

# Session 4 - Data visualization

R training - Georgia RS-WB DIME

---

Luis Eduardo San Martin  
The World Bank | September 2023



# Table of contents // სარჩევი

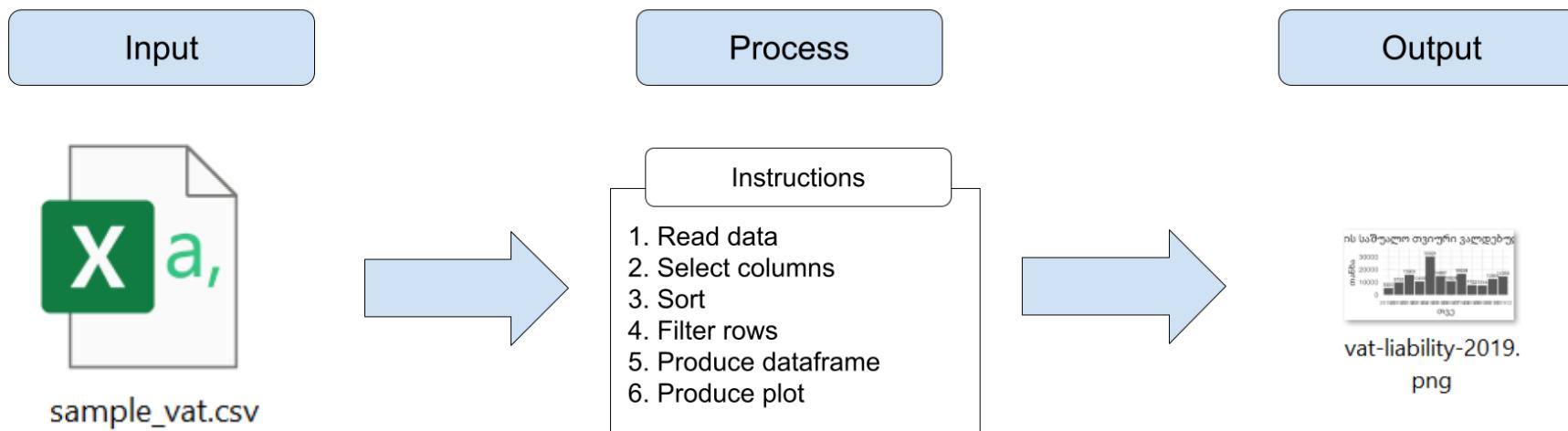
- Introduction
- The grammar of graphics
- Bar plots
- Line plots
- Text encodings
- Scatter plots
- Wrapping up

# Introduction // გაცნობა

---

# Introduction // გაცნობა

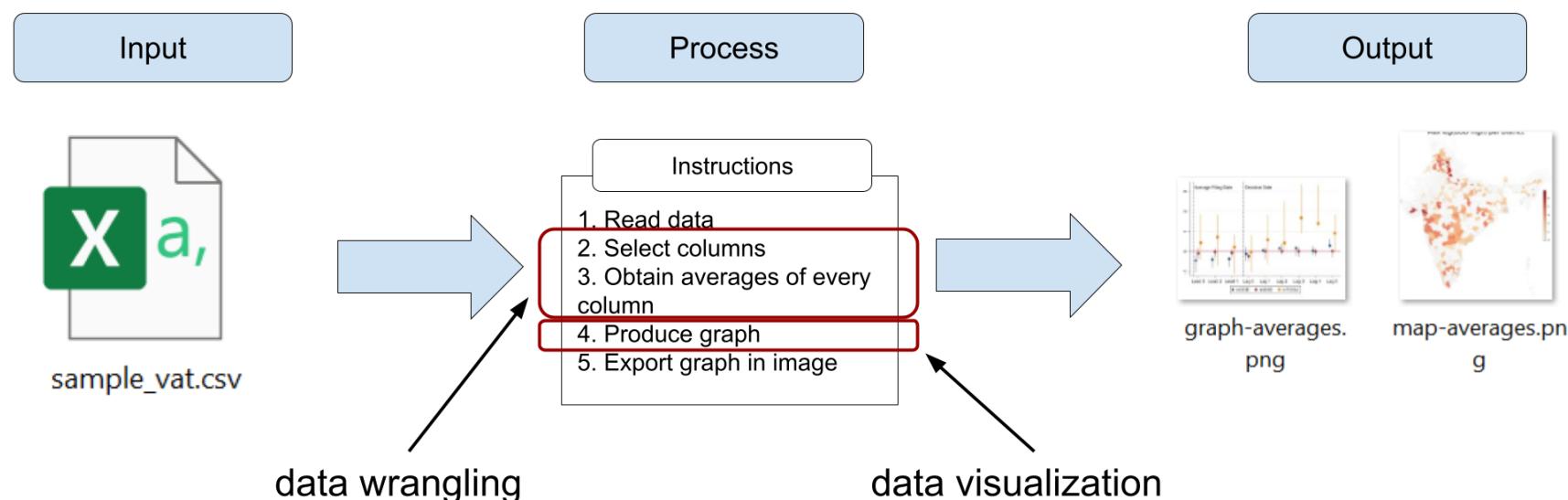
## About this session



# Introduction // ડાયુનિવર્સિટી

## Data visualization in the data work pipeline

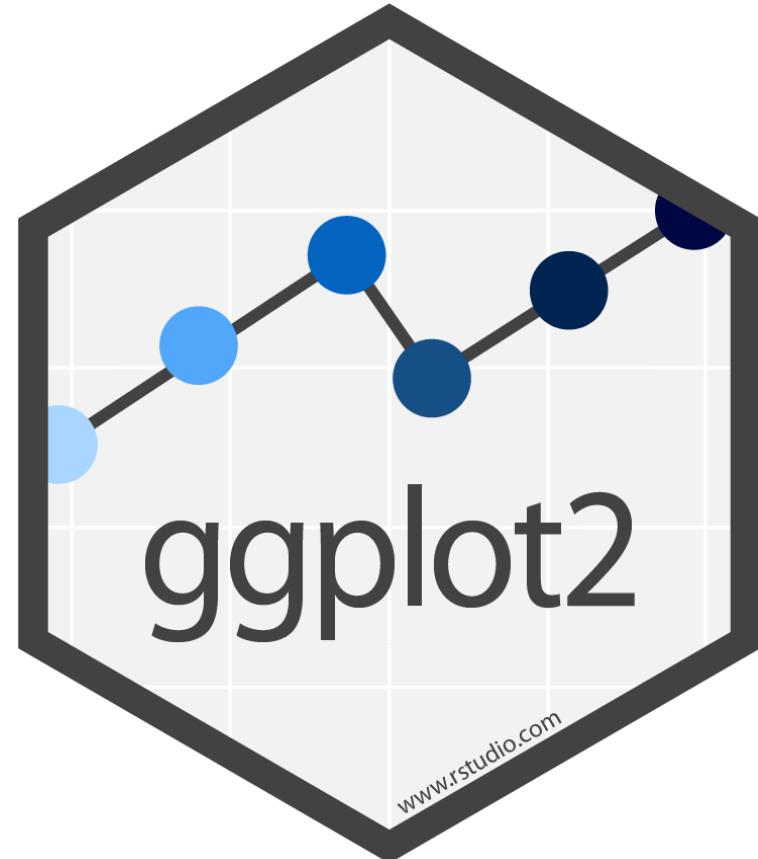
- Compared to other kind of outputs, data visualization involves an extra step after data wrangling: producing the visualization itself
- We also us R code to produce data visualizations
- The input for that code is a wrangled dataframe, dataframe that is ready to be used



# Introduction // გაცნობა

## Data visualization in R

- We'll use the package `ggplot2` to create data visualizations
- `ggplot2` greatly facilitates producing plots in R
  - It follows a syntax based on a description of the plot you want to obtain
  - This syntax is called **grammar of graphics**, a benchmark method of data visualization definition in statistical programming



# The grammar of graphics // გრაფიკის გრამატიკა

---

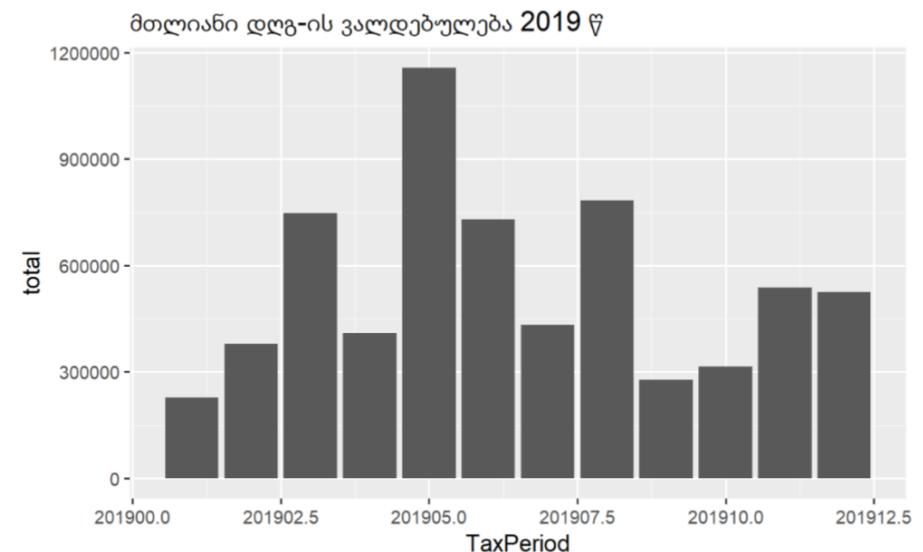
# The grammar of graphics // გრაფიკის გრამატიკა

## The grammar of graphics in `ggplot2`

	TaxPeriod	total	average
1	201901	227896.3	5299.914
2	201902	379579.4	9732.806
3	201903	747746.1	15909.492
4	201904	409453.0	10498.796
5	201905	1156129.5	30424.461
6	201906	729448.7	14886.709
7	201907	433130.7	10828.267
8	201908	782968.2	16658.898
9	201909	277992.6	7722.017
10	201910	314504.6	7314.061

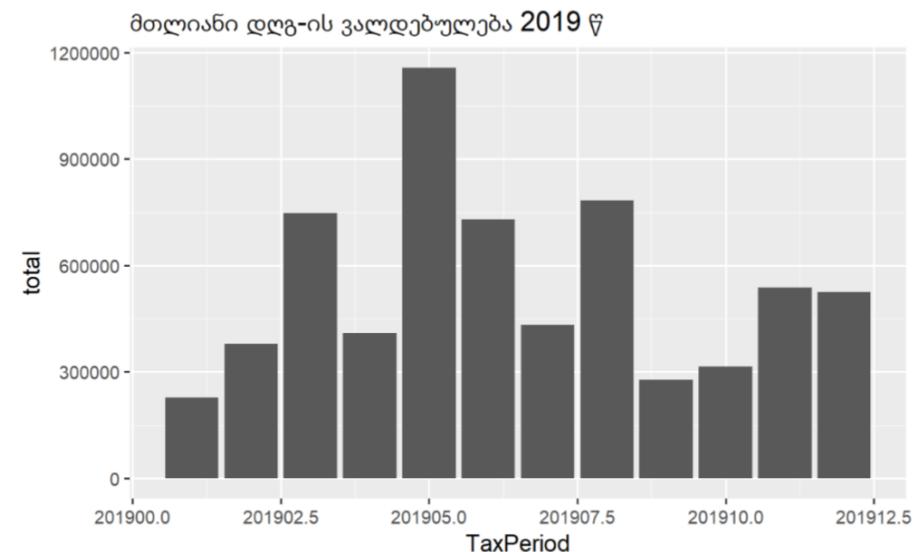
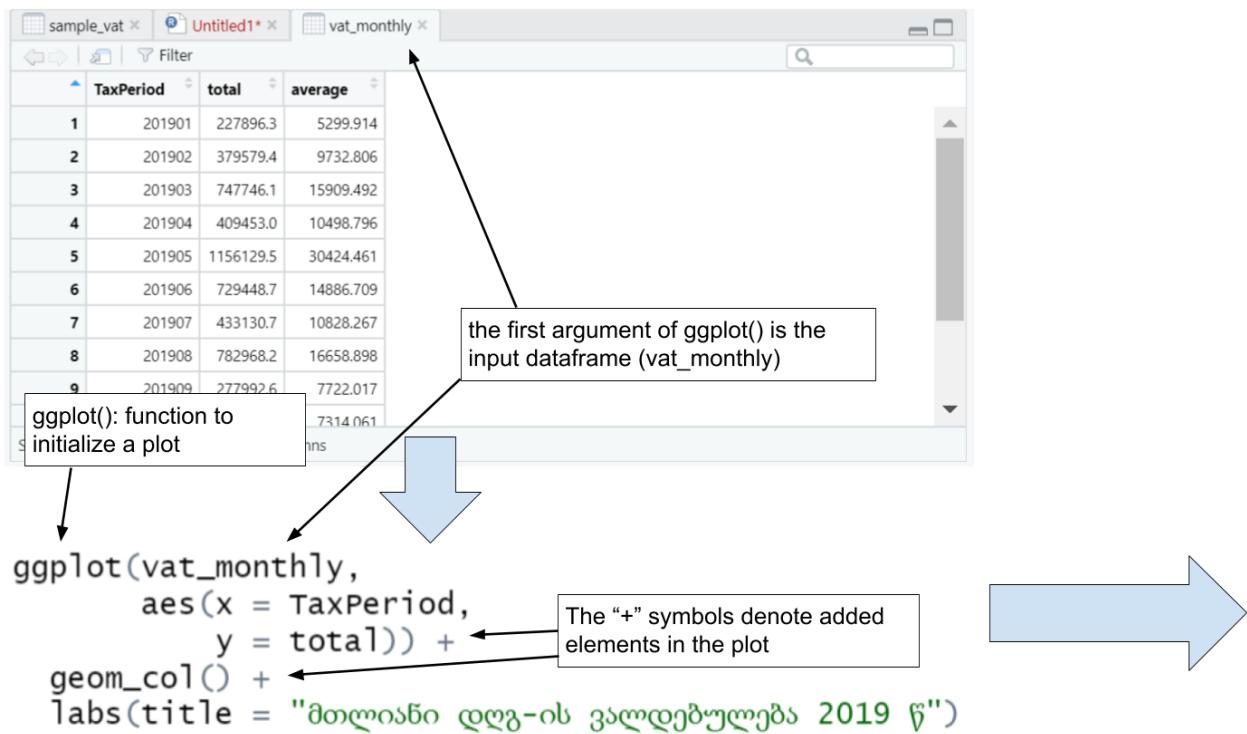


```
ggplot(vat_monthly,
       aes(x = TaxPeriod,
            y = total)) +
  geom_col() +
  labs(title = "მთლიანი დღგ-ის ვალდებულება 2019 წ")
```



# The grammar of graphics // გრაფიკის გრამატიკა

## The grammar of graphics in ggplot2



# The grammar of graphics // გრაფიკის გრამატიკა

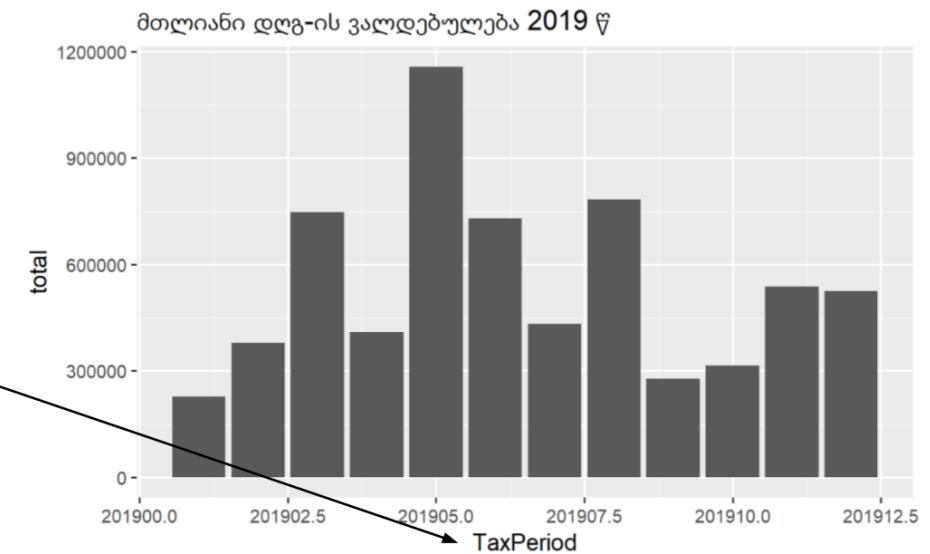
## The grammar of graphics in ggplot2

	TaxPeriod	total	average
1	201901	227896.3	5299.914
2	201902	375579.4	9732.806
3	201903	747746.1	15909.492
4	201904	409453.0	10498.796
5	201905	1156129.5	30424.461
6	201906	729448.7	14886.709
7	201907	433130.7	10828.267
8	201908	782968.2	16658.898
9	201909	277992.6	7722.017
10			

the second argument of ggplot()  
defines the plot aesthetics

```
ggplot(vat_monthly,  
       aes(x = TaxPeriod,  
            y = total)) +  
  geom_col() +  
  labs(title = "მთლიანი დღგ-ის ვალდებულება 2019 წ")
```

x axis (TaxPeriod)

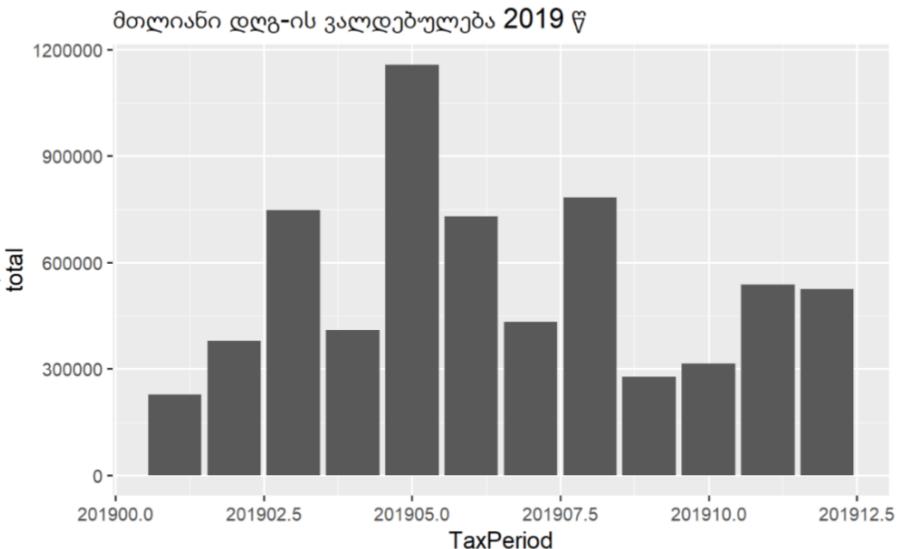


# The grammar of graphics // გრაფიკის გრამატიკა

## The grammar of graphics in ggplot2

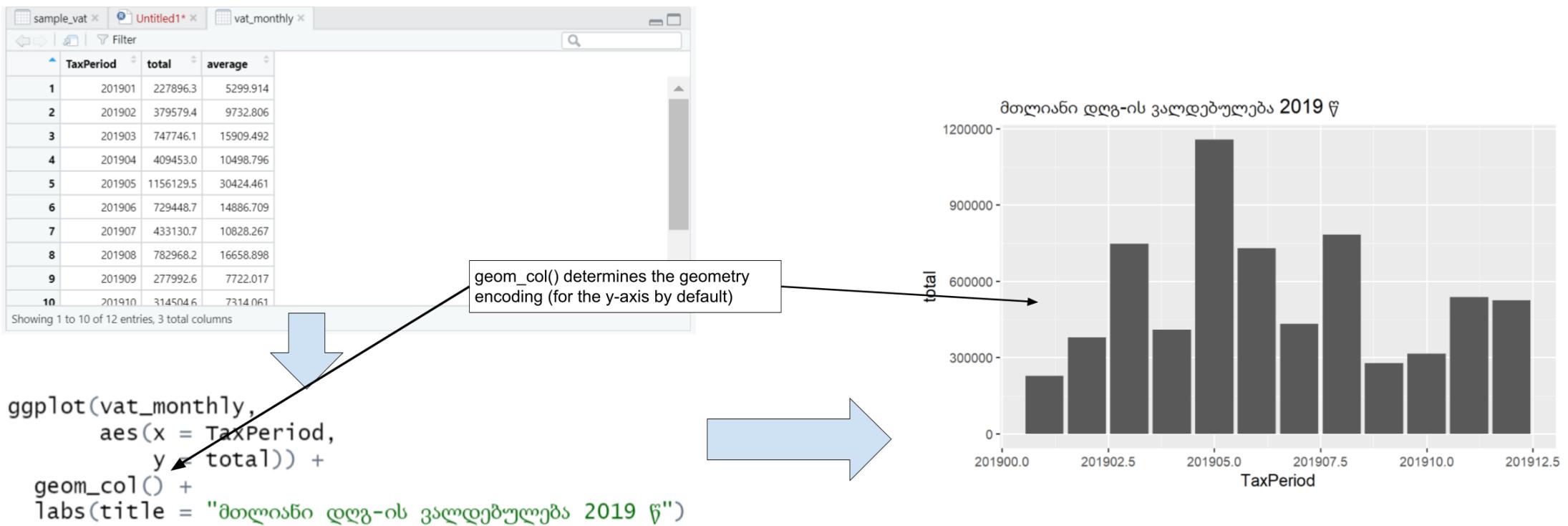
	TaxPeriod	total	average
1	201901	227896.3	5299.914
2	201902	379579.4	9732.806
3	201903	747746.1	15909.492
4	201904	409453.0	10498.796
5	201905	1156129.5	30424.461
6	201906	729448.7	14886.709
7	201907	433130.7	10828.267
8	201908	782968.2	16658.898
9	201909	277992.6	7722.017

```
ggplot(vat_monthly,  
       aes(x = TaxPeriod,  
            y = total)) +  
  geom_col() +  
  labs(title = "მთლიანი დღგ-ის ვალდებულება 2019 წ")
```



# The grammar of graphics // გრაფიკის გრამატიკა

## The grammar of graphics in `ggplot2`



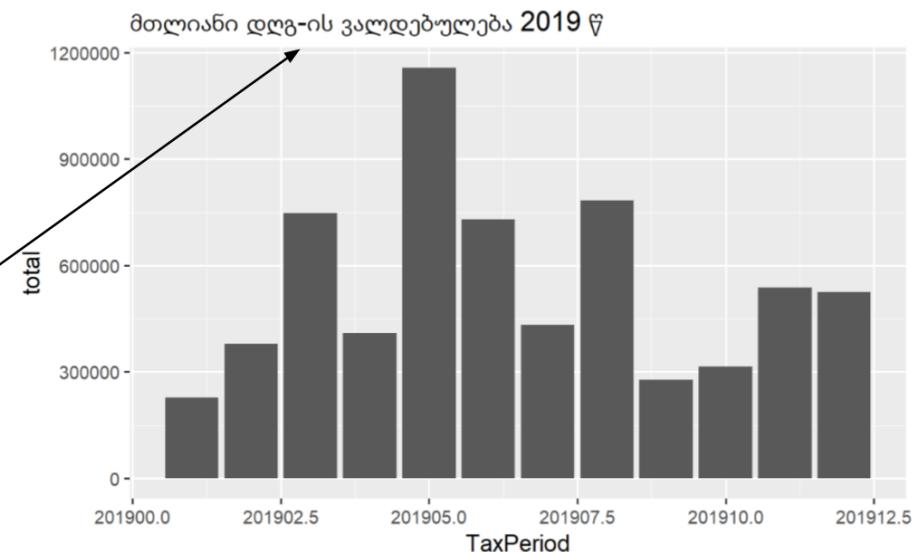
# The grammar of graphics // გრაფიკის გრამატიკა

## The grammar of graphics in ggplot2

	TaxPeriod	total	average
1	201901	227896.3	5299.914
2	201902	379579.4	9732.806
3	201903	747746.1	15909.492
4	201904	409453.0	10498.796
5	201905	1156129.5	30424.461
6	201906	729448.7	14886.709
7	201907	433130.7	10828.267
8	201908	782968.2	16658.898
9	201909	277992.6	7722.017
10	201910	314504.6	7314.061

```
ggplot(vat_monthly,  
       aes(x = TaxPeriod,  
            y = total)) +  
  geom_col() +  
  labs(title = "მთლიანი დღგ-ის ვალდებულება 2019 წ")
```

labs() determines the labels, for the plot title and others as well



# Bar plots // ბარის ნაკვეთები

---

# Bar plots // ბარის ნაკვეთები

## Exercise 1a: Create a basic bar plot

1. Open a new script for this session by clicking on **File** >> **New File** >> **R Script**

2. Load **ggplot2**

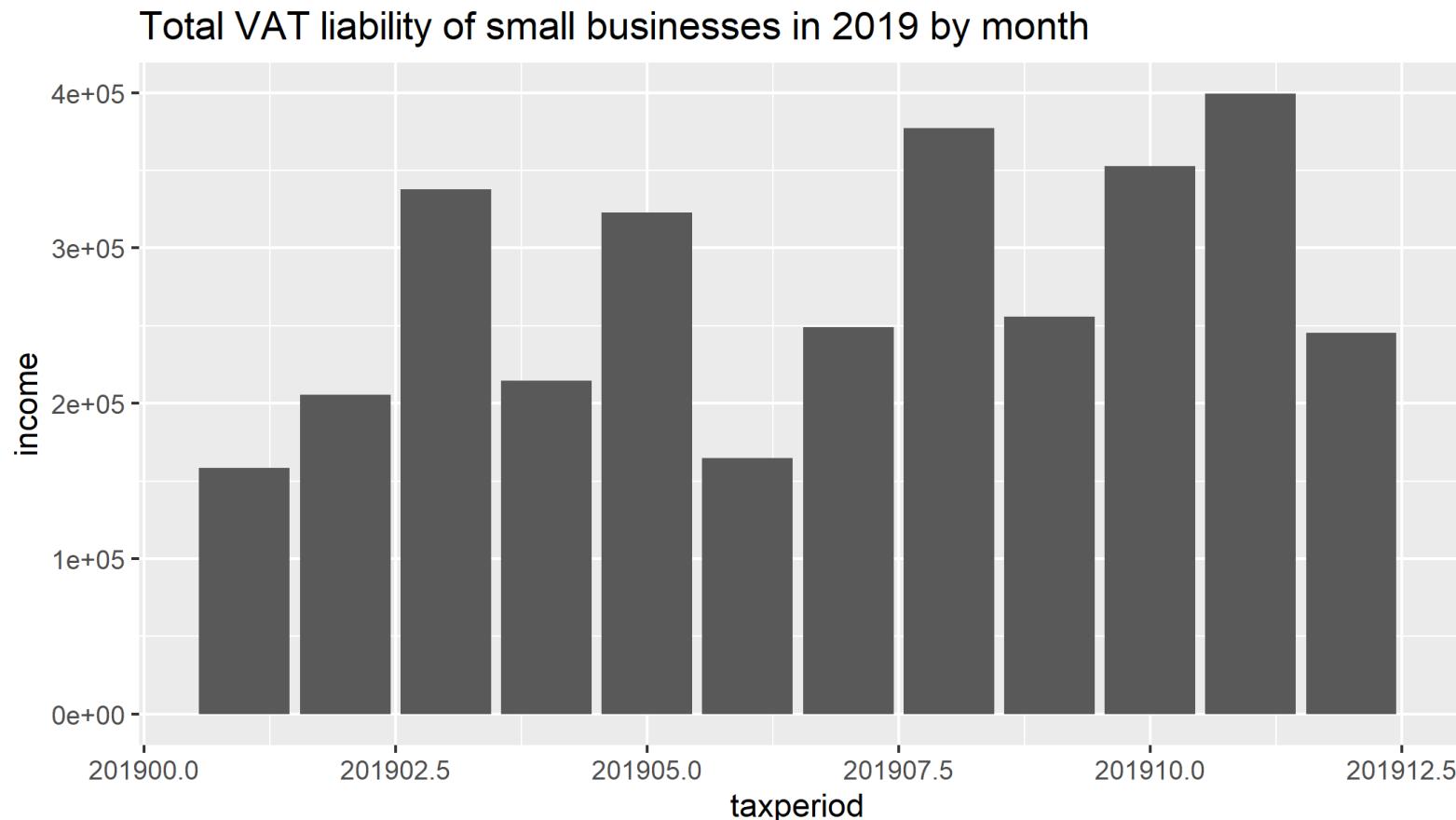
```
library(ggplot2)
```

1. Produce a basic bar plot with the following code:

```
ggplot(small_business_2019_all) +  
  aes(x = taxperiod,  
       y = vat_liability) +  
  geom_col() +  
  labs(title = "Total VAT liability of small businesses in 2019 by month")
```

