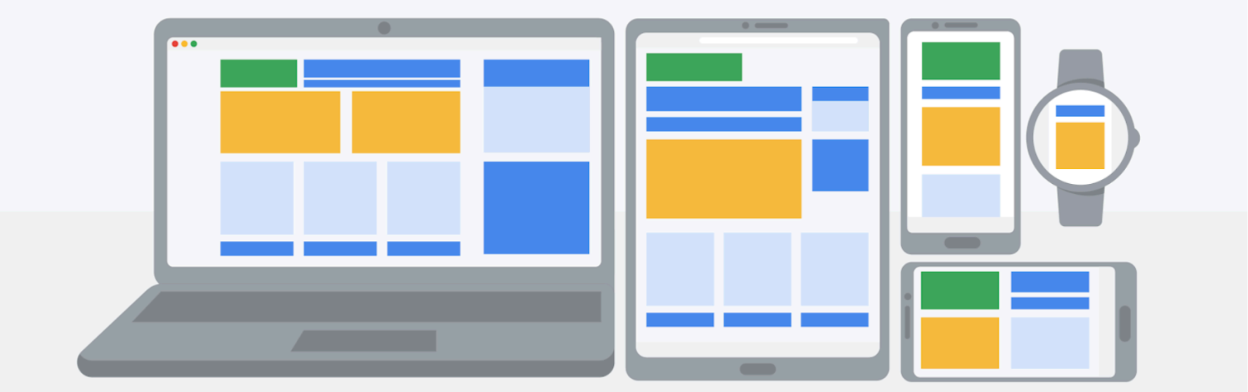
Use Boolean logic

In this reading, you will explore the basics of Boolean logic and learn how to use single and multiple conditions in a Boolean statement. These conditions are created with Boolean operators, including **`AND`**, **`OR`**, and **`NOT`**. These operators are similar to mathematical operators and can be used to create logical statements that filter your results. Data analysts use Boolean statements to do a wide range of data analysis tasks, such as writing queries for searches and checking for conditions when writing programming code.

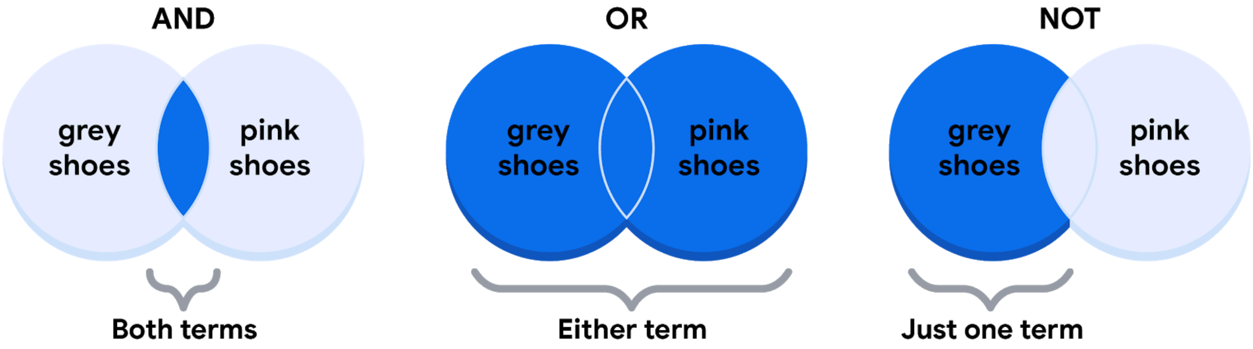


Boolean logic example

Imagine you are shopping for shoes, and are considering certain preferences:

- You will buy the shoes only if they are any combination of pink and grey
- You will buy the shoes if they are entirely pink, entirely grey, or if they are pink and grey
- You will buy the shoes if they are grey, but not if they have any pink

These Venn diagrams illustrate your shoe preferences. **`AND`** is the center of the Venn diagram, where two conditions overlap. **`OR`** includes either condition. **`NOT`** includes only the part of the Venn diagram that doesn't contain the exception.



The intersection of these circles is highlighted to indicate the AND condition requires shoes to be both grey and pink. The Venn diagram that represents OR includes a circle labeled grey shoes overlapping with a circle labeled pink shoes. The entirety of both circles is highlighted to indicate the OR condition means any shoe with grey, pink, or some combination satisfies the requirement. The Venn diagram that represents NOT includes a circle labeled grey shoes overlapping with a circle labeled pink shoes. The portion of the grey shoes circle that does not intersect with the pink shoes circle is highlighted to indicate the NOT condition requires shoes to not include pink.

Use Boolean logic in statements

In queries, Boolean logic is represented in a statement written with Boolean operators. An **operator** is a symbol that names the operation or calculation to be performed. Read on to discover how you can convert your shoe preferences into Boolean statements.

