Representing data

Refresher Mathematics for Economics course

Economics Department, University of Warwick

2023-09-08

Contents

Preface

This is a short e-book supporting the section on Data Appreciation of the Refresher Mathematics for Economics course at Warwick.

The book uses a variety of data sources to provides examples of representing data. The exposition is structured around three topics

1. Representing differences in living standards across countries and over time
2. Representing data on climate change
3. Representing income inequality

Data is analyzed using R (R Core Team, 2023) - one of the most commonly used programming languages for statistical computing and graphics.

The purpose of the book is three-fold.

1. To enhance your understanding of what data is and of different types of data structures, by using real world examples.
2. To provide examples of meaningful ways for representing data - in different contexts, using real-world data.
3. To provide a brief introduction to R and some of its common packages.

With reference to point 3. above, the book presents all R codes used together with the outputs from the code (such as graphs and tables). The codes are fully replicable - if you run all the blocks of code in your own R/RStudio installation (or on the cloud) in the order that they appear within each chapter, you will get identical results. This being said, a complete introduction to R is beyond the scope of this course and not all of the code will be explained in detail. You are nonetheless strongly encouraged to run the code yourself, and hopefully learn intuitively key aspects of the syntax of the language. Still, some parts of the document will include relatively advanced code, so do not worry if you do not understand everything (or even most of the code).

0.1 How to read this book

We imagine that there could be two possible ways of reading the book, depending on whether you want to simply reflect on different ways of visualizing the data, or you also want to gain some understanding of coding in R.

If you are not interested in using R to replicate the results yourself, then you can simply read through the e-book, reflect on the tables and figures presented, and their interpretation. This should be sufficient to gain some understanding of the variety of different types of tables and charts used to represent data.

However, a more thorough way of engaging with the book and with data, is to additionally try to reflect on the codes producing the representations, ideally by replicating them in your own R/RStudio installation or on the cloud. As some understanding of data analysis and coding are part of the essential toolkit of an economist, we would strongly recommend this approach.

0.1.1 How to run the codes yourself

To run the codes and replicate the results yourself, you need to run R either on your own computer or on the cloud. We propose two different ways of doing this.

First, if you would like to run the codes interactively (with the ability to make small changes in code and see how output is affected), but not install R on your machine yet, we have created a virtual online environment on mybinder.org where the codes from different chapters of the book can be run interactively. You can access the virtual environment here. Please do note that it is possible to get an error message - if this is the case try refreshing the page several times until (and it should) it works. This is perfectly sufficient for exploring all the codes related to the book and playing around by exploring how small changes in the code affect the output.

Alternatively, you can install R and RStudio (both R and RStudio are free and open-source) on your own machine and replicate the codes in your own RStudio installation. The advantage of this approach, is that you will be able to use the software for subsequent work on data analysis - something that may be useful during your degree. Additionally, after the installation, running from your own installation will likely work faster than in the virtual environment on the cloud.

- If you are using a University computer, R and RStudio should already be installed.
- To install R and RStudio on your own personal computer, first download the most recent version of R for your operating system from CRAN and follow the instructions there to install. Next, download the most recent version of RStudio Desktop for your operating system from Posit website and follow the instructions there to install.

Once you have your own installation, you will be able to copy chunks of code from the e-book into a script in your RStudio editor, and run from there. To make this easier, you can download all the data and Rscripts (containing all code) for the chapters from the book's GitHub repository. Follow the instructions there to set up your local environment.

0.1.2 Additional resources on R

Beyond this Refresher course, if you would like to learn more about programming with R, the Department provides an online course *Introduction to R* here. The course is open for all students, not for credit, and you can enroll by simply clicking on the link.

Chapter 1

Some basics

This chapter discusses some essential concepts related to data and data analysis in R, and should be read prior to the subsequent chapters of the book.

For ease of replication, all the data used in this chapter is available as an Excel file `data_ch0.xlsx`.

- You can replicate all analysis in the cloud here.
- Alternatively, to replicate the chapter in your own R installation, download the data file and corrsponding R-script from here and extract them inside a folder on your computer. Then set the working directory in R to the folder where the files are. For me, this is the following folder:

```r
setwd("/home/emil/Desktop/book")
```

In addition, run the following code to install all the R libraries that will be used for the analysis

```r
install.packages("readxl")
install.packages("dplyr")
install.packages("ggplot2")
install.packages("tidyr")
```

and load them

```r
library(readxl)
library(dplyr)
library(ggplot2)
library(tidyr)
```

1.1 Data

While a precise definition of *data* is beyond the scope of this book, for practical purposes data can be understood as a structured collection of observations on a number of characteristics or variables. Data is typically organized into a structure in the form of a table (or matrix) with rows corresponding to distinct observations and columns corresponding to different variables.

Data could be obtained in a variety of ways and from a variety of sources, and prior to analyzing data it is essential to understand the structure of the dataset obtained from a given source.

While the rest of the book uses larger datasets, for clarity this chapter uses a mini-dataset obtained as a subsample of the data from the next chapter. This "mini-data" is stored in the Excel file `data_ch0.xlsx` under the sheet `mini`. First, load the data into R, allocating it to a *dataframe*[1] called `data`, and print it as follows

```r
data <- read_excel("data_ch0.xlsx", sheet="mini")

data
```

```
## # A tibble: 4 x 5
##   Country      Region                    `2019` `2020` `2021`
##   <chr>        <chr>                      <dbl>  <dbl>  <dbl>
## 1 Bangladesh   South Asia                  2210   2300   2570
## 2 Switzerland  Europe and Central Asia    83160  82490  88910
## 3 Turkmenistan Europe and Central Asia     7080     NA     NA
## 4 UK           Europe and Central Asia    43380  38590  44790
```

The "mini-dataset" includes information on region and GNI per capita (PPP)[2] in 2019, 2020 and 2021 for four countries. The data is structured in the form of a table with each row corresponding to an individual country. It includes

[1] In R and many other programming languages, a *dataframe* is a named object storing data in a 2-dimensional table of rows and columns, similarly to a spreadsheet.

[2] **Gross National Income (GNI)** is the total income earned by a country's residents (individuals or companies) from both domestic and foreign sources over a period of time. GNI is closely related to GDP, but accounts differently for income earned domestically and abroad.

- For example, when a UK resident earns income from operations in both UK and France, the income earned in France is part of UK's GNI (but not UK's GDP) and France's GDP (but not France's GNI).

GNI per capita (GNI pc), the average income earned by a country's residents, is commonly used as a measure of material standard of living. However, given differences in price levels across countries, the same amount of income will have the "power" to purchase different amounts of goods and services in different countries. In order to account for this, the World Bank produces Purchasing Power Parity (PPP) adjusted measures of GNI per capita, which are more suitable for comparisons of living standard between countries.

In particular, in what follows we use data GNI per capita PPP-adjusted by the so called *Atlas method* - the indicator based on which the World Bank groups countries into low, middle, and high-income.

the country's region in the second column. Finally, the value of GNI per capita (PPP) for years 2019-2021 is represented in separate columns, labeled by the year. Note that there is some missing data (the GNI per capita (PPP) of Turkmenistan in 2020 and 2021).

1.1.1 Cross-sectional, time-series, and panel data

One important aspect of data structure is whether the data is cross-sectional, time-series, or panel.

- *Cross-sectional* data consists of observations on subjects (such as individuals, firms, countries) at a single point or period of time.
- *Time-series* data consists of observations of a single subject over several points or periods of time.
- *Panel* (or *longitudinal*) data consists of observations on subjects over points or periods of time. It has both cross-sectional and time-series dimensions.

The data loaded above is an example of panel data, as it includes observations on four cross-sectional units (countries) for three distinct periods of time (years).

1.1.1.1 Panel data - long and wide form

When panel data is structured so that cross-sectional units vary across rows and time periods across columns (as above) it is said to be in *wide form*. While wide form is seemingly a natural way to arrange data, for most computational purposes it is more appropriate to arrange panel data in *long form* - i.e., collapsing the time period so that a row of the data now identifies a cross-sectional unit at a specific period of time:

```r
long.data <- data %>%
  pivot_longer(
    cols = `2019`:`2021`,
    names_to = "Year",
    values_to = "GNIpc"
  )

long.data
```

```
## # A tibble: 12 x 4
##    Country      Region                  Year  GNIpc
##    <chr>        <chr>                   <chr> <dbl>
##  1 Bangladesh   South Asia              2019   2210
##  2 Bangladesh   South Asia              2020   2300
##  3 Bangladesh   South Asia              2021   2570
##  4 Switzerland  Europe and Central Asia 2019  83160
##  5 Switzerland  Europe and Central Asia 2020  82490
##  6 Switzerland  Europe and Central Asia 2021  88910
##  7 Turkmenistan Europe and Central Asia 2019   7080
##  8 Turkmenistan Europe and Central Asia 2020     NA
##  9 Turkmenistan Europe and Central Asia 2021     NA
## 10 UK           Europe and Central Asia 2019  43380
## 11 UK           Europe and Central Asia 2020  38590
## 12 UK           Europe and Central Asia 2021  44790
```

The same dataset in “long” form has 12 observations, with each observation (row) corresponding to a country in a given year. The dataset includes 4 variables (columns) identifying the country, its region, the year, and the country’s GNI per capita (PPP) in the specific year.

As a side note, panel data in long form can also be transformed into wide form:

```
wide.data <- long.data %>%
  pivot_wider(names_from = Year, values_from = GNIpc)

wide.data
```

```
## # A tibble: 4 x 5
##   Country      Region                  `2019` `2020` `2021`
##   <chr>        <chr>                    <dbl>  <dbl>  <dbl>
## 1 Bangladesh   South Asia                2210   2300   2570
## 2 Switzerland  Europe and Central Asia  83160  82490  88910
## 3 Turkmenistan Europe and Central Asia   7080     NA     NA
## 4 UK           Europe and Central Asia  43380  38590  44790
```

1.1.1.2 Cross-sectional data

An example of cross-sectional data, is the subset of the panel data above, at a particular year only. It has the following structure:

```
cs.data <- subset(long.data, Year==2021)

cs.data
```

```
## # A tibble: 4 x 4
##   Country      Region                  Year  GNIpc
##   <chr>        <chr>                   <chr> <dbl>
## 1 Bangladesh   South Asia              2021   2570
## 2 Switzerland  Europe and Central Asia 2021  88910
## 3 Turkmenistan Europe and Central Asia 2021     NA
## 4 UK           Europe and Central Asia 2021  44790
```

In the context of our dataset, cross-sectional data is appropriate for understanding differences in living standards (measured by GNI per capita (PPP)) between countries (and between and within regions) at a specific point in time.

1.1.1.3 Time-series data

An example of time-series data, is the subset of the panel data above, for a particular country only. It has the following structure:

```r
ts.data <- subset(long.data, Country=="UK")

ts.data
```

```
## # A tibble: 3 x 4
##   Country Region                  Year  GNIpc
##   <chr>   <chr>                   <chr> <dbl>
## 1 UK      Europe and Central Asia 2019  43380
## 2 UK      Europe and Central Asia 2020  38590
## 3 UK      Europe and Central Asia 2021  44790
```

In the context of our dataset, time-series data is appropriate for understanding the dynamic evolution of living standards (measured by GNI per capita (PPP)) in a given country over time.

As panel data has both cross-sectional and time-series dimensions, it is appropriate for analyzing both cross-country differences and dynamic evolution of living standards over time.

1.1.2 Visualizing the structure of your data

As illustrated above, before starting any data analysis, it is essential to get a clear understanding of the structure of your data. For this reason, it is important to always inspect the data (i.e., the “table” or rows and columns), understanding what does an observation (row of data) identify, how are variables (columns of data) recorded, and so forth. In the rest of the book, whenever new dataset is loaded we will explicitly print the first few rows of the data to clarify the structure. In R, this can be easily done by using the function `head()` which will print the first few rows of a dataframe. The syntax is `head(name_of_dataframe)`.

While this is often sufficient for the purposes of the book, when you use more complicated datasets it is better to view the data in the data browser, using the function `View()` with syntax `View(name_of_dataframe)` which will return the whole “table” of data.

1.1.3 Categorical and quantitative variables

Another important aspect of the data is the nature of the variables in it. A complete discussion of the types of variables and data types in R is beyond the scope of the document but a key distinction relevant for the subsequent analysis is between quantitative and qualitative (categorical) variables.

To set ideas, consider the cross-sectional mini-dataset `cs.data` obtained above, consisting of 4 variables - `Country`, `Region`, `Year` and `GNIpc`.

- GNI per capita (measured in PPP adjusted international dollars) is clearly a quantitative variable. If we summarize the variable in R we can see that it has been correctly identified as quantitative

```
summary(cs.data$GNIpc)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    2570   23680   44790   45423   66850   88910       1
```

as the `summary()` function reports a number of summary statistics of the distribution (e.g. mean, median, etc).

- Region is clearly a categorical variable, as there is no single natural order of its possible values. We can see that among the four countries in the mini-dataset, three are from Europe and Central Asia, and one from South Asia. However, if we summarize the variable in R

```
summary(cs.data$Region)
```

```
##    Length     Class      Mode
##         4 character character
```

we see that it is classified as `character` (meaning text or string) which means that R does not recognize that three of the countries are from the same region. In such situations, it is important to explicitly tell R that the variable is categorical. In R, categorical variables are referred to as factor variables. We can declare the variable to be categorical as follows:

```
cs.data$Region <- as.factor(cs.data$Region)
```

Now note that a summary of the variable

```
summary(cs.data$Region)
```

```
## Europe and Central Asia             South Asia
##                       3                      1
```

correctly recognises that the data consists of 3 countries from Europe and Central Asia and one country from South Asia.

While this discussion is not necessarily crucial for understanding the data representations in the rest of the book, it is important for understanding some bits of code, as we will often have to declare the type of variables.

1.2 A note on R packages

While a large number of functions are available as default in any R installation, one of the strengths of R, relative to other languages for statistical analysis, is that it has an extensive collection of user-contributed packages (collections of functions) which extend its functionality beyond the basis packages.

In order to use such packages within your code you need to first install the relevant package by typing `install.packages("name_of_package")` and then loading them inside your code by typing `library(name_of_package)`. You will see that all chapters of the book use a variety of user-contributed packages which are installed and loaded in the beginning of the code. For example, at the beginning of this chapter we loaded four packages:

- `readxl` - which includes a number of functions for reading data from Excel files. One of the functions in the package is `read_excel()` which we used to read data from the Excel file. Since this is not a *base* function, the statement `data <- read_excel("data_ch0.xlsx")` would not have worked if we had not first installed and loaded the package.
- `dplyr` and `tidyr` - which include a large number of functions for reorganizing data. Without this the `pivot_longer()` function that we used to convert the panel dataset from wide to long format (this would have been possible with base R functions but much more cumbersome)
- `ggplot2` - this is an excellent package for producing graphs. At this stage we have not yet used the package but it will be used extensively throughout the book.

While a discussion of all the packages used is beyond the scope of the book, you can find all the key information related to them online. However, given the emphasis on producing graphs for the rest of the book, we will conclude this section with a brief discussion of the `ggplot2` package (Wickham, 2016). To illustrate the use of `ggplot()` we load the second sheet from the `data_ch0.xlsx` file and allocate it to a dataframe `csdata`

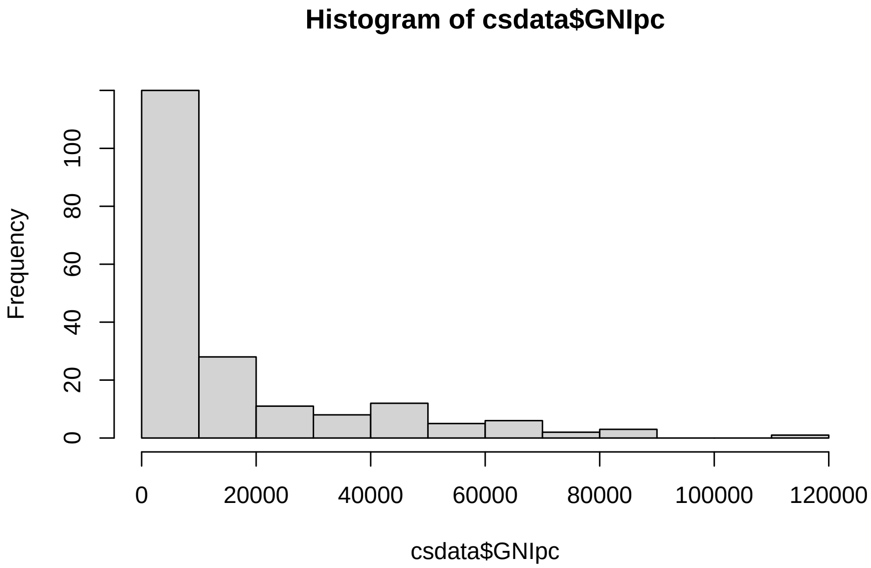
```
csdata <- read_excel("data_ch0.xlsx", sheet="csdata")

csdata
```

```
## # A tibble: 212 x 3
##    Country          Year GNIpc
##    <chr>           <dbl> <dbl>
##  1 Afghanistan      2019   530
##  2 Albania          2019  5230
##  3 Algeria          2019  4050
##  4 American Samoa   2019    NA
##  5 Andorra          2019 46530
##  6 Angola           2019  2040
##  7 Antigua          2019 17480
##  8 Argentina        2019 11220
##  9 Armenia          2019  4880
## 10 Aruba            2019 30330
## # i 202 more rows
```

Observe that the new dataset loaded is now a cross-section of observations for GNI per capita for 212 countries in 2019. We will say more about this data in the next chapter, but at this stage suppose that we want to produce a histogram summarizing the distribution of GNI per capita across countries in 2019.

R has a base in-built function `hist()` for producing histograms which we can use to produce a histogram of the GNI per capita distribution in our data as follows:

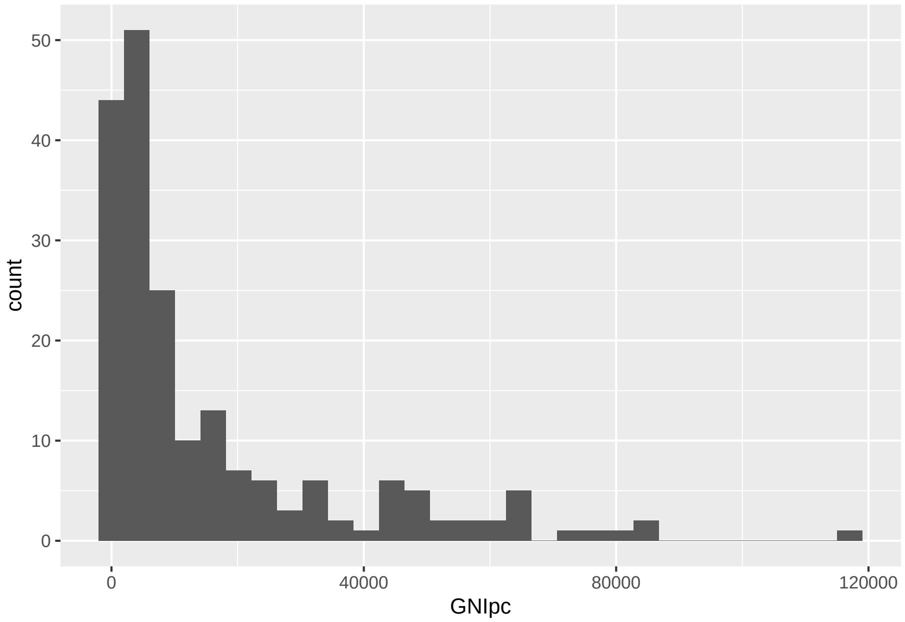
```
hist(csdata$GNIpc)
```



While the R's base in-built library for graphs `graphics` is sufficiently extensive (you can find more about it here), for the remainder of this book we will make use of the more advanced and customizable library `ggplot2` (more here).

To produce a histogram using `ggplot2`

```
ggplot(csdata, aes(x=GNIpc)) +
  geom_histogram()
```

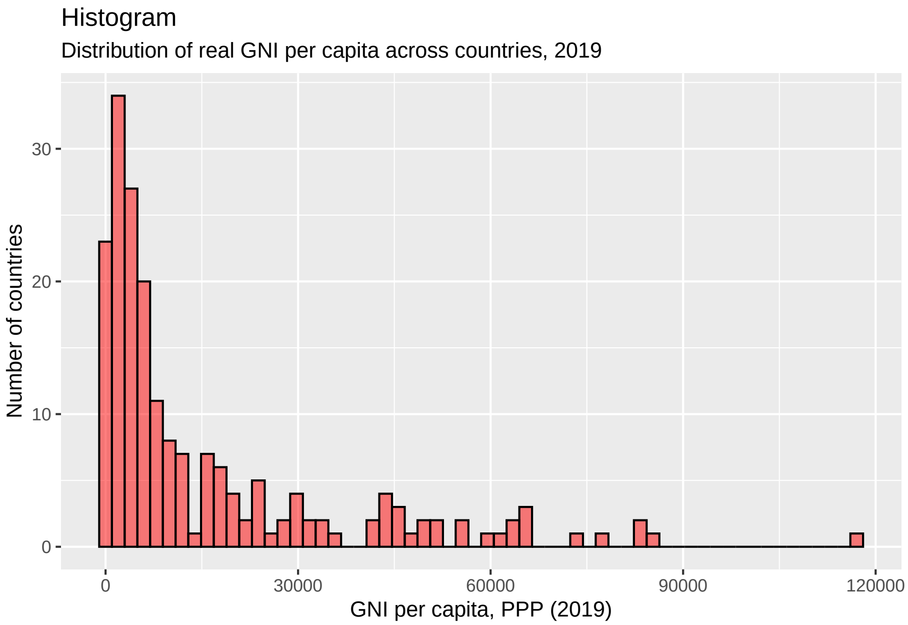


Before proceeding further let's briefly discuss the philosophy of the `ggplot` syntax.

- `ggplot` is called through the function `ggplot()` which takes as first argument the name of the dataframe used (and possibly specification of *aesthetics* - see below).
- Then with `+` we add *layers* of types of graphs we want to plot. In the above example `geom_histogram()` specifies we are to plot a histogram. We could instead use `geom_bar()` for barcharts, `geom_boxplot()` for boxplots, and so forth. You will see many examples of layers in the book.
- Either inside `ggplot()` or inside `geom_...()` we need to specify the variables based on which the graph is to be plotted. In `ggplot` this is specified via *aesthetics* through a function `aes()`. Loosely, this refers to a set of statements about what is being plotted - e.g., what goes on `x` and/or `y` axis, should different groups be plotted in different colors, etc.
- Then all this can be customized further, by adding labels, color schemes, etc, as we will see throughout the book.

For example, we can customize the histogram further, by changing the color and fill, adding labels and titles as below:

```
ggplot(csdata, aes(x=GNIpc)) +
  geom_histogram(bins=60, color="black", fill="red", alpha=0.5)+
  xlab("GNI per capita, PPP (2019)") + ylab("Number of countries") +
  labs(title="Histogram",
       subtitle="Distribution of real GNI per capita across countries, 2019")
```



This all sounds quite complicated, but the philosophy of plotting will hopefully become increasingly clear through examples.

Chapter 2

Representing differences in living standards across countries and over time

Most of this chapter uses data on GNI per capita (PPP)[1] from the World Bank's World Development Indicators (WDI) database (World Bank, 2023) available here. The main dataset used includes observations on annual GNI per capita (PPP) for 212 countries from 1973 to 2022. The chapter provides examples of using the data to represent differences in living standards between countries and over time.

For ease of replication, all the data used in the chapter is available as an Excel file `data_ch1.xlsx` with different datasets saved as different tabs.

- You can replicate all analysis in the cloud here.
- Alternatively, to replicate the chapter in your own R installation, download

[1] **Gross National Income (GNI)** is the total income earned by a country's residents (individuals or companies) from both domestic and foreign sources over a period of time. GNI is closely related to GDP, but accounts differently for income earned domestically and abroad.

- For example, when a UK resident earns income from operations in both UK and France, the income earned in France is part of UK's GNI (but not UK's GDP) and France's GDP (but not France's GNI).

GNI per capita (GNI pc), the average income earned by a country's residents, is commonly used as a measure of material standard of living. However, given differences in price levels across countries, the same amount of income will have the "power" to purchase different amounts of goods and services in different countries. In order to account for this, the World Bank produces Purchasing Power Parity (PPP) adjusted measures of GNI per capita, which are more suitable for comparisons of living standard between countries.

In particular, in what follows we use data GNI per capita PPP-adjusted by the so called *Atlas method* - the indicator based on which the World Bank groups countries into low, middle, and high-income.

the data file and corrsponding R-script from here and extract them inside a folder on your computer. Then set the working directory in R to the folder where the files are. For me, this is the following folder:

```
setwd("/home/emil/Desktop/book")
```

In addition, run the following code to install all the R packages that will be used for the analysis:

```
install.packages("readxl")
install.packages("tidyr")
install.packages("dplyr")
install.packages("ggplot2")
install.packages("forcats")
install.packages("stringr")
install.packages("maps")
install.packages("knitr")
```

and load them

```
library(readxl)
library(tidyr)
library(dplyr)
library(ggplot2)
library(forcats)
library(stringr)
library(maps)
library(knitr)
```

The complete dataset on GNI per capita (PPP) from WDI is available in the gnipcppp_long sheet of the data_ch1.xlsx Excel file. First, we load and inspect the the first few rows of the data

```
# Load full panel dataset in long form
longdata <- read_excel("data_ch1.xlsx", sheet = "gnipcppp_long")
head(longdata)
```

```
## # A tibble: 6 x 5
##   Country     Code  Region      Year rGNIpc
##   <chr>       <chr> <chr>      <dbl>  <dbl>
## 1 Afghanistan AFG   South Asia  1973     NA
## 2 Afghanistan AFG   South Asia  1974     NA
## 3 Afghanistan AFG   South Asia  1975     NA
## 4 Afghanistan AFG   South Asia  1976     NA
## 5 Afghanistan AFG   South Asia  1977     NA
## 6 Afghanistan AFG   South Asia  1978     NA
```

This is a panel dataset in long form, with each row identifying an individual country in a specific year. There are 10600 observations (country-year pairs) in total (212 countries followed over 50 years).

In the remainder of this chapter, we will illustrate how individual cross sections, individual time series, or the whole panel dataset can be utilized to insightfully represent information from the data.

The following code cleans the R environment so that the dataframes used in this section are no longer loaded:

```r
rm(list = ls())
```

2.1 Representing cross sectional data: Cross-country differences in living standards in 2019

This section uses a cross section of the panel dataset for 2019 only, and presents examples of insightful ways of representing this data, with view of understanding differences in (distribution of) living standards as measured by GNI per capita (PPP), across countries in 2019.

2.1.1 Loading data

First, to obtain cross-sectional data for 2019, we can load the full panel dataset in long form (found in the third sheet of the `data_ch1.xlsx` Excel file) and then only keep observations for 2019.

```r
# Load full panel dataset in long form
longdata <- read_excel("data_ch1.xlsx", sheet = "gnipcppp_long")

# Obtain a cross-section for Year 2019 only, and allocate to dataframe `csdata`

csdata <- subset(longdata, Year==2019)

# Prepare by declaring `Country`, `Code` and `Region` to be factor (or categorical) variables

csdata$Code <- as.factor(csdata$Code)
csdata$Region <- as.factor(csdata$Region)
csdata$Country <- as.factor(csdata$Country)

# Remove full panel dataset from memory
rm(longdata)
```

For completeness, the full cross sectional dataset we work with is presented in the following table

```r
csdata <- csdata %>% arrange(desc(rGNIpc))
```

```
csdata
```

```
## # A tibble: 212 x 5
##    Country        Code  Region                      Year rGNIpc
##    <fct>          <fct> <fct>                      <dbl>  <dbl>
##  1 Bermuda        BMU   North America               2019 117280
##  2 Isle of Man    IMN   Europe and Central Asia     2019  84490
##  3 Switzerland    CHE   Europe and Central Asia     2019  83160
##  4 Norway         NOR   Europe and Central Asia     2019  82430
##  5 Luxembourg     LUX   Europe and Central Asia     2019  77500
##  6 Iceland        ISL   Europe and Central Asia     2019  72980
##  7 Faroe Islands  FRO   Europe and Central Asia     2019  66380
##  8 USA            USA   North America               2019  66130
##  9 Cayman Islands CYM   Latin America and Caribbean  2019  65690
## 10 Ireland        IRL   Europe and Central Asia     2019  63570
## # i 202 more rows
```

The dataset contains 212 observations (rows of data) on 5 variables (columns). An observation corresponds to an individual country. Variables are as follows:

- `Country` is a categorical (factor) variable identifying the country.
- `Code` is a categorical (factor) variable containing the country code - a standard 3-letter identifier of a country. It contains exactly the same information as `Country`, so adds nothing further to the information in the dataset. Nonetheless, in some representations below we will be interested in identifying individual countries in graphs, and the country code may be more compact label for the country than its name.
- `Region` is a categorical (factor) variable, identifying a country's region according to the World Bank's standard classification of countries by regions. We will see more about that below. However, the presence of data on region adds a meaningful new dimension to the data as it allows us to represent not only features of the distribution of living standards across the world overall, but also between and within regions.
- `Year` lists the year of observation. The variable is inherited from the original panel dataset. However, it equals `2019` for all observations in the cross section. Given the lack of variation, it contains no meaningful information for the purposes of the following representations (aside from the fact that we are looking at distributions in 2019 which we already know).
- Finally, `rGNIpc` corresponds to GNI p.c. (PPP) - our main object of interest - and is a numeric variable. It is measured in *2022 international dollars* - a counterfactual currency which is constructed as if 1 international dollar has the same purchasing power in each country (of course, imperfectly).

2.1.2 Summary of key variables

To get a feel of the data, let's summarize two of the key variables[2]:

```
kable(summary(csdata$Region))
```

	x
East Asia and Pacific	34
Europe and Central Asia	57
Latin America and Caribbean	41
Middle East and North Africa	21
North America	3
South Asia	8
Sub-Saharan Africa	48

When `summary()` is used with a categorical variable, it returns the number of observations in the sample for each category (that is, the complete distribution, in terms of counts, in the sample). As we can see countries are categorized into 7 regions based on World Bank's region classification. We can see that different regions include different numbers of countries. For example, there are only three countries in North America (by inspection Bermuda, Canada and the United States). Most countries (57) are in the region of Europe and Central Asia.

```
summary(csdata$rGNIpc)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##     230    2195    6435   15890   19693  117280      16
```

When `summary()` is used with a continuous variabe, it returns a set of summary statistics for the distribution of the variable in the sample. For example

- we can see that GNI per capita is unevenly distributed between countries, ranging from 230 international dollars per year to 117280 international dollars per year.
- The mean (or average) GNI per capita in the sample is 15890 international dollars. Recall, that the sample mean equals the sum of all countries' GNI per capita divided by the number of countries.
- The median GNI per capita in the sample is 6435. Recall that the sample median equals the GNI per capita in the country such that half of the countries in the sample have lower, and half have higher GNI per capita.
- It can be seen that the average GNI per capita in the sample - 15890 - is well above the median - 6435. While both the mean and the median are measures of central tendency, it is important to know that in many cases it is not unambiguous how to think of the centre of a distribution, and different measures emphasise different notions of what "typical" value is.

[2] The function `kable()` is used here to format the tables better for the book. You can obtain the same information by simply using `summary(csdata$Region)` instead of `ktable(summary(csdata$Region))`, but will be formatted slightly differently - feel free to try.

- We see that the 25th percentile and 75th percentile of GNI per capita in the sample are 2195 and 19693 international dollars respectively. 25% of countries have GNI per capita of 2195 international dollars or less; 75% of countries have GNI per capita of 19693 dollars or less.
- One common measure of the spread or dispersion of a variable, called the *interquartile range* (IQR) is defined as the difference between the 75th and 25th percentile. Given the results we can see that the IQR for GNI per capita in the sample is 17498 (= 19693 - 2195) international dollars.
- Data is not available (recorded as having value `NA`) for 16 out of the 212 countries .

2.1.3 Maps (for fun)

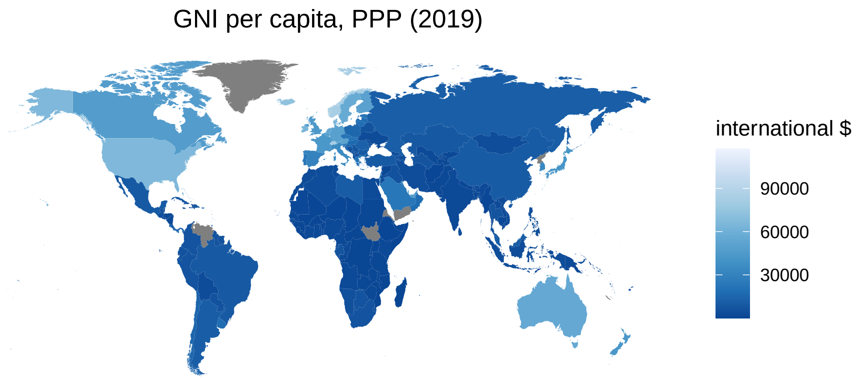
Given that the unit of observation is a country, a (possibly) attractive way of visualizing cross-country differences, is through a map.

