

Linear Equations

The equations in the previous lab included one variable, for which you solved the equation to find its value. Now let's look at equations with multiple variables. For reasons that will become apparent, equations with two variables are known as linear equations.

Solving a Linear Equation

Consider the following equation:

$$2y + 3 = 3x - 1$$

This equation includes two different variables, **x** and **y**. These variables depend on one another; the value of **x** is determined in part by the value of **y** and vice-versa; so we can't solve the equation and find absolute values for both **x** and **y**. However, we *can* solve the equation for one of the variables and obtain a result that describes a relative relationship between the variables.

For example, let's solve this equation for **y**. First, we'll get rid of the constant on the right by adding 1 to both sides:

$$2y + 4 = 3x$$

Then we'll use the same technique to move the constant on the left to the right to isolate the **y** term by subtracting 4 from both sides:

$$2y = 3x - 4$$

Now we can deal with the coefficient for **y** by dividing both sides by 2:

$$y = \frac{3x - 4}{2}$$

Our equation is now solved. We've isolated **y** and defined it as $^{3x-4}/_2$

While we can't express **y** as a particular value, we can calculate it for any value of **x**. For example, if **x** has a value of 6, then **y** can be calculated as:

$$y = \frac{3 \cdot 6 - 4}{2}$$

This gives the result $^{14}/_2$ which can be simplified to 7.

You can view the values of **y** for a range of **x** values by applying the equation to them using the following Python code:

```
In [1]: import pandas as pd

# Create a dataframe with an x column containing values from -10 to 10
df = pd.DataFrame({'x': range(-10, 11)})

# Add a y column by applying the solved equation to x
df['y'] = (3*df['x'] - 4) / 2

#Display the dataframe
df
```

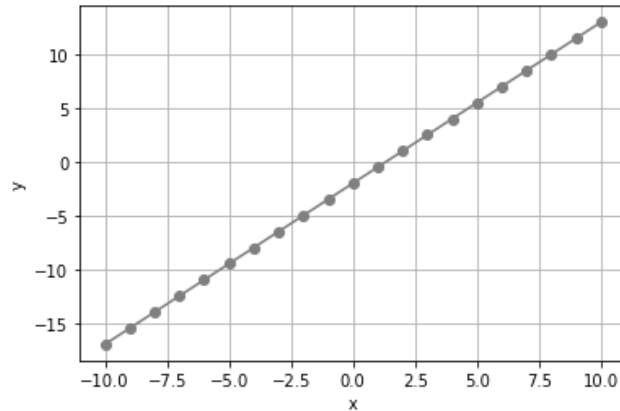
Out[1]:

	x	y
0	-10	-17.0
1	-9	-15.5
2	-8	-14.0
3	-7	-12.5
4	-6	-11.0
5	-5	-9.5
6	-4	-8.0
7	-3	-6.5
8	-2	-5.0
9	-1	-3.5
10	0	-2.0
11	1	-0.5
12	2	1.0
13	3	2.5
14	4	4.0
15	5	5.5
16	6	7.0
17	7	8.5
18	8	10.0
19	9	11.5
20	10	13.0

We can also plot these values to visualize the relationship between x and y as a line. For this reason, equations that describe a relative relationship between two variables are known as *linear equations*:

```
In [2]: %matplotlib inline
from matplotlib import pyplot as plt

plt.plot(df.x, df.y, color="grey", marker = "o")
plt.xlabel('x')
plt.ylabel('y')
plt.grid()
plt.show()
```



In a linear equation, a valid solution is described by an ordered pair of x and y values. For example, valid solutions to the linear equation above include:

- (-10, -17)
- (0, -2)
- (9, 11.5)

The cool thing about linear equations is that we can plot the points for some specific ordered pair solutions to create the line, and then interpolate the x value for any y value (or vice-versa) along the line.

