1. Perform a complete data inspection including – (20 marks)
    a. Missing Data Treatment
    b. Descriptive Statistics of each variable (Eg. Boxplot, Histogram etc.)
    c. Visualization of all continuous variables

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

"""
1.  Perform a complete data inspection including
a.  Missing Data Treatment
b.  Descriptive Statistics of each variable (Eg.
Boxplot, Histogram etc.)
c.  Visualization of all continuous variables

"""
df = pd.read_csv('Dataset_Day5.csv')
print(df.info())
skewness = df.skew()
print(skewness)
missing_value_percent = df.isna().sum() / len(df) *
100
print(missing_value_percent)
descriptive_statistics = df.describe()
print(descriptive_statistics)
# proportion of non-retail business acres per town
(boxplot is also done for continuous variables)
sns.boxplot(data=df["INDUS"], orient='v')
plt.plot()
# sns.histplot(df, x='NOX')
# plt.plot()
# for continuous variables
sns.pairplot(df[['CRIM', 'ZN', 'INDUS', 'NOX',
'RM', 'AGE', 'DIS', 'PTRATIO', 'B', 'LSTAT',
'MEDV']])
plt.show()
# plot between CRIM(per capita crime rate by town)
and B(1000(bk-0.63)^2 bk is proportion of black
people in town)
```

```
plt.scatter(df['B'], df['CRIM'])
plt.xlabel('B')
plt.ylabel('CRIM')
plt.title('Scatter Plot of B vs CRIM')
plt.show()
sns.heatmap(df.corr())
plt.show()
```

```
C:\Users\tejas\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\tejas\PycharmProjects\pythonProject\START\Q1.py
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 506 entries, 0 to 505
Data columns (total 14 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   CRIM     506 non-null    float64
 1   ZN       506 non-null    float64
 2   INDUS    506 non-null    float64
 3   CHAS     506 non-null    float64
 4   NOX      506 non-null    float64
 5   RM       506 non-null    float64
 6   AGE      506 non-null    float64
 7   DIS      506 non-null    float64
 8   RAD      506 non-null    float64
 9   TAX      506 non-null    float64
 10  PTRATIO  506 non-null    float64
 11  B        506 non-null    float64
 12  LSTAT    506 non-null    float64
 13  MEDV     506 non-null    float64
dtypes: float64(14)
memory usage: 55.5 KB
None
CRIM       5.223149
ZN         2.225666
INDUS      0.295022
CHAS       3.405904
NOX        0.729308
RM         0.403612
AGE       -0.598963
DIS        1.011781
RAD        1.004815
TAX        0.669956
PTRATIO   -0.802325
B         -2.890374
LSTAT      0.906460
MEDV       1.108098
dtype: float64
CRIM       0.0
ZN         0.0
INDUS      0.0
CHAS       0.0
NOX        0.0
RM         0.0
AGE        0.0
DIS        0.0
RAD        0.0
TAX        0.0
PTRATIO    0.0
B          0.0
LSTAT      0.0
MEDV       0.0
dtype: float64
              CRIM          ZN       INDUS  ...           B       LSTAT        MEDV
count  506.000000  506.000000  506.000000  ...  506.000000  506.000000  506.000000
mean     3.613524   11.363636   11.136779  ...  356.674032   12.653063   22.
```

```
56 532806
57 std      8.601545   23.322453    6.860353  ...   91.294864    7.141062    9.
   197104
58 min      0.006320    0.000000    0.460000  ...    0.320000    1.730000    5.
   000000
59 25%      0.082045    0.000000    5.190000  ...  375.377500    6.950000   17.
   025000
60 50%      0.256510    0.000000    9.690000  ...  391.440000   11.360000   21.
   200000
61 75%      3.677083   12.500000   18.100000  ...  396.225000   16.955000   25.
   000000
62 max     88.976200  100.000000   27.740000  ...  396.900000   37.970000   50.
   000000
63
64 [8 rows x 14 columns]
65
```

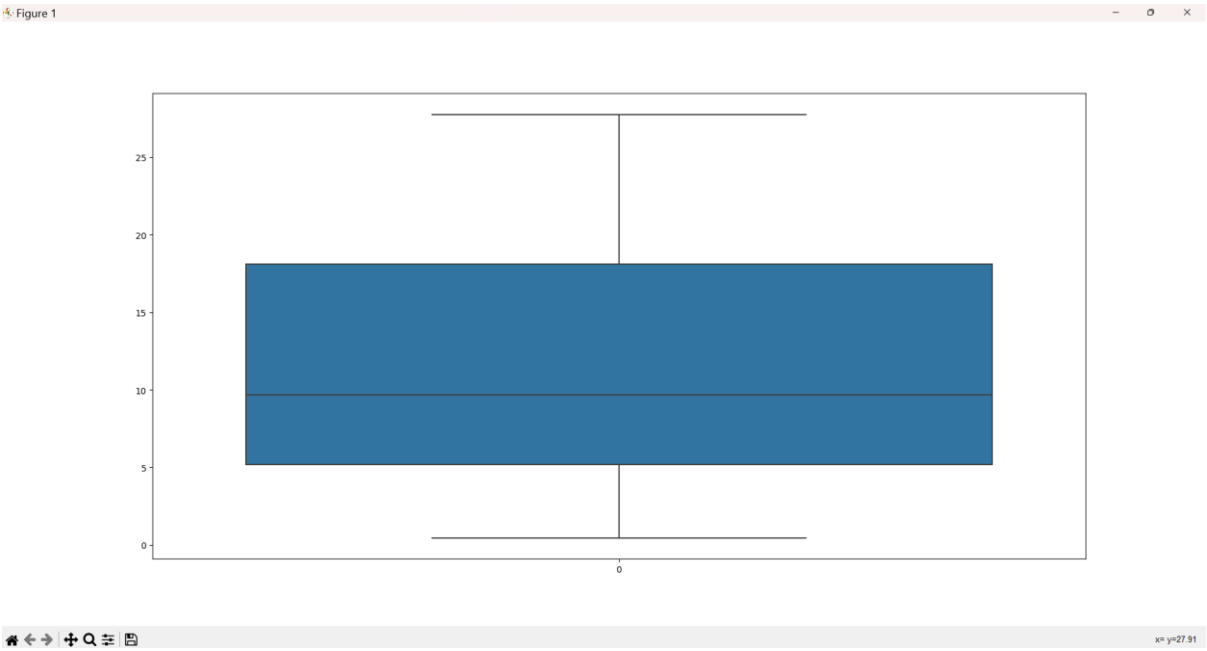
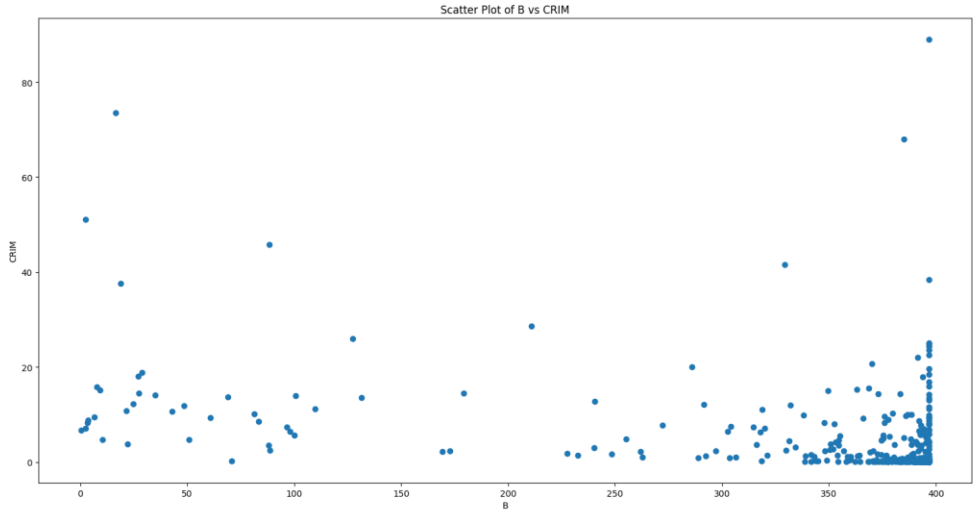
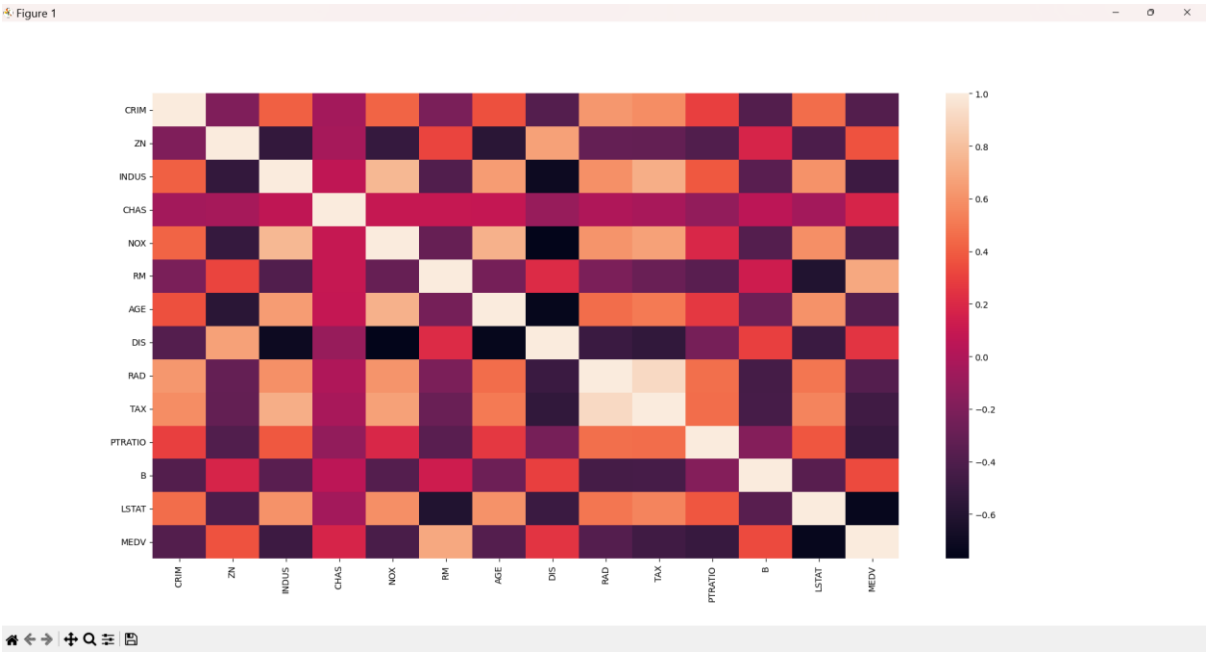


Figure 2

Figure 1





2.Create a simple linear regression model that quantitatively relates '*MEDV*' with '*RM*'. (10 marks)

a. Share the model performance metrics and print the full regression model with coefficients.

b. Use the model to predict the price of the house for '*RM*' = 7

