Introduction to ggplot2

Dr. Austin R. Brown

Kennesaw State University

Introduction

- Within R, there are two primary methods for creating visualizations: the Base R method and the `ggplot2` method. We're going to restrict our focus mostly to the latter method.
- The "gg" in `ggplot2` is shorthand for "grammar of graphics."
- It offers an incredible array of options for plotting all sorts of data types (including maps).
- We're going to learn how to plot some of the more common graphs as well as get an understanding of some options available to us prior to moving into more specific visualizations for answering specific types of questions.

Creating a ggplot

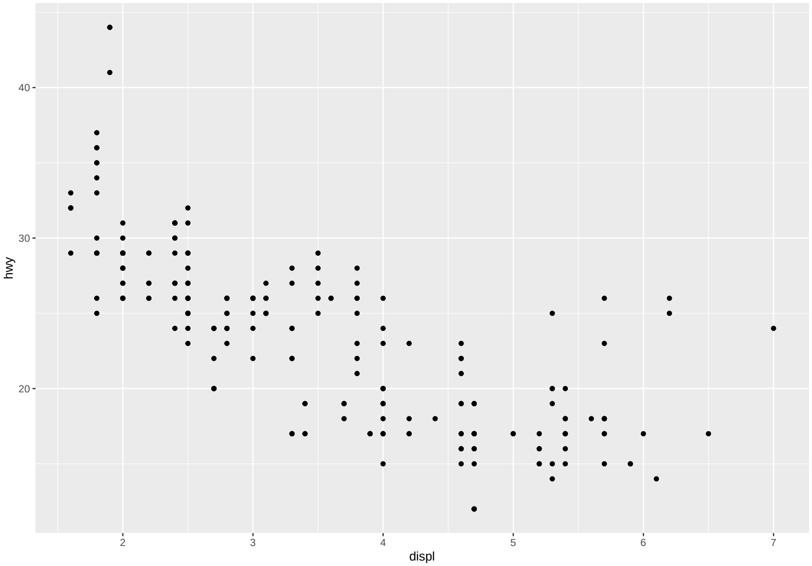
- Let's check out a basic example. Using the `mpg` dataframe, suppose we want to explore the relationship between a car's engine displacement (measured in liters) and its highway miles per gallon (mpg).

- Since both of these are quantitative variables and we're interested in better understanding the relationship between them, the most appropriate visualization would be a scatterplot.

Creating a ggplot

```r
library(tidyverse)
mpg <- ggplot2::mpg
mpg |>
ggplot() +
  geom_point(mapping = aes(x = displ,y = hwy))
```

Creating a ggplot



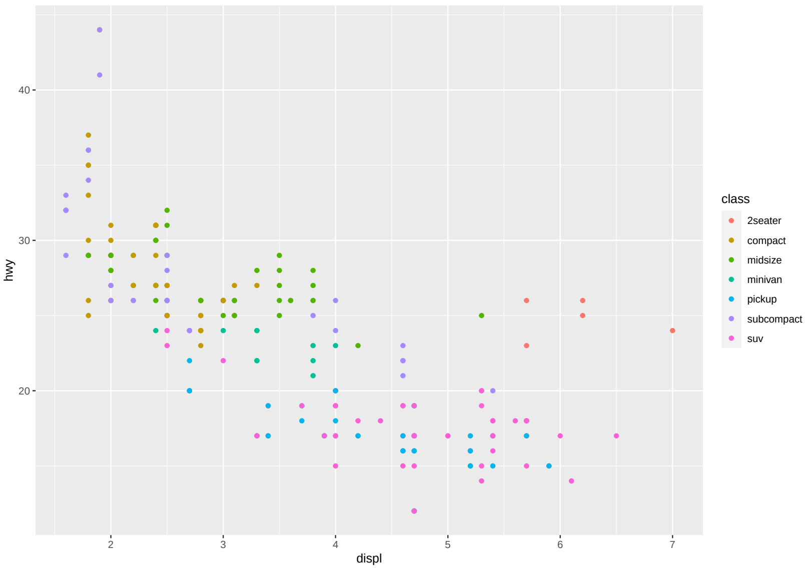
Creating a ggplot

- The visual characteristics of the objects in our plots are referred to as "aesthetics" (hence the `aes()` function).
- `aes()` controls lots of different aspects of our plots, including the size, shape, and color of the objects (bars, points, lines, etc) in the plots.
- To illustrate, let's say we now want to visualize the relationship between engine displacement and highway mpg controlling for the type of vehicle (e.g., minivan, compact, etc.)

Creating a ggplot

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

Creating a ggplot

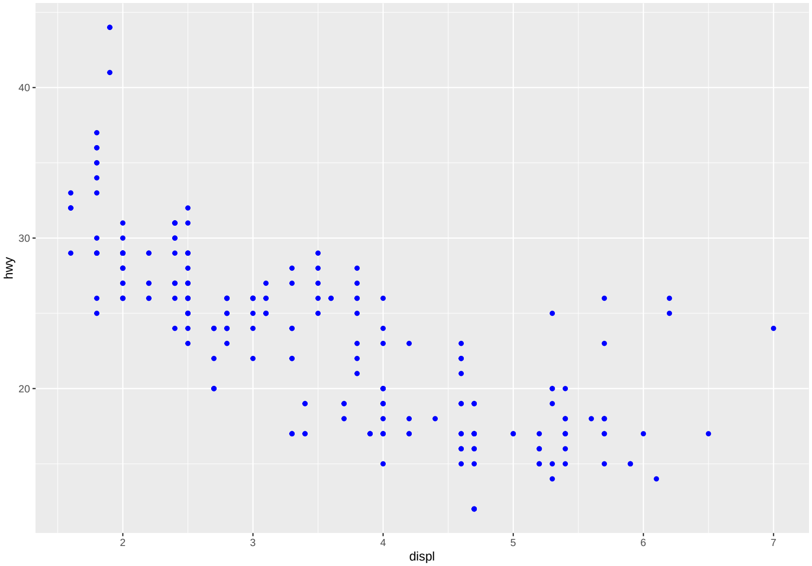


Creating a ggplot

- Now, we also have options that we can specify outside of the `aes()` function,including those we can specify inside of the `aes()` function.
- Let's say, for example, that we want the color of our points to be uniformly blue.

```r
mpg |>
ggplot() +
  geom_point(mapping = aes(x = displ, y = hwy),
             color='blue')
```

Creating a ggplot

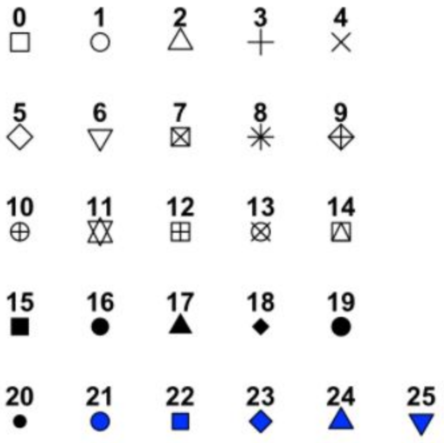


Creating a ggplot

- If we want to have a uniform aesthetic, we have to manually specify it outside of the `aes()` function with a value that is accepted by the aesthetic.

Aesthetic	Input
`color`	Name of color in quotes
`size`	Size of the points in millimeters
`shape`	A number (1-15) that indicates a shape
`alpha`	A number between 0 and 1 that indicates the opacity

Creating a ggplot



Creating a ggplot: Data Structure

- Now, let's try to create another visualization still using the MPG data.
- Suppose this time I want to visually compare the number (i.e., frequency) of vehicle classes contained in the dataset.
- Here, we need to understand what we are trying to visualize from a tabular perspective.

Creating a ggplot: Data Structure

- In the first example, we didn't have to transform the data in any way because each row represented a piece of information we wanted to visualize as it was.
- In this case, each row should represent one of the car classes (e.g., suv, compact, etc).
- And really, all we need is two columns: one for the class of vehicle and one for the number of times said class was observed in our dataset.
- Let's use some functionality in `dplyr` to help us solve this problem!

Creating a ggplot: Data Structure

```r
new_mpg <- mpg |>
  dplyr::group_by(class) |>
  dplyr::count()

new_mpg
```

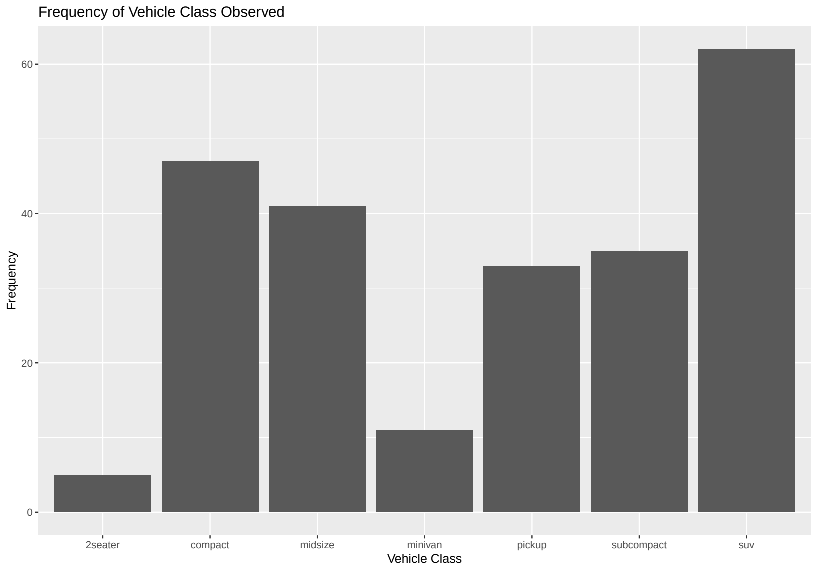
```
# A tibble: 7 x 2
# Groups:   class [7]
  class          n
  <chr>      <int>
1 2seater        5
2 compact       47
3 midsize       41
4 minivan       11
5 pickup        33
6 subcompact    35
7 suv           62
```

Creating a ggplot: Data Structure

- Now that the data are in the right structure, we can use a new `geom` called `geom_bar` to generate a bar chart:

```
new_mpg |>
  ggplot(aes(x=class,y=n)) +
  geom_bar(stat='identity') +
  labs(x = "Vehicle Class",
       y = "Frequency",
       title = "Frequency of Vehicle Class Observed")
```

Creating a ggplot: Data Structure



Creating a ggplot: More Customization

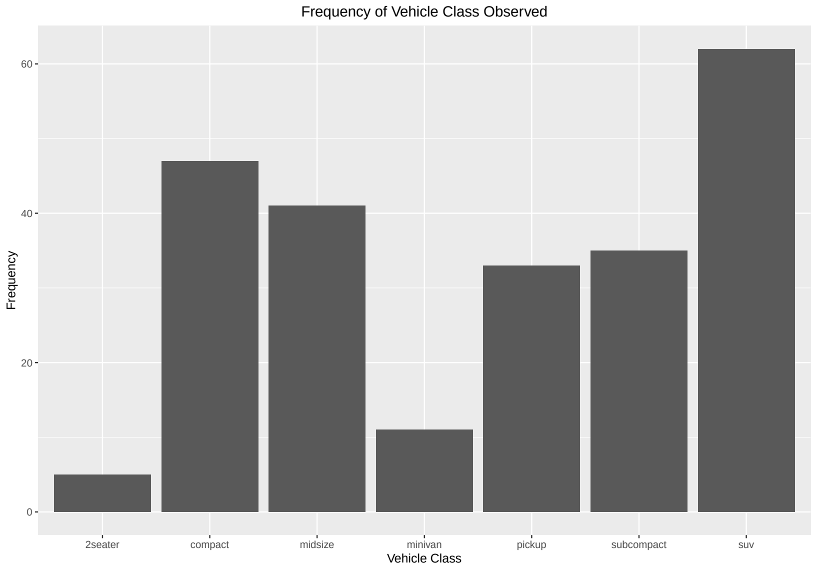
- Notice here, we used the `labs` function to generate descriptive axis and overall labels for the visualization.
    - This aids the reader in understanding what information is being conveyed by the visualization.

- By default, the overall visualization title and subtitle are left justified. If we would like them to be center justified, we have to make use of a new, general function called `theme`:

Creating a ggplot: More Customization

```r
new_mpg |>
  ggplot(aes(x=class,y=n)) +
  geom_bar(stat='identity') +
  labs(x = "Vehicle Class",
       y = "Frequency",
       title = "Frequency of Vehicle Class Observed") +
  theme(plot.title = element_text(hjust=0.50))
```

Creating a ggplot: More Customization



Creating a ggplot: More Customization

- We can also modify the overall theme or *look* of a visualization by adding theme prefixed functions, some of which come by default in the `ggplot2` package and some which can be added using outside packages, like `ggthemes`.
- Some of my favorite default themes are `theme_minimal`, `theme_classic`, and `theme_bw`