QALL401: Data Analysis for Researchers

Course Contents

1. 2022.12.07: Introduction: About the course [lead by TK]
   - An introduction to open and public data, and data science
2. 2022-12-14: Exploratory Data Analysis (EDA) 1 [lead by hs]
   - R Basics with RStudio and/or RStudio.cloud; Toy Data
3. 2022-12-21: Exploratory Data Analysis (EDA) 2 [lead by hs]
   - R Markdown, `tidyverse` I: `dplyr`; `gapminder`
4. 2023-01-11: Exploratory Data Analysis (EDA) 3 [lead by hs]
   - `tidyverse`II: `readr`, `ggplot2`; Public Data, WDI, WIR, etc
5. **2023-01-18: Exploratory Data Analysis (EDA) 4 [lead by hs]**
   - `tidyverse` III: `tidyr`, etc.; WDI, WIR, etc
6. 2023-01-25: Exploratory Data Analysis (EDA) 5 [lead by hs]
   - `tidyverse` IV; WDI, WIR, etc
7. 2023-02-01: Introduction to PPDAC
   - Problem-Plan-Data-Analysis-Conclusion Cycle: [lead by TK]
8. 2023-02-08: Model building I [lead by TK]
   - Collecting and visualizing data and Introduction to WDI
     (World Development Indicators by World Bank)
9. 2023-02-15: Model building II [lead by TK]
   - Analyzing data and communications
10. 2023-02-22: Project Presentation

1 Exploratory Data Analysis (EDA) I

2 Exploratory Data Analysis II

3 Exploratory Data Analysis III

4 Exploratory Data Analysis (EDA) IV

4.1 Tidy Data

4.1.1 Reviews and Previews

4.1.2 Example: World Inequility Report - WIR2022

- World Inequality Report: https://wir2022.wid.world/
- Executive Summary: https://wir2022.wid.world/executive-summary/
- Methodology: https://wir2022.wid.world/methodology/
- Data URL: https://wir2022.wid.world/www-site/uploads/2022/03/WIR2022TablesFigures-Summary.xlsx

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   1.0.0
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(readxl)
```

```
url_summary <- "https://wir2022.wid.world/www-site/uploads/2022/03/WIR2022TablesFigures-Summary.xlsx"
download.file(url = url_summary, destfile = "data/WIR2022s.xlsx")
```

```
excel_sheets("data/WIR2022s.xlsx")
```

```
##  [1] "Index"     "F1"        "F2"        "F3"        "F4"        "F5."
##  [7] "F6"        "F7"        "F8"        "F9"        "F10"       "F11"
## [13] "F12"       "F13"       "F14"       "F15"       "T1"        "data-F1"
## [19] "data-F2"   "data-F3"   "data-F4"   "data-F5"   "data-F6"   "data-F7"
## [25] "data-F8"   "data-F9"   "data-F10"  "data-F11"  "data-F12"  "data-F13."
## [31] "data-F14." "data-F15"
```

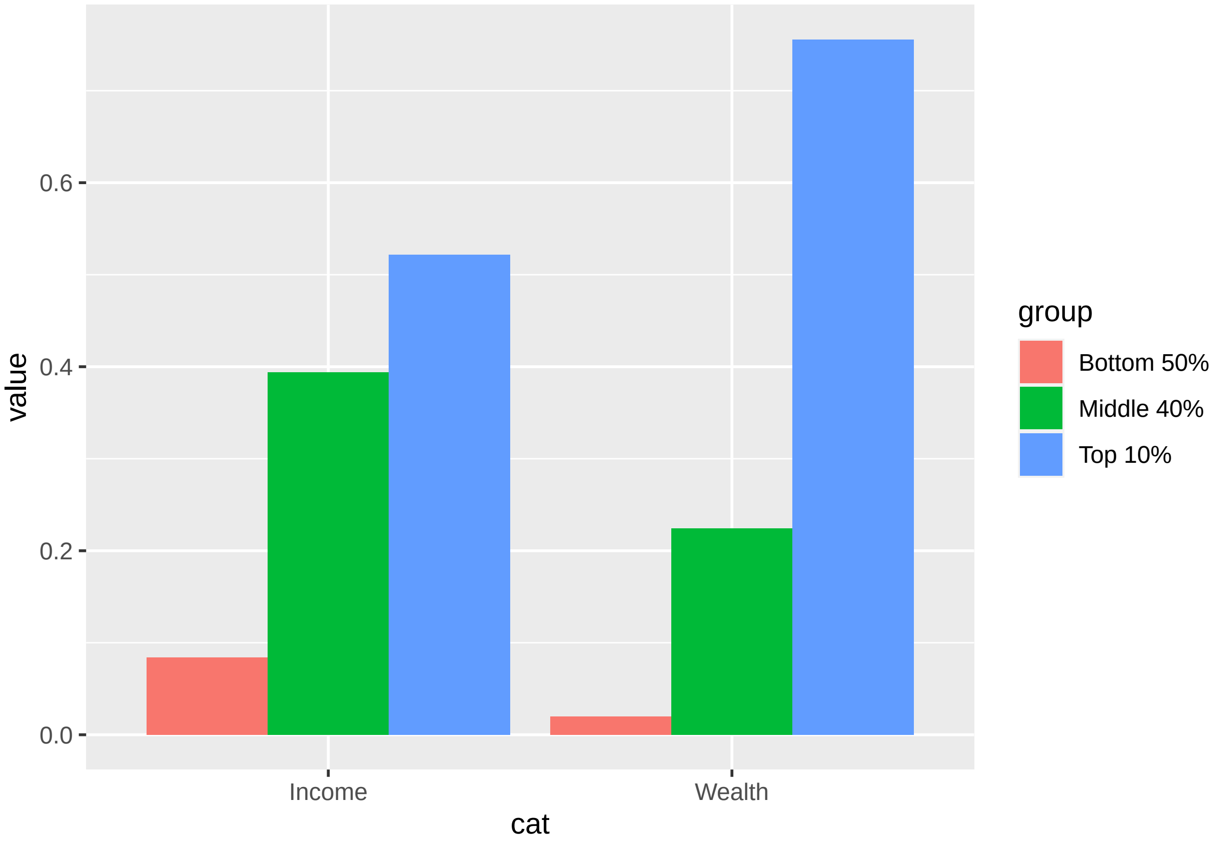
4.1.3 F1: Global income and wealth inequality, 2021

```
df_f1 <- read_excel("data/WIR2022s.xlsx", sheet = "data-F1")
df_f1
```

```
## # A tibble: 2 x 5
##   ...1   `Bottom 50%` `Middle 40%` `Top 10%` `Top 1%`
##   <chr>         <dbl>        <dbl>     <dbl>    <dbl>
## 1 Income       0.084         0.394     0.522    0.192
## 2 Wealth       0.0199        0.224     0.756    0.378
```

```
## # A tibble: 6 x 3
##   cat    group       value
##   <chr>  <chr>       <dbl>
## 1 Income Bottom 50% 0.084
## 2 Income Middle 40% 0.394
## 3 Income Top 10%    0.522
## 4 Wealth Bottom 50% 0.0199
## 5 Wealth Middle 40% 0.224
## 6 Wealth Top 10%    0.756
```

```
df_f1_rev %>%
  ggplot(aes(x = cat, y = value, fill = group)) +
  geom_col(position = "dodge")
```



4.1.4 References of `tidyr`

- Textbook: R for Data Science,Tidy Data

4.1.4.1 RStudio Primers: See References in Moodle at the bottom Tidy Your Data

- Reshape Data
- Separate and Unite Columns
- Join Data Sets

4.1.5 Variables, values, and observations: Definitions

- A **variable** is a quantity, quality, or property that you can measure.
- A **value** is the state of a variable when you measure it. The value of a variable may change from measurement to measurement.
- An **observation** or **case** is a set of measurements made under similar conditions (you usually make all of the measurements in an observation at the same time and on the same object). An observation will contain several values, each associated with a different variable. I'll sometimes refer to an observation as a case or data point.
- **Tabular data** is a table of values, each associated with a variable and an observation. Tabular data is tidy if each value is placed in its own cell, each variable in its own column, and each observation in its own row.
- So far, all of the data that you've seen has been tidy. In real-life, most data isn't tidy, so we'll come back to these ideas again in Data Wrangling.

4.1.6 Tidy Data

> "Data comes in many formats, but R prefers just one: tidy data." — Garrett Grolemund

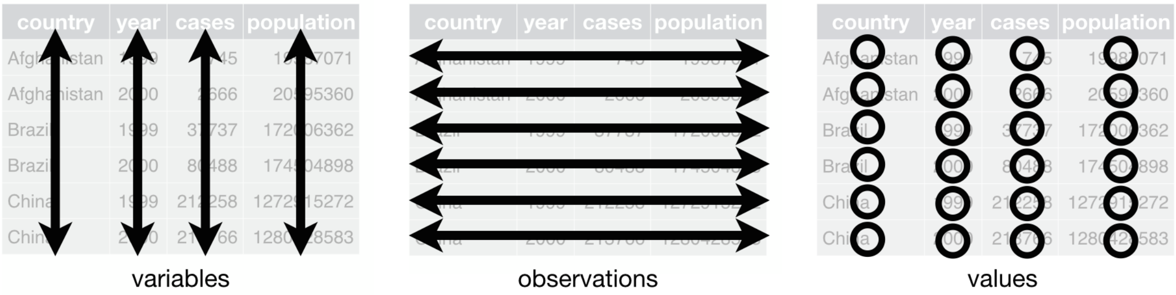
Data can come in a variety of formats, but one format is easier to use in R than the others. This format is known as tidy data. A data set is tidy if:

1. Each variable is in its own column
2. Each observation is in its own row
3. Each value is in its own cell (this follows from #1 and #2)

> "Tidy data sets are all alike; but every messy data set is messy in its own way." — Hadley Wickham

> "all happy families are all alike; each unhappy family is unhappy in its own way" - Tolstoy's Anna Karenina

4.1.7 `tidyr` Basics



1. Each variable is in its own column
2. Each observation is in its own row

4.1.8 Pivot data from wide to long: `pivot_longer()`

```
pivot_longer(data, cols = <columns to pivot into longer format>,
  names_to = <name of the new character column>, # e.g. "group", "category", "class"
  values_to = <name of the column the values of cells go to>) # e.g. "value", "n"
```

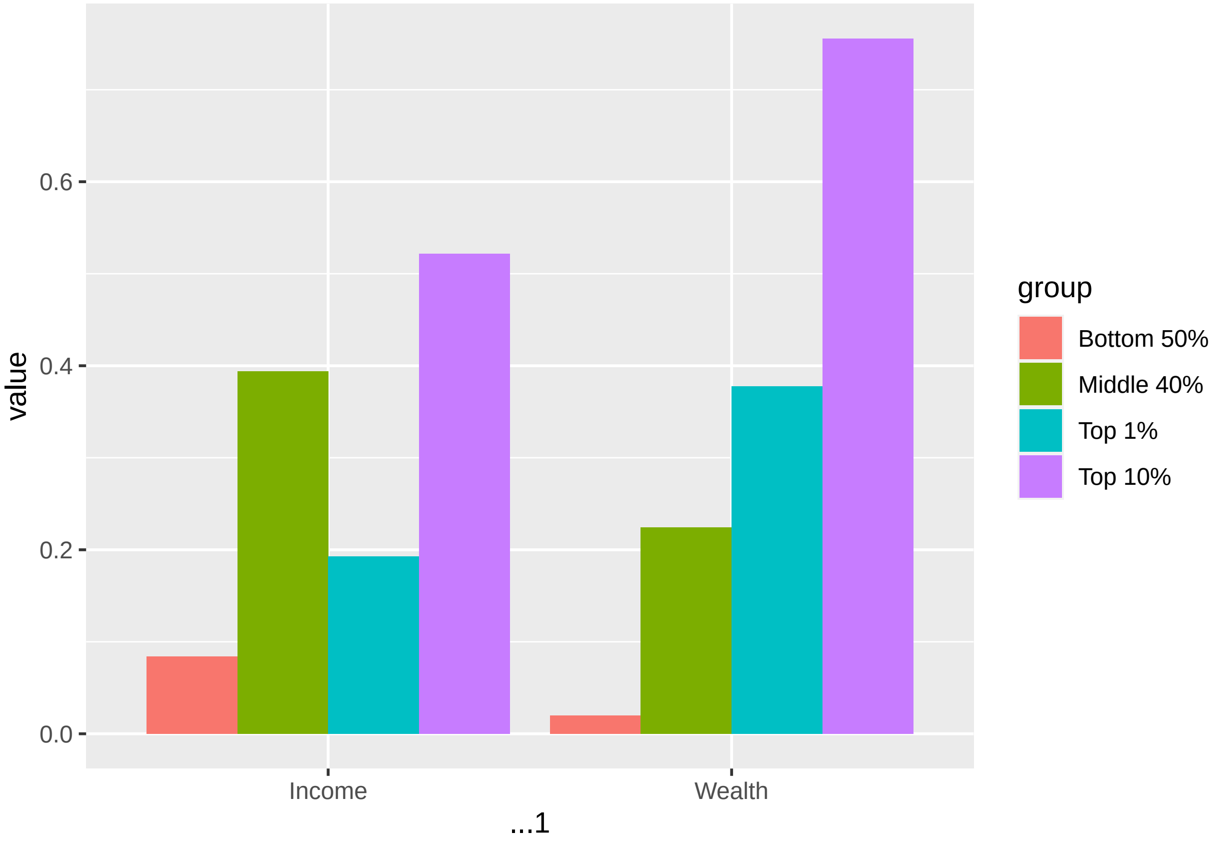
```
df_f1
```

```
## # A tibble: 2 x 5
##   ...1   `Bottom 50%` `Middle 40%` `Top 10%` `Top 1%`
##   <chr>         <dbl>        <dbl>     <dbl>    <dbl>
## 1 Income       0.084         0.394     0.522    0.192
## 2 Wealth       0.0199        0.224     0.756    0.378
```

```
(df_f1_rev <- df_f1 %>% pivot_longer(-1, names_to = "group", values_to = "value"))
```

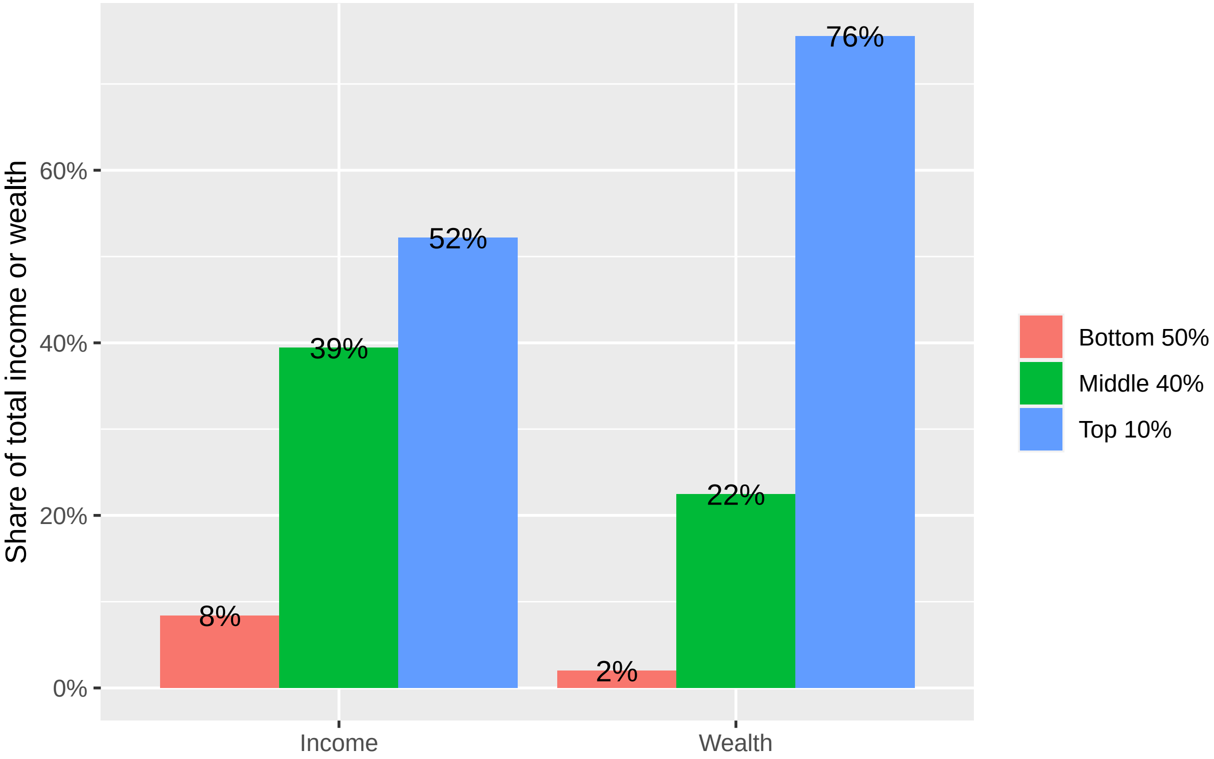
```
## # A tibble: 8 x 3
##   ...1   group        value
##   <chr>  <chr>       <dbl>
## 1 Income Bottom 50% 0.084
## 2 Income Middle 40% 0.394
## 3 Income Top 10%    0.522
## 4 Income Top 1%     0.192
## 5 Wealth Bottom 50% 0.0199
## 6 Wealth Middle 40% 0.224
## 7 Wealth Top 10%    0.756
## 8 Wealth Top 1%     0.378
```

```
df_f1_rev %>%
  ggplot(aes(x = ...1, y = value, fill = group)) +
  geom_col(position = "dodge")
```



```
df_f1_rev %>% filter(group != "Top 1%") %>%
  ggplot() +
  geom_col(aes(x = ...1, y = value, fill = group), position = "dodge") +
  geom_text(aes(x = ...1, y = value, group = group,
          label = scales::label_percent(accuracy=1)(value)),
          position = position_dodge(width = 0.9)) +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
  labs(title = "Figure 1. Global income and wealth inequality, 2021",
       x = "", y = "Share of total income or wealth", fill = "")
```

Figure 1. Global income and wealth inequality, 2021



Interpretation: The global bottom 50% captures 8.5% of total income measured at Purchasing Power Parity (PPP). The global bottom 50% owns 2% of wealth (at Purchasing Power Parity). The global top 10% owns 76% of total Household wealth and captures 52% of total income in 2021. Note that top wealth holders are not necessarily top income holders. Incomes are measured after the operation of pension and unemployment systems and before taxes and transfers.
Sources and series: wir2022.wid.world/methodology.

4.1.9 F2: The poorest half lags behind: Bottom 50%, middle 40% and top 10% income shares across the world in 2021

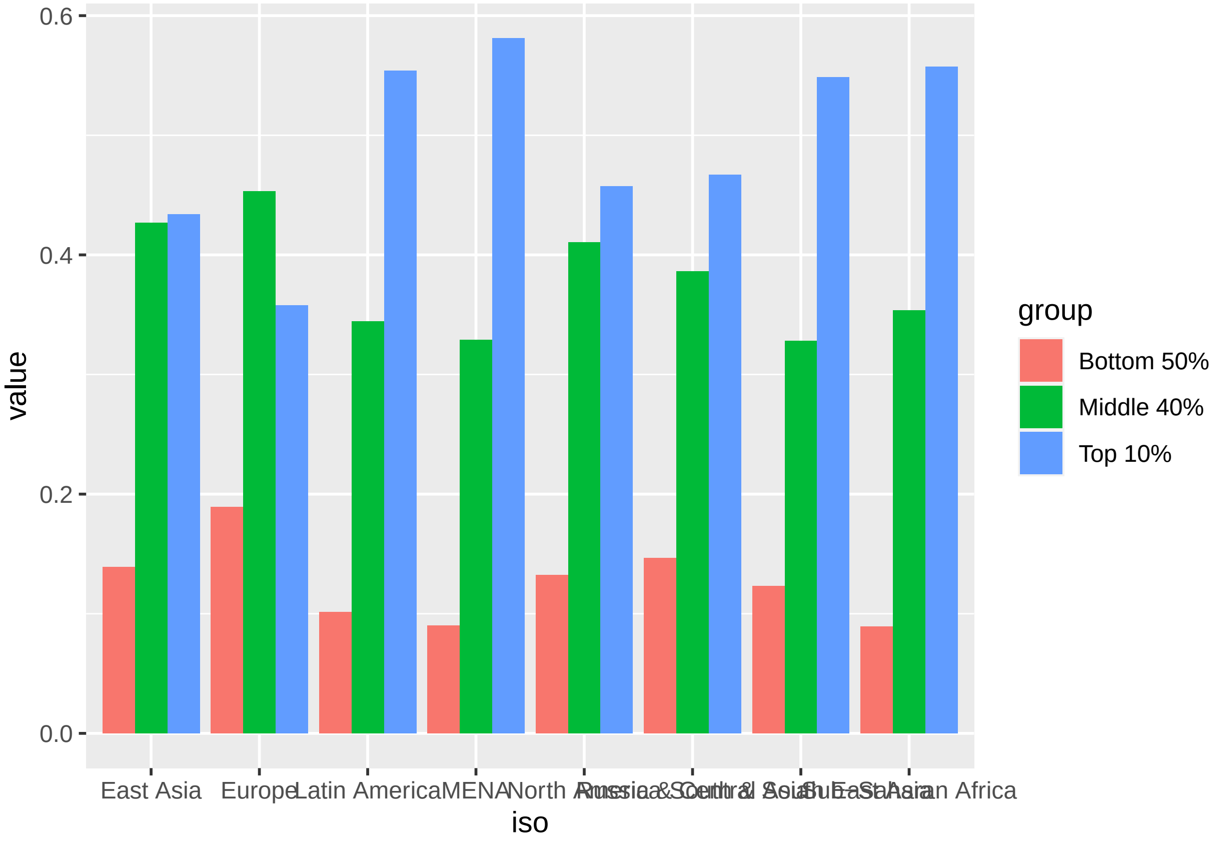
```r
df_f2 <- read_excel("data/WIR2022s.xlsx", sheet = "data-F2")
df_f2
```

```
## # A tibble: 8 x 5
##    year iso                     `Bottom 50%` `Middle 40%` `Top 10%`
##   <dbl> <chr>                          <dbl>        <dbl>     <dbl>
## 1  2021 Europe                        0.189         0.453     0.358
## 2  2021 East Asia                     0.139         0.427     0.434
## 3  2021 North America                 0.132         0.411     0.457
## 4  2021 Russia & Central Asia         0.147         0.386     0.467
## 5  2021 South & South East Asia       0.123         0.328     0.548
## 6  2021 Latin America                 0.102         0.344     0.554
## 7  2021 Sub-Saharan Africa            0.0892        0.354     0.557
## 8  2021 MENA                          0.09          0.329     0.581
```

```
df_f2 %>% pivot_longer(cols = 3:5, names_to = "group", values_to = "value")
```

```
## # A tibble: 24 x 4
##     year iso                   group      value
##    <dbl> <chr>                 <chr>      <dbl>
##  1  2021 Europe                Bottom 50% 0.189
##  2  2021 Europe                Middle 40% 0.453
##  3  2021 Europe                Top 10%    0.358
##  4  2021 East Asia             Bottom 50% 0.139
##  5  2021 East Asia             Middle 40% 0.427
##  6  2021 East Asia             Top 10%    0.434
##  7  2021 North America         Bottom 50% 0.132
##  8  2021 North America         Middle 40% 0.411
##  9  2021 North America         Top 10%    0.457
## 10  2021 Russia & Central Asia Bottom 50% 0.147
## # ... with 14 more rows
```

```
df_f2 %>% pivot_longer(cols = 3:5, names_to = "group", values_to = "value") %>%
  ggplot(aes(x = iso, y = value, fill = group)) +
  geom_col(position = "dodge")
```



4.1.10 Pivot data from long to wide:

`pivot_wider()` In Console: vignette(“pivot”)

```
pivot_wider(data,
  names_from = <name of the column (or columns) to get the name of the output column>,
  values_from = <name of the column to get the value of the output>)
```

```
## # A tibble: 24 x 4
##     year iso                    group      value
##    <dbl> <chr>                  <chr>      <dbl>
##  1  2021 Europe                 Bottom 50% 0.189
##  2  2021 Europe                 Middle 40% 0.453
##  3  2021 Europe                 Top 10%    0.358
##  4  2021 East Asia              Bottom 50% 0.139
##  5  2021 East Asia              Middle 40% 0.427
##  6  2021 East Asia              Top 10%    0.434
##  7  2021 North America          Bottom 50% 0.132
##  8  2021 North America          Middle 40% 0.411
##  9  2021 North America          Top 10%    0.457
## 10  2021 Russia & Central Asia Bottom 50% 0.147
## # ... with 14 more rows
```

```
pivot_wider(data, names_from = group, values_from = value)
```

4.1.11 Practice: F4 and F13

F4 and F13 are similar. Please use `pivot_longer` to tidy the data and create charts.

- **References**: https://ds-sl.github.io/data-analysis/wir2022.nb.html

4.1.11.1 Done Last Week

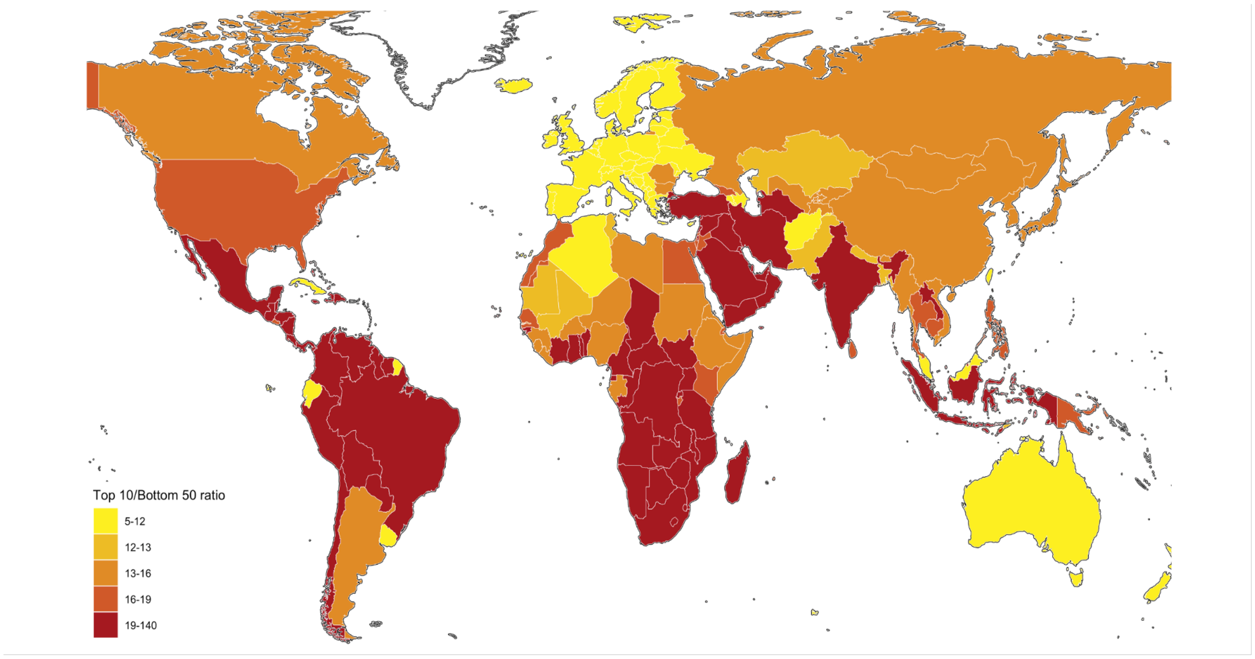
- F12: Female share in global labor incomes, 1990-2020
- F14: Global carbon inequality, 2019. Group contribution to world emissions (%)

4.1.12 F3: Top 10/Bottom 50 income gaps across the world, 2021

```
df_f3 <- read_excel("data/WIR2022s.xlsx", sheet = "data-F3")
df_f3
```

```
## # A tibble: 177 x 3
##     year Country                 T10B50
##    <dbl> <chr>                    <dbl>
##  1  2021 United Arab Emirates     19.2
##  2  2021 Afghanistan              11.7
##  3  2021 Albania                   8.99
##  4  2021 Armenia                  11.0
##  5  2021 Angola                   32.1
##  6  2021 Argentina                13.2
##  7  2021 Austria                   7.68
##  8  2021 Australia                10.4
##  9  2021 Azerbaijan                9.63
## 10  2021 Bosnia and Herzegovina   9.32
## # ... with 167 more rows
```

4.1.13 F3: Top 10/Bottom 50 income gaps across the world, 2021 - Original



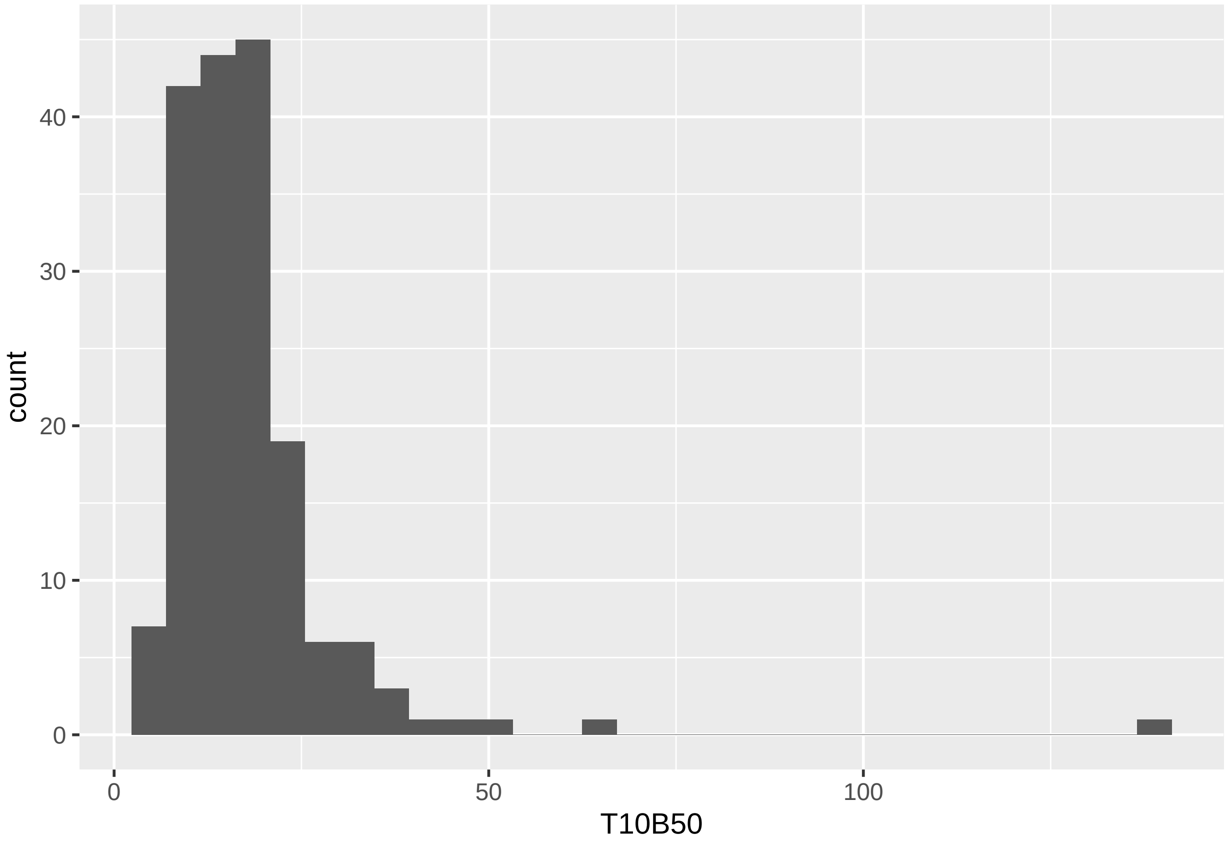
- To 10 / Bottom 50 ratio has 5 classes: 5-12, 12-13, 13-16, 16-19, 19-140

```
df_f3$T10B50 %>% summary()
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   5.394  10.958  15.676  17.635  19.838 139.591
```

```
df_f3 %>% ggplot() + geom_histogram(aes(T10B50))
```

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



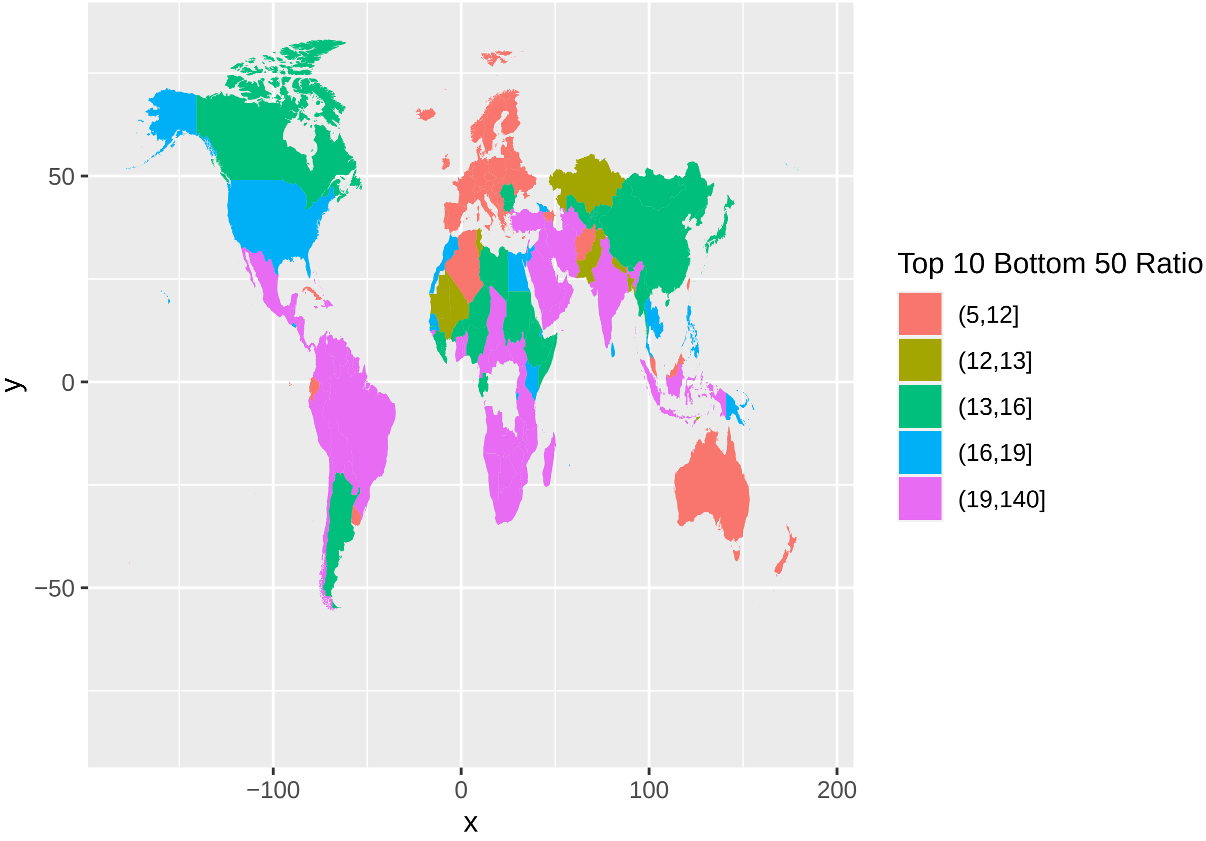
```
df_f3 %>% arrange(desc(T10B50))
```

```
## # A tibble: 177 x 3
##     year Country                  T10B50
##    <dbl> <chr>                     <dbl>
##  1  2021 Oman                      140. 
##  2  2021 South Africa               63.1
##  3  2021 Namibia                    49.0
##  4  2021 Zambia                     44.4
##  5  2021 Central African Republic   42.5
##  6  2021 Mozambique                 38.9
##  7  2021 Swaziland                  38.1
##  8  2021 Botswana                   36.5
##  9  2021 Angola                     32.1
## 10  2021 Yemen                      32.0
## # ... with 167 more rows
```

```
df_f3 %>%
  mutate(`Top 10 Bottom 50 Ratio` = cut(T10B50,breaks = c(5, 12, 13, 16, 19,140),
                                        include.lowest = FALSE))
```

```
## # A tibble: 177 x 4
##     year Country                T10B50 `Top 10 Bottom 50 Ratio`
##    <dbl> <chr>                   <dbl> <fct>
##  1  2021 United Arab Emirates    19.2  (19,140]
##  2  2021 Afghanistan            11.7  (5,12]
##  3  2021 Albania                 8.99 (5,12]
##  4  2021 Armenia                11.0  (5,12]
##  5  2021 Angola                 32.1  (19,140]
##  6  2021 Argentina              13.2  (13,16]
##  7  2021 Austria                 7.68 (5,12]
##  8  2021 Australia              10.4  (5,12]
##  9  2021 Azerbaijan              9.63 (5,12]
## 10  2021 Bosnia and Herzegovina   9.32 (5,12]
## # ... with 167 more rows
```

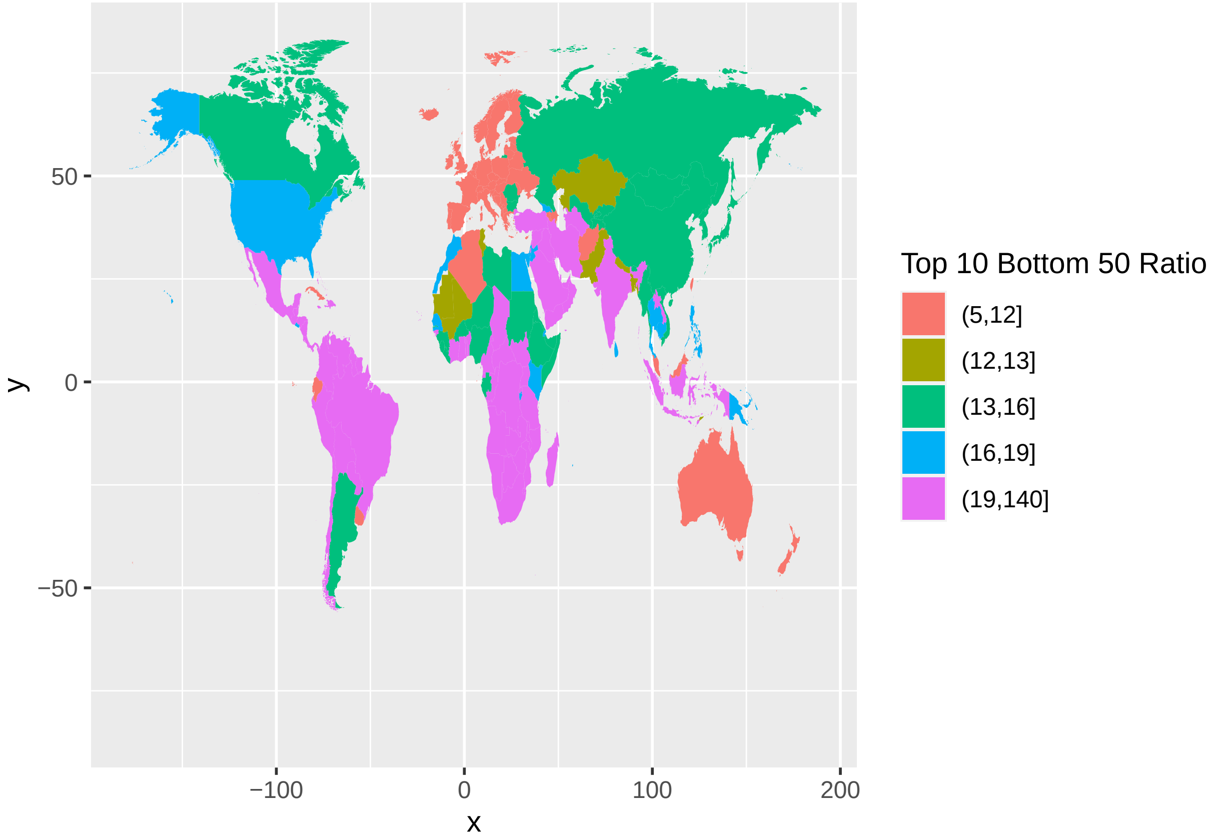
```
world_map <- map_data("world")
df_f3 %>% mutate(`Top 10 Bottom 50 Ratio` = cut(T10B50,breaks = c(5, 12, 13, 16, 19,140),
                                        include.lowest = FALSE)) %>%
  ggplot(aes(map_id = Country)) +
  geom_map(aes(fill = `Top 10 Bottom 50 Ratio`), map = world_map) +
  expand_limits(x = world_map$long, y = world_map$lat)
```



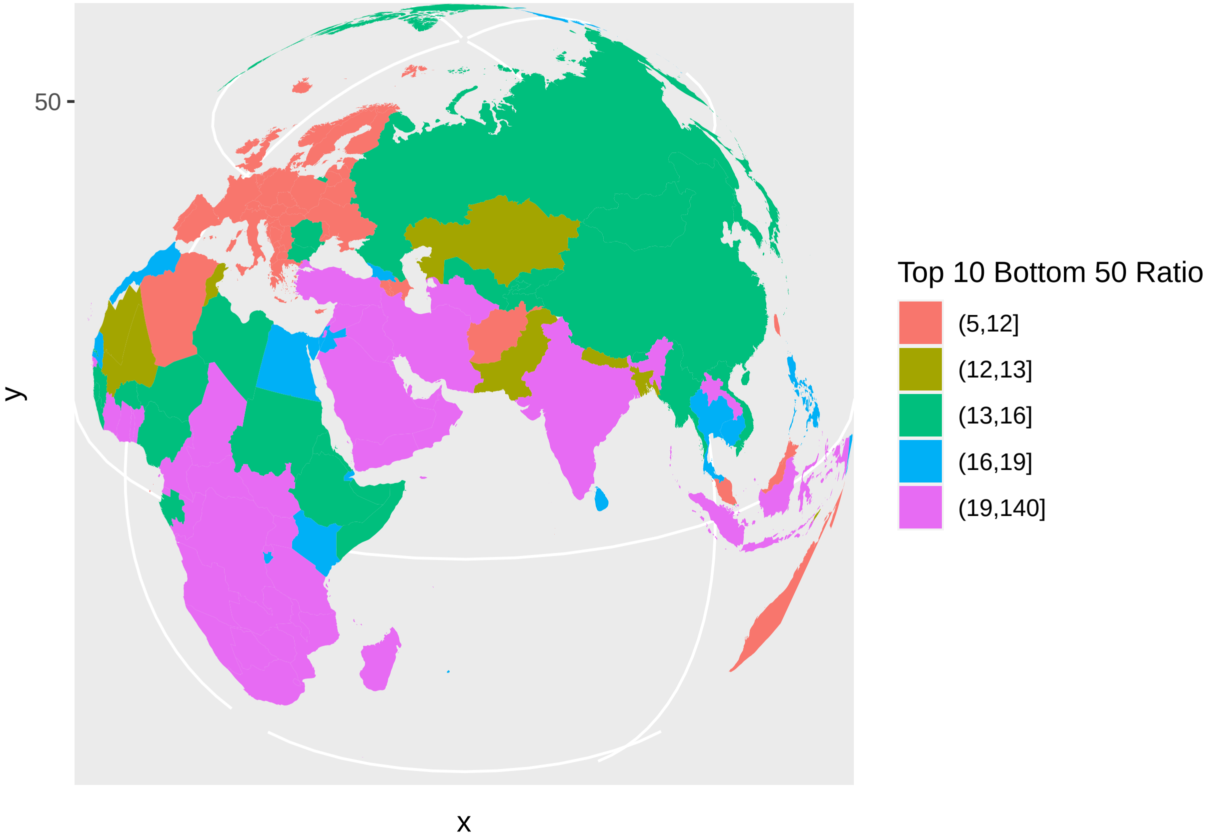
```
world_map_wir <- world_map
world_map_wir$region[
  world_map_wir$region=="Democratic Republic of the Congo"]<-"DR Congo"
world_map_wir$region[world_map_wir$region=="Republic of Congo"]<-"Congo"
world_map_wir$region[world_map_wir$region=="Ivory Coast"]<-"Cote dIvoire"
world_map_wir$region[world_map_wir$region=="Vietnam"]<-"Viet Nam"
world_map_wir$region[world_map_wir$region=="Russia"]<-"Russian Federation"
```

```
world_map_wir$region[world_map_wir$region=="South Korea"]<-"Korea"
world_map_wir$region[world_map_wir$region=="UK"]<-"United Kingdom"
world_map_wir$region[world_map_wir$region=="Brunei"]<-"Brunei Darussalam"
world_map_wir$region[world_map_wir$region=="Laos"]<-"Lao PDR"
world_map_wir$region[world_map_wir$region=="Cote dIvoire"]<-"Cote d'Ivoire"
world_map_wir$region[world_map_wir$region=="Cape Verde"]<- "Cabo Verde"
world_map_wir$region[world_map_wir$region=="Syria"]<- "Syrian Arab Republic"
world_map_wir$region[world_map_wir$region=="Trinidad"]<- "Trinidad and Tobago"
world_map_wir$region[world_map_wir$region=="Tobago"]<- "Trinidad and Tobago"
```

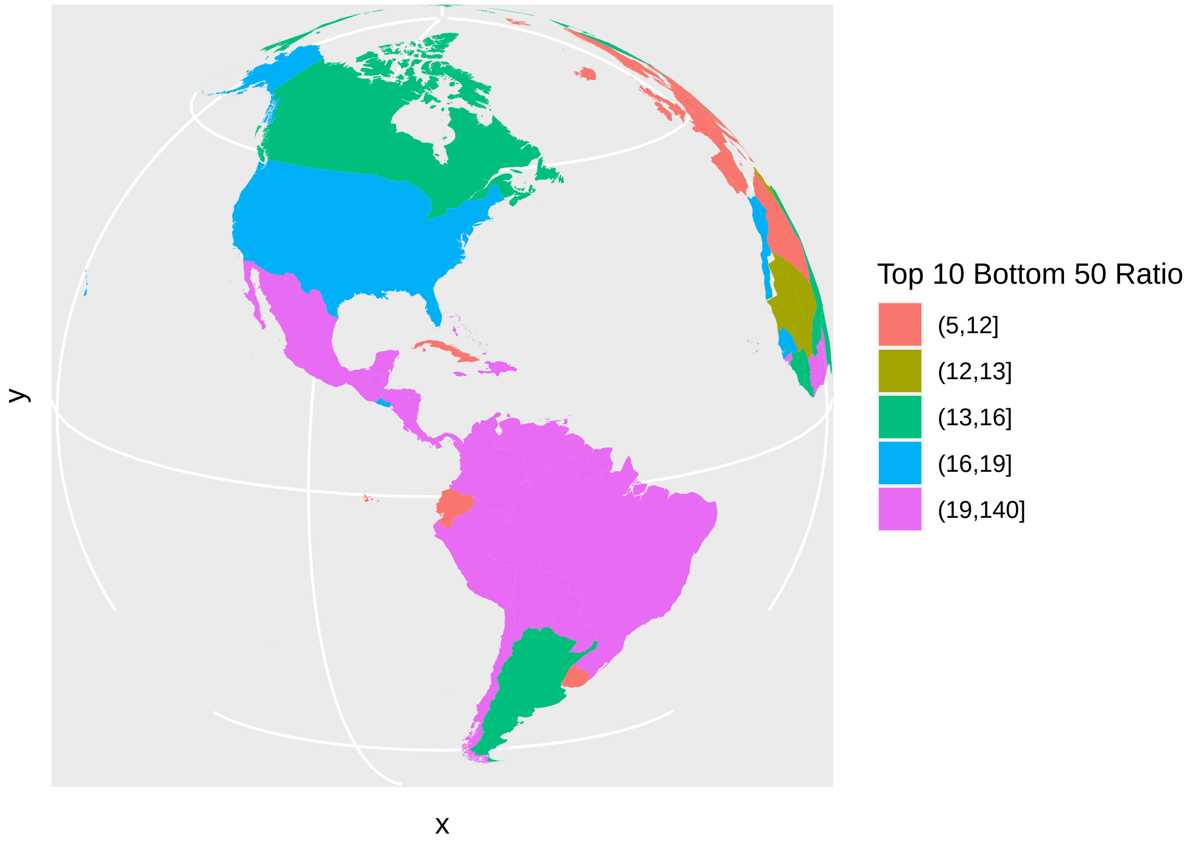
```
df_f3 %>% mutate(`Top 10 Bottom 50 Ratio` =
    cut(T10B50, breaks = c(5, 12, 13, 16, 19,140), include.lowest = FALSE)) %>%
  ggplot(aes(map_id = Country)) +
  geom_map(aes(fill = `Top 10 Bottom 50 Ratio`),
    map = world_map_wir) +
    expand_limits(x = world_map_wir$long, y = world_map_wir$lat)
```



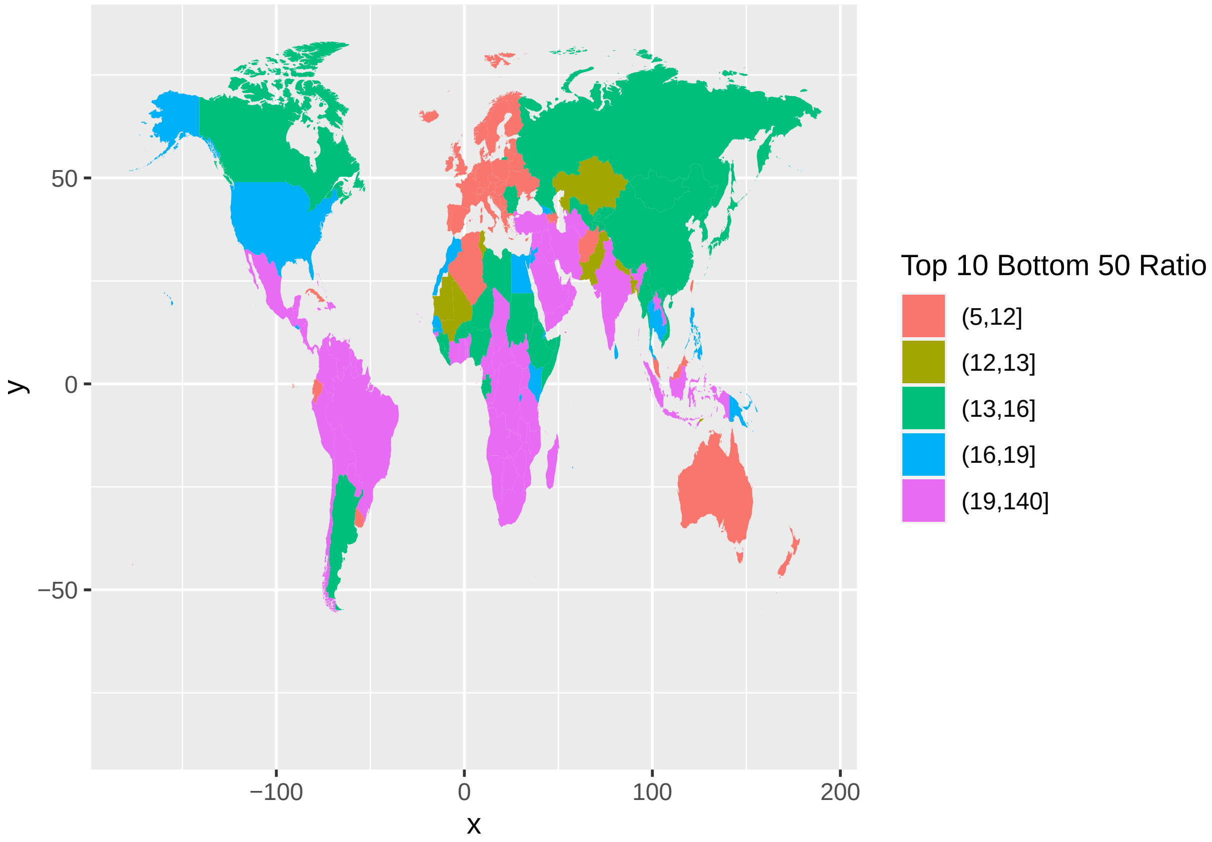
```
df_f3 %>% mutate(`Top 10 Bottom 50 Ratio` =
    cut(T10B50,breaks = c(5, 12, 13, 16, 19,140), include.lowest = FALSE)) %>%
  ggplot(aes(map_id = Country)) + geom_map(aes(fill = `Top 10 Bottom 50 Ratio`),
    map = world_map_wir) + expand_limits(x = world_map_wir$long, y = world_map_wir$lat) +
  coord_map("orthographic", orientation = c(25, 60, 0))
```



```
df_f3 %>% mutate(`Top 10 Bottom 50 Ratio` =
  cut(T10B50,breaks = c(5, 12, 13, 16, 19,140), include.lowest = FALSE)) %>%
  ggplot(aes(map_id = Country)) + geom_map(aes(fill = `Top 10 Bottom 50 Ratio`),
    map = world_map_wir) + expand_limits(x = world_map_wir$long, y = world_map_wir$lat) +
  coord_map("orthographic", orientation = c(15, -80, 0))
```



```
df_f3 %>% mutate(`Top 10 Bottom 50 Ratio` =
  cut(T10B50,breaks = c(5, 12, 13, 16, 19,140), include.lowest = FALSE)) %>%
  ggplot(aes(map_id = Country)) + geom_map(aes(fill = `Top 10 Bottom 50 Ratio`),
    map = world_map_wir) +
  expand_limits(x = world_map_wir$long, y = world_map_wir$lat)
```



```
df_f3 %>%
  mutate(`Top 10 Bottom 50 Ratio` =
        cut(T10B50,breaks = c(5, 12, 13, 16, 19,140), include.lowest = FALSE)) %>%
  ggplot(aes(map_id = Country)) +
  geom_map(aes(fill = `Top 10 Bottom 50 Ratio`), map = world_map_wir) +
  expand_limits(x = world_map_wir$long, y = world_map_wir$lat)  +
  labs(title = "Figure 3. Top 10/Bottom 50 income gaps across the world, 2021",
       x = "", y = "", fill = "Top 10/Bottom 50 ratio") +
  theme(legend.position="bottom",
        axis.text.x=element_blank(), axis.ticks.x=element_blank(),
        axis.text.y=element_blank(), axis.ticks.y=element_blank()) +
  scale_fill_brewer(palette='YlOrRd')
```

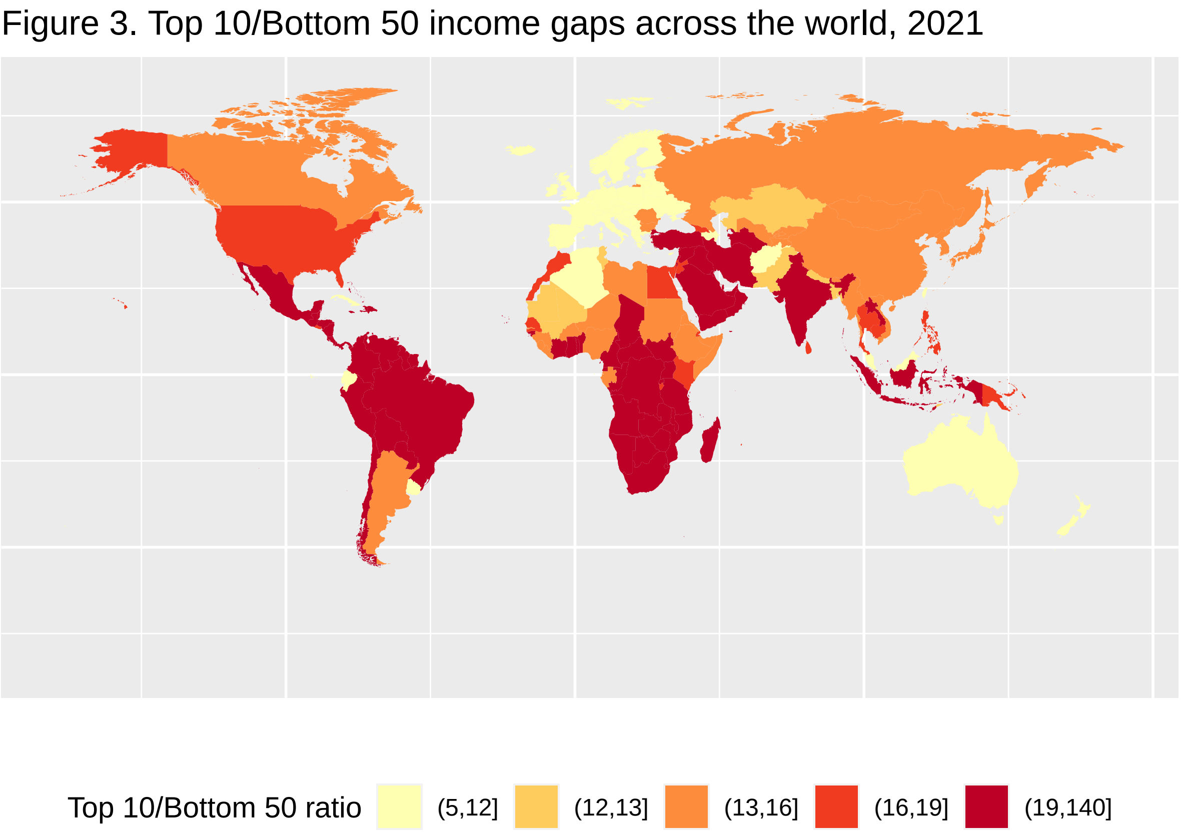


Figure 3. Top 10/Bottom 50 income gaps across the world, 2021
Top 10/Bottom 50 ratio
(5,12]
(12,13]
(13,16]
(16,19]
(19,140]

```r
df_f3 %>% anti_join(world_map_wir, by = c("Country" = "region"))
```

```
## # A tibble: 3 x 3
##    year Country   T10B50
##   <dbl> <chr>      <dbl>
## 1  2021 Hong Kong   17.7
## 2  2021 Macao       14.5
## 3  2021 Zanzibar    19.8
```

Filtering joins

- `anti_join(x,y, ...)`: return all rows from x without a match in y.
- `semi_join(x,y, ...)`: return all rows from x with a match in y.

Check `dplyr` cheat sheet, and Posit Primers Tidy Data.

4.1.14 Remaining Charts

- F5: Global income inequality: T10/B50 ratio, 1820-2020 - fit curve
- F9: Average annual wealth growth rate, 1995-2021 - fit curve + alpha
- F7: Global income inequality, 1820-2020 - pivot + fit curve
- F10: The share of wealth owned by the global 0.1% and billionaires, 2021 - pivot + fit curve
- F6: Global income inequality: Between vs. Within country inequality (Theil index), 1820-2020 - pivot + area

- F11: Top 1% vs bottom 50% wealth shares in Western Europe and the US, 1910-2020 - pivot name_sep + fit curve
- F8: The rise of private versus the decline of public wealth in rich countries, 1970-2020 - rename + pivot + pivot + fit curve
- F15: Per capita emissions acriss the world, 2019 - add row names + dodge

---

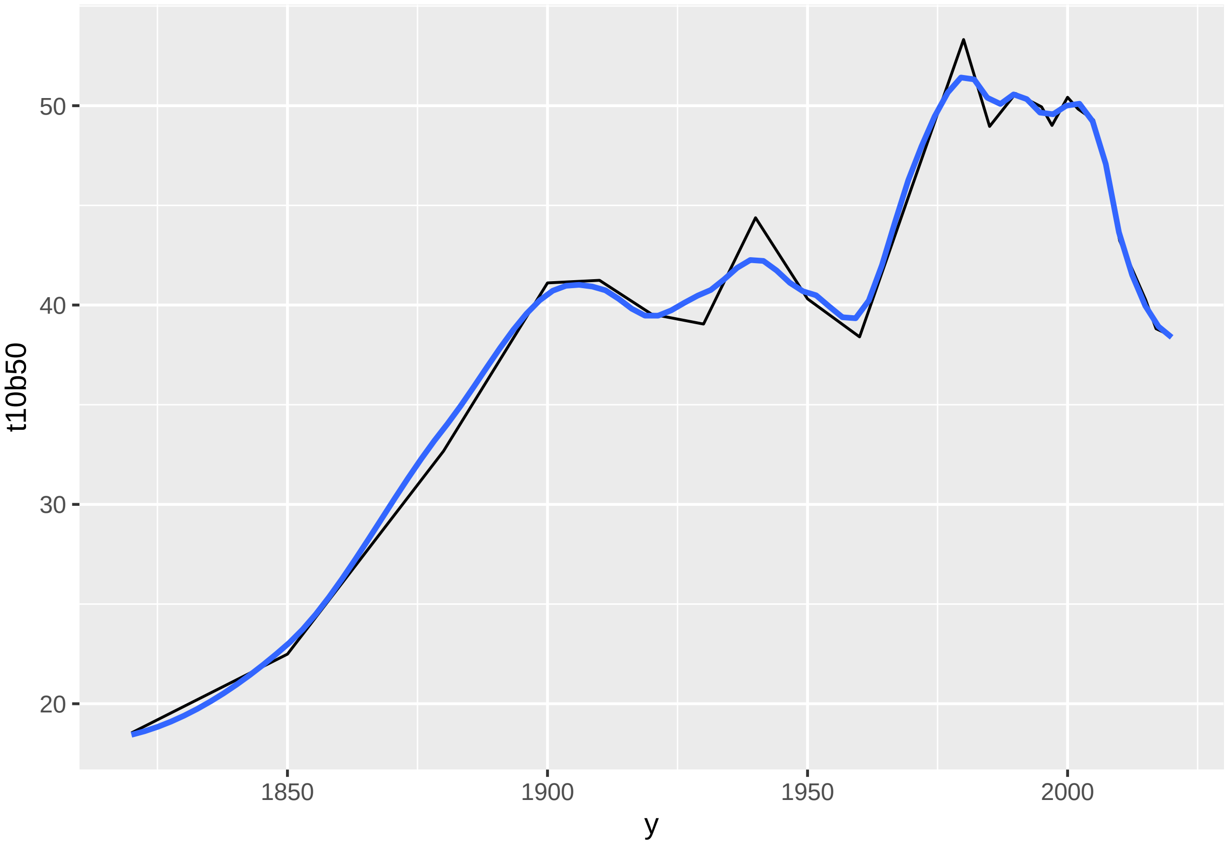
4.1.15 F5: Global income inequality: T10/B50 ratio, 1820-2020

```
(df_f5 <- read_excel("data/WIR2022s.xlsx", sheet = "data-F5"))
```

```
## # A tibble: 24 x 2
##        y t10b50
##    <dbl>  <dbl>
##  1  1820   18.5
##  2  1850   22.5
##  3  1880   32.7
##  4  1900   41.1
##  5  1910   41.2
##  6  1920   39.5
##  7  1930   39.0
##  8  1940   44.4
##  9  1950   40.3
## 10  1960   38.4
## # ... with 14 more rows
```

```
df_f5 %>% ggplot(aes(x = y, y = t10b50)) + geom_line() + geom_smooth(span=0.25, se=FALSE)
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



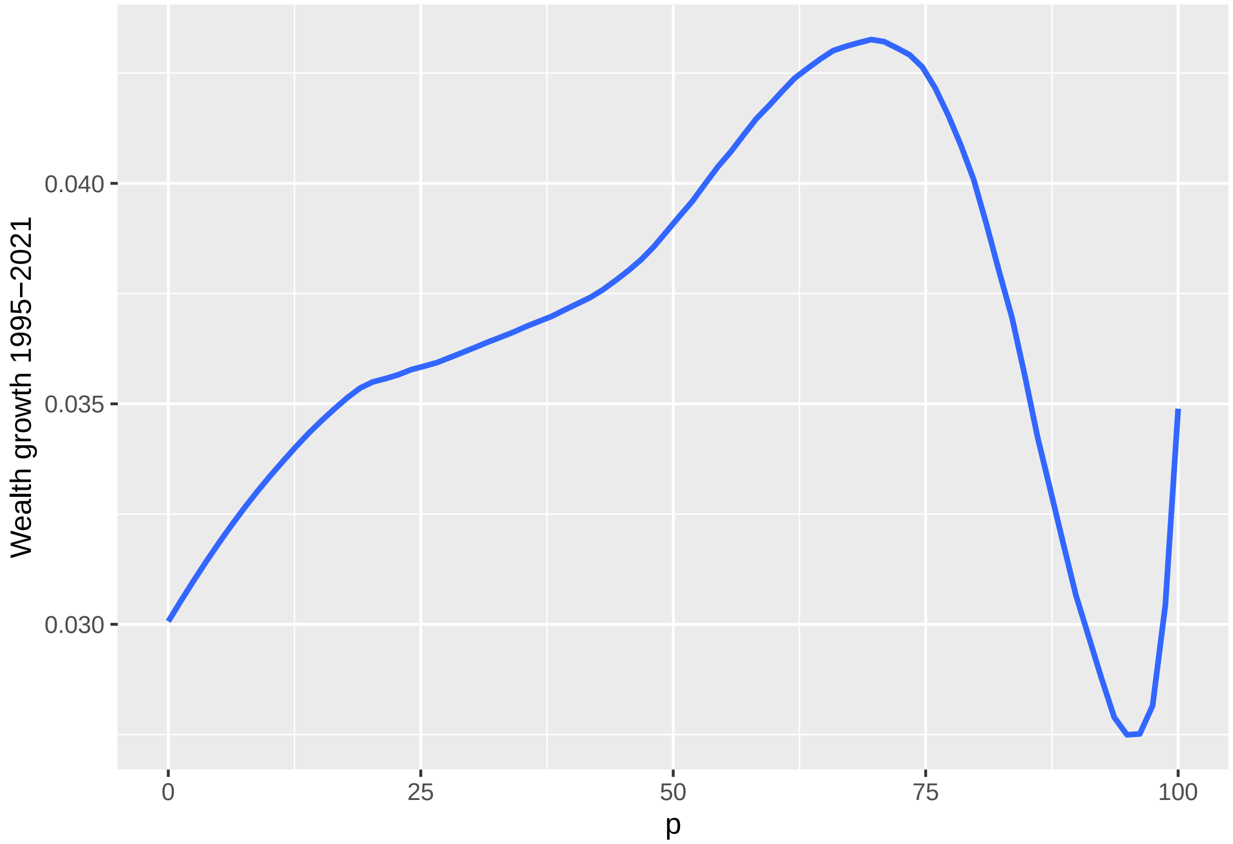
4.1.16 F9: Average annual wealth growth rate, 1995-2021 - fit curve + alpha

```
df_f9 <- read_excel("data/WIR2022s.xlsx", sheet = "data-F9"); df_f9
```

```
## # A tibble: 127 x 2
##         p `Wealth growth 1995-2021`
##     <dbl>                     <dbl>
##  1      0                    0.0310
##  2      1                    0.0310
##  3      2                    0.0310
##  4      3                    0.0310
##  5      4                    0.0310
##  6      5                    0.0310
##  7      6                    0.0312
##  8      7                    0.0317
##  9      8                    0.0322
## 10      9                    0.0328
## # ... with 117 more rows
```

```
df_f9 %>%
  ggplot(aes(x = p, y = `Wealth growth 1995-2021`)) + geom_smooth(span = 0.30, se = FALSE)
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

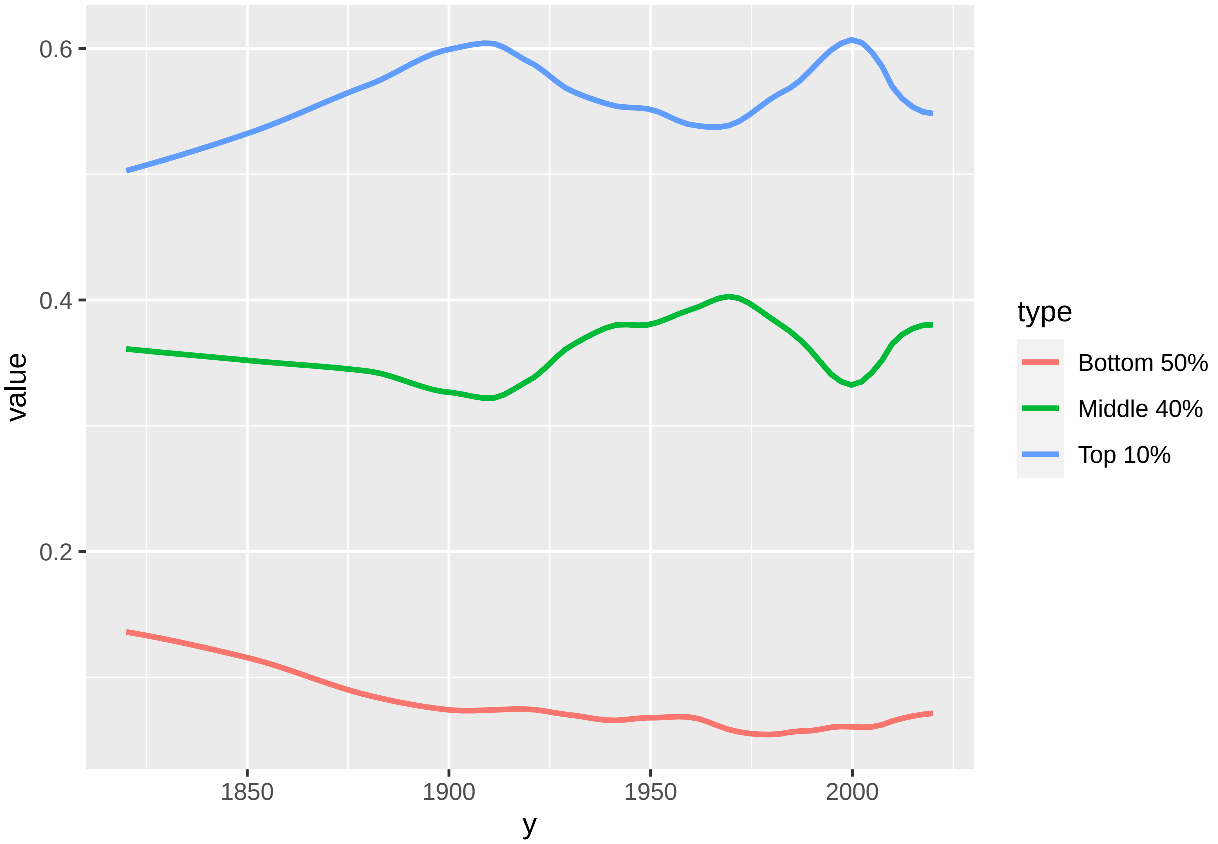


4.1.17 F7: Global income inequality, 1820-2020 - pivot + fit curve

```
df_f7 <- read_excel("data/WIR2022s.xlsx", sheet = "data-F7"); df_f7
```

```
## # A tibble: 24 x 4
##        y `Bottom 50%` `Middle 40%` `Top 10%`
##    <dbl>        <dbl>        <dbl>     <dbl>
##  1  1820       0.136         0.361     0.503
##  2  1850       0.118         0.350     0.532
##  3  1880       0.0870        0.345     0.568
##  4  1900       0.0724        0.332     0.595
##  5  1910       0.0729        0.326     0.601
##  6  1920       0.0755        0.328     0.597
##  7  1930       0.0714        0.371     0.558
##  8  1940       0.0629        0.379     0.558
##  9  1950       0.0687        0.377     0.554
## 10  1960       0.0701        0.392     0.538
## # ... with 14 more rows
```

```
df_f7 %>%
  pivot_longer(cols = 2:4, names_to = "type", values_to = "value") %>%
  ggplot(aes(x = y, y = value, color = type)) +
  stat_smooth(formula = y~x, method = "loess", span = 0.25, se = FALSE)
```



4.1.18 F10: The share of wealth owned by the global 0.1% and billionaires, 2021 - pivot + fit curve

```
df_f10 <- read_excel("data/WIR2022s.xlsx", sheet = "data-F10"); df_f10
```

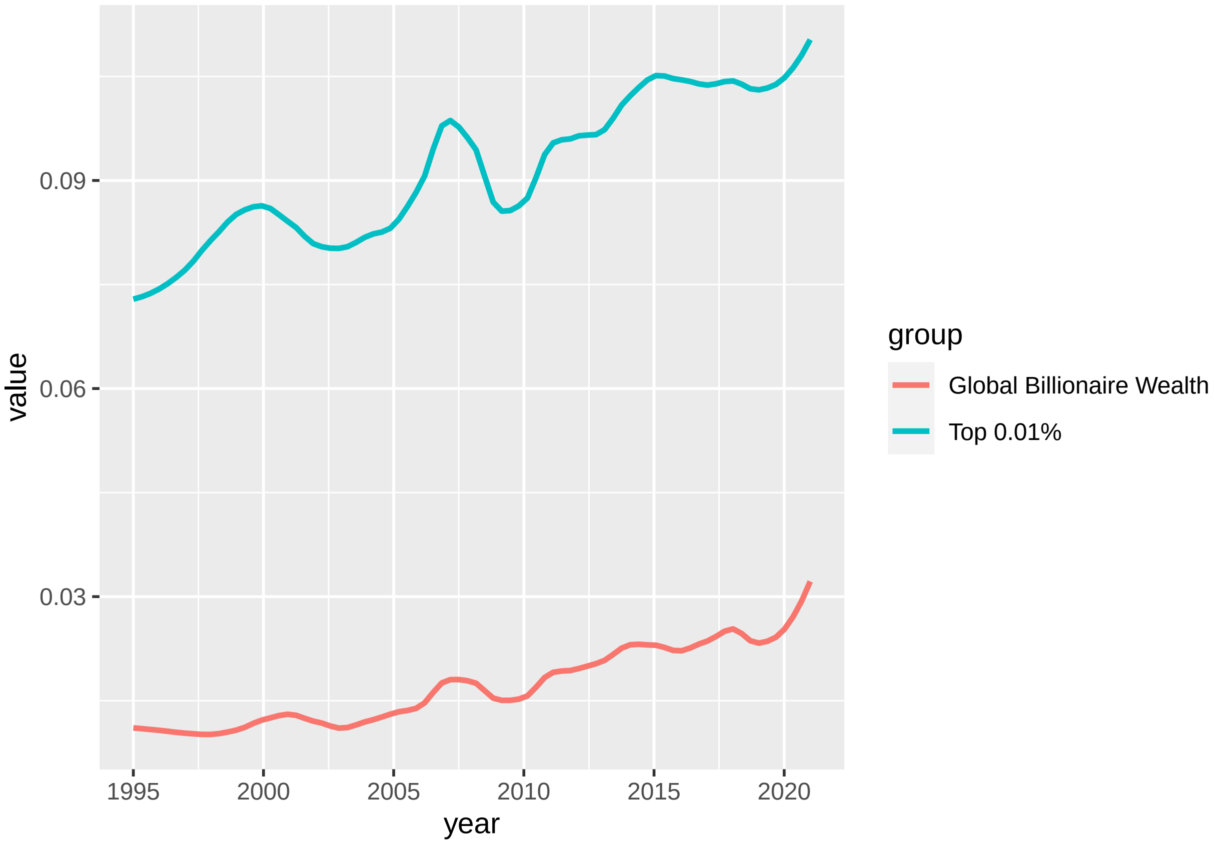
```
## New names:
## * `` -> `...4`
## * `` -> `...5`
## * `` -> `...6`
```

```
## # A tibble: 27 x 6
##     year bn_hhweal top0.1_hhweal ...4   ...5  ...6
##    <dbl>     <dbl>         <dbl> <lgl> <dbl> <dbl>
##  1  1995   0.0108         0.0729 NA       NA    NA
##  2  1996   0.0114         0.0744 NA       NA    NA
##  3  1997   0.0101         0.0768 NA       NA    NA
##  4  1998   0.00995        0.0815 NA       NA    NA
##  5  1999   0.0112         0.0855 NA       NA    NA
##  6  2000   0.0115         0.0862 NA       NA    NA
##  7  2001   0.0139         0.0846 NA       NA    NA
##  8  2002   0.0119         0.08   NA       NA    NA
##  9  2003   0.0103         0.08   NA       NA    NA
## 10  2004   0.0127         0.0828 NA       NA    NA
## # ... with 17 more rows
```

```r
df_f10 %>%
  select(year, "Global Billionaire Wealth" = bn_hhweal, "Top 0.01%" = top0.1_hhweal) %>%
  pivot_longer(!year, names_to = "group",".value", values_to = "value")
```

```
## # A tibble: 54 x 3
##     year group                     value
##    <dbl> <chr>                     <dbl>
##  1  1995 Global Billionaire Wealth 0.0108
##  2  1995 Top 0.01%                 0.0729
##  3  1996 Global Billionaire Wealth 0.0114
##  4  1996 Top 0.01%                 0.0744
##  5  1997 Global Billionaire Wealth 0.0101
##  6  1997 Top 0.01%                 0.0768
##  7  1998 Global Billionaire Wealth 0.00995
##  8  1998 Top 0.01%                 0.0815
##  9  1999 Global Billionaire Wealth 0.0112
## 10  1999 Top 0.01%                 0.0855
## # ... with 44 more rows
```

```r
df_f10 %>%
  select(year, "Global Billionaire Wealth" = bn_hhweal, "Top 0.01%" = top0.1_hhweal) %>%
  pivot_longer(!year, names_to = "group",".value", values_to = "value") %>%
  ggplot() +
  stat_smooth(aes(x = year, y = value, color = group), formula = y~x, method = "loess", span = 0.25, se
```



4.1.19 F6: Global income inequality: Between vs. Within country inequality (Theil index), 1820-2020 - pivot + area

```
df_f6 <- read_excel("data/WIR2022s.xlsx", sheet = "data-F6"); df_f6
```

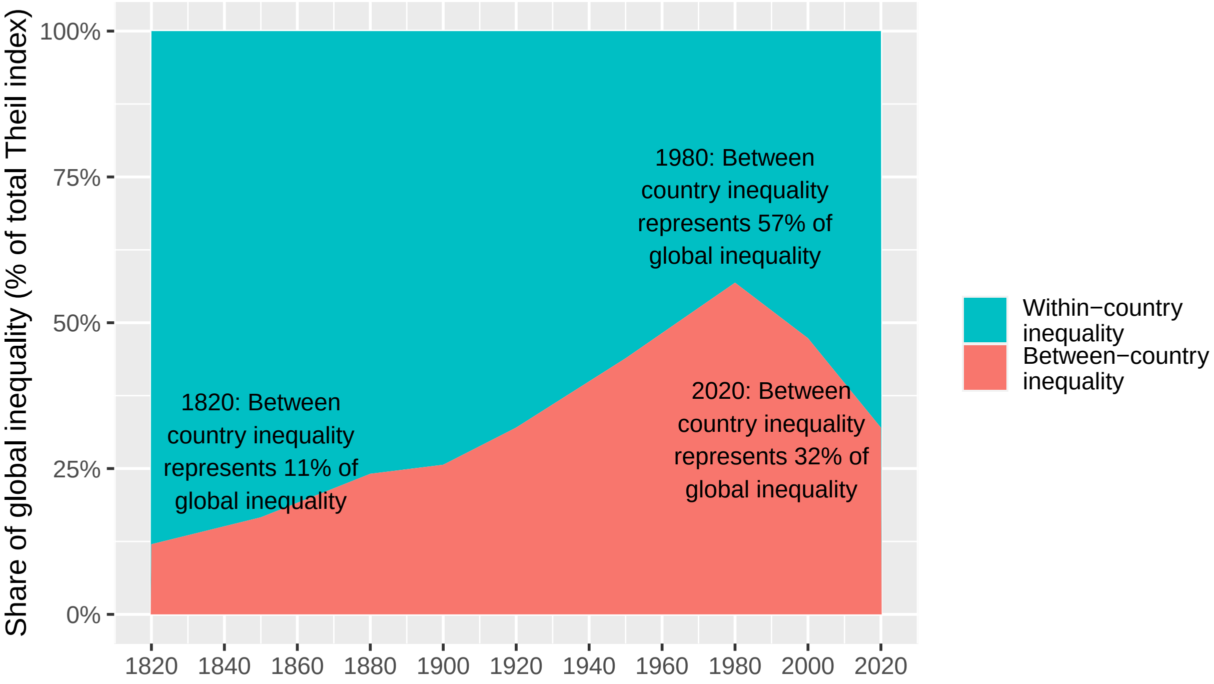
```
## New names:
## * `` -> `...1`
```

```
## # A tibble: 9 x 3
##    ...1 `Between-country inequality` `Within-country inequality`
##   <dbl>                        <dbl>                       <dbl>
## 1  1820                        0.120                       0.880
## 2  1850                        0.166                       0.834
## 3  1880                        0.241                       0.759
## 4  1900                        0.257                       0.743
## 5  1920                        0.320                       0.680
## 6  1950                        0.439                       0.561
## 7  1980                        0.569                       0.431
## 8  2000                        0.473                       0.527
## 9  2020                        0.320                       0.680
```

```
df_f6 %>% select(year = "...1", 2:3) %>%
  pivot_longer(cols = 2:3, names_to = "type", values_to = "value") %>%
  mutate(types = factor(type,
      levels = c("Within-country inequality", "Between-country inequality"))) %>%
  ggplot(aes(x = year, y = value, fill = types)) +
  geom_area() +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
  scale_x_continuous(breaks = round(seq(1820, 2020, by = 20),1)) +
  scale_fill_manual(values=rev(scales::hue_pal()(2)),
      labels = function(x) str_wrap(x, width = 15)) +
  labs(title = "Figure 6. Global income inequality:
       \nBetween vs. within country inequality (Theil index), 1820-2020",
       x = "", y = "Share of global inequality (% of total Theil index)", fill = "") +
  annotate("text", x = 1850, y = 0.28,
      label = stringr::str_wrap("1820: Between country inequality represents 11%
                                of global inequality", width = 20), size = 3) +
  annotate("text", x = 1980, y = 0.70,
      label = stringr::str_wrap("1980: Between country inequality represents 57%
                                of global inequality", width = 20), size = 3) +
  annotate("text", x = 1990, y = 0.30,
      label = stringr::str_wrap("2020: Between country inequality represents 32%
                                of global inequality", width = 20), size = 3)
```

Figure 6. Global income inequality:

Between vs. within country inequality (Theil index), 1820−2020



4.1.20 F11: Top 1% vs bottom 50% wealth shares in Western Europe and the US, 1910-2020 - pivot name_sep + fit curve

```
df_f11 <- read_excel("data/WIR2022s.xlsx", sheet = "data-F11"); df_f11
```

```
## # A tibble: 12 x 5
##     year USbot50 UStop1 EUbot50 EUtop1
##    <dbl>   <dbl>  <dbl>   <dbl>  <dbl>
##  1  1910 0.00700  0.425  0.0128  0.554
##  2  1920 0.0102   0.410  0.0139  0.496
##  3  1930 0.00737  0.409  0.0175  0.464
##  4  1940 0.0112   0.319  0.0282  0.412
##  5  1950 0.0270   0.276  0.0272  0.339
##  6  1960 0.0232   0.279  0.0503  0.304
##  7  1970 0.0221   0.241  0.0649  0.235
##  8  1980 0.0260   0.251  0.0658  0.200
##  9  1990 0.0211   0.294  0.0535  0.200
## 10  2000 0.0162   0.323  0.0543  0.214
## 11  2010 0.0111   0.357  0.0500  0.219
## 12  2020 0.0149   0.354  0.0576  0.219
```

```
df_f11 %>%
  rename(!year, US_bot50 = USbot50, US_top1 = UStop1,
         EU_bot50 = EUbot50, EU_top1 = EUtop1) %>%
  pivot_longer(!year, names_to = c("group",".value"), names_sep = "_") %>%
  pivot_longer(3:4, names_to = "type", values_to = "value") %>%
  ggplot() +
  stat_smooth(aes(x = year, y = value, color = group, linetype = type),
              span = 0.25, se = FALSE) +
  scale_x_continuous(breaks = round(seq(1910, 2020, by = 10),1)) +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
  labs(title = "Figure 11. Top 1% vs bottom 50% wealth shares
       \n in Western Europe and the US, 1910-2020",
       x = "", y = "Share of total personal wealth (%)", color = "", linetype = "") +
  scale_linetype_manual(values = c("dotted","solid")) +
  annotate("text", x = 2000, y = 0.50,
      label = stringr::str_wrap("Wealth inequality has been rising at
        different speeds after a historical decline. The bottom 50% has always been
                                extremely low.", width = 30), size = 3)
```

```
df_f11 %>% rename(!year, US_bot50 = USbot50, US_top1 = UStop1,
                  EU_bot50 = EUbot50, EU_top1 = EUtop1)
```

4.1.20.1 Step 1.

```
## # A tibble: 12 x 5
##     year US_bot50 US_top1 EU_bot50 EU_top1
##    <dbl>    <dbl>   <dbl>    <dbl>   <dbl>
##  1  1910  0.00700   0.425   0.0128   0.554
##  2  1920  0.0102    0.410   0.0139   0.496
##  3  1930  0.00737   0.409   0.0175   0.464
##  4  1940  0.0112    0.319   0.0282   0.412
##  5  1950  0.0270    0.276   0.0272   0.339
##  6  1960  0.0232    0.279   0.0503   0.304
##  7  1970  0.0221    0.241   0.0649   0.235
##  8  1980  0.0260    0.251   0.0658   0.200
##  9  1990  0.0211    0.294   0.0535   0.200
## 10  2000  0.0162    0.323   0.0543   0.214
## 11  2010  0.0111    0.357   0.0500   0.219
## 12  2020  0.0149    0.354   0.0576   0.219
```

```
df_f11 %>%
  rename(!year, US_bot50 = USbot50, US_top1 = UStop1,
         EU_bot50 = EUbot50, EU_top1 = EUtop1) %>%
  pivot_longer(!year, names_to = c("group",".value"), names_sep = "_")
```

4.1.20.2 Step 2.

4.1.20.3 Step 2.

```
## # A tibble: 24 x 4
##     year group   bot50  top1
##    <dbl> <chr>   <dbl> <dbl>
## 1 1910 US 0.00700 0.425
## 2 1910 EU 0.0128 0.554
## 3 1920 US 0.0102 0.410
## 4 1920 EU 0.0139 0.496
## 5 1930 US 0.00737 0.409
## 6 1930 EU 0.0175 0.464
## 7 1940 US 0.0112 0.319
## 8 1940 EU 0.0282 0.412
## 9 1950 US 0.0270 0.276
## 10 1950 EU 0.0272 0.339
## # ... with 14 more rows
```

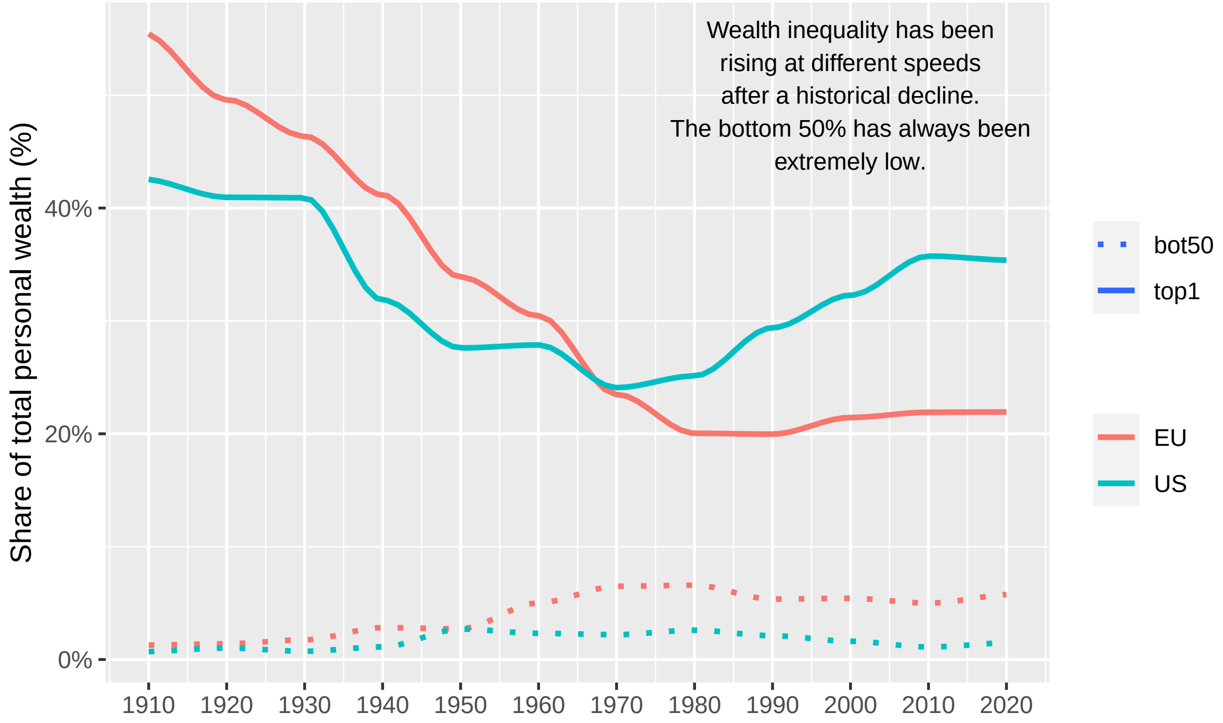
```
df_f11 %>%
  rename(!year, US_bot50 = USbot50, US_top1 = UStop1,
         EU_bot50 = EUbot50, EU_top1 = EUtop1) %>%
  pivot_longer(!year, names_to = c("group",".value"),
               names_sep = "_") %>%
  pivot_longer(3:4, names_to = "type", values_to = "value")
```

4.1.20.4 Step 3.

4.1.20.5 Step 3.

```
## # A tibble: 48 x 4
##     year group type    value
##    <dbl> <chr> <chr>   <dbl>
##  1  1910 US    bot50 0.00700
##  2  1910 US    top1  0.425
##  3  1910 EU    bot50 0.0128
##  4  1910 EU    top1  0.554
##  5  1920 US    bot50 0.0102
##  6  1920 US    top1  0.410
##  7  1920 EU    bot50 0.0139
##  8  1920 EU    top1  0.496
##  9  1930 US    bot50 0.00737
## 10  1930 US    top1  0.409
## # ... with 38 more rows
```

Figure 11. Top 1% vs bottom 50% wealth shares
in Western Europe and the US, 1910−2020



Wealth inequality has been rising at different speeds after a historical decline. The bottom 50% has always been extremely low.

4.1.21 F8: The rise of private versus the decline of public wealth in rich countries, 1970-2020 - rename + pivot + pivot + fit curve

```
df_f8 <- read_excel("data/WIR2022s.xlsx", sheet = "data-F8"); df_f8
```

```
## # A tibble: 51 x 17
##     year Germany Germany (pri~1 Spain Spain~2 France Franc~3    UK UK (p~4 Japan
##    <dbl>   <dbl>          <dbl> <dbl>   <dbl>  <dbl>   <dbl> <dbl>   <dbl> <dbl>
##  1  1970   1.11            2.30 0.604    4.06  0.422    3.12 0.601    2.85 0.719
##  2  1971   1.12            2.25 0.657    4.53  0.443    3.06 0.689    2.86 0.782
##  3  1972   1.11            2.27 0.624    4.36  0.467    3.08 0.790    2.94 0.842
##  4  1973   1.11            2.23 0.596    4.46  0.478    3.06 0.929    2.92 0.895
##  5  1974   1.13            2.25 0.586    4.64  0.498    3.03 1.09     2.94 0.936
##  6  1975   1.12            2.35 0.602    4.83  0.545    3.12 1.00     2.65 0.944
##  7  1976   1.03            2.34 0.581    4.46  0.561    3.08 0.918    2.54 0.902
##  8  1977   1.01            2.42 0.586    4.10  0.567    3.10 0.867    2.47 0.880
##  9  1978   0.990           2.52 0.604    4.10  0.580    3.20 0.881    2.51 0.860
## 10  1979   0.989           2.55 0.621    4.20  0.624    3.30 0.955    2.62 0.884
## # ... with 41 more rows, 7 more variables: `Japan (private)` <dbl>,
## #   Norway <dbl>, `Norway (private)` <dbl>, USA <dbl>, `USA (private)` <dbl>,
## #   gwealAVGRICH <dbl>, pwealAVGRICH <dbl>, and abbreviated variable names
## #   1: `Germany (private)`, 2: `Spain (private)`, 3: `France (private)`,
## #   4: `UK (private)`
```

```
df_f8 %>%
  select(year, Germany_public = Germany, Germany_private = 'Germany (private)',
         Spain_public = Spain, Spain_private = 'Spain (private)',
         France_public = France, France_private = 'France (private)',
         UK_public  = UK, UK_private = 'UK (private)',
         Japan_public = Japan, Japan_private = 'Japan (private)',
         Norway_public = Norway, Norway_private = 'Norway (private)',
         USA_public = USA, USA_private = 'USA (private)') %>%
  pivot_longer(!year, names_to = c("country",".value"), names_sep = "_") %>%
  pivot_longer(3:4, names_to = "type", values_to = "value") %>%
  ggplot() +
  stat_smooth(aes(x = year, y = value, color = country, linetype = type),
              span = 0.25, se = FALSE, size=0.75) +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
  labs(title = "Figure 8. The rise of private versus the decline of public
      wealth in rich countries, 1970-2020",
      x = "", y = "wealth as as % of national income", color = "", type = "")
```

```
df_f8 %>%
  select(year, Germany_public = Germany, Germany_private = 'Germany (private)',
         Spain_public = Spain, Spain_private = 'Spain (private)',
         France_public = France, France_private = 'France (private)',
         UK_public  = UK, UK_private = 'UK (private)',
         Japan_public = Japan, Japan_private = 'Japan (private)',
         Norway_public = Norway, Norway_private = 'Norway (private)',
         USA_public = USA, USA_private = 'USA (private)')
```

4.1.21.1 Step 1

```
## # A tibble: 51 x 15
##     year Germa~1 Germa~2 Spain~3 Spain~4 Franc~5 Franc~6 UK_pu~7 UK_pr~8 Japan~9
##    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1  1970   1.11     2.30   0.604    4.06   0.422    3.12   0.601    2.85   0.719
##  2  1971   1.12     2.25   0.657    4.53   0.443    3.06   0.689    2.86   0.782
##  3  1972   1.11     2.27   0.624    4.36   0.467    3.08   0.790    2.94   0.842
##  4  1973   1.11     2.23   0.596    4.46   0.478    3.06   0.929    2.92   0.895
##  5  1974   1.13     2.25   0.586    4.64   0.498    3.03   1.09     2.94   0.936
##  6  1975   1.12     2.35   0.602    4.83   0.545    3.12   1.00     2.65   0.944
##  7  1976   1.03     2.34   0.581    4.46   0.561    3.08   0.918    2.54   0.902
##  8  1977   1.01     2.42   0.586    4.10   0.567    3.10   0.867    2.47   0.880
##  9  1978   0.990    2.52   0.604    4.10   0.580    3.20   0.881    2.51   0.860
## 10  1979   0.989    2.55   0.621    4.20   0.624    3.30   0.955    2.62   0.884
## # ... with 41 more rows, 5 more variables: Japan_private <dbl>,
## #   Norway_public <dbl>, Norway_private <dbl>, USA_public <dbl>,
## #   USA_private <dbl>, and abbreviated variable names 1: Germany_public,
## #   2: Germany_private, 3: Spain_public, 4: Spain_private, 5: France_public,
## #   6: France_private, 7: UK_public, 8: UK_private, 9: Japan_public
```

```
df_f8 %>%
  select(year, Germany_public = Germany, Germany_private = 'Germany (private)',
         Spain_public = Spain, Spain_private = 'Spain (private)',
         France_public = France, France_private = 'France (private)',
         UK_public  = UK, UK_private = 'UK (private)',
         Japan_public = Japan, Japan_private = 'Japan (private)',
         Norway_public = Norway, Norway_private = 'Norway (private)',
         USA_public = USA, USA_private = 'USA (private)') %>%
  pivot_longer(!year, names_to = c("country",".value"), names_sep = "_")
```

4.1.21.2 Step 2.

```
## # A tibble: 357 x 4
##     year country public private
##    <dbl> <chr>    <dbl>   <dbl>
##  1  1970 Germany  1.11     2.30
##  2  1970 Spain    0.604    4.06
##  3  1970 France   0.422    3.12
##  4  1970 UK       0.601    2.85
##  5  1970 Japan    0.719    3.09
##  6  1970 Norway  NA       NA
##  7  1970 USA      0.364    3.26
##  8  1971 Germany  1.12     2.25
##  9  1971 Spain    0.657    4.53
## 10  1971 France   0.443    3.06
## # ... with 347 more rows
```

```
df_f8 %>%
  select(year, Germany_public = Germany, Germany_private = 'Germany (private)',
         Spain_public = Spain, Spain_private = 'Spain (private)',
         France_public = France, France_private = 'France (private)',
         UK_public  = UK, UK_private = 'UK (private)',
         Japan_public = Japan, Japan_private = 'Japan (private)',
         Norway_public = Norway, Norway_private = 'Norway (private)',
         USA_public = USA, USA_private = 'USA (private)') %>%
  pivot_longer(!year, names_to = c("country",".value"), names_sep = "_") %>%
  pivot_longer(3:4, names_to = "type", values_to = "value")
```

4.1.21.3 Step 3.

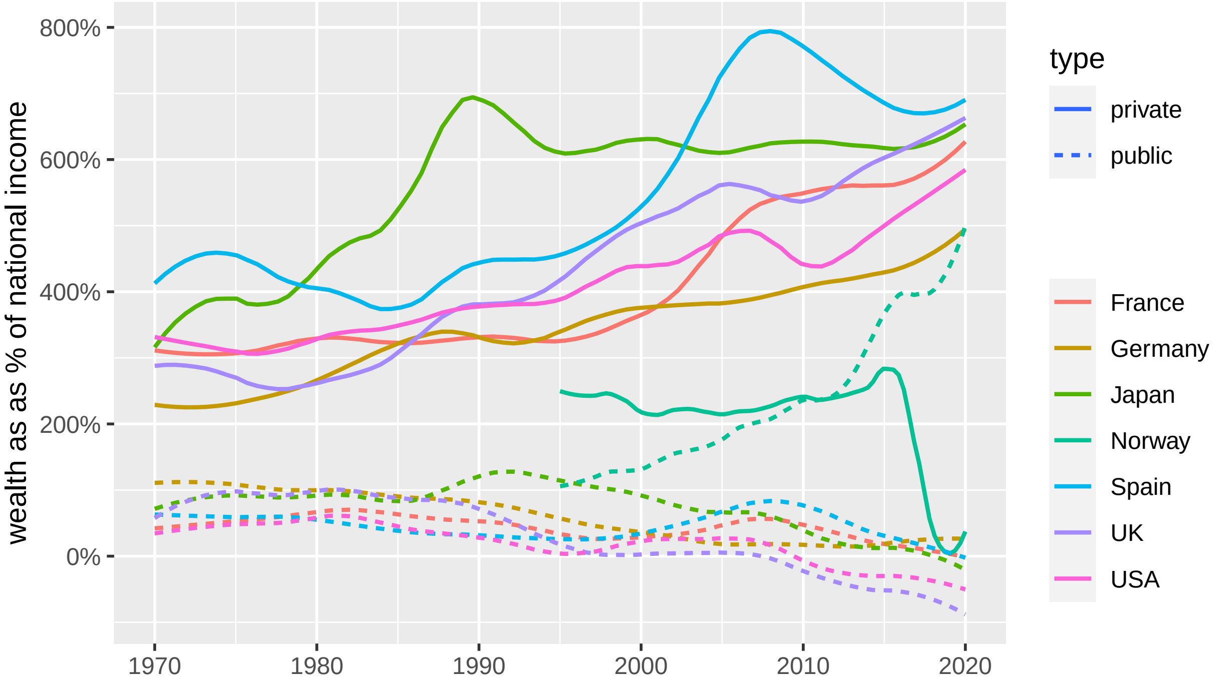
```
## # A tibble: 714 x 4
##     year country type    value
##    <dbl> <chr>   <chr>   <dbl>
##  1  1970 Germany public  1.11
##  2  1970 Germany private 2.30
##  3  1970 Spain   public  0.604
##  4  1970 Spain   private 4.06
##  5  1970 France  public  0.422
##  6  1970 France  private 3.12
##  7  1970 UK      public  0.601
##  8  1970 UK      private 2.85
##  9  1970 Japan   public  0.719
## 10  1970 Japan   private 3.09
## # ... with 704 more rows
```

```
df_f8 %>%
  select(year, Germany_public = Germany, Germany_private = 'Germany (private)',
         Spain_public = Spain, Spain_private = 'Spain (private)',
         France_public = France, France_private = 'France (private)',
         UK_public  = UK, UK_private = 'UK (private)',
         Japan_public = Japan, Japan_private = 'Japan (private)',
         Norway_public = Norway, Norway_private = 'Norway (private)',
         USA_public = USA, USA_private = 'USA (private)') %>%
  pivot_longer(!year, names_to = c("country",".value"), names_sep = "_") %>%
  pivot_longer(3:4, names_to = "type", values_to = "value") %>%
  ggplot() +
  stat_smooth(aes(x = year, y = value, color = country, linetype = type),
              formula = y~x, method = "loess", span = 0.25, se = FALSE, size=0.75) +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
  labs(title = "Figure 8. The rise of private versus the decline of public wealth
       \nin rich countries, 1970-2020",
       x = "", y = "wealth as as % of national income", color = "", type = "")
```

4.1.21.4 Step 3. Final Step

Figure 8. The rise of private versus the decline of public wealth

in rich countries, 1970−2020



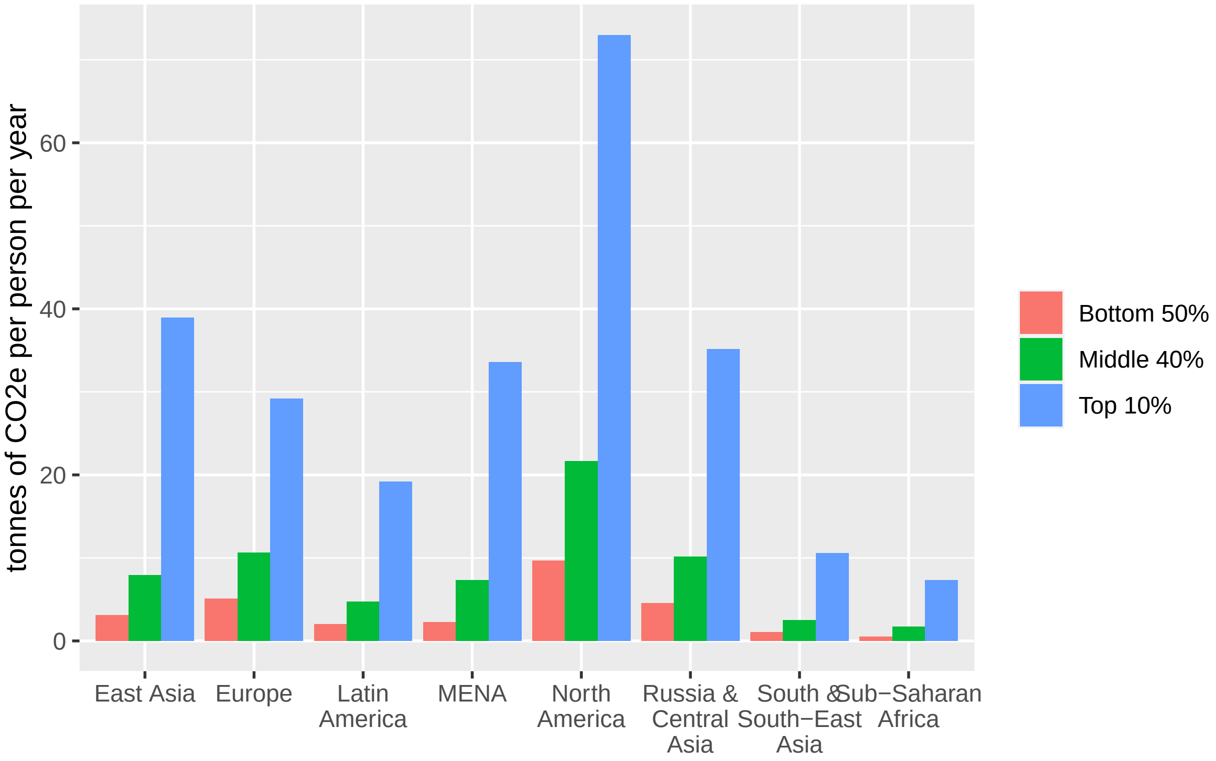
4.1.22 F15: Per capita emissions acriss the world, 2019 - add row names + dodge

```
df_f15 <- read_excel("data/WIR2022s.xlsx", sheet = "data-F15"); df_f15
```

```
## # A tibble: 24 x 4
##    regionWID                group       tcap  mark
##    <chr>                    <chr>      <dbl> <dbl>
##  1 East Asia                Bottom 50%  3.12     1
##  2 <NA>                     Middle 40%  7.91     1
##  3 <NA>                     Top 10%    38.9      1
##  4 Europe                   Bottom 50%  5.09     2
##  5 <NA>                     Middle 40% 10.6      2
##  6 <NA>                     Top 10%    29.2      2
##  7 North America            Bottom 50%  9.67     3
##  8 <NA>                     Middle 40% 21.7      3
##  9 <NA>                     Top 10%    73.0      3
## 10 South & South-East Asia Bottom 50%  1.04     4
## # ... with 14 more rows
```

```
df_f15 %>% mutate(region = rep(regionWID[!is.na(regionWID)], each = 3)) %>%
  select(region, group, tcap) %>%
  ggplot(aes(x = region, y = tcap, fill = group)) +
  geom_col(position = "dodge") +
  scale_x_discrete(labels = function(x) stringr::str_wrap(x, width = 10)) +
  labs(title = "Figure 15 Per capita emissions across the world, 2019",
       x = "", y = "tonnes of CO2e per person per year", fill = "")
```

Figure 15 Per capita emissions across the world, 2019



4.2 EDA Workflow

4.2.1 EDA Step 0

1. Choose and clarify a topic to study.
2. List questions to study
3. Find data:

- link to data with a url: universal resource locator in a webpage
- download data in csv, Excel, etc.

Repeat the process during your EDA.

4.2.2 EDA by R Studio: Step 1

In RStudio,

1.1. Project

- Create a new project: File > New Project; or

- Open a project: File > Open Project, Open Project in New Session, Open Recent Project

- *Check there is a file `project_name.Rproj` in your project folder (directory)*

1.2. data folder (directory) `data`

- Create a data folder: Press New Folder at the right bottom pane; or

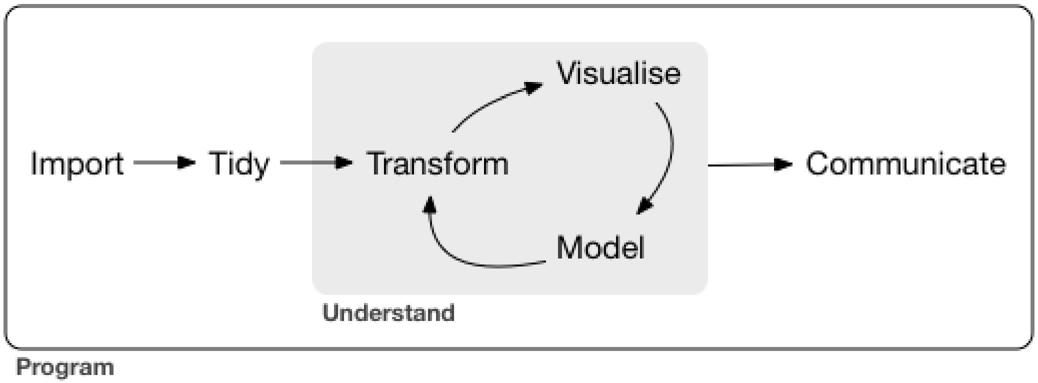


Figure 1: image

- Confirm the data folder previously created: Press Files at the right bottom pane
- *If you follow 1, the data folder exists in your project folder*

1.3. Move (or copy) data for the project to the data folder

- If you downloaded the data, it is in your Download folder. Move it to `data`.
- *Check in your RStudio that your data is in `data`: Press Files at the right bottom pane and click `data`, the data folder.*

4.2.3 EDA by R Studio: Step 2

2.1. Project Notebook: Memo

- Create an R Notebook: File > New File > R Notebook
  - You can use R Notebook template in Moodle by moving the template (template.Rmd or template.nb.Rmd) file in your project folder or copy and paste the text file into your new R Notebook.
  - If you use template.nb.Rmd (R Notebook File), choose Open in Editor.
- Add descriptive title.

2.2. Setup Code Chunk

- Create a code chunk and add packages to use in the project and RUN the code.
  - library(tidyverse)
  - library(WDI)
  - or any other packages

2.3. Choose `Source` or `Visual` editor mode, and start editing Project Notebook

- Set up Headings such as: About, Data, Analysis and Visualizations, Conclusions
- Under About or Data, paste url of the sites and/or the data
  - eg. World Development Indicator: https://datatopics.worldbank.org/world-development-indicators/)
  - eg. Public expenditure on education: https://data.un.org/_Docs/SYB/CSV/SYB65_245_202209_Public% 20expenditure%20on%20education.csv)

2.4. Edit a new file by saving as for a report

- File > Save As…

4.2.4 EDA by R Studio: Step 3 - Importing Data

Assign a name you can recall easily when you import data. You may need to reload the data with options.

3.1. Use a package:

- WDI, wir, eurostat, etc/
- 'wdi_shortname <- WDI(indicator = "indicator's name", ...)
- Store the data and use it: `write_csv(wdi_shortname, "data/wdi_shortname.csv")`
- `wdi_shortname <- read_csv("data/wdi_shortname.csv")`

3.2. Use `readr` to read from `data`, your data folder

- `df1_shortname <- read_csv("data/file_name.csv")`

3.3. Use `readr` to read using the url of the data

- `df2_shortname <- read_csv("url_of_the_data")`
- Store the data and use it: `write_csv(df2_shortname, "data/df2_shortname.csv")`
- `df2_shortname <- read_csv("data/df2_shortname.csv")`

3.5. Use `readxl` to read Excel data. Add `library(readxl)` in the setup and run.

- `df4 <- read_excel("data/file_name.xlsx", sheet = 1)`

References: Cheat Sheet - `readr`, readr, readxl

4.2.5 EDA by R Studio: Step 4 - Data Trasnformation

4.1. Look at the data: suppose `df` is the data frame

- It is a good option to change into a tibble: `dt <- as_tibble(df)`
- `head(df)`, `str(df)`, `summary(df)`, `dt`, `glimpse(dt)`

4.2. Look at each variable

- categorical? numerical?
- factor? - forcats

4.3. Variation of each data: suppose `x1` is a column name.

- `df %>% ggplot() + geom_histogram(aes(x1), bins = 30)`
- `df %>% drop_na(x1)`: see the rows with a value in `x1`. If the value is NA, the row is not shown.
  - `df_wo_na <- df %>% drop_na(x1)` if you want to use only the rows without NA in `x1`

4.4. Use `dpylr` and `tidyr` to change column names, tidy data, and/or summarize data

- `rename`, `select`, `filter`, `arrange`, `mutate`, `pivot_longer()`, `pivot_wider()`, `group_by` and `summarize`

References: Cheat Sheet - `dplyr` and `tidyr`, dplyr, tidyr

4.2.6 EDA by R Studio: Step 5 - Visualize Data

5.1. In combination with Stap 4 - data transformation, try various data visualization.

- What type of variation occurs within my variables?
- What type of covariation occurs between my variables?

5.2. Keep a record of what you can observe by the visualization

5.3. Edit the list of questions by adding or polishing

5.4. Select several informative chart and add options

5.5. Look at examples from the textbooks or teaching site to have better visualization

References: Cheat Sheet - `ggplot2` ggplot2, ggplot2 book

4.2.7 EDA by R Studio: Step 6 - Conclusions and Questions for Further Study

1. EDA is an iterative cycle that helps you understand what your data says. When you do EDA, you:
2. Generate questions about your data
3. Search for answers by visualising, transforming, and/or modeling your data

Use what you learn to refine your questions and/or generate new questions

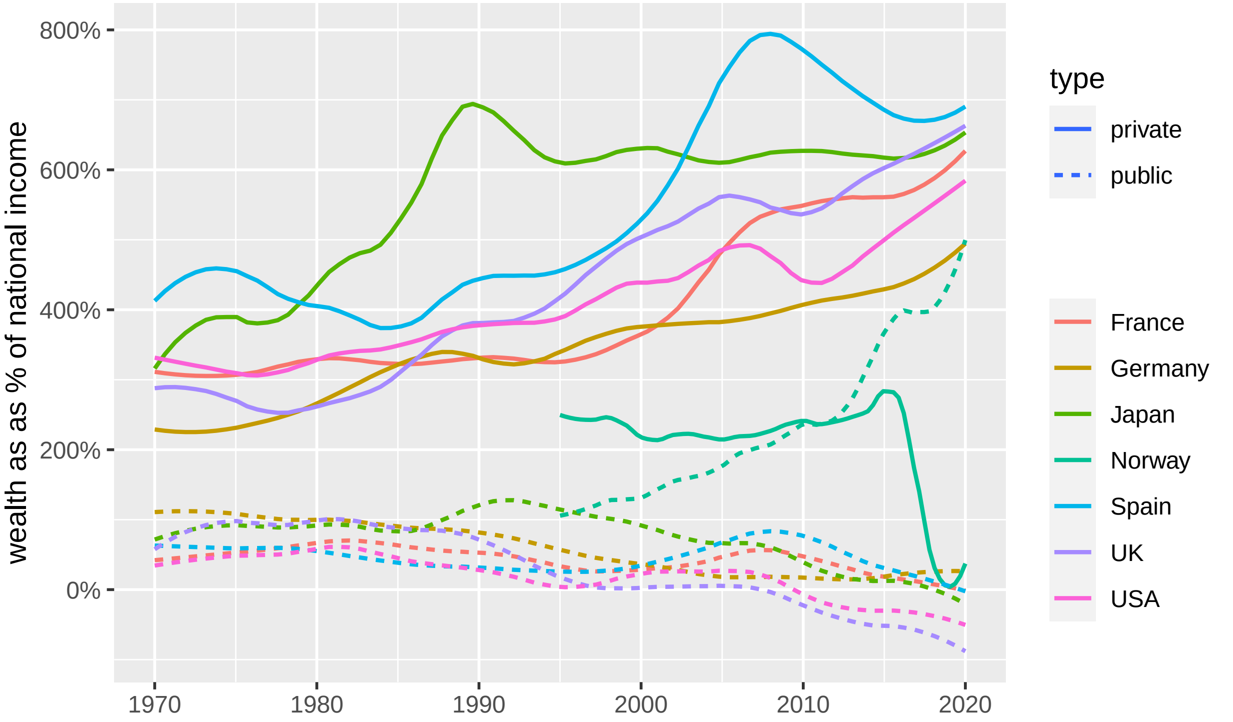
EDA is an important part of any data analysis. You can use EDA to make discoveries about the world; or you can use EDA to ensure the quality of your data, asking questions about whether the data meets your standards or not.

4.2.8 Example: WDI

- Government expenditure on education, total (% of GDP)
  - https://data.worldbank.org/indicator/SE.XPD.TOTL.GD.ZS
- ID: SE.XPD.TOTL.GD.ZS

4.2.9 Example: WIR2022

Figure 8. The rise of private versus the decline of public wealth in rich countries, 1970−2020



4.3 The Week Five Assignment (in Moodle)

`tidyr` and WIR2022

- Create an R Notebook of a Data Analysis containing the following and submit the rendered HTML file (eg. `a3_123456.nb.html` by replacing 123456 with your ID)
  1. create an R Notebook using the R Notebook Template in Moodle, save as `a3_123456.Rmd`,
  2. write your name and ID and the contents,
  3. run each code block,
  4. preview to create `a3_123456.nb.html`,
  5. submit `a3_123456.nb.html` to Moodle.

1. Choose a data with at least two categorical variables and at least two numerical variables.

   - Information of the data: Name, Indicator, Description, Source, etc.
   - Explain why you chose the indicator
   - List questions you want to study

---

2. Explore the data using visualization using `ggplot2`

   - Create various charts
   - Create at least one chart with at least two categocial variables and at least one numerical variable.
   - Create at least one chart with at least two numerical variables and at least one categorical variable.

3. Observations based on your data visualization, and difficulties and questions encountered if any.

Due: 2023-01-23 23:59:00. Submit your R Notebook file in Moodle (The Fourth Assignment). Due on Monday!