The AND operator

Your condition is "If the color of the shoe has any combination of grey and pink, you will buy them." The Boolean statement would break down the logic of that statement to filter your results by both colors. It would say `IF (Color="Grey") AND (Color="Pink") then buy them`

The `AND` operator lets you stack both of your conditions.

Below is a simple truth table that outlines the Boolean logic at work in this statement. In the **Color is Grey** column, there are two pairs of shoes that meet the color condition. And in the **Color is Pink** column, there are two pairs that meet that condition. But in the **If Grey AND Pink** column, only one pair of shoes meets both conditions. So, according to the Boolean logic of the statement, there is only one pair marked true. In other words, there is one pair of shoes that you would buy.

Color is Grey	Color is Pink	If Grey AND Pink, then Buy	Boolean Logic
Grey/True	Pink/True	True/Buy	True AND True = True
Grey/True	Black/False	False/Don't buy	True AND False = False
Red/False	Pink/True	False/Don't buy	False AND True = False
Red/False	Green/False	False/Don't buy	False AND False = False

The OR operator

The `OR` operator lets you move forward if either one of your two conditions is met. Your condition is "If the shoes are grey or pink, you will buy them." The Boolean statement would be `IF (Color="Grey") OR (Color="Pink") then buy them`.

Notice that any shoe that meets either the **Color is Grey** or the **Color is Pink** condition is marked as true by the Boolean logic. According to the truth table below, there are three pairs of shoes that you can buy.

Color is Grey	Color is Pink	If Grey OR Pink, then Buy	Boolean Logic
Red/False	Black/False	False/Don't buy	False OR False = False
Black/False	Pink/True	True/Buy	False OR True = True
Grey/True	Green/False	True/Buy	True OR False = True
Grey/True	Pink/True	True/Buy	True OR True = True

The NOT operator

Finally, the **`NOT`** operator lets you filter by subtracting specific conditions from the results. Your condition is "You will buy any grey shoe except for those with any traces of pink in them." Your Boolean statement would be **`IF (Color="Grey") AND (Color=NOT "Pink") then buy them`** Now, all of the grey shoes that aren't pink are marked true by the Boolean logic for the **`NOT`** Pink condition. The pink shoes are marked false by the Boolean logic for the **`NOT`** Pink condition. Only one pair of shoes is excluded in the truth table below.

Color is Grey	**Color is Pink**	**Boolean Logic for NOT Pink**	**If Grey AND (NOT Pink), then Buy**	**Boolean Logic**
Grey/True	Red/False	Not False = True	True/Buy	True AND True = True
Grey/True	Black/False	Not False = True	True/Buy	True AND True = True
Grey/True	Green/False	Not False = True	True/Buy	True AND True = True
Grey/True	Pink/True	Not True = False	False/Don't buy	True AND False = False

The power of multiple conditions

For data analysts, the real power of Boolean logic comes from being able to combine multiple conditions in a single statement. For example, if you wanted to filter for shoes that were grey or pink, and waterproof, you could construct a Boolean statement such as: “**`IF ((Color = "Grey") OR (Color = "Pink")) AND (Waterproof="True")`**

Notice that you can use parentheses to group your conditions together.

Key takeaways

Operators are symbols that name the operation or calculation to be performed. The operators **`AND`**, **`OR`**, and **`NOT`** can be used to write Boolean statements in programming languages. Whether you are doing a search for new shoes or applying this logic to queries, Boolean logic lets you create multiple conditions to filter your results. Now that you know a little more about Boolean logic, you can start using it!

Resources for more information

- Learn about who pioneered Boolean logic in this historical article: Origins of Boolean Algebra in the Logic of Classes.
- Find more information about using **`AND`**, **`OR`**, and **`NOT`** from these tips for searching with Boolean operators.

Step-by-Step: Meet wide and long data

This reading outlines the steps the instructor performs in the following video, Meet wide and long data. In this video, the instructor presents wide and long data formats and discusses the types of questions each format can help you answer.

Keep this guide open as you watch the video. It can serve as a helpful reference if you need additional context or clarification while following the video steps. This is not a graded activity, but you can complete these steps to practice the skills demonstrated in the video.

What you'll need

If you would like to access the spreadsheets the instructor uses in this video, select the link to a dataset to create a copy. If you don't have a Google account, download the data directly from the attachments below.

Link to population datasets:

- Population, Latin, and Caribbean Countries, 2010–2019, wide format
- Population, Latin, and Caribbean Countries, 2010–2019, long format

OR

Download data:

Population, Latin, and Caribbean Countries, 2010–2019, wide format
XLSX File
Population, Latin, and Caribbean Countries, 2010–2019, long format
XLSX File

Example 1: Examine wide data

Wide data is a dataset in which every data subject has a single row with multiple columns to hold the values of various attributes of the subject. It is helpful for comparing specific attributes across different subjects.

