Visualization is a crucial step for initial data exploration.

It helps us discern relationships, patterns, and outliers.

This video will give us a starting point on how to make plots in R, but more advanced and way cooler visualization tips will be given in Week 8 of this class.

Let us first create a scatterplot with Protein on the x-axis and Fat on the y-axis.

To do this we can use the plot function in R and give it as a first input the Protein vector on the x-axis, and as a second input the TotalFat vector on the y-axis.

And now pressing Enter, a new window pops up.

If you are on a PC, the new Windows is called R Graphics, and if you are on a Mac it is called Quartz.

The plot has a very interesting triangular shape.

It looks like foods that are higher in protein are typically lower in fat, and vice versa.

Now, looking at the aesthetics of the graph, we realize that R gives default names for the x-axis and the y-axis using the vector dollar notation.

Now we can modify these labels by adding more arguments to the plot function.

To go back to the R console, you can simply switch windows using the Control tab if you are on a Windows machine.

On a Mac, the windows are not overlaid by default, so accessing the console should be easy.

OK, now let's use the Up arrow to go back to our plot function, and add the argument xlab = "Protein" and that gives us the label of the x-axis.

Then ylab = "Fat" and this gives us the label to the y-axis.

And let's add a title to the plot using the argument main, say the title is "Protein vs Fat".

And let's change the color to red.

And remember to put quotation marks around the values of all these arguments.

And now pressing Enter and going back to the graphics window, we see that R made all the modifications we requested.

Another way we can visualize our data is by plotting histograms.

We can do this using the histogram function in R, but note that this function now only takes one variable as an input, because the y-axis should have the frequencies.

So let's go back to our console and create a histogram of VitaminC, for instance.

So we're going to use the hist function, or histogram.

And then the argument that it takes is the VitaminC vector.

Let's label the x-axis as "Vitamin C", and this is given to us in milligrams.

And give it a title, say "Histogram of Vitamin C Levels".

Hmm.

Even though the maximum vitamin C content is 2000 milligrams, most of our foods-- well, to be more precise, more than 6,000 of them-- have less than 200 milligrams of vitamin C.

And the histogram lumps them all together in one cell.

Well, it would be nice if we can zoom into this section here and get a finer understanding of the data.

To do this we need to limit the x-axis to go from zero to, say, 100 milligrams.

So let's go back to the console, and then we're going to add the argument xlim to limit the x-axis.

And using the combine function or the c function, we're going to set the first input to be 0, which is the lowest value that we want to see on the x-axis, and the second argument as being 100, which is the highest value that we want to see on the x-axis.

And now pressing Enter, and we can see that R gives us 0 to 100 on the x-axis.

But we only see this one big cell.

It seems that R only zoomed into the area, but it didn't break that huge cell, and this doesn't give us any additional information.

So we really need to break up the cell into smaller pieces.

And say we want 100 cells, and since the interval goes from 0 to 100, then we would expect R to create divisions that are one milligrams in length.

So let's do this.

Let's go back to the console, and then we're going to add the argument breaks = 100 and this sets the number of cells that we want to see to 100.

So let's see.

Oh.

We actually only see five cells, and each cell is 20 milligrams long.

Well, what happened?

We were expecting 100 cells.

Well remember that the histogram originally went far beyond 100 milligrams.

The maximum was 2000 milligrams.

And now if we were to divide the original interval from 0 to 2000 into 100 cells, then 2000 divided by 100, each cell would be 20 milligrams long.

And this is exactly what R did.

It actually divided all of the spectrum of values from 0 to 2000 into 100 cells, and not only the spectrum from 0 to 100.

But we still want to divide the interval 0 to 100 into 100 cells, each of length 1 milligram.

And how can we do this?

Well, we simply need to think in terms of the original interval, which was 0 to 2000.

And if we were to break it into 2000 cells, then each one will be of length one milligram.

So now we know that actually we needed to set the breaks to 2000.

So let's do this.

And now we obtain our refined histogram.

And we see new information here come up.

Remember our initial conclusion was that more than 6,000 foods have less than 200 milligrams of vitamin C.

But now that we refined our graph, we obtained an additional level of information.

Actually, more than 5,000 of them have less than one milligram of vitamin C.

Now a third way we can visualize the data is using box plots.

So let's go back to the console and create a box plot for sugar.

So the function we're going to be using is simply boxplot, and similarly to the histogram function, the boxplot function only takes as an input a single vector.

And in this case it would be the Sugar vector.

And let's create a title that says "Boxplot of Sugar Levels".

And is it the y-axis or the x-axis that we have to label it as the sugar level?

So if we're not so sure, let's just plot it, and, oh, this should be the y-axis.

So let's go back to the console, and simply set the y label to be "Sugar (g)".

And now we have our box plot with the right labels.

What is it trying to tell us here?

It looks a little bit strange.

Well, the average of sugar across the data set seems to be pretty low.

It's somewhere around five grams.

But we have a lot of outliers with extremely high values of sugar.

There exist some foods that have almost 100 grams of sugar in 100 grams.

Well, candies are definitely among these foods.

So we just reviewed three ways in which we can visualize our data.

In Week 8, we will see more advanced visualization tools to make more informative plots.

In our next video we will see how we can construct and add new variables to our data set.