# STAT167 HW2 - Spring 2025

Ethan Choi

## Contents

### Homework #2 instructions

**Review textbook Chapter 9 "Visualize > Layers" and the lecture notes on `ggplot2` before answering the homework questions**.

This homework contains 2 questions, each with multiple parts, 100 points in total.

Replace **INSERT_YOUR_ANSWER** with your own answers.

- First open this `rmd` file in RStudio and click `Knit -> Knit to PDF` to render it to PDF format. You need to have `LaTex` installed on the computer to render it to PDF format. If not, you can also render it to HTML format.
- It is best to read this `rmd` file and the rendered `pdf/html` file side-by-side, while you are working on this homework.
- If the question asks you to write some R code, remember to put your code into a **R code chunk**. Make sure both your R code chunk and its output are visible in the rendered `pdf/html` file.
- For this homework, use **ggplot2** to visualize the data. Do **NOT** use R base graphics functions.
- **Please comment your R code thoroughly, and follow the R coding style guideline (https://google.github.io/styleguide/Rguide.xml). Partial credit will be deducted for insufficient commenting or poor coding styles.**
- If you have any question about this homework assignment, we encourage you to post it on **Piazza**.

#### Homework submission guideline

- **This homework is DUE at *11:59 PM* on *Sunday April 20, 2025*.**
- Late submission penalties.

  - Submissions up to 24 hours late will incur a 10% deduction.
  - Submissions up to 48 hours late will incur a 30% deduction.

- **If you are using one or both of your free late days, please state here: INSERT_YOUR_ANSWER**
- After you complete all questions, save your `rmd` file to `FirstnameLastname-SID-HW2.rmd` and save the rendered pdf file to `FirstnameLastname-SID-HW2.pdf`. If you can not knit it to pdf, knit it to html first and then print/save it to pdf format.
- Submit **BOTH your source `rmd` file and the knitted `pdf` file** to **GradeScope**. Do NOT create a zip file. For the `pdf` submission, please tag specific pages that correspond with each question in the assignment.
- You can submit multiple times, you last submission will be graded.

---

## Acknowledgments

Please list all the help you have received for completing this homework.

**INSERT_YOUR_ANSWER** I used geeksforgeeks for R code help

---

### Load necessary packages

It is a recommended best practice to load all of your R packages and datasets at the beginning of your file in one chunk.

```r
# install the tidyverse package first if you have not done it yet.
#install.packages("tidyverse") # you can comment out this line after you have installed `tidyverse`

library(tidyverse) # for the `ggplot2` package
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
library(datasets) # for the `quakes` dataset
```

---

## Question 1 [60pt] Visualization of the mpg dataset

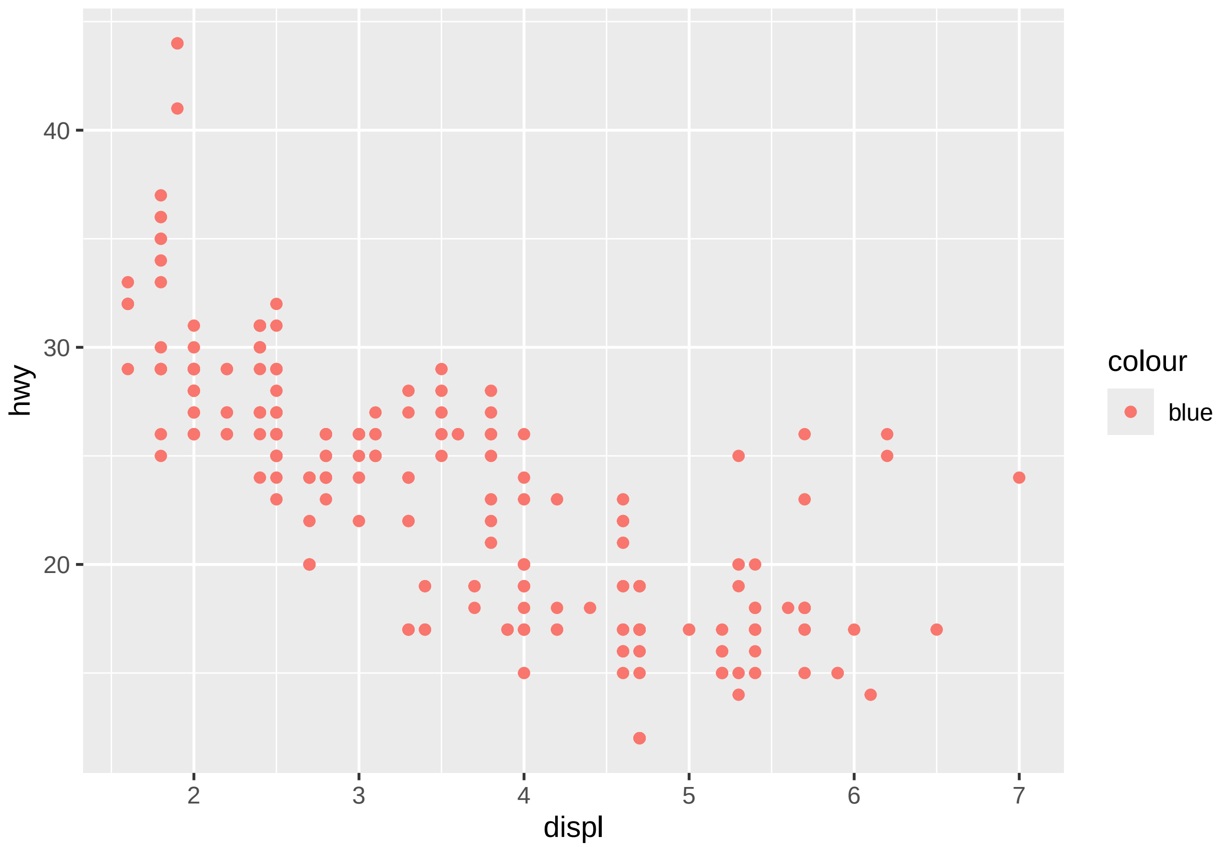
This dataset contains fuel economy data 1999 - 2008 for 38 popular car models (source: EPA http://fueleconomy.gov).

```
?mpg
## starting httpd help server ... done
dim(mpg)  # dimension of the table
## [1] 234  11
names(mpg)  # list the variables in mpg
##  [1] "manufacturer" "model"        "displ"        "year"         "cyl"
##  [6] "trans"        "drv"          "cty"          "hwy"          "fl"
## [11] "class"
str(mpg)  # list the structures in mpg
## tibble [234 x 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" ...
glimpse(mpg) # get a glimpse of the mpg data
## Rows: 234
## Columns: 11
## $ manufacturer <chr> "audi", "audi", "audi", "audi", "audi", "audi", "audi", "~
## $ model        <chr> "a4", "a4", "a4", "a4", "a4", "a4", "a4", "a4 quattro", "~
## $ displ        <dbl> 1.8, 1.8, 2.0, 2.0, 2.8, 2.8, 3.1, 1.8, 1.8, 2.0, 2.0, 2.~
## $ year         <int> 1999, 1999, 2008, 2008, 1999, 1999, 2008, 1999, 1999, 200~
## $ cyl          <int> 4, 4, 4, 4, 6, 6, 6, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6, 8, 8, ~
## $ trans        <chr> "auto(l5)", "manual(m5)", "manual(m6)", "auto(av)", "auto~
## $ drv          <chr> "f", "f", "f", "f", "f", "f", "f", "4", "4", "4", "4", "4~
## $ cty          <int> 18, 21, 20, 21, 16, 18, 18, 18, 16, 20, 19, 15, 17, 17, 1~
## $ hwy          <int> 29, 29, 31, 30, 26, 26, 27, 26, 25, 28, 27, 25, 25, 25, 2~
## $ fl           <chr> "p", "p", "p", "p", "p", "p", "p", "p", "p", "p", "p", "p~
## $ class        <chr> "compact", "compact", "compact", "compact", "compact", "c~
```

### (a) [10pt] Aesthetic mapping of color

(i) What's gone wrong with the following code? Why are the points not blue?

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

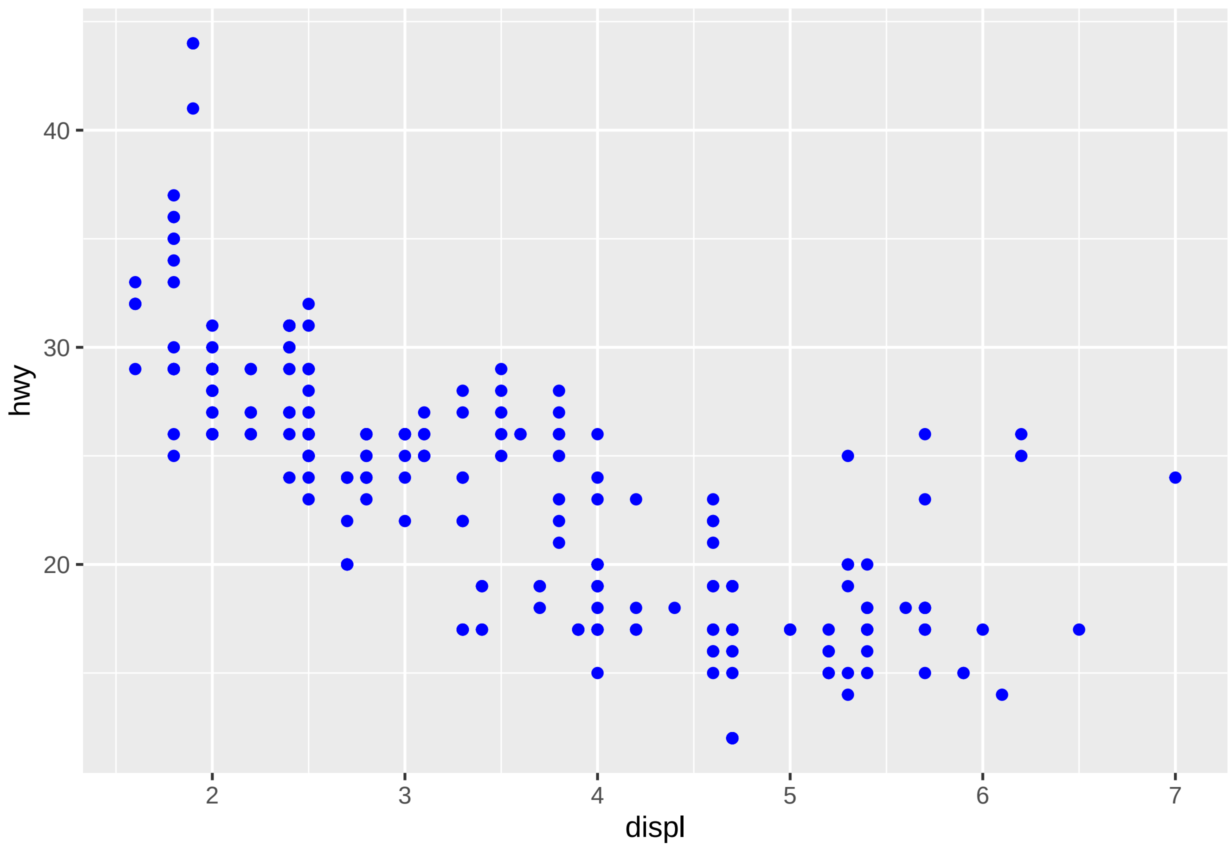


**INSERT_YOUR_ANSWER** Blue was set inside of aes, which makes ggplot think of it as a variable or factor, rather than a color.

Modify the code to plot blue points.

**INSERT_YOUR_ANSWER**

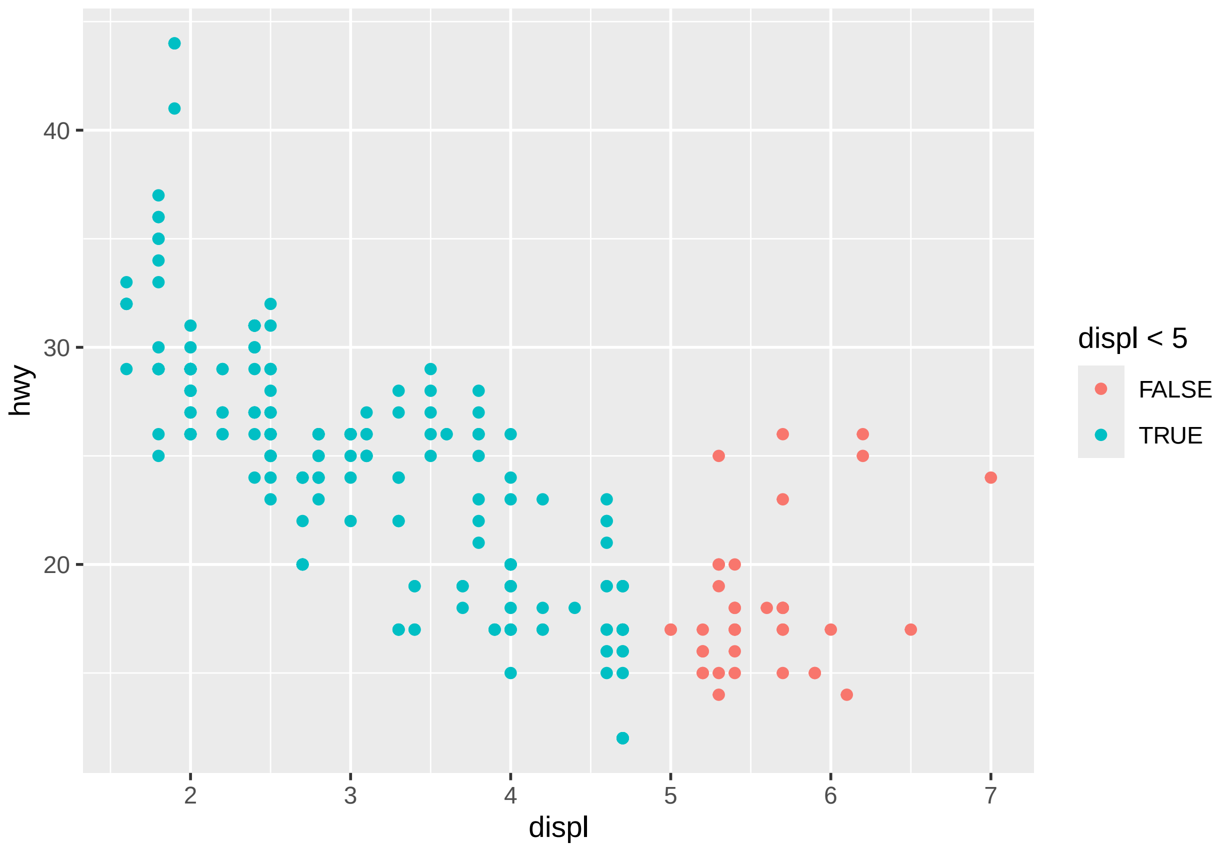
```
ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy), color = "blue")
```



(ii) 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.

**INSERT_YOUR_ANSWER**

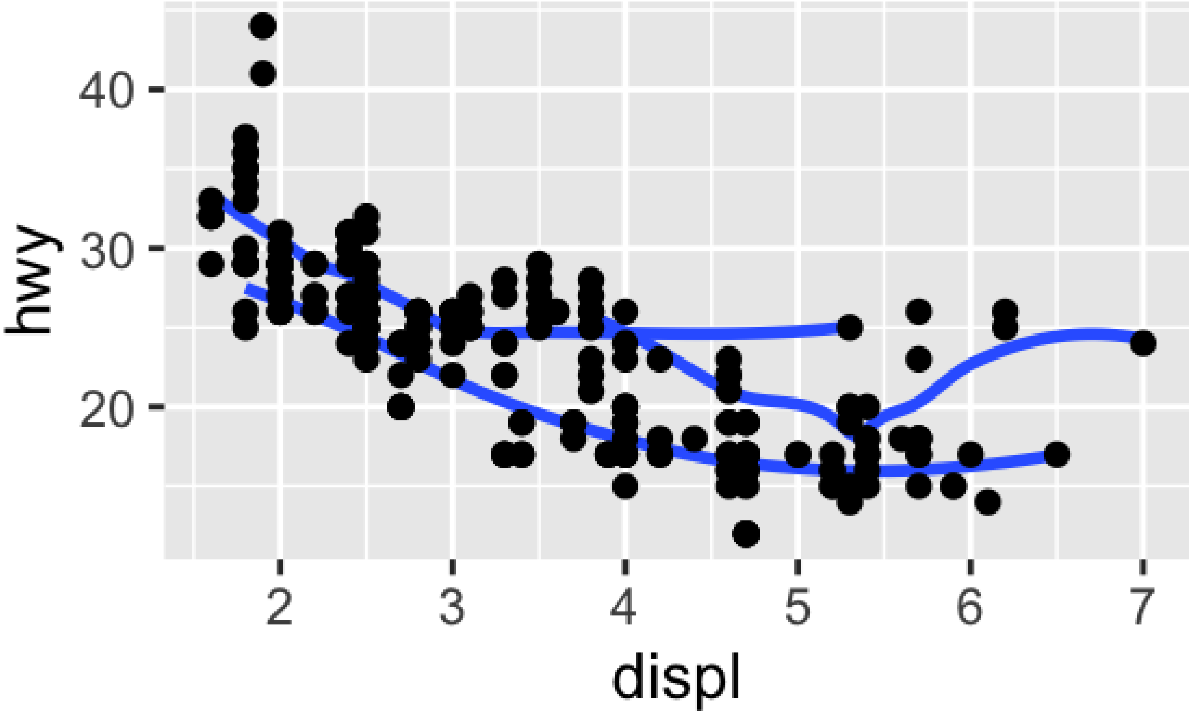
```
ggplot(data = mpg) +geom_point(mapping = aes(x = displ, y = hwy, color = displ < 5))
```



**(b) [20pt] Reproducing figures**

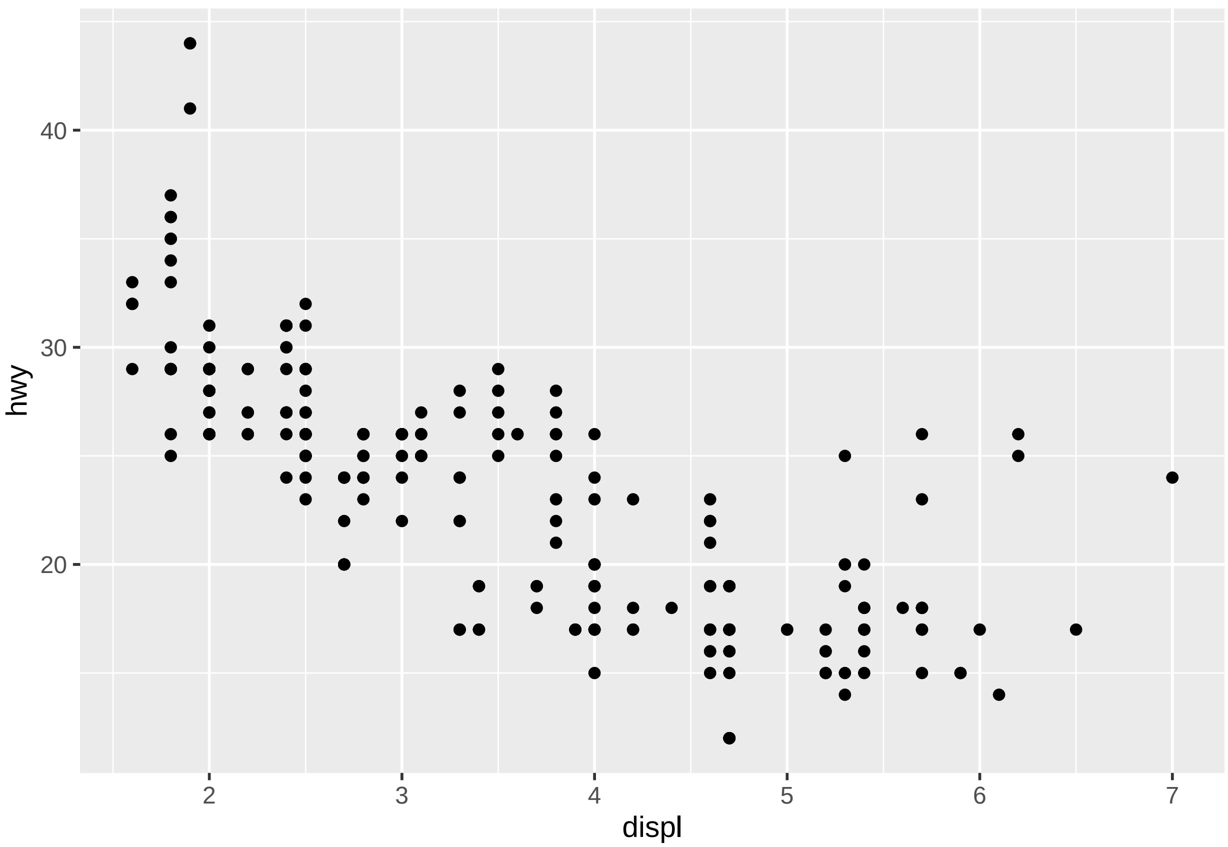
Write your own `ggplot2` code to reproduce the following four plots. When generating these plots, focus on matching the aesthetic mappings in the reference plot. It is acceptable if the text font size or point size in your plots look different from the reference plots.

(i) Reproduce this plot:

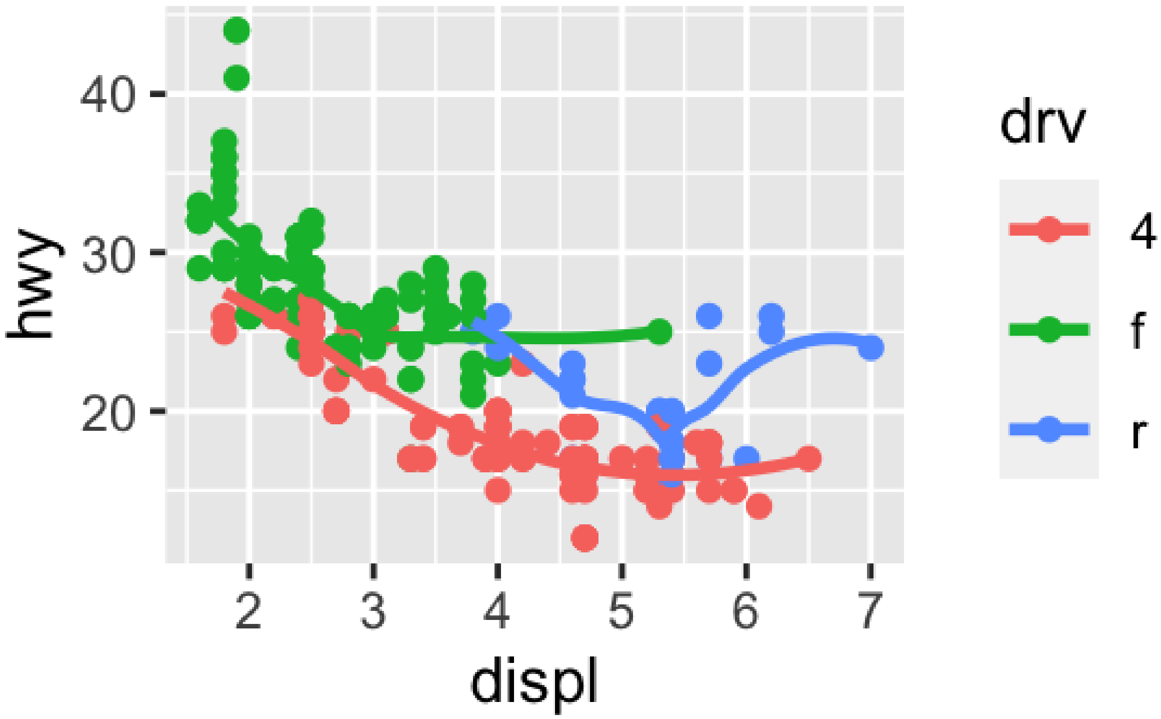


**INSERT_YOUR_ANSWER**

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

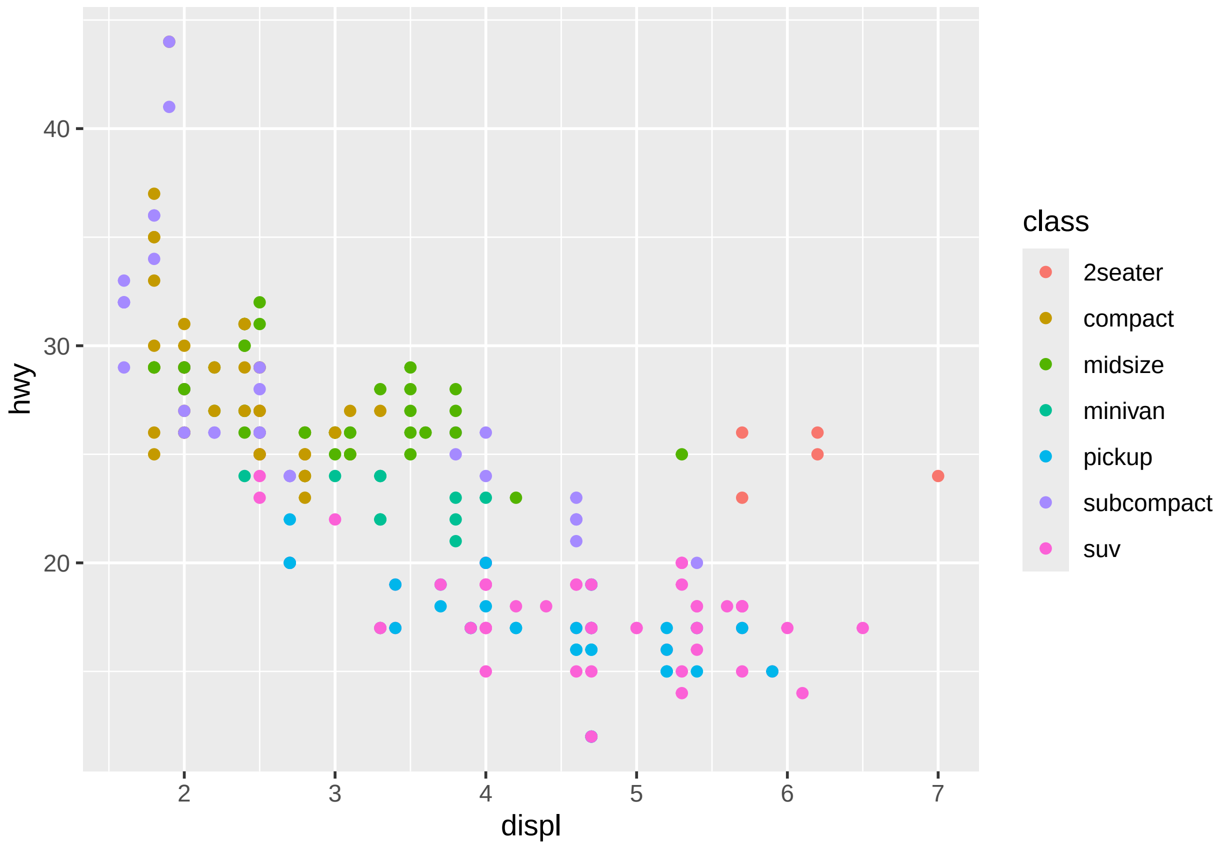


(ii) Reproduce this plot:

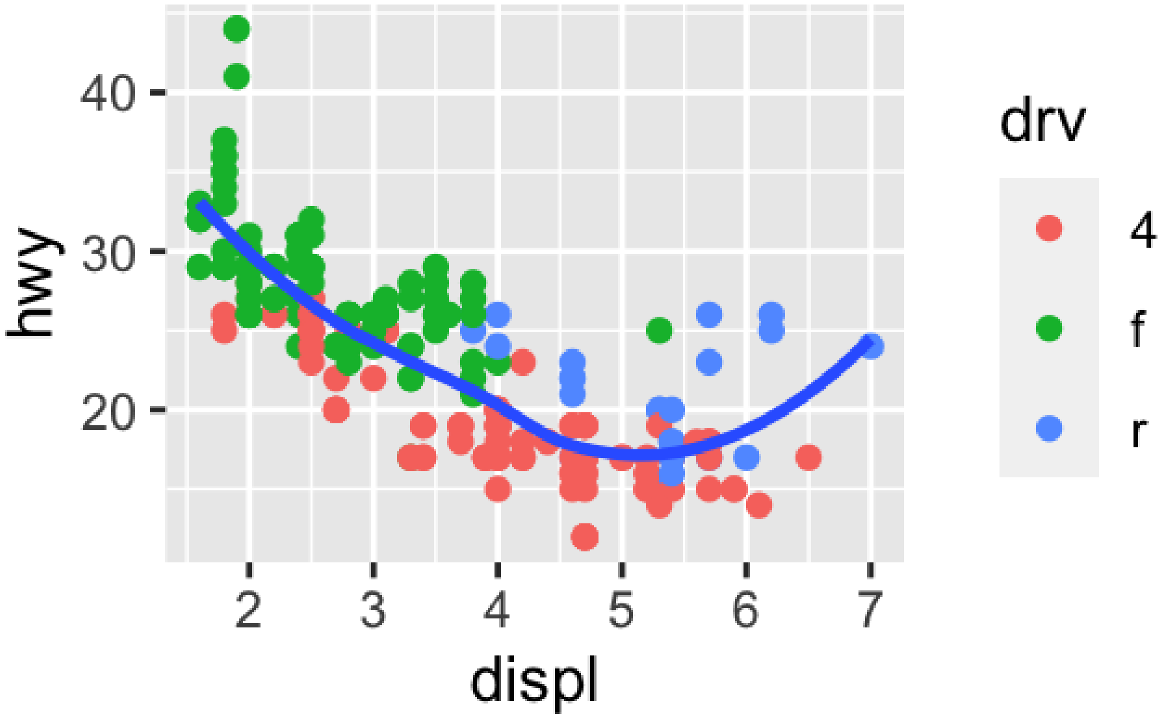


**INSERT_YOUR_ANSWER**

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



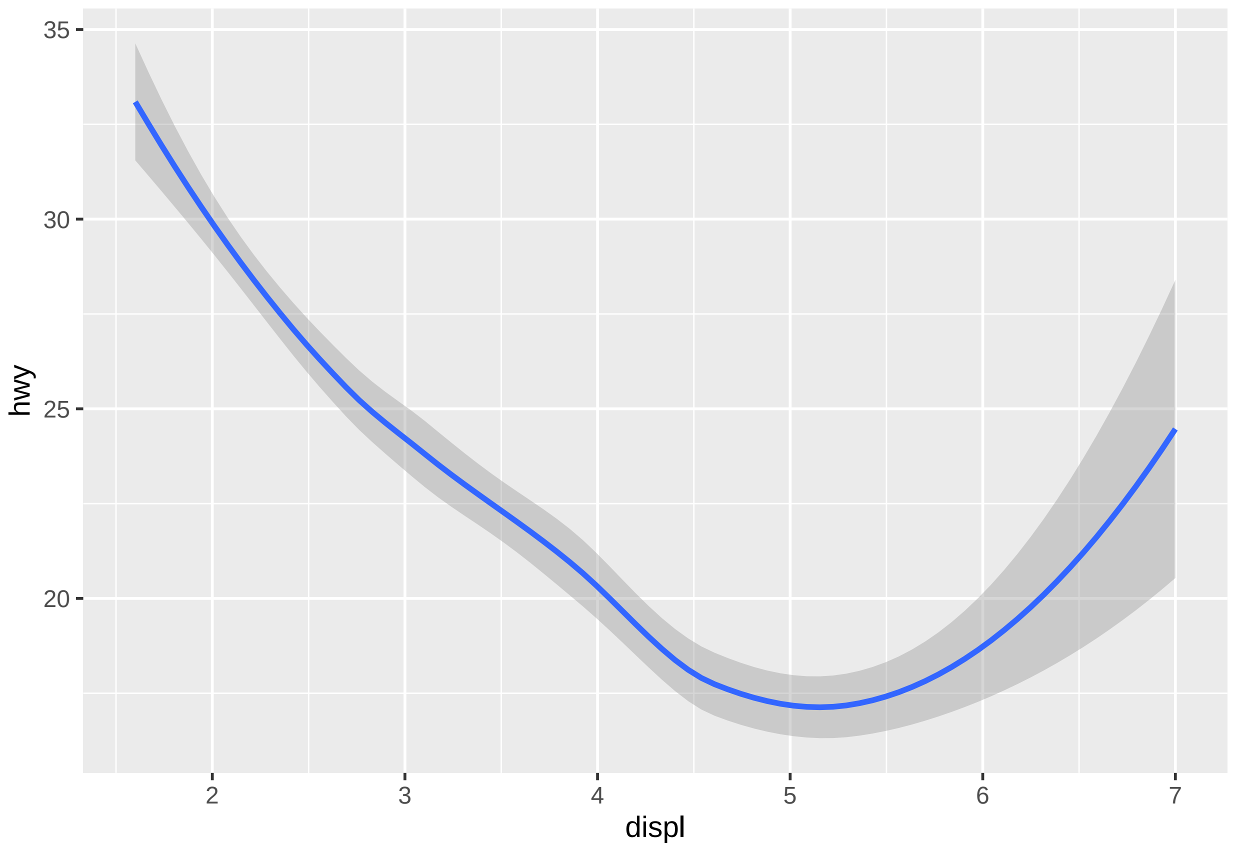
(iii) Reproduce this plot:



