

# Freedom House Data Visualization

Grishma Bhattarai

2/17/2022

```
#installing and calling all the necessary packages
#install.packages("openxlsx")
library(openxlsx)
#install.packages(tidyr)
library("tidyr")
#install.packages("reshape2")
library(reshape2)
```

```
##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##      smiths
```

```
#install.packages(tidyverse)
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5      v dplyr   1.0.7
## v tibble  3.1.5      v stringr 1.4.0
## v readr   2.0.2      v forcats 0.5.1
## v purrr   0.3.4
```

```
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
```

```
#reading the required data into R
```

```
data <- read.xlsx("data.xlsx", sheet = "Country Ratings, Statuses ", startRow = 2, fillMergedCells = TRUE)
```

```
data2 <- data[, -c(2:67)] # delete columns 2 through 67 to drop data that are not between 1995-2020
```

```
#replacing unique column names into year.[PR]/CL/Status names
```

```
names(data2) <- gsub(x = names(data2), pattern = "\\..1", replacement = ".CL")
```

```
names(data2) <- gsub(x = names(data2), pattern = "\\..2", replacement = ".Status")
```

```
#DATA CLEANING
```

```
#pivot data from wide to long except the first row
```

```
data3 <- pivot_longer(data2, (!1), names_to = "Year", values_to = "Value")
```

```
data3 <- rename(data3, "Countries" = "Year(s).Under.Review" )
```

```
data3 <- data3[-c(1:78), ] #delete the first 78 rows as these values are NA formed from the pivot
```

```
data3$Category <- substr(data3$Year, 6, 7) #extract 6th and 7th sub-string from Year variable to get the
```

```
data3$Category[which(data3$Category == "St")] = "Status" #replace "St" with "Status" in Category variable
```

```
data3$Category[which(data3$Category == "")] = "PR" #replace "" with "PR" in Category variable
```

#### *#FINAL DATASET CREATION*

```
#create final data set with clean variable names and re-pivot the table to a wide table  
finaldata <- pivot_wider(data3, names_from = Category, values_from = Value)
```

```
#delete the unnecessary strings from the Year variable and only keep the year value  
finaldata$Year <- substr(finaldata$Year, 1, 4)
```

```
#change necessary character data types into numeric data types
```

```
finaldata$Year <- as.numeric(finaldata$Year)
```

```
finaldata$PR <- as.numeric(finaldata$PR)
```

```
## Warning: NAs introduced by coercion
```

```
finaldata$CL <- as.numeric(finaldata$CL)
```

```
## Warning: NAs introduced by coercion
```

```
#create a new variable 'Year.Range' with 5 year-interval periods
```

```
finaldata$Year.Range = cut(finaldata$Year,seq(1995,2025,5), right = FALSE, left = TRUE) #included 2020-
```

```
#create a graph of share of free, partially free or not free countries in a time interval of 5 years for  
finaldata%>%
```

```
  filter(Status != '-')%>% # remove countries whose data is missing but we include them
```

```
  drop_na(Status)%>% #removing NAs that emerged after data transfiguration
```

```
  ggplot(aes(Year.Range, fill = Status))+
```

