

Basics of Data Visualization

Data visualization with R and ggplot2

Jieun Byeon

jbyeon@email.sc.edu

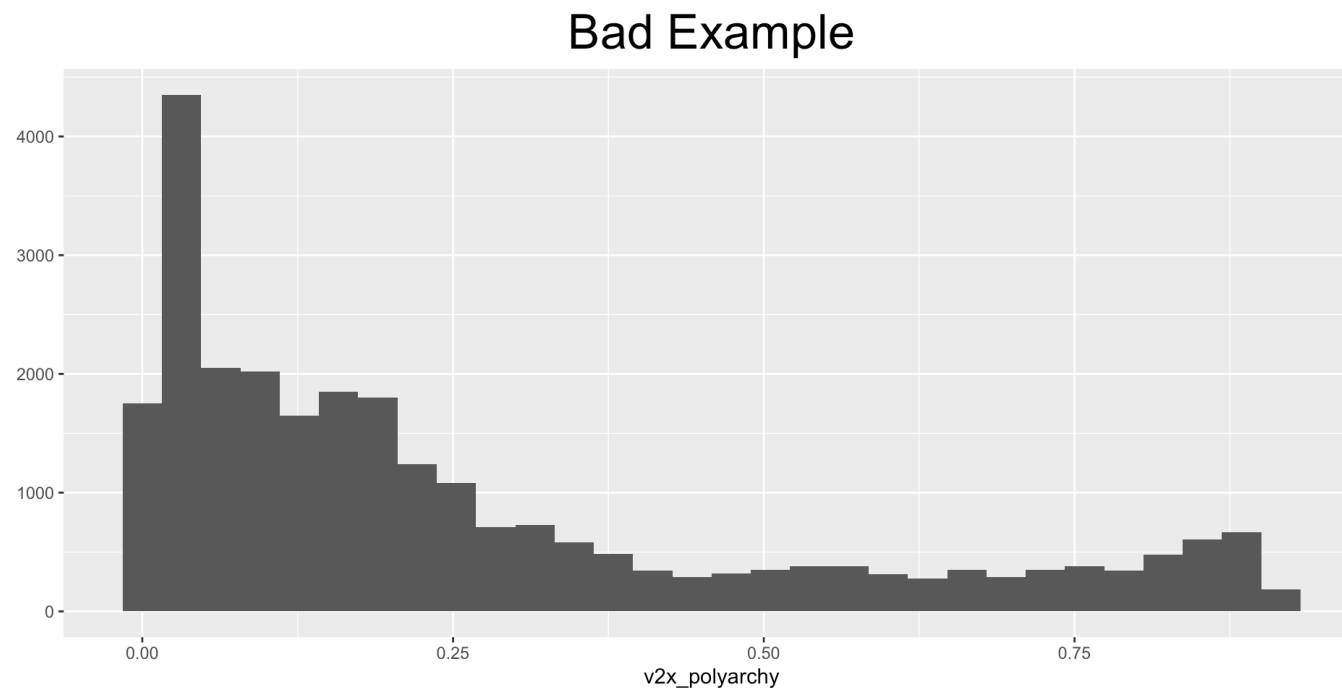
University of South Carolina

Labels

1. Labels

Title, x axis, and y axis

- When creating graphs, the most important thing is to convey to people what information you are showing about the data
 - you should always set the title of the graph and the names of the x-axis and y-axis

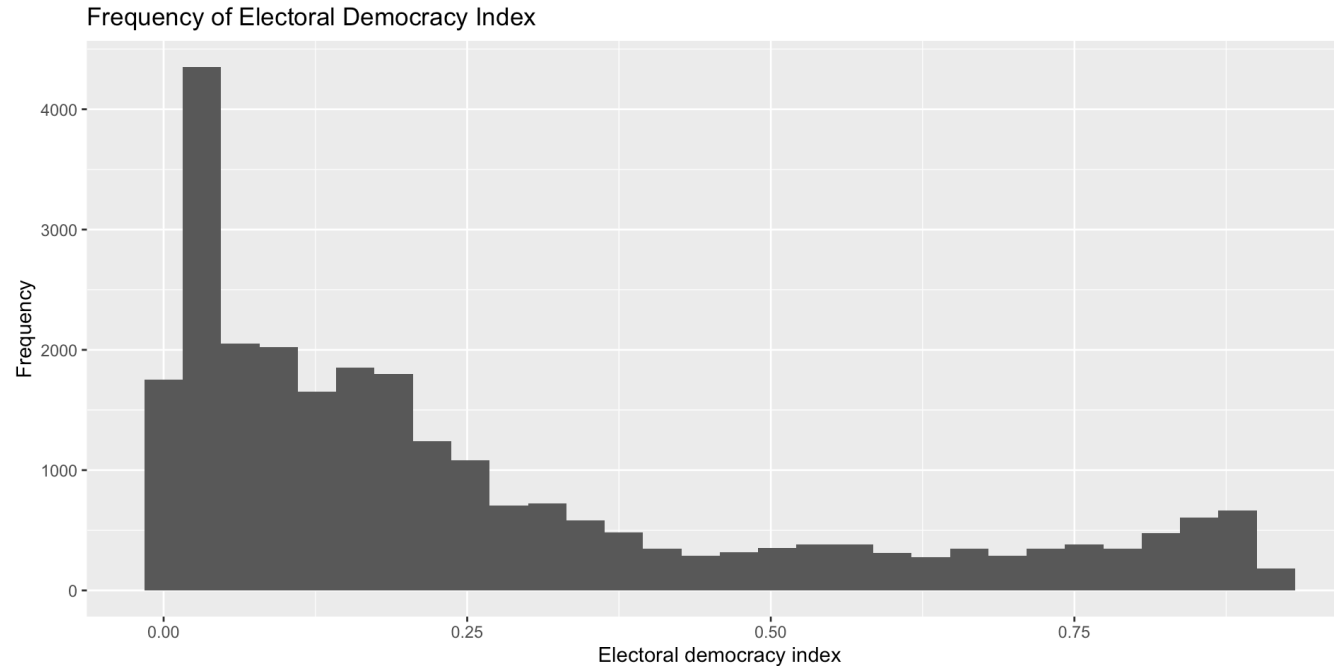


1. Labels

Title, x axis, and y axis

- In ggplot, you can display information about the data using `labs()`

```
1 vdem %>%  
2   ggplot(aes(x = v2x_polyarchy)) +  
3   geom_histogram() +  
4   labs(title = "Frequency of Electoral Democracy Index",  
5         x = "Electoral democracy index", y = "Frequency")
```



1. Labels

Title, x axis, and y axis

- Align your title to the center:
→ `theme(plot.title = element_text(hjust = 0.5))`
- Resize the labels:
→ `element_text(size = )`

```
1 vdem %>%
2   ggplot(aes(x = v2x_polyarchy)) +
3   geom_histogram() +
4   labs(title = "Frequency of Electoral Democracy Index",
5         x = "Electoral democracy index", y = "Frequency") +
6   theme(
7     plot.title = element_text(hjust = 0.5, size = 16),
8     axis.title.x = element_text(size = 14),
9     axis.title.y = element_text(size = 14))
```

