R für die Sozialwissenschaften - Teil 4

Jan-Philipp Kolb

23 Juni 2017

HTML Dokumente, Präsentationen und Dashboards mit Rmarkdown

Präsentationen - Rpres der einfachste Weg

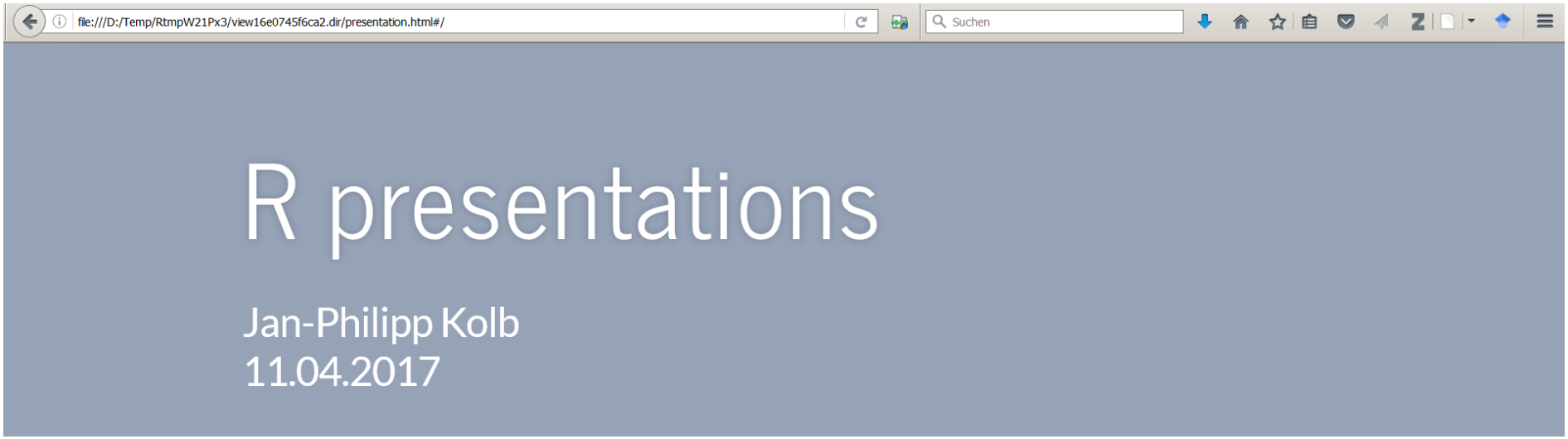


Figure 1

Eine erste Präsentation

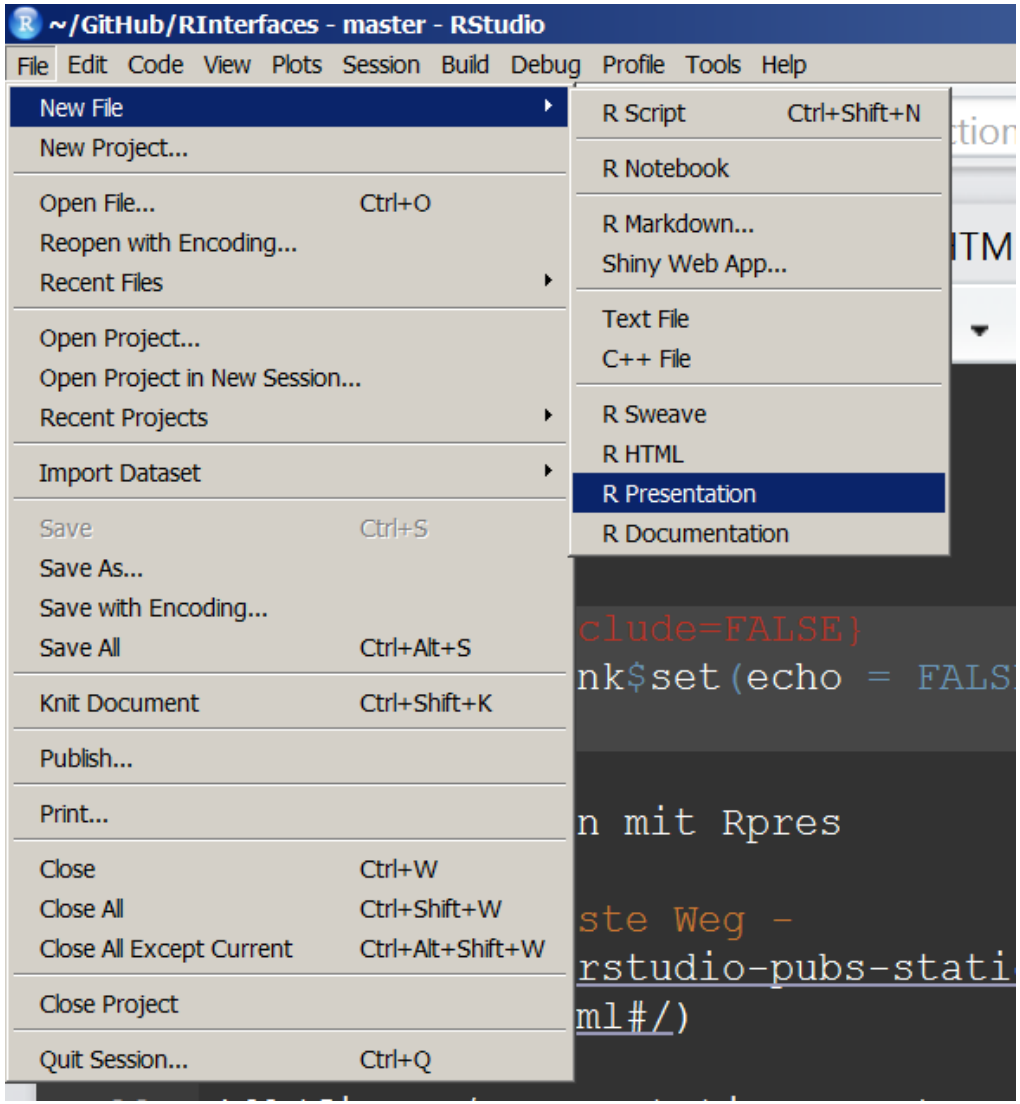


Figure 2

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

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

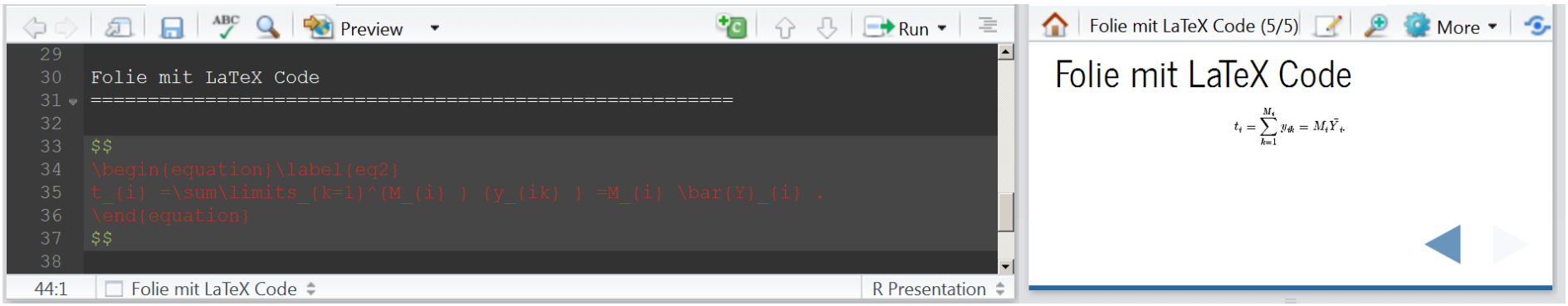


Figure 3

Zwei Spalten

```
Folie mit zwei Spalten
======================================
Erste Spalte
***
Zweite Spalte
```

Folienübergänge

`transition: rotate`

```
Meine Erste Präsentation mit Markdown
=====================================================
author: Jan-Philipp Kolb
date: Thu Apr 20 09:06:19 2017
autosize: true
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'
```

Kleineren Text

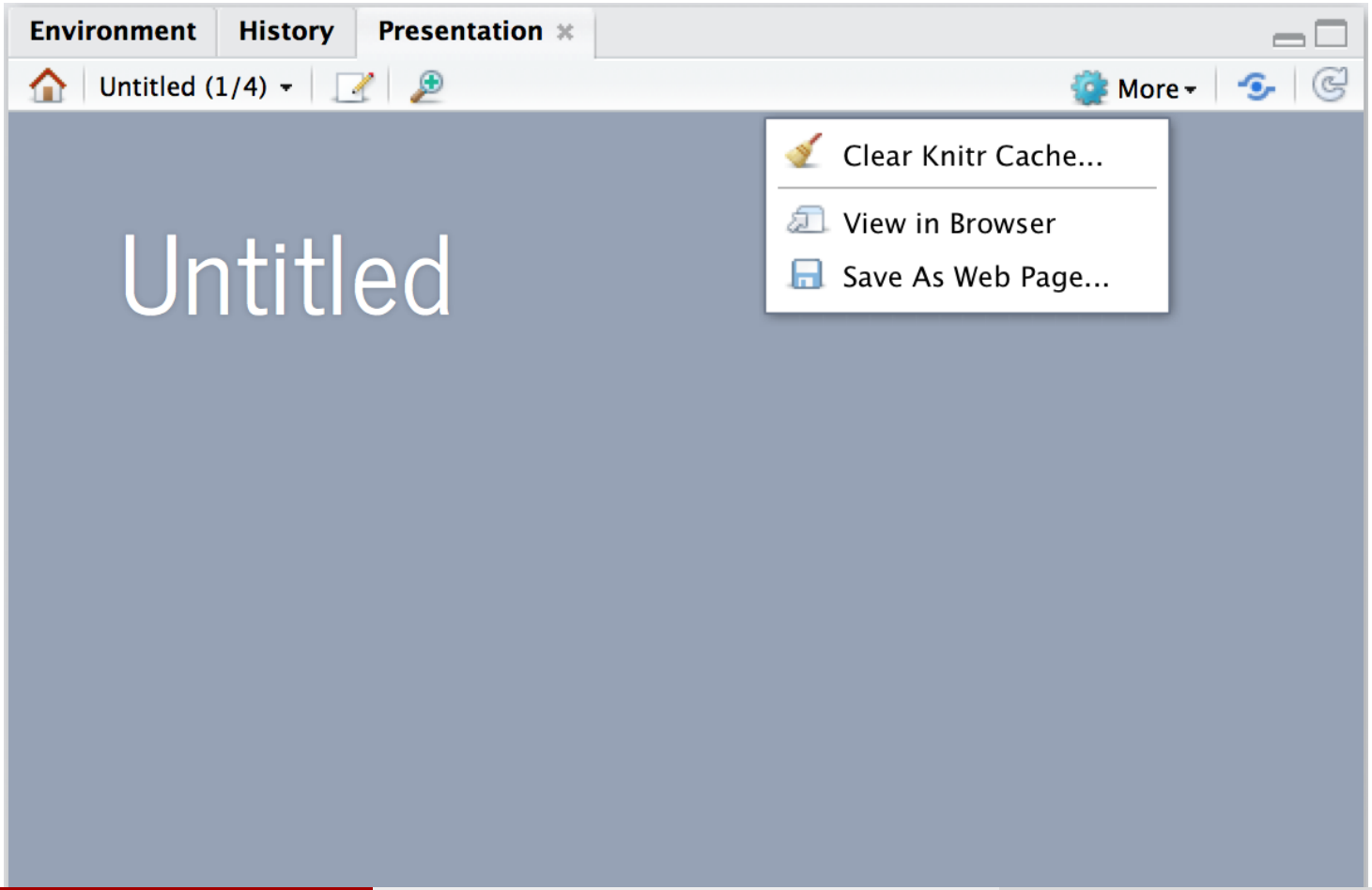
Normale Schriftgröße

```
<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

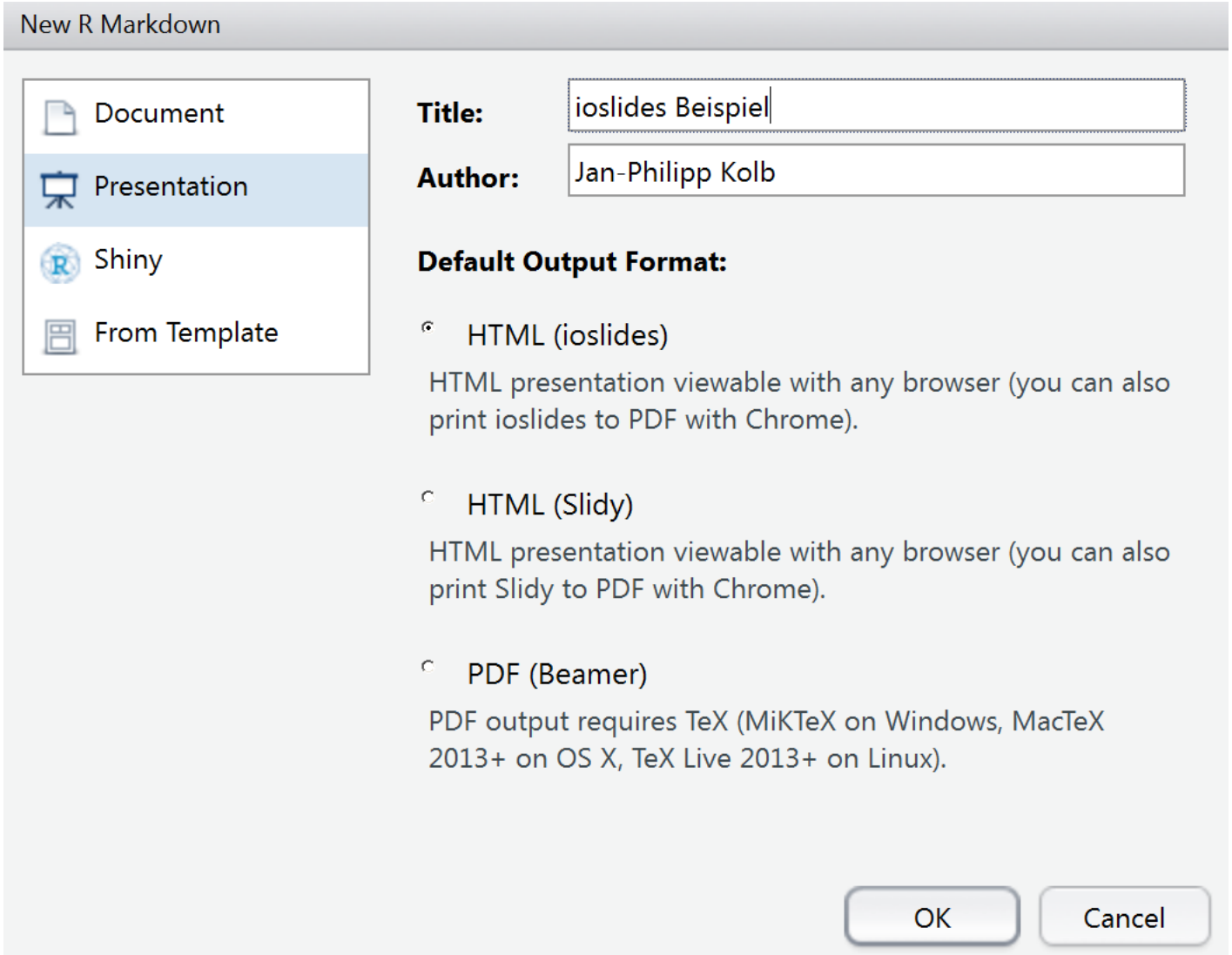


Figure 8

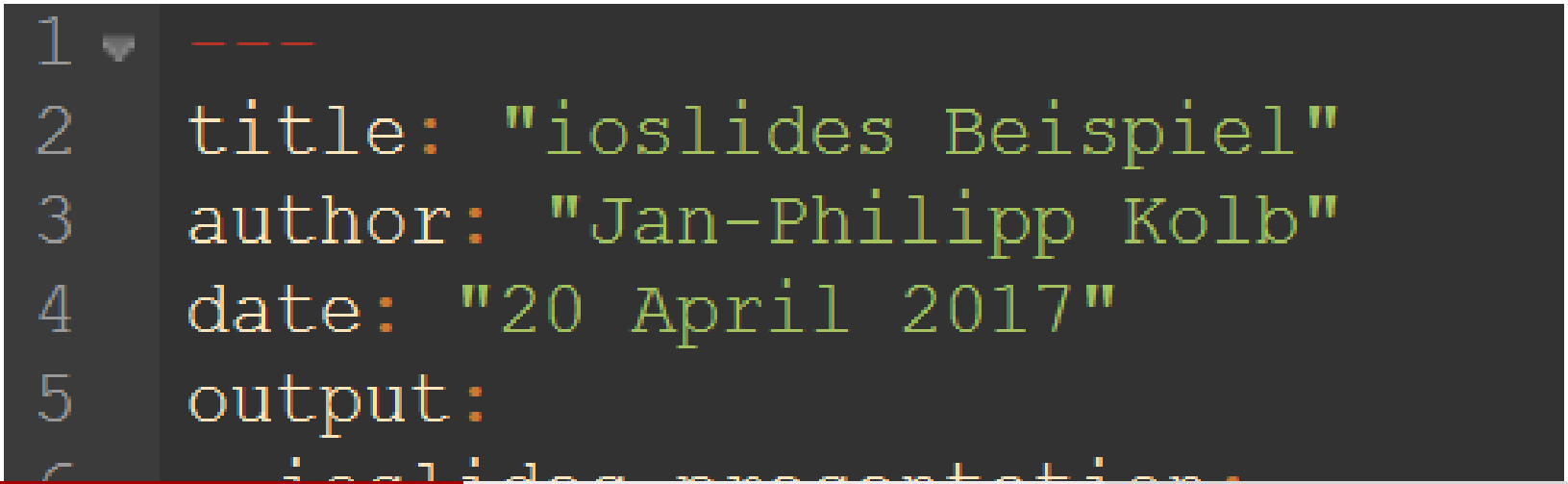
Weitere Dinge tun

- Ein Bild einbinden

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

Ein Logo hinzu

```
---
title: "ioslides Beispiel"
author: "Jan-Philipp Kolb"
date: "20 April 2017"
output:
  ioslides_presentation:
    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

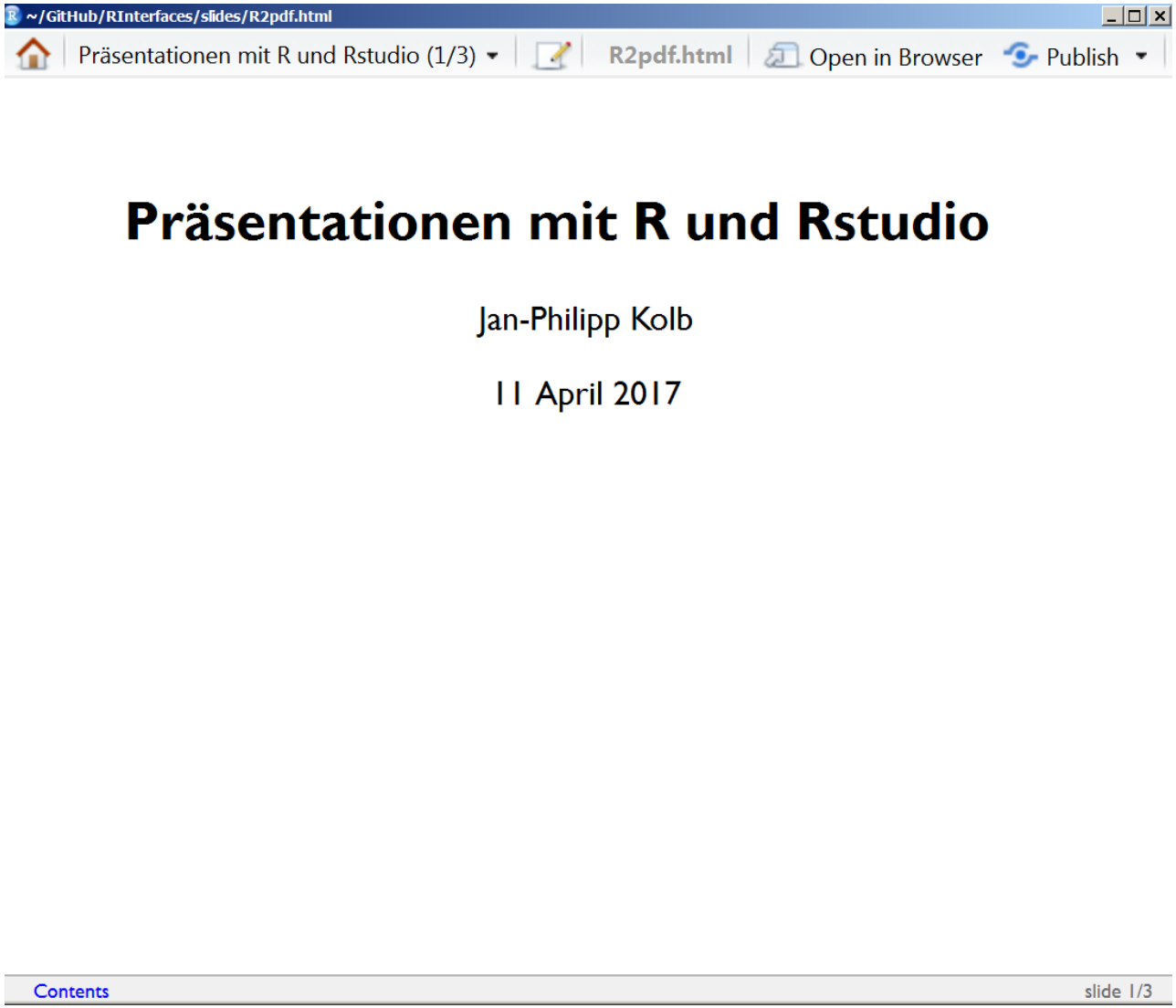


Figure 10

[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

```
body, td {
    font-family: Lucida Console;
    background-color: transparent;
    font-size: 20px;
}
```

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

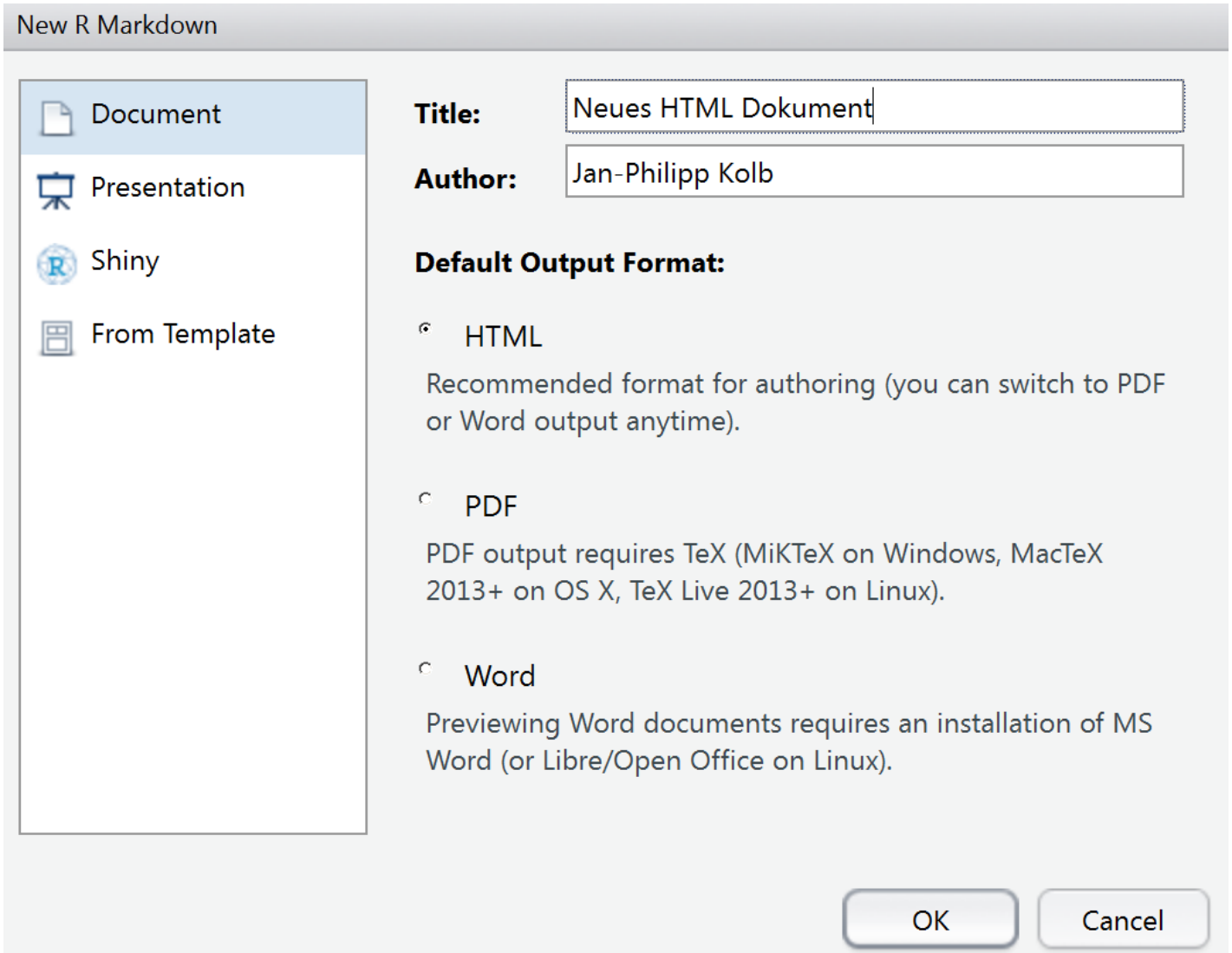


Figure 13

Ein Template verwenden

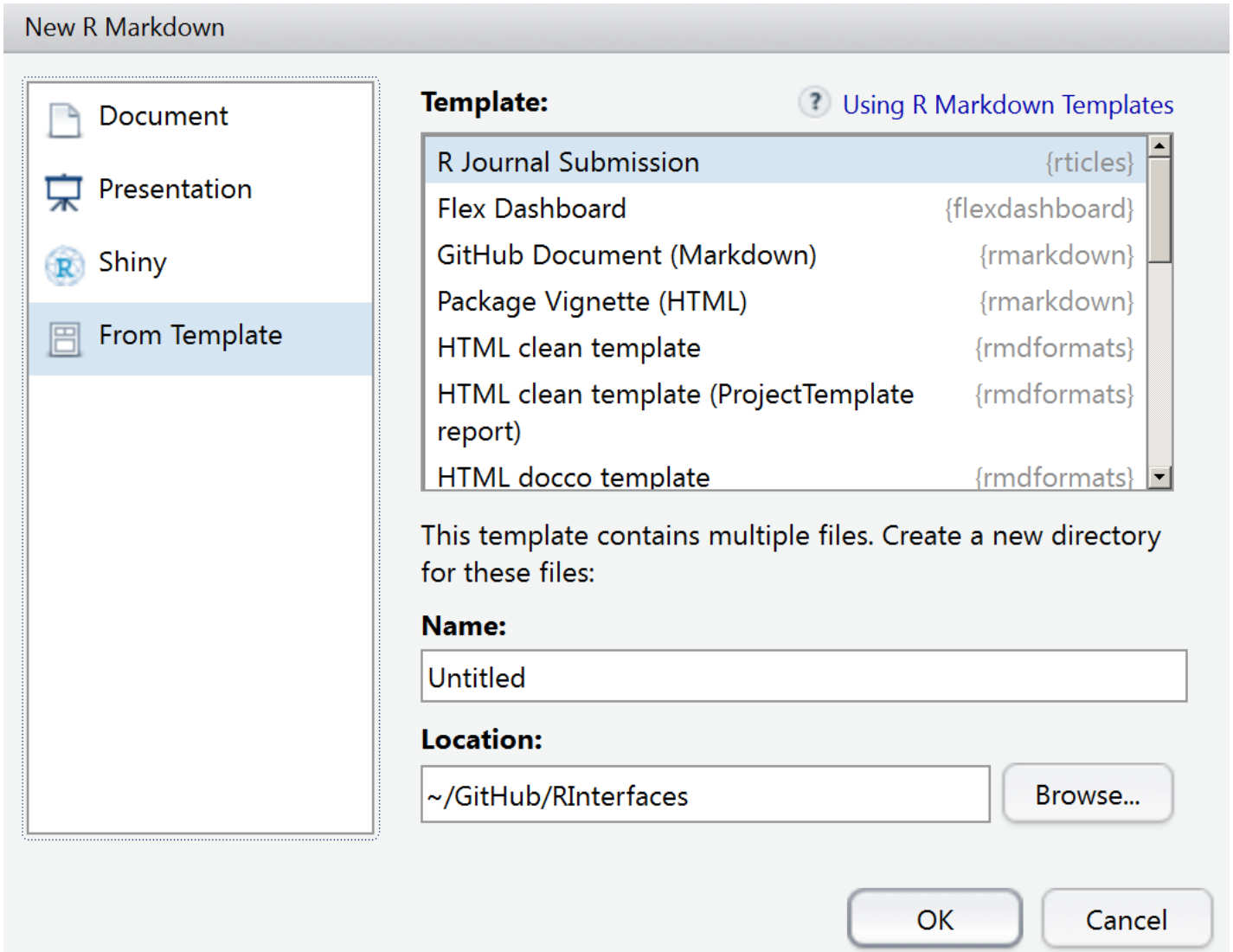
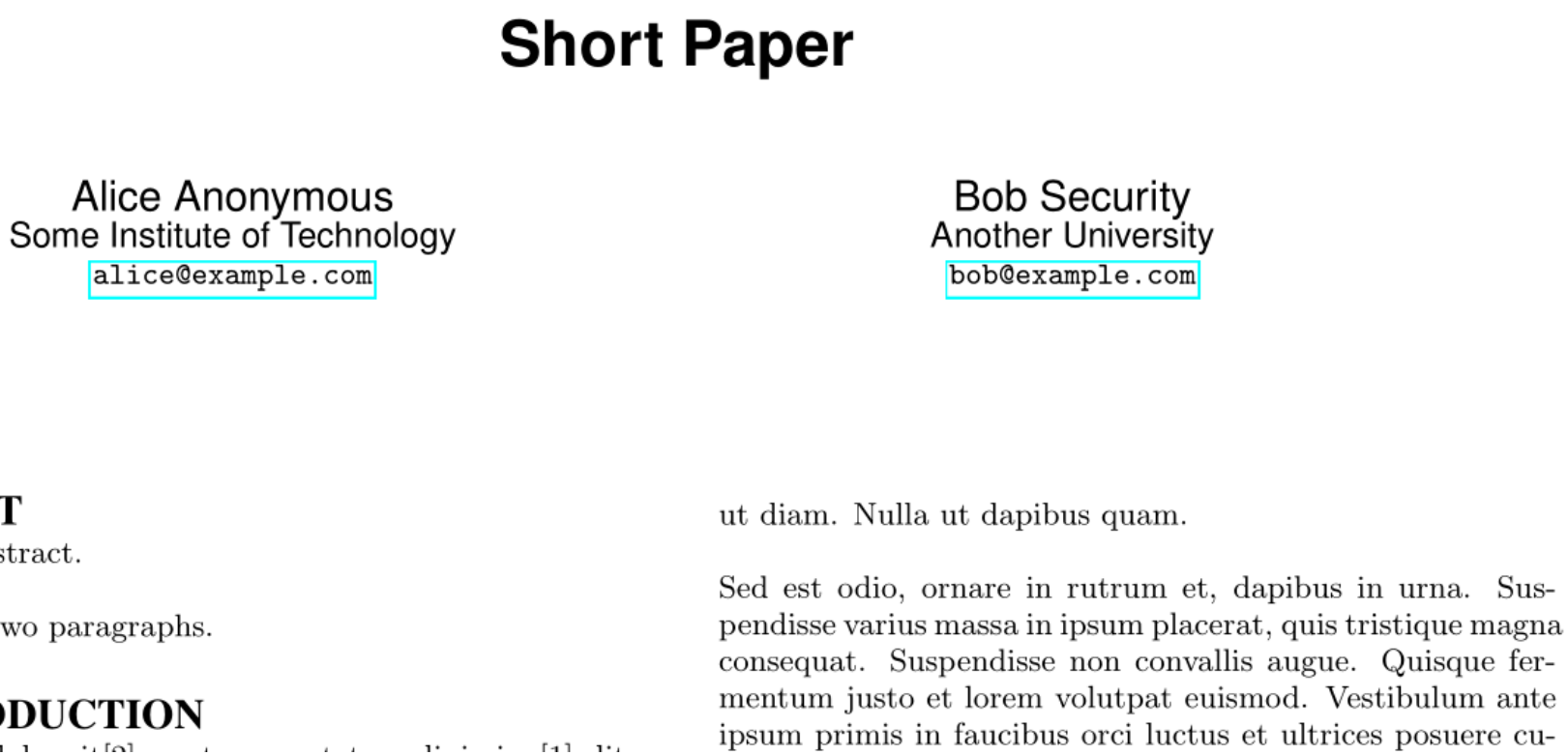


Figure 14

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

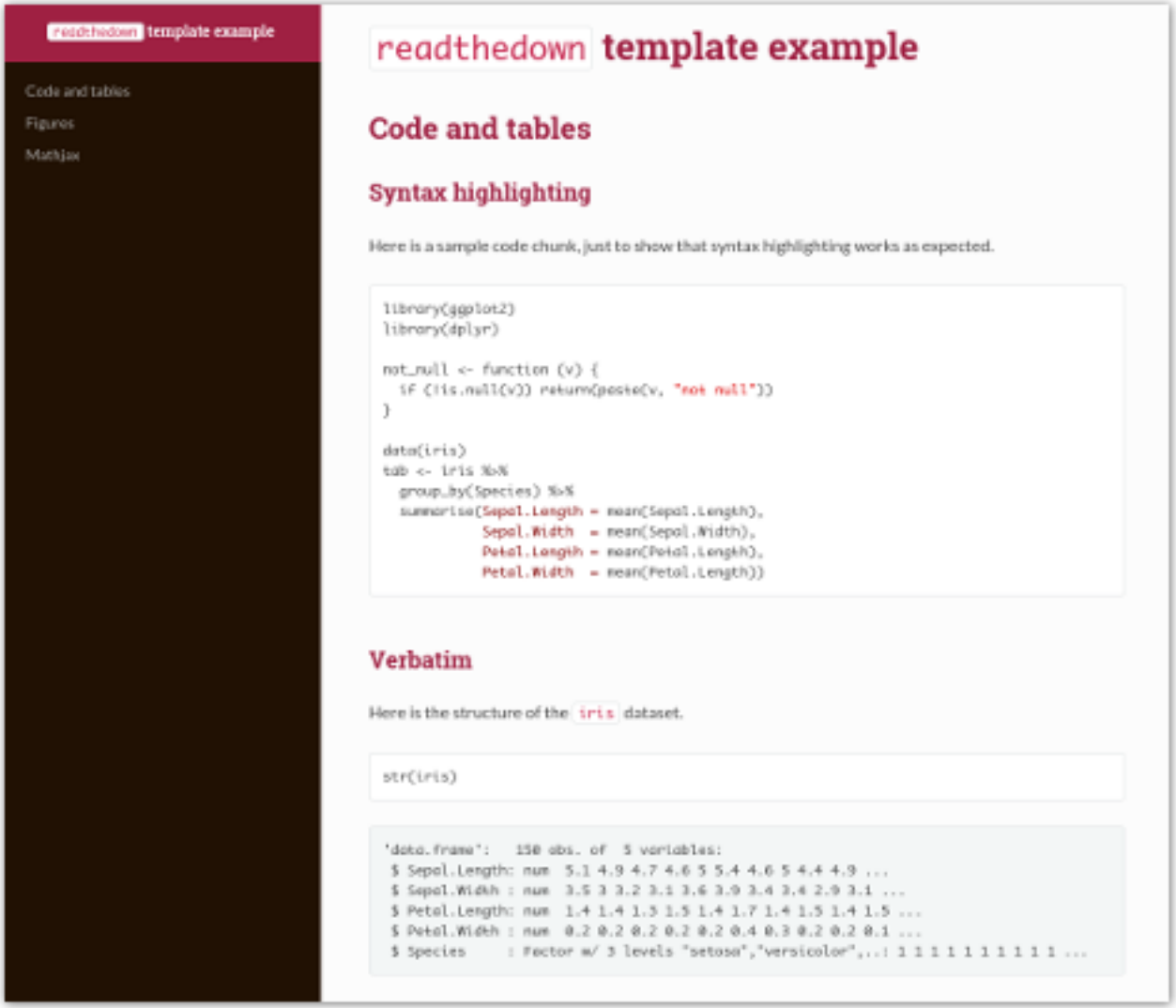


Figure 16

Dashboards

Beispiel R-Pakete

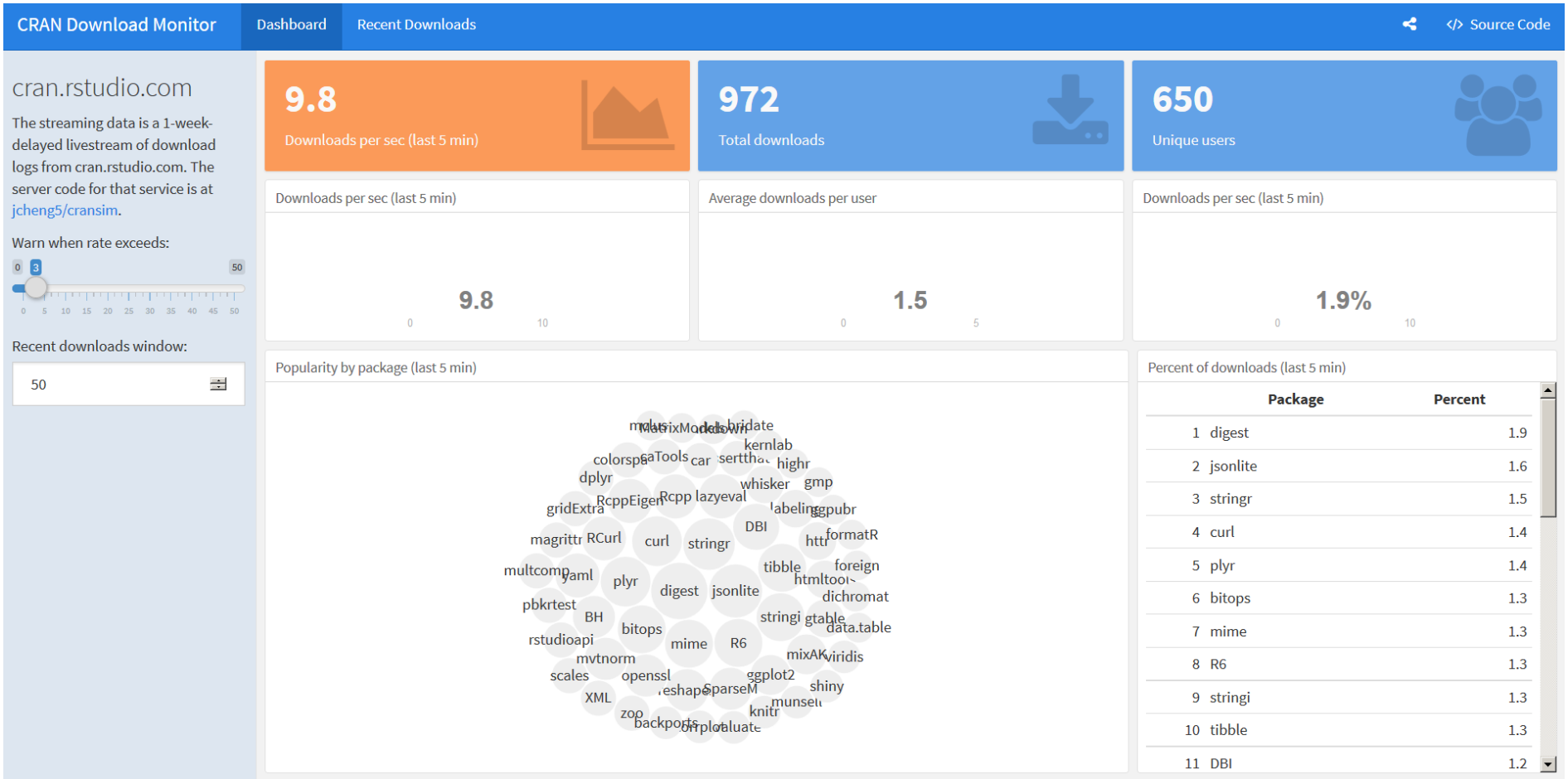


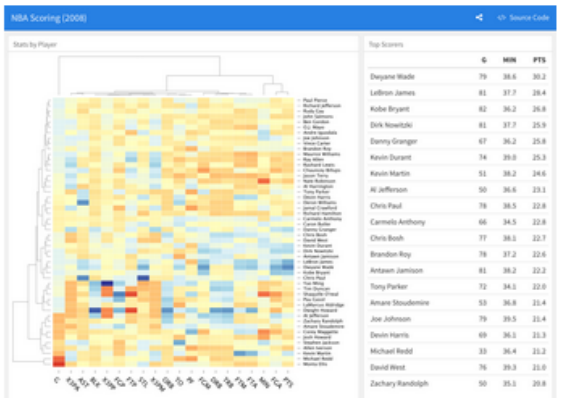
Figure 17

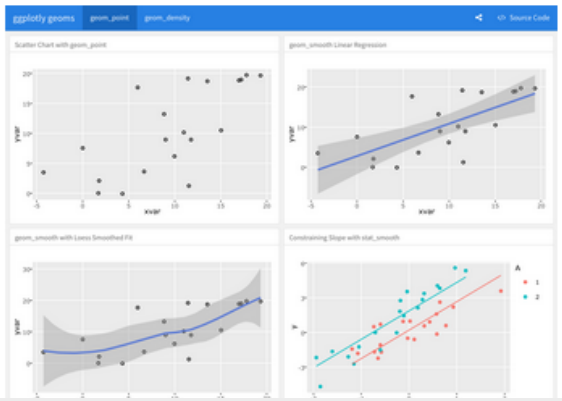
Paket installieren

```
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.





Ein Dashboard erstellen mit Rstudio

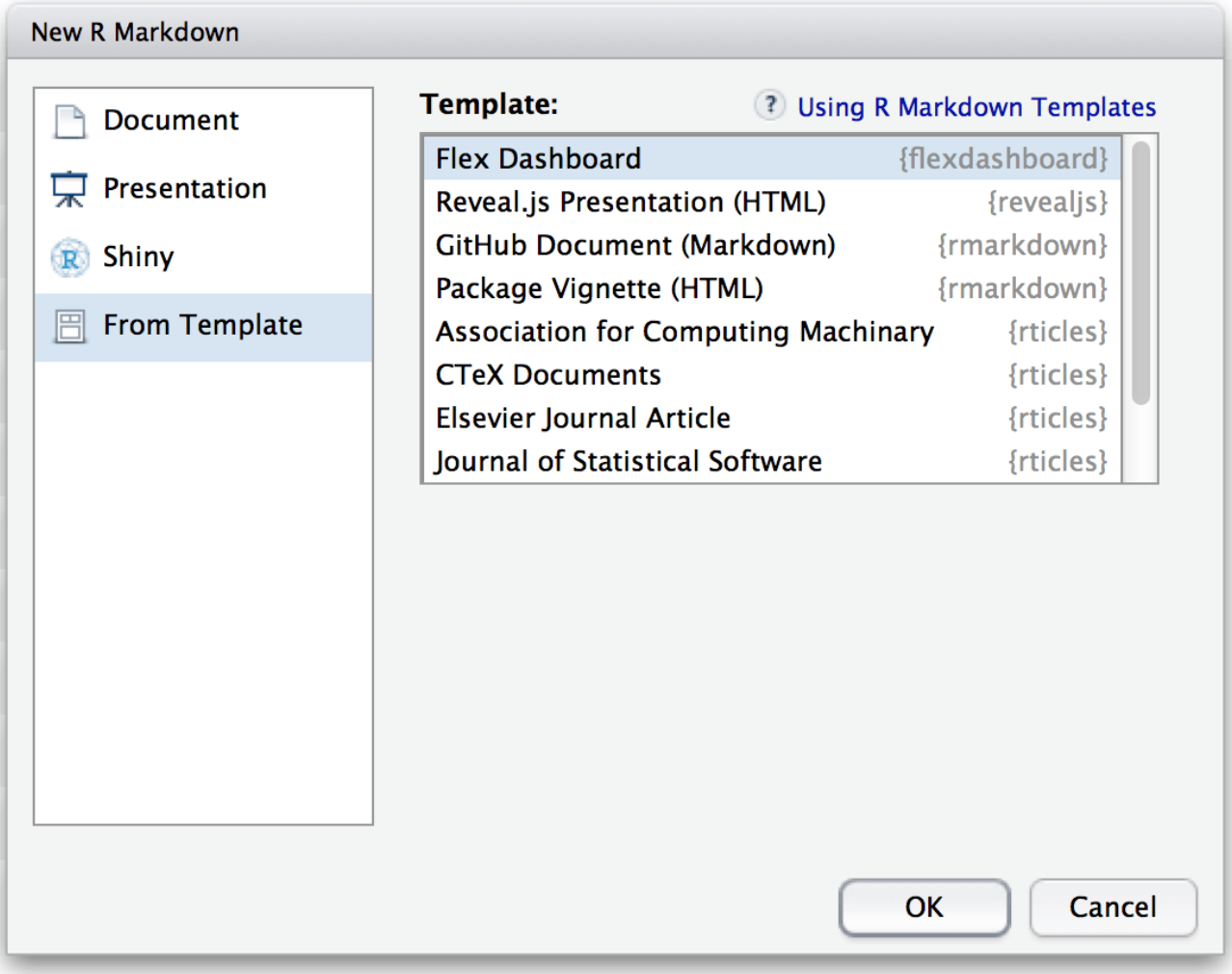


Figure 19

Mein erstes Dashboard

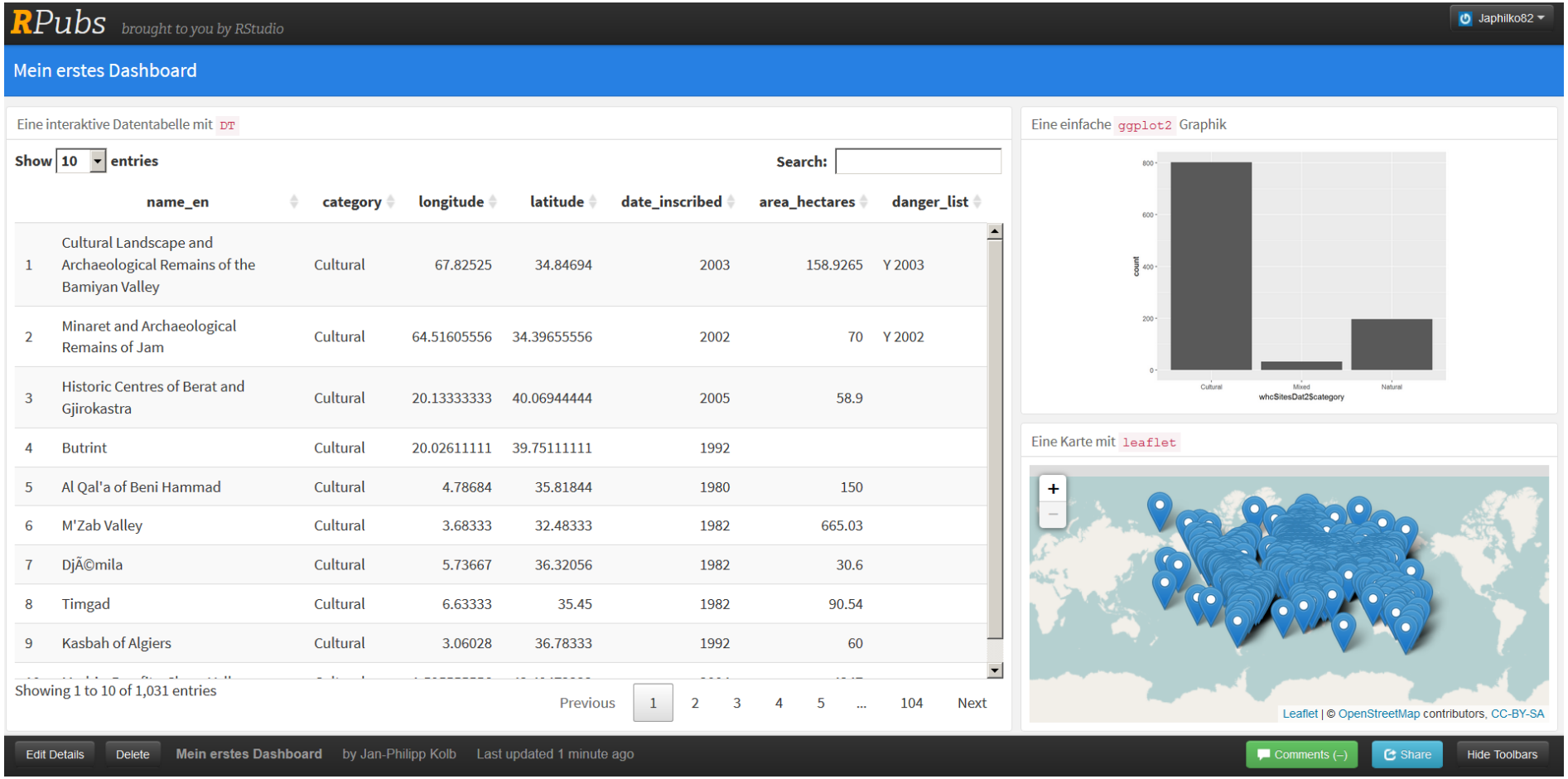
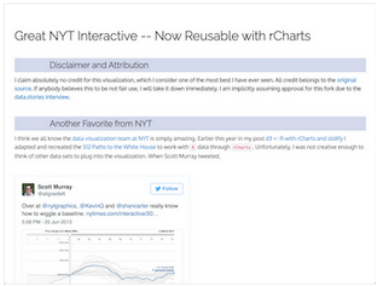


Figure 20

Gallerie

Documents

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



HTML

HTML documents for web publishing.

PDF

PDF documents for printing. Example Code

Microsoft Word

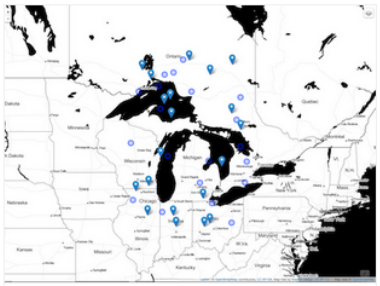
Microsoft Word documents for Office workflows.

Handouts

Tufte styled documents for handouts. Example Code

Interactive Documents

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

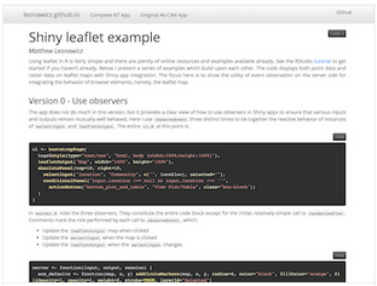


HTML Widgets

Add interactive graphics with htmlwidgets, such as the leaflet map widget.

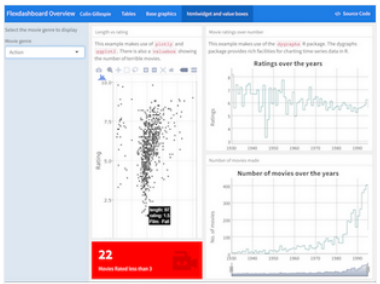
HTML Widgets

Embed htmlwidgets such as dygraphs and datatables directly into your reports.



Shiny

Add interactive analysis with shiny, which lets your user rerun the actual analysis within your report.



Shiny

Shiny components and htmlwidgets will work in any HTML based output, such as a file, slide show or dashboard.

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'}
```

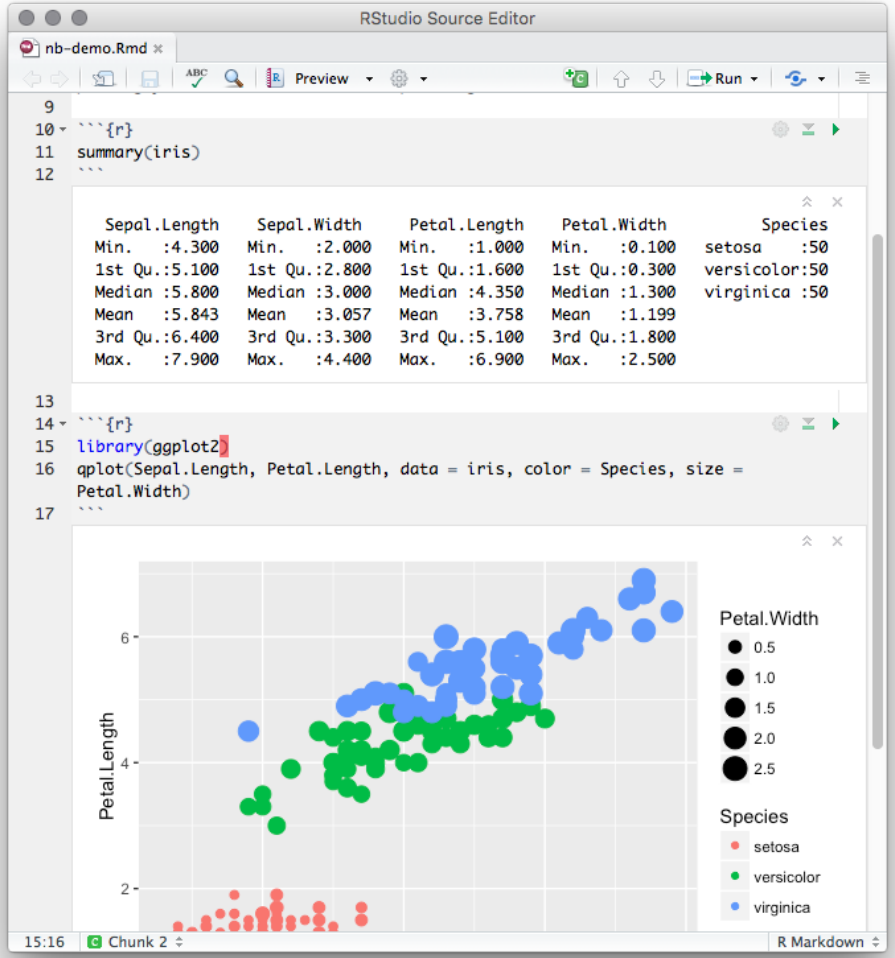
```{r child = 'chapter2.Rmd'}
```
```

Figure 22

Notebooks

Notebooks

- Warum R Notebook nutzen



Rnotebooks

Ein Rnotebook anlegen

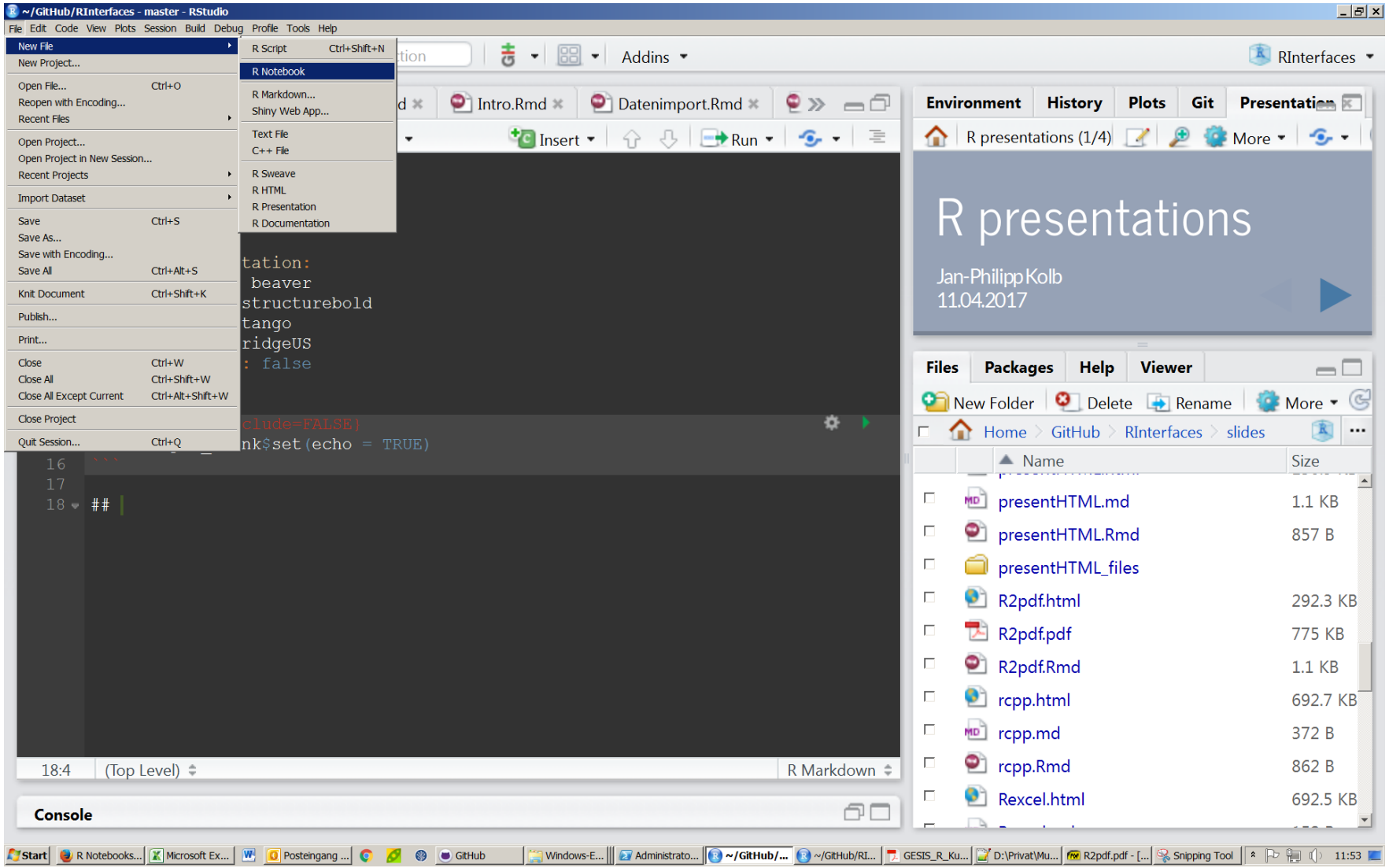


Figure 24

Rnotebook - erste Schritte

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

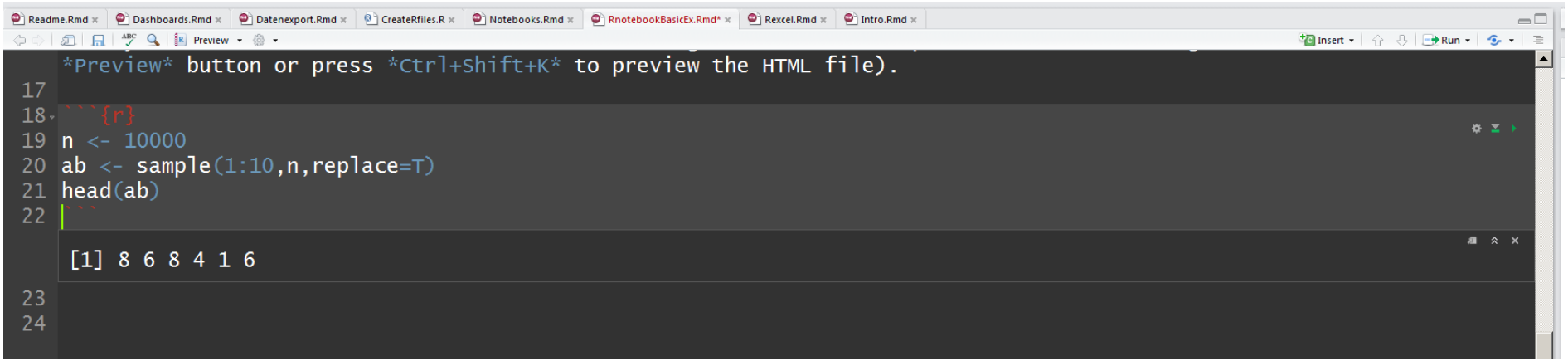


Figure 25

Python Code integrieren

- Ebenso lässt sich Python code implementieren

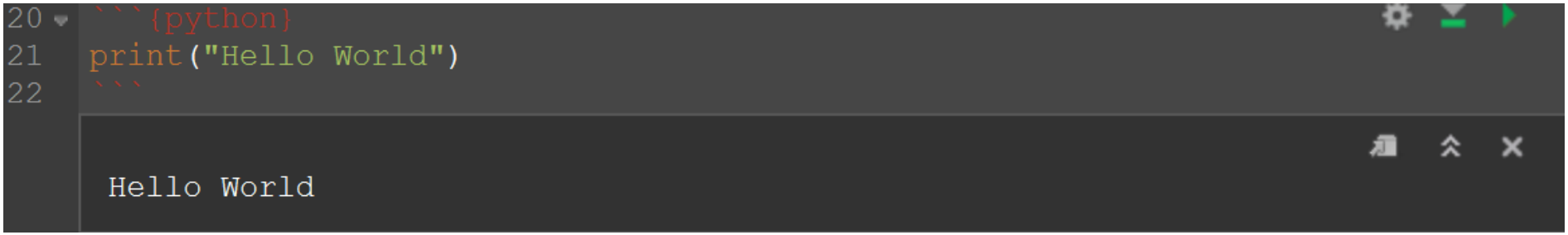


Figure 26

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

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

LaTeX Code integieren

- LaTeX code wird mit zwei Dollarzeichen gekennzeichnet

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

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

Figure 27

Notebook veröffentlichen I

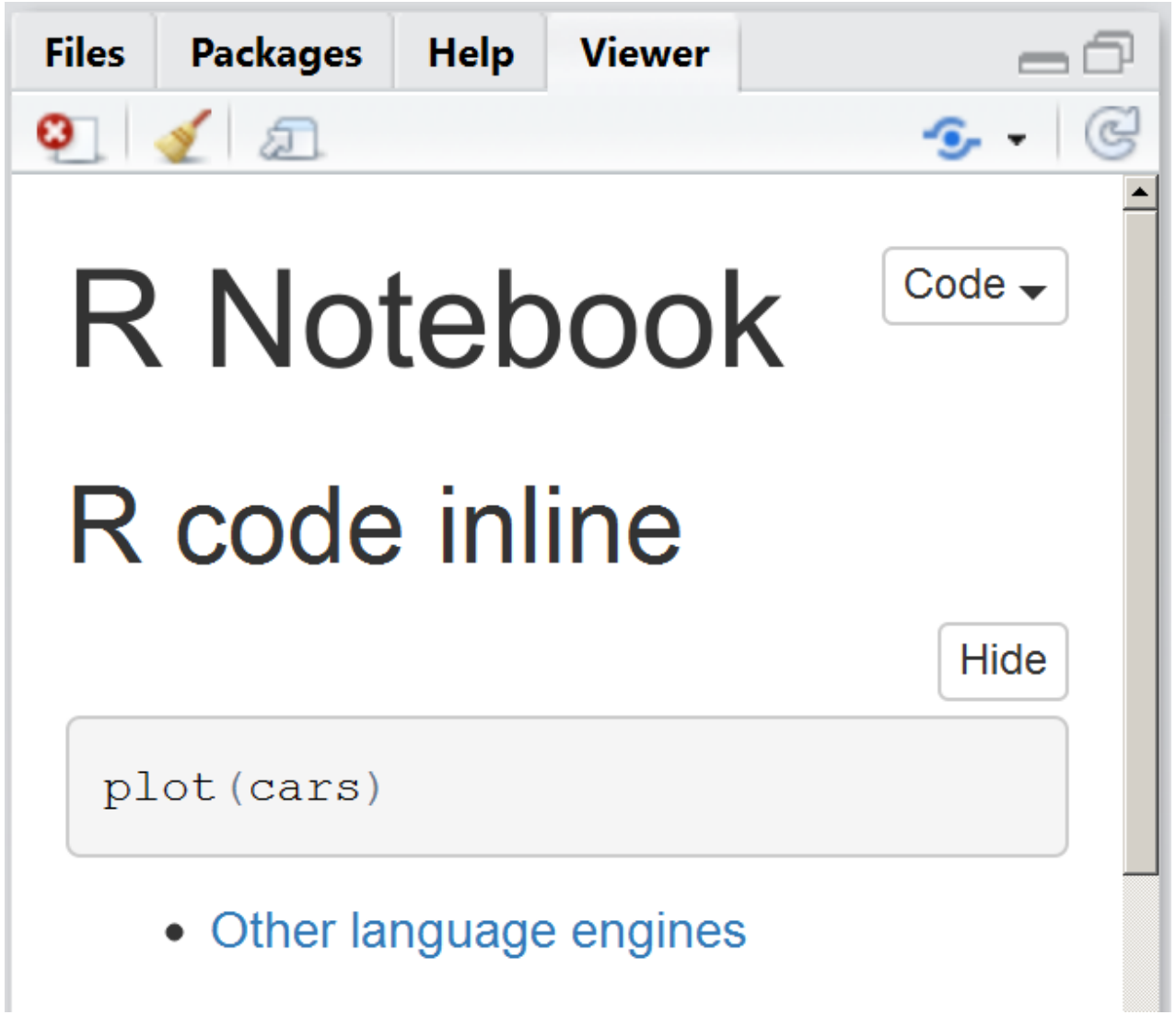


Figure 28

Notebook veröffentlichen II

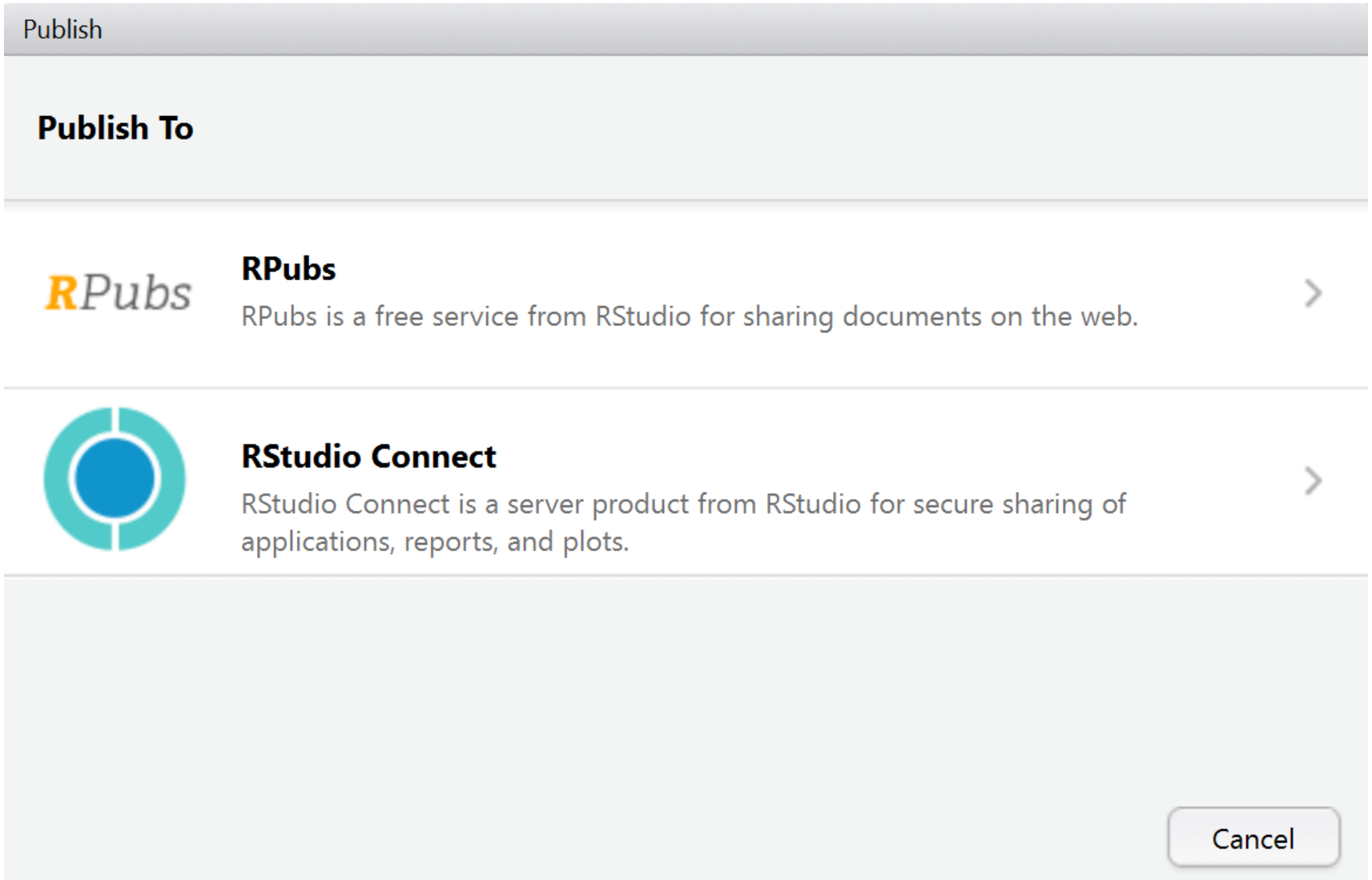


Figure 29

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

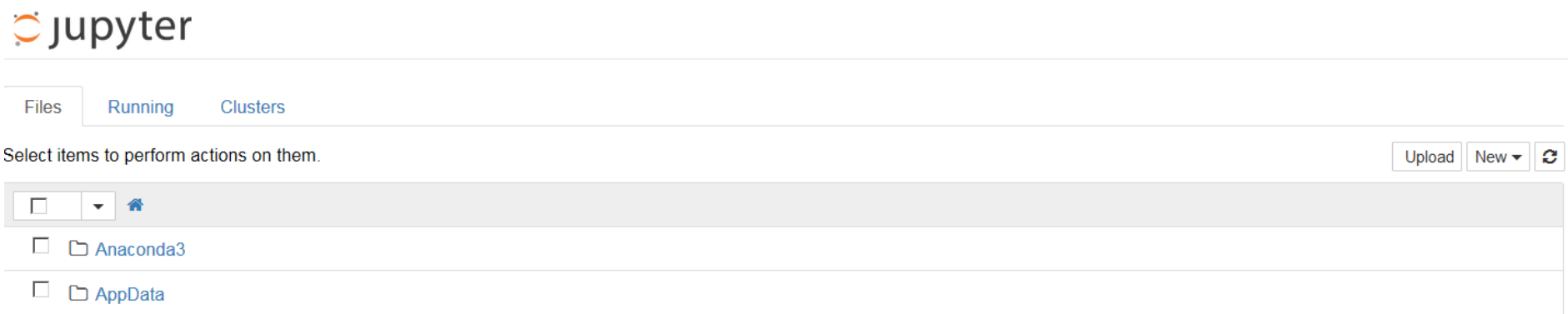


Figure 30

Beispiel Eingabe Code

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

File Edit View Insert Cell Kernel Help R

Code CellToolbar

```
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

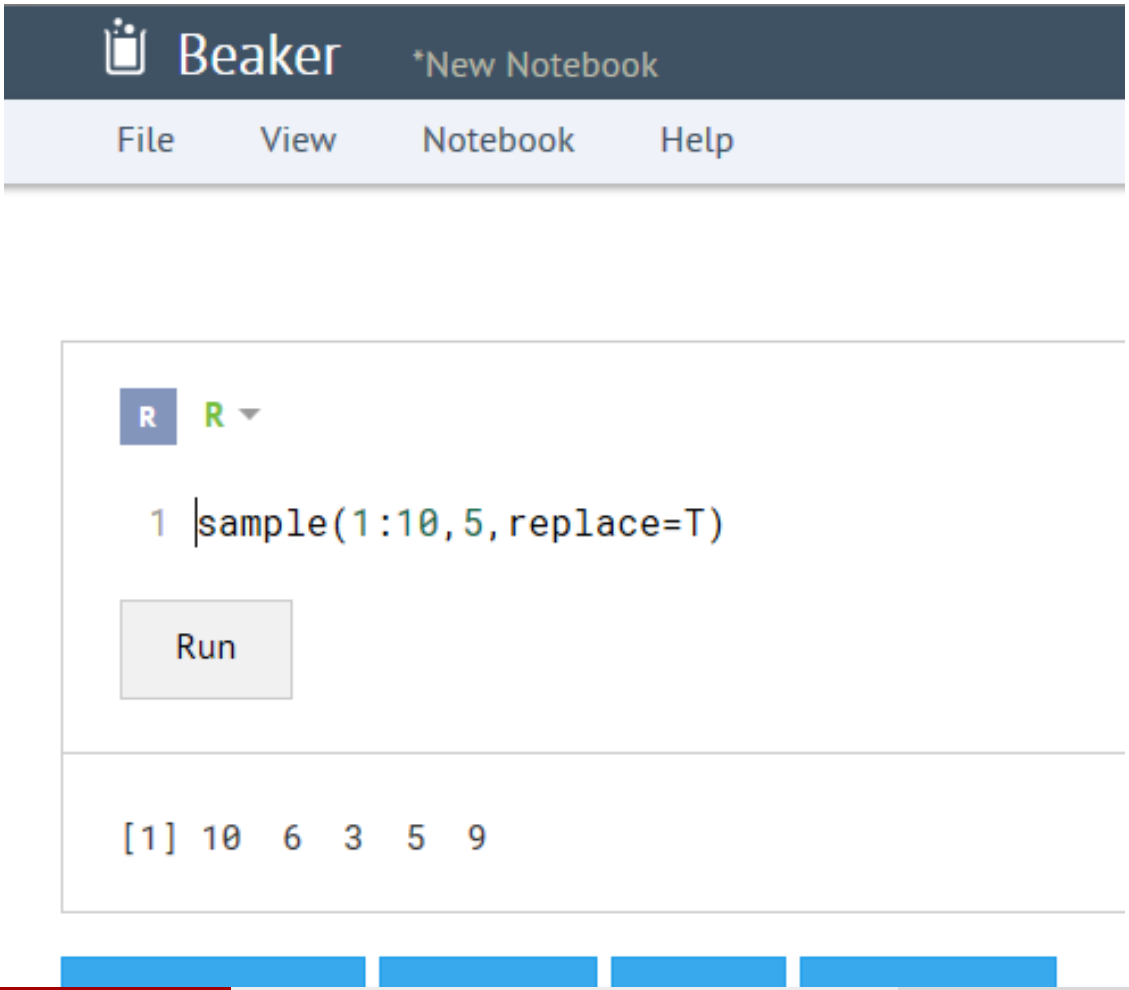
Beaker Notebook

- Auch bei Beaker kann man R-code einbauen

Figure 32

Beaker starten

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



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')
```

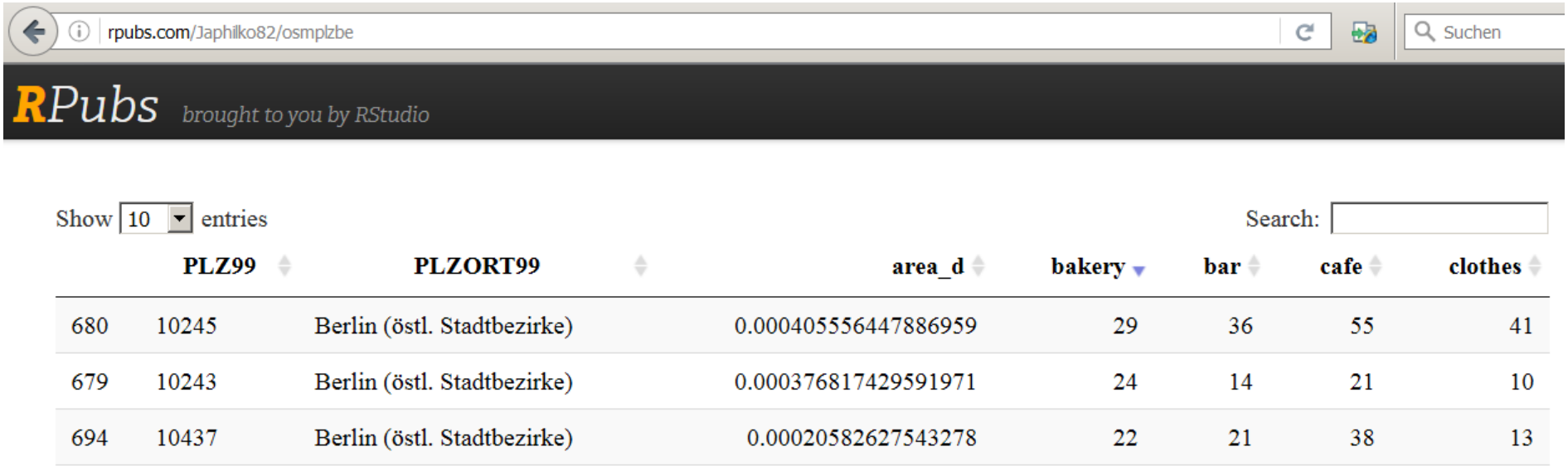
```
library('DT')
```

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

```
datatable(exdat)
```

Beispiel für interaktive Tabelle

Hier ist das Ergebnis - Beispiel für eine interaktive Tabelle



	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 5 entries Search:

	mpg	cyl	disp	hp	drat	model
	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

Figure 25

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	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`

RPubs brought to you by RStudio

Show 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

Share Tweet 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 hinzu

- 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

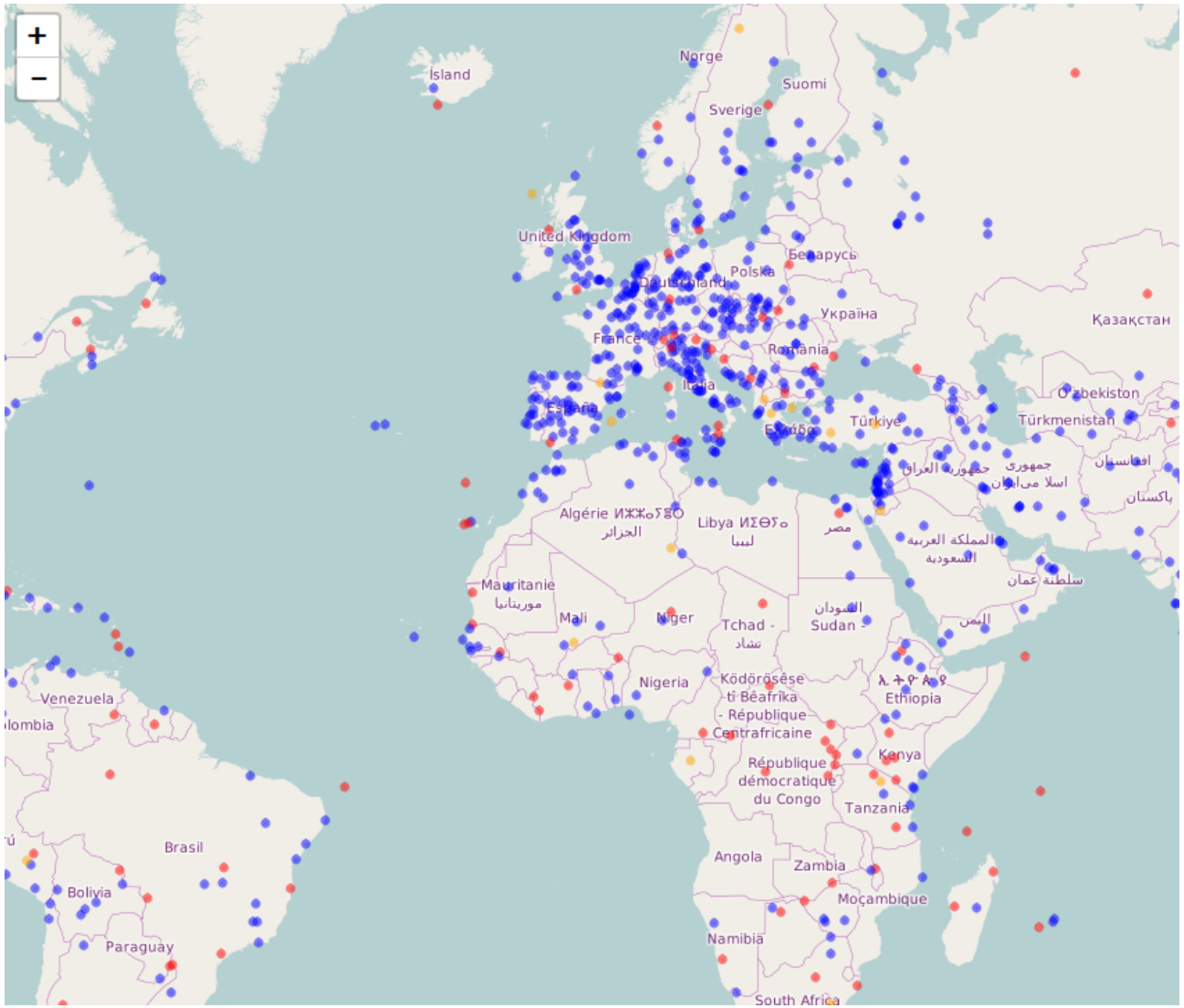


Figure 40: Weltkulturerbe

Die Karte abspeichern

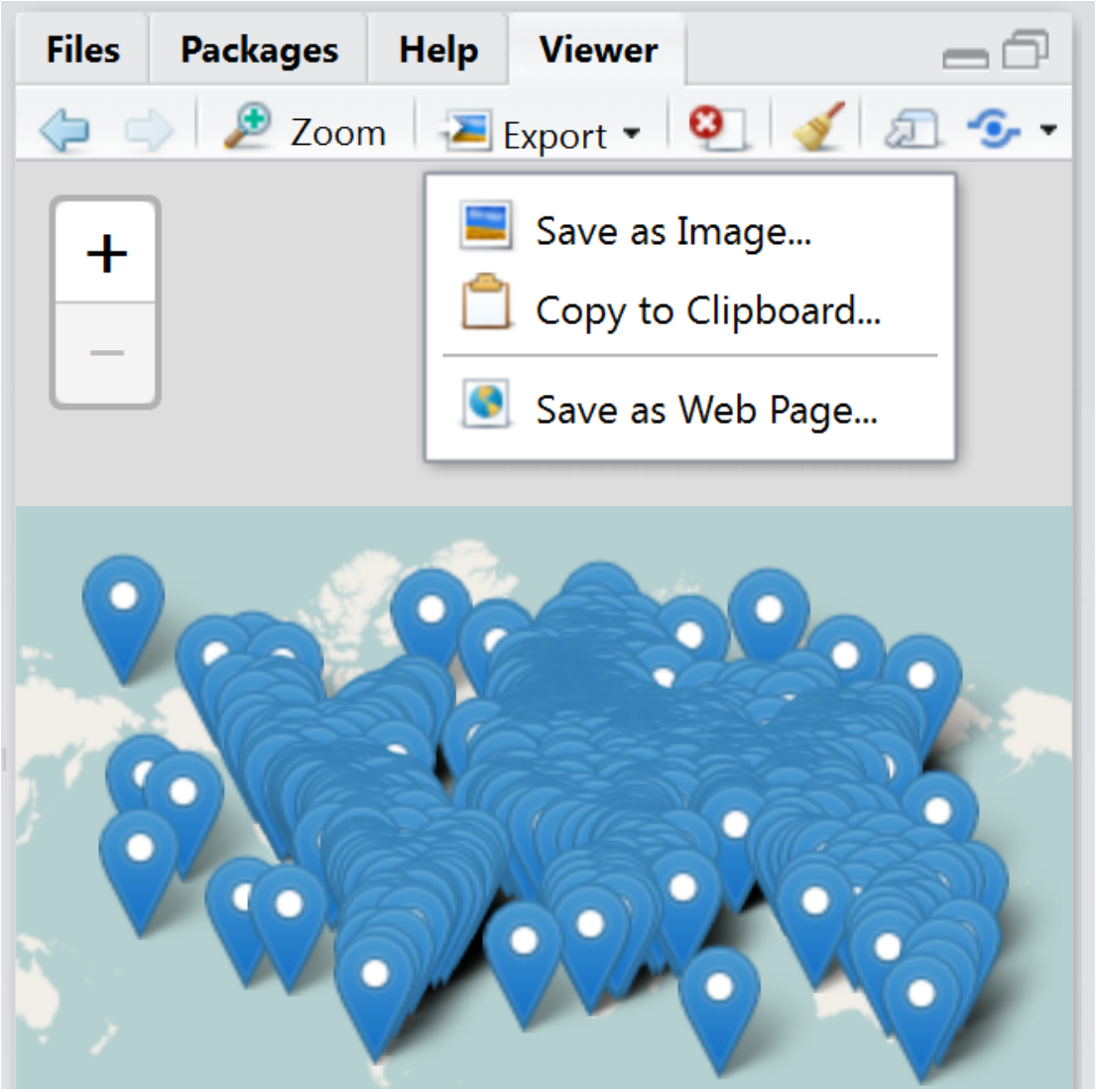


Figure 41

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

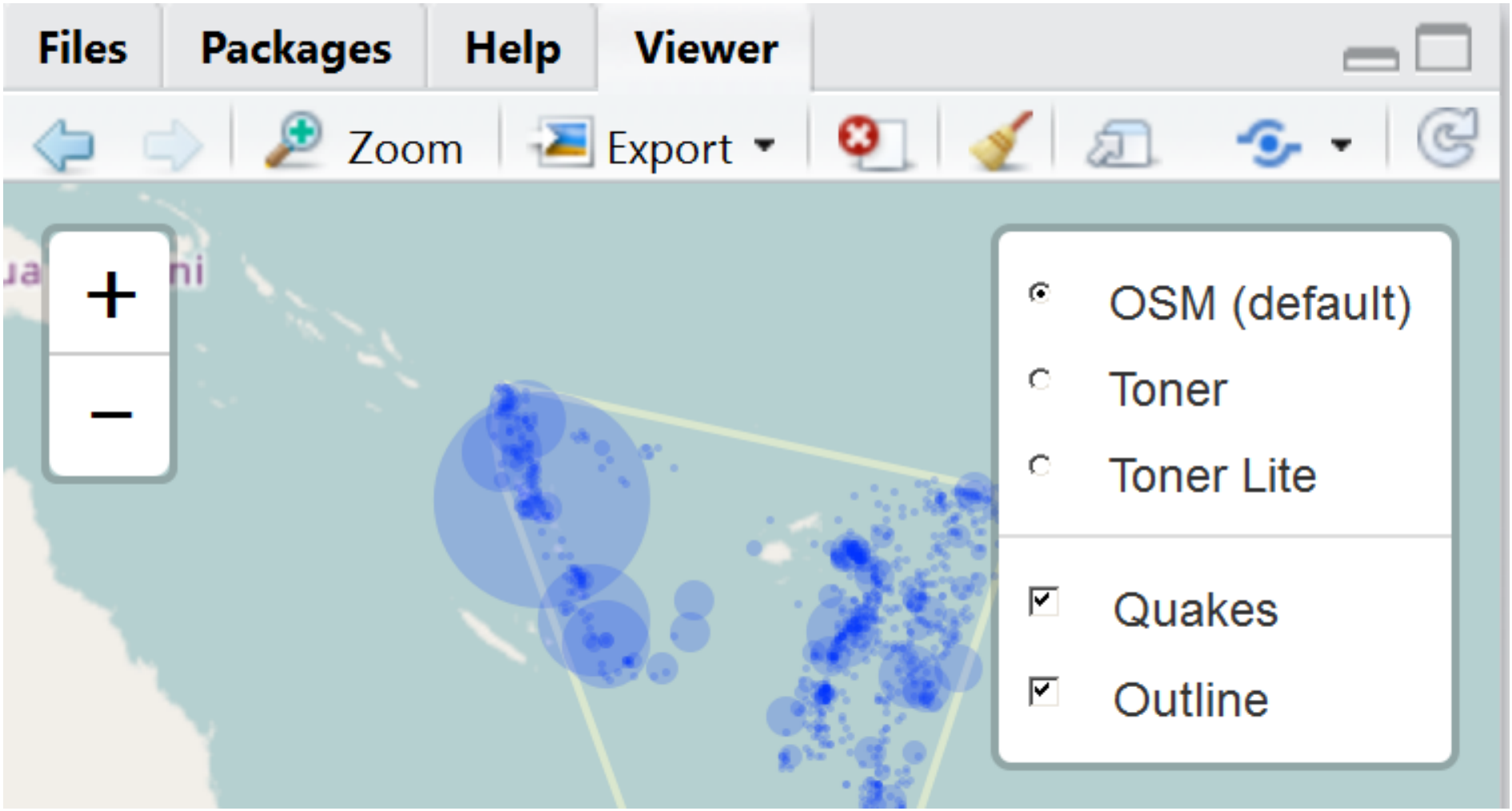


Figure 43

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
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
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.c
    layers = "nexrad-n0r-900913",
    options = WMSTileOptions(format = "image/png", transparent
    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
  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/`
- `https://rstudio.github.io/leaflet/`