

# R for Data Science Exercises

Dave Anderson

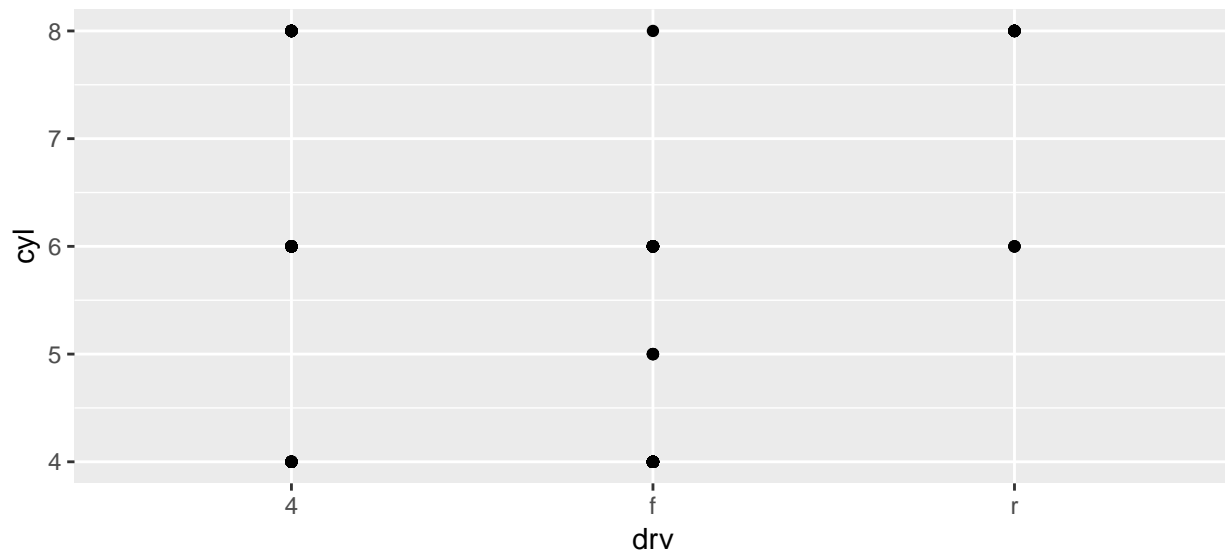
September 23, 2018

## R for Data Science Exercises (Assignment 2)

### 3.5.1 #2 and 3

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

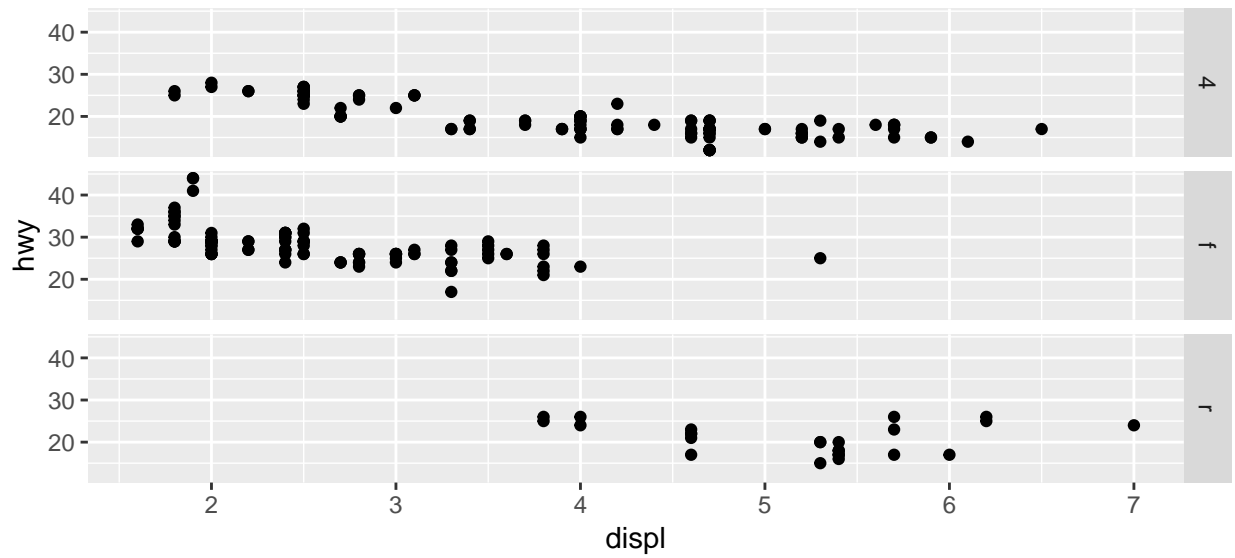
```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = drv, y = cyl))
```



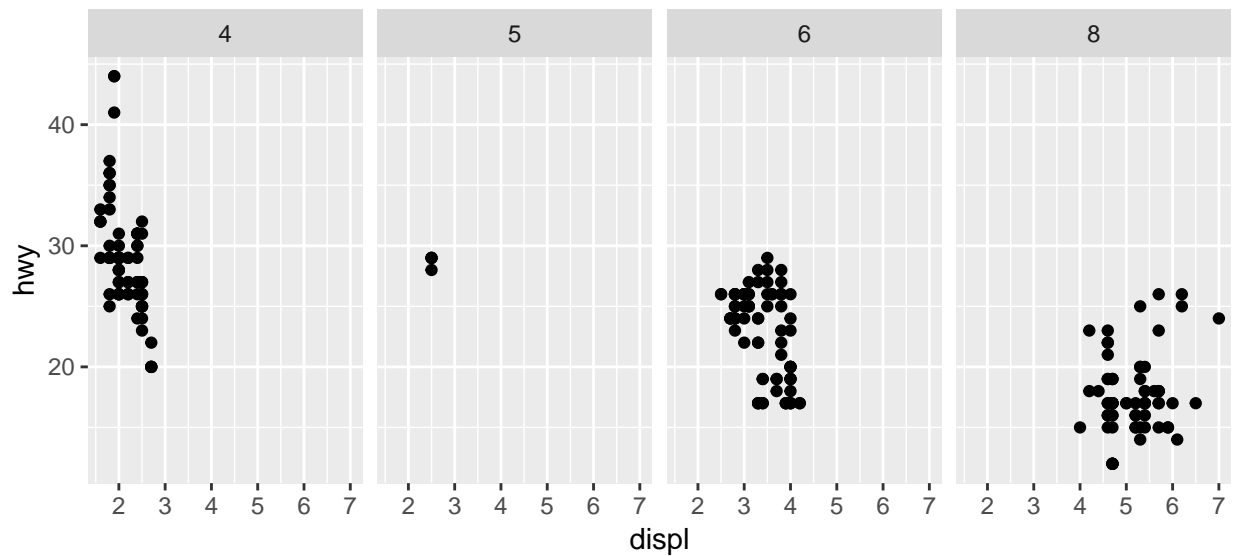
*Answer: The empty grids represent areas where that combination of variables does not exist. Same with empty spots in this plot. Example: There are no cars with 4 wheel drive and a 5 cylinder engine.*

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

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



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

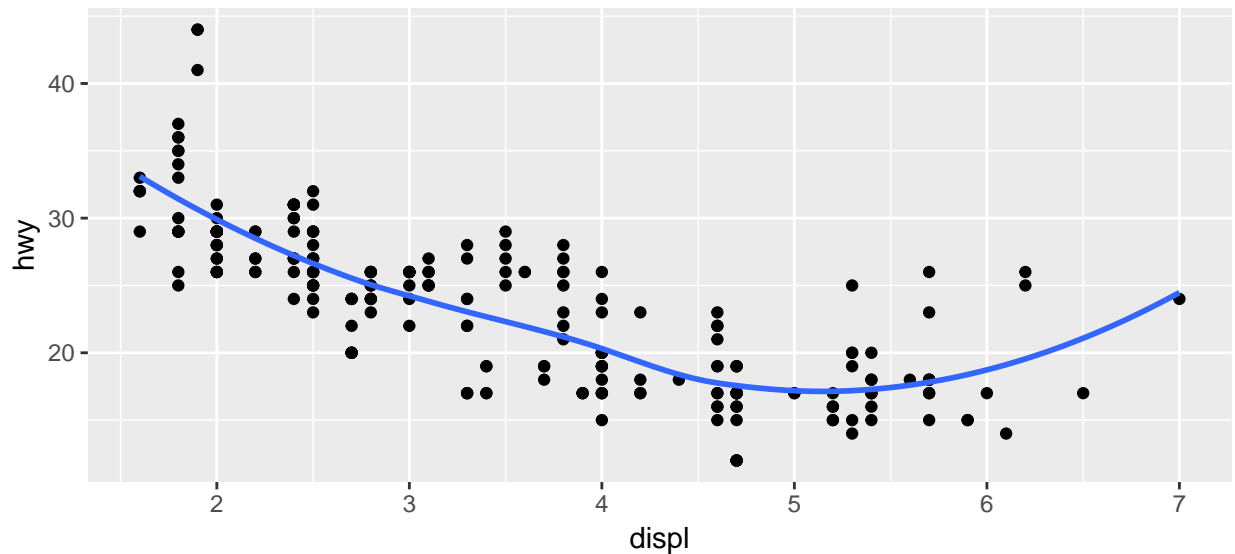


*Answer: The period is an empty placeholder in the rows or columns spot. It is similar when we are using [3,] to select a specific row. It is letting you sort your facets by row or column, vertical vs. horizontal*

### 3.6.1 #6

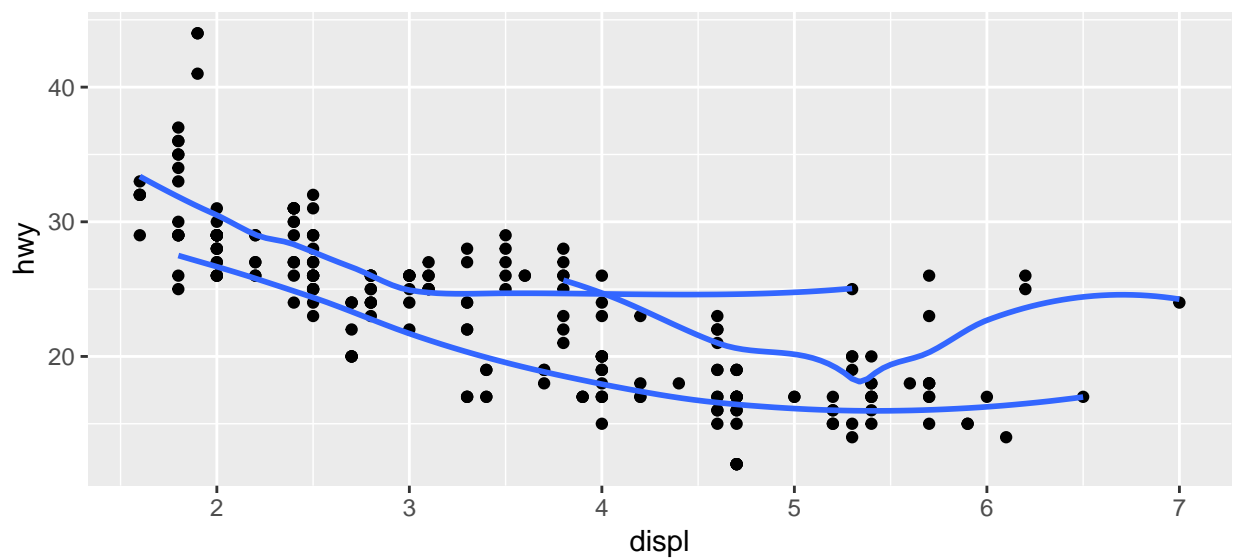
```
#1
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  geom_smooth(mapping = aes(x = displ, y = hwy), se = FALSE)

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



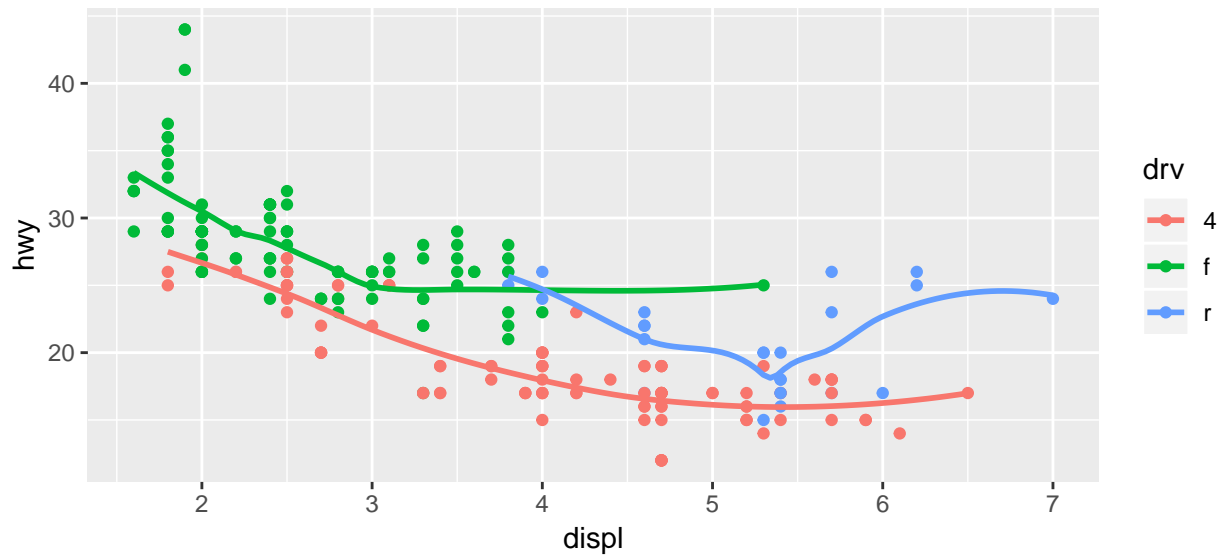
```
#2
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  geom_smooth(mapping = aes(x = displ, y = hwy, group = drv), se = FALSE)

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



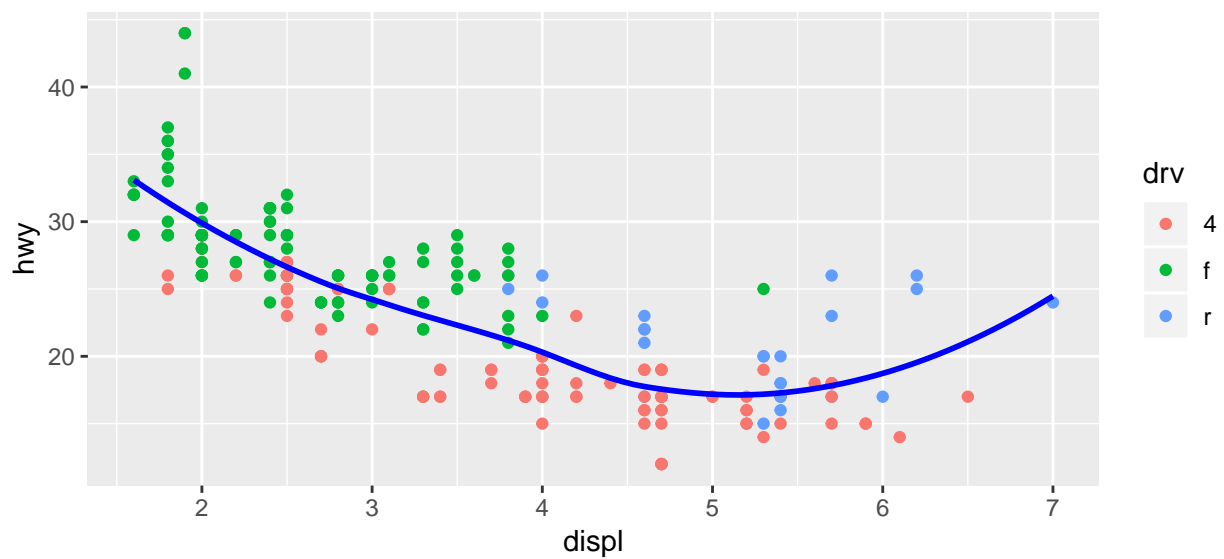
```
#3
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, color = drv)) +
  geom_smooth(mapping = aes(x = displ, y = hwy, group = drv, color = drv), se = FALSE)

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



```
#4
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, color = drv)) +
  geom_smooth(mapping = aes(x = displ, y = hwy), color = "blue", se = FALSE)

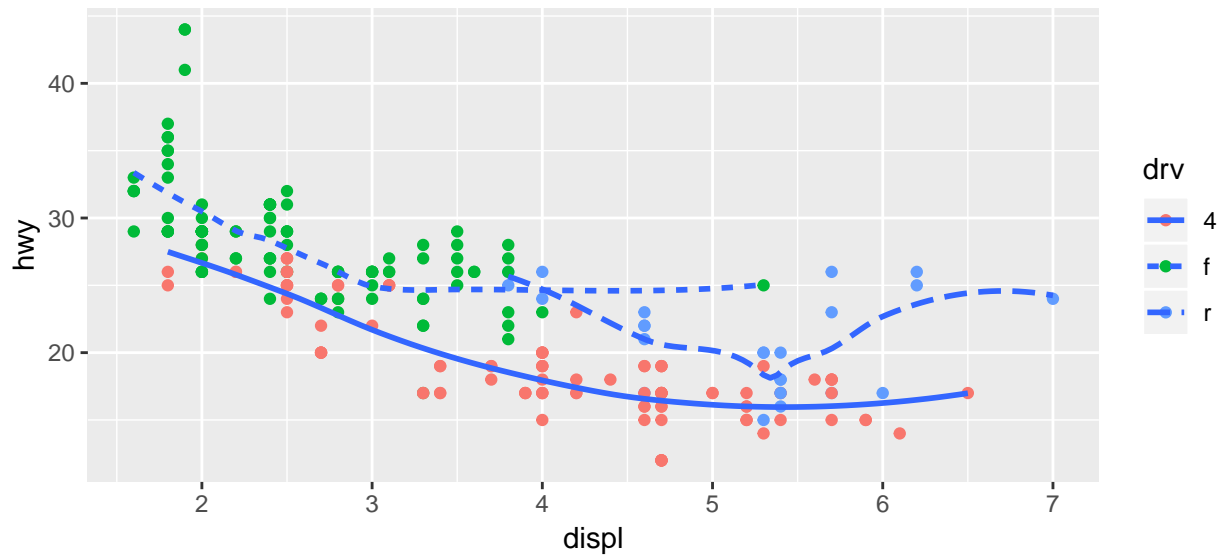
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



#5

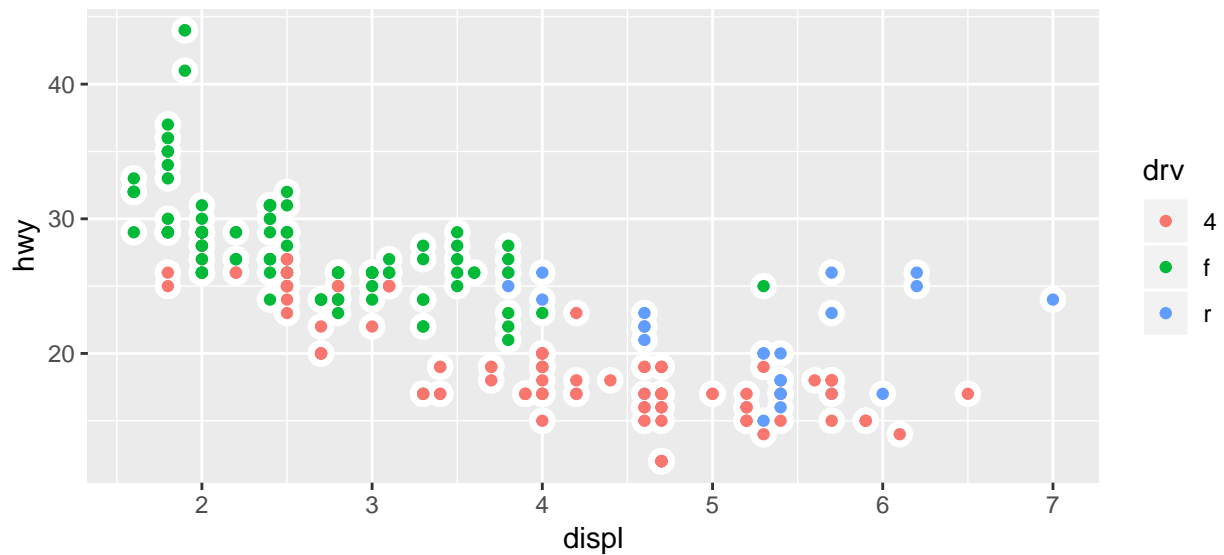
```
ggplot(mpg, aes(x = displ, y = hwy)) +  
  geom_point(aes(colour = drv)) +  
  geom_smooth(aes(linetype = drv), se = FALSE)
```

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'



#6

```
ggplot(mpg, aes(x = displ, y = hwy)) +  
  geom_point(size = 4, color = "white") +  
  geom_point(aes(colour = drv))
```



## 5.2.4 #1,2,3,4

1) Find all flights that:

```
?nycflights13::flights

## starting httpd help server ... done

flights <- nycflights13::flights
#1 Had an arrival delay of two or more hours
Two_hour_delay <- filter(flights, arr_delay >= 120)

#2 Flew to Houston (IAH or HOU)
Houston <- filter(flights, dest == "IAH" | dest == "HOU")

#3 Were operated by United, American, or Delta
United_American_Delta <- filter(flights, carrier %in% c("AA", "DL", "UA"))

#4 Departed in Summer
Summer <- filter(flights, month >= 7 & month <= 9)

#5 Arrived more than two hours late, but didn't leave late
arrive_delay <- filter(flights, arr_delay >= 120 & dep_delay <= 0)

#6 Were delayed by at least an hour, but made up over 30 minutes in flight
quick_fly <- filter(flights, dep_delay >= 60, arr_delay <= 30)

#7 Departed between midnight and 6am (inclusive)
late_flight <- filter(flights, dep_time >= 0 & dep_time <= 600)
```

2) Another useful dplyr filtering helper is `between()`. What does it do? Can you use it to simplify the code needed to answer the previous challenges?

*Answer: `between()` simply allows us to filter data between two values. It would have been useful for the codes in problems 4 and 7. Simplified codes:*

```
#4
Summer <- filter(flights, between(month, 7, 9))

#7
late_flight <- filter(flights, between(dep_time, 0, 600))
```

3) How many flights have a missing `dep_time`? What other variables are missing? What might these rows represent?

```
missing <- filter(flights, is.na(dep_time))
```

*Answer: These rows are also missing arrival times, delay times, and flight times. These rows probably represent canceled flights.*

4) Why is `NA ^ 0` not missing? Why is `NA | TRUE` not missing? Why is `FALSE & NA` not missing? Can you figure out the general rule? (`NA * 0` is a tricky counterexample!)

*Answer: `NA ^ 0` is not missing because anything to the 0 power, even NA, has an output of 1. `NA | TRUE` will read as `TRUE`. Similarly, anything and `FALSE` will read as `FALSE`. `NA times 0` reads as `NA` because if the missing spot read as infinity, the result would read as not a number.*