Facets and Geoms

Alireza Mostafizi

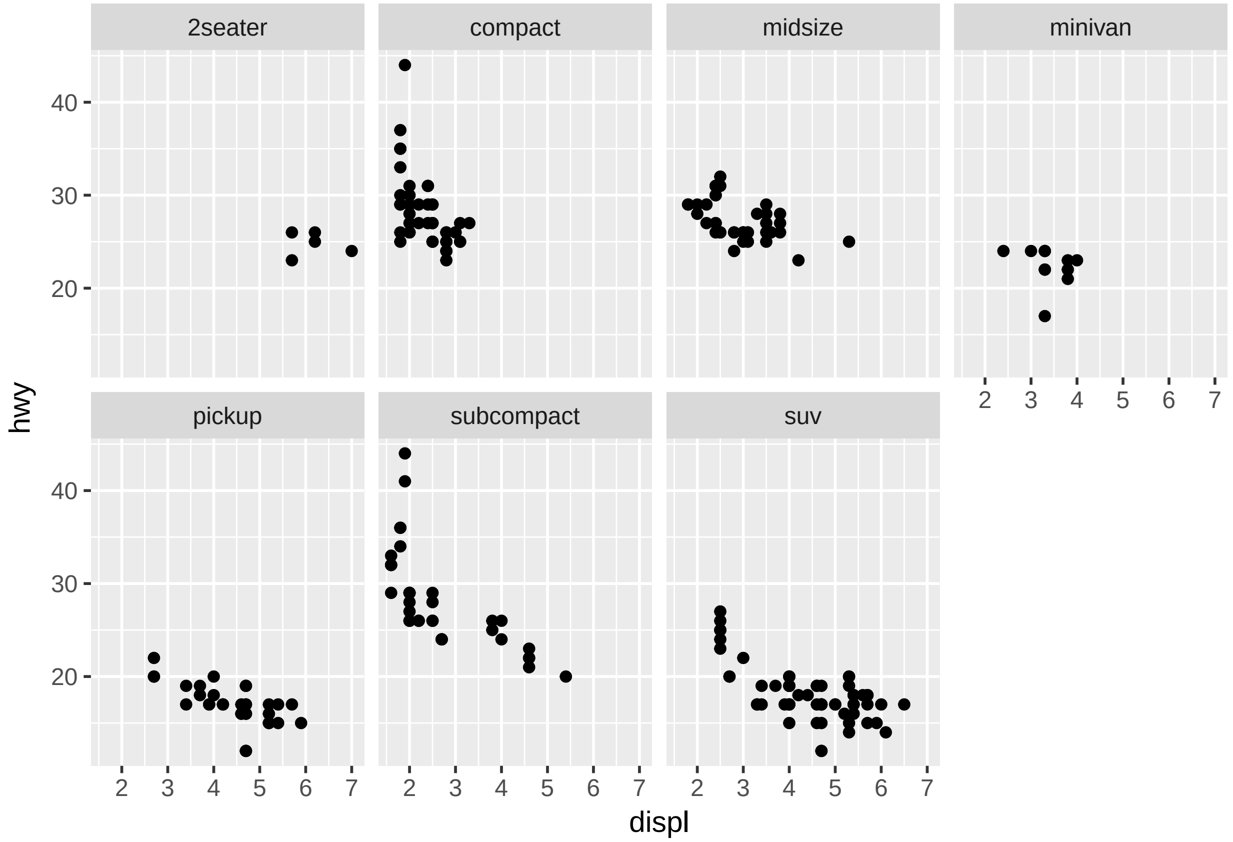
12 April 2018

3.5 Facetss

Facets are a way to break the data into subsets based on a categorical variable and study each subset in individual plots. To do this, you have to use `facet_wrap()` as the following example.

