# Note with R4DS

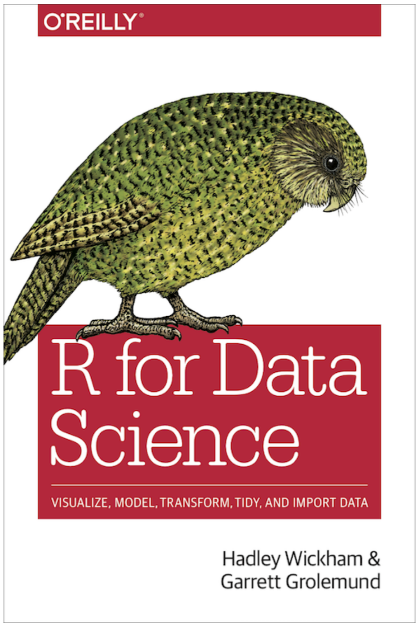
*2019-05-18*

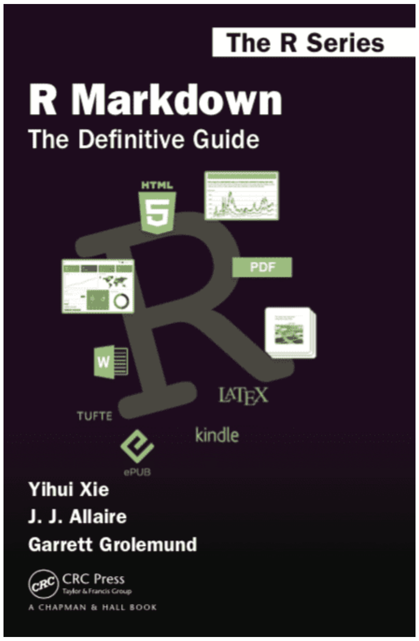
# Contents

# About this notebook

This notebook is my practice after reading those books:

- **R for Data Science**
- **R Markdown: The Definitive Guide**
- **bookdown: Authoring Books and Technical Documents with R Markdown**





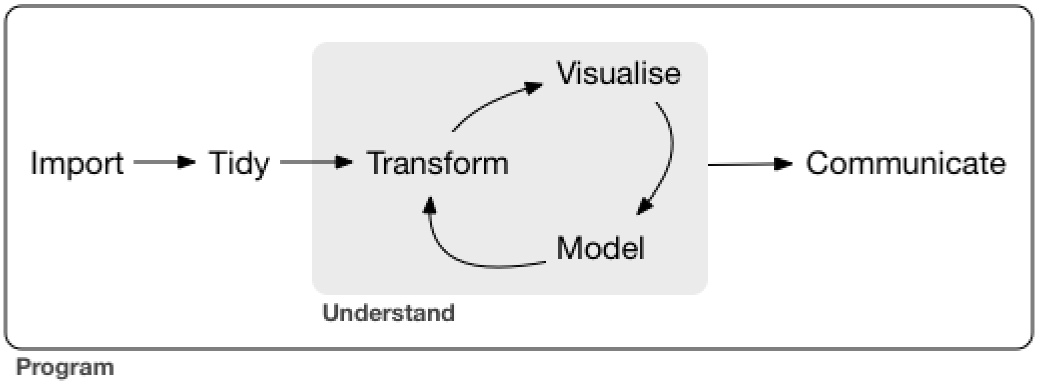
# Chapter 1

# Introduction

Data science is an exciting discipline that allows you to turn raw data into understanding, insight, and knowledge. The goal of "R for Data Science" is to help you learn the most important tools in R that will allow you to do data science. After reading this book, you'll have the tools to tackle a wide variety of data science challenges, using the best parts of R.

## 1.1 Data Science Project process:

Data science is a huge field, and there's no way you can master it by reading a single book. The goal of this book is to give you a solid foundation in the most important tools. Our model of the tools needed in a typical data science project looks something like this:



## 1.2 What you won't learn

- Big Data
- Python, Julia, and friends
- Non-rectangular data
- Hypothesis confirmation

## 1.3 Prerequisites

### 1.3.1 R

- Download R from CRAN: https://cran.r-project.org
- Cloud mirror: https://cloud.r-project.org (which automatically figures it out for you.)

### 1.3.2 RStudio

- Download and install it from http://www.rstudio.com/download
- RStudio IDE Cheat Sheet: https://www.rstudio.com/resources/cheatsheets/#ide

### 1.3.3 The tidyverse packages

Install the tidyverse packages:

```
if (!require("tidyverse")) install.packages("tidyverse")
```

Load it with the library() function:

```
library(tidyverse)
## Registered S3 methods overwritten by 'ggplot2':
##   method         from 
##   [.quosures     rlang
##   c.quosures     rlang
##   print.quosures rlang
## -- Attaching packages ---------------------------------- tidyverse 1.2.1 --
## v ggplot2 3.1.1     v purrr   0.3.2
## v tibble  2.1.1     v dplyr   0.8.1
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
## -- Conflicts ------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Update the packages:

```
tidyverse_update()
```

### 1.3.4 Other packages

In this book we'll use three data packages from outside the tidyverse:

```
install.packages(c("nycflights13", "gapminder", "Lahman"))
```

# Chapter 2

# Introduction

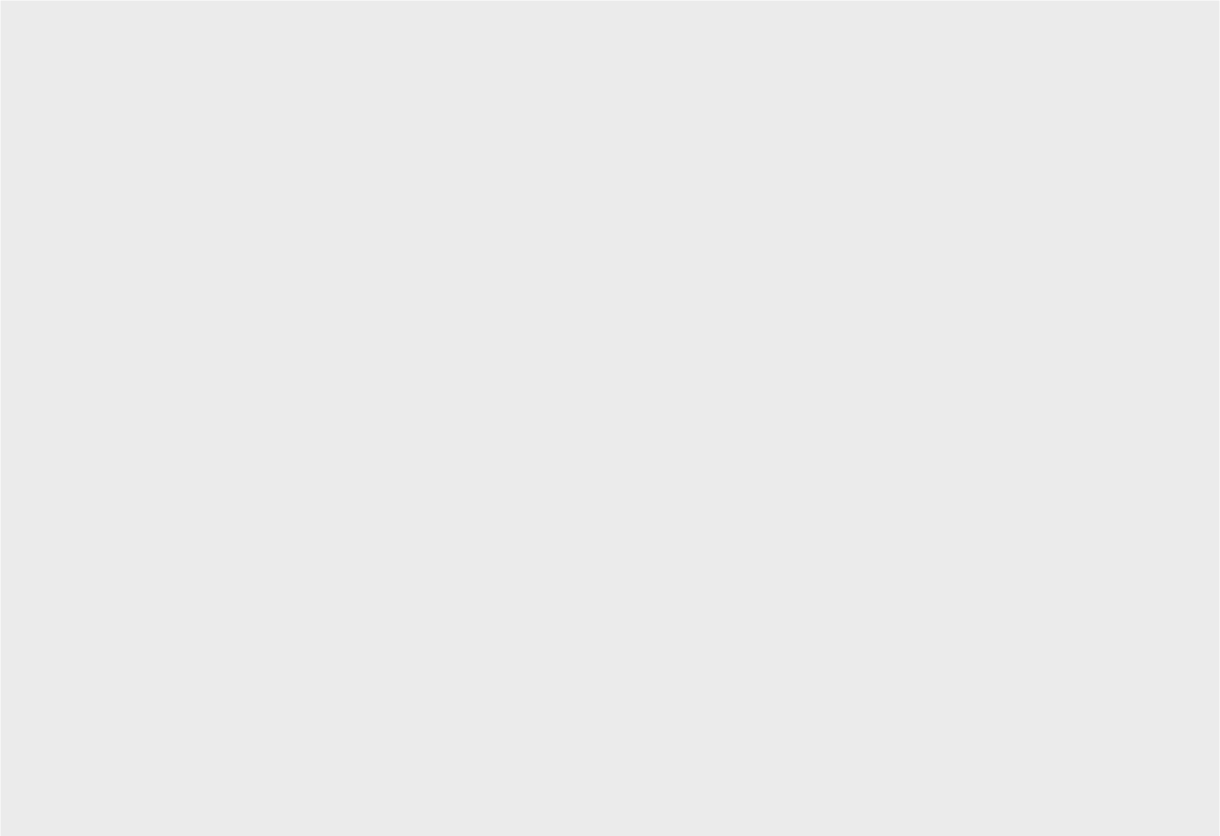
# Chapter 3

# Data visualisation

## 3.2.4 Exercises

1. Run ggplot(data = mpg). What do you see?

```
ggplot(data = mpg)
```



empty graph, because we don't set the aesthetic mapping for plot.

2. How many rows are in mpg? How many columns?

