Assignment 1

Taylor Winter (Cool student number only for cool people)

2024-07-23

Overview

This is a very brief speed run of some core `tidyverse` functions to use in your first assignment. It does not address your assignment one research question but should contain handy tips.

If you want to see some more detail than what we go into here. Then the most brief resource is the data wrangling cheat sheet:

https://www.rstudio.com/wp-content/uploads/2015/02/data-wrangling-cheatsheet.pdf

If you want a more definitive guide on R, then I would advise Hadley Wickhams text book 'R for Data Science' which is free on his website:

https://r4ds.hadley.nz/

Loading default datasets

Default datasets in R can be loaded simply by calling them with their name. We can take a look at the `mtcars` dataset below.

Some people had questions about the meaning of each variable. If you use `?` infront of the dataset name, RStudio will bring up the appropriate documentation. E.g., `?mtcars`.

```
##       mpg             cyl             disp             hp       
##  Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0  
##  1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5  
##  Median :19.20   Median :6.000   Median :196.3   Median :123.0  
##  Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7  
##  3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0  
##  Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0  
##       drat             wt             qsec             vs        
##  Min.   :2.760   Min.   :1.513   Min.   :14.50   Min.   :0.0000  
##  1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89   1st Qu.:0.0000  
##  Median :3.695   Median :3.325   Median :17.71   Median :0.0000  
##  Mean   :3.597   Mean   :3.217   Mean   :17.85   Mean   :0.4375  
##  3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90   3rd Qu.:1.0000  
##  Max.   :4.930   Max.   :5.424   Max.   :22.90   Max.   :1.0000  
##        am              gear             carb      
##  Min.   :0.0000   Min.   :3.000   Min.   :1.000  
##  1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000  
##  Median :0.0000   Median :4.000   Median :2.000  
##  Mean   :0.4062   Mean   :3.688   Mean   :2.812  
##  3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000  
##  Max.   :1.0000   Max.   :5.000   Max.   :8.000  
```

Selecting columns in your dataframe

Using the `select()` function we can select columns based on their name, their column number, or some other filtering step (see the data wrangling cheat sheet for helper functions if interested).

Recall the pipe function takes whatever we have on the left, and parses it to a function on the right. In this case `mtcars` is piped to the `select()` function.

```
##                    mpg
## Mazda RX4         21.0
## Mazda RX4 Wag     21.0
## Datsun 710        22.8
## Hornet 4 Drive    21.4
## Hornet Sportabout 18.7
## Valiant           18.1
```

Filter rows of data

Next is the `filter()` function which you can use to filter your variables. We can use Boolean expressions or any other type of logical test. In the example below we wish to filter down to only vehicles with mpg greater than 30mpg.

```
##                 mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Fiat 128       32.4   4 78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic    30.4   4 75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla 33.9   4 71.1  65 4.22 1.835 19.90  1  1    4    1
## Lotus Europa   30.4   4 95.1 113 3.77 1.513 16.90  1  1    5    2
```

Mutate

When you need to create a new variable based on some existing variable or simply wish to transform an existing variable, you can use `mutate()`.

```
##                    mpg mpg_100
## Mazda RX4         21.0    2100
## Mazda RX4 Wag     21.0    2100
## Datsun 710        22.8    2280
## Hornet 4 Drive    21.4    2140
## Hornet Sportabout 18.7    1870
## Valiant           18.1    1810
```

Summarise variables or groups of variables

The `summarise()` function allows us to operate over an entire variable. In the example below, I have taken both the mean and standard deviation of the `mpg` variable.

```
##   mpg_mean mpg_sd
## 1    20.09   6.03
```

Group by a discrete or categorical variable

The `group_by()` variable is very powerful and allows us to group by one or more variables, then apply a function on each grouping. The grouping will remain applied to the data frame until we overwrite it with a new grouping or explicitly `ungroup()` the data.

```
## # A tibble: 3 x 2
##     cyl mpg_mean
##   <dbl>    <dbl>
## 1      4      26.7
## 2      6      19.7
## 3      8      15.1
```

Visualise your data

GGplot allows you to produce almost any graph you could imagine. The way it works is by forming a canvas where you lay out where you want each variable. You then start layering up each feature and each layer of styling.

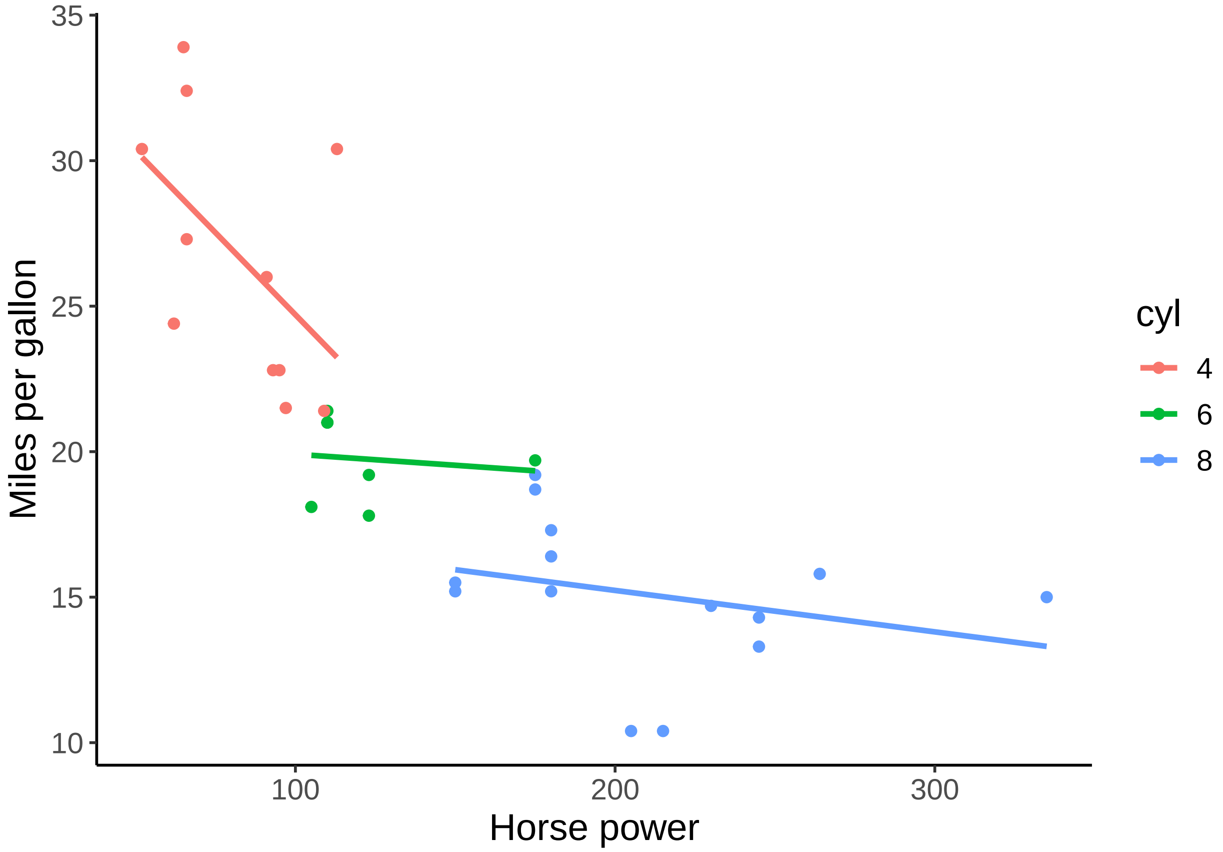
Some examples of what more complex customization looks like in practice can be seen in the Ngāi Tahu state of the nation report where we used GGplot for everything except the maps:

https://ngaitahu.iwi.nz/assets/Documents/State-of-Ngai-Tahu-Nation-2021-web.pdf

The `theme()` function is where you can really customize the heck out of your plots but it gets quite complicated. Note that the labels need to be readable and appropriately named.

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```

Figure 1: Increase in milage based on horsepower and cylinders.


Miles per gallon
Horse power
cyl