barplots and scatterplots

Emily Malcolm-White

`ggplot2` is a package built within the `tidyverse` package for creating awesome graphs!

Figure 1: Artwork by @allisonhorst

```
library(tidyverse)
```

```
#Import the can_lang dataset
can_lang <- read.csv("https://raw.githubusercontent.com/ttimbers/canlang/master/inst/extda
```

Recall our Top 10 example:

This code gave a list of 10 Aboriginal Languages which have the most number of people who speak them as their mother tongue:

```
ten_lang <- can_lang %>%
  filter(category == "Aboriginal languages") %>%
```

```
  arrange(by=desc(mother_tongue)) %>%
  select(language, mother_tongue) %>%
  slice(1:10)
```

Barplots

Suppose we wanted to display this information in a barplot instead of in a table. Let's take a look at the `ggplot` syntax:

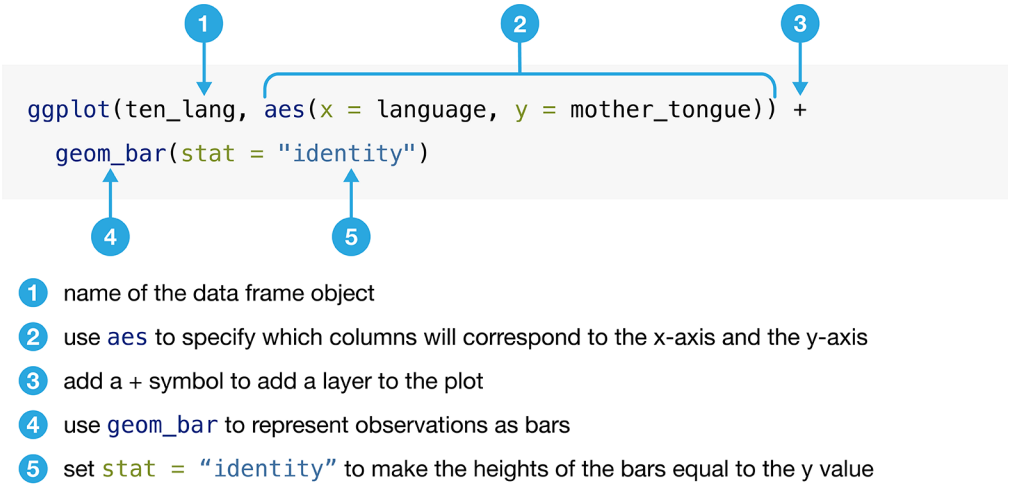
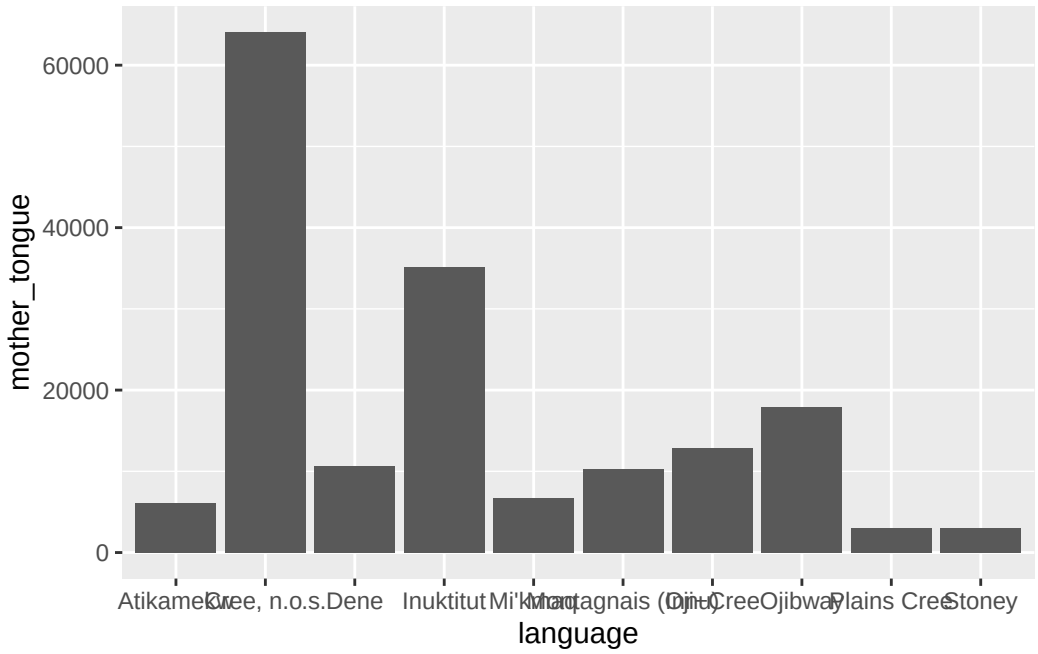


Figure 2: Credit: https://github.com/UBC-DSCI/introduction-to-datascience/

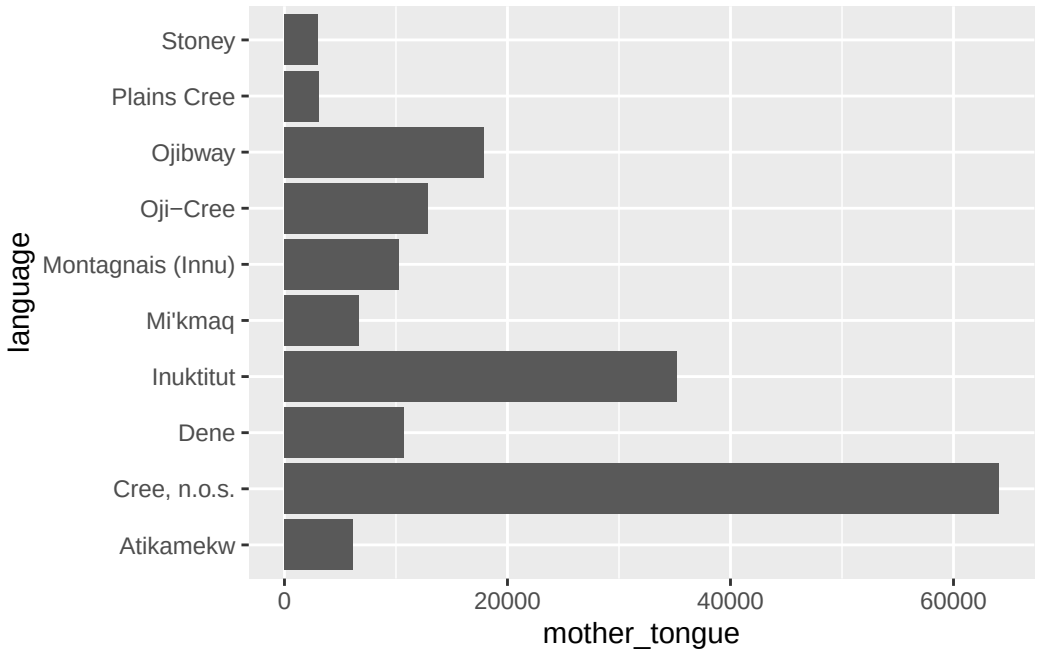
```
ggplot(ten_lang, aes(x = language, y = mother_tongue)) +
  geom_bar(stat = "identity")
```



Is there any improvements we could make to this graph?

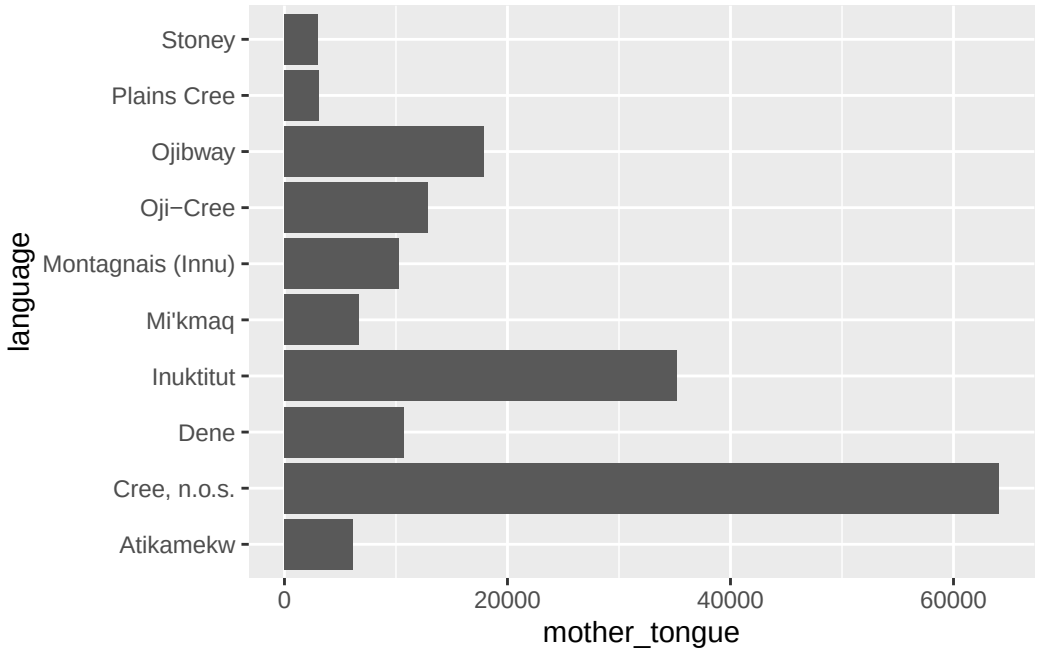
To better view text

```
ggplot(ten_lang, aes(x = language, y = mother_tongue)) +
  geom_bar(stat = "identity") +
  coord_flip()
```



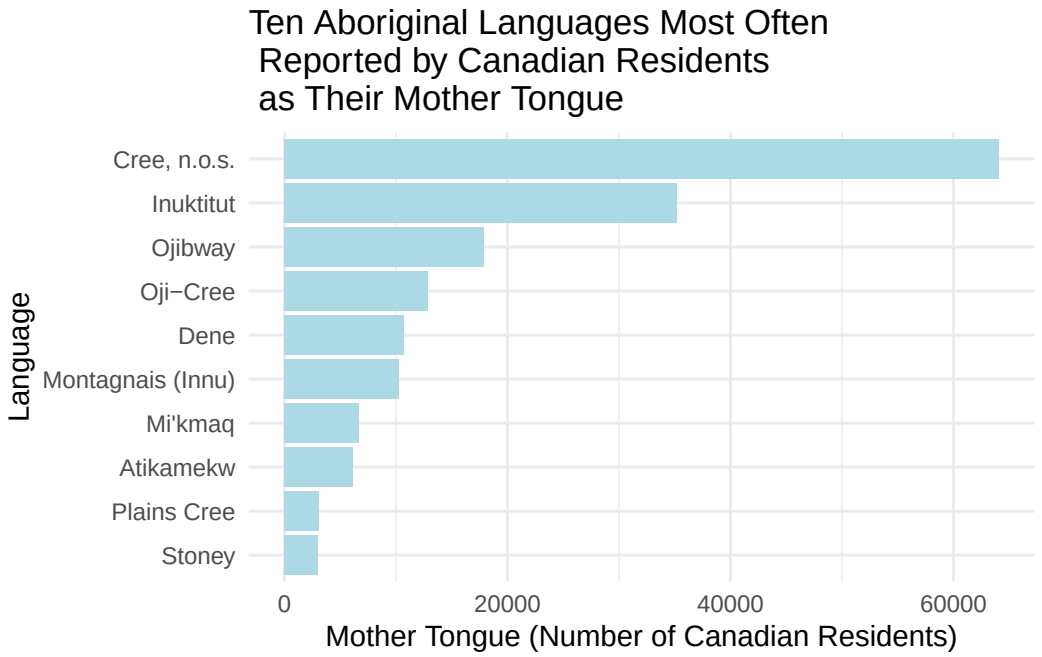

```
#OR

ggplot(ten_lang, aes(x = mother_tongue, y = language)) +
  geom_bar(stat = "identity")
```



Labels, Colors, and Themes

```
ggplot(ten_lang, aes(x = mother_tongue, y = reorder(language, mother_tongue))) +
  geom_bar(fill="lightblue", stat = "identity") +
  ylab("Language") +
  xlab("Mother Tongue (Number of Canadian Residents)") +
  ggtitle("Ten Aboriginal Languages Most Often \n Reported by Canadian Residents \n as The
  theme_minimal()
```



> **Tip**
>
> Barplots are good for displaying one categorical variable and one numeric variable. The number variable could be counts (as above) or they could be averages or totals or maximums or minimums (or many other things!)

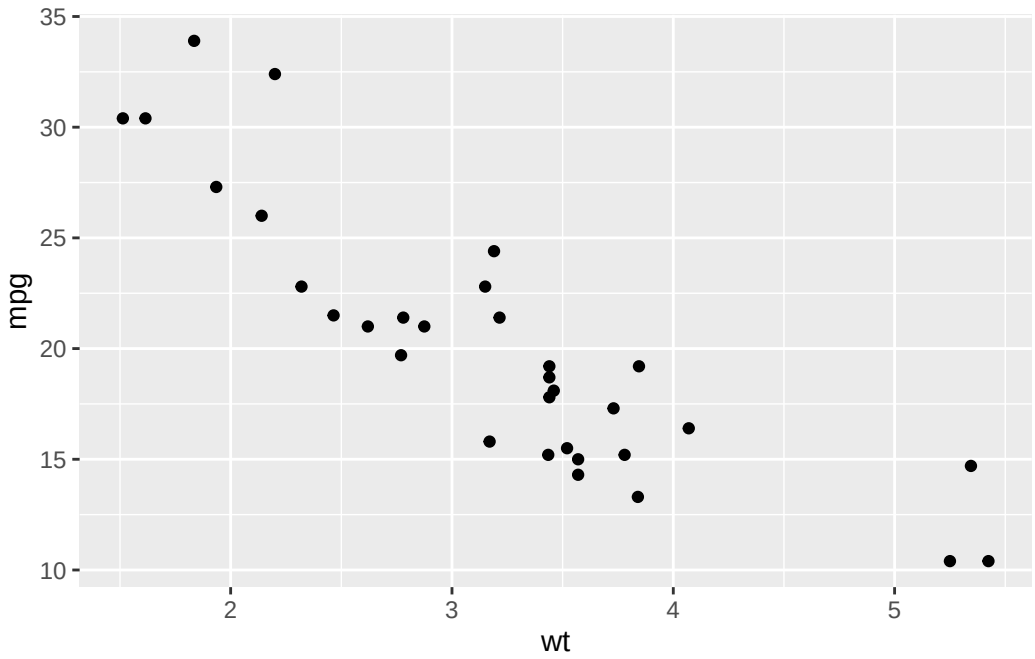
ggplot: scatterplot with `geom_point`

> **Tip**
>
> Scatterplots are good for displaying the relationship between two numerical variables.

The `mtcars` dataset was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973–74 models). It's available inside the `ggplot` package which is already installed.

```
#load the data
data(mtcars)
```

```
ggplot(mtcars, aes(x=wt, y=mpg)) +
  geom_point()
```



Note that you can change the color, shape (`pch` for plotting character) and size of these points!

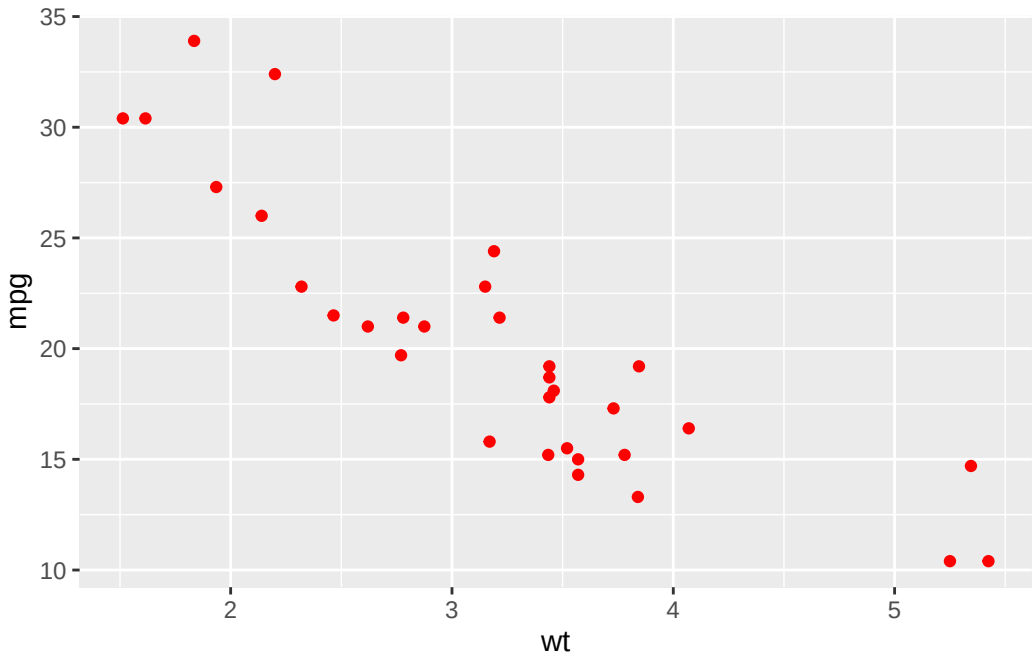
Quick update to the dataset

```
#code to update `mtcars` dataset so that `am` is treated as a factor rather than a continu
mtcars <- mtcars %>%
  mutate(am = as.factor(am))
```

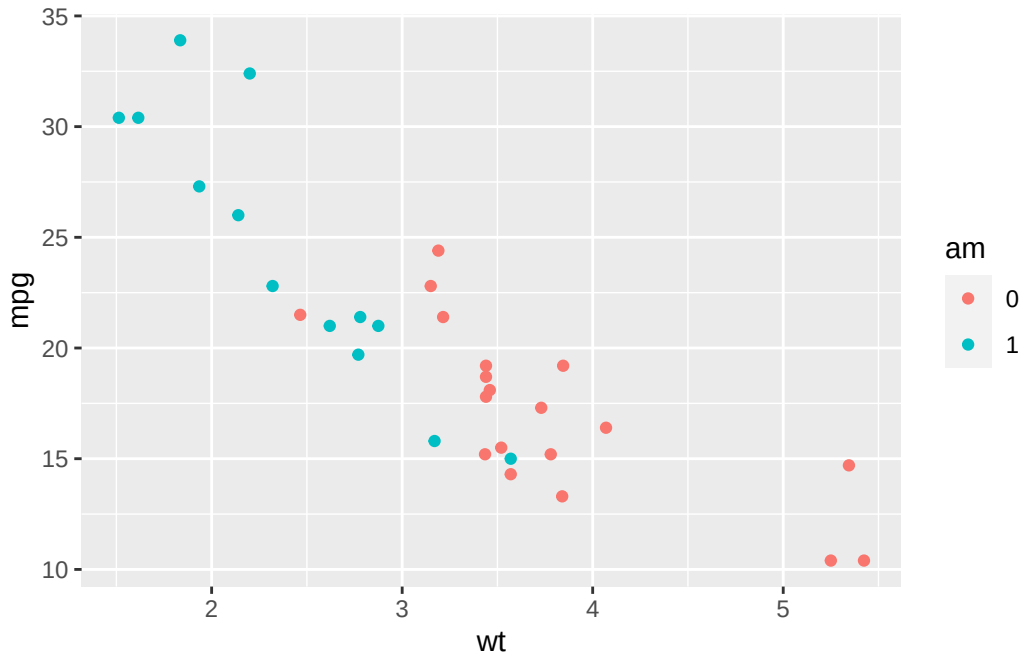
Inside `aes()` or outside `aes()`?

What is the difference between these two graphs?

```
#color not in aesthetics
ggplot(mtcars, aes(x=wt, y=mpg)) +
  geom_point(color="red")
```




```
#color in aesthetics
ggplot(mtcars, aes(x=wt, y=mpg)) +
  geom_point(aes(color=am))
```

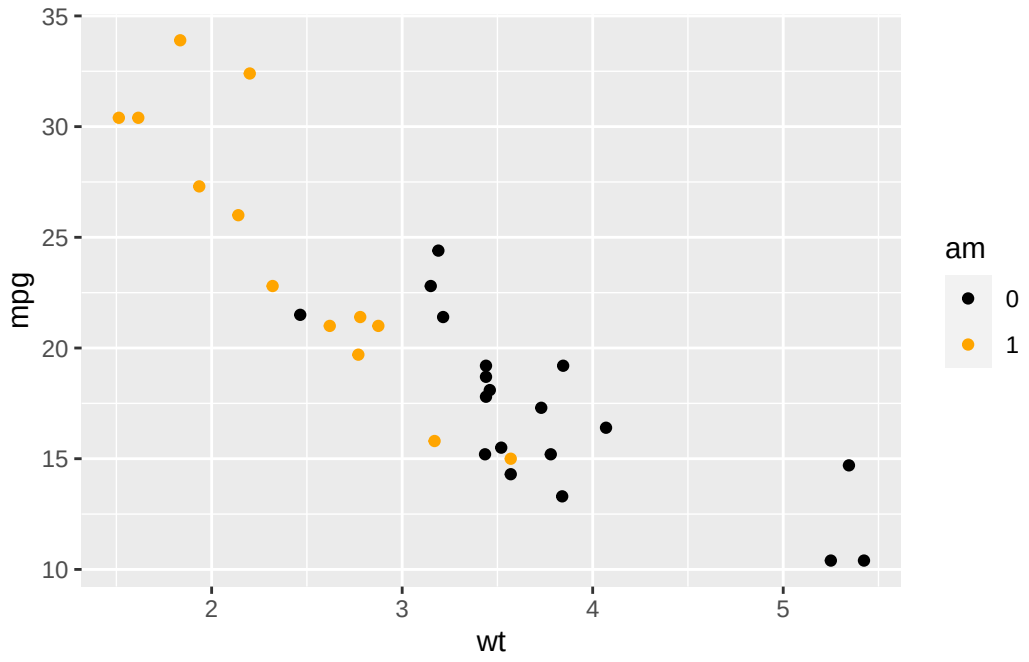


Tip

- If the thing you are trying to change (color, shape, size, etc.) depends on a variable, you should put in *inside* the aesthetics
- If the thing you are trying to change (color, shape, size, etc.) should happen for all things, you should not put it inside the aesthetics.

Customizing Colors in Aesthetics

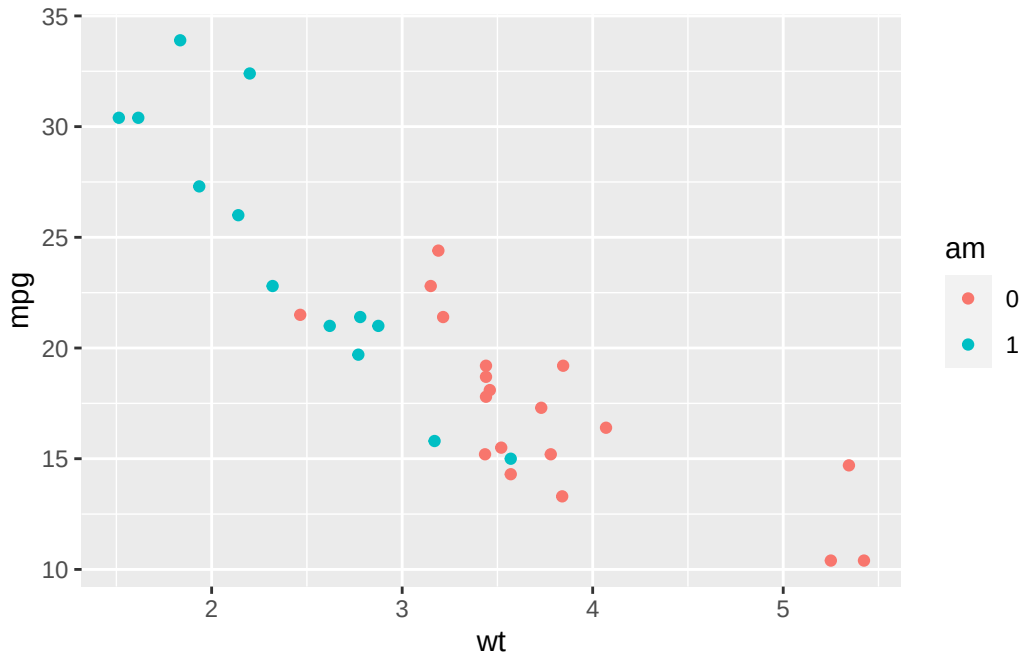
```
#color in aesthetics
ggplot(mtcars, aes(x=wt, y=mpg)) +
  geom_point(aes(color=am)) +
  scale_color_manual(values=c("black", "orange"))
```



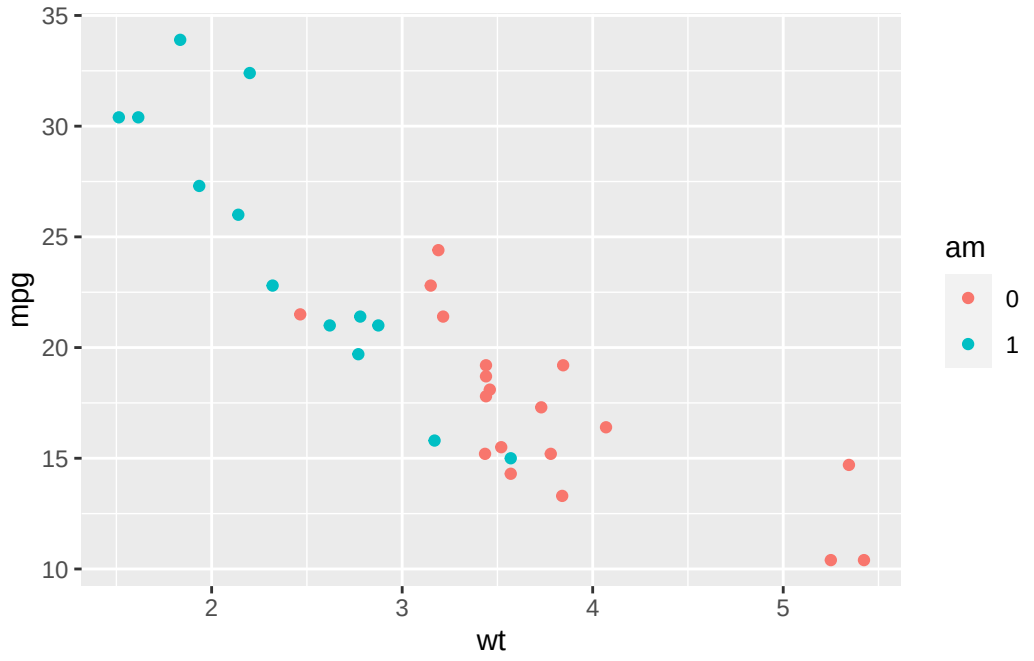
Global vs. Local Aesthetics

Global aesthetic mappings apply to all geometries and can be defined when you initially call `ggplot()`. All the geometries added as layers will default to this mapping. Local aesthetic mappings add additional information or override the default mappings.

```
#color = am as a global aethetic
ggplot(mtcars, aes(x=wt, y=mpg, color=am)) +
  geom_point()
```




```
#color = am as a local aethetic
ggplot(mtcars, aes(x=wt, y=mpg)) +
  geom_point(aes(color=am))
```



```r
#overwriting color = am as a global aethetic with a local aesthetic
ggplot(mtcars, aes(x=wt, y=mpg, color=am)) +
  geom_point(color="purple")
```