ミクロ政治データ分析実習

第11回 可視化（1）

そん じぇひょん
宋 財 泫
関西大学総合情報学部
2021/6/24 (updated: 2021-06-17)

グラフィックの文法と{ggplot2}

グラフを作成する方法

代表的な可視化のパッケージ

Base R

- 別途のパッケージを使わず、R内蔵関数で作図
- 紙にペンでグラフを書くイメージ
- 図が気に入らなかったら一からやり直し
- 作成した図をオブジェクトとして保存することが出来ない
- 最も自由度が高い

{lattice}

- Deepayan Sarkarが開発
- {ggplot2}が登場する前には主流
- 関数1つで可視化ができる（ただし、関数が長くなる）

{ggplot2}

- Hadely Wickhamが大学院生の時に開発
- グラフィックの文法 (grammer of graphics)」の思想をR上で具現化
- グラフの様々な要素をそれぞれ1つの層 (layer)と捉え、積み重ねていく

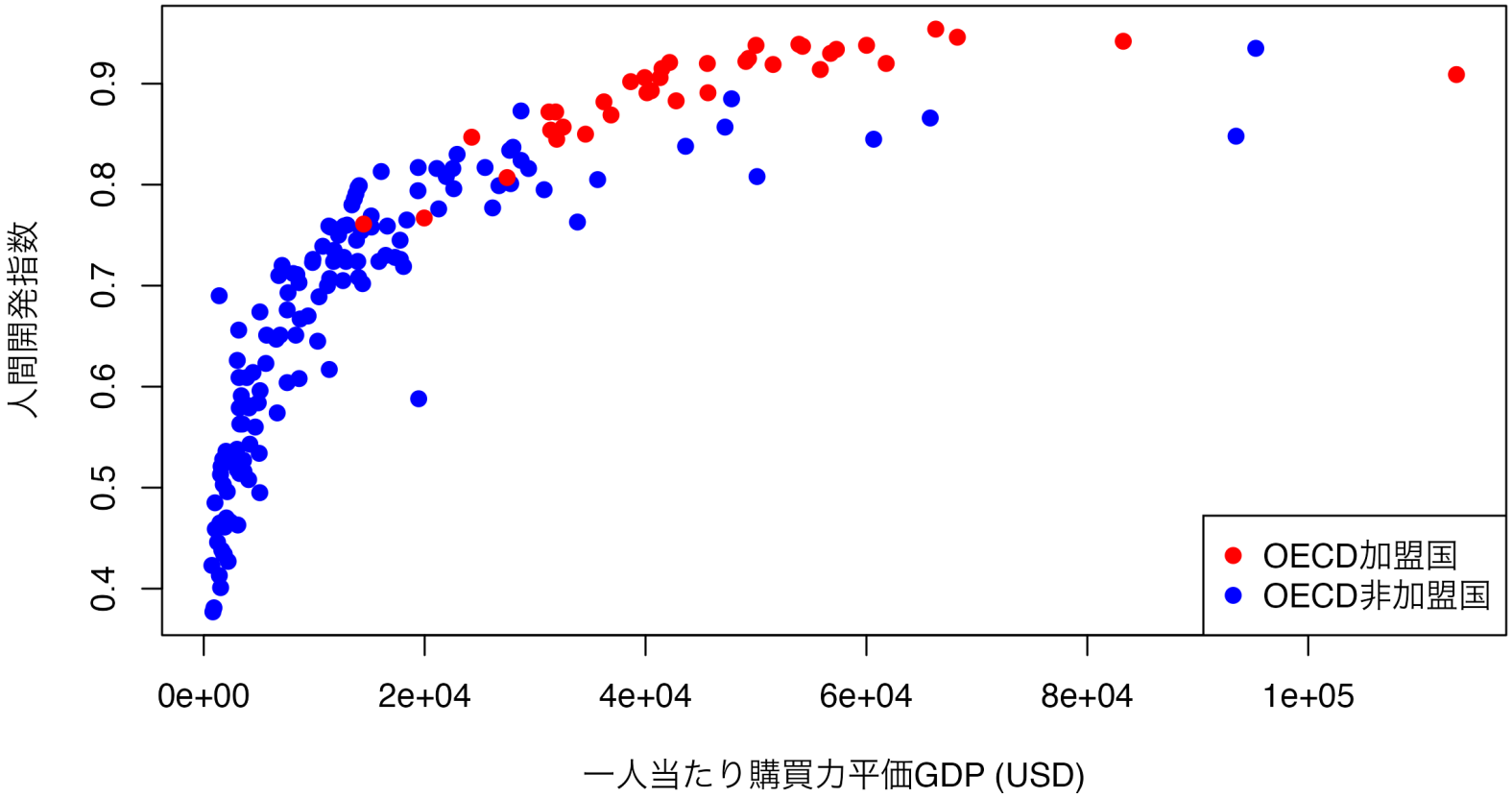
Base Rの例（コード）

- `Micro08.csv`を`df`として使用

```
library(tidyverse)
df <- read_csv("Data/Micro08.csv")
```

```
plot(x = df$PPP_per_capita, y = df$HDI_2018, pch = 19,
     col = ifelse(df$OECD == 1, "red", "blue"),
     xlab = "一人当たり購買力平価GDP (USD)", ylab = "人間開発指数")
legend("bottomright", pch = 19,
       legend = c("OECD加盟国", "OECD非加盟国"),
       col    = c("red", "blue"))
```

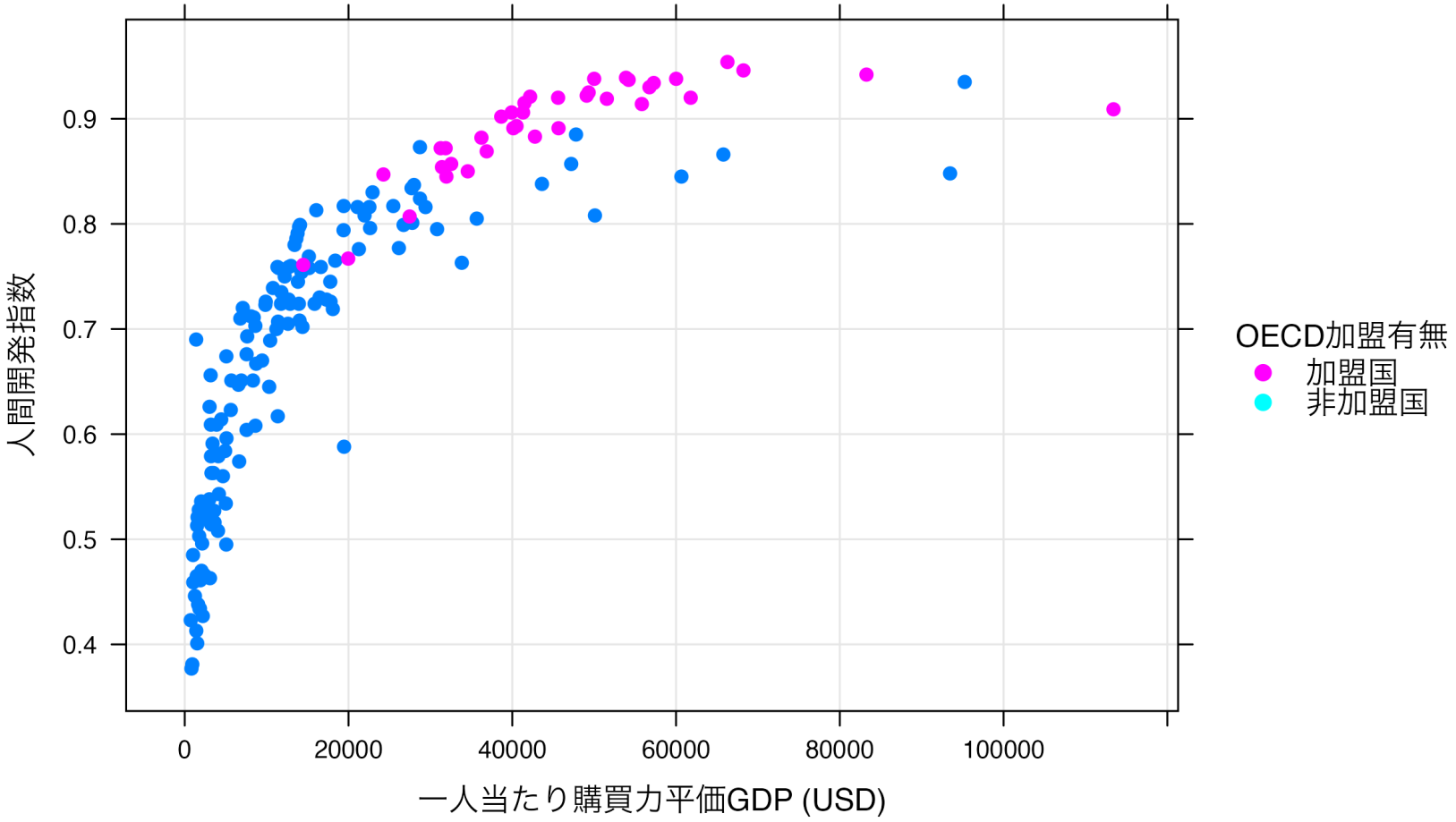
Base Rの例（結果）


人間開発指数
0.4
0.5
0.6
0.7
0.8
0.9
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05
一人当たり購買力平価GDP (USD)
OECD加盟国
OECD非加盟国

{lattice}の例（コード）

```
library(lattice)
xyplot(HDI_2018 ~ PPP_per_capita, data = df,
       group = OECD, pch = 19, grid = TRUE,
       auto.key = TRUE,
       key = list(title     = "OECD加盟有無",
                  cex.title = 1,
                  space     = "right",
                  points    = list(col = c("magenta", "cyan"),
                                   pch = 19),
                  text      = list(c("加盟国", "非加盟国"))),
       xlab = "一人当たり購買力平価GDP (USD)", ylab = "人間開発指数")
```

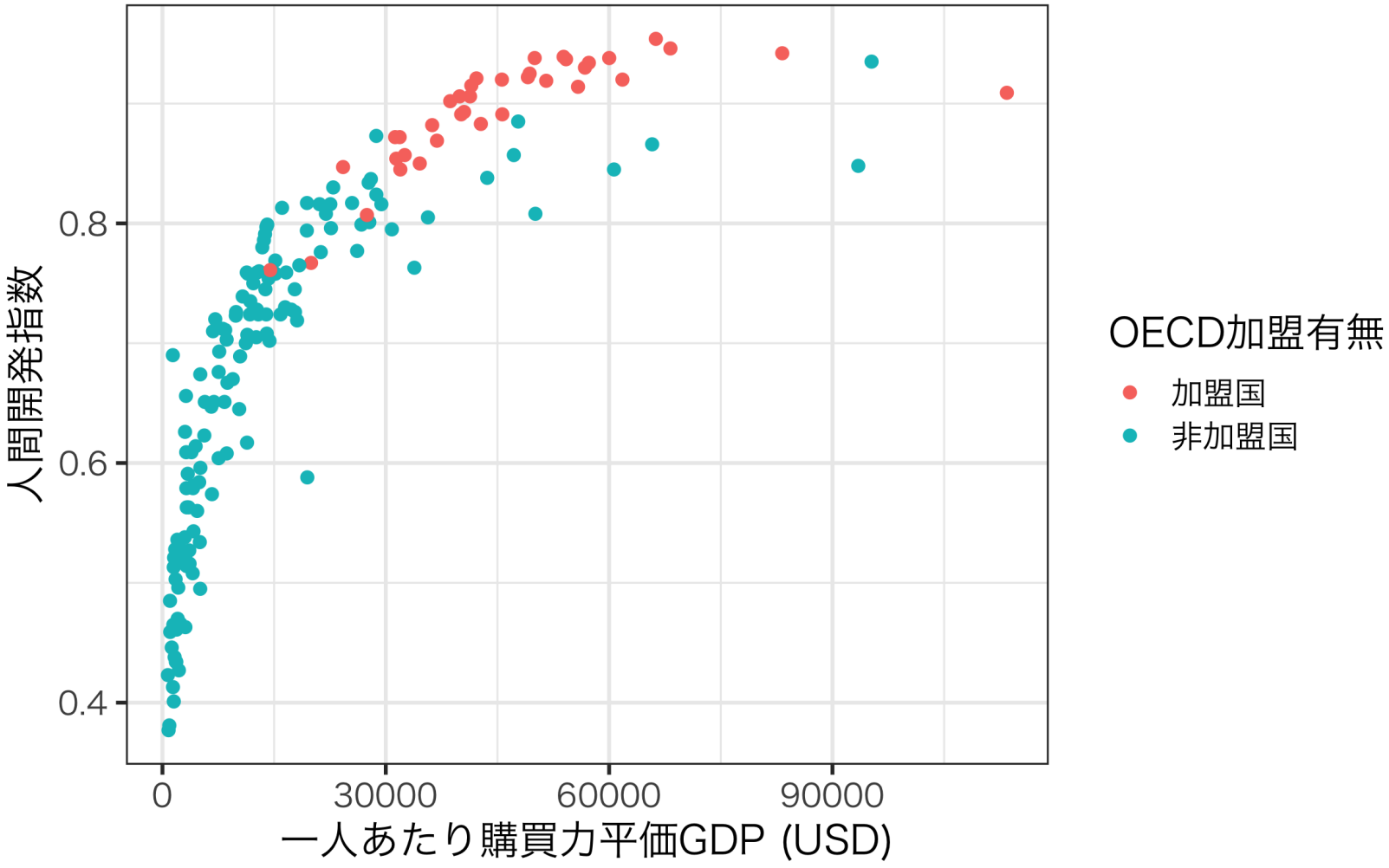
{lattice}の例（結果）



{ggplot2}の例（コード）

```
df %>%
   mutate(OECD = if_else(OECD == 1, "加盟国", "非加盟国")) %>%
   ggplot() +
   geom_point(aes(x = PPP_per_capita, y = HDI_2018, color = OECD),
              size = 2) +
   labs(x = "一人あたり購買力平価GDP（USD)", y = "人間開発指数",
        color = "OECD加盟有無") +
   theme_bw(base_size = 16)
```

{ggplot2}の例（結果）



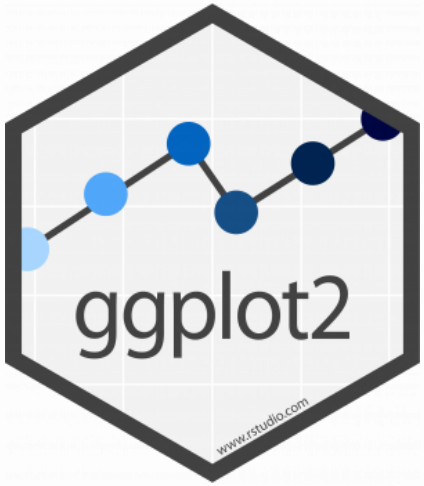
グラフィックの文法

Wilkinsonの「グラフィックの文法」

- Wilkinson, Leland. 2005. *The Grammar of Graphics.* Springer.
- グラフを**構造化**された方法で記述し、**レイヤー (layer; 層)を積み重ねる**ことによってグラフを構築するフレームワーク
- グラフの構成要素の例
  - 横軸と縦軸
    - 目盛りの間隔、ラベルの大きさ
  - 点、線、面
    - 色、太さ、形、透明度など
  - 凡例
  - 図のタイトル
- それぞれの構成要素を一つのレイヤーとして扱い、レイヤーを積み重ねていく

{ggplot2}とは

Hadley Wickhamが大学院生の時に開発した可視化パッケージ



- grammer of graphicsの思想をR上で具現化したもの
- 図の構成要素それぞれに対応する関数が存在し、一つのレイヤーとして機能
  - `ggplot()`: キャンバスを用意
  - `geom_point()`: 点 / `geom_line()`: 線 / `geom_bar()`: 棒
  - `scale_x_continuous()`: 連続変数の横軸
  - `scale_x_discrete()`: 離散変数の横軸など
- 関数を覚える必要は全くない
  - {ggplot2}の仕組みだけを覚え、後はググりながらコーディング

{ggplot2}のイメージ (1)

データの読み込み&ハンドリング

```
library(tidyverse) # {dplyr}, {ggplot2} を含む {tidyverse} を読み込む
df <- read_csv("Data/Micro08.csv") # 第8回の実習データ
df <- df %>% # OECD変数をリコーディングし、OECD_Jへ
   mutate(OECD_J = if_else(OECD == 1, "加盟国", "非加盟国"))
```

{ggplot2}のイメージ (2)

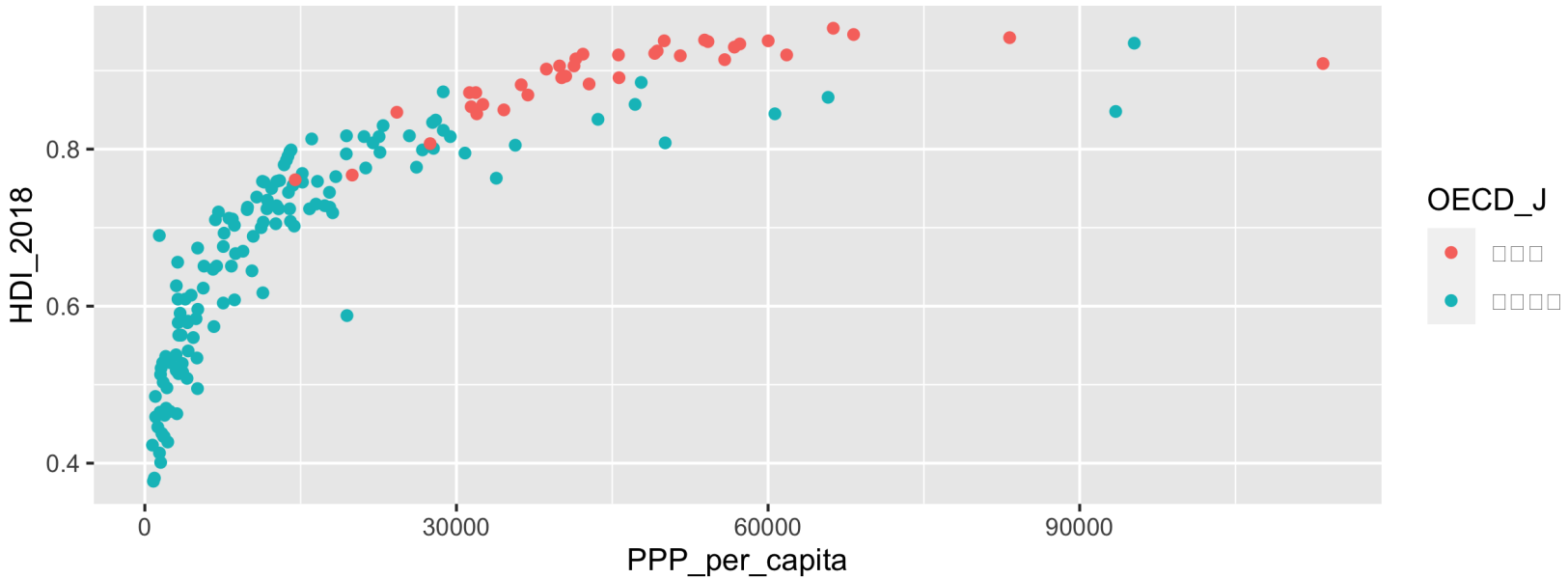
キャンバスの用意

```
df %>% # データdfをggplot()関数に渡し、作図の準備をする
    ggplot()
```