1. Labels

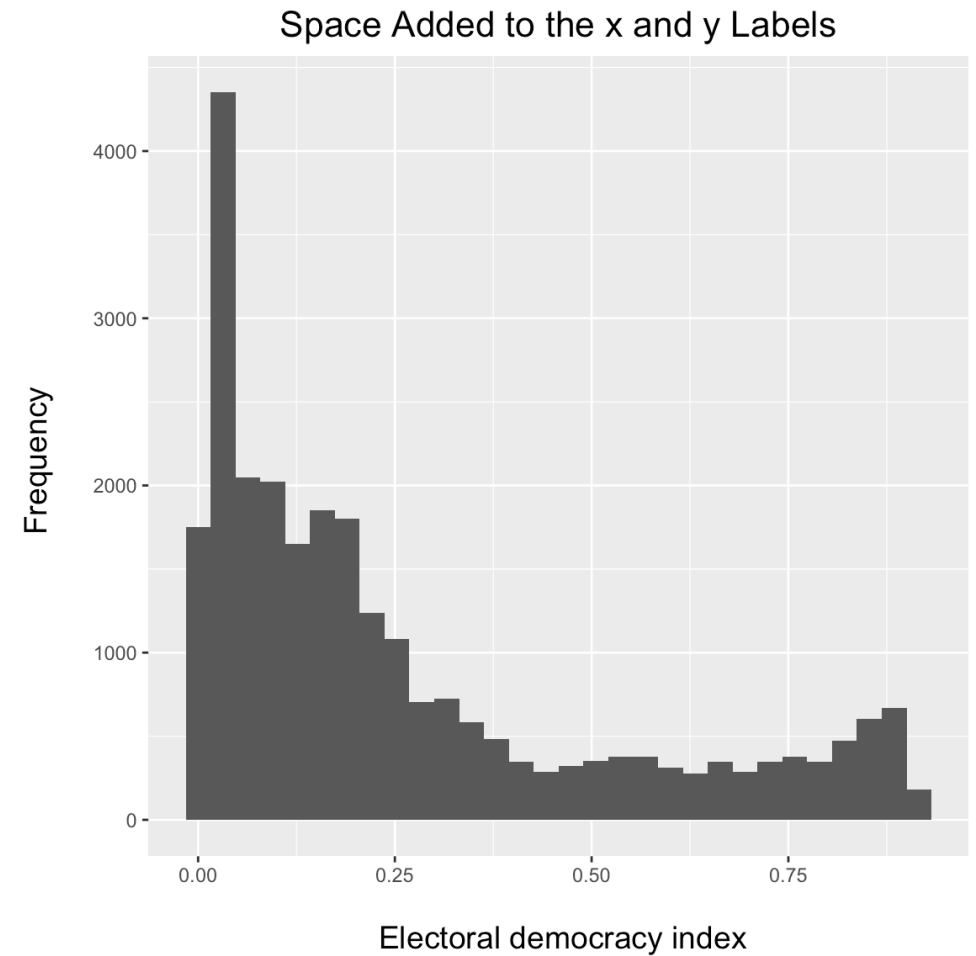
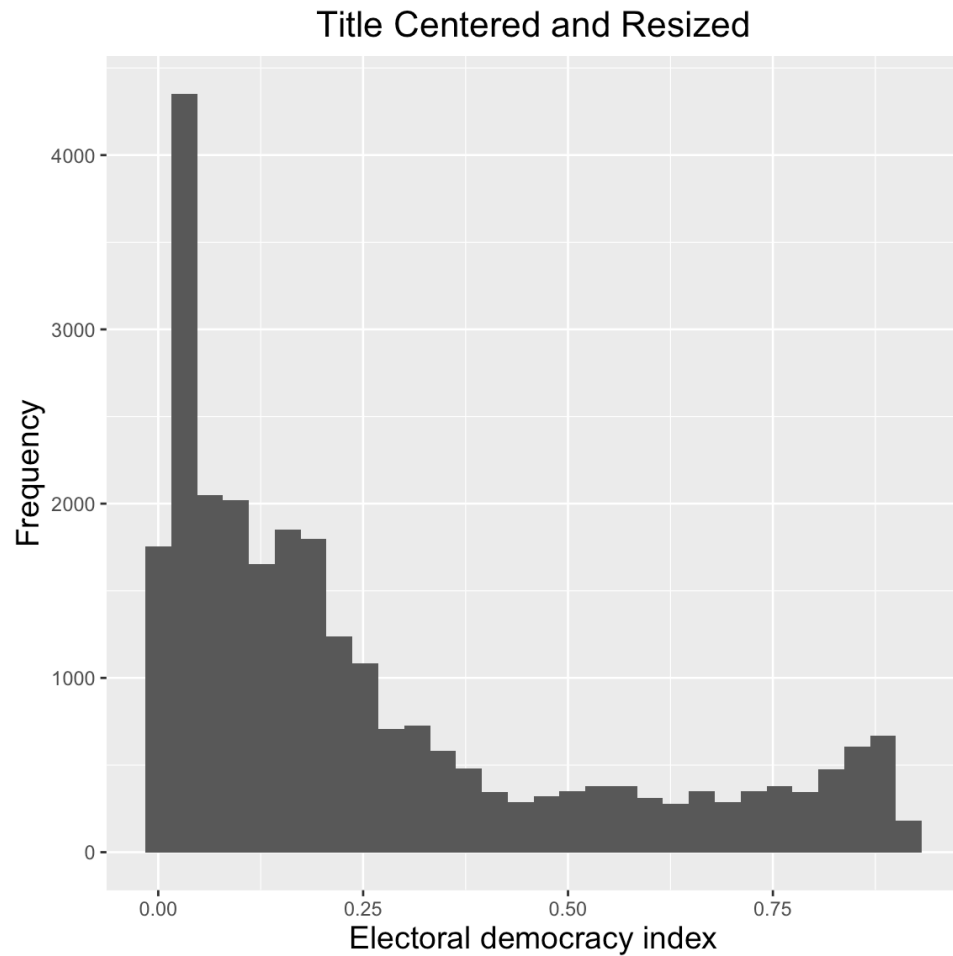
Title, x axis, and y axis

- If the scales and labels are too close together, add `\n` to create some space.

```
1 vdem %>%
2   ggplot(aes(x = v2x_polyarchy)) +
3   geom_histogram() +
4   labs(title = "Frequency of Electoral Democracy Index",
5        x = "\n Electoral democracy index", y = "Frequency \n") +
6   theme(
7     plot.title = element_text(hjust = 0.5, size = 16),
8     axis.title.x = element_text(size = 14),
9     axis.title.y = element_text(size = 14))
```

1. Labels

Title, x axis, and y axis

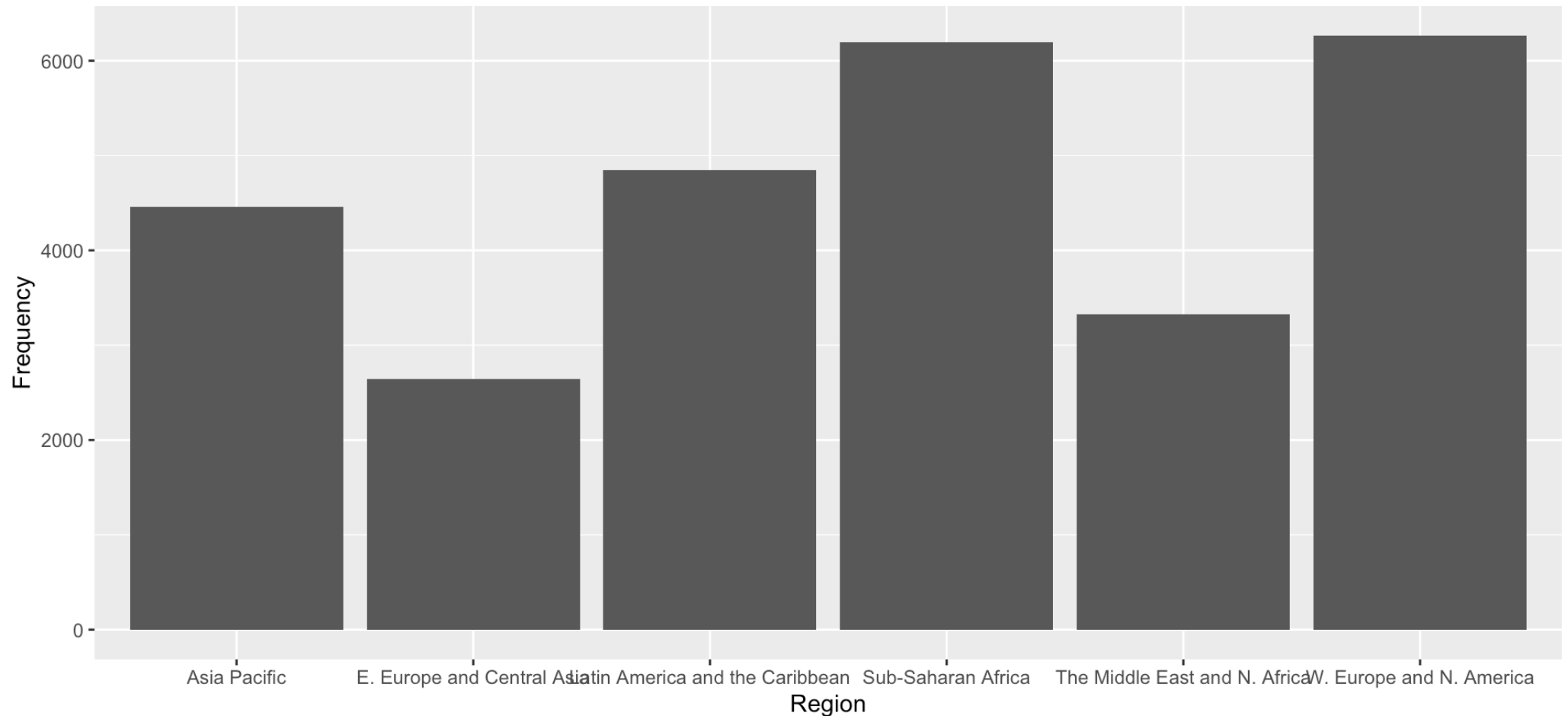


Scales

2. Scales

- “What should I do if the labels are too long and hard to read?”

Bad Example



2. Scales

Ways to deal with long labels

1. Rotate the labels

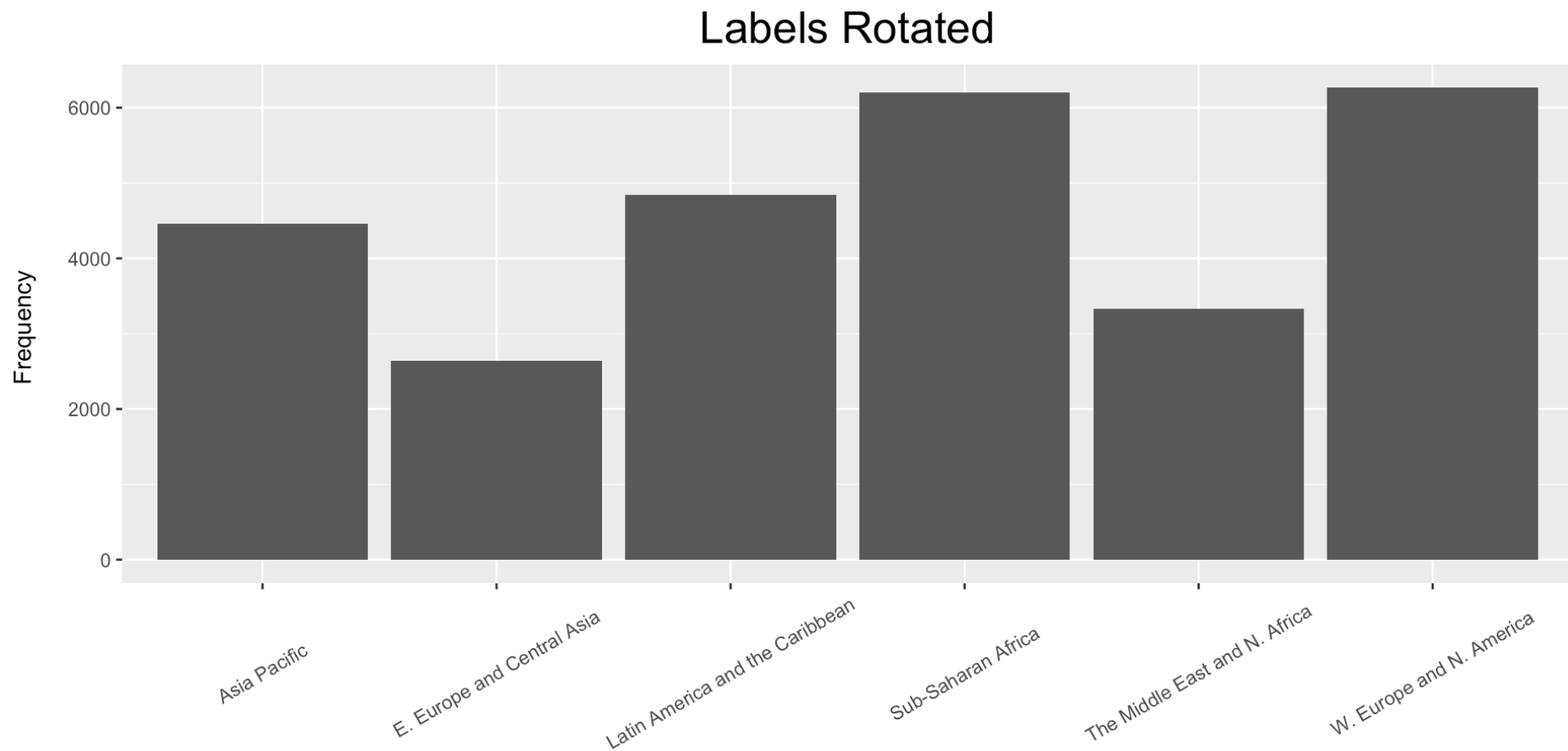
- `element_text(angle = )`: Rotate labels
- `element_text(vjust = )`: Vertical justification

```
1 vdem <- vdem %>%
2   mutate(region = case_when(e_regionpol_6C == 1 ~ "E. Europe and Central Asia",
3                             e_regionpol_6C == 2 ~ "Latin America and the Caribbean",
4                             e_regionpol_6C == 3 ~ "The Middle East and N. Africa",
5                             e_regionpol_6C == 4 ~ "Sub-Saharan Africa",
6                             e_regionpol_6C == 5 ~ "W. Europe and N. America",
7                             e_regionpol_6C == 6 ~ "Asia Pacific"))
8 vdem %>%
9   ggplot(aes(x = region)) +
10  geom_bar() +
11  labs(title = "Labels Rotated", x = "", y = "Frequency \n") +
12  theme(plot.title = element_text(hjust = 0.5, size = 20),
13        axis.text.x = element_text(hjust = 0.5, angle = 30, vjust = 0.5))
```

2. Scales

Ways to deal with long labels

1. Rotate the labels



2. Scales

Ways to deal with long labels

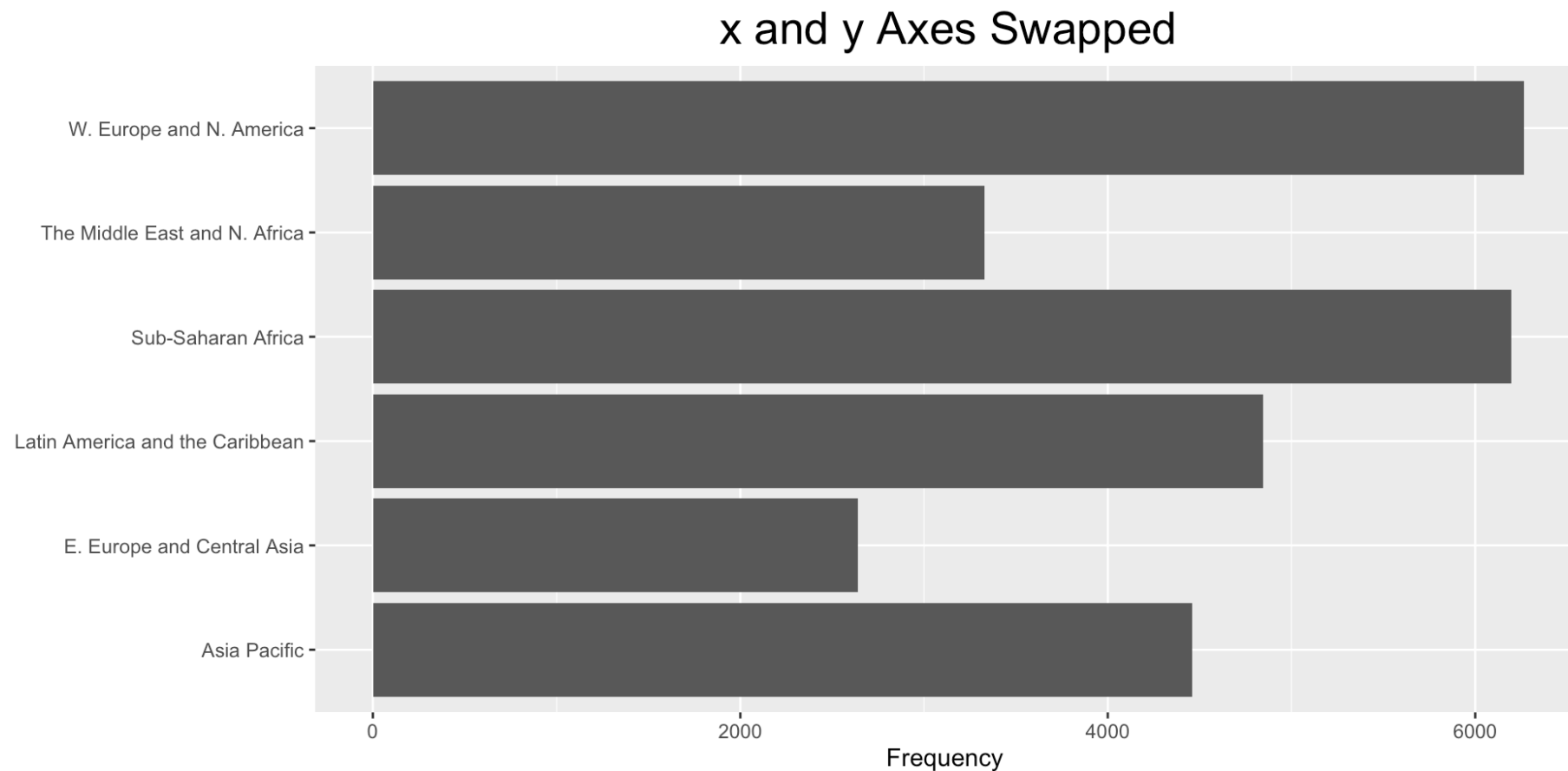
2. Swap the x- and y-axes using `coord_flip()`

```
1 vdem %>%  
2   ggplot(aes(x = region)) +  
3   geom_bar() +  
4   labs(title = "x and y Axes Swapped", x = "", y = "Frequency \n") +  
5   theme(plot.title = element_text(hjust = 0.5, size = 20)) +  
6   coord_flip()
```

2. Scales

Ways to deal with long labels

2. Swap the x- and y-axes using `coord_flip()`



2. Scales

Ways to deal with long labels

3. Add line breaks

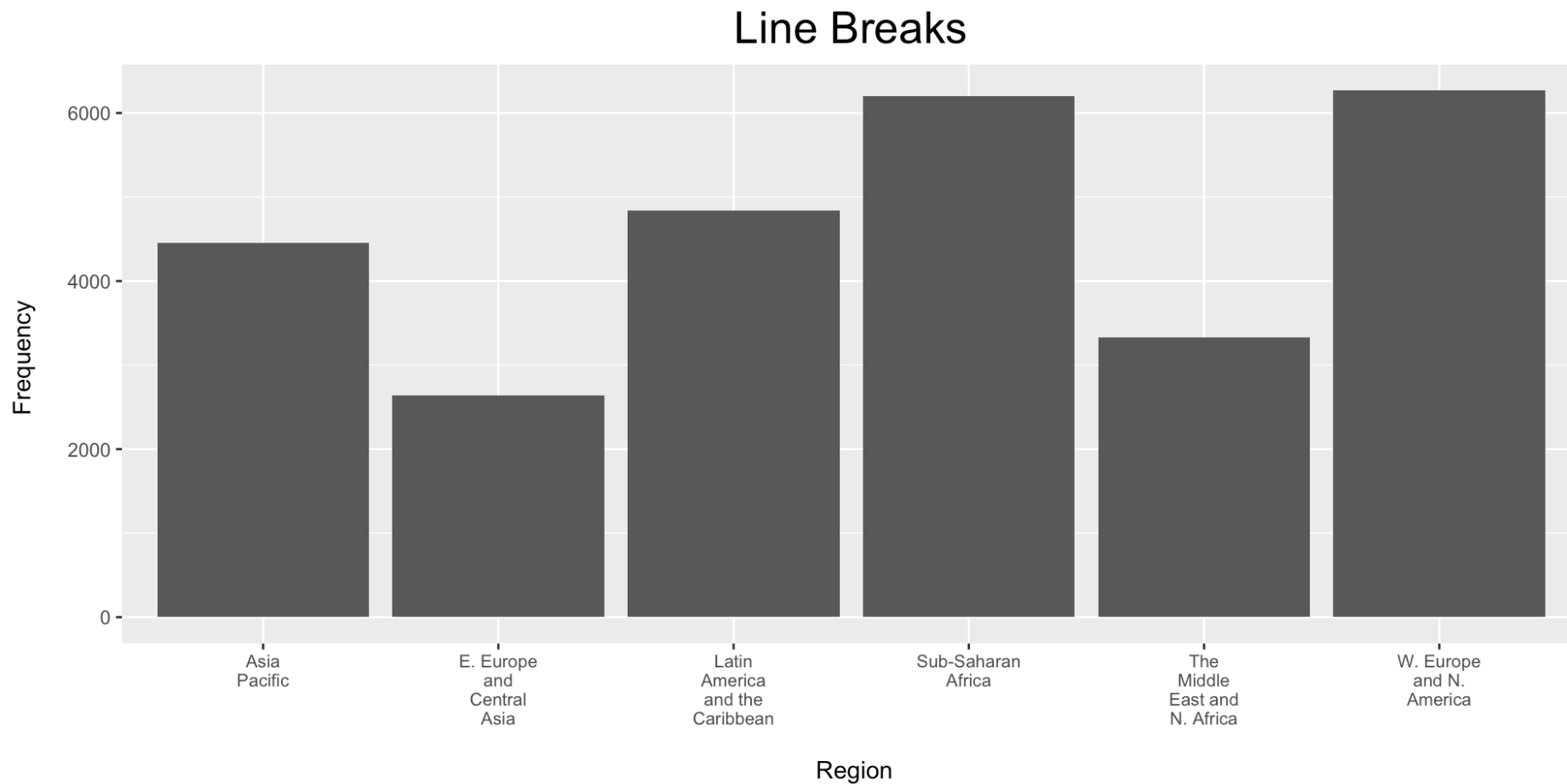
- Use the `label_wrap()` function from the **scales** package

```
1 library(scales)
2 vdem %>%
3   ggplot(aes(x = region)) +
4   geom_bar() +
5   labs(title = "Line Breaks", x = "\n Region", y = "Frequency \n") +
6   scale_x_discrete(labels = label_wrap(10)) +
7   theme(plot.title = element_text(hjust = 0.5),
8         axis.text.x = element_text(hjust = 0.5, size = 8))
```

2. Scales

Ways to deal with long labels

3. Add line breaks

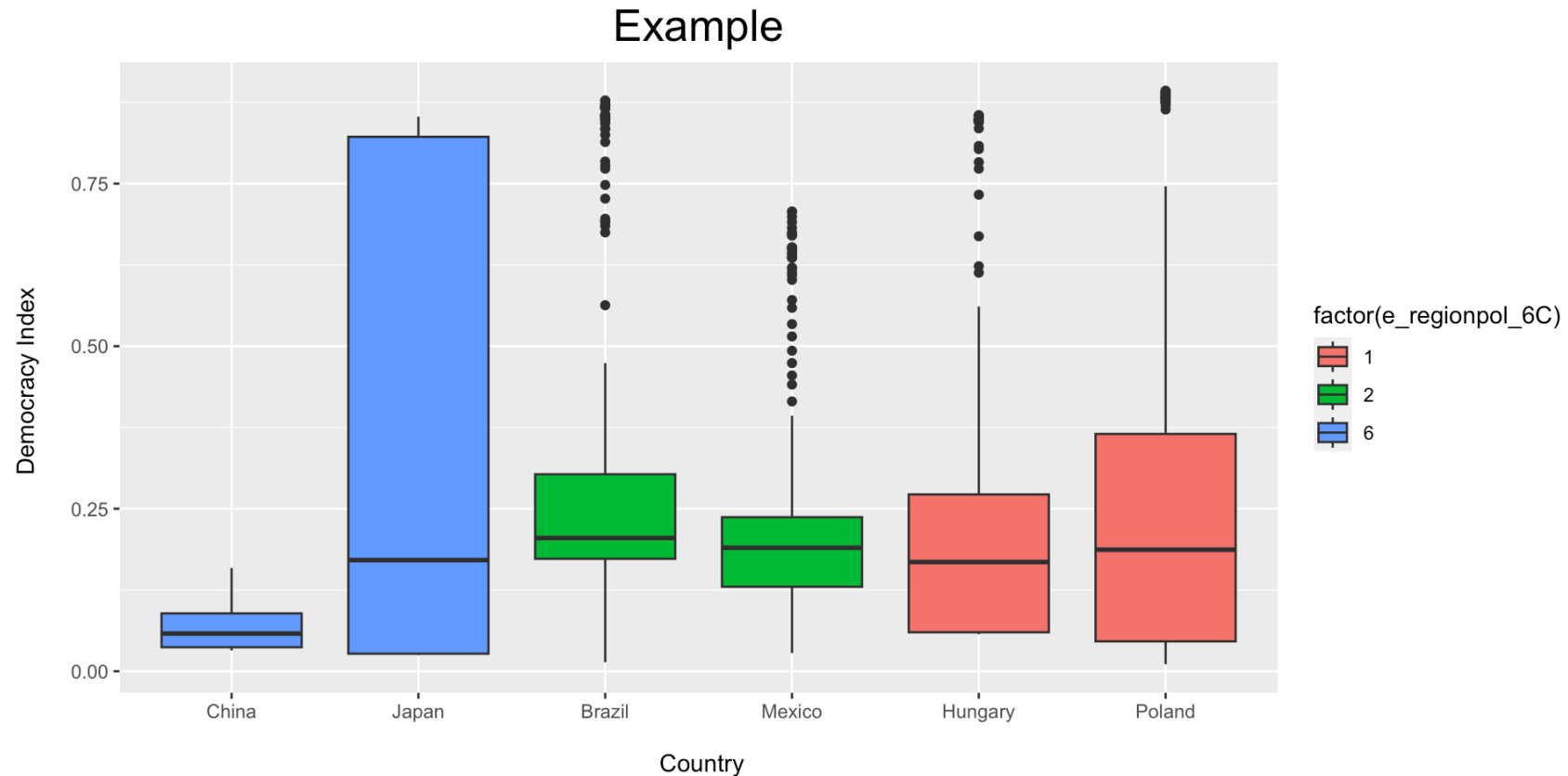


Legends

3. Legends

- ggplot2 will automatically build a legend on your chart as soon as a shape feature is mapped to a variable in `aes()` part

```
1 ggplot(aes(x = country_name, y = v2x_polyarchy,  
2           fill = factor(e_regionpol_6C)))
```



3. Legends

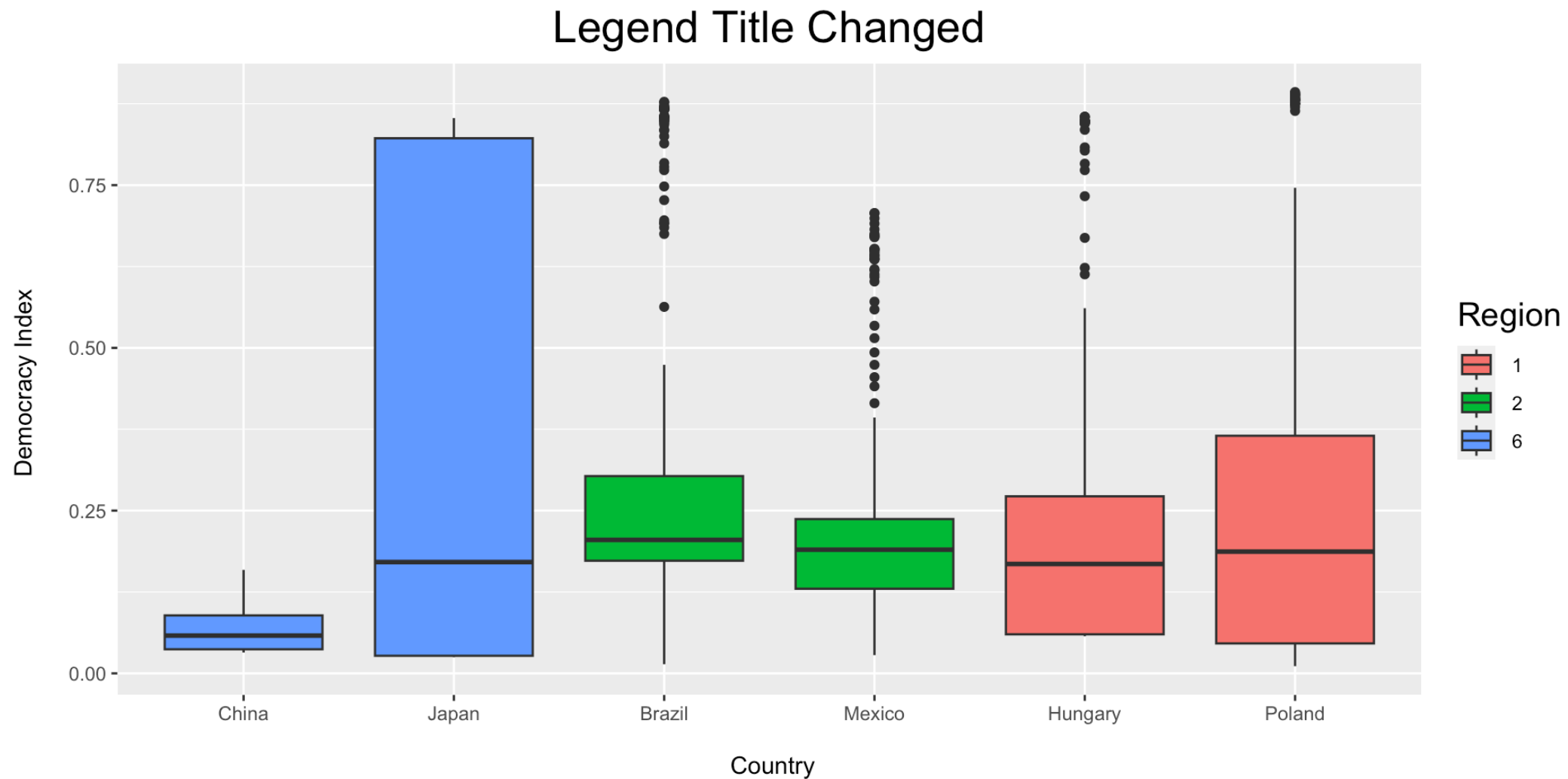
Change legend title

- Change legend title
→ `guides(fill = guide_legend(title = ))`
- Resize legend title
→ `theme(legend.title = element_text(size = ))`

```
1 vdem1 <- vdem %>% filter(country_name %in% c("Mexico", "Brazil", "Japan",
2                                           "China", "Poland", "Hungary"))
3 vdem1 %>%
4   ggplot(aes(x = country_name, y = v2x_polyarchy, fill = factor(e_regionpol_6C))) +
5   geom_boxplot() +
6   labs(title = "Legend Title Changed",
7        x = "\n Country", y = "Democracy Index \n") +
8   theme(plot.title = element_text(hjust = 0.5, size = 20),
9         legend.title = element_text(size = 15)) +
10  guides(fill = guide_legend(title = "Region"))
```

