```python
import pandas as pd

# Load the uploaded CSV file
df = pd.read_csv('BostonHousing.csv')

# Display the first few rows
df.head()
```

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	3
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	3
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	3
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	3
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	3

Next steps: Generate code with df | View recommended plots | New interactive sh

RAM
Disk

BostonHousing.csv

1 to 10 of 506 entries Filter

crim	zn	indus	chas	nox	rm	age	dis
0.00632	18	2.31	0	0.538	6.575	65.2	4.09
0.02731	0	7.07	0	0.469	6.421	78.9	4.9671
0.02729	0	7.07	0	0.469	7.185	61.1	4.9671
0.03237	0	2.18	0	0.458	6.998	45.8	6.0622
0.06905	0	2.18	0	0.458	7.147	54.2	6.0622
0.02985	0	2.18	0	0.458	6.43	58.7	6.0622
0.08829	12.5	7.87	0	0.524	6.012	66.6	5.5605
0.14455	12.5	7.87	0	0.524	6.172	96.1	5.9505
0.21124	12.5	7.87	0	0.524	5.631	100	6.0821
0.17004	12.5	7.87	0	0.524	6.004	85.9	6.5921

Show 10 per page 1 2 10 50 51

```
df.info()
df.describe()
df.isnull().sum()
```

```
<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
```

```
0
crim    0
```

	0
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

```
dtype: int64
```

```
[5]  X = df.drop('medv', axis=1)
     y = df['medv']
```

```
[6]  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=42)
```

```
[7]  from sklearn.linear_model import LinearRegression

     model = LinearRegression()
     model.fit(X_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

[6]

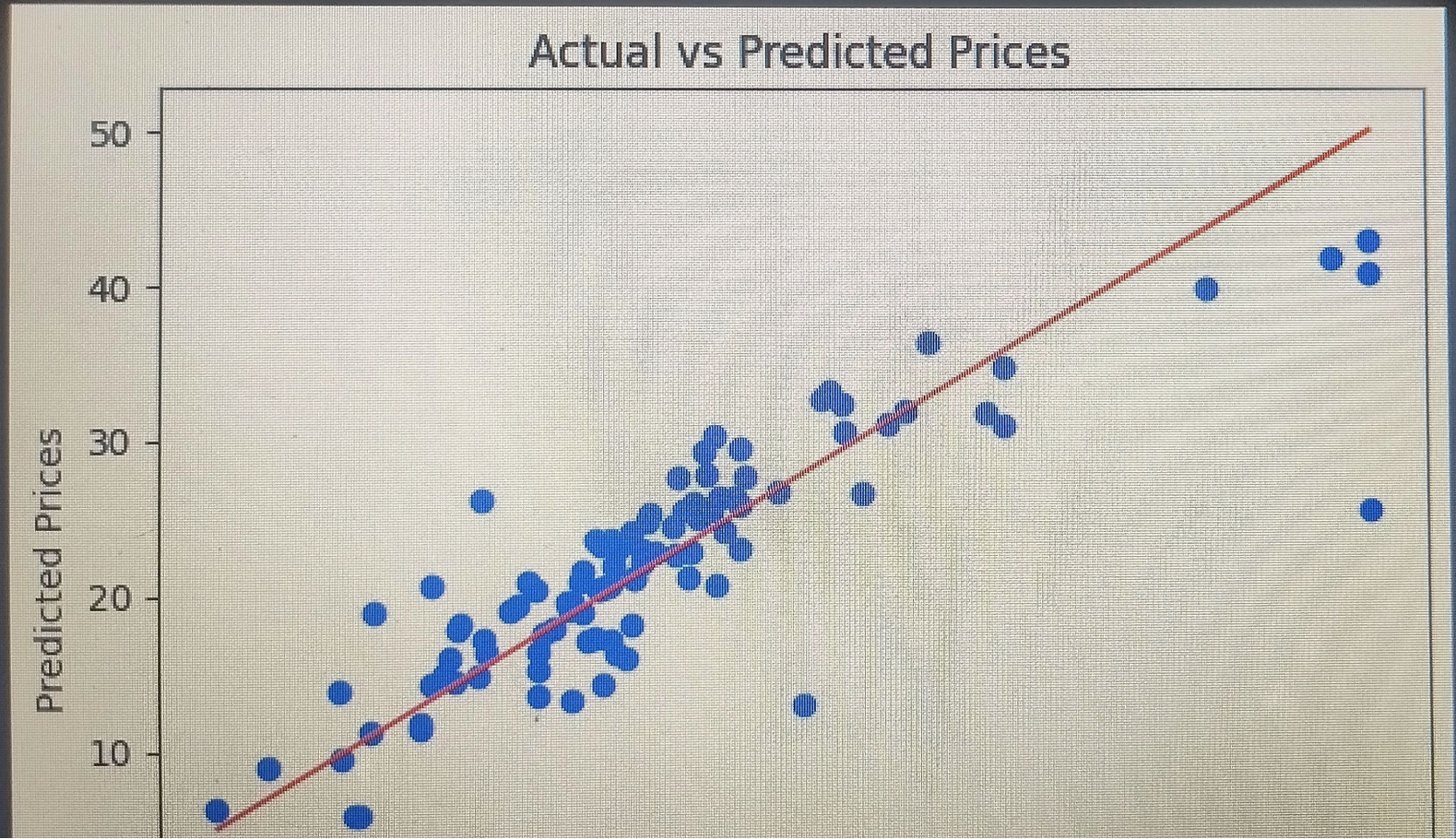
[7]
```
from sklearn.linear_model import LinearRegression

model = LinearRegression()
model.fit(X_train, y_train)
```

```
LinearRegression
LinearRegression()
```

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

y_pred = model.predict(X_test)

print("MAE:", mean_absolute_error(y_test, y_pred))
print("MSE:", mean_squared_error(y_test, y_pred))
print("R² Score:", r2_score(y_test, y_pred))
```

```
MAE: 3.189091965887837
MSE: 24.291119474973478
R² Score: 0.6687594935356326
```

```python
import matplotlib.pyplot as plt

plt.scatter(y_test, y_pred, color='blue')
plt.xlabel('Actual Prices')
plt.ylabel('Predicted Prices')
plt.title('Actual vs Predicted Prices')
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red')
plt.show()
```



Actual Prices

```
print("Intercept:", model.intercept_)
print("Coefficients:")
for feature, coef in zip(X.columns, model.coef_):
    print(f"{feature}: {coef}")
```

```
Intercept: 30.24675099392366
Coefficients:
crim: -0.11305592398537989
zn: 0.030110464145648292
indus: 0.040380720413330 43
chas: 2.7844382035079276
nox: -17.20263391781117
rm: 4.4388519951305
age: -0.006296362210980761
dis: -1.447865368530786
rad: 0.26242973558509075
tax: -0.010646786275308524
ptratio: -0.9154562404680 73
b: 0.012351334729969077
lstat: -0.5085714244487918
```