

ミクロ政治データ分析実習

第13回 可視化（3）

そん じえひよん

宋 財 泓

関西大学総合情報学部

2021/7/8 (updated: 2021-07-05)

授業評価アンケート

講義動画を視聴する前に回答すること（約5分所要）

- PCの場合
 - https://ctl.jm.kansai-u.ac.jp/q/Ans?ms&cd=13490_21sから回答
- スマホ/タブレットの場合
 - 上記のURLか下のQRコードからアクセス



期末テストについて

概要

- **日時:** 2021年7月15日（木）4限
- **場所:** TC304教室（座席指定；座席表は事前送付する）
- **持参物:** 学生証のみ。PCは大学のPCを使用（持ち込みPC不可）
- **持ち込み:** 可（インターネット使用可能）
 - 携帯は**使用禁止**
- 宋は問題そのものに関する質問以外、一切受け付けない
 - エラーメッセージへの対処もテストの一部
- 不正行為厳禁

実施形態

本講義の課題と同じ形式

1. LMSからデータ、問題ファイル、サンプルファイルをダウンロード
2. サンプルファイルと同じ結果が得られるように問題ファイルを編集
3. .Rmdファイルと.htmlファイル**両方**LMSに提出

5 Named Graphs (5NG)

- 棒グラフ (bar plot)
 - `geom_bar()`
- ヒストグラム (histogram)
 - `geom_histogram()`
- 箱ひげ図 (box plot / box-and-whisker plot)
 - `geom_boxplot()`
- 散布図 (scatter plot)
 - `geom_point()`
- 折れ線グラフ (line plot)
 - `geom_line()`

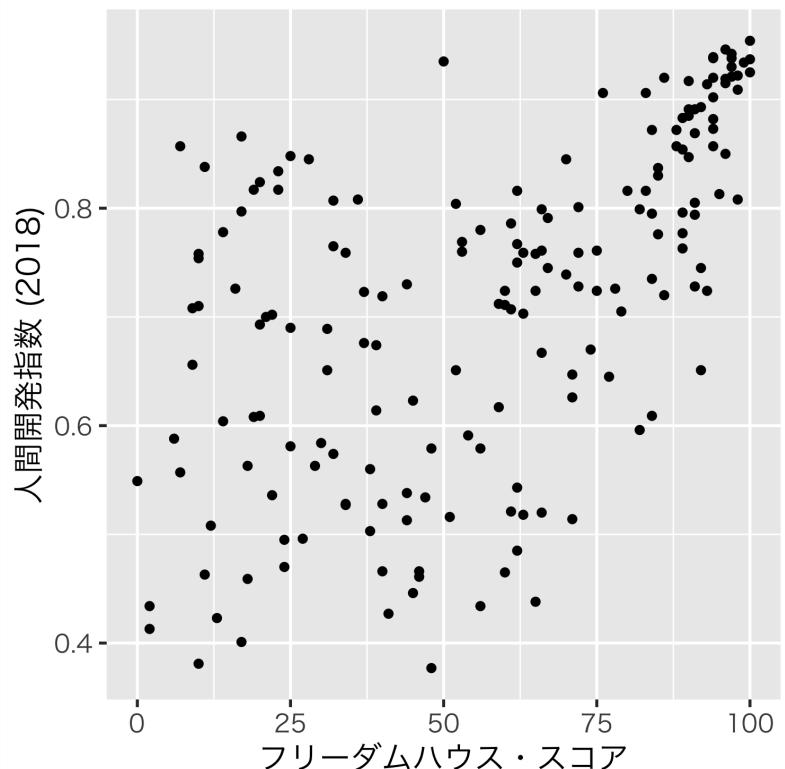
geom_point(): 散布図

散布図の必須要素

散布図を作成する際に必要な**最低限**の情報

データにフリーダムハウス・スコアと人間開発指数の**列**が必要

- **x** : 点の横軸上の位置
 - フリーダムハウス・スコア
- **y** : 点の縦軸上の位置
 - 人間開発指数
- 原因と結果の関係（因果関係）が考えられる2変数の場合、**原因を横軸、結果を縦軸**にする。



データの読み込み

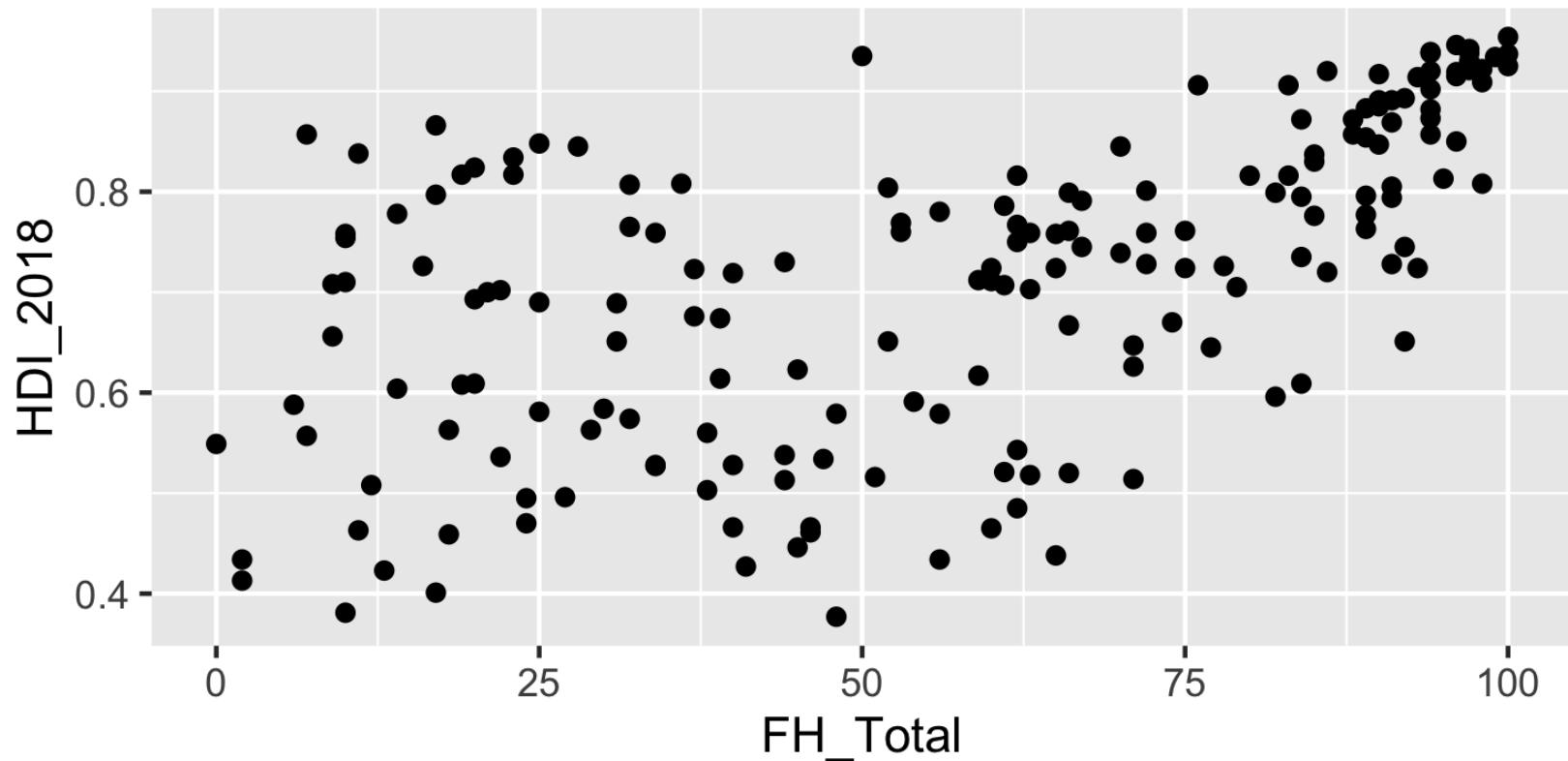
第8回実習用データ（Micro08.csv）を使用

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

散布図の作成 (1)

幾何オブジェクトは `geom_point()` を使用

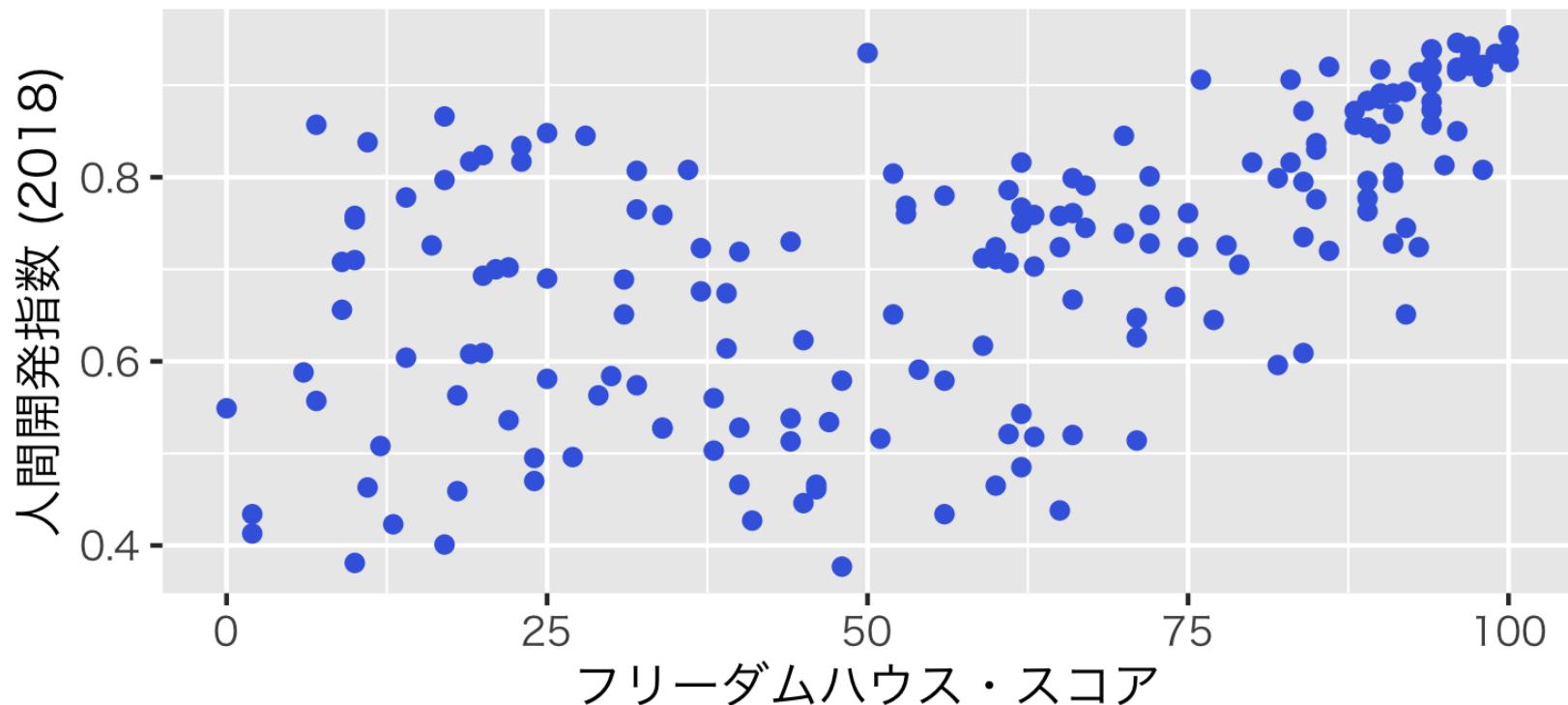
```
ggplot(df) +  
  geom_point(aes(x = FH_Total, y = HDI_2018))
```



散布図の作成 (2)

ラベル修正 / 必要に応じてカスタマイズ

```
ggplot(df) +  
  geom_point(aes(x = FH_Total, y = HDI_2018), color = "royalblue") +  
  labs(x = "フリーダムハウス・スコア", y = "人間開発指数 (2018)") +  
  theme_gray(base_family = "HiraKakuProN-W3")
```



次元の追加

散布図における次元の追加

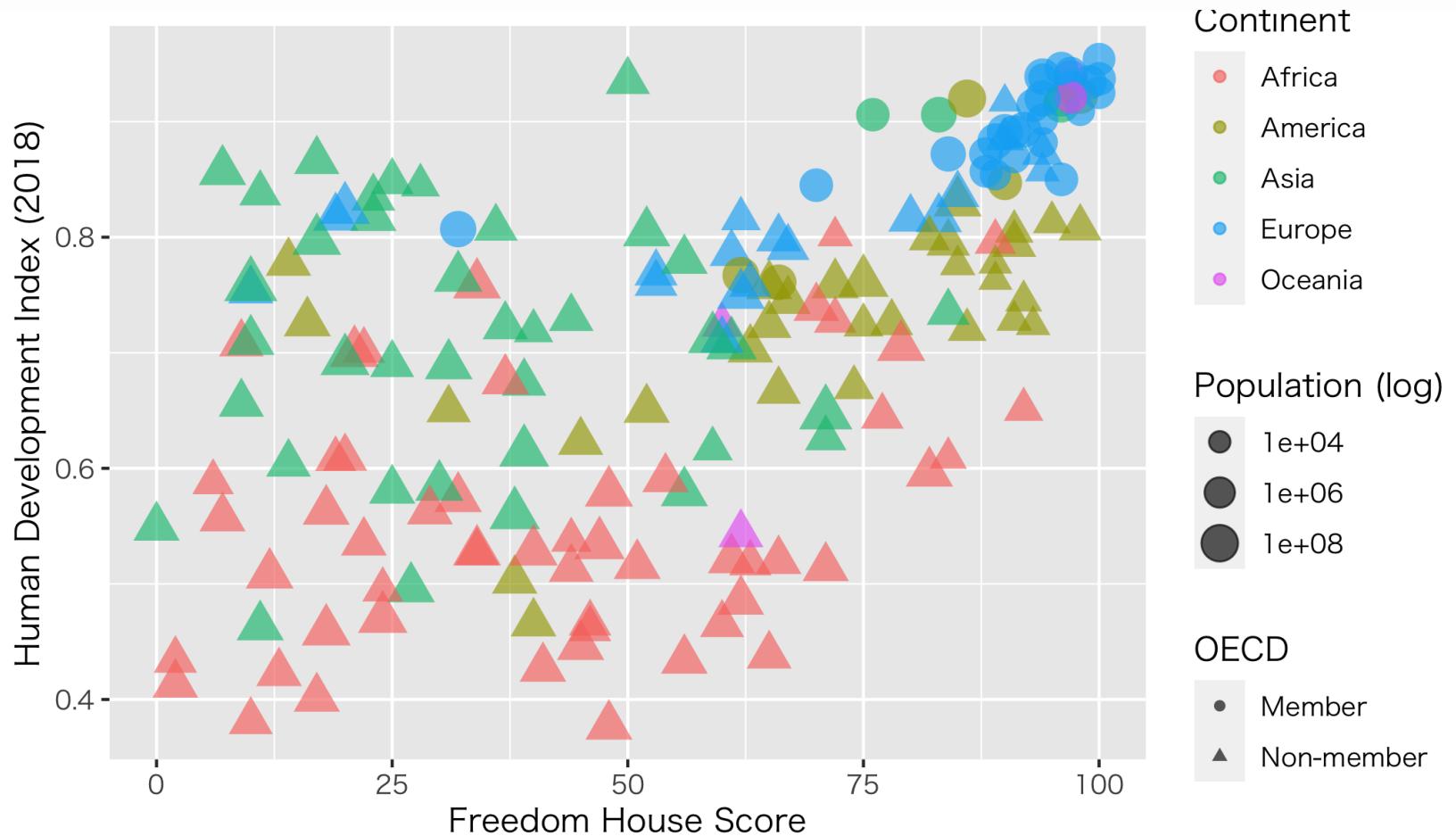
- それぞれの点は各國の (1) フリーダムハウス・スコアと (2) 人間開発指数の情報を持つ
 - = 2次元
- 更に情報を持たせるためには...
 - 色分け: `color`
 - 大きさ: `size`
 - 形: `shape`
 - 透明度(非推奨): `alpha`
 - ファセット分割
- 5次元以上のグラフも可能だが、3次元までを推奨
 - 一つのグラフの情報が多すぎると読みにくくなる
 - 4次元以上になりそうならファセット分割で対応
 - 3次元の場合、色分け (`color`) を推奨するが、白黒の図を作成する場合は形 (`shape`) を使う。

5次元の散布図

```
df %>%
  mutate(OECD = if_else(OECD == 1, "Member", "Non-member")) %>%
  ggplot() +
  geom_point(aes(x = FH_Total, y = HDI_2018, color = Continent,
                 size = Population, shape = OECD), alpha = 0.65) +
  scale_size_continuous(trans = "log10") +
  labs(x = "Freedom House Score", y = "Human Development Index (2018)",
       size = "Population (log)") +
  theme_gray()
```

5次元の散布図

各点はフリーダムハウス・スコア、人間開発指数、人口（対数変換）、大陸、OECD加盟有無の情報を持つ



色分けの例（コード）

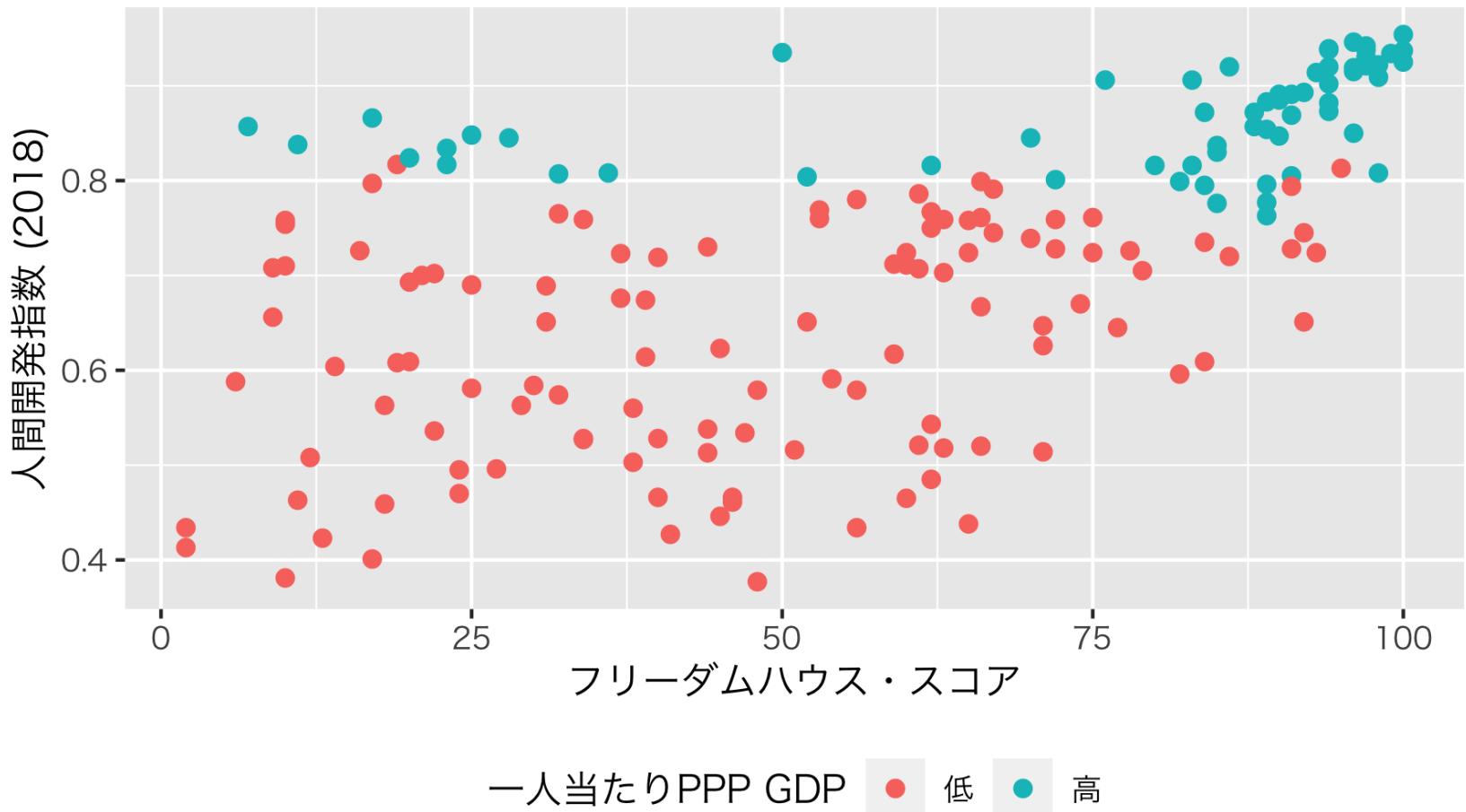
`aes()` の内部に `color = 色分けする変数` を指定

- `HighIncome` 変数を作成し、`PPP_per_capita` が2万以上なら "高"、未満なら "低"
- `HighIncome` 値に応じて点の色分けを行う (`color = HighIncome`)
- 点の大きさは2とする (全体に適用させるため、`aes()` の外側に指定)
- `labs()` 内で凡例タイトルを修正; 凡例を下側へ移動

```
Scatter1 <- df %>%
  drop_na(FH_Total, HDI_2018, PPP_per_capita) %>%
  mutate(HighIncome = if_else(PPP_per_capita >= 20000, "高", "低")) %>%
  ggplot() +
  geom_point(aes(x = FH_Total, y = HDI_2018, color = HighIncome),
             size = 2) +
  labs(x = "フリーダムハウス・スコア", y = "人間開発指数（2018）",
       color = "一人当たりPPP GDP") +
  theme_gray(base_family = "HiraKakuProN-W3") +
  theme(legend.position = "bottom")
```

