School of Economics and Management

Information Science III

6. The Grammar of Graphics and ggplot2 (1)

Yuki Yanai

🌐 https://yukiyanai.github.io

✉ yanai.yuki@kochi-tech.ac.jp

Today's Goals

- Understand the (layered) grammar of graphics
- Review the basic usage of ggplot2 package

Grammar of Graphics

What is a graphic?

- Grammar of graphics
    - Grammatically defines a graphic
    - Makes clear the composition of complicated graphics
    - Reveals connections between seemingly unrelated graphics
- Grammar helps us understand what a well-formed graphic is

Grammar of Graphics

- Original idea can be found in Bertin (1983). *Semiology of Graphics.*
- Idea was clearly defined by Wilson, Anand, and Grossman (2005) (also see Wilkinson 2012)
- Modified for R and ggplot2 by Hadley Wickham (see https://had.co.nz/ggplot2/)

Simple Dataset

A	B	C	D
2	3	4	a
1	2	1	a
4	5	15	b
9	10	80	b

- Draw a scatterplot of A vs C
  - Use the shape of point for D

Mapping

A	B	C	D
2	3	4	a
1	2	1	a
4	5	15	b
9	10	80	b

- A -> position on x-axis
- C -> position on y-axis
- D -> shape

Mapping

x	y	shape
2	4	a
1	1	a
4	15	b
9	80	b

- This mapping can be used not only for a scatterpolot but also for other geometric representations such as a line pilot or a bar chart

Metric Conversion

x	y	shape
2	4	a
1	1	a
4	15	b
9	80	b

- Data units -> Physical units (or aesthetic units) in a graphic
  - Choose a scale: linear transformation, log transformation, etc.
  - Choose a coordinate to use
    - We usually use a Cartesian coordinate system

Metric Conversion

$$\mathrm{floor}\left(\frac{x-\min(x)}{\mathrm{range}(x)}\cdot \mathrm{screen\ \ width}\right)$$

x	y	shape
2	4	a
1	1	a
4	15	b
9	80	b

- Example: we will scale
  - x-position to [0, 200]
  - y-position to [0, 300]
  - Shape: “a” to circle, and “b” to triangle

Metric Conversion

$$\mathrm{floor}\left(\frac{x-\min(x)}{\mathrm{range}(x)}\cdot \mathrm{screen\ \ width}\right)$$

x	y	shape
25	11	circle
0	0	circle
75	53	triangle
200	300	triangle

- Example: we will scale
  - x-position to [0, 200]
  - y-position to [0, 300]
  - Shape: “a” to circle, and “b” to triangle

Draw a graph

- Plot the transformed data onto a screen
- Add annotations to the plot
  - Background
  - Axis labels
  - Title
  - Etc.

Steps to create a plot

- Three sources for a graphic
  - Data
    - Point geom for scatterplots
  - Scales and coordinate system
    - Axes and legends
  - Plot annotations
    - Background, axis labels, plot title, etc.

ggplot2

Make Graphs with ggplot2

- R has some built-in functions for grraphics
  - Different functions for different types of graphics
    - Need to remember many functions
  - Difficult to tweak the details
- ggplot2 enables us to make beautiful graphics easily
- ggplot2 is the de-facto standard for R graphics

★ Let’s learn about ggplot2 more!

ggplot2

- An R package for data visualization
- grammar of graphics
  - Once you master the grammar, you can make many different graphics in a consistent way

