



# 統計学 2

## 5. ggplot2 入門

やない ゆうき  
矢内 勇生



<https://yukiyanai.github.io>



[yanai.yuki@kochi-tech.ac.jp](mailto:yanai.yuki@kochi-tech.ac.jp)



# このトピックの目標

- Rパッケージを利用する方法を理解する
  - ▶ インストール
  - ▶ 読み込み
- データフレームの基本構造を理解する
- ggplot2 を使って図を作る方法をおぼえる
  - ▶ ggplot2 の基本文法を理解する
  - ▶ ggplot2 でよく使う geom の使い方を覚える
  - ▶ ggplot2 でよく使う図のカスタマイズ法を覚える

Rパッケージージ

# Rのパッケージ

- Rの機能は「パッケージ」によって追加できる
- 「こんな機能ないかな？」と思ったことのほとんどは、パッケージによって提供されている
- さまざまなパッケージを使いこなすことが、効率的な分析の鍵
- この授業でも、いくつかのパッケージを使う

# パッケージのインストール

- CRAN に登録されているパッケージは、簡単にインストールすることができる
- **コンソールに**、以下のコマンドを入力する（引用符が必要）

```
install.packages('パッケージ名')
```

- パッケージは1度インストールすれば、（Rのバージョンを変えない限り）インストールする必要はない

# パッケージの読み込み

- インストール済みのパッケージは、以下のコマンドで読み込むことができる

```
library(パッケージ名)
```

- パッケージの読み込みは、R (RStudio) を起動する度に実行することが必要
- R Markdownファイルでパッケージを利用するときは、そのファイルの中に読み込みのコマンドを書かなければいけない
  - ▶ R Markdown の上部（最初のほう）に `library()` をま  
とめて書いておく

# パッケージの一部を利用する

- インストール済みのパッケージは、`library()` で全体を読み込まなくても、関数ごとに利用することができる
- たとえば、`stringr` という名前のパッケージに含まれる、`str_c()` という関数は

```
stringr::str_c()
```

とすれば利用することができる

- このように、コロン2つを利用してパッケージに含まれる関数を直接呼び出すことができる

# 名前の衝突

- パッケージは誰でも作れるので、異なるパッケージに完全に同じ名前の（中身は異なる）関数が含まれていることがある
  - ▶ たとえば、`MASS` と `dplyr` という2つのパッケージは、どちらも `select()` という関数をもっている
- `library(MASS)` と `library(dplyr)` をどちらも実行し、`select()` を使うと、**後から** `library()` したパッケージの `select()` が呼び出される
- 自分が利用したいパッケージを指定するためには、`MASS::select()` や `dplyr::select()` などのようにする



**data.frame**

# 長方形データ (Rectangular Data)

- 最も一般的なデータの「形」は、長方形 (表形式)
- 各行 (row) が観測単位1つを表す
  - ▶ 例：右の図では、「候補者」が1つの行
- 各列 (column) が1つの変数を表す
- 各セル (cell; 行と列の組) が値 (数値または文字列) を持つ

	A	B	C	D	E	F	G	H	w
1	year	ku	kun	party	name	age	status	nocand	
2	1996	aichi	1	1000	KAWAMURA, TAKASHI	47	2	7	
3	1996	aichi	1	800	IMAEDA, NORIO	72	3	7	
4	1996	aichi	1	1001	SATO, TAISUKE	53	2	7	
5	1996	aichi	1	305	IWANAKA, MIHOKO	43	1	7	
6	1996	aichi	1	1014	ITO, MASAKO	51	1	7	
7	1996	aichi	1	1038	YAMADA, HIROSHIB	51	1	7	
8	1996	aichi	1	1	ASANO, KOSETSU	45	1	7	
9	1996	aichi	2	1000	AOKI, HIROYUKI	51	2	8	
10	1996	aichi	2	800	TANABE, HIROO	71	3	8	
11	1996	aichi	2	1001	FURUKAWA, MOTOHISA	30	1	8	
12	1996	aichi	2	305	ISHIYAMA, JYUNICHI	31	1	8	
13	1996	aichi	2	1003	FUJIWARA, MICHIKO	44	1	8	
14	1996	aichi	2	1014	ISHIKAWA, KAZUMI	61	1	8	
15	1996	aichi	2	1	MURAMATSU, YOICHI	47	1	8	
16	1996	aichi	2	1038	YAMAZAKI, YOSHIAKI	43	1	8	
17	1996	aichi	3	1000	YOSHIDA, YUKIHIRO	35	1	7	
18	1996	aichi	3	800	KATAOKA, TAKESHI	46	2	7	
19	1996	aichi	3	1001	KONDO, SHOICHA	38	1	7	
20	1996	aichi	3	305	YANAGIDA, SAEKO	50	1	7	
21	1996	aichi	3	1038	NAKANO, YOKO	54	1	7	
22	1996	aichi	3	1014	OGAWA, OSAMU	35	1	7	
23	1996	aichi	3	1	ATOJI, MASAO	43	1	7	
24	1996	aichi	4	1000	MISAWA, JUN	44	1	6	
25	1996	aichi	4	800	TSUKAMOTO, SABURO	69	3	6	
26	1996	aichi	4	305	SEKO, YUKIKO	49	1	6	
27	1996	aichi	4	1001	TAKAGI, HIROSHI	43	1	6	
28	1996	aichi	4	1038	ITO, TAKAYOSHI	61	1	6	
29	1996	aichi	4	1014	SHIOKAWA, CHIKANAO	40	1	6	

