# 31. 1 Detect missing values with pandas dataframe. functions: .info() and .isna()

In [3]:

```
import pandas as p
df=p.read_csv("titanic.csv")

info=df.info()

print("\n\n\nis_na:\n\n",df.isna().head(7))

is_null_su=df.isna().sum()
print("\n\n\nCount of all Missing values:\n\n",df.isna().sum())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Survived     418 non-null    int64  
 2   Pclass       418 non-null    int64  
 3   Name         418 non-null    object 
 4   Sex          418 non-null    object 
 5   Age          332 non-null    float64
 6   SibSp        418 non-null    int64  
 7   Parch        418 non-null    int64  
 8   Ticket       418 non-null    object 
 9   Fare         417 non-null    float64
 10  Cabin        91 non-null     object 
 11  Embarked     418 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 39.3+ KB



is_na:

    PassengerId  Survived  Pclass   Name    Sex    Age  SibSp  Parch  Ticket  \
0        False     False   False  False  False  False  False  False   False   
1        False     False   False  False  False  False  False  False   False   
2        False     False   False  False  False  False  False  False   False   
3        False     False   False  False  False  False  False  False   False   
4        False     False   False  False  False  False  False  False   False   
5        False     False   False  False  False  False  False  False   False   
6        False     False   False  False  False  False  False  False   False   

    Fare  Cabin  Embarked  
0  False   True     False  
1  False   True     False  
2  False   True     False  
3  False   True     False  
4  False   True     False  
5  False   True     False  
6  False   True     False  



Count of all Missing values:

 PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

# 32. 2 Replace

In [9]:

```
import pandas as p
df=p.read_csv("titanic.csv")
```

```
print(df.isna().sum())

#Replacing all NaN values with -1
df=df.replace({n.nan:-1})

print("\n\n\n")
print(df.isna().sum())
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64




PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin          0
Embarked       0
dtype: int64
```

# 33. 3 Remove data objects with missing values

In [11]:

```
df=p.read_csv("titanic.csv")
print("\nBefore Droping:\n")
df.info()

#drops Entier row data if as nan values in any coloumn
df=df.dropna()

#OR
#dp=dp.dropna(axis=0)

print('\n\nAfter Droping:\n')
df.info()
```

```
Before Droping:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Survived     418 non-null    int64  
 2   Pclass       418 non-null    int64  
 3   Name         418 non-null    object 
 4   Sex          418 non-null    object 
 5   Age          332 non-null    float64
 6   SibSp        418 non-null    int64  
 7   Parch        418 non-null    int64  
 8   Ticket       418 non-null    object 
 9   Fare         417 non-null    float64
 10  Cabin        91 non-null     object 
 11  Embarked     418 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 39.3+ KB


After Droping:

<class 'pandas.core.frame.DataFrame'>
Index: 87 entries, 12 to 414
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  87 non-null     int64  
 1   Survived     87 non-null     int64  
 2   Pclass       87 non-null     int64  
 3   Name         87 non-null     object 
 4   Sex          87 non-null     object 
 5   Age          87 non-null     float64
 6   SibSp        87 non-null     int64  
 7   Parch        87 non-null     int64  
 8   Ticket       87 non-null     object 
 9   Fare         87 non-null     float64
 10  Cabin        87 non-null     object 
 11  Embarked     87 non-null     object 
dtypes: float64(2), int64(5), object(5)
memory usage: 8.8+ KB
```

# 34. Remove the attributes with missing values

In [14]:

```python
df=p.read_csv("titanic.csv")

print("\nBefore Droping:\n")
df.info()

df=df.dropna(axis=1)

#OR
#df=df.drop(columns=df.columns[df.isnull().any()])

print('\n\nAfter Droping:\n')
df.info()
```

```
Before Droping:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Survived     418 non-null    int64  
 2   Pclass       418 non-null    int64  
 3   Name         418 non-null    object 
 4   Sex          418 non-null    object 
 5   Age          332 non-null    float64
 6   SibSp        418 non-null    int64  
 7   Parch        418 non-null    int64  
 8   Ticket       418 non-null    object 
 9   Fare         417 non-null    float64
 10  Cabin        91 non-null     object 
 11  Embarked     418 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 39.3+ KB


After Droping:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 9 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   PassengerId  418 non-null    int64 
 1   Survived     418 non-null    int64 
 2   Pclass       418 non-null    int64 
 3   Name         418 non-null    object
 4   Sex          418 non-null    object
 5   SibSp        418 non-null    int64 
 6   Parch        418 non-null    int64 
 7   Ticket       418 non-null    object
 8   Embarked     418 non-null    object
dtypes: int64(5), object(4)
memory usage: 29.5+ KB
```

# 35. Estimate and impute missing values Filling it with some Arbitrary value here it is 0

In [19]:

```
df=p.read_csv("titanic.csv")

print("Before Filling Null values:\n\n",df.isna().sum())

df=df.fillna(0)

print("\n\nAfter Filling Null Values:\n\n",df.isna().sum())
```

```
Before Filling Null values:

 PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64


After Filling Null Values:

 PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin          0
Embarked       0
dtype: int64
```

# 36. Replacing with Mean Value

In [24]:

```
df=p.read_csv("titanic.csv")

print("Before Replacing:\n\n",df['Age'].head(7))

print("\nMean of Age Column:",df['Age'].mean())

dp=df['Age'].fillna(df['Age'].mean())
print("\nAfter Replacing with Mean:\n\n",dp.head(7))
```

```
Before Replacing:

 0    34.5
1    47.0
2    62.0
3    27.0
4    22.0
5    14.0
6    30.0
Name: Age, dtype: float64

Mean of Age Column: 30.272590361445783

After Replacing with Mean:

 0    34.5
1    47.0
2    62.0
3    27.0
4    22.0
5    14.0
6    30.0
Name: Age, dtype: float64
```

# 37. Replacing with Median Value

In [27]:

```
df=p.read_csv("titanic.csv")

print("Before Replacing:\n\n",df['Age'].head(7))

print("\nMedian of Age Column:",df['Age'].median())

dp=df['Age'].fillna(df['Age'].median())
print("\nAfter Replacing with Mean:\n\n",dp.head(7))
```

```
Before Replacing:

 0    34.5
1    47.0
2    62.0
3    27.0
4    22.0
5    14.0
6    30.0
Name: Age, dtype: float64

Median of Age Column: 27.0

After Replacing with Mean:

 0    34.5
1    47.0
2    62.0
3    27.0
4    22.0
5    14.0
6    30.0
Name: Age, dtype: float64
```

# 38. Replacing with Mode value

In [33]:

```python
df=p.read_csv("titanic.csv")

print("Before Replacing:\n\n",df['Age'].head(7))

print("\nMode of Age Column:",df['Age'].mode()[0])

dp=df['Age'].fillna(df['Age'].mode()[0])
print("\nAfter Replacing with Mode:\n\n",dp.head(7))
```

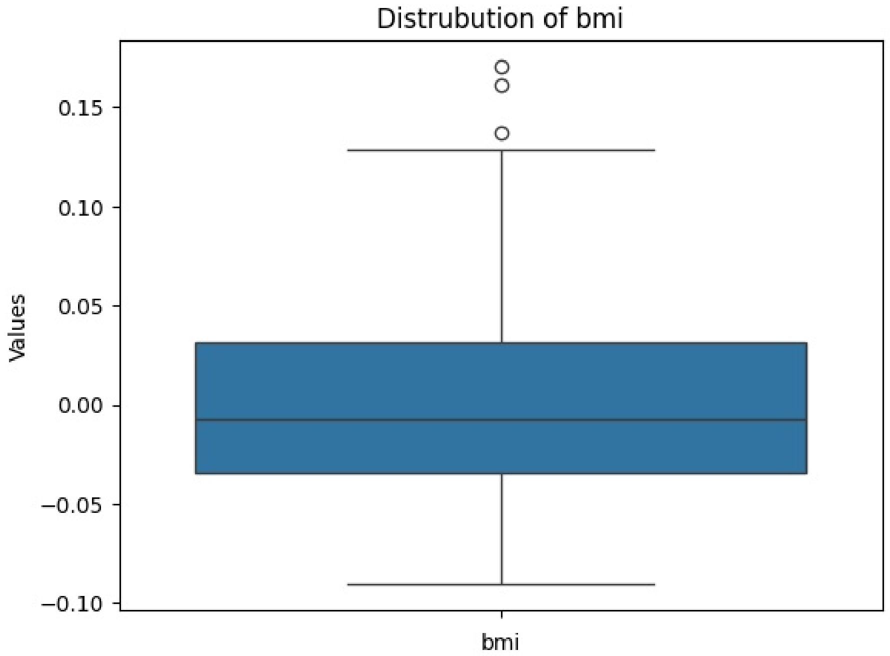
```
Before Replacing:

 0    34.5
1    47.0
2    62.0
3    27.0
4    22.0
5    14.0
6    30.0
Name: Age, dtype: float64

Mode of Age Column: 21.0

After Replacing with Mode:

 0    34.5
1    47.0
2    62.0
3    27.0
4    22.0
5    14.0
6    30.0
Name: Age, dtype: float64
```

# 39. Univariate Outliers

In [36]:

```python
from sklearn.datasets import load_diabetes
import matplotlib.pyplot as m
import seaborn as s

dp=load_diabetes()
col_n =dp.feature_names
df= p.DataFrame(dp.data);
df.columns = col_n

#Visualizing of Outliers
s.boxplot(df['bmi'])
m.ylabel('Values');
m.xlabel('bmi');
m.title('Distrubution of bmi')
m.show()

#IQR
q1=df['bmi'].quantile(0.25)
q3=df['bmi'].quantile(0.75)
```

```
iqr=q3-q1

#Floor and Capping
flo=q1-1.5*iqr
cap=q3+1.5*iqr
out=df[(df.bmi<=flo)|(df.bmi>=cap)]
print("Outliers:\n",out)
```

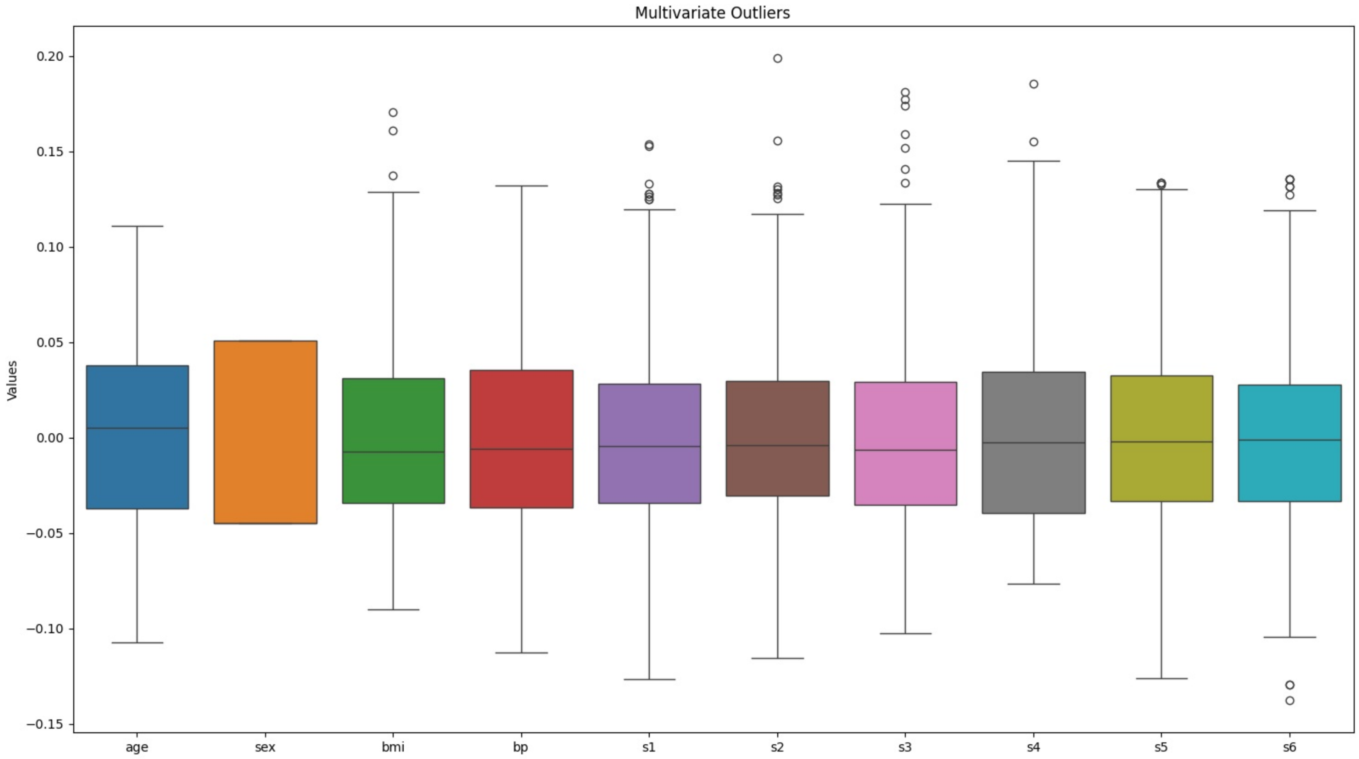


