

Session 2 - Data wrangling

R training - Georgia RS-WB DIME

Luis Eduardo San Martin
The World Bank | September 2023

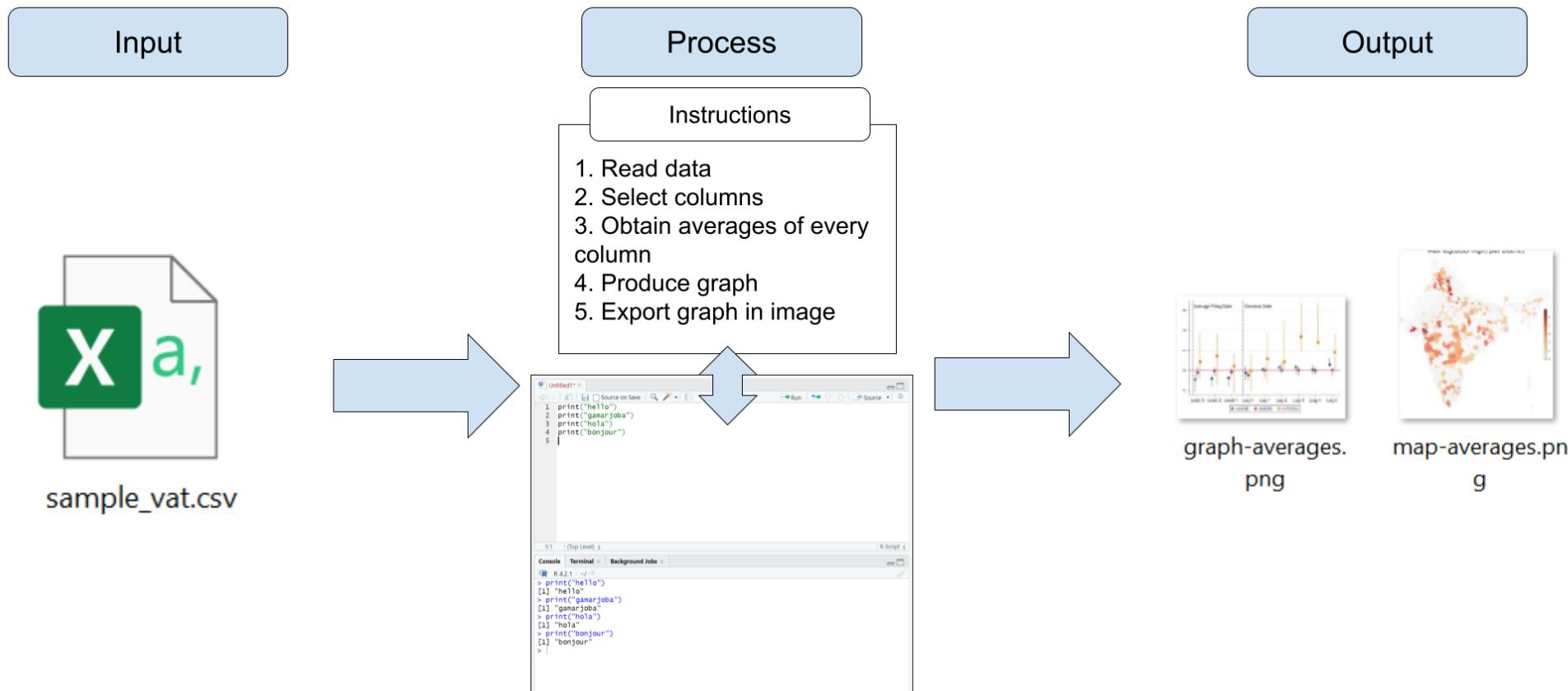


Table of contents // სარჩევი

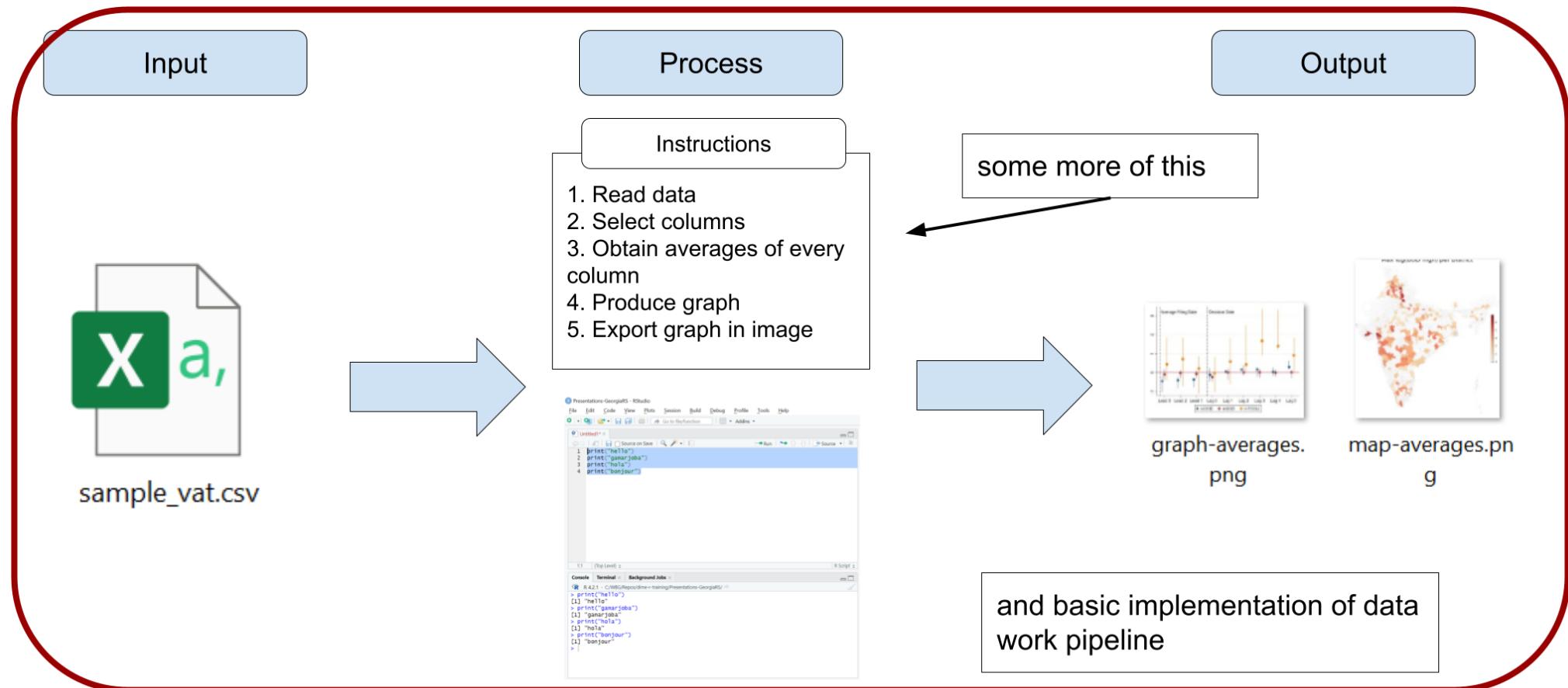
1. About this session
2. R libraries
3. Data wrangling
4. Filtering and sorting
5. Merging dataframes
6. Exporting outputs
7. Wrapping up
8. Appendix

About this session //

About this session // սօ Այլօօն ՋյօՏՏօջ



About this session // სა სესიონის შესახებ



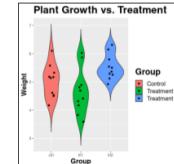
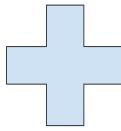
R libraries // R 3s308080

R libraries // R Յայութեան

- Installing R in your computer gives you access to its basic functions
- Additionally, you can also install libraries. Libraries are packages of additional R functions that allow you to do:
 - Operations that basic R functions don't do (example: work with geographic data)
 - Operations that basic R functions do, but easier (example: data wrangling)

R libraries // R 3s300øøø

In a nutshell:



Session 2 - Implementing the data work pipeline in R
R training - Georgia RS-WB DIME
Marc-Andrea Florina, Luis Eduardo San Martin
The World Bank | [WB GitHub](#)
April 2023

Basic R

Libraries

Enhanced R capabilities

R libraries // R 3s308080

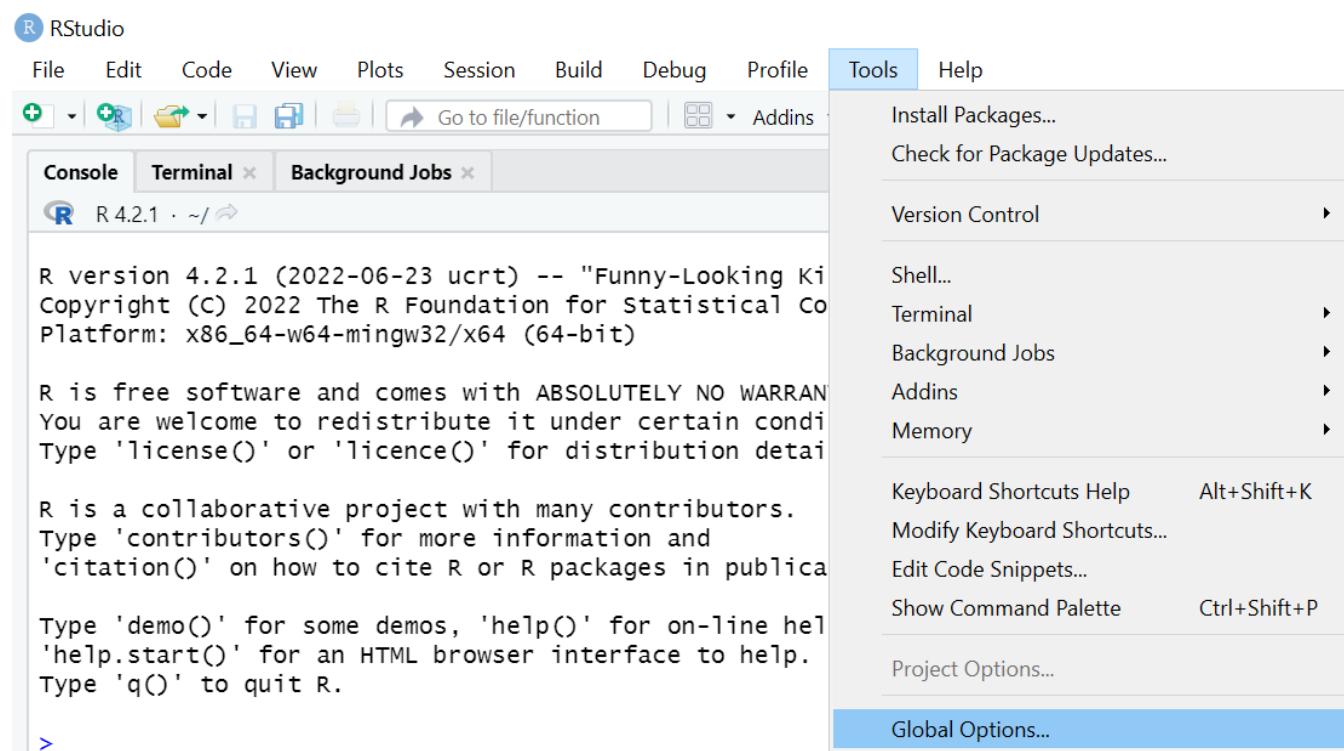
Installing R libraries

- Installing libraries is usually simple, but it can be challenging in institutional network connections such as the World Bank or the Georgia RS
- The next exercise will set up RStudio so that it can install R libraries without problems

R libraries // R 3s300800

Exercise 1: Setting up the installation of libraries

1 - In RStudio, go to **Tools** >> **Global Options...**



R libraries // R 3s300800

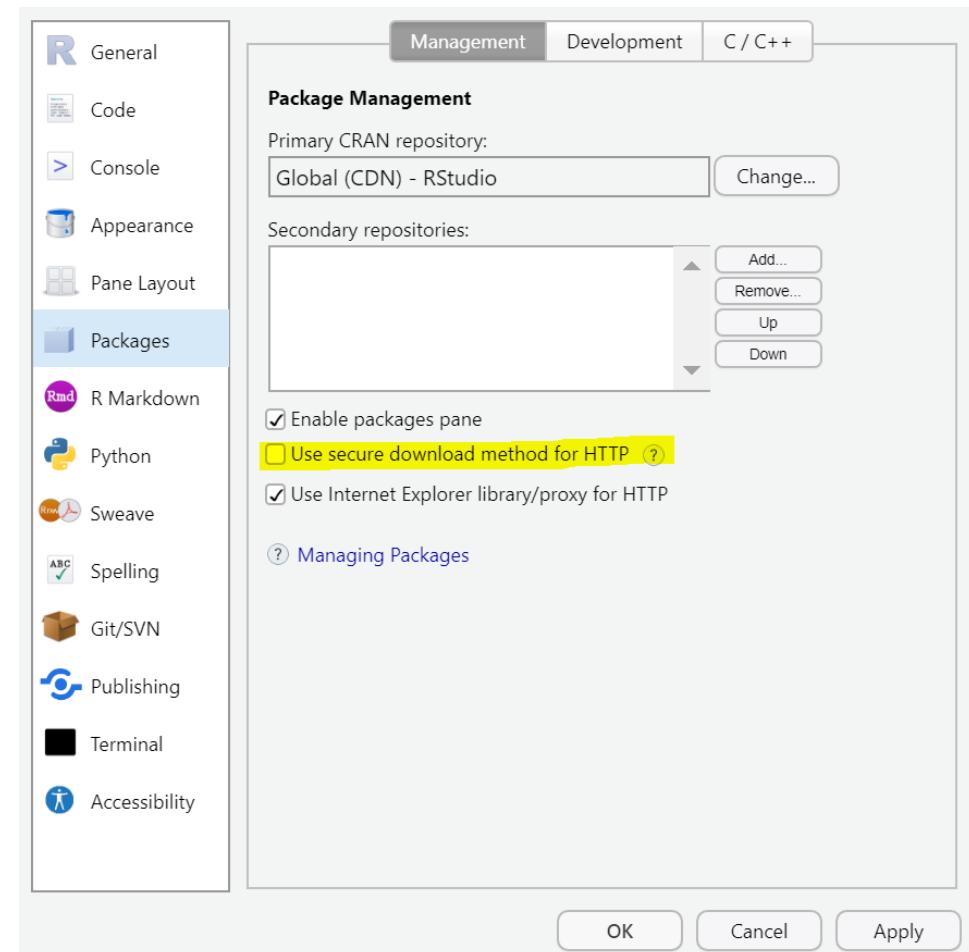
Exercise 1: Setting up the installation of libraries

2 - Select **Packages** in the left pane

3 - Uncheck **Use secure download method for HTTP**

4 - Click **OK**

You will not see any changes in your RStudio window after this, but now you'll be able to install libraries.



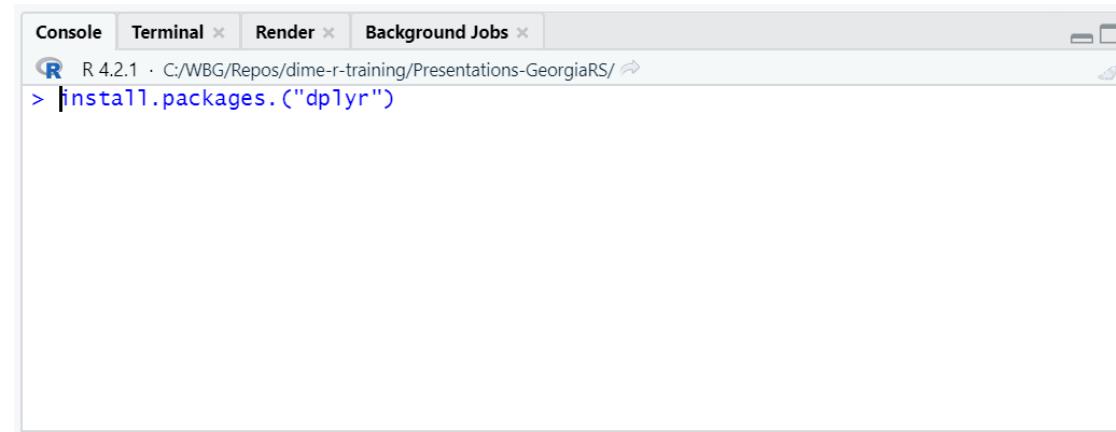
R libraries // R 3s308080

We'll use one library in today's session: `dplyr`

Exercise 2: Installing libraries

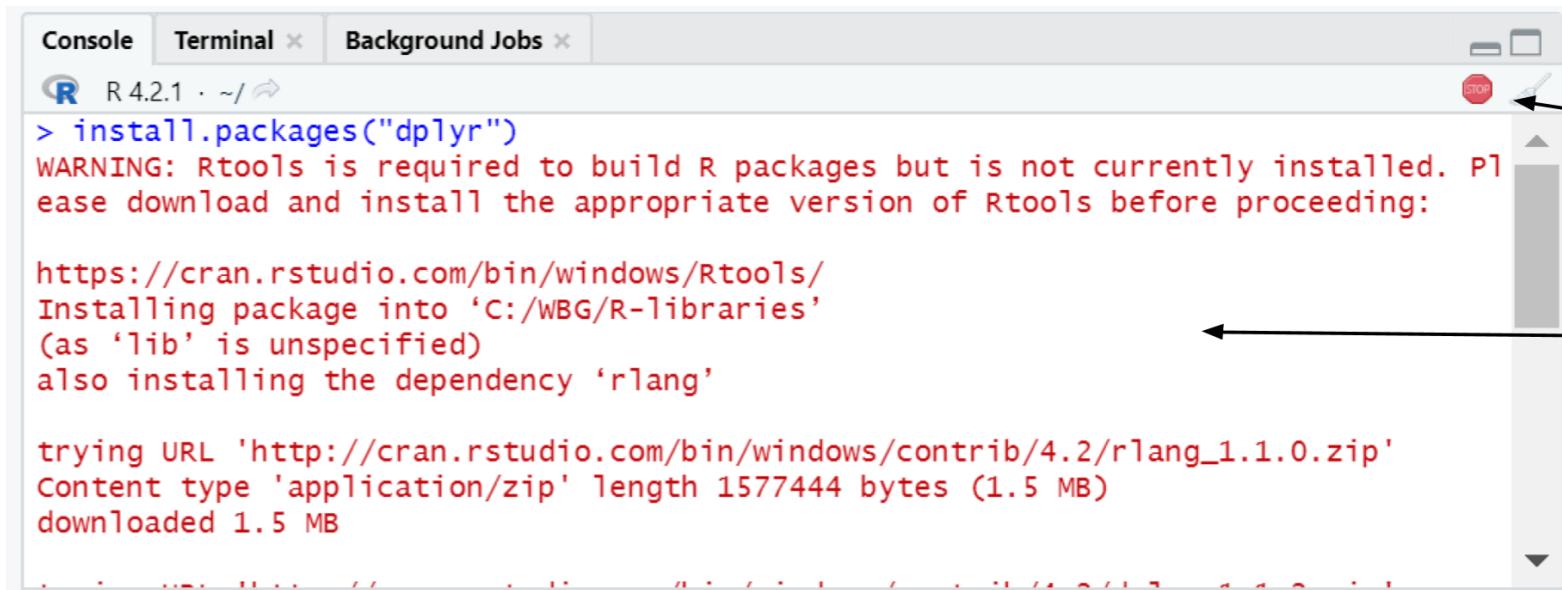
1. Install the libraries by using `install.packages()`

- `install.packages("dplyr")`
- Note the quotes (" ") in the packages names
- **Introduce this code in the console**, not the script panel



R libraries // R 3s308080

Installing libraries



The screenshot shows the RStudio interface with the 'Console' tab selected. The console window displays the following R code and its execution output:

```
R 4.2.1 · ~/ 
> install.packages("dplyr")
WARNING: Rtools is required to build R packages but is not currently installed. Please download and install the appropriate version of Rtools before proceeding:
https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/WBG/R-libraries'
(as 'lib' is unspecified)
also installing the dependency 'rlang'

trying URL 'http://cran.rstudio.com/bin/windows/contrib/4.2/rlang_1.1.0.zip'
Content type 'application/zip' length 1577444 bytes (1.5 MB)
downloaded 1.5 MB
```

The “STOP” sign means that the code is still running, just wait until it finishes

Note that this message is
not an error

R libraries // R 3s300gjbo

Now that `dplyr` is installed, we only need to load them to start using the functions they have.

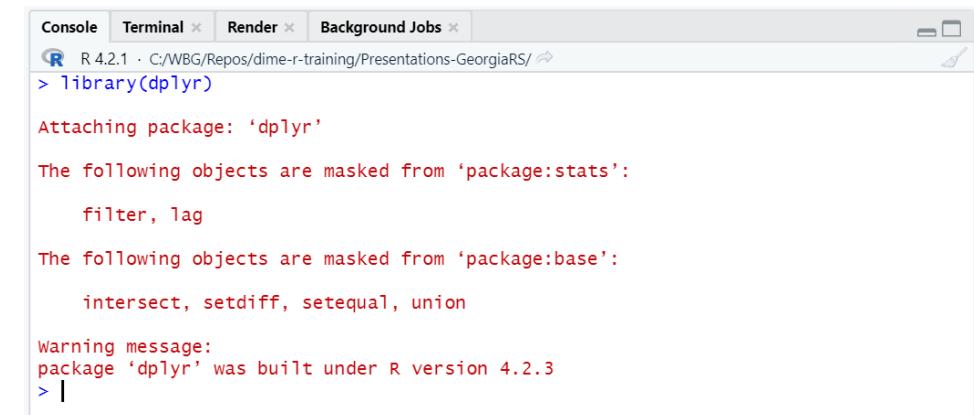
Exercise 3: Loading libraries

1. Open a new script with `File` >> `New File` >> `R`

`Script`

2. Load `dplyr` with: `library(dplyr)`

- Run this code from the new script you just opened
- Notice that we don't use quotes in the library names this time



The screenshot shows the RStudio interface with the 'Console' tab selected. The console window displays the following R session:

```
R 4.2.1 · C:/WBG/Repos/dime-r-training/Presentations-GeorgiaRS/
> library(dplyr)
Attaching package: 'dplyr'

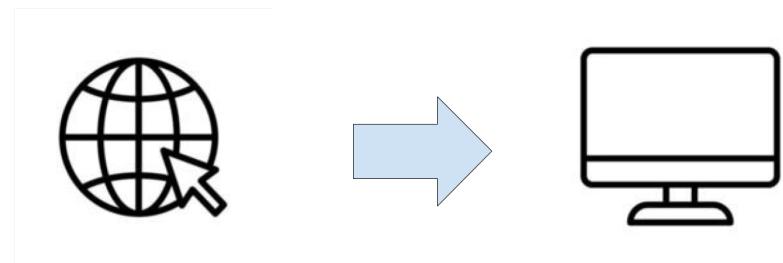
The following objects are masked from 'package:stats':
  filter, lag

The following objects are masked from 'package:base':
  intersect, setdiff, setequal, union

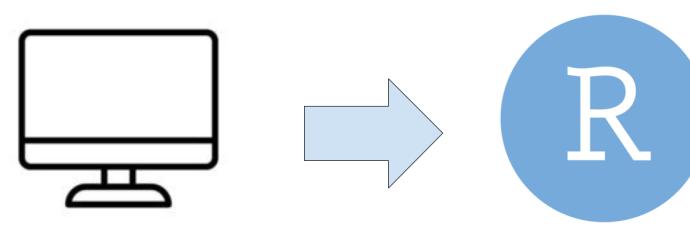
Warning message:
package 'dplyr' was built under R version 4.2.3
> |
```

R libraries // R ՅաՅՅօյծո

- Library installation:



- Library loading:



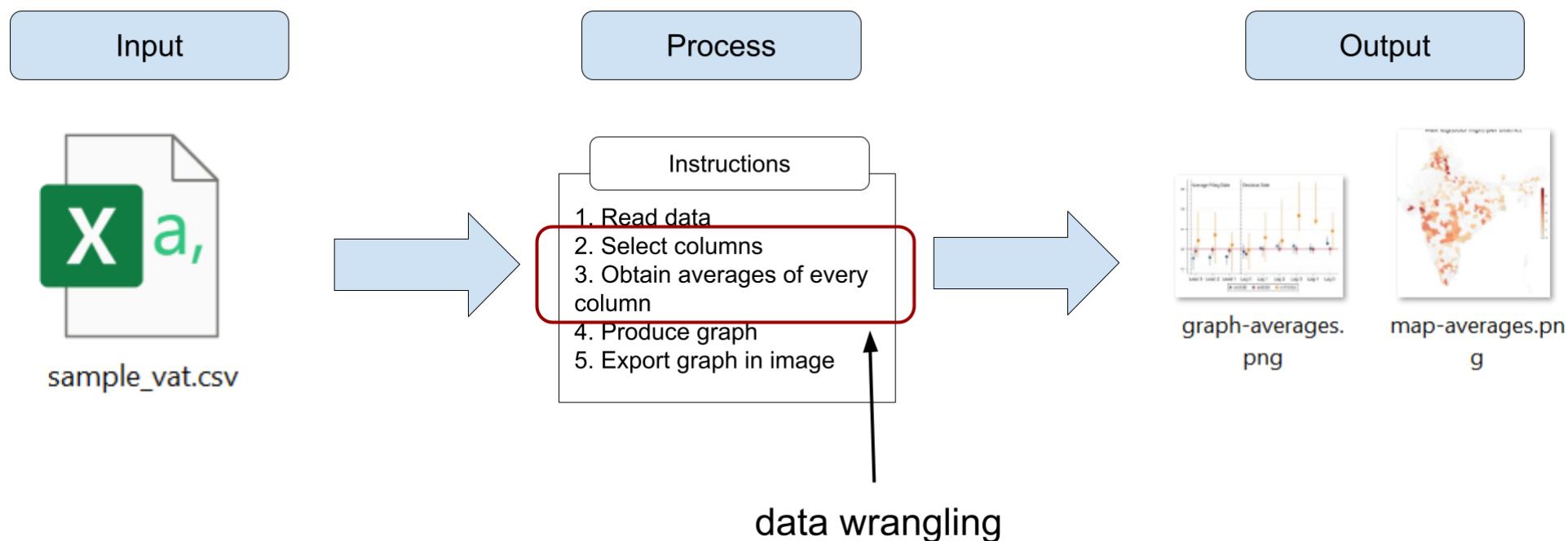
- You install R libraries only once in your computer
- You load libraries every time you open a new RStudio window (only load the libraries you will use)

Data wrangling // የወጪናመጥ኏ የሚሸፍበት

Data wrangling // የመናገድዎች ይቻል

Getting your data ready

- Data is rarely in a format where it can be converted in an output right away
- In statistical programming, the process of transforming data into a condition where it's ready to be converted into an output is called **data wrangling**



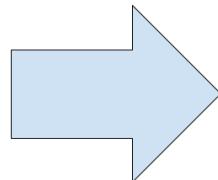
Data wrangling // የመናገዱዎችን ክፍያዎች

Getting your data ready

- Data wrangling is one of the most crucial and time-consuming aspects of data work
- It involves not only coding, but also the mental exercise of thinking what is the shape and condition that your dataframe needs to have in order to produce your desired output

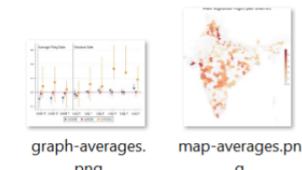
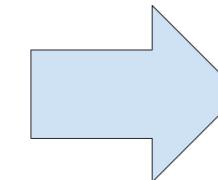
| Modified.ID | TaxPeriod | turnover_taxable18 | turnover_exempt | turnover_exports | vat_liability |
|-------------|-----------|--------------------|-----------------|------------------|---------------|
| 1 | 22172 | 201901 | 13444.64 | 0.00 | 2420.04 |
| 2 | 4592952 | 201906 | 0.00 | 0.00 | 0.00 |
| 3 | 10046564 | 201905 | 13237.30 | 0.00 | 2382.71 |
| 4 | 4797328 | 201912 | 184477.88 | 0.00 | 33206.02 |
| 5 | 5889624 | 201904 | 0.00 | 0.00 | 0.00 |
| 6 | 12233616 | 201912 | 0.00 | 0.00 | 0.00 |
| 7 | 6248240 | 201904 | 2416.93 | 0.00 | 435.05 |
| 8 | 538420 | 201905 | 7090.50 | 0.00 | 1276.29 |
| 9 | 10072288 | 201905 | 8039.20 | 0.00 | 1447.06 |

initial dataframe



| Modified.ID | TaxPeriod | turnover_taxable18 | turnover_exempt | turnover_exports | vat_liability |
|-------------|-----------|--------------------|-----------------|------------------|---------------|
| 1 | 22172 | 0.00 | 0.00 | 0.00 | 2420.04 |
| 2 | 4592952 | 0.00 | 0.00 | 0.00 | 0.00 |
| 3 | 10046564 | 0.00 | 0.00 | 0.00 | 2382.71 |
| 4 | 4797328 | 0.00 | 0.00 | 0.00 | 33206.02 |
| 5 | 5889624 | 0.00 | 0.00 | 0.00 | 0.00 |
| 6 | 12233616 | 0.00 | 0.00 | 0.00 | 0.00 |
| 7 | 6248240 | 0.00 | 0.00 | 0.00 | 435.05 |
| 8 | 538420 | 0.00 | 0.00 | 0.00 | 1276.29 |
| 9 | 10072288 | 0.00 | 0.00 | 0.00 | 1447.06 |

dataframe needed for outputs



outputs

Data wrangling // მონაცემთა ჩბუჯი

Getting your data ready

- As we said before we'll use `dplyr` and `tidyverse` for data wrangling in this training
- You can also use basic R, but we recommend these libraries because its functions are easier to use



Data wrangling // მონაცემთა ჩბუჯი

Exercise 4: Loading data

Note that this part of this is the same exercise we did in session 1, but it's okay to repeat it in order to start using a new RStudio session. **If you have RStudio open, start by closing the window and opening RStudio again.**

1. In your new RStudio window, go to **File > Import Dataset > From Text (base)** and select again the file **small_business_2019_age.csv**

- if you don't know where the file is, check in the **Downloads** folder
- if you need to download it again, it's here:
<https://osf.io/2aph/>

2. Make sure to select **Heading > Yes** in the next window

3. Select **Import**

4. Download this new file: <https://osf.io/v6psa> and repeat steps 1-3 with it

| modified_id | region | income |
|-------------|----------------------|----------|
| 2933828 | Kaxeti | 445.00 |
| 11539816 | Tbilisi | 3610.00 |
| 774836 | Guria | 2600.00 |
| 10763744 | Tbilisi | 29.00 |
| 5443012 | Kaxeti | 95.00 |
| 1303812 | Guria | 4255.60 |
| 2586640 | Guria | 2852.00 |
| 679632 | Guria | 0.00 |
| 10490076 | Tbilisi | 273.00 |
| 11176036 | Samegrelo-Z. SvaneTi | 2711.00 |
| 244516 | Tbilisi | 412.44 |
| 1431424 | Guria | 806.10 |
| 1485348 | Guria | 289.50 |
| 562104 | Guria | 1120.00 |
| 9048544 | Tbilisi | 47114.40 |
| 3735768 | Kaxeti | 605.00 |
| 6253780 | Kaxeti | 2967.00 |
| 11783480 | Tbilisi | 5960.00 |

Data wrangling // የመናገዱዎችን አጭዣ

The screenshot shows the RStudio interface with the Data View tab selected. The top menu bar includes Environment, History, Connections, Tutorial, and a set of icons for file operations. Below the menu is a toolbar with icons for Import Dataset, file size (132 MiB), and a brush tool. The Global Environment dropdown shows "R" and "Global Environment". A search bar with a magnifying glass icon is also present. The main area displays two datasets:

| small_business_2019 | 984 obs. of 3 variables | grid icon |
|-------------------------|-------------------------|-----------|
| small_business_2019_age | 984 obs. of 2 variables | grid icon |

Data wrangling // የመናገዱዎች ምንምዶ

Note: loading data with a function

- You can also load CSV data with the function `read.csv()` instead of using this point-and-click approach
- The first argument of `read.csv()` is the path in your computer where your data is. For example

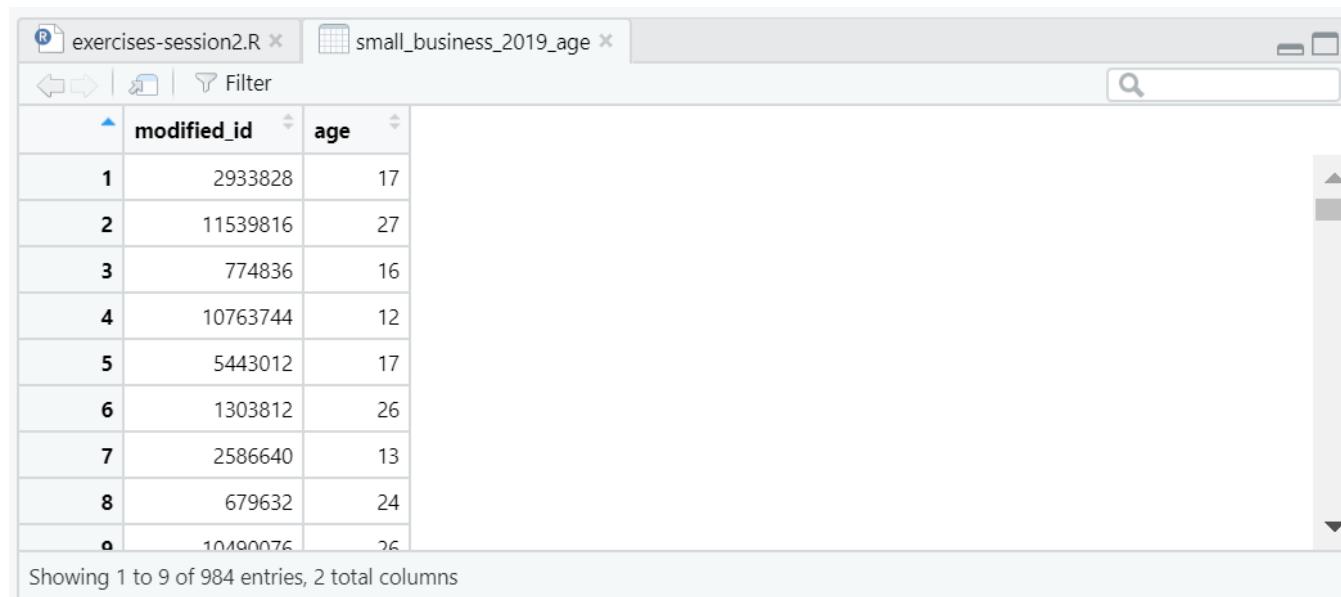
```
sample_vat <- read.csv("C:/Users/wb532468/Downloads/small_business_2019_age.csv")
```

- As usual, you need to save the result of `read.csv()` into a dataframe object with the arrow operator (`<-`) for it to be stored in the environment

Data wrangling // የመንግሥት ስምምነት

Recap: knowing your data

- Dataframe `small_business_2019` is the same dataframe we used last session that contains reported income of small business in 2019 and their locations
- The new dataframe is `small_business_2019_age`
- Each row is one small business with their corresponding firm age for 2019
- Column `Modified_ID` is a taxpayer identifier
- `age` is the firm age



| | modified_id | age |
|---|-------------|-----|
| 1 | 2933828 | 17 |
| 2 | 11539816 | 27 |
| 3 | 774836 | 16 |
| 4 | 10763744 | 12 |
| 5 | 5443012 | 17 |
| 6 | 1303812 | 26 |
| 7 | 2586640 | 13 |
| 8 | 679632 | 24 |
| 9 | 10190076 | 26 |

Showing 1 to 9 of 984 entries, 2 total columns

Data wrangling // የመናገድዎችን አጭዣ

- We will only use this dataframe in one of the next exercises, but we loaded it now because it's in general a good practice to have data loaded into the memory so it's ready to be used
- For the next exercises, we will face (likely) scenarios that will show us data work operations that require data wrangling

Filtering and sorting // ფილტრაცია და დახარისხება

Filtering and sorting // ფილტრაცია და დასტარისებება

Data work request

Scenario 1: Imagine you're approached with the following request:

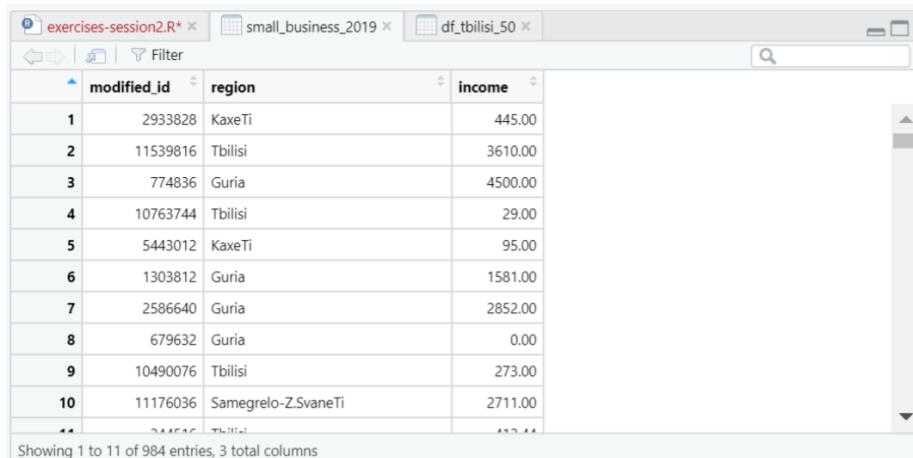
"The Georgia RS directorate is thinking of changing the criteria of what is considered a small business in Tbilisi. A critical input for this would be the list of the 50 biggest small business in Tbilisi in 2019. Can you produce such list? There is data from 2019 reported business income you can use"

Filtering and sorting // ფილტრაცია და დასტარტინგი

Data work request

"There is data from 2019 reported business income you can use"

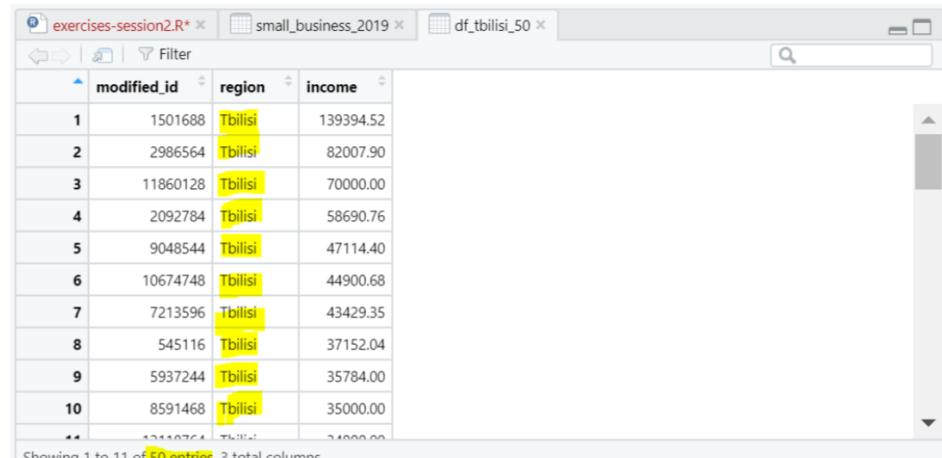
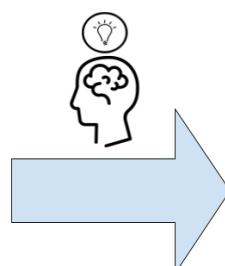
"A critical input for this would be the list of the 50 biggest small business in Tbilisi in 2019"



| | modified_id | region | income |
|----|-------------|----------------------|---------|
| 1 | 2933828 | Kaxeti | 445.00 |
| 2 | 11539816 | Tbilisi | 3610.00 |
| 3 | 774836 | Guria | 4500.00 |
| 4 | 10763744 | Tbilisi | 29.00 |
| 5 | 5443012 | Kaxeti | 95.00 |
| 6 | 1303812 | Guria | 1581.00 |
| 7 | 2586640 | Guria | 2852.00 |
| 8 | 679632 | Guria | 0.00 |
| 9 | 10490076 | Tbilisi | 273.00 |
| 10 | 11176036 | Samegrelo-Z. Svaneti | 2711.00 |

Showing 1 to 11 of 984 entries, 3 total columns

- Observations: Business reported income of 2019



| | modified_id | region | income |
|----|-------------|---------|-----------|
| 1 | 1501688 | Tbilisi | 139394.52 |
| 2 | 2986564 | Tbilisi | 82007.90 |
| 3 | 11860128 | Tbilisi | 70000.00 |
| 4 | 2092784 | Tbilisi | 58690.76 |
| 5 | 9048544 | Tbilisi | 47114.40 |
| 6 | 10674748 | Tbilisi | 44900.68 |
| 7 | 7213596 | Tbilisi | 43429.35 |
| 8 | 545116 | Tbilisi | 37152.04 |
| 9 | 5937244 | Tbilisi | 35784.00 |
| 10 | 8591468 | Tbilisi | 35000.00 |

Showing 1 to 11 of 50 entries, 3 total columns

- Observations: Business reported income of 2019, only for Tbilisi, and the 50 highest
- Columns: notice that the columns are the same as in the starting dataframe

Filtering and sorting // ფილტრაცია და დასტარტისებება

Data work request

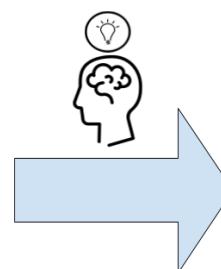
The data wrangling here involves a number of operations:

1. Keeping only the businesses in Tbilisi
2. Sorting by business income
3. Keeping only the 50 first businesses

| | modified_id | region | income |
|----|-------------|---------------------|---------|
| 1 | 2933828 | KaxeTi | 445.00 |
| 2 | 11539816 | Tbilisi | 3610.00 |
| 3 | 774836 | Guria | 4500.00 |
| 4 | 10763744 | Tbilisi | 29.00 |
| 5 | 5443012 | KaxeTi | 95.00 |
| 6 | 1303812 | Guria | 1581.00 |
| 7 | 2586640 | Guria | 2852.00 |
| 8 | 679632 | Guria | 0.00 |
| 9 | 10490076 | Tbilisi | 273.00 |
| 10 | 11176036 | Samegrelo-Z.SvaneTi | 2711.00 |
| 11 | 244516 | Tbilisi | 112.44 |

Showing 1 to 11 of 984 entries, 3 total columns

- Observations: Business reported income of 2019



| | modified_id | region | income |
|----|-------------|---------|-----------|
| 1 | 1501688 | Tbilisi | 139394.52 |
| 2 | 2986564 | Tbilisi | 82007.90 |
| 3 | 11860128 | Tbilisi | 70000.00 |
| 4 | 2092784 | Tbilisi | 58690.76 |
| 5 | 9048544 | Tbilisi | 47114.40 |
| 6 | 10674748 | Tbilisi | 44900.68 |
| 7 | 7213596 | Tbilisi | 43429.35 |
| 8 | 545116 | Tbilisi | 37152.04 |
| 9 | 5937244 | Tbilisi | 35784.00 |
| 10 | 8591468 | Tbilisi | 35000.00 |
| 11 | 244516 | Tbilisi | 34000.00 |

Showing 1 to 11 of 50 entries, 3 total columns

- Observations: Business reported income of 2019, only for Tbilisi, and the 50 highest

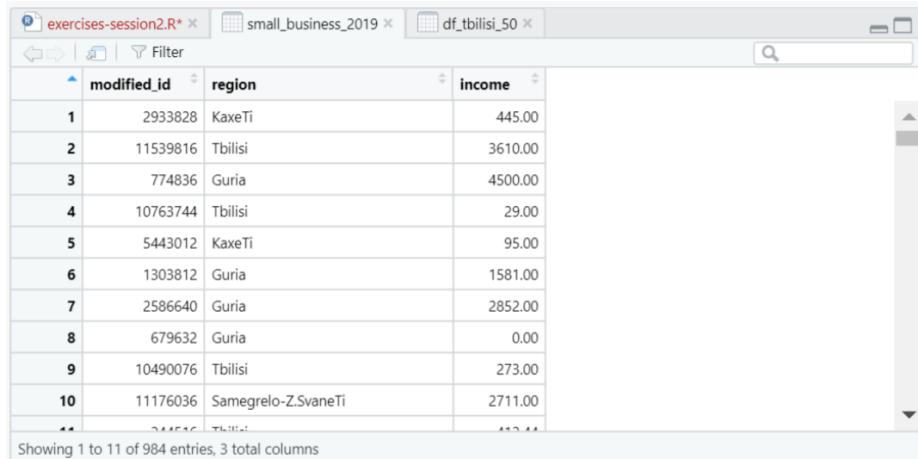
- Columns: notice that the columns are the same as in the starting data frame

Filtering and sorting // ფილტრაცია და დასტარტის ხედი

1. Keeping only the businesses in Tbilisi

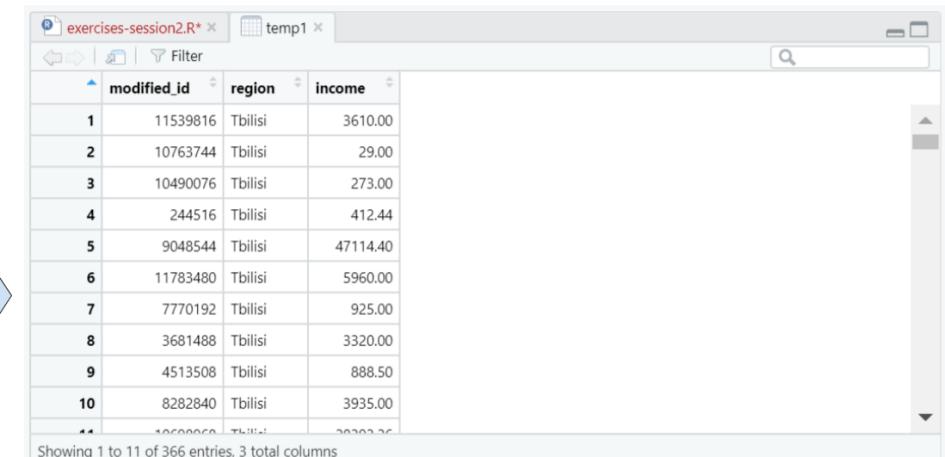
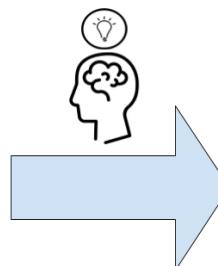
Use `filter()` for this:

```
temp1 <- filter(small_business_2019, region == "Tbilisi")
```



| | modified_id | region | income |
|----|-------------|----------------------|---------|
| 1 | 2933828 | Kaxeti | 445.00 |
| 2 | 11539816 | Tbilisi | 3610.00 |
| 3 | 774836 | Guria | 4500.00 |
| 4 | 10763744 | Tbilisi | 29.00 |
| 5 | 5443012 | Kaxeti | 95.00 |
| 6 | 1303812 | Guria | 1581.00 |
| 7 | 2586640 | Guria | 2852.00 |
| 8 | 679632 | Guria | 0.00 |
| 9 | 10490076 | Tbilisi | 273.00 |
| 10 | 11176036 | Samegrelo-Z. Svaneti | 2711.00 |
| 11 | 244516 | Tbilisi | 412.44 |

Showing 1 to 11 of 984 entries, 3 total columns



| | modified_id | region | income |
|----|-------------|---------|----------|
| 1 | 11539816 | Tbilisi | 3610.00 |
| 2 | 10763744 | Tbilisi | 29.00 |
| 3 | 10490076 | Tbilisi | 273.00 |
| 4 | 244516 | Tbilisi | 412.44 |
| 5 | 9048544 | Tbilisi | 47114.40 |
| 6 | 11783480 | Tbilisi | 5960.00 |
| 7 | 7770192 | Tbilisi | 925.00 |
| 8 | 3681488 | Tbilisi | 3320.00 |
| 9 | 4513508 | Tbilisi | 888.50 |
| 10 | 8282840 | Tbilisi | 3935.00 |
| 11 | 3060060 | Tbilisi | 20202.76 |

Showing 1 to 11 of 366 entries, 3 total columns

- Observations: Business reported income of 2019

- Observations: Business reported income of 2019, only for Tbilisi

Filtering and sorting // ფილტრაცია და დასტარტისებება

2. Sorting by business income

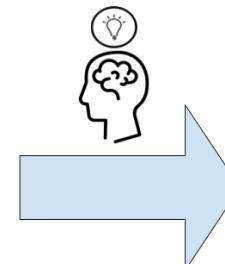
Use the function `arrange()` to sort. Sortings are ascending by default in R, hence we can include the minus (-) symbol before `income` to tell R we want to sort descending

```
small_business_tb_sorted <- arrange(small_business_tb, -income)
```

| | modified_id | region | income |
|----|-------------|---------|----------|
| 1 | 11539816 | Tbilisi | 3610.00 |
| 2 | 10763744 | Tbilisi | 29.00 |
| 3 | 10490076 | Tbilisi | 273.00 |
| 4 | 244516 | Tbilisi | 412.44 |
| 5 | 9048544 | Tbilisi | 47114.40 |
| 6 | 11783480 | Tbilisi | 5960.00 |
| 7 | 7770192 | Tbilisi | 925.00 |
| 8 | 3681488 | Tbilisi | 3320.00 |
| 9 | 4513508 | Tbilisi | 888.50 |
| 10 | 8282840 | Tbilisi | 3935.00 |

Showing 1 to 11 of 366 entries, 3 total columns

- Observations: Business reported income of 2019, only for Tbilisi



| | modified_id | region | income |
|----|-------------|---------|-----------|
| 1 | 1501688 | Tbilisi | 139394.52 |
| 2 | 2986564 | Tbilisi | 82007.90 |
| 3 | 11860128 | Tbilisi | 70000.00 |
| 4 | 2092784 | Tbilisi | 58690.76 |
| 5 | 9048544 | Tbilisi | 47114.40 |
| 6 | 10674748 | Tbilisi | 44900.68 |
| 7 | 7213596 | Tbilisi | 43429.35 |
| 8 | 545116 | Tbilisi | 37152.04 |
| 9 | 5937244 | Tbilisi | 35784.00 |
| 10 | 8591468 | Tbilisi | 35000.00 |

Showing 1 to 11 of 366 entries, 3 total columns

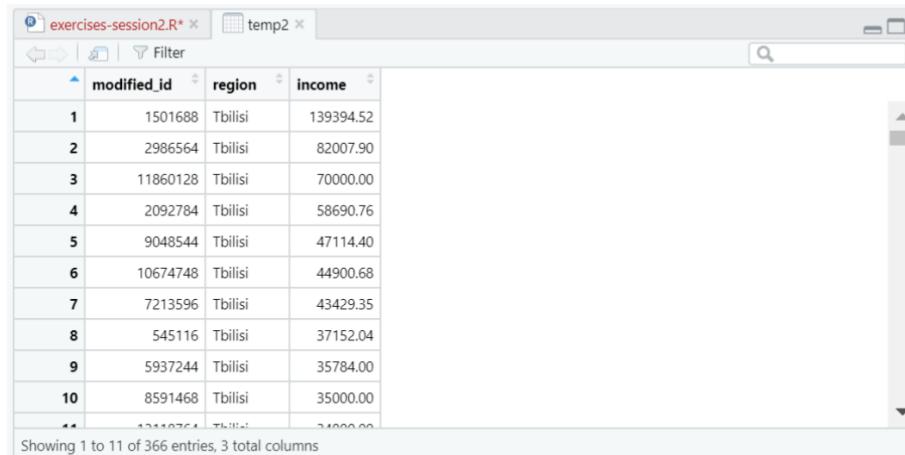
- Observations: Business reported income of 2019, only for Tbilisi, sorted descending by income

Filtering and sorting // ფილტრაცია და დასტარტისებება

3. Keeping only the 50 first businesses after the sorting

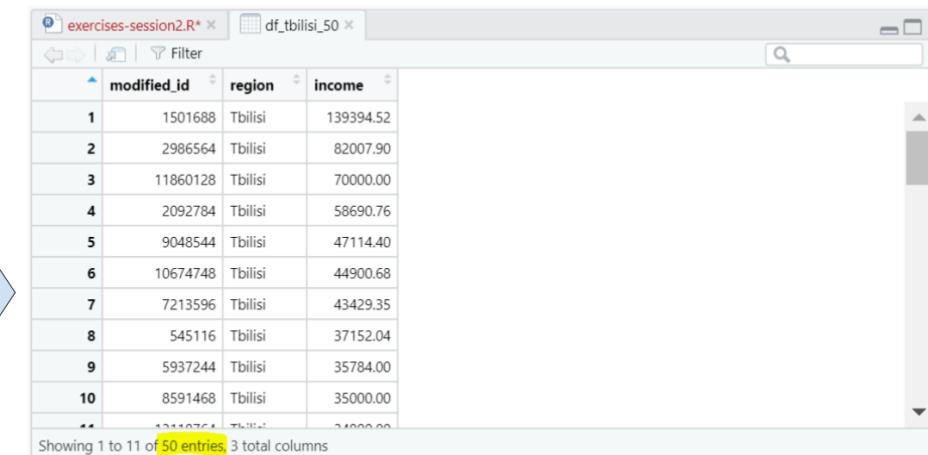
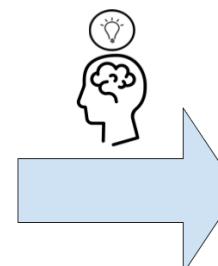
Use `filter()` again and the helper command `nrow()`

```
result_scenario1 <- filter(small_business_tb_sorted, row_number() <= 50)
```



| | modified_id | region | income |
|----|-------------|---------|-----------|
| 1 | 1501688 | Tbilisi | 139394.52 |
| 2 | 2986564 | Tbilisi | 82007.90 |
| 3 | 11860128 | Tbilisi | 70000.00 |
| 4 | 2092784 | Tbilisi | 58690.76 |
| 5 | 9048544 | Tbilisi | 47114.40 |
| 6 | 10674748 | Tbilisi | 44900.68 |
| 7 | 7213596 | Tbilisi | 43429.35 |
| 8 | 545116 | Tbilisi | 37152.04 |
| 9 | 5937244 | Tbilisi | 35784.00 |
| 10 | 8591468 | Tbilisi | 35000.00 |
| .. | 12110764 | Tbilisi | 34800.00 |

Showing 1 to 11 of 366 entries, 3 total columns



| | modified_id | region | income |
|----|-------------|---------|-----------|
| 1 | 1501688 | Tbilisi | 139394.52 |
| 2 | 2986564 | Tbilisi | 82007.90 |
| 3 | 11860128 | Tbilisi | 70000.00 |
| 4 | 2092784 | Tbilisi | 58690.76 |
| 5 | 9048544 | Tbilisi | 47114.40 |
| 6 | 10674748 | Tbilisi | 44900.68 |
| 7 | 7213596 | Tbilisi | 43429.35 |
| 8 | 545116 | Tbilisi | 37152.04 |
| 9 | 5937244 | Tbilisi | 35784.00 |
| 10 | 8591468 | Tbilisi | 35000.00 |
| .. | 12110764 | Tbilisi | 34800.00 |

Showing 1 to 11 of 50 entries, 3 total columns

- Observations: Business reported income of 2019, only for Tbilisi, sorted descending by income

- Observations: Business reported income of 2019, only for Tbilisi, sorted descending by income, and the 50 highest

Filtering and sorting // ფილტრაცია და დასტარტისებება

Exercise 5: filter and sort your data

Now that we figured out the shape the resulting dataframe needs to have and how to get it there, we can write code for this.

1.- Row selection:

```
temp1 <- filter(small_business_2019, region == "Tbilisi")
```

2.- Sort descending by income:

```
temp2 <- arrange(temp1, -income)
```

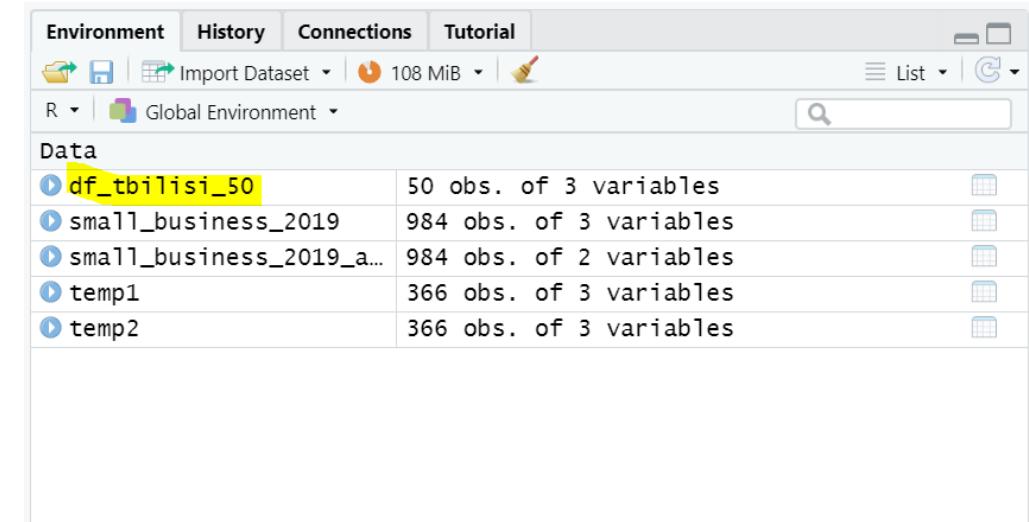
3.- Keep only the 50 first businesses after sorting:

```
df_tbilisi_50 <- filter(temp2, row_number() <= 50)
```

Filtering and sorting // ფილტრაცია და დასტარტინგი

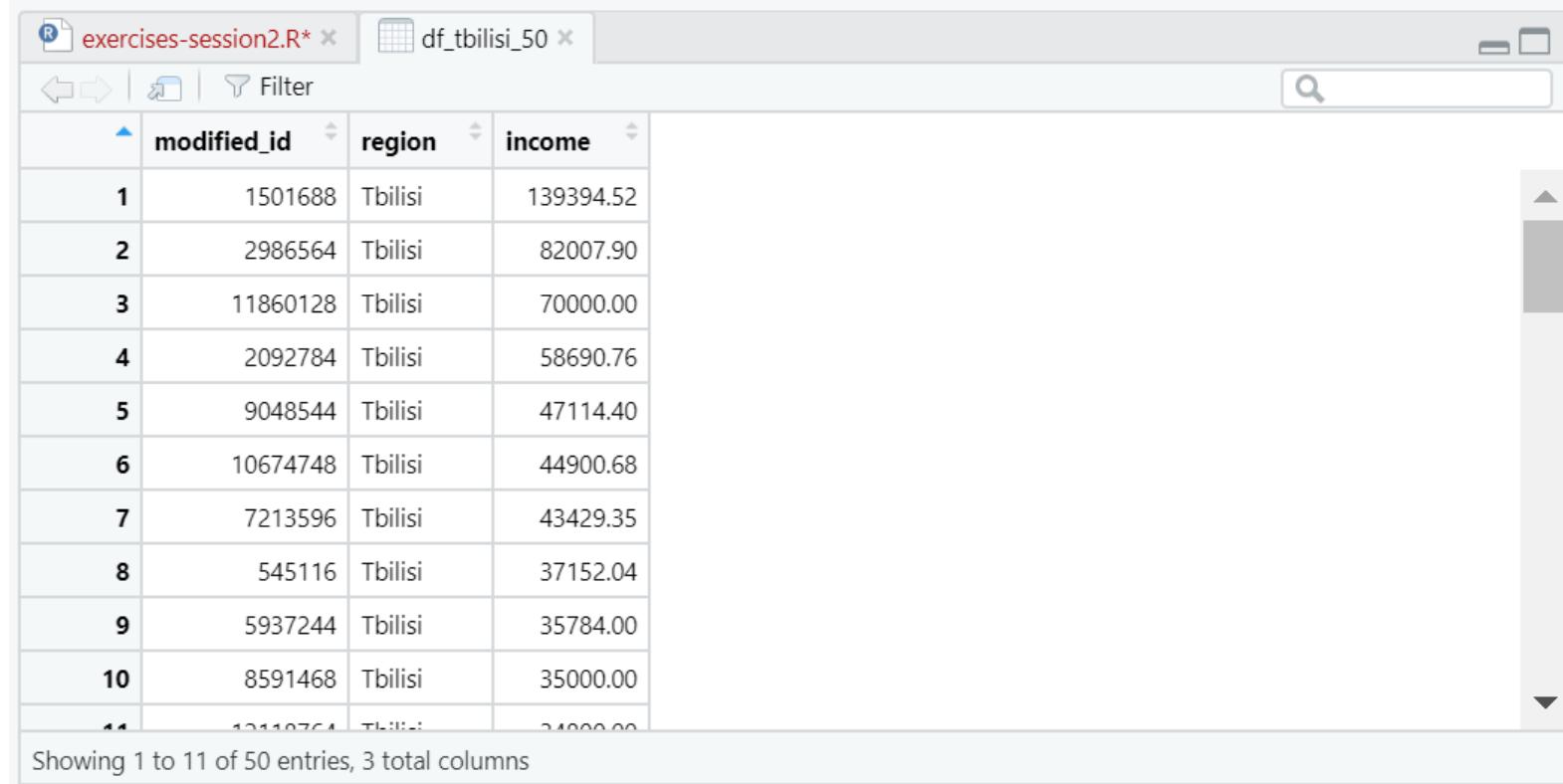
Some notes:

- `filter()`, `arrange()`, and `row_number()` are all functions from `dplyr`. Remember you have to always load `dplyr` first with `library(dplyr)` to be able to use them
- Note that we're creating two intermediate dataframes named `temp1` and `temp2` in the process
- We can avoid this by using the pipes operator (`%>%`). Pipes are very common in R programming, but we're not explaining them in this session
- The resulting dataframe is `df_tbilisi_50`



Filtering and sorting // ფილტრაცია და დასტარტისებება

You can check the result with `View(df_tbilisi_50)`. Now this dataframe has exactly what we wanted!



| | modified_id | region | income |
|----|-------------|---------|-----------|
| 1 | 1501688 | Tbilisi | 139394.52 |
| 2 | 2986564 | Tbilisi | 82007.90 |
| 3 | 11860128 | Tbilisi | 70000.00 |
| 4 | 2092784 | Tbilisi | 58690.76 |
| 5 | 9048544 | Tbilisi | 47114.40 |
| 6 | 10674748 | Tbilisi | 44900.68 |
| 7 | 7213596 | Tbilisi | 43429.35 |
| 8 | 545116 | Tbilisi | 37152.04 |
| 9 | 5937244 | Tbilisi | 35784.00 |
| 10 | 8591468 | Tbilisi | 35000.00 |
| 11 | 12110764 | Tbilisi | 24000.00 |

Showing 1 to 11 of 50 entries, 3 total columns

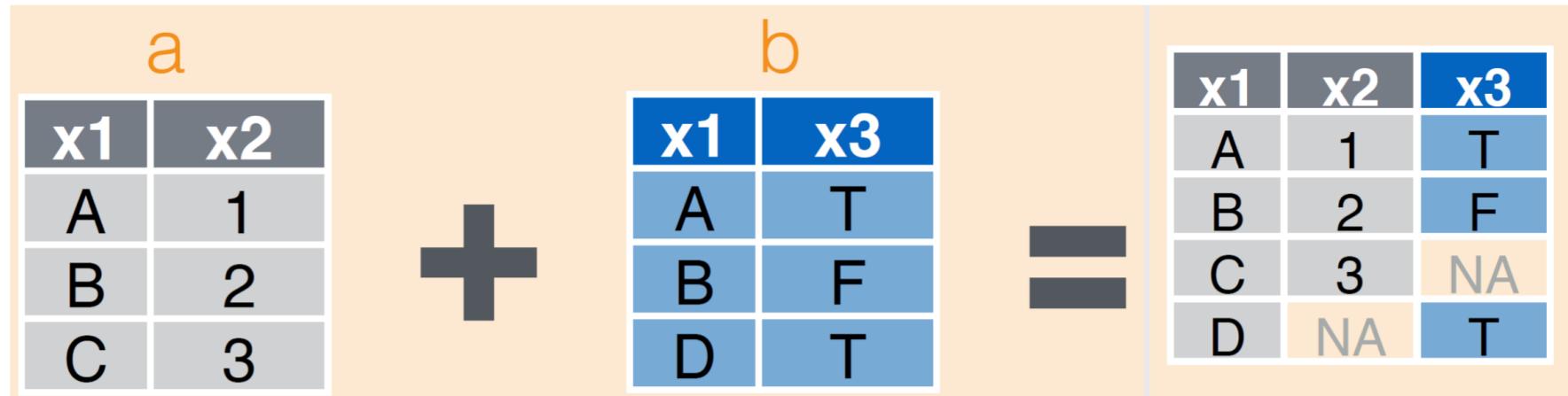
Filtering and sorting // ფილტრაცია და დახვრისება

- Filtering and sorting are two very common data wrangling operations in statistical programming
- Now we'll review a new data wrangling operation that is also quite common and useful: **merging**, also called joining

Merging dataframes // მონაცემთა ჩარჩოების შეერთება

Merging dataframes

- We'll explore one more common data wrangling operation: merging
- Merging is used when you need to bring columns from one dataframe to another
- When merging you need to use a "key" column that identifies the same units in different dataframes



Note: Image taken from RStudio's data wrangling cheat sheet

Merging dataframes

Data work request

Scenario 2:

"We want to know the total business reported income for small businesses with more than 5 years in 2019. You might want to use the data of small businesses income in 2019 that you already know and another file with small businesses age"

Merging dataframes

Data work request

"Use the data of small businesses income in 2019 that you already know and another file with small businesses age"

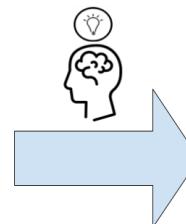
"We want to know the total business income for small businesses with more than 5 years in 2019"

| modified_id | region | income |
|-------------|----------------------|---------|
| 1 | Kaxeti | 445.00 |
| 2 | Tbilisi | 3610.00 |
| 3 | Guria | 4500.00 |
| 4 | Tbilisi | 29.00 |
| 5 | Kaxeti | 95.00 |
| 6 | Guria | 1581.00 |
| 7 | Guria | 2852.00 |
| 8 | Guria | 0.00 |
| 9 | Tbilisi | 273.00 |
| 10 | Samegrelo-Z. Svaneti | 2711.00 |
| .. | .. | .. |

- Observations: Business reported income and region of 2019

| modified_id | age |
|-------------|-----|
| 1 | 17 |
| 2 | 27 |
| 3 | 16 |
| 4 | 12 |
| 5 | 17 |
| 6 | 26 |
| 7 | 13 |
| 8 | 24 |
| 9 | 26 |
| 10 | 5 |
| .. | .. |

- Observations: business age in 2019



2592266

- Total business income for small businesses with more than five years in 2019

Merging dataframes

Data work request

These are the steps we'd need to follow to get the data wrangled for this result:

1. Select only the relevant columns from `small_business_2019`
2. Merge the dataframes
3. Filter only businesses with more than 5 years of age
4. Calculate the total income

Merging dataframes

1. Select only the relevant columns from `small_business_2019`

Use `select()` for this:

```
temp1 <- select(small_business_2019, modified_id, income)
```

| | modified_id | region | income |
|----|-------------|----------------------|---------|
| 1 | 2933828 | Kaxeti | 445.00 |
| 2 | 11539816 | Tbilisi | 3610.00 |
| 3 | 774836 | Guria | 4500.00 |
| 4 | 10763744 | Tbilisi | 29.00 |
| 5 | 5443012 | Kaxeti | 95.00 |
| 6 | 1303812 | Guria | 1581.00 |
| 7 | 2586640 | Guria | 2852.00 |
| 8 | 679632 | Guria | 0.00 |
| 9 | 10490076 | Tbilisi | 273.00 |
| 10 | 11176036 | Samegrelo-Z. Svaneti | 2711.00 |

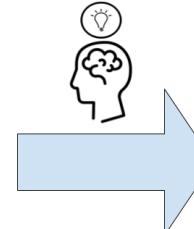
Showing 1 to 11 of 984 entries, 3 total columns

- Observations: Business reported income and region of 2019
- Columns: modified_id, region, income

| | modified_id | income |
|----|-------------|---------|
| 1 | 2933828 | 445.00 |
| 2 | 11539816 | 3610.00 |
| 3 | 774836 | 4500.00 |
| 4 | 10763744 | 29.00 |
| 5 | 5443012 | 95.00 |
| 6 | 1303812 | 1581.00 |
| 7 | 2586640 | 2852.00 |
| 8 | 679632 | 0.00 |
| 9 | 10490076 | 273.00 |
| 10 | 11176036 | 2711.00 |
| 11 | 244516 | 412.44 |
| 12 | 11176036 | 206.10 |

Showing 1 to 12 of 984 entries, 2 total columns

- Observations: Business reported income in 2019
- Columns: modified_id, income



Merging dataframes

2. Merge the dataframes

Use `inner_join()` to merge the dataframes:

```
temp2 <- inner_join(temp1, small_business_2019_age, by = "modified_id")
```

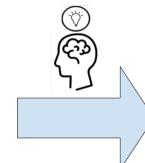
- The first two arguments are the two dataframes we want to join
- The third (named) argument is the "key" variable we merge by

| | modified_id | income |
|----|-------------|---------|
| 1 | 2933828 | 445.00 |
| 2 | 11539816 | 3610.00 |
| 3 | 774836 | 4500.00 |
| 4 | 10763744 | 29.00 |
| 5 | 5443012 | 95.00 |
| 6 | 1303812 | 1581.00 |
| 7 | 2586640 | 2852.00 |
| 8 | 679632 | 0.00 |
| 9 | 10490076 | 273.00 |
| 10 | 11176036 | 2711.00 |
| 11 | 244516 | 412.44 |

- Observations: Business reported income in 2019
- Columns: modified_id, income

| | modified_id | age |
|----|-------------|-----|
| 1 | 2933828 | 17 |
| 2 | 11539816 | 27 |
| 3 | 774836 | 16 |
| 4 | 10763744 | 12 |
| 5 | 5443012 | 17 |
| 6 | 1303812 | 26 |
| 7 | 2586640 | 13 |
| 8 | 679632 | 24 |
| 9 | 10490076 | 26 |
| 10 | 11176036 | 5 |

- Observations: Business age in 2019
- Columns: modified_id, age



| | modified_id | income | age |
|----|-------------|---------|-----|
| 1 | 2933828 | 445.00 | 17 |
| 2 | 11539816 | 3610.00 | 27 |
| 3 | 774836 | 4500.00 | 16 |
| 4 | 10763744 | 29.00 | 12 |
| 5 | 5443012 | 95.00 | 17 |
| 6 | 1303812 | 1581.00 | 26 |
| 7 | 2586640 | 2852.00 | 13 |
| 8 | 679632 | 0.00 | 24 |
| 9 | 10490076 | 273.00 | 26 |
| 10 | 11176036 | 2711.00 | 5 |
| 11 | 244516 | 412.44 | 17 |

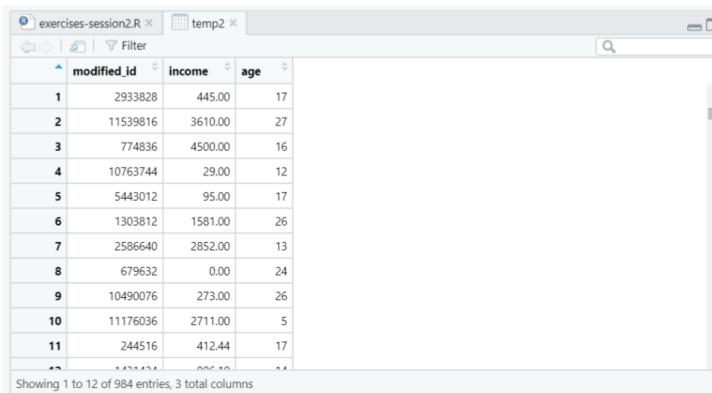
- Observations: Business age and reported income in 2019
- Columns: modified_id, income, age

Merging dataframes

3. Filter only businesses with more than 5 years of age

Use `filter()` once again:

```
temp3 <- filter(temp2, age > 5)
```



A screenshot of the RStudio interface showing a data frame named 'temp2'. The data frame has three columns: 'modified_id', 'income', and 'age'. The 'age' column contains values ranging from 5 to 27. A large blue arrow points from this screen to the next one.

| | modified_id | income | age |
|-----|-------------|---------|-----|
| 1 | 2933828 | 445.00 | 17 |
| 2 | 11539816 | 3610.00 | 27 |
| 3 | 774836 | 4500.00 | 16 |
| 4 | 10763744 | 29.00 | 12 |
| 5 | 5443012 | 95.00 | 17 |
| 6 | 1303812 | 1581.00 | 26 |
| 7 | 2586640 | 2852.00 | 13 |
| 8 | 679632 | 0.00 | 24 |
| 9 | 10490076 | 273.00 | 26 |
| 10 | 11176036 | 2711.00 | 5 |
| 11 | 244516 | 412.44 | 17 |
| 12 | 1431424 | 806.10 | 14 |
| 13 | 560914 | 442.00 | 17 |
| 14 | 1431424 | 806.10 | 14 |
| 15 | 560914 | 442.00 | 17 |
| 16 | 1431424 | 806.10 | 14 |
| 17 | 560914 | 442.00 | 17 |
| 18 | 1431424 | 806.10 | 14 |
| 19 | 560914 | 442.00 | 17 |
| 20 | 1431424 | 806.10 | 14 |
| 21 | 560914 | 442.00 | 17 |
| 22 | 1431424 | 806.10 | 14 |
| 23 | 560914 | 442.00 | 17 |
| 24 | 1431424 | 806.10 | 14 |
| 25 | 560914 | 442.00 | 17 |
| 26 | 1431424 | 806.10 | 14 |
| 27 | 560914 | 442.00 | 17 |
| 28 | 1431424 | 806.10 | 14 |
| 29 | 560914 | 442.00 | 17 |
| 30 | 1431424 | 806.10 | 14 |
| 31 | 560914 | 442.00 | 17 |
| 32 | 1431424 | 806.10 | 14 |
| 33 | 560914 | 442.00 | 17 |
| 34 | 1431424 | 806.10 | 14 |
| 35 | 560914 | 442.00 | 17 |
| 36 | 1431424 | 806.10 | 14 |
| 37 | 560914 | 442.00 | 17 |
| 38 | 1431424 | 806.10 | 14 |
| 39 | 560914 | 442.00 | 17 |
| 40 | 1431424 | 806.10 | 14 |
| 41 | 560914 | 442.00 | 17 |
| 42 | 1431424 | 806.10 | 14 |
| 43 | 560914 | 442.00 | 17 |
| 44 | 1431424 | 806.10 | 14 |
| 45 | 560914 | 442.00 | 17 |
| 46 | 1431424 | 806.10 | 14 |
| 47 | 560914 | 442.00 | 17 |
| 48 | 1431424 | 806.10 | 14 |
| 49 | 560914 | 442.00 | 17 |
| 50 | 1431424 | 806.10 | 14 |
| 51 | 560914 | 442.00 | 17 |
| 52 | 1431424 | 806.10 | 14 |
| 53 | 560914 | 442.00 | 17 |
| 54 | 1431424 | 806.10 | 14 |
| 55 | 560914 | 442.00 | 17 |
| 56 | 1431424 | 806.10 | 14 |
| 57 | 560914 | 442.00 | 17 |
| 58 | 1431424 | 806.10 | 14 |
| 59 | 560914 | 442.00 | 17 |
| 60 | 1431424 | 806.10 | 14 |
| 61 | 560914 | 442.00 | 17 |
| 62 | 1431424 | 806.10 | 14 |
| 63 | 560914 | 442.00 | 17 |
| 64 | 1431424 | 806.10 | 14 |
| 65 | 560914 | 442.00 | 17 |
| 66 | 1431424 | 806.10 | 14 |
| 67 | 560914 | 442.00 | 17 |
| 68 | 1431424 | 806.10 | 14 |
| 69 | 560914 | 442.00 | 17 |
| 70 | 1431424 | 806.10 | 14 |
| 71 | 560914 | 442.00 | 17 |
| 72 | 1431424 | 806.10 | 14 |
| 73 | 560914 | 442.00 | 17 |
| 74 | 1431424 | 806.10 | 14 |
| 75 | 560914 | 442.00 | 17 |
| 76 | 1431424 | 806.10 | 14 |
| 77 | 560914 | 442.00 | 17 |
| 78 | 1431424 | 806.10 | 14 |
| 79 | 560914 | 442.00 | 17 |
| 80 | 1431424 | 806.10 | 14 |
| 81 | 560914 | 442.00 | 17 |
| 82 | 1431424 | 806.10 | 14 |
| 83 | 560914 | 442.00 | 17 |
| 84 | 1431424 | 806.10 | 14 |
| 85 | 560914 | 442.00 | 17 |
| 86 | 1431424 | 806.10 | 14 |
| 87 | 560914 | 442.00 | 17 |
| 88 | 1431424 | 806.10 | 14 |
| 89 | 560914 | 442.00 | 17 |
| 90 | 1431424 | 806.10 | 14 |
| 91 | 560914 | 442.00 | 17 |
| 92 | 1431424 | 806.10 | 14 |
| 93 | 560914 | 442.00 | 17 |
| 94 | 1431424 | 806.10 | 14 |
| 95 | 560914 | 442.00 | 17 |
| 96 | 1431424 | 806.10 | 14 |
| 97 | 560914 | 442.00 | 17 |
| 98 | 1431424 | 806.10 | 14 |
| 99 | 560914 | 442.00 | 17 |
| 100 | 1431424 | 806.10 | 14 |
| 101 | 560914 | 442.00 | 17 |
| 102 | 1431424 | 806.10 | 14 |
| 103 | 560914 | 442.00 | 17 |
| 104 | 1431424 | 806.10 | 14 |
| 105 | 560914 | 442.00 | 17 |
| 106 | 1431424 | 806.10 | 14 |
| 107 | 560914 | 442.00 | 17 |
| 108 | 1431424 | 806.10 | 14 |
| 109 | 560914 | 442.00 | 17 |
| 110 | 1431424 | 806.10 | 14 |
| 111 | 560914 | 442.00 | 17 |
| 112 | 1431424 | 806.10 | 14 |
| 113 | 560914 | 442.00 | 17 |
| 114 | 1431424 | 806.10 | 14 |
| 115 | 560914 | 442.00 | 17 |
| 116 | 1431424 | 806.10 | 14 |
| 117 | 560914 | 442.00 | 17 |
| 118 | 1431424 | 806.10 | 14 |
| 119 | 560914 | 442.00 | 17 |
| 120 | 1431424 | 806.10 | 14 |
| 121 | 560914 | 442.00 | 17 |
| 122 | 1431424 | 806.10 | 14 |
| 123 | 560914 | 442.00 | 17 |
| 124 | 1431424 | 806.10 | 14 |
| 125 | 560914 | 442.00 | 17 |
| 126 | 1431424 | 806.10 | 14 |
| 127 | 560914 | 442.00 | 17 |
| 128 | 1431424 | 806.10 | 14 |
| 129 | 560914 | 442.00 | 17 |
| 130 | 1431424 | 806.10 | 14 |
| 131 | 560914 | 442.00 | 17 |
| 132 | 1431424 | 806.10 | 14 |
| 133 | 560914 | 442.00 | 17 |
| 134 | 1431424 | 806.10 | 14 |
| 135 | 560914 | 442.00 | 17 |
| 136 | 1431424 | 806.10 | 14 |
| 137 | 560914 | 442.00 | 17 |
| 138 | 1431424 | 806.10 | 14 |
| 139 | 560914 | 442.00 | 17 |
| 140 | 1431424 | 806.10 | 14 |
| 141 | 560914 | 442.00 | 17 |
| 142 | 1431424 | 806.10 | 14 |
| 143 | 560914 | 442.00 | 17 |
| 144 | 1431424 | 806.10 | 14 |
| 145 | 560914 | 442.00 | 17 |
| 146 | 1431424 | 806.10 | 14 |
| 147 | 560914 | 442.00 | 17 |
| 148 | 1431424 | 806.10 | 14 |
| 149 | 560914 | 442.00 | 17 |
| 150 | 1431424 | 806.10 | 14 |
| 151 | 560914 | 442.00 | 17 |
| 152 | 1431424 | 806.10 | 14 |
| 153 | 560914 | 442.00 | 17 |
| 154 | 1431424 | 806.10 | 14 |
| 155 | 560914 | 442.00 | 17 |
| 156 | 1431424 | 806.10 | 14 |
| 157 | 560914 | 442.00 | 17 |
| 158 | 1431424 | 806.10 | 14 |
| 159 | 560914 | 442.00 | 17 |
| 160 | 1431424 | 806.10 | 14 |
| 161 | 560914 | 442.00 | 17 |
| 162 | 1431424 | 806.10 | 14 |
| 163 | 560914 | 442.00 | 17 |
| 164 | 1431424 | 806.10 | 14 |
| 165 | 560914 | 442.00 | 17 |
| 166 | 1431424 | 806.10 | 14 |
| 167 | 560914 | 442.00 | 17 |
| 168 | 1431424 | 806.10 | 14 |
| 169 | 560914 | 442.00 | 17 |
| 170 | 1431424 | 806.10 | 14 |
| 171 | 560914 | 442.00 | 17 |
| 172 | 1431424 | 806.10 | 14 |
| 173 | 560914 | 442.00 | 17 |
| 174 | 1431424 | 806.10 | 14 |
| 175 | 560914 | 442.00 | 17 |
| 176 | 1431424 | 806.10 | 14 |
| 177 | 560914 | 442.00 | 17 |
| 178 | 1431424 | 806.10 | 14 |
| 179 | 560914 | 442.00 | 17 |
| 180 | 1431424 | 806.10 | 14 |
| 181 | 560914 | 442.00 | 17 |
| 182 | 1431424 | 806.10 | 14 |
| 183 | 560914 | 442.00 | 17 |
| 184 | 1431424 | 806.10 | 14 |
| 185 | 560914 | 442.00 | 17 |
| 186 | 1431424 | 806.10 | 14 |
| 187 | 560914 | 442.00 | 17 |
| 188 | 1431424 | 806.10 | 14 |
| 189 | 560914 | 442.00 | 17 |
| 190 | 1431424 | 806.10 | 14 |
| 191 | 560914 | 442.00 | 17 |
| 192 | 1431424 | 806.10 | 14 |
| 193 | 560914 | 442.00 | 17 |
| 194 | 1431424 | 806.10 | 14 |
| 195 | 560914 | 442.00 | 17 |
| 196 | 1431424 | 806.10 | 14 |
| 197 | 560914 | 442.00 | 17 |
| 198 | 1431424 | 806.10 | 14 |
| 199 | 560914 | 442.00 | 17 |
| 200 | 1431424 | 806.10 | 14 |
| 201 | 560914 | 442.00 | 17 |
| 202 | 1431424 | 806.10 | 14 |
| 203 | 560914 | 442.00 | 17 |
| 204 | 1431424 | 806.10 | 14 |
| 205 | 560914 | 442.00 | 17 |
| 206 | 1431424 | 806.10 | 14 |
| 207 | 560914 | 442.00 | 17 |
| 208 | 1431424 | 806.10 | 14 |
| 209 | 560914 | 442.00 | 17 |
| 210 | 1431424 | 806.10 | 14 |
| 211 | 560914 | 442.00 | 17 |
| 212 | 1431424 | 806.10 | 14 |
| 213 | 560914 | 442.00 | 17 |
| 214 | 1431424 | 806.10 | 14 |
| 215 | 560914 | 442.00 | 17 |
| 216 | 1431424 | 806.10 | 14 |
| 217 | 560914 | 442.00 | 17 |
| 218 | 1431424 | 806.10 | 14 |
| 219 | 560914 | 442.00 | 17 |
| 220 | 1431424 | 806.10 | 14 |
| 221 | 560914 | 442.00 | 17 |
| 222 | 1431424 | 806.10 | 14 |
| 223 | 560914 | 442.00 | 17 |
| 224 | 1431424 | 806.10 | 14 |
| 225 | 560914 | 442.00 | 17 |
| 226 | 1431424 | 806.10 | 14 |
| 227 | 560914 | 442.00 | 17 |
| 228 | 1431424 | 806.10 | 14 |
| 229 | 560914 | 442.00 | 17 |
| 230 | 1431424 | 806.10 | 14 |
| 231 | 560914 | 442.00 | 17 |
| 232 | 1431424 | 806.10 | 14 |
| 233 | 560914 | 442.00 | 17 |
| 234 | 1431424 | 806.10 | 14 |
| 235 | 560914 | 442.00 | 17 |
| 236 | 1431424 | 806.10 | 14 |
| 237 | 560914 | 442.00 | 17 |
| 238 | 1431424 | 806.10 | 14 |
| 239 | 560914 | 442.00 | 17 |
| 240 | 1431424 | 806.10 | 14 |
| 241 | 560914 | 442.00 | 17 |
| 242 | 1431424 | 806.10 | 14 |
| 243 | 560914 | 442.00 | 17 |
| 244 | 1431424 | 806.10 | 14 |
| 245 | 560914 | 442.00 | 17 |
| 246 | 1431424 | 806.10 | 14 |
| 247 | 560914 | 442.00 | 17 |
| 248 | 1431424 | 806.10 | 14 |
| 249 | 560914 | 442.00 | 17 |
| 250 | 1431424 | 806.10 | 14 |
| 251 | 560914 | 442.00 | 17 |
| 252 | 1431424 | 806.10 | 14 |
| 253 | 560914 | 442.00 | 17 |
| 254 | 1431424 | 806.10 | 14 |
| 255 | 560914 | 442.00 | 17 |
| 256 | 1431424 | 806.10 | 14 |
| 257 | 560914 | 442.00 | 17 |
| 258 | 1431424 | 806.10 | 14 |
| 259 | 560914 | 442.00 | 17 |
| 260 | 1431424 | 806.10 | 14 |
| 261 | 560914 | 442.00 | 17 |
| 262 | 1431424 | 806.10 | 14 |
| 263 | 560914 | 442.00 | 17 |
| 264 | 1431424 | 806.10 | 14 |
| 265 | 560914 | 442.00 | 17 |
| 266 | 1431424 | 806.10 | 14 |
| 267 | 560914 | 442.00 | 17 |

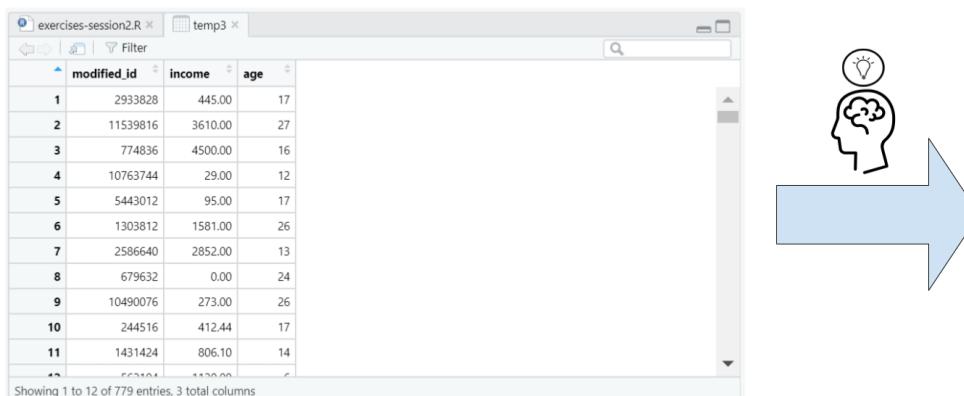
Merging dataframes

4. Calculate the total income

We use `colSums()` and `select()`.

- `colSums()` calculates the sum of all values of a column
- We use `select()` inside `colSums()` to select a single column from `temp3` to calculate the sum for

```
total_income <- colSums(select(temp3, income))
```



| | modified_id | income | age |
|----|-------------|---------|-----|
| 1 | 2933828 | 445.00 | 17 |
| 2 | 11539816 | 3610.00 | 27 |
| 3 | 774836 | 4500.00 | 16 |
| 4 | 10763744 | 29.00 | 12 |
| 5 | 5443012 | 95.00 | 17 |
| 6 | 1303812 | 1581.00 | 26 |
| 7 | 2586640 | 2852.00 | 13 |
| 8 | 679632 | 0.00 | 24 |
| 9 | 10490076 | 273.00 | 26 |
| 10 | 244516 | 412.44 | 17 |
| 11 | 1431424 | 806.10 | 14 |

Showing 1 to 12 of 779 entries, 3 total columns

- Observations: Business age and reported income in 2019, only for firms with more than five years of age
- Columns: modified_id, income, age

2592266

- Total business income for small businesses with more than five years in 2019

Merging dataframes

Exercise 6: Merge the dataframes

Apply all the steps we reviewed in the last slides to calculate the total reported income for small businesses with more than five years of age in 2019.

1.- Select only the relevant columns of `small_business_2019`:

```
temp1 <- select(small_business_2019, modified_id, income)
```

2.- Merge the dataframes

```
temp2 <- inner_join(temp1, small_business_2019_age, by = "modified_id")
```

3.- Filter only businesses with more than five years of age

```
temp3 <- filter(temp2, age > 5)
```

4.- Calculate the total income

```
total_income <- colSums(select(temp3, income))
```

Merging dataframes

`total_income` is a numeric value with the amount we wanted to estimate.

```
print(total_income)
```

```
## income  
## 2592266
```

Merging dataframes

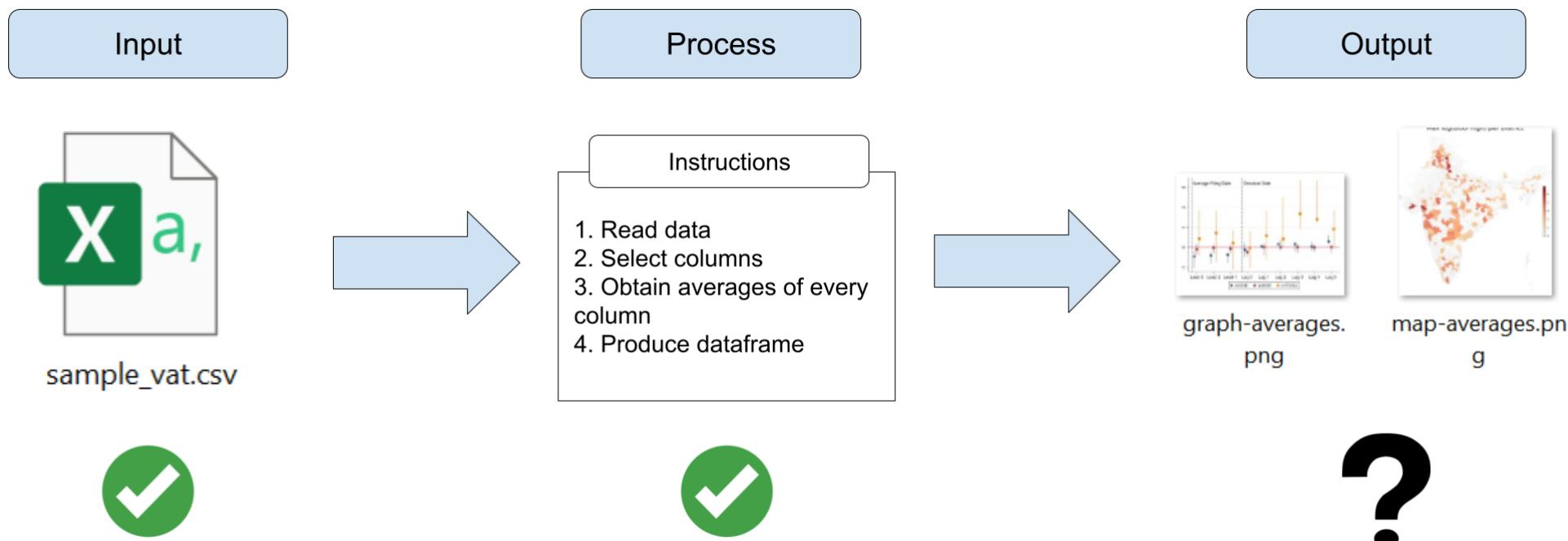
These were two examples we chose to show different possible data wrangling operations. A summary of these and other common operations are:

| Operation | Function in <code>dplyr</code> |
|---|---|
| Subset columns | <code>select()</code> |
| Subset rows (based on condition) | <code>filter()</code> |
| Create new columns | <code>mutate()</code> |
| Create new columns based on condition | <code>mutate()</code> and <code>case_when()</code> |
| Create new rows | <code>add_row()</code> |
| Merge dataframes | <code>inner_join()</code> , <code>left_join()</code> , <code>right_join()</code> , <code>full_join()</code> |
| Append dataframes | <code>bind_rows()</code> |
| Deduplicate | <code>distinct()</code> |
| Collapse and create summary indicators | <code>group_by()</code> , <code>summarize()</code> |
| Pass a result as the first argument for the next function | <code>%>%</code> (operator, not function) |

Exporting outputs // პროდუქციის ექსპორტი

Exporting outputs // პროდუქციის ექსპორტი

- Until now, we've seen full examples of part 1 and 2 of the data work pipeline
- What about exporting outputs?



- We'll see this in the next exercise

Exporting outputs // პროდუქციის ექსპორტი

Exporting dataframes

- The easiest way to export a dataframe or number is with the function `write.csv()`
- `write.csv()` creates a CSV file with the dataframe
- It takes two basic arguments:
 1. The name of the object you want to export
 2. A file path to export the object to
- `write.csv()` includes the row numbers by default. You can add the argument `row.names = FALSE` to avoid this

Exporting outputs // პროდუქციის ექსპორტი

Exercise 7: Export `df_tbilisi_50` and `total_income`

1. Use this code to export the results of the last two exercises:

```
write.csv(df_tbilisi_50,  
          "df_tbilisi_50.csv",  
          row.names = FALSE)  
  
write.csv(total_income,  
          "total_income.csv",  
          row.names = FALSE)
```

Exporting outputs // პროდუქციის ექსპორტი

Now `df_tbilisi_50.csv` and `total_income.csv` will show in your computer (probably in your `Documents` folder).

| Name | Date modified | Type | Size |
|---------------------------|--------------------|------------------------|-------|
| .Rproj.user | 9/19/2023 2:37 AM | File folder | |
| 1-introduction-to-r_cache | 9/19/2023 3:30 AM | File folder | |
| img | 9/19/2023 3:43 PM | File folder | |
| 2-data-wrangling_cache | 9/19/2023 3:46 PM | File folder | |
| data | 9/20/2023 12:11 AM | File folder | |
| libs | 9/20/2023 5:09 AM | File folder | |
| .Rhistory | 9/19/2023 4:20 PM | RHISTORY File | 1 KB |
| 1-introduction-to-r.Rmd | 9/20/2023 1:34 AM | RMD File | 24 KB |
| 1-introduction-to-r.html | 9/20/2023 1:34 AM | Chrome HTML Docu... | 30 KB |
| exercises-session1.R | 9/20/2023 1:41 AM | R File | 1 KB |
| 202309.Rproj | 9/20/2023 1:58 AM | R Project | 1 KB |
| exercises-session2.R | 9/20/2023 4:37 AM | R File | 1 KB |
| 2-data-wrangling.Rmd | 9/20/2023 5:09 AM | RMD File | 25 KB |
| 2-data-wrangling.html | 9/20/2023 5:09 AM | Chrome HTML Docu... | 30 KB |
| total_income.csv | 9/20/2023 5:11 AM | Microsoft Excel Com... | 1 KB |
| df_tbilisi_50.csv | 9/20/2023 5:12 AM | Microsoft Excel Com... | 2 KB |

Exporting outputs // პროდუქციის ექსპორტი

Some notes on file paths

- The second argument of `write.csv()` specifies the file path we export the dataframe to

```
write.csv(df_tbilisi_50,  
          "df_tbilisi_50.csv",  
          row.names = FALSE)
```

- You can include any path in your computer and R will write the file in that location
 - For example: `"C:/Users/wb532468/OneDrive - WBG/Desktop"` exports the file to the desktop of my computer (this will not work in other computers)
 - Note that file paths in R use forward slashes (`/`). Back slashes (`\`) **do not work in R**

Exporting outputs // პროდუქციის ექსპორტი

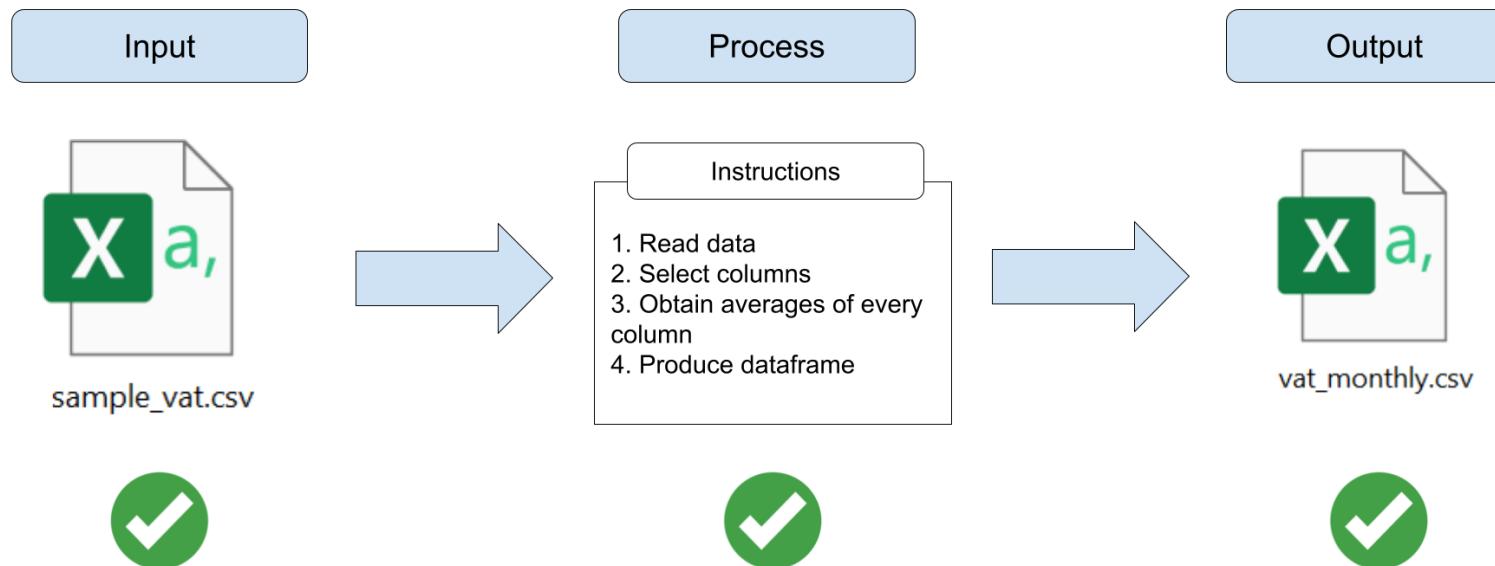
Some notes on file paths

```
write.csv(df_tbilisi_50,  
          "df_tbilisi_50.csv",  
          row.names = FALSE)
```

- If you only include a file name (as in `df_tbilisi_50.csv`), R will export the file to the current location of your RStudio window. This is usually the `Documents` folder in Windows
- You can check the current location of RStudio with the function `getwd()`

Exporting outputs // პროდუქციის ექსპორტი

Our data pipeline has been fully implemented at this point. Great!



Wrapping up // შეფასოვა

Wrapping up // დგენური

Don't forget to save your work!

- If you haven't, add code comments with `#` to differentiate your solutions for each exercise
- Click the floppy disk to save your work
- Make sure to remember where you're saving your file



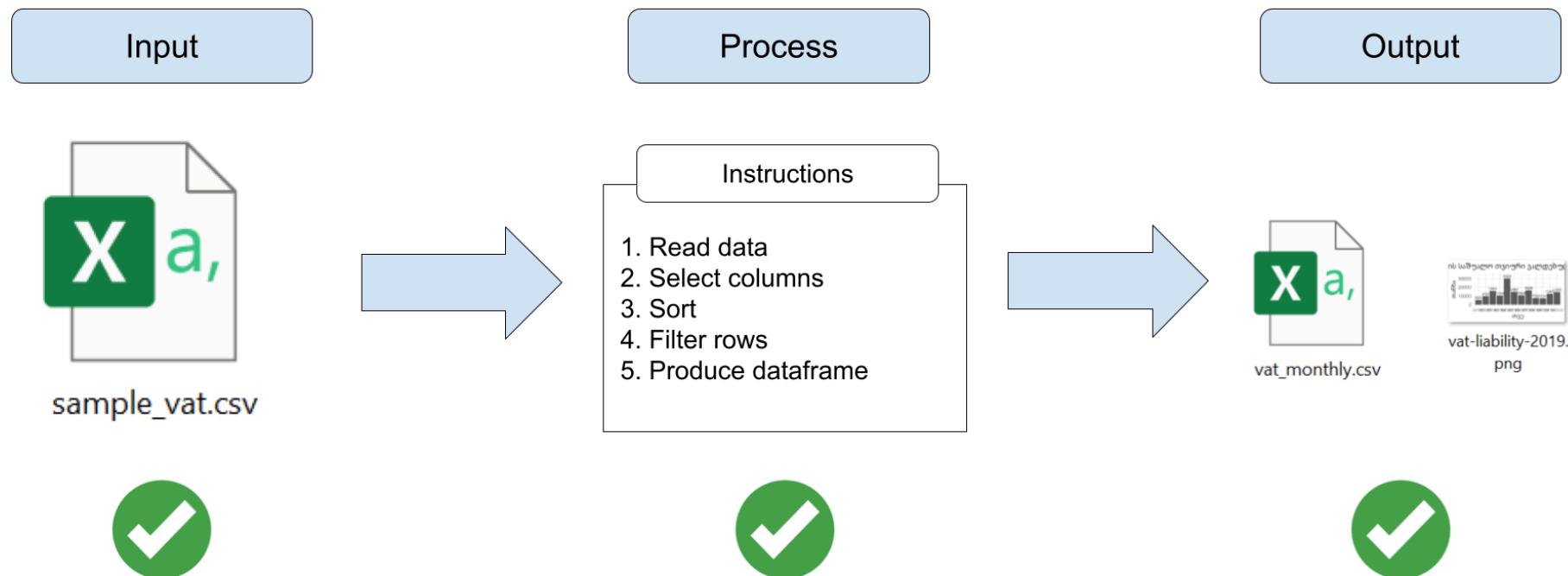
The screenshot shows an RStudio interface with an R script file open. The title bar says "exercises-session2.R x". The toolbar includes icons for back, forward, source on save (which is highlighted), search, and run. The main code area contains the following R code:

```
1 # Exercise 3
2 library(dplyr)
3 library(tidyr)
4
5 # Exercise 5
6 temp1 <- filter(small_business_2019, region == "Tbilisi")
7 temp2 <- arrange(temp1, -income)
8 df_tbilisi_50 <- filter(temp2, row_number() <= 50)
9
10 # Exercise 6
11 temp1 <- select(small_business_2019, modified_id, income)
12 temp2 <- inner_join(temp1, small_business_2019_age, by = "modified_id")
13 temp3 <- filter(temp2, age > 5)
14 total_income <- colsums(select(temp3, income))
15
```

The status bar at the bottom shows "15:1 (Top Level) R Script".

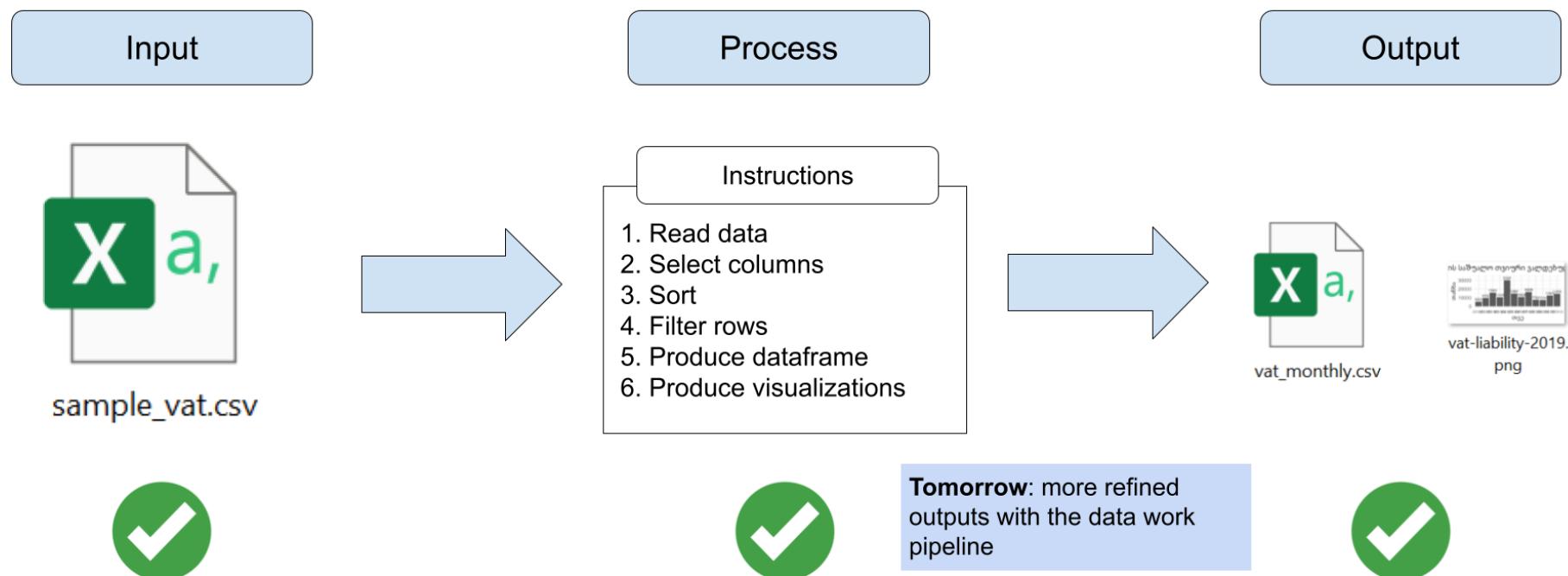
Wrapping up // მიმკვეთვა

Data work pipeline



Wrapping up // მიმკვეთვა

Data work pipeline



Appendix // დანართი

Appendix // ፳፻፲፭

Collapsing

Scenario: Imagine you're approached with the following request:

"We're putting together a report where we want to include the total and average income of small businesses by region in 2019. Can you calculate these numbers? There is data for small businesses income in 2019 you can use for this."

Appendix // დანართი

Collapsing

"There is data for small businesses income in 2019 you can use for this"

"we want to include the total and average income of small businesses by region in 2019"

Appendix // ወሰኩመን

Collapsing

The data wrangling here involves a number of operations:

1. Keeping only the relevant columns and dropping everything else
2. Grouping the dataframe by region
3. Calculating the total income by region
4. Calculating the mean income by region

The operation of transforming a dataframe with a lower level of observations (exm: firms) to an aggregated level (exm: regions) is called **collapsing**.

Appendix // የሰነድዎች

1. Keeping only relevant columns

Use `select()` for this:

```
temp1 <- select(small_business_2019, region, income)
```

Appendix // დანართი

2. Grouping by region

Use `group_by()`:

```
temp2 <- group_by(temp1, region)
```

Appendix // የሰነድዎች

3. Calculating aggregated columns: total and average income

Use `summarize()` combined with `sum()` and `mean()`:

```
region_df <- summarize(temp2,
  total = sum(income),
  average = mean(income))
```

Appendix // የሰነድዎች

Exercise: Collapse your data

Now we can write code to execute the data collapsing.

1. Columns selection: `temp1 <- select(small_business_2019, region, income)`
2. Grouping by region: `temp2 <- group_by(temp1, region)`
3. Calculating the total and average by month:

```
region_df <- summarize(temp2,  
                      total = sum(income),  
                      average = mean(income))
```

Some notes:

- Once again, remember that these are all functions from `dplyr`
- There is a line break and tabulation space between each argument of `summarize()`. We use this for code clarity, R ignores line breaks and space when they are used between function arguments

Appendix // የመሆኑ

Collapsing

You can check your result with `View(region_df)`. Now your data as collapsed as you needed!