

R for Educational & Psychological Research

Fernando Rodriguez & Hye Rin Lee

2021-01-03

Contents

1	Introduction	1
1.1	Who is this book for?	2
1.2	Structure of the book	2
1.3	Additional resources	2
1.4	About the authors	2
1.5	Acknowledgments	3
2	Your first plot	5
2.1	Loading Libraries	5
2.2	Creating code chunks in R-Markdown	5
2.3	Running chunnks	5
2.4	Assignment	6
2.5	Creating d, which is <code>a + b</code>	6
2.6	Exploring Data	6
2.7	<code>mtcars</code> data	6
2.8	<code>pressure</code> data	7
2.9	Graphing Data	8
2.10	A Note on Arguments	8
2.11	<code>aes()</code> function for stating your x and y axis	9
2.12	The power of <code>+</code>	10
2.13	Adding features to your ggplot graph using <code>+</code>	10
2.14	<code>theme()</code> function for modifying components of your graph	11
2.15	<code>labs()</code> function for labeling your graph	12
2.16	<code>scale_color_gradient()</code> function for using a color gradient on mpg	13
2.17	<code>facet_grid()</code> function splitting up the graph by a group	14
2.18	Learning More About Libraries and Functions	16

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 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

Your first plot

2.1 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") <- installing libraries require double quotations ""  
# install.packages("rmdformats")  
library(ggplot2) # <- loading libraries doesn't require double quotations  
library(rmdformats) # for displaying the code in various html styles
```

Before going into the ins-and-outs of coding in R, let's get familiar with how we can use R-markdown to create chunks of code.

2.2 Creating code chunks in R-Markdown

alt + command + i (mac) control + alt + i (windows)

Each chunk of code can be run individually.

```
3 + 3
```

```
## [1] 6
```

2.3 Running chunks

You can run the chunk of code by clicking on the green arrow to the right of the code.

Or use these shortcuts

command + enter (mac) control + enter (windows)
The very basics ##\$ Simple Calculations

```
2 + 2
```

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

```
10/2
```

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

2.4 Assignment

We assign values to variables by using `<-` Assigning `a`, `b`, and `c`

```
a <- 2 + 2
```

```
b <- 10 + 2
```

```
c <- 4 + 12
```

2.5 Creating `d`, which is `a + b`

```
d <- a + b
```

```
d
```

```
## [1] 16
```

2.6 Exploring Data

2.7 `mtcars` data

This data is pre-installed in R, and provides a quick way to fiddle around with coding.

We can see the data by simply typing `mtcars`

```
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
```

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

```
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
```

2.8 pressure data

We can see that there are only two variables in the pressure data

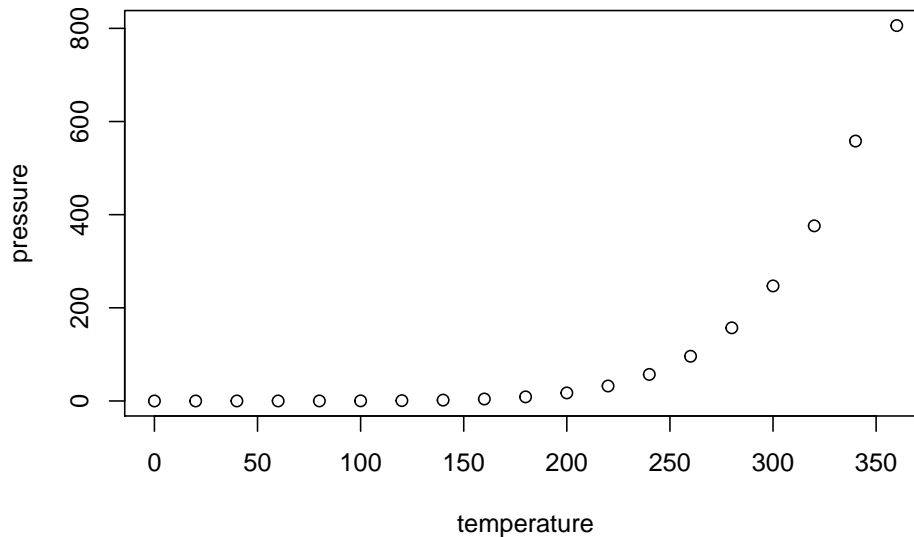
```
head(pressure)
```

```
##   temperature pressure
## 1           0  0.0002
## 2          20  0.0012
## 3          40  0.0060
## 4          60  0.0300
## 5          80  0.0900
## 6         100  0.2700
```

2.9 Graphing Data

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

```
plot(pressure)
```



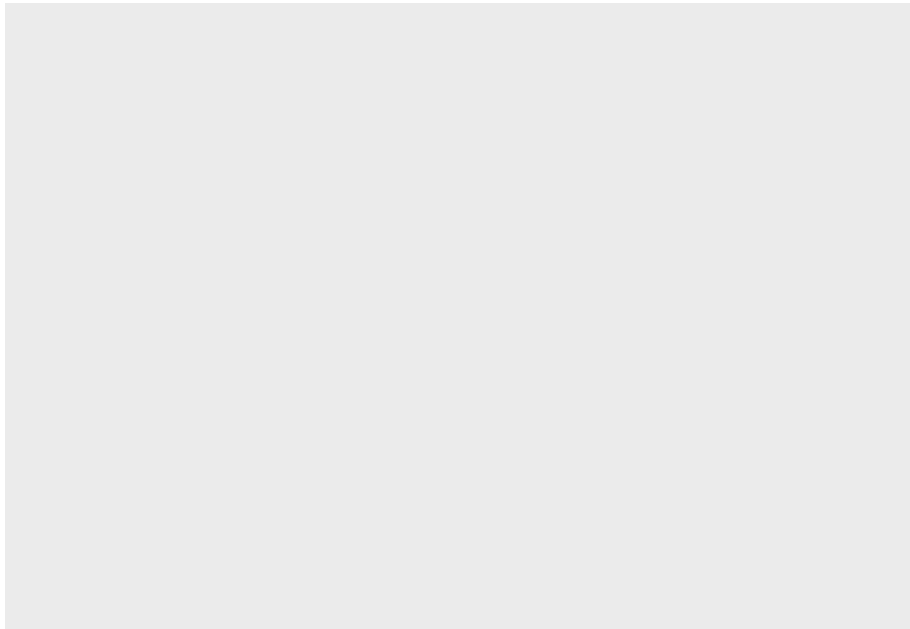
2.10 A Note on Arguments

Notice that functions in R always have `()` beside them `head(mtcars)` `plot(pressure)` In R, we put our arguments (which are extra things we want the argument to do) inside these parentheses.

Intro to ggplot library and functions 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)` We want to plot how miles per gallon `mpg` is related to horsepower `hp`. And then we want to split this up by cylinders `cyl`.

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

```
ggplot(mtcars)
```

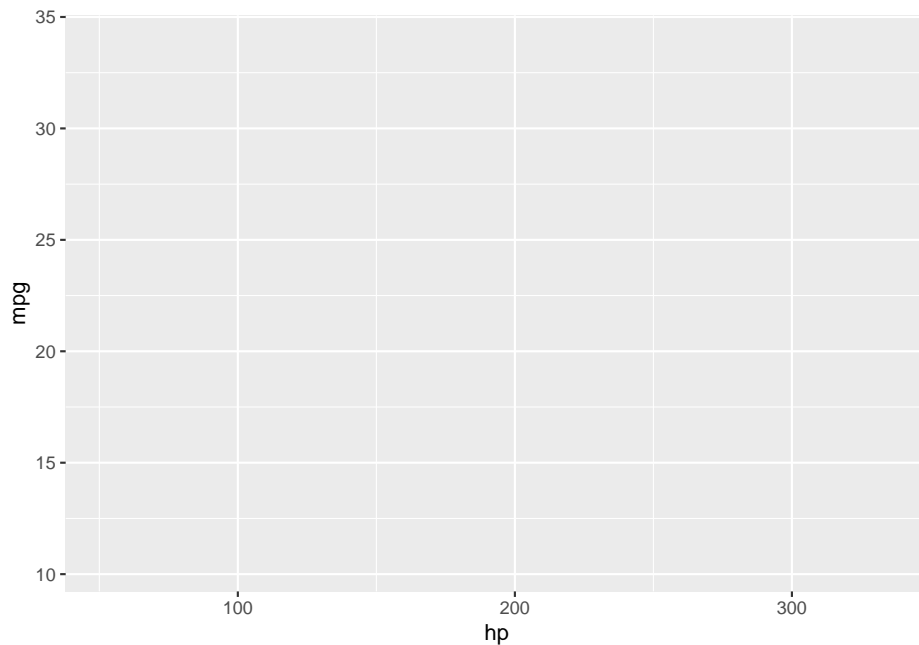


2.11 aes() function for stating your x and y axis

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

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

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



2.12 The power of +

2.13 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 for stating the kind of graph you want
```

geom stands for geometric unit.

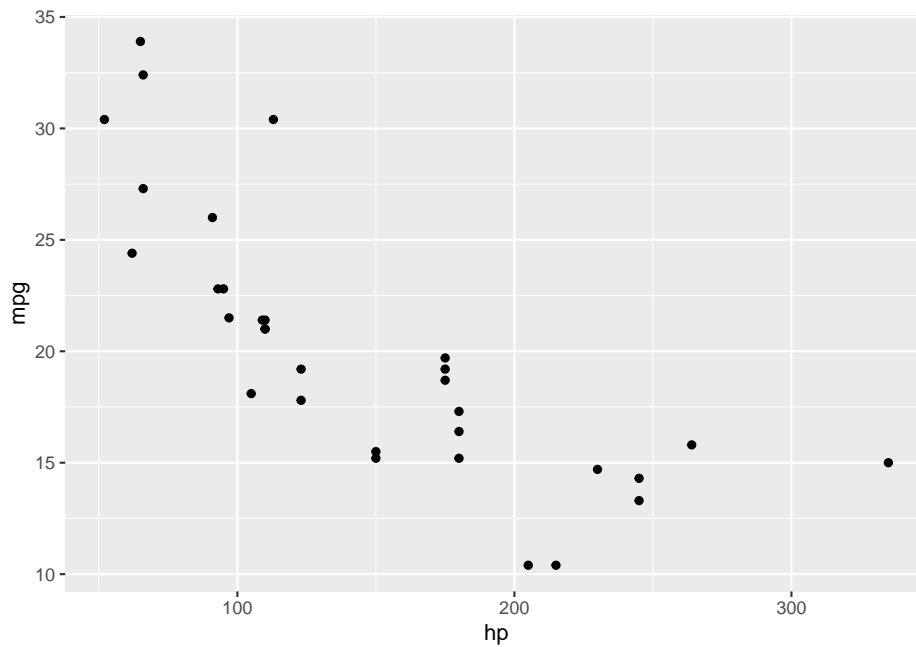
Now let's use the **geom** family of functions to state what kind of graph we want.

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()
```

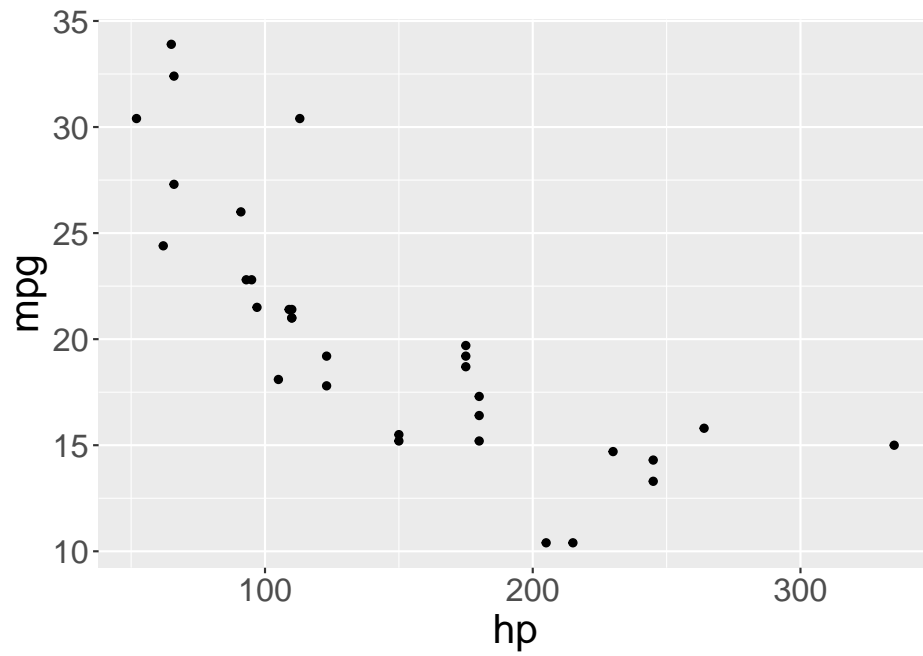
2.14. `THEME()` FUNCTION FOR MODIFYING COMPONENTS OF YOUR GRAPH



2.14 `theme()` function for modifying components of your graph

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

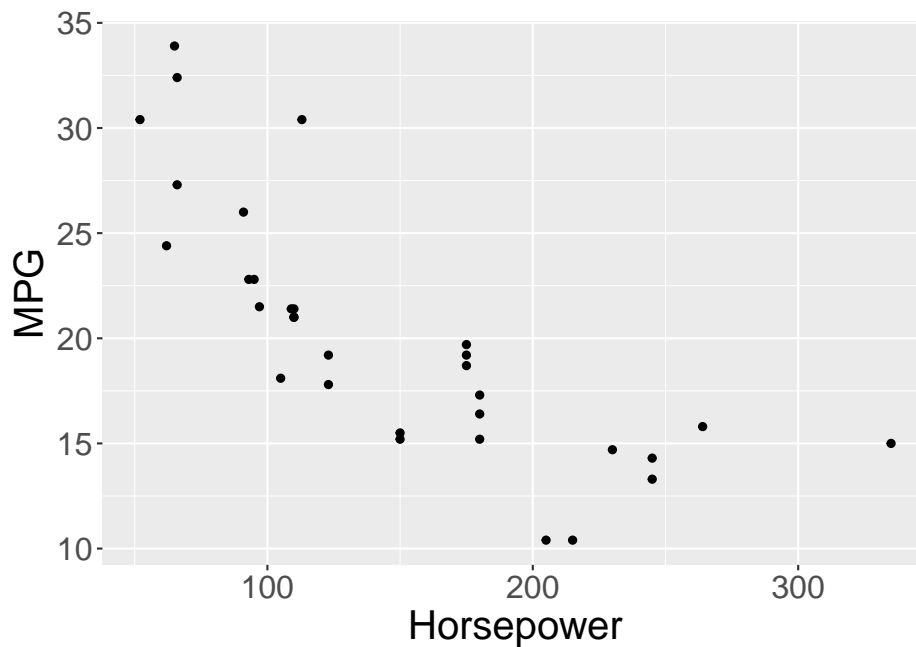
```
ggplot(mtcars, aes(x = hp, y = mpg)) +  
  geom_point() +  
  theme(text = element_text(size = 20))
```



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

2.15 `labs()` function for labeling your graph

2.16. `SCALE_COLOR_GRADIENT()` FUNCTION FOR USING A COLOR GRADIENT ON MPG13



2.16 `scale_color_gradient()` function for using a color gradient on mpg

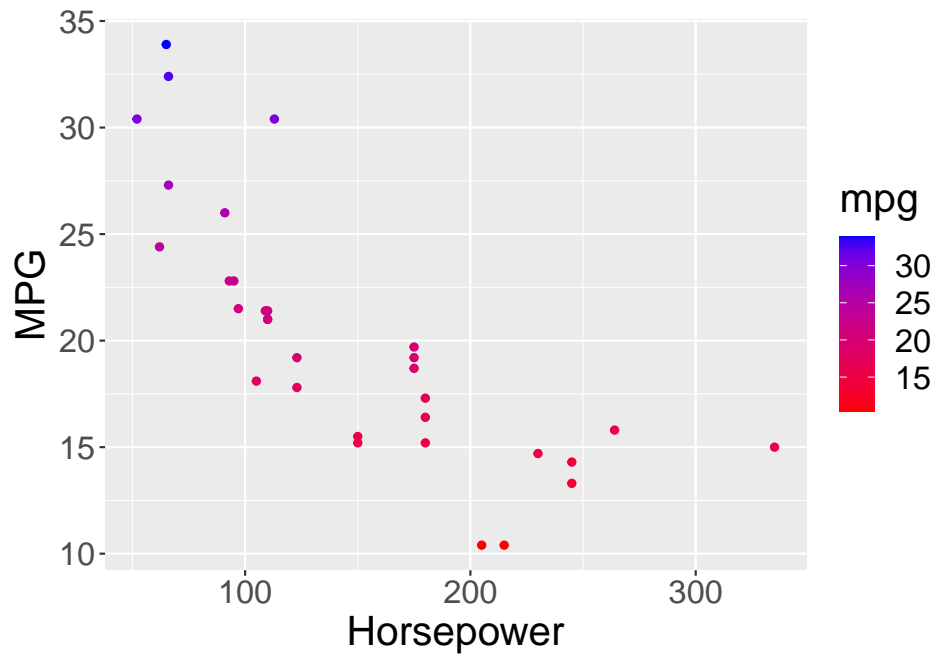
We want the low mpg to be blue and the high mpg to be red.

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 = "blue")
```



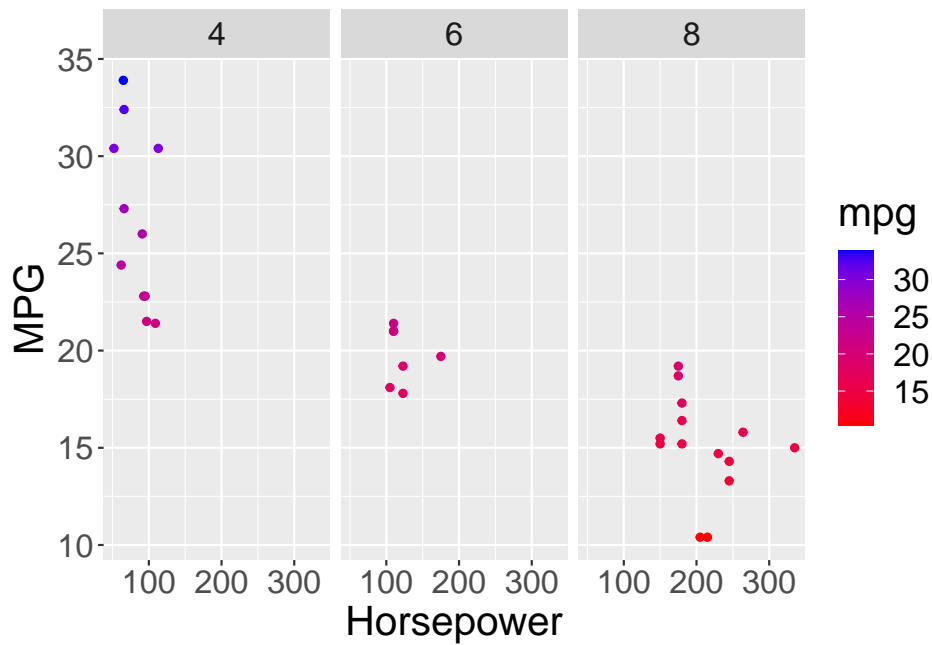
2.17 facet_grid() function splitting up the graph by a group

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

use `~cyl` to get a horizontal version

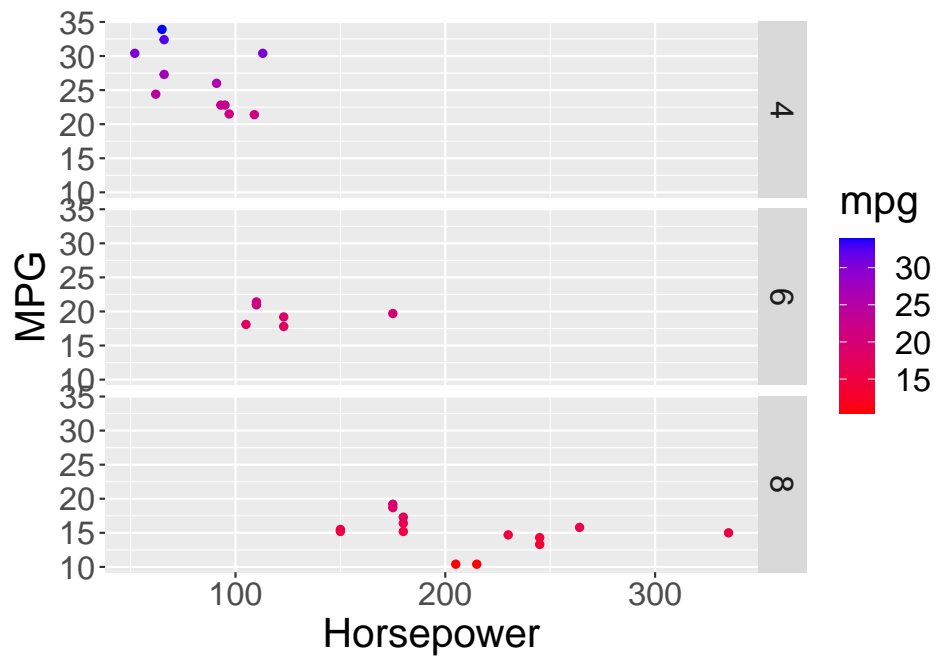
```
ggplot(mtcars, aes(x = hp, y = mpg, color = mpg)) +
  geom_point() +
  theme(text = element_text(size = 20)) +
  labs(x = "Horsepower", y = "MPG") +
  scale_color_gradient(low = "red", high = "blue") +
  facet_grid(~cyl)
```

2.17. `FACET_GRID()` FUNCTION SPLITTING UP THE GRAPH BY A GROUP¹⁵



use `cyl~.` to get a vertical version.

```
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 = "blue") +  
  facet_grid(cyl~.)
```



2.18 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.

```
?ggplot2
```

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

```
?ggplot()
```

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

```
?element_text()
```