

Variables and Their Roles

This section explains types and interactions between variables.

Here is a video on variables and their roles;

YouTube link: <https://www.youtube.com/watch?v=uWp4HFE3PJo>

To best understand variables and their roles, let's revisit the patio situation and consider each of the pieces of data as "potential generalized data". In our patio example we might consider the following pieces of data;

- The length and width of the (rectangular) patio.
- The area of the patio.
- The cost of the cement pavers.
- The size (dimensions; length and width) of a single cement paver.
- The number of necessary cement pavers to make the patio.
- The cost of labor to build the patio.
- The time to build the patio.

Example 1. *When trying to determine which piece of data should be represented by which type of variable, it is helpful to consider how each piece of data is related to the others. For example, the length and width of the patio could give you the (cost/ area \checkmark / size), so those things are related (but notice that, having the (cost/ area \checkmark / size) doesn't give you enough information to determine both length and width). The area of the patio and size of an individual paver is similarly related to the (number \checkmark / value/ size) of cement pavers one needs to build the patio with the pavers. Moreover, with a minimal amount of effort (and some basic geometry and algebra) we could write down these relationships mathematically.*

¹

Relationships between variables... Man that's way too many words!

Thus, we could write down the following relationships:

Learning outcomes:

¹These are mathematical relationships and functions, which we will cover in more detail later. For now we simply state what the relationships are between the specific pieces of data we have.

- The (Number of Pavers Needed) is approximately² the (Area of the Patio) divided by the (Area of Pavers).
- The (Area of Pavers) is equal to the product of the (Width of Pavers) and the (Length of Pavers).
- The (Area of Patio) is the product of the (Width of Patio) and the (Length of Patio).
- The (Total Cost of Pavers) is the product of (Cost of a Paver) and (Number of Pavers Needed)
- The (Total Cost of Labor) is the product of the (Hourly Cost of Labor) and the (Time to Build Patio (in hours)).
- The (Total Cost of Patio) is the sum of the (Total Cost of Pavers) and the (Total Cost of Labor).

Look at the list above and try to read them out loud. Really; go do it. How far did you get before you stopped to come back and read this because it was obnoxious to read all those words for “such obvious relationships”? It should be clear that, although we have to write phrases like “Area of Pavers” and “Width of Patio”, it would be a lot easier if we could encode this information in something faster and easier to read; after all, we know what we mean right?³ This is where variables come into play. We could build an encoding, a kind of “quick reference sheet” for a shorthand to refer to these things. An example of such a thing might be the following;

- n is the number of bricks needed to build our patio.
- A_p is the area of the patio (in square feet).
- A_b is the area of a paver brick.
- L_p and W_p are the length and width of the patio respectively.
- L_b and W_b are the length and the width of the paver bricks respectively.
- C_l is the cost of the labor (in dollars).
- C_b is the cost of the paver bricks (in dollars).
- C_T is the total cost of the patio project (in dollars).
- t_p is the total time to build the patio.

²We say approximately here as there is some difficulty in the situation where the pavers don’t exactly line up with the dimensions of the patio; eg if the patio is fifteen and two thirds of a paver long, it would likely require more paver purchases than this estimate suggests.

³This phrase is often used and almost *always* regretted at some point.

The above looks intimidating. That's an awful lot of letters, but there are a few things to keep in mind when looking at that list.

- Exploration 1** (a) *The variables names I chose were not pulled out of a hat. I deliberately picked names that correspond in some nice/'obvious' way to what it represents. For example, notice that all the 'Cost' based variables are $C_{\text{something}}$, and moreover, that 'something' clues you into what the variable is the cost of; p for patio, l for labor, b for (paver) brick. Thus by intelligently naming your various, you can often make things more sensible.*
- (b) *We have generalized everything we possibly could and as I mentioned earlier, this is almost always overkill. For now it's helpful to see what the possibilities are for generalizing, then we will want to cut back to which specific data would be helpful to generalize.*
- (c) *Despite the first point above, the variable names may make sense in context, but it will be easy to forget that context if we were to put this down and come back to it in six months. For this reason, it's always a good idea to explicitly write down all your variables and what they literally mean. This means writing something like " A_p = area of the patio in square feet" **not** A_p = area of patio. Units are the easiest thing to forget, and typically the (least/ somewhat plausible/ most ✓) likely area to cause errors... just ask NASA!⁴.*

So what does this have to do with variable types?