1. Open the Population, Latin, and Caribbean Countries, 2010–2019, wide format spreadsheet.
2. Each row contains all population data for one country.
3. The population data for each year is contained in a column.
4. Find the annual population of Argentina in row 3.
5. In this wide format, you can quickly compare the annual population of Argentina to the annual populations of Antigua and Barbuda, Aruba, the Bahamas, or any other country.

Find the country with the highest population in 2010

1. Select column **E**, which contains each country's 2010 population data.
2. Right-click column header **E** and choose **Sort Z to A**.
3. Notice that Brazil is now at the top of the list because it had the highest population in the year 2010.

Find the country with the lowest population in 2013

1. Select column **H**.
2. Right-click column header **H** and choose **Sort A to Z**.

3. Notice that the British Virgin Islands are now at the top because they had the lowest population of all countries in 2013.

Example 2: Examine long data

Long data is data in which each row represents one observation per subject, so each subject will be represented by multiple rows. This data format is useful for comparing changes over time or making other comparisons across subjects.

1. Open the Population, Latin, and Caribbean Countries, 2010–2019, long format spreadsheet.
2. Notice the data is no longer organized into columns by year. All of the years are now in one column.
3. Find Argentina's population data in rows 12-21. Each row contains one year of Argentina's population data.

Transforming data

What is data transformation?

A woman presenting data, a hand holding a medal, two people chatting, a ship's wheel being steered, two people high-fiving each other

In this reading, you will explore how data is transformed and the differences between wide and long data. **Data transformation** is the process of changing the data's format, structure, or values. As a data analyst, there is a good chance you will need to transform data at some point to make it easier for you to analyze it.

Data transformation usually involves:

- Adding, copying, or replicating data
- Deleting fields or records
- Standardizing the names of variables
- Renaming, moving, or combining columns in a database
- Joining one set of data with another
- Saving a file in a different format. For example, saving a spreadsheet as a comma separated values (.csv) file.

Why transform data?

Goals for data transformation might be:

- Data **organization**: better organized data is easier to use
- Data **compatibility**: different applications or systems can then use the same data
- Data **migration**: data with matching formats can be moved from one system to another
- Data **merging**: data with the same organization can be merged together
- Data **enhancement**: data can be displayed with more detailed fields

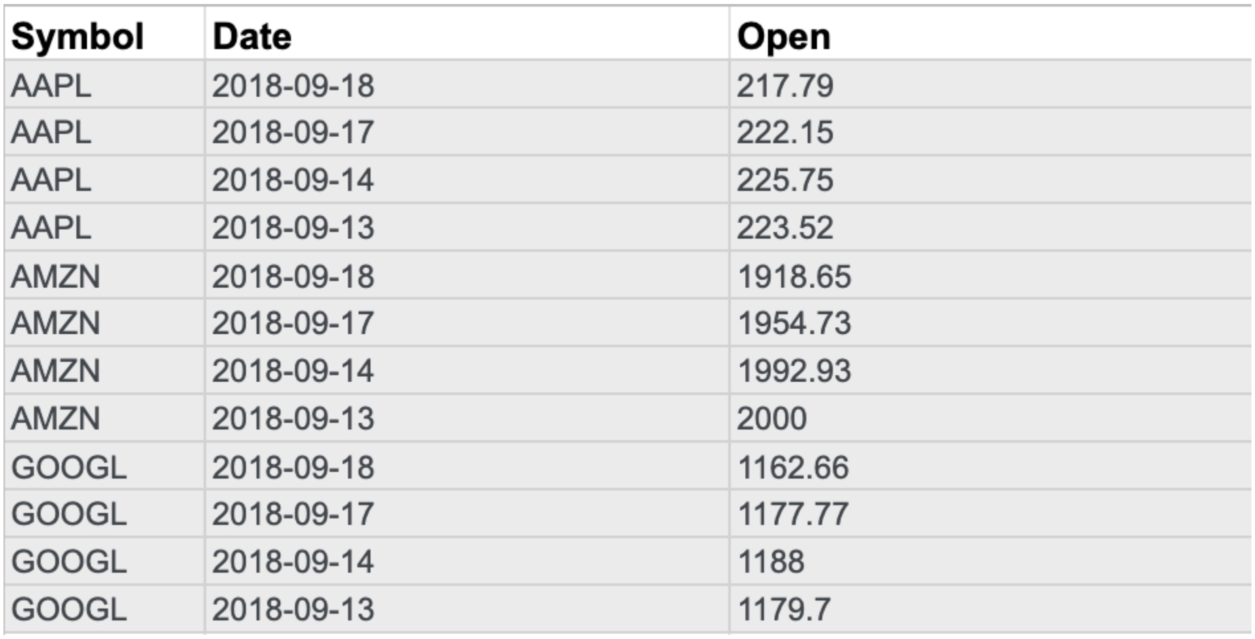
- Data **comparison**: apples-to-apples comparisons of the data can then be made

