

# **R MARKDOWN AND AUTOMATIC REPORTING**

Tobias Niedermaier

# **AUTOMATIC REPORTING**

# STANDARD WAY

1. Run analysis code - Generate graphics - Generate tables - Other quantities (single numbers or summary statistics)
2. Save it
3. Import it into you report script
4. Something changed due to reasons...
  - data changed
  - forgot something
  - style changes
  - .... and repeat 1-3

# PROS / CONS

## PROS

- No need to learn new tools
- Post processing of graphics possible

## CONS

- Time consuming
- Error prone
- Reproducibility questionable

# R MARKDOWN

# DYNAMIC REPORTING WITH R MARKDOWN

- Combine report and analysis in one document:
  - Write report in an `.Rmd` file
  - Integrate code to produce results *in the same* document
  - Generate report from the document
- R Markdown combines Markdown (`.md`) with the R code
- Markdown is a simple markup language
- Done by the R package `knitr`: Runs R code and integrates the result automatically in a markdown document

# HOW TO DO?

1. Open new RMarkdown document
2. Write content
3. Render the document

Rmd files contain 3 parts:

- YAML header
- Markdown text
- R chunks

# YAML HEADER

- The header defines 'global options' such as
  - Title
  - Author
  - Date
  - Output format (e.g. pdf\_document, html\_document, word\_document)
  - Style



# MARKDOWN TEXT

- Write text using markdown syntax:

```
1 # Header 1
2 ## Header 2
3 ### Header 3
```

Note the space after #!

```
1 Backticks for `Code`
2 *Text in italics*
3 **Text in bold**
4 ***Text in italics and bold***
```

Special symbols require a `\` in front, e.g. `\#` for `#`

Math can be included using `$` inline or `$$` for math block.

- e.g. `$_alpha$` will get  $\alpha$
- e.g. the following

```
$$
\sum_{i = 1}^N X_i
```

\$\$

will get

$$\sum_{i=1}^N X_i$$

# CODE CHUNKS

- Code chunks contain the code, that we want to use in the markdown document.
- Either use the button, **Ctrl+Alt+I** or write it by wrapping code using backticks
- Code chunks look like this:

```
```${r}
```

R code here

```
```
```

- You can add further arguments in the code chunk header or a name, e.g.

```
```${r Figure 1, eval=TRUE, echo=FALSE}
```

R code here

```

- We can inline code evaluation using only one backtick and `r` like this:

``r 1 + 1`` will just evaluated to 2

Look at the cheat sheets on the web, e.g.

- [this](#) for general markdown or
- [this](#) for R Markdown

# **EXCERCISE 4 TASK 1 AND 2**