10_regression_confidence_regions

May 7, 2025

[ ]:
```python
import os
user = os.getenv('USER')
os.chdir(f'/scratch/cd82/{user}/notebooks')
```

1 Linear Regression - Confidence region of regression line

Confidence intervals on estimates For a choosen significance level, with degrees of freedom $= n - 2$

$$CSS_x = \sum_{i=1}^{n} x^2 - \frac{(\sum_{i=1}^{n} x)^2}{n}$$

$$\Delta\beta_1 = \pm t_{(\alpha/2, n-2)} \sqrt{\frac{s^2}{CSS_x}}$$

$$\Delta\beta_0 = \pm t_{(\alpha/2, n-2)} \sqrt{\frac{s^2 \cdot \sum_{i=1}^{n} x^2}{n \cdot CSS_x}}$$

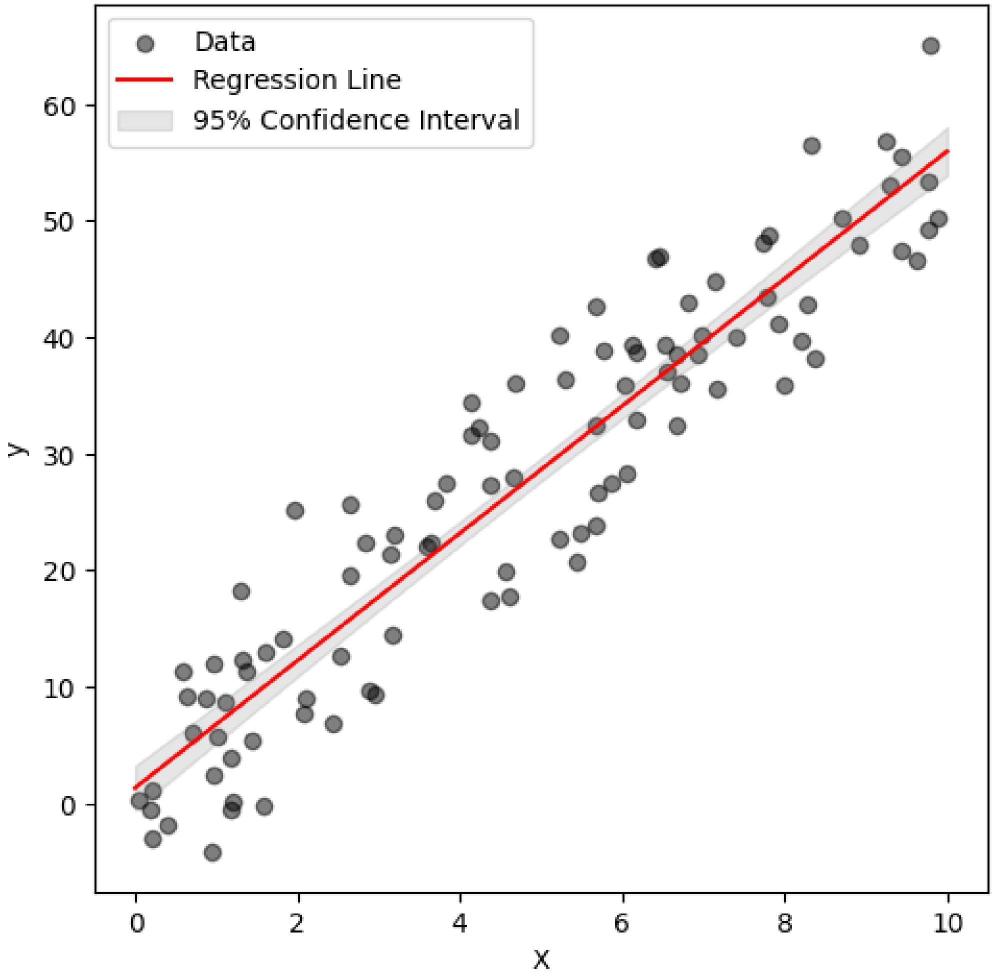
- CSS_x is the corrected sum of squares
- $\Delta\beta_1$ is the error range of the slope
- $\Delta\beta_0$ is the error range of the intercept
- s^2 is the variance

[1]:
```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
from scipy import stats
```

[8]:
```python
# Sample data
N=100
np.random.seed(0)
X = np.random.rand(N, 1) * 10
y = 5.5 * X.squeeze() + np.random.randn(N) * 6.0

# Fit the linear regression model
model = LinearRegression()
```

```
model.fit(X, y)

# Predictions
X_pred = np.linspace(0, 10, 100).reshape(-1, 1)
y_pred = model.predict(X_pred)

# The two tail significance
significance = 0.95
sign_2tail = (1.0 - significance) / 2.0
sign_2tail = significance + sign_2tail
print('significance =', sign_2tail)

# Calculate the confidence intervals
n = N # len(X)
mse = np.mean((y - model.predict(X))**2)
print('mse.shape: ', mse.shape)
t_val = stats.t.ppf(significance, n - 2)  # 95% confidence interval in 2-tail␣
 ↪test
se = np.sqrt(mse * (1/n + (X_pred - np.mean(X))**2 / np.sum((X - np.
 ↪mean(X))**2)))
ci = t_val * se

# Plotting
plt.figure(figsize=(6, 6))
plt.scatter(X, y, color='black', alpha=0.5, label='Data')
plt.plot(X_pred, y_pred, color='red', label='Regression Line')
plt.fill_between(X_pred.squeeze(), (y_pred - ci.T).T.squeeze(), (y_pred + ci.T).
 ↪T.squeeze(),
                 color='gray', alpha=0.2, label='95% Confidence Interval')
plt.xlabel('X')
plt.ylabel('y')
plt.legend()
plt.show()
```

```
significance = 0.975
mse.shape:  ()
```


Data
Regression Line
95% Confidence Interval
y
X
0
10
20
30
40
50
60
0
2
4
6
8
10

[ ]: