

# 初心者セッション：データ可視化

1<sup>st</sup> August 2020, Tokyo.R #87  
Yuta Kanzawa @yutakanzawa



Data Science Senior Analyst at Janssen Pharmaceutical K.K., Tokyo  
A Family Company of Johnson & Johnson



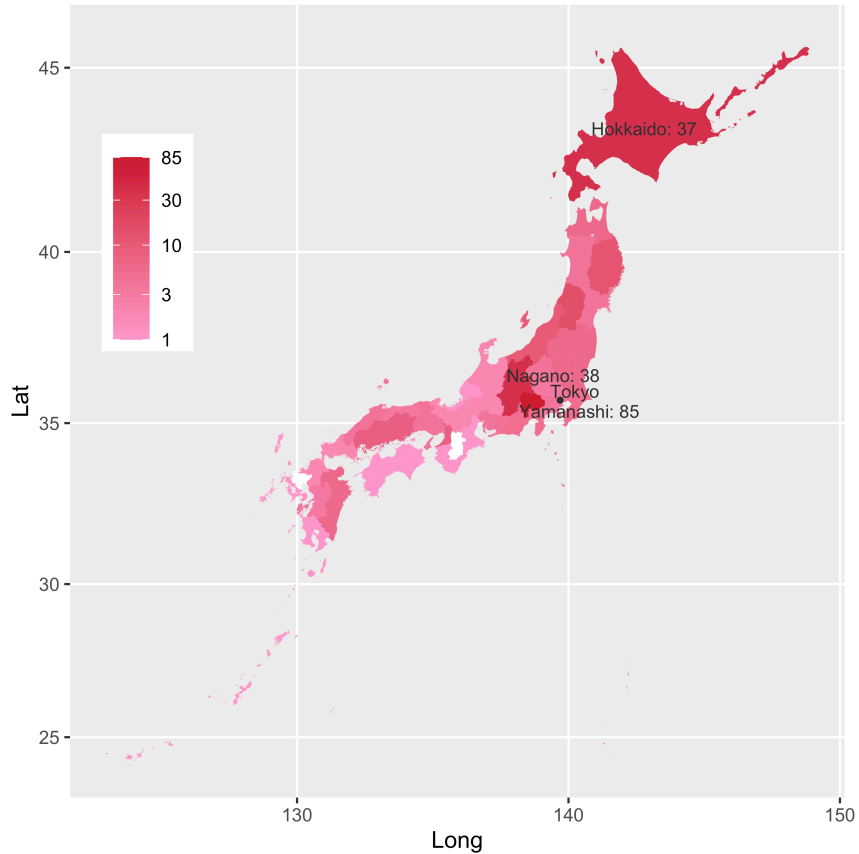
# I am...

- 神沢雄大 **Yuta Kanzawa** (twitter: [@yutakanzawa](https://twitter.com/@yutakanzawa))
- Data scientist at **Janssen Japan**, Tokyo
  - A pharmaceutical company of **J&J**
- Opera & wine lover
  - Wagner
  - Bourgogne
- 7 languages
  - Human: Japanese, English, German
  - Computer: R, Python, SAS, SQL



