

SESSION 1: DATA FRAMES

DR. SOFIA GIL-CLAVEL

- ❖ R and RStudio
- ❖ Data types
- ❖ Data frames

0. R & RSTUDIO

What is R?
What is RStudio?

WHAT IS R?

R is a dialect of S.

S is a language that was developed by John Chambers and others at the old Bell Telephone Laboratories, originally part of AT&T Corp. S was initiated in 1976 as an internal statistical analysis environment.

THE PHILOSOPHY OF S

The S language had its roots in data analysis, and did not come from a traditional programming language background. **Its inventors were focused on figuring out how to make data analysis easier**, first for themselves, and then eventually for others. [...] **They built a language that would be suitable for interactive data analysis** (more command-line based) as well as for writing longer programs (more traditional programming language-like).

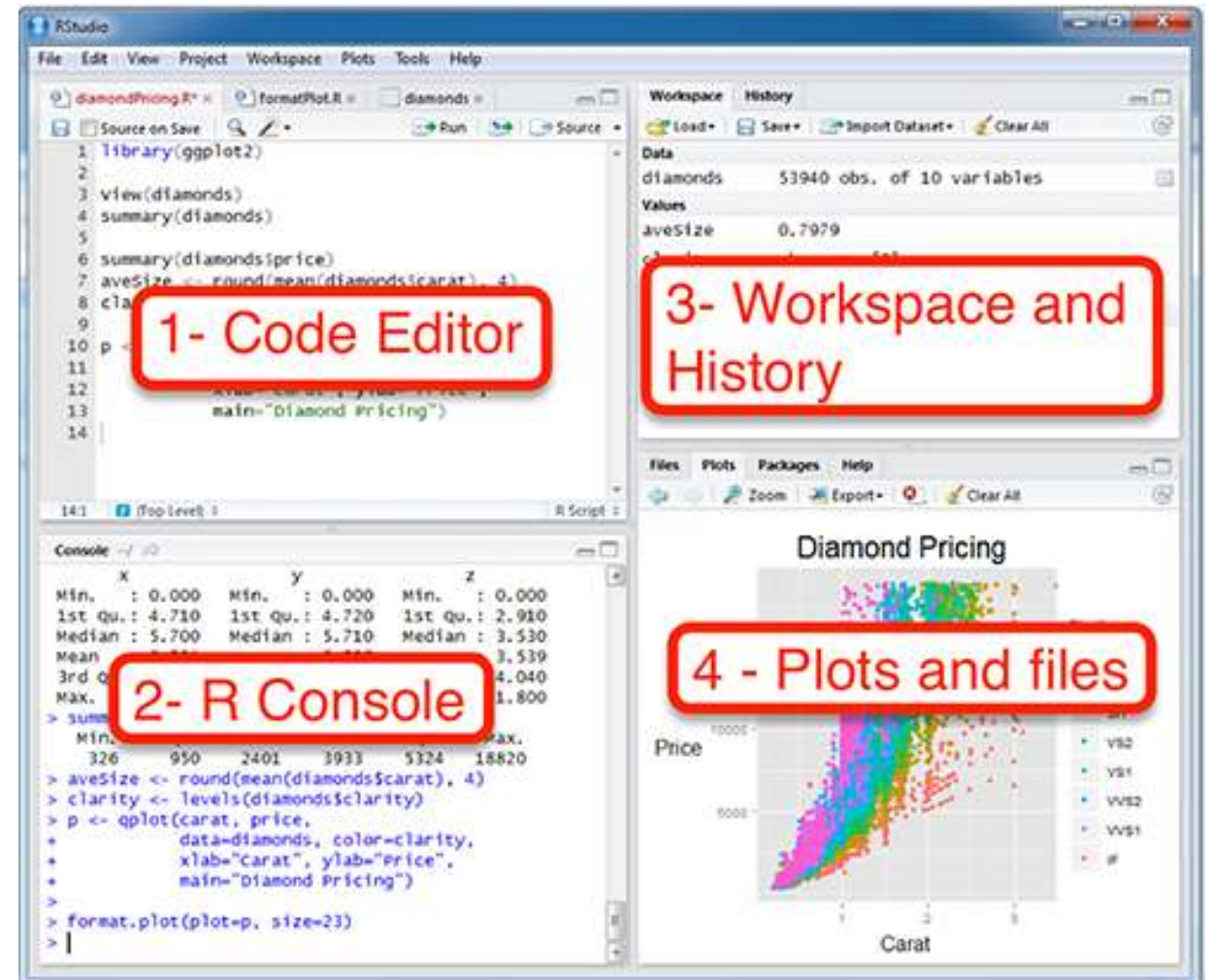
Source: Peng, Roger D. *R Programming for Data Science*. Accessed May 5, 2024. <https://bookdown.org/rdpeng/rprogdatascience/>.

WHAT IS RSTUDIO?

RStudio is an integrated development environment for R.

"RStudio." In Wikipedia, May 1, 2024.

<https://en.wikipedia.org/w/index.php?title=RStudio&oldid=1221670801>



HOMework 0

- Download R & RStudio

LET'S OPEN RSTUDIO!



1. A SHORT INTRODUCTION TO R DATA TYPES

Primitive Data Types
Vectors

❖ DATA TYPES

A data type is a set of values and operations associated with those values.

Primitive: These are the simplest types of data, because they are not constructed from other types and are unique entities that cannot be decomposed into others. These data types are defined by a set of values and a set of operations that act on those values.

- Integer
- Floating/Double
- Character and String
- Boolean

PRIMITIVE DATA

- **Numerical Type:**

- Integer
- Floating/Double

- **Character:** This type of data consists of the set of characters available for a specific language on a specific computer. The most used character code is ASCII.

- A **String** is multiple characters together

- **Boolean:** This is the simplest data type, as it only has two values (TRUE (1) and FALSE (0))

Integer	Values	-inf, ..., -2, -1, 0, 1, 2, ..., inf
	Operations	*, +, -, %, /, ...
Float	Values	..., -0.6, 0.0, 0.5, ...
	Operations	*, +, -, %, /, ...
Character/ String	Values	'\n', 'A', ..., 'Z', 'a', ..., 'z'
	Operations	<, >, ==, ...
Boolean	Values	TRUE, FALSE
	Operations	<, >, ==, ...

PRIMITIVE DATA

- **Numerical Type:**

- Integer: **Age**
- Floating/Double: **Height**

- **Character:** This type of data consists of the set of characters available for a specific language on a specific computer. The most used character code is ASCII.

- A **String** is multiple characters together: **Country names**

- **Boolean:** This is the simplest data type, as it only has two values (TRUE (1) and FALSE (0)): **Gender – Woman (1), Man (0)**

Integer	Values	-inf, ..., -2, -1, 0, 1, 2, ..., inf
	Operations	*, +, -, %, /, ...
Float	Values	..., -0.6, 0.0, 0.5, ...
	Operations	*, +, -, %, /, ...
Character/ String	Values	'\n', 'A', ..., 'Z', 'a', ..., 'z'
	Operations	<, >, ==, ...
Boolean	Values	TRUE, FALSE
	Operations	<, >, ==, ...

ASSIGNMENT OPERATORS

Assignment Operators	
=	Makes a copy of the assigned object.
<=	Makes a copy of the assigned object.

```
> a=10
```

Assign the integer 10 to the variable **a**.

```
> b<-15
```

Assign the integer 15 to the variable **b**.

```
>
```

```
> a+b
```

Perform the addition of variable **a** and **b**.

```
[1] 25
```

```
>
```

```
> c<-a+b
```

Assign the result of the addition of variable **a** and **b** to **c**.

```
> c
```

```
[1] 25
```

```
> rm(a)
```

Remove variable **a** from the environment.

```
> gc()
```

Clean the memory.

EXERCISE 1.1

Choose two operators from each category and apply them to different pairs of primitive data types. When applying them, think of possible contexts where you would use them, e.g. “+” to add the years lived by a person.

Operators					
Aritmetic		Comparative		Logical	
+	addition	<	less than	! x	logic NO
-	subtraction	>	more than	x & y	element-wise AND
*	multiplication	<=	less or equal than	x && y	single comparison AND
/	division	>=	more or equal than	x y	element-wise OR
^	power	==	equal than	x y	single comparison OR
%%	integer division	!=	different than	xor(x,y)	exclusive OR

❖ DATA TYPES

A data type is a set of values and operations associated with those values.

Primitive: These are the simplest types of data, because they are not constructed from other types and are unique entities that cannot be decomposed into others. These data types are defined by a set of values and a set of operations that act on those values.

- Integer
- Floating/Double
- Character and String
- Boolean

Composite Data: These are types of data whose values are collections of primitive data.

- Vectors

VECTORS

A vector is a collection of observations or measurements of the same type, for example the ages of 50 people or the number of coffees we drink on different days of the week. Vectors are the simplest types of aggregated data; it is with them that more complex structures are created.

In R, the function with which a vector is created is `c()`:

```
R> myvec <- c(1,3,1,42)
R> myvec
[1] 1 3 1 42
```

VECTORS: FACTORS

As mentioned in page 10, strings and Booleans data types are the equivalent of categories (country name) and dichotomous (gender) variables. However, to make this clear to R, we need to use the function **factor()**.

Create the vector and check its class.

```
> COUNTRY=c("MX","DE","NL")
> class(COUNTRY)
[1] "character"
```

Use **factor** to turn the vector into a categorical variable.

```
>
> COUNTRY=factor(COUNTRY,levels = c("MX","DE","NL"),labels = c("Mexico","Germany","Netherlands"))
> class(COUNTRY)
[1] "factor"
```

Use **levels** to see the categories. The first one is the reference.

```
> levels(COUNTRY)
[1] "Mexico"      "Germany"     "Netherlands"
```

EXERCISE 1.2

Create a categorical vector with four of the Dutch political parties. You can check them here:

https://en.wikipedia.org/wiki/List_of_political_parties_in_the_Netherlands

How would use their acronyms and their names?

2. DATAFRAMES

Dataframes.
From table/csv to dataframes & vice-versa.
Tidyverse.

❖ DATAFRAMES

A dataframe is a special type of R object where all the items are vectors of the same length. In a dataframe, the elements of different columns can be of different types.

In the example beneath, we are constructing a dataframe with 3 columns ('n', 's', and 'b'). 'n' is numeric. 's' is string. 'b' is Boolean.

To glue them together, we use the R function **data.frame()**.

Data Frame Construction

```
> n = c(2, 3, 5)
> s = c("aa", "bb", "cc")
> b = c(TRUE, FALSE, TRUE)
> df = data.frame(n, s, b)
```

❖ DATAFRAMES

A dataframe is a special type of R object where all the items are vectors of the same length. In a dataframe, the elements of different columns can be of different types.

In the example beneath, we are constructing a dataframe with 3 columns ('n', 's', and 'b'). 'n' is numeric. 's' is string. 'b' is Boolean.

To glue them together, we use the R function **data.frame()**.

Data Frame Construction

```
> n = c(2, 3, 5)
> s = c("aa", "bb", "cc")
> b = c(TRUE, FALSE, TRUE)
> df = data.frame(n, s, b)
```

R Example Data Frame

```
> mtcars
```

	mpg	cyl	disp	hp	drat	wt	...
Mazda RX4	21.0	6	160	110	3.90	2.62	...
Mazda RX4 Wag	21.0	6	160	110	3.90	2.88	...
Datsun 710	22.8	4	108	93	3.85	2.32	...
.....							

❖ DATAFRAMES

The dataframe name

The columns:
`names(mtcars)`
or
`col.names(mtcars)`

The row names:
`row.names(mtcars)`

```
> mtcars
```

	mpg	cyl	disp	hp	drat	wt	...
Mazda RX4	21.0	6	160	110	3.90	2.62	...
Mazda RX4 Wag	21.0	6	160	110	3.90	2.88	...
Datsun 710	22.8	4	108	93	3.85	2.32	...
.....							

A variable:
`mtcars$disp`

A value or element:
`mtcars$wt[3]`

EXERCISE 2.1

Create a dataframe to save the artists information:

1. Mary Leonora Carrington. British-born, naturalized Mexican. Surrealist painter. Died on May 25, 2011.
2. Remedios Varo. Spanish-born, naturalized Mexican. Surrealist painter. Died: October 8, 1963.
3. David Alfaro Siqueiros. Mexican. Social realist painter. Died: January 6, 1974.

OPEN, MODIFY, AND SAVE

With the *foreign* library you can manipulate databases other than ".csv"

Open: `read.csv()`

You can open “csv” files using this function.

```
read.csv(file, header = TRUE, sep = ",", quote = "\"",  
         dec = ".", fill = TRUE, comment.char = "", ...)
```

Save: `write.csv()`

```
write.table(x, file = "", append = FALSE, quote = TRUE, sep = " ",  
           eol = "\n", na = "NA", dec = ".", row.names = TRUE,  
           col.names = TRUE, qmethod = c("escape", "double"),  
           fileEncoding = "")
```


```
write.csv(...)
```

EXERCISE 2.2

Save the artists dataframe. Do you need the row.names parameter?

```
DIR="WRITE HERE THE PATH TO THE DATA\\"
```

```
write.csv(Artist, paste0(DIR, "Data_Name.csv"), row.names = FALSE)
```



With the command **help("paste0")** you can check what this function do.

REFERENCES

- Albert, Jim, and Maria Rizzo. *R by Example: Concepts to Code*. Use R! New York, NY: Springer New York, 2012. <https://doi.org/10.1007/978-1-4614-1365-3>.
- Davies, Tilman M. *The Book of R: A First Course in Programming and Statistics*. San Francisco: No Starch Press, 2016.
https://web.itu.edu.tr/~tokerem/The_Book_of_R.pdf
- Wickham, Hadley, Mine Çetinkaya-Rundel, and Garrett Grolemund. *R for data science*. "O'Reilly Media, Inc.", 2023. Accessed May 7, 2024.
<https://r4ds.hadley.nz/>.

❖ SESSION 2: TIDYVERSE [HTTPS://WWW.TIDYVERSE.ORG/](https://www.tidyverse.org/)



R packages for data science

The tidyverse is an opinionated **collection of R packages** designed for data science. All packages share an underlying design philosophy, grammar, and data structures.

Install the complete tidyverse with:

```
install.packages("tidyverse")
```



Join us!



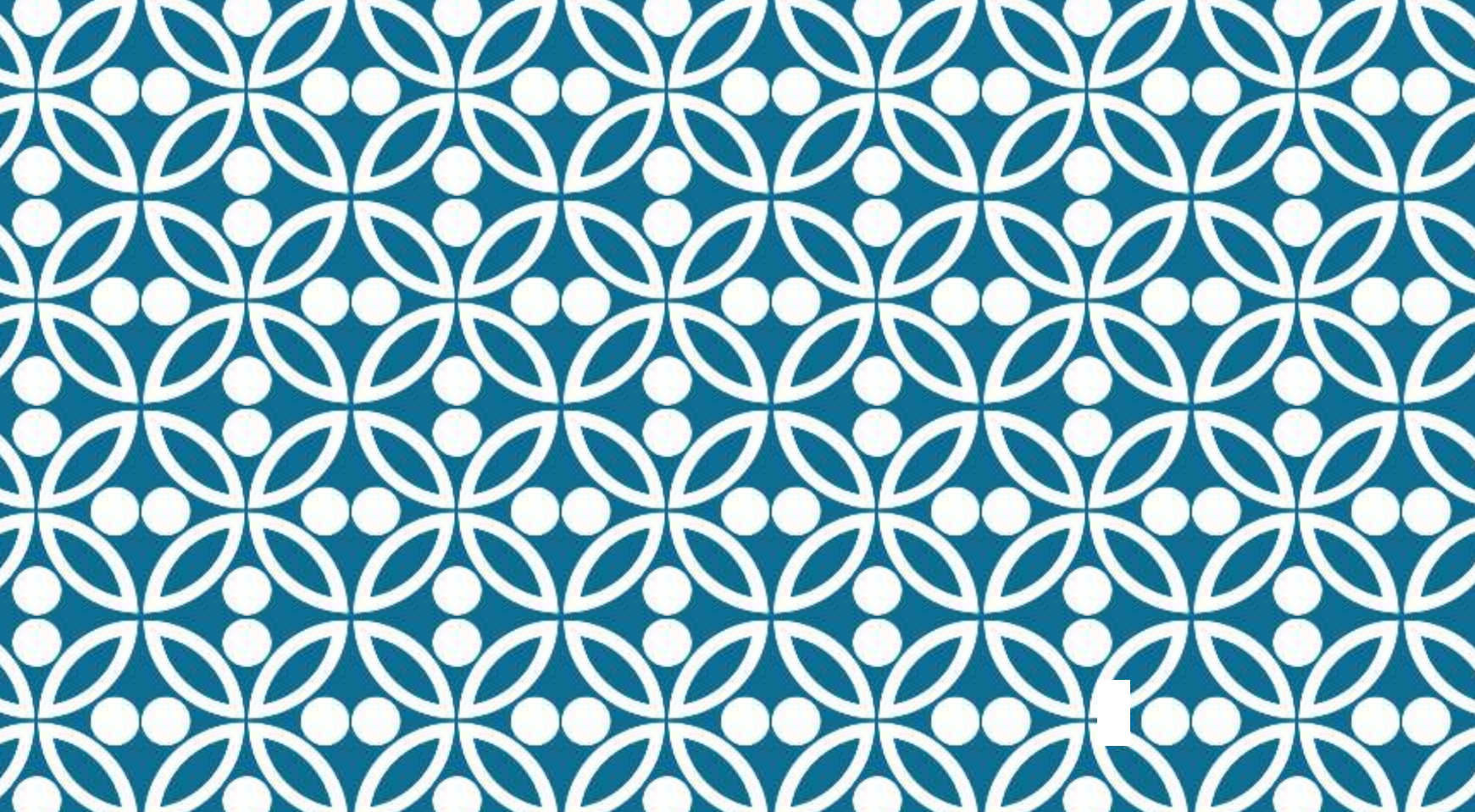
<https://forms.office.com/e/8Bgd2YsasJ>

All about the lab:

<https://societal-analytics.nl/>

Contact us at:

analytics-lab.fsw@vu.nl



<https://sofiagil.github.io/>

THANKS!

Dr. Sofia Gil-Clavel