Exercises

mehdi-271913

1/5/2021

RESET ENVIRONMENT

```
rm(list=ls())
```

EXERCISES

3.2.4

Q1. Run ggplot(data = mpg). What do you see?

> A plane, the first layer. On this plane the points or further layers are placed.

```
ggplot(data = mpg)
```

Q2. How many rows are in mpg? How many columns?

> 234 and 11

```
nrow(mpg)
```

```
## [1] 234
```

```
ncol(mpg)
```

```
## [1] 11
```

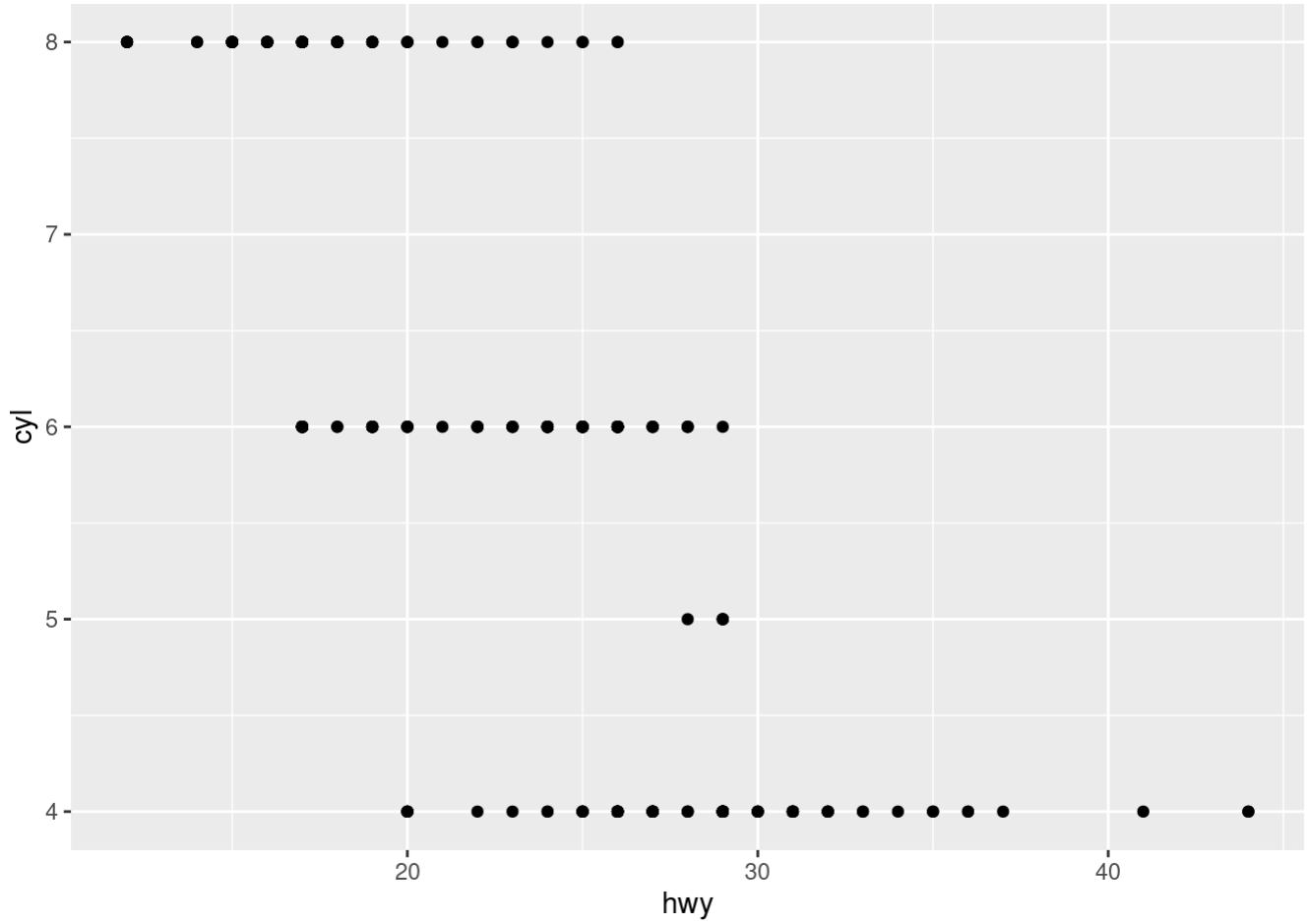
Q3. What does the drv variable describe? Read the help for ?mpg to find out.

> the type of drive train, where f = front-wheel drive, r = rear wheel drive, 4 = 4wd

```
?mpg
```

Q4. Make a scatterplot of hwy vs cyl.

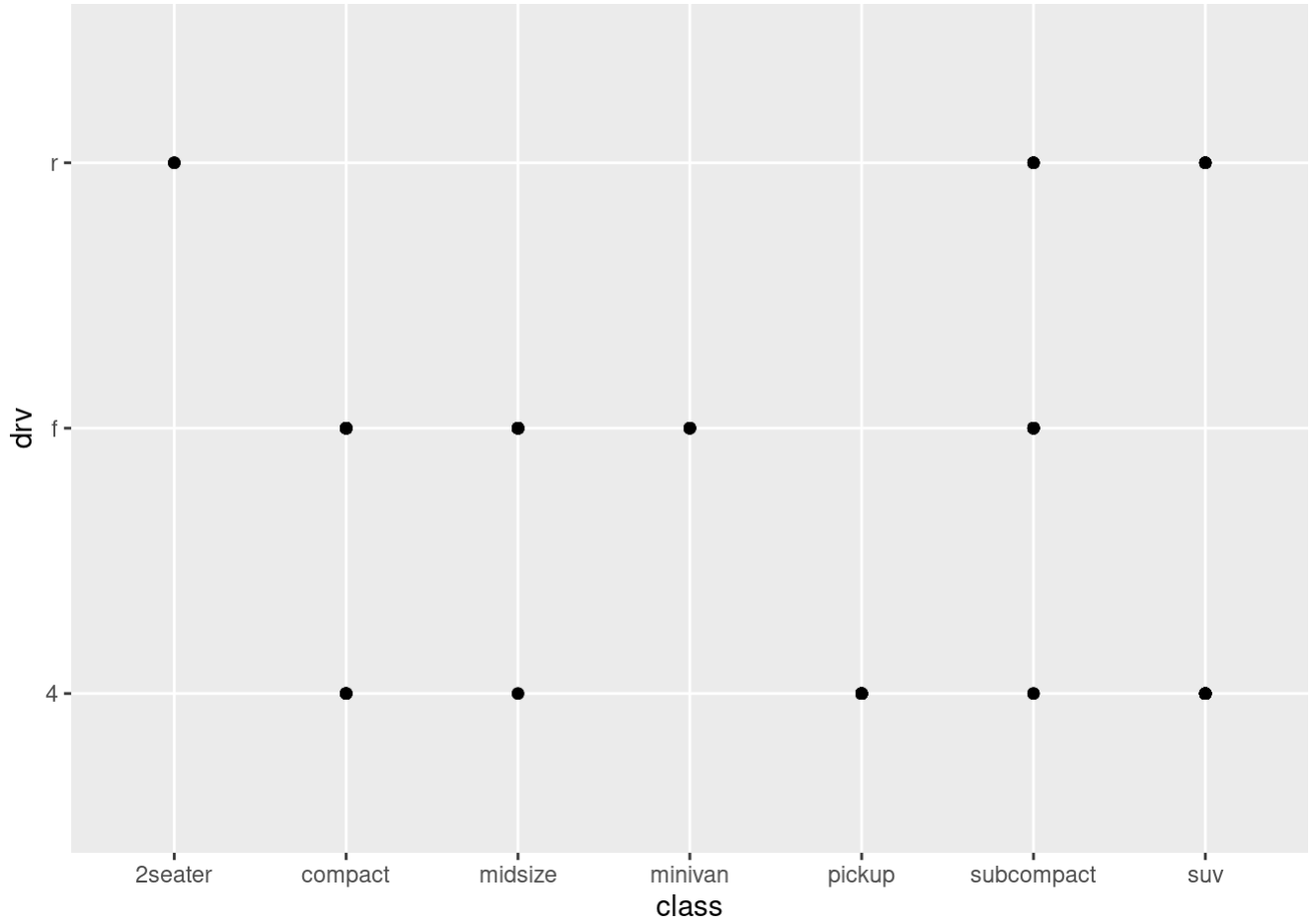
```
mpg %>%
  ggplot() +
  geom_point(mapping = aes(x = hwy, y = cyl))
```



Q5. What happens if you make a scatterplot of class vs drv? Why is the plot not useful?

a. You get 12 points, each point tells what is the car class and its drive-type.
b. Second question is highly biased. The plot could be useful to visualize which car classes in current collection do not have or have particular drive type.

```
mpg %>%
  ggplot() +
  geom_point(mapping = aes(x = class, y = drv))
```

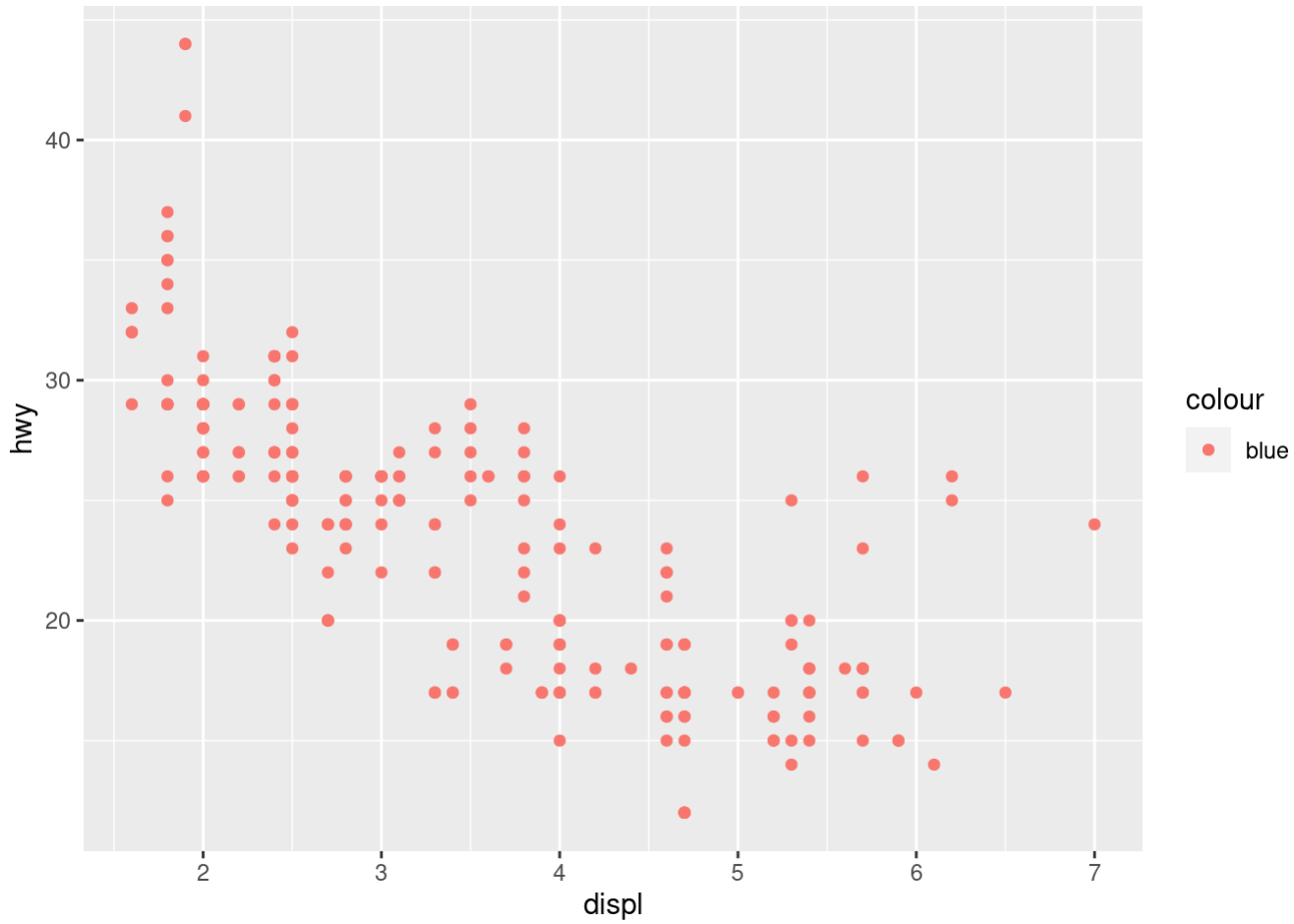


3.3.1

Q1. What's gone wrong with this code? Why are the points not blue?

> When we need to map aesthetic to the variable then we use it within aes, otherwise if no variable is assigned to aesthetics mapping then it means aesthetics is applied to NULL, and you won't see it on graph.

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



Q2. 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?

Categorical
- Model - cyl - Manufacturer - trans - drv - fl - class

Continuous
- displ - year - cty - hwy

> Using R

there is no straightforward way to tell if variable is categorical or continuous, except few trics.

By seeing data type, factors and strings are mostly categorical

```
str(mpg)
```

```
## tibble [234 × 11] (S3: tbl_df/tbl/data.frame)
##  $ manufacturer: chr [1:234] "audi" "audi" "audi" "audi" ...
##  $ model       : chr [1:234] "a4" "a4" "a4" "a4" ...
##  $ displ       : num [1:234] 1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
##  $ year        : int [1:234] 1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
##  $ cyl         : int [1:234] 4 4 4 4 6 6 6 4 4 4 ...
##  $ trans       : chr [1:234] "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
##  $ drv         : chr [1:234] "f" "f" "f" "f" ...
##  $ cty         : int [1:234] 18 21 20 21 16 18 18 18 16 20 ...
##  $ hwy         : int [1:234] 29 29 31 30 26 26 27 26 25 28 ...
##  $ fl          : chr [1:234] "p" "p" "p" "p" ...
##  $ class       : chr [1:234] "compact" "compact" "compact" "compact" ...
```

By seeing unique values, and how much they differ in quantity from total rows or range of values, min vs max value etc.

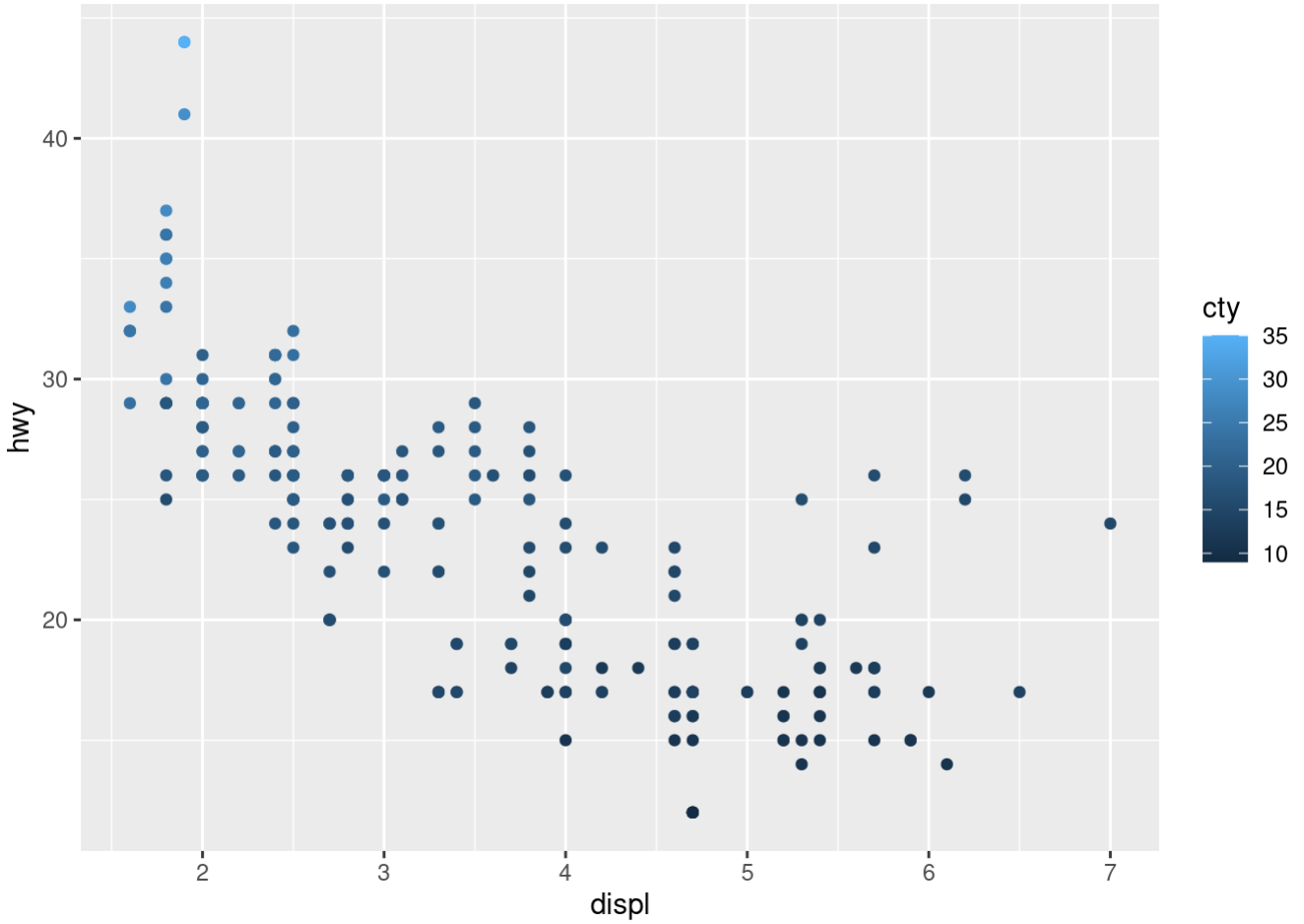
```
sapply(colnames(mpg), function(x) class(mpg[[x]]))
```

```
## manufacturer        model        displ         year          cyl        trans 
##  "character"  "character"    "numeric"    "integer"    "integer"  "character" 
##          drv          cty          hwy           fl        class 
##  "character"    "integer"    "integer"  "character"  "character"
```

Q3. Map a continuous variable to color, size, and shape. How do these aesthetics behave differently for categorical vs. continuous variables?

A continuous variable can not be mapped to shape.
When continuous variable mapped to color it outputs a color gradient (continuous quantity) as reference for different shades of color for the points of different value used as aesthetic variable.
When continuous variable mapped to size, it automaticaaly got binned, i.e categories are made out of it.
For categorical variables it simply use classes for shape, color and size with legend mapping given for each class on the plot.

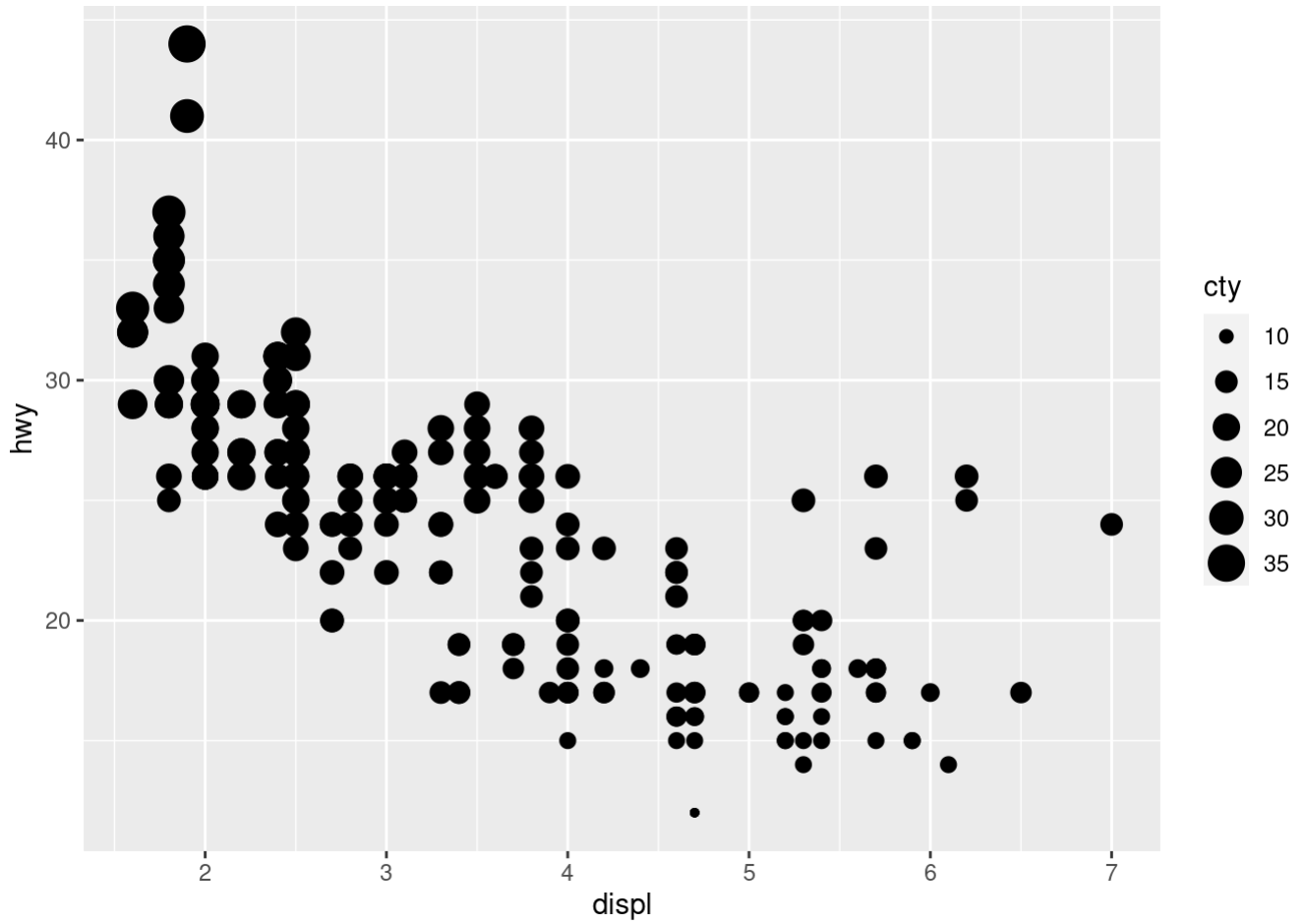
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, color=cty))
```



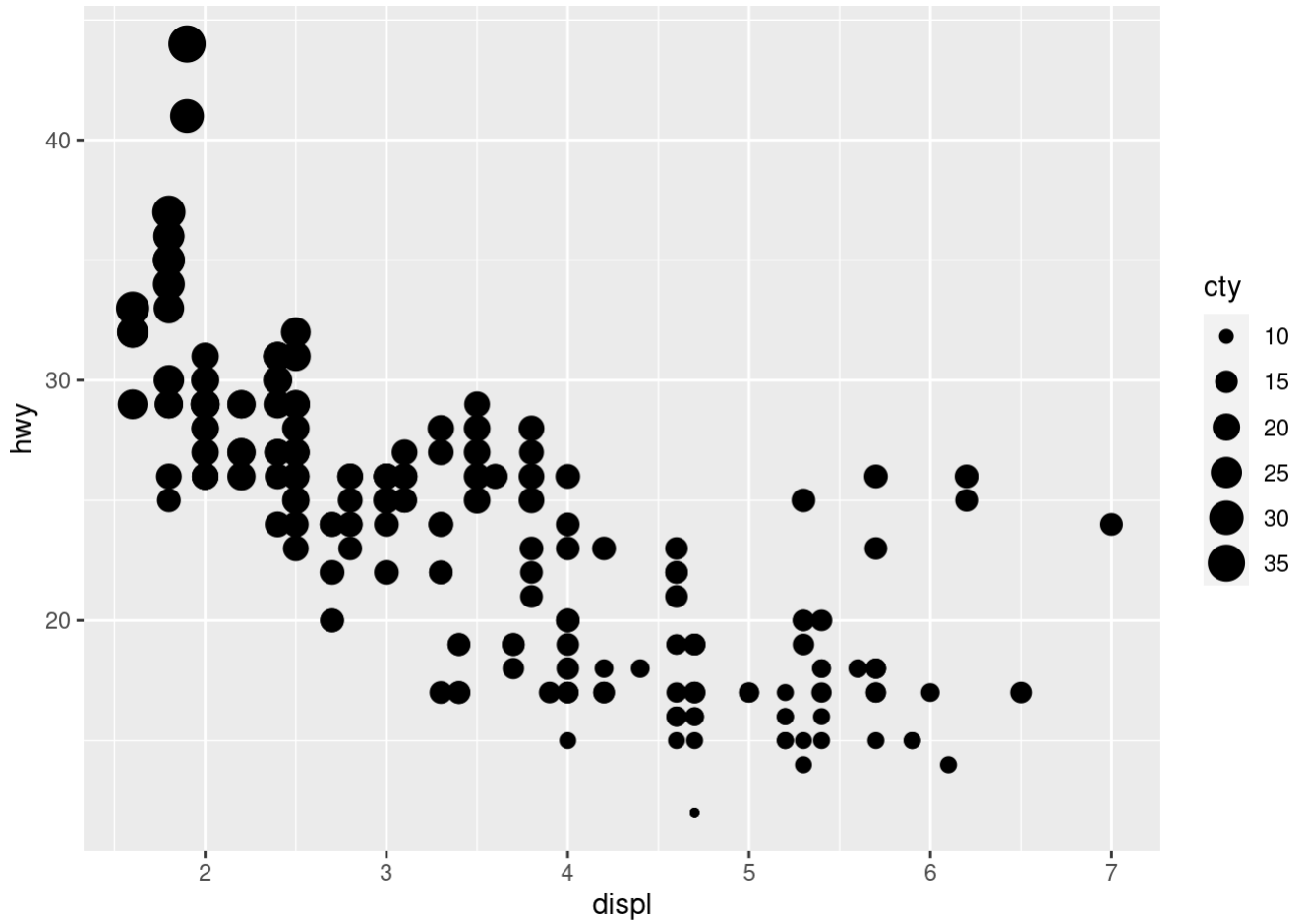
Below code won't work.

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

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




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

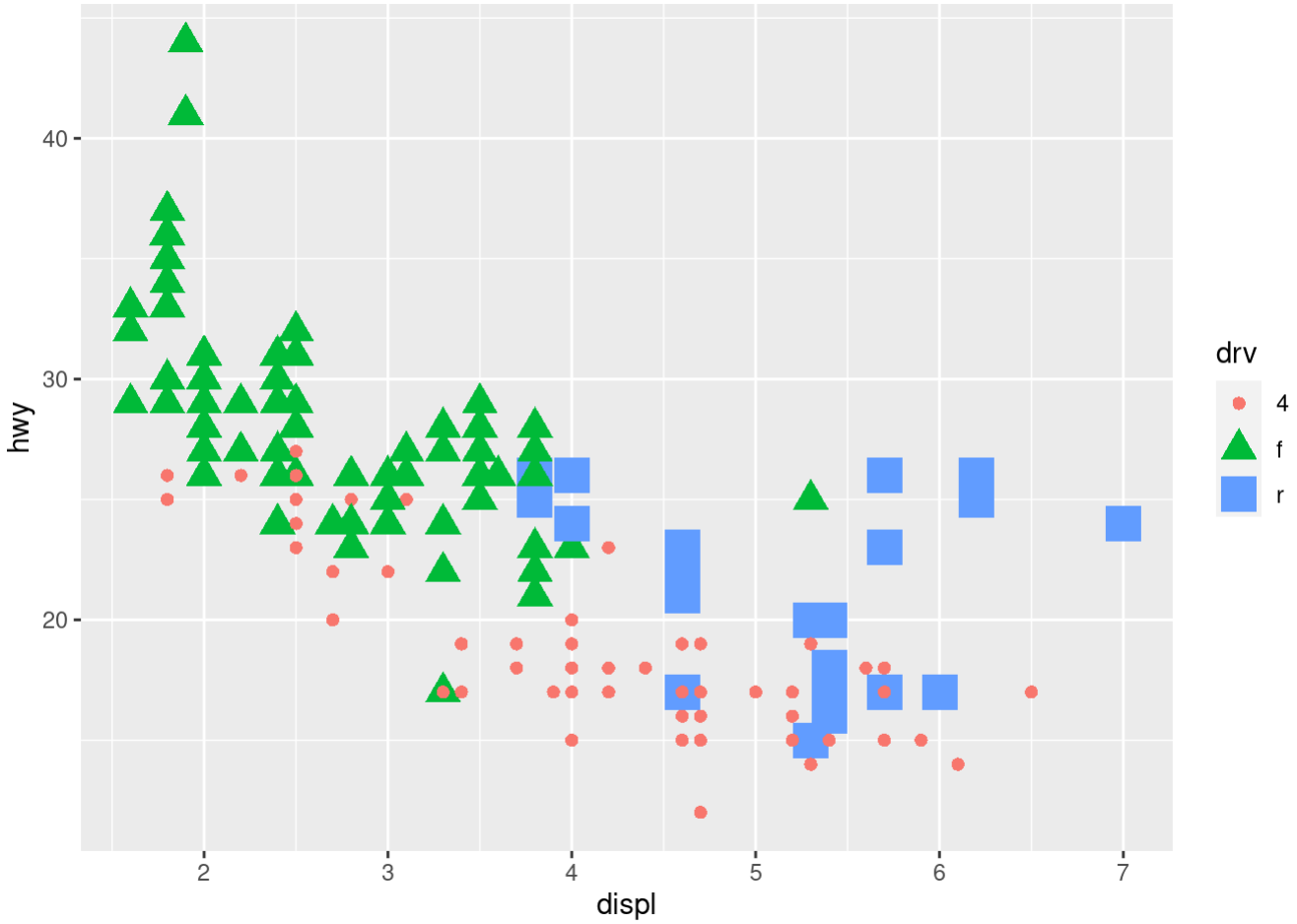


Q4. What happens if you map the same variable to multiple aesthetics?

> All will be applied.

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

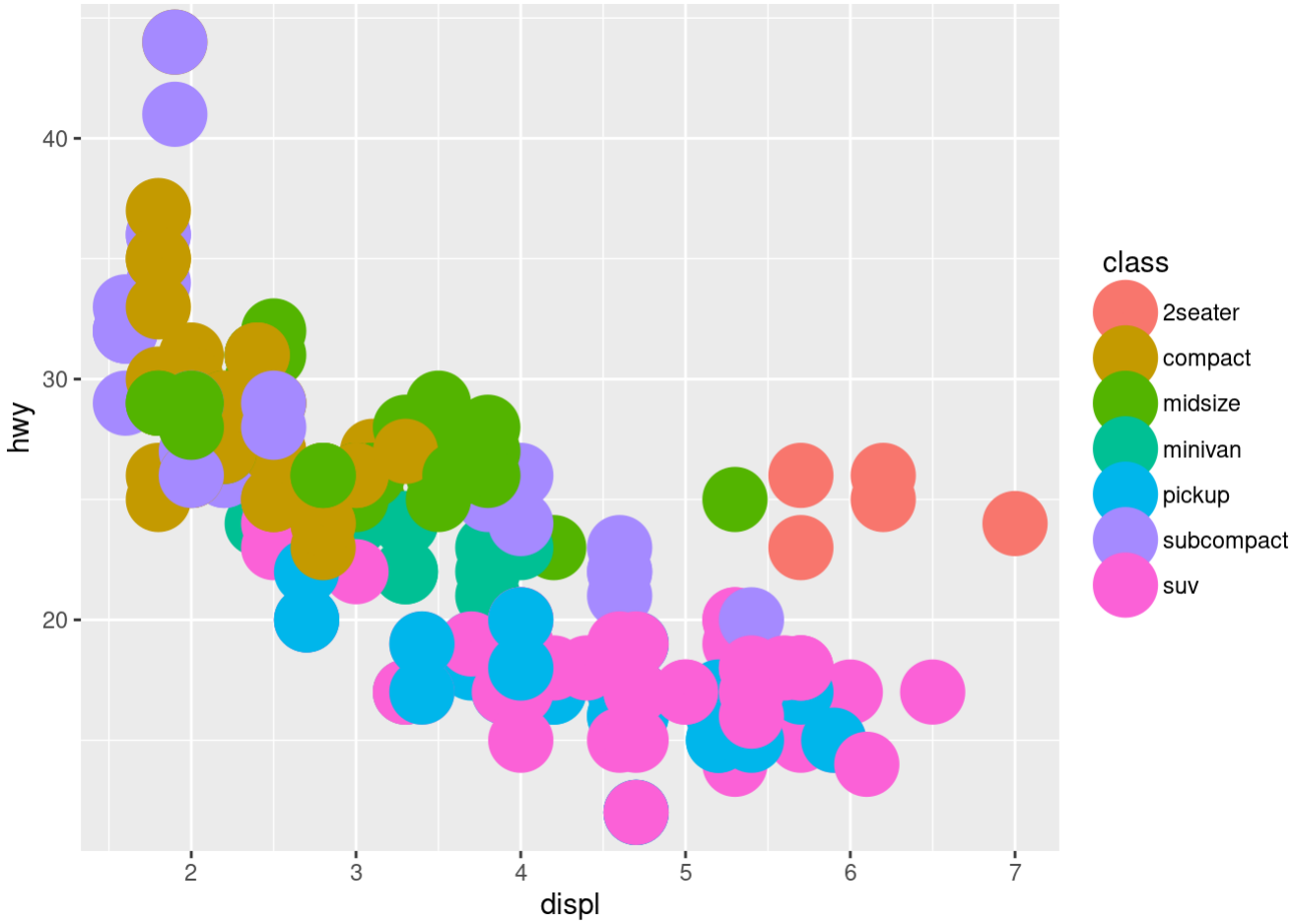
```
## Warning: Using size for a discrete variable is not advised.
```



Q5. What does the stroke aesthetic do? What shapes does it work with? (Hint: use ?geom_point)

> Controls the border size of points, in mm.

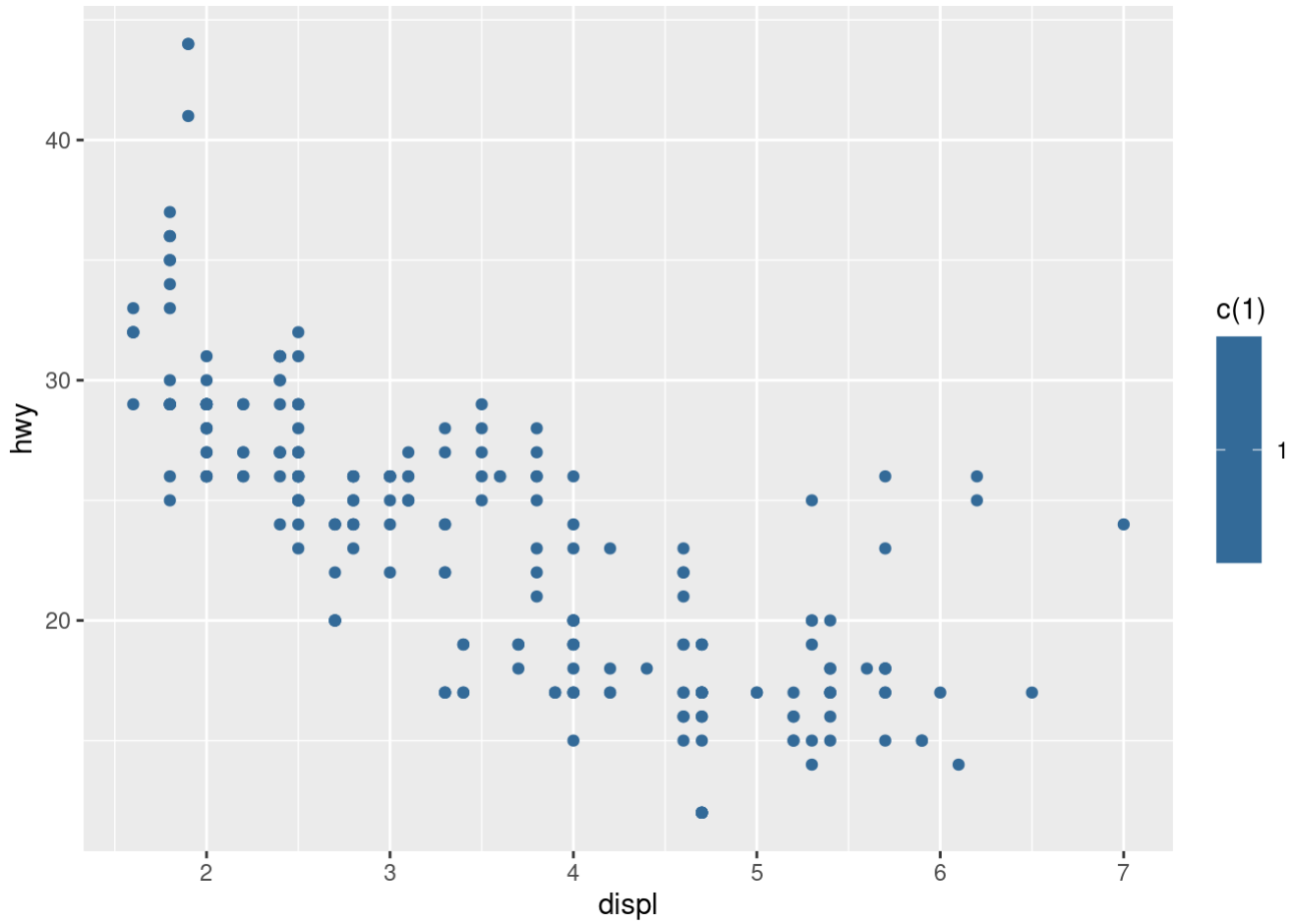
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, color=class), size=1,stroke=8)
```



Q6. What happens if you map an aesthetic to something other than a variable name, like aes(colour = displ < 5)? Note, you'll also need to specify x and y.