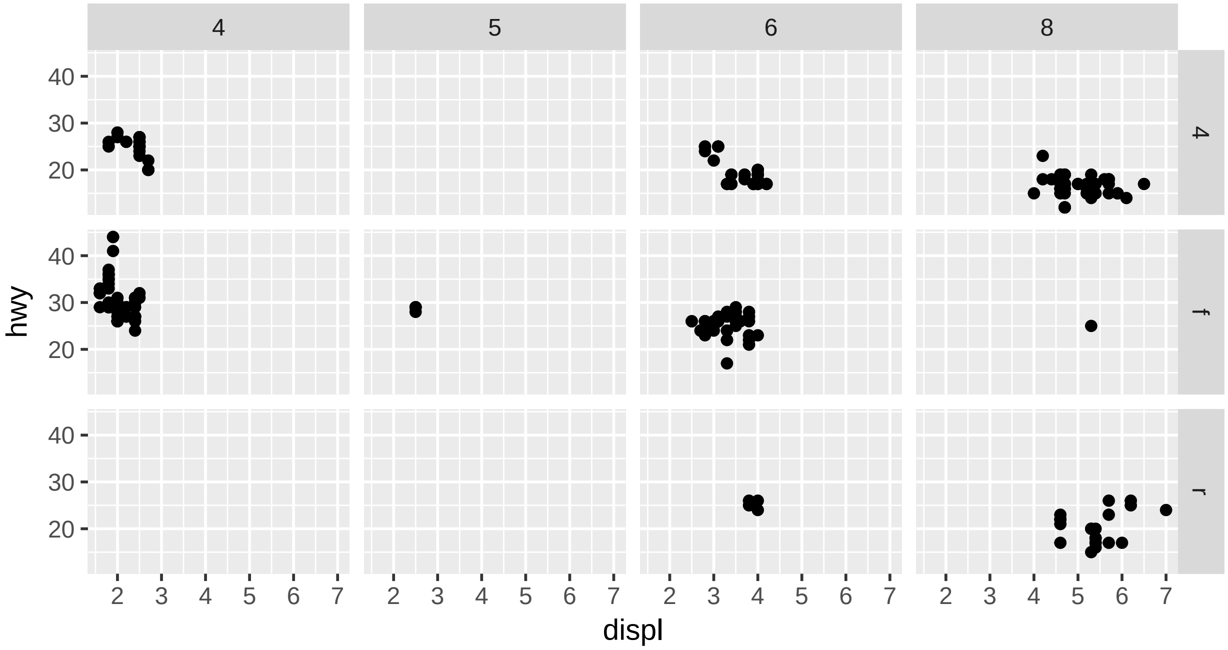
```
library(tidyverse, ggplot2)
```

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



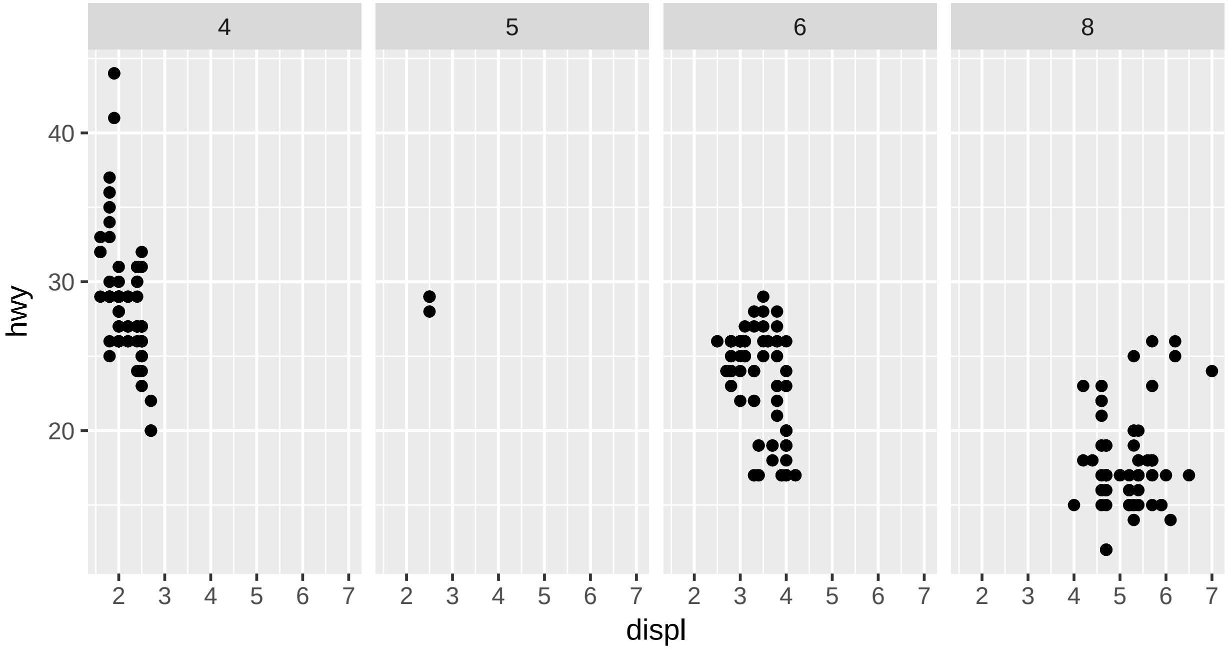
In case of faceting the results with regard to two categorical variables, `facet_grid()` needs to be used.

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



In case you prefer not to facet on one dimension, you can use `.` for the columns or rows as following,

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_grid(. ~ cyl)
```



Which is exactly equal to,

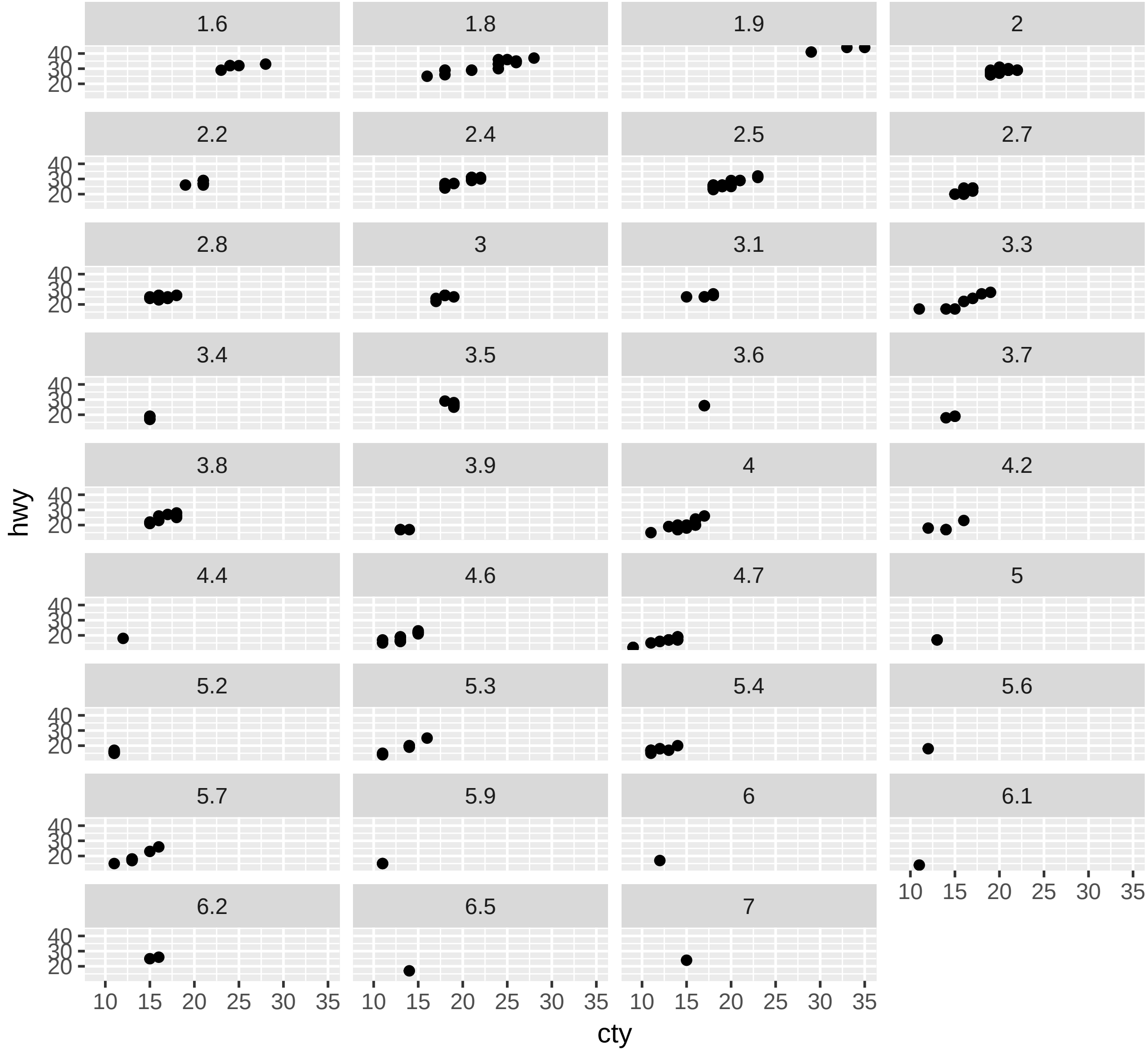
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_wrap(~ cyl, nrow = 1)
```

3.5.1 Exercises

1. What happens if you facet on a continuous variable?

This does not throw an error, but the data is broken into many subsets and shown in many plots that practically is useless and very hard to interpret. For instance, let's look at the relationship between highway and city fuel efficinecy faceted on engine size.

