# Data Visualization

Okan Sarioglu Leon Siefken

2023-08-21

## Contents

**Chapter 4: Data Visualization with GGPlot** **1**

1. GGPlot2: Simpliefied Plotting in R . . . . . 1
2. Varieties of Democracy Dataset . . . . . 2
3. Making a Plot . . . . . 2
4. Other Cosmetics . . . . . 12
5. Concluding Remarks . . . . . 16

Exercises: Make your own Plot . . . . . 16

## Chapter 4: Data Visualization with GGPlot

### 1. GGPlot2: Simpliefied Plotting in R

The package `GGPlot2` is part of the Tidyverse and aims to simplify data visualization by utilizing the "Grammar of Graphics" defined by Leland Wilkinson. While it may appear complicated at first, it is as simple as greating a frame and then adding the elements you want to it! Let us start with loading the package.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

## 2. Varieties of Democracy Dataset

To make the introduction of GGPlot2 more intersting, we will also introduce another essential dataset in political science. The "Varieties of Democarcy" dataset (V-Dem) is an expansive multidimensional dataset, attempting to measure the state of democracy in a large variety of countries and over a long time-span. In addition to democracy, the V-Dem data also incorporates numerous other indices and measures, provided by scientists around the world.

Our goal with this data today, is to visualize the development of democracy over the years. Can we see the "Waves of Democratization" as theorized by Huntington? And if so, what kind of wave are we in right now, increasing or decreasing democratization?

```
setwd("C:/Users/leons/Desktop/R Intro/Data")
vdem <- read.csv("vdem_subset.csv")
```

## 3. Making a Plot

As said earlier, at the basis of every visualization lays it's frame. We can create this frame by simply using the following command and inputting our data.

```
ggplot(vdem)
```

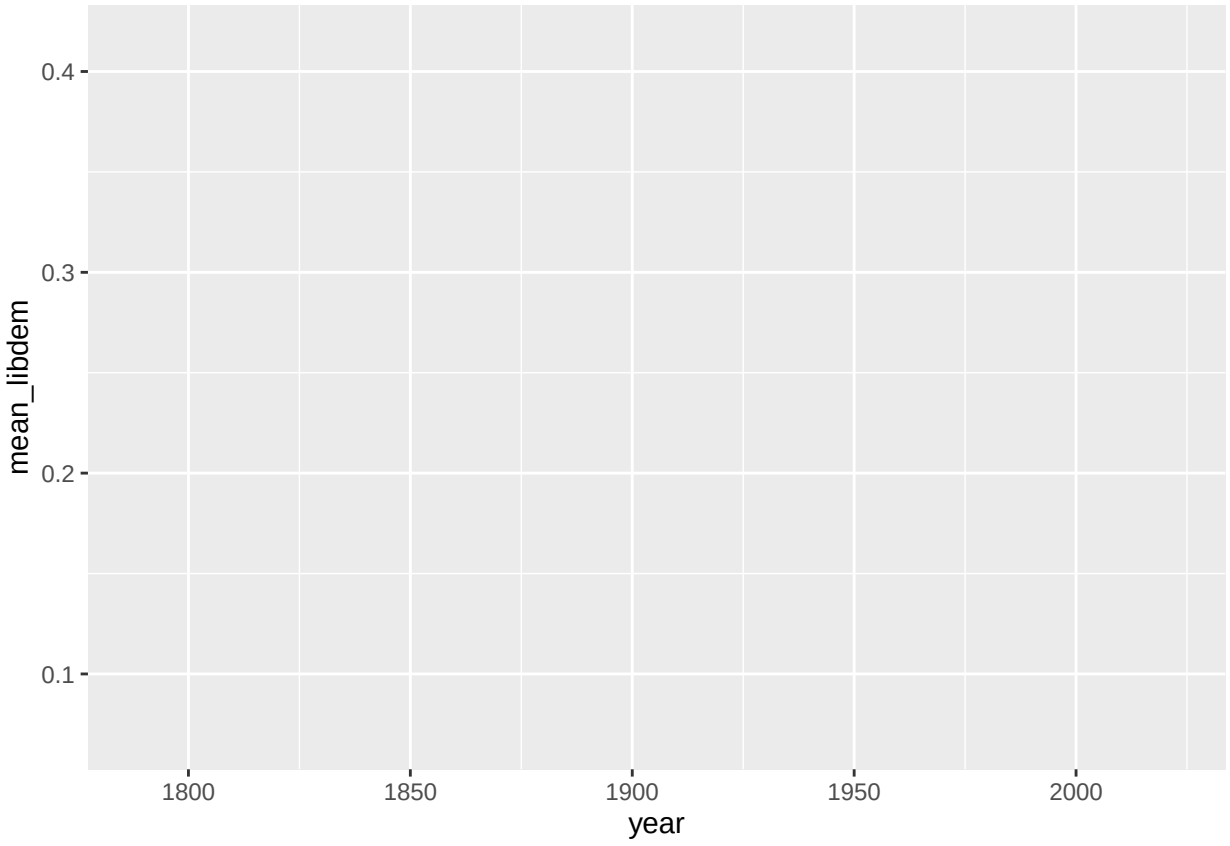
### 3.1 Adding our Plots

A bit empty, no? Let us add our variables to it and make a nice graphic step by step. Let us start by adding our variables. The variable we will be using for this is a mean of V-Dem's main index, the liberal democracy index, around the world, per year. Let us create that variable first.

```
df_pol <- vdem %>% select(year,v2x_libdem)%>%
  group_by(year)%>%
  summarize(mean_libdem = mean(v2x_libdem,na.rm=TRUE))
```

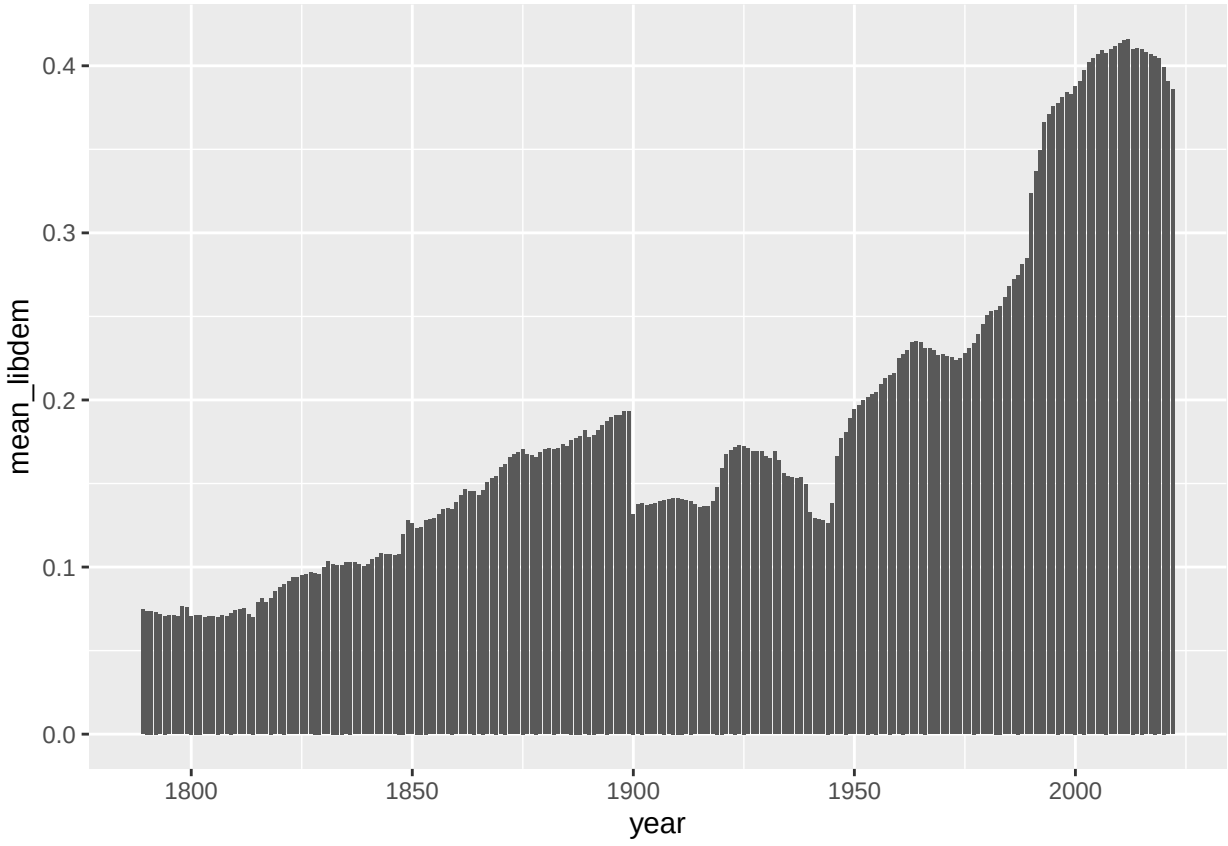
No we add our variables to the previous ggplot code line. For this, next to our data, we to specify our axis by adding the `aes` command in which we can specify the variables on the x- and y-axis.

```
ggplot(data=df_pol, aes(x = year, y = mean_libdem))
```



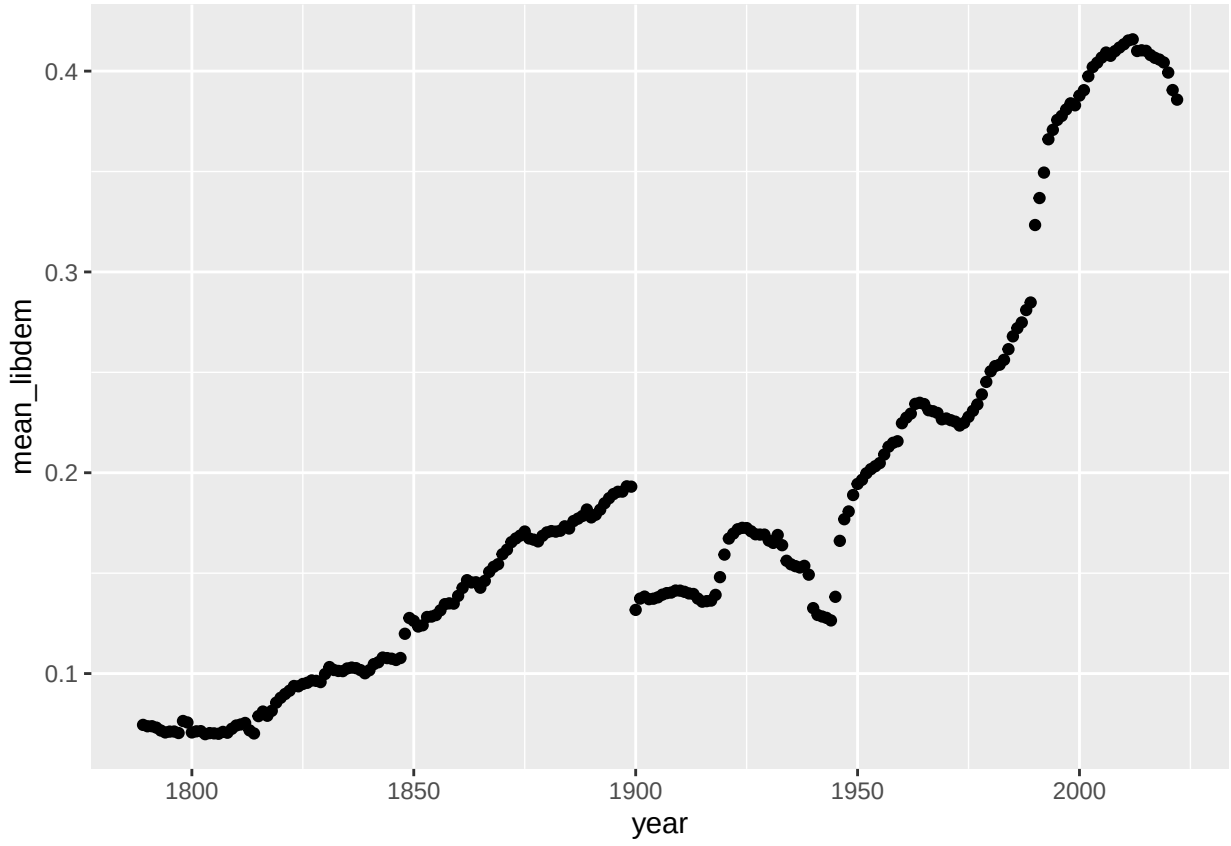

Great! Now we have the year on the x-axis and the mean polyarchy value on the y-axis. But where are the observations? Well, as said before, you have to add everything step by step. You have to tell ggplot which kind of visualization you want for your data. To add this informational lines, you usually use a simple `+`. If you want a bar-plot, you simply add `+geom_bar()` and if you want to add a scatter-plot you would add `+geom_point()` and so on. Let us look at some examples.

```
ggplot(data=df_pol, aes(x = year, y = mean_libdem)) + geom_bar(stat = "identity") #bar plot requires sp
```



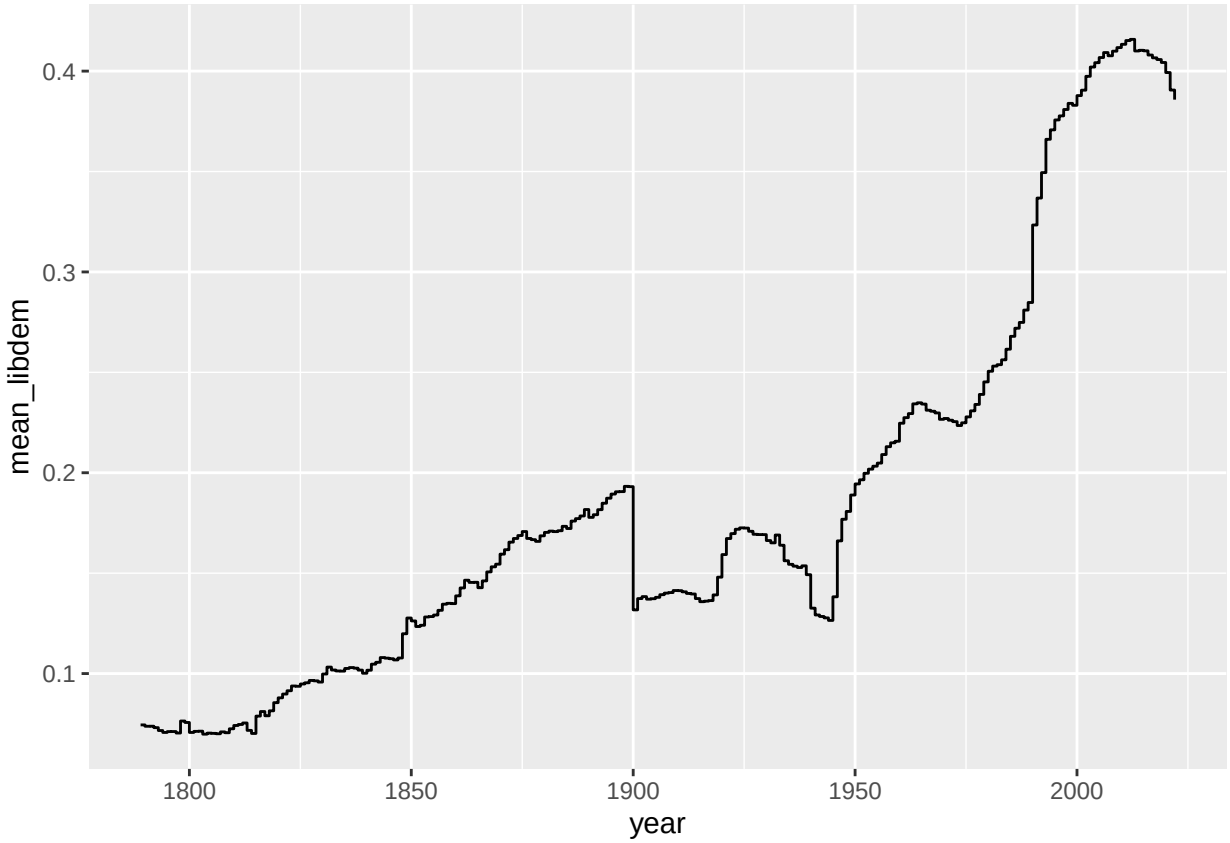

**Bar-Plots:**

```
ggplot(data=df_pol, aes(x = year, y = mean_libdem)) + geom_point()
```




**Scatterplot:**

```
ggplot(data=df_pol, aes(x = year, y = mean_libdem)) + geom_step()
```

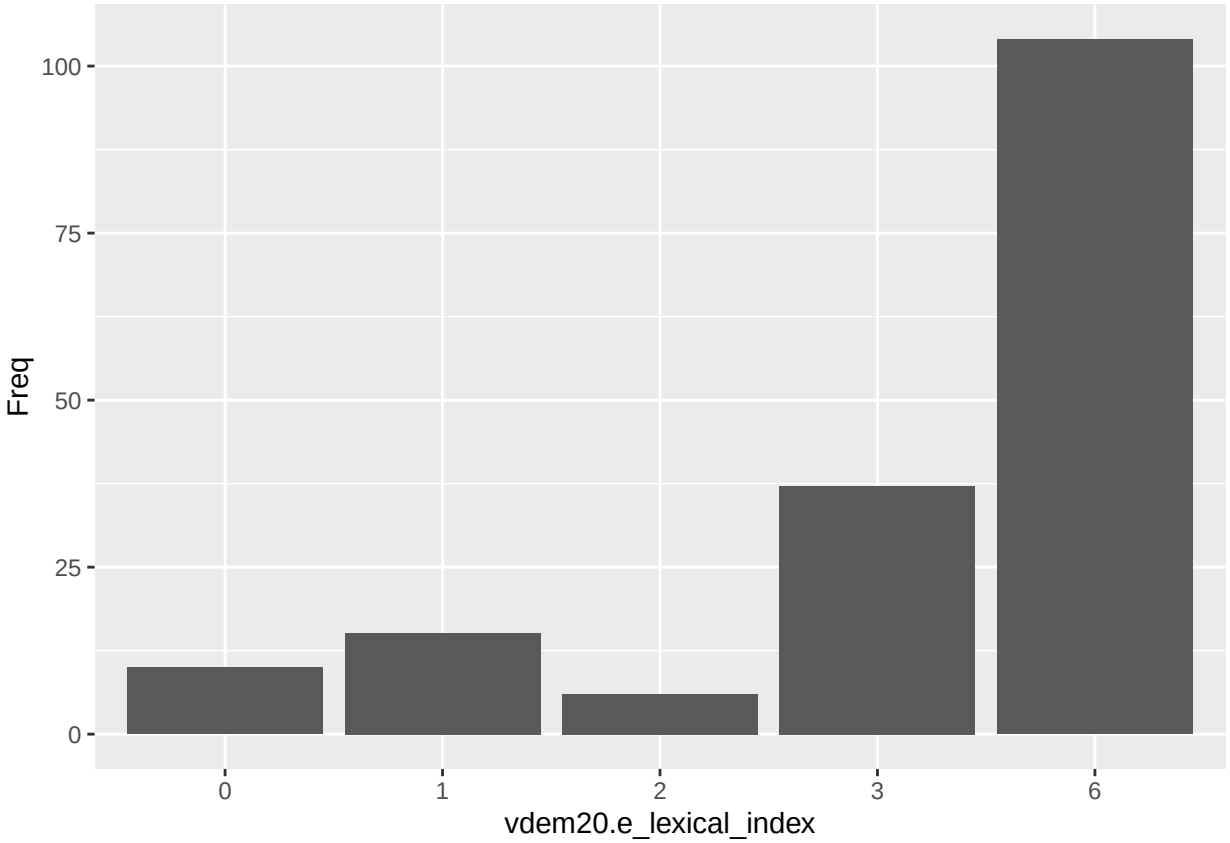



**Steps:**

These are some examples for simple plots using your data. And these examples also show us, that we do actually seem to find Huntington's waves of democratization in the data! Sadly, they also show us that we appear to be in a fourth wave of autocratization which, however, is line with the theory.

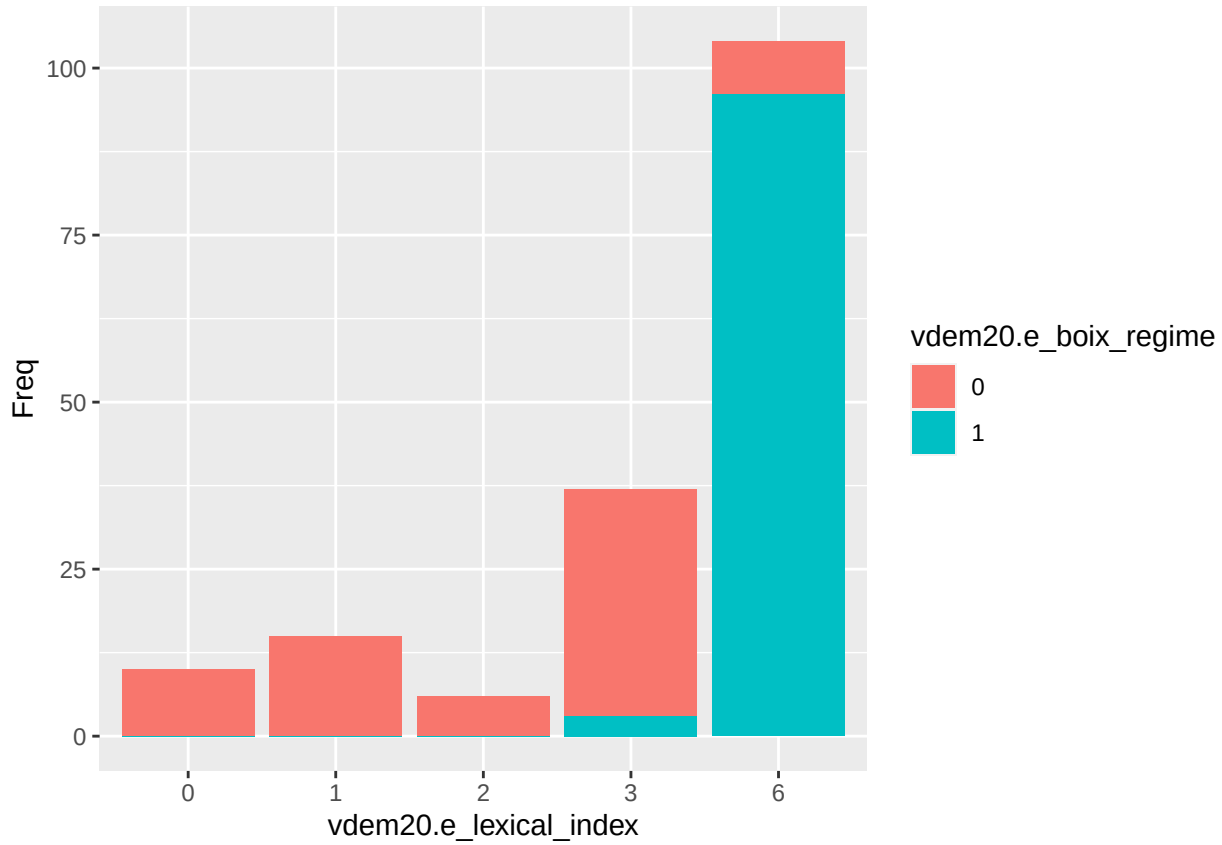
Let us take a look at how countries were rated on V-Dem's liberal democracy index in the most current year of measurement. For this, we first create a table with the necessary information we want to visualize and transform it into a dataset. And just for fun, let us compare the Lexical Index of Democracy with the binary democracy measure by Boix et al.

```
vdem20 <- vdem %>% filter(year == 2020)

tab <- xtabs(~vdem20$e_lexical_index + vdem20$e_boix_regime)

t_df  <- data.frame(tab)
View(t_df)

ggplot(data = t_df, aes(x = vdem20.e_lexical_index, y = Freq)) + geom_col()
```




With a value of 6 standing for universal suffrage and competitive elections, this graph shows that democracy appears to be the most common form of government in the data, at least for the year 2020. Over a 100 countries are seen as fully democratic, while less than a hundred are categorized as less than fully democratic. Does this categorization also match with how Boix et al defined a democracy? To do so, we can add a third variable into the graphic. There are multiple ways to do so. One is to differentiate by color. Let us try that.

```
ggplot(data = t_df, aes(x = vdem20.e_lexical_index, y = Freq, fill = vdem20.e_boix_regime)) + geom_col(
```




Interesting. These Indexes are very similar, however show some differences as well. With a democracy being coded as 1 in the index by Boix, not all countries coded as a perfect democracy by the Lexical Index were also coded as that by Boix. While on the other hand some variables coded as imperfect democracies by Lexical were coded as democratic by Boix. Thus, it is always very important to specifically define which definition of democracy, but generally any concept, your using in your papers, as results may vary based on that.

Alright, now let us get more... full scale! We now want to create a graphic that shows us how democracy has developed in different regions of the earth and which regime was the most prevalent in each year. To do so, we have to combine a few of the things we have learned so far. First, we have to create a variable that categorizes each country into a region.

```
vdem2 <- vdem%>%
  select(year,v2x_regime,country_name) %>%
  mutate(regions_earth = case_when(country_name %in%c("Algeria", "Bahrain", "Egypt","Iran", "Iraq", "Isr

                                   country_name %in%c("Austria","Belgium","France","Germany","Ireland","Luxe
                                   country_name
                                   %in%c("Afghanistan","Kazakhstan","Kyrgyzstan","Tajikistan","Turkmenistan
                                   country_name

%in%c("Barbados","Cuba","Dominican Republic","Guyana","Haiti","Jamaica","Suriname","Trinidad and Tobago
                                   country_name

%in%c("Armenia","Azerbaijan","Georgia","China","Hong Kong","Japan","Mongolia","North Korea","South Korea
                                   country_name
```

```
%in%c("Cameroon","Central African Republic","Chad","Democratic Republic of the Congo","Equatorial Guinea
)
```

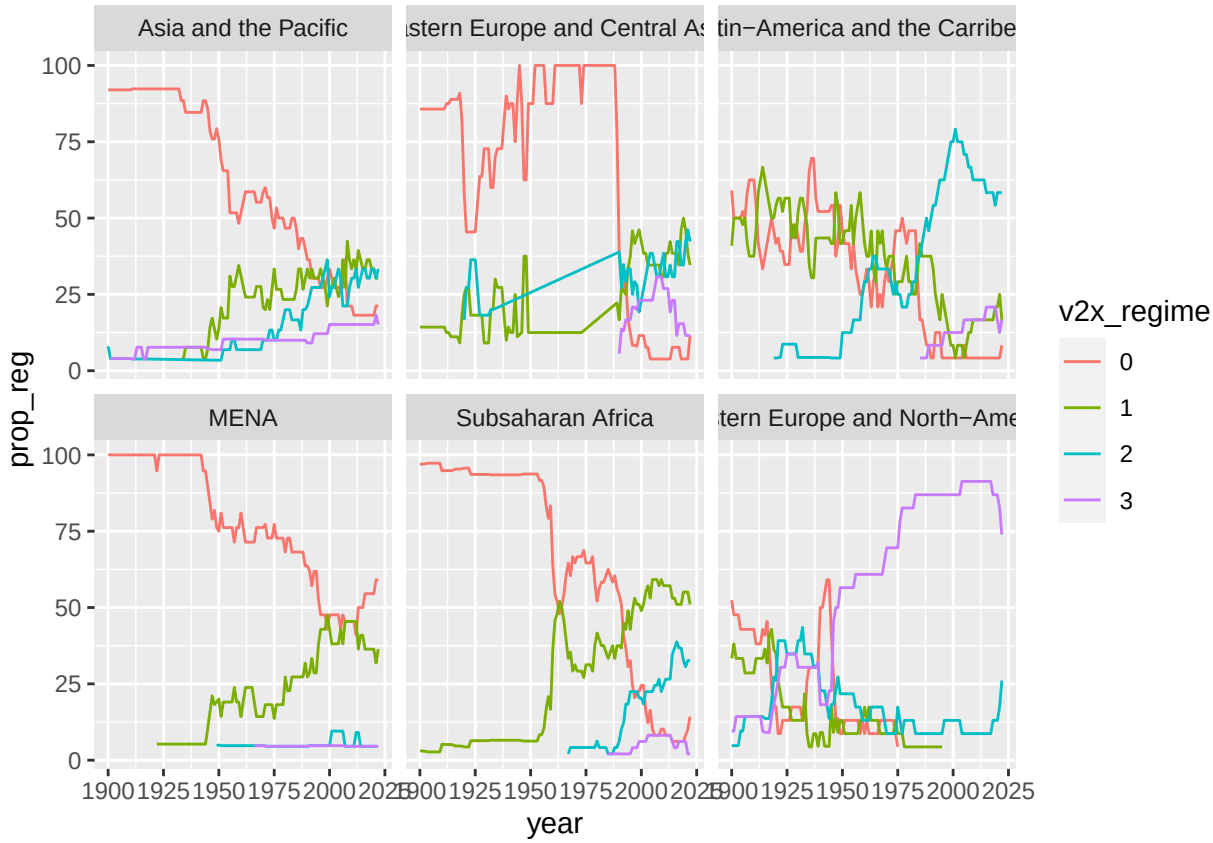
This code has used the mutate command to make a new variable that categorized every country into a region on earth. Now we need to turn the Democarcy index we are using here (Regimes of the World Index) into a factor variable to transform the individual number (1,2,3 or 4) given to countries into categories. Then we omit the missing values. And lastly before we can plot, we create a variable that contains the frequency of every regime withing each region per year. So we need to group the data and count the observations. Then we group again, but only by year and region, and use the count to calculate proportions of each regime. Now we know which percentage of countries had which regime per year and per region. Sound complicated, but go through the code a few times and you will understand.

Lastly, we can plot. I chose a line graph, but you can use any of the previous graphs like `geom_col()` as long as they make sense in this scenario. Then, `facet_wrap()` was added. This neat little command allows us to add a fourth variable to differntiate by! so while x is the year, y is the frequency of regimes, and the color is the regime itself, we split the graphs into one graph for each region using `facet_wrap()`.

```
vdem2$v2x_regime <- as.factor(vdem2$v2x_regime)

vdem2 <- na.omit(vdem2)

vdem3 <- vdem2 %>%
  group_by(year, regions_earth, v2x_regime) %>%
  summarise(count = n()) %>%
  group_by(year, regions_earth) %>%
  mutate(prop_reg = count / sum(count) * 100)
```

```
## `summarise()` has grouped output by 'year', 'regions_earth'. You can override
## using the `.groups` argument.
```

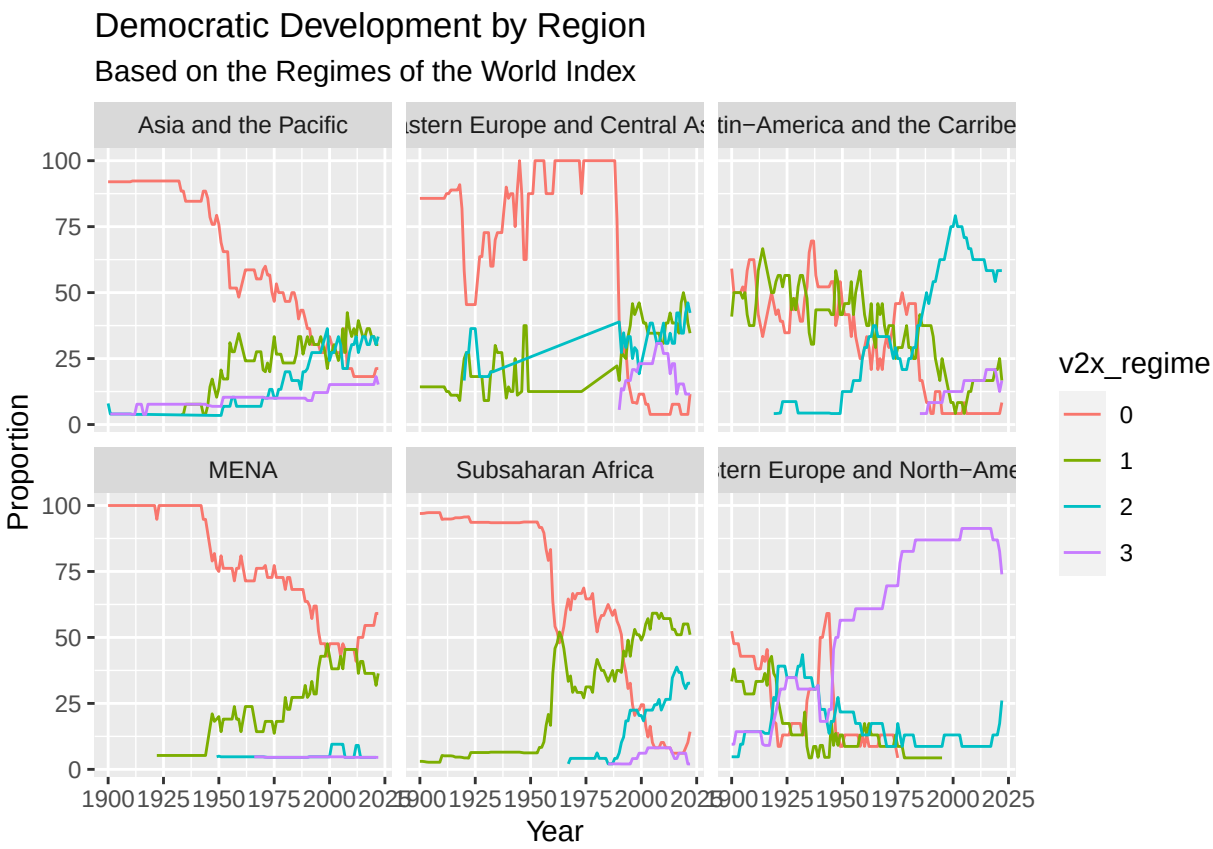
```
ggplot(vdem3, aes(x = year, y = prop_reg, col = v2x_regime)) + geom_line() + facet_wrap(~regions_earth)
```




And here we have it! This graph shows to what percentage which type of regime was present in each region. Now, there are some minor problems caused by incomplete data, for example in 1950 to 1975 in eastern Europe, this graph is easily understandable and transports a lot of information.... however, what do the regime numbers mean? Can you really publish a graphic that has the variable names on the axis? What about a title?

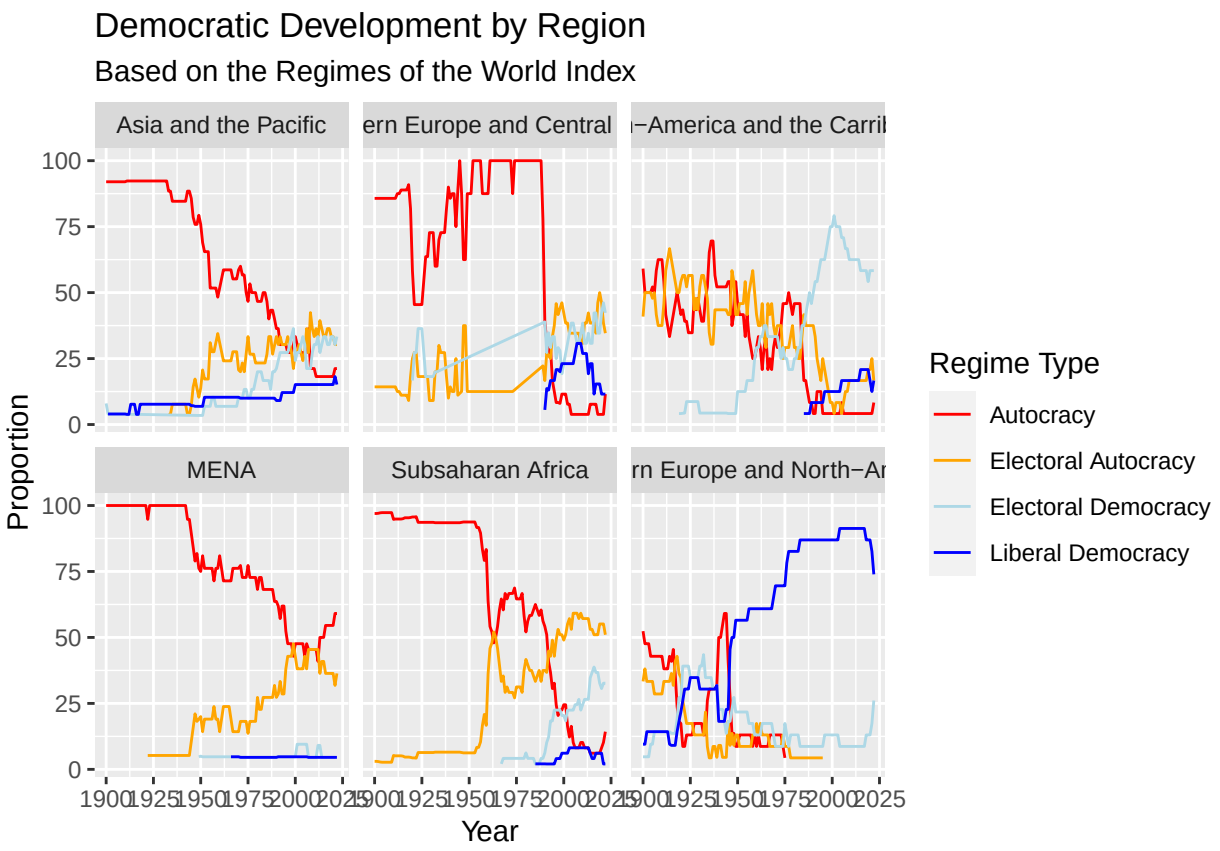
Yes, all these problems can be fixed of course, by introducing labels! Easiest way to understand them is to look at an example. Look at this version of the code.

```
ggplot(vdem3, aes(x = year, y = prop_reg, col = v2x_regime)) +
  geom_line() +
  labs(title = "Democratic Development by Region",
       subtitle = "Based on the Regimes of the World Index",
       x="Year",
       y="Proportion",
       fill = "Regime Type") +
  facet_wrap(~regions_earth)
```




Now we have some nice labels in our graph. But how about that legend?

```
ggplot(vdem3, aes(x = year, y = prop_reg, col = v2x_regime)) +
  geom_line() +
  labs(title = "Democratic Development by Region",
       subtitle = "Based on the Regimes of the World Index",
       x="Year",
       y="Proportion",
       color = "Regime Type") +
  facet_wrap(~regions_earth) +
  scale_color_manual(values = c("red", "orange", "lightblue", "blue"), breaks = c(0,1,2,3), labels =
                       c("Autocracy", "Electoral Autocracy", "Electoral Democracy", "Liberal Democracy"))
```




See what happened? With `scale_color_manual` we can manually change both the color and the labels of the graph and the legend. Color codes are a rabbit hole in ggplot. For some color variety, I recommend the following cheat sheet: Cheat Sheet.

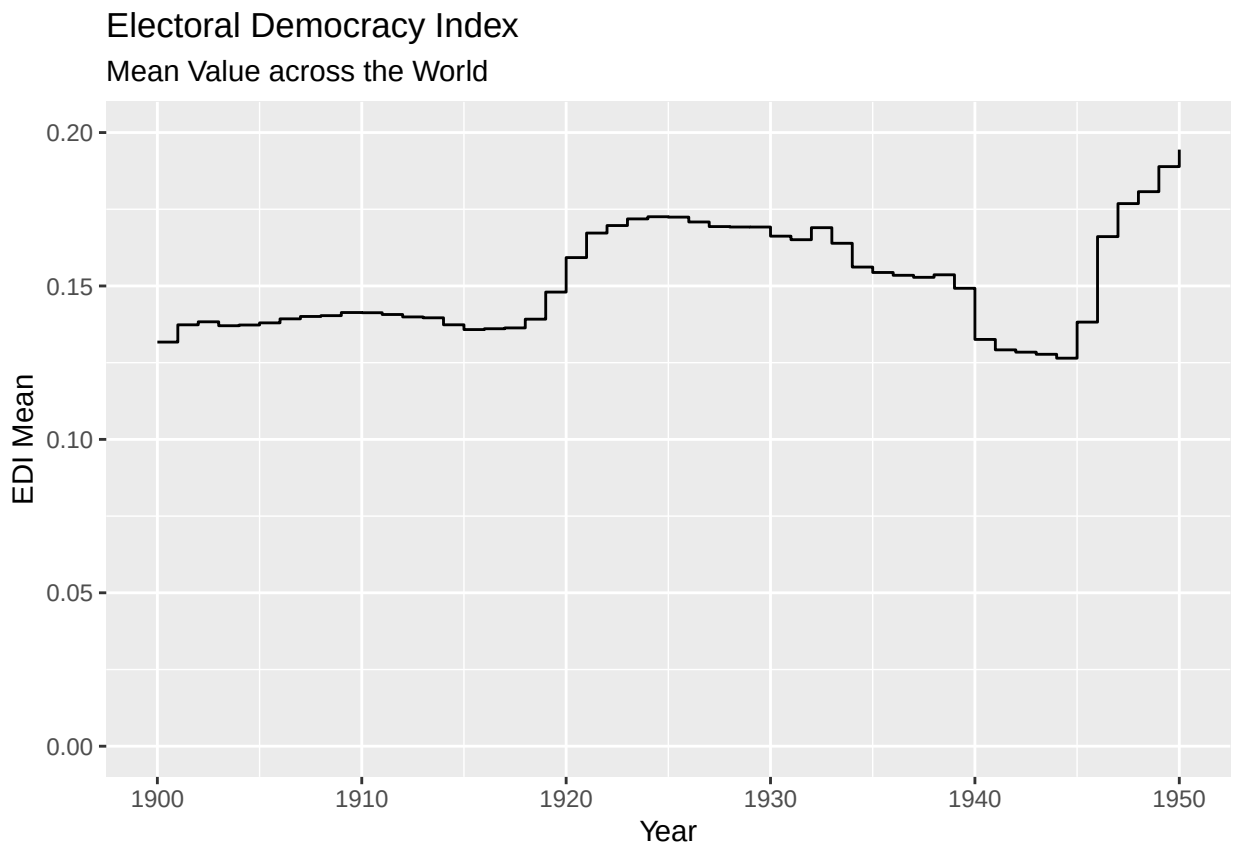
## 4. Other Cosmetics

However, there are lot more things you can change about your graph. A few different things will be demonstrated here:

Changing the axis size

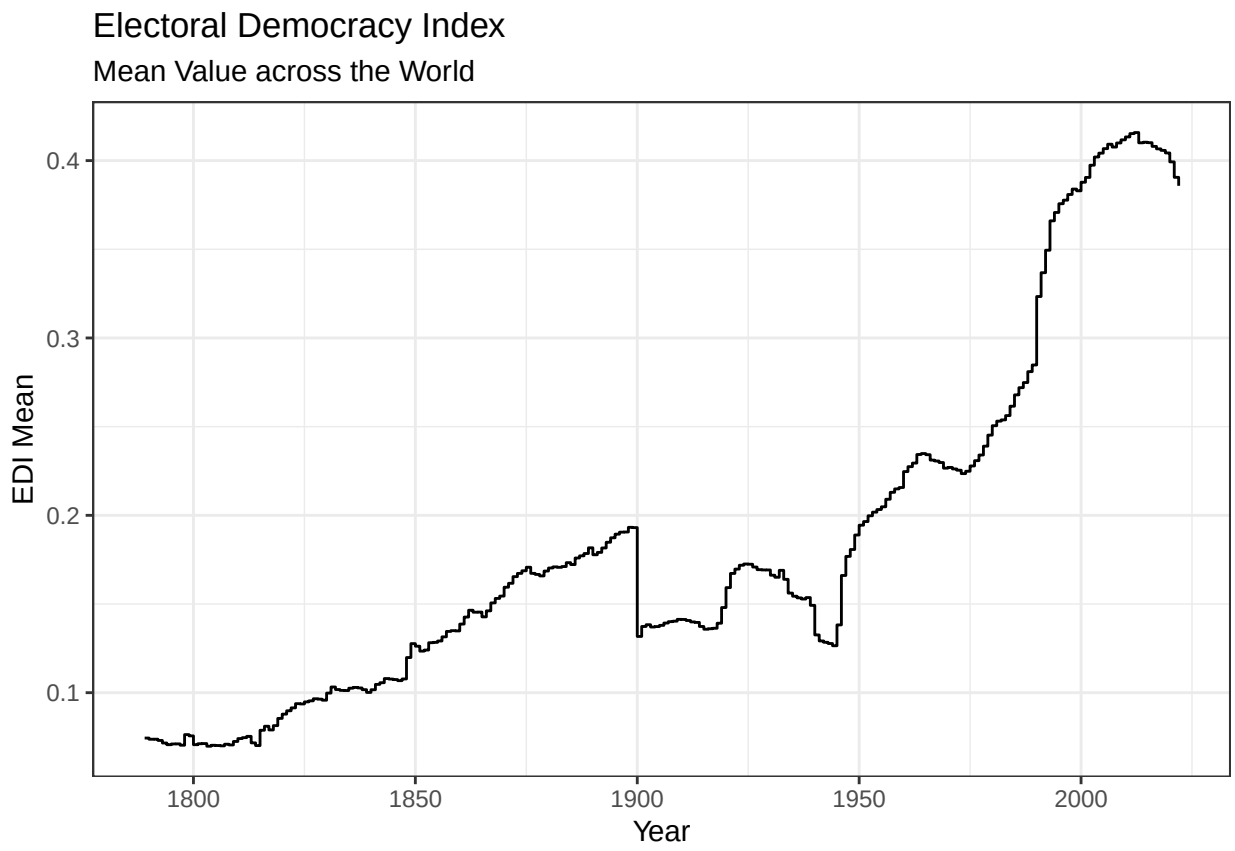
```
ggplot(data=df_pol, aes(x = year, y = mean_libdem)) +
  geom_step() +
  labs(title = "Electoral Democracy Index",
       subtitle = "Mean Value across the World",
       y = "EDI Mean",
       x = "Year")+
  ylim(0,0.2) +    #If we want ggplot to limit the axis to values between 0 and 0.2
  xlim(1900,1950) #And only the years 1900 to 1950
```

```
## Warning: Removed 183 rows containing missing values (‘geom_step()‘).
```

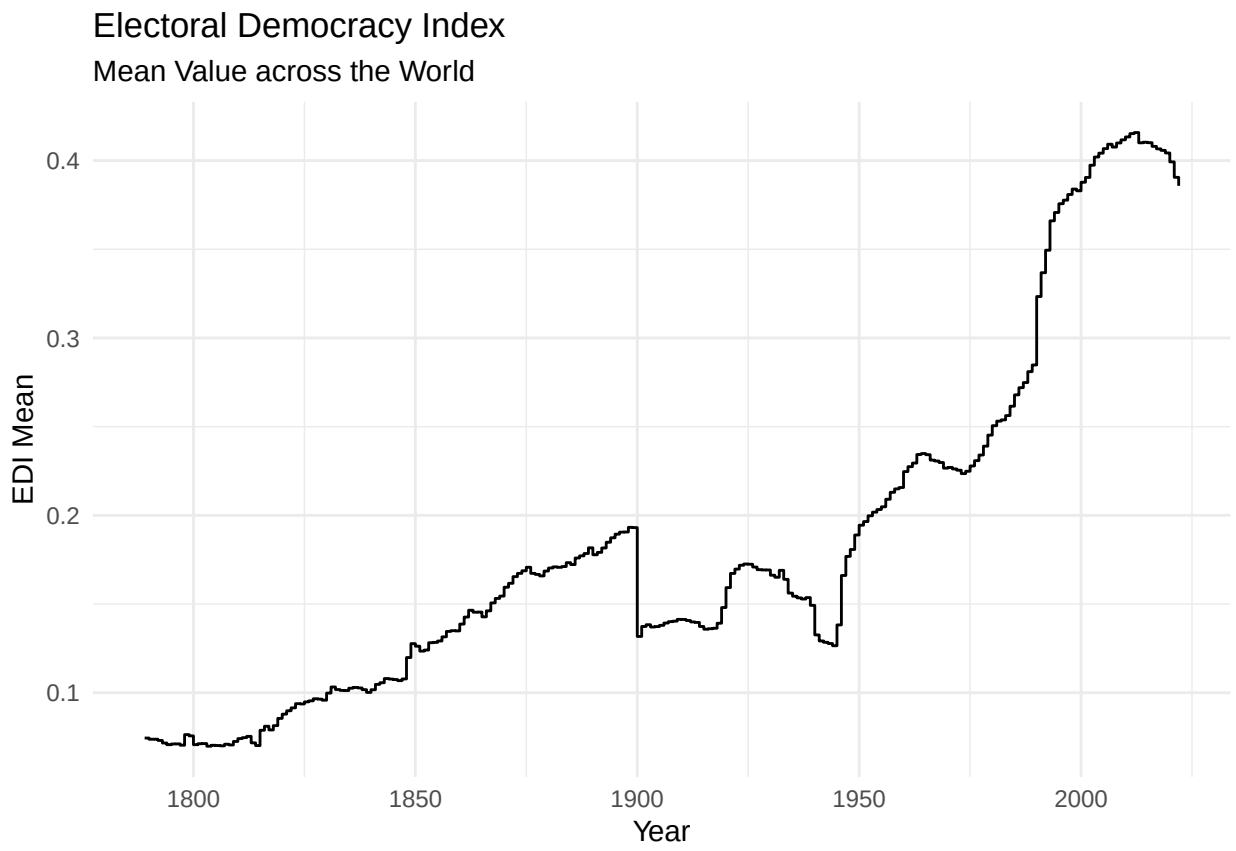



We can change the theme of the whole graphic. GGPlot2 comes with a number of interesting themes.

```
ggplot(data=df_pol, aes(x = year, y = mean_libdem)) +
  geom_step() +
  labs(title = "Electoral Democracy Index",
       subtitle = "Mean Value across the World",
       y = "EDI Mean",
       x = "Year") +
  theme_bw()
```

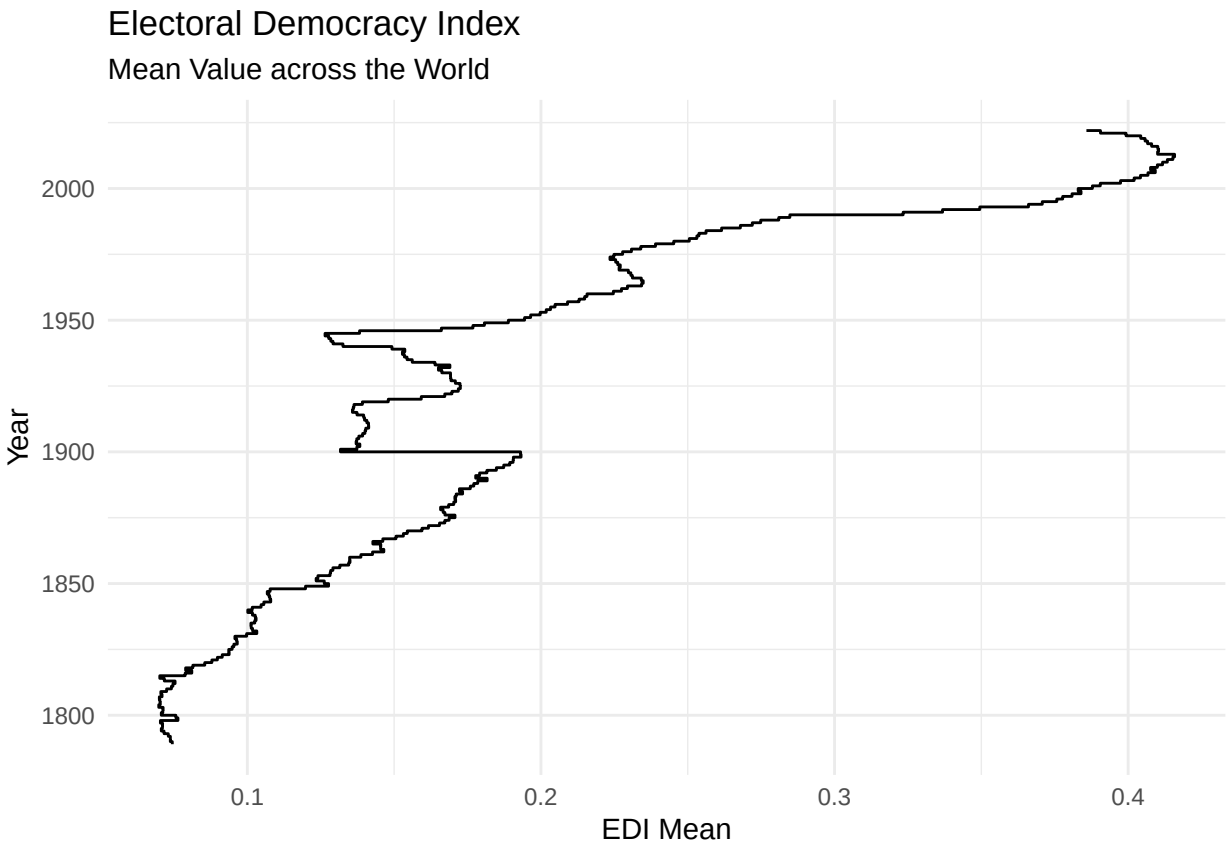



```
ggplot(data=df_pol, aes(x = year, y = mean_libdem)) +
  geom_step() +
  labs(title = "Electoral Democracy Index",
       subtitle = "Mean Value across the World",
       y = "EDI Mean",
       x = "Year") +
  theme_minimal()
```




It does not make much sense in this context, but we can also simply flip the graph.

```
ggplot(data=df_pol, aes(x = year, y = mean_libdem)) +
  geom_step() +
  labs(title = "Electoral Democracy Index",
       subtitle = "Mean Value across the World",
       y = "EDI Mean",
       x = "Year") +
  theme_minimal() +
  coord_flip()
```




## 5. Concluding Remarks

This should give you a good introduction to GGPlot2 and it's logic. Once you have the base, you simply add every detail to your graphic step by step. There are a lot more small details you can adjust in your graphics, but there are too many to cover them all. The idea was to introduce you to the basics to start making your own graphics, that are presentable enough for publication. If you want to save your plots, you can simply click the export button in the plot window. For a cheat sheet, click this Link.

In the next session you will learn about basic programming in R, which will make your life and code a lot easier and faster.

## Exercises: Make your own Plot

This one is bit harder. Using the V-Dem dataset, create boxplots that shows the development of gender equality (`v2clgencl`) for the years 1985 to 2005. Hint: Boxplots only work, if the years are coded as factors.

Now, add labels to the variables and give the graph a title. Give the graph the `classic` theme.