図：浅野・矢内 (2018) の衆院選データ

# CSV ファイル

- CSV: Comma Separated Values (カンマ区切りのファイル)
  - ▶ テキストファイル
  - ▶ 汎用性が高い
    - LibreOffice Calc やMS Excel などの表計算ソフトで編集可能
    - すべてのデータ分析ソフト (アプリ) で開ける
  - ▶ **すべてのデータセットをCSV形式で保存しよう!**
    - 再現性の確保: 他人のため、将来のため

# CSVファイルの例：hr96-17.csv

- テキストエディタで開いた場合

```
1 year,ku,kun,status,name,party,party_code,previous,w1,voteshare,age,nocand,rank,vote,
2 1996,aichi,1,1,"KAWAMURA, TAKASHI",NFP,8,2,1,40,47,7,1,66876,346774,49.2,9828097
3 1996,aichi,1,2,"IMAEDA, NORIO",LDP,1,3,0,25.7,72,7,2,42969,346774,49.2,9311555
4 1996,aichi,1,1,"SATO, TAISUKE",DPJ,3,2,0,20.1,53,7,3,33503,346774,49.2,9231284
5 1996,aichi,1,0,"IWANAKA, MIHOKO",JCP,2,0,0,13.3,43,7,4,22209,346774,49.2,2177203
6 1996,aichi,1,0,"ITO, MASAKO",others,100,0,0,0.4,51,7,5,616,346774,49.2,.
7 1996,aichi,1,0,"YAMADA, HIROSHIB",kokuminto,22,0,0,0.3,51,7,6,566,346774,49.2,.
8 1996,aichi,1,0,"ASANO, KOSETSU",independent,99,0,0,0.2,45,7,7,312,346774,49.2,.
9 1996,aichi,2,1,"AOKI, HIROYUKI",NFP,8,1,1,32.9,51,8,1,56101,338310,51.8,12940178
10 1996,aichi,2,2,"TANABE, HIROO",LDP,1,1,0,26.4,71,8,2,44938,338310,51.8,16512426
11 1996,aichi,2,0,"FURUKAWA, MOTOHISA",DPJ,3,0,2,25.7,30,8,3,43804,338310,51.8,11435567
12 1996,aichi,2,0,"ISHIYAMA, JUNICHI",JCP,2,0,0,12.5,31,8,4,21337,338310,51.8,2128510
13 1996,aichi,2,0,"FUJIWARA, MICHIKO",jiyu-rengo,10,0,0,1.6,44,8,5,2670,338310,51.8,327
14 1996,aichi,2,0,"ISHIKAWA, KAZUMI",others,100,0,0,0.4,61,8,6,701,338310,51.8,.
15 1996,aichi,2,0,"MURAMATSU, YOICHI",independent,99,0,0,0.2,47,8,7,418,338310,51.8,.
16 1996,aichi,2,0,"YAMAZAKI, YOSHIAKI",kokuminto,22,0,0,0.2,43,8,8,348,338310,51.8,.
17 1996,aichi,3,0,"YOSHIDA, YUKIHIRO",NFP,8,1,1,32.3,35,7,1,52478,331808,50.4,11245219
18 1996,aichi,3,1,"KATAOKA, TAKESHI",LDP,1,3,0,27,46,7,2,43884,331808,50.4,5365436
19 1996,aichi,3,0,"KONDO, SHOICHA",DPJ,3,1,2,23.6,38,7,3,38351,331808,50.4,11767342
20 1996,aichi,3,0,"YANAGIDA, SAEKO",JCP,2,0,0,16.1,50,7,4,26225,331808,50.4,2110540
21 1996,aichi,3,0,"NAKANO, YOKO",kokuminto,22,0,0,0.5,54,7,5,773,331808,50.4,.
22 1996,aichi,3,0,"OGAWA, OSAMU",others,100,0,0,0.4,35,7,6,722,331808,50.4,.
23 1996,aichi,3,0,"ITO, TADAHIKO",independent,99,0,0,0.2,43,7,7,246,331808,50.4,.
24 1996,aichi,4,0,"MISAWA, JUN",NFP,8,1,1,35.7,44,6,1,57361,315704,52,12134215
25 1996,aichi,4,2,"TSUKAMOTO, SABURO",LDP,1,10,0,30,69,6,2,48209,315704,52,13303388
26 1996,aichi,4,0,"SEKO, YUKIKO",JCP,2,1,2,19.3,49,6,3,30976,315704,52,3152034
27 1996,aichi,4,0,"TAKAGI, HIROSHI",DPJ,3,0,0,14.6,43,6,4,23411,315704,52,6461034
28 1996,aichi,4,0,"ITO, TAKAYOSHI",kokuminto,22,0,0,0.2,61,6,5,348,315704,52,.
29 1996,aichi,4,0,"SHIOKAWA, CHIKANAO",others,100,0,0,0.2,40,6,6,243,315704,52,.
30 1996,aichi,5,1,"AKAMATSU, HIROTAKA",DPJ,3,2,1,30.9,48,7,1,48648,319846,50.3,11894801
31 1996,aichi,5,0,"KIMURA, TAKAHIDE",LDP,1,1,2,29.5,41,7,2,46485,319846,50.3,5010946
32 1996,aichi,5,0,"BANNO, YUTAKA",NFP,8,0,0,27.3,35,7,3,43028,319846,50.3,9817826
33 1996,aichi,5,0,"NAGATOMO, TADAHIRO",JCP,2,0,0,11.2,37,7,4,17670,319846,50.3,2728093
34 1996,aichi,5,0,"SUZUKI, TSUTOMU",independent,99,0,0,0.5,63,7,5,768,319846,50.3,70000
35 1996,aichi,5,0,"ICHIKAWA, MINORU",kokuminto,22,0,0,0.3,56,7,6,470,319846,50.3,.
36 1996,aichi,5,0,"UCHIYAMA, TAKAO",others,100,0,0,0.2,43,7,7,335,319846,50.3,.
37 1996,aichi,6,1,"KUSAKAWA, SHOZO",NFP,8,8,1,39.7,68,8,1,90812,433930,54.2,11252336
38 1996,aichi,6,0,"ITO, KATSUNDO",LDP,1,0,0,26.1,51,8,2,59631,433930,54.2,10987232
```

