R for Educational & Psychological Research

Fernando Rodriguez & Hye Rin Lee

2021-01-09

Contents

Chapter 1

Introduction

We are looking forward to introducing you to the wonderful world of R.

R is a very powerful statistical programming language that has several advantages over using Stata or SPSS. The most obvious is that R is completely free. Other advantages include the ability to work with different datasets simultaneously, good version control features, tools for sharing reproducible code, and an amazing library for visualizing data.

The downside of using R is that it almost exclusively code-based, meaning that there are very limited point-and-click features. In order to get R to perform an analysis or plot a figure, you have to write several lines of code, which can feel like a big barrier if you are unfamiliar with programming concepts.

The goal of this book is to ease you into learning how to program in R. What makes this book different from others is that we assume you have zero experience with using R (and are also a bit intimidated by learning it!), so everything is explained as straightforward as possible. The book will also guide you through the entire process of analyzing educational data obtained from an online STEM course. This includes importing, inspecting, making decisions about your sample size, generating descriptive data, creating data visualizations, and using inferential statistics to draw conclusions from your sample. By the end of this book, you will have the necessary proficiency to use R on your own research project from start to finish.

As you work your way through the chapters, you will find that programming in R is much easier than it looks. Even more exciting, once you get a good sense of some of the basics, you will soon begin tinkering with code and trying things just for the sake of trying things out. That's where the real fun starts.

1.1 Who is this book for?

We wrote this book for people who do education and/or psychological research, who are at various levels in their careers, and who want a easy-to-follow book for learning R. This includes undergraduate lab assistants who are new to research methods and statistics and are just starting off on their journey in research, and also graduate students, faculty, and those working in education industry who have solid research experience, know how to use SPSS or Stata, but want to branch out to and further expand their skills.

While no experience with R or coding is necessary, we do assume that you have a basic understanding of research methods and statistics.

1.2 Structure of the book

This book consists of three parts. `Part I` walks you through installing R, which is the actual program we need to have open in order to run code in R, and R-Studio, which is a graphical user interface (GUI) that helps us better manage our project files and datasets. We will then walk you through the most basic concepts surrounding the R programming language, as well as popular libraries. Libraries refer to a suite of features we can use in R, but are not part of the main R program. Finally, because we want you to see the immediate appeal of using R, you will also write your first data visualization code.

`Part II` will guide you through importing, inspecting, and exploring your data. It is here you will learn all about the 'tidy' method for working with data. This 'tidy' method was developed by Hadley Wickham along with the R-Studio team, and it refers to a principles for working with data.

`Part III` will help you understand how to conduct inferential statistics, and…..

1.3 Additional resources

While is this book provides a general introduction to using R, we don't cover everything we think you should know about R, so we recommend that you refer the following books:

1.4 About the authors

Fernando Rodriguez, PhD., is an assistant professor of teaching in the School of Education at the University of California, Irvine. He enjoys teaching various undergraduate-level courses and the graduate-level statistics course in the School of Education. His research focuses on learning analytics and higher-order

thinking skills. Dr. Rodriguez earned his B.A. in Psychology from California State University, Northridge. He earned his Master's degree in Developmental Psychology and his Ph.D. in Educational Psychology from the University of Michigan, Ann Arbor.

Hye Rin Lee is a doctoral student in the School of Education at the University of California, Irvine. Her work examines…. She also helps organize and lead workshops in R ….

1.5 Acknowledgments

We would like to thank….

Chapter 2

Grounding Concepts

2.1 A Quick tour of R and R-Studio

Please watch the first 15 minutes of the Lesson 01 workshop video, as we give you a run down of these programs what all of these panes, tabs, and buttons do.

Click here to watch Lesson 01

2.1.0.1 *What's the difference between R and R-Studio?*

Both programs are able to run R.

`R` is the original program. When we install R, we are installing two things: (1) the R programming language, and (2) a Graphical User Interface (GUI) that helps us work with the R-programming language, such as running code, opening and saving files. When you open R, you will notice that the GUI contains only a few buttons and icons. R looks very simple at first sight, but it does have all of the necessary tools you would need to work with data.

`R-Studio` is an add-on to R. It has many additional features that make working with the R-programming language much easier. As you can see from opening R-Studio, the GUI has several different panes, buttons, tabs, and icons.

R-Studio also has important tools for open science practices, like the ability to create notebooks that replicate your data analyses. It also has a tool for uploading your work to code-sharing platforms, like Github.

But because R-Studio is an add-on program, you need to ensure R is installed on your computer. Note that when you open R-Studio, it already imports the R-programming language, so there is no need to open the R program.

2.1.1 Using R-Studio exclusively

We will only be using R-Studio for this book.

2.2 Running and saving code in R-Studio

The Console Pane, which by default appears on the bottom-left pane in R-Studio, is the interface for entering and running code.

For the purpose of your workshop, we will use the console to do quick data calculations, data checks, and experiment with code. We will primarily use R-Studio to write and execute our code, especially as it pertains to creating a data notebook.

2.2.1 R-Markdown Files

R-Markdown files is a document file that serves as a data notebook, where we can write text as well as lines of code. The benefit of using an R-Markdown file is that we can keep a record of everything we do, from importing our data, inspecting and cleaning variables, to analyzing and visualizing our data. This allows us to share our work with others in a completely transparent way. R-Markdown files do have some characteristics that look quite odd, but we'll address those in a bit.

2.2.2 Creating a New R-Markdown File

On the top-left corner in R-Studio, select File -> New File -> R-Markdown

2.2.3 R-Markdown Magic: Code Chunks

Let's get familiar with the concept of Code Chunks. Code chunks

This is specific feature of markdown (.Rmd) files, meaning that

2.3 Creating code chunks in R-Markdown

There are three ways to create a Code Chunk.

1. You can create a new chunk of code by typing the following:

   “‘`{r}`

   CODE HERE

```

The space in between the grave markers, "`{r}` CODE HERE" is where we can write R code and calculations, such as 2 + 2.

2. You can insert a code chunk using the +c menu button, which appears directly above your R-Markdown document. To create a code chunk, click on the +c button, then select R.
3. You can also use the following keyboard shortcut
   - alt + command + i (mac)
   - control + alt + i (windows)

```
2 + 2
```

```
## [1] 4
```

2.4 Running chunks of code

Now that you entered 2+ 2 in the code chunk, you can run this line of code by clicking on the green arrow to the right of the code chunk.

You can also use the following shortcuts to run the code within this code chunk:

command + enter (mac)

control + enter (windows)

2.5 The very basics

2.5.1 Simple Calculations

R works just like a calculator. You can do addition, subtraction, multiplication, etc. Here, we provide two examples, but you can experiment with calculations (+, -, *, /, ^, etc.) on the Console Pane.

Addition

```
2 + 2
```

```
## [1] 4
```

Division

```
10/2
```

```
## [1] 5
```
```

2.5.2 Objects & the Assignment Operator <-

Objects are the virtual space where we can temporarily store the data we load into R. When we want to load a .csv file into R, for example, we save it into an object. We can name these objects whatever we like, as long as it starts with a character string and does not contain special words or special characters that are exclusive to specific R commands or functions (more on this in later chapters).

Remember the simple calculations we just did? We can store those results into an object.

We do this by using the assignment operator `<-`

The assignment operator is an arrow `<-` (which is the `less than` sign and the `dash` sign). This is also what we mean by special characters—you cannot use `<-` for any other purpose in R.

Here's how it works.

Lets creating objects a, b, and c

```
a <- 2
b <- 10 + 2
c <- 2 + 2
```

2.5.3 Environment Pane in R-Studio

Notice that something happened to the environment pane. The environment name shows you the names of the objects we created. You will also see that the stored values are displayed to the right of the object name.

You may have also noticed that the results of a, b, and c, did not show up anywhere other than the environment pane. This is because when we use the assignment operator, we are telling R to save the results (and not displaying them).

2.6 Learning your first function: `print()`

```
print(a)
```

```
## [1] 2
```

2.7 Objects & Functions

2.7.1 The Data Frame Object

here, I we are going to type `mtcars` in the code chunk below which is a dataframe that came pre-installed in R.

```
mtcars
```

```
##                   mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout  18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Duster 360        14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 240D         24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230          22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280          19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C         17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Merc 450SE        16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL        17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC       15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial  14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## 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
## Toyota Corona     21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Dodge Challenger   15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin       15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28        13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird   19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Fiat X1-9         27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2     26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa      30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Ford Pantera L    15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Ferrari Dino      19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Maserati Bora     15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Volvo 142E        21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

2.7.2 Learning your first function in R: `str()`

If you want to see less rows you can use the `head()` function.

```
head(mtcars)
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4        21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag    21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710       22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive   21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant          18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

If you want to move the `mtcars` dataframe into the environment pane, you can duplicate it via the assignment command. Here, we'll save a copy of `mtcars` as `cars` and check the data using the `head()` function. Notice that I just added into the same chunk of code.

```
cars <- mtcars

head(cars)
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4        21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag    21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710       22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive   21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant          18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

2.8 View()

This allows us to view the actual raw data

```
View(mtcars)
```

2.9 A Note on Arguments

Notice that functions in R always have () beside them

head(mtcars)

In R, we put our arguments (which are things the function needs to run, and/or extra things we want the function to do) inside these parentheses.

Chapter 3

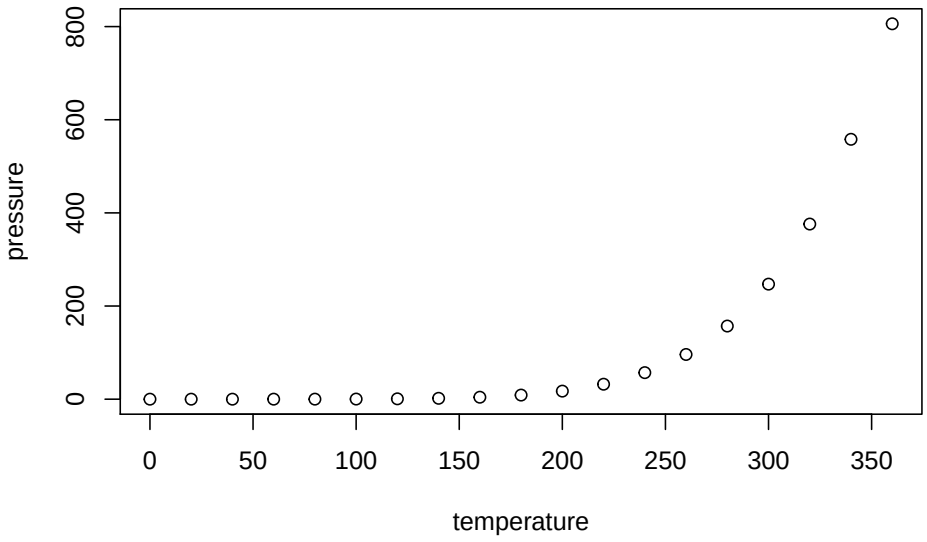
Building Your First Data Visualization

Chapter 4

Intro to plotting and the ggplot library

We can use the `plot` function to create a scatter plot for the `pressure` data

```
plot(pressure)
```



4.1 Loading Libraries

First, let's load the libraries you will use for this lesson. This is the first thing you should do when writing an R-Markdown document. That way, you ensure that you load all of the necessary libraries prior to running code

```
# install.packages("ggplot2")
library(ggplot2)
```

Remember to make sure ggplot is loaded into R.

You can do this by running the first chunk of this document, where it has the code `library(ggplot2)`

for ggplot, our first argument will be the dataset `mtcars`

```
ggplot(mtcars)
```

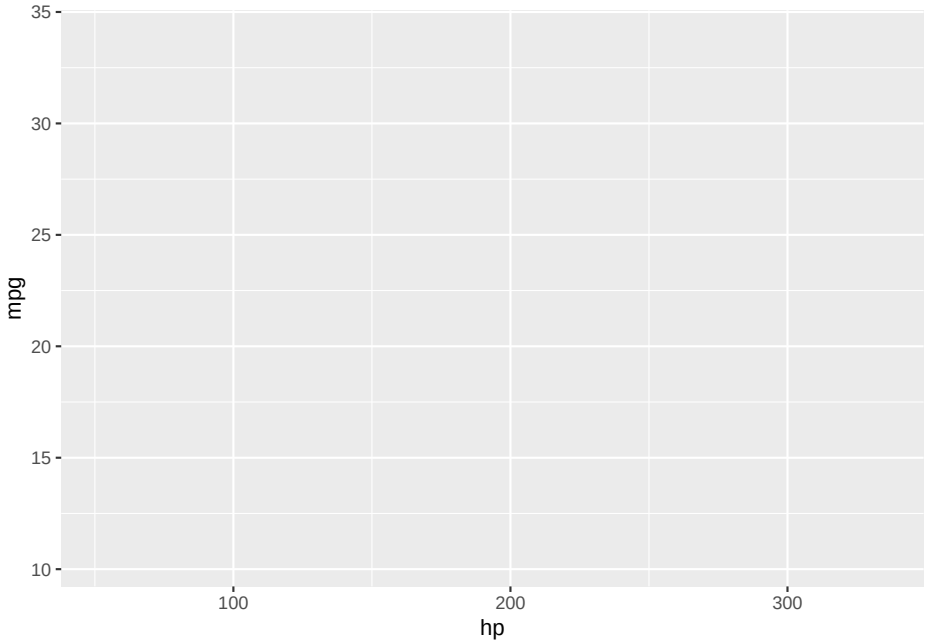

4.2 aes() function for stating your x and y axis

Within the ggplot() function, we'll set up our parameters by using the `aes()` function aes stands for asthetic. For this function, we want to define the x and y axis.

We want to plot how miles per galon `mpg` is related to horsepower `hp`. And then we want to split this up by cylnders `cyl`.

The x-axis will be `hp` and the y-axis will be `mpg`

```
ggplot(mtcars, aes(x = hp, y = mpg))
```



4.3 The power of +

4.4 Adding features to your ggplot graph using +

We can add new features by using other functions that are part of the ggplot library.

We do this by using the + sign

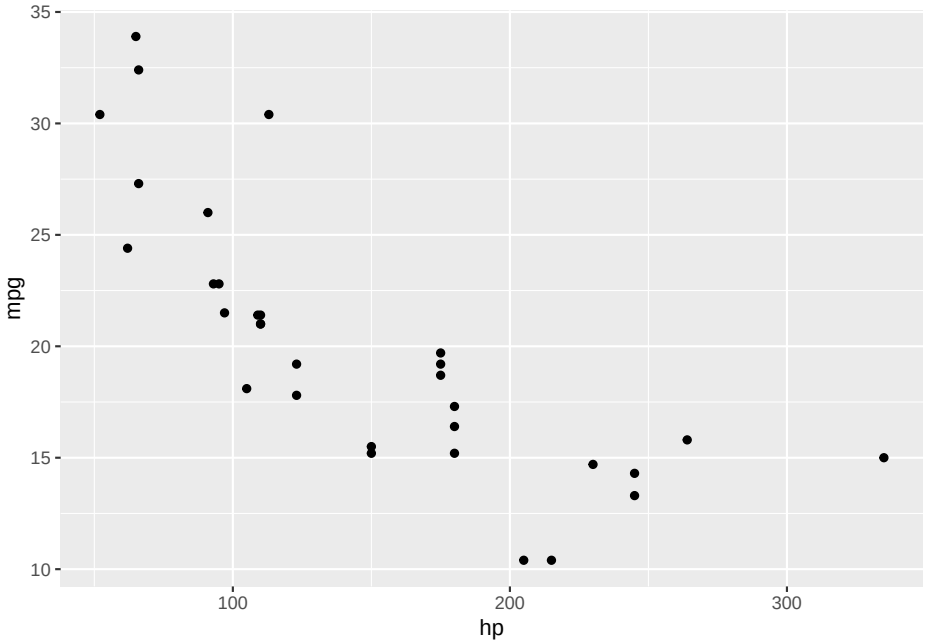
`geom_()` function, which is our geometric object, is used to state the kind of graph we want

Now let's use the `geom_()` family of functions to state what kind of graph we want.

We want a scatterplot, so we are going to use the function `geom_point()`

No arguments are required for `geom_point()`

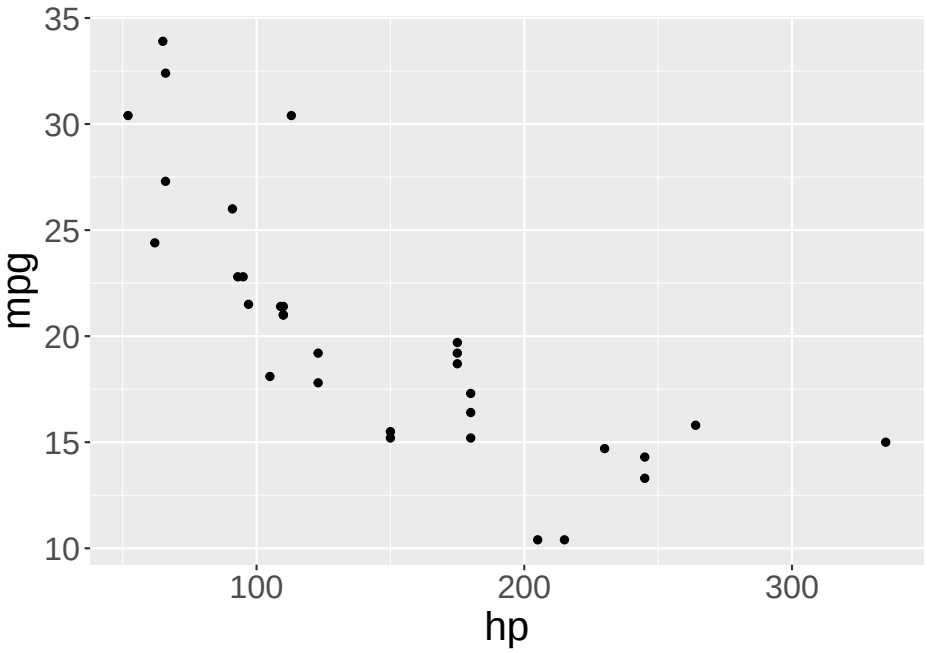
```
ggplot(mtcars, aes(x = hp, y = mpg)) +
  geom_point()
```



4.5 theme() function for modifying components of your graph

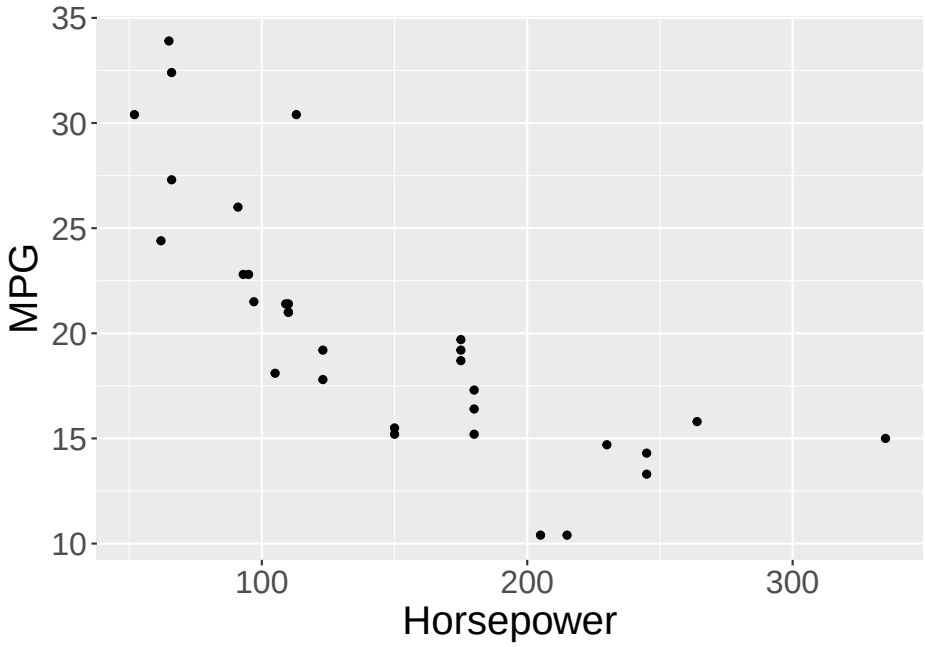
let's set the size of the text by 20 using theme(`text = element_text(size = 20)`)

```
ggplot(mtcars, aes(x = hp, y = mpg)) +
  geom_point() +
  theme(text = element_text(size = 20))
```



4.6 labs() function for labeling your graph

```
ggplot(mtcars, aes(x = hp, y = mpg)) +
  geom_point() +
  theme(text = element_text(size = 20)) +
  labs(x = "Horsepower", y = "MPG")
```



4.7 scale_color_gradient() function for using a color gradient on mpg

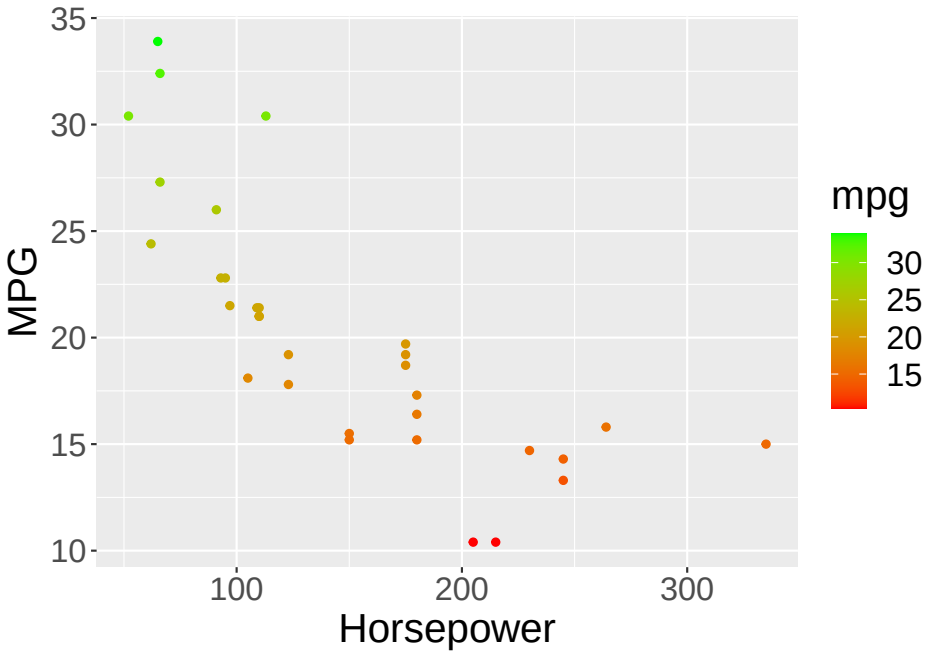
We want the low mpg to be blue and the high mpg to be red.

IMPORTANT! In order to make this function work, you have to state which variable you want it to color. Let's color `mpg`.

In order to state this, we have to go back to the `aes()` function and write an additional argument. Remember, arguments are separated by `,`

So your aes code should look like this now `aes(x = hp, y = mpg, color = mpg)`

```
ggplot(mtcars, aes(x = hp, y = mpg, color = mpg)) +
  geom_point() +
  theme(text = element_text(size = 20)) +
  labs(x = "Horsepower", y = "MPG") +
  scale_color_gradient(low = "red", high = "green")
```



4.8 facet_wrap() function splitting up the graph by a group

We want to split our graph up by the variable `cyl`

use `~cyl` to wrap by columns

use `ncol =` to specify how many columns you want in your layout

4.9 Learning More About Libraries and Functions

If you want to see more information about what you can do with a library like ggplot2, you can put `?` in front of the name of the library.

If you want to know more about how to use a specific function put a ? in front of the function name

You can even do this with sub-functions, like `element_text()`

Chapter 5

Project Workflow

For this lesson, we'll understand how to develop an organized workflow.

Chapter 6

Inspecting Data

Chapter 7

Data cleaning

Chapter 8

Descriptive tables

Chapter 9

Visualization