```
Outliers:
            age       sex       bmi        bp        s1        s2        s3  \
256 -0.049105 -0.044642  0.160855 -0.046985 -0.029088 -0.019790 -0.047082
366 -0.045472  0.050680  0.137143 -0.015999  0.041086  0.031880 -0.043401
367 -0.009147  0.050680  0.170555  0.014987  0.030078  0.033759 -0.021311

           s4        s5        s6
256  0.034309  0.028020  0.011349
366  0.071210  0.071019  0.048628
367  0.034309  0.033654  0.032059
```

# 40. Multivariate Outliers

In [41]:

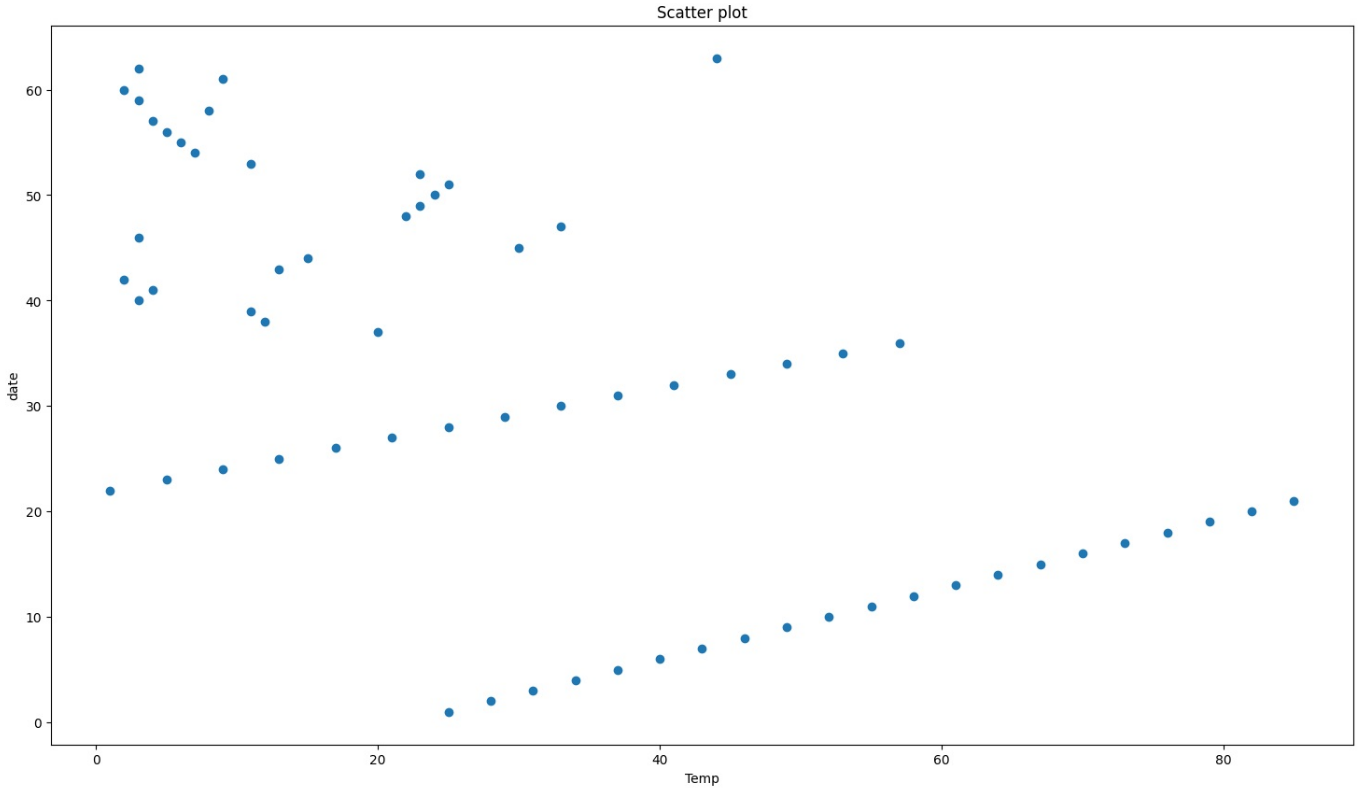
```
from sklearn.datasets import load_diabetes
from matplotlib import pyplot as m
import seaborn as s

dp=load_diabetes()
col_n=dp.feature_names
df=p.DataFrame(dp.data)
df.columns=col_n

m.figure(figsize=(18,10))
s.boxplot(data=df)
m.title('Multivariate Outliers')
m.ylabel('Values')
m.show()
```



# 41. Time series outlier detection

In [49]:

