The R logo, consisting of a blue "R" inside a gray circle.

# Rによるデータの可視化

## ggplot2 入門

2019年12月5日

矢内 勇生

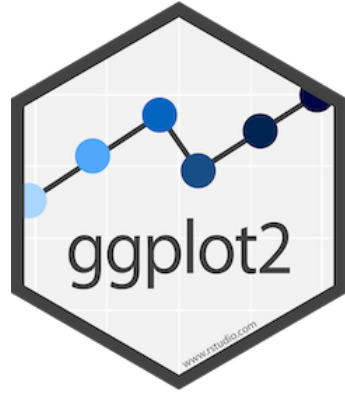
# Intro duction

# 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
  - ▶ ggplot2 の解説書は無料で読める：  
<https://ggplot2-book.org/>
- 詳細：<http://hadley.nz/>



PORepaRation



# 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

# 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

- macOS で図に日本語を使う場合は、次のコードが必要

```
theme_set(theme_gray(base_size = 10,  
                      base_family = "HiraginoSans-W3"))
```

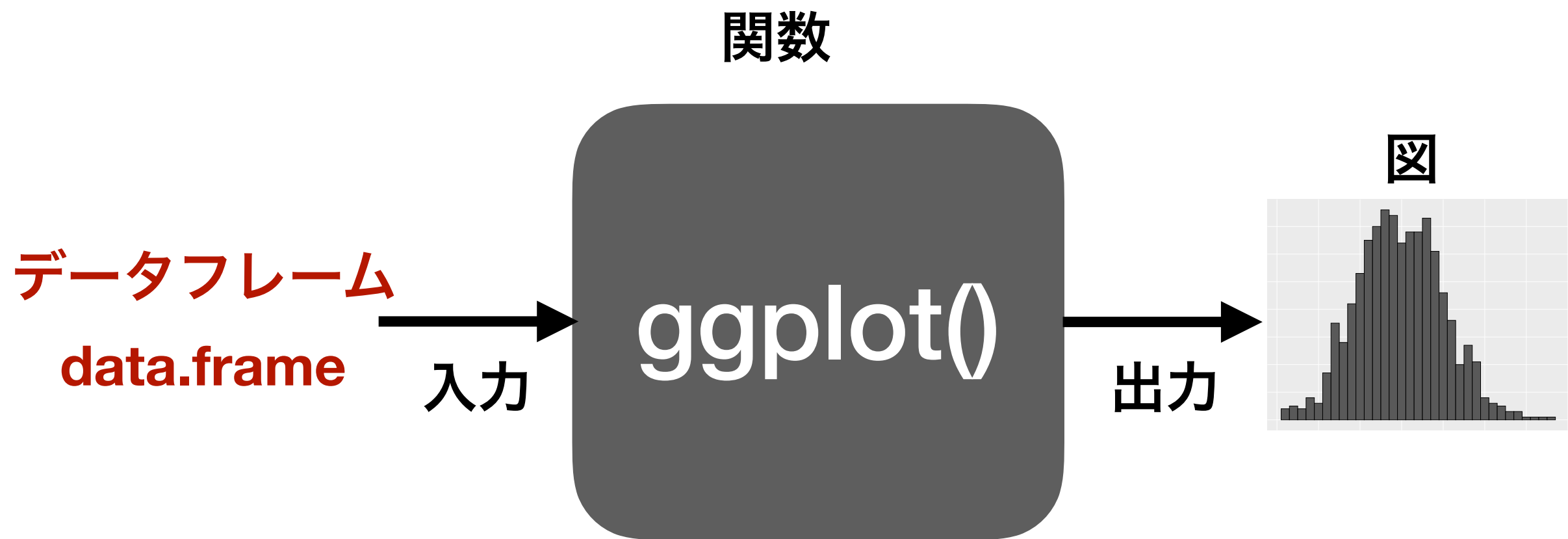
▶ 10pt のヒラギノゴシックを指定

- Windows、Linux では必須ではないが、フォントを指定したければ同じように使う（例：Windows でメイリオ）

```
windowsFonts(MEI = windowsFont("Meiryo"))  
theme_set(theme_gray(base_size = 10,  
                      base_family = "MEI"))
```

The R logo, consisting of a blue capital letter 'R' positioned in front of a gray circle.

