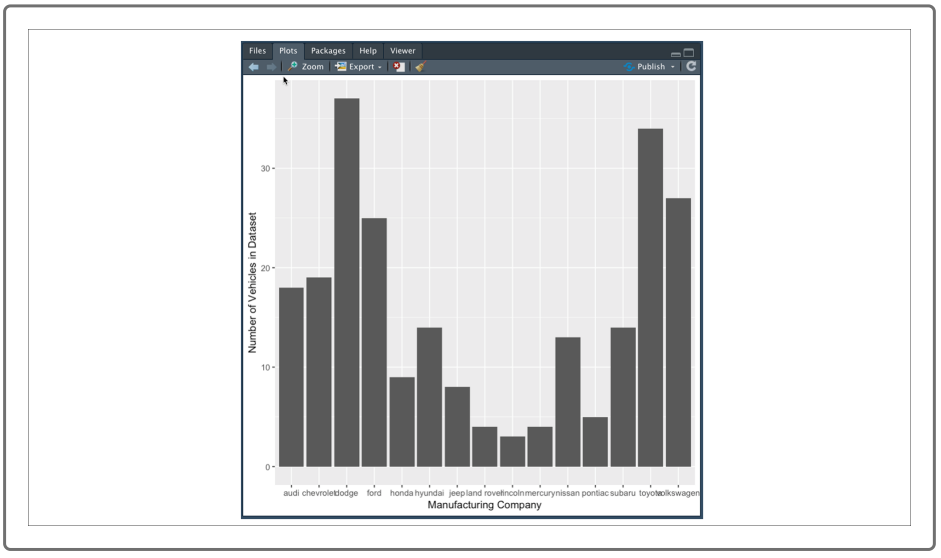
15.3.3 Add Formatting Functions

Jeremy needs to polish his bar plot before presenting to the CEO. First, he needs to format the axis titles so they're consistent, and then he needs to fix the axis labels so they don't run off the page or overlap. It's time for Jeremy to learn some formatting functions!

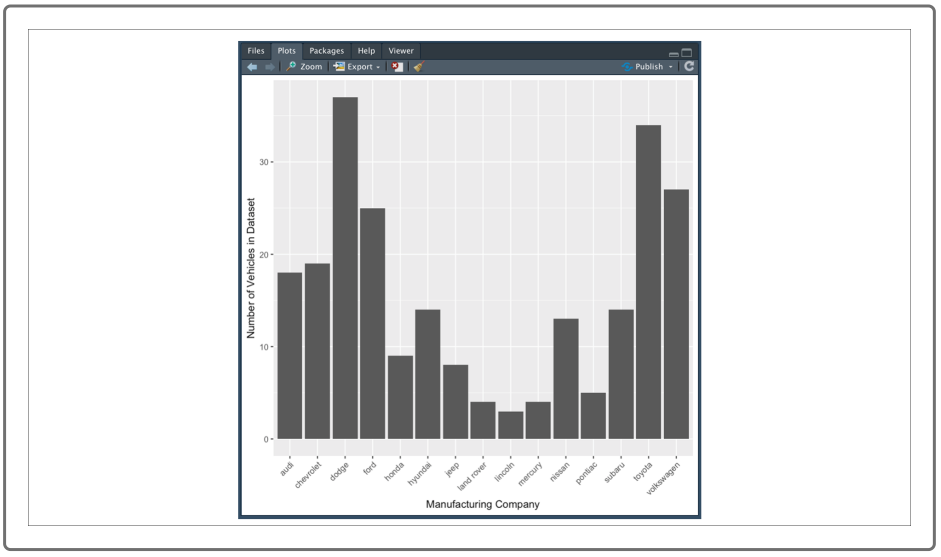
To address the issues with the plot, we'll need to add formatting functions to our plotting statement. To change the titles of our x-axis and y-axis, we can use the `xlab()` and `ylab()` functions, respectively:

```
> plt + geom_col() + xlab("Manufacturing Company") + ylab("Number of Vehicle
```



For our figure, rotate the x-axis labels 45 degrees so they no longer overlap. Our new plotting statement would be as follows, using a "+" sign at the end of the first line to indicate to the interpreter that the code continues onto the next line (note that your CLI prompt character will change from ">" to "+" after the first line to indicate that it expects additional input):

```
> plt + geom_col() + xlab("Manufacturing Company") + ylab("Number of Vehicle
>theme(axis.text.x=element_text(angle=45,hjust=1)) #rotate the x-axis label
```



CAUTION

> Unfortunately, rotating and adjusting the axis labels in ggplot2 is not as straightforward as changing axis titles. Due to the amount of customizability in ggplot2, making small adjustments such as rotating text requires very specific values to be changed in nested functions. Thankfully, there is plenty of **external documentation (http://www.cookbook-r.com/Graphs/Axes_(ggplot2)/)** and **Stack Overflow support (https://stackoverflow.com/questions/1330989/rotating-and-spacing-axis-labels-in-ggplot2)** that addresses these exact use cases, so finding help on how to tweak your ggplot2 visualizations requires only a basic Google search.

In this case, we set the angle argument of our `element_text()` function to 45 degrees and our `hjust` argument to 1. The `hjust` argument tells `ggplot` that our rotated labels should be aligned horizontally to our tick marks.

Similarly, if we want to adjust our y-axis labels, we would do so by using the `axis.text.y` argument of the `theme()` function. Now that we have adjusted our axis labels and titles, our figure is far easier to read and ready for print. Now it's time to generate our line plots!