

# Download data from WID.world into R

Author: Thomas Blanchet (updated by Ignacio Flores)  
Paris School of Economics – EHESS

February 25, 2025

The World Inequality Database (WID.world) is an extensive source on the historical evolution of the distribution of income and wealth both within and between countries. It relies on the combined effort of an international network of over a hundred researchers covering more than seventy countries from all continents.

Anyone can access and plot the data through the website WID.world. For more advanced users, we provide the R package `wid`, which lets them download the data from WID.world directly into R.<sup>1</sup> It exports a single function called `download_wid`. This vignette explains how to use it.

## Arguments of the function

The command `download_wid` has the following arguments:

```
download_wid(  
  indicators, # Codes corresponding to indicators to retrieve  
  areas, # Areas (mostly countries) for which to retrieve the indicators  
  years, # Years for which to retrieve the indicators  
  perc, # Percentiles (part of the distribution)  
  ages, # Age groups (adults, all ages, elderly, etc.)  
  pop, # Population type (individual, households, tax units, etc.)  
  metadata, # Logical: should it fetch metadata too (eg. sources, etc.)  
  verbose, # Logical: should it display messages showing progress  
  include_extrapolations # Logical: should it include data based on extrapolations/interpolations  
)
```

**Indicators** The argument `indicators` is a vector of 6-letter codes that corresponds to a given series type for a given income or wealth concept. The first letter correspond to the type of series. Some of the most common possibilities include:

---

<sup>1</sup>A similar package for Stata users exists: see <http://econpapers.repec.org/software/bocbocode/s458357.htm>.

one-letter code	description
a	average (local currency unit, last year's prices)
b	inverted Pareto-Lorenz coefficient (no unit)
f	female population (fraction, between 0 and 1)
g	Gini coefficient (between 0 and 1)
i	index (no unit)
n	population (people)
s	share (fraction, between 0 and 1)
t	threshold (local currency unit, last year's prices)
m	total (local currency unit, last year's prices)
p	proportion of women (fraction, between 0 and 1)
w	wealth-to-income ratio or labor/capital share (fraction of national income)
r	Top 10/Bottom 50 ratio (no unit)
x	exchange rate (market or PPP) (local currency unit per foreign currency)
e	Total emissions (Tons of CO2 equivalent emissions)
k	Per capita emissions (Tons of CO2 equivalent emissions)
l	Average per capita group emissions (Tons of CO2 equivalent per capita emissions)

Type `?wid_series_type` to access the complete list. The next five letters correspond a concept (usually of income and wealth). Some of the most common possibilities include:

five-letter code	description
ptinc	pre-tax national income
pllin	pre-tax labor income
pkkin	pre-tax capital income
fiinc	fiscal income
hweal	net personal wealth

Type `?wid_concepts` to access the complete list. For example, `sfiinc` corresponds to the share of fiscal income, `ahweal` corresponds to average personal wealth. If you don't specify any indicator, it defaults to "all" and downloads all available indicators.

**Area codes** All data in WID.world is associated to a given area, which can be a country, a region within a country, an aggregation of countries (eg. a continent), or even the whole world. The argument `areas` is a vector of codes that specify the areas for which to retrieve data. Countries and world regions are coded using 2-letter ISO codes. Country subregions are coded as XX-YY where XX is the country 2-letter code. Type `?wid_area_codes` to access the complete list of area codes. If you don't specify any area, it defaults to "all" and downloads data for all available areas.

**Years** All data in WID.world correspond to a year. Some series go as far back as the 1800s. The argument `years` is a vector of integer that specify those years. If you don't specify any year, it defaults to "all" and downloads data for all available years.

**Percentiles** The key feature of WID.world is that it provides data on the whole distribution, not just totals and averages. The argument `perc` is a vector of strings that indicate for which part of the distribution the data should be retrieved. For share and average variables, percentiles correspond to percentile ranges and take the form `pXXpYY`. For example the top 1% share correspond to `p99p100`. The top 10% share excluding the top 1% is `p90p99`. Thresholds associated to the percentile group `pXXpYY` correspond to the minimal income or wealth level that gets you into the group. For example, the threshold of the percentile group `p90p100` or `p90p91` correspond to the 90% quantile. Variables with no distributional meaning use the percentile `p0p100`. See <http://wid.world/percentiles> for more details. If you don't specify any percentile, it defaults to "all" and downloads data for all available parts of the distribution.

**Age groups** Data may only concern the population in a certain age group. The argument `ages` is a vector of age codes that specify which age categories to retrieve. Ages are coded using 3-digit codes. Some of the most common possibilities include:

3-digit code	description
999	all ages
014	ages 0-14
156	ages 15-64
997	ages 65+
991	ages 0-19
992	ages 20+ (adults, including elderly)
993	ages 20-39
994	ages 40-59
995	ages 60+
996	ages 20-64 (adults, excluding elderly)
998	ages 80+

Type `?wid_age_codes` to access the complete list of age codes. If you don't specify any age, it defaults to "all" and downloads data for all available age groups.

**Population types** The data in WID.world can refer to different types of population (i.e. different statistical units). The argument `pop` is a vector of population codes. They are coded using one-letter codes. Some of the most common possibilities include:

one-letter code	description
i	individuals
j	equal-split adults (i.e., income or wealth divided equally among spouses)
m	male
f	female
t	tax units
e	employed

Type `?wid_population_codes` to access the complete list of population types. If you don't specify any code, it defaults to "all" and downloads data for all types of population.

**Metadata** All data in WID.world is associated to a metadata giving in particular sources and methodological details. If the argument `metadata` is `TRUE`, the command will download those as well. Default is `FALSE`.

**Extrapolations/interpolations** Some of the data on WID.world is the result of interpolations (when data is only available for a few years) or extrapolations (when data is not available for the most recent years) that are based on much more limited information than other data points. We include these interpolations/extrapolation by default as a convenience, and also because these values are used to perform regional aggregations. Yet we stress that these estimates, especially at the level of individual countries, can be fragile.

For many purposes, it can be preferable to exclude these data points. For that, use the option `include_extrapolations = FALSE`.

**Verbose** By default, the command is silent. If you set `verbose = TRUE`, it will output some information on the progress of the request.

## Usage

Although all arguments default to "all", you cannot download the entire database by typing `download_wid()`. The command requires you to specify either some indicators or some areas. To download the entire database, please visit <https://wid.world/data/> and choose "download full dataset".

If there is no data matching your selection on WID.world (maybe because you specified an indicator or an area that doesn't exist), the command will return `NULL` with a warning.

The command returns a sorted `data.frame` with the following columns: `country`, `variable`, `percentile`, `year` and `value`.

All monetary amounts for countries and country subregions are in constant local currency of the reference year (i.e. the previous year, the database being updated every year around July). Monetary amounts for world regions are in EUR PPP of the reference year. You can access the price index using the indicator `inyixx`, the PPP exchange rates using `xlccusp` (USD), `xlccup` (EUR), `xlcyup` (CNY), and the market exchange rates using `xlccusx` (USD), `xlccux` (EUR), `xlcyux` (CNY). To check the current reference year, you can look at when the price index is equal to 1.

Shares and wealth/income ratios are given as a fraction of 1. That is, a top 1% share of 20% is given as 0.2. A wealth/income ratio of 300% is given as 3.

## Examples

### Top 1% income share in the United States, 2010–2015

Here we simply seek the top 1% shares of pre-tax national income in the United States over the period 2010–2015. The function `download_wid` returns a `data.frame` with the desired data.

```
data <- download_wid(
  indicators = "sptinc", # Shares of pre-tax national income
  areas = "US", # In the United States
  years = 2010:2015, # Time period: 2010-2015
  perc = "p99p100" # Top 1% only
)
kable(data) # Pretty display of the data.frame
```

country	variable	percentile	year	value
US	sptinc992f	p99p100	2010	0.1650
US	sptinc992f	p99p100	2011	0.1685
US	sptinc992f	p99p100	2012	0.1804
US	sptinc992f	p99p100	2013	0.1726
US	sptinc992f	p99p100	2014	0.1785
US	sptinc992f	p99p100	2015	0.1757
US	sptinc992i	p99p100	2010	0.1896
US	sptinc992i	p99p100	2011	0.1926
US	sptinc992i	p99p100	2012	0.2060
US	sptinc992i	p99p100	2013	0.1963
US	sptinc992i	p99p100	2014	0.2013
US	sptinc992j	p99p100	2010	0.1789
US	sptinc992j	p99p100	2011	0.1808
US	sptinc992j	p99p100	2012	0.1949
US	sptinc992j	p99p100	2013	0.1846
US	sptinc992j	p99p100	2014	0.1897
US	sptinc992j	p99p100	2015	0.1889
US	sptinc992m	p99p100	2010	0.2022
US	sptinc992m	p99p100	2011	0.2031
US	sptinc992m	p99p100	2012	0.2178
US	sptinc992m	p99p100	2013	0.2064
US	sptinc992m	p99p100	2014	0.2106
US	sptinc992m	p99p100	2015	0.2117
US	sptinc992t	p99p100	2010	0.1970
US	sptinc992t	p99p100	2011	0.2005
US	sptinc992t	p99p100	2012	0.2142
US	sptinc992t	p99p100	2013	0.2036
US	sptinc992t	p99p100	2014	0.2092
US	sptinc992t	p99p100	2015	0.2086
US	sptinc996i	p99p100	2010	0.1849
US	sptinc996i	p99p100	2011	0.1880
US	sptinc996i	p99p100	2012	0.2045
US	sptinc996i	p99p100	2013	0.1822
US	sptinc996i	p99p100	2014	0.1871

country	variable	percentile	year	value
US	sptinc996i	p99p100	2015	0.1862
US	sptinc999i	p99p100	2010	0.1896
US	sptinc999i	p99p100	2011	0.1926
US	sptinc999i	p99p100	2012	0.2060
US	sptinc999i	p99p100	2013	0.1963
US	sptinc999i	p99p100	2014	0.2013
US	sptinc999i	p99p100	2015	0.2007

If we also request the metadata, the `data.frame` also contains additional columns with extra information.

```
library(plyr)

## -----

## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)

## -----

##
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':
##
##   arrange, count, desc, failwith, id, mutate, rename, summarise,
##   summarize

## The following object is masked from 'package:purrr':
##
##   compact

data <- wid::download_wid(
  indicators = "sptinc", # Shares of pre-tax national income
  areas = "US", # In the United States
  years = 2010:2015, # Time period: 2010-2015
  perc = "p99p100", # Top 1% only
  metadata = TRUE # Also request metadata
)

##
## Missing Metadata:
```

```
##
## Variable: sptinc_p99p100_992_i
## Missing fields: method
## Areas: US
```

```
colnames(data)
```

```
## [1] "country"      "countryname" "variable"     "percentile"   "year"
## [6] "value"        "shortname"   "shortdes"     "pop"          "age"
## [11] "source"       "imputation"  "quality"      "method"
```

Here, the metadata is the same for all observations because we only requested one variable.

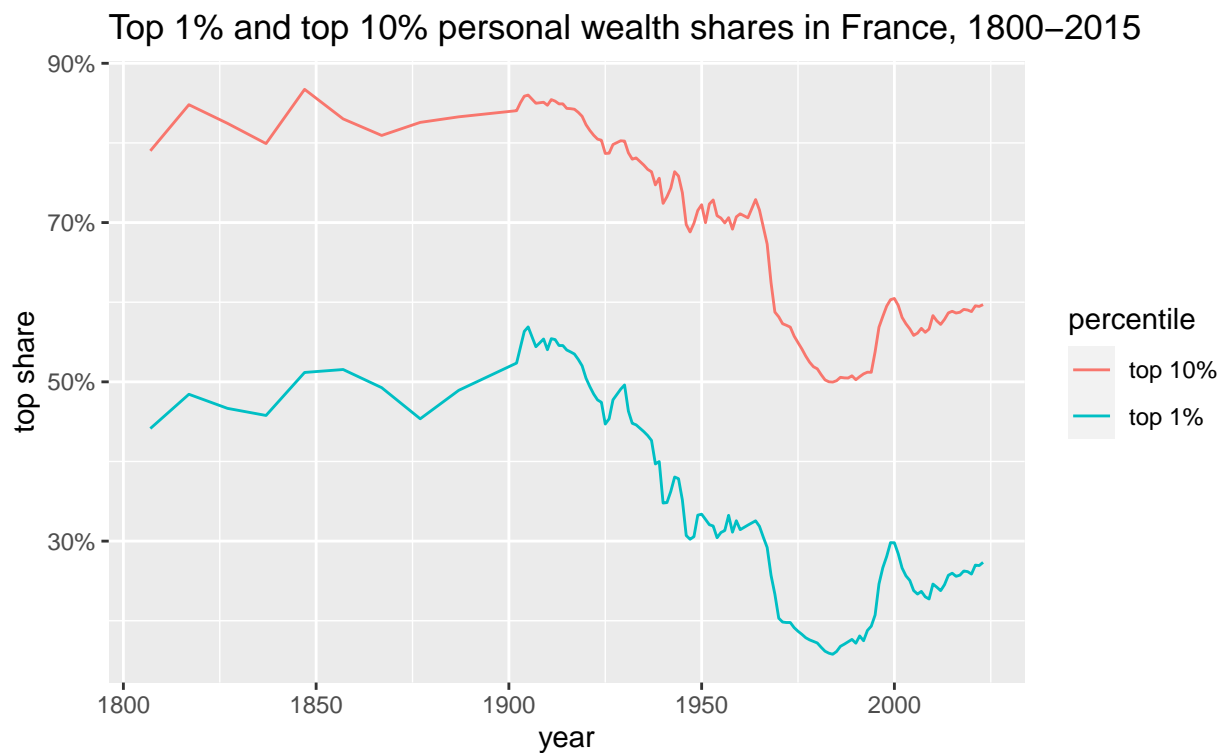
## Plot top wealth shares in France since the 1800s

In this example, we still select only one indicator, but we ask for two different percentiles. The function still returns a data.frame in “long” format, which makes it easy to plot with ggplot2.

```
data <- download_wid(
  indicators = "shweal", # Shares of personal wealth
  areas = "FR", # In France
  perc = c("p90p100", "p99p100") # Top 1% and top 10%
)

library(ggplot2)
library(scales)

ggplot(data) +
  geom_line(aes(x = year, y = value, color = percentile)) +
  ylab("top share") +
  scale_y_continuous(label = percent) +
  scale_color_discrete(labels = c("p90p100" = "top 10%", "p99p100" = "top 1%")) +
  ggtitle("Top 1% and top 10% personal wealth shares in France, 1800-2015")
```



## Evolution of income for the bottom 50% of the population

We now focus solely on the bottom half of the population (p0p50), and look at the average pre-tax national income in three different countries (France, United States and China). Since we are looking at monetary amounts for three different countries, we need to convert them into the same currency using the purchasing power parities in the database.

```
# We use the tidyverse to manipulate the data, see http://tidyverse.org
library(tidyverse)

# Average incomes data
data <- download_wid(
  indicators = "aptinc", # Average pre-tax national income
  areas = c("FR", "CN", "US"), # France, China and United States
  perc = "p0p50", # Bottom half of the population
  ages = "992",
  pop = "j", # Equal-split individuals
  year = 1978:2015
) %>% dplyr::rename(value_lcu = value)

# Purchasing power parities with US dollar
ppp <- download_wid(
  indicators = "xlcusp", # US PPP
  areas = c("FR", "CN", "US"), # France, China and United States
  year = 2016 # Reference year only
```



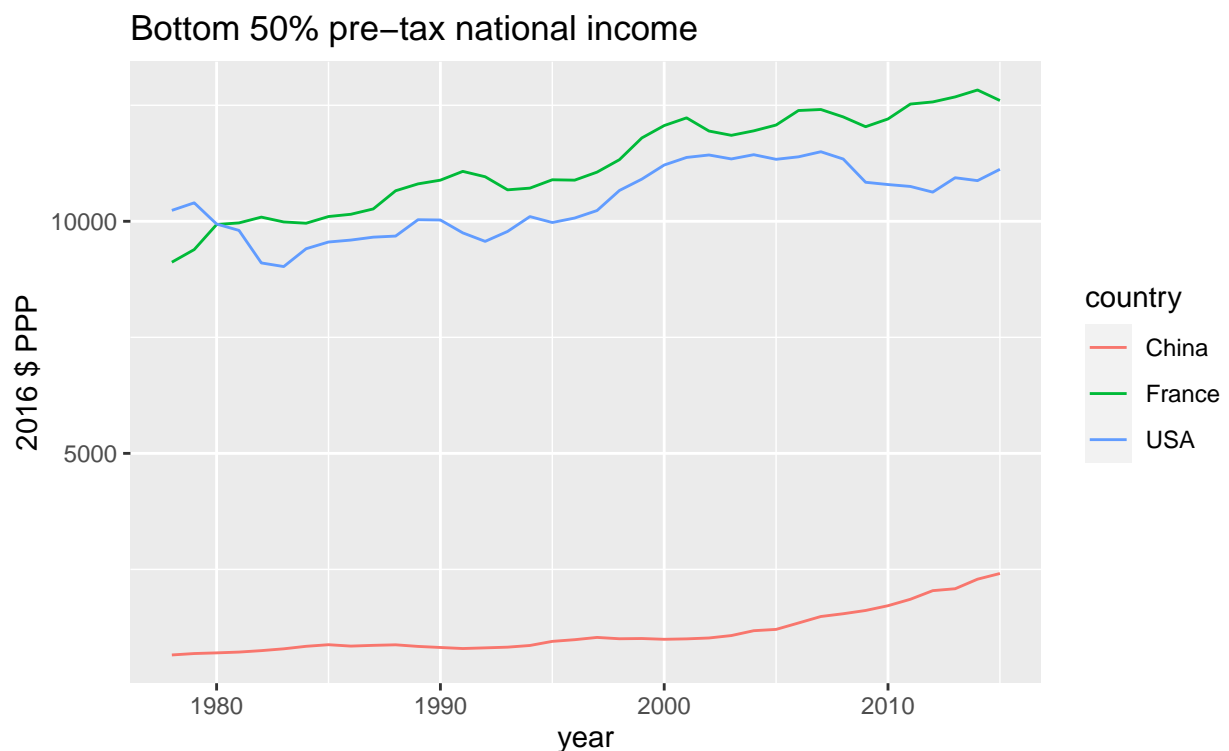
```

) %>% dplyr::rename(ppp = value) %>% select(-year, -percentile)

# Convert from local currency to PPP US dollar
data <- merge(data, ppp, by = "country") %>%
  mutate(value_ppp = value_lcu/ppp)

ggplot(data) +
  geom_line(aes(x = year, y = value_ppp, color = country)) +
  ylab("2016 $ PPP") +
  scale_color_discrete(labels = c("CN" = "China", "US" = "USA", "FR" = "France")) +
  ggtitle("Bottom 50% pre-tax national income")

```



## Evolution of national income over long period

We now plot the evolution of average net national income per adult in France, Germany, the United Kingdom and the United States.

```

# Average national income data
data <- download_wid(
  indicators = "anninc", # Average net national income
  areas = c("FR", "US", "DE", "GB"),
  ages = 992 # Adults
) %>% dplyr::rename(value_lcu = value)

# Purchasing power parities with US dollar
ppp <- download_wid(

```

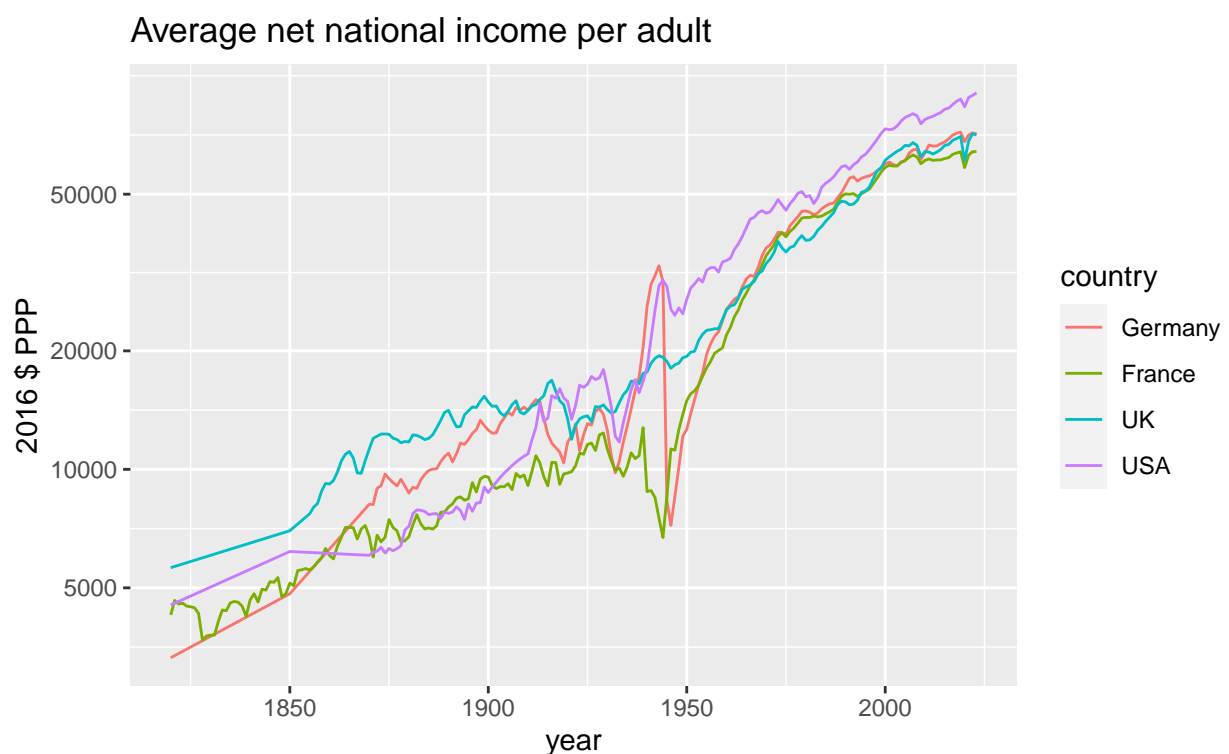
```

indicators = "xlcusp", # US PPP
areas = c("FR", "US", "DE", "GB"), # France, China and United States
year = 2016 # Reference year only
) %>% dplyr::rename(ppp = value) %>% select(-year, -percentile)

# Convert from local currency to PPP US dollar
data <- merge(data, ppp, by = "country") %>%
  mutate(value_ppp = value_lcu/ppp)

ggplot(data) +
  geom_line(aes(x = year, y = value_ppp, color = country)) +
  scale_y_log10(breaks = c(2e3, 5e3, 1e4, 2e4, 5e4)) +
  ylab("2016 $ PPP") +
  scale_color_discrete(
    labels = c("US" = "USA", "FR" = "France", "DE" = "Germany", "GB" = "UK")
  ) +
  ggtitle("Average net national income per adult")

```



### Divergence of incomes in the United States since 1970

Yet another way of observing an increase in inequality is to observe how the different fractiles of the distribution have evolved since a reference year. In the following graph, you can see that the different percentiles of the US distribution of pre-tax national income had a similar evolution throughout the 1970s, and then started to diverge after 1980.

```

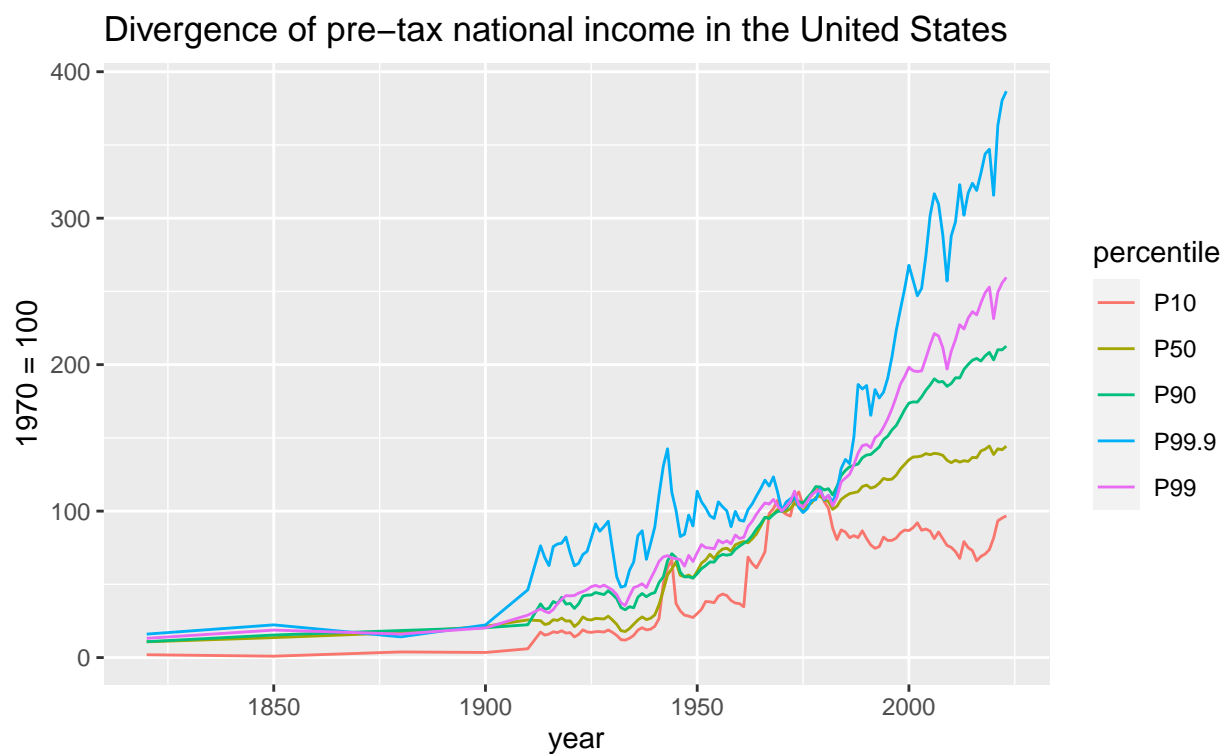
data <- download_wid(
  indicators = "tptinc", # Thresholds of pre-tax national income
  areas = "US", # United States
  pop = "j",
  perc = c("p10p100", "p50p100", "p90p100", "p99p100", "p99.9p100")
)

# Keep the value for 1970 in a separate data.frame
data1970 <- data %>% filter(year == 1970) %>%
  dplyr::rename(value1970 = value) %>%
  select(-year)

# Divide series by the reference year (1970)
data <- merge(data, data1970, by = c("country", "percentile")) %>%
  mutate(value = 100*value/value1970)

ggplot(data) +
  geom_line(aes(x = year, y = value, color = percentile)) +
  ylab("1970 = 100") +
  scale_color_discrete(
    labels = c("p10p100" = "P10", "p50p100" = "P50", "p90p100" = "P90",
               "p99p100" = "P99", "p99.9p100" = "P99.9")
  ) +
  ggtitle("Divergence of pre-tax national income in the United States")

```

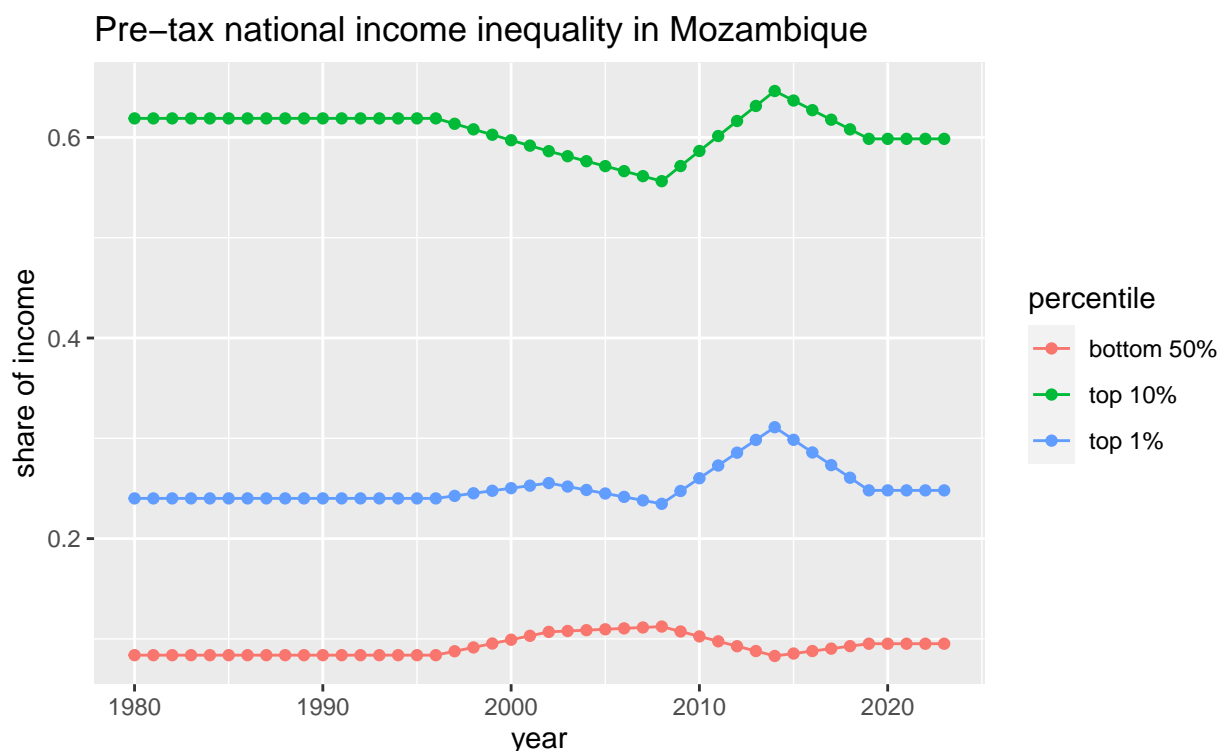


## Including or excluding extrapolations

In some countries, many data points are the result of interpolations or extrapolations. For example, estimates in most African countries are based on surveys that are only realized every few years, which we interpolate to produce yearly series and perform regional aggregations. For example, take the inequality series for Mozambique:

```
data <- download_wid(
  indicators = "sptinc", # Shares of pre-tax national income
  areas = "MZ", # Mozambique
  perc = c("p0p50", "p90p100", "p99p100") # Bottom 50%, top 10% and top 1%
)

ggplot(data, aes(x = year, y = value, color = percentile)) +
  geom_line() + geom_point() +
  ylab("share of income") +
  scale_color_discrete(
    labels = c("p0p50" = "bottom 50%", "p90p100" = "top 10%", "p99p100" = "top 1%")
  ) +
  ggtitle("Pre-tax national income inequality in Mozambique")
```



The linear interpolation is quite visible. In some contexts, this might be undesirable. To exclude interpolated points, use `include_extrapolations = FALSE`:

```
data <- download_wid(
  indicators = "sptinc", # Shares of pre-tax national income
```

```

areas = "MZ", # Mozambique
perc = c("p0p50", "p90p100", "p99p100"), # Bottom 50%, top 10% and top 1%
include_extrapolations = FALSE # Do not include interpolations
)

ggplot(data, aes(x = year, y = value, color = percentile)) +
  geom_line() + geom_point() +
  ylab("share of income") +
  scale_color_discrete(
    labels = c("p0p50" = "bottom 50%", "p90p100" = "top 10%", "p99p100" = "top 1%")
  ) +
  ggtitle("Pre-tax national income inequality in Mozambique")

```

