

# ミクロ政治データ分析実習

## 第12回 可視化 (2)

---

そん じえひょん

宋 財 沄

関西大学総合情報学部

2021/7/1 (updated: 2021-06-23)

# 代表的な5種類のグラフ

---

# 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_box()`
- 散布図 (scatter plot)
  - `geom_point()`
- 折れ線グラフ (line plot)
  - `geom_line()`

# グラフ作成の手順

1. 作成したいグラフを決める
2. 作成したいグラフの完成図を想像する or 描いてみる
3. グラフ上の要素（点、線、面）が持つ情報を考える
4. 3の情報が一つの変数（列）と対応するような整然データを作成する
5. {ggplot2}で作図
6. 図のカスタマイズ
7. 図の保存

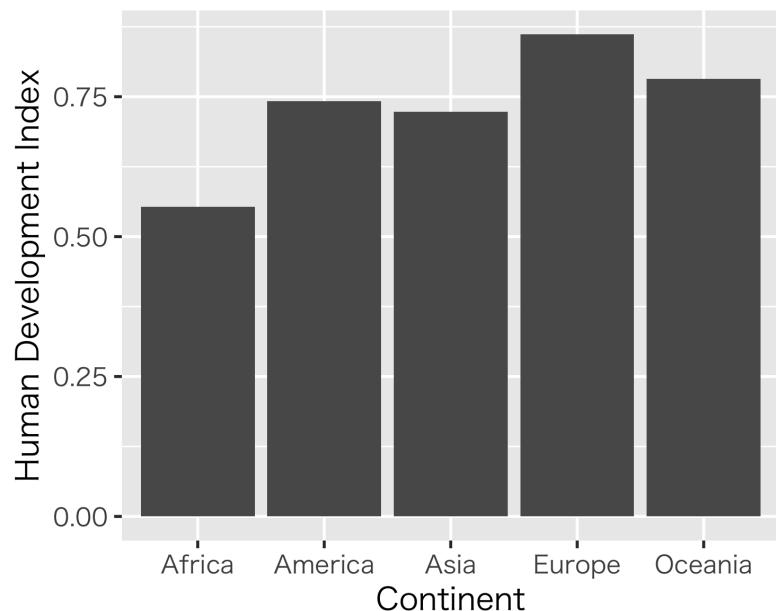
# 棒グラフ

---

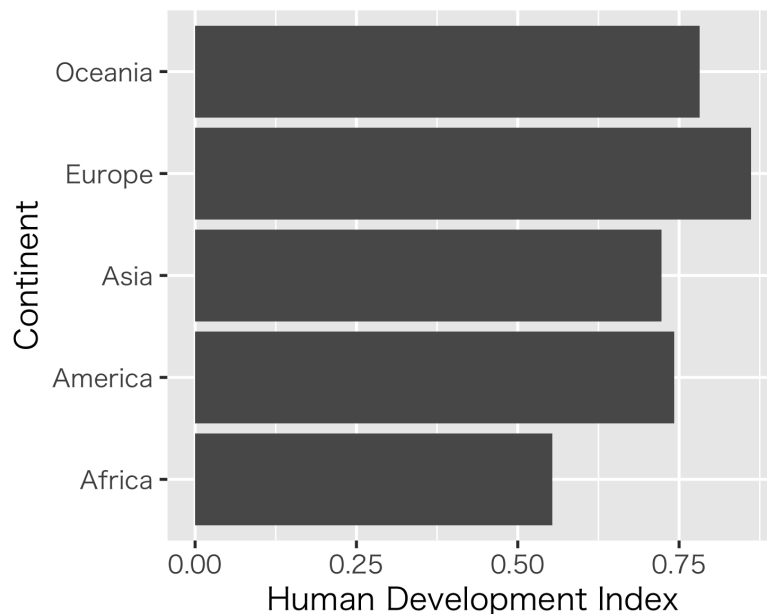
# 棒グラフの必須要素

棒グラフを作成する際に必要な**最低限**の情報

- x: 棒の横軸上の位置 (大陸)
- y: 棒の高さ (人間開発指数の平均値)



- y: 棒の縦軸上の位置 (大陸)
- x: 棒の長さ (人間開発指数の平均値)



いずれも、「大陸」と「人間開発指数の平均値」の列が必要

# データの用意

## 第8回講義のデータを使用

- {dplyr}を使用し、大陸 (Continent) ごとの人間開発指数 (HDI\_2018) の平均値を計算し、df2 という名で格納

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

```
df2 <- df %>%
  group_by(Continent) %>%
  summarise(HDI = mean(HDI_2018, na.rm = TRUE))
```

```
df2
```

```
## # A tibble: 5 x 2
##   Continent    HDI
##   <chr>      <dbl>
## 1 Africa     0.553
## 2 America    0.742
## 3 Asia       0.723
## 4 Europe     0.861
## 5 Oceania    0.782
```

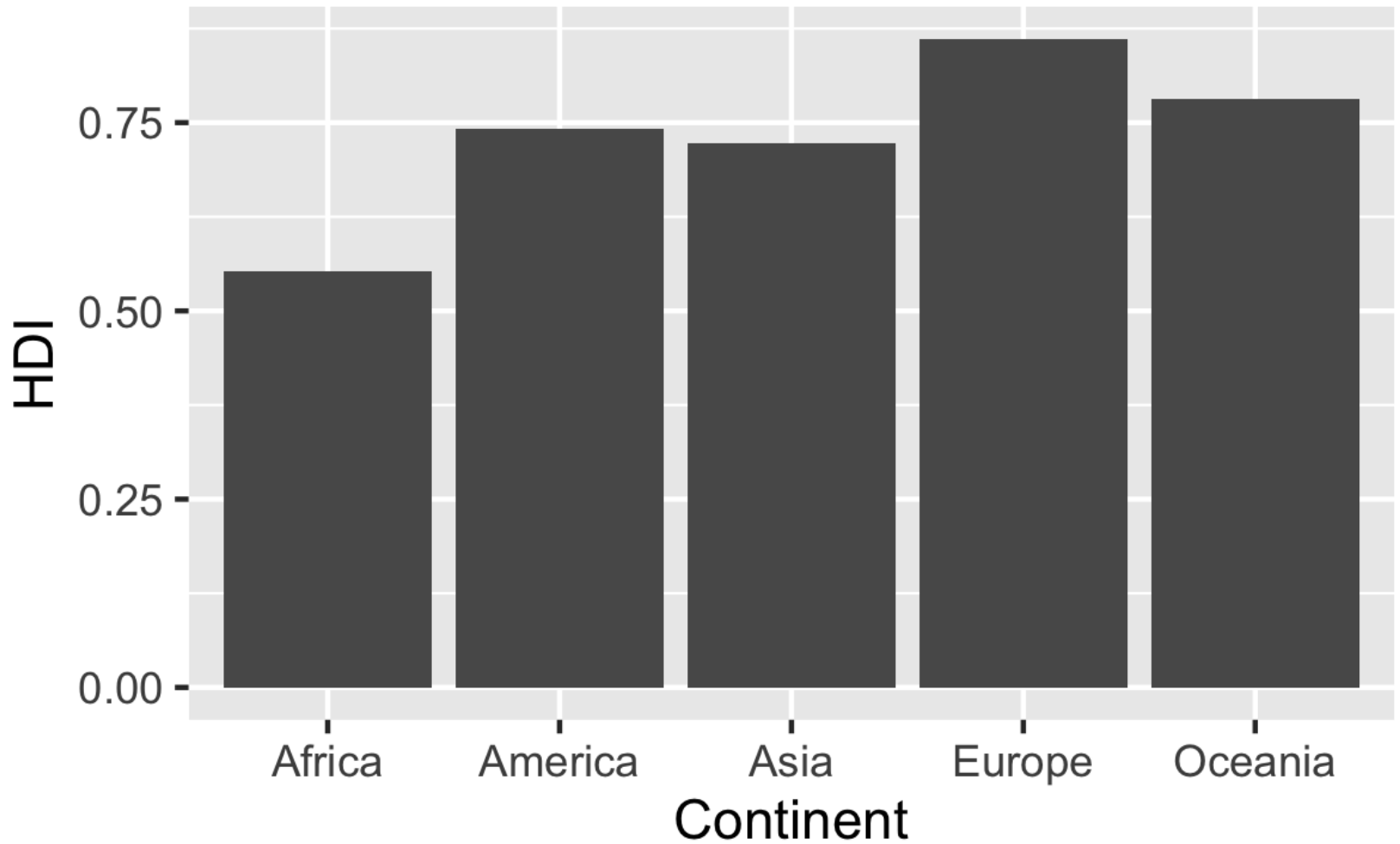
# とりあえず作図してみる (1)

- 使用する幾何オブジェクト: `geom_bar()`
- マッピング要素 (幾何オブジェクト内に `aes()`)
  - 棒の横軸上の位置は大陸 (`Continent`) を意味する
  - 棒の高さは人間開発指数の平均値 (`HDI`) を意味する
- `geom_bar()` 内に `stat = "identity"` を忘れないこと

```
df2 %>%  
  ggplot() +  
  geom_bar(aes(x = Continent, y = HDI), stat = "identity")
```



# とりあえず作図してみる (2)



# 日本語の使用 (1)

**Step1:** df2 の Continent 列を日本語にリコーディング

```
df2 <- df2 %>%  
  mutate(Continent_J = case_when(Continent == "Asia"      ~ "アジア",  
                                   Continent == "Africa"    ~ "アフリカ",  
                                   Continent == "America"    ~ "アメリカ",  
                                   Continent == "Europe"     ~ "ヨーロッパ",  
                                   TRUE                       ~ "オセアニア"))
```

df2

```
## # A tibble: 5 x 3  
##   Continent    HDI Continent_J  
##   <chr>      <dbl> <chr>  
## 1 Africa    0.553 アフリカ  
## 2 America   0.742 アメリカ  
## 3 Asia      0.723 アジア  
## 4 Europe    0.861 ヨーロッパ  
## 5 Oceania   0.782 オセアニア
```

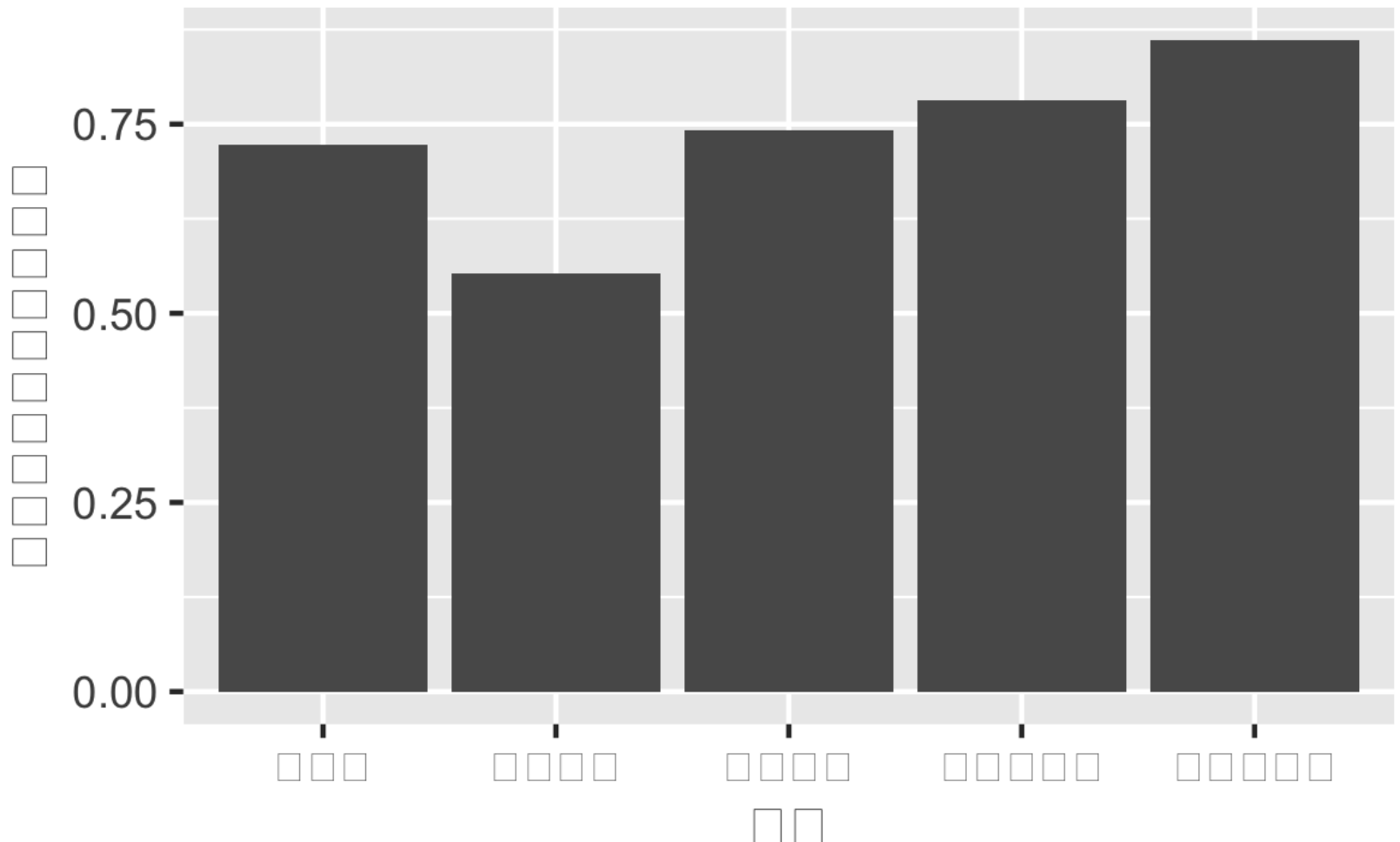
# 日本語の使用 (2)

**Step2:** 図のラベルを修正 ( `labs()` )

```
df2 %>%  
  ggplot() +  
  geom_bar(aes(x = Continent_J, y = HDI), stat = "identity") +  
  labs(x = "大陸", y = "人間開発指数の平均値")
```

# 日本語の使用 (2)

Step2: 図のラベルを修正 (labs())



# 日本語の使用 (3)

## Step3: 文字化けが生じる場合、日本語フォントの設定

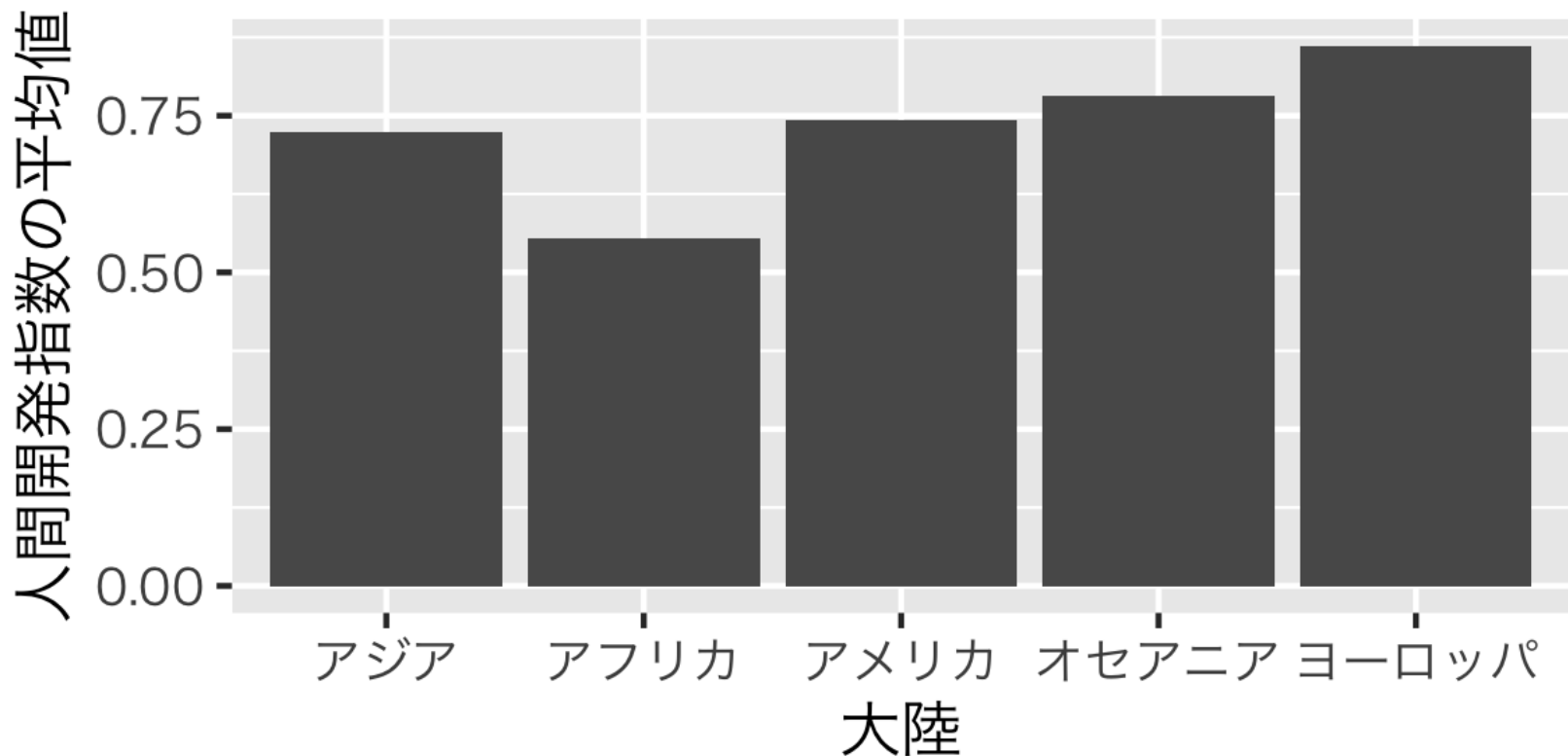
- `theme_gray()`: {ggplot2}の基本テーマ
  - 他にも `theme_bw()`、`theme_minimal()` など
- macOSの場合、`"HiraKakuProN-W3"`、Windowsの場合、`"Yu Gothic"` を指定
- NIIオンライン分析システムを使用する場合、生じない

```
df2 %>%  
  ggplot() +  
  geom_bar(aes(x = Continent_J, y = HDI), stat = "identity") +  
  labs(x = "大陸", y = "人間開発指数の平均値") +  
  theme_gray(base_family = "HiraKakuProN-W3")
```

# 日本語の使用 (3)

**Step3:** 文字化けが生じる場合、日本語フォントの設定

- `theme_gray()`: {ggplot2}の基本テーマ
  - 他にも `theme_bw()`、`theme_minimal()` など
- macOSの場合、`"HiraKakuProN-W3"`、Windowsの場合、`"Yu Gothic"` を指定



# 棒の並び替え

アルファベット順に並べ替えたい場合

- Continet\_J 列をfactor化し、アフリカ、アメリカ、アジア、ヨーロッパ、オセアニア順とする

```
df2 %>%
```

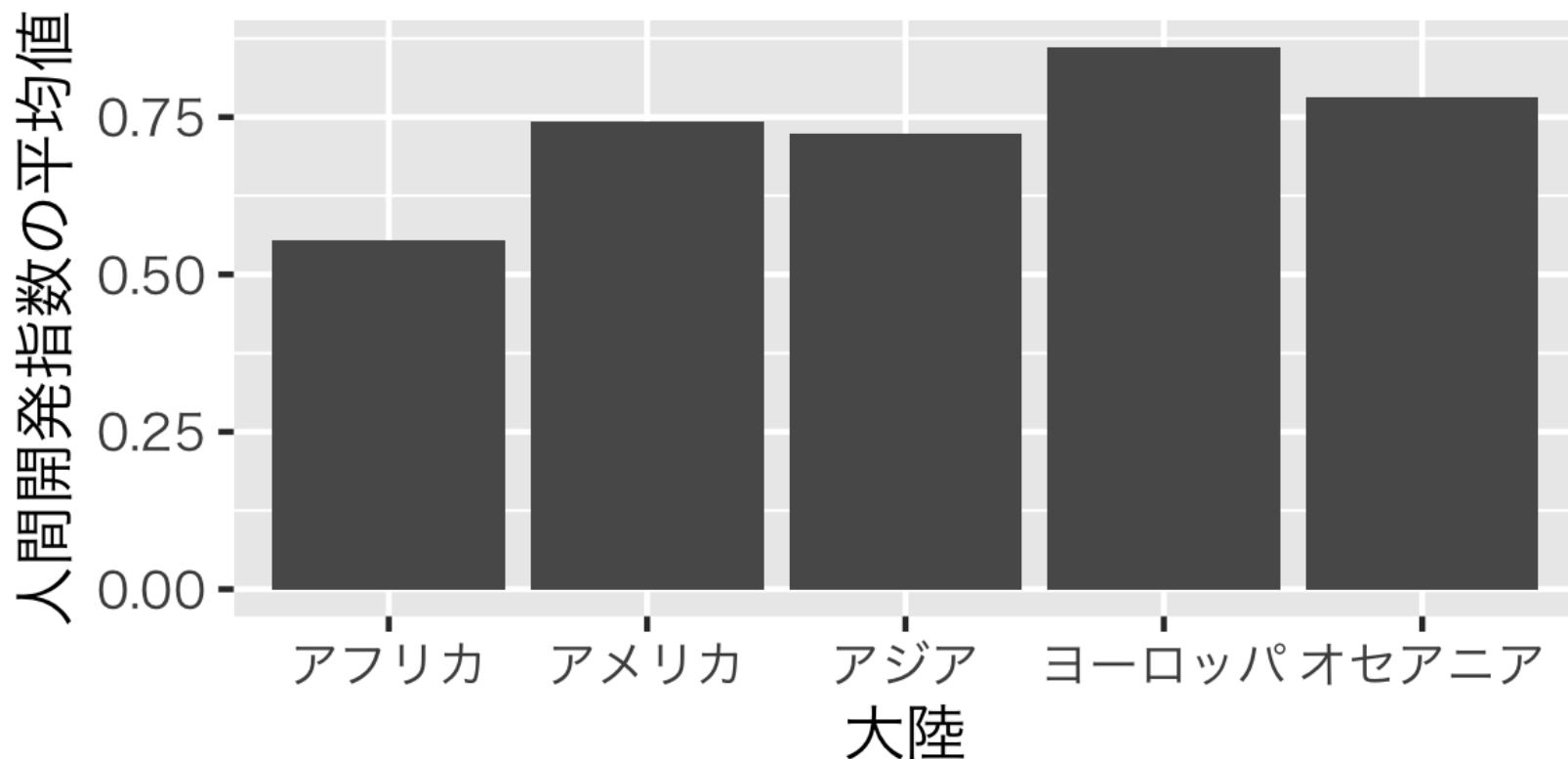
```
mutate(Continet_J = factor(Continet_J,  
                           levels = c("アフリカ", "アメリカ", "アジア",  
                                       "ヨーロッパ", "オセアニア"))) %>%
```

```
ggplot() +  
geom_bar(aes(x = Continet_J, y = HDI), stat = "identity") +  
labs(x = "大陸", y = "人間開発指数の平均値") +  
theme_gray(base_family = "HiraKakuProN-W3")
```

# 棒の並び替え

アルファベット順に並べ替えたい場合

- Continet\_J列をfactor化し、アフリカ、アメリカ、アジア、ヨーロッパ、オセアニア順とする





# 便利な関数: fct\_inorder()

{forcats}の fct\_inorder() 関数

- {forcats}は{tidyverse}の一部
- 使い方: fct\_inorder(factor化する変数名)
- factor化を行い、各要素順番を**表で登場した順番**とする

# 前ページのコードと同じ結果が得られる

df2 %>%

```
mutate(Continent_J = fct_inorder(Continent_J)) %>%  
  ggplot() +  
  geom_bar(aes(x = Continent_J, y = HDI), stat = "identity") +  
  labs(x = "大陸", y = "人間開発指数の平均値") +  
  theme_gray(base_family = "HiraKakuProN-W3")
```

# 次元の追加

---

# ケース数の棒グラフ

各政治体制に属する国家数を計算

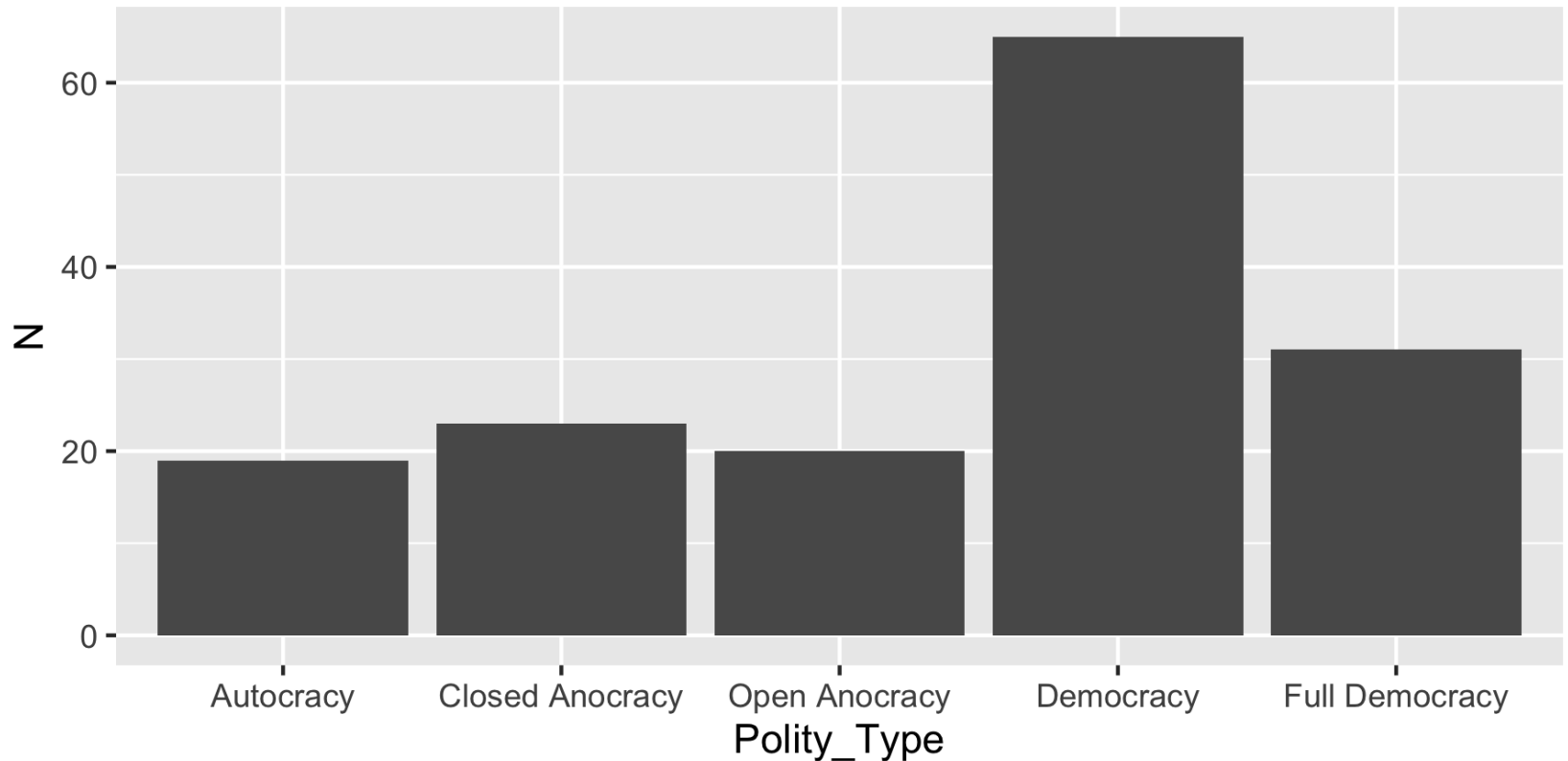
```
df3 <- df %>%  
  drop_na(Polity_Type) %>%  
  mutate(Polity_Type = factor(Polity_Type,  
                               levels = c("Autocracy", "Closed Anocracy",  
                                           "Open Anocracy", "Democracy",  
                                           "Full Democracy"))) %>%  
  
  group_by(Polity_Type) %>%  
  summarise(N = n())
```

df3

```
## # A tibble: 5 x 2  
##   Polity_Type      N  
##   <fct>          <int>  
## 1 Autocracy      19  
## 2 Closed Anocracy 23  
## 3 Open Anocracy  20  
## 4 Democracy     65  
## 5 Full Democracy  31
```

# 作図

```
df3 %>%  
  ggplot() +  
  geom_bar(aes(x = Polity_Type, y = N), stat = "identity")
```



# 棒が持つ情報

- 政治体制のタイプ ( $x = \text{Polity\_Type}$ )
- 国家数 ( $y = N$ )

例) さらに大陸の情報を持たせ、色分けしたい

- 大陸の列が必要

```
df4 <- df %>%  
  drop_na(Polity_Type) %>%  
  mutate(Polity_Type = factor(Polity_Type,  
                              levels = c("Autocracy", "Closed Anocracy",  
                                          "Open Anocracy", "Democracy",  
                                          "Full Democracy"))) %>%  
  
  group_by(Polity_Type, Continent) %>%  
  summarise(N = n(),  
            .groups = "drop")
```

# データ

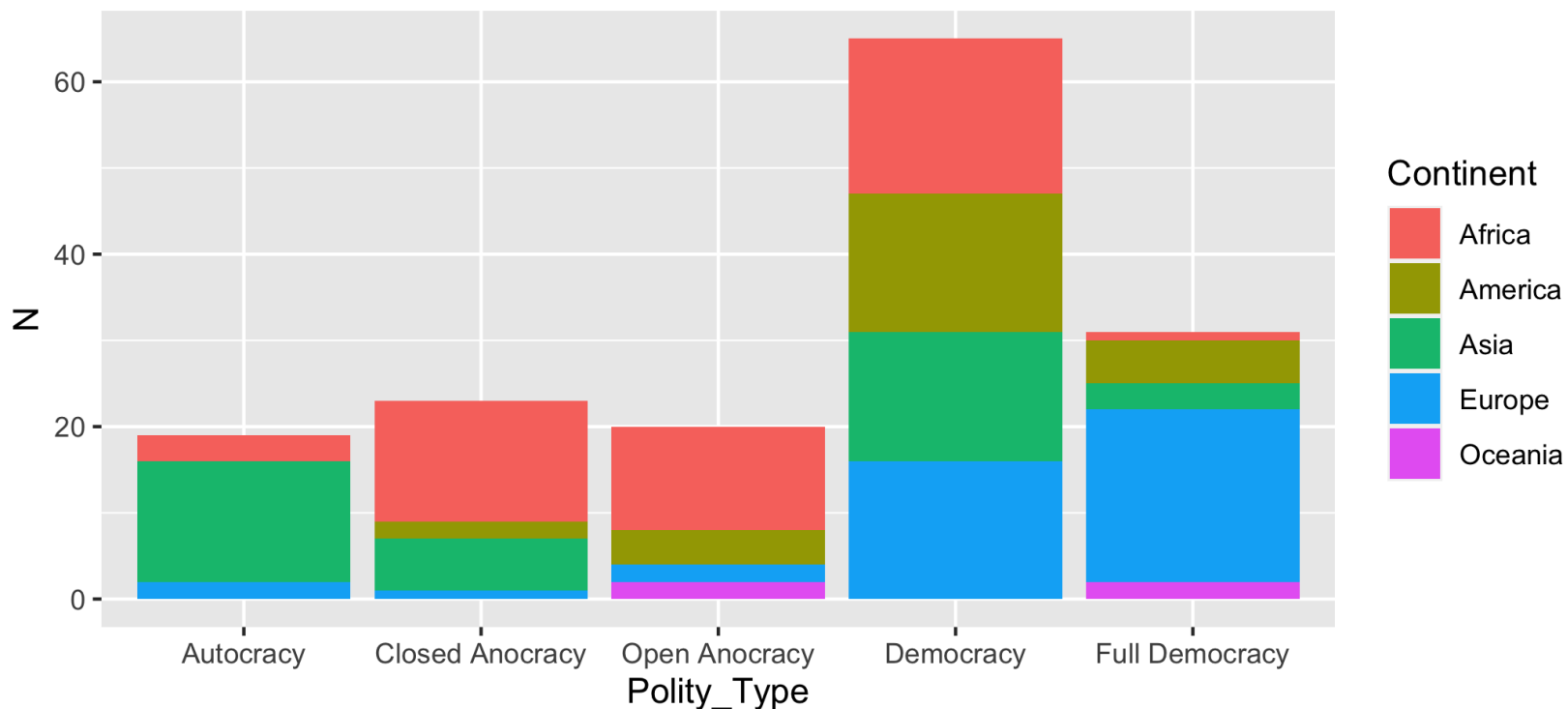
```
df4
```

```
## # A tibble: 20 x 3
##   Polity_Type      Continent      N
##   <fct>          <chr>    <int>
## 1 Autocracy      Africa        3
## 2 Autocracy      Asia         14
## 3 Autocracy      Europe         2
## 4 Closed Anocracy Africa         14
## 5 Closed Anocracy America          2
## 6 Closed Anocracy Asia             6
## 7 Closed Anocracy Europe           1
## 8 Open Anocracy   Africa         12
## 9 Open Anocracy   America          4
## 10 Open Anocracy   Europe           2
## 11 Open Anocracy   Oceania          2
## 12 Democracy      Africa         18
## 13 Democracy      America         16
## 14 Democracy      Asia            15
## 15 Democracy      Europe         16
## 16 Full Democracy Africa            1
```

# 次元の追加

`aes()` 内に `fill = Continent` を追加

```
df4 %>%  
  ggplot() +  
  geom_bar(aes(x = Polity_Type, y = N, fill = Continent),  
    stat = "identity")
```

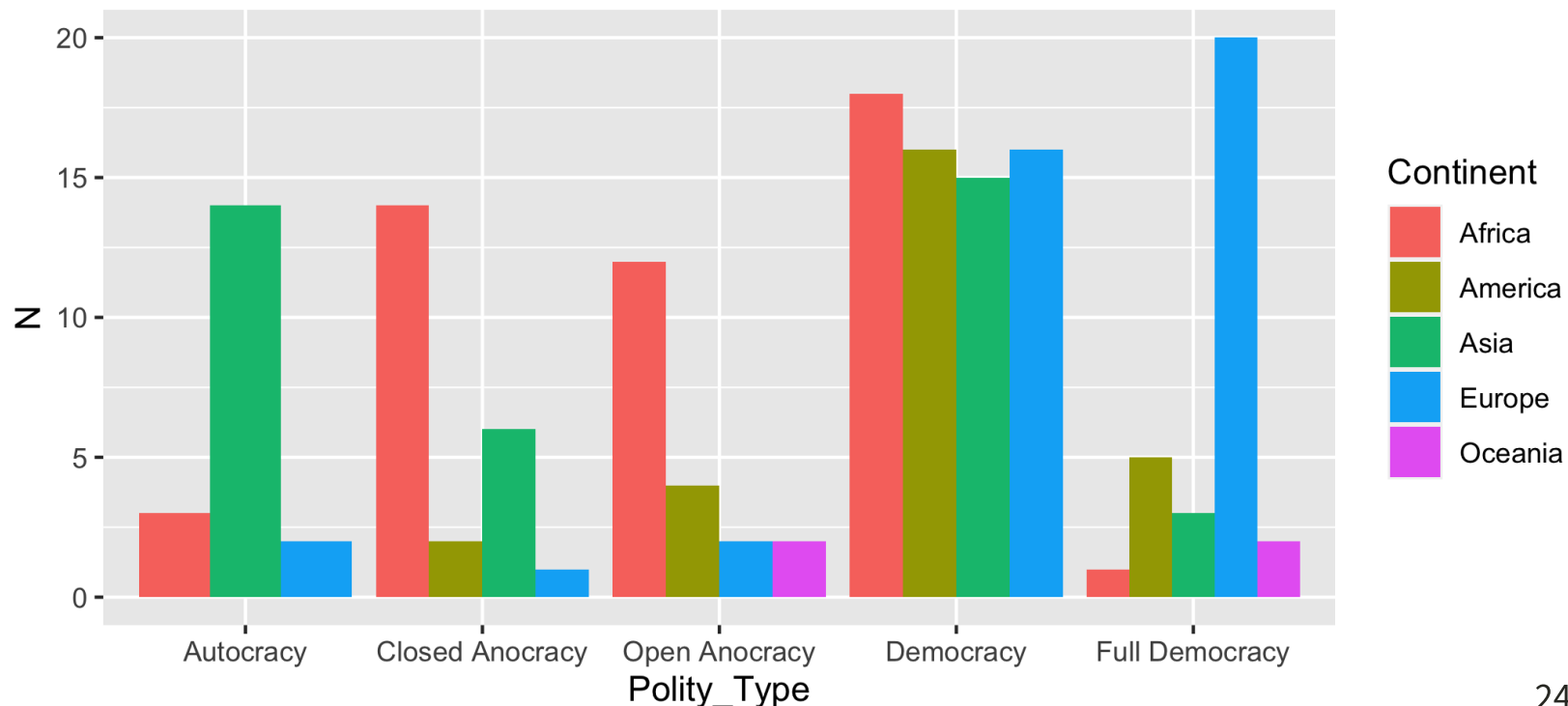


# 棒の位置をずらす

`geom_bar()` 内に `position = "dodge"` を指定

- `"dodge"` の代わりに `position_dodge2()` もOK

```
ggplot(df4) +  
  geom_bar(aes(x = Polity_Type, y = N, fill = Continent),  
    stat = "identity", position = "dodge")
```



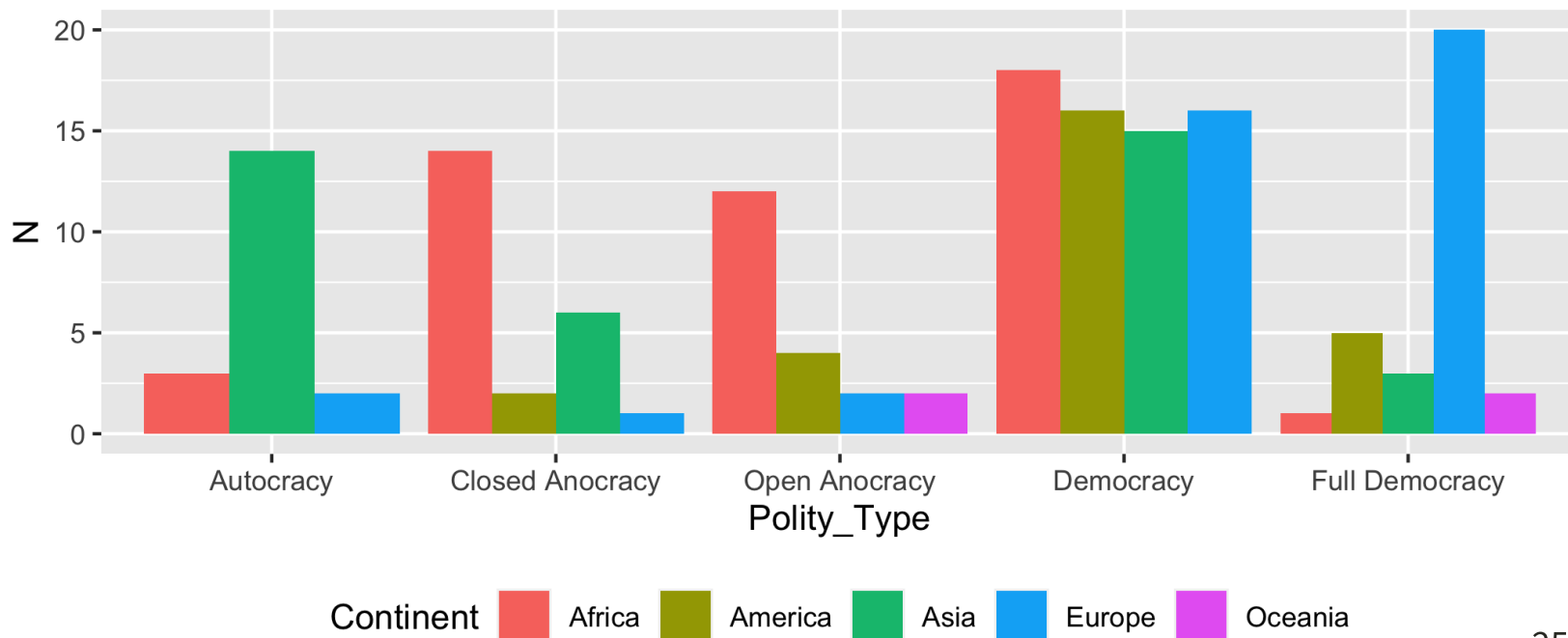


# 凡例の位置調整

theme() 内に legend.position = "bottom" を指定

- デフォルトは "right"; "top" は上段; "none" は削除

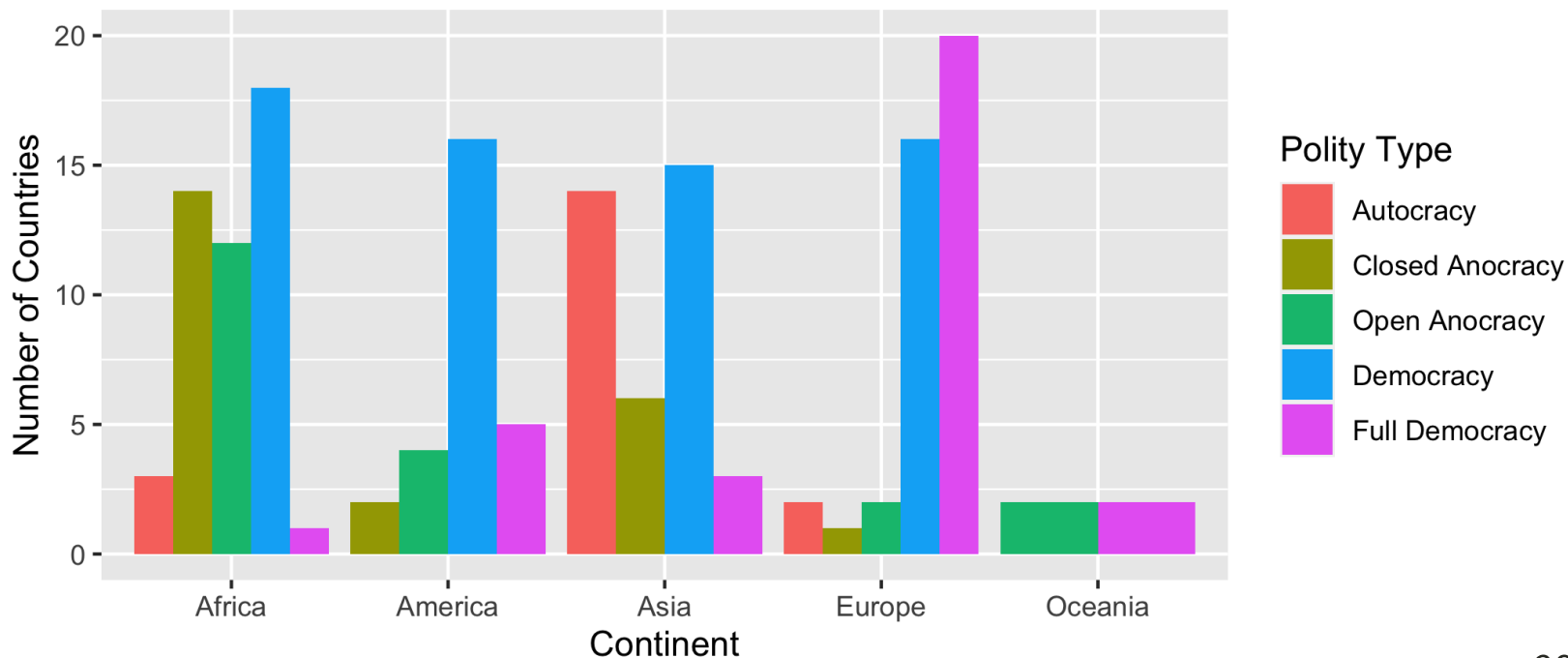
```
ggplot(df4) +  
  geom_bar(aes(x = Polity_Type, y = N, fill = Continent),  
    stat = "identity", position = "dodge") +  
  theme(legend.position = "bottom")
```



# もう一つの方法: マッピングの交換

- 前ページの場合、「ある政治体制内の大陸の分布」を知ることに特化
- 「ある大陸内の政治体制の分布」を見るには? → `x` と `fill` を交換

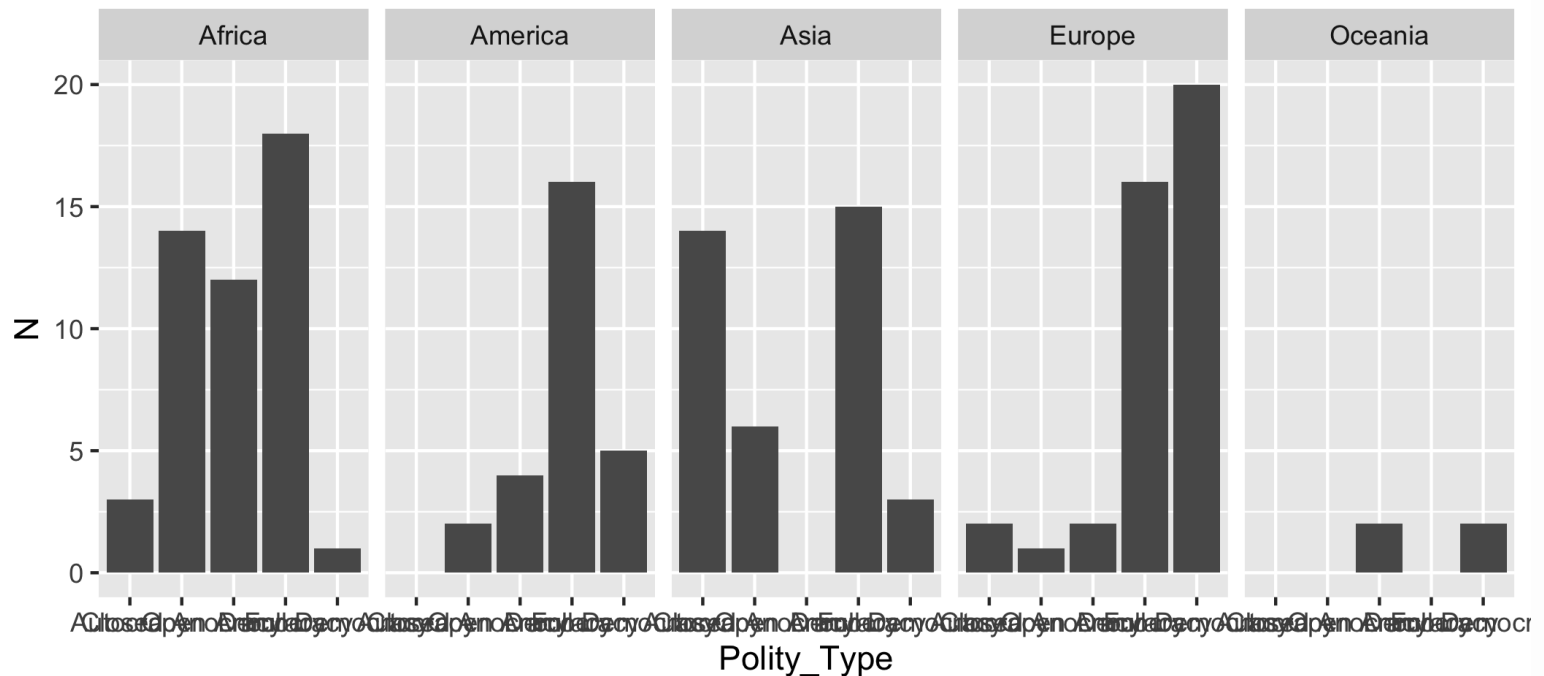
```
ggplot(df4) +  
  geom_bar(aes(x = Continent, y = N, fill = Polity_Type),  
    stat = "identity", position = "dodge") +  
  labs(x = "Continent", y = "Number of Countries", fill = "Polity Type")
```



# もう一つの方法: ファセット分割

- 色分けを出来る限り抑えたい
- `facet_wrap(~分割の基準となる変数名)`

```
Fig1 <- ggplot(df4) +  
  geom_bar(aes(x = Polity_Type, y = N), stat = "identity") +  
  facet_wrap(~Continent, ncol = 5) # ncol (nrow)で列 (行)数の指定が可能  
print(Fig1)
```

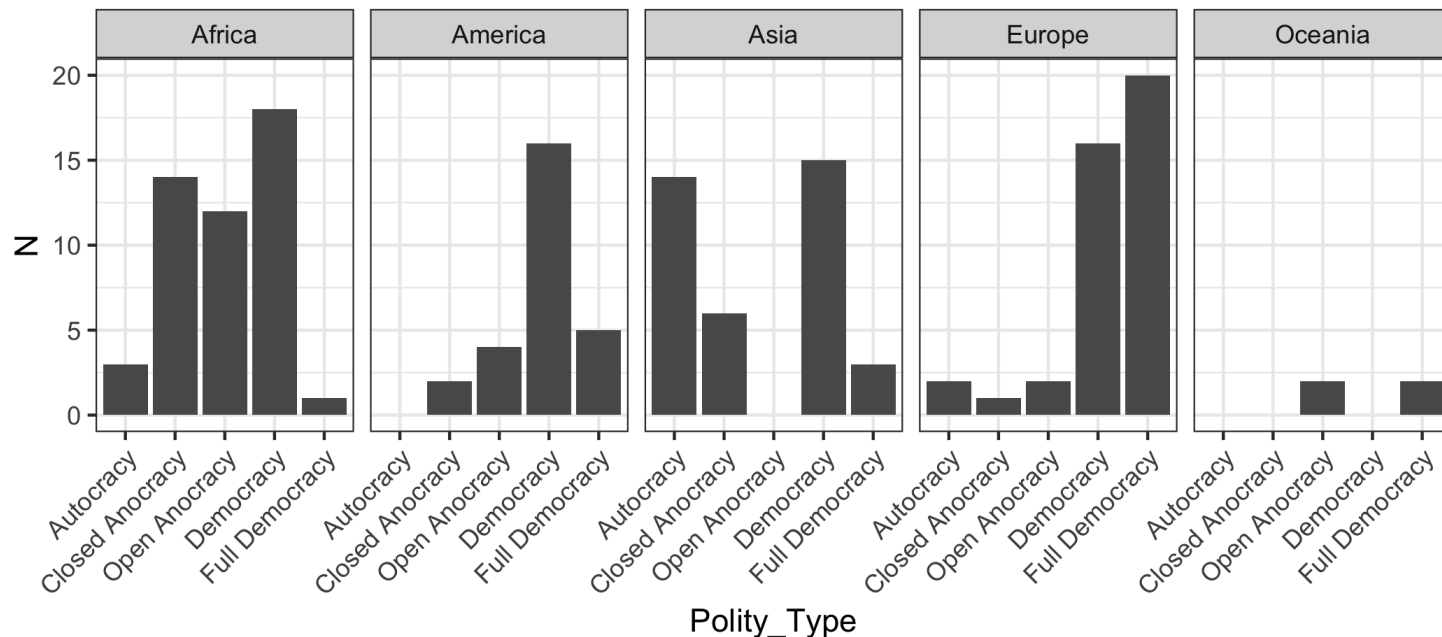


# 値ラベルの回転

値ラベルが長すぎる場合、ラベルを回転することで重複を避ける

- 覚える必要はなく、必要に応じてググる (`theme()` レイヤーはかなり複雑)

```
Fig2 <- Fig1 +  
  theme_bw() + # テーマを変えてみよう  
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))  
  
print(Fig2)
```



# 図の保存

---

# RStudioから保存する方法

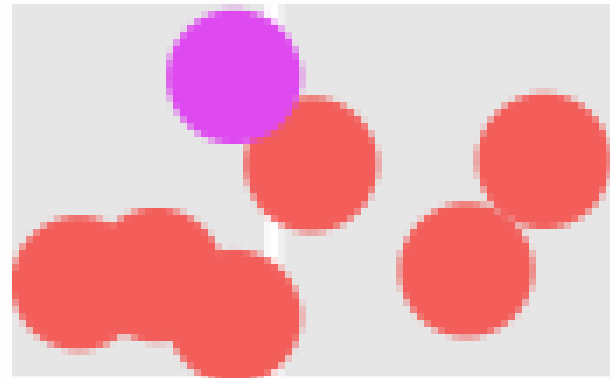
RStudio内のPlotsペインに図を出力し、Plotsペイン上段の「Export」をクリック

- 文字化けの可能性があるため、非推奨
- **PDF**形式で保存する場合: 「Save as PDF...」
  - 拡大しても図が綺麗なまま
  - 複雑な図であれば、ファイルのサイズが大きくなる
- その他の形式（ビットマップ）で保存する場合: 「Save as Image...」
  - 推奨は**PNG**形式（DPI = 300）
  - 拡大すると図がカクカクする（DPI調整で対応可; ファイルサイズに注意）
  - 図が複雑でも、ファイルサイズが安定

PDF形式



PNG形式



# RStudioで図の保存 (PDF)

1. RStudio内のPlotsペインに図を出力し、Plotsペイン上段の「Export」をクリック
2. 「Save as PDF...」をクリック
3. 詳細設定
  - PDF Size: 図のサイズ指定 (インチ)
  - Orientation: Portrait (長辺が縦)、Landscape (長辺が横)
  - Use cairo\_pdf device: **必ずチェックする**
  - Directory: 保存するフォルダ指定
  - File Name: ファイル名を指定
4. 「Save」をクリック
5. 保存された図を必ず確認すること
  - PCの設定によっては日本語が表示されない (豆腐化) 場合もあるため

# RStudioで図の保存 (PNG)

1. RStudio内のPlotsペインに図を出力し、Plotsペイン上段の「Export」をクリック
2. 「Save as Image...」をクリック
3. 詳細設定
  - Directory: 保存するフォルダ指定
  - File Name: ファイル名を指定
  - Width / Height: 図のサイズ指定（ピクセル単位）
4. 「Save」をクリック
5. 保存された図を必ず確認すること
  - PCの設定によっては日本語が表示されない（豆腐化）場合もあるため



# コマンドで保存する方法 (macOS)

## Fig2 を保存する例

- 作業フォルダ内のFigsフォルダにFigure1.pdfという名で保存
- 図のサイズは幅6インチ、高さ3インチ
- PNGの場合、DPIが高いほど鮮明（印刷用の場合、300以上推奨）

## PDF形式

```
quartz(type = "pdf", file = "Figs/Figure1.pdf", width = 6, height = 3)
print(Fig2)
dev.off()
```

## PNG形式 (DPI = 300)

```
ggsave(filename = "Figs/Figure1.png", plot = Fig2,
        width = 6, height = 3, dpi = 300)
```

# コマンドで保存する方法 (Windows)

## Fig2 を保存する例

- 作業フォルダ内のFigsフォルダにFigure1.pdf (.png)という名で保存
- 図のサイズは幅6インチ、高さ3インチ
- PNGの場合、DPIが高いほど鮮明（印刷用の場合、300以上推奨）

## PDF形式

*# 以下のコードはmacOSでも作動する問題なく使える場合もある*

```
ggsave(filename = "Figs/Figure1.pdf", plot = Fig2, device = cairo_pdf,  
        width = 6, height = 3)
```

