Explore ordinary least squares

As previously mentioned, one way for finding the best fit line in regression modeling is to try different models until you find the best one. But for simple linear regression, the formulas for the best beta coefficients have been derived. In this reading, you will go through an example to gain a better understanding of how the sum of squared residuals can change as β^0 and β^1 change. There will be resources for further exploration if you're interested in deriving the formulas for estimating the coefficients using ordinary least squares. In this reading, we will cover:

- Formula and notation review
- Minimizing the sum of squared residuals (SSR)
- Estimating beta coefficients

Formula and notation review

Earlier, you learned about simple linear regression as a method for estimating the linear relationship between a continuous dependent variable and one independent variable. An estimate based on simple linear regression can be represented mathematically as $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 X$.

Remember that the hat symbol indicates that the beta coefficients are just estimates. As a result, the y-values derived from the regression model are also just estimates.

A common technique for calculating the coefficients of a linear regression model is called ordinary least squares, or OLS. Ordinary least squares estimates the beta coefficients in a linear regression model by minimizing a measure of error called the sum of squared residuals.

You can calculate the sum of squared residuals via this formula:

$$\sum_{i=1}^{N}(Observed - Predicted)^2$$

which can be rewritten using mathematical notation as:

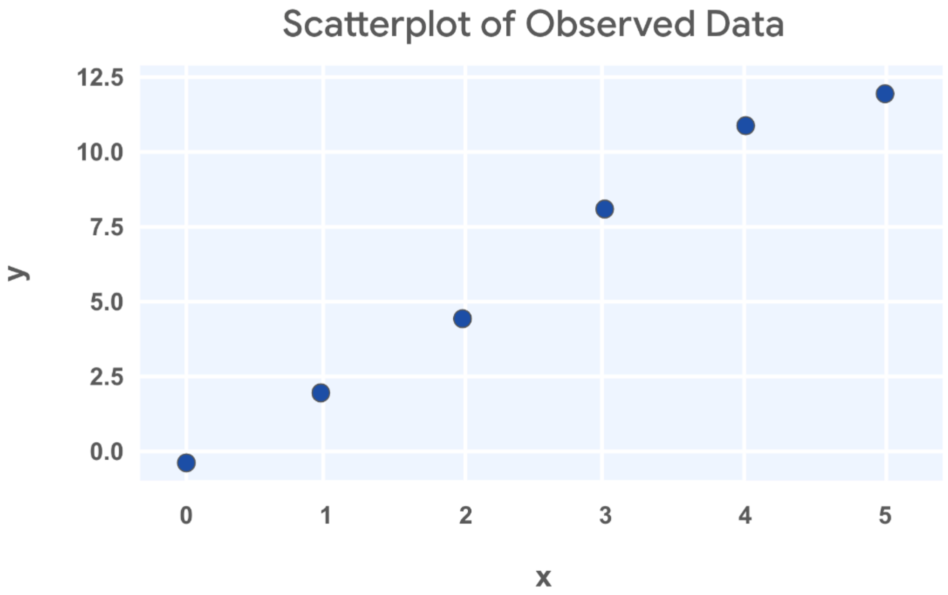
$$\sum_{i=1}^{N}(y_i - \hat{y}_i)^2.$$

The large E shaped symbol is the capital Greek letter, sigma, and it denotes a sum. So the sum of squared residuals is the sum of the squared differences between the observed values and the values predicted by the regression model.

Minimizing the sum of squared residuals (SSR)

For the purposes of this reading, assume that you have a dataset of 6 observations: (0, -1), (1, 2), (2, 4), (3, 8), (4, 11), and (5, 12). These can be plotted on a 2-dimensional X-Y coordinate plane.

X (observed)	**Y (observed)**
0	-1
1	2
2	4
3	8
4	11
5	12



Line 1: $\hat{y} = -0.5 + 3x$

Next, let's assume some values for $\hat{\beta}_0$ and $\hat{\beta}_1$ and calculate the sum of squared residuals. For the first attempt, let's assume $\hat{\beta}_0 = -0.5$ and $\hat{\beta}_1 = 3$. Then the linear equation would be $\hat{y} = -0.5 + 3x$. Since you now have the equation for y, you can calculate the predicted values by plugging in each value of x.

For example, if $x = 0$, then $\hat{y} = -0.5 + 3 * 0 = -0.5$. If $x = 1$, then $\hat{y} = -0.5 + 3 * 1 = 2.5$. So after calculating all the predicted values, then you can calculate the residual for each data point.

$\mathbf{X}$ (observed)	$\mathbf{Y}$ (actual)	$\mathbf{Y}$ (predicted) $= \mathbf{-0.5 + 3X}$	Residual
0	-1	-0.5	$-1 - (-0.5) = -1 + 0.5 = -0.5$
1	2	2.5	$2 - 2.5 = -0.5$
2	4	5.5	$4 - 5.5 = -1.5$
3	8	8.5	$8 - 8.5 = -0.5$
4	11	11.5	$11 - 11.5 = -0.5$
5	12	14.5	$12 - 14.5 = -2.5$

Then you can square each of the residuals by multiplying them by themselves, and then adding them all together to calculate the sum of squared residuals.

Residual	Squared Residual
$-1-(-0.5)=-1+0.5=-0.5$	0.25
$2-2.5=-0.5$	0.25
$4-5.5=-1.5$	2.25
$8-8.5=-0.5$	0.25
$11-11.5=-0.5$	0.25
$12-14.5=-2.5$	6.25

Sum of squared residuals $= 0.25+0.25+2.25+0.25+0.25+6.25=9.5$

Line 2: $\hat{y}=-0.5+2.5x$

Next, let's adjust just the slope from the prior example. So $\hat{\beta}_0=-0.5$ but $\hat{\beta}_1=2.5$. Then the linear equation would be $\hat{y}=-0.5+2.5x$. You can plug in values for x just like last time to calculate the predicted values and get the squared residuals.

$\mathbf{X}$ (observed)	$\mathbf{Y}$ (actual)	$\mathbf{Y}$ (predicted) $\mathbf{-10.5+2.5x}$	Residual	Squared Residuals
0	-1	-0.5	-0.5	0.25
1	2	2	0	0
2	4	4.5	-0.5	0.25
3	8	7	1	1
4	11	9.5	1.5	2.25
5	12	12	0	0

Sum of squared residuals $= 0.25+0+0.25+1+2.25+0=3.75$.

Great! This estimate is way better!

Estimating beta coefficients

You could keep adjusting the slope and intercept, and then calculating the predicted values, residuals, and squared residuals. But there's really no way to be sure you've found the best fit line. Through advanced math, some formulas have been derived to find the beta coefficients that minimize error.

There are multiple ways to write out the formulas for finding the beta coefficients. For simple linear regression, one way to write the formulas is as follows:

- $\hat{\beta}_1=\frac{\sum_{i=1}^{n}(X_i-\bar{X})(Y_i-\bar{Y})}{\sum_{i=1}^{n}(X_i-\bar{X})^2}$
- $\hat{\beta}_0=\bar{Y}-\hat{\beta}_1\bar{X}$

You won't be asked to calculate beta coefficients without help from a computer, but it can be interesting to explore if you desire. We've provided additional resources in case you're interested.

Key takeaways

Given a sample of data, you can try out different lines that could fit your data. You could calculate the sum of squared residuals for each line to determine which fits your data best. As a data professional, it's important to understand what the sum of squared residuals represents, and how to calculate it on your own. Thankfully, we have computers and programming languages that can calculate the sum of squared residuals and perform OLS for us. You can explore the deeper math behind OLS and SSR on your own if you wish!

Resources

- Parameter Estimation - Ordinary Least Squares Method: *Rudolph, A., Krois, J., Hartmann, K. (2023): Statistics and Geodata Analysis using Python (SOGA-Py). Department of Earth Sciences, Freie Universitaet Berlin.*

Correlation and the intuition behind simple linear regression

So far you've learned that simple linear regression is a technique that estimates the linear relationship between one independent variable, X, and one continuous dependent variable, Y. You've also learned about ordinary least squares estimation (OLS), which is a common way to determine the coefficients of the regression line—the line of "best fit" through the data. In this reading, you'll explore the meaning of correlation; learn about *r*, or the "correlation coefficient;" and discover how to determine the regression equation. This knowledge will help you better understand relationships between variables, and thus how linear regression works.

Correlation

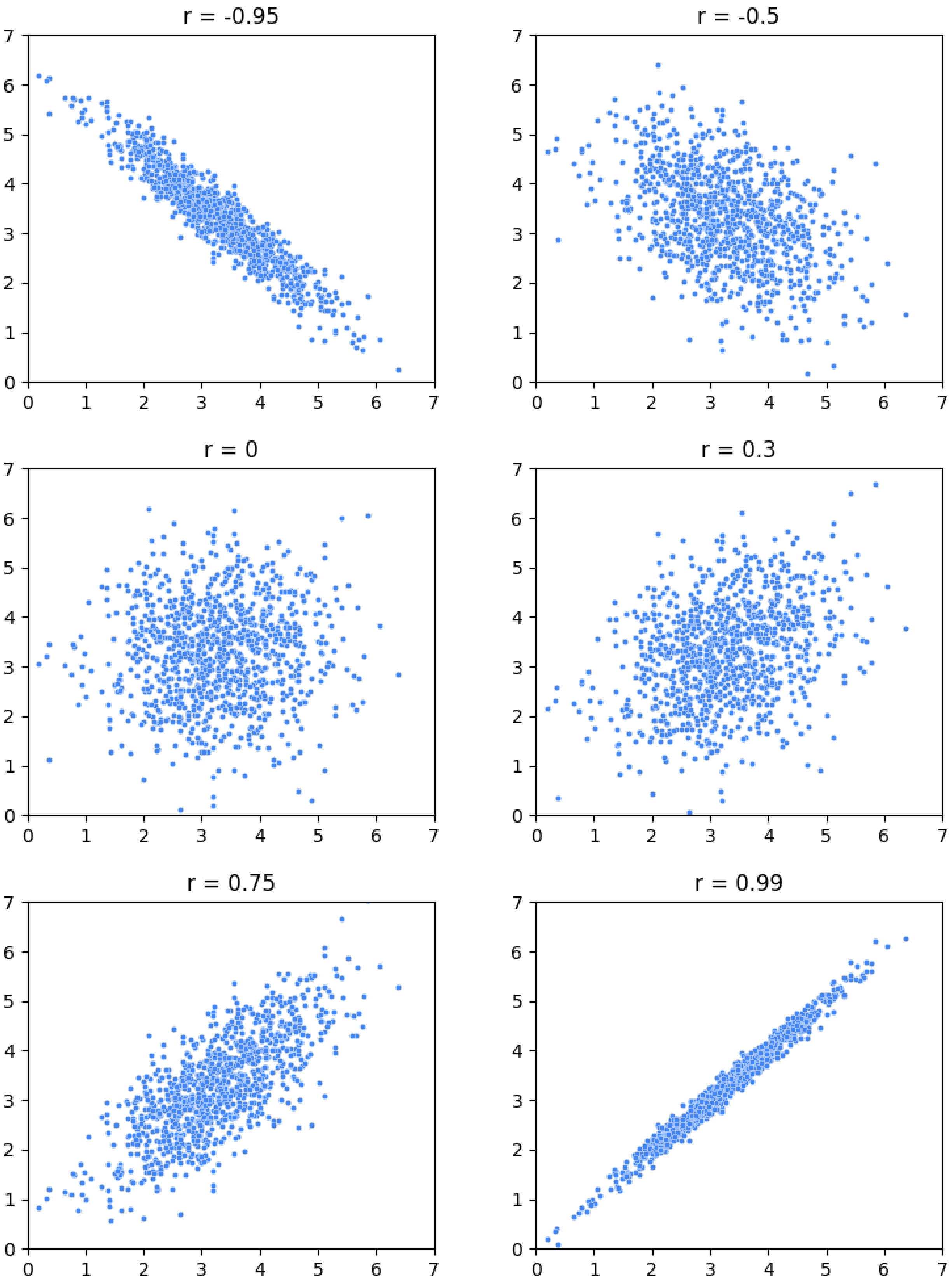
Correlation is a measurement of the way two variables move together. If there is a strong correlation between the variables, then knowing one will be very helpful to predict the other. However, if there is a weak correlation between two variables, then knowing the value of one will not tell you much about the value of the other. In the context of linear regression, correlation refers to *linear* correlation: as one variable changes, so does the other at a constant rate.

In the statistics course, you learned that a continuous variable can be summarized using some basic numbers. Two of these summary statistics are:

- **Average:** A measurement of central tendency (mean, median, or mode)
- **Standard deviation:** A measurement of spread

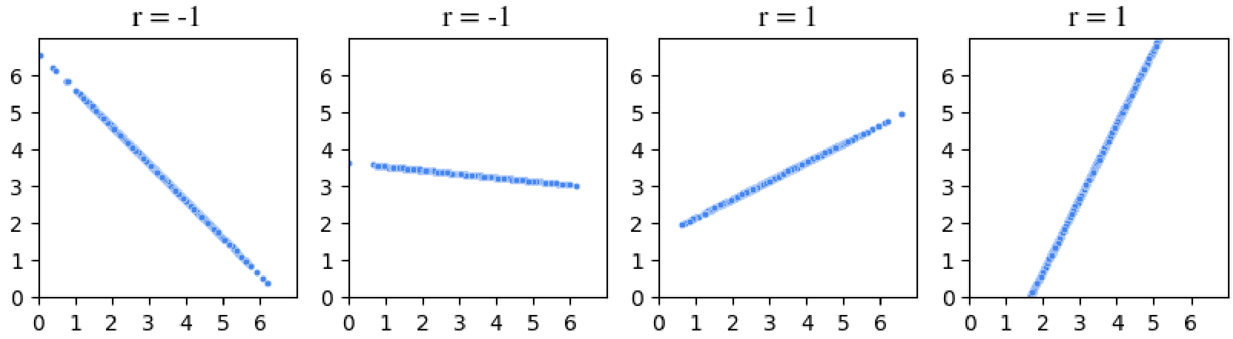
When two variables are summarized together, there is another relevant statistic called ***r***, **Pearson's correlation coefficient** (named after the person who helped develop it), or simply the linear **correlation coefficient.** The correlation coefficient quantifies the strength of the linear relationship between two variables. It always falls in the range of [-1, 1]. When *r* is negative, there is a negative correlation between the variables: as one increases, the other decreases. When *r* is positive, there is a positive correlation between the variables: as one increases, so too does the other. When *r* = 0, there is no *linear* correlation between the variables. Note that there are cases where one variable might be precisely determined by another—like $y=x^2$ or y=sin(x)—but the value of the *linear* correlation between X and Y would nonetheless be low or zero because their relationship is non-linear.

The following figure depicts scatterplots of bivariate (bi = "two", variate = "variables") data where each variable has the same mean and standard deviation and only the correlation coefficient varies.


r = -0.95
r = -0.5
r = 0
r = 0.3
r = 0.75
r = 0.99
0
1
2
3
4
5
6
7

Notice that the closer to -1 or 1 *r* is, the more linear the data appears. When *r* is exactly 1 or exactly -1, then the variables are perfectly correlated, and their graph is a line. When *r* is zero, there is no correlation between the variables, and, in this example, the data appears as a shapeless cloud of points.

However, *r* only tells you the strength of the linear correlation between the variables; it does not tell you anything about the magnitude of the slope of the relationship between the variables aside from its sign. For example, variables with *r*=1 wouldn't tell you if increasing X by one would lead to Y increasing by 10, 100, 0.1, or something else. It would only tell you that you can be sure that it *would* increase. This fact is illustrated in the following figure, where even though the slopes of the lines are all different, *r* is only either -1 or 1. If the line is perfectly horizontal or perfectly vertical, then *r* is undefined. (If you're wondering why, refer to the equation below. One of the terms in the denominator would equal zero, which would make the whole denominator equal zero, which would result in an undefined solution.)



Calculate *r*

The formula for *r* is:

$$r = \frac{covariance(X, Y)}{(SD\ X)(SD\ Y)}$$

where:

$$covariance(X, Y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n}$$

Note: The formulas for *r* and covariance given here represent those used for entire populations. For samples, the denominator of the covariance formula is *n - 1* and, similarly, the standard deviations in the formula for *r* are calculated using *n - 1* instead of *n.* For simplicity, this reading will use the population formulas in its demonstrations.

An easier way of thinking about this calculation is: the numerator—the covariance—represents the extent to which X and Y vary together from their respective means. When this value is positive, it suggests that high values of X tend to be associated with high values of Y, indicating a positive correlation. Conversely, if the value is negative, it suggests that high values of X tend to be associated with low values of Y and vice versa, indicating a negative correlation.

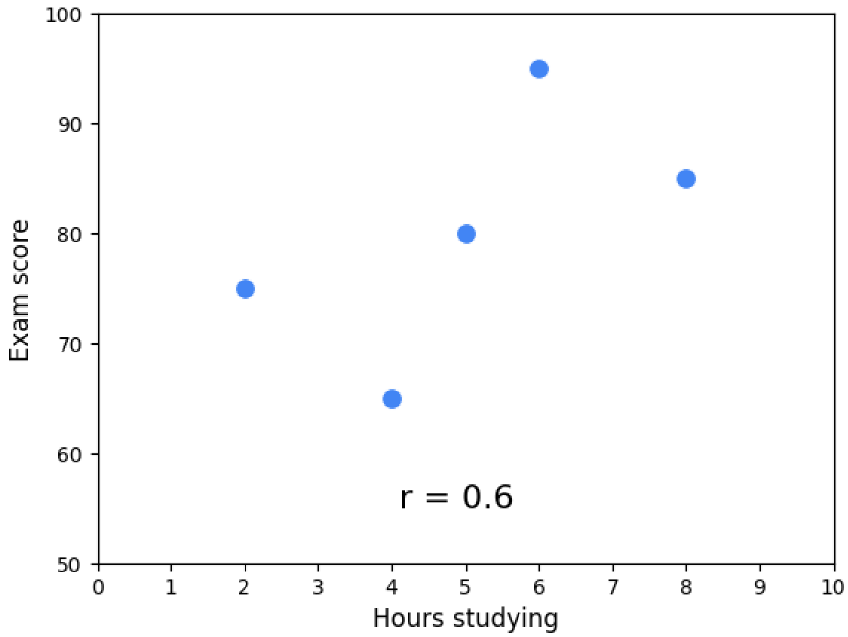
The denominator—the product of the standard deviations—standardizes the units of the numerator. It adjusts for the inherent variability of the individual variables. This makes *r* a statistic without a unit. It is a pure number, without dimension.

An equivalent way to calculate *r* is to convert each data point in each variable to standard units (subtract the mean, divide by the standard deviation), then take the average of the products.

Here's an example. Suppose five students took an exam and you recorded how many hours they spent studying and also their grade. The following table breaks out the calculation of *r.*

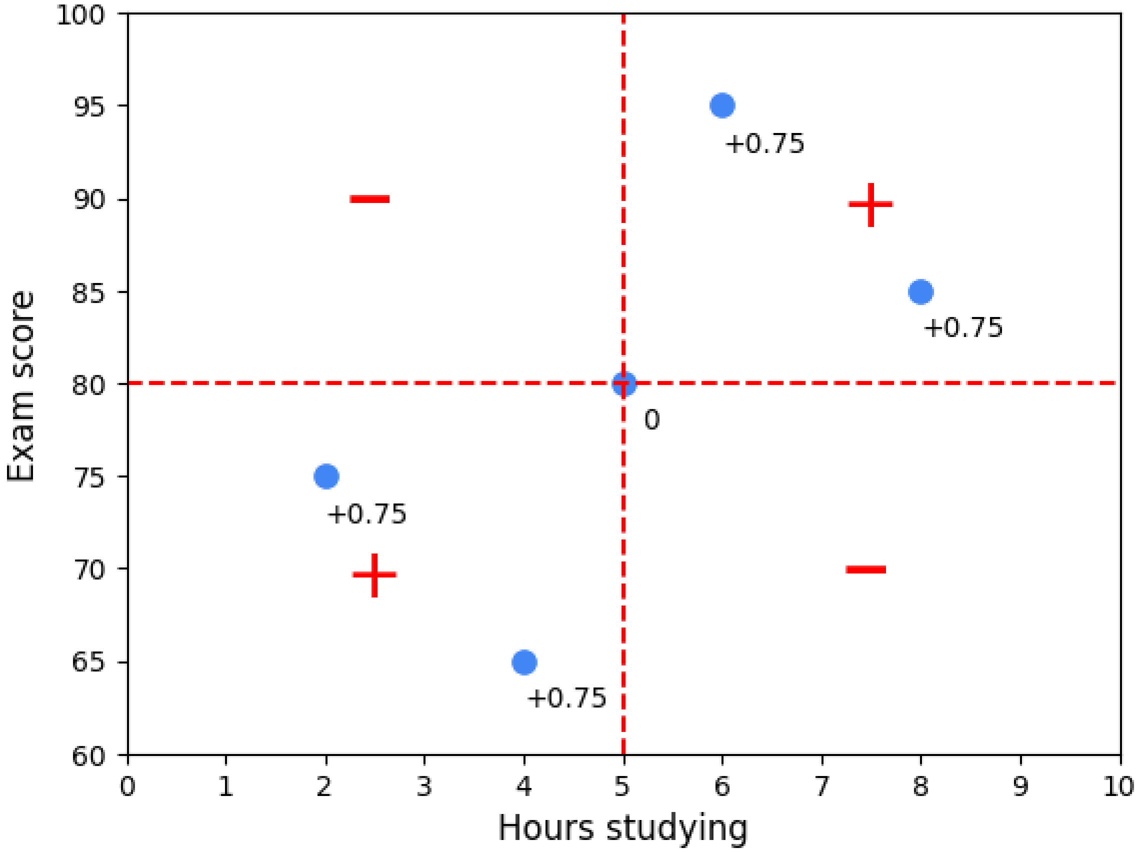
Hours studying (X)	**Exam grade (Y)**	**X in standard units**	**Y in standard units**	**Product of standard units**
2	75	-1.5	-0.5	0.75
4	65	-0.5	-1.5	0.75
5	80	0	0	0
6	95	0.5	1.5	0.75
8	85	1.5	0.5	0.75
mean X = 5 **SD X = 2**	**mean Y = 80** **SD Y = 10**			**mean of products (r) = 0.6**

The correlation coefficient is 0.6. Here is a graph of this data:



Notice that the cloud of points slopes upwards. This corresponds with *r* being positive. The correlation coefficient works as an indicator of association because it uses the product of each variable's deviation from its mean. When the product is positive, it means *both* the X and the Y values are either below their respective means (negative standard units) or above their respective means (positive standard units). They vary together. However, when this product is negative, it means one of the values is above its mean and the other is below it. They vary in opposing directions relative to their respective means.

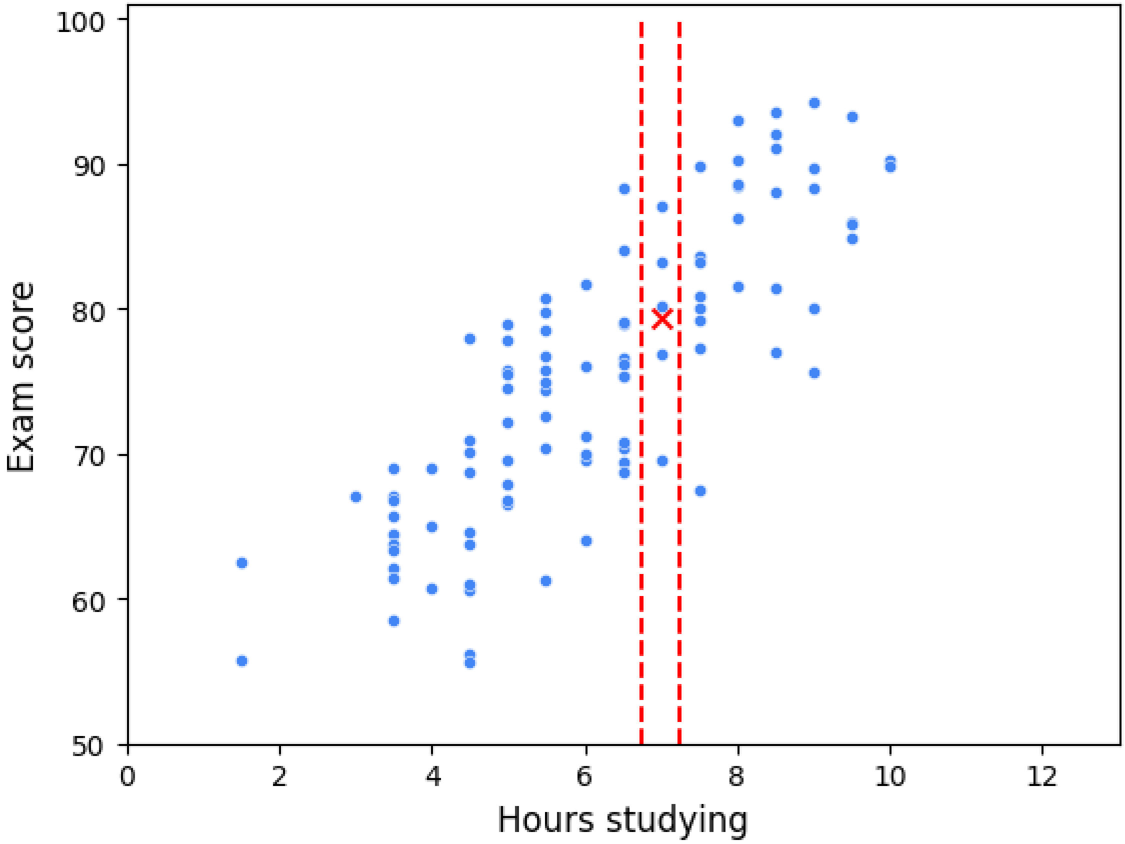
The following figure illustrates this idea. The figure is divided into quadrants. The vertical line represents the mean X value and the horizontal line represents the mean Y value. Each point is labeled with the product of its standardized scores (refer to the table above). The average of these scores is *r*. When *r* is positive, more points will tend to be in the positive quadrants, and vice versa.



Regression

In the absence of any other information, if you had to guess a randomly selected student's exam score, the best way for you to minimize your error would be to guess the average of all the students' scores. But what if you also knew how many hours that student studied? Now, your best guess might be the average score of only the students who studied for that many hours.

Here is an example using a sample of 100 students with study times rounded to the nearest half hour. Suppose you were told a student studied for seven hours. To guess their exam score, one way to minimize error is to guess the average of only the students who studied for seven hours.



In this scatterplot, all of the students who studied for seven hours fall between the two vertical lines. Their mean exam score is represented by an X. Linear regression expands on this concept. A regression line represents the estimated average value of Y for every value of X, given the assumptions and limitations of a linear model. In other words, the actual average Y values for each X might not lie exactly on the regression line if the relationship between X and Y is not perfectly linear or if there are other factors influencing Y that are not included in the model. The regression line attempts to balance out these influences to find a straight-line relationship that best fits the data as a whole. It's an estimation of the central tendency of Y, given X.

The regression equation

Now that you know about *r* and you better understand the concept of regression, you're ready to put everything together to find the line of best fit through the data. The formula for this line is known as the regression equation. There are two keys to this step.

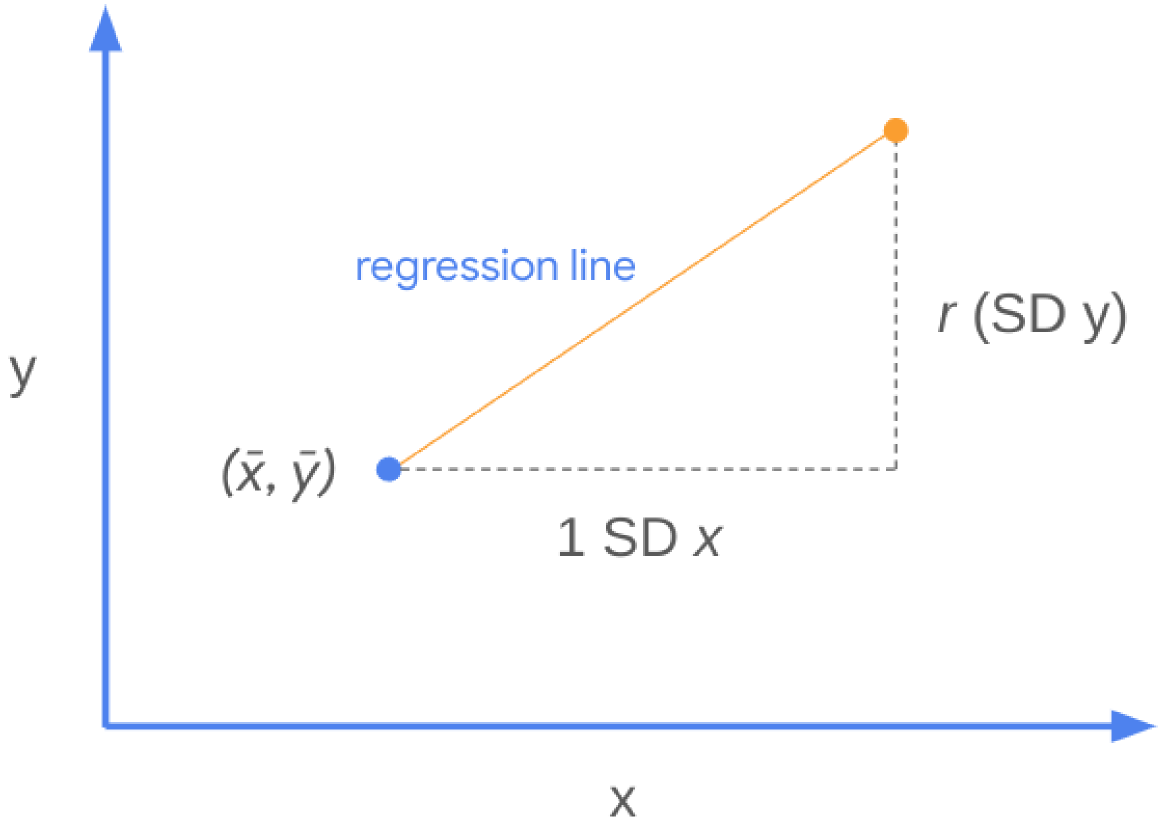
The first is:

- The mean value of X and the mean value of Y (i.e., point ($\bar{x}$, $\bar{y}$)) will always fall on the regression line.

The second is to understand what *r* means:

- For each increase of one standard deviation in X, there is an expected increase of *r* standard deviations in Y, on average over X.

The following figure illustrates how these concepts work together to determine the regression line.



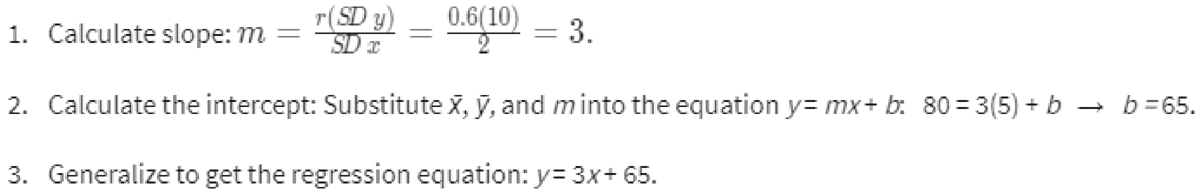
In other words, the slope of the regression line is:

$$m = \frac{r(SD\ y)}{SD\ x}$$

This is *m* in the formula for a line: *y* = *mx* + *b*. The intercept, represented by *b,* is therefore: *b* = *y* - *mx*. Because you know that point (*x̄*, *ȳ*) is always on the regression line, you can plug in the *x* and *y* values from this point to calculate the intercept. Here's an example using the original sample of five students.

	Hours studying (X)	**Exam grade (Y)**
mean:	5	80
SD:	2	10
r:	**0.6**	

Broken into steps:

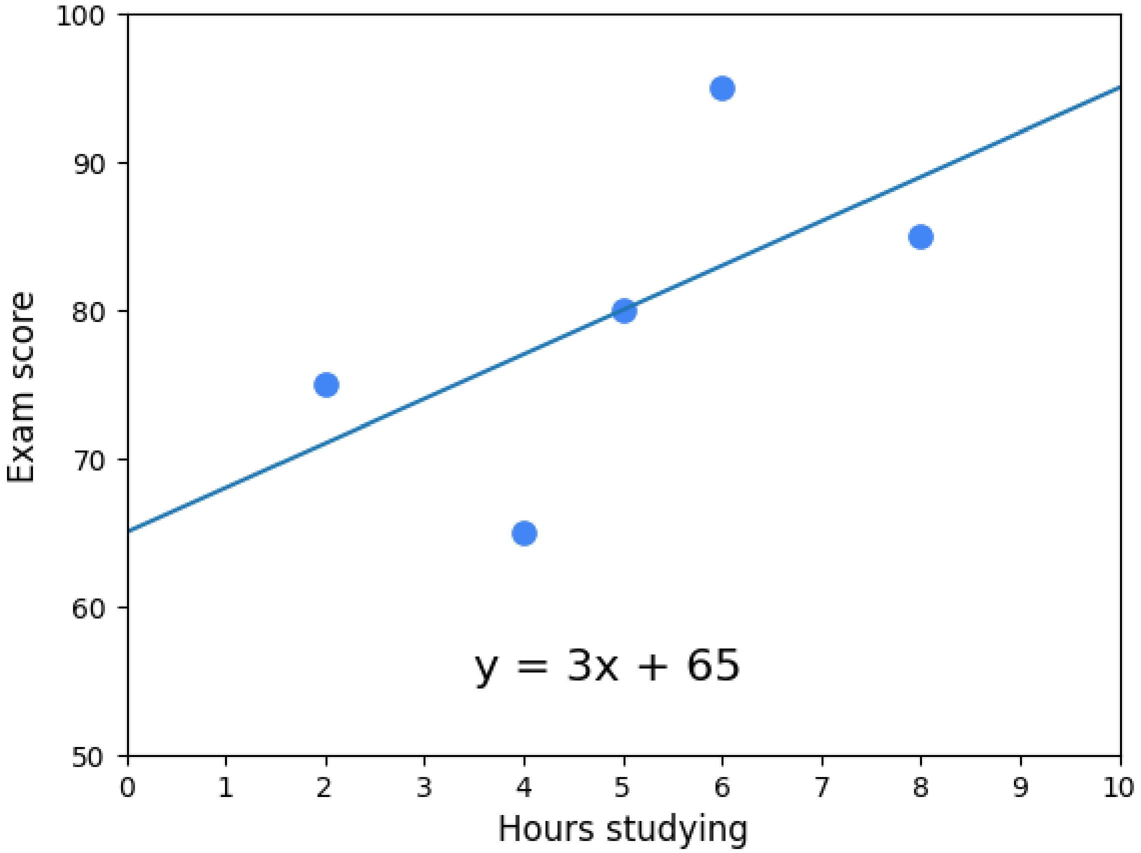


1. Calculate slope: $m = \frac{r(SD\ y)}{SD\ x} = \frac{0.6(10)}{2} = 3.$

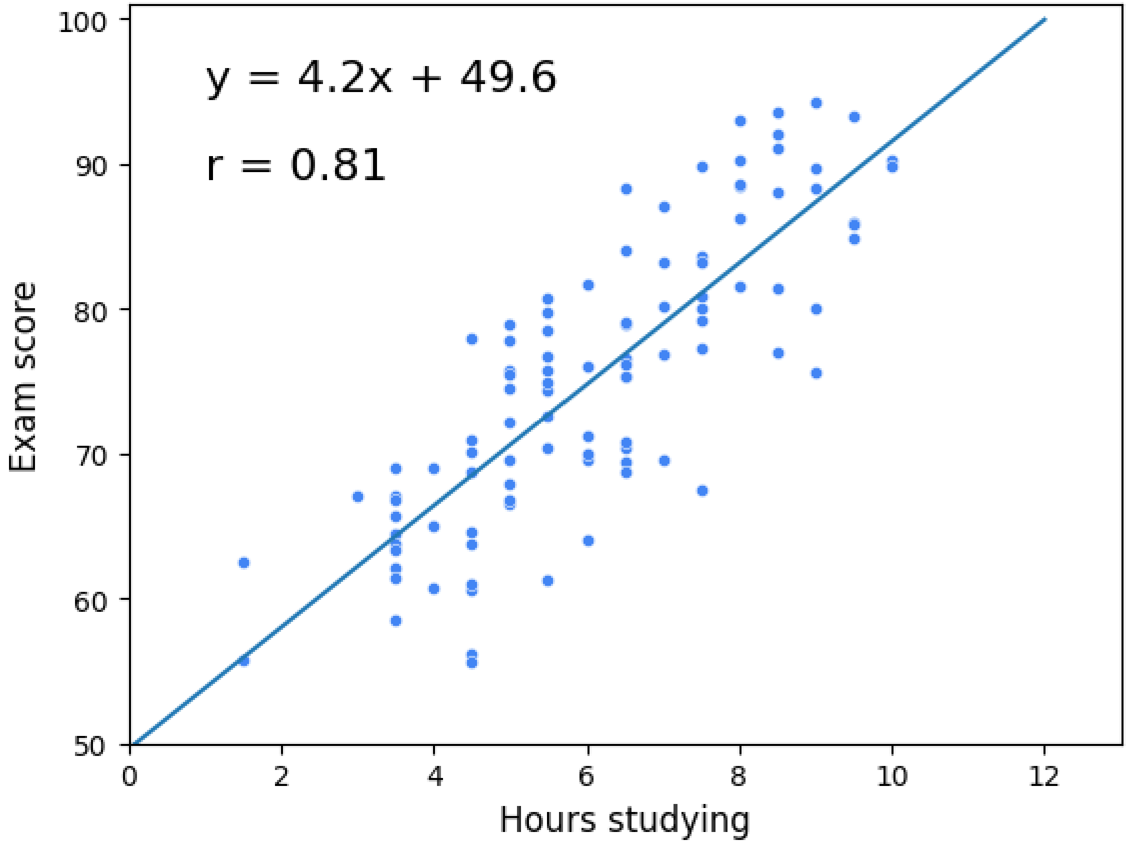
2. Calculate the intercept: Substitute $\bar{x}$, $\bar{y}$, and m into the equation $y = mx + b$: $80 = 3(5) + b \longrightarrow b = 65$.

3. Generalize to get the regression equation: $y = 3x + 65$.

Here is the regression line overlaid onto the data:

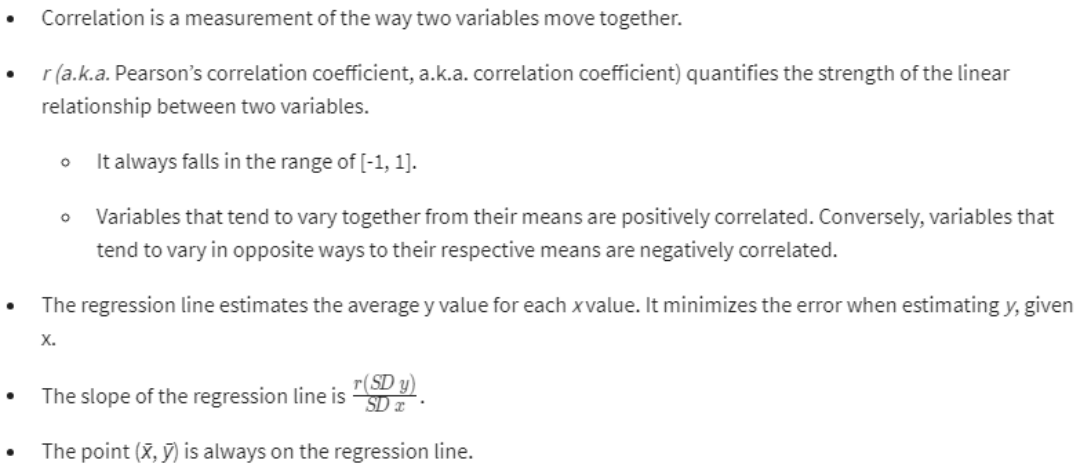


This is referred to as "the regression of Y on X." Here is the regression line for all 100 students:



Key Takeaways

Linear regression is one of the most important tools that data professionals use to analyze data. Understanding the fundamental building blocks of simple linear regression will help you as you continue learning about more complex methods of regression analysis. Here are some key points to keep in mind:



- Correlation is a measurement of the way two variables move together.
- *r (a.k.a.* Pearson's correlation coefficient, a.k.a. correlation coefficient) quantifies the strength of the linear relationship between two variables.
  - It always falls in the range of [-1, 1].
  - Variables that tend to vary together from their means are positively correlated. Conversely, variables that tend to vary in opposite ways to their respective means are negatively correlated.
- The regression line estimates the average y value for each *x* value. It minimizes the error when estimating *y*, given x.
- The slope of the regression line is $\frac{r(SD\ y)}{SD\ x}$.
- The point $(\bar{x}, \bar{y})$ is always on the regression line.

The four main assumptions of simple linear regression

In this reading, you will review the four main assumptions of simple linear regression, how to check that the assumptions are met, and what to do if an assumption is not met. You can use the additional resources to replicate the graphs and explore assumptions on your own. If there are any terms not defined in this reading, refer to the glossary of terms available throughout the course at the end of each module. This reading will cover:

- Simple linear regression assumptions
- How to check the validity of the assumptions
- What to do if an assumption is violated

Simple linear regression assumptions

To recap, there are four assumptions of simple linear regression:

1. **Linearity:** Each predictor variable (Xi) is linearly related to the outcome variable (Y).
2. **Normality:** The errors are normally distributed.*
3. **Independent Observations:** Each observation in the dataset is independent.
4. **Homoscedasticity:** The variance of the errors is constant or similar across the model.*

*Note on errors and residuals

This course has rather interchangeably used the terms "errors" and "residuals" in connection with regression. You may see this in other online resources and materials throughout your time as a data professional. In actuality, there is a difference:

- **Residuals** are the difference between the predicted and observed values. You can calculate residuals after you build a regression model by subtracting the predicted values from the observed values.
- **Errors** are the natural noise assumed to be in the model.
- Residuals are used to estimate errors when checking the normality and homoscedasticity assumptions of linear regression.

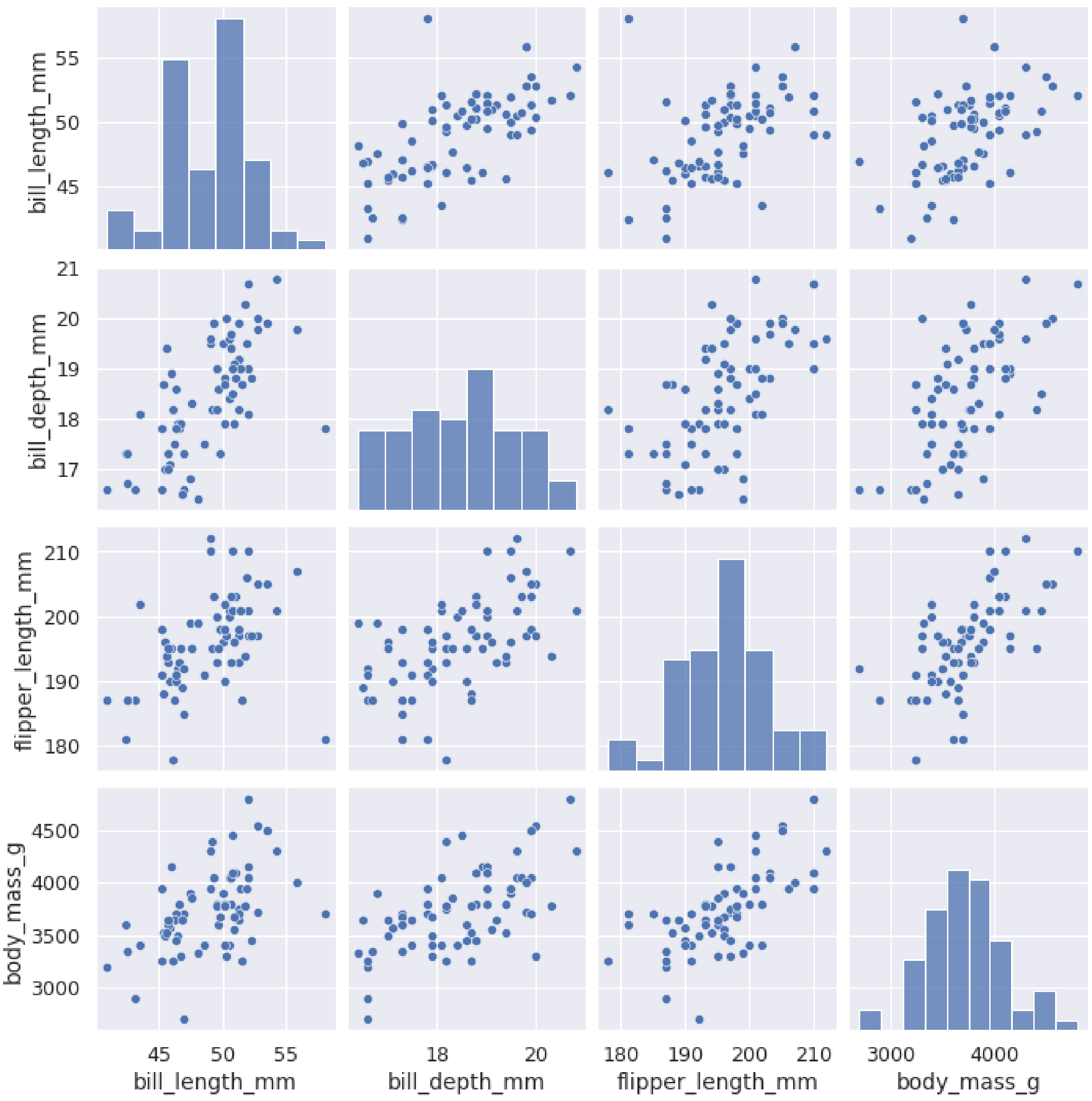
How to check the validity of the assumptions

As previously reviewed, many of the simple linear regression assumptions can be checked through data visualizations. Some assumptions can be checked before a model is built, and others can only be checked after the model is constructed, and predicted values are calculated.

Linearity

In order to assess whether or not there is a linear relationship between the independent and dependent variables, it is easiest to create a scatterplot of the dataset. The independent variable would be on the x-axis, and the dependent variable would be on the y-axis. There are a number of different Python functions that you can use to read in the data and to create a scatterplot. Some packages used for data visualizations include Matplotlib, seaborn, and Plotly. Testing the linearity assumption should occur before the model is built.

```
# Create pairwise scatterplots of Chinstrap penguins data
sns.pairplot(chinstrap_penguins)
```



Normality

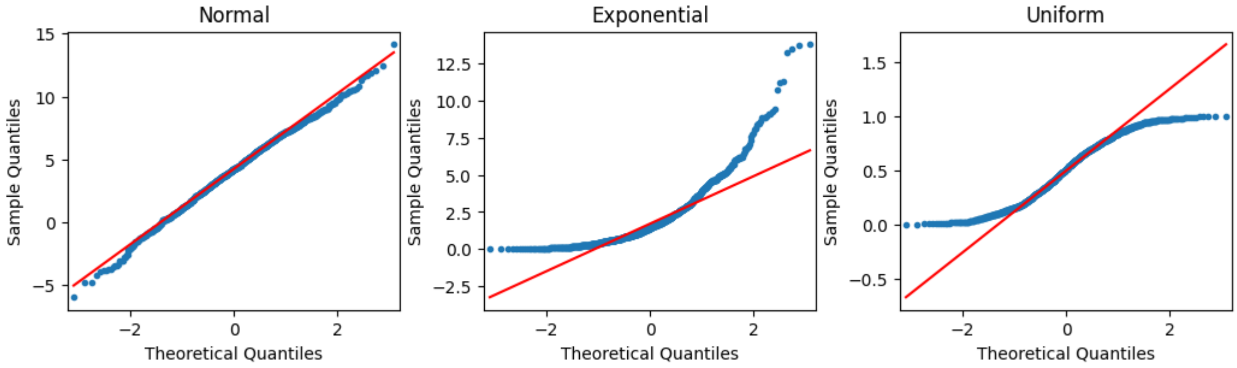
The normality assumption **focuses on the errors**, which can be estimated by the residuals, or the difference between the observed values in the data and the values predicted by the regression model. For that reason, the normality assumption can only be confirmed **after** a model is built and predicted values are calculated. Once the model has been built, you can either create a QQ-plot to check that the residuals are normally distributed, or create a histogram of the residuals. Whether the assumption is met is up to some level of interpretation.

Quantile-quantile plot

The **quantile-quantile plot** (**Q-Q plot**) is a graphical tool used to compare two probability distributions by plotting their quantiles against each other. Data professionals often prefer Q-Q plots to histograms to gauge the normality of a distribution because it's easier to discern whether a plot adheres to a straight line than it is to determine how closely a histogram follows a normal curve. Here's how Q-Q plots work when assessing the normality of a model's residuals:

1. **Rank-order the residuals**. Sort your *n* residuals from least to greatest. For each one, calculate what percentage of the data falls at or below this rank. These are the *n* quantiles of your data.
2. **Compare to a normal distribution.** Divide a standard normal distribution into *n*+1 equal areas (i.e., slice it *n* times). If the residuals are normally distributed, the quantile of each residual (i.e., what percentage of the data falls below each ranked residual) will align closely with the corresponding z-scores of each of the *n* cuts on the standard normal distribution (these can be found in a normal z-score table or, more commonly, using statistical software).
3. **Construct a plot.** A Q-Q plot has the known quantile values of a standard normal distribution along its x-axis and the rank-ordered residual values on its y-axis. If the residuals are normally distributed, the quantile values of the residuals will correspond with those of the standardized normal distribution, and both will increase linearly. If you first standardize your residuals (convert to z-scores by subtracting the mean and dividing by the standard deviation), the two axes will be on identical scales, and, if the residuals are indeed normally distributed, the line will be at a 45° angle. However, standardizing the residuals is not a requirement of a Q-Q plot. In either case, if the resulting plot is not linear, the residuals are not normally distributed.

In the following figure, the first Q-Q plot depicts data that was taken from a normal distribution. It forms a line when plotted against the quantiles of a standard normal distribution. The second plot depicts data that was drawn from an exponential distribution. The third plot uses data drawn from a uniform distribution. Notice how the second and third plots don't adhere to a line.

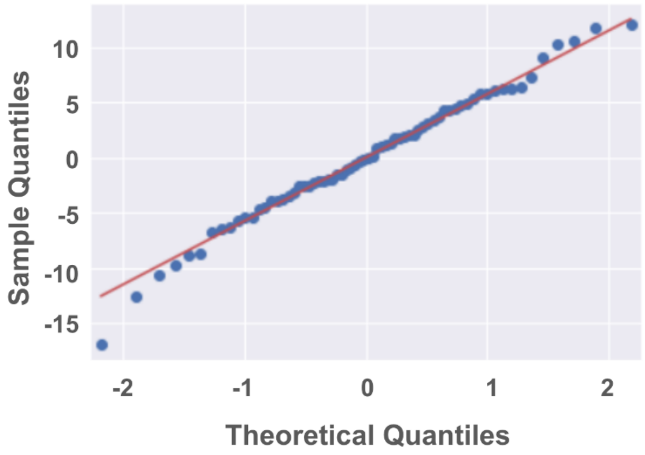


How to code a Q-Q plot

Thankfully, you don't have to manually perform the steps outlined previously. There are computing libraries to handle that. One way to create a Q-Q plot is to use the statsmodels library. If you import `statsmodels.api`, you can use the qqplot() function directly. The example below uses the residuals from a `statsmodels ols` model object. The model regresses penguins' flipper length on their bill depth (Y on X).

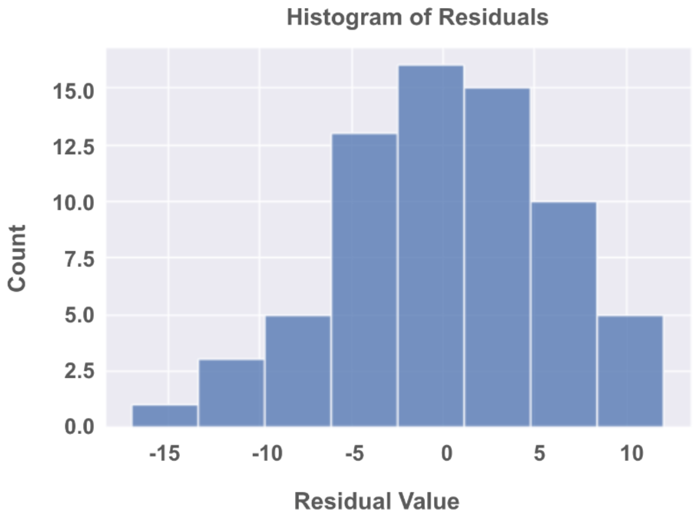
```
import statsmodels.api as sm
import matplotlib.pyplot as plt

residuals = model.resid
fig = sm.qqplot(residuals, line = 's')
plt.show()
```



And here is a histogram of the same data:

```
fig = sns.histplot(residuals)
fig.set_xlabel("Residual Value")
fig.set_title("Histogram of Residuals")
plt.show()
```



Independent Observations

Whether or not observations are independent is dependent on understanding your data. Asking questions like:

- How was the data collected?
- What does each data point represent?
- Based on the data collection process, is it likely that the value of one data point impacts the value of another data point?

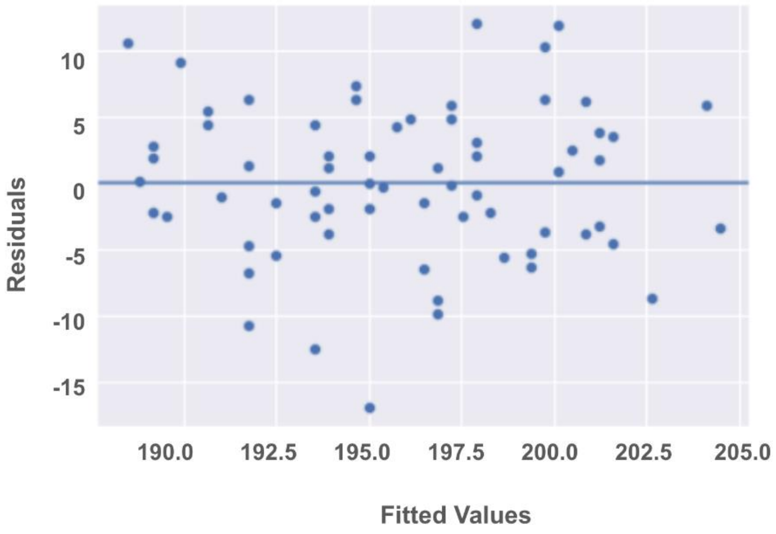
An objective review of these questions, which would include soliciting insights from others who might notice things you don't, can help you determine whether or not the independent observations assumption is violated. This in turn will allow you to determine your next steps in working with the dataset at hand.

Homoscedasticity

Like the normality assumption, the homoscedasticity assumption concerns the residuals of a model, so it can only be evaluated after a regression model has already been constructed. A scatterplot of the fitted values (i.e., the model’s predicted Y values) versus the residuals can help determine whether the homoscedasticity assumption is violated.

```
import matplotlib.pyplot as plt

fig = sns.scatterplot(fitted_values, residuals)
fig.axhline(0)
fig.set_xlabel("Fitted Values")
fig.set_ylabel("Residuals")
plt.show()
```



What to do if an assumption is violated

Now that you've reviewed the four assumptions and how to test for their violations, it’s time to discuss some common next steps you can take once an assumption is violated. Keep in mind that if you transform the data, this might change how you interpret the results. Additionally, if these potential solutions don’t work for your data, you have to consider trying a different kind of model.

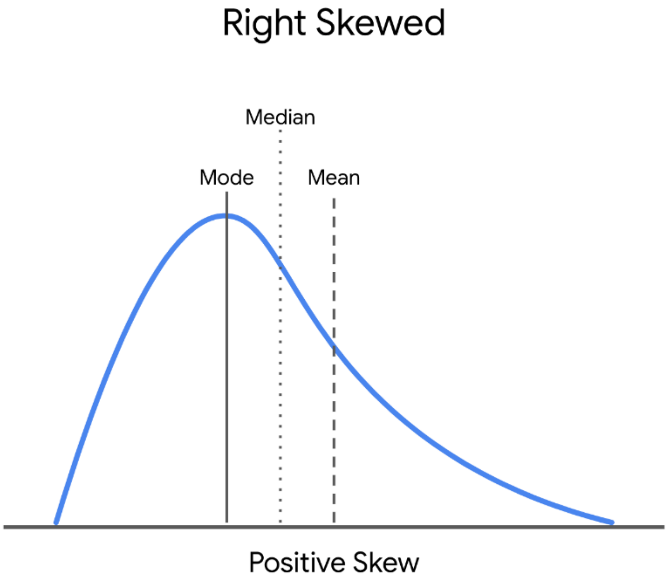
For now, focus on a few essential approaches to get you started!

Linearity

- Transform one or both of the variables, such as taking the logarithm.
  - For example, if you are measuring the relationship between years of education and income, you can take the logarithm of the income variable and check if that helps the linear relationship.

Normality

- Transform one or both variables. Most commonly, this would involve taking the logarithm of the outcome variable.
  - When the outcome variable is right skewed, such as income, the normality of the residuals can be affected. So, taking the logarithm of the outcome variable can sometimes help with this assumption.
  - If you transform a variable, you will need to reconstruct the model and then recheck the normality assumption to be sure. If the assumption is still not satisfied, you’ll have to continue troubleshooting the issue.



Independent observations

- Take just a subset of the available data.
  - If, for example, you are conducting a survey and get responses from people in the same household, their responses may be correlated. You can correct for this by just keeping the data of one person in each household.
  - Another example is if you are collecting data over a time period. Let’s say you are researching data on bike rentals. If you collect your data every 15 minutes, the number of bikes rented out at 8:00 a.m. might correlate with the number of bikes rented out at 8:15 a.m. But, perhaps the number of bikes rented out is independent if the data is taken once every 2 hours, instead of once every 15 minutes.

Homoscedasticity

- Define a different outcome variable.
  - If you are interested in understanding how a city’s population correlates with the number of restaurants in a city, you know that some cities are much more populous than others. You can then redefine the outcome variable as the ratio of population to restaurants.
- Transform the Y variable.
  - As with the above assumptions, sometimes taking the logarithm or transforming the Y variable in another way can potentially fix inconsistencies with the homoscedasticity assumption.

Key takeaways

- There are four key assumptions for simple linear regression: linearity, normality, independent observations, and homoscedasticity.
- There are different ways to check the validity of each assumption. Some assumptions can be checked before the model is built, while some can be checked after the model is built.
- There are ways to work with the data that can correct for violations of model assumptions.
- Changing the variables will change the interpretation.
- If the assumptions are violated, even after data transformations, you should consider other models for your data.

Resources for more information

- Download the seaborn penguins dataset here
- More information about the penguins dataset: Introduction to Palmer penguins
- More information about Q-Q plots: Normal Quantile-Quantile Plots (video from jbstatistics)

Code functions and documentation.

In this reading, you will review some of the code from the videos using a different subset of the penguin data. This reading will also share some tips when approaching the statsmodels documentation. This is a good opportunity to review Python functionality in conjunction with exploratory data analysis, basic data cleaning, and model construction.

Review functions from video

Load the dataset

The first few lines of code set up the coding environment and loaded the data. As you might be familiar with, you can call on the **`import`** function to import any necessary packages. You should use conventional aliases as needed. The example below references a dataset on penguins available through the **`seaborn`** package.

```
# Import packages
import pandas as pd
import seaborn as sns

# Load dataset
penguins = sns.load_dataset("penguins")

# Examine first 5 rows of dataset
penguins.head()
```

Clean data

After loading the data, the data was cleaned up to create a subset of data for the purposes of our course. The example isolates just the Chinstrap penguins from the dataset and drops rows with missing data.

The index of the dataframe is reset using the **`reset_index()`** function. When you subset a dataframe, the original row indices are retained. For example, let's say there were Adelie or Gentoo penguins in rows 2 and 3. By subsetting the data just for Chinstrap penguins, your new dataframe would be listed as row 1 and then row 4, as rows 2 and 3 were removed. By resetting the index of the dataframe, the row numbers become rows 1, 2, 3, etc. The data frame becomes easier to work with in the future.

Review the code below. You are encouraged to run the code in your own notebook.

```
# Subset just Chinstrap penguins from data set
chinstrap_penguins = penguins[penguins["species"] == "Chinstrap"]

# Reset index of dataframe
chinstrap_penguins.reset_index(inplace = True, drop = True)
```

Setup for model construction

Now that the data is clean, you are able to plot the data and construct a linear regression model. First, extract the one X variable, **`bill_depth_mm`**, and the one Y variable, **`flipper_length_mm`**, that you are targeting.

```
ols_data = chinstrap_penguins[["bill_depth_mm", "flipper_length_mm"]]
```

Because this example is using statsmodels, save the ordinary least squares formula as a string so the computer can understand how to run the regression. The Y variable, **`flipper_length_mm`** comes first, followed by a tilde and the name for the X variable, **`bill_depth_mm`**.

```
# Write out formula
ols_formula = "flipper_length_mm ~ bill_depth_mm"
```

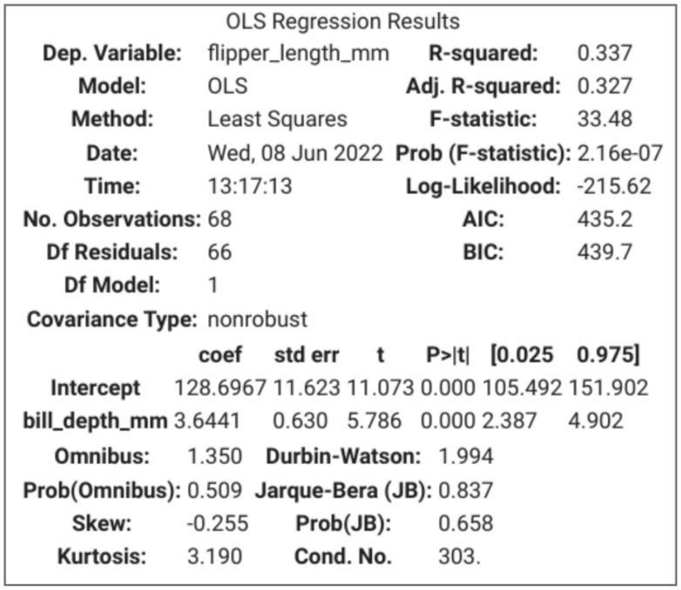
Construct the model

In order to construct the model, you'll first need to import the **`ols`** function from the **`statsmodels.formula.api`** interface.

```
# Import ols function
from statsmodels.formula.api import ols
```

Next, plug in the formula and the saved data into the **`ols`** function. Then, use the **`fit`** method to fit the model to the data. Lastly, use the **`summary`** method to get the results from the regression model.

```
# Build OLS, fit model to data
OLS = ols(formula = ols_formula, data = ols_data)
model = OLS.fit()
model.summary()
```



```
                       OLS Regression Results
   Dep. Variable:   flipper_length_mm       R-squared:   0.337
           Model:   OLS                Adj. R-squared:   0.327
          Method:   Least Squares         F-statistic:   33.48
            Date:   Wed, 08 Jun 2022 Prob (F-statistic): 2.16e-07
            Time:   13:17:13             Log-Likelihood: -215.62
No. Observations:   68                            AIC:   435.2
    Df Residuals:   66                            BIC:   439.7
        Df Model:   1
 Covariance Type:   nonrobust
                    coef   std err    t    P>|t|  [0.025   0.975]
       Intercept  128.6967 11.623  11.073  0.000 105.492  151.902
bill_depth_mm     3.6441    0.630   5.786  0.000   2.387    4.902
       Omnibus:     1.350  Durbin-Watson:    1.994
Prob(Omnibus):     0.509  Jarque-Bera (JB): 0.837
          Skew:    -0.255       Prob(JB):    0.658
      Kurtosis:     3.190       Cond. No.    303.
```

Model predictions and residuals

You can access the predictions and residuals from a fitted statsmodels.regression.linear_model.OLS or statsmodels.regression.linear_model.OLSResults object as follows.

Predictions

Use the model's predict() method, passing to it an array containing the values of the independent variable(s):

```
predictions = model.predict(chinstrap_penguins[["bill_depth_mm"]])
```

Residuals

Use the model's **`resid`** attribute:

```
residuals = model.resid
```

Navigating statsmodels documentation

It can require significant work to approach a new Python package or a new set of Python functions, especially when first coding. The benefit of Python being an open source programming language is that there is a strong Python community asking and answering questions. Part of being a successful data professional is knowing how to make your code work and troubleshooting when your code breaks. One way to do this is to go directly to the source, or the official documentation of a particular package.

You've been using the **`statsmodels`** package to build simple linear regression models. The statsmodels documentation is available online and is updated regularly. Specifically, you are using the statsmodels.formula.api interface to perform ordinary least squares estimation.

Examining the page on the **`ols`** function or the function that performs OLS estimation, you will observe the different function parameters that are allowed, with some notes about each.

Unfortunately, at this time, the statsmodels documentation does not include code examples of how to use the function. If you find documentation that doesn't provide as many examples as you need, or documentation that doesn't provide examples that you need to troubleshoot your code, remember that you can always search online for the function you're trying to use and explore how others in the Python community have handled comparable problems.

Key takeaways

- You can review this reading to refresh your memory about the code in the corresponding video.
- You can view the statsmodels (or any other package's) documentation as needed.
- If the documentation does not hold the answer you're looking for, you can always turn to the internet to check out other people's work.

Interpret measures of uncertainty in regression

Goal of Reading

In this reading, we will continue exploring uncertainty in regression analysis, specifically through confidence intervals, confidence bands, and p-values. Together, we will:

- Review key concepts
- Discuss how to interpret measures of uncertainty
- Review sample graphs

Review of Concepts

Recall that we can represent a simple linear regression line as

$$y = \beta_0 + \beta_1 X.$$

Since regression analysis utilizes **estimation** techniques, there is always a level of uncertainty surrounding the predictions made by regression models. To represent the error, we can actually rewrite the equation to include an error term, represented by the letter ϵ (pronounced "epsilon"):

$$y = \beta_0 + \beta_1 X + \epsilon.$$

There is one residual, also known as the difference between the predicted and actual value, for each data point in the dataset used to construct the model. We can then quantify how uncertain the entire model is through a few measures of uncertainty:

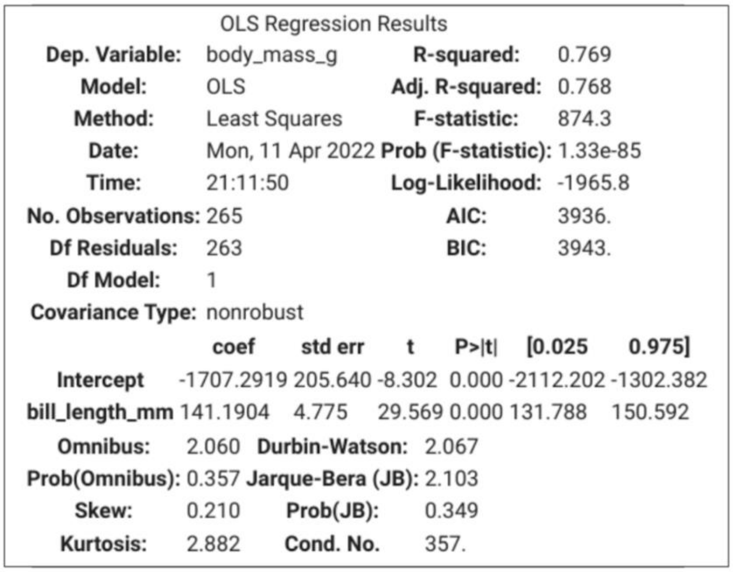
- **Confidence intervals** around beta coefficients
- **P-values** for the beta coefficients
- **Confidence band** around the regression line

You can refer to the glossary of terms to check any key terms and definitions, but we've provided the two key terms here:

- **Confidence interval:** a range of values that describes the uncertainty surrounding an estimate
- **P-value:** the probability of observing results as extreme as those observed when the null hypothesis is true

Interpreting Uncertainty

Let's first revisit the summary of results from the linear regression model we created together in prior videos:



```
                        OLS Regression Results
  Dep. Variable:   body_mass_g            R-squared:   0.769
          Model:   OLS                Adj. R-squared:  0.768
         Method:   Least Squares          F-statistic: 874.3
           Date:   Mon, 11 Apr 2022 Prob (F-statistic): 1.33e-85
           Time:   21:11:50           Log-Likelihood:  -1965.8
No. Observations:  265                           AIC:  3936.
    Df Residuals:  263                           BIC:  3943.
        Df Model:  1
 Covariance Type:  nonrobust
                     coef       std err   t      P>|t|  [0.025     0.975]
     Intercept    -1707.2919 205.640 -8.302   0.000 -2112.202 -1302.382
bill_length_mm    141.1904   4.775    29.569  0.000  131.788    150.592
      Omnibus:   2.060  Durbin-Watson:     2.067
Prob(Omnibus):   0.357 Jarque-Bera (JB):   2.103
         Skew:   0.210     Prob(JB):       0.349
     Kurtosis:   2.882     Cond. No.       357.
```

According to the simple linear regression model we built,

$\hat{\beta}_1$ is 141.1904.

So for every one-millimeter increase in the bill length of a penguin, we would expect a penguin to have about 141.1904 more grams in body mass. The estimate has a p-value of 0.000, which is less than 0.05, meaning that the coefficient is “statistically significant.” Additionally our estimate has a 95% confidence interval of 131.788 and 150.592. Let’s review these short sentences a bit more.

Previously you may have learned about p-values and confidence intervals within the context of hypothesis testing. Even though it may seem unintuitive, even in regression analysis we are testing hypotheses.

P-values

When running regression analysis, you want to know if X is really correlated with y or not. So we do a hypothesis test on the regression results. In regression analysis, for each beta coefficient, we are testing the following set of null and alternative hypotheses:

- H_0 (null hypothesis): $\beta_1 = 0$
- H_1 (alternative hypothesis): $\beta_1 \neq 0$

In our example, because the p-value is less than 0.05, we can reject the null hypothesis that β_1 is equal to 0, and state that the coefficient is statistically significant, which means that a difference in bill length of a penguin is truly correlated with a difference in body mass.

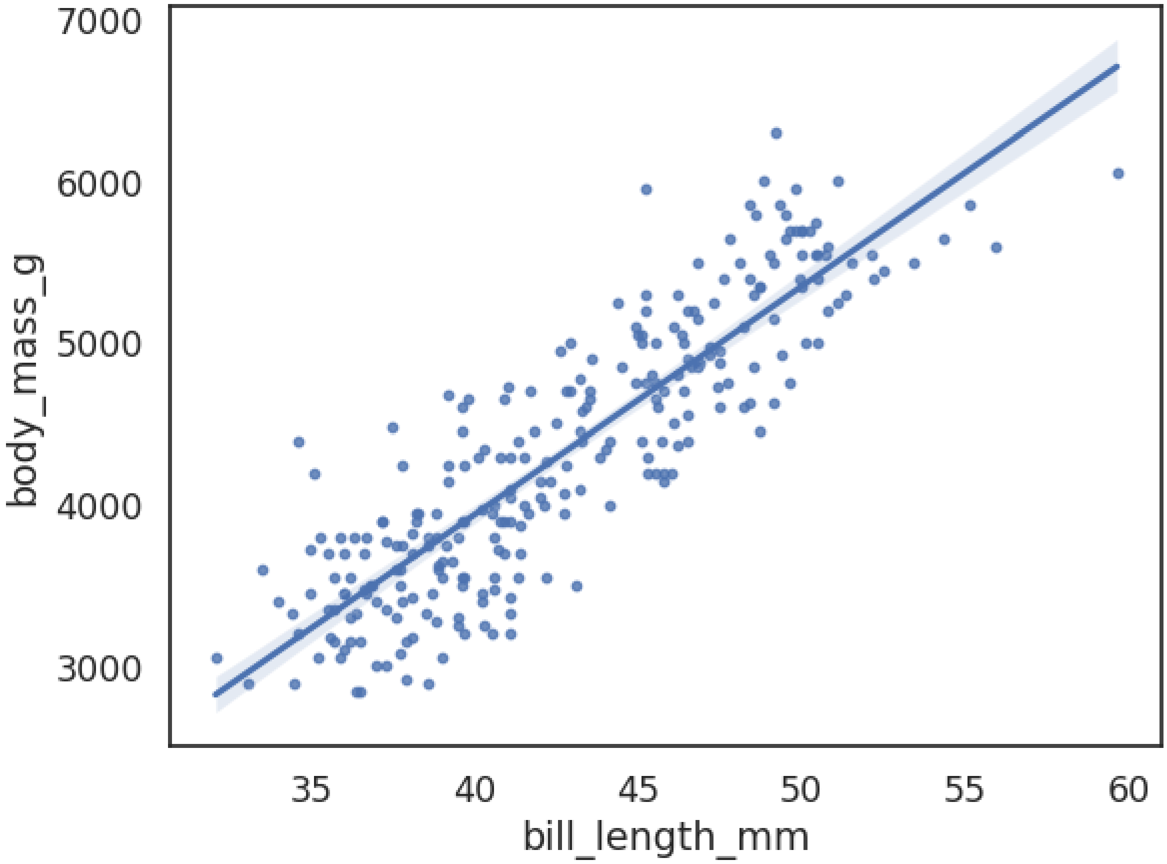
Confidence Intervals

Each beta coefficient also has a confidence interval associated with its estimate. A 95% interval means the interval itself has a 95% chance of containing the true parameter value of the coefficient. So there is 5% chance that our confidence interval [131.788, 150.592] does not contain the true value of β_1. More precisely, this means that if you were to repeat this experiment many times, 95% of the confidence intervals would contain the true value of β_1.

But, since there is uncertainty in both of the estimated beta coefficients, then the estimated y values also have uncertainty. This is where confidence bands become useful.

Example Graph

- **Confidence band:** the area surrounding the line that describes the uncertainty around the predicted outcome. You can think of the confidence band as representing the confidence interval surrounding each point estimate of y. Since there is uncertainty at every point in the line, we use the confidence band to summarize the confidence intervals across the regression model. The confidence band is always narrowest towards the mean of the sample and widest at the extremities.



Key Takeaways

- Regression analysis utilizes **estimation** techniques, so there is always uncertainty around the predictions.
- We can measure uncertainty using confidence intervals, p-values, and confidence bands.
- For every coefficient estimate, we are testing the hypothesis that the coefficient equals 0.

Evaluation metrics for simple linear regression

In this reading, we’ll provide a more comprehensive overview about evaluation metrics for simple linear regression. In a prior video we covered R², and mentioned a few other metrics, MAE and MSE. In this reading, we will review the metrics we’ve previously mentioned, and introduce a few more as well that you may encounter throughout your career as a data professional.

Review of $\mathbf{R^2}$, MSE, and MAE

The main evaluation metric for linear regression is R², or the coefficient of determination.

R²: The coefficient of determination

$\mathbf{R^2}$ measures the proportion of variation in the dependent variable, Y, explained by the independent variable(s), X.

- This is calculated by subtracting the sum of squared residuals divided by the total sum of squares from 1.

$$R^2 = 1 - \frac{\text{Sum of squared residuals}}{\text{Total sum of squares}}$$

R² ranges from 0 to 1. So, if a model has an R² of 0.85, that means that the X variables explain about 85% of the variation in the Y variable. Although R² is a highly interpretable and commonly used metric, you may also encounter mean squared error (MSE) and mean absolute error (MAE) when R² is insufficient in evaluating model performance.

MSE: Mean squared error

MSE (mean squared error) is the average of the squared difference between the predicted and actual values.

- Because of how MSE is calculated, MSE is very sensitive to large errors.

MAE: Mean absolute error

MAE (mean absolute error) is the average of the absolute difference between the predicted and actual values.

- If your data has outliers that you want to ignore, you can use MAE, as it is not sensitive to large errors.

Other evaluation metrics

Beyond the three metrics listed above, you may also encounter **AIC (Akaike information criterion) and BIC (Bayesian information criterion)**.

Lastly, there is **adjusted R^2**, which will be addressed in more detail in upcoming videos. It is a variation of R^2 that accounts for having multiple independent variables present in a linear regression model.

Key takeaways

- There are many evaluation metrics to choose from with regard to simple linear regression.
- The most common evaluation metric you'll encounter is probably R^2. But, there are times when R^2 is insufficient or inappropriate to use.
- Based on your experiences and the particulars of a metric, you can use your best judgment to select an appropriate metric to evaluate a regression model.

Correlation versus causation: Interpret regression results

In previous videos, you learned that correlation is not causation. In this reading, you will continue to explore the differences between correlation and causation so that you will be prepared to report regression results responsibly, honestly, and effectively.

What is correlation?

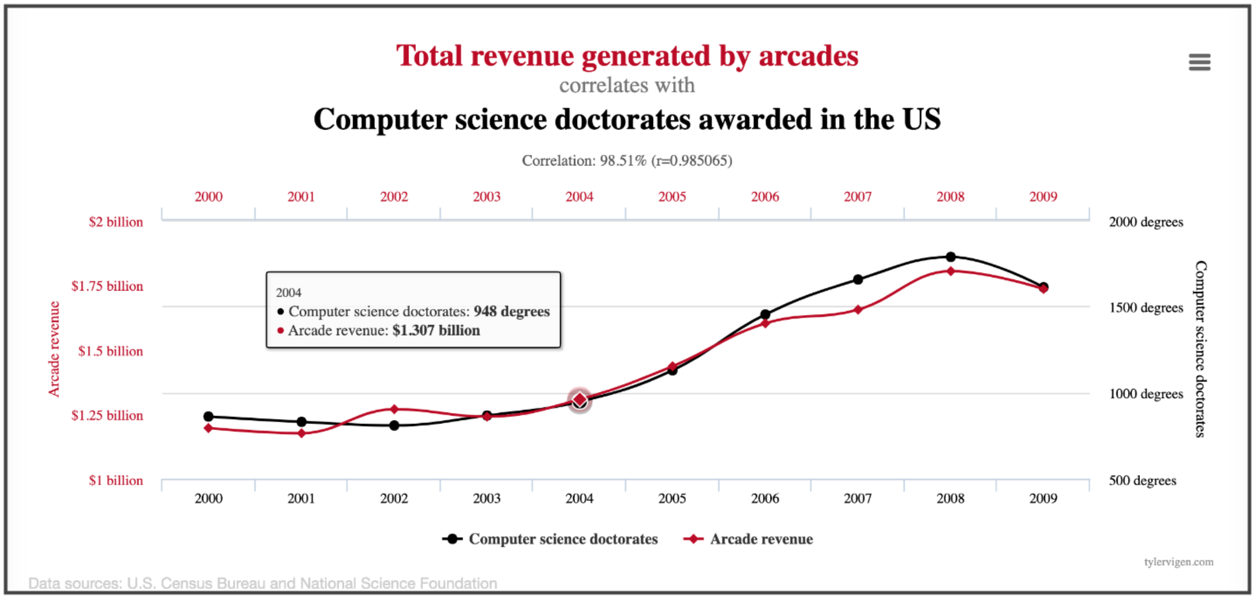
You might recall that there are two main kinds of correlation: positive and negative correlation.

