Linear Regression using a Random Dataset

Here, we generate a random dataset that simulates stock prices over time, build a linear regression model, and visualize the fitted line.

1. Importing necessary libraries

First, we need to import the libraries required for data manipulation, modeling, and visualization.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
```

2. Generating a random dataset

Next, we create a random dataset with 10 data points representing stock prices over a period of time.

```
np.random.seed(42)
time = np.arange(1,11)    # for time from 1 to 10
price = np.random.normal(loc=100, scale=10, size=10)    # for random prices around 100

data = pd.DataFrame({'Time': time, 'Price': price})
data    # to display the generated dataset
```

	Time	Price
0	1	104.967142
1	2	98.617357
2	3	106.476885
3	4	115.230299
4	5	97.658466
5	6	97.658630
6	7	115.792128
7	8	107.674347
8	9	95.305256
9	10	105.425600

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

3. Building a linear regression model

Now, we will build a linear regression model using the generated dataset.

```
X = data['Time'].values.reshape(-1,1)  # features
y = data['Price'].values  # target variable

# creating the linear regression model
model = LinearRegression()
model.fit(X, y)

# getting the slope (coefficient) and intercept
slope = model.coef_[0]
intercept = model.intercept_

slope, intercept
```

```
(-0.06900371683881476, 104.86013155960106)
```

4. Making predictions

With the model built, we can now make predictions based on our input data.

```
predicted_prices = model.predict(X)
data['Predicted Price'] = predicted_prices
```

```
data    # to display the updated dataset
```

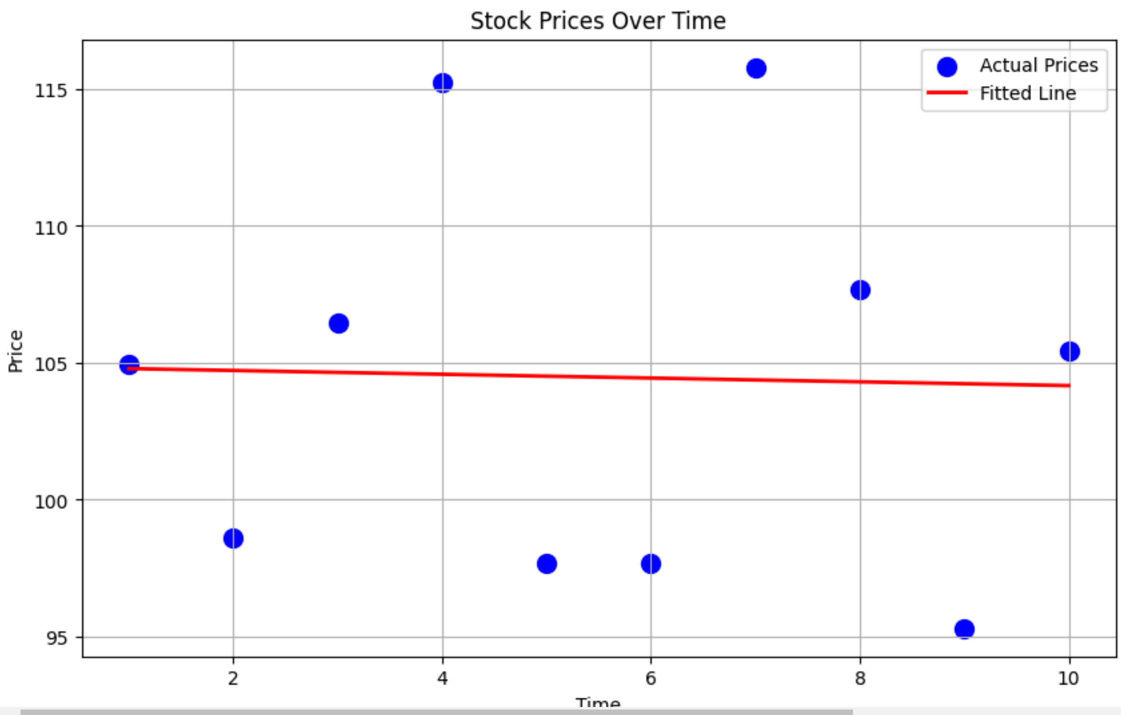
	Time	Price	Predicted Price
0	1	104.967142	104.791128
1	2	98.617357	104.722124
2	3	106.476885	104.653120
3	4	115.230299	104.584117
4	5	97.658466	104.515113
5	6	97.658630	104.446109
6	7	115.792128	104.377106
7	8	107.674347	104.308102
8	9	95.305256	104.239098
9	10	105.425600	104.170094

Next steps: Generate code with data   View recommended plots   New interactive sheet

5. Visualizing the results

Finally, we will visualize the original prices and the fitted line of the linear regression model.

```
plt.figure(figsize=(10,6))
plt.scatter(data['Time'], data['Price'], color='blue', label='Actual Prices', s=100)
plt.plot(data['Time'], data['Predicted Price'], color='red', label='Fitted Line', linewidth=2)
plt.title('Stock Prices Over Time')
plt.xlabel('Time')
plt.ylabel('Price')
plt.legend()
plt.grid()
plt.show()
```


Stock Prices Over Time
Actual Prices
Fitted Line
Price
Time