In this video, we'll create a basic line plot to visualize crime trends.

Let's start by reading in our data.

We'll call it mvt for motor vehicle thefts, and use the read.csv function to read in the file mvt.csv.

We'll add the argument stringsAsFactors = FALSE, since we have a text field, and we want to make sure it's read in properly.

Let's take a look at the structure of our data using the str function.

We have over 190,000 observations of three different variables-- the date of the crime, and the location of the crime, in terms of latitude and longitude.

We want to first convert the Date variable to a format that R will recognize so that we can extract the day of the week and the hour of the day.

We can do this using the strptime function.

So we want to replace our variable, Date, with the output of the strptime function, which takes as a first argument our variable, Date, and then as a second argument the format that the date is in.

Here, we can see in the output from the str function that our format is the month slash the day slash the year, and then the hour colon minutes.

So our format equals, "%m/%d/%y %H:%M", close the parentheses, and hit Enter.

In this format, we can extract the hour and the day of the week from the Date variable, and we can add these as new variables to our data frame.

We can do this by first defining our new variable, mvt$Weekday = weekdays(mvt$Date).

Then, to add the hour, which we'll call mvt$Hour, we just take the hour variable out of Date variable.

This only exists because we converted the Date variable.

Let's take a look at the structure of our data again to see what it looks like.

Now, we have two more variables-- Weekday, which gives the day of the week, and Hour, which gives the hour of the day.

Now, we're ready to make some line plots.

Let's start by creating the line plot we saw in the previous video with just one line and a value for every day of the week.

We want to plot as that value the total number of crimes on each day of the week.

We can get this information by creating a table of the Weekday variable.

This gives the total amount of crime on each day of the week.

Let's save this table as a data frame so that we can pass it to ggplot as our data.

We'll call it WeekdayCounts, and use the as.data.frame function to convert our table to a data frame.

Let's see what this looks like with the str function.

We can see that our data frame has seven observations, one for each day of the week, and two different variables.

The first variable, called Var1, gives the name of the day of the week, and the second variable, called Freq, for frequency, gives the total amount of crime on that day of the week.

Now, we're ready to make our plot.

First, we need to load the ggplot2 package.

So we'll type library(ggplot2).

Now, we'll create our plot using the ggplot function.

So type ggplot, and then we need to give the name of our data, which is WeekdayCounts.

And then we need to define our aesthetic.

So our aesthetic should have x = Var1, since we want the day of the week on the x-axis, and y = Freq, since we want the frequency, the number of crimes, on the y-axis.

Now, we just need to add geom_line(aes(group=1)).

This just groups all of our data into one line, since we want one line in our plot.

Go ahead and hit Enter.

We can see that this is very close to the plot we want.

We have the total number of crime plotted by day of the week, but our days of the week are a little bit out of order.

We have Friday first, then Monday, then Saturday, then Sunday, etc.

What ggplot did was it put the days of the week in alphabetical order.

But we actually want the days of the week in chronological order to make this plot a bit easier to read.

We can do this by making the Var1 variable an ordered factor variable.

This signals to ggplot that the ordering is meaningful.

We can do this by using the factor function.

So let's start by typing WeekdayCounts$Var1, the variable we want to convert, and set that equal to the output of the factor function, where the first argument is our variable, WeekdayCounts$Var1, the second argument is ordered = TRUE.

This says that we want an ordered factor.

And the third argument, which is levels, should be equal to a vector of the days of the week in the order we want them to be in.

We'll use the c function to do this.

So first, in quotes, type "Sunday" -- we want Sunday first-- and then "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday".

Go ahead and close both parentheses and hit Enter.

Now, let's try our plot again by just hitting the up arrow twice and hitting Enter.

Now, this is the plot we want.

We have the total crime by day of the week with the days of the week in chronological order.

The last thing we'll want to do to our plot is just change the x- and y-axis labels, since they're not very helpful as they are now.

To do this, back in the R console, just hit the up arrow to get back to our plotting line, and then we'll add xlab("Day of the Week").

And then we'll add ylab("Total Motor Vehicle Thefts").

Now, this is the plot we were trying to generate with descriptive labels on the x- and y-axis.

In the next video, we'll add the hour of the day to our line plot, and then create a heat map.