Data transformation example: data merging

Mario is a plumber who owns a plumbing company. After years in the business, he buys another plumbing company. Mario wants to merge the customer information from his newly acquired company with his own, but the other company uses a different database. So, Mario needs to make the data compatible. To do this, he has to transform the format of the acquired company's data. Then, he must remove duplicate rows for customers they had in common. When the data is compatible and together, Mario's plumbing company will have a complete and merged customer database.

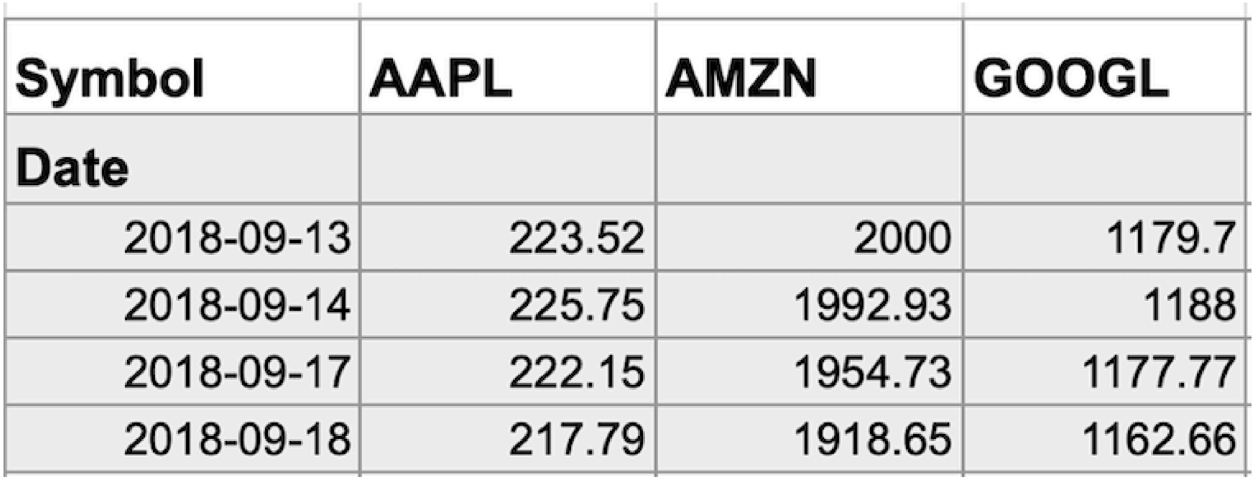
Data transformation example: data organization (long to wide)

To make it easier to create charts, you may also need to transform long data to wide data. Consider the following example of transforming stock prices (collected as long data) to wide data.
Long data is data where **each row contains a single data point** for a particular item. In the long data example below, individual stock prices (data points) have been collected for Apple (AAPL), Amazon (AMZN), and Google (GOOGL) (particular items) on the given dates.
Long data example: Stock prices



Symbol	Date	Open
AAPL	2018-09-18	217.79
AAPL	2018-09-17	222.15
AAPL	2018-09-14	225.75
AAPL	2018-09-13	223.52
AMZN	2018-09-18	1918.65
AMZN	2018-09-17	1954.73
AMZN	2018-09-14	1992.93
AMZN	2018-09-13	2000
GOOGL	2018-09-18	1162.66
GOOGL	2018-09-17	1177.77
GOOGL	2018-09-14	1188
GOOGL	2018-09-13	1179.7

Wide data is data where **each row contains multiple data points** for the particular items identified in the columns.
Wide data example: Stock prices



Symbol	AAPL	AMZN	GOOGL
Date			
2018-09-13	223.52	2000	1179.7
2018-09-14	225.75	1992.93	1188
2018-09-17	222.15	1954.73	1177.77
2018-09-18	217.79	1918.65	1162.66

With data transformed to wide data, you can create a chart comparing how each company's stock changed over the same period of time.

You might notice that all the data included in the long format is also in the wide format. But wide data is easier to read and understand. That is why data analysts typically transform long data to wide data more often than they transform wide data to long data. The following table summarizes when each format is preferred:

Wide data is preferred when	**Long data is preferred when**
Creating tables and charts with a few variables about each subject	Storing a lot of variables about each subject. For example, 60 years worth of interest rates for each bank
Comparing straightforward line graphs	Performing advanced statistical analysis or graphing