

R für die Sozialwissenschaften - Teil 4

Jan-Philipp Kolb

23 Juni 2017

HTML Dokumente, Präsentationen und Dashboards mit Rmarkdown

Präsentationen - Rpres der einfachste Weg

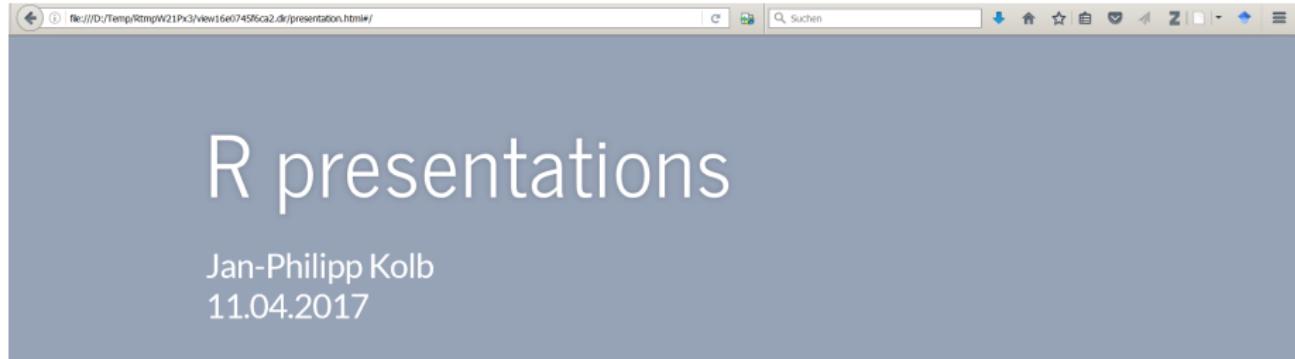


Figure 1

Eine erste Präsentation

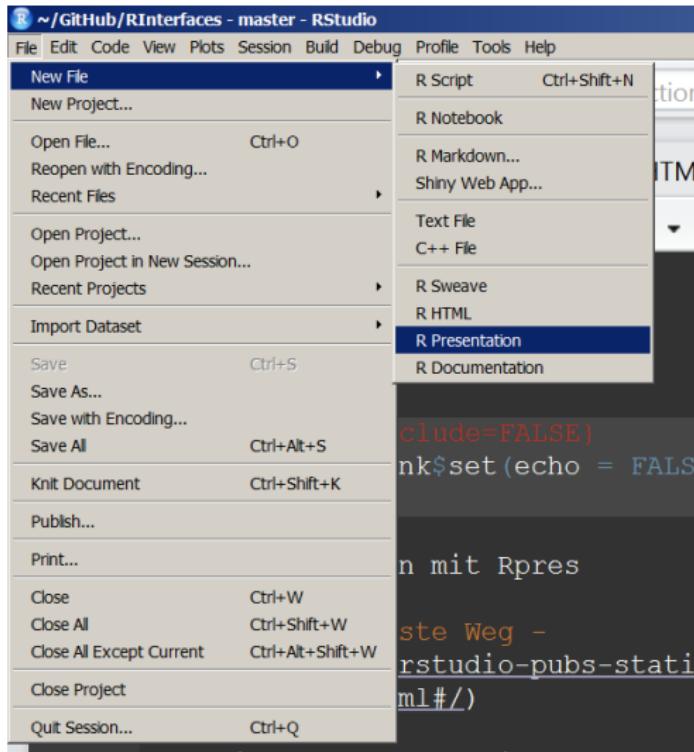


Figure 2

Erste Daten eintragen

- Für Vergessliche:

```
date()
```

```
## [1] "Sat Jul 01 11:29:26 2017"
```

Eine Folie mit Formel

- Die Formel kann wie in LaTeX eingegeben werden

\$\$

```
\begin{equation}\label{eq2}
t_{\{i\}}=\sum\limits_{k=1}^{M_{\{i\}}}{y_{\{ik\}}}=M_{\{i\}}\bar{Y}_{\{i\}}.
\end{equation}
```

\$\$

The screenshot shows the R Studio interface. On the left, the code editor displays the following LaTeX code:

```
29
30 Folie mit LaTeX Code
31 -----
32
33 $$
34 \begin{equation}\label{eq2}
35 t_{\{i\}} =\sum\limits_{k=1}^{M_{\{i\}}}{(y_{\{ik\}})} = M_{\{i\}} \bar{Y}_{\{i\}} .
36 \end{equation}
37 $$
38
```

The status bar at the bottom indicates "Folie mit LaTeX Code" and "R Presentation". On the right, the preview pane shows the rendered LaTeX output:

Folie mit LaTeX Code (5/5)

$$t_i = \sum_{k=1}^{M_i} y_{ik} = M_i \bar{Y}_i$$

Figure 3

Zwei Spalten

Folie mit zwei Spalten

Erste Spalte

Zweite Spalte

Folienübergänge

```
transition: rotate
```

```
1 Meine Erste Präsentation mit Markdown
2 ▾ =====
3 author: Jan-Philipp Kolb
4 date: Thu Apr 20 09:06:19 2017
5 autosize: true
6 transition: rotate|
```

Figure 4

Weitere mögliche Folienübergänge

- none
- linear
- rotate
- fade
- zoom
- concave

Folientypen

Ein neues Kapitel einfügen

=====

type: section

Anderer Folientyp

=====

type: prompt

Noch ein anderer Folientyp

=====

type: alert

Die Schriftart wechseln

- Die CSS Schrifttypen können verwendet werden

Meine Präsentation

author: Jan-Philipp Kolb
font-family: 'Impact'

Schrifttypen können auch importiert werden

Meine Präsentation

author: Jan-Philipp Kolb

font-import: <http://fonts.googleapis.com/css?family=Risque>

font-family: 'Risque'

The screenshot shows a presentation slide with the following details:

- Title Bar:** Shows a house icon, the title "Meine Erste Präsentation mit Markdown...", and several icons for file operations (checkmark, plus, gear).
- Content Area:** Displays the text "Meine Erste Präsentation mit Markdown" in a large, white, serif font.
- Navigation Bar:** At the bottom, it shows "Jan-Philipp Kolb" on the left and "R für die Sozialwissenschaften - Teil 4" in the center.
- Page Number:** On the right side of the navigation bar, it says "23 Juni 2017" and "12 / 93".

Kleineren Text

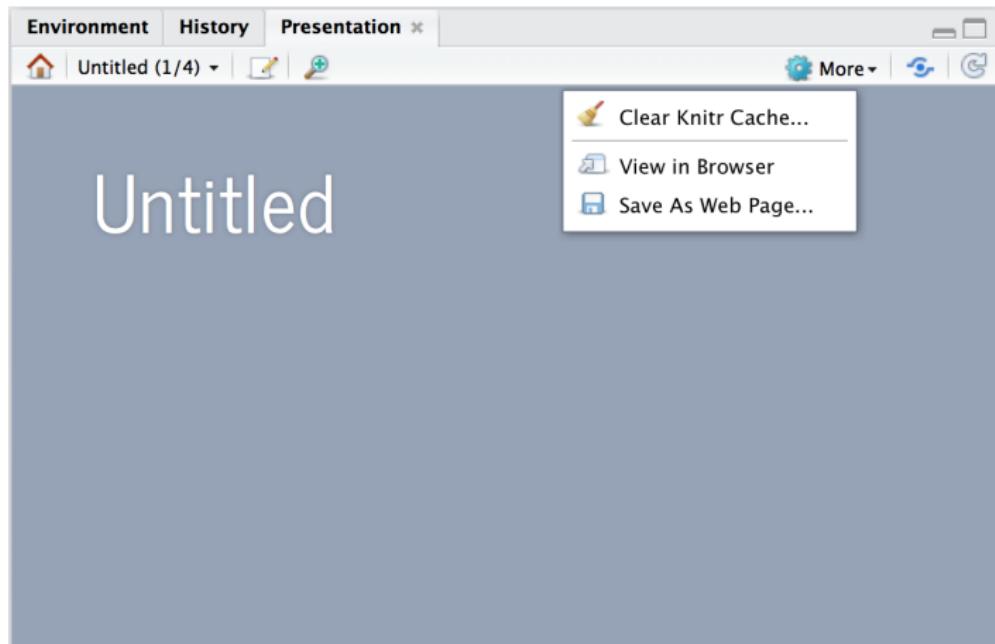
Normale Schriftgröße

<small>This sentence will appear smaller.</small>

Die Präsentation anschauen

- Das Ergebnis ist hier zu sehen:

<http://rpubs.com/Japhilko82/FirstRpubs>



Eine ioslides Präsentation

Eine ioslides Präsentation



Figure 7

ioslides - Der Start

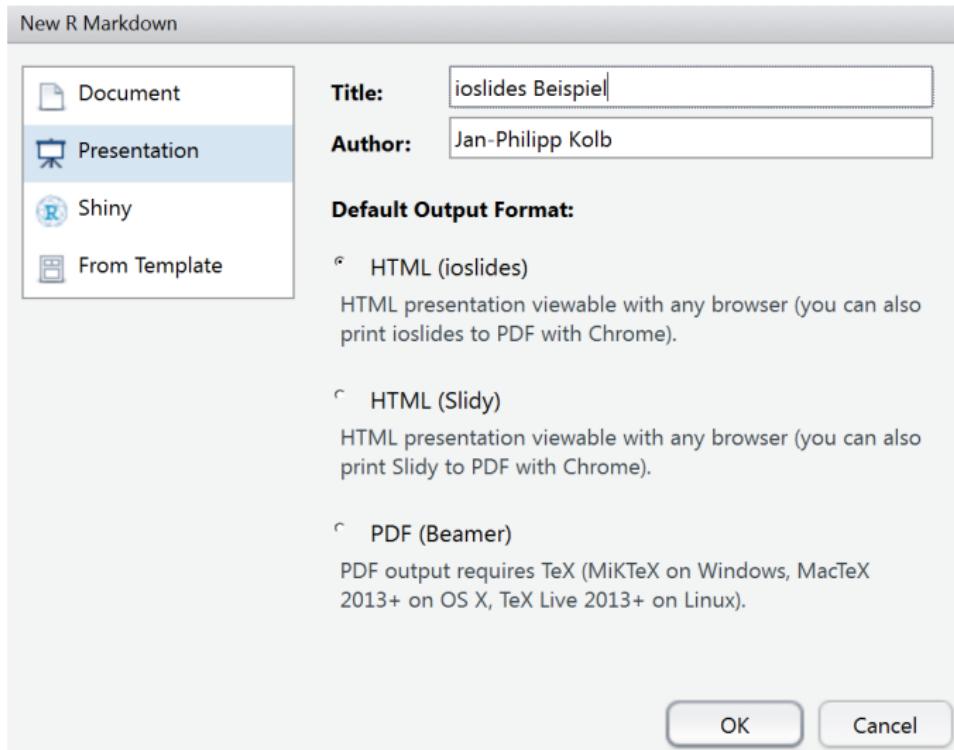


Figure 8

Weitere Dinge tun

- Ein Bild einbinden

! [picture of spaghetti] (images/spaghetti.jpg)

Ein Logo hinzufügen

```
---
```

```
title: "ioslides Beispiel"
author: "Jan-Philipp Kolb"
date: "20 April 2017"
output:
  ioslides_presentation:
    logo: figure/Rlogo.png
---
```

```
1 ---  
2 title: "ioslides Beispiel"  
3 author: "Jan-Philipp Kolb"  
4 date: "20 April 2017"  
5 output:  
6   ioslides_presentation:  
7     logo: figure/Rlogo.png
```

Tabellen

- Quelle: R Studio, and Presentations, and Git! Oh my!

```
library(knitr)
a <- data.frame(a=1:10,b=10:1)
kable(table(a))
```

1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0

knitr Engines

- knitr Language Engines
- slidify

Eine slidy Präsentation

slidy Präsentationen



Präsentationen mit R und Rstudio

Jan-Philipp Kolb

11 April 2017

Contents

slide 1 / 3

Figure 10

[Was sind Cascading Style Files (CSS)]



Figure 11

- Stylesheet-Sprache für elektronische Dokumente
- eine der Kernsprachen des World Wide Webs.
- CSS wurde entworfen, um Darstellungsvorgaben weitgehend von den Inhalten zu trennen

CSS und R

- Custom CSS
- CSS pro tipps

Beispiel CSS

```
1 body, td {  
2   font-family: Lucida Console;  
3   background-color: transparent;  
4   font-size: 20px;  
5 }  
6
```

Figure 12

Das CSS ändern

Um den Präsentationstyp zu ändern kann man das CSS verändern

- Cascading Style Sheets (CSS)
- Bspw. lässt sich die Farbe (HTML) ändern.
- Man kann eine andere Schriftart wählen
- Es gibt zahlreiche Möglichkeiten der Schriftformatierung
- Daneben gibt es viele weitere Dinge, die sich mit dem CSS steuern lassen

HTML Dokumente

Ein HTML Dokument erzeugen

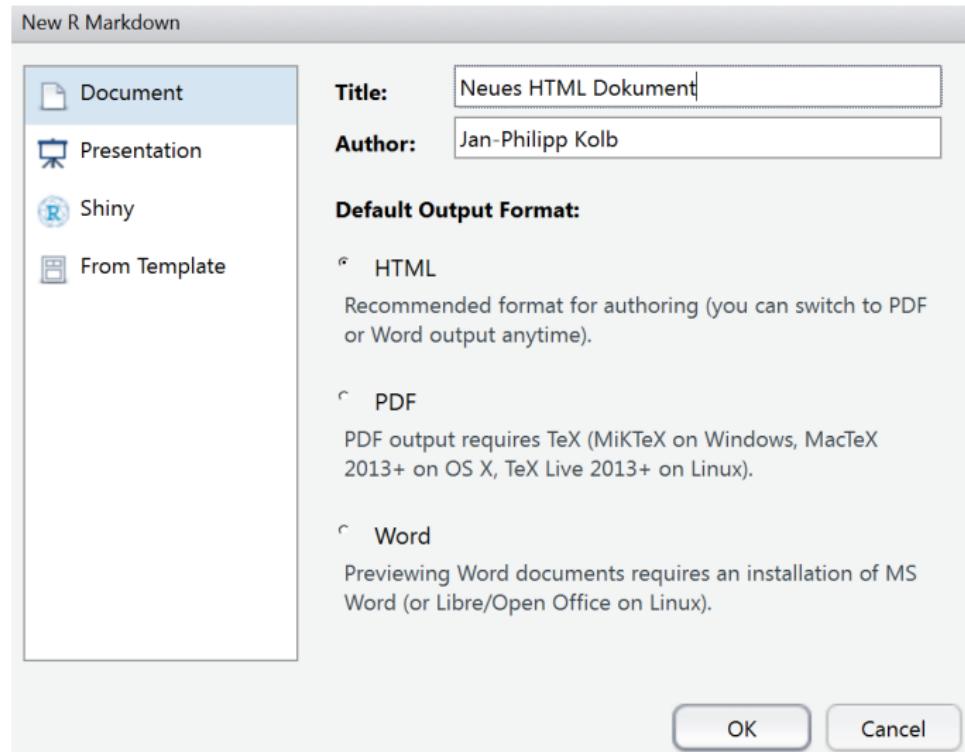


Figure 13

Ein Template verwenden

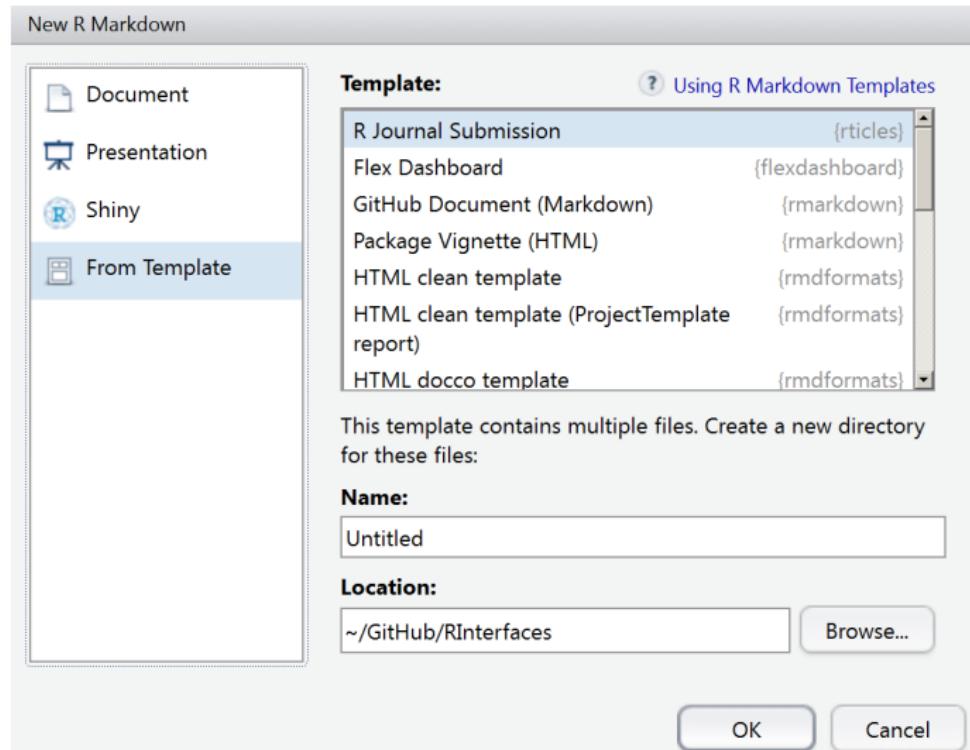


Figure 14

Weitere Vorlagen nutzen

- Es gibt viele Formate - manche müssen erst aktiviert werden:

```
install.packages("rticles")
```

Short Paper

Alice Anonymous
Some Institute of Technology
alice@example.com

Bob Security
Another University
bob@example.com

ABSTRACT

This is the abstract.

It consists of two paragraphs.

1. INTRODUCTION

ut diam. Nulla ut dapibus quam.

Sed est odio, ornare in rutrum et, dapibus in urna. Suspendisse varius massa in ipsum placerat, quis tristique magna consequat. Suspendisse non convallis augue. Quisque fermentum justo et lorem volutpat euismod. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cu-

Figure 15

Vorlagen für Markdown

Das Paket `rmdformats` - HTML Output Formats and Templates for
‘rmarkdown’

```
install.packages("rmdformats")
```

- `ProjectTemplate` - Automates the Creation of New Statistical Analysis

```
install.packages("ProjectTemplate")
```

- `tufte` - Tufte’s Styles for R Markdown Documents

```
install.packages("tufte")
```

Beispiele für Templates

readthedown template example

Code and tables

Figures

MathJax

Code and tables

Syntax highlighting

Here is a sample code chunk, just to show that syntax highlighting works as expected.

```
library(ggplot2)
library(dplyr)

not_null <- function (v) {
  if (!is.null(v)) return(paste(v, "not null"))
}

data(iris)
tab <- iris %>%
  group_by(Species) %>%
  summarise(Sepal.Length = mean(Sepal.Length),
            Sepal.Width = mean(Sepal.Width),
            Petal.Length = mean(Petal.Length),
            Petal.Width = mean(Petal.Width))
```

Verbatim

Here is the structure of the `iris` dataset.

```
str(iris)
```

```
'data.frame': 150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 4.5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",...
```

Figure 16

Dashboards

Beispiel R-Pakete

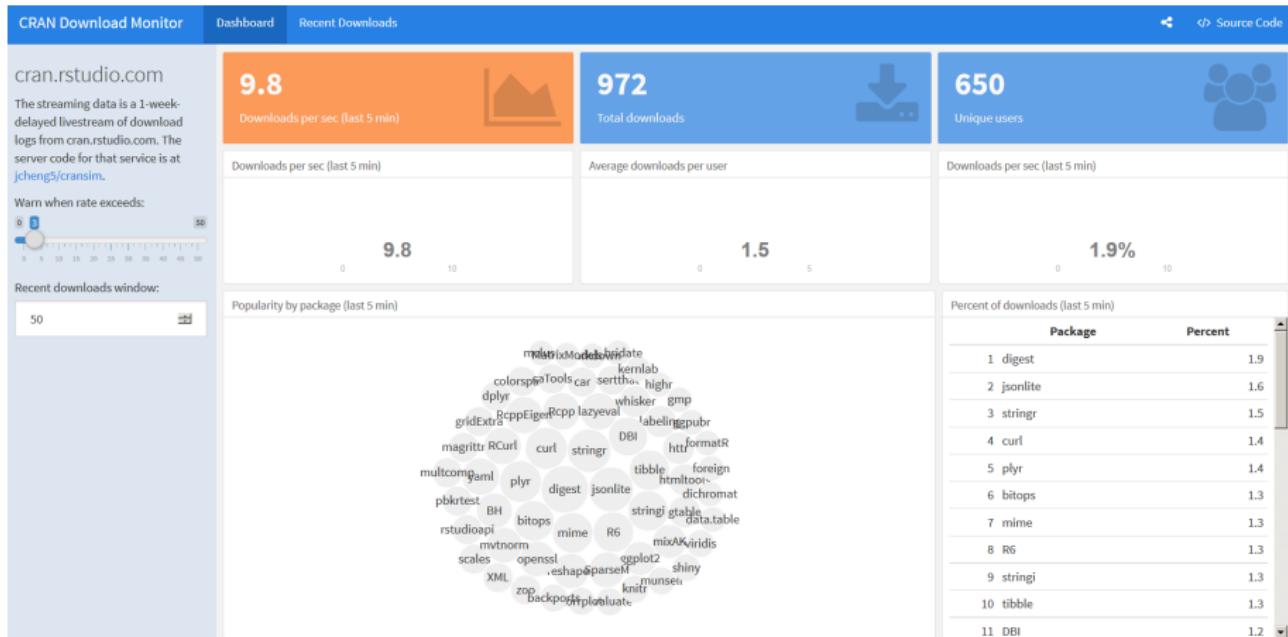


Figure 17

Paket installieren

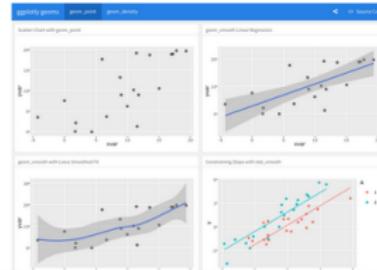
```
install.packages("flexdashboard", type = "source")
```

flexdashboard: Easy interactive dashboards for R

- Use [R Markdown](#) to publish a group of related data visualizations as a dashboard.
- Support for a wide variety of components including [htmlwidgets](#); base, lattice, and grid graphics; tabular data; gauges and value boxes; and text annotations.
- Flexible and easy to specify row and column-based [layouts](#). Components are intelligently re-sized to fill the browser and adapted for display on mobile devices.
- [Storyboard](#) layouts for presenting sequences of visualizations and related commentary.
- Optionally use [Shiny](#) to drive visualizations dynamically.



Jan-Philipp Kolb



R für die Sozialwissenschaften - Teil 4



23 Juni 2017

Ein Dashboard erstellen mit Rstudio

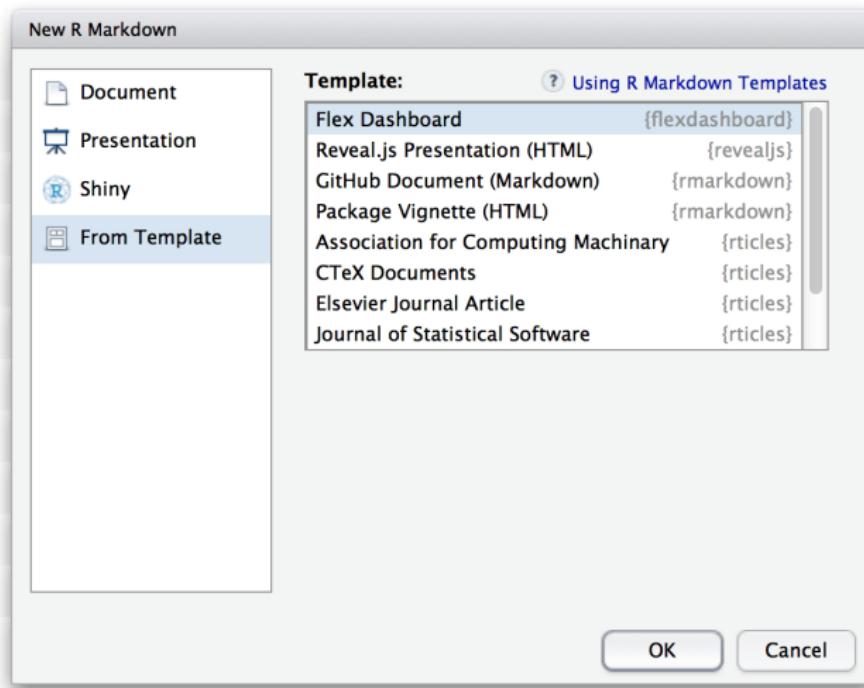


Figure 19

Mein erstes Dashboard

RPubs brought to you by RStudio Japhniko82

Mein erstes Dashboard

Eine interaktive Datentabelle mit [dt](#)

Show 10 entries

	name_en	category	longitude	latitude	date_inscribed	area_hectares	danger_list
1	Cultural Landscape and Archaeological Remains of the Bamiyan Valley	Cultural	67.82525	34.84694	2003	158.9265	Y 2003
2	Minaret and Archaeological Remains of Jam	Cultural	64.51605556	34.39655556	2002	70	Y 2002
3	Historic Centres of Berat and Gjirokastra	Cultural	20.13333333	40.06944444	2005	58.9	
4	Butrint	Cultural	20.02611111	39.75111111	1992		
5	Al Qal'a of Beni Hammad	Cultural	4.78684	35.81844	1980	150	
6	M'Zab Valley	Cultural	3.68333	32.48333	1982	665.03	
7	Djââmila	Cultural	5.73667	36.32056	1982	30.6	
8	Timgad	Cultural	6.63333	35.45	1982	90.54	
9	Kasbah of Algiers	Cultural	3.06028	36.78333	1992	60	

Showing 1 to 10 of 1,031 entries

Previous 1 2 3 4 5 ... 104 Next

Eine einfache [ggplot2](#) Graphik

whcSiteCategory	Count
cultural	~850
natural	~150
mixed	~20

Eine Karte mit [leaflet](#)

Leaflet | © OpenStreetMap contributors, CC-BY-SA

[Edit Details](#) [Delete](#) Mein erstes Dashboard by Jan-Philipp Kolb Last updated 1 minute ago [Comments \(-\)](#) [Share](#) [Hide Toolbars](#)

Figure 20

Gallerie

Documents

With R Markdown, you write a single .Rmd file and then use it to render finished output in a variety of formats.

<p>Great NYT Interactive -- Now Reusable with rCharts</p> <p>Disclaimer and Attribution</p> <p>I am absolutely honored by this invitation, which consider one of the most important web sites, as contributing to its capital news coverage. I am also honored that the site has chosen to do this using a completely working approach for the first day of the site.</p> <p>Another Plot from NYT</p> <p>It's been a pleasure to have my work used in such a prominent setting. I hope that this will encourage many more people to use rCharts to play with the statistics after their first few weeks.</p> <p></p>	<p>A Pandoc Markdown Article Starter and Template*</p> <p>Section 1: Introduction</p> <p>This template generates an article in R Markdown, designed for the benefits, and generates a simple manuscript template intended for an academic audience. It includes basic syntax in R Markdown, and provides a simple example of how the analysis itself can be conducted within R with the source package.</p> <p>Introduction</p> <p>Academic writing, especially in political science, is at a crossroads. The American Journal of Political Science, for example, has recently decided to require authors to submit their research in a digital format, and to publish the results online in the electronic. The submission at AJPS has redefined the manuscript as a "document" rather than a "paper". This might be the best of the intentions, and it could be the most appropriate. But it is also a challenge to the traditional manuscript, which is a well-defined, well-organized, corrected, polished article in a linear direction bound to the printed or digital page.</p> <p>There are wonderful implications in the LaTeX community as well. Political science, for the most part, is a discipline that has not yet adopted LaTeX as its primary document format, if ever. In fact, the LaTeX community has a long history of resistance to the adoption of LaTeX, and the reasons are varied.</p> <p>PDF</p> <p>PDF documents for printing. Example Code</p>	<p>A Microsoft Word document</p> <p>Microsoft Word</p> <p>R Markdown</p> <p>This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For a detailed introduction, see Getting Started.</p> <p>When you click the Knit button a document in HTML is generated that includes both content as well as all the necessary R code chunks within the document. You can edit and re-run the code chunks.</p> <p>Including Plots</p> <p>You can also embed R plots in the document.</p>	<p>Tufte Handout</p> <p>An implementation in R Markdown of the Tufte style for documents.</p> <p>Introduction</p> <p>The Tufte function will create the Microsoft Word file in the background. With each release for the Tufte package, the Microsoft Word file is updated to reflect the latest version of Microsoft Word and its capabilities. Thus far, the last update was released in March 2013. We have provided implementations for the old package. If you prefer the new version, please use the Tufte function from the tufte package, and set the name for the Microsoft Word file to "tufte.docx". The Microsoft Word file will be generated in the current working directory. If you specified the "TUFTE" variable as the beginning of the file name, then the Microsoft Word file will be placed in the root folder. See R documentation for more information about <code>tufte</code>.</p> <p>Handouts</p> <p>Tufte styled documents for handouts. Example Code</p>
--	---	--	---

Interactive Documents

Combine R Markdown with htmlwidgets or the shiny package to make interactive documents.

<p></p> <p>HTML Widgets</p> <p>Add interactive graphics with htmlwidgets, such as the leaflet map widget.</p>	<p>UNCCC Data Report</p> <p>1 Disclaimer</p> <p>2 Data Overview</p> <p>These data are available for download in a number of formats. The package</p>	<p>Shiny leaflet example</p> <p>Version 0 - User Observers</p> <p>Shiny is a framework for creating web applications with R. It is based on the R Markdown language, which makes it easy to mix R code and text, and to generate dynamic reports and dashboards.</p> <p>Shiny</p> <p>Add interactive analysis with shiny, which lets your user rerun the actual analysis within your report.</p>	<p></p> <p>Shiny</p> <p>Shiny components and htmlwidgets will work in any HTML-based output, such as a file, slide show or dashboard.</p>
--	---	---	--

Figure 21

Links

- Verschiedene Markdown Dokumente zusammen fügen



55



I'm not sure this is exactly what you're looking for, but when I want to break a large report into separate Rmd, I usually create a parent Rmd and include the chapters as children. This approach is also easy for new users to understand. It doesn't create a nice title for each chapter, but as long as you include a toc, it is easy to navigate between chapters. One pitfall doing this is that all chunk names between all parent/children need to be unique.

report.Rmd

```
---
```

```
title: My Report
```

```
output:
```

```
  pdf_document:
```

```
    toc: yes
```

```
---
```

```
```{r child = 'chapter1.Rmd'}
```

```
```
```

```
```{r child = 'chapter2.Rmd'}
```

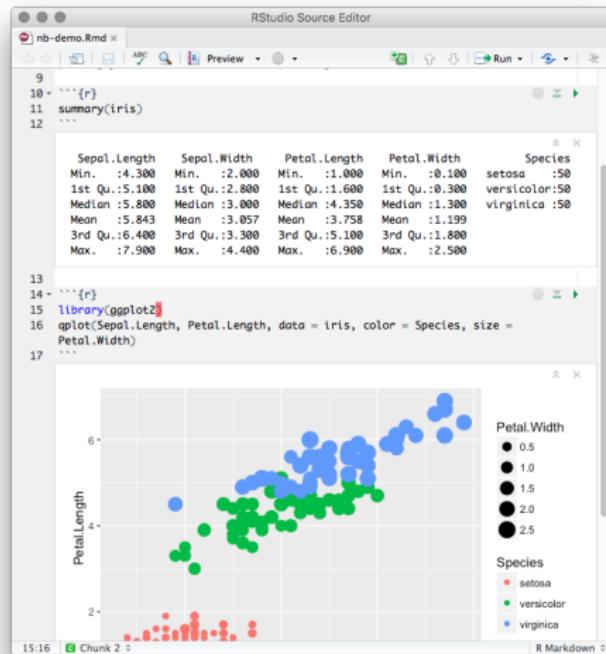
```
```
```

Figure 22

Notebooks

Notebooks

- Warum R Notebook nutzen



Ein Rnotebook anlegen

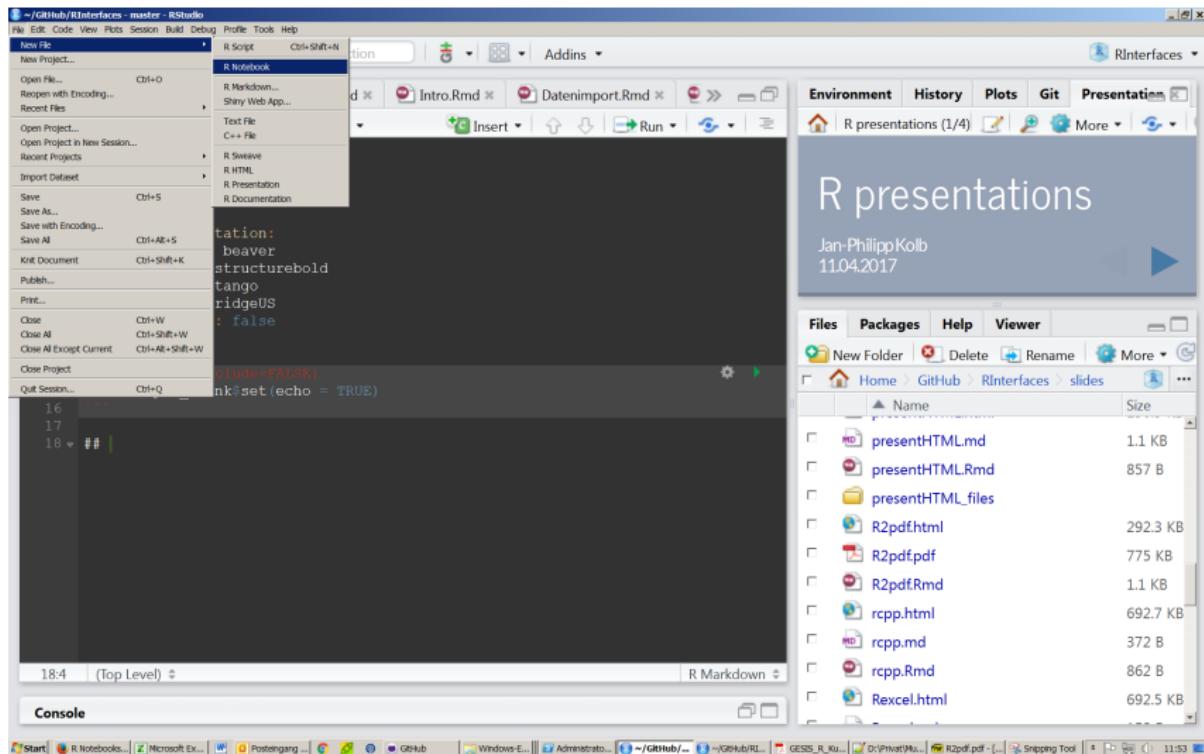
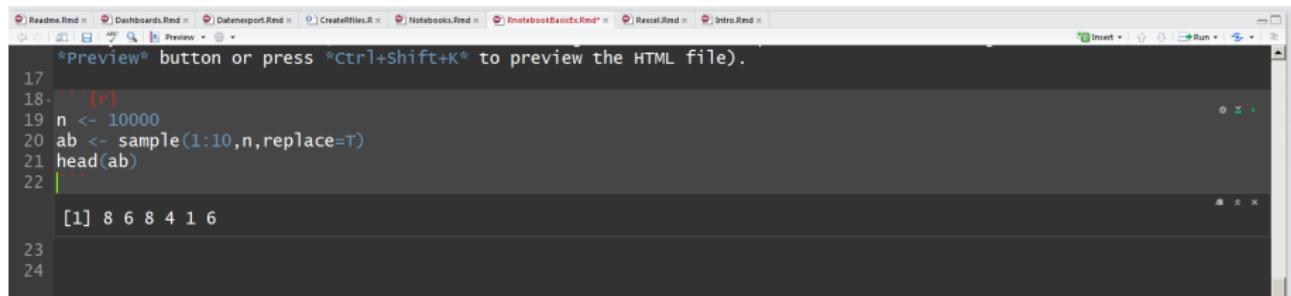


Figure 24

Rnotebook - erste Schritte

- Es lassen sich so genannte Chunks einfügen
- In diesen Chunks wird ganz normaler R-code geschrieben



The screenshot shows the RStudio interface with multiple tabs at the top. The active tab is titled "Notebooks.Rmd". Below the tabs, there is a preview area with the text: "*Preview* button or press *Ctrl+shift+k* to preview the HTML file.". The main code editor area contains the following R code:

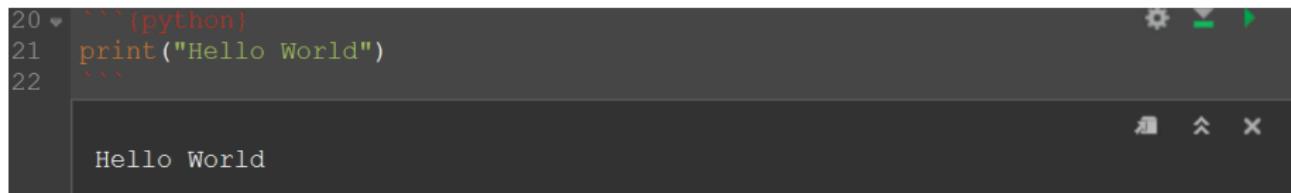
```
17
18: ````{r}
19: n <- 10000
20: ab <- sample(1:10,n,replace=T)
21: head(ab)
22:
[1] 8 6 8 4 1 6
23
24
```

The output of the code is shown in the console below the code editor.

Figure 25

Python Code integrieren

- Ebenso lässt sich Python code implementieren



The screenshot shows a dark-themed code editor window titled 'Rnotebooks'. In the top left, there are line numbers 20, 21, and 22. Lines 20 and 21 contain Python code: 'print("Hello World")'. Line 22 is a blank line. To the right of the code area are several icons: a gear, a downward arrow, a green checkmark, a red X, a file icon, a double arrow icon, and another X icon. Below the code area, the output 'Hello World' is displayed in a light gray box.

Figure 26

```
import sys  
print(sys.version)
```

```
## 2.7.10 (default, May 23 2015, 09:44:00) [MSC v.1500 64 bit]
```

LaTeX Code integrieren

- LaTeX code wird mit zwei Dollarzeichen gekennzeichnet

```
$$  
\alpha = \frac{\beta}{\lambda}  
$$
```

$$\alpha = \frac{\beta}{\lambda}$$



Figure 27

Notebook veröffentlichen I

The screenshot shows the R Notebook interface. At the top, there is a menu bar with tabs: Files, Packages, Help, and Viewer. Below the menu bar is a toolbar with icons for file operations like Open, Save, and Print, as well as a refresh icon. On the right side of the interface, there is a vertical sidebar with a scroll bar. The main content area displays the title "R Notebook" and the subtitle "R code inline". A "Code" dropdown menu is open. Below the subtitle, there is a "Hide" button. A code block contains the R command "plot(cars)". At the bottom of the main content area, there is a bulleted list: "• Other language engines".

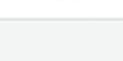
Figure 28

Notebook veröffentlichen II

Publish

Publish To

Rpubs  RPubs is a free service from RStudio for sharing documents on the web. >

RStudio Connect  RStudio Connect is a server product from RStudio for secure sharing of applications, reports, and plots. >

Cancel

Figure 29

Andere Notebooks

Jupyter Notebook

- Anaconda installieren
- folgenden Befehl in die Eingabeaufforderung eingeben
- Bei Windows findet man diese, wenn man cmd in Suche eingibt.

```
jupyter notebook
```

Start Jupyter Notebook



Files Running Clusters

Select items to perform actions on them.

Upload New ▾ ⚙

- Anaconda3
- AppData

Figure 30

Beispiel Eingabe Code

jupyter An Jupyter R notebook Last Checkpoint: 04/27/2016 (unsaved changes)

File Edit View Insert Cell Kernel Help

In [2]: `library(ggmap)`
Loading required package: ggplot2

In [3]: `qmap ("Mannheim")`
Map from URL : <http://maps.googleapis.com/maps/api/staticmap?center=Mannheim&zoom=10&size=640x640&scale=2&maptype=terrain&language=en-EN&sensor=false>
Information from URL : <http://maps.googleapis.com/maps/api/geocode/json?address=Mannheim&sensor=false>

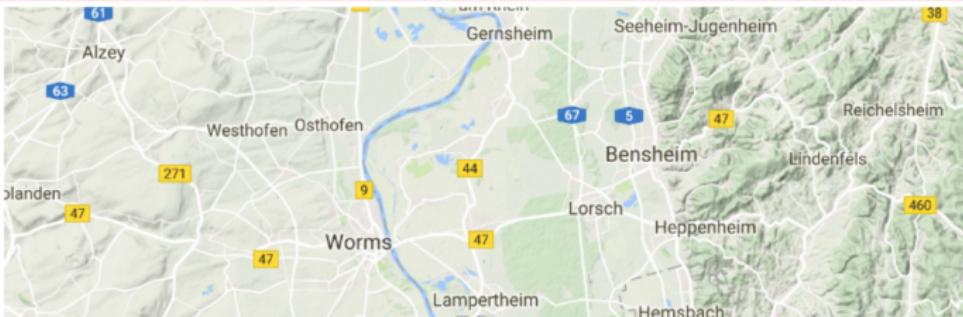


Figure 31

Beaker Notebook

Beaker Notebook

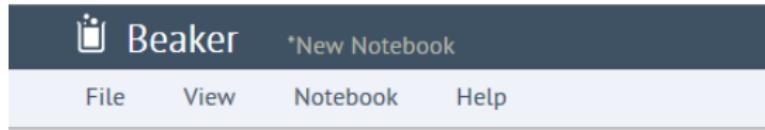
- Auch bei Beaker kann man R-code einbauen



Figure 32

Beaker starten

- Beaker installieren ...
- ... und mit beaker.command.bat starten



The screenshot shows the Beaker notebook interface. At the top, there are two buttons: a blue one labeled "R" and a green one labeled "R ▾". Below them is a text input field containing the R code: `1 | sample(1:10, 5, replace=T)`. To the right of the input field is a "Run" button. The output section below displays the result of the code execution: `[1] 10 6 3 5 9`.

Links

- knitr Language Engines
- More engines
- Andere Programmiersprachen können eingebunden werden
- Video - Einführung in Rnotebook
- R Notebooks
- IPython vs knitr, or Python vs R
- Datacamp Tutorial - Jupyter Notebook
- Better interactive data science with Beaker and Rodeo
- Knit directly to jupyter notebooks from RStudio
- Python-Markdown
- Podcast - die Welt von Python kennenlernen
- Deploying JupyterHub for Education
- JupyterHub - github
- Jupyter autograder

Interaktive Tabellen mit DataTables

The R-package DT

- DT: An R interface to the DataTables library

```
install.packages('DT')
```

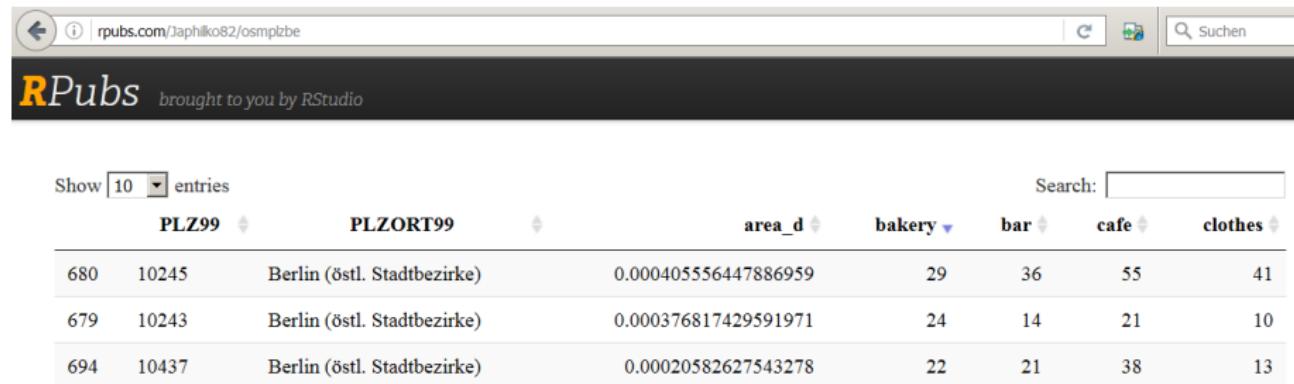
```
library('DT')
```

```
exdat <- read.csv("data/exdat.csv")
```

```
datatable(exdat)
```

Beispiel für interaktive Tabelle

Hier ist das Ergebnis - Beispiel für eine interaktive Tabelle



The screenshot shows a web browser displaying an Rpubs page. The URL in the address bar is rpubs.com/Japhilko82/osmplzbe. The page title is "RPub" followed by "brought to you by RStudio". There are navigation icons for back, forward, and search, along with a "Suchen" (Search) button. A dropdown menu shows "Show 10 entries". A search bar is present. The main content is a data table with the following columns: PLZ99, PLZORT99, area_d, bakery, bar, cafe, and clothes. The data rows are:

| | PLZ99 | PLZORT99 | area_d | bakery | bar | cafe | clothes |
|-----|-------|-----------------------------|----------------------|--------|-----|------|---------|
| 680 | 10245 | Berlin (östl. Stadtbezirke) | 0.000405556447886959 | 29 | 36 | 55 | 41 |
| 679 | 10243 | Berlin (östl. Stadtbezirke) | 0.000376817429591971 | 24 | 14 | 21 | 10 |
| 694 | 10437 | Berlin (östl. Stadtbezirke) | 0.00020582627543278 | 22 | 21 | 38 | 13 |

Figure 34

Default Optionen verändern

```
datatable(head(exdat, 20), options = list(  
  columnDefs = list(list(className = 'dt-center', targets = 5)  
  pageLength = 5,  
  lengthMenu = c(5, 10, 15, 20)  
))
```

Suchoptionen kennzeichnen

```
datatable(exdat, options = list(searchHighlight = TRUE),
         filter = 'top')
```

Show entries

Search:

| | mpg | cyl | disp | hp | drat | model |
|-----|------|-----|------|-----|------|-------------------|
| All | All | All | All | All | All | All |
| 1 | 21 | 6 | 160 | 110 | 3.9 | Mazda RX4 |
| 2 | 21 | 6 | 160 | 110 | 3.9 | Mazda RX4 Wag |
| 3 | 22.8 | 4 | 108 | 93 | 3.85 | Datsun 710 |
| 4 | 21.4 | 6 | 258 | 110 | 3.08 | Hornet 4 Drive |
| 5 | 18.7 | 8 | 360 | 175 | 3.15 | Hornet Sportabout |

Showing 1 to 5 of 20 entries

Previous

1

2

3

4

Next

Interaktive Karten mit dem Javascript Paket leaflet

Die Daten - Weltkulturerbe

- die Daten einlesen:

```
url <- "https://raw.githubusercontent.com/Japhilko/  
GeoData/master/2015/data/whcSites.csv"  
  
whcSites <- read.csv(url)
```

- die Daten werden eingeschränkt:

```
whcSitesDat <- with(whcSites, data.frame(name_en,  
category))
```

Eine Tabelle erzeugen mit knitr

```
library(knitr)
kable(head(whcSitesDat))
```

name_en

| name_en | cate |
|---|------|
| Cultural Landscape and Archaeological Remains of the Bamiyan Valley | Cult |
| Minaret and Archaeological Remains of Jam | Cult |
| Historic Centres of Berat and Gjirokastra | Cult |
| Butrint | Cult |
| Al Qal'a of Beni Hammad | Cult |
| M'Zab Valley | Cult |

Eine erste interaktive Tabelle - Das Paket DT

```
install.packages("DT")
```

DT: An R interface to the DataTables library

The R package **DT** provides an R interface to the JavaScript library **DataTables**. R data objects (matrices or data frames) can be displayed as tables on HTML pages, and **DataTables** provides filtering, pagination, sorting, and many other features in the tables.

You may install the stable version from CRAN, or the development version using `devtools::install_github('rstudio/DT')` if necessary (this website reflects the development version of DT):

Figure 36

Weitere Variablen WHC Datensatz

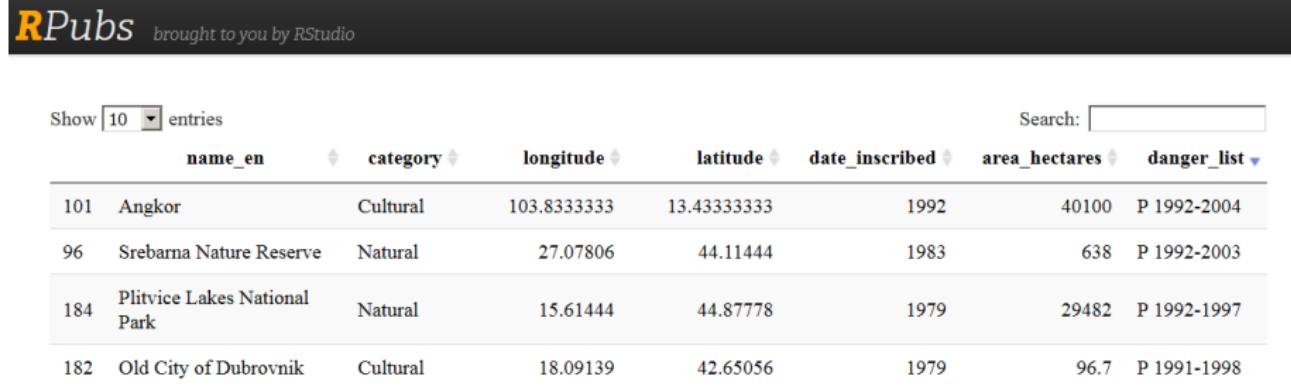
```
whcSitesDat2 <- with(whcSites,data.frame(name_en,category,  
                                             longitude,latitude,da
```

- mit dem Befehl datatable kann man eine erste interaktive Tabelle erstellen:

```
library('DT')  
datatable(whcSitesDat2)
```

Das Ergebnis bei Rpubs

<http://rpubs.com/Japhilko82/WHCdata>



The screenshot shows an Rpubs page with a dark header containing the R logo and the text "brought to you by RStudio". Below the header is a search bar with the placeholder "Search: []". The main content is a data table with the following columns: name_en, category, longitude, latitude, date_inscribed, area_hectares, and danger_list. The table contains five rows of data.

| | | Show <input type="button" value="10"/> entries | Search: [] | | | | | |
|-----|------------------------------|--|-------------|-------------|-------------|----------------|---------------|-------------|
| | | name_en | category | longitude | latitude | date_inscribed | area_hectares | danger_list |
| 101 | Angkor | | Cultural | 103.8333333 | 13.43333333 | 1992 | 40100 | P 1992-2004 |
| 96 | Srebarna Nature Reserve | | Natural | 27.07806 | 44.11444 | 1983 | 638 | P 1992-2003 |
| 184 | Plitvice Lakes National Park | | Natural | 15.61444 | 44.87778 | 1979 | 29482 | P 1992-1997 |
| 182 | Old City of Dubrovnik | | Cultural | 18.09139 | 42.65056 | 1979 | 96.7 | P 1991-1998 |

Figure 37

Das Paket magrittr

- magrittr - für den Pipe Operator in R:

```
install.packages("magrittr")
```

```
library("magrittr")
```

Simpler R coding with pipes > the present and future of the magrittr package



Tal Galili

August 5, 2014

0
SHARES

f Share

t Tweet

e Subscribe

This is a guest post by Stefan Milton, the author of the [magrittr](#) package which introduces the `%>%` operator to R programming.

Die Pipes nutzen

```
library(magrittr)

str1 <- "Hallo Welt"
str1 %>% substr(1,5)

## [1] "Hallo"

str1 %>% substr(1,5) %>% toupper()

## [1] "HALLO"
```

Das Paket leaflet

- leaflet - um interaktive Karten mit der JavaScript Bibliothek leaflet zu erzeugen

```
install.packages("leaflet")
```

```
library("leaflet")
```

- Bei leaflet wird mit so genannten Tiles gearbeitet.
- Robin Lovelace - The leaflet package for online mapping in R

Was sind Tiles?

- Die Übersetzung aus dem englischen ist Fliese und dieses Bild erklärt es eigentlich ganz gut.
- Es geht um Kachelgrafiken.
- Es ist eine Grafik bezeichnet, die mosaikartig zusammengesetzt ein vielfach größeres Gesamtbild ergibt.

Eine interaktive Karte erstellen

```
m <- leaflet() %>%
  addTiles() %>% # Add default OpenStreetMap map tiles
  addMarkers(lng=whcSites$lon,
             lat=whcSites$lat,
             popup=whcSites$name_en)
m
```

Die Karte zeigen



Figure 39

Farbe hinzufügen

- die unterschiedlichen Kategorien farblich einfärben

```
whcSites$color <- "red"  
whcSites$color[whcSites$category=="Cultural"] <- "blue"  
whcSites$color[whcSites$category=="Mixed"] <- "orange"
```

Eine Karte mit Farbe erzeugen

```
m1 <- leaflet() %>%  
  addTiles() %>%  
  addCircles(lng=whcSites$lon,  
             lat=whcSites$lat,  
             popup=whcSites$name_en,  
             color=whcSites$color)
```

Die Karte mit mehr Farbe

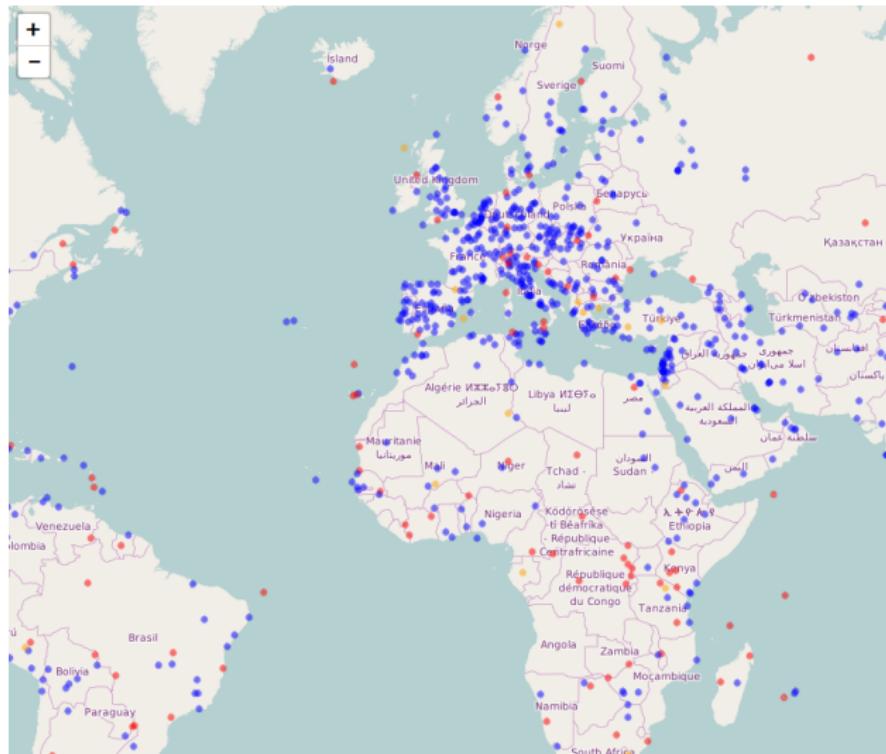


Figure 40: Weltkulturerbe

Die Karte abspeichern

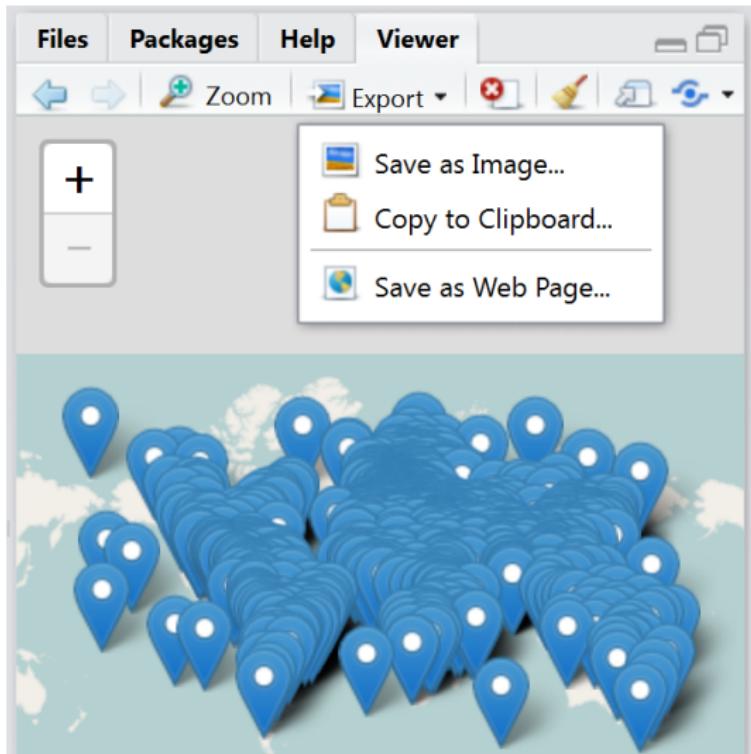


Figure 41

Layers ein- und ausblenden

```
m2 <- leaflet() %>%
  addTiles(group = "OSM (default)") %>%
  addProviderTiles("Stamen.Toner", group = "Toner") %>%
  addProviderTiles("Stamen.TonerLite", group = "Toner Lite") %

  addCircles(lng=whcSites$lon,
             lat=whcSites$lat,
             popup=whcSites$name_en) %>%

  addLayersControl(
    baseGroups = c("OSM (default)", "Toner", "Toner Lite"),
    options = layersControlOptions(collapsed = FALSE)
  )
m2
```

Ein weiteres Beispiel mit Erdbebendaten

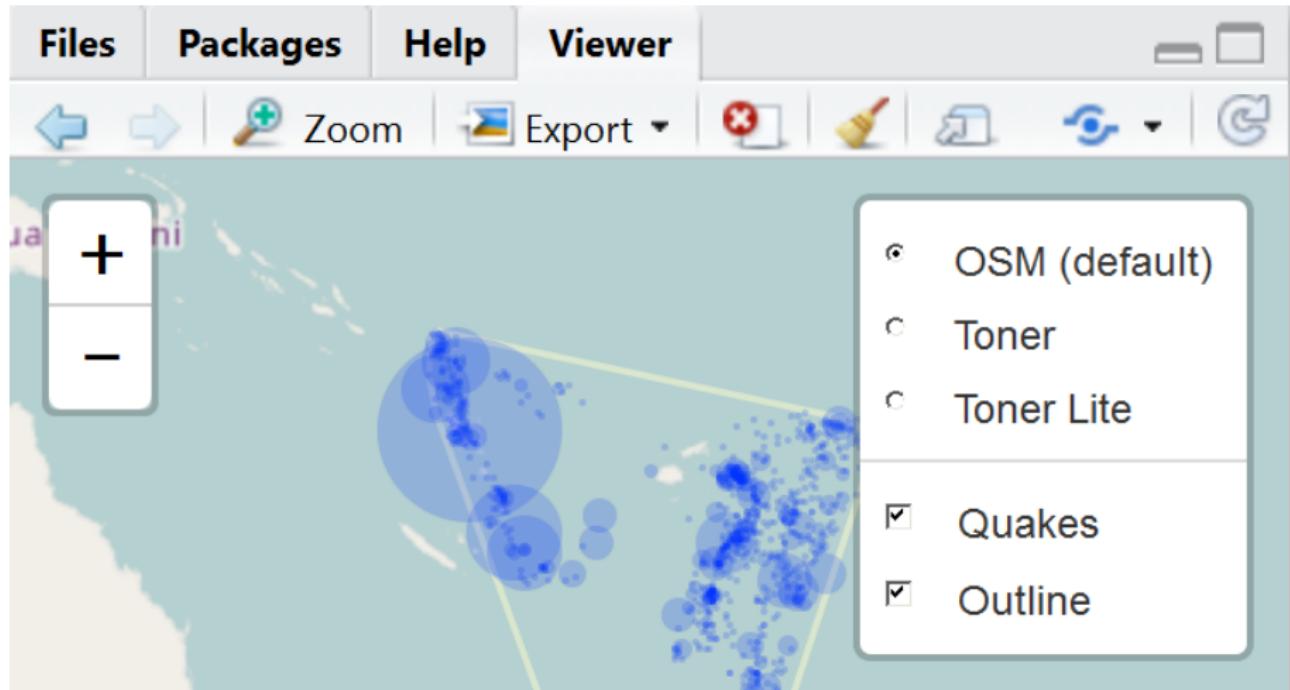


Figure 43

Karte mit Polygonen erzeugen

```
library(sp)
Sr1 = Polygon(cbind(c(2, 4, 4, 1, 2), c(2, 3, 5, 4, 2)))
Sr2 = Polygon(cbind(c(5, 4, 2, 5), c(2, 3, 2, 2)))
Sr3 = Polygon(cbind(c(4, 4, 5, 10, 4), c(5, 3, 2, 5, 5)))
Sr4 = Polygon(cbind(c(5, 6, 6, 5, 5), c(4, 4, 3, 3, 4)), hole=TRUE)
Srs1 = Polygons(list(Sr1), "s1")
Srs2 = Polygons(list(Sr2), "s2")
Srs3 = Polygons(list(Sr4, Sr3), "s3/4")
SpP = SpatialPolygons(list(Srs1, Srs2, Srs3), 1:3)
```

- so wird die Karte erzeugt:

```
leaflet(height = "300px") %>% addPolygons(data = SpP)
```

Beispiel US Staaten

```
library(maps)
mapStates = map("state", fill = TRUE, plot = FALSE)
leaflet(data = mapStates) %>% addTiles() %>%
  addPolygons(fillColor = topo.colors(10, alpha = NULL), strok
```

Der Befehl setView

- mit setView kann man bestimmen welchen Ausschnitt man für die Hintergrundkarte haben möchte
- dazu muss man die latitude und Longitude Koordinaten und ein zoom Level angegeben
- dabei kann man nur ganze Zahlen angeben
- je kleiner die Zahl, desto größer ist der Kartenausschnitt:
- level 3 - Kontinent
- level 10 - Stadt
- level 21 - Gebäude

Die Basiskarte ändern

- Neben der Default Basiskarte kann man auch andere Hintergründe aktivieren

```
m <- leaflet() %>% setView(lng = -71.0589, lat = 42.3601, zoom = 13)
m %>% addTiles()
m %>% addProviderTiles("Stamen.Toner")
```

Basiskarte - CartoDB

```
m %>% addProviderTiles("CartoDB.Positron")
```

Esri.NatGeoWorldMap

```
m %>% addProviderTiles("Esri.NatGeoWorldMap")
```

OpenTopoMap

```
m %>% addProviderTiles("OpenTopoMap")
```

Thunderforest.OpenCycleMap

```
m %>% addProviderTiles("Thunderforest.OpenCycleMap")
```

WMS Tiles hinzufügen

```
leaflet() %>% addTiles() %>% setView(-93.65, 42.0285, zoom = 4  
addWMSTiles(  
  "http://mesonet.agron.iastate.edu/cgi-bin/wms/nexrad/n0r.  
  layers = "nexrad-n0r-900913",  
  options = WMSTileOptions(format = "image/png", transparent = TRUE),  
  attribution = "Weather data © 2012 IEM Nexrad"  
)
```

Mehrere Layer miteinander kombinieren

```
m %>% addProviderTiles("MtbMap") %>%  
  addProviderTiles("Stamen.TonerLines",  
    options = providerTileOptions(opacity = 0.35)) %>%  
  addProviderTiles("Stamen.TonerLabels")
```

Andere Marker benutzen

```
greenLeafIcon <- makeIcon(  
  iconUrl = "http://leafletjs.com/examples/custom-icons/leaf-g  
  iconWidth = 38, iconHeight = 95,  
  iconAnchorX = 22, iconAnchorY = 94,  
  shadowUrl = "http://leafletjs.com/examples/custom-icons/leaf-g  
  shadowWidth = 50, shadowHeight = 64,  
  shadowAnchorX = 4, shadowAnchorY = 62  
)  
  
leaflet(data = quakes[1:4,]) %>% addTiles() %>%  
  addMarkers(~long, ~lat, icon = greenLeafIcon)
```

Andere Icons einfügen

- es lassen sich alle möglichen Icons einfügen

```
menIcon <- makeIcon("https://img.clipartfest.com/707b339dc88f5  
    iconWidth = 38, iconHeight = 95,  
    iconAnchorX = 22, iconAnchorY = 94)  
  
leaflet(data = quakes[1:4,]) %>% addTiles() %>%  
    addMarkers(~long, ~lat, icon = menIcon)
```

Cluster Optionen für Marker

```
leaflet(quakes) %>% addTiles() %>% addMarkers(  
  clusterOptions = markerClusterOptions()  
)
```

Ein Rechteck hinzufügen

```
leaflet() %>% addTiles() %>%
  addRectangles(
    lng1=-118.456554, lat1=34.078039,
    lng2=-118.436383, lat2=34.062717,
    fillColor = "transparent"
)
```

Links und Quellen

- 4 Tricks zum Arbeiten mit Leaflet
- [http://www.r-bloggers.com/
the-leaflet-package-for-online-mapping-in-r/](http://www.r-bloggers.com/the-leaflet-package-for-online-mapping-in-r/)
- <https://rstudio.github.io/leaflet/>