Intro to R: Data Visualization Lab

Hannah Bravo De Rueda

2023-09-19

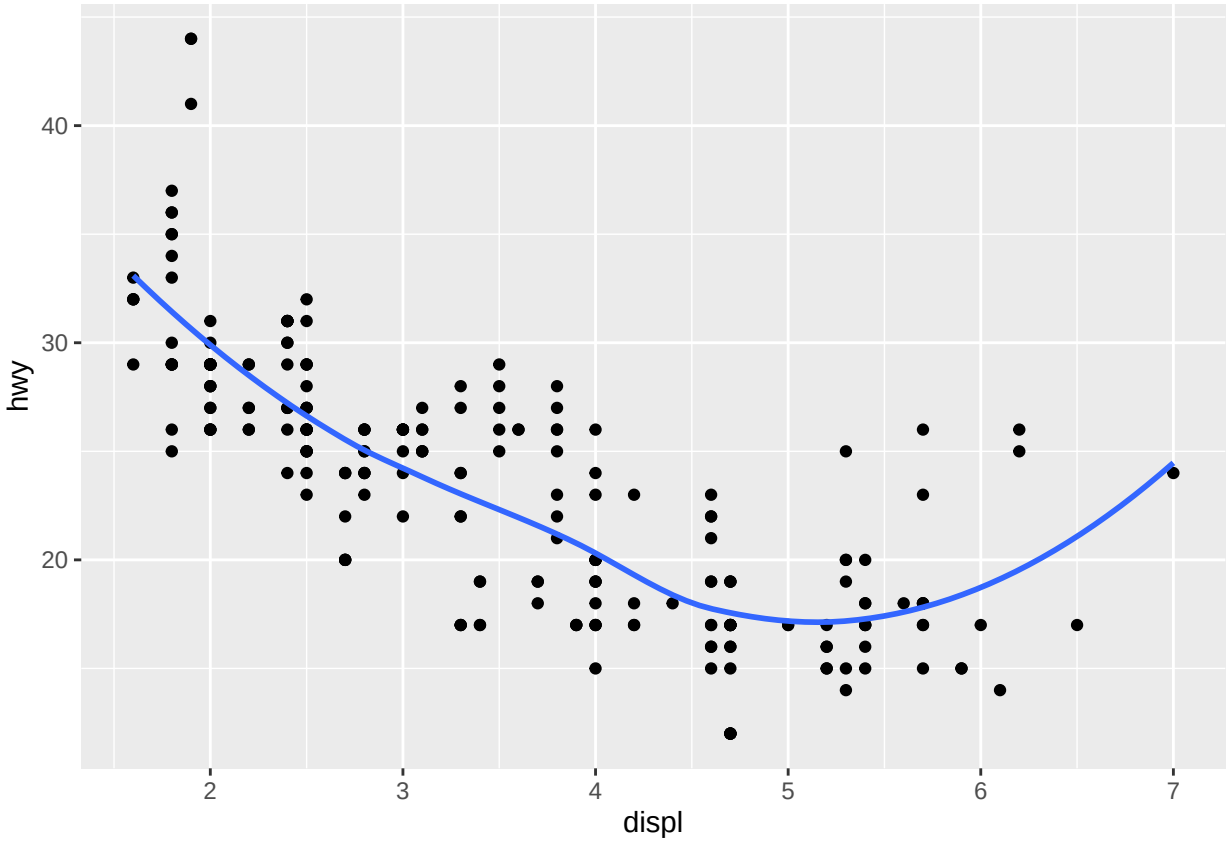
Creating Visualizations using ggplot2

Question 1: Let's recreate a plot using the mpg dataset.

The plot is a scatterplot with a trendline going through the points. 'Displ' is mapped to the x-axis, and 'hwy' to the y-axis. Let's give it a try!

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point() + geom_smooth(se = FALSE)
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



