Lecture 6: Data Visualization with `ggplot2`

STAT 450, Fall 2021

The lecture introduces `ggplot2`, a modern R package for data visualization. Last week we discussed graphics in base R, which is the original plotting system for R. There are pros and cons to each approach – base R graphics tend to be more customizable, while `ggplot2` graphics tend to look nicer without many adjustments. `ggplot2` also has advantages when dealing with categorical data. For the rest of the semester we will focus on the `ggplot2` approach to graphics.

`ggplot2` is part the `tidyverse`, which is a collection of R packages designed for data science. To install the `tidyverse` run the following command in the console:

```
install.packages("tidyverse")
```

You only need to install this package once on your computer. Note that if you are using R Studio Cloud, the `tidyverse` should already be installed.

To load `ggplot2` into your current R session run the following command:

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.3     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

This command needs to be run during each R session when you want to use `ggplot2` or other `tidyverse` packages.

Alternatively, you can just load `ggplot2`, without the other `tidyverse` packages:

```
library(ggplot2)
```

The `mpg` data frame

The `mpg` data frame is part of the `ggplot2` package. The data set is stored as a `tibble`, which is how data frames are represented in the `tidyverse`. A nice feature of tibbles is that when you type the name of data frame, only the first 10 rows and all columns that fit on the screen are displayed. The type of each column (variable) is also shown under its name.

From *R for Data Science*: "A data frame is a rectangular collection variables (in the columns) and observations (in the rows). `mpg` contains observations collected by the US EPA on 38 car models."

```
mpg
```

```
## # A tibble: 234 x 11
##    manufacturer model      displ  year   cyl trans drv     cty   hwy fl    class
##    <chr>        <chr>      <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
##  1 audi         a4           1.8  1999     4 auto~ f        18    29 p     comp~
##  2 audi         a4           1.8  1999     4 manu~ f        21    29 p     comp~
##  3 audi         a4           2    2008     4 manu~ f        20    31 p     comp~
##  4 audi         a4           2    2008     4 auto~ f        21    30 p     comp~
##  5 audi         a4           2.8  1999     6 auto~ f        16    26 p     comp~
##  6 audi         a4           2.8  1999     6 manu~ f        18    26 p     comp~
##  7 audi         a4           3.1  2008     6 auto~ f        18    27 p     comp~
##  8 audi         a4 quattro   1.8  1999     4 manu~ 4        18    26 p     comp~
##  9 audi         a4 quattro   1.8  1999     4 auto~ 4        16    25 p     comp~
## 10 audi         a4 quattro   2    2008     4 manu~ 4        20    28 p     comp~
## # ... with 224 more rows
```

To learn more about this data set, read the documentation in the help menu:

```
help(mpg)
```

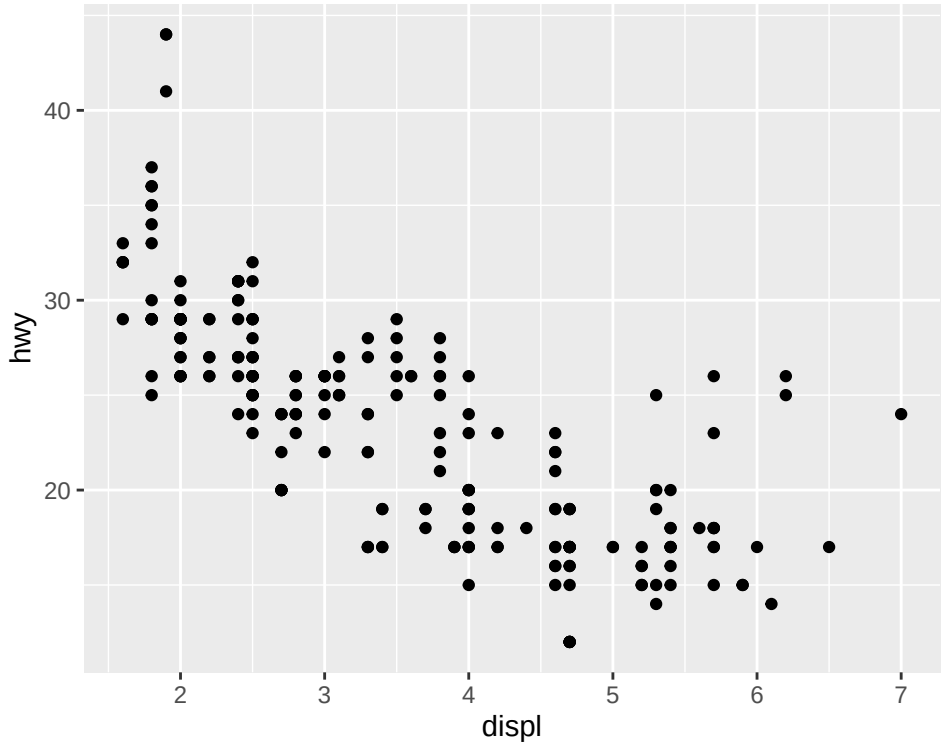
We will focus on the following variables:

- `displ`: a car's engine size, in liters
- `hwy`: highway miles per gallon
- `class`: the type of car

Creating a ggplot

Run the following code to make a scatter plot with `displ` on the x-axis and `hwy` on the y-axis:

```
ggplot(data = mpg) +
  geom_point(aes(x = displ, y = hwy))
```



There are two steps to create this scatter plot. First, `ggplot()` initializes the plot and specifies the `mpg` data frame used for the plot. Then `geom_point()` adds the points to the scatter plot, with `displ` on the x-axis and `hwy` on the y-axis.

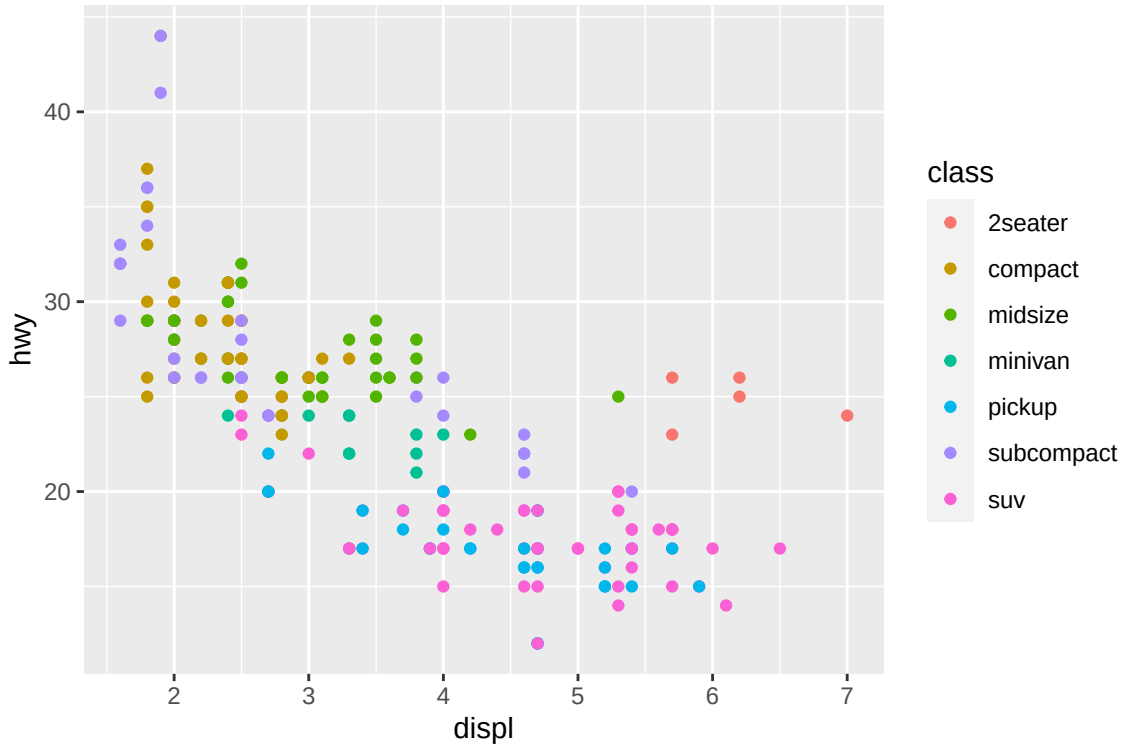
Exercises:

1. Run `ggplot(data = mpg)`. What do you see?
2. Make a scatter plot with `cyl` on the x-axis and `hwy` and the y-axis.

Coloring points

Here we create a scatter plot with points colored according to the `class` of each car.

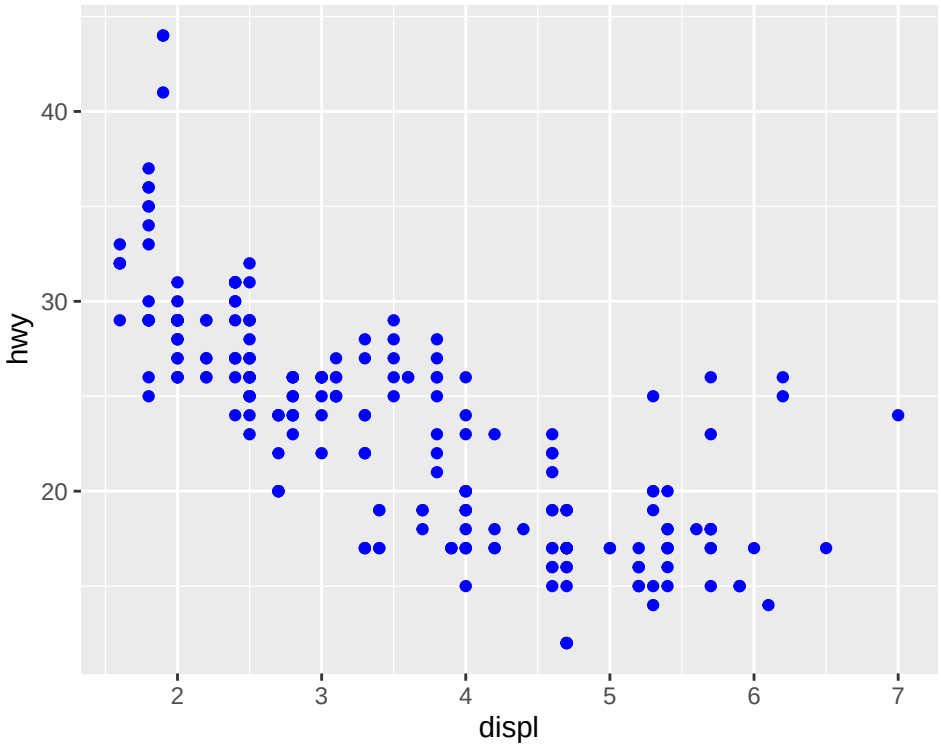
```
ggplot(data = mpg) +
  geom_point(aes(x = displ, y = hwy, color = class))
```



`aes()` stands for aesthetics. Conceptually, `aes()` specifies the mapping of the variables to the different aesthetics, or visual properties of the plot. In this example, `displ` is mapped to the x-axis, `hwy` is mapped to the y-axis, and `class` is mapped to the point color.

To make all the points blue:

```
ggplot(data = mpg) +
  geom_point(aes(x = displ, y = hwy), color = "blue")
```



Here `color` goes outside of `aes()`. This is because the color blue does not convey any information about a particular variable.

Exercises:

3. What's gone wrong with this code? Why are the points not blue?

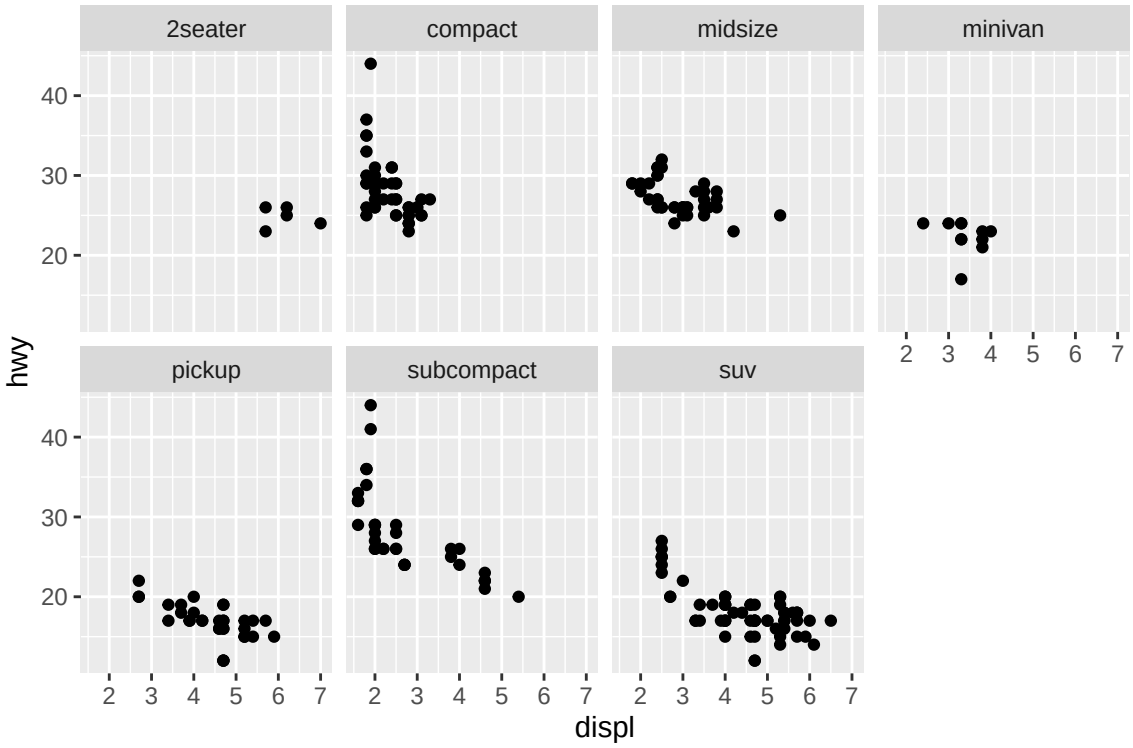
```
ggplot(data = mpg) +
  geom_point(aes(x = displ, y = hwy, color = "blue"))
```

4. Make a scatter plot of `hwy` versus `displ`. Map the categorical variable `drv` to the color and shape of the points. Type `help(mpg)` to read the description of the `drv` variable in the help menu.

Facets

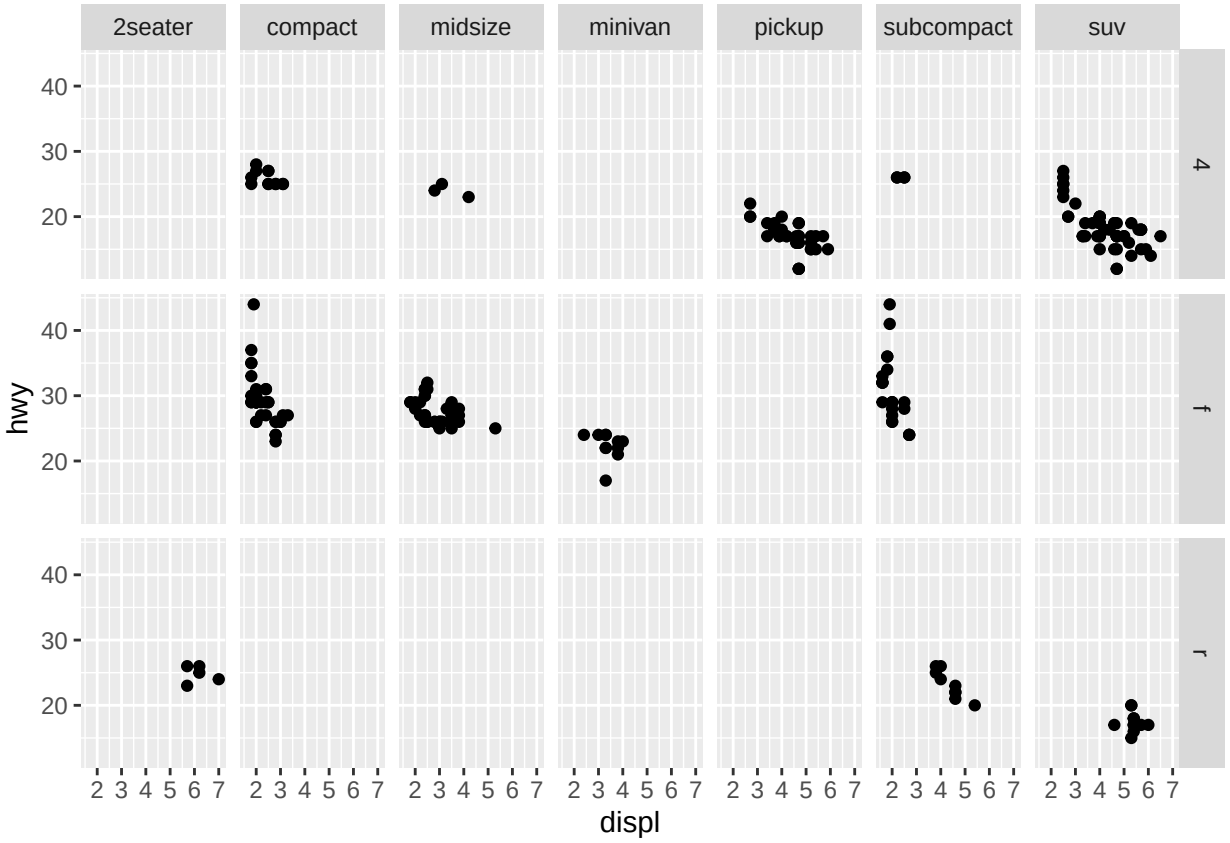
We can use `facet_wrap()` to split the plot into facets, subplots that each display one subset of the data. For example, the code below creates a scatter plot of `hwy` versus `displ` for each category of `class`.

```
ggplot(data = mpg) +
  geom_point(aes(x = displ, y = hwy)) +
  facet_wrap(~ class, nrow = 2)
```



We can use `facet_grid()` to facet the plot using a combination of two variables.

```
ggplot(data = mpg) +
  geom_point(aes(x = displ, y = hwy)) +
  facet_grid(drv ~ class)
```



Exercises:

5. Use `facet_wrap()` to create 3 facets with scatter plots of `hwy` versus `displ` for each of category of `drv`.

6. Run the code below:

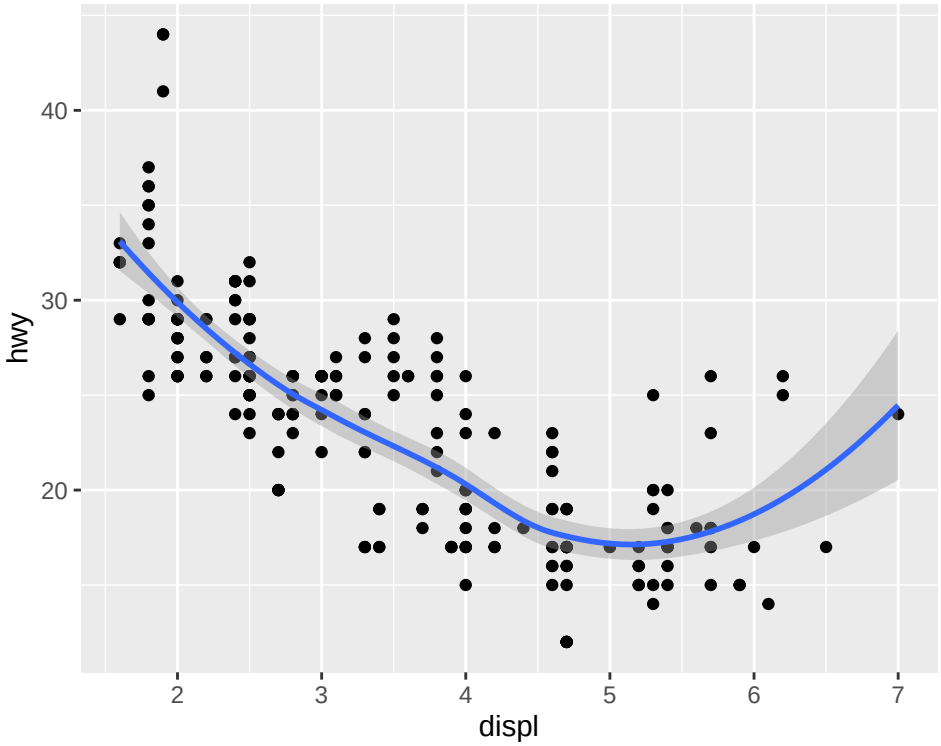
```
ggplot(data = mpg) +
  geom_point(aes(x = displ, y = hwy, color = drv)) +
  facet_wrap(~ class, nrow = 2)
```

How does this visualization compare with plot created using `facet_grid()`? Which plot do you prefer?

Scatter plot smoothing

Use `geom_smooth()` to add a smooth line that displays the average trend in the scatter plot.

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

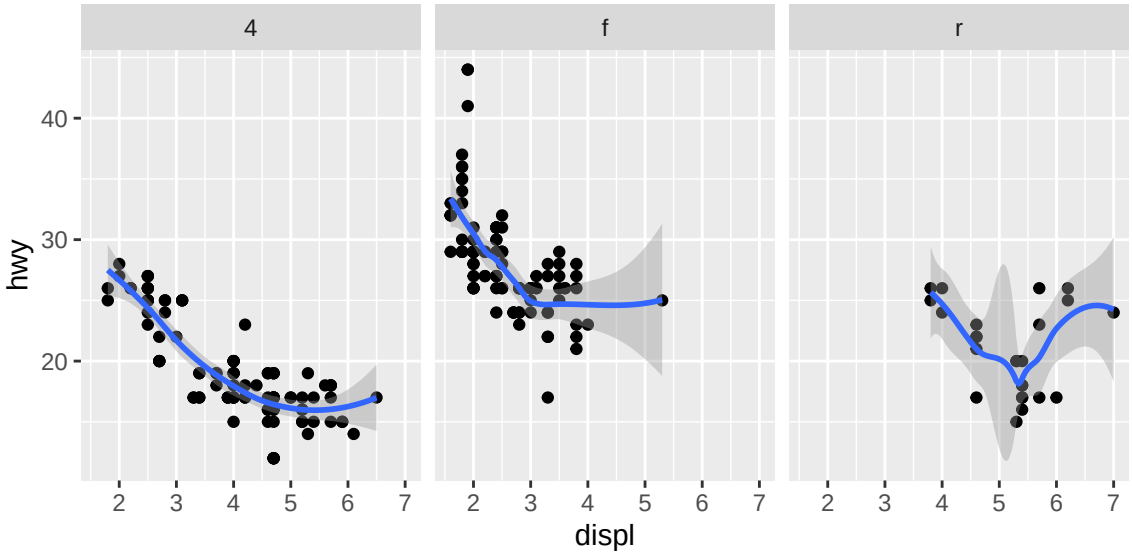


We could have also written this code in the following way, but this coding is more redundant.

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

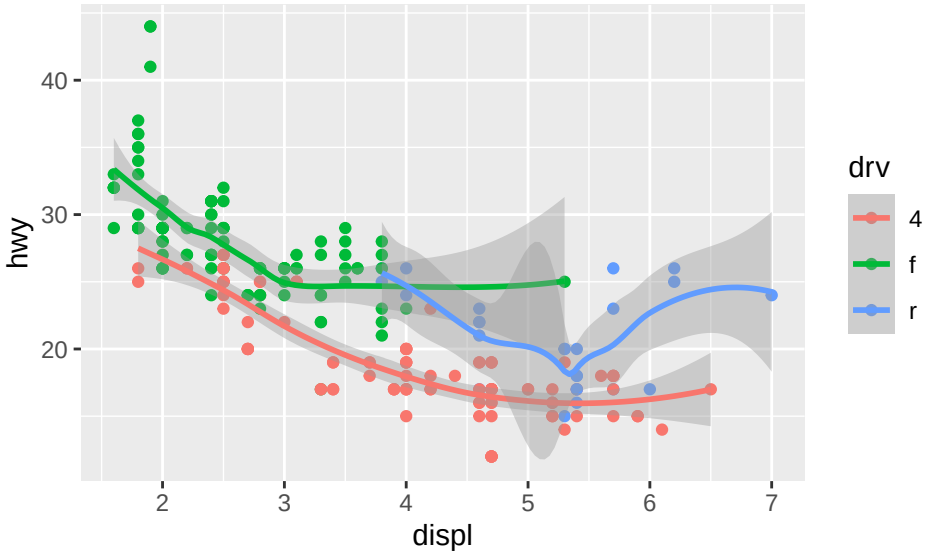
This code adds a smooth trend line to each subplot when using `facet_wrap()`

```
ggplot(data = mpg, aes(x = displ, y = hwy)) +
  geom_point() +
  geom_smooth() +
  facet_wrap(~ drv)
```



This code colors both the points and trend lines according to the categories of `drv`.

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



Exercises:

7. What is wrong with the syntax in the following code:

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

8. Run the following code. What does the argument `se = FALSE` of `geom_smooth()` do?

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

9. Recreate the R code necessary to make the following plot. (If you get a warning message just ignore it.)