

初心者セッション：データ可視化

BeginneR Session: Data Visualisation 101

21st June 2025, Tokyo.R #118
Yuta Kanzawa @yutakanzawa



Senior Data Scientist at Zurich Insurance Company Limited, Japan Branch



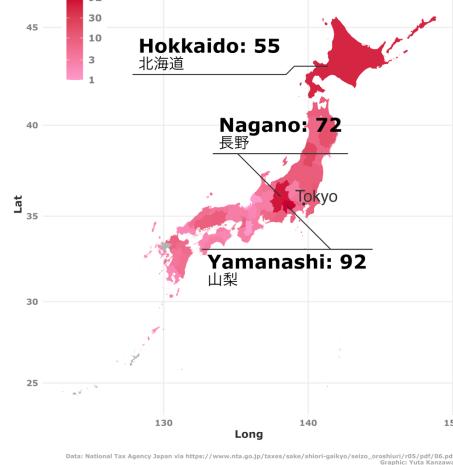
神沢雄大 Yuta Kanzawa

- データサイエンティスト@チューリッヒ保険会社
 - 日本支店
- Twitter: [@yutakanzawa](https://twitter.com/yutakanzawa)
- 好きなもの：オペラとワイン
 - ワーグナー
 - ブルゴーニュ (WSET Lv 3→?)
- 使用可能言語：7
 - 人間：日本語、英語、ドイツ語
 - コンピューター：R, Python, SAS, SQL



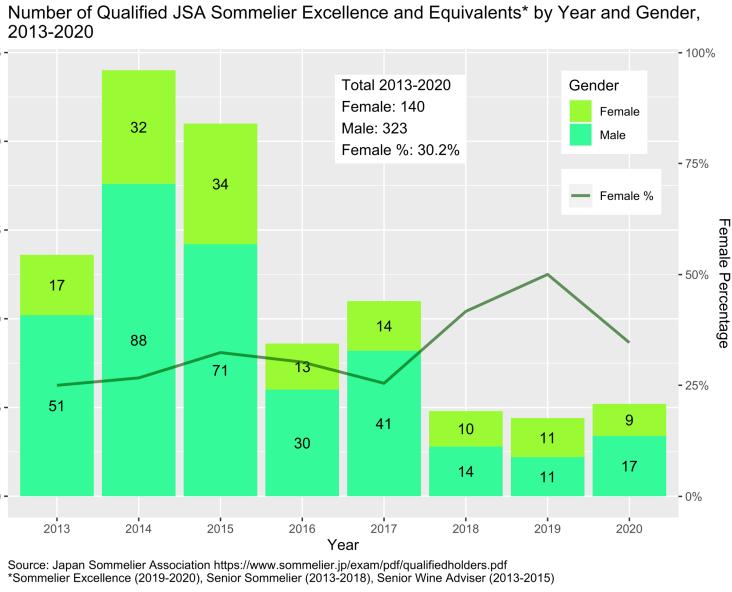
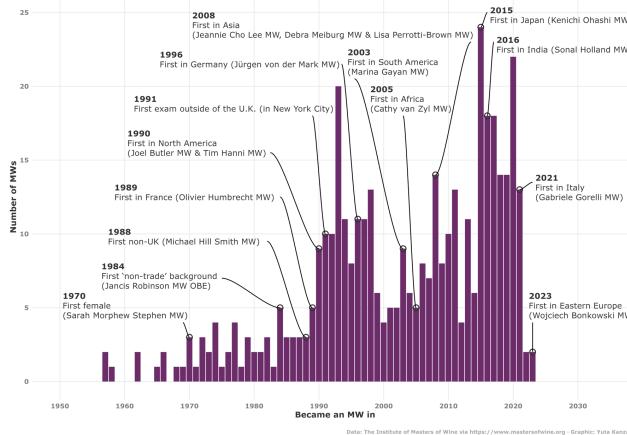
ポートフォリオ

Number of Wineries in Japan in 2022, by Prefecture

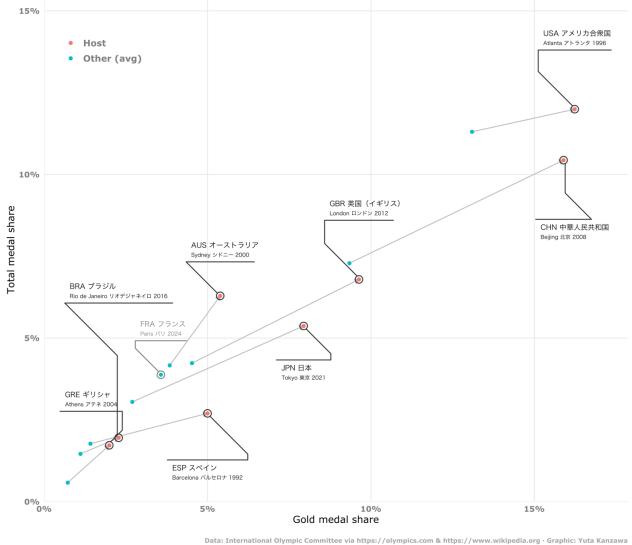


415 Active Masters of Wine by Year of Qualification

As of May 2023, 500 people have gained the title since the inaugural exam in May 1953. NB: 85 deceased or resigned MWs are not counted here.



Medal shares of Olympic host countries in the past 30 years



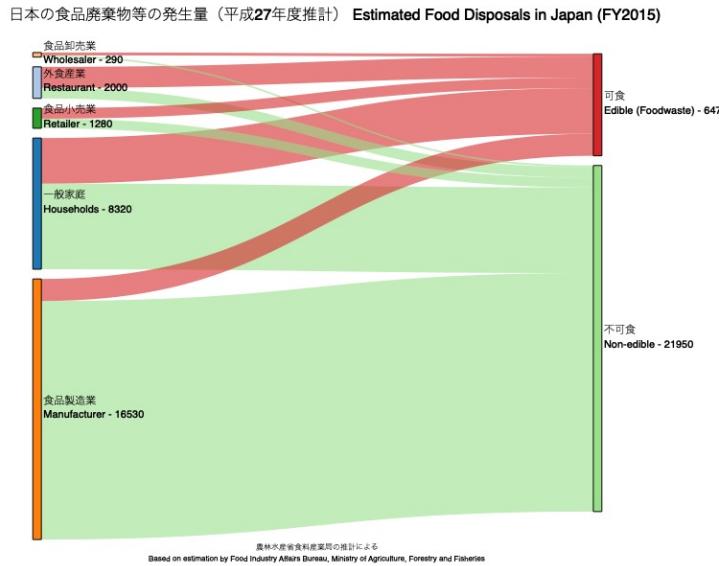
Number of Active MWs by Region Based in
70% of active MWs are based in Europe (mostly Western Europe). NB: Some MWs are multi-based.



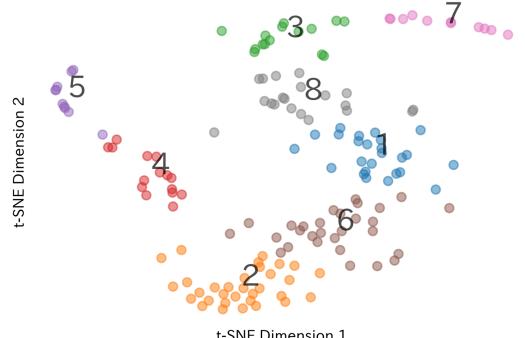
Daily maximum temperature in Tokyo, 1875-2021



ポートフォリオ（参考までにR以外も）

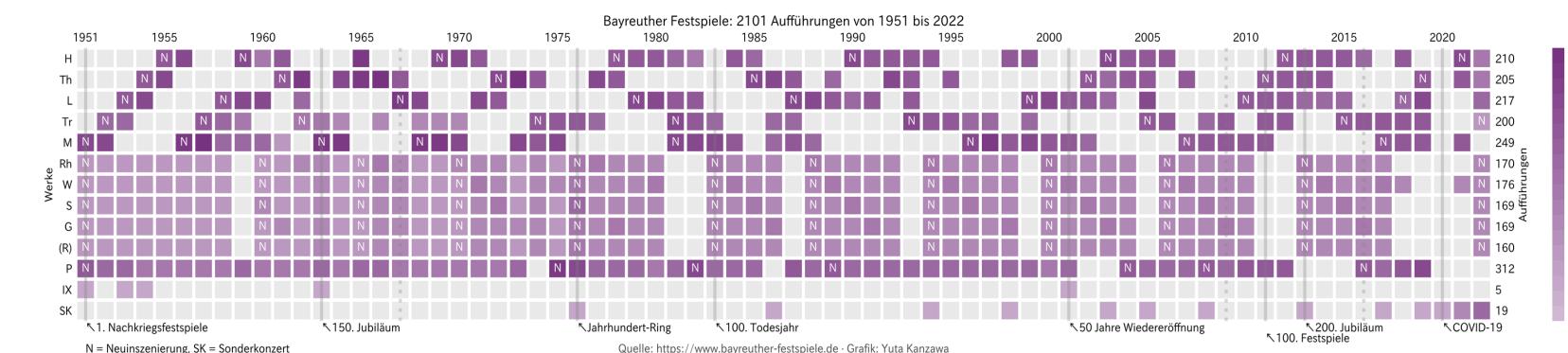
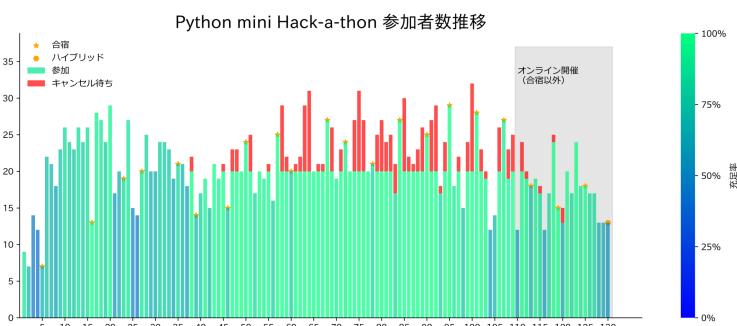


Clustering of Countries and Regions
by Wine Trade Values & Production/Consumption Volumes in 2017 using t-SNE and K-Means



1: Albania, Algeria, Andorra, Aruba, Bahamas, Barbados, Bermuda, Bosnia Herzegovina, Cameron, Costa Rica, Cyprus, Côte d'Ivoire, Ecuador, Guatemala, Iceland, India, Indonesia, Jamaica, Kenya, Malta, Mauritius, Morocco, Panama, Qatar, Viet Nam
2: Argentina, Australia, Austria, Azerbaijan, Belarus, Belgium, Bulgaria, Chile, Costa Rica, Cyprus, Egypt, Ethiopia, Georgia, Greece, Hungary, Ireland, Italy, Jordan, Lebanon, Madagascar, Malawi, Mali, Mauritania, Myanmar, Nepal, Niger, Pakistan, Pelosi, Saint Kitts and Nevis, Saint Vincent and the Grenadines, Samoa, Sierra Leone, Solomon Isds, State of Palestine, Sunnane, Timor-Leste, Tunisia, Uzbekistan
3: Austria, Brazil, Czechia, Denmark, Finland, Ireland, Lithuania, Mexico, Norway, Poland, Rep. of Korea, Singapore, Sweden, Taiwan, United Arab Emirates
4: Armenia, Azerbaijan, Bulgaria, Chile, Costa Rica, Cyprus, Egypt, Ethiopia, Georgia, Hungary, Ireland, Italy, Jordan, Lebanon, Madagascar, Malawi, Mali, Mauritania, Myanmar, Nepal, Niger, Pakistan, Pelosi, Saint Kitts and Nevis, Saint Vincent and the Grenadines, Samoa, Sierra Leone, Solomon Isds, State of Palestine, Sunnane, Timor-Leste, Tunisia, Uzbekistan
5: Austria, Brazil, Czechia, Denmark, Finland, Ireland, Lithuania, Mexico, Norway, Poland, Rep. of Korea, Singapore, Sweden, Taiwan, United Arab Emirates
6: Antigua and Barbuda, Bahrain, Belize, Bolivia, Cabo Verde, Congo, El Salvador, Eswatini, Fiji, Greenland, Honduras, Kyrgyzstan, Lao People's Dem. Rep., Maldives, Mongolia, Nicaragua, Nigeria, Oman, Rwanda, Saint Lucia, Sao Tome and Principe, Senegal, Seychelles, Sri Lanka, Uganda, United Rep. of Tanzania, Zambia, Zimbabwe
7: Argentina, Australia, Austria, Azerbaijan, Belarus, Belgium, Bulgaria, Chile, Costa Rica, Cyprus, Egypt, Ethiopia, Georgia, Hungary, Ireland, Italy, Jordan, Lebanon, Madagascar, Malawi, Mali, Mauritania, Myanmar, Nepal, Niger, Pakistan, Pelosi, Saint Kitts and Nevis, Saint Vincent and the Grenadines, Samoa, Sierra Leone, Solomon Isds, State of Palestine, Sunnane, Timor-Leste, Tunisia, Uzbekistan
8: Angola, Belarus, Colombia, Croatia, Dominica Rep., Estonia, Israel, Kazakhstan, Latvia, Luxembourg, Malaysia, Namibia, Paraguay, Peru, Philippines, Romania, Serbia, Slovakia, Thailand, Ukraine

Sources: UN Comtrade (<https://comtrade.un.org>), FAOSTAT (<http://www.fao.org/faostat/>)



アジェンダ

- 今日話すこと
 - ggplot2
 - plot()との比較 (少しだけ)
 - カラーユニバーサルデザイン
 - 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
 - 今日話ないこと
 - カッコいい絵の描き方
- ggplot2の構造的理

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
 - カラーユニバーサルデザイン : 人によって色の見え方が違う！

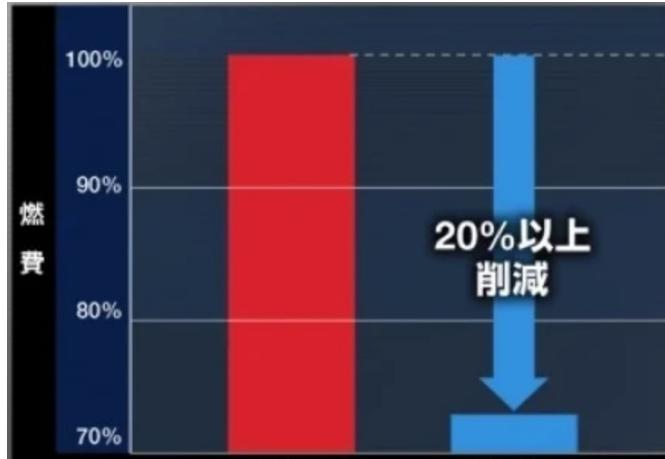
* 日本語では「エステティック」と表されることもある。

データ可視化の心得

Principles of data visualisation

グラフにした方が伝えやすい！伝わりやすい！でも...

- 伝えたい != 伝わる
 - 意味不明（複雑怪奇）
 - 恣意的な表現
 - 客觀性の担保？



- DEI → カラーユニバーサルデザイン
 - Diversity, Equity, and Inclusion
 - 人によって色の見え方が違うことを考慮。



* <https://note.com/kenxxxken/n/nce7762dcec30>
* <https://www.kyusan-u.ac.jp/pdf/led120119.pdf>

ggplot2概論

ggplot2 overview

FYI: 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/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**

A Grammar of Graphics for Python

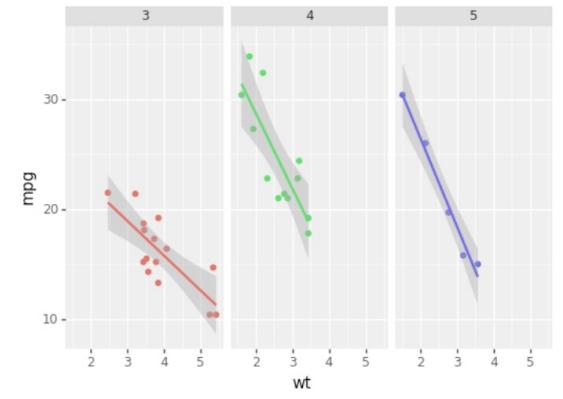
plotnine is an implementation of a *grammar of graphics* in Python, it is based on `ggplot2`. The grammar allows users to compose plots by explicitly mapping data to the visual objects that make up the plot.

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))
```

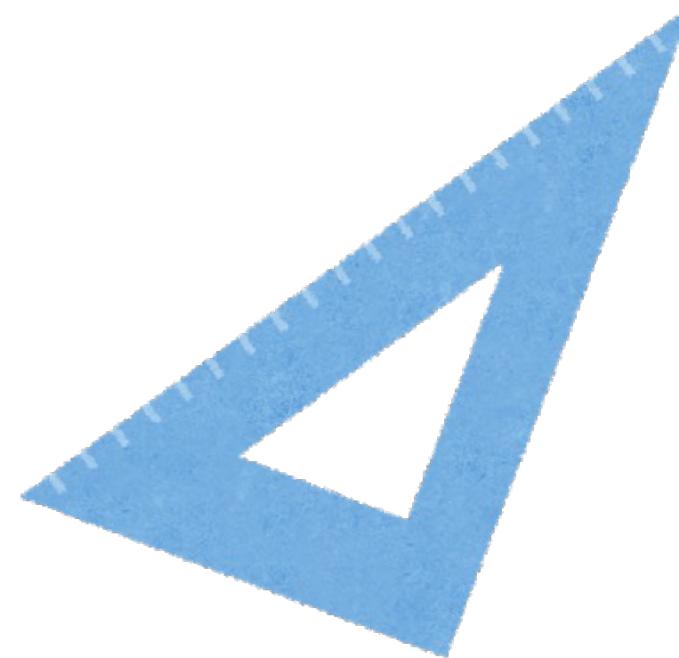


A faceted regression plot showing the relationship between weight (wt) and miles per gallon (mpg) for three different gear types (3, 4, and 5). The plot is faceted by gear type. Each facet contains a scatter plot of points and a linear regression line. The points are colored by gear type: 3 is red, 4 is green, and 5 is blue. The x-axis (wt) ranges from 2 to 5, and the y-axis (mpg) ranges from 10 to 30. The plot shows a general downward trend as weight increases, with steeper slopes for higher gear types.

* 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（ヂオム）？
 - ・ 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）

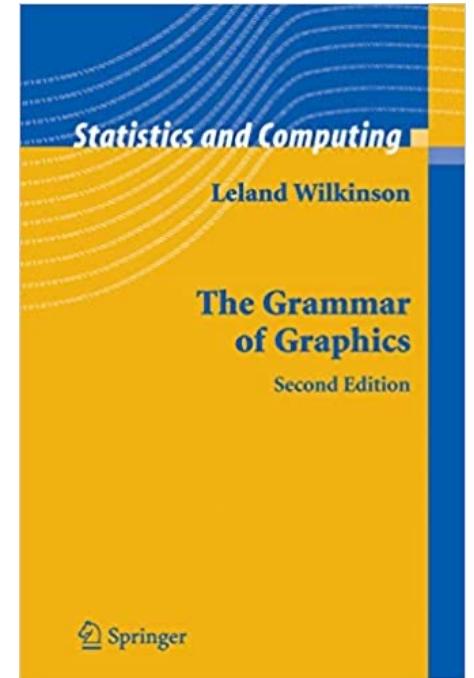
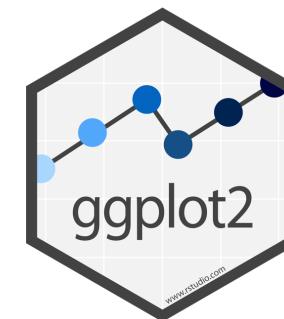


預言の書：'The Grammar of Graphics'*1

- 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

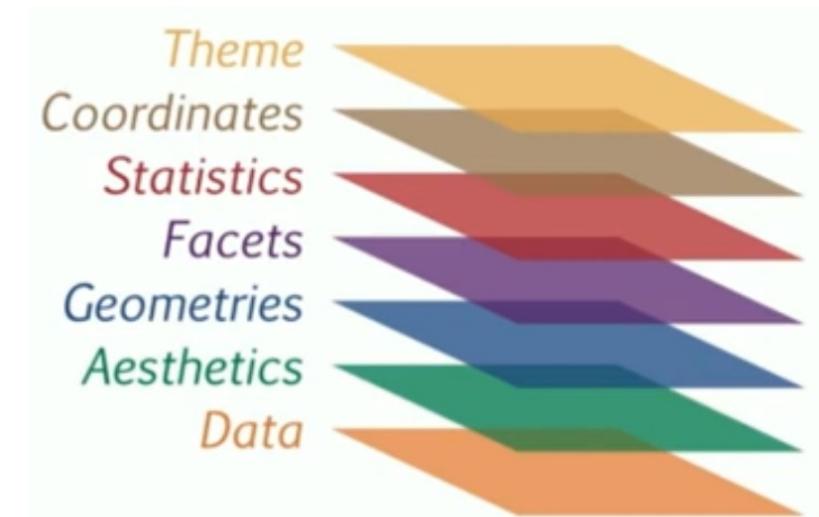


*1 <https://www.amazon.co.jp/Grammar-Graphics-Statistics-Computing/dp/0387245448/>

*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



@yutakanzawa



13

入力データの形

- 'tidy'なデータセット (ggplot2はtidyverseの一部。)
 - 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	1990	745	1987071
Afghanistan	2000	2666	20595360
Brazil	1999	31737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	128042583

variables

country	year	cases	population
Afghanistan	1999	745	1987071
Afghanistan	2000	2666	20595360
Brazil	1999	31737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	128042583

observations

country	year	cases	population
Afghanistan	1999	745	1987071
Afghanistan	2000	2666	20595360
Brazil	1999	31737	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)

- ・見映えに関わる「**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
- ・グループごとの点（マーカー）の形 : shape
- ・点のサイズ（バブルチャート） : size
- ・色の濃淡度合い : alpha

- ・ポイント

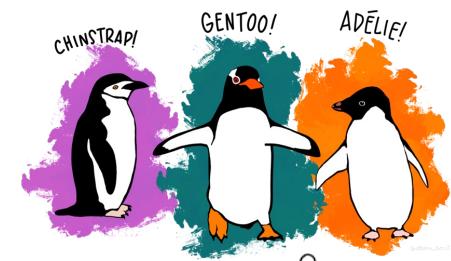
- ・**ggplot()**に指定するのがよいが、**geom**関数でもよい。
 - ・宇宙船本^{*2}p.268参照（エステティックマッピング）
 - ・引数の値が定数のときは、**aes()**に入れない。

例：世界の都市の緯度と気温