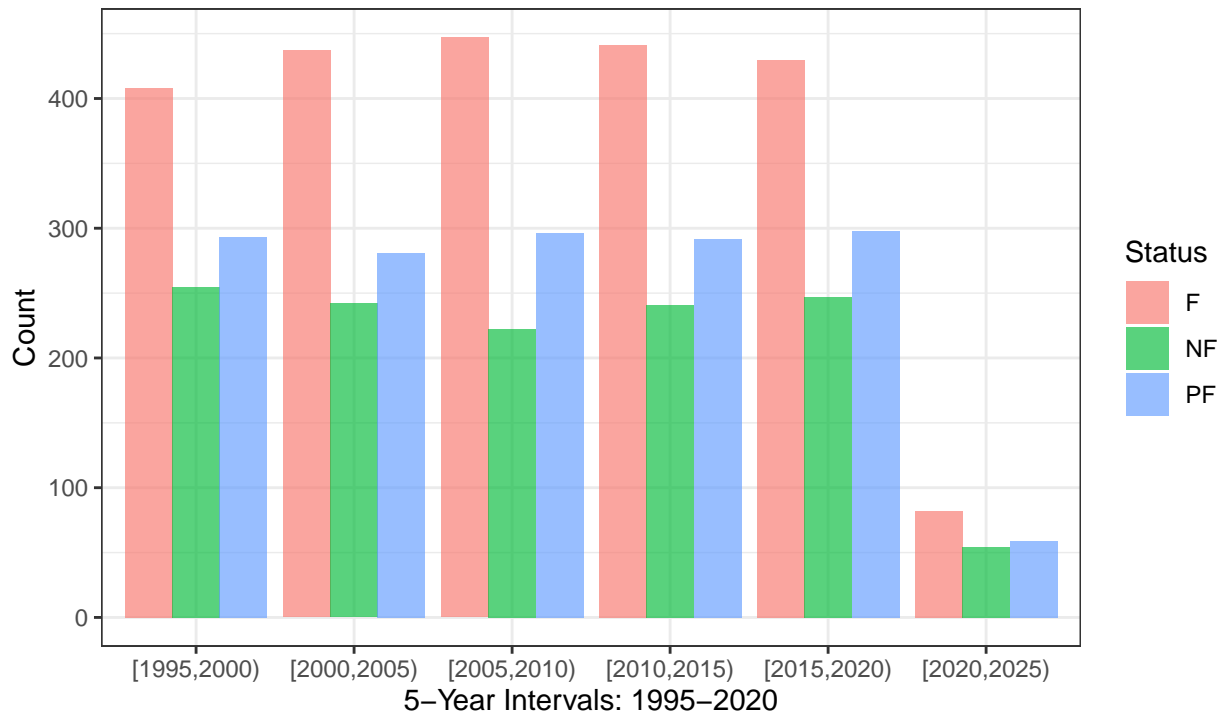
```
  geom_bar(position = "dodge", alpha = 0.65)+
```

```
  theme_bw()+
```

```
  labs(title = "Share of Free (F), Partially Free (PF) or Not Free (NF) Countries", subtitle = "From 19
```

```
    y = "Count", caption = "Source: Country and Territory Ratings and Statuses Data, 1973-2021, Free
```

## Share of Free (F), Partially Free (PF) or Not Free (NF) Countries From 1995–2020



Source: Country and Territory Ratings and Statuses Data, 1973–2021, Freedom House

```
unique <- unique(finaldata[c("Countries", "Year")]) #create a unique dataset with Countries, Year combin
PRdata<- merge(unique, finaldata[,c("Countries", "Year" , "PR")], by = c("Countries", "Year")) #merge t
PRdata <- na.omit(PRdata)

#Repeat the same steps for CL values
CLdata<- merge(unique, finaldata[,c("Countries", "Year" , "CL")], by = c("Countries", "Year")) #merge t
CLdata <- na.omit(CLdata)

Statusdata<- merge(unique, finaldata[,c("Countries", "Year" , "Status")], by = c("Countries", "Year"))
Statusdata <- na.omit(Statusdata)

#merge all 3 datasets into FiW dataset
FiWdata<- merge(PRdata, CLdata, by = c("Countries", "Year"))
FiW<- merge(FiWdata, Statusdata, by = c("Countries", "Year"))

#create new index variable for the Freedom in the World index which is the average of PR and CL, scaled
FiW$Index <- (round((7 - (rowMeans(FiW[,c('PR', 'CL')], na.rm=TRUE))) / (7-1), 3)) #scaling by using th
#following formula: ((e_max - average) / (e_max - e_min))

#create dummy variable for improvement relative to previous year
FiW <- FiW %>%
  group_by(Countries) %>%
  mutate(improv = ifelse(Index > lag(Index), 1, 0))

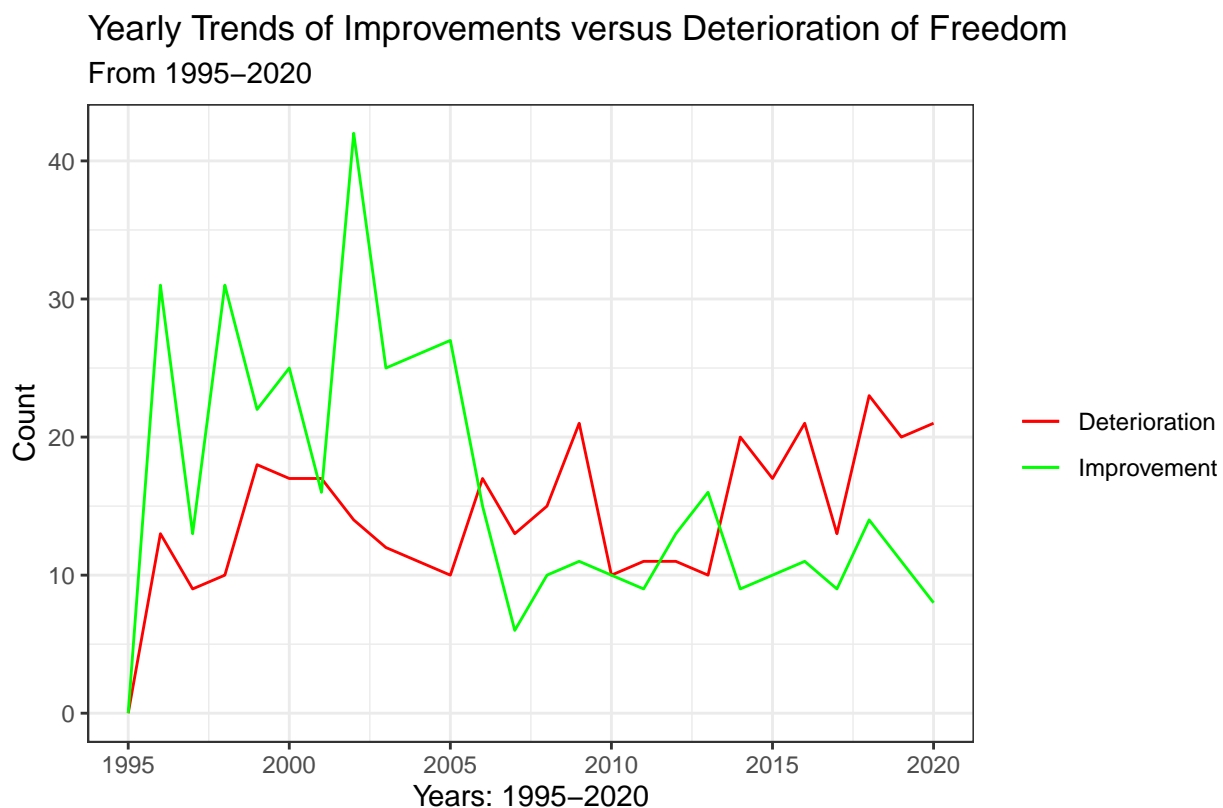
#create dummy variable for worsening relative to previous year
FiW <- FiW %>%
  group_by(Countries) %>%
```

```
mutate(worse = ifelse(Index < lag(Index), 1, 0))

#change NA values to 0 for the dummy variables for convenience, this does not in any way skew our analysis
FiW$improv[is.na(FiW$improv)] <- 0
FiW$worse[is.na(FiW$worse)] <- 0

#create a new dataset 'countdata' that groups by year and calculates the count of better-off and worse-off
countdata <-
  FiW %>%
  group_by(Year) %>%
  summarise(worsecount = sum(worse), improvcount = sum(improv))

#create a line plot of yearly trends of improvements versus deterioration of freedom
ggplot(data = countdata, aes(x = Year)) +
  geom_line(aes(y = worsecount, colour = "Deterioration")) +
  geom_line(aes(y = improvcount, colour = "Improvement")) +
  scale_colour_manual("",
    breaks = c("Deterioration", "Improvement"),
    values = c("red", "green")) +
  xlab('Years: 1995–2020') + theme_bw() +
  ylab('Count') + labs(title = "Yearly Trends of Improvements versus Deterioration of Freedom", subtitle = "From 1995–2020")
```



Source: Country and Territory Ratings and Statuses Data, 1973–2021, Freedom House

```
#install and call package countrycode for merging two datasets based on country codes
#install.packages("countrycode")
library(countrycode)

#read the required UN data into R
undata <- read.xlsx("undata.xlsx", startRow = 1, fillMergedCells = TRUE, colNames = TRUE, detectDates = TRUE)
```

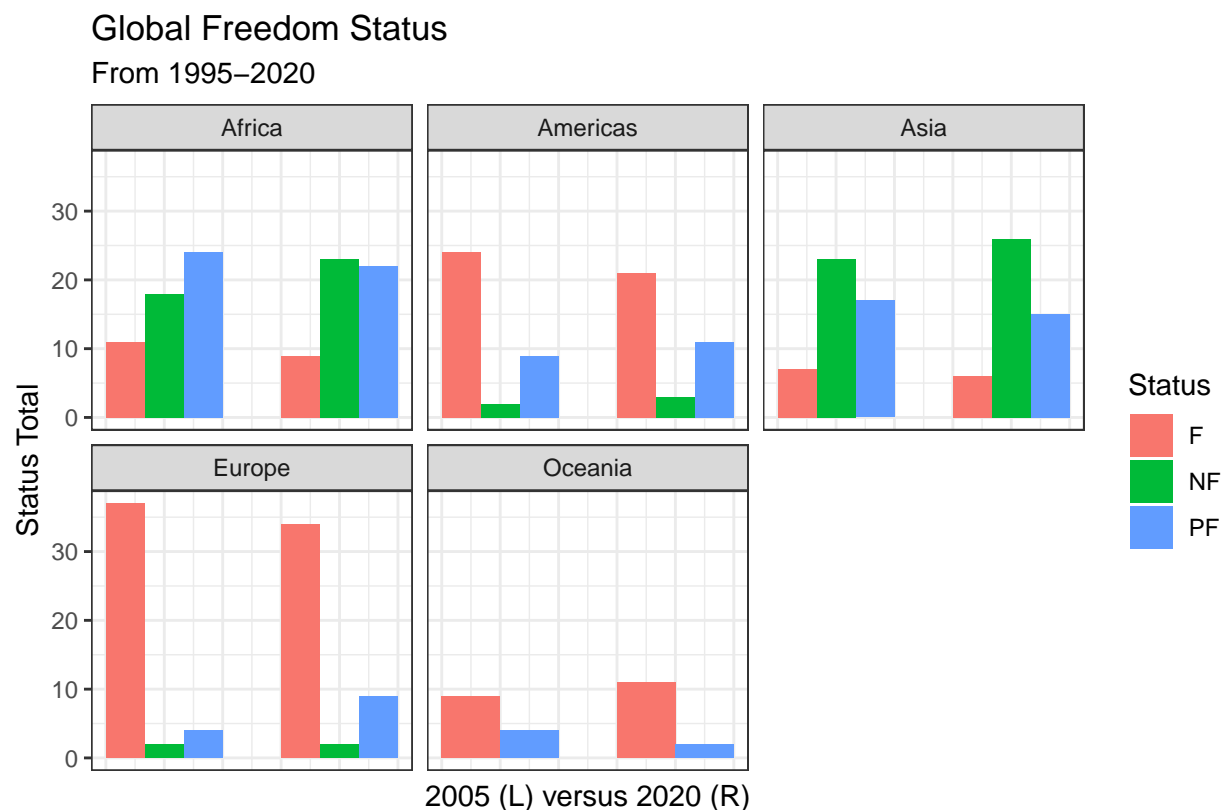
```
colnames(undata)[which(names(undata) == "Country.or.Area")] <- "Countries"

#create a new variable code in 'finaldata' dataset with ISO 3-C coding classification system
FiW$code <- countrycode(FiW$Countries, "country.name", "iso3c")

## Warning in countrycode_convert(sourcevar = sourcevar, origin = origin, destination = dest, : Some va
undata$code <- countrycode(undata$Countries, "country.name", "iso3c")

## Warning in countrycode_convert(sourcevar = sourcevar, origin = origin, destination = dest, : Some va
#merge the undata and FiWdata
mergeddata <- merge(FiW, undata, by= c("code"))

# use subset function to create a dataset exclusively for years 2005 and 2020
yearsusb <- subset(mergeddata, Year == 2005 | Year == 2020)
#create a graph that compares status levels across regions for 2005 and 2020
yearsusb %>%
  drop_na(Status)%>%
  ggplot(aes(Year, fill = Status)) + theme_bw() +
  geom_bar(position = "dodge", width = 10) + facet_wrap(Region.Name~.) +
  theme(axis.text.x = element_blank(), axis.ticks.x = element_blank()) +
  labs(x = "2005 (L) versus 2020 (R)", y = "Status Total", title = "Global Freedom Status", subtitle =
```



