INTRODUCTION TO DATA SCIENCE WITH R

Session 2: Data exploration basics

Ina Bornkessel-Schlesewsky

January 18, 2023

DATA EXPLORATION AND VISUALISATION: FIRST STEPS

REPRODUCIBILITY CHECKLIST

WHAT DOES IT MEAN FOR A DATA ANALYSIS TO BE “REPRODUCIBLE”?

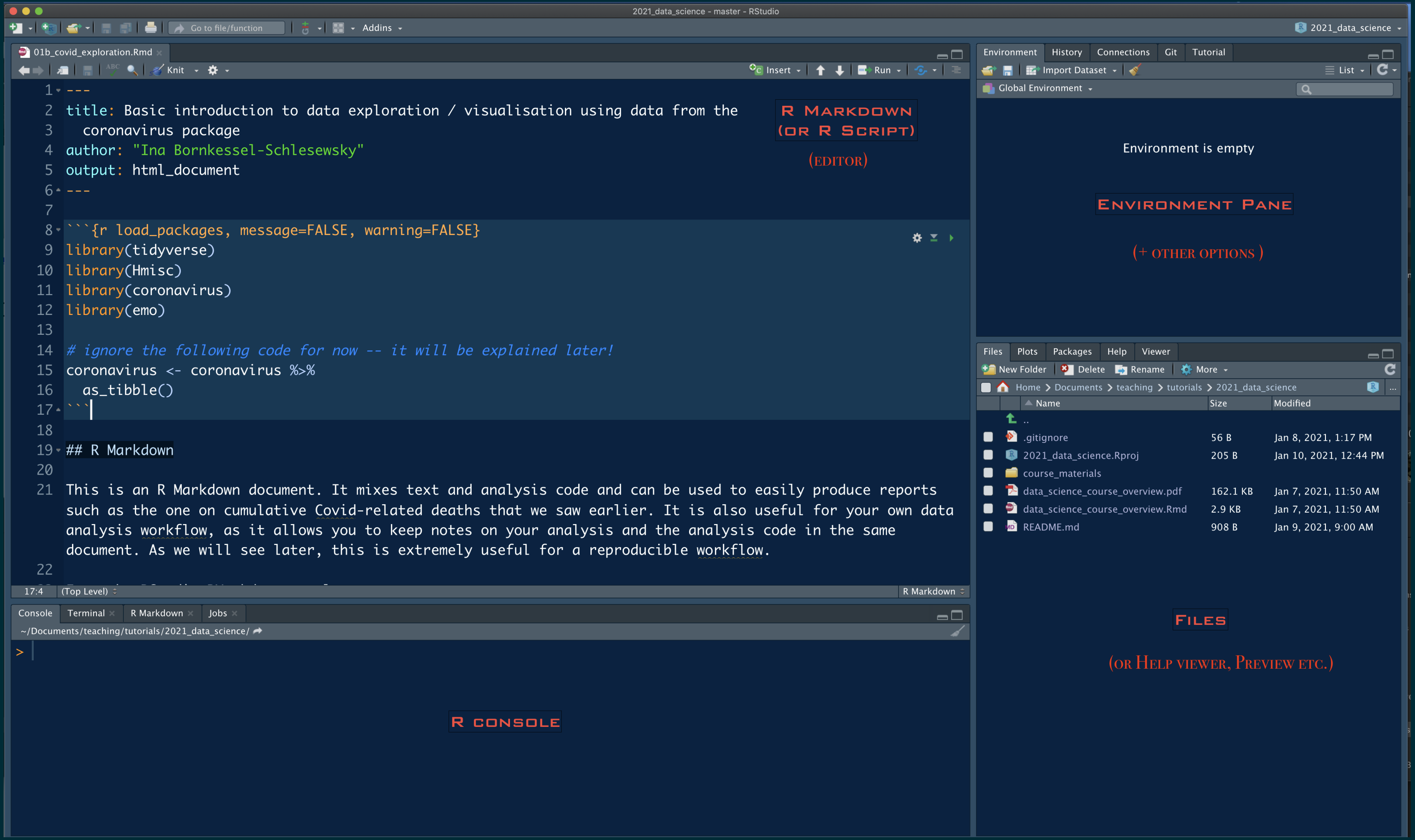
- Are the tables, figures (and any other results reported) reproducible from the code and data?
- Does the code actually do what you think it does?
- In addition to what was done, is it clear *why* it was done?

REPRODUCIBILITY TOOLKIT

- Scriptability → R
- Literate programming (code, narrative, output in one place) → Quarto
- Version control → Git / GitHub (**more on this on Day 3**)

Adapted from https://github.com/rstudio-education/datascience-box/blob/master/course-materials/slides/u1-d02-toolkit-r/u1-d02-toolkit-r.Rmd

A BIT MORE ON R AND RSTUDIO



~~~
---
title: Basic introduction to data exploration / visualisation using data from the
  coronavirus package
author: "Ina Bornkessel-Schlesewsky"
output: html_document
---

```{r load_packages, message=FALSE, warning=FALSE}
library(tidyverse)
library(Hmisc)
library(coronavirus)
library(emo)

# ignore the following code for now -- it will be explained later!
coronavirus <- coronavirus %>%
  as_tibble()
```

## R Markdown

This is an R Markdown document. It mixes text and analysis code and can be used to easily produce reports
such as the one on cumulative Covid-related deaths that we saw earlier. It is also useful for your own data
analysis workflow, as it allows you to keep notes on your analysis and the analysis code in the same
document. As we will see later, this is extremely useful for a reproducible workflow.
~~~

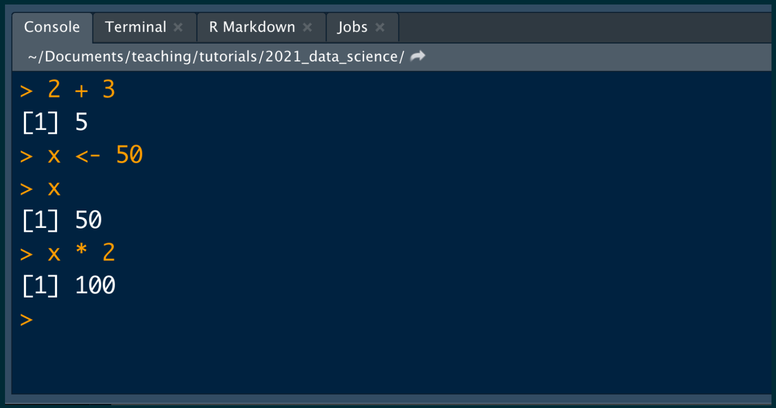
```
~/Documents/teaching/tutorials/2021_data_science/
>
```

Name	Size	Modified
..		
.gitignore	56 B	Jan 8, 2021, 1:17 PM
2021_data_science.Rproj	205 B	Jan 10, 2021, 12:44 PM
course_materials		
data_science_course_overview.pdf	162.1 KB	Jan 7, 2021, 11:50 AM
data_science_course_overview.Rmd	2.9 KB	Jan 7, 2021, 11:50 AM
README.md	908 B	Jan 9, 2021, 9:00 AM

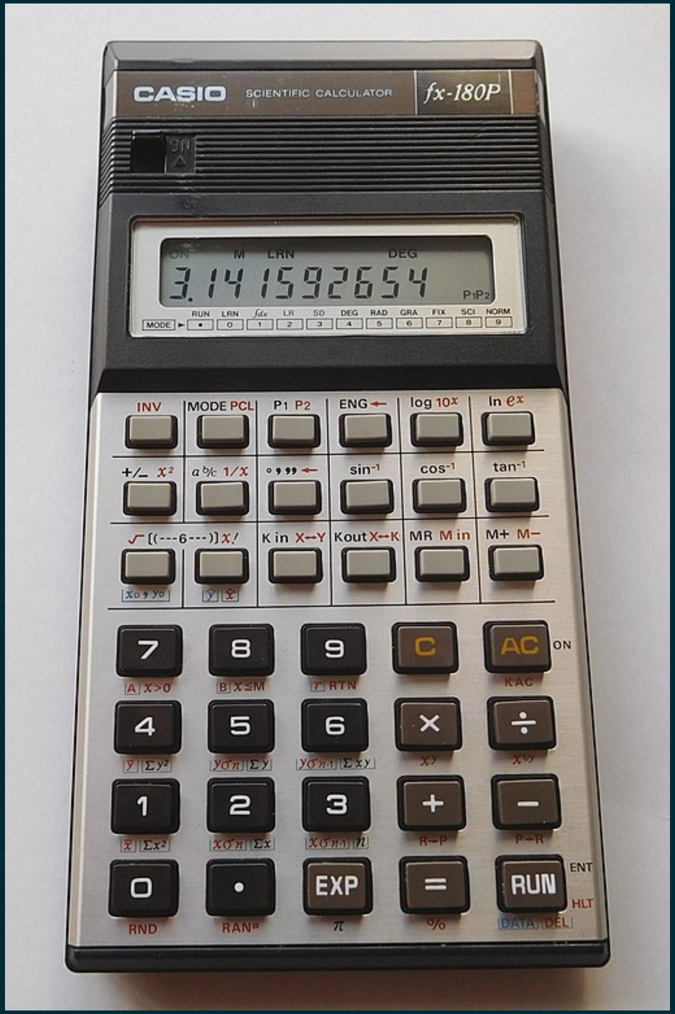
Note: this is the default layout - all the panes can be moved around to best suit your workflow. Note also the possible change of appearance (e.g. dark themes).

R CONSOLE

- this is where the “magic happens”,
  i.e. where the calculations take place
- think of a calculator on steroids 😂
- we can interact with the console directly (see example)
- mostly, however, we will be working with scripted input



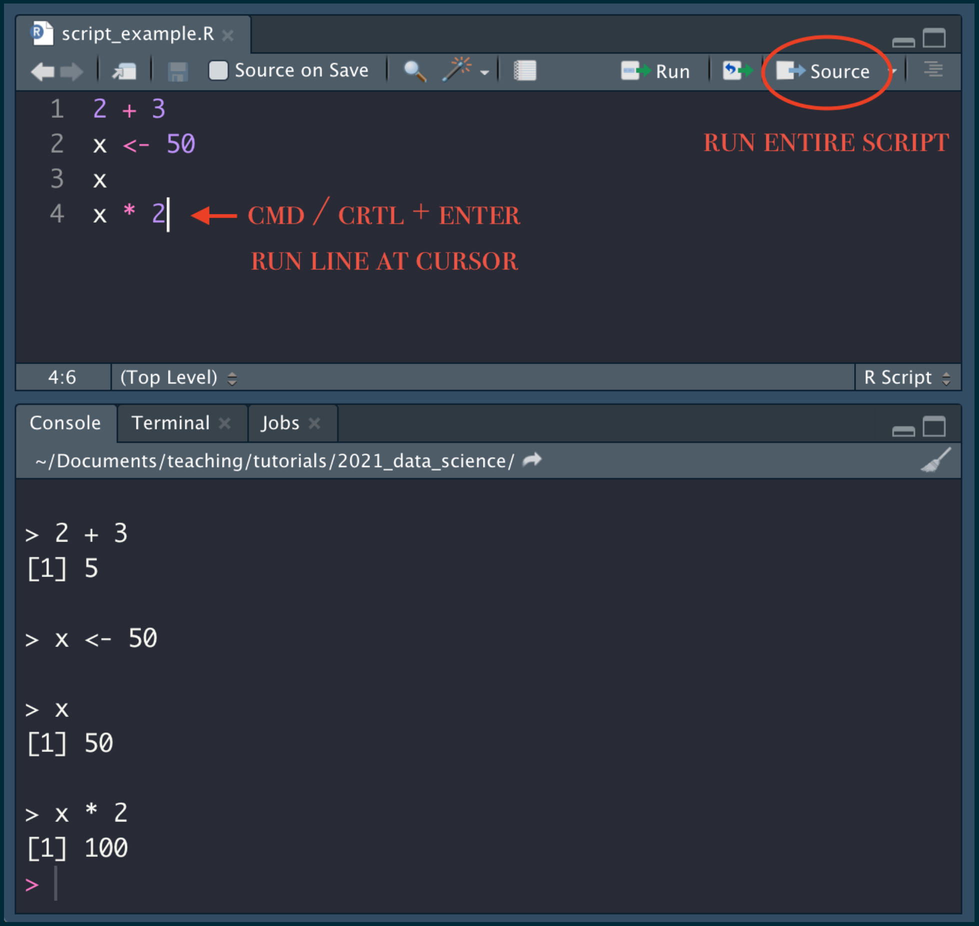
```
> 2 + 3
[1] 5
> x <- 50
> x
[1] 50
> x * 2
[1] 100
>
```



Cjp24 CC BY-SA 4.0, via Wikimedia Commons

SCRIPTED R CODE

- An R “script” is simply a text file with a collection of R commands
- Think of it as a set of instructions that you can feed into your “calculator”
- This is an important step towards reproducibility, as it means that you have a record of your analysis and you can recreate it at any time



```
2 + 3
x <- 50
x
x * 2
```

RUN ENTIRE SCRIPT

CMD / CRTL + ENTER

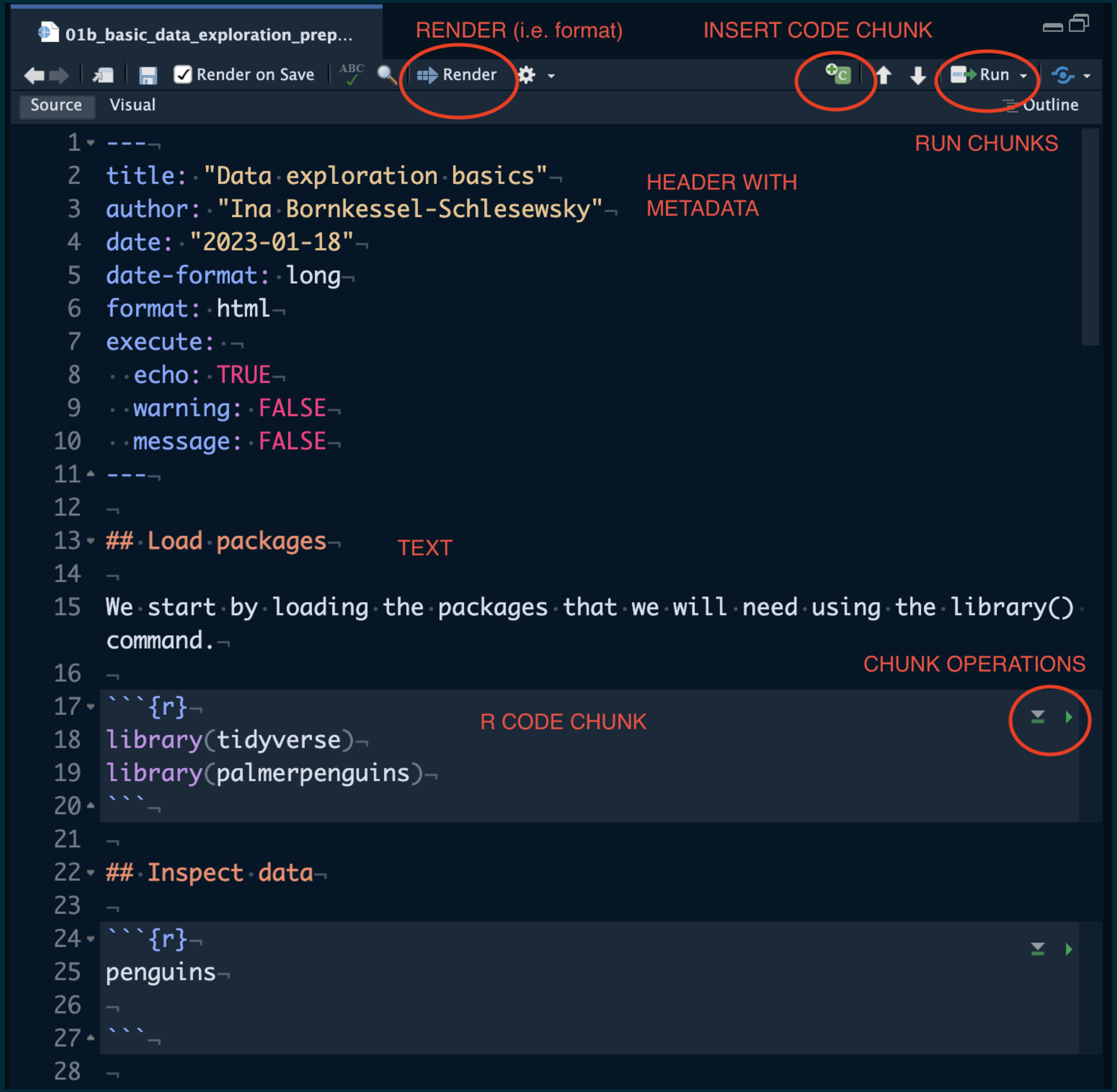
RUN LINE AT CURSOR

```
> 2 + 3
[1] 5

> x <- 50

> x
[1] 50

> x * 2
[1] 100
>
```

QUARTO

- Mixes text (formatted using Markdown) and R code
- Allows for documentation of analysis steps (the “why”): the next step towards reproducibility
- Easily generate reproducible reports in different formats (.html, .pdf, .docx)
- You can even use Quarto to create slides for presentations (these slides were created in Quarto), interactive tutorials and web sites.

QUARTO



RENDER (i.e. format)

INSERT CODE CHUNK

RUN CHUNKS

HEADER WITH METADATA

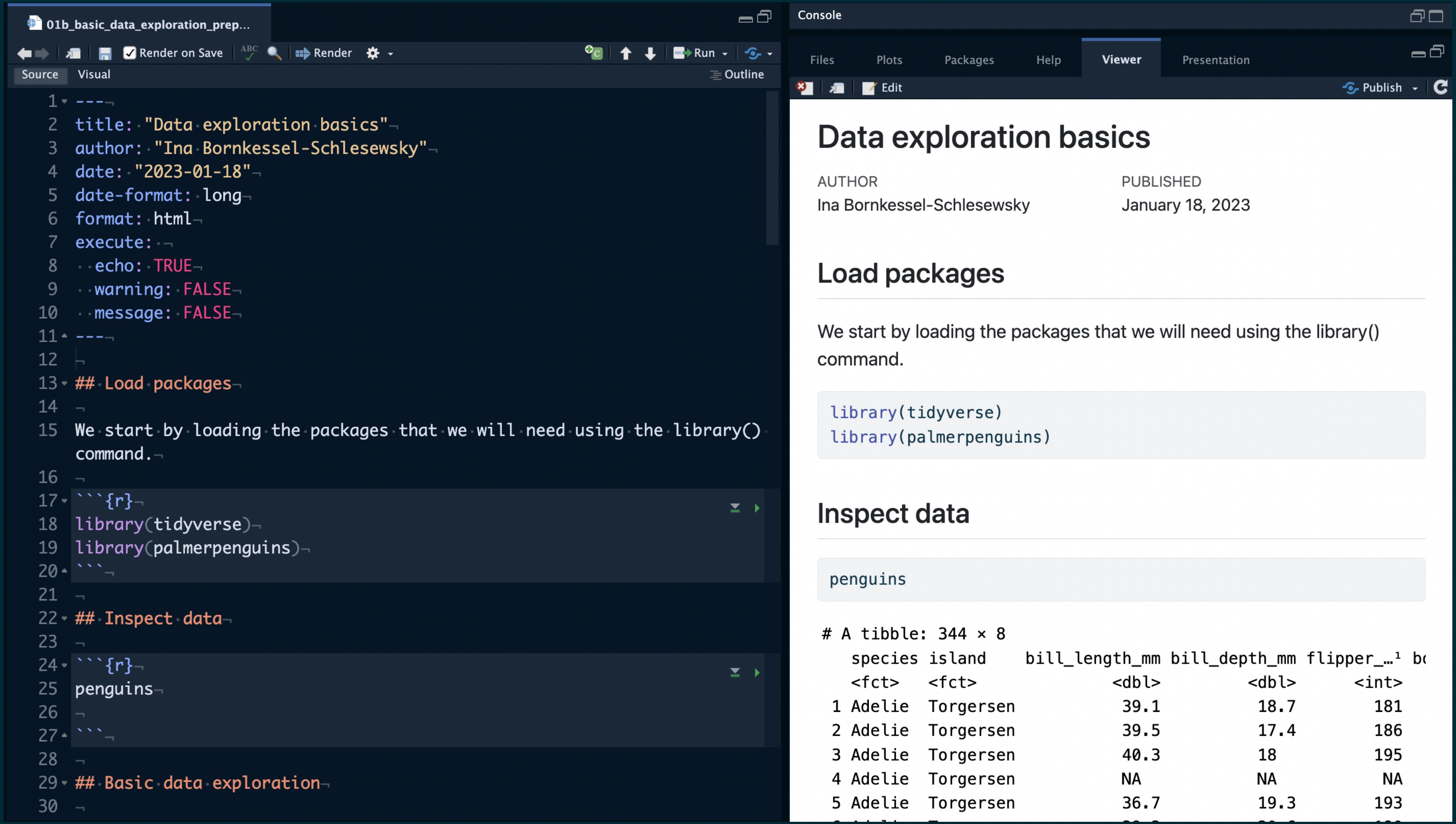
TEXT

CHUNK OPERATIONS

R CODE CHUNK

~~~
---
title: "Data exploration basics"
author: "Ina Bornkessel-Schlesewsky"
date: "2023-01-18"
date-format: long
format: html
execute: 
  echo: TRUE
  warning: FALSE
  message: FALSE
---

## Load packages

We start by loading the packages that we will need using the library() command.

```{r}
library(tidyverse)
library(palmerpenguins)
```

## Inspect data

```{r}
penguins

```

~~~

QUARTO RENDERED DOCUMENT



~~~
---
title: "Data exploration basics"
author: "Ina Bornkessel-Schlesewsky"
date: "2023-01-18"
date-format: long
format: html
execute: 
  echo: TRUE
  warning: FALSE
  message: FALSE
---

## Load packages

We start by loading the packages that we will need using the library() command.

```{r}
library(tidyverse)
library(palmerpenguins)
~~~

Inspect data

```{r}
penguins

```

Basic data exploration

```

## Data exploration basics

AUTHOR
Ina Bornkessel-Schlesewsky

PUBLISHED
January 18, 2023

### Load packages

We start by loading the packages that we will need using the library() command.

```
library(tidyverse)
library(palmerpenguins)
```

Inspect data

```
penguins
```

```
# A tibble: 344 × 8
   species island    bill_length_mm bill_depth_mm flipper_…¹ bo
   <fct>   <fct>              <dbl>         <dbl>      <int>
 1 Adelie  Torgersen           39.1          18.7        181
 2 Adelie  Torgersen           39.5          17.4        186
 3 Adelie  Torgersen           40.3          18          195
 4 Adelie  Torgersen           NA            NA           NA
 5 Adelie  Torgersen           36.7          19.3        193
```
```

# YOUR TURN: LET'S EXPLORE THE PENGUINS DATA WITH THE HELP OF A QUARTO DOCUMENT!

- For this, we will be using a technique called "live coding".
- I will show you how to construct R code, talking you through the rationale for various coding choices.
- By copying my code in real time, you will get a feel for how the process works.
- Always type the code; don't copy and paste it, even if certain aspects keep repeating. This will ensure that you start to build up the "muscle memory" required to conduct data exploration and visualisation yourself in future.

- For this session, we will be working with the file **01b_basic_data_exploration.qmd**, which you can find under **exercises > 01b_basic_data_exploration**

# KEY LEARNING GOALS

## BY THE END OF THIS SESSION, YOU SHOULD

- understand the basics of how to work with a Quarto document for data exploration
- know how to inspect a data frame
- be able to undertake simple data manipulation steps (filtering, sorting, summarising)
- be able to create basic graphs of different types using the `ggplot()` function

## RESOURCES

- R for Data Science (2nd ed.), Chapter 2
- Introduction to dplyr
- Posit Cheatsheets; see the ones on data transformation and visualisation in particular

# FOR YOUR REFERENCE

# USING BASIC DATA EXPLORATION FUNCTIONS

- Basic data exploration functions are typically verbs to reminder you that they “do stuff” with a dataset (e.g. *filter*, *arrange*, *summarise*)

- start with the data that you want to explore / manipulate (e.g. `penguins`)
- use the “pipe” operator `|>` to send the data to a function (N.B. keyboard shortcut for the pipe: CTRL/CMD + SHIFT + M)
- specify the function and any additional parameters in parentheses (these differ depending on the function)
- this basic pattern works for all the functions we’ve looked at and more!

```r
penguins |>
  filter(species == "Adelie")
```

```r
penguins |>
  arrange(body_mass_g)
```

```r
penguins |>
  summarise(m_mass = mean(body_mass_g,
                          na.rm=TRUE))
```

COMBING BASIC FUNCTIONS

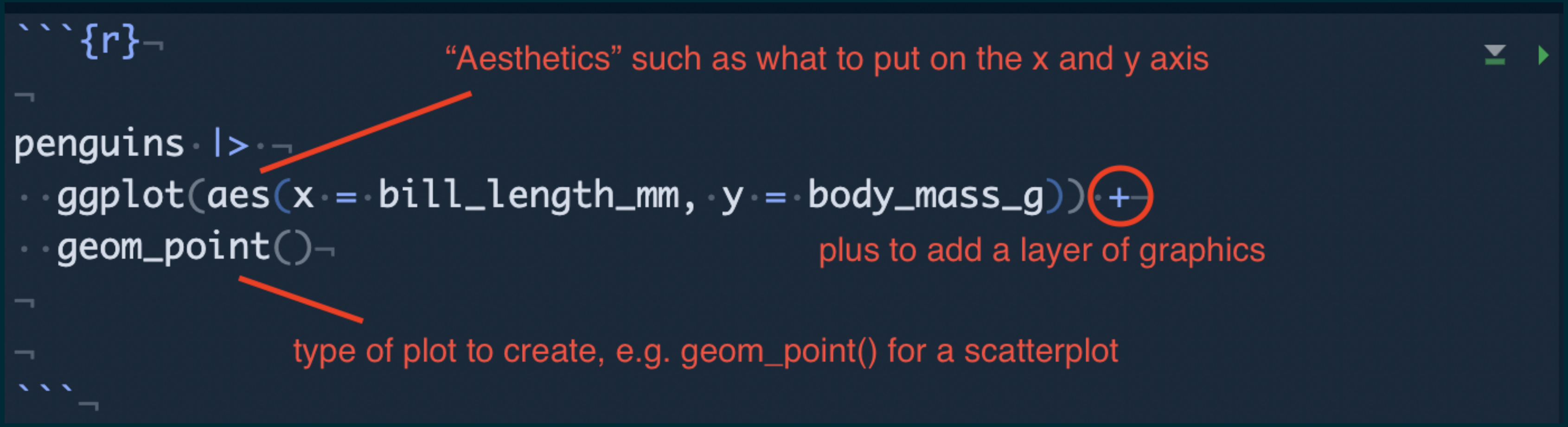
- You can use pipes to chain together different functions
- This is a very powerful basis for data exploration and visualisation!

```r
penguins |>
  filter(species == "Adelie") |>
  arrange(body_mass_g)
```

```r
penguins |>
  filter(island == "Dream") |>
  summarise(m_mass = mean(body_mass_g,
                          na.rm=TRUE))
```

(VERY) BASIC PLOTTING

ANATOMY OF A GGPLOT

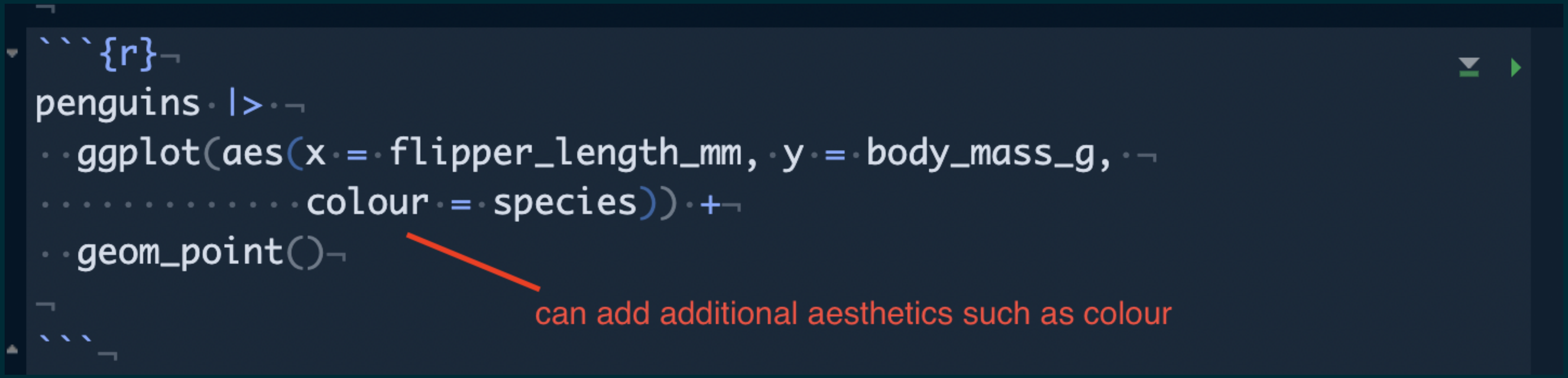


~~~
```{r}

penguins |> 
  ggplot(aes(x = bill_length_mm, y = body_mass_g)) +
  geom_point()


```
~~~

“Aesthetics” such as what to put on the x and y axis

plus to add a layer of graphics

type of plot to create, e.g. geom_point() for a scatterplot



~~~
```{r}
penguins |> 
  ggplot(aes(x = flipper_length_mm, y = body_mass_g, 
             colour = species)) +
  geom_point()

```
~~~

can add additional aesthetics such as colour

USEFUL RSTUDIO KEYBOARD SHORTCUTS

- insert code chunk: **CMD (Mac) / CTRL (Windows) + Option / ALT + i**
- insert pipe: **CMD / CTRL + M**
- run block of code: **CMD / CTRL + Return / Enter**
  - you can place the cursor anywhere in a connected block of code for this
  - if you want to run only a particular selection of code, say part of a a block, you can select it using the cursor and then use this keyboard shortcut

EXERCISES

Complete the exercises in the **session2_exercises.qmd** document (in the *exercises* > *01b_basic_data_exploration* directory)