STAT530 HW5

Tianqi Wu (twu38)

#Problem 1

##Section 3.2.4

###Exercises 1: Run ggplot(data = mpg). What do you see?

Answer: Since only data is given without geom and mapping, ggplot() only generates a background with empty plot.

###Exercises 2: How many rows are in mpg? How many columns?

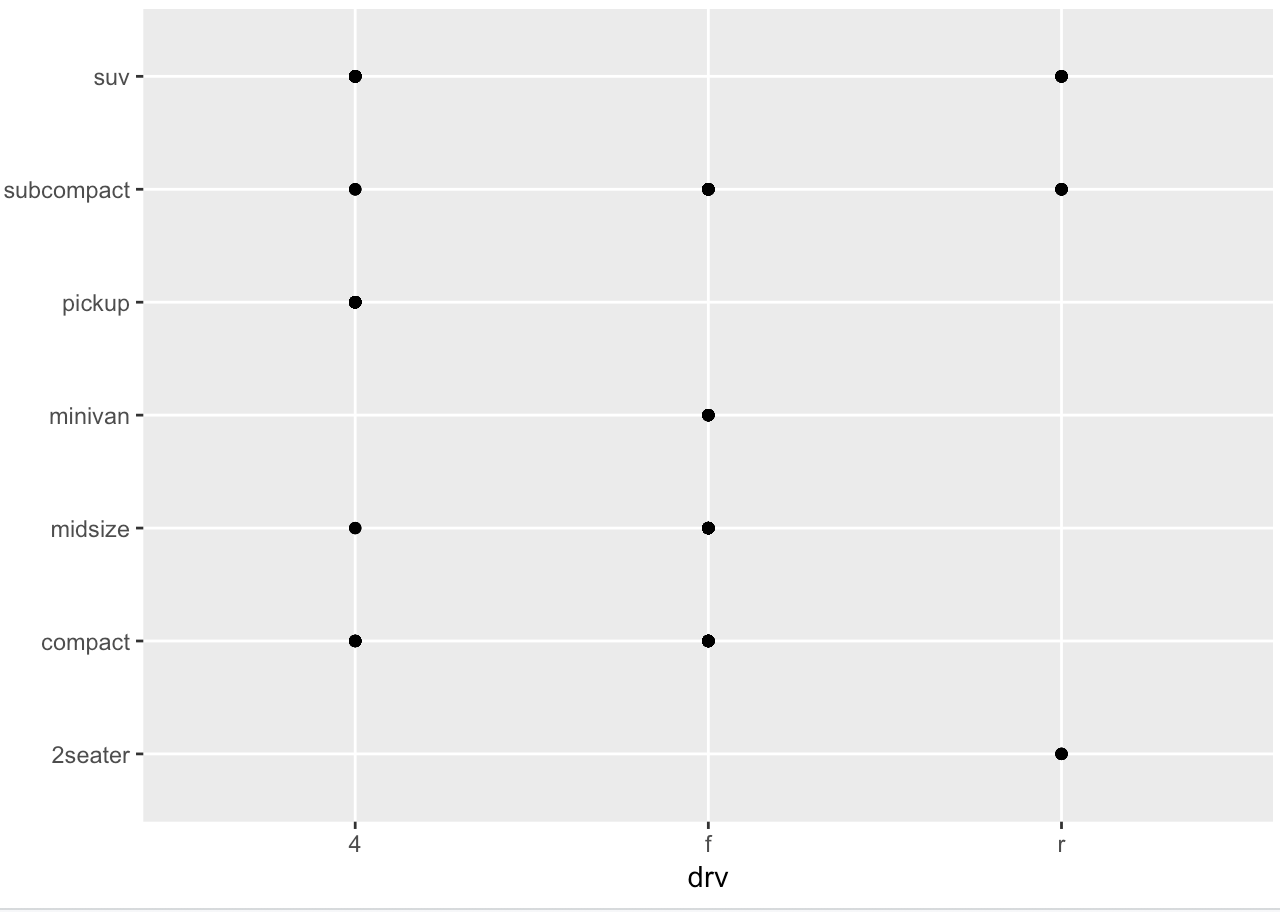
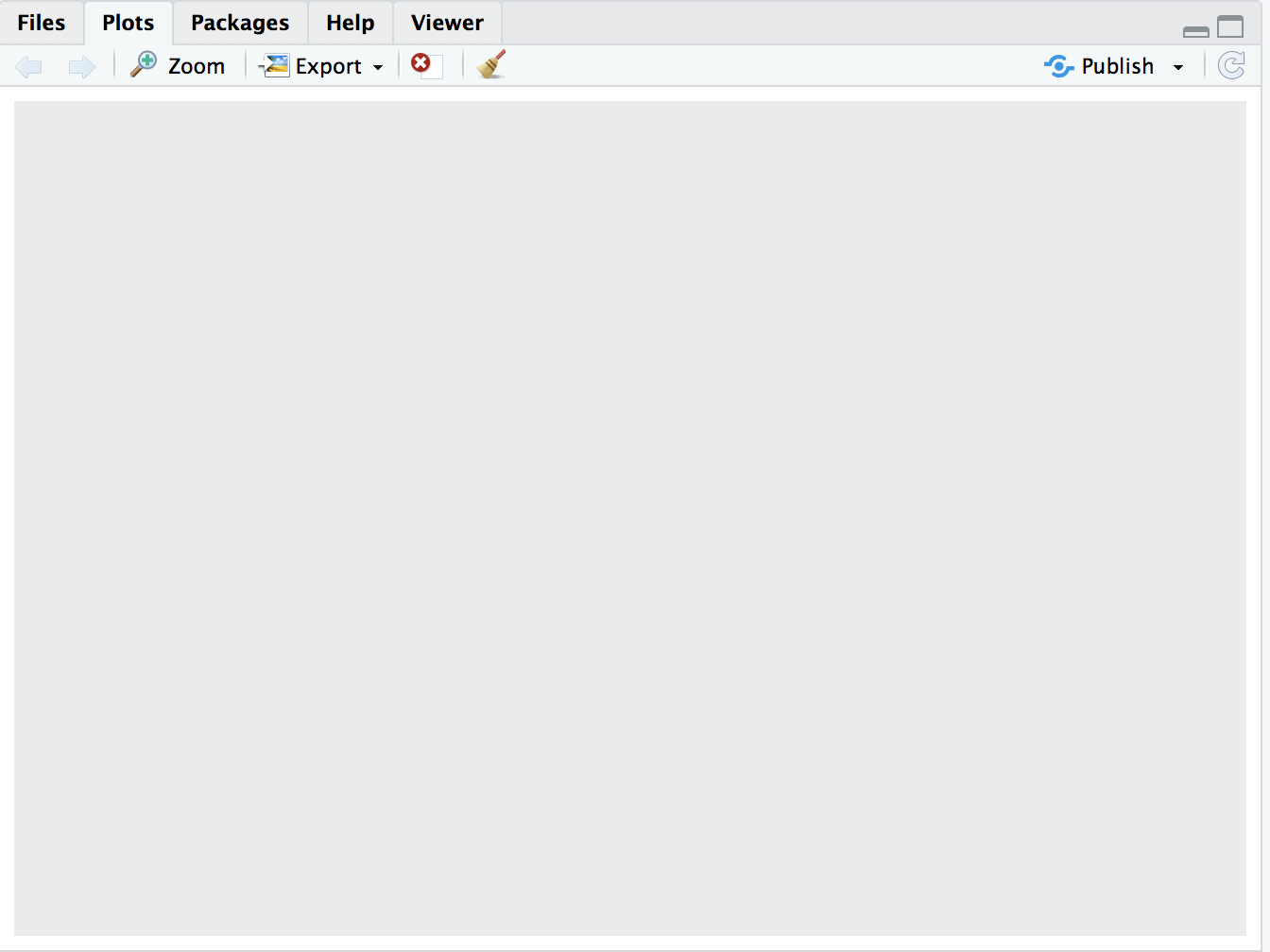
Answer: It has 234 rows and 11 columns.

> ncol(mpg)

[1] 11

> nrow(mpg)

[1] 234

###Exercises 5: What happens if you make a scatterplot of class vs drv? Why is the plot not useful?

Answer: The scatterplot shows the combinations of class and drv in the dataset. Since class and drv are both discrete variables, a scatterplot would not tell us how many observations there are in each combination. For two discrete variables, a better plot would be geom\_jitter().

> ggplot(data = mpg, aes(y=class,x=drv))+geom\_point()

--------------------------

##Section 3.3.1

###Exercises 1: What’s gone wrong with this code? Why are the points not blue?

