R Markdown

A lot more than what we think!

Elisa Schneider and Divya Seernani

RLadies-Freiburg

02/07/2020

Contesnts of Today

- **Intro to R Markdown**
- **Customizing the style**
  - Knitting tables
  - Customized stiles
  - Tabed sections
- **Automated Reporting**
  - Embedded R code
  - Parametrized R-Markdown and automated reports
- **Making presentations**
  - R presentations
  - Xaringan presentation

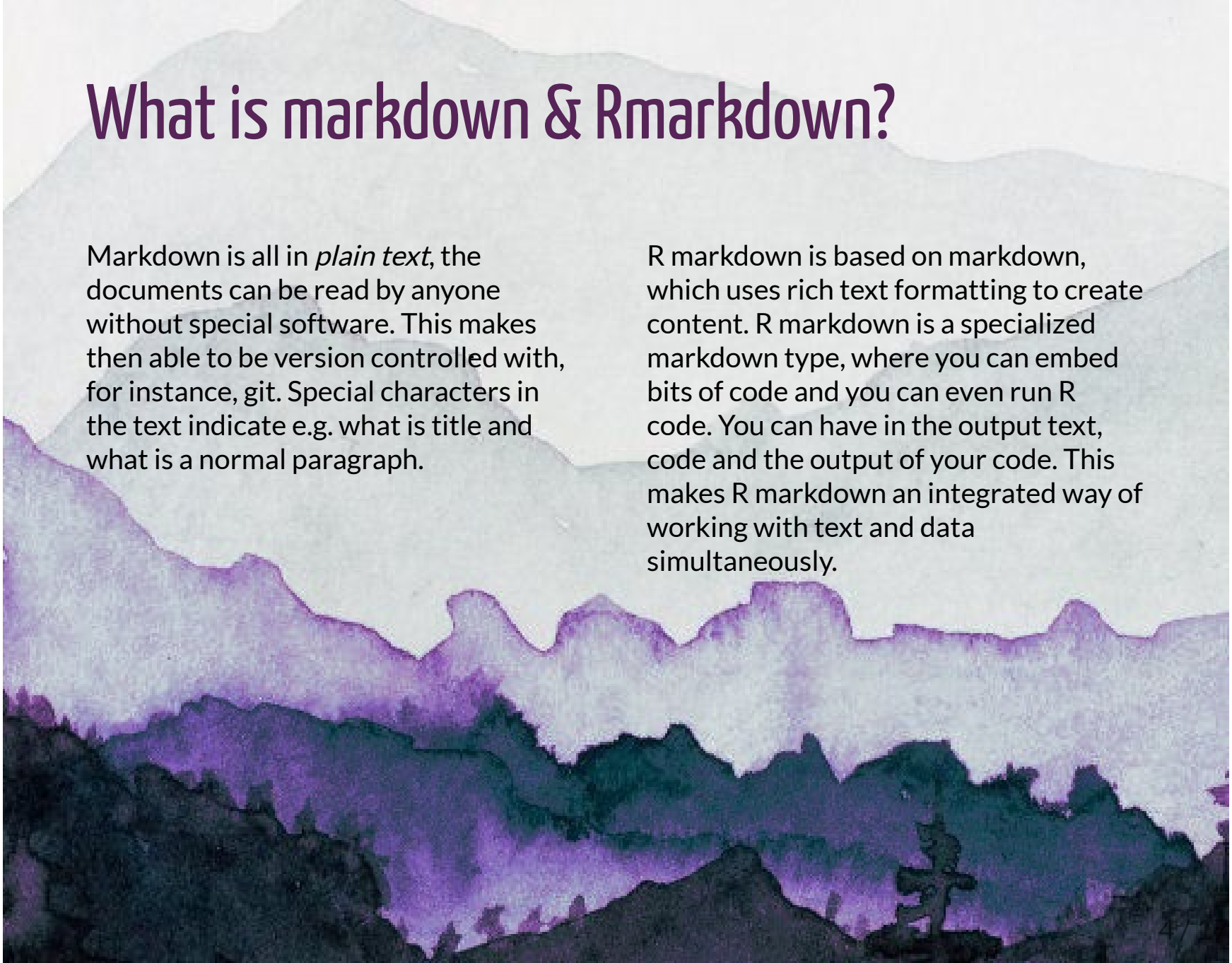
Get Started

Part of this presentation is based on the Oslo meetup on R-Markdown

What is markdown & Rmarkdown?

