```
print("NIM: 221011400957\n")
print("Nama: Farmin Wabula\n")
print("Matkul: Data Mining\n")
```

```
NIM: 221011400957

Nama: Farmin Wabula

Matkul: Data Mining
```

1. Install Kaggle

```
# Menginstal paket kaggle dari PyPI menggunakan pip
!pip install -q kaggle
```

```
# Membuat direktori .kaggle di direktori pengguna (jika belum ada)
!mkdir -p ~/.kaggle
```

```
# Menyalin file kaggle.json ke direktori .kaggle di direktori pengguna
!cp kaggle.json ~/.kaggle/kaggle.json
# !ls -a
```

```
# Fungsi untuk memberikan izin akses file, chmod itu change mode
!chmod 600 ~/.kaggle/kaggle.json
# !ls -a /content
```

```
# Mengunduh dataset diabetes-dataset dari Kaggle
!kaggle datasets download -d akshaydattatraykhare/diabetes-dataset

# Download movie dataset
# !kaggle datasets download -d akshaydattatraykhare/movies-dataset

# Download data-for-admission-in-the-university
# !kaggle datasets download -d akshaydattatraykhare/data-for-admission-in-the-university
```

```
Downloading diabetes-dataset.zip to /content
  0% 0.00/8.91k [00:00<?, ?B/s]
100% 8.91k/8.91k [00:00<00:00, 16.7MB/s]
```

```
# Mengimpor modul zipfile untuk bekerja dengan file arsip ZIP
import zipfile

# Mengekstrak semua file dari arsip ZIP ke direktori /content/
with zipfile.ZipFile('diabetes-dataset.zip', 'r') as zip_ref:zip_ref.extractall('/content/')

# Mengekstrak semua file dari arsip ZIP ke direktori /content/movies-dataset
# !mkdir /content/movies-dataset
# with zipfile.ZipFile('movies-dataset.zip', 'r') as zip_ref:zip_ref.extractall('/content/movies-dataset')

# Mengekstrak semua file dari arsip ZIP ke direktori /content/data-for-admission-in-the-university
# !mkdir /content/data-for-admission-in-the-university
# with zipfile.ZipFile('data-for-admission-in-the-university.zip', 'r') as zip_ref:zip_ref.extractall('/content/data-for-ad
```

2. Import

```
# Mengimpor library NumPy untuk operasi numerik efisien
import numpy as np

# Mengimpor library Pandas untuk manipulasi dan analisis data tabular
import pandas as pd

# Mengimpor library Seaborn untuk visualisasi data
import seaborn as sns

# Mengimpor library Matplotlib untuk visualisasi data
import matplotlib.pyplot as plt

# Mengimpor library Warnings untuk mengelola pesan peringatan
import warnings

# Menonaktifkan pesan peringatan untuk hasil yang lebih bersih
warnings.filterwarnings('ignore')

# Mengimpor fungsi untuk evaluasi model
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report

# Mengimpor fungsi untuk membagi data menjadi data latih dan data uji
from sklearn.model_selection import train_test_split

# Mengimpor kelas RandomForestClassifier dari scikit-learn
from sklearn.ensemble import RandomForestClassifier

# Mengimpor kelas KNeighborsClassifier dari scikit-learn
from sklearn.neighbors import KNeighborsClassifier

# Mengimpor kelas MLPClassifier dari scikit-learn
from sklearn.neural_network import MLPClassifier
```

3. Read CSV

```
# Membaca file CSV
df = pd.read_csv("/content/diabetes.csv")
# df = pd.read_csv("/content/movies-dataset/tmdb_5000_credits.csv")
# df = pd.read_csv("/content/data-for-admission-in-the-university/adm_data.csv")

# Menampilkan semua record dari data
# print("Semua data:\n")
# display(df)

# Menampilkan lima record pertama dari data
print("1. Lima record pertama:\n")
display(df.head(2))

# Menampilkan dua record terakhir dari data
print("\n\n\n2. Dua record terakhir:\n")
display(df.tail(2))

# Menampilkan empat record secara acak dari data
print("\n\n\n3. Empat record acak:\n")
display(df.sample(4))
```

```
1. Lima record pertama:
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesP
0	6	148	72	35	0	33.6	
1	1	85	66	29	0	26.6	