# CSVファイルの例 : hr96-17.csv

	A	B	C	D	E	F	G	H	I	J	K	L
1	year	ku	<u>kun</u>	party	name	age	status	<u>nocand</u>	<u>wl</u>	rank	previous	vote
2	1996	<u>aichi</u>	1	1000	KAWAMURA, TAKASHI	47	2	7	1	1	2	66
3	1996	<u>aichi</u>	1	800	IMAEDA, NORIO	72	3	7	0	2	3	42
4	1996	<u>aichi</u>	1	1001	SATO, TAISUKE	53	2	7	0	3	2	33
5	1996	<u>aichi</u>	1	305	IWANAKA, MIHOKO	43	1	7	0	4	0	22
6	1996	<u>aichi</u>	1	1014	ITO, MASAKO	51	1	7	0	5	0	
7	1996	<u>aichi</u>	1	1038	YAMADA, HIROSHIB	51	1	7	0	6	0	
8	1996	<u>aichi</u>	1	1	ASANO, KOSETSU	45	1	7	0	7	0	
9	1996	<u>aichi</u>	2	1000	AOKI, HIROYUKI	51	2	8	1	1	2	56
10	1996	<u>aichi</u>	2	800	TANABE, HIROO	71	3	8	0	2	1	44
11	1996	<u>aichi</u>	2	1001	FURUKAWA, MOTOHISA	30	1	8	2	3	1	43
12	1996	<u>aichi</u>	2	305	ISHIYAMA, JYUNICHI	31	1	8	0	4	0	21
13	1996	<u>aichi</u>	2	1003	FUJIWARA, MICHIKO	44	1	8	0	5	0	2
14	1996	<u>aichi</u>	2	1014	ISHIKAWA, KAZUMI	61	1	8	0	6	0	
15	1996	<u>aichi</u>	2	1	MURAMATSU, YOICHI	47	1	8	0	7	0	
16	1996	<u>aichi</u>	2	1038	YAMAZAKI, YOSHIAKI	43	1	8	0	8	0	
17	1996	<u>aichi</u>	3	1000	YOSHIDA, YUKIHIRO	35	1	7	1	1	1	52
18	1996	<u>aichi</u>	3	800	KATAOKA, TAKESHI	46	2	7	0	2	3	43
19	1996	<u>aichi</u>	3	1001	KONDO, SHOICHA	38	1	7	2	3	1	38

- 表計算ソフト（LibreOffice Calc）で開いた場合

# data.frame

- Rでよく使われるデータの形式
- ggplot2 で作図するためには、この形式のデータが必要
- 長方形データをCSV ファイルでもっている場合
  - ▶ `read.csv()` または `readr::read_csv()` でファイルを読み込めば、自動的に `data.frame` ができる
- 自分で作る場合
  - ▶ `data.frame()` または `tibble::tibble()` で作れる
- 他の形式（たとえば、`matrix`）でデータを持っている場合
  - ▶ `as.data.frame()` または `tibble::as_tibble()` で変換できる

**ggplot とは？**



# で作図

- Rの特徴：綺麗なグラフが簡単に描ける
- Rが用意する作図用関数の例
  - ▶ ヒストグラム：`hist()`
  - ▶ 棒グラフ：`barplot()`
  - ▶ 箱ひげ図：`boxplot()`
  - ▶ 折れ線グラフ：`matplot()`
  - ▶ 散布図：`plot()`
  - ▶ 曲線：`curve()`



# ggplot2 を使おう

- 組み込み関数でも綺麗な図は作れる
  - ▶ 図の種類ごとに異なる関数：覚えるのが面倒
  - ▶ 細かい調整が困難（熟練の技が必要）
- ggplot2 なら簡単に綺麗な図が作れる
- Rを使っていれば、いずれはggplot2 を使う
- ★ 初めから ggplot2 を使おう！

# ggplot2 とは？



- データ可視化のためのRパッケージ
- **g**rammar of **g**raphics
- **綺麗**な図が、**簡単**に描ける
  - ▶ 文法 (grammar) を覚えれば、**一貫した方法**で作図ができる

# tidyverse を使おう



- tidy + universe
- データサイエンスにとって重要なパッケージの集合体
- 主なパッケージ : **ggplot2**, dplyr, tidyr, readr, purrr, tibble,
- 詳しくは : <https://www.tidyverse.org/>

# Rの基礎知識：Hadley Wickham

- RStudioのチーフサイエンティスト, 統計学者
- 通称：羽鳥先生, 神
- 現在のRに欠かせない多数のパッケージを開発：  
ggplot2 も Hadley が開発
  - ▶ ggplot2 の解説書は無料で読める：[https://  
ggplot2-book.org/](https://ggplot2-book.org/)
- 詳細：<http://hadley.nz/>

**ggplot を使う準備**

# インストール

- tidyverse パッケージをインストールする
  - ▶ Rで以下を実行（一度実行すれば次回から実行の必要はないので、コンソール [Console] に直接入力する）

```
install.packages('tidyverse', dependencies = TRUE)
```

- ▶ 複数のパッケージがインストールされるので、それなりに時間がかかる
- **Topic 3 の実習で実施済みのはず：あらためてインストールする必要はない**

# 読み込み

- tidyverse パッケージを `library()` で読み込む
  - ▶ Rで以下を実行 (**Rを起動するたびに実行する必要がある**ので、RスクリプトまたはRマークダウンに保存して実行する)

```
library(tidyverse)
```

# 図のラベル等に使うフォントの設定

- サンプルとして配布したファイル  
(stat2\_template2021v2.Rmd) の1行目から75行目  
までをコピーすればOK
  - ▶ 以下のフォントが設定される
    - macOS : ヒラギノゴシック
    - Windows : 游ゴシック
    - Linux : IPAexゴシック (このフォントがインストールされていない場合は、使えるフォントを指定する)



ggplot2 による作図



- data.frame 型のデータを ggplot() 関数に入力して図を作る！

# data.frame 型のデータ???

- Rにあらかじめ用意されたデータは `data.frame` 型
- 長方形データ（行が観測対象、列が変数のデータ）を `read.csv()` や `readr::read_csv()` で読み込むと、自動的に `data.frame` 型になる
- 自分で作る：`tibble::tibble()` または `data.frame()`
- 行列を `data.frame` へ変換：`tibble::as_tibble()` または `as.data.frame()`

# data.frame 型のデータの作り方

- n: サンプルサイズ
  - x :  $x_i \sim \text{Uniform}(0,1)$  でランダムに生成
  - y:  $y_i \sim \text{Normal}(0.8x_i, \sigma^2 = 1)$  でランダムに生成
  - 2変数 x と y からなる data.frame 型のデータ myd を作る
- ◆以下のコマンドを実行 (tidyverse パッケージは読み込み済みと想定)

```
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)
```

# 組み込みデータを使う

- Rには、いくつかのデータがあらかじめ用意されている
- `data()` で、どんなデータが利用可能か確認できる
- 今回は、`mtcars` と `diamonds` を使う

```
data(mtcars)  
glimpse(mtcars)
```

```
data(diamonds)  
glimpse(diamonds)
```

# ggplot2による作図の基本

1. `ggplot()` 関数にデータを渡し、どのデータを可視化するか指定する
2. `geom_xxx()` で自分が作りたい図の層 (layer) を加える
3. 軸ラベル (`labs`), 凡例 (`legend`), etc. を指定する
4. `plot()` または `print()` で図を表示する

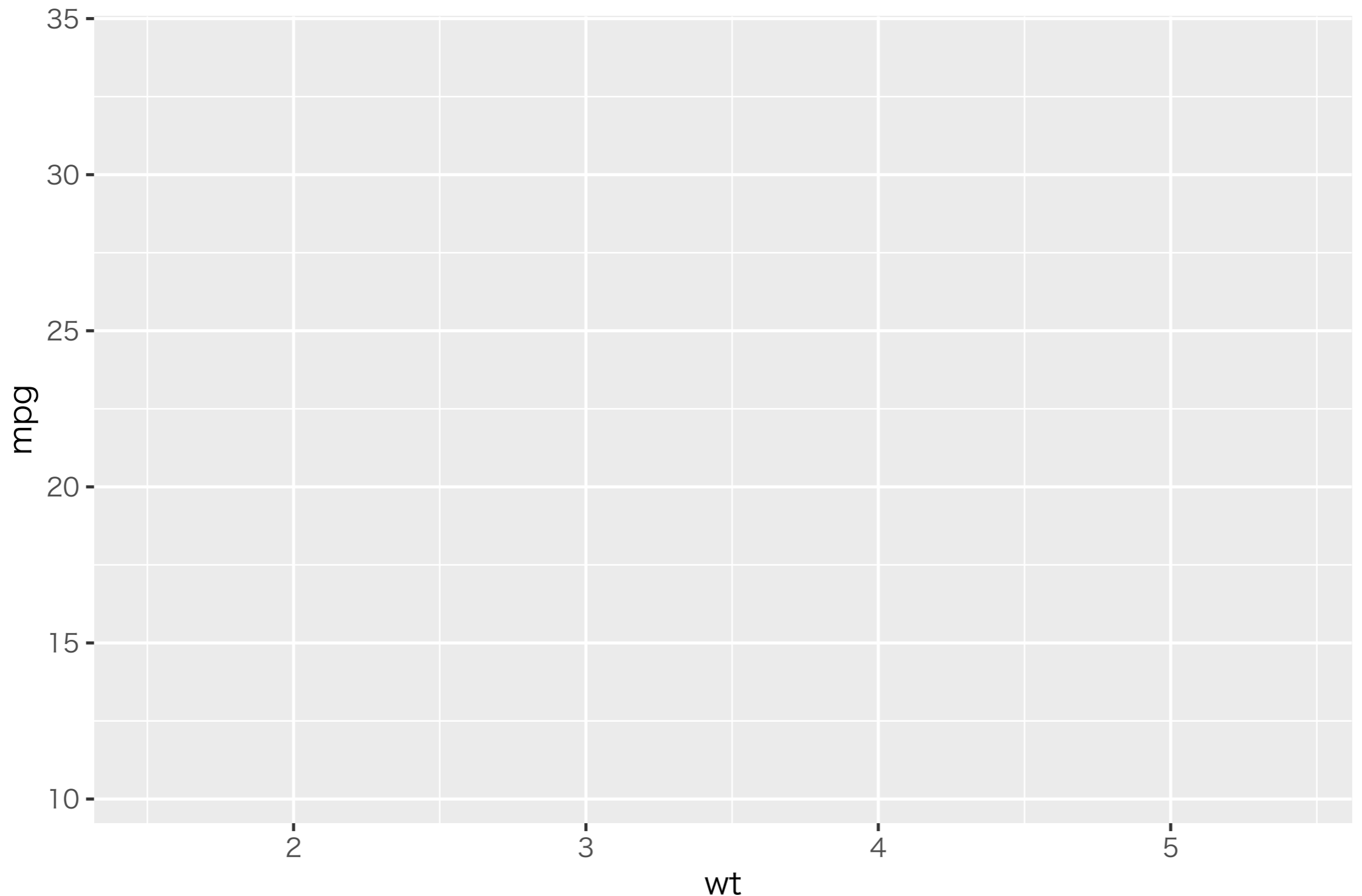
# 1. ggplot()

- 第1引数は data (データフレーム)
- 第2引数は mapping : aes (aesthetics) でデータフレーム内の**どの変数を何のために**使うかを指定する
- 例 : mtcars という名前のデータフレーム内の、wt という変数を横軸 (x軸) に、mpg という変数を縦軸 (y軸) にした図を作る

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

省略することが多い

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





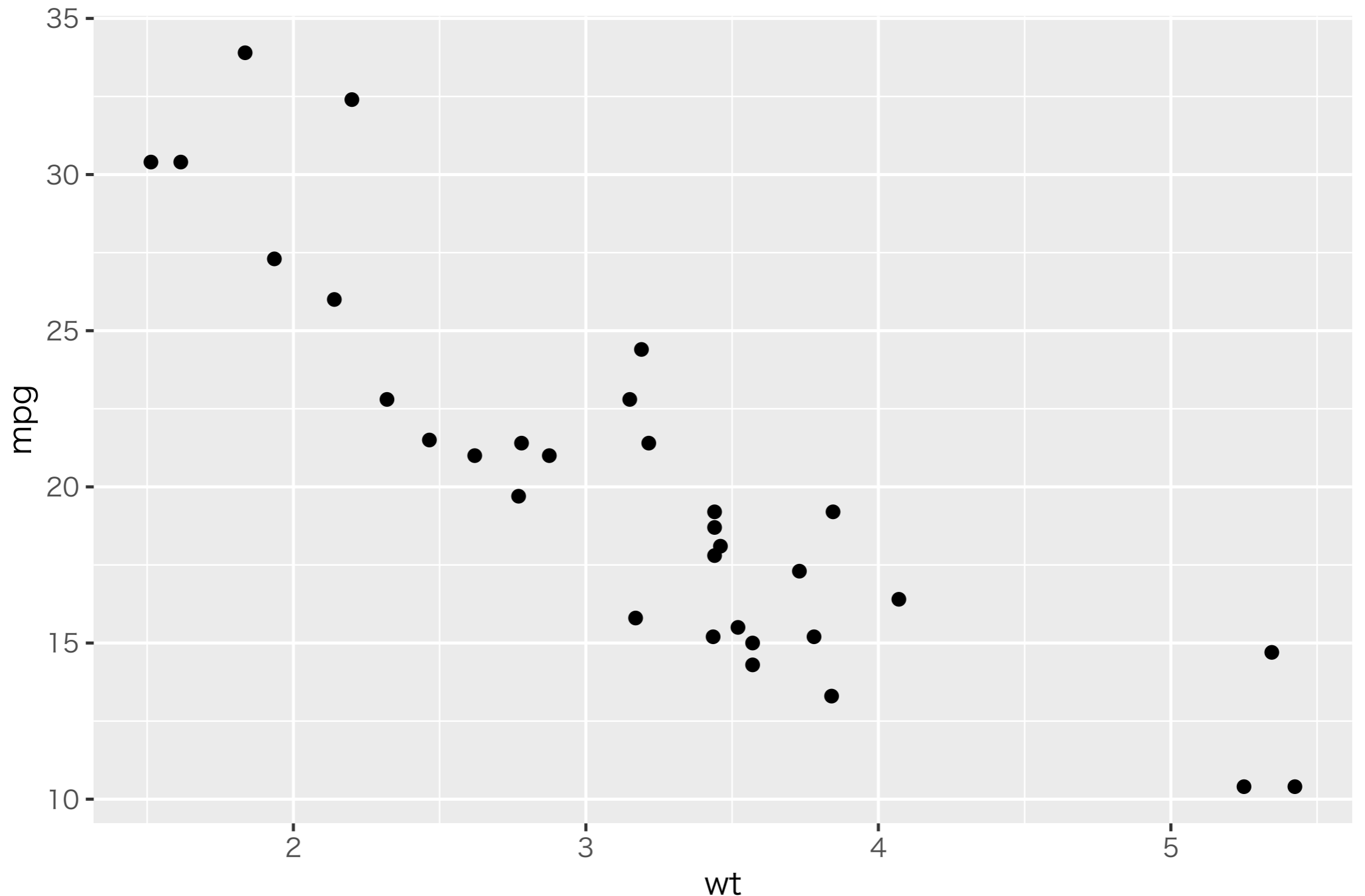
## 2. geom\_xxx()