**R**eady?



- 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
  - 二変数 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

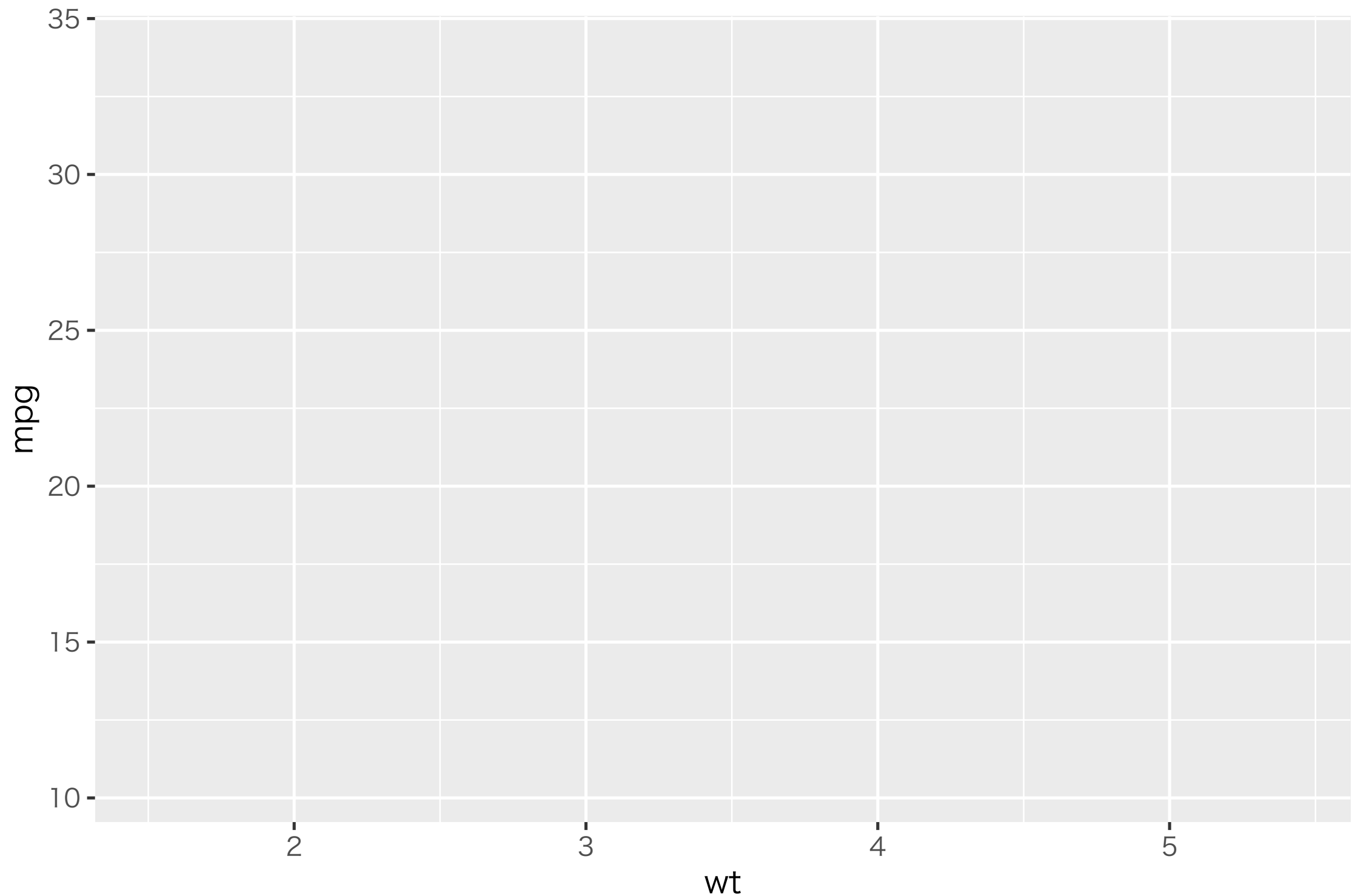
# 1. ggplot()

- 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)
```



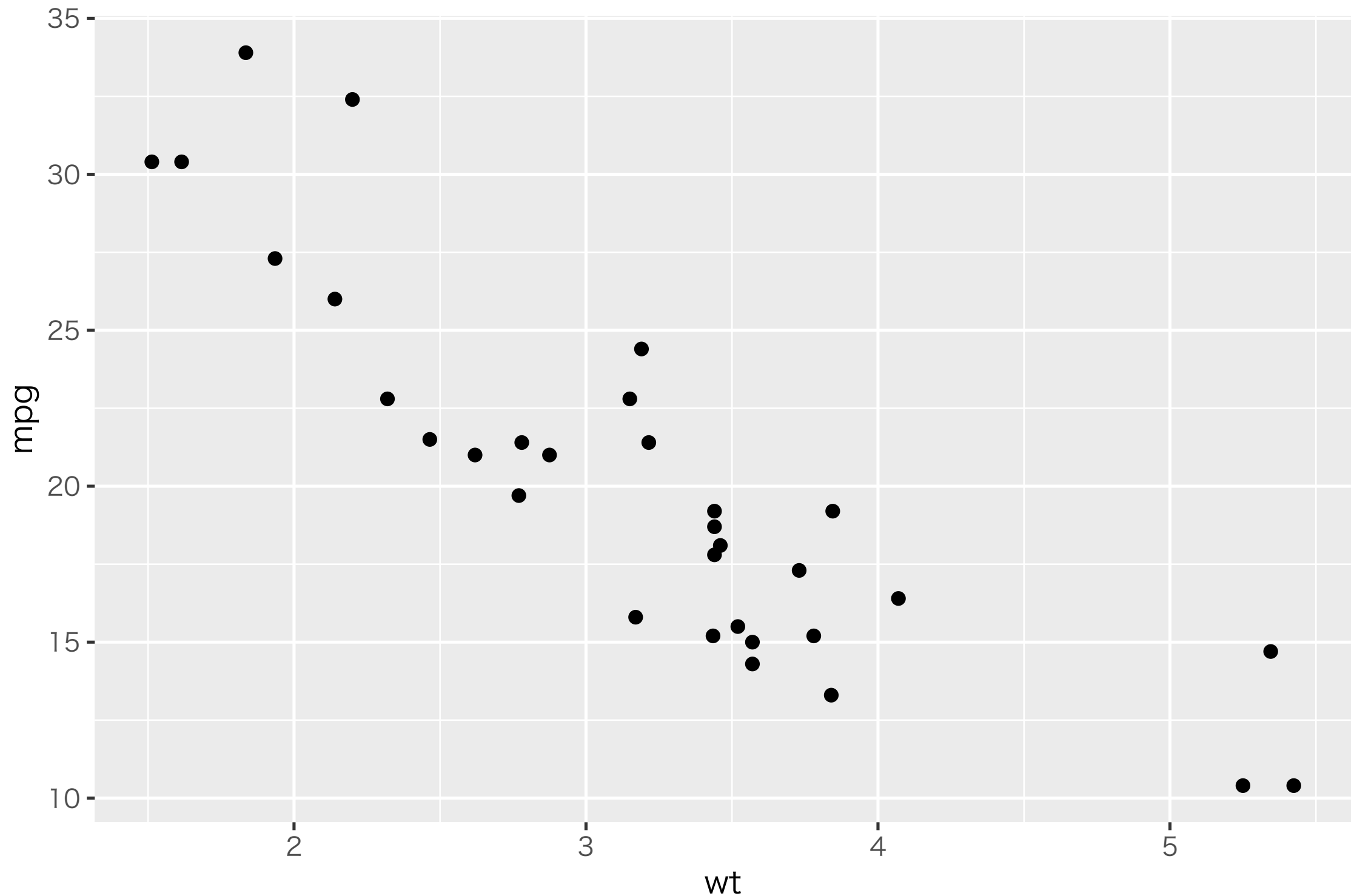
## 2. geom\_xxx()

- 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

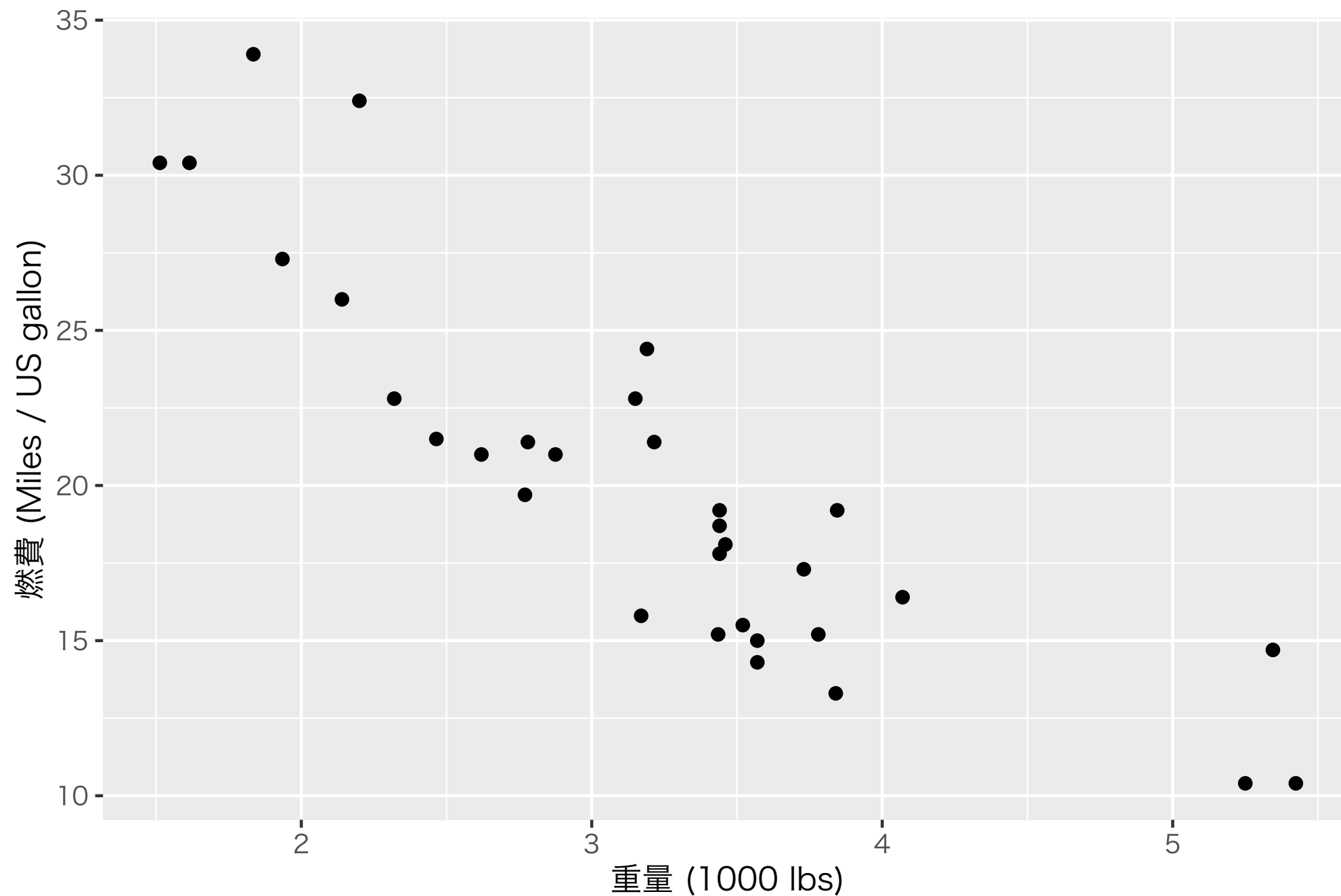


# 3. その他の調整

- 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



# デモンストレーション

参考：