```
2. Dua record terakhir:
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Diabete
766	1	126	60	0	0	30.1	
767	1	93	70	31	0	30.4	

```
3. Empat record acak:
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Diabete
586	8	143	66	0	0	34.9	
439	6	107	88	0	0	36.8	
675	6	195	70	0	0	30.9	
411	1	112	72	30	176	34.4	

The shape of the dataset

```
df.shape
```

```
(768, 9)
```

List types of all columns

```
df.dtypes
```

```
Pregnancies                   int64
Glucose                       int64
BloodPressure                 int64
SkinThickness                 int64
Insulin                       int64
BMI                         float64
DiabetesPedigreeFunction    float64
Age                           int64
Outcome                       int64
dtype: object
```

Info of the dataset

```
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
```

Summary of the dataset

```
df.describe()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	O
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.

Drp the duplicate

```
df.shape
```

```
(768, 9)
```

```
df = df.drop_duplicates()
```

```
df.shape
```

```
(768, 9)
```

Check the null value

```
df.isnull().sum()
```

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

```
df.columns
```

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

Check the no of zero value in dataset

```
print("- No of zero value in Glucose: ", df[df["Glucose"] == 0].shape[0])
print("- No of zero value in BloodPressure: ", df[df["BloodPressure"] == 0].shape[0])
print("- No of zero value in SkinThickness: ", df[df["SkinThickness"] == 0].shape[0])
print("- No of zero value in Insulin: ", df[df["Insulin"] == 0].shape[0])
print("- No of zero value in BMI: ", df[df["BMI"] == 0].shape[0])
```

```
- No of zero value in Glucose:   5
- No of zero value in BloodPressure:   35
- No of zero value in SkinThickness:   227
- No of zero value in Insulin:   374
- No of zero value in BMI:   11
```

Replace the no of zero value with mean of columns

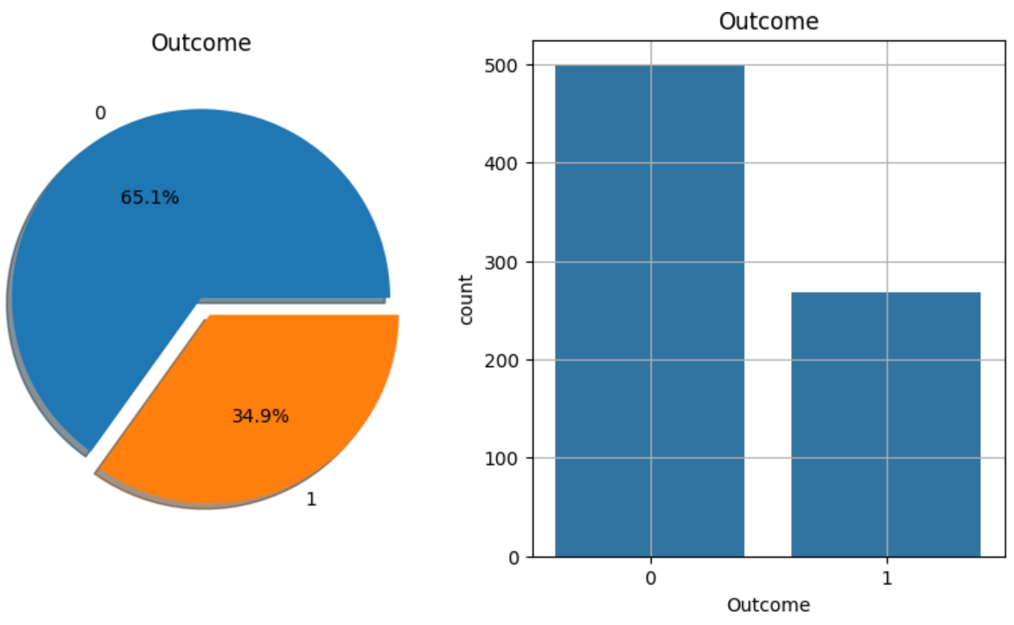
```
df["Glucose"] = df["Glucose"].replace(0, df["Glucose"].mean())
print("- No of zero value in Glucose: ", df[df["Glucose"] == 0].shape[0])

df["BloodPressure"] = df["BloodPressure"].replace(0, df["BloodPressure"].mean())
print("- No of zero value in BloodPressure: ", df[df["BloodPressure"] == 0].shape[0])

df["SkinThickness"] = df["SkinThickness"].replace(0, df["SkinThickness"].mean())
print("- No of zero value in SkinThickness: ", df[df["SkinThickness"] == 0].shape[0])

df["Insulin"] = df["Insulin"].replace(0, df["Insulin"].mean())
print("- No of zero value in Insulin: ", df[df["Insulin"] == 0].shape[0])

df["BMI"] = df["BMI"].replace(0, df["BMI"].mean())
print("- No of zero value in BMI: ", df[df["BMI"] == 0].shape[0])
```

```
    - No of zero value in Glucose:  0
    - No of zero value in BloodPressure:  0
    - No of zero value in SkinThickness:  0
    - No of zero value in Insulin:  0
    - No of zero value in BMI:  0
```

```
df.describe();
```