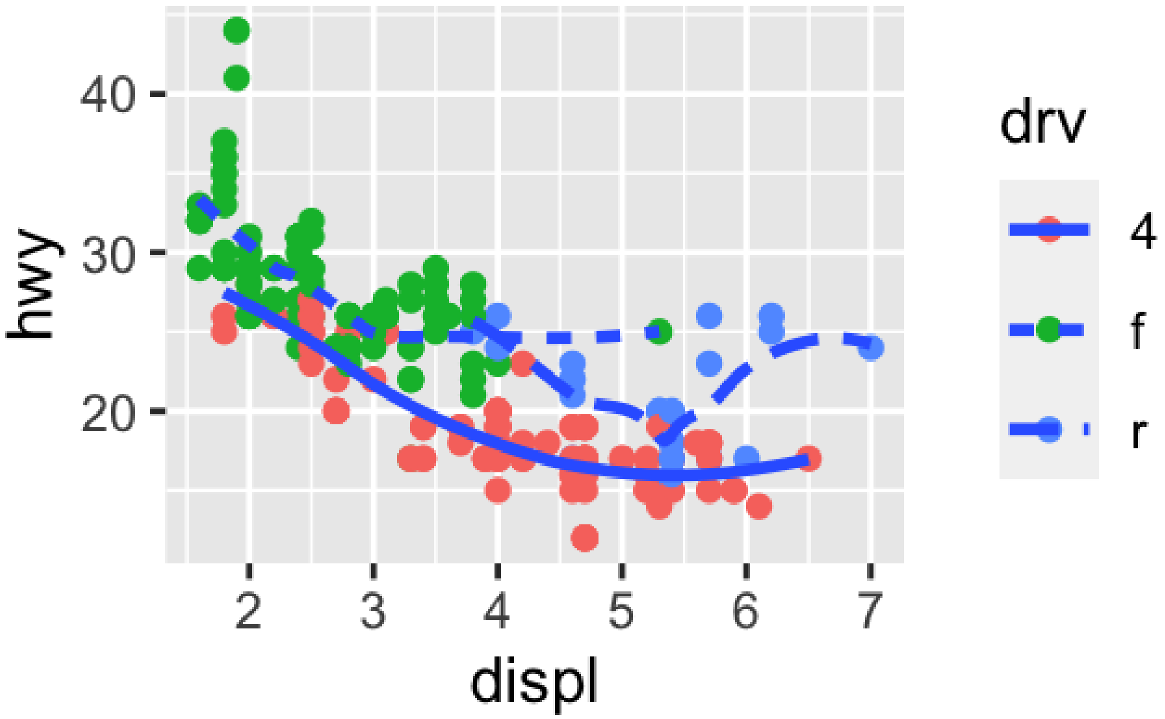
**INSERT_YOUR_ANSWER**

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

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



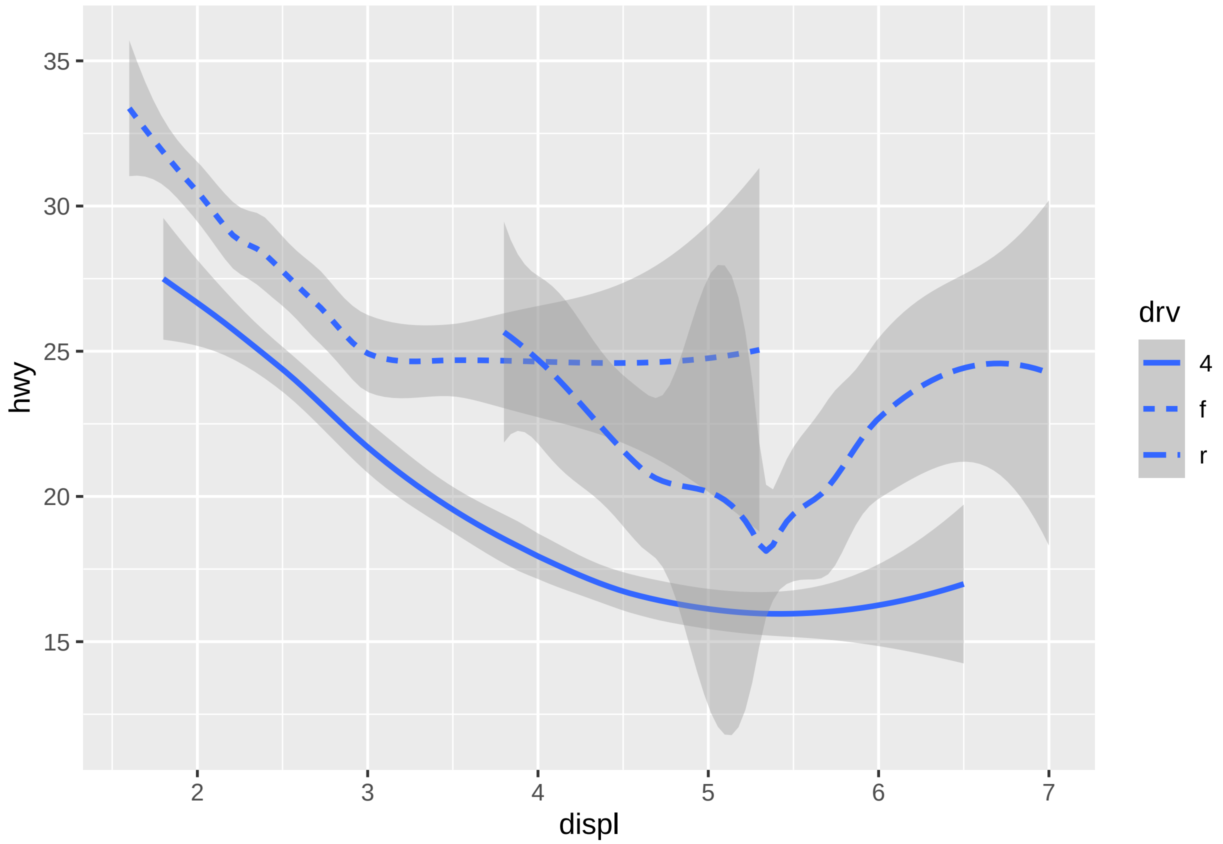
(iv) Reproduce this plot:



**INSERT_YOUR_ANSWER**

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

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



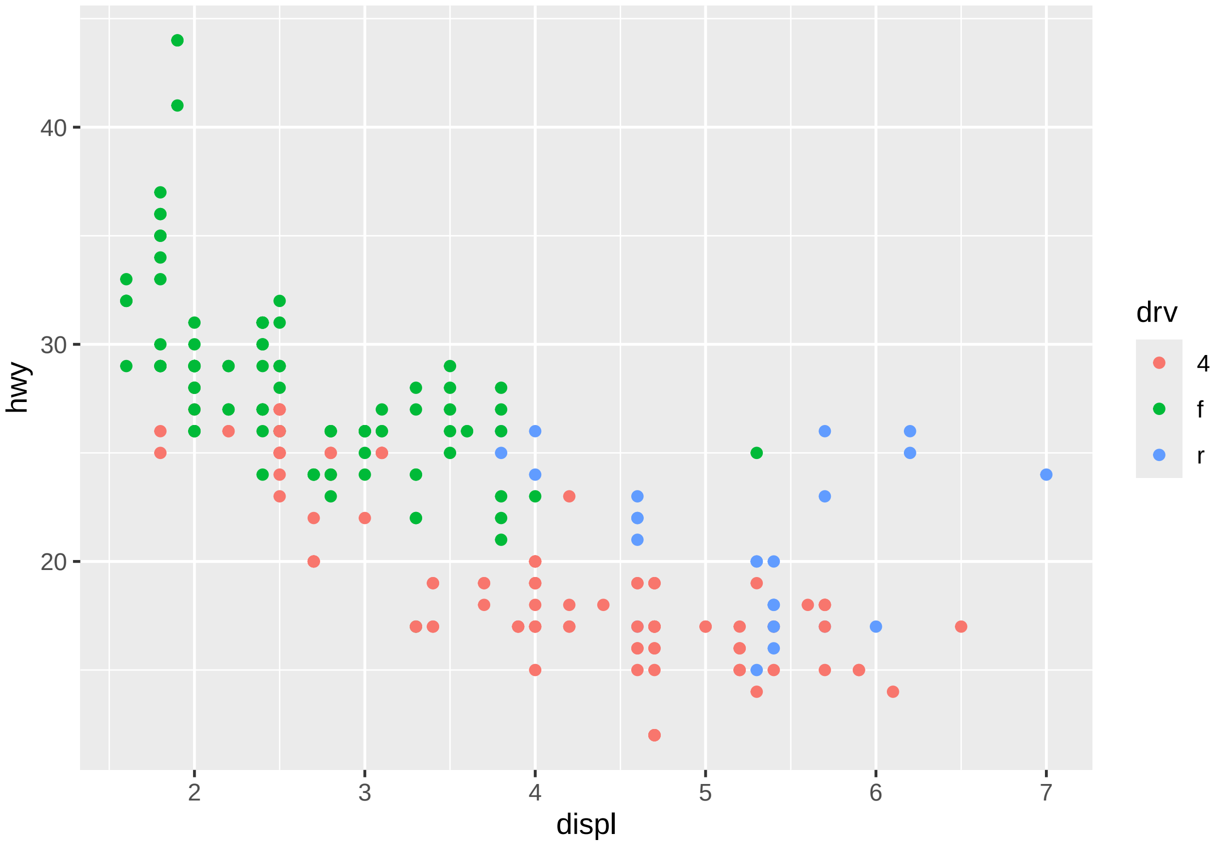
**(c) [10pt] Facets**

There are two ways to add additional variable(s) to a 2D plot. One is using aesthetics, the other one is using facets.

(i) Make a scatter plot that x-axis is `displ` and y-axis is `hwy`. Use different colors to distinguish `drv` types.

**INSERT_YOUR_ANSWER**

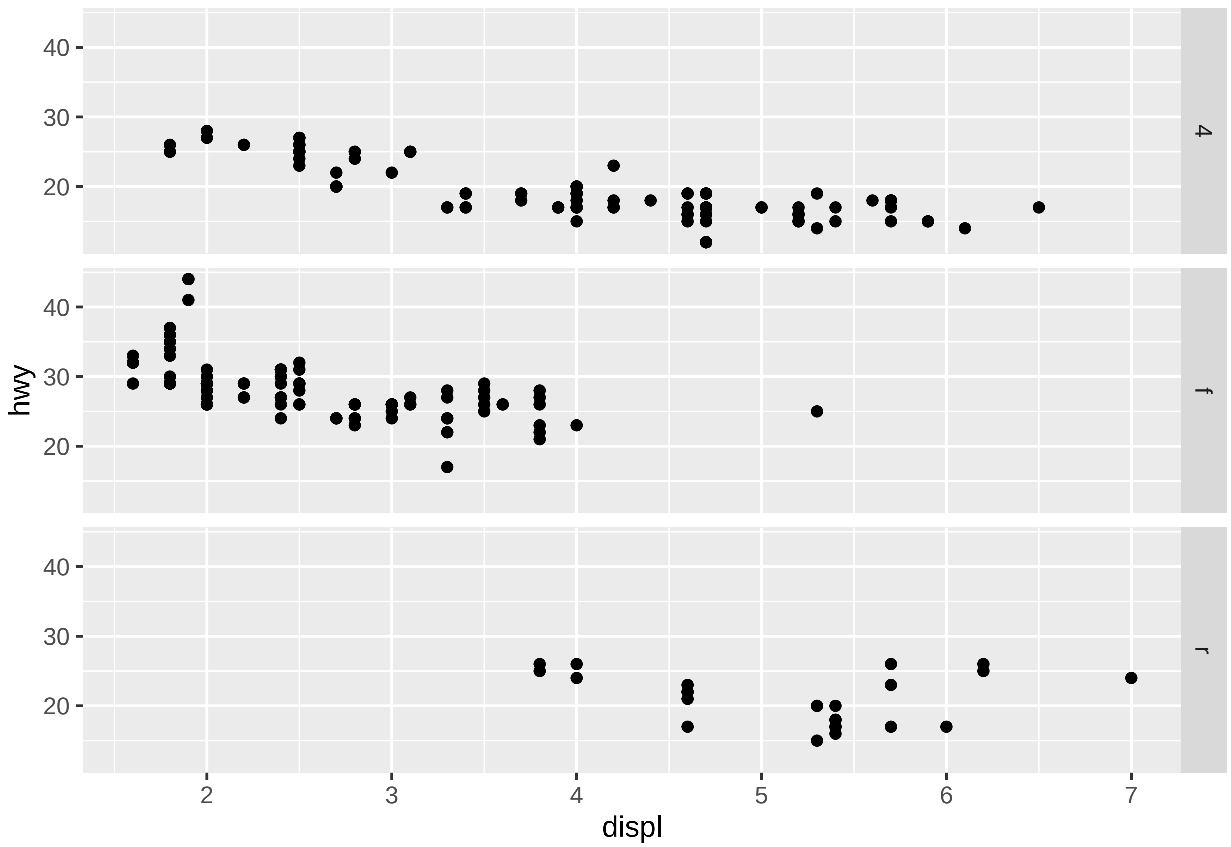
```
ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy, color = drv))
```



(ii) Facet `drv` into the rows. That is, make several **rows** of subplots, one row for each `drv` type. Each subplot has `displ` mapped to the x-axis and `hwy` mapped to the y-axis. **Hint**: Use `nrow` or `ncol` to control the layout of the individual panels.

**INSERT_YOUR_ANSWER**

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



**(d) [10pt] `stat` functions**

Most `geom` functions and `stat` functions come in pairs that are almost always used in concert.

- every `geom` has a default `stat`
- every `stat` has a default `geom`

Look up the default `stat` functions for the `geom` functions listed in the following table. Also, find out the variables computed by the default `stat` function (See the **Computed variables** section in the documentation page).

**INSERT_YOUR_ANSWER (in the table)**

| `geom` function | default `stat` function | variables computed by the default `stat` function |
|---|---|---|
| `geom_bar()` | stat_count | prop, count |
| `geom_histogram()` | stat_bin | count, ncount, density, ndensity |
| `geom_density()` | stat_density | scaled, count, density |
| `geom_point()` | stat_identity | na |
| `geom_smooth()` | stat_smooth | x, y, ymin/max, standard error (se) |

Some `geom` function has stat = "identity" as the default. What does that mean?

**INSERT_YOUR_ANSWER** You fill in its identity by yourself. Fill in the x/y values.

**Notes**: Table formatting are sometimes tricky using R Markdown. Table Generator is a handy tool if you need to make tables in the future.

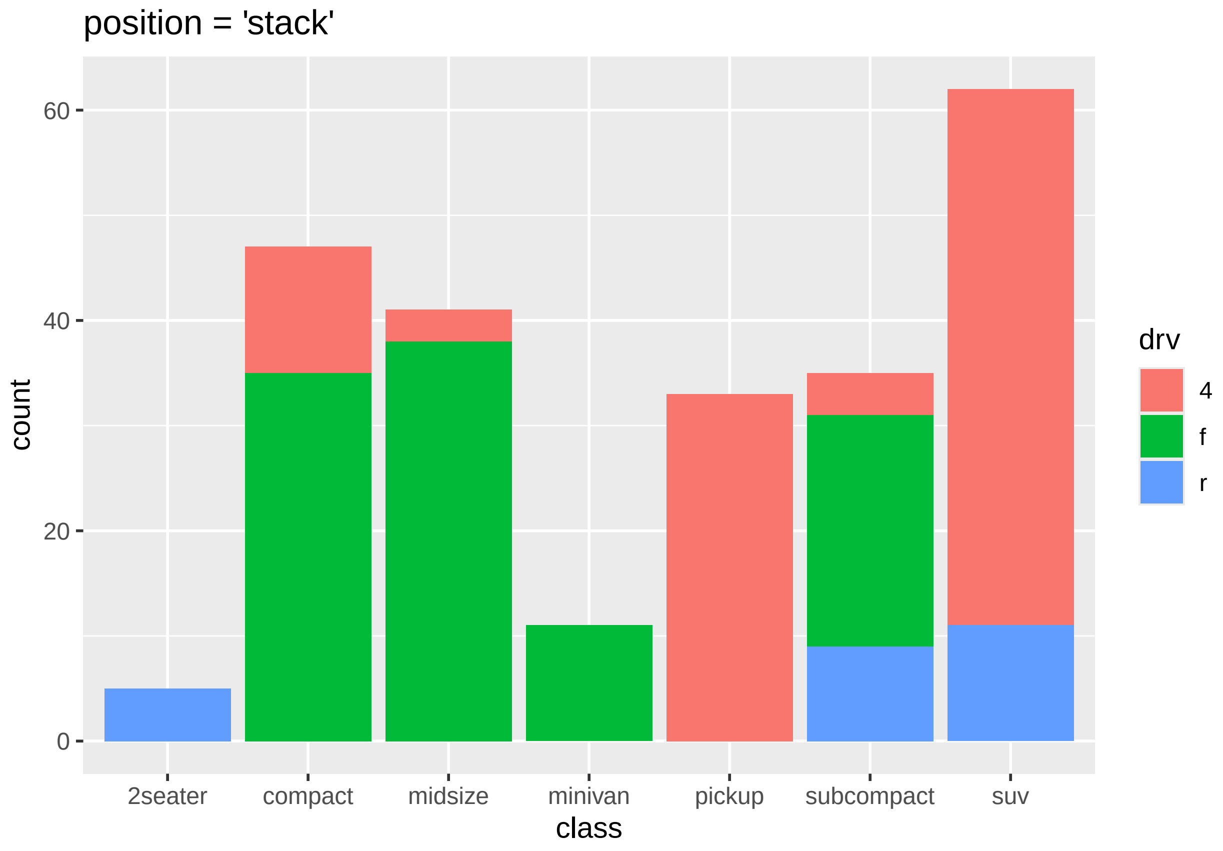
### (e) [10pt] Position adjustment options for `geom_bar()`

Choose two categorical variables from the `mpg` dataset and use them to illustrate the following four position adjustment options for `geom_bar()`:

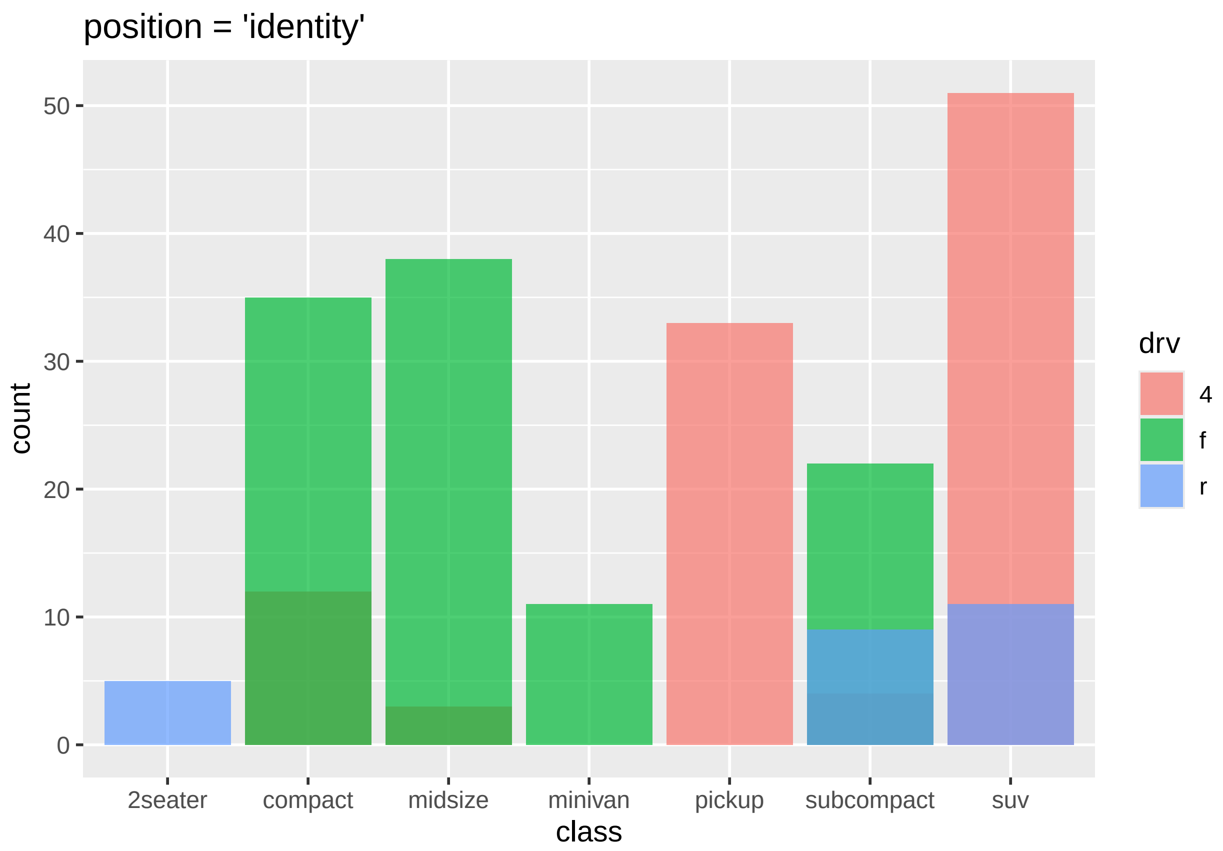
- **default**: position = "stack"
- position = "identity" will place each object exactly where it falls in the context of the graph.
- position = "fill" works like stacking, but makes each set of stacked bars the same height.
- position = "dodge" places overlapping objects directly beside one another. the bars are automatically stacked. Each colored rectangle represents a combination of cut and clarity.