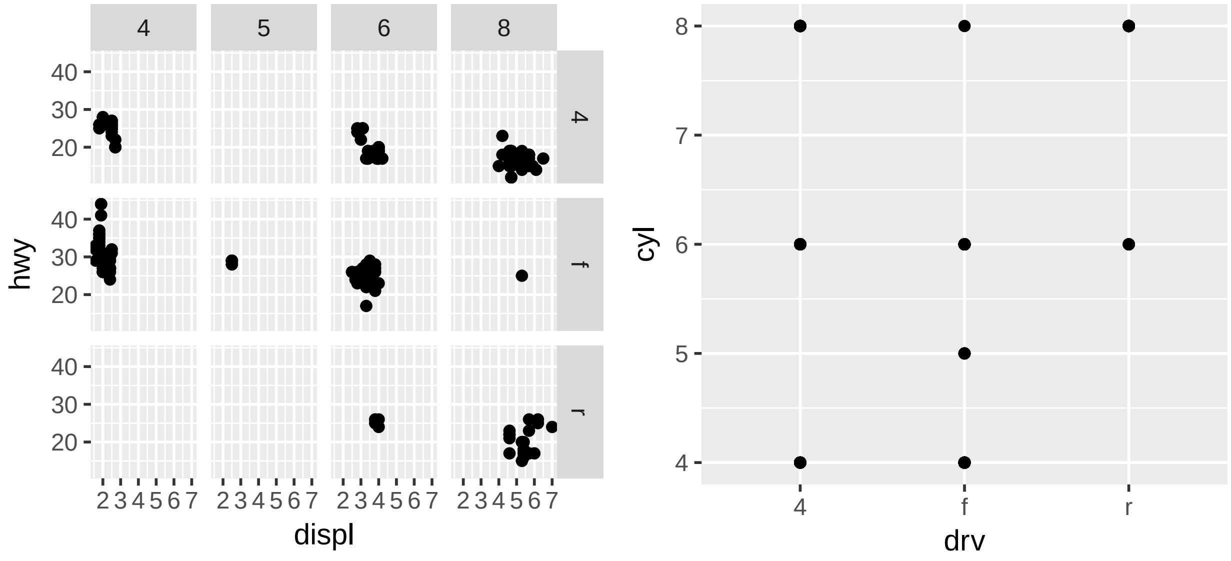
```r
ggplot(data = mpg) +
  geom_point(mapping = aes(x = cty, y = hwy)) +
  facet_wrap(~ displ, ncol = 4)
```



2. What do the empty cells in plot with `facet_grid(drv ~ cyl)` mean? How do they relate to this plot?

The empty plot means that there has been no data at the intersection of the two faceted variables. For instance, Looking at the left figure below, it can be seen that there is no data at the intersections of `(cyl = 5, drv = "4")`, `(cyl = 5, drv = "r")`, and `(cyl = 4, drv = "r")`. The same idea can be taken from the scatter plot of `cyl` vs. `drv`.

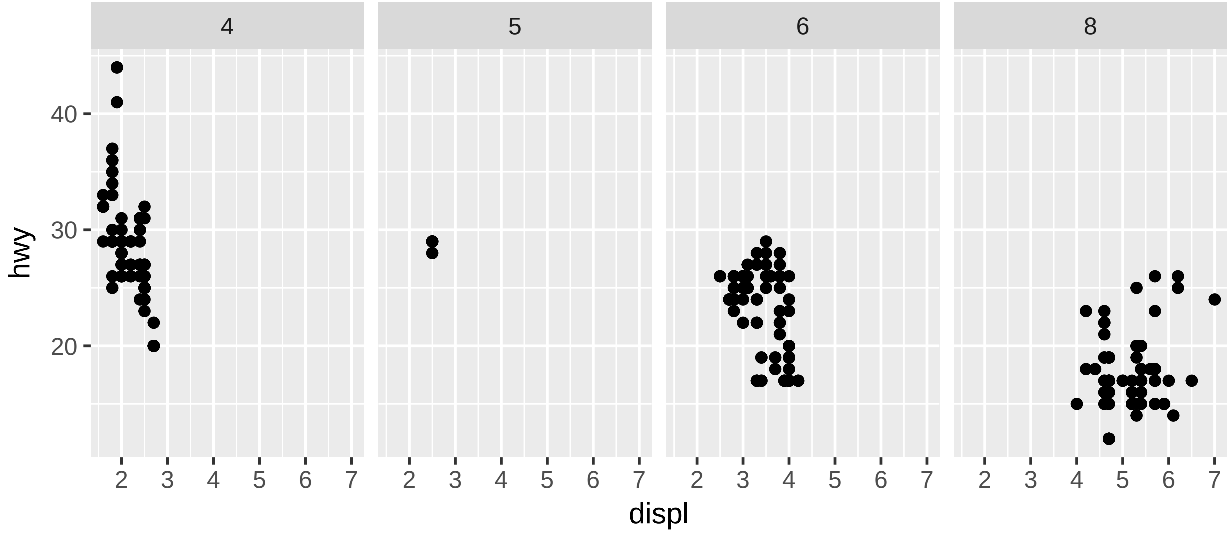
```
library(gridExtra)
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_grid(drv ~ cyl) -> p1

ggplot(data = mpg) +
  geom_point(mapping = aes(x = drv, y = cyl)) -> p2

grid.arrange(p1, p2, ncol = 2)
```



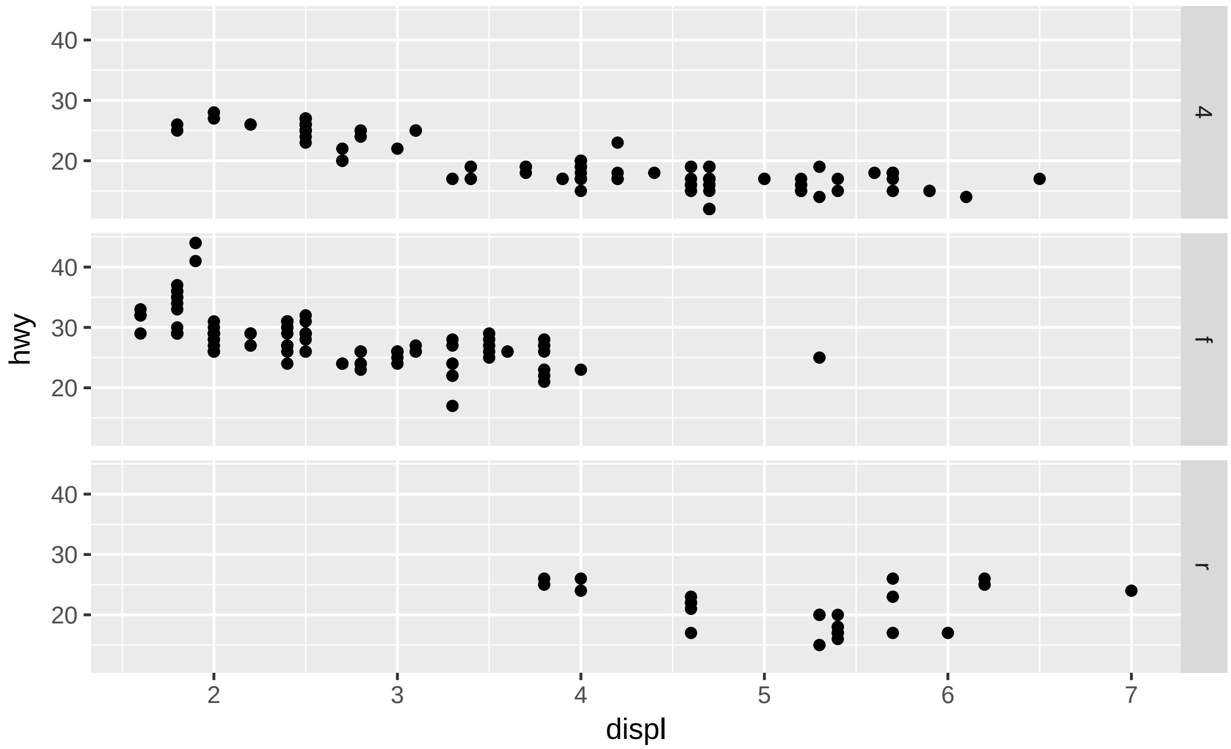
3. What plots does the following code make? What does `.` do?

`.` makes the plot not facet on either columns (`X ~ .`) or rows (`. ~ X`). For instance, the following plots are not faceted on rows and on columns respectively.

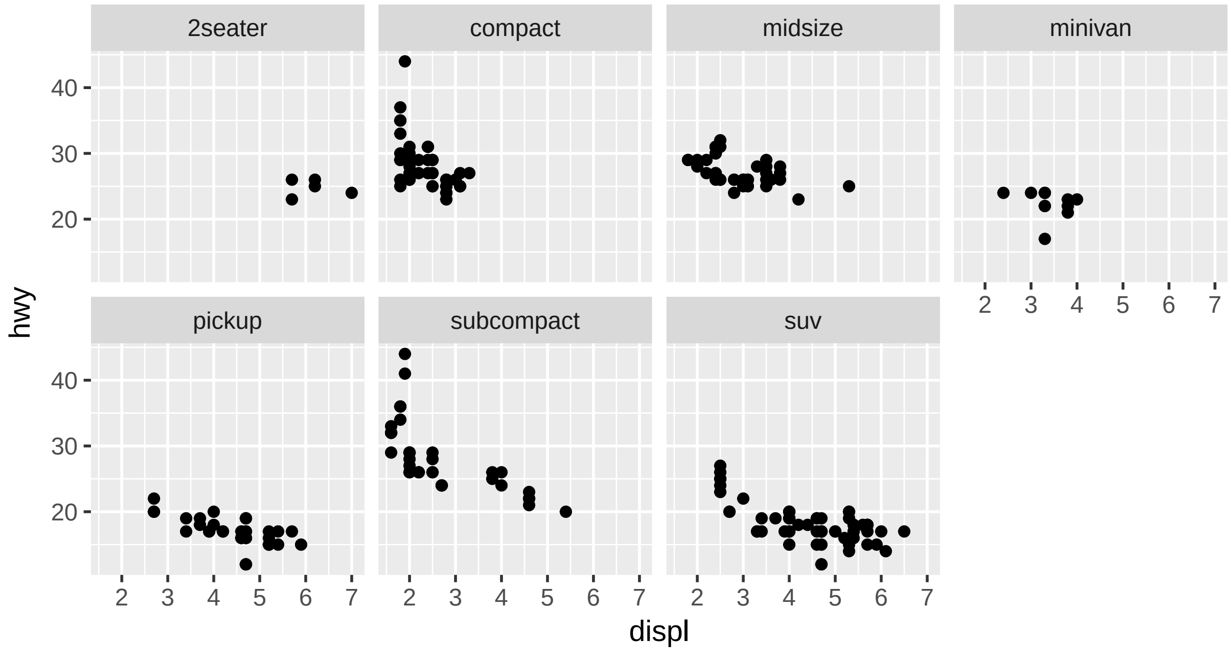
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_grid(. ~ cyl)
```



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



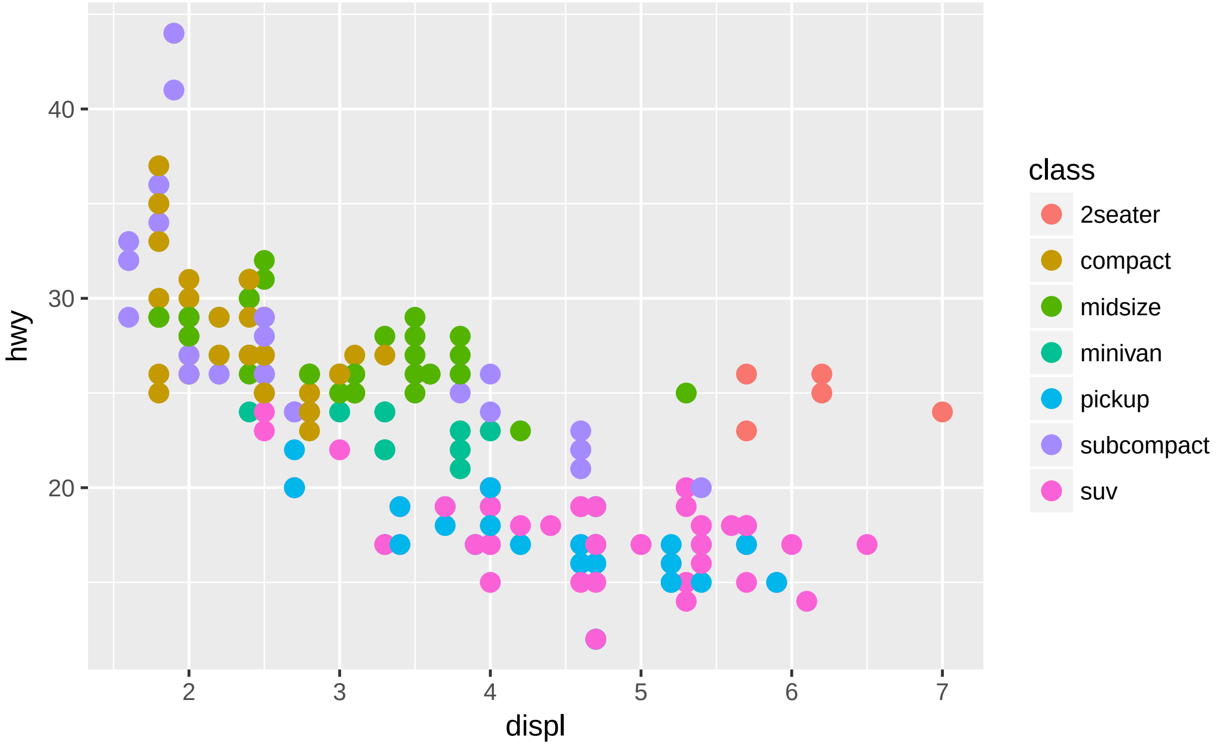
4. Take the first faceted plot in this section:



What are the advantages to using faceting instead of the colour aesthetic? What are the disadvantages? How might the balance change if you had a larger dataset?

One of the advantages of faceting over using just the color aesthetic is that it's much easier to see the trend of the subdata. Another advantage is that faceting works well even in black and white, but the colored plot is almost useless in black and white. On the other hand though, faceting takes much more space. In addition, the comparison between the sub datasets are made easier in the colored plot as the data points are plotted on the same set of axis. Another advantage of faceting technique is that, in case there are two many categories, differentiating the groups by color is not perceived well. Therefore, if either there are two many categories, or there are too many data points for each category, it's not easy to infer the data plotted in colors and all together, so faceting works better. However, in case that there are not too many groups (say less than 7), and there are not too many data points to overlap eachother, the colored scatter plot works just fine.

```
ggplot((dat = mpg)) +
  geom_point(mapping = aes(x = displ, y = hwy, color = class), size = 3)
```



5. Read `?facet_wrap`. What does `nrow` do? What does `ncol` do? What other options control the layout of the individual panels? Why doesn't `facet_grid()` have `nrow` and `ncol` argument?

```
?facet_wrap()
```

`nrow` and `ncol` control the number of rows and columns that the plots are shown at. `scale` controls the axis scale of the individual plots to be either free or fixed. `as.table` controls the layout of the plots, with the highest value beign the at the bottom-right or the top-right. Also, `dir` governs the direction of the layout.

```
?facet_grid()
```

`facet_grid()` does not have `nrow` and `ncol` arguments since the number of columns and rows are defined automatically be the number of unique values of the faceted variables associated with either rows or columns.

6. When using `facet_grid()` you should usually put the variable with more unique levels in the columns. Why?

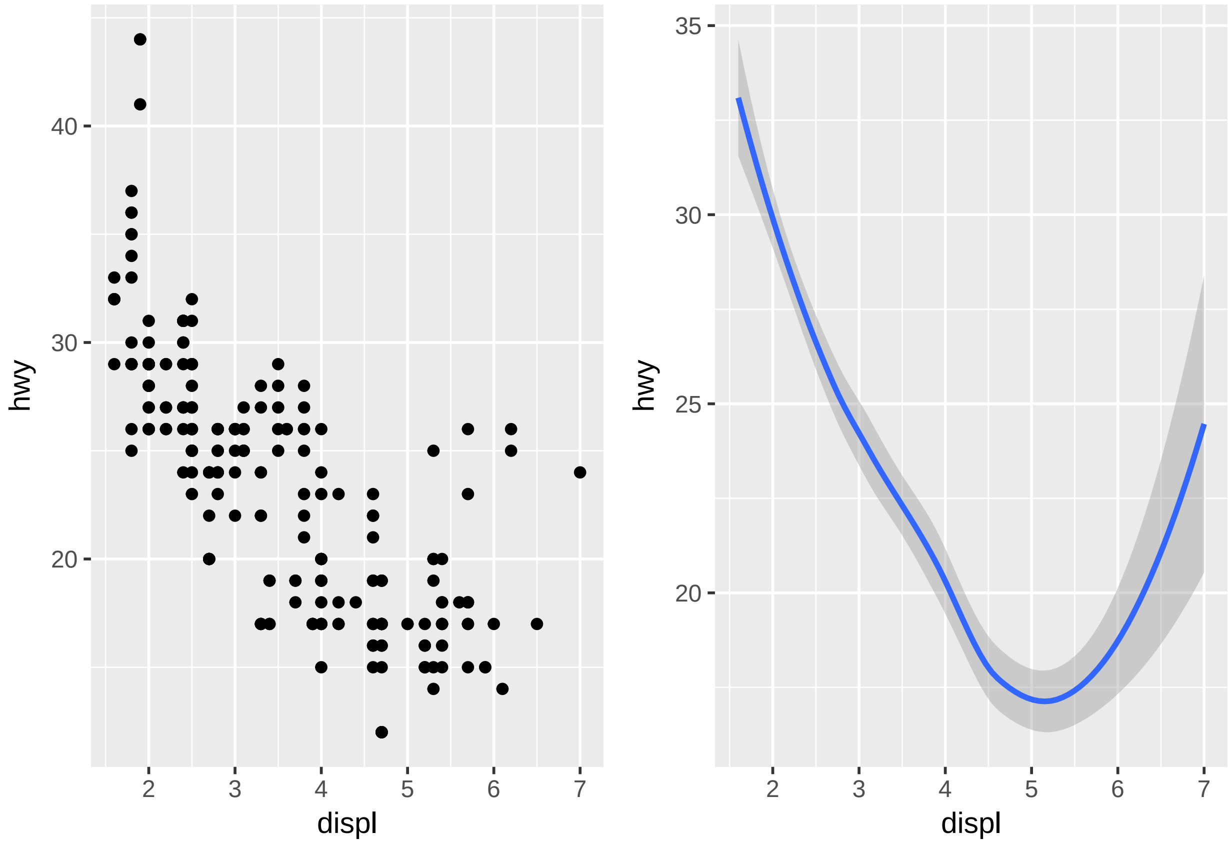
Because otherwise the entire plot will be too long vertically.

3.6 Geometric objects

This section explains different types of geometric objects, **geoms**. For instance, below figures both represent the same dataset, but with different geoms, left one with the point geom, and the right one with the smooth geom.

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) -> p1

ggplot(data = mpg) +
  geom_smooth(mapping = aes(x = displ, y = hwy)) -> p2

grid.arrange(p1, p2, ncol = 2)
```

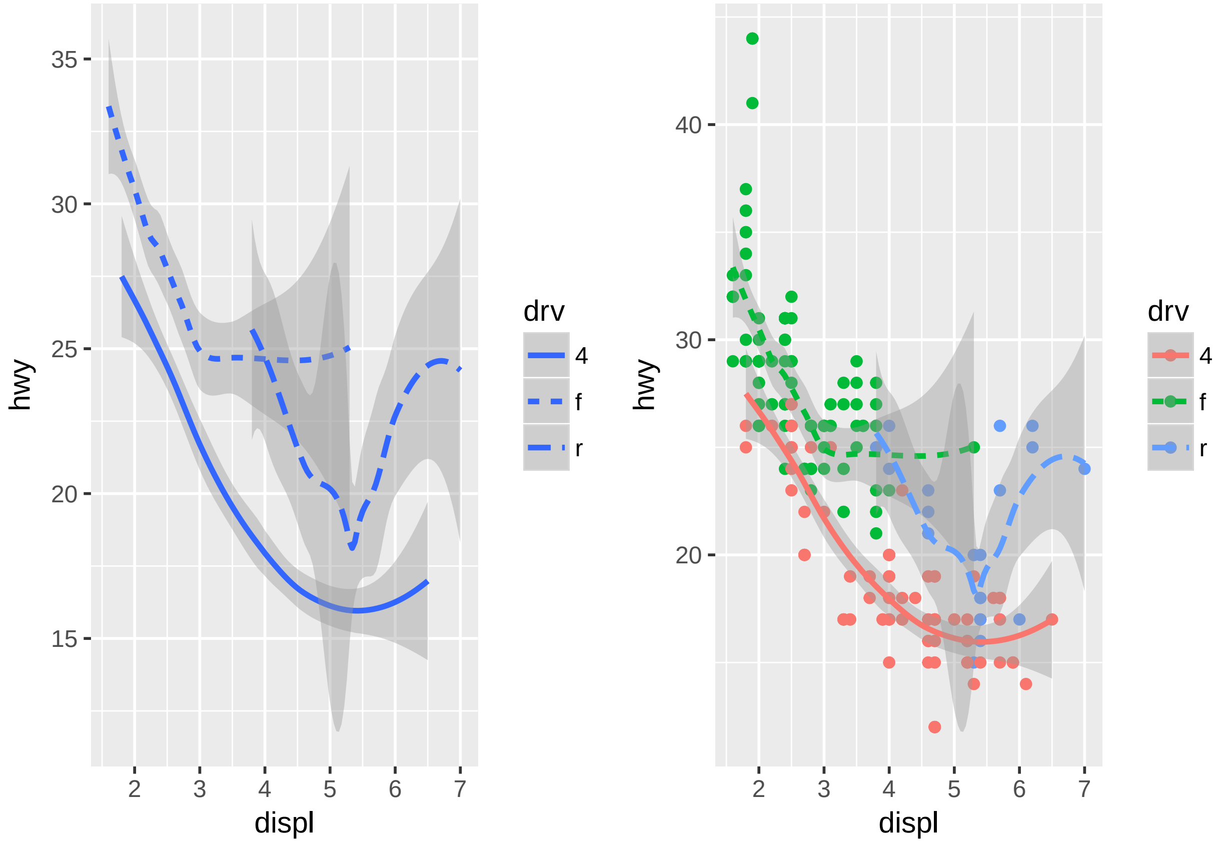
```
## `geom_smooth()` using method = 'loess'
```



`geom_smooth()` can be applied to different subsets of the data with the value mapped to the `linetype` attribute of `aes()`. For instance, the below plots show three smooth lines associated with three different drivetrains, plus the scatter plot of the data points added to the right figure.

```
ggplot(data = mpg) +
  geom_smooth(mapping = aes(x = displ, y = hwy, linetype = drv)) -> p1

ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, color = drv)) +
  geom_smooth(mapping = aes(x = displ, y = hwy, color = drv, linetype = drv))-> p2

grid.arrange(p1, p2, ncol = 2)
```

```
## `geom_smooth()` using method = 'loess'
## `geom_smooth()` using method = 'loess'
```



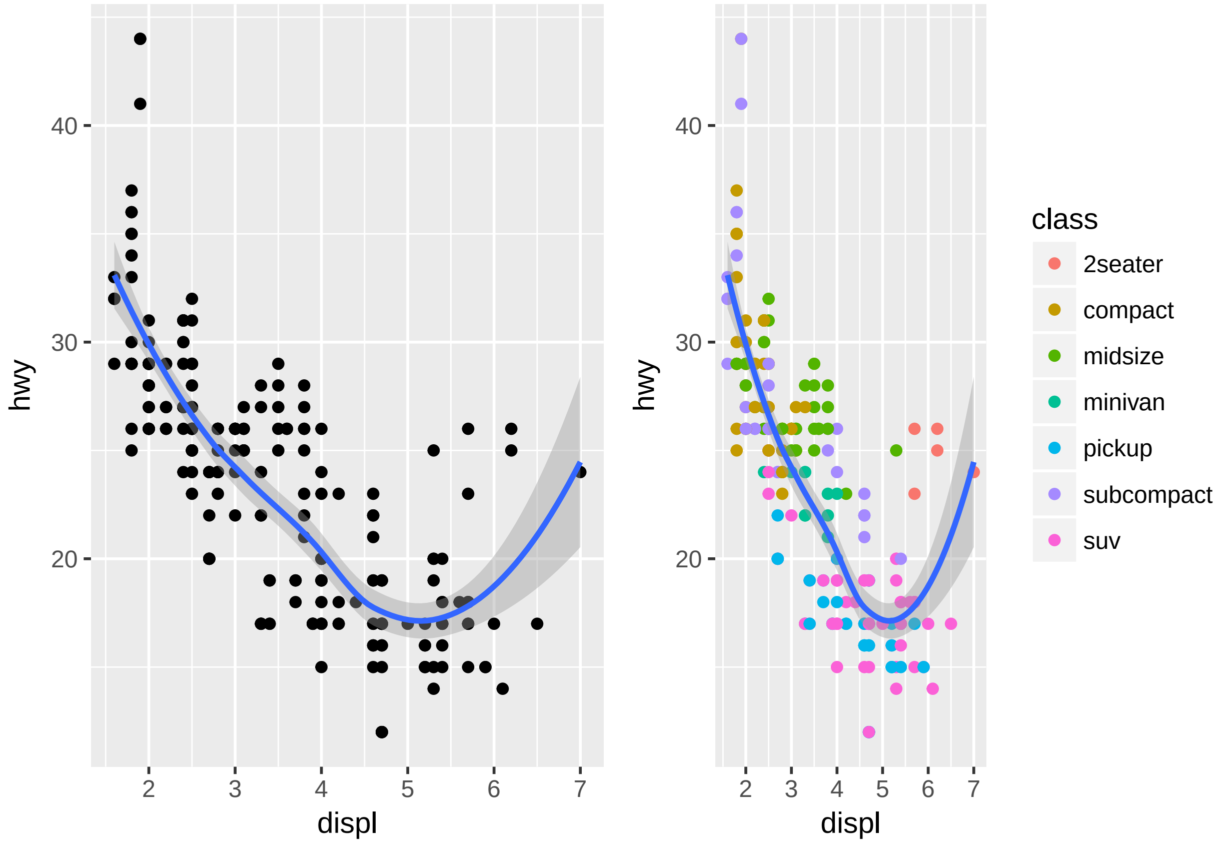
ulternatively, instead of `linetype` we could have used `group`. However, it does not show the legend for each category plotted. Anothe point to notice is plotting multiple geoms on the same plot with just adding various layers.

((many helpful cheatsheets are available at http://rstudio.com/cheatsheets))

Another way of having multiple geoms in one plot is to add the mappings to the `ggplot()` instead of each individual geom. With this way, not only we save some typing and get rid of duplications, but also the code will be less prone to errors as we no longer have to change the mapping in two or more different places every time we want to plot new aspects of the data. In addition, the global mappings (entered in `ggplot()`) will be inherited to the lower geoms, and could be overwritten by locally entered aesthetic mappings by the geom itself. The following two examples clarifies this approach.

```r
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point() +
  geom_smooth()-> p1

ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point(mapping = aes(color = class)) +
  geom_smooth()-> p2

grid.arrange(p1, p2, ncol = 2)
```

```
## `geom_smooth()` using method = 'loess'
## `geom_smooth()` using method = 'loess'
```



3.6.1 Exercises

1. What geom would you use to draw a line chart? A boxplot? A histogram? An area chart?

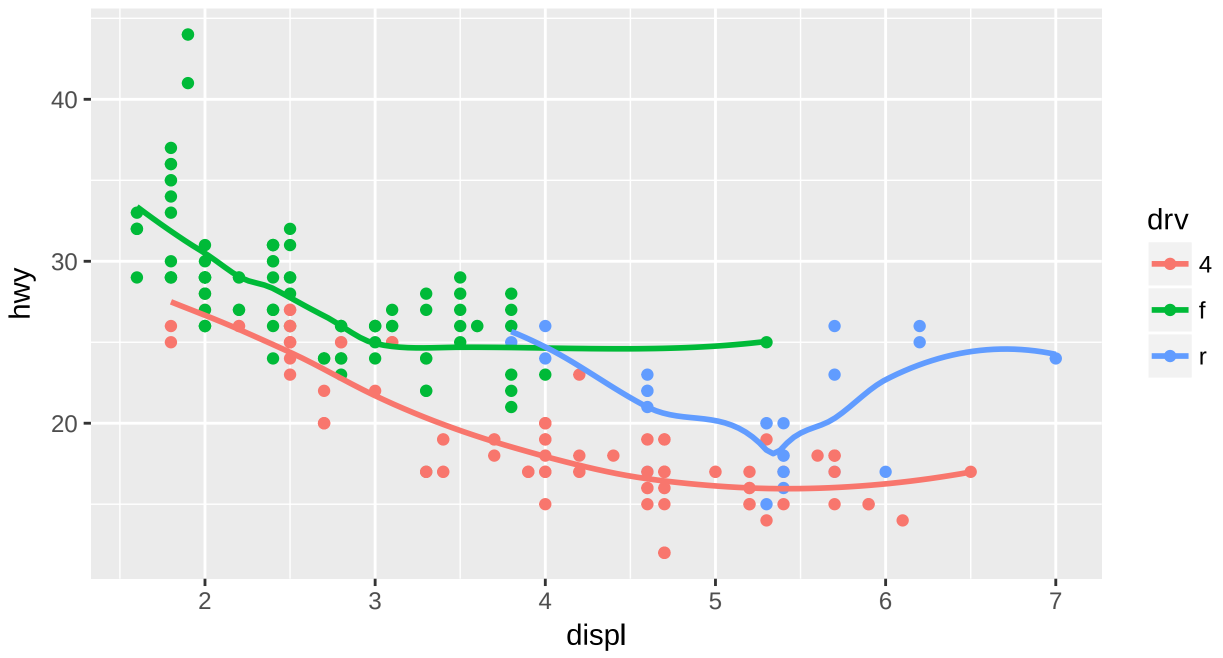
Respectively `geom_line()`, `geom_boxplot()`, `geom_bar()` or geom_histogram(), and `geom_area()`.

2. Run this code in your head and predict what the output will look like. Then, run the code in R and check your predictions.

This plot should contain a scatterplot of fuel efficiency vs. engine size, categorized by specific colors for each type of drivetrain. It also has another layer that shows a smoothed line, but without the standard error confidence interval, again using a specific color for front-wheel, rear-wheel, and four-wheel vehicle categories.

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

```
## `geom_smooth()` using method = 'loess'
```

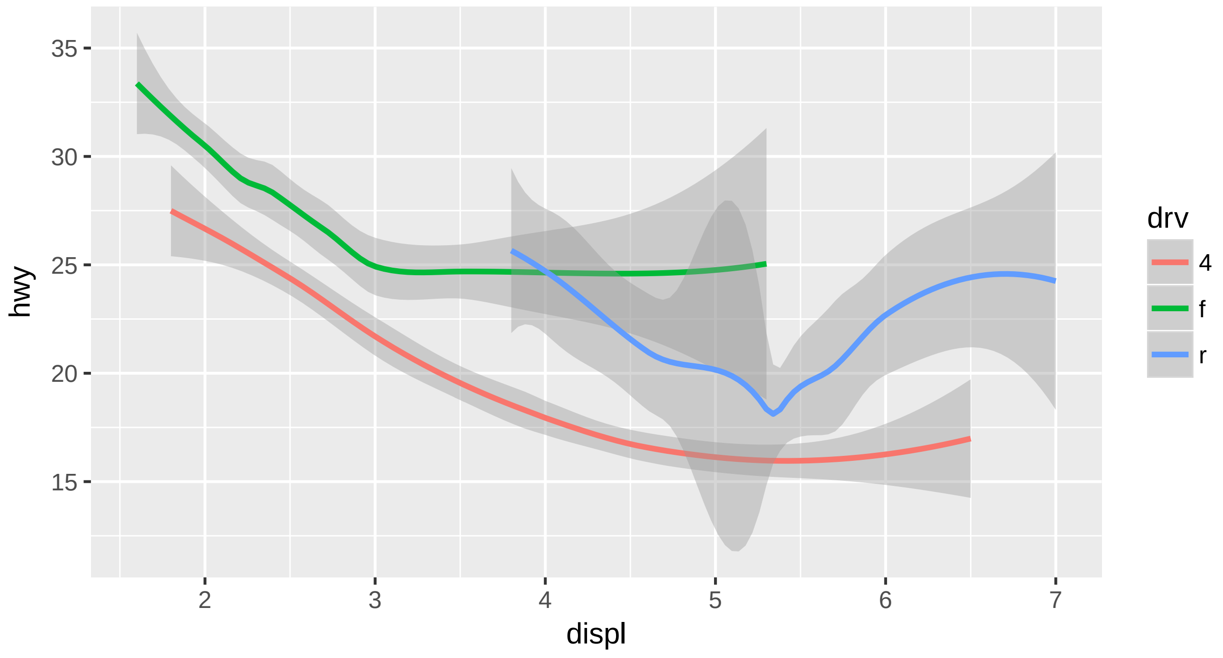


3. What does `show.legend = FALSE` do? What happens if you remove it? Why do you think I used it earlier in the chapter?

It removes the legend that shows the color and the variables associated together. If this line is removed, as the default is `show.legend = TRUE`, the legend comes back like following.

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

```
## `geom_smooth()` using method = 'loess'
```



It was removed earlier to compare `group = drv` and `color = drv`. Or to save space.

4. What does the `se` argument to `geom_smooth()` do?

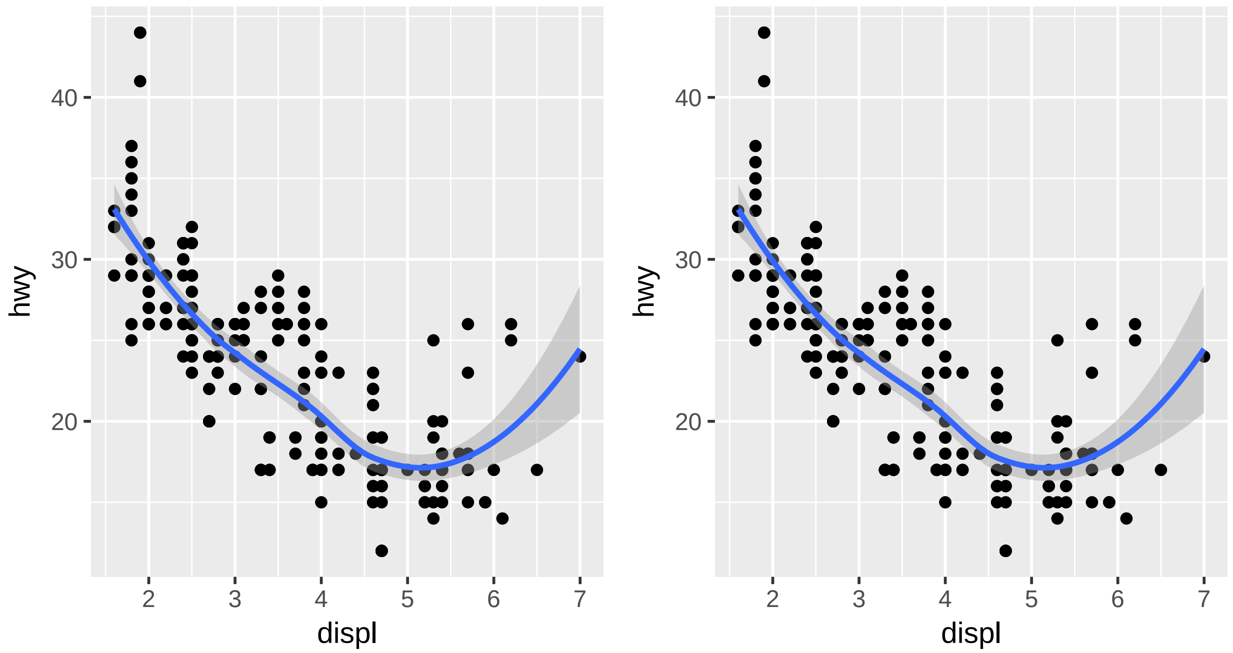
```
?geom_smooth()
```

It controls the display of the confidence interval around the smooth.

5. Will these two graphs look different? Why/why not?

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point() +
  geom_smooth() -> p1

ggplot() +
  geom_point(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_smooth(data = mpg, mapping = aes(x = displ, y = hwy)) -> p2

grid.arrange(p1, p2, ncol = 2)
```

```
## `geom_smooth()` using method = 'loess'
## `geom_smooth()` using method = 'loess'
```



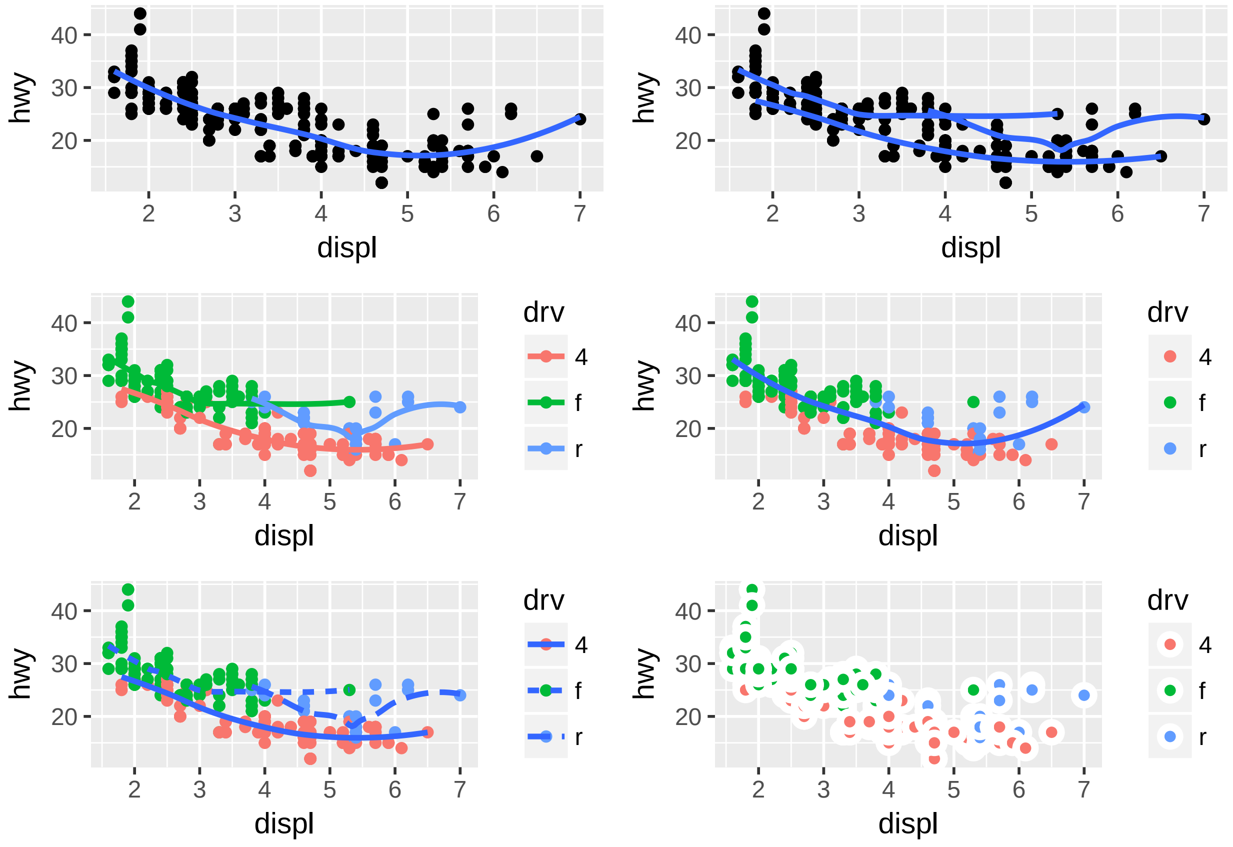
No, they don't. Since the inherited the same data and aesthetic mapping is used in both, in the first plot at the root of `ggplot()` and globaly, and in the second plot at each geom and locally.

6. Recreate the R code necessary to generate the following graphs.

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point() +
  geom_smooth(se = FALSE) -> p1

ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point() +
  geom_smooth(mapping = aes(group = drv), se = FALSE) -> p2
```

```r
ggplot(data = mpg, mapping = aes(x = displ, y = hwy, color = drv)) +
  geom_point() +
  geom_smooth(se = FALSE) -> p3

ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point(mapping = aes(color = drv)) +
  geom_smooth(se = FALSE) -> p4

ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point(mapping = aes(color = drv)) +
  geom_smooth(mapping = aes(linetype = drv), se = FALSE) -> p5

ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point(mapping = aes(fill = drv), shape = 21, size = 2, stroke = 2, color = "white") -> p6

grid.arrange(p1, p2, p3, p4, p5, p6, ncol = 2)
```

```
## `geom_smooth()` using method = 'loess'
## `geom_smooth()` using method = 'loess'
## `geom_smooth()` using method = 'loess'
## `geom_smooth()` using method = 'loess'
## `geom_smooth()` using method = 'loess'
```


hwy
displl
drv
f
r