# Bar plots // ბარის ნაკვეთები

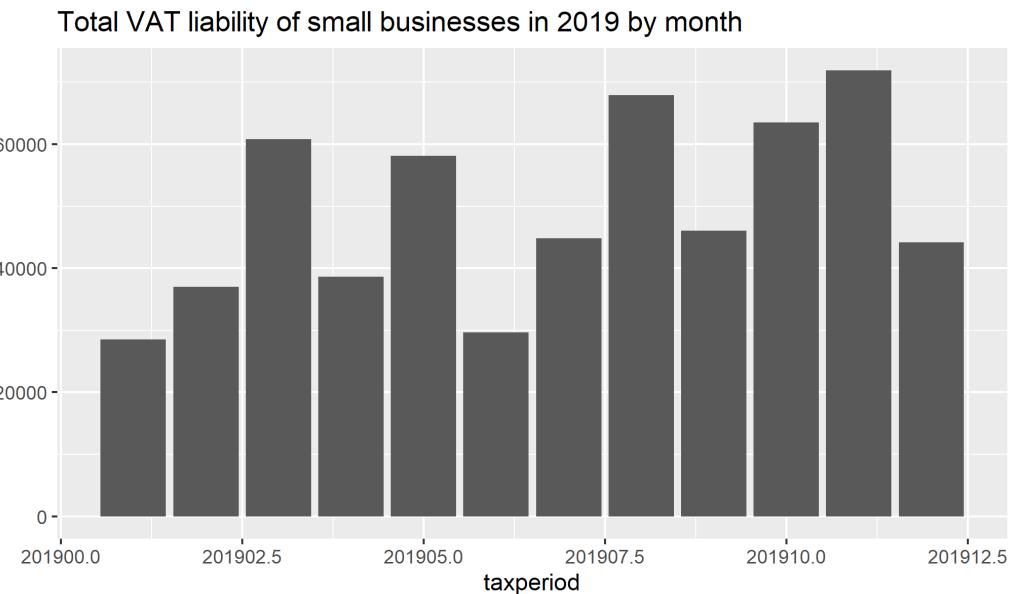
This result should be displayed in the lower right panel of your RStudio window



# Bar plots // ბარის ნაკვეთები

This plot looks acceptable but it can be improved:

- `taxperiod` is a variable representing months but R doesn't know it and it's showing the x-axis with decimals. We need to tell R that those values shouldn't be changed
- We can center the title
- We should add axis labels (instead of just "vat\_liability" and "taxperiod")



# Bar plots // ბარის ნაკვეთები

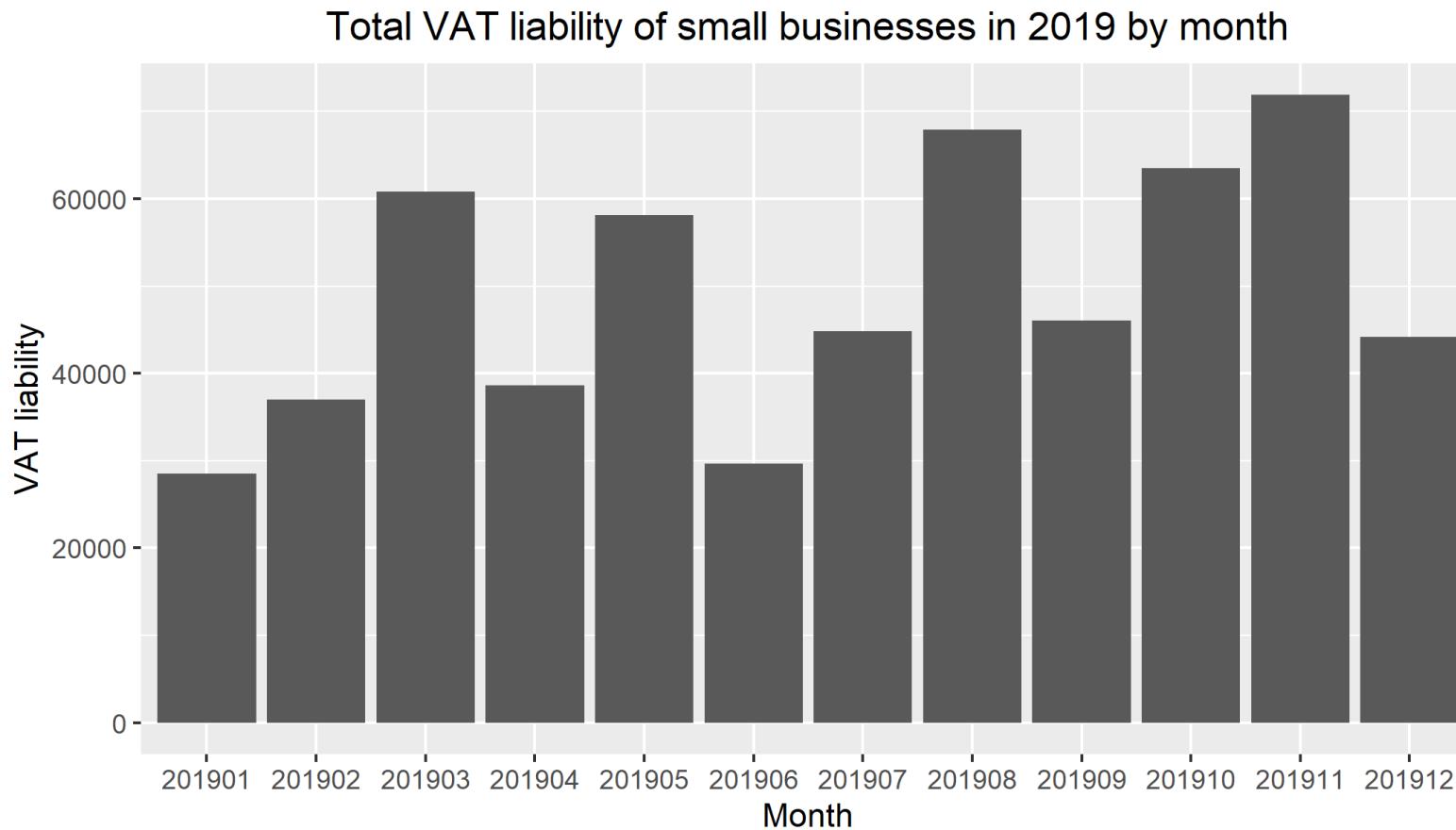
## Exercise 1b: Improve your bar plot

1. Use the following code to improve the aesthetics of your plot

```
ggplot(small_business_2019_all) +  
  aes(x = taxperiod,  
       y = vat_liability) +  
  geom_col() +  
  labs(title = "Total VAT liability of small businesses in 2019 by month",  
        # x-axis title  
        x = "Month",  
        # y-axis title  
        y = "Georgian Lari") +  
  # telling R not to break the x-axis  
  scale_x_continuous(breaks = 201901:201912) +  
  # centering plot title  
  theme(plot.title = element_text(hjust = 0.5))
```

# Bar plots // ბარის ნაკვეთები

Now this looks better:



# Bar plots // ბარის ნაკვეთები

## Exercise 1c: save your plot

Now that your plot looks good, you can save it into an output with `ggsave()`

1. Use this code to save your plot:

```
ggsave("vat_liability_small_2019.png",
       width = 20,
       height = 10,
       units = "cm")
```

# Bar plots // ბარის ნაკვეთები

- `ggsave()` by default saves the last plot you produced
- The first argument in `ggsave()` is the name of the file we save the plot into. We can also use file paths here
- The rest are optional arguments that define the dimensions of the image you export, it's better to define them so the image has the correct proportions and text size

Name	Date modified	Type	Size
1-introduction-to-r_cache	9/19/2023 3:50 AM	File folder	
2-data-wrangling_cache	9/19/2023 3:46 PM	File folder	
3-descriptive-statistics_cache	9/20/2023 7:01 PM	File folder	
4-data-visualization_cache	9/20/2023 11:23 PM	File folder	
4-data-visualization_files	9/21/2023 12:02 AM	File folder	
data	9/20/2023 11:51 PM	File folder	
img	9/20/2023 11:10 PM	File folder	
libs	9/21/2023 4:17 AM	File folder	
1-introduction-to-r.pdf	9/20/2023 9:47 AM	Adobe Acrobat Docu...	4,090 KB
2-data-wrangling.pdf	9/20/2023 9:48 AM	Adobe Acrobat Docu...	5,163 KB
1-introduction-to-r.html	9/20/2023 1:34 AM	Chrome HTML Docu...	30 KB
2-data-wrangling.html	9/20/2023 5:44 AM	Chrome HTML Docu...	33 KB
3-descriptive-statistics.html	9/20/2023 10:37 PM	Chrome HTML Docu...	27 KB
4-data-visualization.html	9/21/2023 4:17 AM	Chrome HTML Docu...	26 KB
df_tbilisi_50.csv	9/20/2023 5:12 AM	Microsoft Excel Com...	2 KB
total_income.csv	9/20/2023 5:11 AM	Microsoft Excel Com...	1 KB
vat_liability_small_2019.png	9/21/2023 4:18 AM	PNG File	63 KB
exercises-session1.R	9/20/2023 1:51 PM	R File	1 KB
exercises-session2.R	9/20/2023 11:50 PM	R File	1 KB
exercises-session4.R	9/21/2023 3:18 AM	R File	3 KB
202309.Rproj	9/20/2023 7:34 PM	R Project	1 KB
.Rhistory	9/20/2023 1:51 PM	RHISTORY File	3 KB
1-introduction-to-r.Rmd	9/20/2023 1:34 AM	RMD File	24 KB
2-data-wrangling.Rmd	9/20/2023 5:44 AM	RMD File	27 KB
3-descriptive-statistics.Rmd	9/20/2023 11:17 PM	RMD File	21 KB
4-data-visualization.Rmd	9/21/2023 4:16 AM	RMD File	24 KB

# Bar plots // ბარის ნაკვეთები

## A note about syntax

- Data visualization usually requires several iterations to add new elements to your initial code and improve your plot
- `ggplot2` adds new elements to a visualization with the symbol `+`
- More customization means that your code can easily become quite long. Using spaces and line breaks helps for clarity, but **there is just no way around it**
- In programming this is known as **heavy syntax**

```
# Exercise 1:  
ggplot(small_business_2019_all) +  
  aes(x = taxperiod,  
       y = vat_liability) +  
  geom_col() +  
  labs(title = "Total VAT liability of small businesses in 2019 by month")
```

```
# Exercise 2  
ggplot(small_business_2019_all) +  
  aes(x = as.factor(taxperiod),  
       y = vat_liability) +  
  geom_col() +  
  labs(title = "Total VAT liability of small businesses in 2019 by month",  
        x = "Month",  
        y = "Georgian Lari") +  
  theme(plot.title = element_text(hjust = 0.5))
```

# Line plots // ხაზის ნაკვეთები

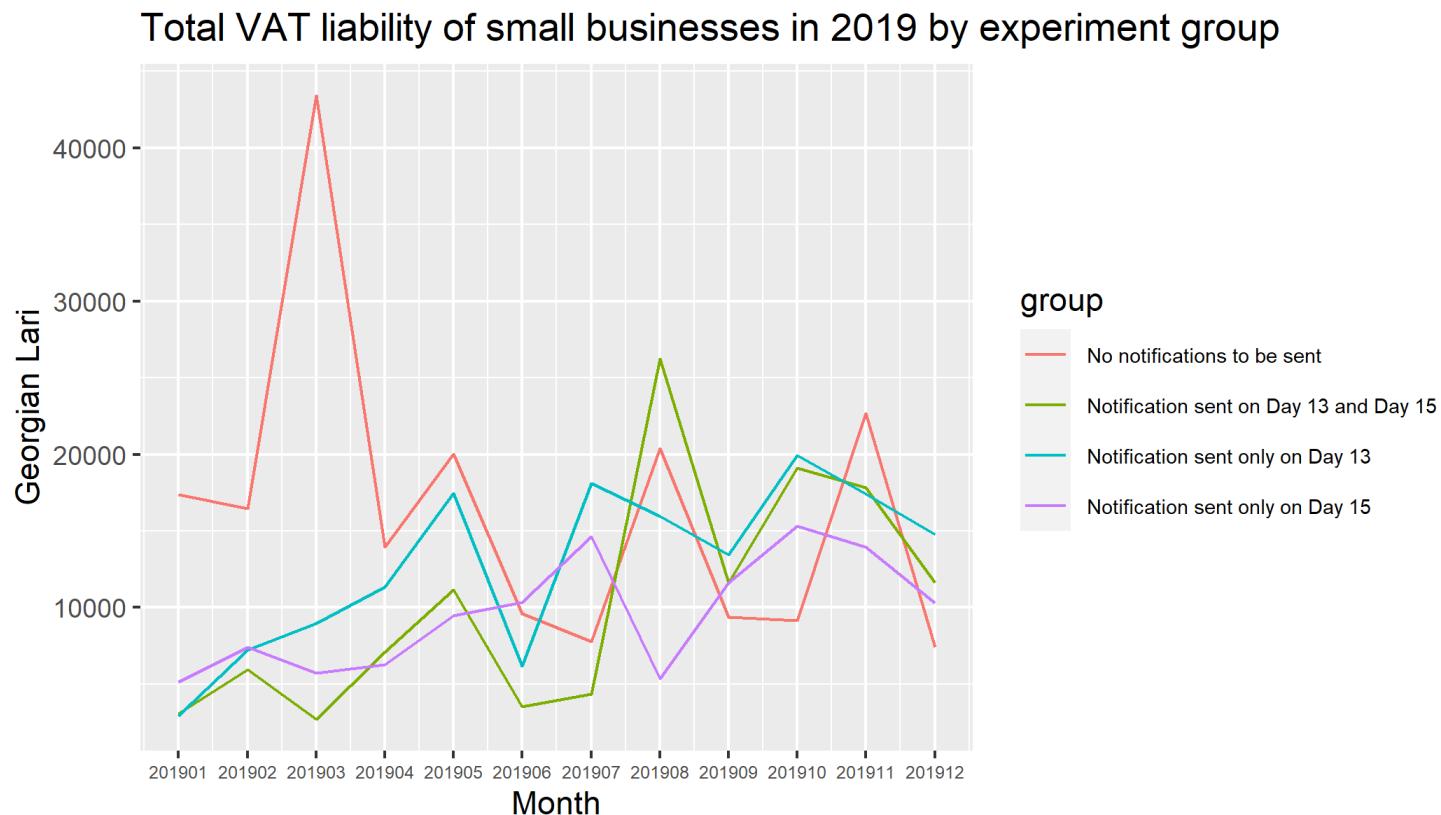
---

# Line plots // ხაზის ხავერდები

- In data visualization, we call **encodings** to the geometry selected to represent the data visually
- In the previous examples, we did that when we used `geom_col()`. This tells R that our encoding to represent the data is **bars** (also called columns)
- `ggplot2` has several different options for encodings in data visualization, for example: **lines**
- The encoding for lines in `ggplot2` is `geom_lines()`

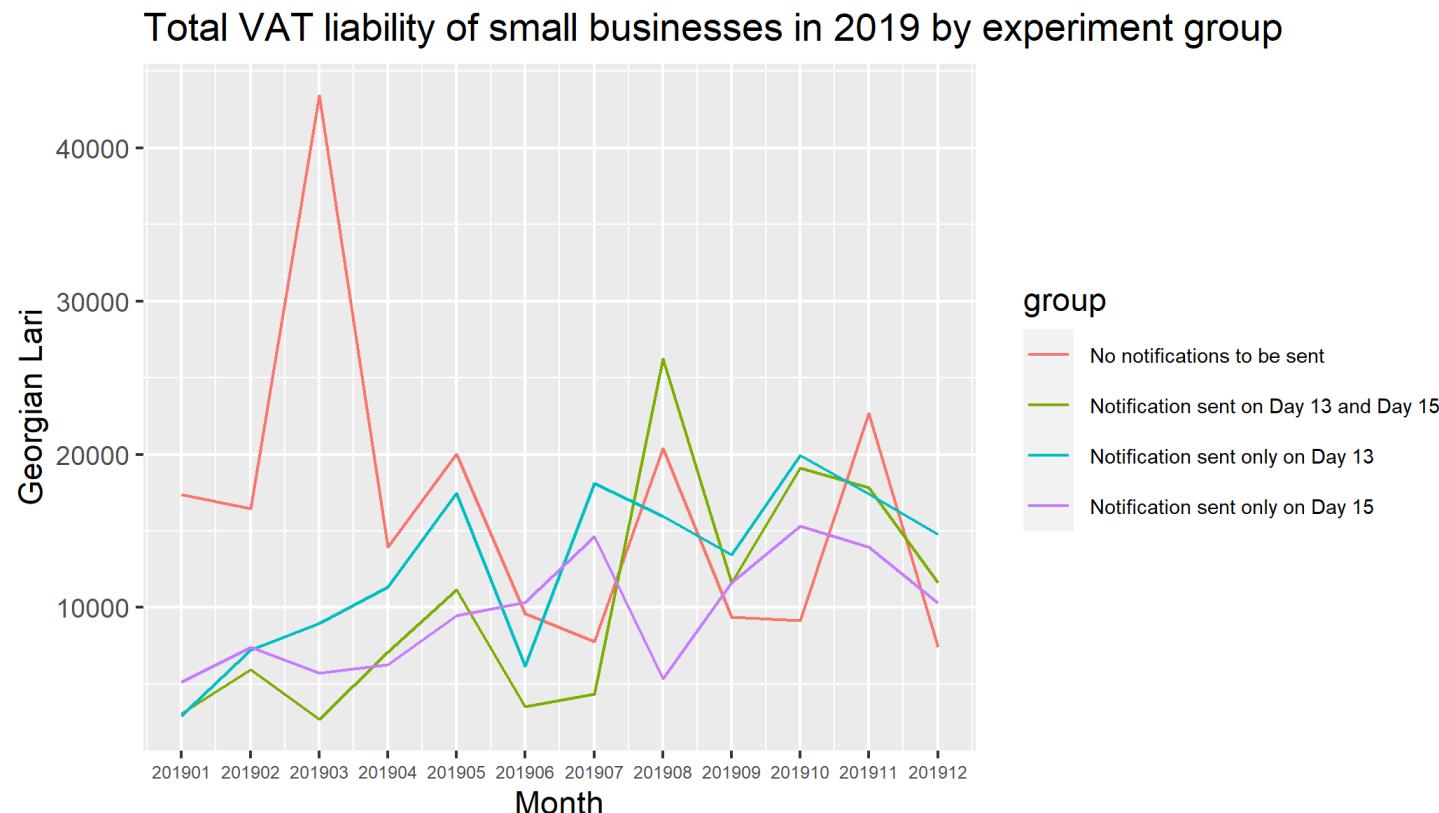
# Line plots // ხაზის ნაკვეთები

- Line plots are a nice option to encode numeric values and include different categories of a second variable at the same time



# Line plots // ხაზის ნაკვეთები

- However, they also usually require some additional data wrangling compared to bar plots: data should be collapsed at the level specified in the x-axis and grouping variable
- This is going to be clearer in the next exercise. The plot we'll produce is below



# Line plots // ხაზის ხავერდები

## Exercise 2a: Collapse your data at the month-group level

Use the following code to create a dataframe collapsed at the month-group level and calculate the total VAT liability for each group in each month

```
df_group_month <- small_business_2019_all %>%
  select(group, taxperiod, vat_liability) %>%
  group_by(group, taxperiod) %>%
  summarize(total = sum(vat_liability))
```

# Line plots // ხაზის ხავერდები

The result will look like this, you can explore it with

```
View(df_group_month).
```

	group	taxperiod	total
1	No notifications to be sent	201901	17393.953
2	No notifications to be sent	201902	16441.432
3	No notifications to be sent	201903	43434.875
4	No notifications to be sent	201904	13957.029
5	No notifications to be sent	201905	20015.730
6	No notifications to be sent	201906	9624.463
7	No notifications to be sent	201907	7783.283
8	No notifications to be sent	201908	20388.409
9	No notifications to be sent	201909	9382.099
10	No notifications to be sent	201910	9147.933
11	No notifications to be sent	201911	22679.408
12	No notifications to be sent	201912	7417.622
13	Notification sent on Day 13 and Day 15	201901	3054.195
14	Notification sent on Day 13 and Day 15	201902	5932.728
15	Notification sent on Day 13 and Day 15	201903	2697.588
16	Notification sent on Day 13 and Day 15	201904	7070.751
17	Notification sent on Day 13 and Day 15	201905	11142.686
18	Notification sent on Day 13 and Day 15	201906	3535.528
19	Notification sent on Day 13 and Day 15	201907	4350.107

Showing 1 to 20 of 48 entries, 3 total columns

# Line plots // ხაზის ნაკვეთები

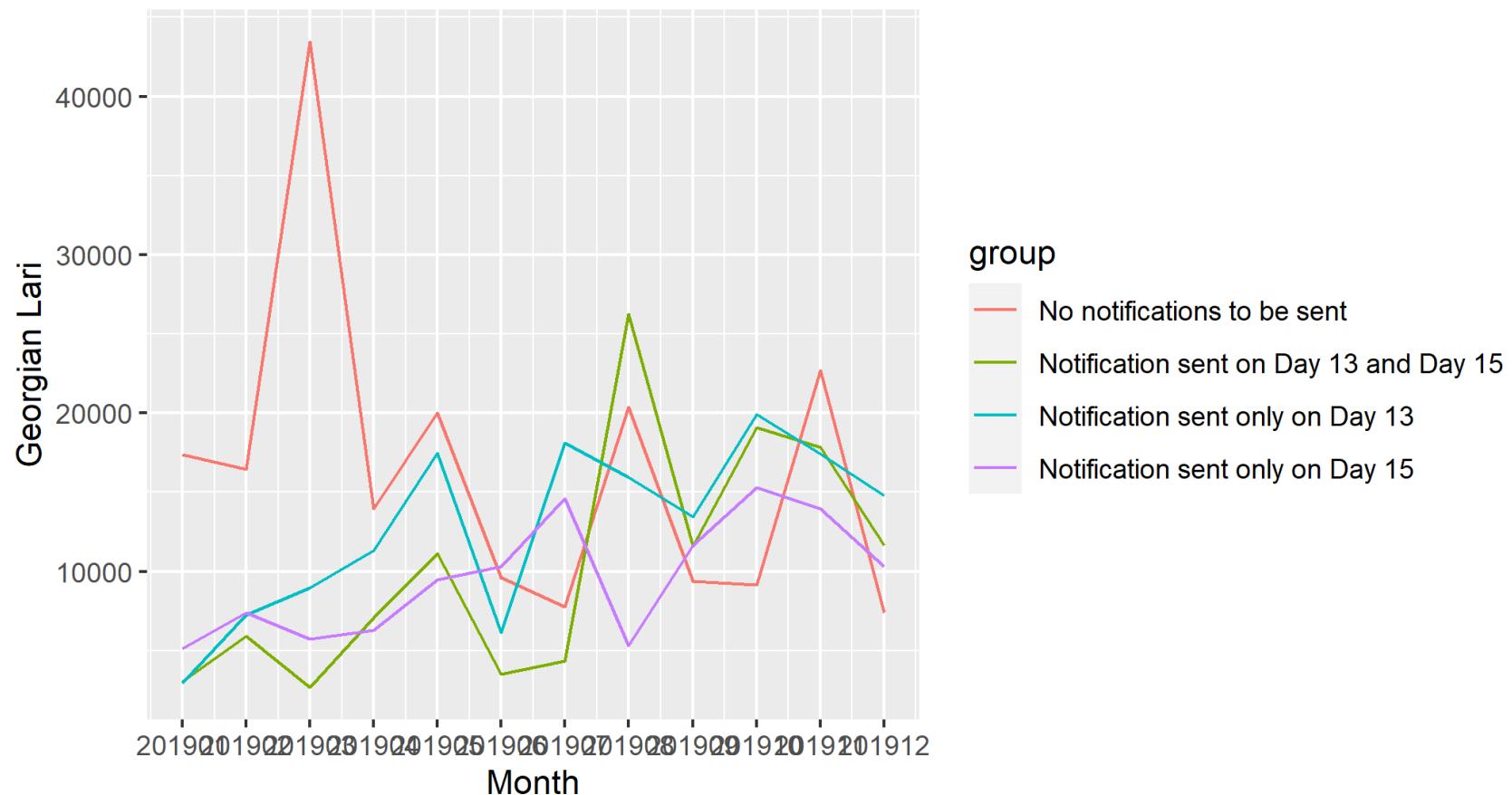
Exercise 2b: Create a line plot of VAT liability by month and group

```
ggplot(df_group_month) +  
  aes(x = taxperiod,  
       y = total) +  
  geom_line(aes(color = group)) +  
  labs(title = "Total VAT liability of small businesses in 2019 by experiment group",  
        x = "Month",  
        y = "Georgian Lari") +  
  scale_x_continuous(breaks = 201901:201912) +  
  theme(plot.title = element_text(hjust = 0.5))
```

# Line plots // ხაზის ნაკვეთები

Your result possibly looks like this:

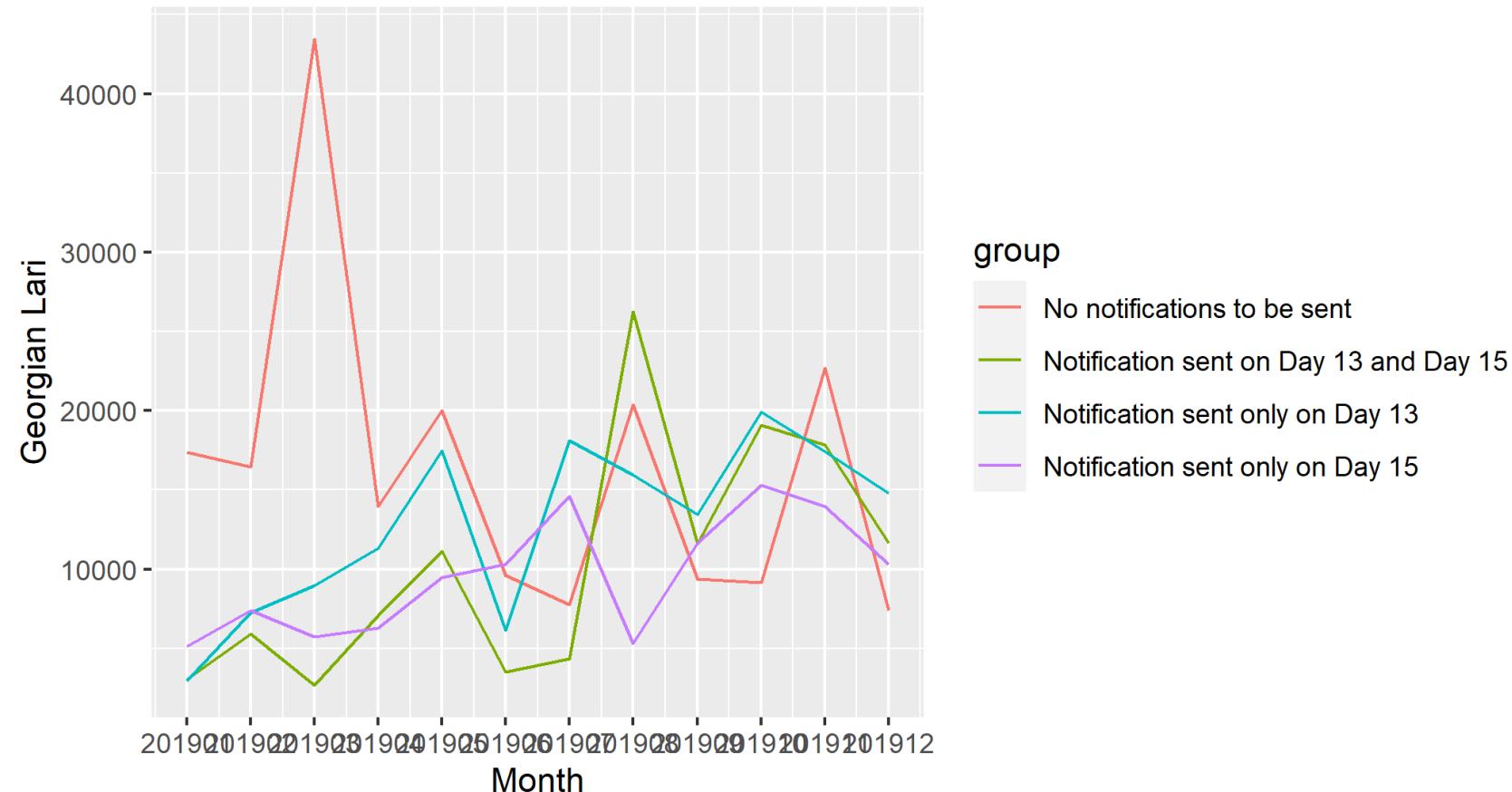
Actual VAT liability of small businesses in 2019 by experiment group



# Line plots // სახის ნაკვეთები

Something looks off, doesn't it? We need to **make the legend labels and x-axis texts smaller** for them without overlapping each other and to **remove the centering of the title** so it's not cropped

Annual VAT liability of small businesses in 2019 by experiment group



# Line plots // ხაზის ხარჯები

## Exercise 2c: Continue improving your plot

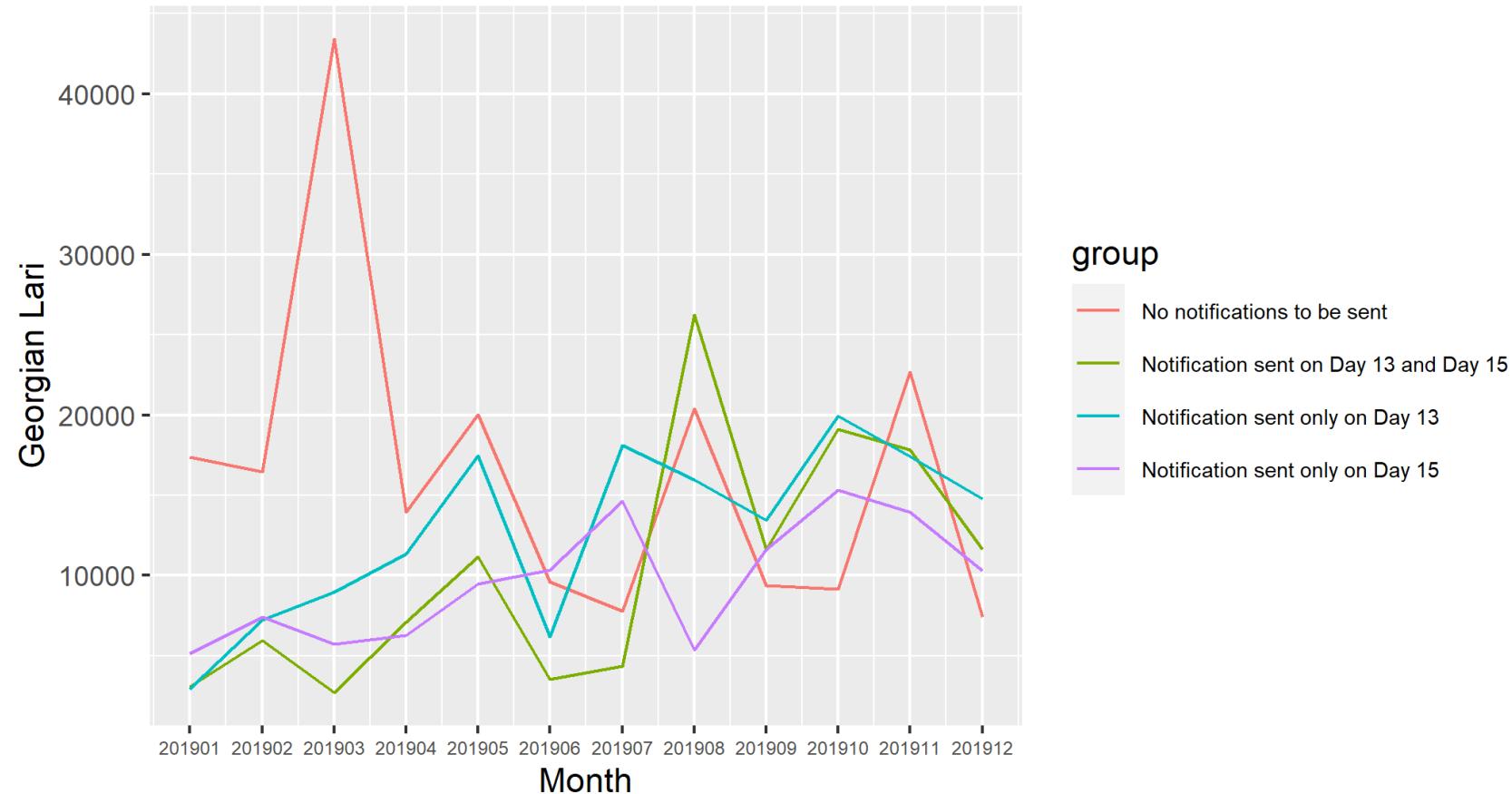
1. Add the argument `legend.text=element_text(size=7)` inside `theme()` to decrease the legend text size
2. Add the argument `axis.text.x=element_text(size=6)` inside `theme()` to decrease the x-axis text size
  - note that both arguments need to be separated by a comma
3. Remove `plot.title = element_text(hjust = 0.5)` from `theme()` to remove the centering of the plot title

The result should be this:

```
ggplot(df_group_month) +
  aes(x = taxperiod,
      y = total) +
  geom_line(aes(color = group)) +
  labs(title = "Total VAT liability of small businesses in 2019 by experiment group",
       x = "Month",
       y = "Georgian Lari") +
  scale_x_continuous(breaks = 201901:201912) +
  theme(legend.text = element_text(size = 7), # don't forget the comma!
        axis.text.x=element_text(size=6))
```

# Line plots // ხაზის ნაკვეთები

Total VAT liability of small businesses in 2019 by experiment group



# Line plots // ხაზის ნაკვეთები

## Exercise 2d: save your plot

Use this code to save your plot:

```
ggsave("vat_liability_small_2019_by_group.png",
       width = 20,
       height = 10,
       units = "cm")
```

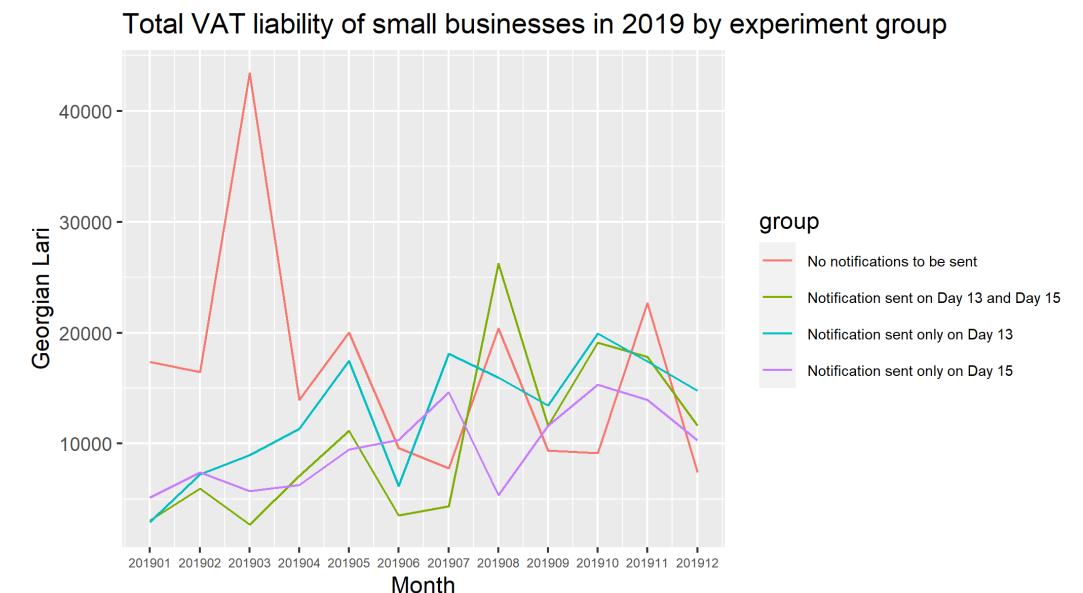
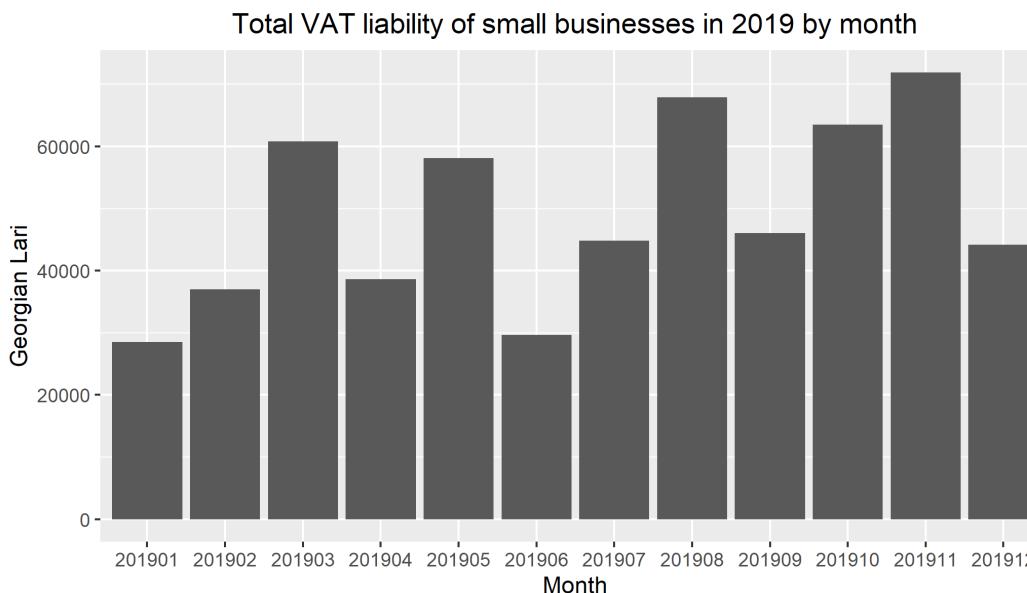
# Line plots // ხაზის ნაკვეთები

Name	Date modified	Type	Size
2-data-wrangling_cache	9/19/2023 5:40 PM	File folder	
3-descriptive-statistics_cache	9/20/2023 7:01 PM	File folder	
4-data-visualization_cache	9/20/2023 11:23 PM	File folder	
4-data-visualization_files	9/21/2023 12:02 AM	File folder	
data	9/20/2023 11:51 PM	File folder	
img	9/20/2023 11:10 PM	File folder	
libs	9/21/2023 4:17 AM	File folder	
1-introduction-to-r.pdf	9/20/2023 9:47 AM	Adobe Acrobat Docu...	4,090 KB
2-data-wrangling.pdf	9/20/2023 9:48 AM	Adobe Acrobat Docu...	5,163 KB
1-introduction-to-r.html	9/20/2023 1:34 AM	Chrome HTML Docu...	30 KB
2-data-wrangling.html	9/20/2023 5:44 AM	Chrome HTML Docu...	33 KB
3-descriptive-statistics.html	9/20/2023 10:37 PM	Chrome HTML Docu...	27 KB
4-data-visualization.html	9/21/2023 4:17 AM	Chrome HTML Docu...	26 KB
df_tbilisi_50.csv	9/20/2023 5:12 AM	Microsoft Excel Com...	2 KB
total_income.csv	9/20/2023 5:11 AM	Microsoft Excel Com...	1 KB
vat_liability_small_2019.png	9/21/2023 4:39 AM	PNG File	79 KB
vat_liability_small_2019_by_group.png	9/21/2023 4:42 AM	PNG File	192 KB
exercises-session1.R	9/20/2023 1:51 PM	R File	1 KB
exercises-session2.R	9/20/2023 11:50 PM	R File	1 KB
exercises-session4.R	9/21/2023 4:22 AM	R File	3 KB
202309.Rproj	9/20/2023 7:34 PM	R Project	1 KB
.Rhistory	9/20/2023 1:51 PM	RHISTORY File	3 KB
1-introduction-to-r.Rmd	9/20/2023 1:34 AM	RMD File	24 KB
2-data-wrangling.Rmd	9/20/2023 5:44 AM	RMD File	27 KB
3-descriptive-statistics.Rmd	9/20/2023 11:17 PM	RMD File	21 KB
4-data-visualization.Rmd	9/21/2023 4:21 AM	RMD File	25 KB

# Line plots // ბაზისური ნაკვეთები

Choosing the right encoding for your data can be tricky. It depends on what you want to show in your plot and how much information you want it to show

- Bar plots show less information than line plots in general, but they are good for cases when you have only one numeric variable and one categorical variable to show
- Line plots can show more information and includes multiple groups to add an additional dimension to your data, but they are not visually appealing when your data varies a lot from one category to another

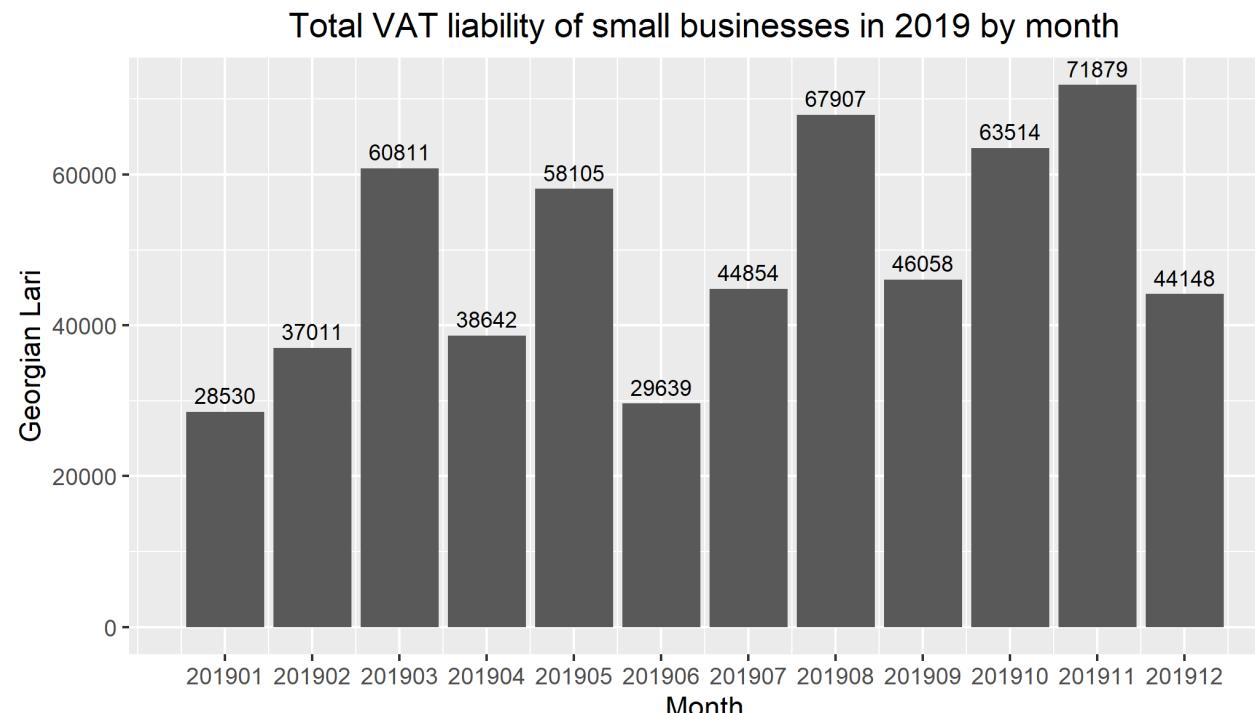


# Text encodings // ტექსტური დაშიგვნები

---

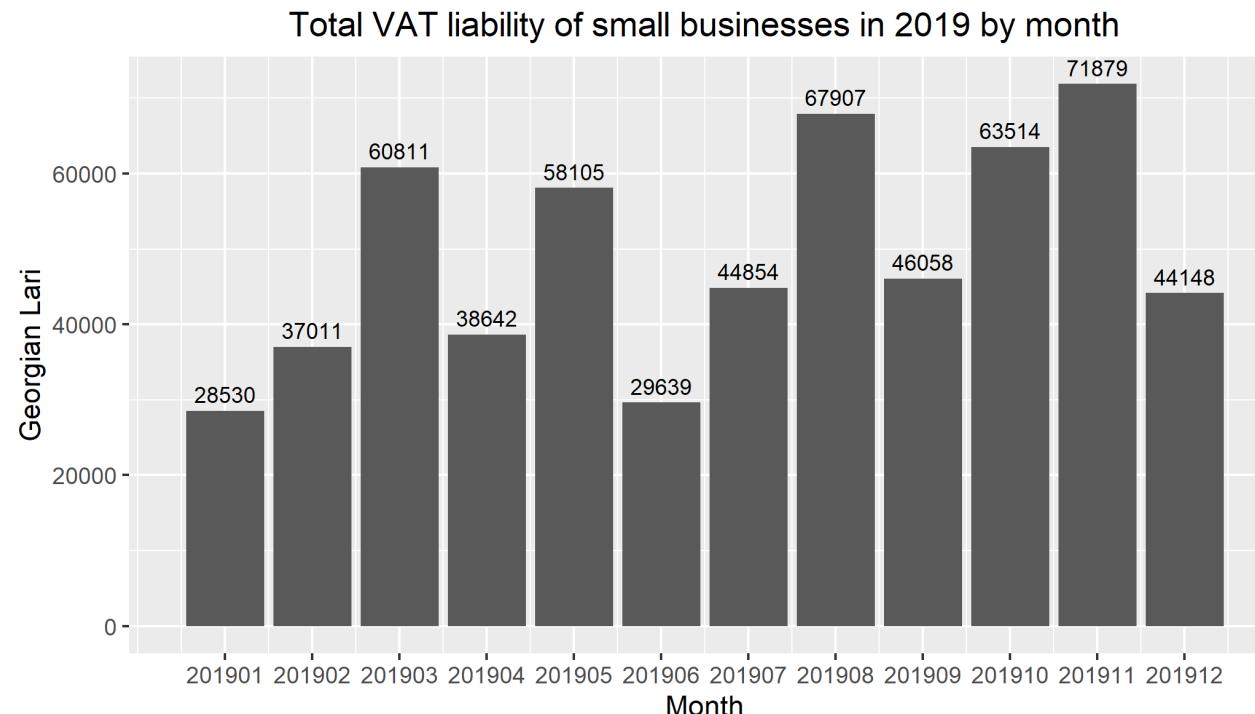
# Text encodings // ტექსტური დამიგვარებელი

- Geometric shapes such as bars and plots are not the only way to encode data in plots
- We can also use text directly to represent information in the plot as in the example below. This is called **text encodings**
- Text encodings can be combined with geometric encodings to highlight information or provide important additional details to your visualization



# Text encodings // ტექსტური დამიფრებელი

- Using text encodings in a bar plot is nice
- However, it also requires additional data wrangling: the data needs to be collapsed at the same level of the x-axis to be able to add encodings
- We'll see this in the next exercise



# Text encodings // የጊዜያዊ ወሳኔዎች

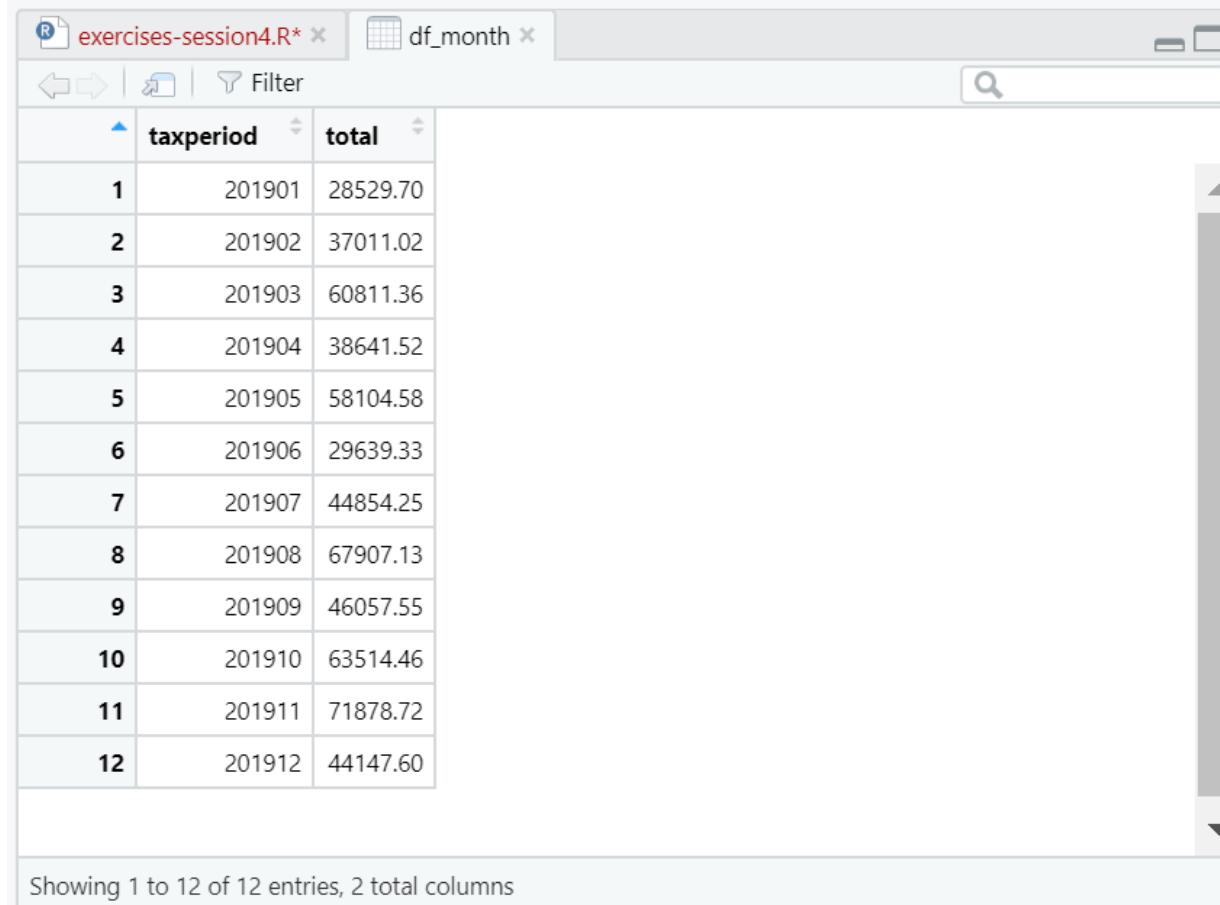
## Exercise 3a: Collapse your data at the month level

This is similar to what we did for exercise 2a, except that this time the collapsing is at month level instead of the month-group level. Use the code below to store the collapsed dataframe in `month_df`

```
df_month <- small_business_2019_all %>%
  select(taxperiod, vat_liability) %>%
  group_by(taxperiod) %>%
  summarize(total = sum(vat_liability))
```

# Text encodings // የጊዢዎች ወጪዎች

This is the result, you can use `View(df_month)` to display it



The screenshot shows the RStudio interface with two tabs open: "exercises-session4.R\*" and "df\_month". The "df\_month" tab is active and displays a data frame with 12 rows and 2 columns. The columns are labeled "taxperiod" and "total". The data is as follows:

	taxperiod	total
1	201901	28529.70
2	201902	37011.02
3	201903	60811.36
4	201904	38641.52
5	201905	58104.58
6	201906	29639.33
7	201907	44854.25
8	201908	67907.13
9	201909	46057.55
10	201910	63514.46
11	201911	71878.72
12	201912	44147.60

Showing 1 to 12 of 12 entries, 2 total columns

# Text encodings // ტექსტური დამზადებები

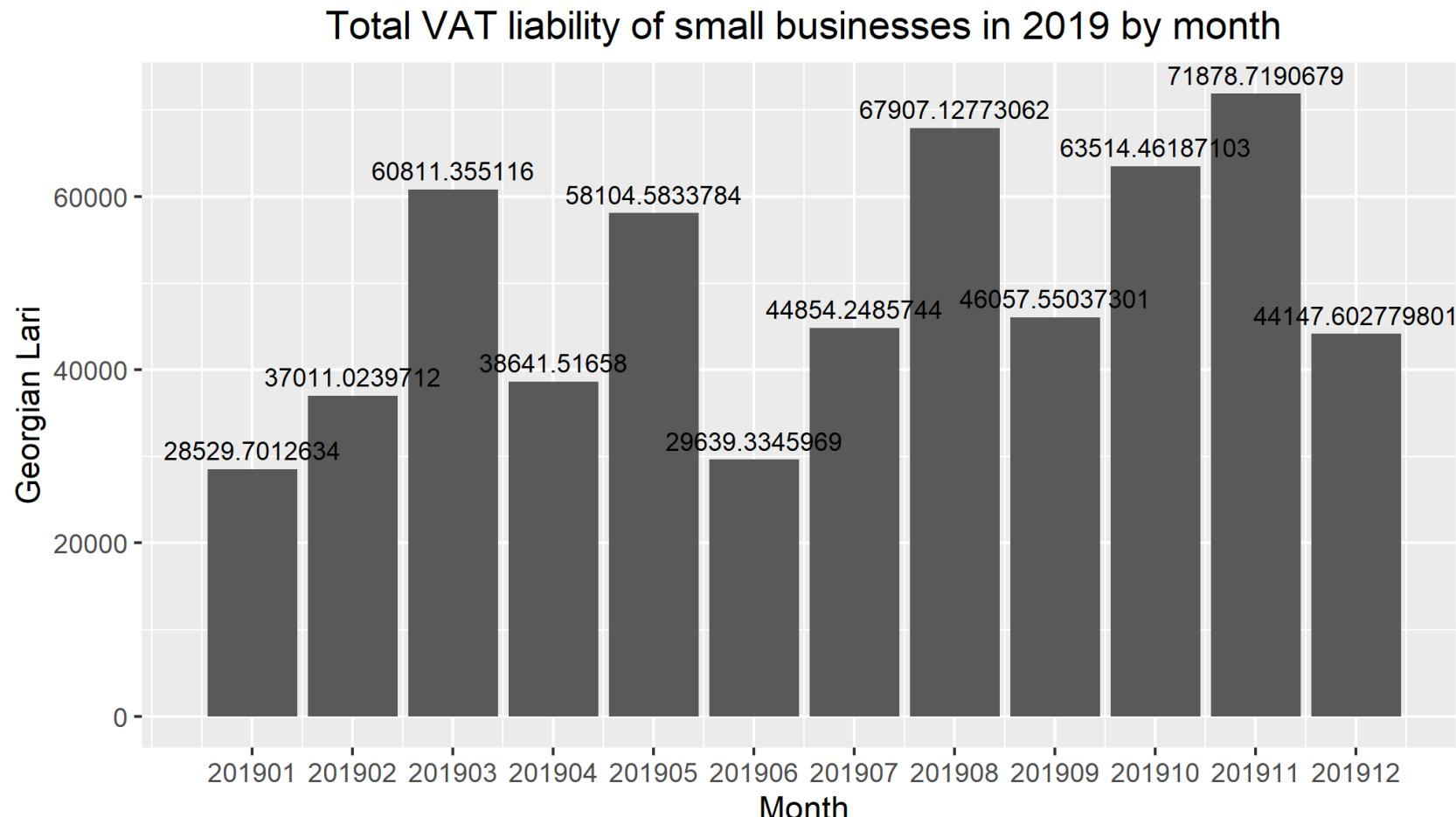
## Exercise 3b: Add encodings to your bar plot

Use the collapsed dataframe `df_month` and the former code from exercise 1 to add encodings to your bar plot. The result is the code below:

```
ggplot(df_month) +  
  aes(x = taxperiod,  
      y = total) +  
  geom_col() +  
  # Note that the text encodings are added here  
  geom_text(aes(label = total),  
            position = position_dodge(width = 1),  
            vjust = -0.5,  
            size = 3) +  
  labs(title = "Total VAT liability of small businesses in 2019 by month",  
        x = "Month",  
        y = "Georgian Lari") +  
  scale_x_continuous(breaks = 201901:201912) +  
  theme(plot.title = element_text(hjust = 0.5))
```

# Text encodings // ტექსტური დამიფრებელი

This result is nice but the text encodings have several decimal places. We can improve it by rounding `total` with the function `round()`



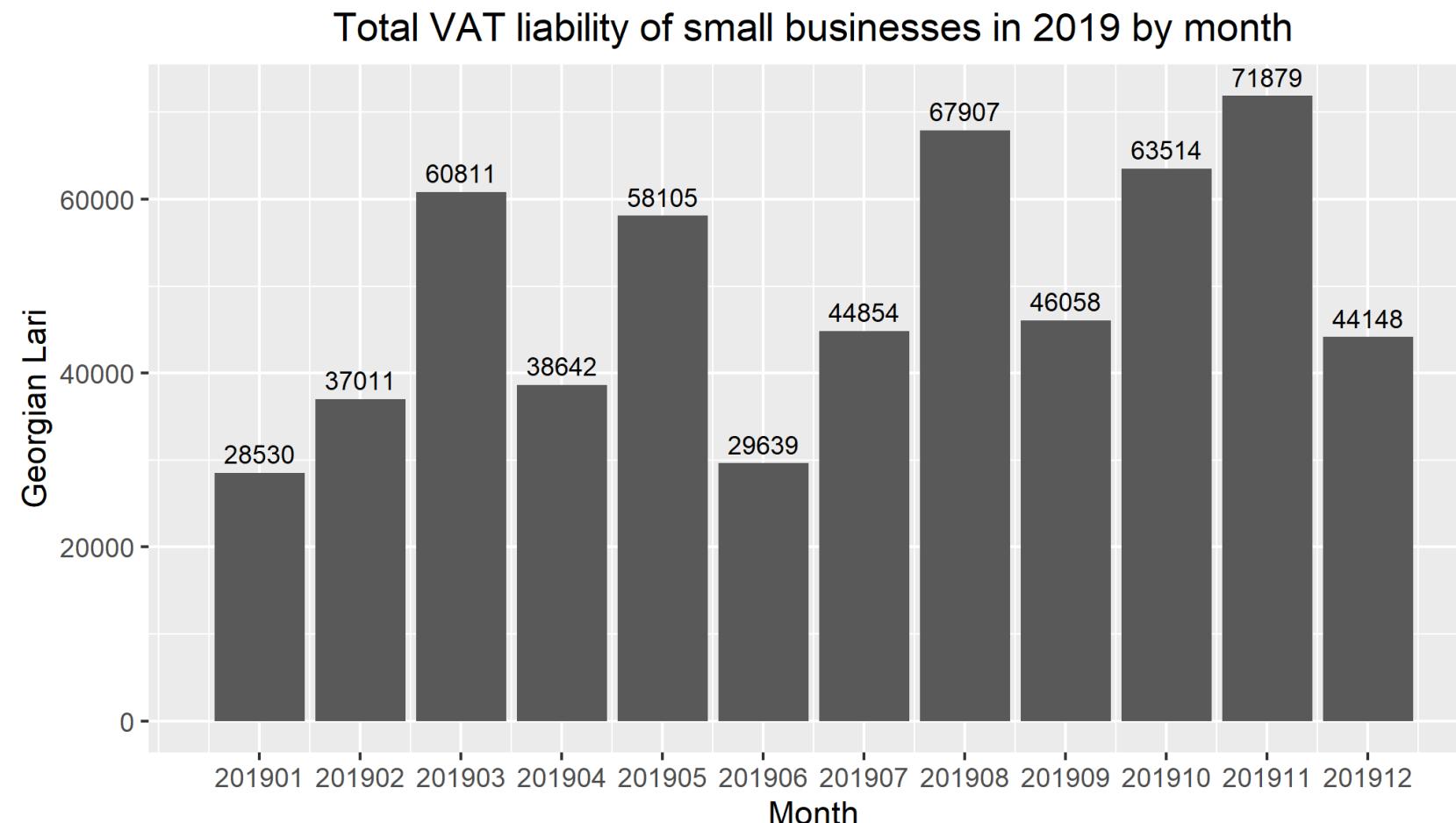
# Text encodings // ტექსტური დამზადებები

## Exercise 3c: Improve your plot once again

Replace `total` with `round(total)` in `geom_text()`. The result should be:

```
ggplot(df_month) +  
  aes(x = taxperiod,  
      y = total) +  
  geom_col() +  
  geom_text(aes(label = round(total)), # <--- the change goes here  
            position = position_dodge(width = 1),  
            vjust = -0.5,  
            size = 3) +  
  labs(title = "Total VAT liability of small businesses in 2019 by month",  
        x = "Month",  
        y = "Georgian Lari") +  
  scale_x_continuous(breaks = 201901:201912) +  
  theme(plot.title = element_text(hjust = 0.5))
```

# Text encodings // ტექსტების დამზადება



# Text encodings // የጊዢዎች ፍልዕስ ጥሩ

Don't forget to save your plot!

Use this code to save this last plot:

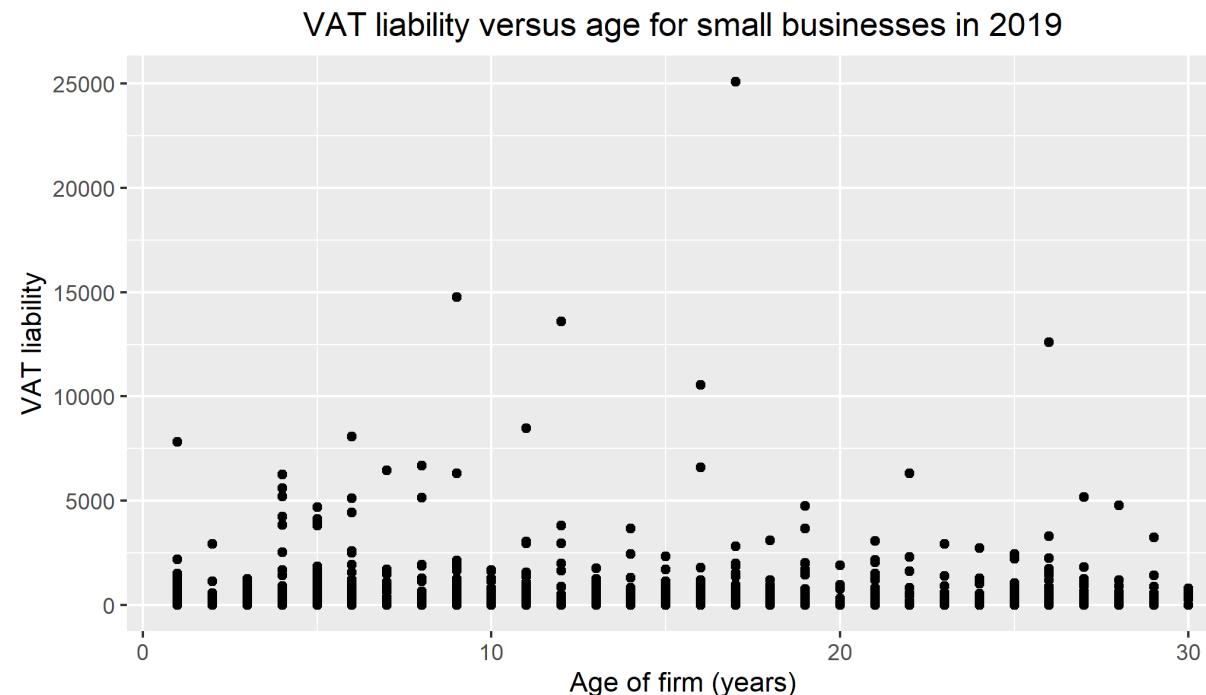
```
ggsave("vat_liability_small_2019_text.png",
       width = 20,
       height = 10,
       units = "cm")
```

# Scatter plots // გაფანტიზებული ნაკვეთები

---

# Scatter plots // გაფანტავს ნაკვეთებს

- We're going to explore one more type of encoding for data visualizations: **scatter plots**
- Scatter plots are useful when you have two continuous numeric variables and want to show that there might be a correlation between them or visualize outliers (values that stand out from the rest because they are very extreme)
- We use the encoding `geom_line()` for scatterplots



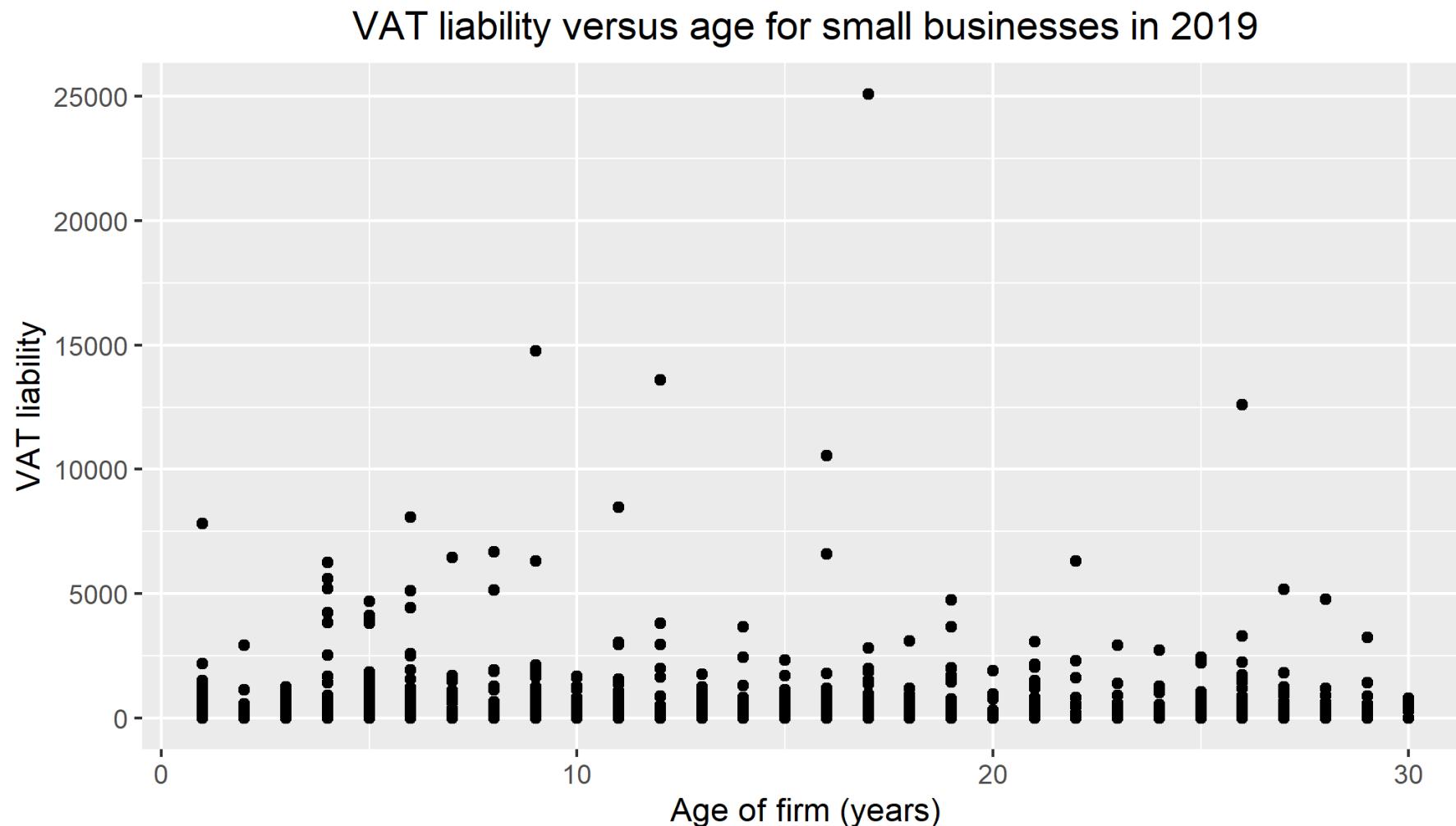
# Scatter plots // გაფანტავს ნაკვეთებს

## Exercise 4: Create a scatter plot

Use the following code to reproduce the scatter plot of the last slide:

```
ggplot(small_business_2019_all) +  
  aes(x = age,  
      y = vat_liability) +  
  geom_point() +  
  labs(title = "VAT liability versus age for small businesses in 2019",  
       x = "Age of firm (years)",  
       y = "VAT liability") +  
  theme(plot.title = element_text(hjust = 0.5))
```

# Scatter plots // გაფანტავს ნაკვეთებს



# Scatter plots // გაფანტავს ნაკვეთებს

Lastly, remember to save your scatter plot with:

```
ggsave("scatter_age_vat.png",
       width = 20,
       height = 10,
       units = "cm"))
```

Wrapping up // მიზანება

---

# Wrapping up // დევილოვა

## Other encodings in `ggplot2`

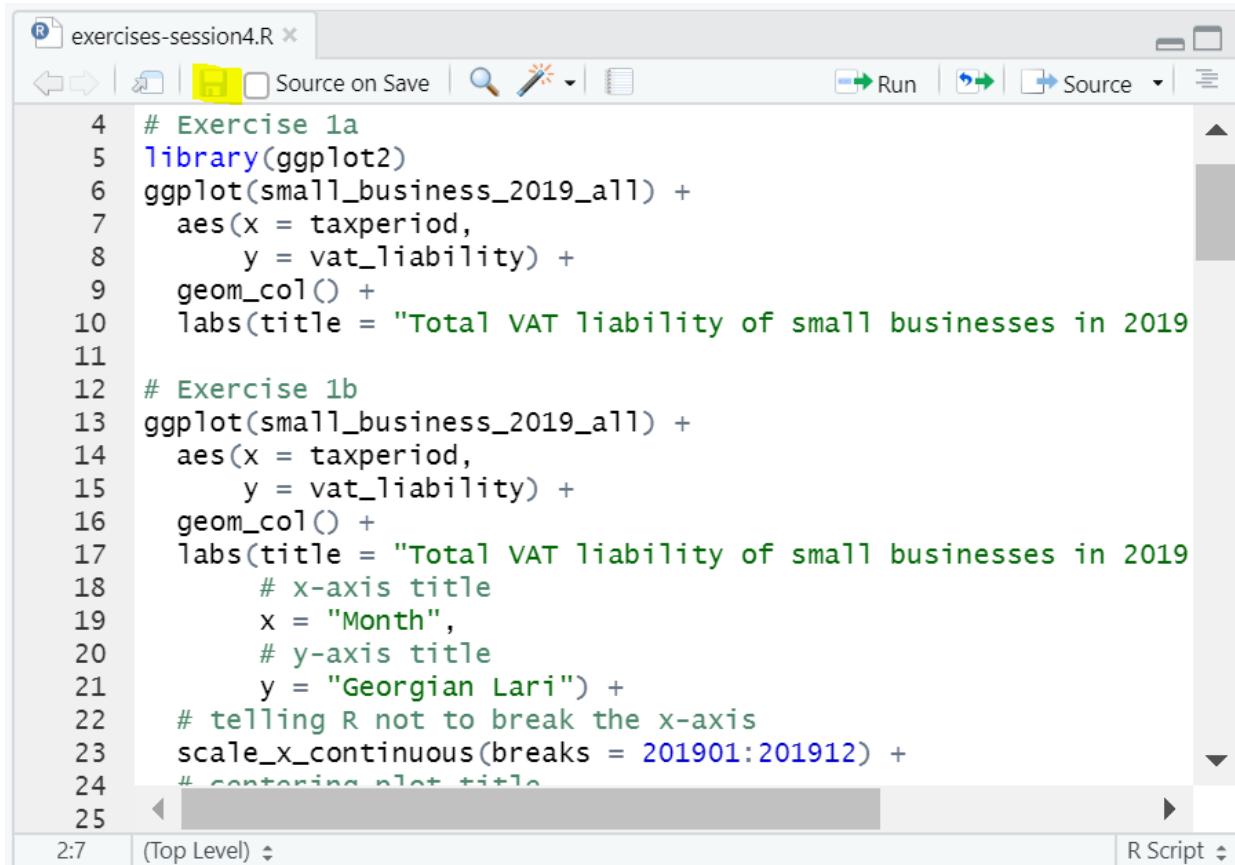
This table lists several of the most popular encoding types in `ggplot2`

Encoding	Function in <code>ggplot2</code>
Bars	<code>geom_col()</code>
Lines	<code>geom_line()</code>
Points (scatterplot)	<code>geom_point()</code>
Area	<code>geom_area()</code>
Histogram	<code>geom_histogram()</code>
Floating labels (texts)	<code>geom_text()</code>
Box plot	<code>geom_boxplot()</code>
Pie chart	<code>geom_bar() + coord_polar()</code>
Smoothed line	<code>geom_smooth()</code>

# Wrapping up // გეგუთვა

## Save your code!

Click on the floppy disk to save your code in a location that you will remember.



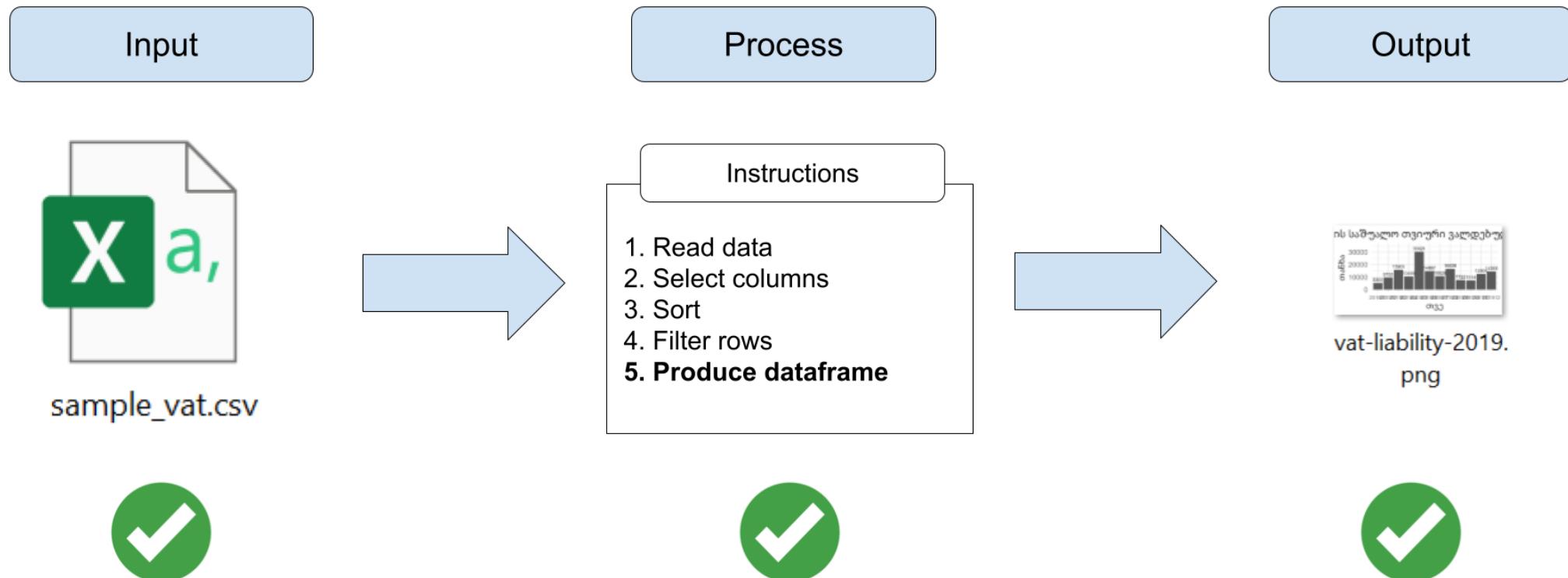
The screenshot shows an RStudio interface with the following details:

- Title Bar:** The title bar displays "exercises-session4.R x".
- Toolbar:** The toolbar includes standard icons for file operations (New, Open, Save, Print), search, and run.
- Code Editor:** The main area contains R code for generating two ggplot2 charts. The code uses the `ggplot2` library to plot VAT liability over time for small businesses in 2019.

```
4 # Exercise 1a
5 library(ggplot2)
6 ggplot(small_business_2019_all) +
7   aes(x = taxperiod,
8       y = vat_liability) +
9   geom_col() +
10  labs(title = "Total VAT liability of small businesses in 2019")
11
12 # Exercise 1b
13 ggplot(small_business_2019_all) +
14   aes(x = taxperiod,
15       y = vat_liability) +
16   geom_col() +
17   labs(title = "Total VAT liability of small businesses in 2019",
18        # x-axis title
19        x = "Month",
20        # y-axis title
21        y = "Georgian Lari") +
22        # telling R not to break the x-axis
23        scale_x_continuous(breaks = 201901:201912) +
24        # centering plot title
```
- Status Bar:** The bottom status bar shows "2:7 (Top Level)" and "R Script".

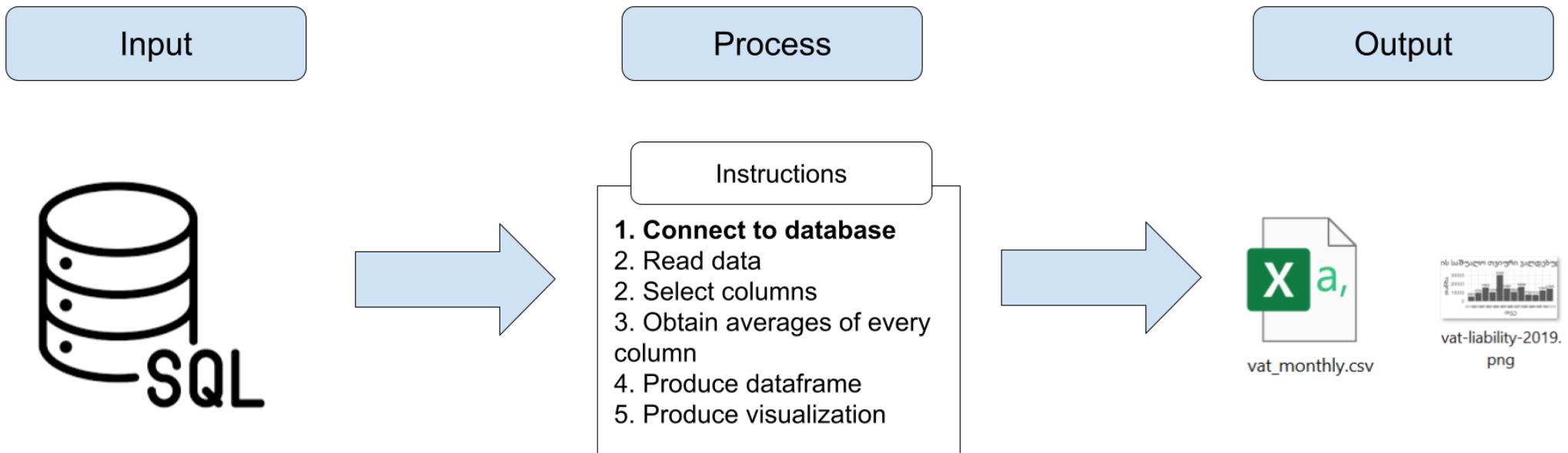
# Wrapping up // მიზანთვა

## Data work pipeline



# Wrapping up // მიმკვეთვა

## Looking ahead



# Wrapping up // მიმკვეთვა

## Looking ahead

- Connecting R with SQL databases. Some libraries:

- `dbConnect`
- `dbplyr`
- `DBI`

- More on data wrangling. More libraries:

- `tidyverse`
- `janitor`

- More on data visualization

Thanks! // əs̪θɔːmθəs! // ¡Gracias! // Obrigado!

---