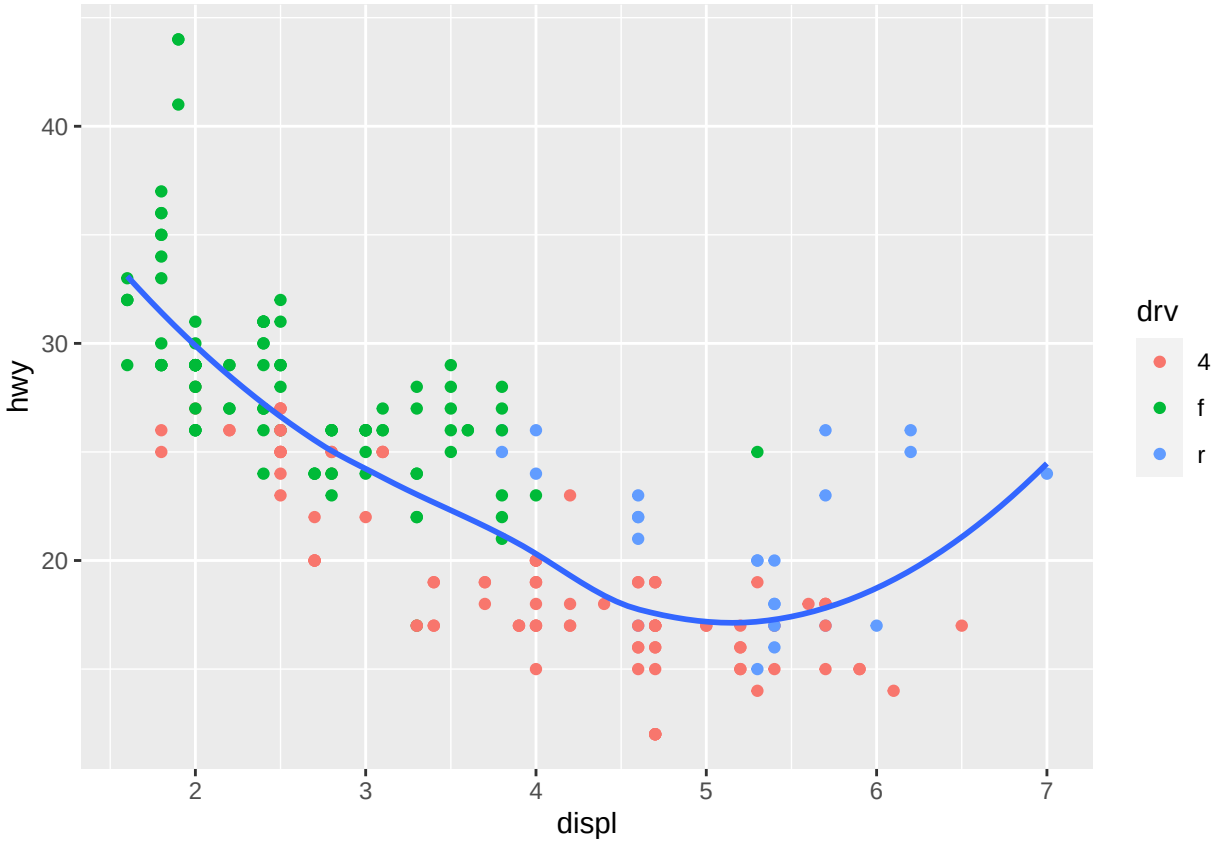
Yayy! Came out perfect.

Question 2:

We're going to work off the last plot, but now map a third variable 'drv' to color.

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point(aes(color = drv)) + geom_smooth(se = FALSE)
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

