Name : Mohit Manish Bhavsar

Roll No : 20U437

Div : 4

```
import pandas as pd
df=pd.read_csv('https://raw.githubusercontent.com/selva86/datasets/master/BostonHousing.cs
df
```

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	b
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90
...	...	...	...	...	...	...	...	...	...	...	...	...
501	0.06263	0.0	11.93	0	0.573	6.593	69.1	2.4786	1	273	21.0	391.99
502	0.04527	0.0	11.93	0	0.573	6.120	76.7	2.2875	1	273	21.0	396.90
503	0.06076	0.0	11.93	0	0.573	6.976	91.0	2.1675	1	273	21.0	396.90
504	0.10959	0.0	11.93	0	0.573	6.794	89.3	2.3889	1	273	21.0	393.45
505	0.04741	0.0	11.93	0	0.573	6.030	80.8	2.5050	1	273	21.0	396.90

506 rows × 14 columns

```
df.shape
```

```
(506, 14)
```

```
df.isna().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
```

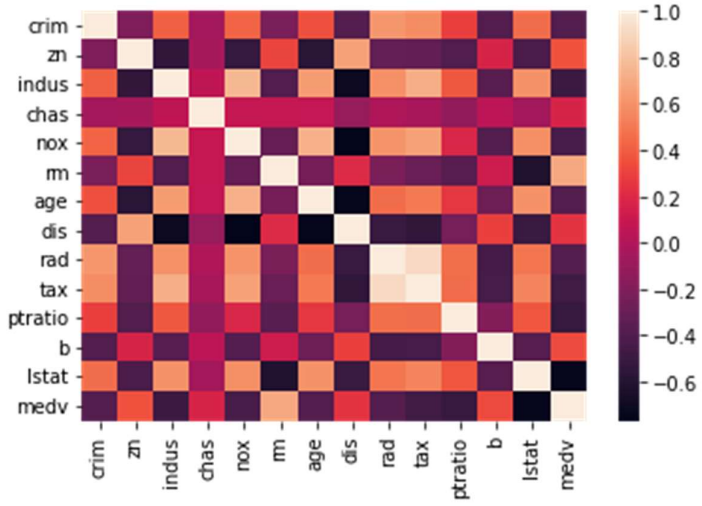
```
df.info()
```

```
<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    int64  
 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    int64  
 9   tax      506 non-null    int64  
 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(11), int64(3)
memory usage: 55.5 KB
```

```
import seaborn as sns
sns.heatmap(df.corr())
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f156470a310>
```

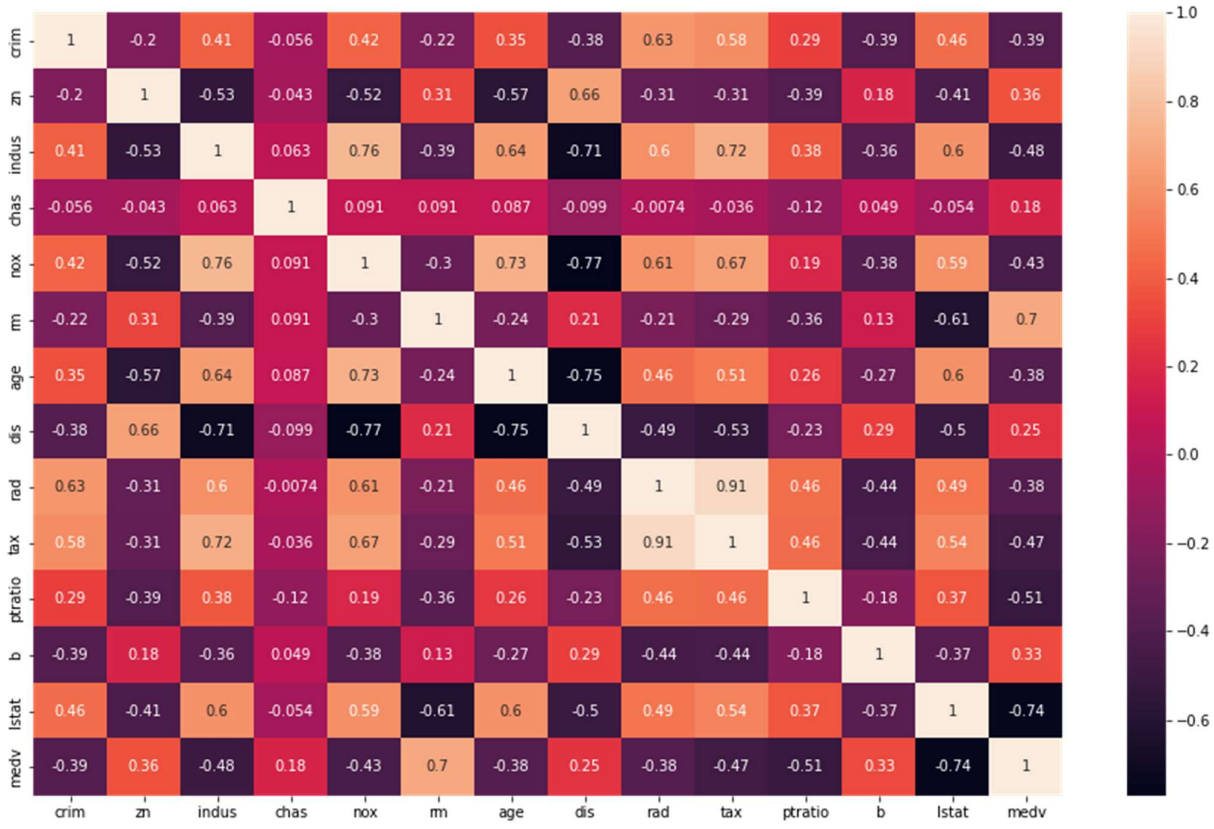



```
sns.heatmap(df.corr(),annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f155bde6650>
```

```
import matplotlib.pyplot as plt
fig,ax=plt.subplots(figsize=(16,10))
sns.heatmap(df.corr(),annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1559cfc750>
```



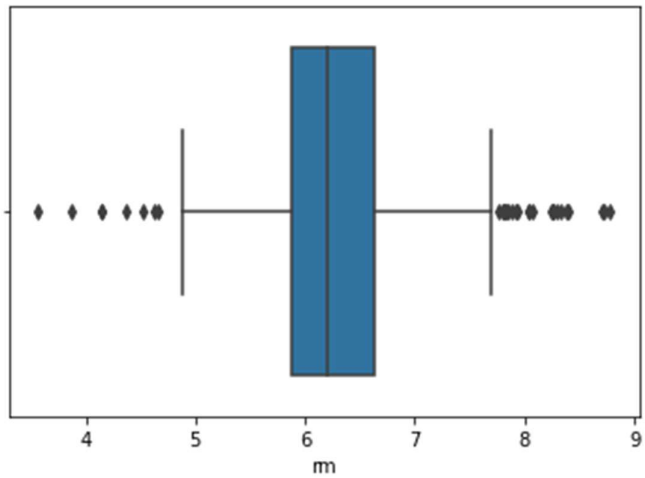

```
data=df[['rm','lstat','medv']]
data
```

	rm	lstat	medv
0	6.575	4.98	24.0
1	6.421	9.14	21.6
2	7.185	4.03	34.7
3	6.998	2.94	33.4
4	7.147	5.33	36.2
...	...	...	...
501	6.593	9.67	22.4
502	6.120	9.08	20.6
503	6.976	5.64	23.9
504	6.794	6.48	22.0
505	6.030	7.88	11.9

506 rows × 3 columns

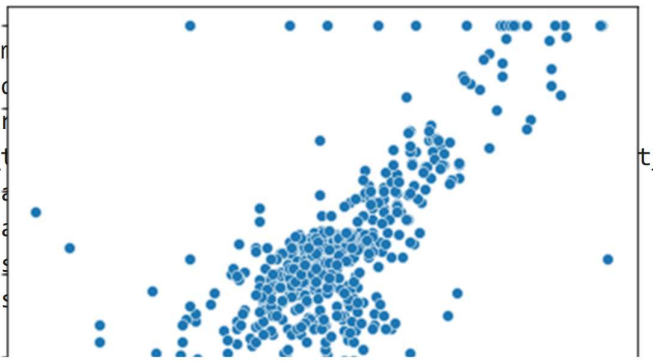
```
sns.boxplot(x=data['rm'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1559cfc310>
```




```
sns.scatterplot(data=data,x='rm',y='medv')
```

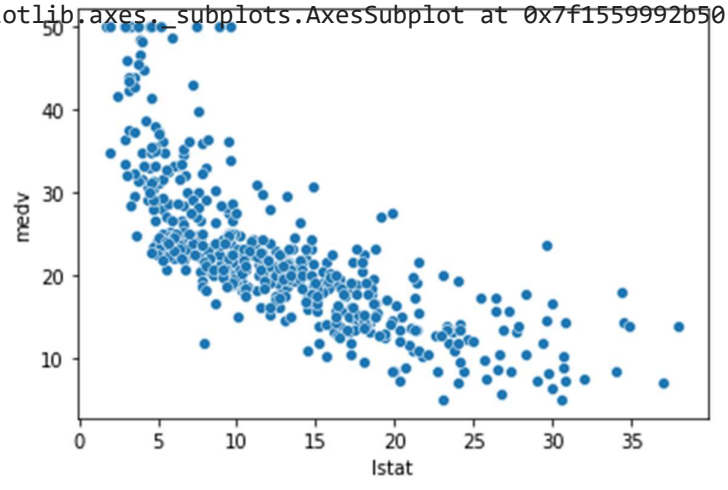
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1559bf0910>
```



```
x=data[['rm
y=data['med
from sklear
x_train,x_t                                                    t_size=0.2,random_state=42)
print(x_tra
print(y_tra
print(x_tes
print(y_tes
```

```
sns.scatterplot(data=data,x='lstat',y='medv')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1559992b50>
```



```
332    19.4
423    13.4
19     18.2
Name: medv, dtype: float64
```

```
from sklearn.linear_model import LinearRegression
model=LinearRegression().fit(x_train,y_train)
output=model.predict(x_test)
print(output)
```

```
[25.50510964 30.8592405  17.42725985 25.81325491 19.64929972 22.90317032
 16.88571841 14.61917747 22.06066668 20.03326882 17.34725148 18.15555053
 -2.53419257 22.41228621 19.92807359 26.93027912 17.16881388  3.46139894
 37.31795503 18.49325989 26.09287374 27.0359285  13.08646469 26.07395803
 19.05197872 14.34439003 22.50507619 21.09395936 17.80756143 18.90537548
 17.3189494  26.72355799 27.62430957 19.01752219 15.4809415  17.37191951
 32.8765884  22.03500171 20.02675295 25.5193256  12.24468569 28.82657024
 38.2438306  18.45117927 25.65604941 16.5588818  15.7090024  26.97797141
 19.55868547 28.94200451 20.60534552 31.33646277 17.88508649 28.36639373
 34.9844153  23.96269158 19.65312104 31.54864539 24.99070281 15.11477808
 27.10508448 32.73300584 29.58795549 18.58249363 28.64129064 10.75991895
 20.69510169 26.34948505 29.46875554 16.79445551 18.83002976 28.15270076
 13.10437033 25.07663224 23.21876915  6.6118755  22.06337978 36.66642406
 18.59482469 10.58587713 22.76126683 10.15832067 22.42113161  7.43849725
 21.66222968 27.82546384 22.3948234  27.34654035 26.17473533 22.20241637
 22.66415707  8.2959451  22.97660538 20.3782187  11.11664989 23.94904829
 24.10216559 -0.28767556 20.05222704 19.11836595 20.89746052 24.85789666]
```

```
print(y_test)
```

```
173    23.6
274    32.4
491    13.6
72     22.8
452    16.1
       ...
412    17.9
436     9.6
411    17.2
86     22.5
75     21.4
Name: medv, Length: 102, dtype: float64
```

```
from sklearn.metrics import mean_absolute_error
print('MAE:',mean_absolute_error(y_test,output))
print("Model Score:",model.score(x_test,y_test))
```

```
MAE: 3.8987597213823584
Model Score: 0.5739577415025858
```