

R Schnittstellen

Jan-Philipp Kolb

8 Mai 2017

Einführung und Motivation

Pluspunkte von R

- Als Weg kreativ zu sein ...
- Graphiken, Graphiken, Graphiken
- In Kombination mit anderen Programmen nutzbar
- Zur Verbindung von Datenstrukturen
- Zum Automatisieren
- Um die Intelligenz anderer Leute zu nutzen ;-)
- ...

Gründe

- R ist frei verfügbar. Es kann umsonst runtergeladen werden.
- R ist eine Skriptsprache
- Gute Möglichkeiten für die Visualisierung
- R wird immer populärer
- Popularität von R ist in vielen Bereichen sehr hoch.

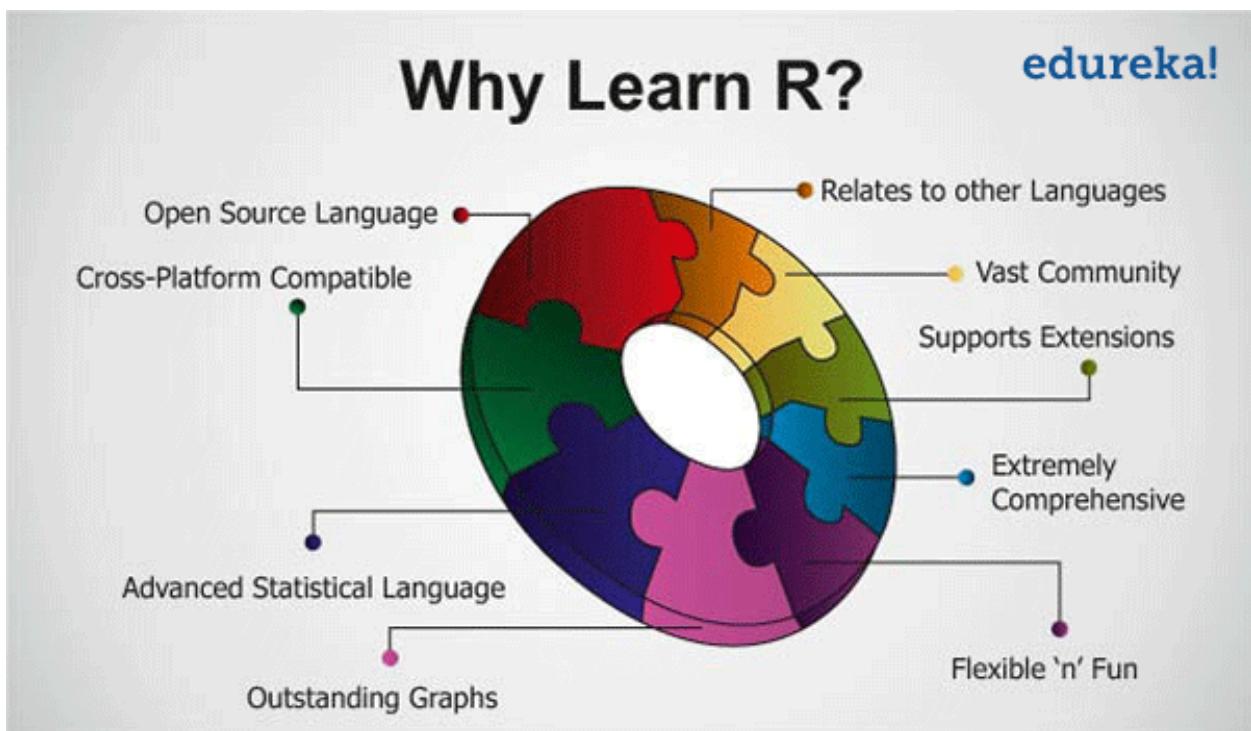


Figure 1:

Warum R?

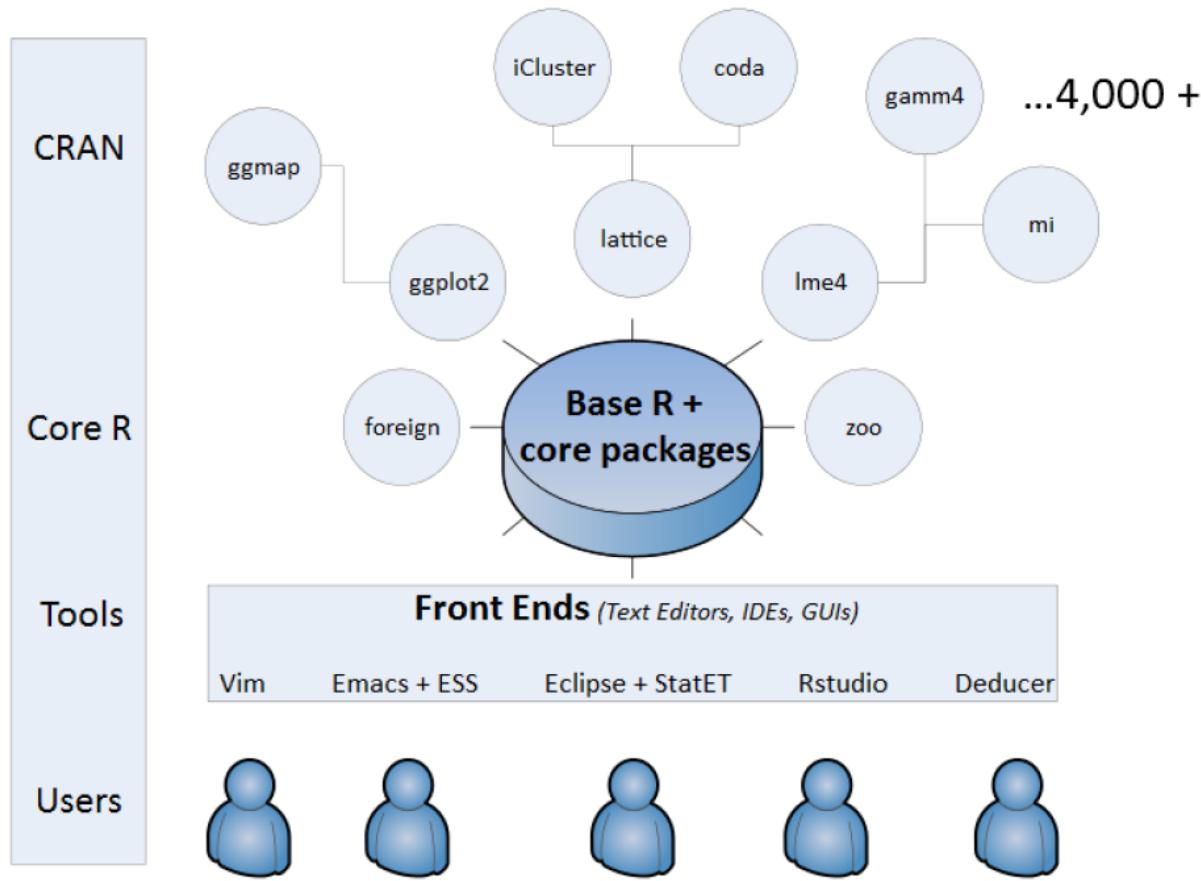


Figure 2: Modularer Aufbau

Die am meisten heruntergeladenen Pakete

CRAN Task Views

Motivation - Nachteile von R

1. Daten werden oft anderswo erfasst/eingegeben (oft Excel, SPSS etc.)
2. Nicht jeder ist bereit mit R zu arbeiten
3. Nicht auf jedem Rechner ist R installiert
4. R ist manchmal zu langsam
5. Schwierigkeiten bei der Arbeit mit großen Datenmengen

Was folgt daraus

1. Schnittstelle zu SPSS/Stata/Excel zum Import von Daten
2. Schnittstelle zu Word/LaTeX
3. Möglichkeit HTML Präsentationen zu erzeugen

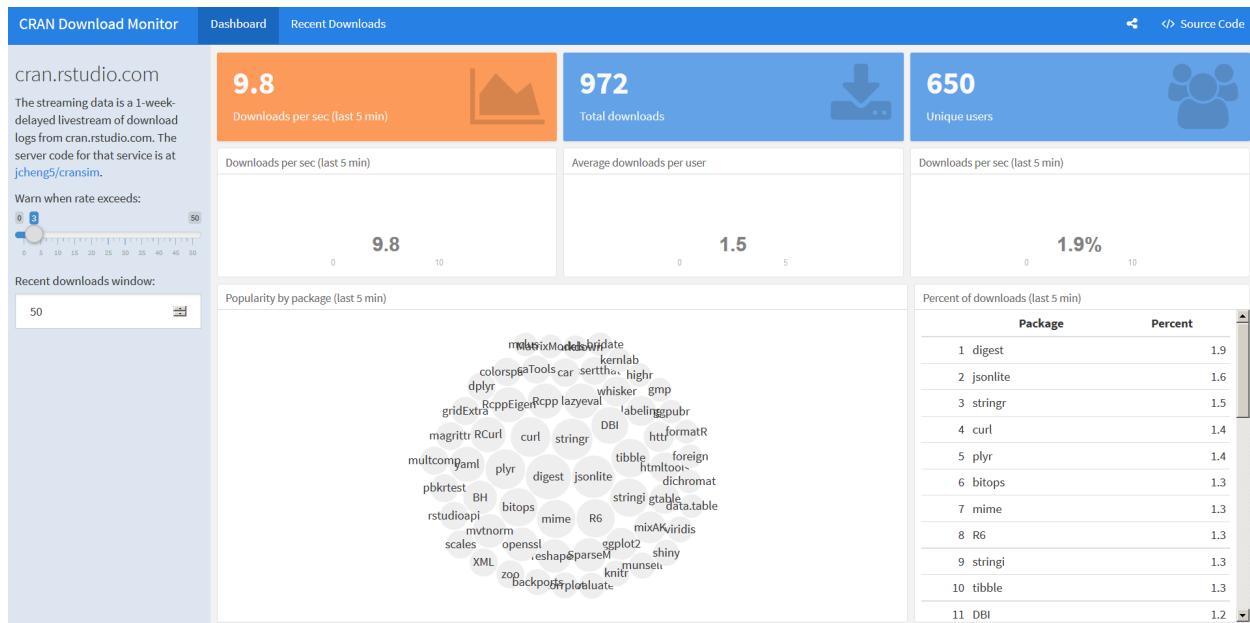


Figure 3:

CRAN Task Views

Bayesian	Bayesian Inference
ChemPhys	Chemometrics and Computational Physics
ClinicalTrials	Clinical Trial Design, Monitoring, and Analysis
Cluster	Cluster Analysis & Finite Mixture Models
DifferentialEquations	Differential Equations
Distributions	Probability Distributions
Econometrics	Econometrics
Environmetrics	Analysis of Ecological and Environmental Data
ExperimentalDesign	Design of Experiments (DoE) & Analysis of Experimental Data
ExtremeValue	Extreme Value Analysis
Finance	Empirical Finance

Figure 4:

4. Nutzung von C++
5. Nutzung von Datenbanken

Die Nutzung von Schnittstellen beim Import/Export

- Interaktion mit Excel, SPSS, Stata, ...



Figure 5: Import

Reproducible Research

Was wird bei Wikipedia unter Reproducibility verstanden?

Darstellung von Ergebnissen

- Mit der Schnittstelle zu Javascript lassen sich interaktive Graphiken erzeugen
- Diese kann man auf Websites, in Präsentationen oder in Dashboards verwenden

Warum die Schnittstelle zu C++?

- Wenn Schnelligkeit wichtig ist, bietet sich C++ an.
- Dies kann bspw. der Fall sein, wenn sich Schleifen nicht vermeiden lassen.
- Man wird bei der Programmierung durch RStudio unterstützt.
- Es gibt eine Rcpp Galerie, wo man sich Anregungen holen kann.
- Allerdings sollte man zunächst versuchen den Rcode so schnell wie möglich zu gestalten.

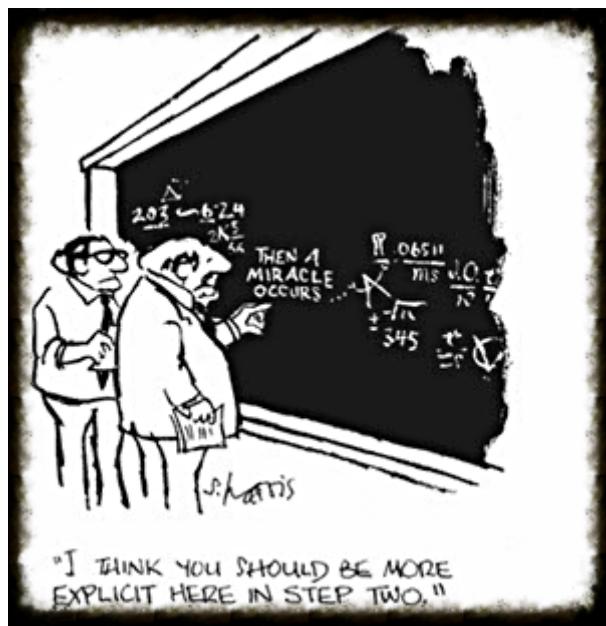


Figure 6:



Figure 7:

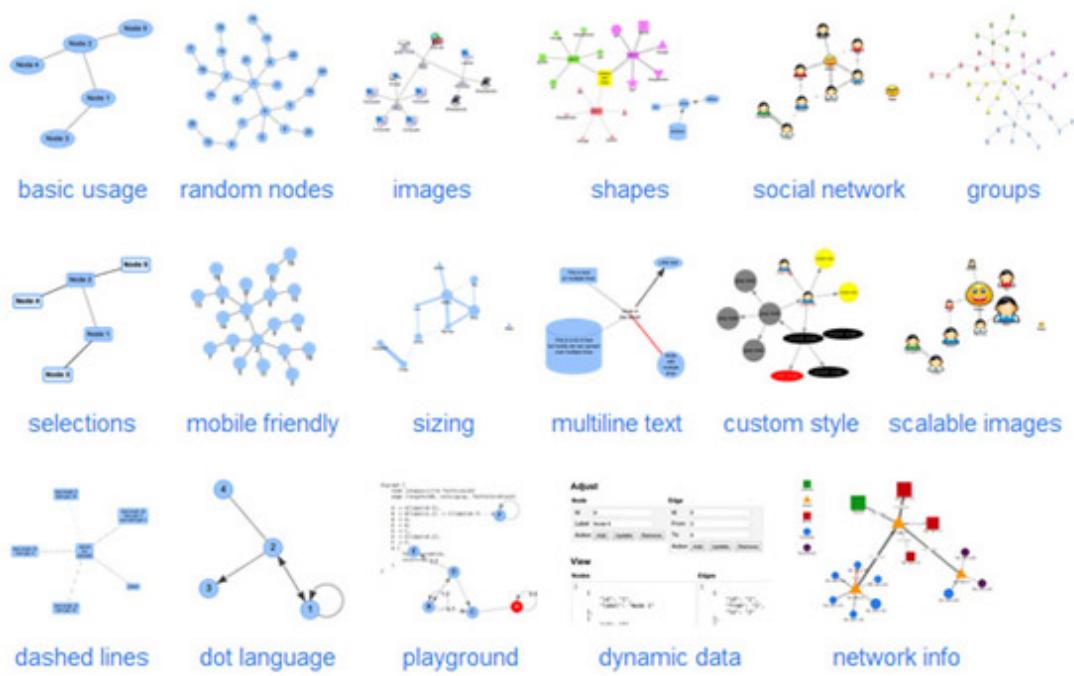


Figure 8:

Die Nutzung von Datenbanken

- Wenn große Datenmengen anfallen, kann die zusätzliche Nutzung von Datenbanken sinnvoll sein.
- In R bestehen Schnittstellen sowohl zu relationalen als auch nicht relationalen Datenbanken.
- Datenbanken sollten allerdings erst genutzt werden, wenn alle Möglichkeiten in R ausgeschöpft sind.

Nutzung der Unterlagen auf GitHub

- Die folgende Seite ist die Startseite für den Kurs:

<https://japhilko.github.io/Interfaces4R/>



Figure 9:

Wo sind die Sourcecodes?

Wie wird das Github Verzeichnis genutzt?

- Auf der folgenden Seite sind alle Sourcecodes enthalten:

<https://github.com/Japhilko/RInterfaces>

- Es lohnt sich immer wieder zu dieser Seite zurückzukehren, weil auch hier alle relevanten Dokumente verlinkt sind.
- Grundsätzlich kann man der Veranstaltung am Besten mit den kompletten File oder der kompletten Browserversion eines Kapitels (sind unter den Kapitelüberschriften verlinkt) folgen. Wenn Teile heruntergeladen werden sollen, bietet es sich an, das entsprechende pdf herunterzuladen.
- Falls Links ins Leere führen - bitte Bescheid sagen.

Informationen ausdrucken

- Zum Ausdrucken eignen sich die pdf-Dateien besser.
- Diese können mit dem Raw Button heruntergeladen werden.

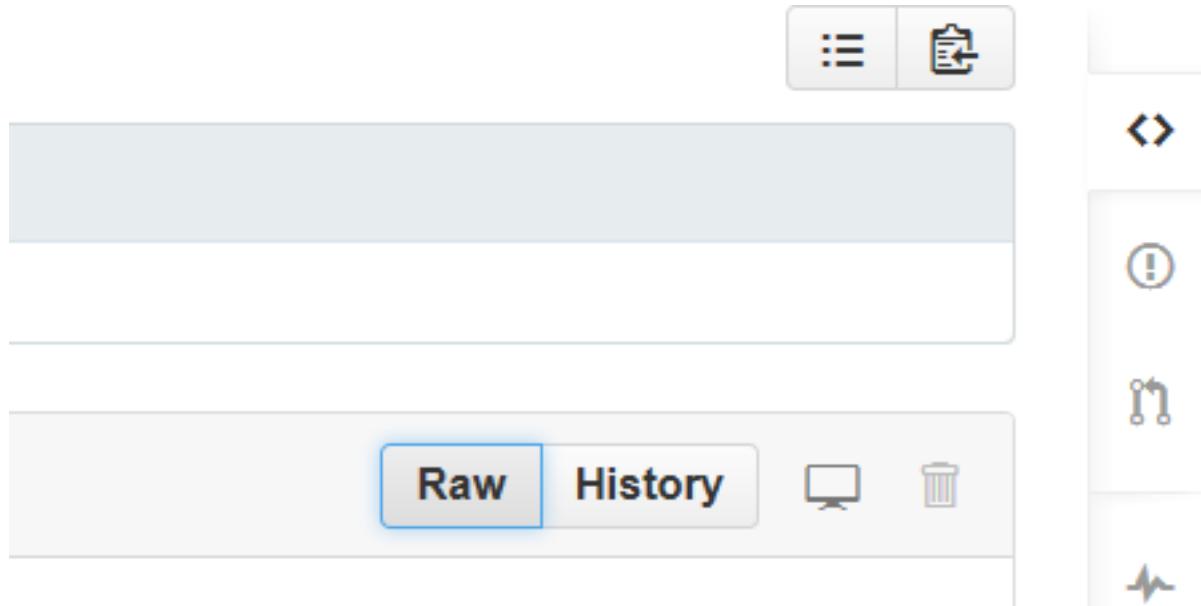


Figure 10: Raw Button zum Download

Weitere Dateien herunterladen

- Begleitend zu den Folien wird meistens auch ein R-File angeboten.
- Hier können Sie entweder das gesamte R-File herunterladen und in R ausführen oder einzelne Befehle per Copy/Paste übernehmen.
- Vereinzelt sind auch Datensätze vorhanden.
- .csv Dateien können direkt von R eingelesen werden (wie das geht werde ich noch zeigen).
- Wenn die .csv Dateien heruntergeladen werden sollen auch den Raw Button verwenden.
- Alle anderen Dateien (bspw. .RData) auch mittels Raw Button herunterladen.

Organisatorisches

- Zusätzlich gibt es in jedem Kapitel eine oder mehrere Aufgabe(n), da man nur durch eigenes Trainieren auf der Lernkurve vorankommt.
- Die Quellen für die Punkte auf den Folien sind als Link meist in der Überschrift hinterlegt.
- Die Links sind nur im HTML Dokument zu sehen aber auch in der pdf vorhanden.

Links und Quellen

Wen Github näher interessiert:

- Hello World

- Understanding the GitHub flow

Basis R ...

- Wenn man nur R herunterlädt und installiert, sieht das so aus:

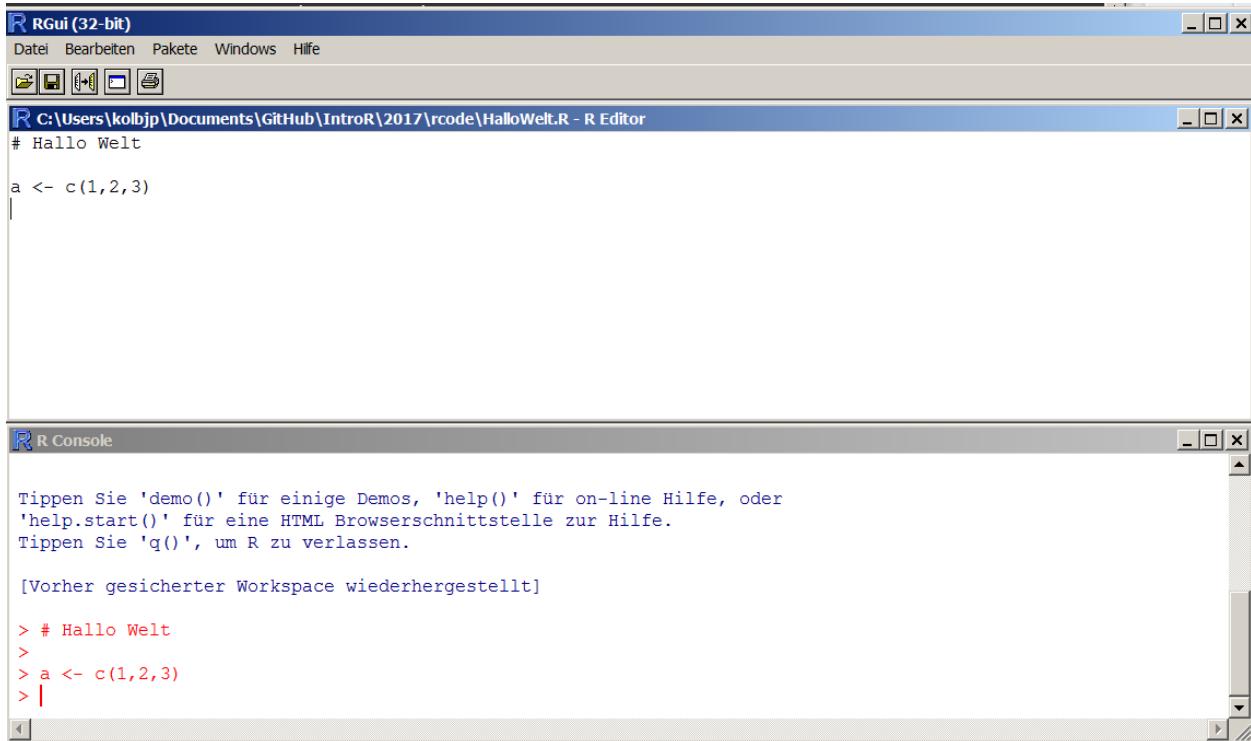


Figure 11:

- So habe ich bis 2012 mit R gearbeitet.

... und Rstudio

- Rstudio bietet Heute sehr viel Unterstützung;
- und macht einige Themen dieses Workshops erst möglich

Aufgabe - Zusatzpakete

Gehen Sie auf <https://cran.r-project.org/> und suchen Sie in dem Bereich, wo die Pakete vorgestellt werden, nach Paketen,...

- für Reproducible Research
- für interaktive Darstellungen
- für High-Performance Computing
- um mit großen Datenmengen umzugehen

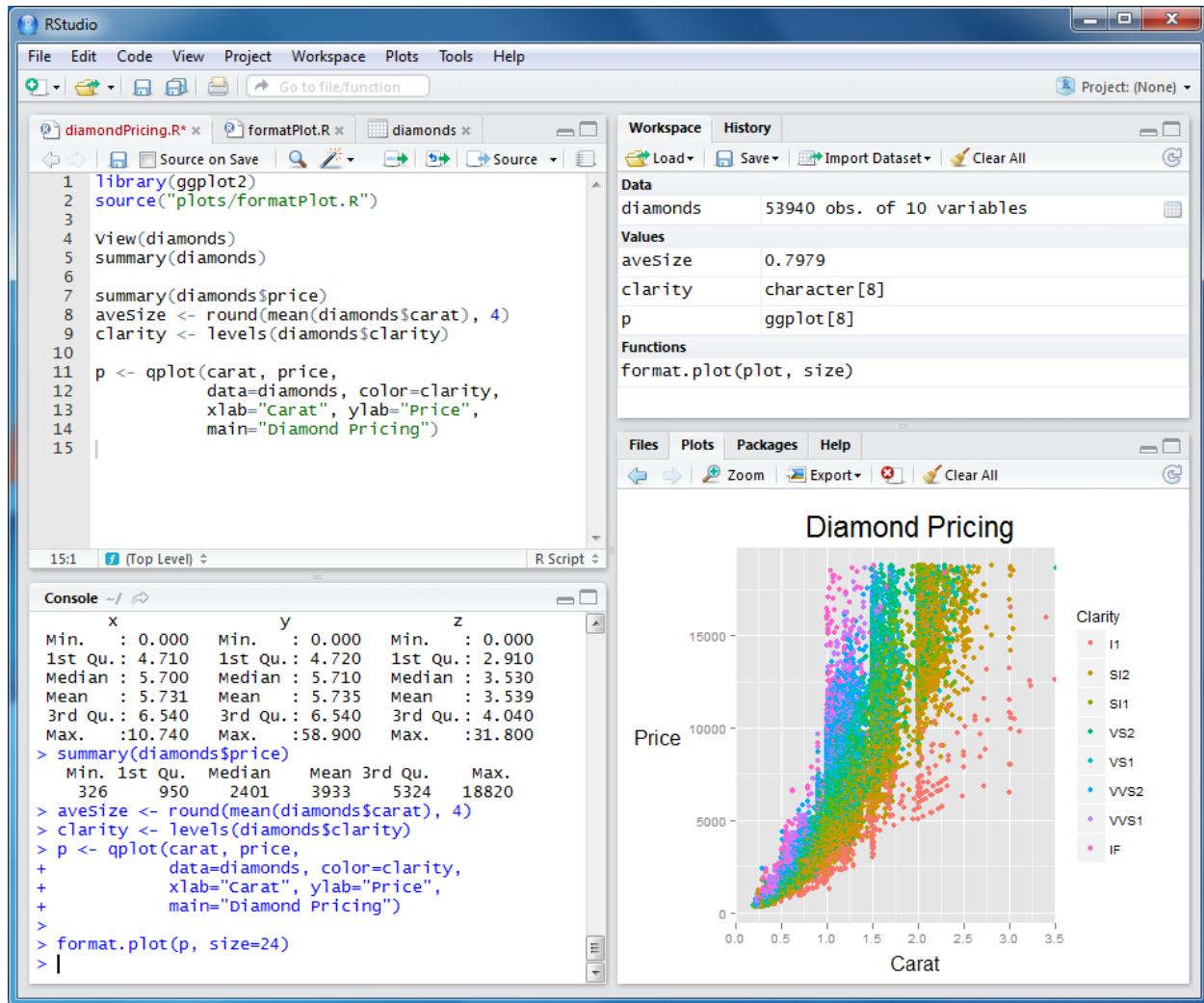


Figure 12:

Datenimport

Dateiformate in R

- Von R werden quelloffene, nicht-proprietary Formate bevorzugt
- Es können aber auch Formate von anderen Statistik Software Paketen eingelesen werden
- R-user speichern Objekte gerne in sog. Workspaces ab
- Auch hier jedoch gilt: (fast) alles andere ist möglich

Formate - base package

R unterstützt von Haus aus schon einige wichtige Formate:

- CSV (Comma Separated Values): `read.csv()`
- FWF (Fixed With Format): `read.fwf()`
- Tab-getrennte Werte: `read.delim()`

Datenimport leicht gemacht mit Rstudio

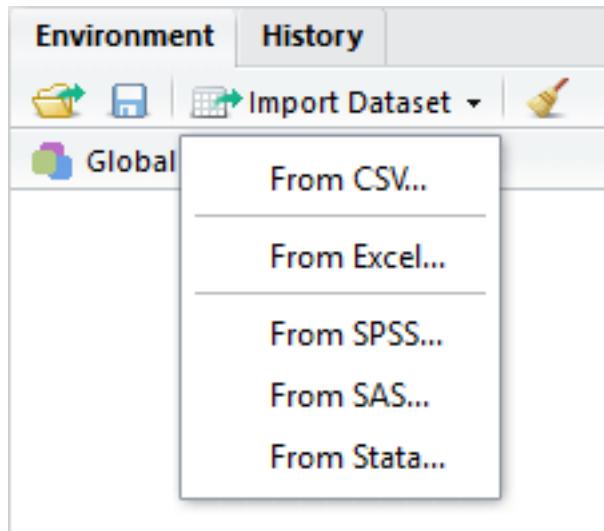


Figure 13: Import Button

CSV aus dem Web einladen

- Datensatz:

<https://data.montgomerycountymd.gov/api/views/6rqk-pdub/rows.csv?accessType=DOWNLOAD>

- Datenimport mit Rstudio

Der Arbeitsspeicher

So findet man heraus, in welchem Verzeichnis man sich gerade befindet

Import Text Data

File/Url: <https://data.montgomerycountymd.gov/api/views/6rqk-pdub/rows.csv?accessType=DOWNLOAD>

Update

Data Preview:

Full Name (character)*	Gender (character)*	Current Annual Salary (character)*	2015 Gross Pay Received (character)*	2015 Overtime Pay (character)*	Department (character)*	Department Name (character)*	Division (character)*	Assignment Category (character)*	Position Title (character)*	Underfilled Job Title (character)
Aarhus, Pam J.	F	\$68878.16	\$72336.79	N/A	POL	Department of Police	MSB Information Management and Technology Division	Fulltime-Regular	Office Services Coordinator	N/A
Aaron, David J.	M	\$96908.09	\$101857.00	\$4640.99	POI	Department of Police	ISB Major Crimes Division Fugitive Section	Fulltime-Regular	Master Police Officer	N/A
Aaron, Marsha M.	F	\$104196.06	\$103019.73	N/A	HHS	Department of Health and Human Services	Adult Protective and Case Management Services	Fulltime-Regular	Social Worker IV	N/A
Abalio, Codred A.	M	\$50697.79	\$51818.46	\$4445.15	COR	Correction and Rehabilitation	PRSS Facility and Security	Fulltime-Regular	Resident Supervisor II	N/A
Abduliv, Essaysas	M	\$92931.00	\$93468.35	N/A	HCA	Department of Housing and Community Affairs	Single Family Housing Program	Fulltime-Regular	Planning Specialist III	N/A
Abbomonte, Drew B.	M	\$67715.00	\$81392.40	\$10027.11	POL	Department of Police	PSB 6th District Special Assignment Team	Fulltime-Regular	Police Officer III	N/A
Abdelmoniem, Marwan M.	M	\$62286.30	\$59663.27	N/A	HHS	Department of Health and Human Services	Head Start	Fulltime-Regular	Administrative Specialist II	N/A
AbduChani, Hasinah J.	F	\$45828.92	\$46783.23	\$6.38	POI	Department of Police	FSB Traffic Division Automated Traffic Enforcement System	Fulltime-Regular	Police Aide	N/A
AbdulJabbar, Soeed	M	\$61040.57	\$66861.98	\$5569.81	DCS	Department of General Services	Facilities Maintenance	Fulltime-Regular	Electrician I	N/A
AbduRahem, Mikael A.	M	\$56404.96	\$71943.08	\$15342.84	DOT	Department of Transportation	Transit Silver Spring Ride On	Fulltime-Regular	Bus Operator	N/A
Abelbe, Hiruth	F	\$151585.60	\$164945.06	N/A	HHS	Department of Health and Human Services	STD and HIV Services	Parttime-Regular	Medical Doctor III - Physician	N/A
Abelbe, Zekarias S.	M	\$44825.99	\$31693.47	\$5240.75	DOT	Department of Transportation	Transit Nicholson Ride On	Fulltime-Regular	Bus Operator	N/A
Abedin, Amireza	M	\$39062.00	\$450.00	N/A	DOT	Department of Transportation	Transportation Management	Fulltime-Regular	Traffic Management Technician II	Traffic Management Technicia
Abelove, Sherry R.	F	\$93436.50	\$90831.10	N/A	HHS	Department of Health and Human Services	Adult Protective and Case Management Services	Fulltime-Regular	Social Worker III	N/A
Abela, Joseph M.	M	\$11781.00	\$115786.22	N/A	DTS	Department of Technology Services	EASD - ERP Applications Support	Fulltime-Regular	Senior Information Technology Specialist	N/A
Abi-Jamra, Rania F.	F	\$53009.99	\$46850.36	N/A	LIB	Department of Public Libraries	Olney Library	Fulltime-Regular	Library Assistant I	N/A
Abijamad, Ryan Z.	M	\$17288.00	\$14665.53	N/A	LIB	Department of Public Libraries	Silver Spring Library	Parttime-Regular	Library Desk Assistant	N/A
Abits, Lydia B.	F	\$40429.58	\$41746.34	\$5500.53	DOT	Department of Transportation	Transit Cathersburg Ride On	Fulltime-Regular	Bus Operator	N/A
Abikiran, Maral	F	\$20295.51	\$9976.52	\$45.28	POL	Department of Police	FSB Traffic Division School Safety Section	Parttime-Regular	Crossing Guard	N/A
Abouraya, Nadia L.	F	\$16602.01	\$15592.92	N/A	HHS	Department of Health and Human Services	Community Support Network for People with Disabilities	Parttime-Regular	Office Clerk	N/A

Previewing first 50 entries.

Import Options:

Name: rows	<input checked="" type="checkbox"/> First Row as Names	Delimiter: Comma	Escape: None
Skip: 0	<input checked="" type="checkbox"/> Trim Spaces	Quotes: Default	Comment: Default
<input checked="" type="checkbox"/> Open Data Viewer			
		Locale: Configure...	NA: Default

Code Preview:

```
library(readr)
rows <- read_csv("https://data.montgomerycountymd.gov/api/views/6rqk-pdub/rows.csv?accessType=DOWNLOAD")
View(rows)
```

Import Cancel

Figure 14:

getwd()

So kann man das Arbeitsverzeichnis ändern:

Man erzeugt ein Objekt in dem man den Pfad abspeichert:

```
main.path <- "C:/"
main.path <- "/users/Name/"
main.path <- "/home/user/"
```

Und ändert dann den Pfad mit setwd()

```
setwd(main.path)
```

Bei Windows ist es wichtig Slashes anstelle von Backslashes zu verwenden.

Alternative - Arbeitsspeicher

Das Paket readr

```
install.packages("readr")
```

```
library(readr)
```

- readr auf dem Rstudio Blogg

Import von Excel-Daten

- library(readr) ist für den Import von fremden Datenformaten hilfreich
- Wenn Excel-Daten vorliegen - als .csv abspeichern

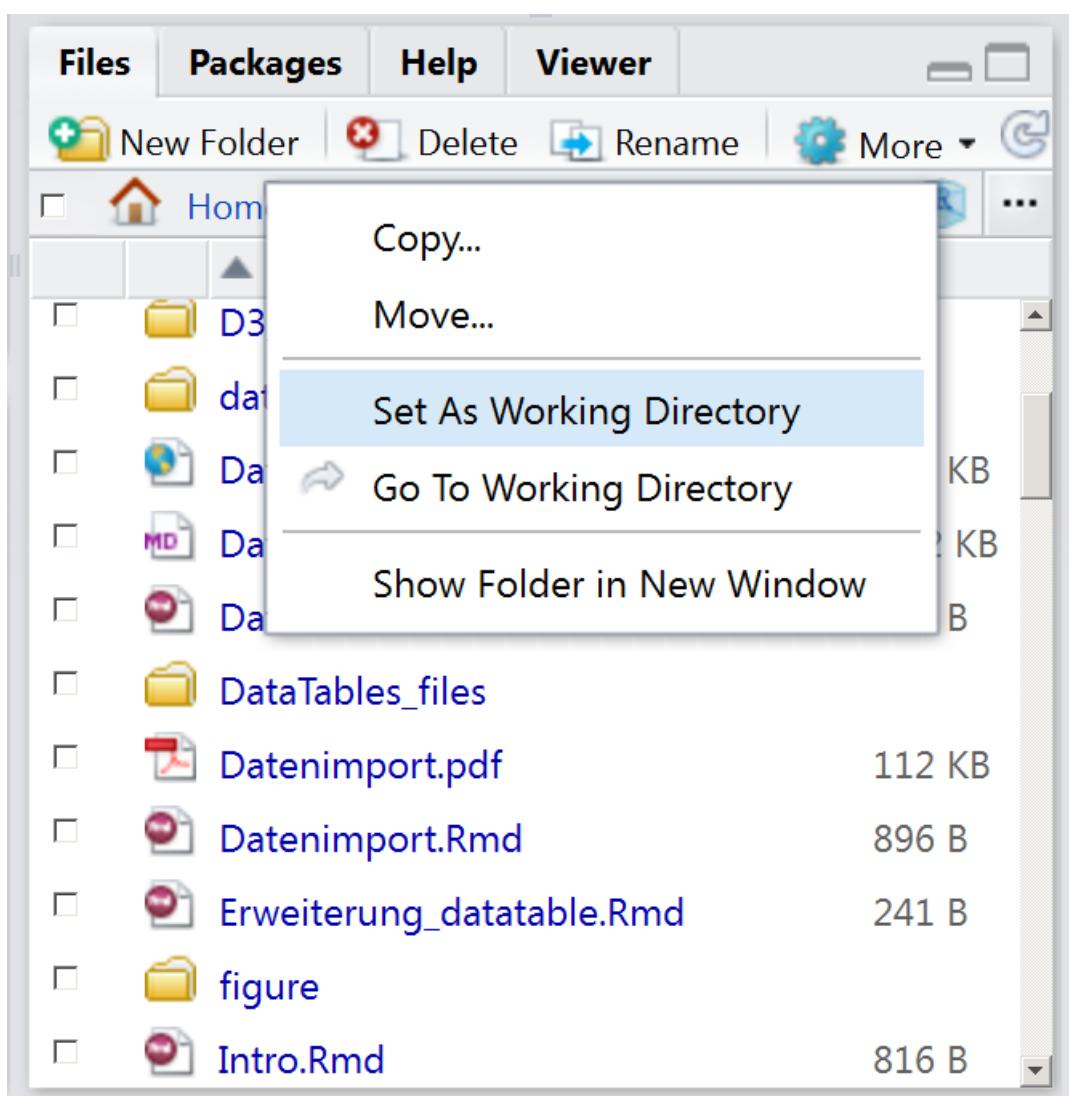


Figure 15:



readr 0.2.0

October 28, 2015 in [Packages](#), [tidyverse](#)

readr 0.2.0 is now available on CRAN. readr makes it easy to read many types of tabular data, including csv, tsv and fixed width. Compared to base equivalents like `read.csv()`, readr is much faster and gives more convenient output: it never converts strings to factors, can parse date/times, and it doesn't munge the column names.