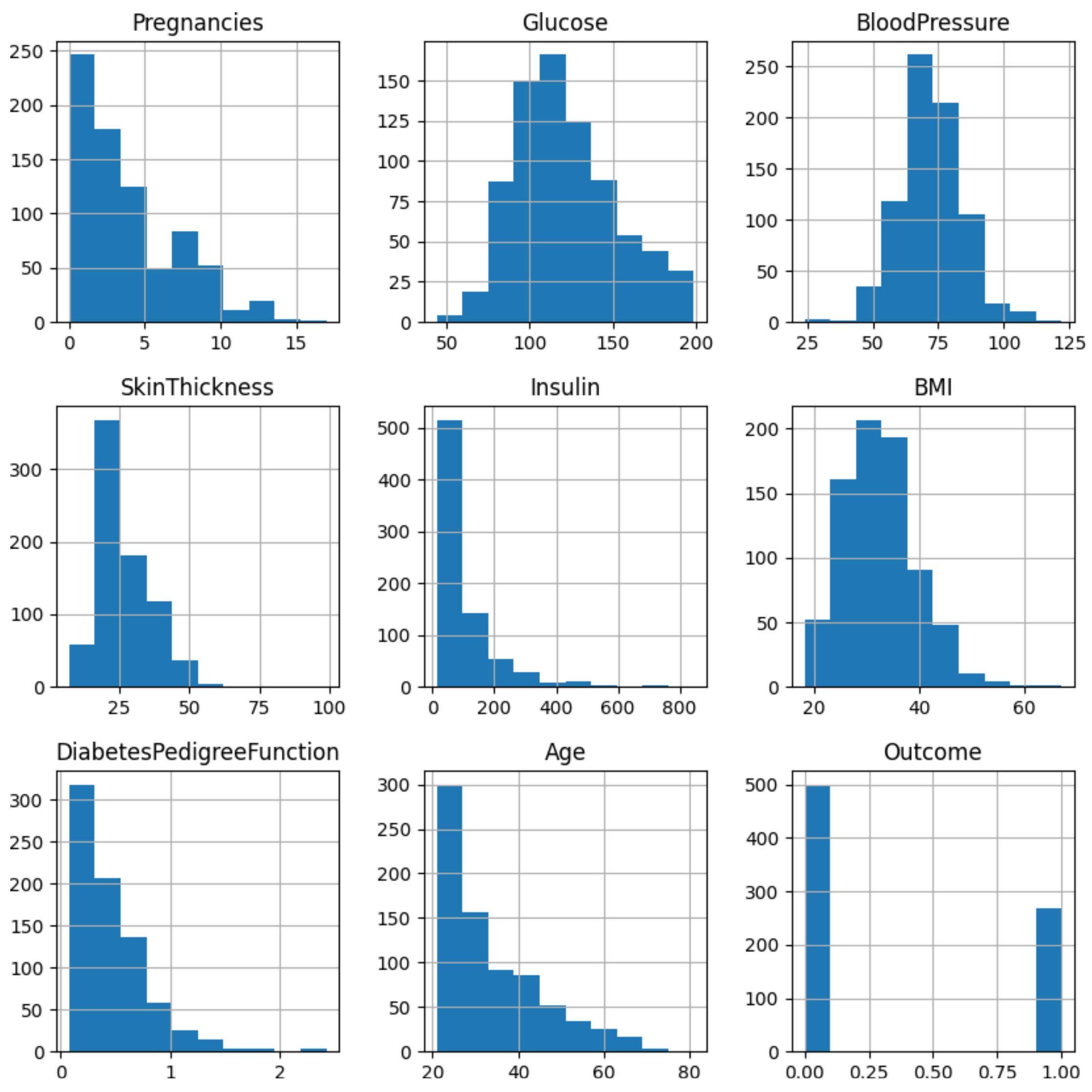
```
f,ax = plt.subplots(1, 2, figsize = (10, 5))
df["Outcome"].value_counts().plot.pie(explode = [0, 0.1],
    autopct = "%1.1f%%",
    ax = ax[0],
    shadow = True
)

ax[0].set_title("Outcome")
ax[0].set_ylabel("")
sns.countplot(x = "Outcome", data = df, ax = ax[1])

ax[1].set_title("Outcome")
N, P = df["Outcome"].value_counts()
print("- Negative(0) : ", N)
print("- Positive(1) : ", P)

plt.grid()
plt.show()
```

```
    - Negative(0) :  500
    - Positive(1) :  268
```