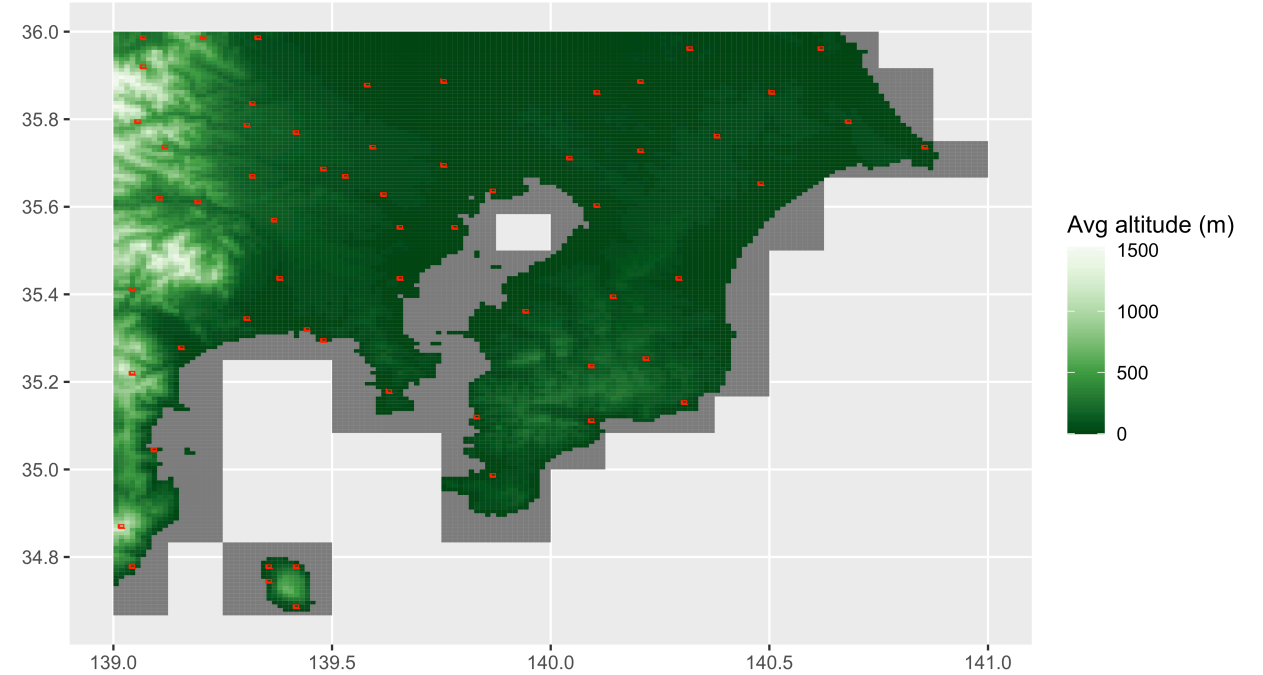
# ポートフォリオ（最近は地図が多い）

Number of Wineries in Japan in 2019, by Prefecture



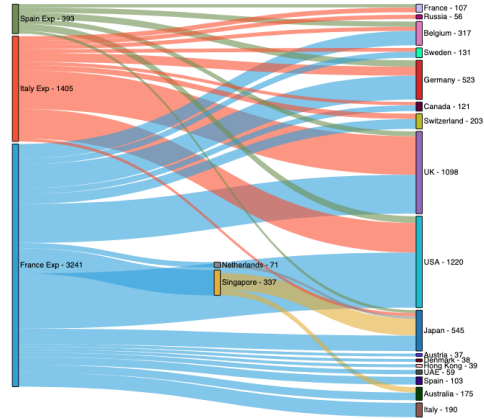
Source: <https://www.nta.go.jp/taxes/sake/shiori-gaikyo/seizogaikyo/kajitsu/pdf/h30/30wine01.pdf>

Avg Altitudes and Weather Observation Stations in Tokyo, Kanagawa, Chiba

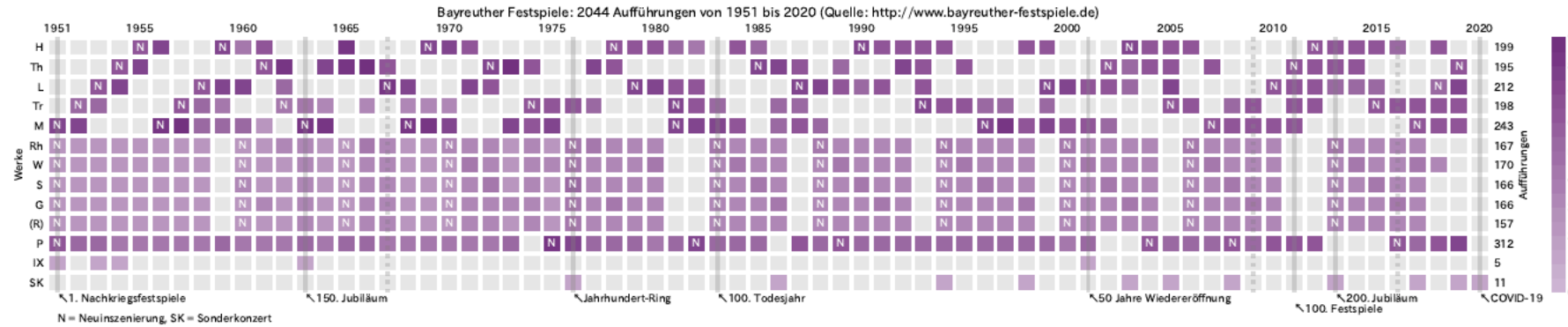
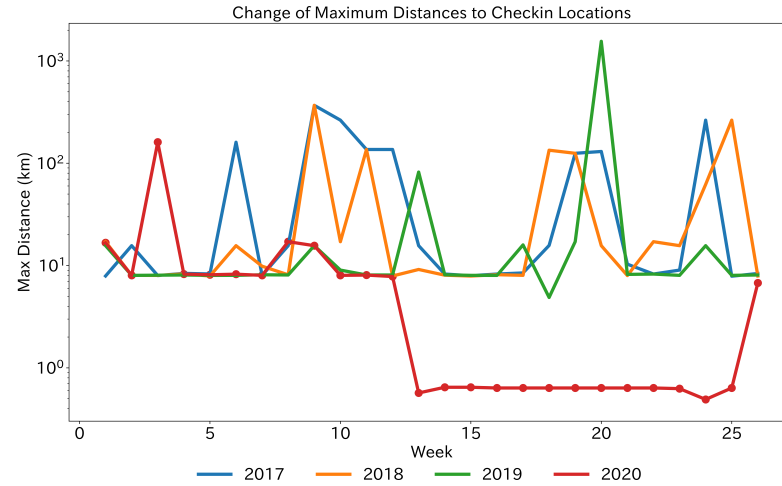