- **Positive correlation** is a relationship between two variables that tend to increase or decrease together.
- **Negative correlation** is an inverse relationship between two variables, where when one variable increases, the other variable tends to decrease, and vice versa.

To generalize, **correlation** measures the way two variables tend to change together. There is a metric called the Pearson correlation coefficient that ranges from -1 to 1 that can measure the relationship between two variables.

Note that correlation is just observational. Two variables can be correlated–they tend to change together without one variable causing the other variable to change. In fact, there is an entire website and book, *Spurious Correlations*, devoted to documenting interesting and unexpected correlations between variables.

For example, here is a graph illustrating the correlation between total revenue generated by arcades and computer science doctorates awarded in the United States. Over time, computer science doctorates and arcade revenue increase at about the same rate. So, the graph definitely shows a correlation between the two variables, but it's pretty hard to argue that one causes the other.



Data sources: U.S. Census Bureau and National Science Foundation

It's difficult to claim causation

Previously you learned that **causation** describes a cause-and-effect relationship where one variable directly causes the other to change in a particular way. Although this is an intuitive definition, proving causation requires a lot of particular circumstances to be met.

To argue for causation between variables, in general, you must run a **randomized controlled experiment**. The following are some key components of a proper randomized controlled experiment:

- You must control for every factor in the experiment.
- You must have a control group under certain conditions.
- You must have at least one treatment group under certain conditions.
- The difference(s) between the control and treatment groups must be observable and measurable.

Setting up a randomized controlled experiment is quite laborious and intensive. There are a number of requirements and factors not included in this reading, but there is a lot of information online and in academic research that you can explore on your own. Understanding the basics of causal claims allows you to responsibly report the results of your data analysis.

Correlation leads to interesting insights

When working as a data professional, you often do not have complete control of how the data is collected. You or your team might not be able to run a randomized controlled experiment. But, even if you cannot make causal claims, correlational research can still yield interesting results that have meaningful business implications.

Scenario 1: Optimizing athletic performance

Suppose a runner is training for a race. There are many ways to track health data—from built-in apps to paid-for apps. But, there are also so many factors that can contribute to the runner's performance—how much water they drank, how sore their muscles are on a given day, the weather, how much sleep they got, what equipment they are using on race day, and the clothes they are wearing. It's very hard to claim that any one factor would make or break their race time. But, over time, one might observe patterns in how sleep, soreness, water, clothing, and other factors tend to correlate with performance. This is why athletes can be so particular about brands of equipment, their diet, and pre-race day routines.

Claims you can make (correlation)

- When the runner drinks more water the day before a race, they tend to have more stamina.
- When the runner doesn't run long distances the week before a race, they tend to feel better on race day.

Claims you cannot make (causation)

- Drinking more water the day before a race causes the runner to run faster.
- Not running long distances the week before a race causes the runner to run faster.

Scenario 2: Improving food quality

Perhaps you’re a new chef at a restaurant or you’re cooking for yourself or family. Every time you make a dish, there are a lot of variables: what pan was used, when the ingredients were purchased, if the ingredients are in season, and how hungry everyone was. Any one of these factors can change how “good” the dish is. But, this data is valuable regardless of causal claims. Over time, you can hone your cooking skills for this particular dish to ensure better food quality.

These are just two examples where gathering data to understand correlation between factors can drastically improve outcomes. The same principles can be applied on a larger scale, with big data, and in different industries, depending on the desired outcome you want to optimize.

Claims you can make (correlation)

- When I use fresher ingredients, the final dish tends to taste better.
- When I am very hungry, the final dish tends to taste better.

Claims you cannot make (causation)

- Using fresher ingredients makes the dish taste better.
- Being hungrier makes the dish taste better.

Key takeaways

- Claiming causation requires specific circumstances that are often not within your control.
- Correlation analyses are an incredibly useful tool for data professionals, and can provide interesting insights and actionable next steps.

Multiple linear regression scenarios Goals of Reading

Now that you have learned what multiple linear regression is, in this reading, you will explore three scenarios in which multiple regression models can help a company or organization understand a business problem. The goal of the reading is to understand the versatility of multiple linear regression, and to get you thinking about various applications of this powerful and flexible regression technique.

Scenario 1: Selling graphic design services

Let's say that you're a data professional working at a company that sells graphic design services. The company you work for might be interested in understanding the factors related to customer satisfaction and retention. There are many ways you can measure this, and you can use any of the following factors to develop a promising multiple linear regression model.

Potential dependent variables (Y)

- Customer satisfaction
- Number of returning customers
- Net Promoter Score
- Satisfaction with customer service

Potential independent variables (X)

- Cost of services
- Customer service response time
- Adding new graphic design packages
- Changing page layout

Scenario 2: Running a restaurant

Imagine that you are working at a restaurant, and you want to determine how to improve the success of your business. Like any other client-facing business, you want to keep your costs down, your revenue high, and your customers happy. Similar to the prior example, there are many ways to measure the restaurant's success. There are also a number of variables that could be correlated with the chosen metric of success.

Potential dependent variables (Y)

- Total revenue
- Number of reviews online
- Number of five-star reviews online
- Number of reservations per week

Potential independent variables (X)

- Spending on advertising/marketing
- Operational costs
- Size of menu
- Foot traffic
- Cancellation of reservations
- Business partnerships (ex: delivery apps, farmers' markets, community organizations)

Scenario 3: Agricultural production

Suppose you are working in agricultural production, perhaps on a farm or a ranch. Even though this is a very different environment from a restaurant or online service, multiple regression can still be helpful. For example, let's say that you are trying to predict crop yield, revenue for the season, or amount of crops sold. From the weather to soil conditions to labor and resource usage, there are many factors that could contribute to a good year or a bad year for a farm or any kind of agricultural production. Multiple regression can be used to help better plan and predict for worse years.

Potential dependent variables (Y)

- Crop yield
- Revenue
- Crops sold

Potential independent variables (X)

- Weather (rainfall, temperature)
- Nutrients in soil
- Historic crop yield
- Cost of fertilizer
- Cost of fuel, water, or energy used to maintain crops
- Cost of labor
- Partnerships with local restaurants or, grocery stores

Key Takeaways

- Multiple regression is a versatile and effective way to understand and describe more complex relationships between variables.
- Multiple regression can be used in a variety of industries and contexts.

Resources for more information

- "Multiple Regression: Definition, Uses, and 5 Examples." *Indeed Editorial Team*.
- "Multivariate Regression Analysis | STATA Data Analysis Examples." *UCLA: Statistical Consulting Group.*

multicollinearity

In prior videos, you have learned about linear regression assumptions. In this reading, you will build off that knowledge base to extend your understanding of multiple linear regression assumptions. This reading will help you review assumptions that apply to both simple linear regression and multiple linear regression, and will then focus more heavily on the concept of multicollinearity.

Multiple linear regression assumptions

Recall that simple linear regression has four main assumptions that provide validity to the results derived from the analysis. To this list of four assumptions, we add the no multicollinearity assumption when working with multiple linear regression.

1. **Linearity:** Each predictor variable (X_i) is linearly related to the outcome variable (Y).
2. **(Multivariate) normality:** The errors are normally distributed.
3. **Independent observations:** Each observation in the dataset is independent.
4. **Homoscedasticity:** The variation of the errors is constant or similar across the model.
5. **No multicollinearity:** No two independent variables (X_i and X_j) can be highly correlated with each other.

*Note on errors and residuals

As noted earlier, "residuals" and "errors" are sometimes used interchangeably, but there is a difference. We use residuals to estimate errors when we are checking the normality and homoscedasticity assumptions of linear regression.

- **Residuals** are the difference between the predicted and observed values. You can calculate residuals after you build a regression model by subtracting the predicted values from the observed values.
- **Errors** are the natural noise assumed to be in the model.

Extending prior assumptions

Much of what you learned about the first four assumptions with regard to simple linear regression can be directly applied to multiple linear regression. The code might be slightly different or longer, but the rationale is the same.

Linearity

- With multiple linear regression, you need to consider whether each *x* variable has a linear relationship with the *y* variable.
- You can make multiple scatterplots instead of just one, using seaborn's pairplot() function, or the scatterplot() function multiple times. Other libraries with plotting capabilities will have similar functions.

Independent observations

- The independent observations assumption is still primarily focused on data collection.
- You can check the validity of the assumption in the same way you would with simple linear regression.

(Multivariate) Normality

- Just as with simple linear regression, you can construct the model, and then create a Q-Q plot of the residuals.
- If you observe a straight diagonal line on the Q-Q plot, then you can proceed in your analysis. You can also plot a histogram of the residuals and check if you observe a normal distribution that way.
- **Note:** It's a common misunderstanding that the independent and/or dependent variables must be normally distributed when performing linear regression. This is not the case. Only the model's residuals are assumed to be normal.

Homoscedasticity

- As with simple linear regression, for multiple linear regression, just create a plot of the residuals vs. fitted values.
- If the data points seem to be scattered randomly across the line where residuals equal 0, then you can proceed.

How to check the no multicollinearity assumption

The no multicollinearity assumption is unique to multiple linear regression as it focuses on potential relationships between different independent (X) variables. When assessing the no multicollinearity assumption, you're interested in identifying any linear relationships between the independent (X) variables. X variables that are linearly related could muddle the interpretation of the model's results. If there are X variables that are linearly related, it is usually best to remove some independent variables from the model.

Note, however, that the assumption of no multicollinearity is most important when you are using your regression model to make inferences about your data, because the inclusion of collinear data increases the standard errors of the model's beta parameter estimates. But there may be times when the primary purpose of your model is to make predictions and the need for accurate predictions outweighs the need to make inferences about your data. In this case, including the collinear independent variables may be justified because it's possible that their inclusion would result in better predictions.

There are a few ways to check the no multicollinearity assumption. This reading will cover two of them. One is purely visual, and the other is numerical in nature. Both can be done prior to building the linear regression model.

Scatterplots or Scatterplot Matrix

A visual way to identify multicollinearity between independent (X) variables is using scatterplots or scatterplot matrices. The process is the same as when you checked the linearity assumption, except now you're just focusing on the X variables, not the relationship between the X variables and the Y variable. If you're using the seaborn library, you can use the **`pairplot`** function, or the **`scatterplot`** function multiple times.

Variance Inflation Factors (VIF)

Calculating the variance inflation factor, or VIF, for each independent (X) variable is a way to quantify how much the variance of each variable is "inflated" due to correlation with other X variables. You can read more about VIFs on the Pennsylvania State University's Eberly College of Science website or on the website for Vilnius University's e-book on *Practical Econometrics and Data Science*. The details of calculating VIF are beyond the scope of this course, but it's helpful to know that VIF_i represents the amount that the standard error of coefficient β_i increases relative to a situation in which all of the predictor variables are uncorrelated.

To calculate the VIF for each predictor variable, you can use the variance_inflation_factor() function from the **statsmodels** package. Here is an example of how you might obtain VIFs for your predictor variables.

```
from statsmodels.stats.outliers_influence import variance_inflation_factor
X = df[['col_1', 'col_2', 'col_3']]
vif = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]
vif = zip(X, vif)
print(list(vif))
```

The smallest value a VIF can take on is 1, which would indicate 0 correlation between the X variable in question and the other predictor variables in the model. A high VIF, such as 5 and above, according to the statsmodels documentation, can indicate the presence of multicollinearity.

What to do if there is multicollinearity in your model

Variable Selection

The easiest way to handle multicollinearity is simply to only use a subset of independent variables in your model. For example, if your multiple linear regression model is something like this:

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3$$

But if X_1 and X_3 are highly correlated, then you can choose to include only X_1 or X_3 in your final model, but not both.

There are a few specific statistical techniques you can use to select variables strategically. You'll learn about these more in future videos:

- Forward selection
- Backward elimination

Advanced Techniques

In addition to the techniques listed above that will be covered in-depth in this course, there are more advanced techniques that you may come across in your career as a data professional, such as:

- Ridge regression
- Lasso regression

- Principal component analysis (PCA)

These techniques can result in more accurate and predictive models, but can complicate the interpretation of regression results.

Key Takeaways

- Many of the assumptions of simple linear regression extend readily to multiple linear regression.
- You can use scatterplots and variance inflation factors to check for multicollinearity in a regression model.
- There are different techniques for variable selection to remove multicollinearity in a model.

Underfitting and overfitting

As you have been learning, a multiple regression model is built using sample data from the population of interest with the goal of applying the model to unseen data from the population and getting reliable results. **Underfitting** and **overfitting** are two obstacles that the multiple regression model must mitigate so it can be applicable. In this reading, you will gain a general understanding of underfitting and get a closer look at overfitting.

The two ways a model can be unreliable

Underfitting

In the case of underfitting, a multiple regression model fails to capture the underlying pattern in the outcome variable. An underfitting model has a low R-squared value.

A model can underfit the data for a variety of reasons. The independent variables in the model might not have a strong relationship with the outcome variable. In this situation, different or additional predictors are needed. It could be the case that the sample dataset is too small, and this prevents the model from being able to learn the relationship between the predictors and the outcome. Using more sample data to build the model might reduce the problem of underfitting.

Consider the example of a multiple regression model that predicts the resale price of a pre-owned car. This model has two predictors: the color of the car and the year it was manufactured. The model's R-squared value is quite low. This indicates that the model is underfitting because the current predictors do not have a strong relationship with the car's resale price. There are likely other important predictors missing from the multiple regression model, like the mileage on the car or the make of the car.

There are additional reasons that a multiple regression model might underfit the data, and the methods used to reduce this obstacle depend on the specific context. Because an underfitting multiple regression model is not able to capture the relationship between predictors and outcome in the sample data, this model will also not be able to produce reliable results when it is used on unseen data from the population.

The difference between training data and test data

Before you learn more about overfitting, it is important to cover a step data scientists take before building a multiple regression model. They divide the sample data into two categories called **training data** and **test data**. Training data is used to build the model, and test data is used to evaluate the model's performance after it has been built. Splitting the sample data in this way is also called **holdout sampling**, with the holdout sample being the test data. Holdout sampling allows data scientists to evaluate how a model performs on data it has not experienced yet.

The holdout sample might also be called the **validation data**. Regardless, the general idea remains the same: this is the data that is used to evaluate the model.

Data scientists obtain the training and test data by randomly splitting the sample dataset so that each record exclusively belongs to one of the two categories. This way, some records are used as the training data and other records are used as the test data.

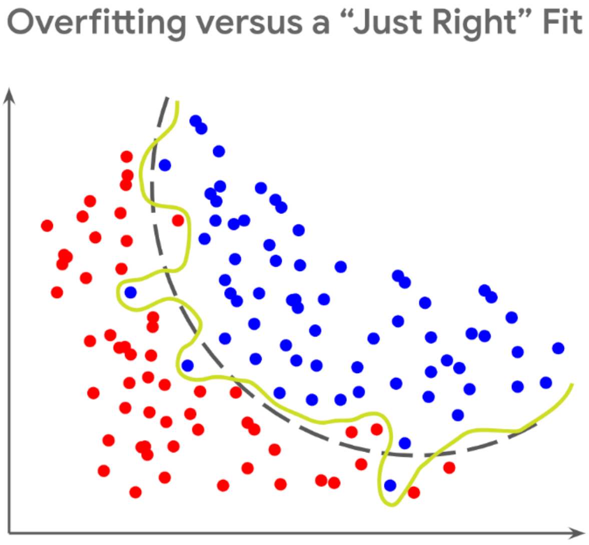
Overfitting

Underfitting causes a multiple regression model to perform poorly on the training data, which indicates that the model performance on test data will also be substandard. In contrast, overfitting causes a model to perform well on training data, but its performance is considerably worse when evaluated using the unseen test data. That's why data scientists compare model performance on training data versus test data to identify overfitting.

Why is there a discrepancy between an overfitting model's performance on training data versus test data?

An overfitting model fits the observed or training data too specifically, making the model unable to generate suitable estimates for the general population. This multiple regression model has captured the **signal** (i.e. the relationship between the predictors and the outcome variable) *and* the **noise** (i.e. the randomness in the dataset that is not part of that relationship). You cannot use an overfitting model to draw conclusions for the population because this model **only** applies to the data used to build it.

In the plot below, the dashed black line represents an optimal multiple regression model that performs well in distinguishing between the red and blue dots without overfitting the data. In contrast, the squiggly yellow line represents a model that overfits the data. Although this line might even do a slightly better job of separating the blue dots from the red ones, it is too specific to this data and will not perform well on another sample from the same population. In contrast, the black line will be continuously reliable in distinguishing between the two colors.



Why does overfitting result in a higher R-squared value?

Earlier you learned that R-squared is a goodness of fit measure because it tells you the proportion of variance in the outcome variable that is captured by the independent variables in the multiple regression model. However, as you add more independent variables to a model, the associated R-squared value will increase regardless of whether or not those predictors have a strong relationship with the outcome variable.

In the example of the multiple regression model that predicts car resale price, you could continue to add more independent variables to the model, such as the number of letters in the name of the person selling the car and the favorite food of the person who bought the car (if you had this data, of course). These predictors are very unlikely to have a relationship with the resale price of the car, but if you add them to your multiple regression model, the R-squared value would still increase. Although this could lead you to think that the model with more predictors is performing better, the inflated R-squared value is a false sign of improvement.

Generally, R-squared will continue to increase with more predictors because the model will become overly specific to the data it was built on even if the predictors do not have a strong relationship with the outcome variable. This is why a high R-squared value is not enough by itself to indicate that the model will perform well and might instead be a sign of overfitting.

When to use adjusted R-squared instead

Along with the R-squared value, a multiple regression model also has an associated **adjusted R-squared value**. The adjusted R-squared penalizes the addition of more independent variables to the multiple regression model. Additionally, the adjusted R-squared only captures the proportion of variation explained by the independent variables that show a significant relationship with the outcome variable. These differences prevent the adjusted R-squared value from becoming inflated like the R-squared value.

When comparing between multiple regression models with varying numbers of predictors, you might find that models with more predictors have a higher R-squared value. This could be a result of overfitting. To avoid selecting an overfitting model with an inflated R-squared, use the adjusted R-squared metric to select the optimal model.

Bias versus variance

A model that underfits the sample data is described as having a high **bias** whereas a model that does not perform well on new data is described as having high **variance**. In data science, there is a phenomenon known as the **bias versus variance tradeoff**. This tradeoff is a dilemma that data scientists face when building any machine learning model because an ideal model should have low bias and low variance. This is another way of saying that it should neither underfit nor overfit. However, as you try to lower bias, variance inevitably increases and vice versa.

This is why you can never fully resolve the problems of underfitting and overfitting. Instead, focus on reducing these problems in your multiple regression model as much as possible.

Key takeaways

Both underfitting and overfitting are obstacles to building a reliable multiple regression model. Although underfitting can be identified by model performance on training data, you must evaluate both training and test performance to identify overfitting. Because overfitting will result in an inflated R-squared value, use the adjusted R-squared value when comparing among multiple regression models with varying numbers of predictors. Although you cannot fully eliminate underfitting and overfitting from the model because of the bias versus variance tradeoff, you can significantly reduce these problems after they have been identified.

Resources for more information

- A detailed description of underfitting and how to mitigate it
- Scikit-learn library documentation for the train_test_split function
- A blog discussing multiple, adjusted, and predicted R-squared values

Chi-squared tests: Goodness of fit versus independence

In the previous course, you learned how hypothesis tests are used to see significant differences among groups. Chi-squared tests are used to determine whether one or more observed categorical variables follow expected distribution(s). For example, you may expect that 50% more movie goers attend movies on weekends in comparison to weekdays. After observing movie goers attendance for a month, you then can perform a chi-squared test to see if your initial hypothesis was correct.

This reading will cover the two main chi-squared tests—goodness of fit and test for independence—which can be used to test your expected hypothesis against what actually occurred. Data professionals perform these hypothesis tests to offer organizations actionable insights that drive decision making.