Histogram of aech feature

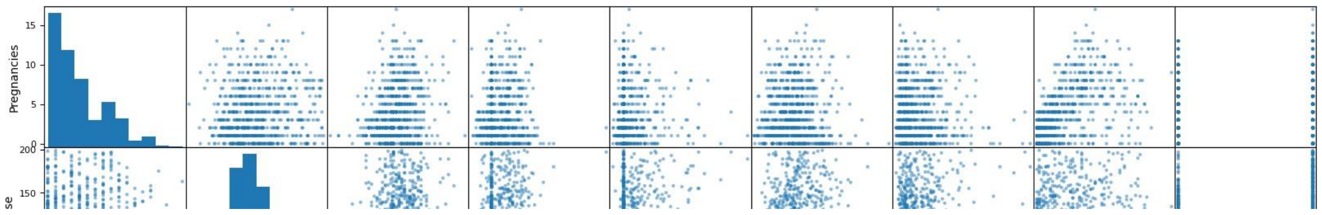
```
df.hist(bins = 10, figsize = (10, 10))
plt.show()
```

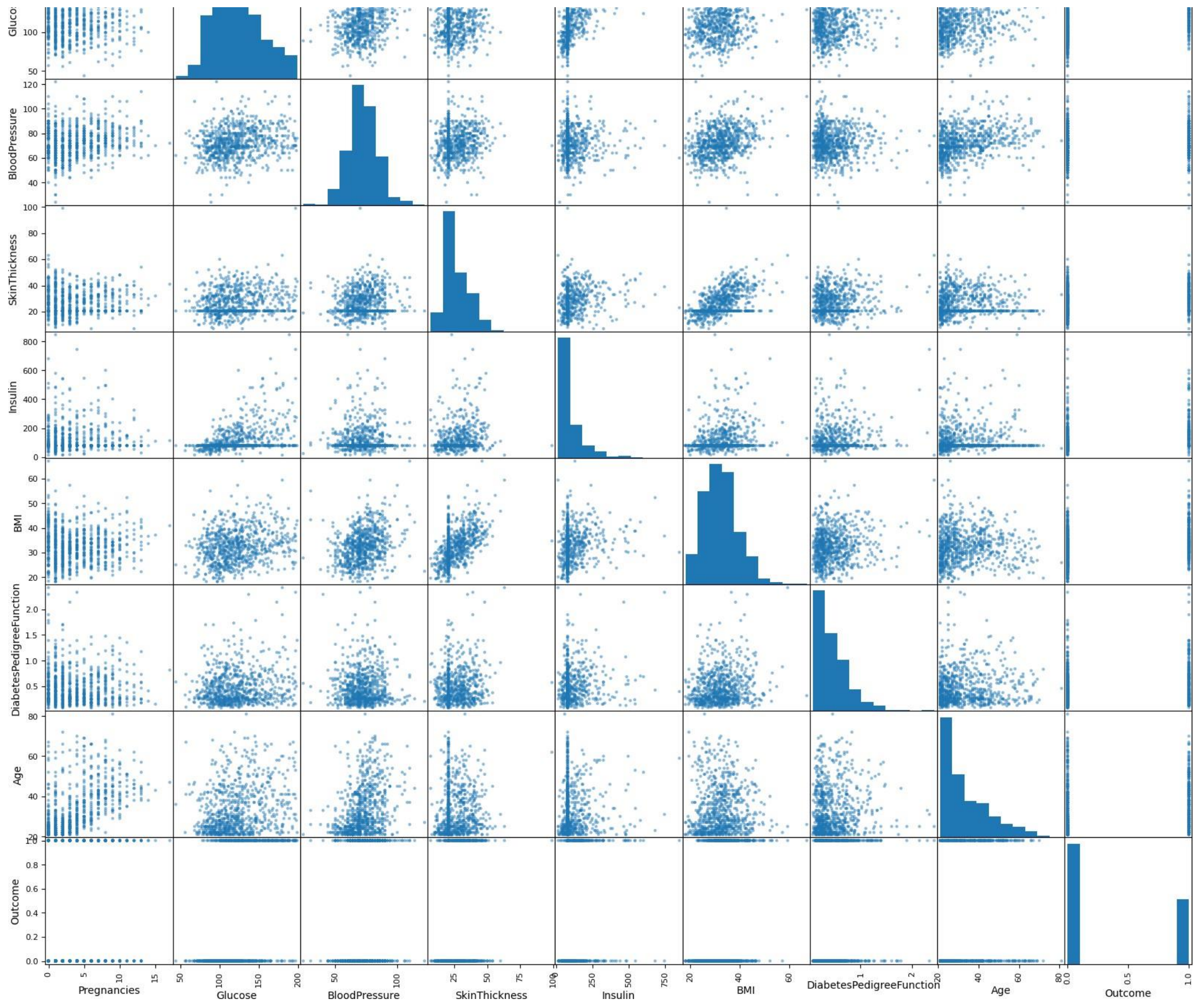


Scatter Plaot

```
from pandas.plotting import scatter_matrix

scatter_matrix(df, figsize = (20, 20))
```