**INSERT_YOUR_ANSWER**

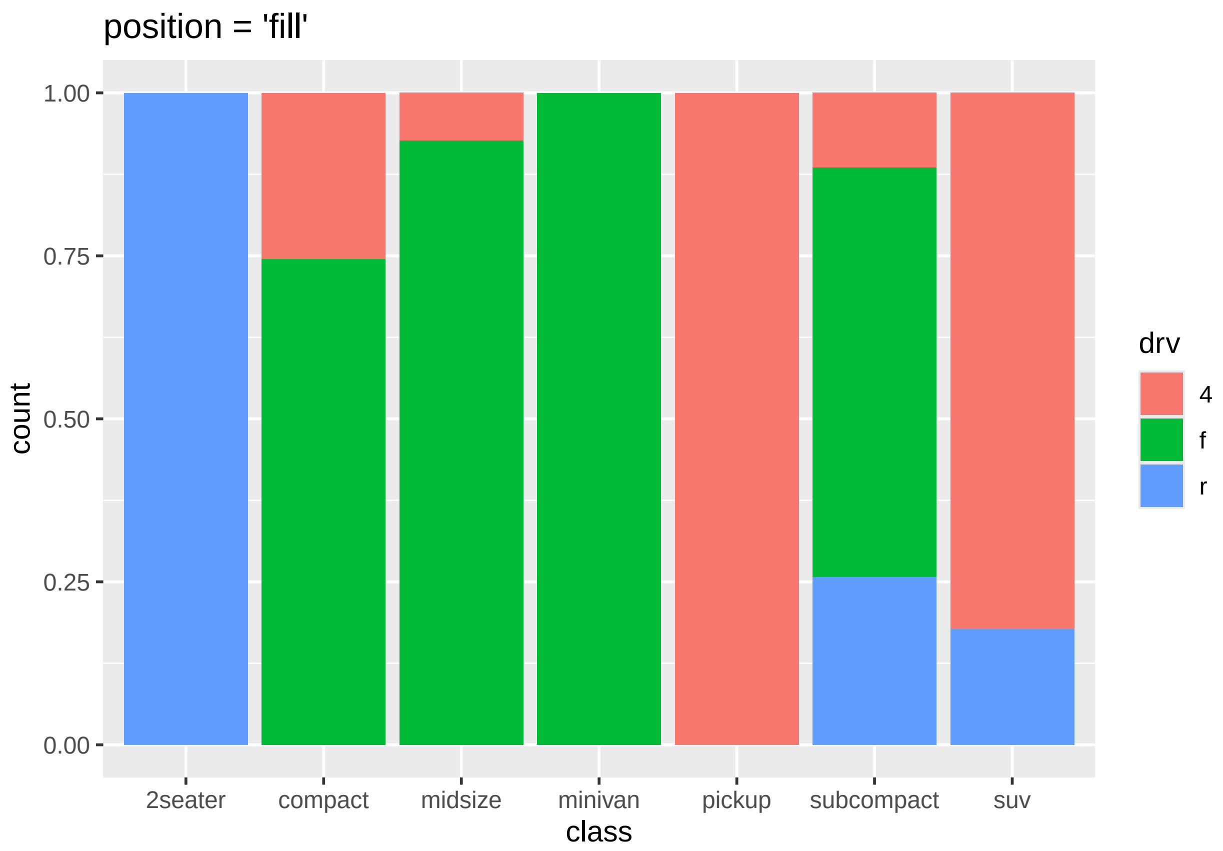
```
# class and drv
ggplot(data = mpg, aes(x = class, fill = drv)) + geom_bar(position = "stack") + ggtitle("position = 'sta
```



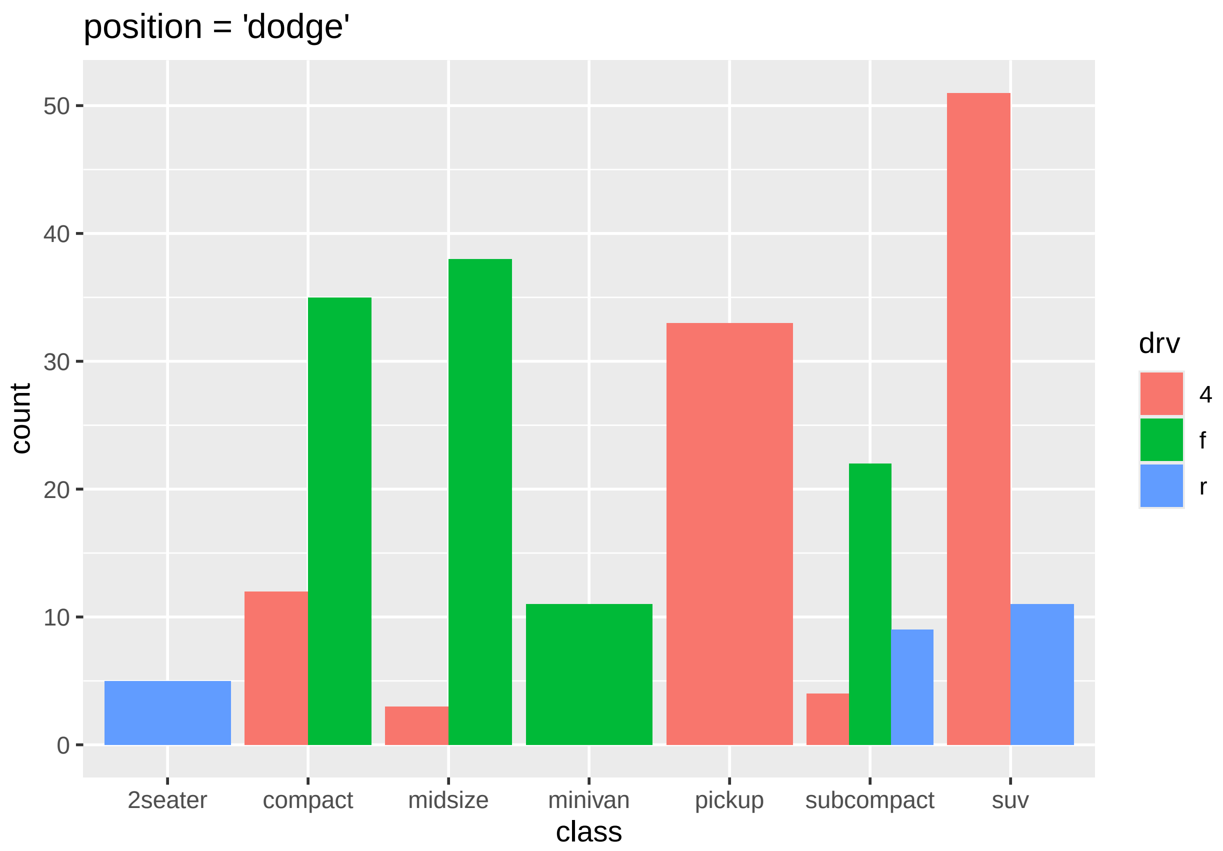
```
ggplot(data = mpg, aes(x = class, fill = drv)) + geom_bar(position = "identity", alpha = 0.7) + ggtitle
```



```
ggplot(data = mpg, aes(x = class, fill = drv)) + geom_bar(position = "fill") + ggtitle("position = 'fil
```



```
ggplot(data = mpg, aes(x = class, fill = drv)) + geom_bar(position = "dodge") + ggtitle("position = 'doc
```



Which position option do you like most? What conclusions can you draw from your plot?

**INSERT_YOUR_ANSWER** I like both identity and dodge, since they more clearly display the data. The categories (of cars) can be compared more easily ***

## Question 2 [40pt] Visualization the quakes dataset

Recall that the quakes dataset contain the locations of 1000 seismic events of MB > 4.0. The events occurred in a cube near Fiji since 1964.

```
?quakes
class(quakes)
## [1] "data.frame"
head(quakes, n=5) # print first 5 rows of quakes
##      lat   long depth mag stations
## 1 -20.42 181.62   562 4.8       41
## 2 -20.62 181.03   650 4.2       15
## 3 -26.00 184.10    42 5.4       43
## 4 -17.97 181.66   626 4.1       19
## 5 -20.42 181.96   649 4.0       11
dim(quakes)  # dimension of the table
## [1] 1000    5
names(quakes)  # list the variables in quakes
## [1] "lat"      "long"     "depth"    "mag"      "stations"
str(quakes)  # list the structures in quakes
## 'data.frame':    1000 obs. of  5 variables:
##  $ lat     : num  -20.4 -20.6 -26 -18 -20.4 ...
##  $ long    : num  182 181 184 182 182 ...
##  $ depth   : int  562 650 42 626 649 195 82 194 211 622 ...
##  $ mag     : num  4.8 4.2 5.4 4.1 4 4 4.8 4.4 4.7 4.3 ...
##  $ stations: int  41 15 43 19 11 12 43 15 35 19 ...
glimpse(quakes) # get a glimpse of the quakes data
## Rows: 1,000
## Columns: 5
## $ lat      <dbl> -20.42, -20.62, -26.00, -17.97, -20.42, -19.68, -11.70, -28.1~
## $ long     <dbl> 181.62, 181.03, 184.10, 181.66, 181.96, 184.31, 166.10, 181.9~
## $ depth    <int> 562, 650, 42, 626, 649, 195, 82, 194, 211, 622, 583, 249, 554~
## $ mag      <dbl> 4.8, 4.2, 5.4, 4.1, 4.0, 4.0, 4.8, 4.4, 4.7, 4.3, 4.4, 4.6, 4~
## $ stations <int> 41, 15, 43, 19, 11, 12, 43, 15, 35, 19, 13, 16, 19, 10, 94, 1~
```

### (a) [30pt] Re-plotting the distribution of earthquake magnitudes.

In Homework 1, Question 1(a), you've plotted the distribution of the earthquake magnitudes using R base graphics. This week, write your own ggplot2 code to reproduce the following four subfigures in a 2-by-2 layout.

