Reproducible Research

Ein Workflow mit R / knitr / RStudio

Detlef Steuer

Hamburg, 18. September 2015

Kursablauf

- Erster Teil: Überblicksvortrag
  - Was ist angestrebt?
  - Was ist möglich?
  - Kleine Codebeispiele zum Mittippen
- Zweiter Teil: Hands-On Kurs
  - Mit echten Daten wird ein *gewinnbringendes* Ergebnis herausgearbeitet.

Zielgruppe

- Jeder der aus und mit Daten Veröffentlichungen produziert.
- Jeder der Interesse an Zusammenarbeit mit neuen Partnern hat.
- Bitte jederzeit fragen!

(Bitte einloggen: Nutzer: xrere, Passwort: 150918Results)

Was ist Reproducible Research?

Ursprung liegt im *literate programming* von Donald Knuth bereits um 1980.

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

Im Kern geht es darum Analysen in ihrer Gesamtheit einem zukünftigen Rezipienten an die Hand zu geben. Die Wiederholbarkeit wird erreicht indem sowohl die Daten **vollständig** vorhanden sind, als auch die Analyse **vollständig** und **vollständig dokumentiert** vorliegt. Die Analyse umfasst (idealerweise) auch die das *computational environments*, welches zur Analyse genutzt wird.

Warum Reproducible Research?

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

Karl Popper

Warum Reproducible Research?

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

Karl Popper

- In 2012, a study by Begley and Ellis was published in Nature that reviewed a decade of research. That study found that 47 out of 53 medical research papers focused on cancer research were irreproducible.

Warum Reproducible Research?

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

Karl Popper

- In 2012, a study by Begley and Ellis was published in Nature that reviewed a decade of research. That study found that 47 out of 53 medical research papers focused on cancer research were irreproducible.

- Es gibt eine wirkliche Krise der wissenschaftlichen Methode.

Warum Reproducible Research?

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

Karl Popper

- In 2012, a study by Begley and Ellis was published in Nature that reviewed a decade of research. That study found that 47 out of 53 medical research papers focused on cancer research were irreproducible.

- Es gibt eine wirkliche Krise der wissenschaftlichen Methode.

- In der Pharmakologie ist die Wiederholbarkeit der Schlüssel zur Zulassung neuer Medikamente.

Zwei unterschiedliche Wiederholbarkeiten

- Replicability und Reproducibility

Zwei unterschiedliche Wiederholbarkeiten

- Replicability und Reproducibility
- Replicability (Replizierbarkeit): Die Wiederholung eines Experiements führt zu denselben Ergebnissen

Zwei unterschiedliche Wiederholbarkeiten

- Replicability und Reproducibility

- Replicability (Replizierbarkeit): Die Wiederholung eines Experiements führt zu denselben Ergebnissen
- Reproducibility (Nachvollziehbarkeit): Gegeben die Daten und die Beschreibung der Analyse kann eine unbeteiligte Partei die Ergebnisse nachvollziehen.

Zwei unterschiedliche Wiederholbarkeiten

- Replicability und Reproducibility

- Replicability (Replizierbarkeit): Die Wiederholung eines Experiements führt zu denselben Ergebnissen
- Reproducibility (Nachvollziehbarkeit): Gegeben die Daten und die Beschreibung der Analyse kann eine unbeteiligte Partei die Ergebnisse nachvollziehen.

- Heute: Nachvollziehbarkeit der Analysen

Zwei unterschiedliche Wiederholbarkeiten

- Replicability und Reproducibility

- Replicability (Replizierbarkeit): Die Wiederholung eines Experiements führt zu denselben Ergebnissen
- Reproducibility (Nachvollziehbarkeit): Gegeben die Daten und die Beschreibung der Analyse kann eine unbeteiligte Partei die Ergebnisse nachvollziehen.

- Heute: Nachvollziehbarkeit der Analysen
- Auf deutsch: Wiederholbare Forschung, kurz WiFo

Zwei unterschiedliche Wiederholbarkeiten

- Replicability und Reproducibility

- Replicability (Replizierbarkeit): Die Wiederholung eines Experiements führt zu denselben Ergebnissen
- Reproducibility (Nachvollziehbarkeit): Gegeben die Daten und die Beschreibung der Analyse kann eine unbeteiligte Partei die Ergebnisse nachvollziehen.

- Heute: Nachvollziehbarkeit der Analysen
- Auf deutsch: Wiederholbare Forschung, kurz WiFo

- Durchaus bereits relevant: Biometrical Journal hat Editor für Wifo. Insbesondere wird die Wiederholbarkeit der Ergbnisse von Datenanalysen überprüft.

Was hindert bisher an RepRes?

- Zeitdruck

Was hindert bisher an RepRes?

- Zeitdruck
- Daten in Excel Sheets

Was hindert bisher an RepRes?

- Zeitdruck
- Daten in Excel Sheets
- Point and Klick Interfaces (SPSS)

Was hindert bisher an RepRes?

- Zeitdruck
- Daten in Excel Sheets
- Point and Klick Interfaces (SPSS)
- Proprietäre Datenformate (STATA, SAS)

Was hindert bisher an RepRes?

- Zeitdruck
- Daten in Excel Sheets
- Point and Klick Interfaces (SPSS)
- Proprietäre Datenformate (STATA, SAS)
- Proprietäre Dateiformate für den Report (.docx)

Was hindert bisher an RepRes?

- Zeitdruck
- Daten in Excel Sheets
- Point and Klick Interfaces (SPSS)
- Proprietäre Datenformate (STATA, SAS)
- Proprietäre Dateiformate für den Report (.docx)
- Fehlende Tools, um den Anreiz zu schaffen, aus diesen Fallen zu entkommen.

Wie kann man Wifo in den Arbeitsalltag von Wissenschaftlern bringen?

- Es muss einfach sein! So einfach, dass man nicht zurück will, zu den proprietären Formaten.

Wie kann man Wifo in den Arbeitsalltag von Wissenschaftlern bringen?

- Es muss einfach sein! So einfach, dass man nicht zurück will, zu den proprietären Formaten.
- Die Vorteile müssen offensichtlich sein.

Was ist bei der Auswahl der Werkzeuge zu beachten?

Anforderungen (und Konsequenzen)

- Jede am Prozess beteiligte Datei muss von Menschen *lesbar* sein. Beliebte lock-ins z.B. MS Office, SAP, STATA

Was ist bei der Auswahl der Werkzeuge zu beachten?

Anforderungen (und Konsequenzen)

- Jede am Prozess beteiligte Datei muss von Menschen *lesbar* sein. Beliebte lock-ins z.B. MS Office, SAP, STATA
- Damit wird jede Art von Vendor-Lock-in vermieden

Was ist bei der Auswahl der Werkzeuge zu beachten?

Anforderungen (und Konsequenzen)

- Jede am Prozess beteiligte Datei muss von Menschen *lesbar* sein. Beliebte lock-ins z.B. MS Office, SAP, STATA
- Damit wird jede Art von Vendor-Lock-in vermieden
- Daten z.B. als .csv oder Ähnliches

Was ist bei der Auswahl der Werkzeuge zu beachten?

Anforderungen (und Konsequenzen)

- Jede am Prozess beteiligte Datei muss von Menschen *lesbar* sein. Beliebte lock-ins z.B. MS Office, SAP, STATA
- Damit wird jede Art von Vendor-Lock-in vermieden
- Daten z.B. als .csv oder Ähnliches
- Datenbanken: SQL Query Texte

Was ist bei der Auswahl der Werkzeuge zu beachten?

Anforderungen (und Konsequenzen)

- Jede am Prozess beteiligte Datei muss von Menschen *lesbar* sein. Beliebte lock-ins z.B. MS Office, SAP, STATA
- Damit wird jede Art von Vendor-Lock-in vermieden
- Daten z.B. als .csv oder Ähnliches
- Datenbanken: SQL Query Texte
- Programmcode und Analyseerbegnisse als Textfiles. Grafiken als Programm zu ihrer Erzeugung.

Was ist bei der Auswahl der Werkzeuge zu beachten?

Anforderungen (und Konsequenzen)

- Jede am Prozess beteiligte Datei muss von Menschen *lesbar* sein. Beliebte lock-ins z.B. MS Office, SAP, STATA
- Damit wird jede Art von Vendor-Lock-in vermieden
- Daten z.B. als .csv oder Ähnliches
- Datenbanken: SQL Query Texte
- Programmcode und Analyseerbegnisse als Textfiles. Grafiken als Programm zu ihrer Erzeugung.

- Gandrud (2014): Die verschiedenen Dateien einer Analyse (Daten, Bilder, Tabellen, Code, Prosa) müssen *explizit* miteinander verbunden werden. Wenn sich z.B. an den Daten sich etwas ändert, sollen sich Tabellen und Graphen automatisch mitändern. Keine implizite Abhängigkeit, sonder explizite!

Konsequenzen!

- Damit fallen schon die üblichen Office-Produkte als Arbeitswerkzeuge aus. Ein Excel Worksheet ist nicht transparent. Natürlich muss in beide Richtungen mit Standardprogrammen kommuniziert werden können!

Konsequenzen!

- Damit fallen schon die üblichen Office-Produkte als Arbeitswerkzeuge aus. Ein Excel Worksheet ist nicht transparent. Natürlich muss in beide Richtungen mit Standardprogrammen kommuniziert werden können!

- Wenn z.B. Reviewer beurteilen sollen, ob ein Artikel angenommen wird, dann muss er in der Lagen sein, die Rechnungen zu reproduzieren. Was ist mit Software,die mit hohen Kosten verbunden ist (Matlab, SAS, Mathematica)?

Konsequenzen!

- Damit fallen schon die üblichen Office-Produkte als Arbeitswerkzeuge aus. Ein Excel Worksheet ist nicht transparent. Natürlich muss in beide Richtungen mit Standardprogrammen kommuniziert werden können!

- Wenn z.B. Reviewer beurteilen sollen, ob ein Artikel angenommen wird, dann muss er in der Lagen sein, die Rechnungen zu reproduzieren. Was ist mit Software,die mit hohen Kosten verbunden ist (Matlab, SAS, Mathematica)?

- Was ist mit alten Versionen von Software? Z.B. STATA hat mal das Datenformat inkompatibel geändert.

Konsequenzen!

- Damit fallen schon die üblichen Office-Produkte als Arbeitswerkzeuge aus. Ein Excel Worksheet ist nicht transparent. Natürlich muss in beide Richtungen mit Standardprogrammen kommuniziert werden können!

- Wenn z.B. Reviewer beurteilen sollen, ob ein Artikel angenommen wird, dann muss er in der Lagen sein, die Rechnungen zu reproduzieren. Was ist mit Software,die mit hohen Kosten verbunden ist (Matlab, SAS, Mathematica)?

- Was ist mit alten Versionen von Software? Z.B. STATA hat mal das Datenformat inkompatibel geändert.
- Eigentlich landet man zwangsläufig bei OpenSource oder mindestens bei “free as free beer” Software, wenn

Konsequenzen!

- Damit fallen schon die üblichen Office-Produkte als Arbeitswerkzeuge aus. Ein Excel Worksheet ist nicht transparent. Natürlich muss in beide Richtungen mit Standardprogrammen kommuniziert werden können!

- Wenn z.B. Reviewer beurteilen sollen, ob ein Artikel angenommen wird, dann muss er in der Lagen sein, die Rechnungen zu reproduzieren. Was ist mit Software,die mit hohen Kosten verbunden ist (Matlab, SAS, Mathematica)?

- Was ist mit alten Versionen von Software? Z.B. STATA hat mal das Datenformat inkompatibel geändert.
- Eigentlich landet man zwangsläufig bei OpenSource oder mindestens bei “free as free beer” Software, wenn

- RStudio ist ein perfektes Beispiel für die Power von Open Source und entsprechender Lizenzierung! Im Wesentlichen ein hervorragendes Interface zu einem ganzen Haufen FOSS!

Worum kümmert sich der Kurs nicht?

- Wie konserviert man Rechenumgebungen? Vmware? Docker? Hardware einlagern?

Worum kümmert sich der Kurs nicht?

- Wie konserviert man Rechenumgebungen? Vmware? Docker? Hardware einlagern?
- Wie konserviert man Daten? Wie kann die Integrität eines extrahierten Datensatzes gesichert werden?

Worum kümmert sich der Kurs nicht?

- Wie konserviert man Rechenumgebungen? Vmware? Docker? Hardware einlagern?
- Wie konserviert man Daten? Wie kann die Integrität eines extrahierten Datensatzes gesichert werden?
- Revision Control Systems: svn, git etc. Extrem nützlich, sprengt aber den Rahmen eines solch kurzen Kurses

Die Werkzeuge (standing on the shoulders of giants)

Was haben wir zur Verfügung?

- TeX bzw. LaTeX (Donald Knuth)

Die Werkzeuge (standing on the shoulders of giants)

Was haben wir zur Verfügung?

- TeX bzw. LaTeX (Donald Knuth)
- Markdown bzw. RMarkdown
  (John Gruber und 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

Die Werkzeuge (standing on the shoulders of giants)

Was haben wir zur Verfügung?

- TeX bzw. LaTeX (Donald Knuth)
- Markdown bzw. RMarkdown
  (John Gruber und 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)

Die Werkzeuge (standing on the shoulders of giants)

Was haben wir zur Verfügung?

- TeX bzw. LaTeX (Donald Knuth)
- Markdown bzw. RMarkdown
  (John Gruber und 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)

Die Werkzeuge (standing on the shoulders of giants)

Was haben wir zur Verfügung?

- TeX bzw. LaTeX (Donald Knuth)
- Markdown bzw. RMarkdown
  (John Gruber und 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.)

Die Werkzeuge (standing on the shoulders of giants)

Was haben wir zur Verfügung?

- TeX bzw. LaTeX (Donald Knuth)
- Markdown bzw. RMarkdown
  (John Gruber und 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)

Die Werkzeuge (standing on the shoulders of giants)

Was haben wir zur Verfügung?

- TeX bzw. LaTeX (Donald Knuth)
- Markdown bzw. RMarkdown
  (John Gruber und 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)
- SWeave (Fritz Leisch)

Die Werkzeuge (standing on the shoulders of giants)

Was haben wir zur Verfügung?

- TeX bzw. LaTeX (Donald Knuth)
- Markdown bzw. RMarkdown
  (John Gruber und 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)
- SWeave (Fritz Leisch)
- git (Linus Torvalds)

Das Ziel

- Ein Workflow, der das wissenschaftliche Arbeiten nicht behindert.

Das Ziel

- Ein Workflow, der das wissenschaftliche Arbeiten nicht behindert.
- Es soll eine Integrierte Umgebung geschaffen werden, die den Anforderungen von WiFo genügt und als Ausgabe z.B. HTML und PDF (und docx) Versionen eines Reports erzeugt.

Das Ziel

- Ein Workflow, der das wissenschaftliche Arbeiten nicht behindert.
- Es soll eine Integrierte Umgebung geschaffen werden, die den Anforderungen von WiFo genügt und als Ausgabe z.B. HTML und PDF (und docx) Versionen eines Reports erzeugt.
- Die Werkzeuge dürfen der inhaltlichen Arbeit nicht im Weg stehen!

Arbeitsschritte eine quantitativen Arbeit

- Daten sammeln und aufbereiten Es gehört schon hier dazu, einen Plan zu haben, wo und wie man die Daten Forschungspartnern, insbesondere zukünftigen(!), zugänglich machen möchte. (Homepage, Owncloud, Dropbox..)

Erinnerung: Alle drei Schritte müssen reproduzierbar sein! Hoffnungsvollerweise fallen die letzten beiden Schritte zusammen, wenn man es richtig macht.

Arbeitsschritte eine quantitativen Arbeit

- Daten sammeln und aufbereiten Es gehört schon hier dazu, einen Plan zu haben, wo und wie man die Daten Forschungspartnern, insbesondere zukünftigen(!), zugänglich machen möchte. (Homepage, Owncloud, Dropbox..)
- Angst vor der Transparenz ist schlechte wissenschftliche Praxis!

Erinnerung: Alle drei Schritte müssen reproduzierbar sein! Hoffnungsvollerweise fallen die letzten beiden Schritte zusammen, wenn man es richtig macht.

Arbeitsschritte eine quantitativen Arbeit

- Daten sammeln und aufbereiten Es gehört schon hier dazu, einen Plan zu haben, wo und wie man die Daten Forschungspartnern, insbesondere zukünftigen(!), zugänglich machen möchte. (Homepage, Owncloud, Dropbox..)
- Angst vor der Transparenz ist schlechte wissenschftliche Praxis!

- Analyse

Erinnerung: Alle drei Schritte müssen reproduzierbar sein! Hoffnungsvollerweise fallen die letzten beiden Schritte zusammen, wenn man es richtig macht.

Arbeitsschritte eine quantitativen Arbeit

- Daten sammeln und aufbereiten Es gehört schon hier dazu, einen Plan zu haben, wo und wie man die Daten Forschungspartnern, insbesondere zukünftigen(!), zugänglich machen möchte. (Homepage, Owncloud, Dropbox..)
- Angst vor der Transparenz ist schlechte wissenschftliche Praxis!

- Analyse
- Präsentation

Erinnerung: Alle drei Schritte müssen reproduzierbar sein! Hoffnungsvollerweise fallen die letzten beiden Schritte zusammen, wenn man es richtig macht.

Geschichte

- 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

Geschichte

- 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

- Hier Literate Data Analysis!

Geschichte

- 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

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

Geschichte

- 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

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

Geschichte

- 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

- Hier Literate Data Analysis!
- Sweave (2002, Fritz Leisch)
- knitr (2011, Yihui Xie)
- org-mode mit babel (2011, Eric Schulte)
  Add-on für emacs, welches über das org-babel Modul eine Interface zu vielen Programmiersprachen bietet (maxima, gnuplot, R, python, C . . .) Inklusive Exportfunktionen nach LaTeX, ODT, Markdown etc.

Die Idee der Literate Data Analysis

Man möchte gern den Progammtext und den Analysetext in einer Datei haben. Alles was man dazu braucht, ist eine Syntax (und die Tools), um aus diesem *gemischten* Text den Programmcode bei Bedarf zu extrahieren und auszuführen.

Bei Knuth war die Ausführung des Programms in situ noch nicht vorgesehen.

Heute hat man den Ausgangstext, in dem Freitext und Programmtext wechselweise auftreten, und durch "weaven" wird ein Enddokument erzeugt, das Freitext und Ausgabe des Programmtexts, wahlweise auch den Quelltext, enthält.

Praktische Probleme machen Bilder und Tabellen.

Begriffe

Die Codeabschnitte im Text heißen traditionell “chunks”. Durch spezielle Syntax wird dem Weavingtool begreiflich gemacht, dass ein Textabschnitt als Programm zu interpretieren ist.

Bei Knuth sah das wie folgt aus:

```
Umgebender Text
<<opt. chunkname >>=
Programmcode
@
Umgebender Text
```

Diese Syntax gibt es immer noch, es gibt aber mittlerweile Leserfreundlicheres.

RStudio - Betriebssystem für WiFo

Hier wird all das zusammen geführt!

- Oben links das Hauptdokument. Aus diesem wird der fertige Report.

RStudio - Betriebssystem für WiFo

Hier wird all das zusammen geführt!

- Oben links das Hauptdokument. Aus diesem wird der fertige Report.
- Unten links eine Art “Schmierzettel”, die R Console.

RStudio - Betriebssystem für WiFo

Hier wird all das zusammen geführt!

- Oben links das Hauptdokument. Aus diesem wird der fertige Report.
- Unten links eine Art “Schmierzettel”, die R Console.
- RStudio ist im Prinzip nicht nötig, da es lediglich das Setup aller Werkzeuge übernimmt. Alle Kommandos sind auch aus von der R Kommandozeile möglich. Allerdings ist z.B. das Setup von pandoc nicht trivial.

RStudio - Betriebssystem für WiFo

Hier wird all das zusammen geführt!

- Oben links das Hauptdokument. Aus diesem wird der fertige Report.
- Unten links eine Art "Schmierzettel", die R Console.
- RStudio ist im Prinzip nicht nötig, da es lediglich das Setup aller Werkzeuge übernimmt. Alle Kommandos sind auch aus von der R Kommandozeile möglich. Allerdings ist z.B. das Setup von pandoc nicht trivial.
- "Lediglich" ist aber die Lüge an dieser Aussage. RStudio löst ein großes Prolem!

Erste Variante für kurze Jobs (spinning)

- File -> New Notebook

Erste Variante für kurze Jobs (spinning)

- File -> New Notebook
- File -> Compile Notebook

Erste Variante für kurze Jobs (spinning)

- File -> New Notebook
- File -> Compile Notebook
- Perfektes Format für Übungsaufgaben (rpub.com).

Erste Variante für kurze Jobs (spinning)

- File -> New Notebook
- File -> Compile Notebook
- Perfektes Format für Übungsaufgaben (rpub.com).
- Dies ist ein Programm, das etwas Text enthält, kein Text mit etwas Programm.

Erste Variante für kurze Jobs (spinning)

- File -> New Notebook
- File -> Compile Notebook
- Perfektes Format für Übungsaufgaben (rpub.com).
- Dies ist ein Programm, das etwas Text enthält, kein Text mit etwas Programm.
- Kommentare können als R-Kommentare eingefügt werden

Erste Variante für kurze Jobs (spinning)

- File -> New Notebook
- File -> Compile Notebook
- Perfektes Format für Übungsaufgaben (rpub.com).
- Dies ist ein Programm, das etwas Text enthält, kein Text mit etwas Programm.
- Kommentare können als R-Kommentare eingefügt werden
- oder Roxygen-Kommentare mit Zeilenanfang #', die dann bereits interpretiert werden.

Erste Variante für kurze Jobs (spinning)

- File -> New Notebook
- File -> Compile Notebook
- Perfektes Format für Übungsaufgaben (rpub.com).
- Dies ist ein Programm, das etwas Text enthält, kein Text mit etwas Programm.
- Kommentare können als R-Kommentare eingefügt werden
- oder Roxygen-Kommentare mit Zeilenanfang #', die dann bereits interpretiert werden.
- Es können auch Headerinformation in einem speziellen Format hinzugefügt werden (YAML).

Die Verschiedenen Eingabeformate für knitr

- *.R: R Skript Format für Spinning
- *.Rnw (R noweb): Ausgabeformat ist LaTeX. Syntax für Codeblocks

  ```
  << >>= \n
  Hierher der R-Code
  @
  ```

- *.Rtex (R TeX): Ausgabeformat LaTeX, aber Syntax
  %%begin.rcode
  Hierher der R-Code
  %%

Die Verschiedenen Eingabeformate für knitr

- *Rhtml

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

- *Rmd (Rmarkdown): Ausgabeformat Markdown. Dann Weiterbehandlung mit pandoc. Ausgabeformat: LaTeX, PDF, Word, etc.

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

  Jeweils drei backticks!

Heute: Rmarkdown!

- Rmarkdown ist eine Variante von Markdown. Entwicklungsziel war eine Auszeichnungssprache, die man auch im Quelltext gut lesen kann. Gleichzeitig sollten eine einige Strukturelemente verfügbar sein: Überschriften, **bold**, *italic*, etc.

Heute: Rmarkdown!

- Rmarkdown ist eine Variante von Markdown. Entwicklungsziel war eine Auszeichnungssprache, die man auch im Quelltext gut lesen kann. Gleichzeitig sollten eine einige Strukturelemente verfügbar sein: Überschriften, **bold**, *italic*, etc.
- Es wird also ein Text in Markdown geschrieben, der Chunks in R enthält.

Heute: Rmarkdown!

- Rmarkdown ist eine Variante von Markdown. Entwicklungsziel war eine Auszeichnungssprache, die man auch im Quelltext gut lesen kann. Gleichzeitig sollten eine einige Strukturelemente verfügbar sein: Überschriften, **bold**, *italic*, etc.
- Es wird also ein Text in Markdown geschrieben, der Chunks in R enthält.
- Die Verarbeitungsreihenfolge ist dabei
  text.Rmd -> knitr -> text.md -> pandoc -> html, tex, doc (-> pdflatex -> pdf)

Heute: Rmarkdown!

- Rmarkdown ist eine Variante von Markdown. Entwicklungsziel war eine Auszeichnungssprache, die man auch im Quelltext gut lesen kann. Gleichzeitig sollten eine einige Strukturelemente verfügbar sein: Überschriften, **bold**, *italic*, etc.
- Es wird also ein Text in Markdown geschrieben, der Chunks in R enthält.
- Die Verarbeitungsreihenfolge ist dabei
  text.Rmd -> knitr -> text.md -> pandoc -> html, tex, doc (-> pdflatex -> pdf)
- pandoc ermöglicht auch die Einbindung von Zitationen, was im puren Markdown nicht möglich ist.

Struktur eines Markdown Dokuments

Header (optional)

```
---
title: Reproducible Research
subtitle: Ein Workflow mit R / knitr / RStudio
author: Detlef Steuer
date: Hamburg, 18. September 2015
output:
  beamer_presentation:
    toc: true
---
```

Struktur eines Markdown Dokuments

Fließtext in markdown

````
Eine Überschrift
================
Etwas Text in **fett** oder *italic*
Eine Unterüberschrift
---------------------
Eine Formel im Text $e^{i\pi} = -1$.
Ein Code-Chunk:
```{ r chunkname, eval=FALSE}
hist(rnorm(100))
```
Oder auch rechnen im Text:
Der Iris Datensatz enthält ` r nrow(iris)` Beobachtungen
````

HTML Code wird unverändert durchgereicht, genau so LaTeX.

Struktur eines Markdown Dokuments

Mehr und vollständig

- Rmarkdown Cheat-Sheet
- Rmarkdown Reference Guide

Chunks im Detail

- Normalerweise laufen die codeblocks in einem Rmd Dokument konsekutiv ab, chunk für chunk.

Chunks im Detail

- Normalerweise laufen die codeblocks in einem Rmd Dokument konsekutiv ab, chunk für chunk.
- Neben dem Namen gibt es eine ganze Reihe von Optionen, die die Abarbeitung der chunks detailliert kontrollieren.

Chunks im Detail

- Normalerweise laufen die codeblocks in einem Rmd Dokument konsekutiv ab, chunk für chunk.
- Neben dem Namen gibt es eine ganze Reihe von Optionen, die die Abarbeitung der chunks detailliert kontrollieren.
- Nicht vollständig! Unbedingt in der knitr Dokumentation nachlesen!

Nützliche Chunk Optionen

Die Optionen werden in der Regel innerhalb der geschweiften Klammern angegeben.

Computation Control

- eval = TRUE | FALSE ; Der Codeblock wird ausgeführt (oder nicht)
- echo = TRUE | FALSE ; Der Quelltext wird in das Ergebnisdokument eingefügt (oder nicht)
- results = markup | asis | . . . ; Die Art der Übernahmen der Ausgaben in das Ergebnisdokument
- error: (TRUE; logical) ; Stoppt nicht bei Fehlern!

Caching

Spezielle Option für aufwendige Rechnungen. knitr kann verfolgen, ob sich Codeblock ändern oder nicht. Wenn sich nichts geändert hat, kann über den cache Parameter festgelegt werden, dass die Werte der Variablen ohne Neuberechnung übernommen werden können.

- cache = TRUE | FALSE oder feiner abgestuft numerisch; Die Ergebnisse des Codeblocks werden vor unnützer Neuberechnung geschützt.
- dependson: Chunkname (NULL; character or numeric) ; Im Falle von cache = TRUE kann über dependson eine explizite Abhängigkeit von einem Block definiert werden.
- cache.vars: (NULL); erlaubt eine Eingrenzung des Cachemechanismus auf bestimmte Variablen.

Abbildungen

- dev: (‘pdf’ for LaTeX output and ‘png’ for HTML/markdown; character) ; Auswahl des Grafikformats. Alles Devices, die R bietet sind möglich.
  dev=c(‘pdf’, ‘png’) möglich!
- fig.width, fig.height, out.width, out.height: Skalierungen für Abbildungen. Relatives Skalieren ist möglich, z.B. fig.width=.8\linewidth in LaTeX

Teildokumente

- child: (NULL; character) Vektor von Dateinamen; Ähnlich Includes

Sonstiges

- Alterativ innerhalb des Codeblocks z.B.
  opts_chunk$set(comment=NA, fig.width=6, fig.height=6))
  opts_chunk$set(dev = c("pdf", "jpg"))
- Werte für die Optionen müssen in einer Zeile eingegeben werden. Die Ausdrücke müssen stets gültige R Ausdrücke sein.
- Punkte und Leerzeichen in Namen und Verzeichnisnamen vermeiden!

Tabellen

- Tabellen kopieren und formatieren gehört zu den langweiligsten und deshalb fehleranfälligsten Tätigkeiten überhaupt.

Eine automatische Erzeugung ist deshalb in jedem Falle vorzuziehen.

- R-Pakete für schöne Tabellen: xtable, stargazer, apsrtable, knitr::kable
- Wichtige Chunkoptionen: “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="Von nichts kommt nichts")
print.xtable(nice.table,type="latex")
```

% latex table generated in R 3.2.2 by xtable 1.7-4 package % Fri Sep 18 08:25:08 2015

	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:Von nichts kommt nichts

Tabellen (Fort.)

Verschiedene Optionen können global gesetzt werden, z.B. options("xtable.type" = "html")

Eigentlich, wenn mit knitr, dann mit knitr::kable, da dieses Kommando automatisch das richtige Ausgabeformat erzeugt!

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

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

Abbildungen

- Extern erzeugte Bilder können mit einfacher Syntax eingebunden werden: ![Text][Pfad/zum/Bild.png]
- Nützliche Chunkoptionen sind z.B: fig.align=‘center’, out.width out.height
- Unabhängig vom Grafiksystem werden die üblichen R Kommandos in den chunk geschrieben, um die Abbildung zu konstruieren. Der Rest wird im Hintergrund von knitr erledigt! (Temporäre Dateien anlegen, device auswählen, etc.)

Reproduzierbare Simulationen

Um Wifo auch im Bereich der Monte-Carlo-Methoden richtig anzuwenden ist set.seed() das unbedingt nötige Kommando. Es erzwingt einen Startwert für die Erzeugung der Pseudozufallszahlen und somit die Wiederholung des kompletten Streams

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

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

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

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

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

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

Literatur

Mittels pandoc ist sogar die Nutzung von Literaturdatenbanken diversen Formaten möglich.

Der YAML-Header wird um eine Zeile mit dem Namen der Literaturdatenbank ergänzt, z.B.:

```
---
title: "Reproducible Research"
author: "Detlef Steuer"
date: "Hamburg, 18. September 2015"
output:
  beamer_presentation:
    toc: yes
subtitle: Ein Workflow mit R / knitr / RStudio
bibliography: Kurs.bib
---
```

Zitiert wird nun mit der Syntax

```
‘‘‘
Programmiersprache der Wahl: R [@R2014]
Ein schönes Buch ist z.B. [@Ligges2005]
Für WiFo z.B. [@Gandrud2014]
‘‘‘
```

Im Endtext sieht das dann wie folgt aus:

Programmiersprache der Wahl: R (R Core Team 2014)
Ein schönes Buch dazu ist z.B. (Ligges 2005)
Für WiFo z.B. (Gandrud 2014)

Das Literaturverzeichnis erscheint dann am Ende des Dokuments.

Fazit

- RMarkdown ist eine einfach gehaltene Auszeichnugssprache, die es im Zusammenspiel mit RStudio ermöglicht, WiFo schnell umzusetzen und den Arbeitsablauf einzuüben.
- Nach der Pause wird ein einfaches RMarkdown Dokument erstellt, das sich auch finanziell sofort auszahlt . . .
- Rtex ermöglicht speziellere Formatierungen, ist jedoch nicht so versatil und nicht so schön im Quelltext zu lesen.

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/`.