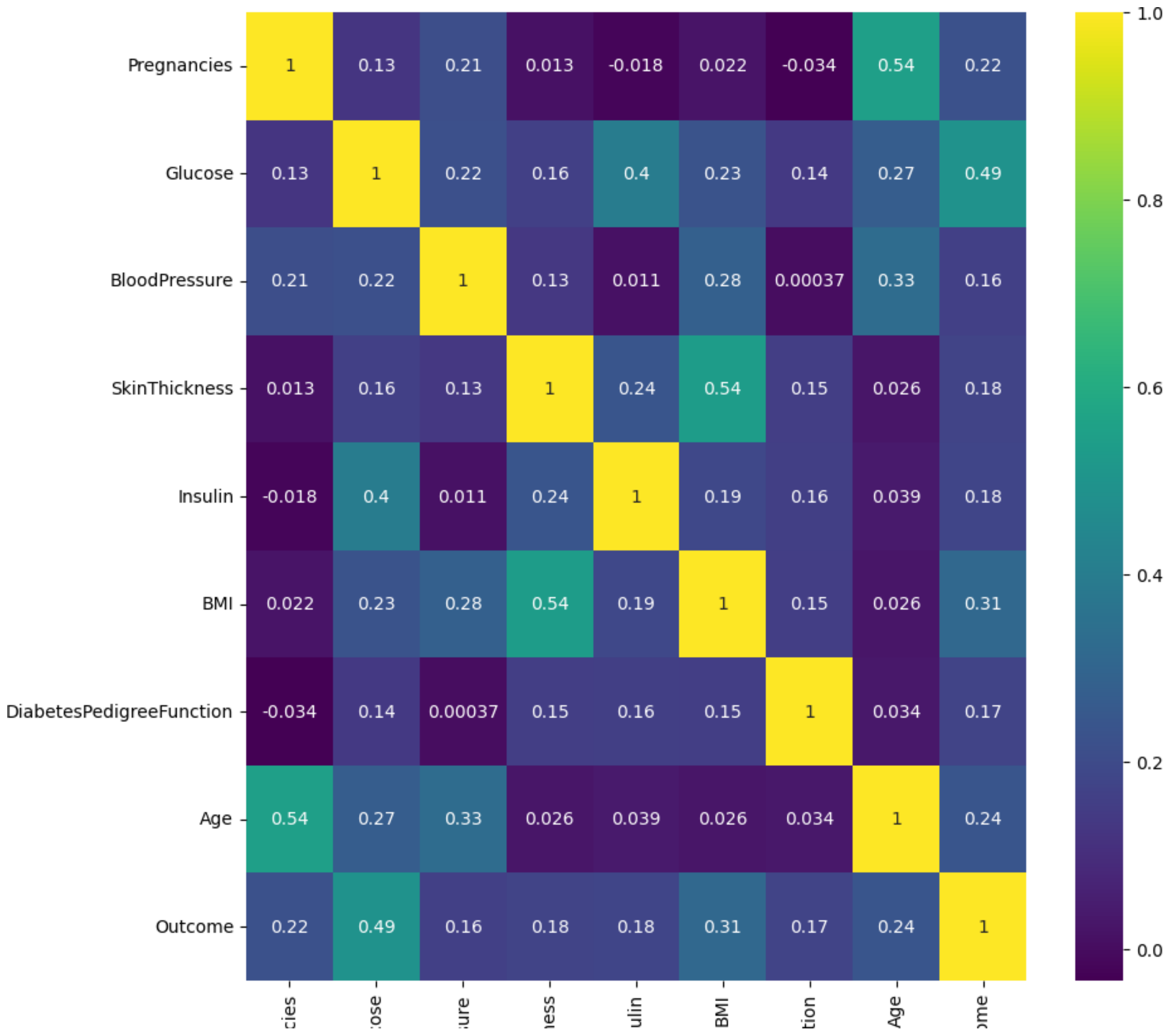
```
array([[<Axes: xlabel='Pregnancies', ylabel='Pregnancies'>,
        <Axes: xlabel='Glucose', ylabel='Pregnancies'>,
        <Axes: xlabel='BloodPressure', ylabel='Pregnancies'>,
        <Axes: xlabel='SkinThickness', ylabel='Pregnancies'>,
        <Axes: xlabel='Insulin', ylabel='Pregnancies'>,
        <Axes: xlabel='BMI', ylabel='Pregnancies'>,
        <Axes: xlabel='DiabetesPedigreeFunction', ylabel='Pregnancies'>,
        <Axes: xlabel='Age', ylabel='Pregnancies'>,
        <Axes: xlabel='Outcome', ylabel='Pregnancies'>],
       [<Axes: xlabel='Pregnancies', ylabel='Glucose'>,
        <Axes: xlabel='Glucose', ylabel='Glucose'>,
        <Axes: xlabel='BloodPressure', ylabel='Glucose'>,
        <Axes: xlabel='SkinThickness', ylabel='Glucose'>,
        <Axes: xlabel='Insulin', ylabel='Glucose'>,
        <Axes: xlabel='BMI', ylabel='Glucose'>,
        <Axes: xlabel='DiabetesPedigreeFunction', ylabel='Glucose'>,
        <Axes: xlabel='Age', ylabel='Glucose'>,
        <Axes: xlabel='Outcome', ylabel='Glucose'>],
       [<Axes: xlabel='Pregnancies', ylabel='BloodPressure'>,
        <Axes: xlabel='Glucose', ylabel='BloodPressure'>,
        <Axes: xlabel='BloodPressure', ylabel='BloodPressure'>,
        <Axes: xlabel='SkinThickness', ylabel='BloodPressure'>,
        <Axes: xlabel='Insulin', ylabel='BloodPressure'>,
        <Axes: xlabel='BMI', ylabel='BloodPressure'>,
        <Axes: xlabel='DiabetesPedigreeFunction', ylabel='BloodPressure'>,
        <Axes: xlabel='Age', ylabel='BloodPressure'>,
        <Axes: xlabel='Outcome', ylabel='BloodPressure'>],
       [<Axes: xlabel='Pregnancies', ylabel='SkinThickness'>,
        <Axes: xlabel='Glucose', ylabel='SkinThickness'>,
        <Axes: xlabel='BloodPressure', ylabel='SkinThickness'>,
        <Axes: xlabel='SkinThickness', ylabel='SkinThickness'>,
