

# 情報処理応用演習

## 第3講: 可視化

SONG Jaehyun (同志社大学)

2020/8/25

# ggplot2の仕組み

# 本日の内容

- グラフィックの文法
- 代表的な5つのグラフ
  1. 棒グラフ
  2. ヒストグラム
  3. 箱ひげ図
  4. 散布図
  5. 折れ線グラフ
- 期末課題について

# ggplot2とは

グラフィックの文法 (grammar of graphics) をRで実装したパッケージ

- グラフをいくつかの要素に分解し、それぞれを一つのレイヤー (layer; 層) として捉える
- レイヤーを重ねることでグラフを作成する

#コードの例

```
library(tidyverse)
Country_df <- read_csv("Data/Countries.csv")
Country_df %>%
  mutate(OECD2 = ifelse(OECD == 1, "加盟国", "非加盟国")) %>%
  ggplot() +
  geom_point(aes(x = PPP_per_capita, y = HDI_2018, color = OECD2),
             size = 2, alpha = 0.5) +
  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_minimal(base_family = "HiraKakuProN-W3") +
  theme(legend.position = "bottom")
```

# 例) データの用意

```
library(tidyverse)
Country_df <- read_csv("Data/Countries.csv")

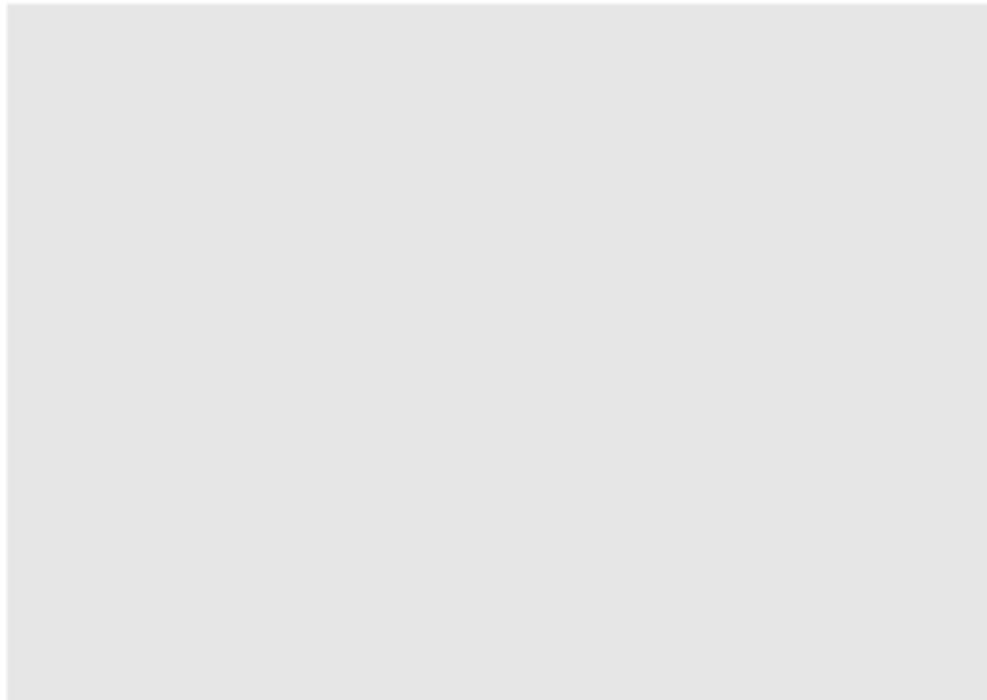
Country_df2 <- Country_df %>%
  mutate(OECD2 = ifelse(OECD == 1, "加盟国", "非加盟国")) %>%
  select(Country, PPP = PPP_per_capita, HDI = HDI_2018, OECD = OECD2)

Country_df2
```

```
## # A tibble: 186 x 4
##   Country           PPP     HDI OECD
##   <chr>        <dbl>  <dbl> <chr>
## 1 Afghanistan    2125.  0.496 非加盟国
## 2 Albania        13781. 0.791 非加盟国
## 3 Algeria         11324. 0.759 非加盟国
## 4 Andorra          NA    0.857 非加盟国
## 5 Angola          6649.  0.574 非加盟国
## 6 Antigua and Barbuda 21267. 0.776 非加盟国
## 7 Argentina       22938. 0.83  非加盟国
## 8 Armenia          12974. 0.76  非加盟国
## 9 Australia        50001. 0.938 加盟国
```

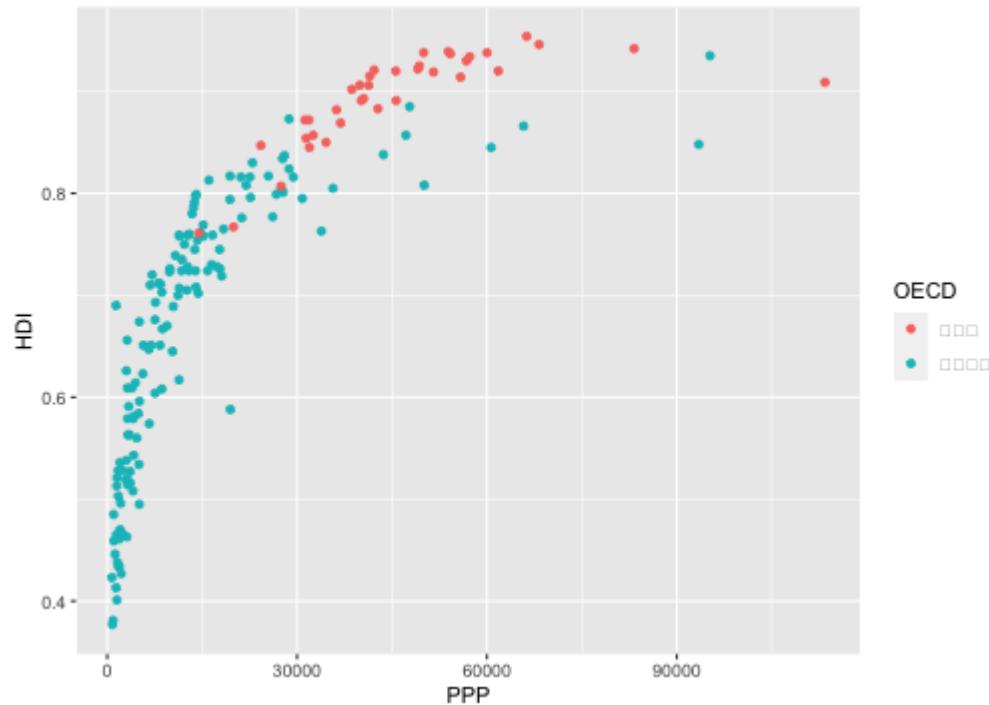
# 例) キャンバスの用意

```
ggplot(data = Country_df2)
```



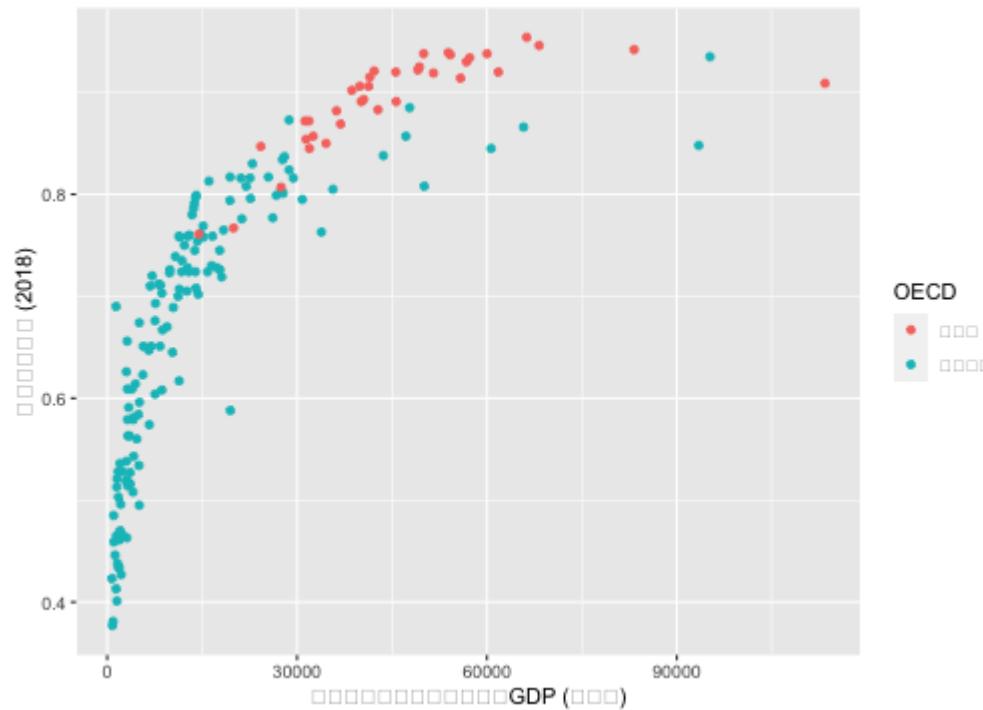
# 例) 散布図の出力

```
ggplot(data = Country_df2) +  
  # x軸はPPP変数、y軸はHDI変数とし、OECDの値ごとに色分け  
  geom_point(aes(x = PPP, y = HDI, color = OECD))
```



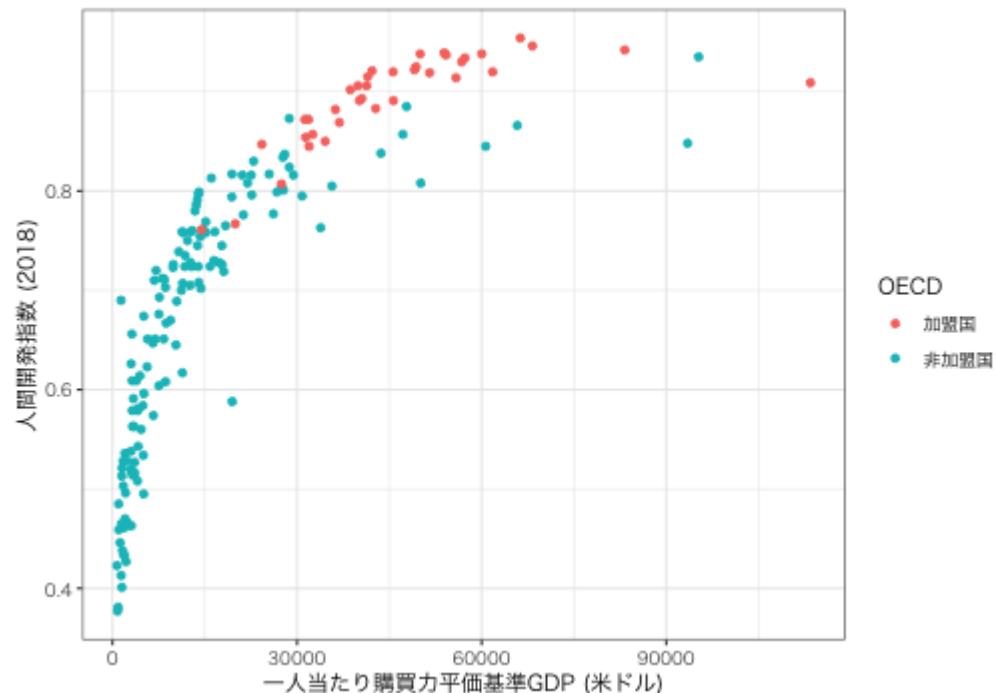
# 例) ラベルの修正

```
ggplot(data = Country_df2) +  
  geom_point(aes(x = PPP, y = HDI, color = OECD)) +  
  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")
```



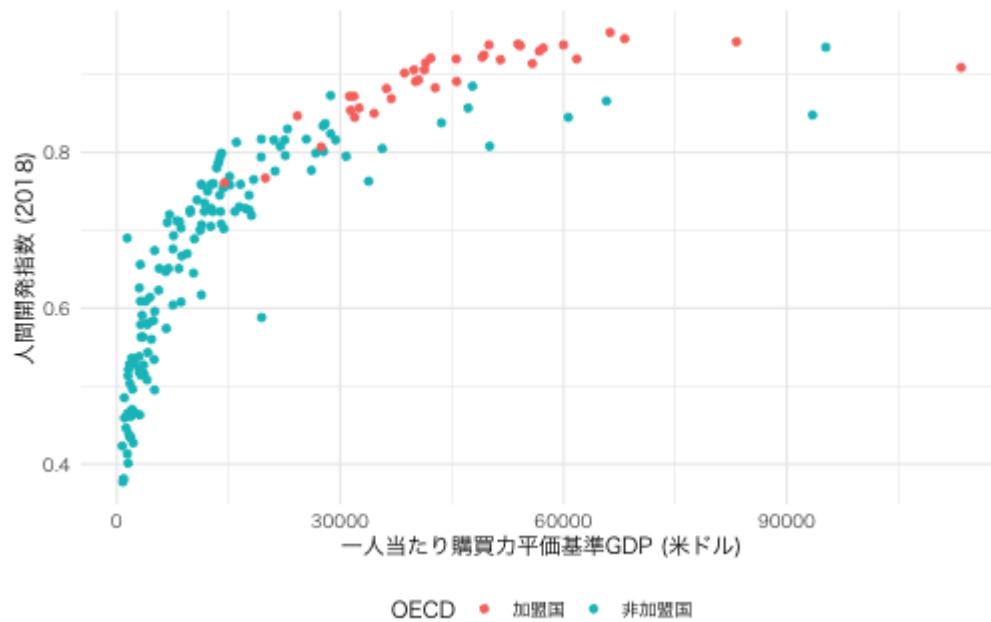
# 例) グラフのテーマ選択

```
ggplot(data = Country_df2) +  
  geom_point(aes(x = PPP, y = HDI, color = OECD)) +  
  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") # フォント指定はmacOSの場合
```



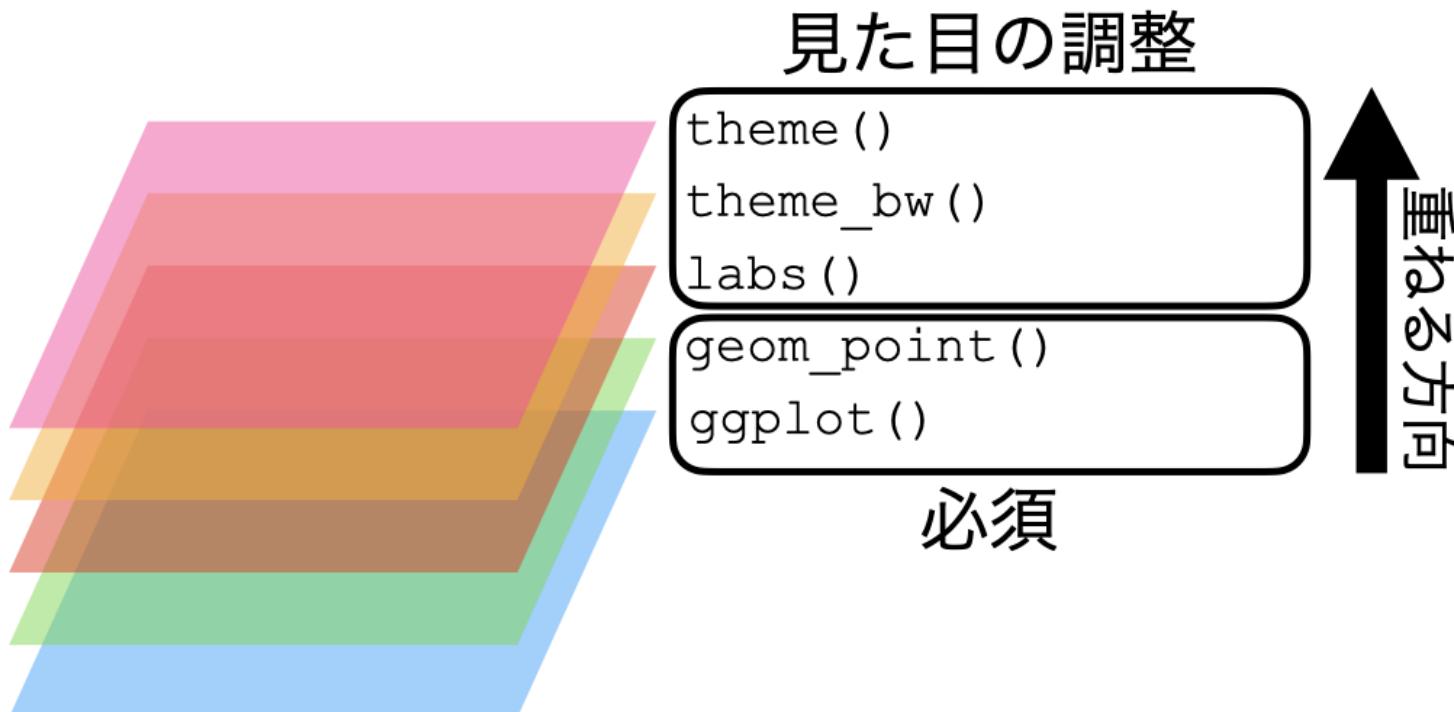
# 例) 凡例の位置調整

```
ggplot(data = Country_df2) +  
  geom_point(aes(x = PPP, y = HDI, color = OECD)) +  
  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_minimal(base_family = "HiraKakuProN-W3") +  
  theme(legend.position = "bottom")
```



# ggplot2の図が出来上がるまで

- ・レイヤーを重ねるイメージ
  - 図の核心部は幾何オブジェクト (`geom_*`()) とマッピング (`aes()`)



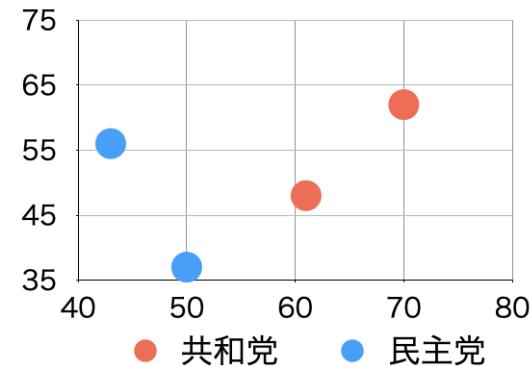
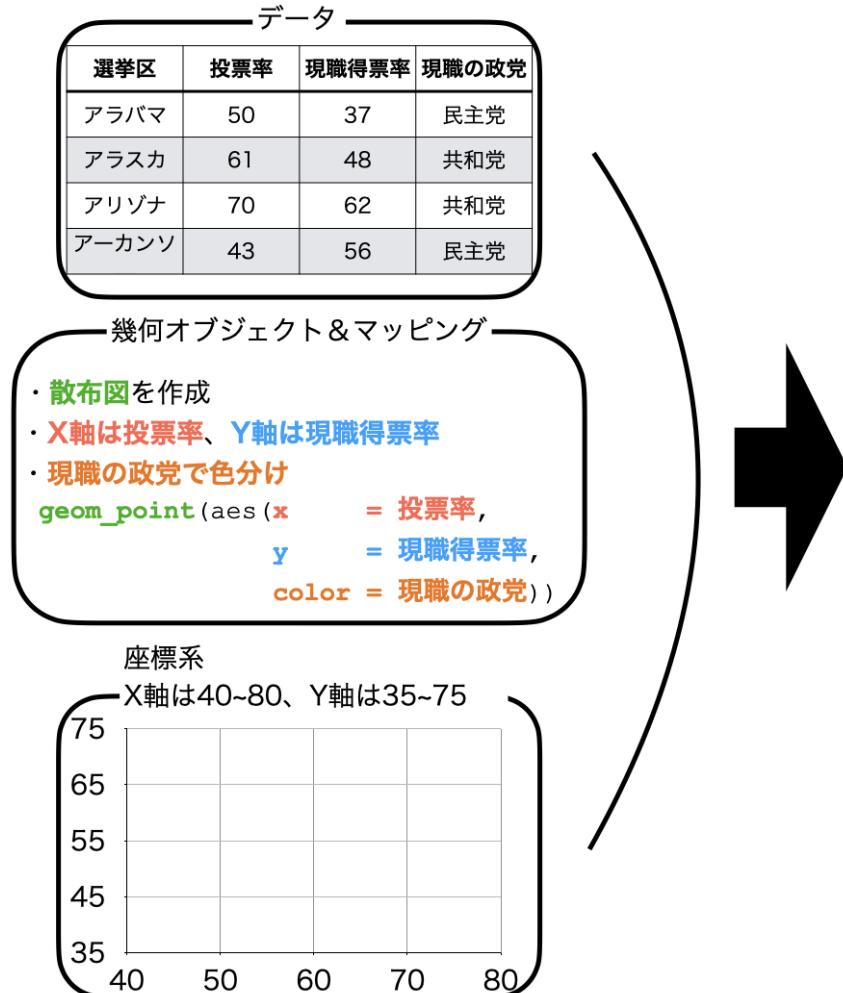
# ggplot2の必須要素

以下の要素があればグラフは出来上がる

1. データ (Data)
2. 幾何オブジェクト (Geometry Object)
  - 散布図、棒グラフ、折れ線グラフ、…
3. マッピング (Mapping)
  - 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)
  - 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



5NG

# 5 Named Graphs (5NG)

1. 棒グラフ: `geom_bar()`
  - 必須マッピング1: 棒の横軸上の位置 (`x`)
  - 必須マッピング2: 棒の横軸上の位置 (`x`) と棒の高さ (`y`)
2. ヒストグラム: `geom_histogram()`
  - 必須マッピング: 分布が知りたい変数 (`x`)
3. 箱ひげ図: `geom_boxplot()`
  - 必須マッピング: 分布が知りたい変数 (`y`)
4. 散布図: `geom_point()`
  - 必須マッピング: 点の横軸上の位置 (`x`) と縦軸上の位置 (`y`)
5. 折れ線グラフ: `geom_line()`
  - 必須マッピング: 線の傾きが変化する点の横軸上の位置 (`x`) と縦軸上の位置 (`y`)

# 目的別グラフ

一変数の分布が知りたい

1. 棒グラフ: 離散変数
2. ヒストグラム: 連続変数
3. 箱ひげ図: 連続変数
  - ヒストグラムより情報量は少ないが、複数のグループ間の分布を素早く比較可能

二変数間の関係が知りたい

1. 棒グラフ: 離散変数と記述統計量
2. 散布図: 連続変数と連続変数
3. 折れ線グラフ: 順序付き離散変数（日付など）と連続変数
4. 離散変数同士はクロス集計表、またはモザイクプロット

# 実習用データ

- Countries.csv: 世界諸国の社会経済状況および民主主義指標
- COVID19\_Asia2.csv:

```
library(tidyverse)
Country_df <- read_csv("Data/Countries.csv")
```

```
## Parsed with column specification:
## cols(
##   Country = col_character(),
##   Population = col_double(),
##   Area = col_double(),
##   GDP = col_double(),
##   PPP = col_double(),
##   GDP_per_capita = col_double(),
##   PPP_per_capita = col_double(),
##   G7 = col_double(),
##   G20 = col_double(),
##   OECD = col_double(),
##   HDI_2018 = col_double(),
##   Polity_Score = col_double(),
##   Polity_Type = col_character(),
```

# 棒グラフ

# 2つの棒グラフ

## 1. 一変数の分布が知りたい場合

- 度数の棒グラフ
- ある変数において各値はいくつ存在するか
- 例) Country\_dfのContinent列においてAsia、Africa、...はそれぞれ  
いくつあるか

## 2. 二変数の関係が知りたい場合

- 記述統計の棒グラフ
- グループごとの記述統計量はいかに異なるか
- 例) 大陸ごとの人間開発指数の平均値は?

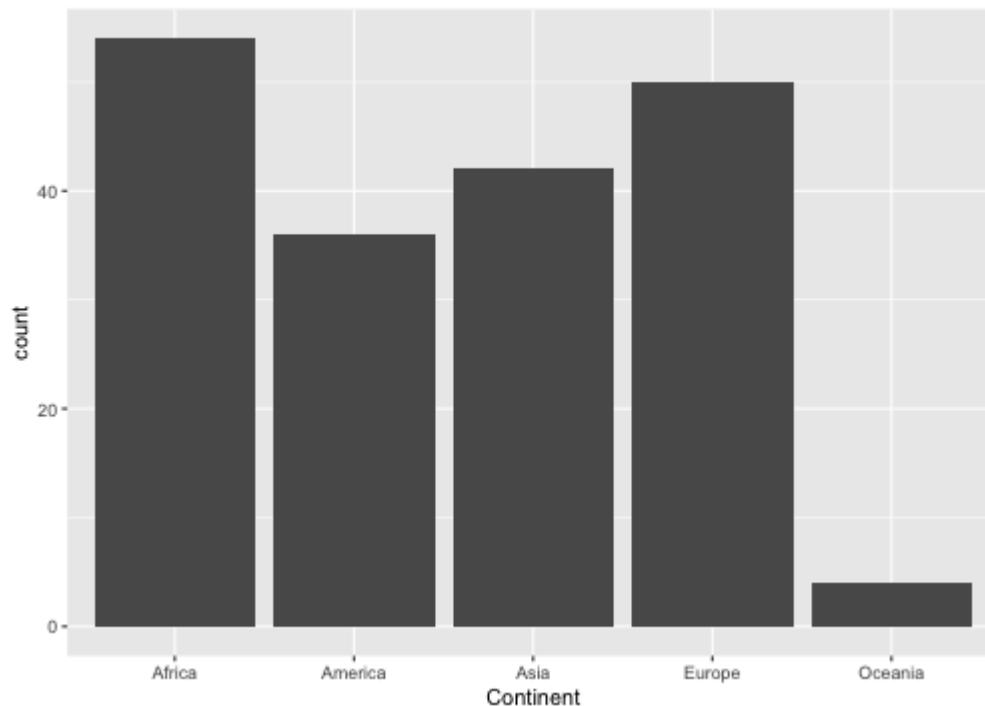
# 度数の棒グラフ

- 幾何オブジェクト: `geom_bar()`
- 必須マッピング: `x = 度数を計算する変数`
- `ggplot()`までは`%>%`を使用し、以降は`+`でレイヤーを追加する

```
データ %>%
  ggplot() +
  geom_bar(aes(x = 変数名))
```

# 度数の棒グラフ

```
Barplot1 <- Country_df %>%
  ggplot() +
  geom_bar(aes(x = Continent))
Barplot1
```



# 見た目の調整

ラベルの修正: `labs()`

```
labs(x = "横軸のラベル", y = "縦軸のラベル")
```

テーマの変更: `theme_*`()

- テーマは`theme_gray()`、`theme_bw()`、`theme_minimal()`など
- macOS、かつ日本語を使う場合は`base_family =`指定

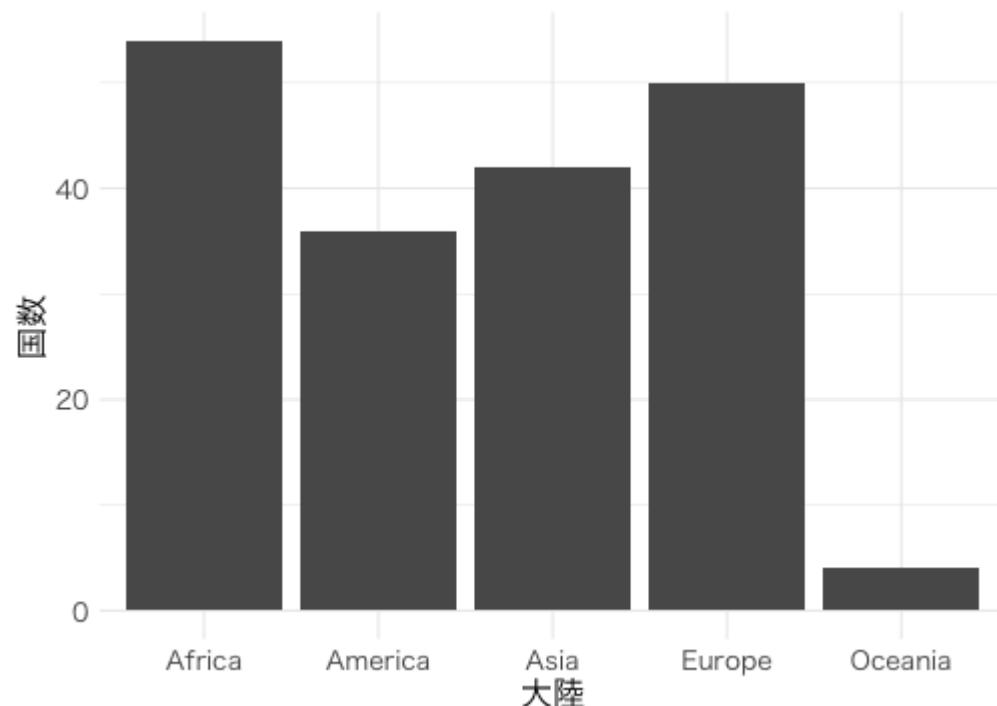
```
theme_minimal(base_family = "HiraKakuProN-W3")
```

文字サイズの変更: `theme()`の`text`引数

```
theme(text = element_text(size = 16))
```

# 度数の棒グラフ

```
Barplot1 +  
  labs(x = "大陸", y = "国数") +  
  theme_minimal(base_family = "HiraKakuProN-W3") +  
  theme(text = element_text(size = 16))
```



# 記述統計の棒グラフ

- 予めグループごとの記述統計量を計算し、オブジェクトとして格納

```
Barplot2_df <- Country_df %>%
  group_by(Continent) %>%
  summarise(HDI = mean(HDI_2018, na.rm = TRUE))
```

# 記述統計の棒グラフ

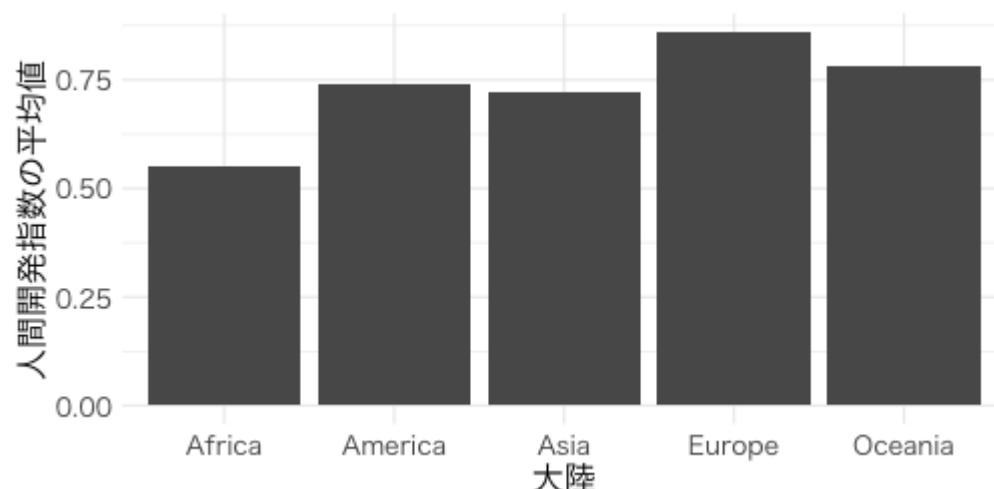
- 幾何オブジェクト: `geom_bar()`
- 必須マッピング: `x = グループ, y = 記述統計量`
- `stat = "identity"`は必須

```
データ %>%
  ggplot() +
  geom_bar(aes(x = グループ, y = 記述統計量), stat = "identity") +
  その他のレイヤー()
```

# 記述統計の棒グラフ

```
Barplot2 <- Barplot2_df %>%
  ggplot() +
  geom_bar(aes(x = Continent, y = HDI), stat = "identity") +
  labs(x = "大陸", y = "人間開発指数の平均値") +
  theme_minimal(base_family = "HiraKakuProN-W3") +
  theme(text = element_text(size = 16))
```

```
Barplot2
```

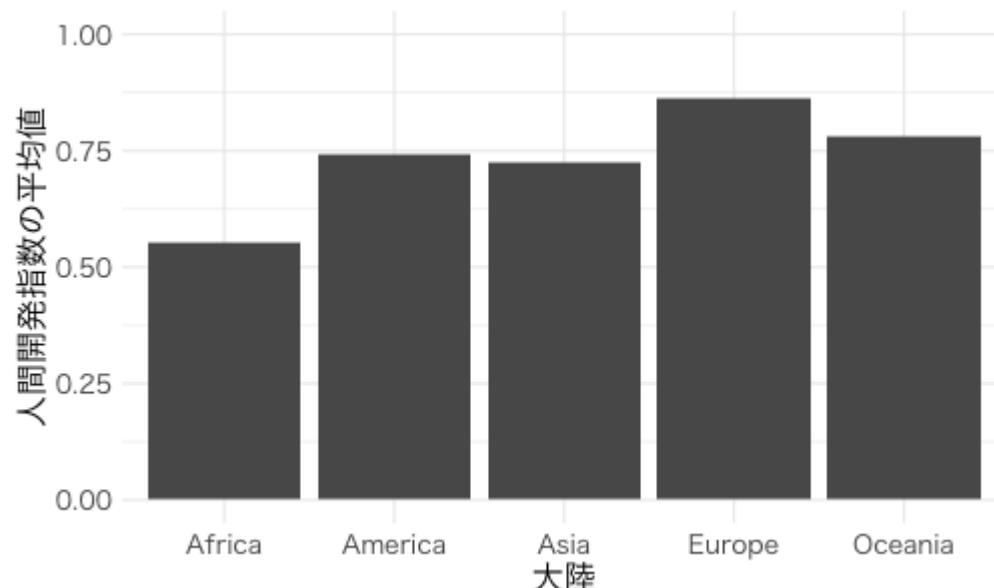


# 座標系の調整

- デカルト座標系（直交座標系）のカスタマイズ: `coord_cartesian()`
  - 横軸の上限・下限は`xlim`、縦軸は`ylim`で調整

```
# 縦軸の下限を0、上限を1に
```

```
Barplot2 +  
  coord_cartesian(ylim = c(0, 1))
```



# 次元の追加

- 今のグラフは2次元
  - 「大陸」と「人間開発指数の平均値」
- 次元を追加することも可能
  - 例) 更にOECD加盟有無で比較する

# 次元の追加: データの準備

```
Barplot2_df2 <- Country_df %>%
  group_by(Continent, OECD) %>%
  summarise(HDI = mean(HDI_2018, na.rm = TRUE)) %>%
  mutate(OECD = recode(OECD,
    "0" = "非加盟国",
    "1" = "加盟国"))

Barplot2_df2
```

```
## # A tibble: 9 x 3
## # Groups:   Continent [5]
##   Continent OECD      HDI
##   <chr>     <chr>    <dbl>
## 1 Africa    非加盟国  0.553
## 2 America   非加盟国  0.726
## 3 America   加盟国    0.843
## 4 Asia      非加盟国  0.708
## 5 Asia      加盟国    0.909
## 6 Europe    非加盟国  0.807
## 7 Europe    加盟国    0.899
## 8 Oceania   非加盟国  0.634
```

# 次元の追加: グラフの作成

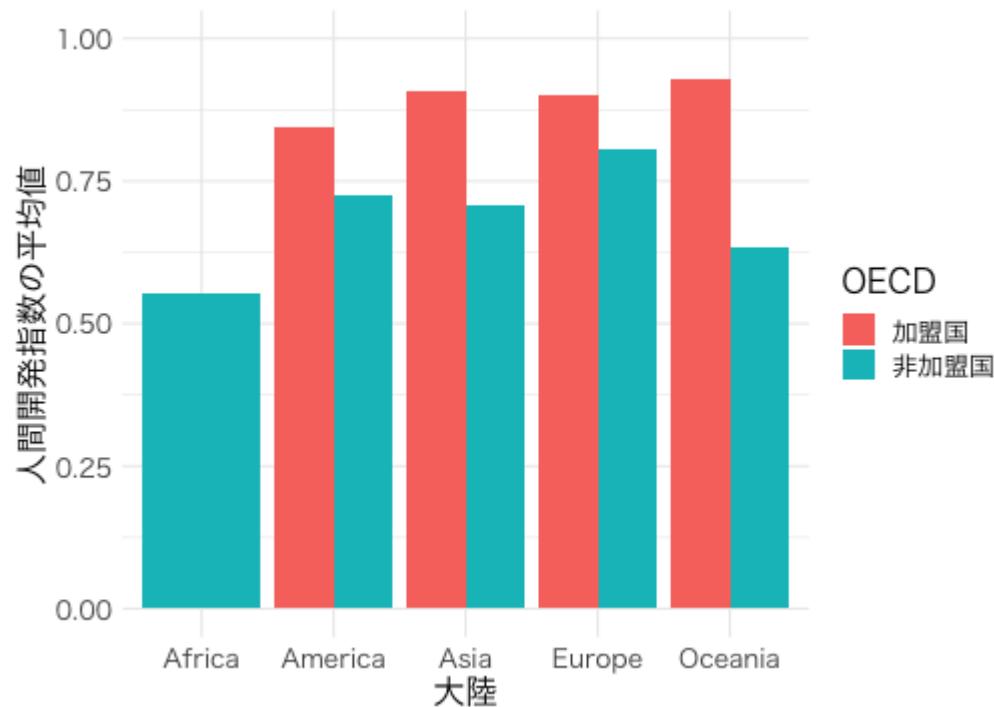
- OECDごとに棒の色を変える場合: `aes()`内に `fill = OECD` を追加
- `aes()` 外に `position = "dodge"` を追加

```
Barplot3 <- Barplot2_df2 %>%
  ggplot() +
  geom_bar(aes(x = Continent, y = HDI, fill = OECD),
           stat = "identity", position = "dodge") +
  labs(x = "大陸", y = "人間開発指数の平均値") +
  coord_cartesian(ylim = c(0, 1)) +
  theme_minimal(base_family = "HiraKakuProN-W3") +
  theme(text = element_text(size = 16))
```

# 次元の追加: グラフの作成

- OECDごとに棒の色を変える場合: `aes()`内に `fill = OECD` を追加
- `aes()` 外に `position = "dodge"` を追加

Barplot3



# 次元の追加: ファセット分割

ファセット (facet): 変数ごとの異なるグラフとして出力

- `facet_wrap(~ 変数名)`

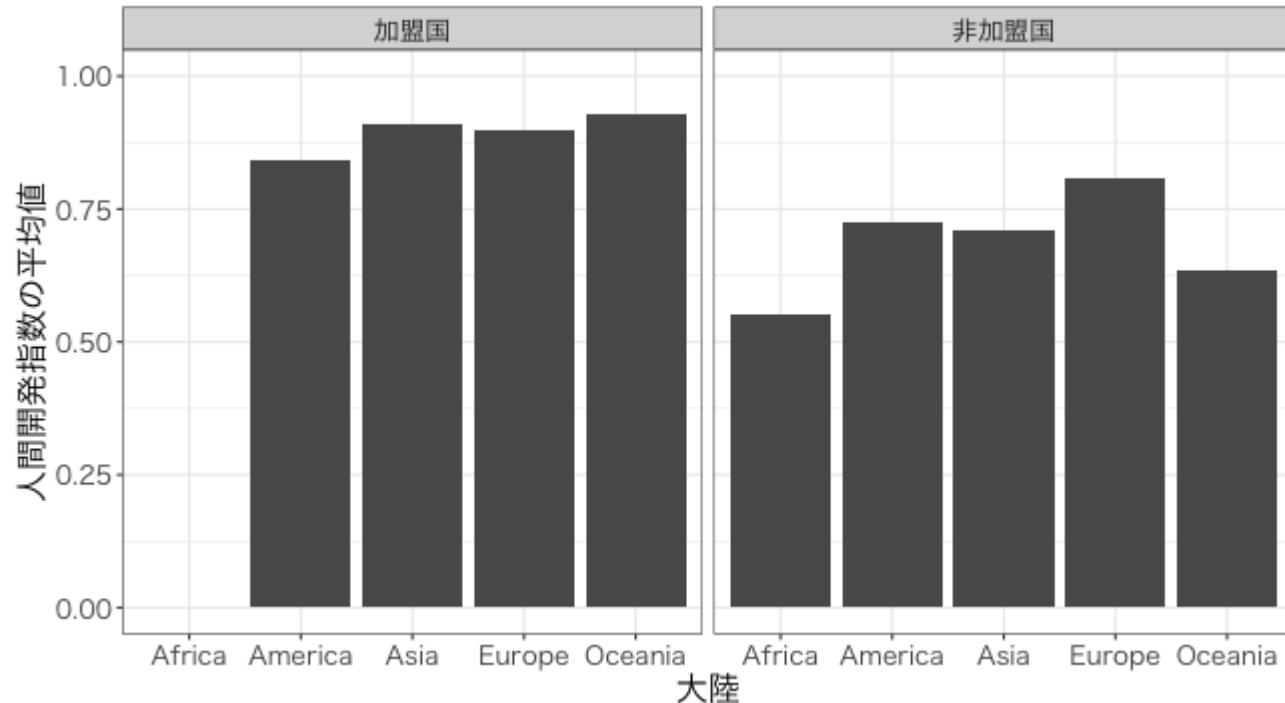
```
Barplot4 <- Barplot2_df2 %>%
  ggplot() +
  geom_bar(aes(x = Continent, y = HDI),
            stat = "identity") +
  labs(x = "大陸", y = "人間開発指数の平均値") +
  coord_cartesian(ylim = c(0, 1)) +
  facet_wrap(~ OECD) + [Yellow highlight]
  theme_minimal(base_family = "HiraKakuProN-W3") +
  theme(text = element_text(size = 16))
```

# 次元の追加: ファセット分割

ファセット (facet): 変数ごとの異なるグラフとして出力

- `facet_wrap(~ 変数名)`

Barplot4



# ヒストグラム

# ヒストグラム

- 幾何オブジェクト: `geom_histogram()`
- 必須マッピング: `x = 度数を計算する変数`
  - 各区間の度数は自動的に計算してくれるため、指定しない

```
# 2次元 (区間、度数)の場合
```

```
データ %>%
```

```
  ggplot() +
```

```
    geom_histogram(aes(x = 変数名))
```

```
# 3次元 (区間、度数、グループ)の場合
```

```
# alphaは透明度であり、0に近いほど透明になる
```

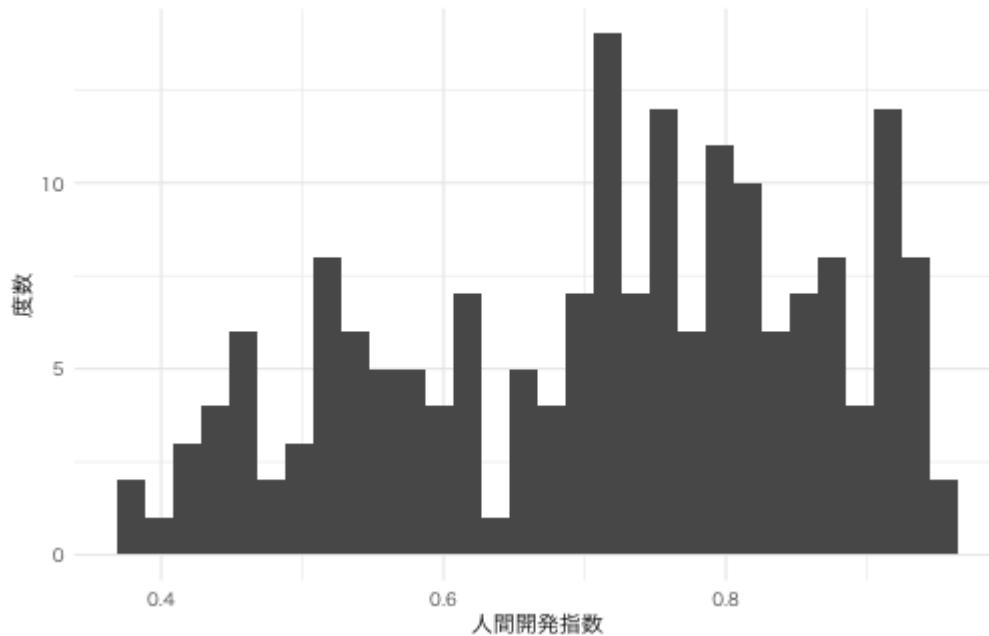
```
データ %>%
```

```
  ggplot() +
```

```
    geom_histogram(aes(x = 変数名, fill = グループ変数),
                   alpha = 0.5, position = "identity")
```

# ヒストグラム

```
Country_df %>%
  ggplot() +
  geom_histogram(aes(x = HDI_2018)) +
  labs(x = "人間開発指数", y = "度数") +
  theme_minimal(base_family = "HiraKakuProN-W3")
```



# ヒストグラム: カスタマイズ

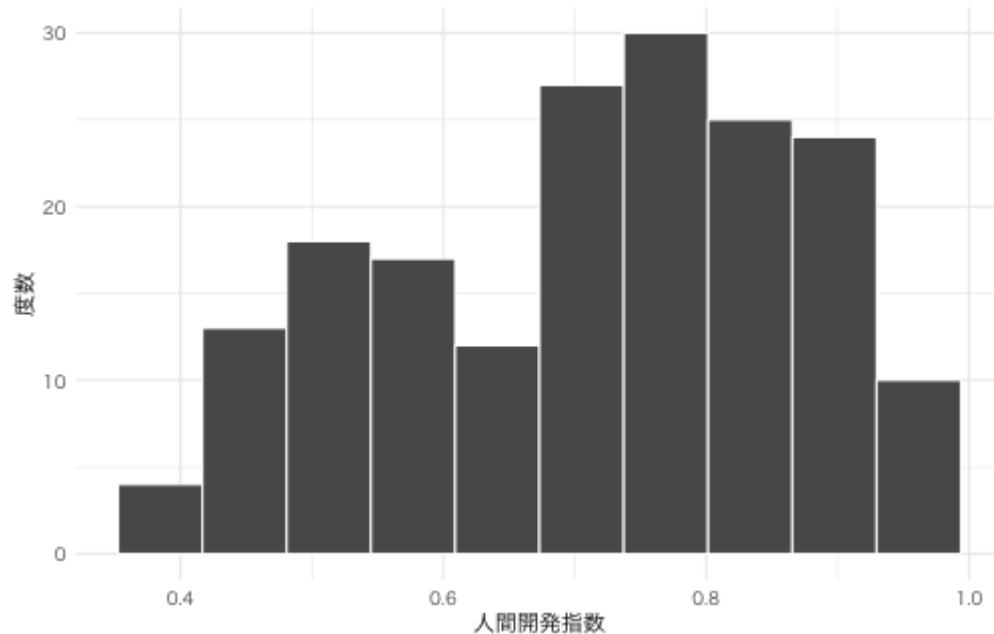
aes()の外側に入れる引数

- bins: 区間の数
- binwidth: 区間の幅
  - binsとbinwidthは一方のみ指定
- color: 枠線の色
- fill: 棒の色
  - colorとfillはaes()の中に入れて変数に対応させることも可

```
Histogram2 <- Country_df %>%
  ggplot() +
  geom_histogram(aes(x = HDI_2018), bins = 10, color = "white") +
  labs(x = "人間開発指数", y = "度数") +
  theme_minimal(base_family = "HiraKakuProN-W3")
```

# ヒストグラム: カスタマイズ

Histogram2



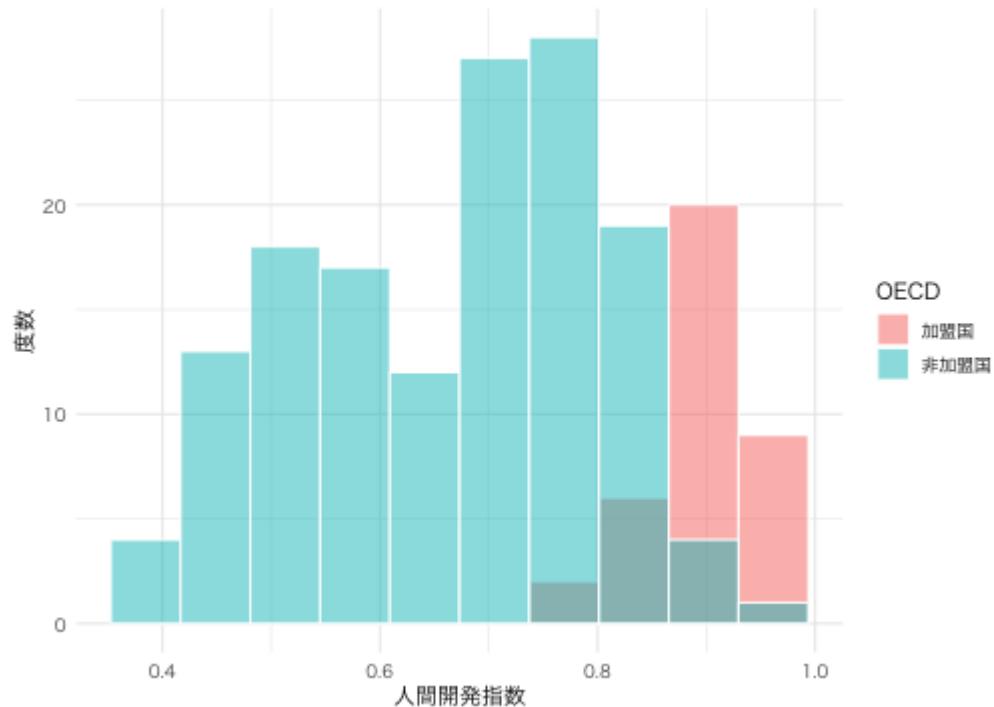
# 次元の追加

- OECD加盟有無で色分け
  - OECD変数を事前に数値型から文字型（または、factor型）へ変換
- `facet_wrap()`によるファセット分割も使用可

```
Histogram3 <- Country_df %>%
  mutate(OECD = recode(OECD,
    "0" = "非加盟国",
    "1" = "加盟国")) %>%
  ggplot() +
  geom_histogram(aes(x = HDI_2018, fill = OECD),
    position = "identity", alpha = 0.5,
    bins = 10, color = "white") +
  labs(x = "人間開発指数", y = "度数") +
  theme_minimal(base_family = "HiraKakuProN-W3")
```

# 次元の追加

Histogram3



箱ひげ図

# 箱ひげ図

- 複数のグループ間で変数の分布を素早く比較可能
  - 中央値、最小値、最大値、第1四分位点、第3四分位点

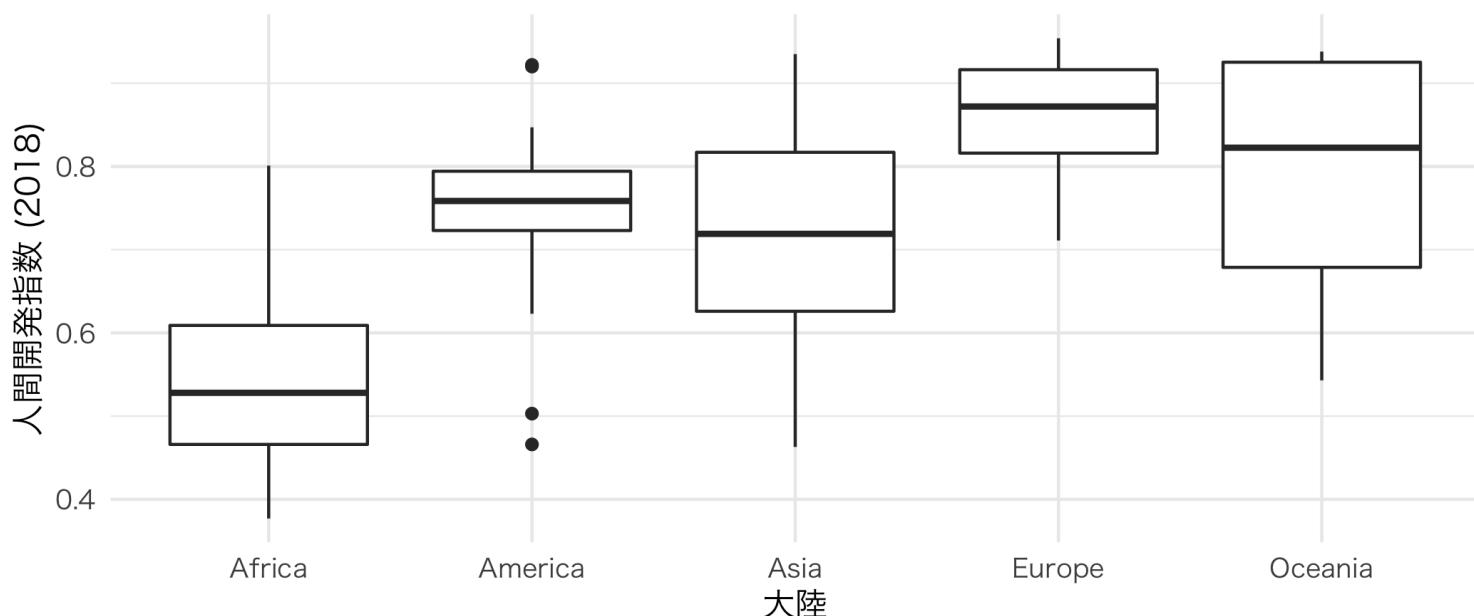
# 箱ひげ図

- 幾何オブジェクト: `geom_boxplot()`
- 必須マッピング: `y = 分布を確認したい変数`
  - 中央値、四分位点などは自動的に計算してくれる
- 推奨マッピング: `x = グループ変数`

```
データ %>%
  ggplot() +
  geom_histogram(aes(x = グループ変数, y = 変数))
```

# 箱ひげ図

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



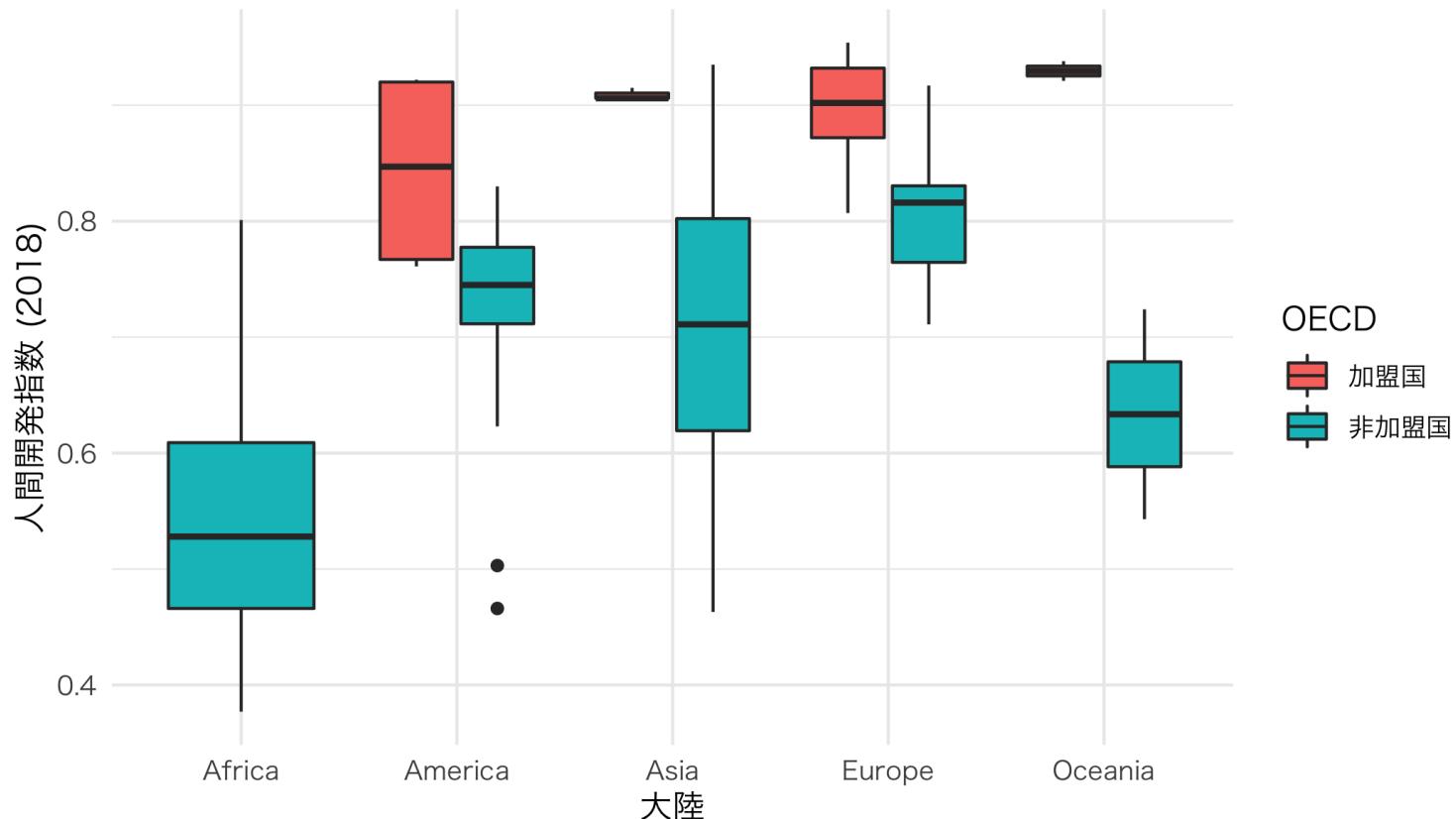
# 次元の追加

- 推奨する方法: `aes()`内に `fill` 引数追加、またはファセット分割

```
Boxplot2 <- Country_df %>%
  mutate(OECD = recode(OECD,
                        "0" = "非加盟国",
                        "1" = "加盟国")) %>%
  ggplot() +
  geom_boxplot(aes(x = Continent, y = HDI_2018, fill = OECD)) +
  labs(y = "人間開発指数 (2018)", x = "大陸") +
  theme_minimal(base_family = "HiraKakuProN-W3")
```

# 次元の追加

Boxplot2



# 散布図

# 散布図

- 幾何オブジェクト: `geom_point()`
- 必須マッピング: 点の位置 (`x` (横軸上) と `y` (縦軸上) )
  - 主に使う次元追加の方法
  - `facet_wrap()`: ファセット分割
  - `color`: 点の色分け
    - 枠線付きの点の場合、`color`は枠線、`fill`が点の色
  - `size`: 点の大きさ

```
データ %>%
  ggplot() +
  geom_point(aes(x = 横軸上の点の位置, y = 縦軸上の点の位置)) +
  その他のレイヤー
```

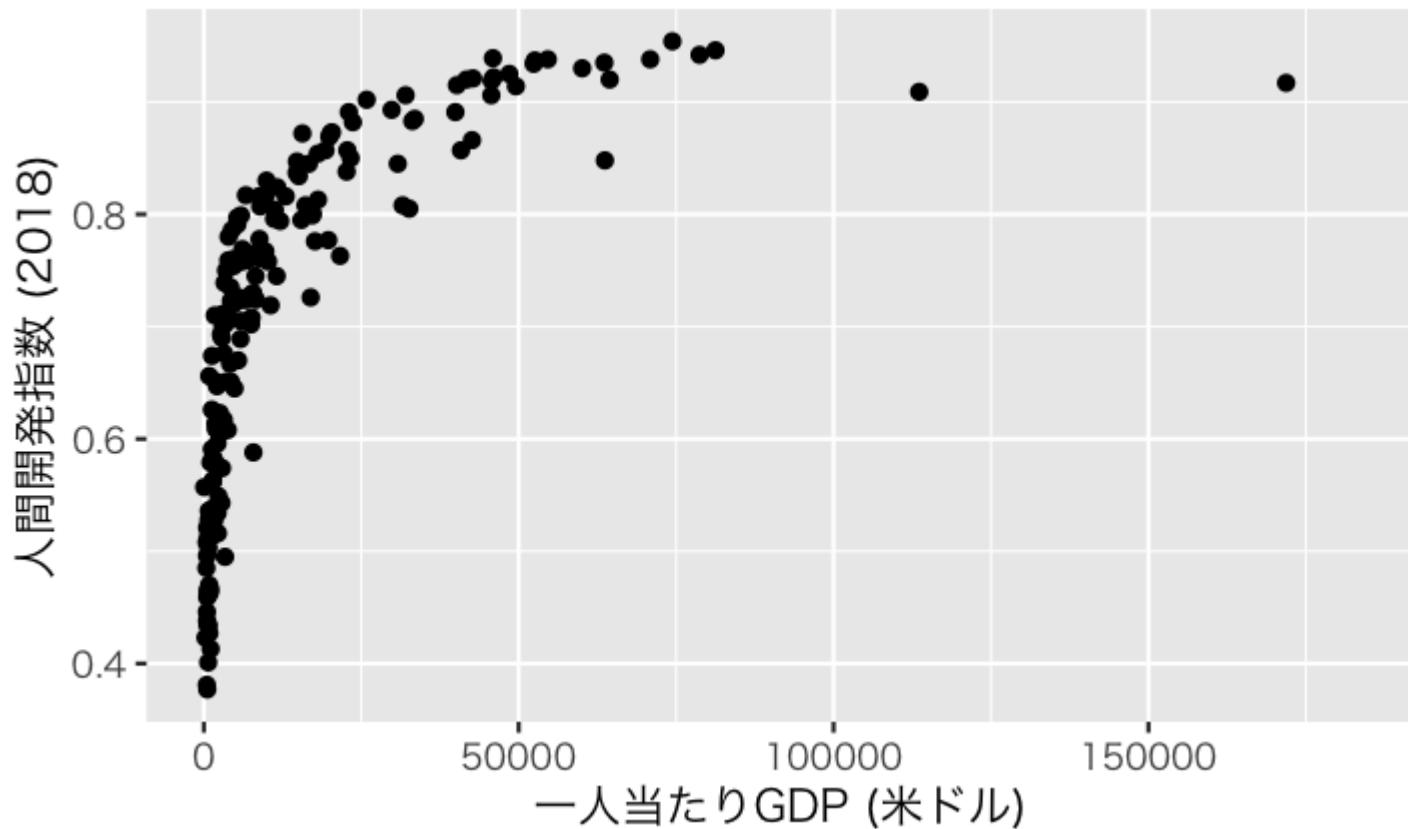
# 散布図

一人当たりGDPと人間開発指数の関係は？

```
Country_df %>%
  ggplot() +
  geom_point(aes(x = GDP_per_capita, y = HDI_2018)) +
  labs(x = "一人当たりGDP (米ドル)", y = "人間開発指数 (2018)") +
  theme_gray(base_family = "HiraKakuProN-W3")
```

# 散布図

一人当たりGDPと人間開発指数の関係は？



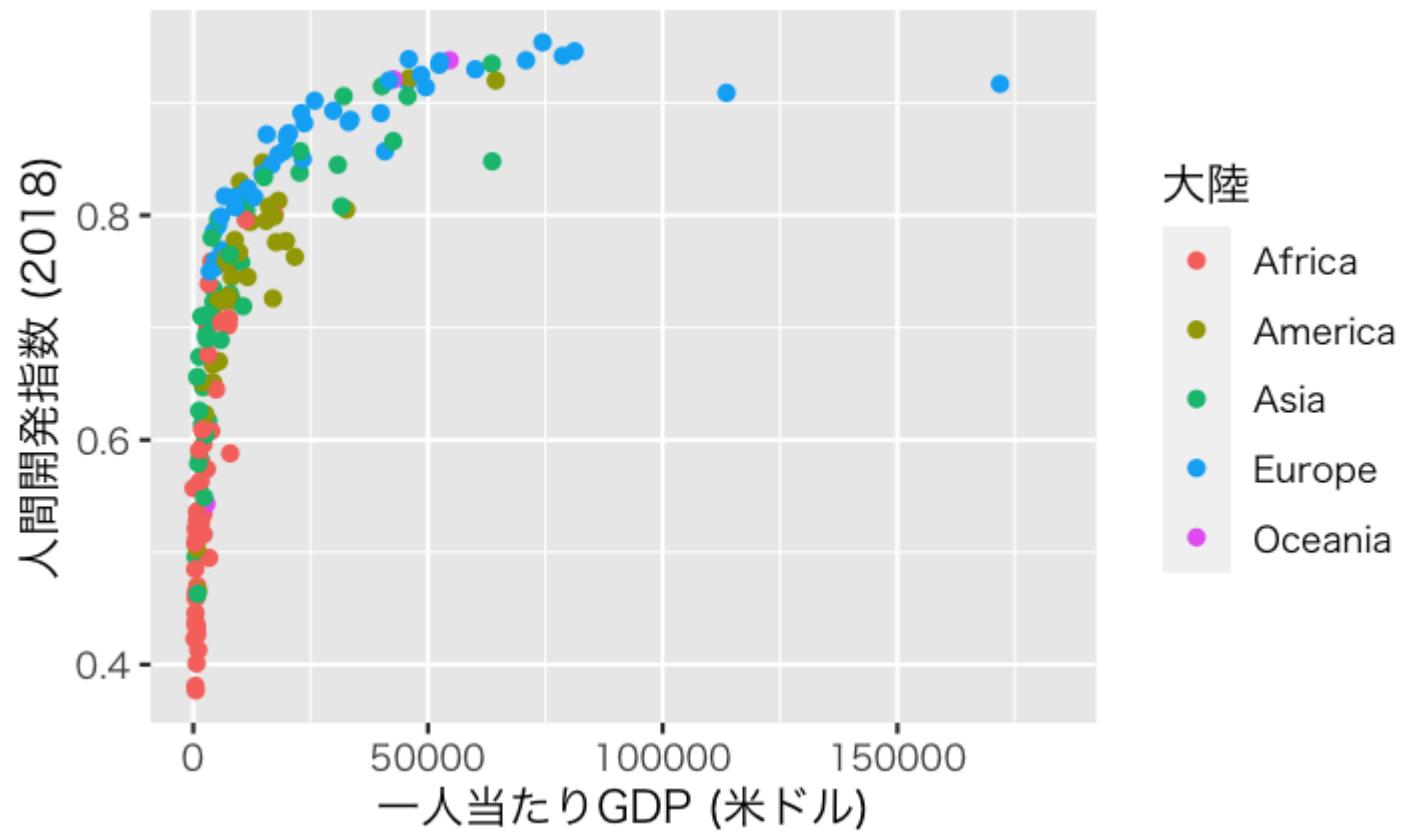
# 次元の追加: 色分け

大陸ごとに色分けをする: `color`

```
Country_df %>%
  ggplot() +
  geom_point(aes(x = GDP_per_capita, y = HDI_2018,
                 color = Continent)) +
  labs(x = "一人当たりGDP (米ドル)", y = "人間開発指数 (2018)",
       color = "大陸") +
  theme_gray(base_family = "HiraKakuProN-W3")
```

# 次元の追加: 色分け

大陸ごとに色分けをする: color



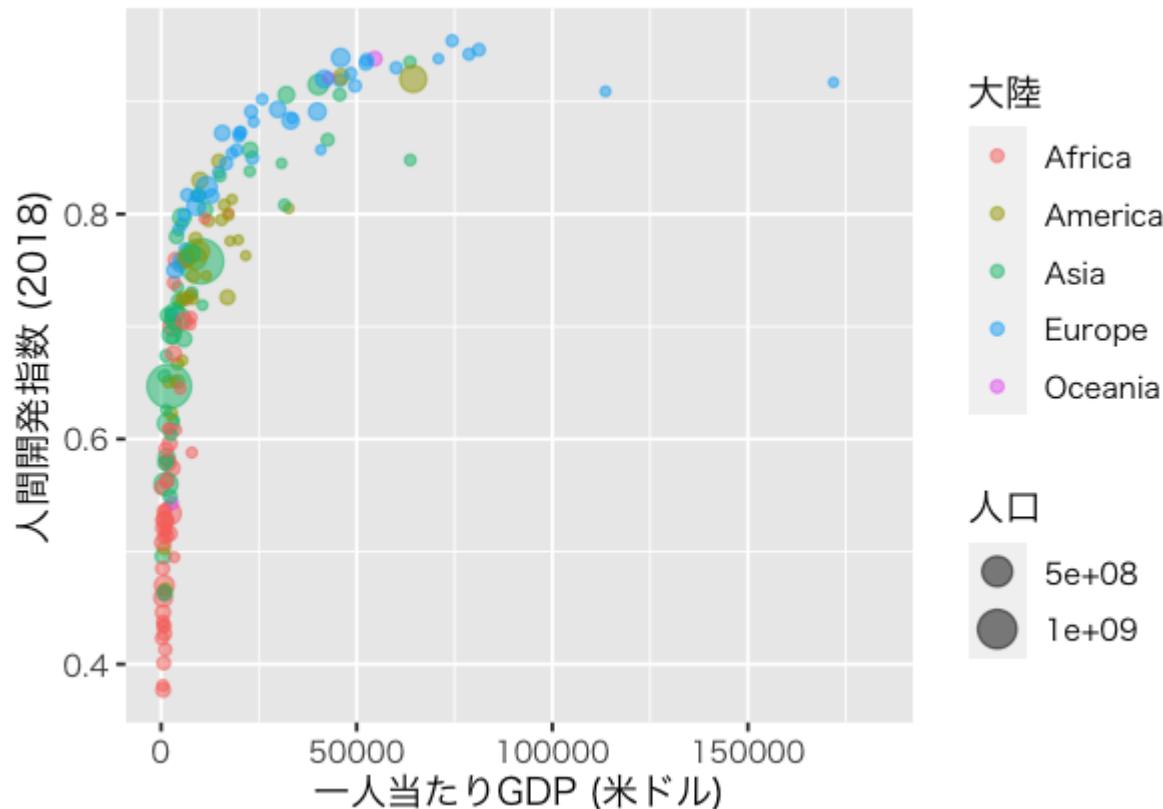
# 次元の追加: 大きさ

更に人口に応じて大きさを変更: `size`

```
Country_df %>%
  ggplot() +
  geom_point(aes(x = GDP_per_capita, y = HDI_2018,
                 color = Continent, size = Population),
              alpha = 0.5) +
  labs(x = "一人当たりGDP (米ドル)", y = "人間開発指数 (2018)",
       color = "大陸") +
  theme_gray(base_family = "HiraKakuProN-W3")
```

# 次元の追加: 大きさ

更に人口に応じて大きさを変更: size



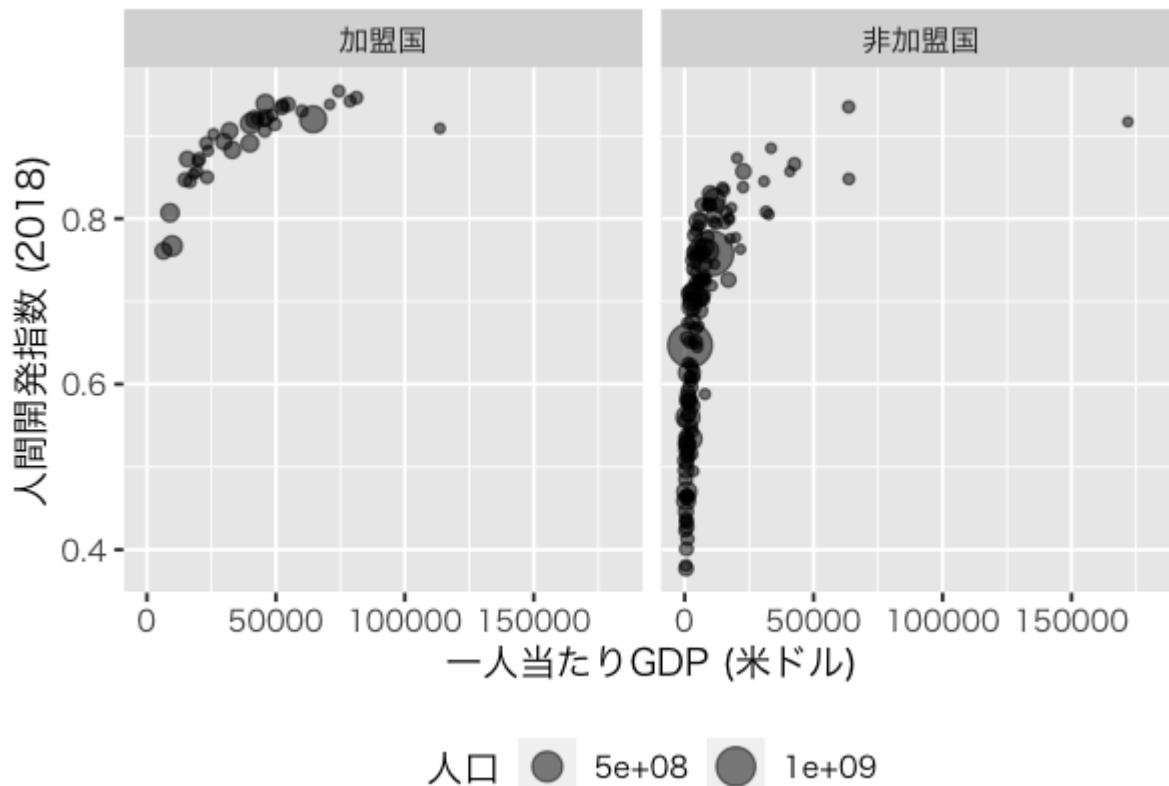
# 次元の追加: ファセット

OECD加盟有無でファセットを分割

```
Country_df %>%
  mutate(OECD2 = recode(OECD,
                        "0" = "非加盟国",
                        "1" = "加盟国")) %>%
  ggplot() +
  geom_point(aes(x = GDP_per_capita, y = HDI_2018,
                 size = Population), alpha = 0.5) +
  labs(x = "一人当たりGDP (米ドル)", y = "人間開発指数 (2018)",
       size = "人口") +
  facet_wrap(~OECD2) +
  theme_gray(base_family = "HiraKakuProN-W3") +
  theme(legend.position = "bottom")
```

# 次元の追加: ファセット

OECD加盟有無でファセットを分割



# 折れ線グラフ

# 折れ線グラフ

- 幾何オブジェクト: `geom_line()`
- 必須マッピング: 線の傾きが変化し得る点の位置 (`x` (横軸上) と `y` (縦軸上) )
  - `y`は数値型の連続変数
  - `x`は順序付き連続変数 (数値型、factor型、日付型)
  - 横軸において、ある値はデータ内において一回のみ登場すること
  - 例) 株価データの場合、「2020年8月25日」は一回のみ登場
  - 数力国データの場合、グループ内において一回のみ登場

```
データ %>%
  ggplot() +
  geom_line(aes(x = 横軸上の点の位置, y = 縦軸上の点の位置)) +
  その他のレイヤー
```

# 折れ線グラフ

- COVID19\_Asia2.csvを使用
- Country: 国
- Date: 年月日
  - 文字型になっているため、as.Date()で日付型へ変換が必要
- Confirmed\_Day: 「新規」感染者数
- Confirmed\_Total: 「累積」感染者数

COVID\_df

```
## # A tibble: 684 x 4
##   Country Date     Confirmed_Day Confirmed_Total
##   <chr>    <chr>        <dbl>          <dbl>
## 1 China    2020/1/22      548            548
## 2 China    2020/1/23      95             643
## 3 China    2020/1/24     277            920
## 4 China    2020/1/25     486           1406
## 5 China    2020/1/26     669           2075
## 6 China    2020/1/27     802           2877
## 7 China    2020/1/28    2632          5509
## 8 China    2020/1/29     578           6087
```

# 日付型へ変換

- データ型に関しては[SONG・矢内資料の第8章](#)を参照

```
COVID_df <- COVID_df %>%
  mutate(Date = as.Date(Date))

COVID_df
```

```
## # A tibble: 684 x 4
##   Country Date       Confirmed_Day Confirmed_Total
##   <chr>    <date>           <dbl>            <dbl>
## 1 China    2020-01-22      548             548
## 2 China    2020-01-23      95              643
## 3 China    2020-01-24     277             920
## 4 China    2020-01-25     486            1406
## 5 China    2020-01-26     669            2075
## 6 China    2020-01-27     802            2877
## 7 China    2020-01-28    2632            5509
## 8 China    2020-01-29     578            6087
## 9 China    2020-01-30    2054            8141
## 10 China   2020-01-31    1661            9802
## # ... with 674 more rows
```

# 折れ線グラフ

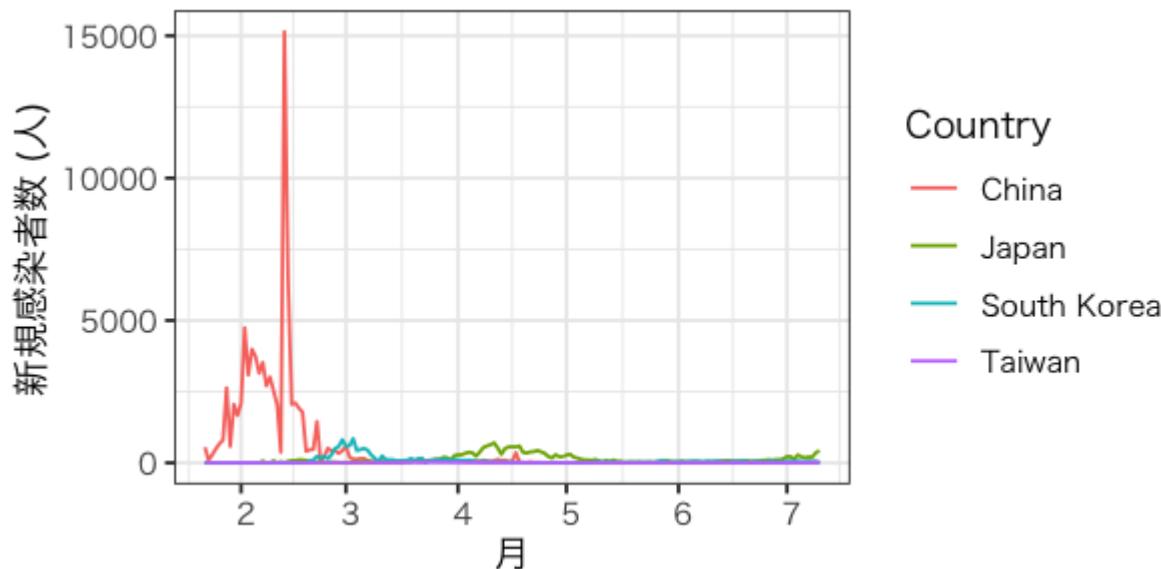
- 日本における日付ごとの新規感染者数

```
COVID_df %>%
  filter(Country == "Japan") %>%
  ggplot() +
  geom_line(aes(x = Date, y = Confirmed_Day)) +
  labs(x = "月", y = "新規感染者数 (人)") +
  theme_bw(base_family = "HiraKakuProN-W3")
```

