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

```python
dataset = pd.read_csv("HousingData.csv")
```

```python
dataset
```

```
        CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD  TAX  
\
0    0.00632  18.0   2.31   0.0  0.538  6.575  65.2  4.0900    1  296   

1    0.02731   0.0   7.07   0.0  0.469  6.421  78.9  4.9671    2  242   

2    0.02729   0.0   7.07   0.0  0.469  7.185  61.1  4.9671    2  242   

3    0.03237   0.0   2.18   0.0  0.458  6.998  45.8  6.0622    3  222   

4    0.06905   0.0   2.18   0.0  0.458  7.147  54.2  6.0622    3  222   

..       ...   ...    ...   ...    ...    ...   ...     ...  ...  ...   

501  0.06263   0.0  11.93   0.0  0.573  6.593  69.1  2.4786    1  273   

502  0.04527   0.0  11.93   0.0  0.573  6.120  76.7  2.2875    1  273   

503  0.06076   0.0  11.93   0.0  0.573  6.976  91.0  2.1675    1  273   

504  0.10959   0.0  11.93   0.0  0.573  6.794  89.3  2.3889    1  273   

505  0.04741   0.0  11.93   0.0  0.573  6.030   NaN  2.5050    1  273   


     PTRATIO       B  LSTAT  MEDV  
0       15.3  396.90   4.98  24.0  
1       17.8  396.90   9.14  21.6  
2       17.8  392.83   4.03  34.7  
3       18.7  394.63   2.94  33.4  
4       18.7  396.90    NaN  36.2  
..       ...     ...    ...   ...  
501     21.0  391.99    NaN  22.4  
502     21.0  396.90   9.08  20.6  
503     21.0  396.90   5.64  23.9  
504     21.0  393.45   6.48  22.0  
505     21.0  396.90   7.88  11.9  

[506 rows x 14 columns]
```

```python
dataset.isnull().sum()
```

```
CRIM       20
ZN         20
INDUS      20
CHAS       20
NOX         0
RM          0
AGE        20
DIS         0
RAD         0
TAX         0
PTRATIO     0
B           0
LSTAT      20
MEDV        0
dtype: int64
```

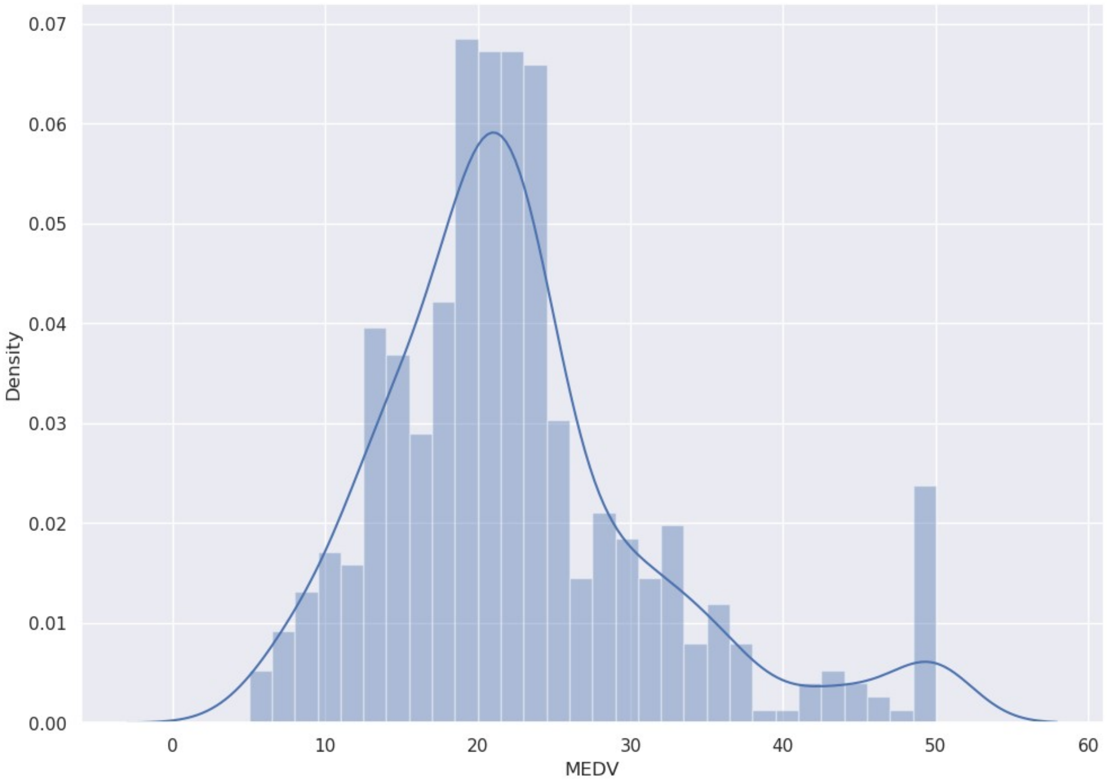
```
dataset.fillna(method = 'ffill', inplace = True)
```

```
dataset.isnull().sum()
```

```
CRIM       0
ZN         0
INDUS      0
CHAS       0
NOX        0
RM         0
AGE        0
DIS        0
RAD        0
TAX        0
PTRATIO    0
B          0
LSTAT      0
MEDV       0
dtype: int64
```

```
sns.set(rc={'figure.figsize':(11.7,8.27)})
sns.distplot(dataset['MEDV'], bins=30)
plt.show()
```

```
/home/mmcoe/anaconda3/lib/python3.9/site-packages/seaborn/
distributions.py:2619: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



```
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
```

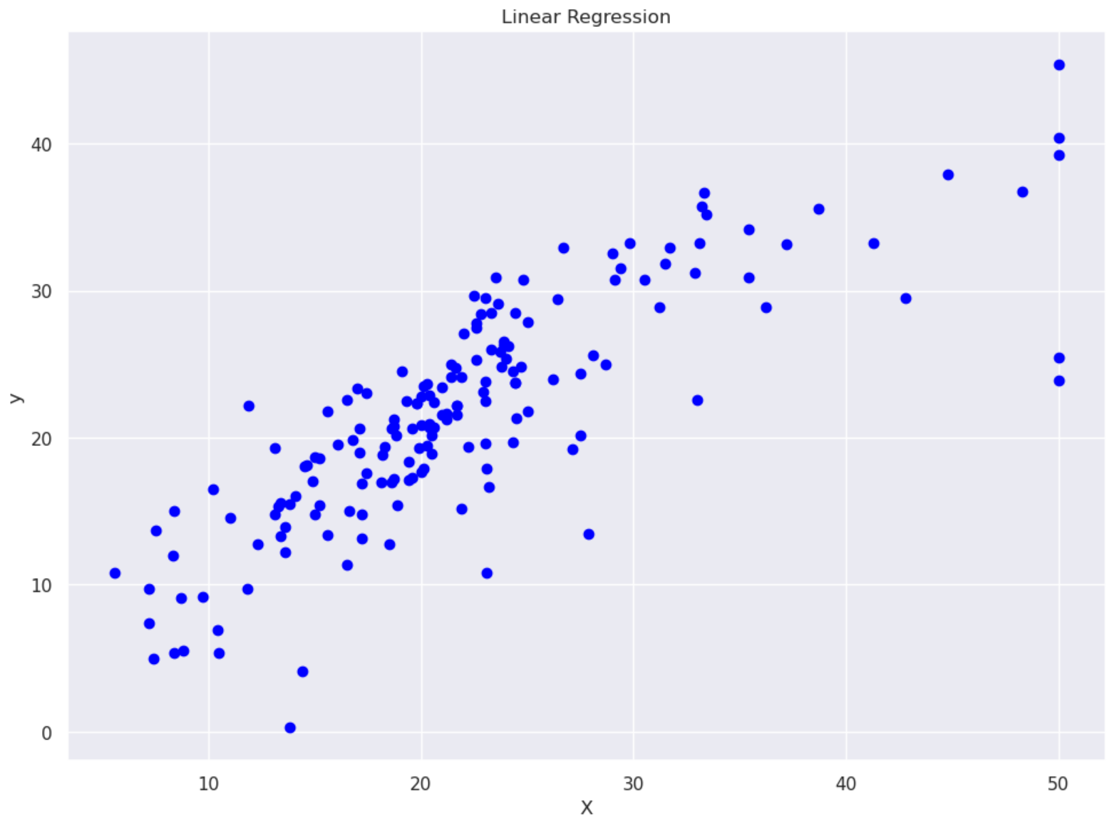
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=1/3, random_state = 0)
```

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

```
LinearRegression()
```

```
y_pred = regressor.predict(X_test)
```

```
plt.scatter(y_test, y_pred, color = "blue")
#plt.plot(y_test, regressor.predict(y_pred), color = "green")
plt.title("Linear Regression")
plt.xlabel("X")
plt.ylabel("y")
plt.show()
```



```
print(X_train.shape)
```

```
(337, 13)
```

```
print(y_train.shape)
```

```
(337,)
```

```
print(X_test.shape)
```

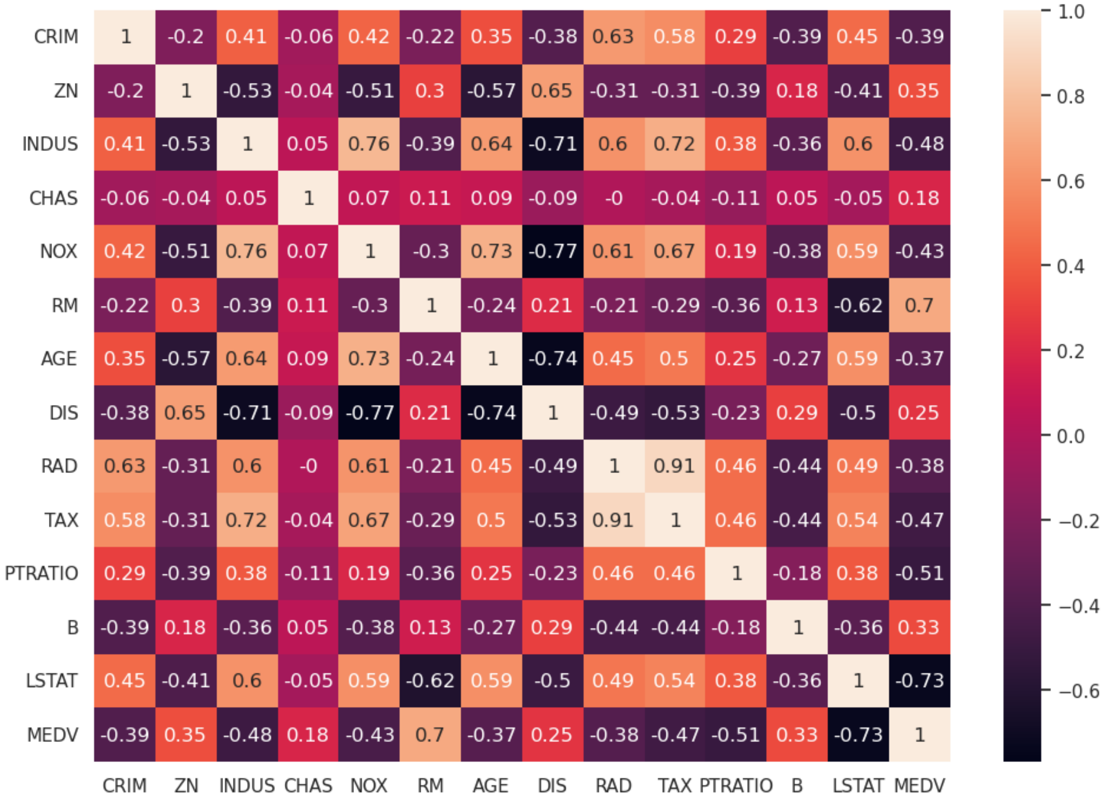
```
(169, 13)
```

```
print(y_test.shape)
```

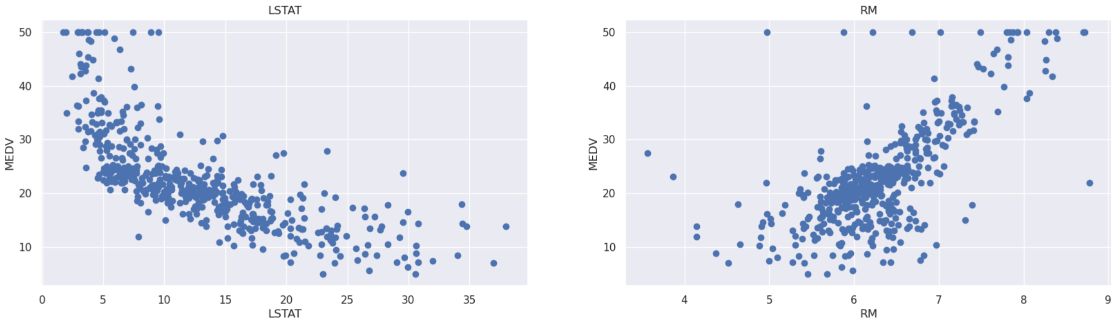
```
(169,)
```

```
correlation_matrix = dataset.corr().round(2)
sns.heatmap(data=correlation_matrix, annot=True)
```

```
<AxesSubplot:>
```



```
plt.figure(figsize=(20, 5))

features = ['LSTAT', 'RM']
target = dataset['MEDV']

for i, col in enumerate(features):
    plt.subplot(1, len(features) , i+1)
    x = dataset[col]
    y = target
    plt.scatter(x, y, marker='o')
    #plt.plot(X, y, color = "green")
    plt.title(col)
    plt.xlabel(col)
    plt.ylabel('MEDV')
```



```
X_rooms = dataset.MEDV
y_price = dataset.RM


X_rooms = np.array(X_rooms).reshape(-1,1)
y_price = np.array(y_price).reshape(-1,1)

print(X_rooms.shape)
print(y_price.shape)
```

```
(506, 1)
(506, 1)
```

```
X_train_1, X_test_1, Y_train_1, Y_test_1 = train_test_split(X_rooms,
y_price, test_size = 0.2, random_state=0)

print(X_train_1.shape)
print(X_test_1.shape)
print(Y_train_1.shape)
print(Y_test_1.shape)
```

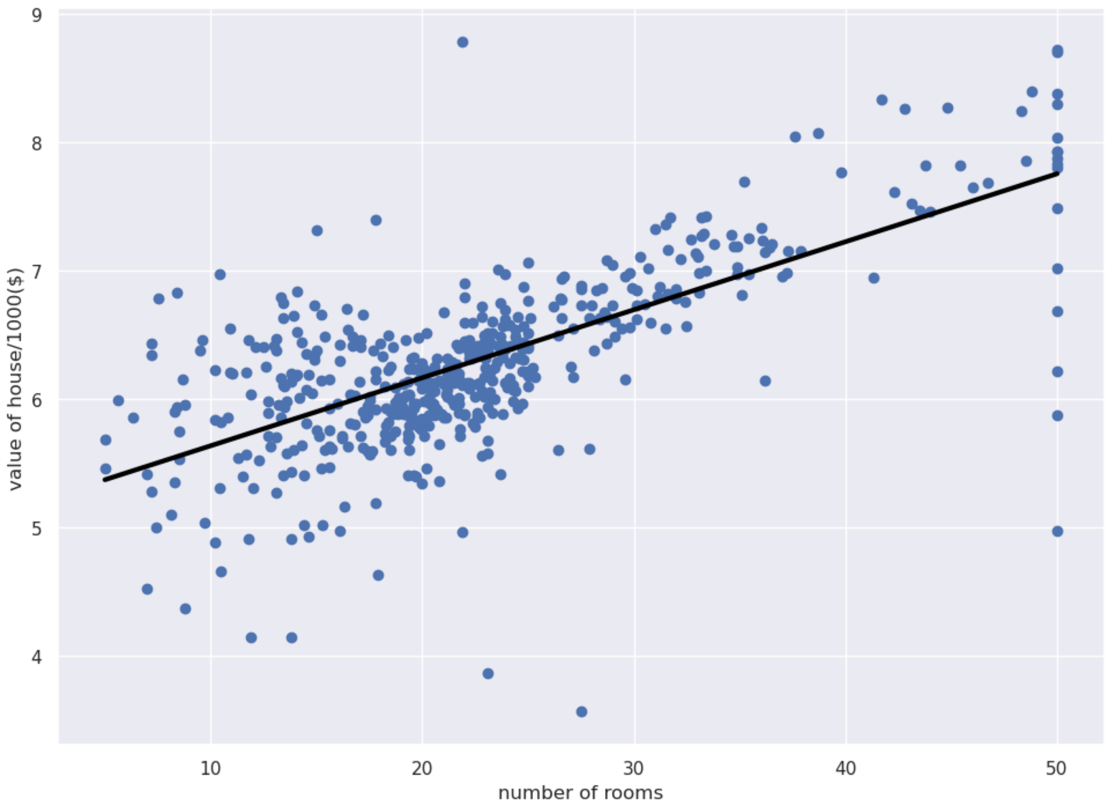
```
(404, 1)
(102, 1)
(404, 1)
(102, 1)
```

```
reg_1 = LinearRegression()
reg_1.fit(X_train_1, Y_train_1)

y_train_predict_1 = reg_1.predict(X_train_1)
rmse = (np.sqrt(mean_squared_error(Y_train_1, y_train_predict_1)))
r2 = round(reg_1.score(X_train_1, Y_train_1),2)

print("The model performance for training set")
print("--------------------------------------")
print('RMSE is {}'.format(rmse))
print('R2 score is {}'.format(r2))
print("\n")
```

```
The model performance for training set
--------------------------------------
RMSE is 0.4920435120933795
R2 score is 0.5
```

```
prediction_space = np.linspace(min(X_rooms), max(X_rooms)).reshape(-
1,1)
plt.scatter(X_rooms,y_price)
plt.plot(prediction_space, reg_1.predict(prediction_space), color =
'black', linewidth = 3)
```

```
plt.ylabel('value of house/1000($)')
plt.xlabel('number of rooms')
plt.show()
```



```
dataset
```

```
          CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD  TAX
\
0      0.00632  18.0   2.31   0.0  0.538  6.575  65.2  4.0900    1  296

1      0.02731   0.0   7.07   0.0  0.469  6.421  78.9  4.9671    2  242

2      0.02729   0.0   7.07   0.0  0.469  7.185  61.1  4.9671    2  242

3      0.03237   0.0   2.18   0.0  0.458  6.998  45.8  6.0622    3  222

4      0.06905   0.0   2.18   0.0  0.458  7.147  54.2  6.0622    3  222

..         ...   ...    ...   ...    ...    ...   ...     ...  ...  ...

501    0.06263   0.0  11.93   0.0  0.573  6.593  69.1  2.4786    1  273

502    0.04527   0.0  11.93   0.0  0.573  6.120  76.7  2.2875    1  273

503    0.06076   0.0  11.93   0.0  0.573  6.976  91.0  2.1675    1  273
```

```
504  0.10959   0.0  11.93   0.0  0.573  6.794  89.3  2.3889    1  273

505  0.04741   0.0  11.93   0.0  0.573  6.030  89.3  2.5050    1  273


     PTRATIO       B  LSTAT  MEDV
0       15.3  396.90   4.98  24.0
1       17.8  396.90   9.14  21.6
2       17.8  392.83   4.03  34.7
3       18.7  394.63   2.94  33.4
4       18.7  396.90   2.94  36.2
..       ...     ...    ...   ...
501     21.0  391.99  14.33  22.4
502     21.0  396.90   9.08  20.6
503     21.0  396.90   5.64  23.9
504     21.0  393.45   6.48  22.0
505     21.0  396.90   7.88  11.9

[506 rows x 14 columns]
```