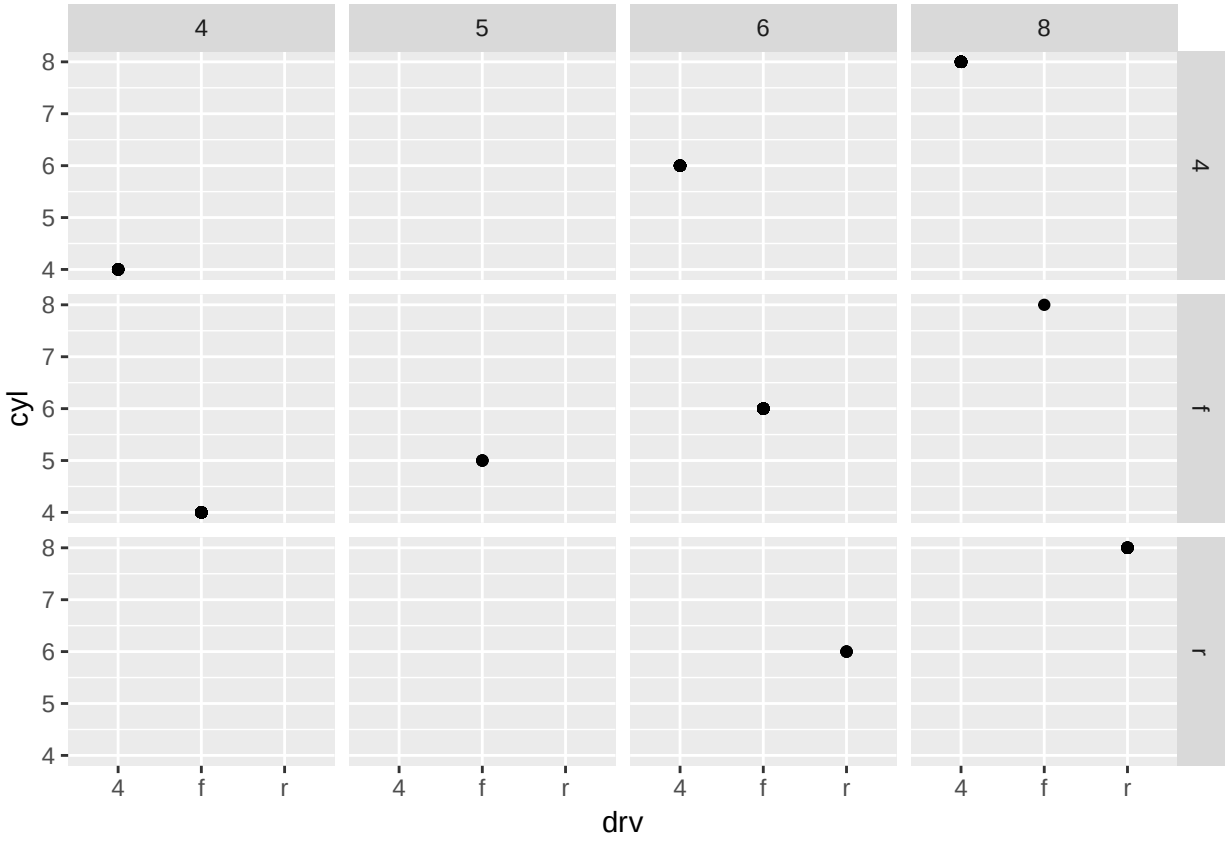
MA615_Assignment2_Exercise

Sky Liu

9/24/2018

3.5.1(2) What do the empty cells in plot with facet_grid(drv ~ cyl) mean? How do they relate to this plot?

