Dynamic documents

Didzis Elferts

Reproducible (dynamic) documents

R markdown

- In R (not only), you can create specific documents (literate programming) that contain both human-friendly text and command lines that can be understood and executed by the program.

- The R markdown language is one of the following:

- In RStudio creating such documents is already integrated with files with an extension. Rmd

- markdown is responsible for the design of the text but knitr package for the codes to be executed, format changes are made by pandoc

Benefits of R markdown

- Light syntax

- The source document is easy to read

- All of the design is visible

- Calculation codes included

Text formating

italic *italic text*

bold **bold text**

Heading:

Level 1

Level 2

Level 3

Lists

Simple lists

* Item 1

* Item 2

```
- Item 2a

- Item 2b
```

Numeric list

1. Item 1

2. Item 2

Document heading

Each R Markdown document begins with the YAML metadata part, e.g.

```
---
title: "This is document"
author: "The Writer"
output:
  ioslides_presentation:
    css: style.css
    highlight: tango
    widescreen: yes
---
```

R codes

- As program codes are perceived only by command lines included in special "signs", the rest is text only

- Included code consists of code designation and code final term

```r
```{r}
x <- 1:10
plot(x)
```
```

Hiden codes

When you include an argument **echo = FALSE**, only the result appears without showing the code itself in the final document.

```r
```{r, echo = FALSE}
x <- 1:10
plot(x)
```
```

Codes that are not executed

Argument **eval = FALSE** ensures that the code is entered in the document but is not executed.

```
```{r, eval = FALSE}
x <- 1:10
plot(x)
```
```

Messages and warnings

You can hide/display warnings and messages displayed by separate r functions or packages with arguments:

- `message = TRUE` or `message = FALSE`.

- `warning = TRUE` or `warning = FALSE`.

Figure parameters

You can resize each figure individually, add captions, etc.

```r
```{r, fig.width = 8, fig.height = 4.5,
          fig.cap = "Caption text"}
x <- 1:10
plot(x)
```
```

Tables

Tables can be made using | sign

one | Two | Three
- - -|- - -|- - -
12 | 56 | 89

One	Two	Three
12	56	89

R objects as tables

To get a more "beautiful" table design, you can use the function **kable** from package **knitr**, adding the argument **results = "asis"** to the beginning of the r code.

```
knitr::kable(head(cars))
```

speed	dist
4	2
4	10
7	4
7	22
8	16
9	10

Additional packages

All additional packages must be added to the document because each document is executed in its "environment". The same applies to all work files and external codes.

Saving results

By default, when you run the Knit command once again, all calculations are done again. If this requires too much time, it may be stipulated that the results should be maintained and not recalculated if the conditions do not change.

The following rows of commands must be included at the top of the file.

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

R codes

- The R code and its result can also be included in the text line using one apostrophe at the beginning and end and a lowercase r.

- The mtcars file contains 32 lines.

Document types

With R markdown, you can create different types of documents and presentation, respectively as `output:` specifying:

- pdf_document - pdf document (TEX system required on your computer to create them)
- html_document - html document
- word_document - docx document
- slidy_presentation - slidy presentation
- ioslides_presentation - ioslides presentation
- beamer_presentation - LATEX beamer presentation (TEX system required to build on your computer)
- different formats offered by additional R packages

Special symbols

To be able to create texts with special symbols, the output file (.Rmd) preferably in UTF-8 encoding.

You must use the xelatex system when creating pdf documents.

Templates for documents

You can create style patterns for Word documents and html documents that you can use when creating new documents. You change styles for html documents with css help, create a sample for word documents as a .docx document.

The following line must then be specified in the metadata:

```
reference_docx: style.docx
```

R markdown

You can find more information about R markdown features on the website of RStudio: http://rmarkdown.rstudio.com/

Book about R markdown: https://bookdown.org/yihui/rmarkdown/

Additional possibilities

- Making blogs with blogdown package:
  https://bookdown.org/yihui/blogdown/

- Making books, reports with bookdown package:
  https://bookdown.org/yihui/bookdown/

Questions?