In the given case it is a function which returns TRUE or FALSE for each point (x,y), point where displ value is less then 5 is TRUE colored and others will be FALSE colored.
It must use col names of the dataset.
It is possible to have something in aesthetic field which is not any of the column, but its length must be 1, eg if it is list of string, then there must not be more then two items in the list

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, color=c(1)))
```

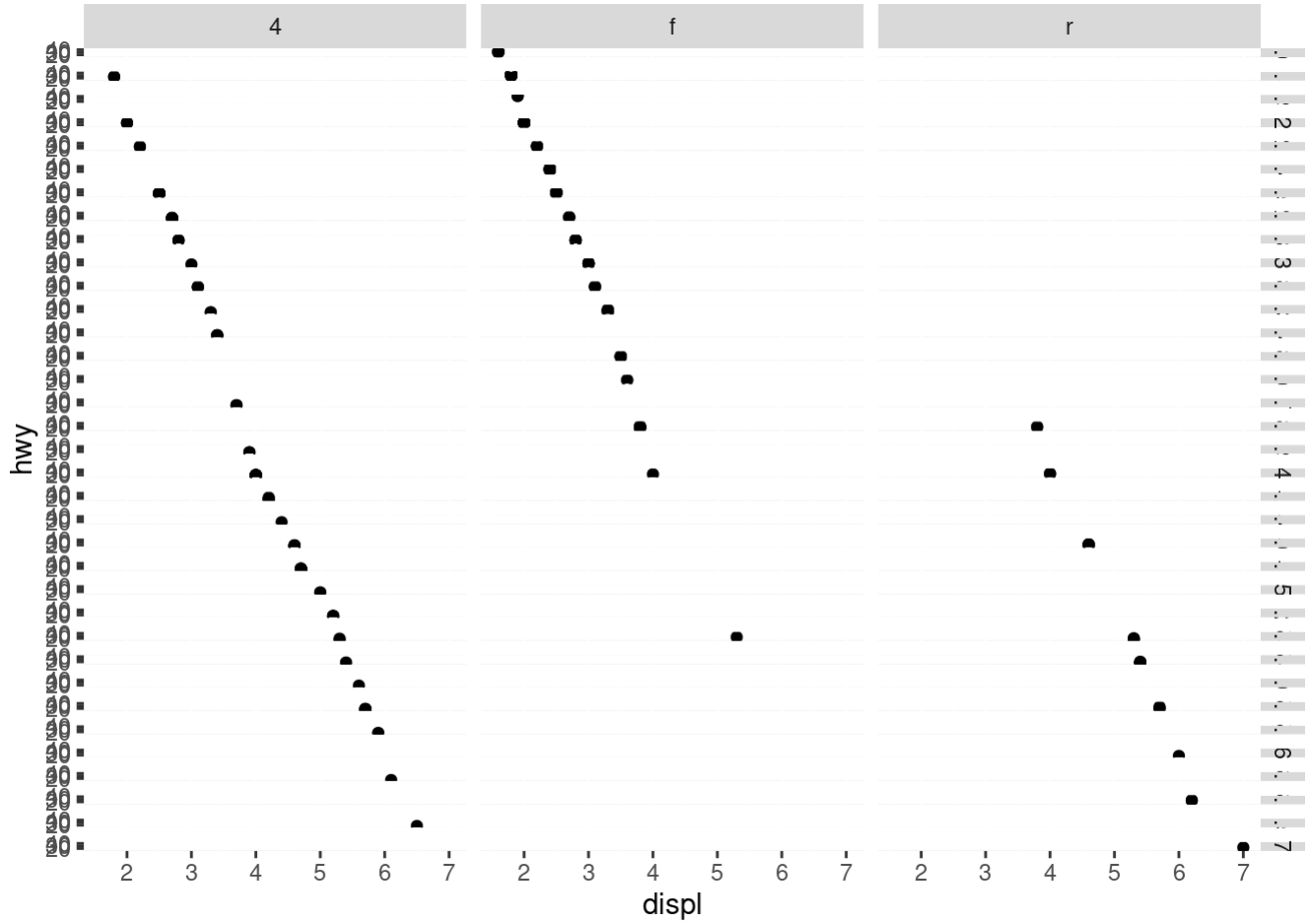


3.5.1

Q1. What happens if you facet on a continuous variable?

> It finds the minuimum and maximum value oft that continuous variable and draw a grid of that range, and use it for plotting.

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



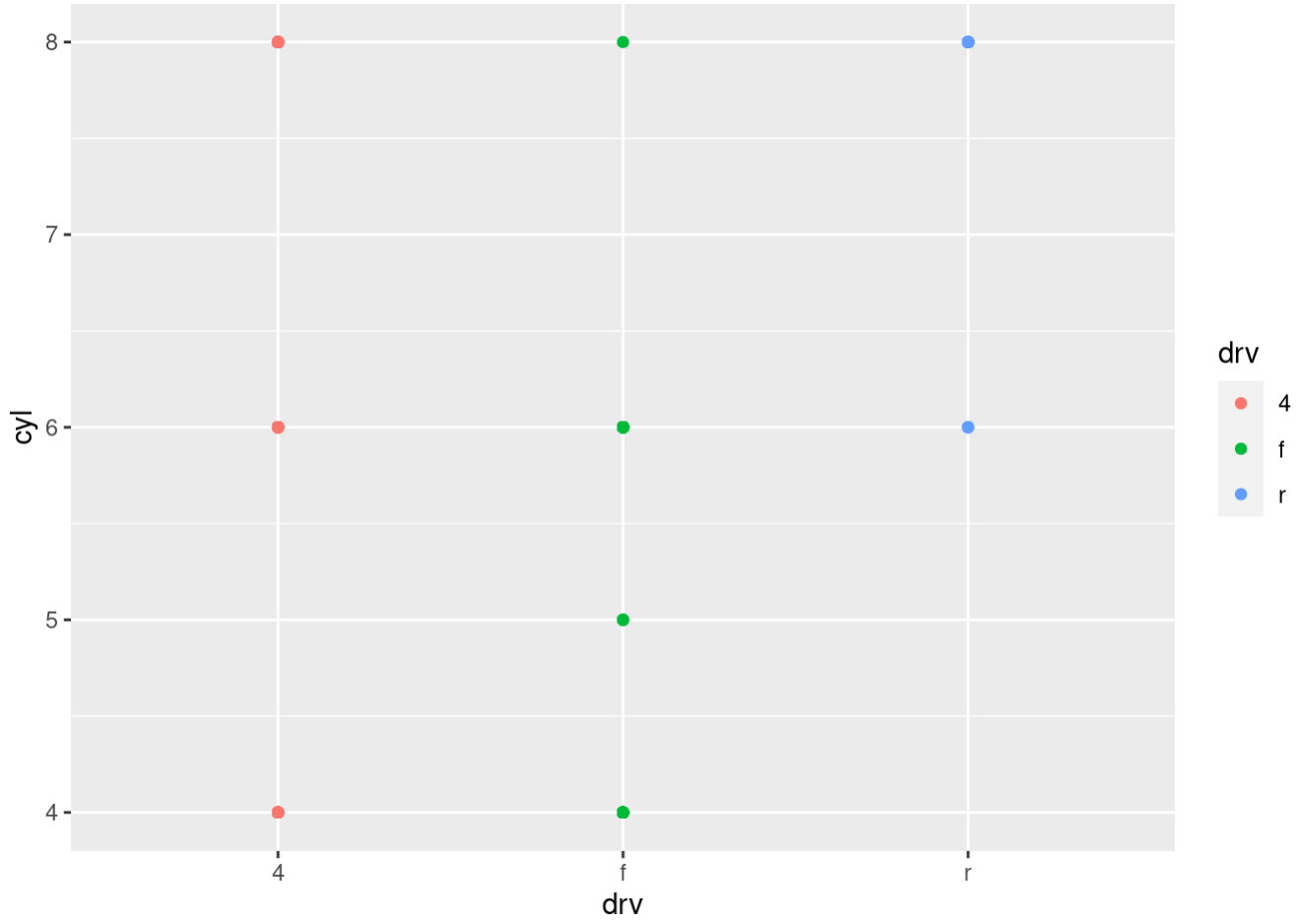
Q2. What do the empty cells in plot with facet_grid(drv ~ cyl) mean? How do they relate to this plot?

No x belongs to this particular facet (combination of variable) class.

How relate to this plot?

There is no point for facet(drv==r & cyl <5)

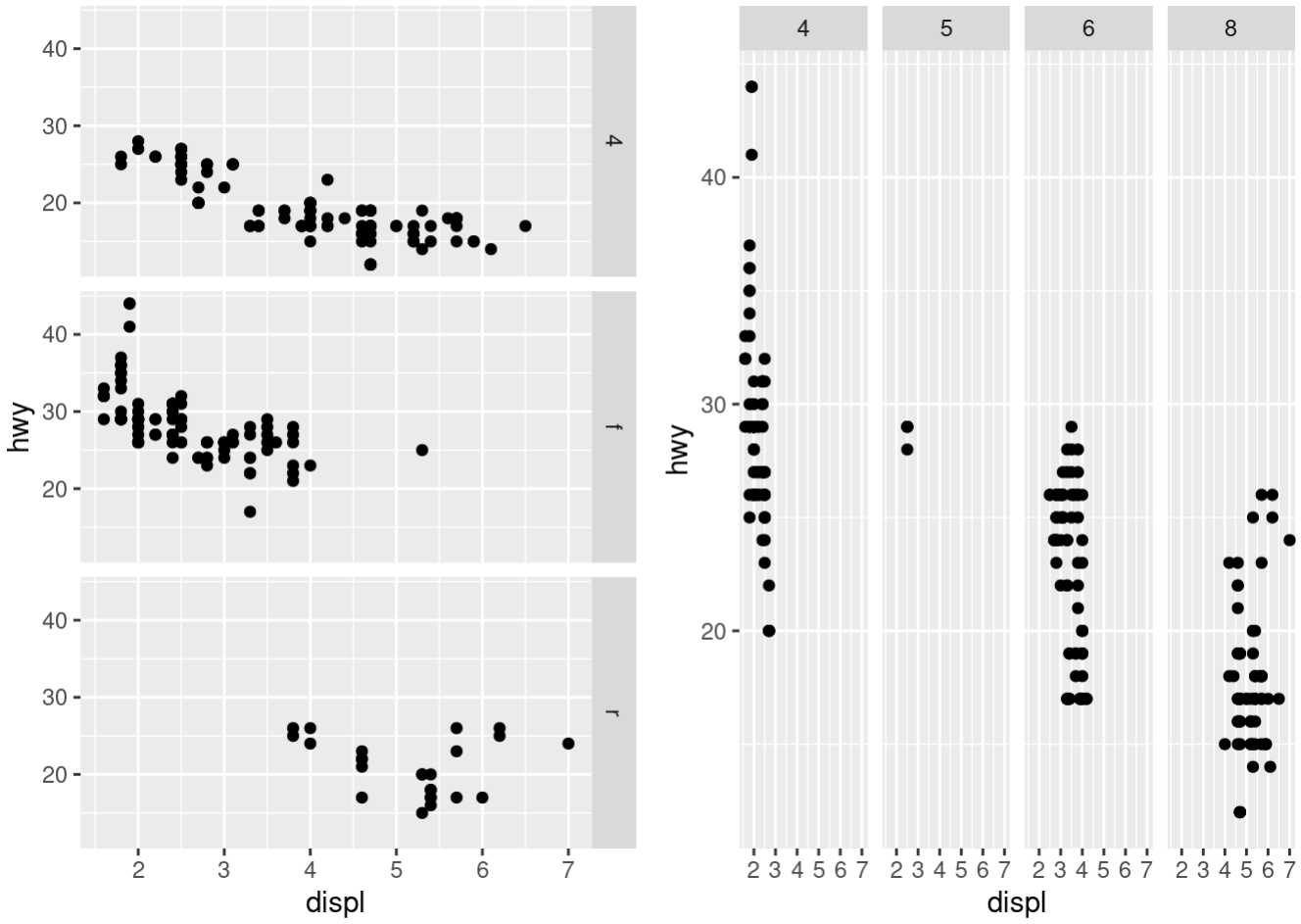
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = drv, y = cyl, col=drv))
```



Q3. What plots does the following code make? What does . do?

