MITOCW | MIT15_071S17_Session_1.4.04_300k

In this video, we will do some basic data analysis.

All that I've done since our previous video is clear the console, but R still has all the information stored.

In fact, if we use the Up Arrow on our keyboard, we retrieve the last command we typed, which was the summary of the USDA data frame.

And as a quick reminder, at the end of our last video, we realized that the maximum level of Sodium was 38,758 milligrams, which is very high.

We would like to see which food this corresponds to.

Well, to check the values of sodium levels in the foods within the data set, we can type USDA$Sodium.

This gives us a series of numbers associated with the amount of sodium in all the foods in our data set.

Remember from the lecture that this is called a vector, and it is associated with the variable Sodium.

For instance, the sodium level of the last food in our data set is 68 milligrams.

Now, to find which food has the highest level of sodium, we can simply use the function which.max, which takes as an input the Sodium vector, and it gives us the index of the food with the highest sodium level.

In this case, the 265th food in our data set has the maximum sodium content.

Now to know which food that is, we need to take a look at the vector corresponding to the text description of the foods.

However, I cannot remember the exact name of that variable on top of my head to be able to call it in R.

But we can use the function names, which takes as an input the USDA data frame and gives us the exact names of all the variables as stored in the USDA data frame.

And now we know that the name of the variable we're looking at is Description.

So now, to get the name of the 265th food, we simply need to ask R to pick the 265th element from the vector Description.

So, using our dollar notation to call the Description vector and then the square brackets around the index 265, and the winner is table salt!

Well, having 38,758 milligrams of sodium in 100 grams of table salt sort of makes sense, but none of us would eat 100 grams of salt in one sitting.

So it might be more interesting to find out which foods, for instance, contain more than, say, 10,000 milligrams of sodium.

To do so, we can create a new data frame, and let's call it HighSodium.

And this is going to be a subset of our original data frame, USDA, with only the foods that have sodium content that exceeds 10,000.

And now we created this new data frame, and to see how many foods there exist in this new data frame, we need to see how many observations this data frame has.

And this can be done by using the function nrow, which computes the number of rows in the data frame HighSodium.

And then we obtain 10 foods with sodium levels above 10,000 milligrams.

Since there are not many, we can output the names of these foods by looking at their Description vector.

But this time, the Description vector is not associated with the USDA data frame but with the HighSodium data frame.

So HighSodium$Description, and now pressing Enter, we obtain the names of these 10 foods.

So definitely table salt is one of them.

We also have dry soup, gravy, some leavening agents, but I thought caviar is well known to be among the top 10 foods with highest levels of sodium.

But it doesn't appear in this list.

Let's find how much sodium it has in 100 grams.

Now, obviously, this task would have been very easy if we knew the index of caviar in our data set, and we simply feed it into the vector Sodium.

However, we need to get the index of caviar, and to do this, we need to track down the word caviar in the text description.

To do this, we can use the match function and ask R to dig the word caviar in the description vector.

So USDA$Description.

And now pressing Enter, we obtain that caviar is the 4,154th food in our data set.

So now finding the sodium level of caviar is a piece of cake.

We just type USDA$Sodium and, using the square brackets with the index 4,154, ask R to pick the sodium level of caviar for us.

And this is 1,500 milligrams.

Now, to find a level of sodium in caviar, we used two steps, but we can actually lump them all in one single step.

So let's use the Up Arrow twice to go back to the match function, and we know that this match function gives us an index that then should be fed into the Sodium vector using square brackets.

So let's enclose it in square brackets, and then at the beginning we're going to just write USDA$Sodium.

And, again, of course, this gives us 1,500 milligrams of sodium in 100 grams of caviar.

Now, the value 1,500 milligrams seems to be very small compared to 10,000 milligrams or 38,000 milligrams, which are the values that we worked with so far with respect to sodium levels.

But this doesn't seem to be a fair comparison.

Maybe the best way to figure out how big this value is, is by comparing it to the mean and the standard deviation of the sodium levels across the data set.

To find the mean, we know that this information is given to us using the summary function.

So let's use the summary function, and this time, give it the input the Sodium vector instead of the whole USDA data frame.

And we can see that the mean sodium value is 322 milligrams.

However, the summary function does not give us standard deviation information, but we can do this using the function sd, which stands for standard deviation.

Give it as an input the Sodium vector, and, oh, we obtain non-available.

Well we got NA because we forgot to remove the non-available entries before computing our statistical measure.

So let's use the Up Arrow to go back to the standard deviation function, and now we have to explicitly tell R to remove these non-available entries by typing na.rm=TRUE.

And now the standard deviation is 1,045 milligrams.

Note that, if we sum the mean and the standard deviation, we obtain around 1,400 milligrams, which is still smaller than the amount of sodium in 100 grams of caviar.

Well, this means that caviar is pretty rich in sodium compared to most of the foods in our data set.

Now that we know how to do a basic analysis of our data, let's look at the plotting functionality in R in our next video.