This is a big release, so below I describe the new features divided into four main categories:

Figure 16:

```
library(readr)
rows <- read_csv("https://data.montgomerycountymd.gov/api/views/6rqk-pdub/rows.csv?accessType=DOWNLOAD")

.csv-Daten aus dem Web importieren - zweites Beispiel

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

whcSites <- read.csv(url)

head(data.frame(whcSites$name_en,whcSites$category))

##
# whcSites.name_en
## 1 Cultural Landscape and Archaeological Remains of the Bamiyan Valley
## 2 Minaret and Archaeological Remains of Jam
## 3 Historic Centres of Berat and Gjirokastra
## 4 Butrint
## 5 Al Qal'a of Beni Hammad
## 6 M'Zab Valley

## whcSites.category
## 1 Cultural
## 2 Cultural
## 3 Cultural
## 4 Cultural
## 5 Cultural
## 6 Cultural
```

Das Paket `haven`

```
install.packages("haven")
```

```
library(haven)
```

- Das R-Paket `haven` auf dem Rstudio Blogg



SPSS Dateien einlesen

- Zunächst muss wieder der Pfad zum Arbeitsverzeichnis angeben werden.
- SPSS-Dateien können auch direkt aus dem Internet geladen werden:

```
install.packages("haven")
```

```
library(haven)
```

```
mtcars <- read_sav("https://github.com/Japhilko/RInterfaces/raw/master/data/mtcars.sav")
```

stata Dateien einlesen

```
library(haven)
```

```
oecd <- read_dta("https://github.com/Japhilko/IntroR/raw/master/2017/data/oecd.dta")
```

Links

- Quick-R - Import Data
- Datenimport bei R-bloggers
- Importing Data into R

- Mapping von Arbeitslosendaten in den USA
- Das Paket readr

Datenexport

Die Exportformate von R

- In R werden offene Dateiformate bevorzugt
- Genauso wie `read.X()` Funktionen stehen viele `write.X()` Funktionen zur Verfügung
- Das eigene Format von R sind sog. Workspaces (`.RData`)

Beispieldatensatz erzeugen

```
A <- c(1,2,3,4)
B <- c("A","B","C","D")

mydata <- data.frame(A,B)
```

Überblick Daten Import/Export

- wenn mit R weitergearbeitet wird, eignet sich das `.RData` Format am Besten:

```
save(mydata, file="mydata.RData")
```

Daten in .csv Format abspeichern

```
write.csv(mydata,file="mydata.csv")
```

- Wenn mit Deutschem Excel weitergearbeitet werden soll, eignet sich `write.csv2` besser

```
write.csv2(mydata,file="mydata.csv")
```

- Sonst sieht das Ergebnis so aus:

	A	
1	,"A","B"	
2	1,1,"A"	
3	2,2,"B"	
4	3,3,"C"	
5	4,4,"D"	
6		

Figure 18:

The screenshot shows a web page titled "R xlsx package : A quick start guide to manipulate Excel files in R". At the top, there are navigation links for "AdChoices", "Microsoft Excel", "Download for Java", and "Excel Tutorial". Below the navigation, there is a sidebar with three items: "Install and load xlsx package", "Read an Excel file", and "Write data to an Excel file".

Figure 19:

Das Paket xlsx

```
library(xlsx)
write.xlsx(mydata,file="mydata.xlsx")
```

Das Paket foreign

Reading/Writing Stata (.dta) files with Foreign

December 4, 2012

By is.R()

Figure 20:

- Funktionen im Paket `foreign`

Daten in stata Format abspeichern

```
library(foreign)
write.dta(mydata,file="data/mydata.dta")
```

Das Paket rio

```
install.packages("rio")
```

Daten als .sav abspeichern (SPSS)

```
library("rio")
# create file to convert

export(mtcars, "data/mtcars.sav")
```

R topics documented:

lookup.xport	2
read.arff	3
read.dbf	4
read.dta	5
read.epiinfo	7
read.mtp	8
read.octave	9
read.spss	10
read.ssd	12
read.systat	14
read.xport	15
S3 read functions	16
write.arff	17
write.dbf	18
write.dta	19
write.foreign	21

Figure 21:

Import, Export, and Convert Data Files

The idea behind `rio` is to simplify the process of importing data into R and exporting data from R. This process is, probably unnecessarily, extremely complex for beginning R users. Indeed, R supplies [an entire manual](#) describing the process of data import/export. And, despite all of that text, most of the packages described are (to varying degrees) out-of-date. Faster, simpler, packages with fewer dependencies have been created for many of the file types described in that document. `rio` aims to unify data I/O (importing and exporting) into two simple functions: `import()` and `export()` so that beginners (and experienced R users) never have to think twice (or even once) about the best way to read and write R data.

Figure 22:

Dateiformate konvertieren

```
export(mtcars, "data/mtcars.dta")  
  
# convert Stata to SPSS  
convert("data/mtcars.dta", "data/mtcars.sav")
```

Links Export

- Quick R für das Exportieren von Daten:
- Hilfe zum Export auf dem cran Server
- Daten aus R heraus bekommen

R und Excel

Das Paket `xlsx`

- Eine wichtige Datenquelle - Eurostat

```
library("xlsx")  
dat <- read.xlsx("cult_emp_sex.xls", 1)
```

Einige Schritte um R und Excel zu verbinden

- Die Excel-Verbindung

```
install.packages("XLConnect")  
  
library("XLConnect")
```



Figure 23: Vignette für XLconnect

Eine Excel Datei aus R erzeugen

```
fileXls <- "data/newFile.xlsx"  
unlink(fileXls, recursive = FALSE, force = FALSE)  
exc <- loadWorkbook(fileXls, create = TRUE)
```

```
createSheet(exc, 'Input')
saveWorkbook(exc)
```

Das Arbeitsblatt mit Daten befüllen

```
input <- data.frame('inputType'=c('Day', 'Month'), 'inputValue'=c(2,5))
writeWorksheet(exc, input, sheet = "input", startRow = 1, startCol = 2)
saveWorkbook(exc)
```

BERT - Eine weitere Verbindung zwischen R und Excel

- Schnellstart mit Excel

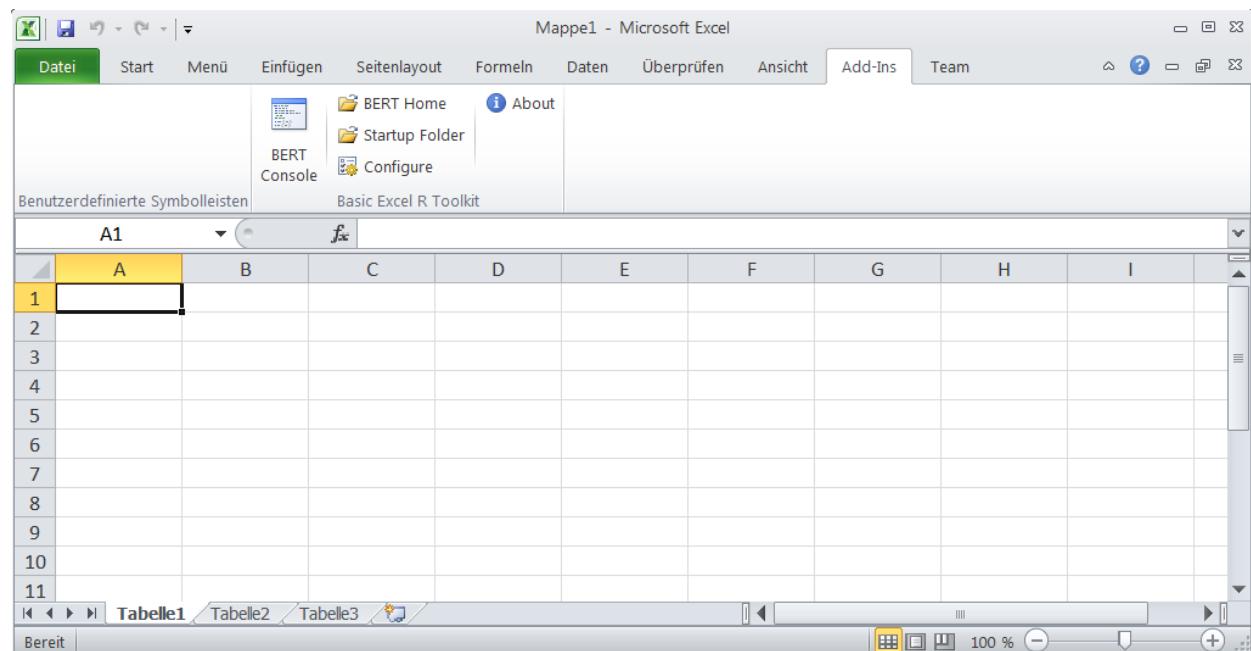


Figure 24:

```
myFunction <- function(){
  aa <- rnorm(200)
  bb <- rnorm(200)
  res <- lm(aa~bb)$res
  return(res)
}
```

Link BERT

- BERT: a newcomer in the R Excel connection

BERT: a newcomer in the R Excel connection

November 30, 2016

By The R Trader



(This article was first published on [R – The R Trader](#), and kindly contributed to [R-bloggers](#))

163
SHARES

f Share

t Tweet

A few months ago a reader point me out this new way of connecting R and Excel. I don't know for how long this has been around, but I never came across it and I've never seen any blog post or article about it. So I decided to write a post as the tool is really worth it and before anyone asks, I'm not related to the company in any way.

Figure 25:

Das Paket readxl

- `readxl`

```
install.packages("readxl")
```

```
library(readxl)
```

Präsentation von Daten - Reproducible Research

CRAN Taskview zu reproducible research

Reproducible Research with R and RStudio

- Ein Online Buch mit vielen nützlichen Tipps

Get data out of excel and into R with readxl

April 15, 2015

By hadleywickham



(This article was first published on [RStudio Blog](#), and kindly contributed to [R-bloggers](#))



I'm pleased to announced that the first version of `readxl` is now available on CRAN. `Readxl` makes it easy to get tabular data out of excel. It:

- Supports both the legacy `.xls` format and the modern xml-based `.xlsx` format. `.xls` support is made possible the with `libxls` C library, which abstracts away many of the complexities of the underlying binary format. To parse `.xlsx`, we use the insanely fast `RapidXML` C++ library.

Figure 26:

CRAN Task View: Reproducible Research
Maintainer: Max Kuhn
Contact: max.kuhn at pfizer.com
Version: 2015-12-03
URL: <https://CRAN.R-project.org/view=ReproducibleResearch>

The goal of reproducible research is to tie specific instructions to data analysis and experimental data so that scholarship can be recreated, better understood and verified. Packages in R for this purpose can be split into groups for: literate programming, package reproducibility, code/data formatting tools, format convertors, and object caching.

Figure 27:

Christopher Gandrud

Reproducible Research with R and RStudio Second Edition

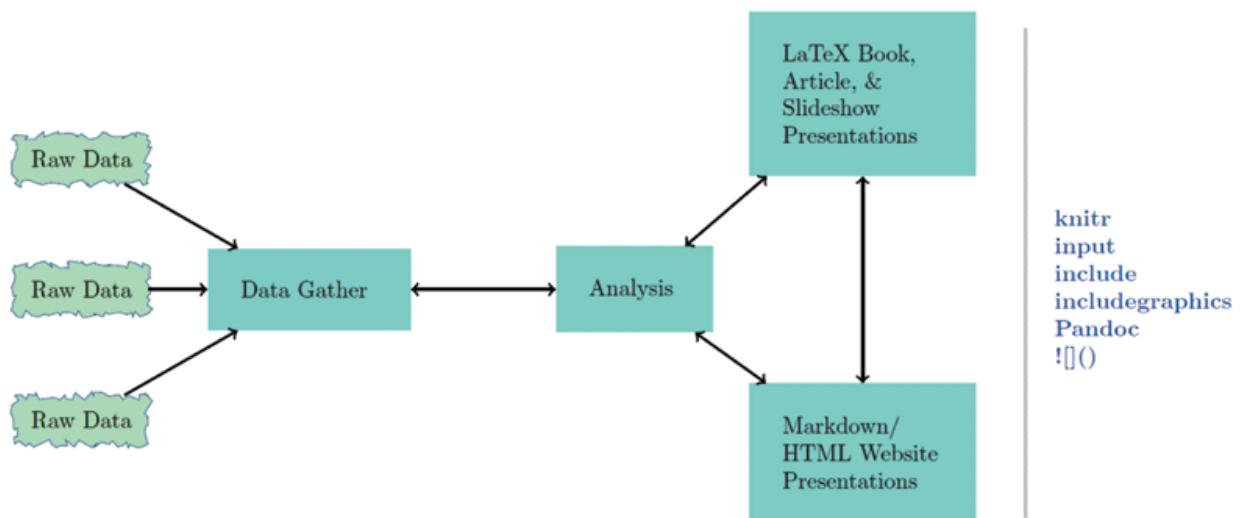
Figure 28:

A crash course in reproducible research in R

written October 14, 2016 in r, programming tips

A couple of weeks ago, I wrote a post giving you an introduction to [reproducible research](#) in Python. While the principles of reproducibility stay the same no matter the language you are using, there are some specific libraries and tools that R has that differ from Python. In this blog post, I'll fill you in on how I conduct a reproducible analysis in R and, like with Python, you'll see how straightforward it is!

Figure 29:



Make download.file source_data source_DropboxData read.table merge getURL API-based packages	source_data source_DropboxData read.table getURL	knitr rmarkdown source source_url kable print(xtable()) texreg
--	---	--

Figure 30:

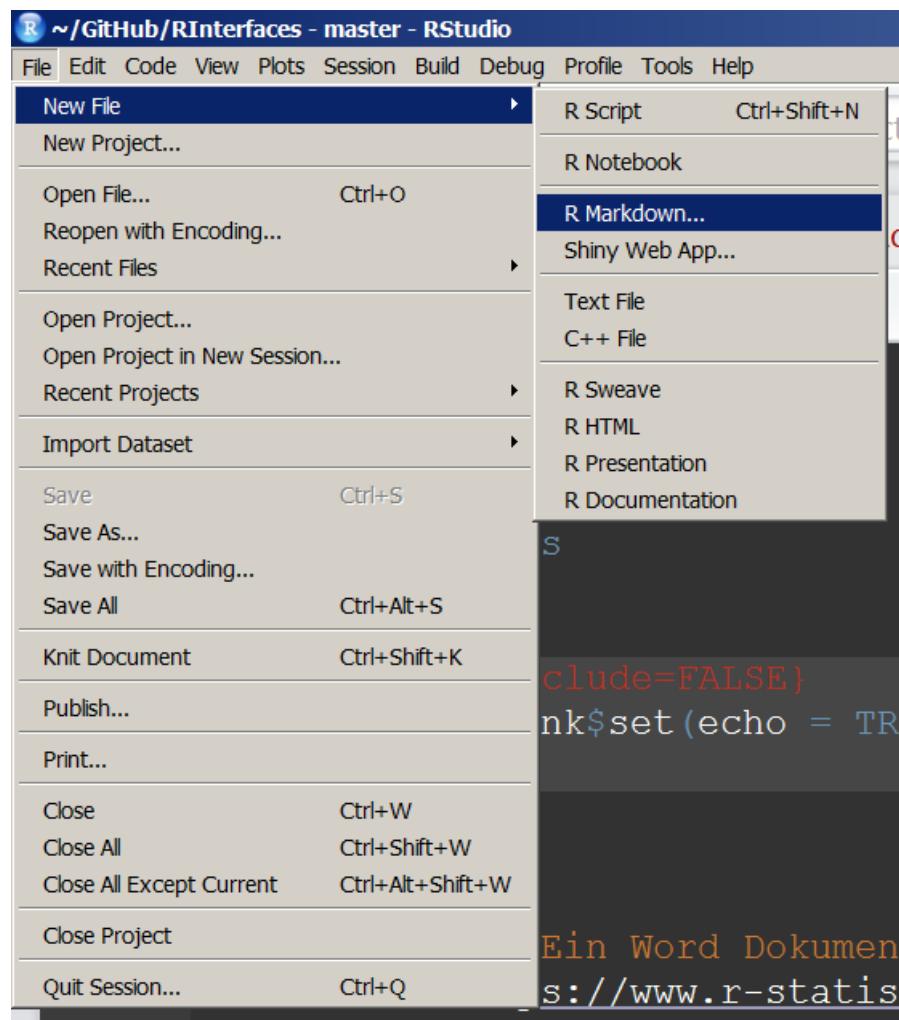


Figure 31:

Ein Crash Kurs in Reproducible Research mit R

Umsetzung in R

Word Dokumente mit R erstellen

Ein Markdown Dokument mit Rstudio erzeugen

Mein erstes mit R erzeugtes Word Dokument

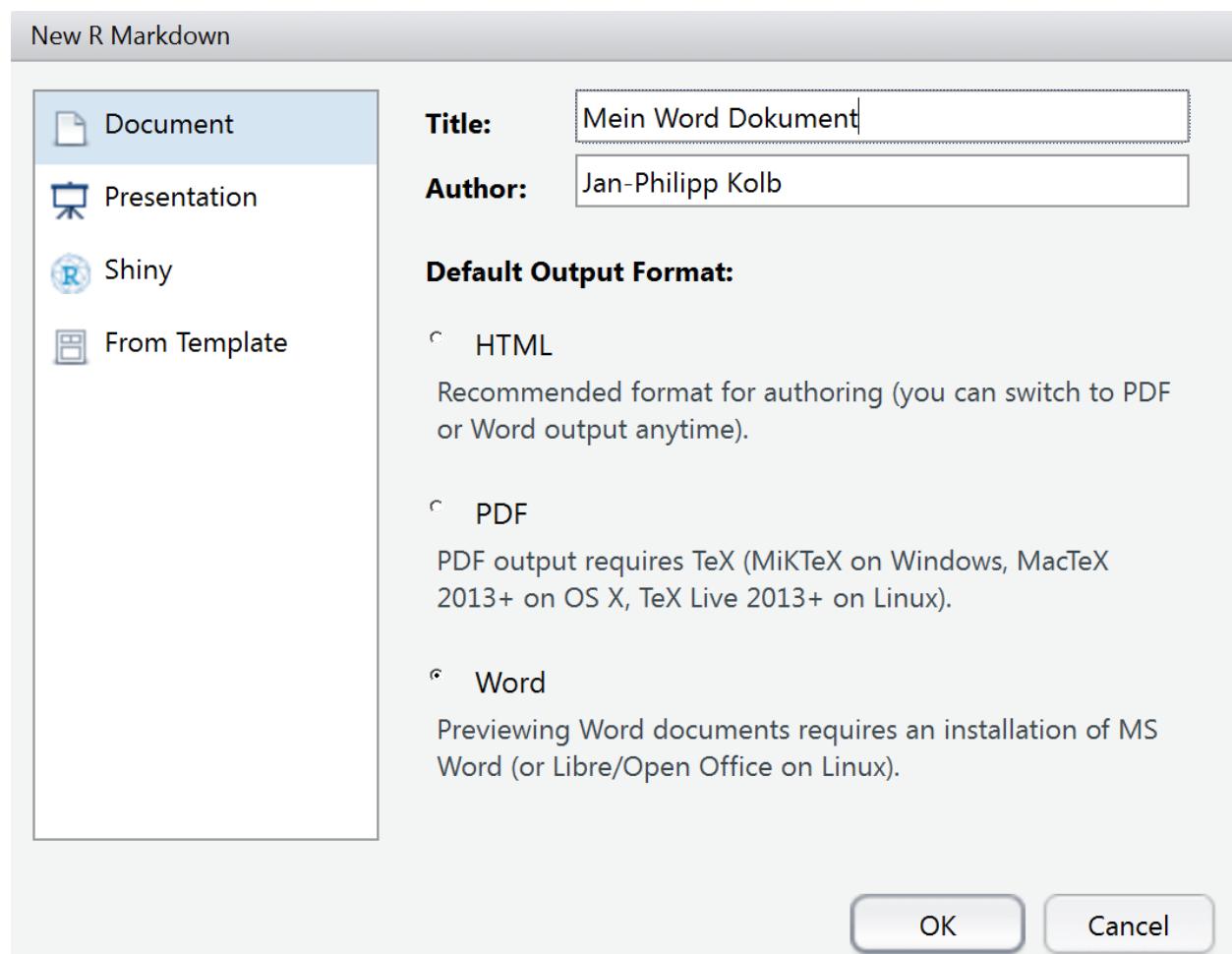


Figure 32:

Erstes Beispiel

Das Arbeiten mit Rmarkdown - erste Schritte

Markdown ist eine sehr einfache Syntax, die es Benutzern erlaubt, aus einfachen Textdateien gut gelayoutete Dokumente zu erstellen.

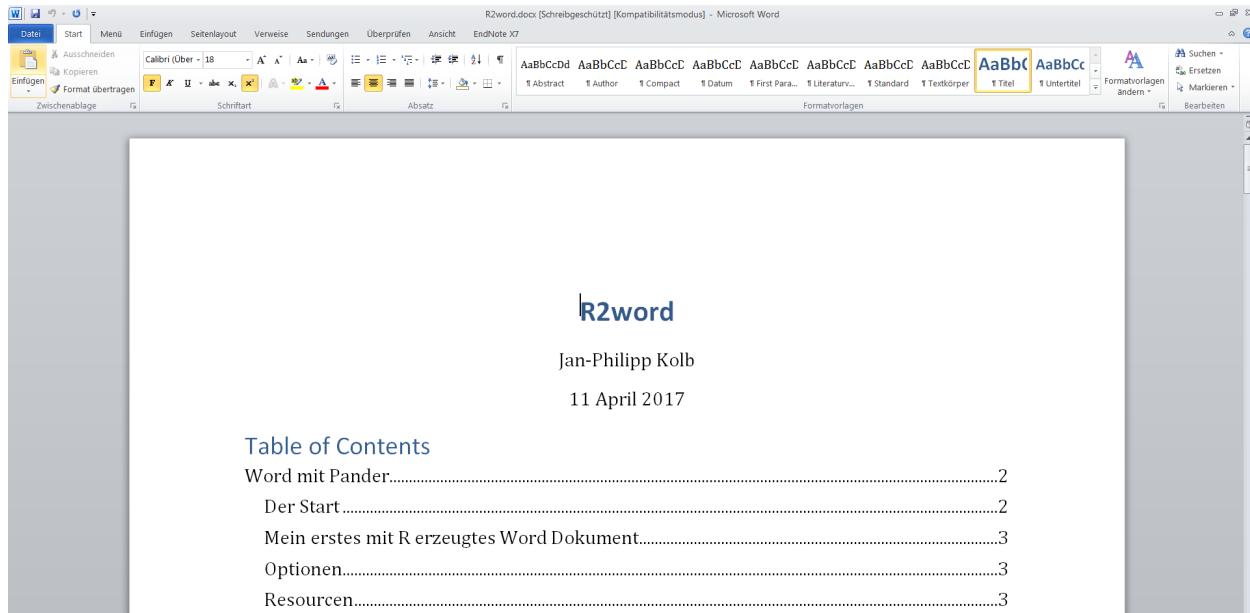


Figure 33:

****fettes Beispiel****
kursives Beispiel
~~durchgestrichen~~
- Aufzählungspunkt

fettes Beispiel

kursives Beispiel

durchgestrichen

- Aufzählungspunkt

Weitere Markdown Befehle

Überschrift Ebene 3
 ##### Überschrift Ebene 4
[\[Meine Github Seite\]](https://github.com/Japhilko) (<https://github.com/Japhilko>)

Überschrift Ebene 3

Überschrift Ebene 4

Meine Github Seite

Weitere Markdown Befehle

- So kann man Bilder einbinden:
- Man kann entweder einen Link angeben:

! [BSP] (http://e-scientifics.de/content/example_kinderbild.jpg)

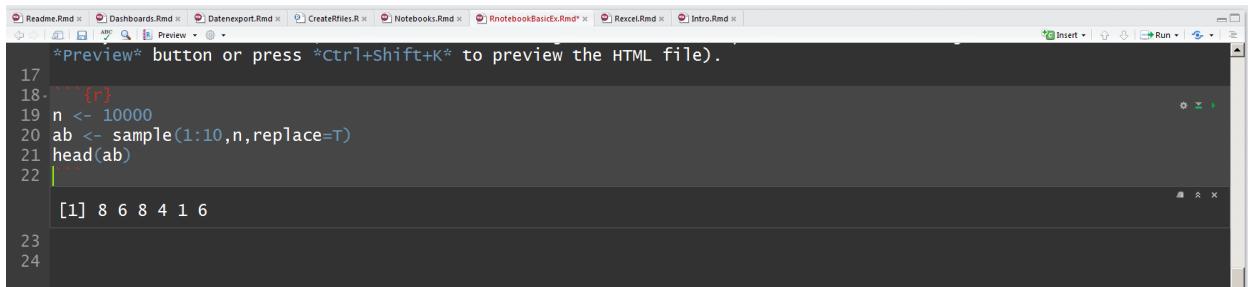
- oder einen (Unterordner) in dem das Bild liegt:

! [BSP 2] (figure/example.png)

- in den eckigen Klammern steht die Bildunterschrift
- alle gängigen Formate (.png, .jpeg,.gif) können so eingebunden werden
- Man kann auch noch weitere Optionen spezifizieren (Größe, Breite etc.) - dazu später mehr

Chunks - Erste Schritte

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



The screenshot shows an RStudio interface with several tabs at the top: Readme.Rmd, Dashboards.Rmd, Datenexport.Rmd, CreateFiles.R, Notebooks.Rmd, RnotebookBasicEx.Rmd*, RExcel.Rmd, and Intro.Rmd*. The main area displays R code and its output. The code is as follows:

```

17 *Preview* button or press *ctrl+shift+K* to preview the HTML file).
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

```

Figure 34:

Button um Chunks einzufügen

- Die default Version eines Chunks ist R
- Man hat aber auch die Möglichkeit andere Programmiersprachen einzubinden

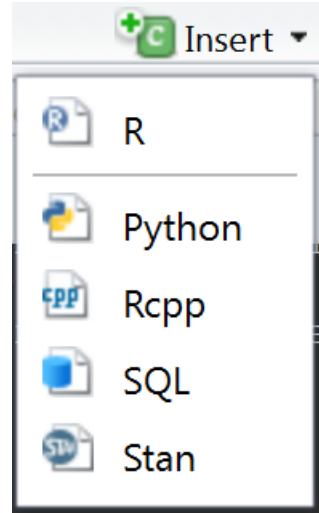


Figure 35:

```
n = 100
```

```
# Ein inline Codeblock: `r n`
```

Figure 36:

Inline Code

```
n=100
```

Ein inline Codeblock: 100

Chunk Optionen

- Man kann den Chunks Optionen mitgeben:

Argument	Beschreibung
eval	Soll Rcode evaluiert werden?
warning	Sollen Warnings angezeigt werden?
cache	Soll der Output gespeichert werden?

- Bei eval kann ein logischer Wert angegeben werden oder eine/mehrere Nummer(n)

Optionen

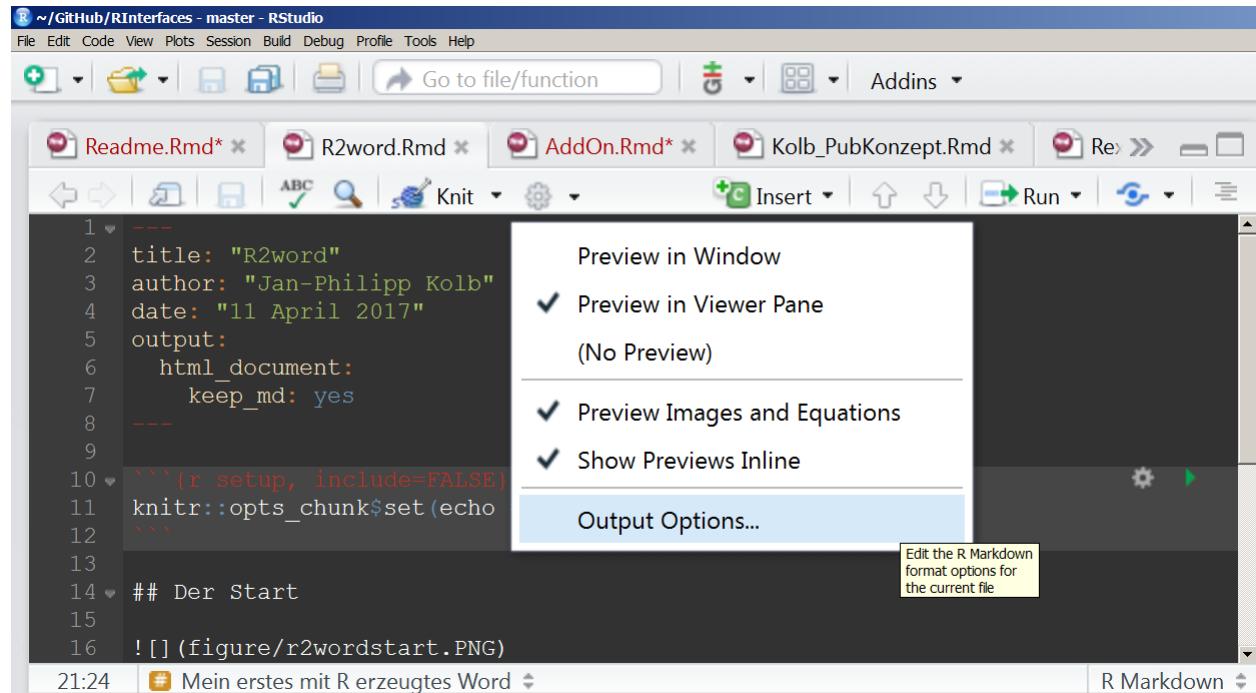


Figure 37:

Optionen für Word Output

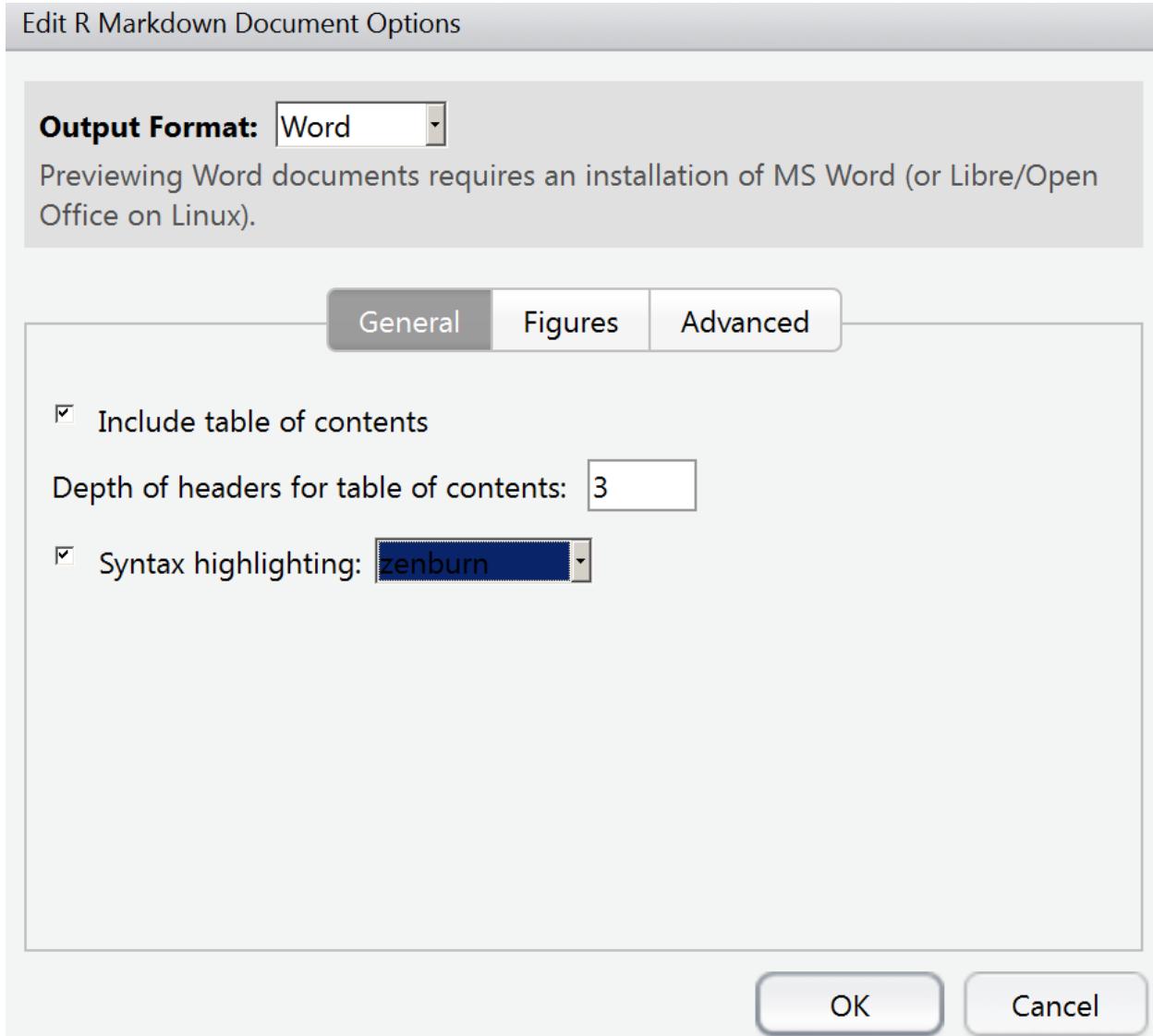


Figure 38:

```
# Beispiel für Code
ab <- sample(1:10,5,replace=T)
summary(ab)

##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.
##      2.0    3.0    5.0    5.4    8.0    9.0
```

Figure 39:

- tango

```
# Beispiel für Code
ab <- sample(1:10,5,replace=T)
summary(ab)

##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.
##      5.0    6.0    6.0    6.6    7.0    9.0
```

Figure 40:

Das Paket knitr

```
install.packages("knitr")
library("knitr")
```

- Das Paket knitr enthält zahlreiche wichtige Funktionen
- Beispiel: Befehl `kable` um Tabellen zu erzeugen

Eine Tabelle mit `kable` erzeugen

```
a <- runif(10)
b <- rnorm(10)
ab <- cbind(a,b)
kable(ab)
```

	a	b
0.2692422	2.0136755	
0.0936691	-0.8991878	
0.4148262	-0.6697934	
0.9064192	-1.4818679	
0.5840679	0.6818610	
0.1346803	-1.0135759	
0.5500504	0.6208644	
0.5811987	0.3415814	
0.5722638	0.0234431	
0.1841931	0.5397654	

Vorlagen verwenden

- Formatvorlagen können verändert werden
1. Ein Word Dokument mit Rmarkdown erstellen
 2. Das Dokument in Word öffnen und Format verändern
 3. Vorlage als Referenz angeben

```
1 ---  
2 title: "R2word"  
3 author: "Jan-Philipp Kolb"  
4 date: "11 April 2017"  
5 output:  
6   word_document:  
7     reference_docx: RefDoc.docx  
8     highlight: zenburn  
9     toc: yes  
10 ---
```

Figure 41:

Immer das aktuelle Datum im Kopf

date: "10 Mai, 2017"

```
1 ---  
2 title: "RPostgreSQL"  
3 author: "Jan-Philipp Kolb"  
4 date: `r format(sys.time(), '%d %B, %Y')`  
5 output:  
6   slidify_presentation:  
7     keep_md: yes  
8 ---
```

Figure 42:

Ein Schummelzettel

Resourcen

- Interview - Ein Word Dokument mit wenig Aufwand schreiben
- pander: Ein R Pandoc Wrapper
- Einführung in Markdown
- Warum TeX besser als Word ist

RPostgreSQL

Jan-Philipp Kolb

06 Mai, 2017

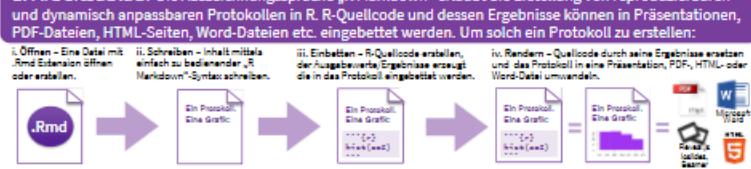
Figure 43:

R Markdown

Schummelzettel
Mehr auf rmarkdown.rstudio.com
rmarkdown 0.2.50 Update: 8/14



1. Arbeitsablauf Die Auszeichnungssprache „R Markdown“ erlaubt die Erstellung von reproduzierbaren und dynamisch anpassbaren Protokollen in R-R-Quellcode und dessen Ergebnissen können in Präsentationen, PDF-Datenien, HTML-Seiten, Word-Datenien etc. eingebettet werden. Um solch ein Protokoll zu erstellen:



2. Datei erstellen Zuerst wird eine Textdatei mit der Extension .Rmd erstellt oder ein RStudio Rmd-Template geöffnet.

- Zum Erstellen eines „R Markdown“ Protokolls im Menü folgendes auswählen:
Datei ► Neues Dokument ► R Markdown... Oder im englischen Menü:
File ► New File ► R Markdown...
- Im neuen Fenster den Typ des Dokuments auswählen, das mittels Rmd erstellt werden soll. Das entsprechende Optionfeld für die Dateianwahl kann nachträglich geändert werden.
- OK klicken.

4. Ausgabeformat auswählen Eine YAML Kopfzeile wird erstellt. Sie verdeutlicht, welchen Dokumenttyp die R Markdown Datei erzeugen soll.

YAML
Eine YAML-Kopfzeile besteht aus Schlüssel-Wert-Paaren am Beginn der Datei. Drei Bindestriche (---) bilden Anfang und Ende dieser Kopfzeile.

```
title: "Unbenannt"
author: "Unbenannt"
output: html_document
---
```

Hier ist der Anfang des Protokolls. Der obige Text enthält die Metadaten der YAML-Kopfzeile.

Anhand dieser Rmd Datei kreiert R eine neue Datei für das Protokoll. Der Wert von output bestimmt die Dateiart (siehe auch 8), z.B.

- output: html_document HTML-Datei (Webseite)
- output: pdf_document PDF-Datei
- output: word_document Microsoft Word-Datei (DOCX)
- output: beamer_presentation beamer Slideshow (PDF)
- output: ioslides_presentation ioslides Slideshow (HTML)

RStudio hat eingetragenes Materialien von RStudio, Inc. - CC-BY RStudio - Info@RStudio.com - 24440-0111 - RStudio.com

3. Markdown schreiben Anschließend wird das Protokoll in Klartext geschrieben. Zur Formattierung wird Markdown-Syntax verwendet.