{ggplot2}のイメージ (3)

キャンバス上に点を出力

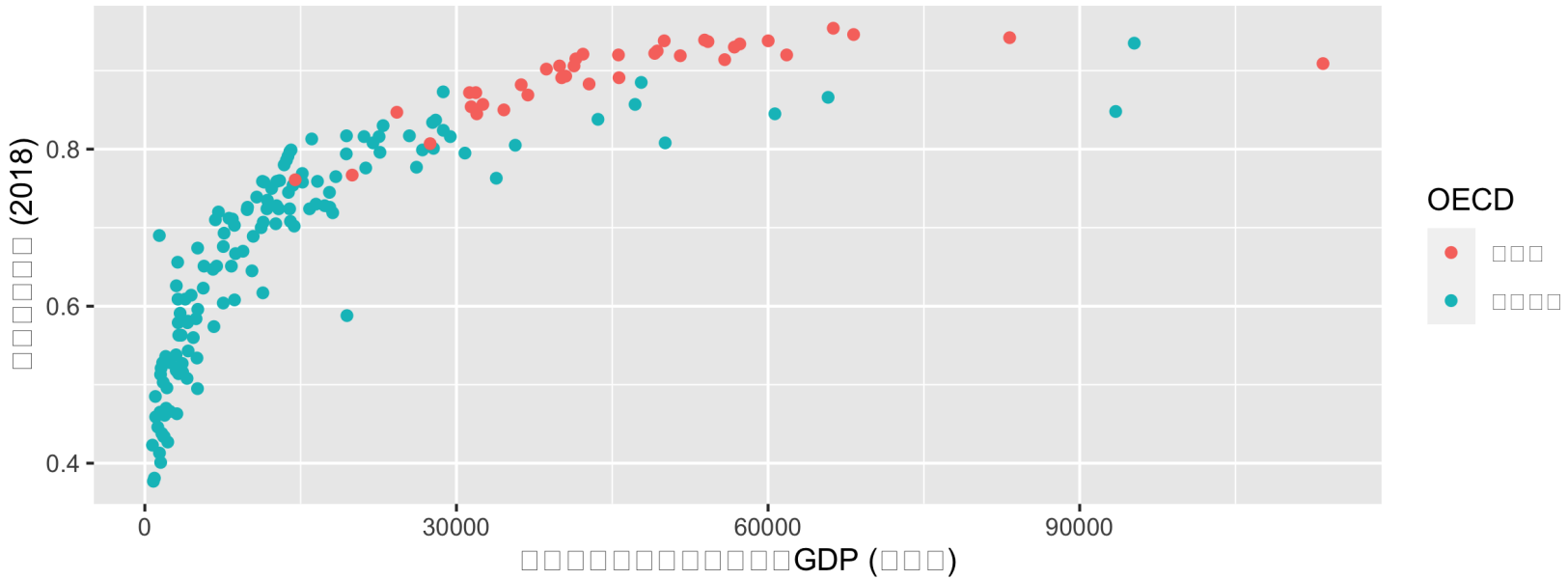
```
df %>%
    ggplot() +
    # 点を出力する。点の横軸上の位置はPPP_per_capita、縦軸上の位置はHDI_2018に対応
    # OECD_Jの値に応じて色分けする。
    geom_point(aes(x = PPP_per_capita, y = HDI_2018, color = OECD_J))
```



{ggplot2}のイメージ (4)

ラベルの修正

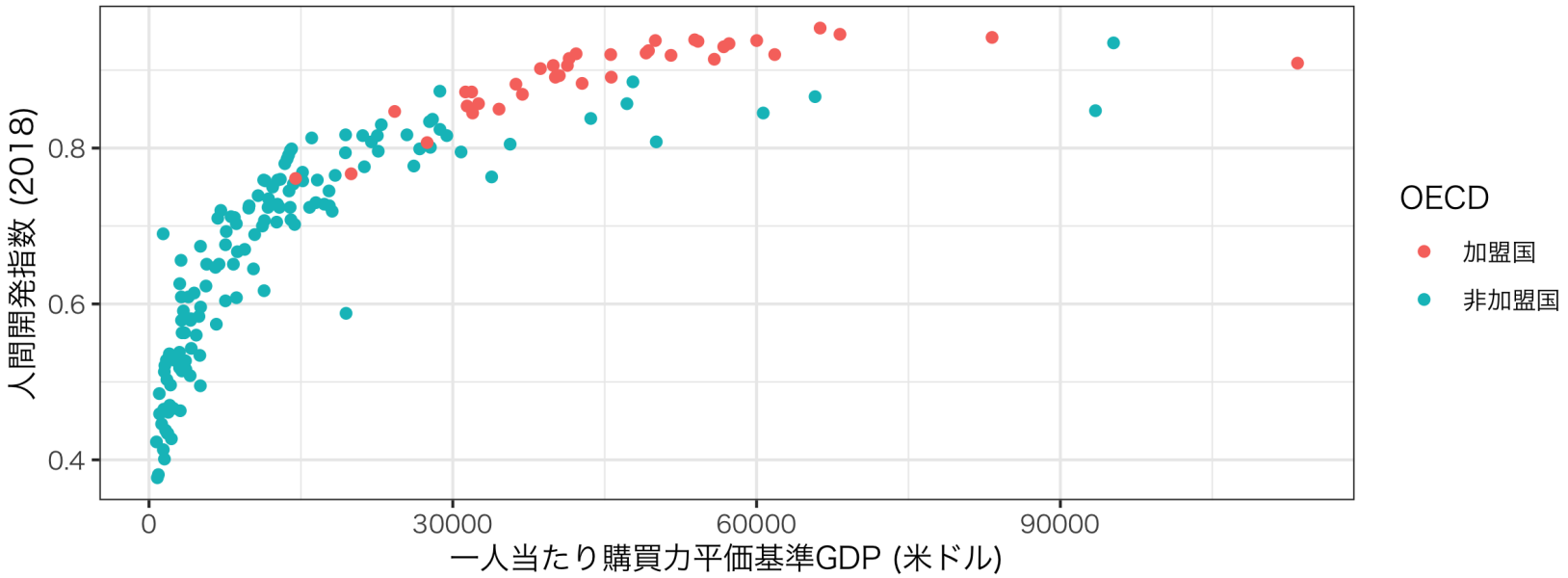
```
df %>%
    ggplot() +
    geom_point(aes(x = PPP_per_capita, y = HDI_2018, color = OECD_J)) +
    labs(x = "一人当たり購買力平価基準GDP （米ドル)", y = "人間開発指数 （2018)",
         color = "OECD")
```



{ggplot2}のイメージ (5)

テーマ変更

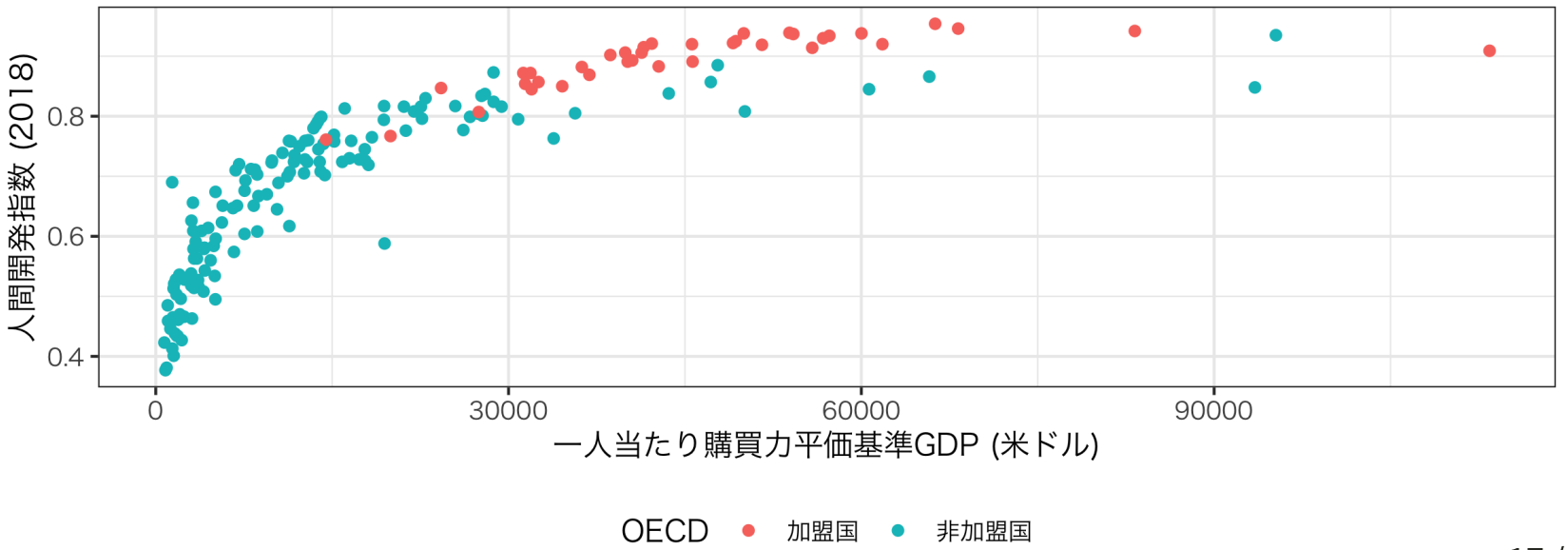
```
df %>%
    ggplot() +
    geom_point(aes(x = PPP_per_capita, y = HDI_2018, color = OECD_J)) +
    labs(x = "一人当たり購買力平価基準GDP (米ドル)", y = "人間開発指数 (2018)",
         color = "OECD") +
    theme_bw(base_family = "HiraKakuProN-W3") # macOSの場合、フォント指定が必要
```



{ggplot2}のイメージ (6)

凡例の位置調整

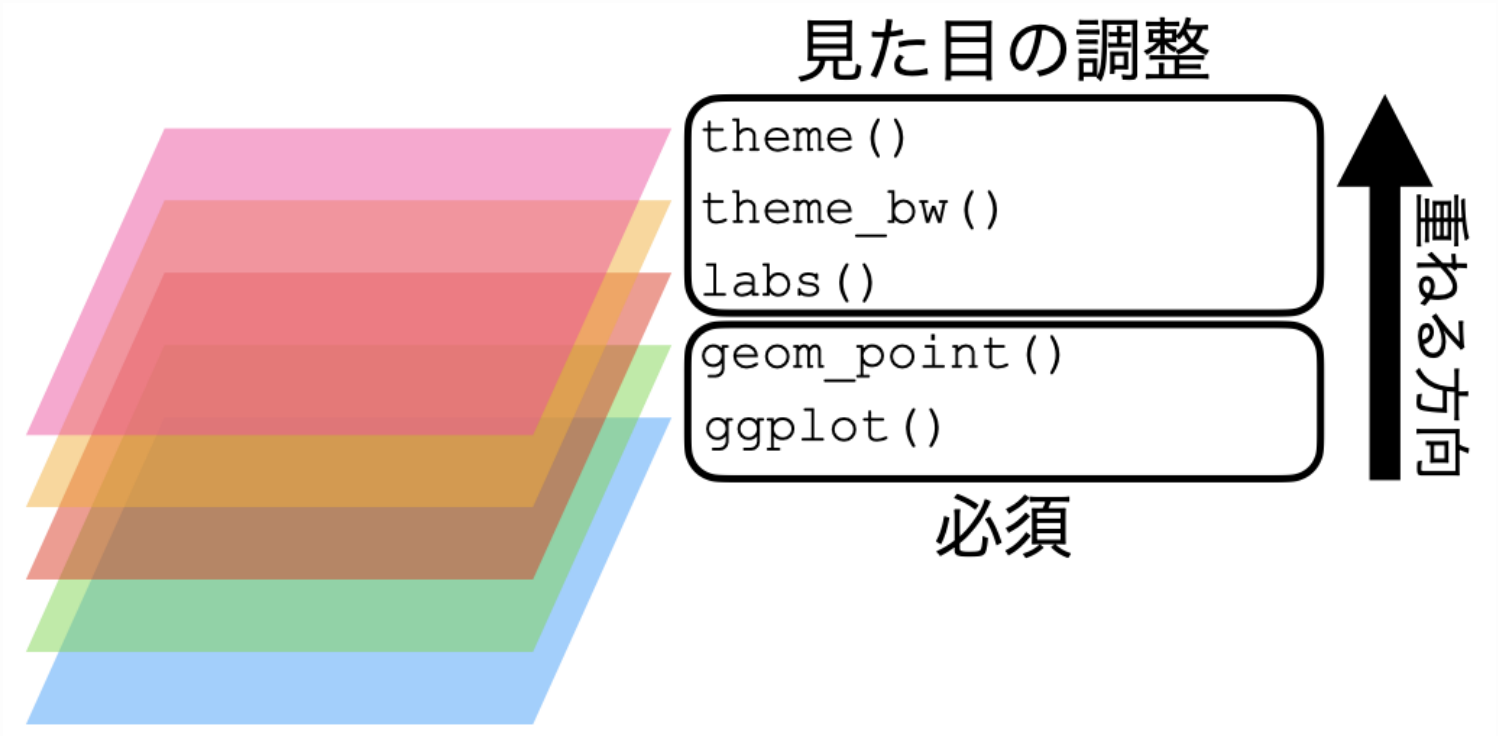
```
df %>%
    ggplot() +
    geom_point(aes(x = PPP_per_capita, y = HDI_2018, color = OECD_J)) +
    labs(x = "一人当たり購買力平価基準GDP (米ドル)", y = "人間開発指数 (2018)",
         color = "OECD") +
    theme_bw(base_family = "HiraKakuProN-W3") +
    theme(legend.position = "bottom") # 凡例を図の下段に
```



{ggplot2}で図が出来上がるまで

レイヤーを積み重ねるイメージ

- 図の核心部は**幾何オブジェクト**（`geom_*()`）と**マッピング**（`aes()`）



グラフの構成要素

{ggplot2}の必須要素

以下の要素があればグラフはとりあえず出来上がる

1. **データ**（Data）
2. **幾何オブジェクト**（Geometry Object）：`geom_*()`関数
    - 散布図、棒グラフ、折れ線グラフ、...
3. **マッピング**（Mapping）：`aes()`関数
    - 散布図の場合、点の位置（横軸と縦軸）
    - 棒グラフの場合、棒の位置（横軸）と高さ（縦軸）
    - 折れ線グラフの場合、線の傾きが変化する点の位置（横軸と縦軸）
4. **座標**系（Coordinate System）：`coord_*()`関数
    - デカルト座標系（直交座標系）、極座標系など
    - 座標系の上限の下限など
    - 座標系は{ggplot2{が自動的に設定してくれるが、カスタマイズ可

凡例の位置、フォント、点の大きさ、軸ラベルの修正などは任意

{ggplot2}の必須要素

座標系はデータに応じて自動的に作成される（カスタマイズ可）

データ

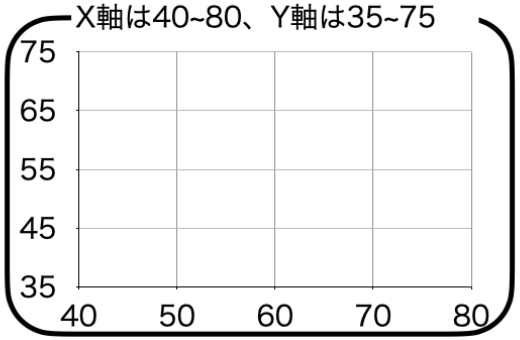
選挙区	投票率	現職得票率	現職の政党
アラバマ	50	37	民主党
アラスカ	61	48	共和党
アリゾナ	70	62	共和党
アーカンソ	43	56	民主党

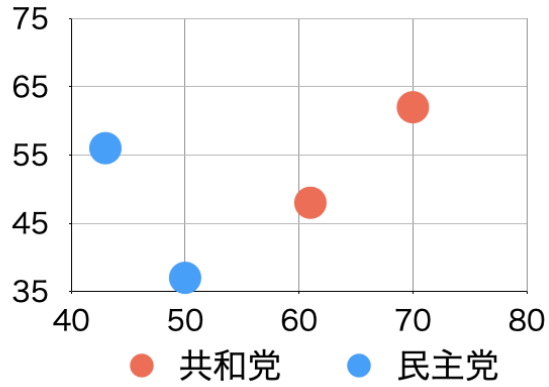
幾何オブジェクト＆マッピング

・散布図を作成
・X軸は投票率、Y軸は現職得票率
・現職の政党で色分け

```
geom_point(aes(x     = 投票率,
               y     = 現職得票率,
               color = 現職の政党))
```

座標系
X軸は40~80、Y軸は35~75





書き方

- **注意:** レイヤーの積み重ねは`%>%`でなく`+`を使用
  - オブジェクトを**渡す**のではなく、レイヤーを**足す**という意味
- 可視化は`ggplot()`からスタート
- 幾何オブジェクトは`geom_`で始まる関数
- 幾何オブジェクト内には`mapping =`でマッピングが必要。
  - 第一引数であるため、`mapping =`は省略し、`aes()`からスタートでOK
- `aes()`の中にはグラフ上に出力される点、線、面などがデータのどの変数に対応するかを記述

```
ggplot(data = データ) +
   幾何オブジェクト関数(mapping = aes(マッピング))
```

データ

- 使用するデータ構造はデータフレーム、またはtibble
- `ggplot()`関数の第一引数（`data =`）で指定
  - `data =`は省略し、`ggplot(データ名)`でもOK
- `データ名 %>% ggplot()`も可能（推奨）
  - データハンドリング後、そのまま可視化を行う場合

```
# データ指定方法（1）
ggplot(data = データ名)

# データ指定方法（2）
ggplot(データ名)

# データ指定方法（3）
# {dplyr}、{tidyr}との組み合わせが可能
データ名 %>%
   ggplot()
```

幾何オブジェクト

```
データ名 %>%
    ggplot() +
    幾何オブジェクト関数()
```

指定されたデータを使ってどのような図を作成するか

- 散布図: `geom_point()`
- 棒グラフ: `geom_bar()`
- 折れ線グラフ: `geom_line()`
- ヒストグラム: `geom_histogram()`
- 箱ひげ図: `geom_boxplot()`
- など

{ggplot2}が提供する幾何オブジェクトも数十種類があり、ユーザーが開発・公開した幾何オブジェクトなどもある

- 非巡回有向グラフ作成のための{ggdag}、ネットワークの可視化のための{ggnetwork}など

マッピング

グラフ上の**点、線、面などの情報をデータと変数に対応**させる

- プロット上に出力されるデータの具体的な在り方を指定する
- 散布図の例) 各点の横軸と縦軸における位置情報
- `geom_*()` 内の `aes()` 関数で指定
  - グラフに複数の幾何オブジェクトが存在し、マッピング情報が同じなら `ggplot()` 内で指定することも可能

例) `geom_point(aes(x = PPP_per_capita, y = HDI_2018, color = OECD_J))`

幾何オブジェクト	マッピング情報	引数	対応する変数
`geom_point()`	点の横軸上の位置	`x`	`PPP_per_capita`
`geom_point()`	点の縦軸上の位置	`y`	`HDI_2018`
`geom_point()`	点の色	`color`	`OECD_J`

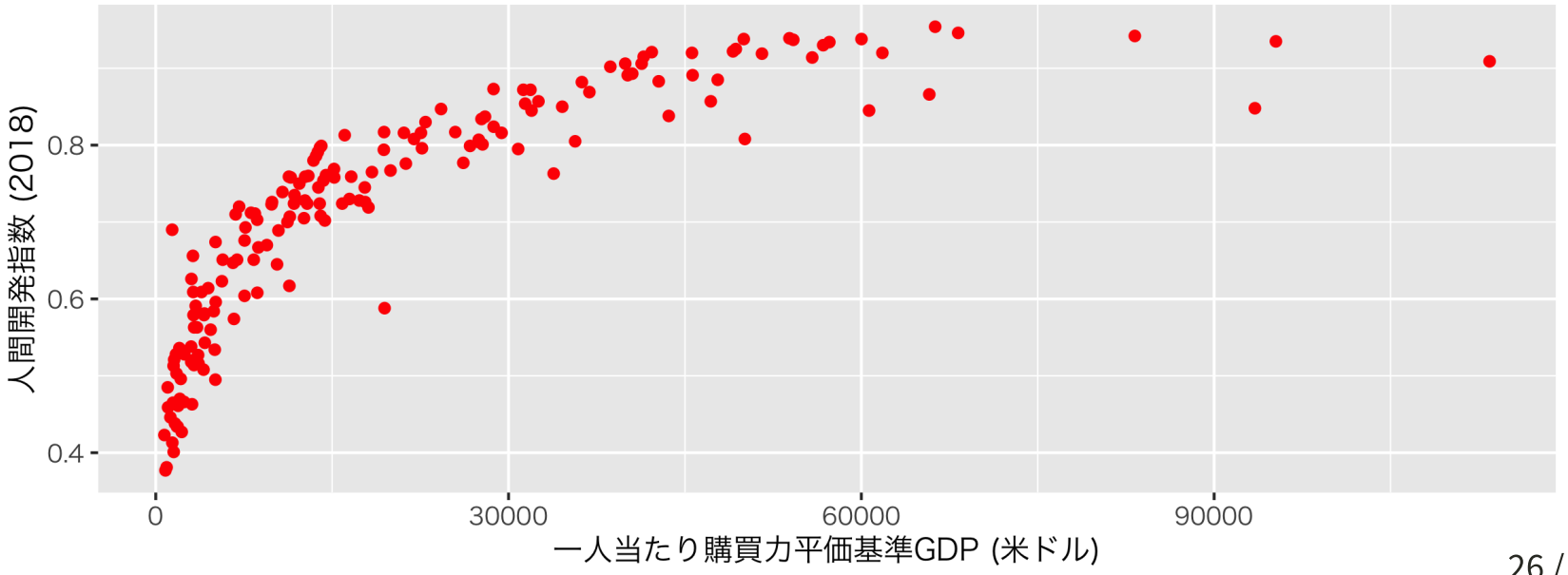
- 点、線、面が持てる情報は他にも色々
  - 大きさ (`size`)、線の種類 (`linetype`)、透明度 (`alpha`)、面の色 (`fill`)、点の形 (`shape`)、グループ (`group`)など

マッピング時の注意

`aes()` の中で指定するか、外で指定するかで挙動が変わる (教科書第17.3.4章)

- `aes()` 内で `color` を指定する場合、**それぞれの点**が指定された変数の値に応じて色分けされる
- `aes()` の外側で `color` を指定する場合、**全ての点**が指定された色となる

```
ggplot(df) +
  geom_point(aes(x = PPP_per_capita, y = HDI_2018), color = "red") +
  labs(x = "一人当たり購買力平価基準GDP (米ドル)", y = "人間開発指数 (2018)")
```



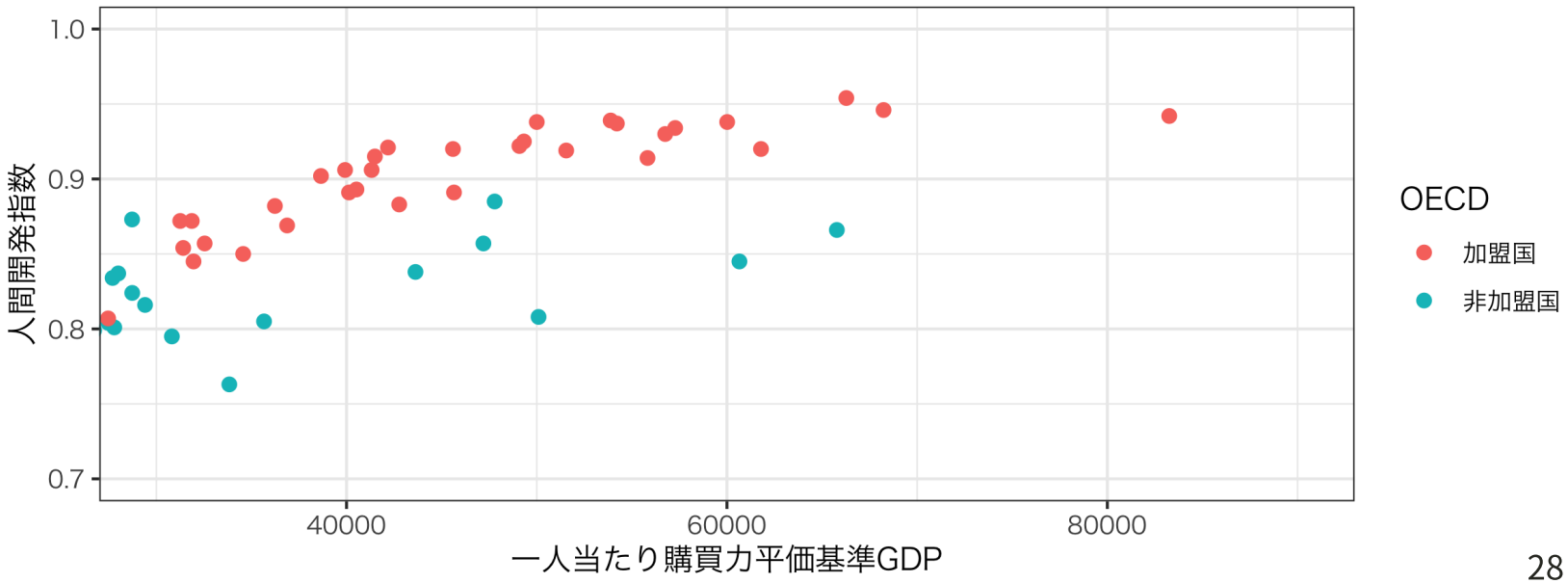
座標系

- 直交座標系の拡大・縮小: `coord_cartesian()`
  - 最もよく使う機能
- 横軸と縦軸の交換: `coord_flip()`
- 横軸と縦軸比の固定: `coord_fixed()`
- 極座標系 (polar coordinates system)へ変換: `coord_polar()`
  - 円グラフを作成する際に使われるが、**円グラフは邪悪なる存在**であるため、省略

直交座標系拡大の例

横軸を30000〜90000、縦軸を0.7〜1にする

```
df %>%
    ggplot() +
    geom_point(aes(x = PPP_per_capita, y = HDI_2018, color = OECD_J),
               size = 2) +
    labs(x = "一人当たり購買力平価基準GDP", y = "人間開発指数", color = "OECD") +
    coord_cartesian(xlim = c(30000, 90000), ylim = c(0.7, 1)) +
    theme_bw(base_family = "HiraKakuProN-W3")
```



スケール (Scale)

マッピング要素のカスタマイズ

- 横/縦軸の目盛り変更、色分けの色を指定など
- `scale_*_*()` 関数を使用
  - `scale_マッピング要素_対応する変数のタイプ()`
- 詳細は次週以降

ファセット (Facet)

グラフを2つ以上の面で分割

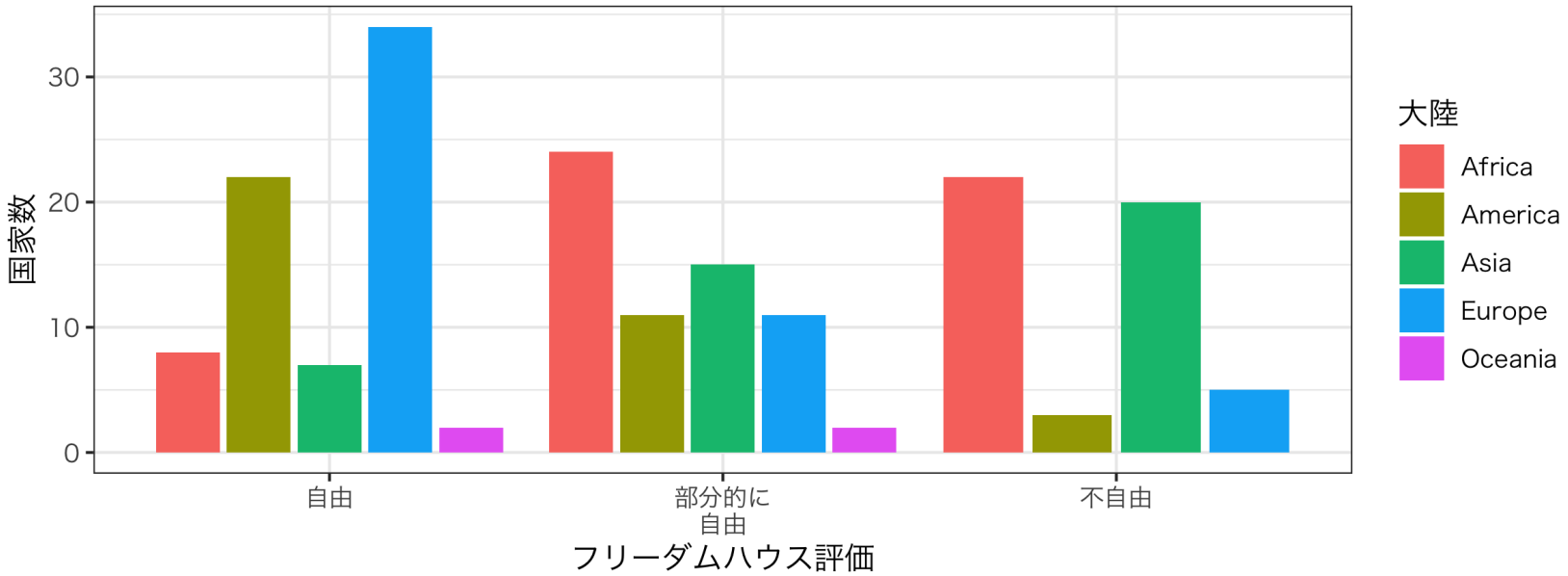
- 例) `FH_Stauts` の棒グラフを大陸ごとに出力

```
# データの用意
df <- df %>%
   mutate(FH_Status = case_when(FH_Status == "F"  ~ "自由",
                                FH_Status == "PF" ~ "部分的に\n自由",
                                TRUE              ~ "不自由"),
          FH_Status = factor(FH_Status,
                             levels = c("自由", "部分的に\n自由", "不自由"))) %>
   drop_na(FH_Status)
```

ファセット分割なし

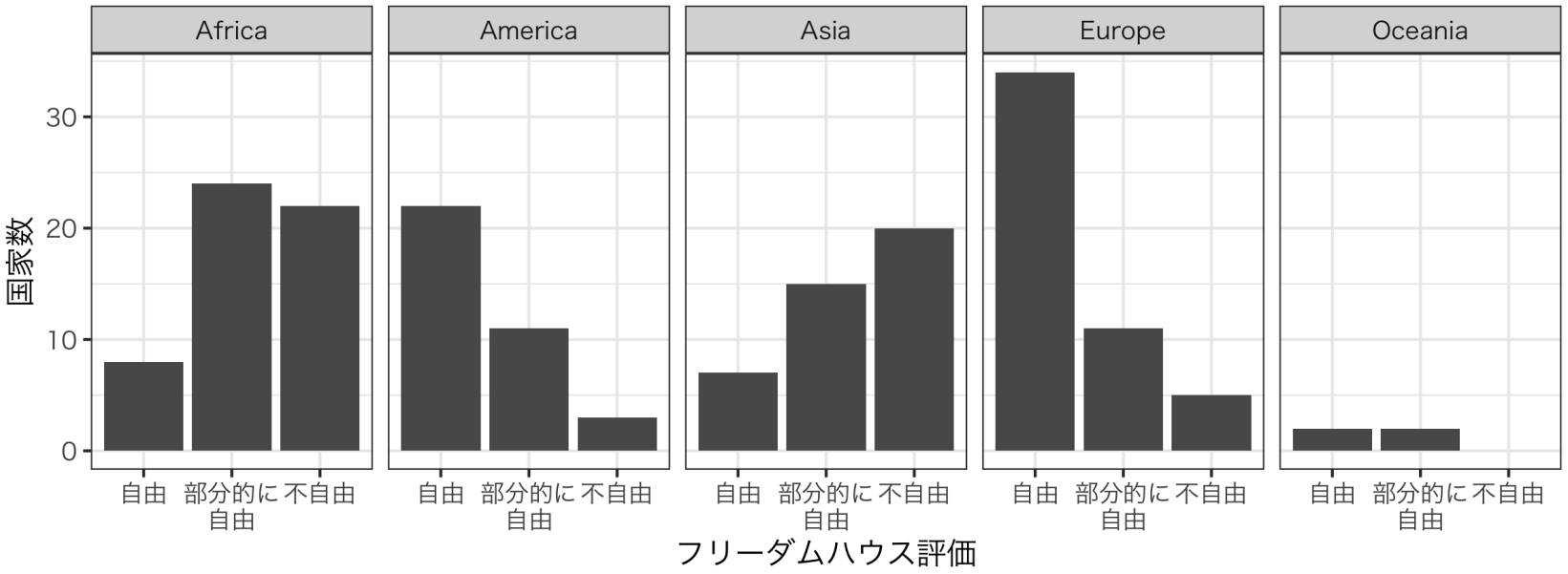
同じ大陸内の`FH_Stauts`の分布を確認するには不向き

```
df %>%
    ggplot() +
    geom_bar(aes(x = FH_Status, fill = Continent),
             position = position_dodge2(1/2)) +
    labs(x = "フリーダムハウス評価", y = "国家数", fill = "大陸") +
    theme_bw(base_family = "HiraKakuProN-W3")
```



ファセット分割あり

```
df %>%
   ggplot() +
   geom_bar(aes(x = FH_Status),
            position = position_dodge2(1/2)) +
   labs(x = "フリーダムハウス評価", y = "国家数") +
   facet_wrap(~ Continent, ncol = 5) +
   theme_bw(base_family = "HiraKakuProN-W3")
```



良いグラフとは

意識すべきところ

- データ・インク比
- カラーユニバーサルデザイン
- 円グラフは邪悪なる存在
- 3次元グラフは更に邪悪なる存在
- 3次元円グラフは概念レベルで駆逐すべき存在

参考図書 (日本語)

1と4は{ggplot2}の教科書としても優れている

1. Hadley Wickham・Garrett Grolemund(著), 黒川利明(訳). 2017. 『Rではじめるデータサイエンス』オライリージャパン.
2. 藤俊久仁・渡部良一. 2019. 『データビジュアライゼーションの教科書』秀和システム.
3. 永田ゆかり. 2020. 『データ視覚化のデザイン』SBクリエイティブ.
4. キーラン・ヒーリー(著), 瓜生真也・江口哲史・三村喬生(訳). 2021.『データ分析のためのデータ可視化入門』講談社.**(おすすめ!)**

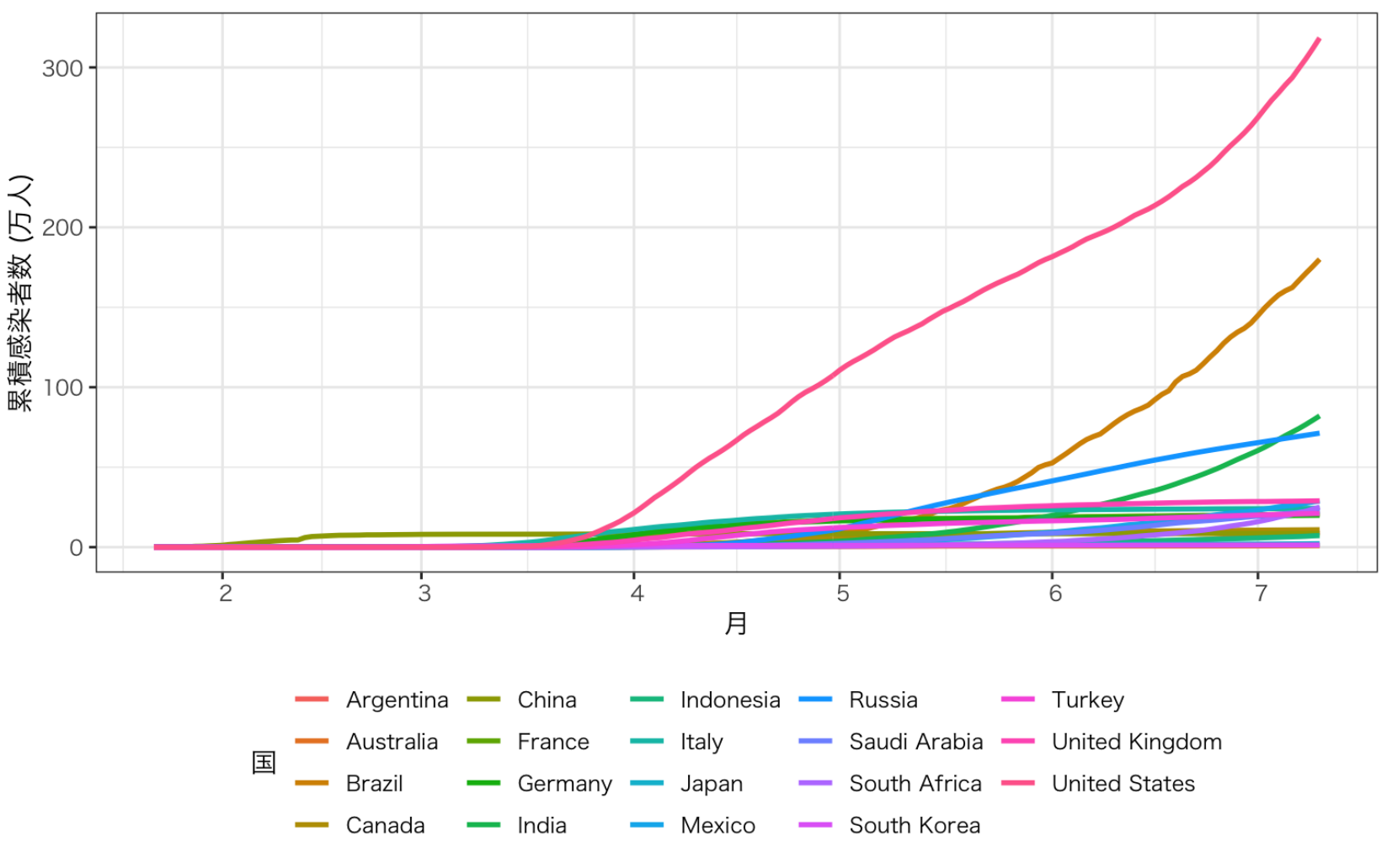
データ・インク比 (Data-ink ratio)

Edward R. Tufte. 2001. *The Visual Display of Quantitative Information (2nd Ed).* Graphics Press.

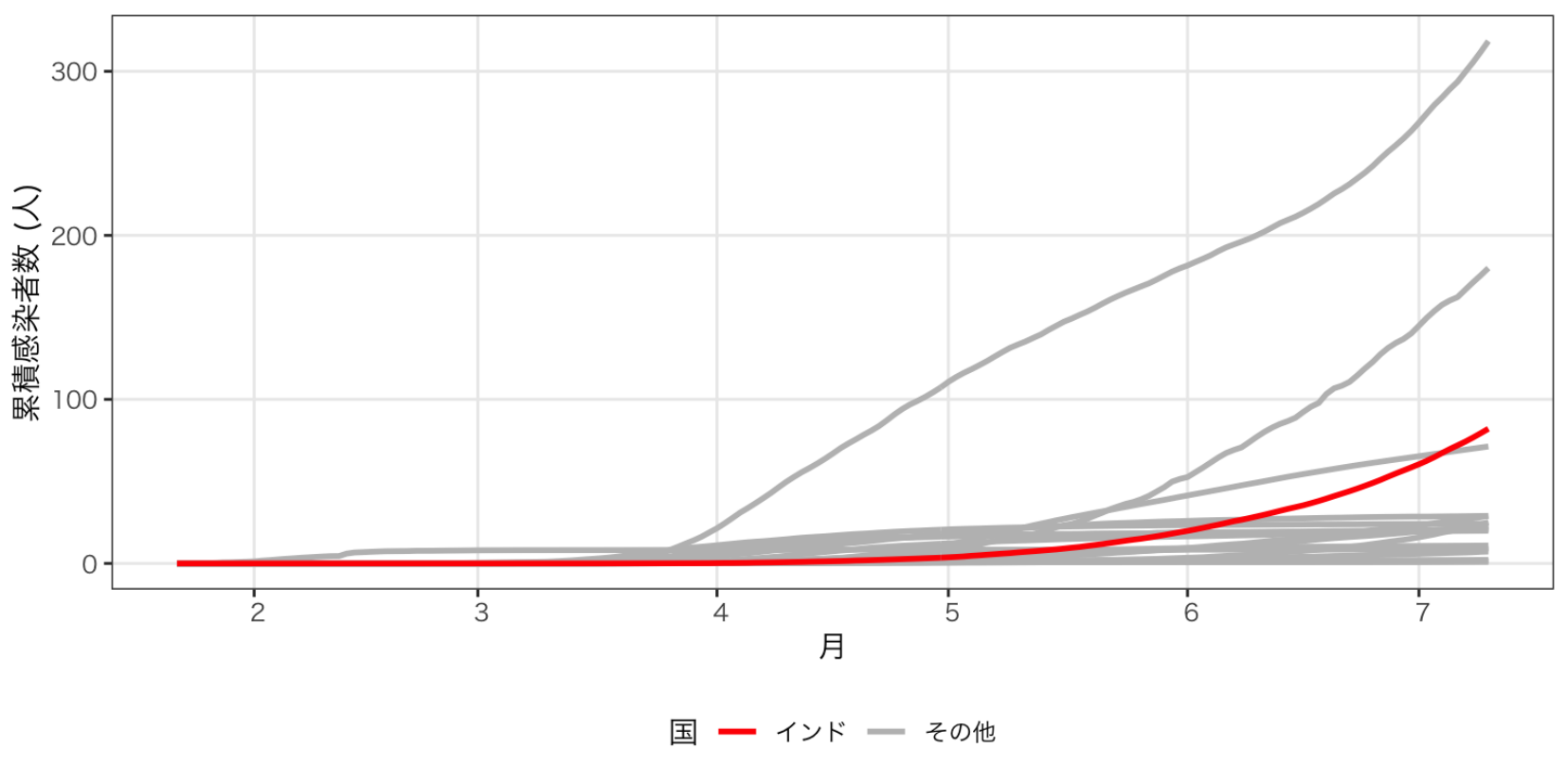
$$\text{データ・インク比} = \frac{\text{データの情報を含むインクの量}}{\text{グラフの出力に使用されたインクの総量}}$$

- 良いグラフとはデータ・インク比を最大化したグラフ
- グラフにおいて情報損失なしに除去できる要素が占める割合を1から引いたもの
- 色分けにも注意

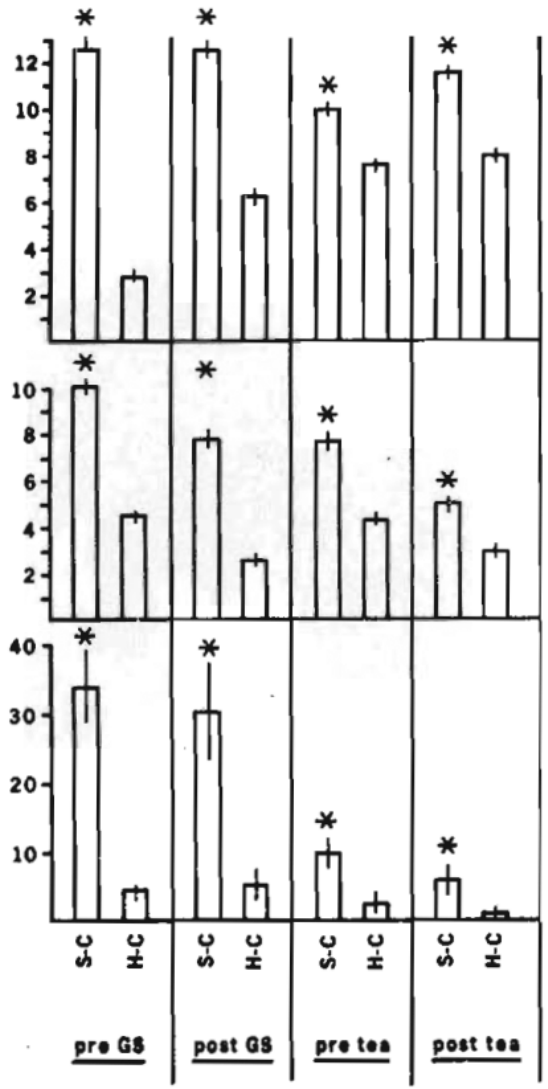
色分けに注意 (1)

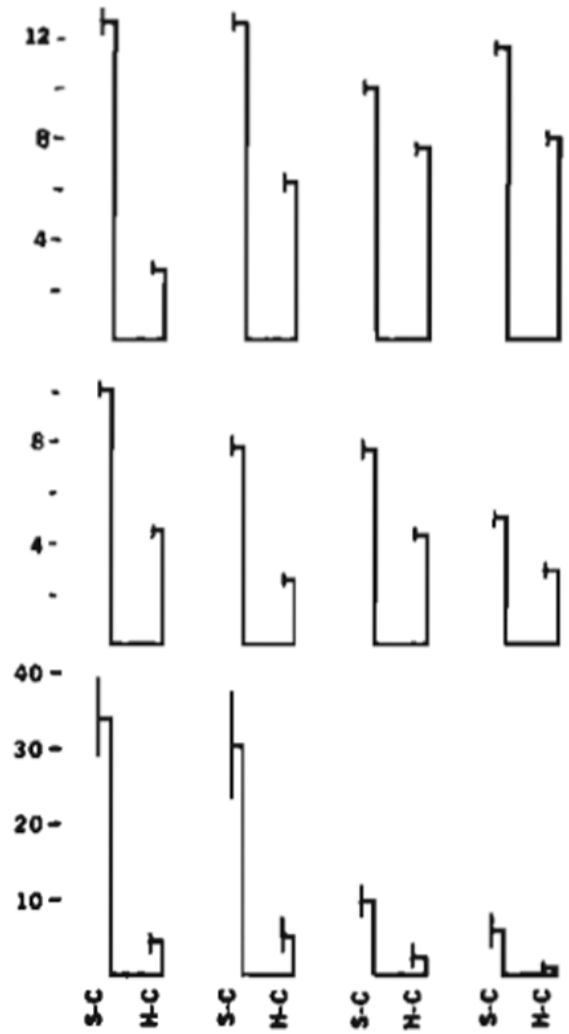

累積感染者数 (万人)
300
200
100
0
2
3
4
5
6
7
月
国
Argentina
Australia
Brazil
Canada
China
France
Germany
India
Indonesia
Italy
Japan
Mexico
Russia
Saudi Arabia
South Africa
South Korea
Turkey
United Kingdom
United States

色分けに注意 (2)



やり過ぎにも注意

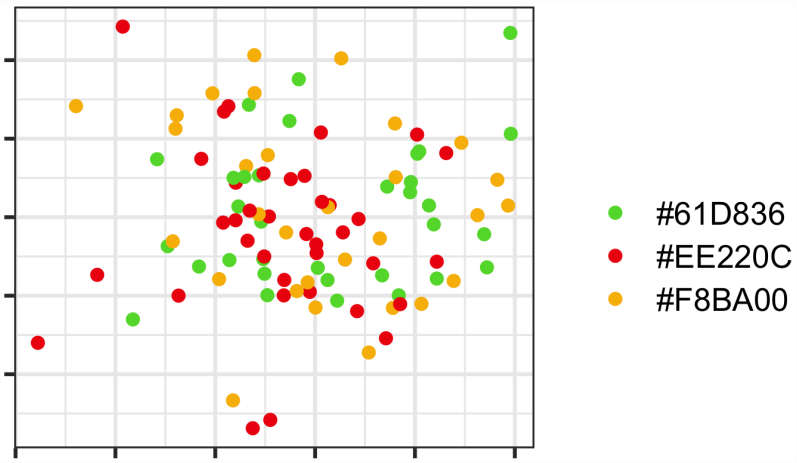


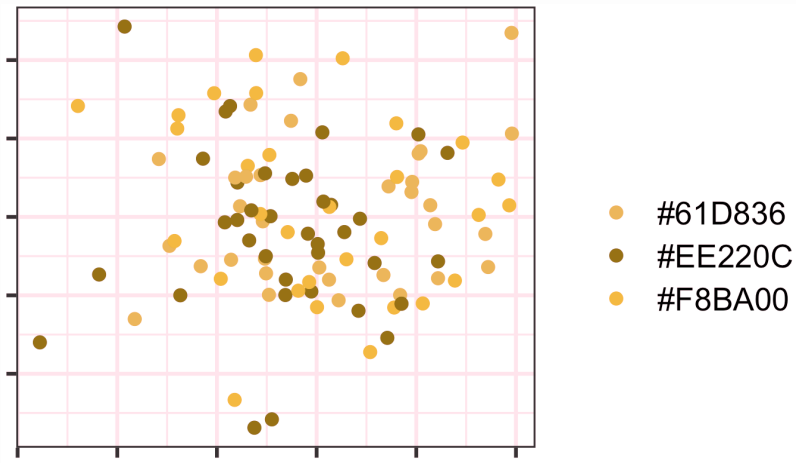


カラーユニバーサルデザイン

色分けを行う際には注意が必要

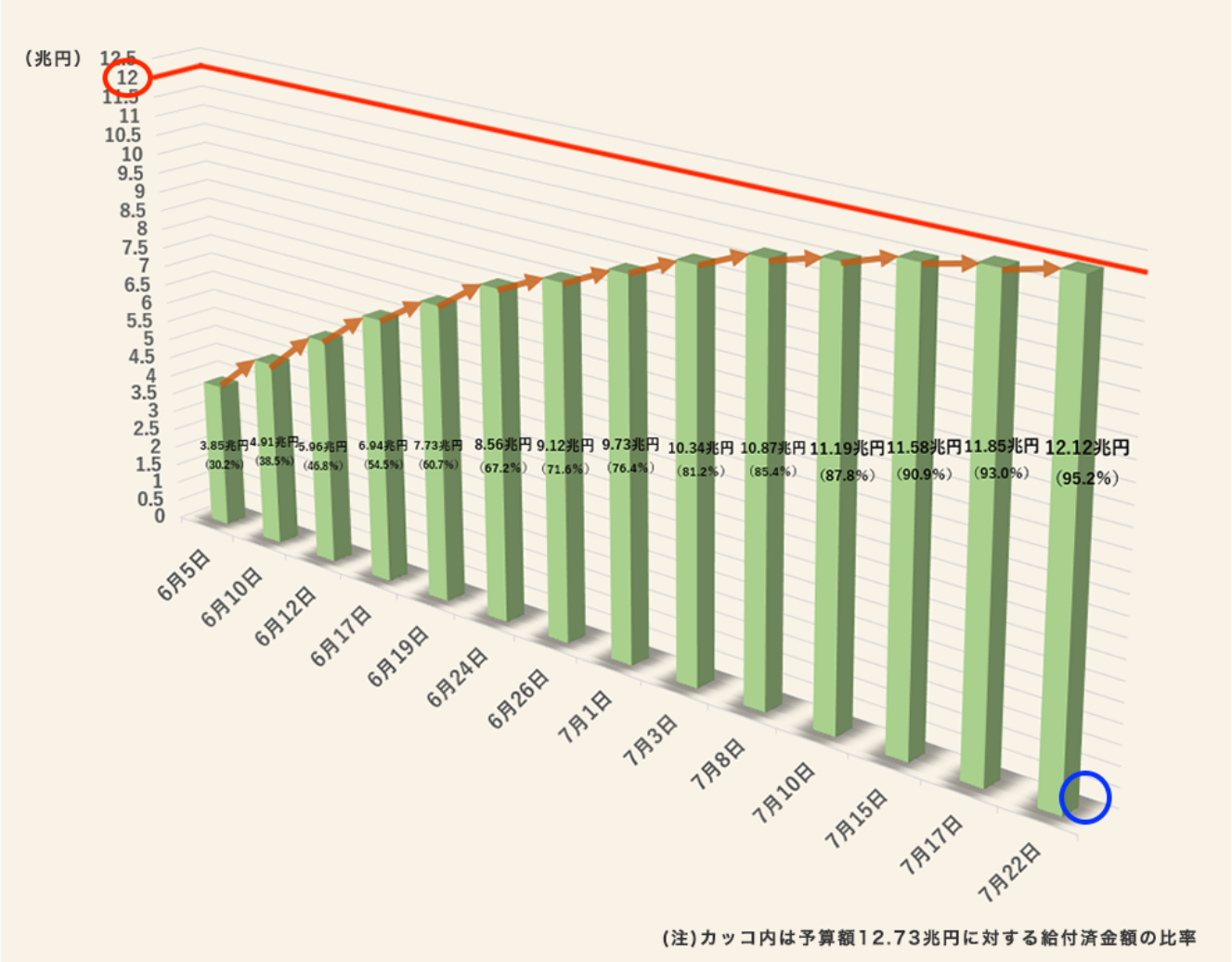
- P型およびD型色弱の場合、緑と赤の認識が困難
  - 日本の場合、男性の5%、女性の0.2%
  - フランス・北欧の場合、男性の約10%
- 色覚シミュレーターで確認可能
  - macOS用のSim Daltonismを使用した第二色盲 (deuteranopia)の例
  - Linux/WindowsならColor Oracleなど





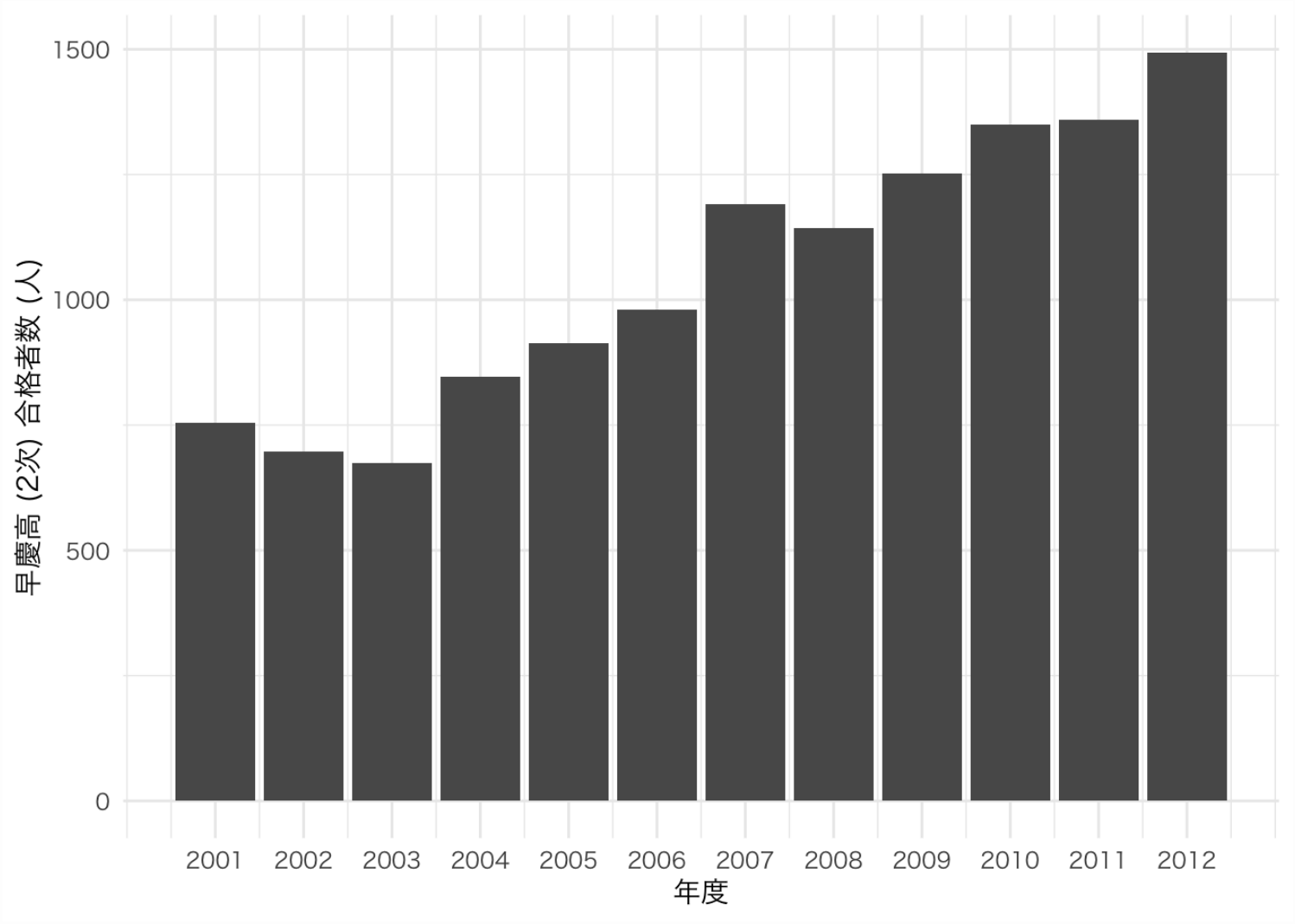
自分が好きな色でなく、誰にも見やすい色を使う

3次元グラフについて (1)



3次元グラフについて (2)

3次元グラフについて (3)



まとめ

今回の内容

よく分からない箇所は教科書を読み返す or 宋&TAに質問 (できれば、**LMSの質問コーナー**で)

- グラフィックの文法と{ggplot2}: 教科書第17.1章
- グラフの構成要素: 教科書第17.2章
- 良いグラフとは: 教科書第17.3章

課題

なし

- ただし、スライド上のコードを直接実行してみることを推奨