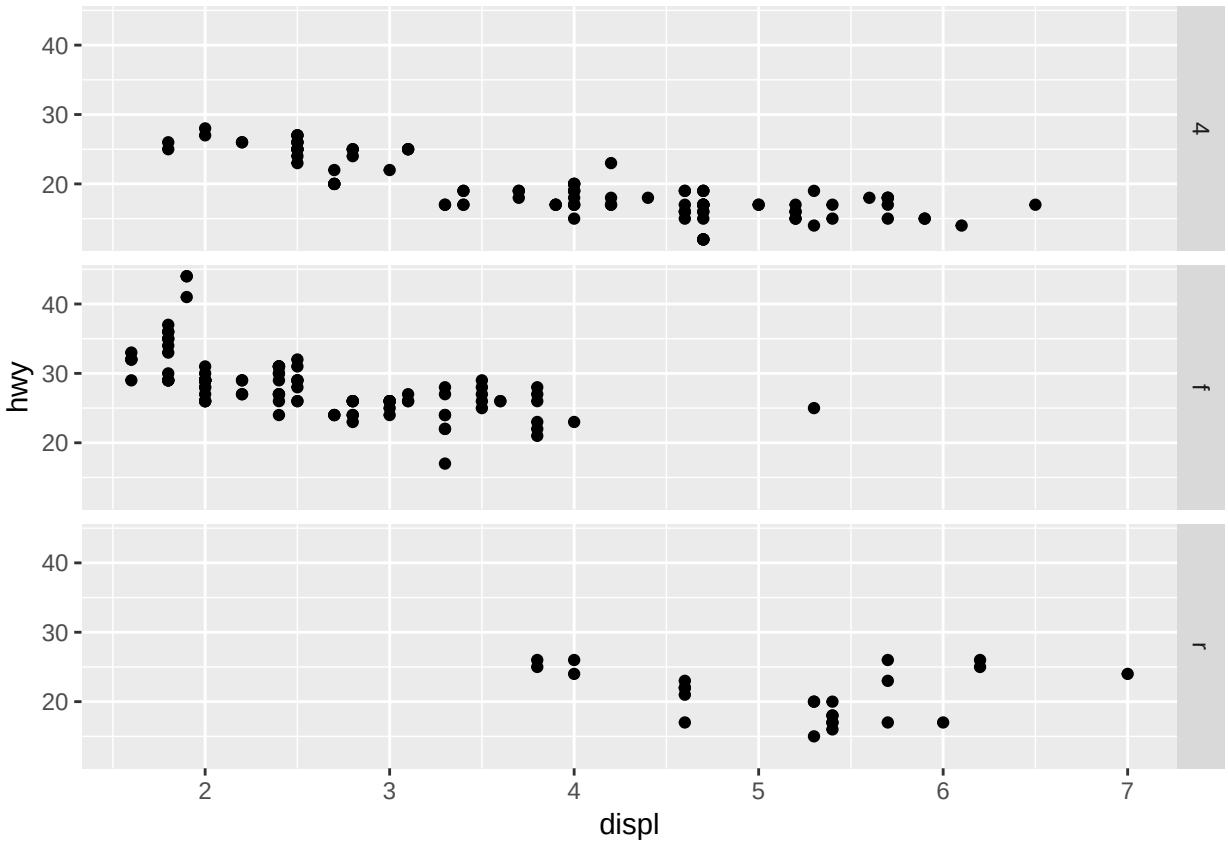
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = drv, y = cyl)) +
  facet_grid(drv ~ cyl)
```



The empty cells imply that the data set has no values within this row range and combination. In this plot, it shows there is four wheel driver with 5 cylinder is missing nor rear wheel driver with 4 or 5 cylinder.

3.5.1(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 ~ .)
```



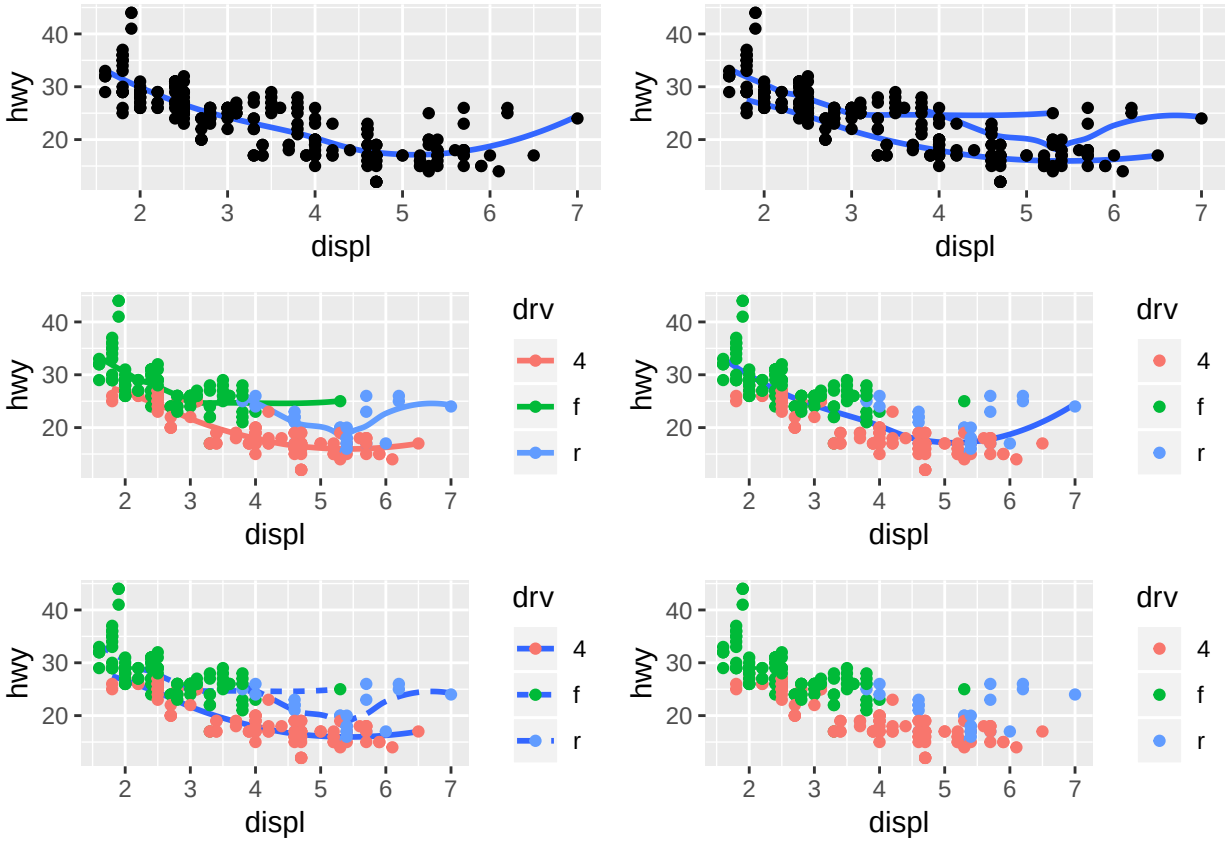
The dot is used to indicate there should be no faceting on this dimension.

3.6.1(6) Recreate the R code necessary to generate the following graphs.

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

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

p3 <- ggplot(data = mpg, mapping = aes(x=displ,y=hwy,group=drv)) +
  geom_smooth(se = FALSE,aes(colour=drv))+
  geom_point(aes(colour=drv))

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

p5 <- ggplot(data = mpg, mapping = aes(x=displ,y=hwy,group=drv)) +
  geom_smooth(se = FALSE,aes(linetype=drv))+
  geom_point(aes(colour=drv))
```

```
p6 <- ggplot(data = mpg, mapping = aes(x=displ,y=hwy)) +
  geom_point(aes(colour=drv))
grid.arrange(p1,p2,p3,p4,p5,p6,nrow=3,ncol=2)
```

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



5.2.4(1) Find all flights that

1. Had an arrival delay of two or more hours

```
flights1 <- filter(flights, arr_delay>=120)
```

2. Flew to Houston (IAH or HOU)

```
flights2 <- filter(flights, dest %in% c('IAH', 'HOU'))
```

3. Were operated by United, American, or Delta

```
flights3 <- filter(flights, carrier %in% c('UA', 'AA', 'DL'))
```

4. Departed in summer (July, August, and September)

```
flights4 <- filter(flights, month %in% c(7, 8, 9))
```

5. Arrived more than two hours late, but didn't leave late

```
flights5 <- filter(flights, arr_delay > 60*2 & dep_delay <= 0)
```

6. Were delayed by at least an hour, but made up over 30 minutes in flight

```
flights6 <- filter(flights, dep_delay >= 60 & dep_delay-arr_delay > 30)
```

7. Departed between midnight and 6am (inclusive)

```
flights7_1 <- filter(flights, dep_time >= 2400 & dep_time <= 600)
```

5.2.4(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?

7. Departed between midnight and 6am (inclusive)

```
flights7_2 <- filter(flights, !between(dep_time, 601, 2359))
```

5.2.4(3) How many flights have a missing dep_time? What other variables are missing? What might these rows represent?

```
sum(is.na(flights$dep_time))
```

```
## [1] 8255
```

```
map(flights, ~ sum(is.na(.x)))
```

```
## $year
## [1] 0
##
## $month
```

```
## [1] 0
##
## $day
## [1] 0
##
## $dep_time
## [1] 8255
##
## $sched_dep_time
## [1] 0
##
## $dep_delay
## [1] 8255
##
## $arr_time
## [1] 8713
##
## $sched_arr_time
## [1] 0
##
## $arr_delay
## [1] 9430
##
## $carrier
## [1] 0
##
## $flight
## [1] 0
##
## $tailnum
## [1] 2512
##
## $origin
## [1] 0
##
## $dest
## [1] 0
##
## $air_time
## [1] 9430
##
## $distance
## [1] 0
##
## $hour
## [1] 0
##
## $minute
## [1] 0
##
## $time_hour
## [1] 0
```

Cancelled flights maybe?

5.2.4(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!)

NA ^ 0 is not missing because anything to the power of 0 is 1.

NA | TRUE is not missing because the boolean value returned by this expression is TRUE.

FALSE & NA is not missing because because the boolean value returned by this expression is FALSE.

The general rule is that as long as there is a logical judgement of the expression, the result should not be NA.