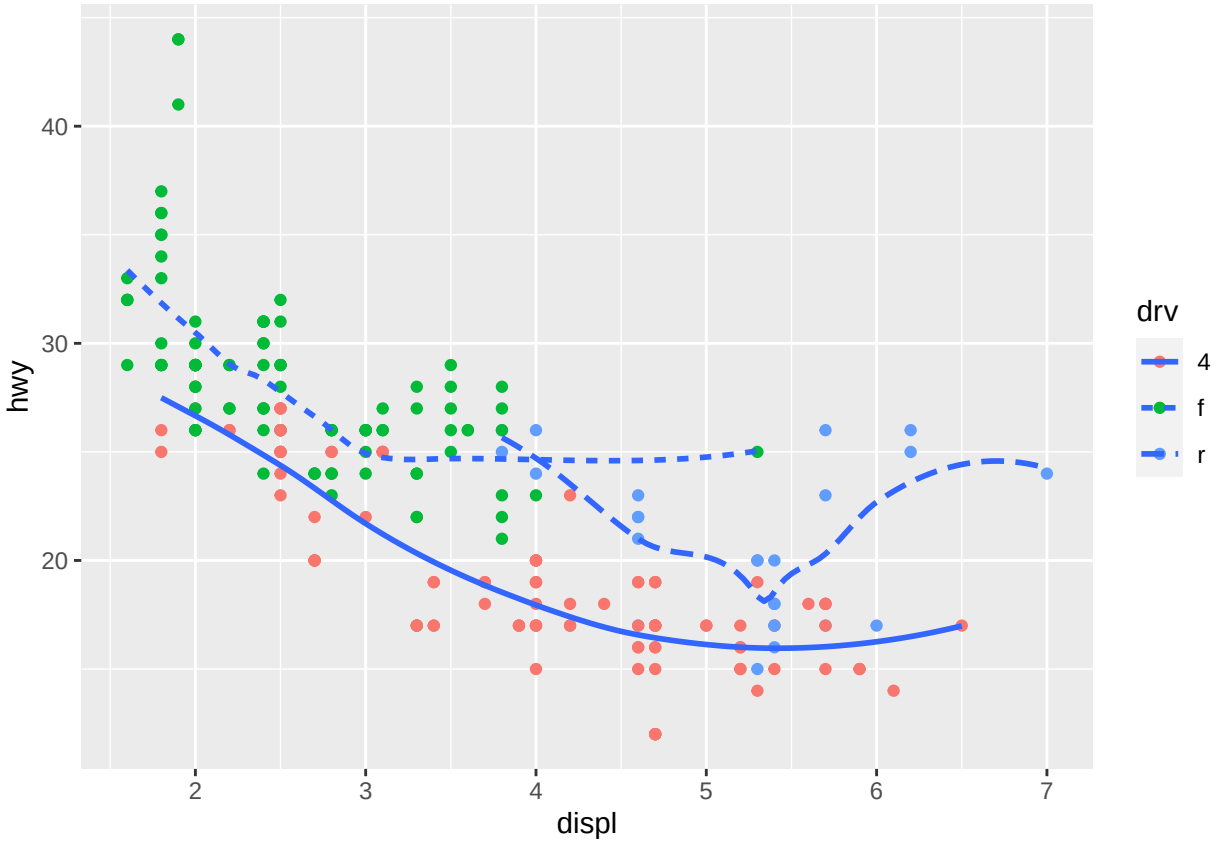


Awesome! I had to make sure to add the color specification to ONLY the geom_plot section.

Question 3: Next, we're going to continue building off the previous graph, but now we're going to differentiate the trendline across 'drv' as well.

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point(aes(color = drv)) + geom_smooth(aes(linetype = drv), se = FALSE)
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

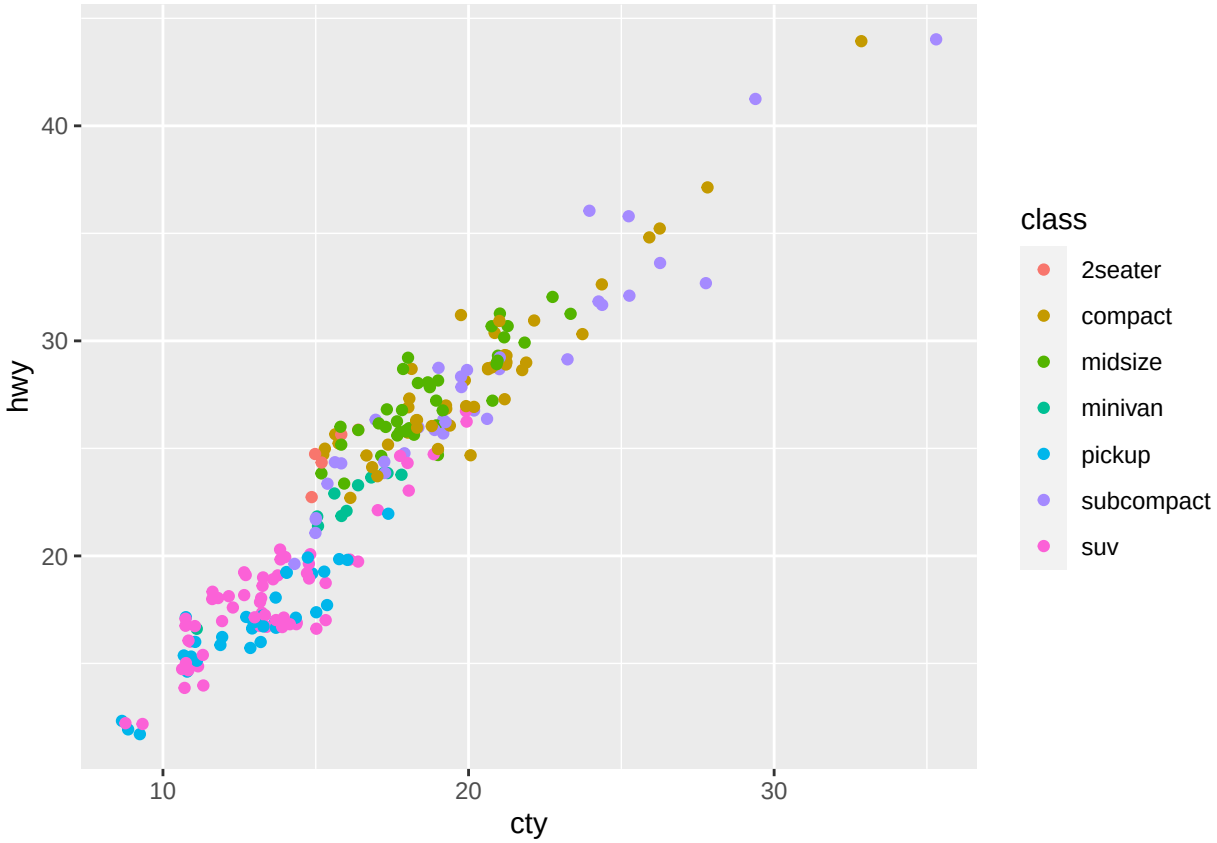


Cool! So far, everything is working!