## PNG形式 (DPI = 300)

```
ggsave(filename = "Figs/Figure1.png", plot = Fig2,  
        width = 6, height = 3, dpi = 300)
```

# コマンドで保存する方法 (クラウド版)

## Fig2 を保存する例

- 作業フォルダ内のFigsフォルダにFigure1.pdfという名で保存
- 図のサイズは幅6インチ、高さ3インチ
- PNGの場合、DPIが高いほど鮮明（印刷用の場合、300以上推奨）

```
ggsave(filename = "Figs/Figure1.pdf", plot = Fig2, device = cairo_pdf,  
        width = 6, height = 3)
```

## PNG形式 (DPI = 300)

```
ggsave(filename = "Figs/Figure1.png", plot = Fig2,  
        width = 6, height = 3, dpi = 300)
```

# 注意事項

英語のみで構成された図の場合、文字化けが生じる可能性は低い

- ただし、日本語の場合は注意が必要
- 同じコマンドでも使用するPC環境によって文字化けが生じる場合も
  - macOSローカル版でも `ggsave()` を使えるが、日本語が出力されない場合がある。
- 保存した図で文字化けが生じた場合は、宋に相談
  - ただし、期末テストで図の保存は出題されない
- 参考記事
  - [片桐智志さんの記事「おまえはもうRのグラフの日本語表示に悩まない \(各OS対応\)」](#)

# ヒストグラム

---

# 変数の分布を確認する方法

## 変数が一つの場合

- 変数が**離散変数**の場合: 棒グラフ
  - 性別、国、都道府県など、数値が意味を持たない変数
  - 順位など取りうる値が有限
- 変数が**連続変数**の場合: ヒストグラム、箱ひげ図
  - 気温、成績、所得、身長、体重、人間開発指数、...
  - 取りうる値が無数

## 変数が2つの場合

- **連続変数**と**連続変数**: 散布図
- **順序付き離散変数**と**連続変数**: 折れ線グラフ
- **離散変数**と**離散変数**: モザイク図

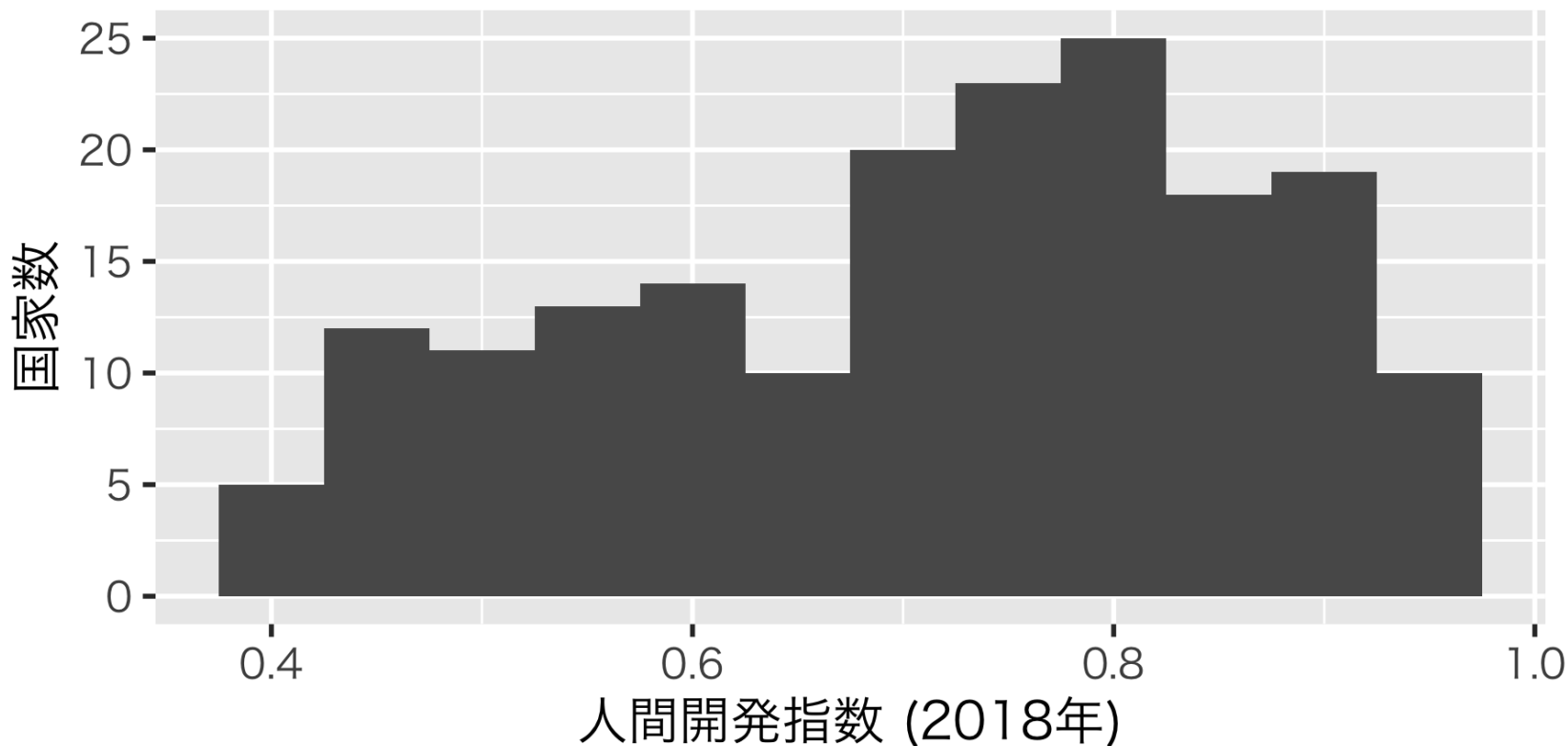
## 変数が3つ以上の場合

- 次元を追加する形で対応

# ヒストグラムの棒が持つ情報

棒の横軸上の位置と高さ

- {ggplot2}の場合、ヒストグラムを出力する変数を `x` にマッピングするだけで、自動的にヒストグラムを生成

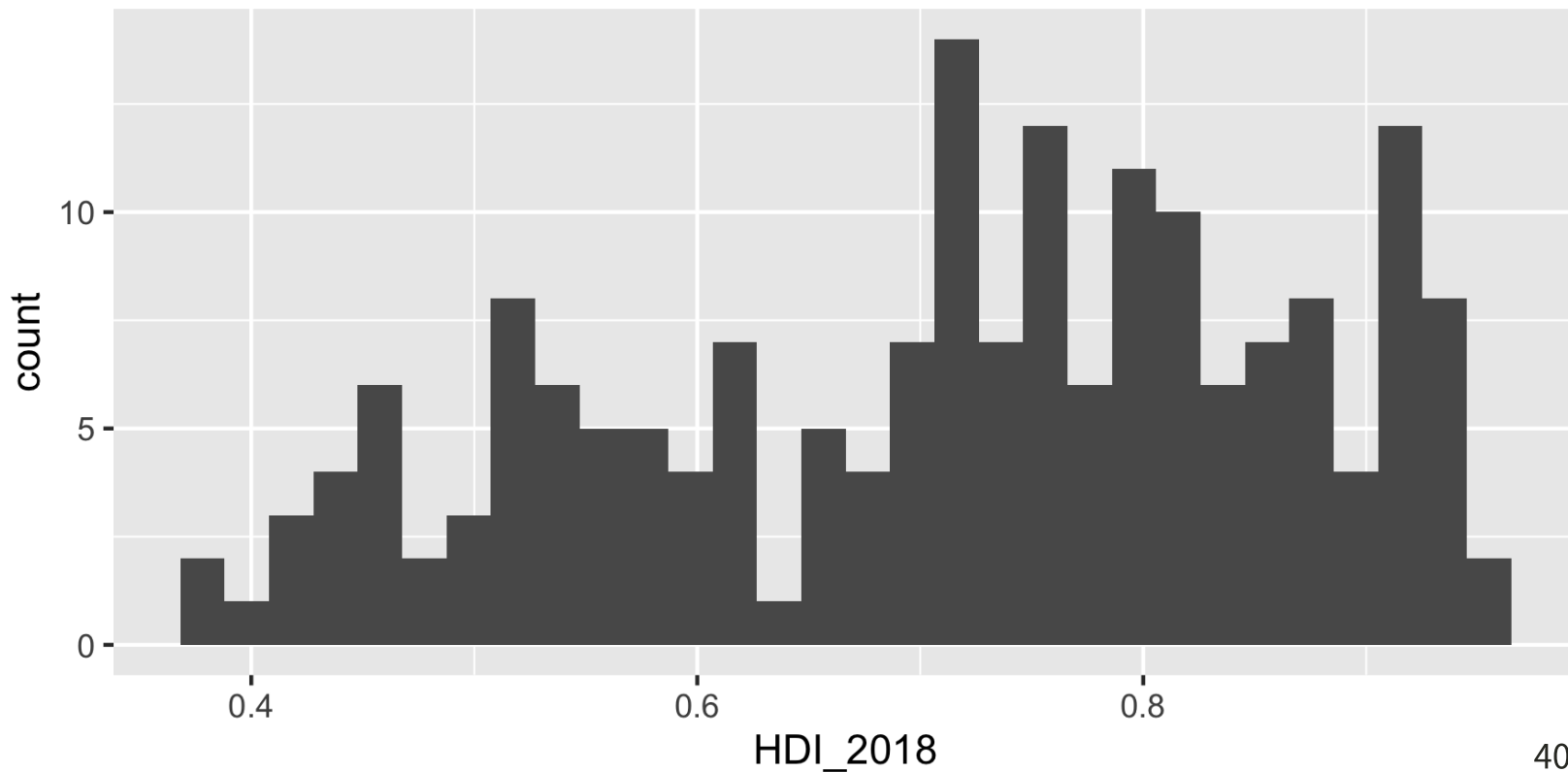


# ヒストグラムの作成

`geom_histogram()` を使用: マッピングは `x` のみ

```
ggplot(df) +  
  geom_histogram(aes(x = HDI_2018)) # HDI_2018のヒストグラム
```

## ``stat_bin()`` using ``bins = 30``. Pick better value with ``binwidth``.



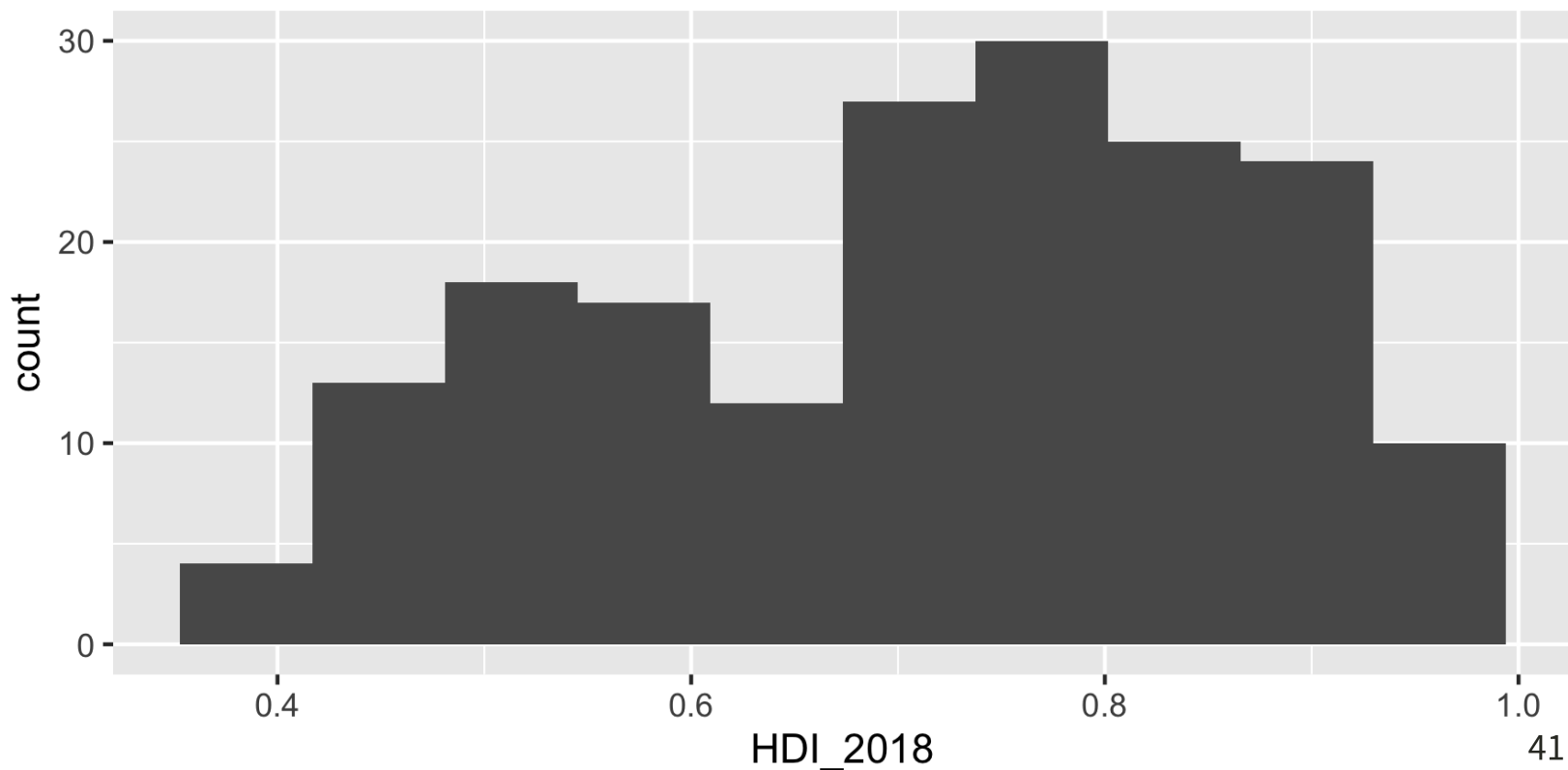


# 棒の幅を調整する (1)

`geom_histogram()` 内に `bins` または `binwidth` 引数を指定

- `bins`: 棒の数

```
ggplot(df) +  
  geom_histogram(aes(x = HDI_2018), bins = 10) # 棒を10個にする
```

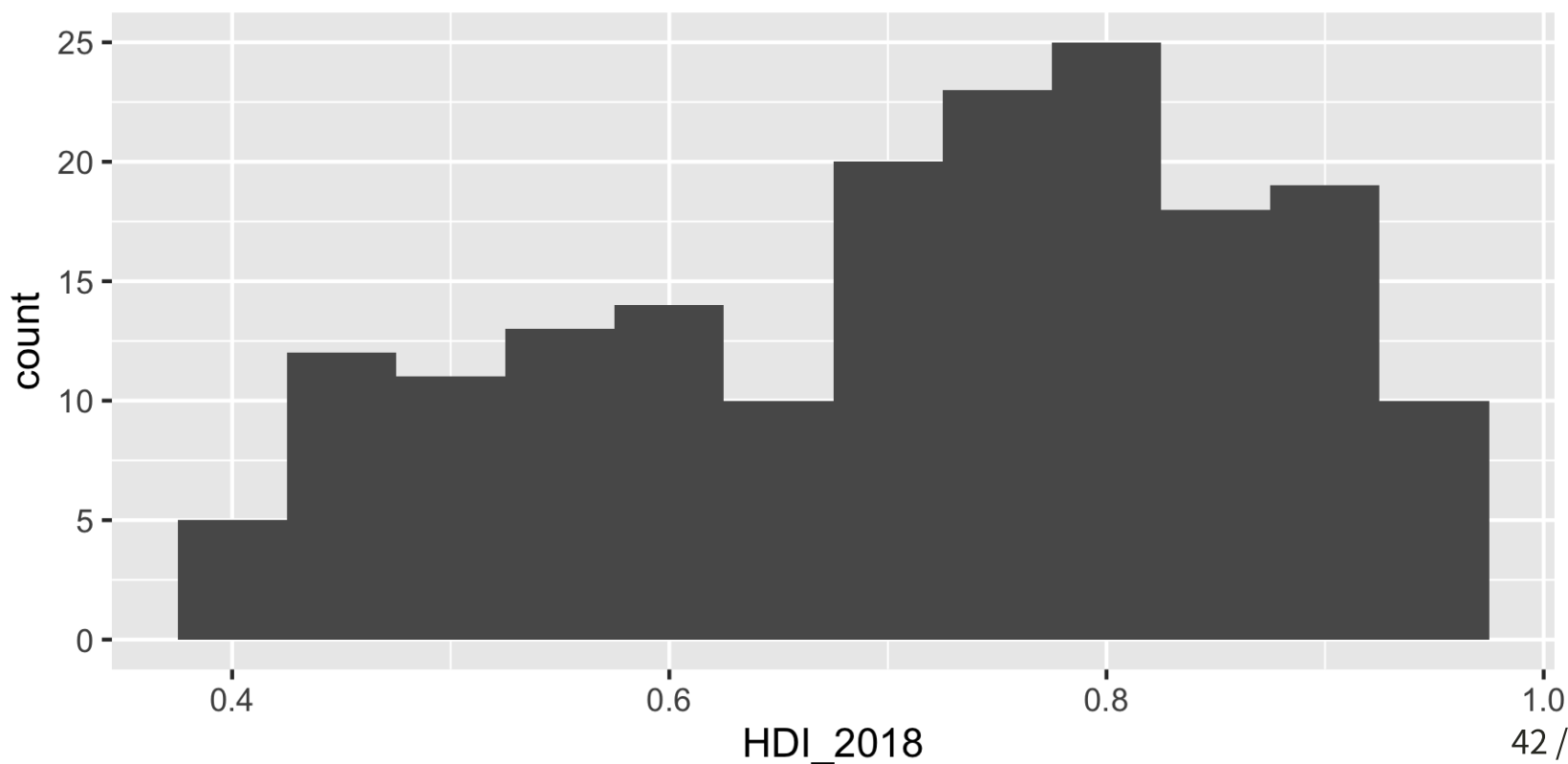


# 棒の幅を調整する (2)

`geom_histogram()` 内に `bins` または `binwidth` 引数を指定

- `binwidth`: 棒の幅

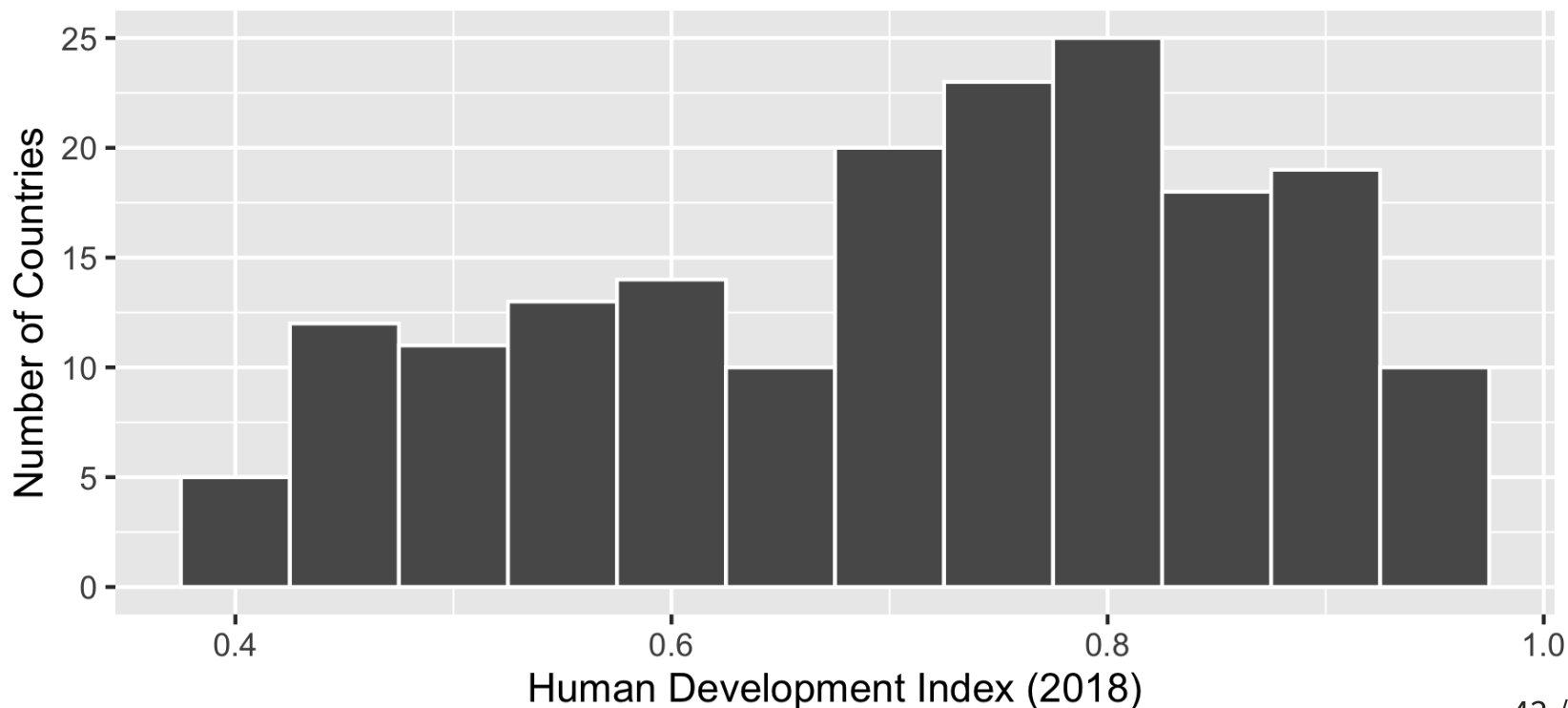
```
ggplot(df) +  
  geom_histogram(aes(x = HDI_2018), binwidth = 0.05) # 棒を0.05単位で刻む
```



# 棒の枠線を入れる

color 引数を指定: ヒストグラムが見やすくなる

```
Fig3 <- ggplot(df) +  
  geom_histogram(aes(x = HDI_2018), binwidth = 0.05, color = "white") +  
  labs(x = "Human Development Index (2018)", y = "Number of Countries")  
print(Fig3)
```



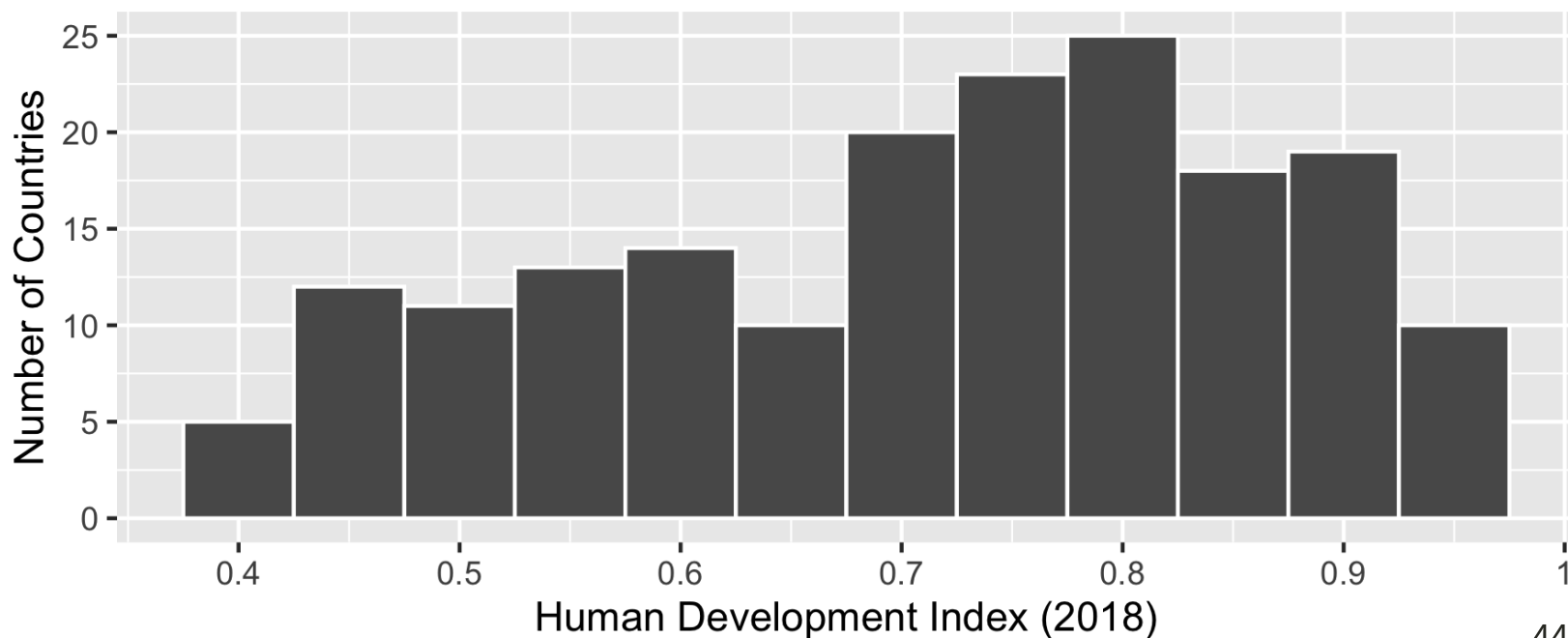
# 横軸のスケール調整

`scale_x_continuous()` を使用 (x を y に変えると縦軸修正)

- `breaks` 引数: 目盛りの位置
- `labels` 引数: 目盛りのラベル

Fig3 +

```
scale_x_continuous(breaks = seq(0.4, 1.0, by = 0.1),  
                  labels = seq(0.4, 1.0, by = 0.1))
```

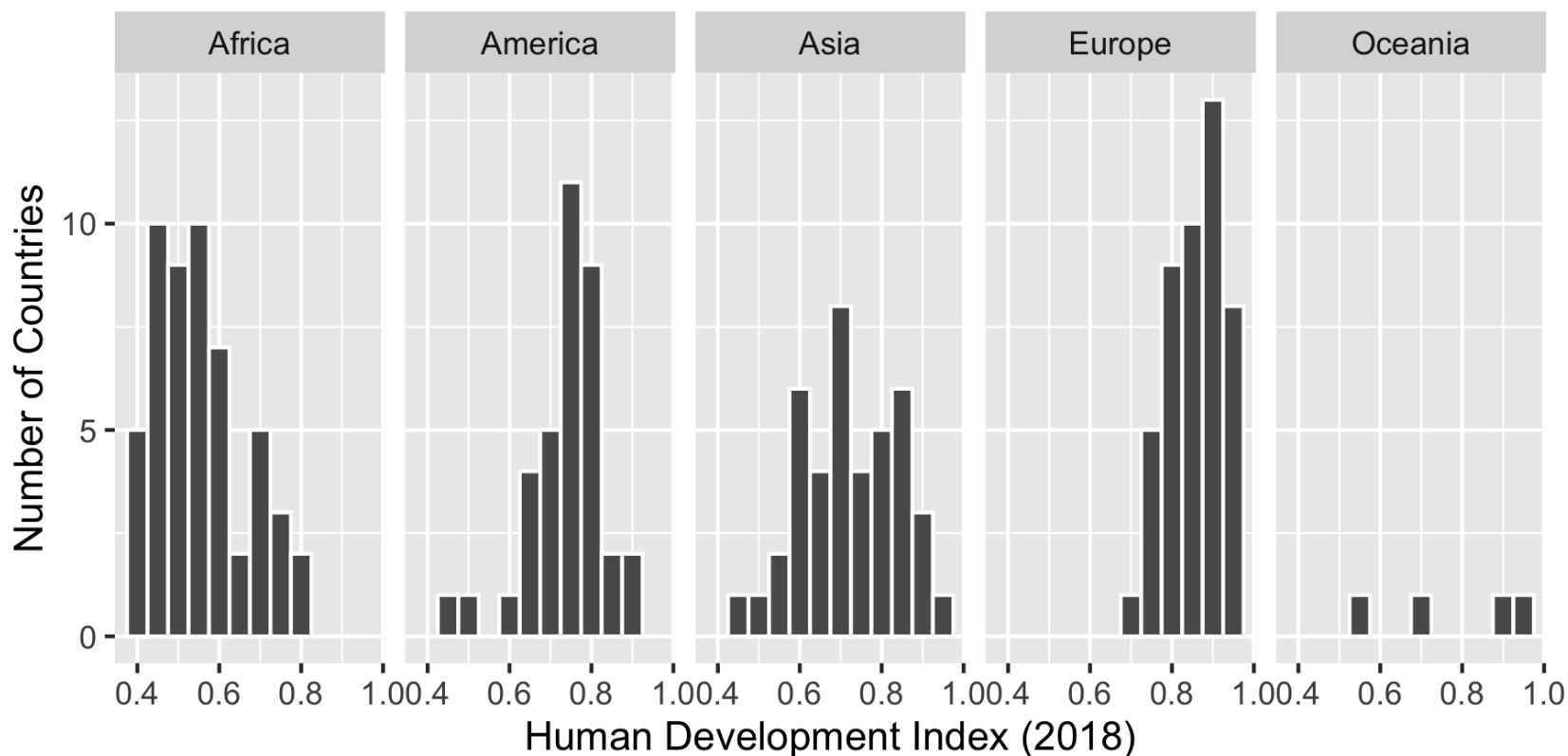


# 次元の追加（ファセット分割）

大陸ごとの HDI\_2018 のヒストグラム: ファセット分割を使用

Fig3 +

```
facet_wrap(~Continent, ncol = 6)
```

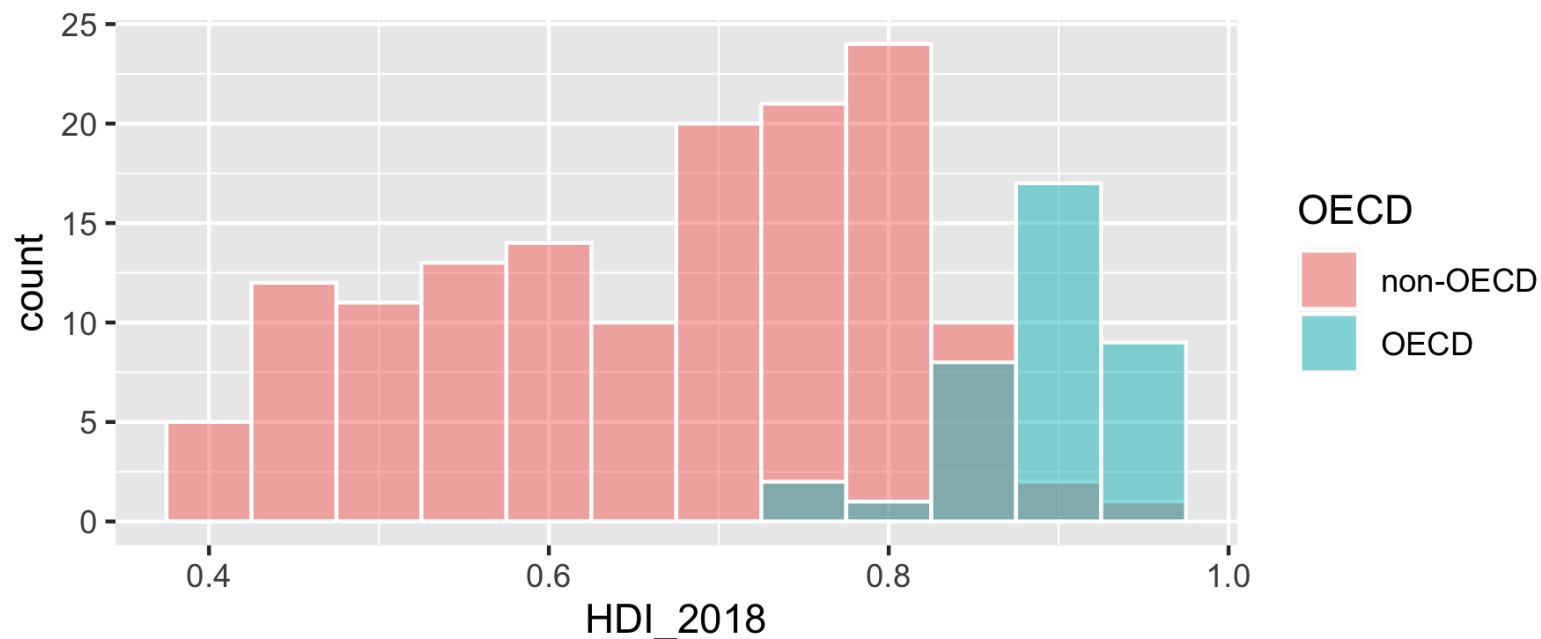


# 次元の追加 (色分け)

`position = "identity"` と `alpha = 0.5` で可能であるが、非推奨

- `alpha = 1` の場合、棒が不透明であるため、0.5程度に調整

```
df %>%  
  mutate(OECD = if_else(OECD == 1, "OECD", "non-OECD")) %>%  
  ggplot() +  
  geom_histogram(aes(x = HDI_2018, fill = OECD), position = "identity",  
                 binwidth = 0.05, color = "white", alpha = 0.5)
```



# 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)

- 棒グラフの作成: 教科書第18.3章
- ヒストグラムの作成: 教科書第18.4章
- 図の保存: 教科書第18.8章
- グラフのカスタマイズ: 教科書第19章



# 課題

1. 今回講義用のプロジェクトを作成する。
2. LMSからデータ（.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. 随時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3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。