```
import pandas as p
from matplotlib import pyplot as m
df=p.read_csv("temp.csv")
x=df.temp
y=df.day
m.figure(figsize=(18,10))
m.scatter(x,y,label="values of x & y")
m.xlabel("Temp")
m.ylabel("date")
m.title("Scatter plot")
m.show()
```



# 42. Titanic Dataset Perform:

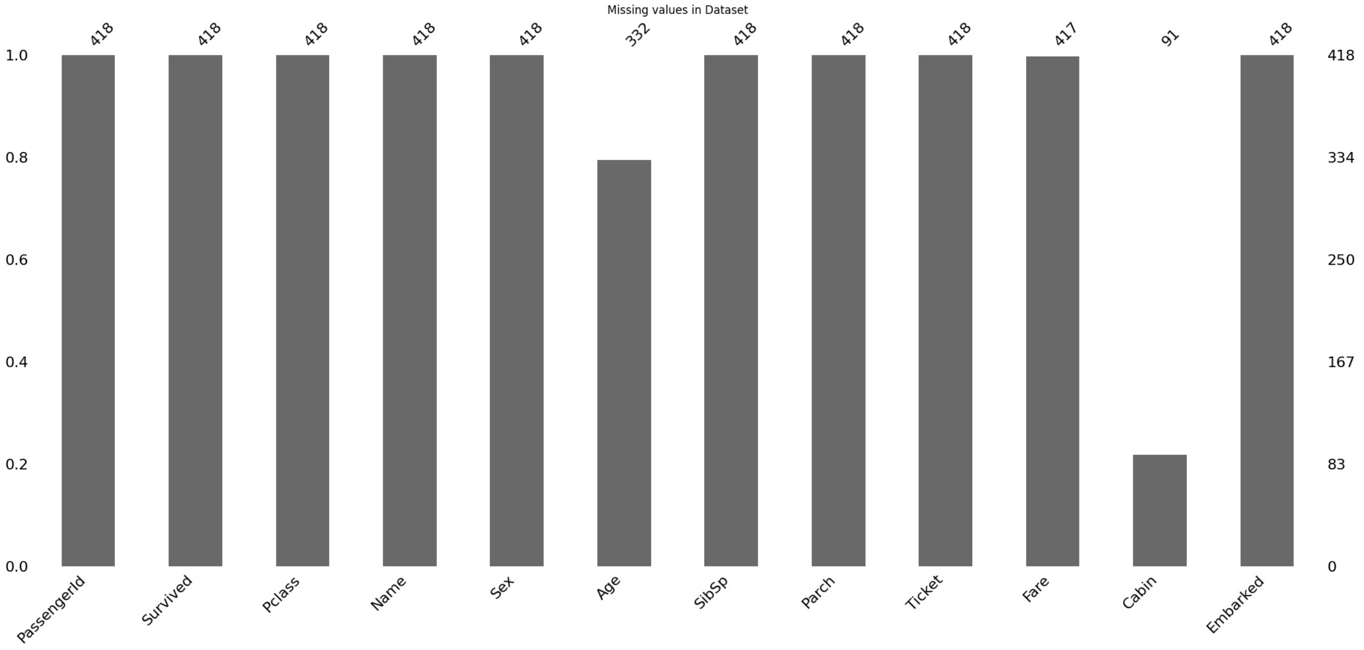
o Visualize missing values as bar plot and matrix plot

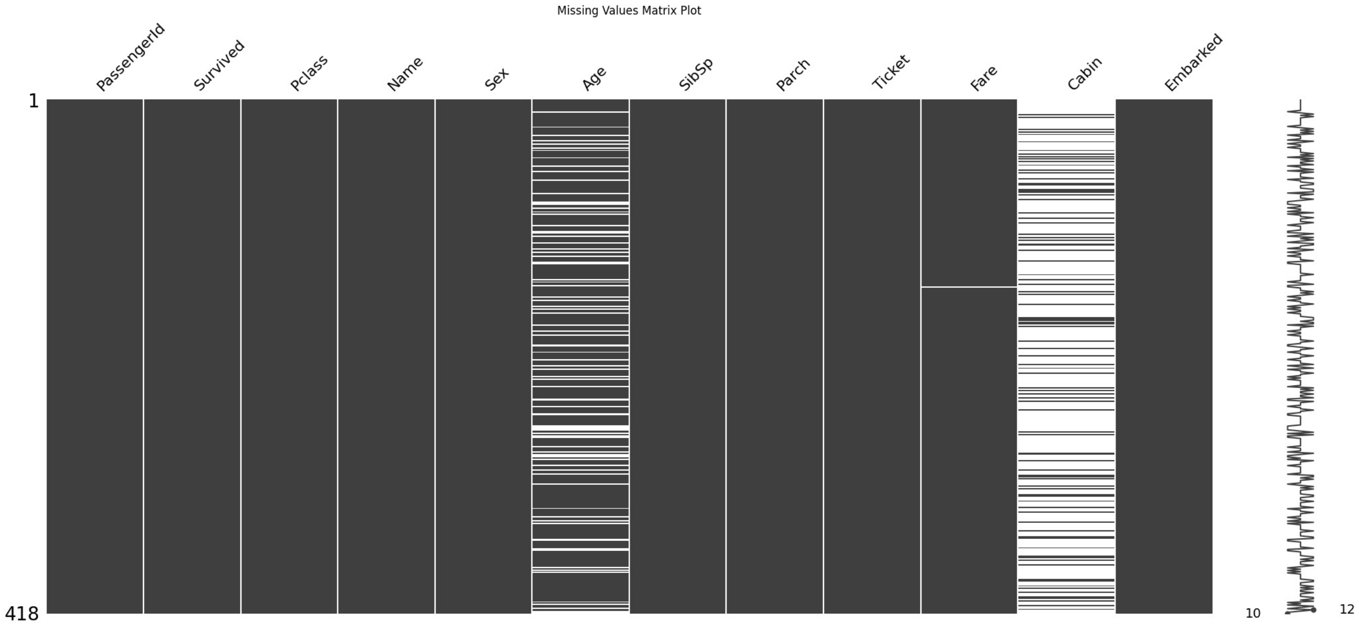
o Handle Missing values by deleting data objects and attributes

o Impute the missing values

In [54]:

```
import missingno as ms
ti_da=p.read_csv("titanic.csv")

#Box Plot
ms.bar(ti_da)
m.title("Missing values in Dataset")
m.show()

#Matrix Plot
ms.matrix(ti_da)
m.title('Missing Values Matrix Plot')
m.show()

#Removing Null Objects
print("Before Droping Objects:\n")
ti_da.info()
ti_d=ti_da.dropna(axis=0)
print("\n\nAfter Droping objects:\n")
ti_d.info()

#Removing Null Attributes
print("\nBefore Droping Attributes:\n")
ti_da.info()
ti=ti_da.dropna(axis=1)
print("\n\nAfter Droping Attributes:\n")
ti.info()

#Imputing Missing value of Age column through Mean
print("\n\nAge column before imputing:\n")
ti_da['Age'].info()
ti_ag=ti_da['Age'].fillna(ti_da['Age'].mean())
print("\n\nAfter Imputing:\n")
ti_ag.info()
```





```
Before Droping Objects:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Survived     418 non-null    int64  
 2   Pclass       418 non-null    int64  
 3   Name         418 non-null    object 
 4   Sex          418 non-null    object 
 5   Age          332 non-null    float64
 6   SibSp        418 non-null    int64  
 7   Parch        418 non-null    int64  
 8   Ticket       418 non-null    object 
 9   Fare         417 non-null    float64
 10  Cabin        91 non-null     object 
 11  Embarked     418 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 39.3+ KB


After Droping objects:

<class 'pandas.core.frame.DataFrame'>
Index: 87 entries, 12 to 414
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  87 non-null     int64  
 1   Survived     87 non-null     int64  
 2   Pclass       87 non-null     int64  
 3   Name         87 non-null     object 
 4   Sex          87 non-null     object 
 5   Age          87 non-null     float64
 6   SibSp        87 non-null     int64  
 7   Parch        87 non-null     int64  
 8   Ticket       87 non-null     object 
 9   Fare         87 non-null     float64
 10  Cabin        87 non-null     object 
 11  Embarked     87 non-null     object 
dtypes: float64(2), int64(5), object(5)
memory usage: 8.8+ KB

Before Droping Attributes:
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Survived     418 non-null    int64  
 2   Pclass       418 non-null    int64  
 3   Name         418 non-null    object 
 4   Sex          418 non-null    object 
 5   Age          332 non-null    float64
 6   SibSp        418 non-null    int64  
 7   Parch        418 non-null    int64  
 8   Ticket       418 non-null    object 
 9   Fare         417 non-null    float64
 10  Cabin        91 non-null     object 
 11  Embarked     418 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 39.3+ KB


After Droping Attributes:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 9 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   PassengerId  418 non-null    int64 
 1   Survived     418 non-null    int64 
 2   Pclass       418 non-null    int64 
 3   Name         418 non-null    object
 4   Sex          418 non-null    object
 5   SibSp        418 non-null    int64 
 6   Parch        418 non-null    int64 
 7   Ticket       418 non-null    object
 8   Embarked     418 non-null    object
dtypes: int64(5), object(4)
memory usage: 29.5+ KB


Age column before imputing:

<class 'pandas.core.series.Series'>
RangeIndex: 418 entries, 0 to 417
Series name: Age
Non-Null Count  Dtype  
--------------  -----  
332 non-null    float64
dtypes: float64(1)
memory usage: 3.4 KB


After Imputing:

<class 'pandas.core.series.Series'>
RangeIndex: 418 entries, 0 to 417
Series name: Age
Non-Null Count  Dtype  
--------------  -----  
418 non-null    float64
dtypes: float64(1)
memory usage: 3.4 KB
```

# 43. For Credit dataset

# o Spot outliers in income using bivariate plot

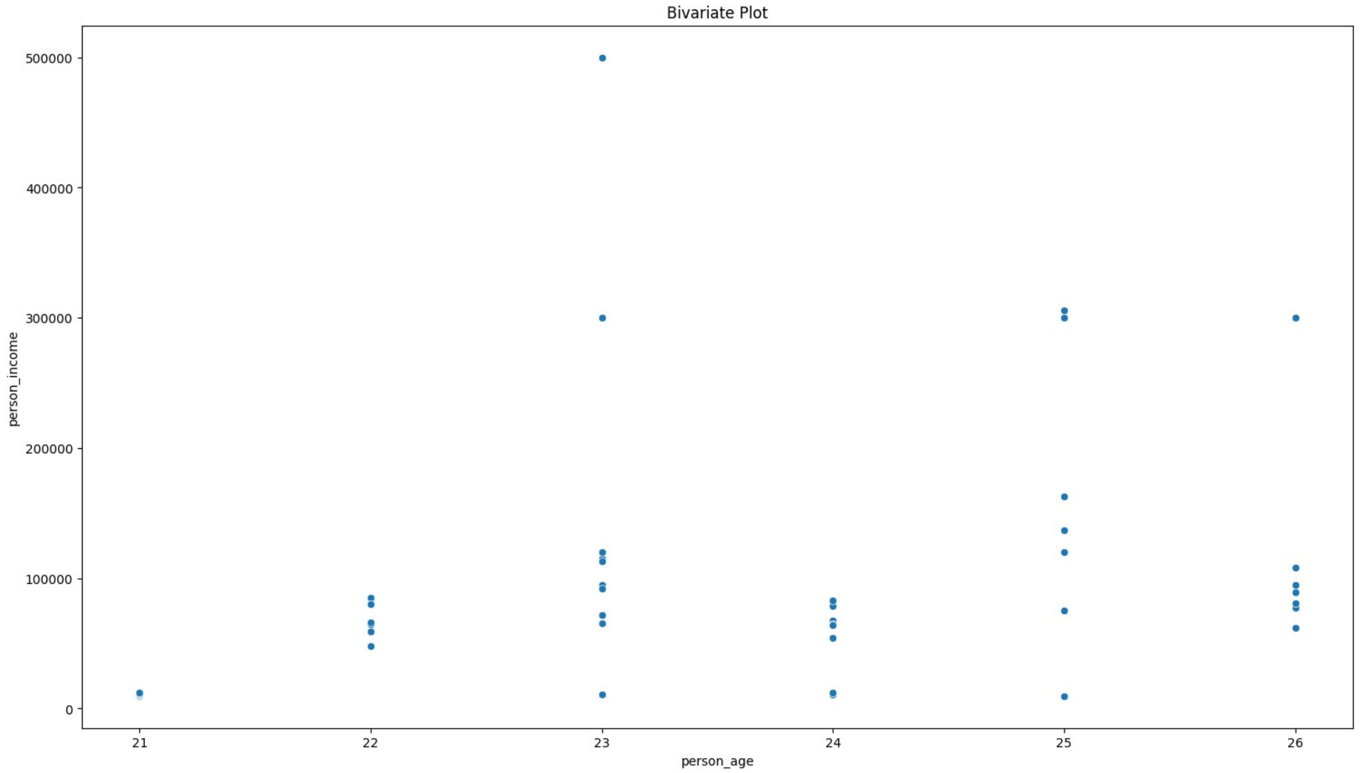
# o Spot outliers in any feature using boxplot

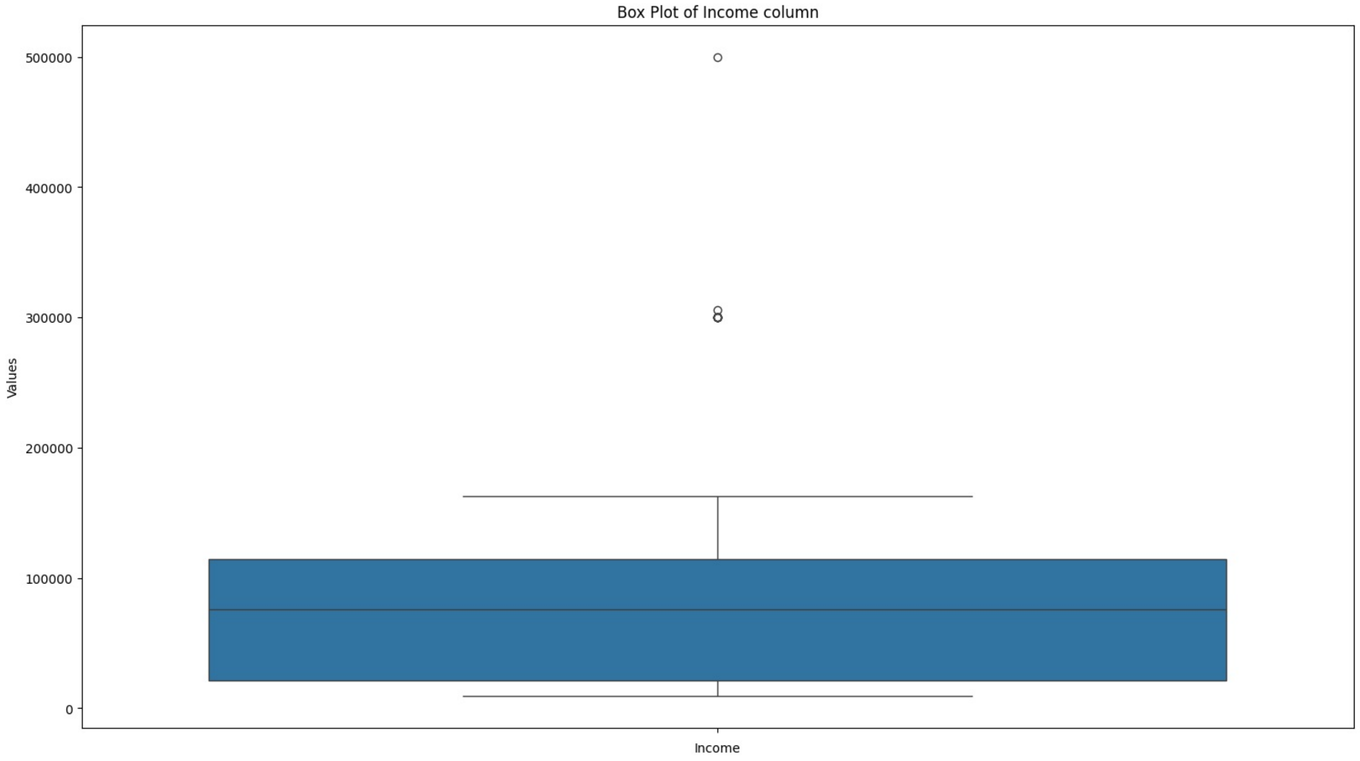
# o Detect outliers in any one feature using IQR method

# o Treat outliers using Imputation [Mean, Median, Zero]

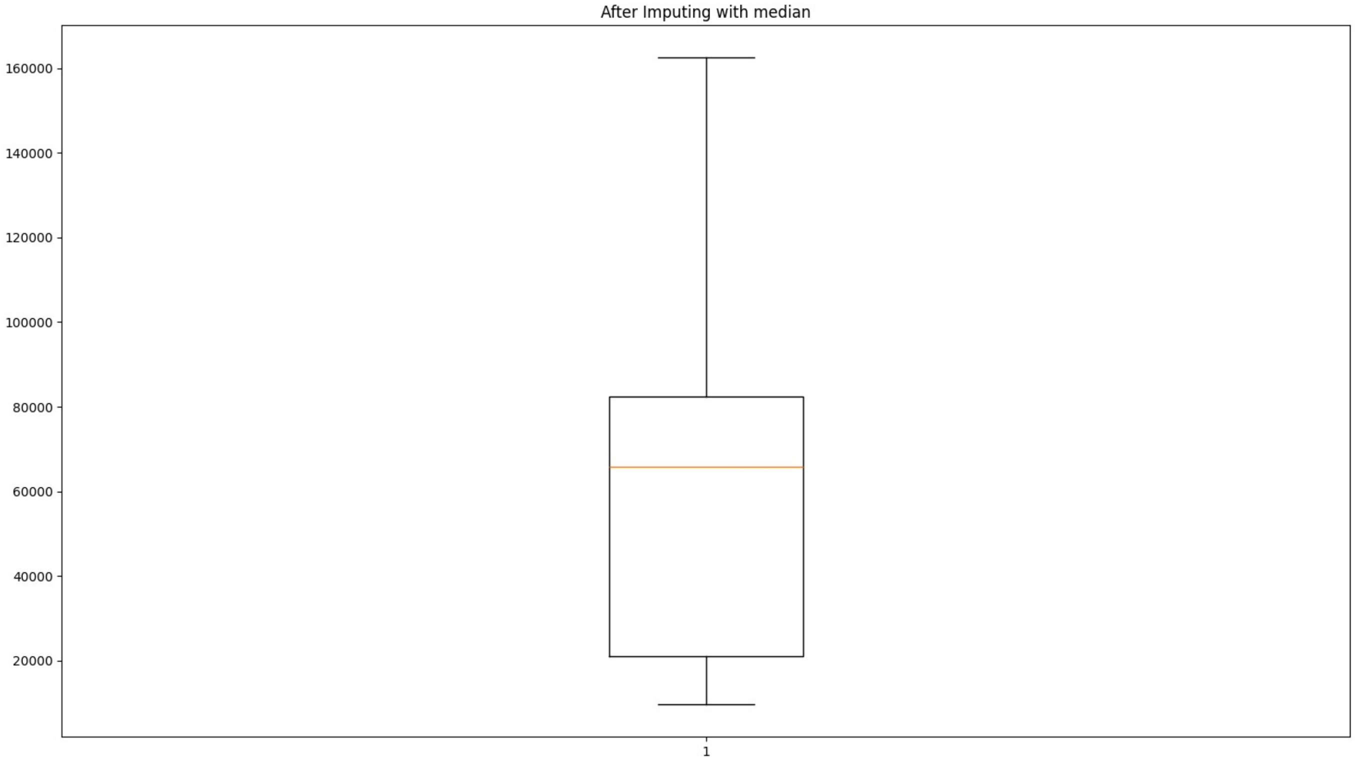
In [61]:

```
import pandas as p
import matplotlib.pyplot as m
import seaborn as s

da=p.read_csv("credit risk.csv").head(50)

#Bivariate Plot
m.figure(figsize=(18,10))
s.scatterplot(x=da['person_age'],y=da['person_income'],data=da)
m.title("Bivariate Plot")
m.show()

#Box Plot
m.figure(figsize=(18,10))
s.boxplot(da['person_income'])
m.xlabel('Income');m.ylabel('Values')
m.title("Box Plot of Income column")
m.show()

#Detect Outliers using IQR Method
inc=da['person_income']
q1=inc.quantile(0.25)
q3=inc.quantile(0.75)
iqr=q3-q1
low=q1-1.5*iqr
hig=q3+1.5*iqr

out=(inc <= low) | (inc>= hig)
print("Outliers:\n",out.sum())
#Impute Outliers using Mean
mean_in=inc[(inc >= low) & (inc <= hig)].mean()
da.loc[out, 'person_income'] = mean_ininc=da['person_income']
out1= (inc <= low) | (inc >= hig)
print("Outliers:\n",out1.sum())

medi=inc[(inc>= low) & (inc<= hig)].median()
da.loc[out1, 'person_income'] = medi

m.figure(figsize=(18,10))
m.boxplot(da['person_income'])
m.title("After Imputing with median")
m.show()

#Impute with Zero
da=p.read_csv("credit risk.csv").head(50)
inc=da['person_income']
out2=(inc <= low) | (inc>= hig)
print("Outliers:\n",out2.sum())

da.loc[out2, 'person_income'] = 0

m.figure(figsize=(18,10))
m.boxplot(da['person_income'])
m.title("After Imputing with Zero [0]")
m.show()

m.figure(figsize=(18,10))
m.boxplot(da['person_income'])
m.title("After Imputing with mean")
m.show()