色分けの例（図）

```
print(Scatter1)
```

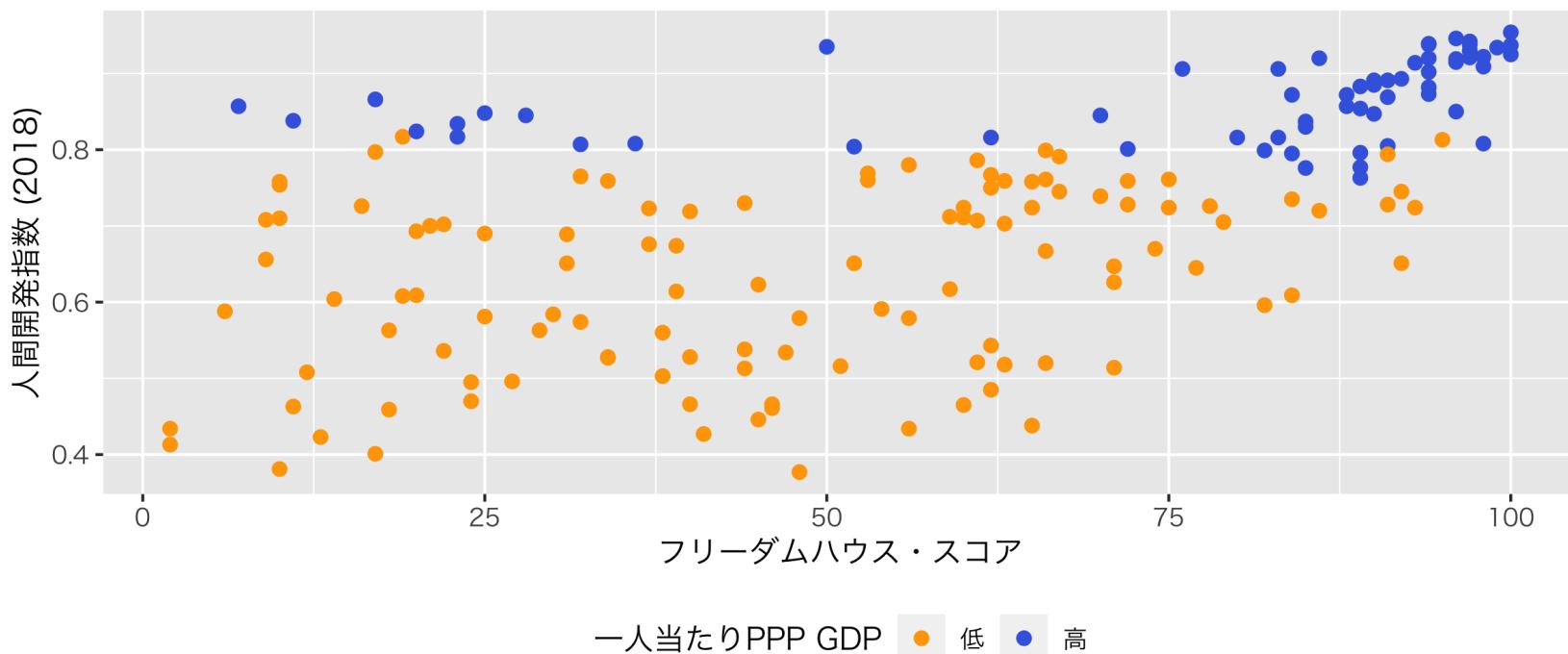


色のカスタマイズ

color にマッピングされている変数（HighIncome）が離散変数

- scale_color_manual() を使用
- 引数: values = c("値1" = "色1", "値2" = "色2", ...)

```
Scatter1 +
  scale_color_manual(values = c("低" = "orange", "高" = "royalblue"))
```



{ggplot2}で使える色

- 文字列で指定
 - "red"、"blue"、"royalblue"など（全657種類）
 - Googleなどで「ggplot2 color」で検索（[例](#)）
- RGBカラー（HEXコード）で指定
 - 細かく調整可能（16,777,216種類の色）
 - 例) "#FF0000":赤、"#4169E1":ロイヤルブルー

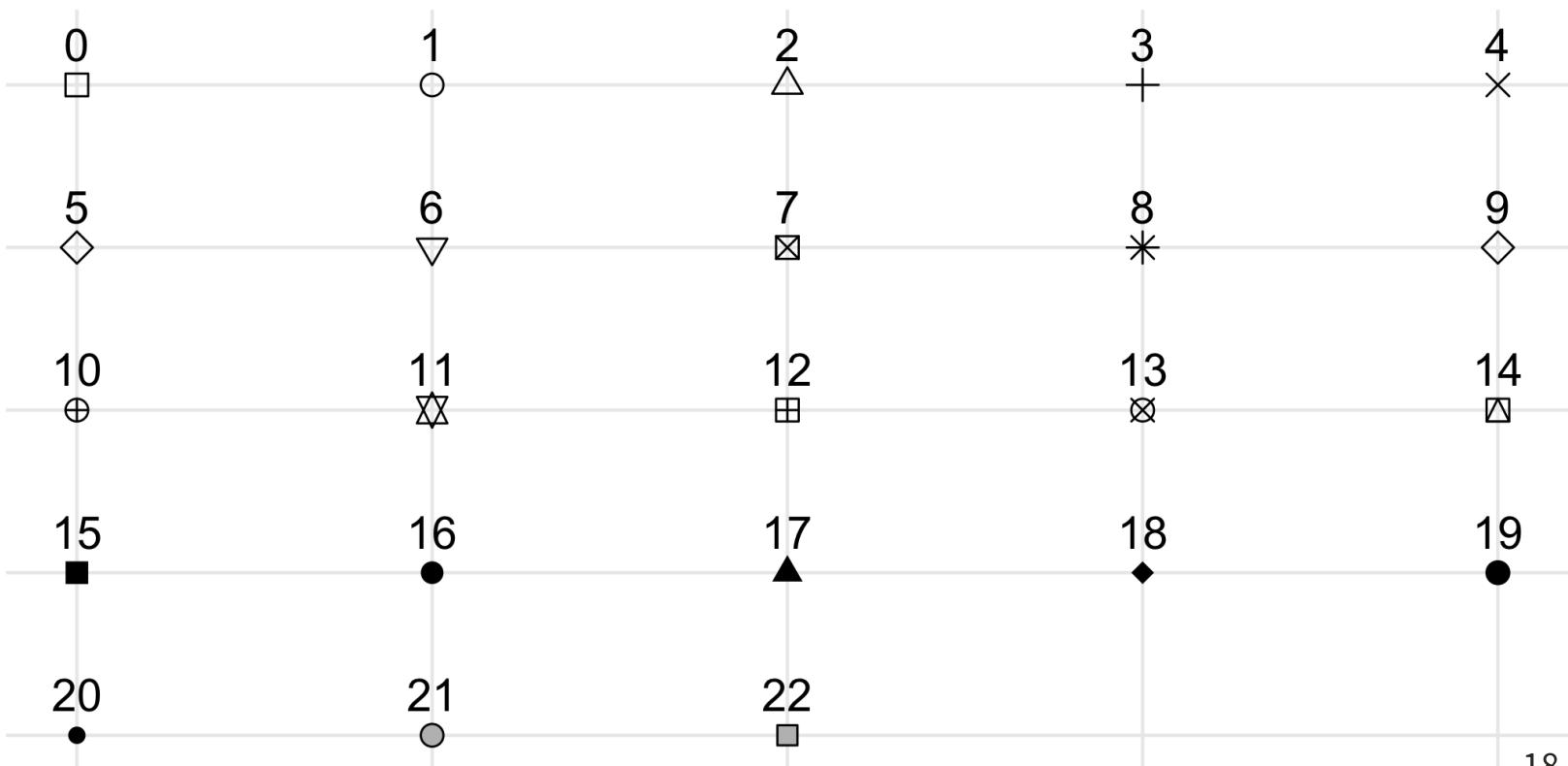
色の見本 (一部)

cornflowerblue	darkslategray	greenyellow	lightgrey	mediumseagreen	palegreen	sienna	
coral	darkslateblue	green	lightgreen	mediumpurple	palegoldenrod	seashell	yellowgreen
chocolate	darkseagreen	gray	lightgray	mediumorchid	orchid	seagreen	yellow
chartreuse	darksalmon	goldenrod	lightgoldenrodyellow	mediumblue	orangered	sandybrown	whitesmoke
cadetblue	darkred	gold	lightgoldenrod	mediumaquamarine	orange	salmon	wheat
burlywood	darkorchid	ghostwhite	lightcyan	maroon	olivedrab	saddlebrown	violetred
brown	darkorange	gainsboro	lightcoral	magenta	oldlace	royalblue	violet
blueviolet	darkolivegreen	forestgreen	lightblue	linen	navyblue	rosybrown	turquoise
blue	darkmagenta	floralwhite	lemonchiffon	limegreen	navy	red	tomato
blanchedalmond	darkkhaki	firebrick	lawngreen	lightyellow	navajowhite	purple	thistle
black	darkgrey	dodgerblue	lavenderblush	lightsteelblue	moccasin	powderblue	tan
bisque	darkgreen	dimgray	lavender	lightslategray	mistyrose	plum	steelblue
beige	darkgray	dimgray	khaki	lightslategray	mintcream	pink	springgreen
azure	darkgoldenrod	deepskyblue	ivory	lightslateblue	midnightblue	peru	snow
aquamarine	darkcyan	deeppink	indianred	lightskyblue	mediumvioletred	peachpuff	slategray
antiquewhite	darkblue	darkviolet	hotpink	lightseagreen	mediumturquoise	papayawhip	slategray
aliceblue	cyan	darkturquoise	honeydew	lightsalmon	mediumspringgreen	palevioletred	slateblue
white	cornsilk	darkslategray	grey	lightpink	mediumslateblue	paleturquoise	skyblue

shape の見本

shape で指定 (デフォルトは shape = 19)

- 黒の部分は color、グレーの部分は fill (または bg) で調整
 - 21と22の場合、枠線は color、内側の色塗りは fill
- 0~14の場合、中身が透明



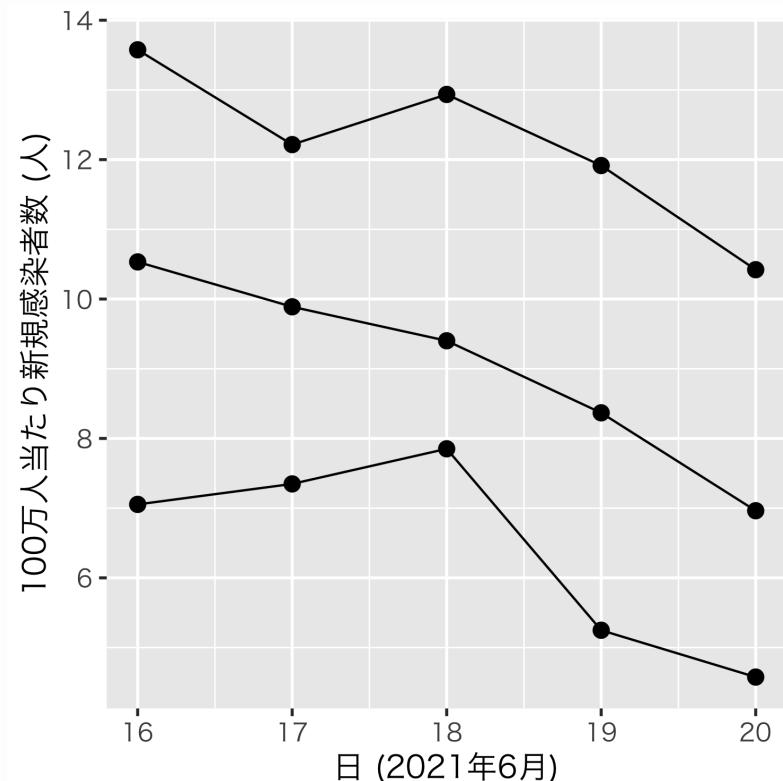
geom_line(): 折れ線グラフ

折れ線グラフの必須要素

折れ線グラフを作成する際に必要な**最低限**の情報

データに日と新規感染者数の**列**が必要

- `x`: 線の傾きが変化し得る点の横軸上の位置
 - 日
- `y`: 線の傾きが変化し得る点の縦軸上の位置
 - 100万人当たり新規感染者数
- 線が2つ以上の場合 `groups` が必要
 - 点をグループ化しないと、どの点を繋げば良いかが分からなくなる



データの用意

授業サポートページから実習用データ（Micro13.csv）をダウンロード

- 整然データとして加工済み
- Day: 日付（2021年6月）；NewCases: 100万人当たりCOVID-19新規感染者数

```
COVID_df <- read_csv("Data/Micro13.csv")
```

```
COVID_df
```

```
## # A tibble: 15 x 3
##   Country Day NewCases
##   <chr>    <dbl>     <dbl>
## 1 Japan      16     13.6
## 2 Japan      17     12.2
## 3 Japan      18     12.9
## 4 Japan      19     11.9
## 5 Japan      20     10.4
## 6 Korea       16     10.5
## 7 Korea       17     9.89
## 8 Korea       18     9.40
## 9 Korea       19     8.37
## 10 Korea      20     6.96
## 11 Taiwan     16     7.05
```

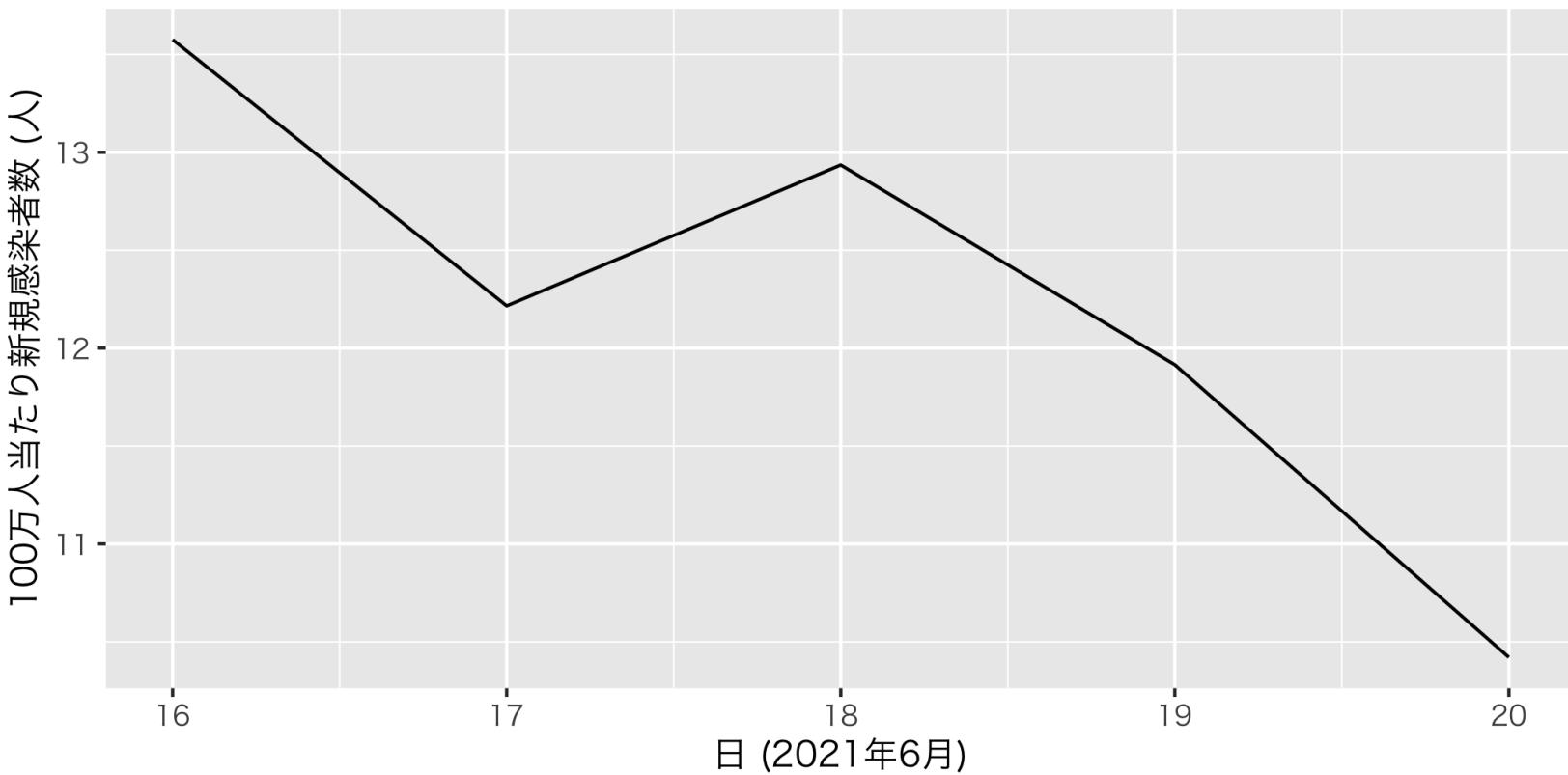
線が一本の場合（コード）

geom_line() 幾何オブジェクトを使用

- `x`: 点の傾きが変化し得る点の横軸上の位置
- `y`: 点の傾きが変化し得る点の縦軸上の位置
- 日本の行のみを抽出し、横軸を日、縦軸を100万人当たり新規感染者数とした折れ線グラフを作成

```
COVID_df %>%
  filter(Country == "Japan") %>%
  ggplot() +
  geom_line(aes(x = Day, y = NewCases)) +
  labs(x = "日 (2021年6月)", y = "100万人当たり新規感染者数 (人)") +
  theme_gray(base_family = "HiraKakuProN-W3")
```

線が一本の場合 (図)



線が二本以上の場合（コード）

group に線のグループ変数を指定

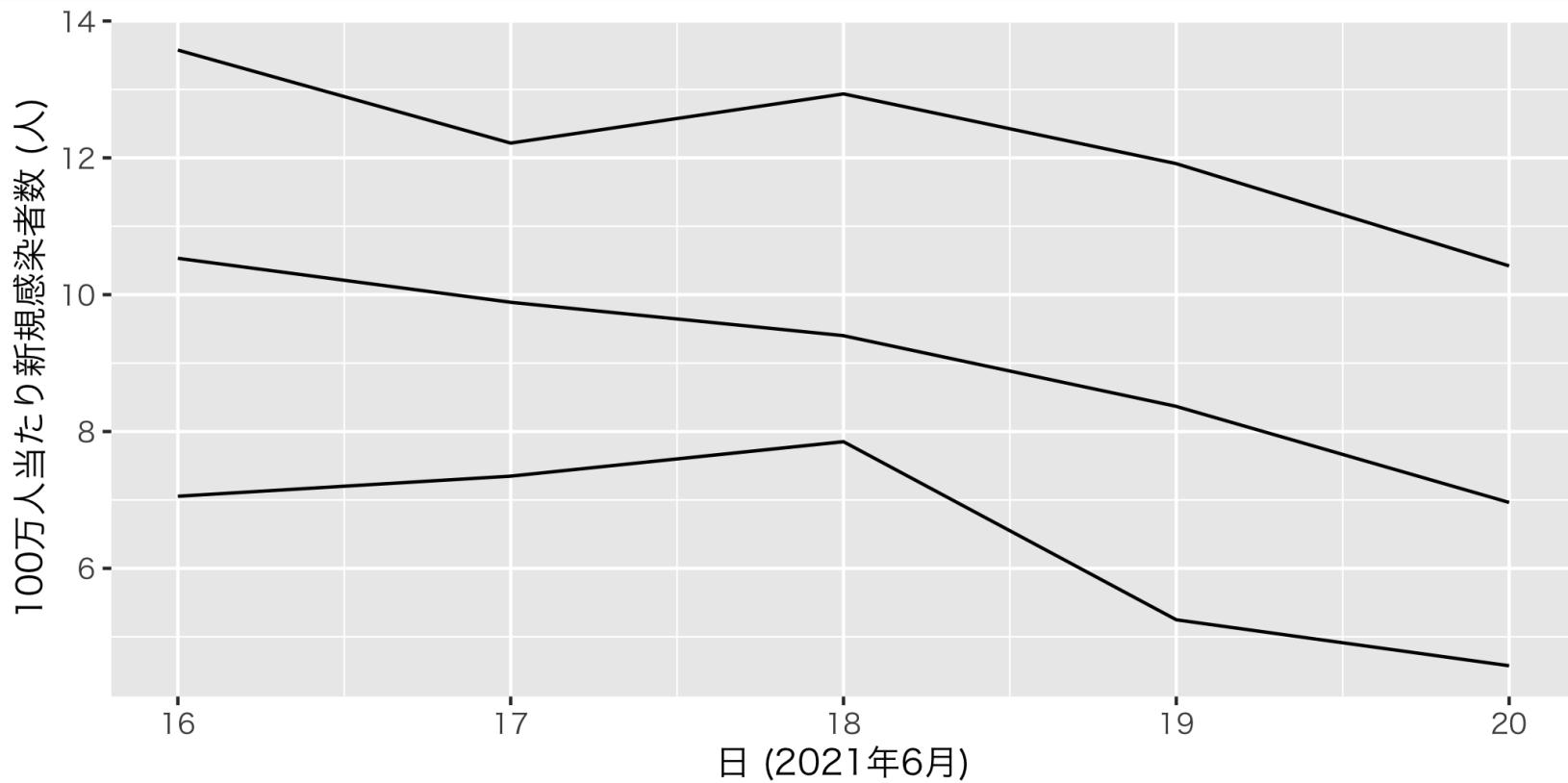
- 国別の折れ線グラフの場合、国変数を指定

```
COVID_df <- COVID_df %>%
  mutate(Country = case_when(Country == "Japan" ~ "日本",
                             Country == "Korea" ~ "韓国",
                             TRUE ~ "台湾"),
         Country = factor(Country, levels = c("日本", "韓国", "台灣")))
COVID_df %>%
  ggplot() +
  geom_line(aes(x = Day, y = NewCases, group = Country)) +
  labs(x = "日 (2021年6月)", y = "100万人当たり新規感染者数 (人)") +
  theme_gray(base_family = "HiraKakuProN-W3")
```

線が二本以上の場合（図）

各線がどの国を示すのかが分からぬ...

- 色分けで解決



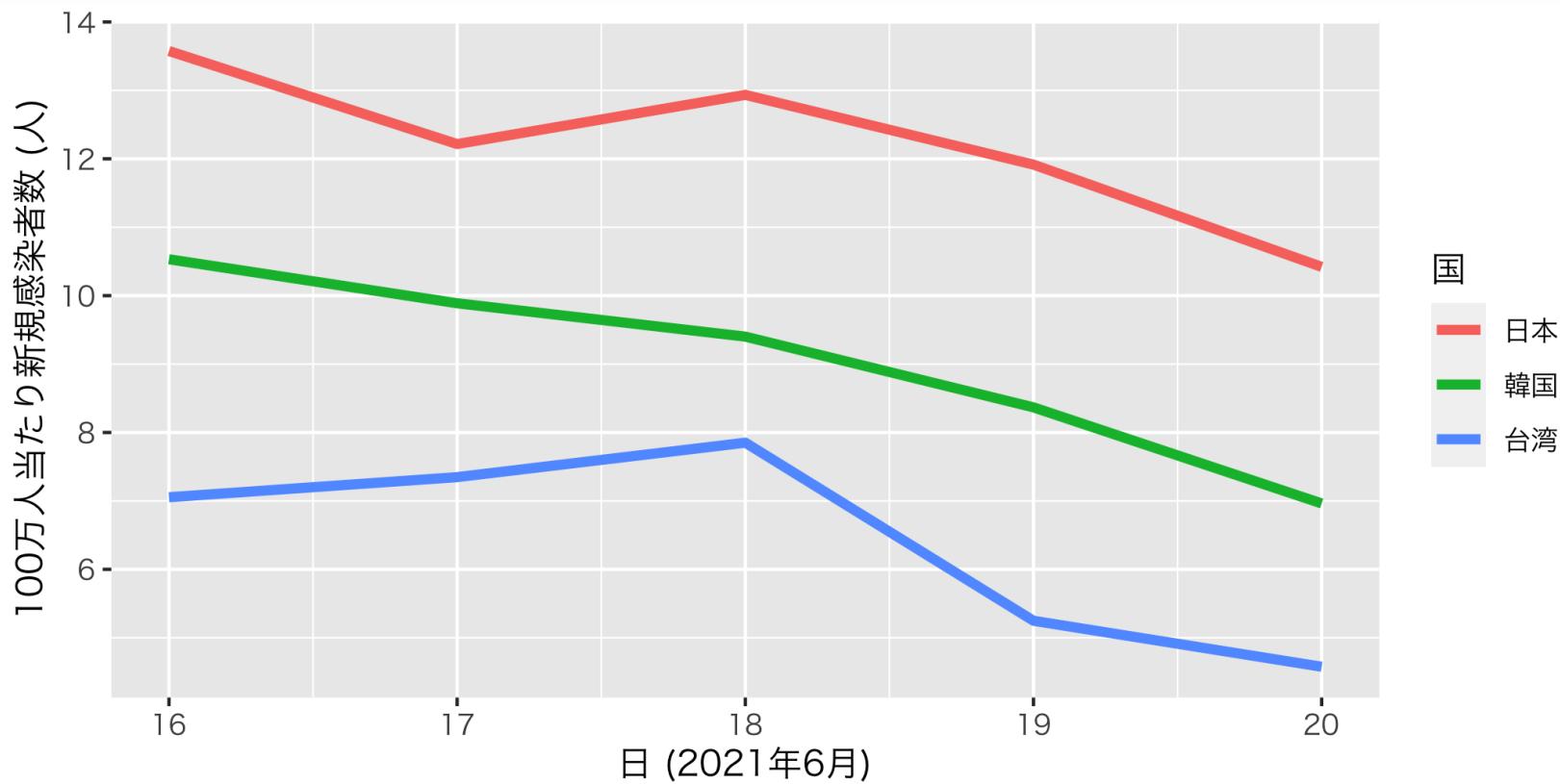
線が二本以上の場合（コード）

`color` で色分け: 国別の折れ線グラフの場合、国変数を指定

- 線をやや太め (1.5) にする
 - `size`: 全線に適用させるために `aes()` の外側に
 - 線が細すぎると色が区別しにくい

```
COVID_df %>%
  ggplot() +
  geom_line(aes(x = Day, y = NewCases, group = Country,
                color = Country), size = 1.5) +
  labs(x = "日 (2021年6月)", y = "100万人当たり新規感染者数 (人)",
       color = "国") +
  theme_gray(base_family = "HiraKakuProN-W3")
```

線が二本以上の場合（図）



次元の追加

多くの折れ線グラフは既に3次元

- 日 (x) 、新規感染者数 (y) 、国 (group or color)
- これ以上次元を増やすならファセット分割で対応するのが賢明
- ファセット分割せずに次元を増やすなら...
 - size: 線の太さ
 - linetype: 線のタイプ (1 = 実線、2 = 破線、4 = 点線、...)
 - 白黒図の場合、color でなく、linetype
 - alpha: 透明度

線のタイプ（コード）

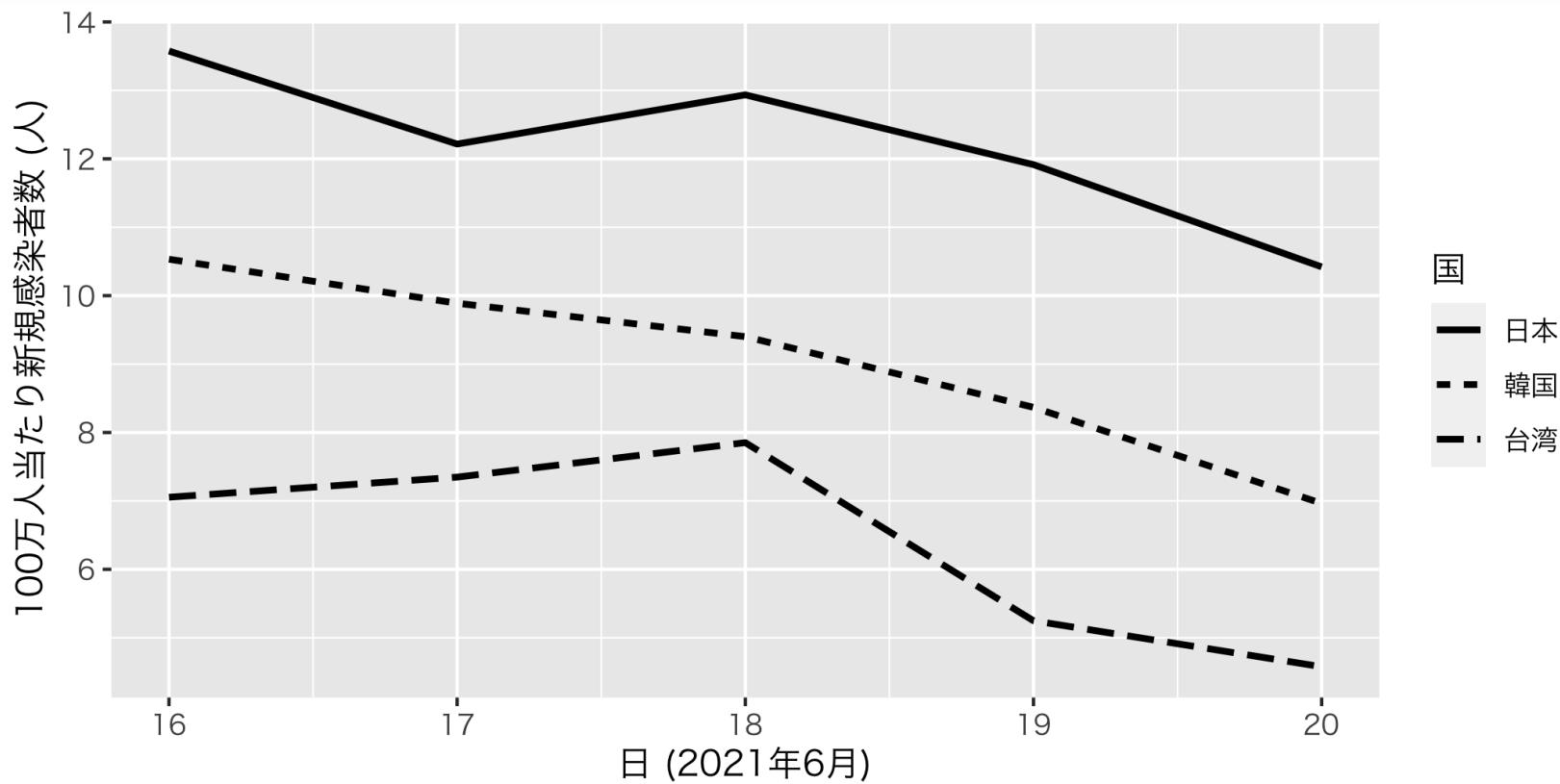
白黒図の場合

```
COVID_df %>%
  ggplot() +
  geom_line(aes(x = Day, y = NewCases, group = Country,
                linetype = Country), size = 1) +
  labs(x = "日 (2021年6月)", y = "100万人当たり新規感染者数 (人)",
       linetype = "国") +
  theme_gray(base_family = "HiraKakuProN-W3")
```

自分で線のタイプを指定したい場合は

```
データ %>%
  ggplot() +
  ...
  scale_linetype_manual(values = c("日本" = 1, "韓国" = 2, "台湾" = 4))
```

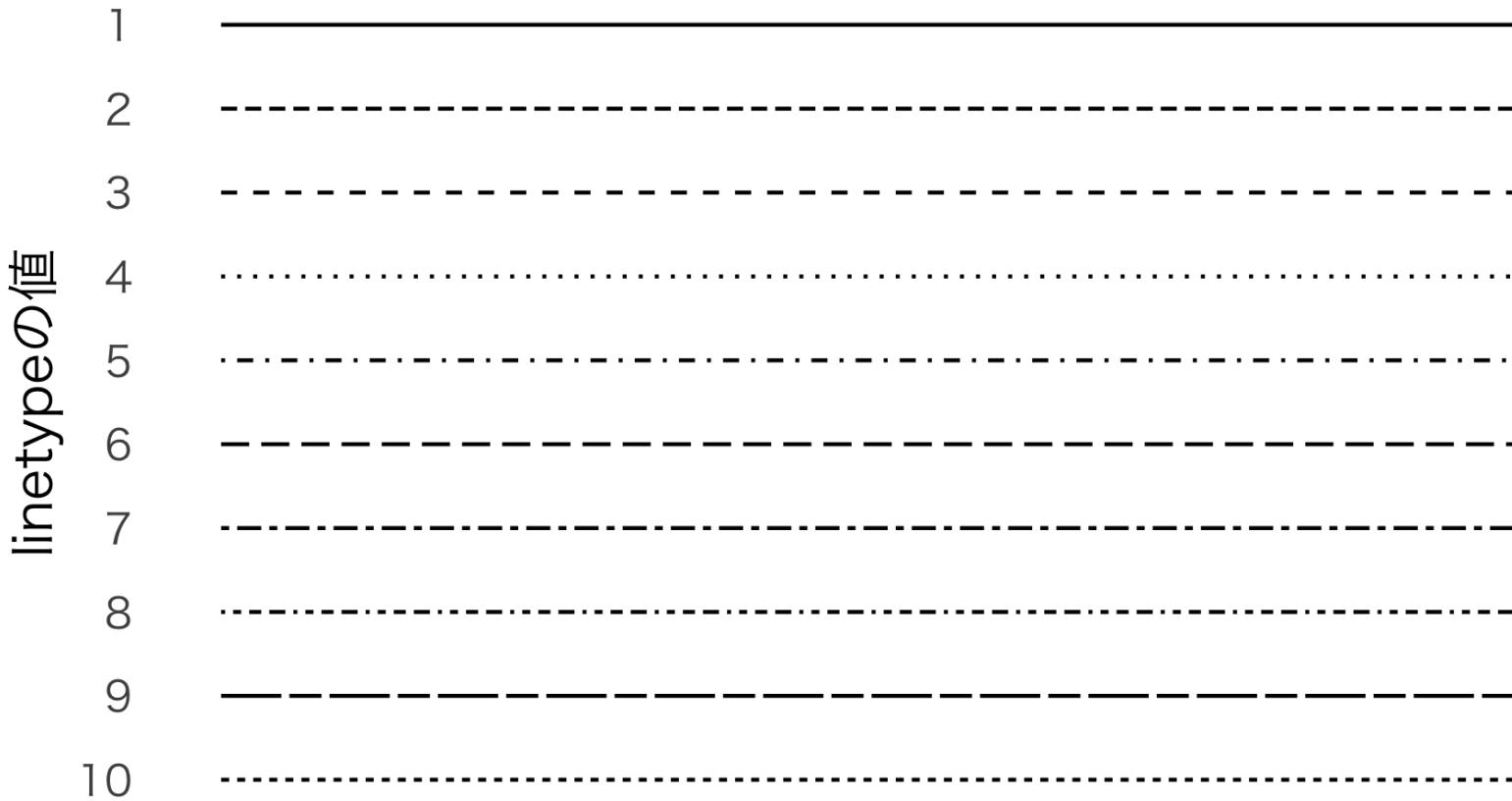
線のタイプ (図)



linetypeの見本

種類が多くなると区別しにくいため、最大3つ程度

- 1, (2 or 3), (4 or 10)
- 一つ面に登場する線は3～4本程度まで（色分けも同様）



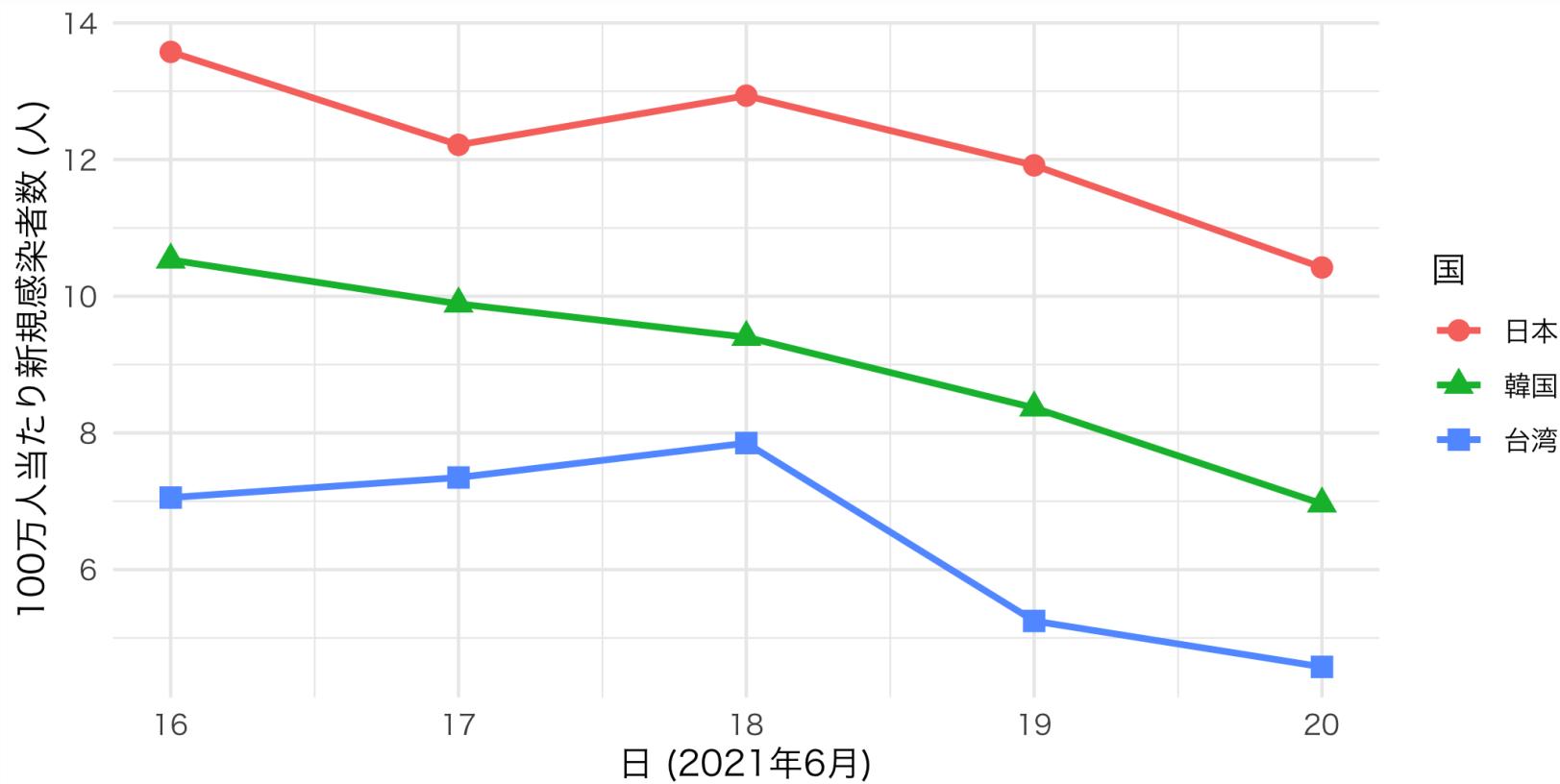
折れ線グラフ + 散布図（コード）

geom_point() と geom_line() を重ねる

- 後（先）に書いたレイヤーが前面（背面）に
- 傾きの変化が激しくない場合に有効
- 点は線より大きめに
- 白黒印刷に備え、点の形分けも可能

```
COVID_df %>%  
  ggplot(aes(x = Day, y = NewCases, color = Country)) +  
  geom_point(aes(shape = Country), size = 3) +  
  geom_line(aes(group = Country), size = 1) +  
  labs(x = "日 (2021年6月)", y = "100万人当たり新規感染者数 (人)",  
       color = "国", shape = "国") +  
  theme_minimal(base_family = "HiraKakuProN-W3")
```

折れ線グラフ + 散布図（図）



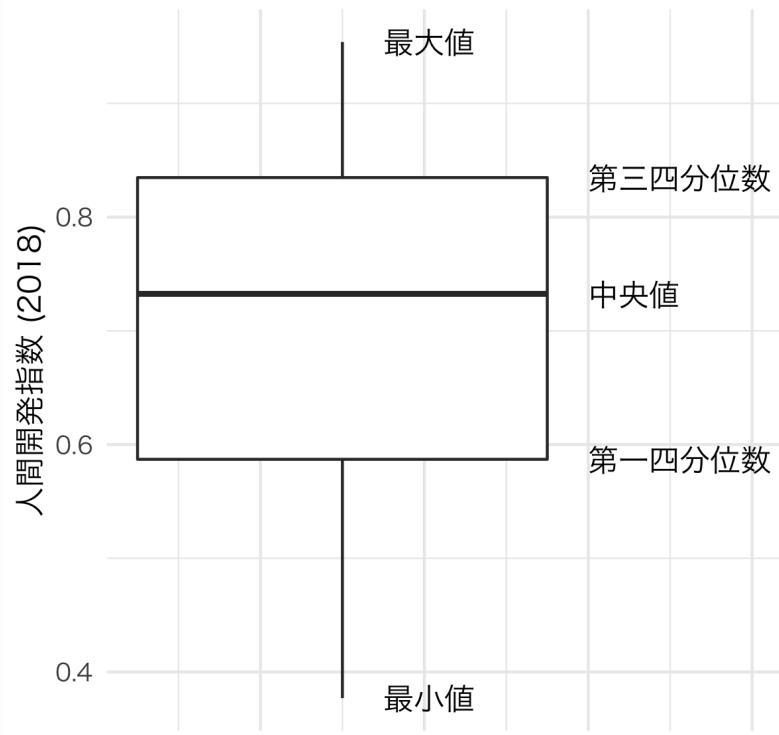
geom_boxplot(): 箱ひげ図

箱ひげ図とは

変数の分布を示す方法の一つ

- 最小値、最大値
 - ひげの両端
- 第一四分位数、第三四分位数
 - 箱の上限と下限
- 中央値（第二四分位数）
 - 箱内の線
- 外れ値がある場合、点

変数の分布を **グループごとに** 見る時に有効



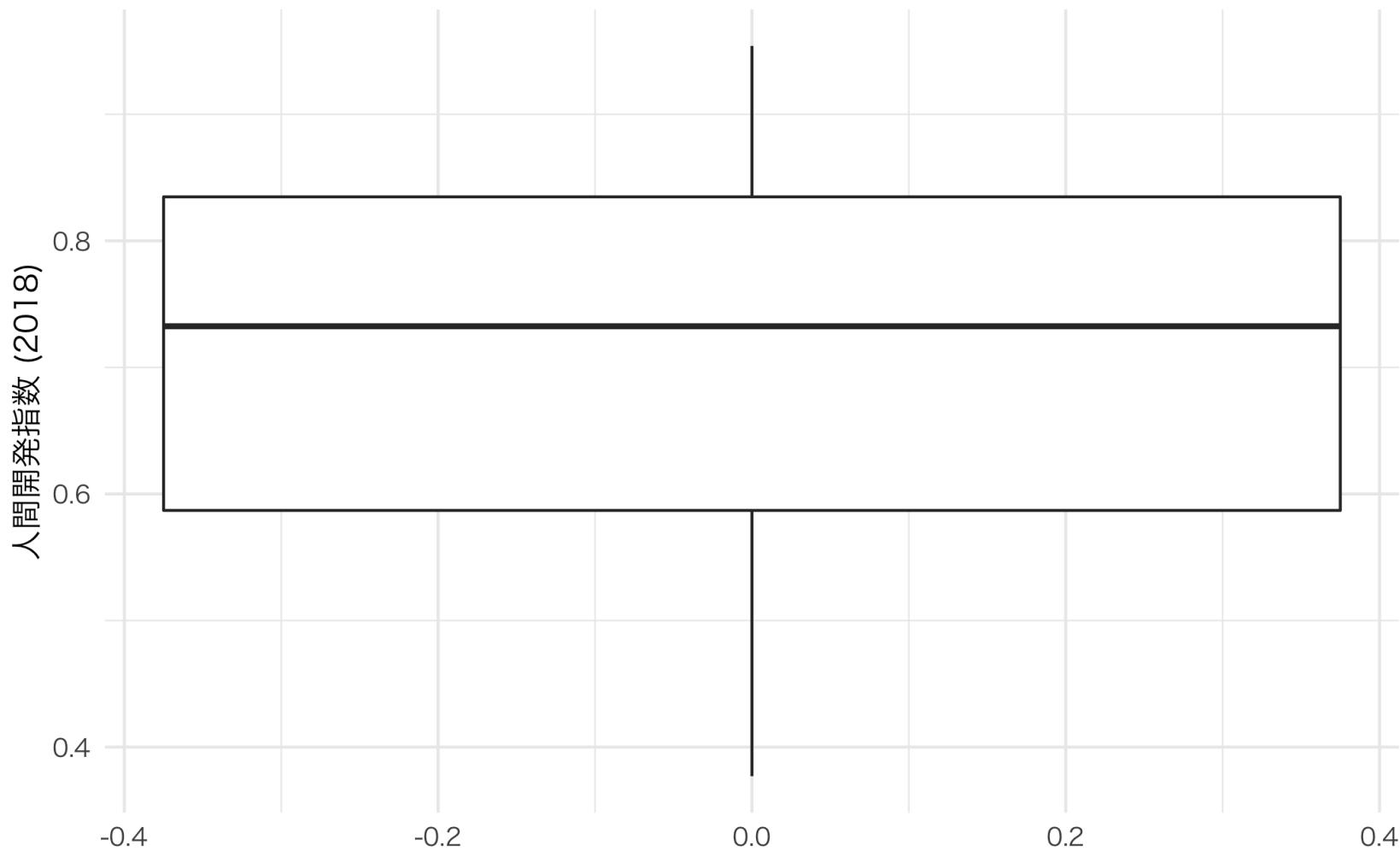
一変数の箱ひげ図（コード）

人間開発指数 (HDI_2018) の箱ひげ図

- x か y に分布を確認した変数をマッピングする
 - x にマッピングするか、 y にするかによって箱の向きが変わる

```
df %>%
  ggplot() +
  geom_boxplot(aes(y = HDI_2018)) +
  labs(x = "", y = "人間開発指數 (2018)") +
  theme_minimal(base_family = "HiraKakuProN-W3")
```

一変数の箱ひげ図（図）



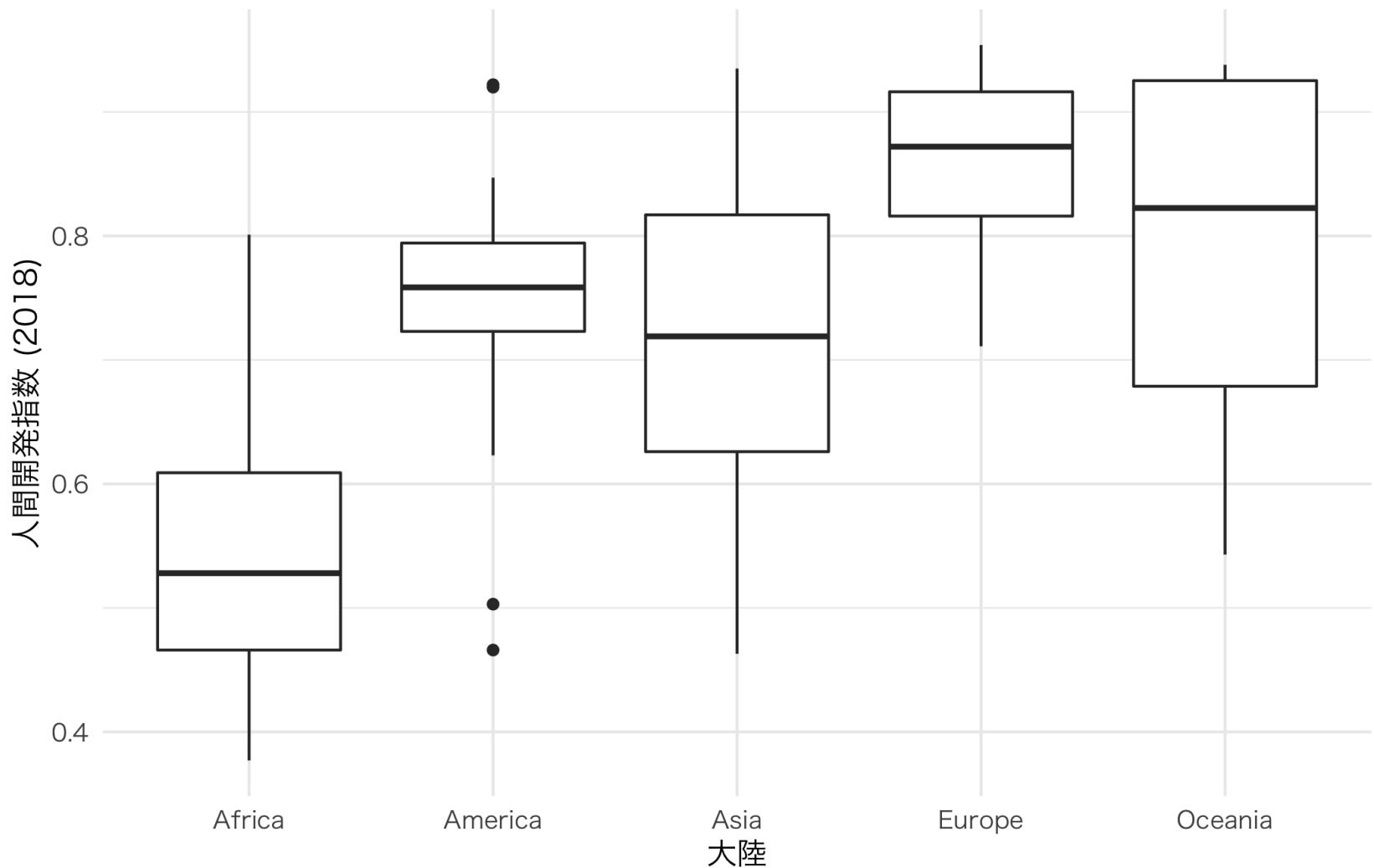
次元の追加（コード）

yにしかマッピングされているため、まだ次元追加の余地が残っている

- xにマッピング（大陸ごとのHDI_2018の箱ひげ図）
- 更に次元を追加したい場合は、ファセット分割

```
df %>%
  ggplot() +
  geom_boxplot(aes(y = HDI_2018, x = Continent)) +
  labs(x = "大陸", y = "人間開発指数 (2018)") +
  theme_minimal(base_family = "HiraKakuProN-W3")
```

次元の追加 (図)



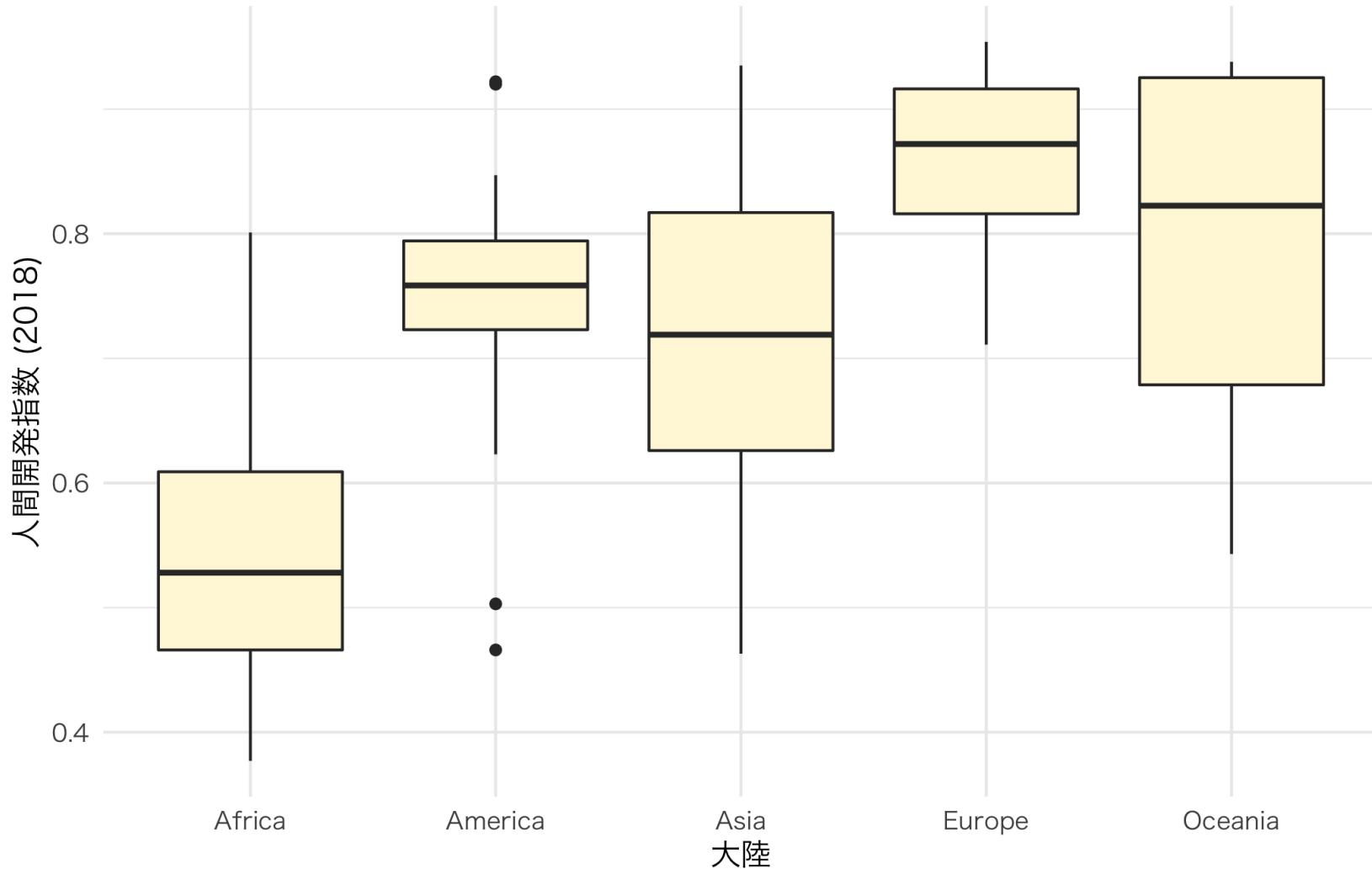
カスタマイズ（色）

箱の色を変える

- すべての色を変える場合、`aes()` の外側に `fill`
- 箱ごとに色を変える場合、`aes()` の内側に `fill`

```
df %>%
  ggplot() +
  geom_boxplot(aes(y = HDI_2018, x = Continent), fill = "cornsilk") +
  labs(x = "大陸", y = "人間開発指数 (2018)") +
  theme_minimal(base_family = "HiraKakuProN-W3")
```

カスタマイズ (色)



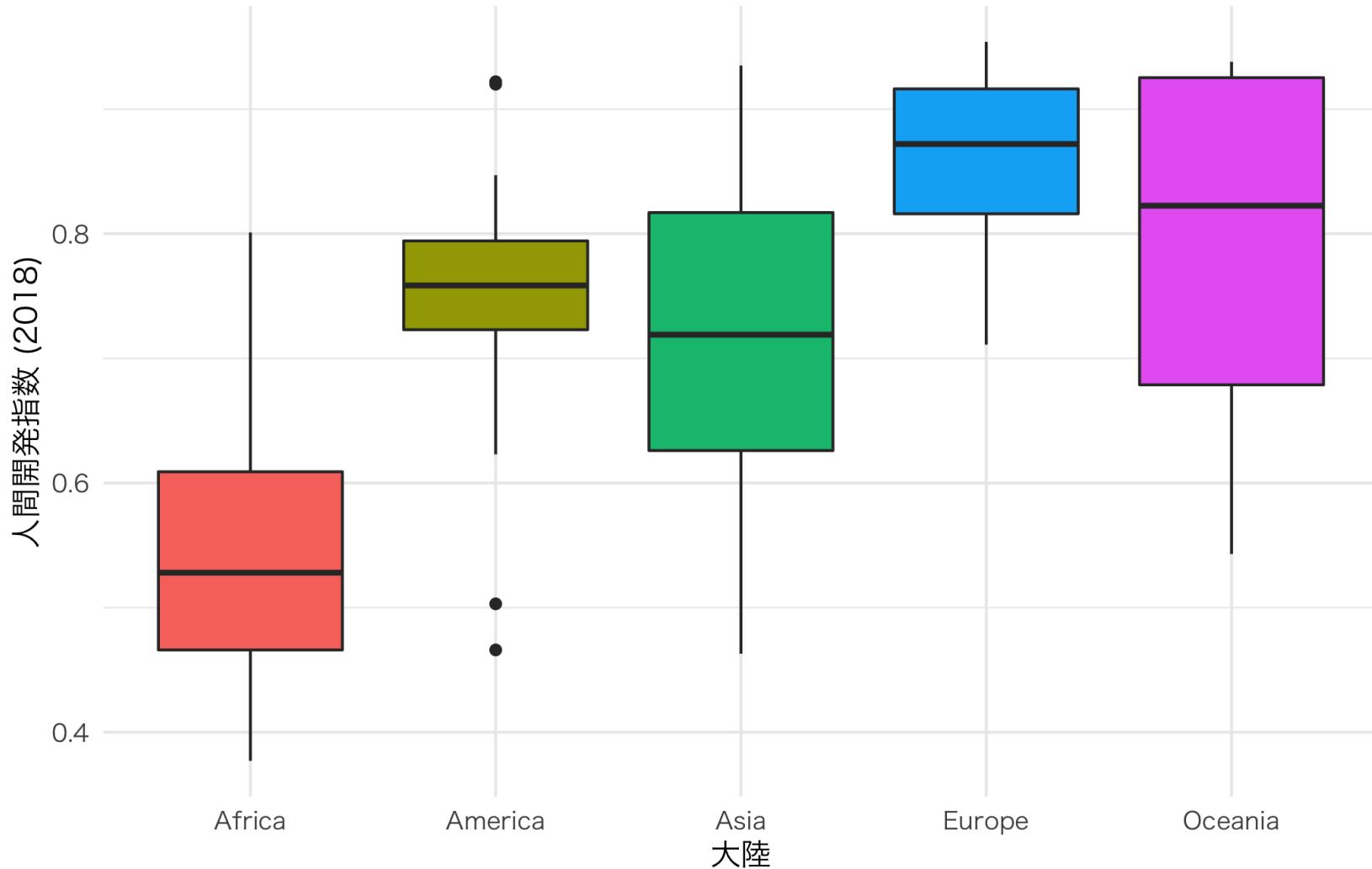
カスタマイズ（色）

箱の色を変える

- すべての色を変える場合、`aes()` の外側に `fill`
- 箱ごとに色を変える場合、`aes()` の内側に `fill`

```
df %>%
  ggplot() +
  geom_boxplot(aes(y = HDI_2018, x = Continent, fill = Continent)) +
  labs(x = "大陸", y = "人間開発指数 (2018)") +
  theme_minimal(base_family = "HiraKakuProN-W3") +
  theme(legend.position = "none")
```