- geom (geometry) で始まる名前の層を加える
- xxx の部分は、作る図によって変える
  - ▶ ヒストグラム : geom\_histogram()
  - ▶ 散布図 : geom\_point()
- 使う geom によって aes() の中で指定すべきものが変わる

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

← 散布図用のgeom

前のステップで作ったもの

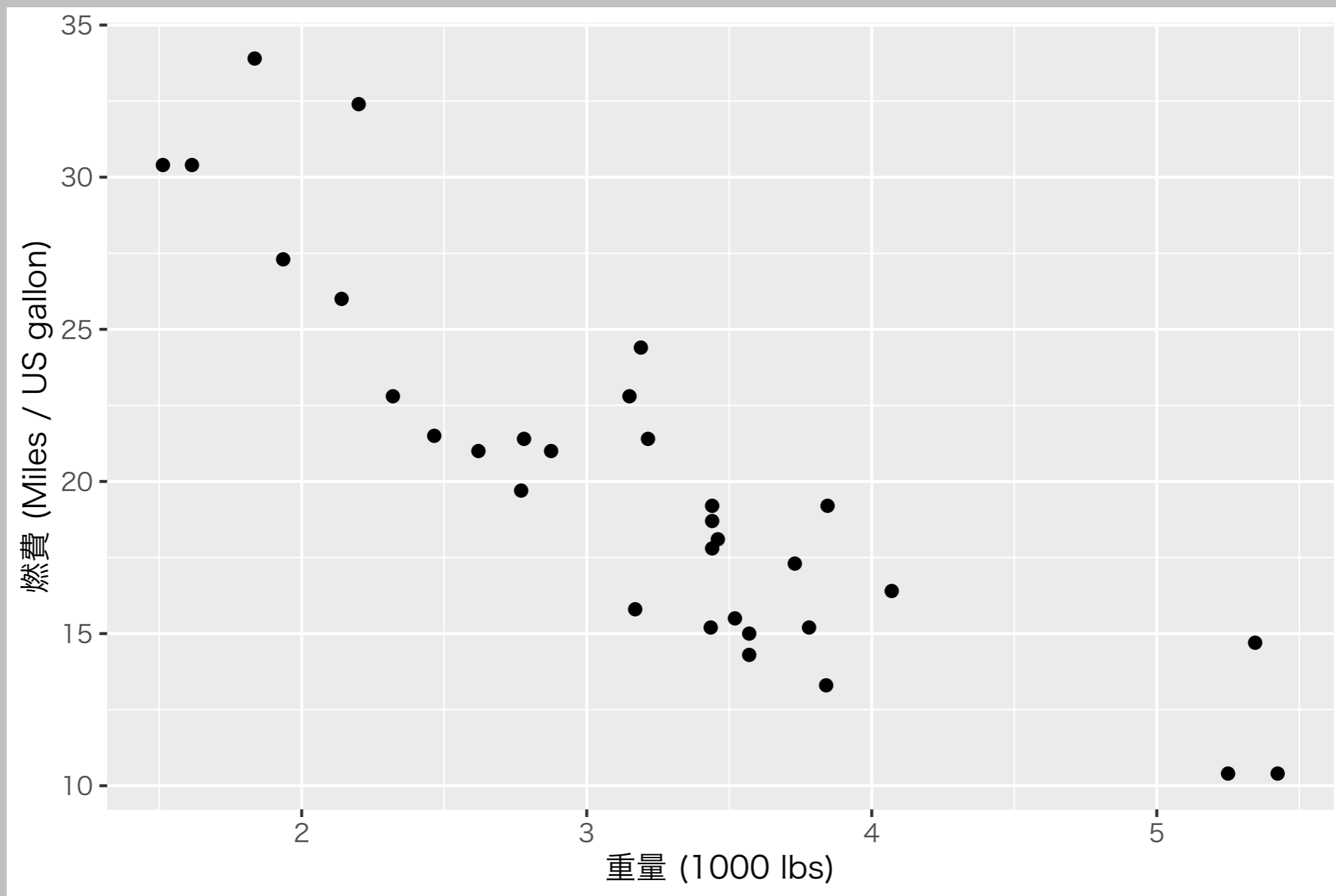


# 3. その他の調整

- 例：`labs()` で軸ラベルを指定する
  - ▶ ラベルは引用符で囲む
  - ▶ 横軸ラベル：`x`
  - ▶ 縦軸ラベル：`y`
  - ▶ 図のタイトル：`title`（不要な場合は指定無しでok）

前のステップで作ったもの

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



# 4. plot() or print()

- ggplot で作った図は、オブジェクトとして保存し、`plot( )` または `print( )` で表示する
- ◆一旦オブジェクトにすることで、再利用が簡単に
  - 図の再確認
  - 複数の図を並べて表示 (`patchwork` パッケージが便利)
  - PDF などのファイルに出力する
- ◆図を表示したいときのみ明示的に `plot()` することで、余計な出力をなくす

# その他の geom

- 授業だけでは紹介しきれないので、以下のサイトを参考に！

<https://ggplot2.tidyverse.org/reference/>

- チートシート：

<https://github.com/rstudio/cheatsheets/raw/master/translations/japanese/ggplot2-cheatsheet-2.0-ja.pdf>

- Kazutan.R にある資料

[https://kazutan.github.io/kazutanR/ggplot2\\_links.html](https://kazutan.github.io/kazutanR/ggplot2_links.html)

# よく使うもの (1)

- $x = a$  の位置に垂線:

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

- $y = b$  の位置に水平線:

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

# よく使うもの (2)

- 可視化の対象を  $x \in [a, b]$  に限定

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

- 可視化の対象を  $y \in [s, t]$  に限定

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

- グラフを描いてから、 $x \in [a, b]$ ,  $y \in [s, t]$  にズームイン

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

- x軸とy軸の入れ替え

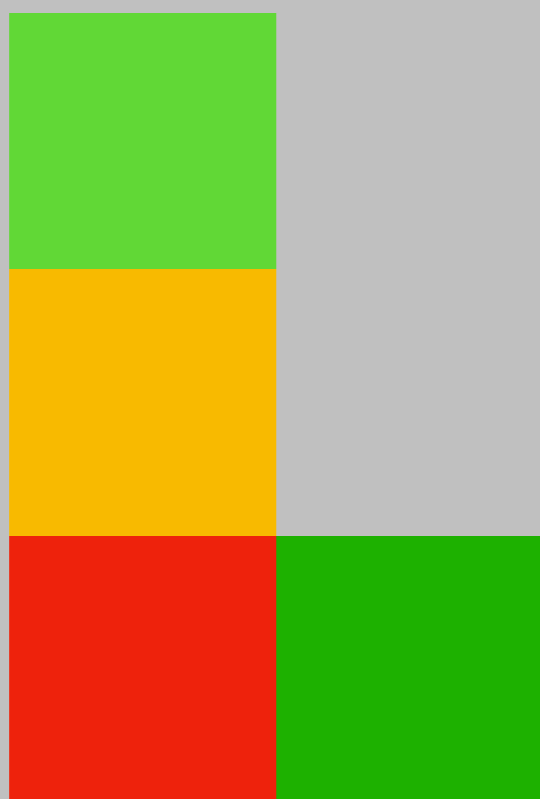
```
coord_flip()
```



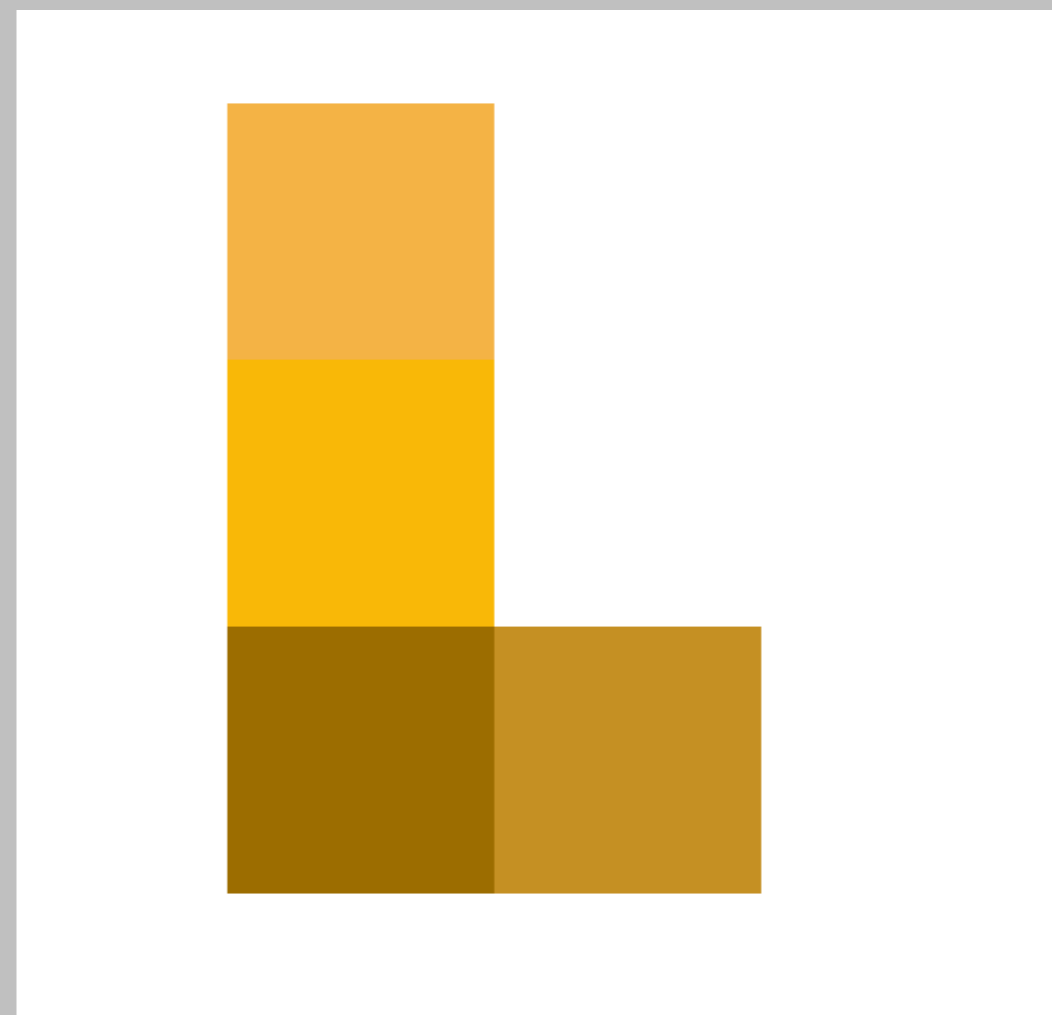
# 色に関する注意

- あなたに見えるように、他の人にも見えているとは限らない
- 色覚障害シミュレーターを使って確認する
  - ▶ macOS: Sim Daltonism (AppStore, 無料)
  - ▶ Win, Linux: Color Oracle (<https://colororacle.org/>)
- 適切なカラーパレットを選ぶ
- 自分の好きな色ではなく、「相手にとって分かりやすい」色を選ぶ

# 注意すべき色使いの例



通常の色



第二色盲 (deuteranopia) の  
シミュレーション

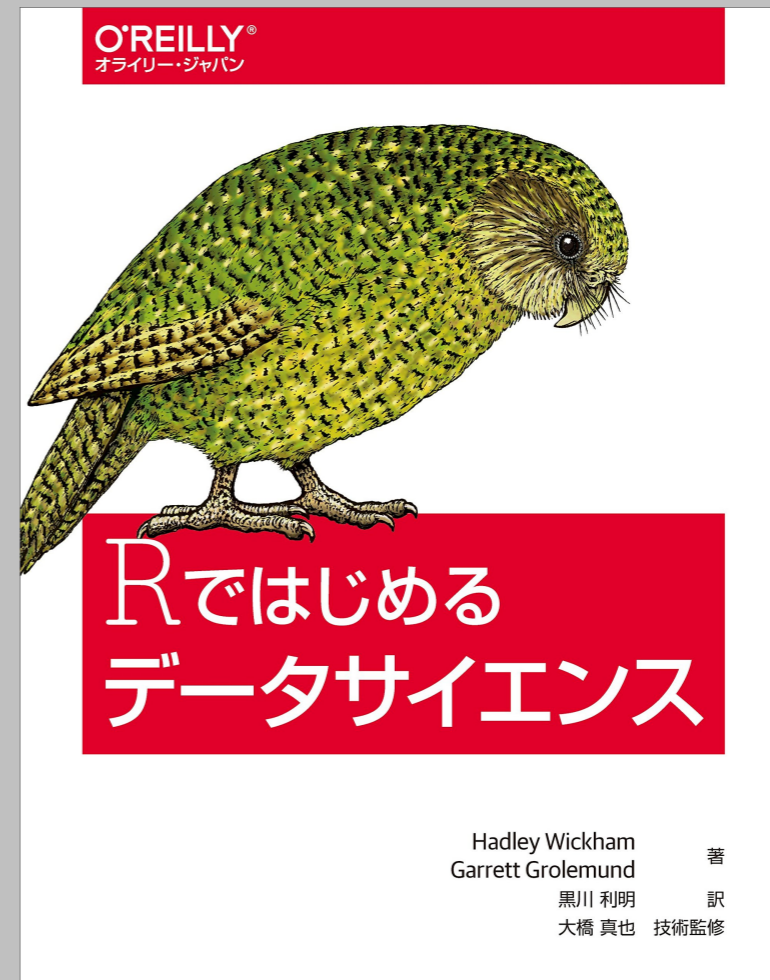
# The R Graph Gallery

- R による可視化の例が見られる


<https://www.r-graph-gallery.com/>

# 参考書

- Wickham, Hadley. *ggplot2: Elegant Graphics for Data Analysis*, 3rd ed. (work in progress) <https://ggplot2-book.org/>
- [副読本](#) の「可視化」の章も参考に



# 参考書

 **【実践Data Science シリーズ】**

データ分析のための

# データ可視化入門

キーラン・ヒーリー 著 瓜生真也 江口哲史 三村喬生 訳

7 6 4 1 0 4 9 3 0 9 8 6 0 1 4 3 6 4 0 1 8 2 5 7 9 3 7 1 8 2 2 4 9 0 0 4 3 8 6 4 1 8 5 7 9 2 7 0 0 2

**全世界のRユーザーが絶賛した  
ベストセラーがついに翻訳!!**

**可視化の奥深さを、すべての人に**

- ggplotとtidyverseの事前知識ゼロでも、すぐに実践できる!
- データを可視化する手順と意図が一冊で学べる!

講談社

永田ゆかり 著  
PATH Data Analytics & Visualization

# データ 視覚化の デザイン

## Data Visualization Design

可視化テク満載! グラフで伝える技術は  
ビッグデータ時代の一般教養だ

株式会社ユーザーローカル 代表取締役社長  
伊藤 将雄氏

漫然とやっていたデータの可視化を体系的に  
学び直せる好著

Japan Digital Design 株式会社 CTO  
楠 正憲氏

この本で可視化の極意を学べば、  
Python使いはあと10年は戦える

「みんなのPython」著者  
柴田 淳氏

達人が語るビジュアライゼーションの極意。

# 実際に作図してみよう！

実習ページ：

- <https://yukiyanai.github.io/jp/classes/stat2/contents/R/intro-to-ggplot2.html>

# 次回予告

6. シミュレーション