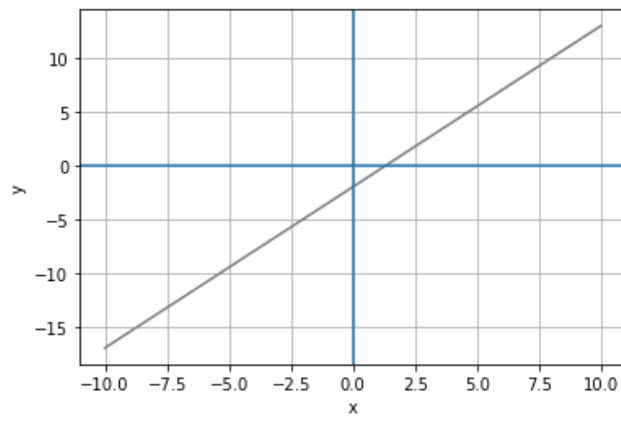
Intercepts

When we use a linear equation to plot a line, we can easily see where the line intersects the X and Y axes of the plot. These points are known as *intercepts*. The *x-intercept* is where the line intersects the X (horizontal) axis, and the *y-intercept* is where the line intersects the Y (vertical) axis.

Let's take a look at the line from our linear equation with the X and Y axis shown through the origin (0,0).

```
In [3]: plt.plot(df.x, df.y, color="grey")
plt.xlabel('x')
plt.ylabel('y')
plt.grid()

## add axis lines for 0,0
plt.axhline()
plt.axvline()
plt.show()
```



The x-intercept is the point where the line crosses the X axis, and at this point, the **y** value is always 0. Similarly, the y-intercept is where the line crosses the Y axis, at which point the **x** value is 0. So to find the intercepts, we need to solve the equation for **x** when **y** is 0.

For the x-intercept, our equation looks like this:

$$0 = \frac{3x - 4}{2}$$

Which can be reversed to make it look more familiar with the x expression on the left:

$$\frac{3x - 4}{2} = 0$$

We can multiply both sides by 2 to get rid of the fraction:

$$3x - 4 = 0$$

Then we can add 4 to both sides to get rid of the constant on the left:

$$3x = 4$$

And finally we can divide both sides by 3 to get the value for x:

$$x = \frac{4}{3}$$

Which simplifies to:

$$x = 1\frac{1}{3}$$

So the x-intercept is $1\frac{1}{3}$ (approximately 1.333).

To get the y-intercept, we solve the equation for y when x is 0:

$$y = \frac{3 \cdot 0 - 4}{2}$$

Since 3×0 is 0, this can be simplified to:

$$y = \frac{-4}{2}$$

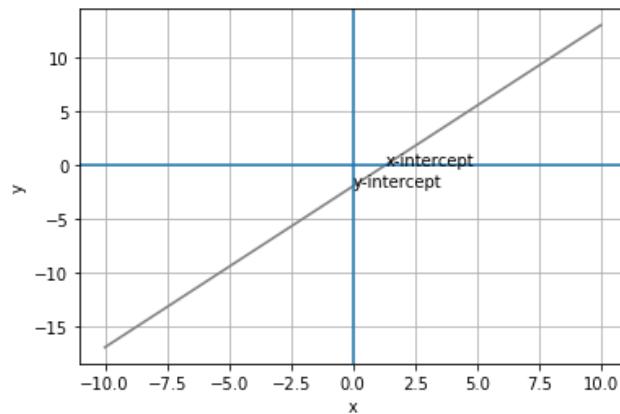
-4 divided by 2 is -2, so:

$$y = -2$$

This gives us our y-intercept, so we can plot both intercepts on the graph:

```
In [4]: plt.plot(df.x, df.y, color="grey")
plt.xlabel('x')
plt.ylabel('y')
plt.grid()

## add axis lines for 0,0
plt.axhline()
plt.axvline()
plt.annotate('x-intercept', (1.333, 0))
plt.annotate('y-intercept', (0, -2))
plt.show()
```



The ability to calculate the intercepts for a linear equation is useful, because you can calculate only these two points and then draw a straight line through them to create the entire line for the equation.

Slope

It's clear from the graph that the line from our linear equation describes a slope in which values increase as we travel up and to the right along the line. It can be useful to quantify the slope in terms of how much **x** increases (or decreases) for a given change in **y**. In the notation for this, we use the greek letter Δ (*delta*) to represent change:

$$slope = \frac{\Delta y}{\Delta x}$$

Sometimes slope is represented by the variable **m**, and the equation is written as:

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

Although this form of the equation is a little more verbose, it gives us a clue as to how we calculate slope. What we need is any two ordered pairs of x,y values for the line - for example, we know that our line passes through the following two points:

- (0,-2)
- (6,7)

We can take the x and y values from the first pair, and label them x_1 and y_1 ; and then take the x and y values from the second point and label them x_2 and y_2 . Then we can plug those into our slope equation:

$$m = \frac{7 - -2}{6 - 0}$$

This is the same as:

$$m = \frac{7 + 2}{6 - 0}$$

That gives us the result $\frac{9}{6}$ which is $1\frac{1}{2}$ or 1.5 .

So what does that actually mean? Well, it tells us that for every change of **1** in **x**, **y** changes by $1\frac{1}{2}$ or 1.5. So if we start from any point on the line and move one unit to the right (along the X axis), we'll need to move 1.5 units up (along the Y axis) to get back to the line.

You can plot the slope onto the original line with the following Python code to verify it fits:

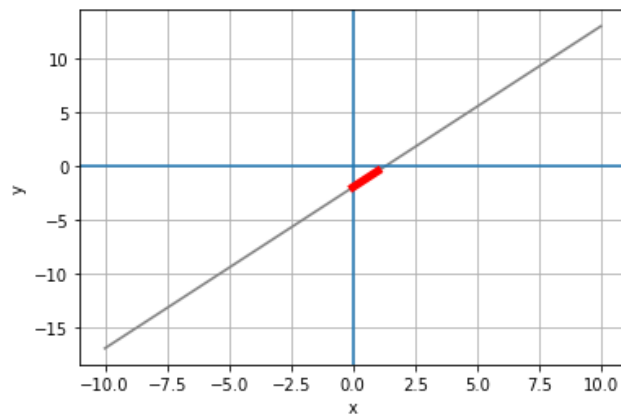
```
In [5]: plt.plot(df.x, df.y, color="grey")
plt.xlabel('x')
plt.ylabel('y')
plt.grid()
plt.axhline()
plt.axvline()

# set the slope
m = 1.5

# get the y-intercept
yInt = -2

# plot the slope from the y-intercept for 1x
mx = [0, 1]
my = [yInt, yInt + m]
plt.plot(mx,my, color='red', lw=5)

plt.show()
```



Slope-Intercept Form

One of the great things about algebraic expressions is that you can write the same equation in multiple ways, or *forms*. The *slope-intercept form* is a specific way of writing a 2-variable linear equation so that the equation definition includes the slope and y-intercept. The generalised slope-intercept form looks like this:

$$y = mx + b$$

In this notation, ***m*** is the slope and ***b*** is the y-intercept.

For example, let's look at the solved linear equation we've been working with so far in this section:

$$y = \frac{3x - 4}{2}$$

Now that we know the slope and y-intercept for the line that this equation defines, we can rewrite the equation as:

$$y = 1\frac{1}{2}x + -2$$

You can see intuitively that this is true. In our original form of the equation, to find y we multiply x by three, subtract 4, and divide by two - in other words, x is half of 3x - 4; which is 1.5x - 2. So these equations are equivalent, but the slope-intercept form has the advantages of being simpler, and including two key pieces of information we need to plot the line represented by the equation. We know the y-intercept that the line passes through (0, -2), and we know the slope of the line (for every x, we add 1.5 to y).

Let's recreate our set of test x and y values using the slope-intercept form of the equation, and plot them to prove that this describes the same line:

```
In [6]: %matplotlib inline
```

```
import pandas as pd
from matplotlib import pyplot as plt

# Create a dataframe with an x column containing values from -10 to 10
df = pd.DataFrame({'x': range(-10, 11)})

# Define slope and y-intercept
m = 1.5
yInt = -2

# Add a y column by applying the slope-intercept equation to x
df['y'] = m*df['x'] + yInt

# Plot the line
from matplotlib import pyplot as plt

plt.plot(df.x, df.y, color="grey")
plt.xlabel('x')
plt.ylabel('y')
plt.grid()
plt.axhline()
plt.axvline()

# label the y-intercept
plt.annotate('y-intercept', (0, yInt))

# plot the slope from the y-intercept for 1x
mx = [0, 1]
my = [yInt, yInt + m]
plt.plot(mx, my, color='red', lw=5)

plt.show()
```