The code below downloads an open-access geographic dataset of country coordinates which we can merge with our current data and produce a map. While the code is short , what happens under the hood is conceptually complex so we will make no attempt to explain the code. If you are interested in learning how to produce maps like this yourself, you can take a look at the discussion on the `maps` package website.

```
# Load world map data from server
world <- map_data("world")

# Merge world map data with GNI data; the function inner_join() takes two dataframes a

gnimap.data <- inner_join(world, csdata, by=c("region"="Country"))

# Set a simple map theme

plainmap <- theme(
  axis.text = element_blank(),
  axis.line = element_blank(),
  axis.ticks = element_blank(),
  panel.border = element_blank(),
  panel.grid = element_blank(),
  axis.title = element_blank(),
  panel.background = element_rect(fill = "white"),
  plot.title = element_text(hjust = 0.5)
)
```

Once we have allocated the merged data into a new dataframe, `gnimap.data`, containing information on both geography and GNI per capita, we can plot a map as follows:

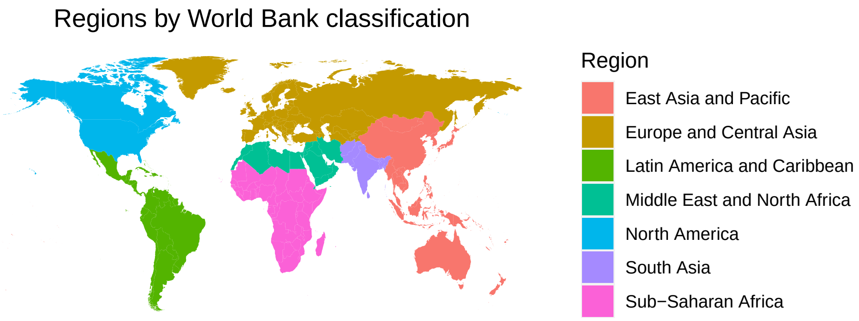
```
ggplot(data = gnimap.data, mapping = aes(x = long, y = lat, group = group)) +
  coord_fixed(1.3) +
  geom_polygon(aes(fill = rGNIpc)) +
  scale_fill_distiller(palette ="Blues", direction = -1, name="international $") +
  ggtitle("GNI per capita, PPP (2019)") + plainmap
```



Notice that the colour (shade of blue) of each country corresponds to its GNI p.c. (PPP) as noted in the legend. While a map is an informative way to represent the geographic variation in GNI per capita, there are many aspects of the distribution which are not so easy to observe and would be better visualized later on.

For the purposes of understanding our data, we next plot a map showing how the regions in our data (as categorised by the World Bank) are defined:

```
ggplot(data = gnimap.data, mapping = aes(x = long, y = lat, group = group)) +
  coord_fixed(1.3) +
  geom_polygon(aes(fill = Region)) +
  ggtitle("Regions by World Bank classification") + plainmap
```


Regions by World Bank classification
Region
East Asia and Pacific
Europe and Central Asia
Latin America and Caribbean
Middle East and North Africa
North America
South Asia
Sub−Saharan Africa

This is all we are going to say about maps at this stage. Before proceeding further, we remove objects that will no longer be needed from the R environment:

```
# Clean up

rm(gnimap.data, plainmap, world,longdata)
```

Also, for the purpose of some of the representations later it would be useful to remove the 16 observations with missing values of GNI per capita (discussed above) from the dataset:

```
csdata <- subset(csdata, !is.na(rGNIpc))
```

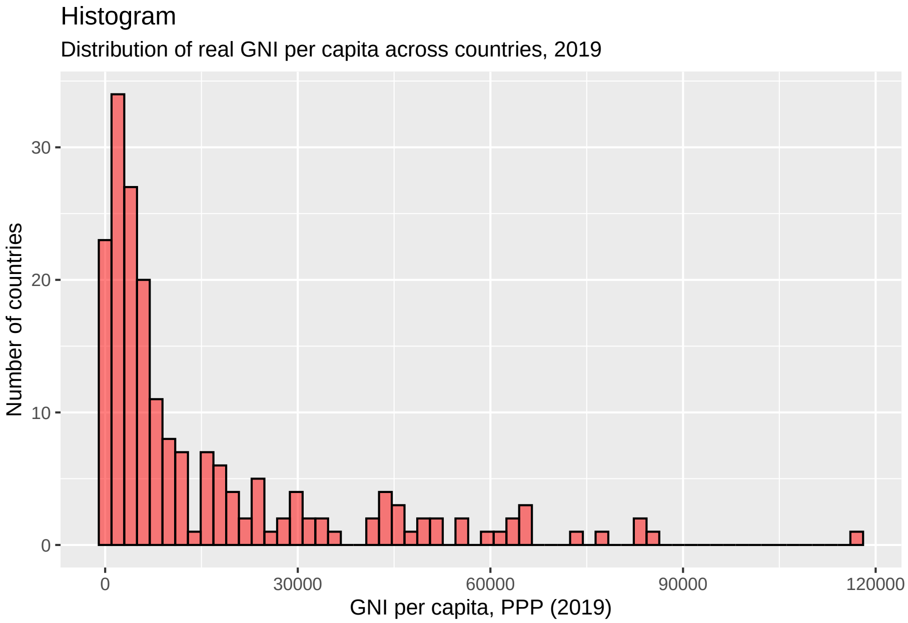
Note that this leaves us with a sample of 196 countries from the initial 212.

2.1.4 Histograms

A histogram of a random variable is a graph which divides the support of the variable, plotted on the horizontal axis, into a number of equally wide “bins”, with the height of each bin (on the vertical axis) equal to the number (or fraction) of observations falling into the bin. The histogram is a graphical representation of the distribution function of the variable.

Using `ggplot2`, we can produce a histogram of the distribution of GNI per capita across countries as follows

```
ggplot(csdata, aes(x=rGNIpc)) +
  geom_histogram(bins=60, color="black", fill="red", alpha=0.5)+
  xlab("GNI per capita, PPP (2019)") + ylab("Number of countries") +
  labs(title="Histogram",
       subtitle="Distribution of real GNI per capita across countries, 2019")
```



As can be seen from the histogram above, living standards (as measured by PPP-adjusted GNI per capita) are very unequally distributed across countries.

- The distribution of GNI p.c. is highly disperse (varying from close to 0 to about 115000 international dollars).
- In addition, the distribution is asymmetric, with much more countries in the left of the distribution, and fewer countries to the right forming a long right tail.
- The extent of asymmetry in a distribution can be formally measured by its *skewness*. The precise definition of skewness as a measure is beyond the scope of this text but skewness has negative value if the tail of the distribution is to the left, and positive value if the tail is to the right. The histogram suggests that GNI per capita (PPP) is strongly *positively skewed* in the sample.
- Consistently with this positive skewness the mean GNI p.c. (at 15890) is well above the median (at 6435) as we saw in the table of summary statistics above.

2.1.5 Quantiles

While a histogram provides a rich description of the distribution of a continuous variable, other types of graphs and tables can visualize better other aspects of the differences between countries in the data.

A way to get a quantitative feeling of the dispersion and asymmetry of the distribution is by calculating some of its quantiles. The n-th quantile of a distribution simply means the $100 \times n$th percentile. For example, the 0.1th quantile is the 10th percentile. The code below calculates the 0.1, 0.2,…,0.9th quantiles of the distribution:

```r
kable(quantile(csdata$rGNIpc,seq(0.1,0.9,0.1)))
```

	x
10%	910
20%	1880
30%	3270
40%	4490
50%	6435
60%	9510
70%	16355
80%	26960
90%	47355

We can see, for example, that

- 10% of the countries in the dataset have GNI p.c. of no more than 910 international dollars;
- 20% of the countries in the dataset have GNI p.c. of no more than 1880 international dollars;
- 50% of the countries in the dataset have GNI p.c. of no more than 6435 international dollars (the median).

By comparing different quantiles we can obtain a clear numerical sense of the differences between countries at different points of the distribution.

2.1.6 Quantile function

While the above table only reports 9 quantiles, in principle we can provide a table of more (for example, 100 quantiles - or percentiles) but so much information would be cumbersome to read in a table. Instead, we may equivalently represent such information graphically.

Given a random variable, e.g. GNI per capita (PPP) in 2019, its quantile function is defined as the mapping from $n \in [0, 1]$ to the n-th quantile of its distribution. For example, if $q(n)$ is the empirical quantile function for GNI per capita in our data, then given the observations of quantiles above, $q(0.1) = 910$,

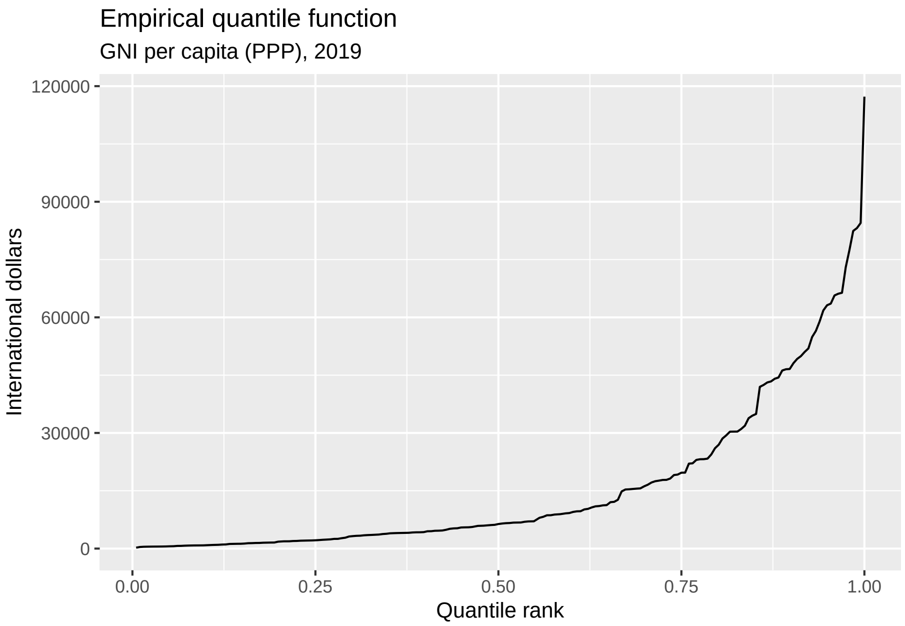
$q(0.2) = 1880$ and $q(0.5) = 6435$. Hence, the way to represent graphically all quantiles of the distribution, is by a graph of its quantile function.

One way to do this is as follows. First, create a new variable which records the quantile rank in the GNI p.c. distribution for each country (i.e., the quantile of GNI pc where each country is located)

```
# Create a variable rGNIpc.rank representing the quantile rank of each country in the distributio

csdata <- csdata %>% mutate(rGNIpc.rank = rank(rGNIpc)/length(rGNIpc))

# (Not essential) Sort country codes by GNI pc

csdata$Code <- fct_reorder(csdata$Code,csdata$rGNIpc)
```

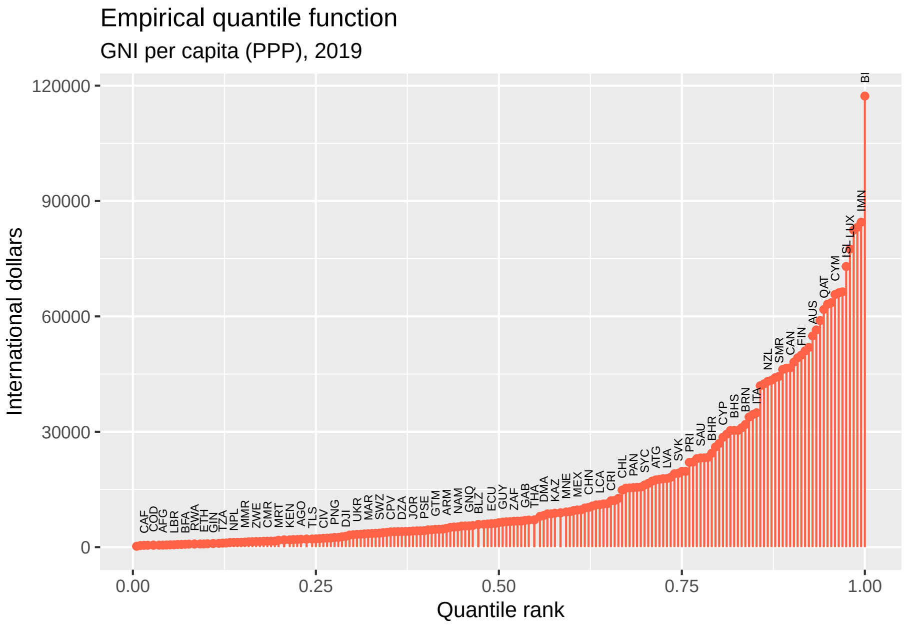
Next, plot quantile rank of each country against its corresponding GNI per capita (PPP)

```
ggplot(csdata, aes(y=rGNIpc, x=rGNIpc.rank)) +
  geom_line() +
  xlab("Quantile rank") + ylab("International dollars") + labs(title="Empirical quantile function
```



While above we used `geom_line()` to fit a curve passing through the coordinates of each country. We can make this explicit by instead using points for each country. Doing so, and customizing the graph a bit further yields the following

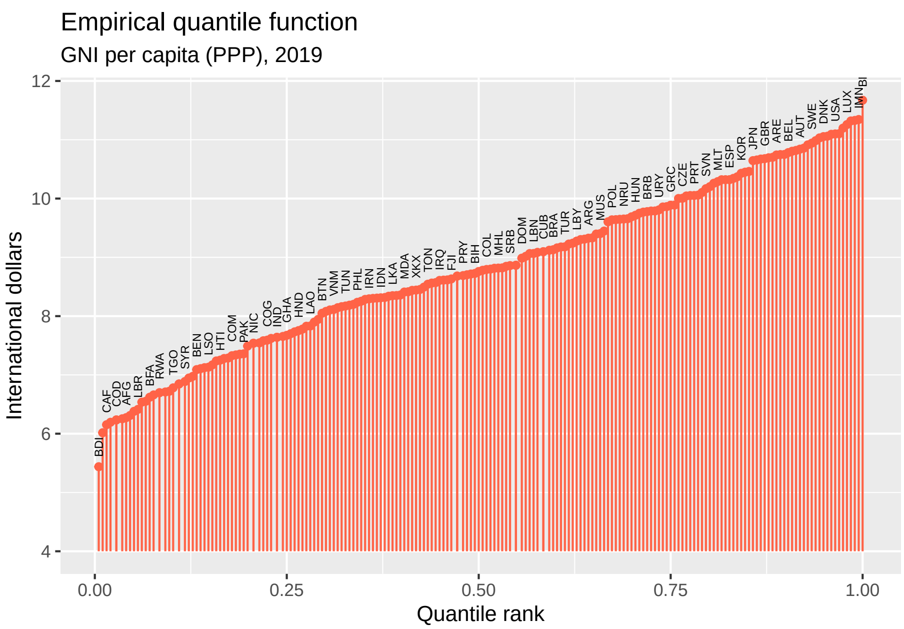
```
ggplot(csdata, aes(y=rGNIpc, x=rGNIpc.rank)) +
  geom_point(col="tomato") +
  geom_segment(aes(x=rGNIpc.rank, xend=rGNIpc.rank, y=0, yend=rGNIpc), col="tomato") +
  geom_text(aes(label=Code), check_overlap = TRUE, angle=90, hjust=-0.5, size=2) +
  xlab("Quantile rank") + ylab("International dollars") + labs(title="Empirical quantil
```



Relative to a histogram, this representation draws more attention to "divergence" of living standards across countries (for example, we can easily see how rich are the richest relative to the poorest).

For example, one interesting observation is as follows. Since countries are ordered by quantile rank (from poorest to richest) it is clear that the quantile function will be increasing. However, the empirical quantile function is not only increasing, but appears to exhibit approximately exponential pattern. To verify this, note that if we plotted the natural logarithm of GNI p.c. (rather than its level) on the vertical axis, we get "approximately" linear pattern:

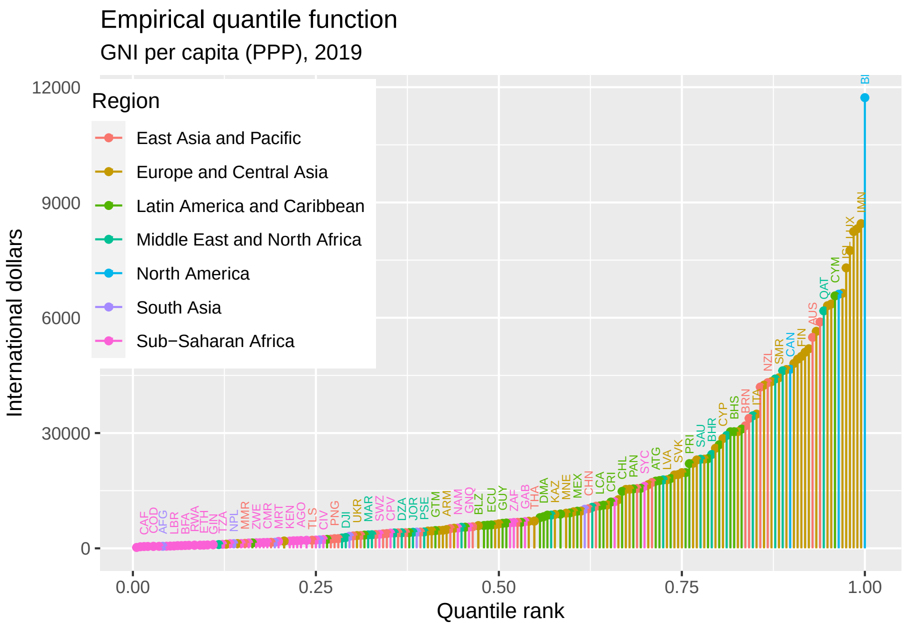
```
ggplot(csdata, aes(y=log(rGNIpc), x=rGNIpc.rank)) +
  geom_point(col="tomato") +
  geom_segment(aes(x=rGNIpc.rank, xend=rGNIpc.rank, y=4, yend=log(rGNIpc)), col="tomato
  geom_text(aes(label=Code), check_overlap = TRUE, angle=90, hjust=-0.5, size=2) +
  xlab("Quantile rank") + ylab("International dollars") + labs(title="Empirical quantil
```



This tells us something fairly specific about the distribution (and in particular its skewness). If we order countries in ascending order of GNI p.c. (PPP), each subsequent country is richer than the previous one by approximately the same proportion (rather than for example, by the same number of international dollars). This is yet another manifestation of the skewness of the distribution, characterised by a long and thinning right tail.

In order to get an idea of how the distribution varies between regions, we can make the colors elements different for different regions of the world:

```
ggplot(csdata, aes(y=rGNIpc, x=rGNIpc.rank, col=Region)) +
  geom_point() +
  geom_segment(aes(x=rGNIpc.rank, xend=rGNIpc.rank, y=0, yend=rGNIpc)) +
  geom_text(aes(label=Code), check_overlap = TRUE, angle=90, hjust=-0.5, size=2) +
  xlab("Quantile rank") + ylab("International dollars") + labs(title="Empirical quantile functio
```



The graph shows clear regional differences. For example, there is a large concentration of countries from Sub-Saharan Africa at the lowest quantiles, and countries from North America and Europe and Central Asia in the highest quantiles.

2.1.7 Visualizing distribution within and between regions

As already seen there seem to be systematic regional differences with respect to the distribution of GNI p.c. In what follows, we will look at ways to represent the data to understand these differences.

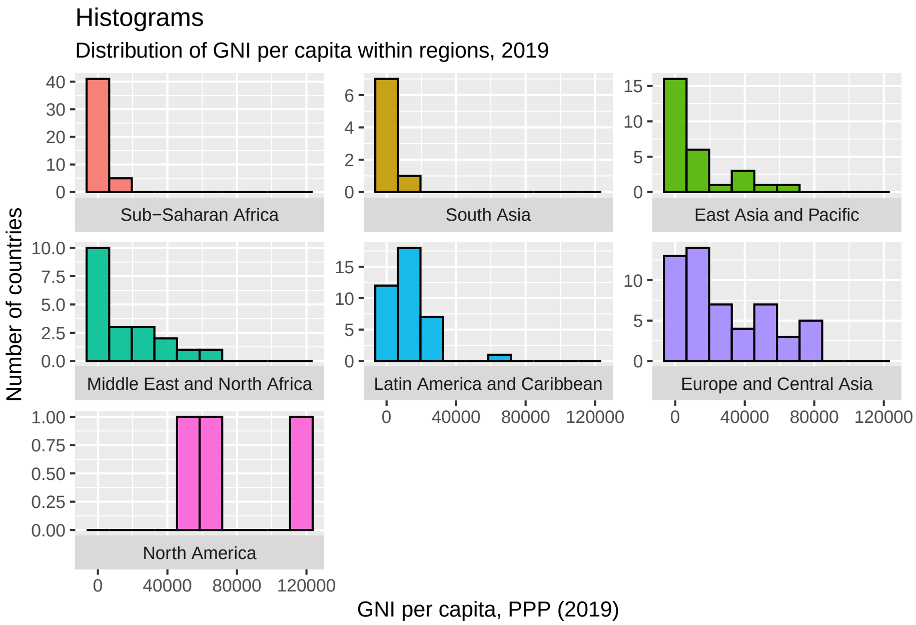
To have a clearer point of reference, let's first reorder the levels of the `Region` variable to be in ascending order of the median (per region):

```
# Rearrange region in order of the median GNI pc.
csdata$Region <- fct_reorder(csdata$Region,csdata$rGNIpc, .fun = median)
```

2.1.7.1 Histograms by region

Just like we can plot a histogram for the whole dataset, we can also plot separate histograms for regions:

```
ggplot(csdata, aes(x=rGNIpc, fill=Region)) +
  geom_histogram(bins=10,color="black", alpha=0.9)+
  labs(title="Histograms",
       subtitle="Distribution of GNI per capita within regions, 2019") +
  xlab("GNI per capita, PPP (2019)") + ylab("Number of countries") +
  facet_wrap(Region~., scales="free_y", strip.position = "bottom")+theme(legend.position = "none'
```



Note that all histograms have the same horizontal axis, but different scales on the vertical axes (this is achieved by `scales="free_y"`).

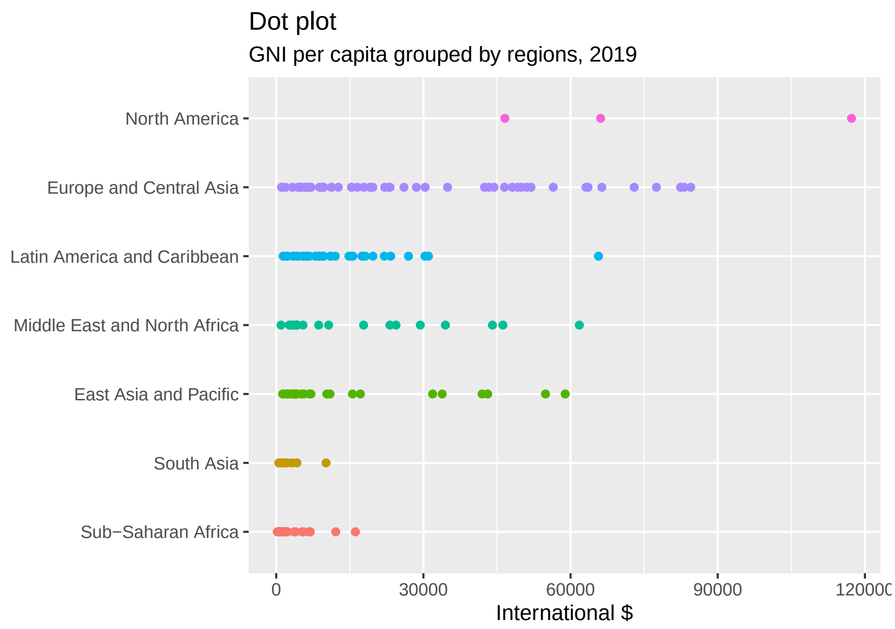
A few things coming out of the graph:

- Countries in Sub-Saharan Africa and South Asia tend to have low GNI p.c. relative to other regions.
- The three countries in North America - Bermuda, Canada and US have high GNI p.c.
- There dispersion across all regions, but the distribution Sub-Saharan Africa and South Asia appears fairly compressed in the histograms, indicating low dispersion relative to other regions.

2.1.7.2 Dot plots by region

Given the relatively small number of observations, an alternative way to represent the same information is through dot plots. Suppose that we measure GNI p.c. on the horizontal axis, draw a separate vertical intercept for each region and plot each country as a point:

```
ggplot(csdata, aes(x=Region, y=rGNIpc)) +
  geom_point(aes(col=Region), show.legend=F) +
  labs(title="Dot plot", subtitle="GNI per capita grouped by regions, 2019") +
  ylab("International $") + xlab("") +  coord_flip()
```



Unlike the histogram, we are not able to clearly see the density of the distribution for different ranges of GNI p.c. (as ranges with many countries contain too many points to differentiate between). For example, while we have 48 countries in Sub-Saharan Africa, most of them are far in the left tail of the distribution, and from the graph it appears as if there are fewer countries. Typically, this type of graph is more useful when we have a small number of observations. Nonetheless, it is still somewhat insightful in the context of our data with 196 observations. One advantage is that GNI p.c. is a continuous variable, and small variations are visible on the dot plot, but not on the histogram (which approximates the continuous distribution through a discrete one).

2.1.7.3 Summary statistics by region

While histograms and dot plots provide rich representation of the distributions of GNI p.c. within and between regions, they contain too much information, which makes it difficult to gain a more quantitative feel of the regional differences (for example, it is not always easy to compare distributions between regions - given the histogram above are countries in Sub-Saharan Africa typically poorer or richer than in South Asia?). We can gain more quantitative feeling of regional differences by using summary statistics (numbers that summarize key aspects of the distributions such as central tendency and spread) by region. For example, we can obtain the median and mean (measures of central tendency) and standard deviation (measuring the “spread”) of the distributions of GNI p.c. for the different regions (i.e., *conditional on region*):

```
csdata %>% group_by(Region) %>% summarize(median=median(rGNIpc), mean=mean(rGNIpc), sd=sd(rGNIpc)
```

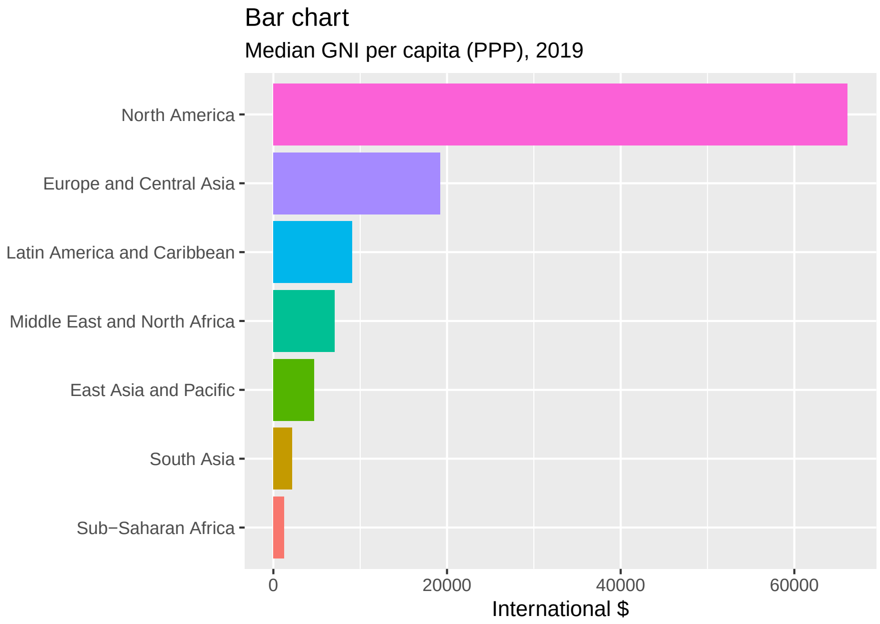
```
## # A tibble: 7 x 4
##   Region                       median  mean     sd
##   <fct>                         <dbl> <dbl>  <dbl>
## 1 Sub-Saharan Africa             1225  2344.  3117.
## 2 South Asia                     2145  3140   3057.
## 3 East Asia and Pacific          4680 13772. 17225.
## 4 Middle East and North Africa   7060 16808. 17882.
## 5 Latin America and Caribbean    9070 13463. 12001.
## 6 Europe and Central Asia       19200 28641. 25212.
## 7 North America                 66130 76670  36500.
```

The numbers should be fairly self-explanatory but in any case we can see that there are large differences in living standards across regions. In addition, we can see that distributions are more disperse the richer the region on average.

2.1.7.4 Bar charts by region

One way to graphically represent key summary statistics (such as the information from the above table) is through bar charts. For example, a bar chart representing the median GNI p.c. by regions can be obtained as follows:

```
ggplot(csdata, aes(y=rGNIpc, x=Region)) +
  geom_bar(stat='summary', fun='median', aes(fill=Region), show.legend=F) +
  labs(title="Bar chart",subtitle="Median GNI per capita (PPP), 2019") +
  xlab("") + ylab("International $") +
  coord_flip()
```

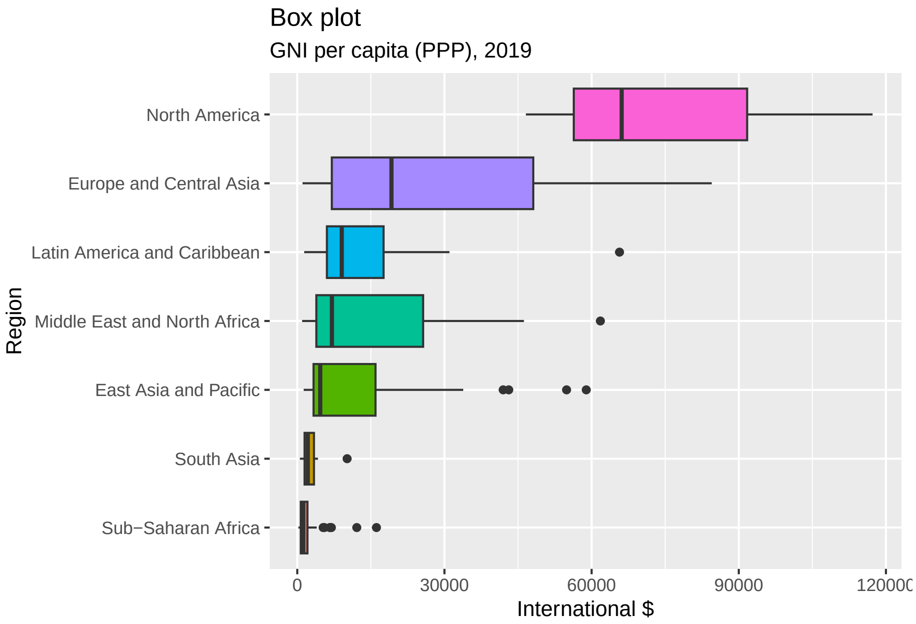


We can produce similar bar charts for the other summary statistics.

2.1.7.5 Box plots by region

A somewhat richer representation of regional differences, based on summary statistics, can be obtained by using *box and whisker plots* (also called simply *box plots*). A box plot of GNI p.c. by regions in our data can be obtained as follows:

```
ggplot(csdata, aes(x=Region, y=rGNIpc)) +
  geom_boxplot(aes(fill=Region), show.legend=F) +
  labs(title="Box plot", subtitle="GNI per capita (PPP), 2019") +
  xlab("Region") + ylab("International $") + coord_flip()
```



The vertical segments in the middle of the "boxes" correspond to the medians of the distributions by region. The boxes span the support from the 25th percentile to the 75th percentile (so 50% of the observations per region lie inside the corresponding box, and the width of the box is the IQR). The horizontal segments - referred to as *whiskers* - around the sides of the boxes are meant to represent parts of the support where observations are "relatively common". Precise statement of what this means is beyond the scope of the discussion here, but by default R plots whiskers as having length equal to $1.5 \times IQR$ or going up(down) to the maximum(minimum) observation in the sample. The points, represent individual observations (countries) lying outside of the whiskers, i.e., ones with "unusually" extreme values - these are referred to as *outliers*.

Unlike the bar chart, the box plot allows us to understand not only the central tendency of the conditional-on-region distributions, but also the dispersion, and skewness.

2.1.7.6 Categorizing countries by quintiles

Another meaningful set of representations can be obtained by categorizing countries into groups according to their ranking in the distribution of GNI p.c. While this certainly leads to some loss of information, it may still be appropriate for representing features of the data that are difficult to achieve with continuous variables, as discussed below.

First, let's create a new variable `qrank.rGNIpc` which assigns countries into a *quintile*[3] of the global GNI p.c. distribution:

[3] First quintile is countries between the 0th and 20th percentile, second between 20th and

```r
csdata <- csdata %>% arrange(rGNIpc)
csdata <- csdata %>% mutate(qrank.rGNIpc=ntile(rGNIpc,5))
csdata$qrank.rGNIpc <- as.factor(csdata$qrank.rGNIpc)
levels(csdata$qrank.rGNIpc) = c("0-20%", "20-40%", "40-60%", "60-80%", "80-100%")
```

Now we can characterize the joint distribution of region and GNI p.c. quintile in the form of a two-way table of counts:

```r
tab <- table(csdata$Region, csdata$qrank.rGNIpc)
kable(tab)
```

	0-20%	20-40%	40-60%	60-80%	80-100%
Sub-Saharan Africa	30	9	5	2	0
South Asia	3	4	0	1	0
East Asia and Pacific	2	12	4	4	6
Middle East and North Africa	1	8	2	4	5
Latin America and Caribbean	2	3	15	14	4
Europe and Central Asia	2	3	13	14	21
North America	0	0	0	0	3

To understand the meaning of the numbers, a few examples follow

- 30 of the countries in our data are in Sub-Saharan Africa and in the bottom quintile of the global GNI p.c. distribution.
- 4 of the countries in the data are in East Asia and Pacific region and are in the third quintile of the global GNI p.c. distribution.

The same information can be represented in terms of proportions:

```r
kable(prop.table(tab), digits=2)
```

	0-20%	20-40%	40-60%	60-80%	80-100%
Sub-Saharan Africa	0.15	0.05	0.03	0.01	0.00
South Asia	0.02	0.02	0.00	0.01	0.00
East Asia and Pacific	0.01	0.06	0.02	0.02	0.03
Middle East and North Africa	0.01	0.04	0.01	0.02	0.03
Latin America and Caribbean	0.01	0.02	0.08	0.07	0.02
Europe and Central Asia	0.01	0.02	0.07	0.07	0.11
North America	0.00	0.00	0.00	0.00	0.02

So, for example,

- 15% of the countries in our data are in Sub-Saharan Africa and in the bottom quintile of the global GNI p.c. distribution.
- 2% of the countries in the data are in East Asia and Pacific region and are in the third quintile of the global GNI p.c. distribution.

40th percentile and so forth.

If we normalize the values in the table so that rows sum to 1, we get a table on the distribution of GNI p.c. quintiles *conditional on regions*:

```r
kable(prop.table(tab,1), digits=2)
```

	0-20%	20-40%	40-60%	60-80%	80-100%
Sub-Saharan Africa	0.65	0.20	0.11	0.04	0.00
South Asia	0.38	0.50	0.00	0.12	0.00
East Asia and Pacific	0.07	0.43	0.14	0.14	0.21
Middle East and North Africa	0.05	0.40	0.10	0.20	0.25
Latin America and Caribbean	0.05	0.08	0.39	0.37	0.11
Europe and Central Asia	0.04	0.06	0.25	0.26	0.40
North America	0.00	0.00	0.00	0.00	1.00

So for example, 65% of countries in Sub-Saharan Africa are in the bottom fifth of the global GNI p.c. distribution; additional 20% are in the second quintile, etc. On the other hand, 100% of countries in North America region are in the top fifth of the global GNI p.c. distribution.

If instead we normalize the values in the table so that columns sum to 1, we get a table on the distribution of regions *conditional on quintile of GNI p.c.*:

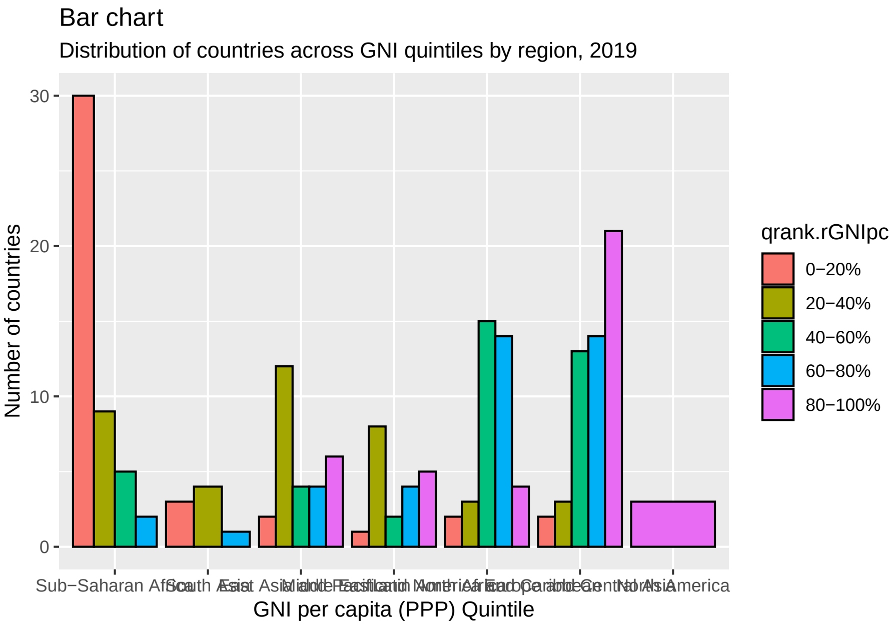
```r
kable(prop.table(tab,2))
```

	0-20%	20-40%	40-60%	60-80%	80-100%
Sub-Saharan Africa	0.750	0.2307692	0.1282051	0.0512821	0.0000000
South Asia	0.075	0.1025641	0.0000000	0.0256410	0.0000000
East Asia and Pacific	0.050	0.3076923	0.1025641	0.1025641	0.1538462
Middle East and North Africa	0.025	0.2051282	0.0512821	0.1025641	0.1282051
Latin America and Caribbean	0.050	0.0769231	0.3846154	0.3589744	0.1025641
Europe and Central Asia	0.050	0.0769231	0.3333333	0.3589744	0.5384615
North America	0.000	0.0000000	0.0000000	0.0000000	0.0769231

So for example, 75% of world's poorest countries (i.e., countries in the bottom 20 percent) are in Sub-Saharan Africa. On the other hand, 8% of the world's richest countries (in terms of quintile of global distribution) are in North America.

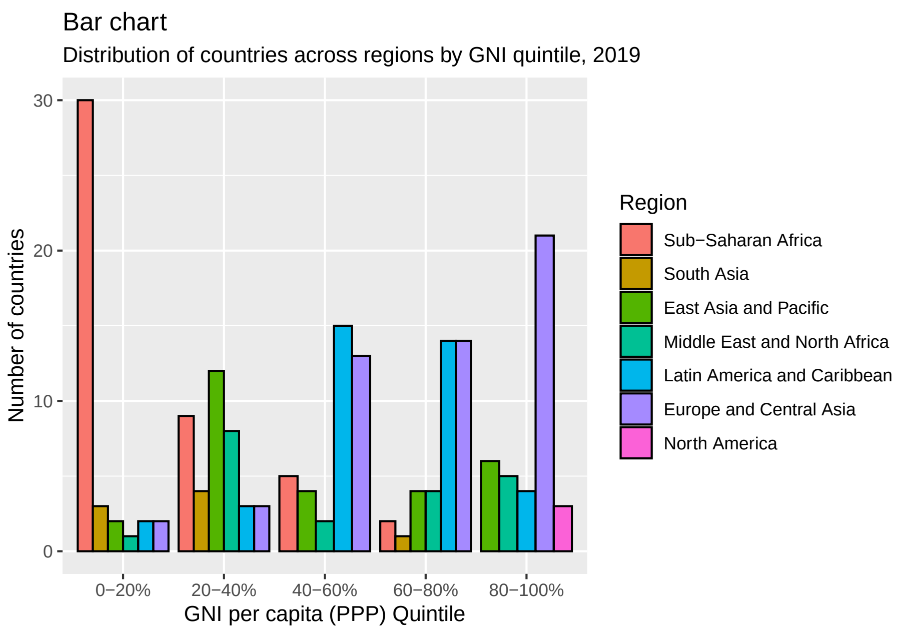
The information from the two-way tables can be equivalently represented in bar charts. For example, the distribution of GNI p.c. quintiles by region (i.e., conditional on regions) can be represented as follows:

```r
ggplot(csdata, aes(x=Region, fill=qrank.rGNIpc)) +
  geom_bar(position='dodge', color="black") +
  labs(title="Bar chart",subtitle="Distribution of countries across GNI quintiles by region, 2019
  xlab("GNI per capita (PPP) Quintile") + ylab("Number of countries")
```



The distribution of regions across GNI p.c. quintiles can be represented as follows:

```
ggplot(csdata, aes(fill=Region, x=qrank.rGNIpc), na.rm = T) +
  geom_bar(position='dodge', color="black") +
  labs(title="Bar chart",subtitle="Distribution of countries across regions by GNI qui
  xlab("GNI per capita (PPP) Quintile") + ylab("Number of countries")
```



Following this let's clean the R environment

```
rm(csdata, tab)
```

2.1.8 Representing relationships between continuous variables

In the data we use so far the only continuous variable is GNI per capita (PPP). However, we typically work with richer datasets and are often interested in representing the relationship between several continuous variables.

To provide some examples of this, we load the dataset from sheet `sc2019_multi` in the `data_ch1.xlsx` Excel file:

```
csdata.all <- read_excel("data_ch1.xlsx", sheet="cs2019_multi")
head(csdata.all)
```

```
## # A tibble: 6 x 7
##   Country        Code   Year Region                 rGNIpc LifeExpectancy  CO2pc
##   <chr>          <chr> <dbl> <chr>                   <dbl>          <dbl>  <dbl>
## 1 Afghanistan    AFG    2019 South Asia                530           63.6  0.298
## 2 Albania        ALB    2019 Europe and Central As~   5230           79.3  1.75
## 3 Algeria        DZA    2019 Middle East and North~   4050           76.5  3.99
## 4 American Samoa ASM    2019 East Asia and Pacific      NA           NA   NA
## 5 Andorra        AND    2019 Europe and Central As~  46530           NA    6.29
## 6 Angola         AGO    2019 Sub-Saharan Africa       2040           62.4  0.754
```

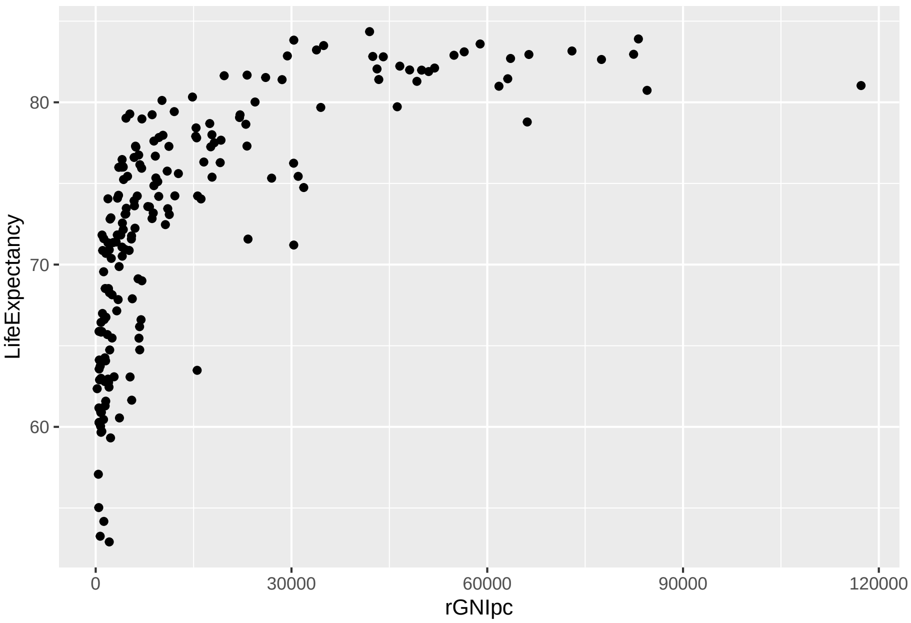
The new dataframe contains the same cross section of countries in 2019, but in addition to GNI per capita (PPP), includes two additional variables from the World Bank's WMD database for 2019:

- `LifeExpectancy`: the average life expectancy at birth for each country
- `CO2pc`: the CO2 emissions per capita for each country.

2.1.8.1 GNI per capita and Life Expenctancy

Let's start by investigating how GNI per capita associates with life expectancy. A convenient way for representing the relationship between two continuous variables is through a scatterplot. In this case this can be a graph with GNI p.c. on the horizontal axis and life expectancy on the vertical axis, with each country's combination of the two variables plotted by a point at the corresponding coordinates:

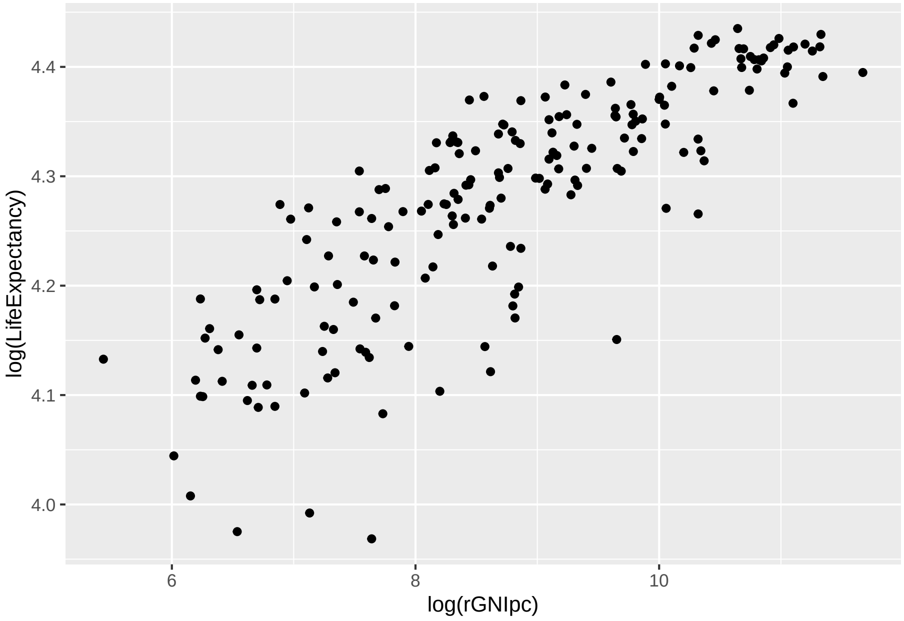
```
ggplot(csdata.all,aes(x=rGNIpc, y=LifeExpectancy)) +
  geom_point()
```



As we can see, there is a clear positive relationship between GNI per capita and life expectancy.

Given the scale of measurement on both axes, the relationship appears to be non-linear. In such cases it is worth checking how changes in the scale of measurement affect the shape of the relationship. To explore this, let's plot the natural log of GNI per capita (PPP) against the natural log of life expectancy:

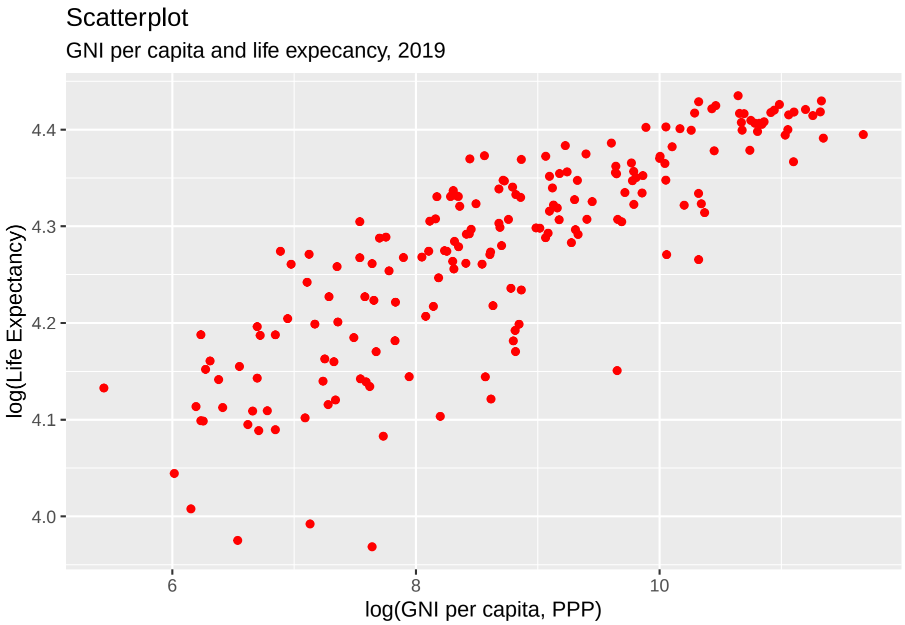
```
ggplot(csdata.all,aes(x=log(rGNIpc), y=log(LifeExpectancy))) +
  geom_point()
```



As we can see, on logarithmic scales, the relationship appears approximately linear. This indicates that proportional increases in GNI per capita associate with proportional increases in life expectancy.

Using the usual syntax we can customize the graph further:

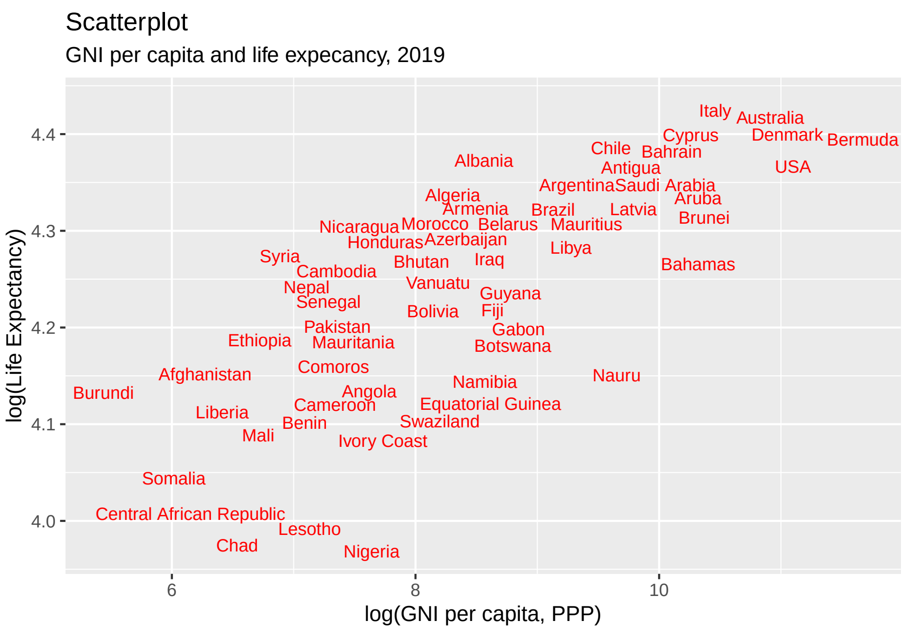
```
ggplot(csdata.all,aes(x=log(rGNIpc), y=log(LifeExpectancy))) +
  geom_point(col="red") + xlab("log(GNI per capita, PPP)") +
  ylab("log(Life Expectancy)") + labs(title="Scatterplot", subtitle="GNI per capita and life expe
```


Scatterplot

GNI per capita and life expecancy, 2019

Alternatively, rather than plotting points at the corresponding coordinates, we can instead plot as text the name of the country:

```
ggplot(csdata.all,aes(x=log(rGNIpc), y=log(LifeExpectancy),label=Country)) +
  geom_text(col="red", check_overlap=TRUE, size=3) + xlab("log(GNI per capita, PPP)") +
  ylab("log(Life Expectancy)") + labs(title="Scatterplot", subtitle="GNI per capita and
```

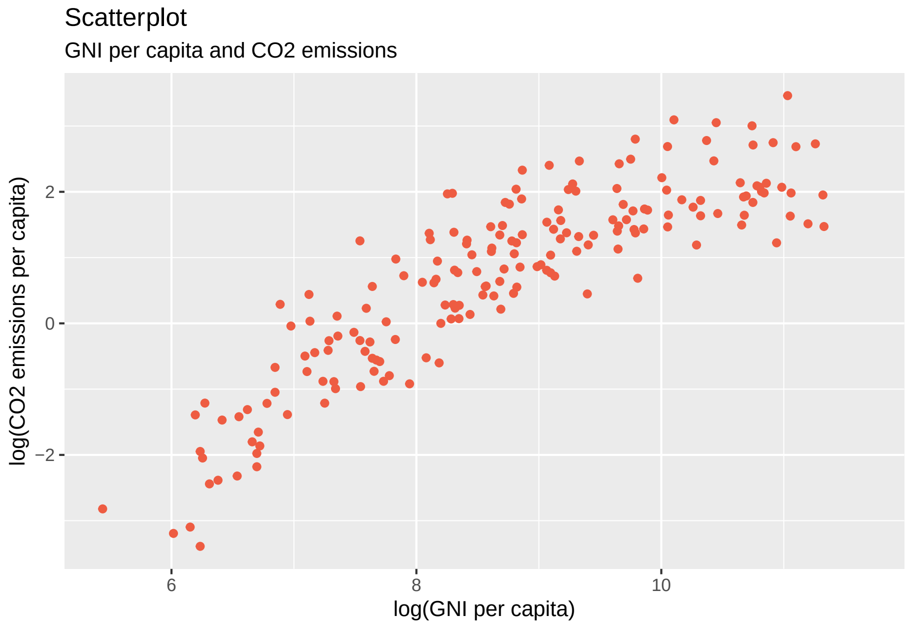


In summary, we observe that countries with higher GNI per capita tend to also have higher life expectancy. It should be noted, however, that association does not imply causal links. Based on what we observe, we are not able to conclude if higher GNI per capita causes higher life expectancy, or high life expectancy causes high GNI per capita, or if something else varies systematically over countries causing some to have high income and high life expectancy, while others to have low income and low life expectancy.

2.1.8.2 GNI per capita and CO2 emissions per capita

In a similar way we can visualize the association between GNI per capita (PPP) and CO2 emissions per capita:

```
ggplot(csdata.all,aes(x=log(rGNIpc), y=log(CO2pc), label=Country)) +
  geom_point(col="tomato2") + xlab("log(GNI per capita)") +
  ylab("log(CO2 emissions per capita)") + labs(title="Scatterplot", subtitle="GNI per capita and
```



The association between the two seems to be even more pronounced. Again we can conclude that countries with high GNI per capita (PPP) tend to also have high CO2 emissions per capita.

This completes the discussion of cross-sectional data. Next we will turn attention to visualizing time series. Before that let's clean the R environment

```
rm(csdata.all)
```

2.2 Representing time-series data: Real GDP per capita in UK over time

This section provides examples of representing time-series data.

Up until now we have used data on GNI per capita (PPP) from the World Bank's World Development Indicators database. The cross-sectional dataset for 2019 that was used in the previous section, was obtained as from the full panel dataset by only keeping observations for 2019. Similarly, one way in which we can obtain a time-series dataset, is as a subset of the full panel dataset for an individual country only.

While this is certainly feasible (and all the below codes can be replicated by using such a dataset) we will choose to use a slightly different dataset instead - namely, a quarterly time-series of UK's real GDP per capita (rather than annual GNI per capita (PPP)) obtained from the Office for National Statistics here. We do so for several reasons:

- Atlas-method PPP-adjusted gross national income per capita is best-suited for comparing living standards between countries, but not so well-suited for understanding the dynamic change of real income within a country. Real GDP per capita is better suited for the latter.
- GNI per capita data from the World Bank's WDI is only available annually. Data on real GDP per capita for UK is available at higher frequencies - in partiular, we will use quarterly data - which allows for better understanding of business cyclical fluctuations.

The data from ONS has been saved in the `uk_gdppc_qr` sheet of the `data_ch1.xlsx` Excel file. Start by loading the data and printing the first few rows:

```r
ukgdp <- read_excel("data_ch1.xlsx", sheet="uk_gdppc_qr")
ukgdp$date <- as.Date(ukgdp$date)

head(ukgdp)
```

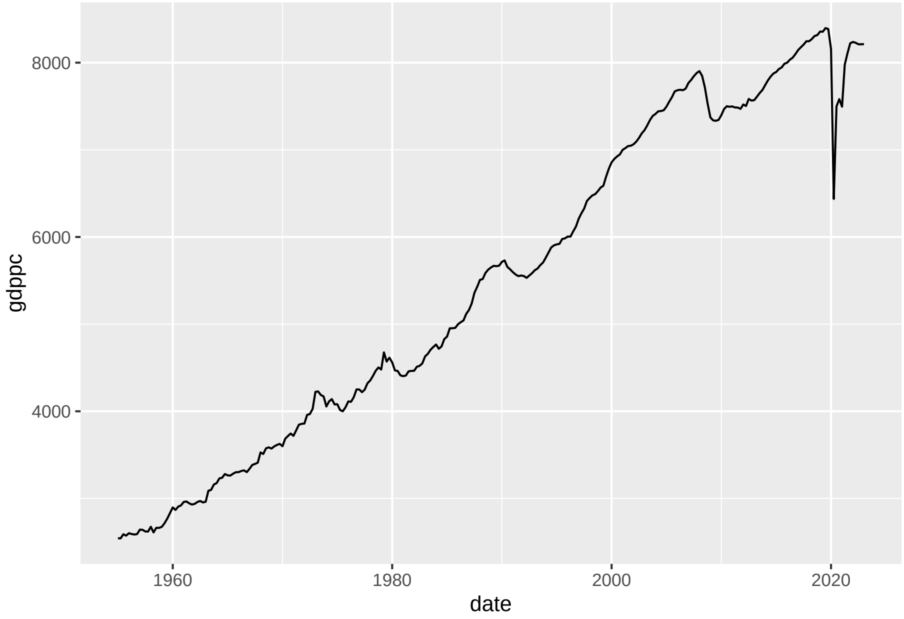
```
## # A tibble: 6 x 2
##   date       gdppc
##   <date>     <dbl>
## 1 1955-01-01  2543
## 2 1955-04-01  2542
## 3 1955-07-01  2589
## 4 1955-10-01  2573
## 5 1956-01-01  2601
## 6 1956-04-01  2592
```

The dataset includes quarterly observations on real GDP per capita from the begining of 1955 until the begining of 2023. There are only two variables: `date` which lists the quarter of the observation, and `gdppc` which gives the value of real GDP per capita measured in 2022 pounds.

2.2.1 Time-series line plots

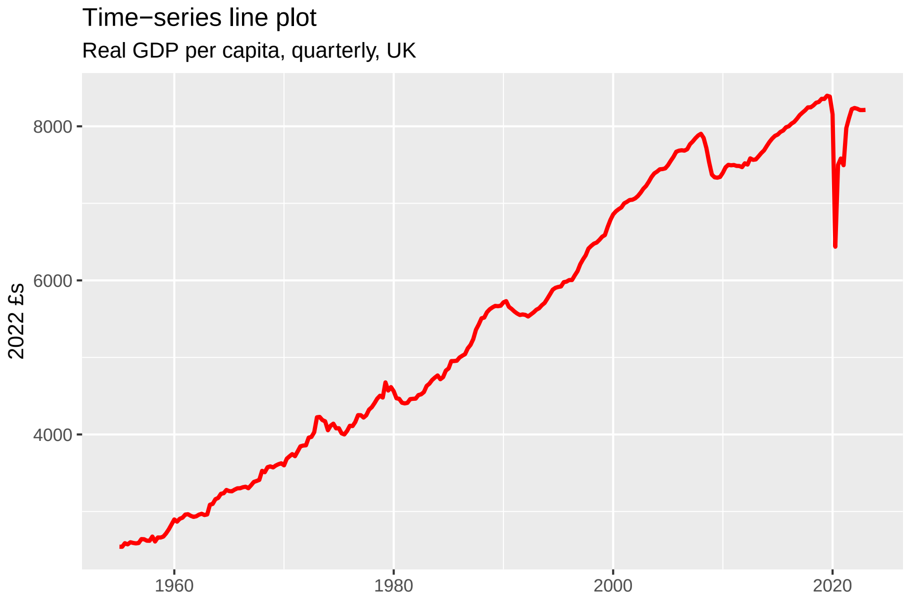
A common way to represent the behaviour of a time-series variable is through a time-series line plot. This is a graph that plots the value of the variable (in our case real GDP per capita) against time

```r
ggplot(ukgdp, aes(x=date, y=gdppc)) +
  geom_line()
```



As usual the graph can be customized. The code below sets a different color and width of the line and adds labels and titles.

```
ggplot(ukgdp, aes(x=date, y=gdppc)) +
  geom_line(col="red", size=1) +
  labs(title="Time-series line plot", subtitle = "Real GDP per capita, quarterly, UK")
  xlab("") + ylab("2022 £s")
```



Over time real GDP per capita has exhibited an overall positive trend. In terms of 2022 GBP (henceforth £), UK's average quarterly income has more than tripled from £2543 in the first quarter of 1955 to £8213 in the first quarter of 2023. We can also see that this overall growth has not been smooth - GDP fluctuates around trend over sequential periods of *expansions* and *recessions* - something known as the **business cycle**. For example, in the second quarter of 2020 (the through of the 2020 pandemic recession), per capita GDP fell to £6437 - a level last seen in 1998 - from a peak of £8386 two quarters ago.

2.2.2 Time-series line plots of the log transformation

When representing the time path of a growing series over time it is often insightful to plot on a logarithmic scale (i.e., to plot the natural log of the series, rather than the series itself, against time). To see why this is the case, suppose that Y_t denotes GDP per capita, Y, at quarter t. When we plot Y_t against t, the slope of the "curve" at any particular point is

$$\frac{\Delta Y_t}{\Delta t} \equiv \frac{Y_{t+1} - Y_t}{(t+1) - t} = Y_{t+1} - Y_t = \Delta Y_t$$

which is measured in the same units of measurement as the Y_t (i.e., pounds per quarter). Hence, if GDP were to grow at a constant positive growth rate over time, its time path would describe an increasing and convex schedule over time.

Suppose that instead we plot the natural logarithm of GDP per capita against time. The slope of this curve at any particular point is instead

$$\frac{\Delta log(Y_t)}{\Delta t} = log(Y_{t+1}) - log(Y_t) = log\left(\frac{Y_{t+1}}{Y_t}\right) = log\left(1 + \frac{Y_{t+1} - Y_t}{Y_t}\right)$$

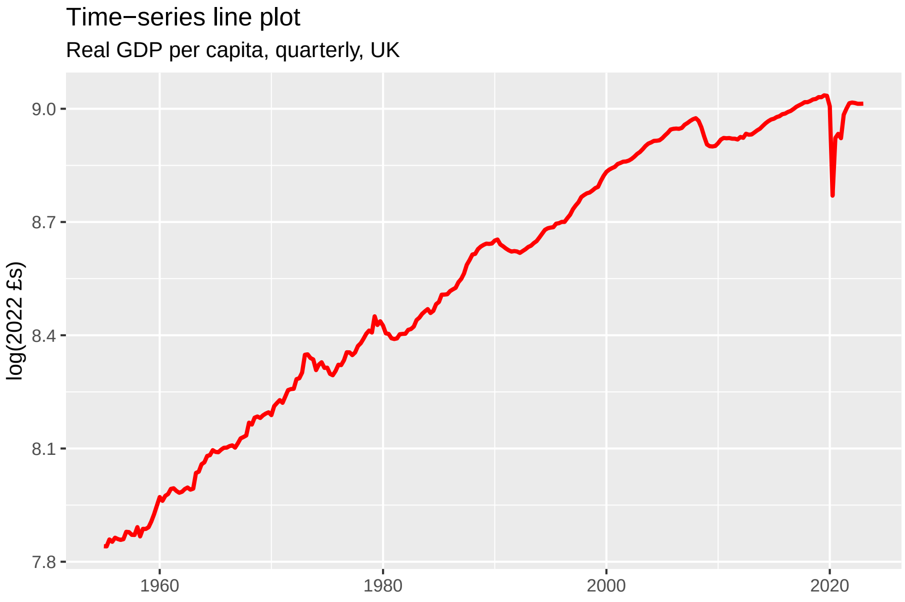
Now using the approximation $log(1+g) \approx g$ when g is close to zero[4] we obtain

$$\frac{\Delta log(Y_t)}{\Delta t} \approx \frac{Y_{t+1} - Y_t}{Y_t}$$

so the slope of $log(Y_t)$ in the (t, Y_t) plane is approximately equal to the growth rate of Y_t. In the context of our data this equals the quarterly growth rate (percentage change quarter-on-quarter) of GDP. Hence, if GDP were to grow at a constant positive growth rate over time, the time path of its natural logarithm would describe a linear schedule over time. This allows us to quickly spot if there are systematic changes in growth rates over periods of time.

For example, plotting the log of UK's GDP per capita over time

```
ggplot(ukgdp, aes(x=date, y=log(gdppc))) +
  geom_line(col="red" ,size=1) +
  labs(title="Time-series line plot", subtitle = "Real GDP per capita, quarterly, UK")
  xlab("") + ylab("log(2022 £s)")
```



We observe that the time path described by the log of real GDP per capita is "approximately" linear from 1955 to 2007, but afterwards shifts to a lower and

[4] This is an instance of a first-order Taylor approximation around $g = 0$.

flatter linear path. This indicates that real GDP growth rates were “approximately constant” from 1955 until the onset of the Great Recession in 2007, but afterwards average growth rates declined. This slowdown of GDP growth rates is one of the manifestations of a phenomenon known as *UK’s productivity slowdown* (although similar patterns have been observed in many other economies for the same time period).

One naive but possibly insightful way to visualize this slowdown is to imagine that after 2007 GDP per capita experienced the same growth rate as it did in the prior decades, and to compare the actual series to a *counterfactual* series for real GDP per capita that would have prevailed under the previous growth rates. To do this

- first, find the *counterfactual* constant average quarterly growth rate from 1955 to 2007[5]

```
avegr.preGR<-(ukgdp$gdppc[ukgdp$date=="2007-01-01"]/ukgdp$gdppc[ukgdp$date=="1955-01-01"])^(1/((2
```

- next, construct a new variable `cfgdppc` which equals to the actual value of GDP per capita in the first quarter of 1955, and then increases by the amount of the average growth rate identified above each subsequent quarter. One (inefficient but transparent) way to do this is using a loop:

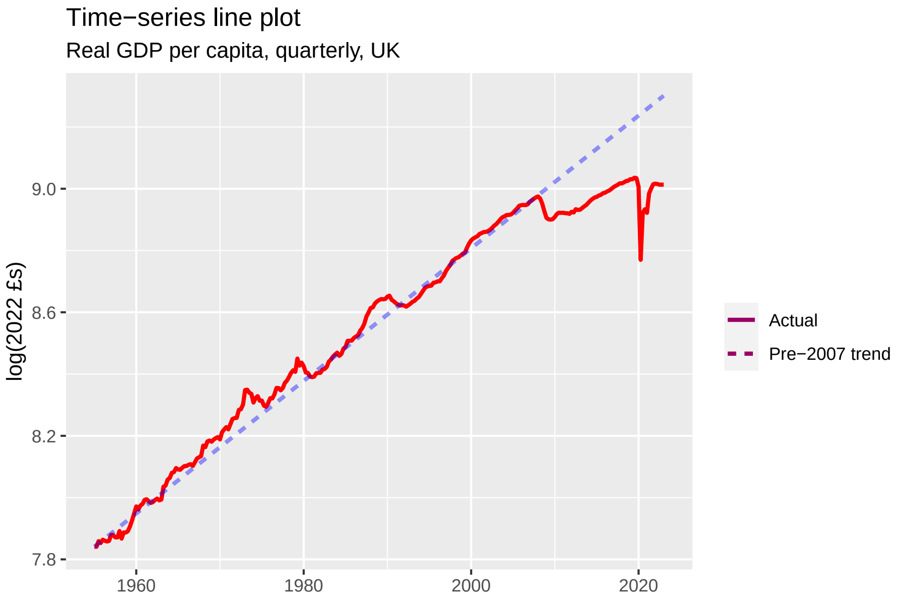
```
ukgdp$cfgdppc <- ukgdp$gdppc[ukgdp$date=="1955-01-01"]

for (i in 1:length(ukgdp$date)){
  ukgdp$cfgdppc[i]=ukgdp$cfgdppc[1]*(1+avegr.preGR)^(i-1)
}
```

- finally, plot both the natural logs of the actual and counterfactual GDP per capita series against time:

```
ggplot(ukgdp) +
  geom_line(aes(x=date, y=log(gdppc), linetype="Actual"),size=1, col="red") +
  geom_line(aes(x=date, y=log(cfgdppc), linetype="Pre-2007 trend"), alpha=0.4, col="blue", size=1
  labs(title="Time-series line plot", subtitle = "Real GDP per capita, quarterly, UK") +
  xlab("") + ylab("log(2022 £s)") + labs(linetype="")
```

[5] Let Y_t denote quarterly GDP per capita at quarter t. Suppose that Y_t grows at a constant growth rate of g per quarter. Then

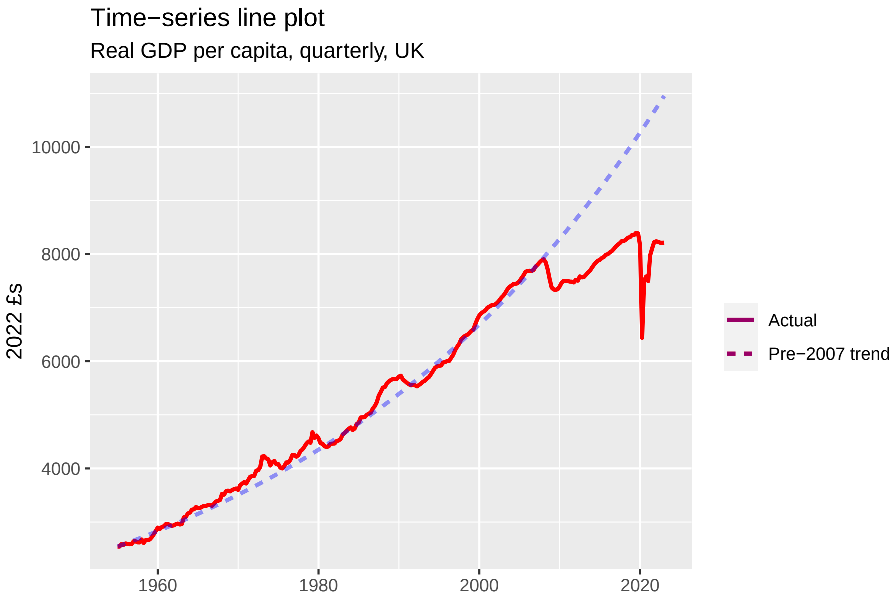
$$Y_{2007,Q1} = Y_{1955,Q1}(1+g)^{(2007-1955)\times 4(=\text{number of quarters from 1955,Q1 to 2007,Q1})}$$



noting that (by construction) the log of the counterfactual series evolves linearly over time;

- or alternatively, in levels:

```
ggplot(ukgdp) +
  geom_line(aes(x=date, y=gdppc, linetype="Actual"), size=1, col="red") +
  geom_line(aes(x=date, y=cfgdppc, linetype="Pre-2007 trend"), alpha=0.4, size=1, col="
  labs(title="Time-series line plot", subtitle = "Real GDP per capita, quarterly, UK")
  xlab("") + ylab("2022 £s") + labs(linetype="")
```



noting that (by construction) the counterfactual series evolves as a geometric series over time;

As we can see the actual series tracks closely the counterfactual one prior to 2007 (indicating approximately constant growth rates over the period) but then diverges sharply from it during the Great Recession and furthermore continues growing at a slower average rate than before.

One aspect in which time series charts in logs represent information more intuitively than in levels comes from the property, previously discussed, that the difference of the logs of two numbers is approximately equal to the proportional difference between the numbers in levels. More specifically, given two numbers, A and B

$$\log(A) - \log(B) = \log\left(\frac{A}{B}\right) = \log\left(\frac{A-B}{B} + 1\right) \approx \frac{A-B}{B}$$

One implication of this is that, on a plot in logs, the vertical distance between subsequent observations is approximately equal to the growth rate. We have already mentioned this, and will return to it later. Another example of how this can be useful can be stated in the context of the above example. As we can see, following the divergence between the actual and counterfactual series around the Great Recession, as of the last quarter of 2019 (just before the Covid pandemic) the difference in logs of the two series was about 0.2[6]. This implies

[6] In fact, looking closely at the data, the log of GDP per capita in the last quarter of 2019 was 9.03419 in the actual series and 9.231319 in the counterfactual series - the difference in

that, up to an approximation, UK economy was operating at about 20% lower GDP per capita in 2019 relative to what we would have expected if the pre-Great-Recession trend had persisted. This is one (possibly naive) way to think of the long-term cost of the productivity slowdown. The point here is that this is clearly visible on the chart in logs without further calculations necessary. On the other hand, what we see in the chart in levels is that the level is approximately 1800 pounds less than the trend. While both numbers are informative, the point is that different representations (even of the same variable) make it easier to focus on different aspects.

It should be noted that this exercise should be viewed with some skepticism (for a number of reasons but especially) due to the relatively naive way of fitting a trend. This being said there is no uncontroversial way of estimating trends. Furthermore, much more general and sophisticated empirical and theoretical approaches detect a systematic decline in productivity growth in the last decades. Understanding the reasons for the productivity growth decline is a hot topic of debate in economics, and you will learn about this during your studies, but further discussion is beyond the scope of this book.

2.2.3 Recessions

As already discussed GDP per capita fluctuates systematically over the business cycle - in some periods it is above trend and in others below trend. In a general sense, periods of significant decline below trend are referred to as recessions. While precise technical definitions of a recession vary across countries, in the UK recessions are technically defined as a period of at least two consecutive quarters of negative GDP growth. When plotting macroeconomic time-series it is common to explicitly visualize periods defined as recessions, and we show how to do this here.

First, the sheet `recs` of the `data_ch1.xlsx` Excel file contains data on all recessions in UK from 1955 to 2023 according to the technical definition.

```r
recs <- read_excel("data_ch1.xlsx", sheet="recs")
recs$rec_st <- as.Date(recs$rec_st)
recs$rec_en <- as.Date(recs$rec_en)

recs
```

```
## # A tibble: 8 x 2
##   rec_st     rec_en
##   <date>     <date>
## 1 1956-04-01 1956-10-01
```

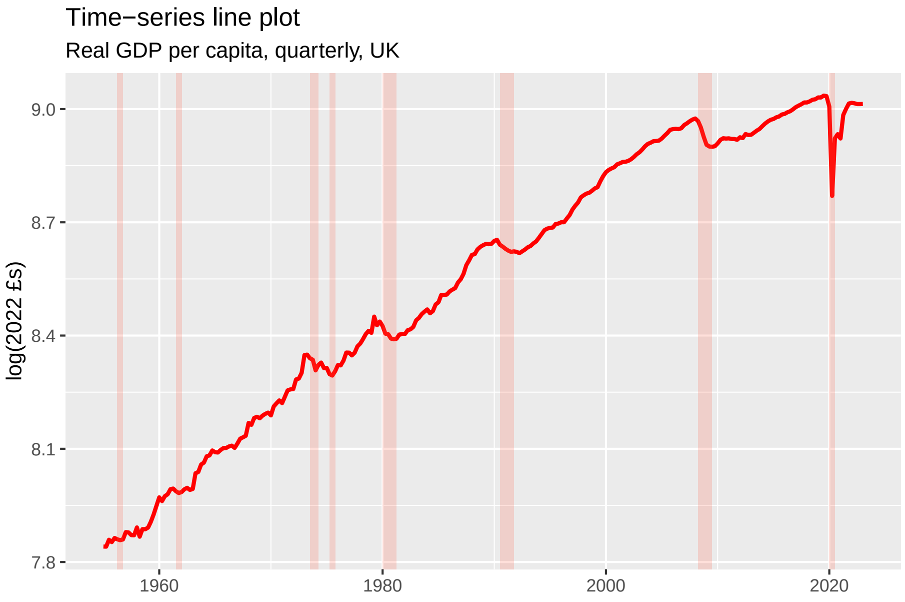
logs was then about 0.197 (9.231319-9.03419). The corresponding quantities in levels were 8386 for the actual, and 10212 for the counterfactual series. The actual difference in levels was 1826 pounds (10212-8386). The actual percentage difference was 0.217 (1826/8386). You can see that the difference in logs (0.197) is close to the actual percentage difference (0.217) even though 0.217 is not so close to zero.

```
## 2 1961-07-01 1962-01-01
## 3 1973-07-01 1974-04-01
## 4 1975-04-01 1975-10-01
## 5 1980-01-01 1981-04-01
## 6 1990-07-01 1991-10-01
## 7 2008-04-01 2009-07-01
## 8 2020-01-01 2020-07-01
```

The two variables `rec_st` and `rec_en` list the starting and ending quarter of each of the recessions.

Now, by plotting the log og real GDP per capita series, and adding colored rectangles between corresponding starting and ending dates of recessions we obtain the following plot:

```
ggplot(ukgdp) +
  geom_line(aes(x=date, y=log(gdppc)), col="red", size=1) +
  labs(title="Time-series line plot", subtitle = "Real GDP per capita, quarterly, UK") +
  xlab("") + ylab("log(2022 £s)") +
  geom_rect(data=recs,aes(xmin=rec_st, xmax=rec_en, ymin=-Inf, ymax=+Inf), fill='tomato', alpha=0
```



As we can see, there have been 8 recessions over the period. Two of the most notable ones are the two most recent - the Great Recession of 2007-2009 and the Covid-19 recession. We can also see that while the Covid-19 recession was associated with the largest quarterly decline in GDP per capita, it was also the shortest among all according to the technical definition.

2.2.4 Growth rates

On many occasions one might be interested in analyzing the dynamic behaviour of the growth rate of GDP per capita, rather than its level.

A common way to measure GDP growth rates with quarterly data is in terms of annual percentage changes relative to a year ago[7]. Let Y_t denote the value of GDP per capita in quarter t. Then its annual growth rate relative to year (4 quarters) ago is defined as

$$\frac{Y_t - Y_{t-4}}{Y_{t-4}}$$

We can create a new variable gdppc.gr based on the above:

```
ukgdp <- ukgdp %>% mutate(gdppc.gr = (gdppc - lag(gdppc,4))/lag(gdppc,4))
```

Inspecting the data

```
head(ukgdp)
```

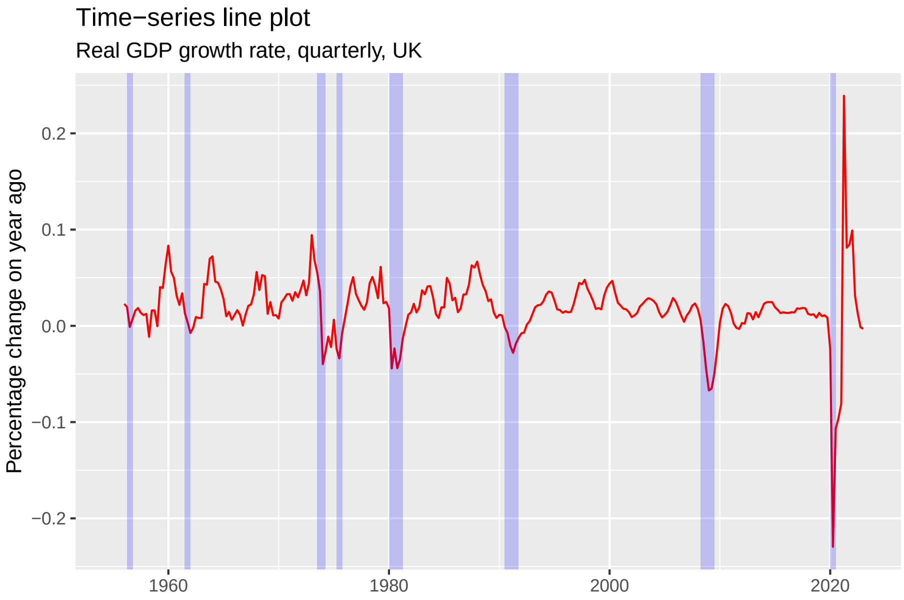
```
## # A tibble: 6 x 4
##   date       gdppc cfgdppc gdppc.gr
##   <date>     <dbl>   <dbl>    <dbl>
## 1 1955-01-01  2543   2543   NA
## 2 1955-04-01  2542   2557.  NA
## 3 1955-07-01  2589   2570.  NA
## 4 1955-10-01  2573   2584.  NA
## 5 1956-01-01  2601   2598.   0.0228
## 6 1956-04-01  2592   2612.   0.0197
```

note that observations on the growth rate are missing in the first 4 quarters of the sample, as the levels from 4 quarters ago are not available.

Plotting growth rates against time yields the following

```
ggplot(ukgdp) +
  geom_line(aes(x=date, y=gdppc.gr), col="red") +
  labs(title="Time-series line plot", subtitle = "Real GDP growth rate, quarterly, UK"
  xlab("") + ylab("Percentage change on year ago") +
  geom_rect(data=recs,aes(xmin=rec_st, xmax=rec_en, ymin=-Inf, ymax=+Inf), fill='blue'
```

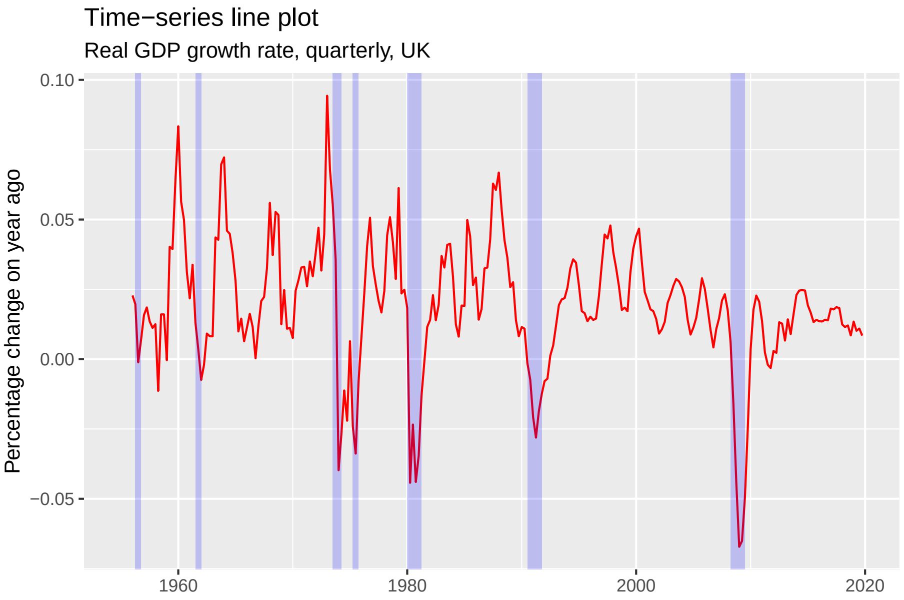
[7] It is also possible to measure it in terms of percentage changes relative to the previous quarter - this would correspond to the quarterly growth rate, which will be on a different scale given the differing lengths of time periods. It is also possible to measure it in terms of "annualized" quarterly growth rates.



While this plot is effectively based on the same information as as the plots of the level of GDP per capita above (in the sense that data on levels can be easily transformed into data on percentage changes and vice versa), given the scale of measurement it draws attention different aspects of the series. For example, now we can readily see that in the second quarter of the Covid-19 recession, quarterly GDP per capita had dropped by more than 20 percent relative to a year ago.

The size of the quarterly fluctuations surrounding the Covid-19 recessions makes it somewhat difficult to visualize what is happening prior to it. If we replicate the same graph only until the last quarter of 2019

```
ggplot(subset(ukgdp,date<"2020-01-01")) +
  geom_line(aes(x=date, y=gdppc.gr), col="red") +
  labs(title="Time-series line plot", subtitle = "Real GDP growth rate, quarterly, UK") +
  xlab("") + ylab("Percentage change on year ago") +
  geom_rect(data=recs[1:7,],aes(xmin=rec_st, xmax=rec_en, ymin=-Inf, ymax=+Inf), fill='blue', alp
```



we can observe more clearly the dynamics of growth rates from 1955 to 2020.

One thing that we can see (perhaps not very clearly) is that while there is no clear monotonic trend (as for GDP per capita in levels) typical growth rates after year 2000 tend to be lower than before - this is again the productivity slowdown discussed above.

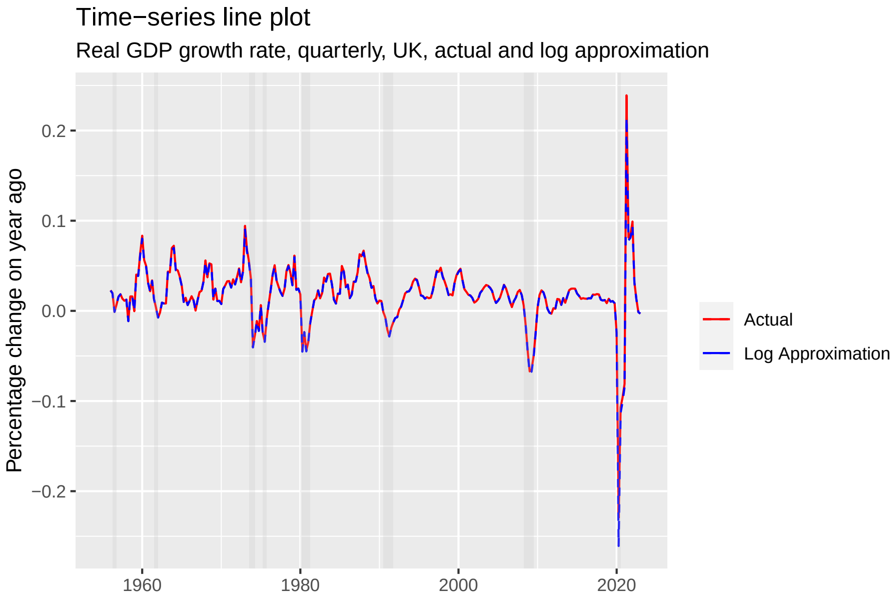
Another thing we can observe is that in some sense growth rates after the 1990s tend to exhibit smaller volatility relative to before. While the Great Recession is a notable exception, it can be noticed that even in periods of expansion, fluctuations were more volatile before some time in the 1990s. This empirical observation is often referred to as the Great Moderation (although, traditionally this term has been used to describe the period from 1990s to the Great Recession, but as we can see volatility has been relatively low in the next decade as well).

As a side note, recall that the earlier section explained that differences in logs can be used to approximate growth rates. To investigate how good this approximation is in the context of the current discussion, we can compare a series of actual growth rates to a series of approximate growth rates constructed as difference in logs. First, construct growth rates relative to same quarter a year ago and allocate in a new variable `gdppc.gr.logs`:

```
ukgdp <- ukgdp %>% mutate(gdppc.gr.logs = log(gdppc) - lag(log(gdppc),4))
```

Then plot both the actual and approximate series of growth rates against time

```
ggplot(ukgdp) +
  geom_line(aes(x=date, y=gdppc.gr, col="Actual"), lty="solid") +
  geom_line(aes(x=date, y=gdppc.gr.logs, col="Log Approximation"), lty="dashed") +
  labs(title="Time-series line plot", subtitle = "Real GDP growth rate, quarterly, UK, actual and
  xlab("") + ylab("Percentage change on year ago")  +
  geom_rect(data=recs,aes(xmin=rec_st, xmax=rec_en, ymin=-Inf, ymax=+Inf), fill='grey', alpha=0.2
```



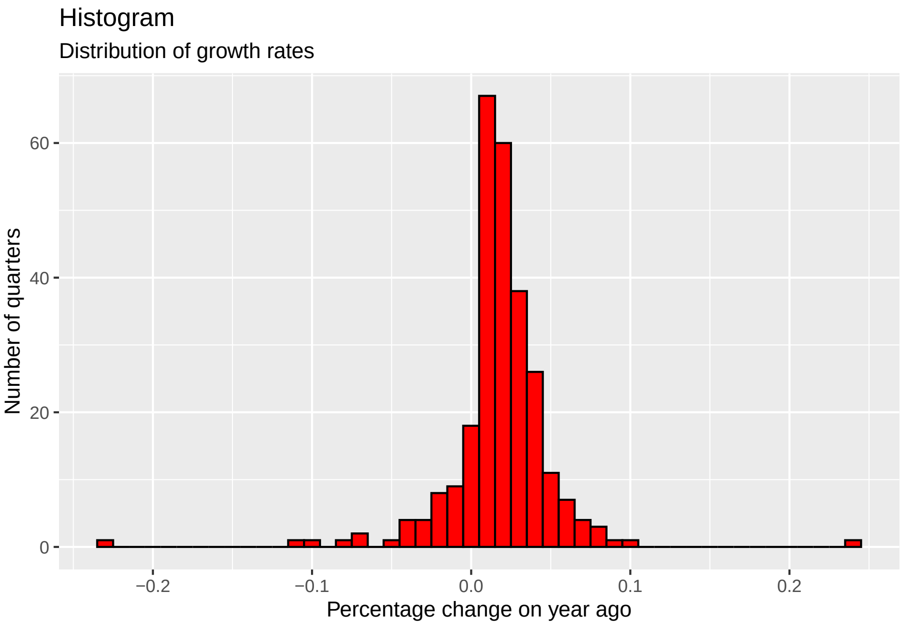
It can be clearly seen that the two series are virtually indistinguishable on the graph for most of the time window. The only clearly visible difference seems to occur in the Covid recession where the two series seem slightly different in the quarters of largest drop and recovery, but even this is barely noticeable. Given that the quality of the approximation depends on the actual growth rate being small in absolute value, it is not surprising that the approximation performs worse exactly at times of extreme changes - however, even there it seems quite close to the actual. Therefore, given the typical range of GDP growth rates it is very common to approximate growth rates by a difference in logs (and in some contexts this is also useful for analytical reasons, but this is beyond the scope of the current discussion).

2.2.5 Distributions again

While this may be obvious, it is worth noting that meaningful representations of time series do not need to involve plotting series over time. In fact, depending on the context any of the tools we have seen in representing cross-sectional data might be useful for illustrating relevant features of time series data.

For example, it is clear that while we can plot a histogram of UK's quarterly GDP per capita time series, this will be a meaningless (you can try yourself and reflect on what you see). However, a histogram of the GDP per capita growth rates time series is actually informative:

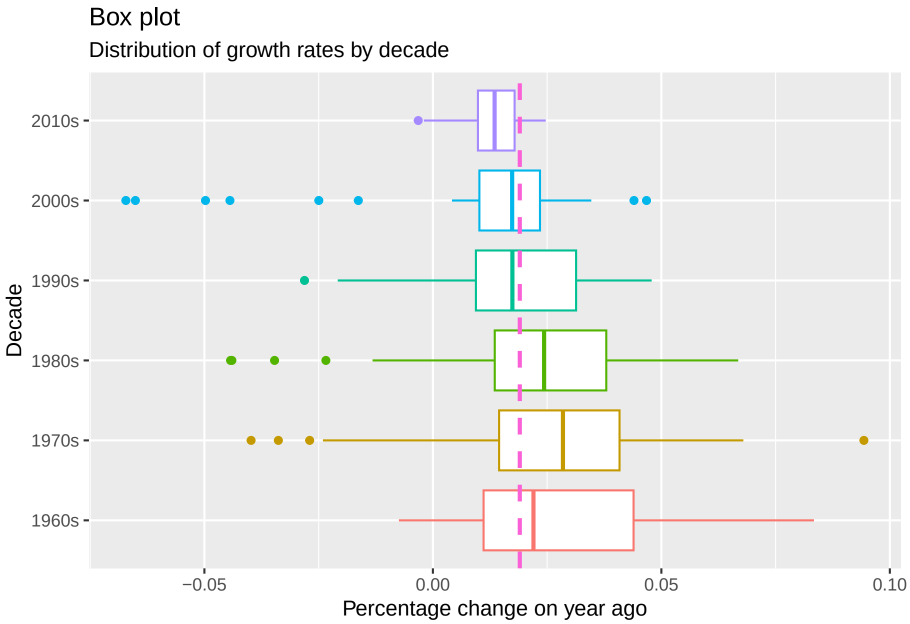
```r
ggplot(ukgdp, aes(x=gdppc.gr)) +
  geom_histogram(col="black", fill="red", binwidth=0.01)+
  labs(title="Histogram", subtitle = "Distribution of growth rates") +
  ylab("Number of quarters") + xlab("Percentage change on year ago")
```



We can see that during the sample period, annual growth rates of around 1%-2% are most common, growth rates slightwly above or below this are relatively common, larger deviations are increasingly uncommon. It can also be noted that the two most extreme observations (which by inspection occur at the onset and recovery from the Covid-19 recession) appear as extreme outliers based on what one might have expected before.

As another example, recall that we have already observed a slowdown of growth over time, as well as a decrease in volatility of fluctuations over time. One way to explore this in terms of summary statistics is by plotting box plots for growth rates by decade. The code below creates a variable for decade and plots box plots for the decades from 1960s to 2010s (while we observe some data from 1950s and 2020s we have smaller number of observations than in the other decades - including some very extreme, such as the Covid pandemic, that can drive sample averages - so we omit them)

```
ukgdp$year <- as.numeric(format(ukgdp$date, format="%Y"))
ukgdp <- ukgdp %>% mutate(decade = floor(year / 10) * 10)
ukgdp$decade <- as.factor(ukgdp$decade)
levels(ukgdp$decade) <- c("1950s", "1960s", "1970s", "1980s", "1990s", "2000s", "2010s", "2020s")

ukgdp.trunc <- subset(ukgdp,year<2020&year>1959)

ggplot(ukgdp.trunc, aes(x=decade)) +
  geom_boxplot(aes(y=gdppc.gr, col=decade)) +
  coord_flip() +
  geom_hline(aes(yintercept=mean(gdppc.gr, na.rm=TRUE), col="Average"), linetype="dashed", size=1
  labs(title="Box plot", subtitle = "Distribution of growth rates by decade") +
  xlab("Decade") + ylab("Percentage change on year ago") + theme(legend.position="False")
```



As we can see, indeed it appears that the distributions of growth rates have shifted leftwards over time (slowdown) and became increasingly less spread out (moderation).

```
rm(ukgdp,ukgdp.decade, avegr.preGR, i, ukgdp.trunc, recs)
```

2.3 Representing panel data: Cross-country differences in living standards over time - convergence and divergence

The last section of the chapter uses the full panel dataset to provide examples of representations that use both cross-sectional and time variation.

First, load the full panel dataset in long form and prepare for analysis:

```
longdata <- read_excel("data_ch1.xlsx", sheet = "gnipcppp_long")

longdata$Code <- as.factor(longdata$Code)
longdata$Region <- as.factor(longdata$Region)
longdata$Country <- as.factor(longdata$Country)
```

The first few rows of data look as follows:

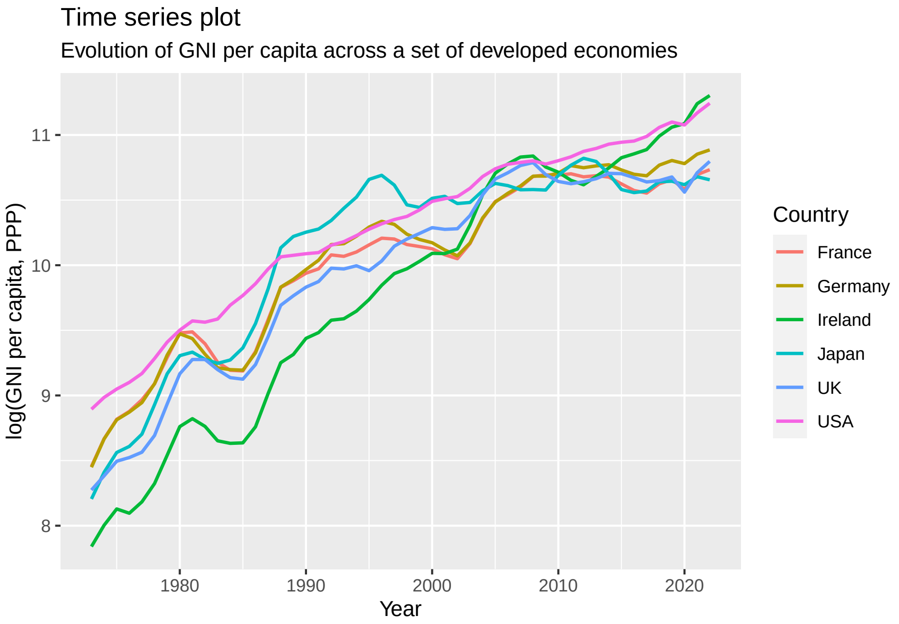
```
head(longdata)
```

```
## # A tibble: 6 x 5
##   Country     Code  Region      Year rGNIpc
##   <fct>       <fct> <fct>      <dbl>  <dbl>
## 1 Afghanistan AFG   South Asia  1973     NA
## 2 Afghanistan AFG   South Asia  1974     NA
## 3 Afghanistan AFG   South Asia  1975     NA
## 4 Afghanistan AFG   South Asia  1976     NA
## 5 Afghanistan AFG   South Asia  1977     NA
## 6 Afghanistan AFG   South Asia  1978     NA
```

An observation (a row of data) is now for country-year pairs. We have data on 212 countries for 50 years (1973-2022), so 10600 (=212x50) observations in total. At this stage it should be clear that panel data allows us to represent both cross-sectional variations across countries at a specific period of time (for example, from the full panel we can obtain a cross section at a specific year, as we did two sections ago) and time-series variations for a specific country (for example, by only keeping data for an individual country, resulting in a time-series data). However, in addition to simply converting our data to an individual cross-section or an individual time series, we can also represent patterns over both countries and time, which we can't do in a simple cross-section or time-series.

For example, we can (informally) treat the panel as a set of individual time series for different countries and present time series plots of GNI per capita for sets of countries. The following graph presents a collection of time series for a set of large developed economies

```
df <- subset(longdata, Country == "UK"|
         Country == "USA"|
         Country == "Germany"|
         Country == "France"|
         Country == "Japan"|
           Country == "Ireland")

ggplot(df, aes(x=Year, y=log(rGNIpc), col=Country)) +
  geom_line(linewidth=0.8) +
  xlab("Year") +
  ylab("log(GNI per capita, PPP)") +
  labs(title = "Time series plot", subtitle="Evolution of GNI per capita across a set of develope
```


Time series plot

Evolution of GNI per capita across a set of developed economies

We observe that from 1973 to 2022 GNI per capita (PPP) evolved in a fairly similar way. In addition, while differences (in proportional terms given the log scale) were larger in 1970s, they converged over time. This is especially strongly pronounced until around the period of the Great Recession after which some relative divergence relative to the US seems to have occurred. It seems that the productivity slowdown which we documented for UK earlier, also seems to be common across the advanced economies, and the divergence relative to the US could be down to the fact that among these US did not experience a slowdown of the same magnitude (subject to the PPP exchange rate adjustments).

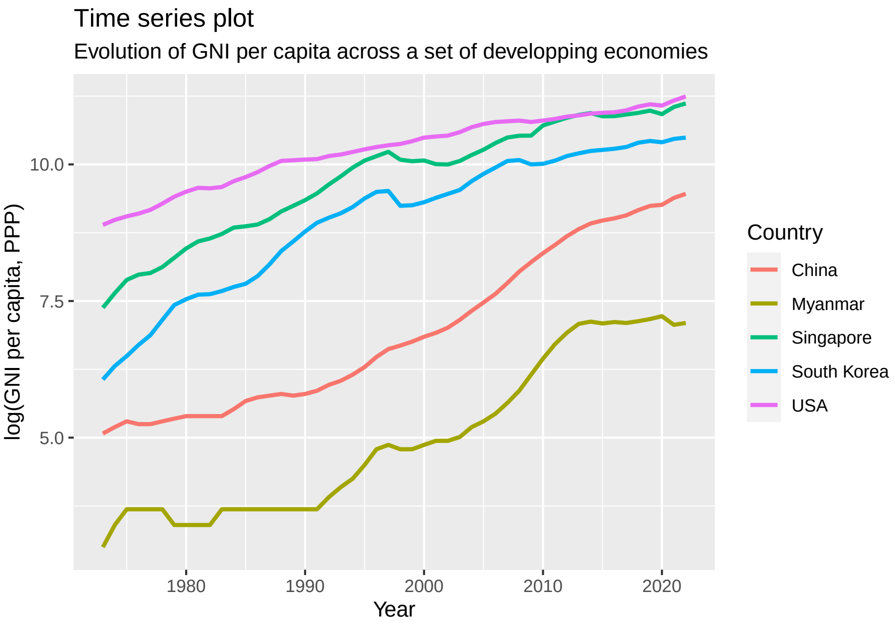
We can also see that the series fluctuate in a fairly similar way relative to trend [8] indicating that business cycles in different countries tend to be somewhat

[8] It can be noted that the observed series for the US seems to exhibit less fluctuations relative to other countries. This is largely a matter of the way GNI per capita (PPP) series are constructed to convert units of measurement to international US dollars. For this reason

synchronized. For example, all countries see declines relative to trend during the early 1980s recessions, the Great Recession, and the Covid-19 recession.

On the other hand, if we make a similar plot for a set of fast growing (typically Asian) economies, with the US for reference, we observe the following:

```r
df <- subset(longdata, Country == "USA"|
               Country == "South Korea"|
               Country == "Singapore"|
               Country == "China"|
               Country == "Myanmar")

ggplot(df, aes(x=Year, y=log(rGNIpc), col=Country)) +
  geom_line(size=1)+
  xlab("Year") +
  ylab("log(GNI per capita, PPP)") +
  labs(title = "Time series plot", subtitle="Evolution of GNI per capita across a set
```



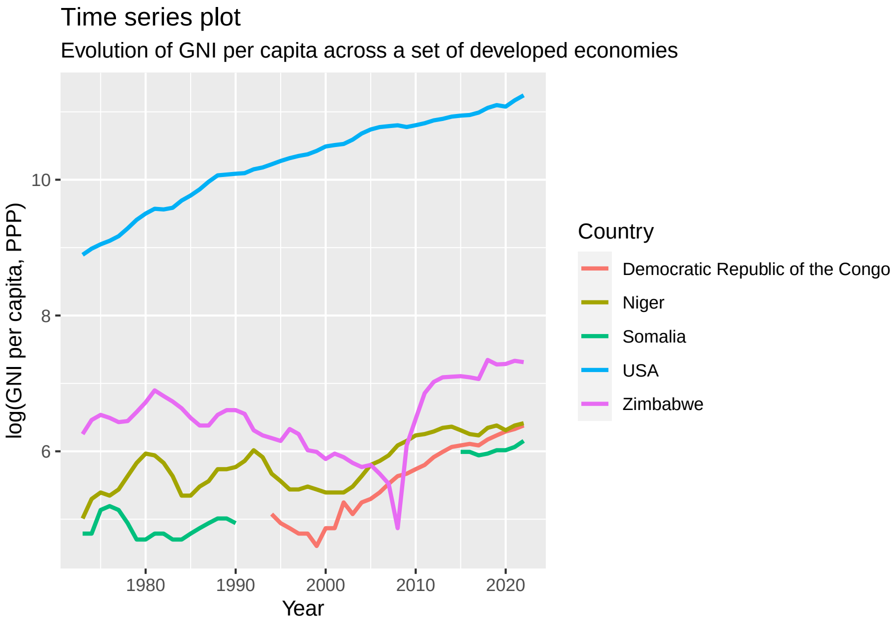
While these countries had much lower GNI per capita in the 1970s, they experienced faster overall growth relative to advanced economies over the next decades, leading to a decline in the proportional gap in living standards.

At this stage it may be natural to conjecture that typically countries with lower income per capita will grow faster than those with higher income per capita,

series for countries other than US fluctuate with exchange rate fluctuations, while the series for the US does not.

leading to convergence of living standards over time. Based on the above observations this appears sensible, and is sometimes referred to as the *convergence hypothesis*. However, we have to be careful because in its basic form the convergence hypothesis does not account for experiences of other countries. For example, a similar plot for a set of countries in Sub-Saharan Africa, yields

```
df <- subset(longdata, Country == "USA"|
               Country == "Democratic Republic of the Congo"|
               Country == "Zimbabwe"|
               Country == "Niger"|
               Country == "Somalia")

ggplot(df, aes(x=Year, y=log(rGNIpc), col=Country)) +
  geom_line(size=1)+
  xlab("Year") +
  ylab("log(GNI per capita, PPP)") +
  labs(title = "Time series plot", subtitle="Evolution of GNI per capita across a set of develope
```



While the chosen countries had much lower per capita income than US in the beginning of the period, over time living standards have not only not converged, but arguably diverged.

To understand the pattern of convergence/divergence further, it may be useful to examine the association between countries' GNI per capita in 1973 and their typical growth rates over the next decades. To do so, first, let's calculate each country's "typical" growth rate form from 1973 to 2019 (not until 2022 to disregard the effects of the Covid-19 recession). There could be different ways to define “typical”. A sensible one is as the counterfactual constant annual growth rate that would have led to the economy growing from its level in 1973 to 2019 (see the previous section). Let $Y_{i,t}$ be country i’s GNI per capita (PPP) in year t. Then if $Y_{i,t}$ grew at a constant annual rate of g_i then

$$Y_{i,2019} = Y_{i,1973} \times (1+g_i)^{46} \Rightarrow g_i = \left(\frac{Y_{i,2019}}{Y_{i,1973}}\right)^{1/46} - 1$$

Construct a new varible equal to the “typical growth rate” from 1973 to 2019:

```
longdata<-longdata %>% group_by(Country) %>% mutate(rGNIpc.gr = (rGNIpc[Year==2019]/rGN
```

Now, keep only one observation from 1973 for each country:

```
df<-longdata %>% group_by(Country) %>% filter(Year==1973&!is.na(rGNIpc.gr))
```

Note that we are left with only 97 countries, as for the others data on GNI p.c. for either 1973 or 2019 was missing.

The data we are using now is presented below:

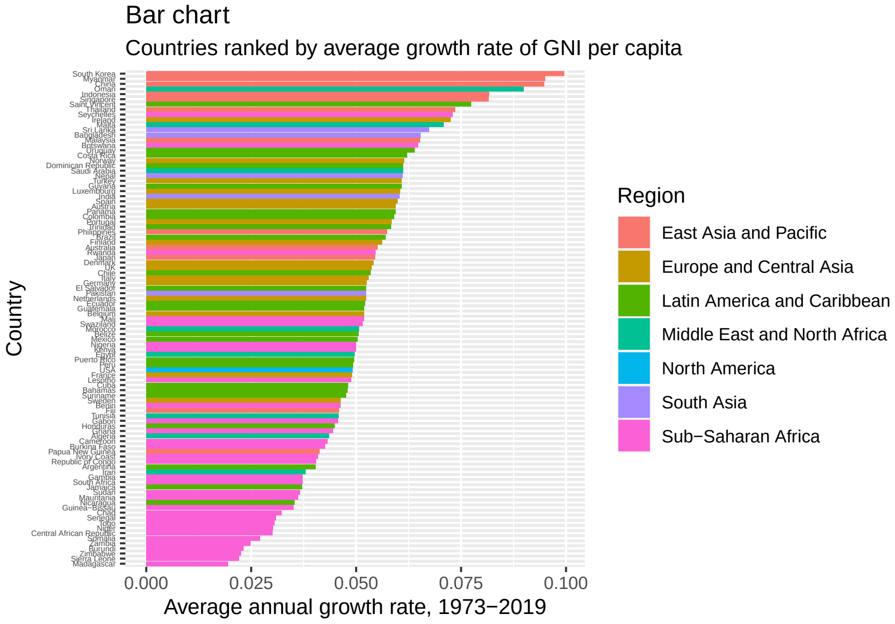
```
df %>% arrange(rGNIpc.gr)
```

```
## # A tibble: 97 x 6
## # Groups:   Country [97]
##    Country                  Code  Region              Year rGNIpc rGNIpc.gr
##    <fct>                    <fct> <fct>              <dbl>  <dbl>     <dbl>
##  1 Madagascar               MDG   Sub-Saharan Africa  1973    210    0.0195
##  2 Sierra Leone             SLE   Sub-Saharan Africa  1973    190    0.0221
##  3 Zimbabwe                 ZWE   Sub-Saharan Africa  1973    520    0.0225
##  4 Burundi                  BDI   Sub-Saharan Africa  1973     80    0.0232
##  5 Zambia                   ZMB   Sub-Saharan Africa  1973    450    0.0248
##  6 Somalia                  SOM   Sub-Saharan Africa  1973    120    0.0271
##  7 Central African Republic CAF   Sub-Saharan Africa  1973    120    0.0301
##  8 Niger                    NER   Sub-Saharan Africa  1973    150    0.0302
##  9 Togo                     TGO   Sub-Saharan Africa  1973    220    0.0306
## 10 Senegal                  SEN   Sub-Saharan Africa  1973    360    0.0309
## # i 87 more rows
```

While this is a cross section of countries in 1973, it includes information on the growth rate for subsequent decades, summarized in a new variable `rGNIpc.gr`, which was obtained from the full panel dataset.

Country differences in average annual growth rates over the period can be observed below:

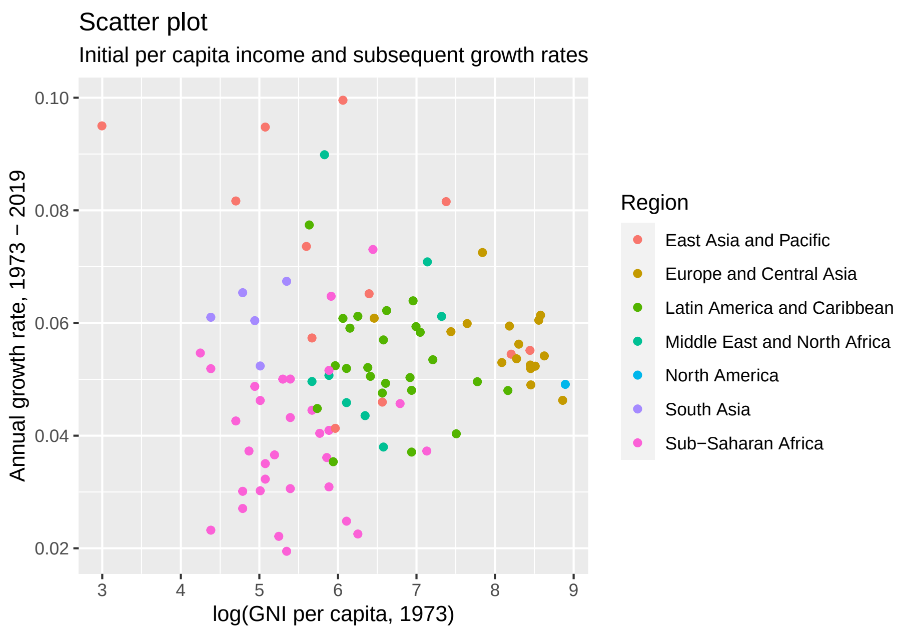
```
ggplot(df,aes(x=rGNIpc.gr,y=fct_reorder(Country,rGNIpc.gr))) +
  geom_bar(stat="identity", aes(fill=Region)) +
  xlab("Average annual growth rate, 1973-2019") +
  ylab("Country") +
  labs(title = "Bar chart", subtitle="Countries ranked by average growth rate of GNI per capita")
```



We observe that there is a large concentration of countries from Sub-Saharan Africa, Middle East and North Africa, and Latin America and the Caribbean among slowest growing countries.

If we now plot initial GNI p.c. (in 1973) against subsequent growth rates we obtain

```
ggplot(df,aes(x=log(rGNIpc),y=(rGNIpc.gr), color=Region)) +
  geom_point() +
  xlab("log(GNI per capita, 1973)") +
  ylab("Annual growth rate, 1973 - 2019") +
  labs(title = "Scatter plot", subtitle="Initial per capita income and subsequent growth rates")
```



```
#+ facet_wrap(Region~., scales="free")
```

If there was systematic convergence across all countries, we would have observed that higher initial GNI per capita associates with lower subsequent growth rates. However, the graph does not seem to exhibit such pattern. In fact, if we calculate the correlation coefficient between the two variables

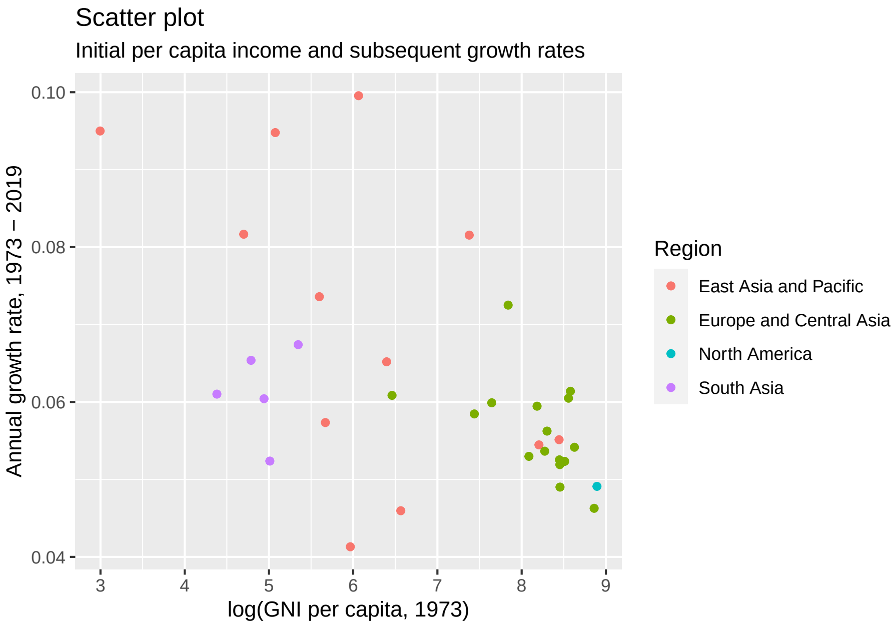
```
cor(log(df$rGNIpc),df$rGNIpc.gr)
```

```
## [1] 0.0828161
```

it turns out that the association between the two are in fact positive even if small. In the sample of 97 countries we work with, countries with high GNI per capita in 1973, also tended to grow faster on average during 1973-2019. If anything, we observe divergence, rather than convergence.

One aspect of this is that countries from Sub-Saharan Africa, Middle East and North Africa, and Latin America and Caribbean tend to be found in the bottom left quadrant of the graph. If we were to plot excluding these three regions

```
df2 <- subset(df, Region!="Sub-Saharan Africa"&Region !="Middle East and North Africa"&

ggplot(df2,aes(x=log(rGNIpc),y=(rGNIpc.gr), color=Region)) +
  geom_point() +
  xlab("log(GNI per capita, 1973)") +
  ylab("Annual growth rate, 1973 - 2019") +
  labs(title = "Scatter plot", subtitle="Initial per capita income and subsequent grow
```



```
#+ facet_wrap(Region~., scales="free")
```

we observe relationship that appears to be indeed negative. Calculating the correlation coefficient

```
cor(log(df2$rGNIpc),df2$rGNIpc.gr)
```

```
## [1] -0.54457
```

confirms that the association is negative and stronger than before (in terms of the magnitude of the coefficient). Nonetheless, it is also clear that even in the final subset of the data (which was in any case obtained somewhat arbitrarily) countries of similar initial income tend to have quite different growth experiences over time.

Understanding the reasons for the obseved convergence between countries (and the observed lack of convergence for some) is beyond the scope of this discussion, but is something you will return to during your studies.

Chapter 3

Representing climate change

This chapter uses data from various public sources to represent key trends related to climate, in global and UK context.

For ease of replication, all the data used in the chapter is available as an Excel file `data_ch2.xlsx` with different datasets saved as different tabs.

- You can replicate all analysis in the cloud here.
- Alternatively, to replicate the chapter in your own R installation, download the data file and corrsponding R-script from here and extract them inside a folder on your computer. Then set the working directory in R to the folder where the files are. For me, this is the following folder:

```r
setwd("/home/emil/Desktop/book")
```

In addition, run the following code to install all the R libraries that will be used for the analysis

```r
install.packages("readxl")
install.packages("dplyr")
install.packages("ggplot2")
install.packages("forcats")
```

and load them

```r
library(readxl)
library(dplyr)
library(ggplot2)
library(forcats)
```

3.1 Global trends

3.1.1 Monthly global surface temperature

This section uses monthly data on global surface (land and ocean) temperature from Rohde and Hausfather (2020) available here. For ease of replication, the data has been saved in the sheet gst of the data_ch2.xlsx Excel file.

```
gst <- read_xlsx("data_ch2.xlsx", sheet="gst")
head(gst)
```

```
## # A tibble: 6 x 3
##    Year Month TemperatureAnomaly
##   <dbl> <dbl>              <dbl>
## 1  1850     1             -0.769
## 2  1850     2             -0.227
## 3  1850     3             -0.374
## 4  1850     4             -0.591
## 5  1850     5             -0.623
## 6  1850     6             -0.359
```

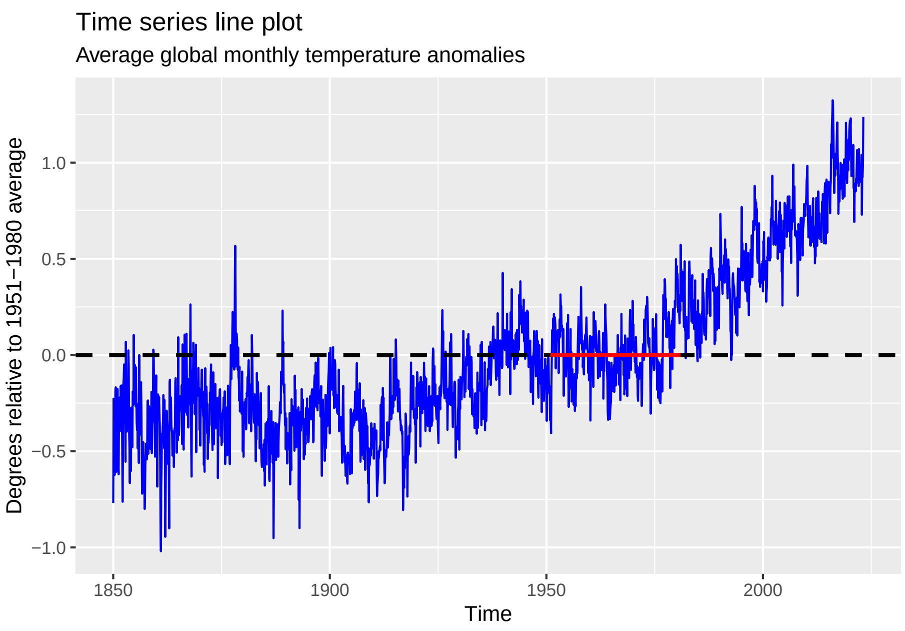
The data contains 2079 monthly observations (from January 1850 to March 2023) of average monthly global surface temperatures, reported in terms of *temperature anomalies* (i.e., differences) relative to average temperatures in the corresponding month from January 1951 to December 1980.

For the purpose of producing time-series plots, combine the Year and Month variables into a single variable Date

```
gst$Date<-as.Date(with(gst,paste(Year,Month,"01",sep="-")),"%Y-%m-%d")
```

Then plotting monthly temperature anomalies against time yields the following

```
ggplot(gst, aes(x=Date, y=TemperatureAnomaly)) +
  geom_line(col="blue") +
  geom_hline(yintercept=0, lty="dashed", col="black", linewidth=1) +
  geom_segment(y=0,yend=0, x=as.Date("1951-01-01"), xend=as.Date("1980-12-01"), col="re
  labs(title="Time series line plot", subtitle = "Average global monthly temperature an
  xlab("Time") + ylab("Degrees relative to 1951-1980 average")
```



From the 1850s until around the 1910s average global temperatures were approximately stable, at levels around 0.3 degrees below the 1951-1980 average. There was a pronounced upward trend from the 1910s until the 1940s. Temperatures remained relatively stable until the 1970s. Afterwards, they have been rising steadily and as of 2023 are around 1 degree above the 1951-1980 average. Throughout the whole period average global temperatures have risen by more than 1.3 degrees Celsius.

While the overall trend is clearly visible, temperature anomalies exhibit large fluctuations at monthly frequency. One way to represent the trend more clearly is to convert the monthly data to 10-year averages.

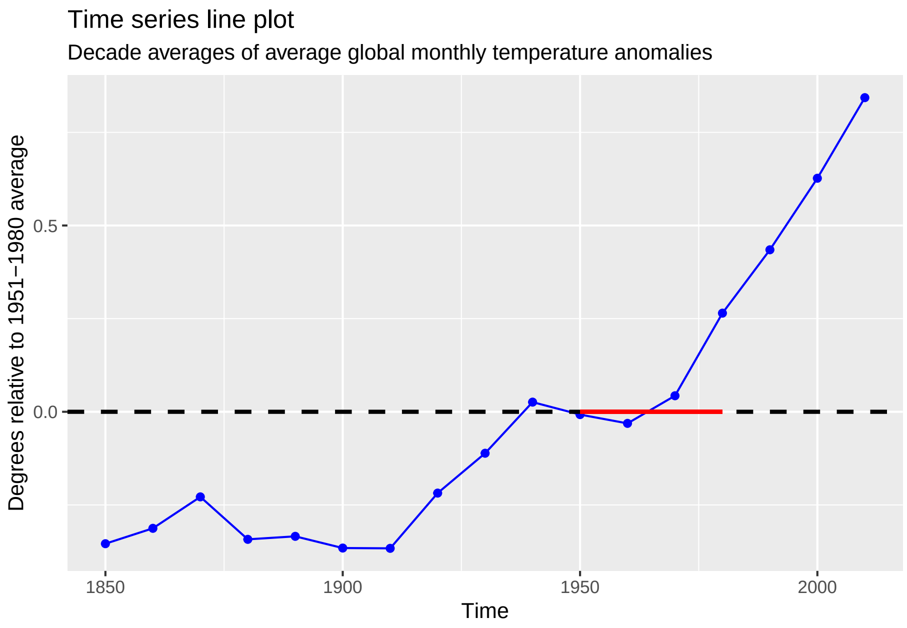
```r
# Create new data frame based on the original one, and create a variable for the decade
gst.decade <- gst %>% mutate(Decade = floor(Year/10)*10)

# Create a new variable which equals the mean monthly anomaly for all months in each decade
gst.decade <- gst.decade %>% group_by(Decade) %>% mutate(aveTempAnomaly = mean(TemperatureAnomaly

# Keep only one observation per decade
gst.decade <- gst.decade %>% group_by(Decade) %>% filter(row_number() == 1)

# Remove observations for the 2020-2029 decade, as sample is smaller
gst.decade <- subset(gst.decade, Decade<2020)

# Report a table of average temperatures by decade
tab <- cbind(gst.decade$Decade, gst.decade$aveTempAnomaly)
tab <- gst.decade %>% select(c("Decade", "aveTempAnomaly"))
tab
```

```
## # A tibble: 17 x 2
## # Groups:   Decade [17]
##    Decade aveTempAnomaly
##     <dbl>          <dbl>
##  1   1850       -0.354
##  2   1860       -0.313
##  3   1870       -0.228
##  4   1880       -0.342
##  5   1890       -0.334
##  6   1900       -0.366
##  7   1910       -0.366
##  8   1920       -0.218
##  9   1930       -0.111
## 10   1940        0.0260
## 11   1950       -0.00751
## 12   1960       -0.0311
## 13   1970        0.0430
## 14   1980        0.265
## 15   1990        0.435
## 16   2000        0.627
## 17   2010        0.843
```

The table of 10-year averages confirms the above discussion.

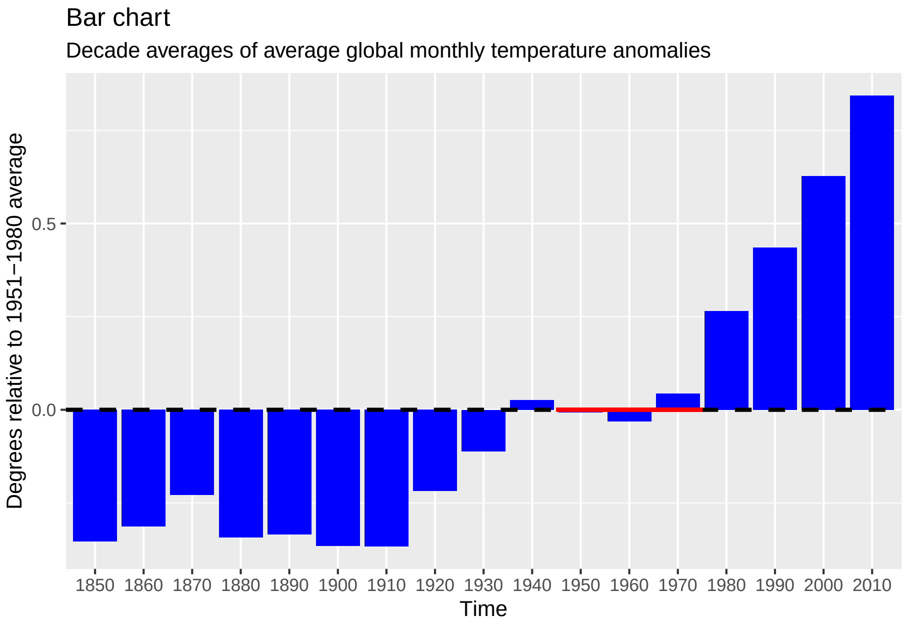
The information from the table can be equivalently presented as a time-series line plot

```r
ggplot(gst.decade, aes(x=Decade, y=aveTempAnomaly)) +
  geom_point(col="blue") +
  geom_line(col="blue")+
  geom_hline(yintercept=0, lty="dashed", col="black", linewidth=1) +
  geom_segment(y=0,yend=0, x=1950, xend=1980, col="red", linewidth=1) +
  labs(title="Time series line plot", subtitle = "Decade averages of average global mon
  xlab("Time") + ylab("Degrees relative to 1951-1980 average")
```



or a bar chart

```
ggplot(gst.decade, aes(x=as.factor(Decade), y=aveTempAnomaly)) +
  geom_bar(stat="identity", fill="blue")+
  geom_hline(yintercept=0, lty="dashed", col="black", linewidth=1) +
  geom_segment(y=0,yend=0, x=10.5, xend=13.5, col="red", linewidth=1) +
  labs(title="Bar chart", subtitle = "Decade averages of average global monthly temperature anoma
  xlab("Time") + ylab("Degrees relative to 1951-1980 average")
```



In summary, since 1850 there has been a marked increase in global surface temperatures, occurring over two distinct stages - 1910s to 1940s and then 1970s to present.

```
# Clear R environment
rm(list = ls())
```

3.1.2 Changes in ice mass

This section uses data from NASA/JPL (2019), on the Antarctic ice mass measured by satellites, available here. For ease of replication, the data has been saved in the sheet `ice` of the `data_ch2.xlsx` Excel file.

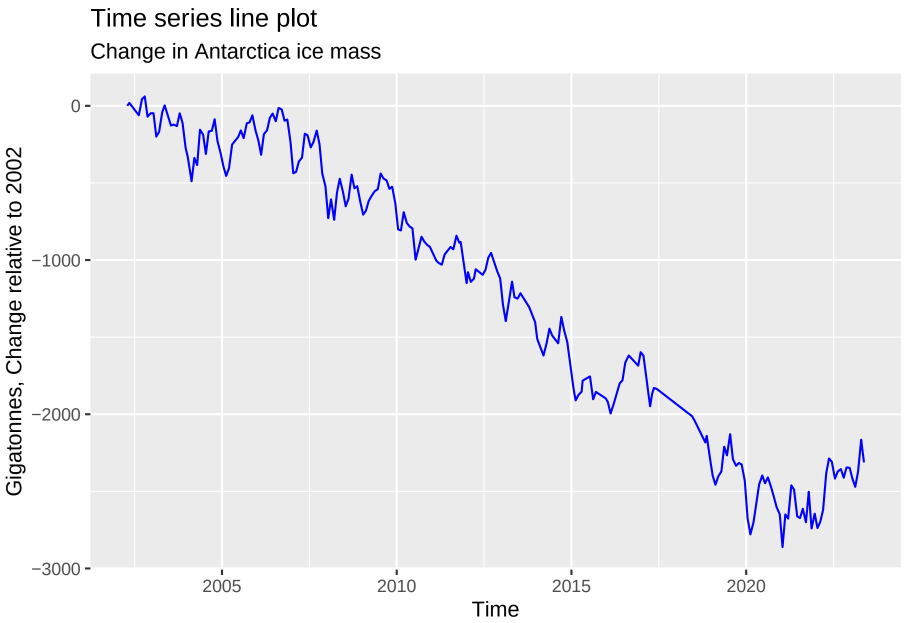
```
antarctic <- read_xlsx("data_ch2.xlsx", sheet="ice")
head(antarctic)
```

```
## # A tibble: 6 x 2
##    Time  Mass
##   <dbl> <dbl>
## 1 2002.   0
## 2 2002.  18.9
## 3 2003. -61.1
## 4 2003.  43.0
## 5 2003.  61.3
## 6 2003. -70.2
```

The data is an irregularly sampled time series: time is measured as year, reported in decimal form depending on the point in the year when satellites made a measurement. The mass of Antarctic ice is reported as *anomaly* (i.e., change) in gigatonnes, relative to February 2002.

Plotting the Antarctic mass anomaly against time yields

```
ggplot(antarctic, aes(x=Time, y=Mass)) + geom_line(col="blue") +
  labs(title="Time series line plot", subtitle = "Change in Antarctica ice mass") +
  xlab("Time") + ylab("Gigatonnes, Change relative to 2002")
```



We observe that the mass of the Antarctic ice cap has decreased by around 3000 gigatonnes from 2002 to 2023.

```
# Clear R environment
rm(list = ls())
```

3.1.3 Sea level

This section uses data from MEaSUREs (2023), on the global mean sea level measured by satellites, available here. For ease of replication, the data has been saved in the sheet `sea` of the `data_ch2.xlsx` Excel file.

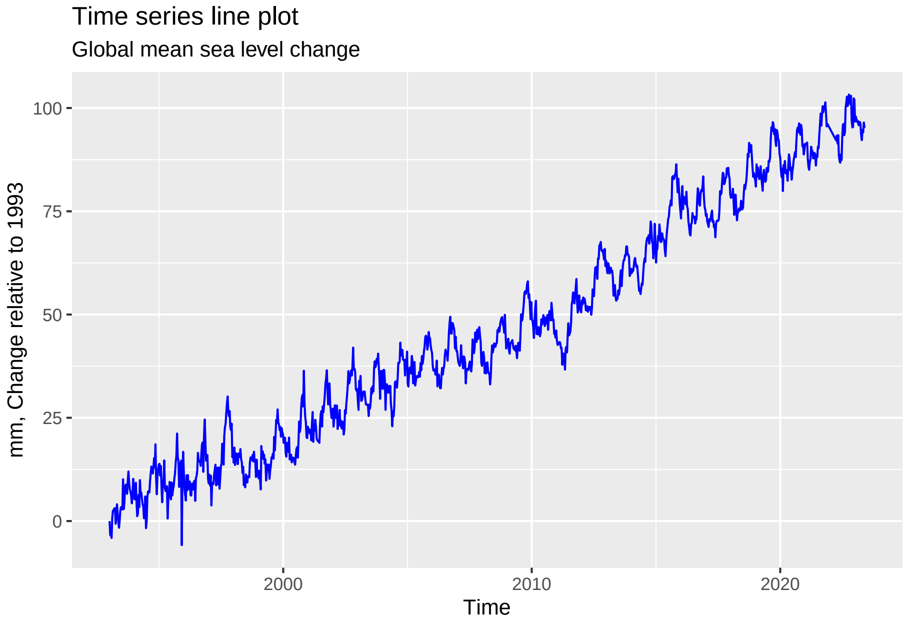
```
sea <- read_xlsx("data_ch2.xlsx", sheet="sea")
head(sea)
```

```
## # A tibble: 6 x 2
##    Time   GMSL
##   <dbl>  <dbl>
## 1 1993.  0
## 2 1993. -3.4
## 3 1993. -3.36
## 4 1993. -4.1
## 5 1993.  0.730
## 6 1993.  2.51
```

Similarly to the data on ice mass, this data is an irregularly sampled time series: time is measured by year reported in decimal form depending on the point in the year when satellites made a measurement. The average global sea level (GMSL) is reported in the form of *anomaly* (i.e., change) in millimeters, relative to January 1993.

Plotting the mean seal level anomaly against time yields

```
ggplot(sea, aes(x=Time, y=GMSL)) + geom_line(col="blue") +
  labs(title="Time series line plot", subtitle = "Global mean sea level change") +
  xlab("Time") + ylab("mm, Change relative to 1993")
```



We observe that from 1993 to 2023, global mean sea level has raised by approximately 100 mm (or 10 cm).

```
# Clear R environment
rm(list = ls())
```

3.1.4 CO2 concentration

It is well understood that one of the key drivers of changes in the climate, is changes in the concentration of greenhouse gases in the atmosphere. Greenhouse gases (including CO2, methane, NO and others) absorb the radiation of heat reflected by the Earth surface and cause heating of the atmosphere.

This section uses monthly time-series data on atmospheric carbon dioxide (CO2) levels measured by NOAA at Mauna Loa Observatory, Hawaii, since 1958. The data comes from Tans and Keeling (2023), and is available here. For ease of replication, the data has been saved in the sheet `co2` of the `data_ch2.xlsx` Excel file.

Loading, preparing, and printing the first few rows of data

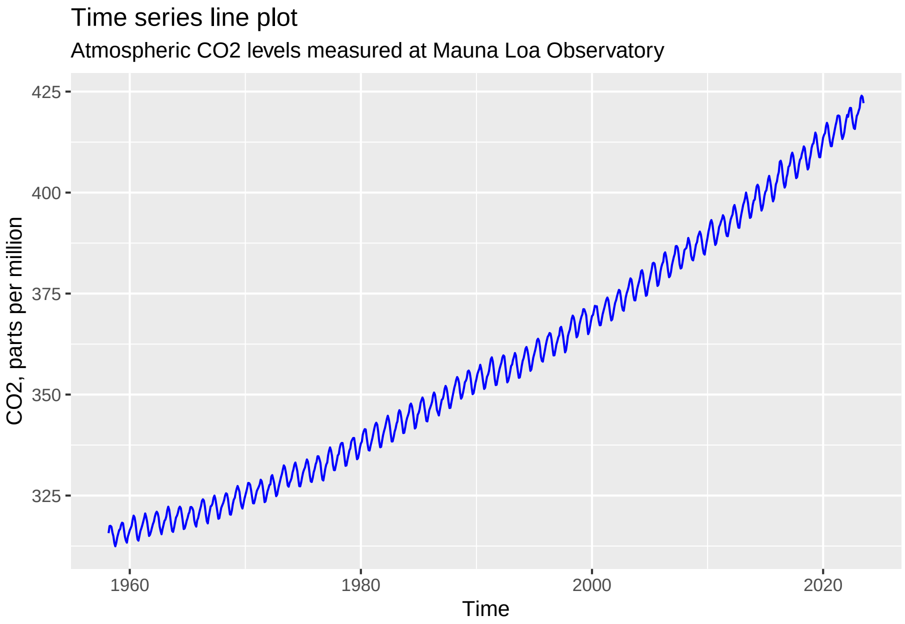
```r
co2df <- read_xlsx("data_ch2.xlsx", sheet = "co2")
co2df$Time <- as.Date(co2df$Time)
head(co2df)
```

```
## # A tibble: 6 x 2
##   Time          co2
##   <date>      <dbl>
## 1 1958-03-01  316.
## 2 1958-04-01  317.
## 3 1958-05-01  318.
## 4 1958-06-01  317.
## 5 1958-07-01  316.
## 6 1958-08-01  315.
```

we observe that data is monthly, from March 1958 to July 2023. Atmospheric CO2 levels (the variable `co2`) are measured in *parts per million.*

Plotting CO2 levels against time yields

```r
ggplot(co2df, aes(x=Time, y=co2)) + geom_line(col="blue")+
  labs(title="Time series line plot", subtitle = "Atmospheric CO2 levels measured at Mauna Loa Ob
  xlab("Time") + ylab("CO2, parts per million")
```



Atmospheric concentration of CO2 has raised steadily from 315 ppm in March 1958 to 422 ppm in July 2023. We observe that CO2 levels fluctuate seasonally, but fluctuations are small relative to the long run trend.

```
# Clear R environment
rm(list = ls())
```

3.1.5 CO2 emissions

There is evidence from ice-core data (e.g., see here) that CO2 concentrations in the atmosphere have varied considerably over hundreds of thousands of years, driving cycles of warming and cooling of the planet. However, current CO2 concentrations are unprecedented, and a key reason for this is human activity, especially the release of greenhouse gases from fossil fuels which has increased considerably since the Industrial Revolution.

This section uses annual time-series data on the World's total CO2 emissions from fossil fuels. The data comes from Global Carbon Budget (2022a), and is available here. For ease of replication, the data has been saved in the sheet `co2emissions` of the `data_ch2.xlsx` Excel file.

Load the data and print first few rows

```
co2world <- read_xlsx("data_ch2.xlsx", sheet = "co2emissions")
head(co2world)
```

```
## # A tibble: 6 x 2
##     Year     co2
##    <dbl>   <dbl>
## 1  1750 9350528
## 2  1751 9350528
## 3  1752 9354192
## 4  1753 9354192
## 5  1754 9357856
## 6  1755 9361520
```

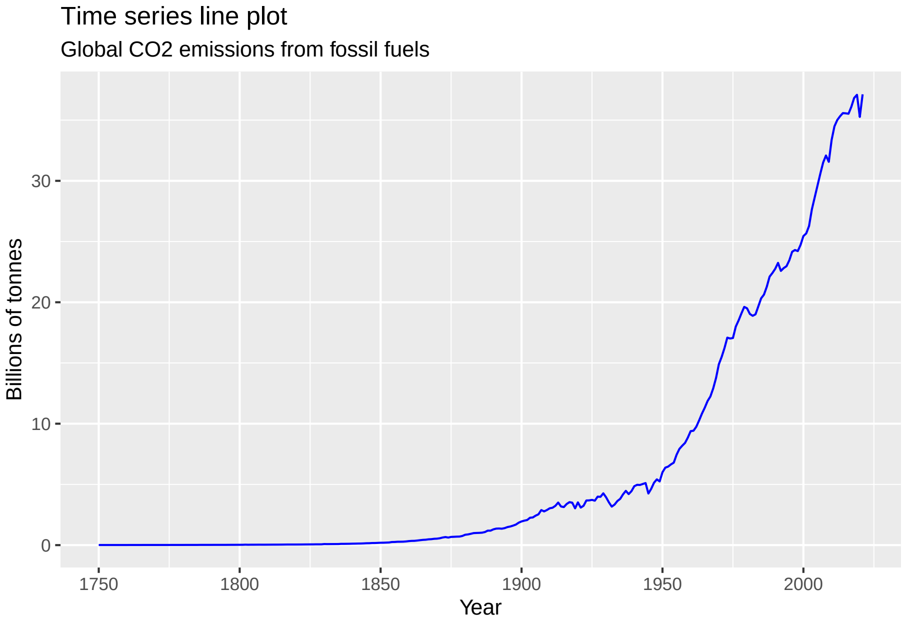
The data contains annual observations from 1750 until 2021 of the global CO2 emissions from fossil fuels, measured by the variable co2 in tonnes.

Given the scale of measurement, it is convenient to change the unit of measurement to billions of tones, by dividing the series by 1000000000:

```
co2world$co2 <- co2world$co2/1000000000
```

Plotting CO2 emissions against time yields

```
ggplot(co2world, aes(x=Year, y=co2)) + geom_line(col="blue")+
  labs(title="Time series line plot", subtitle = "Global CO2 emissions from fossil fuels") +
  xlab("Year") + ylab("Billions of tonnes")
```



It can be observed that CO2 emissions from human activity have increased sharply since 1750, and at an increasing rate for most of the 20th century.

```
# Clear R environment
rm(list=ls())
```

3.1.6 CO2 emissions and GNI per capita

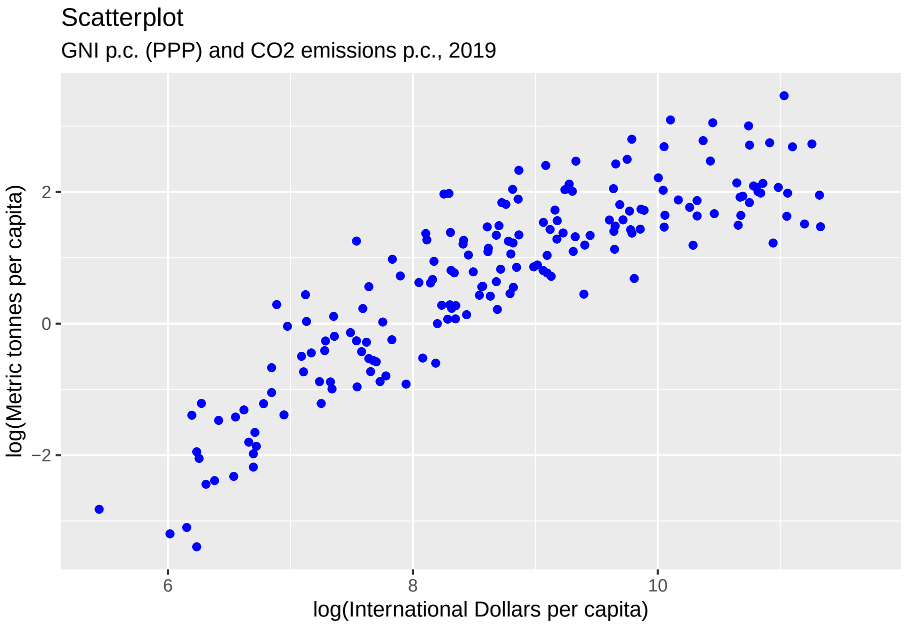
As we have already seen in the previous chapter, at country level CO2 emissions per capita associate closely with the level of economic development measured by GNI per capita (PPP). While the associations was already illustrated previously, we represent it here for completeness. Again, we use cross sectional data on GNI per capita (PPP) and CO2 emissions for 212 countries. The data comes from the World Bank's World Development Indicators database, and is available here. For ease of replication, the data has been saved in the sheet `gni_co2_pc_2019` of the `data_ch2.xlsx` Excel file.

Load the data and print first few rows

```
gnico22019 <- read_xlsx("data_ch2.xlsx", sheet = "gni_co2_pc_2019")
head(gnico22019)
```

```
## # A tibble: 6 x 3
##   Country        GNIpc  CO2pc
##   <chr>          <dbl>  <dbl>
## 1 Afghanistan      530  0.298
## 2 Albania         5230  1.75
## 3 Algeria         4050  3.99
## 4 American Samoa    NA NA
## 5 Andorra        46530  6.29
## 6 Angola          2040  0.754
```

```
ggplot(gnico22019, aes(x=log(GNIpc), y=log(CO2pc))) + geom_point(col="blue")+
  labs(title="Scatterplot", subtitle = "GNI p.c. (PPP) and CO2 emissions p.c., 2019") +
  xlab("log(International Dollars per capita)") + ylab("log(Metric tonnes per capita)")
```



Again, we can see that countries with high per-capita income, also tend to have high per-capita CO2 emissions.

```
# Clear R environment
rm(list = ls())
```

3.2 UK trends

3.2.1 UK temperatures

The data on temperature anomalies discussed above represented average global temperature anomalies, but climate varies geographically, and the overall change in climate affects different regions differently. We now turn attention to specific trend related to the climate of the UK.

This section uses annual time-series data on annual average temperature in the UK. The data comes from Met Office (2023a), and is available here. For ease of replication, the data has been saved in the sheet `uktemp` of the `data_ch2.xlsx` Excel file.

Loading the data and printing the first few rows

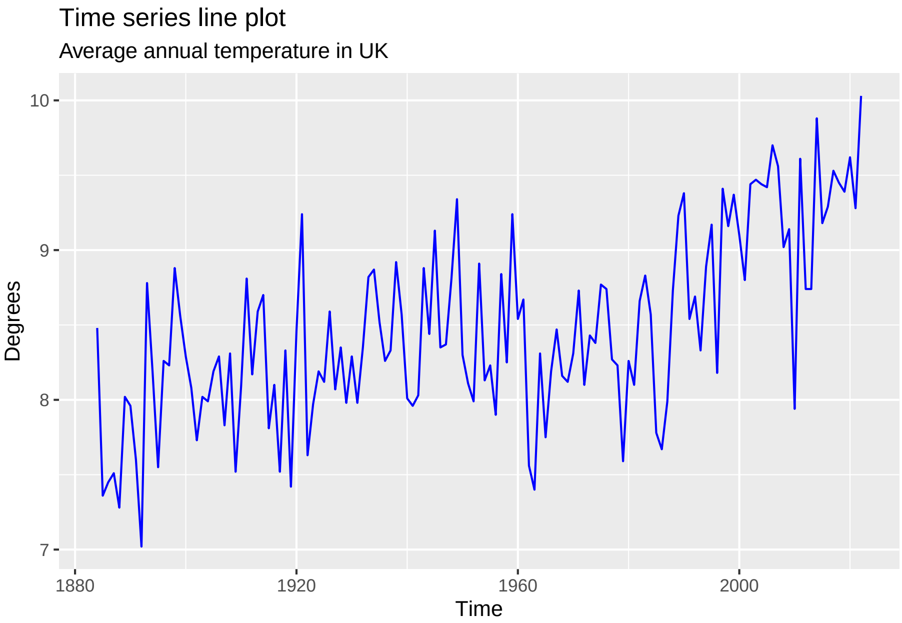
```
uktemp <- read_xlsx("data_ch2.xlsx", sheet = "uktemp")
head(uktemp)
```

```
## # A tibble: 6 x 2
##    Year AnnualMeanTemperature
##    <dbl>                      <dbl>
## 1  1884                        8.48
## 2  1885                        7.36
## 3  1886                        7.45
## 4  1887                        7.51
## 5  1888                        7.28
## 6  1889                        8.02
```

the data contains 139 annual observations (from 1884 to 2022) of the annual average temperature in the UK, measured in degrees Celsius.

Plotting temperatures against time

```
ggplot(uktemp, aes(x=Year, y=AnnualMeanTemperature)) +
  geom_line(col="blue") +
  labs(title="Time series line plot", subtitle = "Average annual temperature in UK") +
  xlab("Time") + ylab("Degrees")
```



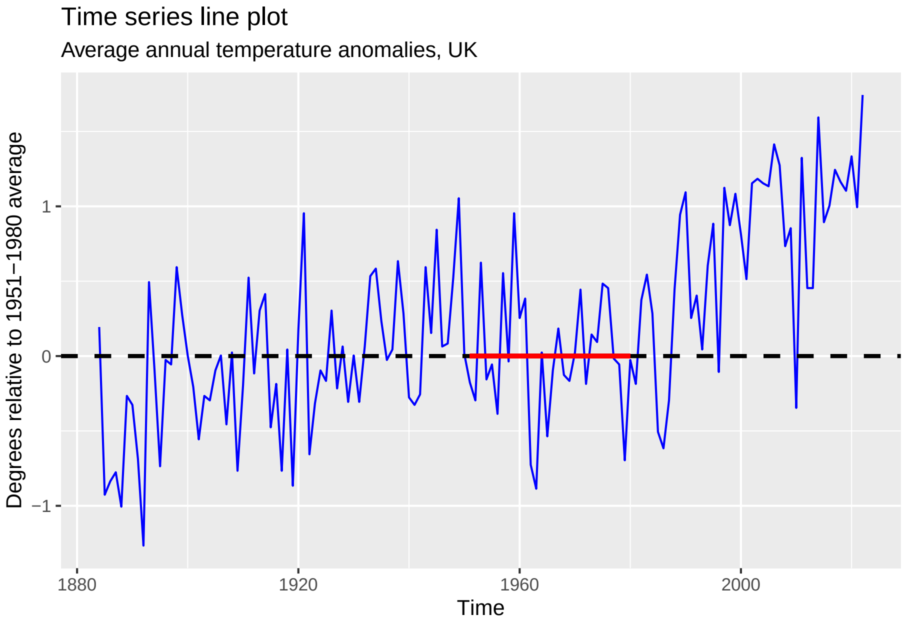
we observe that while temperature fluctuates considerably at annual frequency, overall there is a positive trend (possibly non-monotonic). Average annual temperatures in the end of the 19th century tended to be around 8 degrees, but have been typically above 9.5 degrees in the last years.

For purpose of comparison UK trends to the global trend documented above, we can measure temperatures in terms of anomalies relative to the 1951-1980 average. Construct a new variable, `AnnualAnomaly` equal to the difference between the temperature in a given year and the average of annual temperatures from 1951 to 1980:

```
mean5180 <- mean(subset(uktemp, Year>=1951 & Year<=1980)$AnnualMeanTemperature)
uktemp$AnnualAnomaly <- uktemp$AnnualMeanTemperature - mean5180
```

Plotting the annual anomaly against time

```
ggplot(uktemp, aes(x=Year, y=AnnualAnomaly)) +
  geom_line(col="blue") +
  geom_hline(yintercept=0, lty="dashed", col="black", linewidth=1) +
  geom_segment(y=0,yend=0, x=1951, xend=1980, col="red", linewidth=1) +
  labs(title="Time series line plot", subtitle = "Average annual temperature anomalies, UK") +
  xlab("Time") + ylab("Degrees relative to 1951-1980 average")
```

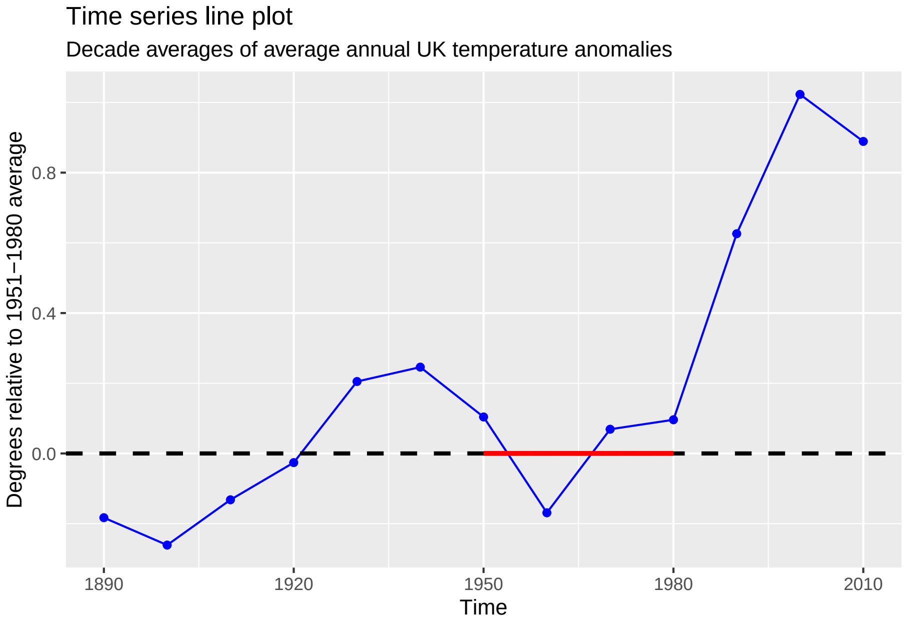


we observe that the pattern of temperatures in the UK looks very similar to the global pattern. Of course, it should be noted that measurement is not directly comparable - for one, global data was available at monthly, while UK at annual frequency. Furthermore, global temperatures were averages across land and ocean surface, while UK temperatures are land surface only.

Again, we can provide a clearer understanding of the long term trend if we average out annual fluctuations, for example, by converting annual data, to data on 10-year averages (as before):

```
# Create a variable measuring decade
uktemp.decade <- uktemp %>% mutate(Decade = floor(Year/10)*10)
```

```
# Create variable aveTempAnomaly equal to the average of annual temperatures for the c
uktemp.decade <- uktemp.decade %>% group_by(Decade) %>% mutate(aveTempAnomaly = mean(An

# Keep single observation for decade
uktemp.decade <- uktemp.decade %>% group_by(Decade) %>% filter(row_number() == 1)

# Remove observations for "incomplete" decades
uktemp.decade <- subset(uktemp.decade, Decade<2020&Decade>1880)

# Plot 10-year average temperatures against corresponding decades
ggplot(uktemp.decade, aes(x=Decade, y=aveTempAnomaly)) +
  geom_point(col="blue") +
  geom_line(col="blue")+
  geom_hline(yintercept=0, lty="dashed", col="black", linewidth=1) +
  geom_segment(y=0,yend=0, x=1950, xend=1980, col="red", linewidth=1) +
  labs(title="Time series line plot", subtitle = "Decade averages of average annual UK
  xlab("Time") + ylab("Degrees relative to 1951-1980 average")
```



The pattern for UK is very similar to the global pattern documented before, with the notable difference that UK temperatures followed a brief downward trend from 1940s to the 1960s.

```
# Clear R environment
rm(list = ls())
```

3.2.2 UK Rainfall

This section uses annual time-series data on annual average rainall in the UK. The data comes from Met Office (2023b), and is available here. For ease of replication, the data has been saved in the sheet `ukrain` of the `data_ch2.xlsx` Excel file.

Loading the data and printing the first few rows

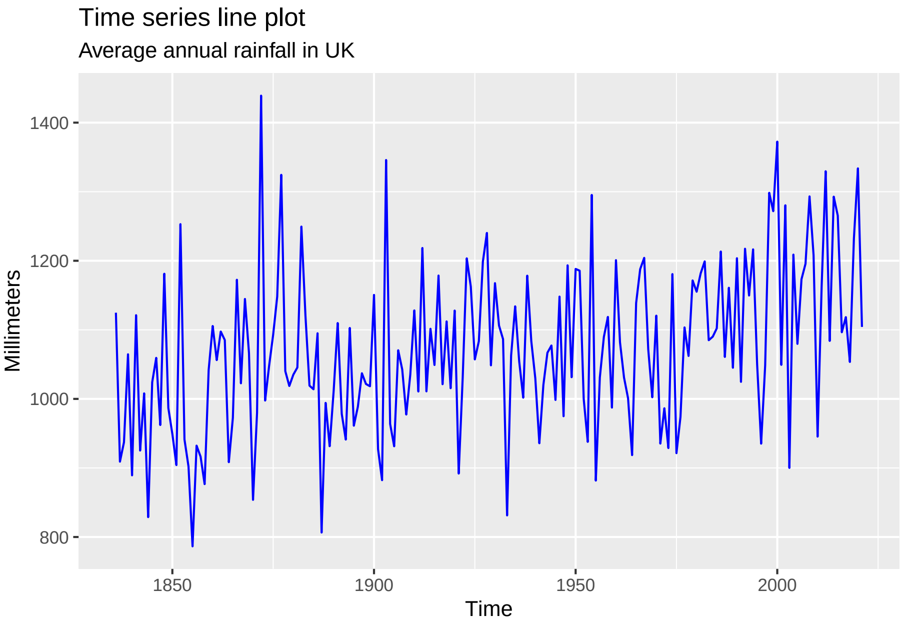
```
ukrain <- read_xlsx("data_ch2.xlsx", sheet = "ukrain")
head(ukrain)
```

```
## # A tibble: 6 x 2
##    Year  Rain
##   <dbl> <dbl>
## 1  1836 1125.
## 2  1837  909.
## 3  1838  937.
## 4  1839 1065.
## 5  1840  889.
## 6  1841 1121.
```

The data consists of 186 annual observations (from 1836 to 2021) of the average annual rainfall in UK, measured in millimeters.

Plotting average annual rainfall against time yields

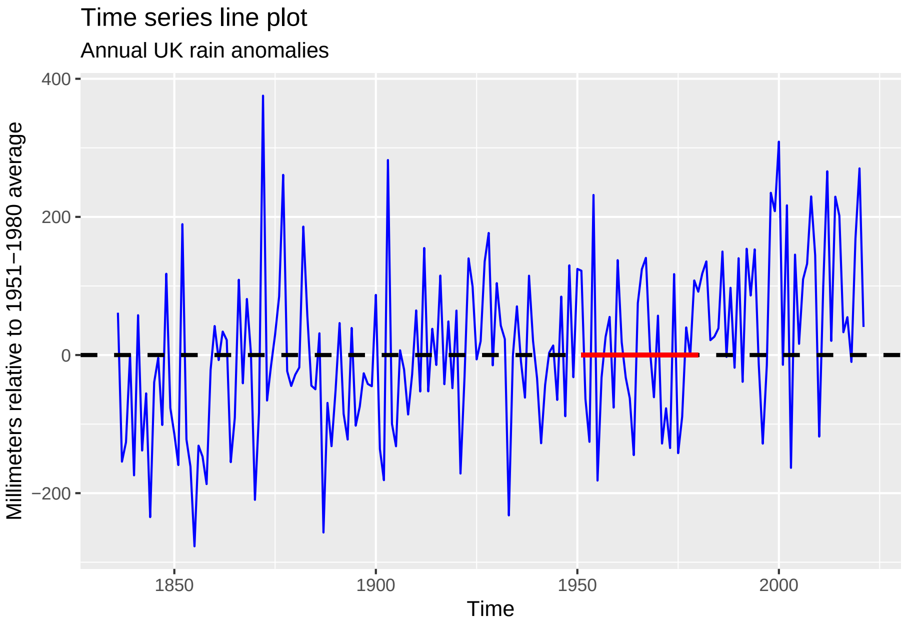
```
ggplot(ukrain, aes(x=Year, y=Rain)) +
  geom_line(col="blue") +
  labs(title="Time series line plot", subtitle = "Average annual rainfall in UK") +
  xlab("Time") + ylab("Millimeters")
```



Relative to average temperature, we can observe that average rainfall tends to be quite more volatile at annual frequency (it is relatively common for very wet and very dry years to occur, and there are large differences between wet and dry years). Nonetheless, we can also observe that there is an increasing long-term trend - typical rainfall in the begining of the 19th century was around 1000 mm, while in recent decades typical years see 1200 mm or rain.

Similarly to prior sections we can represent this more clearly by reporting rainfall in terms of annual anomalies (differences) relative to the 1951-1980 average

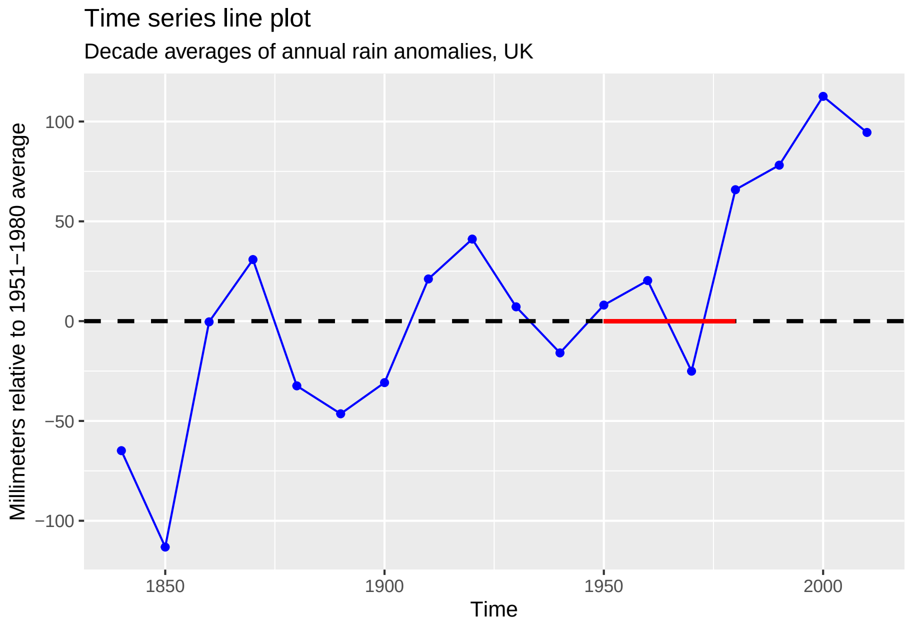
```
# Calucalte mean annual rainfall for 1951-1980
mean5180 <- mean(subset(ukrain, Year>=1951 & Year<=1980)$Rain)

# Calculate annual anomaly as difference between actual annual rainfall and the 1951-1
ukrain$AnnualAnomaly <- ukrain$Rain - mean5180

# Plot annual anomalies against year
ggplot(ukrain, aes(x=Year, y=AnnualAnomaly)) +
  geom_line(col="blue") +
  geom_hline(yintercept=0, lty="dashed", col="black", linewidth=1) +
  geom_segment(y=0,yend=0, x=1951, xend=1980, col="red", linewidth=1) +
  labs(title="Time series line plot", subtitle = "Annual UK rain anomalies") +
  xlab("Time") + ylab("Millimeters relative to 1951-1980 average")
```



where we can clearly see that typical years recently are wetter than typical years in 1951-1980, and 1951-1980 are on average wetter than in the begining of the observation window.

As before, to focus on trends rather than fluctuations, represent the data in terms of 10-year averages:

```
# Create a variable measuring the decade of observation
ukrain.decade <- ukrain %>% mutate(Decade = floor(Year/10)*10)

# Create a variable measuring the average anomaly per decade
ukrain.decade <- ukrain.decade %>% group_by(Decade) %>% mutate(aveAnomaly = mean(AnnualAnomaly))

# Keep single observation per decade
ukrain.decade <- ukrain.decade %>% group_by(Decade) %>% filter(row_number() == 1)

# Remove incomplete decades
ukrain.decade <- subset(ukrain.decade, Decade>1830&Decade<2020)


# Plot
ggplot(ukrain.decade, aes(x=Decade, y=aveAnomaly)) +
  geom_point(col="blue") +
  geom_line(col="blue")+
  geom_hline(yintercept=0, lty="dashed", col="black", linewidth=1) +
  geom_segment(y=0,yend=0, x=1950, xend=1980, col="red", linewidth=1) +
  labs(title="Time series line plot", subtitle = "Decade averages of annual rain anomal
  xlab("Time") + ylab("Millimeters relative to 1951-1980 average")
```



We can see that recent decades typically experienced about 100 mm (10 cm) annual rain, than in 1951-1980.

```
# Clear R environment
rm(list = ls())
```

3.2.3 UK CO2 emissions

This section uses annual time-series data on UK's CO2 emissions. The data comes from Global Carbon Budget (2022b), and is available here. For ease of replication, the data has been saved in the sheet `ukco2` of the `data_ch2.xlsx` Excel file.

Load the data and print first few rows

```
ukco2 <- read_xlsx("data_ch2.xlsx", sheet = "ukco2")
head(ukco2)
```

```
## # A tibble: 6 x 3
##    Year     co2 co2cons
##   <dbl>   <dbl>   <dbl>
## 1  1750 9350528      NA
## 2  1751 9350528      NA
## 3  1752 9354192      NA
## 4  1753 9354192      NA
## 5  1754 9357856      NA
## 6  1755 9361520      NA
```

The dataset contains annual data (from 1750 to 2021) on estimates of UK's - *production-based* CO2 emissions, measured by the variable `co2` in tonnes; and - *consumption-based* CO2 emissions, measured by the variable `co2cons` in tonnes.

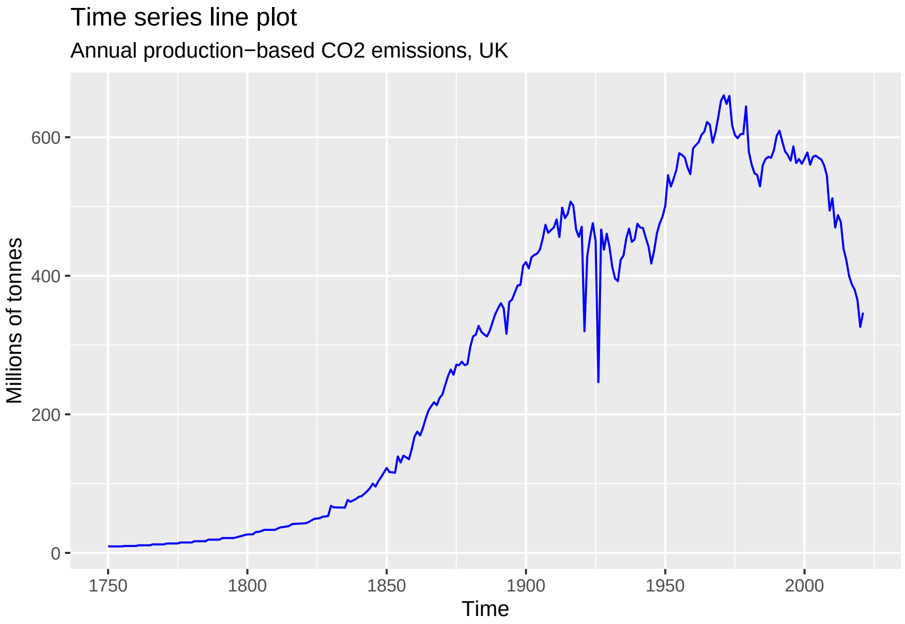
Production-based emissions record the amount of CO2 emitted through production within a country's own borders. Consumption-based emissions record the amount of CO2 emitted to produce the goods and services consumed within an economy (i.e., accounting for trade). While production-based emissions are available since 1750, consumption-based emissions are only available since 1990.

Given the scale of the variables, it is appropriate to report emissions in terms of millions of tonnes, rather than tonnes. To do this, divide emissions in tonnes by 1000000:

```
ukco2$co2 <- ukco2$co2/1000000
ukco2$co2cons <- ukco2$co2cons/1000000
```

Plotting UK's production-based CO2 emissions against time yields

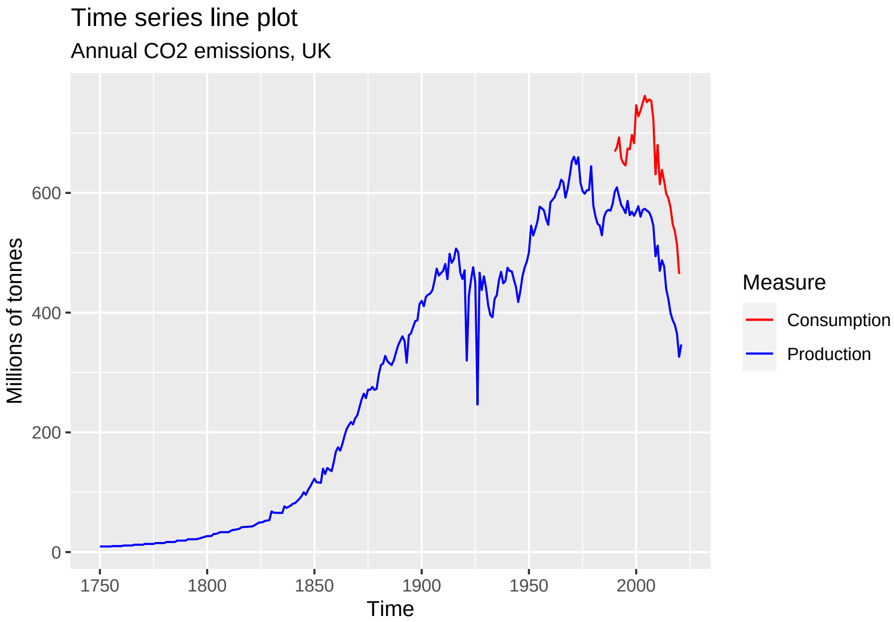
```
ggplot(ukco2, aes(x=Year, y=co2)) +
  geom_line(col="blue") +
  labs(title="Time series line plot", subtitle = "Annual production-based CO2 emissions, UK") +
  xlab("Time") + ylab("Millions of tonnes")
```



Since the outset of the Industrial Revolution, UK's CO2 emissions increased at an approximately exponential rate, peaking in the 1970s at levels around 650 millions of tonnes per year (and seeing a relative decline at the time of WW1 and the Great Depression). Following the sectorial restructuring in the economy away from manufacturing CO2 emissions embodied in production have declined to about 346 millions of tonnes as of 2022.

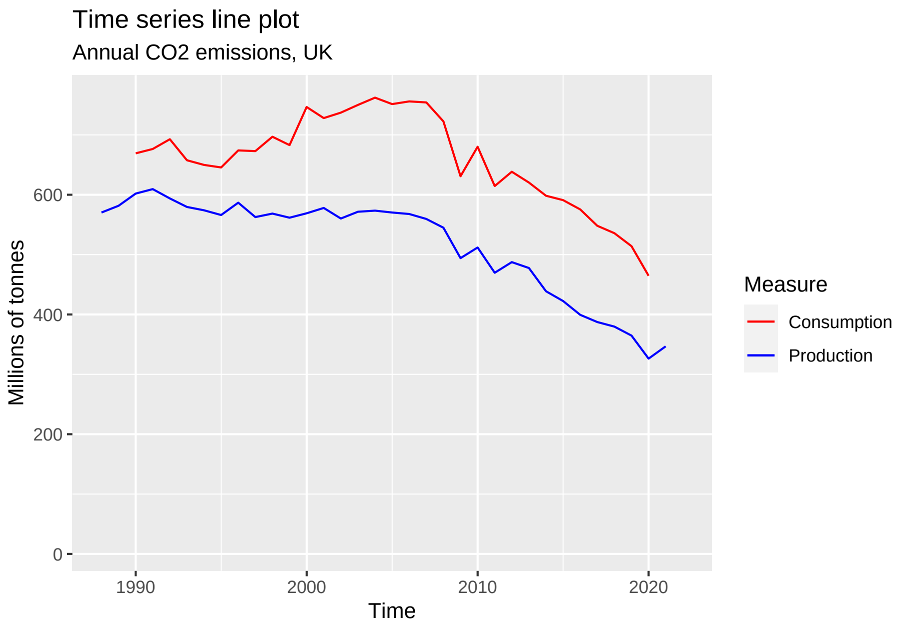
However, plotting consumption and production based emissions on the same graph for the same time window

```
ggplot(ukco2, aes(x=Year)) +
  geom_line(aes(y=co2,col="Production")) +
  geom_line(aes(y=co2cons,col="Consumption")) +
  labs(title="Time series line plot", subtitle = "Annual CO2 emissions, UK", col = "Mea
  xlab("Time") + ylab("Millions of tonnes") +
  scale_color_manual(values=c("red", "blue"))
```



or only for the part of the window where consumption-based measures are available

```
ggplot(ukco2, aes(x=Year)) +
  geom_line(aes(y=co2,col="Production")) +
  geom_line(aes(y=co2cons,col="Consumption")) +
  labs(title="Time series line plot", subtitle = "Annual CO2 emissions, UK", col = "Mea
  xlab("Time") + ylab("Millions of tonnes") +
  xlim(c(1988,2022)) +
  scale_color_manual(values=c("red", "blue"))
```



we observe that while UK's production embodies much less CO2 emissions than its consumption (i.e., UK is a net importer of "CO2" emissions). Still, even consumption-based CO2 emissions have seen a decline since the early 2000s.

Chapter 4

Representing income inequality

This chapter turns attention to representing the extent of income inequality using data.

For ease of replication, all the data used in the documented is available as an Excel file `data_ch3.xlsx` with different datasets saved as different tabs.

- You can replicate all analysis in the cloud here.
- Alternatively, to replicate the chapter in your own R installation, download the data file and corrsponding R-script from here and extract them inside a folder on your computer. Then set the working directory in R to the folder where the files are. For me, this is the following folder:

```r
setwd("/home/emil/Desktop/book")
```

In addition, run the following code to install all the R libraries that will be used for the analysis

```r
install.packages("readxl")
install.packages("dplyr")
install.packages("ggplot2")
install.packages("forcats")
install.packages("pracma")
install.packages("scales")
```

and load them

```r
library(readxl)
library(ggplot2)
library(scales)
library(dplyr)
library(forcats)
library(pracma)
```

4.1 Distribution of income in the FRS 2016/17

The Family Resources Survey (FRS), carried out by the ONS, with the NatCen, and NISRA, is an annual survey collecting information on a representative sample of UK households (Office for National Statistics, Social and Vital Statistics Division, Department for Work and Pensions, National Centre for Social Research, 2023). It includes rich information on the incomes and characteristics of individuals and households.

This section uses an excerpt from the FRS 2016/17 individual level data to provide some examples of representing key cross-sectional patterns related to the unequal distribution of income, and to discuss how income inequality can be measured. The data is not open-access, hence it is not provided with this document. Given this you will not be able to replicate this section (without obtaining the data), but the structure of the data is explained in sufficient detail for you to understand the code.

4.1.1 Summary of data

Let's start by loading the data and listing the first few rows

```
frs <- read_xlsx("FRS2017.xlsx")
head(frs)
```

```
## # A tibble: 6 x 4
##   age             educ                  sex    income
##   <chr>           <chr>                 <chr>   <dbl>
## 1 Age 65 and over <NA>                  Male      225
## 2 Age 65 and over <NA>                  Female    110
## 3 Age 25 to 34    Degree of equivalent  Female      0
## 4 Age 35 to 44    Degree of equivalent  Male     1093
## 5 Age 65 and over A-level or equivalent Female    671
## 6 Age 55 to 64    A-level or equivalent Male       76
```

The data is a cross-section of 33252 observations on four variables from the 2016/17 financial year. An observation (row of the data) corresponds to an individual (aged 16 and over). The four variables are - `age` - recording the age of the individual in bands - `educ` - recording the highest education attainment of the individual in bands - `sex` - recording the sex of the individual - `income` - recording the individual's total weekly income from all sources (including pensions and transfers) before tax.

To prepare the data for analysis 1. note that `age`, `educ` and `sex` are categorical (factor) variables which we can declare explicitly to avoid problems

```
# Declare variables as categorical
frs$age <- as.factor(frs$age)
frs$educ <- as.factor(frs$educ)
frs$sex <- as.factor(frs$sex)

# Change order of categories in `educ` in order of average income per group
frs$educ <- fct_reorder(frs$educ, frs$income, .fun=mean)
```

2. note that `income` is recorded in the form of weekly income (i.e., measured in pounds per week). While equivalent, it is often more natural to think in terms of monthly or annual income. For this reason we will convert the income variable into approximate equivalent annual income by multiplying weekly income by 52 (the number of weeks in a year

```
frs$income <- frs$income*52
```

Now `income` is measured in pounds per year, and corresponds to total income from all sources before tax. Hencefort, by "income" we will mean specifically this notion of income. It should be noted that inequality metrics discussed below are specific to this particular definition of income, and would be different for alternative income concepts (e.g., income after taxes, or labour income only).

To get a sense of the data, calculate some summary statistics for all variables in the dataframe

```
summary(frs)
```

```
##                  age                              educ              sex
##  Age 16 to 24    :2456   GCSE or equivalent   :11265   Female:17509
##  Age 25 to 34    :4821   A-level or equivalent: 5139   Male  :15743
##  Age 35 to 44    :5334   Degree of equivalent : 7795
##  Age 45 to 54    :5678   Postgraduate         : 3162
##  Age 55 to 64    :5734   NA's                 : 5891
##  Age 65 and over:9229
##      income
##  Min.   :      0
##  1st Qu.: 10088
##  Median : 17576
##  Mean   : 22751
##  3rd Qu.: 28340
##  Max.   :885768
```

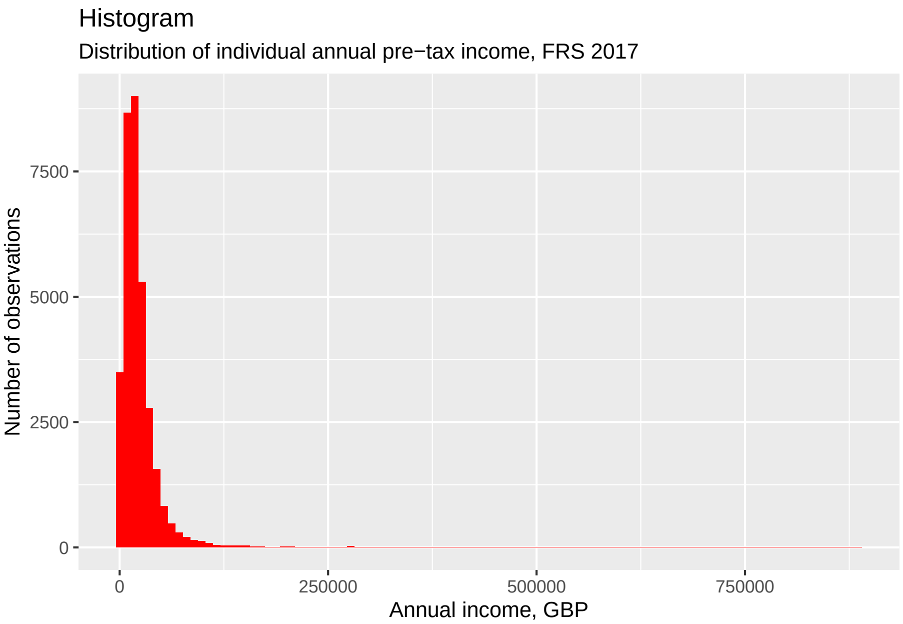
For categorical variables `summary()` reports the number of individuals in different categories. For example, we can see that the sample consists of 17509 females and 15743 males; most individuals' highest educational attainment is GSCE or equivalent (11265 individuals).

For numerical variables (in this case only `income`) `summary()` reports key summary statistics from the distribution. At this stage we can already see that income is very unequally distributed. Unsurprisingly, the minimum of the support of income is 0 (some individuals reported no income), while the richest individual in the data reported annual income of £885768. 25% of individuals in the data receive income below £10088, 50% receive income below £17576 (the median), 75% receive income below £28340. Mean income (at £22751) is above median (at £17576) indicating that the distribution is asymmetric.

4.1.2 Distribution of income: dispersion and skewness

To understand the distribution of income in the data, we start by plotting a histogram of income

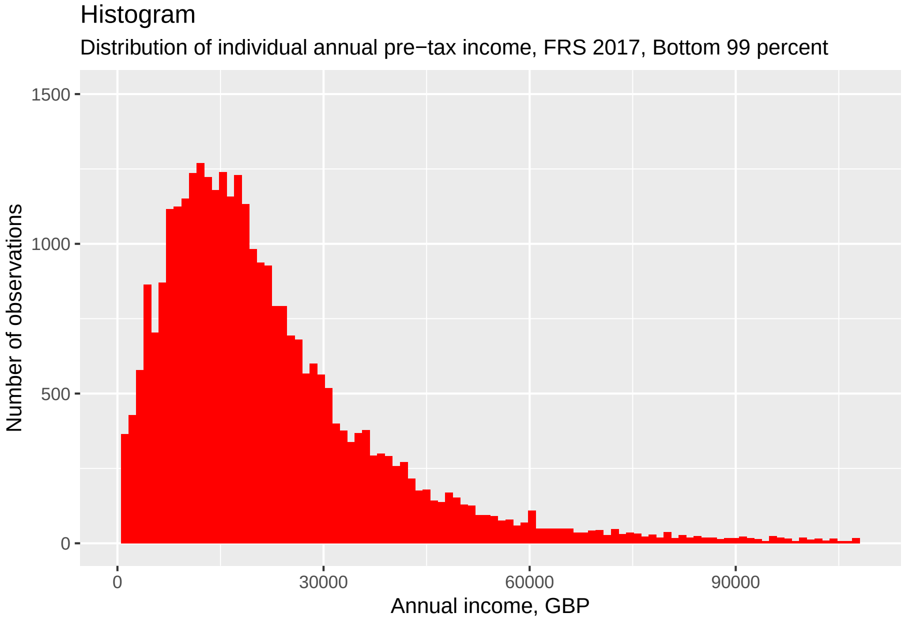
```
ggplot(frs) +
  geom_histogram(aes(x=income), bins = 100, fill="red") +
  xlab("Annual income, GBP") + ylab("Number of observations") +
  labs(title = "Histogram", subtitle = "Distribution of individual annual pre-tax inco
```



Note that by default the histogram is plotted over the whole support of `income` in the sample (i.e. from £0 to £885768). This makes the graph look somewhat peculiar - it looks as if there are no observations with income above £375000 but this is misleading as in fact there are (we know one observation has £885768) but the corresponding bars are so small that they are effectively invisible.

To focus on the part of the distribution where most of the mass lies, we can restrict the horizontal axis to the 99th quantile of the distribution only

```r
ggplot(frs) +
  geom_histogram(aes(x=income), bins = 100, fill="red") +
  xlab("Annual income, GBP") + ylab("Number of observations") +
  xlim(0,quantile(frs$income,0.99)) +
  labs(title = "Histogram", subtitle = "Distribution of individual annual pre-tax income, FRS 201
```



representing the income distribution within the bottom 99 percent of individuals by income only.

Two features of the distribution are immediately clear: 1. the distribution is highly disperse - there are large income differences between individuals. 2. the distribution is very strongly positively skewed - characterised by a long and thin right tail, with most observations found in the extreme left of the distribution.

It is clear that income is very unequally distributed. One implication of this is that when we use measures of central tendency (such as mean or median income) to quantify living standards within a country (for example, in measures such as GNI or GDP per capita) we should recognise the fact that they do not necessarily account for "typical" incomes - relatively few individuals earn income very close to the mean or the median.

Some important aspects of inequality are reflected in standard summary statistics. For example, as we observed above, mean income (£22751) considerably exceeds median income (£17576), which is a feature the observed asymmetry (skewness) of the distribution. Exactly 50% of individuals of the data receive income below £17576 (the median) but a much larger fraction of individuals receive income well below the mean.

To gain further quantitative understanding of the unequal distribution of income, it is informative to calculate some quantiles

```
quantile(frs$income, seq(0.1, 0.9, by=0.1))
```

```
##   10%   20%   30%   40%   50%   60%   70%   80%   90% 
##  4264  8476 11544 14508 17576 20956 25324 31824 43368
```

10% of individuals in the data receive less than £4264 pr year; 20% receive less than £8476 and so forth. One thing that we observe is that the fraction of people receiving less than average income is more than 60% (but less than 70%). Comparisons between different quantiles is very informative about both dispersion and assymetry (and as we will see later is the basis of some popular quantitative measures of income inequality). In particular, we observe large differences between quantiles (indicatig disperison) with differences increasing at higher incomes (indicating assymetry).

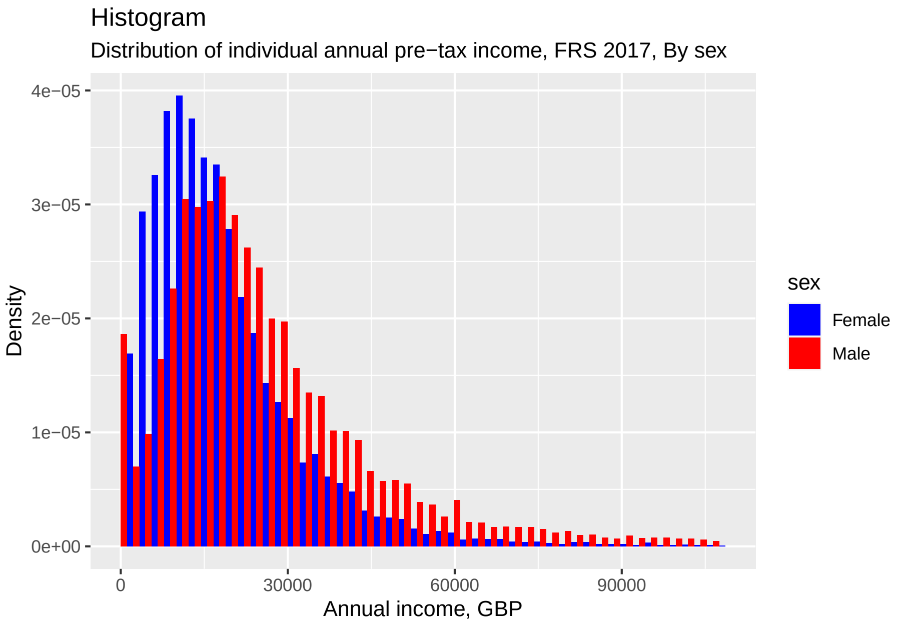
4.1.3 Income differences between and within groups of individuals

The above discussion referred to overall differences in income between individuals. Important aspects of inequality are about differences between (and within) groups. This section illustrates such differences on the basis of the categorical variables observed in the data: `sex`, `age`, `educ`.

4.1.3.1 Inequality between sexes

It is well known that there are systematic income differences between men and women. To explore these within this data, start by plotting separate histograms by sex:

```
ggplot(frs, aes(x=income, fill=sex,  y = ..density..)) +
  geom_histogram(bins = 50, position="dodge") +
  scale_fill_manual(values=c("blue", "red")) +
  xlab("Annual income, GBP") + ylab("Density") +
  xlim(0,quantile(frs$income,0.99)) +
  labs(title = "Histogram", subtitle = "Distribution of individual annual pre-tax incon
```



It can be seen that the income distributions for men and women are different and, in particular, it appears that the distribution for men is shifted to the right relative to the distribution for women (for example, relative to men, women are more likely to be in the left tail of the overall distribution, and less likely in the right tail).

To get a more quantitative (if less rich) description of the differences, we can obtain a set of summary statistics (mean, median, 10th percentile, 90th percentile) for income by sex:

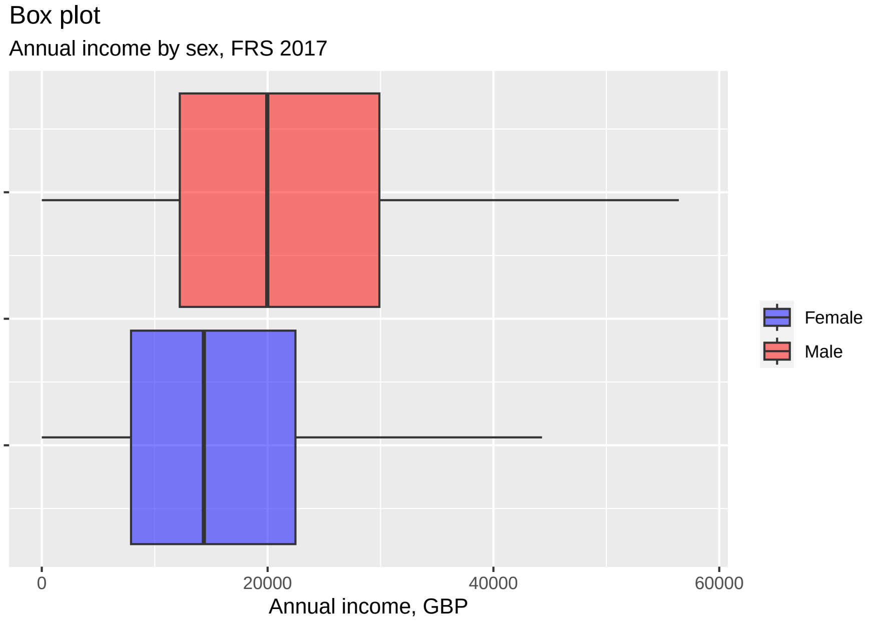
```
frs %>% group_by(sex) %>% summarise(mean=mean(income), median = median(income), q10 = quantile(in
```

```
## # A tibble: 2 x 5
##   sex      mean median   q10    q90
##   <fct>   <dbl>  <dbl> <dbl>  <dbl>
## 1 Female 18038.  14664 3318. 35266.
## 2 Male   27992.  21320 7020  52416
```

Average income for men in the sample (£27992) far exceeds average income for women (£18037) and similarly for medians, 10th percentile and 90th percentile. For example, 90 percent of women receive less than £35266 while 90% of mean receive less than £52416.

As discussed before, a convenient way of presenting key summary statistics (25th percentile, median, 75th percentile, interquartile range) is through a box plot:

```
ggplot(frs,aes(y=income, fill=sex))+
  geom_boxplot(alpha=0.5, outlier.shape = NA) + ylim(0,quantile(frs$income,0.95)) +
  scale_fill_manual(values=c("blue", "red")) +
  xlab("") + ylab("Annual income, GBP") +
  labs(title = "Box plot", subtitle = "Annual income by sex, FRS 2017", fill="") +
  theme(axis.text.y = element_blank())+ coord_flip()
```



Indeed we observe systematic differences in the distribution on income between men and women. It should be noted, however, that this accounts for only small part of the overall income inequality as the distributions of incomes within groups are still very unequal.

4.1.3.2 Income differences between age groups

It is similarly well known income varies systematically over the life cycle, and some of the observed overall differences in incomes reflects differences in individuals' age.

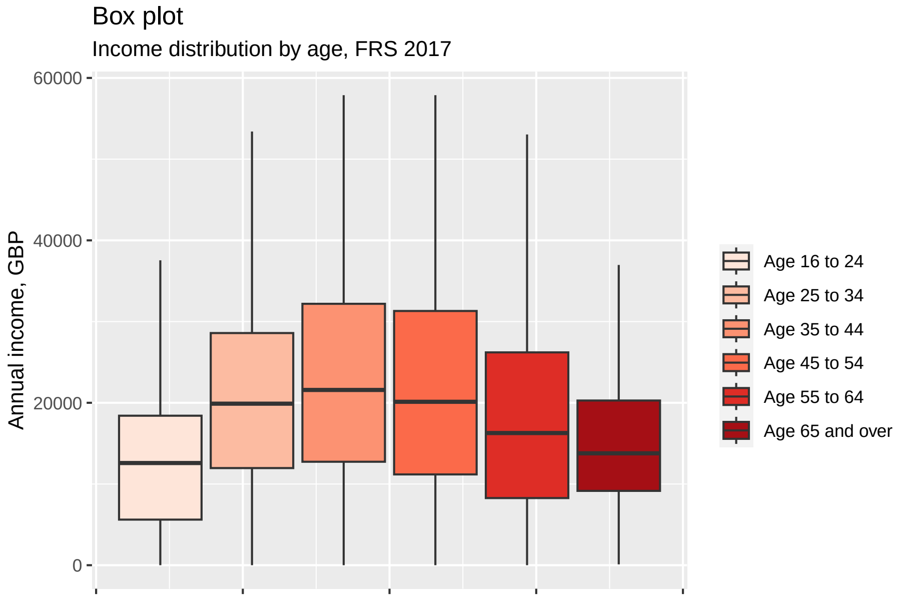
Given that we have 6 age groups, histograms will tend to be somewhat uninformative, but as before we can obtain a set of summary statistics for income by age

```
frs %>% group_by(age) %>% summarise(mean=mean(income), median = median(income), q10 = 
```

```
## # A tibble: 6 x 5
##   age                mean median   q10    q90
##   <fct>              <dbl>  <dbl> <dbl>  <dbl>
## 1 Age 16 to 24     12800.  12584    0   23972
## 2 Age 25 to 34     22968.  20436 4004   40300
## 3 Age 35 to 44     29057.  23244 5356   54168.
## 4 Age 45 to 54     29444.  22100 5184.  56784
## 5 Age 55 to 64     22972.  17290 1784.  45053.
## 6 Age 65 and over  17385.  13936 6084   31252
```

or use a box plot

```
ggplot(frs,aes(y=income, fill=age))+
  geom_boxplot(outlier.shape = NA) + ylim(0,quantile(frs$income,0.95)) +
  xlab("") + ylab("Annual income, GBP") +
  labs(title = "Box plot", subtitle = "Income distribution by age, FRS 2017", fill="") +
  scale_fill_brewer(palette = "Reds") +
  theme(axis.text.x = element_blank())
```



The age profile of income over age probably reflects life-cyclical patterns of labour income, where income increases on average as people progress through their careers, and then declines as they enter retirement. This clarifies that there are systematic income differences by age. It should be noted, however, that this accounts for only small part of the overall income inequality as distributions of incomes within age groups are still very unequal.

4.1.3.3 Income differences by education

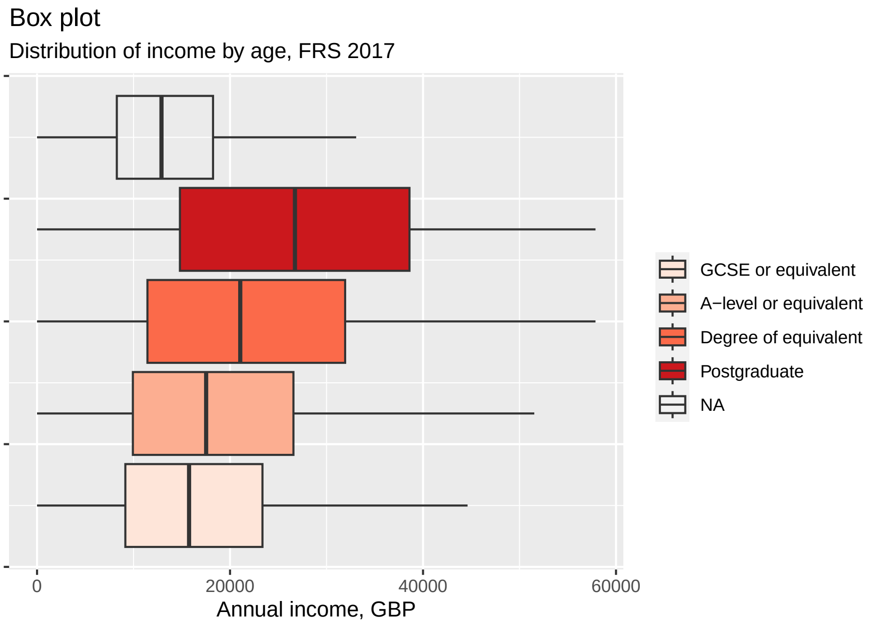
Repeating the analysis for individuals with different education in terms of summary statistics

```
frs %>% group_by(educ) %>% summarise(mean=mean(income), median = median(income), q10 =
```

```
## # A tibble: 5 x 5
##   educ                      mean median   q10    q90
##   <fct>                    <dbl>  <dbl> <dbl>  <dbl>
## 1 GCSE or equivalent      18717.  16068  4056 34216
## 2 A-level or equivalent 22405.  18148  4004 42390.
## 3 Degree of equivalent  28754.  22932  4992 54142.
## 4 Postgraduate          37994.  31096  5980 73585.
## 5 <NA>                  14641.  13000  4108 24752
```

and box plot

```
ggplot(frs,aes(y=income, fill=educ))+
  geom_boxplot(outlier.shape = NA) + ylim(0,quantile(frs$income,0.95)) +
  xlab("") + ylab("Annual income, GBP") +
  labs(title = "Box plot", subtitle = "Distribution of income by age, FRS 2017", fill=
  scale_fill_brewer(palette = "Reds") +
  theme(axis.text.y = element_blank())+ coord_flip()
```



we observe that more education associates strongly with higher income, likely reflecting returns to education. For example, the median income among those with postgraduate degree (£31096) is just slightly below the 80th percentile of the overall income distribution (£31824), while the median for those with GCSE or equivalent (£16068) is just slightly above the 40th percentile of the overall income distribution (£14508). Again, while we observe systematic income differences between education groups, there is also significant income inequality within groups.

4.1.4 Quantitative measures of income inequality

While histograms allow us to represent the distribution of income in all its complexity, we are often interested in comparing the extent of inequality in different income distributions. For example: - we might be interested in comparing different countries by income inequality. This amounts to comparing the whole distribution of income in Country A to the distribution of income in Country B. - we might be interested in analyzing how income inequality changes over time in a given country. This amounts to comparing the distribution of income in Year A to the distribution of income in Year B.

As comparisons between histograms could be difficult (as observed) it would be more appropriate to use numerical measures of inequality instead. A numerical measure of inequality, is one that summarizes some aspect of the inequality into a single number. As inequality is a complex phenomenon, there are many different numerical measures, which emphasise different aspects of inequality. While a thorough discussion of different metrics is beyond the scope of the document, this section briefly outlines several popular metrics and illustrates their meaning through the data.

4.1.4.1 Percentile ratios

We have already seen that comparison between different percentiles of the distribution is informative about important aspects of inequality. Percentile ratios (or p-ratios) are simply ratios of percentiles. Two common percentile ratios used to quantify inequality of income are the P50P10 ratio (the ratio between median income and the 10th percentile of income) and the P90P50 ratio (the ratio between the 90th percentile of income and median income).

For example, for the whole data, we can calculate the 10th, 50th and 90th percentiles, and the corresponding percentile ratios as follows

```
frs %>% summarise(q10 = quantile(income, 0.1), q50 = median(income),  q90 = quantile(income, 0.9)
```

```
## # A tibble: 1 x 5
##     q10   q50   q90 p90p50 p50p10
##   <dbl> <dbl> <dbl>  <dbl>  <dbl>
## 1  4264 17576 43368   2.47   4.12
```

We observe that those at the 90th percentile of the distribution have 2.47 times higher income than those at the median; while the latter have 4.12 times more income than those at the 10th percentile.

To show how p-ratios can be used to compare inequality across distributions, we can calculate the p-ratios for the income distributions within education groups in the data:

```
frs %>% group_by(educ) %>% summarise(q10 = quantile(income, 0.1), q50 = median(income)
```

```
## # A tibble: 5 x 6
##   educ                     q10   q50    q90 p90p50 p50p10
##   <fct>                  <dbl> <dbl>  <dbl>  <dbl>  <dbl>
## 1 GCSE or equivalent      4056 16068 34216    2.13   3.96
## 2 A-level or equivalent   4004 18148 42390.   2.34   4.53
## 3 Degree of equivalent    4992 22932 54142.   2.36   4.59
## 4 Postgraduate            5980 31096 73585.   2.37   5.2
## 5 <NA>                    4108 13000 24752    1.90   3.16
```

We have already seen that overall average and median incomes increase with education. However, we can now see that the distributions within groups become increasingly unequal with higher education.

4.1.4.2 Measures of concentration

So far we have discussed two important features of the distribution of income - dispersion and asymmetry. Another important feature, which is somewhat distinct from the former, is that income distributions tend to be highly *concentrated*. Loosely, this means that relatively small fractions of the overall population receive relatively large shares of the total income. Some of the most popular metrics of income inequality (shares of top 10% or top 1%, GINI coefficient, etc) are in fact measures of concentration.

4.1.4.2.1 Lorenz curve To understand measures of concentration, suppose that we order all individuals in the data by increasing order of their income

```
frs <- frs %>% arrange(income)
```

Calculate the quantile rank of each individual (i.e., the percentile of the overall distribution corresponding to the particular individual's income)

```
frs <- frs %>% mutate(inc.qrank = row_number(income)/length(income))
```

Now suppose that we move from the poorest to the richest individual, and for each we record the total income received by herself and those poorer, in a new variable, `cuminc` (for cumulative income)

```
frs$cuminc <- cumsum(frs$income)
```

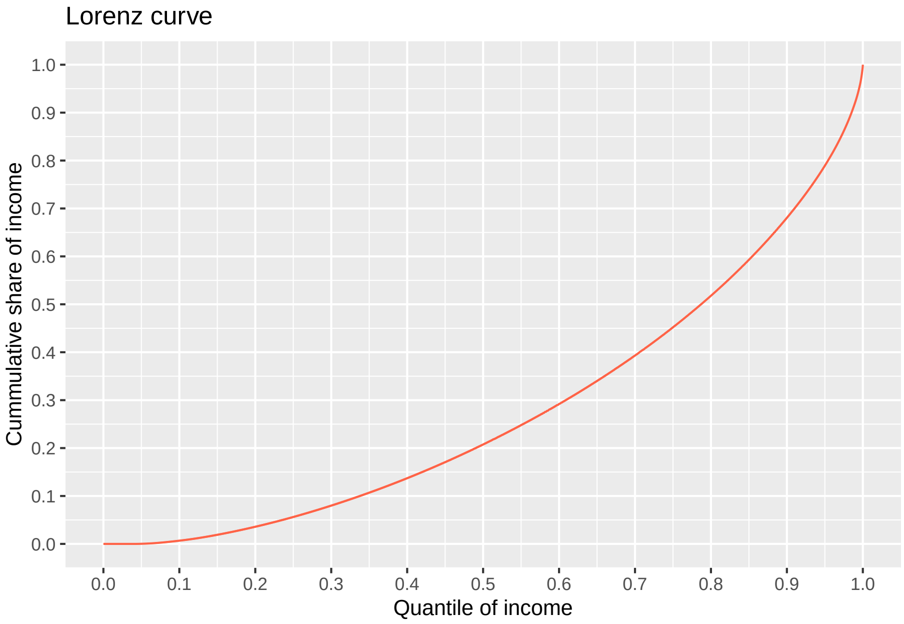
Next, divide `cuminc` by the total amount of income earned by individuals in the data

```
frs$cumincshare <- frs$cuminc/sum(frs$income)
```

to obtain the cumulative income share at each quantile rank.

The mapping from quantile rank to cumulative income share is known as a the Lorenz curve. The Lorenz curve for income in the FRS data can be represented graphically as follows

```
ggplot(frs) +
  geom_line(aes(x=inc.qrank, y=cumincshare), col="tomato") +
  scale_x_continuous(breaks = seq(0,1, by=0.1)) +
  scale_y_continuous(breaks = seq(0,1, by=0.1)) +
  xlab("Quantile of income") +
  ylab("Cummulative share of income") +
  labs(title = "Lorenz curve")
```



From the Lorenz curve, we can observe that, for example - the bottom 50 % of individuals by income receive around 20% of all the income in the sample. Therefore, the top 50% receive around 80%. - the bottom 90% of individuals receive around 68% of all the income. Therefore, the top 10 % receive the remaining 32%. - the top 5% receive around 21% of all income - and so forth.

4.1.4.2.2 Top income shares The points on the Lorenz curve (known as Lorenz coordinates) are the basis of many popular measures of income inequality (and more specifically income concentration). Two common measures are as follows:

-the share of income of the top 10 %. We can obtain this by finding the share of the bottom 90 %

```
lorenz_fun <- approxfun(x=frs$inc.qrank, y=frs$cumincshare)
lorenz_fun(0.9)
```

```
## [1] 0.6804861
```

implying that the share of the top 10% in the FRS is approximately 32% of all income. It should be noted that given the design of FRS (sampling only one household from an address with multiple households) results from the sample are not necessarily representative of the whole UK population. Nonetheless, the numbers above seem to be in line with estimates from administrative data for the UK as a whole.

- the share of income of the top 1 $. We can obtain this by finding the share of the bottom 99 %

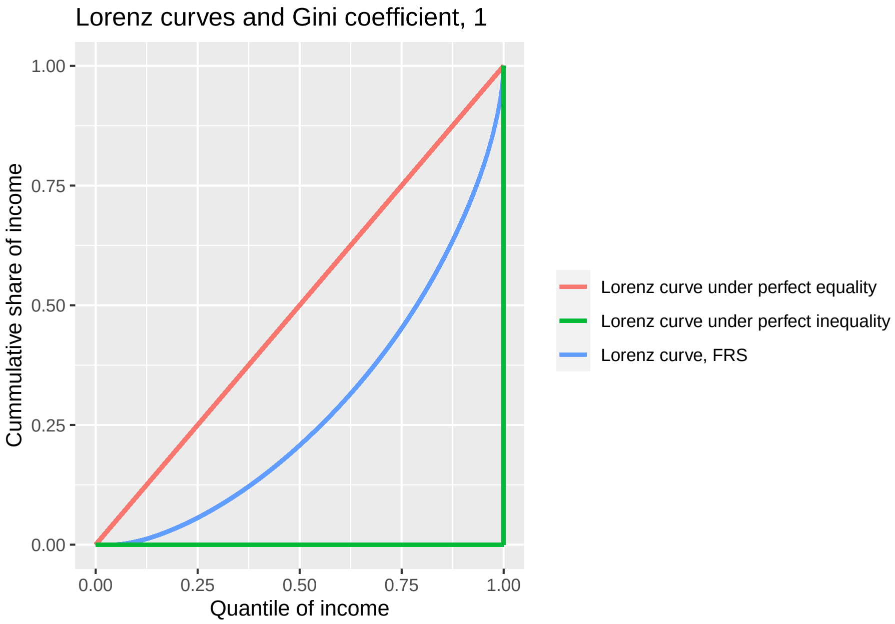
```
lorenz_fun(0.99)
```

```
## [1] 0.9218844
```

implying that the share of the top 1% in the FRS is approximately 9.2% of all income. This seems to be slightly lower than estimates from administrative data for the UK as a whole, possibly due to underrepresentation of richest individuals in the survey.

4.1.4.2.3 Gini coefficient Probably the best known measure of income inequality, the Gini coefficient, is in fact a measure of concentration related to the Lorenz curve. The figure below plots the actual empirical Lorenz curve for the FRS (in blue) together with - a counterfactual Lorenz curve for an economy with "perfect equality" (in red). If all individuals in an economy receive the same income (perfect equality), and are ordered arbitrarily along the horizontal axis, then the "poorest" (arbitrarily defined) n% of the population, would be receiving n% of the total income. The Lorenz curve maps n to n and is therefore a straight line with slope 45 degrees. - a counterfactual Lorenz curve for an economy with "perfect inequality" (in green). If all but one individuals in the economy receive no income, and one receives some (and therefore all the income), then the income share of the "all but one" will be 0 (implying horizontal portion of the Lorenz curve over $[0, 1)$) and a discontinuous jump at 1.

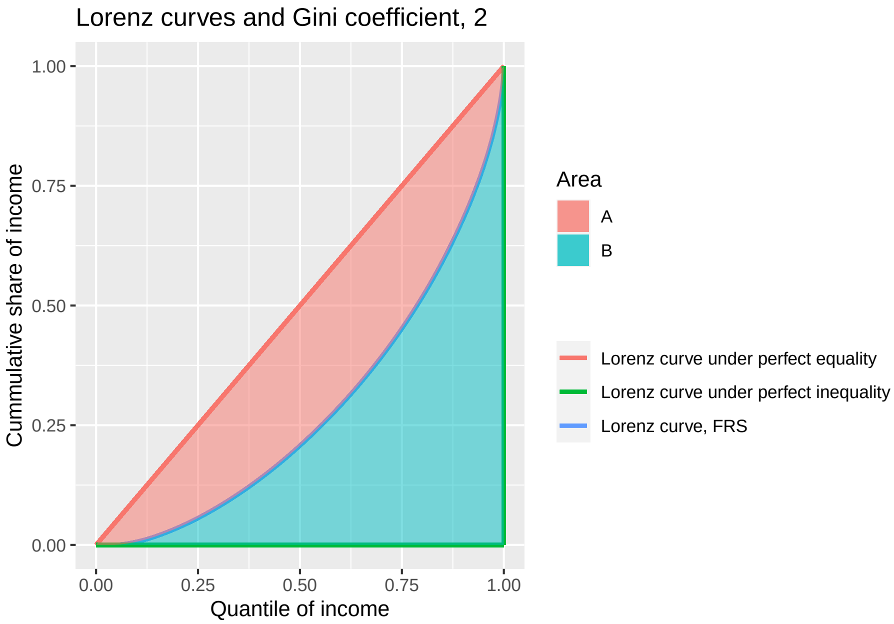
```
ggplot(frs) +
  geom_line(aes(x=inc.qrank, y=cumincshare, col="Lorenz curve, FRS"), size=1) +
  geom_segment(x=0, xend=1, y=0, yend=1,
               aes(col="Lorenz curve under perfect equality"), size=1) +
  geom_segment(x=0, xend=1, y=0, yend=0,
               aes(col="Lorenz curve under perfect inequality"), size=1) +
  geom_segment(x=1, xend=1, y=0, yend=1,
               aes(col="Lorenz curve under perfect inequality"), size=1) +
  xlab("Quantile of income") +
  ylab("Cummulative share of income") +
  labs(title = "Lorenz curves and Gini coefficient, 1", color = "")
```



Intuitively, the closer is the actual Lorenz curve to the perfect equality one, the more equal is the distribution; the closer to the perfect inequality one, the more unequal is the distribution. The Gini coefficient measures exactly how close the actual Lorenz curve is to the perfect inequality line and more specifically is defined as follows:

Let A denote the area between the actual Lorenz curve and the perfect equality line, and B denote the area between the perfect inequality line and the actual Lorenz curve:

```
ggplot(frs) +
  geom_line(aes(x=inc.qrank, y=cumincshare, col="Lorenz curve, FRS"), size=1) +
  geom_segment(x=0, xend=1, y=0, yend=1, aes(col="Lorenz curve under perfect equality"), size=1)
  geom_segment(x=0, xend=1, y=0, yend=0, aes(col="Lorenz curve under perfect inequality"), size=1
  geom_segment(x=1, xend=1, y=0, yend=1, aes(col="Lorenz curve under perfect inequality"), size=1
  geom_ribbon(aes(x=inc.qrank, ymin = cumincshare, ymax = inc.qrank, fill = "A"), alpha=0.5) +
  geom_ribbon(aes(x=inc.qrank, ymin = 0, ymax = cumincshare, fill = "B"), alpha = 0.5) +
  ylab("Cummulative share of income") +
  labs(title = "Lorenz curves and Gini coefficient, 2", color = "", fill = "Area")
```



Then the Gini coefficient is defined as

$$Gini = \frac{A}{A+B}$$

Note that since $A + B = 1/2$, the Gini coefficient is also equal to $2A$ or $1 - 2B$. It is easy to see that the value of the Gini coefficient is a number between 0 and 1, with 0 indicating perfect equality, and 1 indicating perfect inequality.

One way to calculate the Gini coefficient in our data is to obtain the area B as the definite integral of the Lorenz curve from 0 to 1. Integrating the Lorenz curve numerically yields

```
B <- integral(lorenz_fun, 0,1)
gini = 1-2*B
gini
```

```
## [1] 0.4410313
```

so the Gini coefficient in the FRS data is 0.44. This seems in line with estimates from administrative data for income from all sources before taxes. Unlike other numerical measures, the scale of the Gini coefficient is difficult to interpret in itself. However, as all numerical measures it is useful for comparing the inequality of different distributions, e.g., between countries, or within a country over time, as we see below.

```r
# Clear R environment
rm(list = ls())
```

4.2 Cross-country differences in income inequality

The World Inequality Database provides open access to the most extensive available database on the historical evolution of the world distribution of income and wealth, both within countries and between countries.

This section uses cross sectional data of countries on a set of measures of income inequality in 2017. The data comes from Alvaredo and Saez (2022), and is available here. For ease of replication, the data has been saved in the sheet `WID17` of the `data_ch3.xlsx` Excel file.

Load the data and print first few rows

```r
wid2017 <- read_xlsx("data_ch3.xlsx", sheet = "WID2017")
wid2017$Country <- as.factor(wid2017$Country)
head(wid2017)
```

```
## # A tibble: 6 x 8
##   Country    Year  gini   s10    s1  s0.1 p90p50 p50p10
##   <fct>     <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>  <dbl>
## 1 Argentina  2017 0.581  46.4  13.9  2.98   3.44  NA
## 2 Australia  2017 0.488  34.4  10.0  3.12   2.53   7.87
## 3 Belarus    2017 0.433  33.0  10.1  3.17   2.25   4.62
## 4 Belgium    2017 0.436  31.8   8.5  2.27   2.33   5.41
## 5 Brazil     2017 0.682  58.4  20.9  5.35   4.20  NA
## 6 Bulgaria   2017 0.549  44.3  19.6  9.10   2.75   5.46
```

The dataset contains observations for 48 countries in 2017, on 6 aggregate measures of inequality in pre-tax income. - `gini` is the Gini coefficient - `s10`, `s1`, and `s0.1` are the income shares of the top 10%, 1%, and 0.1% - `p90p50` and `p50p10` are the P90P50 and P50P10 percentile ratios.

To get a sense of the data, report a set of summary statistics

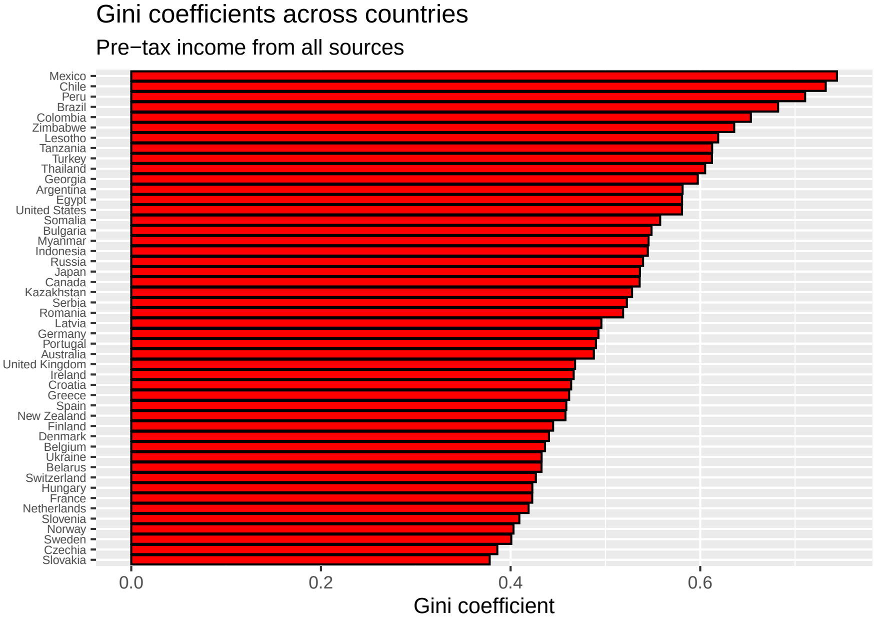
```r
summary(wid2017)
```

```
##       Country          Year           gini             s10              s1
##  Argentina: 1   Min.   :2017   Min.   :0.3779   Min.   :27.47   Min.   : 7.05
##  Australia: 1   1st Qu.:2017   1st Qu.:0.4394   1st Qu.:33.18   1st Qu.:10.50
##  Belarus  : 1   Median :2017   Median :0.5072   Median :37.73   Median :12.86
##  Belgium  : 1   Mean   :2017   Mean   :0.5192   Mean   :40.84   Mean   :14.35
##  Brazil   : 1   3rd Qu.:2017   3rd Qu.:0.5809   3rd Qu.:46.89   3rd Qu.:17.65
##  Bulgaria : 1   Max.   :2017   Max.   :0.7442   Max.   :64.19   Max.   :27.06
##  (Other)  :42
##       s0.1            p90p50          p50p10
##  Min.   : 1.640   Min.   :1.676   Min.   :4.578
##  1st Qu.: 3.397   1st Qu.:2.237   1st Qu.:5.080
##  Median : 4.660   Median :2.656   Median :5.403
##  Mean   : 4.915   Mean   :2.926   Mean   :5.589
##  3rd Qu.: 5.575   3rd Qu.:3.452   3rd Qu.:5.903
##  Max.   :12.670   Max.   :6.179   Max.   :8.298
##                                   NA's   :8
```

It is immediately observed, that there are large differences in the values of all income-inequality metrics across countries. For example, the income share of the top 0.1 % of individuals varies from 1.6% to 12.67% across countries.

Given the relatively small number of observations, a convenient way to represent cross country differences is through ordered bar charts. Arranging countries in increasing order of Gini coefficients, and representing the value of each country's Gini coefficient by the height of the corresponding bar yields

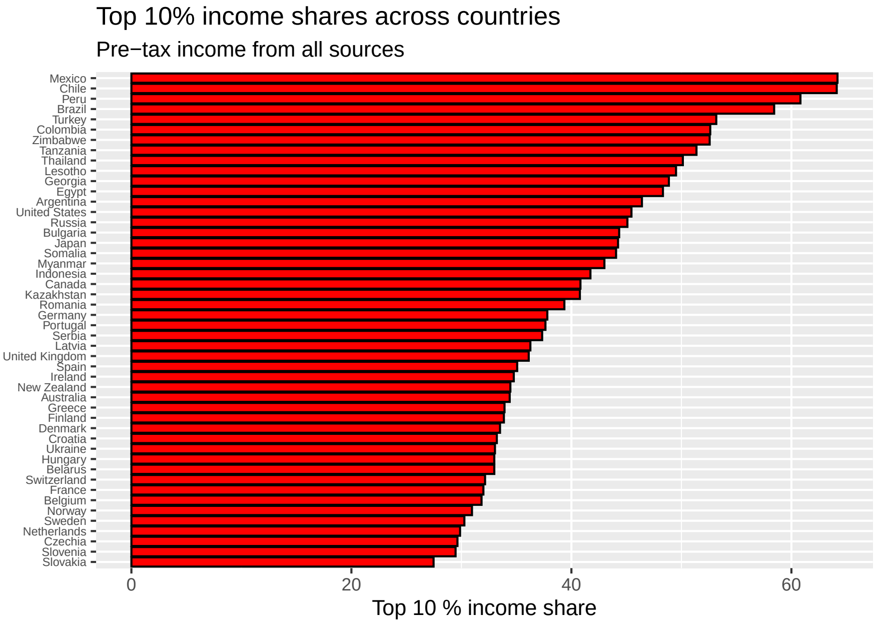
```r
ggplot(wid2017) +
  geom_bar(aes(x=fct_reorder(Country, gini), y=gini), stat="identity", fill="red", col
  theme(axis.text.y= element_text(size=6)) +
  xlab("") + ylab("Gini coefficient") +
  labs(title = "Gini coefficients across countries",
       subtitle = "Pre-tax income from all sources") +
  coord_flip()
```



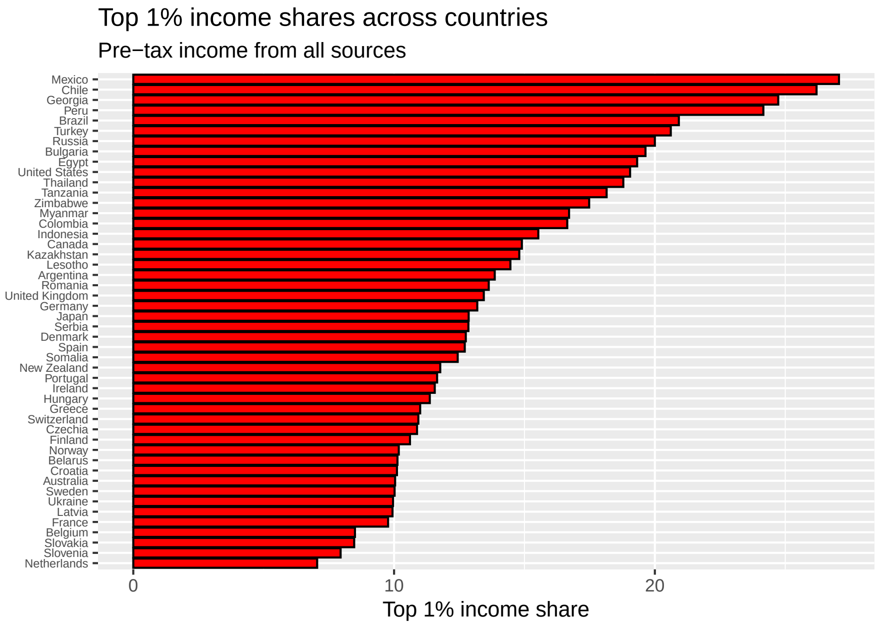
We observe large cross-country differences in terms of Gini coefficients. Within the sample, in terms of Gini coefficients, Slovakia has the most equal income distribution (0.378) and Mexico has the most unequal income distribution (0.744).

Producing similar graphs for the other metrics

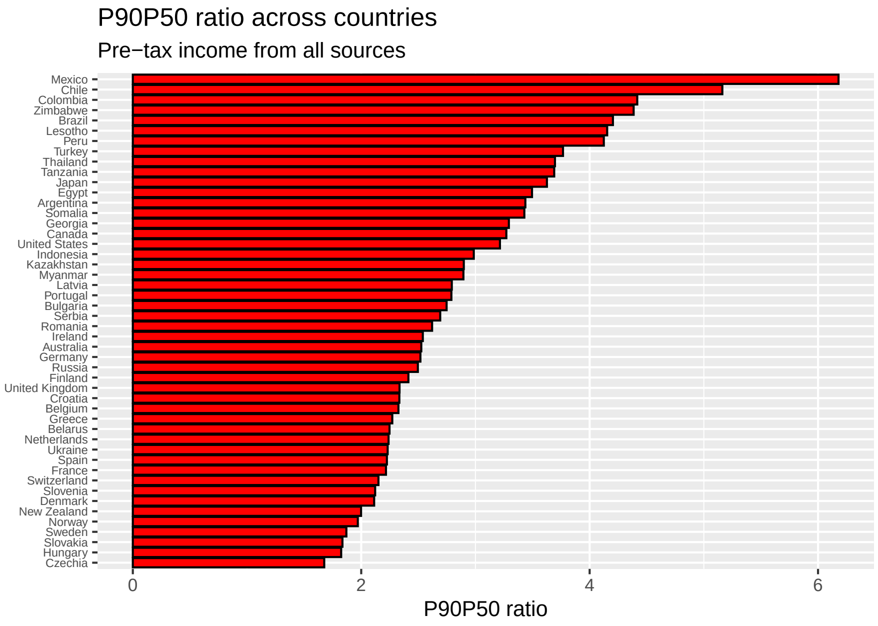
```r
ggplot(wid2017) +
  geom_bar(aes(x=fct_reorder(Country, s10), y=s10), stat="identity", fill="red", col="black") +
  theme(axis.text.y= element_text(size=6)) +
  xlab("") + ylab("Top 10 % income share") +
  labs(title = "Top 10% income shares across countries",
       subtitle = "Pre-tax income from all sources") +
  coord_flip()
```



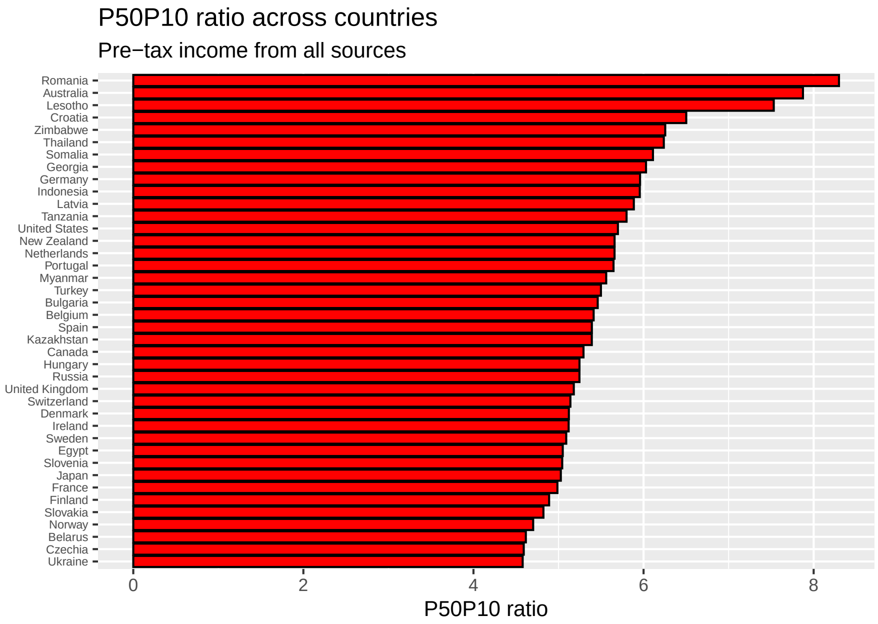
```r
ggplot(wid2017) +
  geom_bar(aes(x=fct_reorder(Country, s1), y=s1), stat="identity", fill="red", col="black") +
  theme(axis.text.y= element_text(size=6)) +
  xlab("") + ylab("Top 1% income share") +
  labs(title = "Top 1% income shares across countries",
       subtitle = "Pre-tax income from all sources") +
  coord_flip()
```



```
ggplot(wid2017) +
  geom_bar(aes(x=fct_reorder(Country, p90p50), y=p90p50), stat="identity", fill="red",
  theme(axis.text.y= element_text(size=6)) +
  xlab("") + ylab("P90P50 ratio") +
  labs(title = "P90P50 ratio across countries",
       subtitle = "Pre-tax income from all sources") +
  coord_flip()
```



```
ggplot(subset(wid2017, !is.na(p50p10))) +
  geom_bar(aes(x=fct_reorder(Country, p50p10), y=p50p10), stat="identity", fill="red", col="black
  theme(axis.text.y= element_text(size=6)) +
  xlab("") + ylab("P50P10 ratio") +
  labs(title = "P50P10 ratio across countries",
       subtitle = "Pre-tax income from all sources") +
  coord_flip()
```

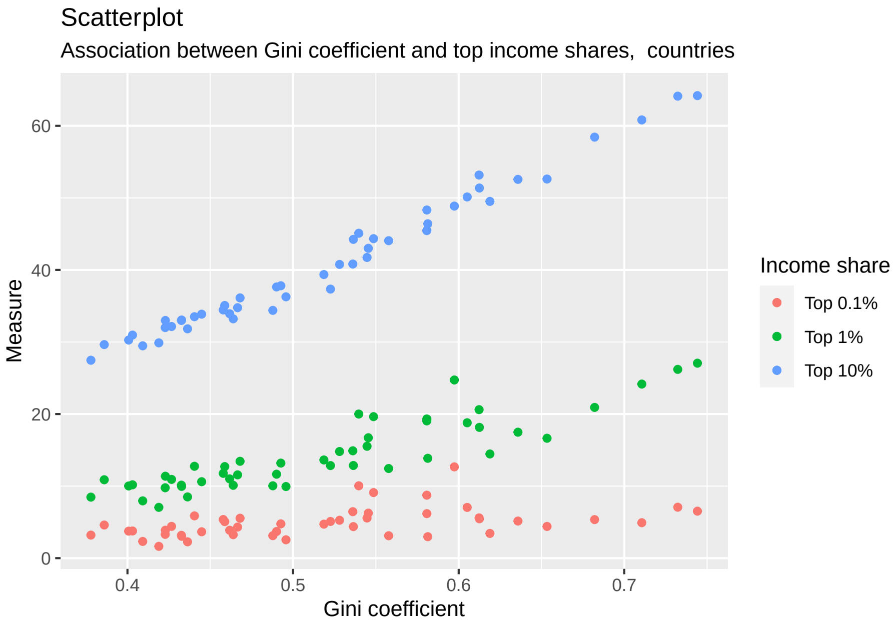


we observe that there are significant differences in terms of all different measures. Of course, different measures capture different aspects of inequality (e.g., Gini coefficients and top income shares in particular are measures of concentration, while the percentile ratios measure specific aspects of dispersion).

Nonetheless, inspecting the graphs, it seems countries with high inequality in one measure, tend to also exhibit high inequality in terms of the other measures. To inspect this relationship more clearly we next present scatterplots showing the association netween Gini coefficients and other metrics.

In terms of Gini coefficient and top income shares

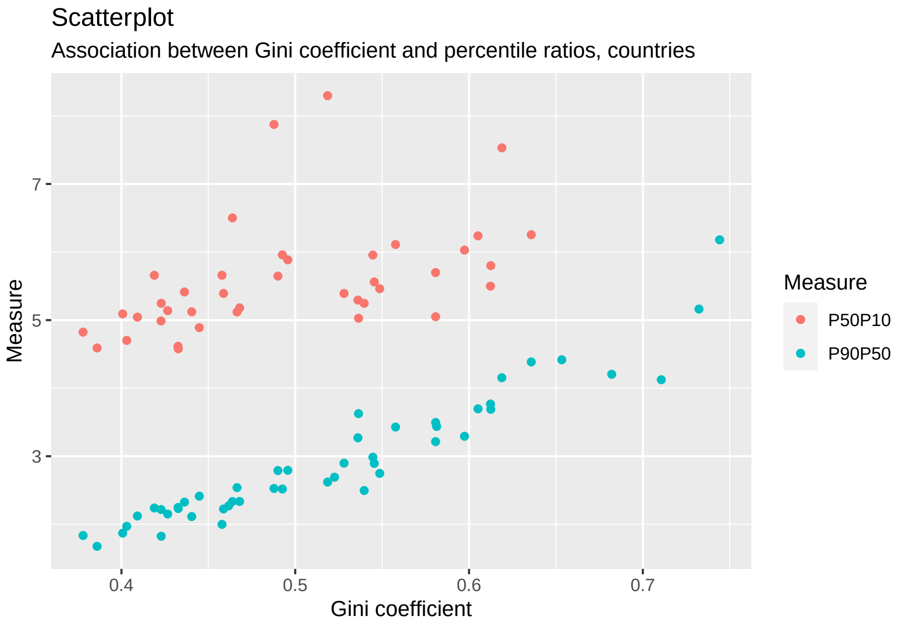
```
ggplot(wid2017, aes(x=gini)) +
  geom_point(aes(y=s10, col="Top 10%")) +
  geom_point(aes(y=s1, col="Top 1%")) +
  geom_point(aes(y=s0.1, col="Top 0.1%")) +
  xlab("Gini coefficient") +
  ylab("Measure") +
  labs(title = "Scatterplot", subtitle = "Association between Gini coefficient and top
```



unsurprisingly, we observe very strong positive association - countries with high Gini coefficients tend to also have high top income shares. Of course, this is not surprising as both Gini coefficients and top income shares are fundamentally measures of concentration, capturing similar aspects of income distributions within countries.

Perhaps more surprisingly, we observe also very strong association between Gini coefficients and percentile ratios:

```r
ggplot(wid2017, aes(x=gini)) +
  geom_point(aes(y=p50p10, col="P50P10")) +
  geom_point(aes(y=p90p50, col="P90P50")) +
  xlab("Gini coefficient") +
  ylab("Measure") +
  labs(title = "Scatterplot", subtitle = "Association between Gini coefficient and percentile rat
```



In general, the data shows that there are large and systematic differences in income inequality across countries.

4.3 Income inequality over time

This section uses a panel dataset for several large developped economies, tracking measures of income inequality from the begining of the 20th century, until 2022. As in the previous section, the data comes from Alvaredo and Saez (2022) and is available here The . For ease of replication, the data has been saved in the sheet `WIDpanel` of the `data_ch3.xlsx` Excel file.

Load the data and print first few rows

```
panel <- read_xlsx("data_ch3.xlsx", sheet = "WIDpanel")
panel$Country <- as.factor(panel$Country)
head(panel)
```

```
## # A tibble: 6 x 6
##   Country    Year  gini   s10    s1  s0.1
##   <fct>     <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Australia  1912 0.557  42.4  15.9  6.97
## 2 Australia  1913 0.532  39    14.6  6.34
## 3 Australia  1914 0.515  36.7  13.7  6
## 4 Australia  1915 0.499  34.5  13.0  5.78
## 5 Australia  1916 0.465  29.9  11.2  5.05
## 6 Australia  1917 0.470  30.6  11.5  5.19
```

We have a panel data in long form, with records of the Gini coefficient and Top 10%, Top 1% and Top 0.1 % income shares, for each country in each year, from 1912 until 2021.

The set of countries included

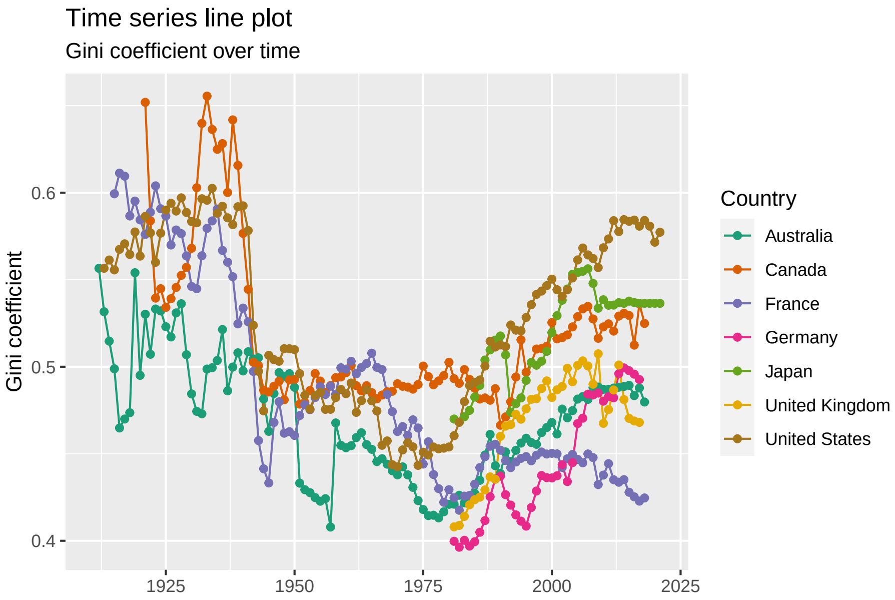
```
levels(panel$Country)
```

```
## [1] "Australia"      "Canada"         "France"         "Germany"
## [5] "Japan"          "United Kingdom" "United States"
```

are large developed economies.

Plotting time series plots of Gini coefficient across time, for each country yields

```
ggplot(panel, aes(x=Year, y=gini, col=Country)) +
  geom_line() +
  geom_point() +
  scale_color_brewer(palette = "Dark2") +
  xlab("") + ylab("Gini coefficient") +
  labs(title = "Time series line plot", subtitle = "Gini coefficient over time")
```



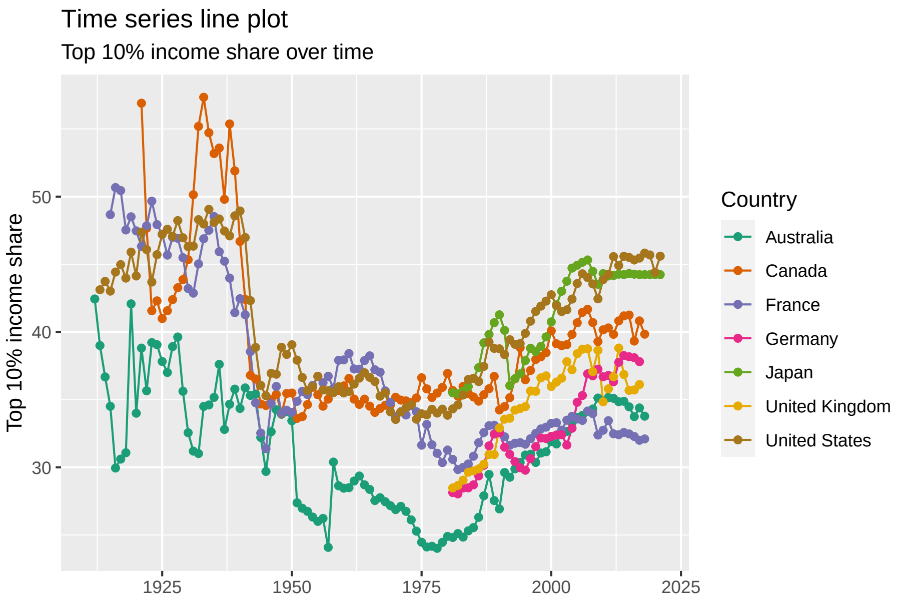
We can see that data on Gini coefficient for Australia, Canada, France and US are available since 1912, but for the rest data only becomes available late in the second half of the 20th century.

While there are some differences across countries, the overall pattern can be summarized as follows: - income inequality (measured by Gini coefficient) was relatively high in the beginning of the 20th century and remained elevated until the 1930s-1940s, when it decreased substantially. - inequality remained relatively low for the next several decades, starting to pick up again at various points in the 1970s. - since then inequality has been overall increasing.

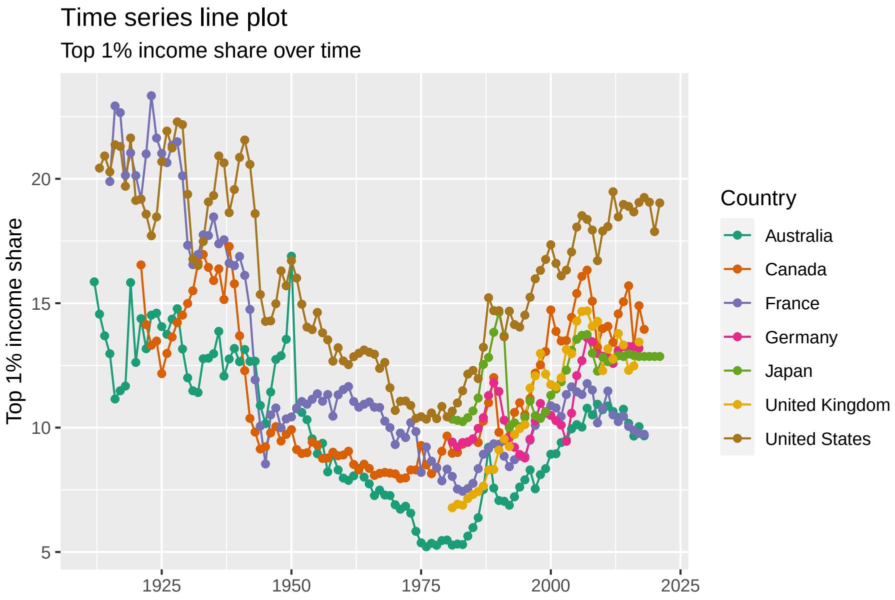
This pattern is well known and documented. While various explanations for it have been proposed, a discussion is beyond the scope of this document.

However, something that we can check in our data is whether the evolution of inequality in terms of Gini coefficients is similar to the evolution in terms of other measures. Unsurprisingly, it turns out that top income shares have followed a similar pattern:

```r
ggplot(panel, aes(x=Year, y=s10, col=Country)) +
  geom_line() +
  geom_point() +
  scale_color_brewer(palette = "Dark2") +
  xlab("") + ylab("Top 10% income share") +
  labs(title = "Time series line plot", subtitle = "Top 10% income share over time")
```



```r
ggplot(panel, aes(x=Year, y=s1, col=Country)) +
  geom_line() +
  geom_point() +
  scale_color_brewer(palette = "Dark2") +
  xlab("") + ylab("Top 1% income share") +
  labs(title = "Time series line plot", subtitle = "Top 1% income share over time")
```



In summary, we observe a common pattern across a set of large developped economies, where income inequality has followed a U-shaped pattern over time for the last 100 years or so. At present, measures of inequality are elevated relative to the norm of the middle of the 20th century.

Alvaredo, P., Atkinson, Saez, 2022. World inequality database.

Global Carbon Budget, 2022a. Annual CO2 emissions.

Global Carbon Budget, 2022b. Annual CO2 emissions.

MEaSUREs, 2023. Global mean sea level trend from integrated multi-mission ocean altimeters TOPEX/poseidon, jason-1, OSTM/jason-2, and jason-3 version 5.1. https://doi.org/10.5067/GMSLM-TJ151

Met Office, 2023a. Data for the regional average mean temperature with trends 1884 - 2022.

Met Office, 2023b. Regional average climate observations, UK annual rainfall.

NASA/JPL, 2019. JPL GRACE and GRACE-FO mascon ocean, ice, and hydrology equivalent water height release 06 version 02. https://doi.org/10.5067/TEMSC-3MJ62

Office for National Statistics, Social and Vital Statistics Division, Department for Work and Pensions, National Centre for Social Research, 2023. Family resources survey, 2016-2017. https://doi.org/http://doi.org/10.5255/UKDA-SN-8336-1

R Core Team, 2023. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.

Rohde, R.A., Hausfather, Z., 2020. The berkeley earth land/ocean temperature record. Earth System Science Data 12, 3469–3479. https://doi.org/10.5194/essd-12-3469-2020

Tans and Keeling, 2023. Trends in atmospheric carbon dioxide.

Wickham, H., 2016. ggplot2: Elegant graphics for data analysis. Springer-Verlag New York.

World Bank, 2023. World development indicators. https://doi.org/https://doi.org/10.57966/6rwy-0b07