        <Axes: xlabel='Insulin', ylabel='SkinThickness'>,
        <Axes: xlabel='BMI', ylabel='SkinThickness'>,
        <Axes: xlabel='DiabetesPedigreeFunction', ylabel='SkinThickness'>,
        <Axes: xlabel='Age', ylabel='SkinThickness'>,
        <Axes: xlabel='Outcome', ylabel='SkinThickness'>],
       [<Axes: xlabel='Pregnancies', ylabel='Insulin'>,
        <Axes: xlabel='Glucose', ylabel='Insulin'>,
        <Axes: xlabel='BloodPressure', ylabel='Insulin'>,
        <Axes: xlabel='SkinThickness', ylabel='Insulin'>,
        <Axes: xlabel='Insulin', ylabel='Insulin'>,
        <Axes: xlabel='BMI', ylabel='Insulin'>,
        <Axes: xlabel='DiabetesPedigreeFunction', ylabel='Insulin'>,
        <Axes: xlabel='Age', ylabel='Insulin'>,
        <Axes: xlabel='Outcome', ylabel='Insulin'>],
       [<Axes: xlabel='Pregnancies', ylabel='BMI'>,
        <Axes: xlabel='Glucose', ylabel='BMI'>,
        <Axes: xlabel='BloodPressure', ylabel='BMI'>,
        <Axes: xlabel='SkinThickness', ylabel='BMI'>,
        <Axes: xlabel='Insulin', ylabel='BMI'>,
        <Axes: xlabel='BMI', ylabel='BMI'>,
        <Axes: xlabel='DiabetesPedigreeFunction', ylabel='BMI'>,
        <Axes: xlabel='Age', ylabel='BMI'>,
        <Axes: xlabel='Outcome', ylabel='BMI'>],
       [<Axes: xlabel='Pregnancies', ylabel='DiabetesPedigreeFunction'>,
        <Axes: xlabel='Glucose', ylabel='DiabetesPedigreeFunction'>,
        <Axes: xlabel='BloodPressure', ylabel='DiabetesPedigreeFunction'>,
        <Axes: xlabel='SkinThickness', ylabel='DiabetesPedigreeFunction'>,
        <Axes: xlabel='Insulin', ylabel='DiabetesPedigreeFunction'>,
        <Axes: xlabel='BMI', ylabel='DiabetesPedigreeFunction'>,
        <Axes: xlabel='DiabetesPedigreeFunction', ylabel='DiabetesPedigreeFunction'>,
        <Axes: xlabel='Age', ylabel='DiabetesPedigreeFunction'>,