- [http://htmlpreview.github.com/?https://github.com/yukiyanai/KUT\\_R/blob/master/htmls/yanai\\_kutR\\_001.html](http://htmlpreview.github.com/?https://github.com/yukiyanai/KUT_R/blob/master/htmls/yanai_kutR_001.html)
- <https://rstudio.cloud/project/762403>

# その他の geom

- 短時間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))`

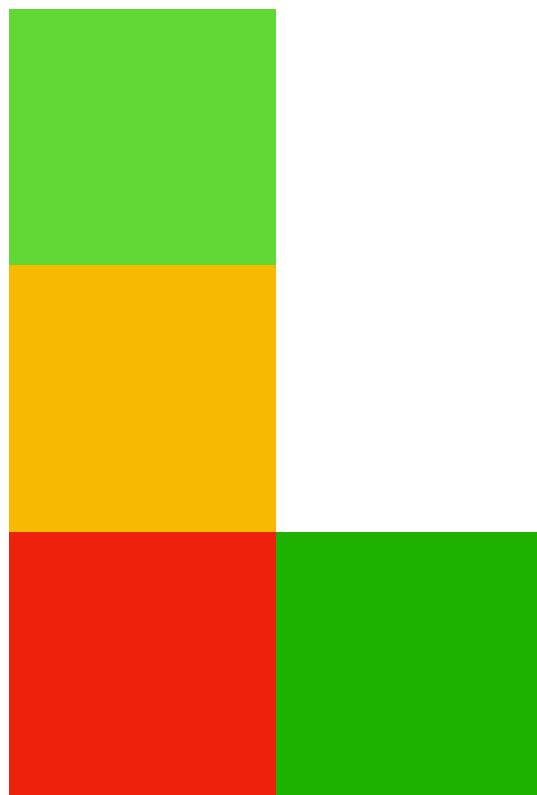
- x軸とy軸の入れ替え（横向きに箱ひげ図が欲しいときなど）

`coord_flip()`

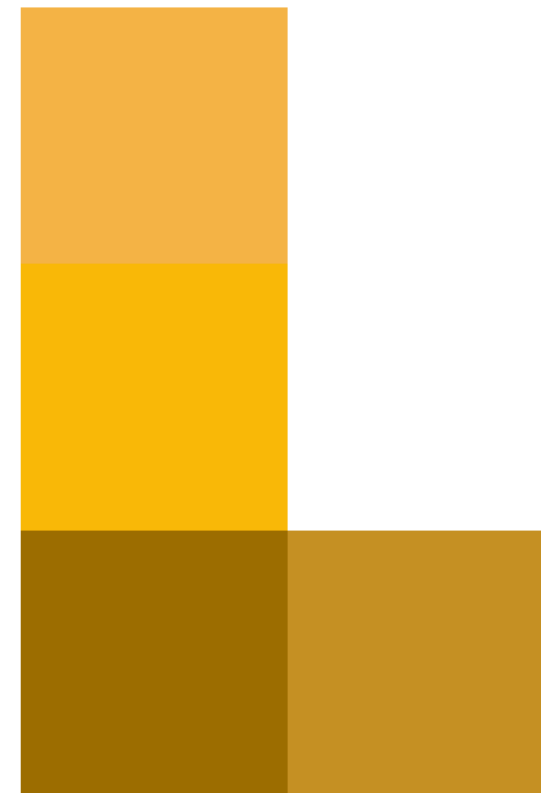
# 色に関する注意

- 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

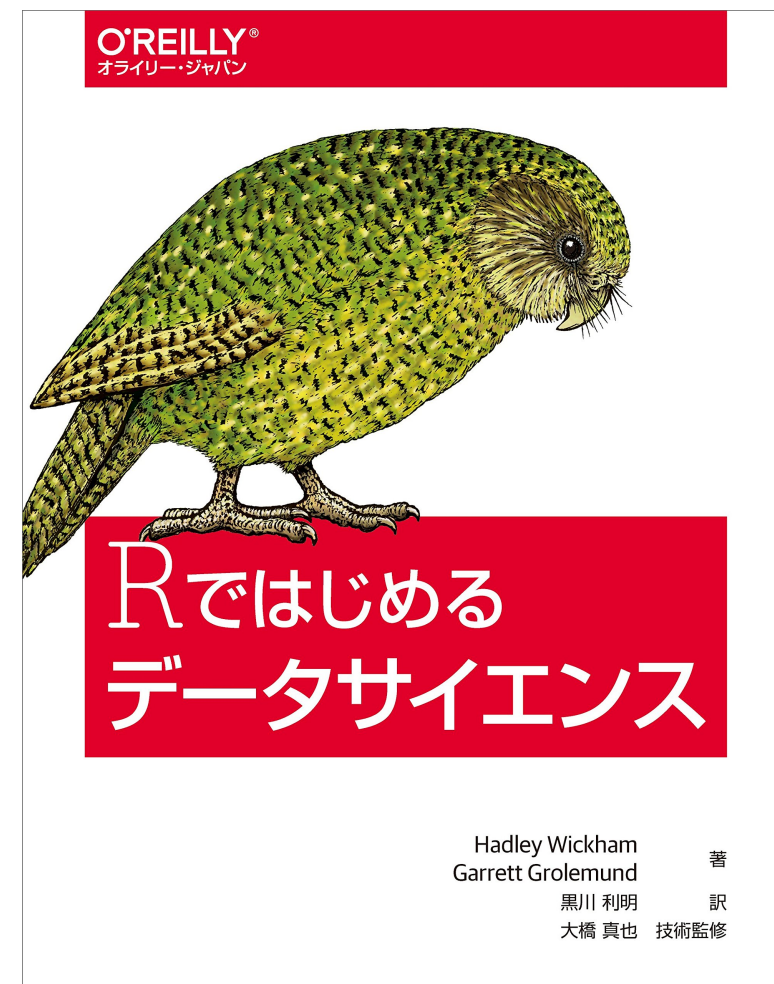
# The R Graph Gallery

- 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/>





Enjoy!