# 折れ線グラフ: 複数の線の場合

- 線ごとに色を変える場合: `color`を指定

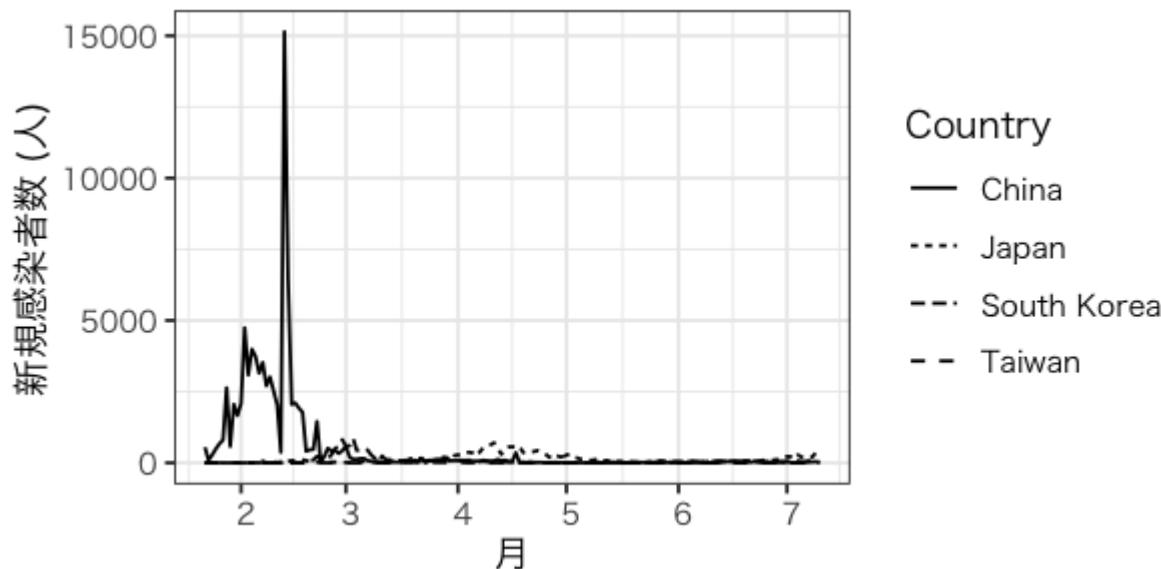
```
COVID_df %>%
  ggplot() +
  geom_line(aes(x = Date, y = Confirmed_Day, color = Country)) +
  labs(x = "月", y = "新規感染者数 (人)") +
  theme_bw(base_family = "HiraKakuProN-W3")
```



# 折れ線グラフ: 複数の線の場合

- 線ごとに線の種類を変える場合: `linetype`を指定

```
COVID_df %>%
  ggplot() +
  geom_line(aes(x = Date, y = Confirmed_Day, linetype = Country)) +
  labs(x = "月", y = "新規感染者数 (人)") +
  theme_bw(base_family = "HiraKakuProN-W3")
```



# 整然データの必要性

- 新規感染者数という1つの変数が4列にわたって格納
  - 雑線データ
- `pivot_longer()`などを使って整然データへ

Country	2020/05/01	2020/05/02	2020/05/03	2020/05/04
China	3	0	5	2
Japan	274	303	200	168
South Korea	6	13	8	3
Taiwan	0	3	4	2

# 整然データの必要性

- 2020/05/01列から2020/05/04列をLong型へ
  - 変数名が数字や記号で始まる場合は、変数名を` `で囲む
    - 推奨する方法は` `
  - 日付をDate列へ
  - 新規感染者数をConfirmed列へ
  - 日付はデータ型をDate型へ

```
データ名 %>%
  pivot_longer(cols = "2020/05/01":"2020/05/04",
               names_to = "Date",
               values_to = "Confirmed") %>%
  mutate(Date = as.Date(Date))
```

# 整然データの必要性

Country	Date	Confirmed
China	2020/05/01	3
China	2020/05/02	0
China	2020/05/03	5
China	2020/05/04	2
Japan	2020/05/01	274
Japan	2020/05/02	303
Japan	2020/05/03	200
Japan	2020/05/04	168
South Korea	2020/05/01	6
South Korea	2020/05/02	13
South Korea	2020/05/03	8
South Korea	2020/05/04	3
Taiwan	2020/05/01	0

# ggplot2いろいろ

# 図の保存

`ggsave()`関数を使用

- 保存する図をオブジェクトに格納しておくこと
- ファイル名の拡張子を必ず指定（ファイル名.png、ファイル名.pdfなど）
- 大きさの単位はインチ ("in")、センチメートル ("cm")、ミリメートル ("mm")

```
ggsave(filename = "保存するファイル名",
        plot      = 図のオブジェクト名,
        units    = "幅と高さの単位",
        width     = 図の幅,
        height    = 図の高さ)
```

macOS使用、かつ図内に日本語が含まれている場合

- `quartz()`を使用: 教科書第17.3.3章

# 幾何オブジェクトの引数

## color引数の例

- `aes()`内で指定: 変数ごとに色分け（変数を指定）

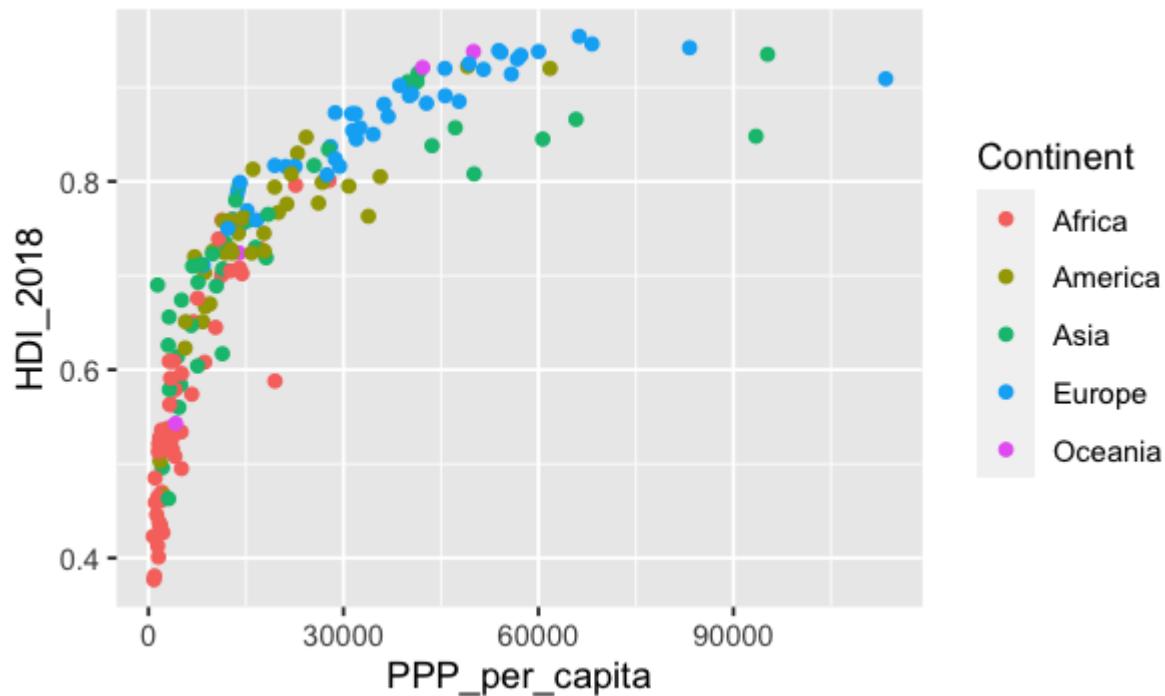
```
geom_point(aes(x = PPP_per_capita, y = HDI_2018, color = Continent))
```

- `aes()`外で指定: 全要素の色指定（具体的な色を指定）

```
geom_point(aes(x = PPP_per_capita, y = HDI_2018), color = "Red")
```

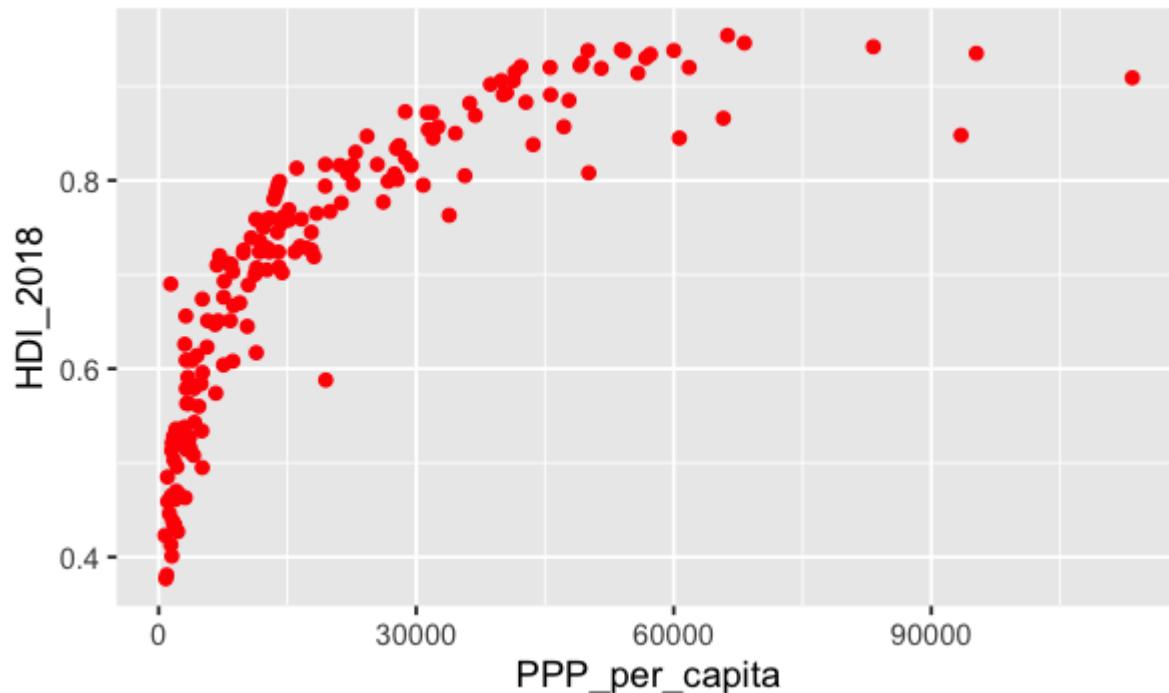
# 幾何オブジェクトの引数

```
Country_df %>%  
  ggplot() +  
  geom_point(aes(x = PPP_per_capita, y = HDI_2018, color = Continent))
```



# 幾何オブジェクトの引数

```
Country_df %>%  
  ggplot() +  
  geom_point(aes(x = PPP_per_capita, y = HDI_2018), color = "Red")
```



# theme() レイヤー

図全体を貫通する見た目を細かく調整可能

```
データ %>%  
  ggplot() +  
  geom_*( ) +  
  theme()
```

凡例の位置調整 ("left"、 "bottom"、 "top"、 "right")

```
theme(legend.position = "bottom")
```

文字サイズの調整

```
theme(text = element_text(size = 16))
```

# R Graph Gallery

様々なグラフの例とコードが見られるホームページ

- The R Graph Gallery

## Density with `geom_density`

A density chart is built thanks to the `geom_density` geom of ggplot2 (see a [basic example](#)). It is possible to plot this density upside down by specifying `y = -..density...`. It is advised to use `geom_label` to indicate variable names.

```
# Libraries
library(ggplot2)
library(hrbrthemes)

# Dummy data
data <- data.frame(
  var1 = rnorm(1000),
  var2 = rnorm(1000, mean=2)
)

# Chart
p <- ggplot(data, aes(x=x)) +
  # Top
  geom_density(aes(x = var1, y = ..density..), fill="#69b3a2") +
  geom_label(aes(x=4.5, y=0.25, label="variable1"), color="#69b3a2") +
  # Bottom
  geom_density(aes(x = var2, y = -..density..), fill="#404080") +
  geom_label(aes(x=4.5, y=-0.25, label="variable2"), color="#404080") +
  theme_ipsum() +
  xlab("value of x")

#p
```