# ポートフォリオ (参考までにR以外も)

World's Top Sparkling Wine Trade Routes in 2018 (Value in million USD)



Based on AAWE's post on Facebook <https://www.facebook.com/wineecon/posts/355758349761764/>



# アジェンダ

- 今日話すこと
  - ggplot2
  - plot()との比較 (少しだけ)
- 対象 (以下のいずれか)
  - ggplot2を初めて触る人
  - 普段plot()を使っている人
  - ggplot2をなんとなく使っている人
- 今日話さないこと
  - 気象観測データ
  - 国土数値情報
  - Python

# TL;DR

- ggplot2でのグラフ描画
  - グラフの要素をそれぞれ指定していく。→レイヤー
- 必須：
  - ggplot() : 全レイヤーに関わる要素を指定
  - aes()\* : 見映えの要素となる変数
  - geom関数 : グラフの種類
- オプション：
  - scale関数 : カラーパレット、凡例、軸の調整

\* 日本語では「エステティック」と表されることもある。

# ggplot2概論

ggplot2 Overview

# その前に：今日知ったこと → Pythonでも使えるらしい👁️👁️

## plotnine

 **Koo@医療職からデータサイエンティストへ**  
@medi\_data0826

pythonでggplotがかける！  
[plotnine.readthedocs.io/en/stable/inde...](https://plotnine.readthedocs.io/en/stable/index.html)

8:14 am · 1 Aug 2020 · Twitter Web App

9 Retweets and comments 22 Likes

plotnine 0.7.0 API Gallery Tutorials Site Page Search

### A Grammar of Graphics for Python

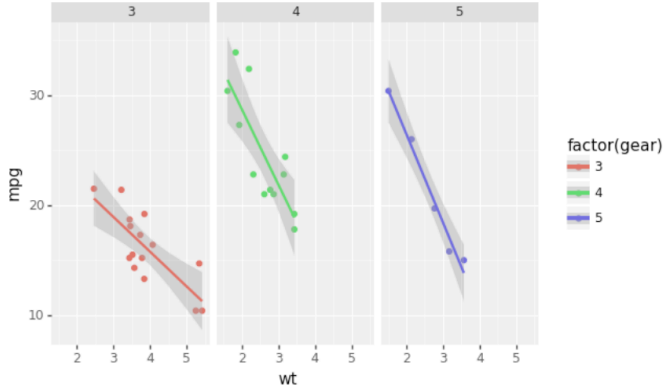
plotnine is an implementation of a *grammar of graphics* in Python, it is based on [ggplot2](#). The grammar allows users to compose plots by explicitly mapping data to the visual objects that make up the plot.

Plotting with a grammar is powerful, it makes custom (and otherwise complex) plots are easy to think about and then create, while the simple plots remain simple.

#### Example

```
from plotnine import ggplot, geom_point, aes, stat_smooth, facet_wrap
from plotnine.data import mtcars

(ggplot(mtcars, aes('wt', 'mpg', color='factor(gear)'))
 + geom_point()
 + stat_smooth(method='lm')
 + facet_wrap('~gear'))
```



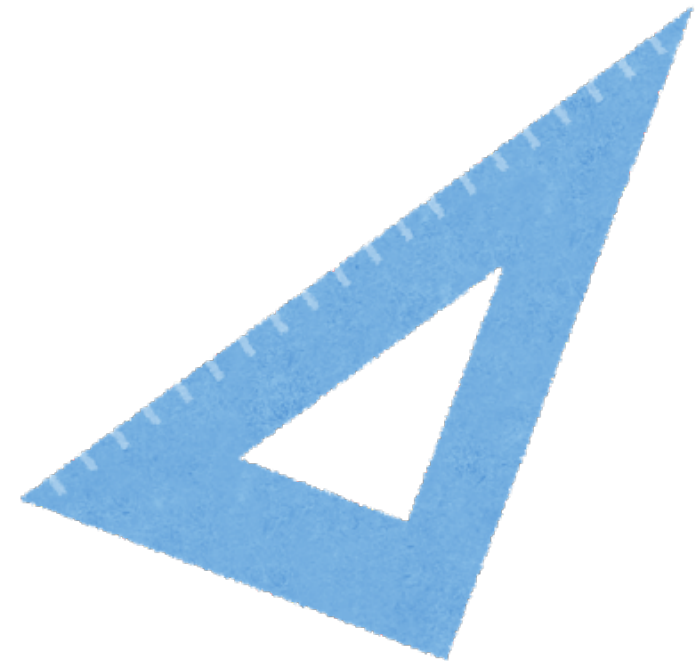
\* [https://twitter.com/medi\\_data0826/status/1289338732349255682](https://twitter.com/medi_data0826/status/1289338732349255682)

\* <https://plotnine.readthedocs.io/en/stable/>



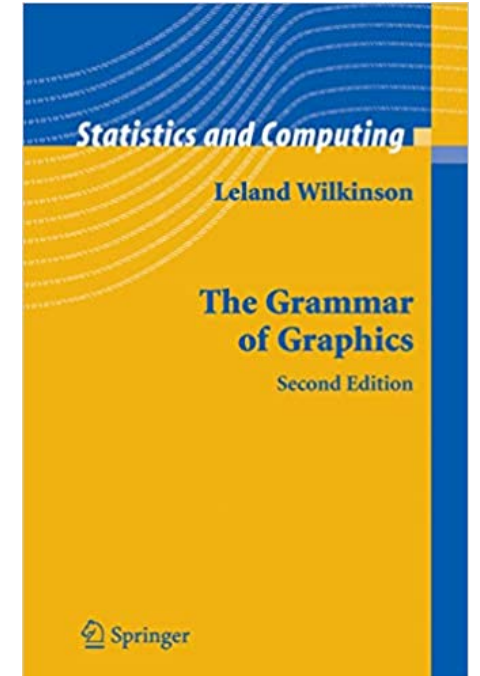
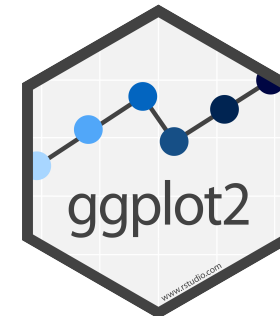
# はじめに：`geom`の読み方

- ゲオム？ジオム（チオム）？
  - 単なる宗教論争に過ぎない...
    - → **お好み**でどちらでも。
  - 類例
    - CRAN, RStudio, tidyverse
    - Jupyter Notebook, Kubernetes
- ここでは「ジオム」とする。
  - `geometry`（幾何）



# 預言の書：`The Grammar of Graphics`\*1

- 告解：ちゃんと読んだことはありません...
- ggplot2の哲学的土台
  - グラフとは何か？
  - グラフ作成の基本的ルール
  - →Hadley Wickhamがコードで実装。
    - `A **layered** grammar of graphics`\*2



\*1 <https://www.amazon.co.jp/Grammar-Graphics-Statistics-Computing/dp/0387245448/>

\*2 <https://vita.had.co.nz/papers/layered-grammar.html>

# グラフの内部構造としてのレイヤー

- 'A **layered** grammar of graphics'
- 参考
  - 'Making the complex simple in data viz'\*
    - T. Vasilikioti, PyCon DE & PyData Berlin 2019
- ggplot2の原理
  - 表現の層（レイヤー）を重ねてグラフを描く。
  - 層ごとに異なる役割



\* <https://www.youtube.com/watch?v=pwzsGHjTDa4>

# ggplot2入門

ggplot2 101



# 入力データの形

- 'tidy'なデータセット
  - ここでは説明を省略。
  - `plot()`に最適な形とは異なることがある。

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	128042583

variables

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	128042583

observations

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	128042583

values

\* <https://r4ds.had.co.nz/tidy-data.html#fig:tidy-structure>

# ggplot2の基本的用例（構文）\*

- (1) 最初に、`ggplot()`を呼ぶ。+でつないでいく。
  - 元になるデータセットを指定し、「見映えの要素」となる変数を`aes()`に指定。
- それに加え、(2) グラフの種類 `geom_...()`
  - 例：散布図、棒グラフ、折れ線グラフ
- 必要に応じて以下も。
  - (3) scale関数：カラーパレット、凡例、軸
  - (4) ファセット：グループごとに描き分け
  - (5) 座標系：座標反転
- 画像として保存：`ggsave()`

