In [1]:

```
# Import libraries here
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:

```
# Load the dataset from the CSV file
df = pd.read_csv('diabetes.csv')
# Display the first few rows of the dataset
print(df.head())
print("_______________________________________________________________________
print("")
# Get information about the dataset
print(df.info())
```

```
   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
____________________________________________________________________________
________________________________

<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
None
```

In [3]:

```
# print the number of missing values within the dataset
print("Number of Missing Values:")
print("-------------------------")
for col in df.columns:
    missing_row = df.loc[df[col] == 0].shape[0]
    print(col + ": " + str(missing_row))
print("")
```

```
Number of Missing Values:
-------------------------
Pregnancies: 111
Glucose: 5
BloodPressure: 35
SkinThickness: 227
Insulin: 374
BMI: 11
DiabetesPedigreeFunction: 0
Age: 0
Outcome: 500
```

In [4]:

```
# replace missing values with 'NaN'
print("Replacing values of '0' with 'NaN'...")
for col in df.columns:
    if col != 'Outcome':
        if col != 'Pregnancies':
            df[col] = df[col].replace(0, np.NaN)
print("")
```

```
Replacing values of '0' with 'NaN'...
```

In [5]:

```
# confirm that these columns no longer have values of zero
print("Number of Entries Equal to Zero:")
print("-------------------------")
for col in df.columns:
    missing_row = df.loc[df[col] == 0].shape[0]
    print(col + ": " + str(missing_row))
print("")
```

```
Number of Entries Equal to Zero:
-------------------------
Pregnancies: 111
Glucose: 0
BloodPressure: 0
SkinThickness: 0
Insulin: 0
BMI: 0
DiabetesPedigreeFunction: 0
Age: 0
Outcome: 500
```

In [6]:

```
# Check for missing values
df.isna().sum()
```

Out[6]:

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

In [7]:

```
# replace 'NaN' values with the mean of non-missing values
print("Replacing 'NaN' values with the mean of non-missing values...")
for col in df.columns:
    if col != 'Outcome':
        if col != 'Pregnancies':
            df[col] = df[col].fillna(df[col].mean())
print("")
```

```
Replacing 'NaN' values with the mean of non-missing values...
```

In [8]:

```
# Check for missing values
df.isna().sum()
```

Out[8]:

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

In [9]:

```
# Check for duplicated values
df.duplicated().sum()
```

Out[9]:

```
0
```

In [10]:

```
# Check the statistical summary of the dataset
print("Statistical Summary:")
print("--------------------")
print(df.describe())
```

```
Statistical Summary:
--------------------
       Pregnancies     Glucose  BloodPressure  SkinThickness     Insulin  \
count   768.000000  768.000000     768.000000     768.000000  768.000000   
mean      3.845052  121.686763      72.405184      29.153420  155.548223   
std       3.369578   30.435949      12.096346       8.790942   85.021108   
min       0.000000   44.000000      24.000000       7.000000   14.000000   
25%       1.000000   99.750000      64.000000      25.000000  121.500000   
50%       3.000000  117.000000      72.202592      29.153420  155.548223   
75%       6.000000  140.250000      80.000000      32.000000  155.548223   
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    32.457464                  0.471876   33.240885    0.348958  
std      6.875151                  0.331329   11.760232    0.476951  
min     18.200000                  0.078000   21.000000    0.000000  
25%     27.500000                  0.243750   24.000000    0.000000  
50%     32.400000                  0.372500   29.000000    0.000000  
75%     36.600000                  0.626250   41.000000    1.000000  
max     67.100000                  2.420000   81.000000    1.000000  
```

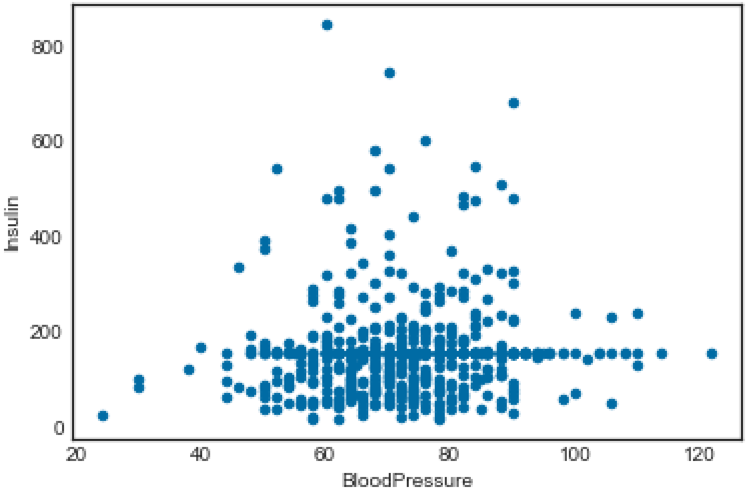
In [26]:

```
print(plt.style.available)
plt.style.use('tableau-colorblind10')
```

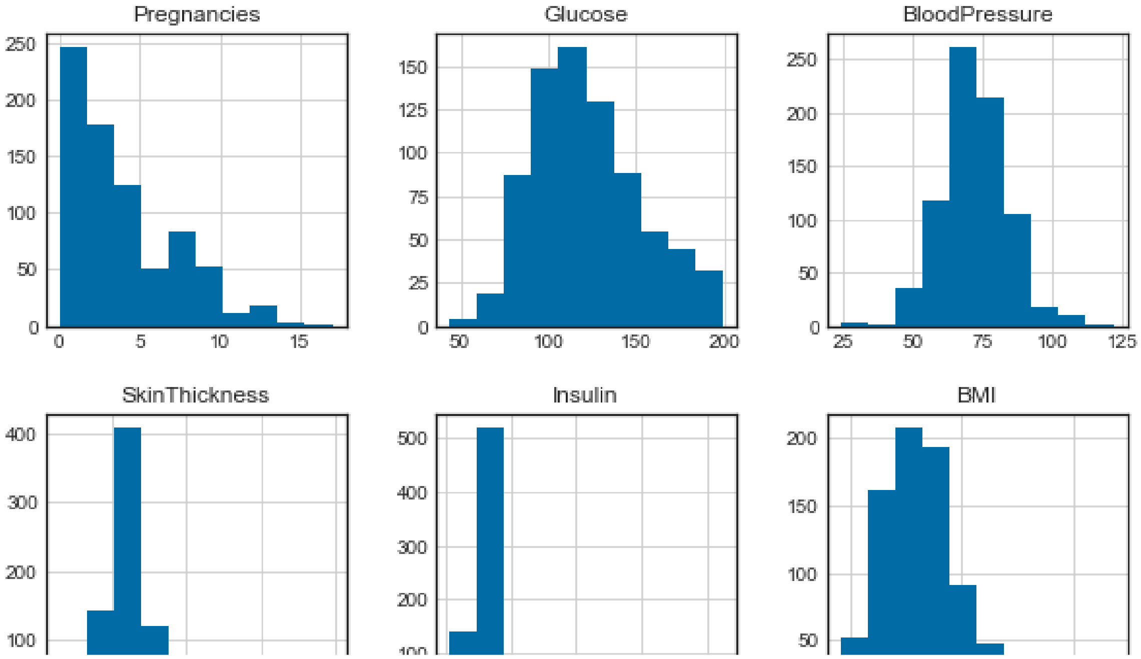
```
['Solarize_Light2', '_classic_test_patch', '_mpl-gallery', '_mpl-gallery-nogr
id', 'bmh', 'classic', 'dark_background', 'fast', 'fivethirtyeight', 'ggplo
t', 'grayscale', 'seaborn', 'seaborn-bright', 'seaborn-colorblind', 'seaborn-
dark', 'seaborn-dark-palette', 'seaborn-darkgrid', 'seaborn-deep', 'seaborn-m
uted', 'seaborn-notebook', 'seaborn-paper', 'seaborn-pastel', 'seaborn-poste
r', 'seaborn-talk', 'seaborn-ticks', 'seaborn-white', 'seaborn-whitegrid', 't
ableau-colorblind10']
```

In [46]:
```
df.plot.scatter(x="BloodPressure", y="Insulin")
```

Out[46]:
```
<AxesSubplot:xlabel='BloodPressure', ylabel='Insulin'>
```



In [30]:
```
# plot histogram
df.hist(figsize=(10, 10))
plt.show()
```



In [ ]:
```
```