- subfigure #1: plot a density histogram of the earthquake magnitudes, and then plot the estimated probability density curve in red color in the same plot
- subfigure #2: plot a horizontal boxplot of the earthquake magnitudes
- subfigure #3: plot the empirical cdf of the earthquake magnitudes
- subfigure #4: make a Q-Q plot to compare the observed earthquake magnitudes distribution with the Normal distribution. Add a *thick* Q-Q line in blue color.

**Hints**:

- In the lecture notes, we used the grid.arrange() function from the gridExtra package (see this vignette page) to layout multiple plots in a single figure. For alternative options, please refer to this vignette page.

- Check out the `geom_histogram()` function for plotting the **density** histogram.
- Check out the `stat_ecdf()` function for plotting the empirical distribution.
- Check out the `geom_qq()` and `geom_qq_line()` functions for plotting the Q-Q plot and Q-Q line.

**INSERT_YOUR_ANSWER**

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
install.packages("gridExtra")
```

```
## Warning: package 'gridExtra' is in use and will not be installed
```

```
# Density
p1 <- ggplot(quakes, aes(x = mag)) +
  geom_histogram(aes(y = ..density..), bins = 30, fill = "lightblue", color = "black") +
  geom_density(color = "red", size = 1) +
  labs(title = "Density Histogram of Magnitude", x = "Magnitude", y = "Density")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

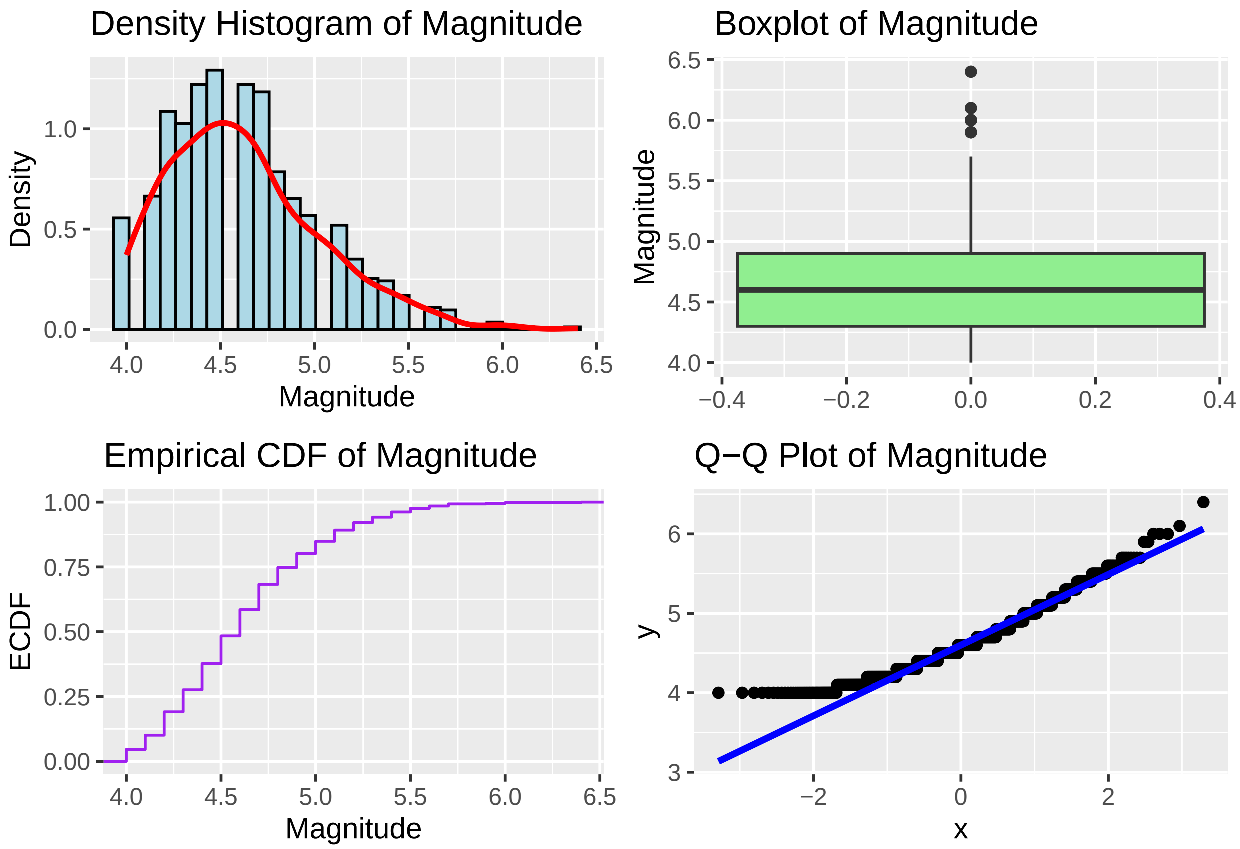
```
# Boxplot
p2 <- ggplot(quakes, aes(x = mag)) +
  geom_boxplot(fill = "lightgreen") +
  coord_flip() +
  labs(title = "Boxplot of Magnitude", x = "Magnitude")

# ecdf
p3 <- ggplot(quakes, aes(x = mag)) +
  stat_ecdf(geom = "step", color = "purple") +
  labs(title = "Empirical CDF of Magnitude", x = "Magnitude", y = "ECDF")

# QQ thick
p4 <- ggplot(quakes, aes(sample = mag)) +
  geom_qq(color = "black") +
  geom_qq_line(color = "blue", size = 1.2) +
  labs(title = "Q-Q Plot of Magnitude")

grid.arrange(p1, p2, p3, p4, nrow = 2)
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



**(b)** [10pt] **Earthquake location map**

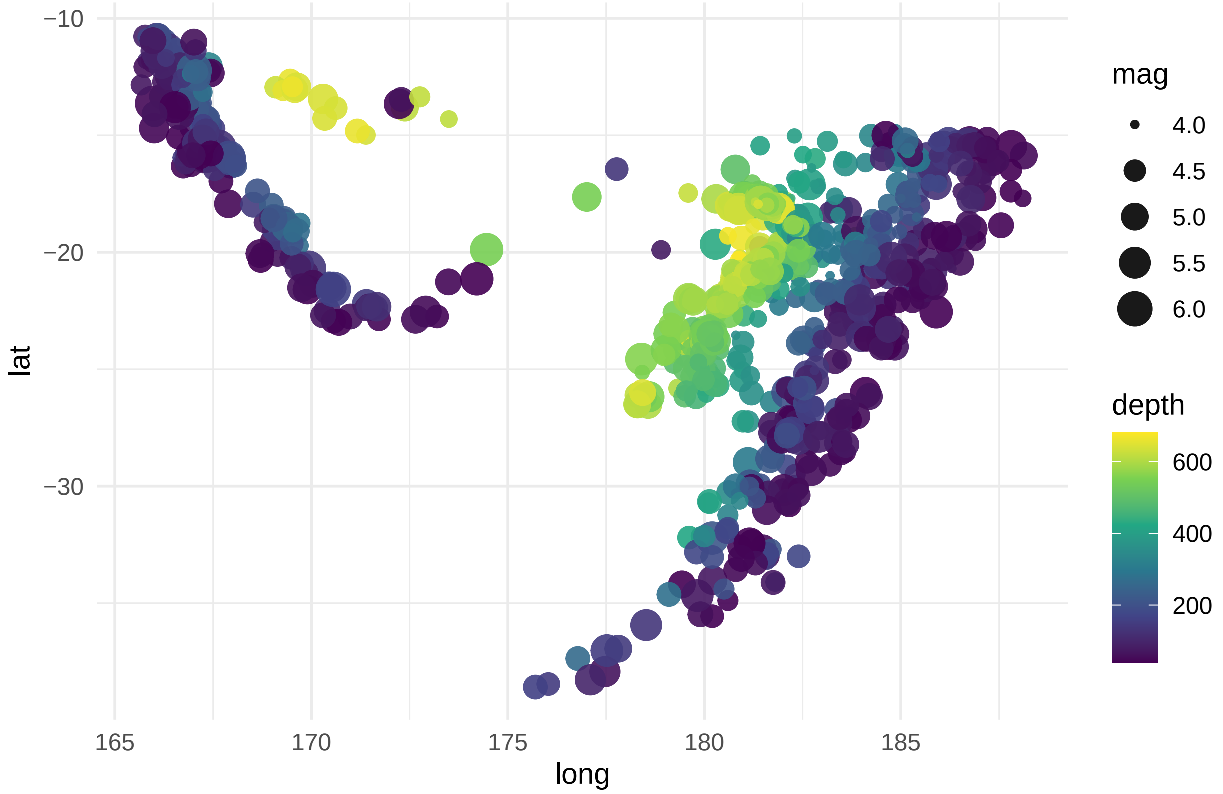
Make a scatter plot of the earthquake locations. Use `long` as the x-axis and `lat` as the y-axis. Map `mag` to the size aesthetic and `depth` to the color aesthetic.

[**Extra credit 10pt**] If you can plot earthquakes point on top of a map layer, you will earn extra points!

**INSERT_YOUR_ANSWER**

```r
ggplot(quakes, aes(x = long, y = lat)) +
  geom_point(aes(size = mag, color = depth), alpha = 0.9) +
  scale_color_viridis_c() + theme_minimal() + ggtitle("Earthquakes (size = magnitude, color = depth)")
```

Earthquakes (size = magnitude, color = depth)


lat
long
mag
depth