Syntax	Resultat
# Überschrift 1	Überschrift 1
## Überschrift 2	Überschrift 2
### Überschrift 3	Überschrift 3
#### Überschrift 4	Überschrift 4
##### Überschrift 5	Überschrift 5
###### Überschrift 6	Überschrift 6
italic	Fett mit 2 Klemmchen um neuen Absatz zu beginnen, fettfett und ...fett
bold	fett
underline	hochgestrichen
[hyperlink](www.rstudio.com)	Hyperlink
* Überschriften 1	
** Überschriften 2	
*** Überschriften 3	
**** Überschriften 4	
***** Überschriften 5	
***** Überschriften 6	
---	Unterstrich (engl. endash): -- Gedankenstrich (engl. emdash): --- Auslassungspunkte (engl. ellipsis): ... einzelne Zeichen: s a r [s] [a] [r]
file: [/path/to/file.Rnw]	file: [/path/to/file.Rnw]
weisse Strich:	weisse Strich:

> Blockzitat	Blockzitat
+ unordnete Liste	+ unordnete Liste
* Punkt 1	* Punkt 1
+ Unterpunkt 1	+ Unterpunkt 1
+ Unterpunkt 2	+ Unterpunkt 2
1. geordnete Liste	1. geordnete Liste
2. Punkt 1	2. Punkt 1
+ Unterpunkt 1	+ Unterpunkt 1
+ Unterpunkt 2	+ Unterpunkt 2
Tabellenüberschrift	Zweite Überschrift
Tabellenelement	Element 2
Element 3	Element 4

Figure 44:

32

PDF Dokumente und Präsentationen mit LaTeX, Beamer und Sweave

Präsentationen mit Rmarkdown - beamer Präsentationen

```
Import csv
```

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

```
Jan-Philipp Kolb Datenimport
```

Figure 45:

Beamer Optionen

Beamer Themen

Chunks einfügen

- Auch hier lassen sich natürlich Chunks einfügen
- Wenn `cache=T` angegeben ist, wird das Ergebnis des Chunks abgespeichert
- Es ist sinnvoll die Chunks zu benennen, dann findet man auch das Ergebnis einfacher

Edit R Markdown Presentation Options

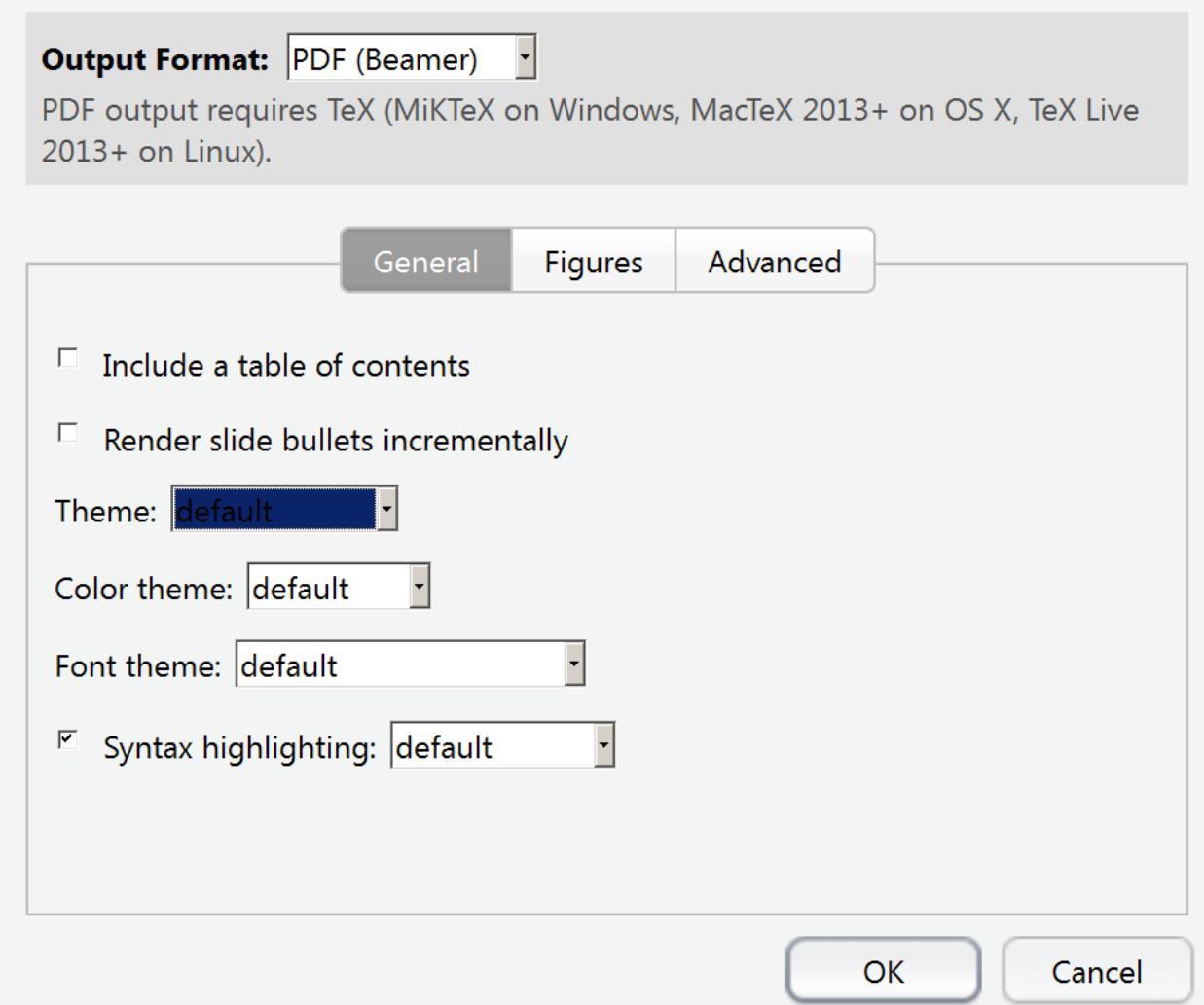


Figure 46:

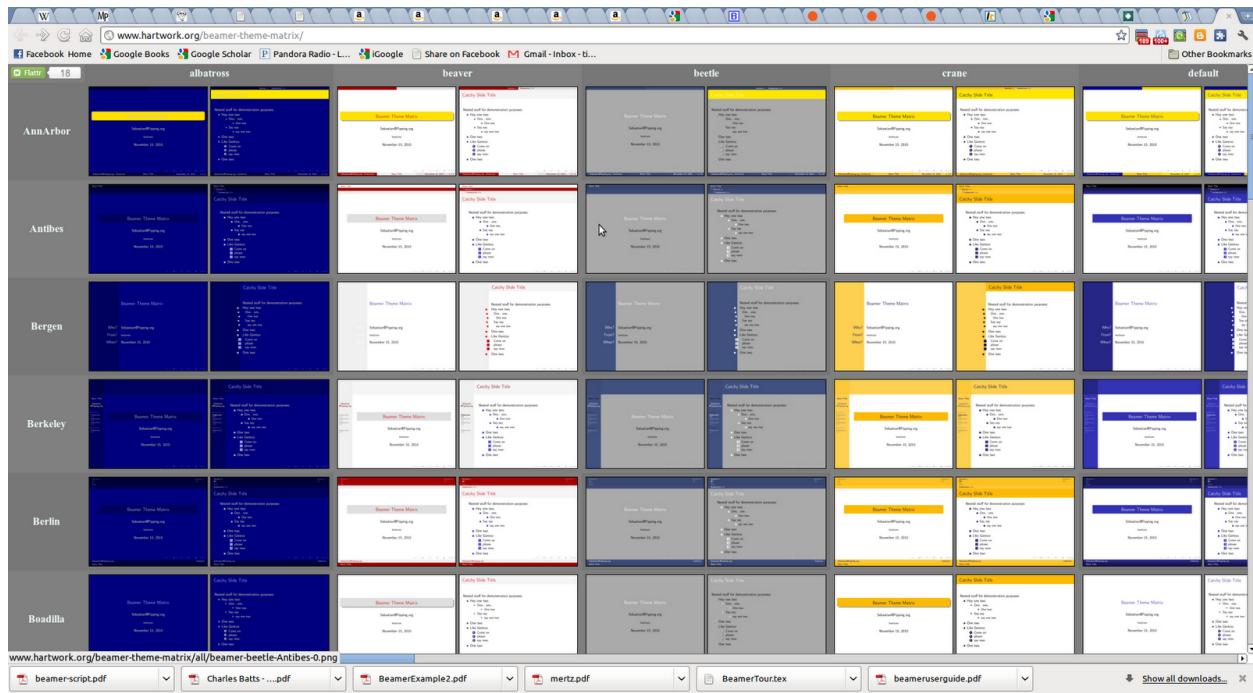


Figure 47:

```
```{r Zufallszahlen, cache=TRUE}
ab <- runif(1000)
```

```

Figure 48:

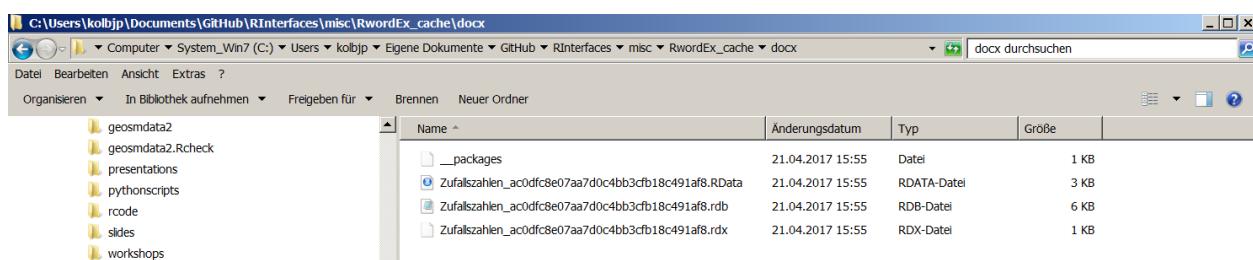


Figure 49:

Ergebnis - Cache

Wie man das im Header des Dokuments angibt

```
---
```

```
title: "Intro - Erste Schritte"
author: "Jan-Philipp Kolb"
date: "10 April 2017"
output:
  beamer_presentation:
    colortheme: beaver
    theme: CambridgeUS
---
```

Inhaltsverzeichnis I

```
---
```

```
output:
  beamer_presentation:
    toc: true
---
```

Figure 50:

```
output:
  beamer_presentation:
    toc: yes
```

Optionen für die Graphikeinbindung

- *fig_caption: false*, wenn man keine Bildunterschriften haben möchte

```
---
```

```
title: "Habits"
output:
  beamer_presentation:
    fig_width: 7
    fig_height: 6
    fig_caption: true
---
```

Figure 51:

- Man sollte keine Bilder im Format .svg einbinden

Präsentationen mit Sweave

- Das Dokument erstellen

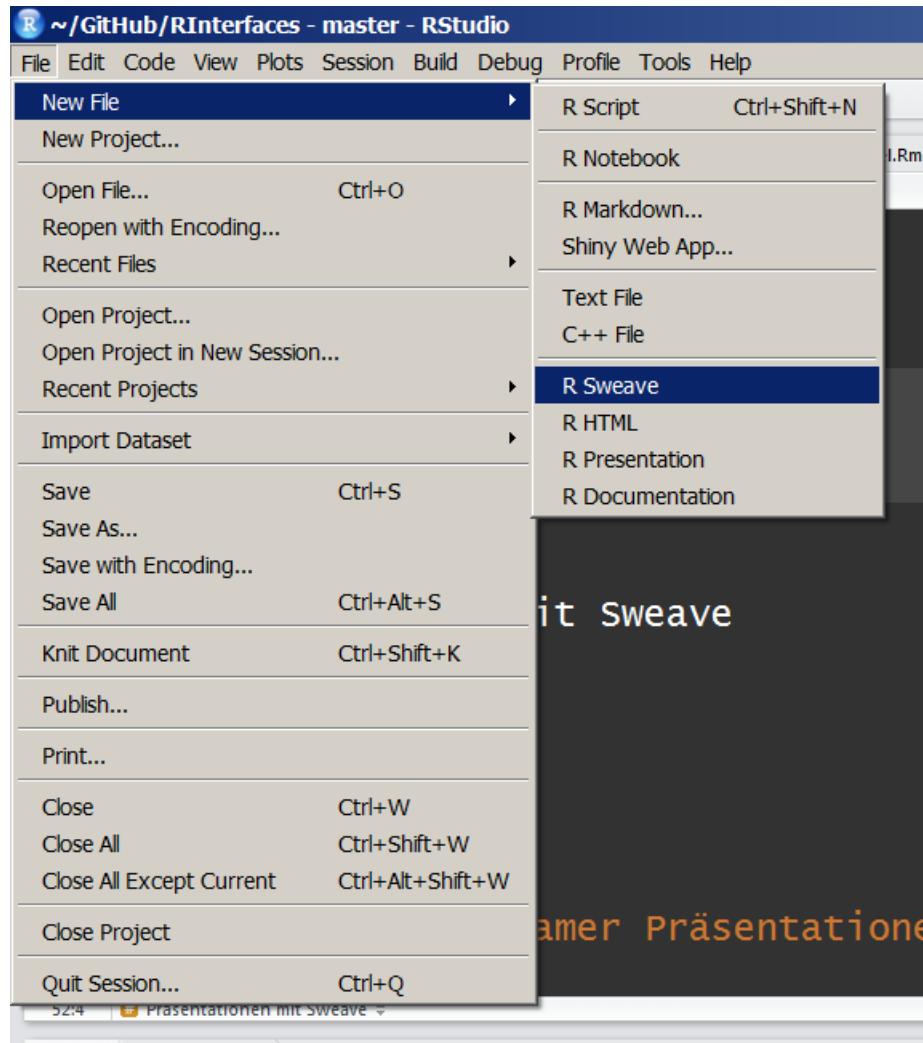


Figure 52:

Sweave Präsentation

- Ganz normaler LaTeX Code wird verwendet

```
1 \documentclass{beamer}
2
3 \begin{document}
4
5 \begin{frame}
6 \frametitle{Erste Folie}
7 \end{frame}
8
9
10 \end{document}
```

Figure 53:

Chunks bei Sweave

- Auch hier kann R-code verwendet werden

```
9 \begin{frame}
10 \frametitle{zweite Folie}
11
12 <>>=
13 ab <- sample(1:10,8)
14 @
15
16 \end{frame}
```

Figure 54:

Chunk Optionen

- Auch bei Sweave Chunks können Optionen mitgegeben werden

Inline Code

- Manchmal braucht man das Ergebnis direkt auf der Folie

\Sexpr{}

| | |
|---------------------------|--|
| <code>echo=false</code> | R-Code ist nicht sichtbar |
| <code>results=tex</code> | die Ergebnisse werden wie TeX-Code behandelt |
| <code>results=hide</code> | die Ergebnisse sind nicht sichtbar |
| <code>fig=true</code> | Grafiken werden eingebunden |
| <code>width=</code> | Breite der Grafik in Inch (z. B. 4) |
| <code>height=</code> | Höhe der Grafik in Inch |

Figure 55:

```

5 \begin{frame}
6 <>>=
7 CRANmirror <- "http://cran.revolutionanalytics.com"
8
9 cran <- contrib.url(repos = CRANmirror, type = "source")
10
11 info <- available.packages(contriburl = cran, type = x)
12 @
13
14 Es gibt aktuell \Sexpr{nrow(info)} Pakete auf CRAN.
15 \end{frame}

```

Figure 56:

```

CRANmirror <- "http://cran.revolutionanalytics.com"
cran <- contrib.url(repos = CRANmirror, type = "source")

info <- available.packages(contriburl = cran, type = x)

```

Es gibt aktuell 10423 Pakete auf CRAN.

Figure 57:

Inline Code - das Ergebnis

```
CRANmirror <- "http://cran.revolutionanalytics.com"
cran <- contrib.url(repos = CRANmirror, type = "source")
info <- available.packages(contriburl = cran, type = x)
nrow(info)

## [1] 10501
```

PDF Paper mit R

- Mit R ist es möglich Berichte oder Paper zu erzeugen
- Dies eignet sich besonders gut, wenn man viel Code hat oder einen Bericht sehr oft erzeugen muss
- Literatur lässt sich am Besten mit einem bibtex file einbauen

Jabref

- Literaturverwaltungssystem

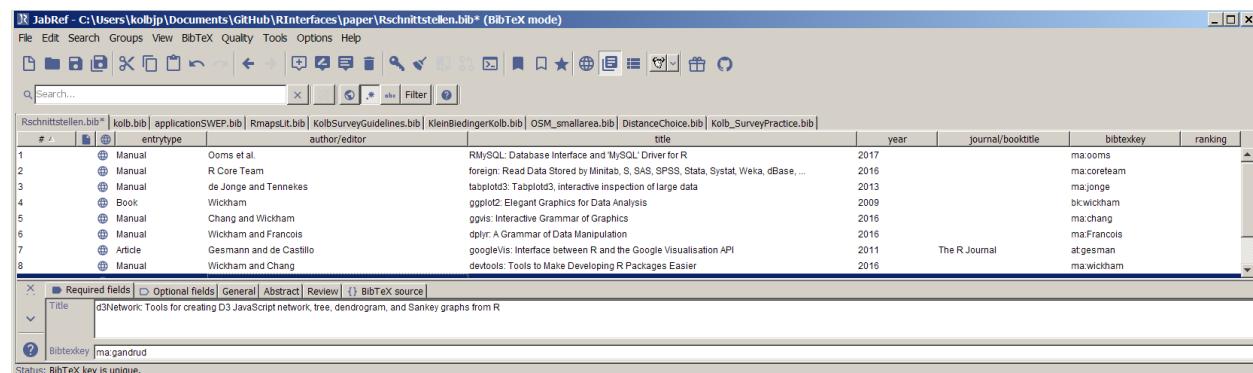


Figure 58:

Referenz mit R bekommen

- Mit dem Befehl `citation()` bekommt man sehr schnell die Referenz

```
install.packages("RMySQL")
```

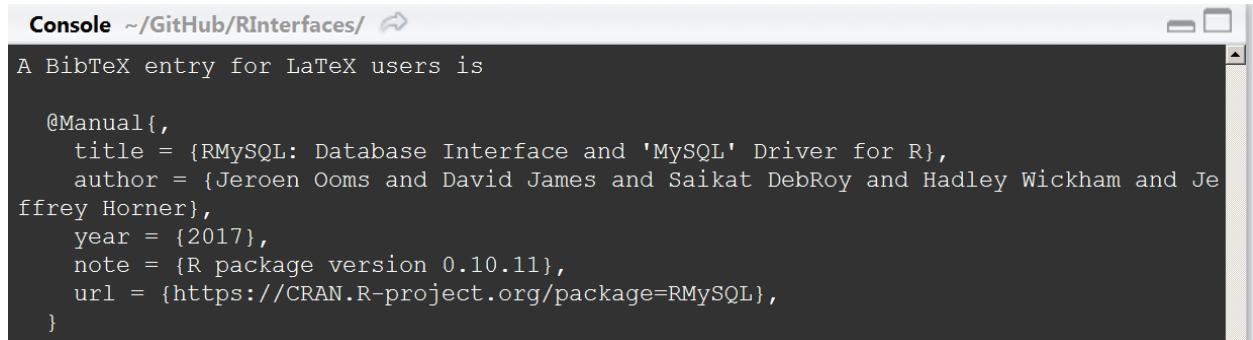
```
citation("RMySQL")
```

```
##  
## To cite package 'RMySQL' in publications use:  
##  
##   Jeroen Ooms, David James, Saikat DebRoy, Hadley Wickham and  
##   Jeffrey Horner (2017). RMySQL: Database Interface and 'MySQL'  
##   Driver for R. R package version 0.10.11.  
##   https://CRAN.R-project.org/package=RMySQL  
##  
## A BibTeX entry for LaTeX users is  
##
```

```

## @Manual{,
##   title = {RMySQL: Database Interface and 'MySQL' Driver for R},
##   author = {Jeroen Ooms and David James and Saikat DebRoy and Hadley Wickham and Jeffrey Horner},
##   year = {2017},
##   note = {R package version 0.10.11},
##   url = {https://CRAN.R-project.org/package=RMySQL},
## }

```



A BibTeX entry for LaTeX users is

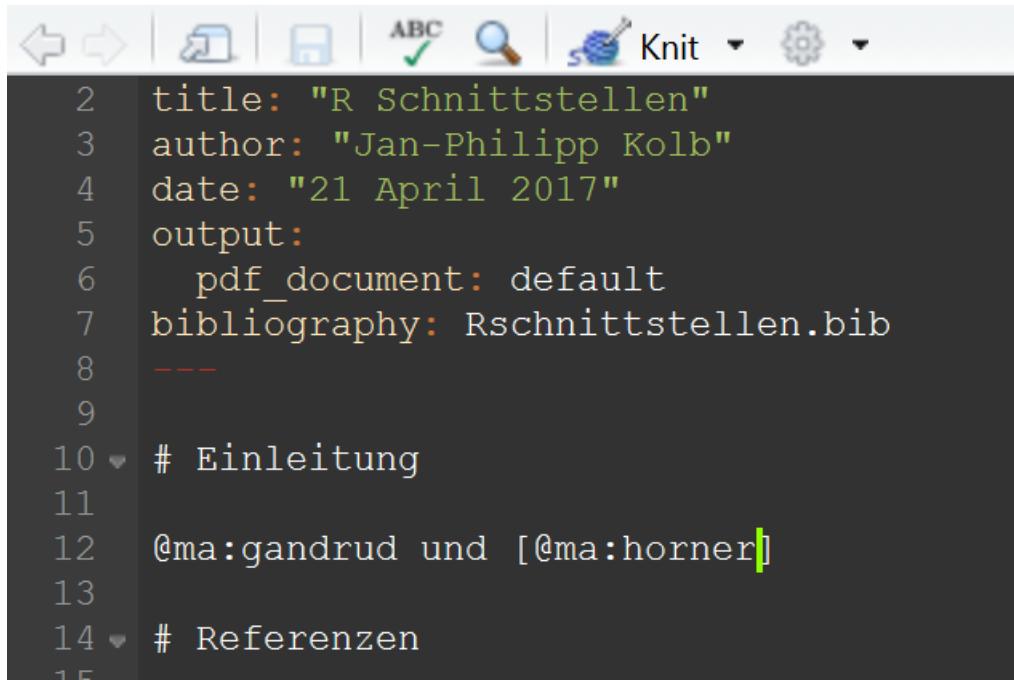
```

@Manual{,
  title = {RMySQL: Database Interface and 'MySQL' Driver for R},
  author = {Jeroen Ooms and David James and Saikat DebRoy and Hadley Wickham and Jeffrey Horner},
  year = {2017},
  note = {R package version 0.10.11},
  url = {https://CRAN.R-project.org/package=RMySQL},
}

```

Figure 59:

Das bibtex file einbinden I



The screenshot shows the RStudio interface with a LaTeX document open. The code editor contains the following LaTeX code:

```

2 title: "R Schnittstellen"
3 author: "Jan-Philipp Kolb"
4 date: "21 April 2017"
5 output:
6   pdf_document: default
7 bibliography: Rschnittstellen.bib
8 ---
9
10 # Einleitung
11
12 @ma:gandrud und [@ma:horner]
13
14 # Referenzen
15

```

Figure 60:

Das bibtex file einbinden II

```
title: "R Schnittstellen"
author: "Jan-Philipp Kolb"
date: "21 April 2017"
output:
  pdf_document: default
bibliography: Rschnittstellen.bib
---
```

Das Ergebnis

R Schnittstellen

Jan-Philipp Kolb
21 April 2017

Einleitung

Gandrud (2015) und (Horner 2014)

Referenzen

Gandrud, Christopher. 2015. *D3Network: Tools for Creating D3 Javascript Network, Tree, Dendrogram, and Sankey Graphs from R*. <https://CRAN.R-project.org/package=d3Network>.

Horner, Jeffrey. 2014. *Rook: Rook - a Web Server Interface for R*. <https://CRAN.R-project.org/package=Rook>.

Figure 61:

Links

- Optionen für Beamer Präsentationen
- Wie R und LaTeX zusammen funktionieren

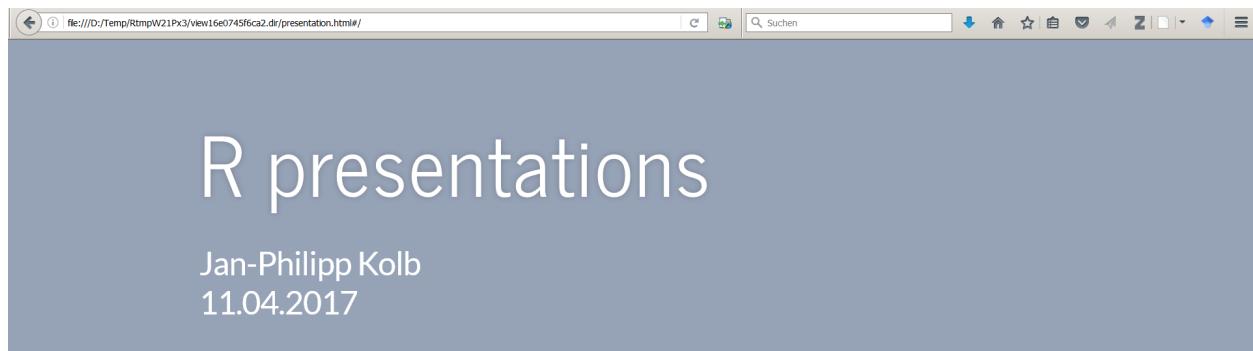


Figure 62:

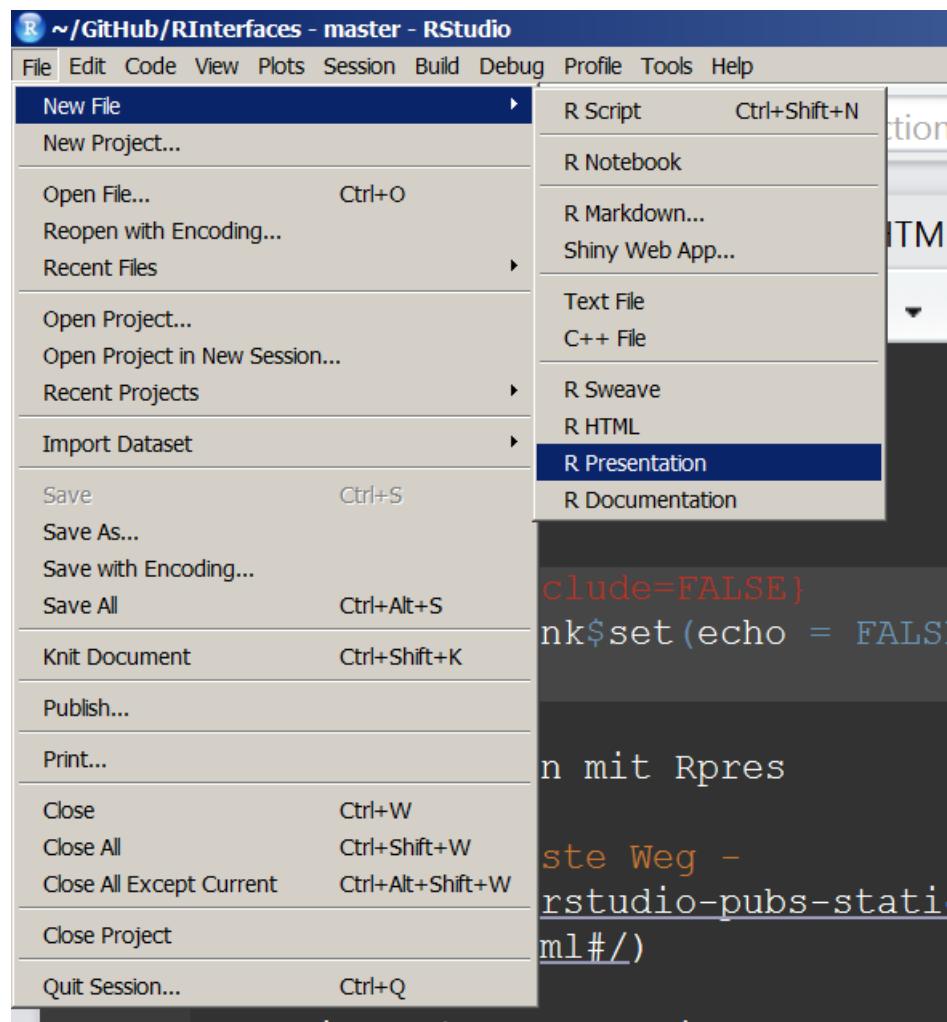


Figure 63:

HTML Dokumente, Präsentationen und Dashboards mit Rmark-down

Präsentationen - Rpres der einfachste Weg

Eine erste Präsentation

Erste Daten eintragen

- Für Vergessliche:

```
date()
```

```
## [1] "Wed May 10 12:07:13 2017"
```

Eine Folie mit Formel

- Die Formel kann wie in LaTeX eingegeben werden

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

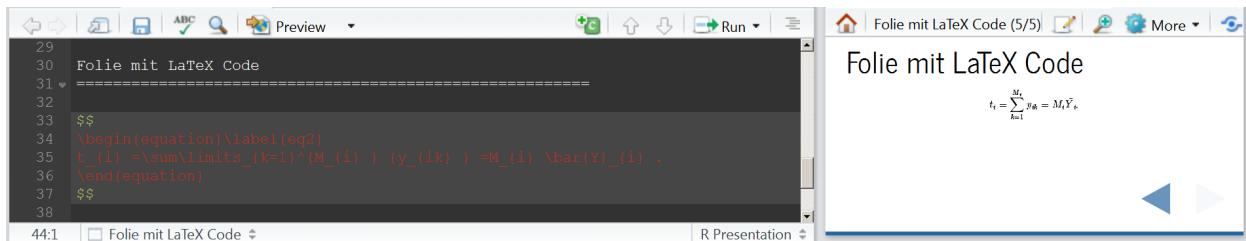


Figure 64:

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|
7
```

Figure 65:

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'
```



Figure 66:

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>

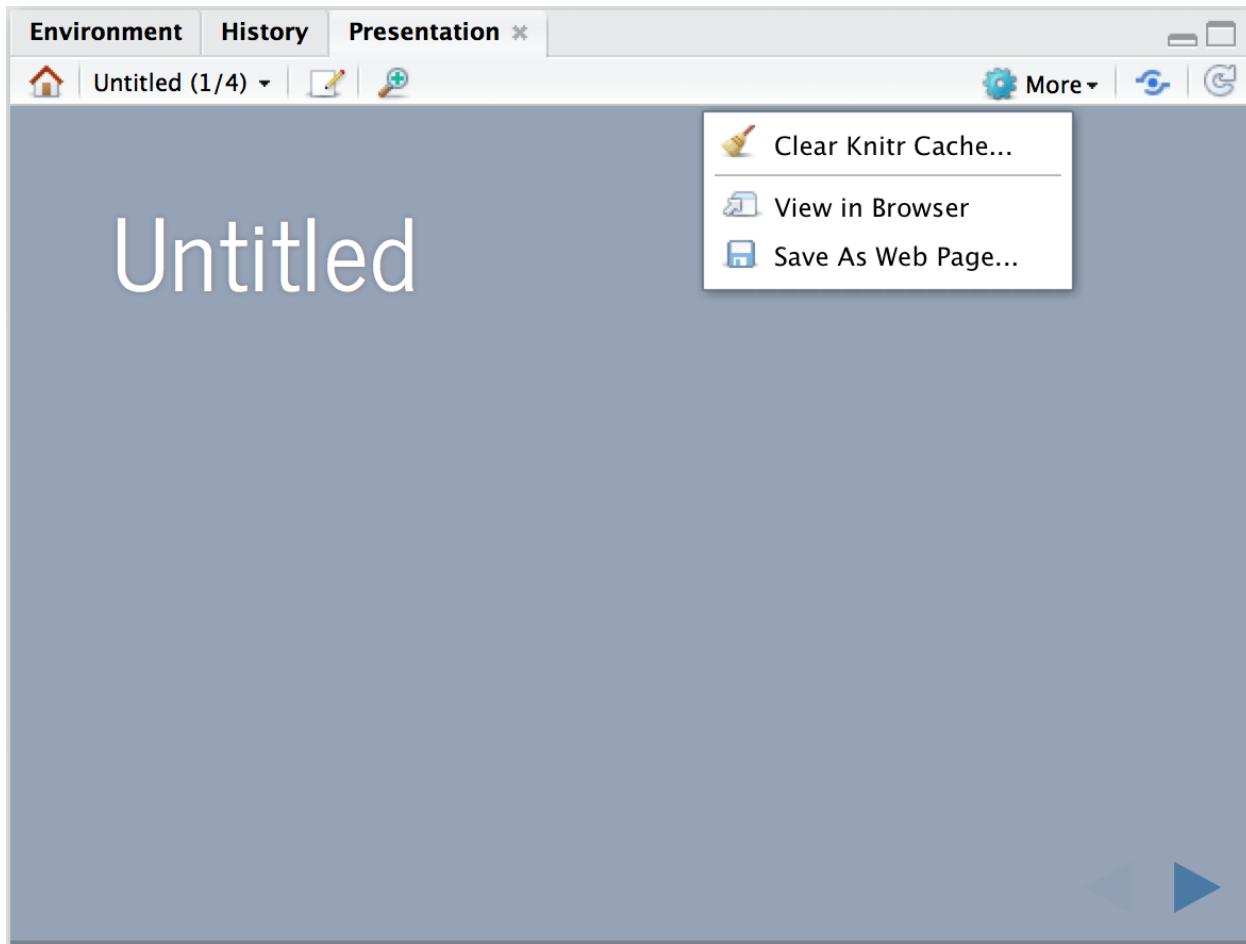


Figure 67:



Figure 68:

Eine ioslides Präsentation

Eine ioslides Präsentation

ioslides - Der Start

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
---
```

Tabellen

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

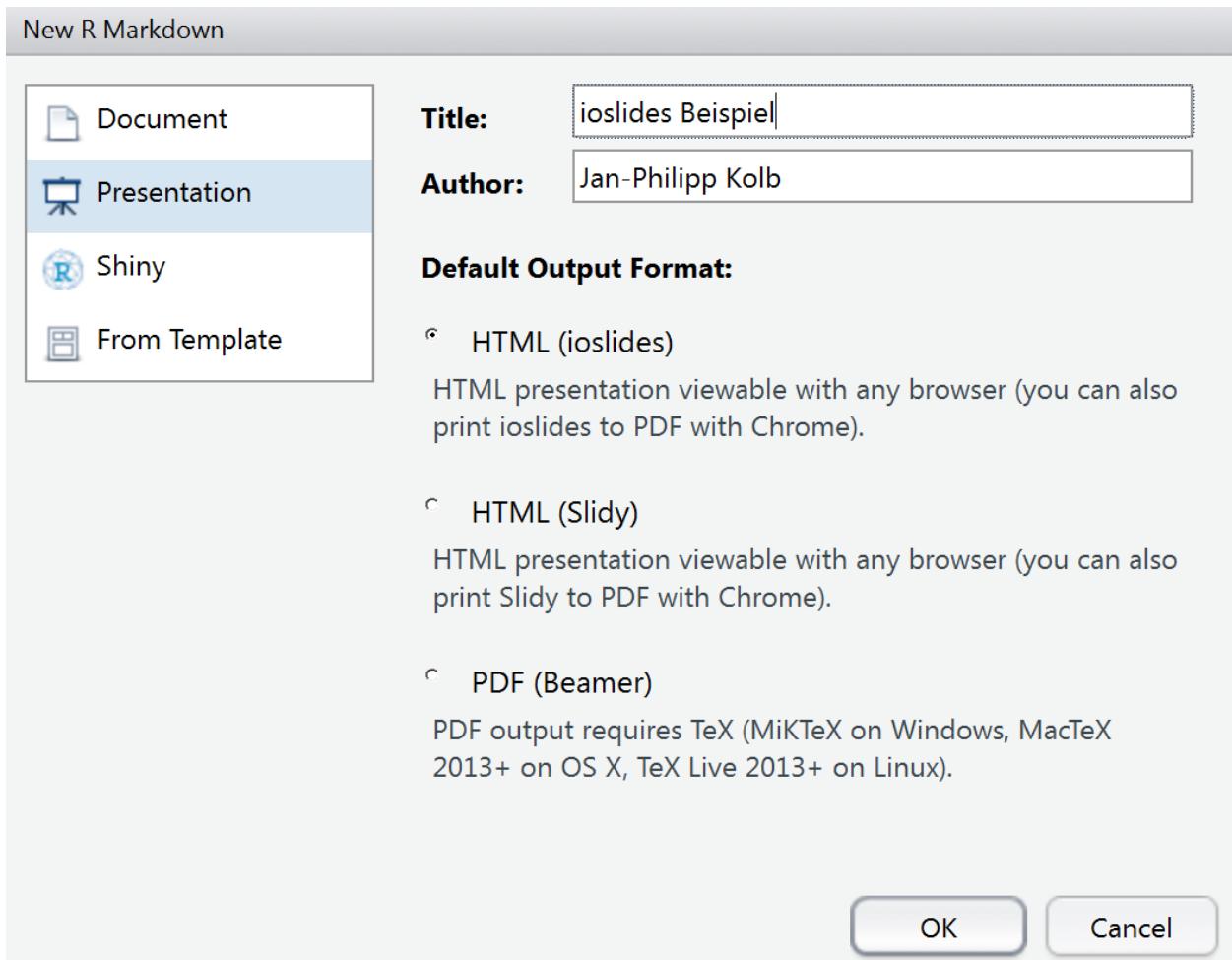


Figure 69:

```
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  
8 ---
```

Figure 70:

```
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 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

knitr Engines

- knitr Language Engines
- slidify

Eine slidify Präsentation

slidify Präsentationen

[Was sind Cascading Style Files (CSS)]

- 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

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

Präsentationen mit R und Rstudio

Jan-Philipp Kolb

11 April 2017

Contents

slide 1/3

Figure 71:



Figure 72:

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

Figure 73:

- Daneben gibt es viele weitere Dinge, die sich mit dem CSS steuern lassen

HTML Dokumente

Ein HTML Dokument erzeugen

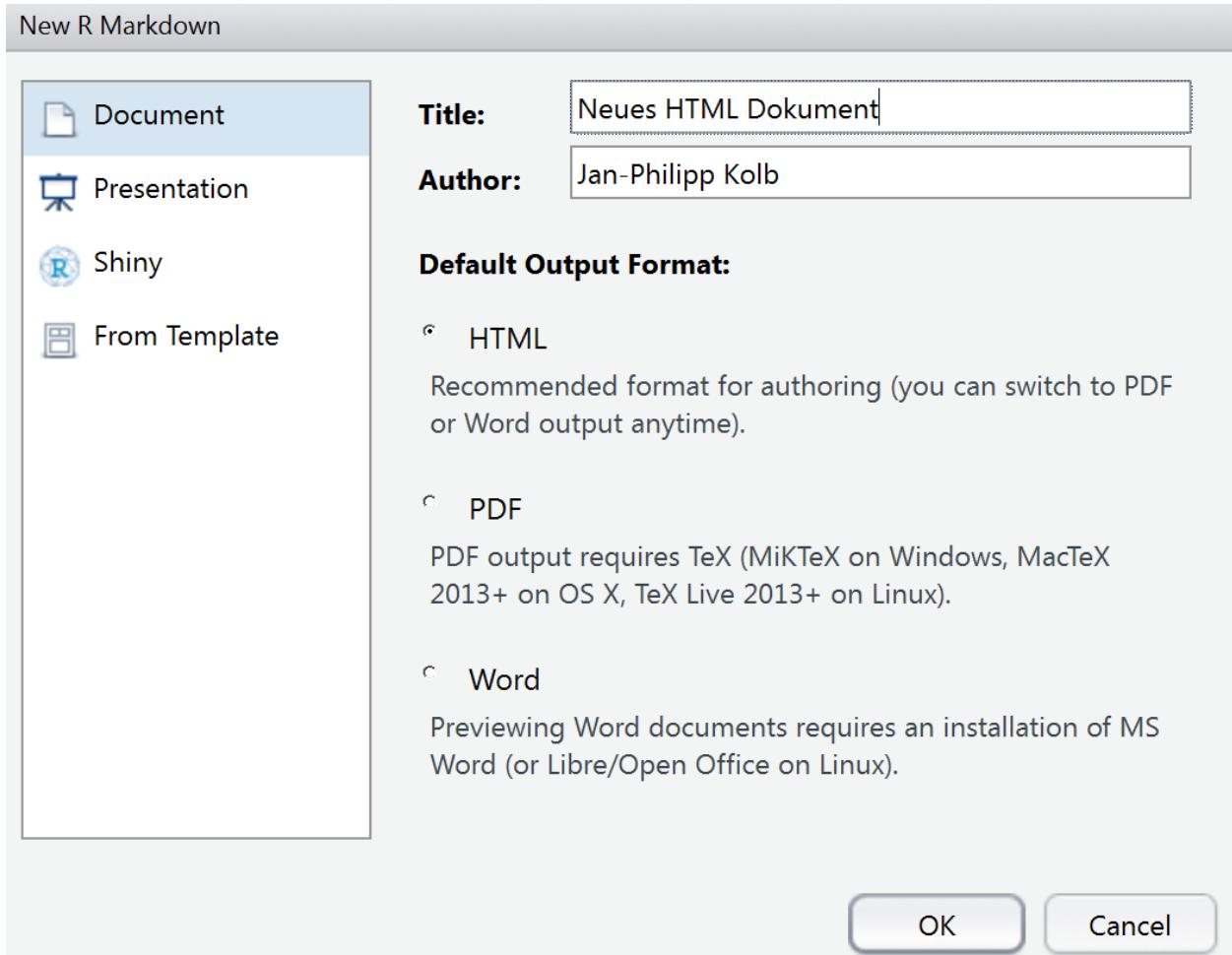


Figure 74:

Ein Template verwenden

Weitere Vorlagen nutzen

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

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

Vorlagen für Markdown

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

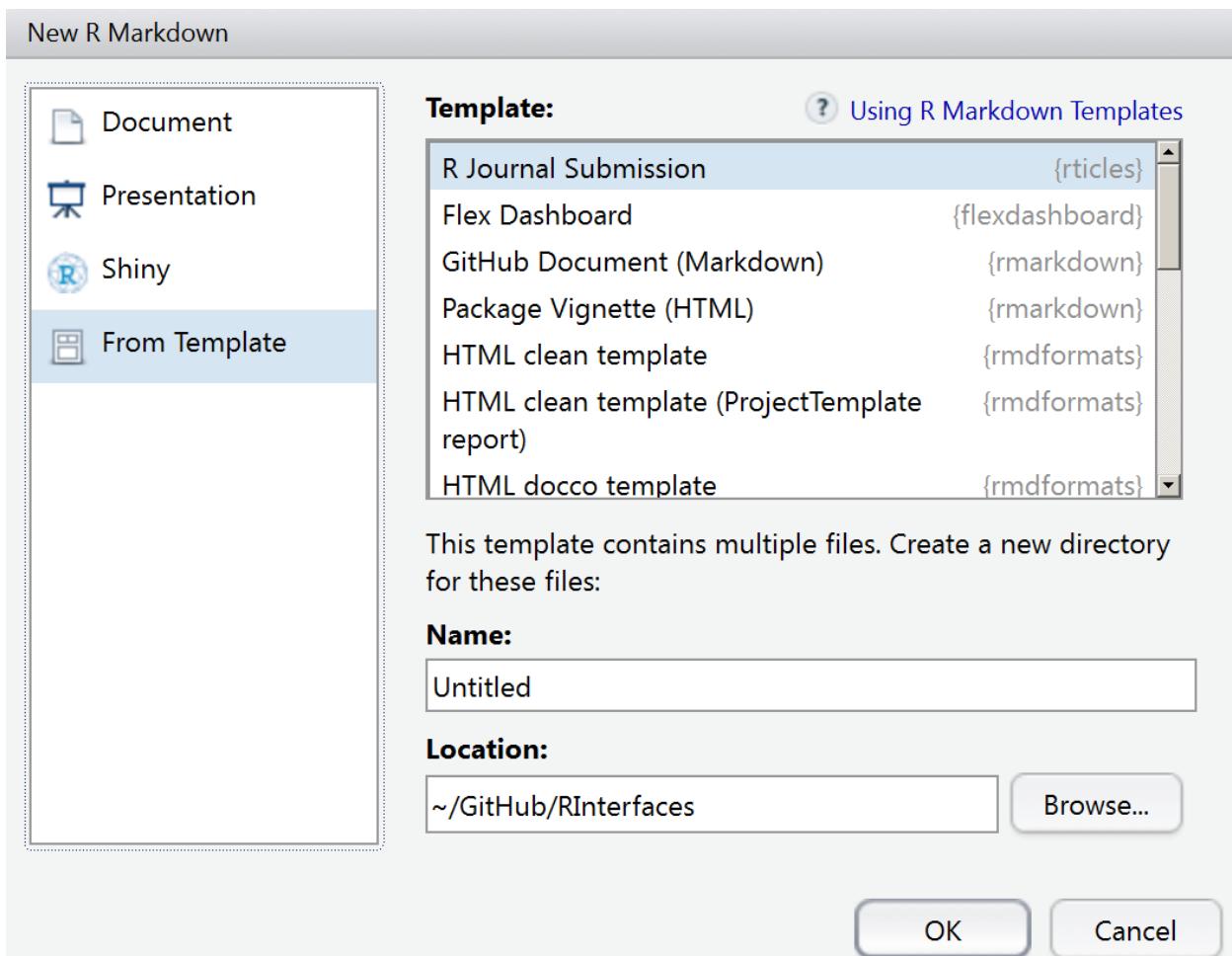


Figure 75:

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 76:

```
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

The screenshot shows a dark-themed dashboard titled "readthedown template example". On the left, there is a sidebar with a red header containing the title. Below the title, the sidebar lists "Code and Tables", "Figures", and "MathJax". The main content area has a white background and features several sections:

- Code and tables**: A section showing R code for generating summary statistics for the Iris dataset.
- Syntax highlighting**: A note stating "Here is a sample code chunk, just to show that syntax highlighting works as expected."
- Verbatim**: A section showing the structure of the `iris` dataset using the `str()` function.

The R code in the "Code and tables" section is as follows:

```
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))
```

The output of the `str(iris)` command in the "Verbatim" section is as follows:

```
'data.frame': 150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5.5 4.6 5.4 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 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 1.5 ...
 $ Petal.Width : num  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",..: 1 1 1 1 1 1 1 1 1 1 ...
```

Figure 77:

Dashboards

Beispiel R-Pakete

Paket installieren

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

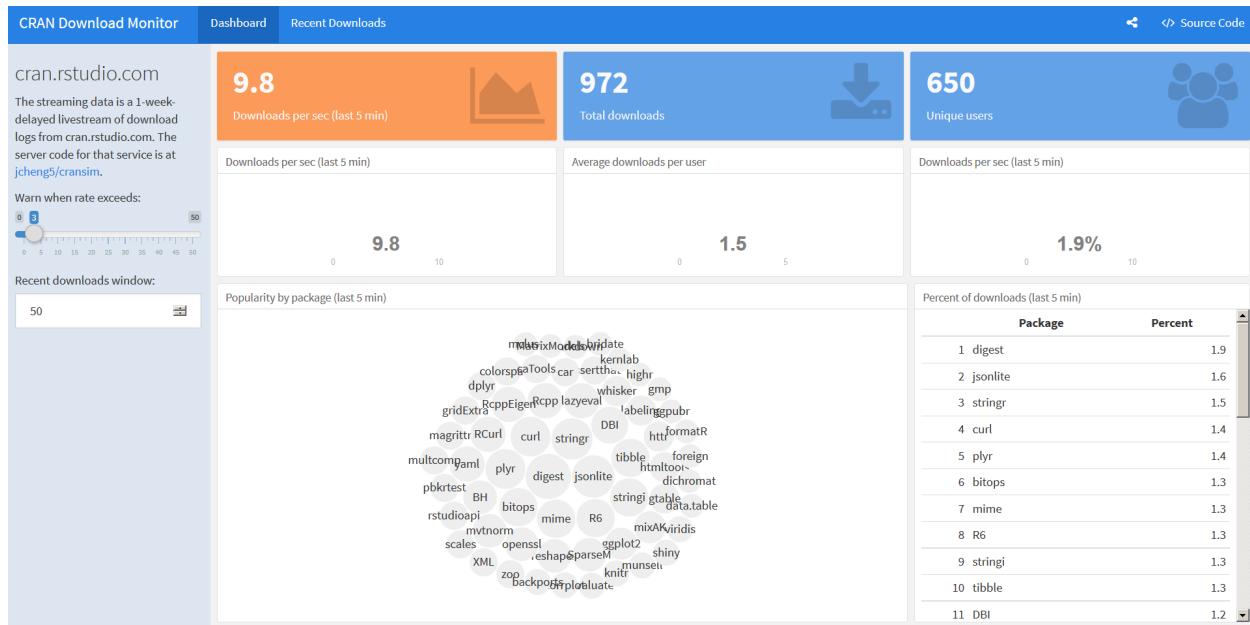


Figure 78:

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.

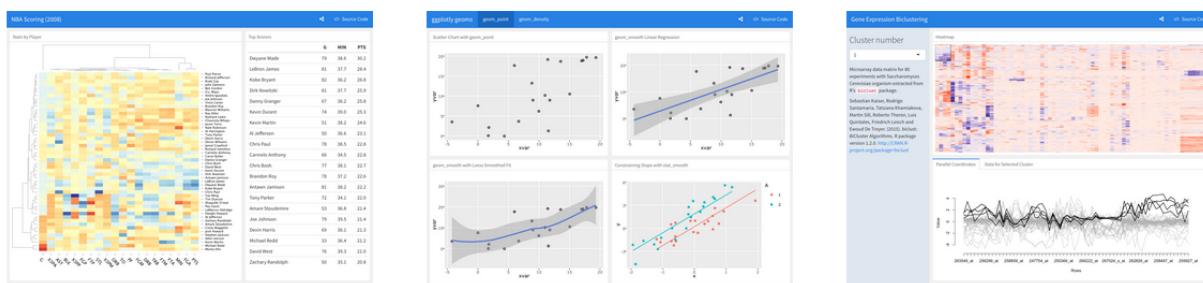


Figure 79:

Ein Dashboard erstellen mit Rstudio

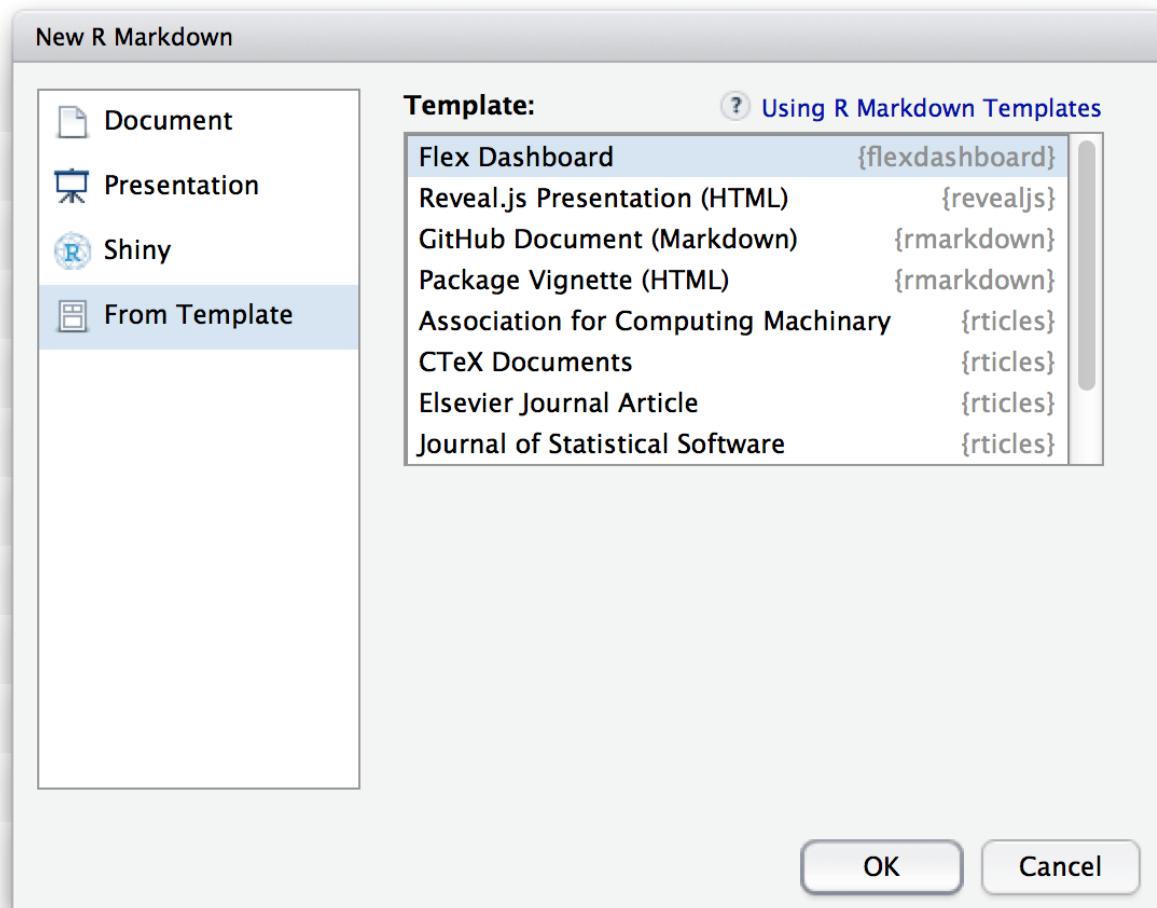


Figure 80:

Mein erstes Dashboard

Gallerie

Links

- Verschiedene Markdown Dokumente zusammen fügen
- Verschiedene CSS Fonts
- Überblick über die verschiedenen Rmarkdown Formate

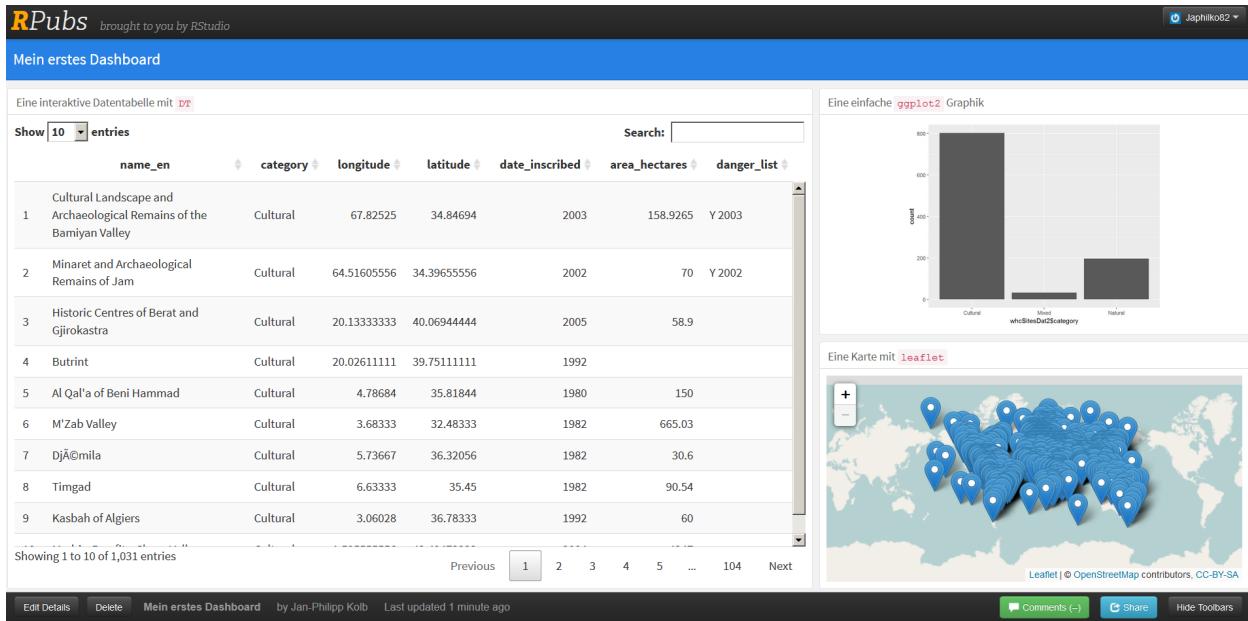


Figure 81:

Documents

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

The screenshot displays five examples of R Markdown documents:

- Great NYT Interactive**: An interactive visualization showing a map of the United States with data points. It includes a "Disclaimer and Attribution" section.
- Pandoc Markdown Article Starter and Template***: A template for a journal article, featuring a header, abstract, introduction, and sections for methods, results, and discussion.
- A Microsoft Word document**: A template for a Microsoft Word document, showing how R code can be embedded directly into the document.
- Tufte Handout**: A template for a Tufte-style handout, characterized by its clean, minimalist design and tight integration of graphics with text.
- Handouts**: A template for handouts, showing how R code and plots can be included in a presentation slide.

Interactive Documents

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

The screenshot shows four examples of interactive documents:

- HTML Widgets**: A map of the United States with blue location pins, demonstrating the use of the leaflet package.
- HTML Widgets**: A Shiny application titled "UNCCD Data Report" showing a scatter plot of data over time.
- Shiny**: A Shiny application titled "Shiny leaflet example" showing a map with data points and a sidebar with R code.
- Shiny**: A Shiny application titled "Shiny" showing a scatter plot of movie ratings over time and a line graph of movie counts over years.

Figure 82:

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 83:

Notebooks zur Integration von anderen Programmiersprachen (Python,LaTeX,Julia)

Notebooks

- Warum R Notebook nutzen

Rnotebooks

Ein Rnotebook anlegen

Rnotebook - erste Schritte

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

Python Code integrieren

- Ebenso lässt sich Python code implementieren

```
import sys
```

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

```
## 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)]
```

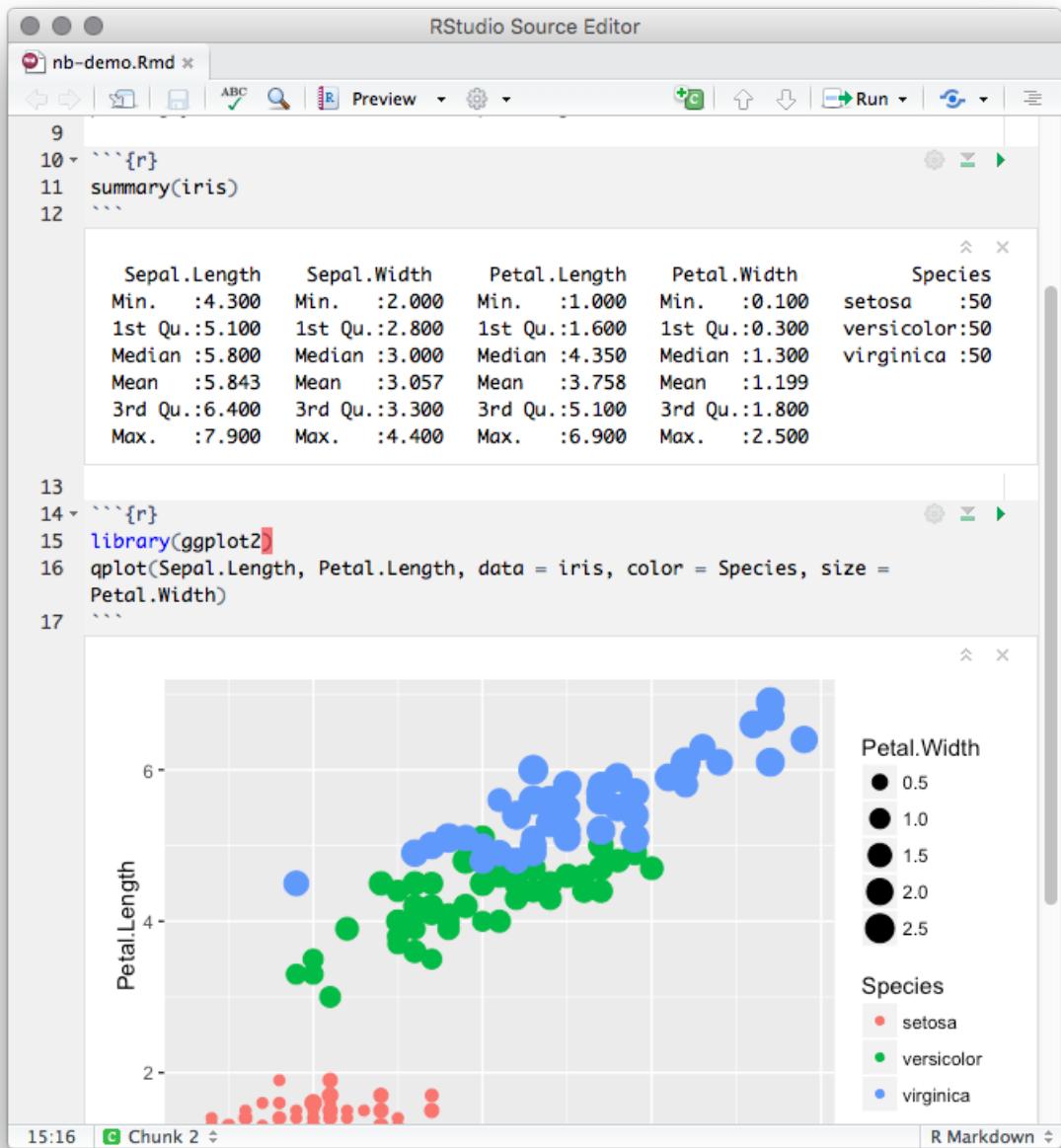


Figure 84:

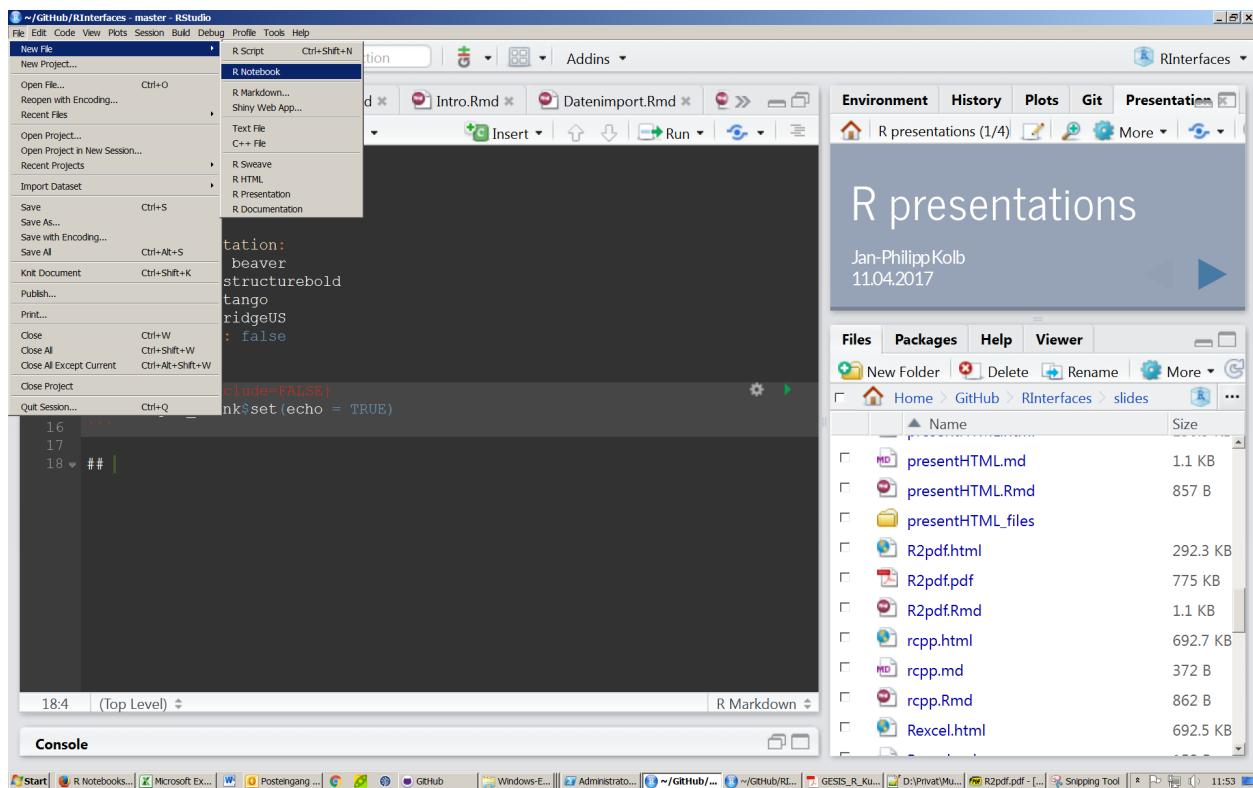


Figure 85:

The screenshot shows the RStudio interface with the code editor containing R code. The code includes a preview note and generates a sequence of numbers:

```
*Preview* button or press *ctrl+shift+k* to preview the HTML file).
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
```

The status bar at the bottom left shows '23' and '24'. The code editor has line numbers 17 through 24.

Figure 86:

The screenshot shows the RStudio interface with a code editor containing a multi-language block. It starts with R code and then switches to Python:

```
20 ````{python}
21 print("Hello World")
22
```

Below the code editor is a text box displaying the output 'Hello World'.

Figure 87:

LaTeX Code integrieren

- LaTeX code wird mit zwei Dollarzeichen gekennzeichnet

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

Figure 88:

Notebook veröffentlichen I

Notebook veröffentlichen II

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

Beispiel Eingabe Code

Beaker Notebook

Beaker Notebook

- Auch bei Beaker kann man R-code einbauen

Beaker starten

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

Links

- knitr Language Engines

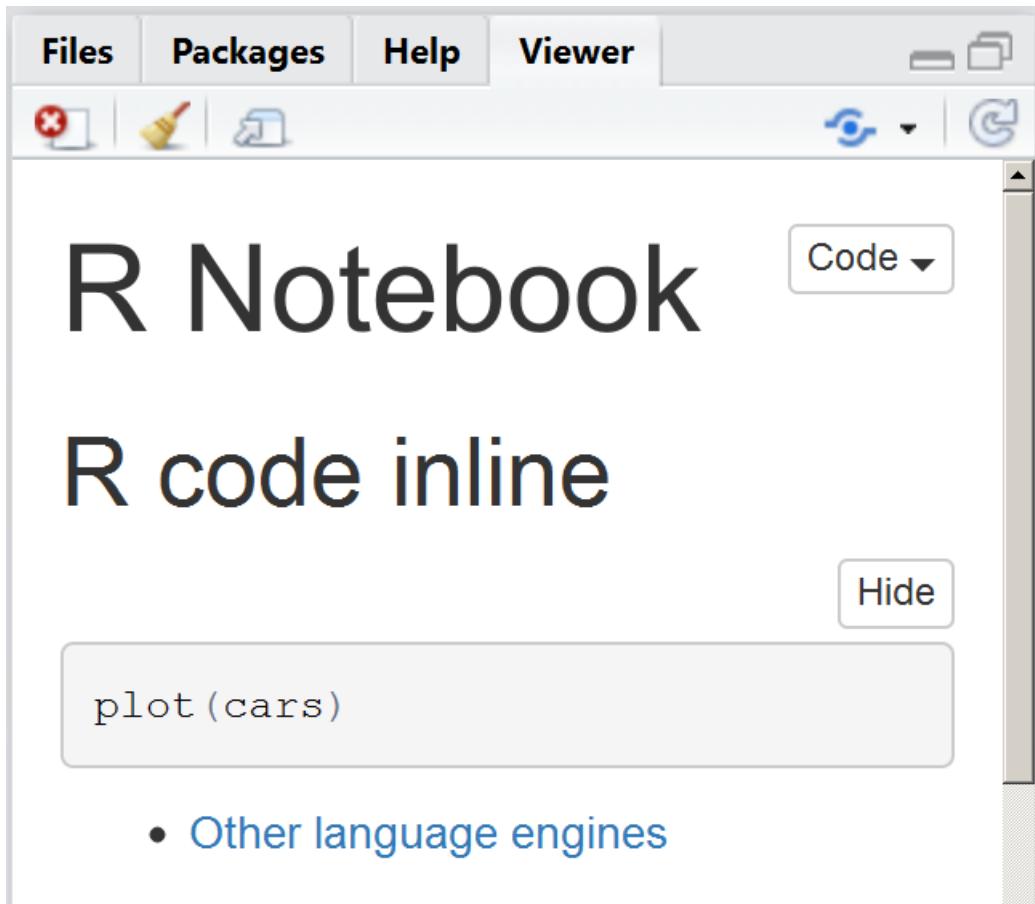


Figure 89:

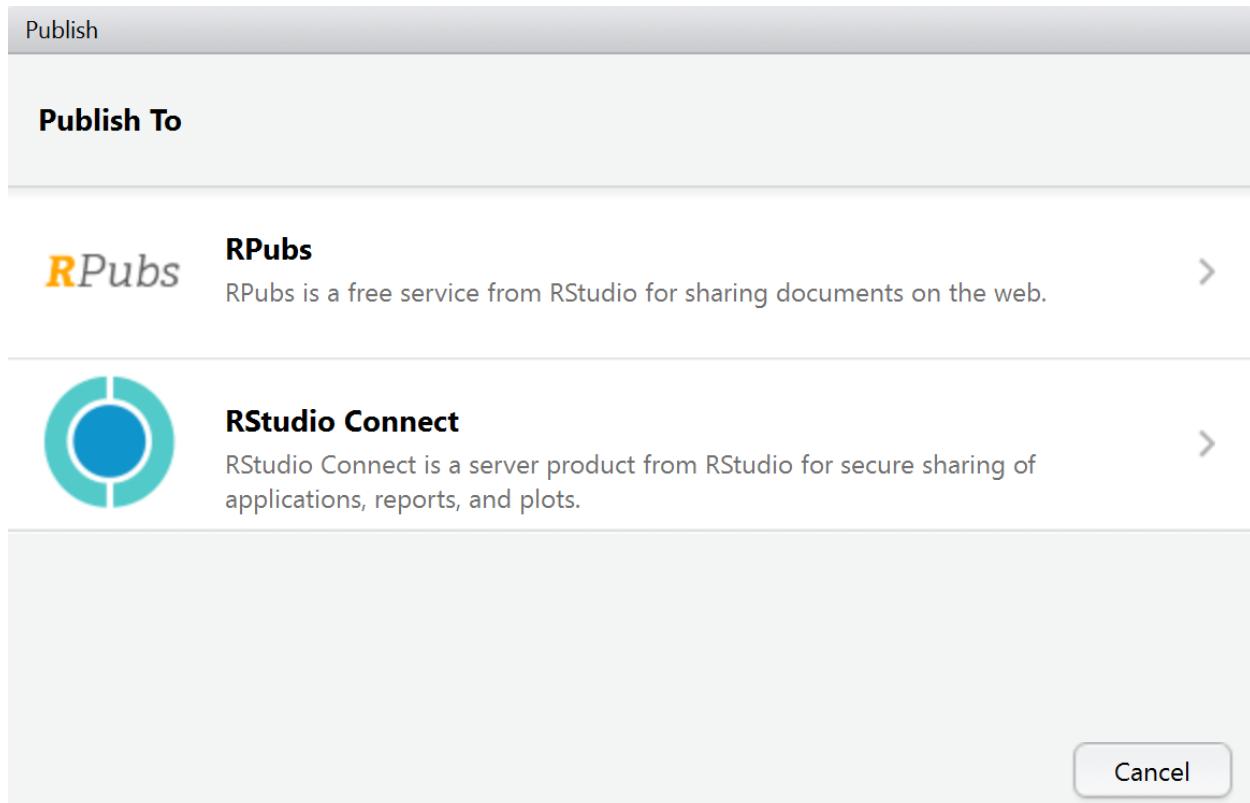


Figure 90:



Figure 91:

The screenshot shows a Jupyter R notebook interface. At the top, there's a toolbar with various icons for file operations like saving, opening, and running code. Below the toolbar, the menu bar includes File, Edit, View, Insert, Cell, Kernel, and Help. A status bar at the bottom right indicates 'R'.

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&maptyle=terrain&language=en-EN&sensor=false>
Information from URL : <http://maps.googleapis.com/maps/api/geocode/json?address=Mannheim&sensor=false>

Figure 92:



Figure 93:

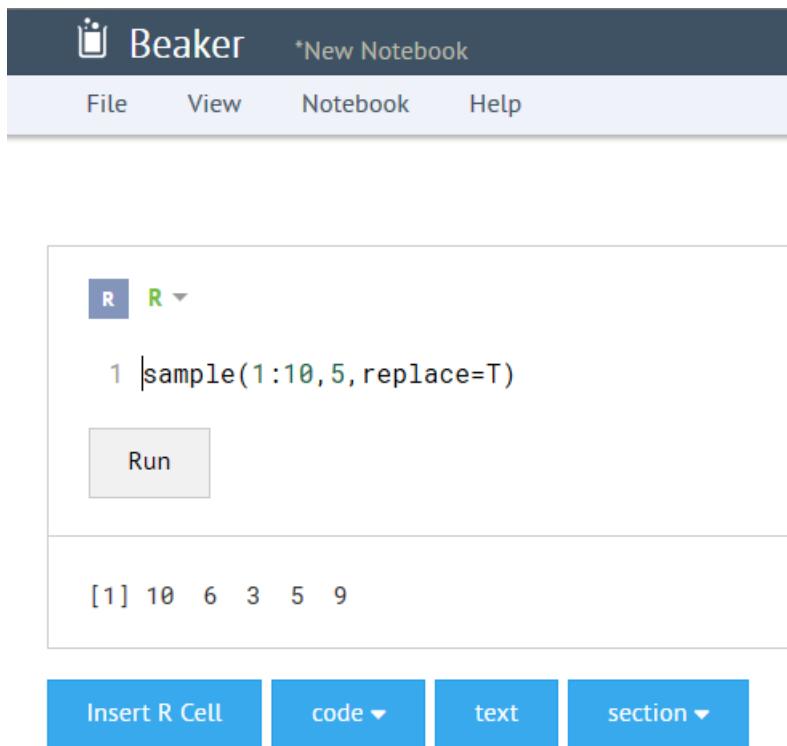


Figure 94:

- 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

Aufgabe: Bearbeiten Sie ein Notebook weiter

- Suchen Sie folgendes Notebook auf meinem Github Ordner:
- Lassen Sie sich die Raw Version anzeigen

Figure 95:



Figure 96:

- Laden Sie das Notebook herunter:

The screenshot shows a browser window with the URL https://raw.githubusercontent.com/Japhilko/RInterfaces/master/misc/notebooks/Geokodierung_ggmap.Rmd. The page content is an R notebook with the following code:

```
---
title: "Geokodierung mit R und der Google API"
output: html_notebook
---

## Das notwendige Paket `ggmap` laden

```{r, eval=F}
install.packages("ggmap")
```

```
library(ggmap)
```

The context menu is open over the page, with the option "Seite speichern unter..." highlighted.
```

Figure 97:

- Und bearbeiten Sie es so weiter, dass sie die Geokodierung für eine Adresse bekommen, die für Sie interessant ist (Hbf, Hotel, Restaurant etc.)

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

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 98:

Weitere Variablen WHC Datensatz

```
whcSitesDat2 <- with(whcSites,data.frame(name_en,category,
                                             longitude,latitude,date_inscribed,area_hectares,danger_list))
```

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

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

Das Ergebnis bei Rpubs

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

Das Paket magrittr

- `magrittr` - für den Pipe Operator in R:

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

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 99:

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

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



Tal Galili

August 5, 2014

Guest Post, R, R
programming



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

Figure 100:

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)
```

Die Karte zeigen

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)
```



Figure 101:

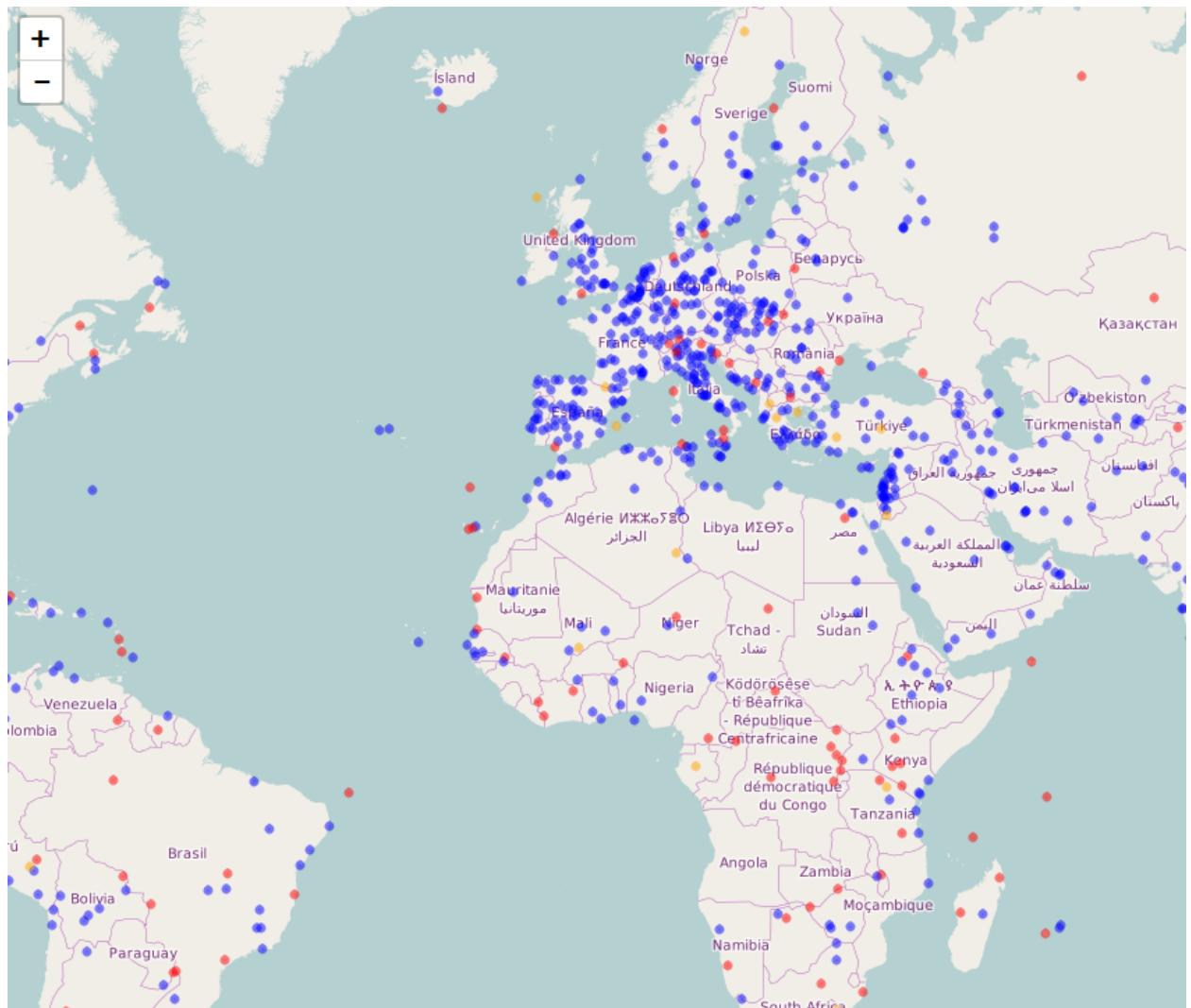


Figure 102: Weltkulturerbe

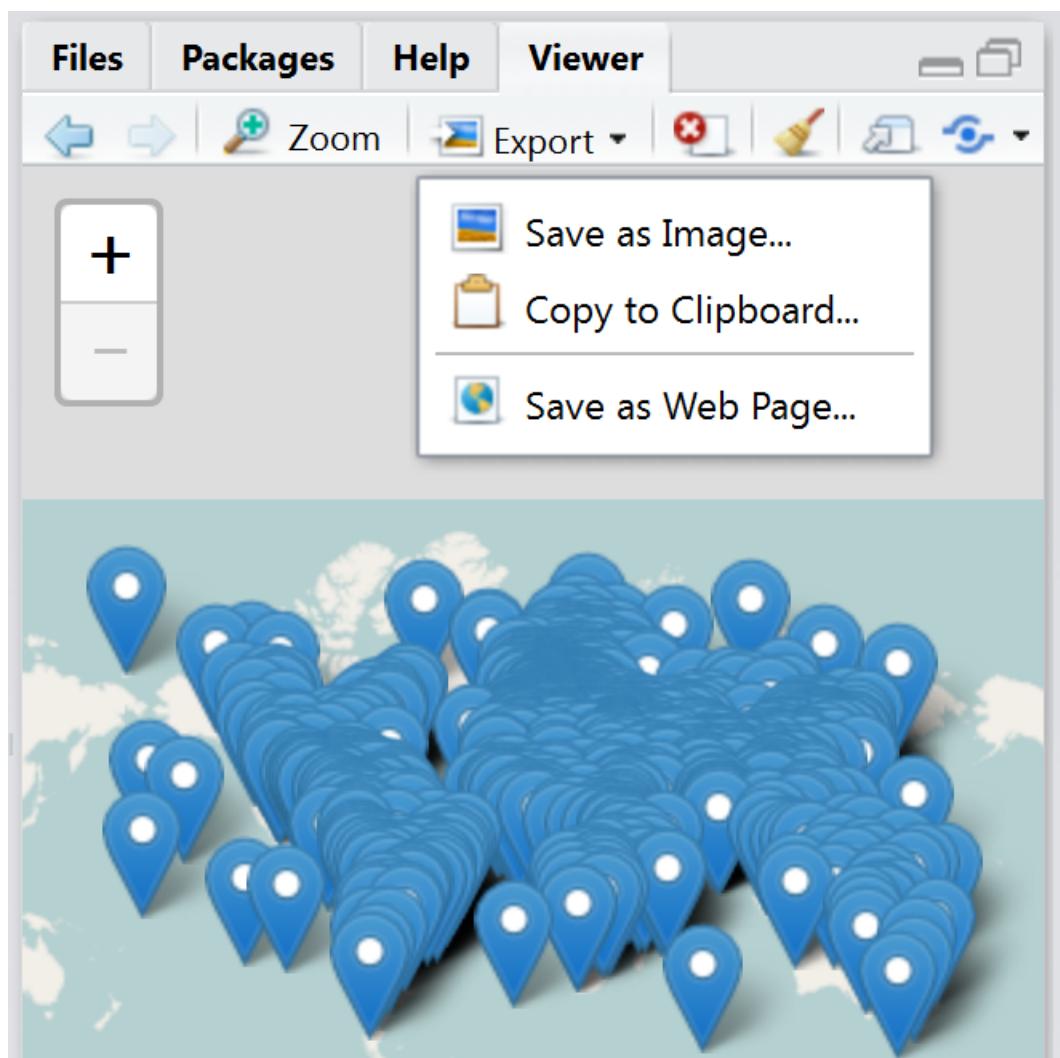


Figure 103:

Die Karte mit mehr Farbe

Die Karte abspeichern

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
```

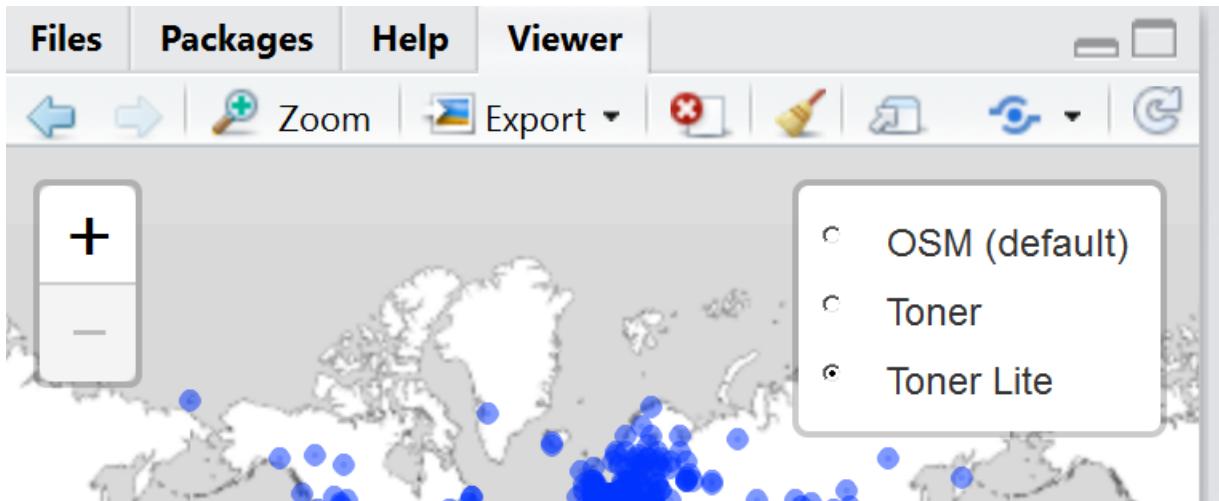


Figure 104:

Ein weiteres Beispiel mit Erdbebendaten

```
outline <- quakes[chull(quakes$long, quakes$lat),]

map <- leaflet(quakes) %>%
  # Base groups
  addTiles(group = "OSM (default)") %>%
  addProviderTiles("Stamen.Toner", group = "Toner") %>%
  addProviderTiles("Stamen.TonerLite", group = "Toner Lite") %>%
  # Overlay groups
  addCircles(~long, ~lat, ~10^mag/5, stroke = F, group = "Quakes") %>%
  addPolygons(data = outline, lng = ~long, lat = ~lat,
```

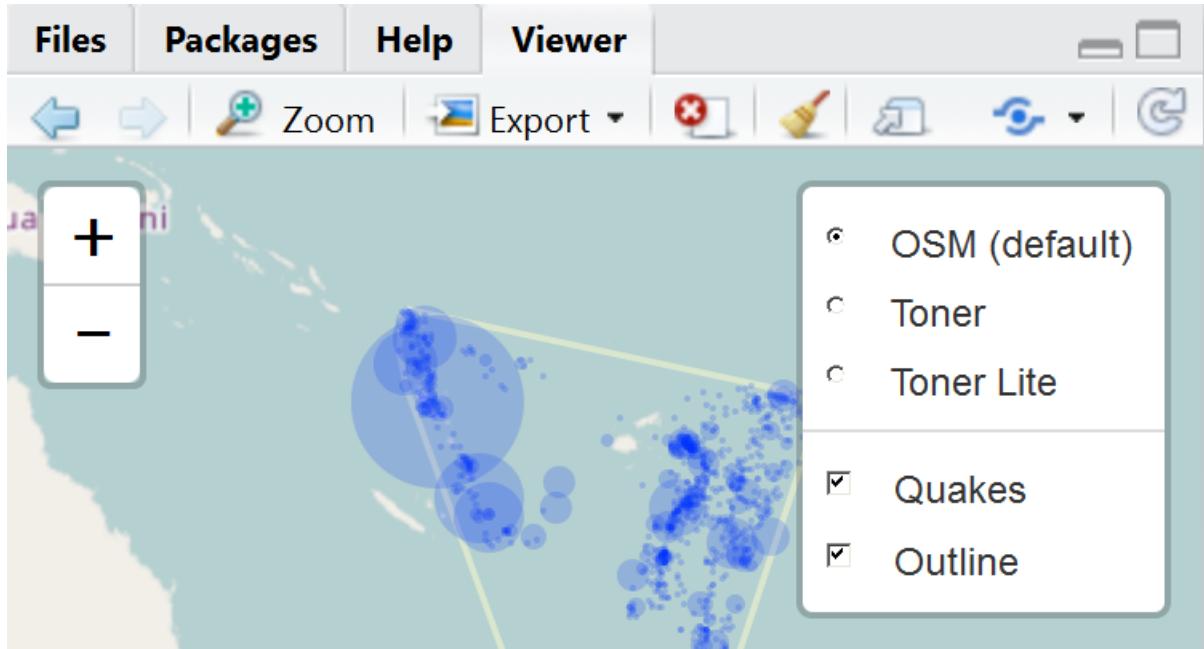


Figure 105:

```

    fill = F, weight = 2, color = "#FFFFCC", group = "Outline") %>%
# Layers control
addLayersControl(
  baseGroups = c("OSM (default)", "Toner", "Toner Lite"),
  overlayGroups = c("Quakes", "Outline"),
  options = layersControlOptions(collapsed = FALSE)
)
map

```

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), stroke = FALSE)
```

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 = 12)
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.cgi",
    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-green.png",
  iconWidth = 38, iconHeight = 95,
  iconAnchorX = 22, iconAnchorY = 94,
  shadowUrl = "http://leafletjs.com/examples/custom-icons/leaf-shadow.png",
  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/707b339dc88f57bbd5d88377891131e3_beans-people-clipart-c
  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/>

Interaktive Tabellen mit DataTables

The R-package DT

- DT: An R interface to the DataTables library

```
install.packages('DT')

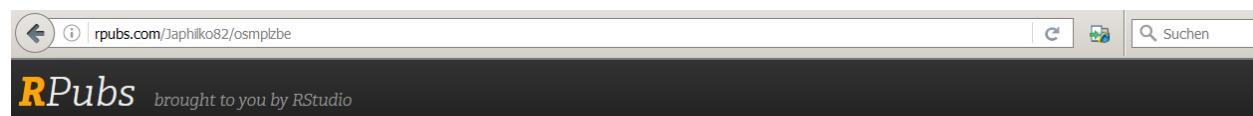
library('DT')

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 interactive DataTables table on RPubs. The table has columns for PLZ99, PLZORT99, area_d, bakery, bar, cafe, and clothes. The first three rows of data are shown:

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

Figure 106:

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')
```

| 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

Figure 107:

R und die Javascript Data-Driven Documents (D3)

JavaScript - Data-Driven Documents

- D3 ist eine der mächtigsten unter der Vielzahl aktuell verfügbarer JavaScript-Bibliotheken zur Datenvisualisierung.
- Big Data im Netz visualisieren
- Möglichkeiten mit der JS D3 Bibliothek
- Die Power von R und D3.js zusammen bringen
- 5 Interaktive R Visualisierungen mit D3, ggplot2 und RStudio
- Das D3 Netzwerk

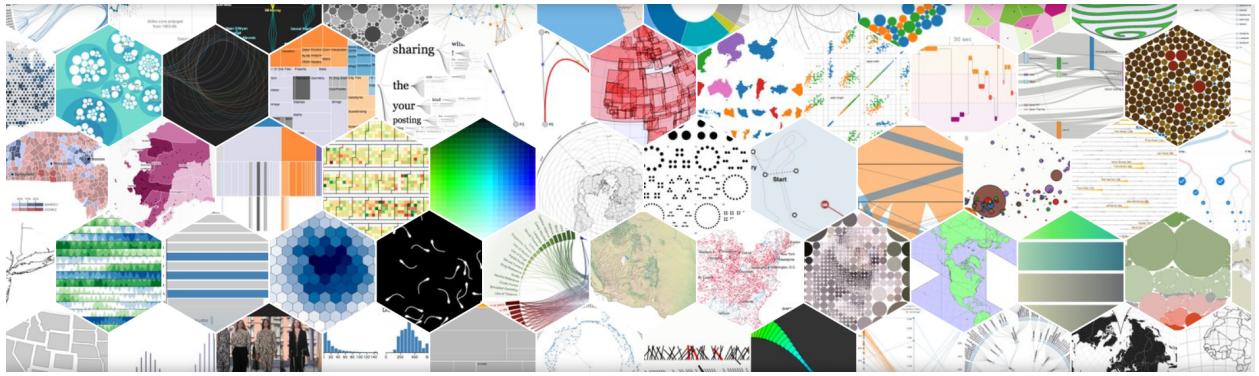


Figure 108:

gigvis

```
install.packages("ggvis")
library(ggvis)
library(dplyr)
```

- Übersicht zu ggvis

Examples of ggvis graphics

Histogram:

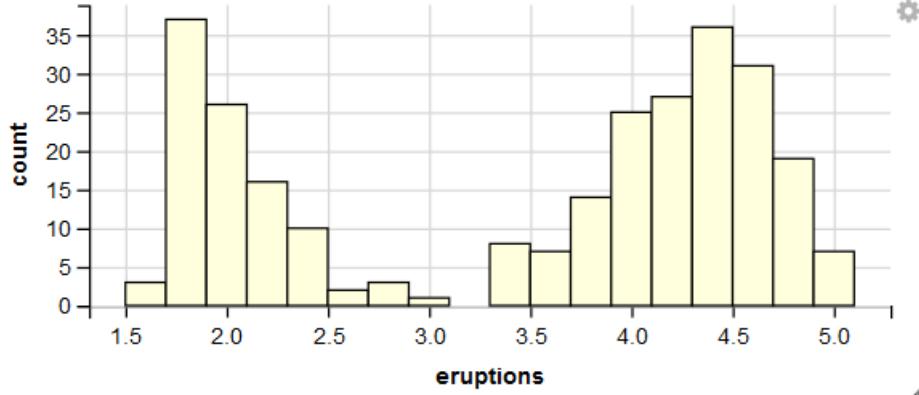


Figure 109:

Kochbuch für ggvis

```
mtcars %>% ggvis(~wt, ~mpg) %>% layer_points()
```

Plots mit Gruppierung

```
mtcars %>%
  ggvis(~wt, ~mpg, fill = ~factor(cyl)) %>%
  layer_points() %>%
  group_by(cyl) %>%
  layer_model_predictions(model = "lm")
```

Interaktive Graphiken mit `ggvis`

```
mtcars %>%
  ggvis(~wt, ~mpg) %>%
  layer_smooths(span = input_slider(0.5, 1, value = 1)) %>%
  layer_points(size := input_slider(100, 1000, value = 100))
```

- `ggvis` - Interaktivität

googleVis

```
install.packages("googleVis")
library(googleVis)
```

- Das Paket `googleVis` bietet eine Schnittstelle zur Google Charts API
- Einführung in `googleVis`
- Tutorial für `googleVis`
- Inspiration durch Hans Rosling: No more boring data

Ein Datensatz mit Früchten

```
library(DT)
datatable(Fruits)
```

Beispiel mit `googleVis`

```
plot(gvisMotionChart(Fruits, "Fruit", "Year", options = list(width = 600, height = 400)))
```

Ein weiterer Beispieldatensatz

```
df <- data.frame(year=1:11, x=1:11,
                  x.scope=c(rep(TRUE, 8), rep(FALSE, 3)),
                  y=11:1, y.html.tooltip=LETTERS[11:1],
                  y.certainty=c(rep(TRUE, 5), rep(FALSE, 6)),
                  y.emphasis=c(rep(FALSE, 4), rep(TRUE, 7)))
```

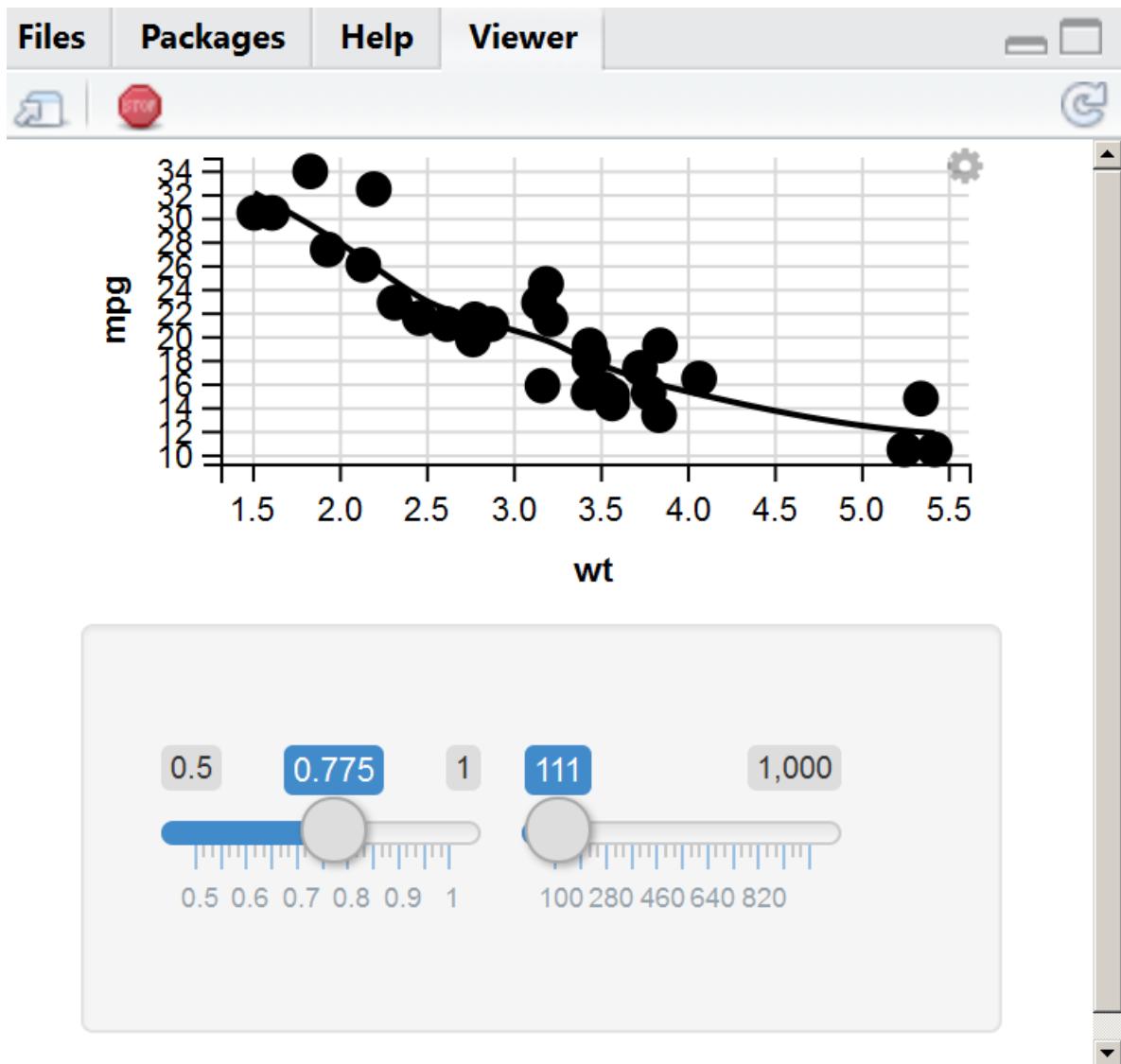


Figure 110:

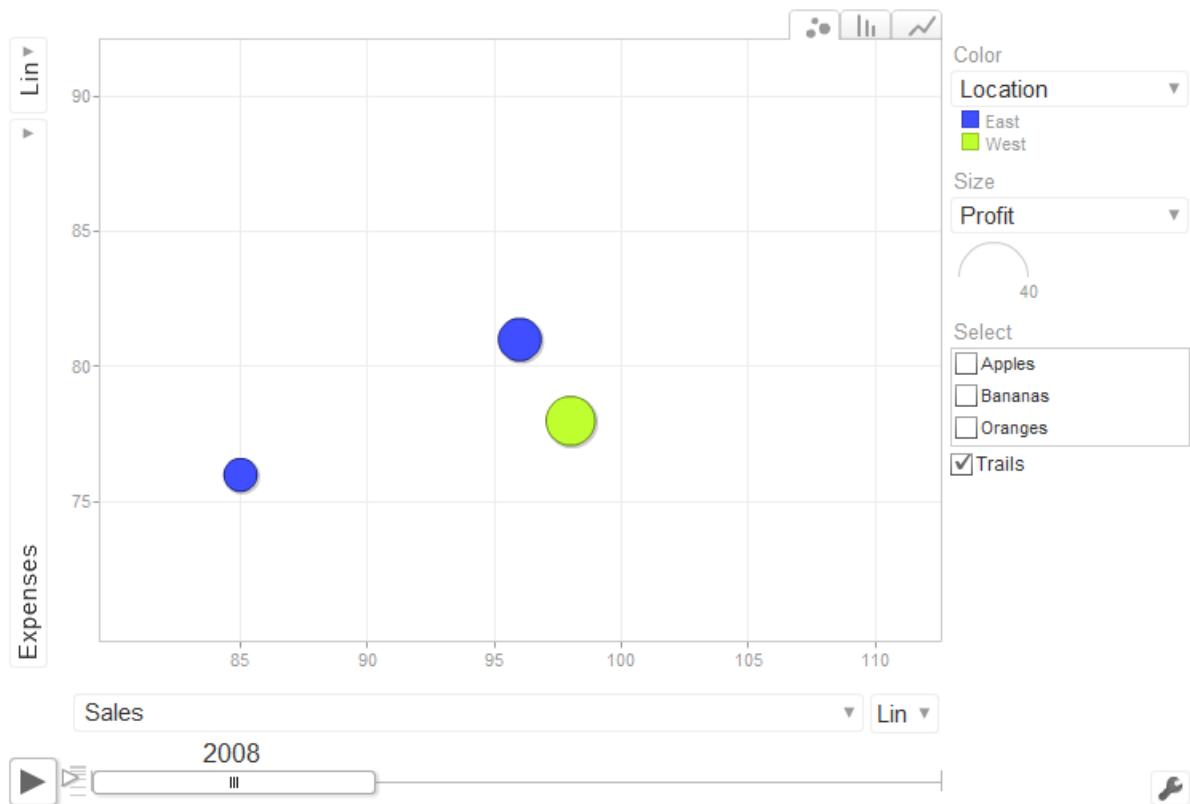


Figure 111:

Ein weiteres Beispiel für googleVis

```
plot(gvisScatterChart(df, options=list(lineWidth=2)))
```

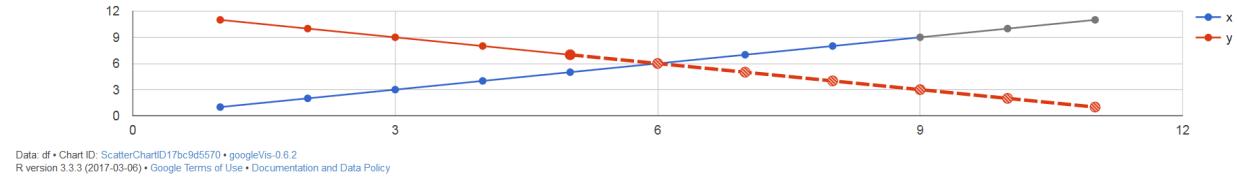


Figure 112:

Click me

```
install.packages("devtools")
library(devtools)

install_github("clickme", "nachocab")
```

Straight from R to JS: Create interactive visualizations from R



Figure 113:

Einfaches Beispiel mit clickme

```
library(clickme)
clickme("points", 1:10)
```

Ein weiteres clickme Beispiel

```
n <- 500
clickme("points",
  x = rbeta(n, 1, 10), y = rbeta(n, 1, 10),
  names = sample(letters, n, r = T),
  color_groups = sample(LETTERS[1:3], n, r = T),
  title = "Zoom Search Hover Click")
```

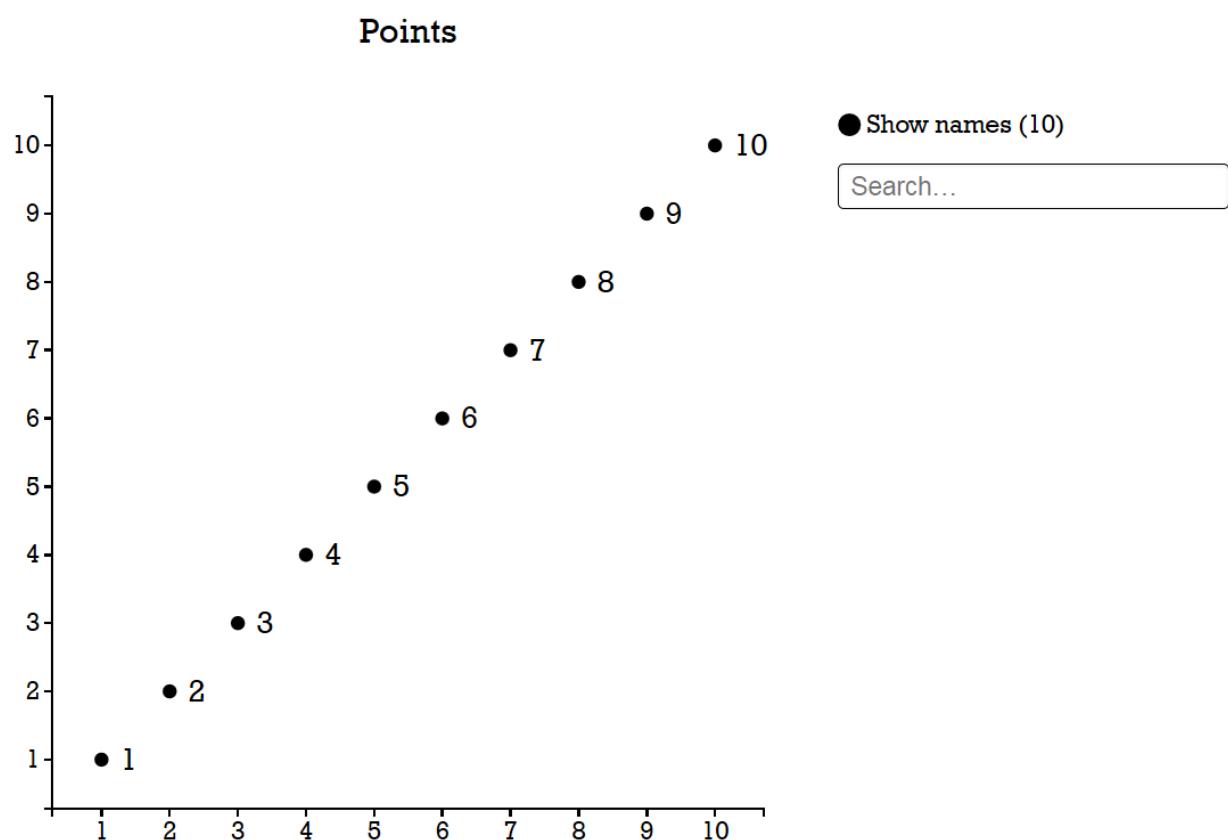


Figure 114:

Zoom Search Hover Click

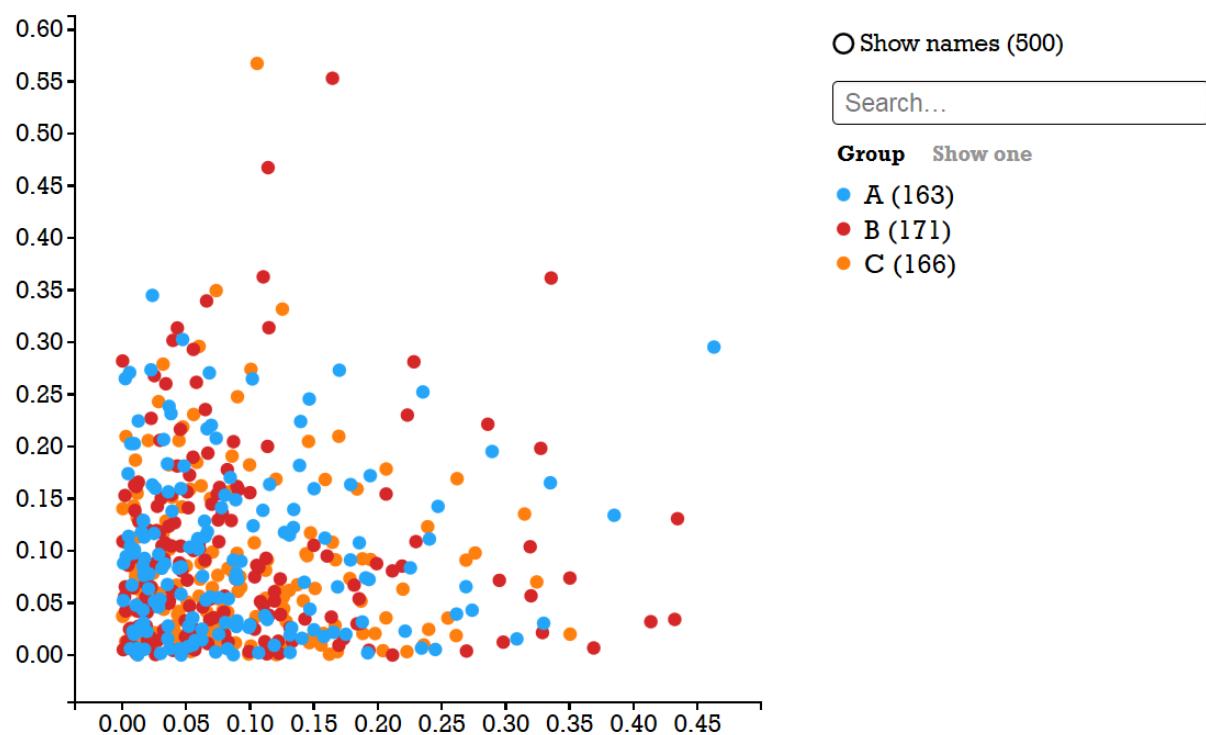


Figure 115:

Das Paket networkD3

```
install.packages("networkD3")
```

networkD3:

D3 JavaScript Network Graphs from R

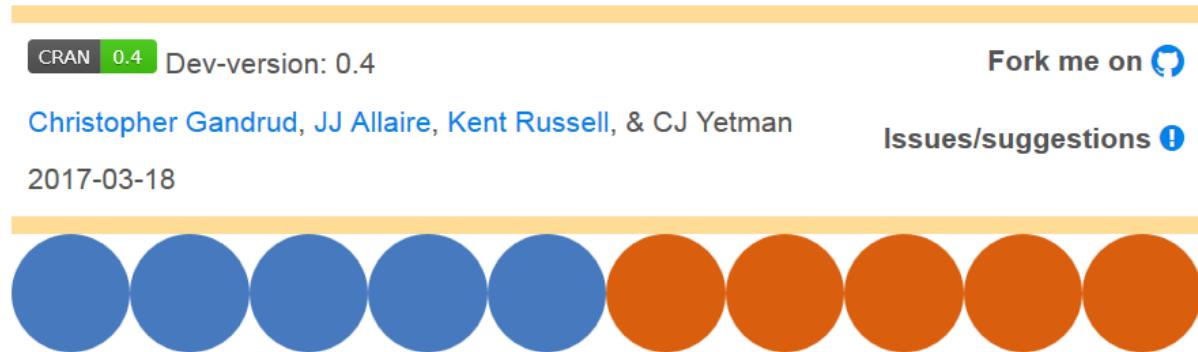


Figure 116:

- der Vorgänger d3Network

Ein Beispiel mit networkD3

```
library(networkD3)
src <- c("A", "A", "A", "A", "B", "B", "C", "C", "D")
target <- c("B", "C", "D", "J", "E", "F", "G", "H", "I")
networkData <- data.frame(src, target)
simpleNetwork(networkData)
```

Zeitreihen interaktiv darstellen mit dygraphs

- HTML widgets in R

```
library(dygraphs)
dygraph(nhtemp, main = "New Haven Temperatures") %>%
  dyRangeSelector(dateWindow = c("1920-01-01", "1960-01-01"))
```

Das Paket threejs

- Introducing JS

```
install.packages("threejs")

library(threejs)
z <- seq(-10, 10, 0.01)
x <- cos(z)
y <- sin(z)
scatterplot3js(x,y,z, color=rainbow(length(z)))
```

Links

- Google Charts
- Three little circles
- Wie schreibt man gute html widgets
- Die widgets exportieren
- R wie Javascript programmieren
- Mit Javascript in R arbeiten
- Paket rbokeh
- Interaktive Graphiken mit SVG
- HTMLwidgets
- Rook - Tools um Webapplikationen mit R zu erstellen

```
install.packages("Rook")
```

Interaktive Graphiken mit D3 und plotly

plotly

- plotly ist ein Online Analyse und Visualisierungstool
- es gibt Pakete für verschiedene Programmiersprachen

plotly und R

plotly Installieren

```
install.packages("plotly")

library("plotly")
```

Der Anfang mit plotly für R

```
plot_ly(midwest, x = ~percollege, color = ~state, type = "box")
```

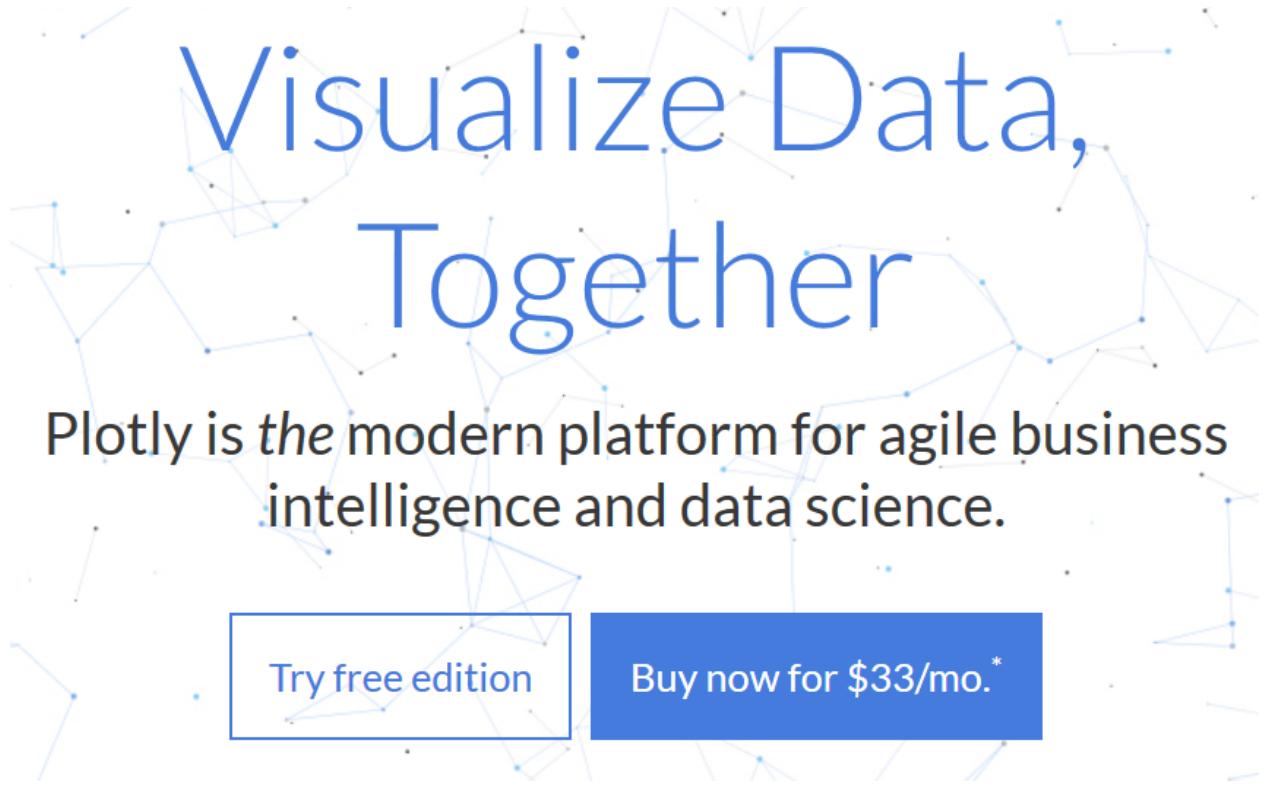


Figure 117:

Getting Started with Plotly for R



Figure 118:

plotly Beispiel mit eigenen Daten

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

plot_ly(whcSites, x = ~date_inscribed, color = ~category_short, type = "box")
```

Links

- der Start mit Plotly: einfache Grafiken

Getting started with Plotly: basic Plots

January 10, 2017

By Sai Lalith

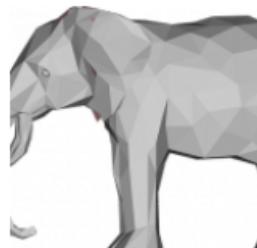
 Like 79  Share  Share 2

(This article was first published on [R-exercises](#), and kindly contributed to R-bloggers)

79 SHARES

 Share

 Tweet



Plotly is a d3 based graphing library used to produce interactive and high quality graphs in R. In the following exercises, we will look at the basic plots' syntax and some basic features in the plotly functions.

We will use datasets available in base R packages.

Refer to the documentation of the plotly packages when in need of help with the functions and arguments.

Also, refer to this [link](#) for figure references

Figure 119:

Netzwerkgraphiken mit vis.js

Einführung in visNetwork

- Rein kommen
- d3-Netzwerkgraphiken mit R

```
install.packages("visNetwork")
```

```
library(visNetwork)
```

Ein Minimalbeispiel

```
nodes <- data.frame(id = 1:3)
edges <- data.frame(from = c(1,2), to = c(1,3))
visNetwork(nodes, edges, width = "100%")
```

Wie es funktioniert

```
visDocumentation()
vignette("Introduction-to-visNetwork") # with CRAN version
```

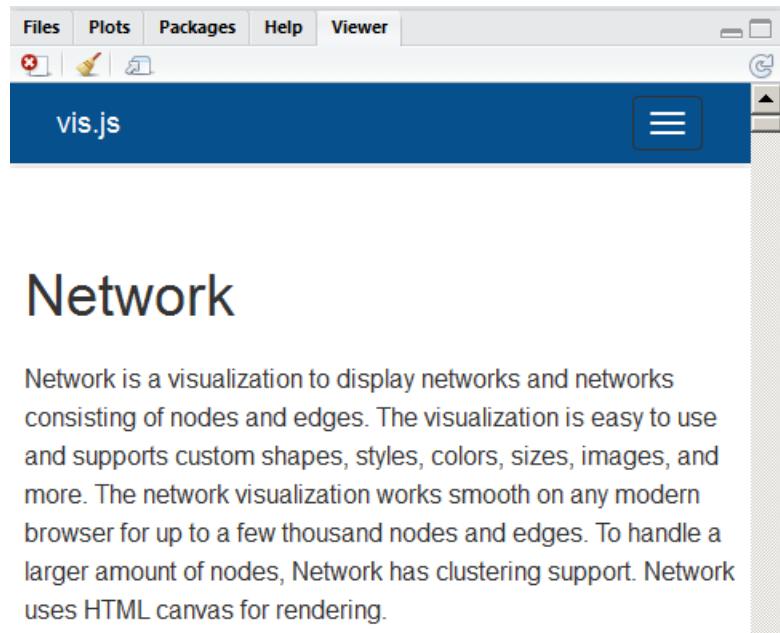


Figure 120:

shiny Beispiel

```
install.packages("shiny")
shiny::runApp(system.file("shiny", package = "visNetwork"))
```

Das Erstellen von Ablaufdiagrammen mit mermaid

Um was geht es?

- Der Start
- Das R-Paket DiagrammeR

```
install.packages('DiagrammeR')
```

```
library('DiagrammeR')
```

Eine einfache Grafik erzeugen

```
DiagrammeR()  
graph LR  
A-->B  
A-->C  
C-->E  
B-->D  
C-->D  
D-->F  
E-->F  
")
```

Ein GANTT Diagramm erstellen

```
DiagrammeR()  
gantt  
dateFormat YYYY-MM-DD  
title Adding GANTT diagram functionality to mermaid  
section A section  
Completed task :done, des1, 2014-01-06,2014-01-08  
Active task :active, des2, 2014-01-09, 3d  
Future task : des3, after des2, 5d  
Future task2 : des4, after des3, 5d  
section Critical tasks  
Completed task in the critical line :crit, done, 2014-01-06,24h  
Implement parser and json :crit, done, after des1, 2d  
Create tests for parser :crit, active, 3d  
Future task in critical line :crit, 5d  
Create tests for renderer :2d  
Add to mermaid :1d  
")
```

Ein weiteres Ganttt Diagramm

```
library(DiagrammeR)  
mermaid()  
gantt  
dateFormat YYYY-MM-DD  
title A Very Nice Ganttt Diagram  
  
section Basic Tasks  
This is completed :done, first_1, 2014-01-06, 2014-01-08  
This is active :active, first_2, 2014-01-09, 3d  
Do this later : first_3, after first_2, 5d
```

```

Do this after that      :           first_4,    after first_3, 5d

section Important Things
Completed, critical task   :crit, done,    import_1, 2014-01-06,24h
Also done, also critical    :crit, done,    import_2, after import_1, 2d
Doing this important task now :crit, active, import_3, after import_2, 3d
Next critical task         :crit,        import_4, after import_3, 5d

section The Extras
First extras                :active,     extras_1, after import_4, 3d
Second helping               :           extras_2, after extras_1, 20h
More of the extras          :           extras_3, after extras_1, 48h
")

```

Links

- Github repo
- Gantt chart
- Javascript Bibliothek mermaid
- viz.js

Internetressourcen und Schnittstellen nutzen

Was sind API's?

Eine Programmierschnittstelle, genauer Schnittstelle zur Anwendungsprogrammierung, häufig nur kurz **API** genannt (englisch application programming interface, wörtlich ‚Anwendungsprogrammierschnittstelle‘), ist ein Programmteil, der von einem Softwaresystem anderen Programmen zur Anbindung an das System zur Verfügung ...

[Programmierschnittstelle – Wikipedia](#)
<https://de.wikipedia.org/wiki/Programmierschnittstelle>

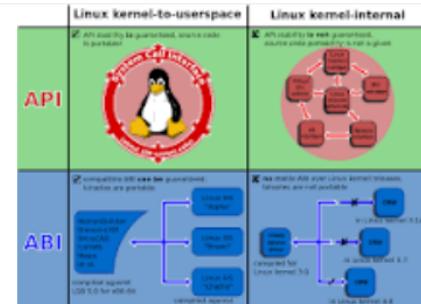


Figure 121:

Programmierschnittstellen

- Relevanz hat der Begriff der API vor allem durch seine Verwendung von Webdiensten erhalten.
- APIs dienen also zum Austausch und der Weiterverarbeitung von Daten und Inhalten zwischen verschiedenen Webseiten, Programmen und Content-Anbietern.

Beispiel

Über die YouTube-API haben Entwickler die Möglichkeit, nach Videos mit gewünschten Parametern, wie zum Beispiel Name oder Länge zu suchen. Die API schickt die Antwort in Form einer XML-Datei zurück. Diese kann dann nach einer Auswertung für die eigene Webseite verwendet werden.

Figure 122:

Bedeutung

- es ist unter anderem möglich Daten von Programmierschnittstellen zu beziehen
- diese Daten sind allerdings nicht als .xlsx, .csv, .dta oder ähnliches abgespeichert sondern in einem der folgenden Formate: .json, .xml etc.

JavaScript Object Notation

Die JavaScript Object Notation, kurz **JSON** [ˈdʒeɪsən], ist ein kompaktes Datenformat in einer einfach lesbaren Textform zum Zweck des Datenaustauschs zwischen Anwendungen. Jedes gültige **JSON**-Dokument soll ein gültiges JavaScript sein und per eval() interpretiert werden können.



[JSON - Wikipedia](#)
https://de.wikipedia.org/wiki/JavaScript_Object_Notation

Figure 123:

Das GeoJSON Format

- GeoJSON ist ein offenes Format um geografische Daten nach der Simple-Feature-Access-Spezifikation zu repräsentieren.
- Dafür wird die JavaScript Object Notation verwendet.

Die Struktur der Daten kann man sich mit einem JSON Viewer anschauen

GeoJSON

- GeoJSON ist ein offenes Format um geografische Daten nach der Simple-Feature-Access-Spezifikation zu repräsentieren.



Figure 124:

- Simple Feature Access ist eine Spezifikation des Open Geospatial Consortium, welche eine allgemein gültige Architektur für geografische Daten und deren Geometrien definiert.
- Die Spezifikation beschreibt einerseits die Speicherung und den Zugriff auf Geometrien und andererseits verschiedene räumliche Operatoren.

OpenStreetMap Daten

Leaflet for R

[Introduction](#)

[The Map Widget](#)

[Basemaps](#)

Working with GeoJSON & TopoJSON

For working with GeoJSON/TopoJSON data you have two options: either read it into `sp` objects; or use the `addGeoJSON()` and `addTopoJSON()` functions.

Figure 125:

Beispiele für GeoJSON

Import von JSON-Objekten und XML Dateien

Import von JavaScript Object Notation (JSON)

- Jedes gültige JSON-Dokument soll ein gültiges JavaScript sein
- JSON wird zur Übertragung und zum Speichern von strukturierten Daten eingesetzt
- Insbesondere bei Webanwendungen und mobilen Apps wird es in Verbindung mit JavaScript, Ajax oder WebSockets zum Transfer von Daten zwischen dem Client und dem Server häufig genutzt.

Einfache Geometrien

| Typ | Beispiel | |
|-----------------------|---|---|
| Point
(Punkt) |  | <pre>{ "type": "Point", "coordinates": [30, 10] }</pre> |
| LineString
(Linie) |  | <pre>{ "type": "LineString", "coordinates": [[30, 10], [10, 30], [40, 40]] }</pre> |
| Polygon |  | <pre>{ "type": "Polygon", "coordinates": [[[30, 10], [40, 40], [20, 40], [10, 20], [30, 10]]] }</pre> |
| |  | <pre>{ "type": "Polygon", "coordinates": [[[35, 10], [45, 45], [15, 40], [10, 20], [35, 10]], [[20, 30], [35, 35], [30, 20], [20, 30]]] }</pre> |

Figure 126:

```
{
    "Herausgeber": "Xema",
    "Nummer": "1234-5678-9012-3456",
    "Deckung": 2e+6,
    "Waehrung": "EURO",
    "Inhaber":
    {
        "Name": "Mustermann",
        "Vorname": "Max",
        "maennlich": true,
        "Hobbys": [ "Reiten", "Golfen", "Lesen" ],
        "Alter": 42,
        "Kinder": [],
        "Partner": null
    }
}
```

Figure 127:

The screenshot shows a JSON viewer interface from jsonviewer.stack.hu. The top bar includes a back arrow, a refresh icon, and the URL 'jsonviewer.stack.hu'. Below the bar, there are tabs for 'Viewer' (which is selected) and 'Text'. The main area displays a hierarchical JSON structure. At the top level, there is a 'type : "FeatureCollection"' entry. Below it, under 'features', there is an entry '0' which contains a 'Feature' object with properties like 'id : "node/246574149"', 'amenity : "drinking_water"', 'flow : "push-button"', and 'type : "nasone"'. This entry also has a 'geometry' object with a 'Point' type and a 'coordinates' array. There is another entry '1' below '0'. The entire interface has a light gray background with blue highlights for selected items.

Figure 128:

Download von Beispieldaten

- Overpass Turbo kann verwendet werden um Beispieldaten zu bekommen

<https://overpass-turbo.eu/>

```

1 /*
2 This is an example Overpass query.
3 Try it out by pressing the Run button above!
4 You can find more examples with the Load
5 tool.
6 */
7 node
8     [amenity=drinking_water]
9     {{bbox}};
10 out;

```

Figure 129:

Exkurs OpenStreetMap Daten

- Auf Overpass Turbo können Daten für Map Features exportiert werden
- Eine Liste der erhältlichen Map Features gibt es auf http://wiki.openstreetmap.org/wiki/DE:Map_Features

| Schlüssel | Wert | Element | Kommentar | Darstellung | Foto |
|----------------------|------------|--|---|-------------|------|
| Verpflegung, Einkehr | | | | | |
| amenity | bar | <input type="checkbox"/> <input checked="" type="checkbox"/> | Bar, Nachtlökal. Es werden hauptsächlich alkoholische Getränke serviert. Siehe auch Beschreibung von amenity=bar und amenity=pub zur Unterscheidung von Bar und Pub (Kneipe). | | |
| amenity | bbq | <input type="checkbox"/> | Grillplatz. Kann mit fuel=* (wood/gas/electric) kombiniert werden. | | |
| amenity | biergarten | <input type="checkbox"/> <input checked="" type="checkbox"/> | Biergarten | | |

Figure 130:

Das Paket jsonlite

```

install.packages("jsonlite")

library(jsonlite)
citation("jsonlite")

##
## To cite jsonlite in publications use:
##
##   Jeroen Ooms (2014). The jsonlite Package: A Practical and
##   Consistent Mapping Between JSON Data and R Objects.
##   arXiv:1403.2805 [stat.CO] URL https://arxiv.org/abs/1403.2805.
##

```

```

## A BibTeX entry for LaTeX users is
##
## @Article{,
##   title = {The jsonlite Package: A Practical and Consistent Mapping Between JSON Data and R Objects},
##   author = {Jeroen Ooms},
##   journal = {arXiv:1403.2805 [stat.CO]},
##   year = {2014},
##   url = {https://arxiv.org/abs/1403.2805},
## }

```

JSON importieren

- die Daten für die Trinkstationen in Rom habe ich mit Overpass Turbo exportiert

```

library("jsonlite")
DRINKWATER <- fromJSON("data/RomDrinkingWater.geojson")

names(DRINKWATER)[1:3]

## [1] "type"      "generator"  "copyright"

names(DRINKWATER)[4:5]

## [1] "timestamp" "features"

```

Die Daten anschauen

```

head(DRINKWATER$features)

## #> type          id properties.@id properties.amenity properties.flow
## #> 1 Feature node/246574149 node/246574149  drinking_water    push-button
## #> 2 Feature node/246574150 node/246574150  drinking_water    <NA>
## #> 3 Feature node/246574151 node/246574151  drinking_water    <NA>
## #> 4 Feature node/248743324 node/248743324  drinking_water    <NA>
## #> 5 Feature node/251773348 node/251773348  drinking_water    <NA>
## #> 6 Feature node/251773551 node/251773551  drinking_water    <NA>
## #> properties.type properties.name properties.name:fr properties.wheelchair
## #> 1           nasone        <NA>            <NA>            <NA>
## #> 2           <NA>         <NA>            <NA>            <NA>
## #> 3           <NA>         <NA>            <NA>            <NA>
## #> 4           <NA>         <NA>            <NA>            <NA>
## #> 5           nasone        <NA>            <NA>            <NA>
## #> 6           <NA>       Acqua Marcia     Eau potable      yes
## #> properties.created_by properties.indoor geometry.type
## #> 1             <NA>            <NA>            Point
## #> 2             <NA>            <NA>            Point
## #> 3             <NA>            <NA>            Point
## #> 4             <NA>            <NA>            Point
## #> 5             <NA>            <NA>            Point
## #> 6             <NA>            <NA>            Point
## #> geometry.coordinates
## #> 1 12.49191, 41.89479
## #> 2 12.49095, 41.89489
## #> 3 12.48774, 41.89450

```

```

## 4 12.48773, 41.89354
## 5 12.48529, 41.88539
## 6 12.48386, 41.89332

```

Github JSON Daten

- Es lassen sich auch Dinge aus dem Web auslesen:

```

my_repos <- fromJSON("https://api.github.com/users/japhilko/repos")

head(my_repos)

##           id          name      full_name
## 1 29143362 2015-01-15-EMBLHeidelberg Japhilko/2015-01-15-EMBLHeidelberg
## 2 39427013             DataAnalysis Japhilko/DataAnalysis
## 3 26485588            DataGeneration Japhilko/DataGeneration
## 4 26164276            DLR_IntroR  Japhilko/DLR_IntroR
## 5 20760765            GeoData    Japhilko/GeoData
## 6 55756271            geosmdata Japhilko/geosmdata
##   owner.login owner.id
## 1     Japhilko 7593396
## 2     Japhilko 7593396
## 3     Japhilko 7593396
## 4     Japhilko 7593396
## 5     Japhilko 7593396
## 6     Japhilko 7593396
##                               owner.avatar_url owner.gravatar_id
## 1 https://avatars2.githubusercontent.com/u/7593396?v=3
## 2 https://avatars2.githubusercontent.com/u/7593396?v=3
## 3 https://avatars2.githubusercontent.com/u/7593396?v=3
## 4 https://avatars2.githubusercontent.com/u/7593396?v=3
## 5 https://avatars2.githubusercontent.com/u/7593396?v=3
## 6 https://avatars2.githubusercontent.com/u/7593396?v=3
##                               owner.url          owner.html_url
## 1 https://api.github.com/users/Japhilko https://github.com/Japhilko
## 2 https://api.github.com/users/Japhilko https://github.com/Japhilko
## 3 https://api.github.com/users/Japhilko https://github.com/Japhilko
## 4 https://api.github.com/users/Japhilko https://github.com/Japhilko
## 5 https://api.github.com/users/Japhilko https://github.com/Japhilko
## 6 https://api.github.com/users/Japhilko https://github.com/Japhilko
##                               owner.followers_url
## 1 https://api.github.com/users/Japhilko/followers
## 2 https://api.github.com/users/Japhilko/followers
## 3 https://api.github.com/users/Japhilko/followers
## 4 https://api.github.com/users/Japhilko/followers
## 5 https://api.github.com/users/Japhilko/followers
## 6 https://api.github.com/users/Japhilko/followers
##                               owner.following_url
## 1 https://api.github.com/users/Japhilko/following{/other_user}
## 2 https://api.github.com/users/Japhilko/following{/other_user}
## 3 https://api.github.com/users/Japhilko/following{/other_user}
## 4 https://api.github.com/users/Japhilko/following{/other_user}
## 5 https://api.github.com/users/Japhilko/following{/other_user}
## 6 https://api.github.com/users/Japhilko/following{/other_user}

```

```

##                                     owner.gists_url
## 1 https://api.github.com/users/Japhilko/gists{/gist_id}
## 2 https://api.github.com/users/Japhilko/gists{/gist_id}
## 3 https://api.github.com/users/Japhilko/gists{/gist_id}
## 4 https://api.github.com/users/Japhilko/gists{/gist_id}
## 5 https://api.github.com/users/Japhilko/gists{/gist_id}
## 6 https://api.github.com/users/Japhilko/gists{/gist_id}
##                                     owner.starred_url
## 1 https://api.github.com/users/Japhilko/starred{/owner}{/repo}
## 2 https://api.github.com/users/Japhilko/starred{/owner}{/repo}
## 3 https://api.github.com/users/Japhilko/starred{/owner}{/repo}
## 4 https://api.github.com/users/Japhilko/starred{/owner}{/repo}
## 5 https://api.github.com/users/Japhilko/starred{/owner}{/repo}
## 6 https://api.github.com/users/Japhilko/starred{/owner}{/repo}
##                                     owner.subscriptions_url
## 1 https://api.github.com/users/Japhilko/subscriptions
## 2 https://api.github.com/users/Japhilko/subscriptions
## 3 https://api.github.com/users/Japhilko/subscriptions
## 4 https://api.github.com/users/Japhilko/subscriptions
## 5 https://api.github.com/users/Japhilko/subscriptions
## 6 https://api.github.com/users/Japhilko/subscriptions
##                                     owner.organizations_url
## 1 https://api.github.com/users/Japhilko/orgs
## 2 https://api.github.com/users/Japhilko/orgs
## 3 https://api.github.com/users/Japhilko/orgs
## 4 https://api.github.com/users/Japhilko/orgs
## 5 https://api.github.com/users/Japhilko/orgs
## 6 https://api.github.com/users/Japhilko/orgs
##                                     owner.repos_url
## 1 https://api.github.com/users/Japhilko/repos
## 2 https://api.github.com/users/Japhilko/repos
## 3 https://api.github.com/users/Japhilko/repos
## 4 https://api.github.com/users/Japhilko/repos
## 5 https://api.github.com/users/Japhilko/repos
## 6 https://api.github.com/users/Japhilko/repos
##                                     owner.events_url
## 1 https://api.github.com/users/Japhilko/events{/privacy}
## 2 https://api.github.com/users/Japhilko/events{/privacy}
## 3 https://api.github.com/users/Japhilko/events{/privacy}
## 4 https://api.github.com/users/Japhilko/events{/privacy}
## 5 https://api.github.com/users/Japhilko/events{/privacy}
## 6 https://api.github.com/users/Japhilko/events{/privacy}
##                                     owner.received_events_url owner.type
## 1 https://api.github.com/users/Japhilko/received_events      User
## 2 https://api.github.com/users/Japhilko/received_events      User
## 3 https://api.github.com/users/Japhilko/received_events      User
## 4 https://api.github.com/users/Japhilko/received_events      User
## 5 https://api.github.com/users/Japhilko/received_events      User
## 6 https://api.github.com/users/Japhilko/received_events      User
##   owner.site_admin private
## 1          FALSE  FALSE
## 2          FALSE  FALSE
## 3          FALSE  FALSE
## 4          FALSE  FALSE

```

```

## 5      FALSE  FALSE
## 6      FALSE  FALSE
##                                     html_url
## 1 https://github.com/Japhilko/2015-01-15-EMBLHeidelberg
## 2           https://github.com/Japhilko/DataAnalysis
## 3           https://github.com/Japhilko/DataGeneration
## 4           https://github.com/Japhilko/DLR_IntroR
## 5           https://github.com/Japhilko/GeoData
## 6           https://github.com/Japhilko/geosmdata
##                                     description fork
## 1 R programming and development (EMBL, Jan 2015)  TRUE
## 2           My research on data analysis FALSE
## 3           Rcode for generating synthatic data FALSE
## 4           Unterlagen für DLR Workshop FALSE
## 5           Research on statistics and geodata FALSE
## 6           package to import OpenstreetMap data FALSE
##                                     url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg
## 2           https://api.github.com/repos/Japhilko/DataAnalysis
## 3           https://api.github.com/repos/Japhilko/DataGeneration
## 4           https://api.github.com/repos/Japhilko/DLR_IntroR
## 5           https://api.github.com/repos/Japhilko/GeoData
## 6           https://api.github.com/repos/Japhilko/geosmdata
##                                     forks_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/forks
## 2           https://api.github.com/repos/Japhilko/DataAnalysis/forks
## 3           https://api.github.com/repos/Japhilko/DataGeneration/forks
## 4           https://api.github.com/repos/Japhilko/DLR_IntroR/forks
## 5           https://api.github.com/repos/Japhilko/GeoData/forks
## 6           https://api.github.com/repos/Japhilko/geosmdata/forks
##                                     keys_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/keys{/key_id}
## 2           https://api.github.com/repos/Japhilko/DataAnalysis/keys{/key_id}
## 3           https://api.github.com/repos/Japhilko/DataGeneration/keys{/key_id}
## 4           https://api.github.com/repos/Japhilko/DLR_IntroR/keys{/key_id}
## 5           https://api.github.com/repos/Japhilko/GeoData/keys{/key_id}
## 6           https://api.github.com/repos/Japhilko/geosmdata/keys{/key_id}
##                                     collaborators_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/collaborators{/collaborator}
## 2           https://api.github.com/repos/Japhilko/DataAnalysis/collaborators{/collaborator}
## 3           https://api.github.com/repos/Japhilko/DataGeneration/collaborators{/collaborator}
## 4           https://api.github.com/repos/Japhilko/DLR_IntroR/collaborators{/collaborator}
## 5           https://api.github.com/repos/Japhilko/GeoData/collaborators{/collaborator}
## 6           https://api.github.com/repos/Japhilko/geosmdata/collaborators{/collaborator}
##                                     teams_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/teams
## 2           https://api.github.com/repos/Japhilko/DataAnalysis/teams
## 3           https://api.github.com/repos/Japhilko/DataGeneration/teams
## 4           https://api.github.com/repos/Japhilko/DLR_IntroR/teams
## 5           https://api.github.com/repos/Japhilko/GeoData/teams
## 6           https://api.github.com/repos/Japhilko/geosmdata/teams
##                                     hooks_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/hooks
## 2           https://api.github.com/repos/Japhilko/DataAnalysis/hooks

```

```

## 3      https://api.github.com/repos/Japhilko/DataGeneration/hooks
## 4      https://api.github.com/repos/Japhilko/DLR_IntroR/hooks
## 5      https://api.github.com/repos/Japhilko/GeoData/hooks
## 6      https://api.github.com/repos/Japhilko/geosmdata/hooks
##
## issue_events_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/issues/events{/number}
## 2      https://api.github.com/repos/Japhilko/DataAnalysis/issues/events{/number}
## 3      https://api.github.com/repos/Japhilko/DataGeneration/issues/events{/number}
## 4      https://api.github.com/repos/Japhilko/DLR_IntroR/issues/events{/number}
## 5      https://api.github.com/repos/Japhilko/GeoData/issues/events{/number}
## 6      https://api.github.com/repos/Japhilko/geosmdata/issues/events{/number}
##
## events_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/events
## 2      https://api.github.com/repos/Japhilko/DataAnalysis/events
## 3      https://api.github.com/repos/Japhilko/DataGeneration/events
## 4      https://api.github.com/repos/Japhilko/DLR_IntroR/events
## 5      https://api.github.com/repos/Japhilko/GeoData/events
## 6      https://api.github.com/repos/Japhilko/geosmdata/events
##
## assignees_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/assignees{/user}
## 2      https://api.github.com/repos/Japhilko/DataAnalysis/assignees{/user}
## 3      https://api.github.com/repos/Japhilko/DataGeneration/assignees{/user}
## 4      https://api.github.com/repos/Japhilko/DLR_IntroR/assignees{/user}
## 5      https://api.github.com/repos/Japhilko/GeoData/assignees{/user}
## 6      https://api.github.com/repos/Japhilko/geosmdata/assignees{/user}
##
## branches_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/branches{/branch}
## 2      https://api.github.com/repos/Japhilko/DataAnalysis/branches{/branch}
## 3      https://api.github.com/repos/Japhilko/DataGeneration/branches{/branch}
## 4      https://api.github.com/repos/Japhilko/DLR_IntroR/branches{/branch}
## 5      https://api.github.com/repos/Japhilko/GeoData/branches{/branch}
## 6      https://api.github.com/repos/Japhilko/geosmdata/branches{/branch}
##
## tags_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/tags
## 2      https://api.github.com/repos/Japhilko/DataAnalysis/tags
## 3      https://api.github.com/repos/Japhilko/DataGeneration/tags
## 4      https://api.github.com/repos/Japhilko/DLR_IntroR/tags
## 5      https://api.github.com/repos/Japhilko/GeoData/tags
## 6      https://api.github.com/repos/Japhilko/geosmdata/tags
##
## blobs_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/git/blobs{/sha}
## 2      https://api.github.com/repos/Japhilko/DataAnalysis/git/blobs{/sha}
## 3      https://api.github.com/repos/Japhilko/DataGeneration/git/blobs{/sha}
## 4      https://api.github.com/repos/Japhilko/DLR_IntroR/git/blobs{/sha}
## 5      https://api.github.com/repos/Japhilko/GeoData/git/blobs{/sha}
## 6      https://api.github.com/repos/Japhilko/geosmdata/git/blobs{/sha}
##
## git_tags_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/git/tags{/sha}
## 2      https://api.github.com/repos/Japhilko/DataAnalysis/git/tags{/sha}
## 3      https://api.github.com/repos/Japhilko/DataGeneration/git/tags{/sha}
## 4      https://api.github.com/repos/Japhilko/DLR_IntroR/git/tags{/sha}
## 5      https://api.github.com/repos/Japhilko/GeoData/git/tags{/sha}
## 6      https://api.github.com/repos/Japhilko/geosmdata/git/tags{/sha}
##
## git_refs_url

```

```

## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/git/refs{/sha}
## 2           https://api.github.com/repos/Japhilko/DataAnalysis/git/refs{/sha}
## 3           https://api.github.com/repos/Japhilko/DataGeneration/git/refs{/sha}
## 4           https://api.github.com/repos/Japhilko/DLR_IntroR/git/refs{/sha}
## 5           https://api.github.com/repos/Japhilko/GeoData/git/refs{/sha}
## 6           https://api.github.com/repos/Japhilko/geosmdata/git/refs{/sha}
##
##                                         trees_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/git/trees{/sha}
## 2           https://api.github.com/repos/Japhilko/DataAnalysis/git/trees{/sha}
## 3           https://api.github.com/repos/Japhilko/DataGeneration/git/trees{/sha}
## 4           https://api.github.com/repos/Japhilko/DLR_IntroR/git/trees{/sha}
## 5           https://api.github.com/repos/Japhilko/GeoData/git/trees{/sha}
## 6           https://api.github.com/repos/Japhilko/geosmdata/git/trees{/sha}
##
##                                         statuses_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/statuses/{sha}
## 2           https://api.github.com/repos/Japhilko/DataAnalysis/statuses/{sha}
## 3           https://api.github.com/repos/Japhilko/DataGeneration/statuses/{sha}
## 4           https://api.github.com/repos/Japhilko/DLR_IntroR/statuses/{sha}
## 5           https://api.github.com/repos/Japhilko/GeoData/statuses/{sha}
## 6           https://api.github.com/repos/Japhilko/geosmdata/statuses/{sha}
##
##                                         languages_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/languages
## 2           https://api.github.com/repos/Japhilko/DataAnalysis/languages
## 3           https://api.github.com/repos/Japhilko/DataGeneration/languages
## 4           https://api.github.com/repos/Japhilko/DLR_IntroR/languages
## 5           https://api.github.com/repos/Japhilko/GeoData/languages
## 6           https://api.github.com/repos/Japhilko/geosmdata/languages
##
##                                         stargazers_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/stargazers
## 2           https://api.github.com/repos/Japhilko/DataAnalysis/stargazers
## 3           https://api.github.com/repos/Japhilko/DataGeneration/stargazers
## 4           https://api.github.com/repos/Japhilko/DLR_IntroR/stargazers
## 5           https://api.github.com/repos/Japhilko/GeoData/stargazers
## 6           https://api.github.com/repos/Japhilko/geosmdata/stargazers
##
##                                         contributors_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/contributors
## 2           https://api.github.com/repos/Japhilko/DataAnalysis/contributors
## 3           https://api.github.com/repos/Japhilko/DataGeneration/contributors
## 4           https://api.github.com/repos/Japhilko/DLR_IntroR/contributors
## 5           https://api.github.com/repos/Japhilko/GeoData/contributors
## 6           https://api.github.com/repos/Japhilko/geosmdata/contributors
##
##                                         subscribers_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/subscribers
## 2           https://api.github.com/repos/Japhilko/DataAnalysis/subscribers
## 3           https://api.github.com/repos/Japhilko/DataGeneration/subscribers
## 4           https://api.github.com/repos/Japhilko/DLR_IntroR/subscribers
## 5           https://api.github.com/repos/Japhilko/GeoData/subscribers
## 6           https://api.github.com/repos/Japhilko/geosmdata/subscribers
##
##                                         subscription_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/subscription
## 2           https://api.github.com/repos/Japhilko/DataAnalysis/subscription
## 3           https://api.github.com/repos/Japhilko/DataGeneration/subscription
## 4           https://api.github.com/repos/Japhilko/DLR_IntroR/subscription
## 5           https://api.github.com/repos/Japhilko/GeoData/subscription

```

```

## 6 https://api.github.com/repos/Japhilko/geosmdata/subscription
##                                         commits_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/commits{/sha}
## 2 https://api.github.com/repos/Japhilko/DataAnalysis/commits{/sha}
## 3 https://api.github.com/repos/Japhilko/DataGeneration/commits{/sha}
## 4 https://api.github.com/repos/Japhilko/DLR_IntroR/commits{/sha}
## 5 https://api.github.com/repos/Japhilko/GeoData/commits{/sha}
## 6 https://api.github.com/repos/Japhilko/geosmdata/commits{/sha}
##
##                                         git_commits_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/git/commits{/sha}
## 2 https://api.github.com/repos/Japhilko/DataAnalysis/git/commits{/sha}
## 3 https://api.github.com/repos/Japhilko/DataGeneration/git/commits{/sha}
## 4 https://api.github.com/repos/Japhilko/DLR_IntroR/git/commits{/sha}
## 5 https://api.github.com/repos/Japhilko/GeoData/git/commits{/sha}
## 6 https://api.github.com/repos/Japhilko/geosmdata/git/commits{/sha}
##
##                                         comments_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/comments{/number}
## 2 https://api.github.com/repos/Japhilko/DataAnalysis/comments{/number}
## 3 https://api.github.com/repos/Japhilko/DataGeneration/comments{/number}
## 4 https://api.github.com/repos/Japhilko/DLR_IntroR/comments{/number}
## 5 https://api.github.com/repos/Japhilko/GeoData/comments{/number}
## 6 https://api.github.com/repos/Japhilko/geosmdata/comments{/number}
##
##                                         issue_comment_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/issues/comments{/number}
## 2 https://api.github.com/repos/Japhilko/DataAnalysis/issues/comments{/number}
## 3 https://api.github.com/repos/Japhilko/DataGeneration/issues/comments{/number}
## 4 https://api.github.com/repos/Japhilko/DLR_IntroR/issues/comments{/number}
## 5 https://api.github.com/repos/Japhilko/GeoData/issues/comments{/number}
## 6 https://api.github.com/repos/Japhilko/geosmdata/issues/comments{/number}
##
##                                         contents_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/contents/{+path}
## 2 https://api.github.com/repos/Japhilko/DataAnalysis/contents/{+path}
## 3 https://api.github.com/repos/Japhilko/DataGeneration/contents/{+path}
## 4 https://api.github.com/repos/Japhilko/DLR_IntroR/contents/{+path}
## 5 https://api.github.com/repos/Japhilko/GeoData/contents/{+path}
## 6 https://api.github.com/repos/Japhilko/geosmdata/contents/{+path}
##
##                                         compare_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/compare/{base}...{head}
## 2 https://api.github.com/repos/Japhilko/DataAnalysis/compare/{base}...{head}
## 3 https://api.github.com/repos/Japhilko/DataGeneration/compare/{base}...{head}
## 4 https://api.github.com/repos/Japhilko/DLR_IntroR/compare/{base}...{head}
## 5 https://api.github.com/repos/Japhilko/GeoData/compare/{base}...{head}
## 6 https://api.github.com/repos/Japhilko/geosmdata/compare/{base}...{head}
##
##                                         merges_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/merges
## 2 https://api.github.com/repos/Japhilko/DataAnalysis/merges
## 3 https://api.github.com/repos/Japhilko/DataGeneration/merges
## 4 https://api.github.com/repos/Japhilko/DLR_IntroR/merges
## 5 https://api.github.com/repos/Japhilko/GeoData/merges
## 6 https://api.github.com/repos/Japhilko/geosmdata/merges
##
##                                         archive_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg{/archive_format}{/ref}
## 2 https://api.github.com/repos/Japhilko/DataAnalysis{/archive_format}{/ref}
## 3 https://api.github.com/repos/Japhilko/DataGeneration{/archive_format}{/ref}

```

```

## 4 https://api.github.com/repos/Japhilko/DLR_IntroR/{archive_format}{/ref}
## 5 https://api.github.com/repos/Japhilko/GeoData/{archive_format}{/ref}
## 6 https://api.github.com/repos/Japhilko/geosmdata/{archive_format}{/ref}
##
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/downloads
## 2 https://api.github.com/repos/Japhilko/DataAnalysis/downloads
## 3 https://api.github.com/repos/Japhilko/DataGeneration/downloads
## 4 https://api.github.com/repos/Japhilko/DLR_IntroR/downloads
## 5 https://api.github.com/repos/Japhilko/GeoData/downloads
## 6 https://api.github.com/repos/Japhilko/geosmdata/downloads
##
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/Downloads
## 2 https://api.github.com/repos/Japhilko/DataAnalysis/Downloads
## 3 https://api.github.com/repos/Japhilko/DataGeneration/Downloads
## 4 https://api.github.com/repos/Japhilko/DLR_IntroR/Downloads
## 5 https://api.github.com/repos/Japhilko/GeoData/Downloads
## 6 https://api.github.com/repos/Japhilko/geosmdata/Downloads
##
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/Issues{/number}
## 2 https://api.github.com/repos/Japhilko/DataAnalysis/Issues{/number}
## 3 https://api.github.com/repos/Japhilko/DataGeneration/Issues{/number}
## 4 https://api.github.com/repos/Japhilko/DLR_IntroR/Issues{/number}
## 5 https://api.github.com/repos/Japhilko/GeoData/Issues{/number}
## 6 https://api.github.com/repos/Japhilko/geosmdata/Issues{/number}
##
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/Pulls{/number}
## 2 https://api.github.com/repos/Japhilko/DataAnalysis/Pulls{/number}
## 3 https://api.github.com/repos/Japhilko/DataGeneration/Pulls{/number}
## 4 https://api.github.com/repos/Japhilko/DLR_IntroR/Pulls{/number}
## 5 https://api.github.com/repos/Japhilko/GeoData/Pulls{/number}
## 6 https://api.github.com/repos/Japhilko/geosmdata/Pulls{/number}
##
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/Milestones{/number}
## 2 https://api.github.com/repos/Japhilko/DataAnalysis/Milestones{/number}
## 3 https://api.github.com/repos/Japhilko/DataGeneration/Milestones{/number}
## 4 https://api.github.com/repos/Japhilko/DLR_IntroR/Milestones{/number}
## 5 https://api.github.com/repos/Japhilko/GeoData/Milestones{/number}
## 6 https://api.github.com/repos/Japhilko/geosmdata/Milestones{/number}
##
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/Notifications{/since,all,participants}
## 2 https://api.github.com/repos/Japhilko/DataAnalysis/Notifications{/since,all,participants}
## 3 https://api.github.com/repos/Japhilko/DataGeneration/Notifications{/since,all,participants}
## 4 https://api.github.com/repos/Japhilko/DLR_IntroR/Notifications{/since,all,participants}
## 5 https://api.github.com/repos/Japhilko/GeoData/Notifications{/since,all,participants}
## 6 https://api.github.com/repos/Japhilko/geosmdata/Notifications{/since,all,participants}
##
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/Labels{/name}
## 2 https://api.github.com/repos/Japhilko/DataAnalysis/Labels{/name}
## 3 https://api.github.com/repos/Japhilko/DataGeneration/Labels{/name}
## 4 https://api.github.com/repos/Japhilko/DLR_IntroR/Labels{/name}
## 5 https://api.github.com/repos/Japhilko/GeoData/Labels{/name}
## 6 https://api.github.com/repos/Japhilko/geosmdata/Labels{/name}
##
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/Releases{/id}
## 2 https://api.github.com/repos/Japhilko/DataAnalysis/Releases{/id}
## 3 https://api.github.com/repos/Japhilko/DataGeneration/Releases{/id}
## 4 https://api.github.com/repos/Japhilko/DLR_IntroR/Releases{/id}
## 5 https://api.github.com/repos/Japhilko/GeoData/Releases{/id}
## 6 https://api.github.com/repos/Japhilko/geosmdata/Releases{/id}
##
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/Deployments

```

```

## 2           https://api.github.com/repos/Japhilko/DataAnalysis/deployments
## 3           https://api.github.com/repos/Japhilko/DataGeneration/deployments
## 4           https://api.github.com/repos/Japhilko/DLR_IntroR/deployments
## 5           https://api.github.com/repos/Japhilko/GeoData/deployments
## 6           https://api.github.com/repos/Japhilko/geosmdata/deployments
##   created_at      updated_at      pushed_at
## 1 2015-01-12T15:59:33Z 2015-01-12T15:59:34Z 2015-01-10T22:26:12Z
## 2 2015-07-21T06:00:37Z 2016-02-04T13:01:54Z 2017-04-24T14:20:11Z
## 3 2014-11-11T13:14:01Z 2015-04-21T14:51:01Z 2015-07-27T13:59:39Z
## 4 2014-11-04T10:34:17Z 2016-07-26T08:22:47Z 2016-08-11T13:23:54Z
## 5 2014-06-12T08:51:41Z 2017-03-23T06:00:42Z 2017-03-23T15:31:16Z
## 6 2016-04-08T06:35:45Z 2016-06-06T10:36:01Z 2016-06-08T11:06:58Z
##   git_url
## 1 git://github.com/Japhilko/2015-01-15-EMBLHeidelberg.git
## 2           git://github.com/Japhilko/DataAnalysis.git
## 3           git://github.com/Japhilko/DataGeneration.git
## 4           git://github.com/Japhilko/DLR_IntroR.git
## 5           git://github.com/Japhilko/GeoData.git
## 6           git://github.com/Japhilko/geosmdata.git
##   ssh_url
## 1 git@github.com:Japhilko/2015-01-15-EMBLHeidelberg.git
## 2           git@github.com:Japhilko/DataAnalysis.git
## 3           git@github.com:Japhilko/DataGeneration.git
## 4           git@github.com:Japhilko/DLR_IntroR.git
## 5           git@github.com:Japhilko/GeoData.git
## 6           git@github.com:Japhilko/geosmdata.git
##   clone_url
## 1 https://github.com/Japhilko/2015-01-15-EMBLHeidelberg.git
## 2           https://github.com/Japhilko/DataAnalysis.git
## 3           https://github.com/Japhilko/DataGeneration.git
## 4           https://github.com/Japhilko/DLR_IntroR.git
## 5           https://github.com/Japhilko/GeoData.git
## 6           https://github.com/Japhilko/geosmdata.git
##   svn_url homepage    size
## 1 https://github.com/Japhilko/2015-01-15-EMBLHeidelberg <NA> 5667
## 2           https://github.com/Japhilko/DataAnalysis <NA> 55636
## 3           https://github.com/Japhilko/DataGeneration <NA> 336
## 4           https://github.com/Japhilko/DLR_IntroR <NA> 32546
## 5           https://github.com/Japhilko/GeoData <NA> 1589706
## 6           https://github.com/Japhilko/geosmdata <NA> 19931
##   stargazers_count watchers_count      language has_issues has_projects
## 1             0            0        TeX     FALSE     TRUE
## 2             0            0       HTML     TRUE     TRUE
## 3             0            0         R     TRUE     TRUE
## 4             2            2         R     TRUE     TRUE
## 5             6            6       HTML     TRUE     TRUE
## 6             0            0 ActionScript     TRUE     TRUE
##   has_downloads has_wiki has_pages forks_count mirror_url
## 1      TRUE    TRUE    FALSE          0       NA
## 2      TRUE    TRUE    FALSE          1       NA
## 3      TRUE    TRUE    TRUE          0       NA
## 4      TRUE    TRUE    FALSE          0       NA
## 5      TRUE    TRUE    TRUE          1       NA
## 6      TRUE    TRUE    FALSE          0       NA

```

```

##   open_issues_count forks open_issues watchers default_branch
## 1             0      0        0       0      master
## 2             0      1        0       0      master
## 3             0      0        0       0      master
## 4             0      0        0       2      master
## 5             1      1        1       6      master
## 6             0      0        0       0      master

```

Weiteres Beispiel für JSON Daten

- Die Ergast Developer API ist ein experimenteller Web Service, der eine historische Aufzeichnung von Motorsportdaten liefert.



Figure 131:

Ergast Daten lesen

```

library(knitr)
library(jsonlite)
res <- fromJSON('http://ergast.com/api/f1/2004/1/results.json')
drivers <- res$MRData$RaceTable$Races$Results[[1]]$Driver
kable(head(drivers))

```

| driverId | code | url | givenName | familyName | dateO |
|--------------------|------|---|-----------|---------------|-------|
| michael_schumacher | MSC | http://en.wikipedia.org/wiki/Michael_Schumacher | Michael | Schumacher | 1969- |
| barrichello | BAR | http://en.wikipedia.org/wiki/Rubens_Barrichello | Rubens | Barrichello | 1972- |
| alonso | ALO | http://en.wikipedia.org/wiki/Fernando_Alonso | Fernando | Alonso | 1981- |
| ralf_schumacher | SCH | http://en.wikipedia.org/wiki/Ralf_Schumacher | Ralf | Schumacher | 1975- |
| montoya | MON | http://en.wikipedia.org/wiki/Juan_Pablo_Montoya | Juan | Pablo Montoya | 1975- |
| button | BUT | http://en.wikipedia.org/wiki/Jenson_Button | Jenson | Button | 1980- |

Daten der New York Times

- Die New York Times hat mehrere APIs als Teil des NYT-Entwickler-Netzwerks.
- Es ist eine Schnittstelle zu Daten aus verschiedenen Abteilungen, wie Nachrichtenartikel, Buchbesprechungen, Immobilien, etc.
- Registrierung ist erforderlich (aber kostenlos) und ein Schlüssel kann hier erhalten werden.

New York Times Beispiel

```

article_key <- "&api-key=c2fede7bd9aea57c898f538e5ec0a1ee:6:68700045"
url <- "http://api.nytimes.com/svc/search/v2/articlesearch.json?q=obamacare+socialism"

```

```

req <- fromJSON(paste0(url, article_key))
articles <- req$response$docs
# kable(head(articles))

```

XML Dateien einlesen

```

<?xml version="1.0" encoding="UTF-8"?>
- <osm generator="Overpass API" version="0.6">
  <note>The data included in this document is from www.openstreetmap.org. The data is made available under ODbL.</note>
  <meta areas="2017-05-07T07:22:03Z" osm_base="2017-05-07T08:19:03Z"/>
  - <node lon="8.5028074" lat="49.5190994" id="30560755">
    <tag v="Potlatch 0.5d" k="created_by"/>
    <tag v="playground" k="leisure"/>
  </node>
  - <node lon="8.5393963" lat="49.4963345" id="76468450">
    <tag v="Potlatch 0.4a" k="created_by"/>
    <tag v="playground" k="leisure"/>
    <tag v="Rutsche, Schaukel, großer Sandkasten, Tischtennis" k="note"/>
  </node>
  - <node lon="8.5529589" lat="49.4967807" id="76468534">
    <tag v="playground" k="leisure"/>
  </node>
  - <node lon="8.5487501" lat="49.4923030" id="76468535">
    <tag v="playground" k="leisure"/>
  </node>
  - <node lon="8.5481399" lat="49.5024260" id="76468536">
    <tag v="playground" k="leisure"/>
    <tag v="Schaukel, Rutsche, Sandkasten, Spielhäuser, Tischtennis" k="note"/>
  </node>
  - <node lon="8.5420356" lat="49.4975937" id="76468558">
    <tag v="playground" k="leisure"/>
  </node>

```

Figure 132:

Import von XML Dateien

```

install.packages("XML")

library(XML)
citation("XML")

##
## To cite package 'XML' in publications use:
##
## Duncan Temple Lang and the CRAN Team (2016). XML: Tools for
## Parsing and Generating XML Within R and S-Plus. R package
## version 3.98-1.5. https://CRAN.R-project.org/package=XML
##
## A BibTeX entry for LaTeX users is
##
## @Manual{,
##   title = {XML: Tools for Parsing and Generating XML Within R and S-Plus},
##   author = {Duncan Temple Lang and the CRAN Team},
##   year = {2016},
##   note = {R package version 3.98-1.5},
##   url = {https://CRAN.R-project.org/package=XML},
## }
##
## ATTENTION: This citation information has been auto-generated from

```

```
## the package DESCRIPTION file and may need manual editing, see  
## 'help("citation")'.
```

Das R-Paket XML - Gaston Sanchez

```
library("XML")
```



Figure 133: Gaston Sanchez - Dataflow

Seine Arbeit sieht man hier.

Das Arbeiten mit XML Daten

Funktionen im XML Paket

| Function | Description |
|----------------|--|
| xmlName() | name of the node |
| xmlSize() | number of subnodes |
| xmlAttrs() | named character vector of all attributes |
| xmlGetAttr() | value of a single attribute |
| xmlValue() | contents of a leaf node |
| xmlParent() | name of parent node |
| xmlAncestors() | name of ancestor nodes |
| getSibling() | siblings to the right or to the left |
| xmlNamespace() | the namespace (if there's one) |

Getting Data from the Web with R

Part 4: Parsing XML/HTML Content

Gaston Sanchez

April-May 2014

Content licensed under [CC BY-NC-SA 4.0](#)

Figure 134: Gaston Sanchez - Webdaten bekommen

Das neuere xml2 Paket

```
install.packages("xml2")
library(xml2)
citation("xml2")

##
## To cite package 'xml2' in publications use:
##
##   Hadley Wickham, James Hester and Jeroen Ooms (2017). xml2: Parse
##   XML. R package version 1.1.0.
##   https://CRAN.R-project.org/package=xml2
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {xml2: Parse XML},
##     author = {Hadley Wickham and James Hester and Jeroen Ooms},
##     year = {2017},
##     note = {R package version 1.1.0},
##     url = {https://CRAN.R-project.org/package=xml2},
##   }
```

Beispiel Daten - die OpenStreetMap API

- OpenStreetMap hat eine Editier-API welche zum Lesen und Speichern von Rohen Geodaten von bzw. auf die OpenStreetMap Datenbank benutzt werden kann.

