Diabete Data Set

- Data Analyzing
- Treat A Outlier
- Data Visualization

In [1]:
```
import os
os.getcwd()
```

Out[1]:
```
'C:\\Users\\ap983'
```

Import a Various Type of Libraries

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
data = pd.read_csv('Downloads//diabetes.csv')
data
```

Out[3]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunctio
0	6	148	72	35	0	33.6	0.62
1	1	85	66	29	0	26.6	0.35
2	8	183	64	0	0	23.3	0.67
3	1	89	66	23	94	28.1	0.16
4	0	137	40	35	168	43.1	2.28
...	...	...	...	...	...	...	.
763	10	101	76	48	180	32.9	0.17
764	2	122	70	27	0	36.8	0.34
765	5	121	72	23	112	26.2	0.24
766	1	126	60	0	0	30.1	0.34
767	1	93	70	31	0	30.4	0.31

768 rows × 9 columns

In [4]:
```
## Check the Tope five head in a data set

data.head()
```

Out[4]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction
0	6	148	72	35	0	33.6	0.627
1	1	85	66	29	0	26.6	0.351
2	8	183	64	0	0	23.3	0.672
3	1	89	66	23	94	28.1	0.167
4	0	137	40	35	168	43.1	2.288

In [5]:
```
## Check the Bottom  five rows
data.tail()
```

Out[5]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunctio
763	10	101	76	48	180	32.9	0.17
764	2	122	70	27	0	36.8	0.34
765	5	121	72	23	112	26.2	0.24
766	1	126	60	0	0	30.1	0.34
767	1	93	70	31	0	30.4	0.31

In [6]:
```
## Check the data set Shape
data.shape
```

Out[6]:
```
(768, 9)
```

In [7]:
```
## Check the Columns name
data.columns
```

Out[7]:
```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

In [8]:
```
## Collect information form data set which column values are int type ,obeject
data.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 [9]:
```
## isna mathod return a boolean value is true or false
data.isna()
```

Out[9]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunctio
0	False	False	False	False	False	False	Fal
1	False	False	False	False	False	False	Fal
2	False	False	False	False	False	False	Fal
3	False	False	False	False	False	False	Fal
4	False	False	False	False	False	False	Fal
...	...	...	...	...	...	...	
763	False	False	False	False	False	False	Fal
764	False	False	False	False	False	False	Fal
765	False	False	False	False	False	False	Fal
766	False	False	False	False	False	False	Fal
767	False	False	False	False	False	False	Fal

768 rows × 9 columns

In [10]:
```
## Any method return one value for each column true or false
data.isna().any()
```

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

In [11]:
```
data.isna().sum()
```

Out[11]:
```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

In [12]:
```
## Check the statical method
data.describe( include ='all')
```

Out[12]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Diabetes
count	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	
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	

In [13]:
```
data['Glucose']
```

Out[13]:
```
0      148
1       85
2      183
3       89
4      137
      ...
763    101
764    122
765    121
766    126
767     93
Name: Glucose, Length: 768, dtype: int64
```

In [14]:
```
from numpy import nan
```

In [15]:
```
data.isna().sum()
```

Out[15]:
```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

In [16]:
```
## Replace the nan value with respect to 0

data['BloodPressure'] = data['BloodPressure'].replace(0,np.nan)


data['Glucose'] = data['Glucose'].replace(0,np.nan)

data['SkinThickness'] = data['SkinThickness'].replace(0,np.nan)

data['Insulin'] = data['Insulin'].replace(0,np.nan)

