Assignment 4

Chang An Le Harry Jr

3/30/2022

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.4.1     v purrr   1.0.1
## v tibble  3.1.7     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

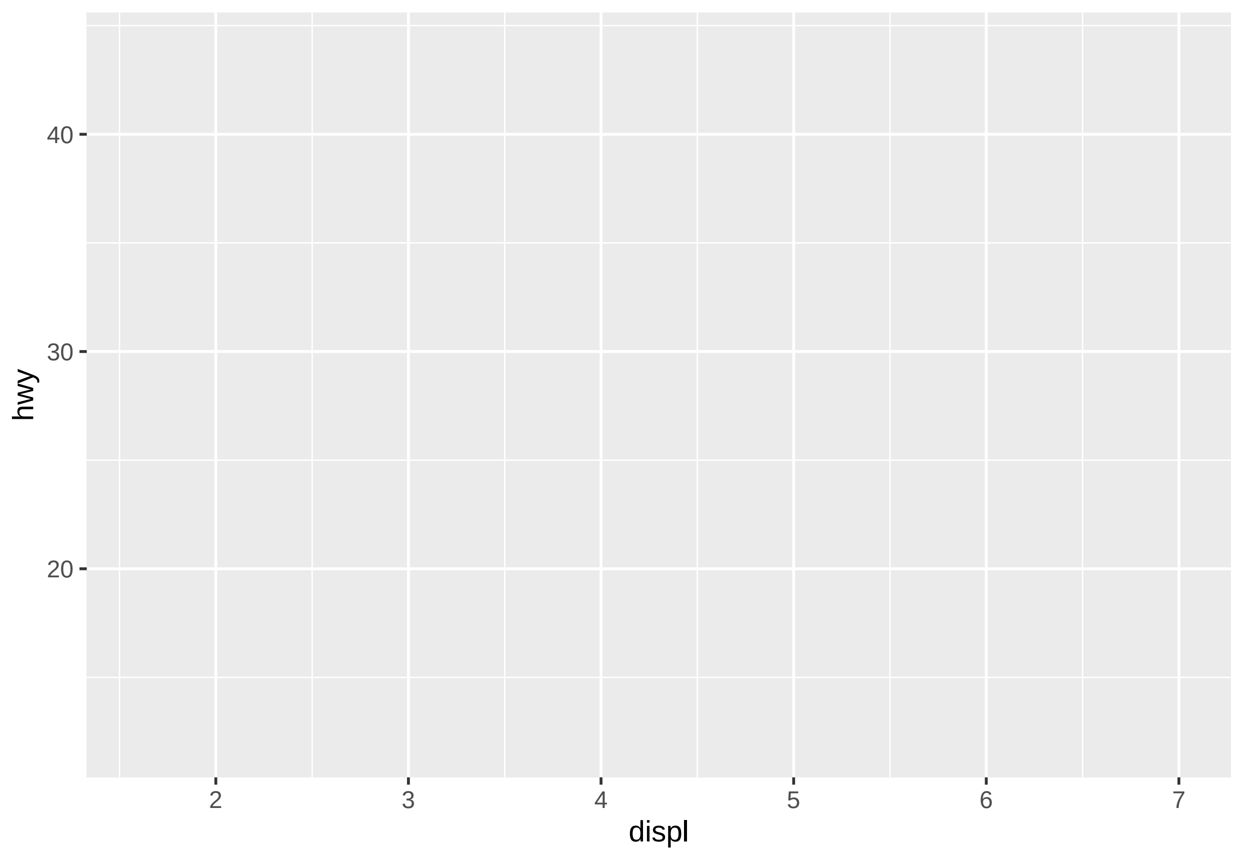
```
library(nycflights13)
```

Question 1. Develop a plot layer-by-layer

So far we have learned different geoms to build plots in ggplot2. In this question, we will develop a plot layer-by-layer to understand each individual steps. We will also explore different geoms available in ggplot2.

1. We first create a plot with the mpg dataset in R and set the global aesthetic mappings. Describe (in words) the output of the following code.

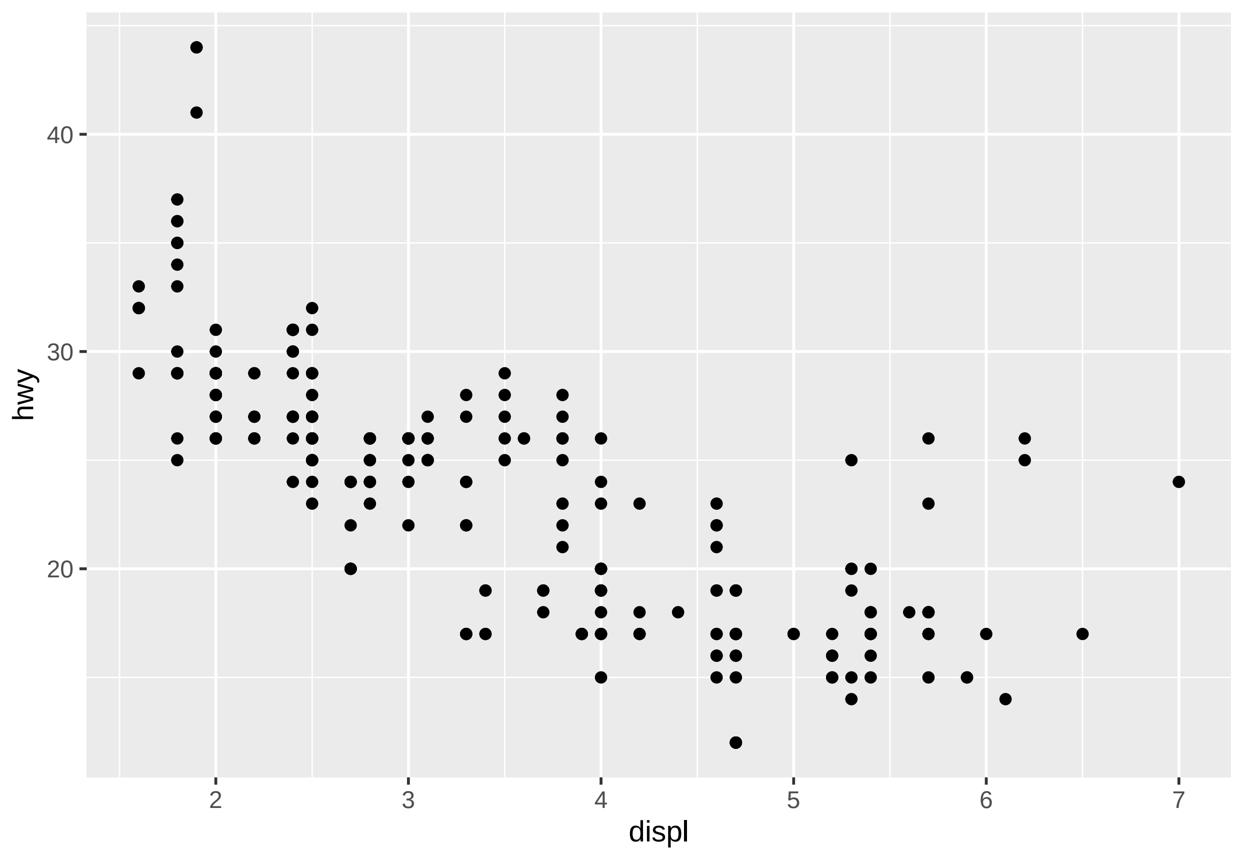
```
p = ggplot(mpg, aes(x = displ, y = hwy))
p
```



The above output shows an empty plot, with the displ variable on the horizontal axis and the hwy variable on the vertical axis.

2. Next, we add a geom_point() layer. Describe (in words) the output of the following code.

```
p + geom_point()
```

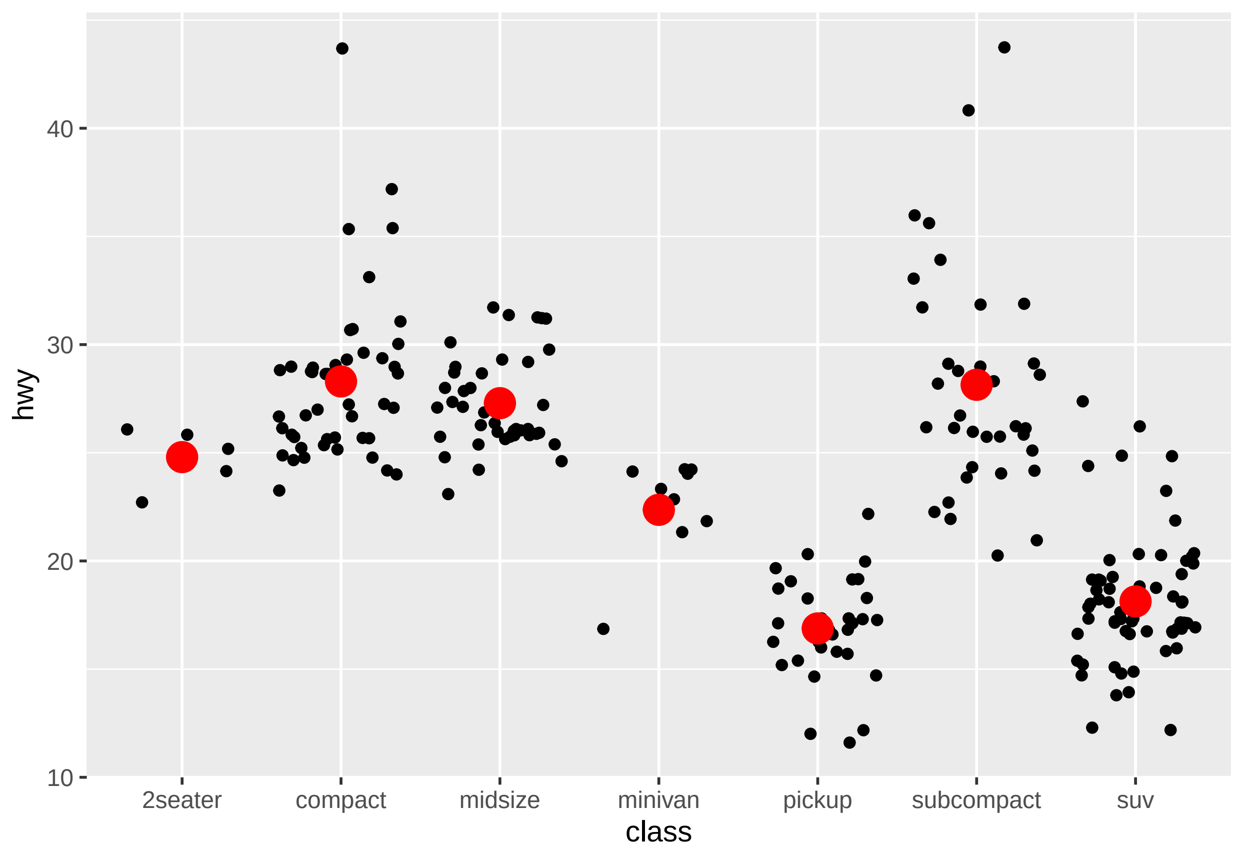


The above plot shows a scatterplot to help visualise the relationship between the displ and hwy variables. Unlike the previous plot, this plot also includes the data points in black representing each observation, which can only be seen when including the geom_point() layer.

3.1. The following code uses tidyverse syntax to generate some summary statistics about each class of cars. Use the data to recreate the plot below:

```
stat_class = mpg %>%
  group_by(class) %>%
  summarize(avg_hwy = mean(hwy))
```

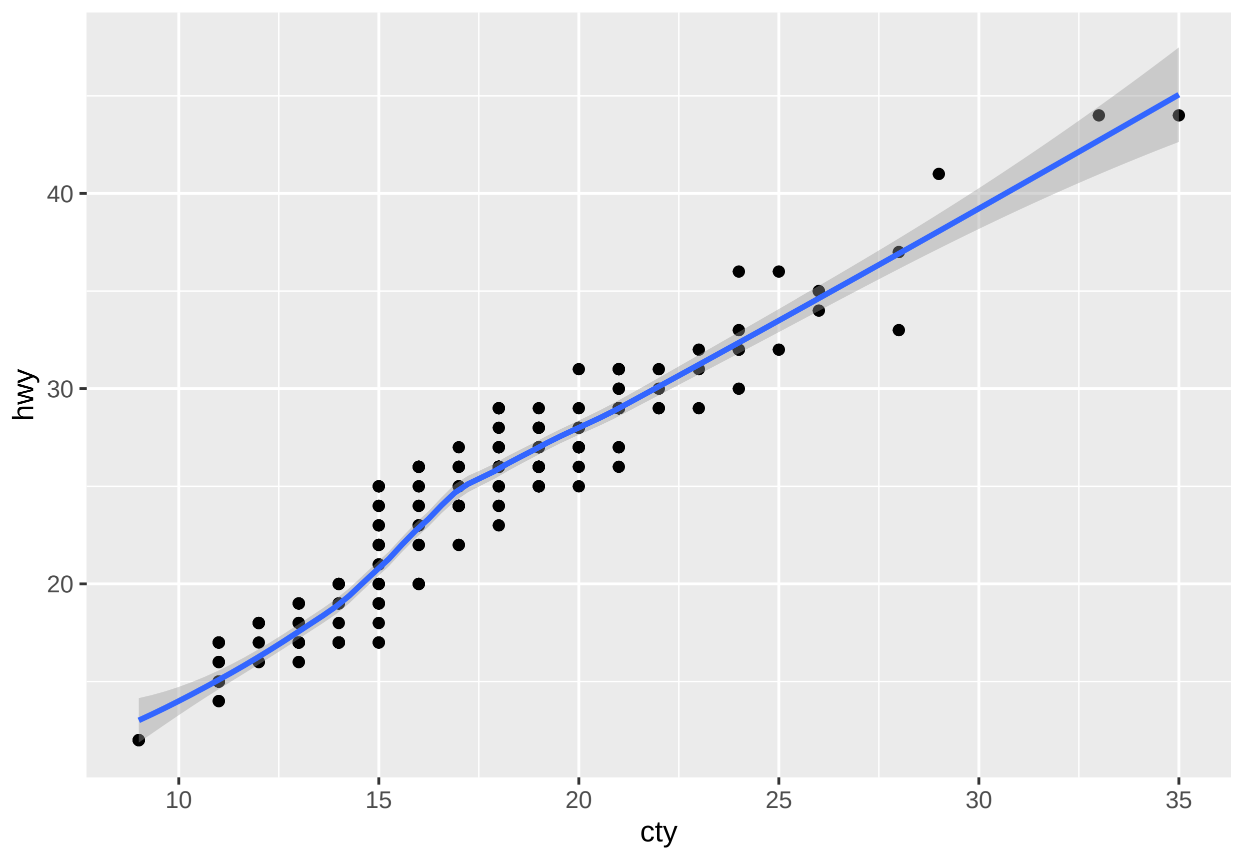
```
ggplot(data = mpg, aes(x = class, y = hwy)) + geom_jitter() +
  geom_point(data = stat_class, aes(x = class, y = avg_hwy), shape = 19, color = "red", size = 5)
```



3.2. Simplify the following plot specifications:

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

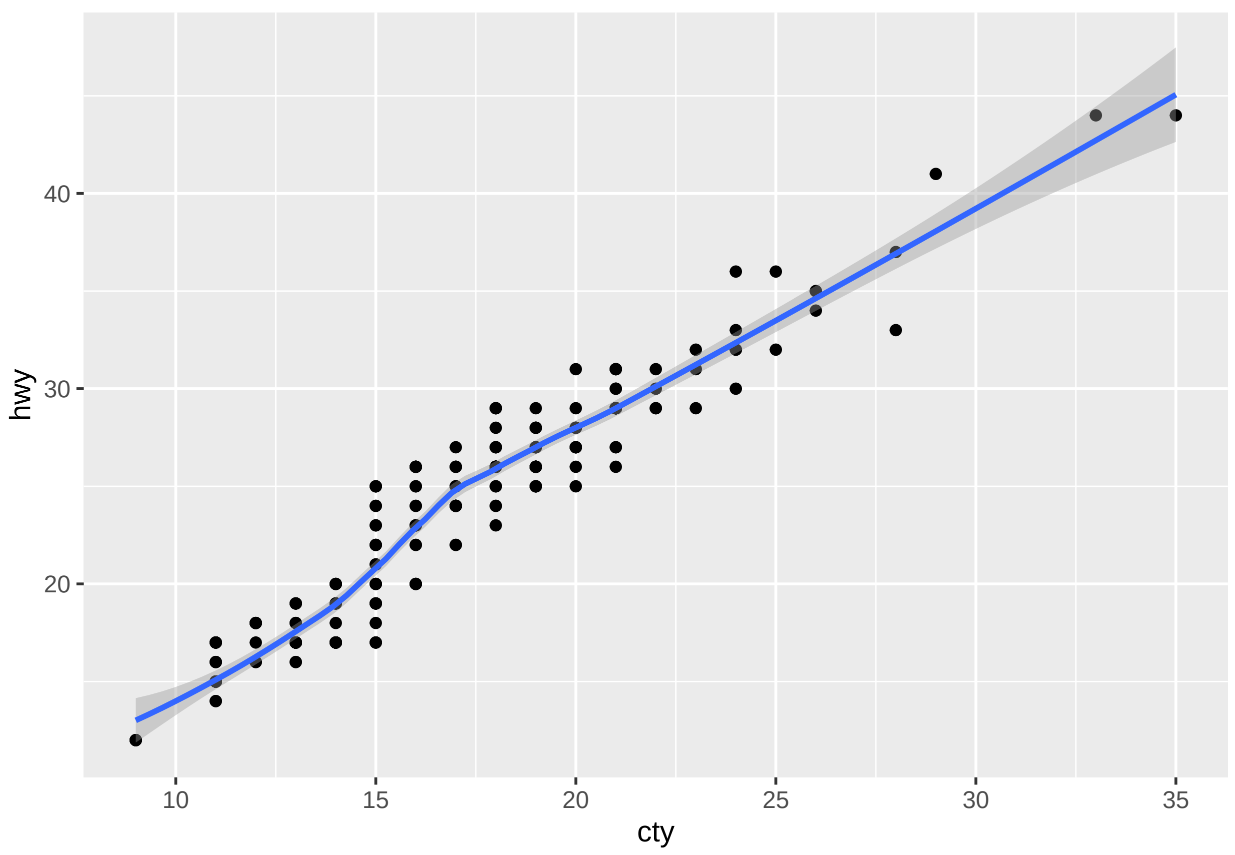
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



The code above can be simplified using the code below:

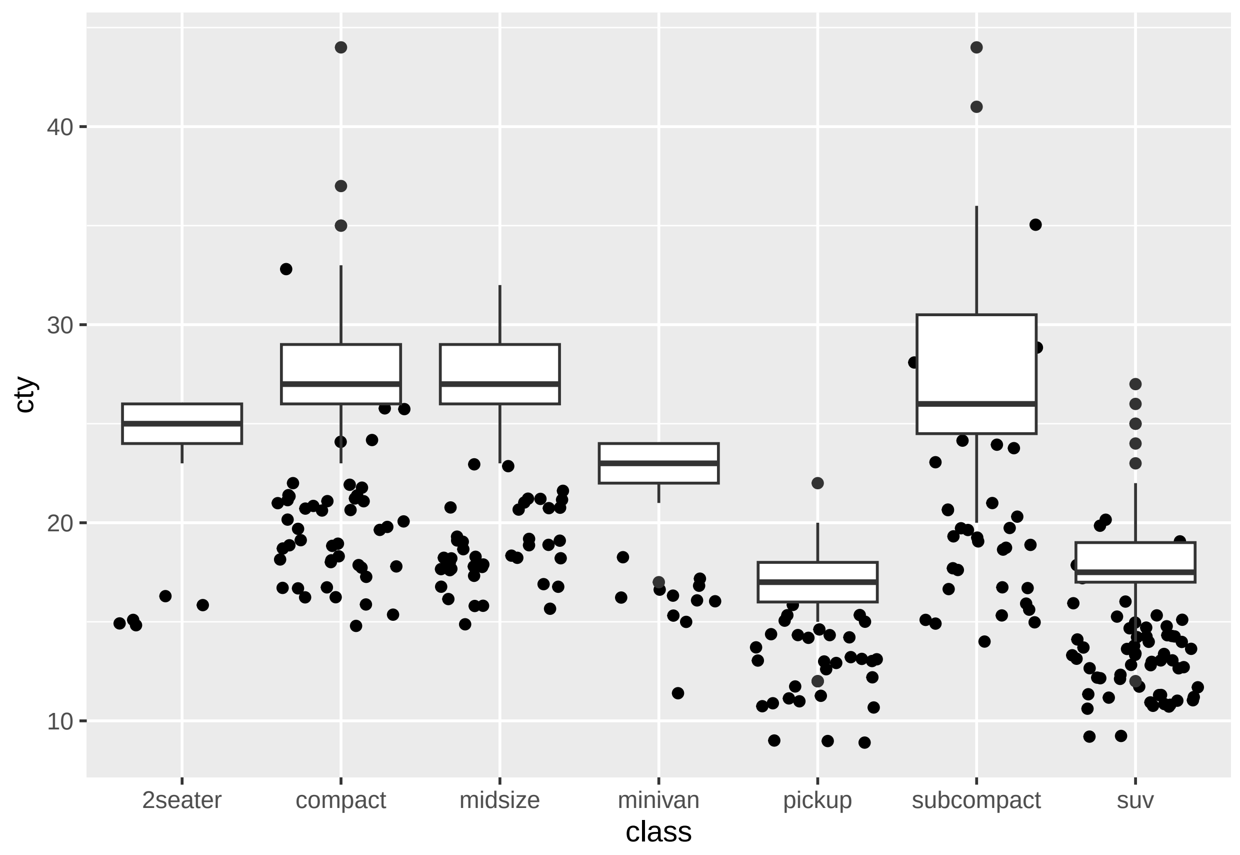
```
ggplot(data = mpg, aes(x = cty, y = hwy)) + geom_point() + geom_smooth()
```

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



4. What does the following code do? Does it work? Does it make any sense? Why/why not?

```r
ggplot(mpg) +
  geom_point(aes(class, cty), position = "jitter") +
  geom_boxplot(aes(class, hwy))
```



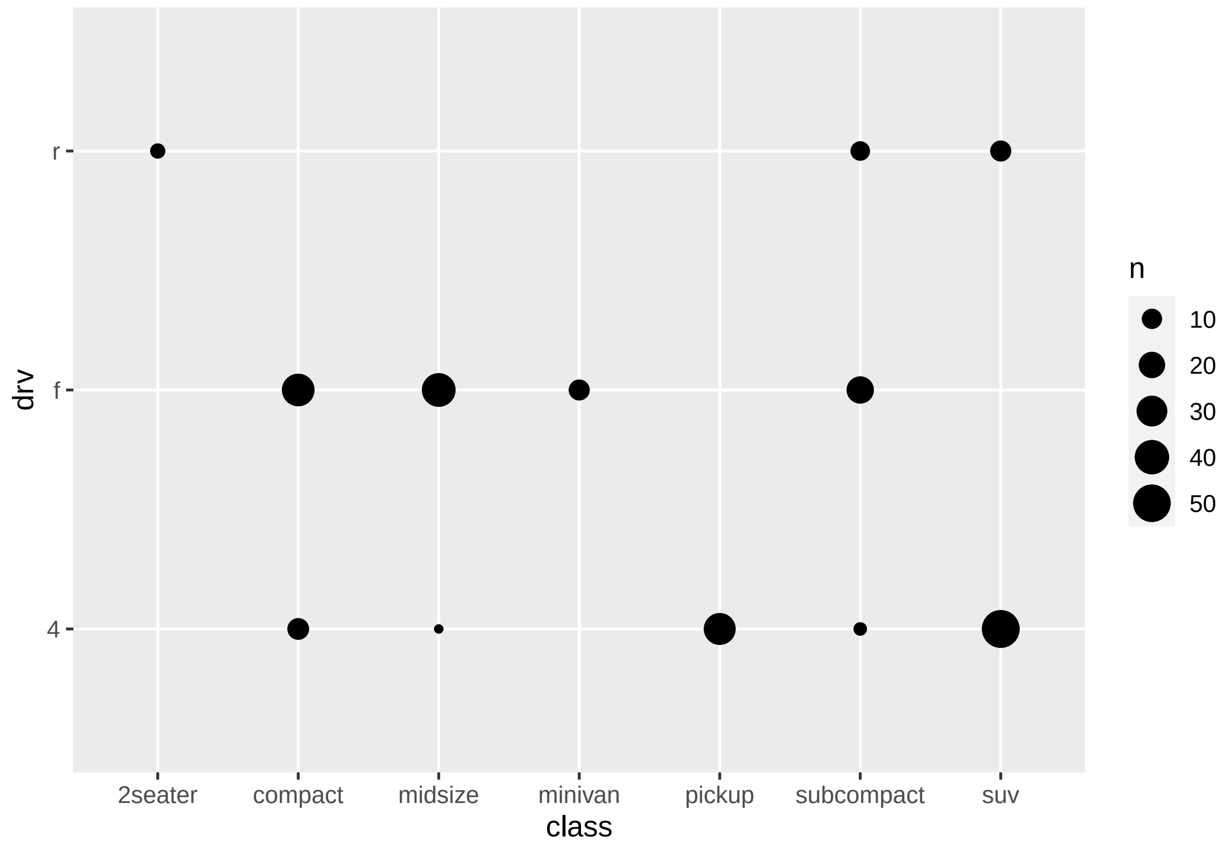
The plot produced above is made of 2 layers: - the first layer is a scatterplot which illustrates the relationship between class and cty. The data points for each variable under class have been "jittered" so that is easier to distinguish the cty value of each individual point.
- the second layer is a boxplot illustrating the relationship between class and hwy variables. For each class, a boxplot is drawn to show the distribution of the hwy values within each class.

By overlaying both plots against each other, however, the combined plot will not make sense because the vertical axis of the overall plot only indicates the cty variable and not the hwy variable (for the boxplot). Moreover, this will provide confusion for readers not familiar with ggplot2 as they may visually misinterpret that there are many outliers within the boxplot, which will not make sense theoretically.

5. Download the ggplot2 cheatsheet from https://www.rstudio.com/resources/cheatsheets/ so you have a handy visual reference for all the geoms. For each of the plots below, identify in words the geom(s) used to draw it.

Plot 1: geom_point() + scale_size()

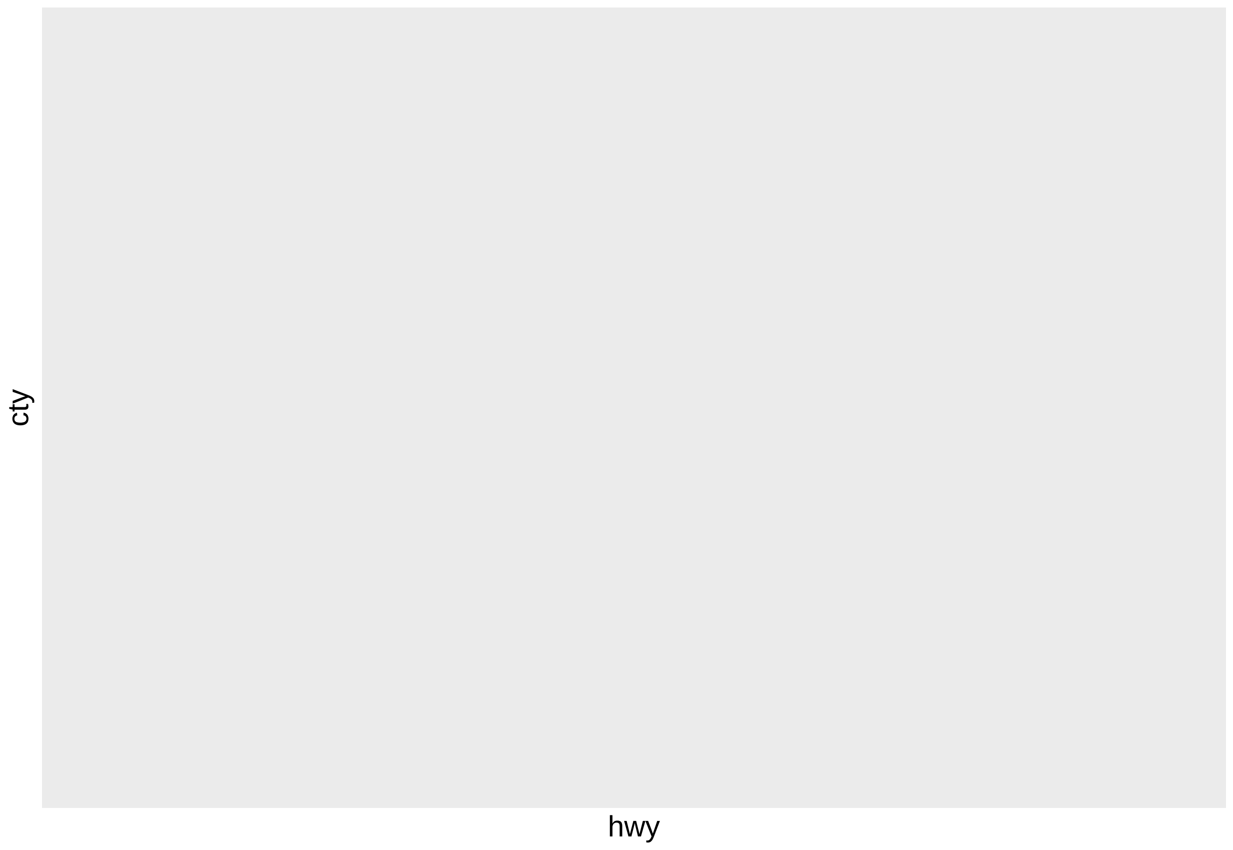
```
plot1 = mpg %>%
  count(class, drv)

ggplot(data = plot1, aes(x = class, y = drv, size = n)) +
  geom_point() + scale_size()
```



Plot 2: geom_hex() + scale_fill_gradient()

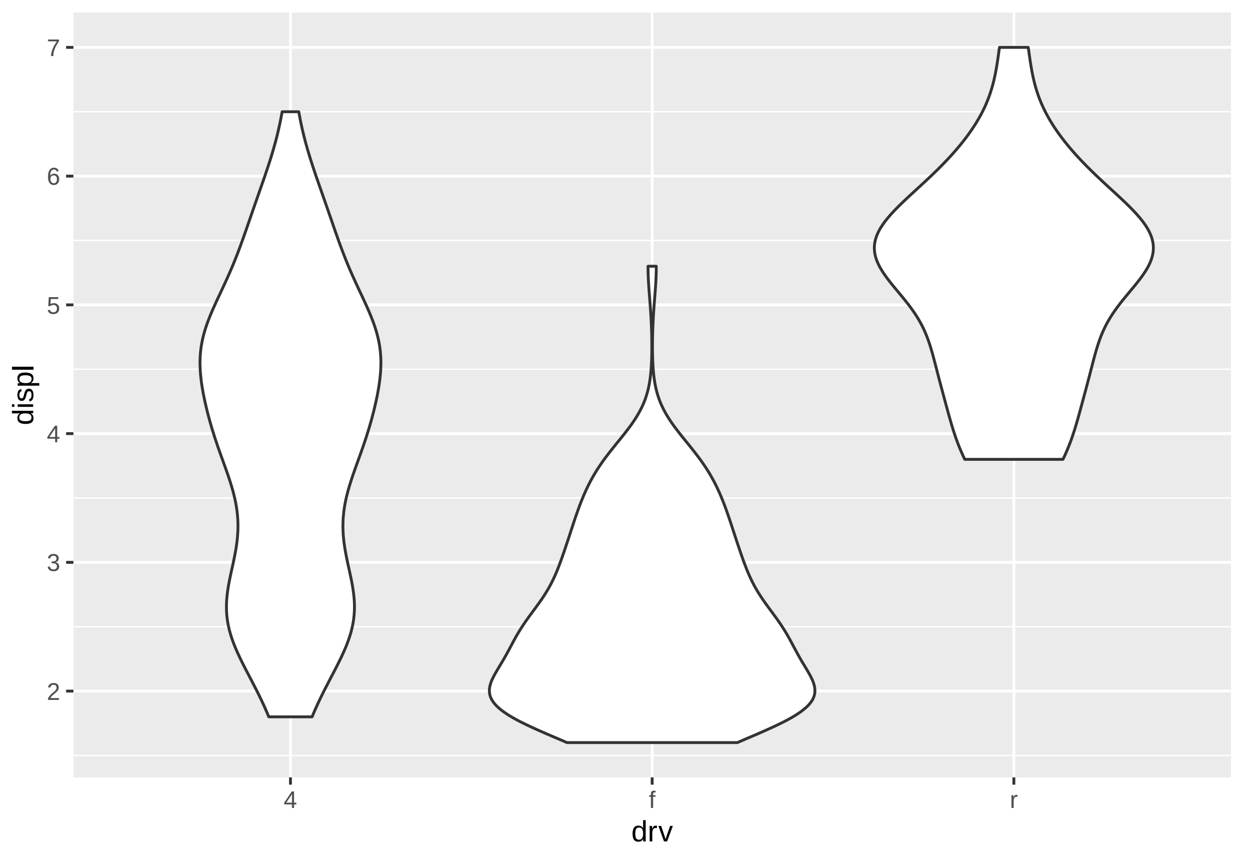
```
ggplot(data = mpg, aes(x = hwy, y = cty)) +
  geom_hex()+
  scale_fill_gradient()
```

```
## Warning: Computation failed in ‘stat_binhex()‘
## Caused by error in ‘compute_group()‘:
## ! The package "hexbin" is required for ‘stat_binhex()‘
```



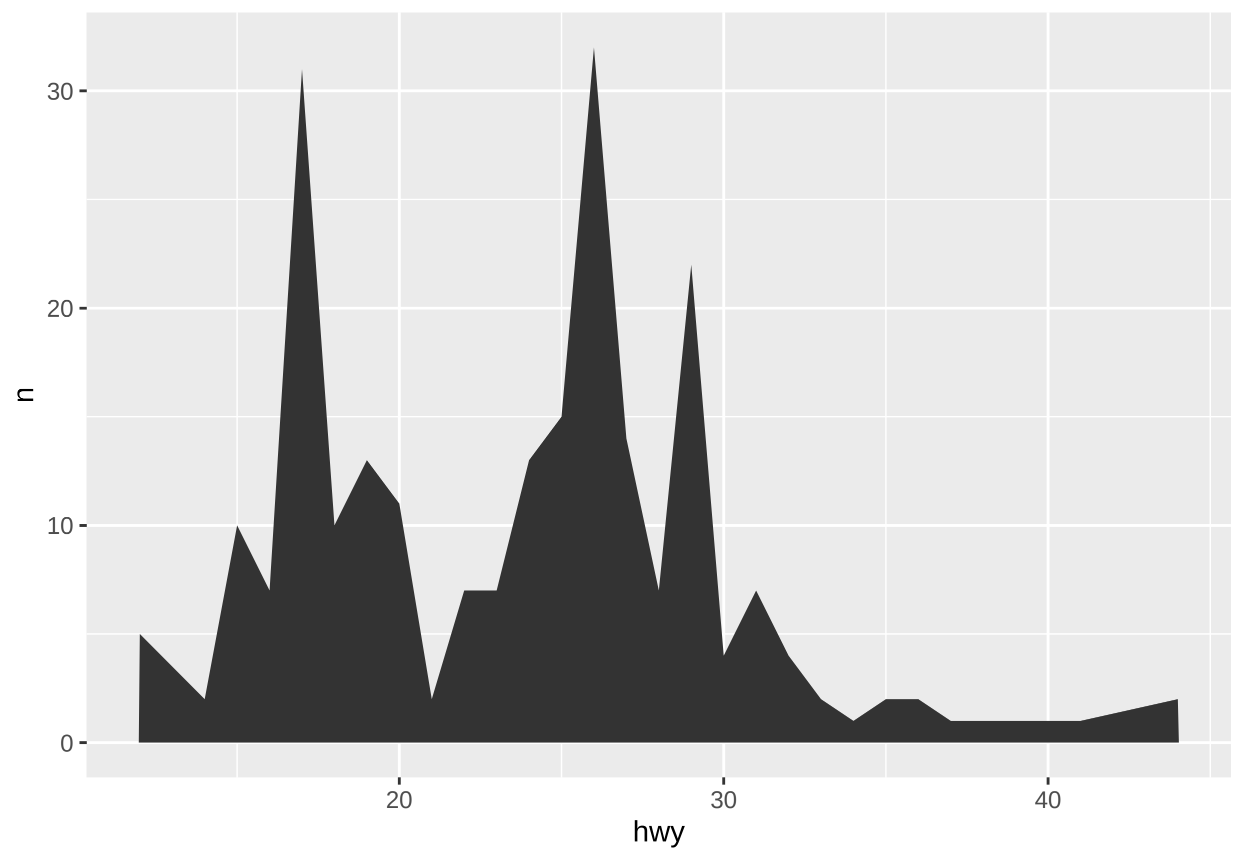
Plot 3: geom_violin()

```
ggplot(data = mpg, aes(x = drv, y = displ)) +
  geom_violin()
```



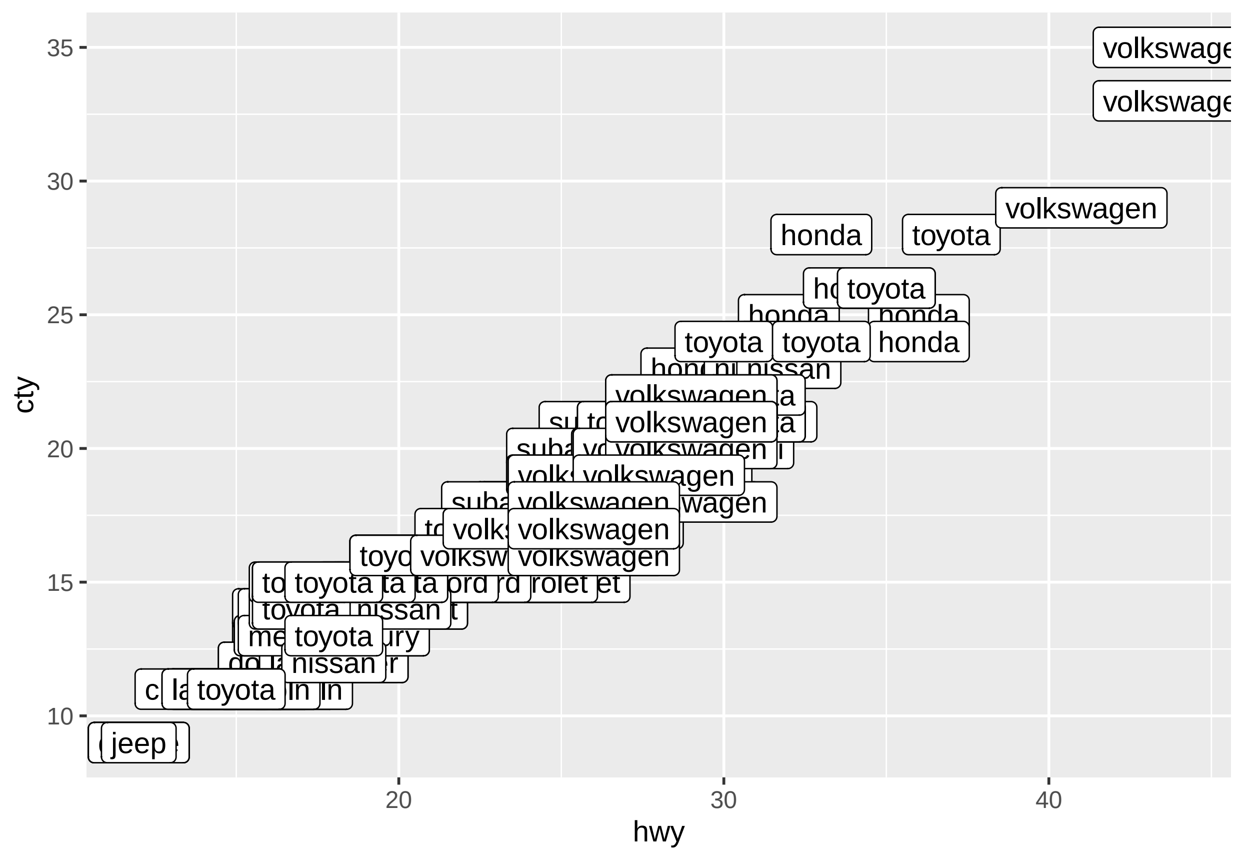
Plot 4: geom_area()

```
plot4 = mpg %>%
  count(hwy)

ggplot(data = plot4, aes(x = hwy, y = n))+
  geom_area()
```



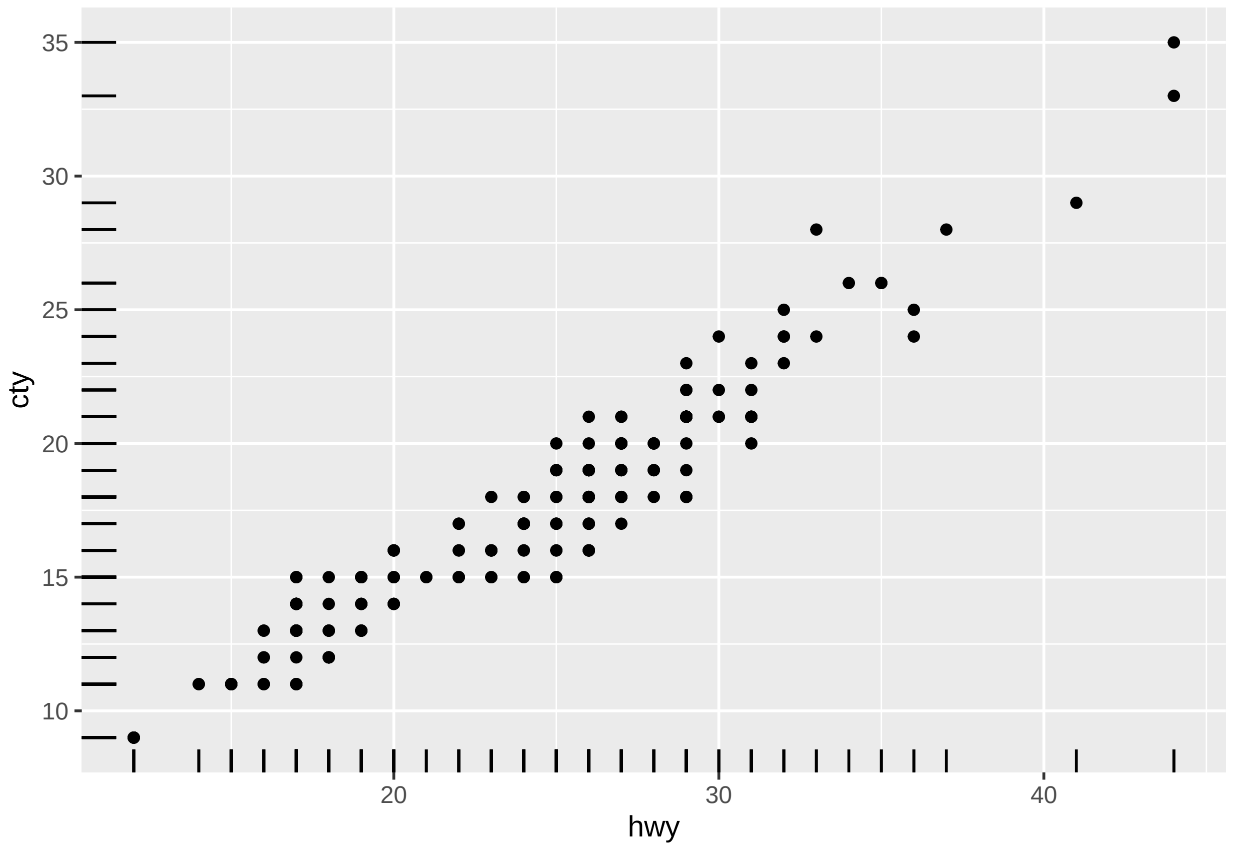
Plot 5: geom_label()

```
ggplot(data = mpg, aes(x = hwy, y = cty)) +
  geom_label(aes(label = manufacturer))
```



Plot 6: geom_rug() + geom_point()

```
ggplot(data = mpg, aes(x = hwy, y = cty)) +
  geom_rug()+
  geom_point()
```



Question 2. Revisit nycflights13 using ggplot2

In this question, we revisit flights dataset from the nycflights13 package. Recall that it contains all flights that departed NYC (from three airports, JFK, LGA, and EWR) in 2013.

1. Explain (in words) what the following code does.

```
df = flights %>%
  filter(origin == "JFK") %>%
  group_by(tailnum) %>%
  summarize(count = n(),
            dist = mean(distance, na.rm = TRUE),
            delay = mean(arr_delay, na.rm = TRUE)) %>%
  filter(count > 20, dist < 2000)
```

The above code creates the df dataframe, which is a subset of the flights dataset with the following features for each observation: - tailnum - tail number of each flight, which serves as the primary key/unique identifier of the df dataframe
- count - number of observations which belongs to each tailnum
- dist - average distance travelled for each tailnum's flights
- delay - average arrival delay time for each tailnum's flights
Furthermore, there are additional filters applied such that only flights with a minimum of 20 observations and have an average distance traveled of less than 2000 miles will appear in the df dataframe.

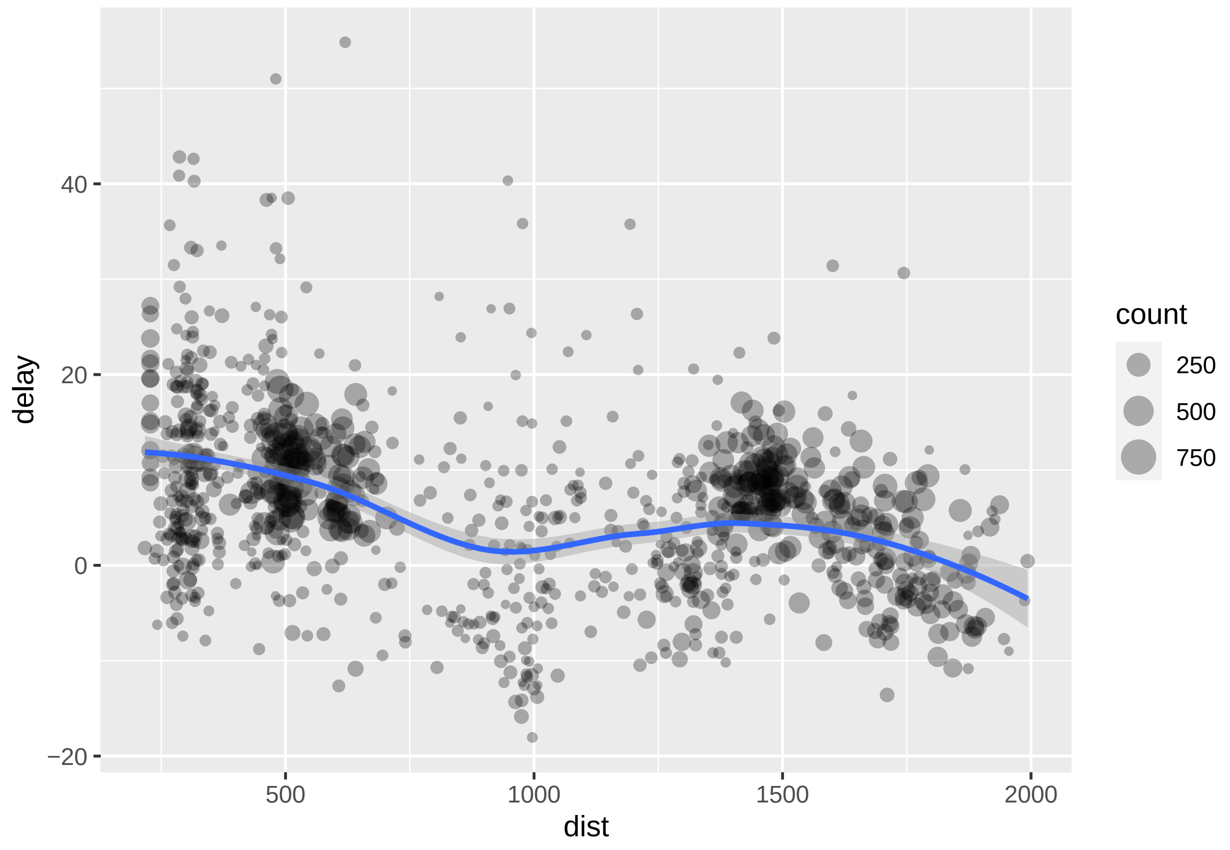
2. Use the df data frame in part 1 to recreate the following graph using ggplot2. Describes (in words) the trend(s) you observe from the graph.

```r
ggplot(data = df, aes(x = dist,y = delay)) +
  geom_point(aes(size = count), alpha = 0.3) +
  geom_smooth()
```

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

```
## Warning: Removed 1 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 1 rows containing missing values (`geom_point()`).
```



Question 3. HDB resale prices in Singapore

In this question, we want to replicate some findings from a Straits Times article on January 6, 2022, titled "Record 261 million-dollar HDB flats in 2021; resale prices rise in December as volume dips".

Download the raw data directly from the website: https://data.gov.sg/dataset/resale-flat-prices

After that, unzip the downloaded file and import the following CSV into R: resale-flat-prices-based-on-registration-date-from-jan-2017-onwards.csv

Use this dataset to verify the following statements from the ST article. In your answer, simply state whether each statement is true or false. If the statement is not correct, provide the right number from your analysis.

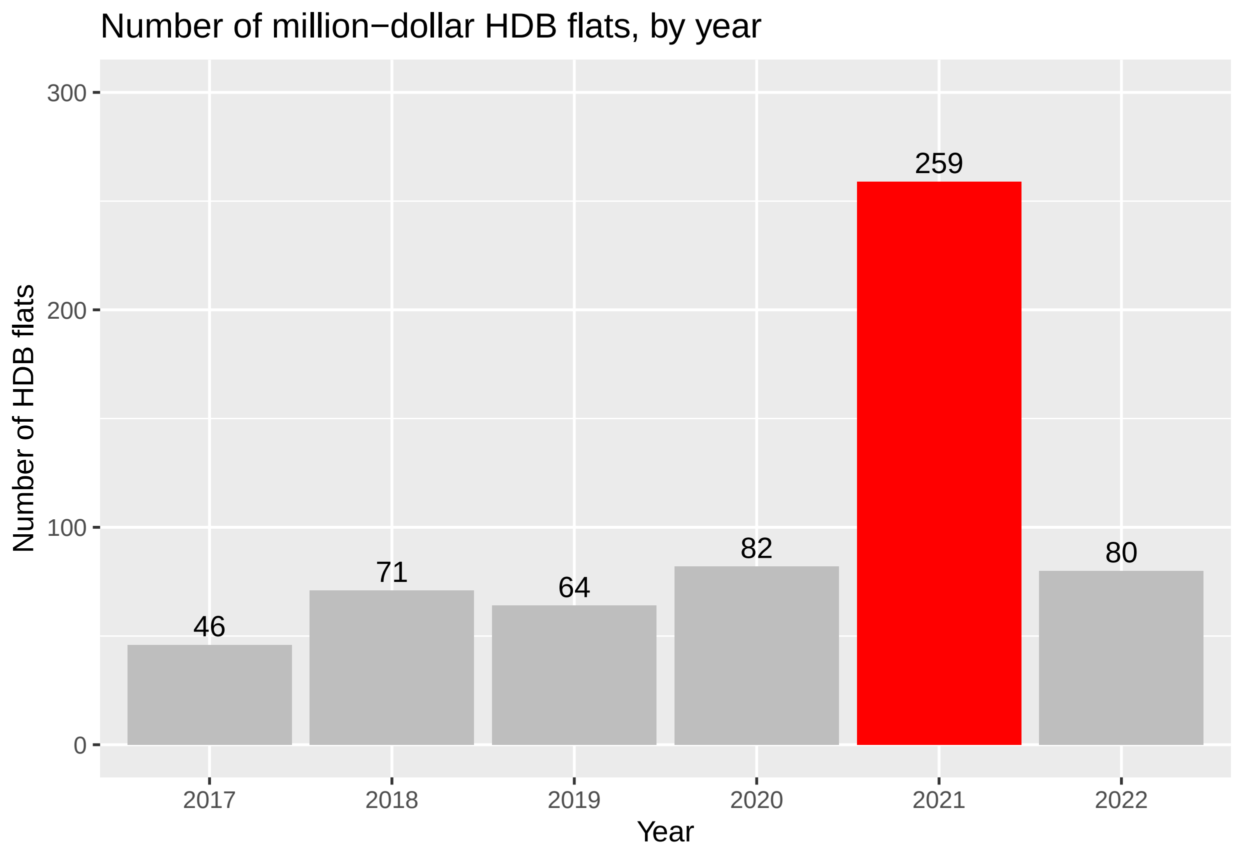
```r
#load dataset
resale = read_csv("resale-flat-prices-based-on-registration-date-from-jan-2017-onwards.csv")
```

```
## Rows: 123053 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): month, town, flat_type, block, street_name, storey_range, flat_mode...
## dbl (3): floor_area_sqm, lease_commence_date, resale_price
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

1. The total number of million-dollar HDB flats was 261 for the whole year of 2021.

```r
q1 = resale %>%
  mutate(year = substr(month, 1, 4))

q1a = q1 %>%
  filter(resale_price>=1000000) %>%
  group_by(year) %>%
  summarise(count = n())
```

```r
ggplot(data = q1a, aes(x = year, y = count, fill = year)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  scale_fill_manual(values = c("grey", "grey", "grey", "grey", "red", "grey")) +
  geom_text(aes(label = count), vjust = -0.4) +
  ylim(0,300) +
  ggtitle("Number of million-dollar HDB flats, by year") +
  xlab("Year") + ylab("Number of HDB flats")
```

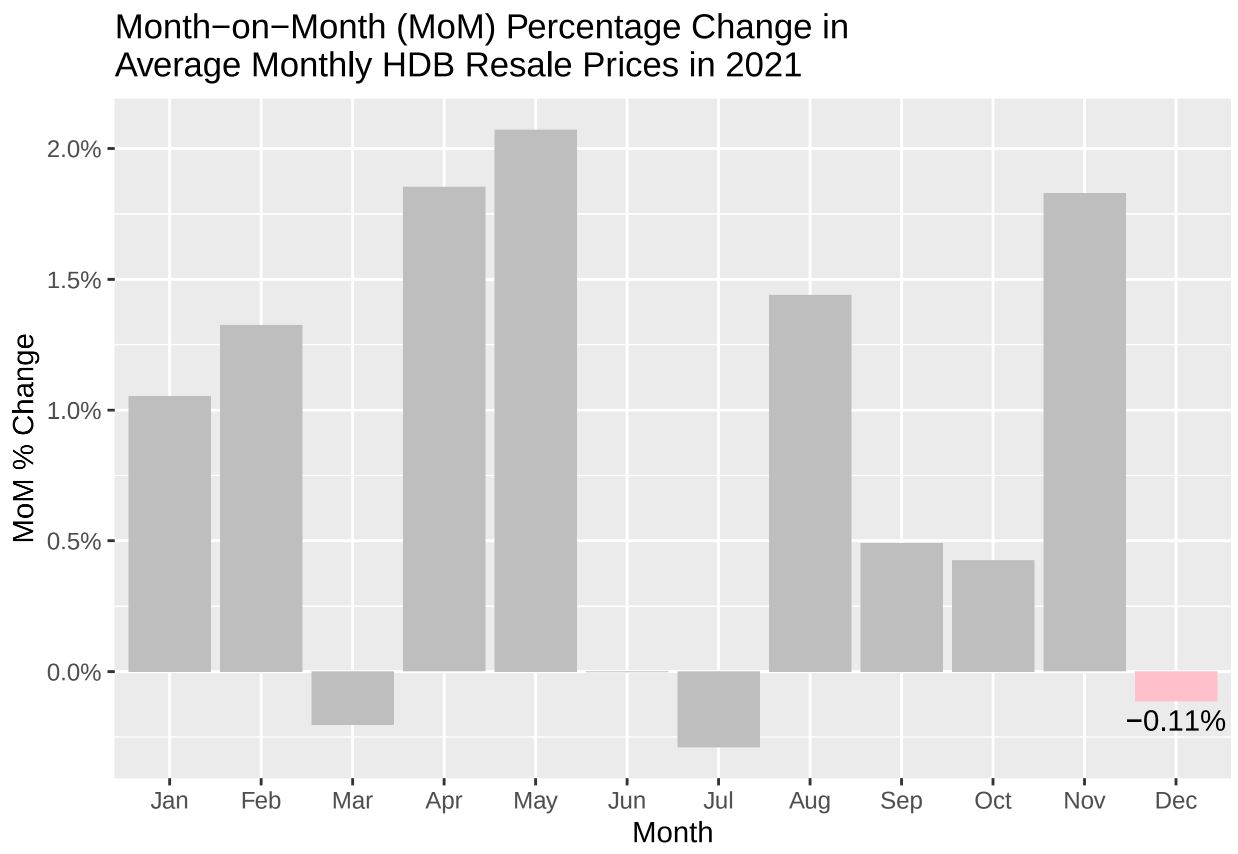


Based on the plot above, the above statement is false. In 2021, there were only 259 million-dollar (resale_value >= 1000000) HDB flats.

2. HDB resale prices rose 0.8 per cent in December 2021 from the previous month.

```
options(scipen = 999) #turn off scientific notation

q2 = resale %>%
  group_by(month) %>%
  summarise(avg_month_price = mean(resale_price)) %>%
  mutate(mom_per_change = (avg_month_price - lag(avg_month_price))/lag(avg_month_price)) #column for Mo

q2a = q2[grep("2021", q2$month),] #subset data to include observations for year 2021 only

ggplot(data = q2a, aes(x = month, y = mom_per_change, fill = month)) +
  geom_col(show.legend = FALSE) +
  scale_y_continuous(labels = scales::percent) +
  geom_text(data = subset(q2a, month == "2021-12"),
            aes(label = scales::percent(mom_per_change, accuracy = 0.01L)),
            vjust = 1.4)+
  scale_x_discrete(labels = c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "No
  scale_fill_manual(values = c("grey", "grey", "grey", "grey", "grey", "grey", "grey", "grey", "grey",
  ggtitle("Month-on-Month (MoM) Percentage Change in \nAverage Monthly HDB Resale Prices in 2021") +
```

```r
  xlab("Month") + ylab("MoM % Change")
```

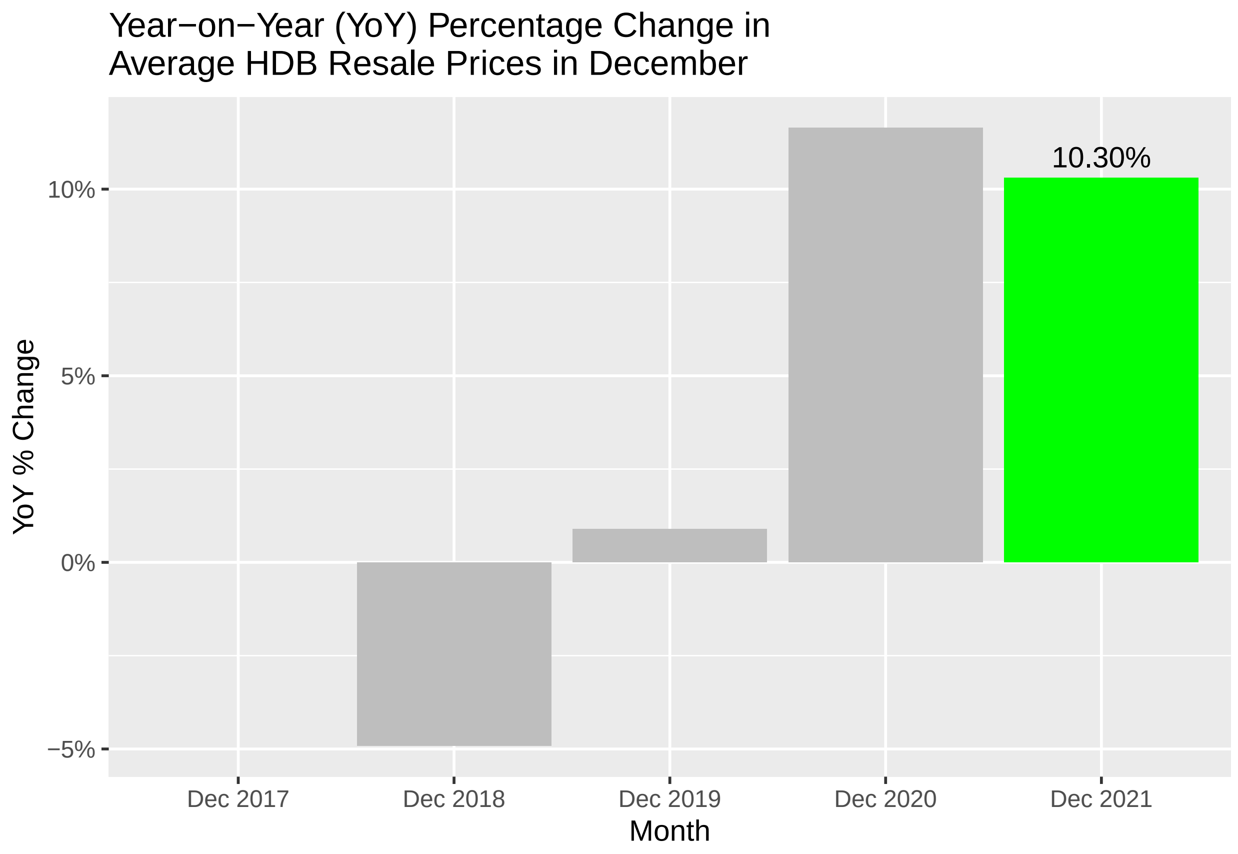


Based on the plot above, the statement is false. On average, HDB resale prices fell by 0.11% in December 2021 from the previous month.

3. HDB resale prices in December 2021 were 13.6 percent higher than a year ago.

```r
q3 = resale %>%
  group_by(month) %>%
  summarise(avg_month_price = mean(resale_price))

q3a = subset(q3, endsWith(month, "-12")) #subset q3 dataset using Dec values only across the years betw

q3b = q3a %>%
  mutate(yoy_per_change = (avg_month_price - lag(avg_month_price))/lag(avg_month_price)) #column for Yo

ggplot(data = q3b, aes(x = month, y = yoy_per_change, fill = month)) +
  geom_col(show.legend = FALSE) +
  scale_y_continuous(labels = scales::percent) +
  geom_text(data = subset(q3b, month == "2021-12"),
            aes(label = scales::percent(yoy_per_change, accuracy = 0.01L)),
            vjust = -0.5)+
  scale_fill_manual(values = c( "grey", "grey", "grey", "green")) +
  scale_x_discrete(labels = c("Dec 2017", "Dec 2018", "Dec 2019", "Dec 2020", "Dec 2021")) +
  xlab("Month") + ylab("YoY % Change") +
  ggtitle("Year-on-Year (YoY) Percentage Change in \nAverage HDB Resale Prices in December")
```

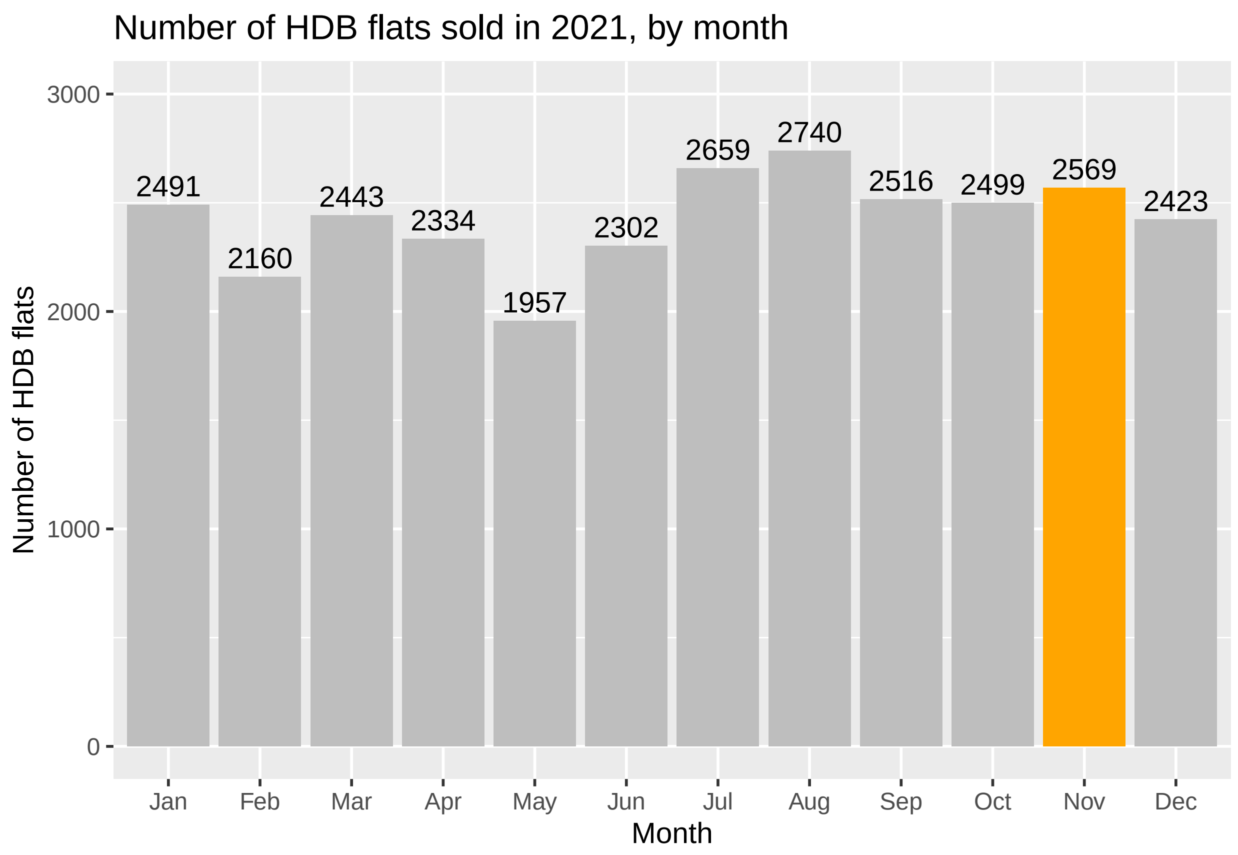
```
## Warning: Removed 1 rows containing missing values (‘position_stack()‘).
```



Based on the plot above, the statement is false. On average, HDB resale prices only increased by 10.3% in December 2021, compared to the previous year (December 2020).

4. There were 2,587 HDB flats sold in November 2021.

```
q4 = subset(q1, year == "2021") #df for transactions in 2021 only

q4a = q4 %>%
  count(month) #count based on month

ggplot(data = q4a, aes(x = month, y = n, fill = month)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  scale_fill_manual(values = c("grey", "grey", "grey", "grey", "grey", "grey", "grey", "grey", "grey",
  geom_text(aes(label = n), vjust = -0.4) +
  ylim(0,3000) +
  ggtitle("Number of HDB flats sold in 2021, by month") +
  xlab("Month") + ylab("Number of HDB flats") +
  scale_x_discrete(labels = c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "No
```



Based on the plot above, the statement is false. Only 2569 HDB flats were sold in November 2021.