

```
In [33]: import pandas as pd
df = pd.read_csv('diabetes.csv')
df.head()
```

```
Out[33]:
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Diabe
0	6.0	148	72.0	35.0	NaN	33.6	
1	1.0	85	66.0	29.0	NaN	26.6	
2	8.0	183	64.0	NaN	NaN	23.3	
3	1.0	89	66.0	23.0	94.0	28.1	
4	0.0	137	40.0	35.0	168.0	43.1	

```
In [34]: df.describe()
```

```
Out[34]:
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin
count	767.000000	768.000000	763.000000	648.000000	755.000000
mean	3.850065	120.894531	69.558322	21.825617	81.173510
std	3.368910	31.972618	18.589365	15.239728	115.751817
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	1.000000	99.000000	64.000000	5.250000	0.000000
50%	3.000000	117.000000	72.000000	24.000000	38.000000
75%	6.000000	140.250000	80.000000	33.000000	130.000000
max	17.000000	199.000000	122.000000	60.000000	846.000000

```
In [35]: df.isnull().sum()
#check missing value
```

```
Out[35]: Pregnancies      1
Glucose      0
BloodPressure      5
SkinThickness    120
Insulin      13
BMI      2
DiabetesPedigreeFunction      0
Age      0
Outcome      0
dtype: int64
```

```
In [36]: missing_percentage = df.isnull().mean()*100
columns_with_missing_30 = missing_percentage[missing_percentage > 30]
print(columns_with_missing_30)
```

```
Series([], dtype: float64)
```

```
In [37]: missing_percentage = df.isnull().mean()*100
columns_with_missing_30 = missing_percentage[missing_percentage > 20]
print(columns_with_missing_30)
```

Series([], dtype: float64)

```
In [38]: missing_percentage = df.isnull().mean()*100
columns_with_missing_30 = missing_percentage[missing_percentage > 10]
print(columns_with_missing_30)
```

SkinThickness 15.625
dtype: float64

```
In [39]: missing_percentage = df.isnull().mean()*100
columns_with_missing_30 = missing_percentage[missing_percentage > 20]
print(columns_with_missing_30)
```

Series([], dtype: float64)

```
In [43]: import pandas as pd
df = pd.read_csv('diabetes.csv')
df.head()
```

```
Out[43]:
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Diabe
0	6.0	148	72.0	35.0	NaN	33.6	
1	1.0	85	66.0	29.0	NaN	26.6	
2	8.0	183	64.0	NaN	NaN	23.3	
3	1.0	89	66.0	23.0	94.0	28.1	
4	0.0	137	40.0	35.0	168.0	43.1	

```
In [44]: df.isnull().sum()
#check missing value
```

```
Out[44]: Pregnancies      1
Glucose      0
BloodPressure  5
SkinThickness 286
Insulin      13
BMI          2
DiabetesPedigreeFunction  0
Age          0
Outcome      0
dtype: int64
```

```
In [45]: missing_percentage = df.isnull().mean()*100
columns_with_missing_30 = missing_percentage[missing_percentage > 30]
print(columns_with_missing_30)
```

SkinThickness 37.239583
dtype: float64

```
In [46]: df.describe()
```

Out [46]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin
count	767.000000	768.000000	763.000000	482.000000	755.000000
mean	3.850065	120.894531	69.558322	21.717842	81.173510
std	3.368910	31.972618	18.589365	15.311840	115.751817
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	1.000000	99.000000	64.000000	8.000000	0.000000
50%	3.000000	117.000000	72.000000	24.000000	38.000000
75%	6.000000	140.250000	80.000000	33.000000	130.000000
max	17.000000	199.000000	122.000000	60.000000	846.000000

```
In [49]: missing_percentage = df.isnull().mean() * 100
threshold = 30
columns_to_drop = missing_percentage[missing_percentage > threshold]
df_cleaned = df.drop(columns=columns_to_drop)
print(f"columns dropped: {list(columns_to_drop)}")
print(f"remaining columns: {df_cleaned.columns}")
```

```
columns dropped: ['SkinThickness']
remaining columns: Index(['Pregnancies', 'Glucose', 'BloodPressure',
'Insulin', 'BMI',
'DiabetesPedigreeFunction', 'Age', 'Outcome'],
dtype='str')
```

