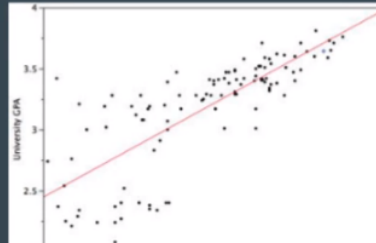


Simple Linear Regression

- So let's re-state our problem in more general terms
- We are given a set of points: $\{ (x_1, y_1), (x_2, y_2), \dots, (x_N, y_N) \}$
- We plot them in a 2-D chart
- We find the line of best fit
- Is there a more systematic way of doing it, other than drawing it using paper and a ruler? Of course!



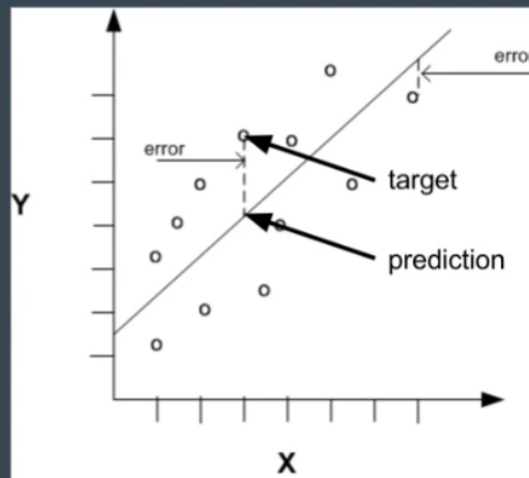
Simple Linear Regression

- Our line of best fit is defined as:

$$\hat{y}_i = ax_i + b$$

- How can we make sure this “fits” the data well? We would like:

$$y_i \text{ close to } \hat{y}_i, i = 1..N$$



Simple Linear Regression

- What we want:
- For any target != prediction, a +ve contribution to error
- Standard way is to square the difference
- Called the “sum of squared errors”

$$E = \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

- Note that we use the square of the error to counteract the fact that any given error can be positive or negative!
- We want to find E such that E is the smallest possible value for the sum of squared errors. We are trying to minimize E

- Substitute with our expression for a line

$$E = \sum_{i=1}^N (y_i - (ax_i + b))^2$$

- Remember! y_i and x_i are given (it's data we collected during our experiment)
 - What we want to find is a and b
- We want to minimize E with respect to a and b , we can use partial derivatives

Finding derivative of E with respect to a

$$E = \sum_{i=1}^N (y_i - (ax_i + b))^2$$

$$= \sum_{i=1}^N (y_i - ax_i - b)^2$$

$$\frac{\partial E}{\partial a} = \sum_{i=1}^N 2(y_i - ax_i - b)(-x_i)$$

Set to 0, to minimize E with respect to a

$$\sum_{i=1}^N 2(y_i - ax_i - b)(-x_i) = 0$$

$$0 = -2 \sum_{i=1}^N y_i x_i + a \sum_{i=1}^N x_i^2 + b \sum_{i=1}^N x_i$$

$$\sum_{i=1}^N y_i x_i = a \sum_{i=1}^N x_i^2 + b \sum_{i=1}^N x_i$$

Finding derivative of E with respect to b

$$\frac{\partial E}{\partial b} = \sum_{i=1}^N 2(y_i - ax_i - b)(-1)$$

Set to 0, to minimize E with respect to b

$$\sum_{i=1}^N 2(y_i - ax_i - b)(-1) = 0$$

$$0 = -\sum_{i=1}^N y_i + a \sum_{i=1}^N x_i + b \sum_{i=1}^N 1$$

$$\sum_{i=1}^N y_i = a \sum_{i=1}^N x_i + bN$$

We now have 2 equations and 2 unknowns, we can solve for a and b

Lets replace the summations with letters to make the algebra easier

$$\sum_{i=1}^N y_i x_i = a \sum_{i=1}^N x_i^2 + b \sum_{i=1}^N x_i \longrightarrow E = aC + bD$$

$$\sum_{i=1}^N y_i = a \sum_{i=1}^N x_i + bN \longrightarrow F = aD + bN$$

Lets use elimination

$$\begin{aligned} (E = aC + bD) \cdot D \\ (F = aD + bN) \cdot C \end{aligned} \longrightarrow \begin{aligned} ED &= aCD + bD^2 \\ -FC &= aCD + bNC \\ \hline (ED - FC) &= b(D^2 - NC) \end{aligned}$$

$$b = \frac{ED - FC}{D^2 - NC}$$

$$\begin{aligned} (E = aC + bD) \cdot N \\ (F = aD + bN) \cdot D \end{aligned} \longrightarrow \begin{aligned} EN &= aCN + bDN \\ -FD &= aDN + bDN \\ \hline (EN - FD) &= a(CN - D^2) \end{aligned}$$

$$a = \frac{EN - FD}{CN - D^2}$$

Lets change the denominator of b so both equations have the same denominator

$$b = \frac{ED - FC}{D^2 - NC} \left(\frac{-1}{-1} \right) = \frac{FC - ED}{NC - D^2}$$

Now lets substitute back in our summations and simplify the results

$$b = \frac{FC - ED}{NC - D^2} = \frac{\left(\sum_{i=1}^N y_i \cdot \sum_{i=1}^N x_i^2 \right) - \left(\sum_{i=1}^N x_i y_i \cdot \sum_{i=1}^N x_i \right)}{\left(N \sum_{i=1}^N x_i^2 \right) - \left(\sum_{i=1}^N x_i \right)^2}$$

$$a = \frac{EN - FD}{NC - D^2} = \frac{N \sum_{i=1}^N x_i y_i - \left(\sum_{i=1}^N y_i \cdot \sum_{i=1}^N x_i \right)}{\left(N \sum_{i=1}^N x_i^2 \right) - \left(\sum_{i=1}^N x_i \right)^2}$$

recall that the sample mean $\Rightarrow \bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$ and $\bar{xy} = \frac{1}{N} \sum_{i=1}^N x_i y_i$

$$a = \frac{N \sum_{i=1}^N x_i y_i - \left(\sum_{i=1}^N y_i \cdot \sum_{i=1}^N x_i \right)}{\left(N \sum_{i=1}^N x_i^2 \right) - \left(\sum_{i=1}^N x_i \right)^2} \cdot \frac{1/N^2}{1/N^2} = \frac{\frac{1}{N} \sum_{i=1}^N x_i y_i - \frac{1}{N^2} \left(\sum_{i=1}^N y_i \cdot \sum_{i=1}^N x_i \right)}{\frac{1}{N} \sum_{i=1}^N x_i^2 - \frac{1}{N^2} \left(\sum_{i=1}^N x_i \right)^2}$$

$$a = \frac{\bar{xy} - \bar{x}\bar{y}}{\bar{x}^2 - \bar{x}^2}$$

Note that $\bar{x} \cdot \bar{y} = \frac{1}{N^2} \sum_{i=1}^N x_i \sum_{i=1}^N y_i$

$$\bar{x}^2 = \bar{x} \cdot \bar{x} = \frac{1}{N^2} \left(\sum_{i=1}^N x_i \right)^2$$

$$b = \frac{\left(\sum_{i=1}^N y_i \cdot \sum_{i=1}^N x_i^2 \right) - \left(\sum_{i=1}^N x_i y_i \cdot \sum_{i=1}^N x_i \right)}{\left(N \sum_{i=1}^N x_i^2 \right) - \left(\sum_{i=1}^N x_i \right)^2} \cdot \frac{1/N^2}{1/N^2} = \frac{\frac{1}{N^2} \left(\sum_{i=1}^N y_i \cdot \sum_{i=1}^N x_i^2 \right) - \frac{1}{N^2} \left(\sum_{i=1}^N x_i y_i \cdot \sum_{i=1}^N x_i \right)}{\frac{1}{N} \sum_{i=1}^N x_i^2 - \frac{1}{N^2} \left(\sum_{i=1}^N x_i \right)^2}$$

$$b = \frac{\bar{y}\bar{x}^2 - \bar{xy}\bar{x}}{\bar{x}^2 - \bar{x}^2}$$

Programming linear regression using numpy

- In order to make the computation of a and b easier we can actually use.

- We can actually re-write the equation we just derived to something simpler that will make it easier to calculations with Numpy

Handwritten derivation of the linear regression formula:

$$a = \frac{N \sum_{i=1}^N y_i x_i - \sum_{i=1}^N x_i \sum_{i=1}^N y_i}{N \sum_{i=1}^N x_i^2 - \left(\sum_{i=1}^N x_i \right)^2} \quad b = \frac{\sum_{i=1}^N y_i \sum_{i=1}^N x_i^2 - \sum_{i=1}^N x_i \sum_{i=1}^N y_i x_i}{N \sum_{i=1}^N x_i^2 - \left(\sum_{i=1}^N x_i \right)^2}$$

recall the definition of a dot product

$$a = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad b = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad a \cdot b = x_1 y_1 + x_2 y_2 + \dots + x_n y_n$$

$$a \cdot b = \sum_{i=1}^N x_i y_i \quad a \cdot a = \sum_{i=1}^N x_i^2 = x_1 x_1 + x_2 x_2 + \dots$$

Let multiply our equation by $\frac{1}{N} = \frac{1}{N}$

$$a = \frac{N \sum_{i=1}^N y_i x_i - \sum_{i=1}^N x_i \sum_{i=1}^N y_i}{N \sum_{i=1}^N x_i^2 - \left(\sum_{i=1}^N x_i \right)^2} \cdot \frac{1}{N} = \frac{\sum_{i=1}^N y_i x_i - \frac{1}{N} \left(\sum_{i=1}^N x_i \right) \left(\sum_{i=1}^N y_i \right)}{\sum_{i=1}^N x_i^2 - \frac{1}{N} \left(\sum_{i=1}^N x_i \right) \left(\sum_{i=1}^N x_i \right)}$$

$$= \frac{x \cdot y - \bar{x} \sum_{i=1}^N y_i}{x \cdot x - \bar{x} \sum_{i=1}^N x_i}$$

$$b = \frac{\sum_{i=1}^N y_i \sum_{i=1}^N x_i^2 - \sum_{i=1}^N x_i \sum_{i=1}^N y_i x_i}{N \sum_{i=1}^N x_i^2 - \left(\sum_{i=1}^N x_i \right)^2} \cdot \frac{1}{N} = \frac{\frac{1}{N} \left(\sum_{i=1}^N y_i \right) \left(\sum_{i=1}^N x_i^2 \right) - \frac{1}{N} \left(\sum_{i=1}^N x_i \right) \left(\sum_{i=1}^N y_i x_i \right)}{x \cdot x - \bar{x} \sum_{i=1}^N x_i}$$

$$= \frac{(\bar{y} x \cdot x) - (\bar{x}) (x \cdot y)}{x \cdot x - \bar{x} \sum_{i=1}^N x_i}$$

Implementation using numpy

```
denominator = X.dot(X) - X.mean() * X.sum()
a = (X.dot(Y) - (X.mean()*Y.sum())) / denominator
b = ((Y.mean()*X.dot(X)) - (X.mean()*X.dot(Y))) / denominator
```

Determining how good the model is

- We can use the R^2 formula to determine how good the model is, the formula is
 - $R^2 = 1 - \frac{SS_{residual}}{SS_{total}}$
 - $SS_{residual} = \sum_{i=1}^N (y_i - \hat{y}_i)^2$
 - $SS_{total} = \sum_{i=1}^N (y_i - \bar{y})^2$
- Notice that $SS_{residual}$ is just our error formula
- SS_{total} is the difference between each y against the mean of y.

Cases

- If our error formula is close to 0 then our formula will be $R^2 = 1 - 0 = 1$ which means our model is pretty much perfect, the closer the value is to 1 the better
- If $R^2 = 0$ that means to formula was $R^2 = 1 - 1 = 0$ which means that our $SS_{residual}$ was very close to SS_{total} meaning our predictions were just taking the mean of y to make predictions, which is not good.
 - This can happen if your data does not have a clear trend
- If $R^2 < 0$ that means that $\frac{SS_{residual}}{SS_{total}} > 1$ and this points out that your model is performing worse than just predicting the mean of y