dl-1

April 11, 2025

[7]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

[8]:
```
BostonTrain = pd.read_csv("boston_test.csv")
```

[9]:
```
BostonTrain.head()
```

[9]:

	ID	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat
0	3	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03
1	6	0.02985	0.0	2.18	0	0.458	6.430	58.7	6.0622	3	222	18.7	394.12	5.21
2	8	0.14455	12.5	7.87	0	0.524	6.172	96.1	5.9505	5	311	15.2	396.90	19.15
3	9	0.21124	12.5	7.87	0	0.524	5.631	100.0	6.0821	5	311	15.2	386.63	29.93
4	10	0.17004	12.5	7.87	0	0.524	6.004	85.9	6.5921	5	311	15.2	386.71	17.10

[10]:
```
BostonTrain.info()
BostonTrain.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 173 entries, 0 to 172
Data columns (total 14 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   ID       173 non-null    int64
 1   crim     173 non-null    float64
 2   zn       173 non-null    float64
 3   indus    173 non-null    float64
 4   chas     173 non-null    int64
 5   nox      173 non-null    float64
```

```
 6   rm       173 non-null    float64
 7   age      173 non-null    float64
 8   dis      173 non-null    float64
 9   rad      173 non-null    int64
 10  tax      173 non-null    int64
 11  ptratio  173 non-null    float64
 12  black    173 non-null    float64
 13  lstat    173 non-null    float64
dtypes: float64(10), int64(4)
memory usage: 19.1 KB
```

[10]:

	ID	crim	zn	indus	chas	nox
count	173.000000	173.000000	173.000000	173.000000	173.000000	173.000000
mean	258.404624	4.100862	12.661850	10.835145	0.086705	0.549981
std	143.289788	10.607761	24.536277	6.596488	0.282219	0.117826
min	3.000000	0.013810	0.000000	0.460000	0.000000	0.392000
25%	136.000000	0.082210	0.000000	5.320000	0.000000	0.447000
50%	268.000000	0.251990	0.000000	8.560000	0.000000	0.538000
75%	381.000000	3.673670	20.000000	18.100000	0.000000	0.624000
max	505.000000	88.976200	95.000000	27.740000	1.000000	0.871000

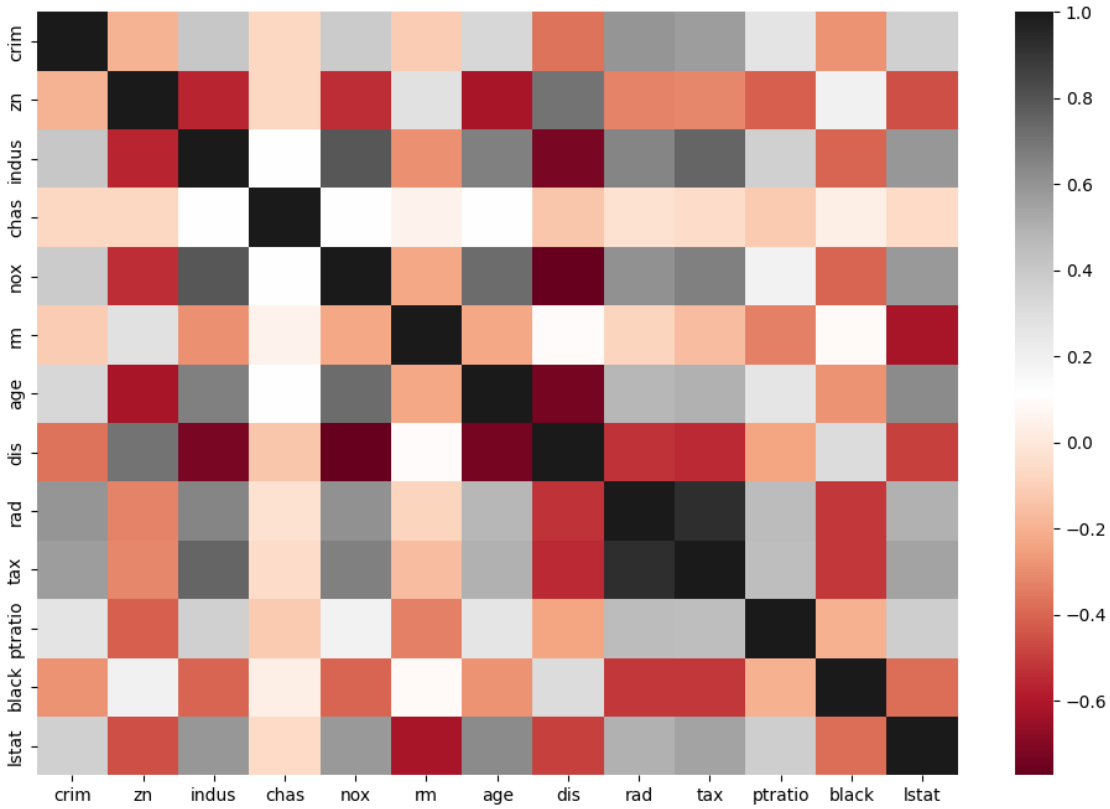
	rm	age	dis	rad	tax	ptratio
count	173.000000	173.000000	173.000000	173.000000	173.000000	173.000000
mean	6.321237	69.245665	3.958865	9.387283	406.231214	18.469942
std	0.700621	28.248244	2.324131	8.662621	164.480626	2.196196
min	4.138000	2.900000	1.178100	1.000000	187.000000	12.600000
25%	5.895000	42.800000	2.010700	4.000000	279.000000	17.000000
50%	6.223000	79.200000	3.421100	5.000000	330.000000	19.100000
75%	6.674000	94.600000	5.400700	24.000000	666.000000	20.200000
max	8.780000	100.000000	12.126500	24.000000	711.000000	22.000000

	black	lstat
count	173.000000	173.000000
mean	351.299711	12.917977
std	99.781464	7.293408
min	0.320000	1.920000
25%	371.720000	6.870000
50%	390.070000	12.120000
75%	396.060000	17.210000
max	396.900000	34.370000

[11]:
```
BostonTrain.drop('ID', axis = 1, inplace=True)
```

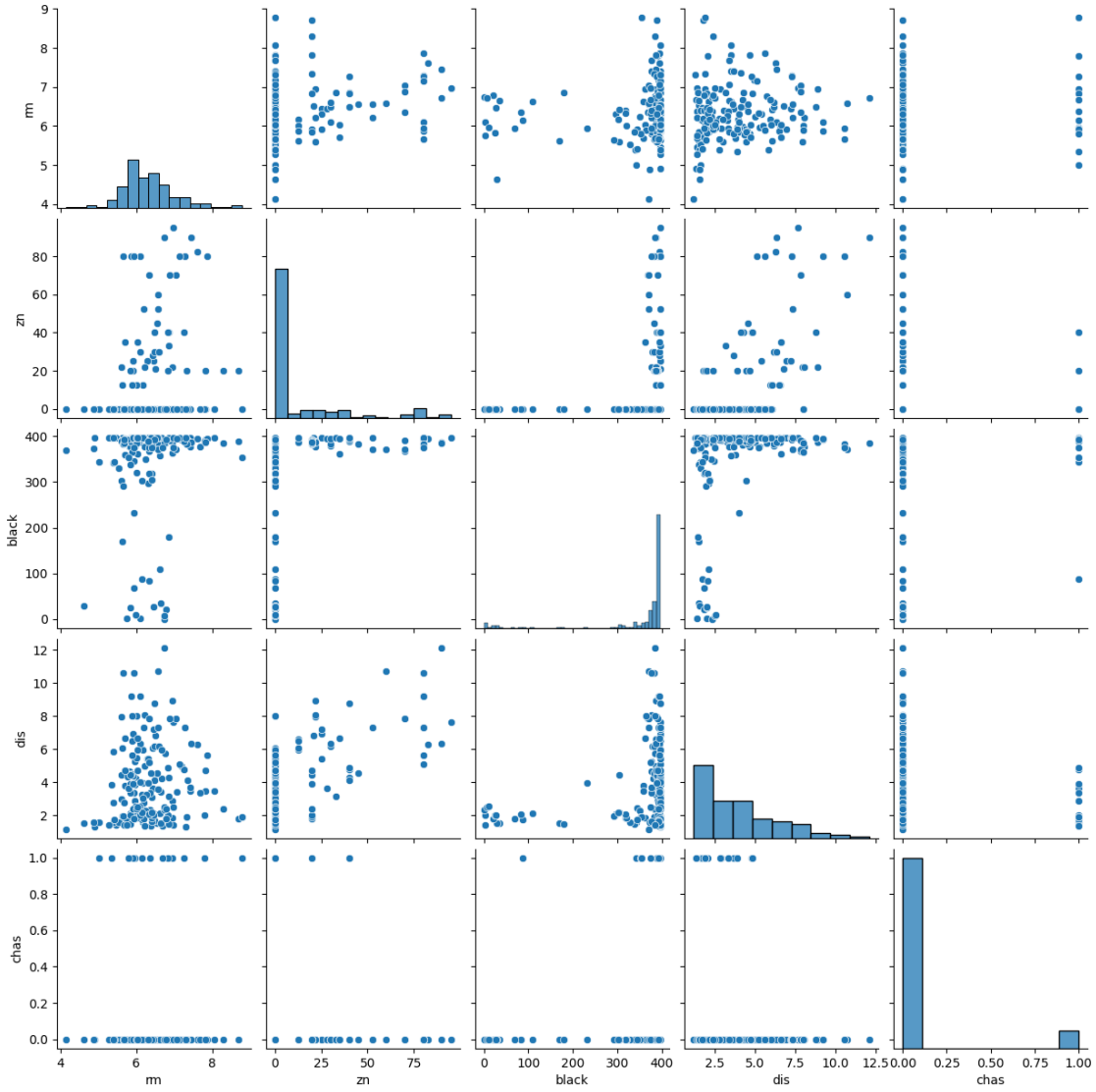
[13]:
```
plt.subplots(figsize=(12,8))
sns.heatmap(BostonTrain.corr(), cmap = 'RdGy')
```

[13]: <Axes: >



[15]:
```
sns.pairplot(BostonTrain, vars = ['rm', 'zn', 'black', 'dis', 'chas'])
```

[15]: <seaborn.axisgrid.PairGrid at 0x7bab0a107150>



[16]:
```
X = BostonTrain[['crim', 'zn', 'indus', 'chas', 'nox', 'rm', 'age', 'dis',␣
↪'rad', 'tax',
       'ptratio', 'black', 'lstat']]
y = BostonTrain['rad']
```

[17]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

[18]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4)
```

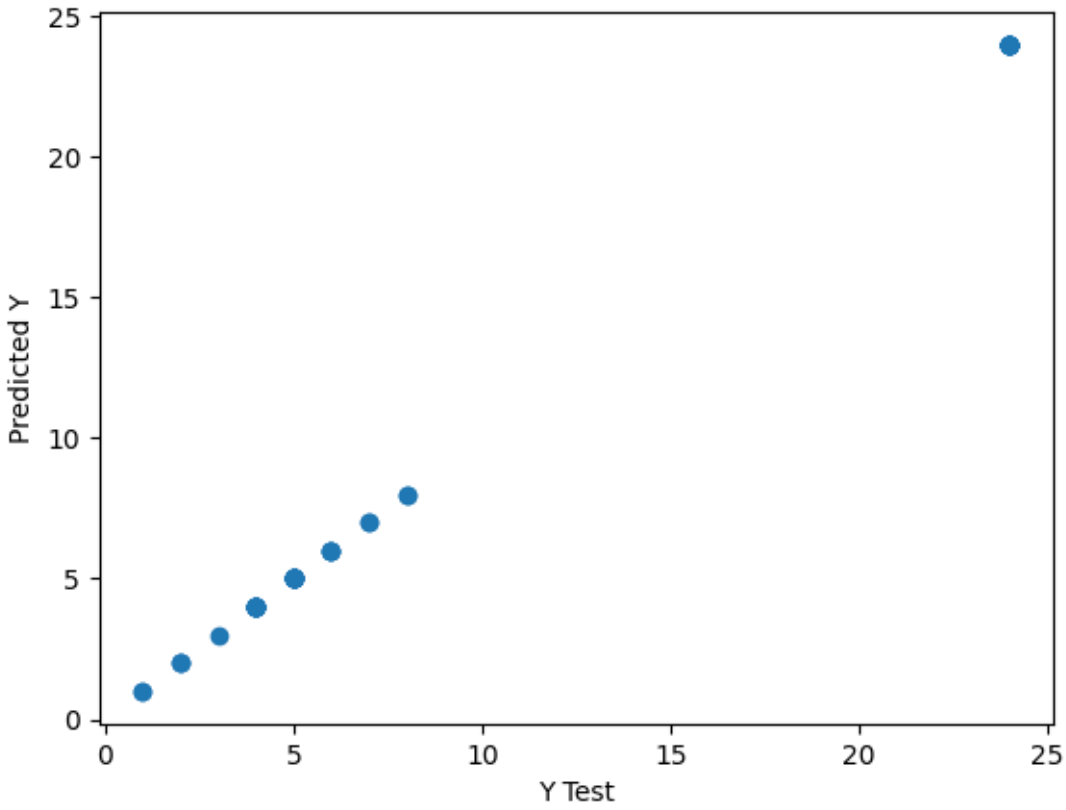
[19]:
```
lm = LinearRegression()
lm.fit(X_train,y_train)
```

[19]:
```
LinearRegression()
```

[20]:
```
predictions = lm.predict(X_test)
```

[21]:
```
plt.scatter(y_test,predictions)
plt.xlabel('Y Test')
plt.ylabel('Predicted Y')
```

[21]:
```
Text(0, 0.5, 'Predicted Y')
```



[22]:
```
from sklearn import metrics

print('MAE:', metrics.mean_absolute_error(y_test, predictions))
print('MSE:', metrics.mean_squared_error(y_test, predictions))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))
```

```
MAE: 1.0705722023171153e-14
MSE: 1.5758693959949422e-28
RMSE: 1.2553636767001310e-14
```

[23]:
```
sns.distplot((y_test-predictions),bins=50);
```

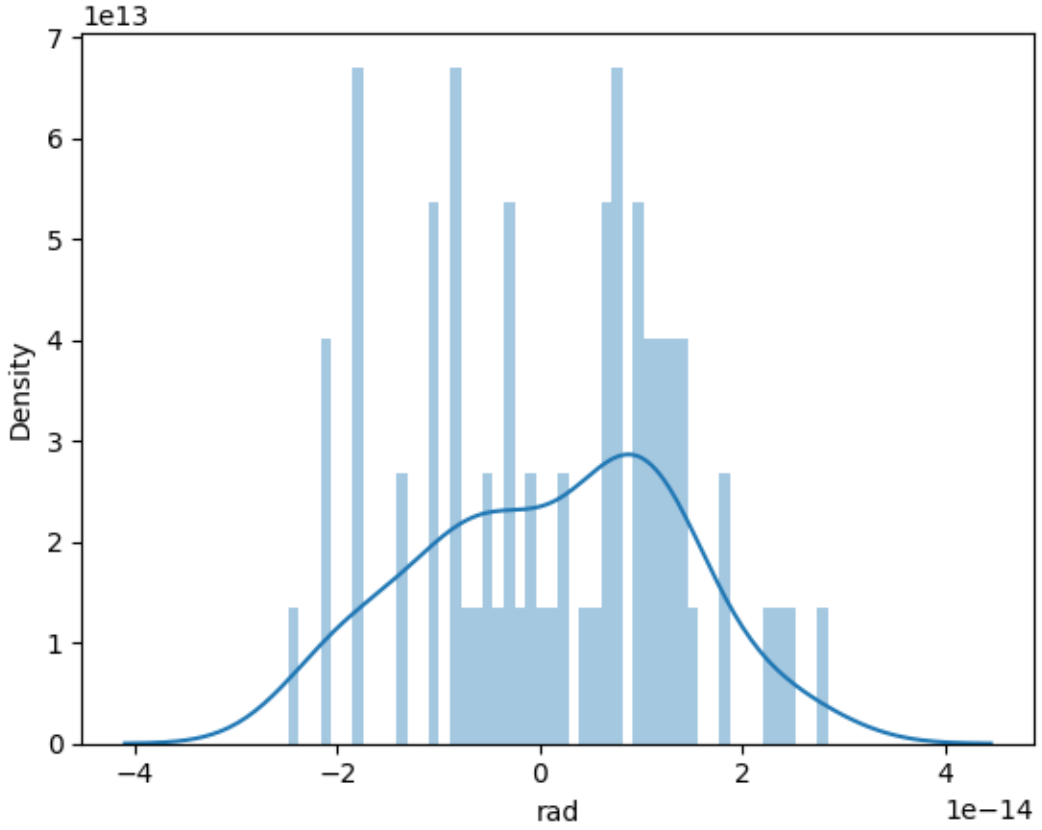
```
<ipython-input-23-5f2bc21c0ef7>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot((y_test-predictions),bins=50);
```



[24]:
```
coefficients = pd.DataFrame(lm.coef_,X.columns)
coefficients.columns = ['coefficients']
coefficients
```

[24]:

	coefficients
crim	-8.635243e-17
zn	1.804112e-16
indus	-1.567756e-16

```
chas     1.141646e-14
nox      2.328294e-15
rm      -4.165776e-16
age     -6.071532e-18
dis     -4.302656e-16
rad      1.000000e+00
tax      1.249001e-16
ptratio  2.592869e-16
black    1.144917e-16
lstat    9.454243e-17
```

[ ]: