15.3 396.90

396.90

392.83

394.63

18.7 396.90

17.8

17.8

18.7

b 1stat medv

4.98

9.14

2.94

5.33

24.0

21.6

34.7

33.4

36.2

```
6/29/23, 9:18 PM
   import pandas as pd
   import numpy as np
   import matplotlib.pyplot as plt
   import seaborn as sns
   %matplotlib inline
   boston_dataset=pd.read_csv("/content/BostonHousing.csv")
   boston_dataset.head()
    ₽
                                                            dis rad tax ptratio
                          indus
                                chas
                                        nox
                                                rm
                                                    age
         0.00632
                                    0 0.538
                                             6.575 65.2 4.0900
                    18.0
                            2.31
         1 0.02731
                     0.0
                            7.07
                                    0 0.469 6.421 78.9 4.9671
                                                                   2 242
            0.02729
                            7.07
                                    0 0.469
                                             7.185 61.1 4.9671
                                                                   2 242
         3 0.03237
                      0.0
                            2.18
                                    0 0.458
                                             6.998 45.8 6.0622
                                                                   3 222
         4 0.06905
                     0.0
                            2.18
                                    0 0.458 7.147 54.2 6.0622
                                                                   3 222
   boston_dataset.isnull().sum()
        crim
                   0
                   0
        indus
                   0
        chas
                   a
        nox
                   0
        rm
        age
                   0
        dis
                   0
        rad
                   0
        tax
        ptratio
                   0
                   0
                   0
        lstat
        medv
        dtype: int64
```

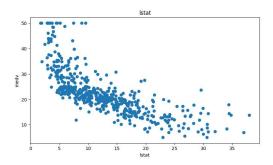
correlation matrix = boston dataset.corr().round(2) sns.heatmap(data=correlation_matrix,annot=True)

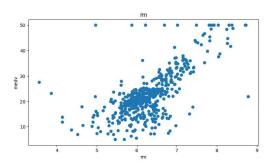
plt.figure(figsize=(20, 5))

```
<Axes: >
                                                                         1.0
   crim - 1 -0.20.410.060.420.220.350.380.630.580.290.390.460.39
     zn --0.2 1 0.530.040.520.310.570.660.310.310.390.180.410.30
                                                                         0.8
  indus -0.41-0.53 1 0.060.760.390.640.710.6 0.720.380.36 0.6-0.48
                                                                         0.6
   chas -0.060.040.06 1 0.090.090.09-0.1-0.010.040.120.050.050.18
    nox -0.420.520.760.09 1 -0.30.730.770.610.670.190.380.590.43
                                                                         0.4
    m -0.220.31-0.390.09-0.3 1 -0.240.21-0.210.290.360.130.61 0.7
    age -0.350.570.640.090.730.24 1 0.750.460.510.260.270.6-0.38
                                                                         0.2
     dis -0.38<mark>0.66</mark>0.71-0.1-0.770.210.75 1 0.490.530.230.29-0.50.25
                                                                         0.0
    rad -0.630.310.6-0.010.610.210.460.49 1 0.910.460.440.490.38
    tax -0.580.310.720.040.670.290.510.530.91 1 0.460.440.540.47
                                                                          -0.2
 ptratio -0.290.390.380.120.190.360.260.230.460.46 1 0.180.370.5
                                                                          -0.4
      b -0.390.180.360.050.380.130.270.290.440.440.18 1 -0.370.3
   lstat -0.460.410.6-0.050.590.610.6-0.50.490.540.370.37 1 0.7
                                                                          -0.6
  medy -0.390.360.480.180.430.7-0.380.250.380.470.510.330.74
```

```
features = ['lstat', 'rm']
   target = boston_dataset['medv']
https://colab.research.google.com/drive/1gxknlpJo0B7-TvmTSptV8IJ7PooHhatj#printMode=true
```

```
for i, col in enumerate(features):
   plt.subplot(1, len(features) , i+1)
   x = boston_dataset[col]
   y = target
   plt.scatter(x, y, marker='o')
   plt.title(col)
   plt.xlabel(col)
   plt.ylabel('medv')
```





```
X = pd.DataFrame(np.c_[boston_dataset['lstat'], boston_dataset['rm']], columns = ['lstat','rm'])
Y = boston_dataset['medv']
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, random_state=5)
print(X_train.shape)
print(X_test.shape)
print(Y_train.shape)
print(Y_test.shape)
     (404, 2)
     (102, 2)
     (404,)
     (102,)
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
{\tt import\ statsmodels.api\ as\ sm}
lin_model = LinearRegression()
lin_model.fit(X_train, Y_train)
     ▼ LinearRegression
     LinearRegression()
model = sm.OLS(Y_train, X_train).fit()
print(model.summary())
                                      OLS Regression Results
    Dep. Variable:
                                      medv
                                            R-squared (uncentered):
    Model:
                                      OLS
                                            Adj. R-squared (uncentered):
                                                                                        0.947
    Method:
                            Least Squares
                                             F-statistic:
                                                                                        3581.
    Date:
                          Thu, 29 Jun 2023
                                             Prob (F-statistic):
                                                                                    6.67e-257
                                            Log-Likelihood:
                                  14:36:10
                                                                                      -1272.2
    Time:
    No. Observations:
                                       404
                                             AIC:
                                                                                        2548.
    Df Residuals:
                                       402
                                             BIC:
                                                                                        2556.
```

Df Model: Covariance Type: nonrobust

covariance Type.		nom obuse				
	coef	std err	t	P> t	[0.025	0.975]
lstat rm	-0.6911 4.9699	0.036 0.081	-19.367 61.521	0.000 0.000	-0.761 4.811	-0.621 5.129
Omnibus: Prob(Omnibus Skew: Kurtosis:):	0	.000 Jarq .370 Prob	in-Watson: ue-Bera (JB) (JB): . No.):	2.063 389.671 2.42e-85 4.70

- Notes: [1] R^2 is computed without centering (uncentered) since the model does not contain a constant.
- [2] Standard Errors assume that the covariance matrix of the errors is correctly specified.