

L^AT_EX-Tipps & knitr

Propädeutikum im Sommersemester 2016

Paul Fink

Institut für Statistik, LMU

08.04.2016

- Direkte Verknüpfung von \LaTeX und R
- Übertragungsfehler vermeiden
- Funktionalität von R nutzen um \LaTeX -Code automatisch zu erzeugen
- Einbindung von Grafiken
- Auch bei größeren Dokumenten geeignet dank Cache

Dokument besteht aus \LaTeX und R Blöcken (*Chunks*)

⇒ Datei-Endung auf .Rnw statt .tex

.....

Hier steht Text in \LaTeX

<<chunk-name, optionen>>=*%Beginn R-Block, mit Name und Optionen*

1+1

a <- 1:4

@% Hiermit wird der R-Block wieder beendet

Hier geht es nun weiter im $\text{\texttt{\textbackslash textbf{\textbackslash LaTeX}}}$ -Modus

Mit $\text{\texttt{\textbackslash Sexpr\{...\}}}$ wird R-Code auch inline ausgewertet, zum Beispiel

$\text{\texttt{\textbackslash Sexpr\{1+1\}}}$

.....

- RStudio hat Unterstützung für knitr bereits eingebaut
Achtung: Package `knitr` muss manuell installiert werden!
- RStudio bietet nur marginale \LaTeX -Unterstützung
 \implies Gerade bei größeren Projekten \LaTeX -Editor verwenden
- \LaTeX -Editoren können knitr-Dateien kompilieren, bieten aber keine Unterstützung in R
- Modularisierung sehr hilfreich

Kompilierung

- ① Auswertung der R-Blöcke (Funktion `knit` aus `knitr`)
- ② Kompilierung mit (pdf) \LaTeX

Beides zusammen auf Knopfdruck in RStudio oder dem \LaTeX -Editor

Wichtige lokale Optionen

Alle Optionen müssen gültige R-Ausdrücke sein; Übergabe wie Argumente!

Option	Erklärung
<code>child</code> (char)	Entspricht <code>\input{Dateipfad}</code> in \LaTeX für Rnw-Dateien
<code>eval</code> (logical, <i>TRUE</i>)	Soll Block ausgewertet werden?; spezielle Bedeutung bei Angabe als numeric
<code>echo</code> (logical, <i>TRUE</i>)	Soll R-Code ausgegeben werden?; spezielle Bedeutung bei Angabe als numeric
<code>results</code> (char, <i>"markup"</i>)	Formatierung R-Ausgabe; weitere Möglichkeiten sind <i>"asis"</i> , <i>"hide"</i>
<code>include</code> (logical, <i>TRUE</i>)	Soll R-Ausgabe ins Dokument; Code wird immer ausgeführt, im Hintergrund bei <i>FALSE</i>
<code>cache</code> (logical, <i>FALSE</i>)	Soll ein Cache für Ergebnis und Code verwendet werden? Empfehlenswert bei lange dauernden Blöcken.
Viele weitere	...

- Optionen können pro Block einzeln gesetzt werden
- Globales Setzen von Optionen über die Funktion `opts_chunk$set()`, lokal gesetzte Optionen können dann die globalen überschreiben
- Setzen von Theme zum Hervorheben des Codes über `knit_theme$set(knit_theme$get("themename"))`
- Links:
 - [Alle \(dokumentierten\) Optionen](#)
 - [Breite Liste mit Beispielen](#)
 - [Alle aktuell verfügbaren Themes](#)

- Durch R erzeugte Grafiken werden automatisch durch knitr eingebunden
- Grafikausgabe wird gesteuert wie über Kommandozeile
- Verhalten der Grafik steuerbar über Optionen des Blocks u.A.
 - Größe der Grafik bei Erstellung und Einbindung
 - Verwendetes Device (auch TikZ-Device möglich)
 - Einbinden direkt als Float-Objekt
 - Zusammenfassung mehrerer Grafiken

Beispiel – Setup

```
\documentclass[a4paper,10pt]{article}%Anfang wie in normalem LaTeX-Dokument

\usepackage{tikz}% Für TikZ-Device
\usepackage{...}% Alles weitere Laden/Definieren wie in jedem LaTeX-Dokument
% Achtung: knitr lädt automatisch color und graphicx!

% Nun folgen die globalen knitr Einstellungen
% Durch include=FALSE wird nur der R-Code innen ausgeführt
% Durch cache=FALSE wird der R-Code nicht gecached
<<setup, include=FALSE, cache=FALSE>>=
opts_chunk$set(prompt = FALSE, comment = NA, size="normalsize")
# Keine Anzeige der Prompt ("> ")
# R-Ausgaben landen nicht mit vorangestelltem 'comment' im Code
# Textgröße auf normal

# Folgende Funktion ersetzt das Laden des color-Package mit xcolor
knit.hooks$set(document = function(x) {
  sub("\\usepackage[]{color}", "\\usepackage[]{xcolor}", x, fixed = TRUE)
})
@
.....
```


Bespiel – Erste Schritte

.....

```
\begin{document}
```

Dies ist mein Erstes gemischtes R-\LaTeX-Dokument.

Als Erstes schreibe ich viel Unsinn, auf den ich nachher zur{"u}ckkomme!

```
<<meinblock1>>=
```

```
a <- 1:2
```

```
b <- matrix(1:4, ncol = 2)
```

```
b %*% a #Boah es geht!
```

```
@
```

Nun kommt wieder mehr textueller Unsinn, unterbrochen durch eine Welt-Formel, die ich nun teste: $1 + 1 = \text{\Sexpr{1+1}}$. Und es stimmt! So clever!

Jetzt mal nur die Ergebnisausgabe:

```
<<meinblock2, echo=FALSE>>=
```

```
rnorm(10)
```

```
@
```

Weiterer Unsinn

.....

Beispiel – Grafiken

.....

Jetzt mal ein `\textbf{P}`aar Grafiken

```
<<grafik1, fig.out="0.5\textwidth">>=%pdf-Device ist Default
plot(2:10, (2:10)^4, main="R Grafik in LaTeX", ylab=expression(R[i]) )
@
```

Man kann mit TikZ-Device auch direkt `\LaTeX`-Code in der Grafik verwenden.
Diesmal wird es ohne Code angezeigt (`\texttt{echo=FALSE}`)

```
<<grafik2, echo=FALSE, dev="tikz", fig.out="0.5\textwidth">>=
plot(2:10, (2:10)^4, main="R Grafik in \\LaTeX", ylab="$R_{i}$" )
@
```

```
\end{document}
```

Objekte aus R direkt in \LaTeX einbinden (Matrix als Tabelle)

Verwendung von knitr-Output direkt als \LaTeX -Input

⇒ cat und knitr-Optionen results="asis" und echo=TRUE

```
<<matrix, results="hide">>=% Matrix Erstellung simuliert ein Ergebnis
m <- matrix(rnorm(18), ncol = 3)
@

\begin{tabular}{lrrr}\toprule
Gruppe &  $\beta_1$  &  $\beta_2$  &  $\beta_3$ \\midrule
<<tabelle, results="asis", echo=FALSE>>=
for(i in seq_len(NROW(m))) {
  cat("Gruppe", i, "&",
      paste0("$", format(m[i, ], digits = 2), "$", collapse = " & "),
      "\\\n")
}
@
\bottomrule
\end{tabular}
```