Assessing accuracy of the LSR line

Chris Cornwell

Feb 13, 2025

Outline

Assuming the data does have a linear relationship

Outline

Assuming the data does have a linear relationship

Underlying assumption

- Modeled points in a plane as being from a line, but with noise in the y-coordinate direction. In other words, we assumed an underlying relationship

$$y = mx + b + \varepsilon$$

for some m and b, and a random variable ε[1] that has expected value 0. Alternatively, among the "entire population" there is an LSR line $mx + b$.

[1] ε is called the error term.

Underlying assumption

- Modeled points in a plane as being from a line, but with noise in the y-coordinate direction. In other words, we assumed an underlying relationship
$$y = mx + b + \varepsilon$$
for some m and b, and a random variable ε[1] that has expected value 0. Alternatively, among the "entire population" there is an LSR line $mx + b$.
- Assumption: ε is independent of x.

When we have a data set $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, from the population, our procedure determines an LSR line $\hat{m}x + \hat{b}$. However, $\hat{m}$ and $\hat{b}$ are not the slope and intercept for the population curve m and b.

[1] ε is called the error term.

Example

Simulate noisy linear data: make 30 points, using a standard deviation $\sigma = 0.5$. We'll use slope -1.6 and intercept 0.8.

Example

Simulate noisy linear data: make 30 points, using a standard deviation $\sigma = 0.5$. We'll use slope -1.6 and intercept 0.8.

```
x = np.random.uniform(0, 2, size=30)

def simulate_data(x, std):
    return -1.6*x + 0.8 + np.random.normal(0, std, size=len(x))
y = simulate_data(x, 0.5)
```

Example

Simulate noisy linear data: make 30 points, using a standard deviation $\sigma = 0.5$. We'll use slope -1.6 and intercept 0.8.

```
x = np.random.uniform(0, 2, size=30)

def simulate_data(x, std):
    return -1.6*x + 0.8 + np.random.normal(0, std, size=len(x))
y = simulate_data(x, 0.5)
```

In groups, compute slope and intercept of the LSR line for a size 30 simulated data set; store $\hat{m}$ and $\hat{b}$ (in two lists). Iterate this 1000 times $\longrightarrow$ a list of 1000 slopes and intercepts.

Example

Simulate noisy linear data: make 30 points, using a standard deviation $\sigma = 0.5$. We'll use slope -1.6 and intercept 0.8.

```
x = np.random.uniform(0, 2, size=30)

def simulate_data(x, std):
    return -1.6*x + 0.8 + np.random.normal(0, std, size=len(x))
y = simulate_data(x, 0.5)
```

In groups, compute slope and intercept of the LSR line for a size 30 simulated data set; store $\hat{m}$ and $\hat{b}$ (in two lists). Iterate this 1000 times $\longrightarrow$ a list of 1000 slopes and intercepts.

What is the mean of the slopes and of the intercepts?

Sample statistic, relation to population statistic

This fundamental to statistics.

- Say that a sample of 2000 people are selected from around the country and their height is measured. Mean of these 2000 heights: sample mean.

Sample statistic, relation to population statistic

This fundamental to statistics.

- Say that a sample of 2000 people are selected from around the country and their height is measured. Mean of these 2000 heights: sample mean.
- Sample mean differs from the true mean height of the entire population of the country. (Perhaps, not by much.)
  - Weak Law of Large Numbers: if s random samples of 2000 people taken, and each sample mean calculated, as $s \longrightarrow \infty$, mean of the sample means limits to population mean.

Sample statistic, relation to population statistic

This fundamental to statistics.

- Say that a sample of 2000 people are selected from around the country and their height is measured. Mean of these 2000 heights: sample mean.
- Sample mean differs from the true mean height of the entire population of the country. (Perhaps, not by much.)
  - Weak Law of Large Numbers: if s random samples of 2000 people taken, and each sample mean calculated, as $s \longrightarrow \infty$, mean of the sample means limits to population mean.
- Analogous thing happens with data from linear relationship with noise – think of parameters $\hat{m}$ and $\hat{b}$ as sample statistics (like sample mean).

Confidence intervals

How close do we suspect $\hat{m}$ and $\hat{b}$ to be to the “true” (population) slope and intercept?

Confidence intervals

How close do we suspect $\hat{m}$ and $\hat{b}$ to be to the "true" (population) slope and intercept?

Standard error (SE): Suppose that for our error term ε, we have $\mathrm{Var}(\varepsilon) = \sigma^2$. Sample size: n.

Confidence intervals

How close do we suspect $\hat{m}$ and $\hat{b}$ to be to the “true” (population) slope and intercept?

Standard error (SE): Suppose that for our error term ε, we have $\mathrm{Var}(\varepsilon) = \sigma^2$. Sample size: n.

Using $\bar{x}$ for the average of $x_1, \ldots, x_n$,

$$SE(\hat{m})^2 = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2};$$

$$SE(\hat{b})^2 = \sigma^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \right).$$

Confidence intervals

How close do we suspect $\hat{m}$ and $\hat{b}$ to be to the "true" (population) slope and intercept?

Standard error (SE): Suppose that for our error term ε, we have $\mathrm{Var}(\varepsilon) = \sigma^2$. Sample size: n.

Using $\bar{x}$ for the average of $x_1, \ldots, x_n$,

$$SE(\hat{m})^2 = \frac{\sigma^2}{\sum_{i=1}^n (x_i - \bar{x})^2};$$

$$SE(\hat{b})^2 = \sigma^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \right).$$

Roughly, these are the amount, on average, that $\hat{m}$ (resp. $\hat{b}$) differs from true slope m (resp. true intercept b).

Confidence intervals

How close do we suspect $\hat{m}$ and $\hat{b}$ to be to the "true" (population) slope and intercept?

Standard error (SE): Suppose that for our error term ε, we have $\mathrm{Var}(\varepsilon) = \sigma^2$. Sample size: n.

Using $\bar{x}$ for the average of $x_1, \ldots, x_n$,

$$SE(\hat{m})^2 = \frac{\sigma^2}{\sum_{i=1}^n (x_i - \bar{x})^2};$$

$$SE(\hat{b})^2 = \sigma^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \right).$$

Roughly, these are the amount, on average, that $\hat{m}$ (resp. $\hat{b}$) differs from true slope m (resp. true intercept b).

σ is unknown, but can estimate it with **residual standard error**:

$$\hat{\sigma}^2 = RSE^2 = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n-2}.$$

Confidence intervals

How close do we suspect $\hat{m}$ and $\hat{b}$ to be to the "true" (population) slope and intercept?

Formulae:

$$SE(\hat{m})^2 = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2};$$

$$SE(\hat{b})^2 = \sigma^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \right).$$

Estimate:

$$\sigma^2 \approx RSE^2 = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-2}.$$

[2] 95% of the time, these intervals contain m, b.

Confidence intervals

How close do we suspect $\hat{m}$ and $\hat{b}$ to be to the "true" (population) slope and intercept?

Formulae:

$$SE(\hat{m})^2 = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2};$$

$$SE(\hat{b})^2 = \sigma^2\left(\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right).$$

Estimate:

$$\sigma^2 \approx RSE^2 = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-2}.$$

Can get (roughly) 95% confidence interval[2] with $\pm 2SE$:

$$(\hat{m} - 2SE(\hat{m}), \hat{m} + 2SE(\hat{m}))$$

and

$$(\hat{b} - 2SE(\hat{b}), \hat{b} + 2SE(\hat{b})).$$

[2] 95% of the time, these intervals contain m, b.