```
dim(mpg)
## [1] 234  11
```

rows: 234, columns: 11

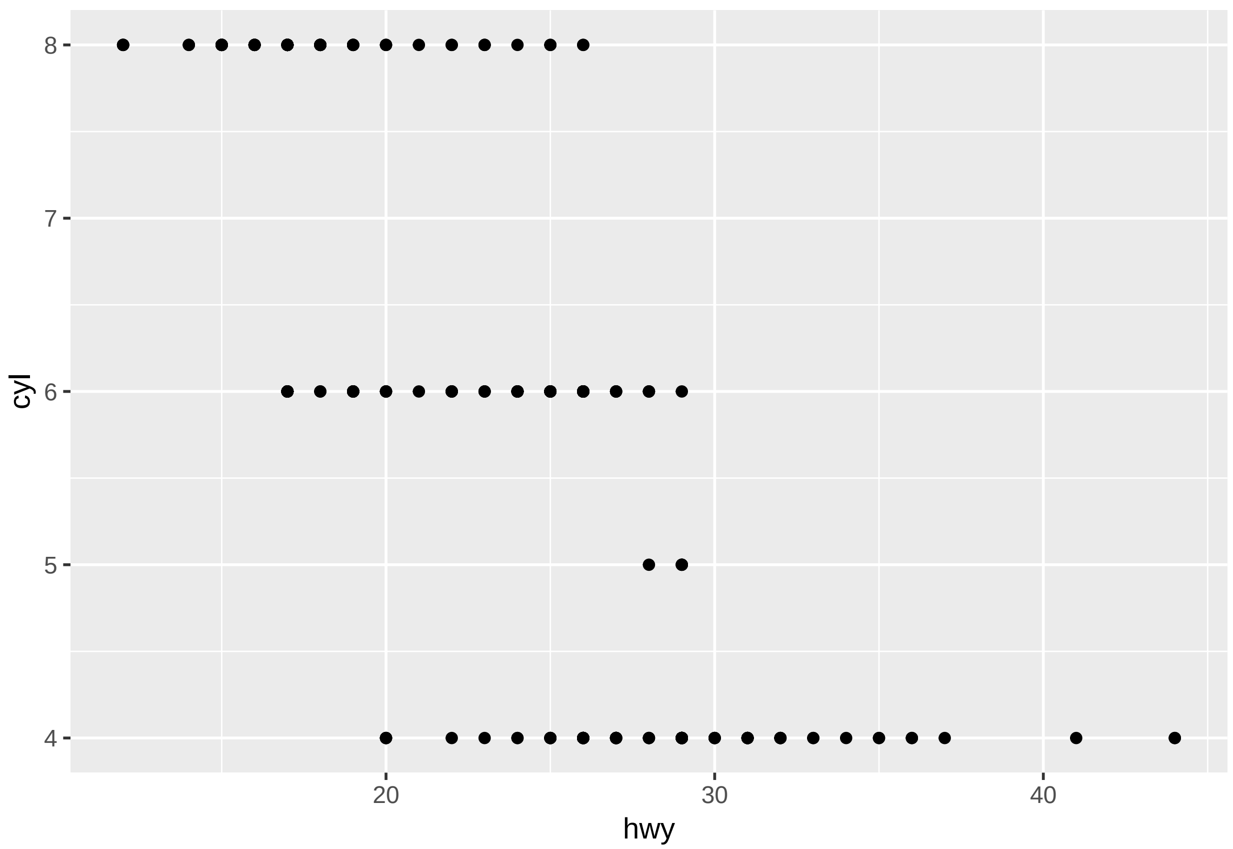
3. What does the drv variable describe? Read the help for ?mpg to find out.

```
?mpg
```

drv: f = front-wheel drive, r = rear wheel drive, 4 = 4wd

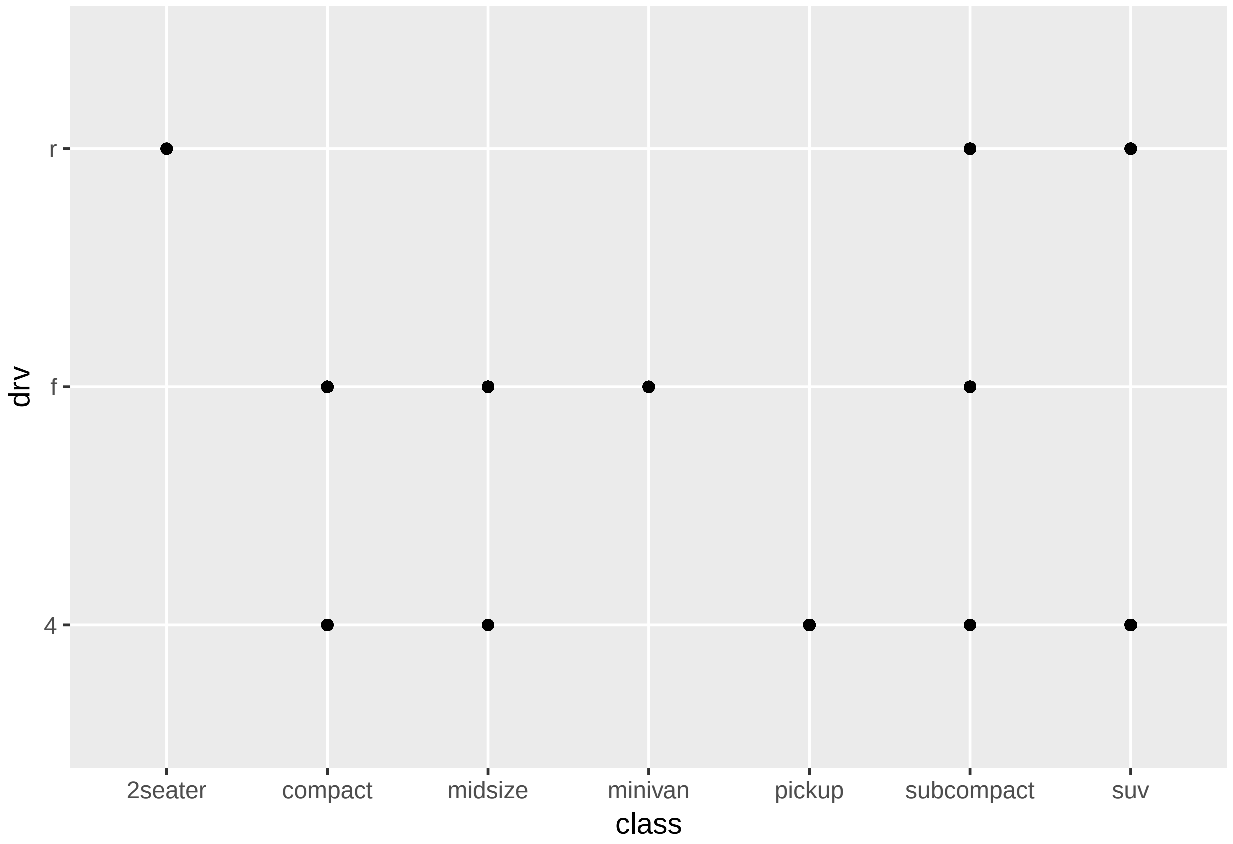
4. Make a scatterplot of hwy vs cyl.

```
ggplot(mpg) +
  geom_point(aes(x = hwy, y = cyl))
```



5. What happens if you make a scatterplot of class vs drv? Why is the plot not useful?

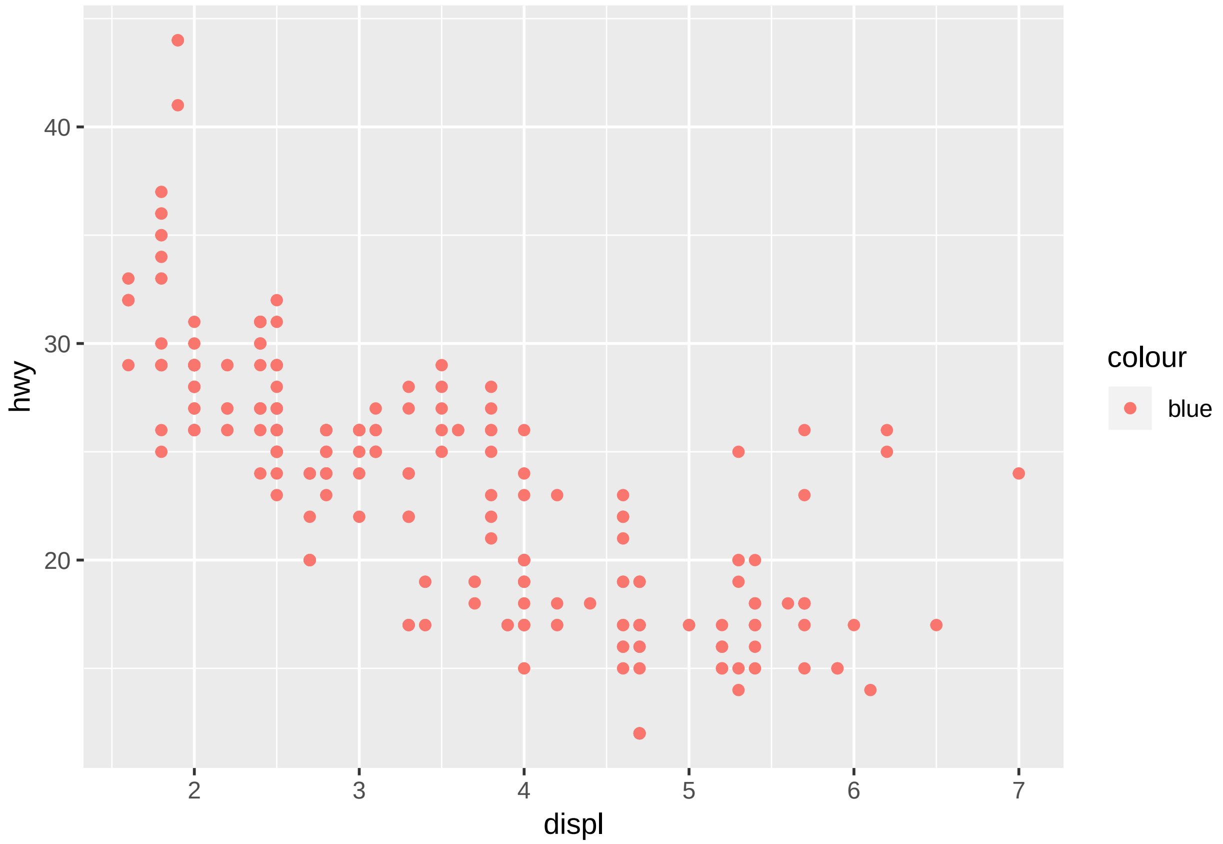
```
ggplot(mpg) +
  geom_point(aes(class, drv))
```



### 3.3.1 Exercises

1. What's gone wrong with this code? Why are the points not blue?

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, color = "blue"))
```



Need to put color attribute outside the aes()
Because the color argument was set within aes(), not geom_point()

2. Which variables in mpg are categorical? Which variables are continuous? (Hint: type ?mpg to read the documentation for the dataset). How can you see this information when you run mpg?

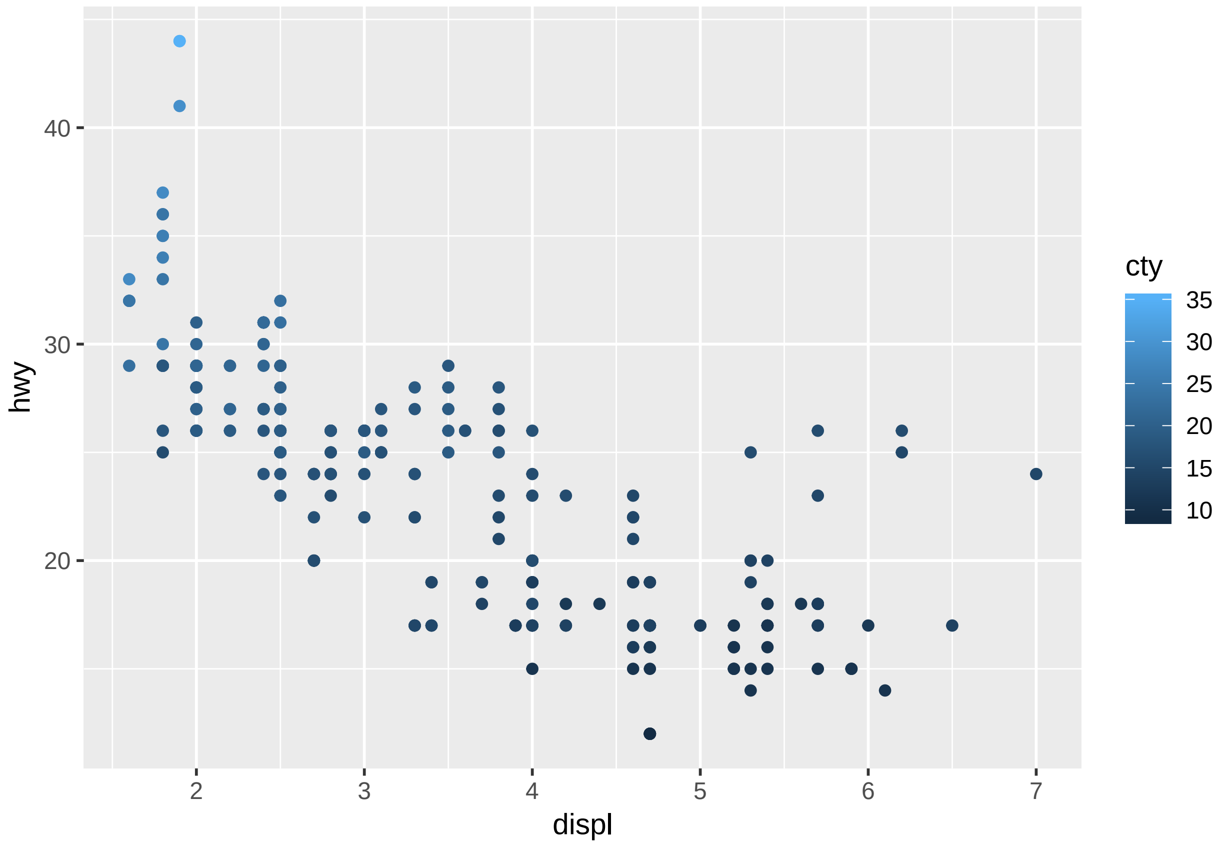
```
str(mpg)
## Classes 'tbl_df', 'tbl' and 'data.frame':    234 obs. of  11 variables:
##  $ manufacturer: chr  "audi" "audi" "audi" "audi" ...
##  $ model       : chr  "a4" "a4" "a4" "a4" ...
##  $ displ       : num  1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
##  $ year        : int  1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
##  $ cyl         : int  4 4 4 4 6 6 6 4 4 4 ...
##  $ trans       : chr  "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
##  $ drv         : chr  "f" "f" "f" "f" ...
##  $ cty         : int  18 21 20 21 16 18 18 18 16 20 ...
##  $ hwy         : int  29 29 31 30 26 26 27 26 25 28 ...
##  $ fl          : chr  "p" "p" "p" "p" ...
##  $ class       : chr  "compact" "compact" "compact" "compact" ...
```

Categorical: manufacturer, model, trans, drv, fl, class
Continuous: displ, cyl, cty, hwy

3. Map a continuous variable to color, size, and shape. How do these aesthetics behave differently for categorical vs. continuous variables?

**color:**

```
ggplot(mpg) +
  geom_point(aes(displ, hwy, color = cty))
```

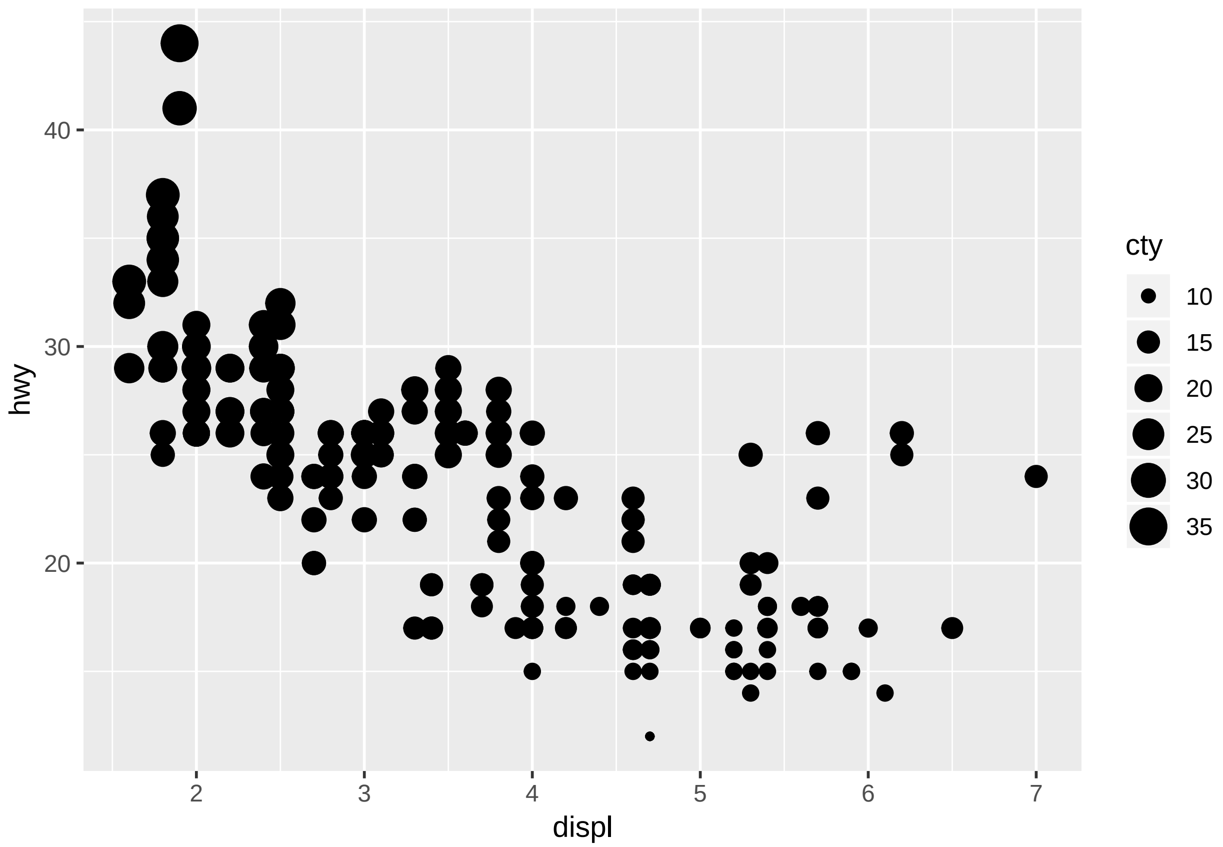


**shape:**

```
ggplot(mpg) +
  geom_point(aes(displ, hwy, shape = cty))
## Error: A continuous variable can not be mapped to shape
```

**size:**

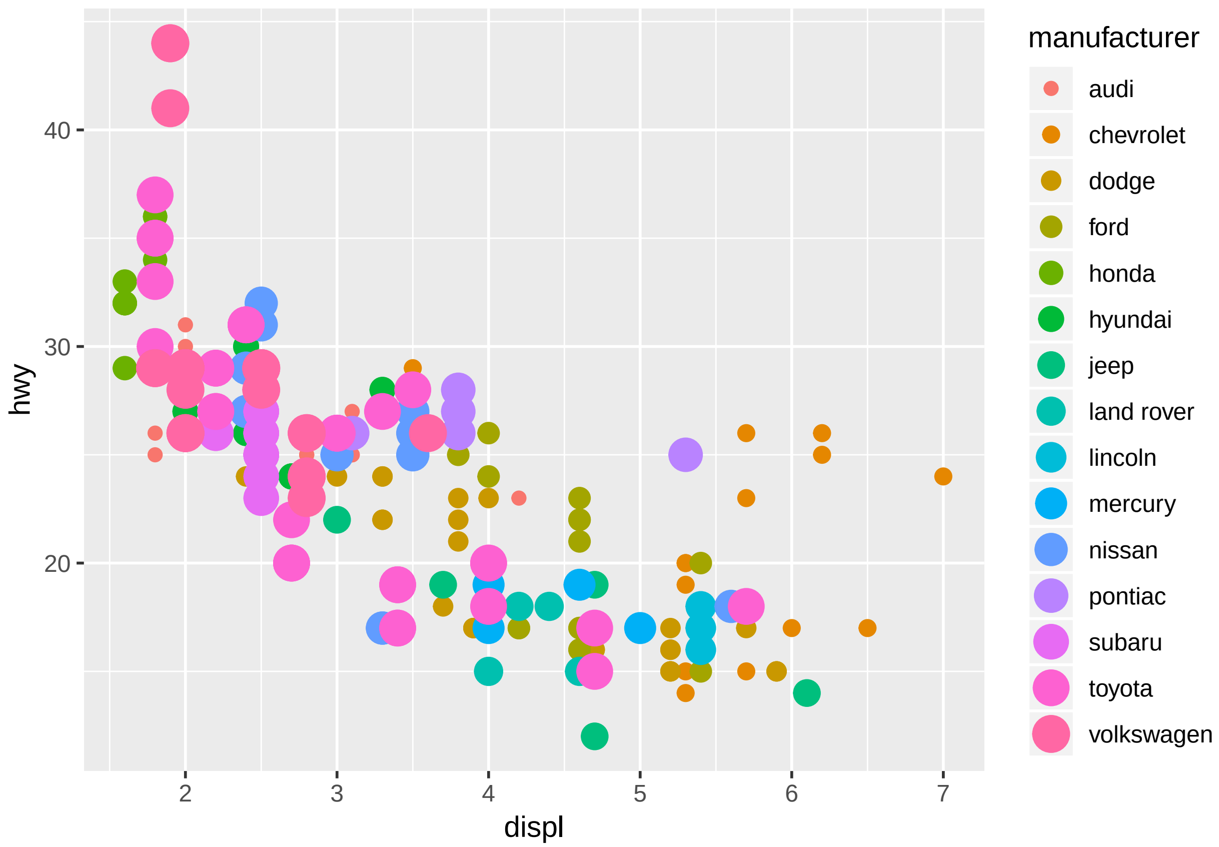
```
ggplot(mpg) +
  geom_point(aes(displ, hwy, size = cty))
```



4. What happens if you map the same variable to multiple aesthetics?

```
ggplot(mpg) +
  geom_point(aes(displ, hwy, color = manufacturer, size = manufacturer))
```

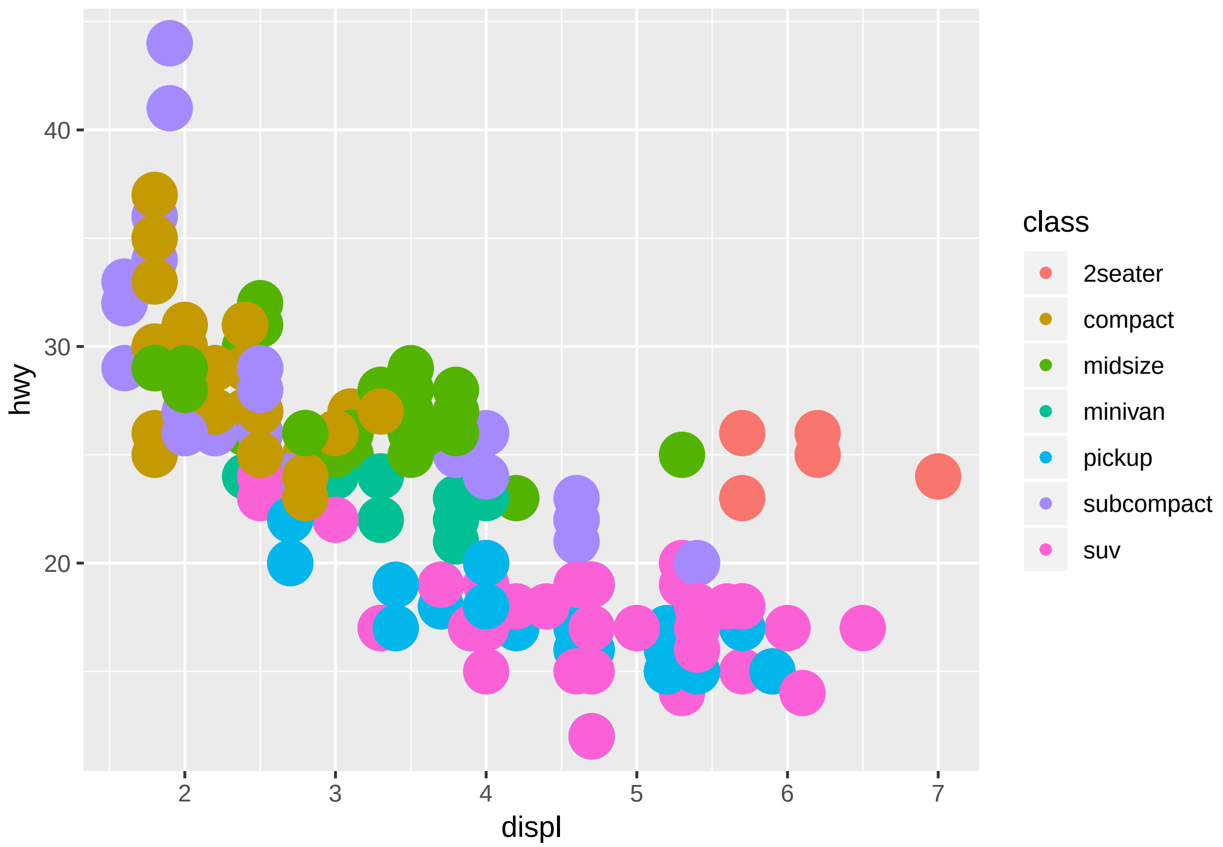
```
## Warning: Using size for a discrete variable is not advised.
```



5. What does the stroke aesthetic do? What shapes does it work with? (Hint: use ?geom_point)

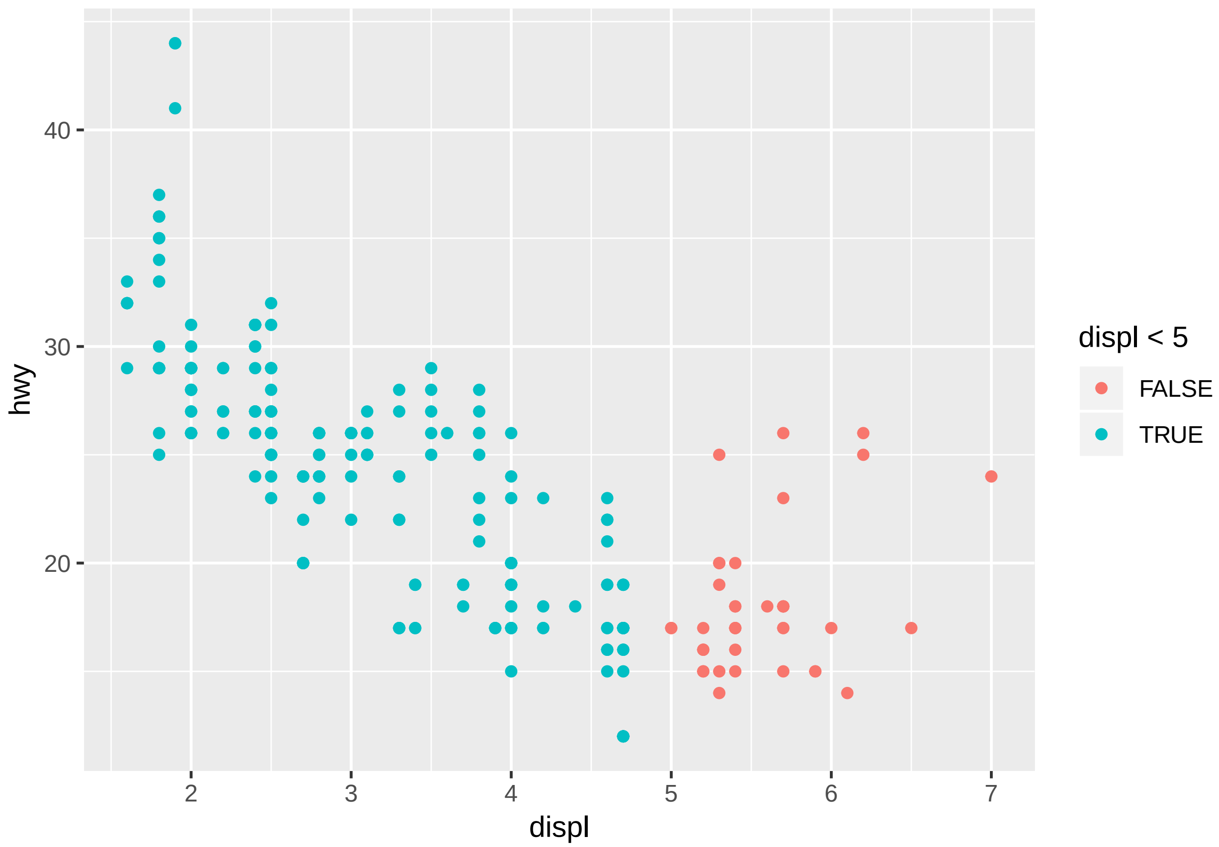
To modify the width of the border

```
ggplot(mpg) +
  geom_point(aes(displ, hwy, color = class, stroke = 5))
```



6. What happens if you map an aesthetic to something other than a variable name, like aes(colour = displ < 5)? Note, you'll also need to specify x and y.

```r
ggplot(mpg) +
  geom_point(aes(displ, hwy, color = displ < 5))
```


40
30
20
hwy
2
3
4
5
6
7
displ
displ < 5
FALSE
TRUE