A solution to ggplot2-book

Raju Rimal

2017-05-19

Contents

Prerequisites

This is a solution to the problems in ggplot2-book. In order to run all the solution, following packages need to be installed and loaded.

```
# devtools::install_github("hadley/tidyverse")
pkgs <- c("ggplot2", "dplyr", "pander", "stringr")
for (pkg in pkgs) require(pkg, character.only = TRUE)
```

Chapter 1

Getting Started

1.1 Fuel economy data

1.1.1 Exercise 2.2.1

1. List five functions that you could use to get more information about the `mpg` dataset. `str`, `summary`
2. How can you find out what other datasets are included with ggplot2?

```
data_ggplot <- data(package = "ggplot2")
pander(data_ggplot$result[, -c(1:2)], justify = "rl",
       split.cells = 50, emphasize.verbatim.cols = 1)
```

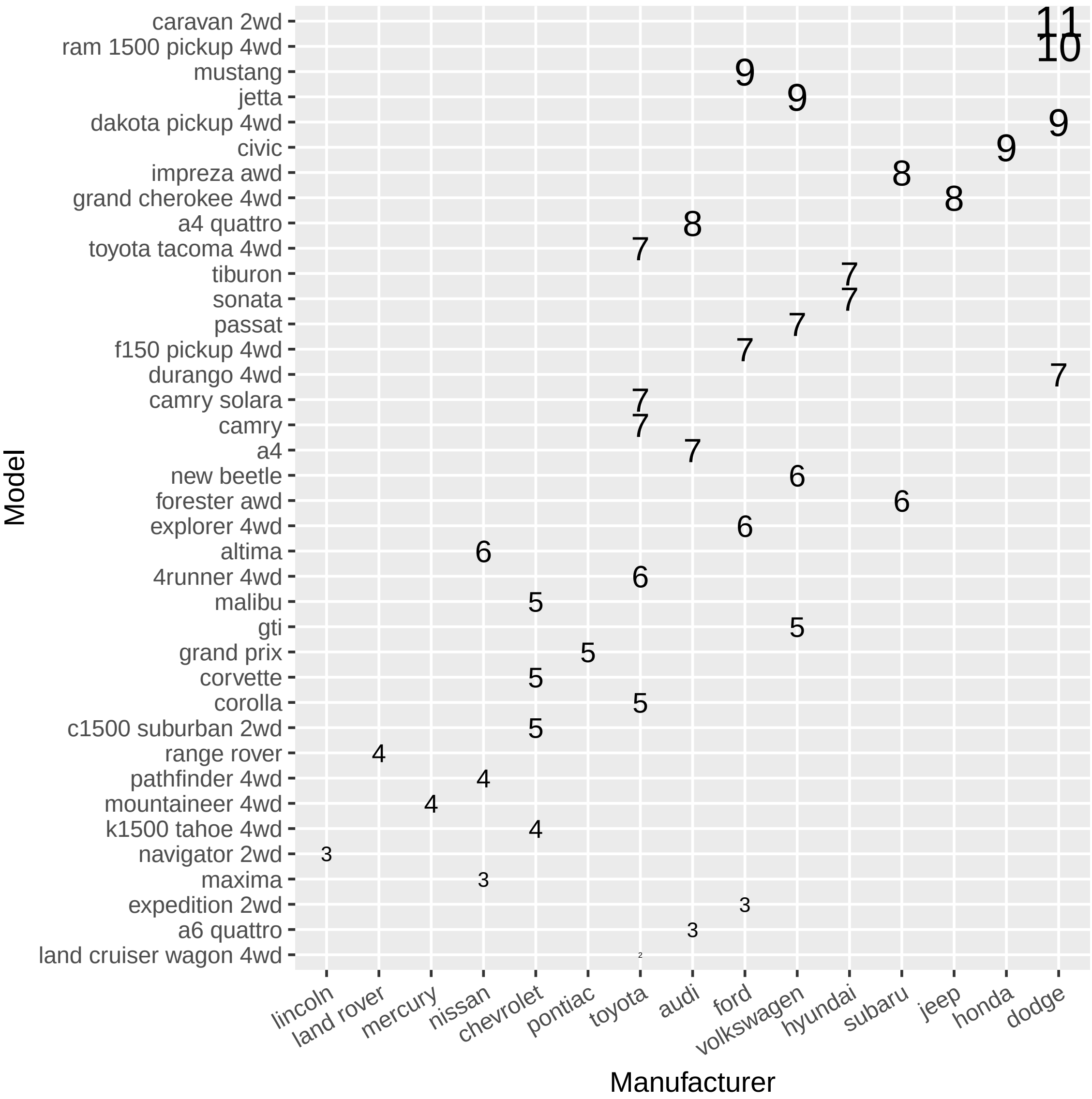
Item	Title
`diamonds`	Prices of 50,000 round cut diamonds
`economics`	US economic time series
`economics_long`	US economic time series
`faithfuld`	2d density estimate of Old Faithful data
`luv_colours`	'colors()' in Luv space
`midwest`	Midwest demographics
`mpg`	Fuel economy data from 1999 and 2008 for 38 popular models of car
`msleep`	An updated and expanded version of the mammals sleep dataset
`presidential`	Terms of 11 presidents from Eisenhower to Obama
`seals`	Vector field of seal movements
`txhousing`	Housing sales in TX

3. Apart from the US, most countries use fuel consumption (fuel consumed over fixed distance) rather than fuel economy (distance travelled with fixed amount of fuel). How could you convert `cty` and `hwy` into the European standard of l/100km?

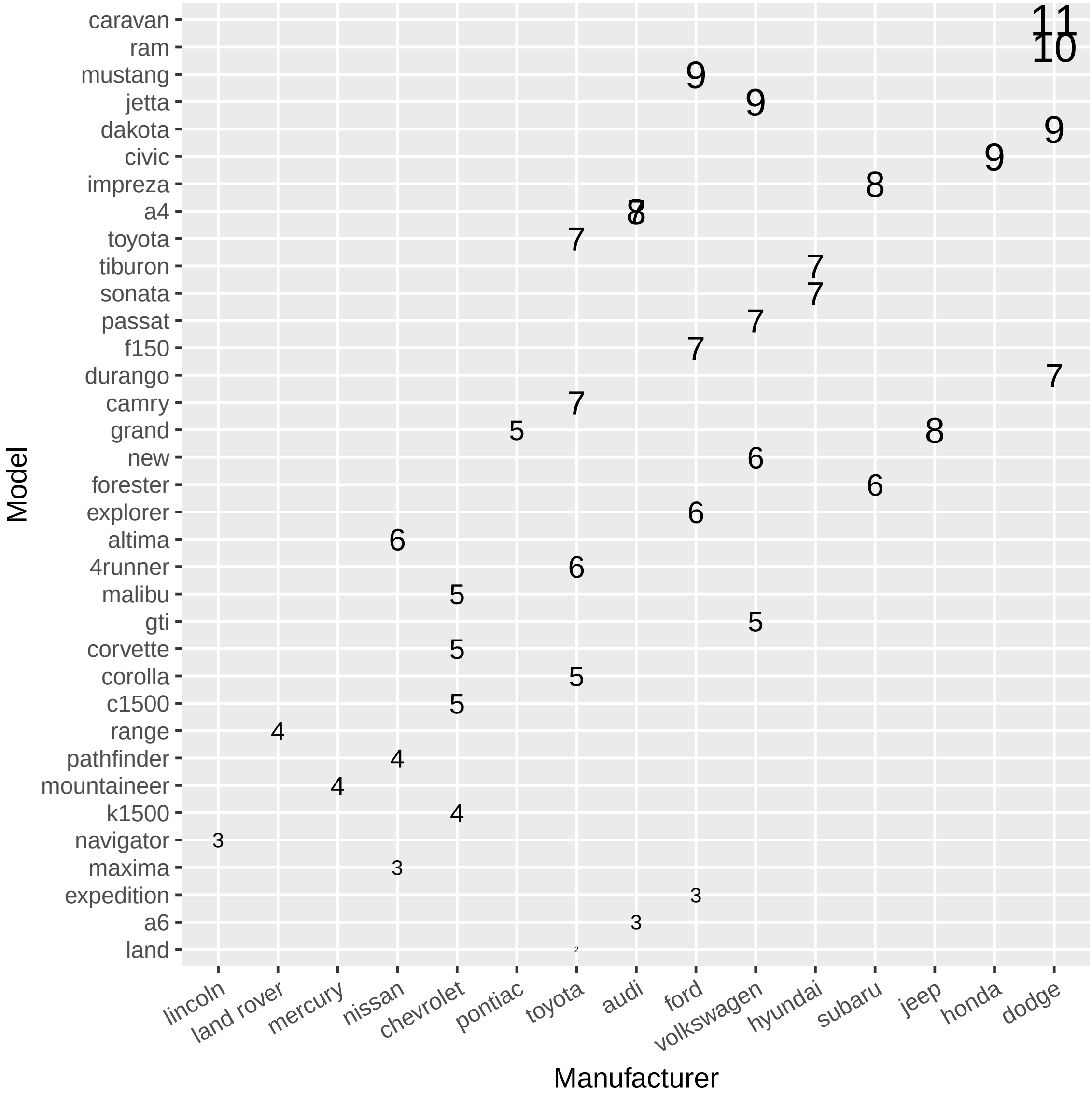
```
litre_per_km <- function(mile_per_gallon) {
  return(3.78541 / (1.60934 * mile_per_gallon))
}

mpg_eu <- mpg %>%
  mutate(cty = litre_per_km(cty) * 100,
         hwy = litre_per_km(hwy) * 100)
```