Markdown is all in *plain text*, the documents can be read by anyone without special software. This makes then able to be version controlled with, for instance, git. Special characters in the text indicate e.g. what is title and what is a normal paragraph.

R markdown is based on markdown, which uses rich text formatting to create content. R markdown is a specialized markdown type, where you can embed bits of code and you can even run R code. You can have in the output text, code and the output of your code. This makes R markdown an integrated way of working with text and data simultaneously.



Rmarkdown pipeline

Most commonly R markdown is turned into another document type using {knitr} and pandoc

This information is not crucial to remember, but good to know.

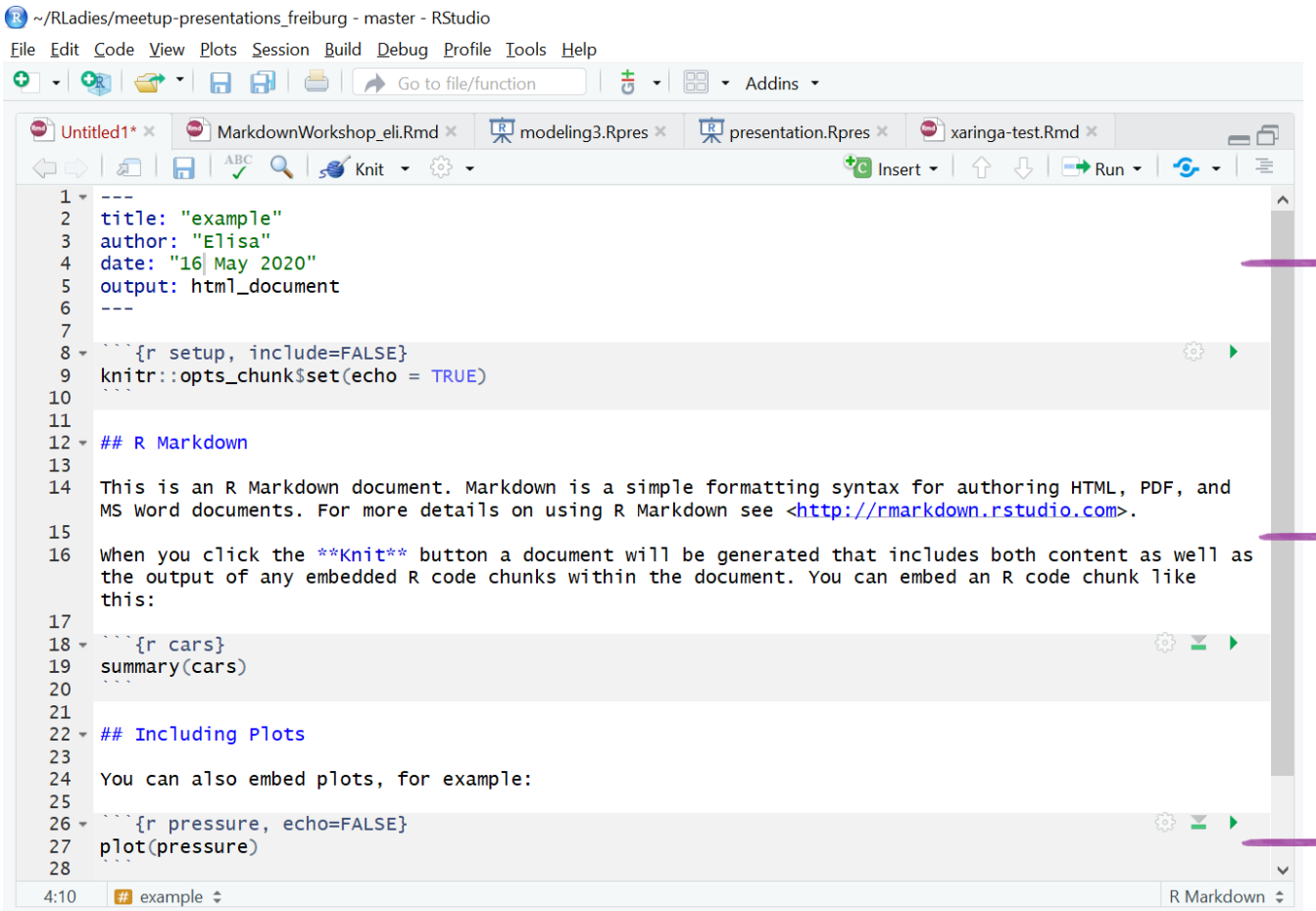
So it's just text?

Basically, yes.

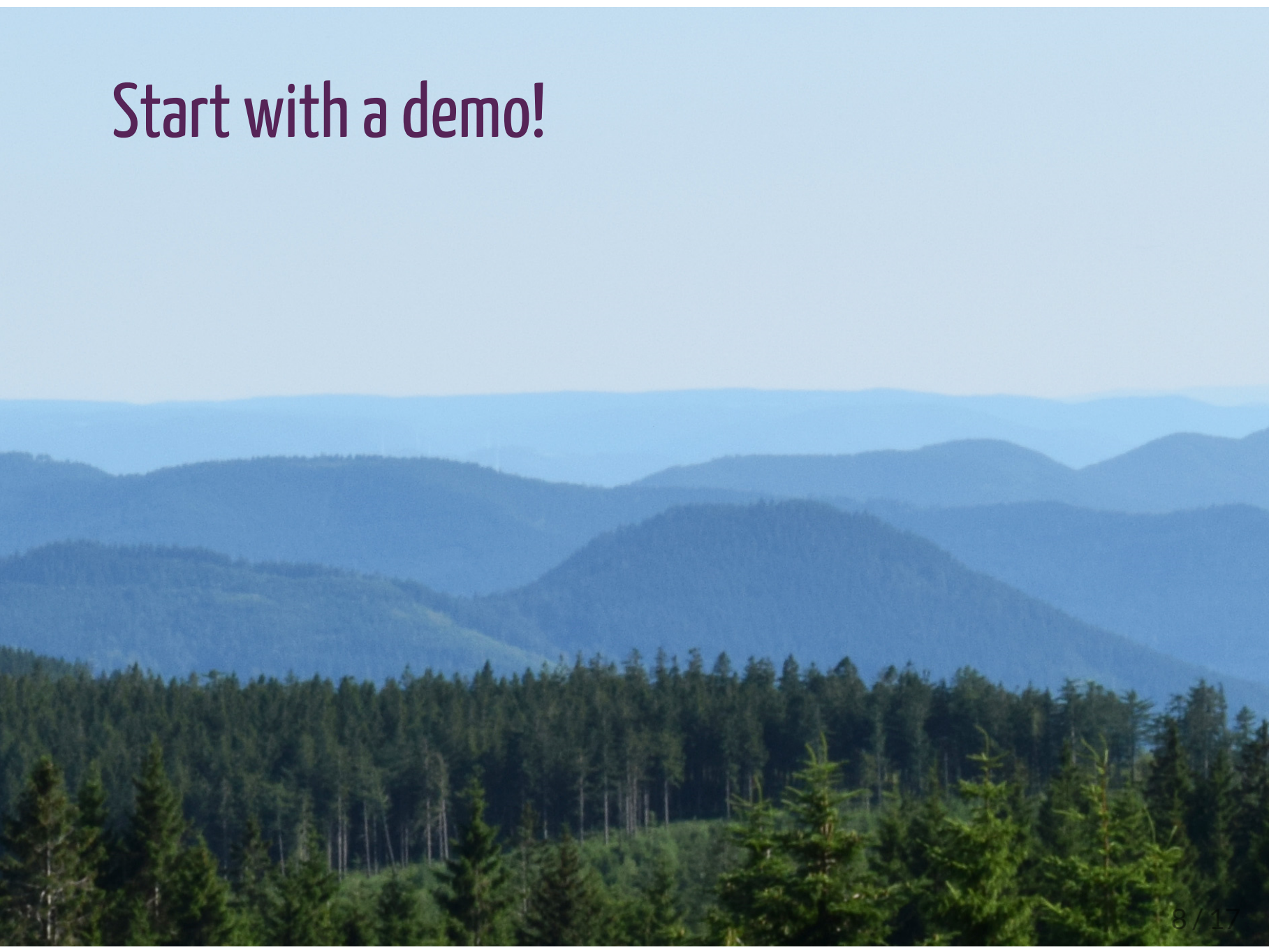
While any text-editor that understands markdown (many new editors do) will render the formatted text, even without the rendering anyone with a text-editor can *read* the text in the file.

This makes is easy to share with people who might not have access to specialized, expensive tools can still access the documents.

Components of the RMarkdown file



Start with a demo!



The clean environment

Rmarkdown runs in a completely clean R environment.

Nothing in your current working environment will affect compiling your document.

This is good because it means the document is reproducible.

This is frustrating *to begin with* because it is an unfamiliar way of working to many.

But working like this is the best way of working, learn from it and adopt it as a general coding style.

Adding navigation bar

```
---
title: "Bäumen"
author: "Elisa Schneider"
date: "11 Februar 2020"
output:
  html_document:
    toc: true
    toc_float: true
    number_sections: true
---

# Eiche
Here some info about oak trees

# Buche
And here about beech trees
```

It has to be added into the YAML metadata.

You can also add section number.

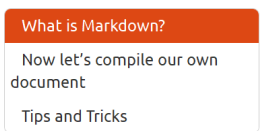
There are some tools also to organize sections while you are working on R and R-Markdown.

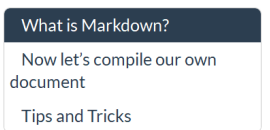
One of the thinks I love the most since I discovered it!

Customized styles in Rmarkdown

- Valid themes include *default*, *cerulean*, *journal*, *flatly*, *readable*, *spacelab*, *united*, *cosmo*, *lumen*, *paper*, *sandstone*, *simplex*, *and* *yeti*. Pass null for no theme (in this case you can use the css parameter to add your own styles).
- Highlight specifies the syntax highlighting style. Supported styles include *default*, *tango*, *pygments*, *kate*, *monochrome*, *espresso*, *zenburn*, *haddock*, *breezedark*.

```
title: "Markdown Workshop"
author: "Divya and Elisa"
date: "23 5 2020"
output:
  html_document:
    toc: True
    toc_float: true
    theme: united
    highlight: tango
```





Tabbed sections

Adding tabs to the Markdown Document

You can organize content using tabs by applying the `.tabset` class attribute to headers within a document. This will cause all sub-headers of the header with the .tabset attribute to appear within tabs rather than as standalone sections. For example:

```
## Oak - _Quercus robur_ {.tabset}

### Leaves

(tab content)

### Flowers

(tab content)
```

R outside code chunks

This is very useful to insert summary stats within the text.

There are two possibilities to include code outside chunks in and `Rmd` file.

- Including code to show it. The code will not be run and will be displayed as code.
- Running the code and parsing the output as text. It might me useful to show the output fo some code as text. You have to specify that the code has to be run using r.

```
# R code inside Rmarkdown

Using the following code inside the text part of an Rmarkdown file we can include the output of the
command as plain text and include a list of the injury mechanisms inside the text without the need
to type them.

`unique(tbi_age$injury_mechanism)`

The injury mechanisms present in the data are:

`r unique(tbi_age$injury_mechanism)`
```

Automated Reporting

Being able to program reports is a super power of R markdown.

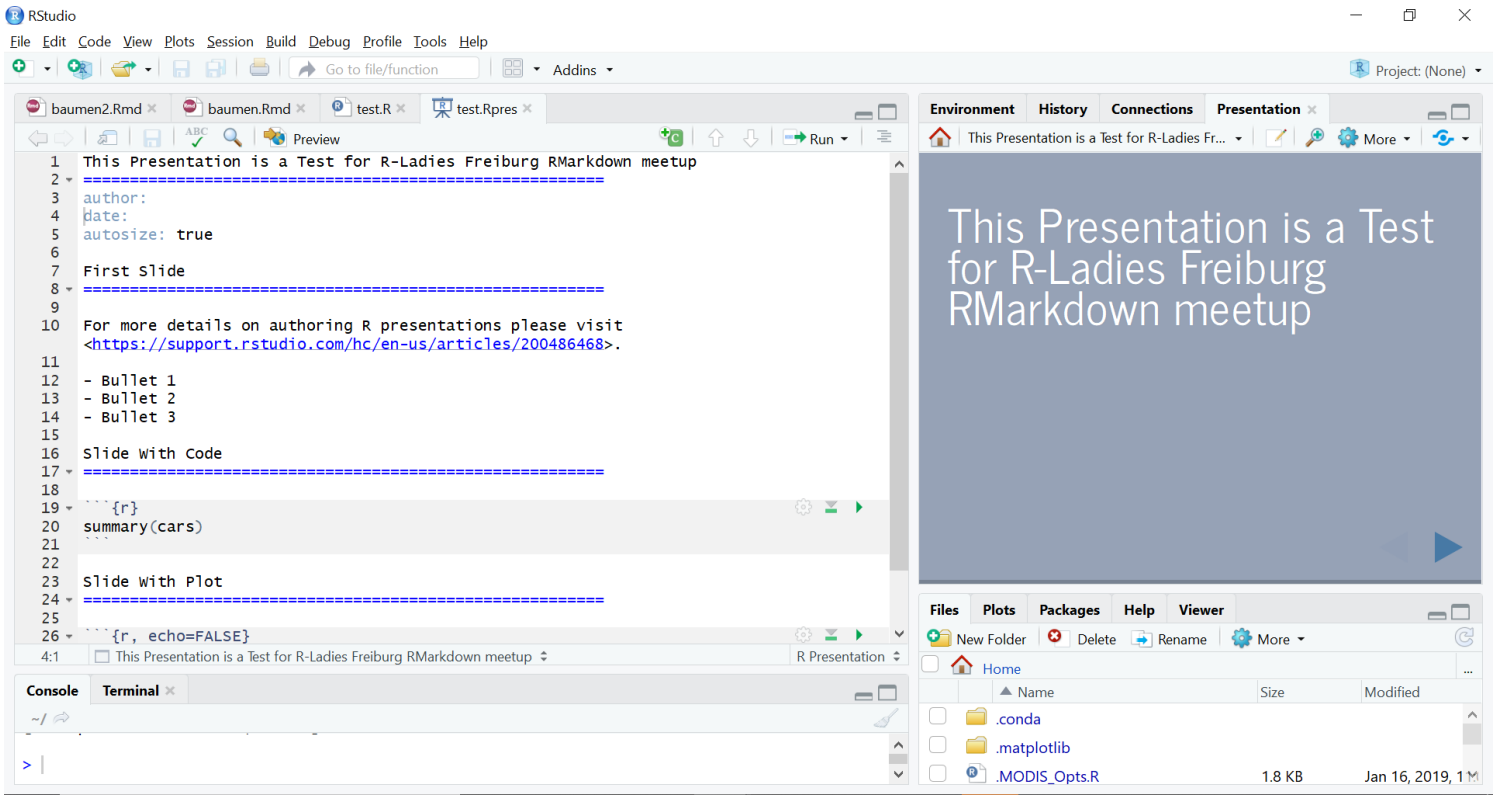
The motivation is really twofold: efficiency (maximize the reusabililty of code, minimize copying and pasting errors) and reproducibility (maximize the number of people and computers that can reproduce findings).

```
---
title: "Markdown Workshop"
author: "Divya and Elisa"
date: "23 5 2020"
output:
  html_document:
    toc: True
    toc_float: true
    theme: journal
    highlight: tango
params:
  tbi_age: !r readr::read_csv('data/tbi_age.csv')
```

R presentations

It is quite easy to use and very similar to R markdown.

`File -> New File -> R Presentation`



Xaringan Presentations

Install the **xaringan** package from Github or from *cran*:

```
devtools::install_github("yihui/xaringan")

install.packages('xaringan')
```

You are recommended to use the RStudio IDE, but you do not have to.

- Create a new R Markdown document from the menu `File -> New File -> R Markdown -> From Template -> Ninja Presentation`
- Click the `Knit` button to compile it;

Sources

- R Studio's Rmarkdown cheatsheet
- R Markdown: The definitive giude
- Automating Reports: Automating summary of surveys with rmarkdown
- Reports to impress your boss - R-Ladies Oslo - Video
- Reports to impress your boss - R-Ladies Oslo - github
- Definitive guide to xaringan