カスタマイズ (色)



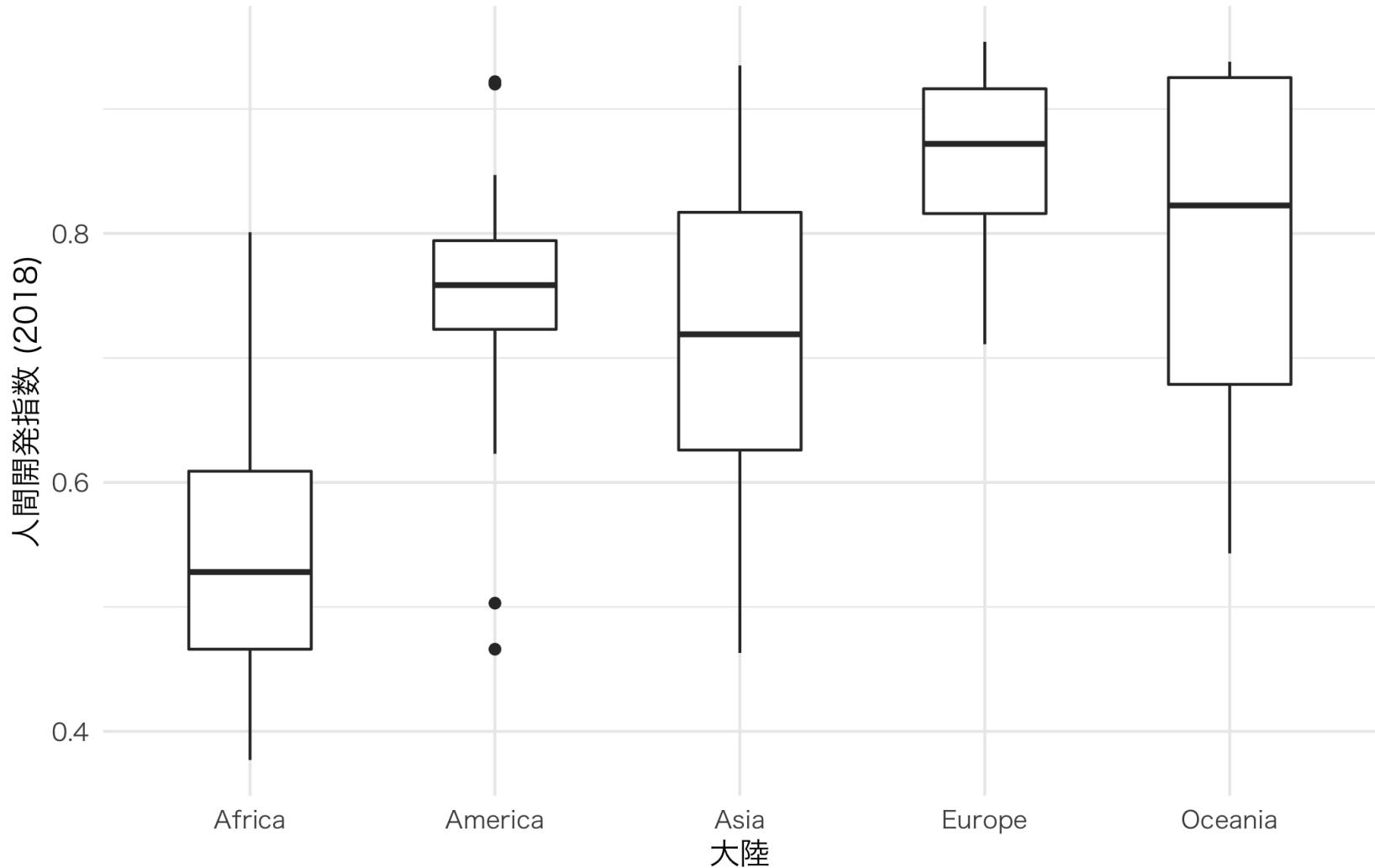
カスタマイズ（大きさ）

箱の幅を変える

- `aes()` の外側に `width` を指定

```
df %>%
  ggplot() +
  geom_boxplot(aes(y = HDI_2018, x = Continent), width = 0.5) +
  labs(x = "大陸", y = "人間開発指数（2018）") +
  theme_minimal(base_family = "HiraKakuProN-W3") +
  theme(legend.position = "none")
```

カスタマイズ (大きさ)

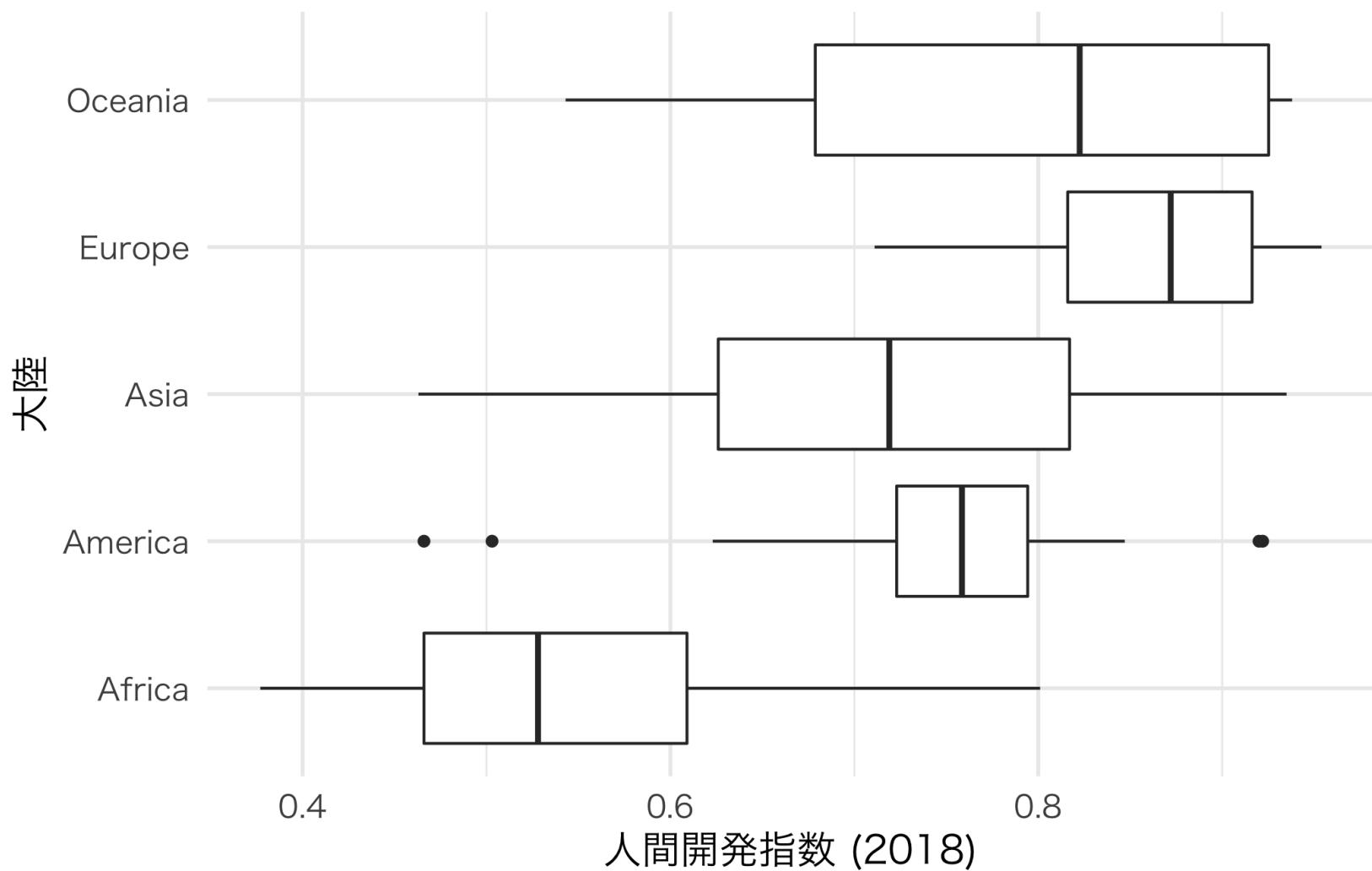


横軸と縦軸の交換（コード）

箱が多すぎて読みにくい場合、`x` と `y` を交換もOK

```
df %>%
  ggplot() +
  geom_boxplot(aes(x = HDI_2018, y = Continent)) +
  labs(x = "人間開発指数 (2018)", y = "大陸") +
  theme_minimal(base_family = "HiraKakuProN-W3",
                base_size    = 14) +
  theme(legend.position = "none")
```

横軸と縦軸の交換（図）



まとめ

今回の内容

よく分からぬ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)

- 散布図の作成: [教科書第18.6章](#)
- 折れ線グラフの作成: [教科書第18.7章](#)
- 箱ひげ図: [教科書第18.5章](#)
- グラフのカスタマイズ: [教科書第19章](#)

その他のグラフ

- [教科書第20章 \(執筆中\)](#)
- [The R Graph Gallery](#)

課題

1. 今回講義用のプロジェクトを作成する。
2. LMSから**2つ**のデータ (.csv) 問題ファイル (.Rmd) とサンプルファイル (.html) をダウンロードし、プロジェクトのフォルダーに保存する。
 - **ファイル名は変更しないこと**
3. プロジェクトを開き、.Rmd ファイルを開く
4. サンプルファイルと同じ結果が得られるようにR Markdown文書を作成する。
5. 隨時Knitし、結果を確認する。
 - Knitできないファイルは評価の対象外
6. .html ファイルを関大LMSに提出する。
 - **注意!** .Rmd ファイルでなく、Knit後の.html ファイルを提出
7. **期限は2021年7月10日（土）の23時59分とする。**
 - 時間に余裕を持って取り組むこと。期限直前に取り組み始めてPCトラブルがあっても期限延長はない
8. 答案は次回の講義までに公開する。