例：x軸がログスケールの散布図

```
ggplot(data, aes(...)) +  
  geom_point() +  
  scale_colour_brewer(...) +  
  scale_x_log10()
```

\* <https://ggplot2.tidyverse.org/>

# aes() (エステティックマッピング)

- 見栄えに関わる「**変数**」を指定

- x軸、y軸の値
- グループごとの塗り分け\*1 : colour, fill
- 点のサイズ (バブルチャート) : size
- 色の濃淡度合い : alpha

例：世界の都市の緯度と気温  
aes(x = latitude,  
y = temperature,  
colour = region,  
size = population)

- ポイント

- ggplot()に指定するのがよいが、geom関数でもよい。
  - 宇宙本\*2 p.131参照 (データやマッピングの継承)
- 引数の値が定数のときは、aes()に入れない。

\*1 塗る色を指定する訳ではない。→ scale関数

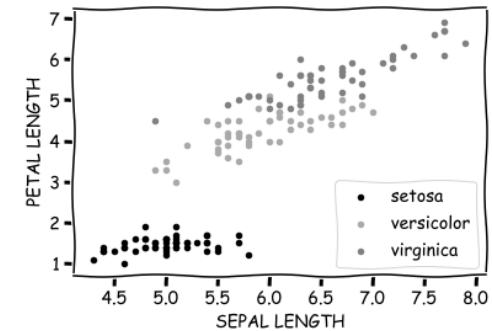
\*2 『RユーザのためのRStudio入門 tidyverseによるモダンな分析フローの世界』(松村、湯谷、紀ノ定、前田、2018年)

# 例：アヤメのデータセット

- ボス：大至急、萼と花弁の長さの関係を種ごとに示せ！



- 私：（そうだ、**散布図**を描こう！）
  - x軸：萼の長さ
  - y軸：花弁の長さ
  - 種に応じて各点を色付けする。



\* <https://xkcd.com/2207/>

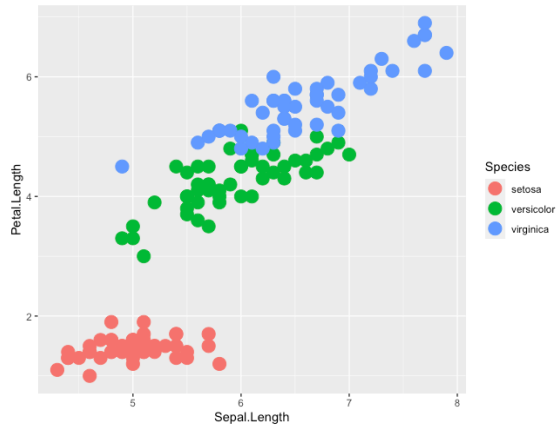
\* Matplotlib's xkcd style: [https://matplotlib.org/3.1.1/api/\\_as\\_gen/matplotlib.pyplot.xkcd.html](https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.xkcd.html)



# コーディング例

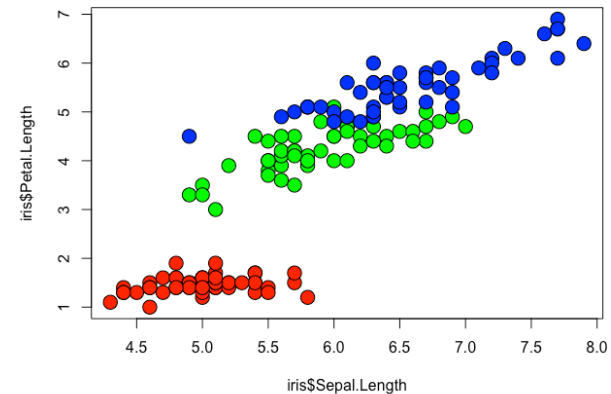
## ggplot2

```
library(ggplot2)
ggplot(iris,
      aes(x = Sepal.Length,
          y = Petal.Length,
          colour = Species)) +
  geom_point(size = 5)
```



## plot()

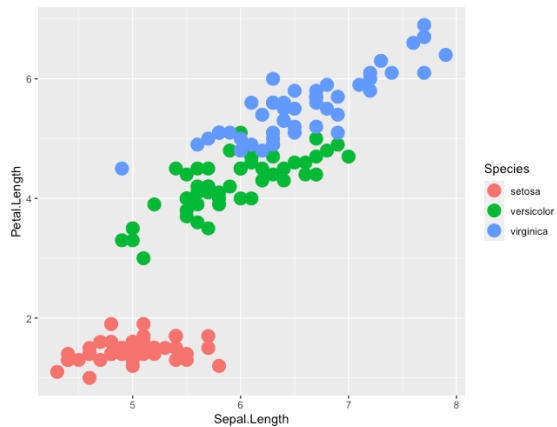
```
plot(x = iris$Sepal.Length,
     y = iris$Petal.Length,
     type = "p", pch = 21, cex = 2,
     bg = c("red", "green", "blue")[unclass(iris$Species)])
```



# 引数をaes()に入れるべきか否か

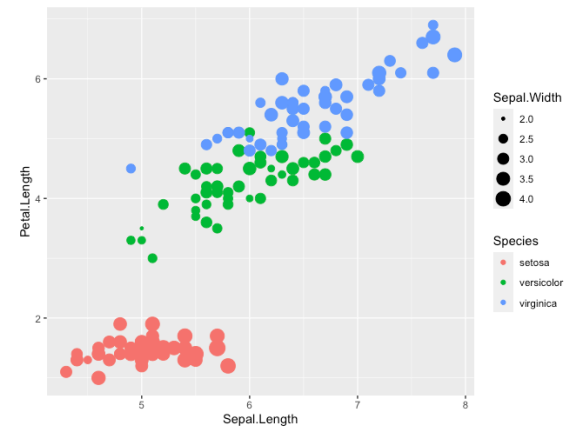
(再掲) 点のサイズが定数 (=5)

```
library(ggplot2)
ggplot(iris,
  aes(x = Sepal.Length,
      y = Petal.Length,
      colour = Species)) +
  geom_point(size = 5)
```



点のサイズが変数 (=Sepal.Width)

```
library(ggplot2)
ggplot(iris,
  aes(x = Sepal.Length,
      y = Petal.Length,
      colour = Species)) +
  geom_point(aes(size = Sepal.Width))
```

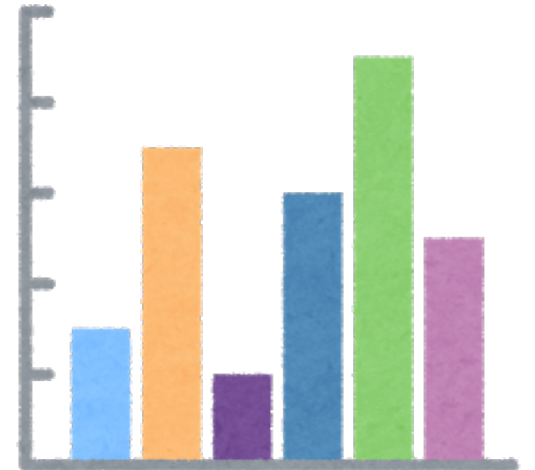
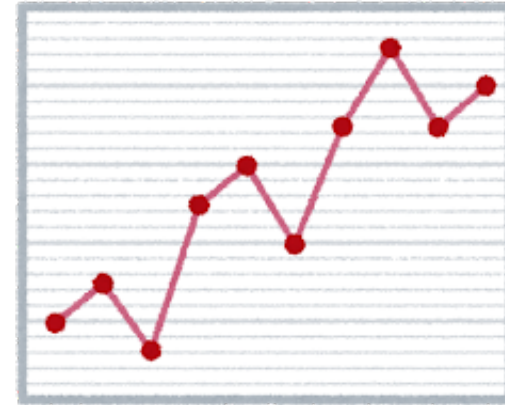


# geom関数

geom functions

# グラフの種類とgeom関数（主なもの）

- 散布図
  - `geom_point()`
- 棒グラフ
  - `geom_col()`, `geom_bar()`
- 折れ線グラフ
  - `geom_line()`, `geom_path()`
- ヒートマップ
  - `geom_tile()`



# コーディング例（のための準備）

- `airquality`
  - 1973年のアメリカ・ニューヨーク市の大気品質データ
    - 詳細は?airquality
  - 日付カラムを持つデータセットを作る。
    - `aq_dat`

```
library(tidyverse)
aq_dat <- airquality %>%
  mutate(date = lubridate::ymd(paste(1973, Month, Day,
                                     sep = "-"))))
```

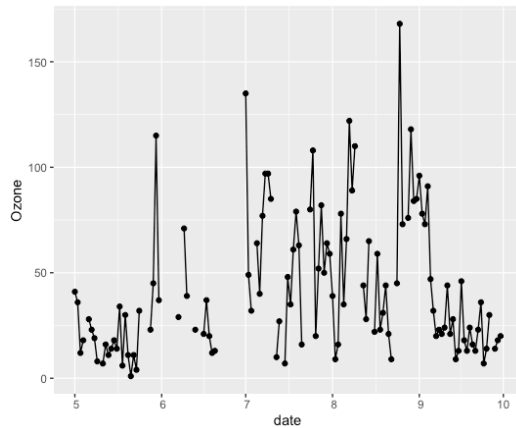
```
> airquality
  Ozone Solar.R Wind Temp Month Day
1    41    190  7.4  67     5    1
2    36    118  8.0  72     5    2
3    12    149 12.6  74     5    3
4    18    313 11.5  62     5    4
5    NA     NA 14.3  56     5    5
```

→オゾン濃度の推移を折れ線で描き、分かりやすいように点を付ける。

# コーディング例：オゾン濃度の推移

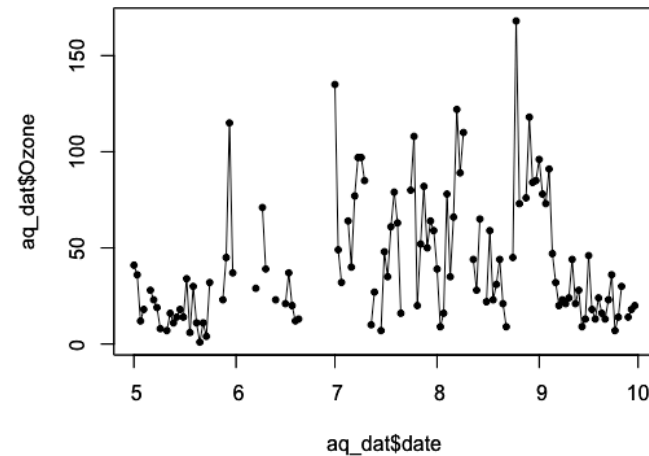
## ggplot2

```
library(ggplot2)
ggplot(aq_dat,
       aes(x = date,
           y = Ozone)) +
  geom_line() +
  geom_point()
```



## plot()\*

```
plot(x = aq_dat$date, y = aq_dat$Ozone,
     type = "l")
par(new = TRUE)
plot(x = aq_dat$date, y = aq_dat$Ozone,
     type = "p", pch = 20)
```



\* type = "o"を使えば、重ね打ちする必要はない。ここでは、重ね打ちを比較するために、敢えて使わなかった。

# scale関数

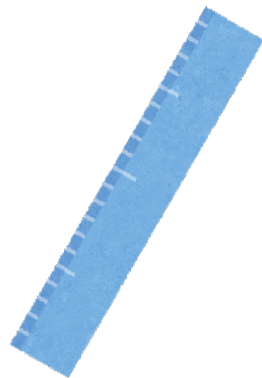
scale functions

# scale関数でできること

- カラーパレット\*と凡例
  - aes()で指定したfill, colour, sizeなどと連動
  - 塗る色と凡例の見映えの操作
    - 凡例についてはtheme関数も有用。



- 軸
  - 対数
  - 反転
  - 目盛



\* カラーパレット：色のセットのこと。カラーマップともいう。宇宙本p.156参照



# scale関数の種類

- 関数名のテンプレート: `scale_xxx_yyy()`
  - `scale_xxx_continuous()` → 連続値
  - `scale_xxx_discrete()` → 離散値
  - `scale_xxx_brewer()` → 色の設定
- 軸関係も同様。
  - `scale_x_log10()`
- 参考
  - ggplot2公式サイトの一覧\*1
  - Cookbook for Rの説明 (右表) \*2

<i>xxx</i>	<i>Description</i>
colour	Color of lines and points
fill	Color of area fills (e.g. bar graph)
linetype	Solid/dashed/dotted lines
shape	Shape of points
size	Size of points
alpha	Opacity/transparency

<i>yyy</i>	<i>Description</i>
hue	Equally-spaced colors from the color wheel
manual	Manually-specified values (e.g., colors, point shapes, line types)
gradient	Color gradient
grey	Shades of grey
discrete	Discrete values (e.g., colors, point shapes, line types, point sizes)
continuous	Continuous values (e.g., alpha, colors, point sizes)