```
In [50]: df_cleaned.describe()
```

Out [50]:

	Pregnancies	Glucose	BloodPressure	Insulin	BMI	DiabetesPedigreeFunction
count	767.000000	768.000000	763.000000	755.000000	766.000000	767.000000
mean	3.850065	120.894531	69.558322	81.173510	32.07611	0.332481
std	3.368910	31.972618	18.589365	115.751817	7.72267	0.332481
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	1.000000	99.000000	64.000000	0.000000	27.32500	0.000000
50%	3.000000	117.000000	72.000000	38.000000	32.00000	0.000000
75%	6.000000	140.250000	80.000000	130.000000	36.60000	0.000000
max	17.000000	199.000000	122.000000	846.000000	67.10000	0.671000

```
In [51]: df_cleaned.isnull().sum()
```

```
Out[51]: Pregnancies      1
         Glucose          0
         BloodPressure    5
         Insulin          13
         BMI              2
         DiabetesPedigreeFunction  0
         Age              0
         Outcome          0
         dtype: int64
```

```
In [52]: pip install numpy
```

Requirement already satisfied: numpy in /opt/anaconda3/envs/code1/lib/python3.13/site-packages (2.4.3)
Note: you may need to restart the kernel to use updated packages.

```
In [53]: pip install seaborn
```

```
Collecting seaborn
  Downloading seaborn-0.13.2-py3-none-any.whl.metadata (5.4 kB)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in /opt/anaconda3/envs/code1/lib/python3.13/site-packages (from seaborn) (2.4.3)
Requirement already satisfied: pandas>=1.2 in /opt/anaconda3/envs/code1/lib/python3.13/site-packages (from seaborn) (3.0.1)
Collecting matplotlib!=3.6.1,>=3.4 (from seaborn)
  Downloading matplotlib-3.10.8-cp313-cp313-macosx_11_0_arm64.whl.metadata (52 kB)
Collecting contourpy>=1.0.1 (from matplotlib!=3.6.1,>=3.4->seaborn)
  Downloading contourpy-1.3.3-cp313-cp313-macosx_11_0_arm64.whl.metadata (5.5 kB)
Collecting cycler>=0.10 (from matplotlib!=3.6.1,>=3.4->seaborn)
  Downloading cycler-0.12.1-py3-none-any.whl.metadata (3.8 kB)
Collecting fonttools>=4.22.0 (from matplotlib!=3.6.1,>=3.4->seaborn)
  Downloading fonttools-4.62.1-cp313-cp313-macosx_10_13_universal2.whl.metadata (117 kB)
Collecting kiwisolver>=1.3.1 (from matplotlib!=3.6.1,>=3.4->seaborn)
  Downloading kiwisolver-1.5.0-cp313-cp313-macosx_11_0_arm64.whl.metadata (5.1 kB)
Requirement already satisfied: packaging>=20.0 in /opt/anaconda3/envs/code1/lib/python3.13/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (25.0)
Collecting pillow>=8 (from matplotlib!=3.6.1,>=3.4->seaborn)
  Downloading pillow-12.1.1-cp313-cp313-macosx_11_0_arm64.whl.metadata (8.8 kB)
Collecting pyparsing>=3 (from matplotlib!=3.6.1,>=3.4->seaborn)
  Downloading pyparsing-3.3.2-py3-none-any.whl.metadata (5.8 kB)
Requirement already satisfied: python-dateutil>=2.7 in /opt/anaconda3/envs/code1/lib/python3.13/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in /opt/anaconda3/envs/code1/lib/python3.13/site-packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.17.0)
Downloading seaborn-0.13.2-py3-none-any.whl (294 kB)
Downloading matplotlib-3.10.8-cp313-cp313-macosx_11_0_arm64.whl (8.1 MB)
```

8.1/8.1 MB 666.1 kB/s
0:00:12 663.1 kB/s eta 0:00:01

Downloading contourpy-1.3.3-cp313-cp313-macosx_11_0_arm64.whl (274 kB)
 Downloading cycler-0.12.1-py3-none-any.whl (8.3 kB)
 Downloading fonttools-4.62.1-cp313-cp313-macosx_10_13_universal2.whl (2.9 MB)

2.9/2.9 MB 470.4 kB/s
 0:00:05476.9 kB/s eta 0:00:01

Downloading kiwisolver-1.5.0-cp313-cp313-macosx_11_0_arm64.whl (64 kB)

Downloading pillow-12.1.1-cp313-cp313-macosx_11_0_arm64.whl (4.7 MB)
 4.7/4.7 MB 607.8 kB/s

0:00:075.5 kB/s eta 0:00:01:02

Downloading pyparsing-3.3.2-py3-none-any.whl (122 kB)

Installing collected packages: pyparsing, pillow, kiwisolver, fonttools, cycler, contourpy, matplotlib, seaborn

8/8 [seaborn] — 6/8 [matplotlib]

Successfully installed contourpy-1.3.3 cycler-0.12.1 fonttools-4.62.1 kiwisolver-1.5.0 matplotlib-3.10.8 pillow-12.1.1 pyparsing-3.3.2 seaborn-0.13.2

Note: you may need to restart the kernel to use updated packages.