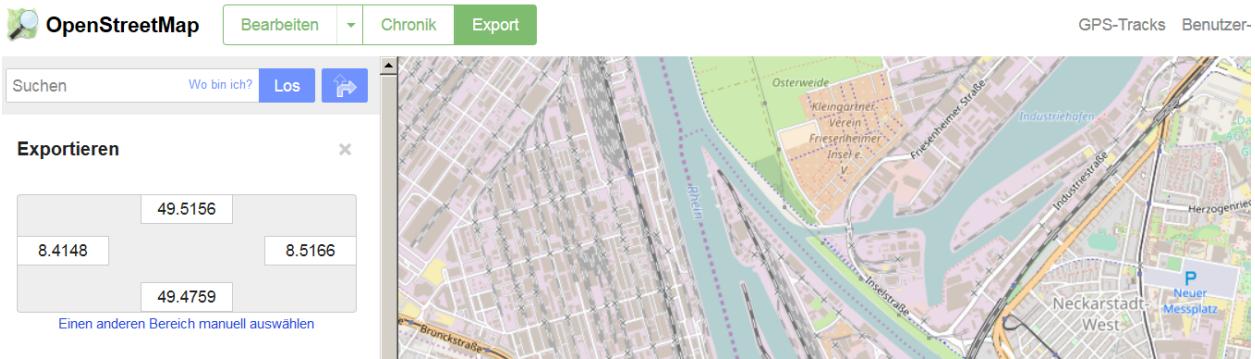


Figure 135:

Die OpenStreetMap ID herausfinden

| is_in | Mannheim, Baden-Württemberg, Bundesrepublik Deutschland, Europe |
|-----------|---|
| name | Jungbusch |
| place | suburb |
| wikidata | Q1713309 |
| wikipedia | de:Jungbusch |

Figure 136:

Einzelne Objekte finden

<www.openstreetmap.org/export>

OSM Ausschnitte herunterladen

<www.openstreetmap.org/export>

Erstes Beispiel

```
url <- "http://api.openstreetmap.org/api/0.6/
relation/62422"
```

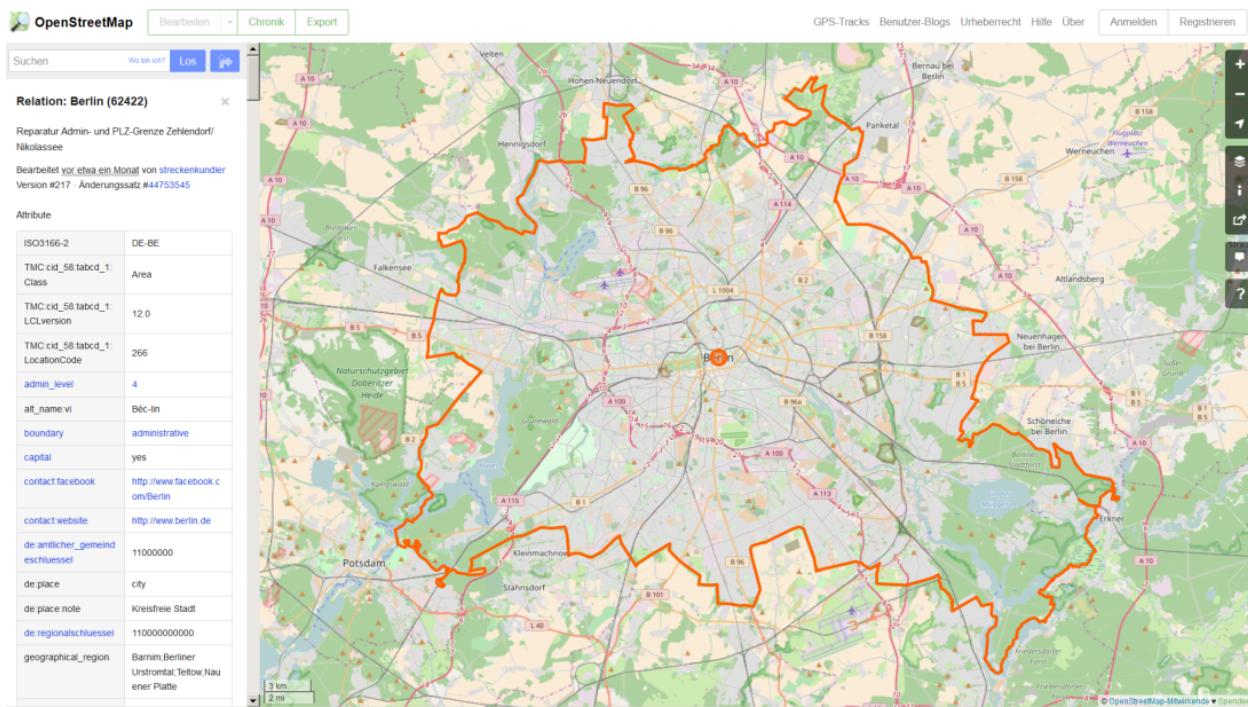


Figure 137: osm export

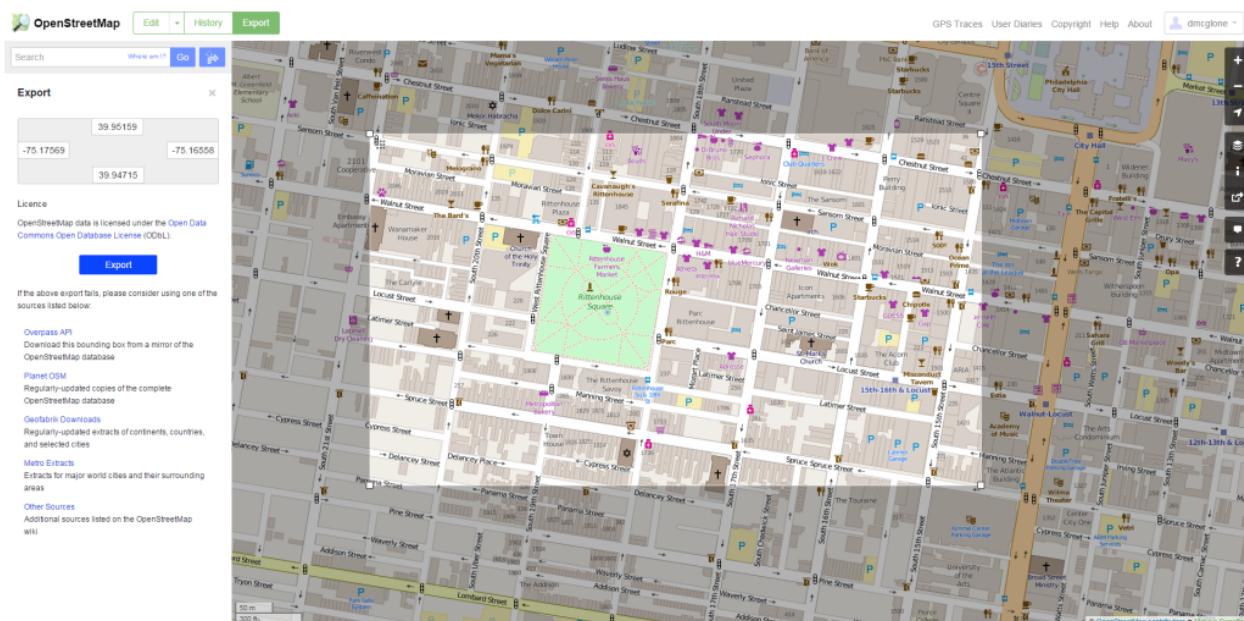


Figure 138: osm export

```

library(xml2)
BE <- xmlParse(url)

<osm version="0.6" generator="CGImap 0.4.0 (19884 thorn-03.openstreetmap.org)" copyright="OpenStreetMap and contributors" attribution="http://www.openstreetmap.org/copyright"
license="http://opendatacommons.org/licenses/odbl/1-0/"
-><relation id="62422" visible="true" version="209" changeset="36072269" timestamp="2015-12-20T19:49:52Z" user="tbcr" uid="278800">
<member type="node" ref="240109189" role="admin_centre"/>
<member type="way" ref="50291800" role="outer"/>
<member type="way" ref="77913336" role="outer"/>
<member type="way" ref="315222039" role="outer"/>
<member type="way" ref="77487568" role="outer"/>
<member type="way" ref="315222038" role="outer"/>
<member type="way" ref="98035898" role="outer"/>
<member type="way" ref="77501737" role="outer"/>

```

Figure 139: Administrative Grenzen Berlin

Das XML analysieren

- Tobi Bosede - Working with XML Data in R

```

xmltop = xmlRoot(BE)
class(xmltop)

## [1] "XMLInternalElementNode" "XMLInternalNode"
## [3] "XMLAbstractNode"

xmlSize(xmltop)

## [1] 1
xmlSize(xmltop[[1]])

## [1] 328

```

Nutzung von Xpath

Xpath, the XML Path Language, is a query language for selecting nodes from an XML document.

```

xpathApply(BE, "//tag[@k = 'source:population']")
## [[1]]
## <tag k="source:population" v="http://www.statistik-berlin-brandenburg.de/Publikationen/Stat_Berichte"
## 
## attr("class")
## [1] "XMLNodeSet"

```

Beispiel: administrative Grenzen Berlin

Administrative Grenzen für Deutschland

```

url <- "http://api.openstreetmap.org/api/0.6/relation/62422"
BE <- xmlParse(url)

```

```

- <osm version="0.6" generator="CGImap 0.4.0 (19884 thorn-03.openstreetmap.org)" copyright="OpenStreetMap and contributors" attribution="http://www.openstreetmap.org/copyright"
  license="http://opendatacommons.org/licenses/odbl/1-0/"
- <relation id="62422" visible="true" version="209" changeset="36072269" timestamp="2015-12-20T19:49:52Z" user="tbcr" uid="278800">
  <member type="node" ref="240109189" role="admin_centre"/>
  <member type="way" ref="50291800" role="outer"/>
  <member type="way" ref="7913336" role="outer"/>
  <member type="way" ref="315222039" role="outer"/>
  <member type="way" ref="77487568" role="outer"/>
  <member type="way" ref="315222038" role="outer"/>
  <member type="way" ref="98035898" role="outer"/>
  <member type="way" ref="77501737" role="outer"/>

```

Figure 140: Administrative Grenzen Berlin

Quelle für die Bevölkerungsgröße

```
xpathApply(BE, "//tag[@k = 'source:population']")
```

```

## [[1]]
## <tag k="source:population" v="http://www.statistik-berlin-brandenburg.de/Publikationen/Stat_Berichte"
##
## attr(,"class")
## [1] "XMLNodeSet"

```

-Statistik Berlin Brandenburg

Etwas überraschend:

```
xpathApply(BE, "//tag[@k = 'name:ta']")
```

```

## [[1]]
## <tag k="name:ta" v="<U+0BAA><U+0BC6><U+0BB0><U+0BCD><U+0BB2><U+0BBF><U+0BA9><U+0BCD>" />
##
## attr(,"class")
## [1] "XMLNodeSet"

```

Geographische Region

```

region <- xpathApply(BE,
  "//tag[@k = 'geographical_region']")
# regular expressions
region[[1]]

## <tag k="geographical_region" v="Barnim;Berliner Urstromtal;Teltow;Nauener Platte"/>
<tag k="geographical_region"
  v="Barnim;Berliner Urstromtal;
  Teltow;Nauener Platte"/>

```

Landkreis

Weiteres Beispiel

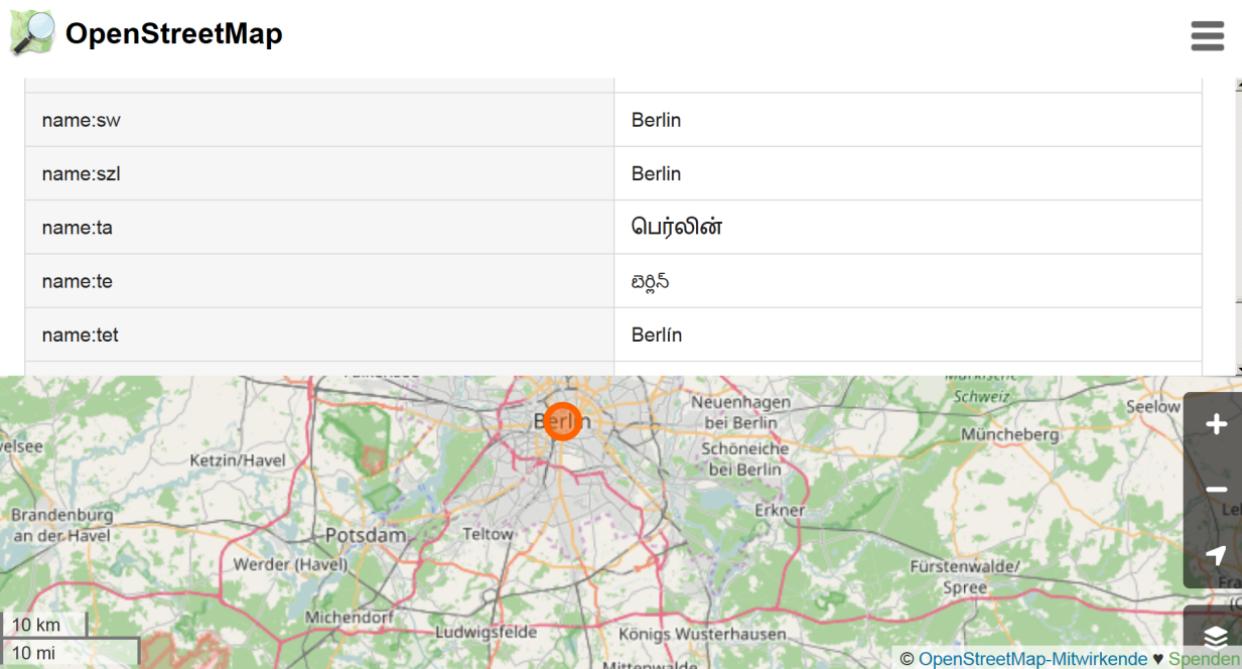


Figure 141:



Figure 142: Barnim

```

url2<-"http://api.openstreetmap.org/api/0.6/node/25113879"
obj2<-xmlParse(url2)
obj_amenity<-xpathApply(obj2,"//tag[@k = 'amenity']")[[1]]
obj_amenity

## <tag k="amenity" v="university"/>

```

Wikipedia Artikel

```

xpathApply(obj2,"//tag[@k = 'wikipedia']")[[1]]

## <tag k="wikipedia" v="de:Universität Mannheim"/>
xpathApply(obj2,"//tag[@k = 'wheelchair']")[[1]]

## <tag k="wheelchair" v="limited"/>
xpathApply(obj2,"//tag[@k = 'name']")[[1]]

## <tag k="name" v="Universität Mannheim"/>

```

Das C und das A

```

url3<-"http://api.openstreetmap.org/api/0.6/node/303550876"
obj3 <- xmlParse(url3)
xpathApply(obj3,"//tag[@k = 'opening_hours']")[[1]]

## <tag k="opening_hours" v="Mo-Sa 09:00-20:00; Su,PH off"/>

```

Nur Fliegen ist schöner

```

url5<-"http://api.openstreetmap.org/api/0.6/way/162149882"
obj5<-xmlParse(url5)
xpathApply(obj5,"//tag[@k = 'name']")[[1]]

## <tag k="name" v="City-Airport Mannheim"/>
xpathApply(obj5,"//tag[@k = 'website']")[[1]]

## <tag k="website" v="http://www.flugplatz-mannheim.de"/>
xpathApply(obj5,"//tag[@k = 'iata']")[[1]]

## <tag k="iata" v="MHG"/>

```

Einen Punkt parsen

```

url2 <- "http://api.openstreetmap.org/api/0.6/node/2923760808"
RennesBa <- xmlParse(url2)
RennesBa

```

```

## <?xml version="1.0" encoding="UTF-8"?>
## <osm version="0.6" generator="CGImap 0.6.0 (7701 thorn-03.openstreetmap.org)" copyright="OpenStreetMa
##   <node id="2923760808" visible="true" version="7" changeset="47392918" timestamp="2017-04-02T20:42:0
##     <tag k="addr:city" v="Rennes"/>
##     <tag k="addr:country" v="FR"/>
##     <tag k="addr:housenumber" v="25"/>
##     <tag k="addr:postcode" v="35000"/>
##     <tag k="addr:street" v="Avenue Jean Janvier"/>
##     <tag k="amenity" v="restaurant"/>
##     <tag k="capacity" v="90"/>
##     <tag k="name" v="Il Basilico"/>
##     <tag k="source:addr:housenumber" v="Rennes MÃ©tropole"/>
##     <tag k="source:addr:housenumber:ref" v="66075"/>
##     <tag k="source:addr:housenumber:version" v="2013-04-02"/>
##     <tag k="website" v="http://ilbasilico.fr"/>
##     <tag k="wheelchair" v="limited"/>
##     <tag k="wheelchair:description" v="Aucune sonnette pour indiquer sa prÃ©sence mais une rampe d'a
##   </node>
## </osm>
##

```

Einen Weg parsen

```

url3 <- "http://api.openstreetmap.org/api/0.6/way/72799743"
MadCalle <- xmlParse(url3)
MadCalle

## <?xml version="1.0" encoding="UTF-8"?>
## <osm version="0.6" generator="CGImap 0.6.0 (17113 thorn-02.openstreetmap.org)" copyright="OpenStreet
##   <way id="72799743" visible="true" version="5" changeset="11915713" timestamp="2012-06-16T14:49:40Z
##     <nd ref="869268876"/>
##     <nd ref="1790008568"/>
##     <nd ref="864117544"/>
##     <nd ref="1790008571"/>
##     <nd ref="1790008601"/>
##     <nd ref="864117511"/>
##     <nd ref="1790008612"/>
##     <nd ref="1790008618"/>
##     <nd ref="864117819"/>
##     <tag k="highway" v="residential"/>
##     <tag k="name" v="Calle Alfonso Ercilla"/>
##     <tag k="oneway" v="yes"/>
##     <tag k="surface" v="asphalt"/>
##   </way>
## </osm>
##

```

The Overpass API

The Overpass API is a read-only API that serves up custom selected parts of the OSM map data.
[\(\[http://wiki.openstreetmap.org/wiki/Overpass_API\]\(http://wiki.openstreetmap.org/wiki/Overpass_API\)\)](http://wiki.openstreetmap.org/wiki/Overpass_API)



Figure 143: Logo Overpass API

Wichtige Information

http://wiki.openstreetmap.org/wiki/Map_Features

Beispiel: Nutzung der Overpass API

Export der Rohdaten

Import von der Overpass API zu R

```
library(XML)
place <- "Mannheim"
type_obj <- "node"
object <- "leisure=playground"

InfoList <- xmlParse(paste(Link1,place,"\\"],",
type_obj,"(area)[",object,"];out;",sep=""))
```

XML Output

Das Arbeiten mit XML Daten (xpath)

Die Liste der ID's mit dem Wert *playground*:

```
node_id <- xpathApply(InfoList,
"//tag[@v= 'playground']/parent::node/@ id")
## node_id[[1]]
```

latitude und longitude bekommen

```
lat_x <- xpathApply(InfoList,
"//tag[@v= 'playground']/parent::node/@ lat")
# lat_x[[1]];lat_x[[2]]
```

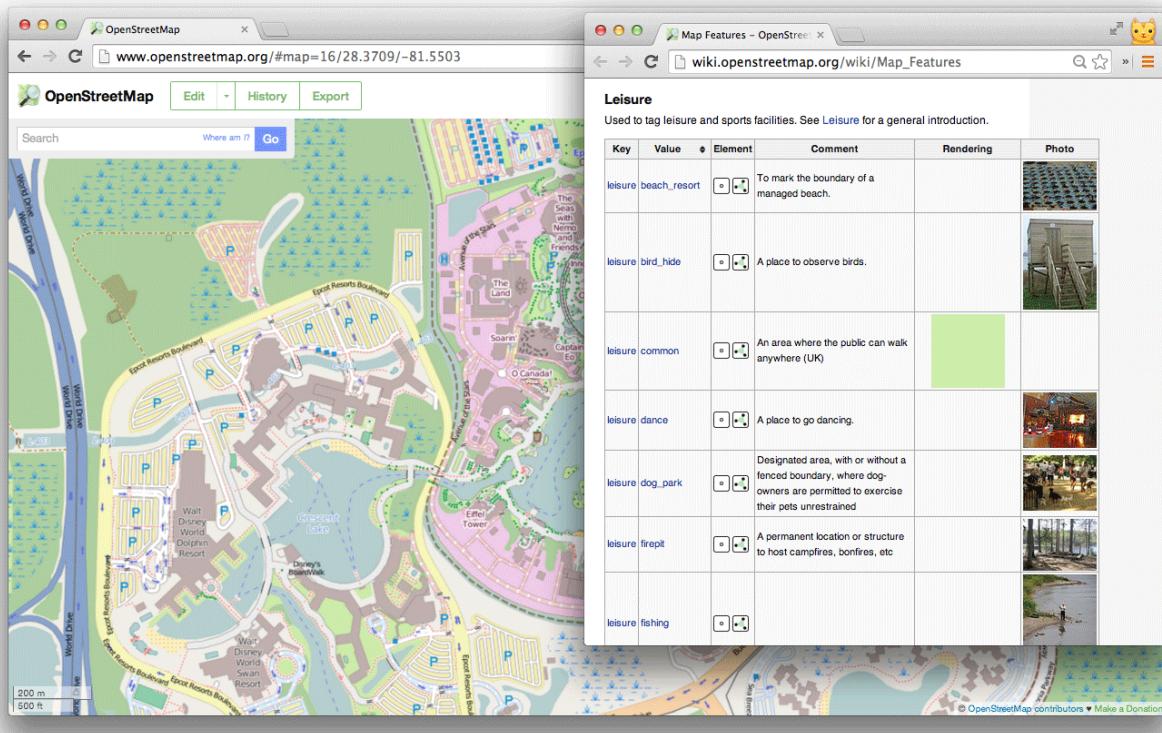


Figure 144: osm map features

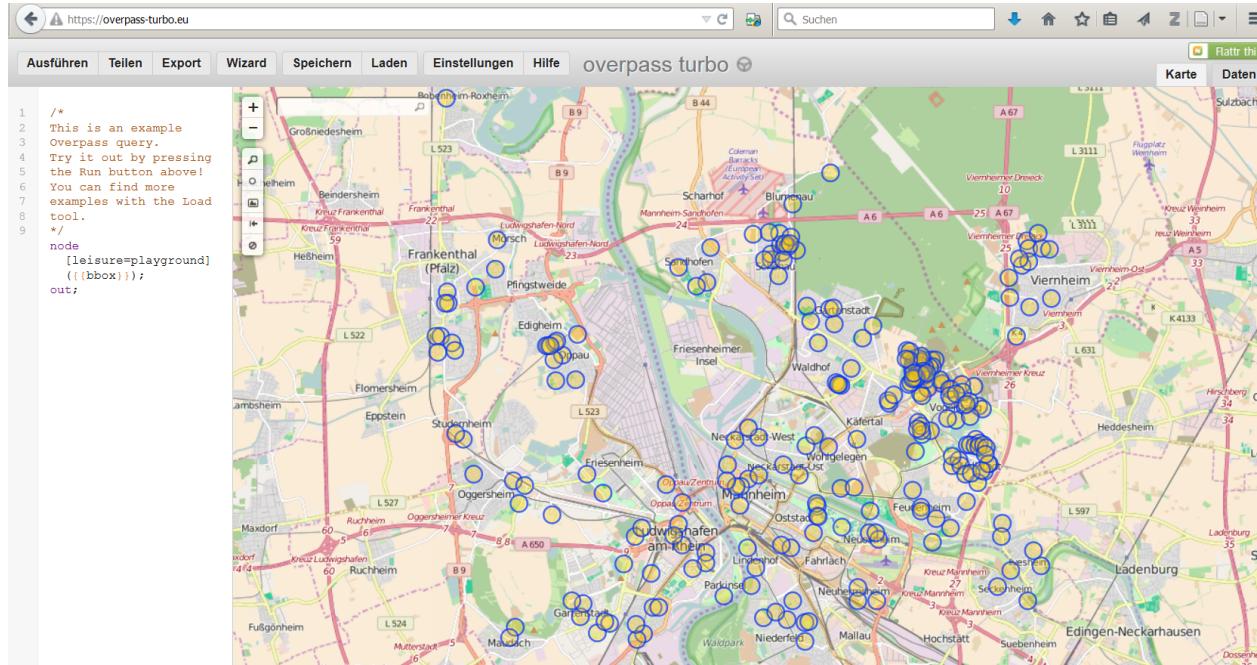


Figure 145: Spielplätze Mannheim

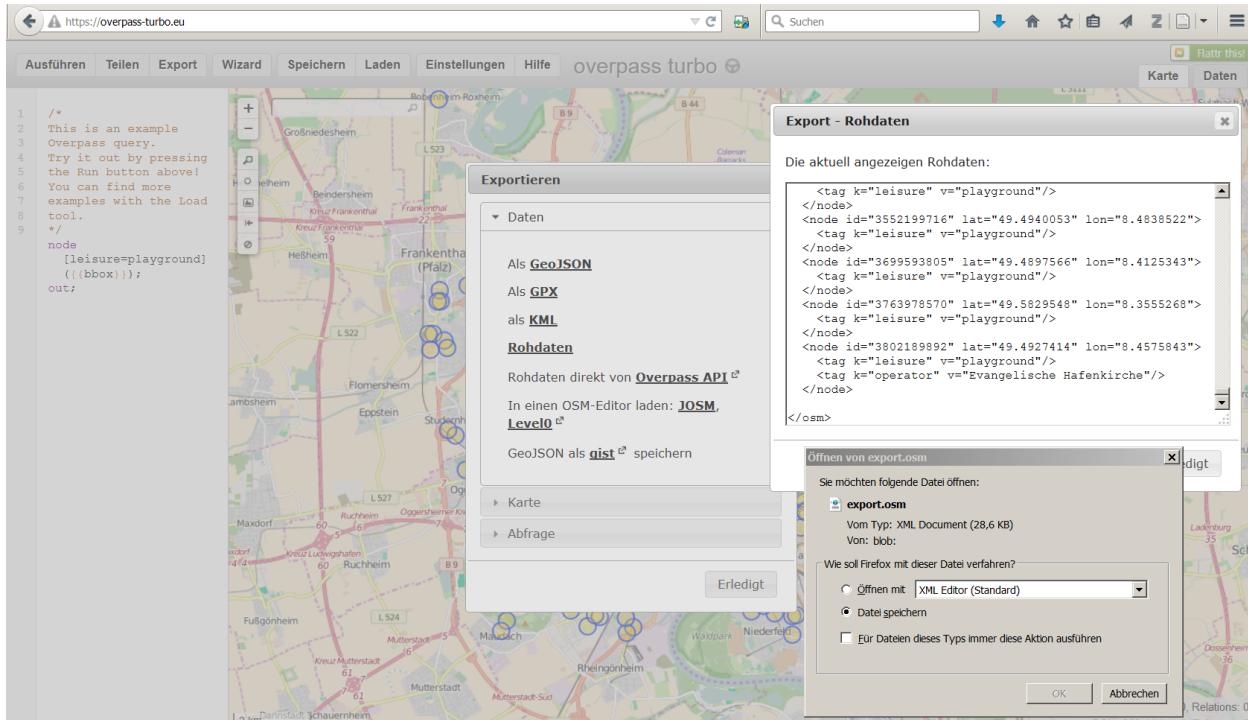


Figure 146: Export Rohdaten

```

<?xml version="1.0" encoding="UTF-8"?>
- <osm generator="Overpass API" version="0.6">
  <note>The data included in this document is from www.openstreetmap.org. The data is made available under ODbL.</note>
  <meta areas="2017-02-06T06:35:03Z" osm_base="2017-02-06T06:48:02Z"/>
  - <node lon="8.5028074" lat="49.5190994" id="30560755">
    <tag v="Potlatch 0.5d" k="created_by"/>
    <tag v="playground" k="leisure"/>
  </node>
  - <node lon="8.5393963" lat="49.4963345" id="76468450">
    <tag v="Potlatch 0.4a" k="created_by"/>
    <tag v="playground" k="leisure"/>
    <tag v="Rutsche, Schaukel, großer Sandkasten, Tischtennis" k="note"/>
  </node>
  - <node lon="8.5529589" lat="49.4967807" id="76468534">
    <tag v="playground" k="leisure"/>
  </node>
  - <node lon="8.5487501" lat="49.4923030" id="76468535">
    <tag v="playground" k="leisure"/>
  </node>

```

Figure 147: Spielplätze in Mannheim

```

> node_id[[1]]
      id
"30560755"
attr(,"class")
[1] "XMLAttributeValue"
> |

```

Figure 148: Erste node id

```
lat_x <- xpathApply(InfoList,
  "//tag[@v= 'playground']/parent::node/@ lon")
```

```
      lat
    "49.5190994"
attr(,"class")
[1] "XMLAttributeValue"
      lat
    "49.4963345"
attr(,"class")
[1] "XMLAttributeValue"
```

Figure 149: Latitude Koordinate

Paket auf Github

```
library(devtools)
install_github("Japhilko/gosmd")

library(gosmd)
pg_MA <- get_osm_nodes(object="leisure=playground",
                         "Mannheim")
info <- extract_osm_nodes(OSM.Data=pg_MA,
                           value="playground")
```

Ausschnitt der Ergebnisse

| | leisure | lat | lon | note |
|----------|------------|----------|----------|---|
| 30560755 | playground | 49.51910 | 8.502807 | NA |
| 76468450 | playground | 49.49633 | 8.539396 | Rutsche, Schaukel, großer Sandkasten, Tischtennis |
| 76468534 | playground | 49.49678 | 8.552959 | NA |
| 76468535 | playground | 49.49230 | 8.548750 | NA |
| 76468536 | playground | 49.50243 | 8.548140 | Schaukel, Rutsche, Sandkasten, Spielhaus, Tischtennis |
| 76468558 | playground | 49.49759 | 8.542036 | NA |

Link

- Tutorial zur Nutzung der Overpass API
- Vignette xml2

Mehr Beispiele, wie man mit XML Daten umgeht:

- Deborah Nolan - Extracting data from XML
- Duncan Temple Lang - A Short Introduction to the XML package for R

Noch mehr Informationen

- Web Daten manipulieren
- Tutorial zu xquery
- R und das Web (für Anfänger), Teil II: XML und R
- String Manipulation
- Nutzung, Vor- und Nachteile OSM
- Forschungsprojekte im Zusammenhang mit OpenStreetMap
- XML parsen - Stackoverflow
- Processing of GeoJson data in R

Die Pakete rvest und RCurl

Das Paket rvest

```
install.packages("rvest")  
  
• rvest - Easily Harvest (Scrape) Web Pages  
  
library(rvest)  
  
ht <- read_html('https://www.google.co.in/search?q=guitar+repair+workshop')  
links <- ht %>% html_nodes(xpath='//h3/a') %>% html_attr('href')  
gsub('/url\\?q=', '', sapply(strsplit(links[as.vector(grep('url', links))]), split='&'), '[', 1))  
  
## [1] "http://theguitarrepairworkshop.com/"  
## [2] "http://www.guitarservices.com/"  
## [3] "http://www.guitarrepairbench.com/guitar-building-projects/guitar-workshop/guitar-workshop-project"  
## [4] "https://www.facebook.com/The-Guitar-Repair-Workshop-847517635259712/"  
## [5] "https://www.taylorguitars.com/dealer/guitar-repair-workshop-ltd"  
## [6] "http://www.laweekly.com/music/10-best-guitar-repair-shops-in-los-angeles-4647166"  
## [7] "http://guitarworkshopglasgow.com/pages/repairs-1"  
## [8] "https://www.justdial.com/Delhi-NCR/Guitar-Repair-Services/nct-10988623"  
## [9] "https://www.justdial.com/Mumbai/Guitar-Repair-Services/nct-10988623"
```

Hin und weg

```
library(XML)  
url4<-"http://api.openstreetmap.org/api/0.6/node/25439439"  
obj4 <- xmlParse(url4)  
xpathApply(obj4, "//tag[@k = 'railway:station_category']")[[1]]  
  
## <tag k="railway:station_category" v="2"/>
```

- Wikipedia Artikel Bahnhofskategorien

| Stufe | Bahnsteigkanten | Bahnsteilänge | Reisende/Tag | Zughalte/Tag | Prozent | Kategorie |
|-------|-----------------|-----------------|-------------------|--------------|---------------|-----------|
| 6 | 1 | bis 90 m | bis 49 | bis 10 | > 90 % | 1 |
| 5 | 2 | > 90 bis 140 m | 50 bis 299 | 11 bis 50 | > 80 bis 90 % | 2 |
| 4 | 3 bis 4 | > 140 bis 170 m | 300 bis 999 | 51 bis 100 | > 60 bis 80 % | 3 |
| 3 | 5 bis 9 | > 170 bis 210 m | 1000 bis 9999 | 101 bis 500 | > 50 bis 60 % | 4 |
| 2 | 10 bis 14 | > 210 bis 280 m | 10.000 bis 49.999 | 501 bis 1000 | > 40 bis 50 % | 5 |
| 1 | ab 15 | > 280 m | ab 50.000 | ab 1001 | > 25 bis 40 % | 6 |
| | | | | | bis 25 % | 7 |

Figure 150:

Exkurs: Bahnhofskategorien

- rvest: Easily Harvest (Scrape) Web Pages

```
library(rvest)
bhfkat<-read_html(
  "https://de.wikipedia.org/wiki/Bahnhofskategorie")
df_html_bhfkat<-html_table(
  html_nodes(bhfkat, "table")[[1]], fill = TRUE)
```

Bahnhofskategorien Übersicht

| Stufe | Bahnsteigkanten | Bahnsteilänge | Reisende/Tag | Zughalte/Tag |
|-------|-----------------|-----------------|-------------------|--------------|
| 6 | 01 | > 000 bis 090 m | 00000 bis 00049 | 000 bis 0010 |
| 5 | 02 | > 090 bis 140 m | 00050 bis 00299 | 011 bis 0050 |
| 4 | 03 bis 04 | > 140 bis 170 m | 00300 bis 00999 | 051 bis 0100 |
| 3 | 05 bis 09 | > 170 bis 210 m | 01000 bis 09999 | 101 bis 0500 |
| 2 | 10 bis 14 | > 210 bis 280 m | 10.000 bis 49.999 | 501 bis 1000 |
| 1 | 00i ab 15 | > 280 m | 00000i ab 50.000 | 000i ab 1001 |

Links

- Scraping CRAN with rvest

Webscraping

Notwendige Pakete

```
install.packages("tidyverse")
```

```
library(tidyverse)
```



tidyverse 1.0.0

September 15, 2016 in [Packages](#), [tidyverse](#)

The tidyverse is a set of packages that work in harmony because they share common data representations and API design. The **tidyverse** package is designed to make it easy to install and load core packages from the tidyverse in a single command.

The best place to learn about all the packages in the tidyverse and how they fit together is [R for Data Science](#). Expect to hear more about the tidyverse in the coming months as I work on improved package websites, making [citation easier](#), and providing a common home for discussions about data analysis with the tidyverse.

Figure 151:

- R für DataScience

Weitere benötigte Pakete

- Das Paket `stringr`

```
library(stringr)
```

```
library(forcats)
library(ggmap)
library(rvest)
```

Daten von Wikipedia einsammeln

```
html.world_ports <- read_html("https://en.wikipedia.org/wiki/List_of_busiest_container_ports")
df.world_ports <- html_table(html_nodes(html.world_ports, "table"))[[2]], fill = TRUE)
```

```
library(DT)
datatable(df.world_ports)
```

Die Daten anschauen

```
glimpse(df.world_ports)
```

```
## Observations: 50
```

```

## Variables: 15
## $ Rank      <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16...
## $ Port      <chr> "Shanghai", "Singapore", "Shenzhen", "Ningbo-Zhoushan...
## $ Economy   <chr> "China", "Singapore", "China", "China", "Hong Kong", ...
## $ 2015[1]   <chr> "36,516", "30,922", "24,142", "20,636", "20,073", "19...
## $ 2014[2]   <chr> "35,268", "33,869", "23,798", "19,450", "22,374", "18...
## $ 2013[3]   <chr> "33,617", "32,240", "23,280", "17,351", "22,352", "17...
## $ 2012[4]   <chr> "32,529", "31,649", "22,940", "16,670", "23,117", "17...
## $ 2011[5]   <chr> "31,700", "29,937", "22,570", "14,686", "24,384", "16...
## $ 2010[6]   <chr> "29,069", "28,431", "22,510", "13,144", "23,532", "14...
## $ 2009[7]   <chr> "25,002", "25,866", "18,250", "10,502", "20,983", "11...
## $ 2008[8]   <chr> "27,980", "29,918", "21,414", "11,226", "24,248", "13...
## $ 2007[9]   <chr> "26,150", "27,932", "21,099", "9,349", "23,881", "13,...
## $ 2006[10]  <chr> "21,710", "24,792", "18,469", "7,068", "23,539", "12,...
## $ 2005[11]  <chr> "18,084", "23,192", "16,197", "5,208", "22,427", "11,...
## $ 2004[12]  <chr> "14,557", "21,329", "13,615", "4,006", "21,984", "11,...

```

Das Paket rvest

```

library(rvest)
ht <- read_html('https://www.google.co.in/search?q=guitar+repair+workshop')
links <- ht %>% html_nodes(xpath='//h3/a') %>% html_attr('href')
gsub('/url\\?q=', '', sapply(strsplit(links[as.vector(grep('url', links))]), split='&'), '[', 1))

## [1] "http://theguitarrepairworkshop.com/"
## [2] "http://www.guitarservices.com/"
## [3] "http://www.guitarrepairbench.com/guitar-building-projects/guitar-workshop/guitar-workshop-project"
## [4] "https://www.facebook.com/The-Guitar-Repair-Workshop-847517635259712/"
## [5] "https://www.taylorguitars.com/dealer/guitar-repair-workshop-ltd"
## [6] "http://www.laweekly.com/music/10-best-guitar-repair-shops-in-los-angeles-4647166"
## [7] "http://guitarworkshopglasgow.com/pages/repairs-1"
## [8] "https://www.justdial.com/Delhi-NCR/Guitar-Repair-Services/nct-10988623"
## [9] "https://www.justdial.com/Mumbai/Guitar-Repair-Services/nct-10988623"

```

Links

- How to really do an analysis in R (part 1, data manipulation)
- Read CSV From The Web
- Scraping CRAN with rvest

Use Case - Scraping Wikipedia

Einleitung

Im Folgenden werde ich zeigen, wie man Textinformationen aus Wikipedia herunterladen, verarbeiten und analysieren kann.

```

install.packages("NLP")
install.packages("tm")
install.packages("FactoMineR")

```

Die verwendeten Pakete

- Das R-Paket `stringi` von Marek Gagolewski und Bartek Tartanus bietet Möglichkeiten zur String Verarbeitung.

```
library("stringi")
```

- `tm` ist ein R-Paket um Text Mining zu realisieren. Es wurde von Ingo Feinerer, Kurt Hornik, und David Meyer geschrieben.

```
library("tm")
```

- Und schließlich brauchen wir das `FactoMineR`-Paket, das von Sébastien Le, Julie Josse und François Husson zur Durchführung der Hauptkomponentenanalyse erstellt wurde.

```
library("FactoMineR")
```

Die Text Daten herunterladen

- Als Beispiel verwenden wir Daten zu verschiedenen Krankheiten.
- In diesem Fall habe ich 7 deutsche Webseiten für Infektionskrankheiten ausgewählt.

```
wiki <- "http://de.wikipedia.org/wiki/"  
  
titles <- c("Zika-Virus", "Influenza-A-Virus_H1N1",  
          "Spanische_Grippe", "Influenzavirus",  
          "Vogelgrippe_H5N1",  
          "Legionellose-Ausbruch_in_Warstein_2013",  
          "Legionellose-Ausbruch_in_Jülich_2014")
```

Das Herunterladen der Seiten

- Zunächst wird ein Container erstellt um die Ergebnisse abzuspeichern
- Dann wird der Text für jeden Artikel heruntergeladen und in dem Container gespeichert.

```
articles <- character(length(titles))  
  
for (i in 1:length(titles)){  
  articles[i] <- stri_flatten(  
    readLines(stri_paste(wiki, titles[i])), col = " ")  
}  
  
docs <- Corpus(VectorSource(articles))
```

Die Daten vorbereiten

Das Folgende basiert auf einem Blogpost von Norbert Ryciak über die automatische Kategorisierung von Wikipedia-Artikeln.

