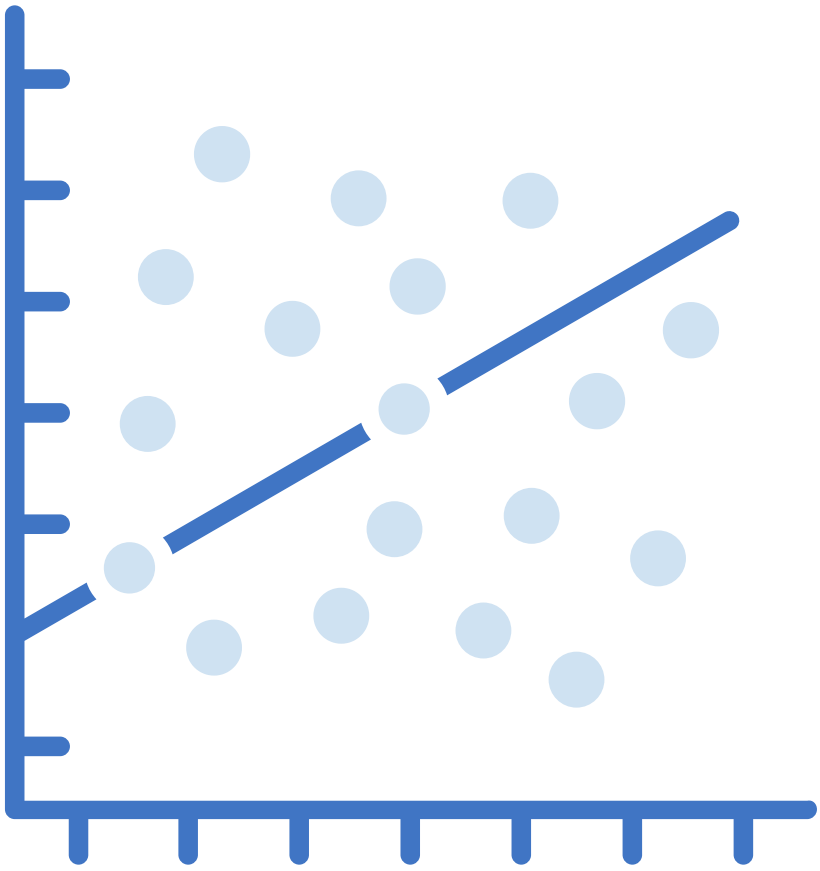
LEARN MORE 365



GGPLOT - Geometries Lesson

GGPLOT - geometries

Greg Martin

More about geometries

The most used geometries

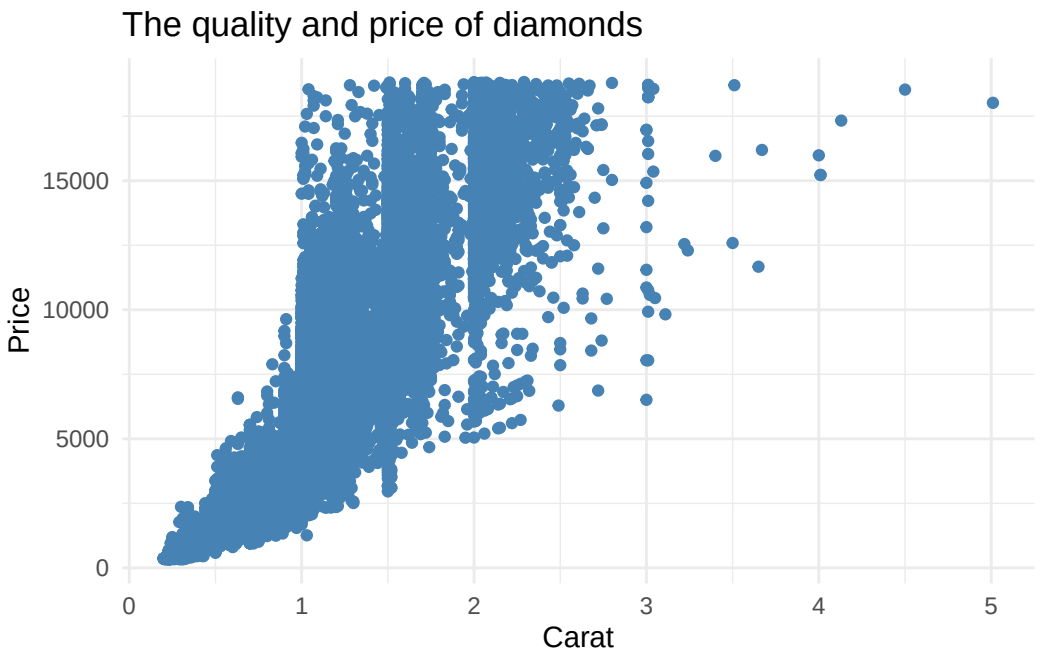
Lets take a look at a few of the `geometry` options. Here are some of the most commonly used geometries in ggplot2:

1. `geom_point()`: Displays points at the given x and y coordinates. We've been using this geometry in the examples above.
2. `geom_line()`: Connects points with a line to show trends or changes (usually over time).
3. `geom_bar()`: Displays bars of a given height, often used for displaying counts or proportions.
4. `geom_histogram()`: Displays a histogram to show the distribution of a numeric variable.
5. `geom_density()`: Displays a smoothed density plot to show the distribution of a numeric variable.
6. `geom_boxplot()`: Displays a box-and-whisker plot to show the distribution of a numeric variable.
7. `geom_area()`: Displays a filled area between the x axis and a line representing the values of a numeric variable.
8. `geom_raster` Displays a raster image or heatmap made up of a grid of pixels, with each pixel being assigned a color based on the data value it represents.

Scatter plots

We've already used scatter plots in the examples above. Scatter plots are usually used to visualize the relationship between two numeric variables. In the example below we see a clear relationship between the quality of diamonds.

```
diamonds %>%
  ggplot(aes(carat, price))+
  geom_point(color = "steelblue")+
  labs(title = "The quality and price of diamonds",
       x = "Carat", y = "Price")+
  theme_minimal()
```



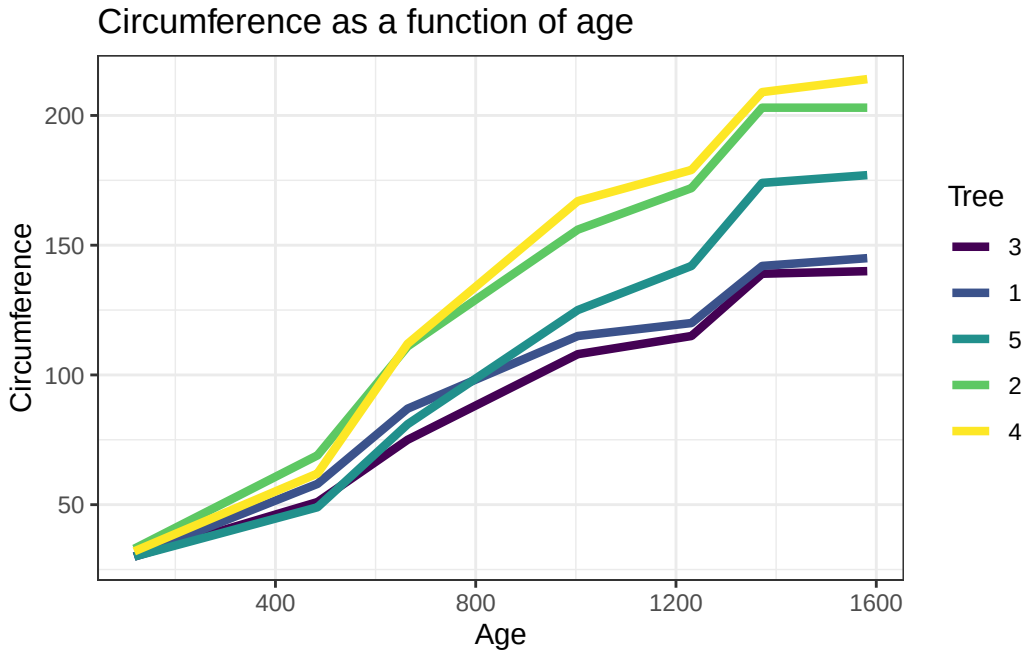
Line graphs

Line graphs are a powerful tool to demonstrate trends and change (usually over time). The `Orange` dataset contains data about a number of trees, including their circumference at different ages. Let's take a look:

```
Orange %>%
  ggplot(aes(x = age,
             y = circumference,
```

```
            color = Tree))+
  geom_line(size = 1.5)+
  labs(title = "Circumference as a function of age",
       x = "Age",
       y = "Circumference")+
  theme_bw()
```

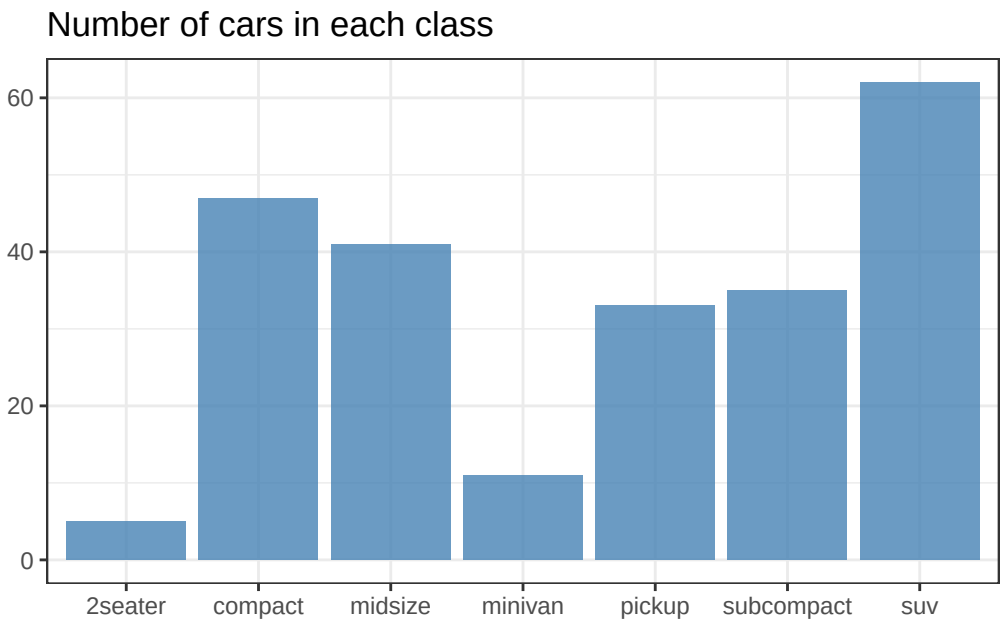
```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```



Bar charts

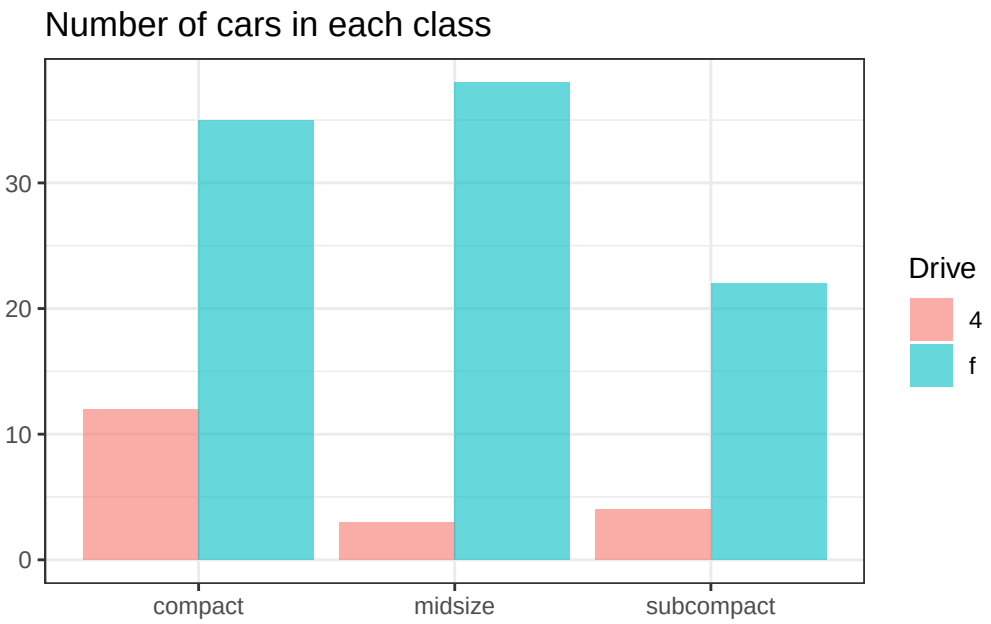
One of the more comment graphs used to represent a single categorical variables is the good old bar chart. Bar charts are the most common way of representing a count or proportion of a categorical variable (or factor). Let's take a look at an example using the `mpg` dataset that you're already familiar with:

```
mpg %>%
  ggplot(mapping = aes(x = class))+
  geom_bar(fill = "steelblue",
           alpha = 0.8)+
  labs(title = "Number of cars in each class",
       x = "",
       y = "")+
  theme_bw()
```

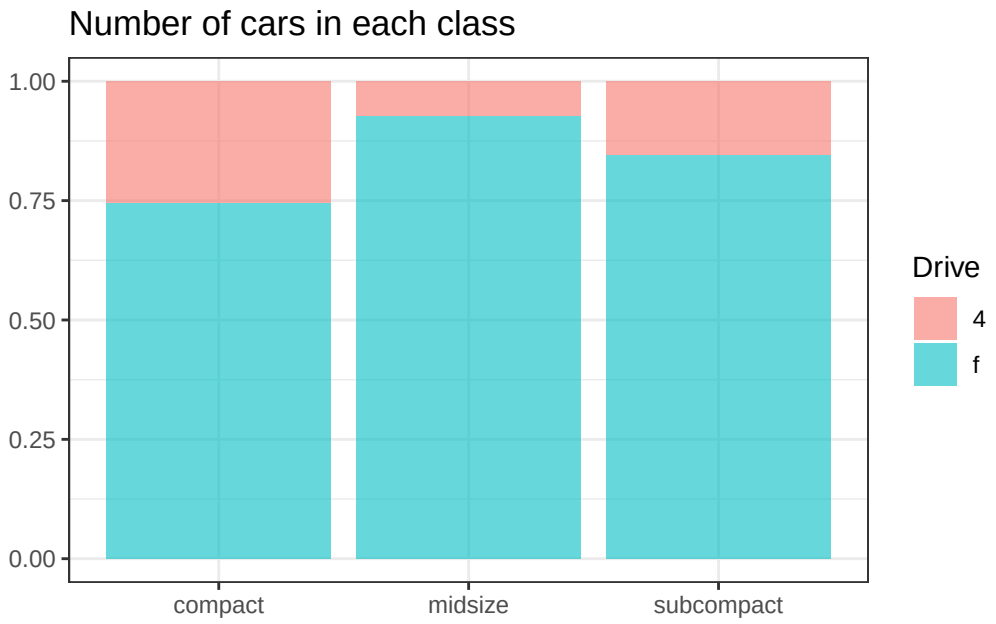


We might be interested in considering the number of cars in each class disaggregated by `drv` variable. In the examples below, we'll look at `compact`, `subcmpact` and `midsize` cars and compare four wheel drives and front wheel drives. Notice in the code below we've included the argument `position = "dodge"` in the `geom_bar()` function to indicate that we want the bars of the subcategories of data to be next to each other.

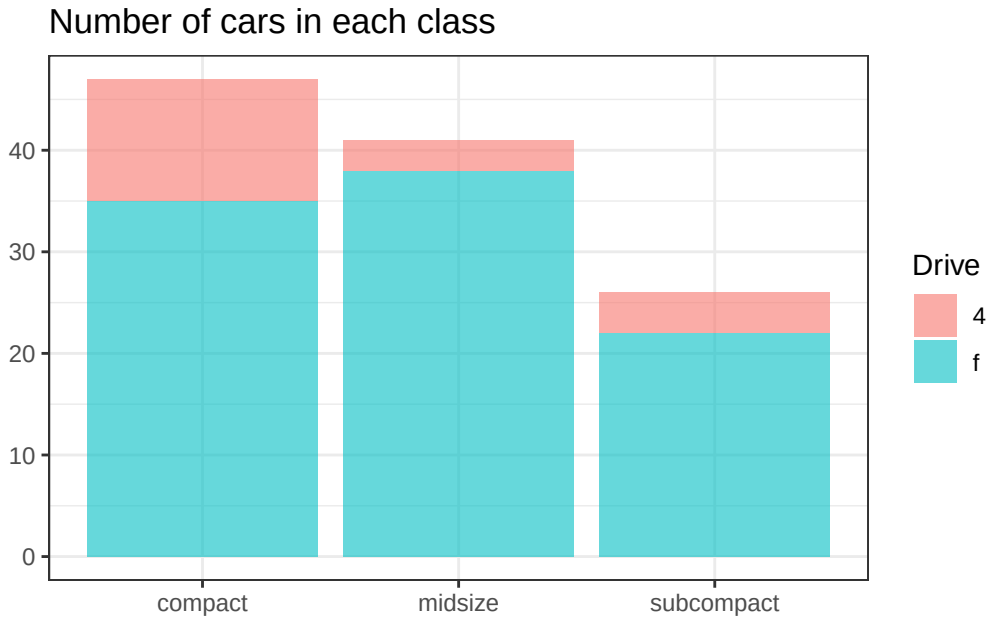
```r
mpg %>%
  filter(class %in% c("compact", "subcompact","midsize"),
         drv %in% c("4", "f")) %>%
  ggplot(mapping = aes(x = class, fill = drv))+
  geom_bar(alpha = 0.6, position = "dodge")+
  labs(title = "Number of cars in each class",
       x = "",
       y = "",
       fill = "Drive")+
  theme_bw()
```



By changing the `position = "dodge"` argument to `position = "fill"` we can show the relative proportion of each.



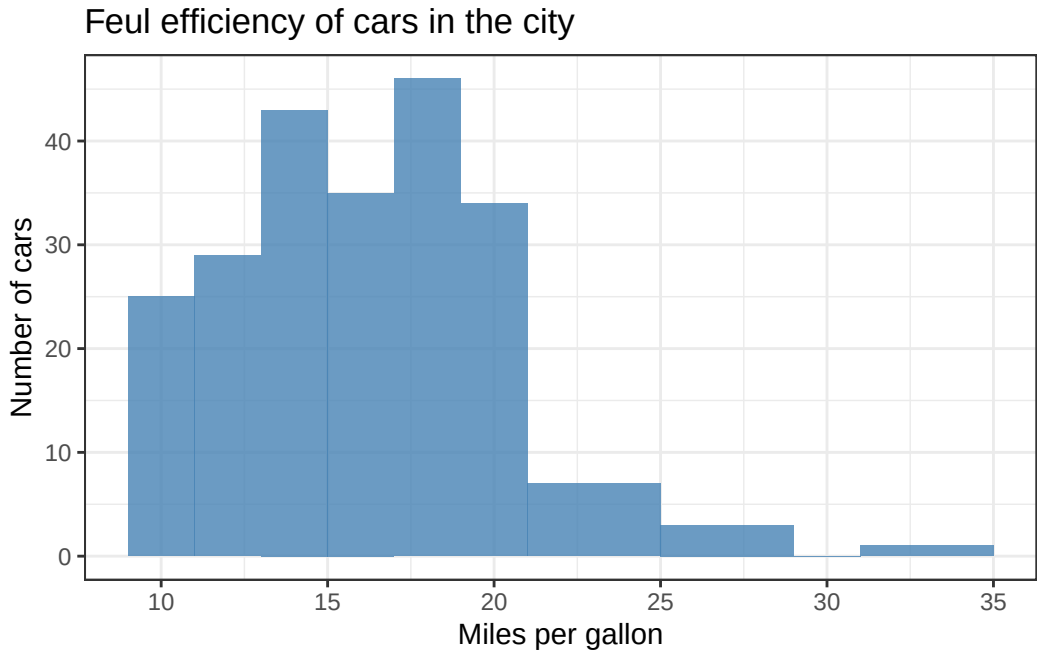
And finally, `position = "stacked"` will (as the argument suggests) give you this:



Histogram

To visualize a single numeric variable we have few options. Let's start with the histogram. This shows the distribution of values within increments that we call `bins`. There isn't a default bin width but rather, ggplot tried to calculate the most appropriate bin width based on the range of data in your dataset. You will however often need to define the bin width yourself. To do this you use `binwidth` argument. Take a look:

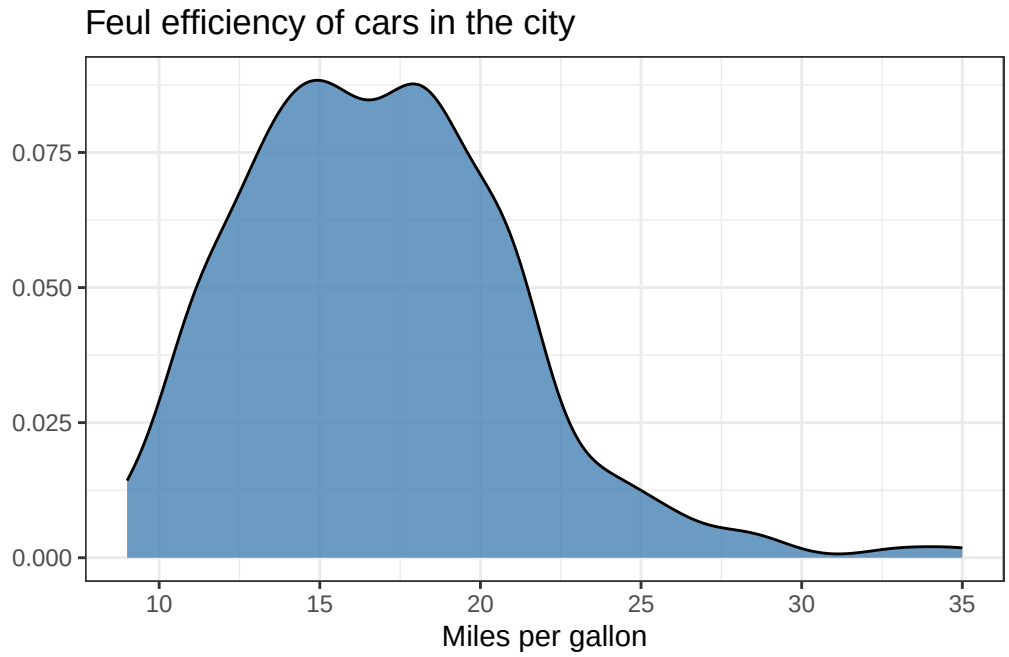
```
mpg %>%
  ggplot(aes(x = cty))+
  geom_histogram(binwidth = 2,
                 fill = "steelblue",
                 alpha = 0.8)+
  labs(title = "Feul efficiency of cars in the city",
       x = "Miles per gallon",
       y = "Number of cars")+
  theme_bw()
```



Density plots

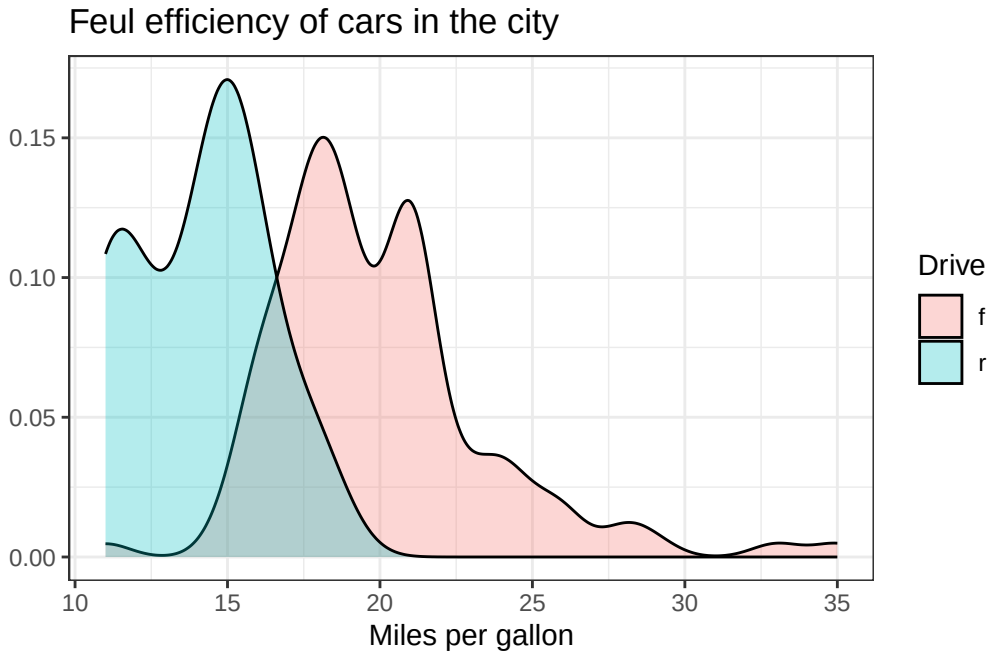
We can plot numeric variables using a density plot in the same way that we do with a histogram. In this case however, the plot represents the probability (0-1) of a given value on the x axis. If that doesn't make sense, just look at the plot below and it will all fall into place for you:

```
mpg %>%
  ggplot(aes(x = cty))+
  geom_density(fill = "steelblue",
               alpha = 0.8)+
  labs(title = "Feul efficiency of cars in the city",
       x = "Miles per gallon",
       y = "")+
  theme_bw()
```



A useful application of density plots is to layer them onto of each other to visualize the different in distribution for the values of a numeric variable separated out by the values of a categorical variable. Lets compare the fuel efficiency of front wheel and rear wheel drives in the city.

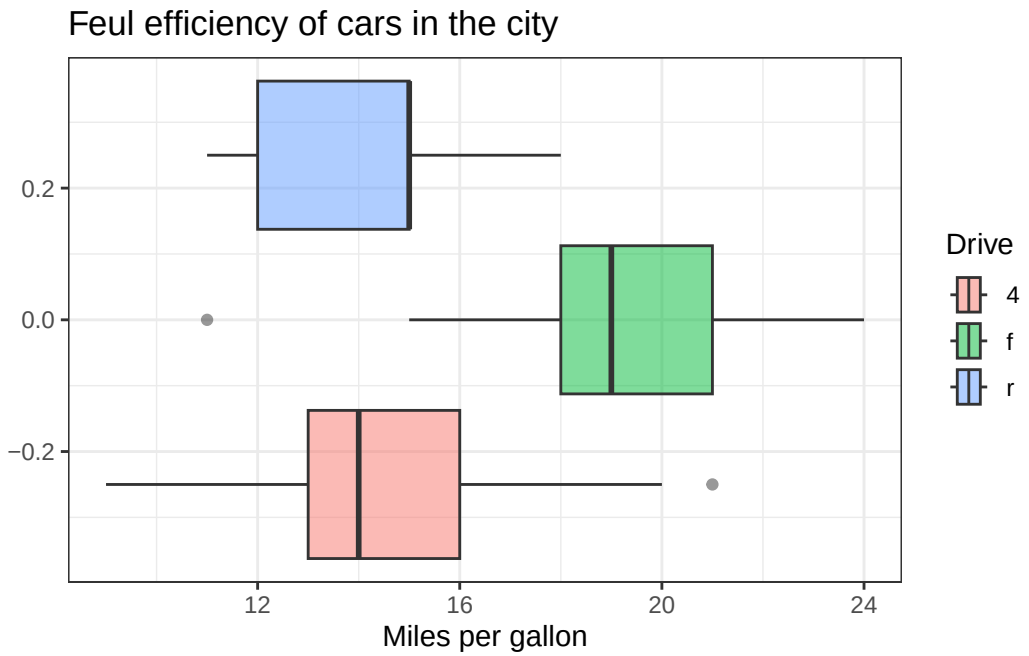
```
mpg %>%
  filter(drv %in% c("f", "r")) %>%
  ggplot(aes(x = cty,
             fill = drv))+
  geom_density(alpha = 0.3)+
  labs(title = "Feul efficiency of cars in the city",
       x = "Miles per gallon",
       y = "",
       fill = "Drive")+
  theme_bw()
```



Boxplots

Numeric variables are often represented using boxplots. In the plots below, 50% of the data (the interquartile range) is within the box and the line inside the box represents the median value. Let's restrict the data to only those cars with with a `cty` value of less than 25.

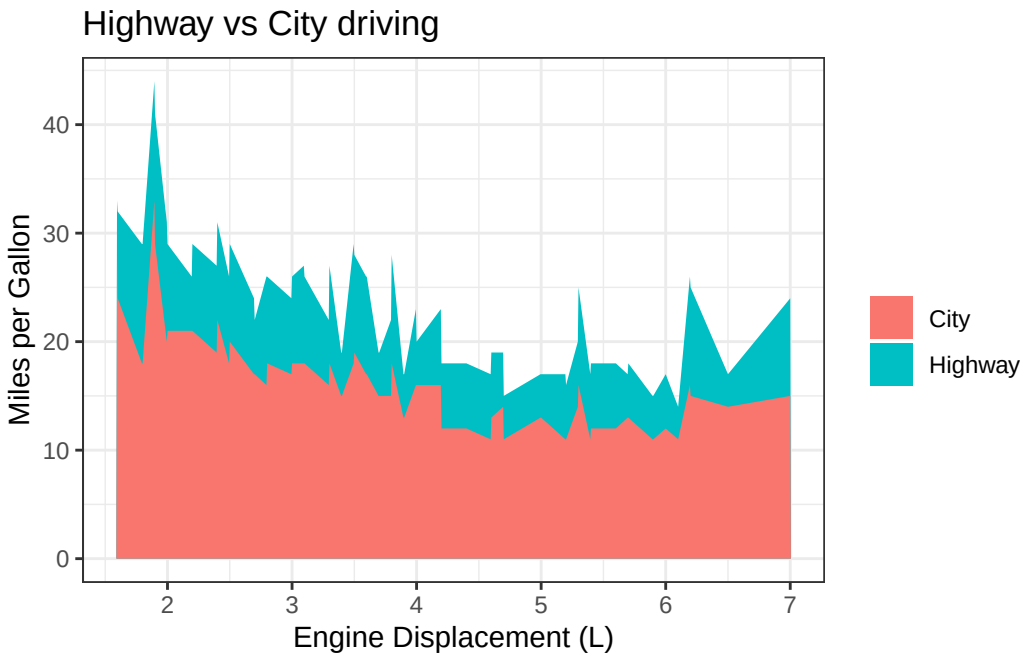
```
mpg %>%
  filter(cty < 25) %>%
  ggplot(aes(x = cty,
             fill = drv))+
  geom_boxplot(alpha = 0.5)+
  labs(title = "Feul efficiency of cars in the city",
       x = "Miles per gallon",
       fill = "Drive")+
  theme_bw()
```



Area plots

The `geom_area()` will fill in the area between the x axis and what would have been a line graph. This is often used to compare the overall effect of two factors (or categories). In the example below, we've plotted the areas for Mile per Gallon for different size engines. Notice that in order to overlay the area plot for highway and city driving, we needed to define the aesthetic mapping for the `y axis` separately for each layer.

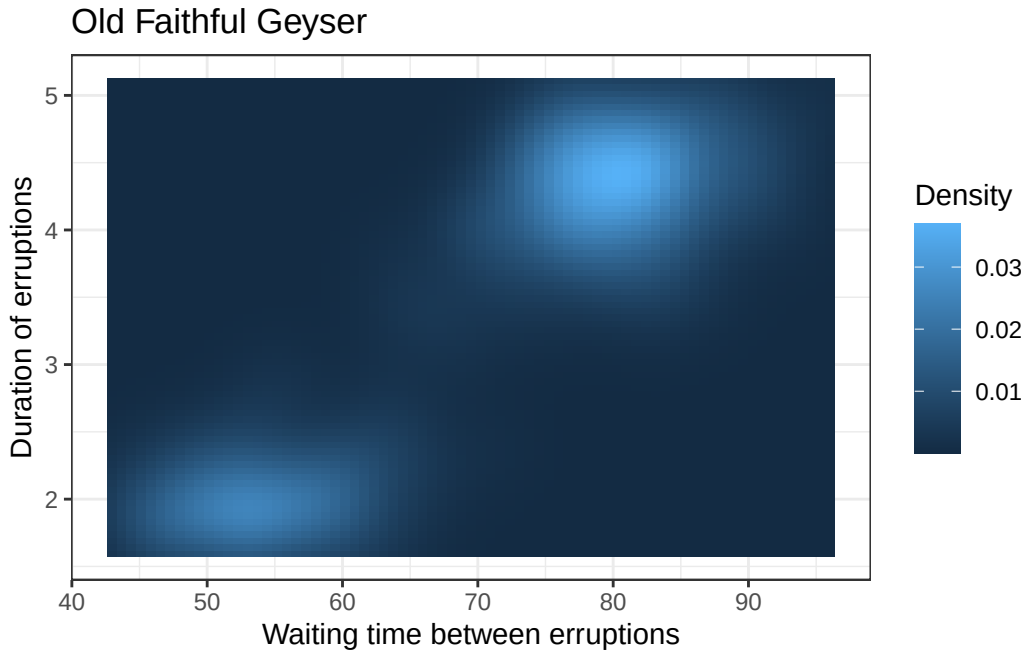
```
ggplot(mpg, aes(x = displ))+
  geom_area(aes(y = hwy, fill =  "Highway")) +
  geom_area(aes(y = cty, fill = "City"))+
  labs(title = "Highway vs City driving",
       x = "Engine Displacement (L)",
       y = "Miles per Gallon",
       fill = "")+
  theme_bw()
```



Raster plots

`geom_raster()` creates a colored heatmap, with two variables acting as the x- and y-coordinates and a third variable mapping onto a color. In the example below we have data for the "Old Faithful Geyser" including waiting time between eruptions, eruption duration and density. The strength of this plot is that it helps us visualize three variables at once.

```
faithfuld %>%
  ggplot(aes(x = waiting,
             y = eruptions,
             fill = density)) +
  geom_raster()+
  labs(title = "Old Faithful Geyser",
       x = "Waiting time between erruptions",
       y = "Duration of erruptions",
       fill = "Density")+
  theme_bw()
```



Learn More 365

Share this cheat sheet with others

Unlock a wealth of FREE resources TODAY!

Click on the links below to enrich your knowledge in these topics:

- Statistics & Research Methods Resource Library
- R Programming & Data Visualization Resource Library
- Discover the Public Health & Epidemiology Resource Library

Explore video lessons on **Statistics & Research Methods, R Programming & Data Visualization Public Health & Epidemiology,** and more at Learn More 365!

Click **HERE** or scan the QR code below to start learning!