Reading and writing files and the concept of Working Directory

Silvie Cinková

2025-07-24

Table of contents

1 Working directory

- folder from which R sees other files and folders

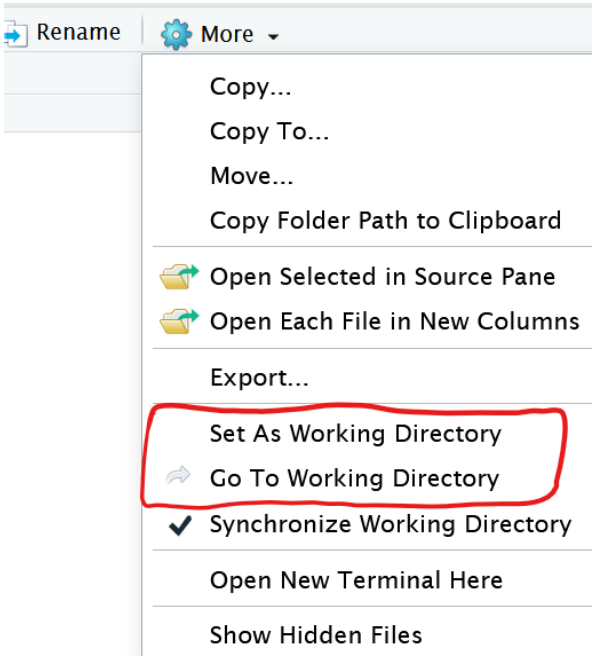

Figure 1: Interactive control of Working Directory location

Print path to your current Working Directory

```
getwd()
```

```
[1] "/lnet/aic/personal/cinkova/R_BEGINNERS_SHORT"
```

Set a different Working Directory

```
setwd("~/folder/subfolder/") # ~ means your home
```

2 Let's have a common file path

1. Make sure that your Working directory is your home.
2. Create a new folder in your home. Call it `R_BEGINNERS_SHORT`.

   Enter that folder. Make it your Working Directory. (Gear icon → `Set As Working Directory`).
3. Create new folders `datasets_ATRIUM` and `my_output_files`.

In the `File` tab, select `New project` → `In an existing directory`, and pick `R_BEGINNERS_SHORT`.

If you execute this procedure, you will not need to adapt file paths in the teaching materials to your user account, except perhaps the user account name.

3 RStudio Projects

- `.Rproj` file stores project configuration
- When you open this project next time, it tries to restore the workspace from last time.

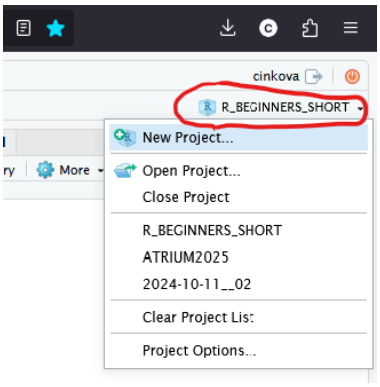

Figure 2: Project List in RStudio

4 Download a file from (GitHub) URL

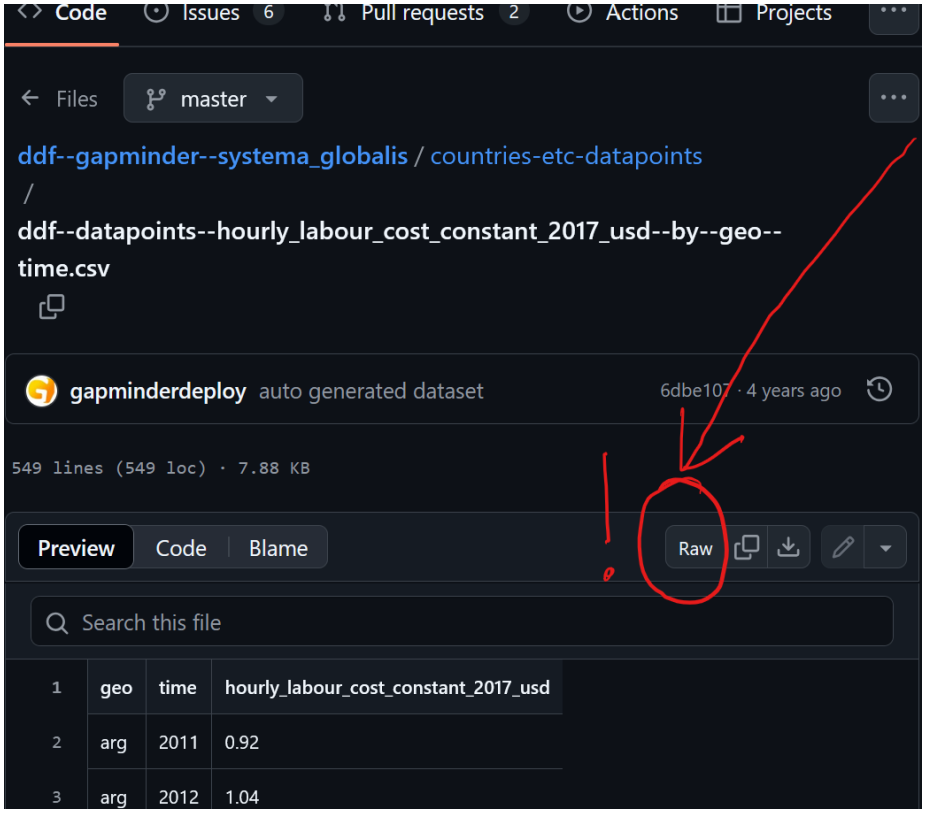

Figure 3: GitHub default view

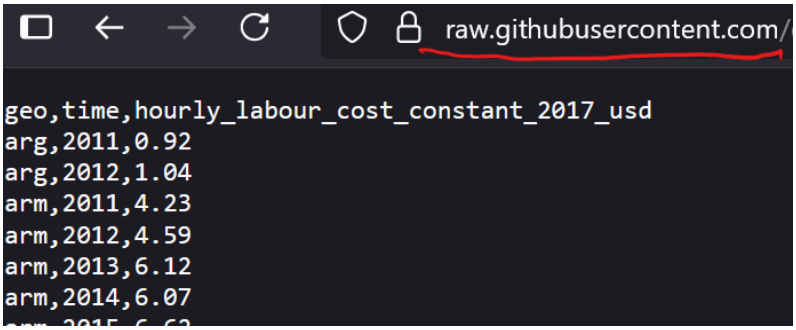

Figure 4: Switched to raw file URL

```
library(glue) # enables multiline with \\
URL <- glue(
 ↪  "https://raw.githubusercontent.com/open-numbers/ddf--gapminder--\\

 ↪  systema_globalis/refs/heads/master/countries-etc-datapoints/ddf--
 ↪  \\

 ↪  datapoints--hourly_labour_cost_constant_2017_usd--by--geo--time.csv"
 ↪  )
my_destination <- glue("datasets_ATRIUM/\\
gapminder_hourly_labour_cost_constant_2017_\\
                        usd--by--geo--time.csv")
download.file(
  url = URL,
  destfile = my_destination
  )
```

The `download.file` function is universal to download any file from anywhere. Sometimes you can copy a download link from a website and use this URL to download the file programmatically.

This is how to download some data from GitHub, which is a bit specific. Here I work with data from Gapminder on Github. Their repository is very large and this was a largely random pick: `https://github.com/open-numbers/ddf-gapminder-systema_globalis/tree/master/countries-etc-datapoints`. This repository contains a table that explains each data set, but I am going to select one that is intelligible without reading much metadata. It is going to be a table about average labor cost in a given country in a given year: https://raw.githubusercontent.com/open-numbers/ddf–gapminder–systema_globalis/refs/heads/master/countries-etc-datapoints/ddf–datapoints–hourly_labour_cost_constant_2017_usd–by–geo–time.csv.

Manually navigate to the file you want and copy its URL. Mind to use the URL that appears when you hit the `Raw` button (starting with `https://raw.githubusercontent`) to download the contents of the file. On the default `https://github.com/….` you would only download a html file of the website you are seeing.

Use the `download.file` function. Leave all arguments at default, except `url` and `destfile`. Put the file into the new empty `datasets_ATRIUM` folder. Use the end part of the original file name and give it a prefix `gapminder_` and keep doing this with all files that you happen do download from this source. This will help you keep a system in your files.

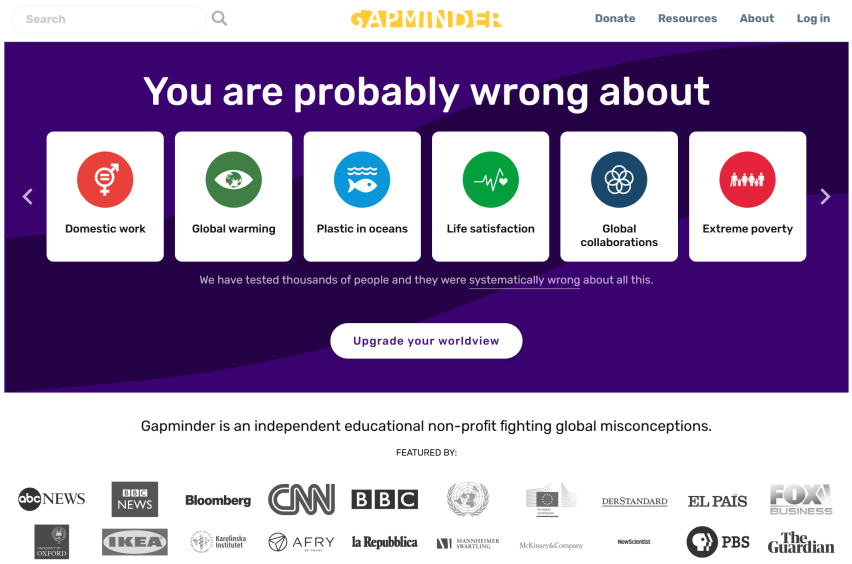

Figure 5: Gapminder

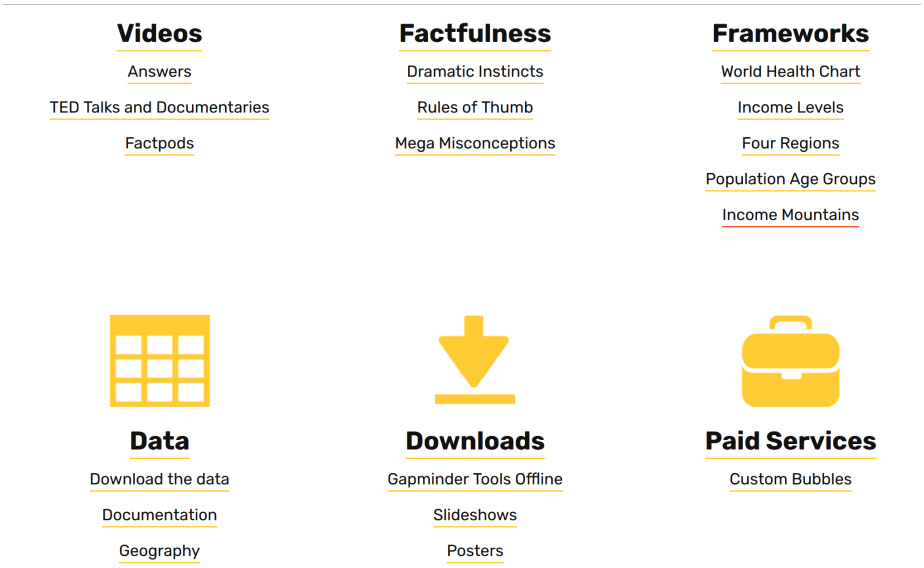

Figure 6: Introducing Gapminder

5 https://www.gapminder.org/

6 Read a `.csv`/`.tsv` file

- plain text with column separators: `;` `,` or tabulator
- inspect the file reading it as text (first 3 lines)

```
mypath <- glue(
 ↪  "datasets_ATRIUM/gapminder_hourly_labour_cost_constant_2017_usd\\
               --by--geo--time.csv")
```

```
library(readr)
read_lines(
  file = mypath,
  n_max = 3)
```

```
[1] "geo,time,hourly_labour_cost_constant_2017_usd"
[2] "arg,2011,0.92"
[3] "arg,2012,1.04"
```

```
readLines(
  con = mypath,
  n = 3)
```

```
[1] "geo,time,hourly_labour_cost_constant_2017_usd"
[2] "arg,2011,0.92"
[3] "arg,2012,1.04"
```

What you are seeing are the first three lines of a tabular file we have just read as a text file, assuming no columns or headers. This comes handy when a file is too large to open interactively in a text editor, for instance.

A tabular file is a plaintext file where each line is one table row and the columns are on each line separated by the same character (throughout the file). The best-known tabular format is **comma-separated values** (`csv`). The original U.S. format uses comma. The European csv uses semicolons because comma is often reserved for the decimal operator (vs. decimal point in the U.S.). To skip these issues altogether, you better save your files as `tsv` (tab-separated values).

In the code above you see two functions that look similar and whose output looks exact the same. One is a base-R function, the other is from a `tidyverse` package called `readr`. Feel free to choose either and just make a mental note

that there is an alternative. Sometimes, when a file is tricky to read in with one function, it goes well with the other.

Look at the Help to either function and explore its other arguments using the file you have just loaded.

7 Reading a table with `readr`

- `read_csv`, `read_csv2`, `read_tsv`: tailored to the common separators `,`, `;`, tab
- `read_delim` : you name the separator (aka delimiter), more arguments

```
read_csv(file = mypath,
         n_max = 3) #just top 3 rows
```

```
Rows: 3 Columns: 3
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (1): geo
dbl (2): time, hourly_labour_cost_constant_2017_usd

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.

# A tibble: 3 x 3
  geo    time hourly_labour_cost_constant_2017_usd
  <chr> <dbl>                                <dbl>
1 arg    2011                                 0.92
2 arg    2012                                 1.04
3 arm    2011                                 4.23
```

8 Other arguments in `read_csv`

```
read_csv(file = mypath,
         col_names = c("country", "year", "USD_hour_2017"),
         n_max = 3)
```

```
Rows: 3 Columns: 3
-- Column specification --------------------------------------------------------
Delimiter: ","
```

```
chr (3): country, year, USD_hour_2017

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# A tibble: 3 x 3
  country year  USD_hour_2017
  <chr>   <chr> <chr>
1 geo     time  hourly_labour_cost_constant_2017_usd
2 arg     2011  0.92
3 arg     2012  1.04
```

9 Read directly from URL

```
URL2 <- glue(
 ↪  "https://raw.githubusercontent.com/open-numbers/ddf--gapminder--\\

 ↪  systema_globalis/refs/heads/master/countries-etc-datapoints/\\

 ↪  ddf--datapoints--hourly_labour_cost_constant_2017_usd--by--geo--\\
            time.csv")
read_csv(file = URL2,
         n_max = 3)
```

```
Rows: 3 Columns: 3
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (1): geo
dbl (2): time, hourly_labour_cost_constant_2017_usd

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# A tibble: 3 x 3
  geo    time hourly_labour_cost_constant_2017_usd
  <chr> <dbl>                                <dbl>
1 arg    2011                                 0.92
2 arg    2012                                 1.04
3 arm    2011                                 4.23
```

10 Download an Excel file

```
URL3 <- glue("https://docs.google.com/spreadsheets/d/1qHalit8s\\
             XCOR8oVXibc2wa2gY7bkwGzOybEMTWp-08o/export?format=xlsx")
download.file(url = URL3,
              destfile =
                "datasets_ATRIUM/gapminder_geonames.xlsx",
              mode = "wb")
```

With Windows formats and on Windows-operated computers, set `mode` to `wb`. Otherwise the file may get corrupted during the transmission.

11 Read Excel

- `readxl` reads only local file paths, not URLs.

```
library(readxl)
read_xlsx(path = "datasets_ATRIUM/gapminder_geonames.xlsx",
          n_max = 3) # just three rows
```

```
New names:
* `` -> `...2`
* `` -> `...3`
* `` -> `...5`

# A tibble: 3 x 7
  Data: Geographies - v~1 ...2  ...3  Free data from www.g~2 ...5  id    version
  <chr>                   <chr> <lgl> <chr>                  <lgl> <chr> <chr>
1 Updated: July 1, 2021   <NA>  NA    CC BY 4.0 LICENCE      NA    geo   v2
2 Concept:                Geog~ NA    Are you seeing this o~ NA    <NA>  <NA>
3 Unit:                   <NA>  NA    gapm.io/datageo        NA    <NA>  <NA>
# i abbreviated names: 1: `Data: Geographies - v2`,
#   2: `Free data from www.gapminder.org`
```

```
# readxl::read_xlsx(path =
 ↪  "datasets_ATRIUM/DataGeographies-v2-by-Gapminder.xlsx") #the same
 ↪  file
```

