In [1]:
```
#imprting libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
#loading the dataset
df=pd.read_csv('diabetes.csv')
df
```

Out[3]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows × 9 columns

In [4]:
```
df.shape
```

Out[4]:
```
(768, 9)
```

In [5]:
```
#Top 5 records
df.head()
```

Out[5]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1

3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

In [6]:
```
#Bottom 5 records
df.tail()
```

Out[6]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

In [7]:
```
# info about the data set
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]:
```
#statiscal describtion of the data set
df.describe()
```

Out[8]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	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	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951

min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

In [10]:
```
#check for null values
df.isnull().sum()
```

Out[10]:
```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

In [13]:
```
#replacing 0 values that are not logical with the mean or median
df['Glucose']=df['Glucose'].replace(0,df['Glucose'].mean())
df['BloodPressure']=df['BloodPressure'].replace(0,df['BloodPressure'].mean())
df['BMI']=df['BMI'].replace(0,df['BMI'].median())
df['SkinThickness']=df['SkinThickness'].replace(0,df['SkinThickness'].median())
df['Insulin']=df['Insulin'].replace(0,df['Insulin'].median())
```

In [14]:
```
#Top 5 records
df.head()
```

Out[14]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148.0	72.0	35	30.5	33.6	0.627	50	1
1	1	85.0	66.0	29	30.5	26.6	0.351	31	0
2	8	183.0	64.0	23	30.5	23.3	0.672	32	1
3	1	89.0	66.0	23	94.0	28.1	0.167	21	0
4	0	137.0	40.0	35	168.0	43.1	2.288	33	1

In [15]:
```
#Bottom 5 records
df.tail()
```

Out[15]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome

763	10	101.0	76.0	48	180.0	32.9	0.171	63	0
764	2	122.0	70.0	27	30.5	36.8	0.340	27	0
765	5	121.0	72.0	23	112.0	26.2	0.245	30	0
766	1	126.0	60.0	23	30.5	30.1	0.349	47	1
767	1	93.0	70.0	31	30.5	30.4	0.315	23	0

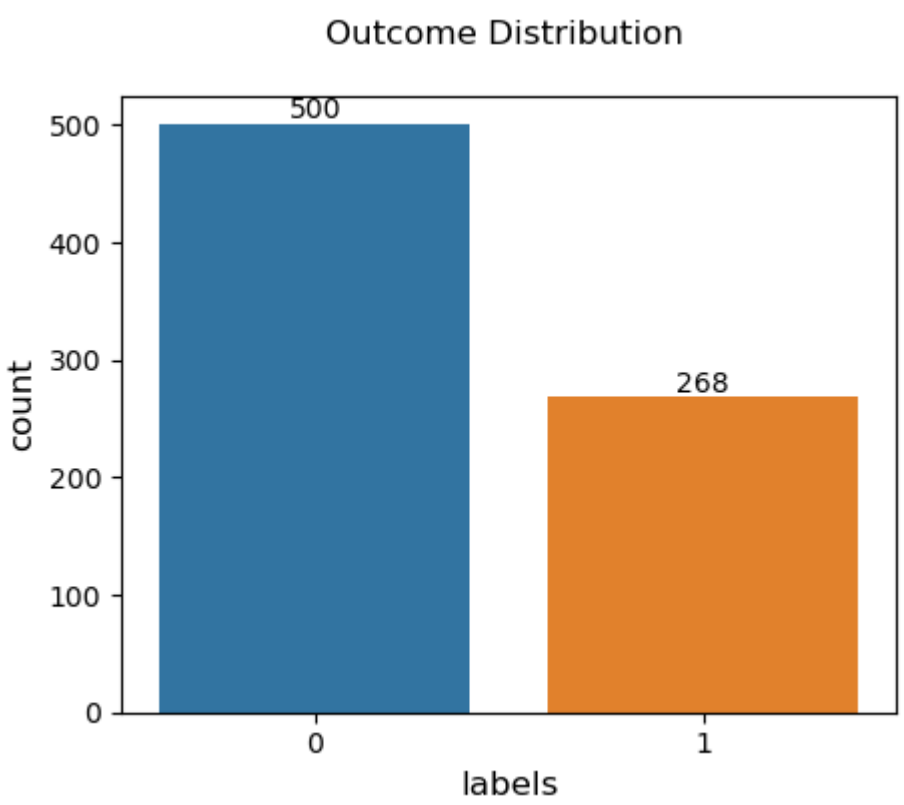
In [16]:
```
# check for duplicated values
df.duplicated().sum()
```

Out[16]:
```
0
```

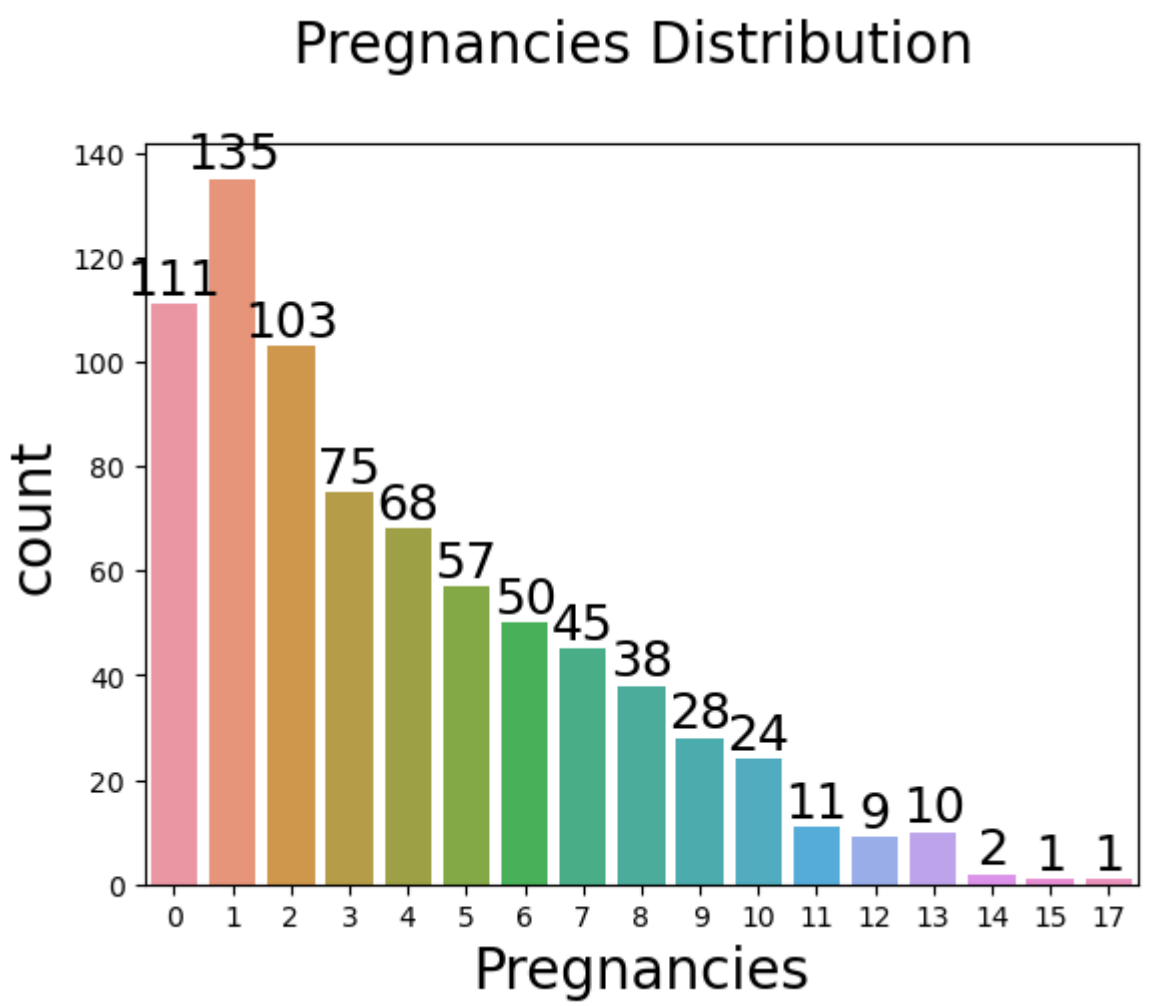
In [17]:
```
df.columns
```

Out[17]:
```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

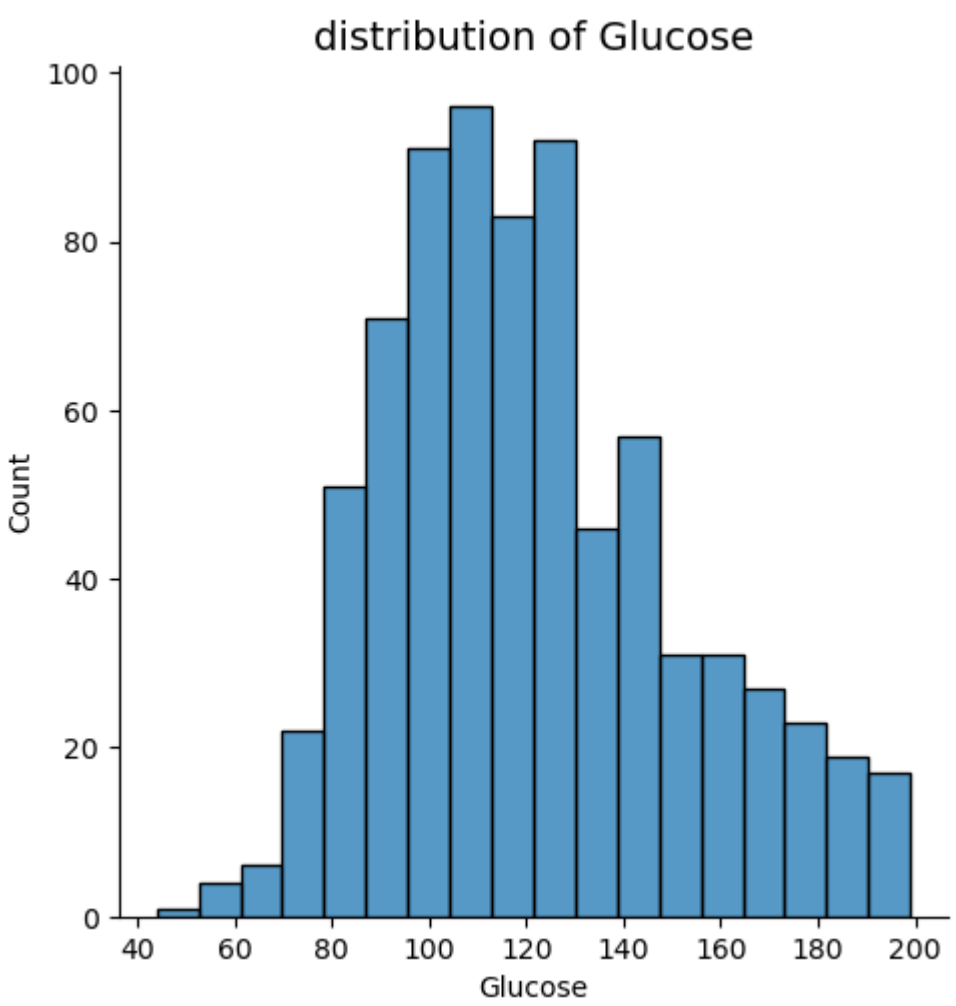
In [21]:
```
#check the distribution of outcome feature (0=non diabetic,1=diabetic)
plt.figure(figsize=(5,4))
ax=sns.barplot(x=df['Outcome'].value_counts().index,  y=df['Outcome'].value_counts())
for bars in ax.containers:
    ax.bar_label(bars)
plt.xlabel('labels', size = 12)
plt.ylabel('count', size = 12)
plt.title('Outcome Distribution \n',size = 12)
plt.show()
```



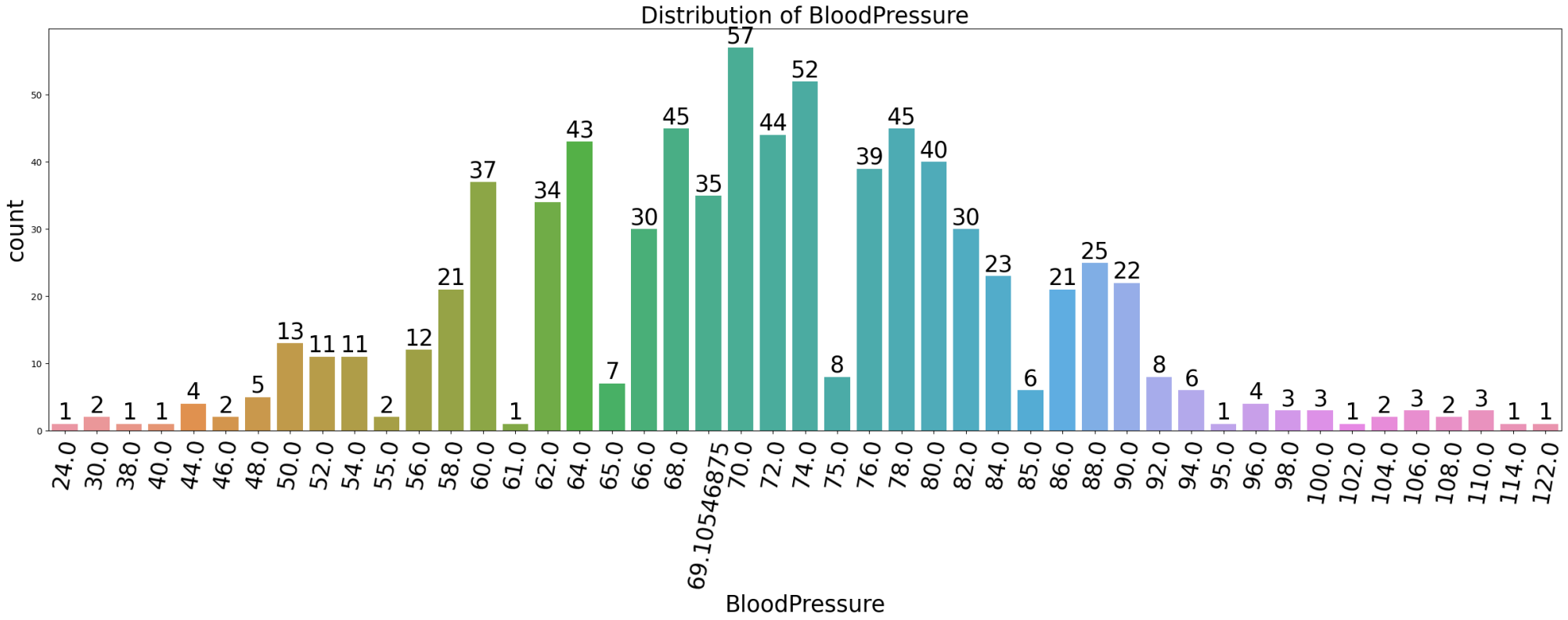
In [22]:
```
#Pregnancies Distribution
ax=sns.barplot(x=df['Pregnancies'].value_counts().index,  y=df['Pregnancies'].value_counts())
for bars in ax.containers:
    ax.bar_label(bars,size=18)
plt.xlabel('Pregnancies', size = 20)
plt.ylabel('count', size = 20)
plt.title('Pregnancies Distribution \n',size = 20)
plt.show()
```



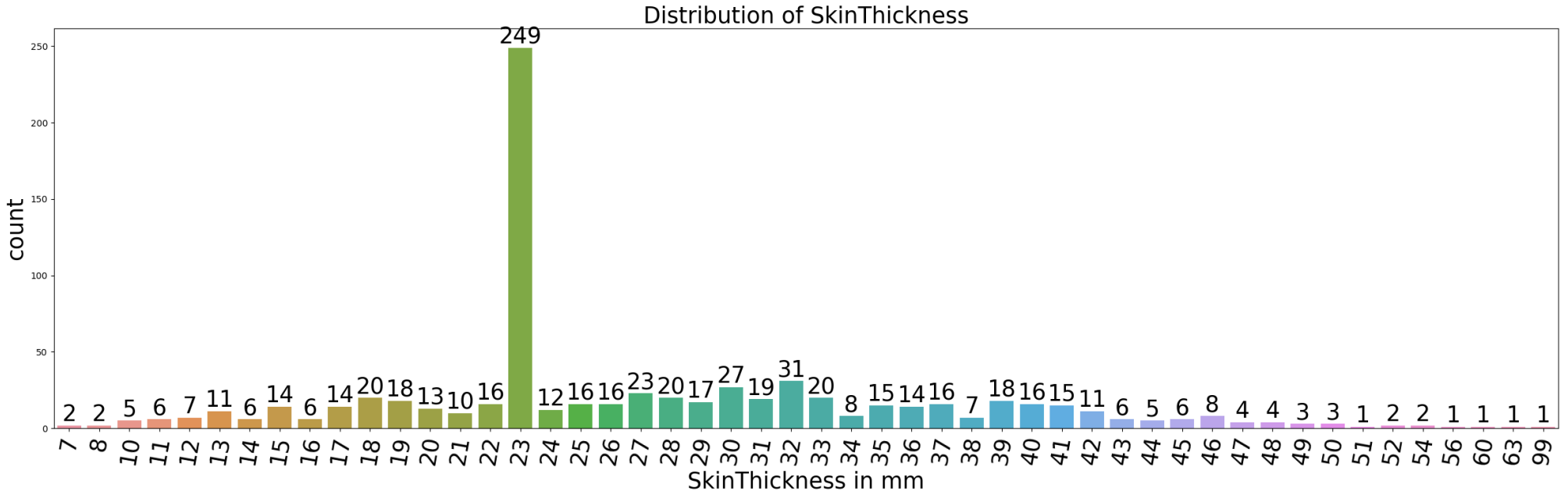
In [23]:
```
#Distibution of Glucose
sns.displot(df,x='Glucose')
plt.title('distribution of Glucose',size = 14)
plt.show()
```



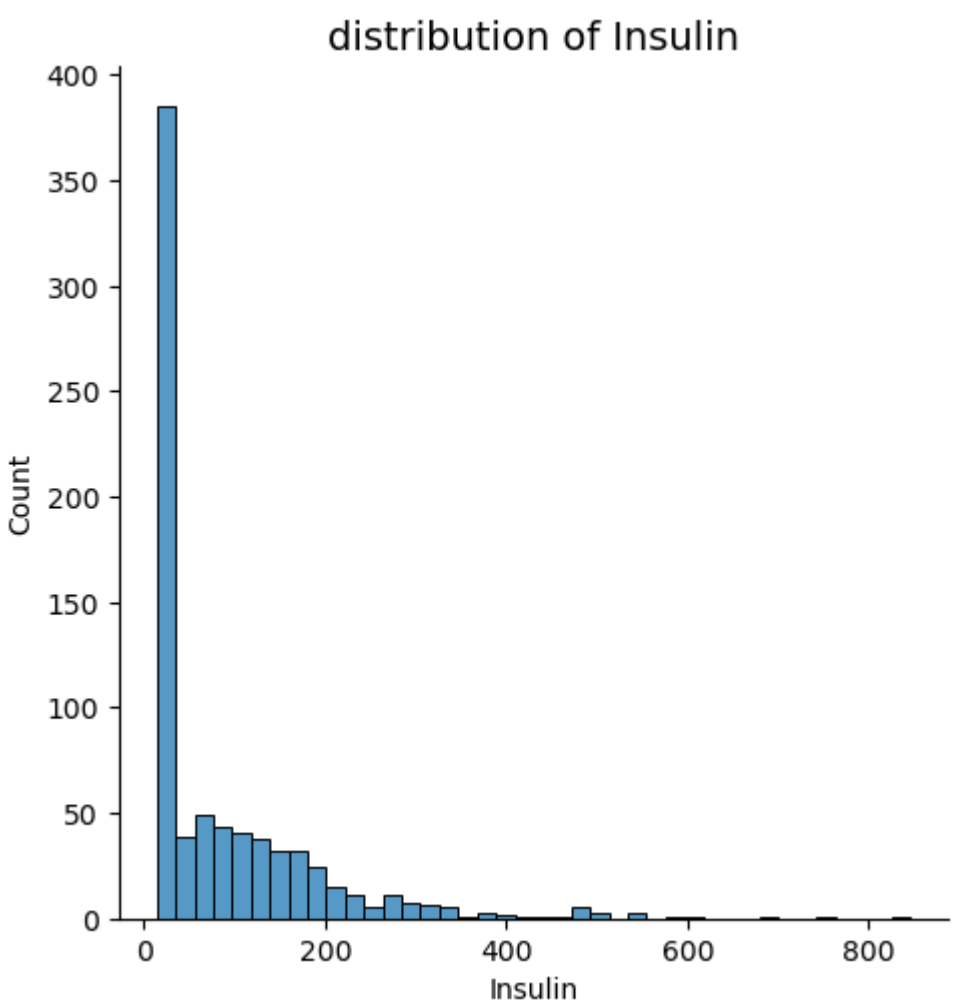
In [24]:
```
#Distribution of blood pressure
plt.figure(figsize=(30,8))
ax=sns.countplot(data=df, x=df.BloodPressure)
for bars in ax.containers:
    ax.bar_label(bars,size=25)
plt.xlabel('BloodPressure', size = 25)
plt.ylabel('count', size = 25)
plt.title('Distribution of BloodPressure',size = 25)
plt.xticks(rotation =80,size=25)
plt.show()
```



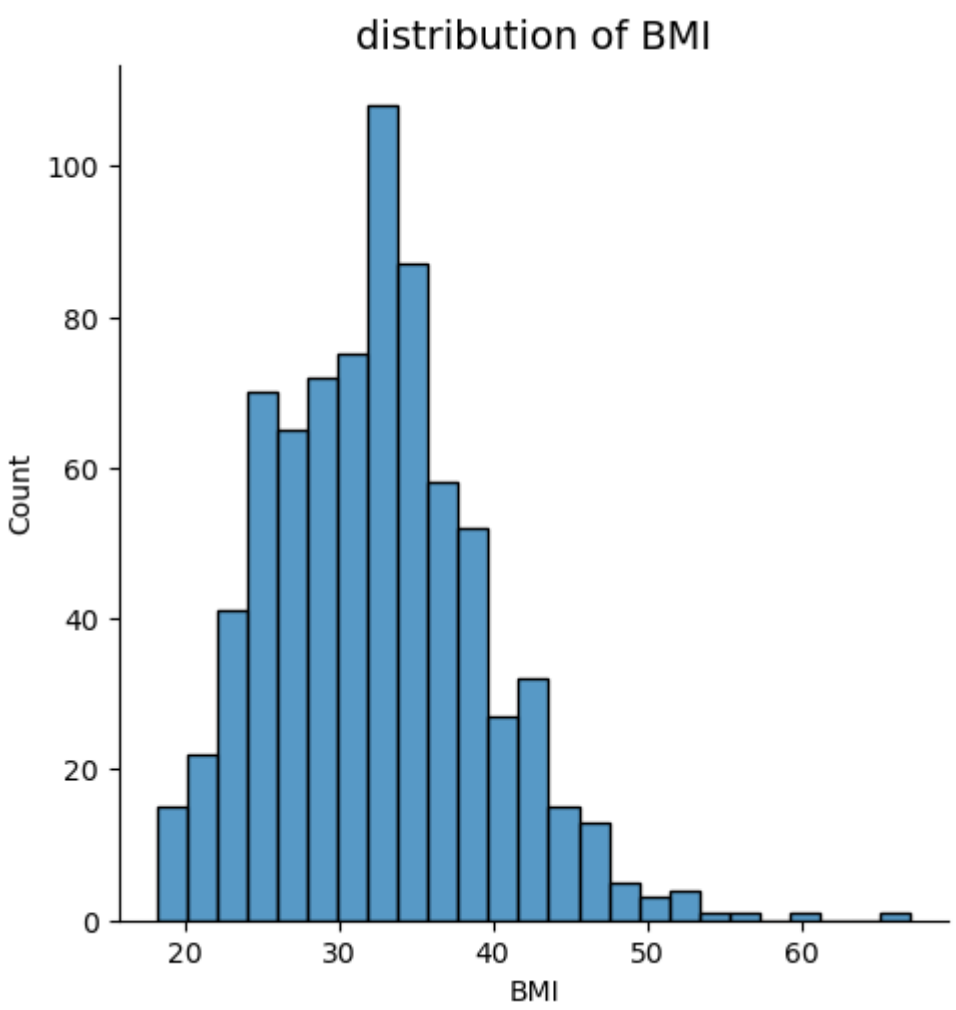
In [25]:
```
#Distribution of Skin thickness
plt.figure(figsize=(30,8))
ax=sns.countplot(data=df, x=df.SkinThickness)
for bars in ax.containers:
    ax.bar_label(bars,size=25)
plt.xlabel('SkinThickness in mm', size = 25)
plt.ylabel('count', size = 25)
plt.title('Distribution of SkinThickness',size = 25)
plt.xticks(rotation =80,size=25)
plt.show()
```



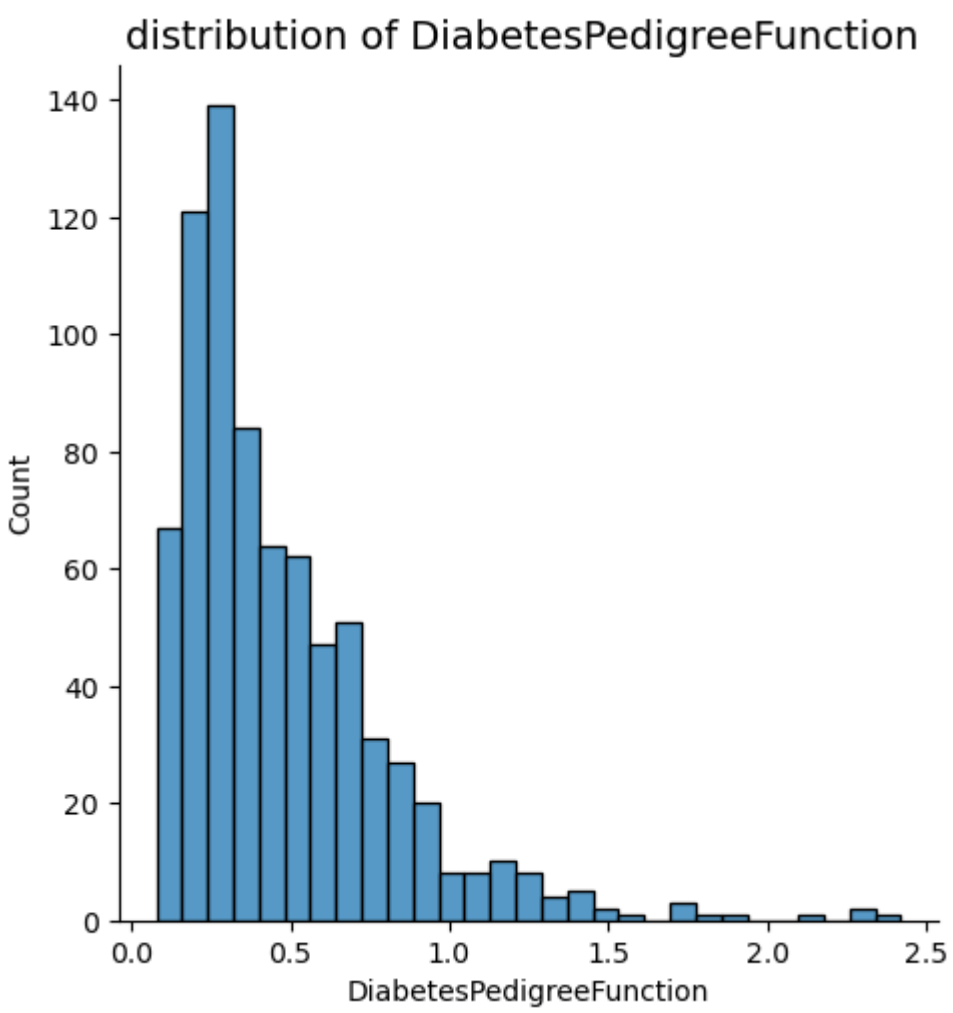
In [28]:
```
#Distribution of Insulin
sns.displot(df,x='Insulin')
plt.title('distribution of Insulin',size = 14)
plt.show()
```



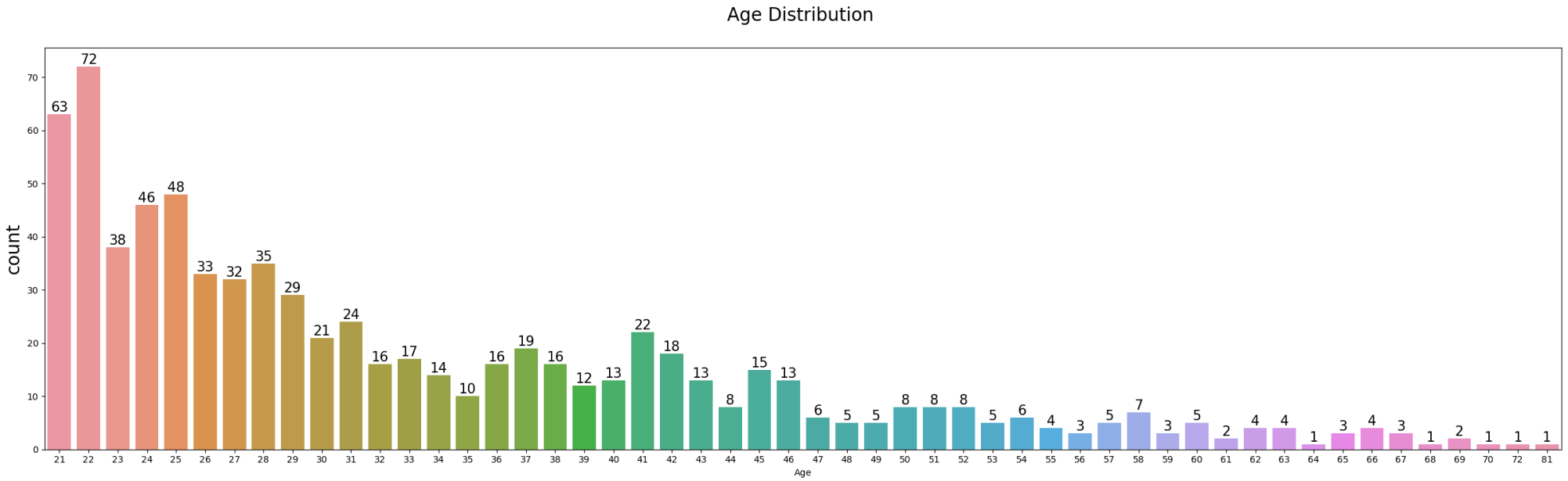
In [29]:
```
#Distribution of BMI
sns.displot(df,x='BMI')
plt.title('distribution of BMI',size = 14)
plt.show()
```



In [31]:
```
#Distribution  of DiabetesPedigreeFunction
sns.displot(df,x='DiabetesPedigreeFunction')
plt.title('distribution of DiabetesPedigreeFunction ',size = 14)
plt.show()
```

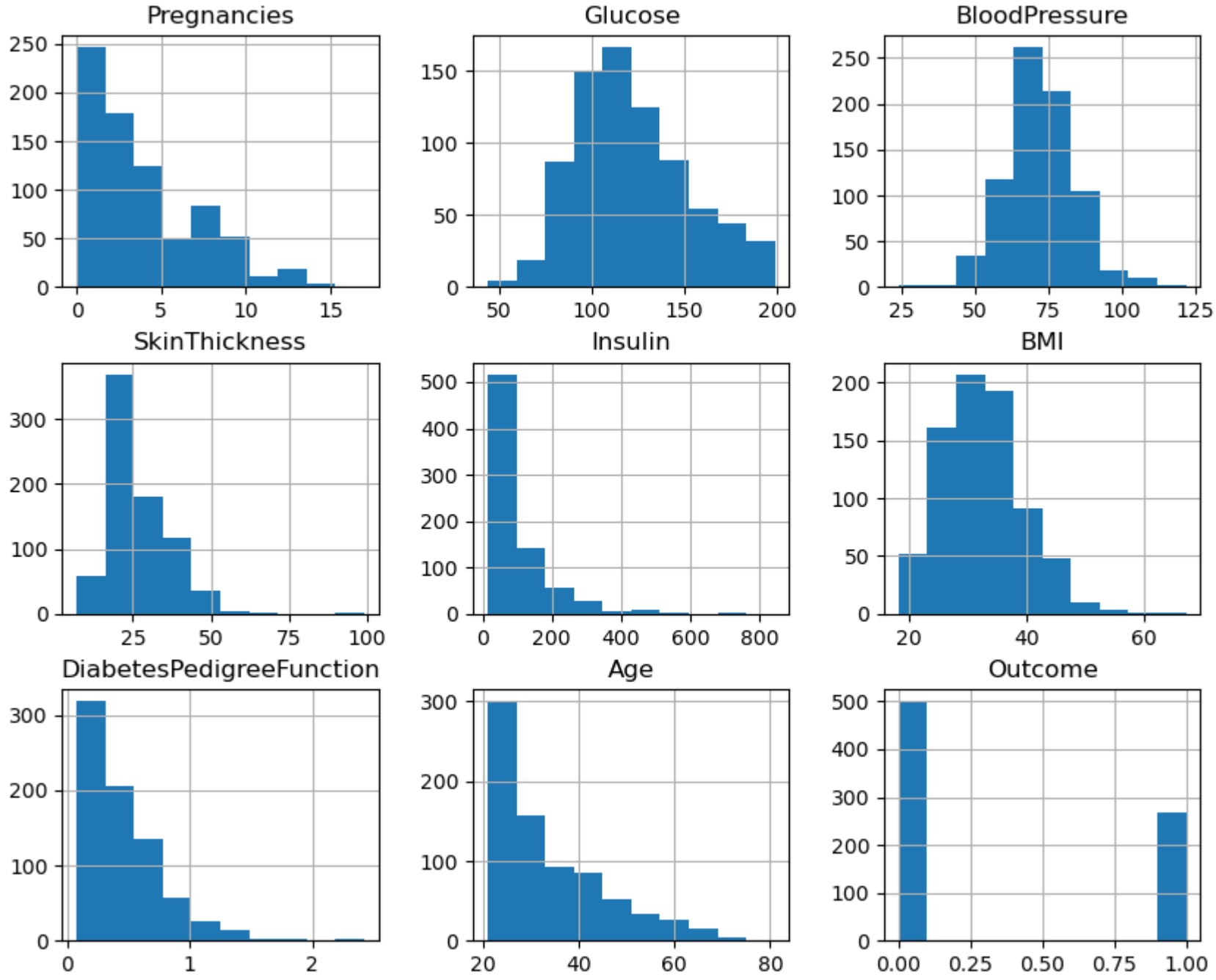


In [41]:
```
#Distribution of Age
plt.figure(figsize=(30,8))
ax=sns.barplot(x=df['Age'].value_counts().index,  y=df['Age'].value_counts())
for bars in ax.containers:
    ax.bar_label(bars,size=15)
plt.xlabel('Age', size = 10)
plt.ylabel('count', size = 20)
plt.title('Age Distribution \n',size = 20)
plt.show()
```



In [40]:
```
#Histogram of the entire dataframe
df.hist(figsize=(10,8))
```

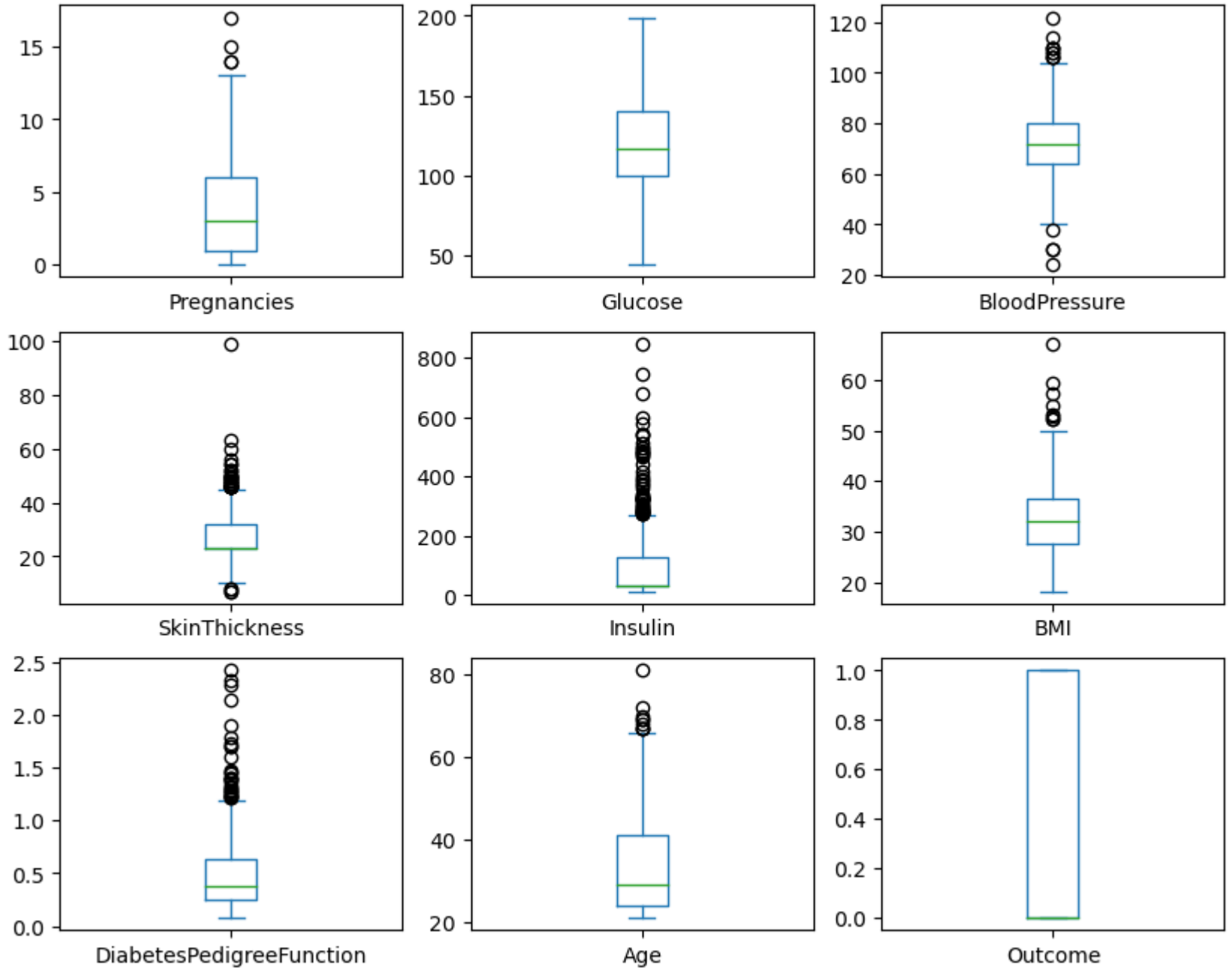
Out[40]:
```
array([[<Axes: title={'center': 'Pregnancies'}>,
        <Axes: title={'center': 'Glucose'}>,
        <Axes: title={'center': 'BloodPressure'}>],
       [<Axes: title={'center': 'SkinThickness'}>,
        <Axes: title={'center': 'Insulin'}>,
        <Axes: title={'center': 'BMI'}>],
       [<Axes: title={'center': 'DiabetesPedigreeFunction'}>,
        <Axes: title={'center': 'Age'}>,
        <Axes: title={'center': 'Outcome'}>]], dtype=object)
```



In [44]:
```
#check for outliers
df.plot(kind='box', subplots = True, layout = (3,3), sharex = False, sharey = False,figsize=(10,8))
```

Out[44]:
```
Pregnancies                 Axes(0.125,0.653529;0.227941x0.226471)
Glucose                  Axes(0.398529,0.653529;0.227941x0.226471)
BloodPressure            Axes(0.672059,0.653529;0.227941x0.226471)
```

```
SkinThickness                Axes(0.125,0.381765;0.227941x0.226471)
Insulin                    Axes(0.398529,0.381765;0.227941x0.226471)
BMI                        Axes(0.672059,0.381765;0.227941x0.226471)
DiabetesPedigreeFunction         Axes(0.125,0.11;0.227941x0.226471)
Age                        Axes(0.398529,0.11;0.227941x0.226471)
Outcome                    Axes(0.672059,0.11;0.227941x0.226471)
dtype: object
```



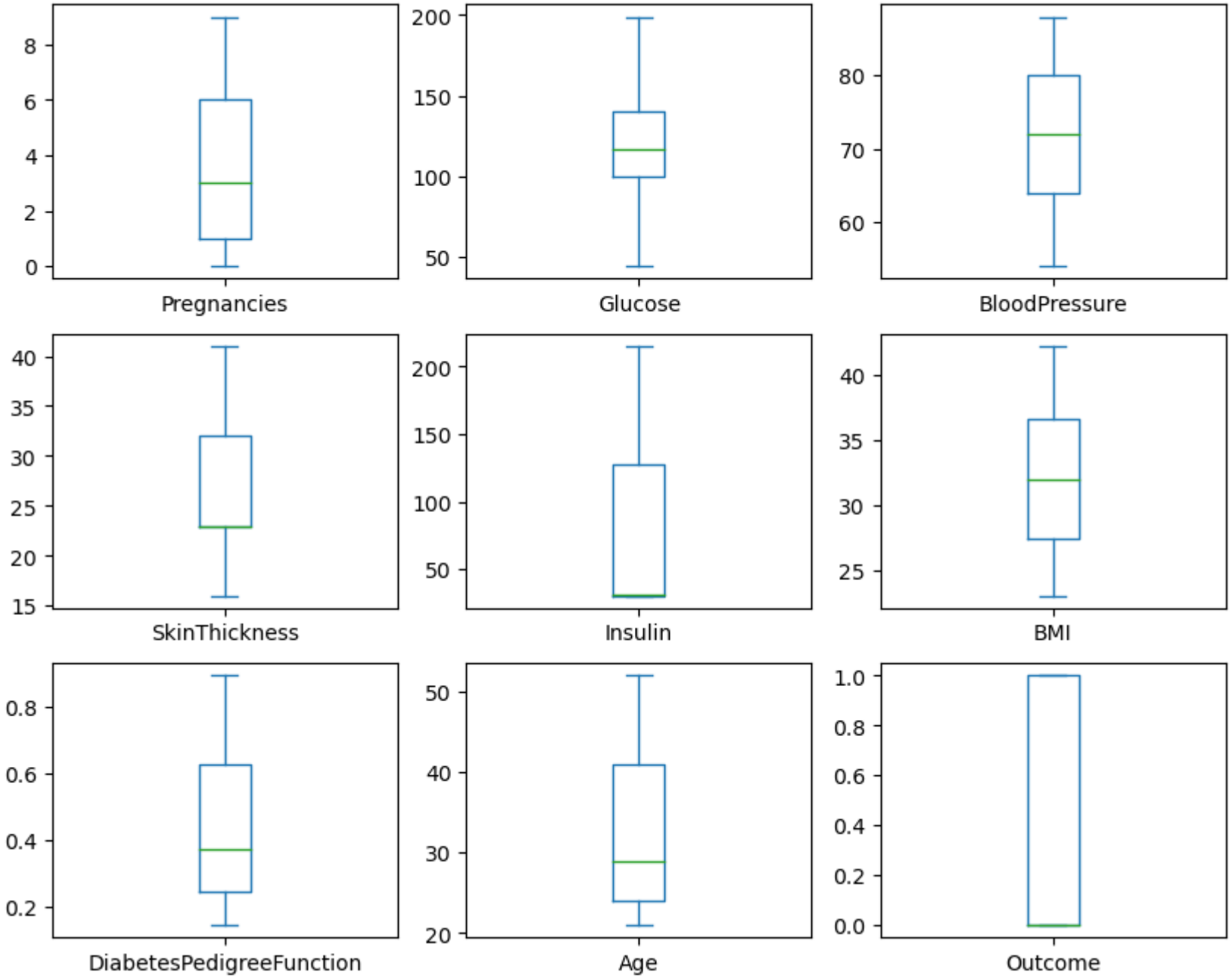
In [47]:
```
from scipy.stats.mstats import winsorize
df['Pregnancies']=winsorize(df.Pregnancies,limits=[0.07,0.093])
df['BloodPressure']=winsorize(df.BloodPressure,limits=[0.06,0.094])
df['SkinThickness']=winsorize(df.SkinThickness,limits=[0.07,0.093])
df['Insulin']=winsorize(df.Insulin,limits=[0.06,0.094])
df['BMI']=winsorize(df.BMI,limits=[0.07,0.093])
df['DiabetesPedigreeFunction']=winsorize(df.DiabetesPedigreeFunction,limits=[0.06,0.094])
df['Age']=winsorize(df.Age,limits=[0.07,0.093])
```

In [48]:
```
df.columns
```

Out[48]:
```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

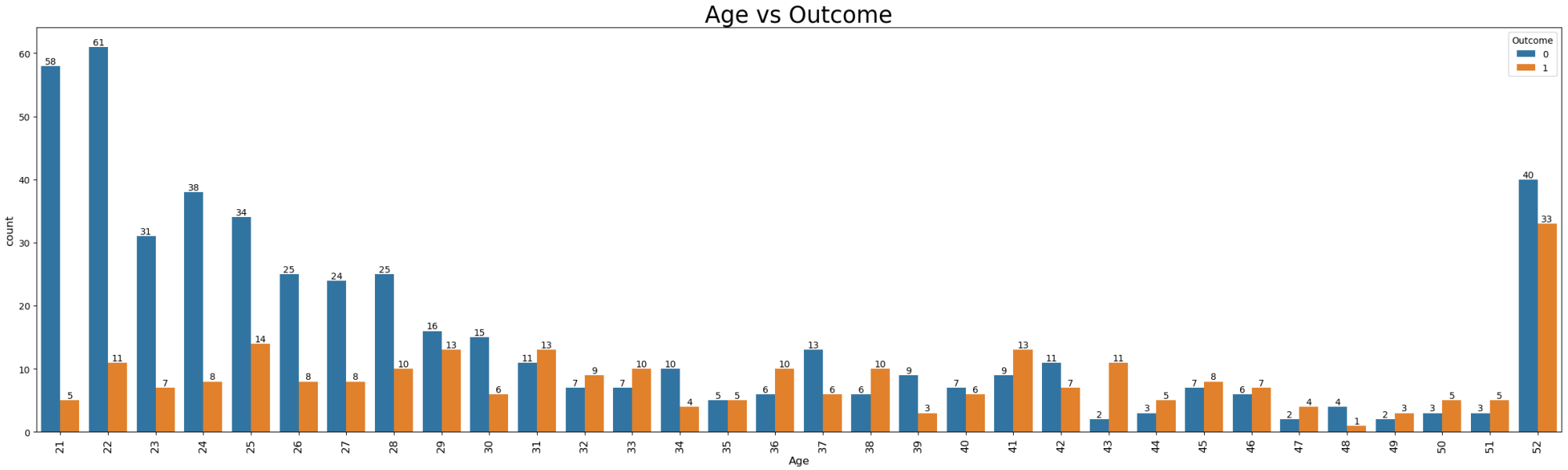
In [49]:
```
#check for outliers after winsorization
df.plot(kind='box', subplots = True, layout = (3,3), sharex = False, sharey = False,figsize=(10,8))
```

Out[49]:
```
Pregnancies                 Axes(0.125,0.653529;0.227941x0.226471)
Glucose                  Axes(0.398529,0.653529;0.227941x0.226471)
BloodPressure            Axes(0.672059,0.653529;0.227941x0.226471)
SkinThickness               Axes(0.125,0.381765;0.227941x0.226471)
Insulin                  Axes(0.398529,0.381765;0.227941x0.226471)
BMI                      Axes(0.672059,0.381765;0.227941x0.226471)
DiabetesPedigreeFunction        Axes(0.125,0.11;0.227941x0.226471)
Age                      Axes(0.398529,0.11;0.227941x0.226471)
Outcome                  Axes(0.672059,0.11;0.227941x0.226471)
dtype: object
```

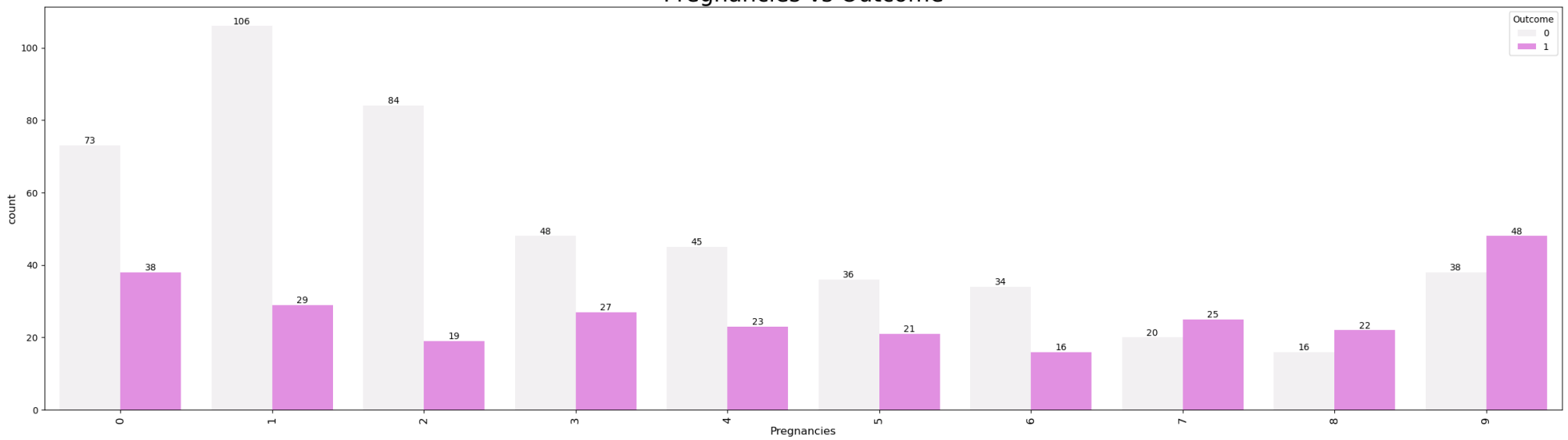


In [51]:
```
#Age vs outcome
plt.figure(figsize=(30,8))
ax=sns.countplot(x=df['Age'],hue = df['Outcome'],data = df)
for bars in ax.containers:
    ax.bar_label(bars)
plt.xlabel('Age', size = 12)
plt.ylabel('count', size = 12)
```

```
plt.title('Age vs Outcome',size = 25)
plt.xticks(rotation =90,size=12)
plt.show()
```

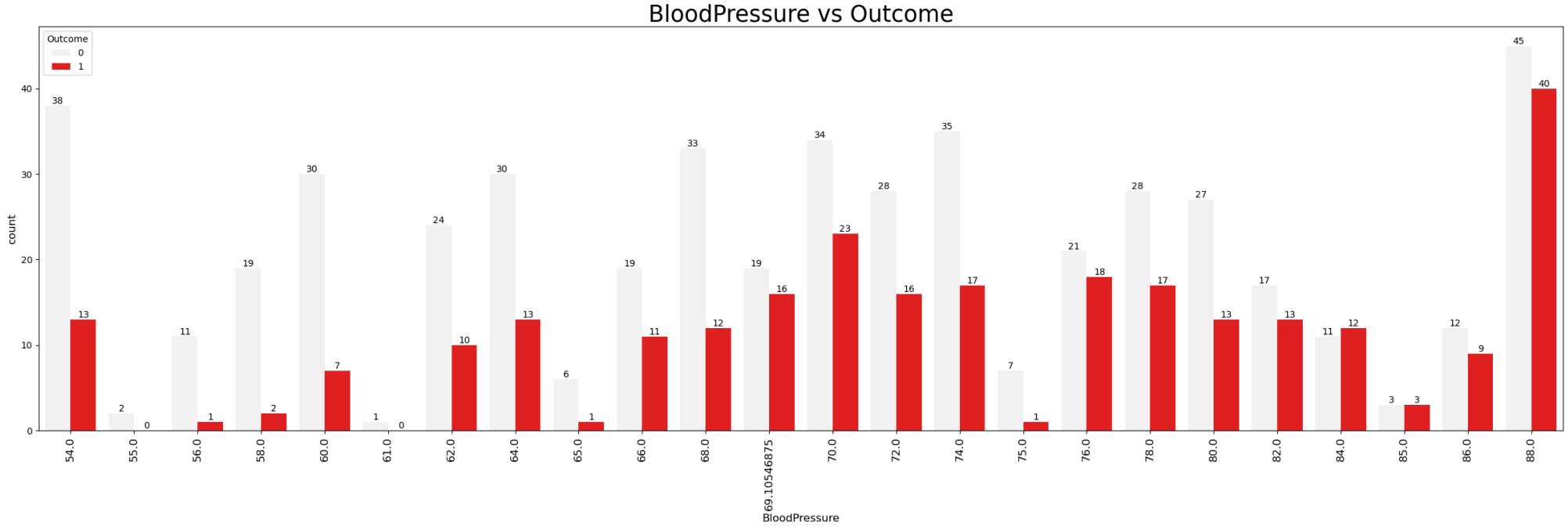


In [54]:
```
#Pregnancies vs Outcome
plt.figure(figsize=(30,8))
ax=sns.countplot(x=df['Pregnancies'],hue = df['Outcome'],data = df,color='violet')
for bars in ax.containers:
    ax.bar_label(bars)
plt.xlabel('Pregnancies', size = 12)
plt.ylabel('count', size = 12)
plt.title('Pregnancies vs Outcome',size = 25)
plt.xticks(rotation =90,size=12)
plt.show()
```

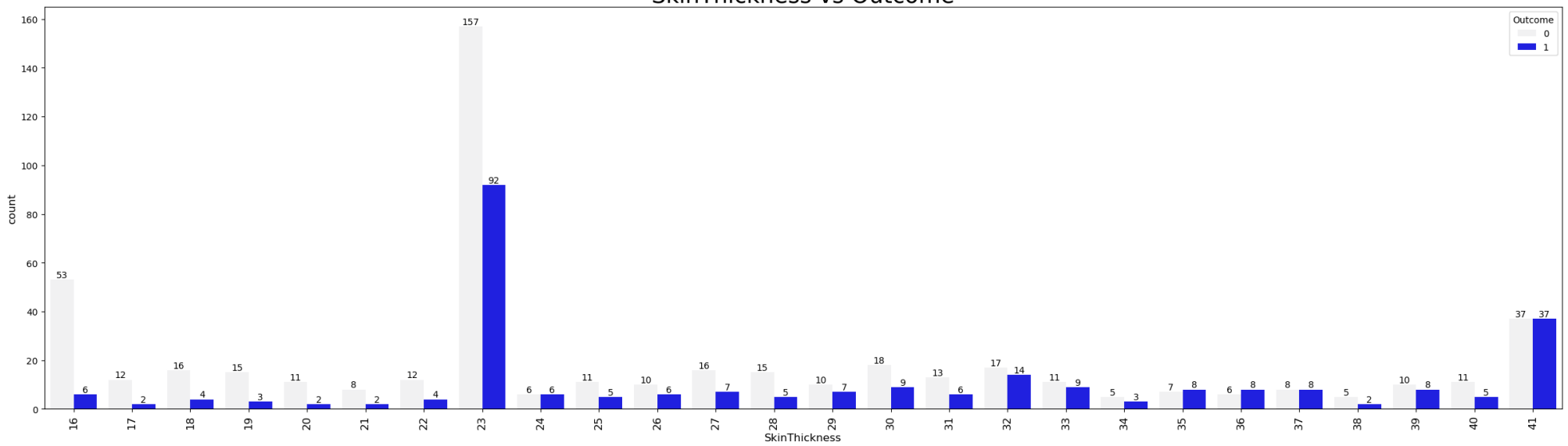


In [55]:

```
#BloodPressure vs Outcome
plt.figure(figsize=(30,8))
ax=sns.countplot(x=df['BloodPressure'],hue = df['Outcome'],data = df,color='red')
for bars in ax.containers:
    ax.bar_label(bars)
plt.xlabel('BloodPressure', size = 12)
plt.ylabel('count', size = 12)
plt.title('BloodPressure vs Outcome',size = 25)
plt.xticks(rotation =90,size=12)
plt.show()
```



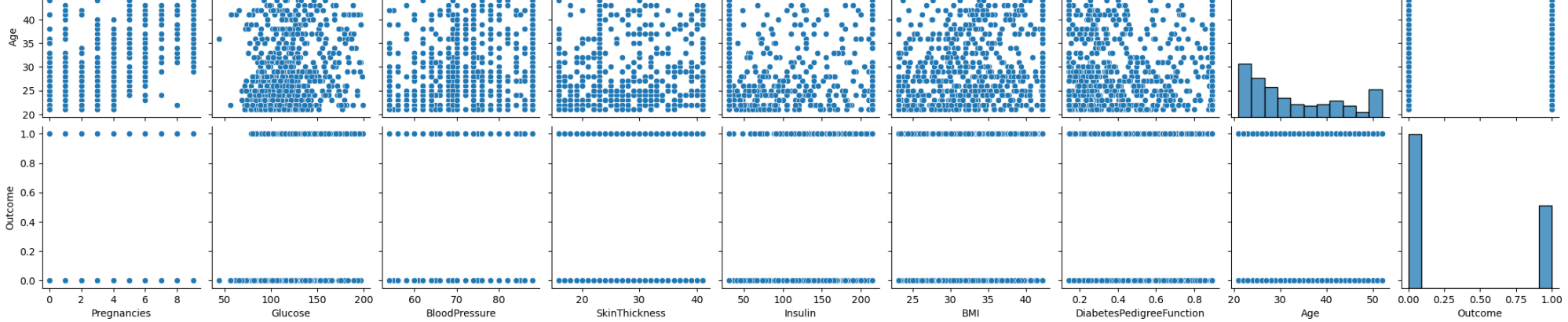
In [56]:
```
#SkinThickness vs Outcome
plt.figure(figsize=(30,8))
ax=sns.countplot(x=df['SkinThickness'],hue = df['Outcome'],data = df,color='blue')
for bars in ax.containers:
    ax.bar_label(bars)
plt.xlabel('SkinThickness', size = 12)
plt.ylabel('count', size = 12)
plt.title('SkinThickness vs Outcome',size = 25)
plt.xticks(rotation =90,size=12)
plt.show()
```



In [57]:
```
#multvariant Analysis
sns.pairplot(df)
```

Out[57]:
```
<seaborn.axisgrid.PairGrid at 0x136419210>
```


Pregnancies
Glucose
BloodPressure
SkinThickness
Insulin
BMI
DiabetesPedigreeFunction



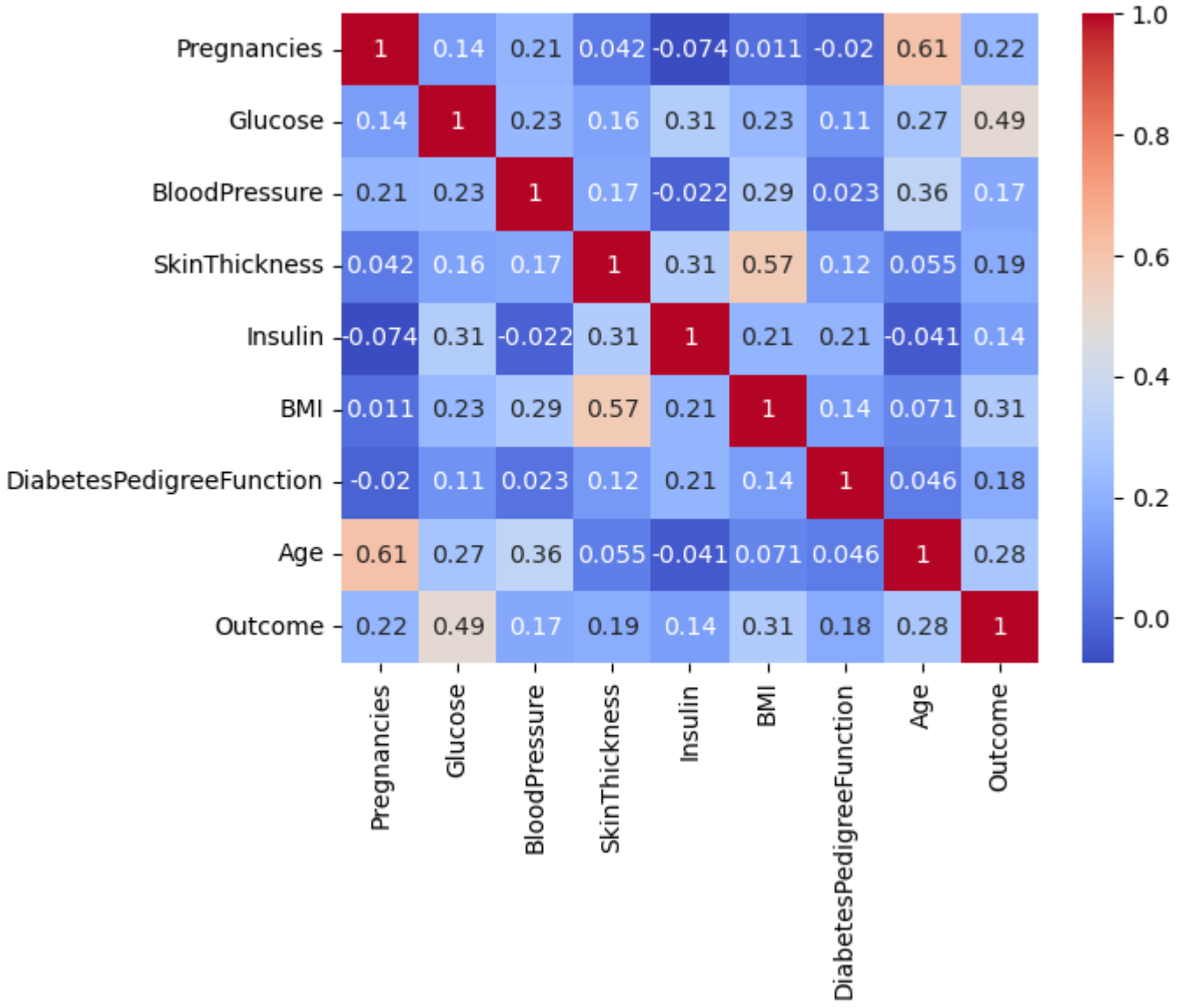
In [58]:
```
corr = df.corr()
corr
```

Out[58]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1.000000	0.136131	0.213259	0.042077	-0.074346	0.011103	-0.019510	0.609083	0.221354
Glucose	0.136131	1.000000	0.226211	0.155425	0.314589	0.228126	0.106038	0.272314	0.492908
BloodPressure	0.213259	0.226211	1.000000	0.167052	-0.021627	0.293312	0.023420	0.357636	0.169715
SkinThickness	0.042077	0.155425	0.167052	1.000000	0.307417	0.568028	0.123840	0.055250	0.187046
Insulin	-0.074346	0.314589	-0.021627	0.307417	1.000000	0.214567	0.213582	-0.040506	0.142288
BMI	0.011103	0.228126	0.293312	0.568028	0.214567	1.000000	0.137851	0.071340	0.306664
DiabetesPedigreeFunction	-0.019510	0.106038	0.023420	0.123840	0.213582	0.137851	1.000000	0.045905	0.179747
Age	0.609083	0.272314	0.357636	0.055250	-0.040506	0.071340	0.045905	1.000000	0.282376
Outcome	0.221354	0.492908	0.169715	0.187046	0.142288	0.306664	0.179747	0.282376	1.000000

In [68]:
```
sns.heatmap(corr,cmap= 'coolwarm', annot=  True)
plt.xticks(rotation = 90)
```

Out[68]:
```
(array([0.5, 1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.5, 8.5]),
 [Text(0.5, 0, 'Pregnancies'),
  Text(1.5, 0, 'Glucose'),
  Text(2.5, 0, 'BloodPressure'),
  Text(3.5, 0, 'SkinThickness'),
  Text(4.5, 0, 'Insulin'),
  Text(5.5, 0, 'BMI'),
  Text(6.5, 0, 'DiabetesPedigreeFunction'),
  Text(7.5, 0, 'Age'),
  Text(8.5, 0, 'Outcome')])
```


Pregnancies 1 0.14 0.21 0.042 -0.074 0.011 -0.02 0.61 0.22
Glucose 0.14 1 0.23 0.16 0.31 0.23 0.11 0.27 0.49
BloodPressure 0.21 0.23 1 0.17 -0.022 0.29 0.023 0.36 0.17
SkinThickness 0.042 0.16 0.17 1 0.31 0.57 0.12 0.055 0.19
Insulin -0.074 0.31 -0.022 0.31 1 0.21 0.21 -0.041 0.14
BMI 0.011 0.23 0.29 0.57 0.21 1 0.14 0.071 0.31
DiabetesPedigreeFunction -0.02 0.11 0.023 0.12 0.21 0.14 1 0.046 0.18
Age 0.61 0.27 0.36 0.055 -0.041 0.071 0.046 1 0.28
Outcome 0.22 0.49 0.17 0.19 0.14 0.31 0.18 0.28 1
Pregnancies
Glucose
BloodPressure
SkinThickness
Insulin
BMI
DiabetesPedigreeFunction
Age
Outcome
1.0
0.8
0.6
0.4
0.2
0.0

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: