Raporlama Faslı

oevkaya

7/1/2021

ABC of Rmarkdown

You can use a single R Markdown file to both

- save and execute code
- generate high quality reports that can be shared with an audience
- a plain text file that has the extension .Rmd

file contains three types of content

1. An (optional) YAML header surrounded by —s
2. R code chunks surrounded by “‘s
3. text mixed with simple text formatting

To get an output

- render(“filename.Rmd”)
- Knit button

How it works ?

- The .Rmd document is the original format of the document.
- the knit() function in knitr is used to execute all code embedded within the .Rmd file, and prepare the code output to be displayed within the output document. All these results are converted into the correct markup language to be contained within the temporary .md file.
- Then the .md file is processed by Pandoc, a multipurpose tool designed to convert files from one markup language to another. It takes any parameters specified within the YAML frontmatter of the document (e.g., title, author, and date) to convert the document to the output format specified in the output parameter.
- If the output format is PDF, there is an additional layer of processing, as Pandoc will convert the intermediate .md file into an intermediate .tex file. The rmarkdown package calls the latexmk() function in the tinytex package, which in turn calls LaTeX to compile .tex to .pdf

How it works summary

In short, rmarkdown::render() = knitr::knit() + Pandoc (+ LaTeX for PDF output only)

link

Nice Summary

How it works ?

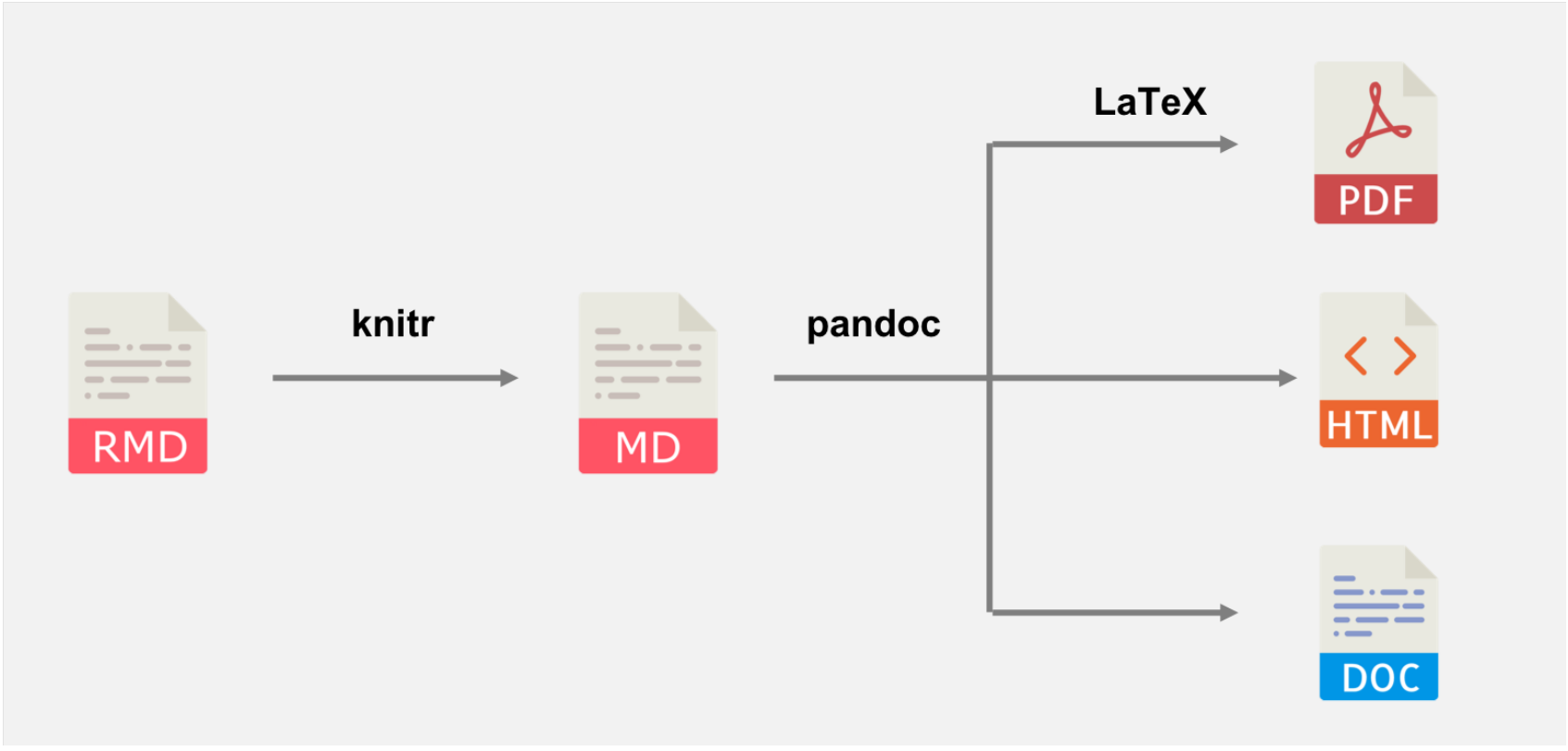


Figure 1: Process

Materials

For reading;

1. Yihui's Rmarkdown book
2. Yihui's Bookdown book

Overview for the components;

- YAML header

A typical YAML header contains basic metadata about the document and rendering instructions

- Body:
  - Text
  - Code chunks

YAML header

- Set title, author,. . .
- Set options for each output-format

Example:

title: “Rmarkdown” author: “Kirill” date: “2020-11-16” output: html_document: toc: true theme: readable

About Syntax simply

Classic text

italics and *italics*

bold and **bold**

superscript^2

~~strikethrough~~

For more details;

About syntax

About writing text

- Markup using (expanded) markdown-syntax
- Sections/subsections/... are created using #/##/...
- Math can be added using latex-code when put between dollar-signs (examples)
- If the output is a pdf, other latex-code (e.g. nnewpage) can be included (will be ignored if other output-format)

ABC of Code Chunks

You can insert chunks like these into your file with;

1. the keyboard shortcut Ctrl + Alt + I (OS X: Cmd + Option + I)
2. the Add Chunk command in the editor toolbar
3. by typing the chunk delimiters `{r}` `and`

Code Chunks

- Contain R (python/sql/...)-code and print output

```
print("hello R markdown")
```

```
## [1] "hello R markdown"
```

```
list = list(c(1,2,3,4))
list
```

```
## [[1]]
## [1] 1 2 3 4
```

Code Chunks

```
print("hello R markdown")
```

```
## hello R markdown
```

```
list  = [1,2,4]
print(list)
```

```
## [1, 2, 4]
```

More details about reticulate

About code chunks

Use knitr options to style the output of a chunk. Place options in brackets above the chunk

- echo: print code itself as well
- eval: do/don't actually evaluate code
- include: evaluate code, but do/don't include the output
- warning: Whether to display warnings
- error: Whether to display errors
- message: Whether to display messages
- fig.cap = "..." adds a caption to graphical results
- cache: 'save/remember' the output so it doesn't have to be recalculated on a next run, except when the code of the chunk has changed. Be aware, if the input of the chunk changes, but not the chunk itself, the chunk will not be re-evaluated and the output is not updated

About code chunks

```
x <- 5  # radius of a circle
pi * x^2 # area of circle
```

```
## [1] 78.53982
```

Inline code

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

- display the results of inline code, but not the code
- apply relevant text formatting to the results

Inline code

```
x <- 5  # radius of a circle
```

For a circle with the radius 5, its area is 78.5398163.

Output formats

- documents:
  - Word
  - html
  - pdf
- presentations:
  - powerpoint
  - beamer (pdf)
  - ioslides (html)

Notes about output

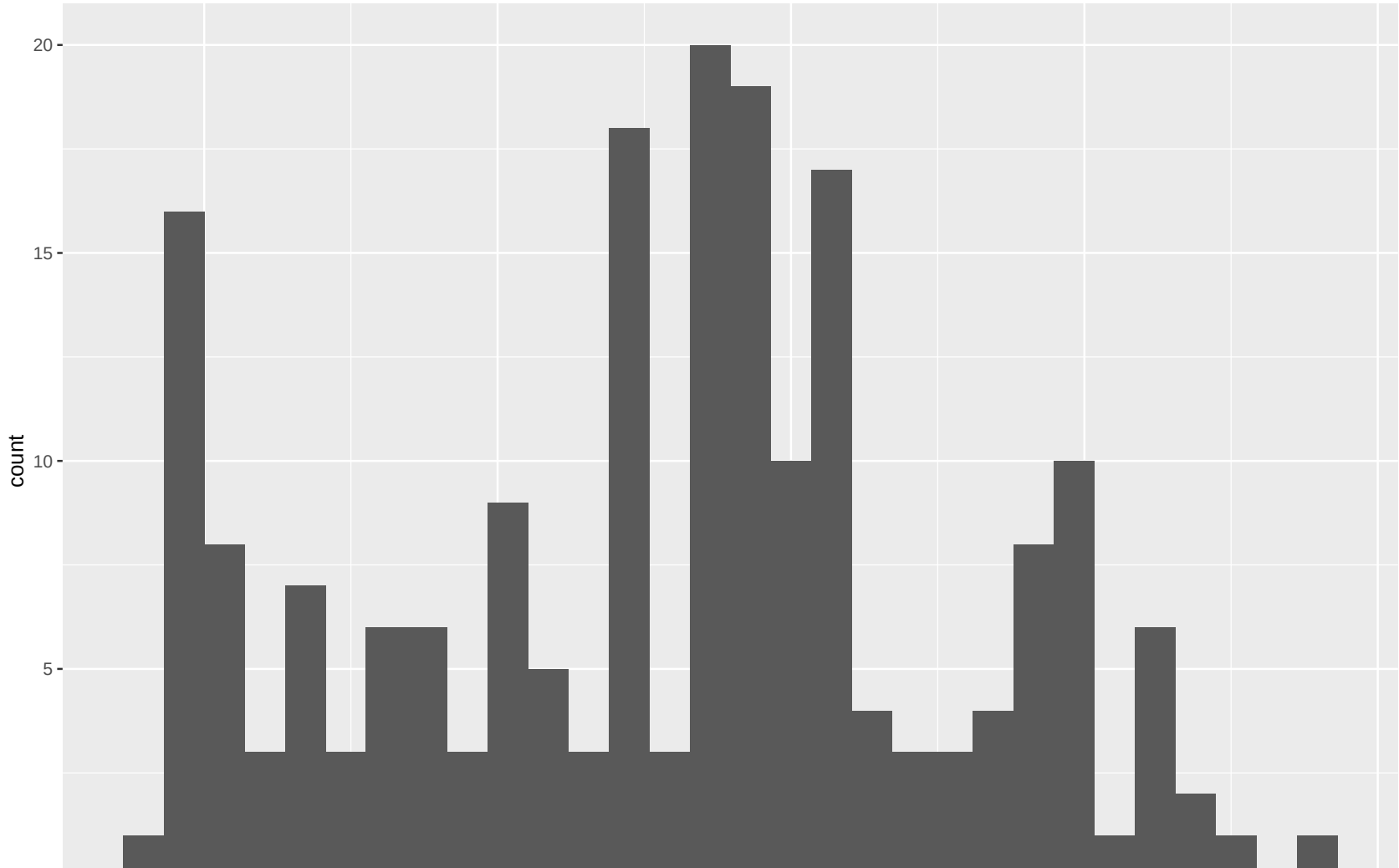
- Word: Implementation is far from ideal. It works, but changing layout is a hassle and many things cannot be performed within rmarkdown. Only solution: change after conversion. After creating via rmarkdown, one can change the format and save it in the folder where it is necessary to use it with

  include in yaml: template: newtemplate.docx

- If you know html, almost everything is possible. You can use several themes or add plain html.
- Rendering rmarkdown to pdf, uses latex. You need an installation of Miktex, TexLive or TinyTex

Images/Figures

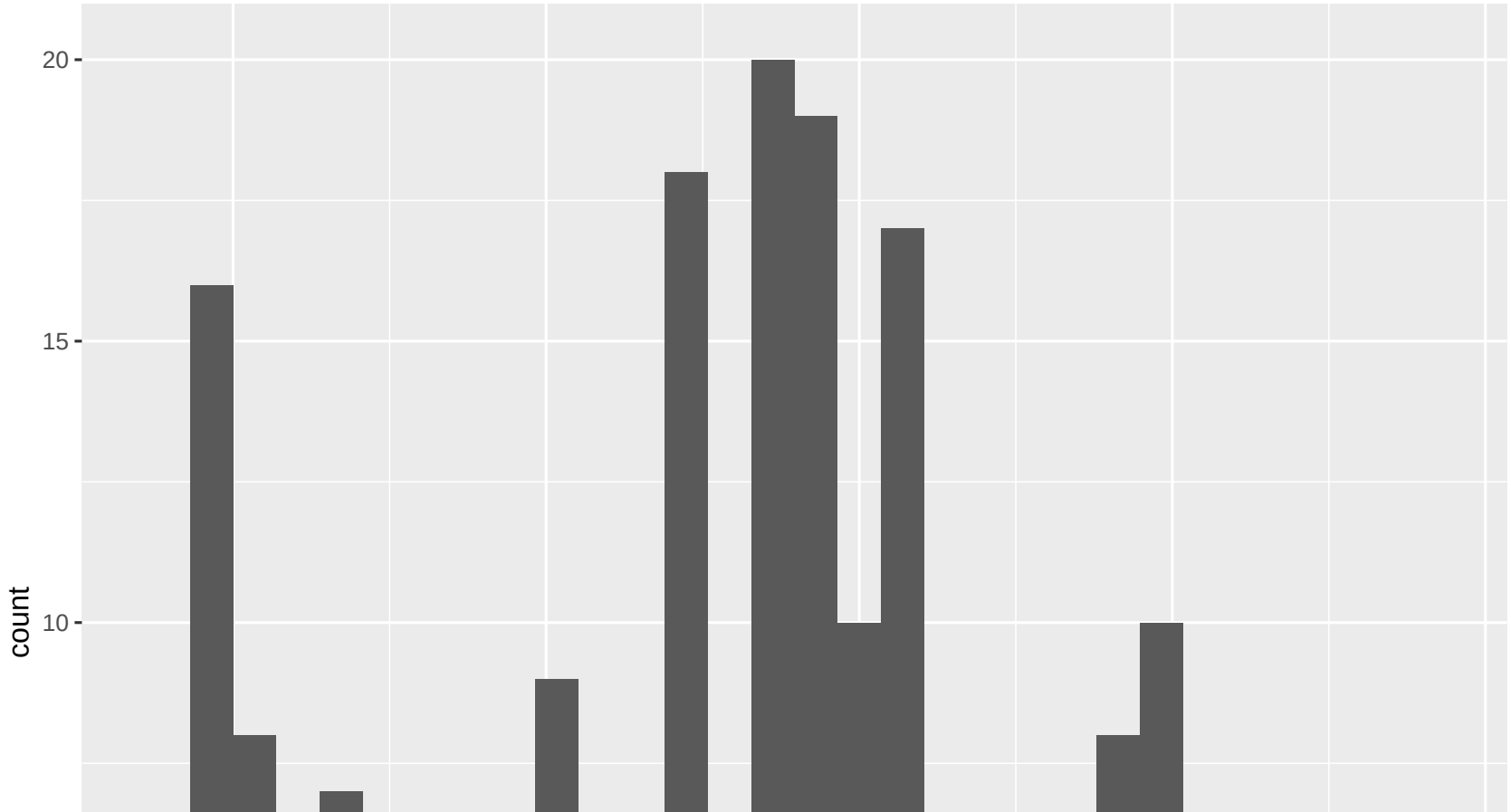
```
library(ggplot2)
ggplot(data, aes(x = AGE)) +
  geom_histogram()
```



Customization on it

```
library(ggplot2)
ggplot(data, aes(x = AGE)) +
  geom_histogram()
```

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



External Figures

```
knitr::include_graphics("flow.png")
```

External Figures

```
knitr::include_graphics("flow.png")
```

Images via url

```
download.file(url = "https://d33wubrfki0l68.cloudfront.net/
              destfile = "flow.png", mode = 'wb')
knitr::include_graphics(path = "flow.png")
```

About Tables

Basically you can create as;

Table Header	Second Header
Cell 1	Cell 2
Cell 3	Cell 4

Header	
20 34	23 34
31 35	27 49
30 52	37 20
55 20	42 39
46 38	38 27

- R Markdown displays data frames and matrixes as they would be in the R terminal (in a monospaced font).
- If you prefer that data be displayed with additional formatting

Packages for Tables

- knitr (GITHUB, WEBSITE)
- kableExtra (GITHUB, WEBSITE)
- formattable (GITHUB, WEBSITE)
- DT (GITHUB, WEBSITE)
- pander (GITHUB, WEBSITE)
- huxtable (GITHUB, WEBSITE)
- reactable (GITHUB, WEBSITE)
- flextable (GITHUB, WEBSITE)
- ftextra (GITHUB, WEBSITE)
- pixiedust (GITHUB)
- tangram (GITHUB)
- ztable (GITHUB)
- condformat (GITHUB)
- stargazer (CRAN)
- xtable (CRAN)
- gt (WEBSITE)

About Tables

Prepare a data.frame with all the desired info to be printed

1. If a simple table:

- pander(df): similar look as kable, but better defaults and basic-options

2. If a complex table:

- use kableExtra: many options (not all available for pdf and html and word)

Table example

```
data(iris)
# head(iris)

library(pander)
pander::pander(head(iris))
```

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa

Table example

```
data(iris)
library(kableExtra)
kable(head(iris), booktabs = TRUE)
```

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa

Use of Parameters

- R Markdown documents can include one or more parameters whose values can be set when you render the report
- Parameters are made available within the knit environment as a read-only list named params. To access a parameter in code, call params$
- Add a params argument to render to create a report that uses a new set of parameter values For instance, render("filename.Rmd", params = list(data = "value"))
- Better yet, click the "Knit with Parameters" option in the dropdown menu next to the RStudio IDE knit button to set parameters, render, and preview the report in a single user friendly step

Advantages

Parameters are useful when you want to re-render the same report with distinct values for various key inputs, for example:

- Running a report specific to a department or geographic region.
- Running a report that covers a specific period in time.
- Running multiple versions of a report for distinct sets of core assumptions.

About bibliography

- Store bibliography in any of the approved formats (.bib, .xml, .json, etc)
- include in yaml: bibliography: biblio.bib
- cite in text using [@ref]
- you can add the citr-addin in RStudio to simplify the matter. For this you need to use citr package

For more details

Available Formats

The bibliography may have any of these formats:

- CSL-JSON: .json
- MODS: .mods
- BibLaTeX: .bib
- BibTeX: .bibtex
- RIS: .ris
- EndNote: .enl
- EndNote XML: .xml
- ISI: .wos
- MEDLINE: .medline
- Copac: .copac

Or

Alternatively you can use a references field in the document's YAML metadata. This should include an array of YAML-encoded references

Some Features

It is possible to extract only codes from the .Rmd file.

- When you want to extract all R code from an R Markdown document, you can call the function knitr::purl()

Usually chunk options take constant values (e.g., fig.width $= 6$), but they can actually take values from arbitrary R expressions, no matter how simple or complicated the expressions are

```
my_width <- 3
```

```
plot(cars)
```