STAT 107 – Spring 2021

Simple Linear Regression

Ha Khanh Nguyen (hknguyen)

What is Regression?

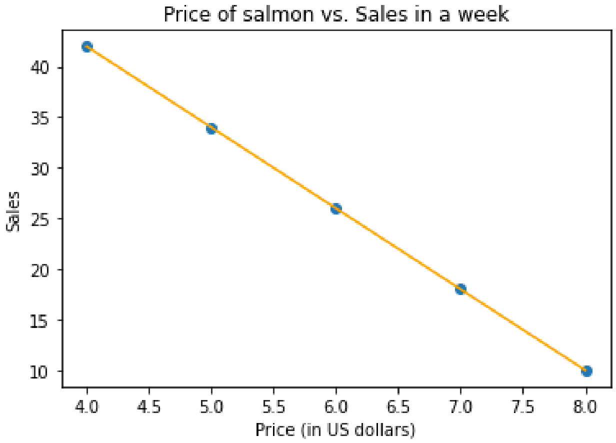
- Regression is a type of statistical analysis that goes beyond simply determining whether there is some effect, or even how strong the effect is. It is the process of using one (or more) variables to model the relationship of another variable.
- **Statistical Modeling**: Using data to define the general relationship of multiple variables in the form of an equation.
- Example: How can we predict the car price based on the car's miles per gallon (mpg)? Predict the death rate of COVID-19 in an area based on the infection rate?

What is Simple Linear Regression?

- Simple Linear Regression (SLR) is restricted to comparing no more than two continuous/discrete variables to determine if there is a linear relationship between them.
- Two variables are denoted as X (Predictor variable) and Y (Response variable) and are plotted against each other on a scatterplot.
  - Sometimes, it doesn't matter much which is which, but in experiments, the predictor (X) variable is what we think *might* be the causal agent.
- The **regression line** is the model we use for our prediction.

Example: Consider the following study in which a store owner is interested in exploring the relationship between the price of salmon and the number of sales at that price for that week.

He conducted the study over 5 weeks with the price being held constant at a different price each week.



Price (X)	Sales (Y)
$4.00	42
$5.00	34
$6.00	26
$7.00	18
$8.00	10

What do we see?

- As the price increases, the number of sales decreases.
- Specifically, as the price goes up by $1, the number of weekly sales drop by exactly ________.

Regression line: $y = mx + y_0$

- m is the **slope** which tells you the rate at which the response variable changes with respect to a unit change in the predictor.
- y_0 is the **y-intercept** which provides you the model's prediction for the response value when the predictor value is at 0.

How do we compute m and y_0?

Slope: $m = r \cdot \frac{s_Y}{s_X}$ y-intercept: $y_0 = \bar{y} - m \cdot \bar{x}$

Example: Compute the regression line for the above example?

How do we interpret this equation/line?

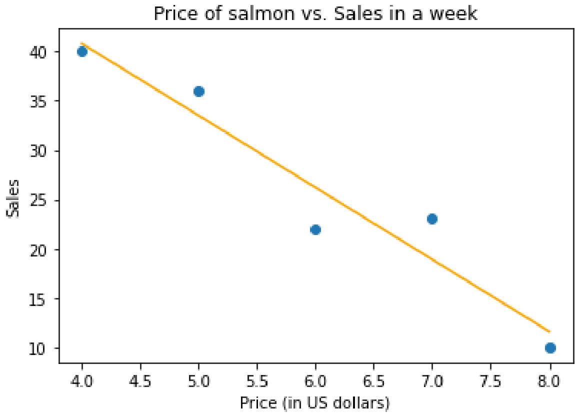
For every _______ increase in ________, we expect _________ to increase/decrease by _____ .

If the ______ is 0, then we expect _______ to be ________.

If the price of salmon were $9.00, then according to this model, we'd expect the # of weekly sales to be?

If the price of salmon were $4.50, then according to this model, we'd expect the # of weekly sales to be?

Example 2:
The above example is unrealistic as it is rarely ever the case in practice that all of our data points fall on a single line. Consider the following data:



Price (X)	Sales (Y)
$4.00	40
$5.00	36
$6.00	22
$7.00	23
$8.00	10

Recompute the regression line:

Residual: the vertical distance from a data point to the regression line.

- It measures the **error** between our best fit line and an individual data point.
- We use the notation y and $\hat{y}$ to distinguish between an actual data point and the predicted value for the Y variable using the model.
- **Residual = $y - \hat{y}$ = actual – predicted**.

Linear Regression in Python

Step 1: Load/create the data

```
import pandas as pd
faithful = pd.read_csv('https://stat107.hknguyen.org/files/datasets/faithful.csv')
```

Step 2: Create the Linear Regression model

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
```

Step 3: Fit the model to our data (the waiting time (X) and the eruption time (Y) in this case)

```
model = model.fit(faithful[['waiting']], faithful['eruptions'])
```

Step 4: Get the slope and y-intercept

```
# slope
model.coef_
#y-intercept
model.intercept_
```

Step 5: Use the model to predict the value of the response variable for a given value of the predictor

```
model.predict([[60]])
```

Step 6: Compute the residual(s)

```
faithful.loc[faithful['waiting'] == 60, 'eruptions'] - model.predict([[60]])
```

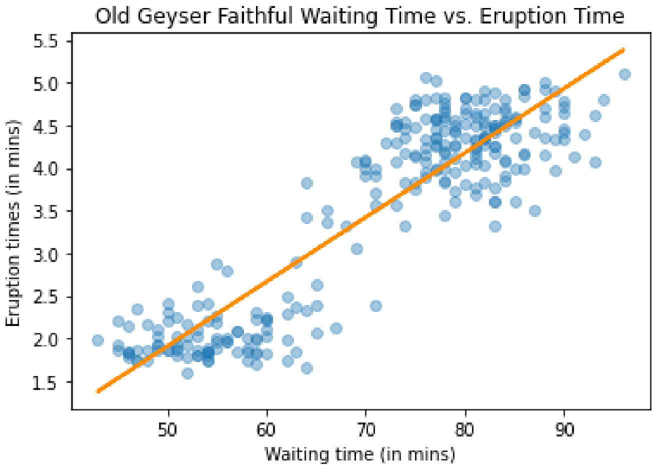
Plotting: Plot is often very useful in Regression. The following code was used to produce the plot below.

```
plt.scatter(faithful['waiting'],
faithful['eruptions'], alpha=0.4)

plt.xlabel('Waiting time (in mins)')
plt.ylabel('Eruption times (in mins)')
plt.title('Old Geyser Faithful Waiting Time vs.
Eruption Time')

m = model.coef_
b = model.intercept_
plt.plot(faithful['waiting'],
m*faithful['waiting']+b,
      color='darkorange', linewidth=2)

plt.show()
```


Old Geyser Faithful Waiting Time vs. Eruption Time
Eruption times (in mins)
Waiting time (in mins)