Reproducible Research

A Workflow using R / knitr / RStudio

Detlef Steuer, Helmut-Schmidt-Universität Hamburg,
steuer@hsu-hh.de

Kolding, 29. September 2016

Plan for the day

- Part 1: Overview
  - What should be achieved?
  - What can be achieved?
  - Small examples to type for yourself
- Part 2: Hands-On Course
  - Some real data, and a real result to save some money (at least for germans)

Some questions

- Whats your usage of R?
- RStudio installed?
- Miktex (or similar) installed?

Target audience

- (You are familiar with R)
- Everybody who produces publication from and containing data.
- Everybody who has interest to find new partners to work on some data.
- Questions: as soon as they come up!
- Course download `http://fawn.hsu-hh.de/~steuer/downloads/Kolding/kurs-kolding.pdf`

What is Reproducible Research?

The roots of RepRes go back to the notion of *literate programming* by Donald Knuth around 1980.

```
"Programs are meant to be read by humans, and
 only incidentally for computers to execute."
```

In principle this is about giving **complete** analyses to future readers. This is made possible by making available

1. **all** the data,
2. **all** the steps of analysis,
3. with **all** the documentation to recipients (reader, editor).

In an ideal world even 4. the **computational environment** is part of the analysis!

Why should I practice Reproducible Research?

- Non-reproducible single occurrences are of no significance to science.

Karl Popper

Why should I practice Reproducible Research?

- Non-reproducible single occurrences are of no significance to science.

Karl Popper

- In 2012 a study bei Begley and Ellis was published in Nature, which re-examined a decade of cancer research. They found that 47 of 53 results in those papers could not be reproduced.

Why should I practice Reproducible Research?

- Non-reproducible single occurrences are of no significance to science.

Karl Popper

- In 2012 a study bei Begley and Ellis was published in Nature, which re-examined a decade of cancer research. They found that 47 of 53 results in those papers could not be reproduced.
- A symposium in the UK found half of all results may be wrong in medicine, psychology and other fields. (Horton, 2015, Lancet)

Why should I practice Reproducible Research?

- Non-reproducible single occurrences are of no significance to science.

Karl Popper

- In 2012 a study bei Begley and Ellis was published in Nature, which re-examined a decade of cancer research. They found that 47 of 53 results in those papers could not be reproduced.
- A symposium in the UK found half of all results may be wrong in medicine, psychology and other fields. (Horton, 2015, Lancet)
- There is a real crisis of our beloved scientific method.

Why should I practice Reproducible Research?

- Non-reproducible single occurrences are of no significance to science.

Karl Popper

- In 2012 a study bei Begley and Ellis was published in Nature, which re-examined a decade of cancer research. They found that 47 of 53 results in those papers could not be reproduced.
- A symposium in the UK found half of all results may be wrong in medicine, psychology and other fields. (Horton, 2015, Lancet)
- There is a real crisis of our beloved scientific method.
- In pharmacology nowadays reproducibilty is key for allowing new pharmacies on the market.

Why should I practice Reproducible Research?

- Non-reproducible single occurrences are of no significance to science.

Karl Popper

- In 2012 a study bei Begley and Ellis was published in Nature, which re-examined a decade of cancer research. They found that 47 of 53 results in those papers could not be reproduced.
- A symposium in the UK found half of all results may be wrong in medicine, psychology and other fields. (Horton, 2015, Lancet)
- There is a real crisis of our beloved scientific method.
- In pharmacology nowadays reproducibilty is key for allowing new pharmacies on the market.
- Practical point of view: **You** should at least be able to reproduce your results **exactly** and **easily**.

Two different kinds of “doing it again”

- Replicability und Reproducibility

Two different kinds of “doing it again”

- Replicability und Reproducibility
- Replicability: Repetition of an experiment leads to the same result (modulo noise)

Two different kinds of “doing it again”

- Replicability und Reproducibility

- Replicability: Repetition of an experiment leads to the same result (modulo noise)
- Reproducibility: Given data and the documentation a third party can replay the whole analysis ang get identical results!

Two different kinds of “doing it again”

- Replicability und Reproducibility
- Replicability: Repetition of an experiment leads to the same result (modulo noise)
- Reproducibility: Given data and the documentation a third party can replay the whole analysis ang get identical results!
- Today: Reproducibility

Two different kinds of “doing it again”

- Replicability und Reproducibility

- Replicability: Repetition of an experiment leads to the same result (modulo noise)
- Reproducibility: Given data and the documentation a third party can replay the whole analysis ang get identical results!

- Today: Reproducibility
- RepRes is shorter and easier to pronounce than Reproducible Research

Two different kinds of “doing it again”

- Replicability und Reproducibility

- Replicability: Repetition of an experiment leads to the same result (modulo noise)
- Reproducibility: Given data and the documentation a third party can replay the whole analysis ang get identical results!

- Today: Reproducibility
- RepRes is shorter and easier to pronounce than Reproducible Research

- Already (very) relevant in some research areas.

Two different kinds of "doing it again"

- Replicability und Reproducibility

- Replicability: Repetition of an experiment leads to the same result (modulo noise)
- Reproducibility: Given data and the documentation a third party can replay the whole analysis ang get identical results!

- Today: Reproducibility
- RepRes is shorter and easier to pronounce than Reproducible Research

- Already (very) relevant in some research areas.
- Biometrics Journal has an editor especially for RepRes Authors must provide all the data and all the programs, so the editor can replay the generation of all results.

Why doesn't everybody do RepRes already?

- Time pressure

Why doesn't everybody do RepRes already?

- Time pressure
- Data (and analyses) in Excel sheets

Why doesn't everybody do RepRes already?

- Time pressure
- Data (and analyses) in Excel sheets
- Point and Klick Interfaces (SPSS)

Why doesn't everybody do RepRes already?

- Time pressure
- Data (and analyses) in Excel sheets
- Point and Klick Interfaces (SPSS)
- Propriatary data formats (STATA, SAS)

Why doesn't everybody do RepRes already?

- Time pressure
- Data (and analyses) in Excel sheets
- Point and Klick Interfaces (SPSS)
- Propriatary data formats (STATA, SAS)
- Proprietäre file formats for the reports (.docx)

Why doesn't everybody do RepRes already?

- Time pressure
- Data (and analyses) in Excel sheets
- Point and Klick Interfaces (SPSS)
- Propriatary data formats (STATA, SAS)
- Proprietäre file formats for the reports (.docx)
- Missing attractive tool chain to change the workflow in the direction of reproducibility.

How to bring RepRes to a scientists day-to-day job

- It has to be simple! Better even: simpler! It must be more fun to work that way, than working with propriatary formats.

How to bring RepRes to a scientists day-to-day job

- It has to be simple! Better even: simpler! It must be more fun to work that way, than working with propriatary formats.
- Advantages must be obvious!

What to consider while choosing tools?

Requirements (and direct consequences)

- Each and every file (with data) used and produced during analysis must be human readable. (text files!) Avoiding vendor lock-in (MS Office, SAP, STATA)

What to consider while choosing tools?

Requirements (and direct consequences)

- Each and every file (with data) used and produced during analysis must be human readable. (text files!) Avoiding vendor lock-in (MS Office, SAP, STATA)
- File format i.e. .csv or similar

What to consider while choosing tools?

Requirements (and direct consequences)

- Each and every file (with data) used and produced during analysis must be human readable. (text files!) Avoiding vendor lock-in (MS Office, SAP, STATA)
- File format i.e. .csv or similar
- Databases: save SQL queries as texts! Databases must be frozen (conflicting with real time analyses)!

What to consider while choosing tools?

Requirements (and direct consequences)

- Each and every file (with data) used and produced during analysis must be human readable. (text files!) Avoiding vendor lock-in (MS Office, SAP, STATA)
- File format i.e. .csv or similar
- Databases: save SQL queries as texts! Databases must be frozen (conflicting with real time analyses)!
- If in doubt, save a dump! Or an SQL query that reproduces the state at the start point of analyses.

What to consider while choosing tools?

Requirements (and direct consequences)

- Each and every file (with data) used and produced during analysis must be human readable. (text files!) Avoiding vendor lock-in (MS Office, SAP, STATA)
- File format i.e. .csv or similar
- Databases: save SQL queries as texts! Databases must be frozen (conflicting with real time analyses)!
- If in doubt, save a dump! Or an SQL query that reproduces the state at the start point of analyses.
- Code and results are saved as textfiles (with optional markup)

What to consider while choosing tools?

Requirements (and direct consequences)

- Each and every file (with data) used and produced during analysis must be human readable. (text files!) Avoiding vendor lock-in (MS Office, SAP, STATA)
- File format i.e. .csv or similar
- Databases: save SQL queries as texts! Databases must be frozen (conflicting with real time analyses)!
- If in doubt, save a dump! Or an SQL query that reproduces the state at the start point of analyses.
- Code and results are saved as textfiles (with optional markup)
- Graphics are saved as code that produces them!

What to consider while choosing tools?

Requirements (and direct consequences)

- Gandrud (2014): All file belonging to an analyses (data, graphics, tables, code, text) must be connected *explicitly*! That is, it must be more than a collection of files containing all you need! It must be a collection of files connected, i.e. by a makefile or by some other programmatical way, so that if data changes in one file all analyses reflects that change (semi-)automatically. Dependencies must be made explicit!

Consequences!

- We can not use the usual office tools. Excel sheets are not transparent!

Consequences!

- We can not use the usual office tools. Excel sheets are not transparent!
- If a reviewer has to decide if an article gives valueable results, how can she do it, if the the software has restrictive and/or expensive licenses? (Matlab, SAS, Mathematica) But it should be a requirement, that the reviewer can *review* how results are derived.

Consequences!

- We can not use the usual office tools. Excel sheets are not transparent!
- If a reviewer has to decide if an article gives valueable results, how can she do it, if the the software has restrictive and/or expensive licenses? (Matlab, SAS, Mathematica) But it should be a requirement, that the reviewer can *review* how results are derived.
- What about old versions of Software? I.e. STATA did in the past change its file format in an incompatible way. The same happend to some bibliography software used at our department. Suddenly old versions of important data had lost all their value.

Consequences!

- We can not use the usual office tools. Excel sheets are not transparent!
- If a reviewer has to decide if an article gives valueable results, how can she do it, if the the software has restrictive and/or expensive licenses? (Matlab, SAS, Mathematica) But it should be a requirement, that the reviewer can *review* how results are derived.
- What about old versions of Software? I.e. STATA did in the past change its file format in an incompatible way. The same happend to some bibliography software used at our department. Suddenly old versions of important data had lost all their value.
- In a very natural way that leads to OpensSource or at least "free as in free beer" Software.

Consequences!

- We can not use the usual office tools. Excel sheets are not transparent!
- If a reviewer has to decide if an article gives valueable results, how can she do it, if the the software has restrictive and/or expensive licenses? (Matlab, SAS, Mathematica) But it should be a requirement, that the reviewer can *review* how results are derived.
- What about old versions of Software? I.e. STATA did in the past change its file format in an incompatible way. The same happend to some bibliography software used at our department. Suddenly old versions of important data had lost all their value.
- In a very natural way that leads to OpensSource or at least "free as in free beer" Software.
- RStudio is a perfect example for the power of OpenSource and their licenses! In principle RStudio is a great inferface to a big, big bundle of FOSS!

What won't be discussed?

- How to conserve your computational environment? Vmware? Docker? Hardware storing?

What won't be discussed?

- How to conserve your computational environment? Vmware? Docker? Hardware storing?
- How to conserve data? How can the integrity of an extracted data set be assured?

What won't be discussed?

- How to conserve your computational environment? Vmware? Docker? Hardware storing?
- How to conserve data? How can the integrity of an extracted data set be assured?
- Revision Control Systems: svn, git etc. Extremely useful, but too much for such a one-day-course. (use git!)

The Tools (standing on the shoulders of giants)

What do we have?

- TeX bzw. LaTeX (Donald Knuth)

The Tools (standing on the shoulders of giants)

What do we have?

- TeX bzw. LaTeX (Donald Knuth)
- Markdown bzw. RMarkdown
  (John Gruber and Aaron Swartz)
  A markdown-formated document should be publishable as-is, as plain text, without looking like it's been marked up with tags or formatting instructions. – John Gruber

The Tools (standing on the shoulders of giants)

What do we have?

- TeX bzw. LaTeX (Donald Knuth)
- Markdown bzw. RMarkdown
  (John Gruber and Aaron Swartz)
  A markdown-formated document should be publishable as-is, as plain text, without looking like it's been marked up with tags or formatting instructions. – John Gruber
- pandoc (John MacFarlane)

The Tools (standing on the shoulders of giants)

What do we have?

- TeX bzw. LaTeX (Donald Knuth)
- Markdown bzw. RMarkdown
  (John Gruber and Aaron Swartz)
  A markdown-formated document should be publishable as-is, as plain text, without looking like it's been marked up with tags or formatting instructions. – John Gruber
- pandoc (John MacFarlane)
- R (The R-core team)

The Tools (standing on the shoulders of giants)

What do we have?

- TeX bzw. LaTeX (Donald Knuth)
- Markdown bzw. RMarkdown
  (John Gruber and Aaron Swartz)
  A markdown-formated document should be publishable as-is, as plain text, without looking like it's been marked up with tags or formatting instructions. – John Gruber
- pandoc (John MacFarlane)
- R (The R-core team)
- RStudio (Hadley Wickham, ggplot, dplyr, devtools etc.)

The Tools (standing on the shoulders of giants)

What do we have?

- TeX bzw. LaTeX (Donald Knuth)
- Markdown bzw. RMarkdown
  (John Gruber and Aaron Swartz)
  A markdown-formated document should be publishable as-is, as plain text, without looking like it's been marked up with tags or formatting instructions. – John Gruber
- pandoc (John MacFarlane)
- R (The R-core team)
- RStudio (Hadley Wickham, ggplot, dplyr, devtools etc.)
- knitr (Yihui Xie, now Rstudio, too)

The Tools (standing on the shoulders of giants)

What do we have?

- TeX bzw. LaTeX (Donald Knuth)
- Markdown bzw. RMarkdown
  (John Gruber and Aaron Swartz)
  A markdown-formated document should be publishable as-is, as plain text, without looking like it's been marked up with tags or formatting instructions. – John Gruber
- pandoc (John MacFarlane)
- R (The R-core team)
- RStudio (Hadley Wickham, ggplot, dplyr, devtools etc.)
- knitr (Yihui Xie, now Rstudio, too)
- SWeave (Fritz Leisch)

The Tools (standing on the shoulders of giants)

What do we have?

- TeX bzw. LaTeX (Donald Knuth)
- Markdown bzw. RMarkdown
  (John Gruber and Aaron Swartz)
  A markdown-formated document should be publishable as-is, as plain text, without looking like it's been marked up with tags or formatting instructions. – John Gruber
- pandoc (John MacFarlane)
- R (The R-core team)
- RStudio (Hadley Wickham, ggplot, dplyr, devtools etc.)
- knitr (Yihui Xie, now Rstudio, too)
- SWeave (Fritz Leisch)
- git (Linus Torvalds)

The aim

- A workflow, that doesn't hinder scientific work

The aim

- A workflow, that doesn't hinder scientific work
- An evironment should be created, that follows all the requirements of RepRes and that results as an output in reports in various output formats like HTML, PDF or docx.

The aim

- A workflow, that doesn't hinder scientific work
- An evironment should be created, that follows all the requirements of RepRes and that results as an output in reports in various output formats like HTML, PDF or docx.
- Very important: the tools must have a very low overhead. It must feel natural to use the tools for the job.

Steps in quantitative work

- Collection data, and preparing data for work

Steps in quantitative work

- Collection data, and preparing data for work
- Part of this is having a plan how to share the date with partners, especially future partners! (homepage, cloud . . .)

Steps in quantitative work

- Collection data, and preparing data for work
- Part of this is having a plan how to share the date with partners, especially future partners! (homepage, cloud . . .)
- Being afraid of transparency is bad science!

Steps in quantitative work

- Collection data, and preparing data for work
- Part of this is having a plan how to share the date with partners, especially future partners! (homepage, cloud . . .)
- Being afraid of transparency is bad science!

- Analysis

Steps in quantitative work

- Collection data, and preparing data for work
- Part of this is having a plan how to share the date with partners, especially future partners! (homepage, cloud . . .)
- Being afraid of transparency is bad science!

- Analysis
- Presentation

Steps in quantitative work

- Collection data, and preparing data for work
- Part of this is having a plan how to share the date with partners, especially future partners! (homepage, cloud . . .)
- Being afraid of transparency is bad science!

- Analysis
- Presentation

- Remember: All three steps must be reproducible! If done correctly the last two steps coincide!

Some history

- Literate Programming (1979, Donald Knuth) “Let us change our traditional attitude to the construction of programs: Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to humans what we want the computer to do.”

`Donald E. Knuth, Literate Programming, 1984`

Some history

- Literate Programming (1979, Donald Knuth) “Let us change our traditional attitude to the construction of programs: Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to humans what we want the computer to do.”

`Donald E. Knuth, Literate Programming, 1984`

- Here **Literate Data Analysis!**

Some history

- Literate Programming (1979, Donald Knuth) "Let us change our traditional attitude to the construction of programs: Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to humans what we want the computer to do."

`Donald E. Knuth, Literate Programming, 1984`

- Here **Literate Data Analysis!**
- Sweave (2002, Fritz Leisch)

Some history

- Literate Programming (1979, Donald Knuth) "Let us change our traditional attitude to the construction of programs: Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to humans what we want the computer to do."

`Donald E. Knuth, Literate Programming, 1984`

- Here **Literate Data Analysis!**
- Sweave (2002, Fritz Leisch)
- knitr (2011, Yihui Xie)

Some history

- Literate Programming (1979, Donald Knuth) “Let us change our traditional attitude to the construction of programs: Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to humans what we want the computer to do.”

`Donald E. Knuth, Literate Programming, 1984`

- Here **Literate Data Analysis!**
- Sweave (2002, Fritz Leisch)
- knitr (2011, Yihui Xie)
- org-mode with babel (2011, Eric Schulte)
  Add-on for emacs. It is possible to combine sources from many different programming languages in one file (i.e. maxima, gnuplot, R, python, C, julia . . .). Exports to LaTeX, ODT, Markdown etc.

The Idea of Literate Data Analysis

We want to have code and results and analysis in one file.
All we need is a syntax and the tools, that can extract and evaluate source code from this *mixed* text and source file and insert the results of the computation back into the same or another file.
When Knuth introduced the idea that kind of execution of the literate program in situ wasn't prepared.
Nowadays we have one starting document containing mixed text and sources which is "weaved" if needed into a result document containing the results of the executed sources.
Practical problems arise from pictures and tables.

Definitions

Code snippets contained in text traditionally are called “chunks”.
Using a special syntax a tool used for weaving learns which parts are to be interpreted as programs.
When Knuth first presented his idea the syntax looked as follows:

```
surrounding text
<<opt. chunkname >>=
code
@
surrounding text
```

This is still a possible syntax, even for R, but there more readerfriendly ways.

RStudio - An OS for RepRes

Bringing all parts to work

- Upper left area shows the main document. What is written there will constitute the report.

RStudio - An OS for RepRes

Bringing all parts to work

- Upper left area shows the main document. What is written there will constitute the report.
- Lower left area ist some scratch pad. A plain R console.

RStudio - An OS for RepRes

Bringing all parts to work

- Upper left area shows the main document. What is written there will constitute the report.
- Lower left area ist some scratch pad. A plain R console.
- For what we do RStudio in not essential in a sense. Everything could be made working with plain R and a lot of tools. You could do the same with just an R console. RStudio offers a ready made setup to start the work. I.e. pandoc is not so easy to setup.

RStudio - An OS for RepRes

Bringing all parts to work

- Upper left area shows the main document. What is written there will constitute the report.
- Lower left area ist some scratch pad. A plain R console.
- For what we do RStudio in not essential in a sense. Everything could be made working with plain R and a lot of tools. You could do the same with just an R console. RStudio offers a ready made setup to start the work. I.e. pandoc is not so easy to setup.
- In that sense RStudio solved a **huge** problem!

First variant for short and quick jobs (spinning)

- File -> New R Script

First variant for short and quick jobs (spinning)

- File -> New R Script
- File -> Compile Notebook

First variant for short and quick jobs (spinning)

- File -> New R Script
- File -> Compile Notebook
- Perfect format for excercises (rpub.com).

First variant for short and quick jobs (spinning)

- File -> New R Script
- File -> Compile Notebook
- Perfect format for excercises (rpub.com).
- In this case we have mainly an R program that contains some text, not a text with a bit of code.

First variant for short and quick jobs (spinning)

- File -> New R Script
- File -> Compile Notebook
- Perfect format for excercises (rpub.com).
- In this case we have mainly an R program that contains some text, not a text with a bit of code.
- Comments are added in the form of R comments

First variant for short and quick jobs (spinning)

- File -> New R Script
- File -> Compile Notebook
- Perfect format for excercises (rpub.com).
- In this case we have mainly an R program that contains some text, not a text with a bit of code.
- Comments are added in the form of R comments
- or as roxygen comments in lines starting with #', which already are interpreted.

First variant for short and quick jobs (spinning)

- File -> New R Script
- File -> Compile Notebook
- Perfect format for excercises (rpub.com).
- In this case we have mainly an R program that contains some text, not a text with a bit of code.
- Comments are added in the form of R comments
- or as roxygen comments in lines starting with #', which already are interpreted.
- It is possible to add some header info in a specially formated header (YAML).

Hands-on example

Violin plots

The different possible inputs for knitr

- *.R: R script Format for* spinning*
- *.Rnw (R noweb): Output is LaTeX. Syntax for codeblocks

```
<< >>=
here comes the R code
@
```

- *.Rtex (R TeX): Output is LaTeX, but (nicer) Syntax

```
%%begin.rcode
here comes the R code
%%
```

The different possible inputs for knitr

- *Rhtml: Output is HTML

  ```
  <!--begin.rcode
  Hierher der R-Code
  end.rcode-->
  ```

- *Rmd (Rmarkdown): Output is Markdown. Postprocessing with pandoc. Therefore everything is possible: LaTeX, PDF, Word, etc.

  ```
  ```{ r eval=FALSE}
  n = 10
  rnorm(n)
  ```
  ```

  The blocks start and end with three backticks!

Today: Rmarkdown!

- Rmarkdown is a superset of plain Markdown. It was developed to have a mrakup language that was easily readable in source. At the same time some markup should be available. (headlines, **bold**, *italic*, etc.)

Today: Rmarkdown!

- Rmarkdown is a superset of plain Markdown. It was developed to have a mrakup language that was easily readable in source. At the same time some markup should be available. (headlines, **bold**, *italic*, etc.)
- The source of our report is a Markdown text, which contains chunks written in R.

Today: Rmarkdown!

- Rmarkdown is a superset of plain Markdown. It was developed to have a mrakup language that was easily readable in source. At the same time some markup should be available. (headlines, **bold**, *italic*, etc.)

- The source of our report is a Markdown text, which contains chunks written in R.
- The processing chain is
  text.Rmd -> knitr -> text.md -> pandoc -> html, tex, doc (-> pdflatex -> pdf)

Today: Rmarkdown!

- Rmarkdown is a superset of plain Markdown. It was developed to have a mrakup language that was easily readable in source. At the same time some markup should be available. (headlines, **bold**, *italic*, etc.)

- The source of our report is a Markdown text, which contains chunks written in R.
- The processing chain is
  text.Rmd -> knitr -> text.md -> pandoc -> html, tex, doc (-> pdflatex -> pdf)
- pandoc extends the possibilities of pure markdown. I.e. citations are added, which are nor part of pure Markdown.

Structure of a Markdown document

Header (optional)

```
---
title: Reproducible Research
subtitle: A workflow  mit R / knitr / RStudio
author:  Detlef Steuer
date: Kolding, 29. September 2016
output:
  beamer_presentation:
    toc: true
---
```

- Header is in YAML format.

Structure of a Markdown document

Header (optional)

```
---
title: Reproducible Research
subtitle: A workflow  mit R / knitr / RStudio
author:  Detlef Steuer
date: Kolding, 29. September 2016
output:
  beamer_presentation:
    toc: true
---
```

- Header is in YAML format.
- Many options, special syntax

Structure of a Markdown document

Header (optional)

```
---
title: Reproducible Research
subtitle: A workflow  mit R / knitr / RStudio
author:  Detlef Steuer
date: Kolding, 29. September 2016
output:
  beamer_presentation:
    toc: true
---
```

- Header is in YAML format.
- Many options, special syntax
- Provided by another R package yaml (surprise!) Complete manual availabel on CRAN

Structure of a Markdown document

Text markup in Markdown

````
A Headline!
===========
Some text in **bold** or *italic*
Some subheadline
--------------------
A formula inline in the text $e^{i\pi} = -1$.
A code-chunk:
```{ r chunkname, eval=FALSE}
hist(rnorm(100))
```
Even some inline calculation ist possible:
The iris dataframe contains ` r nrow(iris)` observations.
````

HTML will be passed throug withoout any change, just like LaTeX.

Structure of a Markdown document

More and complete

- Rmarkdown Cheat-Sheet
- Rmarkdown Reference Guide

Chunks and their computation en detail

- Normally chunks are executed one by one in the order they appear in the document, as if they constituted one long R script.

Chunks and their computation en detail

- Normally chunks are executed one by one in the order they appear in the document, as if they constituted one long R script.
- But: there are tons of otions controlling how and when chunks are executed and how their results are used.

Chunks and their computation en detail

- Normally chunks are executed one by one in the order they appear in the document, as if they constituted one long R script.
- But: there are tons of otions controlling how and when chunks are executed and how their results are used.
- The following list is not complete! If you miss something you need, read the knitr documentation! It's probably there!

Useful chunk options

Chunk options will be declared in the curly braces.

Computation Control

- eval = TRUE | FALSE ; codeblock is executed (or not)
- echo = TRUE | FALSE ; source will be containd in final document (or not)
- results = markup | asis | . . . ; kind of syntax used to put results in report
- error: (TRUE; logical) ; doesn't stop in case of an error in one chunk.
- message = TRUE | FALSE : are messages shown. *****

Useful chunk options

Caching

Options for *expensive* calculations. knitr has mechanisms to decide if a codeblock has changed or not. If nothing has changed, the parameter 'cache' can be used to decide that no re-calculation is neccessary.

- cache = TRUE | FALSE (or more fine grained numerically); Results are proteced from useless recalculation.
- dependson: chunkname (NULL; character or numeric) ; In case of cache = TRUE you can define explicit dependencies using dependson. (automatic resolution of knitr is very good!)
- cache.vars: (NULL); only cache variables with given names.

Useful chunk options

Figures

- dev: (‘pdf’ for LaTeX output and ‘png’ for HTML/markdown; character) ; Choosing format of figures, all R devices are possible.
  dev=c(‘pdf’, ‘png’), multiple output formats are possible!
- fig.width, fig.height, out.width, out.height: scales for figures, relativ sclaing is possible, i.e. fig.width=.8\linewidth in LaTeX

Useful chunk options

Sub-documents

- child: (NULL; character) vector of filenames; similar to include

Some further notes

- Options may be specified in the chunk:
  opts_chunk$set(comment=NA, fig.width=6, fig.height=6))
  opts_chunk$set(dev = c("pdf", "jpg"))
- Values for chunks must be given in **one** line. Values must be valid R expressions.
- Do not use points and spaces in filenames! You are only asking for hard to debug trouble.

Tables

- Copying and preparing tables for publication is one of the most boring and therefor most error prone activities **ever**.

Therefore, if possible in any way, let a program generate your tables!

- R packages for nice (publication ready) tables: xtable, stargazer, apsrtable, knitr::kable, pander, htmlTable. See their galleries!
- Important options: “results=‘asis’ ”, “hide”, “markup”

```
library(xtable)
set.seed(17041967)
learnhours <- sample(100:200,30)
result <- learnhours + rbinom(30,30,0.5) -15
reg1 <- lm(result~ learnhours)
nice.table <- xtable(
  reg1,caption="No work, no points")
print.xtable(nice.table,type="latex",comment=FALSE)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-2.9579	2.8494	-1.04	0.3081
learnhours	1.0164	0.0185	54.91	0.0000

Table 1: No work, no points

Tables

Some options are useful to be set globally, i.e.
options("xtable.type" = "html")
If we are using knitr, it is natural to use knitr::kable. Big advantage: knitr::kable chooses the correct output format automagically.

```
knitr::kable(data.frame(
  cbind(learnhours[1:5],result[1:5])))
```

X1	X2
182	184
135	133
176	183
150	153
174	172

Tables from kable are rudimentary. It is programmatically possible to use xtable even if multiple output formats are needed.

Figures

- Most of the time you don't have to do anything. If a code chunk produces a figure, that figure will apear in your report. The dev.args chunk options passes its arguments to the graphics device.
- Useful chunk options: fig.align='center', out.width, out.height
- Independently of the graphics system you just put your usual R command in the chunk. All the neccassary background work will be done by knitr! (temporary files, file names, choosing a device, etc.)
- It is easy to include externally generated figures: ![descriptive text][path/to/picture.png]

Reproducible Simulation Studies

To make reproducibility possible for Monte-Carlo-Studies the command set.seed() is key. Using it your stream of pseudo random numbers becomes reproducible.

```
set.seed (17041967)
mean(runif(100))
```

```
## [1] 0.4951386
```

```
mean(runif(100))
```

```
## [1] 0.5177741
```

```
set.seed(17041967)
mean(runif(100))
```

```
## [1] 0.4951386
```

Literature

With pandoc even bibliographies become a possiblitiy for this simple markup. The YAML header is enriched by a line containing the name of a bibliography file (in bibtex format!):

```
---
title: "Reproducible Research"
author: "Detlef Steuer"
date: "Kolding, 29. September 2016"
output:
  beamer_presentation:
    toc: yes
subtitle: A workflow with R / knitr / RStudio
bibliography: Kurs.bib
---
```

Now, in the text you can cite using:

```
Language of choice: R [@R2014]
Nice book about R programming [@Ligges2005]
RepRes with RSTudio [@Gandrud2014]
Dynamic Documents in R [@Yihui2014]
```

In the report you will see:
Language of choice: R (R Core Team 2014)
Nice book about R programming (Ligges 2005)
RepRes with RStudio (Gandrud 2014) Dynamic Documents in R (Xie 2015)
The bibliography will show up at the end of the document, after the last headline.

Conclusion

- RMarkdown is a lightweight markup language, which, if used in RStudio give a nice headstart into RepRes.1
- In the second part we will construct a simple analysis of gasoline prices in Germany.
- Rtex makes everything possible, that is possible with latex. It therefore has a steeper learning curve.

Literaturverzeichnis

Gandrud, Christopher. 2014. *Reproducible Research with R and Rstudio*. Chapman; Hall.

Ligges, Uwe. 2005. *Programmieren Mit R*. Springer.

R Core Team. 2014. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. `http://www.R-project.org/`.

Xie, Yihui. 2015. *Dynamic Documents with R and Knitr*. 2nd ed. Boca Raton, Florida: Chapman; Hall/CRC. `http://yihui.name/knitr/`.