Exploration 2 *The next step is formalizing the relationship between these variables. This is something we will cover much more in the next topic, but for now we could probably conclude the following relationship from the above variables;*

$$A_p = L_p \cdot W_p$$

Which tells us that the ((length of the patio) / (width of the patio) / (area of the patio) ✓) is equal to the ((length of the patio) ✓ / (area of the patio)) times the ((width of the patio) ✓ / (area of the patio)); the basic formula for the area of a rectangle. Observe that with any two of these pieces of data, we could get the third (eg with width and total area, we could calculate the length). So the question is, what does our model expect to have provided to it, and what do we want our model to tell us? Take the following two examples:

⁴NASA crashed a probe into a planet at great rates of speed because one data team used metric units and the other used imperial units... and nobody bothered to check before they put them together; they just used the numbers without units. That was an expensive mistake.

Example 2. You're given length and width.

Let's say you work for a construction company and you are asked to make a model to determine the price of building a patio for a customer. That customer has very specific dimensions that they want, and so you know you will be given the length and width of the patio, but your calculations require area. In this case you would use the original equation above;

$$A_p = L_p \cdot W_p$$

Here the relationship between these variables in your model expects to have you supply (length and width ✓/ area and length/ area and width) of the patio (the given information from your customer) and it will in turn calculate the (area ✓/ width/ length) of the patio.

Explanation. You're given Area and width

Now let's say a customer is planning an above-ground pool. They know they need a certain amount of minimum square footage to have a foundation for their pool, a grill, and a lounge area. Furthermore they know they want the patio to run the full width of their main porch area. So when building the model, you want to build it assuming you will know the area and the width, but not the length. Thus you would use the following relationship;

$$L_p = \frac{A_p}{W_p}$$

In both these cases we have the same variables and the model has the same end goal (to calculate the cost to build a patio), but in one situation we expect to know the length and width, but need the area. In the other situation you expect to know the area and width, but need the length.

Exploration 3 Although we call A_p , L_p , and W_p 'variables' in both cases, we have special names to denote this "expectation" aspect that is, in some sense, equally important to include in a model. For variables that we expect to be provided to us, we call them **independent variables**. They are called "independent" because they are (suppose to be) supplied independent of the model, meaning that they are the data that is "fed into" the model to get results. The variables that are calculated or deduced by the model are called **dependent variables**. These variables are called dependent because their value depends on what is put into the model (ie the dependent variables may⁵ change value for different independent variable values). In general, if you (for a moment) think of a model as a magic machine, then independent variables are 'fed into' that

⁵Dependent variables are *capable* of changing value based on independent variable values, but that is not the same as saying they *must* change value. This is a subtle distinction, but it turns out it's *incredibly* important as we'll see much later when we are discussing functions and especially inverse functions.

machine, and dependent variables are ‘spit back out’ as your “answers”.⁶ So, in Example One above, L_p and $(W_p \checkmark / A_p)$ are both independent variables and $(W_p / A_p \checkmark)$ is a dependent variable, but in Example Two, A_p and $(W_p \checkmark / L_p)$ are both independent variables and $(W_p / L_p \checkmark)$ is a dependent variable.⁷

There are other kinds of variables one could encounter as well; of specific importance in calculus is the *arbitrary constant*. This is typically a result of some initial information used in your model, and is a byproduct of choices in your model, but *they are unaffected by independent variables*. One can think of the arbitrary constant as being a sort of “starting spot” for your model. That is to say, even though your “starting spot” is typically of great importance to your outcome (your starting height when throwing a ball and measuring how far it goes for example) no matter what information you “feed into” the model (eg throwing speed, throwing angle, etc), your starting spot doesn’t change. Thus the arbitrary constant doesn’t change based on any of these “input” values.

One might wonder what the difference is, then, between a constant and an arbitrary constant. It may be clear that it doesn’t vary based on independent variables, so it seems like it is constant. The key point though is that you may not know the intended “starting spot” (ie the height someone will throw from) of your model when you are designing it.

Consider our patio example again. You are building a generic model to calculate the cost of building a patio for your company, and part of that is travel costs. The travel costs themselves will depend on the location (where the customer lives, how accessible the construction site is, etc). But once you have determined the cost for travel, it won’t change depending on the size of the patio (the independent variable). Thus it will be a constant value, but one that depends on the customer’s location, not the customer’s specific project. This is cost would then be an arbitrary constant; something that varies from project to project (specific model to specific model) but is constant within the specific model.

⁶ “Answers” here is in quotes as dependent variables often occur in substeps of models. In our examples above, our real “answer” would be the cost of the patio, whereas the dependent variables (the area in the first example and length in the second) are “answers” to their respective equations but one would typically not call them an “answer” to the model, which tends to be what we mean when we say “The Answer”. This is why using words like “solution” or “answer” can be dangerous, and *one should always be clear as to what they are claiming their result is an “answer” to specifically*.

⁷ Often in math classes dependent and independent variables are described by rote, meaning they simply say “ x is an independent variable” and “ y is a dependent variable”. This is *often* true, but it’s very important to notice that there was no comment made in our definition about a certain letter needing to be a certain type of variable. In fact we demonstrated that the same letter could be one or the other depending on the model we are building!