Remember that in our previous video, we created four new variables, HighSodium, HighFat, HighCarbs, and HighProtein.

Now in this video, we will try to understand our data and the relationships between our variables better, using the table and tapply functions.

To figure out how many foods have higher sodium level than average, we want to look at the HighSodium variable and count the foods that have values 1.

We can do this using the table function, and give it as an input the HighSodium vector.

Now pressing Enter, we obtain the following information.

Most of the foods in our data set, and precisely 4,800 of them, have lower sodium than average, and we have 2090 foods that have higher sodium content than average.

Now let's see how many foods have both high sodium and high fat.

Well, to do this we can also use the table function, but instead of giving it one input, now we can give it two inputs.

So let's go back using the Up arrow, and now have the first input being the HighSodium vector and the second input being the HighFat vector.

And we obtain the following table.

The rows belong to the first input, which is HighSodium, and the columns correspond to the second input, which is HighFat.

So from the table we see that we have 3,529 foods with low sodium and low fat, 1,355 foods with low sodium and high fat, 1,378 foods with high sodium but low fat, and finally 712 foods with both high sodium and high fat.

Now, what if we want to compute the average amount of iron sorted by high and low protein?

Well, to do this we can use the tapply function.

Let us have a little refresher on how the tapply function works.

The tapply function takes three arguments, and groups the first argument according to the second argument, and then applies argument three.

For instance, we wanted to compute the average amount of iron sorted by high and low protein.

In this case, the first argument is whatever we are trying to analyze, which is the Iron vector, and we are sorting it according to the vector HighProtein, which is our second argument.

And finally we apply the mean function in R on the sorted Iron values.

And we should not forget to remove the nonavailable entries.

So what does tapply do exactly?

Suppose that we have the following data frame with the foods one through six, along with information about their Iron levels and their values of HighProtein that we just added earlier.

The first step is grouping the Iron data according to the values of HighProtein.

So, first group, all the foods that have HighProtein equal 1, and that would be food number two with 12.8 milligrams of iron, food number three with 1.44 milligrams of iron, and food number six with 2.29 milligrams of iron.

Then we group the remaining foods that have protein levels below average, and this corresponds to food one, food four, and food five.

Then we compute the mean of Iron level for each group.

In this case, the mean of the group with high protein levels is 5.51, and the mean of the group with low protein levels is 1.72.

And this is the result of the tapply function.

Now let's go back to R and have a hands on practice on how to use the tapply function.

So let's compute the average amount of iron sorted by protein levels.

So we're going to type tapply, and then the first argument is the Iron vector which we are trying to analyze.

And we are sorting it according to the HighProtein vector, so this is our second argument.

And then the mean statistic is used, because we're trying to calculate the average level of Iron.

And do not forget to remove the non available entries by typing na.rm=TRUE.

And here's the result.

Foods with low protein content have on average 2.55 milligrams of iron and foods with high protein content have on average 3.2 milligrams of iron.

Now how about the maximum level of vitamin C in foods with high and low carbs?

Again, we're going to use the tapply function, so use the Up arrow to go back to the previous command, but now we're trying to analyze the VitaminC vector.

So this is our first argument, And we are sorting it according to high and low carbs, so the second argument is the vector HighCarbs.

And instead of the mean, we're applying here the max statistic, and we obtain the following.

The maximum vitamin C level, which is 2,400 milligrams is actually present in a food that is high in carbs.

Well, is it true that foods that are high in carbs have generally high vitamin C content?

Well, to see if this is the case, now we're going to go back to our tapply function, and instead of the max statistic we're going to use the summary function.

We obtain the following two sets of information.

The first set corresponds to the foods with low carb content, and the second set of information corresponds to foods with higher carb content than average.

Now the statistical information that the summary function gives us pertains to the vitamin C levels.

This means that we have on average 6.36 milligrams of vitamin C in foods with low carb content, and on average 16.31 milligrams of vitamin C in foods with high carb content.

So, it does seem like a general trend.

Foods with high carb content are on average richer in vitamin C compared to foods with low carb content.

Now we reach the end of our first recitation.

I hope this was a good exercise to familiarize yourself better with R, and learn some fun facts about nutrition.

Stay healthy.