```
aes(x = latitude,  
y = temperature,  
colour = region,  
shape = region,  
size = population)
```

*1 塗る色を指定する訳ではない。→ scale関数

*2 『改訂2版 RユーザのためのRstudio[実践]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、2021年）



例：ペンギンのデータセット

- ・ボス：大至急、くちばしとひれ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：くちばしの長さ
- y軸：ひれの長さ
- 種に応じた色、形を使う。
- データセットはdata("penguins")で呼び出す（4.5.0で標準搭載）。
- 4.5未満はpalmerpenguinsパッケージをインストール（注：カラム名が違う。）

	species	island	bill_len	bill_dep	flipper_len	body_mass	sex	year
1	Adelie	Torgersen	39.1	18.7	181	3750	male	2007
2	Adelie	Torgersen	39.5	17.4	186	3800	female	2007
3	Adelie	Torgersen	40.3	18.0	195	3250	female	2007
4	Adelie	Torgersen	NA	NA	NA	NA	<NA>	2007
5	Adelie	Torgersen	36.7	19.3	193	3450	female	2007
6	Adelie	Torgersen	39.3	20.6	190	3650	male	2007

* <https://allisonhorst.github.io/palmerpenguins/>
* <https://xkcd.com/2207/>

コーディング例

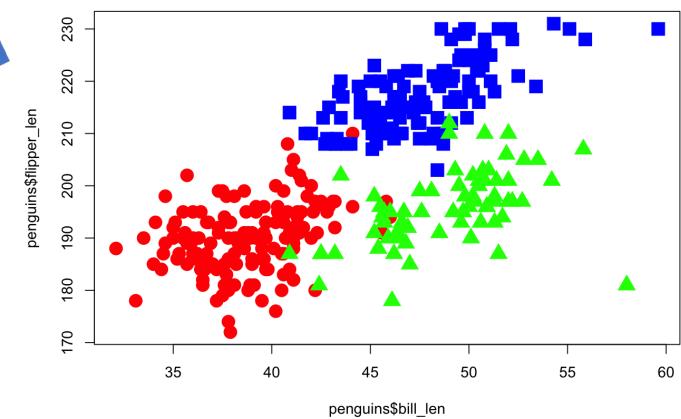
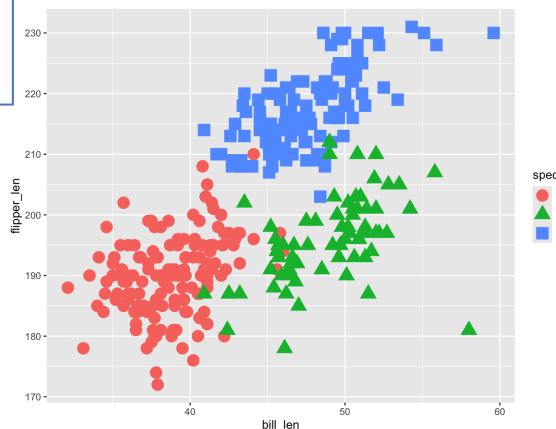
ggplot2

```
library(ggplot2)
data(penguins)
ggplot(penguins,
       aes(x = bill_len,
            y = flipper_len,
            colour = species,
            shape = species)) +
  geom_point(size = 5)
```



plot()

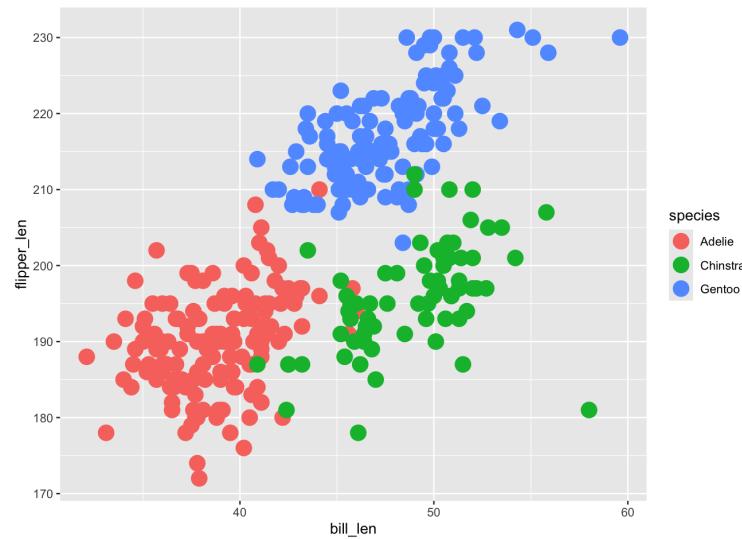
```
data(penguins)
plot(x = penguins$bill_len,
      y = penguins$flipper_len,
      type = "p",
      pch = c(16, 17, 15)[unclass(penguins$species)], cex = 2,
      col = c("red", "green", "blue")[unclass(penguins$species)])
```



引数をaes()に入れるべきか否か

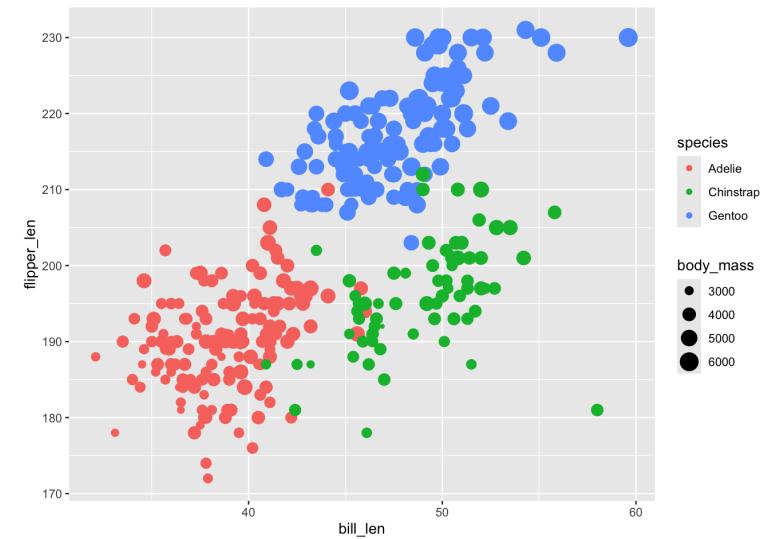
点のサイズが定数 (=5)

```
ggplot(penguins,  
       aes(x = bill_len,  
            y = flipper_len,  
            colour = species)) +  
  geom_point(size = 5)
```



点のサイズが変数 (=body_mass)

```
ggplot(penguins,  
       aes(x = bill_len,  
            y = flipper_len,  
            colour = species)) +  
  geom_point(aes(size = body_mass))
```

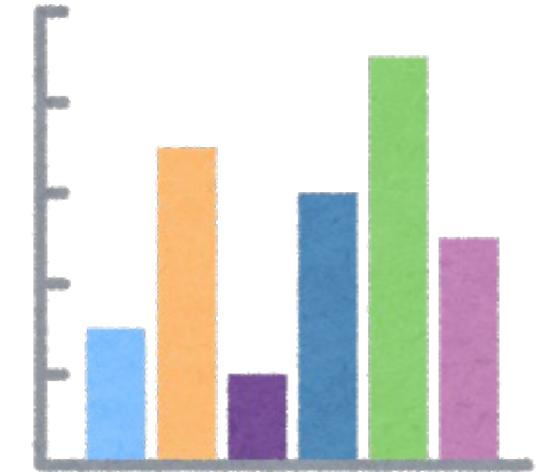
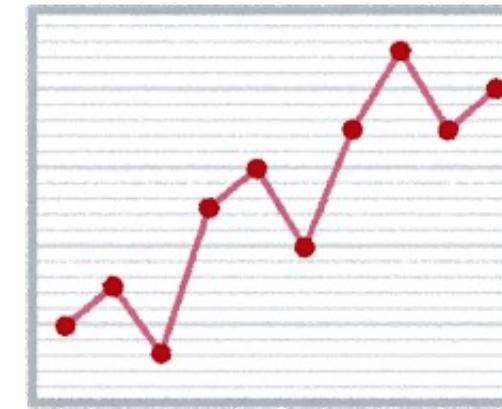


geom 関数

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）

- 引き続きペンギン
 - 年別の平均体重を種ごとに計算。

```
library(tidyverse)
data(penguins)
penguins_mass_avg <- penguins %>% group_by(species, year) %>%
  summarise(body_mass = mean(body_mass, na.rm = TRUE))

# plot()用に種ごとのカラムに変換。
penguins_mass_avg_species <- penguins_mass_avg %>%
  pivot_wider(id_cols = year, names_from = species, values_from = body_mass)
```

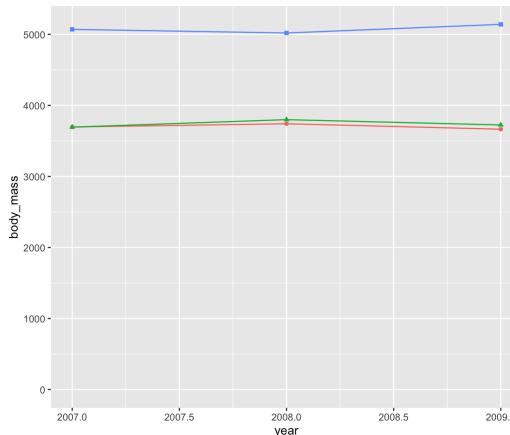
```
> head(penguins_mass_avg)
# A tibble: 6 × 3
# Groups:   species [2]
  species   year   body_mass
  <fct>     <int>     <dbl>
1 Adelie    2007    3696.
2 Adelie    2008    3742
3 Adelie    2009    3665.
4 Chinstrap 2007    3694.
5 Chinstrap 2008    3800
6 Chinstrap 2009    3725
```

→平均体重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：平均体重の推移

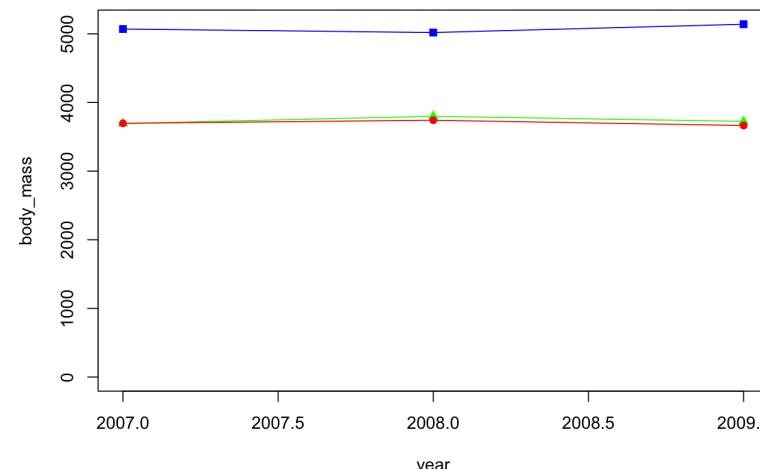
ggplot2

```
library(ggplot2)
ggplot(penguins_mass_avg,
       aes(x = year, y = body_mass,
            colour = species,
            shape = species)) +
  geom_line() +
  geom_point() +
  ylim(0, NA)
```



plot()*

```
plot(x = penguins_mass_avg_species$year, y = penguins_mass_avg_species$Gentoo,
      ylim = c(0, max(penguins_mass_avg_species$Gentoo)),
      type = "o", pch = 15, col = "blue", xlab = "year", ylab = "body_mass")
lines(x = penguins_mass_avg_species$year,
      y = penguins_mass_avg_species$Chinstrap, type = "o", pch = 17, col = "green")
lines(x = penguins_mass_avg_species$year,
      y = penguins_mass_avg_species$Adelie, type = "o", pch = 16, col = "red")
```



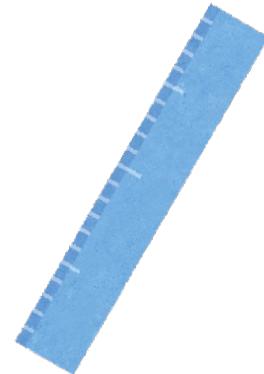
* 別途凡例を付ける必要がある。

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, shape, 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。宇宙船本p.319参照

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>

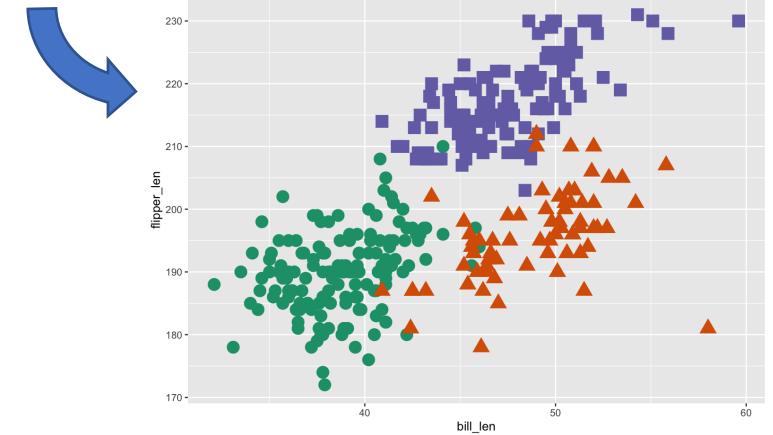
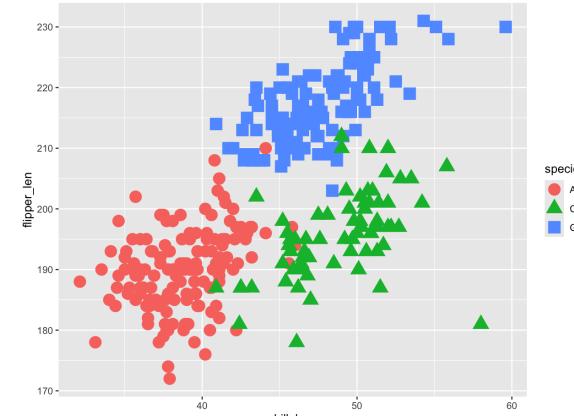
*2 [http://www.cookbook-r.com/Graphs/Legends_\(ggplot2\)/#kinds-of-scales](http://www.cookbook-r.com/Graphs/Legends_(ggplot2)/#kinds-of-scales)

コーディング例

ペンギンの散布図（前出） → カラーパレットをDark2に変更*、凡例を消す。