        <Axes: xlabel='Outcome', ylabel='DiabetesPedigreeFunction'>],
       [<Axes: xlabel='Pregnancies', ylabel='Age'>,
        <Axes: xlabel='Glucose', ylabel='Age'>,
        <Axes: xlabel='BloodPressure', ylabel='Age'>,
        <Axes: xlabel='SkinThickness', ylabel='Age'>,
        <Axes: xlabel='Insulin', ylabel='Age'>,
        <Axes: xlabel='BMI', ylabel='Age'>,
        <Axes: xlabel='DiabetesPedigreeFunction', ylabel='Age'>,
        <Axes: xlabel='Age', ylabel='Age'>,
        <Axes: xlabel='Outcome', ylabel='Age'>],
       [<Axes: xlabel='Pregnancies', ylabel='Outcome'>,
        <Axes: xlabel='Glucose', ylabel='Outcome'>,
        <Axes: xlabel='BloodPressure', ylabel='Outcome'>,
        <Axes: xlabel='SkinThickness', ylabel='Outcome'>,
        <Axes: xlabel='Insulin', ylabel='Outcome'>,
        <Axes: xlabel='BMI', ylabel='Outcome'>,
        <Axes: xlabel='DiabetesPedigreeFunction', ylabel='Outcome'>,
        <Axes: xlabel='Age', ylabel='Outcome'>,
        <Axes: xlabel='Outcome', ylabel='Outcome'>]], dtype=object)
```





Analyzing relationships between variable

```
import seaborn as sns
corrmat = df.corr()
top_corr_features = corrmat.index
plt.figure(figsize = (10, 10))
g = sns.heatmap(df[top_corr_features].corr(), annot = True, cmap = "viridis")
```



	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1	0.13	0.21	0.013	-0.018	0.022	-0.034	0.54	0.22
Glucose	0.13	1	0.22	0.16	0.4	0.23	0.14	0.27	0.49
BloodPressure	0.21	0.22	1	0.13	0.011	0.28	0.00037	0.33	0.16
SkinThickness	0.013	0.16	0.13	1	0.24	0.54	0.15	0.026	0.18
Insulin	-0.018	0.4	0.011	0.24	1	0.19	0.16	0.039	0.18
BMI	0.022	0.23	0.28	0.54	0.19	1	0.15	0.026	0.31
DiabetesPedigreeFunction	-0.034	0.14	0.00037	0.15	0.16	0.15	1	0.034	0.17
Age	0.54	0.27	0.33	0.026	0.039	0.026	0.034	1	0.24
Outcome	0.22	0.49	0.16	0.18	0.18	0.31	0.17	0.24	1