Question 4: We're going to continue working with the mpg dataset, but we're going to move onto a new visualization now.

This new plot is still a scatterplot, with 'cty' mapped to the x-axis, 'hwy' mapped to the y-axis, and 'class' mapped to color. Let's give it a try!

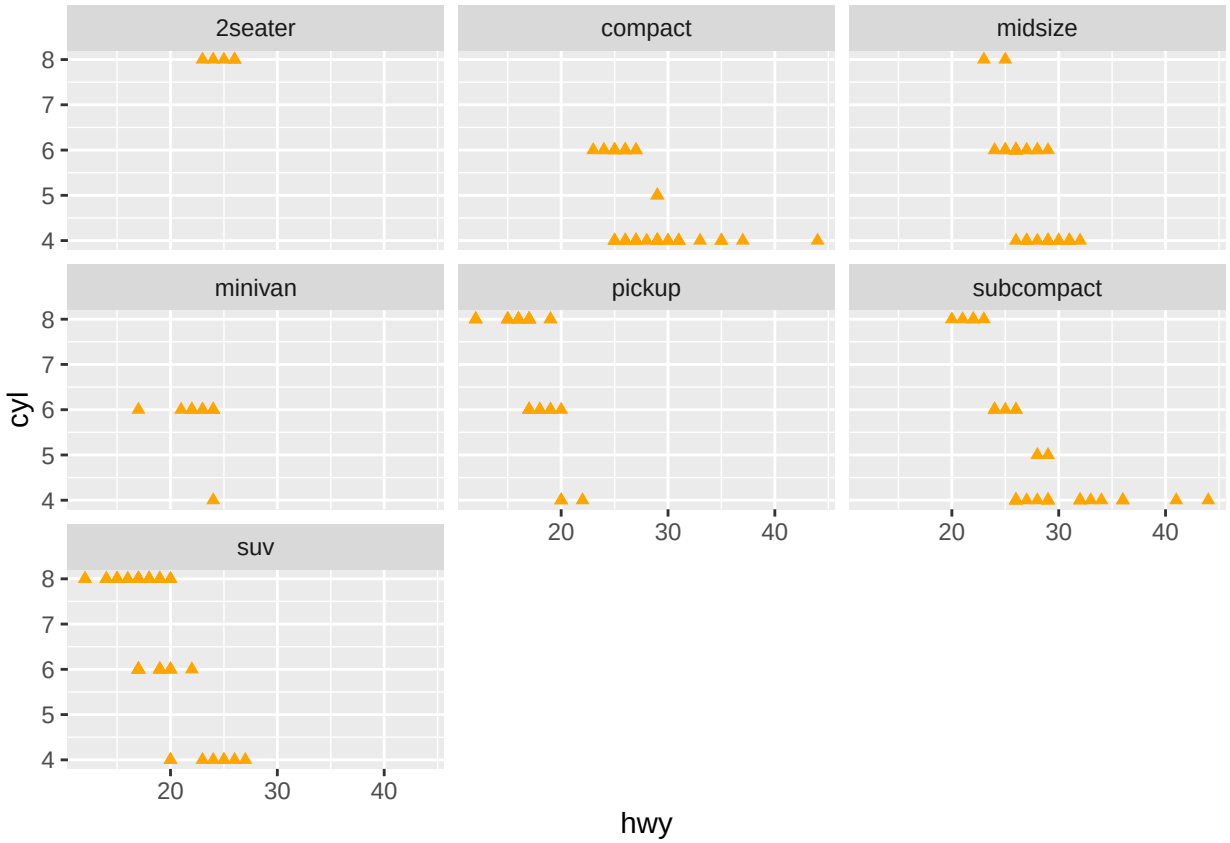
```
ggplot(data = mpg, mapping = aes(x = cty, y = hwy, color = class)) +
  geom_point(position = "jitter")
```



Awesome! I originally plotted it as a scatterplot, but since both variables are not continuous, so it graphed various points over one another and didn't show ALL of the data points. Using geom_jitter(), kind of spaced some out the points out from one another, showing much more of the data. Looking back, you could change geom_point() to geom_jitter(), OR you could add the argument 'position = "jitter"' to geom_point(). They both produce the same plot.

Question 5: Okay, now we're going to distinguish each class as it's own facet, rather than using color, and we're going to flip the coordinates. Let's see what we get!

```
ggplot(data = mpg, mapping = aes(x = hwy, y = cyl)) +
  geom_point(color = "orange", shape = 17, show.legend = FALSE) + facet_wrap(~class)
```



Yayy!! That took a bit longer, but we got there. I was originally including the color and shape parameters in the 'aes' which was a mistake.