In this video, we'll plot crime on a map of Chicago.

First, we need to install and load two new packages, the maps package and the ggmap package.

So start by installing the package maps.

So type install.packages("maps").

When the CRAN mirror window pops up, go ahead and pick a location near you.

When the package is done installing and you're back at the blinking cursor, also type install.packages("ggmap").

When that package is also done installing, load both packages using the library command.

So type library(maps), and then library(ggmap).

Now, let's load a map of Chicago into R.

We can easily do this by using the get_map function.

So we'll call it chicago = get_map(location = "chicago", zoom = 11).

Let's take a look at the map by using the ggmap function.

Now, in your R graphics window, you should see a geographical map of the city of Chicago.

Now let's plot the first 100 motor vehicle thefts in our data set on this map.

To do this, we start by typing ggmap(chicago).

This is instead of using ggplot like we've been using in the previous videos.

Then we want to add geom_point, and here, we'll define our data set to be equal to motor vehicle thefts, where we'll take the first through 100th observations, and in our aesthetic, we'll define our x-axis to be the longitude of the points and our y-axis to be the latitude of the points.

Now, in your R graphics window, you should see the map of Chicago with black points marking where the first 100 motor vehicle thefts were.

If we plotted all 190,000 motor vehicle thefts, we would just see a big black box, which wouldn't be helpful at all.

We're more interested in whether or not an area has a high amount of crime, so let's round our latitude and longitude to two digits of accuracy and create a crime counts data frame for each area.

We'll call it LatLonCounts, and use the as.data.frame function run on the table that compares the latitude and longitude rounded to two digits of accuracy.

So our first argument to table is round(mvt$Longitude, 2).

And our second argument is round(mvt$Latitude, 2).

This gives us the total crimes at every point on a grid.

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

We have 1,638 observations and three variables.

The first two variables, Var1 and Var2, are the latitude and longitude coordinates, and the third variable is the number of motor vehicle thefts that occur in that area.

Let's convert our longitude and latitude variables to numbers and call them Lat and Long.

So first, we'll define the variable in our LatLonCounts data frame, called Long, and set that equal to as.numeric, run on as.character.

Remember, this is how we convert a factor variable to a numerical variable.

And we'll give the variable, LatLonCounts$Var1.

Now let's just repeat this for latitude.

So LatLonCounts$Lat = as.numeric(as.character(LatLonCounts$Var2)).

Now, let's plot these points on our map, making the size and color of the points depend on the total number of motor vehicle thefts.

So first, again we type ggmap(chicago) + geom_point(LatLonCounts, aes(x = Long, y = Lat, color = Freq, size = Freq)).

Now, in our R graphics window, our plot should have a point for every area defined by our latitude and longitude areas, and the points have a size and color corresponding to the number of crimes in that area.

So we can see that the lighter and larger points correspond to more motor vehicle thefts.

This helps us see where in Chicago more crimes occur.

If we want to change the color scheme, we can do that too by just hitting the up arrow in our R console and then adding scale_color_gradient(low="yellow", high="red").

If you hit Enter, you should see the same plot as before, but this time, the areas with more crime are closer to red and the areas with less crime are closer to yellow.

We can also use geom_tile to make something that looks more like a traditional heat map.

To do this, we type ggmap(chicago), just like before, but now we're going to use geom_tile, where our data frame again is LatLonCounts.

And in our aesthetic, we have that the x-axis is Long, the y-axis is Lat, and then we have alpha=Freq.

This will define how to scale the colors on the heat map according to the crime counts.

Then close the parentheses and type a comma, and then type fill="red", defining our color scheme.

Close the parentheses and hit Enter.

This map takes a minute to load.

While we're waiting, let's discuss what we've done in this video.

We've created a geographical heat map, which in our case shows a visualization of the data, but it could also show the predictions of a model.

Now that our heat map is loaded, let's take a look.

In each area of Chicago, now that area is colored in red by the amount of crime there.

This looks more like a map that people use for predictive policing.

In the next video, we'll use data from the FBI to make a heat map on a map of the United States.