```
2.import numpy as np
  import pandas as pd
  import matplotlib.pyplot as plt
  import seaborn as sns
  from sklearn.linear_model import
  LinearRegression
  from sklearn.model_selection import
  train_test_split
  from sklearn.metrics import mean_squared_error,
  r2_score, mean_absolute_error

  """
  2.  Create a simple linear regression model
  that quantitatively relates 'MEDV' with 'RM'.
  (10 marks)
  a.  Share the model performance metrics and
  print the full regression model with
  coefficients.
  b.  Use the model to predict the price of the
  house for 'RM' = 7
  """
```

```
df = pd.read_csv('Dataset_Day5.csv')
print(df.info())
X = df[['RM']]
y = df[['MEDV']]
# split
X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.2,
random_state=42)
# calling and fitting the model on training
data
lm = LinearRegression()
lm = lm.fit(X_train, y_train)
# predict using test set
y_pred = lm.predict(X_test)
print(lm.coef_)  # scale parameter
print(lm.intercept_)  # intercept parameter
print(r2_score(y_test, y_pred))
print(mean_absolute_error(y_test, y_pred))
print(mean_squared_error(y_test, y_pred))
rm_value = 7
price_prediction = lm.predict([[rm_value]])
print("Predicted price for 'RM' = 7:",
price_prediction[0])
```

```
C:\Users\tejas\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\
tejas\PycharmProjects\pythonProject\START\Q2.py
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 506 entries, 0 to 505
Data columns (total 14 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   CRIM     506 non-null    float64
 1   ZN       506 non-null    float64
 2   INDUS    506 non-null    float64
 3   CHAS     506 non-null    float64
 4   NOX      506 non-null    float64
 5   RM       506 non-null    float64
 6   AGE      506 non-null    float64
 7   DIS      506 non-null    float64
 8   RAD      506 non-null    float64
 9   TAX      506 non-null    float64
 10  PTRATIO  506 non-null    float64
 11  B        506 non-null    float64
 12  LSTAT    506 non-null    float64
 13  MEDV     506 non-null    float64
dtypes: float64(14)
memory usage: 55.5 KB
None
[[9.34830141]]
[-36.2463189]
0.3707569232254778
4.478335832064147
46.144775347317264
Predicted price for 'RM' = 7: [29.19179095]
C:\Users\tejas\PycharmProjects\pythonProject\venv\Lib\site-packages\sklearn\
base.py:464: UserWarning: X does not have valid feature names, but
LinearRegression was fitted with feature names
  warnings.warn(

Process finished with exit code 0
```

3.Create a simple linear regression model that quantitatively relates '*MEDV*' with '*DIS*'. (10 marks)

a. Share the model performance metrics and print the full regression model with coefficients.

b. Use the model to predict the price of the house for '*DIS*' = 15

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import
train_test_split
from sklearn.metrics import mean_squared_error,
r2_score, mean_absolute_error
```

```
"""

3.  Create a simple linear regression model that
quantitatively relates ‘MEDV’ with ‘DIS’. (10
marks)
a.  Share the model performance metrics and print
the full regression model with coefficients.
b.  Use the model to predict the price of the
house for ‘DIS’ = 15

"""
df = pd.read_csv('Dataset_Day5.csv')
print(df.info())
X = df[['DIS']]
y = df['MEDV']
X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.2,
random_state=42)
lm = LinearRegression()
lm = lm.fit(X_train, y_train)
y_pred = lm.predict(X_test)
print(lm.coef_)  # scale parameter
print(lm.intercept_)  # intercept parameter
print(r2_score(y_test, y_pred))
print(mean_absolute_error(y_test, y_pred))
print(mean_squared_error(y_test, y_pred))
DIS_value = 7
price_prediction = lm.predict([[DIS_value]])
print("Predicted price for 'DIS' = 15:",
price_prediction[0])
```

```
C:\Users\tejas\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\tejas\PycharmProjects\pythonProject\START\Q3.py
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 506 entries, 0 to 505
Data columns (total 14 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   CRIM     506 non-null    float64
 1   ZN       506 non-null    float64
 2   INDUS    506 non-null    float64
 3   CHAS     506 non-null    float64
 4   NOX      506 non-null    float64
 5   RM       506 non-null    float64
 6   AGE      506 non-null    float64
 7   DIS      506 non-null    float64
 8   RAD      506 non-null    float64
 9   TAX      506 non-null    float64
 10  PTRATIO  506 non-null    float64
 11  B        506 non-null    float64
 12  LSTAT    506 non-null    float64
 13  MEDV     506 non-null    float64
dtypes: float64(14)
memory usage: 55.5 KB
None
[1.0295094]
18.875962058273238
0.07332042069244615
5.967846118518974
67.95691932803946
Predicted price for 'DIS' = 15: 26.08252784851876
C:\Users\tejas\PycharmProjects\pythonProject\venv\Lib\site-packages\sklearn\base.py:464: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(

Process finished with exit code 0
```