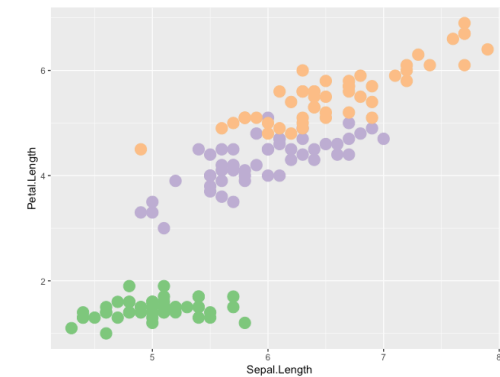
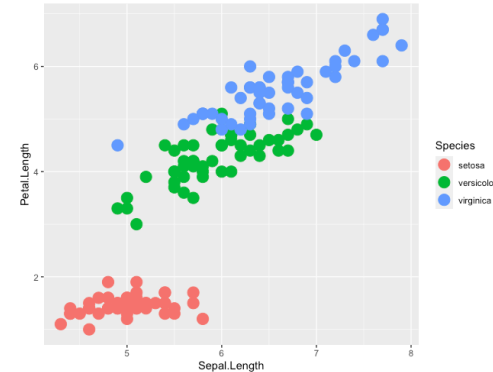
\*1 <https://ggplot2.tidyverse.org/reference/#section-scales>

\*2 [http://www.cookbook-r.com/Graphs/Legends\\_\(ggplot2\)/#kinds-of-scales](http://www.cookbook-r.com/Graphs/Legends_(ggplot2)/#kinds-of-scales)

# コーディング例

アヤメの散布図（前出） → カラーパレットをAccentに変更\*、凡例を消す。

```
library(ggplot2)
ggplot(iris,
       aes(x = Sepal.Length,
           y = Petal.Length,
           colour = Species)) +
  geom_point(size = 5) +
  scale_colour_brewer(palette = "Accent",
                     guide = FALSE)
```



\* RColorBrewerのインストールが必要。宇宙本p.156参照

# まとめ

Long story short

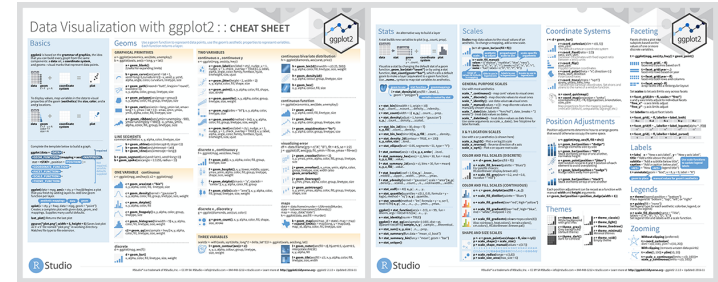
# 要点

- ggplot2でのグラフ描画
  - グラフの要素をそれぞれ指定していく。→レイヤー
- 必須：
  - ggplot() : 全レイヤーに関わる要素を指定
  - aes() : 見映えの要素となる変数
  - geom関数 : グラフの種類
- オプション：
  - scale関数 : カラーパレット、凡例、軸の調整

# ネットで公開されている便利なリソース

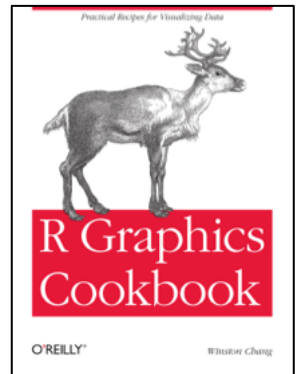
- 公式チートシート

- <https://github.com/rstudio/cheatsheets/blob/master/data-visualization-2.1.pdf>



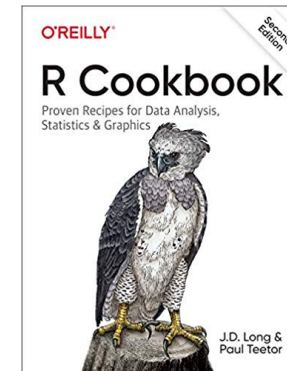
- Cookbook for R

- R Graphics Cookbookの著者によるwebサイト
- <http://www.cookbook-r.com/Graphs/>



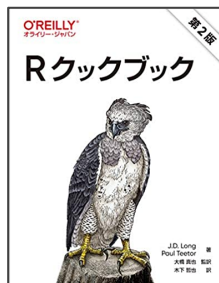
- R Cookbook

- R Cookbookの著者によるwebサイト
- <https://rc2e.com/graphics>



# 参考書（宇宙本は言うまでもなく）

- 『Rグラフィックスクックブック』第2版（Chang、2019年）
- 『Rクックブック』第2版（Long, Teetor、2020年）
- 『Rではじめるデータサイエンス』（Wickham, Grolemund、2017年）
- 『データ可視化のデザイン』（永田、2020年）



**Enjoy!**