First one uses drv as Facet variable and make facets paralle of x-axis
Second one uses cyl as Facet variable and make facets paralle of y-axis

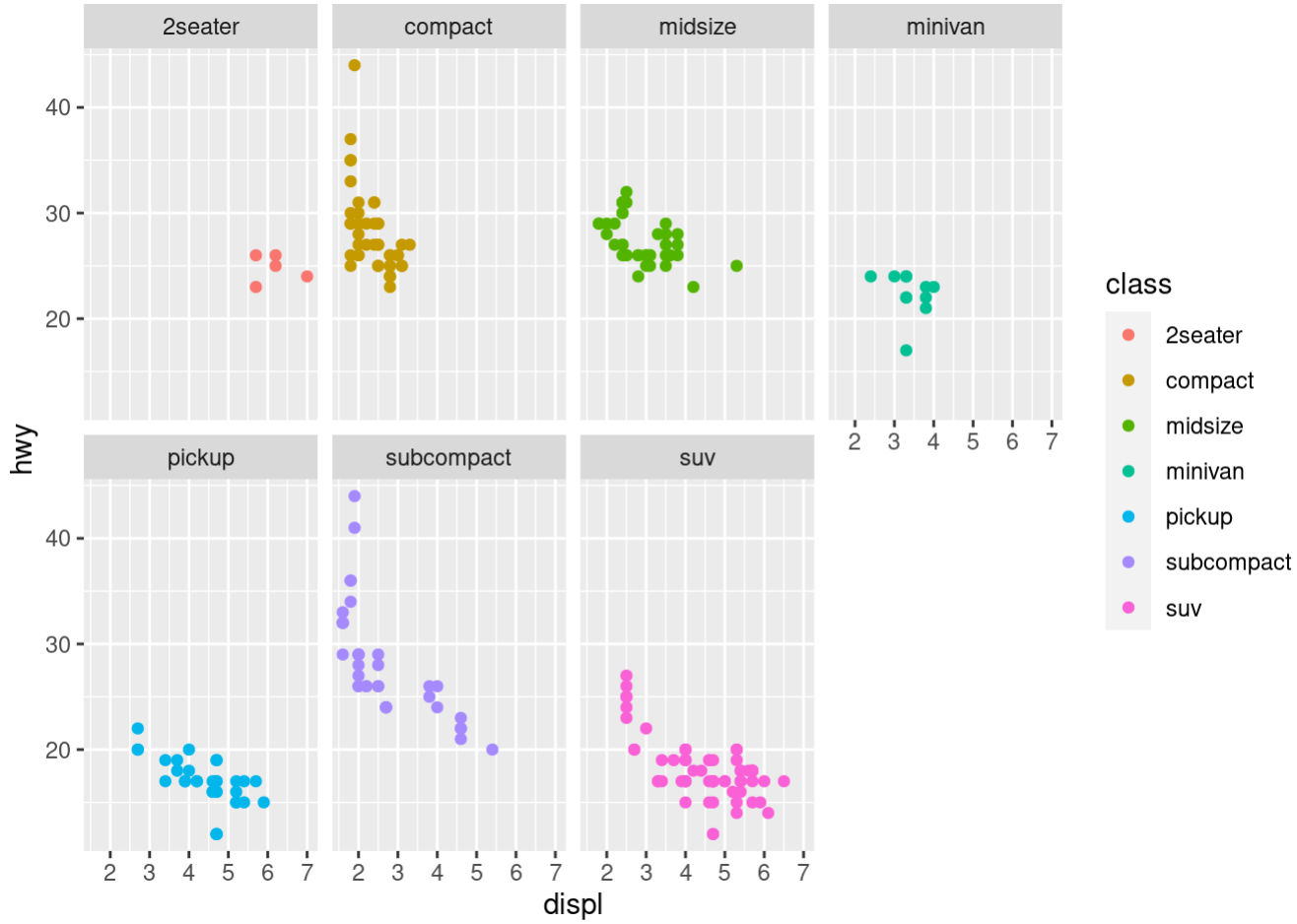
```
p1 = ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_grid(drv ~ .)

p2 = ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_grid(. ~ cyl)

plot_grid(p1,p2)
```



Q4. Take the first faceted plot in this section

Faceting helps to better visualize the relation between x and y in presence of a thir variable, mostly categorical.
Disadvantage : Shrinking of x-axis could make it difficult to visualize point values.
Larger Dataset : Shrinking of axis would be considerable and points would merge into one anothter for each facet.

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



Q5. 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?

nrow : how many rows for total Facets be created parallel to y-axis or rows parallel to x-axix
ncol : how many rows for total Facets be created parallel to x-axis or cols parallel to y-axis
other options : scales, shrink, labeller, as.table, switch, drop, dir, strip.position.
Why facet_grid() have no nrow/ncol : Facet_grid is specialized case of facet_wrap where there is just one plane whose rows and columns are determined by the number of facet variables.

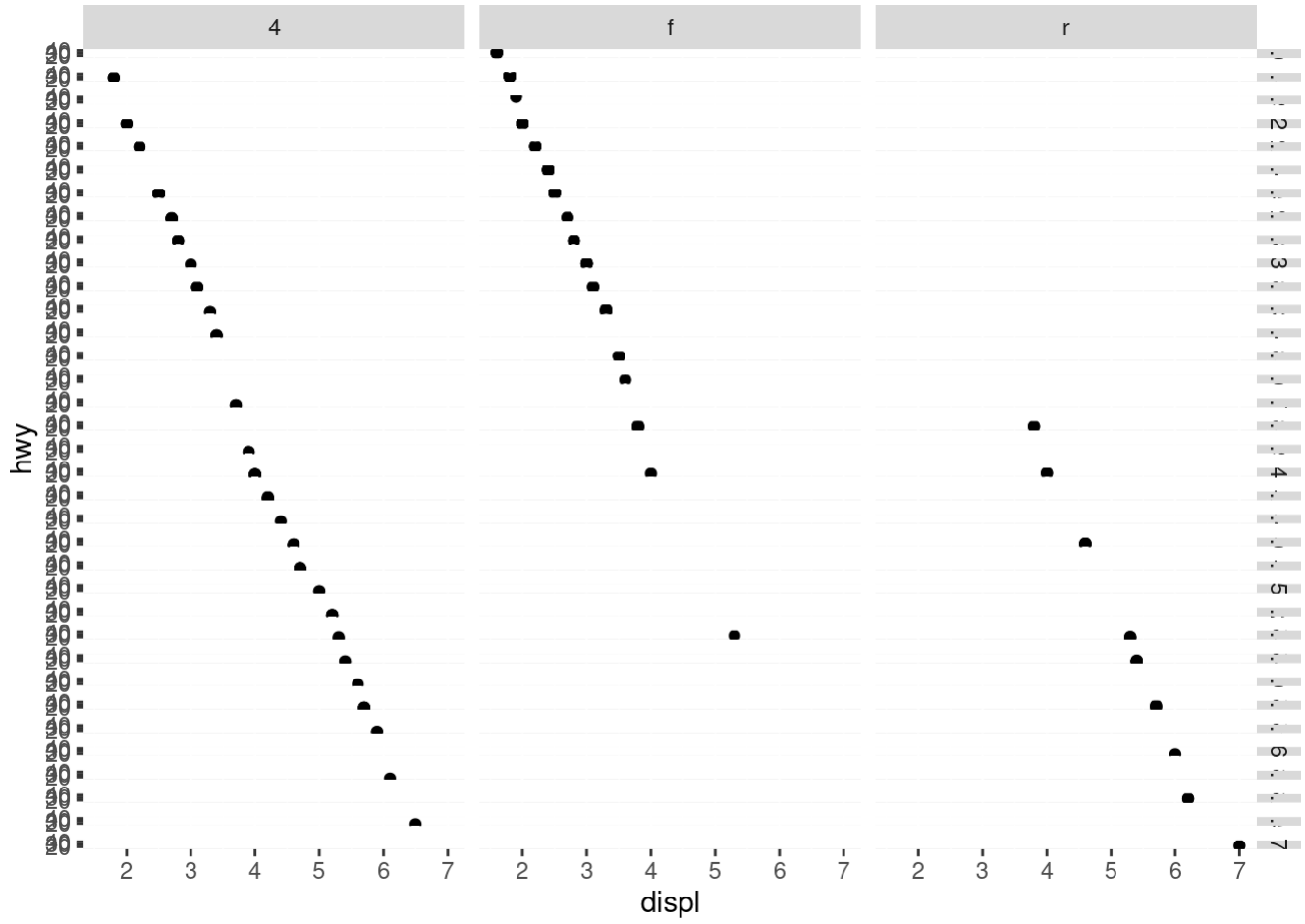
```
?facet_wrap
```

Q6. When using facet_grid() you should usually put the variable with more unique levels in the columns. Why?

To make it easy to visualize, otherwise space taken by just one facet category would be so much that entire plot would shrink too much to make it fit to the plane, making everythin too small.

Take below plot as example, and exchange values of facet vars and then again observe

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



3.6.1

Q1. What geom would you use to draw a line chart? A boxplot? A histogram? An area chart?

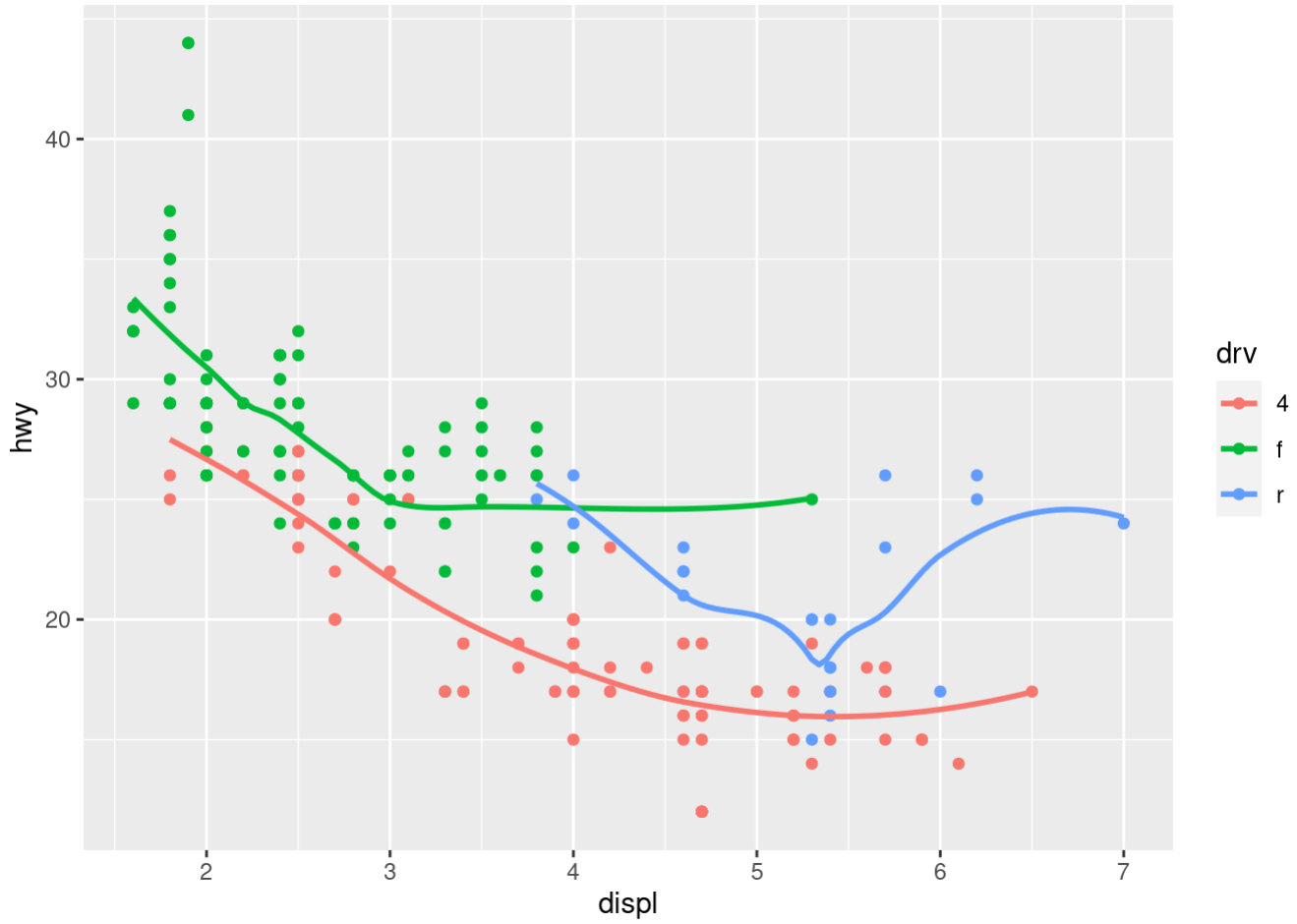
geom_line, geom_box, geom_historgam and geom_area

Q2. Run this code in your head and predict what the output will look like. Then, run the code in R and check your predictions.

> Simple chrt with points and smooth line but not the confidence interval i.e se=FALSE, se is show error.

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

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



Q3. What does show.legend = FALSE do? What happens if you remove it? Why do you think I used it earlier in the chapter?

> It does not show legend, hence output appears to be a little big and clean.

Q4. What does the se argument to geom_smooth() do?

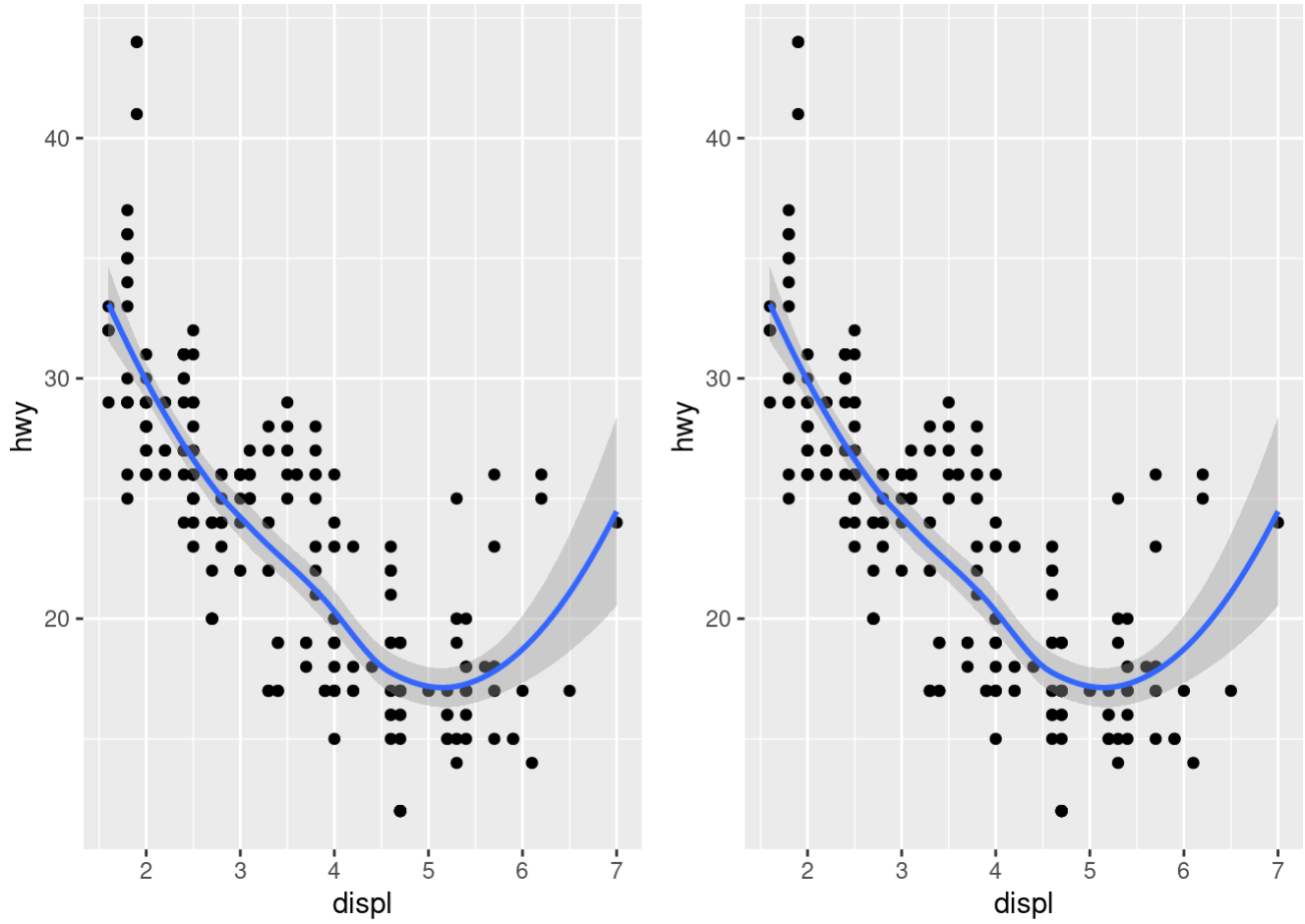
> Confidence interval i.e grey area is not shown. se stands for show error.

Q5. Will these two graphs look different? Why/why not?

> No, because they are using the same points for plotting with same aesthetics.

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

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

plot_grid(p1,p2,nrow=1, ncol=2)
```

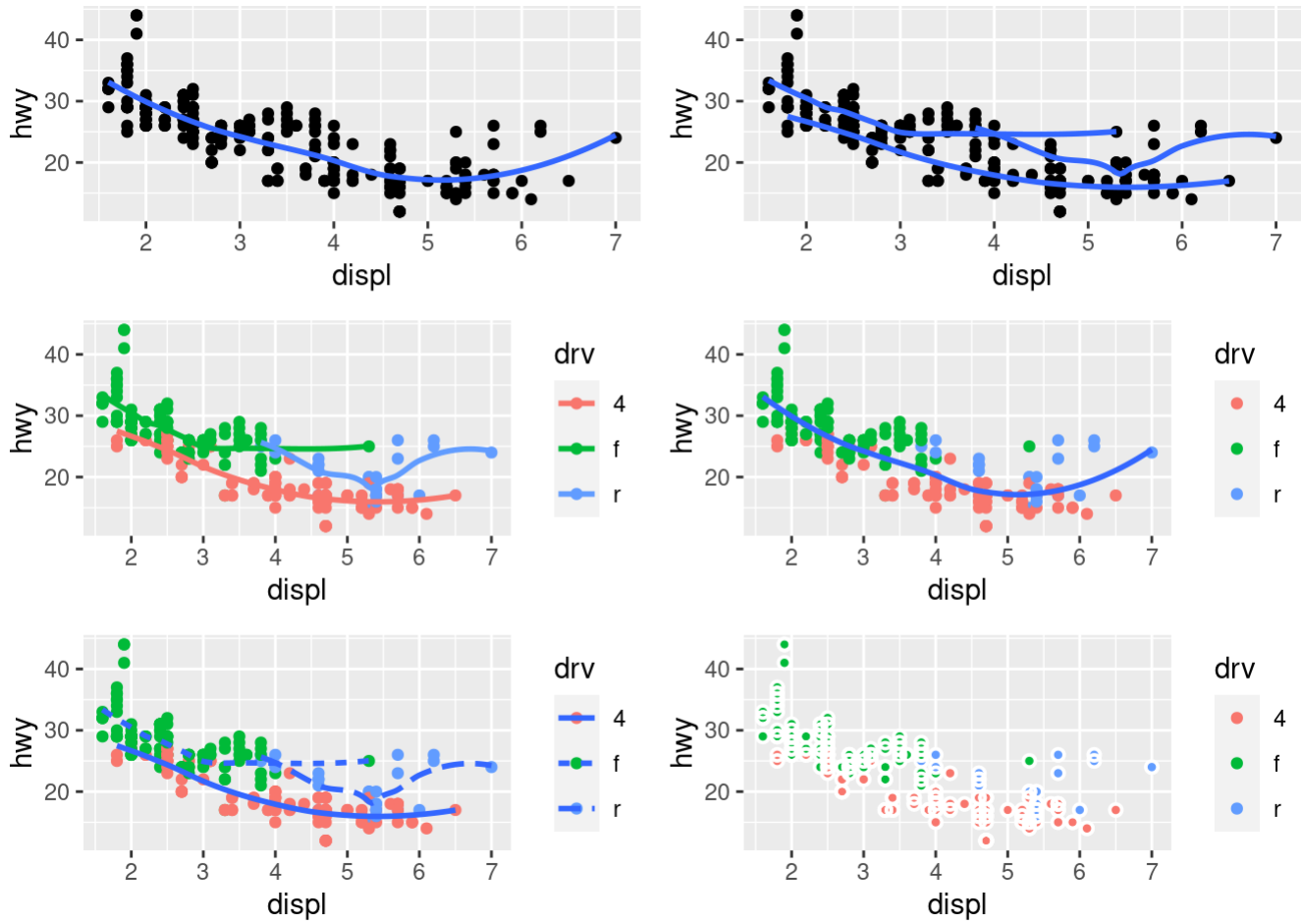
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Q6. Recreate the R code necessary to generate the following graphs.

```
plt1 = ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point() +
  geom_smooth(se=FALSE)

plt2 = ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))+
  geom_smooth(mapping = aes(x = displ, y = hwy, group=drv),se=FALSE)


plt3 = 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)


plt4 =ggplot(data = mpg,mapping = aes(x = displ, y = hwy)) +
  geom_point(mapping=aes(color=drv))+
  geom_smooth(se=FALSE)



plt5 = ggplot(data = mpg,mapping = aes(x = displ, y = hwy)) +
  geom_point(mapping = aes(color=drv))+
  geom_smooth(se=FALSE,mapping = aes(linetype = drv))


plt6 = ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point(mapping = aes(color=drv)) +
  geom_point(shape = 21, color = "white", stroke = 1)


plot_grid(plt1,plt2,plt3,plt4,plt5,plt6,ncol=2, nrow = 3)
```

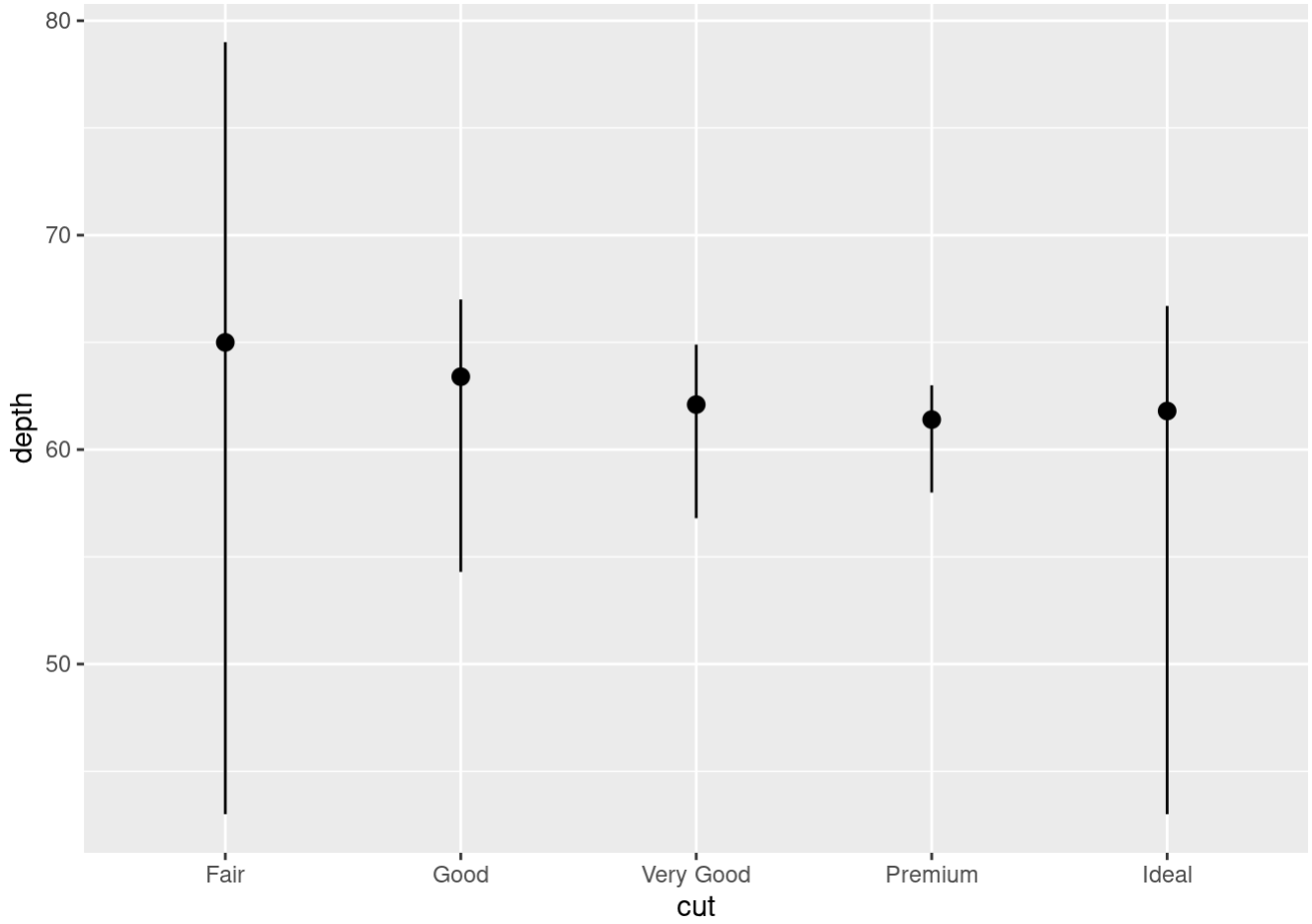
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula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



3.7.1

Q1. What is the default geom associated with stat_summary()? How could you rewrite the previous plot to use that geom function instead of the stat function?

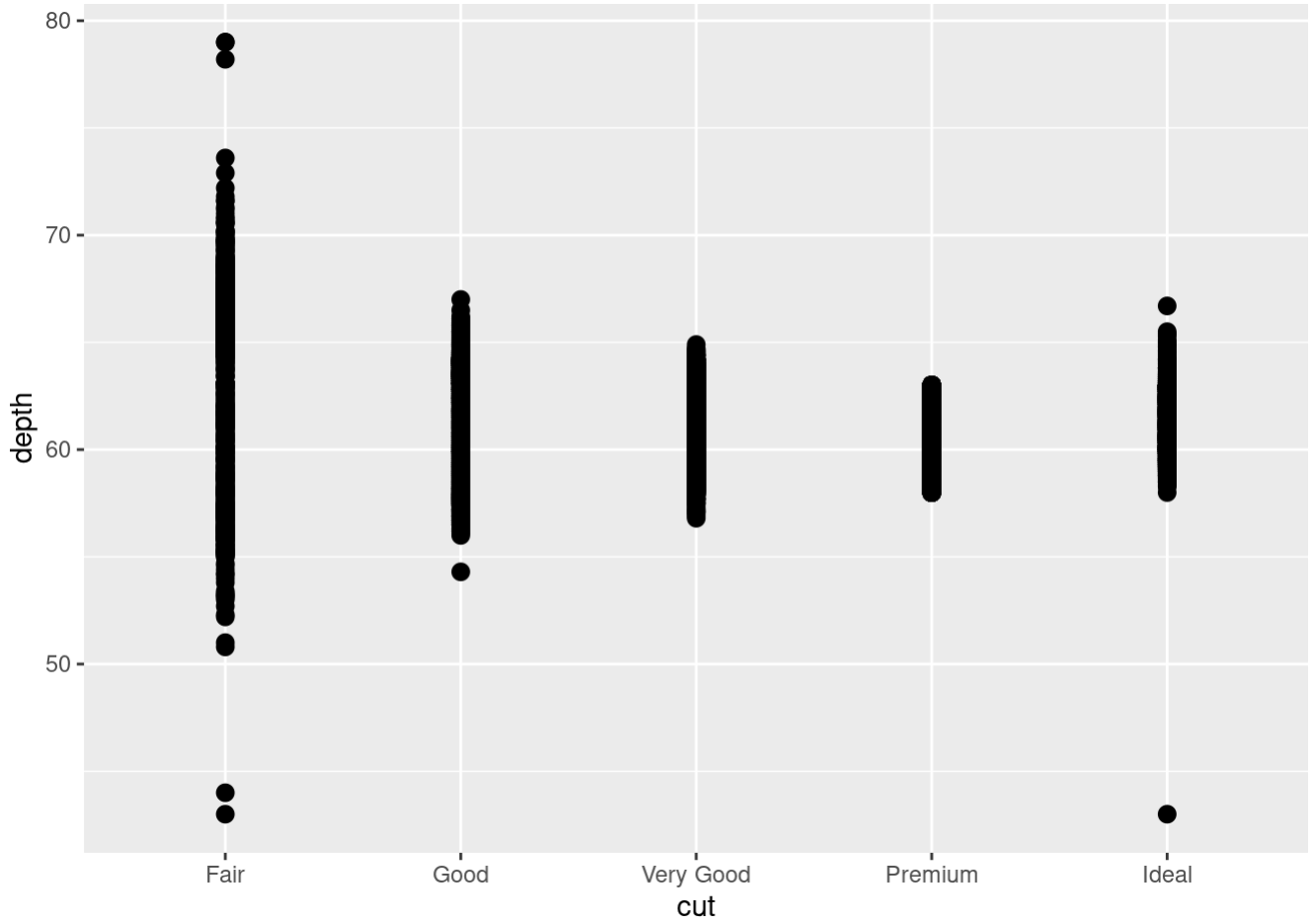
```
# That lot
ggplot(data = diamonds) +
  stat_summary(
    mapping = aes(x = cut, y = depth),
    fun.min = min,
    fun.max = max,
    fun = median
  )
```



`stat_summary` could work wihh different geoms default being `geom_pointrange` , as we can see in usage:

```
stat_summary_bin(
  mapping = NULL,
  data = NULL,
  geom = "pointrange",
  position = "identity",
  ...,
  fun.data = NULL,
  fun = NULL,
  fun.max = NULL,
  fun.min = NULL,
  fun.args = list(),
  bins = 30,
  binwidth = NULL,
  breaks = NULL,
  na.rm = FALSE,
  orientation = NA,
  show.legend = NA,
  inherit.aes = TRUE,
  fun.y,
  fun.ymin,
  fun.ymax
)
```

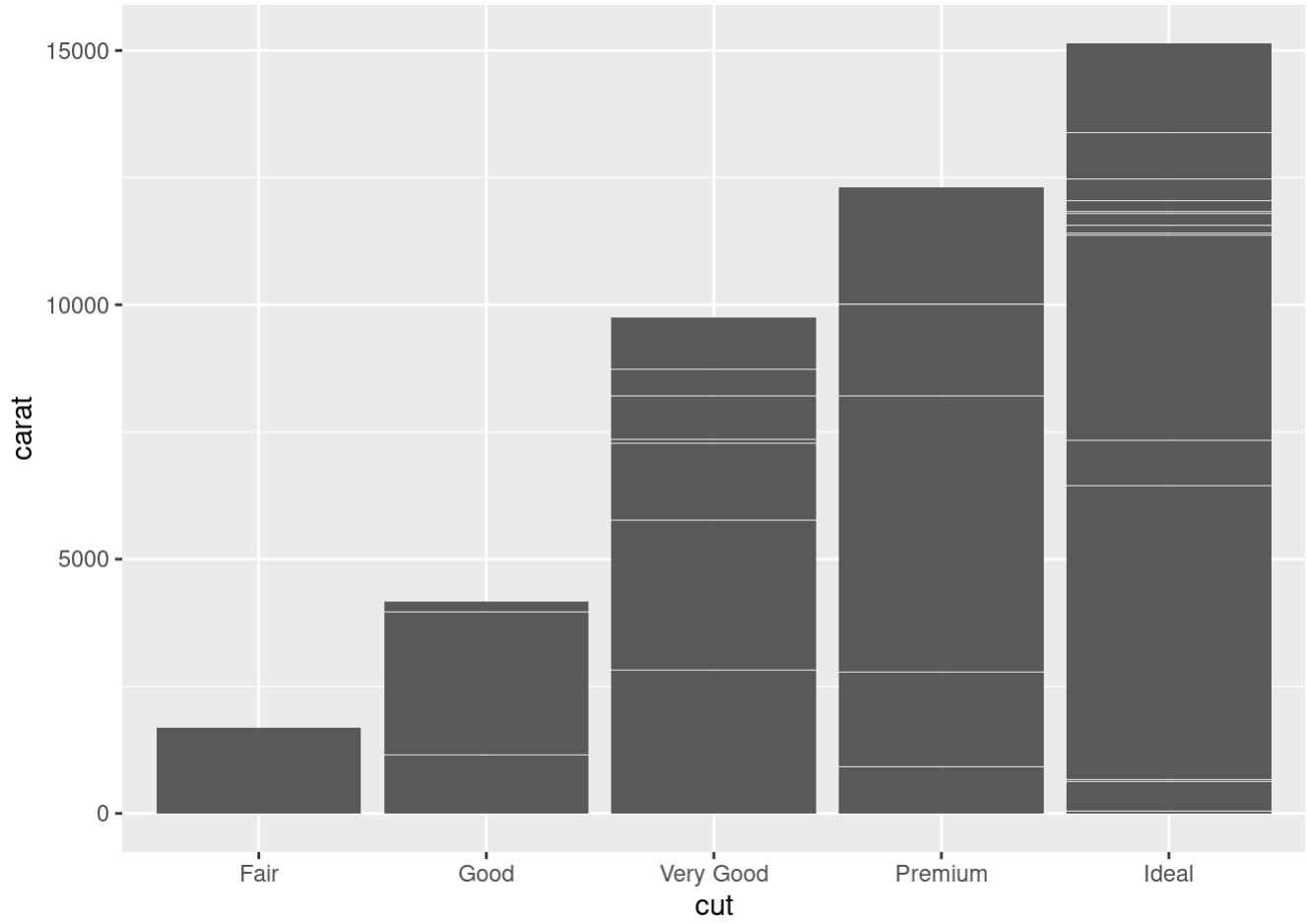
```
ggplot(diamonds) +
  geom_pointrange(aes(cut, depth, ymin = depth, ymax = depth))
```



Q2. What does geom_col() do? How is it different to geom_bar()?

```
// can have only x or y aesthetic
geom_bar
(
  ...
  stat="count"
  height_heuristic = "no of data points",
  orientation="NA"
  ...
)


// Must have only x and y aesthetic
geom_col
(
  ...
  stat="identity",
  height_heuristic = "range of data points"
  ...
)
```

```
ggplot(data = diamonds) +
  geom_col(mapping = aes(x = cut, y = carat))
```



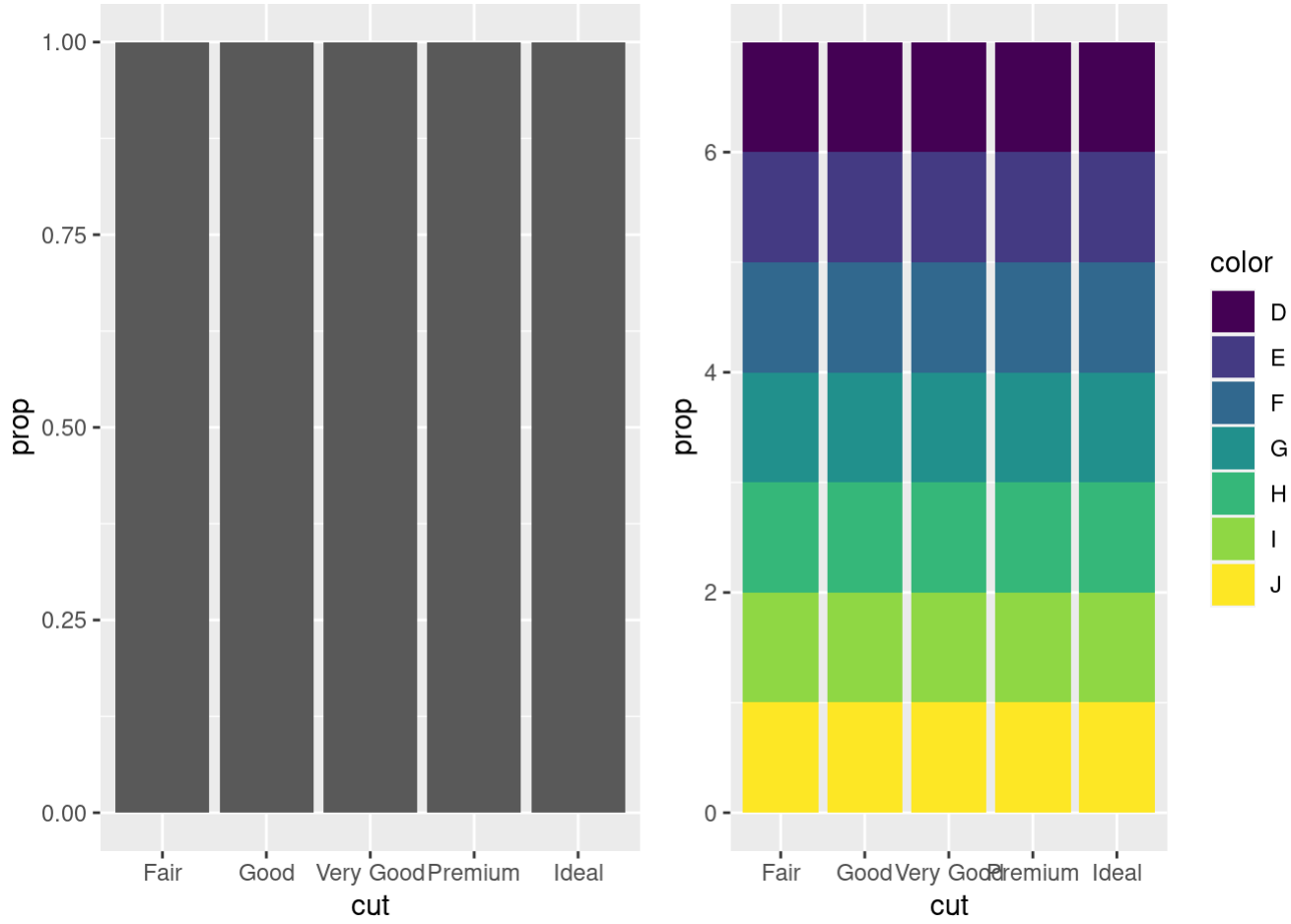
Q3. Most geoms and stats come in pairs that are almost always used in concert. Read through the documentation and make a list of all the pairs. What do they have in common?

Q4. What variables does stat_smooth() compute? What parameters control its behaviour?

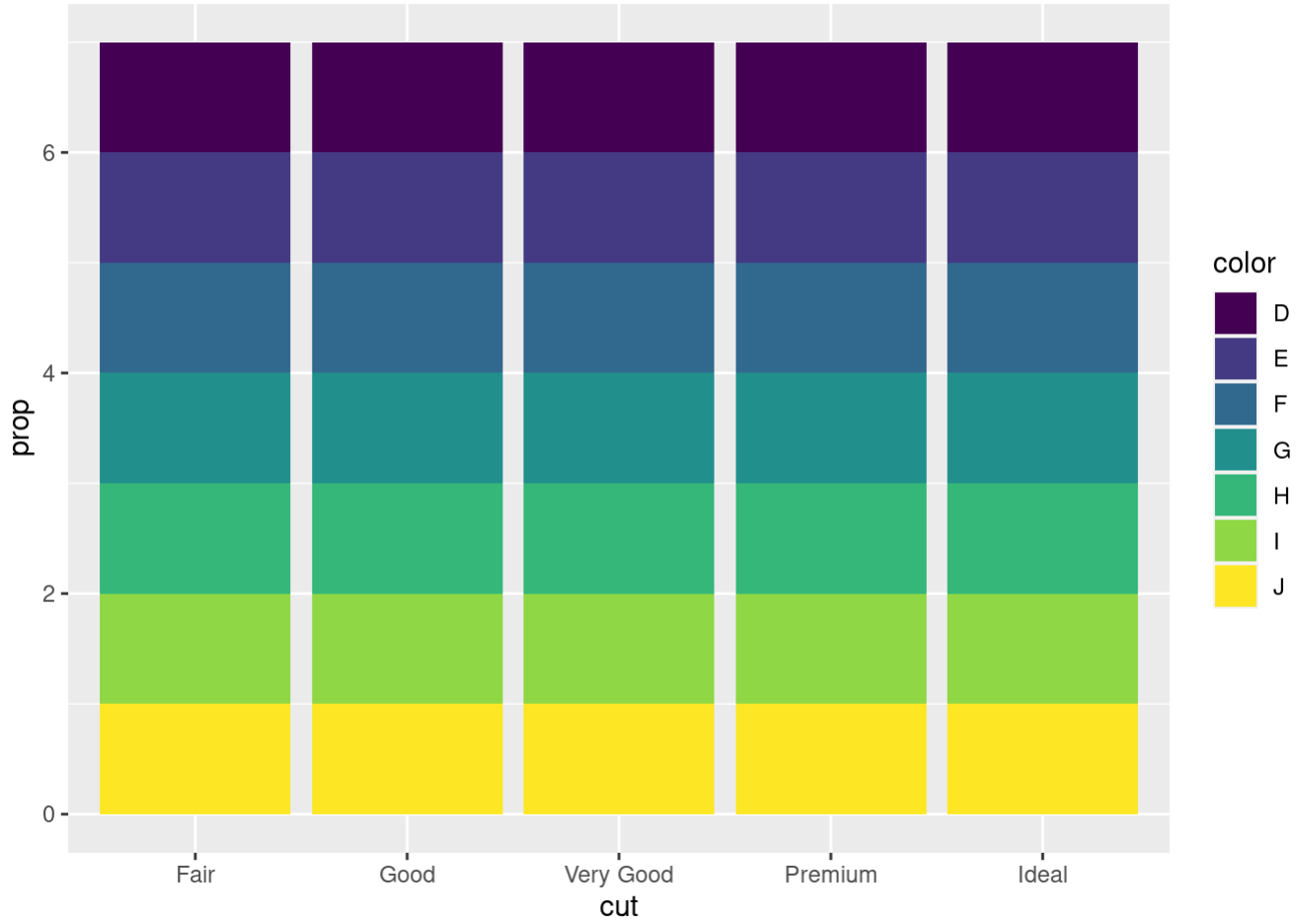
`stat_smooth()` predicts y based on any of the following heuristic 'lm', 'lowess', 'glm'

Q5. In our proportion bar chart, we need to set group = 1. Why? In other words what is the problem with these two graphs?

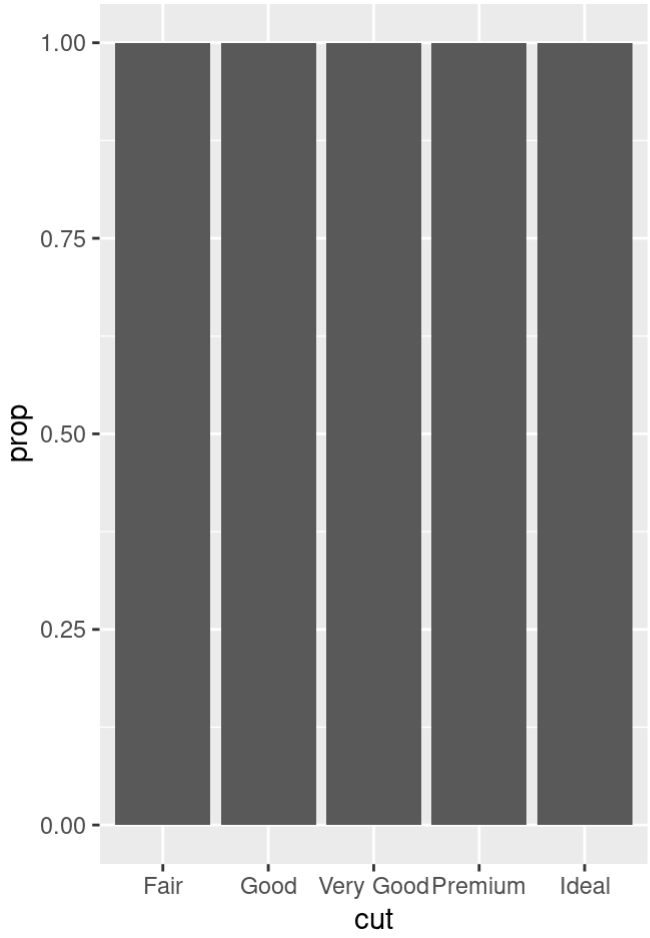
```
p1 = ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, y = after_stat(prop)))
p2 = ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, fill = color, y = after_stat(prop)))


plot_grid(p1,p2)
```

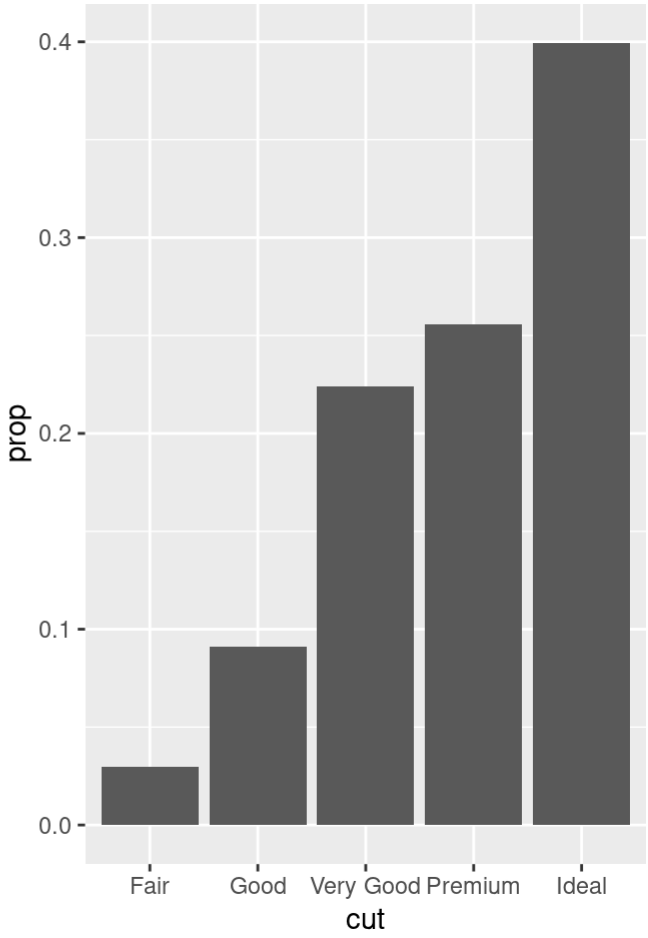



```
# Rather then plotting the proportion of each cut against its label along x-axis, plot t
he joint proportion, which in this case is 1
p1 = ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, y = after_stat(prop)))
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, fill = color, y = after_stat(prop)))
```




```
# Plotting the proportion of each cut against its label along x-axis, maximum val of y-axis is maximum proportion among all cuts.
p2 = ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, y = stat(prop), group = 1))


plot_grid(p1,p2)
```


1.00
0.75
0.50
0.25
0.00
prop
Fair
Good
Very Good
Premium
Ideal
cut


0.4
0.3
0.2
0.1
0.0
prop
Fair
Good
Very Good
Premium
Ideal
cut