Get used to tidyverse

- tidy + universe

- A set of packages for data analysis

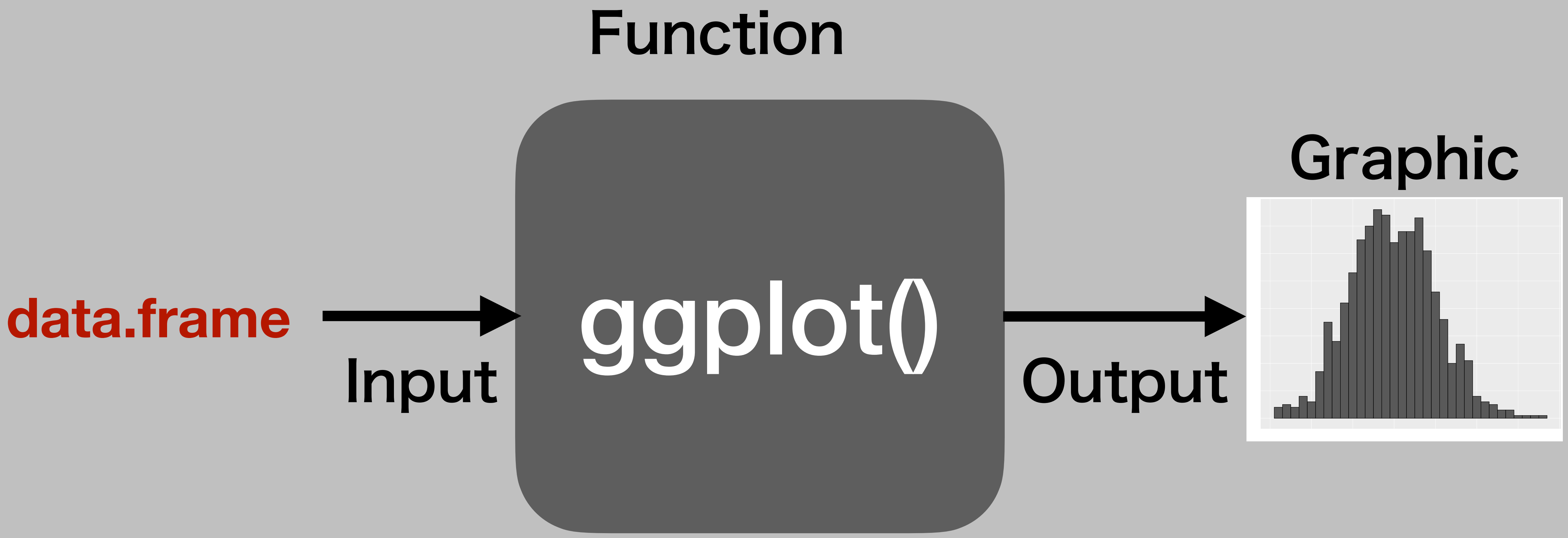
  - Includes: **ggplot2**, dplyr, tidyr, readr, purrr, tibble, *etc*.

- For more details: `https://www.tidyverse.org/`

Hadley Wickham

- Chief Scientist at RStudio
- Creator of many essential R packages including ggplot2
  - You can use the textbook of ggplot2 online for free: https://ggplot2-book.org/
- Website: https://hadley.nz/

How ggplot2 works



Function

Graphic

data.frame

ggplot()

Input

Output

- Pass `data.frame` to `ggplot()` to get a graphic output

data.frame?

- A most frequently used data format in R is data.frame
  - You get data.frame when you read a rectangular data set (tidy data) by `read.csv()` or `readr::read_csv()`
  - You can create a data frame by `tibble::tibble()` or `data.frame()`
  - You can transform a matrix into a data.frame by `tibble::as_tibble()` or `as.data.frame()`

How to make a data.frame

- n: sample size
- x：a random variable, $x_i \sim$ Uniform$(0,1)$
- y: a random variable, $y_i \sim$ Normal$(0.8x_i, \sigma^2 = 1)$
- Create a data.frame named `myd` containing 2 variables x and y

Run the following (assuming that tidyverse has been loaded)

```
n <- 100
x_vec <- runif(n, min = 0, max = 1)
y_vec <- rnorm(n, mean = 0.8 * x_vec, sd = 1)
myd <- tibble(x = x_vec, y = y_vec)
class(myd)
```

Use built-in data

- R provides a variety of data sets
- You can see the available data by `data()`
- E.g., `mtcars`; `diamonds`

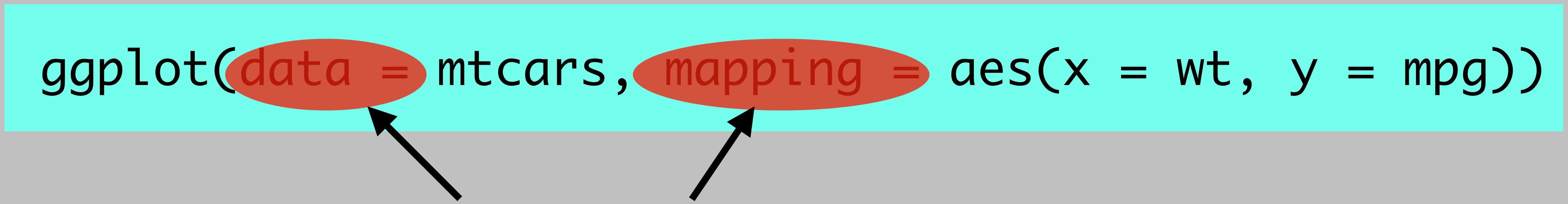
```
data(mtcars)
glimpse(mtcars)

data(diamonds)
glimpse(diamonds)
```

Basic Usage of ggplot2

1. Pass a data.frame that you'd like to visualize to `ggplot()` function
2. Add a layer[s] of graphics you'd like to make
3. Customize labels, legends, etc.
4. Display the graphic by `plot()` or `print()`

1. ggplot()

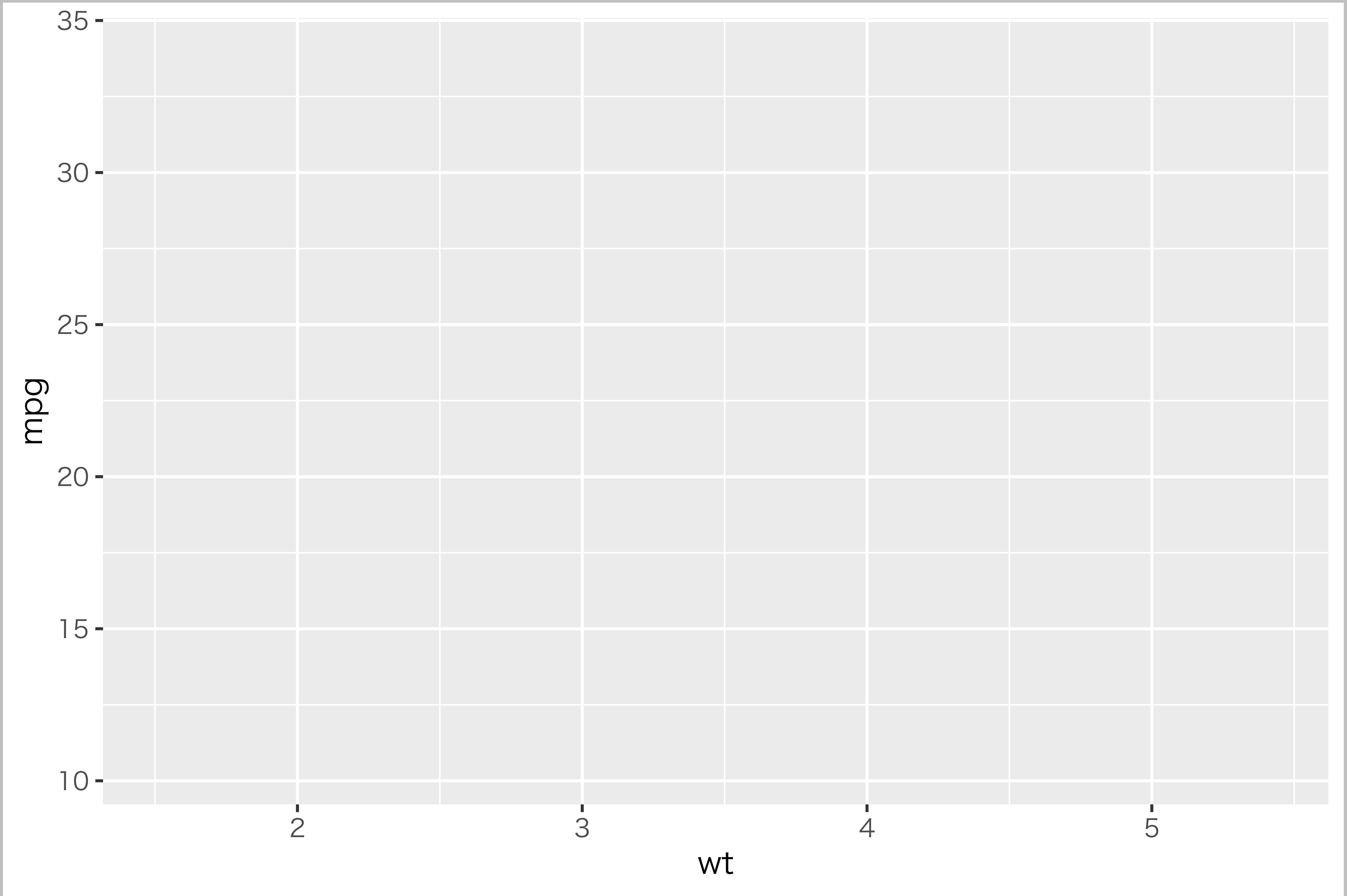
- First argument is `data`: pass a `data.frame`
- Second argument is `mapping`：specify which variables represent what by `aes` (**<u>aes</u>**thetics)
- E.g., With the data.frame `mtcars`, map `wt` onto x-axis and `mpg` onto y-axis



```
ggplot(data = mtcars, mapping = aes(x = wt, y = mpg))
```

Frequently omitted

```
p1_1 <- ggplot(mtcars, aes(x = wt, y = mpg))
plot(p1_1)
```



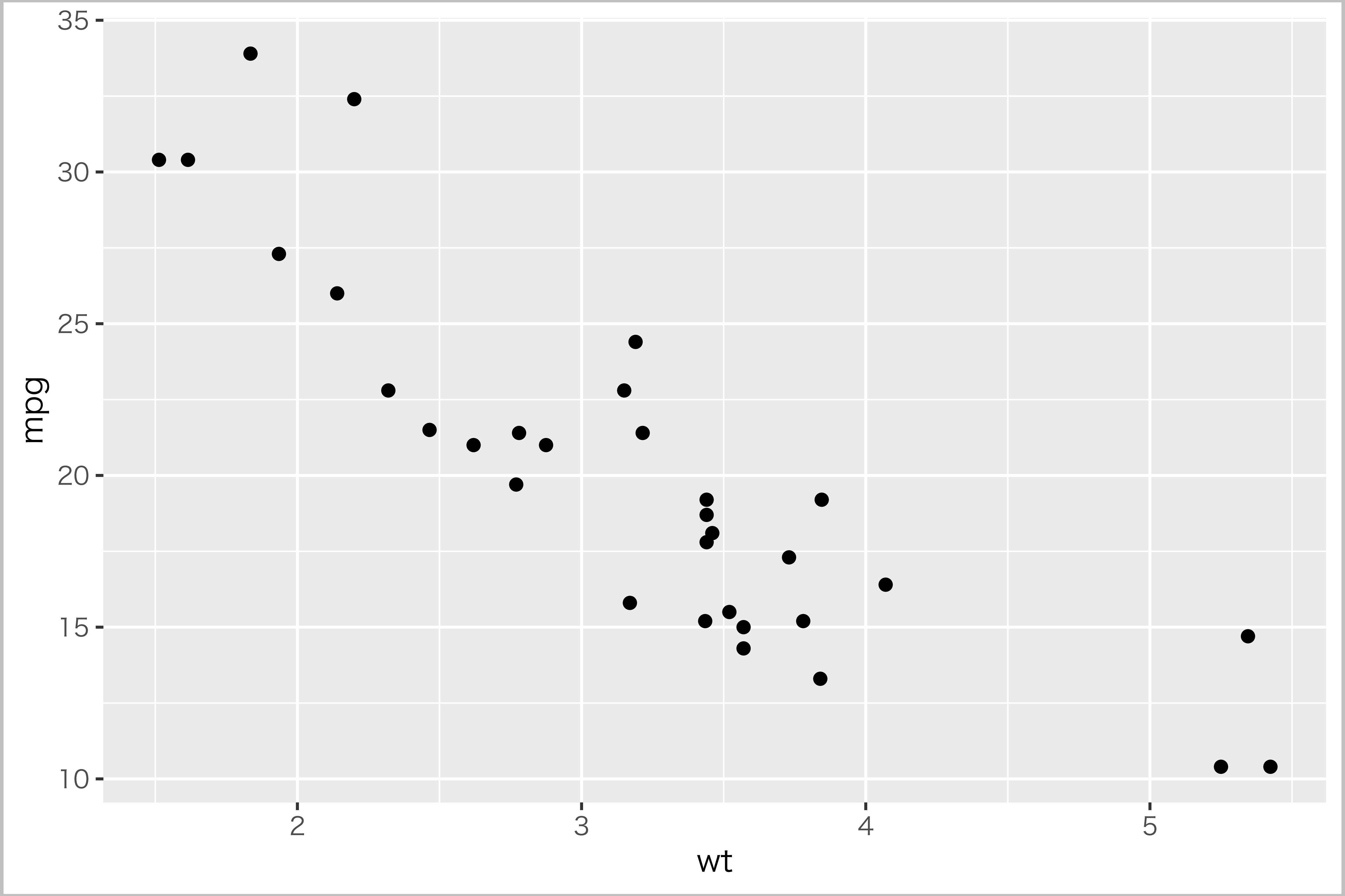
2. `geom_xxx()`

- Add a graphic layer by a function beginning with geom (geometry)
- xxx can be may different things
  - Histogram: `geom_histogram()`
  - Scatter plot: `geom_point()`
- Depending on which geom you use, what you should specify for `aes` might differ

```
p1_2 <- p1_1 + geom_point()
plot(p1_2)
```

geom for scatter plot

Previously saved object

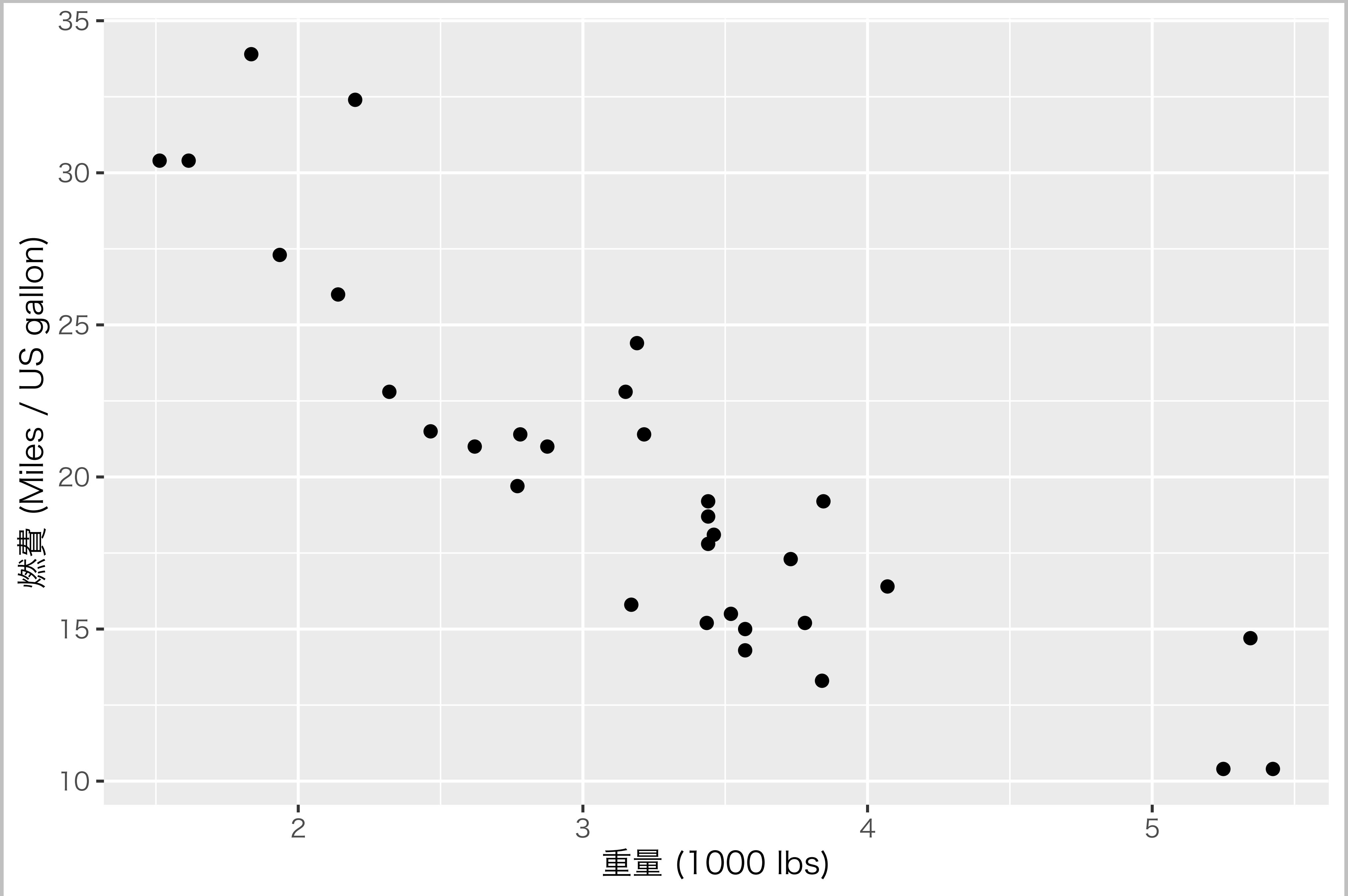


3. Other customizations

- E.g. Modify axis labels by `labs()`
  - Label should be surrounded by quotation marks
  - x-axis: x
  - y-axis: y
  - Title of the plot: title (for blank title, `""` or just omit the argument)

Previously saved object

```
p1_3 <- p1_2 +
    labs(x = "重量 (1000 lbs)",
         y = "燃費 (Miles / US gallon)")
plot(p1_3)
```



4. `plot()` or `print()`

- Save a ggplot output as an object. Then, display it by `plot( )` or `print( )`

✦ Save a graphic as an object makes it easy to re-use the graphic

- Re-examine the graphic
- Layout multiple graphics in a single picture (using patchwork package)
- Export graphics to other files (PDF, PNG, etc.)

Demonstrations

- http://htmlpreview.github.com/?https://github.com/yukiyanai/KUT_R/blob/master/htmls/yanai_kutR_001.html
- https://rstudio.cloud/project/762403

Some frequently used functions (1)

- A vertical line at x = a

```
geom_vline(xintercept = a, color = "red",)
           linetype = "dashed")
```

- A horizontal line at y = b

```
geom_hline(yintercept = b, color = "blue",)
           linetype = "dotted")
```

Some frequently used functions (1)

- Visualize x in a range: $x \in [a, b]$

```
xlim(a, b)
```

- Visualize y in a range: $y \in [s, t]$

```
ylim(s, t)
```

- Zoom in to $x \in [a, b]$, $y \in [s, t]$

```
coord_cartesian(xlim = c(a, b), ylim = c(s, t))
```

- Exchange x-axis and y-axis

```
coord_flip()
```

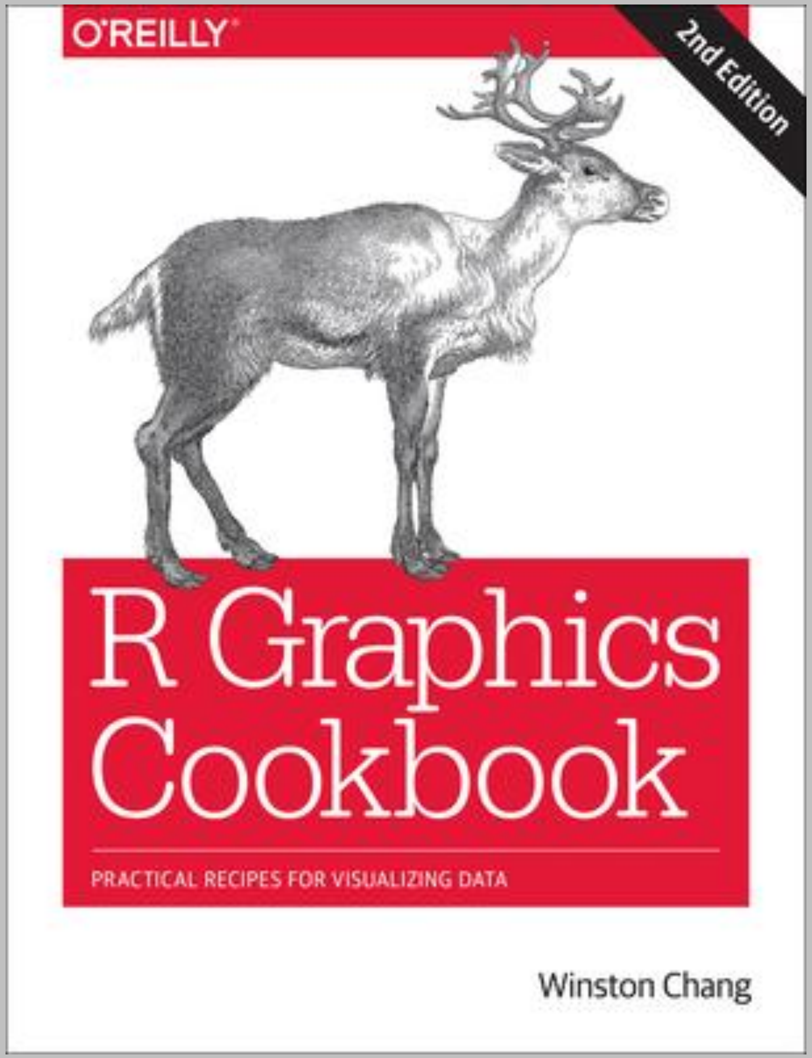
The R Graph Gallery

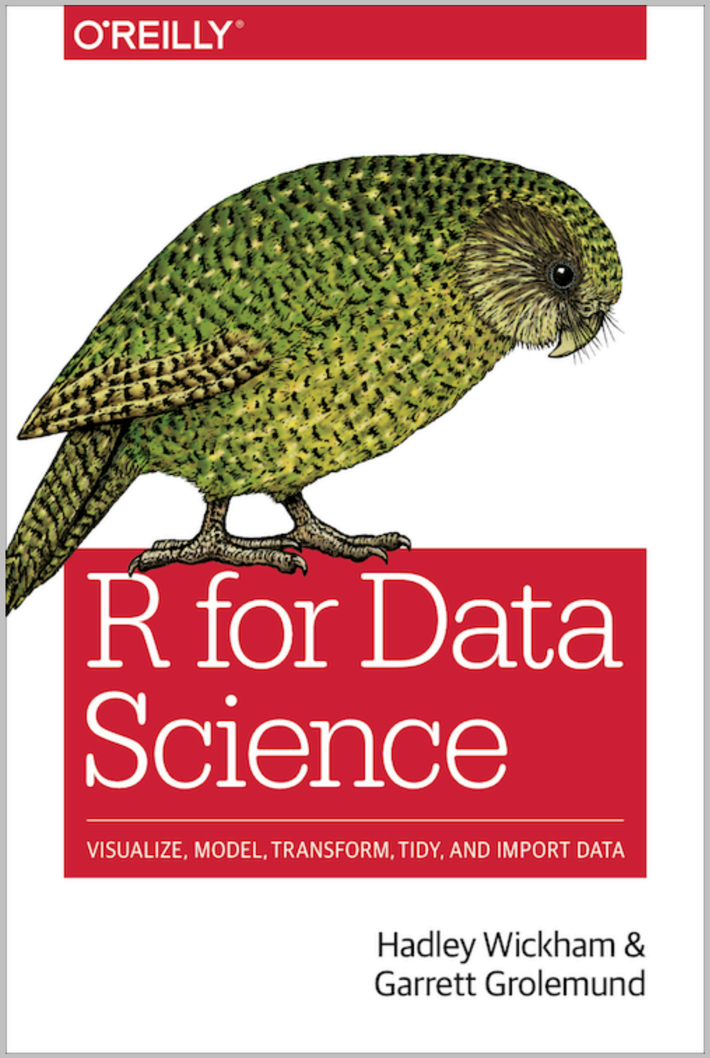
- Visualization examples using R

  `https://www.r-graph-gallery.com/`

References

Wickham, Hadley. *ggplot2: Elegant Graphics for Data Analysis*, 3rd ed. (work in progress) https://ggplot2-book.org/





Next class

7. The Grammar of Graphics and ggplot2 (2)