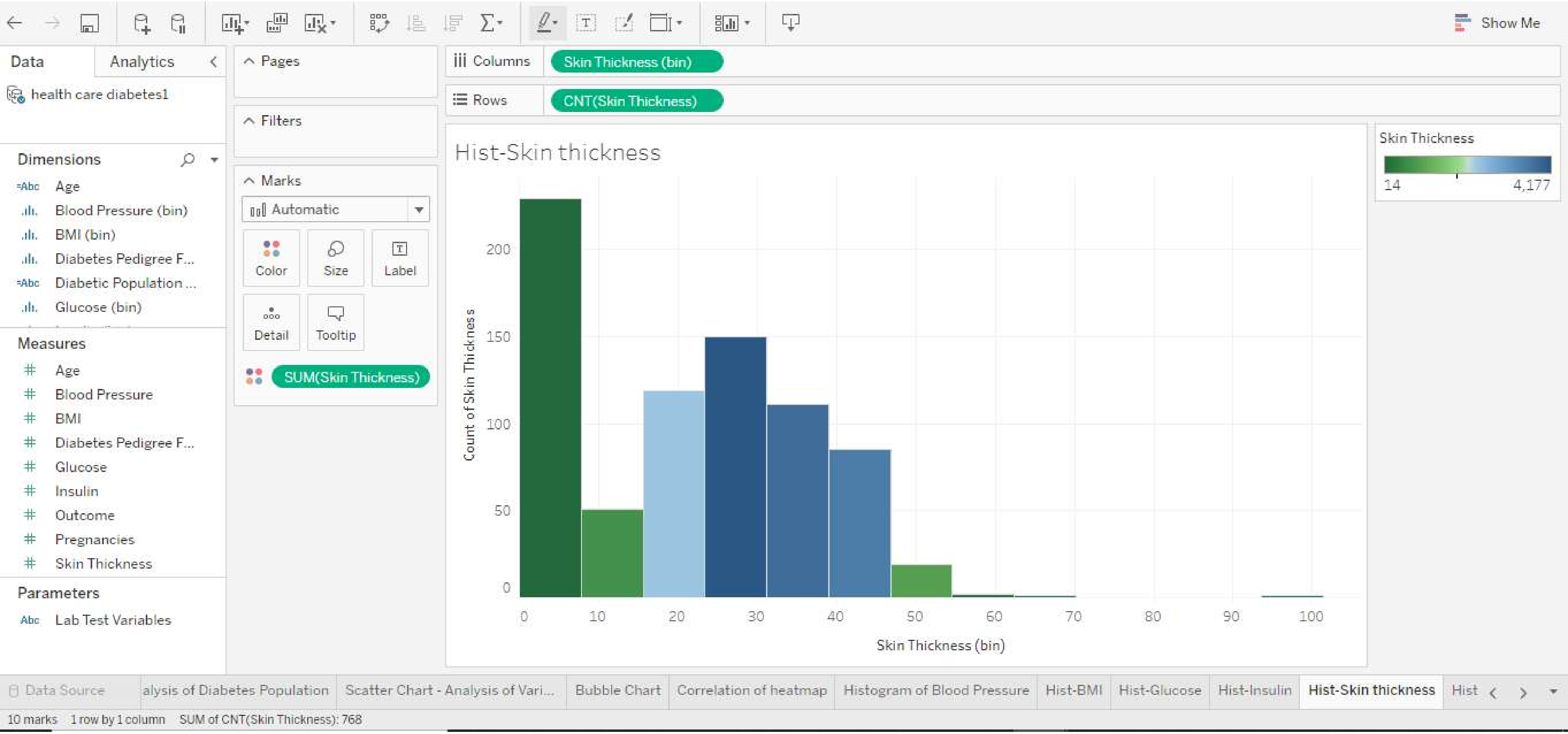


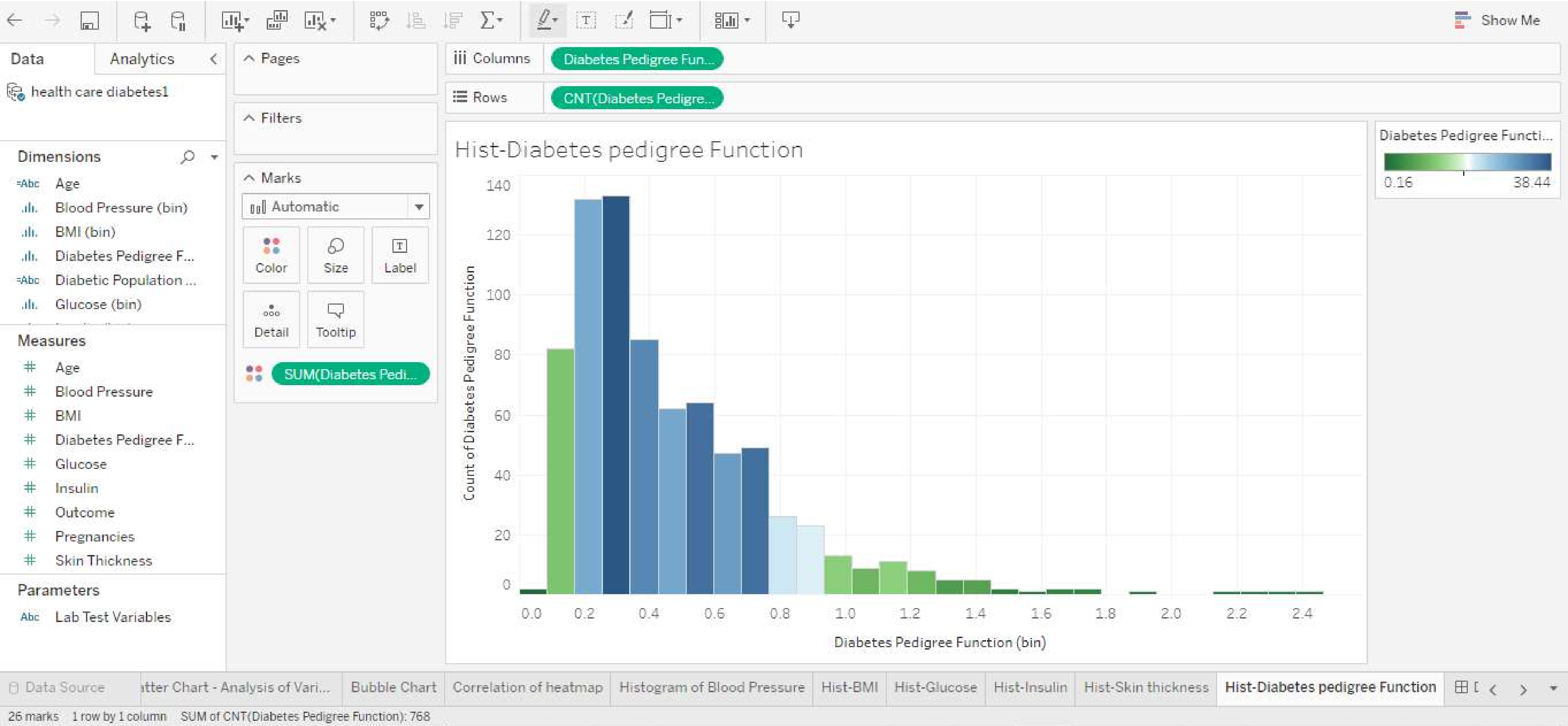
Show Me
Data
Analytics
health care diabetes1
Dimensions
Age
Blood Pressure (bin)
BMI (bin)
Diabetes Pedigree F...
Diabetic Population ...
Glucose (bin)
Measures
Age
Blood Pressure
BMI
Diabetes Pedigree F...
Glucose
Insulin
Outcome
Pregnancies
Skin Thickness
Parameters
Lab Test Variables
Pages
Filters
Marks
Automatic
Color
Size
Label
Detail
Tooltip
CNT(BMI)
Columns
BMI (bin)
Rows
CNT(BMI)
Hist-BMI
Count of BMI
1
135
Count of BMI
140
120
100
80
60
40
20
0
0
5
10
15
20
25
30
35
40
45
50
55
60
65
70
BMI (bin)
Data Source
Analysis of Diabetes Population
Scatter Chart - Analysis of Vari...
Bubble Chart
Correlation of heatmap
Histogram of Blood Pressure
Hist-BMI
Hist-Glucose
Hist-Insulin
Hist-Skin thickness
17 marks
1 row by 1 column
SUM of CNT(BMI): 768