The Chi-squared goodness of fit test

Chi-squared (χ^2) goodness of fit test is a hypothesis test that determines whether an observed categorical variable with more than two possible levels follows an expected distribution. The null hypothesis (H_0) of the test is that the categorical variable follows the expected distribution. The alternative hypothesis (H_a) is that the categorical variable does not follow the expected distribution. Consider the scenario in this reading that will define the null and alternative hypotheses based on the scenario, set up a Goodness of Fit test, evaluate the test results, and draw a conclusion.

Chi-squared goodness of fit scenario

Imagine that you work as a data professional for an online clothing company. Your boss tells you that they expect the number of website visitors to be the same for each day of the week. You decide to test your boss's hypothesis and pull data every day for the next week and record the number of website visitors in the table below:

Day of the Week	**Observed Values**
Sunday	650
Monday	570
Tuesday	420
Wednesday	480
Thursday	510
Friday	380
Saturday	490
Total	3,500

Here are the main steps you will take:

1. Identify the Null and Alternative Hypotheses
2. Calculate the chi-square test statistic (χ^2)
3. Calculate the p-value
4. Make a conclusion

Step 1: Identify the null and alternative hypotheses

The first step in performing a chi-squared goodness of fit test is to determine your null and alternative hypothesis. Since you are testing if the number of website visitors follows your boss's expectations, the below are your null and alternative hypotheses :

H_0: The week you observed follows your boss's expectations that the number of website visitors is equal on any given day

H_a: The week you observed does not follow your boss's expectations; therefore, the number of website visitors is not equal across the days of the week

Step 2: Calculate the chi-squared test statistic (χ^2)

Next, calculate a test statistic to determine if you should reject or fail to reject your null hypothesis. This test statistic is known as the chi-squared statistic and is calculated based on the following formula:

$$\Sigma \frac{(observed - expected)^2}{expected}$$

The intuition behind this formula is that it should quantify the extent of any discrepancies between observed frequencies and expected frequencies for each categorical level. Squaring these differences does two things. First, it ensures that all discrepancies between observed and expected contribute positively to the chi-squared statistic. Second, it penalizes larger discrepancies. Dividing the sum of the squared differences by the expected frequency of each category level standardizes the differences. In other words, it accounts for the fact that larger discrepancies are more significant when the expected frequencies are small, and less so when the expected frequencies are large.

Returning to the example, since there were a total of 3,500 website visitors you observed; your boss's expectation is that 500 visitors would visit each day (3,500/7). In the formula above, 500 would serve as the "expected" value. A column has been added to your original table to include the test statistic calculation for each weekday:

Day of the Week	**Observed Values**	**Chi-Squared Test Statistic**
Sunday	650	(650−500)2500=45500(650−500)2=45
Monday	570	(570−500)2500=9.8500(570−500)2=9.8
Tuesday	420	(420−500)2500=12.8500(420−500)2=12.8
Wednesday	480	(480−500)2500=0.8500(480−500)2=0.8
Thursday	510	(510−500)2500=0.2500(510−500)2=0.2
Friday	380	(380−500)2500=28.8500(380−500)2=28.8
Saturday	490	(490−500)2500=0.2500(490−500)2=0.2

The X^2 statistic would be the sum of the third column above:

$X^2 = 45 + 9.8 + 12.8 + 0.8 + 0.2 + 28.8 + 0.2$

$X^2 = 97.6$

Note that the X^2 goodness of fit test does not produce reliable results when there are any expected values of less than five.

Step 3: Find the p-value

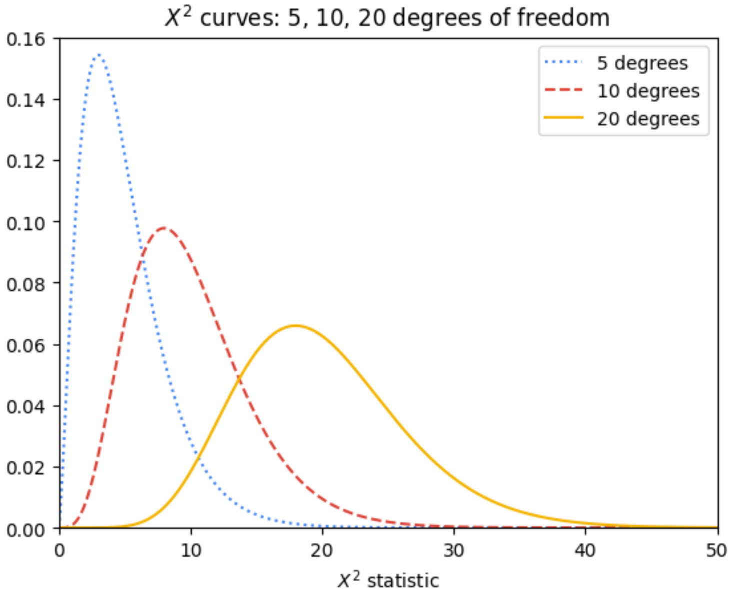
Now that you have calculated the X^2 statistic, consider the following question: What is the probability of obtaining a X^2 statistic of 97.6 or greater when examining 3,500 website visits if they null hypothesis is true? This is the question that the p-value—or “observed significance level"—will answer.

A long time ago, Pearson realized that p-values for X^2 statistics very closely corresponded with areas under certain curves, known as X^2 curves. X^2 curves represent probability density functions, and their shapes vary based on how many degrees of freedom are present in the experiment. Degrees of freedom are determined by the model, not by the data. This means that, in the website traffic example, the degrees of freedom are determined by how many different days a given visit can occur on—not by how many visits are sampled nor by the daily frequencies of the samples themselves. When the model is fully specified (i.e., you know all the possible categorical levels), then:

degrees of freedom = number of categorical levels – 1.

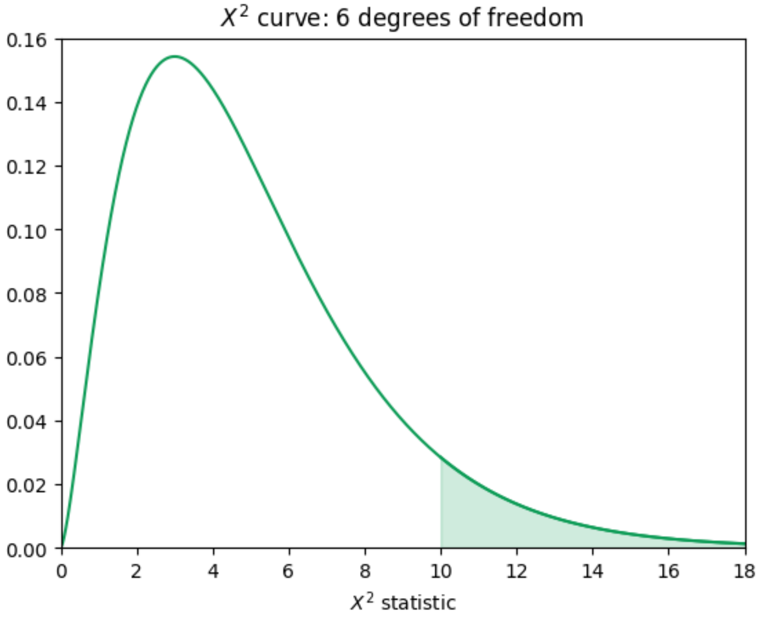
In this example, there are seven categorical levels (one for each day of the week). Therefore, there are six degrees of freedom. This is because the counts of each level (day) are free to fluctuate, but once you know the counts for six days, the seventh day cannot vary. It must result in a total of 3,500 when summed with the other six days.

The following figure depicts the X^2 curves for three different degrees of freedom: five, 10, and 20.



The p-value for a given X^2 test statistic is very closely approximated by the area to its right beneath the X^2 curve of the appropriate degrees of freedom. Notice that the more degrees of freedom there are in the experiment, the greater the area under the right tail of the curve for any given X^2 test statistic, and therefore the greater the probability of getting a given X^2 test statistic if the null hypothesis is true.

The following figure contains the X^2 curve for six degrees of freedom. For a X^2 test statistic of, for instance, 10, the value of P is approximated by the shaded area under the curve where x ≥ 10.



In the case of the website visits, there are six degrees of freedom, but the X^2 test statistic is 97.6—far along the x-axis in the right-skewed tail of the curve. The area under this interval is miniscule: 7.94e-19. In other words, the chances of getting a X^2 test statistic ≥ 97.6 from 3,500 website visits if the null hypothesis were true are 7.94e-17%—practically zero.

Step 4: Make a conclusion

Since the p-value is far less than 0.05, there is sufficient evidence to suggest that the number of visitors is not equal per day.

Coding

Thankfully, you don't need to manually calculate your X^2 test statistic or determine P by hand. You can use the chisquare() function from Python's **`scipy.stats`** package to do this. The following code uses your observed and expected values to calculate the chi-squared test statistic and the p-value. Note that the degrees of freedom are set to the number of observed frequencies minus one. This can be adjusted using the **`ddof`** parameter, but note that this parameter represents **`k - 1 - ddof`** degrees of freedom, where **`k`** is the number of observed frequencies. So, by default, **`ddof=0`** when you call the function, and setting **`ddof=1`** means that your degrees of freedom are reduced by two.

```
import scipy.stats as stats

observations = [650, 570, 420, 480, 510, 380, 490]

expectations = [500, 500, 500, 500, 500, 500, 500]

result = stats.chisquare(f_obs=observations, f_exp=expectations)

result
```

The output confirms your calculation of the chi-square test statistic in Step 2 and also gives you the associated p-value. Because the p-value is less than the significance level of 5%, you can reject the null hypothesis.

The Chi-squared test for independence

Chi-squared (χ^2) Test for Independence is a hypothesis test that determines whether or not two categorical variables are associated with each other. It is valid when your data comes from a random sample and you want to make an inference about the general population. The null hypothesis (H_0) of the test is that two categorical variables are independent. The alternative hypothesis (H_a) is that two categorical variables are not independent.

Chi-squared test for independence scenario

Now suppose that you have been asked to expand your analysis to look at the relationship between the device that a website user used and their membership status. To do this, you must use the X^2 test for independence. In this example, the X^2 test of independence determines whether the type of device a visitor uses to visit the website (Mac or PC) is dependent on whether he or she has a membership account or browses as a guest (member or guest).

Step 1: Identify the null and alternative hypotheses

Just like the goodness of fit scenario, the first step is to determine your null and alternative hypotheses. You are comparing if the device used to visit your clothing store (Mac or PC) is independent from the visitor's membership status (member or guest). From that information you can determine that your null and alternative hypotheses are as follows:

H_0: The type of device a website visitor uses to visit the website is independent of the visitor's membership status.

H_a: The type of device a website visitor uses to visit the website is not independent of the visitor's membership status.

Step 2: Calculate the chi-squared test statistic (χ^2)

To calculate the X^2 test statistic, arrange the data as a table that contains *m* x *n* values, where *m* and *n* are the number of possible levels contained within each respective categorical variable. The following table below breaks down the website visitors based on the device they used and their membership status.

Observed Values	**Member**	**Guest**	**Total**
Mac	850	450	1,300
PC	1,300	900	2,200
Total	2,150	1,350	3,500

Notice that the table starts with 2 x 2 known values (two levels for each category), from which totals are derived. These totals can be used to calculate the expected values, which are necessary to get the X^2 test statistic.

To calculate the expected values, use the following formula:

$$expected\ value\ =\ \frac{column\ total\ *\ row\ total}{overall\ total}$$

For example, the expected value for a Mac member would be:

$$expected\ value\ =\ \frac{2,150\ *\ 1,300}{3,500}\ = 799$$

The logic of this calculation is as follows: if device and membership status are truly independent, then the rate of Mac users who are members should be the same as the rate of Mac users who are guests. The percentage of users who use Macs out of *all* the users is 1,300 / 3,500 = 0.371 * 100 = 37.1%. Accordingly, 37.1% of members and 37.1% of guests would be expected to use Macs. So, 0.371 * 2,150 members ≈ 799.

The following table contains all the expected values:

Expected Values	**Member**	**Guest**
Mac	799	501
PC	1,351	849

Step 3: Find the p-value

Finding the p-value associated with a particular X^2 test statistic is similar to the process outlined already for the goodness of fit test. The only minor difference is how to determine the number of degrees of freedom. For an independence test with two categorical variables with *m* x *n* possible levels, there are (*m* – 1) (*n* – 1) degrees of freedom, assuming there are no other constraints on the probabilities. So, in the working example, this means there is (2 – 1)(2 – 1) = 1 degree of freedom. The p-value in this example is 0.00022. It was determined using Python.

Step 4: Make a conclusion

Because the p-value is 0.00022, you can reject the null hypothesis in favor of the alternative. You conclude that the type of device a website user uses is not independent of his or her membership status. You may recommend to your boss to dive into the reasons behind why visitors sign up for paid memberships more on a particular device. Perhaps the sign-up button appears differently on a particular device. Or maybe there are device-specific bugs that need to be fixed. These are just a couple examples of things you might consider for further exploration.

Coding

You can use the **`scipy.stats`** package's chi2_contingency() function to obtain the X^2 test statistic and p-value of a X^2 independence test. The **`chi2_contingency()`** function only needs the observed values; it will calculate the expected values for you. Here is the Python code:

```
import numpy as np

import scipy.stats as stats

observations = np.array([[850, 450],[1300, 900]])

result = stats.contingency.chi2_contingency(observations, correction=False)

result
```

The output above is in the following order: the X^2 statistic, p-value, degrees of freedom, and expected values in array format. One thing to note is that when degrees of freedom = 1 (i.e., you have a 2 x 2 table), the default behavior of the **`stats.chi2_contingency()`** function is to apply Yates' correction for continuity. This is to make it less likely that small discrepancies will result in significant X^2 values. It is designed to be used when it's possible for an expected frequency in the table to be small (generally < 5). In the given example, it is known that the expected values are all well over five. Therefore, the **`correction`** parameter was set to False.

Key takeaways

- The X^2 goodness of fit test is used to test if an observed categorical variable follows a particular expected distribution.
- The X^2 test for independence is used to test if two categorical variables are independent of each other or not (when samples are drawn at random and you want to make an inference about the whole population).
- Both X^2 tests follow the same hypothesis testing steps to determine whether you should reject or fail to reject the null hypothesis to drive decision making, as you have explored elsewhere in this program.

More about ANOVA

You've learned that analysis of variance—ANOVA—is a group of statistical techniques that test the difference of means between groups. ANOVA testing is useful when you want to test a hypothesis about group differences based on categorical independent variables. For example, if you wanted to determine whether changes in people's weight when following different diets are statistically significant or due to chance, you could use ANOVA to analyze the results. Data professionals routinely must ascertain if there are meaningful differences between groups in their data. This reading will examine more closely the intuition behind ANOVA using a worked example. Later in the program, you will learn how to implement ANOVA in Python.

An overview of ANOVA

The intuition behind ANOVA is to compare the variability between different groups with the variability within the groups. If they are comparable, then the differences between groups are more likely to be due to sampling variability. On the other hand, if the variability between groups is much larger than the variability expected from the samples within their respective groups, then those groups are probably drawn from significantly different subpopulations.

The variation between groups and within groups is calculated as sums of squares, which are then expressed as a ratio. This ratio is known as the F-statistic. The formula for each component of these calculations is presented in the worked example that follows.

Previously, you learned about one-way and two-way ANOVA. To review:

- **One-way ANOVA:** Compares the means of one continuous dependent variable based on three or more groups of **one** categorical variable
- **Two-way ANOVA:** Compares the means of one continuous dependent variable based on three or more groups of **two** categorical variables

To help you understand the intuition behind ANOVA, this reading will break down a worked example of a simple one-way ANOVA test.

One-way ANOVA

5 steps

There are five steps in performing a one-way ANOVA test:

1. Calculate group means and grand (overall) mean
2. Calculate the sum of squares between groups (SSB) and the sum of squares within groups (SSW)
3. Calculate mean squares for both SSB and SSW
4. Compute the F-statistic
5. Use the F-distribution and the F-statistic to get a p-value, which you use to decide whether to reject the null hypothesis

Example

Return to the example of students studying for an exam. Suppose that in this case you wanted to compare three different studying programs, A, B, and C to determine whether they have an effect on exam score. Here is the data:

Student	**Study program (X)**	**Exam score (Y)**
1	A	88
2	A	79
3	A	86
4	A	90
5	B	94
6	B	84
7	B	87
8	B	89
9	C	85
10	C	76
11	C	81
12	C	78

First, state your hypotheses:

H_0: $\mu_A = \mu_B = \mu_C$

The mean score of group A = the mean score of group B = the mean score of group C.

H_1: NOT ($\mu_A = \mu_B = \mu_C$)

The means of each group are not all equal. Remember, even if only one mean differs, that is sufficient evidence to reject the null hypothesis.
Next, determine your confidence level—the threshold above which you will reject the null hypothesis. This value is dependent on your situation and usually requires some domain knowledge. A common threshold is 95%.
Now, begin the steps of ANOVA.

Step 1

Calculate group means and grand mean. The grand mean is the overall mean of all samples in all groups.

The following table restructures the data in the previous table such that the scores for each study group are contained in their own column. Additionally, the mean score of each group has been calculated.

Study program A scores	**Study program B scores**	**Study program C scores**
88	94	85
79	84	76
86	87	81
90	89	78
Mean: 85.75	**Mean: 88.5**	**Mean: 80**

Grand mean (M_G) = 84.75

Step 2

A. Calculate the sum of squares between groups (SSB).

$$\mathrm{SSB} = \sum_{g=1} n_g (M_g - M_G)^2$$

where:

n_g = the number of samples in the g^{th} group

M_g = mean of the g^{th} group

M_G = grand mean

→ **SSB** = [4(85.75 – 84.75)^2 + 4(88.5 – 84.75)^2 + 4(80 – 84.75)^2]

= 4 + 56.25 + 90.25

= 150.5

B. Calculate the sum of squares within groups (SSW).

$$\mathrm{SSW} = \sum_{g=1} \sum_{i=1} (x_{gi} - M_g)^2$$

where:

x_{gi} = sample *i* of the g^{th} group

M_g = mean of the g^{th} group

The double summation acts like a nested loop. The outer loop is for each group and the inner loop is for all the samples in that group. So, for each sample in group 1, subtract from it the group's mean and square the result. Then, do the same thing with the samples in group 2, using the group 2 mean. Continue this way for all the groups and sum all the results.

The following table shows the squared difference between each observation and its group mean. It also contains the sums of these squared differences for each of the three study groups, A, B, and C.

Program A	**Program B**	**Program C**	$(x_{Ai}-M_A)2$	$(x_{Bi}-M_B)2$	$(x_{Ci}-M_C)2$
88	94	85	5.06	30.25	25
79	84	76	45.56	20.25	16
86	87	81	0.06	2.25	1
90	89	78	18.06	0.25	4
M_A = 85.75	M_B = 88.5	M_C = 80	**Sum: 68.75**	**Sum: 53**	**Sum: 46**

→ **SSW** = 68.75 + 53 + 46

= 167.75

Step 3

Calculate mean squares between groups and within groups. The mean square is the sum of squares divided by the degrees of freedom, respectively.

A. Mean squares between groups (MSSB):

$$\mathrm{MSSB} = \frac{\mathrm{SSB}}{k-1}$$

where:

k = the number of groups

Note: $k-1$ represents the degrees of freedom between groups

→ **MSSB** = 150.53−13−1150.5

= 75.25

B. Mean squares within groups (MSSW):

$$\mathrm{MSSW} = \frac{\mathrm{SSW}}{n-k}$$

where:

n = the total number of samples in all groups

k = the number of groups

Note: $n-k$ represents the degrees of freedom within groups

→ **MSSW** = 167.7512−312−3167.75

= 18.64

Step 4

Compute the F-statistic.

The F-statistic is the ratio of the mean sum of squares between groups (MSSB) to the mean sum of squares within groups (MSSW):

$$\text{F-statistic} = \frac{\text{MSSB}}{\text{MSSW}}$$

→ **F-statistic** = 75.25/18.64

= 4.04

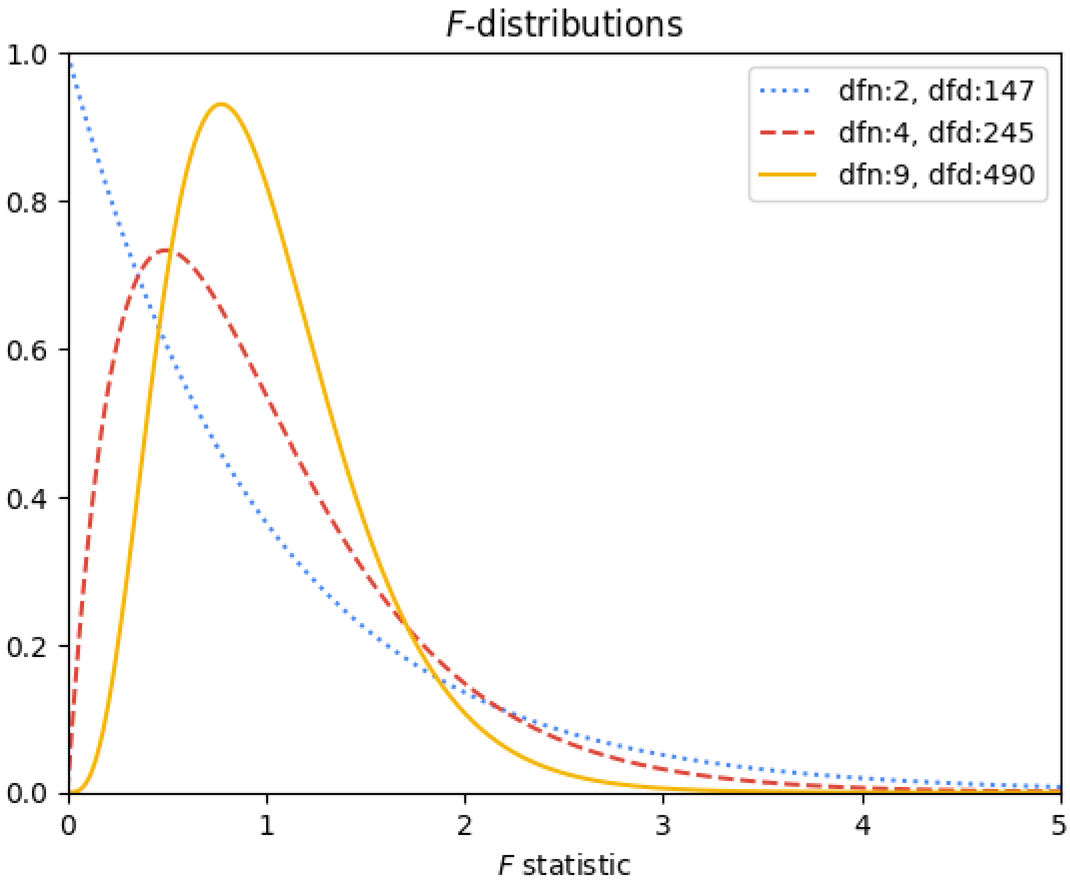
A higher F-statistic indicates a greater variability between group means relative to the variability within groups, suggesting that at least one group mean is significantly different from the others.

Step 5

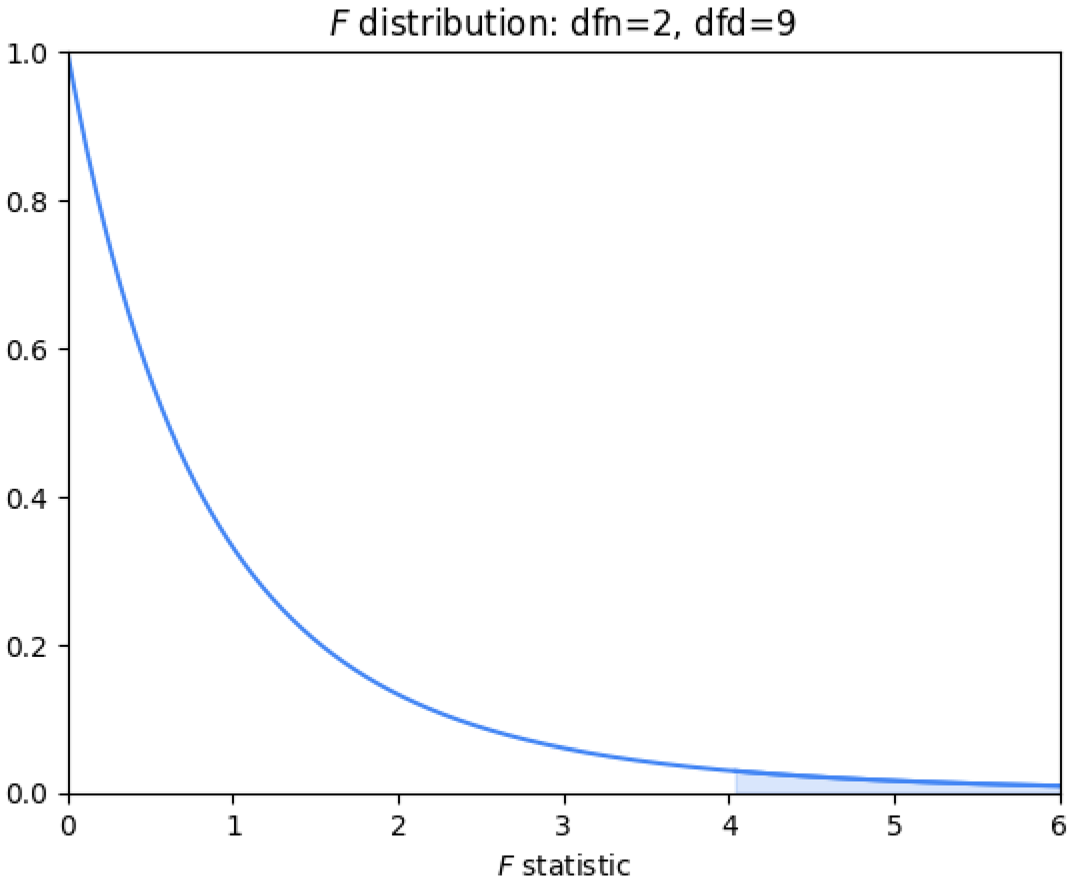
Use the F-distribution and the F-statistic to get a p-value, which you use to decide whether to reject the null hypothesis.

Similar to t-tests and X^2 tests, ANOVA testing finds the area under a particular probability distribution curve—the F-distribution—of the null hypothesis to determine a p-value. The larger the F-statistic, the lesser the area beneath the curve and the more evidence against the null hypothesis, thus resulting in a lower p-value.

The shape of the F-distribution curve is determined by the degrees of freedom between and within groups. Here is a graph depicting F-distributions for three, five, and 10 groups—each group containing 50 samples. Note that “dfn” represents the degrees of freedom in the numerator (between groups) and “dfd” represents the degrees of freedom in the denominator (within groups). Notice how the degrees of freedom affect the shape of the curve.



Similar to X^2 curves, F-distributions help determine the probability of falsely rejecting the null hypothesis. In the case of ANOVA, this probability is represented by the area of the F-distribution beneath the curve where x ≥ your F-statistic. For example, the following graph depicts the F-distribution for the exam scores example. It has two degrees of freedom in the numerator and nine degrees of freedom in the denominator. The area beneath the curve where x ≥ 4.04 (the computed F-statistic) is shaded.



You can use statistical software to calculate this area. You will learn how to do this in a later activity. In this case, the area beneath the F-distribution to the right of 4.04 is 0.05604. This is the probability of observing an F-statistic greater than 4.04 if the null hypothesis were true. Whether this is sufficient to reject the null hypothesis is a decision you make at the beginning of your hypothesis test. For example, if you decided that you wanted a confidence level of 95% or greater, you cannot reject the null hypothesis that the means of the distributions for each program of study are all the same, because the p-value is 0.056.

Assumptions of ANOVA

ANOVA will only work if the following assumptions are true:

1. The dependent values for each group come from normal distributions
    - Note that this assumption does NOT mean that *all* of the dependent values, taken together, must be normally distributed. Instead, it means that *within each group*, the dependent values should be normally distributed.
    - ANOVA is generally robust to violations of normality, especially when sample sizes are large or similar across groups, due to the central limit theorem. However, significant violations can lead to incorrect conclusions.
2. The variances across groups are equal
    - ANOVA compares means across groups and assumes that the variance around these means is the same for all groups. If the variances are unequal (i.e., heteroscedastic), it could lead to incorrect conclusions
3. Observations are independent of each other

    - ANOVA assumes that one observation does not influence or predict any other observation. If there is autocorrelation among the observations, the results of the ANOVA test could be biased.

Key takeaways

- ANOVA tests are statistical tests that examine whether or not the means of a continuous dependent variable are significantly different from one another based on the different levels of one or more independent categorical variables.
- It is sufficient for one group's mean to be significantly different from the others to reject the null hypothesis; however, ANOVA testing is limited in that it doesn't tell you *which* group is different. To make such a determination, other tests are necessary.
- ANOVA works by comparing the variance between each group to the variance within each group. The greater the ratio of variance between groups to variance within groups, the greater the likelihood of rejecting the null hypothesis.
- ANOVA depends on certain assumptions, so it is important to check that your data meets them in order to avoid drawing false conclusions. At the very least, if your data does not meet all of them, identify these violations.

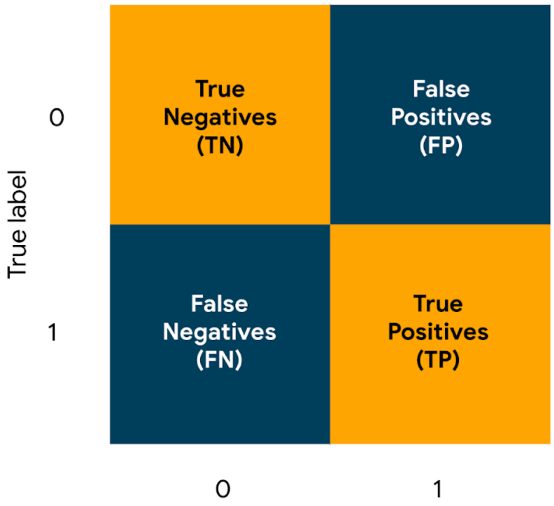
Common logistic regression metrics in Python

Logistic regression is a powerful technique for categorical prediction tasks in data science. Data professionals often use metrics such as precision, recall, and accuracy, as well as visualizations such as ROC curves, to gauge the performance of their logistic regression models. It is important to evaluate the performance of a model, as this shows how well the model can make predictions. The results from applying metrics can be used to report how well a model performs to relevant stakeholders.

In this reading, you will review parts of a confusion matrix and understand how to compute and visualize metrics for evaluating logistic regression through code in Python.

Parts of a confusion matrix

A confusion matrix helps summarize the performance of a classifier. The components of a confusion matrix are used to compute metrics for evaluating logistic regression classifiers.



The four key parts of a confusion matrix, in the context of binary classification, are the following:

1. **True negatives:**

The count of observations that a classifier correctly predicted as False (0)

2. **True positives:**

The count of observations that a classifier correctly predicted as True (1)

3. **False positives:**

The count of observations that a classifier incorrectly predicted as True (1)

4. **False negatives:**

The count of observations that a classifier incorrectly predicted as False (0)

These counts are useful in computing metrics such as precision, recall, accuracy, and ROC for evaluating logistic regression classifiers.

Precision

One of the major metrics for evaluating a logistic regression classifier is **precision**. Precision measures the proportion of data points predicted as True that are actually True. Imagine that you have built a logistic regression classifier for email spam detection, trained this classifier on a relevant dataset, and used this classifier to generate predictions for a set of emails. The predictions consist of True and False values. True represents an email predicted as spam, and False represents an email predicted as not spam. The precision for this classifier would convey the proportion of emails that are actually spam, out of all the emails that have been predicted as spam.

The formula for precision is as follows:

$$Precision = \frac{True\ Positives}{True\ Positives\ +\ False\ Positives}$$

To compute precision in Python, you can use the **`precision_score()`** function from the **`metrics`** module in the **`sklearn`** library. You can start with the following import statement.

```
import sklearn.metrics as metrics
```

The **`precision_score()`** function takes in true values and predicted values as arguments and returns the precision score. Assume that **`y_test`** and **`y_pred`** are variables that contain true values and predicted values respectively. You can use the following code to compute the precision.

```
metrics.precision_score(y_test,y_pred)
```

In the context of email spam detection, if precision_score() returns 0.91, that means 91% of the emails predicted as spam are indeed spam.

Recall

Another major metric for evaluating a logistic regression classifier is **recall**. Recall measures the proportion of data points that are predicted as True, out of all the data points that are actually True. Imagine that you have built a logistic regression classifier for fraud detection and generated predictions. In the predictions, True represents a credit card transaction predicted as fraudulent, and False represents a credit card transaction predicted as not fraudulent. The recall for this classifier would convey the proportion of fraudulent credit card transactions that the classifier correctly identified as such.

The formula for recall is as follows:

$$Recall = \frac{True\ Positives}{True\ Positives\ +\ False\ Negatives}$$

To compute recall in Python, you can use the **`recall_score()`** function from the metrics module. The function takes in true values and predicted values as arguments and returns the recall score. You can use the following code to compute the recall.

```
metrics.recall_score(y_test,y_pred)
```

In the context of fraud detection among credit card transactions, if the **`recall_score()`** function returns 0.87, that means 87% of the fraudulent credit card transactions are correctly detected as fraudulent.

Accuracy

Another important metric for evaluating logistic regression is **accuracy**. Accuracy measures the proportion of data points that are correctly classified. Imagine that you have built a logistic regression classifier for loan approval prediction. In the predictions, True represents a prediction that the loan will be approved, and False represents a prediction that the loan will not be approved. The accuracy score for this classifier would convey the proportion of loans that have been correctly classified.

The formula for accuracy is as follows:

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Total\ Predictions}$$

To compute accuracy in Python, you can use the **`accuracy_score()`** function from the metrics module. The function takes in true values and predicted values as arguments and returns the accuracy score. You can use the following code to compute the accuracy.

```
metrics.accuracy_score(y_test,y_pred)
```

In the context of loan approval prediction, if the **`accuracy_score()`** function returns 0.90, that means 90% of the loans are correctly predicted, with the predictions being either will be approved or will not be approved.

ROC curves

An **ROC curve** helps in visualizing the performance of a logistic regression classifier. ROC curve stands for receiver operating characteristic curve. To visualize the performance of a classifier at different classification thresholds, you can graph an ROC curve. In the context of binary classification, a classification threshold is a cutoff for differentiating the positive class from the negative class.

An ROC curve plots two key concepts

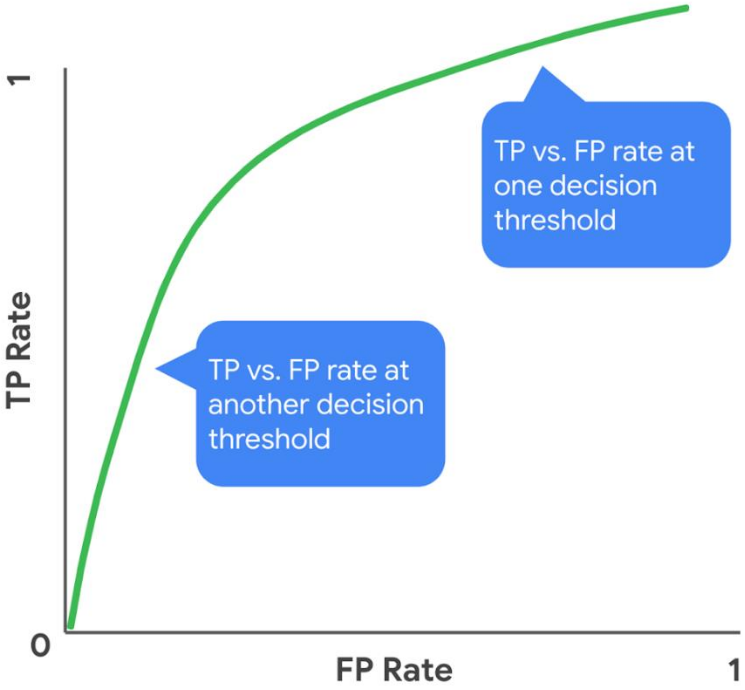
1. **True Positive Rate:** equivalent to **Recall.** The formula for True Positive Rate is as follows:

$$True\ Positive\ Rate = \frac{True\ Positives}{True\ Positives\ +\ False\ Negatives}$$

2. **False Positive Rate:** The ratio between the False Positives and the total count of observations that should be predicted as False. The formula for False Positive Rate is as follows:

$$False\ Positive\ Rate = \frac{False\ Positives}{False\ Positives\ +\ True\ Negatives}$$

For each point on the curve, the x and y coordinates represent the False Positive Rate and the True Positive Rate respectively at the corresponding threshold.



An example of an ROC curve. For each point on the curve, the x and y coordinates represent the False Positive Rate and the True Positive Rate respectively at the corresponding threshold.

You can examine an ROC curve to observe how the False Positive Rate and True Positive Rate change together over the different thresholds. In the ROC curve for an ideal model, there would exist a threshold at which the True Positive Rate is high and the False Positive Rate is low. The more that the ROC curve hugs the top left corner of the plot, the better the model does at classifying the data.

You can use the following steps to graph an ROC curve in Python.

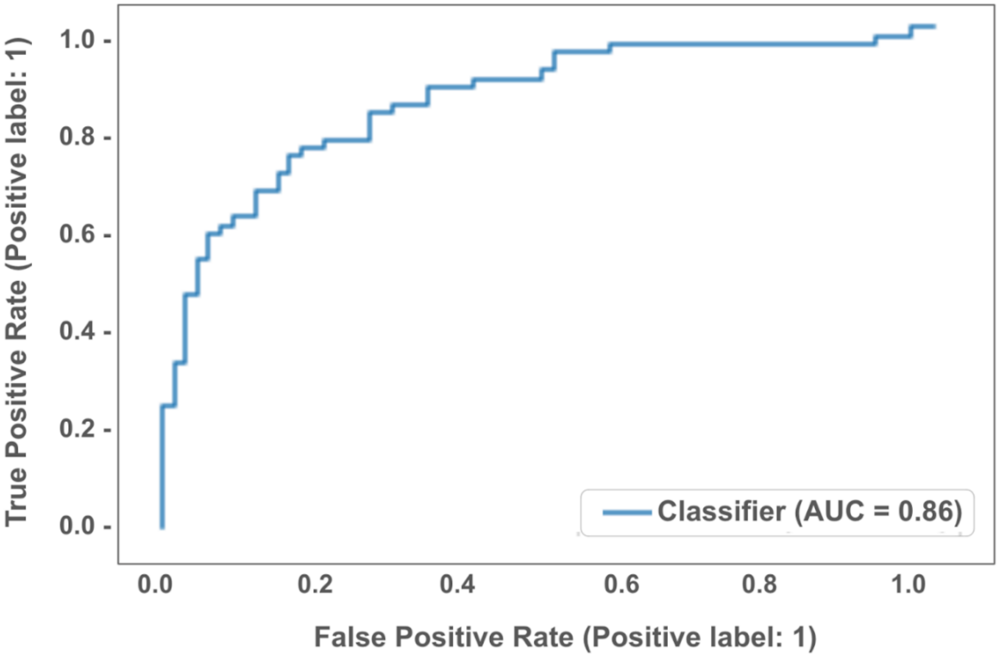
Start by importing the necessary modules as follows.

```
import matplotlib.pyplot as plt
from sklearn.metrics import RocCurveDisplay
```

Then use the following code to plot the ROC curve.

```
RocCurveDisplay.from_predictions(y_test, y_pred)
plt.show()
```

Using these steps to generate an ROC curve could result in a graph.

