

Lecture 4

R Markdown: reproducible presentations & reports from R

me

19-3-2021

Packages

For this notebook we need the **rmarkdown** package. Install if necessary, and load the package.

Make sure you installed all the necessary software (i.e. TinyTex) as explained in the Course Syllabus.

IMPORTANT Test your LaTeX installation by creating a New R notebook (File -> New File -> New R Notebook). Then choose the option “Knit to PDF” in the box labeled “Preview”. This should produce a PDF document that pops up in a new window.

Introduction

You already learned about **markdown** in the start of this course. Markdown is a simple “language” for formatting text documents. Here we are going to combine markdown with R code in so-called R Markdown Notebooks, or R Notebooks for short.

You are already familiar with them, since this document as well as the other Notebooks you worked in before are all R Notebooks.

R notebooks have an “Rmd” extension (R-mark-down). Files with an Rmd extension can only be opened by **Rstudio** .

Rstudio uses the **rmarkdown** package to convert an R notebook into many different types of documents.

For example, we can convert this notebook into an HTML file. HTML files can be easily exchanged because they are readable in any web browser.

You can also compile the R notebooks into PDFs and Word files. It is therefore possible to write complete notes, presentations, scientific papers and reports using Markdown.

Below, we focus on two important ones: a presentation and a report.

Motivation: why use R Markdown?

The most important advantage of using R markdown is that the document becomes *reproducible*. The notebook contains all the information for others to reproduce exactly all the tables, numbers and figures in a report.

Creating an R Notebook

You can create a new R Notebook by selecting **File** , **New File** and then **R notebook** in **Rstudio**. When you start a new notebook, immediately save the notebook somewhere on disk.

Exercise 1: Create a New Notebook and save it.

Every time a notebook is saved, R studio automatically creates an HTML file in the same folder. If you want to see how your notebook looks in HTML version, press “Preview” above your notebook.

About Rmarkdown files

Let us have a closer look at your newly created Notebook document. This document is a Rmarkdown file, because its name ends with `.Rmd` .

An Rmarkdown file is made of three different 3 parts:

- An (optional) YAML Header with information about the document (title, output type etc)
- Markdown text
- A “code block” (also know as a “code chunk”)

YAML header

An Rmarkdown document starts with a single YAML header, after that you can have as many markdown text and code blocks as you want. (In case you are wondering: YAML stands for *YAML Aint Markup Language*).

It contains metadata, i.e. data about the document. For example, the author, the title, the desired output formats etc. The header starts with `---` and ends with `---`.

Exercise 2 Examine the header of your document you created above. What type of output will be produced? Add a `date` field to the header with the current date. Add an `author` field to the document. Use “Preview” to test if it works. If it does not work, google for it!

Markdown text

There is a separate notebook that refreshes and builds on the markdown commands you learned earlier in the course.

Exercise 3: Open the `markdown.Rmd` notebook in Rstudio and complete the exercises in it. Then return here.

Code blocks (“Chunks”)

With “Insert” you can insert an R code block

Below is an example of a code block with the output underneath.

You can run a code block in 2 ways: 1. By pressing the green “play” button in the top right corner of the code block 2. Using `Ctrl + Shift + Enter` on your keyboard.

Exercise 4: Type `1 + 1` in the code block and run the code block.

```
## [1] 2
```

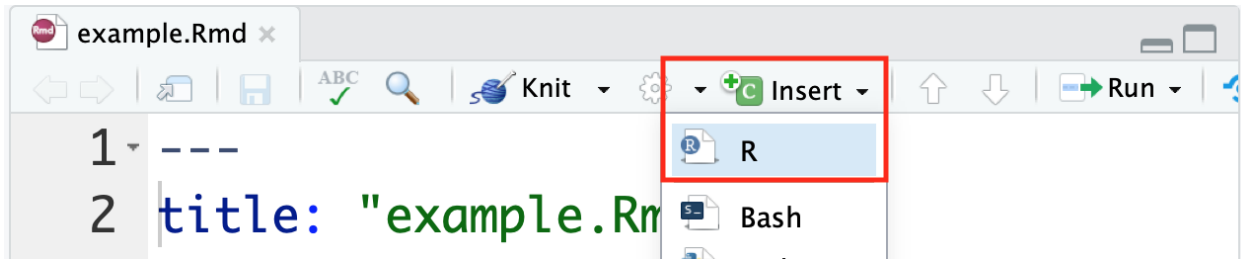


Figure 1: code block

Inline R code

Code results can be inserted directly into the text of a .Rmd file by enclosing the code with `r`.

Exercise 5: Create a new R Notebook. By default it already contains some text and a code chunk. At the end of the document, add the following line:

The dataset `mtcars` contains 32 records of car models.

Now preview the document: what happened to the R code?

it was executed

Chunk options

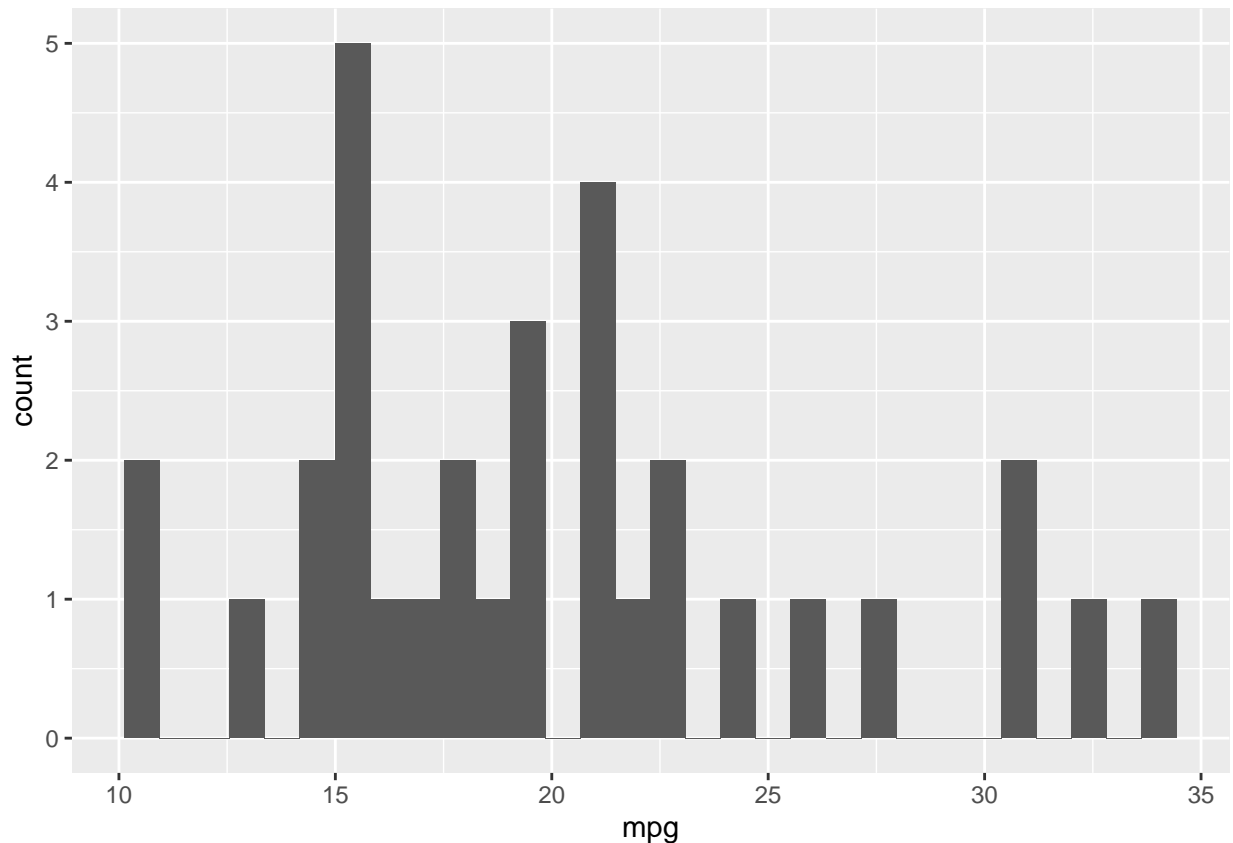
There are several useful chunk options. Include them between brackets (`{r ...}`) at the start of a new code chunk.

These are the most important ones for reproducible reports.

- `include = FALSE` prevents code and results from appearing in the finished file. R Markdown still runs the code in the chunk, and the results can be used by other chunks.
- `echo = FALSE` prevents code, but not the results from appearing in the finished file. This is a useful way to embed figures.
- `message = FALSE` prevents messages that are generated by code from appearing in the finished file.
- `warning = FALSE` prevents warnings that are generated by code from appearing in the finished.

Exercise 6: Create a new R-notebook and add the code chunk below to it. Use chunk options to create an HTML file as output that includes the results from `ggplot`, but excludes the code, and excludes the message from `ggplot()` about `stat_bin()`.

```
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```



Globally changing chunk options for the document

Sometimes you want all chunks in a document to have the same chunk options, for example, you only want the **output** of each chunk and not the code itself displayed in the final output. Use this code chunk at the top of your document to set chunk options for all chunks that follow it.

```
{r setup, include=FALSE} #knitr::opts_chunk$set(echo = FALSE) #
```

Other formats: PDF report

When you make a reproducible report, for example your bachelors thesis, the PDF document output format is a good choice.

PDFs are designed to look identical on all computer platforms and printers. Another advantage is that (similar to Word documents) content such as images and tables is included in a single file, making it easy to share the document with others.

Exercise 7 Knit the current document to a PDF document using the **Knit to PDF** option shown after clicking the ‘Knit’ button in Rstudio. Set up the global chunk options such that only the output from the code chunks is shown in the final PDF.

Other formats: PDF presentation

When you present the results of a data analysis, you can also benefit from a reproducible workflow. For reproducible presentations, so-called **Beamer presentations** are a good choice. Beamer presentations are slide shows in PDF document format.

You can create a Beamer presentation by Choosing **File -> New File -> R markdown then Presentation then PDF (Beamer)**.

Exercise 8 Create a PDF beamer presentation with a title “My Beamer slide show” , the slide content that is created by default, with a slide added that contains the ggplot graph from the code chunk exercise above.

Bonus: How does it work? From code to final document

This is bonus material for those who like to know what is going on “under the hood”.

We are using a set of (open source) software tools that work together to produce the final document.

First **rmarkdown** (not shown) sends the Rmd file to **knitr**. This runs the R code and converts the code output to markdown.

At this point we have a plain text file with markdown formatting. Then the markdown file is send to **pandoc** , this is a universal document converter tool.

Depending on the output specified in the header of the Rmd file, it converts the file to the desired output format. For example if we choose **html_document**, we get a HTML document. If we choose **pdf_document** , pandoc converts to TeX formatting, which is converted by LaTeX to a PDF document.

The Pandoc software comes prepackaged with Rstudio, but to use Latex you needed to install a so-called latex distribution.

The processes leading to HTML and PDF documents respectively are shown below.



Figure 2: Toolchain

End of Notebook