3. Legends

Change legend title



3. Legends

Hide legend

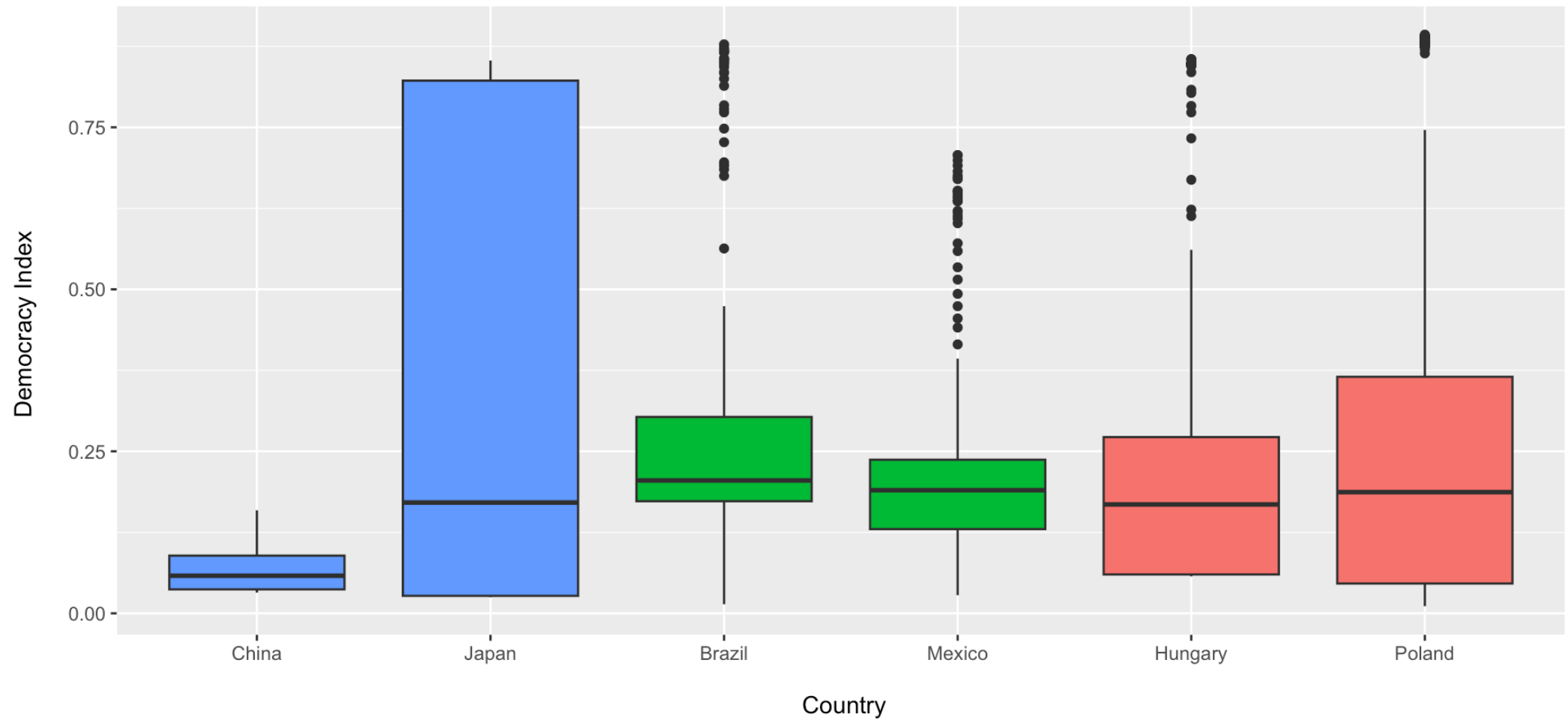
- You can decide turn off legends by using the code `show.legend = F`

```
1 vdem1 %>%
2   ggplot(aes(x = country_name, y = v2x_polyarchy,
3     fill = factor(e_regionpol_6C))) +
4   geom_boxplot(show.legend = F) +
5   labs(title = "Legend Title Changed",
6     x = "\n Country", y = "Democracy Index \n") +
7   theme(plot.title = element_text(hjust = 0.5, size = 20),
8     legend.title = element_text(size = 15))
```

3. Legends

Hide legend

Legend Turned Off



3. Legends

Change legend position

```
theme(legend.position = "bottom" / "left" / "top")
```

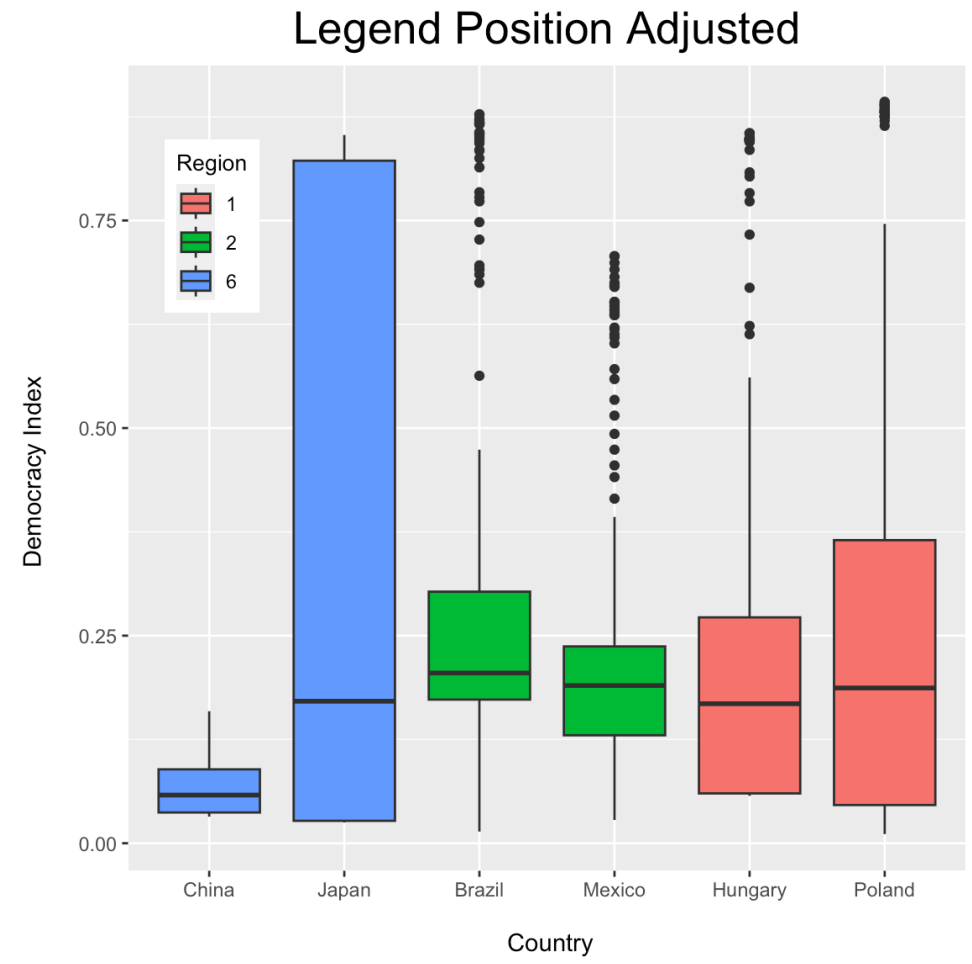
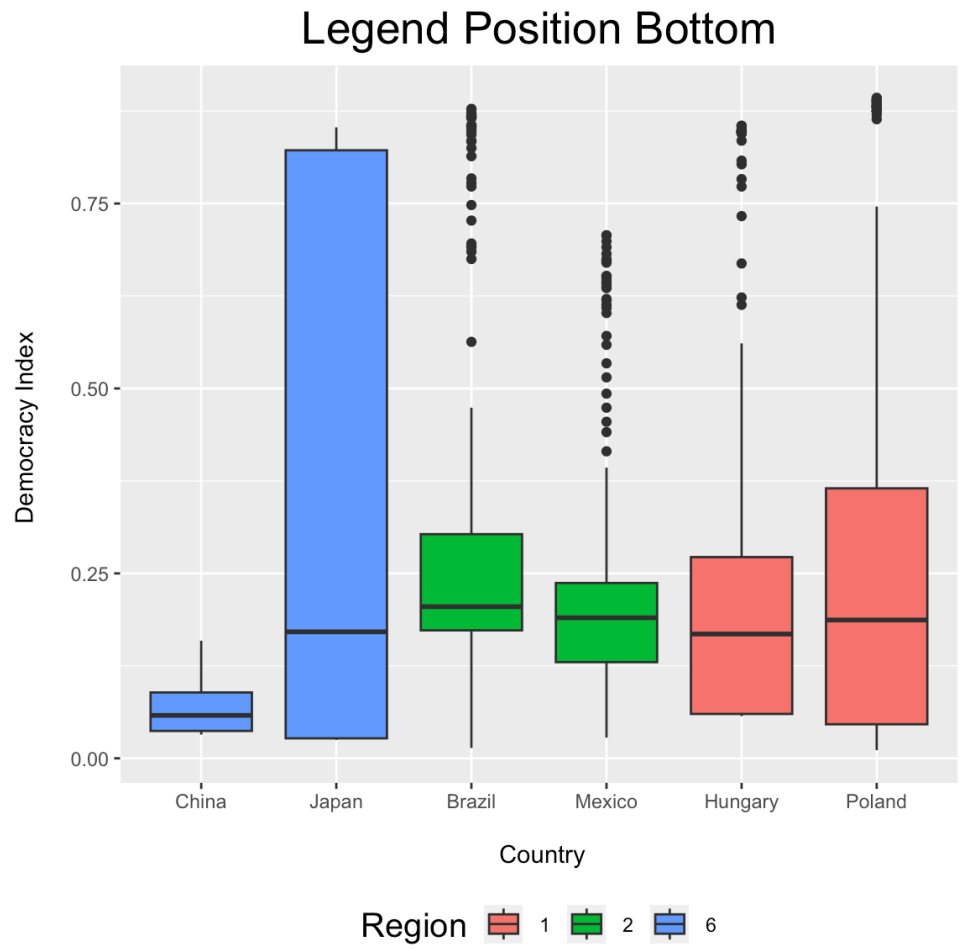
```
1 vdem1 %>%
2   ggplot(aes(x = country_name, y = v2x_polyarchy, fill = factor(e_regionpol_6C))) +
3   geom_boxplot() +
4   labs(title = "Legend Position Bottom", x = "\n Country", y = "Democracy Index \n") +
5   theme(plot.title = element_text(hjust = 0.5, size = 20),
6         legend.title = element_text(size = 15),
7         legend.position = "bottom") +
8   guides(fill = guide_legend(title = "Region"))
```

```
theme(legend.position = c(x-coordinate, y-coordinate))
```

```
1   theme(plot.title = element_text(hjust = 0.5, size = 20),
2         legend.title = element_text(size = 10),
3         legend.position = c(0.1, 0.8))
```

3. Legends

Change legend position



Colors and Themes

4. Colors and Themes

Basic Themes

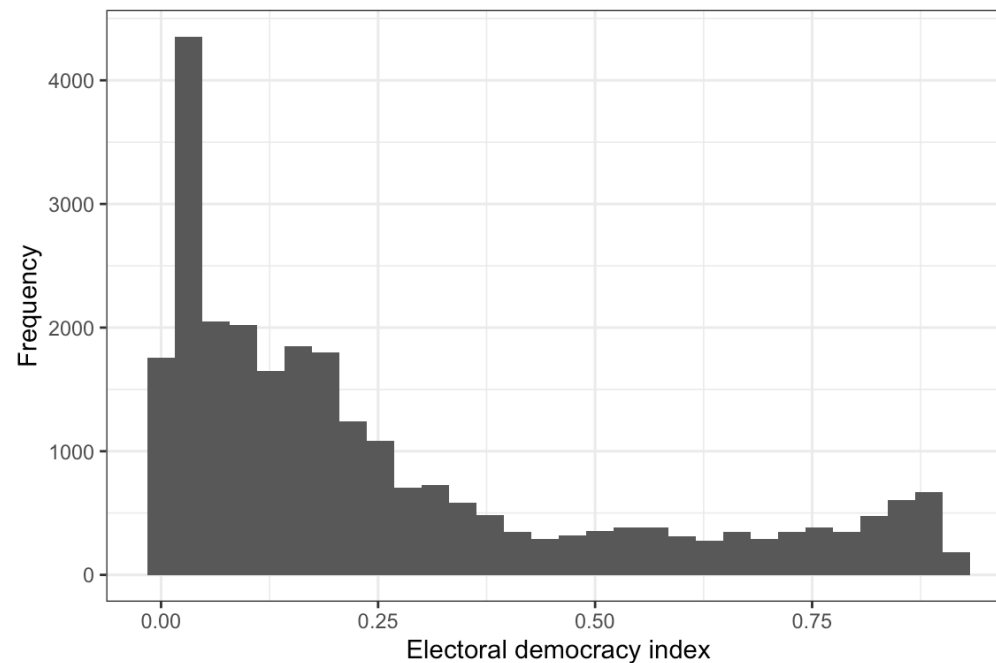
- The ggplot2 package comes with eight different themes.
- `theme_grey()`, `theme_bw()`, `theme_linedraw()`,
`theme_light()`, `theme_dark()`, `theme_minimal()`,
`theme_classic()`, `theme_void()`

4. Colors and Themes

Basic Themes

- For white background and black gridlines: `theme_bw()`

```
1 vdem %>%  
2   ggplot(aes(x = v2x_polyarchy)) +  
3   geom_histogram() +  
4   labs(x = "Electoral democracy index", y = "Frequency") +  
5   theme_bw()
```



4. Colors and Themes

fill and color

- `fill = "colorname/colorcode"` and `color = "colorname/colorcode"` customize the colors of your graph
- `fill` changes the color inside the shape, while `color` modifies its border color.

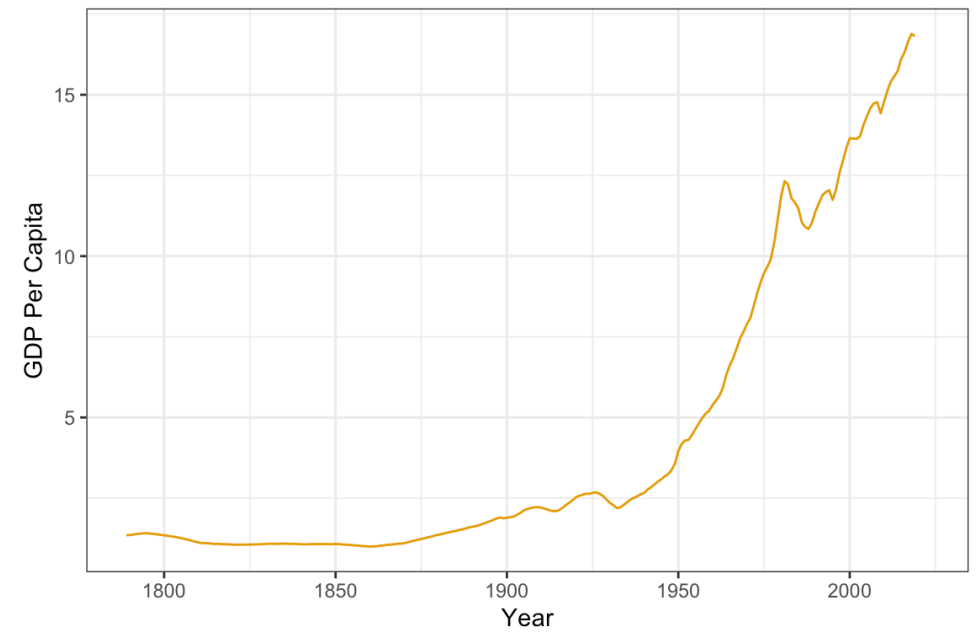
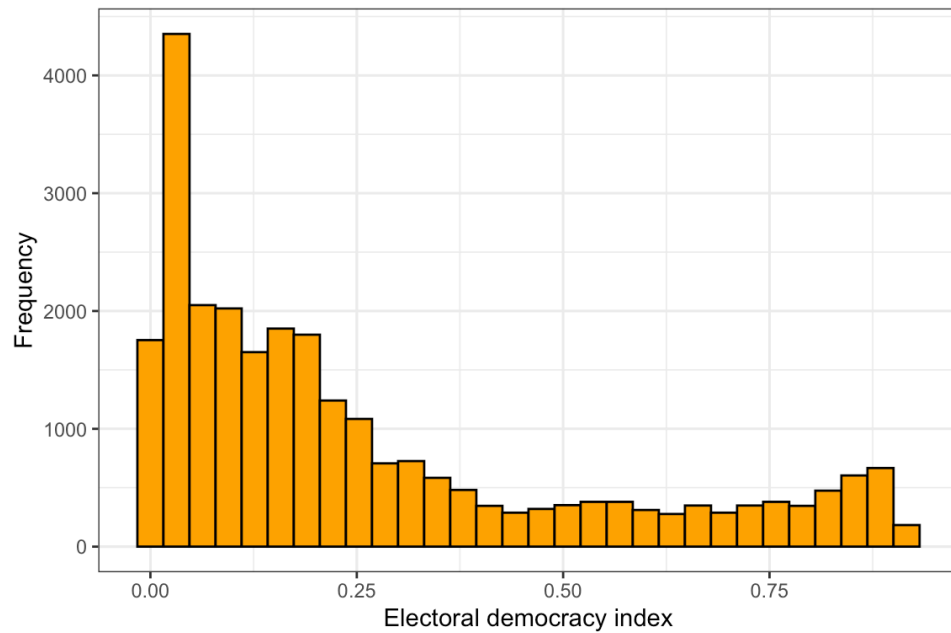
```
1 vdem %>%  
2   ggplot(aes(x = v2x_polyarchy)) +  
3   geom_histogram(fill = "orange", color = "black") +  
4   labs(x = "Electoral democracy index", y = "Frequency") +  
5   theme_bw
```

```
1 vdem %>% filter(country_name == "Mexico") %>%  
2   ggplot(aes(x = year, y = e_gdppc)) +  
3   geom_line(color = "#E69F00") +  
4   labs(x = "Year", y = "GDP Per Capita") +  
5   theme_bw
```

4. Colors and Themes

fill and color

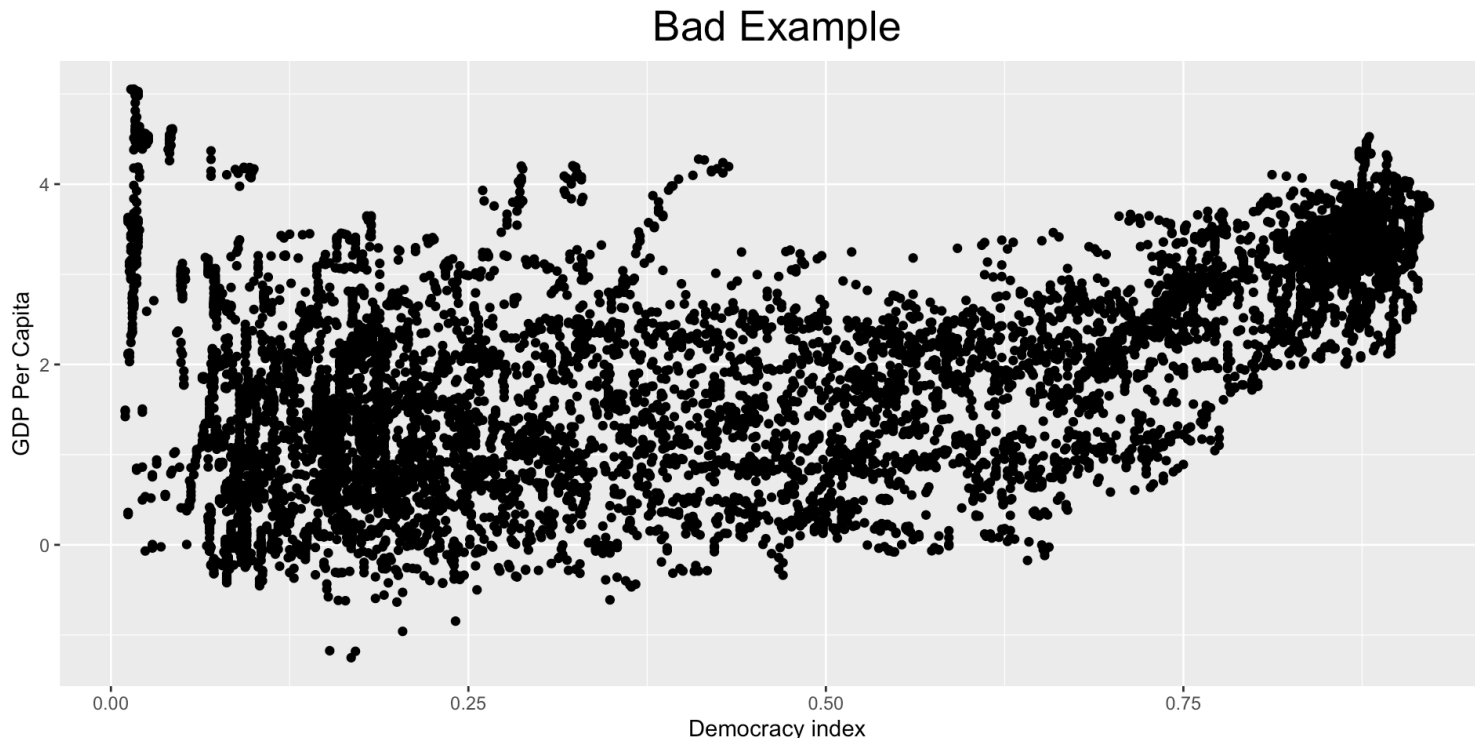
- **fill** changes the color inside the shape, while **color** modifies its border color.
- Histogram
- line graph



4. Colors and Themes

alpha

- **Alpha** refers to the opacity of a geom
- Range from 0 to 1, with lower values corresponding to more transparent colors
- Important for managing overplotting and improving visibility

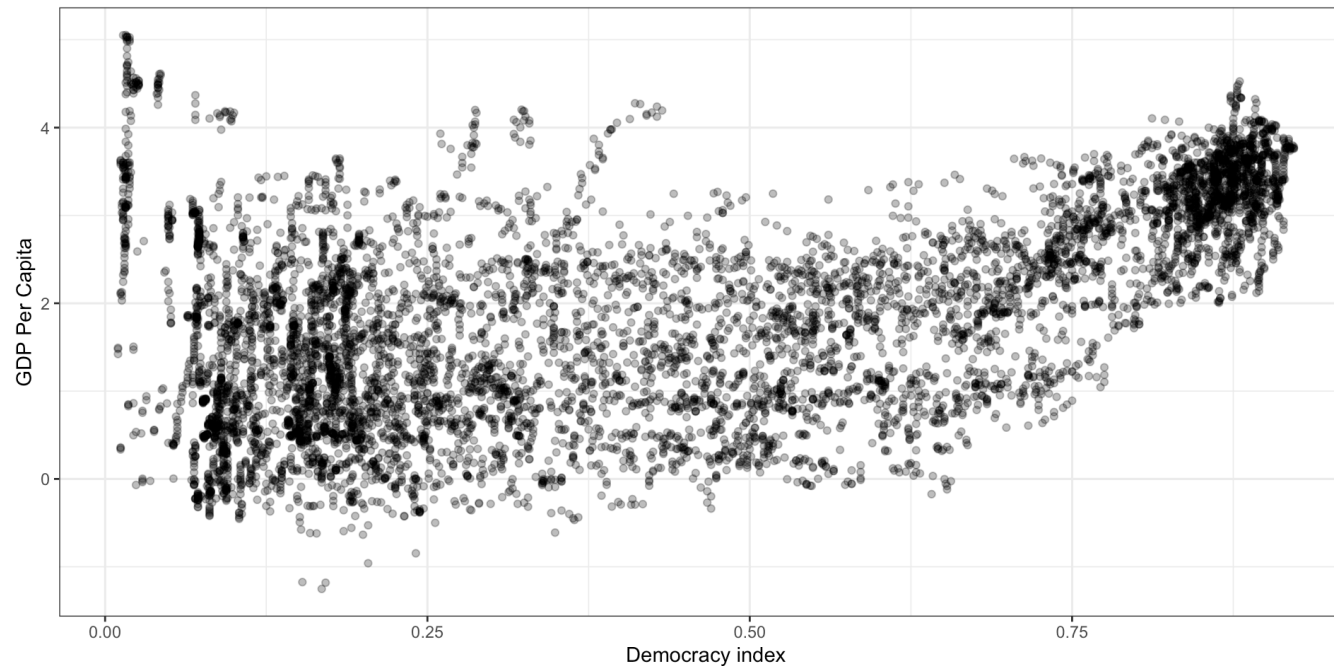


4. Colors and Themes

alpha

- $\alpha = 0$ to 1

```
1 vdem %>% filter(!is.na(v2x_polyarchy) & !is.na(e_gdppc) & year >= 1970) %>%  
2   ggplot(aes(x = v2x_polyarchy, y = log(e_gdppc))) +  
3   geom_point(alpha = 0.3) +  
4   labs(x = "Democracy index", y = "GDP Per Capita") +  
5   theme_bw()
```



4. Colors and Themes

alpha

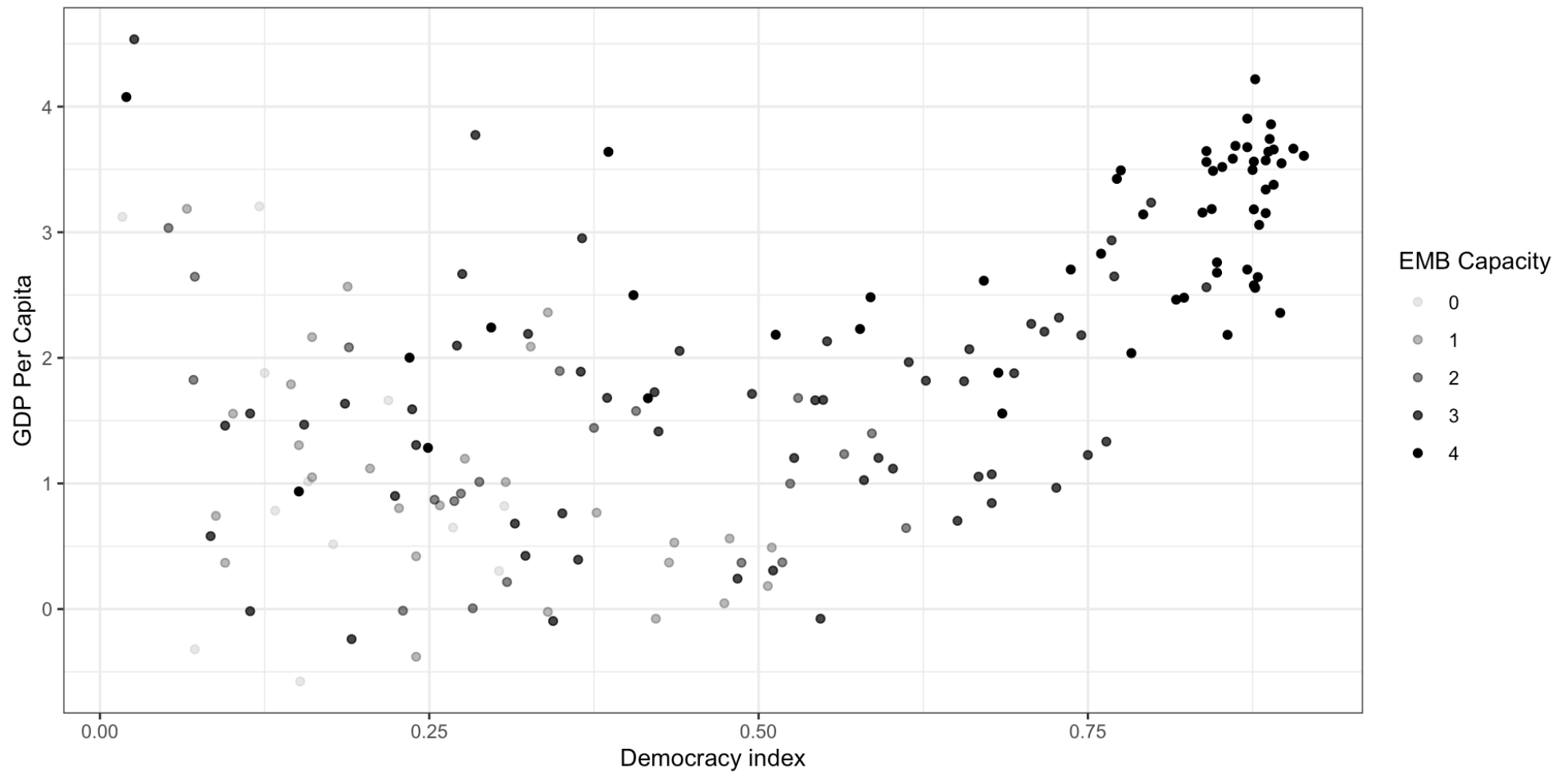
- You can use alpha to down-weight less important observations
- `aes(alpha = variable name)`
- Changing the title: `scale_alpha("your title")`

```
1 vdem %>% filter(!is.na(v2x_polyarchy) & !is.na(e_gdppc) & year == 2000) %>%  
2   ggplot(aes(x = v2x_polyarchy, y = log(e_gdppc))) +  
3   geom_point(aes(alpha = v2elembcap_ord)) +  
4   scale_alpha("EMB Capacity") +  
5   labs(x = "Democracy index", y = "GDP Per Capita") +  
6   theme_bw
```

4. Colors and Themes

alpha

Alpha adjusted by a 3rd variable



4. Colors and Themes

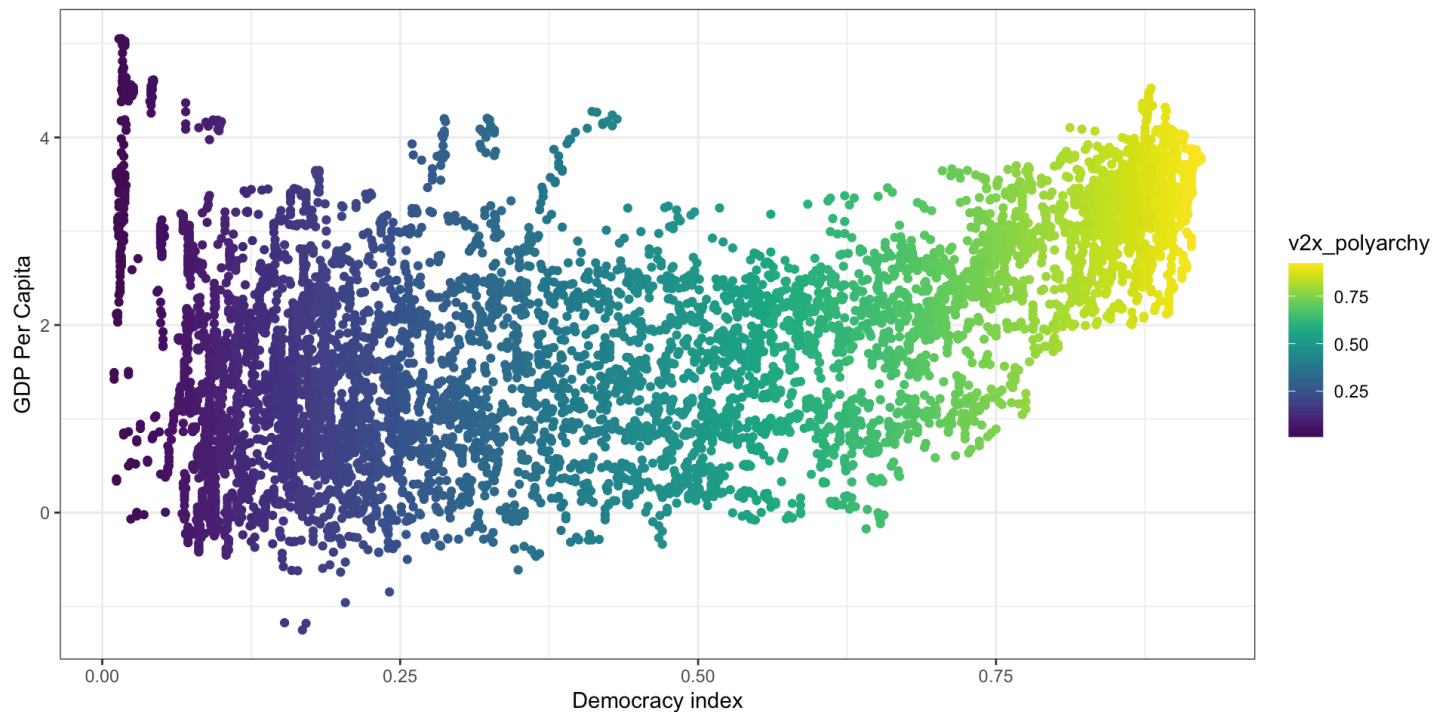
Color Palettes

- You can download color palettes in R
- R color palettes to know for data visualization: **viridis**, **RColorBrewer**, **ggsci**

4. Colors and Themes

Color Scales - *viridis*

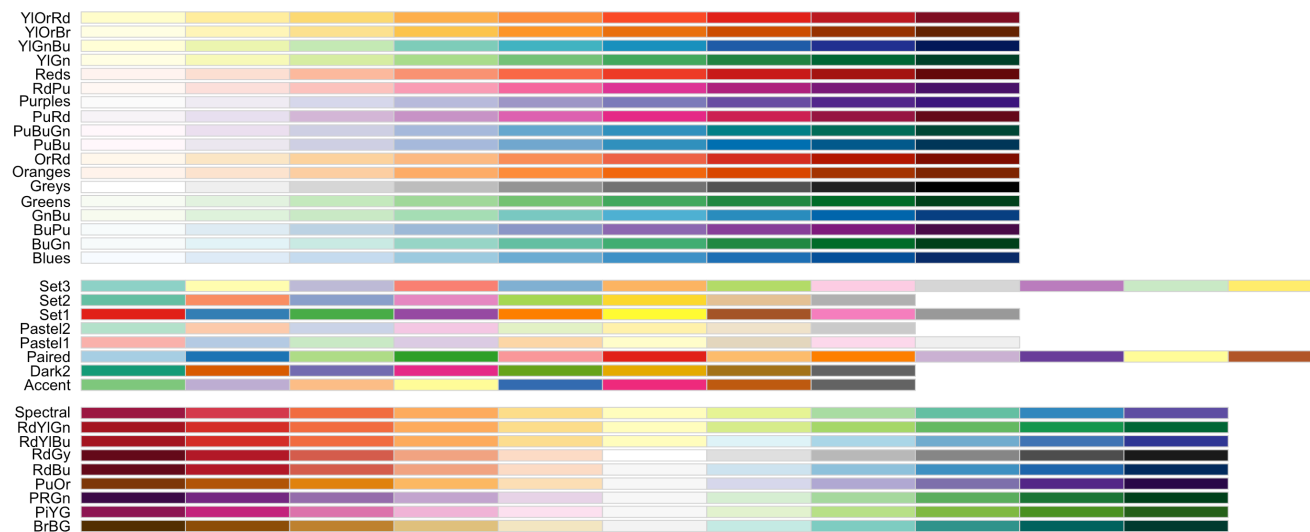
```
1 install.packages('viridis')
2 library(viridis)
3 vdem %>% filter(!is.na(v2x_polyarchy) & !is.na(e_gdppc) & year >= 1970) %>%
4   ggplot(aes(x = v2x_polyarchy, y = log(e_gdppc), color = v2x_polyarchy)) +
5   geom_point(aes()) +
6   labs(x = "Democracy index", y = "GDP Per Capita") + theme_bw() +
7   scale_color_viridis()
```



4. Colors and Themes

Color Palettes - RColorBrewer

- Brewer palettes:

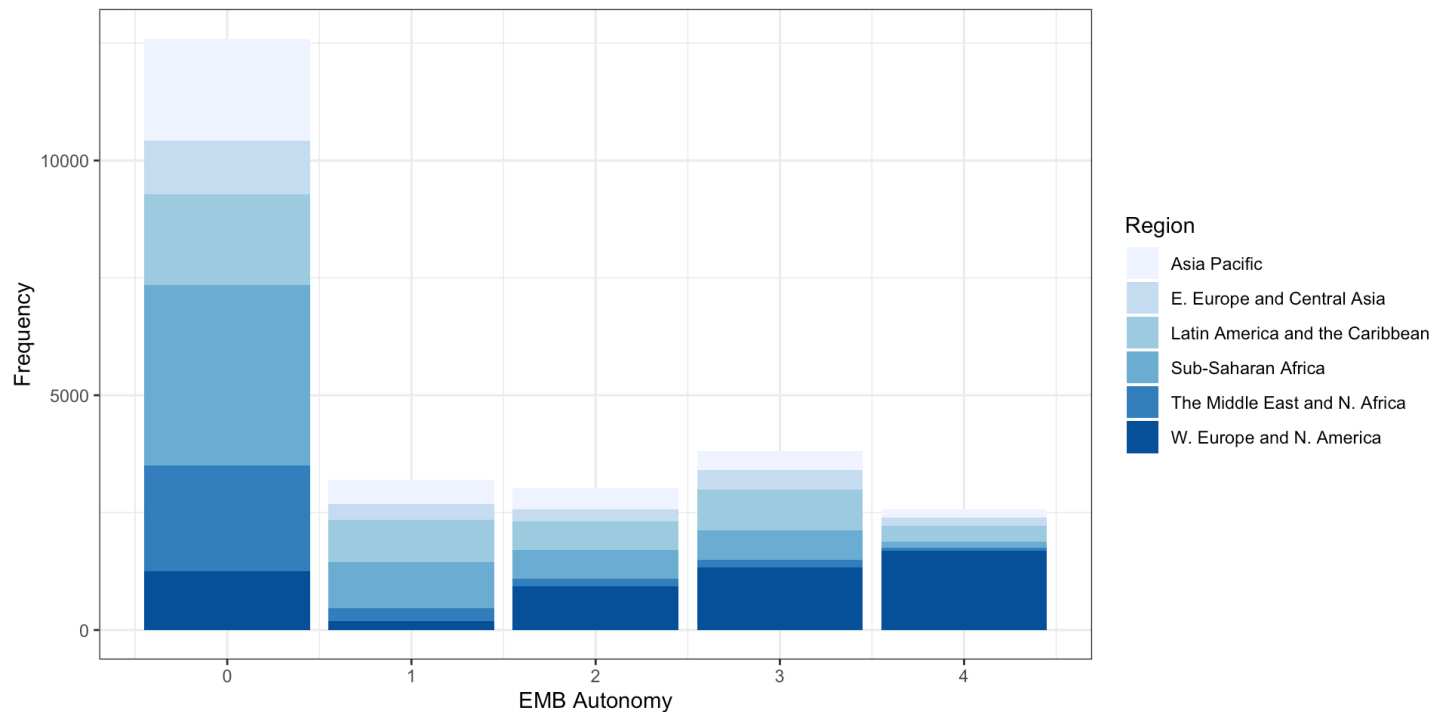


- Adopting color palettes: `scale_fill_brewer(palette = "palette name")`

4. Colors and Themes

Color Palettes - *RColorBrewer*

```
1 install.packages('RColorBrewer')
2 library(RColorBrewer)
3 vdem %>%
4   ggplot(aes(x = v2elembaut_ord, fill = region)) +
5   geom_bar() +
6   labs(x = "EMB Autonomy", y = "Frequency", fill = "Region") + theme_bw() +
7   scale_fill_brewer(palette = "Blues")
```



4. Colors and Themes

Color Palettes - ggsci

- ggsci: scientific journal color palettes
- For palette names, refer to <https://cran.r-project.org/web/packages/ggsci/vignettes/ggsci.html>
- Adopting color palettes:

`scale_color_palettename()`

`scale_fill_palettename()`

4. Colors and Themes

Color Palettes - *ggsci*

```
1 install.packages('ggsci')
2 library(ggsci)
3 vdem1 %>%
4   ggplot(aes(x = country_name, y = v2x_polyarchy, fill = factor(e_regionpol_6C))) +
5   geom_boxplot() + labs(x = "\n Country", y = "Democracy Index \n") +
6   theme(plot.title = element_text(hjust = 0.5, size = 20)) +
7   guides(fill = guide_legend(title = "Region")) + theme_bw() +
8   ggsci::scale_fill_lancet()
```

