

Session 1 - Introduction to R

R training - Georgia RS-WB DIME

Luis Eduardo San Martin

The World Bank | September 2023



Table of contents // სარჩევი

1. Introduction
2. Data work and statistical programming
3. Statistical programming
4. Writing R code
5. Data in R
6. Object types
7. Functions in R
8. Wrapping up
9. Appendix

Introduction // შესავალი

About this training

- This is an **introduction** to data work and statistical programming in R
- The training does not require any background in statistical programming
- A computer with R and RStudio installed is required to complete the exercises
- Internet connection is required to download training materials

Learning objectives

By the end of the training, you will know:

- How to approach data work through statistical programming using R and RStudio
- How to apply data wrangling and create outputs with descriptive statistics and data visualization by developing and running reproducible R code

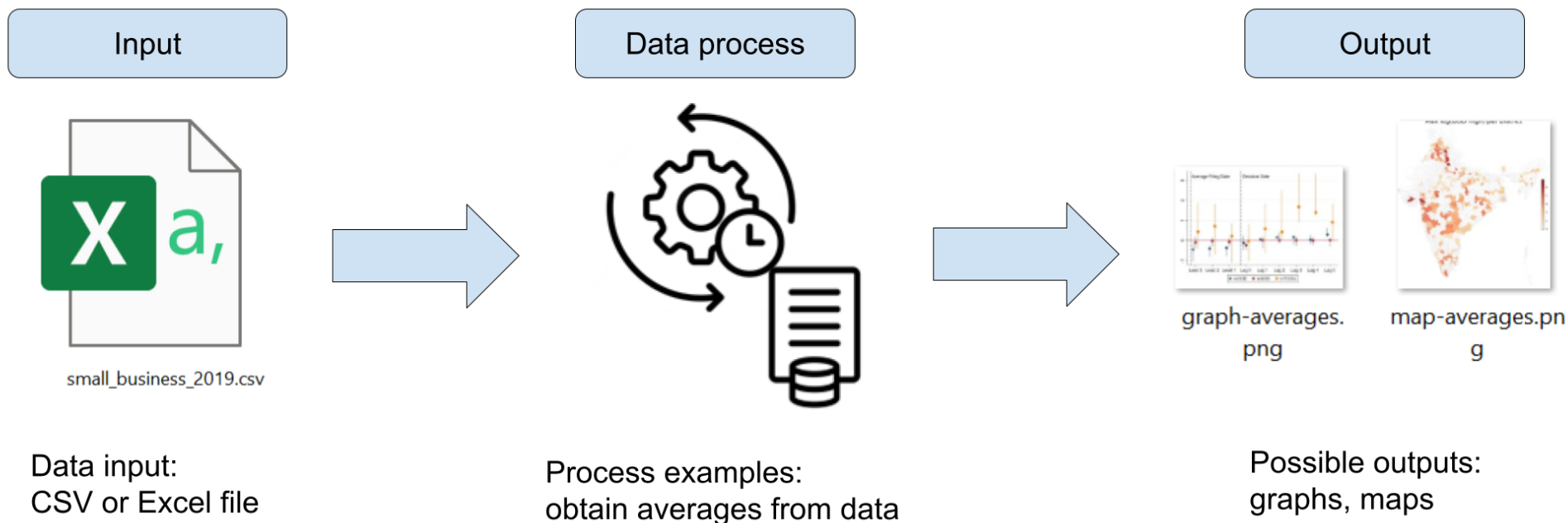
Data work and statistical programming //

სტატისტიკური პროგრამირება

Data work // მონაცემთა მუშაობა

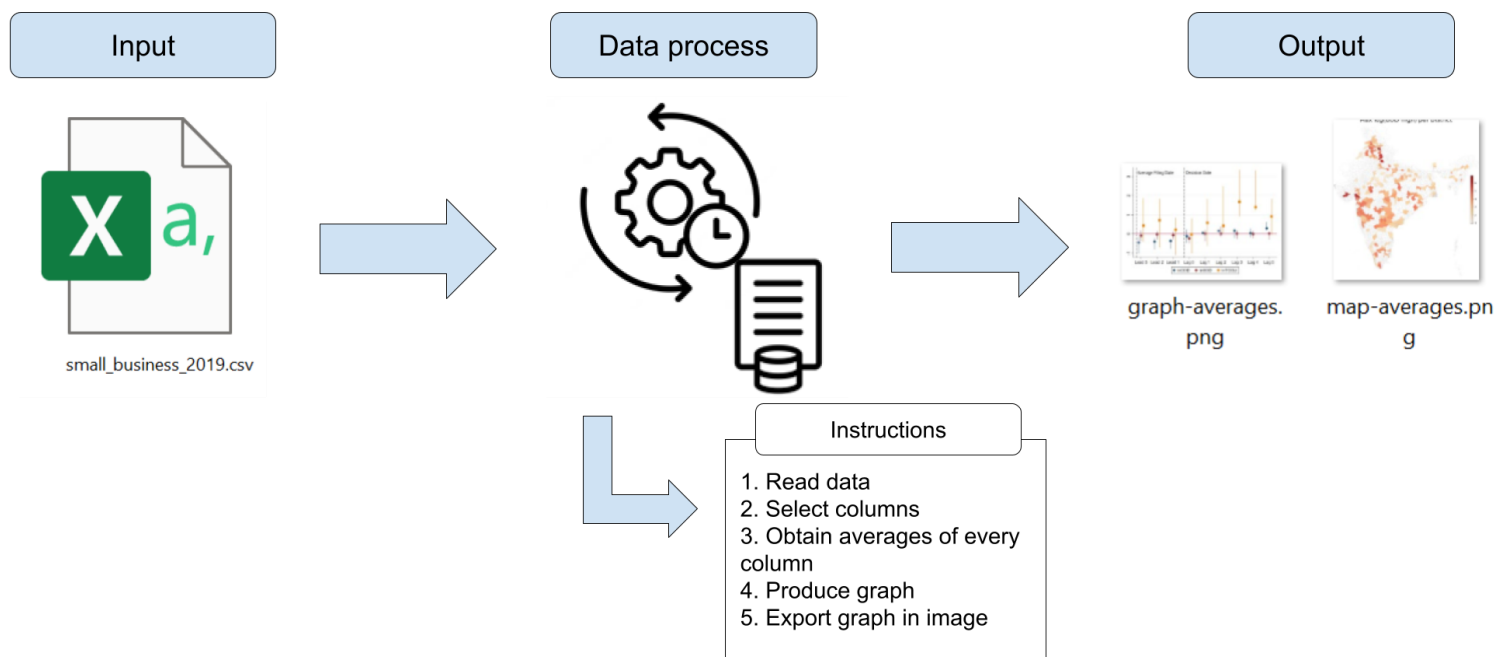
For the context of this training, we'll call data work everything that:

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



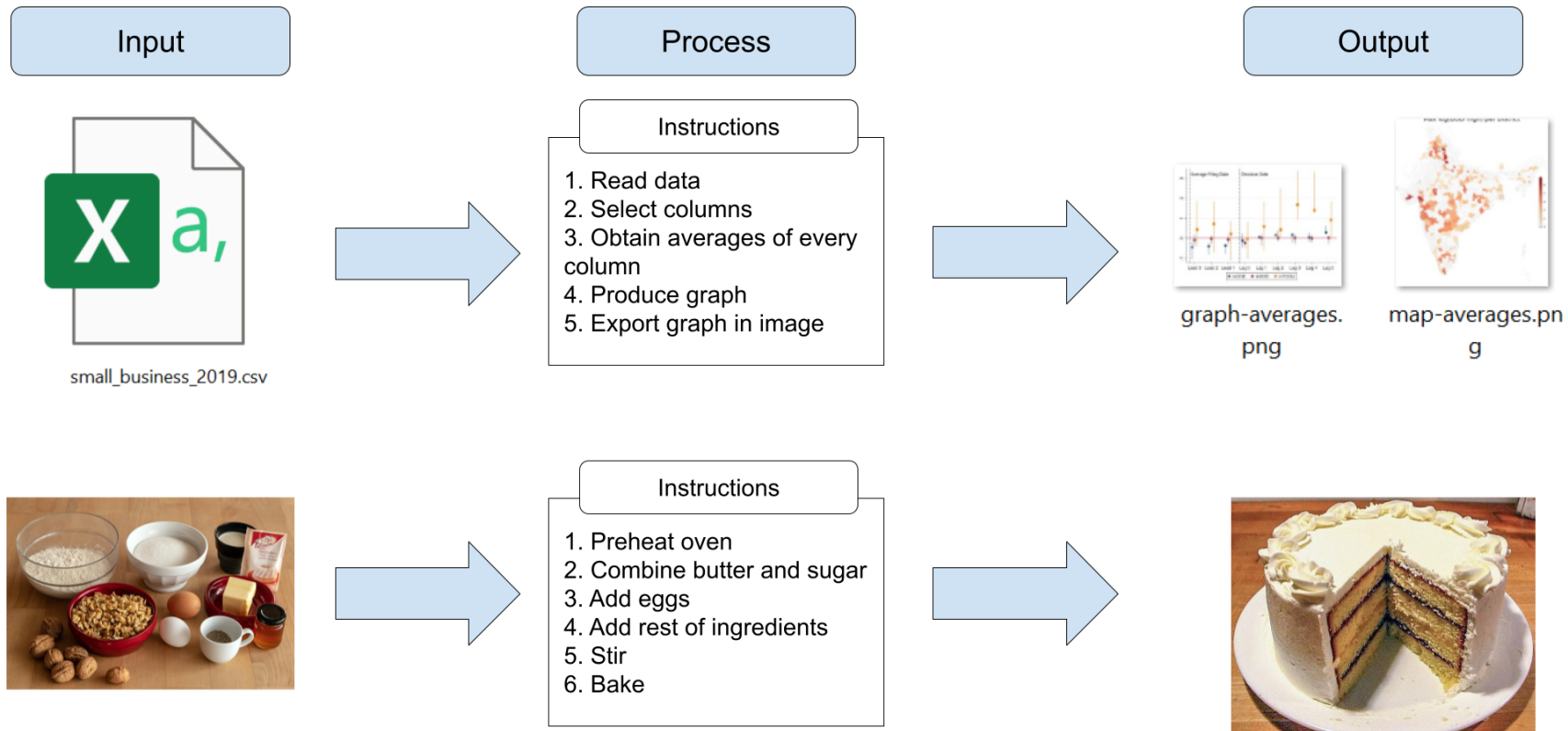
Statistical programming // პროგრამირება

- 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 // პროგრამირება

- You can think of statistical programming as writing a recipe



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

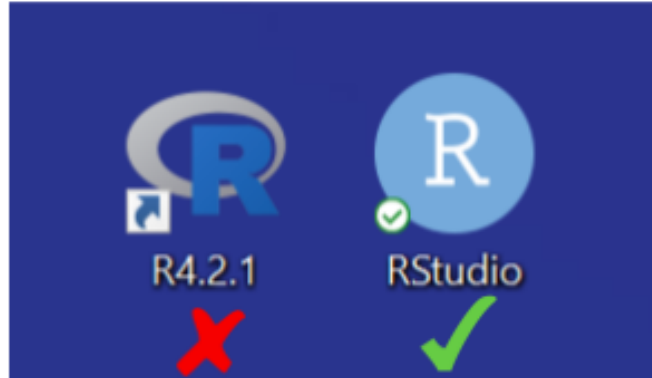


How to write R code?

- 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

How to write R code?

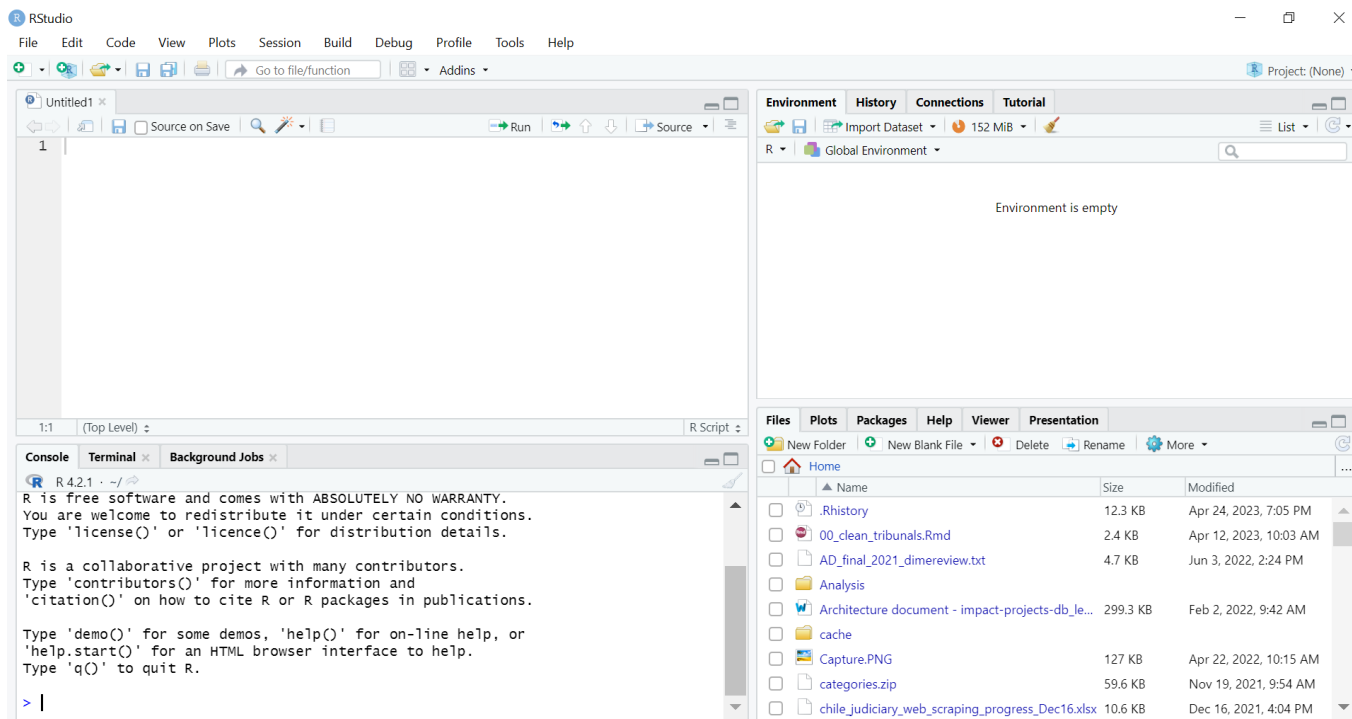
- Now open RStudio in your computer
- Please make sure you're opening RStudio and not R



Statistical programming // პროგრამირება

How to write R code?

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

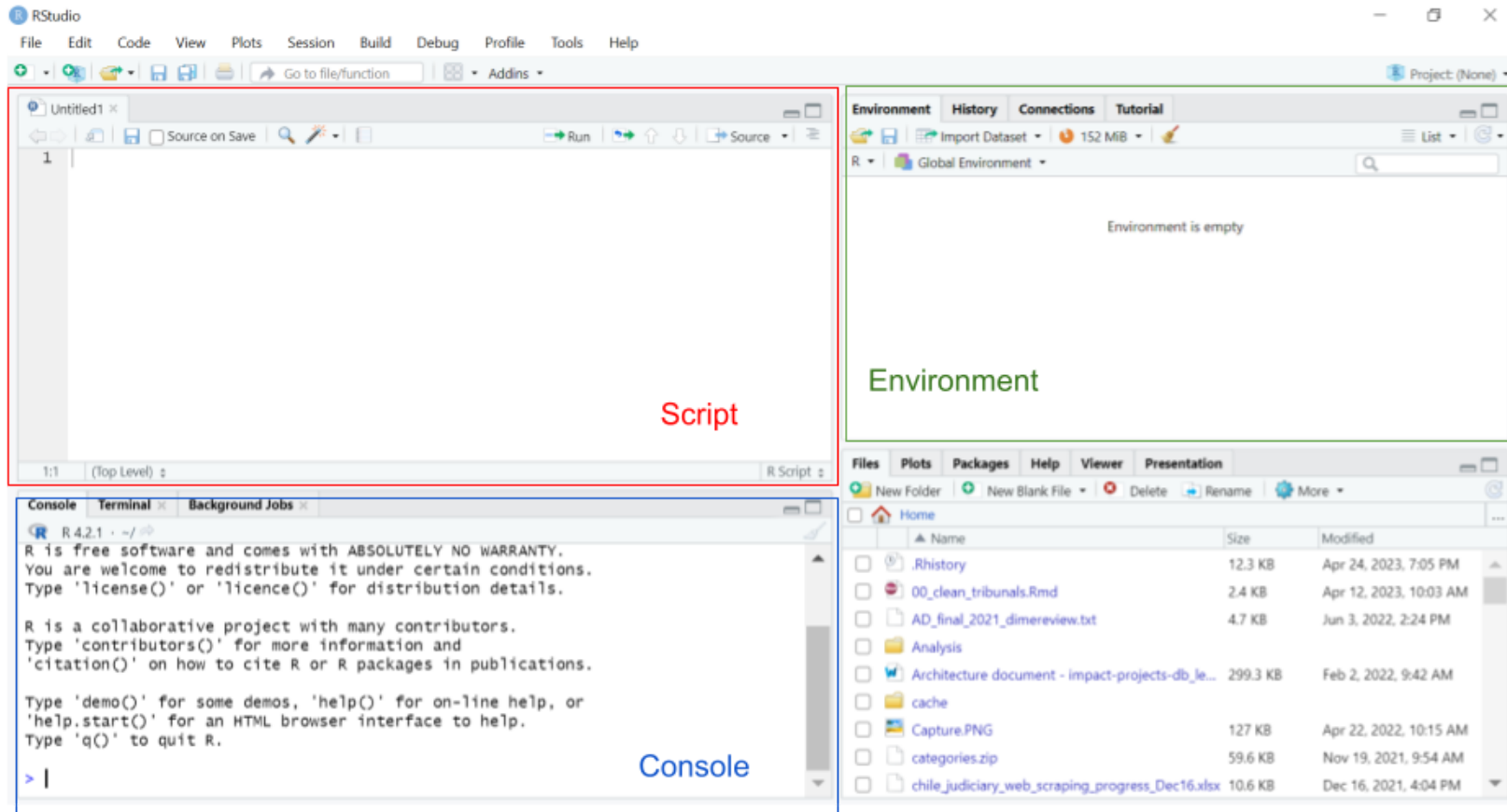


Questions? // კითხვები?

Writing R code // R კოდის დაწერა

Writing R code // R კოდის დაწერა

RStudio interface



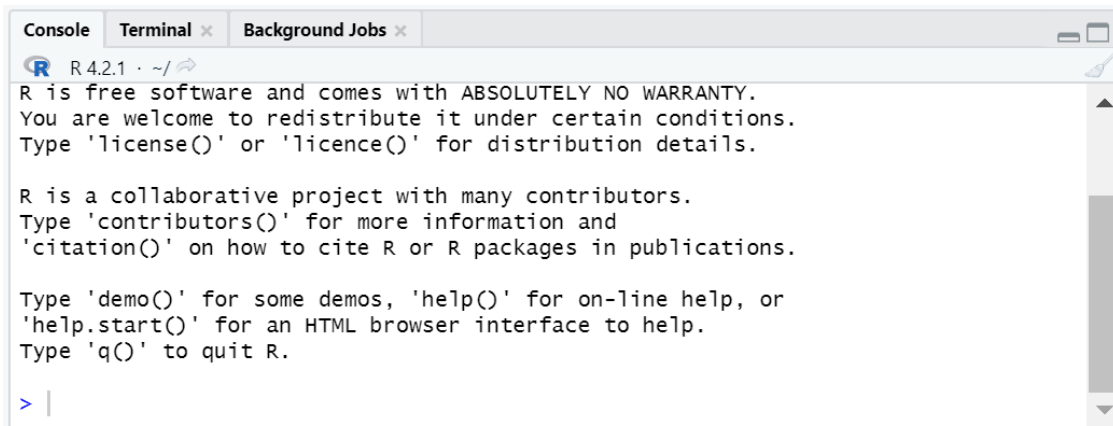
Writing R code // R კოდის დაწერა

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



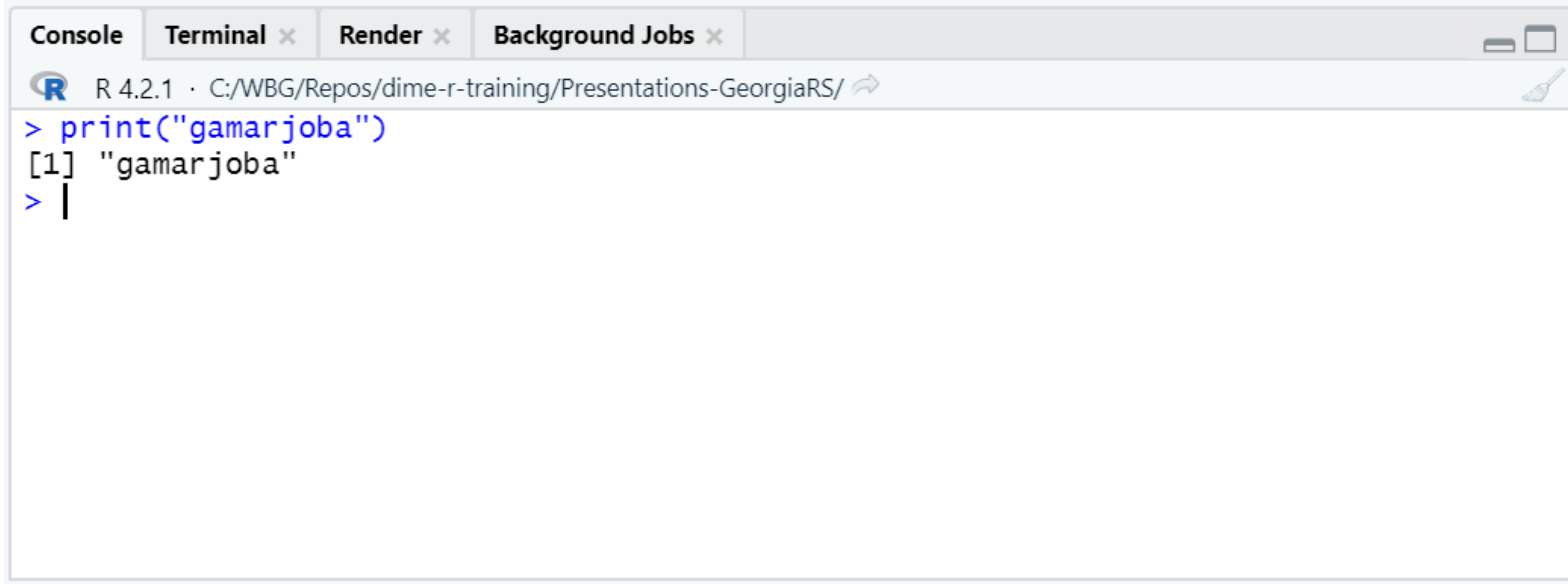
```
R 4.2.1 · ~/
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

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

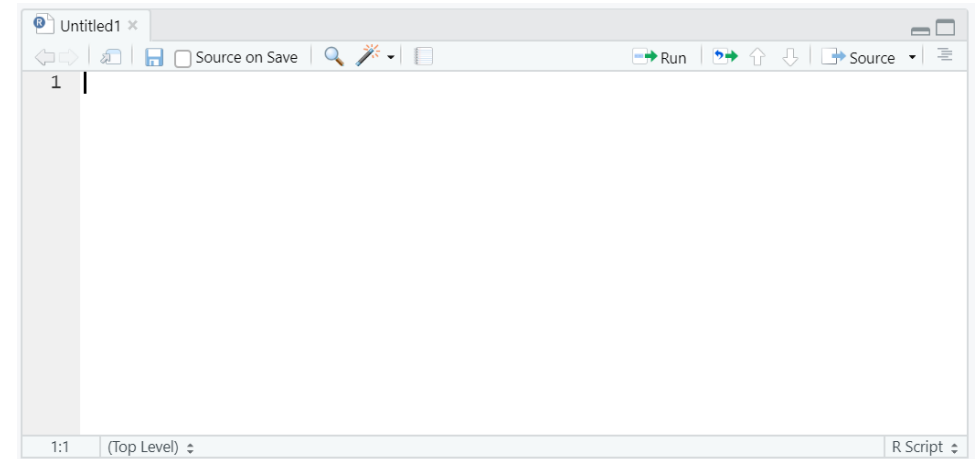
Exercise 2: writing a short script

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

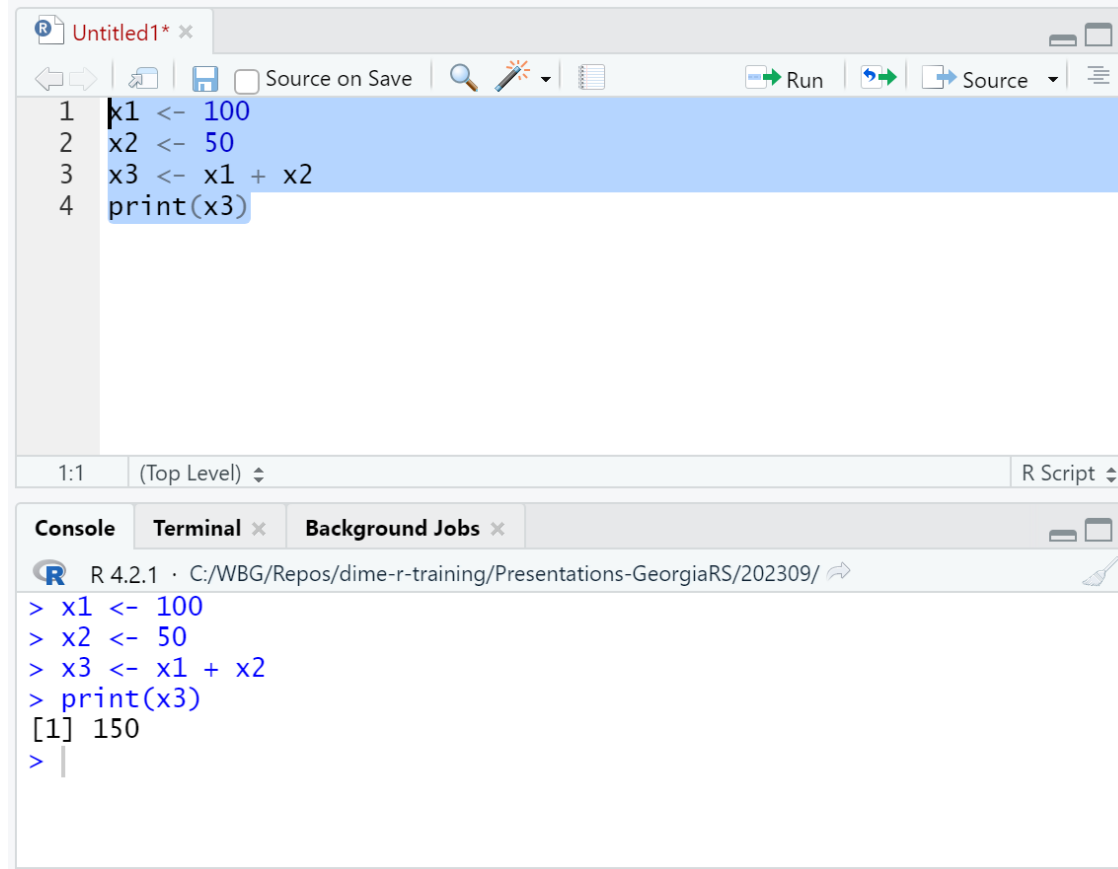
```
x1 <- 100  
x2 <- 50  
x3 <- x1 + x2  
print(x3)
```

2- Select the text you introduced with your mouse

3- Press "Run"



Writing R code // R კოდის დაწერა



The image shows a screenshot of an R script editor window titled "Untitled1*" and a console window below it. The script editor contains four lines of R code: `x1 <- 100`, `x2 <- 50`, `x3 <- x1 + x2`, and `print(x3)`. The console window shows the output of running this code: `> x1 <- 100`, `> x2 <- 50`, `> x3 <- x1 + x2`, `> print(x3)`, and the result `[1] 150`.

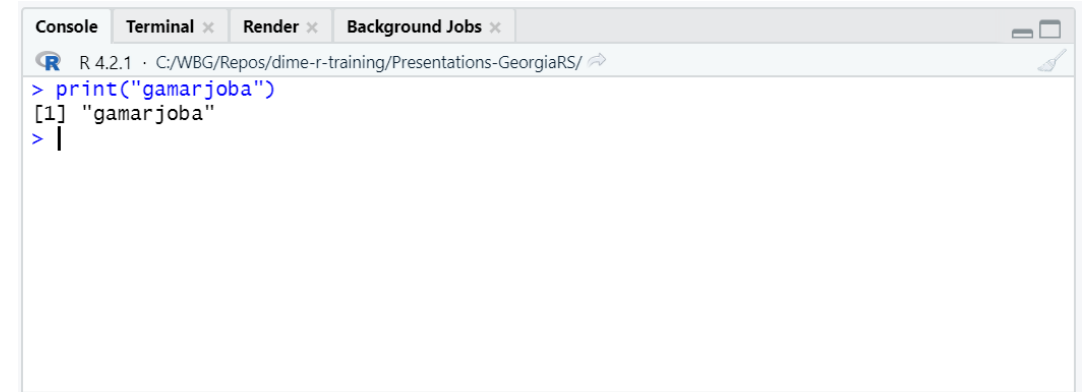
```
1 x1 <- 100
2 x2 <- 50
3 x3 <- x1 + x2
4 print(x3)
```

```
> x1 <- 100
> x2 <- 50
> x3 <- x1 + x2
> print(x3)
[1] 150
>
```

Writing R code // R კოდის დაწერა

R scripts

- Writing and running code from the console will execute it immediately



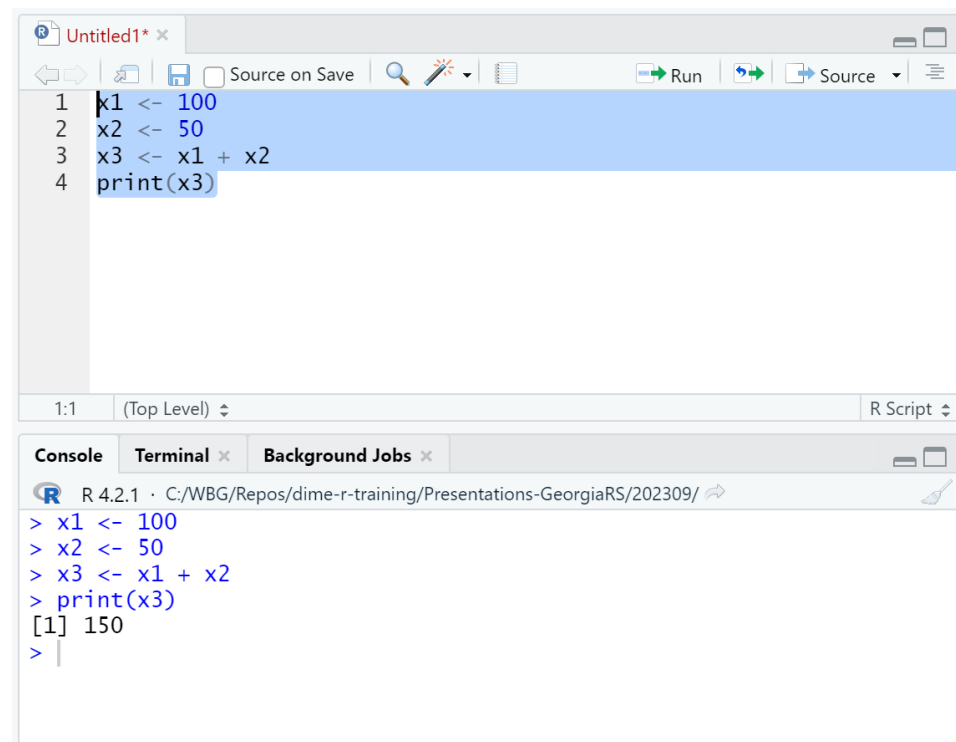
The screenshot shows the R console window with the following content:

```
R 4.2.1 · C:/WBG/Repos/dime-r-training/Presentations-GeorgiaRS/
> print("gamarjoba")
[1] "gamarjoba"
> |
```

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 pane, titled 'Untitled1*', contains a script with four lines of R code: `x1 <- 100`, `x2 <- 50`, `x3 <- x1 + x2`, and `print(x3)`. The bottom pane is divided into 'Console', 'Terminal', and 'Background Jobs' tabs. The 'Console' tab is active, showing the execution of the script: `> x1 <- 100`, `> x2 <- 50`, `> x3 <- x1 + x2`, `> print(x3)`, followed by the output `[1] 150`. The status bar at the bottom indicates 'R 4.2.1' and the current file path.

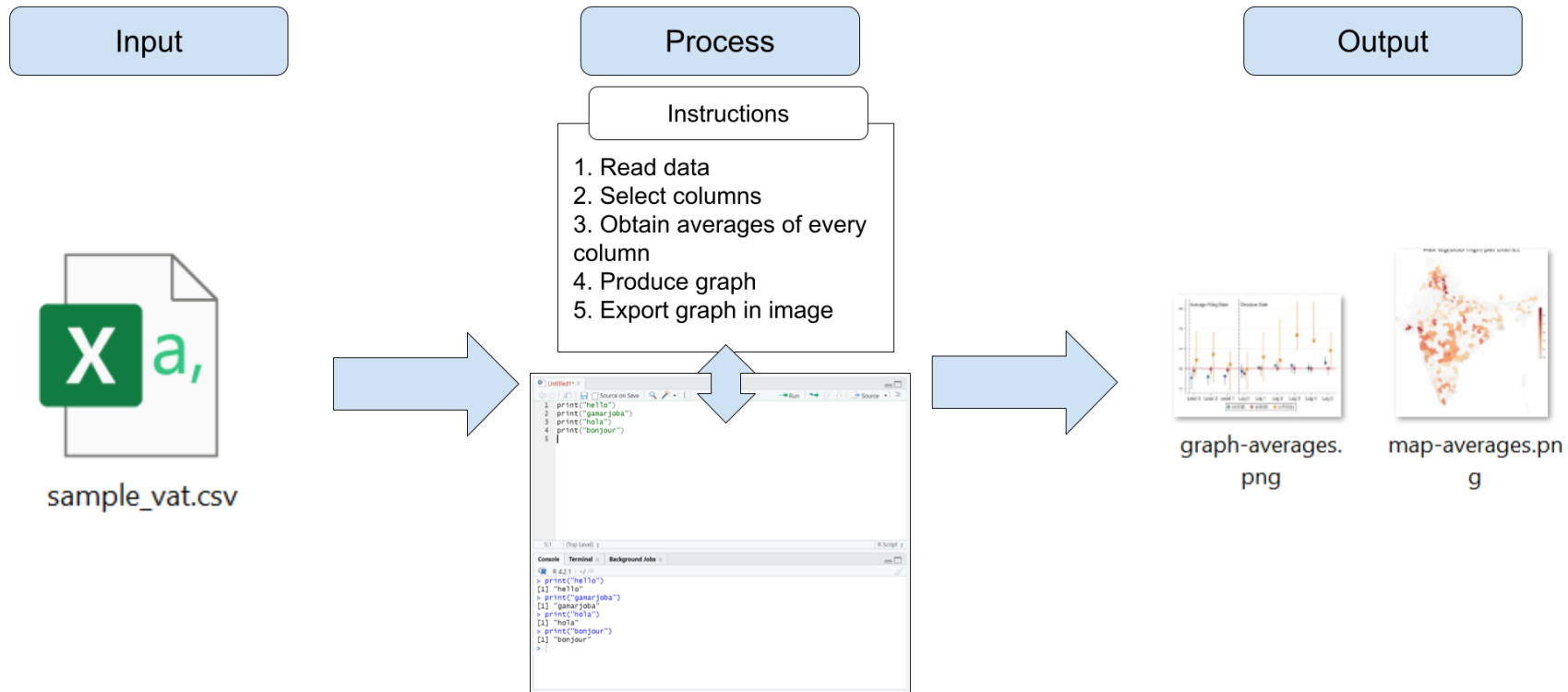
```
1 x1 <- 100
2 x2 <- 50
3 x3 <- x1 + x2
4 print(x3)
```

```
> x1 <- 100
> x2 <- 50
> x3 <- x1 + x2
> print(x3)
[1] 150
>
```

Writing R code // R კოდის დაწერა

R scripts

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



Writing R code // R კოდის დაწერა

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 (`<-`)
- In exercise 2, we created objects each time we used `<-`
- After an object is created, we can refer to it using its name:

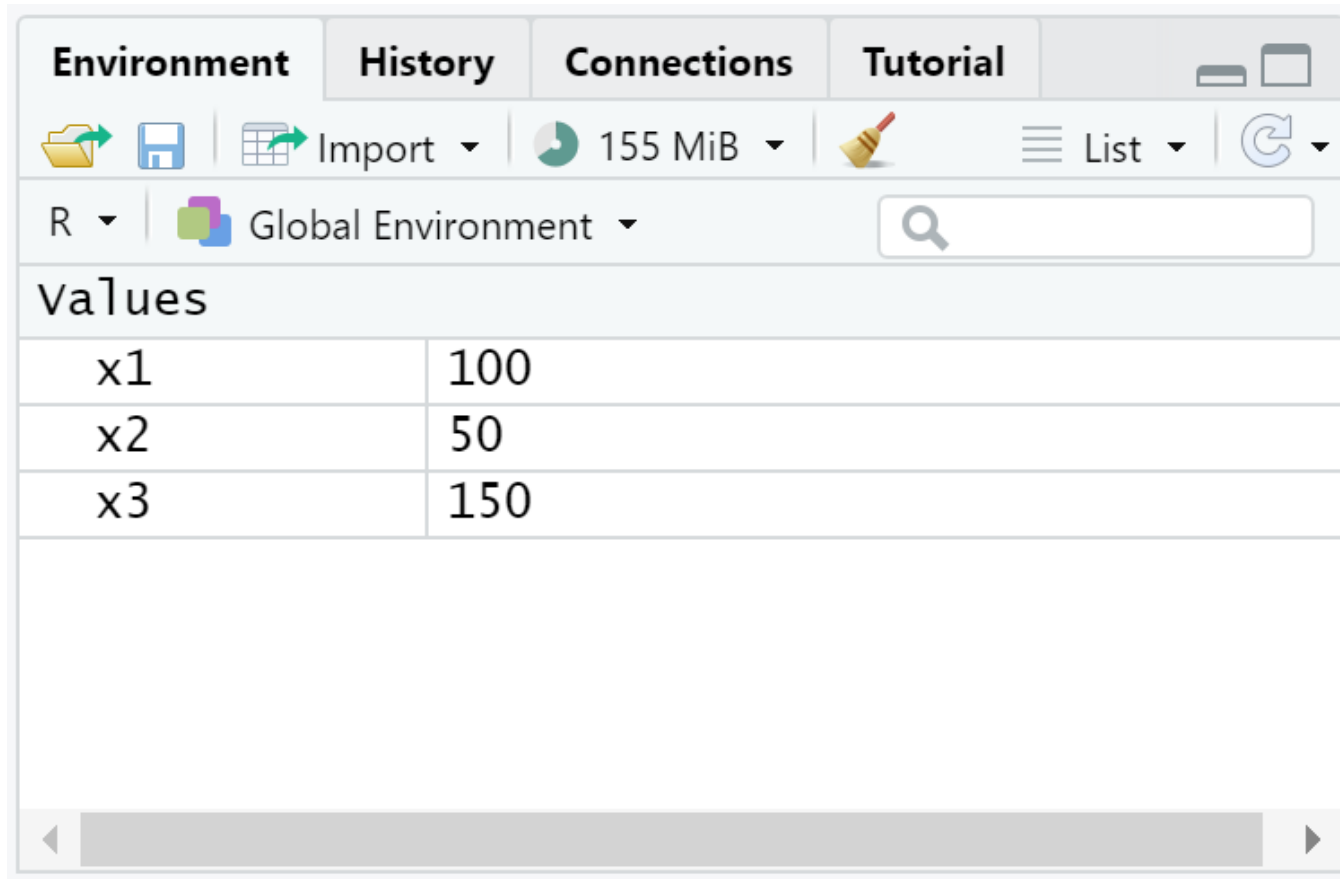
```
print(x3 + 8)
```

```
## [1] 158
```


Writing R code // R კოდის დაწერა

Creating objects in R

- After any objects are created, they will show in the environment panel



Writing R code // R კოდის დაწერა

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

Data in R // მონაცემები R

Data in R // მონაცემები R

Exercise 3: Loading data into R

1.- Go to this page: <https://osf.io/2apht> and download the file `small_business_2019.csv`

The screenshot shows the OSF Home interface. At the top, there is a navigation bar with the OSF Home logo, a search bar, and links for My Projects, Search, Support, Donate, and DIME Analytics. Below the navigation bar, the page title is 'R training Georgia RS - WB September 2023'. The main content area displays the file 'small_business_2019.csv'. To the right of the file name, there is a dropdown menu with options: Download, Delete, Embed, and Share. Below the file name, there is a table with the following data:

modified_id	region	income
2933828	KaxeTi	445
11539816	Tbilisi	3610
774836	Guria	2600
10763744	Tbilisi	29
5443012	KaxeTi	95
1303812	Guria	4255.6001
2586640	Guria	2852
679632	Guria	0
10490076	Tbilisi	273
11176036	Samegrelo-Z. SvaneTi	2711

To the right of the table, there is a sidebar with the following sections:

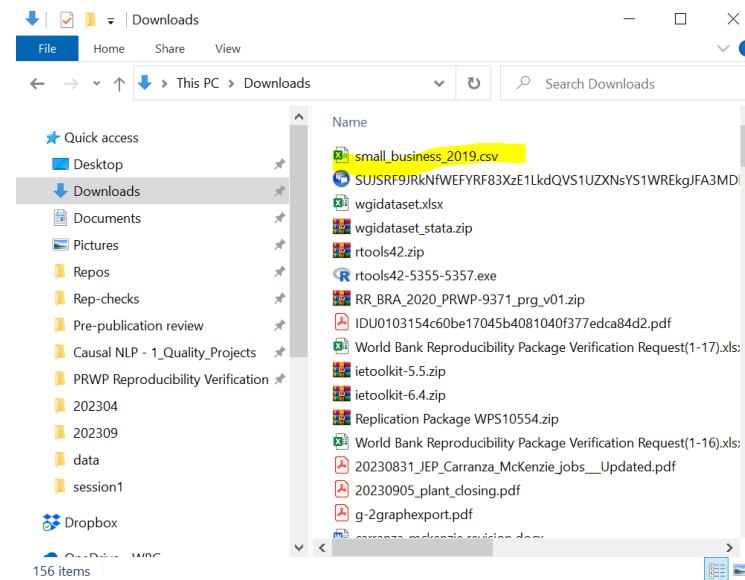
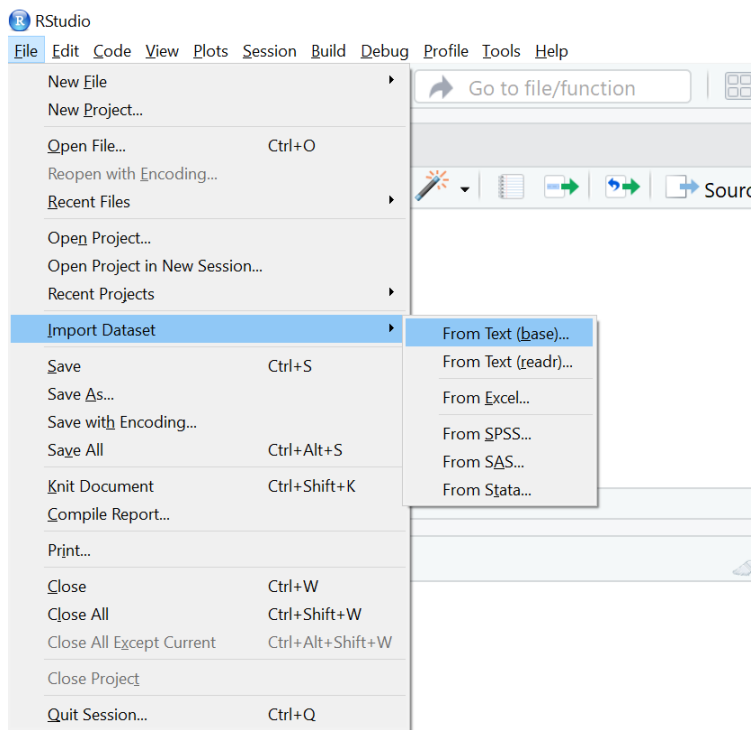
- File Metadata**: Press the edit button to add metadata to this file.
- Project Metadata**:
 - Title**: R training Georgia RS - WB September 2023
 - Date created**: September 19, 2023
 - Date modified**: September 19, 2023
 - Contributors**: DIME Analytics

Data in R // მონაცემები R

Exercise 3: Loading data into R

2.- In RStudio, go to **File** > **Import Dataset** > **From Text (base)** and select the file **small_business_2019.csv**

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



Data in R // მონაცემები R

Exercise 3: Loading data into R

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

4 - Select **Import**

Import Dataset

Name: small_business_2019

Input File

modified_id,region,income
2933828,KaxeTi,445
11539816,Tbilisi,3610
774836,Guria,2600
10763744,Tbilisi,29
5443012,KaxeTi,95
1303812,Guria,4255.6001
2586640,Guria,2852
679632,Guria,0
10490076,Tbilisi,273
11176036,Samegrelo-Z.SvaneTi,2711
244516,Tbilisi,412.44
1431424,Guria,806.09998
1485348,Guria,289.5
562104,Guria,1120
9048544,Tbilisi,47114.398
3735768,KaxeTi,605
6253780,KaxeTi,2967
11783480,Tbilisi,5960

Encoding: Automatic

Heading: ☒ Yes ☐ No

Row names: Automatic

Separator: Comma

Decimal: Period

Quote: Double (")

Comment: None

na.strings: NA

☐ Strings as factors

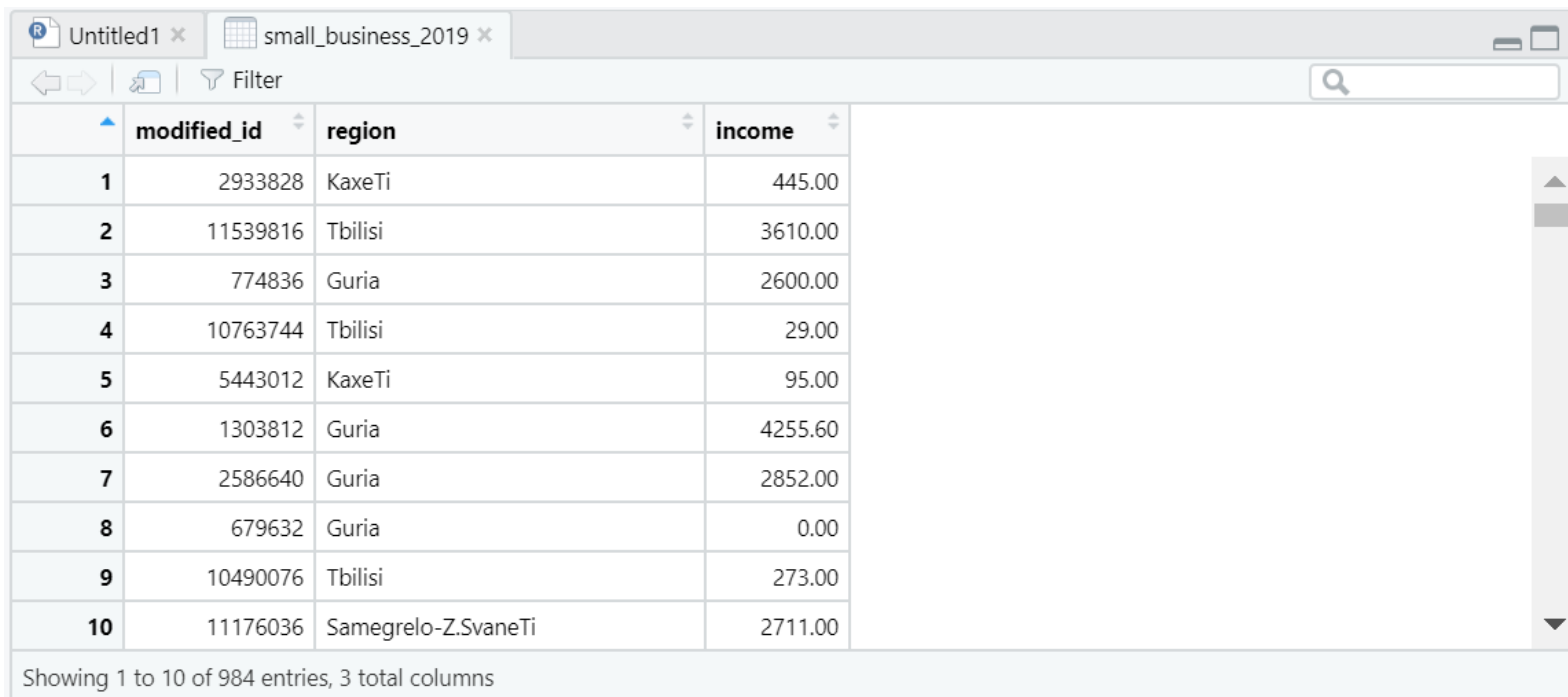
Data Frame

modified_id	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

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 `small_business_2019` to return to the script
- To open the viewer again, use the code: `View(small_business_2019)` (notice the uppercase "V")

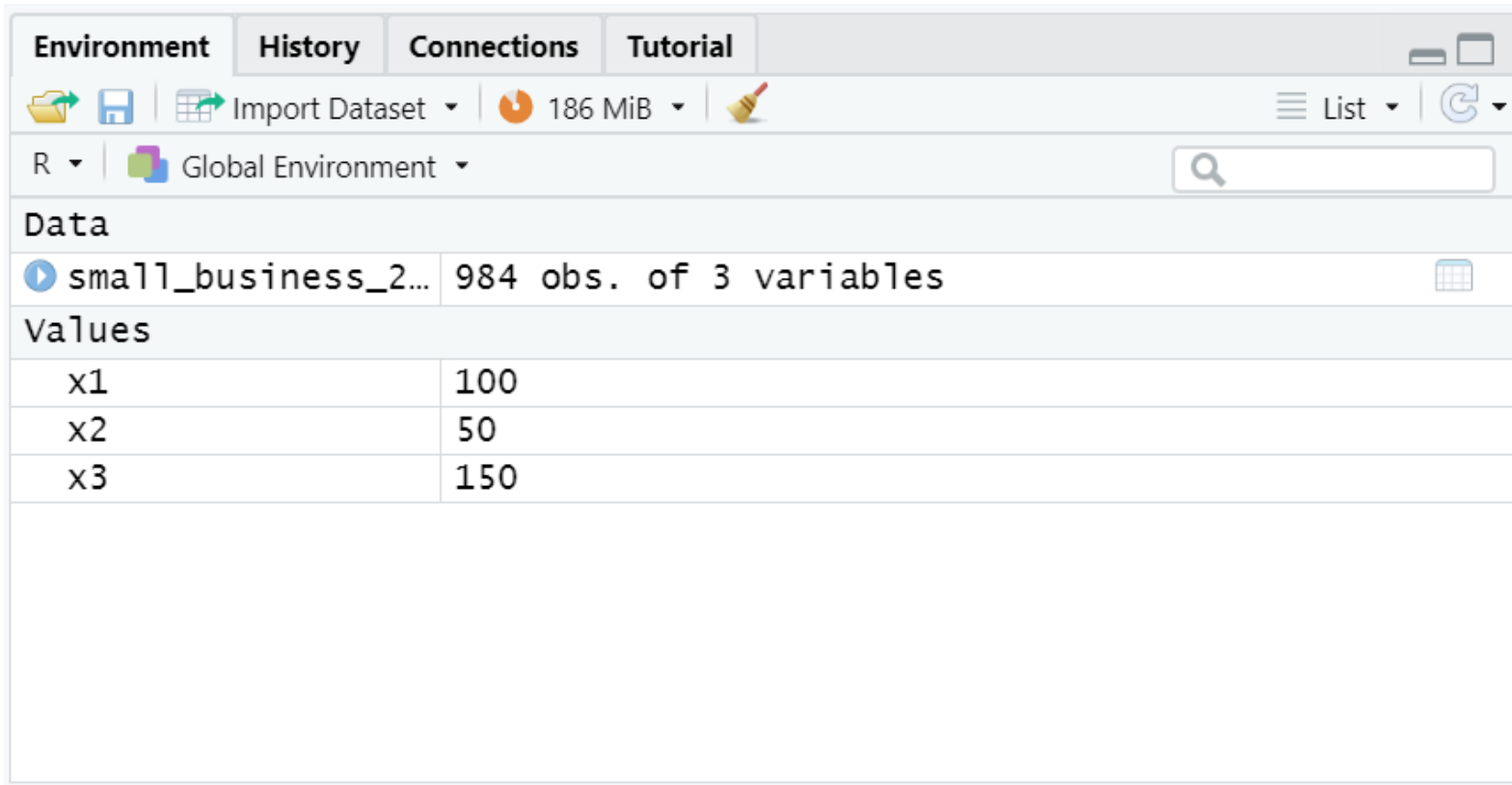


Showing 1 to 10 of 984 entries, 3 total columns

	modified_id	region	income
1	2933828	KaxeTi	445.00
2	11539816	Tbilisi	3610.00
3	774836	Guria	2600.00
4	10763744	Tbilisi	29.00
5	5443012	KaxeTi	95.00
6	1303812	Guria	4255.60
7	2586640	Guria	2852.00
8	679632	Guria	0.00
9	10490076	Tbilisi	273.00
10	11176036	Samegrelo-Z.SvaneTi	2711.00

Data in R // მონაცემები R

- Additionally, you will now see an object named `small_business_2019` in your environment

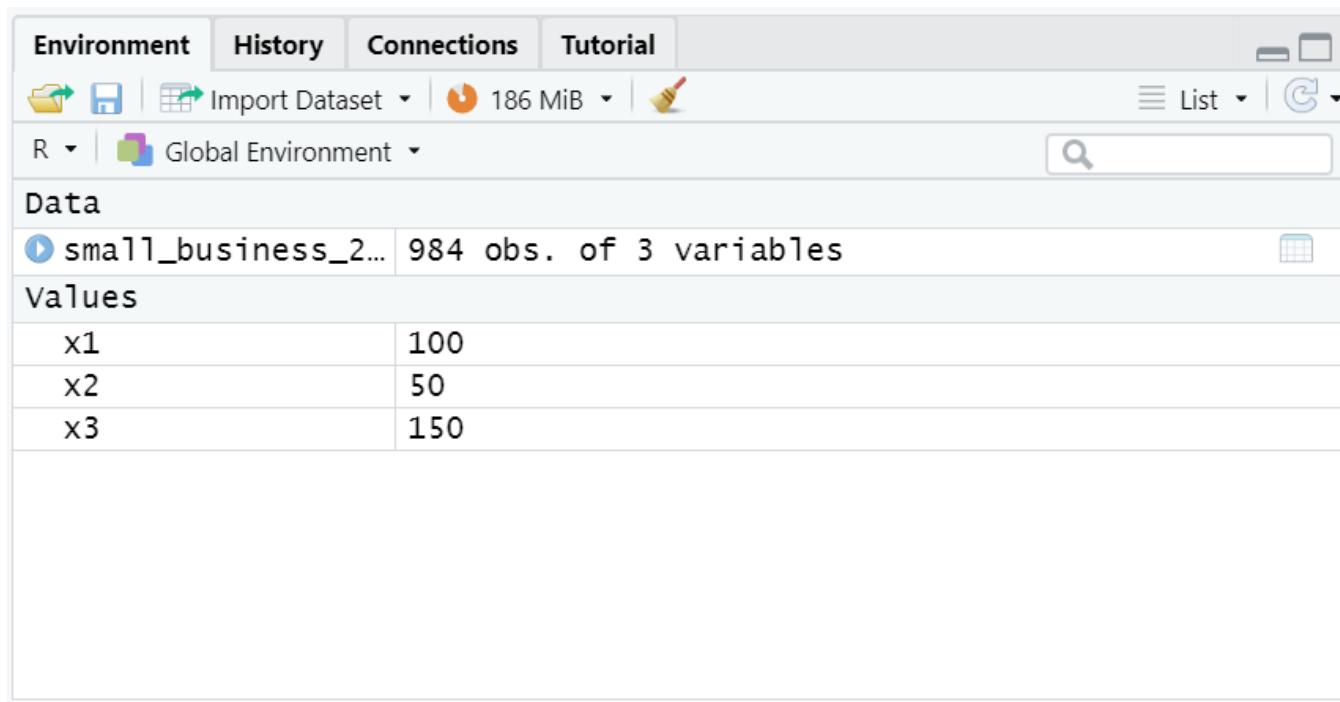


The screenshot shows the RStudio Environment pane. At the top, there are tabs for 'Environment', 'History', 'Connections', and 'Tutorial'. Below the tabs is a toolbar with icons for file operations and a search bar. The 'Environment' tab is active, showing the 'Global Environment'. The object 'small_business_2019' is listed with 984 observations and 3 variables. Below the object name, the values for the three variables (x1, x2, x3) are displayed in a table.

Data	
▶ small_business_2...	984 obs. of 3 variables
Values	
x1	100
x2	50
x3	150

Data in R // მონაცემები R

- Remember we mentioned objects before? For R, `small_business_2019` is an object just like `x1`, `x2`, or `x3`
- The difference is that `small_business_2019` 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 '186 MiB'. The main area of the pane is titled 'Global Environment' and contains a search bar. Below the search bar, the word 'Data' is displayed. Under 'Data', there is a single entry: 'small_business_2...' with a blue play button icon on the left and '984 obs. of 3 variables' on the right. Below this entry, the word 'Values' is displayed. Under 'Values', there is a table with three rows and two columns. The first column contains the variable names 'x1', 'x2', and 'x3'. The second column contains the corresponding values '100', '50', and '150'.

Values	
x1	100
x2	50
x3	150

Data in R // მონაცემები R

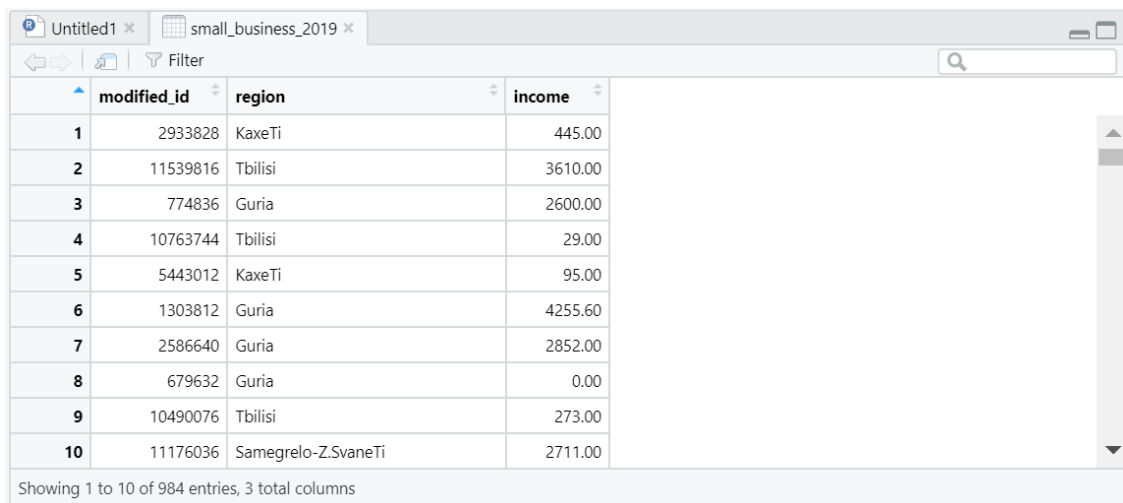
- Since dataframes are also objects, we can refer to them with their names (exm: `small_business_2019`)
- We'll see an example of that in the next exercise

Data in R // მონაცემები R

A note about this dataframe

Understanding the data you use is very important. For this training, `small_business_2019` is an example dataframe with business income data for 2019

- `modified_id` is a business identifier
- `region` is region where the business is
- `income` is the income the business reported in 2019



	modified_id	region	income
1	2933828	KaxeTi	445.00
2	11539816	Tbilisi	3610.00
3	774836	Guria	2600.00
4	10763744	Tbilisi	29.00
5	5443012	KaxeTi	95.00
6	1303812	Guria	4255.60
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 10 of 984 entries, 3 total columns

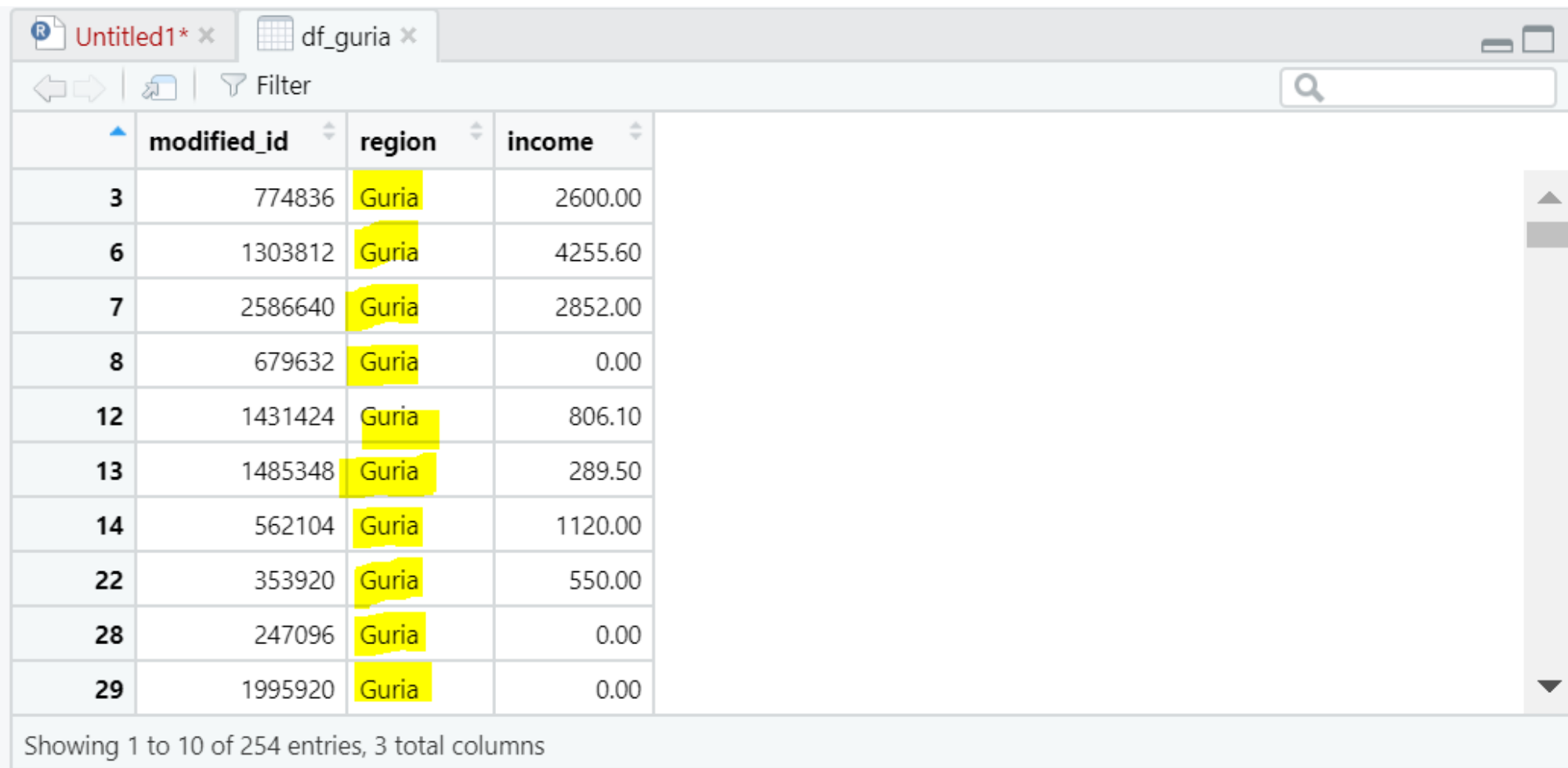
Exercise 4: Subset the data

1. Use the following code to subset `small_business_2019` and leave only the observations in the region named "Guria":

```
df_guria <- subset(small_business_2019, region == "Guria")
```

- Note that we are using the arrow operator (`<-`) to store the result
 - Note that there are **two equal signs** in the condition, not one
 - Also note that you need to write `"Guria"` enclosed in quotes and with uppercase `G`, because that's how it is in the data
2. Use `View(df_guria)` to visualize the dataframe again and see how it changed (note the uppercase "V")

Data in R // მონაცემები R



Untitled1* x df_guria x

Filter

	modified_id	region	income
3	774836	Guria	2600.00
6	1303812	Guria	4255.60
7	2586640	Guria	2852.00
8	679632	Guria	0.00
12	1431424	Guria	806.10
13	1485348	Guria	289.50
14	562104	Guria	1120.00
22	353920	Guria	550.00
28	247096	Guria	0.00
29	1995920	Guria	0.00

Showing 1 to 10 of 254 entries, 3 total columns

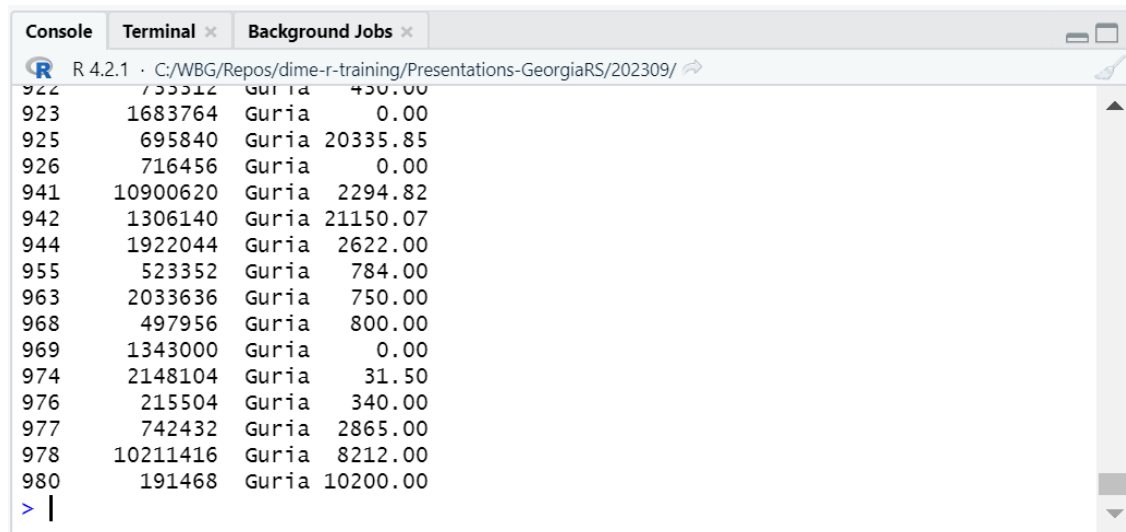
Data in R // მონაცემები R

Storing results in R

There is an important difference between using `<-` and not using it

- Not using `<-` **simply displays the result** in the console. The input dataframe will remain unchanged and the result **will not be stored**

```
subset(small_business_2019, region == "Guria")
```



The screenshot shows the R console output for the command `subset(small_business_2019, region == "Guria")`. The output is a table with 4 columns: an index, a numeric ID, the region name 'Guria', and a numeric value. The data is as follows:

Index	ID	Region	Value
922	755512	Guria	450.00
923	1683764	Guria	0.00
925	695840	Guria	20335.85
926	716456	Guria	0.00
941	10900620	Guria	2294.82
942	1306140	Guria	21150.07
944	1922044	Guria	2622.00
955	523352	Guria	784.00
963	2033636	Guria	750.00
968	497956	Guria	800.00
969	1343000	Guria	0.00
974	2148104	Guria	31.50
976	215504	Guria	340.00
977	742432	Guria	2865.00
978	10211416	Guria	8212.00
980	191468	Guria	10200.00

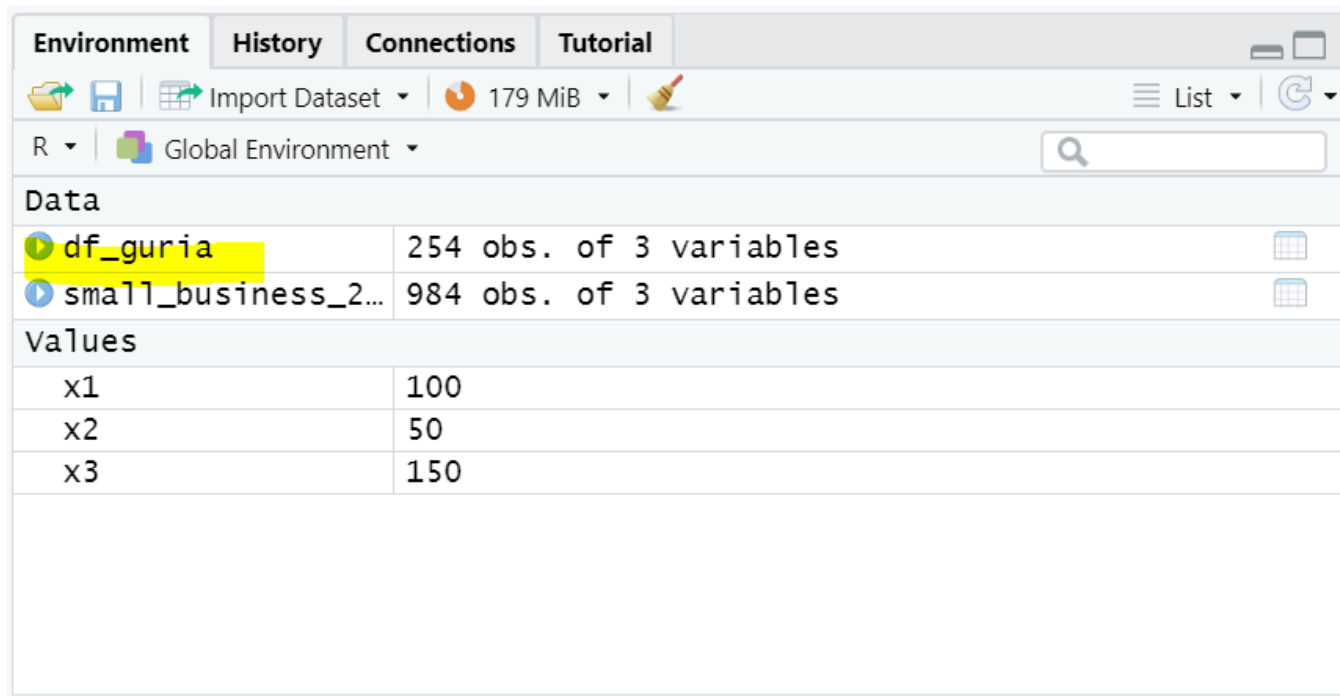
The console prompt `> |` is visible at the bottom.

Data in R // მონაცემები R

Storing results in R

- Using `<-` tells R that we want to **store the result in a new object**, which is the object at the left side of the arrow. This time the result will not be printed in the console but the new dataframe will show in the environment panel

```
df_guria <- subset(small_business_2019, region == "Guria")
```



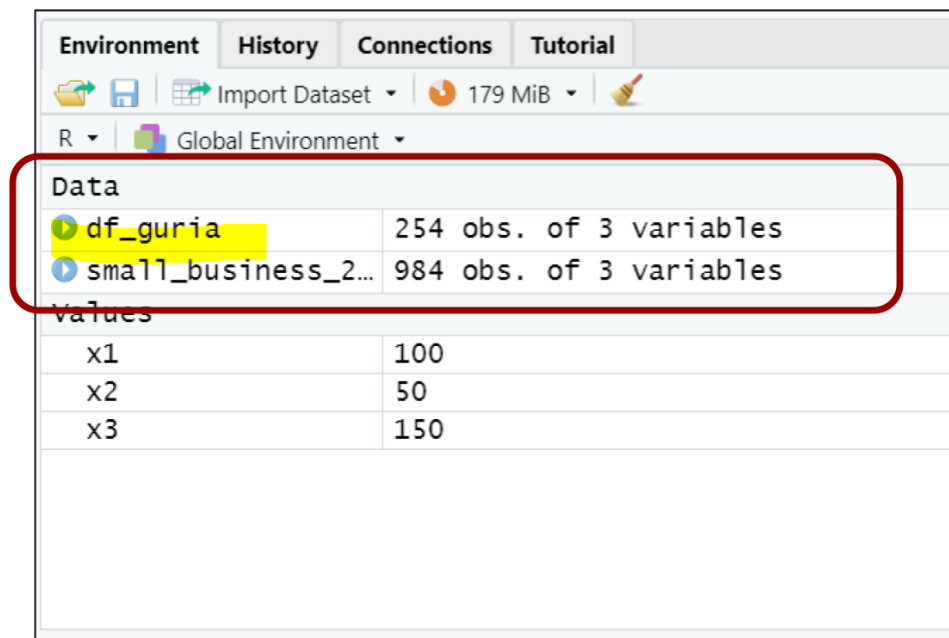
The screenshot shows the RStudio Environment panel. At the top, there are tabs for Environment, History, Connections, and Tutorial. Below the tabs, there are icons for file operations and a status bar showing 179 MiB. The main area displays the Global Environment with a search bar. Under the 'Data' section, two objects are listed: 'df_guria' (254 obs. of 3 variables) and 'small_business_2019' (984 obs. of 3 variables). Below the 'Data' section, there is a 'Values' section showing the first few rows of the data.

Data	
df_guria	254 obs. of 3 variables
small_business_2019	984 obs. of 3 variables

Values	
x1	100
x2	50
x3	150

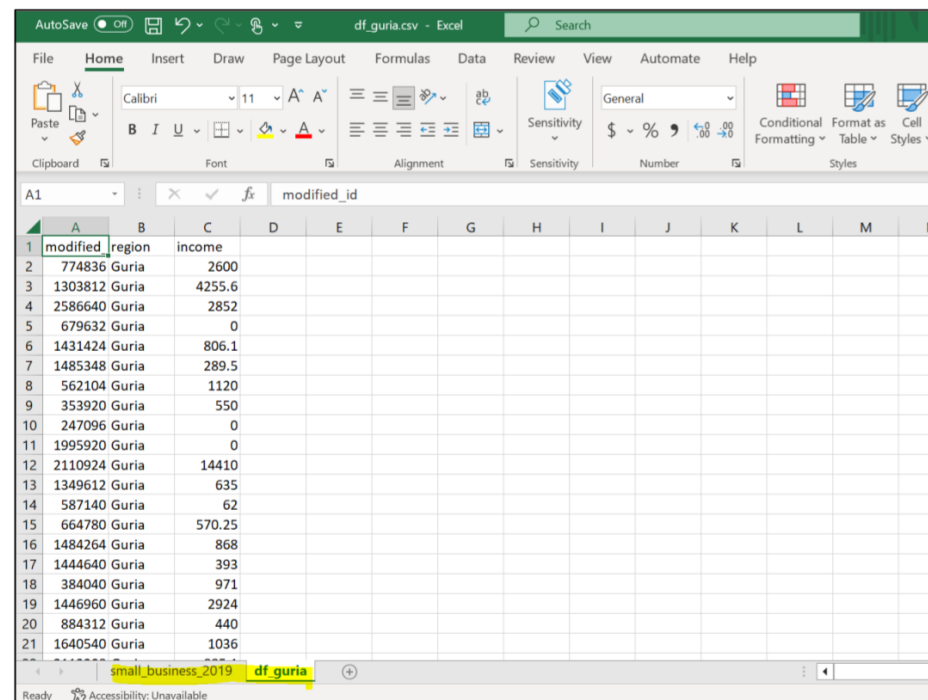
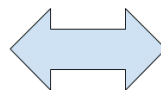
Data in R // მონაცემები R

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



The R Environment window displays the Global Environment. A red box highlights the 'Data' section, which contains two dataframes: 'df_guria' (254 obs. of 3 variables) and 'small_business_2019' (984 obs. of 3 variables). Below the 'Data' section, the 'Values' section shows the first few rows of the selected dataframe 'df_guria'.

Variable	Value
x1	100
x2	50
x3	150



The Excel spreadsheet displays data from the R environment. The first sheet, 'df_guria', shows columns 'modified_id', 'region', and 'income'. The second sheet, 'small_business_2019', shows columns 'modified_id', 'region', and 'income'.

modified_id	region	income
774836	Guria	2600
1303812	Guria	4255.6
2586640	Guria	2852
679632	Guria	0
1431424	Guria	806.1
1485348	Guria	289.5
562104	Guria	1120
353920	Guria	550
247096	Guria	0
1995920	Guria	0
2110924	Guria	14410
1349612	Guria	635
587140	Guria	62
664780	Guria	570.25
1484264	Guria	868
1444640	Guria	393
384040	Guria	971
1446960	Guria	2924
884312	Guria	440
1640540	Guria	1036

Object types // ობიექტების ტიპები

Object types // ობიექტების ტიპები

- The objects in your environment have different types depending on the type of data they represent
- Different types of objects allow to apply different operations to them or apply the same operation in a different way

Object types // ობიექტების ტიპები

- You can always check the type of an object with the function `class()`

```
class(x1)
```

```
## [1] "numeric"
```

```
class(df_guria)
```

```
## [1] "data.frame"
```

Object types // ობიექტების ტიპები

We have worked with two classes of object until now:

- **Numeric:** single number values we can use for mathematical operations. The objects `x1`, `x2`, and `x3` are numeric. They are similar to values you would store in a single Excel cell
- **Dataframe:** a collection of values organized in rows and columns. `small_business_2019` and `df_guria` are dataframes. They are similar to an Excel spreadsheet

In the next exercise we will learn about another object type.

Object types // ობიექტების ტიპები

Vectors

- Vectors are a collection of values **with a single dimension**, instead of being organized in rows and columns as dataframes
- You can think of a vector in R as a single column in an Excel spreadsheet or an R dataframe
- You can create vectors with the function `c()`, the vector elements are separated by commas

```
my_vector <- c(4, 8, 2, 5)
```

Object types // ობიექტების ტიპები

Exercise 5: create and operate vectors

1- Create a vector with the elements 3, 8, and 10 and name it `v1`:

```
v1 <- c(3, 8, 10)
```

2- Create a second vector with the elements 7, 2, and 5 and name it `v2`

```
v2 <- c(7, 2, 5)
```

3- Create a third vector named `result1` with the sum of `v1` and `v2`:

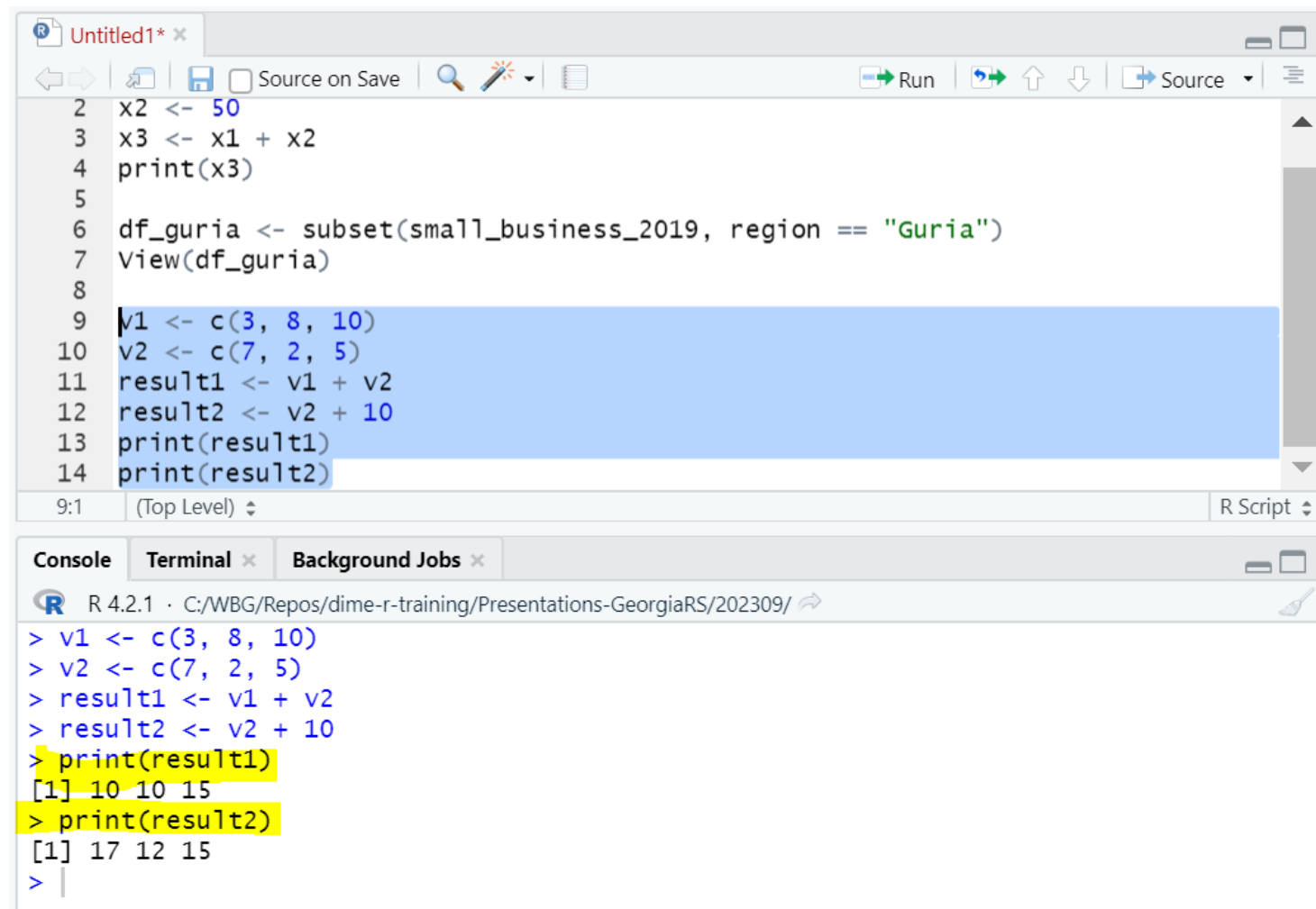
```
result1 <- v1 + v2
```

4- Lastly, create a fourth vector named `result2` with the sum of `v2` plus ten:

```
result2 <- v2 + 10
```

5- Print `result1` and `result2` and observe the results

Object types // ობიექტების ტიპები



```
2 x2 <- 50
3 x3 <- x1 + x2
4 print(x3)
5
6 df_guria <- subset(small_business_2019, region == "Guria")
7 View(df_guria)
8
9 v1 <- c(3, 8, 10)
10 v2 <- c(7, 2, 5)
11 result1 <- v1 + v2
12 result2 <- v2 + 10
13 print(result1)
14 print(result2)
```

9:1 (Top Level) R Script

Console Terminal Background Jobs

R 4.2.1 · C:/WBG/Repos/dime-r-training/Presentations-GeorgiaRS/202309/

```
> v1 <- c(3, 8, 10)
> v2 <- c(7, 2, 5)
> result1 <- v1 + v2
> result2 <- v2 + 10
> print(result1)
[1] 10 10 15
> print(result2)
[1] 17 12 15
>
```

Object types // ობიექტების ტიპები

Environment		History	Connections	Tutorial
R		Global Environment	123 MiB	
Data				
df_guria	254 obs. of 3 variables			
\$ modified_id:	int 774836 1303812 2586640 679632 1431424 1485...			
\$ region	: chr "Guria" "Guria" "Guria" "Guria" ...			
\$ income	: num 2600 4256 2852 0 806 ...			
small_business_2...	984 obs. of 3 variables			
Values				
result1	num [1:3] 10 10 15			
result2	num [1:3] 17 12 15			
v1	num [1:3] 3 8 10			
v2	num [1:3] 7 2 5			
x1	100			

Object types // ობიექტების ტიპები

Notice two things:

1.- Operating two vectors applies the operation **element-wise**

```
> v1  
[1] 3 8 10  
+  
> v2  
[1] 7 2 5  
=  
> result1  
[1] 10 10 15
```

2.- Operating a vector with a numeric object will apply the **same operation to every element of the vector**

```
> v2  
[1] 7 2 5  
+ 10  
=  
> result2  
[1] 17 12 15
```

Functions in R // ფუნქციონირებს R

Functions in R // ფუნქციონირებს 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 `paste()` are functions
- Everything that has a name plus parentheses is a function in R

```
subset(small_business_2019, region == "Guria")
```

Functions in R // ფუნქციონირებს R

Functions have the following syntax:

```
subset(small_business_2019, region == "Guria")
```



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 // ფუნქციონირებს R

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

```
df_guria <- subset(small_business_2019, region == "Guria")
```

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

Functions in R // ფუნქციონირებს R

Exercise 6: Using the function `summary()`

1. Compute the summary statistics of the variables of `small_business_2019` and save the result with `summary_table <- summary(small_business_2019)`
2. Print the stored result with `print(summary_table)`

Functions in R // ფუნქციონირებს R

Note that this code is both creating a new object (with `summary_table <- summary(small_business_2019)`) 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. Lines 18 and 19, `summary_table <- summary(small_business_2019)` and `print(summary_table)`, are highlighted in blue. A red box encloses these two lines. A red arrow points from this box to the 'Run' button.
- Run Button:** A red box encloses the 'Run' button in the top toolbar. A red arrow points from the 'Run' button to the 'summary_table' entry in the Environment pane.
- Environment Pane:** Shows the 'Data' environment. The 'summary_table' object is listed with its class and structure: `'table' chr [1:6, 1:3] "Min. : 19832 " "...'`. A red box encloses this entry. A red arrow points from this box to the console output.
- Console:** Shows the output of the `print(summary_table)` command. The output is a summary table with columns `modified_id`, `region`, and `income`. A red box encloses the entire console output. A red arrow points from this box to the `print(summary_table)` line in the source editor.

Console Output:

modified_id	region	income
Min. : 19832	Length:984	Min. : 0.0
1st Qu.: 2012424	Class :character	1st Qu.: 112.7
Median : 5004096	Mode :character	Median : 900.0
Mean : 5450890		Mean : 3299.8
3rd Qu.: 9112758		3rd Qu.: 3200.0
Max. : 12296912		Max. : 139394.5

Questions? // კითხვები?

Wrapping up // შეფუთვა

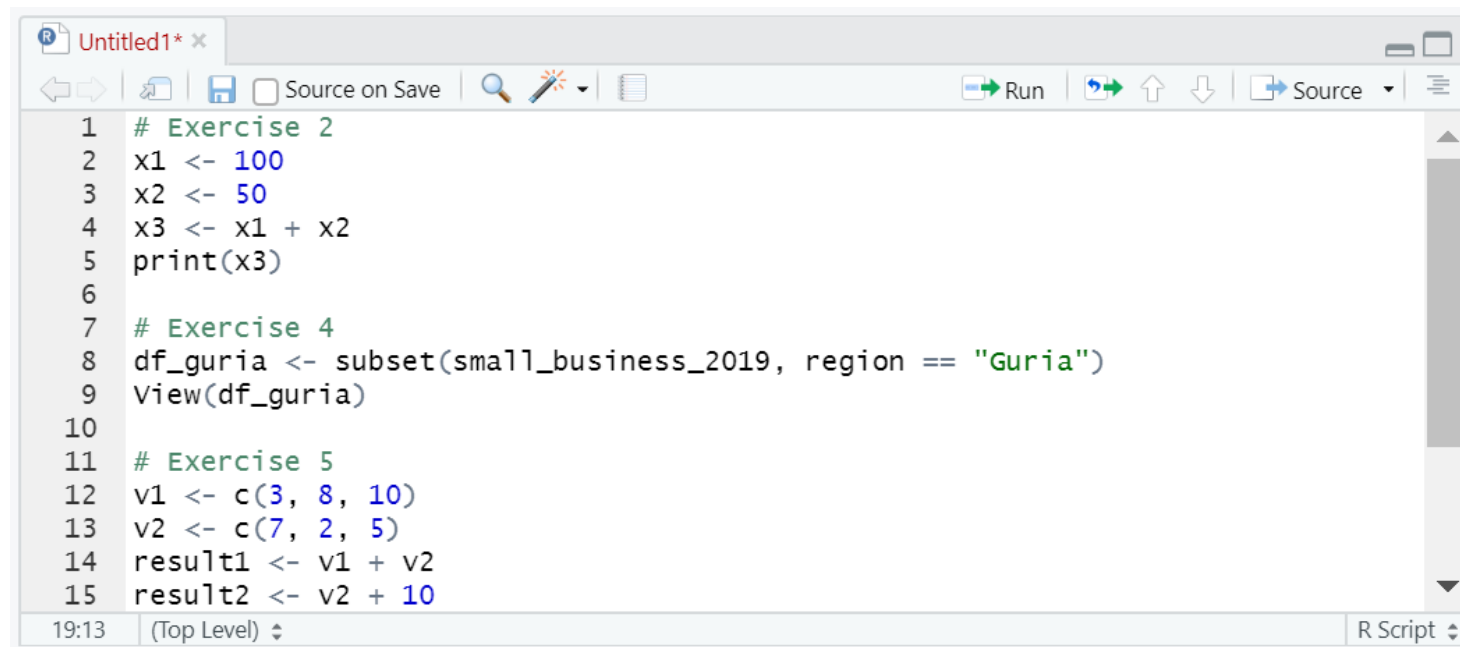
Wrapping up // შეფუთვა

Add code comments!

- Every line of code that starts with the pound symbol (`#`) will be ignored when R executes the code
- This means that you can add any clarifying comment with `#`. These are called **code comments**
- It's always a good practice to add code comments for yourself to later remember what the code is doing or to explain your code to others if you'll share it

Wrapping up // შეფუთვა

- Try adding code comments to your script so you will remember which part corresponds to each exercise



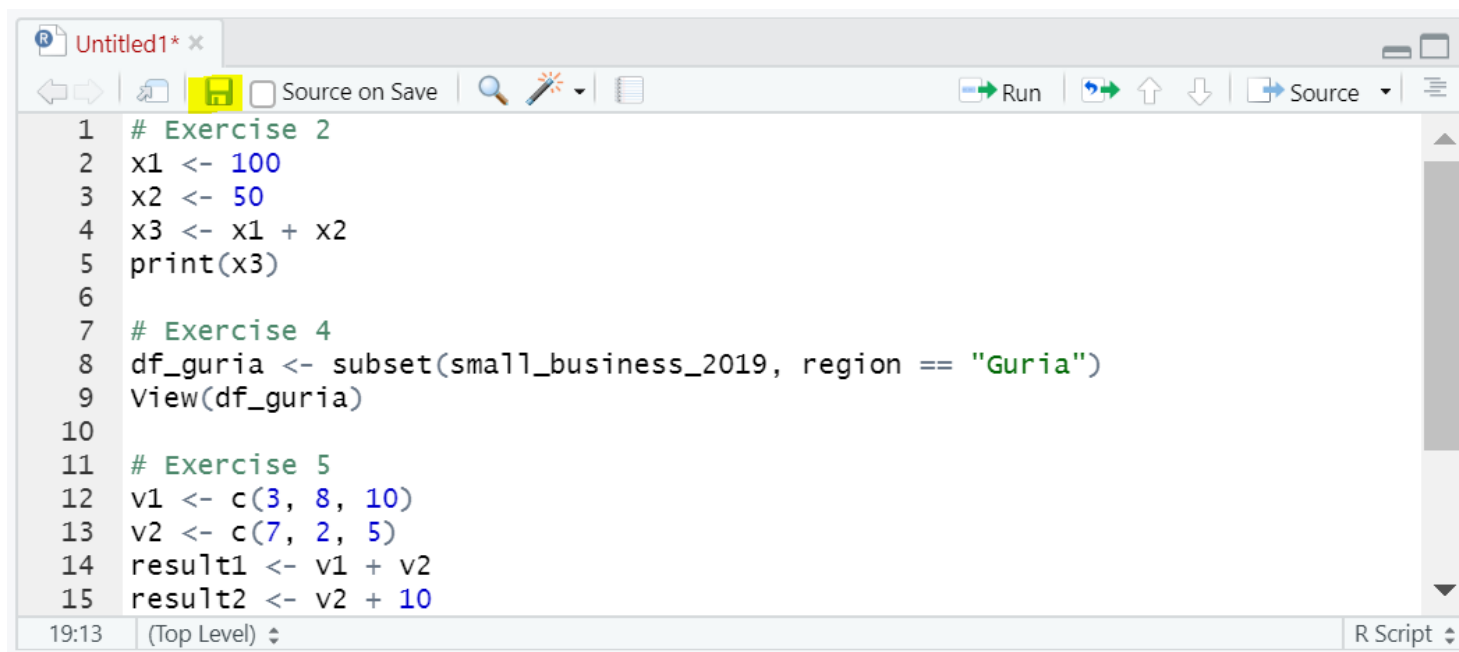
```
1 # Exercise 2
2 x1 <- 100
3 x2 <- 50
4 x3 <- x1 + x2
5 print(x3)
6
7 # Exercise 4
8 df_guria <- subset(small_business_2019, region == "Guria")
9 View(df_guria)
10
11 # Exercise 5
12 v1 <- c(3, 8, 10)
13 v2 <- c(7, 2, 5)
14 result1 <- v1 + v2
15 result2 <- v2 + 10
```

19:13 (Top Level) R Script

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



The screenshot shows the RStudio editor interface. The title bar indicates the file is 'Untitled1*.x'. The toolbar includes icons for navigation, saving (a floppy disk icon), and running code. The source editor contains the following R code:

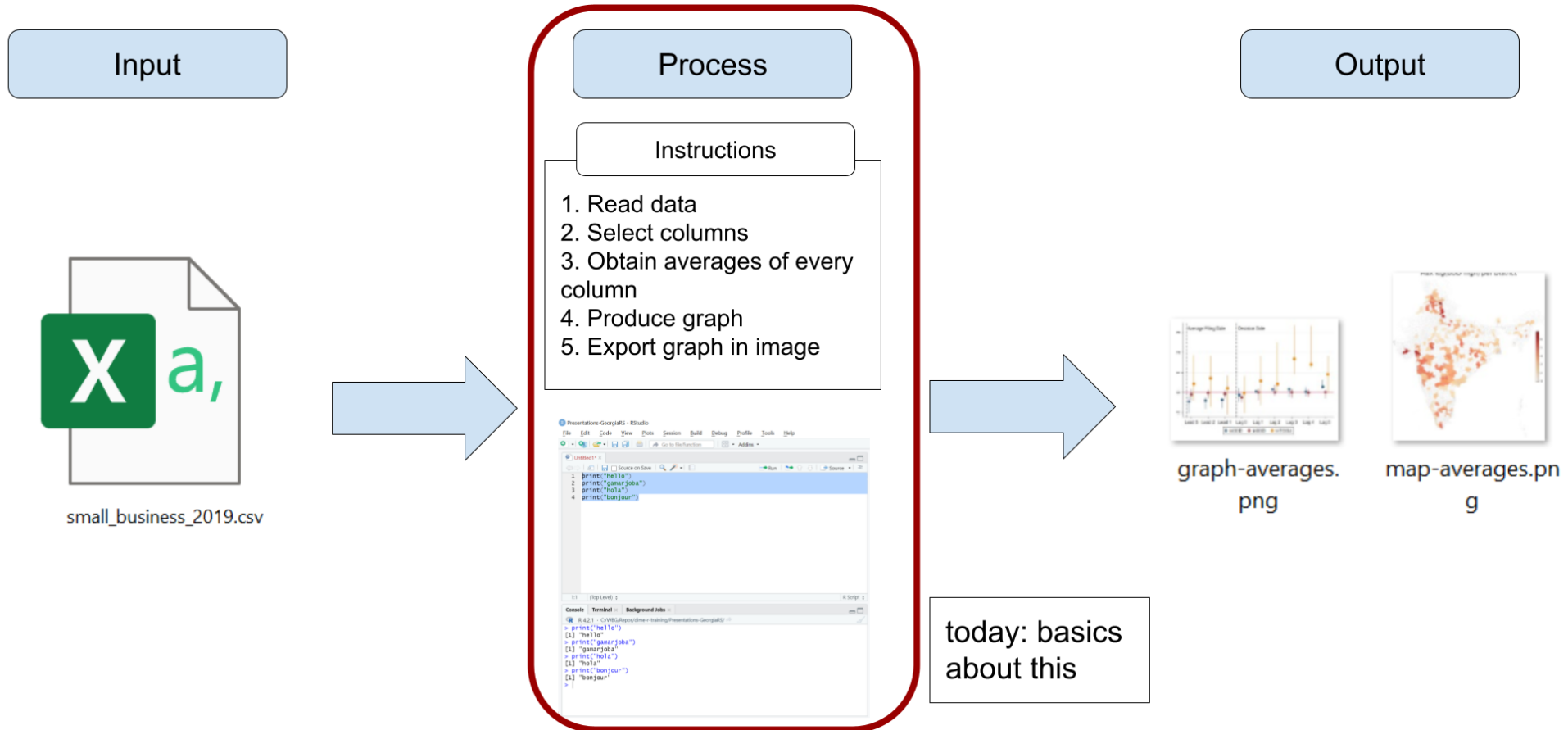
```
1 # Exercise 2
2 x1 <- 100
3 x2 <- 50
4 x3 <- x1 + x2
5 print(x3)
6
7 # Exercise 4
8 df_guria <- subset(small_business_2019, region == "Guria")
9 view(df_guria)
10
11 # Exercise 5
12 v1 <- c(3, 8, 10)
13 v2 <- c(7, 2, 5)
14 result1 <- v1 + v2
15 result2 <- v2 + 10
```

The status bar at the bottom shows the time '19:13', the environment '(Top Level)', and the file type 'R Script'.

Wrapping up // შეჯუთვა

This session

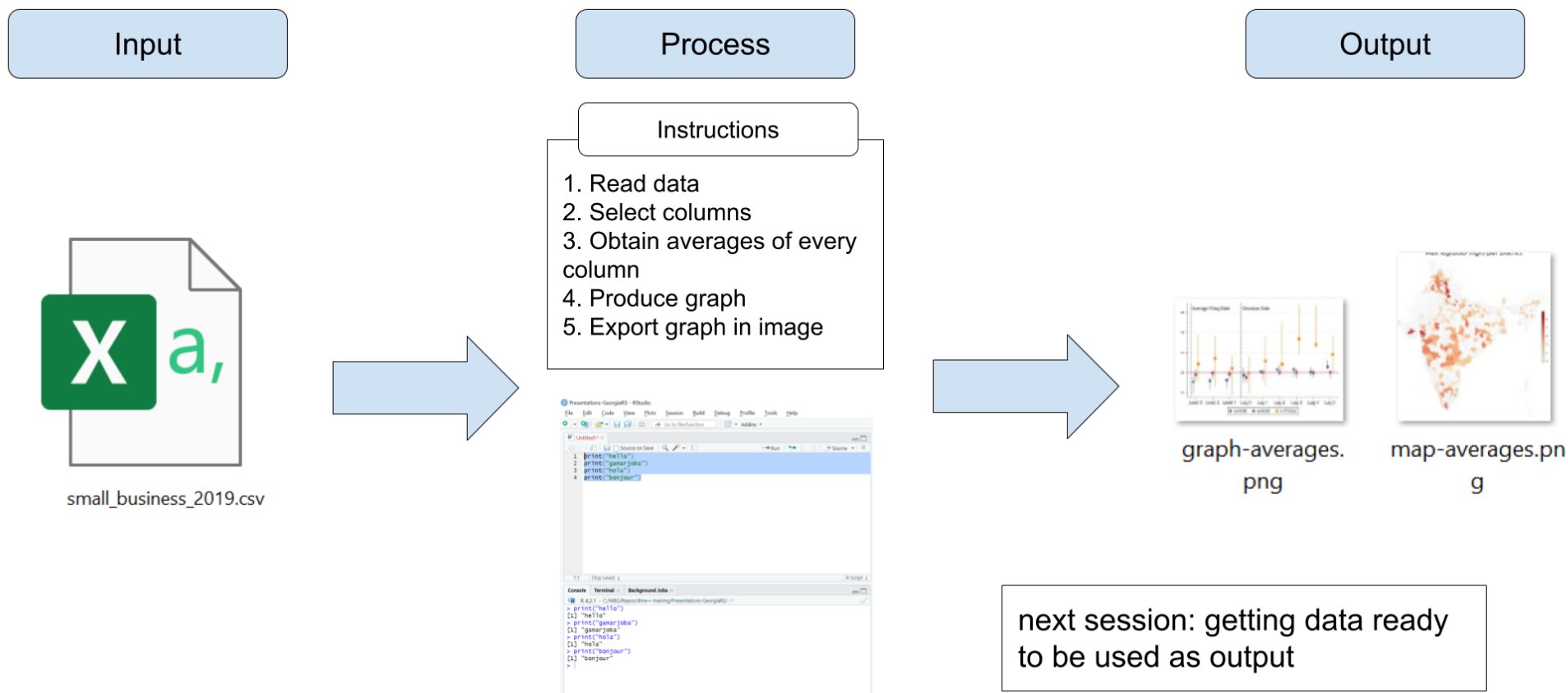
This first session focused on the basics for writing R code



Wrapping up // შეჯუთვა

Next session

In the next session we will learn how to get data ready to be exported as outputs



Thanks! // მადლობა! // ¡Gracias! // Obrigado!

Appendix // ღანაშობი

Object types: character strings

- Character strings are collections of alphanumeric characters usually representing words or texts, or just characters in general

```
s1 <- "gamar joba"  
print(s1)
```

```
## [1] "gamar joba"
```

- Strings characters are **always enclosed in quotes** (" ")
- They are usually referred to as just **strings**

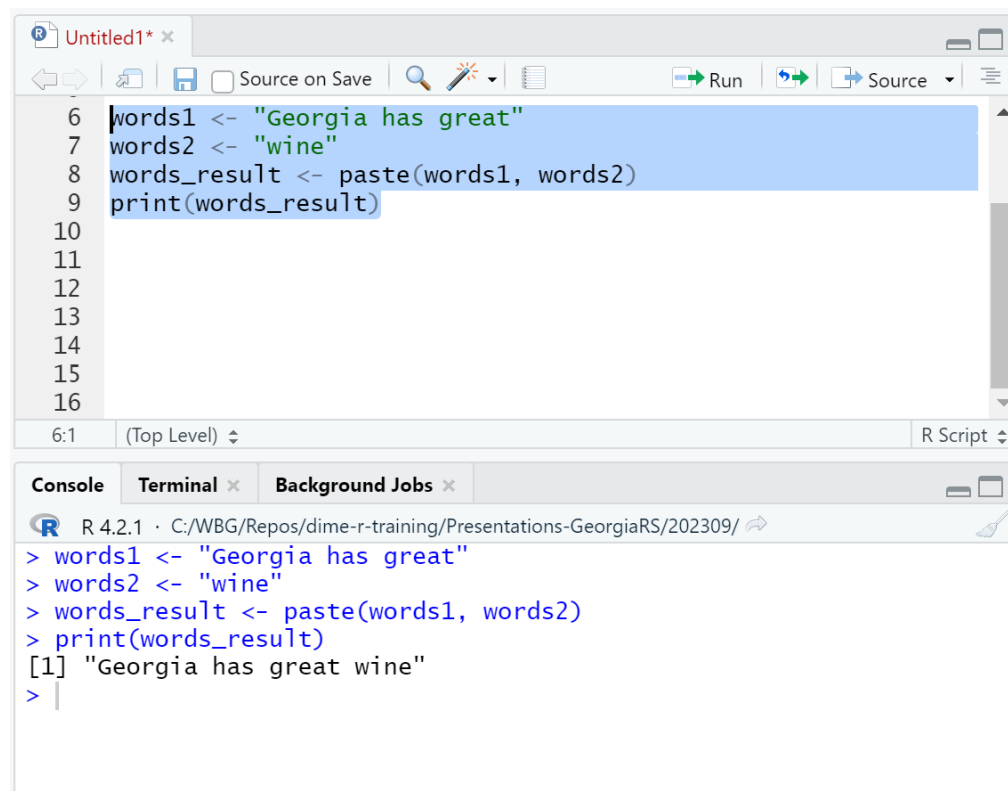
Exercise: create and operate character strings

1. Create a character string object with the words "Georgia has great" and name it `words1`
2. Create a second string with the words "wine" and name it `words2`
 - Don't forget to use `<-` to create the string objects
 - Remember to include the quotes: " "
3. Use the following code to concatenate `words1` and `words2`, save the result in `words_result`, and print it:

```
words_result <- paste(words1, words2)
print(words_result)
```

Appendix // ღანაო

Object types: character strings



The image shows a screenshot of an R script editor and its console. The script editor, titled 'Untitled1*', contains the following R code:








```
6 words1 <- "Georgia has great"
7 words2 <- "wine"
8 words_result <- paste(words1, words2)
9 print(words_result)
```

The code is highlighted in blue. The console, located at the bottom, shows the execution of the script:

```
R 4.2.1 · C:/WBG/Repos/dime-r-training/Presentations-GeorgiaRS/202309/
> words1 <- "Georgia has great"
> words2 <- "wine"
> words_result <- paste(words1, words2)
> print(words_result)
[1] "Georgia has great wine"
>
```

Appendix // ღანაო

Object types: character strings

Environment	History	Connections	Tutorial	
 	 Import ▾	 107 MiB ▾		≡ List ▾ 
R ▾	 Global Environment ▾	<input type="text"/>		
Values				
words_resu...	"Georgia has great wine"			
words1	"Georgia has great"			
words2	"wine"			
x1	100			
x2	50			
x3	150			