

MITOCW | MIT15_071S17_Session_7.3.07_300k

In this video, we'll add the hour of the day to our line plot, and then create an alternative visualization using a heat map.

We can do this by creating a line for each day of the week and making the x-axis the hour of the day.

We first need to create a counts table for the weekday, and hour.

So we'll use the table function and give as the first variable, the Weekday variable in our data frame.

and as the second variable, the Hour variable.

This table gives, for each day of the week and each hour, the total number of motor vehicle thefts that occurred.

For example, on Friday at 4 AM, there were 473 motor vehicle thefts, whereas on Saturday at midnight, there were 2,050 motor vehicle thefts.

Let's save this table to a data frame so that we can use it in our visualizations.

We'll call it DayHourCounts and use the as.data.frame function, run on our table, where the first variable is the Weekday and the second variable is the Hour.

Let's take a look at the structure of the data frame we just created.

We can see that we have 168 observations-- one for each day of the week and hour pair, and three different variables.

The first variable, Var1, gives the day of the week.

The second variable, Var2, gives the hour of the day.

And the third variable, Freq for frequency, gives the total crime count.

Let's convert the second variable, Var2, to actual numbers and call it Hour, since this is the hour of the day, and it makes sense that it's numerical.

So we'll add a new variable to our data frame called Hour = as.numeric(as.character(DayHourCounts\$Var2)).

This is how we convert a factor variable to a numeric variable.

Now we're ready to create our plot.

We just need to change the group to Var1, which is the day of the week.

So we'll use the `ggplot` function where our data frame is `DayHourCounts`, and then in our aesthetic, we want the x-axis to be `Hour` this time, the y-axis to be `Freq`, and then in the `geom_line` option, like we used in the previous video, we want the aesthetic to have the group equal to `Var1`, which is the day of the week.

Go ahead and hit `Enter`.

You should see a new plot show up in the graphics window.

It has seven lines, one for each day of the week.

While this is interesting, we can't tell which line is which day, so let's change the colors of the lines to correspond to the days of the week.

To do that, just scroll up in your R console, and after `group = Var1`, add `color = Var1`.

This will make the colors of the lines correspond to the day of the week.

After that parenthesis, go ahead and type comma, and then `size = 2`.

We'll make our lines a little thicker.

Now in our plot, each line is colored corresponding to the day of the week.

This helps us see that on Saturday and Sunday, for example, the green and the teal lines, there's less motor vehicle thefts in the morning.

While we can get some information from this plot, it's still quite hard to interpret.

Seven lines is a lot.

Let's instead visualize the same information with a heat map.

To make a heat map, we'll use our data in our data frame `DayHourCounts`.

First, though, we need to fix the order of the days so that they'll show up in chronological order instead of in alphabetical order.

We'll do the same thing we did in the previous video.

So for `DayHourCounts$Var1`, which is the day of the week, we're going to use the `factor` function where the first argument is our variable, `DayHourCounts$Var1`, the second argument is `ordered = TRUE`, and the third argument

is the order we want the days of the week to show up in.

So we'll set levels, equals, and then c, and then list your days of the week.

Let's put the weekdays first and the weekends at the end.

So we'll start with Monday, and then Tuesday, then Wednesday, then Thursday, Friday, Saturday and Sunday.

Now let's make our heat map.

We'll use the ggplot function like we always do, and give our data frame name, DayHourCounts.

Then in our aesthetic, we want the x-axis to be the hour of the day, and the y-axis to be the day of the week, which is Var1.

Then we're going to add geom_tile.

This is the function we use to make a heat map.

And then in the aesthetic for our tiles, we want the fill to be equal to Freq.

This will define the colors of the rectangles in our heat map to correspond to the total crime.

You should see a heat map pop up in your graphics window.

So how do we read this?

For each hour and each day of the week, we have a rectangle in our heat map.

The color of that rectangle indicates the frequency, or the number of crimes that occur in that hour and on that day.

Our legend tells us that lighter colors correspond to more crime.

So we can see that a lot of crime happens around midnight, particularly on the weekends.

We can change the label on the legend, and get rid of the y label to make our plot a little nicer.

We can do this by just scrolling up to our previous command in our R console and then adding scale_fill_gradient.

This defines properties of the legend, and we want name = "Total MV Thefts", for total motor vehicle thefts.

Then let's add, in the theme(axis.title.y = element_blank()).

This is what you can do if you want to get rid of one of the axis labels.

Go ahead and hit Enter.

And now on our heat map, the legend is titled "Total MV Thefts" and the y-axis label is gone.

We can also change the color scheme.

We can do this by scrolling up in our R console, and going to that `scale_fill_gradient` function, the one that defines properties of our legend, and after `name = "Total MV Thefts"`, `low = "white"`, `high = "red"`.

We'll make lower values correspond to white colors and higher values correspond to red colors.

If you hit enter, a new plot should show up with different colors.

This is a common color scheme in policing.

It shows the hot spots, or the places with more crime, in red.

So now the most crime is shown by the red spots and the least crime is shown by the lighter areas.

It looks like Friday night is a pretty common time for motor vehicle thefts.

We saw something that we didn't really see in the heat map before.

It's often useful to change the color scheme depending on whether you want high values or low values to pop out, and the feeling you want the plot to portray.

In this video, we've seen how to create some new types of plots.

In the next video, we'll see how to add data to geographical maps.