data['BMI'] = data['BMI'].replace(0,np.nan)
```

In [17]:
```
data.head()
```

Out[17]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction
0	6	148.0	72.0	35.0	NaN	33.6	0.627
1	1	85.0	66.0	29.0	NaN	26.6	0.351
2	8	183.0	64.0	NaN	NaN	23.3	0.672
3	1	89.0	66.0	23.0	94.0	28.1	0.167
4	0	137.0	40.0	35.0	168.0	43.1	2.288

In [18]:
```
data.describe(include = 'all')
```

Out[18]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Diabetes
count	768.000000	763.000000	733.000000	541.000000	394.000000	757.000000	
mean	3.845052	121.686763	72.405184	29.153420	155.548223	32.457464	
std	3.369578	30.535641	12.382158	10.476982	118.775855	6.924988	
min	0.000000	44.000000	24.000000	7.000000	14.000000	18.200000	
25%	1.000000	99.000000	64.000000	22.000000	76.250000	27.500000	
50%	3.000000	117.000000	72.000000	29.000000	125.000000	32.300000	
75%	6.000000	141.000000	80.000000	36.000000	190.000000	36.600000	
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	

In [19]:
```
data.isnull().sum()
```

Out[19]:
```
Pregnancies                 0
Glucose                     5
BloodPressure              35
SkinThickness             227
Insulin                   374
BMI                        11
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

In [20]:
```
## fillna method fill the value  using median method
data.fillna(data.median(),inplace = True)
data
```

Out[20]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunctio
0	6	148.0	72.0	35.0	125.0	33.6	0.62
1	1	85.0	66.0	29.0	125.0	26.6	0.35
2	8	183.0	64.0	29.0	125.0	23.3	0.67
3	1	89.0	66.0	23.0	94.0	28.1	0.16
4	0	137.0	40.0	35.0	168.0	43.1	2.28
...	...	...	...	...	...	...	.
763	10	101.0	76.0	48.0	180.0	32.9	0.17
764	2	122.0	70.0	27.0	125.0	36.8	0.34
765	5	121.0	72.0	23.0	112.0	26.2	0.24
766	1	126.0	60.0	29.0	125.0	30.1	0.34
767	1	93.0	70.0	31.0	125.0	30.4	0.31

768 rows × 9 columns

In [21]:
```
data.head()
```

Out[21]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction
0	6	148.0	72.0	35.0	125.0	33.6	0.627
1	1	85.0	66.0	29.0	125.0	26.6	0.351
2	8	183.0	64.0	29.0	125.0	23.3	0.672
3	1	89.0	66.0	23.0	94.0	28.1	0.167
4	0	137.0	40.0	35.0	168.0	43.1	2.288

In [22]:
```
data.isnull().sum()
```

Out[22]:
```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

Outlier Detection And Treatment

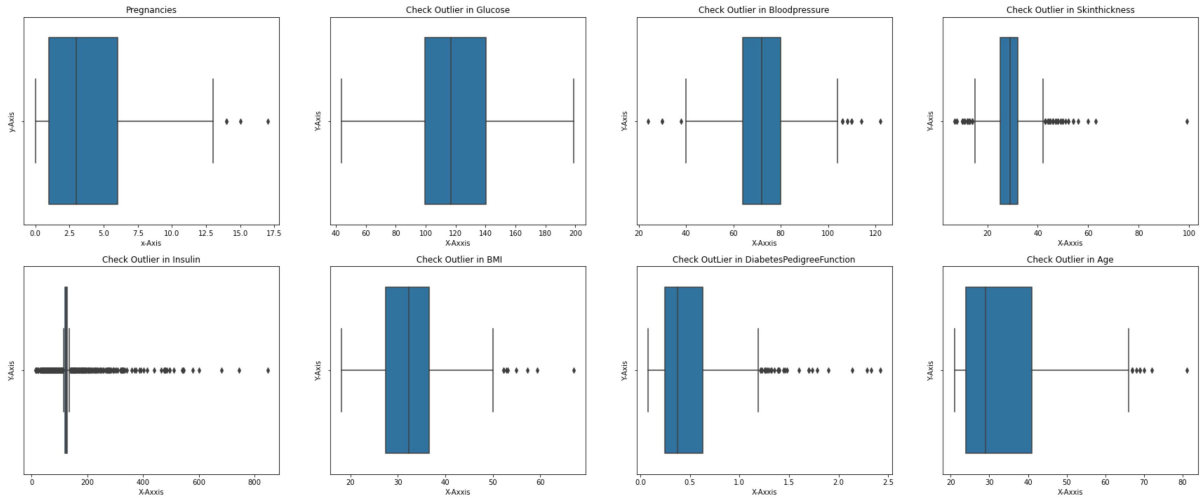
- Outlier Detection and Using Boxplot

In [23]:

```
## Check Outliesr in Prefnancies

plt.figure(figsize = (30,25))
plt.subplot(4,4,1)
sns.boxplot(data['Pregnancies'])
plt.title('Pregnancies')
plt.xlabel('x-Axis')
plt.ylabel('y-Axis')

## Check Outlier in Glucose

plt.subplot(4,4,2)
sns.boxplot(data['Glucose'])
plt.xlabel('X-Axxis')
plt.ylabel('Y-Axis')
plt.title('Check Outlier in Glucose')

## Check Outlier in Bloodpressure

plt.subplot(4,4,3)
sns.boxplot(data['BloodPressure'])
plt.xlabel('X-Axxis')
plt.ylabel('Y-Axis')
plt.title('Check Outlier in Bloodpressure')

## Check Outlier in Skinthickness

plt.subplot(4,4,4)
sns.boxplot(data['SkinThickness'])
plt.xlabel('X-Axxis')
plt.ylabel('Y-Axis')
plt.title('Check Outlier in Skinthickness')

## Check Outlier in Insulin

plt.subplot(4,4,5)
sns.boxplot(data['Insulin'])
plt.xlabel('X-Axxis')
plt.ylabel('Y-Axis')
plt.title('Check Outlier in Insulin')

## Check Outlier in BMI

plt.subplot(4,4,6)
sns.boxplot(data['BMI'])
plt.xlabel('X-Axxis')
plt.ylabel('Y-Axis')
plt.title(' Check Outlier in BMI')

## Check OutLier in DiabetesPedigreeFunction

plt.subplot(4,4,7)
sns.boxplot(data['DiabetesPedigreeFunction'])
plt.xlabel('X-Axxis')
plt.ylabel('Y-Axis')
plt.title('Check OutLier in DiabetesPedigreeFunction')
```

```python
## Check Outlier in Age

plt.subplot(4,4,8)
sns.boxplot(data['Age'])
plt.xlabel('X-Axxis')
plt.ylabel('Y-Axis')
plt.title('Check Outlier in Age')
plt.show()
```

```
C:\Users\ap983\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureW
arning: Pass the following variable as a keyword arg: x. From version 0.12, t
he only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\ap983\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureW
arning: Pass the following variable as a keyword arg: x. From version 0.12, t
he only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\ap983\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureW
arning: Pass the following variable as a keyword arg: x. From version 0.12, t
he only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\ap983\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureW
arning: Pass the following variable as a keyword arg: x. From version 0.12, t
he only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\ap983\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureW
arning: Pass the following variable as a keyword arg: x. From version 0.12, t
he only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\ap983\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureW
arning: Pass the following variable as a keyword arg: x. From version 0.12, t
he only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\ap983\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureW
arning: Pass the following variable as a keyword arg: x. From version 0.12, t
he only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\ap983\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureW
arning: Pass the following variable as a keyword arg: x. From version 0.12, t
he only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



Inside a Boxplot show the dot point this a Outlier then i am treat the Outlier using Clip method .Clip method work of the print the lower value and upper value i am mention below how to print lower value and upper value

In [24]:
```
data['Pregnancies'] = data['Pregnancies'].clip(lower = data['Pregnancies'].quar
```

In [25]:
```
data['BloodPressure'] = data['BloodPressure'].clip(lower = data['BloodPressure'
```

In [26]:
```
data['SkinThickness'] = data['SkinThickness'].clip(lower = data['SkinThickness'
```

In [27]:
```
data['Insulin'] = data['Insulin'].clip(lower = data['Insulin'].quantile(0.05),u
```

In [28]:
```
data['BMI'] = data['BMI'].clip(lower = data['BMI'].quantile(0.05),upper = data[
```

In [29]:
```
data['DiabetesPedigreeFunction'] = data['DiabetesPedigreeFunction'].clip(lower
```