4. Which manufacturer has the most the models in this dataset? Which model has the most variations? Does your answer change if you remove the redundant specification of drive train (e.g. "pathfinder 4wd", "a4 quattro") from the model name?

```
dta <- mpg %>%
  mutate(
    model_clean = str_extract(model, "(^[a-zA-Z0-9]+)")
  ) %>%
  group_by(manufacturer, model_clean, model) %>%
  summarize(count = n())
dta %>%
  ggplot(aes(reorder(manufacturer, count),
             reorder(model, count),
             label = count, size = count)) +
  geom_text() +
  theme(axis.text.x = element_text(angle = 30, hjust = 1),
        legend.position = "none") +
  labs(x = "Manufacturer", y = "Model")
```



```
dta %>%
  ggplot(aes(reorder(manufacturer, count),
             reorder(model_clean, count),
             label = count, size = count)) +
  geom_text() +
  theme(axis.text.x = element_text(angle = 30, hjust = 1),
        legend.position = "none") +
  labs(x = "Manufacturer", y = "Model")
```

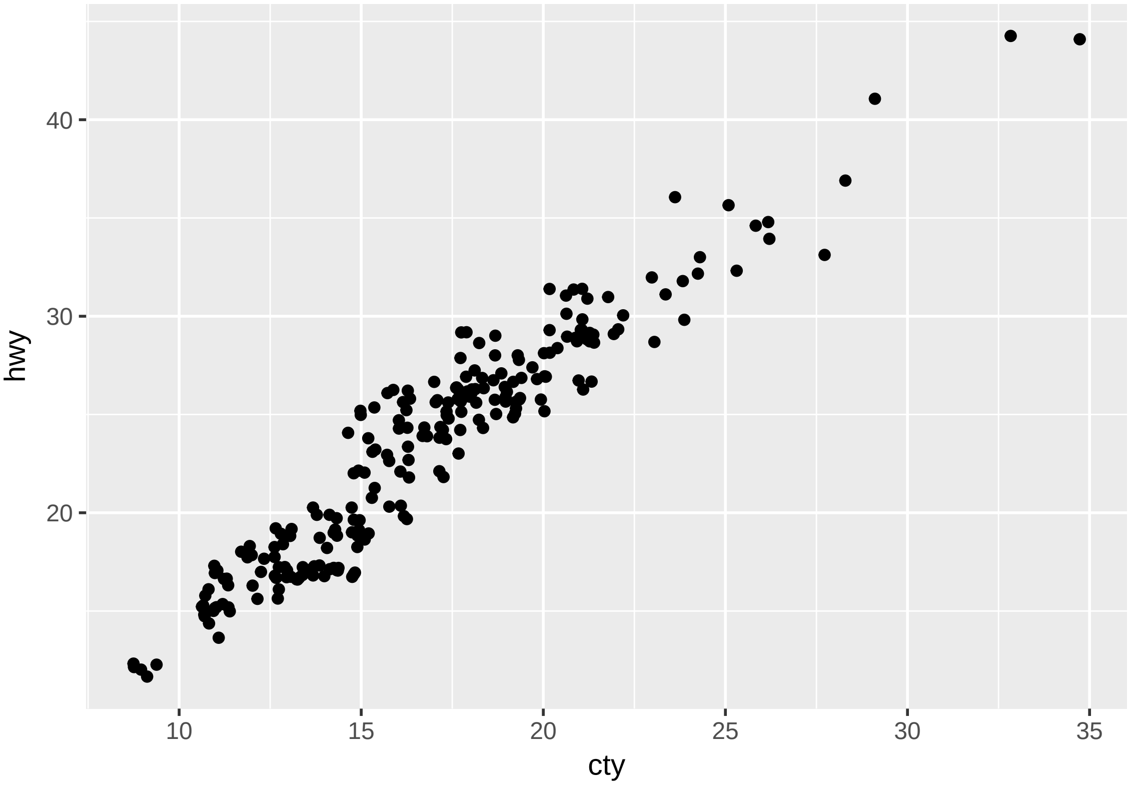


1.2 Key components

1.2.1 Exercises 2.3.1

1. How would you describe the relationship between `cty` and `hwy` ? Do you have any concerns about drawing conclusions from that plot?

```
ggplot(mpg, aes(cty, hwy)) + geom_point(position = "jitter")
```



Its a linear relationship.

2. What does `ggplot(mpg, aes(model, manufacturer)) + geom point()` show? Is it useful? How could you modify the data to make it more informative?

   Not very informative but shows how may models a manufacture have. It is more visible if we use `geom_count` which uses the `count` summary statistics for each combination of `model` and `manufacturer`.

3. Describe the data, aesthetic mappings and layers used for each of the following plots. You'll need to guess a little because you haven't seen all the datasets and functions yet, but use your common sense! See if you can predict what the plot will look like before running the code.

   a. `ggplot(mpg, aes(cty, hwy)) + geom point()`

      Data is `mpg`, `x` and `y` axis are mapped to `cty` and `hwy` variables and a layer of `point` is added.

   b. `ggplot(diamonds, aes(carat, price)) + geom point()`

      Data is `diamonds`, `x` and `y` axis are mapped to `carat` and `price` variables and a layer of `point` is added.

   c. `ggplot(economics, aes(date, unemploy)) + geom line()`

      Data is `economics`, `x` and `y` axis are mapped to `date` and `unemploy` variables and a layer of `line` is added.

   d. `ggplot(mpg, aes(cty)) + geom histogram()`

      Data is `mpg`, `x` axis are mapped to `cty` and a layer of `histogram` is added which uses the default bins of `30`.