```
In [54]: df_cleaned['Pregnancies'] = df_cleaned['Pregnancies'].fillna(df_cleaned['Pregnancies'].median())
```

```
In [55]: df_cleaned.isnull().sum()
```

```
Out[55]: Pregnancies      0
         Glucose         0
         BloodPressure    5
         Insulin        13
         BMI             2
         DiabetesPedigreeFunction  0
         Age             0
         Outcome         0
         dtype: int64
```

```
In [56]: df_cleaned['BloodPressure'] = df_cleaned['BloodPressure'].fillna(df_cleaned['BloodPressure'].median())
```

```
In [57]: df_cleaned['Insulin'] = df_cleaned['Insulin'].fillna(df_cleaned['Insulin'].median())
```

```
In [58]: df_cleaned['BMI'] = df_cleaned['BMI'].fillna(df_cleaned['BMI'].median())
```

```
In [59]: df_cleaned.isnull().sum()
```

```
Out[59]: Pregnancies      0
         Glucose         0
         BloodPressure    0
         Insulin         0
         BMI             0
         DiabetesPedigreeFunction  0
         Age             0
         Outcome         0
         dtype: int64
```

```
In [60]: df_cleaned.head()
```

Out[60]:

	Pregnancies	Glucose	BloodPressure	Insulin	BMI	DiabetesPedigreeFunc
0	6.0	148	72.0	38.0	33.6	(
1	1.0	85	66.0	38.0	26.6	(
2	8.0	183	64.0	38.0	23.3	(
3	1.0	89	66.0	94.0	28.1	(
4	0.0	137	40.0	168.0	43.1	2

In [60]: `df_cleaned.head()`

Out[60]:

	Pregnancies	Glucose	BloodPressure	Insulin	BMI	DiabetesPedigreeFunc
0	6.0	148	72.0	38.0	33.6	(
1	1.0	85	66.0	38.0	26.6	(
2	8.0	183	64.0	38.0	23.3	(
3	1.0	89	66.0	94.0	28.1	(
4	0.0	137	40.0	168.0	43.1	2

In [65]:

```
import numpy as np
data = df_cleaned.copy()
def detect_outliers_iqr(columns):
    q1 = columns.quantile(0.25)
    q3 = columns.quantile(0.75)
    iqr= q3-q1
    lower_bound = q1 - 1.5 * iqr
    upper_bound = q3 + 1.5 * iqr
    return (columns < lower_bound) | (columns > upper_bound)

for col in data.select_dtypes(include=np.number).columns:
    data[f'{col}_Outlier'] = detect_outliers_iqr(data[col])

outliers_only = data[data.filter(like='_Outlier').any(axis=1)]
outliers_only
```

Out[65]:

	Pregnancies	Glucose	BloodPressure	Insulin	BMI	DiabetesPedigreeFi
4	0.0	137	40.0	168.0	43.1	
7	10.0	115	0.0	38.0	35.3	
8	2.0	197	70.0	543.0	30.5	
9	8.0	125	96.0	0.0	0.0	
12	10.0	139	80.0	0.0	27.1	
...	...	...	...	...	...	
697	0.0	99	0.0	0.0	25.0	
707	2.0	127	46.0	335.0	34.4	
710	3.0	158	64.0	387.0	31.2	
715	7.0	187	50.0	392.0	33.9	
753	0.0	181	88.0	510.0	43.3	

128 rows x 16 columns

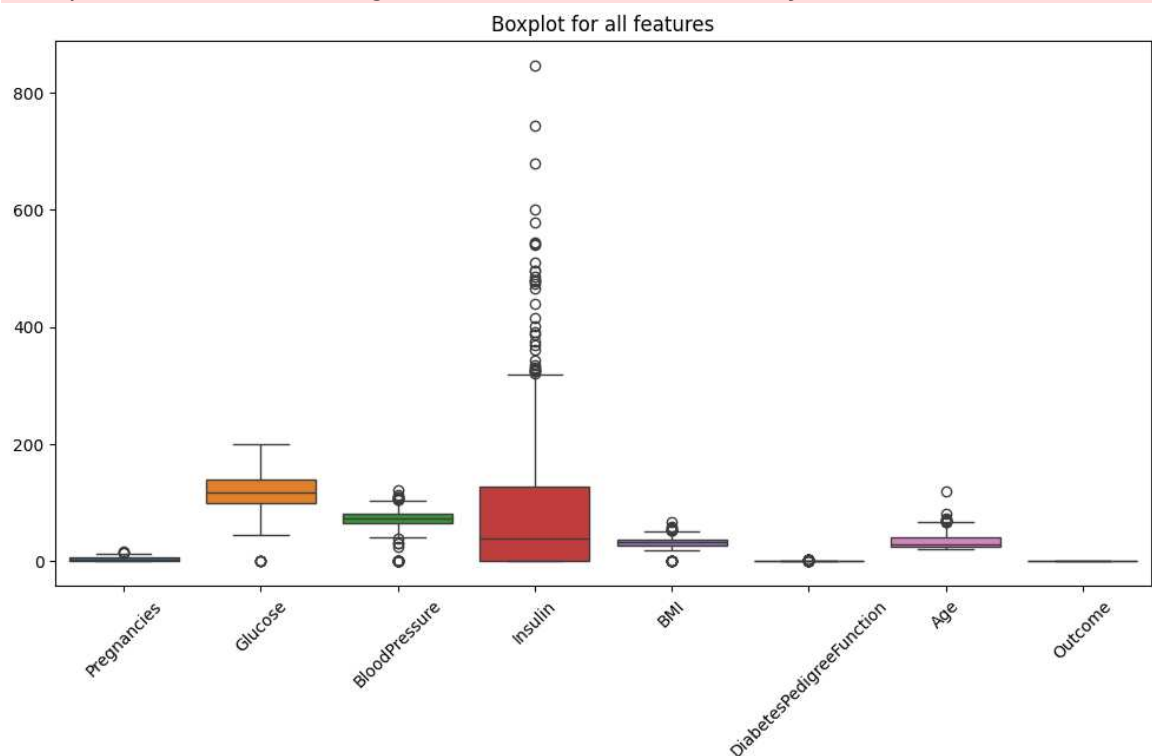
```

In [66]: import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(12,6))
sns.boxplot(data=df_cleaned)
plt.xticks(rotation=45)
plt.title("Boxplot for all features")
plt.show()

```

Matplotlib is building the font cache; this may take a moment.



```
In [69]: import numpy as np

for columns in df_cleaned.select_dtypes(include='number').columns:

    Q1 = df_cleaned[columns].quantile(0.25)
    Q3 = df_cleaned[columns].quantile(0.75)
    IQR = Q3 - Q1

    lower = Q1 - 1.5 * IQR
    upper = Q3 + 1.5 * IQR

    df_cleaned[columns] = np.where(
        (df_cleaned[columns] < lower) | (df_cleaned[columns] > upper),
        np.nan,
        df_cleaned[columns]
    )
```

```
In [70]: df_cleaned.isnull().sum()
```

```
Out[70]: Pregnancies      4
         Glucose          5
         BloodPressure    44
         Insulin          34
         BMI              17
         DiabetesPedigreeFunction  29
         Age              10
         Outcome          0
         dtype: int64
```

```
In [72]: missing_percentage = df_cleaned.isnull().mean()*100
         columns_with_missing_30 = missing_percentage[missing_percentage > 30]
         print(columns_with_missing_30)
```

```
Series([], dtype: float64)
```

```
In [74]: for columns in df_cleaned.select_dtypes(include='number').columns:
         df_cleaned[columns] = df_cleaned[columns].fillna(df_cleaned[columns].median())
```

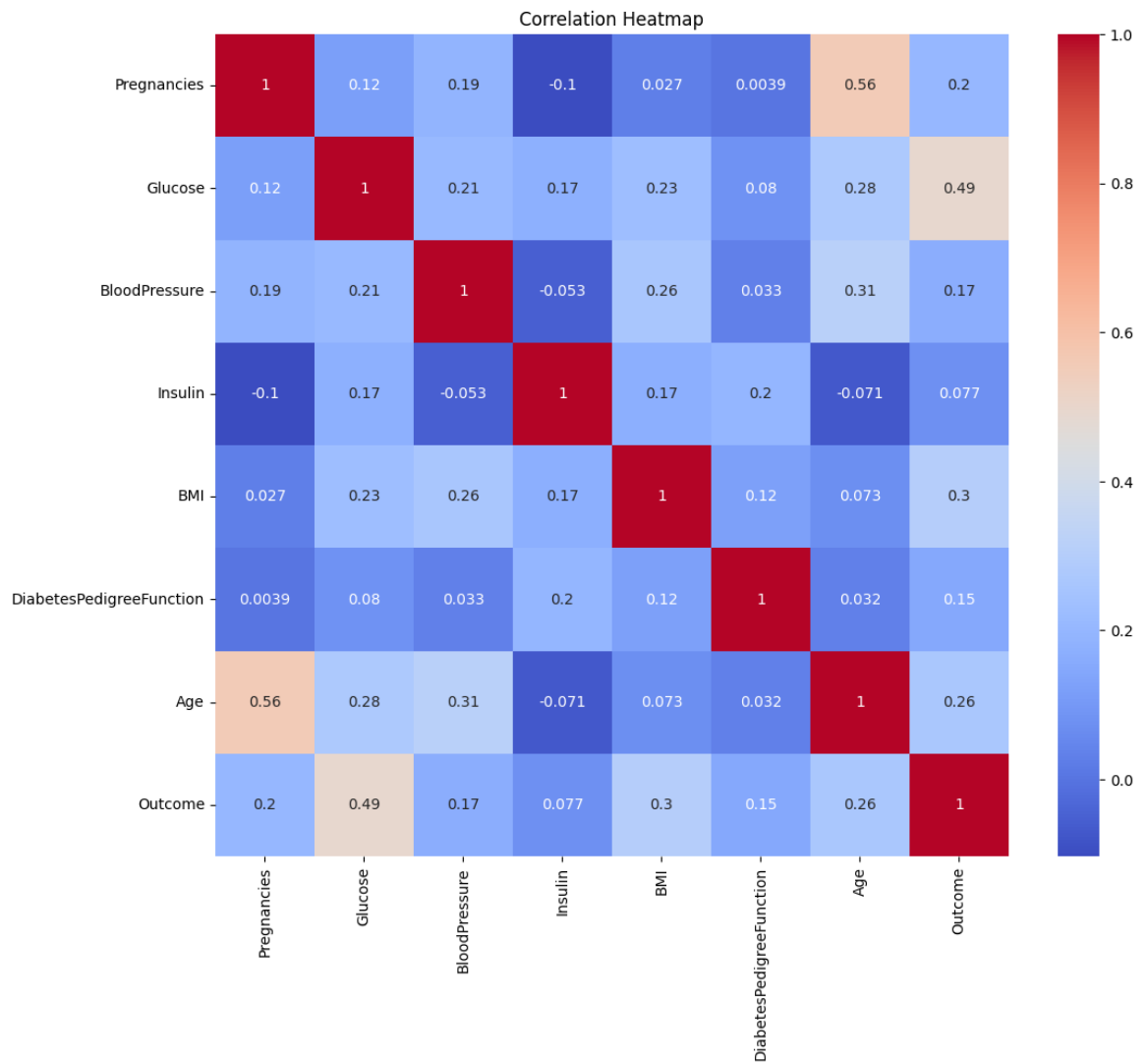
```
In [75]: df_cleaned.isnull().sum()
```

```
Out[75]: Pregnancies      0
         Glucose          0
         BloodPressure    0
         Insulin          0
         BMI              0
         DiabetesPedigreeFunction  0
         Age              0
         Outcome          0
         dtype: int64
```

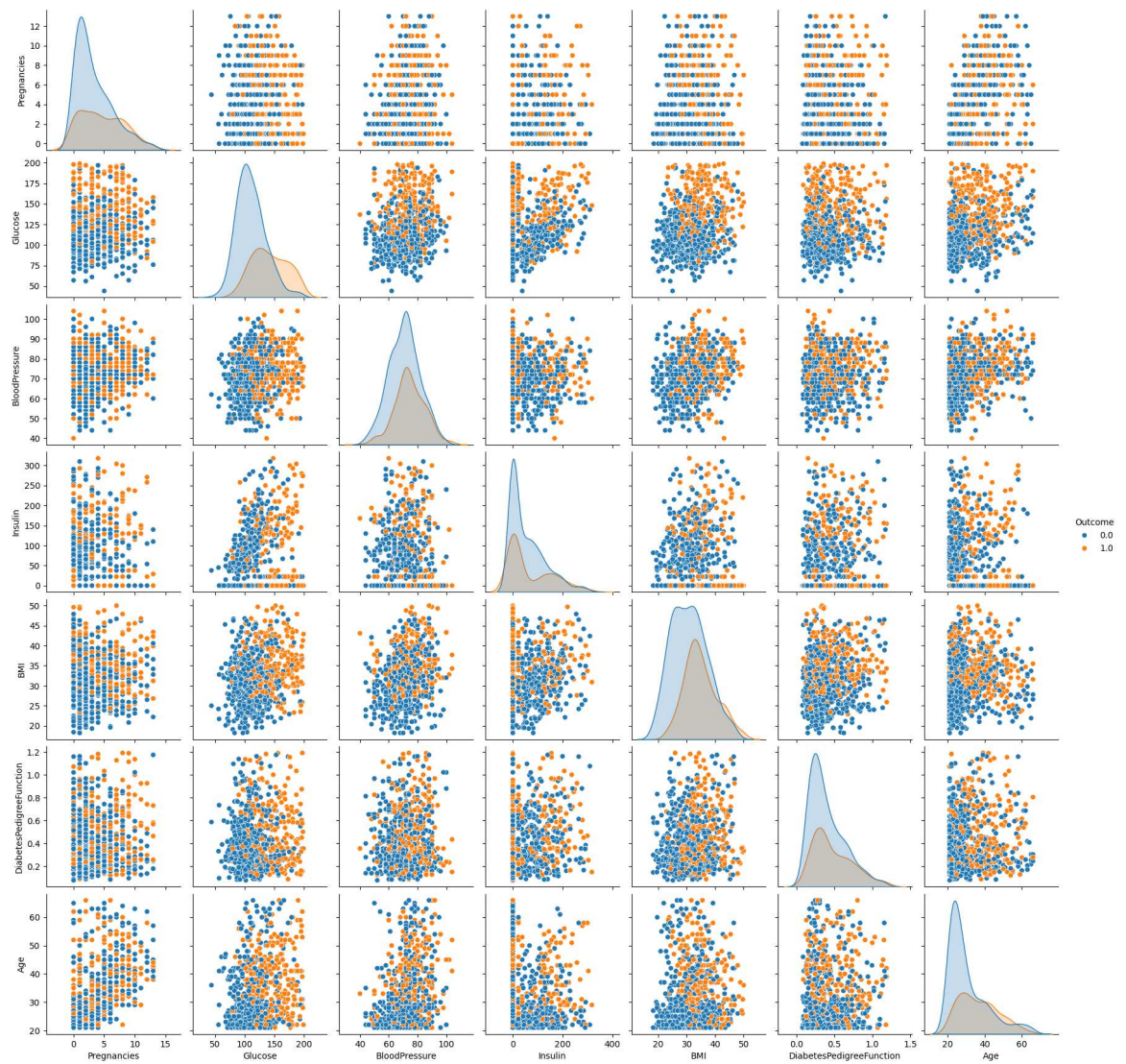
```
In [76]: import seaborn as sns
         import matplotlib.pyplot as plt

         plt.figure(figsize=(12,10))
         sns.heatmap(df_cleaned.corr(), annot=True, cmap='coolwarm')
```

```
plt.title("Correlation Heatmap")
plt.show()
```

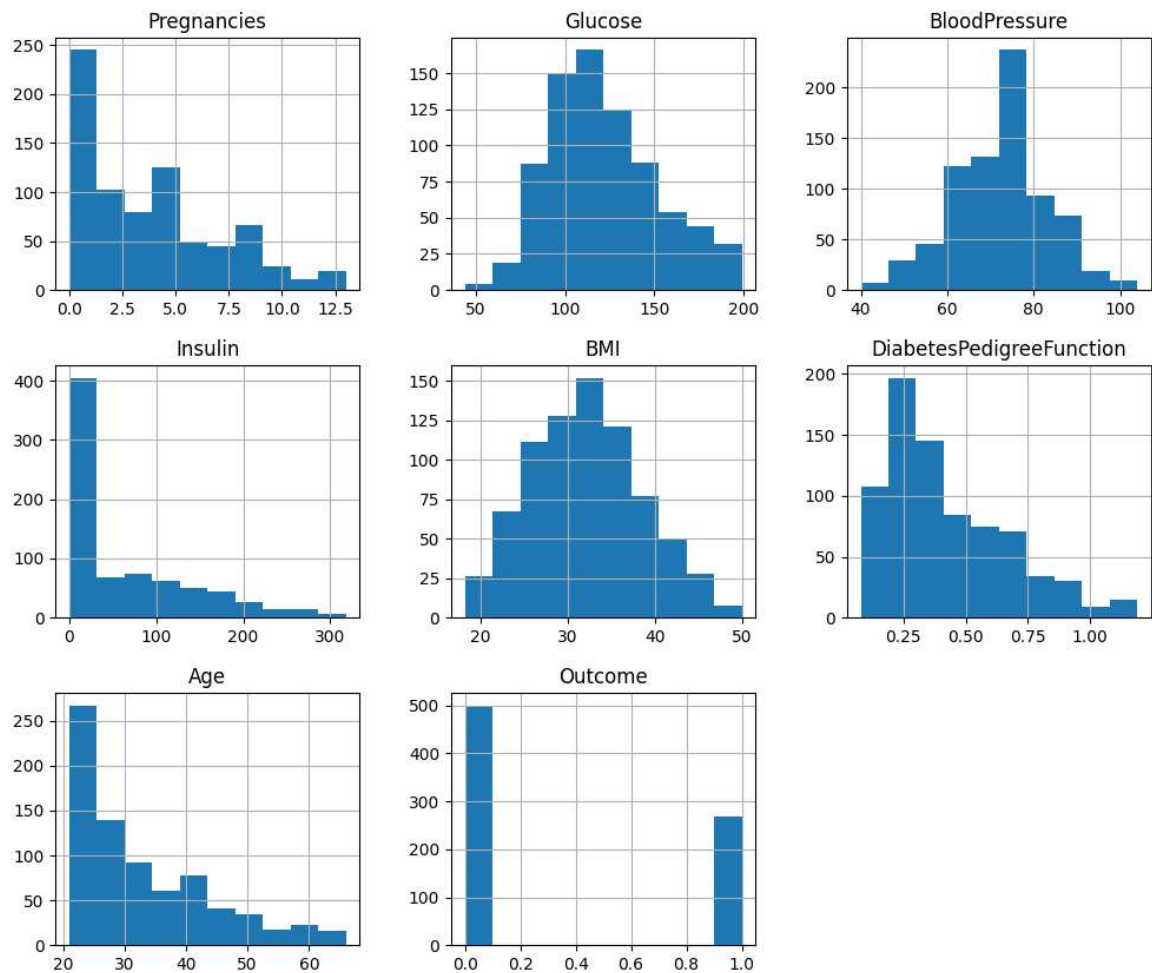


```
In [77]: sns.pairplot(df_cleaned, hue="Outcome")
plt.show()
```



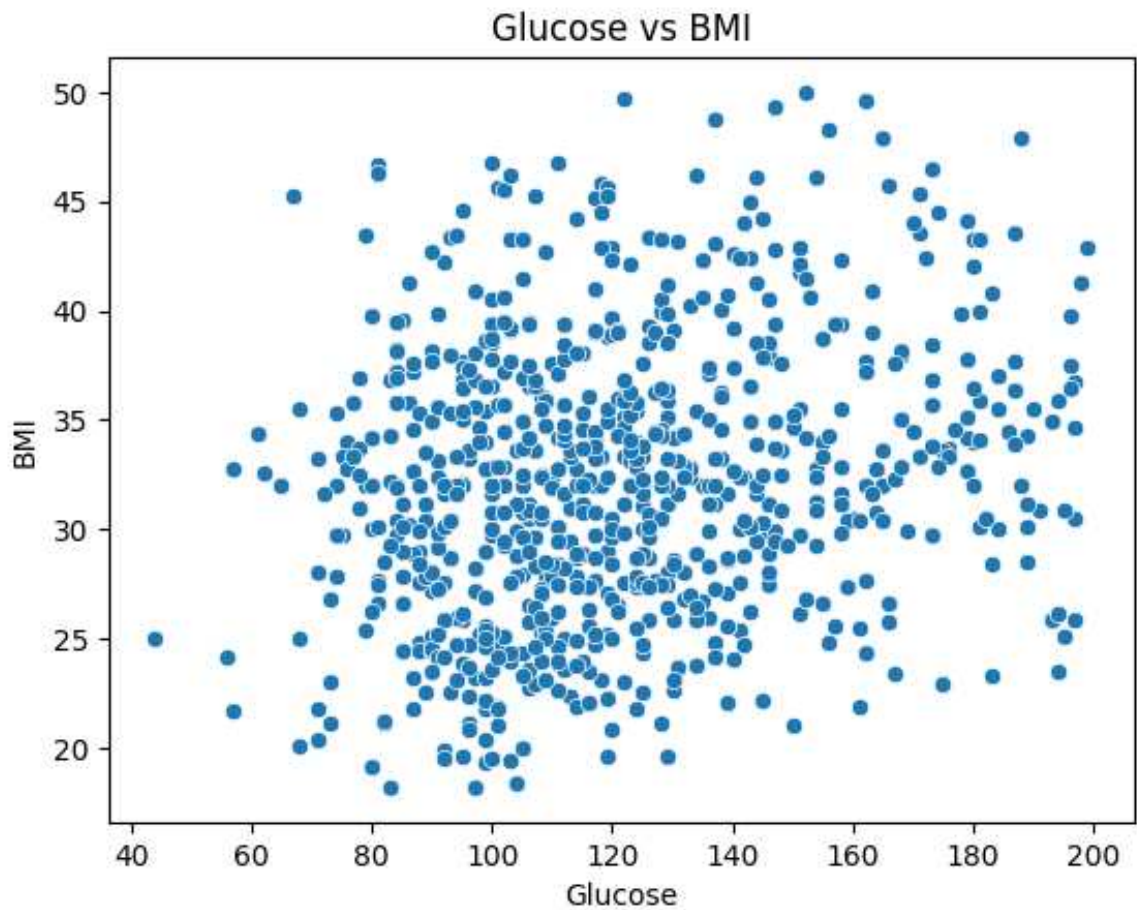
```
In [78]: import matplotlib.pyplot as plt

df_cleaned.hist(figsize=(12,10))
plt.show()
```

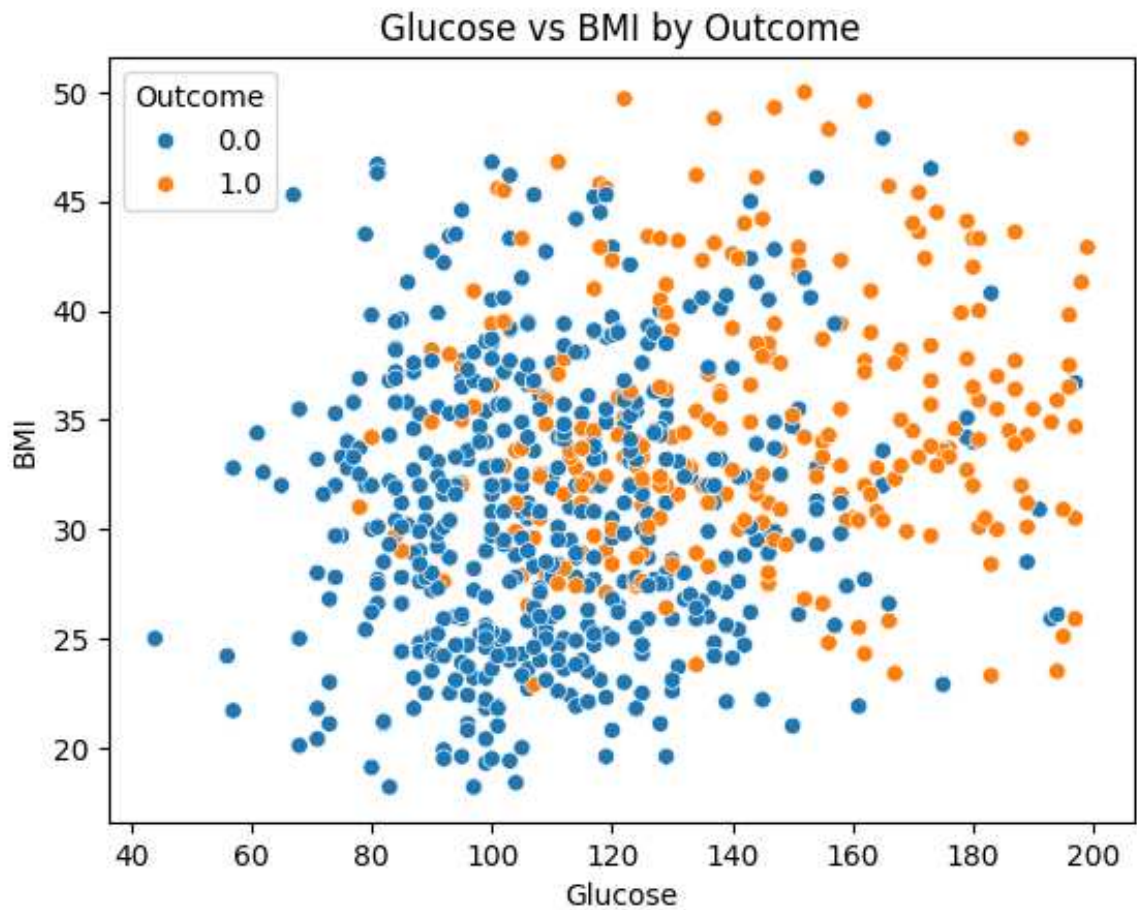


```
In [79]: import seaborn as sns
import matplotlib.pyplot as plt

sns.scatterplot(x="Glucose", y="BMI", data=df_cleaned)
plt.title("Glucose vs BMI")
plt.show()
```



```
In [80]: sns.scatterplot(x="Glucose", y="BMI", hue="Outcome", data=df_cleaned)
plt.title("Glucose vs BMI by Outcome")
plt.show()
```



In []: