

Lab 2 - Linear Regression

July 21, 2022

```
[16]: import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
```

```
[17]: # Generate 'random' data
np.random.seed(0)
X = 2.5 * np.random.randn(100) + 1.5    # Array of 100 values with mean = 1.5,
    ↳ stddev = 2.5
res = 0.5 * np.random.randn(100)        # Generate 100 residual terms
y = 2 + 0.3 * X + res                    # Actual values of Y
```

```
[18]: # Create pandas dataframe to store our X and y values
df = pd.DataFrame(
    {'X': X,
     'y': y}
)

# Show the first five rows of our dataframe
df.head()
```

```
[18]:
```

	X	y
0	5.910131	4.714615
1	2.500393	2.076238
2	3.946845	2.548811
3	7.102233	4.615368
4	6.168895	3.264107

```
[19]: # Calculate the mean of X and y
xmean = np.mean(X)
ymean = np.mean(y)
```

```
[20]: # Calculate the terms needed for the numerator and denominator of beta
df['xycov'] = (df['X'] - xmean) * (df['y'] - ymean)
df['xvar'] = (df['X'] - xmean)**2
```

```
[21]: # Calculate beta and alpha
beta = df['xycov'].sum() / df['xvar'].sum()
```

```
alpha = ymean - (beta * xmean)
print(f'alpha = {alpha}')
print(f'beta = {beta}')
```

```
alpha = 2.0031670124623426
beta = 0.3229396867092763
```

```
[22]: ypred = alpha + beta * X
```

```
[23]: # Plot regression against actual data
plt.figure(figsize=(12, 6))
plt.plot(X, ypred)      # regression line
plt.plot(X, y, 'ro')    # scatter plot showing actual data
plt.title('Actual vs Predicted')
plt.xlabel('X')
plt.ylabel('y')

plt.show()
```