12 Excel sheets listed

- `read_xlsx` reads the first sheet by default

- Have the spreadsheets listed:

```
readxl::excel_sheets(path = "datasets_ATRIUM/gapminder_geonames.xlsx")
```

```
[1] "ABOUT"                 "list-of-countries-etc" "list-of-regions"
[4] "list-of-income-levels" "global"                "geo-names"
```

```
readxl::read_xlsx(path = "datasets_ATRIUM/gapminder_geonames.xlsx",
 ↪  sheet = 2,
                  n_max = 3) # or sheet = "list-of-countries-etc"
```

```
# A tibble: 3 x 13
  geo   name    four_regions eight_regions six_regions members_oecd_g77 Latitude
  <chr> <chr>   <chr>        <chr>         <chr>       <chr>                <dbl>
1 aus   Austra~ asia         east_asia_pa~ east_asia_~ oecd                 -25
2 brn   Brunei  asia         east_asia_pa~ east_asia_~ g77                    4.5
3 khm   Cambod~ asia         east_asia_pa~ east_asia_~ g77                   13
# i 6 more variables: Longitude <dbl>, `UN member since` <dttm>,
#   `World bank region` <chr>, `World bank, 4 income groups 2017` <chr>,
#   `World bank, 3 income groups 2017` <chr>, UNHCR <chr>
```

13 Google sheets

- inspect it manually and pick one worksheet

```
library(googlesheets4)
shURL <- glue("https://docs.google.com/spreadsheets/d/1qHalit8sXC\\

 ↪  0R8oVXibc2wa2gY7bkwGzOybEMTWp-08o/edit?gid=425865495#gid=425865495"
 ↪  )
gs4_deauth() # skip logging in at GoogleDrive
googlesheets4::read_sheet(shURL, sheet = 2,
                          n_max = 3)
```

```
v Reading from "Data Geographies - v2 - by Gapminder".
```

```
v Range ''list-of-countries-etc''.
```

```
# A tibble: 3 x 13
  geo   name    four_regions eight_regions six_regions members_oecd_g77 Latitude
  <chr> <chr>   <chr>        <chr>         <chr>       <chr>                <dbl>
1 aus   Austra~ asia         east_asia_pa~ east_asia_~ oecd                 -25
```

```
2 brn   Brunei  asia         east_asia_pa~ east_asia_~ g77                    4.5
3 khm   Cambod~ asia         east_asia_pa~ east_asia_~ g77                   13
# i 6 more variables: Longitude <dbl>, `UN member since` <dttm>,
#   `World bank region` <chr>, `World bank, 4 income groups 2017` <chr>,
#   `World bank, 3 income groups 2017` <chr>, UNHCR <chr>
```

14 Saving tabular files

```
gapminder_countries <-
 ↪  readxl::read_xlsx("datasets_ATRIUM/gapminder_geonames.xlsx",
                                          sheet = 2,
                                          n_max = 3)
readr::write_tsv(x = gapminder_countries,
                 file = "my_output_files/gapminder_countries.tsv")
```

15 Some file management functions

- create a file to save your exercise scripts

```
dir.create(path = "~/R_BEGINNERS_SHORT/my_exercise_scripts/",
           mode = '750', recursive = TRUE )
```

```
Warning in dir.create(path = "~/R_BEGINNERS_SHORT/my_exercise_scripts/", :
'/home/cinkova/R_BEGINNERS_SHORT/my_exercise_scripts' already exists
```

- list files in a folder
  - just those with `qmd` in their names
  - `recursive`: search in subfolders?

```
list.files(path = "~/R_BEGINNERS_SHORT", recursive = FALSE,
 ↪  include.dirs = FALSE, pattern = "qmd", full.names =
 ↪  TRUE)
```

```
[1] "/home/cinkova/R_BEGINNERS_SHORT/01_Introduction.qmd"
[2] "/home/cinkova/R_BEGINNERS_SHORT/02_HowToRStudio.qmd"
[3] "/home/cinkova/R_BEGINNERS_SHORT/03_RStudioFileManagement.qmd"
[4] "/home/cinkova/R_BEGINNERS_SHORT/04_NavigatingRStudioForProgramming.qmd"
[5] "/home/cinkova/R_BEGINNERS_SHORT/05_VariablesFunctions.qmd"
[6] "/home/cinkova/R_BEGINNERS_SHORT/06_WorkingDirectory.qmd"
[7] "/home/cinkova/R_BEGINNERS_SHORT/07_Exploring_dataframes.qmd"
[8] "/home/cinkova/R_BEGINNERS_SHORT/index.qmd"
```

`mode` = **octal notation** (access rights to file, just Unix)With `mode = '750'` you allow other students and teachers to see and execute files in this folder.