```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

```
dataframe=pd.read_csv("/content/Linear Regression.csv")
```

```
dataframe.head(10)
```

	X	Y
0	1	3.888889
1	2	4.555556
2	3	5.222222
3	4	5.888889
4	5	6.555556
5	6	7.222222
6	7	7.888889
7	8	8.555556
8	9	9.222222
9	10	9.888889

Next steps: Generate code with `dataframe` | View recommended plots | New interactive sheet

```
dataframe.isnull().sum()
```

	0
X	0
Y	0

dtype: int64

```
dataframe.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300 entries, 0 to 299
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   X       300 non-null    int64  
 1   Y       300 non-null    float64
dtypes: float64(1), int64(1)
memory usage: 4.8 KB
```

```
dataframe.describe()
```

	X	Y
count	300.000000	300.000000
mean	150.500000	102.215556
std	86.746758	57.842711
min	1.000000	1.888889
25%	75.750000	52.388889
50%	150.500000	102.222222
75%	225.250000	152.055556
max	300.000000	201.888889

```
x1=dataframe['X']
y1=dataframe['Y']
```

```
dataframe_sample=dataframe.sample(n=200,random_state=45)
```

```
x1 = dataframe['X'].values.reshape(-1, 1)
y1 = dataframe['Y'].values.reshape(-1, 1)
```

```
x1
```

```
       [ 77],
       [ 78],
       [ 79],
       [ 80],
       [ 81],
       [ 82],
       [ 83],
       [ 84],
       [ 85],
       [ 86],
       [ 87],
       [ 88],
       [ 89],
       [ 90],
       [ 91],
       [ 92],
       [ 93],
       [ 94],
       [ 95],
       [ 96],
       [ 97],
       [ 98],
       [ 99],
       [100],
       [101],
       [102],
       [103],
       [104],
       [105],
       [106],
       [107],
       [108],
       [109],
       [110],
       [111],
       [112],
       [113],
       [114],
       [115],
       [116],
```

```
x1_train, x1_test, y1_train, y1_test = train_test_split(x1, y1, test_size=0.25, random_st
```

```
scale=StandardScaler()
```

```
model=LinearRegression()
model.fit(x1_train,y1_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
dataframe["X"].shape
dataframe["Y"].shape
```

```
(300,)
```

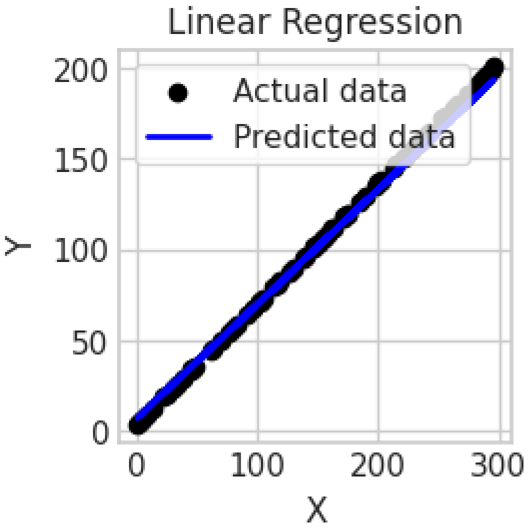
```
y1_pred = model.predict(x1_test)


mse=mean_squared_error(y1_test,y1_pred)


print(f'Mean Squared Error: {mse}')


# Visualize the results
plt.scatter(x1_test, y1_test, color='black', label='Actual data')
plt.plot(x1_test, y1_pred, color='blue', linewidth=2, label='Predicted data')
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Linear Regression')
plt.legend()
plt.show()
```

```
Mean Squared Error: 12.533970650684061
```



Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.