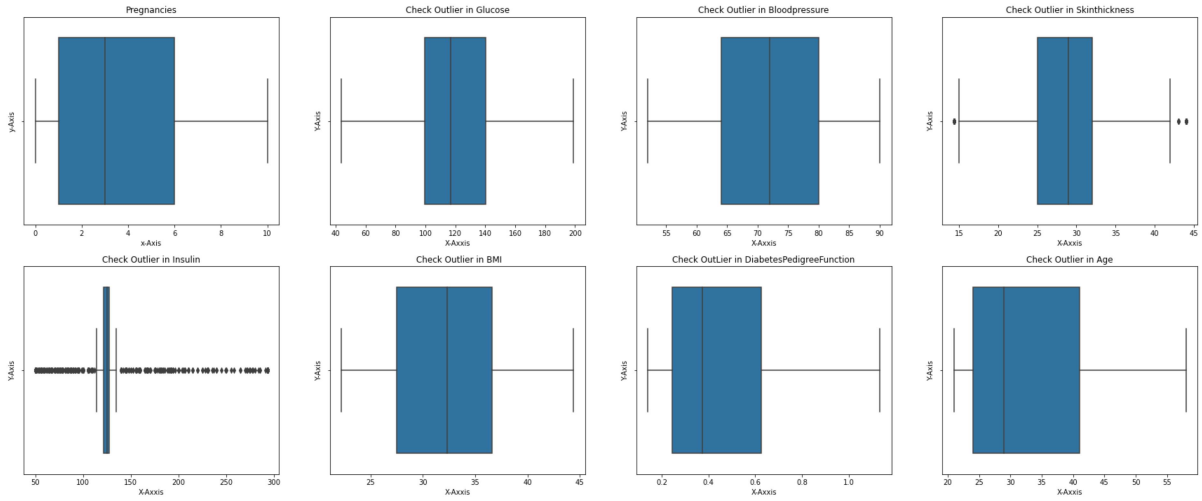
In [30]:
```
data['Age'] = data['Age'].clip(lower = data['Age'].quantile(0.05),upper = data[
```

After Treated Outlier show the box plot

In [31]:
```
## After Treated  Outliesr in Prefnancies

plt.figure(figsize = (30,25))
plt.subplot(4,4,1)
sns.boxplot(data['Pregnancies'])
plt.title('Pregnancies')
plt.xlabel('x-Axis')
plt.ylabel('y-Axis')

## After Treated  Outlier in Glucose

plt.subplot(4,4,2)
sns.boxplot(data['Glucose'])
plt.xlabel('X-Axxis')
plt.ylabel('Y-Axis')
plt.title('Check Outlier in Glucose')

## After Treated  Outlier in Bloodpressure

plt.subplot(4,4,3)
sns.boxplot(data['BloodPressure'])
plt.xlabel('X-Axxis')
plt.ylabel('Y-Axis')
plt.title('Check Outlier in Bloodpressure')

## After Treated  Outlier in Skinthickness

plt.subplot(4,4,4)
sns.boxplot(data['SkinThickness'])
plt.xlabel('X-Axxis')
plt.ylabel('Y-Axis')
plt.title('Check Outlier in Skinthickness')

## After Treated  Outlier in Insulin

plt.subplot(4,4,5)
sns.boxplot(data['Insulin'])
plt.xlabel('X-Axxis')
plt.ylabel('Y-Axis')
plt.title('Check Outlier in Insulin')

## Check Outlier in BMI

plt.subplot(4,4,6)
sns.boxplot(data['BMI'])
plt.xlabel('X-Axxis')
plt.ylabel('Y-Axis')
plt.title(' Check Outlier in BMI')

## After Treated  treated a OutLier in DiabetesPedigreeFunction

plt.subplot(4,4,7)
sns.boxplot(data['DiabetesPedigreeFunction'])
plt.xlabel('X-Axxis')
plt.ylabel('Y-Axis')
plt.title('Check OutLier in DiabetesPedigreeFunction')
```

```python
## After Treated  Outlier in Age

plt.subplot(4,4,8)
sns.boxplot(data['Age'])
plt.xlabel('X-Axxis')
plt.ylabel('Y-Axis')
plt.title('Check Outlier in Age')
plt.show()
```

```
C:\Users\ap983\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureW
arning: Pass the following variable as a keyword arg: x. From version 0.12, t
he only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\ap983\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureW
arning: Pass the following variable as a keyword arg: x. From version 0.12, t
he only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\ap983\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureW
arning: Pass the following variable as a keyword arg: x. From version 0.12, t
he only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\ap983\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureW
arning: Pass the following variable as a keyword arg: x. From version 0.12, t
he only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\ap983\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureW
arning: Pass the following variable as a keyword arg: x. From version 0.12, t
he only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\ap983\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureW
arning: Pass the following variable as a keyword arg: x. From version 0.12, t
he only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\ap983\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureW
arning: Pass the following variable as a keyword arg: x. From version 0.12, t
he only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\ap983\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureW
arning: Pass the following variable as a keyword arg: x. From version 0.12, t
he only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



As we can see there are still outlier in column skinthikness and insulin lets try to manipulate the percentile velue

- manipulate the skinthickness and insulin

In [32]:
```
data['SkinThickness'] = data['SkinThickness'].clip(lower = data['SkinThickness'
```

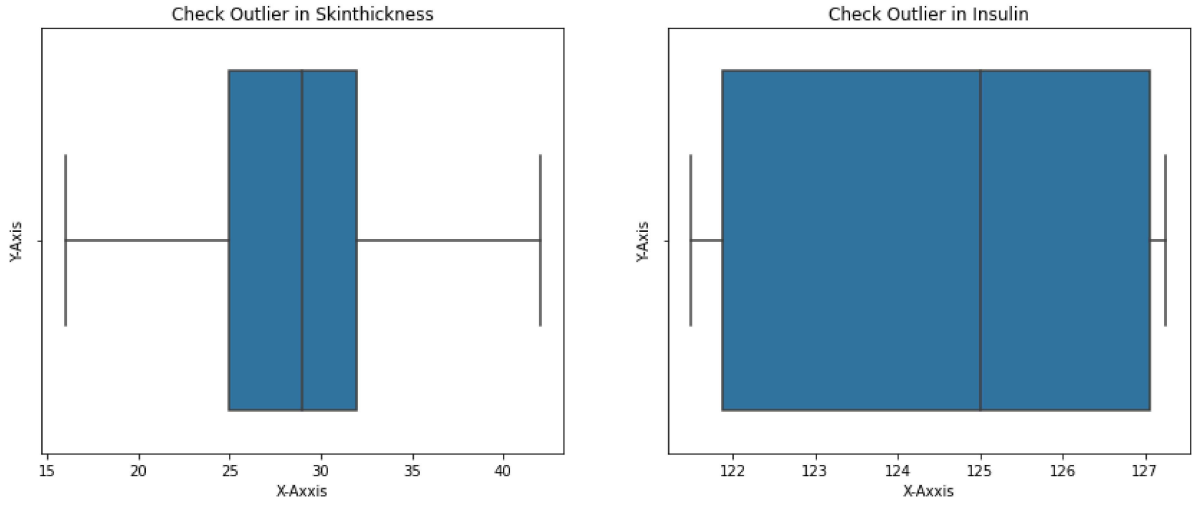
In [33]:
```
data['Insulin'] = data['Insulin'].clip(lower = data['Insulin'].quantile(0.25),u
```

After manipulate Box plot show the without outlier

In [34]:
```
###aftemanipulate the skinthiclness
plt.figure(figsize = (30,25))
plt.subplot(4,4,1)
sns.boxplot(data['SkinThickness'])
plt.xlabel('X-Axxis')
plt.ylabel('Y-Axis')
plt.title('Check Outlier in Skinthickness')

## After Manipulate  Outlier in Insulin

plt.subplot(4,4,2)
sns.boxplot(data['Insulin'])
plt.xlabel('X-Axxis')
plt.ylabel('Y-Axis')
plt.title('Check Outlier in Insulin')
```

```
C:\Users\ap983\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureW
arning: Pass the following variable as a keyword arg: x. From version 0.12, t
he only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\ap983\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureW
arning: Pass the following variable as a keyword arg: x. From version 0.12, t
he only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[34]: Text(0.5, 1.0, 'Check Outlier in Insulin')

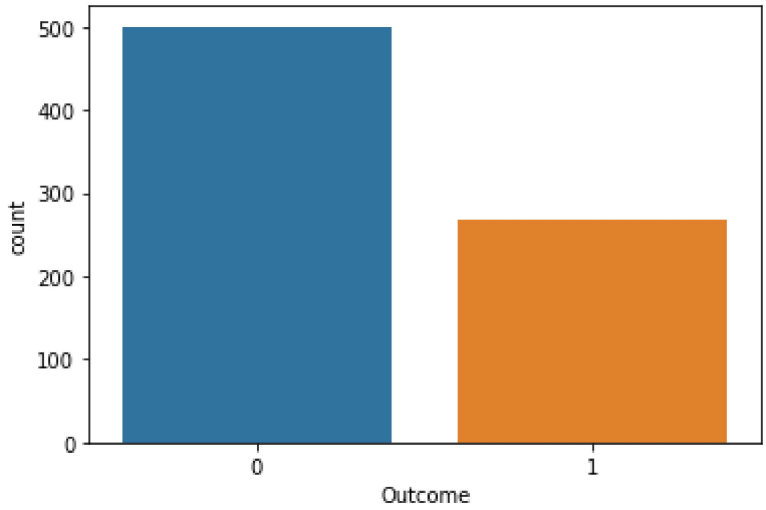


Data Visualization

- Using various type of Graph

In [35]:
```
## Let's start understanding the distribution of diabitic vs non diabitic patie

sns.countplot(data['Outcome'])
plt.show()
```

```
C:\Users\ap983\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureW
arning: Pass the following variable as a keyword arg: x. From version 0.12, t
he only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



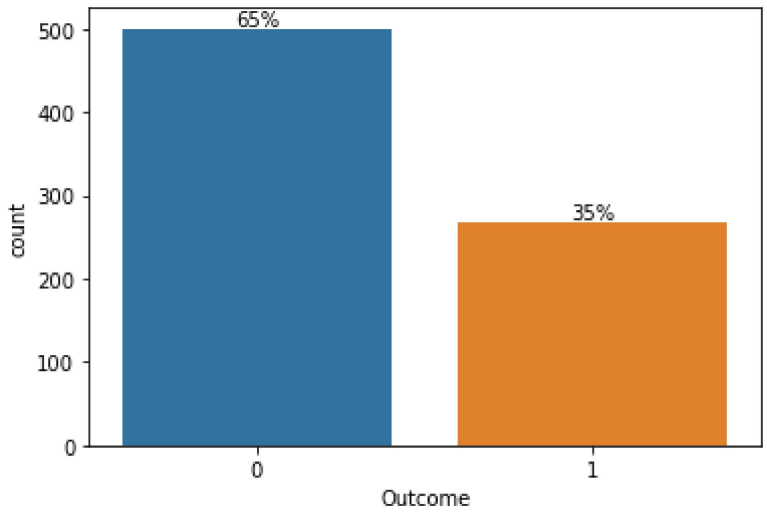
In [36]:
```
data.head(768)
```

Out[36]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunctio
0	6.0	148.0	72.0	35.0	125.00	33.6	0.6270
1	1.0	85.0	66.0	29.0	125.00	26.6	0.3510
2	8.0	183.0	64.0	29.0	125.00	23.3	0.6720
3	1.0	89.0	66.0	23.0	121.50	28.1	0.1670
4	0.0	137.0	52.0	35.0	127.25	43.1	1.1328
...	...	...	...	...	...	...	.
763	10.0	101.0	76.0	42.0	127.25	32.9	0.1710
764	2.0	122.0	70.0	27.0	125.00	36.8	0.3400
765	5.0	121.0	72.0	23.0	121.50	26.2	0.2450
766	1.0	126.0	60.0	29.0	125.00	30.1	0.3490
767	1.0	93.0	70.0	31.0	125.00	30.4	0.3150

768 rows × 9 columns

In [37]:
```
## Let's start the Disribution of Pecentile in the outcome
total = float(len(data))
ax = sns.countplot(x='Outcome',data=data)
for p in ax.patches:
    height = p.get_height()
    ax.text(p.get_x()+p.get_width()/2.,height+3,'{0:.0%}'.format(height/total),
```



In [52]:
```
## Pairplot Analysis
sns.pairplot(data,hue='Outcome',diag_kind='kde')
```

Out[52]: <seaborn.axisgrid.PairGrid at 0x21ada287820>

In [ ]:

In [ ]:

In [ ]:

In [ ]: