

# “Reproducible Research” mit Emacs’ Org mode und R

Bernd Weiß

<http://berndweiss.net>  
[bernd.weiss@uni-koeln.de](mailto:bernd.weiss@uni-koeln.de)

2012-07-06

# Outline

Vorbemerkungen

Einleitung

- Problem

- Emacs

- Org mode

- Babel: Active code in Org mode

Babel und R

- Einführung

- Tangling und weaving

- Ein kurzes vollständiges Org-Beispiel

R und andere Sprachen (Python)

References

# Topic

## Vorbemerkungen

### Einleitung

Problem

Emacs

Org mode

Babel: Active code in Org mode

### Babel und R

Einführung

Tangling und weaving

Ein kurzes vollständiges Org-Beispiel

### R und andere Sprachen (Python)

### References

# Vorbemerkungen

- ▶ Bin fortgeschrittenen Anfänger was Org angeht
- ▶ Vielen Dank an Carsten Dominik (von 2003 bis 2011 Hauptentwickler), Eric Schulte und Dan Davison (haben die “Babel-Funktionen” in Org Mode entwickelt)
- ▶ Alle Materialien (inkl. engl. Fassung der Folien) dieser Präsentation finden sich auf github:  
[https://github.com/berndweiss/ps2012-07-KRUG\\_org\\_r](https://github.com/berndweiss/ps2012-07-KRUG_org_r).

# Topic

Vorbemerkungen

## Einleitung

Problem

Emacs

Org mode

Babel: Active code in Org mode

Babel und R

Einführung

Tangling und weaving

Ein kurzes vollständiges Org-Beispiel

R und andere Sprachen (Python)

References

# Topic

## Vorbemerkungen

### Einleitung

- Problem

- Emacs

- Org mode

- Babel: Active code in Org mode

### Babel und R

- Einführung

- Tangling und weaving

- Ein kurzes vollständiges Org-Beispiel

### R und andere Sprachen (Python)

### References

# Problem

- ▶ Das Verfassen eines empirischen (Forschungs-)Artikels
  - ▶ ist selten ein linearer Prozess.
  - ▶ ist außerdem mit dem Problem verbunden, dass die wissenschaftliche Gemeinschaft meinen Ausführungen vertrauen können muss. Doch wie schaffe ich dieses Vertrauen?
- ▶ Wir benötigen zum Verfassen eines Artikels also ein Tool, das
  - ▶ “dynamische Elemente” enthält. Wenn sich die Datengrundlage ändert, sorgen diese “dynamischen Elemente” dafür, dass etwa alle Tabellen oder Abbildungen erneuert werden.
  - ▶ vollständigen Zugang zum Analysecode ermöglicht und damit auch Replikationen (i.e. “reproducible research”).

# Literate programming und reproducible research

**Literate programming (LP)** Enhances traditional software development by embedding code in explanatory essays and encourages treating the act of development as one of communication with future maintainers.

**Reproducible research (RR)** Embeds executable code in research reports and publications, with the aim of allowing readers to re-run the analyses described.

(Source: Schulte et al. 2012: 3)



# Existing tools

Tool	LP	RR	$\text{\LaTeX}$ Export	HTML Export	Language
Javadoc	partial	no	no	yes	Java
POD	partial	no	no	yes	Perl
Haskell	partial	no	yes	yes	Haskell
noweb	yes	no	yes	yes	any
cweb	yes	no	yes	yes	C/C++
Sweave	partial	yes	yes	yes	R
SASweave	partial	yes	yes	yes	R/SAS
Statweave	partial	yes	yes	yes	any
Scribble	yes	yes	yes	yes	scheme
Knitr	partial	yes	yes	yes	R
dexy	yes	yes	yes	yes	any
Org-mode	yes	yes	yes	yes	any

(Source: Schulte et al. 2012: 6, modified and updated)

# Topic

Vorbemerkungen

**Einleitung**

Problem

**Emacs**

Org mode

Babel: Active code in Org mode

Babel und R

Einführung

Tangling und weaving

Ein kurzes vollständiges Org-Beispiel

R und andere Sprachen (Python)

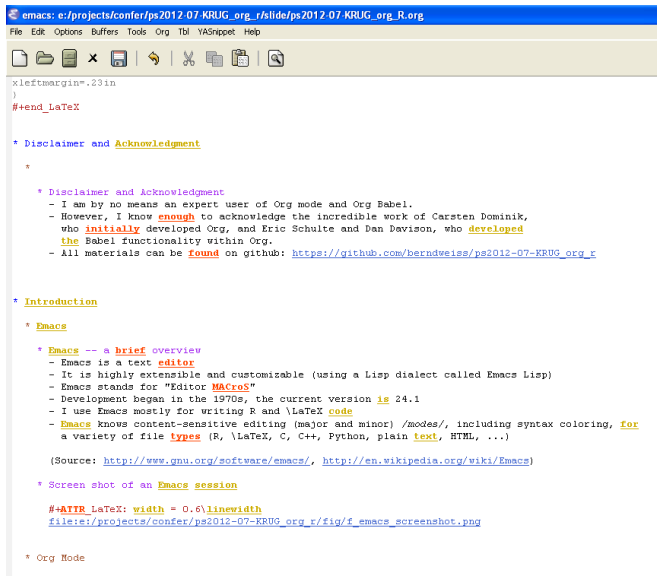
References

# Emacs – ein kurzer Überblick

- ▶ Emacs ist a Texteditor
- ▶ Erweiterbar und an eigene Bedürfnisse anpassbar durch Emacs Lisp
- ▶ Emacs steht für “Editor MACroS”
- ▶ Entwicklung begann bereits in den 1970er Jahren, die aktuelle Version ist 24.1
- ▶ Ich nutze Emacs vor allem für R und  $\text{\LaTeX}$
- ▶ Emacs kennt sg. “content-sensitive” editing (major and minor) *modes* (u.a. für R,  $\text{\LaTeX}$ , C, C++, Python, plain text, HTML, ...)

(Source: <http://www.gnu.org/software/emacs/>,  
<http://en.wikipedia.org/wiki/Emacs>)

# Screen shot einer Emacs session



```
emacs: e:/projects/confer/ps2012-07-KRUG_org_r/slide/ps2012-07-KRUG_org_R.org
File Edit Options Buffers Tools Org Tbl YASnippet Help

xleftmargin=.23in
)
#+end_LaTeX

* Disclaimer and Acknowledgment

*
* Disclaimer and Acknowledgment
- I am by no means an expert user of Org mode and Org Babel.
- However, I know enough to acknowledge the incredible work of Carsten Dominik,
  who initially developed Org, and Eric Schulte and Dan Davison, who developed
  the Babel functionality within Org.
- All materials can be found on github: https://github.com/berndweiss/ps2012-07-KRUG\_org\_r

* Introduction

* Emacs

* Emacs -- a brief overview
- Emacs is a text editor
- It is highly extensible and customizable (using a Lisp dialect called Emacs Lisp)
- Emacs stands for "Editor MACroS"
- Development began in the 1970s, the current version is 24.1
- I use Emacs mostly for writing R and \LaTeX code
- Emacs knows content-sensitive editing (major and minor) /modes/, including syntax coloring, for
  a variety of file types (R, \LaTeX, C, C++, Python, plain text, HTML, ...)

(Source: http://www.gnu.org/software/emacs/, http://en.wikipedia.org/wiki/Emacs)

* Screen shot of an Emacs session

#+ATTR LaTeX: width = 0.6\linewidth
file:e:/projects/confer/ps2012-07-KRUG_org_r/fig/f_emacs_screenshot.png

* Org Mode
```

# Topic

Vorbemerkungen

**Einleitung**

Problem

Emacs

**Org mode**

Babel: Active code in Org mode

Babel und R

Einführung

Tangling und weaving

Ein kurzes vollständiges Org-Beispiel

R und andere Sprachen (Python)

References

# Org mode

- ▶ Org ist ein “Emacs (major) mode for notes, planning, and authoring”
- ▶ Org files sind plain-text files
- ▶ Ich nutze Org für
  - ▶ technische Dokumentationen (Export nach PDF via pdfLaTeX or HTML) oder zur Erstellung von Folien
  - ▶ Organisation meines beruflichen und privaten Lebens (Verabredungen, Todo-Listen etc.)
  - ▶ Für meine persönliche Wissensdatenbank
  - ▶ ...
- ▶ Mehr unter: <http://www.orgmode.org> (tutorials, videos, ...)

# Ein Org-Beispiel (source file)

```
1  #+TITLE: Hello World!
2  #+AUTHOR: Martin Mosquito
3  #+DATE: <2012-07-06 Fr>
4
5  \thispagestyle{empty}
6
7  *  Headline
8
9  Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed
10 diam nonumy eirmod tempor invidunt ut labore et dolore magna
11 aliquyam erat, sed diam voluptua. At vero eos et accusam et
12 justo duo dolores et ea rebum. Stet clita kasd gubergren, no
13 sea takimata sanctus est Lorem ipsum dolor sit.
14
15 A list:
16 - Item 1
17 - Item 2
18
19 ** Subsection
20
21 Lorem ipsum dolor sit amet, consetetur sadipscing elitr,
22 sed diam nonumy eirmod tempor invidunt ut labore et dolore
23 magna aliquyam erat, sed diam
24
25 \begin{equation}
26 y = \alpha + \beta_1 x_1 + \epsilon
27 \end{equation}
```

# Ein Org-Beispiel (exported to PDF)

Hello World!

Martin Mosquito

2012-07-06

## Contents

<b>1</b>	<b>Headline</b>	<b>1</b>
1.1	Subsection . . . . .	1

## 1 Headline

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit.

A list:

- Item 1
- Item 2

### 1.1 Subsection

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam

$$y = \alpha + \beta_1 x_1 + \epsilon \tag{1}$$



# Topic

Vorbemerkungen

Einleitung

Problem

Emacs

Org mode

**Babel: Active code in Org mode**

Babel und R

Einführung

Tangling und weaving

Ein kurzes vollständiges Org-Beispiel

R und andere Sprachen (Python)

References

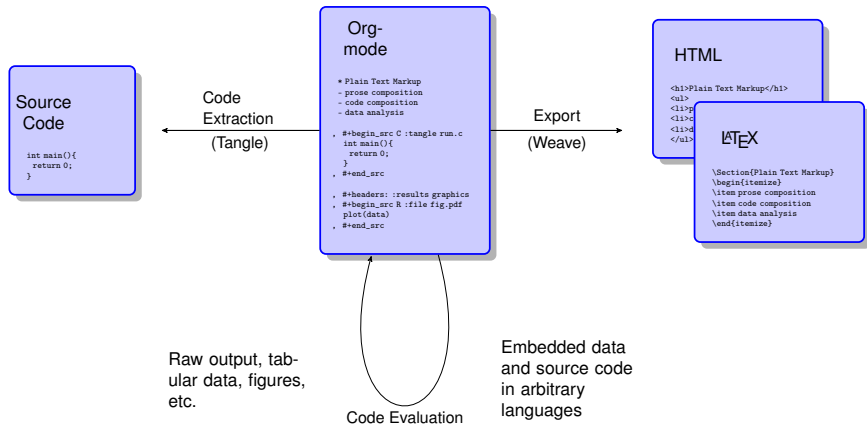
# Babel: Active code in Org mode

*Babel is about letting many different languages work together. Programming languages live in blocks inside natural language Org-mode documents. A piece of data may pass from a table to a Python code block, then maybe move on to an R code block, and finally end up embedded as a value in the middle of a paragraph or possibly pass through a gnuplot code block and end up as a plot embedded in the document.*

(Source:

<http://orgmode.org/worg/org-contrib/babel/intro.html>)

# Using code with(in) Org mode



Source: Schulte et al. (2012)

# Topic

Vorbemerkungen

Einleitung

Problem

Emacs

Org mode

Babel: Active code in Org mode

**Babel und R**

Einführung

Tangling und weaving

Ein kurzes vollständiges Org-Beispiel

R und andere Sprachen (Python)

References

# Topic

Vorbemerkungen

Einleitung

Problem

Emacs

Org mode

Babel: Active code in Org mode

**Babel und R**

Einführung

Tangling und weaving

Ein kurzes vollständiges Org-Beispiel

R und andere Sprachen (Python)

References

# Syntax eines Org-Code-Blocks

`#+NAME: <name>`

`#+BEGIN_SRC <language> <switches> <header arguments>`  
`<body>`

`#+END_SRC`

# R code blocks I (exports code and results)

## R code block

This is a short sentence. This is another sentence.  
And, finally, here comes some R code:

```
#+BEGIN_SRC R
set.seed(1)
x1 <- rnorm(100)
mean(x1)
x1.sd <- sd(x1)
#+END_SRC
```

```
#+RESULTS:
: [1] 0.1088874
```

## Evaluated and exported R code block

This is a short sentence. This is another sentence. And, finally, here comes some R code:

```
1 set.seed(1)
2 x1 <- rnorm(100)
3 mean(x1)
4 x1.sd <- sd(x1)
```

```
[1] 0.1088874
```

# R code blocks II (exports only results)

## R code block

This is a short sentence. Hey, there is another sentence. And, finally, here comes some R code:

```
#+BEGIN_SRC R :results output :exports results
set.seed(1)
x1 <- rnorm(100)
mean(x1)
x1.sd <- sd(x1)
#+END_SRC

#+RESULTS:
: [1] 0.1088874
```

## Evaluated and exported R code block

This is a short sentence. Hey, there is another sentence. And, finally, here comes some R code:

```
[1] 0.1088874
```



# R inline code blocks (source)

Der Mittelwert von x ist

```
src_R[:exports results :results raw]{round(mean(x1), 3)}
```

und der SD beträgt

```
src_R[:exports results :results raw]{round(x1.sd, 3)}.
```

Vektor =x1= and =x1.sd= wurden auf der vorherigen Folie in einem separaten Code-Block definiert.

# R inline code blocks (evaluated)

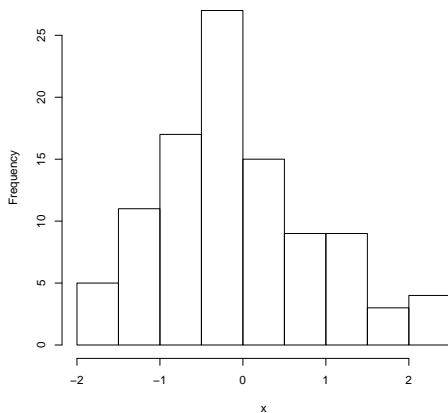
Der Mittelwert von  $x$  ist [1] 0.109 und der SD beträgt [1] 0.898. Vektor  $x1$  and  $x1.sd$  wurden auf der vorherigen Folie in einem separaten Code-Block definiert.

# Base graphics plot (source)

```
1:  #+headers: :exports both
2:  #+headers: :results graphics
3:  #+headers: :file img.pdf
4:  #+begin_src R
5:  x <- rnorm(100)
6:  hist(x)
7:  #+end_src
8:
9:  #+RESULTS:
10:  [[file:img.pdf]]
```

# Base graphics plot (evaluated)

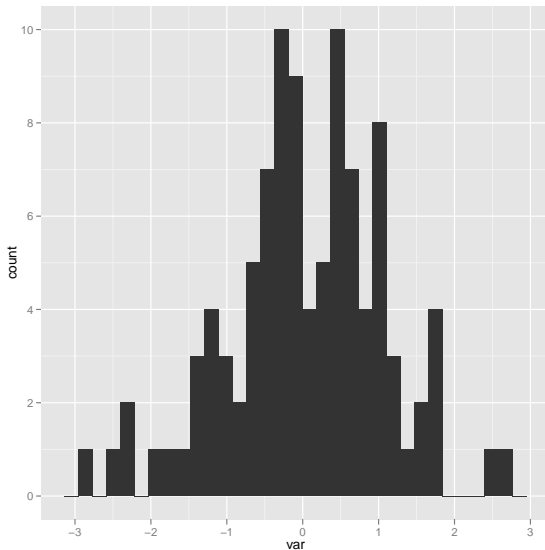
```
1 x <- rnorm(100)
2 hist(x, main = "")
```



# ggplot2 plot (Org source)

```
1:  #+headers: :exports results
2:  #+headers: :results graphics
3:  #+headers: :file f_ggplot2_ex.pdf
4:  #+begin_src R
5:  library(ggplot2)
6:  x <- data.frame(var = rnorm(100))
7:  ggplot(aes(x = var), data = x)
8:      + geom_histogram()
9:  #+end_src
10:
11:  #+RESULTS:
12:  [[file:f_ggplot2_ex.pdf]]
```

# ggplot2 plot (evaluated)



# Topic

Vorbemerkungen

Einleitung

Problem

Emacs

Org mode

Babel: Active code in Org mode

**Babel und R**

Einführung

**Tangling und weaving**

Ein kurzes vollständiges Org-Beispiel

R und andere Sprachen (Python)

References

# Tangling and weaving code

- ▶ *Weaving* refers to the exportation of a mixed code/prose document to a format that can be read by a human (HTML,  $\text{\LaTeX}$ , Ascii, ...).
- ▶ *Tangling* means to extract only the code to a separate file

(Source: Schulte et al. 2012: 12)



# Beispiel für “code tangling”

```
#+BEGIN_SRC R :tangle ../src/example_tangled.R
## Comment: Assign value 1 to variable x
x <- 1
#+END_SRC
```

# Beispiel (Fortsetzung): Tangled R code

Hier ist der Inhalt der Datei ../src/example\_tangled.R

```
../src/example_tangled.R
```

```
## [[file:e:/projects/confer/ps2012-07-KRUG_org_r/slide/ps2012-0
```

```
## Comment: Assign value 1 to variable x
```

```
x <- 1
```

```
## Example-of-code-tangling:2 ends here
```

# Topic

Vorbemerkungen

Einleitung

Problem

Emacs

Org mode

Babel: Active code in Org mode

**Babel und R**

Einführung

Tangling und weaving

**Ein kurzes vollständiges Org-Beispiel**

R und andere Sprachen (Python)

References

# Ein kurzes vollständiges Org-Beispiel

Siehe hierfür `pub/d_example.org` and  
`pub/d_example.pdf`.

# Topic

Vorbemerkungen

Einleitung

Problem

Emacs

Org mode

Babel: Active code in Org mode

Babel und R

Einführung

Tangling und weaving

Ein kurzes vollständiges Org-Beispiel

**R und andere Sprachen (Python)**

References

# Chaining (source)

```
1 Define variable =y= in a Python code block:
2 #+name: chainexamp
3 #+begin_src python :results value :session *python*
4 y = 9**4
5 y
6 #+end_src
7
8 #+RESULTS: chainexamp
9 : 10000
10
11 Then pass variable =y= to an R code block:
12 #+BEGIN_SRC R :var x=chainexamp :session *R2*
13 print(x)
14 print(x*2)
15 #+END_SRC
16
17 #+RESULTS:
18 : [1] 6561
19 : [1] 13122
```

# Chaining (evaluated)

Define variable `y` in a Python code block:

```
1 y = 10**4  
2 y
```

10000

Then pass variable `y` to an R code block:

```
1 print(x)  
2 print(x*2)
```

[1] 10000

[1] 20000

# Topic

Vorbemerkungen

Einleitung

Problem

Emacs

Org mode

Babel: Active code in Org mode

Babel und R

Einführung

Tangling und weaving

Ein kurzes vollständiges Org-Beispiel

R und andere Sprachen (Python)

References



# References

- ▶ Schulte, Eric, Dan Davison, Thomas Dye, und Carsten Dominik. 2012. A Multi-Language Computing Environment for Literate Programming and Reproducible Research. *Journal of Statistical Software* 46: 1–24.
- ▶ <http://orgmode.org/worg/org-contrib/babel/how-to-use-Org-Babel-for-R.html>