Southwestern University of Finance and Economics

Multivariate Statistical Analysis

Homework1

Student:
Wenjie Lan

Supervisor:
Prof Xu

An Assignment submitted:

Data Visualization With ggplot2

March 11, 2022

1 Exercise

PROBLEM 1.

SOLUTION.
This code creates an empty plot which plays as an "background".

Figure 1: 1.1

PROBLEM 2.

SOLUTION.
There are 234 rows and 11 columns in the mpg data frame.

PROBLEM 3.

SOLUTION.
The drv variable categorizes cars into front-wheels, rear-wheels,and four-wheel.

PROBLEM 4.

SOLUTION.

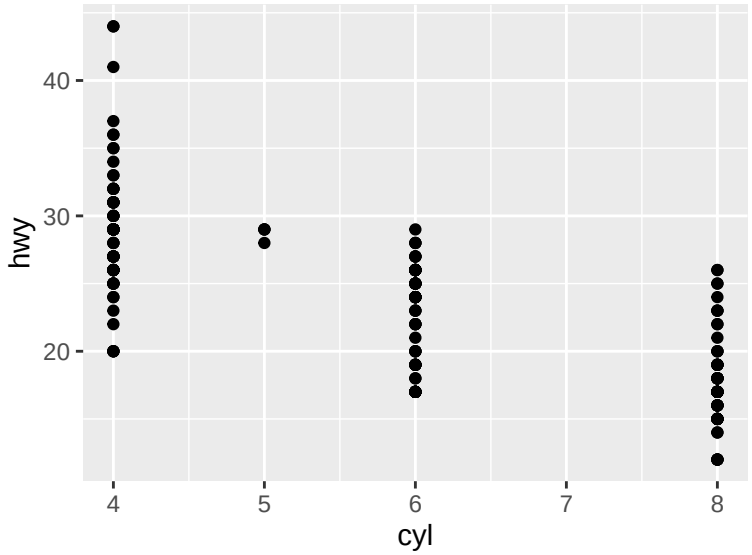


Figure 2: 1.4

PROBLEM 5.

SOLUTION.
Because drv and class are both categorical variables, and it only results in a few isolated points in the picture.

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

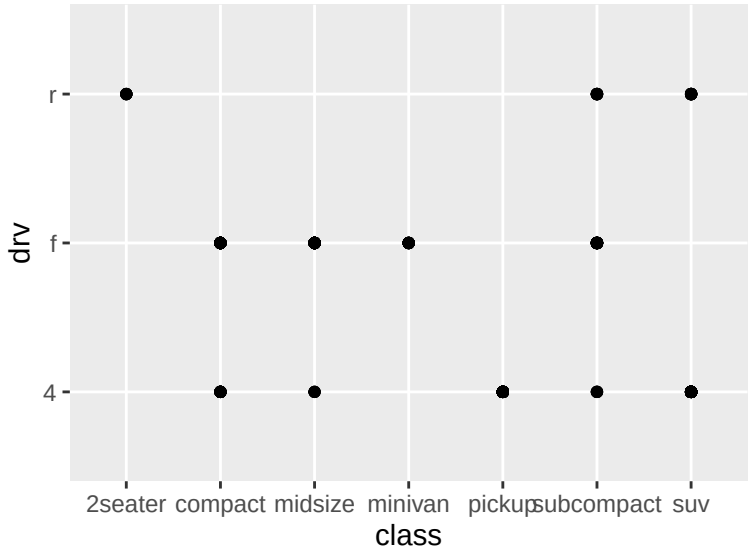


Figure 3: 1.5

2 Exercise

PROBLEM 6.

SOLUTION. The code "color = "blue"" should be outside of the aes argument.

Correct code:

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

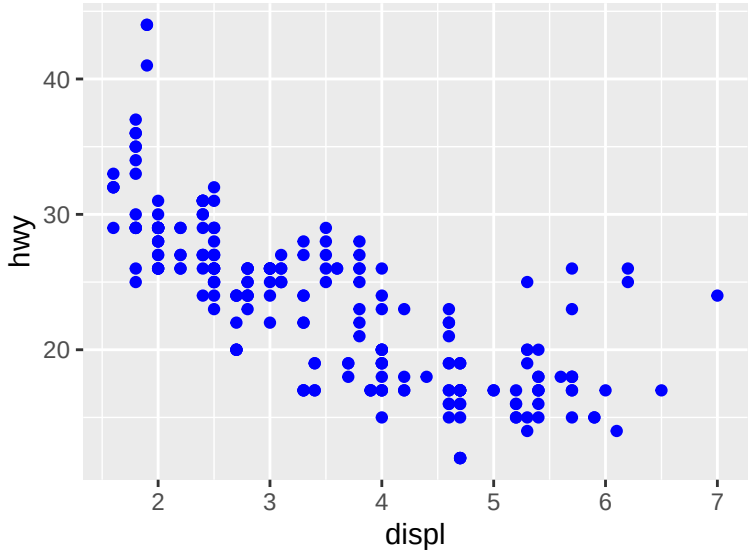


Figure 4: 2.1

PROBLEM 7.

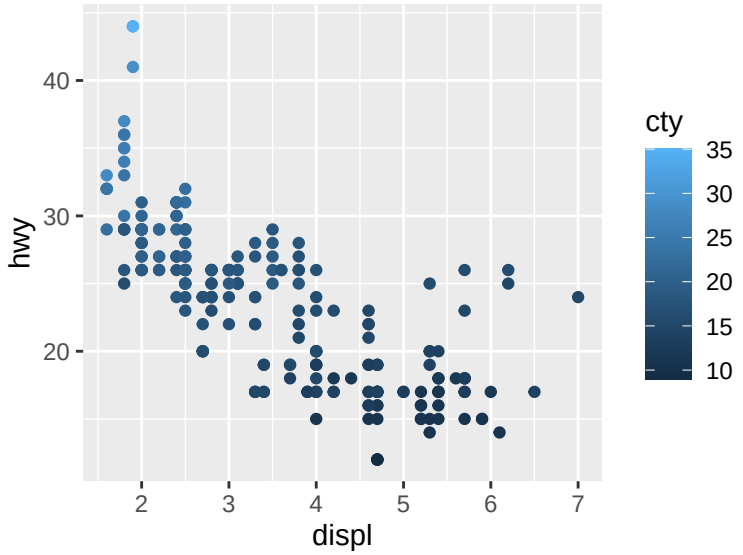
SOLUTION.
Categorical variables:
manufacturer
model
trans
drv
fl
class
Continuous variables:
displ
year
cyl
cty

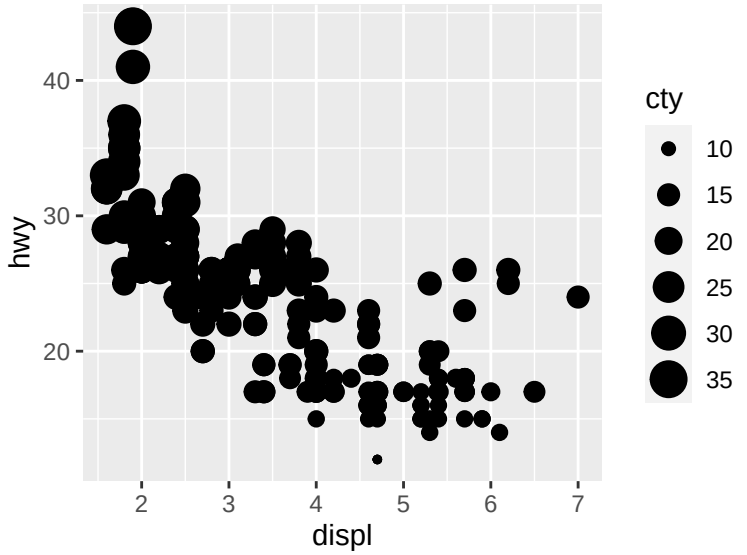
hwy

PROBLEM 8.

SOLUTION.
Color:appears as gradient color.
Size:appear as different size continuously.
shape:error since cty is a continuous variable.

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


hwy
40
30
20
2
3
4
5
6
7
displ
cty
35
30
25
20
15
10


hwy
40
30
20
2
3
4
5
6
7
displ
cty
10
15
20
25
30
35

```
## Error in `scale_f()`:## !  A continuous variable can not be
                           mapped to shape
```

PROBLEM 9.

SOLUTION.

```
ggplot(mpg, aes(x = displ, y = hwy, colour = hwy,
    size = displ)) +
  geom_point()
```

3.

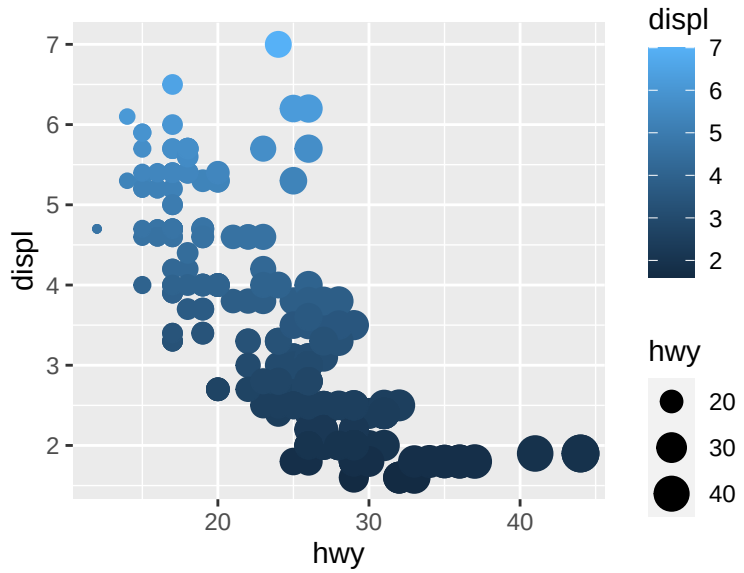


Figure 5: 2.4

PROBLEM 10.

SOLUTION.
It can change points' fill color and border.

3 Exercise

PROBLEM 11.

SOLUTION.
The continuous variable appears as categorical form.

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

3.

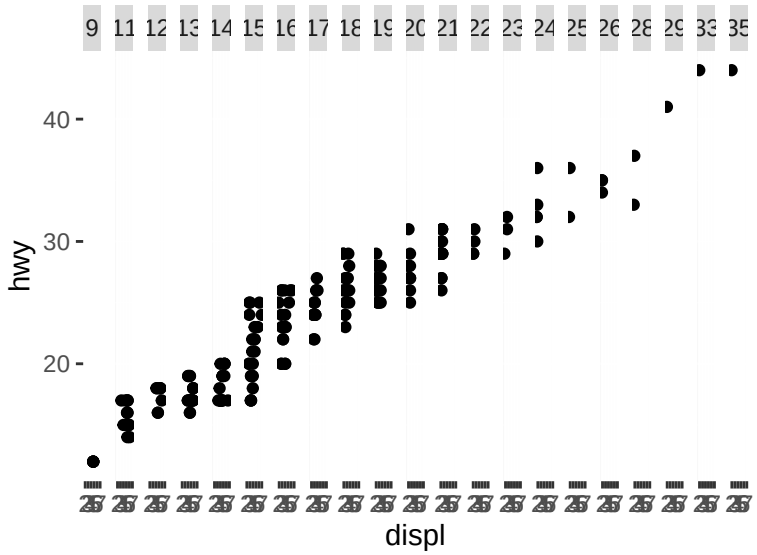


Figure 6: 3.1

PROBLEM 12.

SOLUTION.

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

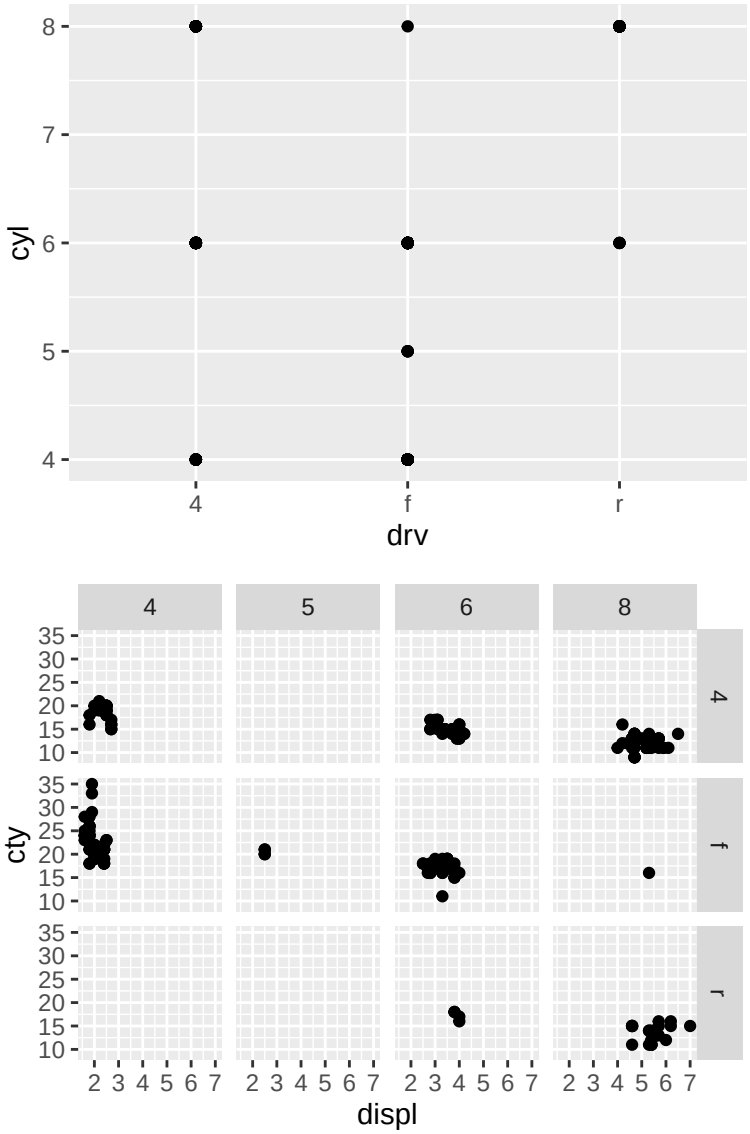


Figure 7: 3.2

PROBLEM 13.

SOLUTION. The "." ignores dimensions and facet by values on x or y axis according to where it is placed in the code.

```
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)
```

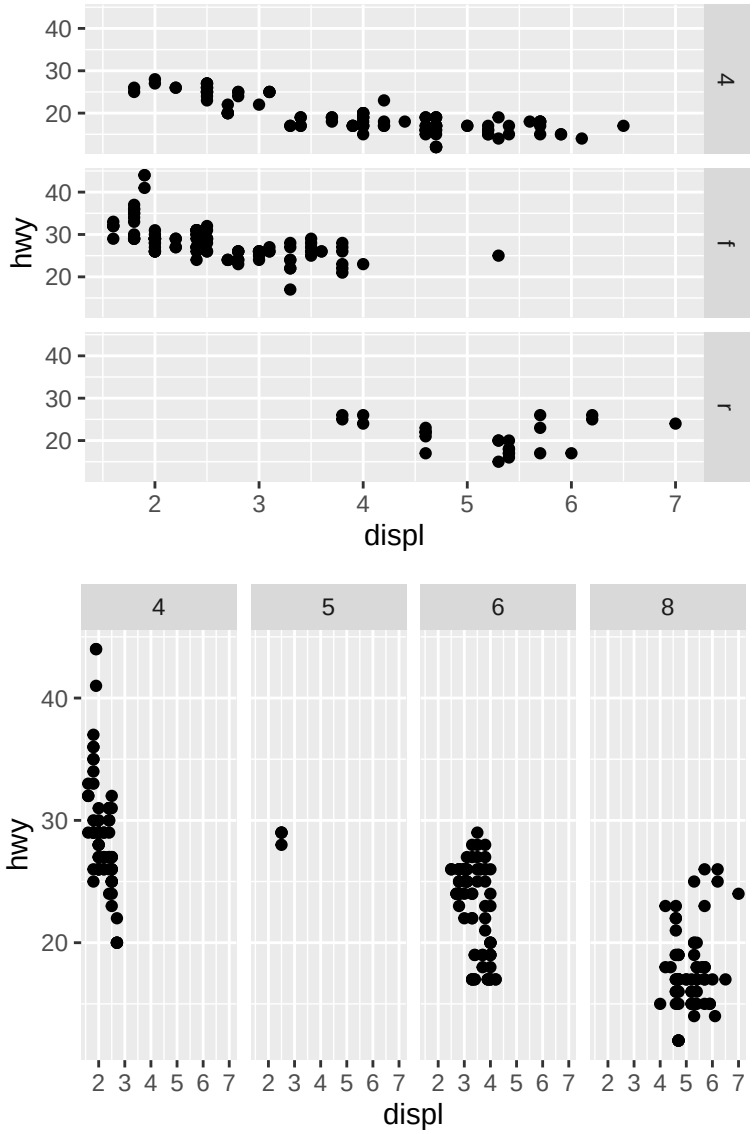


Figure 8: 3.3

PROBLEM 14.

SOLUTION.
Advantage:It is convenient to observe the condition within each catalog.
Disadvantage:It is inconvenient to compare different catalogs.
I would choose faceting instead of color when having a large dataset since colors can be difficult when distinguishing.

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

```
    color = class))
```

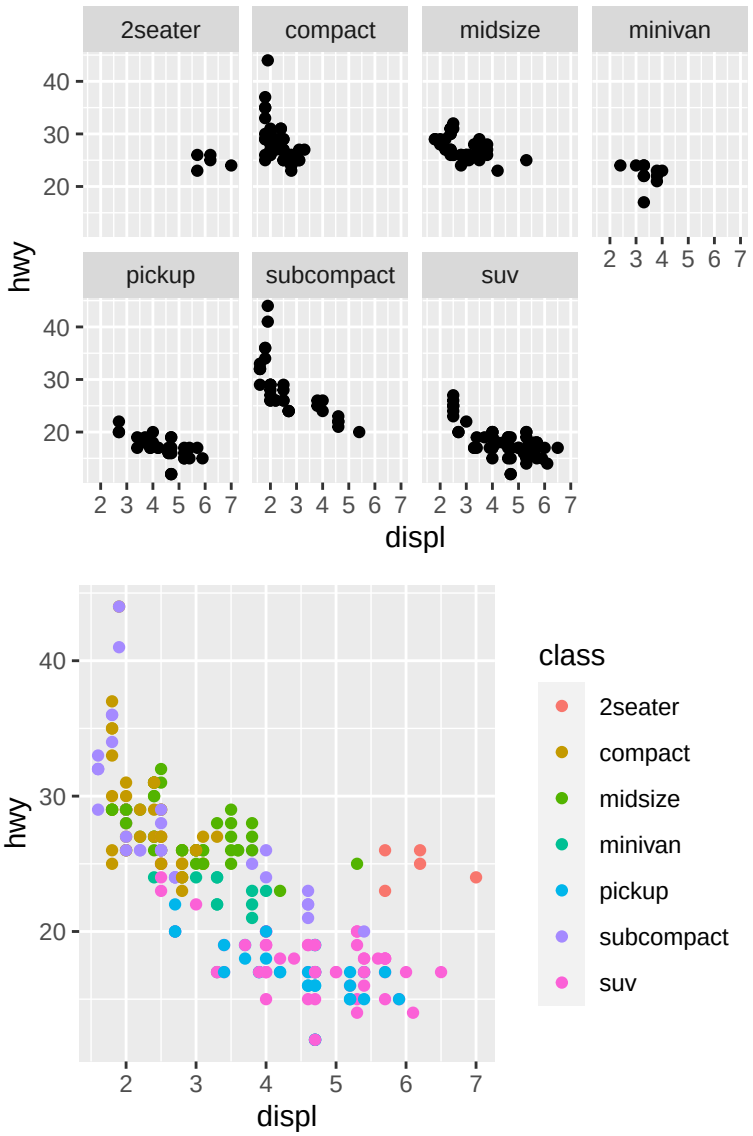


Figure 9: 3.4

PROBLEM 15.

SOLUTION.
"nrow (ncol)" determines the number of rows (columns) of facets we create.
Because the number of rows and columns represent the number of variables.

PROBLEM 16.

SOLUTION.
When we use facet_wrap we usaully put the variable with more unique levels in the

columns to create less columns, since it is not easy to display too many columns and inconvenient for we to observe data.

4 Exercise

PROBLEM 17.

SOLUTION.
Line chart: geom_line()
Boxplot: geom_boxplot()
Histogram: geom_histogram()
Area chart: geom_area()

PROBLEM 18.

SOLUTION.

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

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

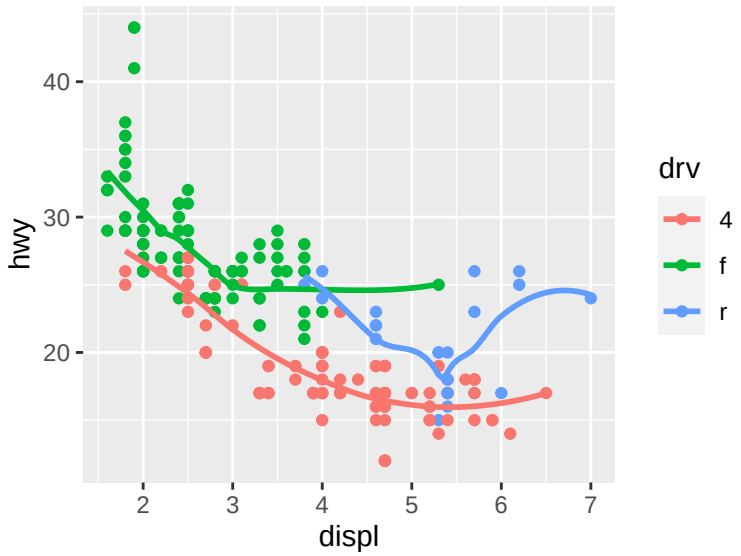


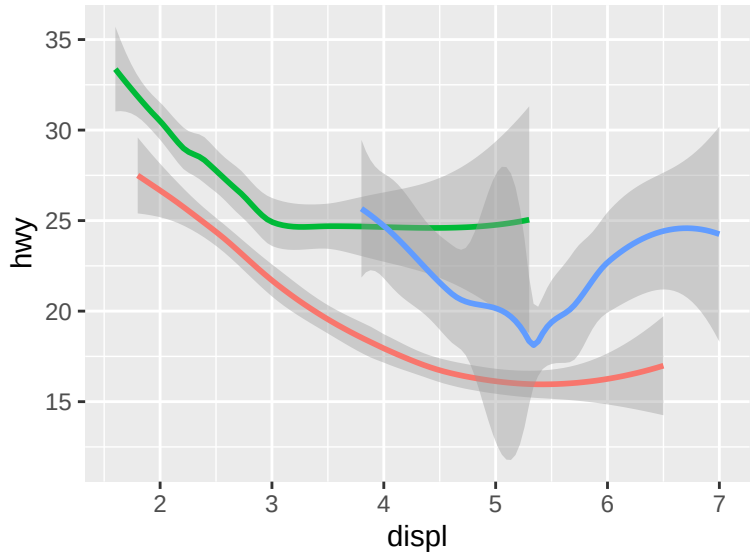
Figure 10: 4.2

PROBLEM 19.

SOLUTION.
"show.legend = FALSE" help to remove the legend.
The author use it

```
ggplot(data = mpg) +
  geom_smooth(
    mapping = aes(x = displ, y = hwy, colour = drv)
        ,
    show.legend = FALSE
  )
ggplot(data = mpg) +
  geom_smooth(
    mapping = aes(x = displ, y = hwy, colour = drv)
  )
```

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




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

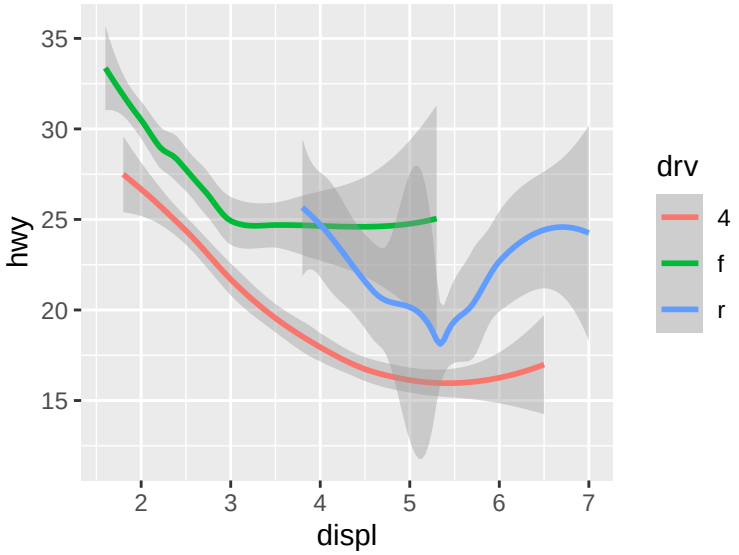


Figure 11: 4.3

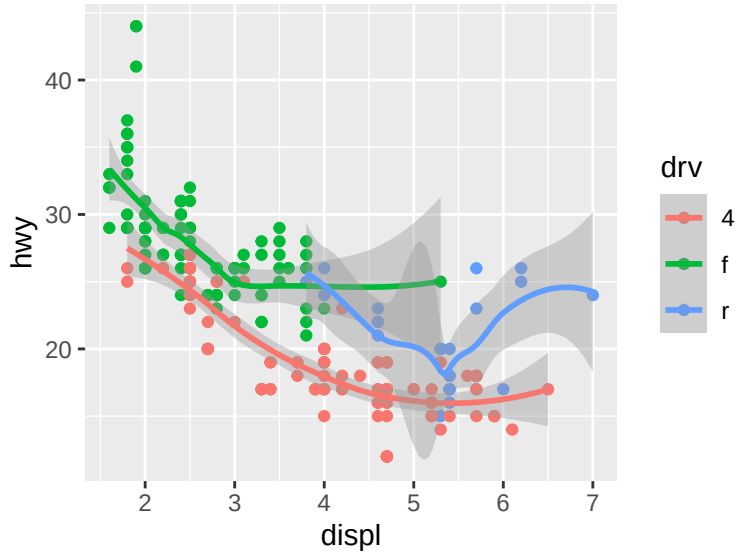
PROBLEM 20.

SOLUTION.
It shows standard error bands by adding shadow to the lines.

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

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

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




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

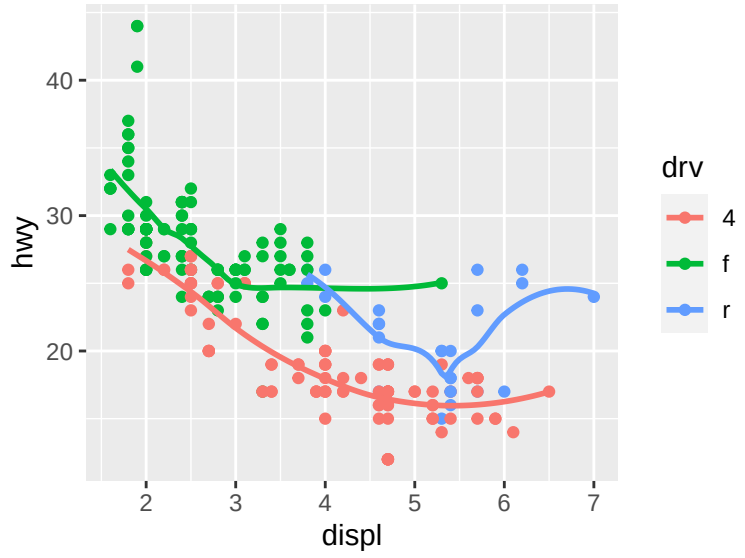


Figure 12: 4.4

PROBLEM 21.

SOLUTION.
The two graphs look the same.Since they display the same data by same method.

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

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

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

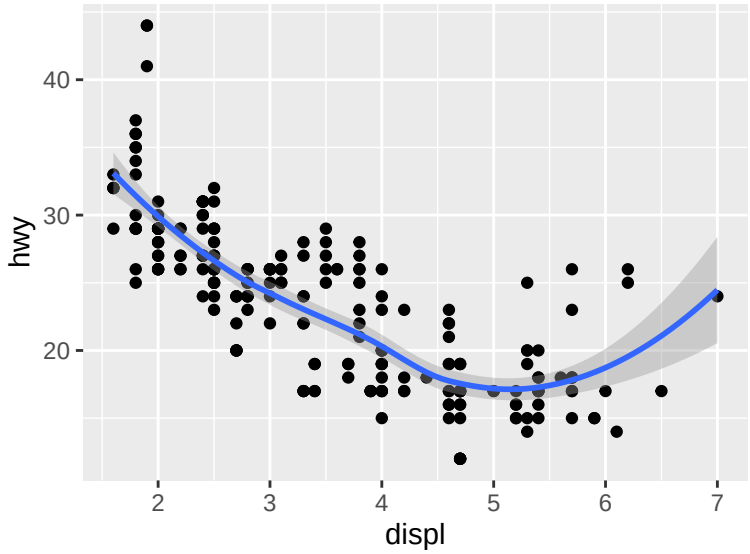


Figure 13: 4.5

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

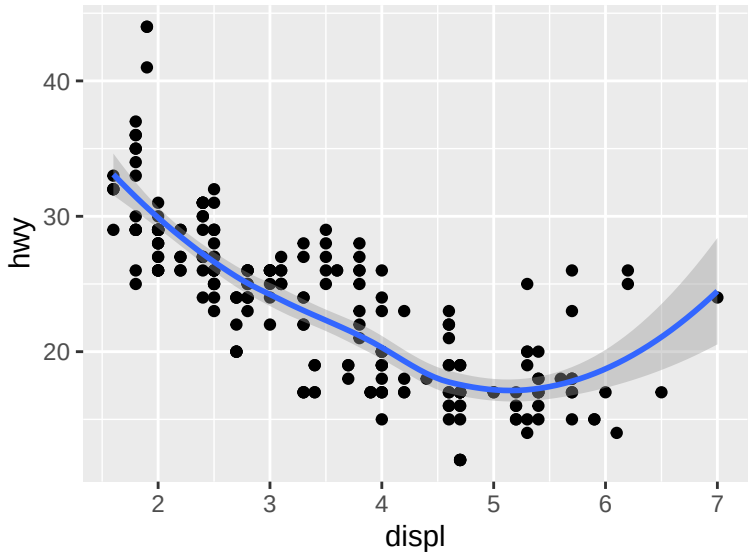


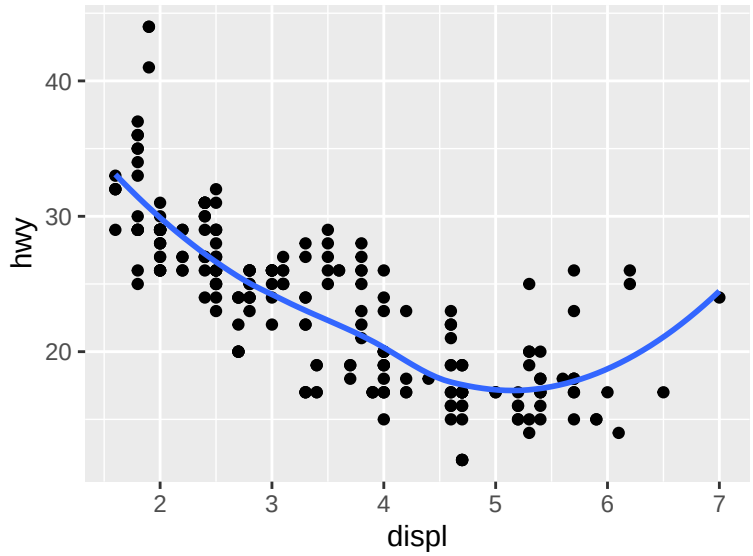
Figure 14: 4.5

PROBLEM 22.

SOLUTION.

```r
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point()+
  geom_smooth(se = FALSE)
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point()+
  geom_smooth(mapping = aes(group = drv),se = FALSE
      )
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(mapping = aes(color = drv))+
  geom_smooth(mapping = aes(color = drv),se = FALSE
      )
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(mapping = aes(color = drv))+
  geom_smooth(se = FALSE)
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(mapping = aes(color = drv))+
  geom_smooth(mapping = aes(linetype = drv),se =
     FALSE)
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(size = 3,color = "white") +
  geom_point(aes(colour = drv))
```

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




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

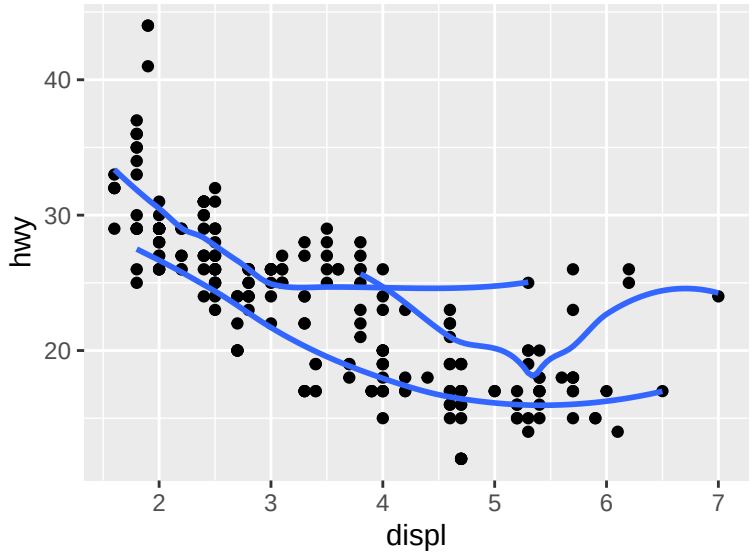
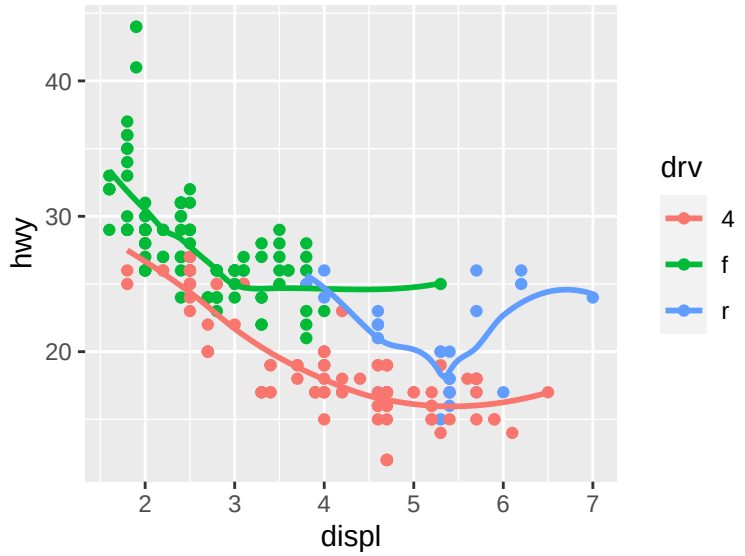


Figure 15: 4.6

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




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

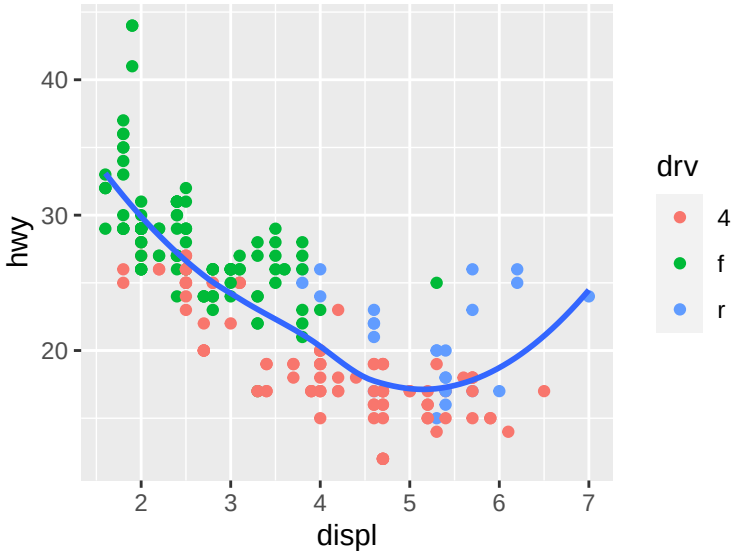


Figure 16: 4.6

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

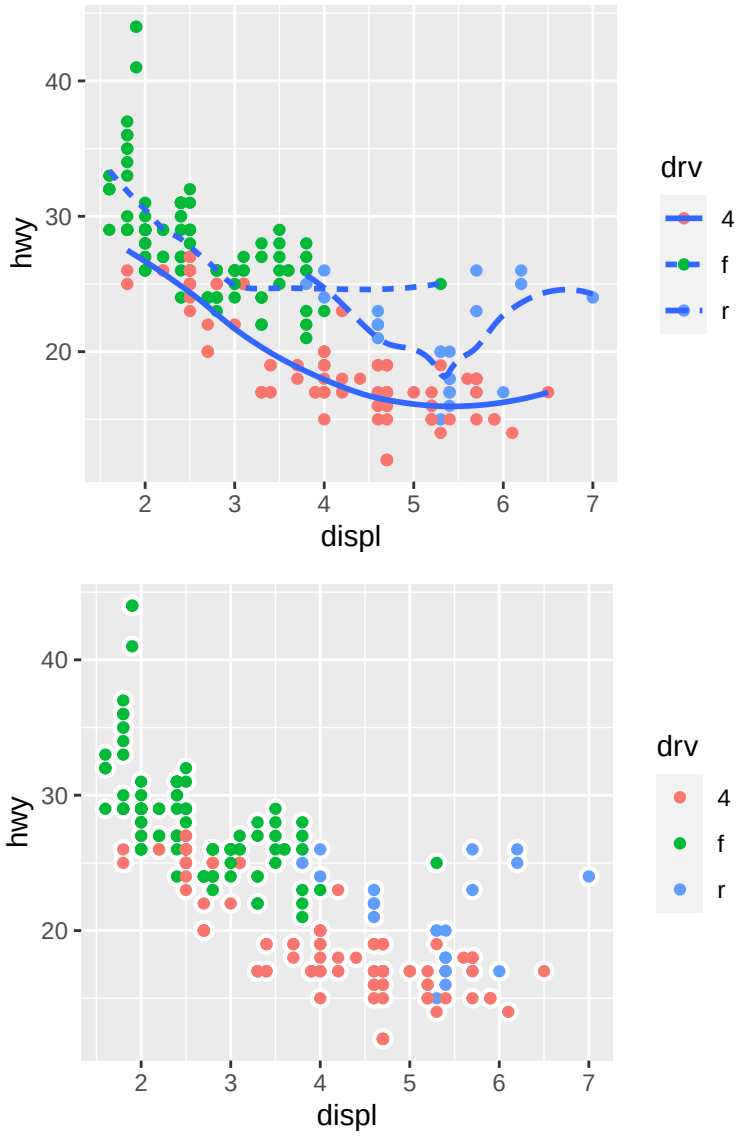


Figure 17: 4.6

5 Exercise

PROBLEM 23.

SOLUTION.

```
ggplot(data = diamonds) +
  stat_summary(
    mapping = aes(x = cut, y = depth),
```

```
    fun.min = min,
    fun.max = max,
    fun = median
  )
ggplot(data = diamonds) +
  geom_pointrange(
    mapping = aes(x = cut, y = depth),
    stat = "summary",
    fun.min = min,
    fun.max = max,
    fun = median
  )
```

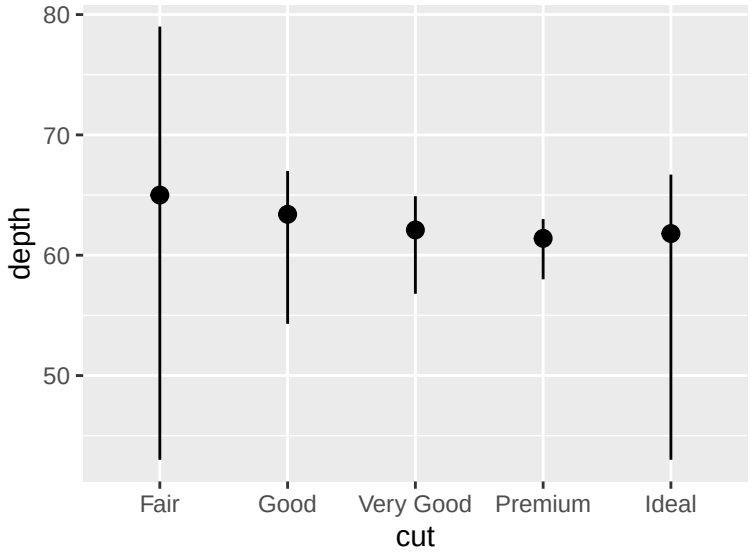


Figure 18: 5.1

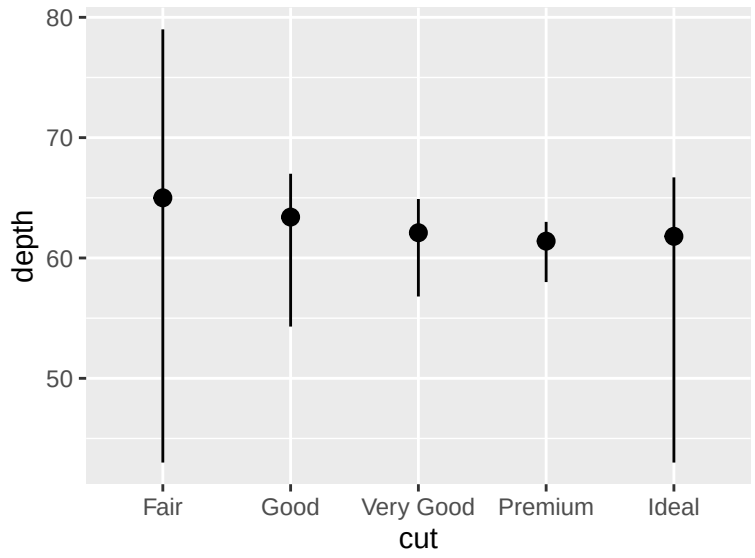


Figure 19: 5.1

PROBLEM 24.

SOLUTION.
Geom_col and geom_bar are used to create different types of bar Charts.
Official explanation are as follows:
Geom_bar() makes the height of the bar proportional to the number of cases in each group (or if the weight aesthetic is supplied, the sum of the weights). If you want the heights of the bars to represent values in the data, use geom_col() instead. geom_bar() uses stat_count() by default: it counts the number of cases at each x position. geom_-col() uses stat_identity(): it leaves the data as is.

```
geom_bar(
  mapping = NULL,
  data = NULL,
  stat = "count",
  position = "stack",
  ...,
  width = NULL,
  na.rm = FALSE,
  orientation = NA,
  show.legend = NA,
  inherit.aes = TRUE
)

geom_col(
  mapping = NULL,
  data = NULL,
  position = "stack",
  ...,
```

```
    width = NULL,
    na.rm = FALSE,
    show.legend = NA,
    inherit.aes = TRUE
)
```

PROBLEM 25.

SOLUTION.
Some geom have stat as the default element while some stat also have geom as the default element.

geom	stat
geom_bar()	stat_count()
geom_bin2d()	stat_bin_2d()
geom_boxplot()	stat_boxplot()
geom_contour_filled()	stat_contour_filled()
geom_contour()	stat_contour()
geom_count()	stat_sum()
geom_density()	stat_density()
geom_dotplot()	stat_bindot()
geom_function()	stat_function()
geom_sf()	stat_sf()
geom_smooth()	stat_smooth()
geom_violin()	stat_ydenstity()
geom_hex()	stat_bin_hex()
geom_qq()	stat_qq()
geom_qq_line()	stat_qq_line()
geom_quantitle()	stat_quantile()

list pairs of geom and stat

PROBLEM 26.

SOLUTION.
"stat_smooth" is used to fit a continuous curve to discrete data points.
It calculates the following variables:
Predicted value(y) and its confidence interval
Example is as follow:

```
stat_smooth(
    mapping = NULL,
    data = NULL,
    geom = "smooth",
    position = "identity",
    ...,
```

```
    method = NULL,
    formula = NULL,
    se = TRUE,
    n = 80,
    span = 0.75,
    fullrange = FALSE,
    level = 0.95,
    method.args = list(),
    na.rm = FALSE,
    orientation = NA,
    show.legend = NA,
    inherit.aes = TRUE
)
```

```
ggplot(mtcars, aes(qsec, wt))+
  stat_smooth()+
  stat_smooth() + geom_point()
```

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

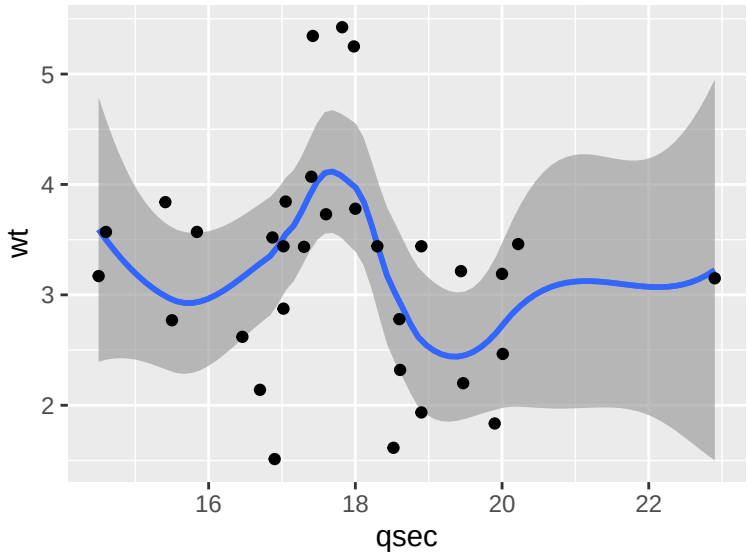


Figure 20: 5.4

PROBLEM 27.

SOLUTION.
The problem with these two graphs is that the proportions are calculated within every each type which results in 1 for every x.

```r
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, y = ..prop..))

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

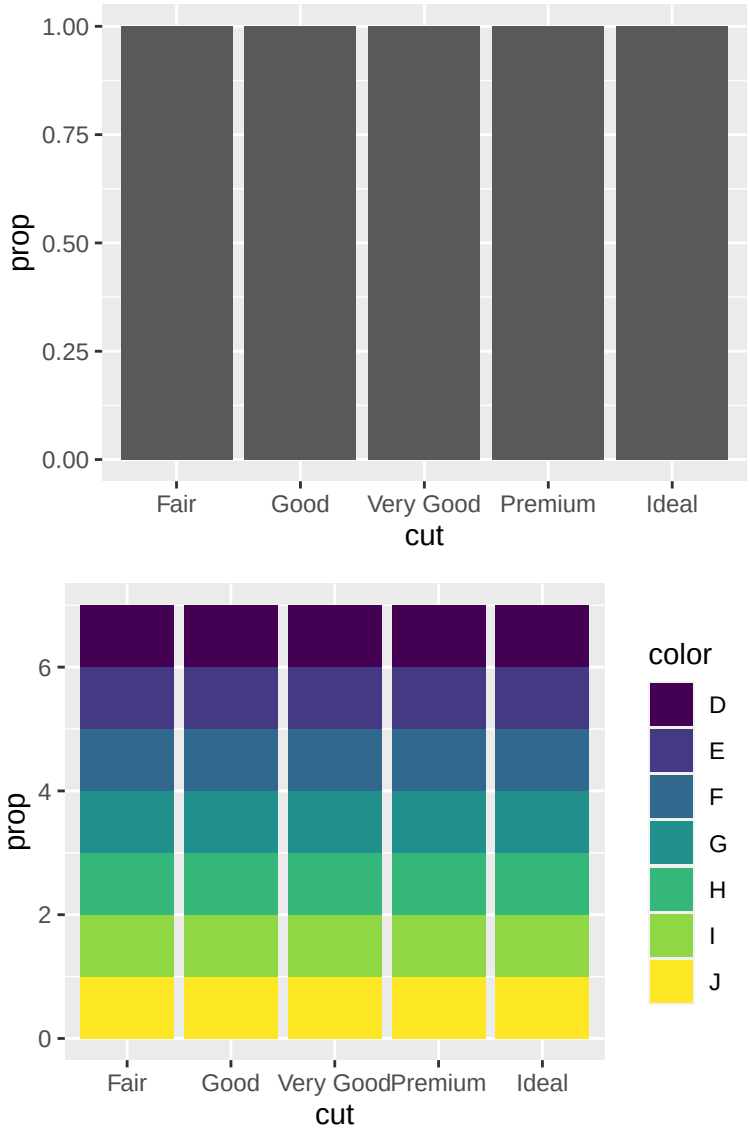


Figure 21: 5.5

6 Exercise

PROBLEM 28.

SOLUTION.
The problem with the plot is that many points overlap together.Thus, we have difficulty in seeing where the mass of data is.To solve the problem, we can add some random noise to the points.