1.3 Colour, size, shape and other aesthetic attributes

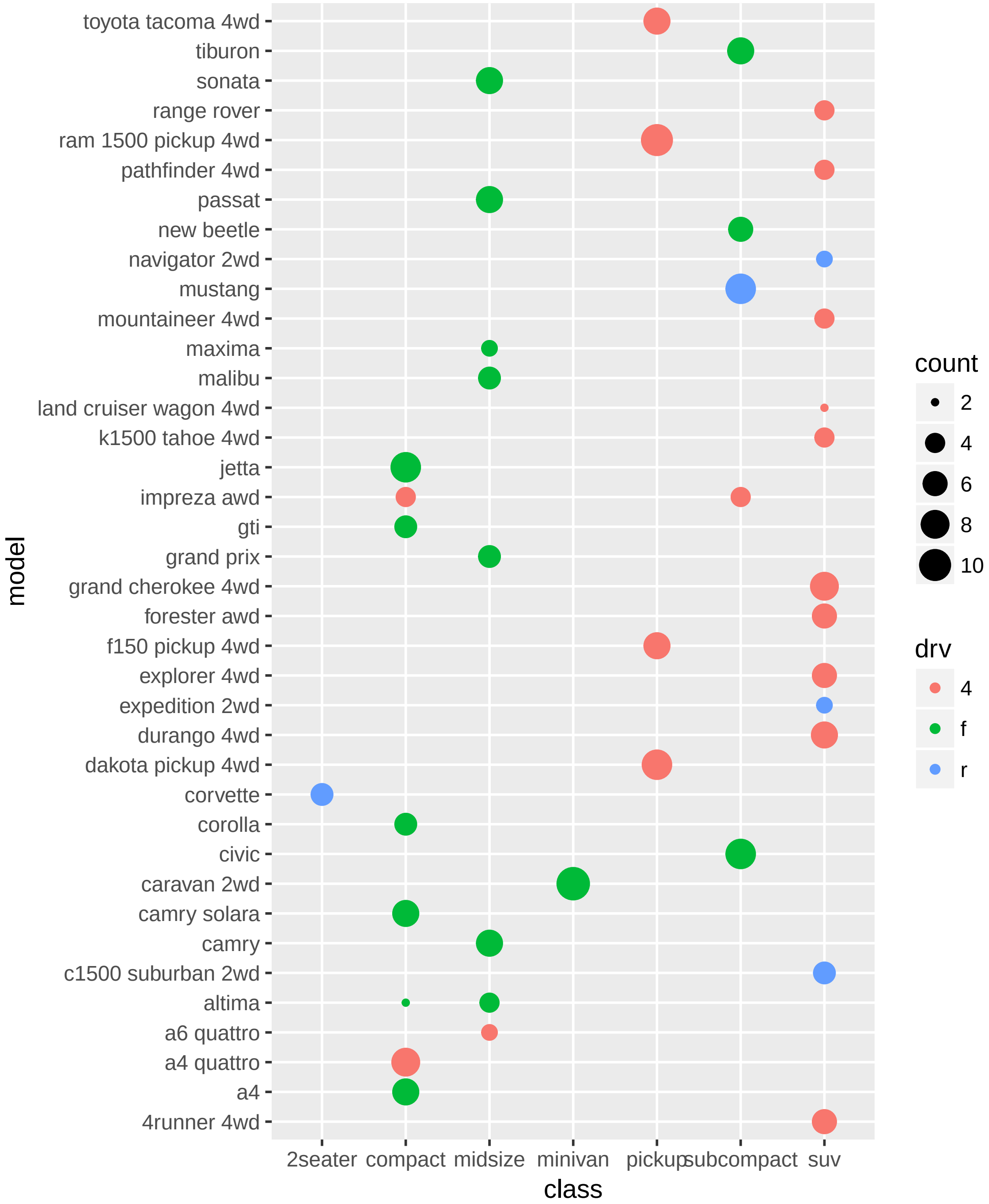
1.3.1 Exercises 2.4.1

1. Experiment with the colour, shape and size aesthetics. What happens when you map them to continuous values? What about categorical values? What happens when you use more than one aesthetic in a plot?

2. What happens if you map a continuous variable to shape? Why? What happens if you map trans to shape? Why?

3. How is drive train related to fuel economy? How is drive train related to engine size and class?

```
mpg %>%
  group_by(model, class, drv) %>%
  summarize(count = n()) %>%
  ggplot(aes(class, model, color = drv, size = count)) +
  geom_point()
```



Here, we see that `suv` and `pickup` has mostly 4 whell drive while rest are front-wheeled drive.

1.4 Facetting

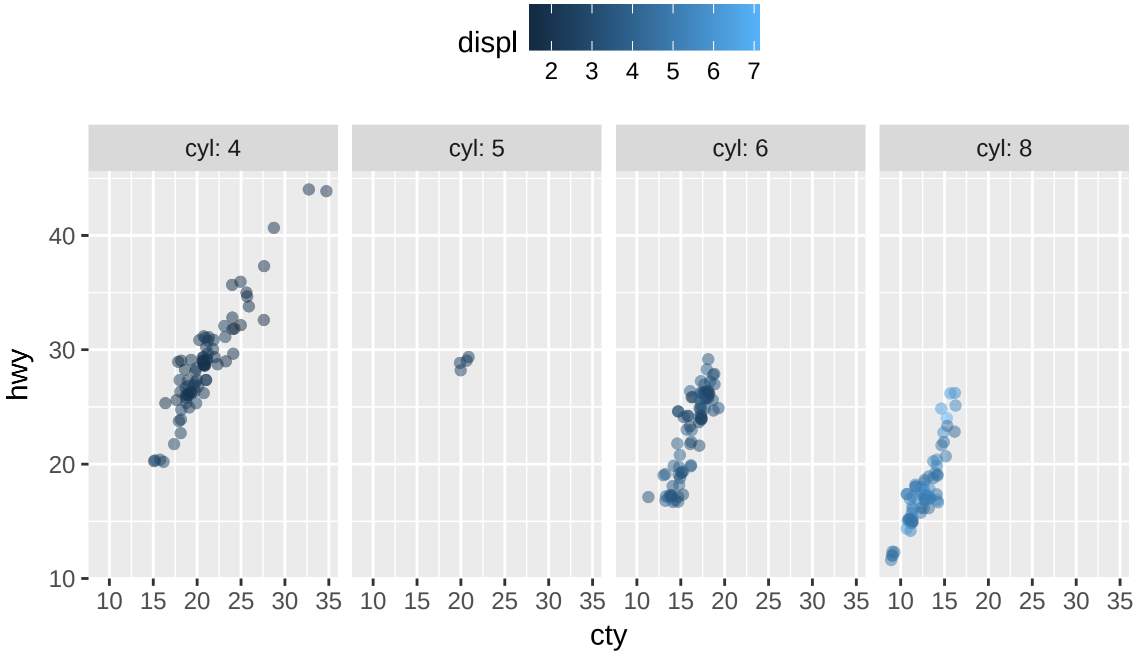
1.4.1 Exercises 2.5.1

1. What happens if you try to facet by a continuous variable like `hwy` ? What about `cyl` ? What's the key difference?

When a continuous variables, like hwy, is used for facet, ggplot converts it into factor and creates facet from all unique value of that continuous variable. Here hwy has many unique values so we will get many facets for each of them while cyl has few discrete values and is useful to use for faceting.

2. Use facetting to explore the 3-way relationship between fuel economy, engine size, and number of cylinders. How does facetting by number of cylinders change your assessement of the relationship between engine size and fuel economy?

```
ggplot(mpg, aes(cty, hwy, color = displ)) +
  geom_point(alpha = 0.5, position = "jitter") +
  facet_grid(.~cyl, labeller = label_both) +
  theme(legend.position = "top")
```



Here we can see that larger engine size has lower milage in both city and highway. In addition, vechile with large number of cylender has larger engine size. Further there are very few vechile having 5 cylinder.

3. Read the documentation for facet wrap(). What arguments can you use to control how many rows and columns appear in the output?

   The nrow and ncol arguments in facet_wrap() controls the number of rows and columns.

4. What does the scales argument to facet wrap() do? When might you use it?

   Here scales can take three values – free, free_x and free_y. free_x gives separate x-axis for each facet, free_y gives separate y-axis for each facet and similarly, free gives separate x and y axis for each facet.

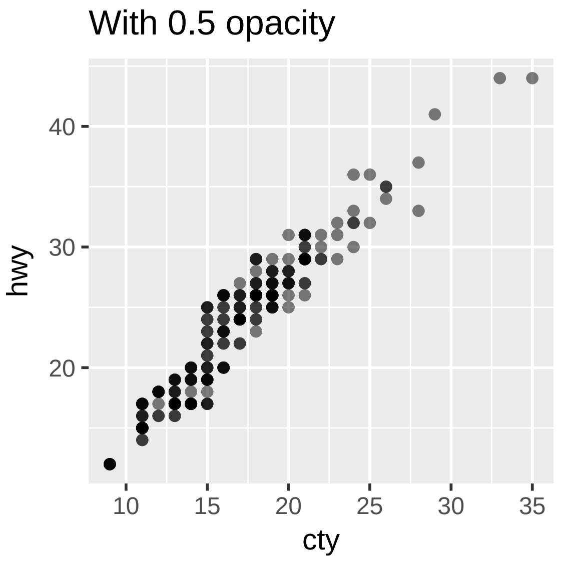
1.5 Plot geoms

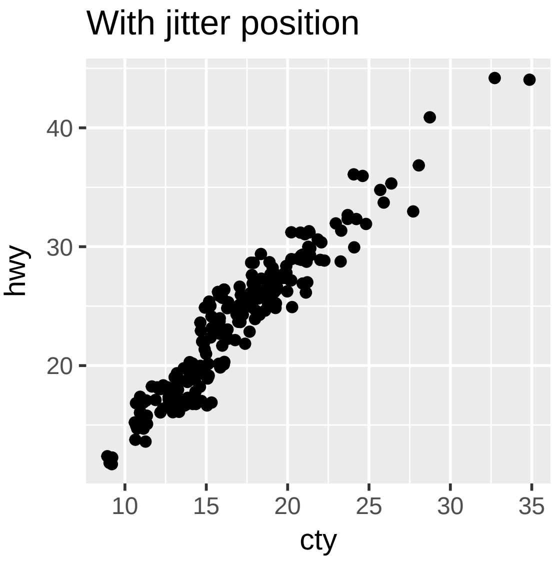
1.5.1 Exercises 2.6.6

1. What's the problem with the plot created by ggplot(mpg, aes(cty, hwy)) + geom_point()? Which of the geoms described above is most effective at remedying the problem?

Many points in this plots are overlapped so we can see only few points. In these situation, we can either use `alpha` argument for making the points transparent so that we can see the points underneath or use `jitter` position to add some randomness on the points.

```r
ggplot(mpg, aes(cty, hwy)) +
  geom_point(alpha = 0.5) +
  ggtitle("With 0.5 opacity")

ggplot(mpg, aes(cty, hwy)) +
  geom_point(position = "jitter") +
  ggtitle("With jitter position")
```

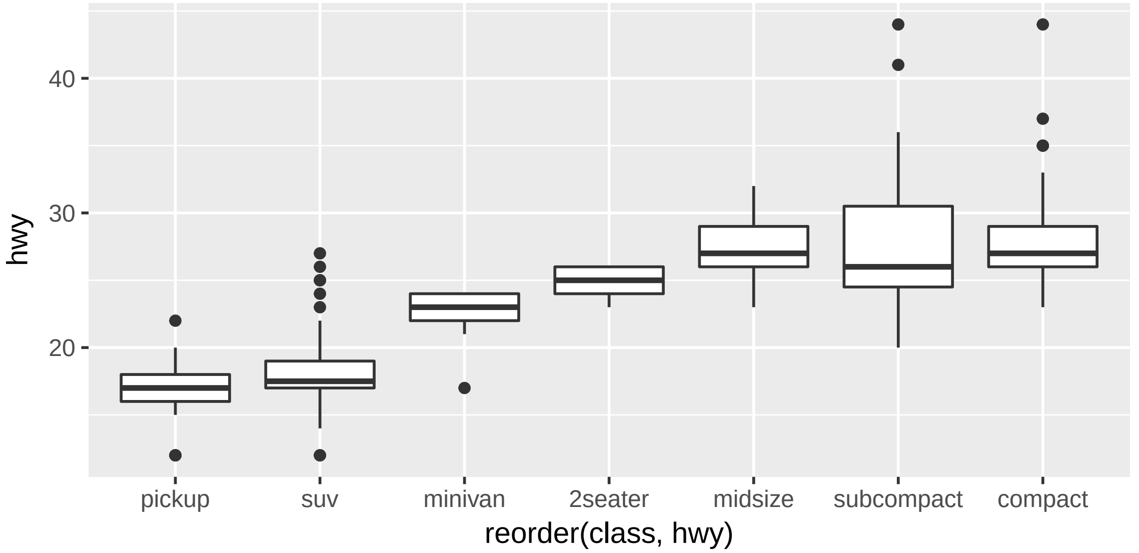




2. One challenge with `ggplot(mpg, aes(class, hwy)) + geom_boxplot()` is that the ordering of `class` is alphabetical, which is not terribly useful. How could you change the factor levels to be more informative?

Rather than reordering the factor by hand, you can do it automatically based on the data: `ggplot(mpg, aes(reorder(class, hwy), hwy)) + geom_boxplot()`. What does `reorder()` do? Read the documentation.

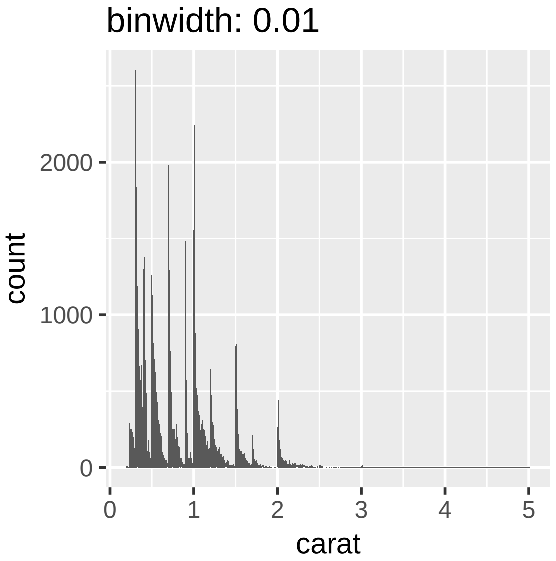
```r
ggplot(mpg, aes(reorder(class, hwy), hwy)) + geom_boxplot()
```

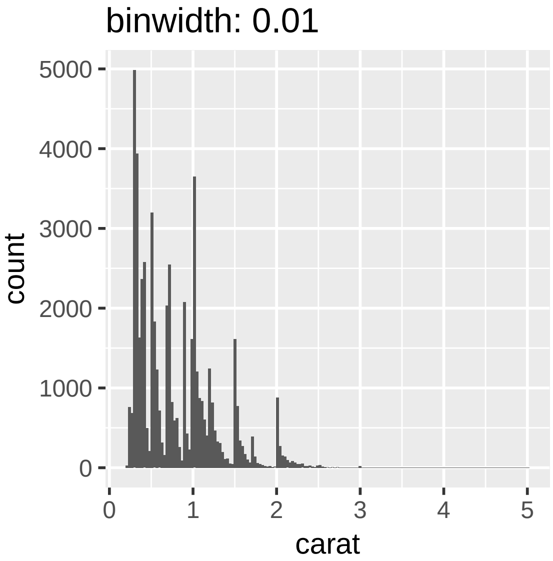


Here `reorder` arrange the factor `class` according to the magnitute of `hwy` in ascending order.

3. Explore the distribution of the carat variable in the `diamonds` dataset. What binwidth reveals the most interesting patterns?

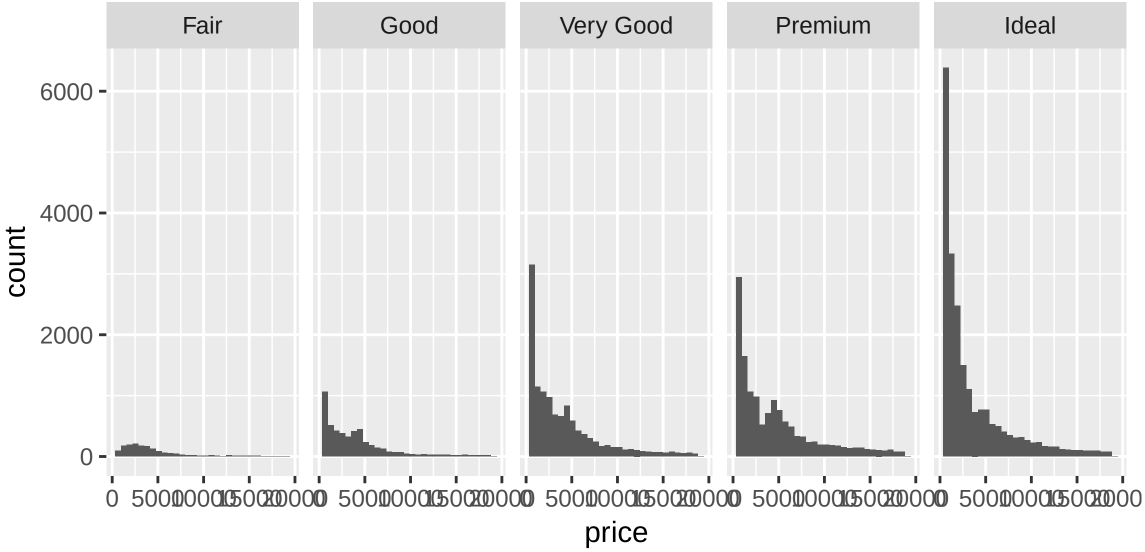
```r
ggplot(diamonds, aes(carat)) +
  geom_histogram(binwidth = 0.01) +
  ggtitle("binwidth: 0.01")

ggplot(diamonds, aes(carat)) +
  geom_histogram(binwidth = 0.03) +
  ggtitle("binwidth: 0.01")
```





4. Explore the distribution of the price variable in the `diamonds` data. How does the distribution vary by cut?

```r
ggplot(diamonds, aes(price)) +
  geom_histogram(bins = 30) +
  facet_grid(.~cut)
```



5. You now know (at least) three ways to compare the distributions of subgroups: `geom_violin()`, `geom_freqpoly()` and the colour aesthetic, or `geom_histogram()` and facetting. What are the strengths and weaknesses of

each approach? What other approaches could you try?

6. Read the documentation for `geom_bar()`. What does the `weight` aesthetic do?
7. Using the techniques already discussed in this chapter, come up with three ways to visualise a 2d categorical distribution. Try them out by visualising the distribution of `model` and `manufacturer`, `trans` and `class`, and `cyl` and `trans`.

Bibliography