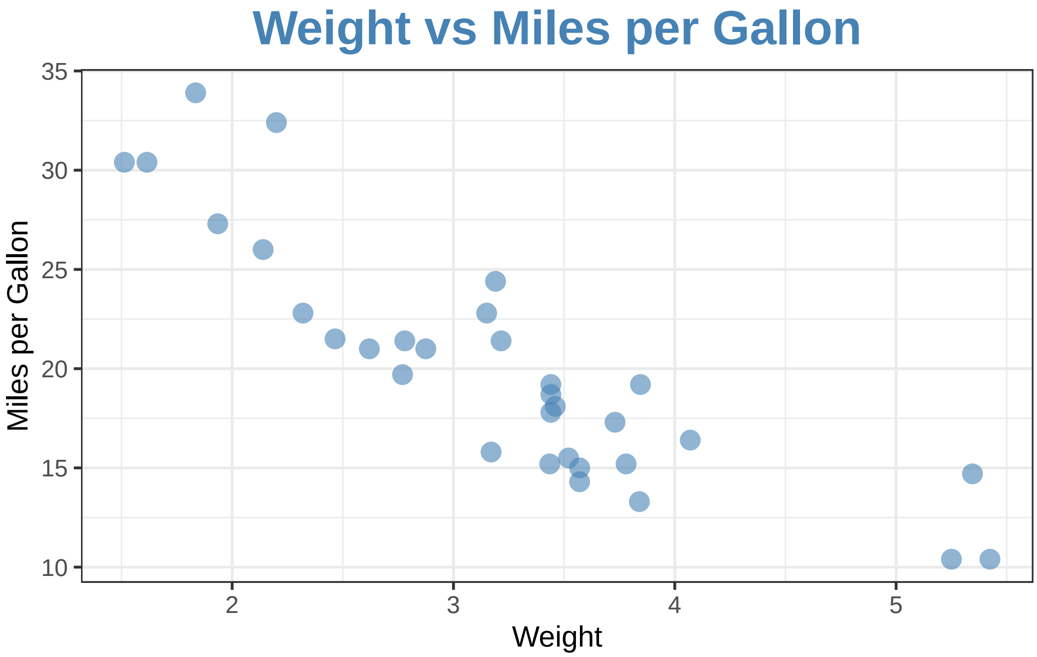
Titles, axis labels, legends and annotations

Greg Martin

Titles, axis labels and annotation

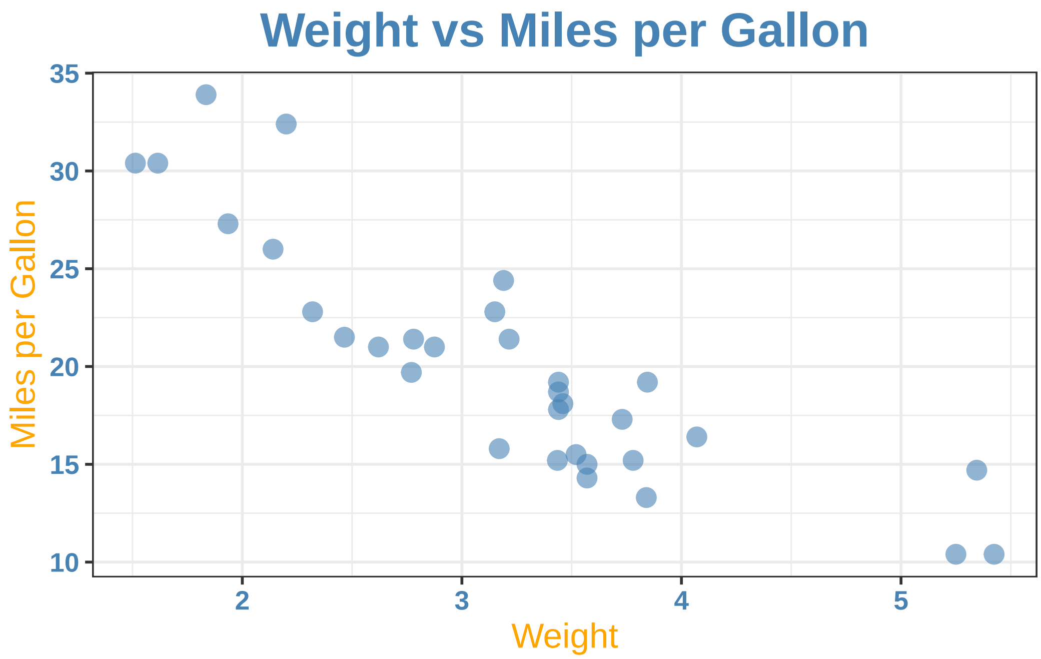
Once you've created a beautiful plot, you'll want to create some clear signposting in the form of titles, subtitles, axis labels, legends and perhaps even some annotations. The `labs` function let's you define quite a few of these (as you will have already seen in all of the preceding examples). Using the `theme()` function you can start to tweak the look and feel of your titles and labels. In the code below, we've stipulated values for `element_text()` within the `theme()` function that change the text of the title. The elements changes are all pretty self explanatory with the exception of `hjust` that stands for horizontal adjustment. The value 0.5 means that we want the text to be half way across the horizontal axis (in other words centered).

```
mtcars %>%
  ggplot(aes(x = wt, y = mpg)) +
  geom_point(size = 3,
             colour = "steelblue",
             alpha = 0.6)+
  labs(title = "Weight vs Miles per Gallon",
       x = "Weight",
       y = "Miles per Gallon")+
  theme_bw()+
  theme(plot.title =
          element_text(size = 18,
                       face = "bold",
                       color = "steelblue",
                       hjust = 0.5))
```



Similarly, we can change the font size, color, weight etc. of both the x and y axis titles and the tic mark labels too. Take a look:

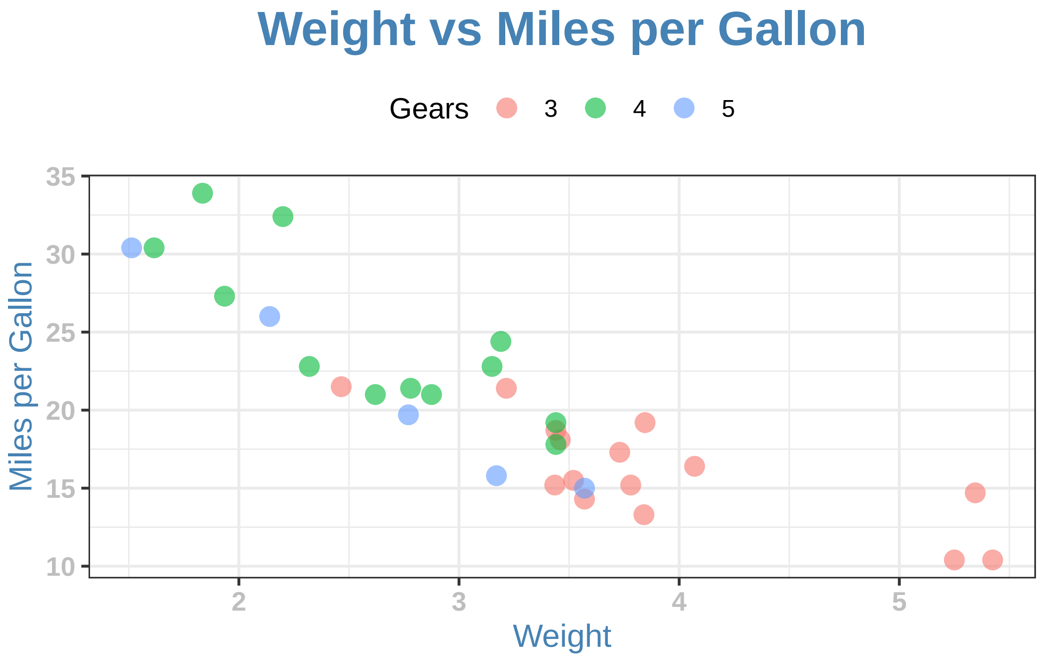
```
mtcars %>%
  ggplot(aes(x = wt, y = mpg)) +
  geom_point(size = 3,
             colour = "steelblue",
             alpha = 0.6)+
  labs(title = "Weight vs Miles per Gallon",
       x = "Weight",
       y = "Miles per Gallon")+
  theme_bw()+
  theme(plot.title =
          element_text(size = 18,
                       face = "bold",
                       color = "steelblue",
                       hjust = 0.5),
        axis.text =
          element_text(size = 10,
                       color = "steelblue",
                       face = "bold"),
        axis.title =
          element_text(size = 13,
                    color = "orange"))
```



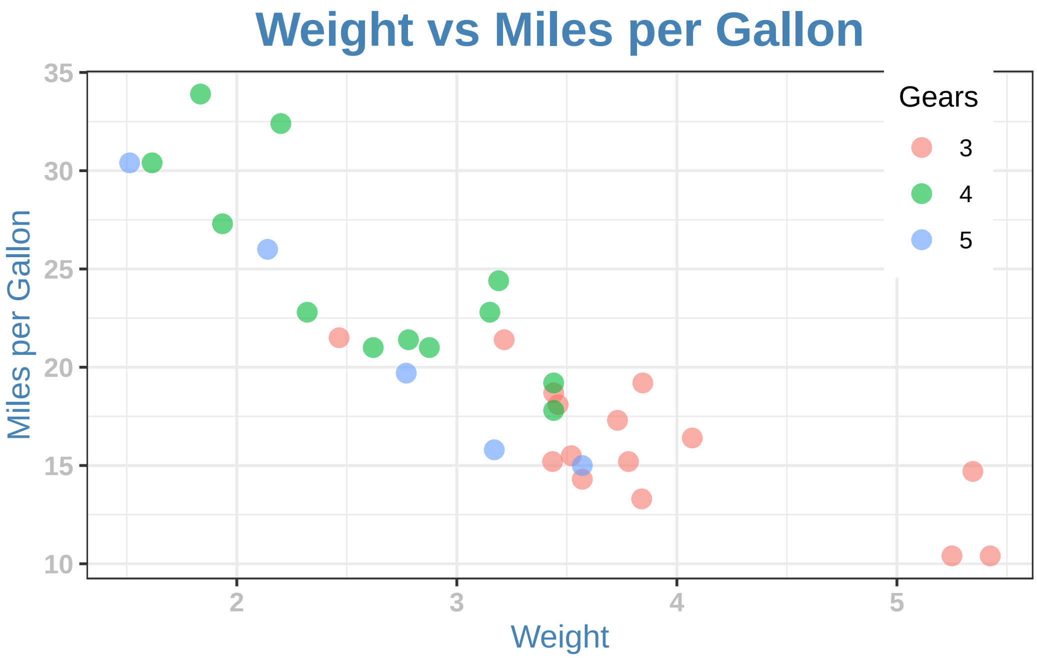
Legend position

Now let's take a look at the position of a legend. The default is to place the legend on the right hand side of the plot. This can easily be changed by adding an argument into the `theme()` that specifies `legend.position = "top"`

```
mtcars %>%
  ggplot(aes(x = wt, y = mpg, color = factor(gear))) +
  geom_point(size = 3,
             alpha = 0.6)+
  labs(title = "Weight vs Miles per Gallon",
       x = "Weight",
       y = "Miles per Gallon",
       color = "Gears")+
  theme_bw()+
  theme(plot.title =
          element_text(size = 18,
                       face = "bold",
                       color = "steelblue",
                       hjust = 0.5),
        axis.text =
          element_text(size = 10,
                       color = "grey",
                       face = "bold"),
        axis.title =
          element_text(size = 12,
                       color = "steelblue"))+
  theme(legend.position = "top")
```



As you can imagine, you can also specify that the legend is positioned at the `bottom` or `left` of the plot. If you want to have the legend inside the plot itself you can specify how far along the x and y axis you want it to be with the code `theme(legend.position = c(0.9, 0.8))` that will ensure that the legend is positioned 90% across from left to right, and 80% from bottom to top, to give you a plot that looks like this:

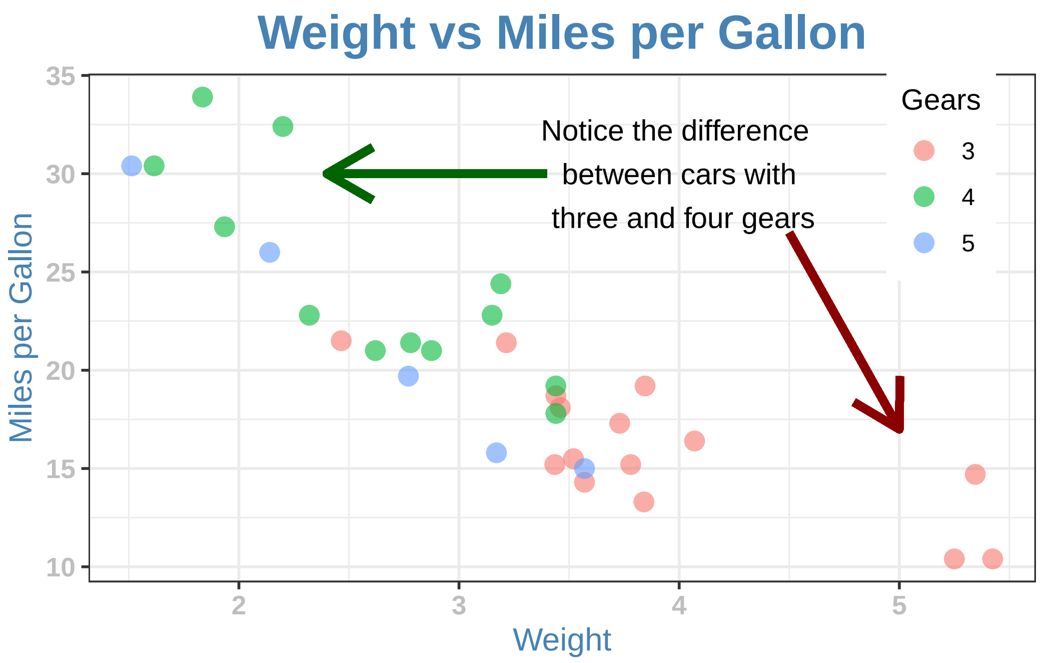


Adding annotations, lines and arrows

In the plot below we've added an annotation and two arrows. Most of the code is self explanatory but we'll point out a few details that are worth noting. When using the `annotate` function you need to specify where the annotation is going to start in terms of x and y coordinates that map against your x and y axis. The actual text is specified in the `label` argument. If you want to include a line break within the text, as we have below, then simply add `\n` to indicate a new line within the text. The usual arguments can then be applied (`size`, `face`, `colour`, etc.)

When creating the arrows we used the `geom_segment()` function and specified the `x` and `y` coordinates for the beginning of the line and then the `xend` and `yend` coordinates for the end of the line. The `arrow` argument let's you specify the size of the arrow. Thereafter, the usual arguments apply (`size`, `color`, etc.)

```
mtcars %>%
  ggplot(aes(x = wt, y = mpg, color = factor(gear))) +
  geom_point(size = 3,
             alpha = 0.6)+
  labs(title = "Weight vs Miles per Gallon",
       x = "Weight",
       y = "Miles per Gallon",
       color = "Gears")+
  theme_bw()+
  theme(plot.title =
          element_text(size = 18,
                       face = "bold",
                       color = "steelblue",
                       hjust = 0.5),
        axis.text =
          element_text(size = 10,
                       color = "grey",
                       face = "bold"),
        axis.title =
          element_text(size = 12,
                       color = "steelblue"))+
  theme(legend.position = c(0.9,0.8))+
  annotate("text", x = 4, y = 30,
           label = "Notice the difference \n between cars with \n three and four gears",
           color = "black",
           face = "bold",
           size = 4) +
  geom_segment(x = 4.5, y = 27,
               xend = 5, yend = 17,
               arrow = arrow(length = unit(0.7, "cm")),
               color = "darkred",
               size = 1.5)+
  geom_segment(x = 3.4, y = 30,
               xend = 2.4, yend = 30,
               arrow = arrow(length = unit(0.7, "cm")),
               color = "darkgreen",
               size = 1.5)
```


Weight vs Miles per Gallon
Notice the difference
between cars with
three and four gears
Gears
3
4
5
Miles per Gallon
35
30
25
20
15
10
2
3
4
5
Weight

More help

For more help and support, click on the title (hyperlink) or scan any of the QR codes below.

Labels and annotation	Wrangle your data video	Visualize your data video

Course on data analysis	Course on how to visualize data	Course on statistical analysis

Learn More 365

Share this cheat sheet with others

Course on:

- Introduction to R programming
- Data visualisation with R
- Advanced ggplot2
- Statistical analysis with R
- Research methods
- Epidemiology
- How to write an academic paper ... and more

Get access to the Resource Library by clicking the links below:

R Programming, Data Visualization and Research Methods

Resource Library

Or scan this QR code to go to www.LearnMore365.com