# #Impute with Median
da=p.read_csv("credit risk.csv").head(50)
```

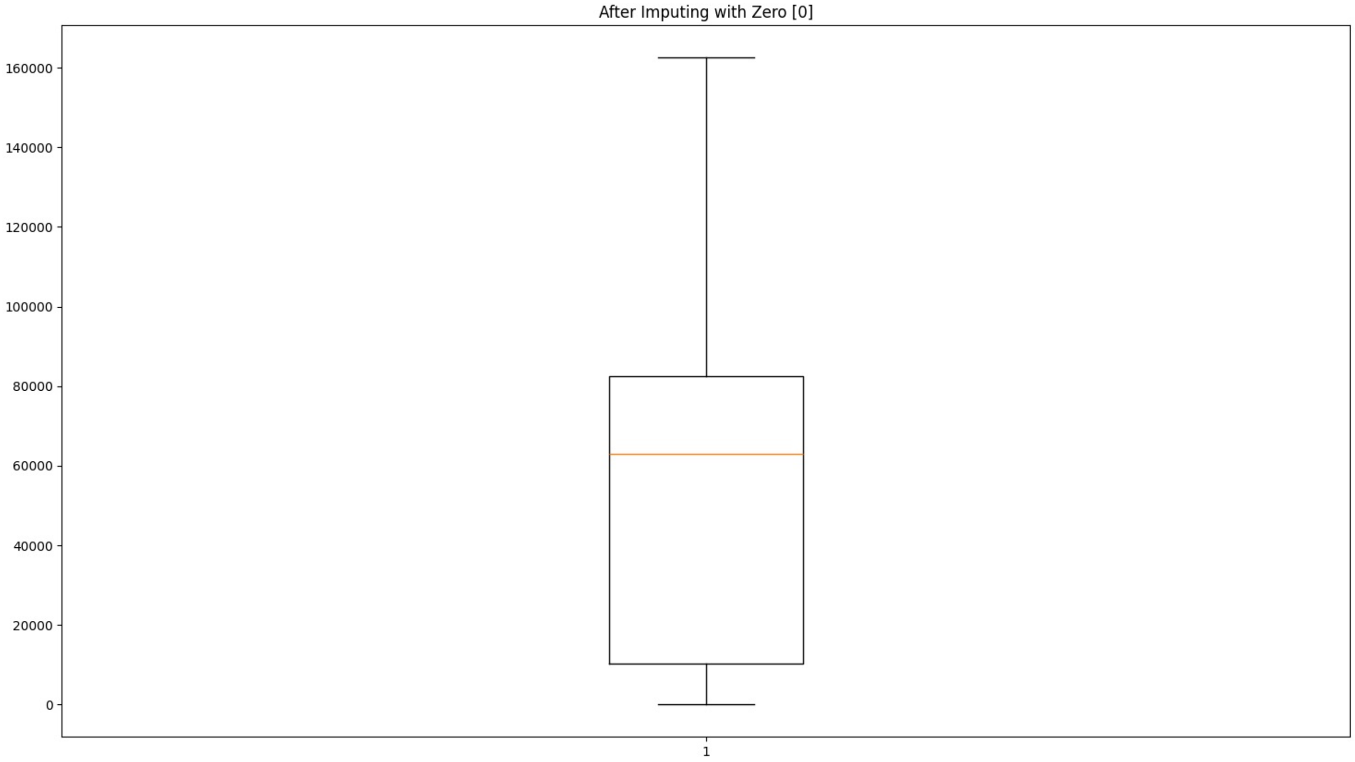


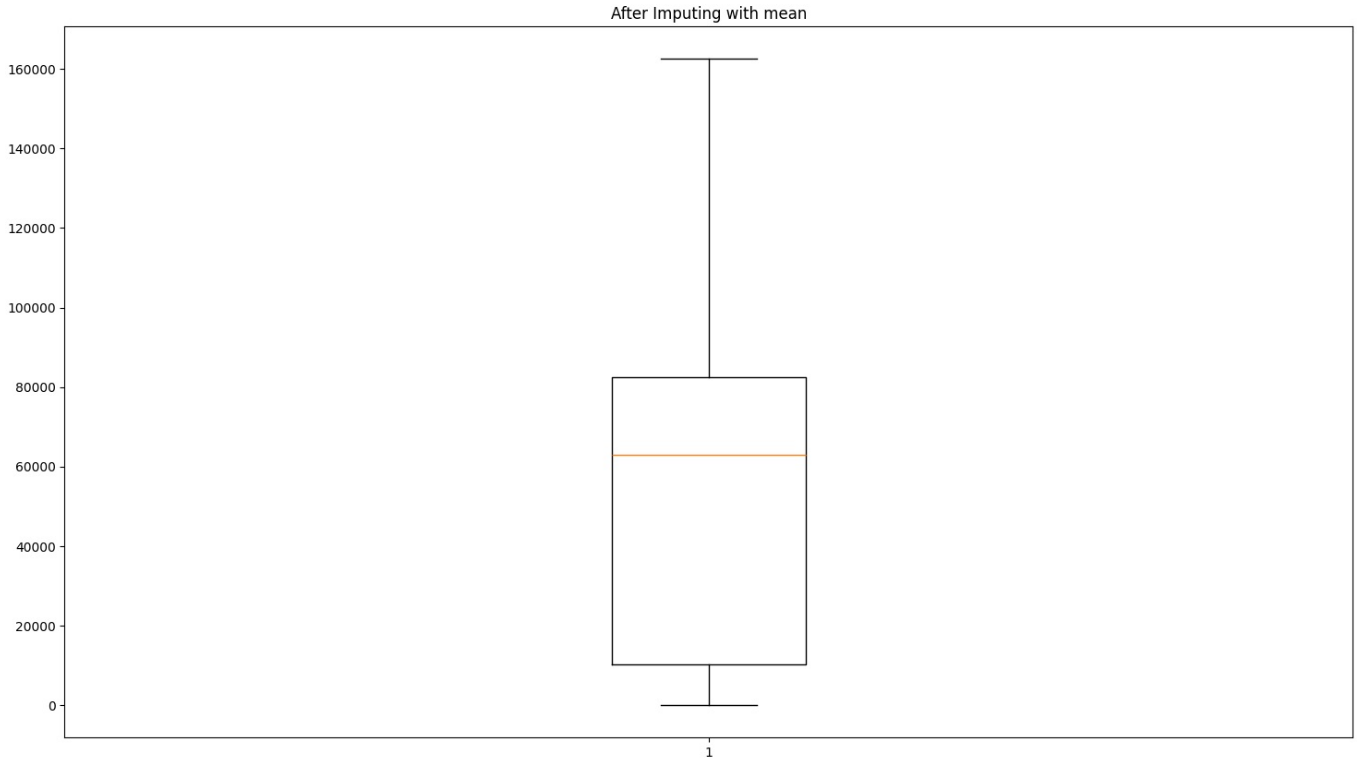


```
Outliers:
 8
Outliers:
 8
```


After Imputing with median
160000
140000
120000
100000
80000
60000
40000
20000
1

```
Outliers:
 8
```


After Imputing with Zero [0]
160000
140000
120000
100000
80000
60000
40000
20000
0
1


After Imputing with mean
160000
140000
120000
100000
80000
60000
40000
20000
0
1

In [ ]: