# Chapter 02: First steps

Stan Piotrowski

September 14 2021

## Contents



### Setup

This chapter focuses on getting familiar with the basic recipes to create graphics using ggplot2.

### Fuel economy data exercises

1) *List five functions that you could use to get more information about the `mpg` dataset.*

If you wanted more general information about the `mpg` dataset (e.g., descriptions of the underlying data, or where to find more detailed information or the source), you could use `?`, `??`, or `help()`. If you wanted to get a quick summary of the data and see the distribution of each variable, you could use the `summary()` function.

There are a few additional functions that you could use, some of which are described at the following page: https://www.r-project.org/help.html. There is no such vignette for the `mpg` dataset, but `browseVignettes()` or `vignette()` can be used to find tutorials for selected packages. Additionally, if you really don't know the name of the package you are interested in and don't want to consult Google, you could use the `apropos()` function to identify the object or function in the R environment using regular expression pattern matching.

2) *How can you find out what other datasets are included with ggplot2?*

To find which datasets are included in ggplot2 (or any other package you are interested in), use the command `data(package = "<package_name>")`. For example, using `data(package = "ggplot2")`, we can see that there are datasets not only on fuel economy (`mpg`), but others on diamond prices and characteristics (`diamonds`), seal movements (`seals`), and housing sales in Texas (`txhousing`).

3) *Apart from the US, most countries use fuel consumption (fuel consumed over fixed distance) rather than fuel economy (distance traveled with a fixed amount of fuel). How could you convert the `cty` and `hwy` into the European standard of l/100km?*

To convert the fuel economy variables `cty` and `mpg` into fuel consumption by the European standard of l/100km, we can simply create a new variable `lpkm` by dividing the mpg estimate by the conversion factor 235.21.

4) *Which manufacturer has the most models in this dataset? Which model has the most variations? Does your answer change if you remove the redundant specification of drive train (e.g., "pathfinder 4wd", "a4 quattro") from the model name?*

Table 1: Fuel economy (distance traveled with one US gallon) and fuel consumption (liters consumed per 100 kiometers.

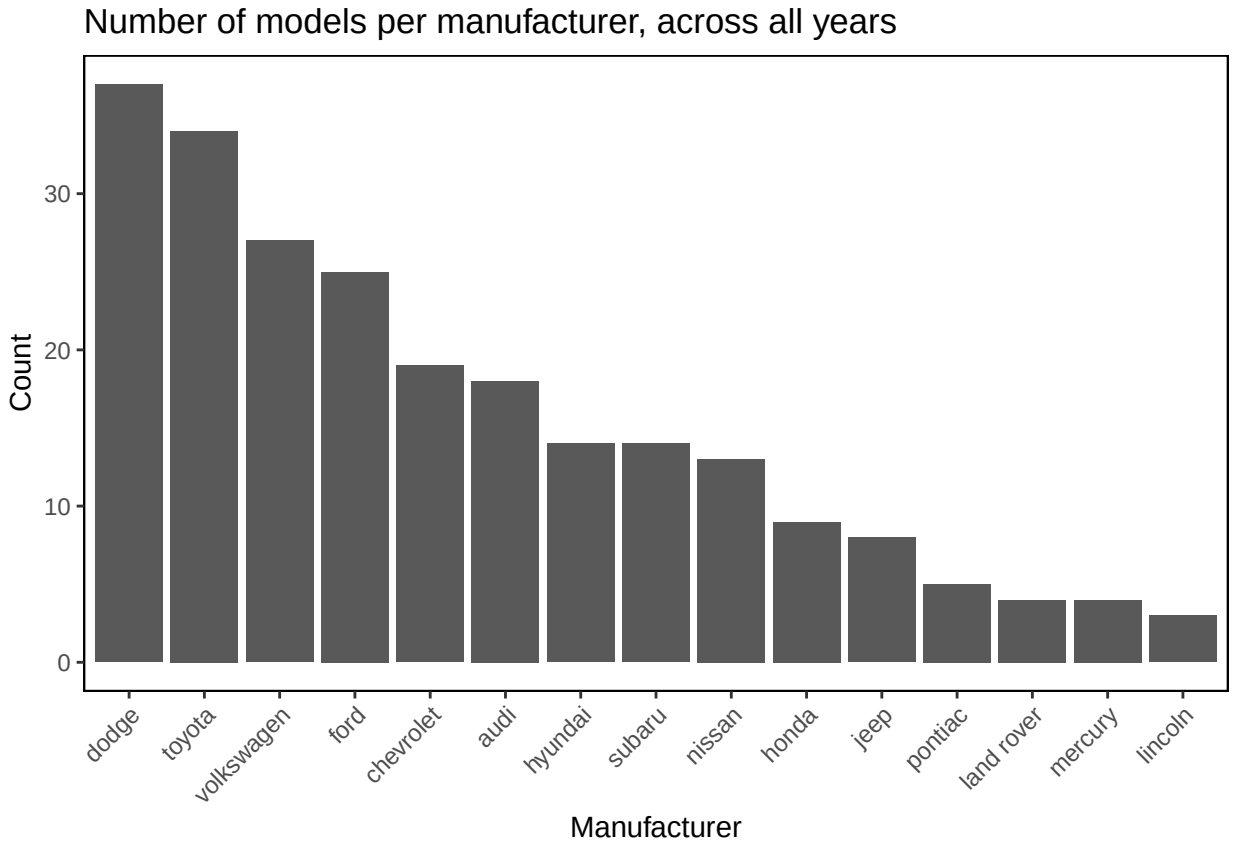
| manufacturer | model | year | cty | hwy | cty_lpkm | hwy_lpkm |
|---|---|---|---|---|---|---|
| audi | a4 | 1999 | 18 | 29 | 13.07 | 8.11 |
| audi | a4 | 1999 | 21 | 29 | 11.20 | 8.11 |
| audi | a4 | 2008 | 20 | 31 | 11.76 | 7.59 |
| audi | a4 | 2008 | 21 | 30 | 11.20 | 7.84 |
| audi | a4 | 1999 | 16 | 26 | 14.70 | 9.05 |
| audi | a4 | 1999 | 18 | 26 | 13.07 | 9.05 |

Dodge has the most models in the dataset (37; see visualization below), and the Dodge caravan 2wd has the most variations (11). Interestingly, the number doesn't change if we remove the redundant information (e.g., "quattro" or "4wd"), likely because manufacturers don't want to use the same model name as competitors.

```
## # A tibble: 1 x 2
##   manufacturer     n
##   <chr>        <int>
## 1 dodge           37
```

```
## # A tibble: 1 x 2
##   model           n
##   <chr>       <int>
## 1 caravan 2wd    11
```

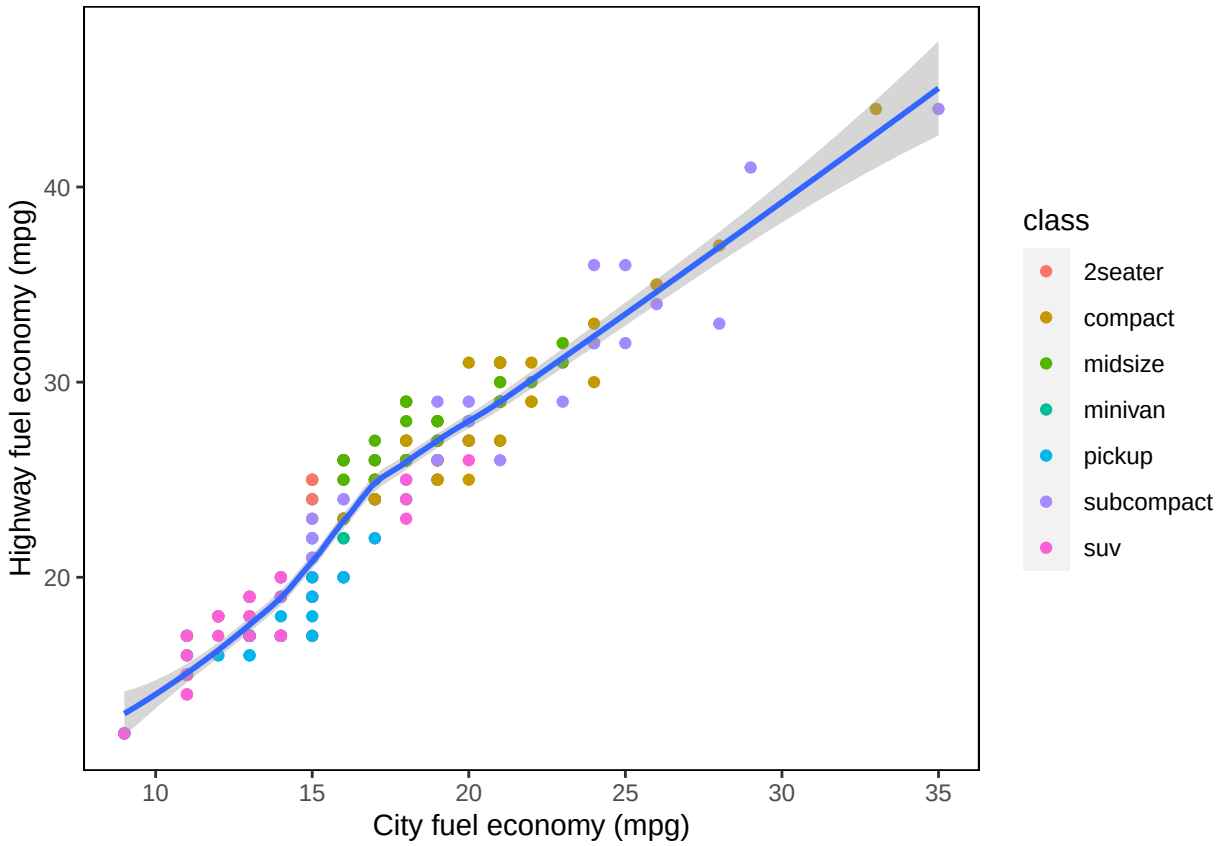
```
## # A tibble: 1 x 2
##   model          n
##   <chr>      <int>
## 1 "caravan "    11
```




## Key components exercises

1) *How would you describe the relationship between* `cty` *and* `hwy`*? Do you have any concerns about drawing conclusions for that plot?*

First, let's generate the plot of the two variables.



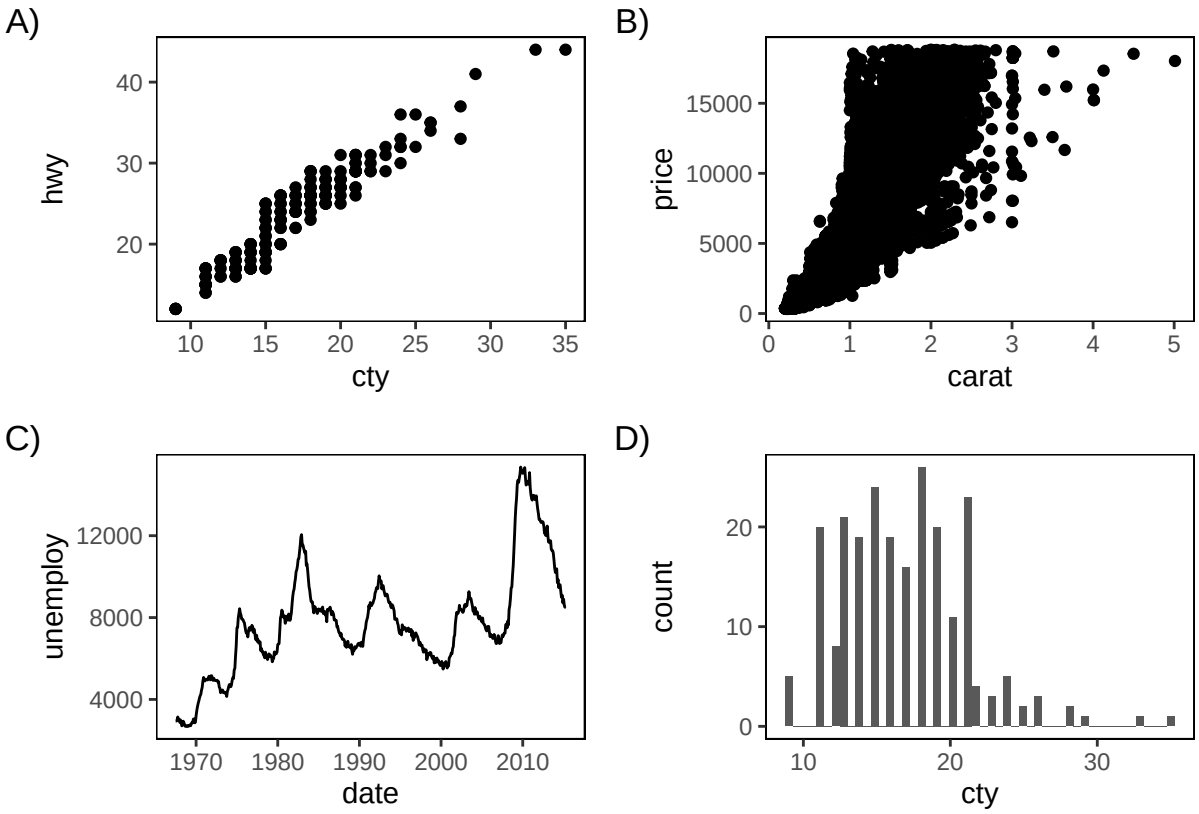

From the plot above, we can see that in general, the highway fuel economy increases as the city fuel economy increases. However, by mapping `class` to the color aesthetic, we can see that there are differences within each vehicle class that likely have to do with other features of the vehicles, like the year, perhaps.

2) *What does `ggplot(mpg, aes(model, manufacturer)) + geom_point()` show? Is it useful? How could you modify the data to make it more informative?*

In general, the plot produced from the code `ggplot(mpg, aes(model, manufacturer)) + geom_point()` is not very useful because it only shows us the manufacturer for each car model in the dataset in the same way we would view the data in a table. To modify the data and make it more informative for visualization, we could create a plot to show the total number of vehicles for each manufacturer using a bar chart, for example.

3) *Describe the data, aesthetic mappings and layers used for each of the following plots. You'll need to guess a little because you haven't see all the datasets and functions yet, but use your common sense! See if you can predict what the plot will look like before running the code.*

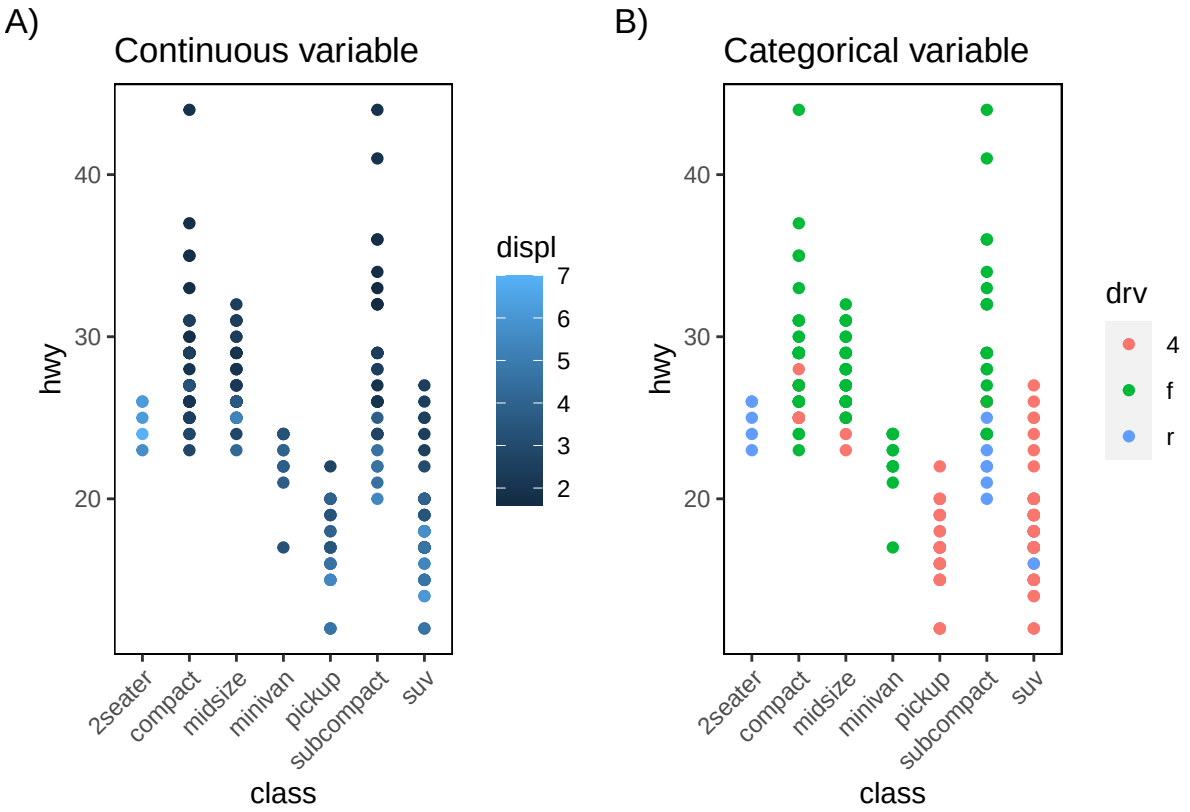
The first plot is identical to one of the plots we generated previously, except there is no variable mapped to the color aesthetic. The second plot shows the distribution diamond prices as a function of the number of carats. The third plot is a time series line plot of the number of unemployed Americans over time, starting in 1970 and extending until April 2015. Finally, the fourth plot shows the distribution of city fuel economy (mpg) for all vehicle models and manufacturers.



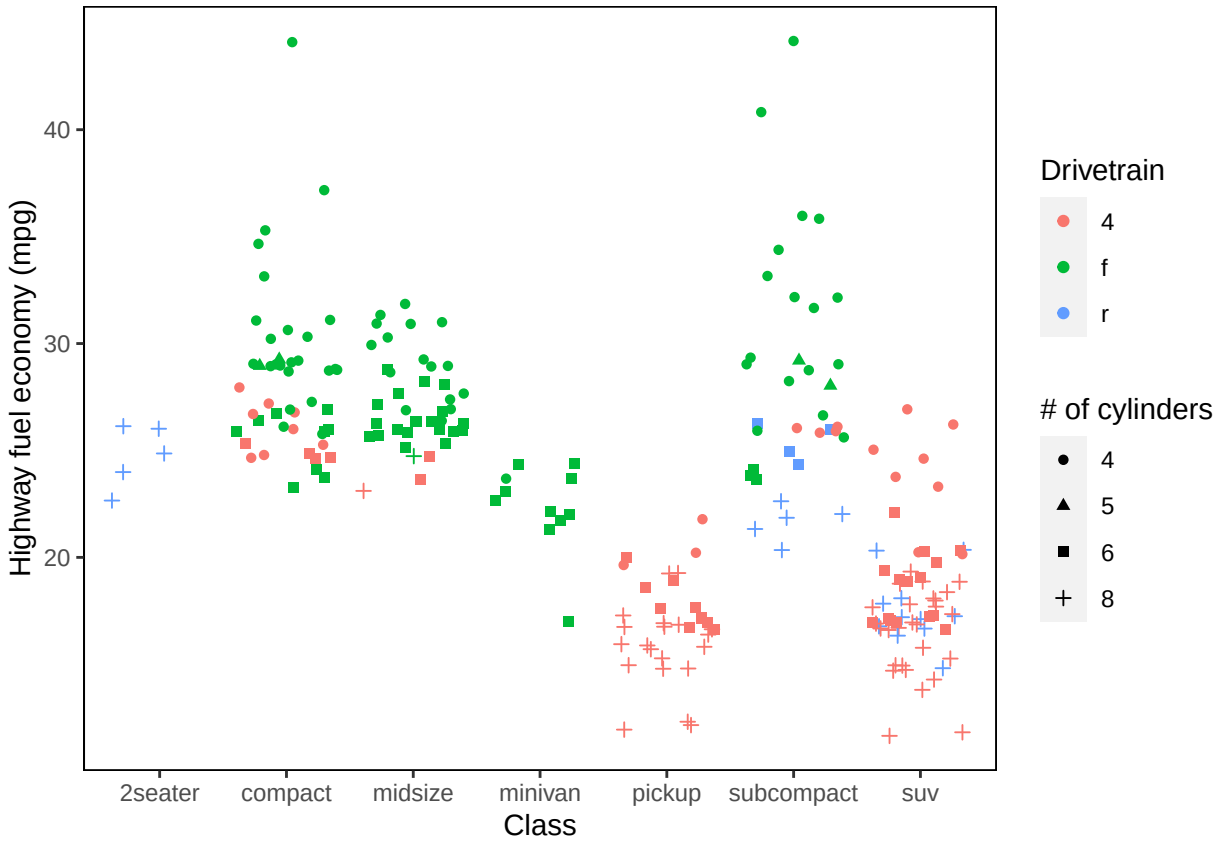

## Aesthetics exercises

1) *Experiment with the color, shape, and size aesthetics. What happens when you map them to continuous values? What about categorical values? What happens when you use more than one aesthetic in a plot?*

When you map a continuous variable to an aesthetic like color, for example, a continuous color scale is used. In contrast, when you map a categorical variable to the same aesthetic, the discrete color scale is used. To demonstrate, I've plotted the vehicle class (`class`) and the highway fuel economy (`hwy`) with the engine displacement (`displ`) variable mapped to the color aesthetic in one plot and the drive train (`drv`) mapped to the same aesthetic in another.




When you map more than one variable to an aesthetic in a plot, the legend populates to display how the aesthetic maps back to the original data. For example, we can map the `drv` variable to the color aesthetic and the number of cylinders (`cyl`) to the shape aesthetic to see the relationship between `class` and `hwy`.

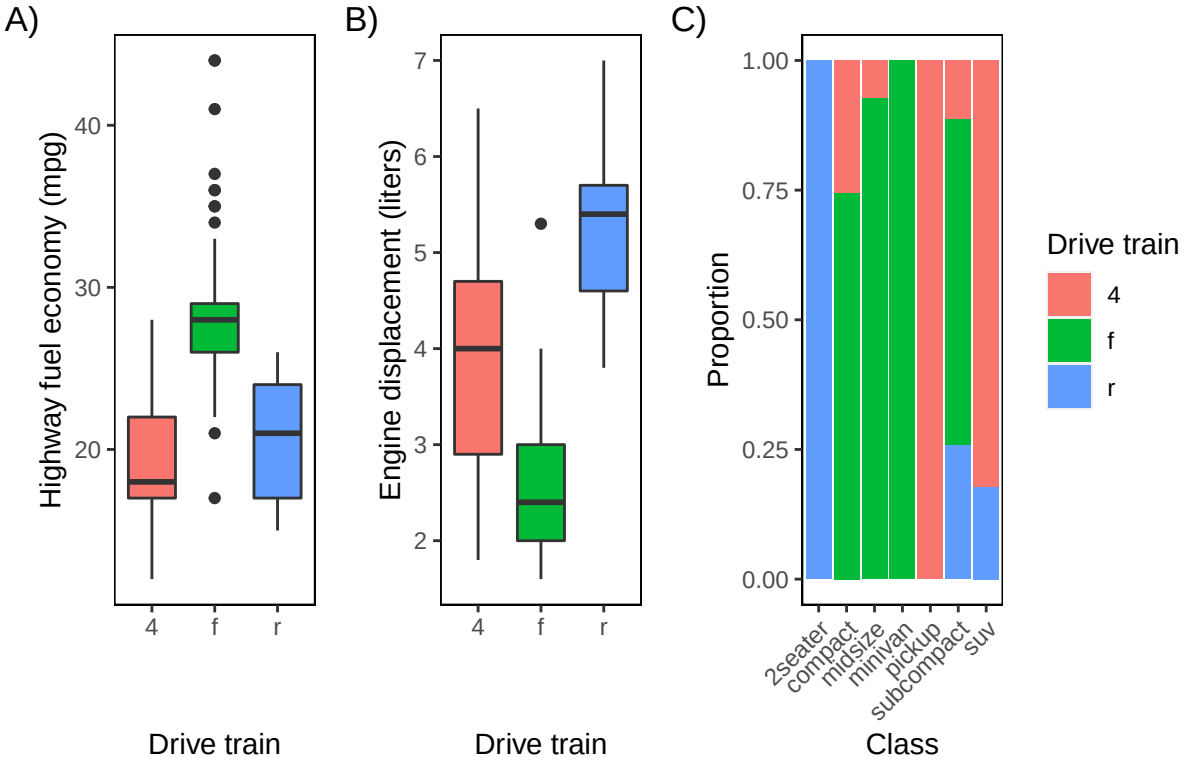



2) *What happens if you map a continuous variable to shape? Why? What happens if you map* `trans` *to shape? Why?*

If you try to map a continuous variable, like `displ`, to a shape, ggplot2 will throw an error because shapes are used for distinguishing distinct classes of categorical data. In other words, there is no continuous scale to apply to various shapes for continuous data. If you try to map `trans` to shape, technically the variable is categorical, but there are too many shapes to accurately discern (ggplot2 will only map data to 6 distinct shapes). When the plot becomes overly complicated, it becomes difficult to interpret and alternative strategies are needed.

3) *How is drive train related to fuel economy? How is drive train related to engine size and class?*

Relationship between drive train, fuel economy, engine size, and vehicle class

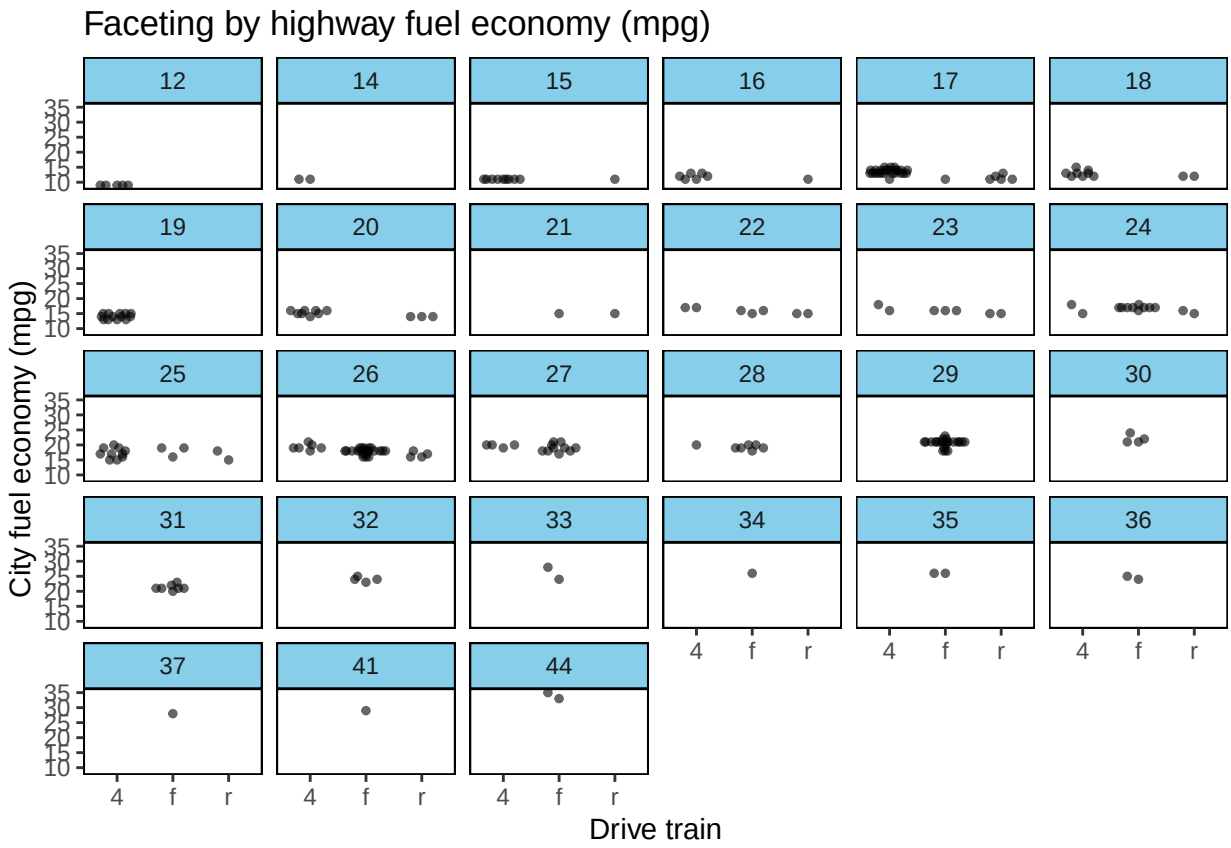



In plot A), we can see that in general, the front-wheel drive drive train has the best fuel economy, with four-wheel drive having the worst fuel economy. In plot B), we can see that rear-wheel drive vehciles have the largest engines in terms of displacement (liters), with front-wheel drive vehicles having the smallest engines. When we look at plot C), we can see the proportion of the vehicles in each class by drive train. For example, we can see that the 2seater vehicles (e.g. ,corvettes) all have rear-wheel drive trains and since they are more like sports cars, should have the largest engines. In another example, we can see that the pickup trucks in plot C) all have four-wheel drive, and we know that pickups aren't known for exceptional fuel economy. Low and behold, in plot A) we can see that the four-wheel drive vehicles have the worst highway fuel economy overall (although it's worth noting that a large proportion of SUV's are also four-wheel drive).

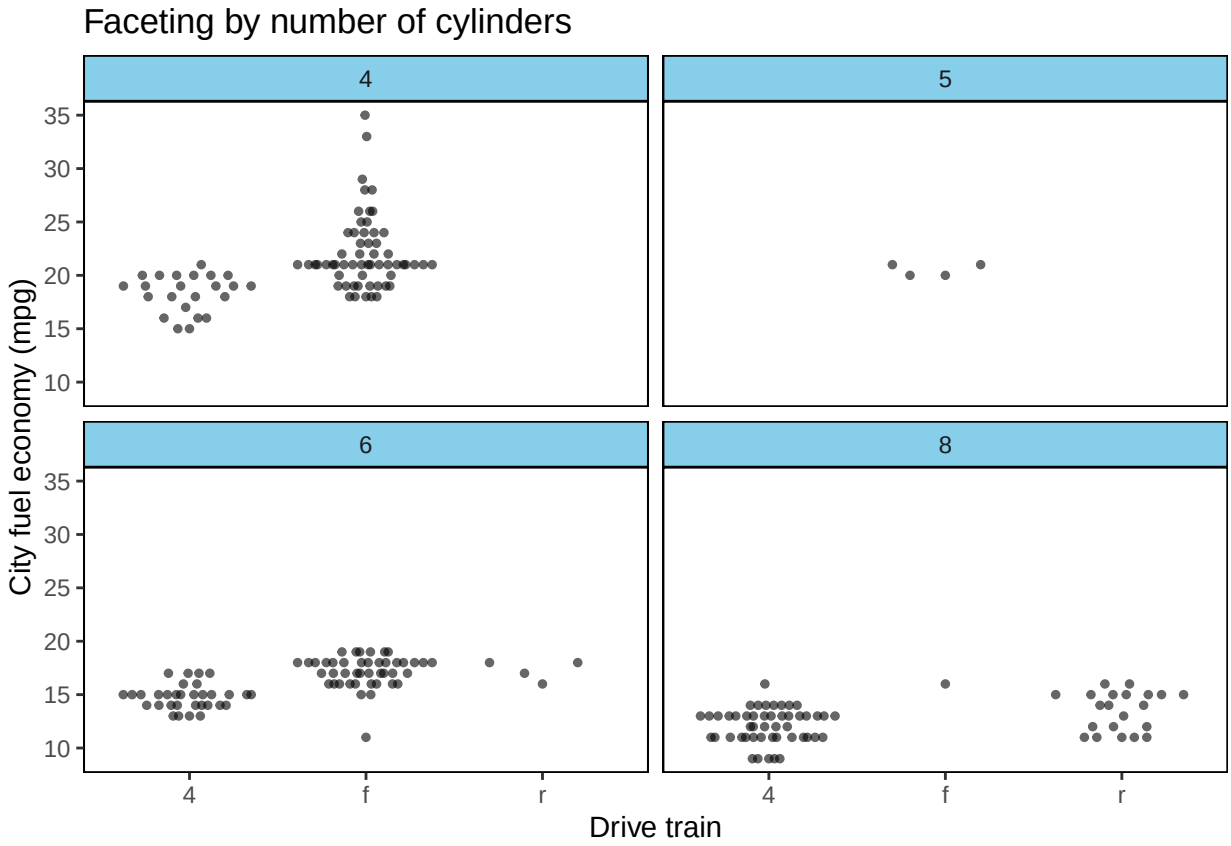
## Faceting exercises

1) *What happens if you try to facet by a continuous variable like* `hwy`*? What about* `cyl`*? What's the key difference?*

First, let's create a series of plots faceted by the continuous variable `hwy`.



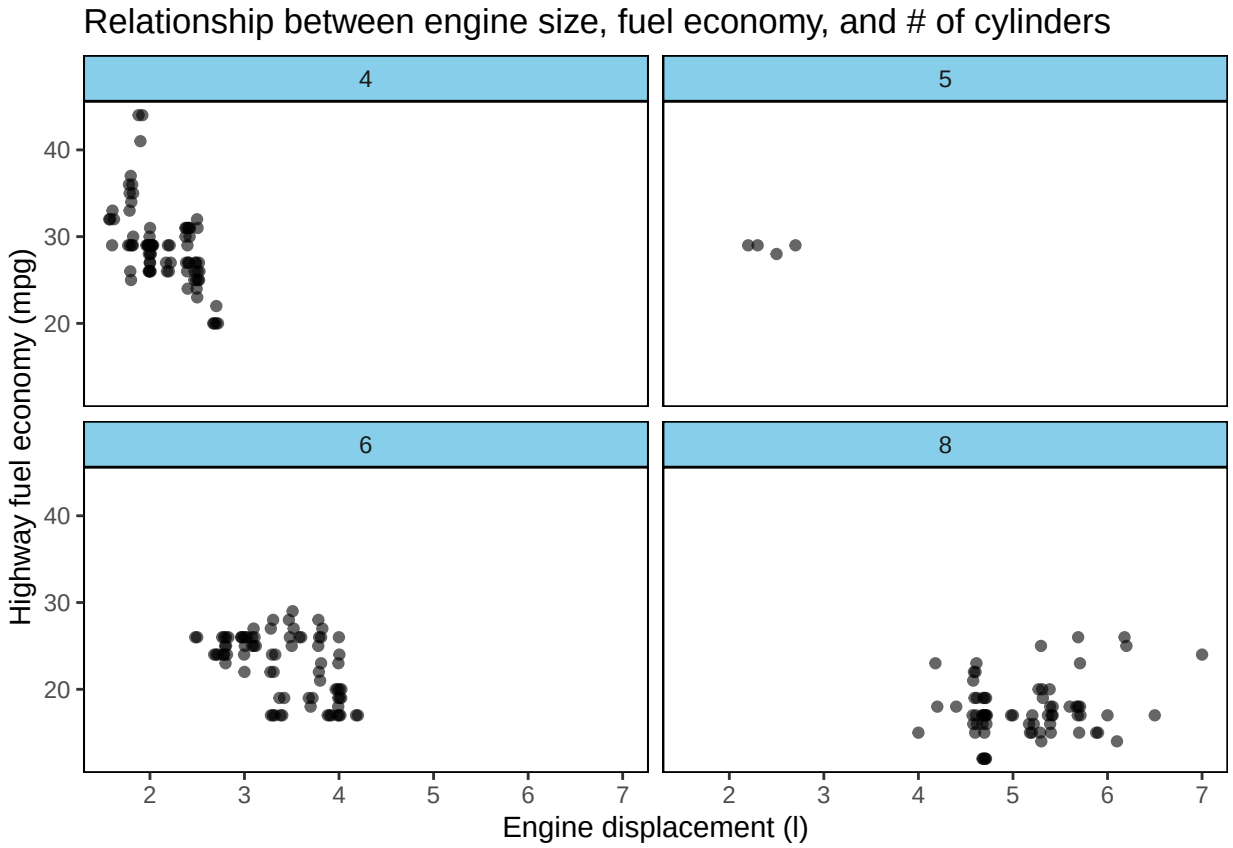

Next, we'll facet by another continuous variable, `cyl`, and compare the differences.




The main difference between the two plots are the number of values in each continuous variable to facet on. For example, if we look at the first plot faceted by highway fuel economy, we can see that faceting creates a plot for each value, from 12-44 mpg. In contrast, there are fewer values for the number of cylinders (4, 5, 6, and 8), so faceting on this particular continuous variable is much more informative and easier to interpret.

2) *Use faceting to explore the 3-way relationship between fuel economy, engine size, and number of cylinders. How does faceting by the number of cylinders change your assessment of the relationship between engine size and fuel economy?*

Let's create a series of plots looking at the engine size and highway fuel economy, faceted by the number of cylinders.



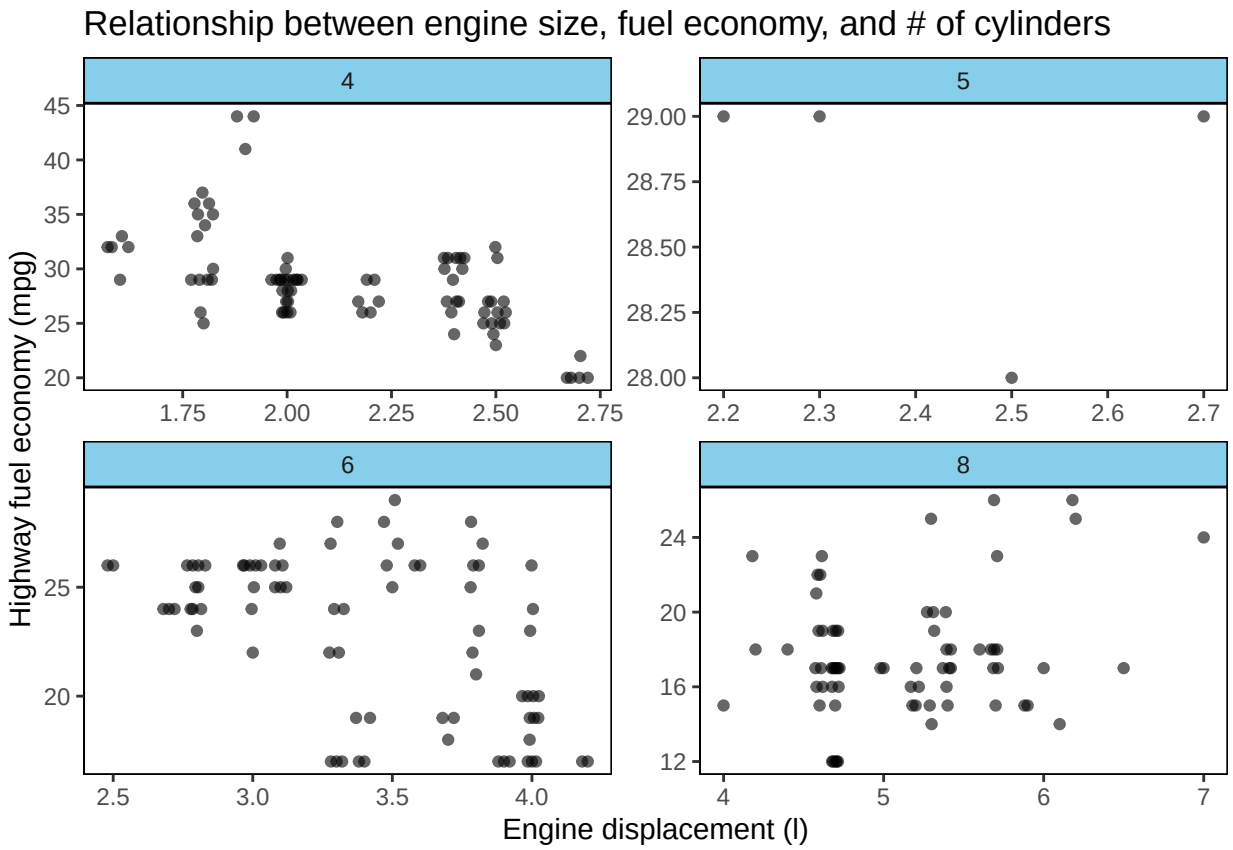

Creating a series of plots like this faceted on the engine size (displacement, in liters; `displ`) shows how in general, smaller engines with fewer cylinders have higher highway fuel economy compared to the largest engines with more cylinders. This is rather intuitive, as we would expect smaller engines to have fewer cylinders and larger engines to have more cylinders.

3) *Read the documentation for `facet_wrap()`. What arguments can you use to control how many rows and columns appear in the output?*

The arguments `nrow` and `ncol` in `facet_wrap()` control the number of rows and columns that appear in the output, respectively.

4) *What does the `scales` argument to facet_wrap()‘ do? When might you use it?*

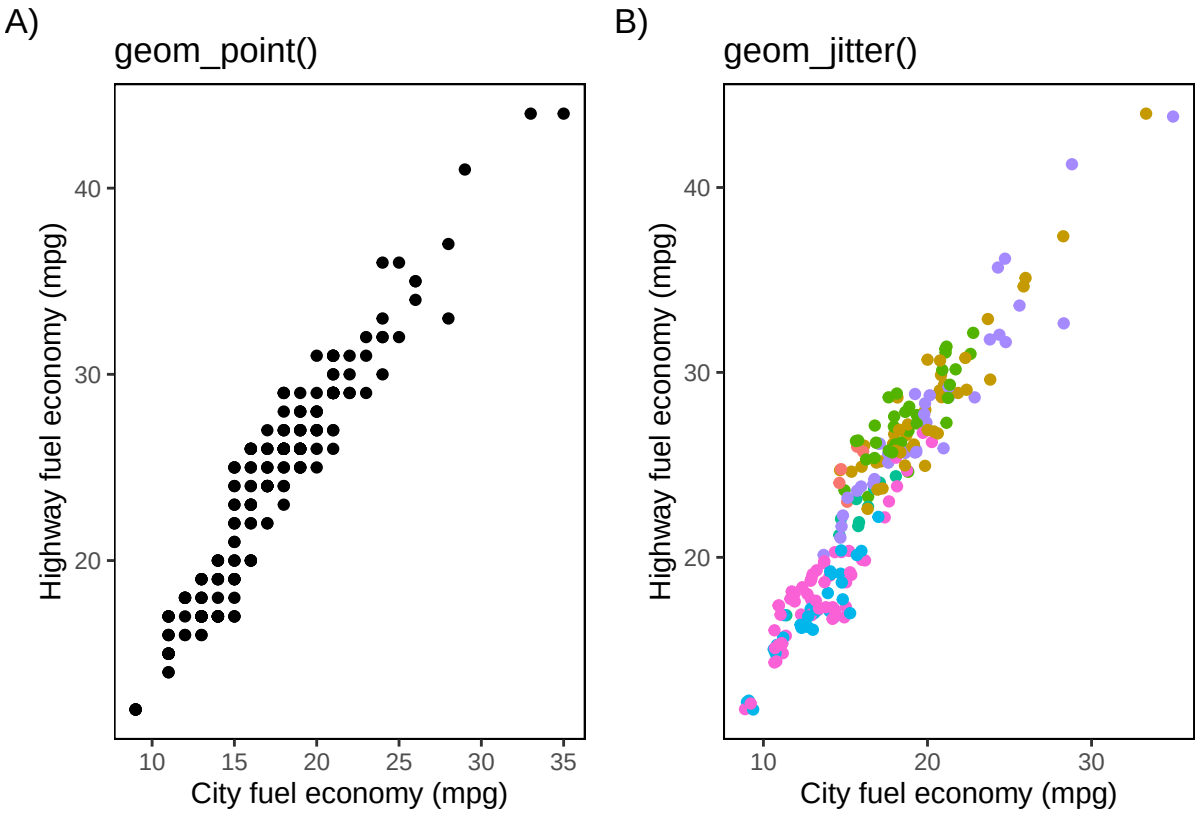
The `scales` argument in `facet_wrap()` controls whether or not the x- and y-axes in the faceted plots should all share the same scale (`scales = "fixed"`, the default), or be free to vary in two dimensions (`scales = "free"`), or a single dimension (e.g., `scales = "free_x"`). When faceting by categorical variables where the distribution of values of the continuous variables may be wildly different it may be a wise choice to allow the scales to be free in the faceted plots. For example, we can allow both the x- and y-axes of the plot above to have free scales and compare the outputs.




## Plot geoms exercises

1) *What's the problem with the plot created by 'ggplot(mpg, aes(cty, hwy)) + geom_point()? Which of the geoms described above is most effective at remedying the problem?*

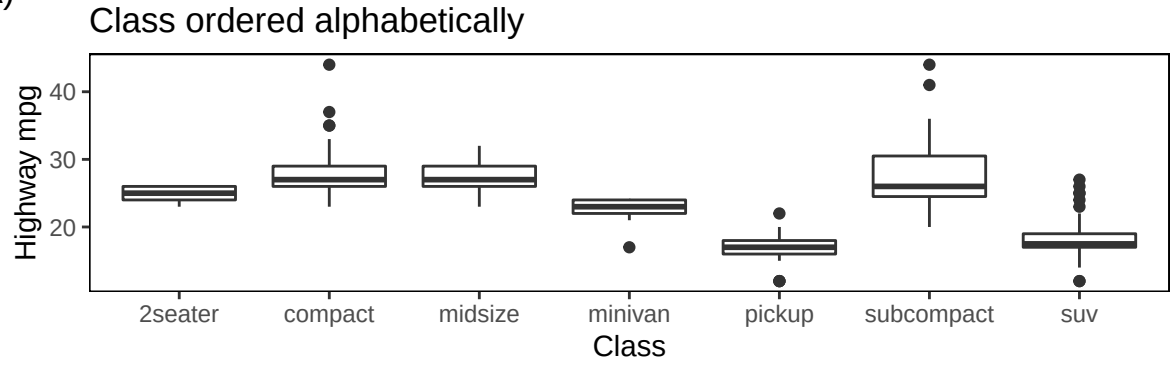
The main problem produced by the above code is that many of the points overlap, leading to overplotting of the data. Plotting the same data with `geom_jitter()` and mapping the `class` variable to the color aesthetic may alleviate this problem.



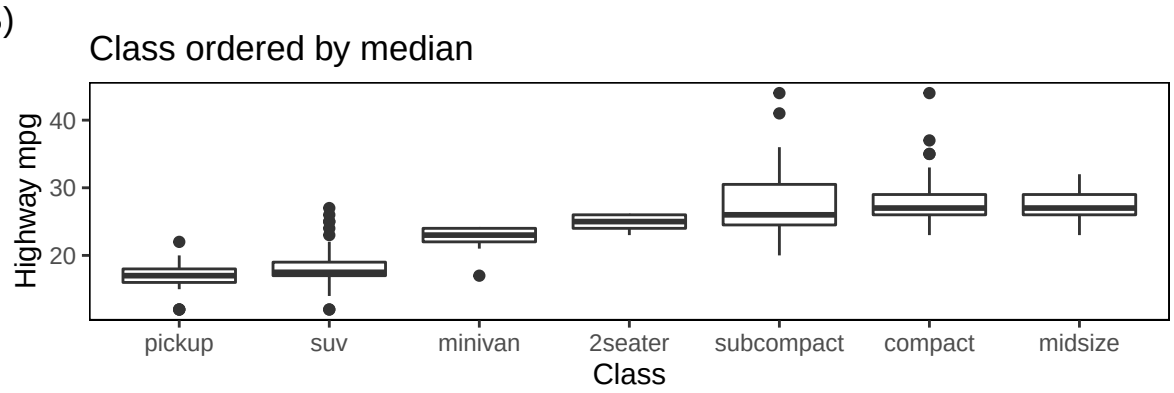

2) *One challenge with* `ggplot(mpg, aes(class, hwy)) + geom_boxplot()` *is that the ordering of* `class` *is alphabetical, which is not terribly useful. How could you change the factor levels to be more informative?*

Below, I've plotted the original boxplot, with `class` in alphabetical order, and the modified boxplot with `class` reordered by median.

A)

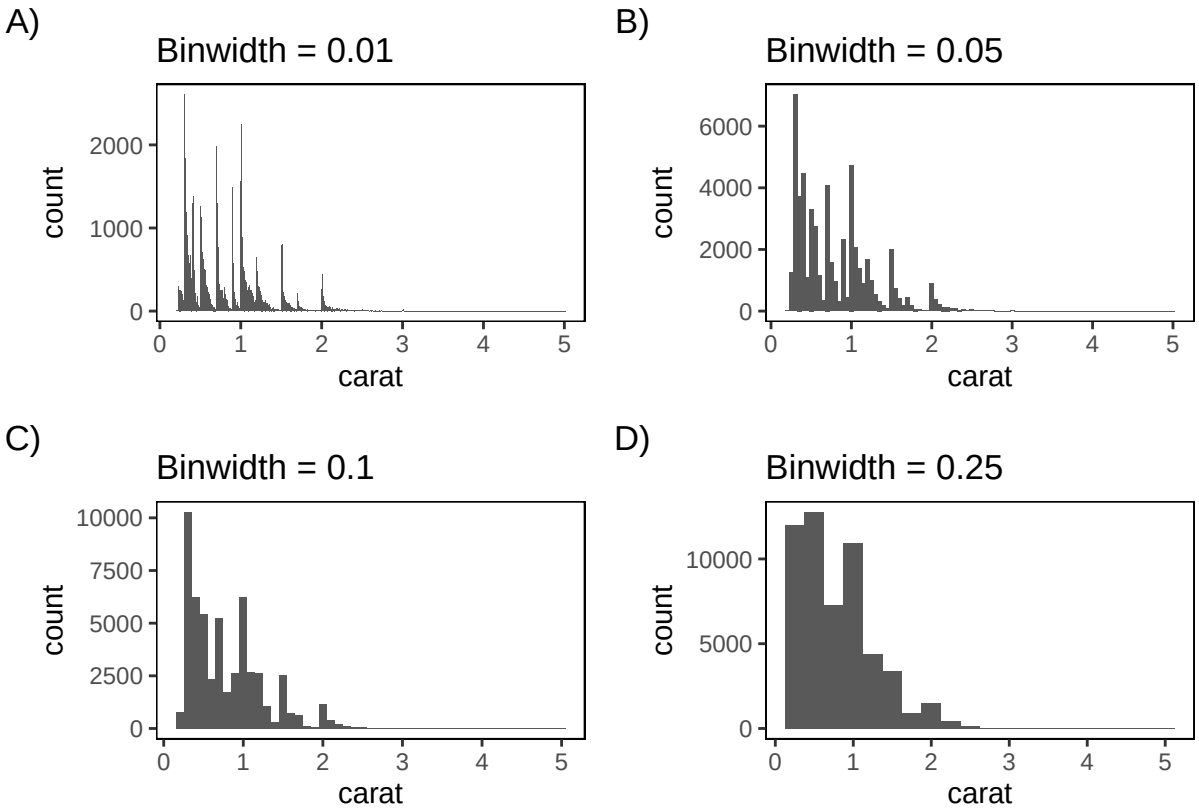



B)




3) *Explore the distribution of the carat variable in the* `diamonds` *dataset. What binwidth reveals the most interseting patterns?*

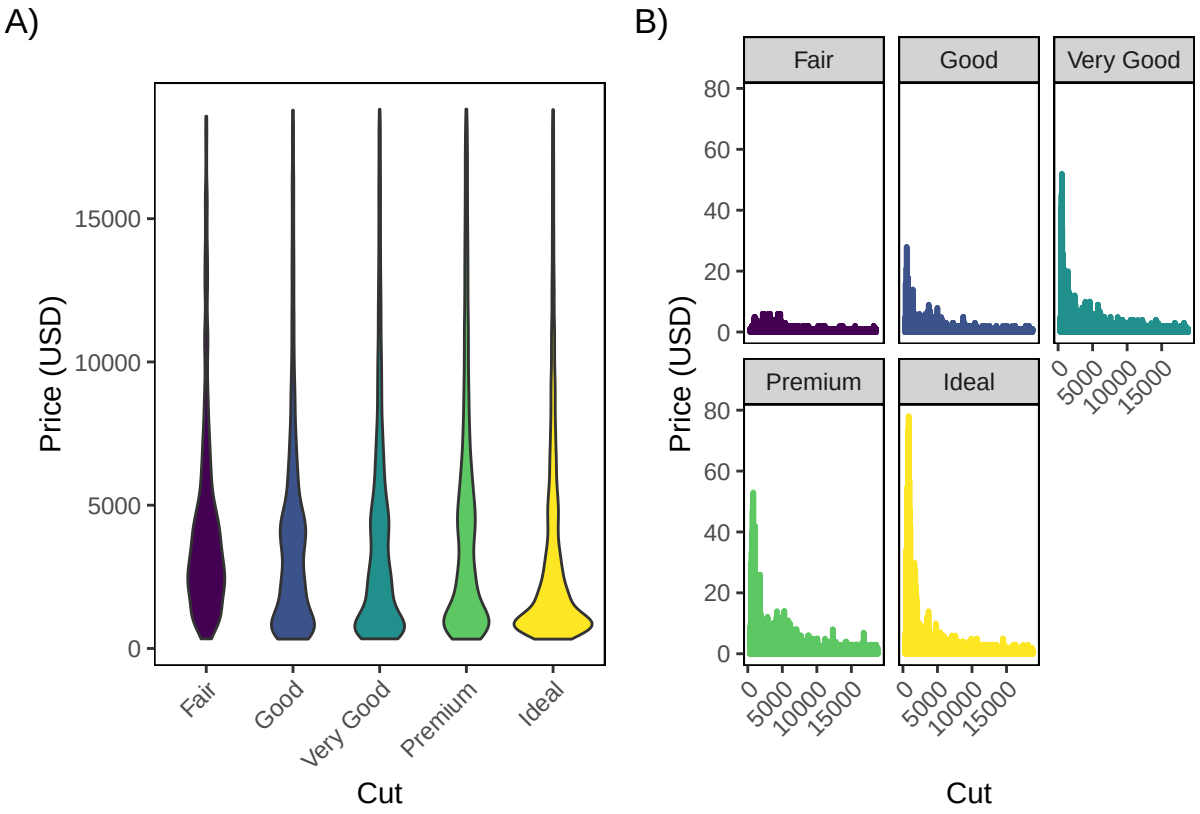
Below are several histograms to visualize the distribution of the `carat` variable in the `diamonds` dataset using different binwidths.




Viewing the plots above, we can see that the smaller binwidths reveal an interesting pattern- a tendency for more diamonds to be near whole-number carats. This seems plausible, given that it is likely easier for a jeweler to score a diamond using whole numbers instead of fractions of carats.

4) *Explore the distribution of the price variable in the* `diamonds` *data. How does the distribution vary by cut?*

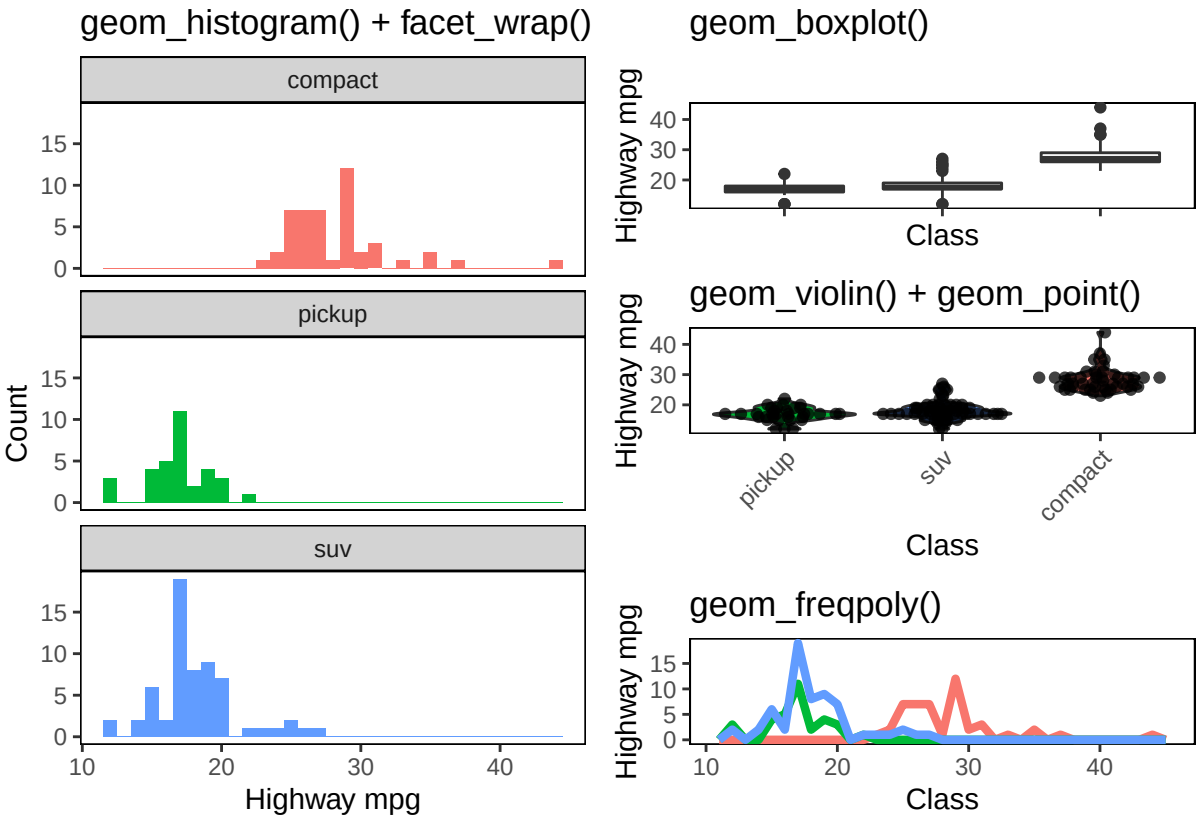
We can create a series of violin plots for diamond prices and frequency polygon plots faceted by cut to see the distribution of the price in diamonds for different cuts.




From plots A) and B) above, in general we can see that most of the diamonds are less than $5,000 for each category of `cut`, although some cuts are more variable than others. For example, there appears to be two distinct peaks in the price of diamonds in the good, very good, and premium categories, but less of any discernable pattern in the fair and ideal categories.

5) *You know (at least) three ways to compare the distributions of subgroups:* `geom_violin()`*,* `geom_freqpoly()` *and the* `color aesthetic,` *or* `geom_histogram()` *and faceting. What are the strengths and weaknesses of each approach? What are other approaches you could try?*

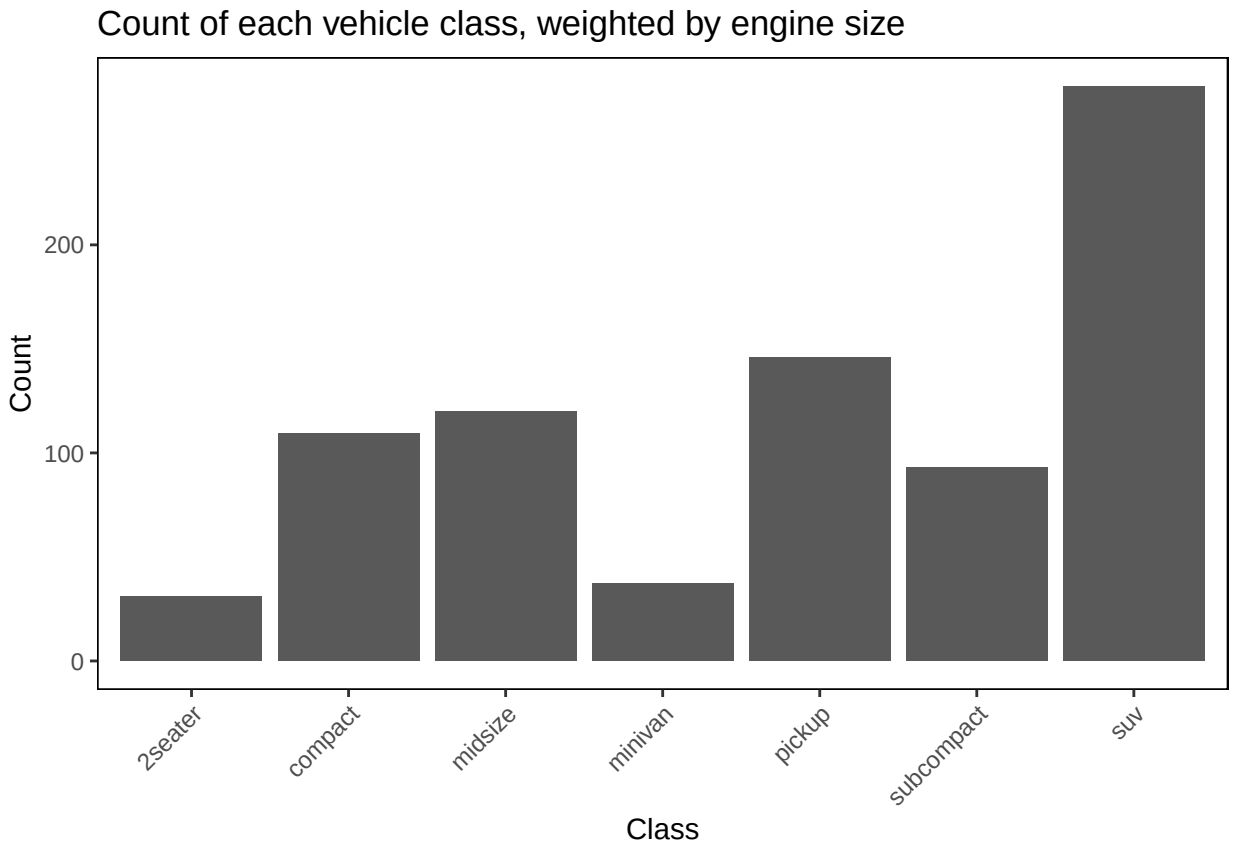
Below, we'll plot the distributions of highway fuel economy for each class of vehicle in a subset of the `mpg` dataset and compare the strengths and weaknesses of different approaches.




Faceting histograms by the `class` variable makes it easier to interpret the differences in the distributions of highway fuel economy, but this approach is likely only applicable to variables with relatively few levels. Faceting by a variable with many different categories may make it difficult to interpret the various subplots. The statistical transformations used in `geom_boxplot()` and `geom_violin()` allow for easy interpretation of the summaries of the data in a relatively compact format. Coupled with a `geom_point()` or `geom_jitter()` layer, we can easily see the distribution of individual points, while still describing the overall summary of the data. The `geom_freqpoly()` layer is useful as a compact representation of a faceted series of histograms, but in many situations where there are many values for a given variable, interpreting the plot may become difficult.

4) *Read the documentation for `geom_bar()`. What does the `weight` aesthetic do?*

In the default case, `geom_bar()` makes the height of the bar for a categorical variable proportional to the count of each case. If the `weight` aesthetic is used, the sum of the weights is used. For example, we can use one of the examples from the help documentation and plot the weights of the engine displacement as a function of the class of each vehicle.




5) *Using the techniques already discussed in this chapter, come up with three ways to visualize a 2d categorical distribution. Try them out by visualizing the distribution of model and manufacturer, trans and class, and cyl and trans.*