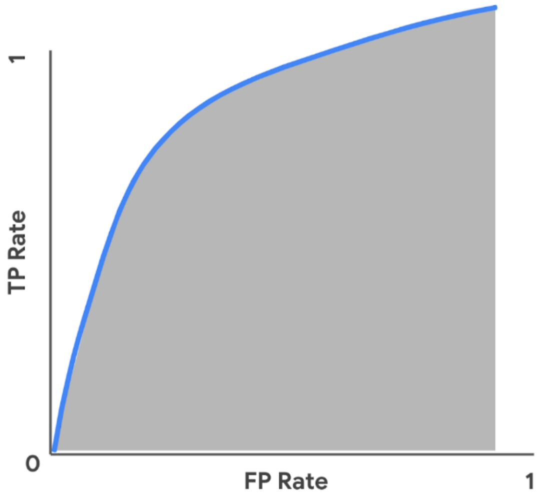


In this graph, the ROC curve indicates that the corresponding classifier performs decently well.

AUC

AUC stands for area under the ROC curve. AUC provides an aggregate measure of performance across all possible classification thresholds. AUC ranges in value from 0.0 to 1.0. A model whose predictions are 100% wrong has an AUC of 0.0, and a model whose predictions are 100% correct has an AUC of 1.0. An AUC smaller than 0.5 indicates that the model performs worse than a random classifier (i.e. a classifier that randomly assigns each example to True or False), and an AUC larger than 0.5 indicates that the model performs better than a random classifier.

In the following visualization, AUC is the area of the shaded region.



To compute AUC in Python, you can use the roc_auc_score() function from the metrics module. The function takes in true values and predicted values as arguments and returns the accuracy score. You can use the following code to compute the AUC.

```
metrics.roc_auc_score(y_test,y_pred)
```

For example, in the context of a logistic regression classifier for email spam detection, if the **`roc_auc_score()`** function returns 0.99, that means 99% of the classifier's predictions are correct across all classification thresholds.

Key takeaways

- Precision, Recall, and Accuracy are common metrics for evaluating the performance of a logistic regression classifier.
- You can use functions from the metrics module in the sklearn library to compute these metrics in Python.
- Graphing an ROC curve helps visualize how a classifier performs across different classification thresholds.
- Computing AUC helps aggregate a classifier's performance across thresholds into one measure.

Resources for more information

- precision_score: Documentation on the precision_score() function
- recall_score: Documentation on the recall_score() function
- accuracy_score: Documentation on the accuracy_score() function
- RocCurveDisplay.from_predictions: Documentation on the RocCurveDisplay.from_predictions() function
- roc_auc_score: Documentation on the roc_auc_score() function

Interpret logistic regression models

Interpreting a logistic regression model involves examining coefficients and computing metrics. After you fit your logistic regression model to training data, you can access the coefficient estimates from the model using code in Python. You can then use those values to understand how the model makes predictions. This reading will show you an example of how to interpret coefficients from a logistic regression model, as well as things to consider when choosing metrics for model evaluation.

Coefficients from the model

To understand how a logistic regression model works, it is important to start with the equation that describes the relationship between the variables. That equation is also called the logit function.

The logit function

When the logit function is written in terms of the independent variables, it conveys the following: there is a linear relationship between each independent variable, X, and the logit of the probability that the dependent variable, Y, equals 1. The logit of that probability is the logarithm of the odds of that probability.

The equation for the logit function in binomial logistic regression is shown below. This involves the probability that Y equals 1, because 1 is the typical outcome of interest in binary classification, where the possible values of Y are 1 and 0.

$$logit(p) = log(\frac{p}{1 \ - p}) = \beta_0 + \beta_1 X_1 \quad where\, p = P(Y = 1)$$

Interpret coefficients

Imagine you have built a binomial logistic regression model for predicting emails as spam or non-spam. The dependent variable, Y, is whether an email is spam (1) or non-spam (0). The independent variable, X_1, is the message length. Assume that **`clf`** is the classifier you fitted to training data.

You can use the following code to access the coefficient β_1 estimated by the model:

```
clf.coef_
```

If the estimated β_1 is 0.186, for example, that means a one-unit increase in message length is associated with a 0.186 increase in the log odds of p. To interpret change in odds of Y as a percentage, you can exponentiate β_1, as follows.

$$e^{\beta_1} = e^{0.186} \approx 1.204$$

So, for every one-unit increase in message length, you can expect that the odds the email is spam increases by 1.204, or 20.4%.

Things to consider when choosing metrics

The next important step after examining the coefficients from a logistic regression model is evaluating the model through metrics. The most commonly used metrics include precision, recall, and accuracy. The following sections describe things to keep in mind when choosing between these.

When to use precision

Using precision as an evaluation metric is especially helpful in contexts where the cost of a false positive is quite high and much higher than the cost of a false negative. For example, in the context of email spam detection, a false positive (predicting a non-spam email as spam) would be more costly than a false negative (predicting a spam email as non-spam). A non-spam email that is misclassified could contain important information, such as project status updates from a vendor to a client or assignment deadline announcements from an instructor to a class of students.

When to use recall

Using recall as an evaluation metric is especially helpful in contexts where the cost of a false negative is quite high and much higher than the cost of a false positive. For example, in the context of fraud detection among credit card transactions, a false negative (predicting a fraudulent credit card charge as non-fraudulent) would be more costly than a false positive (predicting a non-fraudulent credit card charge as fraudulent). A fraudulent credit card charge that is misclassified could lead to the customer losing money, undetected.

When to use accuracy

It is helpful to use accuracy as an evaluation metric when you specifically want to know how much of the data at hand has been correctly categorized by the classifier. Another scenario to consider: accuracy is an appropriate metric to use when the data is balanced, in other words, when the data has a roughly equal number of positive examples and negative examples. Otherwise, accuracy can be biased. For example, imagine that 95% of a dataset contains positive examples, and the remaining 5% contains negative examples. Then you train a logistic regression classifier on this data and use this classifier predict on this data. If you get an accuracy of 95%, that does not necessarily indicate that this classifier is effective. Since there is a much larger proportion of positive examples than negative examples, the classifier may be biased towards the majority class (positive) and thus the accuracy metric in this context may not be meaningful. When the data you are working with is imbalanced, consider either transforming it to be balanced or using a different evaluation metric other than accuracy.

Key takeaways

- Examine the beta coefficients from a model to understand how the model predicts the dependent variable.
- When determining which metrics are meaningful for evaluating a logistic regression classifier, consider the context of the data involved, how the predictions will be used, and how impactful False Positives versus False Negatives are in that context.

Resources for more information

- LogisticRegression: Documentation for implementing Logistic Regression models using **`sklearn`** and accessing intercept and coefficients from a model

Prediction with different types of regression

As you have been learning, key regression techniques that you will encounter in your work as a data professional include linear regression, hypothesis testing, and logistic regression. When your goal is to make predictions with data, it is important to consider these different approaches and think about which approach will best help you achieve your task. In this reading, you will learn more about how to choose the most relevant regression technique for a project, based on the question you want to answer, the outcome variable, and how it is measured.

How to choose a regression technique

When choosing a regression technique, it is important to consider the data you are working with and the question you want to address.

Things to consider

1. What is the question you want to answer? In other words, what do you want to predict?
2. Which variable in your data can be the outcome variable?
3. How is the outcome variable measured? If the outcome variable is continuous, it is more likely that either linear regression or hypothesis testing will be most appropriate. However, if the outcome variable is binary, you will find logistic regression to be more useful.

Example contexts for regression

The following examples demonstrate how the questions about prediction, outcome variable, and measurement can be navigated in order to choose a regression technique.

Example context: User engagement

In your work as a data professional, imagine that you are interested in making predictions about user engagement for a mobile app.

First, you might ask, what is the question you want to answer?

One possible question could be “How much does each in-app feature influence user engagement?” The in-app features might include a live chat with customer support, an FAQ section that updates weekly, and a community space to connect with other users. Next, you might ask, which variable in your data can be the outcome variable? If you have access to data about users’ session lengths (in other words, how long users spend in the app each time they open it), the outcome variable can be session length. Your next question might be: how is the outcome variable measured? Session length can be measured by number of minutes, which is continuous. Because the outcome variable is continuous, and you are interested in how much each feature influences the outcome variable, you could proceed with linear regression and check the relevant model assumptions. If there is only one feature of interest, you would build a simple linear regression model. If there are multiple features of interest, you would build a multiple linear regression model.

Another question of interest could be “Does a dynamic landing page versus a static landing page make a difference in user engagement?” The outcome variable can be session length, measured by number of minutes, for this example, too. Since the outcome variable is continuous, and the target question is about whether there is a difference in user engagement when one type of landing page is used over the other, you could proceed with hypothesis testing. You can then frame the hypotheses, which could be the following:

- Null hypothesis (H_0): Users spend approximately the same amount of time in the app when the landing page is dynamic versus when it is static.
- Alternative hypothesis (H_1): Users do NOT spend approximately the same amount of time in the app when the landing page is dynamic versus when it is static.

Another question you might be interested in is “Will a user engage with the new line of products in-app?” Next, you might ask, which variable in your data can be the outcome variable? If you have access to data about whether a user clicks to view the new line of products, that could be the outcome variable. The next question is: how is the outcome variable measured? Whether a user clicks to view that content can be represented as a binary variable, with 1 indicating they clicked to view the content and 0 indicating that they did not click to view that content. Since this outcome variable is binary, you could proceed with binomial logistic regression.

Example context: Patient response

Now imagine that you are tasked with making predictions about patient responses to medical treatments.

You can start by asking, what is the question you want to answer?

A possible question could be “How much does each factor influence a patient’s response to a medical treatment?” If the goal of the treatment is to improve white blood cell (WBC) count and you have access to that data, WBC count can be the outcome variable. The outcome variable is a continuous measure, and you could use linear regression to address this task.

Another question of interest could be “Will Treatment A, Treatment B, or Treatment C have a stronger impact on a patient’s WBC count?” The outcome variable in this case would also be WBC count, which is continuous. Since the target question is about comparing different treatments, it would be best to proceed with hypothesis testing. You can then form the hypotheses, which could be the following.

- Null hypothesis (H_0): Patients have approximately the same white blood cell count with each treatment.
- Alternative hypothesis (H_1): Patients do NOT have approximately the same white blood cell count with each treatment.

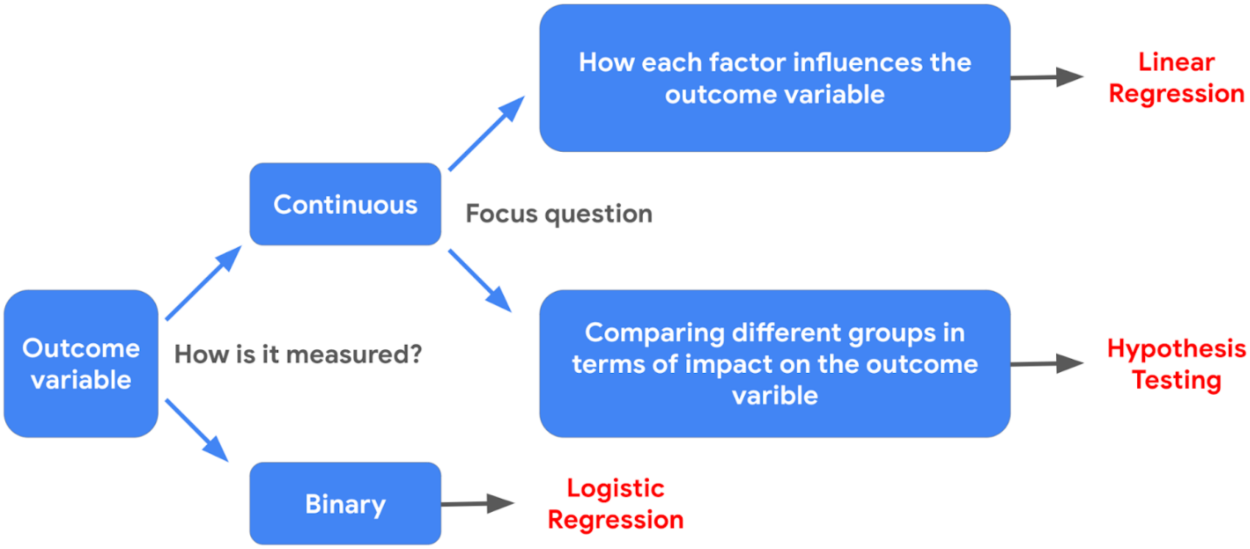
A different question you might be interested in: “With Treatment A, will a patient’s WBC count reach the ideal range?” If you have access to the associated data, the outcome variable would be whether a patient’s WBC count reaches the ideal range or not, which is a binary variable: 1

indicating that their WBC count falls within the ideal range and 0 indicating that it does not. You could build a logistic regression model to make predictions in this scenario.

Key takeaways

- Consider the question you want to answer and the data you have access to when choosing a regression technique for making predictions.
- Identifying the outcome variable of interest and how it is measured will help you decide which regression technique is most suitable for your task.

The following flowchart captures a high-level approach for choosing a regression technique, starting from the outcome variable, as discussed in this reading. Also note that hypothesis testing is connected to regression analysis. For example, in linear regression, the process of testing whether there is a correlation between two variables (in other words, determining if the coefficients are statistically significant in the linear model) involves a hypothesis test.



Resources for more information

- If you want to learn more about different types of regression models, you can check out this article about different types of regression models, covering linear regression, logistic regression, and more.
- If you want to learn more about hypothesis testing, you can check out this article that provides an overview of the key steps for approaching hypothesis testing in data science.

Terms and definitions from Course 5

A

Absolute values: (Refer to **observed values**)

Accuracy: The proportion of data points that were correctly categorized

Adjusted R^2: A variation of R^2 that accounts for having multiple independent variables present in a linear regression model

Analysis of Variance (ANOVA): A group of statistical techniques that test the difference of means between three or more groups

ANCOVA (Analysis of Covariance): A statistical technique that tests the difference of means between three or more groups while controlling for the effects of covariates, or variable(s) irrelevant to the test

B

Backward elimination: A stepwise variable selection process that begins with the full model, with all possible independent variables, and removes the independent variable that adds the least explanatory power to the model

Best fit line: The line that fits the data best by minimizing some loss function or error

Bias: Simplifying the model predictions by making assumptions about the variable relationships

Bias-variance trade-off: Balance between two model qualities, bias and variance, to minimize overall error for unobserved data

Binomial logistic regression: A technique that models the probability of an observation falling into one of two categories, based on one or more independent variables

Binomial logistic regression linearity assumption: An assumption stating that there should be a linear relationship between each X variable and the logit of the probability that Y equals one

C

Causation: Describes a cause-and-effect relationship where one variable directly causes the other to change in a particular way

Chi-squared (χ^2) Goodness of Fit Test: A hypothesis test that determines whether an observed categorical variable follows an expected distribution

Chi-squared (χ^2) Test for Independence: A hypothesis test that determines whether or not two categorical variables are associated with each other

Confidence band: The area surrounding a line that describes the uncertainty around the predicted outcome at every value of X

Confusion matrix: A graphical representation of how accurate a classifier is at predicting the labels for a categorical variable

Confidence interval: A range of values that describes the uncertainty surrounding an estimate

Correlation: Measures the way two variables tend to change together

D

Dependent variable (Y): The variable a given model estimates

E

Errors: In a regression model, the natural noise assumed to be in a model

Explanatory variable: (Refer to **independent variable**)

Extra Sum of Squares F-test: A test that quantifies the difference between the amount of variance that is left unexplained by a reduced model that is explained by the full model

F

Feature selection: (Refer to **variable selection**)

Forward selection: A stepwise variable selection process that begins with the null mode—with 0 independent variables—considers all possible variables to add; incorporates the independent variable that contributes the most explanatory power to the model

H

Hold-out sample: A random sample of observed data that is not used to fit the model

Homoscedasticity assumption: An assumption of simple linear regression stating that the variation of the residuals (errors) is constant or similar across the model

Hypothesis testing: A statistical procedure that uses sample data to evaluate an assumption about a population parameter

I

Independent observation assumption: An assumption of simple linear regression stating that each observation in the dataset is independent

Independent variable (X): The variable whose trends are associated with the dependent variable

Interaction term: The term that represents how the relationship between two independent variables is associated with changes in the mean of the dependent variable

Intercept (constant B_0): The y value of the point on the regression line where it intersects with the y-axis

L

Likelihood: The probability of observing the actual data, given some set of beta parameters

Line: A collection of an infinite number of points extending in two opposite directions

Linearity assumption: An assumption of simple linear regression stating that each predictor variable (X_i) is linearly related to the outcome variable (Y)

Linear regression: A technique that estimates the linear relationship between a continuous dependent variable and one or more independent variables

Link function: A nonlinear function that connects or links the dependent variable to the independent variables mathematically

Logistic regression: A technique that models a categorical dependent variable (Y) based on one or more independent variables (X)

Loss function: A function that measures the distance between the observed values and the model's estimated values

Logit: The logarithm of the odds of a given probability

Log-Odds function: (Refer to **logit**)

M

MAE (Mean Absolute Error): The average of the absolute difference between the predicted and actual values

MANCOVA (Multivariate Analysis of Covariance): An extension of ANCOVA and MANOVA that compares how two or more continuous outcome variables vary according to categorical independent variables, while controlling for covariates

MANOVA (Multivariate Analysis of Variance): An extension of ANOVA that compares how two or more continuous outcome variables vary according to categorical independent variables

Maximum Likelihood Estimation (MLE): A technique for estimating the beta parameters that maximize the likelihood of the model producing the observed data

Model assumptions: Statements about the data that must be true in order to justify the use of a particular modeling technique

MSE (Mean Squared Error): The average of the squared difference between the predicted and actual values

Multiple linear regression: A technique that estimates the relationship between one continuous dependent variable and two or more independent variables

Multiple regression: (Refer to **multiple linear regression**)

N

Negative correlation: An inverse relationship between two variables, where when one variable increases, the other variable tends to decrease, and vice versa

Normality assumption: An assumption of simple linear regression stating that the residuals are normally distributed

No multicollinearity assumption: An assumption of simple linear regression stating that no two independent variables (X_i and X_j) can be highly correlated with each other

O

Observed values: The existing sample of data, where each data point in the sample is represented by an observed value of the dependent variable and an observed value of the independent variable

One hot encoding: A data transformation technique that turns one categorical variable into several binary variables

One-Way ANOVA: A type of statistical testing that compares the means of one continuous dependent variable based on three or more groups of one categorical variable

Ordinary least squares estimation (OLS): A common way to calculate linear regression coefficients

Outcome variable (Y): (Refer to **dependent variable**)

Overfitting: When a model fits the observed or training data too specifically and is unable to generate suitable estimates for the general population

P

P-value: The probability of observing results as extreme as those observed when the null hypothesis is true

Positive correlation: A relationship between two variables that tend to increase or decrease together

Post hoc test: An ANOVA test that performs a pairwise comparison between all available groups while controlling for the error rate

Precision: The proportion of positive predictions that were true positives

Predicted values: The estimated Y values for each X calculated by a model

Predictor variable: (Refer to **independent variable**)

R

R^2 (The Coefficient of Determination): The coefficient that measures the proportion of variation in the dependent variable, Y, explained by the independent variable(s), X

Recall: The proportion of positives the model was able to identify correctly

Regression analysis: A group of statistical techniques that use existing data to estimate the relationships between a single dependent variable and one or more independent variables

Regression coefficient: The estimated betas in a regression model

Regression models: (Refer to **regression analysis**)

Regularization: A set of regression techniques that shrinks regression coefficient estimates towards zero, adding in bias, to reduce variance

Residual: The difference between observed or actual values and the predicted values of the regression line

Response variable: (Refer to **dependent variable**)

S

Scatterplot matrix: A series of scatterplots that show the relationships between pairs of variables

Simple linear regression: A technique that estimates the linear relationship between one independent variable, X, and one continuous dependent variable, Y

Slope: The amount that y increases or decreases per one-unit increase of x

Sum of squared residuals (SSR): The sum of the squared difference between each observed value and its associated predicted value

T

Two-Way ANOVA: A type of statistical testing that compares the means of one continuous dependent variable based on three or more groups of two categorical variables

V

Variable selection: The process of determining which variables or features to include in a given model

Variance: Model flexibility and complexity, so the model learns from existing data

Variance inflation factors (VIF): Factors that quantify how correlated each independent variable is with all of the other independent variables