

Session 1 - Introduction to data work in R

R training - Georgia RS-WB DIME

Marc-Andrea Fiorina, Luis Eduardo San Martin

The World Bank | [WB Github](#)

April 2023



Table of contents // სარჩევი

1. About the training
2. Data work
3. Statistical programming
4. Writing R code
5. Data in R
6. Functions in R
7. Wrapping up

About this training // ამ ტრენინგის შესახებ

About this training // ამ ტრენინგის შესახებ

- This is an **introduction** to data work and statistical programming in R
- The course does not require any background in statistical programming
- The course requires a computer with R and RStudio installed

About this training // ამ ტრენინგის შესახებ

Learning objectives

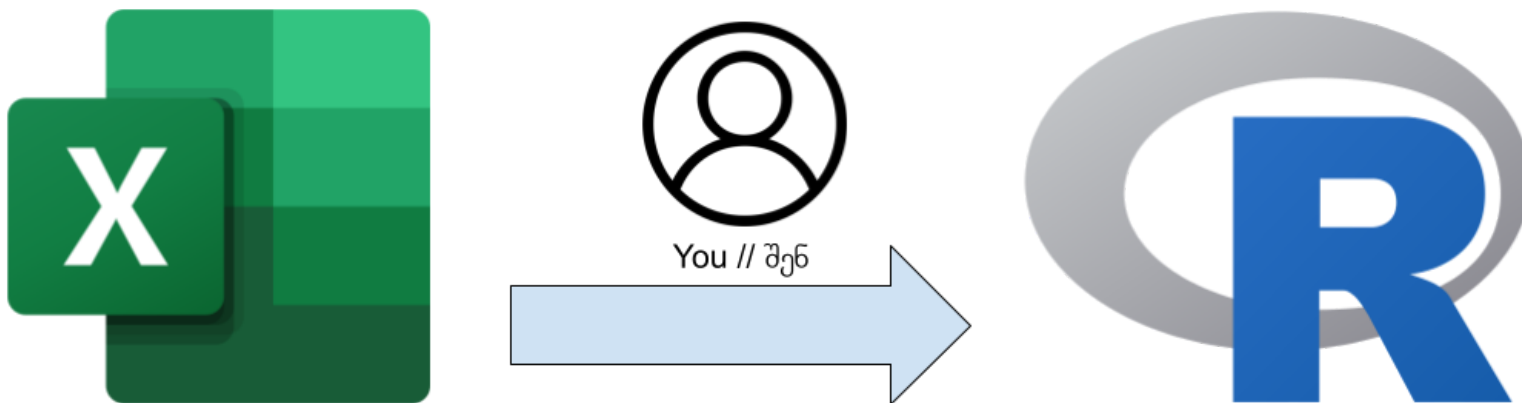
By the end of the training, you will know:

- How to write **basic** R code
- A notion of how to conduct data work in R and how it differentiates from Excel

About this course // ამ კურსის შესახებ

Previous knowledge

- We assume that you have some experience working with data in Excel
- The idea of this short training is to use that knowledge and "translate" some of it to R

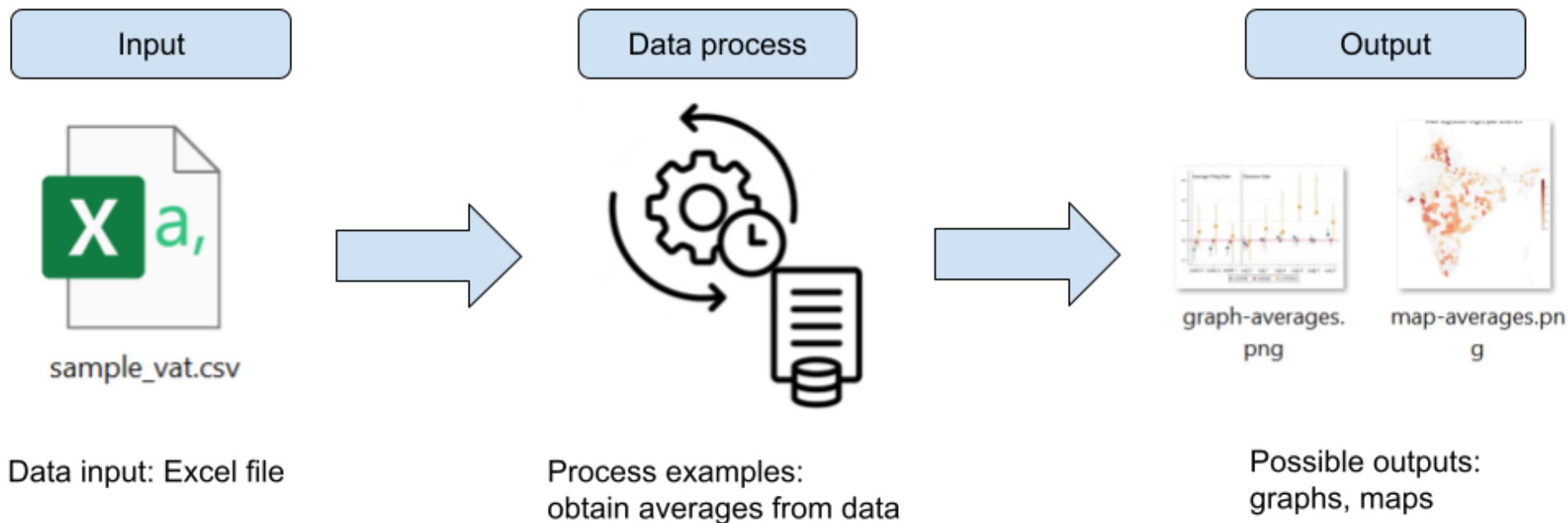


Data work // მონაცემთა მუშაობა

Data work // მონაცემთა მუშაობა

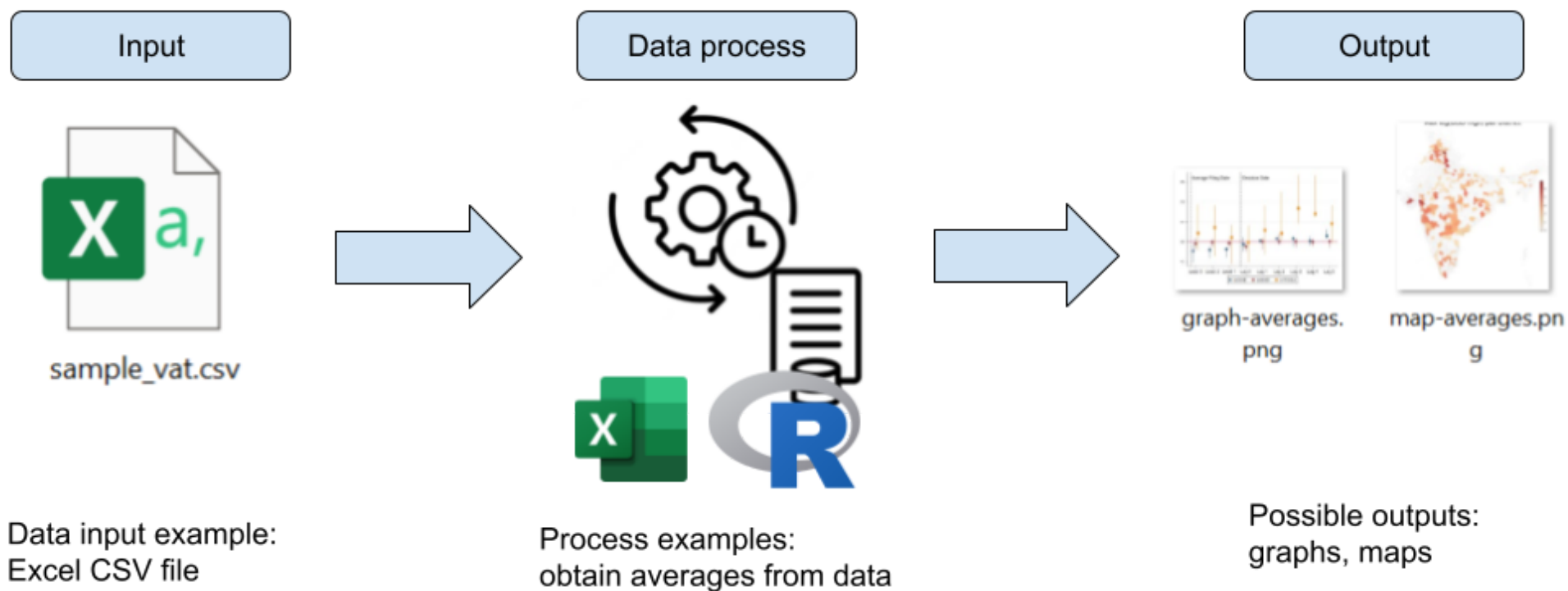
For the context of this training, we'll call data work everything that:

1. Starts with a data input
2. Runs some process with the data
3. Produces an output with the result



Data work // მონაცემთა მუშაობა

- It's also possible to do data work with Excel
- However, we will show in this training why using statistical programming (through R) is a better way of conducting data work

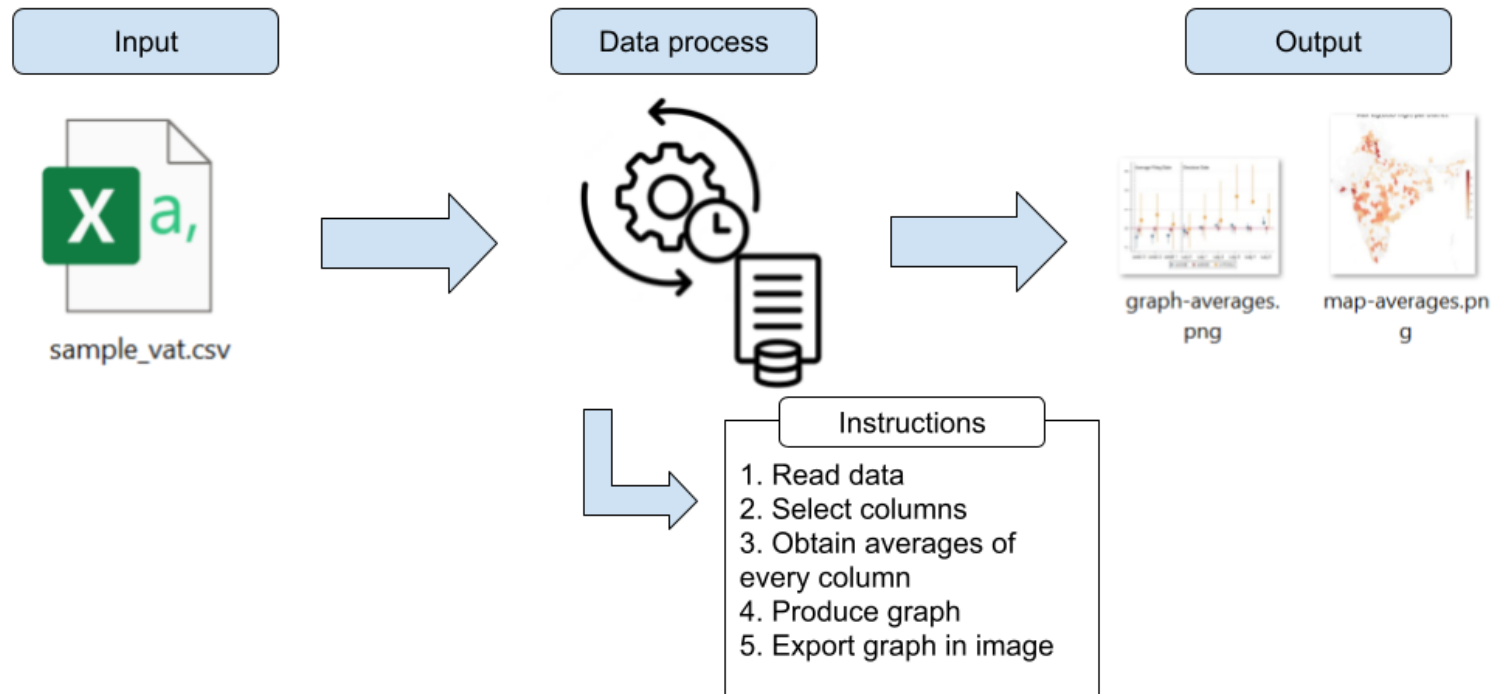


Statistical programming // სტატისტიკური პროგრამირება

Statistical programming

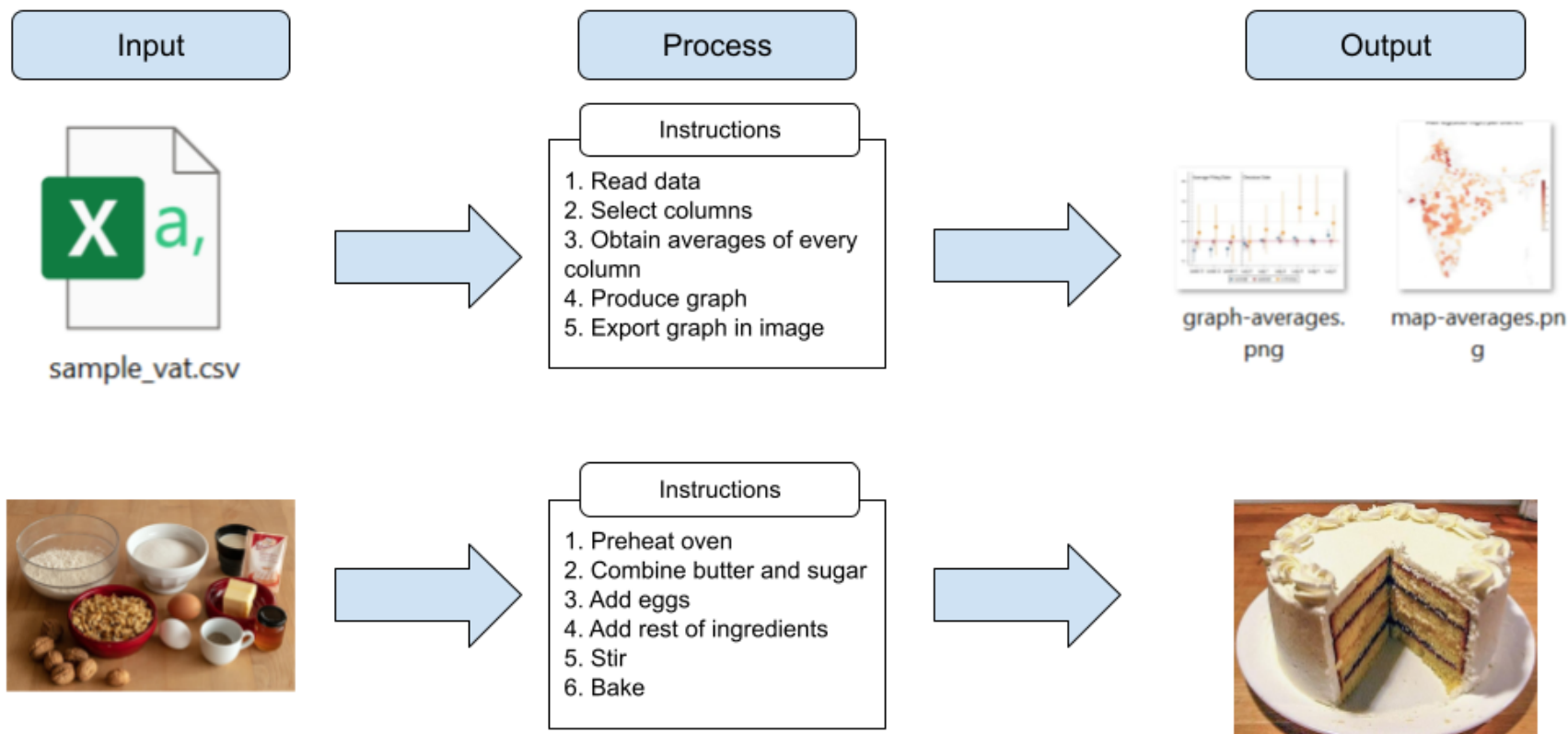
What do we mean by statistical programming?

- Programming consists of producing instructions to a computer to do something
- In the context of data work, that "something" is statistical analysis or mathematical operations
- Hence, statistical programming consists of producing instructions so our computers will conduct statistical analysis on data



Statistical programming

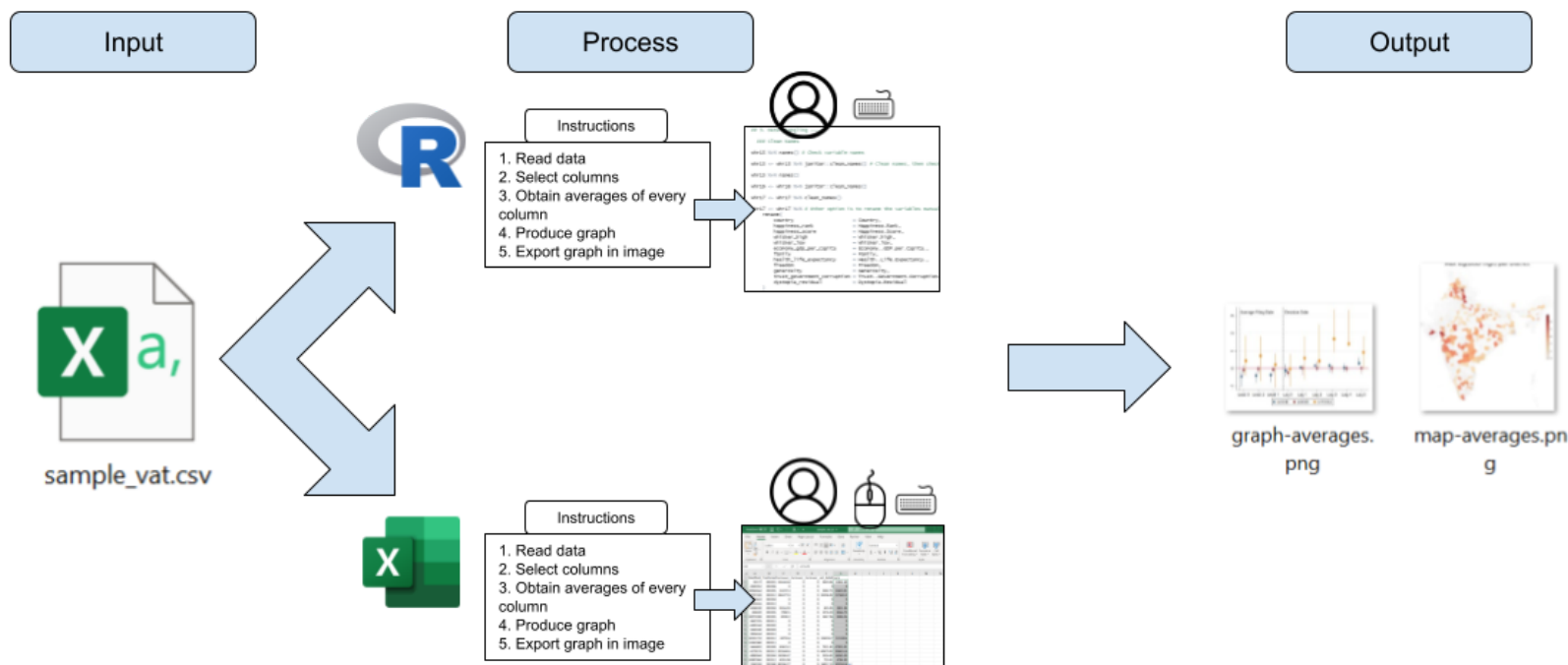
- You can think of statistical programming as writing a recipe



Statistical programming

How is R different than Excel?

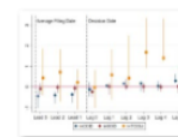
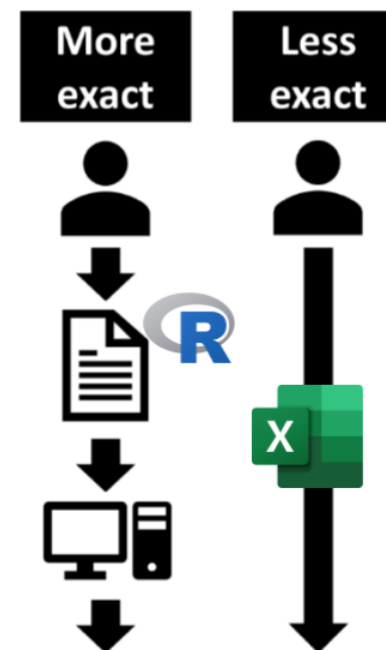
- In Excel, you usually work on the data directly. The input changes as you work on it
- In R, you produce code which contains the instructions your computer follows to conduct the process you want to do



Statistical programming

How can this benefit my work?

- Data work should be very exact to be correct
- Programming consists of giving computers "instructions" about what to do with data. This is more exact than manually working directly with the data, as we do in Excel
- Computers are **very exact**. In programming, they will do exactly what you tell them to do
- This means that you can generate code that is an exact record of how a result was generated



graph-averages.
png

Statistical programming

Why use R

- Statistical programming can be implemented through many different software. Other options are Stata and Python
- We recommend using R for these reasons:
 - R is free
 - R was designed specifically for statistical programming
 - There is a large worldwide community of R users. This means you can easily look for help or examples of code in the internet



Statistical programming

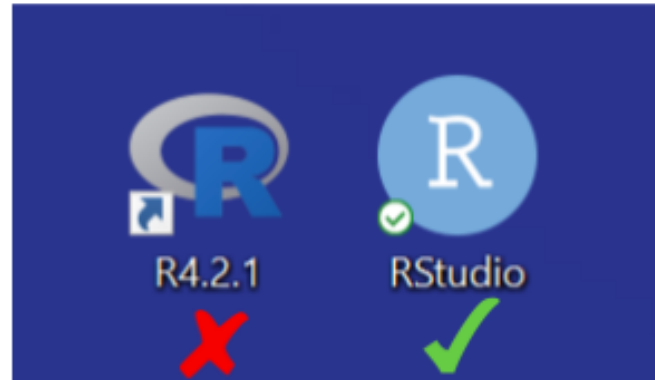
How to write R code?

- The rest of today's session focuses on the basics of writing R code
- We'll use RStudio to write R code in this training

Statistical programming

How to write R code?

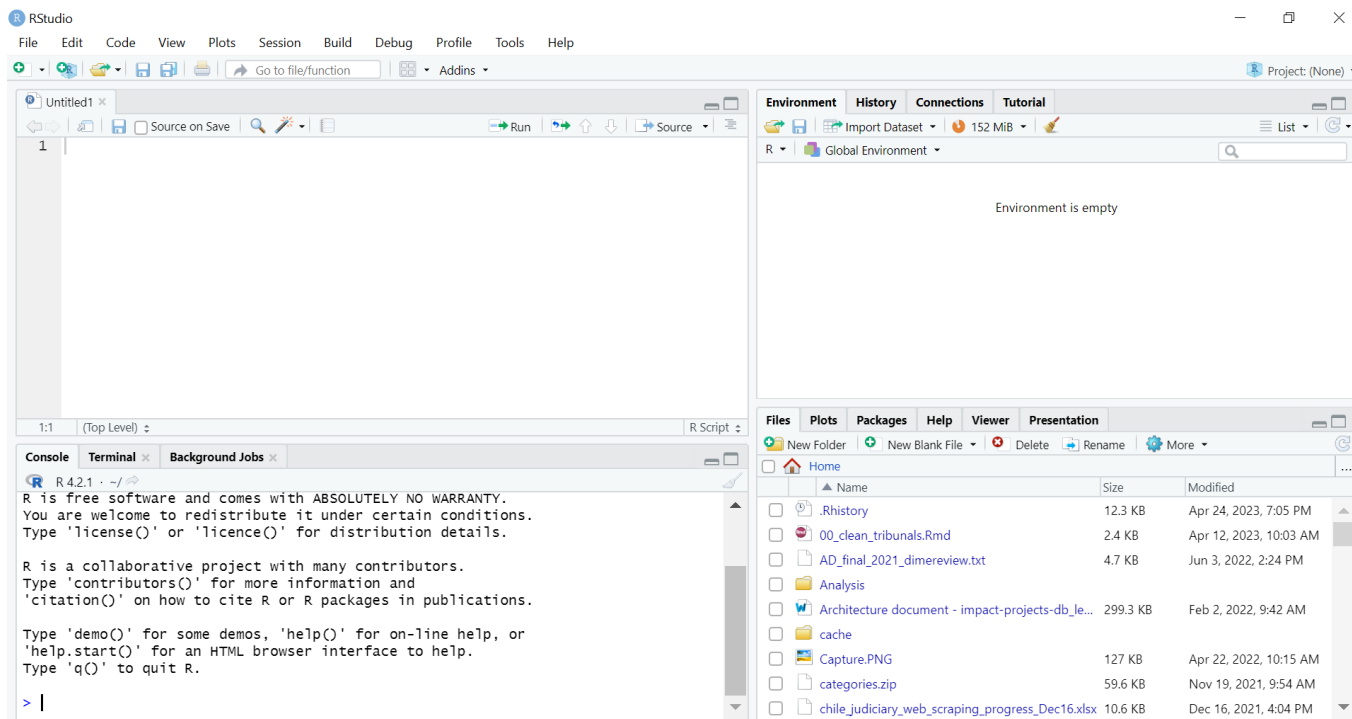
- Now open RStudio in your computer
- Please make sure you're opening RStudio and not R



Statistical programming

How to write R code?

- Now open RStudio in your computer
- Please make sure you're opening RStudio and not R

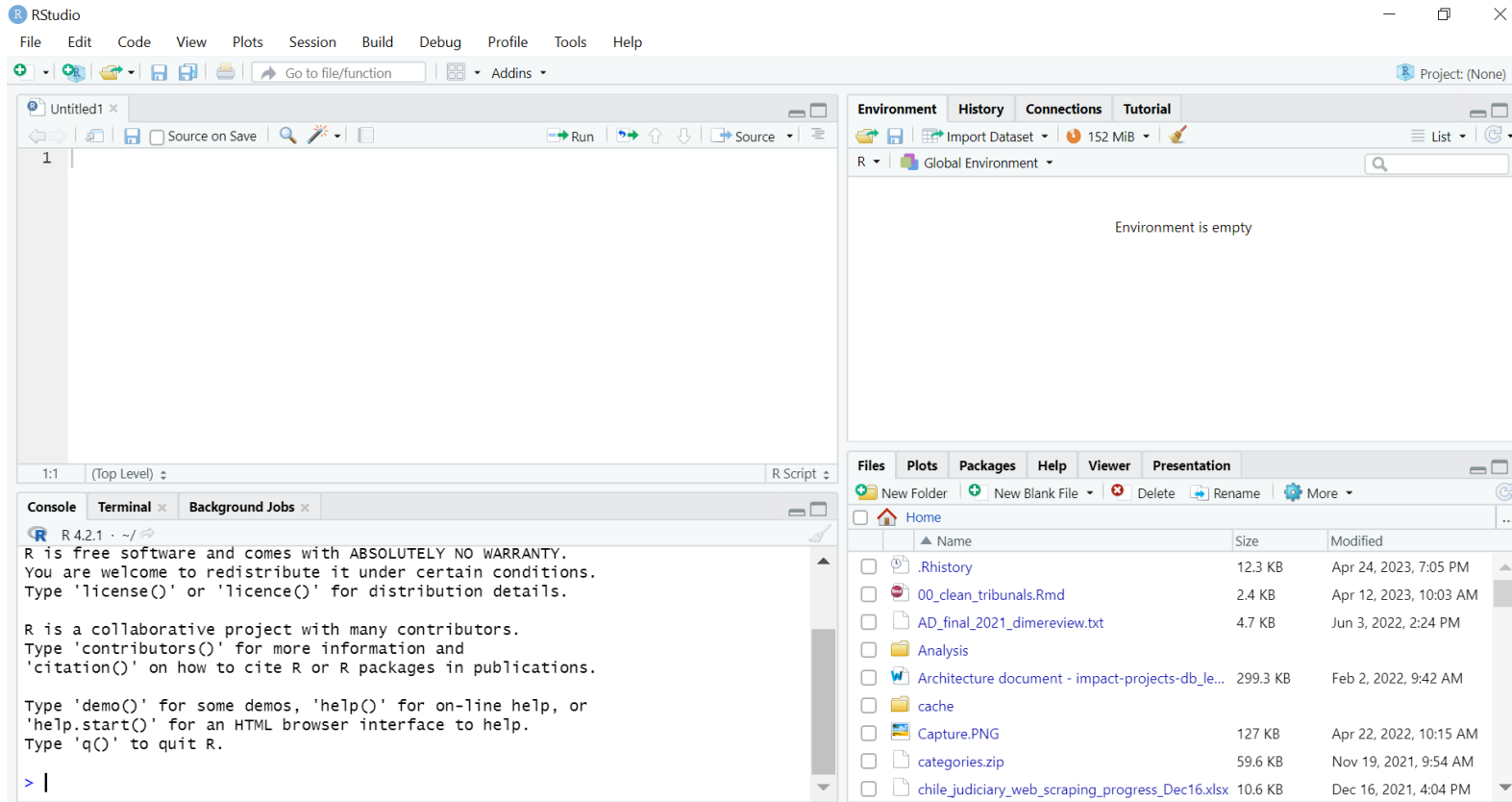


Questions? // კითხვები?

Writing R code // R კოდის დაწერა

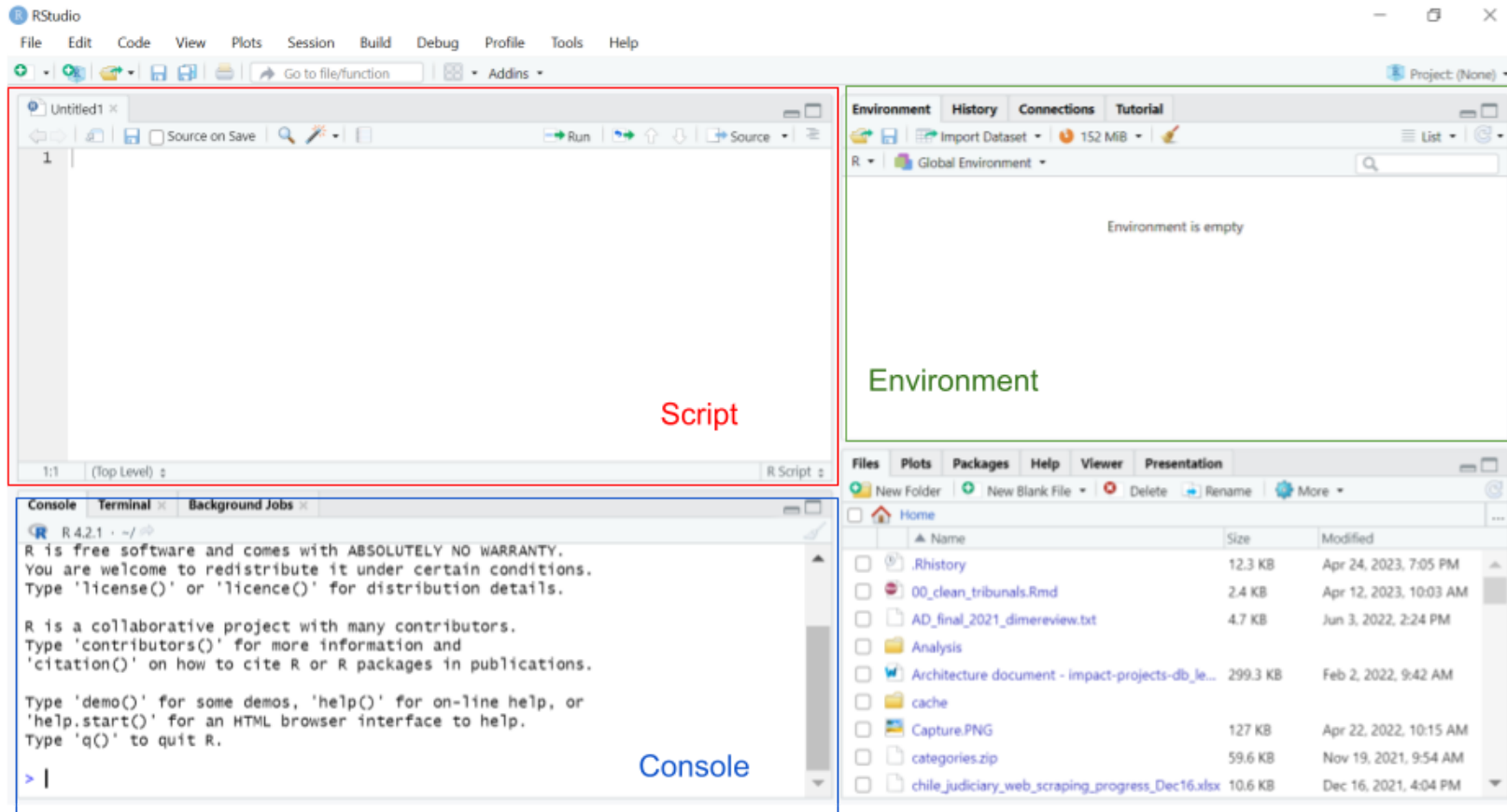
Writing R code // R კოდის დაწერა

RStudio interface



Writing R code // R კოდის დაწერა

RStudio interface



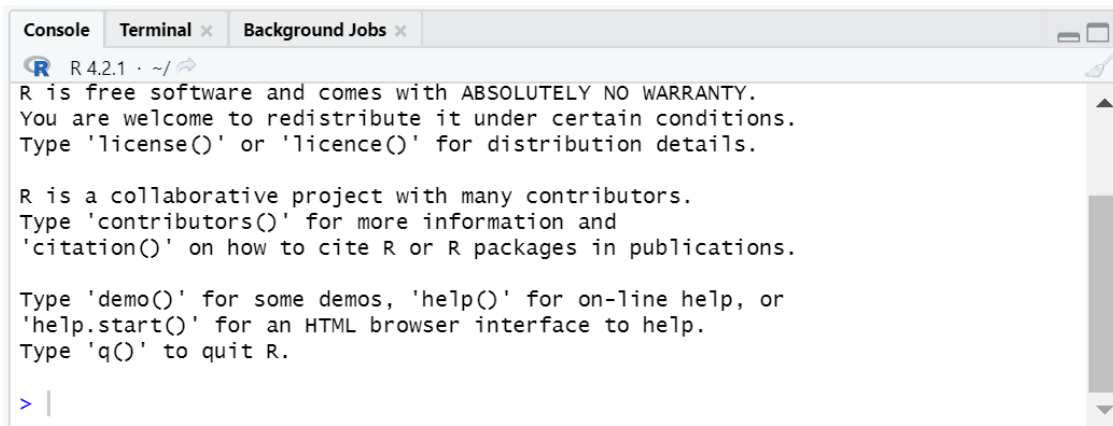
Writing R code // R კოდის დაწერა

Exercise 1: writing code in the console

1. Write the following code in the console of RStudio

- `print("gamarjoba")`
- Make sure to include the quotes: " "

2. Press Enter to run the code



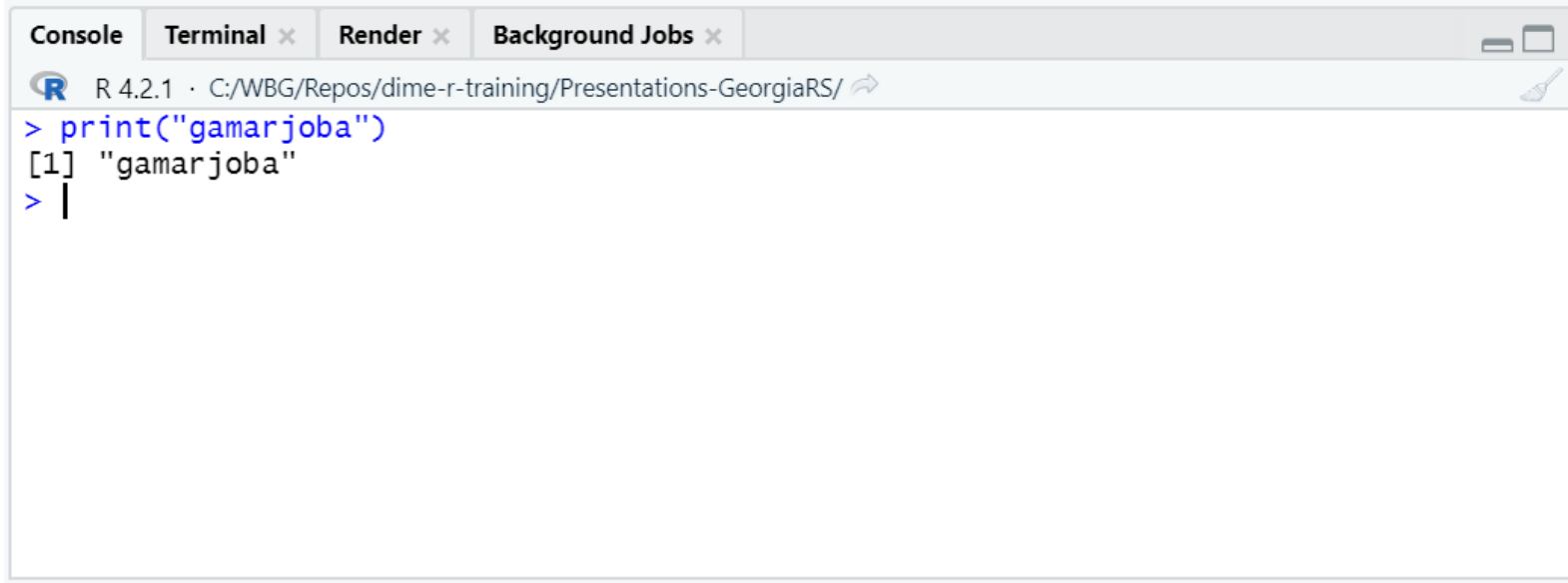
```
R 4.2.1 · ~/
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```

Writing R code // R კოდის დაწერა



The screenshot shows an R console window with the following elements:

- Tab bar: Console (selected), Terminal x, Render x, Background Jobs x.
- Header bar: R 4.2.1 · C:/WBG/Repos/dime-r-training/Presentations-GeorgiaRS/
- Code input: `> print("gamarjoba")`
- Output: `[1] "gamarjoba"`
- Next prompt: `> |`

Writing R code // R კოდის დაწერა

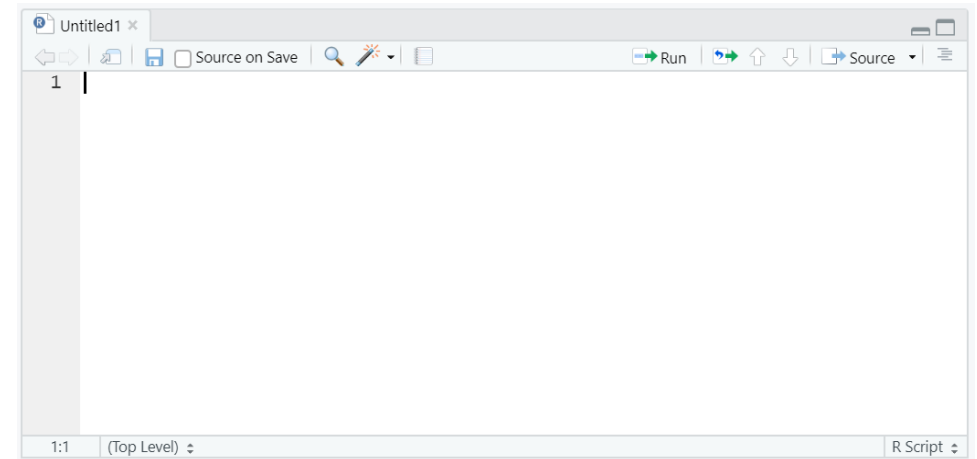
Exercise 2: writing a short script

1. Write or copy the following text into the script section of RStudio

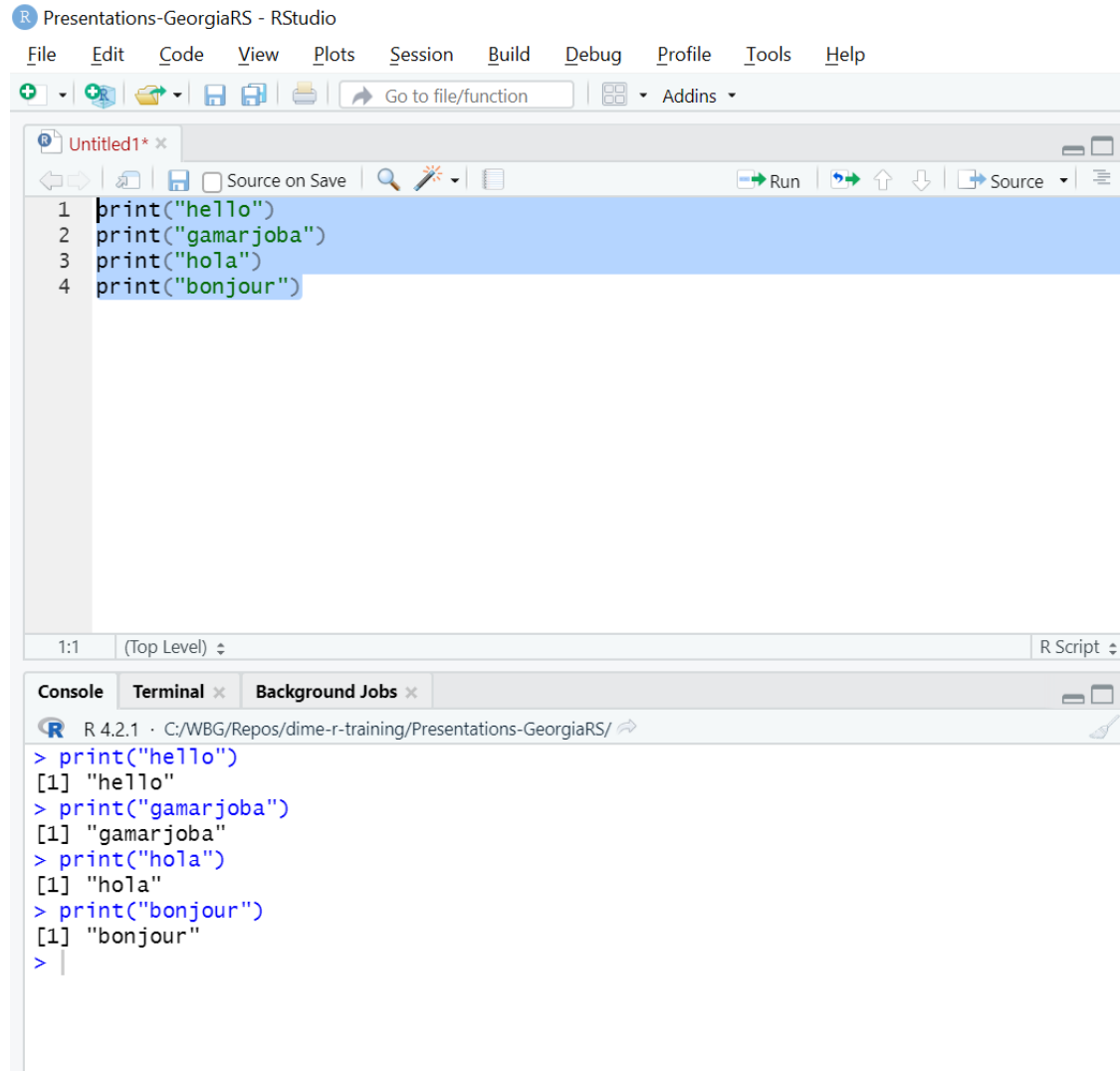
- `print("gamarjoba")`
- `print("hello")`
- `print("hola")`
- `print("bonjour")`

2. Select the text you introduced with your mouse

3. Press "Run"



Writing R code // R კოდის დაწერა



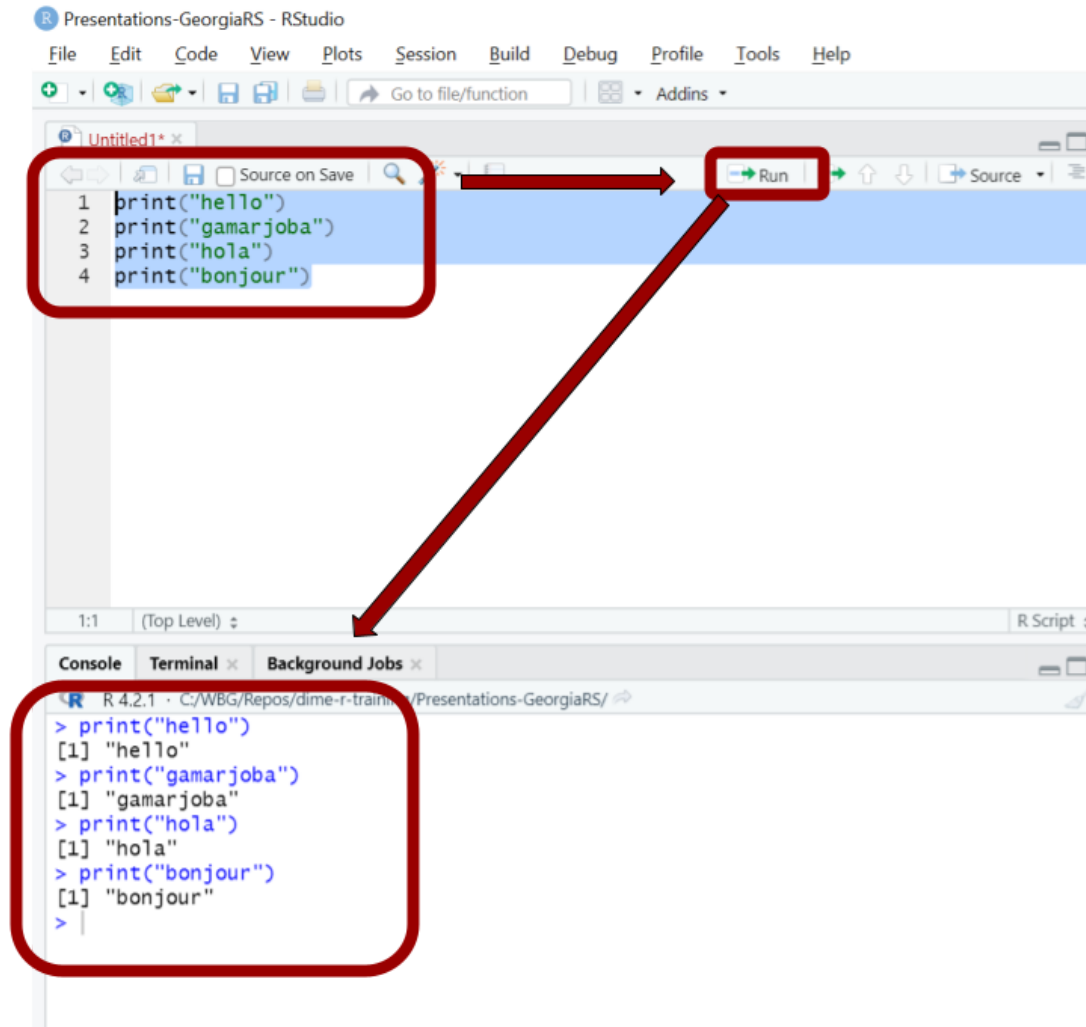
The screenshot displays the RStudio interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu is a toolbar with icons for file operations and a 'Go to file/function' search bar. The main editor window, titled 'Untitled1*', contains the following R code:

```
1 print("hello")
2 print("gamarjoba")
3 print("hola")
4 print("bonjour")
```

The code is highlighted in blue. Below the editor is a console window with tabs for Console, Terminal, and Background Jobs. The console shows the execution of the code, with each line of code followed by its output:

```
> print("hello")
[1] "hello"
> print("gamarjoba")
[1] "gamarjoba"
> print("hola")
[1] "hola"
> print("bonjour")
[1] "bonjour"
> |
```

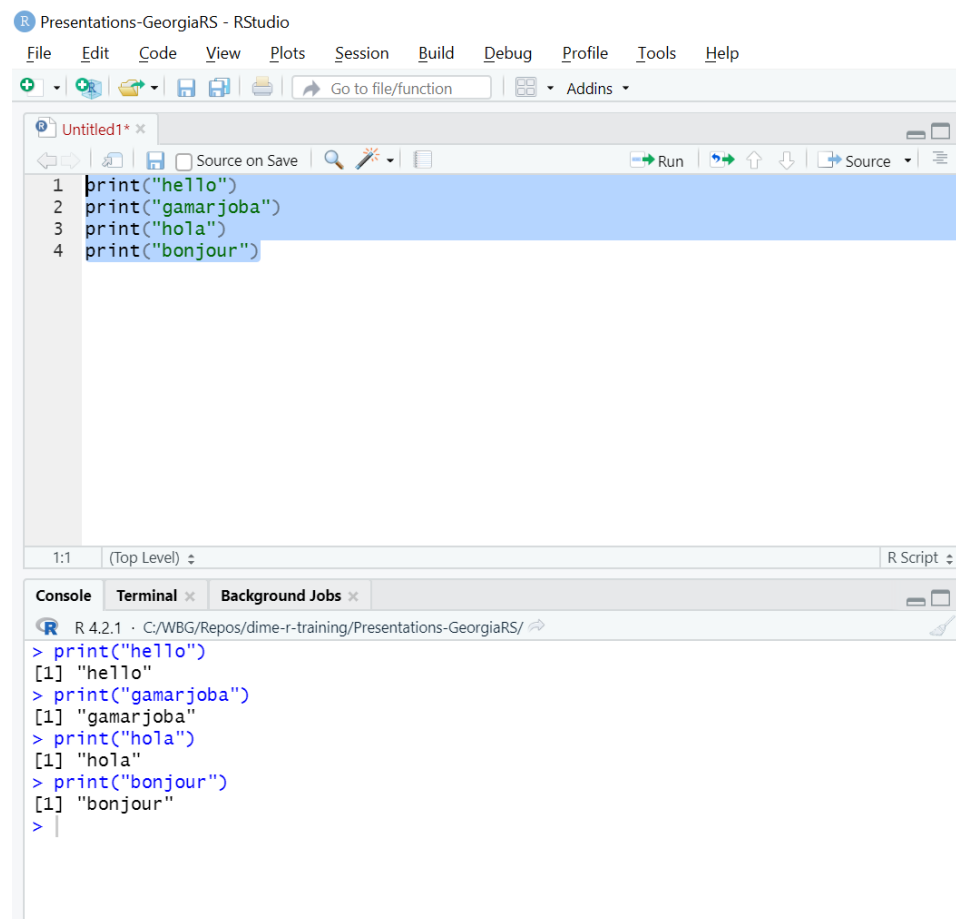
Writing R code // R კოდის დაწერა



Writing R code // R კოდის დაწერა

R scripts

- Writing and running code from the console will execute it immediately
- Writing code in the script panel allow us to write multiple lines of code and execute them later
 - Each line is executed in order
 - The line and the results will show in the console
- **Important:** for the rest of the training, remember to always introduce your code in the script (and not in the console) so you can keep record of what you did

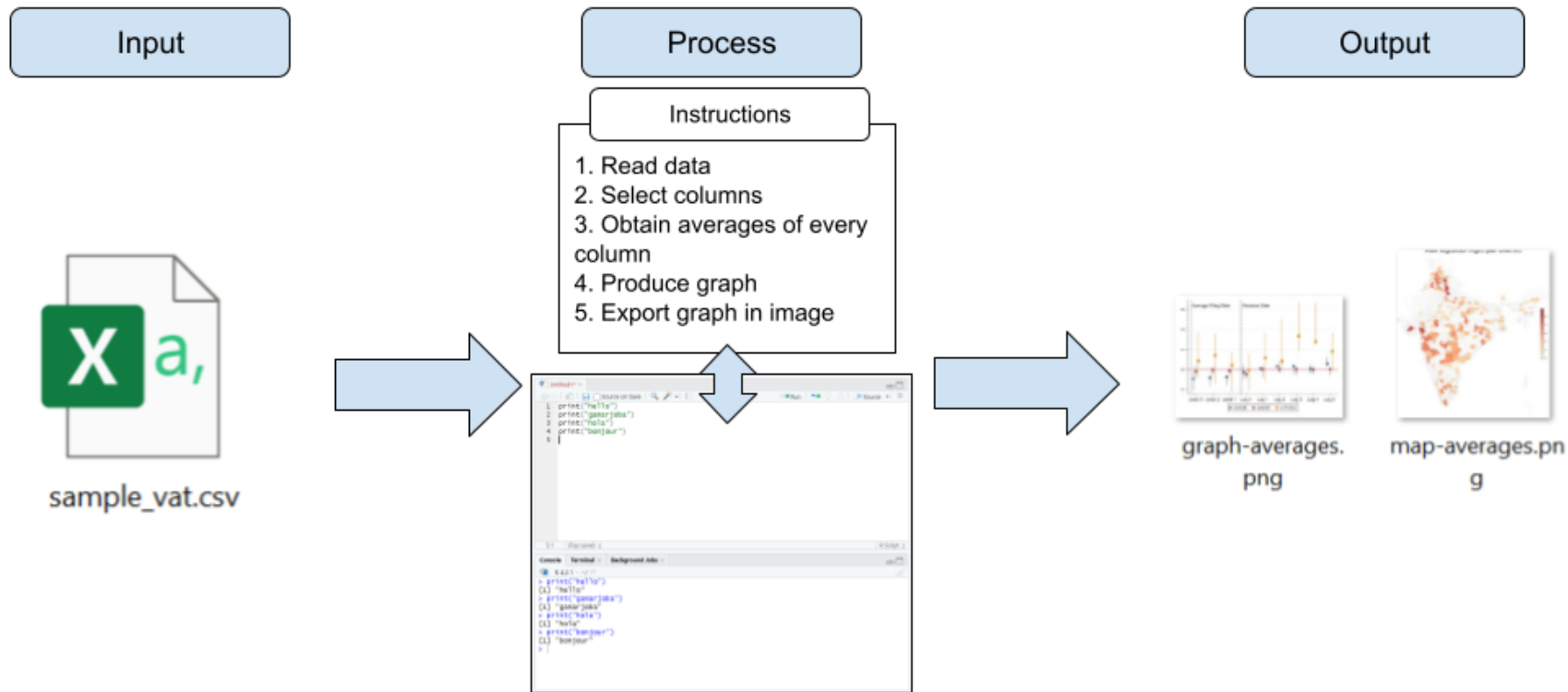


The screenshot displays the RStudio environment. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu is a toolbar with icons for file operations and execution. The main script editor window, titled 'Untitled1*', contains four lines of R code: `1 print("hello")`, `2 print("gamarjoba")`, `3 print("hola")`, and `4 print("bonjour")`. The bottom panel shows the 'Console' tab, which displays the output of the executed code: `> print("hello")` resulting in `[1] "hello"`, `> print("gamarjoba")` resulting in `[1] "gamarjoba"`, `> print("hola")` resulting in `[1] "hola"`, and `> print("bonjour")` resulting in `[1] "bonjour"`. The status bar at the bottom indicates the R version is 4.2.1 and the current working directory is `C:/WBG/Repos/dime-r-training/Presentations-GeorgiaRS/`.

Writing R code // R კოდის დაწერა

R scripts

- In other words: scripts contain the instructions you give to your computer when doing data work



Creating objects in R

- Remember we also mentioned the environment panel? that's where R keeps track of objects
- Objects are representations of data that currently exist in R's memory
 - A single number can be an object
 - A word can be an object
 - Even an entire data file can be an object
- We create objects in R with the arrow operator (`<-`)
- Example: creating an object called `x`

```
x <- 10
```

Creating objects in R

- After an object is created, we can refer to it using its name

```
x <- 10  
print(x)
```

```
## [1] 10
```

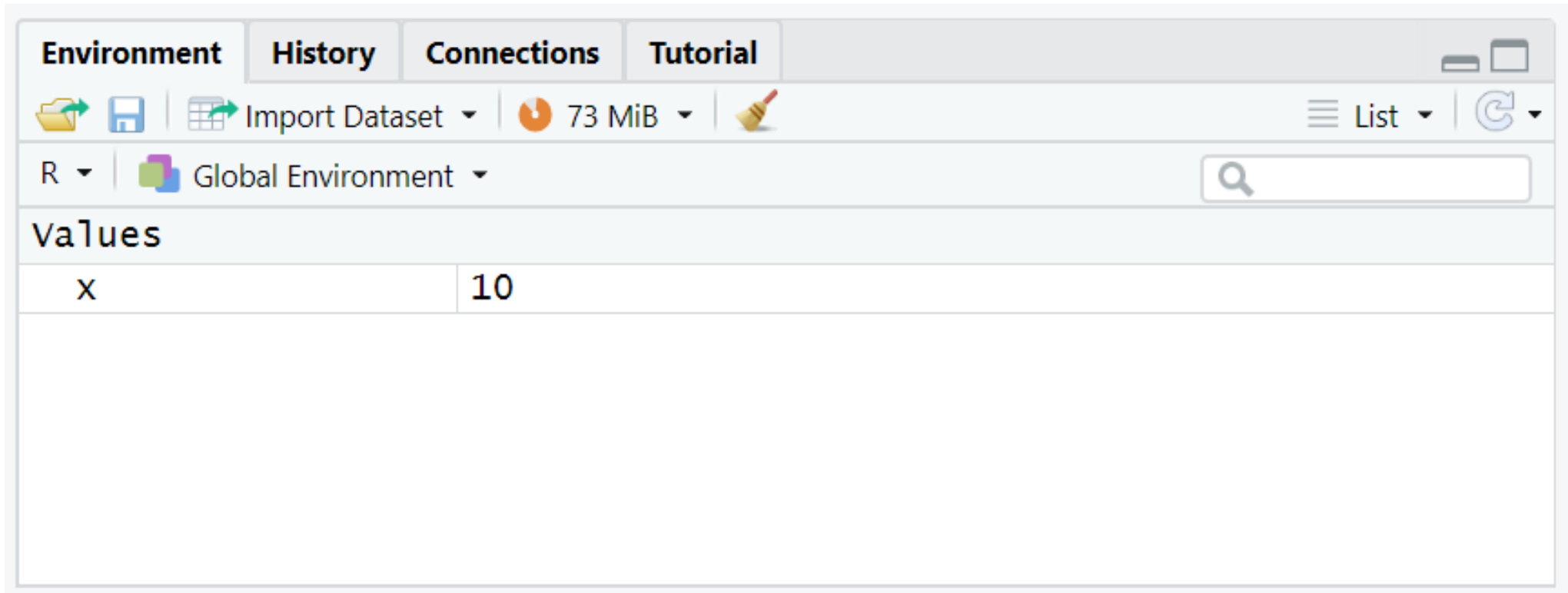
```
print(x + 5)
```

```
## [1] 15
```

Writing R code // R კოდის დაწერა

Creating objects in R

- After an object is created, it will show in the environment panel



Exercise 3: operations with objects

1. Create an object called `x1` in your script and assign to it the value of 50
2. Create an object called `x2` and make it equal to 15
3. Create an object called `x3` and make it the sum of `x1` and `x2`
4. Print `x3`. It should show the value 65

Writing R code // R კოდის დაწერა

Presentations-GeorgiaRS - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Untitled1*

```
1 print("hello")
2 print("gamarjoba")
3 print("hoia")
4 print("bonjour")
5
6 x1 <- 50
7 x2 <- 15
8 x3 <- x1 + x2
9 print(x3)
```

Environment History Connections Tutorial

Import Dataset 115 MiB

R Global Environment

Values

x1	50
x2	15
x3	65

Files Plots Packages Help Viewer Presentation

New Folder New Blank File Delete Rename More

C: > WBG > Repos > dime-r-training > Presentations-GeorgiaRS

	Name	Size	Modified
	..		
<input type="checkbox"/>	.Rhistory	150 B	Apr 25, 2023, 11:42 AM
<input type="checkbox"/>	data		
<input type="checkbox"/>	img		
<input type="checkbox"/>	libs		
<input type="checkbox"/>	Presentations-GeorgiaRS.Rproj	222 B	Apr 25, 2023, 12:50 PM
<input type="checkbox"/>	README.md	660 B	Apr 24, 2023, 4:06 PM
<input type="checkbox"/>	script1.R	86 B	Apr 25, 2023, 12:48 PM
<input type="checkbox"/>	session1_cache		
<input type="checkbox"/>	session1.html	16.3 KB	Apr 25, 2023, 2:13 PM
<input type="checkbox"/>	session1.Rmd	10.7 KB	Apr 25, 2023, 2:12 PM

Console Terminal Background Jobs

R 4.2.1 · C:/WBG/Repos/dime-r-training/Presentations-GeorgiaRS/

```
> x1 <- 50
> x2 <- 15
> x3 <- x1 + x2
> print(x3)
[1] 65
>
```

Writing R code // R კოდის დაწერა

The screenshot displays the RStudio interface with the following components:

- Source Editor:** Contains R code for printing messages and calculating the sum of two variables. Lines 6-9 are highlighted in blue.
- Run Button:** A red box highlights the 'Run' button in the toolbar, with an arrow pointing to it from the highlighted code.
- Environment Pane:** A red box highlights the 'Values' table, showing the results of the code execution.
- Console:** A red box highlights the output of the code execution in the console.

Code in Source Editor:

```
1 print("hello")
2 print("gamarjoba")
3 print("hoia")
4 print("honour")
5
6 x1 <- 50
7 x2 <- 15
8 x3 <- x1 + x2
9 print(x3)
```

Environment Pane Values:

Variable	Value
x1	50
x2	15
x3	65

Console Output:

```
> x1 <- 50
> x2 <- 15
> x3 <- x1 + x2
> print(x3)
[1] 65
>
```

Writing R code // R კოდის დაწერა

- Now we know how to use RStudio to write R code and produce scripts
- However, we haven't still introduced the data to our data work. That comes next

Data in R // მონაცემები R

Exercise 4: Loading data into R

1 - Go to this page: <https://osf.io/ds5w4> and download the file `sample_vat.csv`

The screenshot shows the OSFHOME interface. At the top, there's a navigation bar with 'OSFHOME' and a dropdown arrow, followed by links for 'Search', 'Support', 'Donate', 'Sign Up', and 'Sign In'. Below the navigation bar, the page title is 'R training Georgia RS - World Bank'. The main content area displays the file 'sample_vat.csv'. To the right of the file name, there's a menu with options: 'Download' (highlighted in yellow), 'Embed', and 'Share'. Below the file name, there's a 'Sheet 1' tab. A table is shown with the following data:

Modified_ID	TaxPeriod	turnover_taxable...	turnover_exempt	turnover_exports	vat_liability
22172	201901	13444.64	0	0	2420.04
4592952	201906	0	0	0	0
10046564	201905	13237.3	0	0	2382.71
4797328	201912	184477.88	0	0	33206.02
5889624	201904	0	0	0	0
12233616	201912	0	0	0	0
6248240	201904	2416.93	0	0	435.05
538420	201905	7090.5	0	0	1276.29
10072288	201905	8039.2	0	0	1447.06
3447376	201911	0	0	0	0
6392344	201902	0	0	0	0

To the right of the table, there's a 'File Metadata' section with a download icon. Below it, there's a 'Project Metadata' section with the following information:

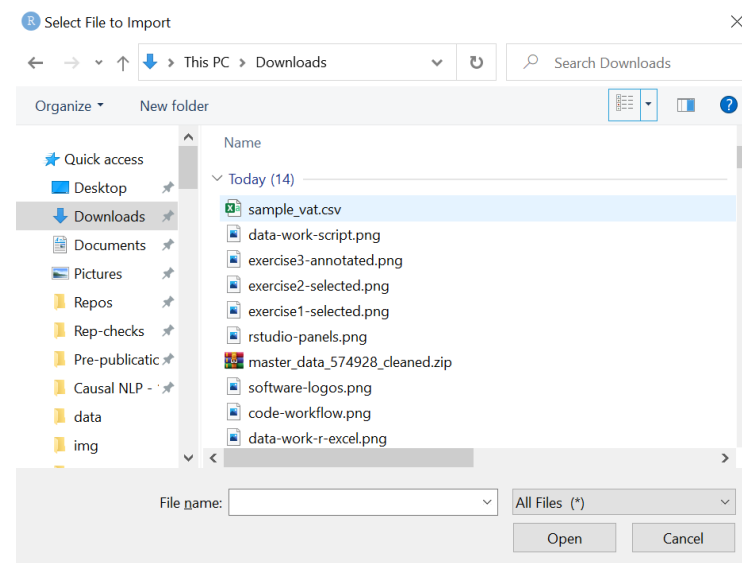
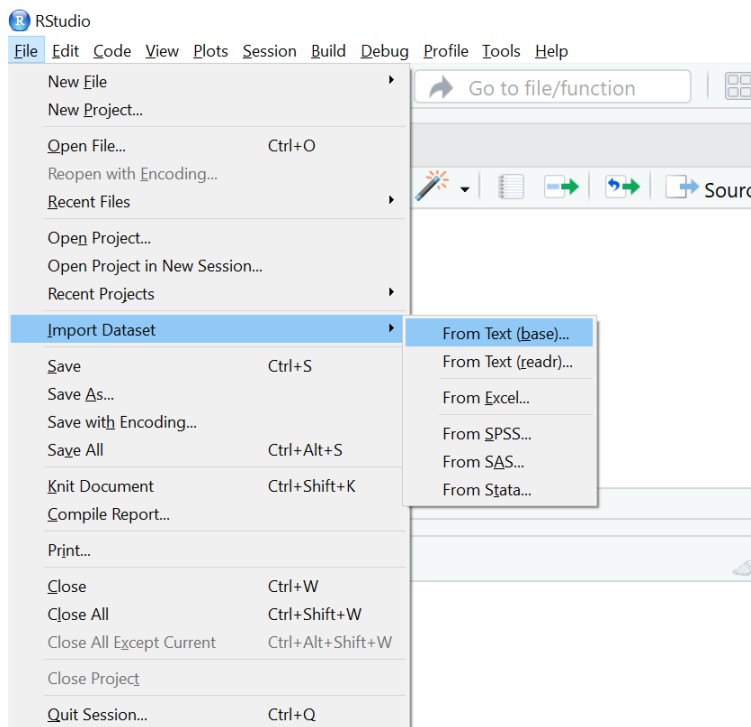
- Title**: R training Georgia RS - World Bank
- Date created**: April 25, 2023
- Date modified**: April 25, 2023
- Contributors**: DIME Analytics

Data in R // მონაცემები R

Exercise 4: Loading data into R

2 - In RStudio, go to **File** > **Import Dataset** > **From Text (base)** and select the file **sample_vat.csv**

- If you don't know where the file is, check in the **Downloads** folder



Data in R // მონაცემები R

Exercise 4: Loading data into R

3 - Make sure to select **Heading** > **Yes** in the next window

4 - Select **Import**

Import Dataset

Name: sample_vat

Input File: Modified_ID,TaxPeriod,turnover_taxable18,turnover_exempt,t
22172,201901,13444.64,0,0,2420.04
4592952,201906,0,0,0,0
10046564,201905,13237.3,0,0,2382.71
4797328,201912,184477.88,0,0,33206.02
5889624,201904,0,0,0,0
12233616,201912,0,0,0,0
6248240,201904,2416.93,0,0,435.05
538420,201905,7090.5,0,0,1276.29
10072288,201905,8039.2,0,0,1447.06
3447376,201911,0,0,0,0
6392344,201902,0,0,0,0
5545240,201903,0,0,0,0
2926244,201912,0,0,0,0
10221732,201912,857553.97,0,0,154359.71
11341060,201911,0,0,0,0
5646892,201909,40621.2,0,0,7311.82

Encoding: Automatic

Heading: ☒ Yes ☐ No

Row names: Automatic

Separator: Comma

Decimal: Period

Quote: Double (")

Comment: None

na.strings: NA

☐ Strings as factors

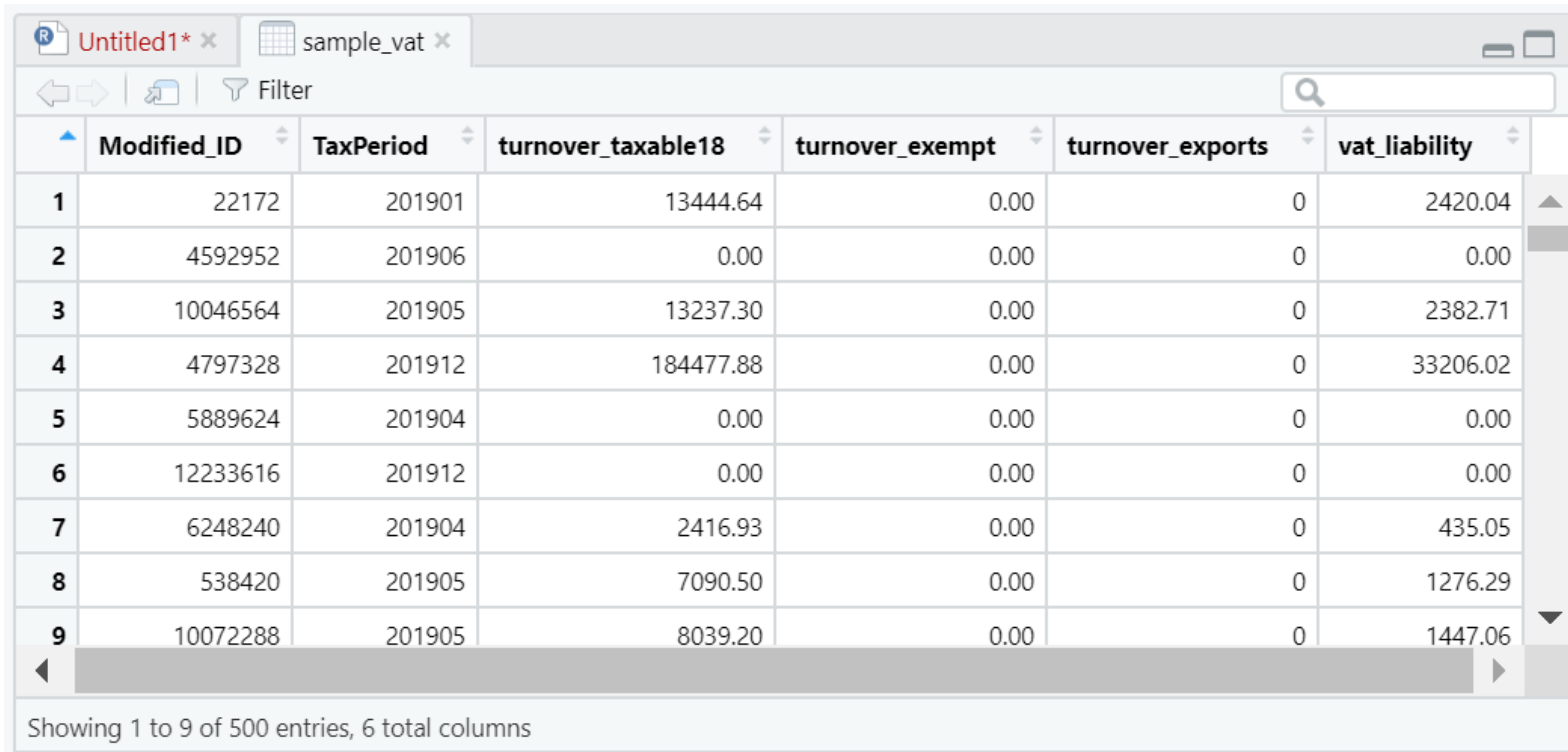
Data Frame

Modified_ID	TaxPeriod	turnover_taxable18	turnover_e
22172	201901	13444.64	0
4592952	201906	0.00	0
10046564	201905	13237.30	0
4797328	201912	184477.88	0
5889624	201904	0.00	0
12233616	201912	0.00	0
6248240	201904	2416.93	0
538420	201905	7090.50	0
10072288	201905	8039.20	0
3447376	201911	0.00	0
6392344	201902	0.00	0
5545240	201903	0.00	0
2926244	201912	0.00	0
10221732	201912	857553.97	0
11341060	201911	0.00	0
5646892	201909	40621.20	0

Import Cancel

Data in R // მონაცემები R

- If you did this correctly, you will note that a viewer of the data now appears in RStudio
- You can click on the **x** next to **sample_vat** to return to the script
- To open the viewer again, use the code: **View(sample_vat)** (notice the uppercase "V")



Untitled1* x sample_vat x

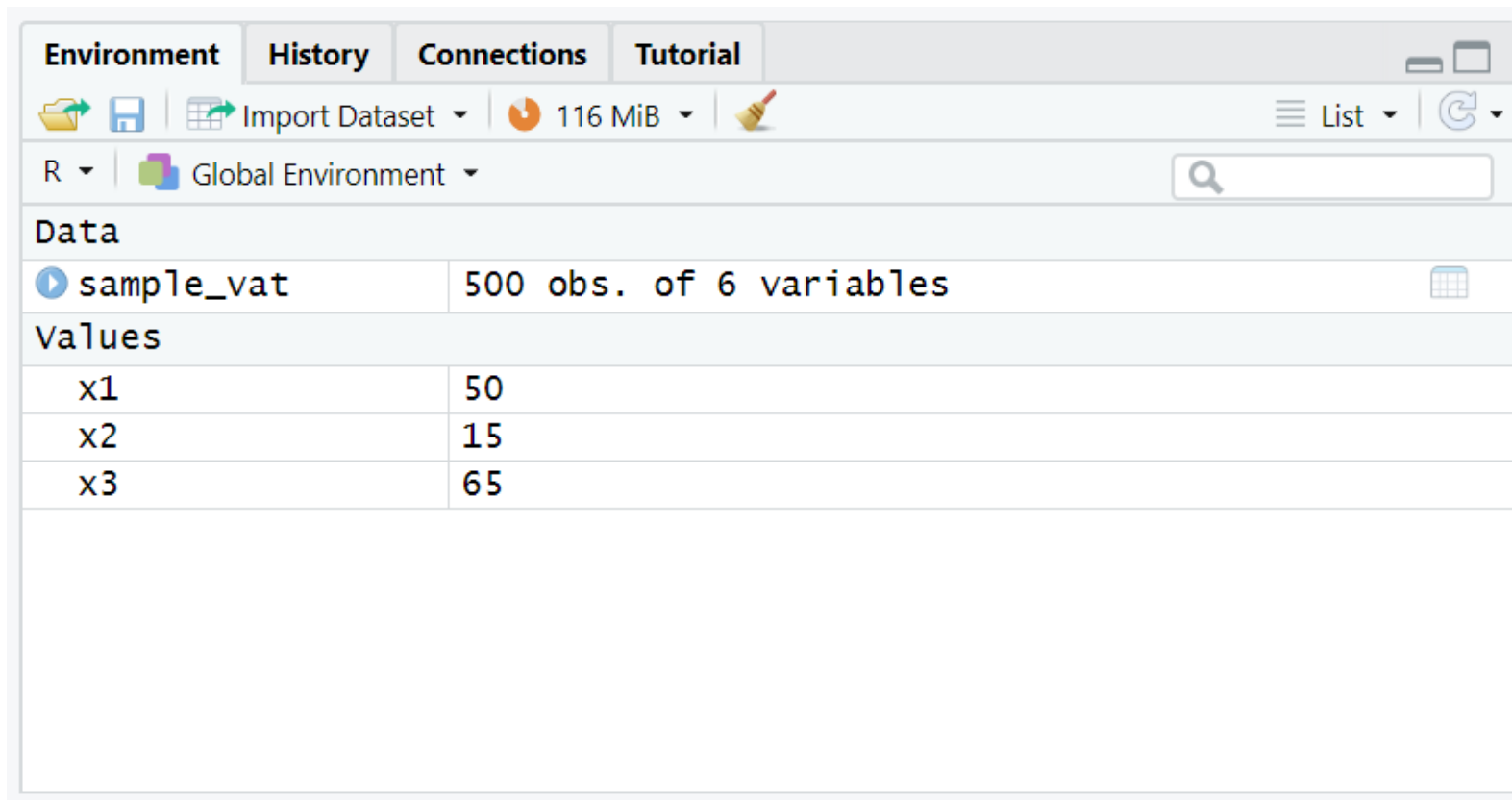
Filter

	Modified_ID	TaxPeriod	turnover_taxable18	turnover_exempt	turnover_exports	vat_liability
1	22172	201901	13444.64	0.00	0	2420.04
2	4592952	201906	0.00	0.00	0	0.00
3	10046564	201905	13237.30	0.00	0	2382.71
4	4797328	201912	184477.88	0.00	0	33206.02
5	5889624	201904	0.00	0.00	0	0.00
6	12233616	201912	0.00	0.00	0	0.00
7	6248240	201904	2416.93	0.00	0	435.05
8	538420	201905	7090.50	0.00	0	1276.29
9	10072288	201905	8039.20	0.00	0	1447.06

Showing 1 to 9 of 500 entries, 6 total columns

Data in R // მონაცემები R

- Additionally, you will now see an object named `sample_vat` in your environment

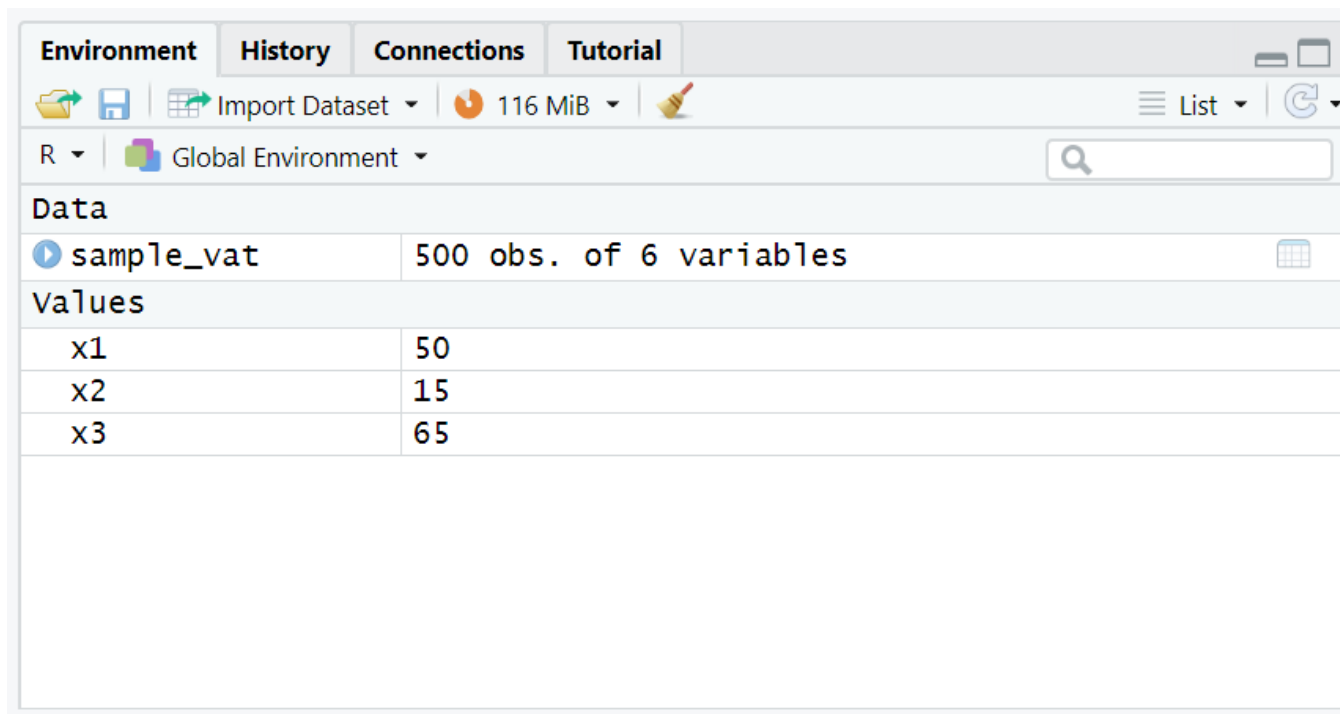


The screenshot shows the RStudio Environment pane. At the top, there are tabs for 'Environment', 'History', 'Connections', and 'Tutorial'. Below the tabs is a toolbar with icons for file operations and a status bar showing '116 MiB'. The main area of the pane displays the 'Global Environment'. Under the 'Data' section, the object 'sample_vat' is listed with the description '500 obs. of 6 variables'. Below this, under the 'Values' section, a table shows the first three values of the object:

Values	
x1	50
x2	15
x3	65

Data in R // მონაცემები R

- Remember we mentioned objects before? For R, `sample_vat` is an object just like `x1`, `x2`, or `x3`
- The difference is that `sample_vat` is not a single number like `x1`, but a collection of numeric values similar to an Excel spreadsheet. In R, this type of objects are called **dataframes**
- From now, we will refer to data loaded into R as **dataframes**



The screenshot shows the RStudio Environment pane. At the top, there are tabs for 'Environment', 'History', 'Connections', and 'Tutorial'. Below the tabs, there are icons for file operations and a status bar showing '116 MiB'. The 'Environment' pane displays the 'Global Environment' with a search bar. Under the 'Data' section, the object 'sample_vat' is listed with the description '500 obs. of 6 variables'. Below this, the 'Values' section shows a table with three rows and two columns:

x1	50
x2	15
x3	65

Data in R // მონაცემები R

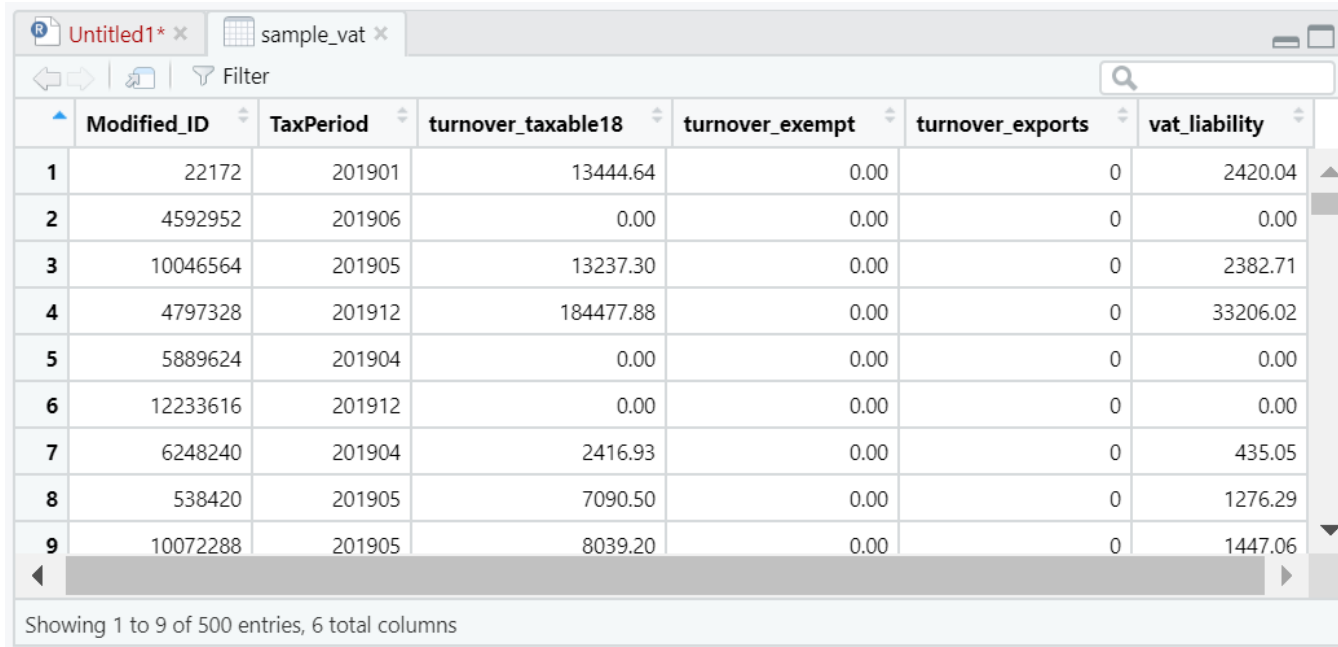
- Since dataframes are also objects, we can refer to them with their names (exm: `sample_vat`)
- We'll see an example of that in the next exercise

Data in R // მონაცემები R

A note about this dataframe

Understanding the data you use is very important. For this training, `sample_vat` is a "toy" dataframe that simulates tax data.

- `Modified_ID` is a taxpayer identifier
- `TaxPeriod` is a month variable (year + month)
- The rest are tax-related variables that we are not going to focus on in this session



The screenshot shows a RStudio window with a tab titled 'sample_vat'. Below the tab is a toolbar with navigation icons and a search bar. The main area displays a table with 6 columns: Modified_ID, TaxPeriod, turnover_taxable18, turnover_exempt, turnover_exports, and vat_liability. The table contains 500 rows, with the first 9 rows visible. The status bar at the bottom indicates 'Showing 1 to 9 of 500 entries, 6 total columns'.

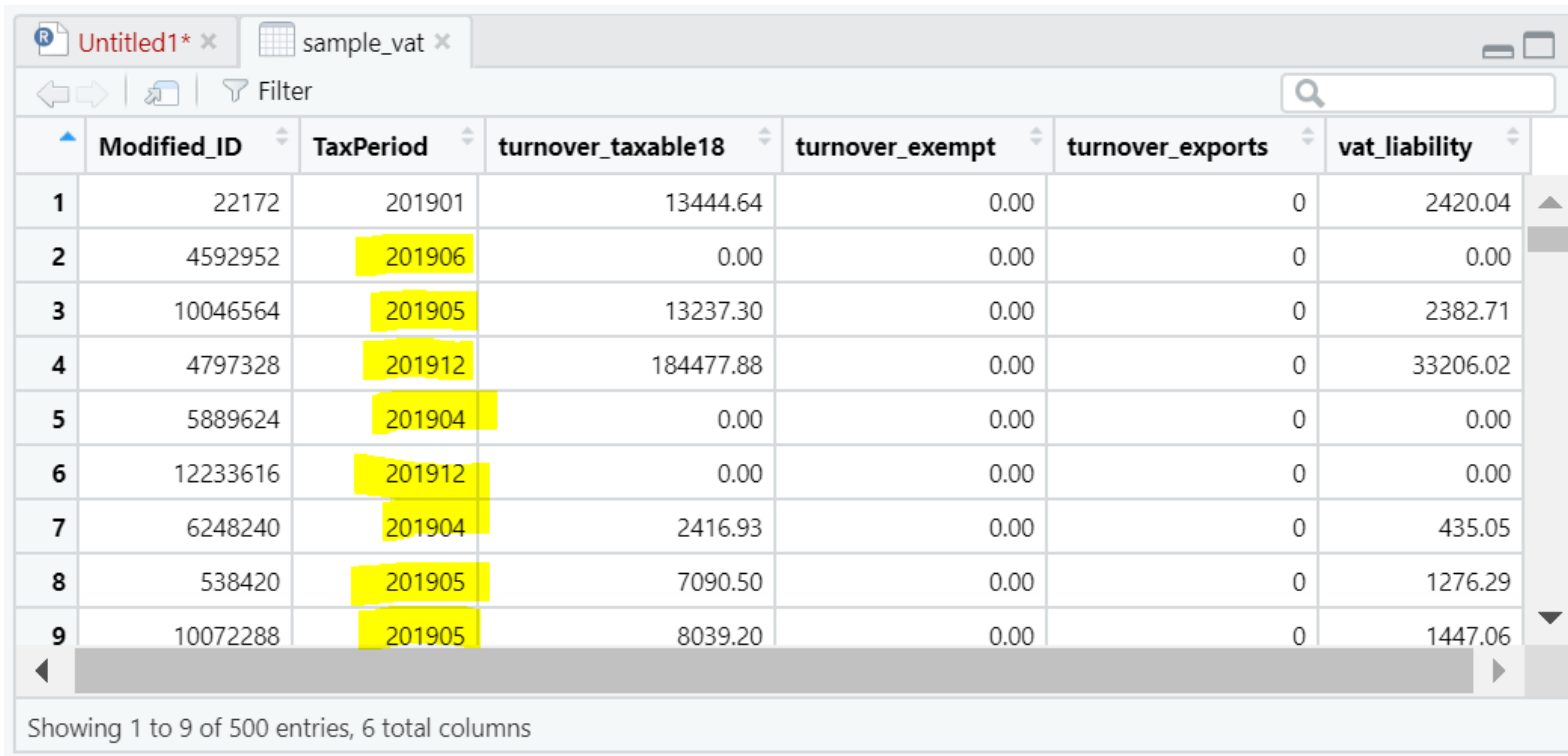
	Modified_ID	TaxPeriod	turnover_taxable18	turnover_exempt	turnover_exports	vat_liability
1	22172	201901	13444.64	0.00	0	2420.04
2	4592952	201906	0.00	0.00	0	0.00
3	10046564	201905	13237.30	0.00	0	2382.71
4	4797328	201912	184477.88	0.00	0	33206.02
5	5889624	201904	0.00	0.00	0	0.00
6	12233616	201912	0.00	0.00	0	0.00
7	6248240	201904	2416.93	0.00	0	435.05
8	538420	201905	7090.50	0.00	0	1276.29
9	10072288	201905	8039.20	0.00	0	1447.06

Exercise 5: Subset the data

1. Use the following code to subset `sample_vat` and leave only the observations in January 2019: `subset(sample_vat, TaxPeriod == 201901)`
 - Note that the "T" and "P" in `TaxPeriod` are **uppercase**
 - Note that there are **two equal signs** in the condition, not one
2. Use `View(sample_vat)` to visualize the dataframe again and see how it changed (note the uppercase "V")

Data in R // მონაცემები R

Does anything look strange?



	Modified_ID	TaxPeriod	turnover_taxable18	turnover_exempt	turnover_exports	vat_liability
1	22172	201901	13444.64	0.00	0	2420.04
2	4592952	201906	0.00	0.00	0	0.00
3	10046564	201905	13237.30	0.00	0	2382.71
4	4797328	201912	184477.88	0.00	0	33206.02
5	5889624	201904	0.00	0.00	0	0.00
6	12233616	201912	0.00	0.00	0	0.00
7	6248240	201904	2416.93	0.00	0	435.05
8	538420	201905	7090.50	0.00	0	1276.29
9	10072288	201905	8039.20	0.00	0	1447.06

Showing 1 to 9 of 500 entries, 6 total columns

Data in R // მონაცემები R

- Indeed, the dataframe `sample_vat` didn't change
- That is because we didn't use the arrow operator (`<-`) to store the result in an object
- Instead, R only printed the result in the console for us (and nothing else)

Data in R // მონაცემები R

The screenshot displays the RStudio interface with several key components highlighted by red boxes and arrows:

- Source Editor:** Contains R code. Line 11, `subset(sample_vat, TaxPeriod == 201901)`, is highlighted. A red box is also around the `Run` button in the toolbar.
- Environment Pane:** Shows the `Global Environment` with a data object `sample_vat` (500 obs. of 6 variables). The `Values` table below it shows:

x1	50
x2	15
x3	65

A red box around this pane contains the text "No changes here".
- Console:** Shows the output of the code execution, including a large table of values. A red box highlights the bottom portion of this output.
- Files Pane:** Shows a list of files in the current project directory, including `.Rhistory`, `00_clean_tribunals.Rmd`, `AD_final_2021_dimereview.txt`, `Analysis`, `Architecture document - impact-projects-db_le...`, `cache`, `Capture.PNG`, `categories.zip`, and `chile_judiciary_web_scraping_progress_Dec16.xlsx`.

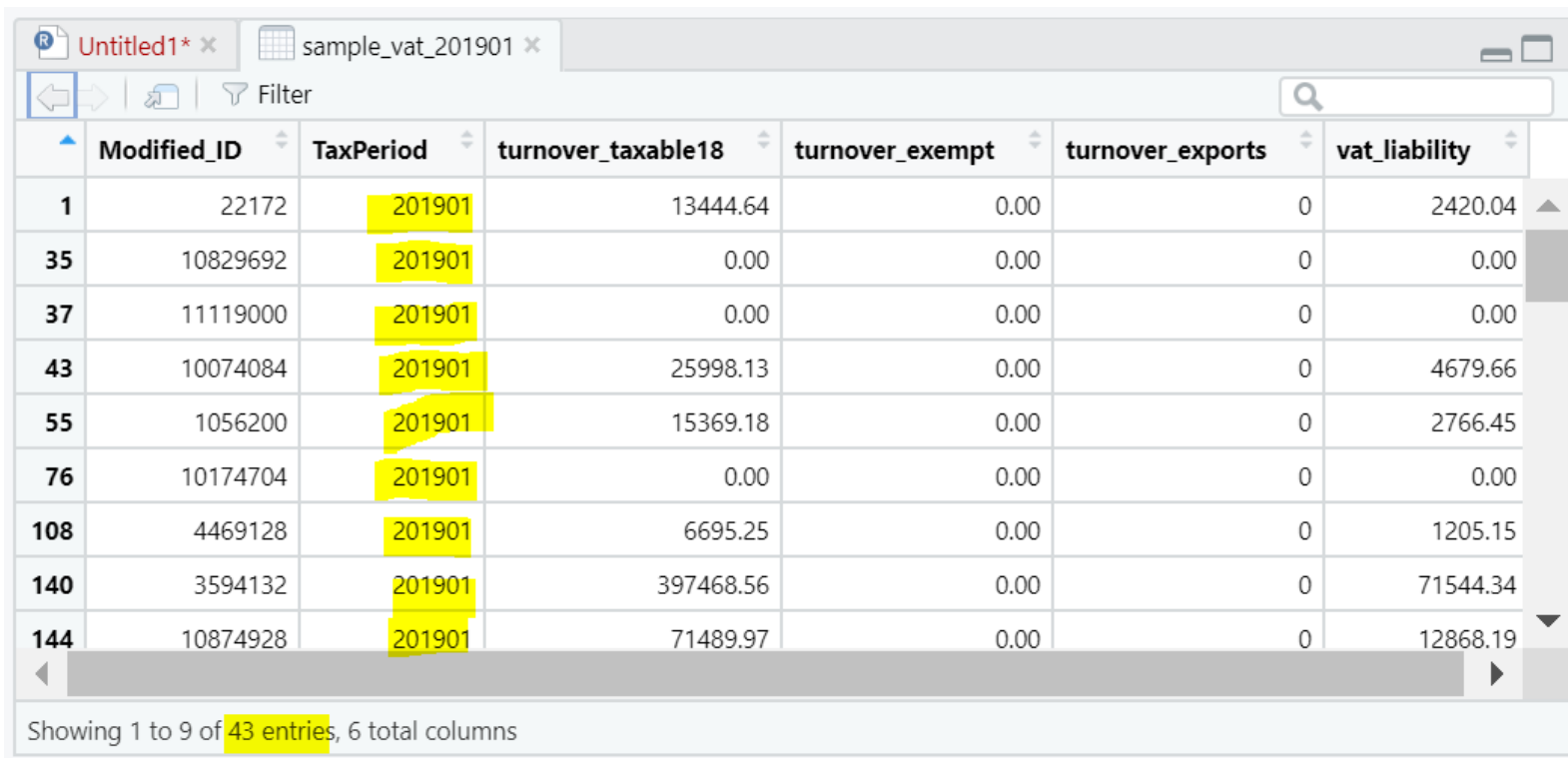
Red arrows indicate the flow of execution: one arrow points from the `Run` button to the console output, and another points from the highlighted code line to the console output.

Exercise 6: Subsetting and storing the result

1. Subset the dataframe again and save the result into a new dataframe called `sample_vat_201901`. You can use this code for that: `sample_vat_201901 <- subset(sample_vat, TaxPeriod == 201901)`
2. Use `View(sample_vat_201901)` to visualize the result (notice the uppercase "V")

Data in R // მონაცემები R

Now the resulting dataframe looks correct!



Untitled1* x sample_vat_201901 x

Filter

	Modified_ID	TaxPeriod	turnover_taxable18	turnover_exempt	turnover_exports	vat_liability
1	22172	201901	13444.64	0.00	0	2420.04
35	10829692	201901	0.00	0.00	0	0.00
37	11119000	201901	0.00	0.00	0	0.00
43	10074084	201901	25998.13	0.00	0	4679.66
55	1056200	201901	15369.18	0.00	0	2766.45
76	10174704	201901	0.00	0.00	0	0.00
108	4469128	201901	6695.25	0.00	0	1205.15
140	3594132	201901	397468.56	0.00	0	71544.34
144	10874928	201901	71489.97	0.00	0	12868.19

Showing 1 to 9 of 43 entries, 6 total columns

Data in R // მონაცემები R

Note that this time R didn't print the resulting dataframe in the console, it only showed the code we were running. Also, now the new dataframe appears in the environment.

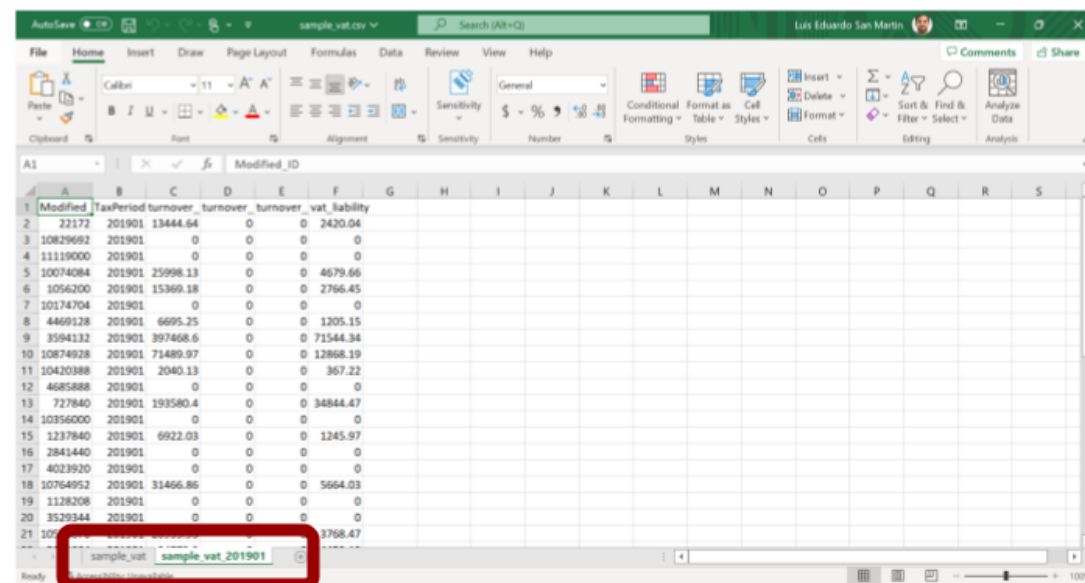
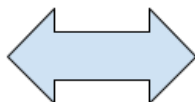
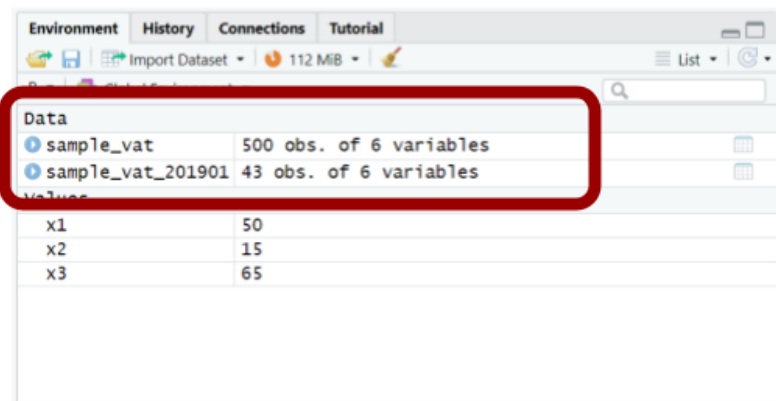
The screenshot illustrates the RStudio interface during the execution of R code. Red boxes and arrows highlight the workflow:

- Code Editor:** Lines 11 and 12 are highlighted: `subset(sample_vat, TaxPeriod == 201901)` and `sample_vat_201901 <- subset(sample_vat, TaxPeriod == 201901)`.
- Run Button:** The 'Run' button in the top toolbar is highlighted, with an arrow pointing from the code to it.
- Environment Pane:** The 'Data' section shows the creation of a new object: `sample_vat_201901` (43 obs. of 6 variables).
- Console:** The command `> sample_vat_201901 <- subset(sample_vat, TaxPeriod == 201901)` is shown, with an arrow pointing from the code to the console output.

The bottom pane shows a file explorer with various files and folders, including `.Rhistory`, `00_clean_tribunals.Rmd`, `AD_final_2021_dimereview.txt`, `Analysis`, `Architecture document - impact-projects-db_le...`, `cache`, `Capture.PNG`, `categories.zip`, and `chile_judiciary_web_scraping_progress_Dec16.xlsx`.

Data in R // მონაცემები R

- R can store multiple dataframes in the environment. This is analogous to having different spreadsheets in the same Excel window
- Always that dataframes are just objects in R. R differentiates which dataframe the code refers to with the dataframe name



Functions in R // ფუნქციები R

Functions in R // ფუნქციები R

- Functions are how we apply operations to objects in R
- We have used a few functions in the previous exercises. For example, `subset()` and `View()` are functions
- Everything that has a name plus parentheses is a function in R

```
subset(sample_vat, TaxPeriod == 201901)
```

Functions in R // ფუნქციები R

Functions have the following syntax:

`subset(sample_vat, TaxPeriod == 201901)`

function name arguments

- **Function name:** the name we use to call a function. It goes before the parentheses
- **Arguments:** inputs and specifications for the function to be applied.
 - Arguments go inside the parentheses
 - The first argument is the object you apply the function on

Functions in R // ფუნქციები R

- The results of a function can always be stored in an object with the arrow operator (`<-`)

```
sample_vat_201901 <- subset(sample_vat, TaxPeriod == 201901)
```

- As we saw in exercise 5, the results of a function will only be printed in the console if you don't store them

Functions in R // ფუნქციები R

Exercise 7: Using the function `summary()`

1. Compute the summary statistics of the variables of `sample_vat` and save the result with `summary_table <- summary(sample_vat)`
2. Print the stored result with `print(summary_table)`

Functions in R // ფუნქციები R

Note that this code is both creating a new object (with `summary_table <- summary(sample_vat)`) and printing the result in the console (with `print(summary_table)`)

The screenshot shows the RStudio interface with the following components:

- Source Editor:** Contains R code. The line `summary_table <- summary(sample_vat)` is highlighted in blue. A red box highlights the `Run` button (a green arrow icon).
- Environment Pane:** Shows objects in the global environment. `sample_vat` is a data frame with 6 variables. `sample_vat_201901` has 43 observations. A red box highlights the `summary_table` object, which is a 'table' of character type with dimensions [1:6, 1:6].
- Console:** Displays the output of the code. A red box highlights the console output, which shows a summary of the `sample_vat` data frame. A red box also highlights the `print(summary_table)` command in the source editor.
- Object Summaries Pane:** Shows the documentation for the `summary` function, including its description and usage.

Red arrows indicate the flow of execution: from the `Run` button to the `summary_table` object in the Environment pane, and from the `print(summary_table)` command to the console output.

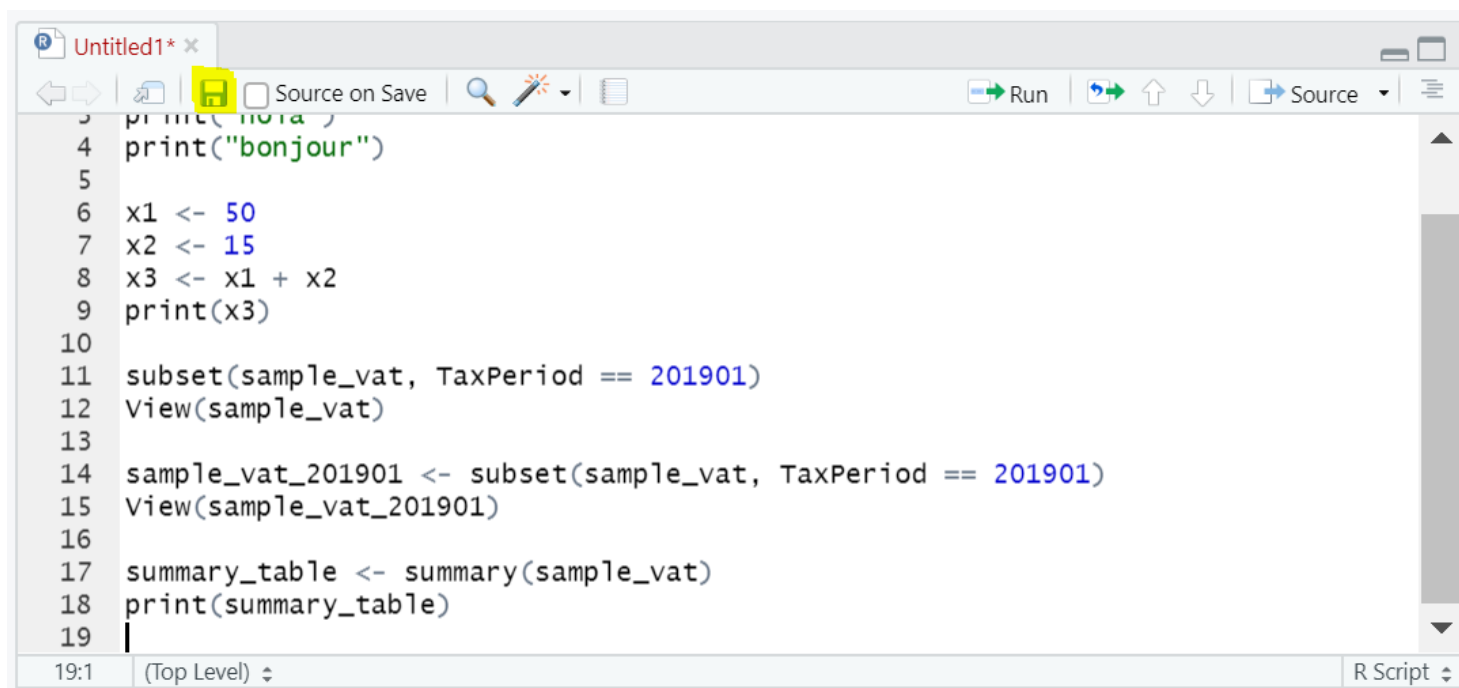
Questions? // კითხვები?

Wrapping up // შეფუთვა

Wrapping up // შეფუთვა

Always save your work!

- Click the floppy disk icon to save your work
- Select a location for your file and remember where you're saving it

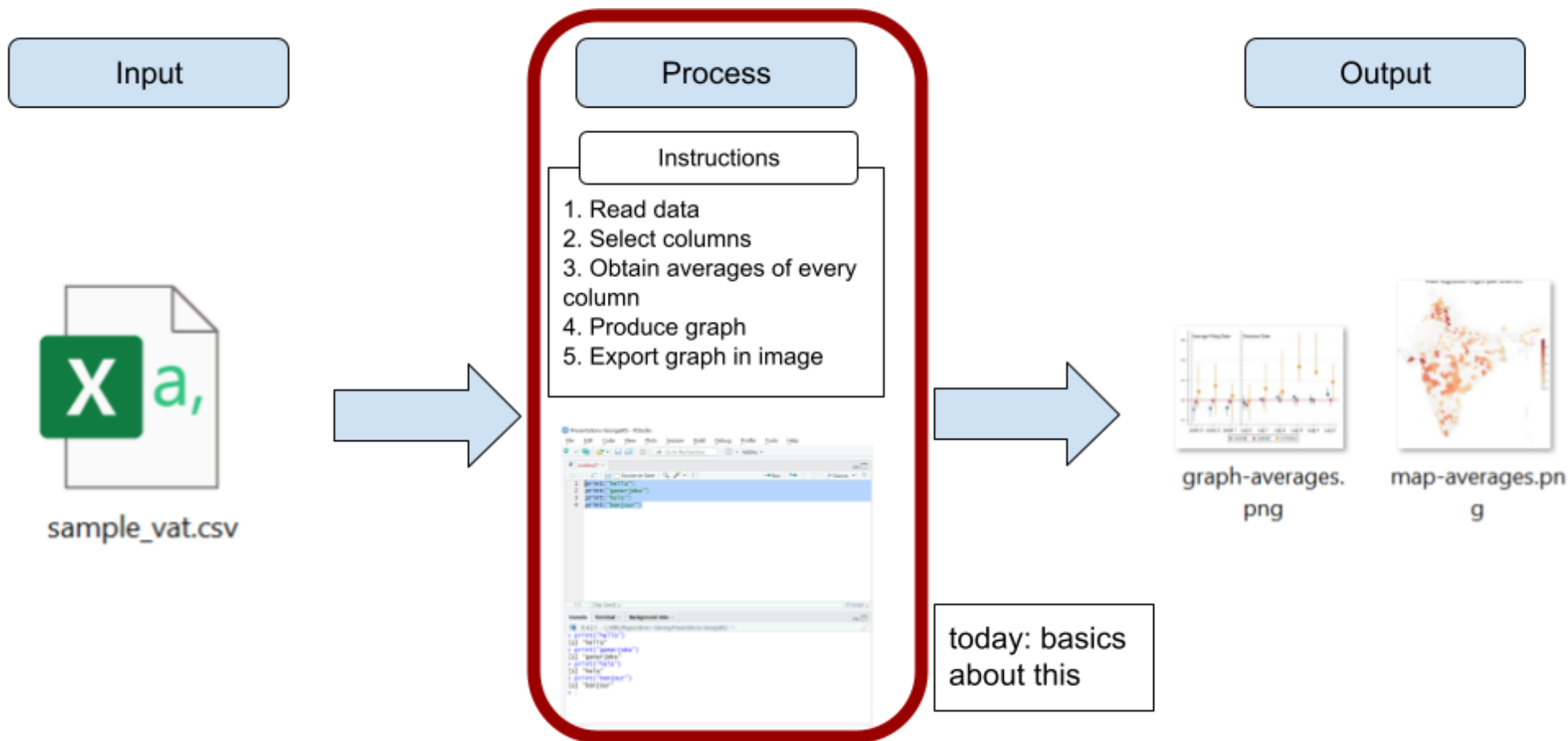


```
1 print("hello")
2
3 print("bonjour")
4
5
6 x1 <- 50
7 x2 <- 15
8 x3 <- x1 + x2
9 print(x3)
10
11 subset(sample_vat, TaxPeriod == 201901)
12 View(sample_vat)
13
14 sample_vat_201901 <- subset(sample_vat, TaxPeriod == 201901)
15 View(sample_vat_201901)
16
17 summary_table <- summary(sample_vat)
18 print(summary_table)
19
```

Wrapping up // შეჯუთვა

Today

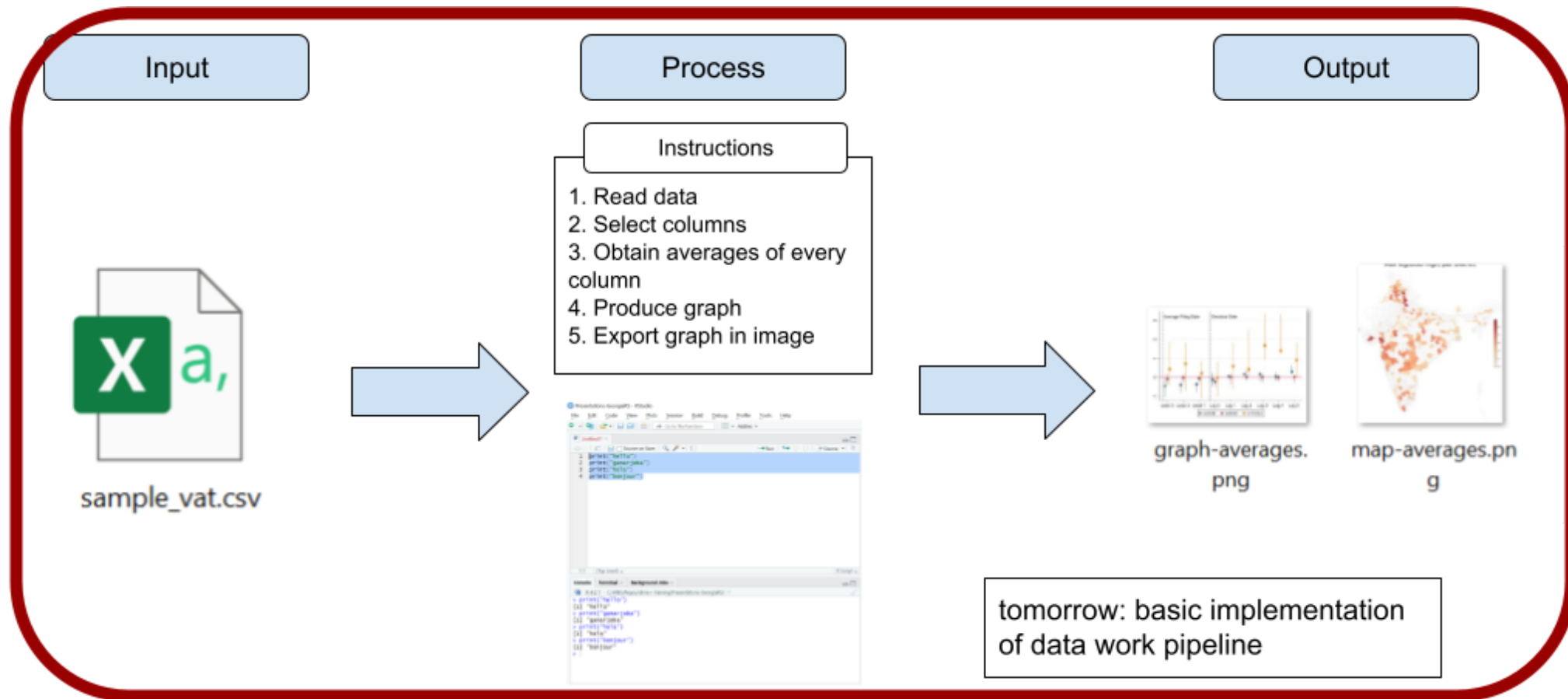
Today we focused on the basics for writing R code



Wrapping up // შეჯუთვა

Tomorrow

Tomorrow we'll review a few simple examples of the entire data work pipeline



Thanks! // მადლობა! // ¡Gracias!