Freedom House Country and Territory Ratings and Statuses Data & United Nations Geoscheme Data

```
#rename the required column to 'LDC'
colnames(mergeddata)[which(names(mergeddata) == "Least.Developed.Countries.(LDC)")] <- "LDC"
#subset average value of index for LDC's and other countries
averageindex <- mergeddata %>%
  group_by(Year, LDC) %>%
```

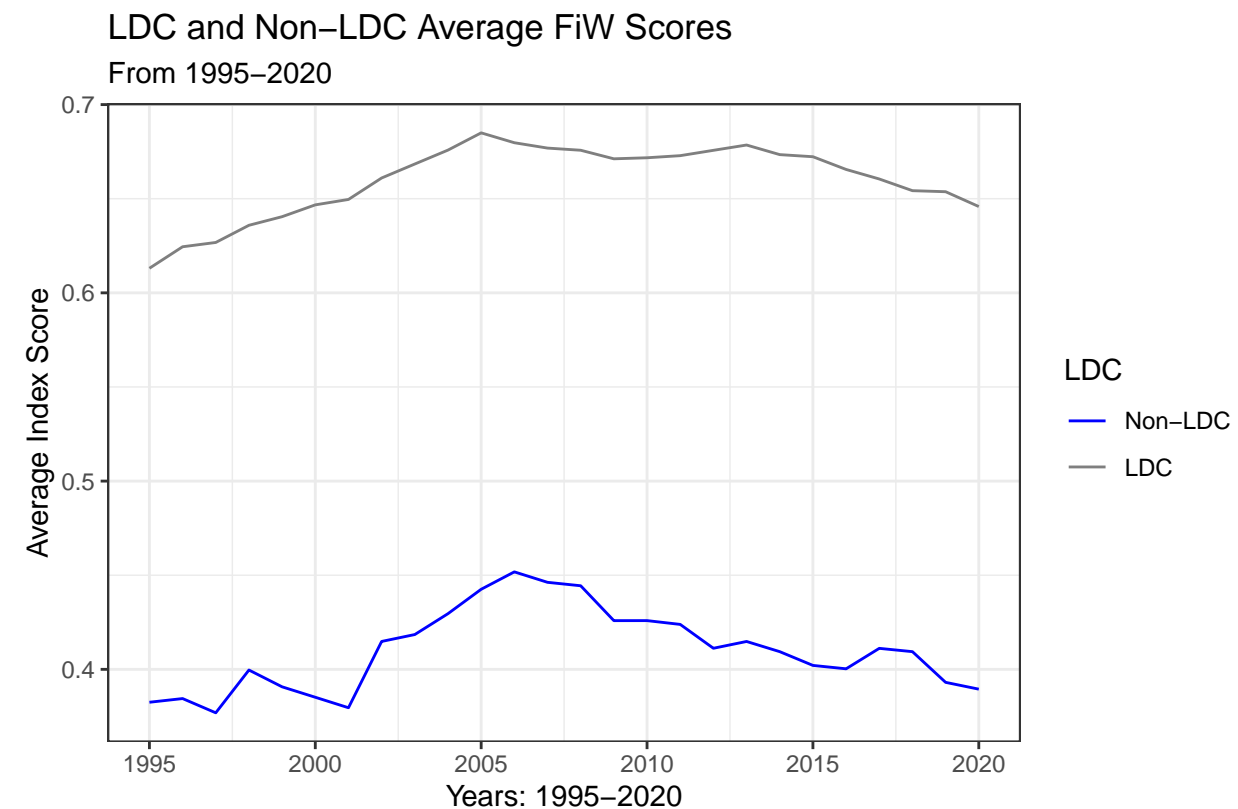
```

summarise(avgindex = mean(Index), n = n())

## `summarise()` has grouped output by 'Year'. You can override using the `.groups` argument.
#dummy code variable LDC for a clean graph
averageindex <- averageindex %>%
  mutate(LDC = ifelse(LDC == "x", 1, 0))

averageindex$LDC <- factor(averageindex$LDC) #change the numeric LDC variable into factor data type
#create a
ggplot(data = averageindex) + theme_bw()+
  geom_line(aes(x = Year, y = avgindex, colour = LDC, group = LDC)) + labs(y = "Average Index Score", x = "Years: 1995–2020") +
  scale_color_manual(labels = c("Non-LDC", "LDC"), values = c("blue", "red")) + labs(subtitle = "From 1995–2020")

```



From House Country and Territory Ratings and Statuses Data & United Nations Geoscheme Data