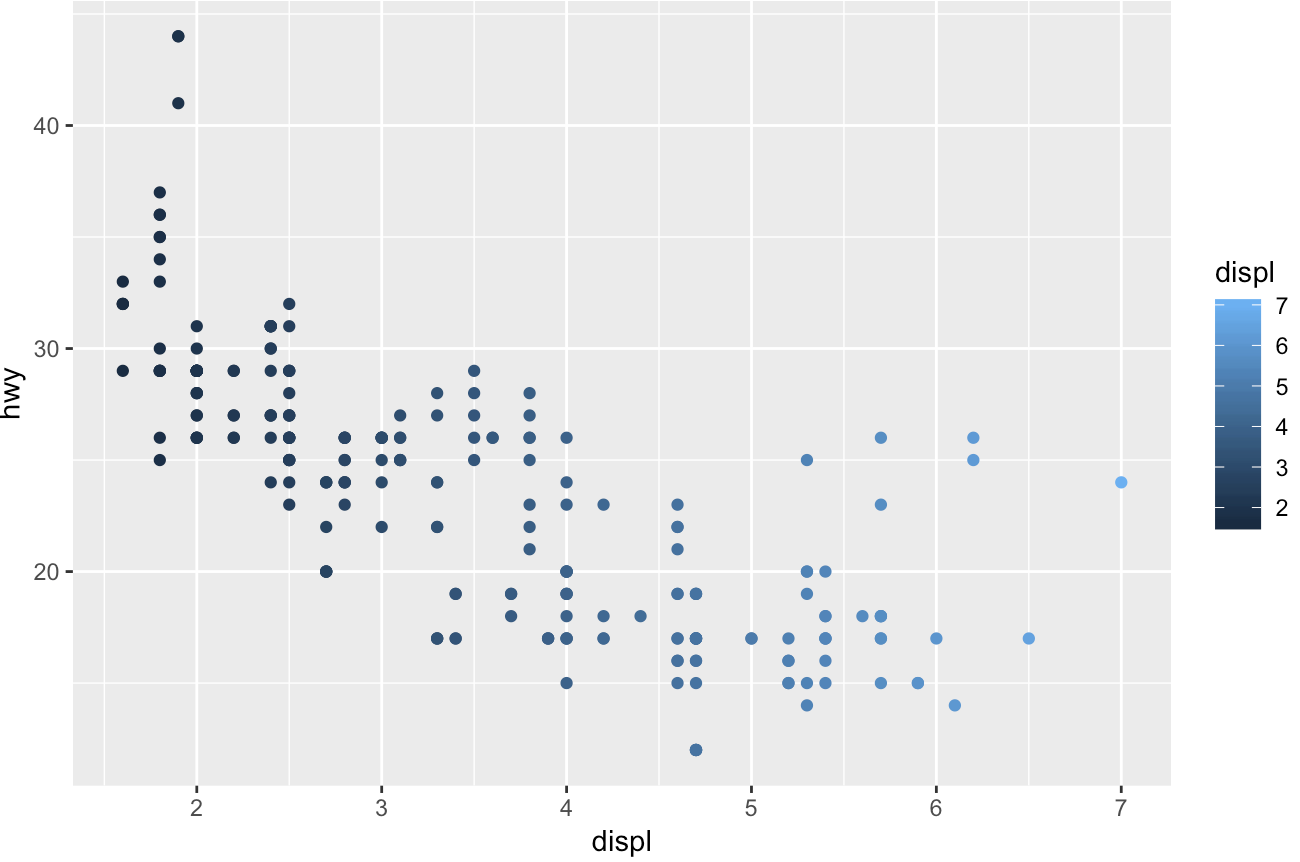
Answer: Since color="blue" is included in the aes() mapping, it maps the variable color to character "blue". The correct way would be included color="blue" outside the aes() mapping:

> ggplot(data = mpg) +

+ geom\_point(mapping = aes(x = displ, y = hwy), color = "blue")

###Exercises 2: Which variables in mpg are categorical? Which variables are continuous? (Hint: type ?mpg to read the documentation for the dataset). How can you see this information when you run mpg?

Answer: The output running mpg tells us the types of the variables with <chr>:categorical and <int>:continuous.

Hence, manufacturer, model, trans, drv, fl and class are categorical variables.

displ, year, cyl, cty, hwy are continuous variabels.

> 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>

###Exercises 3: Map a continuous variable to color, size, and shape. How do these aesthetics behave differently for categorical vs. continuous variables?

Answer:When mapping continuous variable to color, it results a color scale from light(smaller) to dark(larger).

> ggplot(data = mpg) +

* geom\_point(mapping = aes(x = displ, y = hwy, color=displ))

When mapping continuous variable to size, the larger values are represented with larger circle.

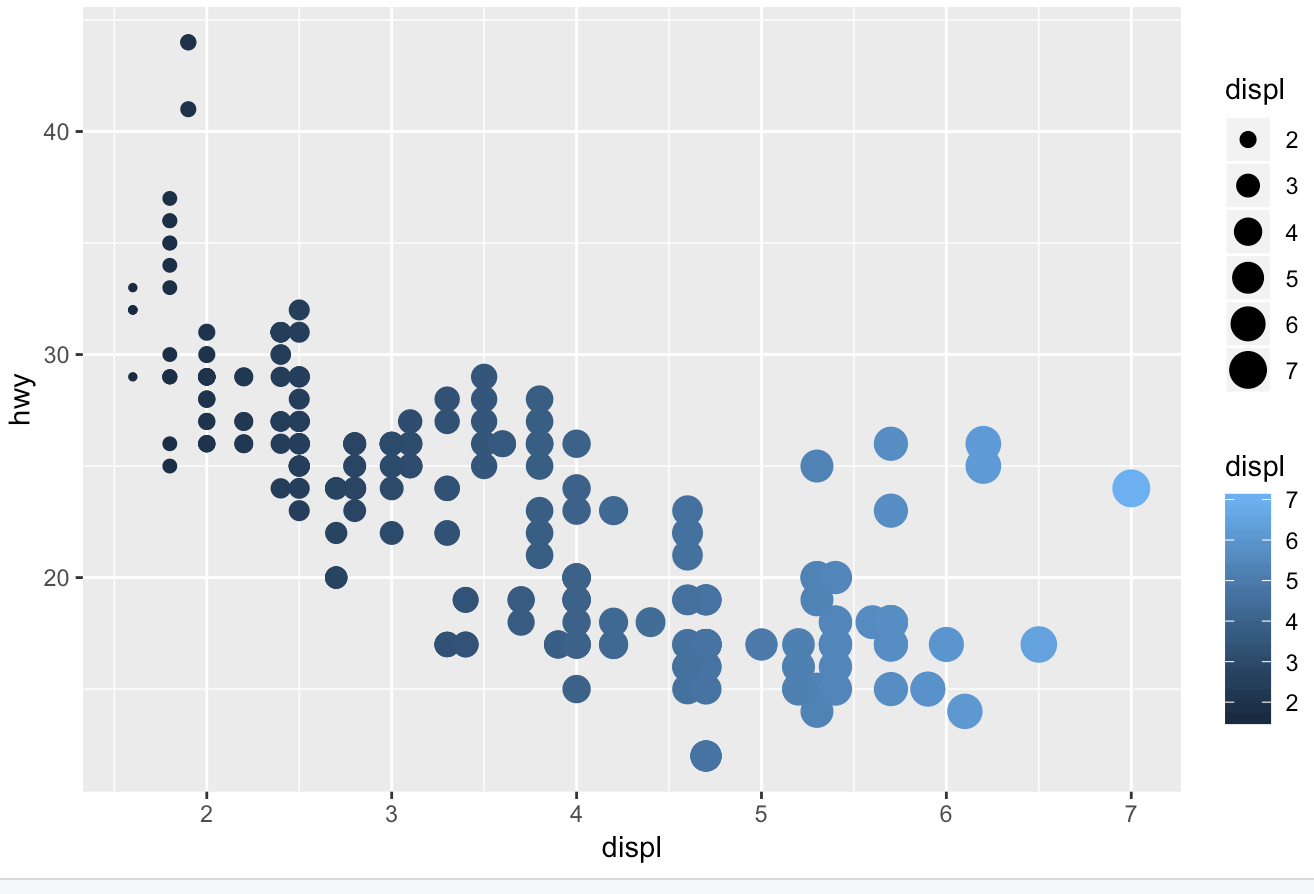
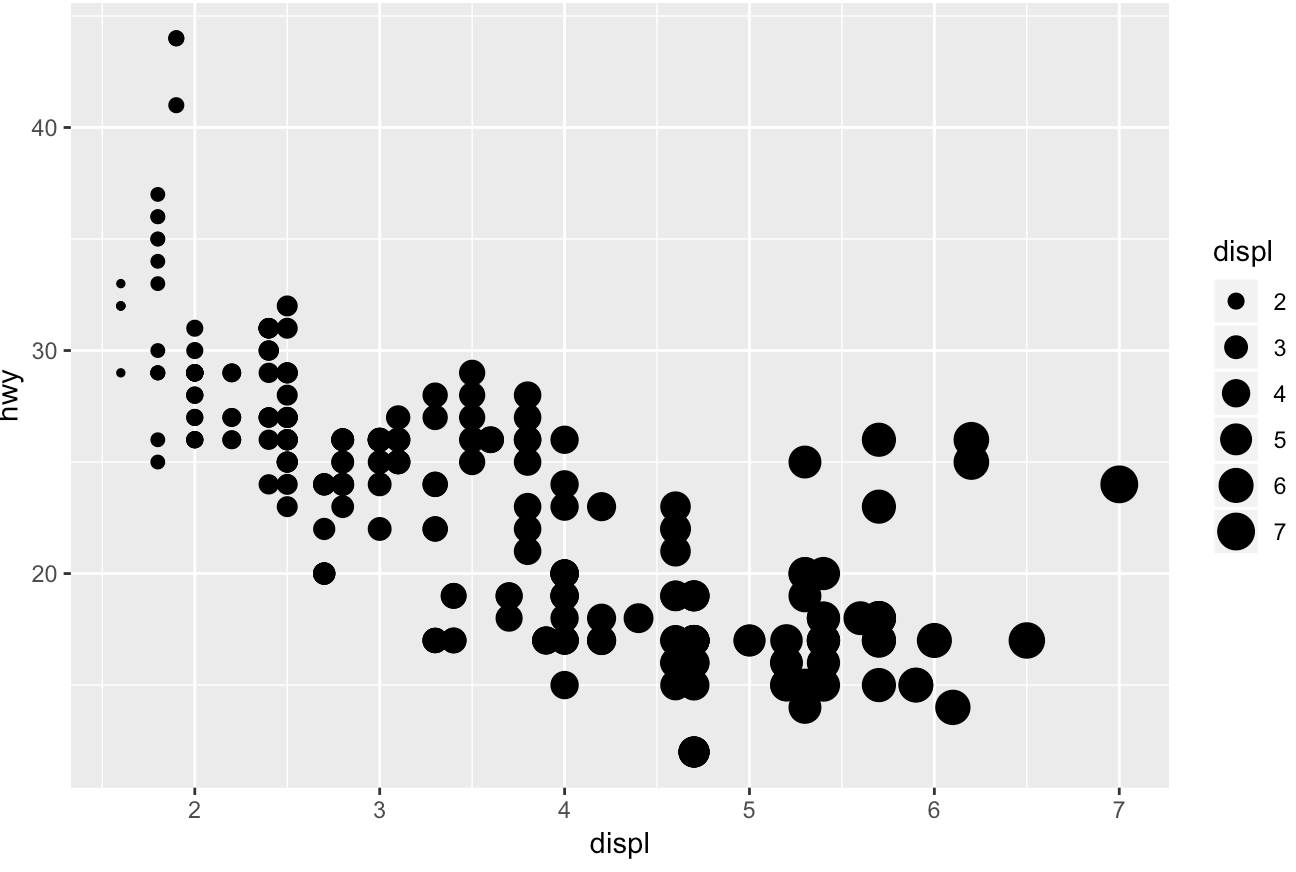
> ggplot(data = mpg) +

+ geom\_point(mapping = aes(x = displ, y = hwy, size=displ))

when mapping continuous variable to shape, it results an error.

+ geom\_point(mapping = aes(x = displ, y = hwy, shape=displ))

Error: A continuous variable can not be mapped to shape



###Exercises 4: What happens if you map the same variable to multiple aesthetics?

Answer: When mapping displ to x, color and size, plot could still be generated with combination of effects described with previous question. However, the information is redundant.

> ggplot(data = mpg) +

+ geom\_point(mapping = aes(x = displ, y = hwy, color=displ,size=displ))

--------------------------

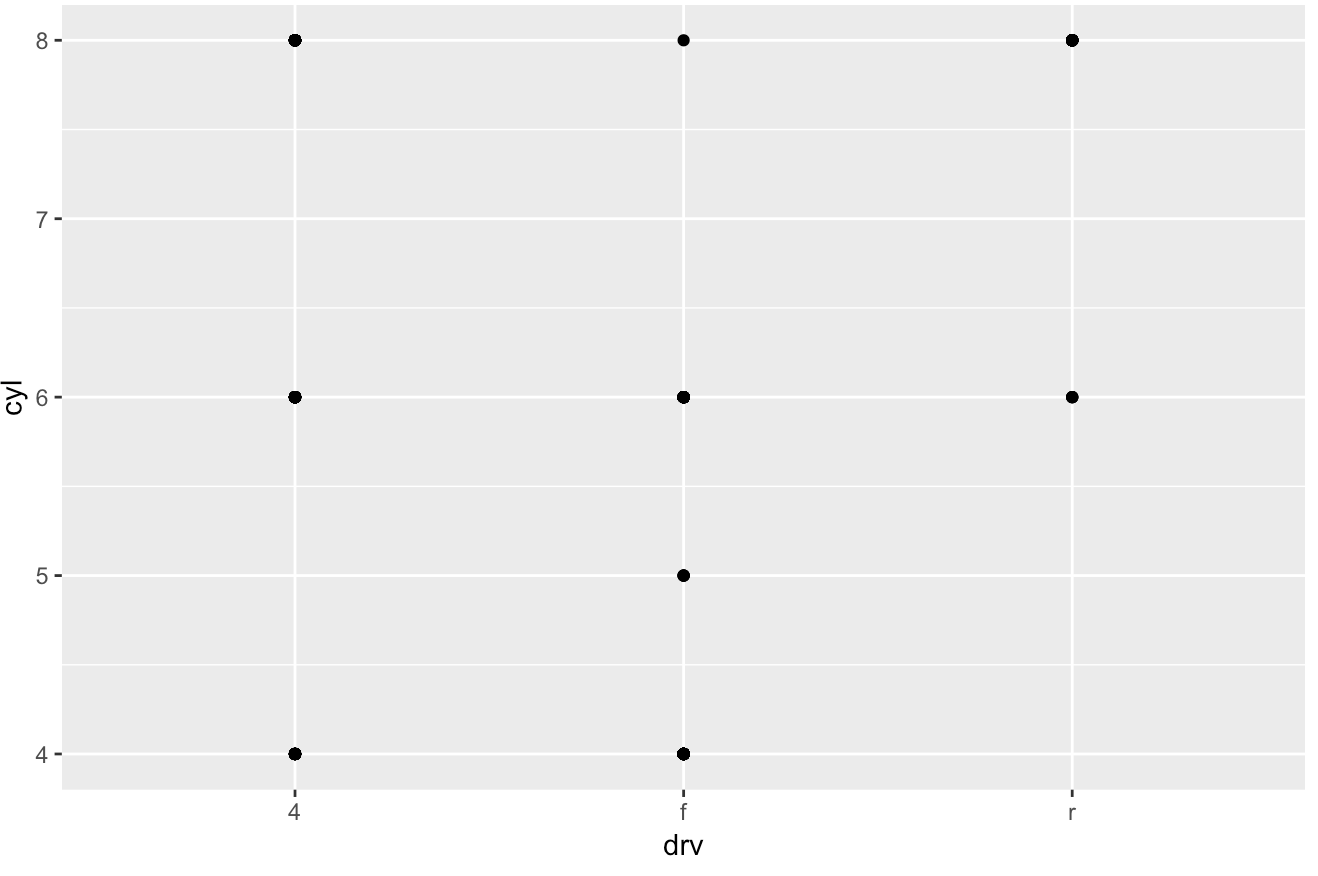
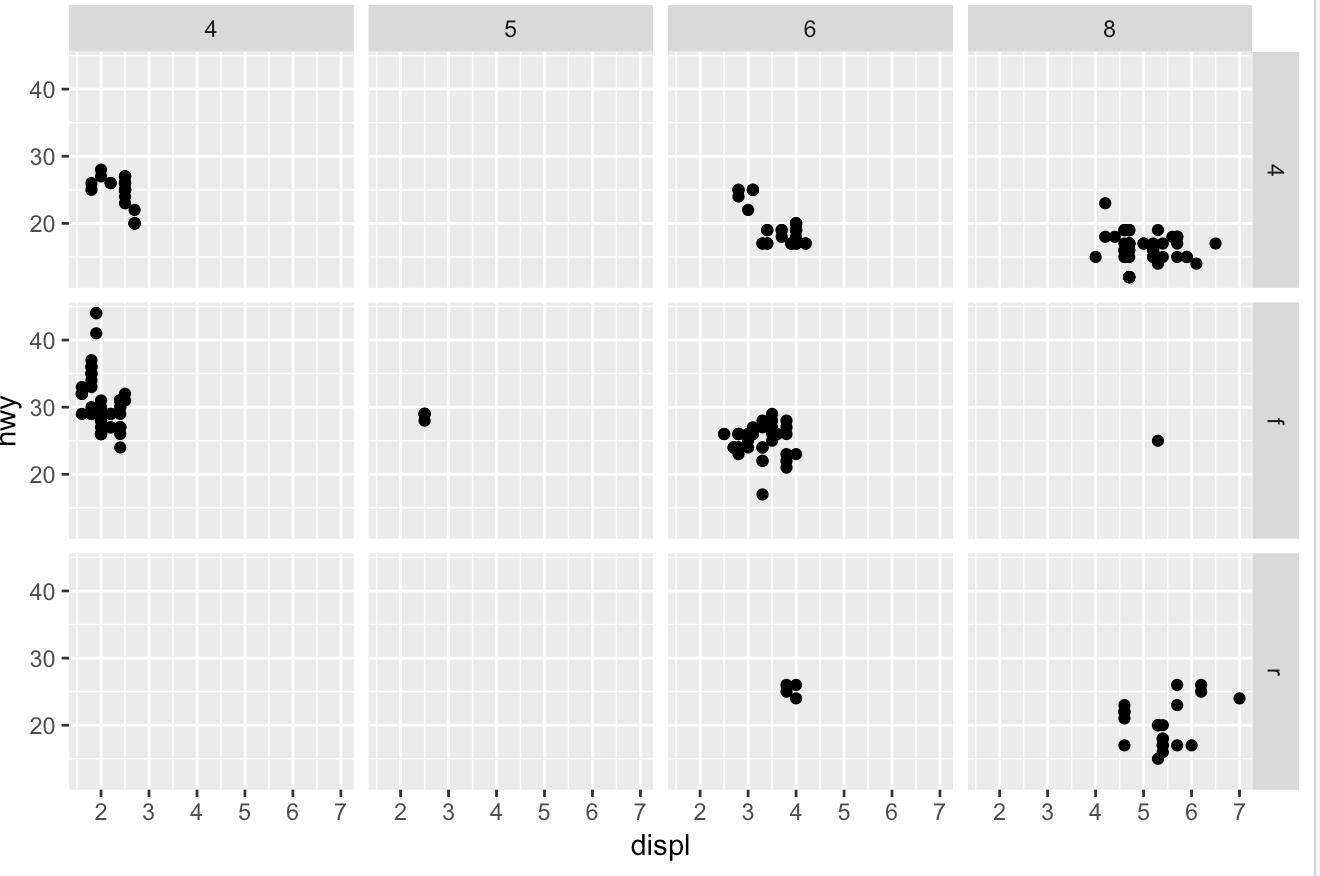
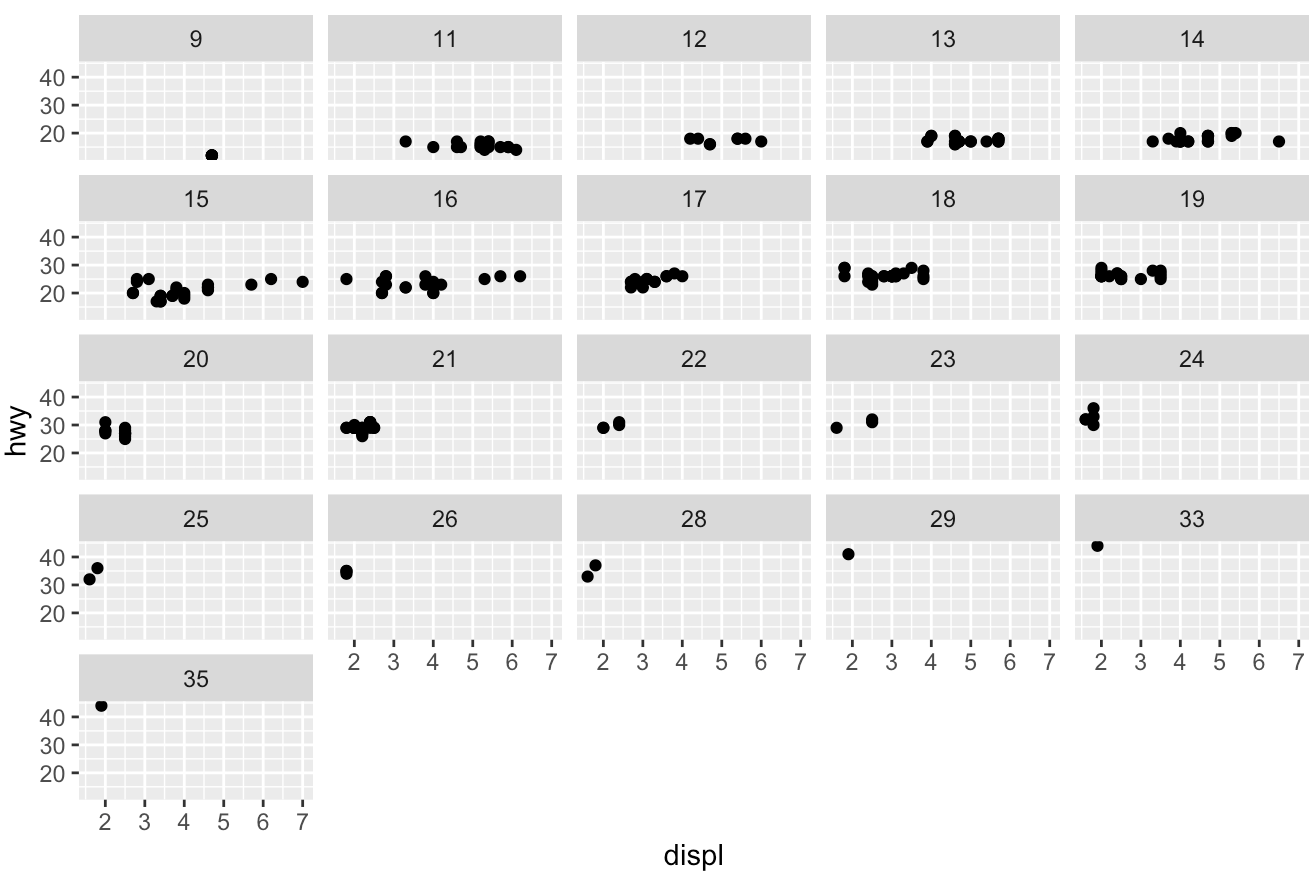
##Section 3.5.1

###Exercises 1: What happens if you facet on a continuous variable?

Answer: It will plot a graph fro each value of the continuous variable.

> ggplot(data = mpg) +

+ geom\_point(mapping = aes(x = displ, y = hwy))+facet\_wrap(~cty)

###Exercises 2: What do the empty cells in plot with facet\_grid(drv ~ cyl) mean? How do they relate to this plot?

Answer: The empty cells indicate that there is no such combination specified with those values int the dataset. The scatter plot essentially tells the same information.

> ggplot(data = mpg) +

+ geom\_point(mapping = aes(x = displ, y = hwy)) +

+ facet\_grid(drv ~ cyl)

>

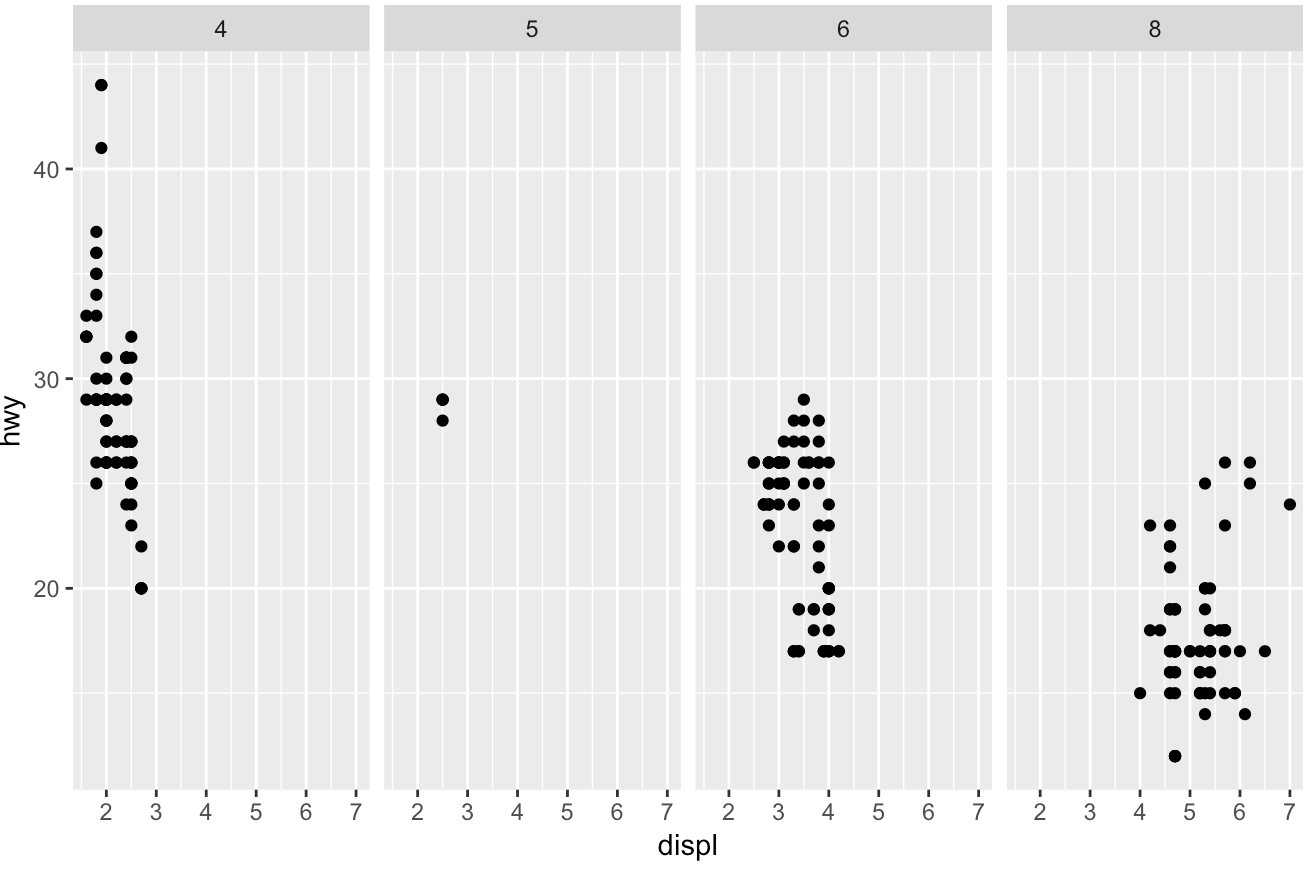
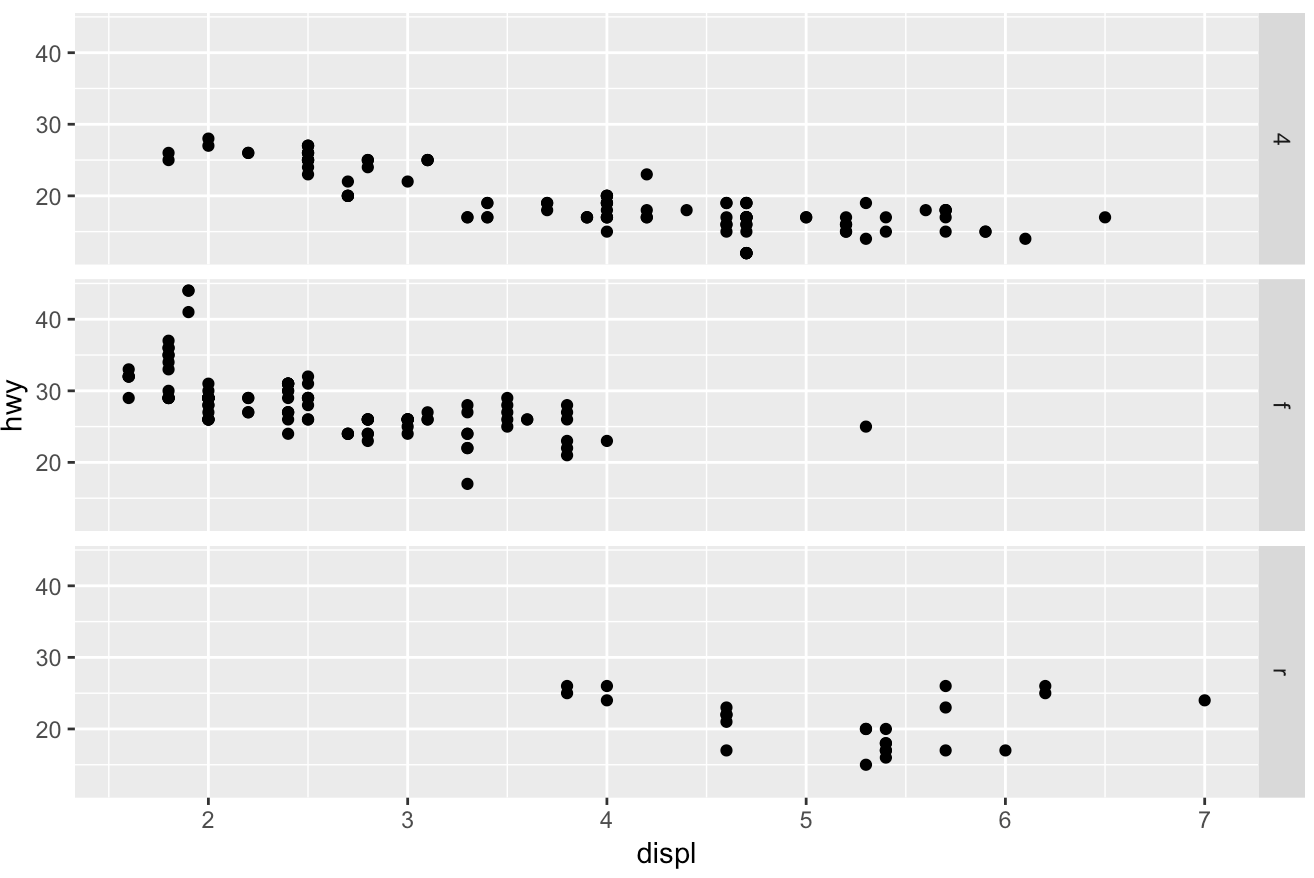
> ggplot(data = mpg) +

+ geom\_point(mapping = aes(x = drv, y = cyl))

###Exercises 3: What plots does the following code make? What does . do?

Answer: facet\_grid(drv ~ .) plots the facets on rows and empty on columns and facet\_grid(. ~ cyl) plots the facets on columns and empty on rows. "." indicates that the corresponding dimension is empty.

###Exercises 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 arguments?

Answer: nrow and ncol refer to the the number of rows and columns for the layout of the panel. Other options include 'switch' and 'strip.position'. The number of rows and columns of facet\_grid() are determined by the variables and we do not need to set them manually.

--------------------------

##Section 3.6.1

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

Answer: Line chart: geom\_line(). Boxplot: geom\_boxplot(). Histogram: geom\_histogram(). Area chart: geom\_area().

###Exercises 4: What does the se argument to geom\_smooth() do?

Answer: se argument controls if displaying confidence interval around smooth or not? (TRUE by default)

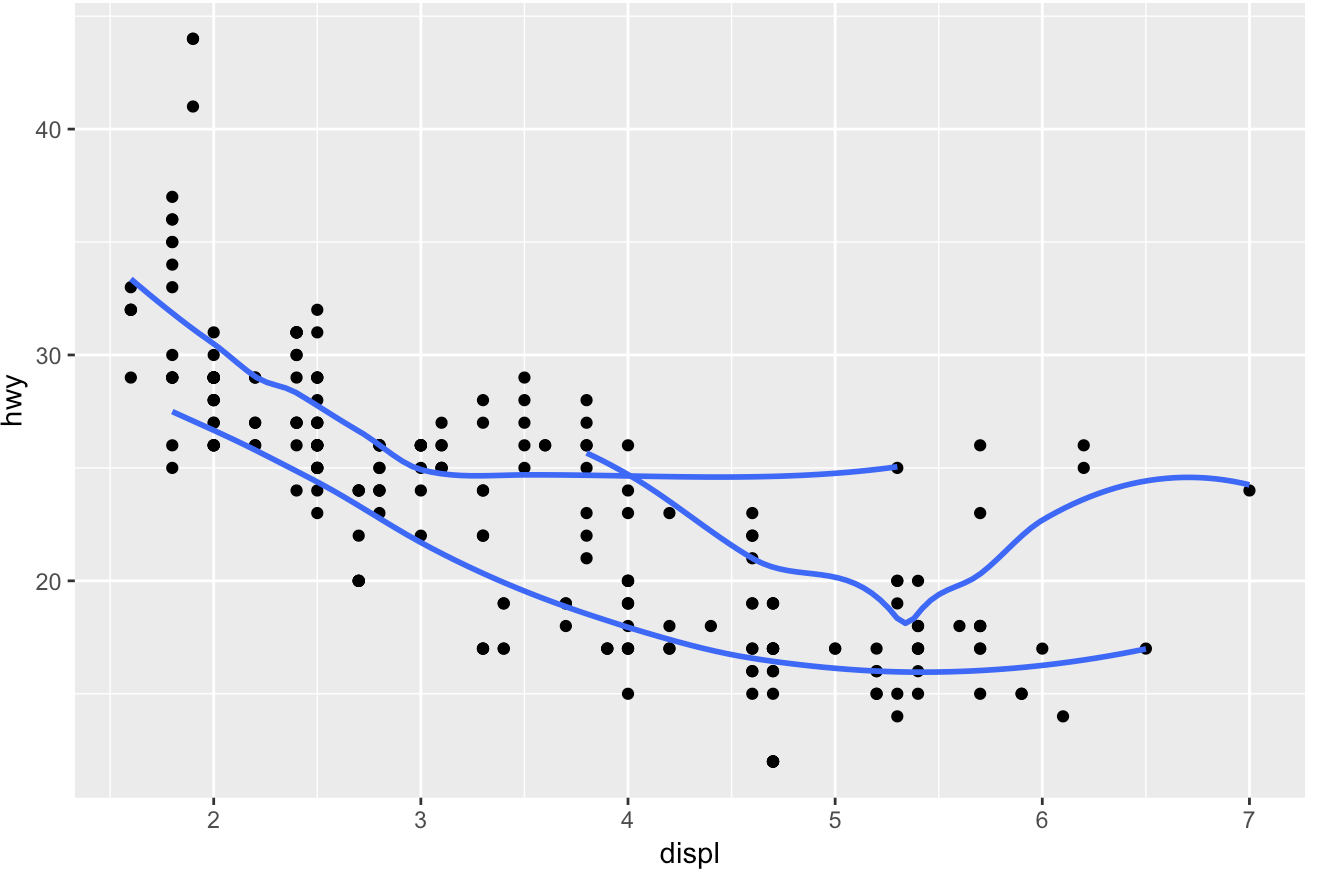
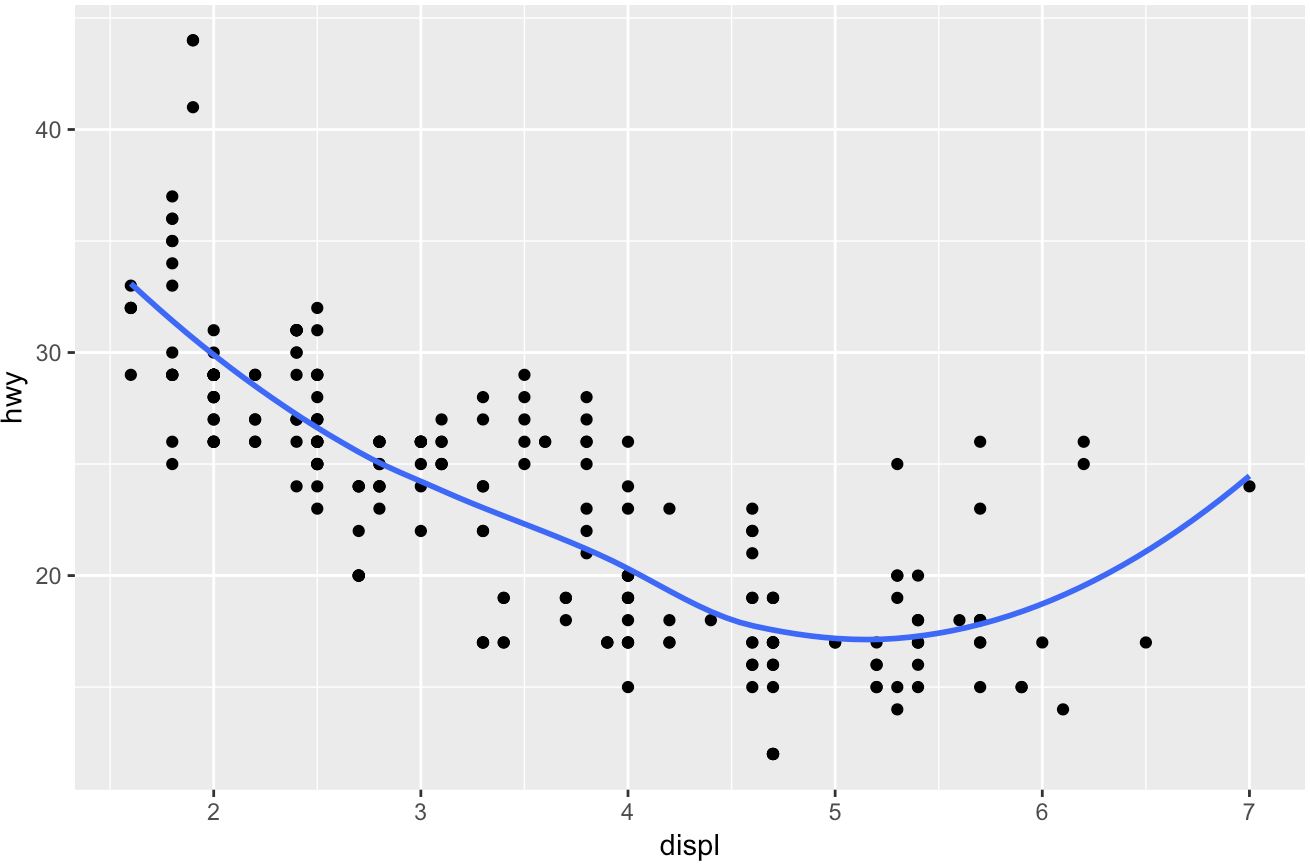
###Exercises 6: Recreate the R code necessary to generate the following graphs.

From left to right, up to down:

(1,1)

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

+ geom\_point() +

+ geom\_smooth(se=FALSE)

(1,2)

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

+ geom\_point() +

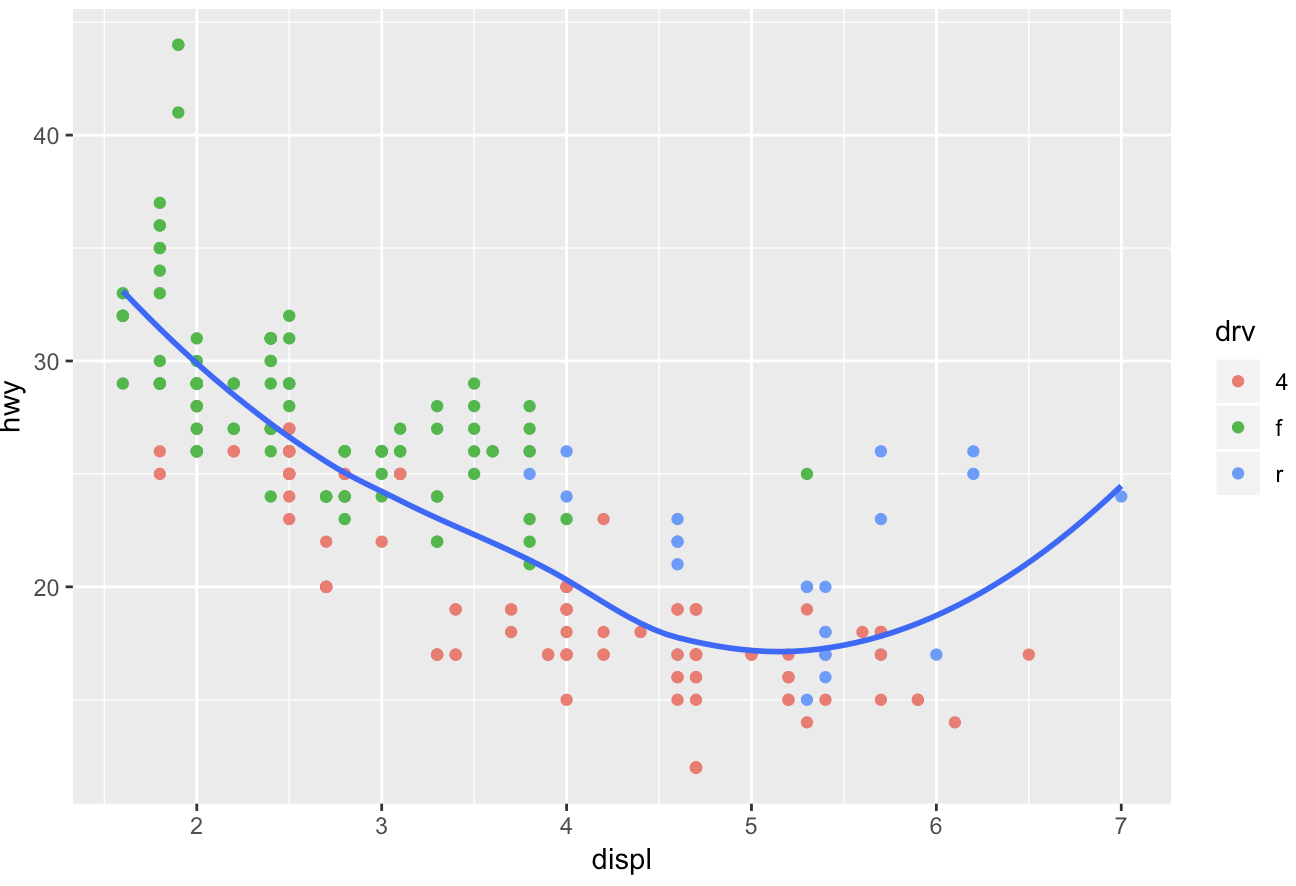
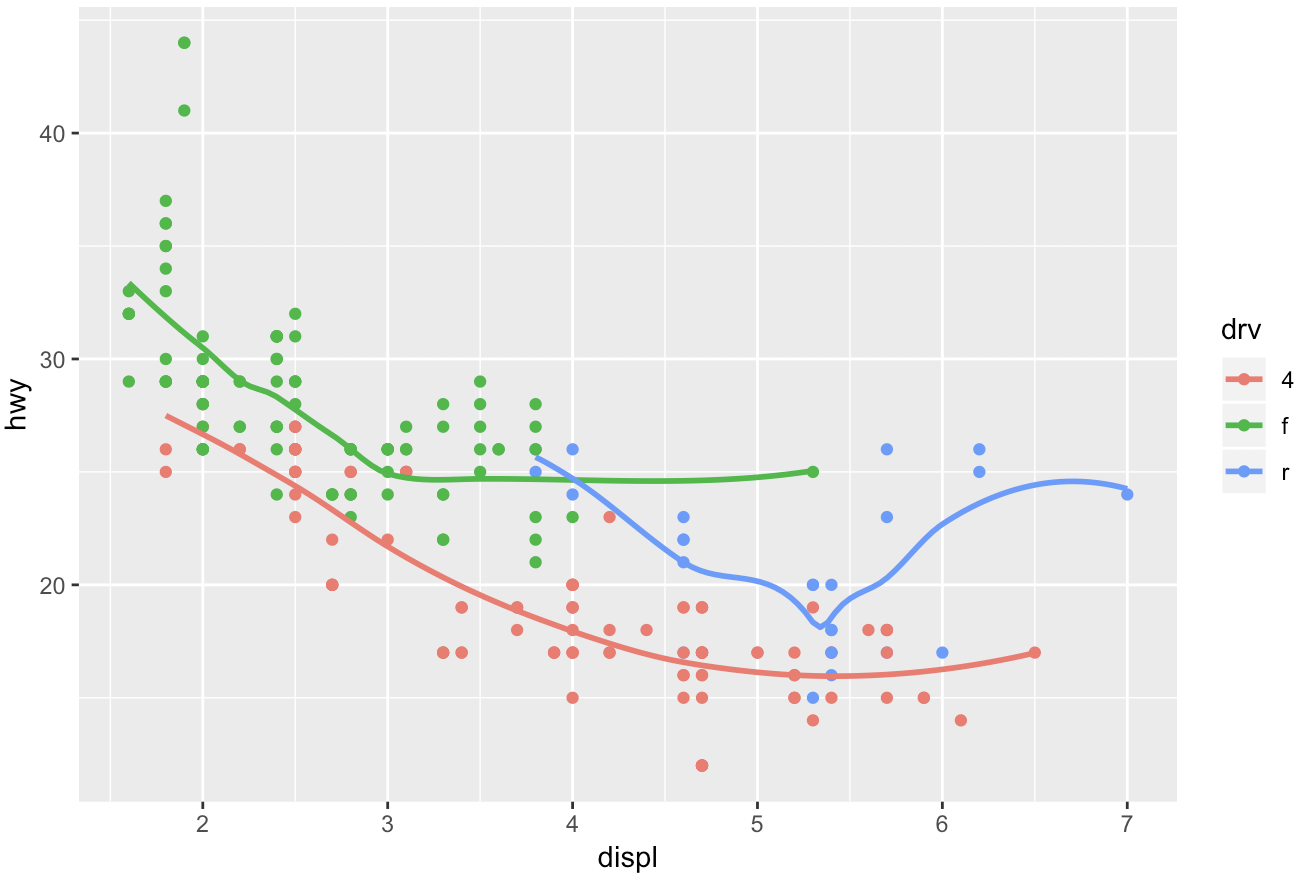
+ geom\_smooth(mapping=aes(group=drv),se=FALSE)

(2,1)

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

+ geom\_point() +

+ geom\_smooth(mapping=aes(group=drv),se=FALSE)



(2,2)

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

+ geom\_point(aes(color=drv)) +

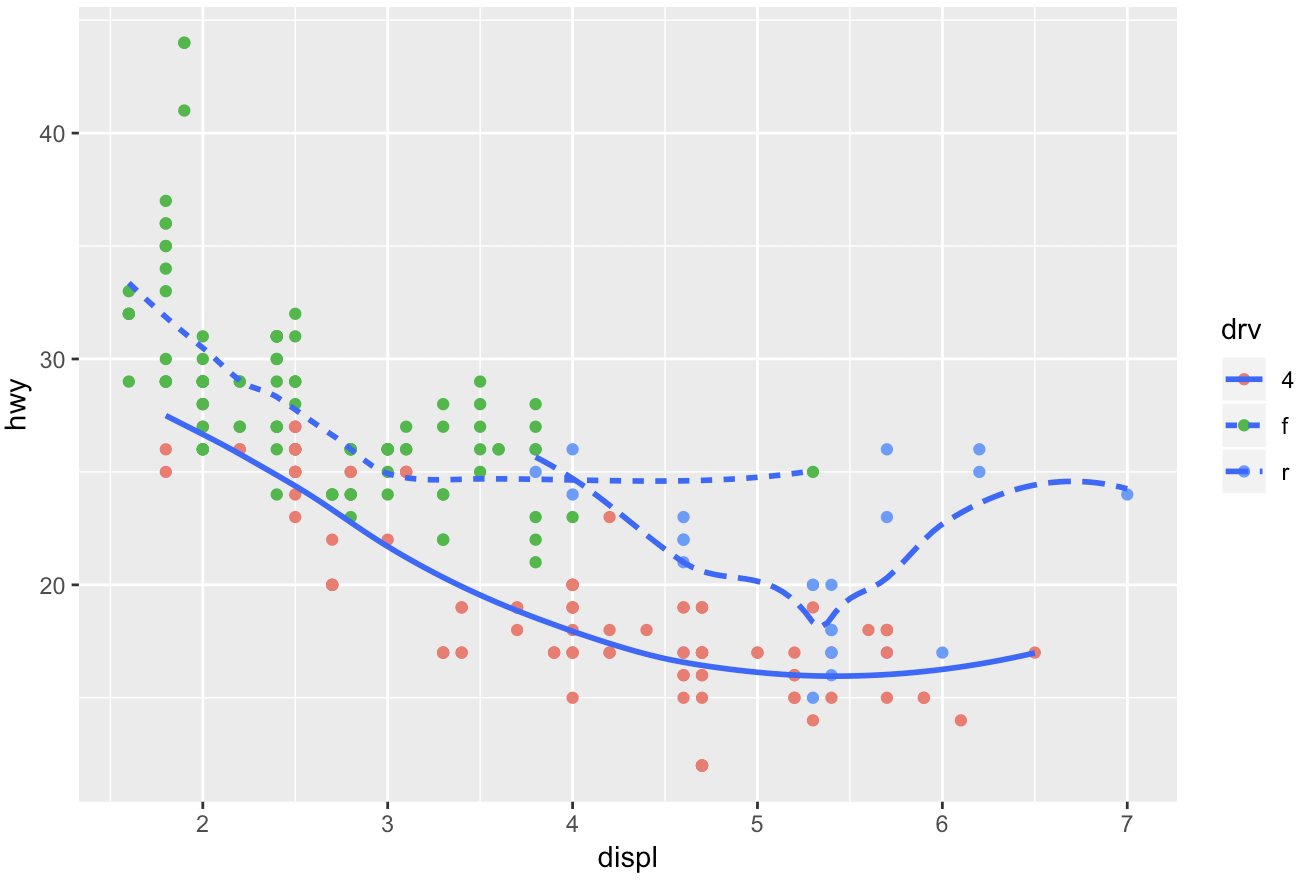
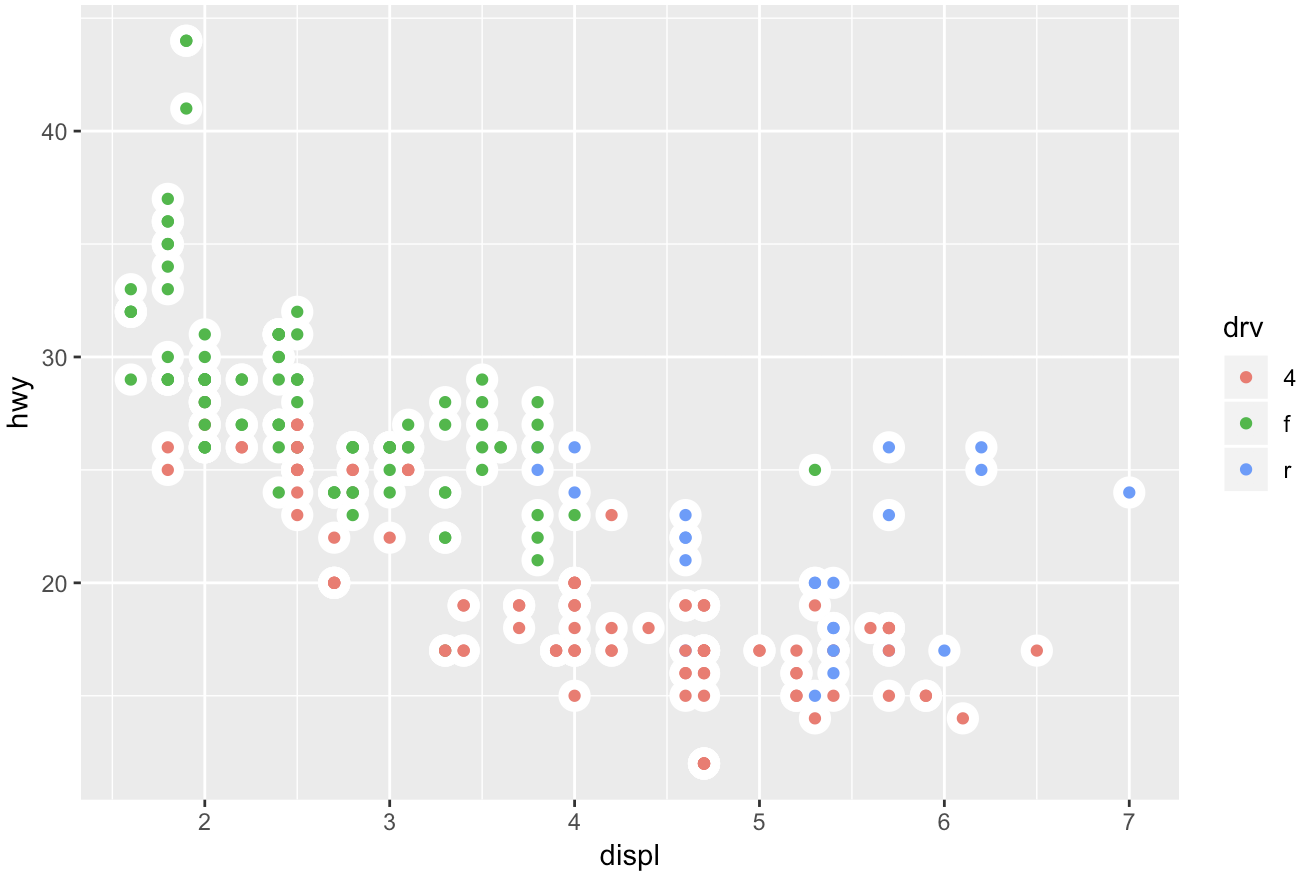
+ geom\_smooth(se=FALSE)

(3,1)

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

+ geom\_point(aes(color=drv)) +

+ geom\_smooth(mapping=aes(linetype=drv),se=FALSE)



(3,2)

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

+ geom\_point(color='white',size=5)+

+ geom\_point(aes(color=drv))

--------------------------

#Problem 2

##Section 5.2.4

###Exercises 1:

1. Had an arrival delay of two or more hours

Answer:

> filter(flights,arr\_delay>=120)

2. Flew to Houston (IAH or HOU

Answer:

> filter(flights,dest %in% c('IAH','HOU'))

3. Were operated by United, American, or Delta

Answer:

> filter(flights,carrier %in% c('UA','AA','DL'))

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

Answer:

> filter(flights,month %in% c(7,8,9))

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

Answer:

> filter(flights,arr\_delay>=120 & dep\_delay<=0)

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

Answer:

> filter(flights,dep\_delay>=60 & dep\_delay>arr\_delay+30)

7. Departed between midnight and 6am (inclusive)

Answer: Since midnight is denoted as 2400.

> filter(flights,dep\_time==2400 | (dep\_time>=0 & dep\_time<=600))

###Exercises 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() is a shortcut for x >= left & x <= right. We can simplify the code of (7) from the previous question:

> filter(flights,dep\_time==2400 | between(dep\_time,0,600))

###Exercises 3: How many flights have a missing dep\_time? What other variables are missing? What might these rows represent?

Answer: 8255 flights have a missing dep\_time. dep\_delay,arr\_time,arr\_delay are also missing. It may indicate that those flights are cancelled.

> filter(flights,is.na(dep\_time))

# A tibble: 8,255 x 19

--------------------------

##Section 5.3.1

###Exercises 1: How could you use arrange() to sort all missing values to the start? (Hint: use is.na()).

Answer:

> arrange(flights,desc(is.na(dep\_time)))

###Exercises 4: Which flights travelled the farthest? Which travelled the shortest?

Answer:

Farthest: > arrange(flights,desc(distance))

Shortest: > arrange(flights,distance)

--------------------------

##Section 5.5.2

###Exercises 1: Currently dep\_time and sched\_dep\_time are convenient to look at, but hard to compute with because they’re not really continuous numbers. Convert them to a more convenient representation of number of minutes since midnight.

Answer:

> flights\_new=mutate(flights,dep\_time\_minutes = (dep\_time %/% 100 \* 60 + dep\_time %% 100)%%1440,

+ sched\_dep\_time\_minutes = (sched\_dep\_time %/% 100 \* 60 + sched\_dep\_time %% 100)%%1440)

> select(flights\_new,dep\_time,sched\_dep\_time,dep\_time\_minutes,sched\_dep\_time\_minutes)

###Exercises 5: What does 1:3 + 1:10 return? Why?

Answer: When adding vectors with different length. Vector with smaller length would be added repeatedly to match the vector with larger length.

> 1:3 + 1:10

[1] 2 4 6 5 7 9 8 10 12 11

--------------------------

##Section 5.7.1

###Exercises 2: Which plane (tailnum) has the worst on-time record?

Answer: If we define worst on-time record as largest mean arr\_delay. From output, tialnum N844MH with mean arr\_delay 320 has the worst on-time record.

> flights %>%

+ group\_by(tailnum) %>%

+ filter(arr\_delay>0) %>%

+ summarize(mean\_delay = mean(arr\_delay)) %>%

+ arrange(desc(mean\_delay))

# A tibble: 3,874 x 2

tailnum mean\_delay

<chr> <dbl>

1 N844MH 320

2 N911DA 294

3 N922EV 276

4 N665MQ 270

5 N587NW 264

6 N828AW 229

7 N851NW 219

8 N928DN 201

9 N305AS 196

10 N907MQ 191

# ... with 3,864 more rows

###Exercises 3: What time of day should you fly if you want to avoid delays as much as possible?

If we are considering arr\_delay, we need to fly at hour 5 according to the output. The mean arr\_delay is 18.3 minutes.

> flights %>%

+ group\_by(hour) %>%

+ filter(arr\_delay>0) %>%

+ summarize(mean\_delay = mean(arr\_delay)) %>%

+ arrange(mean\_delay)

# A tibble: 19 x 2

hour mean\_delay

<dbl> <dbl>

1 5 18.3

2 6 22.8

3 7 24.7

4 9 28.5

5 8 28.7

6 10 31.8

7 11 32.8

8 12 33.0

9 23 33.5

10 13 36.5

11 14 42.0

12 15 43.1

13 22 44.0

14 16 47.7

15 21 48.7

16 18 49.6

17 17 50.7

18 20 50.8

19 19 53.0