```
%original codes
ggplot(data=mpg,mapping = aes(x=cty,y=hwy))+
  geom_point()
%improved codes
ggplot(data=mpg,mapping = aes(x=cty,y=hwy))+
  geom_jitter()
```

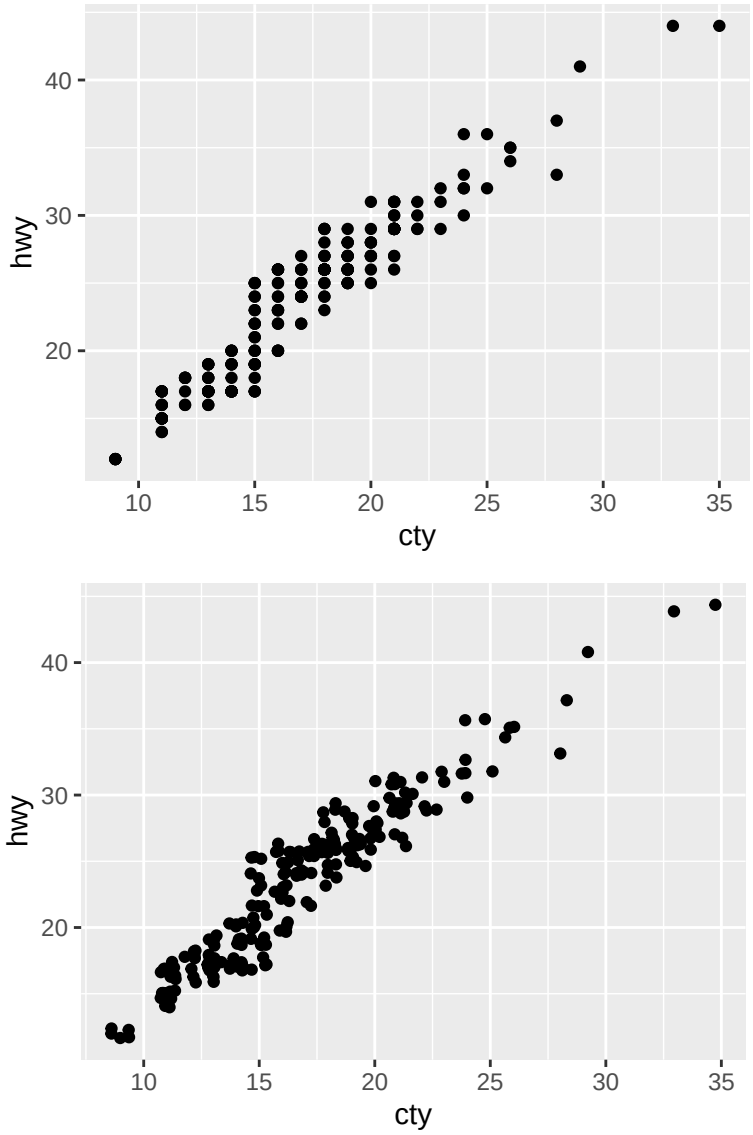


Figure 22: 6.1

PROBLEM 29.

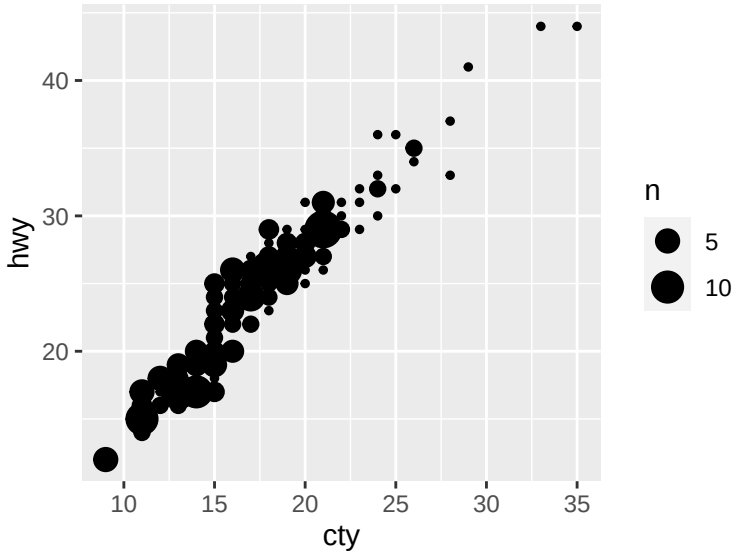
SOLUTION.
The argument "width" and "height" are used to control the amount of jittering.
Specifically, The jitter is added in both positive and negative directions, so the total spread is twice the value specified here.defaults to 0.4.
And for categorical data which is aligned on the integers, it not possible to see the distinction.So a width or height of 0.5 will spread the categorical data.

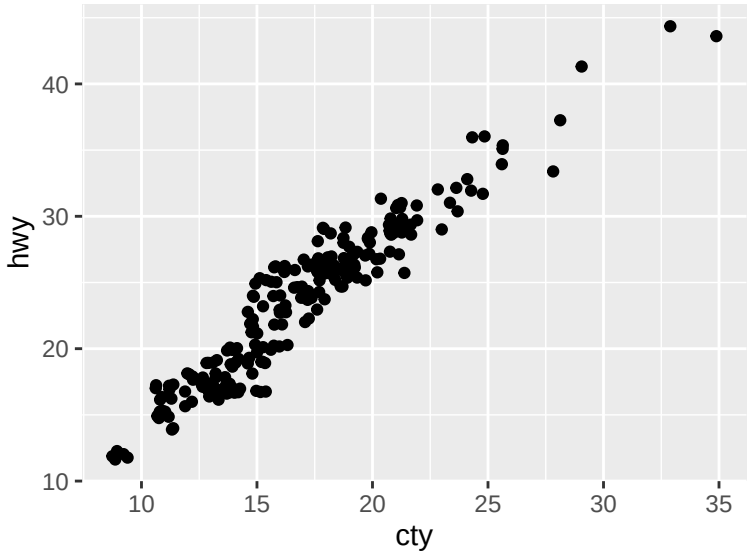
```
geom_jitter(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "jitter",
  ...,
  width = NULL,
  height = NULL,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)
```

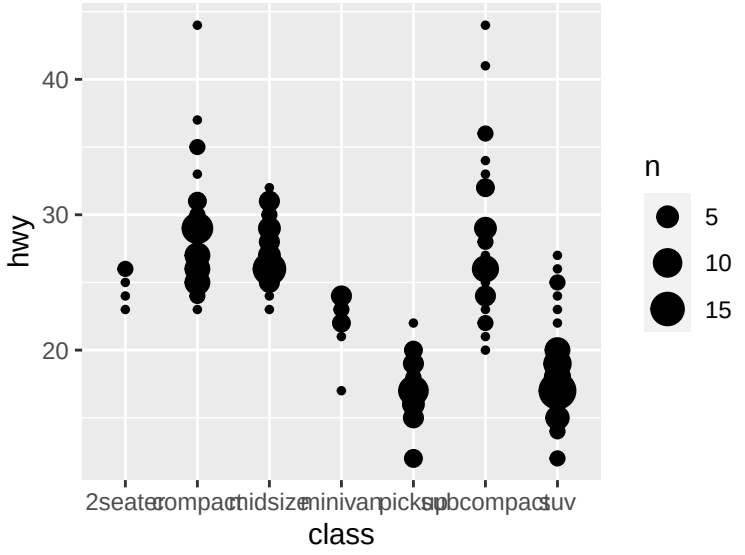
PROBLEM 30.

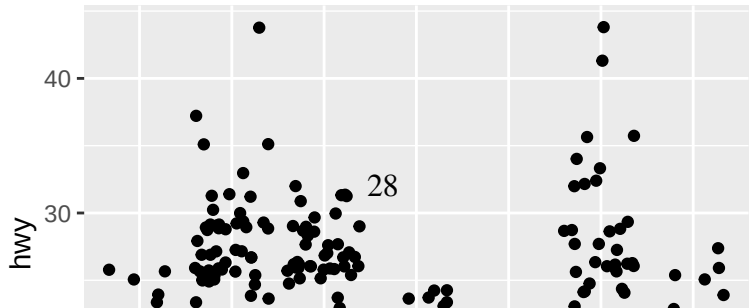
SOLUTION.
Both of geom_count and geom_jitter are used to solve overlapping. But, geom_count is used for categorical variables while geom_jitter is suitable for continuous variables.

```
ggplot(data=mpg,mapping = aes(x=cty,y=hwy))+
  geom_count()
ggplot(data=mpg,mapping = aes(x=cty,y=hwy))+
  geom_jitter()
ggplot(data=mpg,mapping = aes(x=class,y=hwy))+
  geom_count()
ggplot(data=mpg,mapping = aes(x=class,y=hwy))+
  geom_jitter()
```


hwy
cty
n
5
10
40
30
20
10
15
20
25
30
35


hwy
cty
40
30
20
10
15
20
25
30
35


hwy
class
n
5
10
15
40
30
20
2seater
compact
midsize
minivan
pickup
subcompact
suv


hwy
40
30

PROBLEM 31.

SOLUTION.
The boxplot compactly displays the distribution of a continuous variable. It visualises five summary statistics (the median, two hinges and two whiskers), and all "outlying" points individually.

```
ggplot(data=mpg, mapping = aes(x=class, y=hwy))+
  geom_boxplot()
```

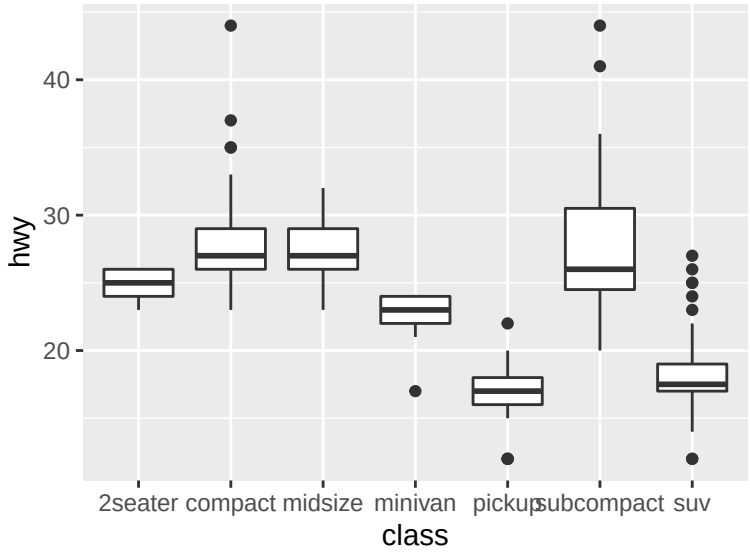


Figure 24: 6.4

7 Exercise

PROBLEM 32.
Turn a staked bar chart to a pie chart.

SOLUTION.