```
ggplot(penguins,  
       aes(x = bill_len,  
            y = flipper_len,  
            colour = species,  
            shape = species)) +  
  geom_point(size = 5) +  
  scale_colour_brewer(palette = "Dark2",  
                      guide = "none") +  
  scale_shape_discrete(guide = "none")
```

※凡例を消すのは `geom_point(show.legend = FALSE)`
または `theme(legend.position = "none")` としてもよい。



* RColorBrewerのインストールが必要。宇宙船本p.319参照

カラーユニバーサルデザインとは？

What's colour universal design?

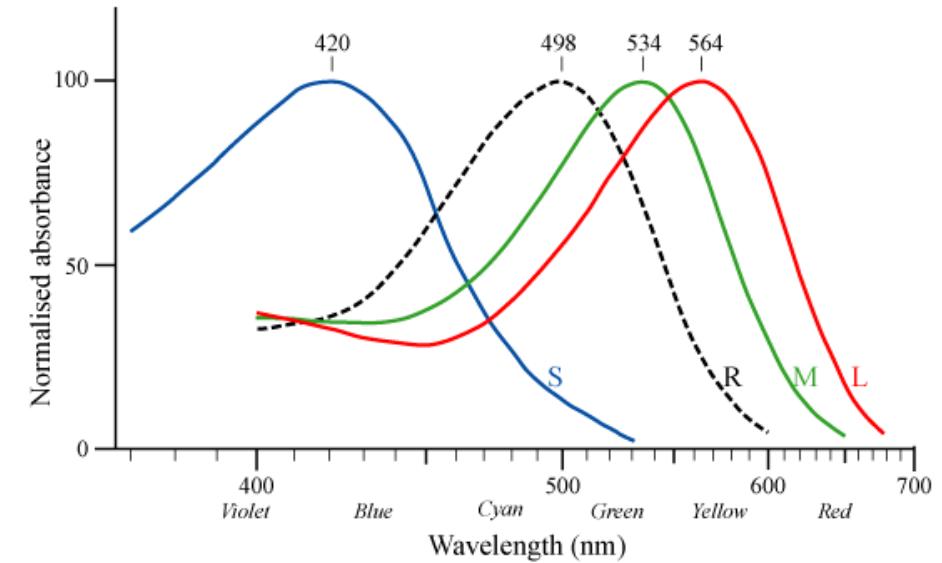
同じに見える信号機（再掲）



* <https://www.kyusan-u.ac.jp/pdf/led120119.pdf>

色覚多様性

- ・医学上の診断名：色覚異常^{*1}
- ・昔の呼ばれ方：「色盲」、「色弱」^{*2}
- ・英語：'colour blindness', 'colour vision deficiency'
- ・人によって「色」の見え方が違う^{*3}。
 - ・赤緑
 - ・青黄



*1 <https://www.gankaikai.or.jp/health/50/index.html>

*2 <https://ja.wikipedia.org/wiki/%E8%89%B2%E8%A6%9A%E7%95%B0%E5%B8%B8>

*3 <https://www.fukushihoken.metro.tokyo.lg.jp/kiban/machizukuri/kanren/color.files/colorudguideline.pdf>

カラーユニバーサルデザイン (CUD)

- ・『多様な色覚に配慮して、情報がなるべくすべての人に正確に伝わるように、利用者の視点に立ってデザインすること』*1
 - ・色使い（カラーパレット）だけの問題ではない。→形や大きさも工夫。
- ・'Set of colors that is **unambiguous** both to colorblinds and non-colorblinds'*2

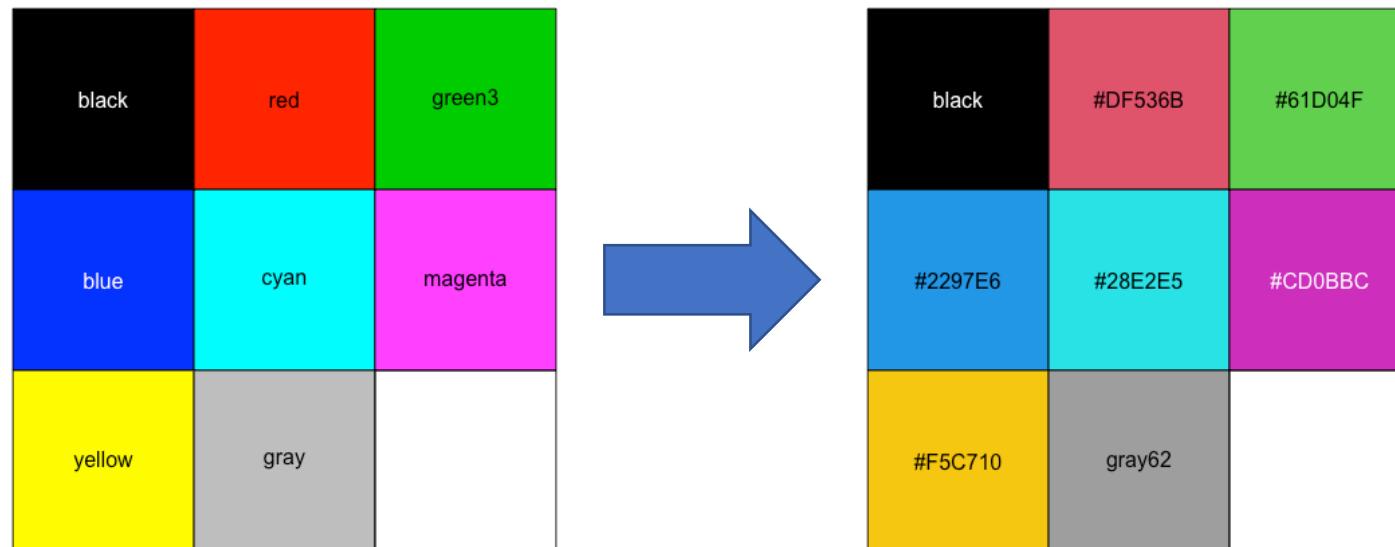
Original	Simulation			Hue	for Photoshop, Illustrator, Freehand, etc.		for Word, Power Point, Canvas, etc.	
	Protan	Deutan	Tritan		C,M,Y,K (%)	R,G,B (0-255)	R,G,B (%)	
1				—°	(0,0,0,100)	(0,0,0)	(0,0,0)	
2				41°	(0,50,100,0)	(230,159,0)	(90,60,0)	
3				202°	(80,0,0,0)	(86,180,233)	(35,70,90)	
4				164°	(97,0,75,0)	(0,158,115)	(0,60,50)	
5				56°	(10,5,90,0)	(240,228,66)	(95,90,25)	
6				202°	(100,50,0,0)	(0,114,178)	(0,45,70)	
7				27°	(0,80,100,0)	(213,94,0)	(80,40,0)	
8				326°	(10,70,0,0)	(204,121,167)	(80,60,70)	

*1 <https://www.fukushi.metro.tokyo.lg.jp/kiban/machizukuri/kanren/color.html>

*2 <https://jfly.uni-koeln.de/color/>

Rも4.0からデフォルト対応

- 'The **palette()** function has a new default set of colours (which are less saturated and have **better accessibility properties**).'



* <https://stat.ethz.ch/pipermail/r-announce/2020/000653.html>

* <https://www.r-bloggers.com/2020/04/4-for-4-0-0-four-useful-new-features-in-r-4-0-0/>

ggplot2で カラーユニバーサルデザイン

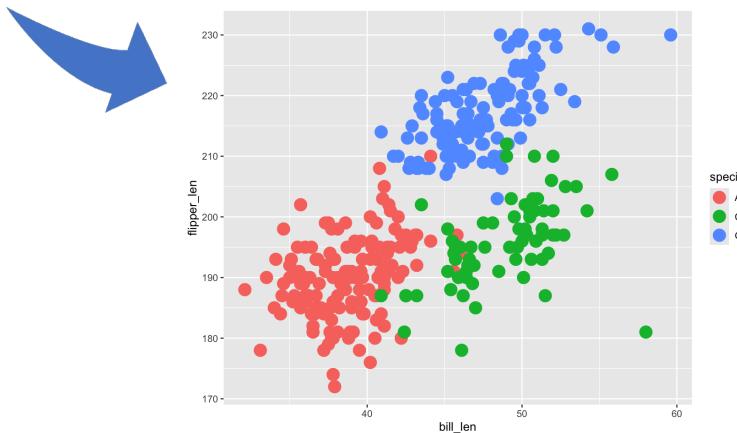
Colour universal design with ggplot2



引数shapeに変数を渡す（基本！）

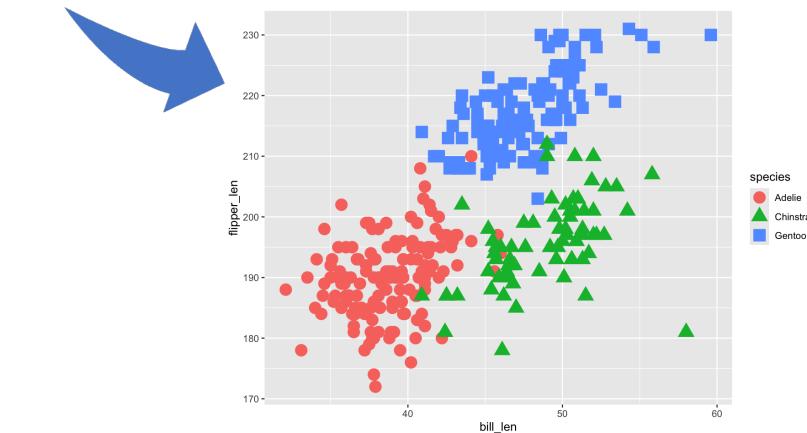
shapeなし（または定数）

```
library(ggplot2)  
data(penguins)  
ggplot(penguins,  
       aes(x = bill_len,  
            y = flipper_len,  
            colour = species)) +  
  geom_point(size = 5)
```



shapeに変数

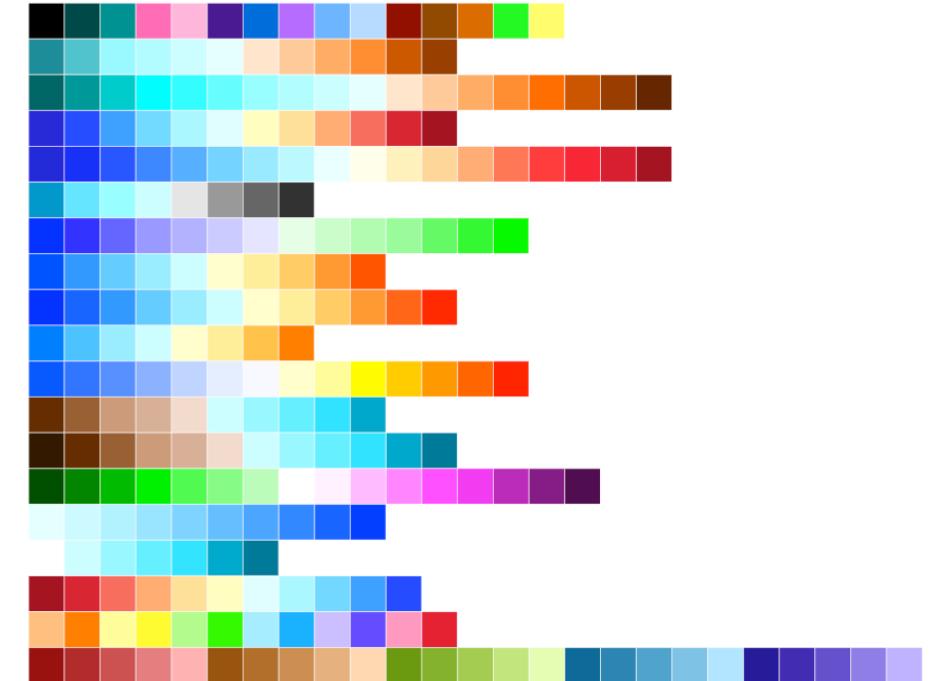
```
library(ggplot2)  
data(penguins)  
ggplot(penguins,  
       aes(x = bill_len,  
            y = flipper_len,  
            colour = species,  
            shape = species)) +  
  geom_point(size = 5)
```



colorBlindnessパッケージ

- ・カラーユニバーサルデザインに沿ったカラーパレット
- ・見え方のシミュレーション機能
- ・色の置換機能

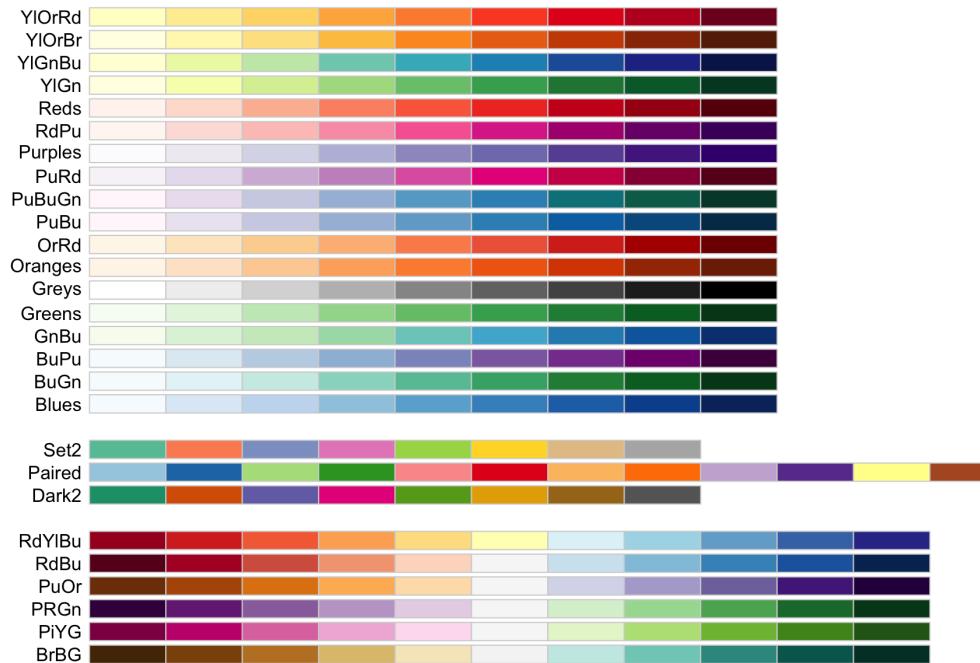
paletteMartin
Blue2DarkOrange12Steps
Blue2DarkOrange18Steps
Blue2DarkRed12Steps
Blue2DarkRed18Steps
Blue2Gray8Steps
Blue2Green14Steps
Blue2Orange10Steps
Blue2Orange12Steps
Blue2Orange8Steps
Blue2OrangeRed14Steps
Brown2Blue10Steps
Brown2Blue12Steps
Green2Magenta16Steps
LightBlue2DarkBlue10Steps
LightBlue2DarkBlue7Steps
ModifiedSpectralScheme11Steps
PairedColor12Steps
SteppedSequential5Steps



* <https://cran.r-project.org/web/packages/colorBlindness/vignettes/colorBlindness.html>

RColorBrewerパッケージ

- ・カラーユニバーサルデザインに沿ったカラーパレットもある。
 - ・`display.brewer.all(colorblindFriendly=TRUE)`



* <https://cran.r-project.org/package=RColorBrewer>

Cookbook for Rで紹介されているカラーパレット

- 前出の' **Set of colors that is unambiguous both to colorblinds and non-colorblinds**' (右図) を ggplot2で使用するためのメモ (コード)
 - グレーと黒の2種類 (左図)

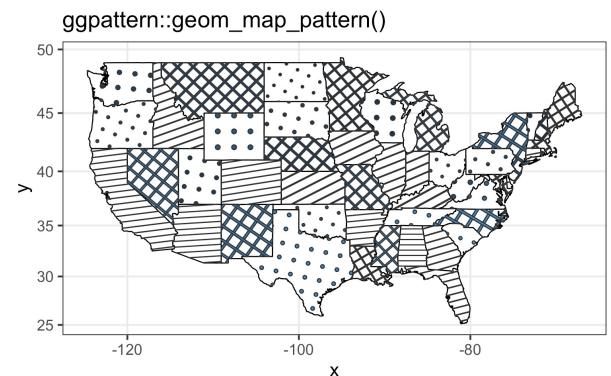
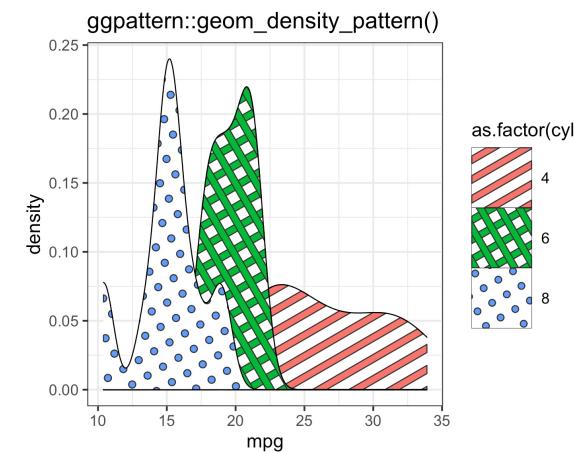
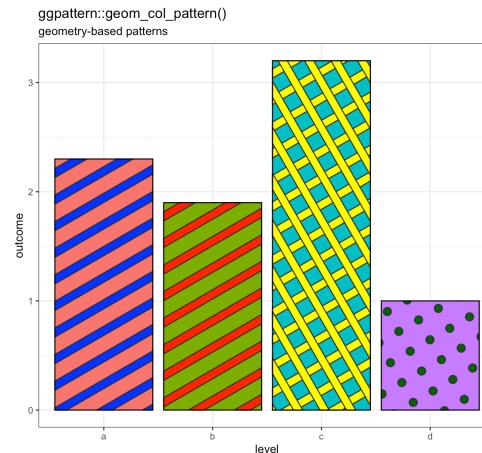
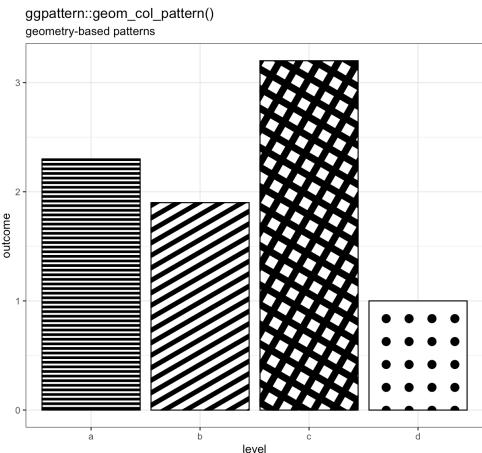


	Original	Simulation			for Photoshop, Illustrator, for Word, Power Point, etc.			
		Protan	Deutan	Tritan	Hue	C,M,Y,K (%)	R,G,B (0-255)	R,G,B (%)
1	Black	Black	Black	Black	-°	(0,0,0,100)	(0,0,0)	(0,0,0)
2	Orange	Orange	Orange	Red	41°	(0,50,100,0)	(230,159,0)	(90,60,0)
3	Sky Blue	Light Blue	Light Blue	Cyan	202°	(80,0,0,0)	(86,180,233)	(35,70,90)
4	bluish Green	Brown	Brown	Teal	164°	(97,0,75,0)	(0,158,115)	(0,60,50)
5	Yellow	Yellow	Yellow	Pink	56°	(10,5,90,0)	(240,228,66)	(95,90,25)
6	Blue	Blue	Blue	Teal	202°	(100,50,0,0)	(0,114,178)	(0,45,70)
7	Vermilion	Brown	Brown	Red	27°	(0,80,100,0)	(213,94,0)	(80,40,0)
8	reddish Purple	Grey	Grey	Red	326°	(10,70,0,0)	(204,121,167)	(80,60,70)

* [http://www.cookbook-r.com/Graphs/Colors_\(ggplot2\)/#a-colorblind-friendly-palette](http://www.cookbook-r.com/Graphs/Colors_(ggplot2)/#a-colorblind-friendly-palette)

ggpatternパッケージ

- ・色が見分けにくいなら、**模様**で！
- ・グラフを**模様**（パターン）で描き分けるgeom群を提供^{*1}。
 - ・多色か白黒かは設定可能。
- ・参考：patternplot^{*2}（plot関数に対応したライブラリー）



*1 <https://coolbutuseless.github.io/package/ggpattern/>

*2 <https://cran.r-project.org/web/packages/patternplot/vignettes/patternplot-intro.html>

まとめ

Long story short

Long story short (1/2)

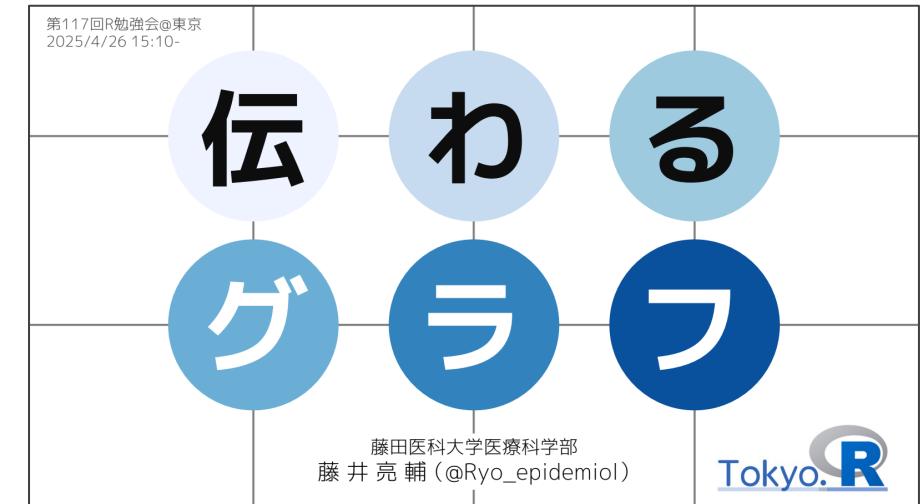
- ggplot2でのグラフ描画
 - グラフの要素をそれぞれ指定していく。→レイヤー
- 必須：
 - ggplot() : 全レイヤーに関わる要素を指定
 - aes() : 見映えの要素となる変数
 - geom関数 : グラフの種類
- オプション：
 - scale関数 : カラーパレット、凡例、軸の調整

Long story short (2/2)

- カラーユニバーサルデザインとは？
 - 人によって色の見え方が違うことを考慮。
 - Rも4.0から対応。
- ggplot2でカラーユニバーサルデザイン（引数shape以外に）
 - カラーパレット
 - colorBlindnessパッケージ
 - RColorBrewerパッケージ
 - Cookbook for Rで紹介されているカラーパレット
 - 模様（パターン）で表現する。
 - ggpatternパッケージ

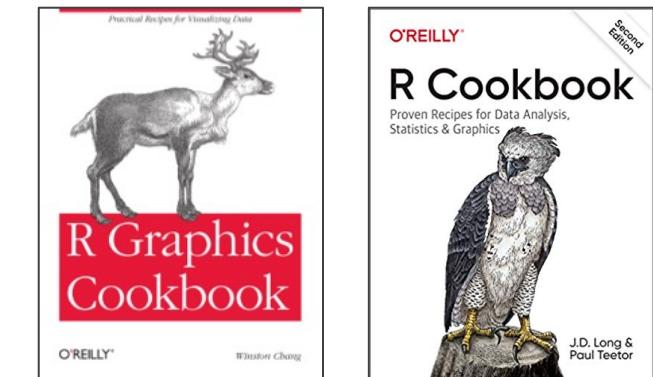
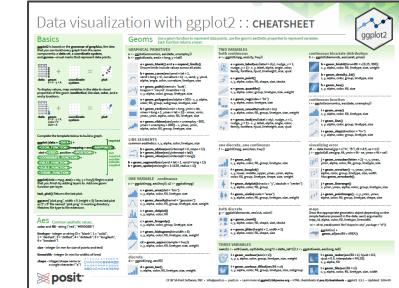
ネットで公開されている便利なリソース (1/2)

- ・『伝わるグラフ』（藤井、2025年）
 - ・第117回Tokyo.R初心者セッション
 - ・https://drive.google.com/file/d/1iHE2Kh_3aFXA6iBjjxVijglaWT1QSABv/view



ネットで公開されている便利なリソース (2/2)

- ggplot2公式チートシート
 - <https://github.com/rstudio/cheatsheets/blob/main/data-visualization.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（宇宙船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, Teator, 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)
- ・『データ分析のためのデータ可視化入門』 (Healy, 2021年)



Enjoy!