Show Me
Data
Analytics
health care diabetes1
Dimensions
Age
Blood Pressure (bin)
BMI (bin)
Diabetes Pedigree F...
Diabetic Population ...
Glucose (bin)
Measures
Age
Blood Pressure
BMI
Diabetes Pedigree F...
Glucose
Insulin
Outcome
Pregnancies
Skin Thickness
Parameters
Lab Test Variables
Pages
Filters
Marks
Automatic
Color
Size
Label
Detail
Tooltip
SUM(Glucose)
Columns
Glucose (bin)
Rows
CNT(Glucose)
Hist-Glucose
Count of Glucose
Glucose (bin)
Glucose
0
13,907
Data Source
Analysis of Diabetes Population
Scatter Chart - Analysis of Vari...
Bubble Chart
Correlation of heatmap
Histogram of Blood Pressure
Hist-BMI
Hist-Glucose
Hist-Insulin
Hist-Skin thickness
16 marks
1 row by 1 column
SUM of CNT(Glucose): 768



Show Me
Data
Analytics
health care diabetes1
Pages
Filters
Marks
Automatic
Color
Size
Label
Detail
Tooltip
SUM(Insulin)
Columns
Insulin (bin)
Rows
CNT(Insulin)
Dimensions
Age
Blood Pressure (bin)
BMI (bin)
Diabetes Pedigree F...
Diabetic Population ...
Glucose (bin)
Measures
Age
Blood Pressure
BMI
Diabetes Pedigree F...
Glucose
Insulin
Outcome
Pregnancies
Skin Thickness
Parameters
Lab Test Variables
Hist-Insulin
Count of Insulin
400
350
300
250
200
150
100
50
0
0
100
200
300
400
500
600
700
800
900
Insulin (bin)
Insulin
440
9,834
Data Source
Bubble Chart
Correlation of heatmap
Histogram of Blood Pressure
Hist-BMI
Hist-Glucose
Hist-Insulin
Hist-Skin thickness
Hist-Diabetes pedigree Function
Dashboard 1
18 marks
1 row by 1 column
SUM of CNT(Insulin): 768


Show Me
Data
Analytics
health care diabetes1
Dimensions
Age
Blood Pressure (bin)
BMI (bin)
Diabetes Pedigree F...
Diabetic Population ...
Glucose (bin)
Measures
Age
Blood Pressure
BMI
Diabetes Pedigree F...
Glucose
Insulin
Outcome
Pregnancies
Skin Thickness
Parameters
Lab Test Variables
Pages
Filters
Marks
Automatic
Color
Size
Label
Detail
Tooltip
SUM(Skin Thickness)
Columns
Skin Thickness (bin)
Rows
CNT(Skin Thickness)
Hist-Skin thickness
Count of Skin Thickness
200
150
100
50
0
0
10
20
30
40
50
60
70
80
90
100
Skin Thickness (bin)
Skin Thickness
14
4,177
Data Source
alysis of Diabetes Population
Scatter Chart - Analysis of Vari...
Bubble Chart
Correlation of heatmap
Histogram of Blood Pressure
Hist-BMI
Hist-Glucose
Hist-Insulin
Hist-Skin thickness
Hist
10 marks
1 row by 1 column
SUM of CNT(Skin Thickness): 768



Show Me
Data
Analytics
health care diabetes1
Dimensions
Age
Blood Pressure (bin)
BMI (bin)
Diabetes Pedigree F...
Diabetic Population ...
Glucose (bin)
Measures
Age
Blood Pressure
BMI
Diabetes Pedigree F...
Glucose
Insulin
Outcome
Pregnancies
Skin Thickness
Parameters
Lab Test Variables
Pages
Filters
Marks
Automatic
Color
Size
Label
Detail
Tooltip
SUM(Diabetes Pedi...
Columns
Diabetes Pedigree Fun...
Rows
CNT(Diabetes Pedigre...
Hist-Diabetes pedigree Function
Diabetes Pedigree Functi...
0.16
38.44
Count of Diabetes Pedigree Function
Diabetes Pedigree Function (bin)
Data Source
atter Chart - Analysis of Vari...
Bubble Chart
Correlation of heatmap
Histogram of Blood Pressure
Hist-BMI
Hist-Glucose
Hist-Insulin
Hist-Skin thickness
Hist-Diabetes pedigree Function
26 marks
1 row by 1 column
SUM of CNT(Diabetes Pedigree Function): 768