```
bar = ggplot(data = diamonds) +
 geom_bar(mapping = aes(x = cut, fill = cut),show.
    legend = FALSE, width = 1)
bar
bar + coord_polar()
```

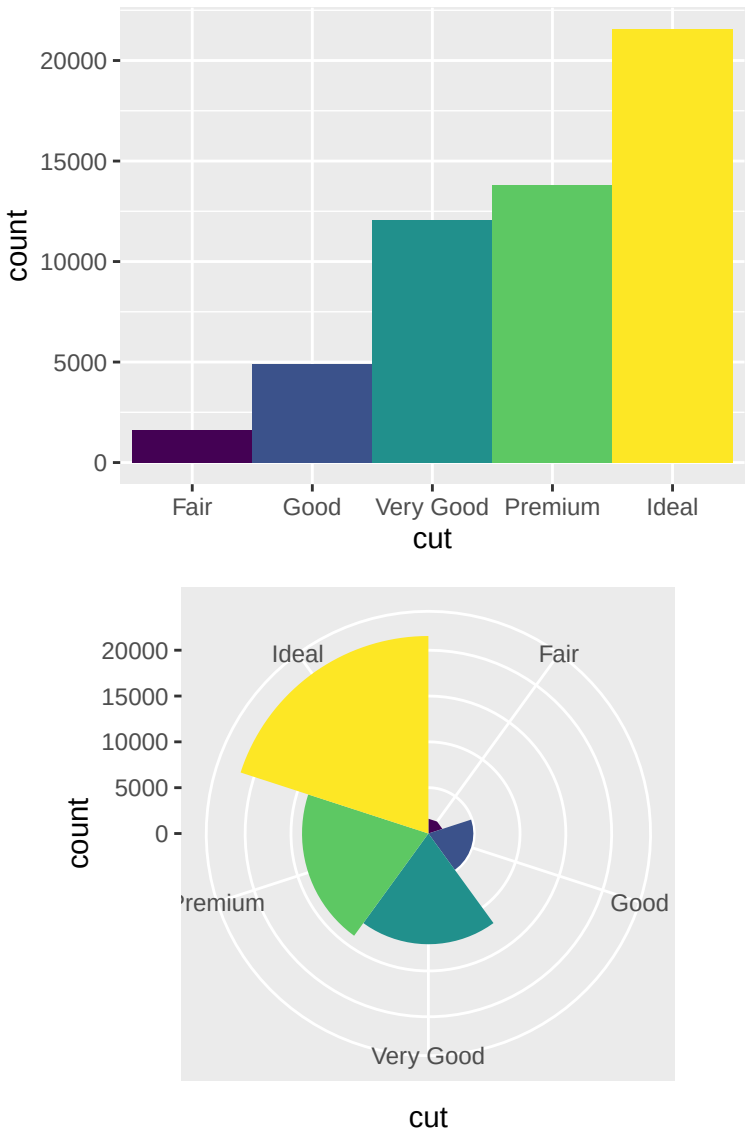


Figure 25: 7.1

PROBLEM 33.

SOLUTION.
"labs" modify axis, legend, and plot labels.

```
labs(
  ...,
  title = waiver(),
  subtitle = waiver(),
  caption = waiver(),
  tag = waiver(),
  %tag can be used for adding identification tags
     to differentiate between multiple plots.
```

```
  alt = waiver(),
  alt_insight = waiver()
)

xlab(label)

ylab(label)

ggtitle(label, subtitle = waiver())
```

PROBLEM 34.

SOLUTION.
They are projected differently. "Quick_mapping" ignores the curvature of the earth, so it is faster when calculating.

```
library(maps)
nz = map_data("nz")
map = ggplot(nz,aes(long,lat,group=group))+
  geom_polygon(fill="white",color="black")
map + coord_map()
map + coord_quickmap()
```

```
#### 'maps' ## The following object is masked from
                'package:purrr':####     map
```

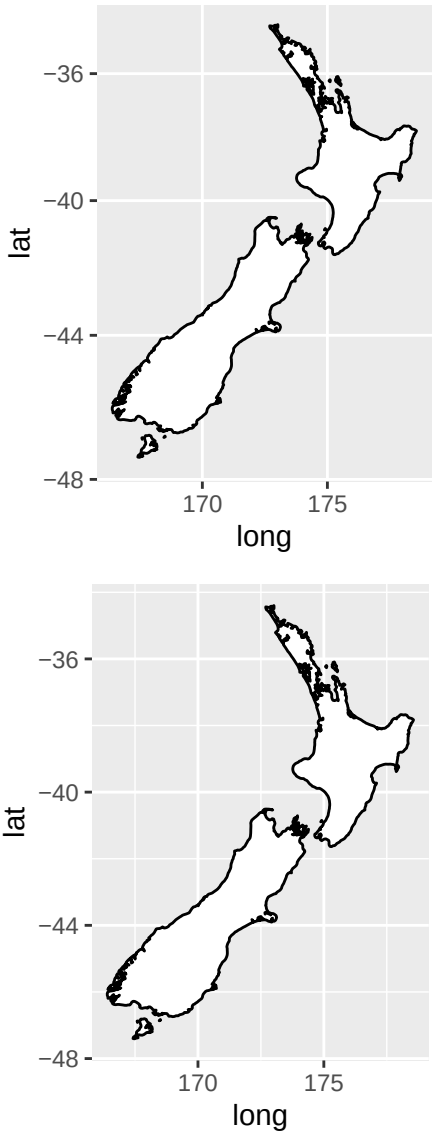


Figure 26: 7.3

PROBLEM 35.

SOLUTION.
"geom_abline" add line y=x to the plot, helping us to explore their size and relationship.
"coord_fixed" helps to remain the fixed ratio of the graph.

```
a <- ggplot(data = mpg, mapping = aes(x = cty, y =
    hwy))+
  geom_point()+
```

```
  geom_abline()
a
a + coord_fixed()
```

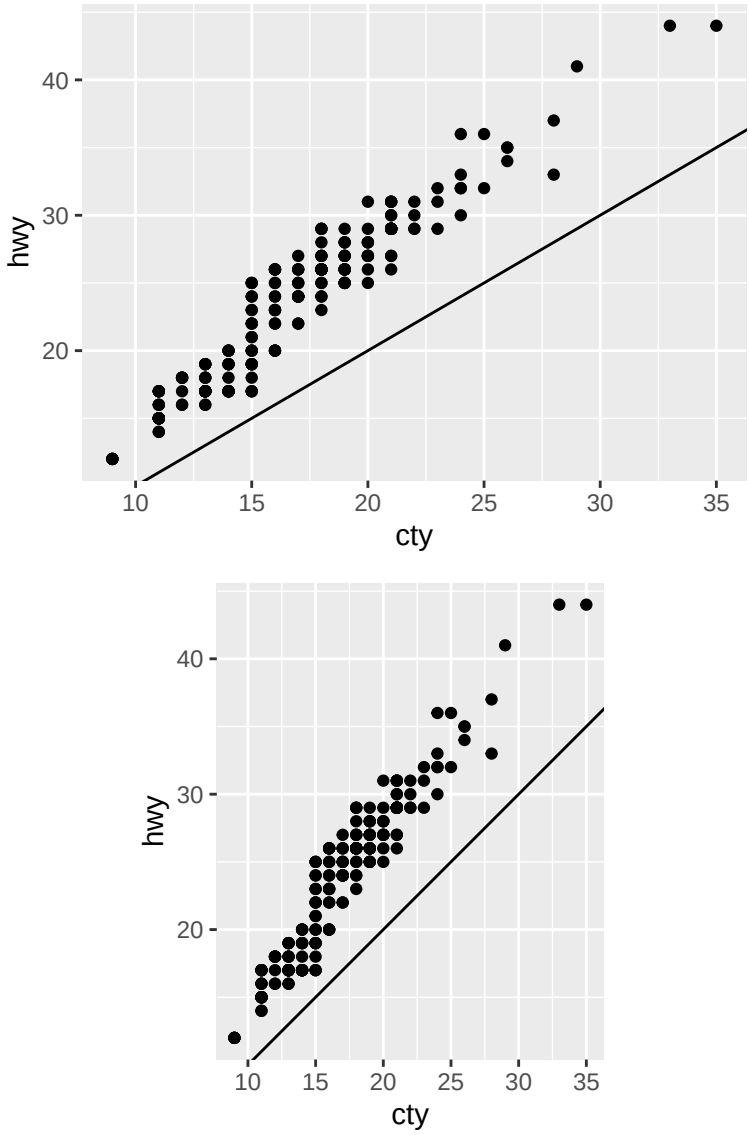


Figure 27: 7.4