- Eine Fehlermeldung ist aufgetreten, als ich den Code ausgewertet habe.
- Es war möglich, dieses Problem mit Hinweisen aus einer Diskussion auf Stackoverflow zu lösen.

```

docs2 <- tm_map(docs, function(x) stri_replace_all_regex(
  x, "<.+?>", " "))
docs3 <- tm_map(docs2, function(x) stri_replace_all_fixed(
  x, "\t", " "))

```

Den Text weiterverarbeiten

```

docs4 <- tm_map(docs3, PlainTextDocument)
docs5 <- tm_map(docs4, stripWhitespace)
docs6 <- tm_map(docs5, removeWords, stopwords("german"))
docs7 <- tm_map(docs6, removePunctuation)
docs8 <- tm_map(docs7, tolower)
# docs8 <- tm_map(docs8, PlainTextDocument)

dtm <- DocumentTermMatrix(docs8)

```

Principal Component Analysis

- Der folgende Code ist auf einem Blog post von Arthur Charpentier über das Mining von Wikipedia basiert.

```

dtm2 <- as.matrix(dtm)
frequency <- colSums(dtm2)
frequency <- sort(frequency, decreasing=TRUE)
words <- frequency[frequency>20]
s <- dtm2[,which(colnames(dtm2) %in% names(words))]

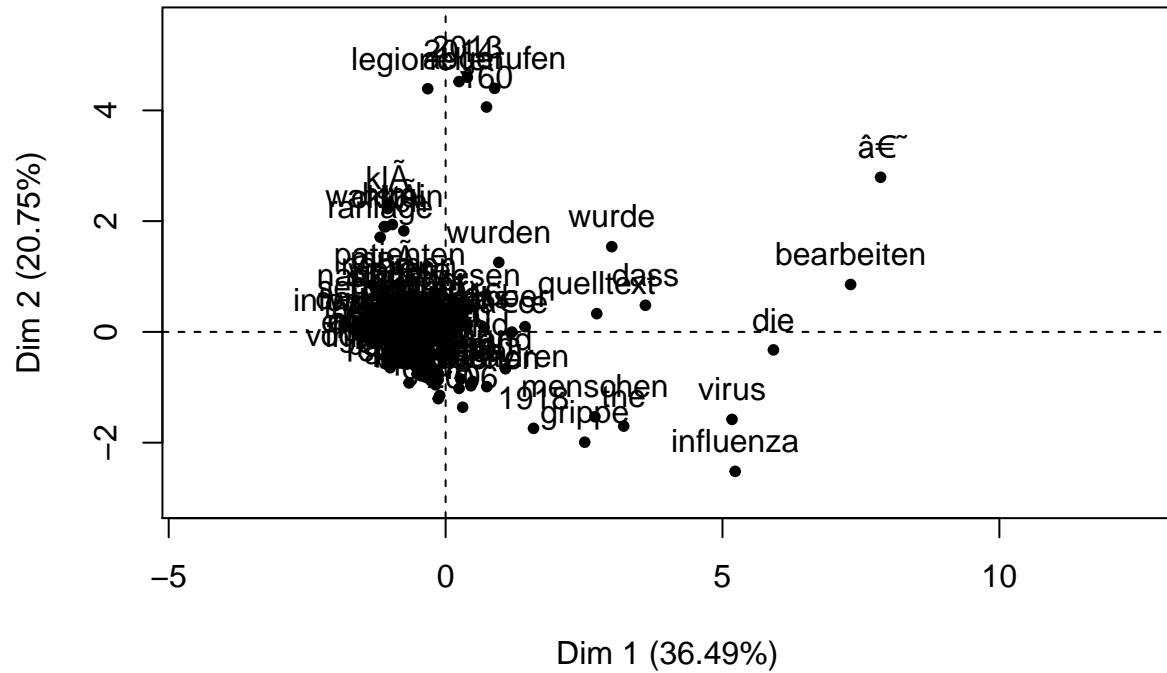
for(i in 2:nrow(dtm2)){
  s <- cbind(s,dtm2[i,which(colnames(dtm2) %in%
                                names(words))])
}
colnames(s) <- titles

```

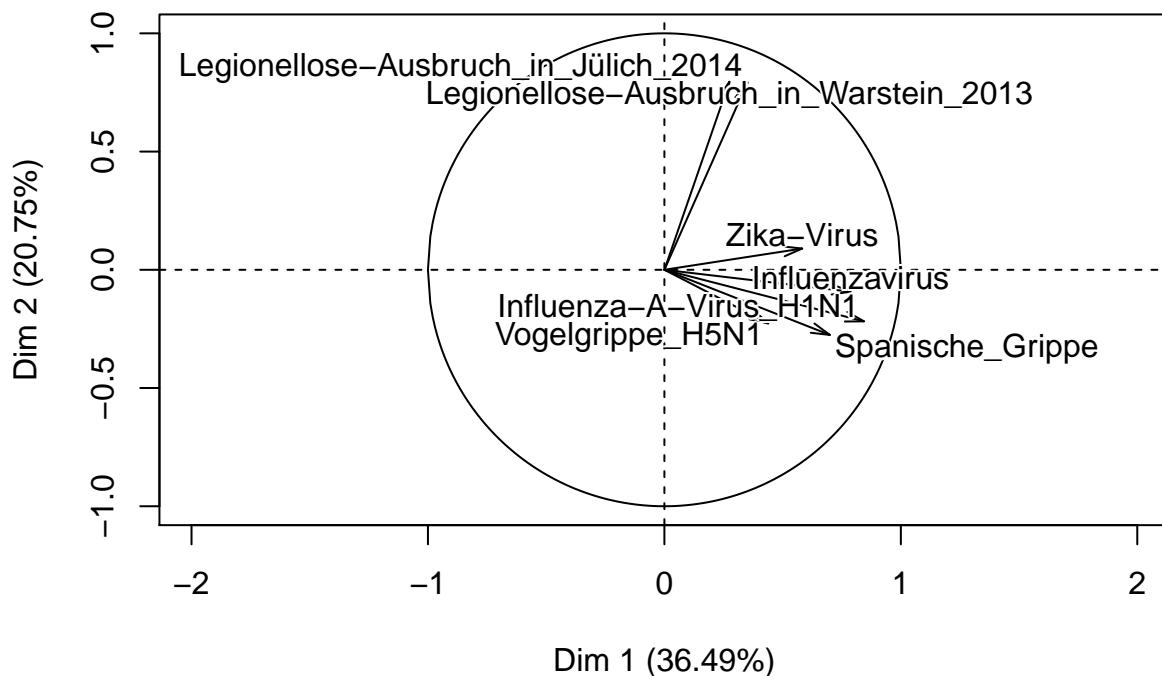
Ergebnis

```
PCA(s)
```

Individuals factor map (PCA)



Variables factor map (PCA)



```
## **Results for the Principal Component Analysis (PCA)**
## The analysis was performed on 125 individuals, described by 7 variables
## *The results are available in the following objects:
##
##      name           description
## 1  "$eig"          "eigenvalues"
## 2  "$var"          "results for the variables"
## 3  "$var$coord"   "coord. for the variables"
## 4  "$var$cor"     "correlations variables - dimensions"
## 5  "$var$cos2"    "cos2 for the variables"
## 6  "$var$contrib" "contributions of the variables"
## 7  "$ind"          "results for the individuals"
## 8  "$ind$coord"   "coord. for the individuals"
## 9  "$ind$cos2"    "cos2 for the individuals"
## 10 "$ind$contrib" "contributions of the individuals"
## 11 "$call"         "summary statistics"
## 12 "$call$centre"  "mean of the variables"
## 13 "$call$ecart.type" "standard error of the variables"
## 14 "$call$row.w"   "weights for the individuals"
## 15 "$call$col.w"   "weights for the variables"
```

Ergebnis

- In der Factor Map sehen wir das erwartete Ergebnis.
- Die Seiten zur Legionellen Krankheit sind sehr nah beianander, während die Seiten zur Influenza in

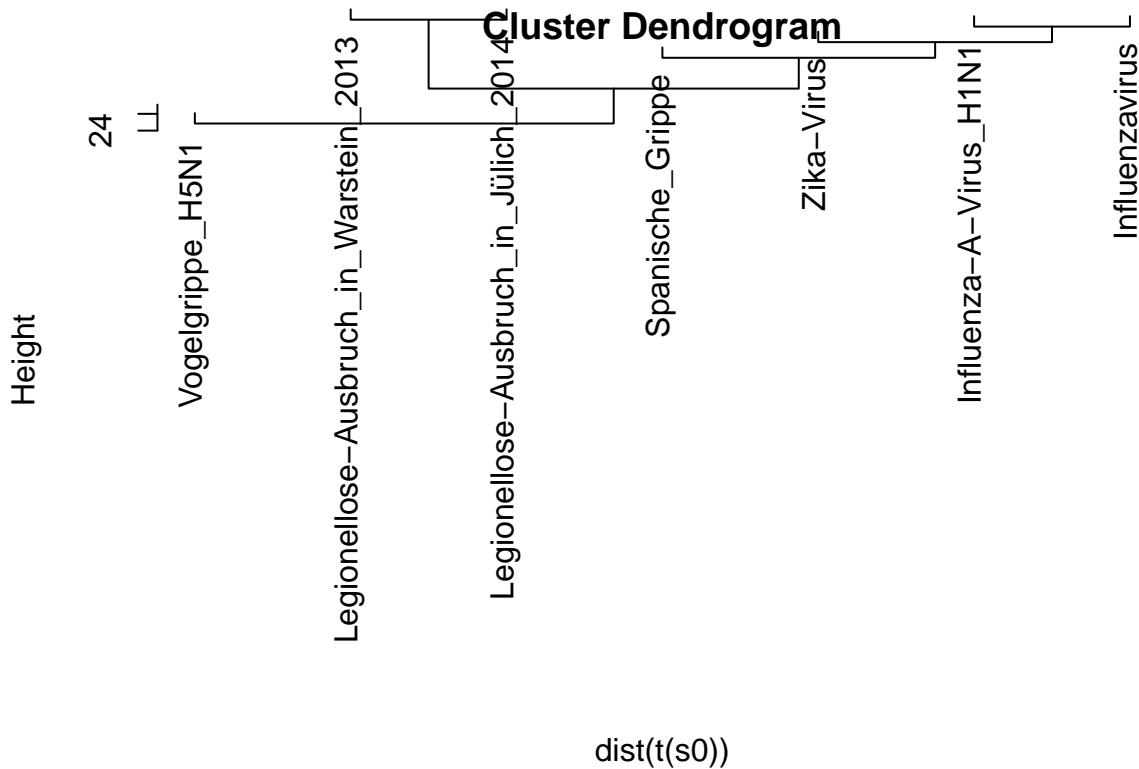
einem anderen Teil sind.

Das Dendrogramm

- Im Folgenden wird die Normalisierung durchgeführt und die Ergebnisse werden geplottet.

```
s0 <- s/apply(s,1,sd)
h <- hclust(dist(t(s0)), method = "ward")

plot(h, labels = titles, sub = "")
```



- Youtube Video zu Text Mining

Shiny Apps

Das shiny Paket installieren

```
install.packages("shiny")
```

Wer hat's erfunden?

```
citation("shiny")
```



Figure 152:

```
##  
## To cite package 'shiny' in publications use:  
##  
##   Winston Chang, Joe Cheng, JJ Allaire, Yihui Xie and Jonathan  
##   McPherson (2017). shiny: Web Application Framework for R. R  
##   package version 1.0.0. https://CRAN.R-project.org/package=shiny  
##  
## A BibTeX entry for LaTeX users is  
##  
##   @Manual{,  
##     title = {shiny: Web Application Framework for R},  
##     author = {Winston Chang and Joe Cheng and JJ Allaire and Yihui Xie and Jonathan McPherson},  
##     year = {2017},  
##     note = {R package version 1.0.0},  
##     url = {https://CRAN.R-project.org/package=shiny},  
##   }
```

Eine erste Beispielapp

```
library(shiny)  
runExample("01_hello")
```

Der Start

Dem Kind einen Namen geben

Die erste App

- man muss den Run App Button drücken

Hello Shiny!

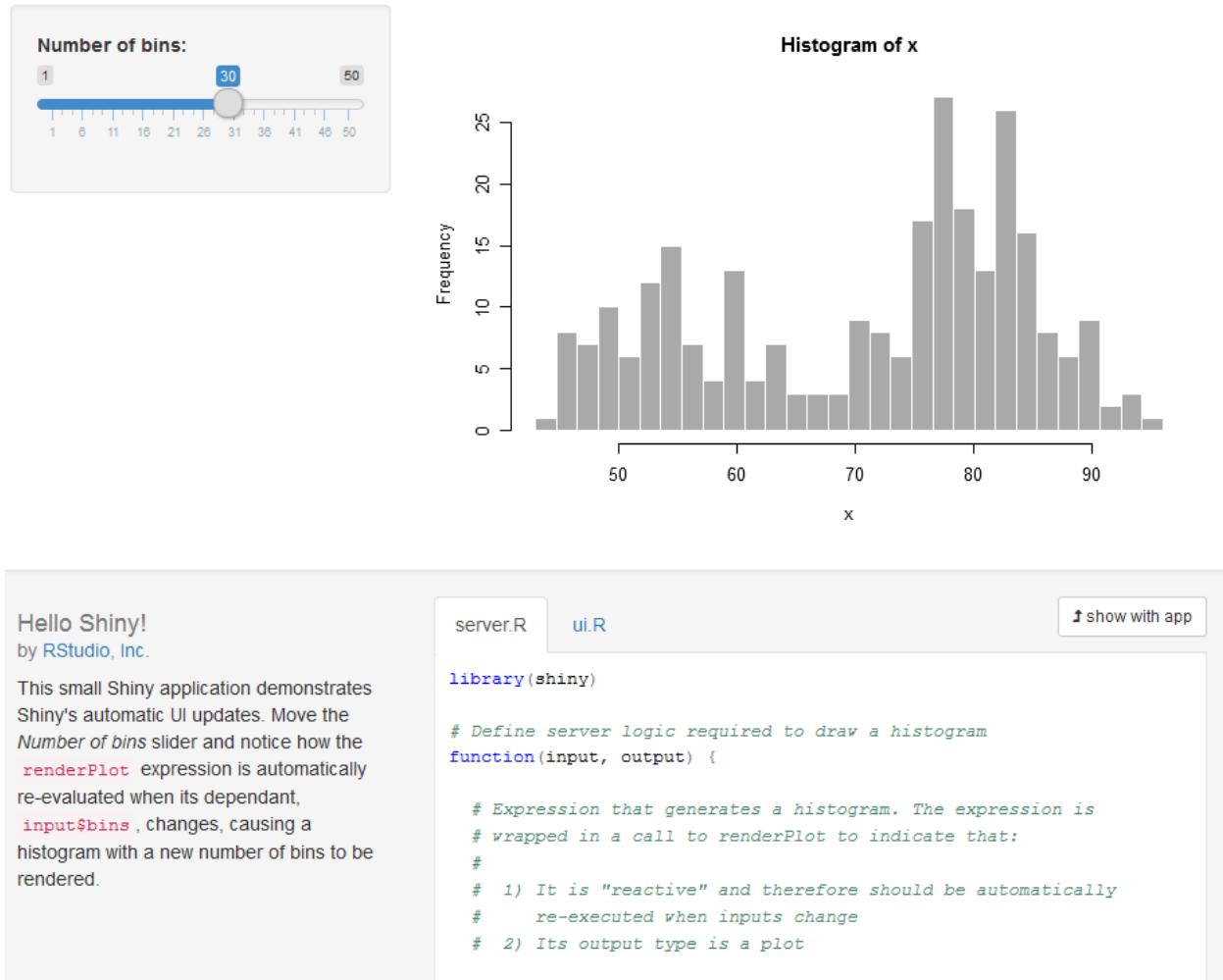


Figure 153:

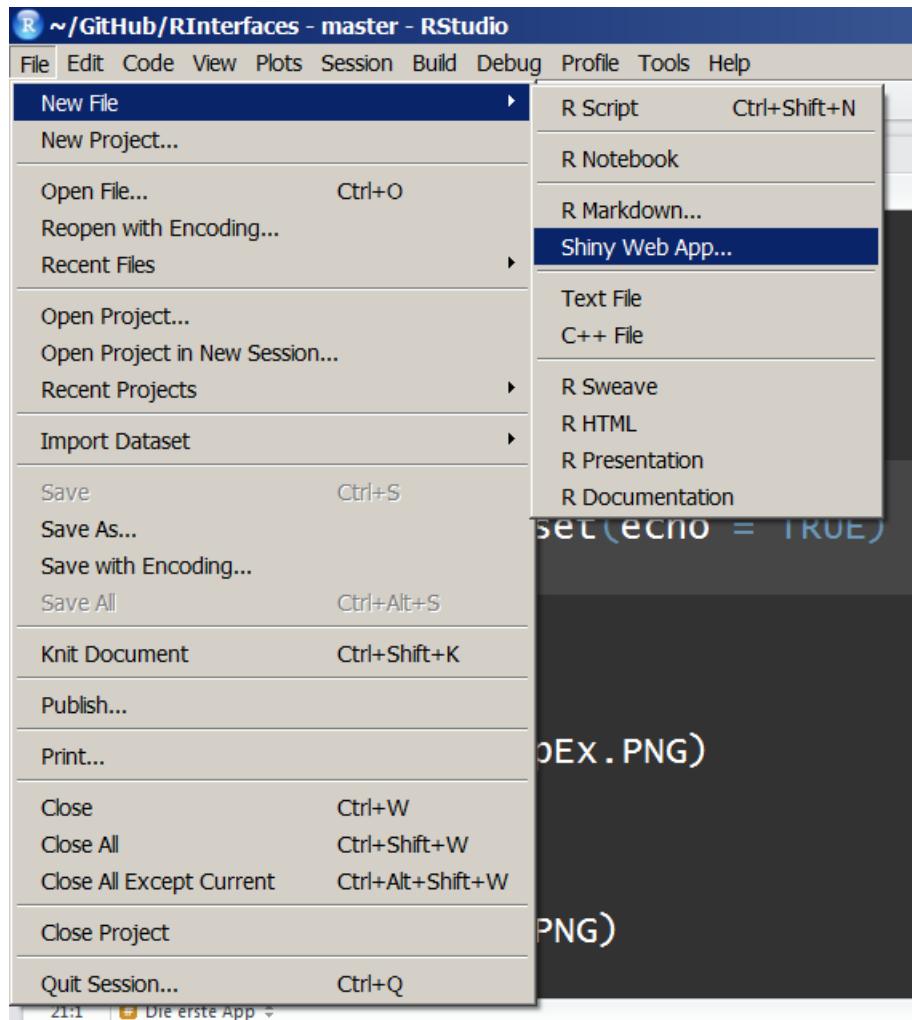


Figure 154:

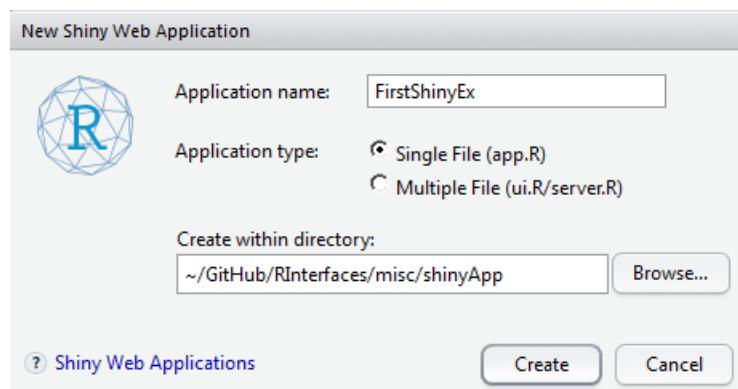


Figure 155:



Figure 156:

- Das Ergebnis:

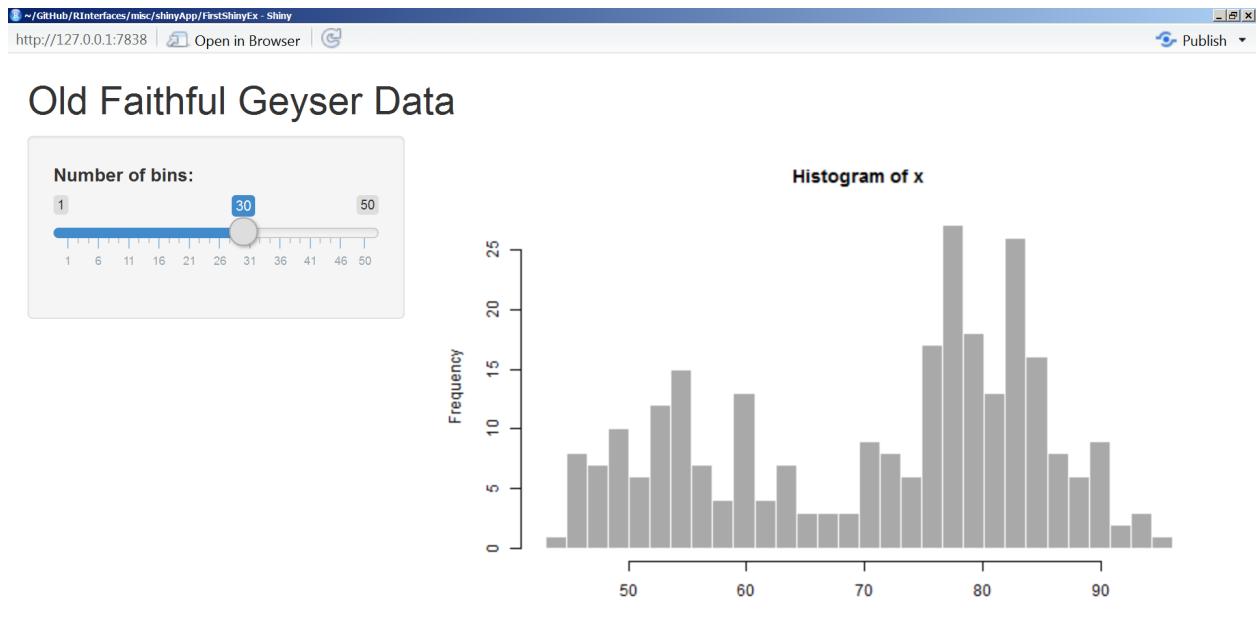


Figure 157:

Zur Erklärung

- Zumeist arbeitet man mit mindestens zwei Dateien
- Das user interface wird mit einer Datei erzeugt werden, die **ui.R** genannt werden muss
- Für die Server Seite brauchen wir auch ein eigenes File, dieses benennen wir mit **server.R**

Eine zweite Beispiel App

```
library(shiny)
runExample("02_text")
```

Einführung in Shiny

Links

- Eine Dashboard App erzeugen

The screenshot shows a Shiny application window titled "Shiny Text - Mozilla Firefox". On the left, there is a sidebar with two input fields: "Choose a dataset:" containing "rock" and "Number of observations to view:" containing "10". To the right, there are two sections: a summary statistics table and a data table. The summary statistics table has columns for "area", "peri", "shape", and "perm", with rows for Min., 1st Qu., Median, Mean, 3rd Qu., and Max. The data table below it has four columns: "area", "peri", "shape", and "perm", with 10 rows of data.

| | area | peri | shape | perm |
|----|------|---------|-------|--------|
| 1 | 4990 | 2791.90 | 0.09 | 6.30 |
| 2 | 7002 | 3892.60 | 0.15 | 6.30 |
| 3 | 7558 | 3930.66 | 0.18 | 6.30 |
| 4 | 7352 | 3869.32 | 0.12 | 6.30 |
| 5 | 7943 | 3948.54 | 0.12 | 17.10 |
| 6 | 7979 | 4010.15 | 0.17 | 17.10 |
| 7 | 9333 | 4345.75 | 0.19 | 17.10 |
| 8 | 8209 | 4344.75 | 0.16 | 17.10 |
| 9 | 8393 | 3682.04 | 0.20 | 119.00 |
| 10 | 6425 | 3098.65 | 0.16 | 119.00 |

Figure 158:

Teach yourself Shiny

The How to Start Shiny video series will take you from R programmer to Shiny developer. Watch the complete tutorial here, or jump to a specific chapter by clicking a link below. The entire tutorial is two hours and 25 minutes long.

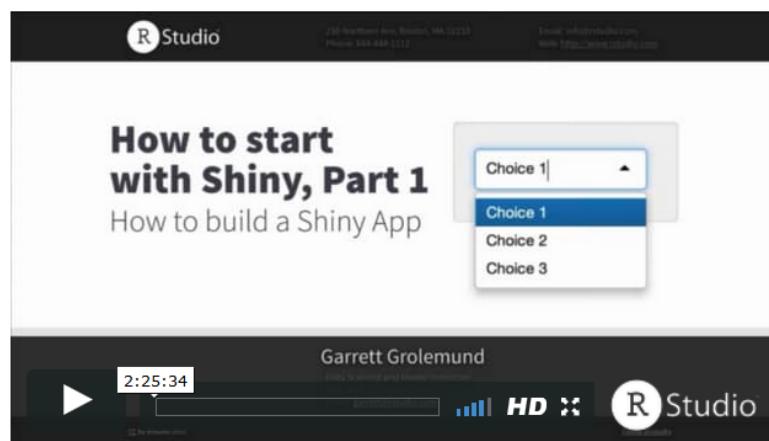


Figure 159:

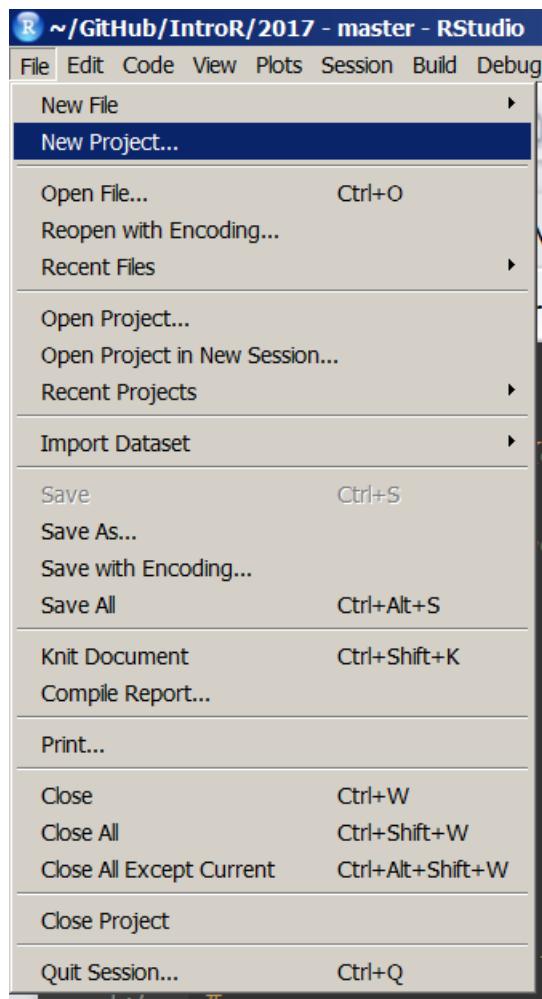


Figure 160:

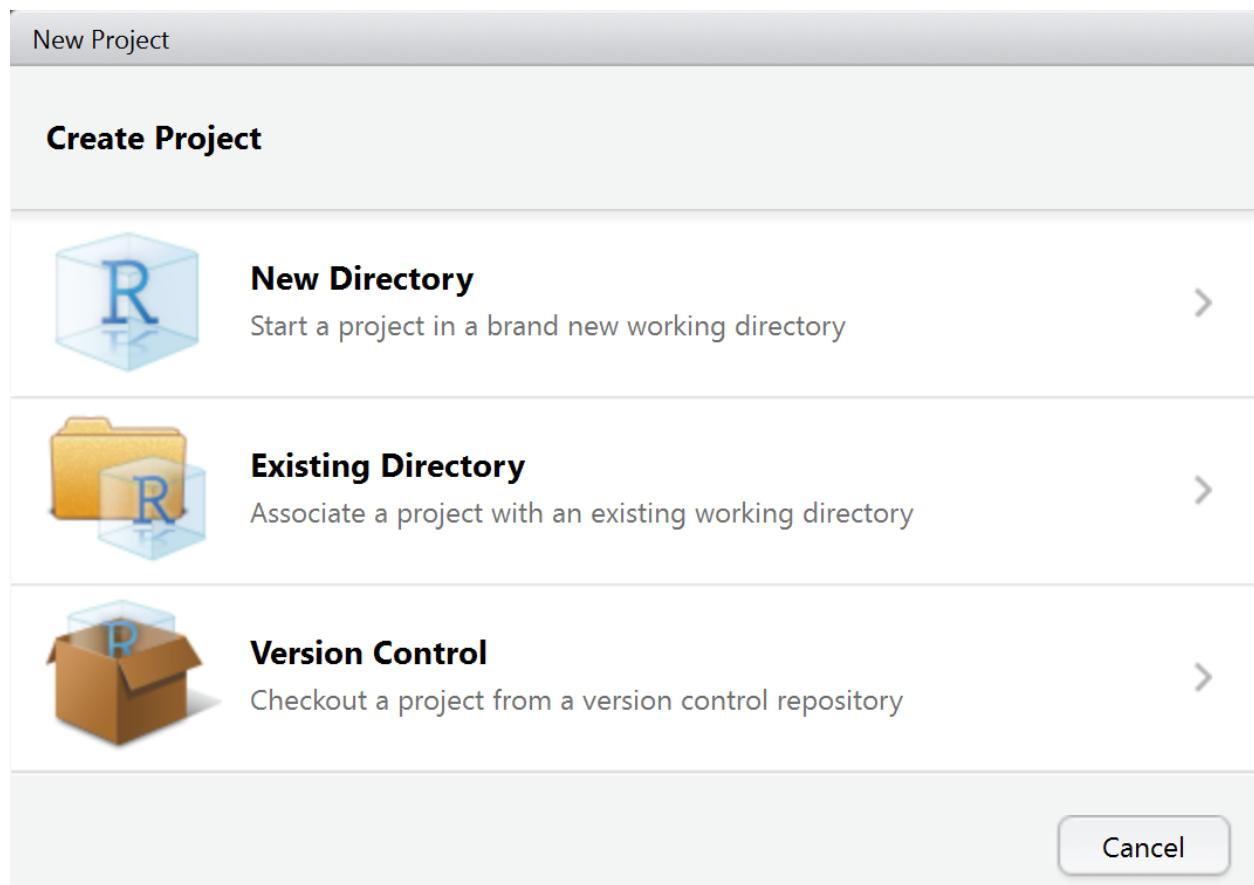


Figure 161:

R und Git

Rstudio und git - ein Projekt anlegen

Ein Projekt mit Versionskontrolle

Auswahl Versionskontrolle

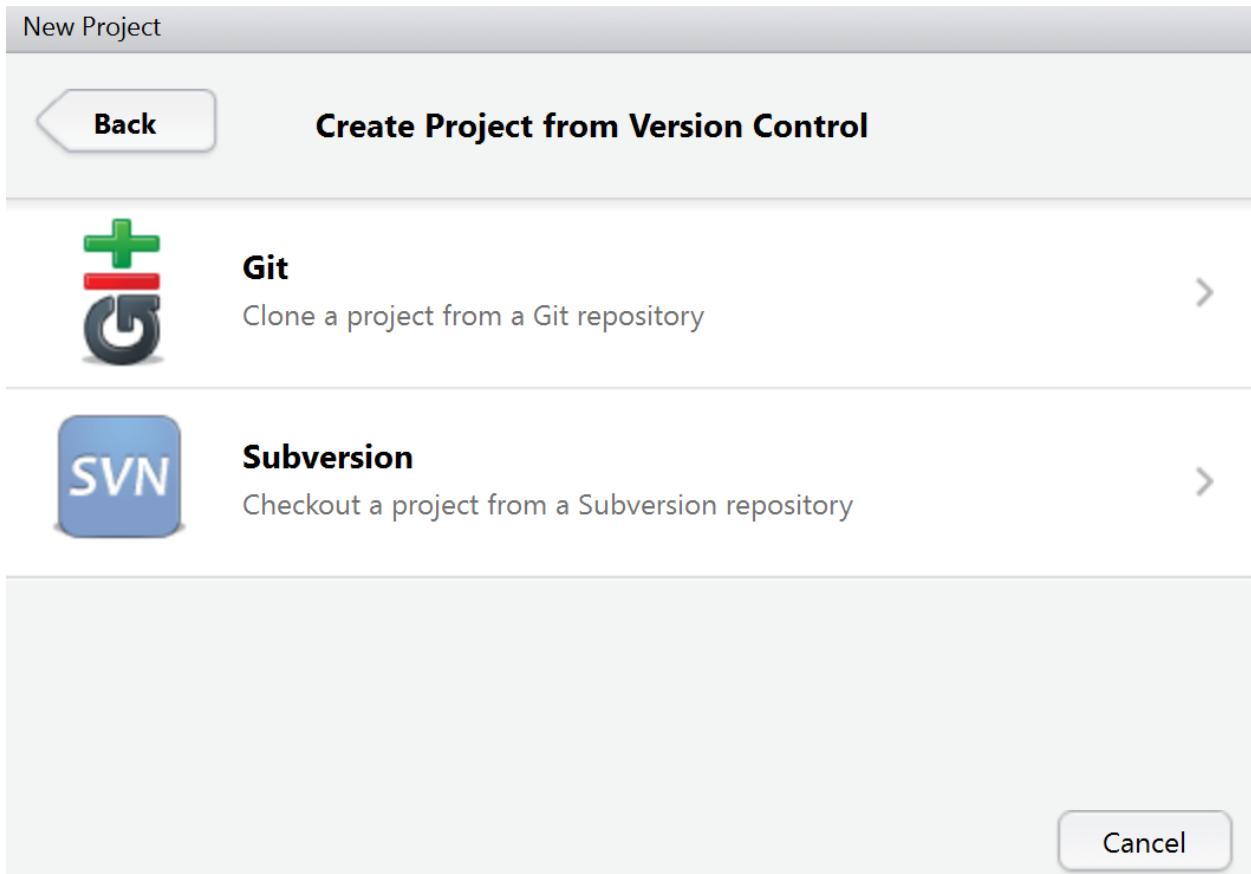


Figure 162:

Ein Projekt clonen

Der git-Reiter in Rstudio

Aktuelle eigene Änderungen committen

Der übliche Ablauf

- Datei bearbeiten und speichern
- Änderungen commiten
- Änderungen von anderen ziehen (pull)
- Eigene Änderung hochladen (push)

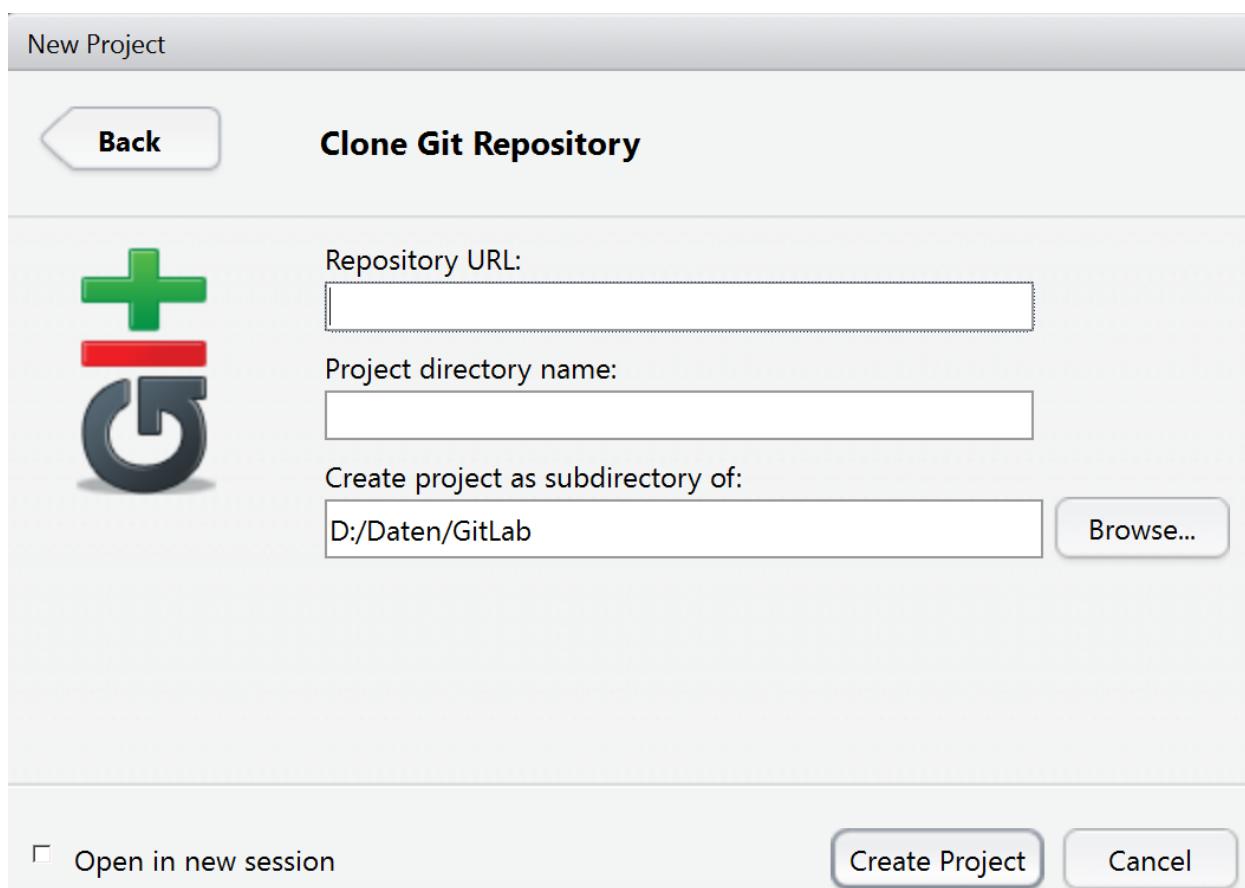


Figure 163:

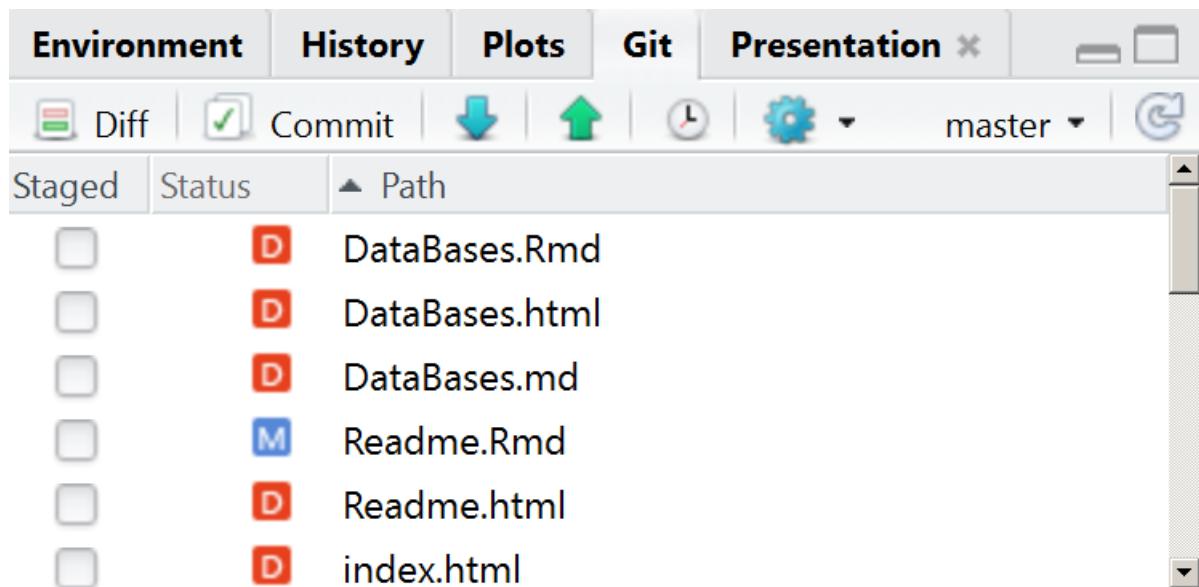


Figure 164:

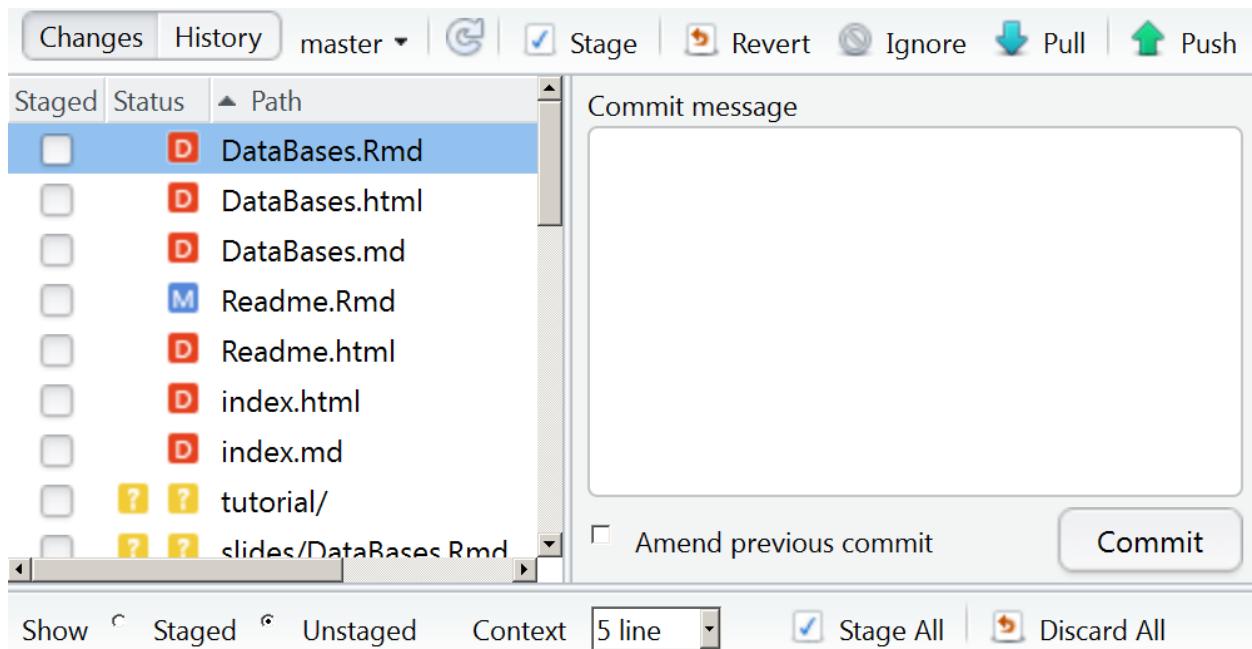


Figure 165:

Links

- Commit failed - git shell
- Git cheatsheet

Commands

`git commit`

`git push`

<http://stackoverflow.com/questions/1125968/force-git-to-overwrite-local-files-on-pull>

Problems with disk space

WinDirStat <https://support.microsoft.com/de-de/kb/912997> <http://www.pcwelt.de/tipps/Update-Dateien-loeschen-8357046.html>

Quelle für Pakete

Ein Paket von Github installieren

```
install.packages("devtools")
library(devtools)
install_github("Japhilko/gosmd")
```

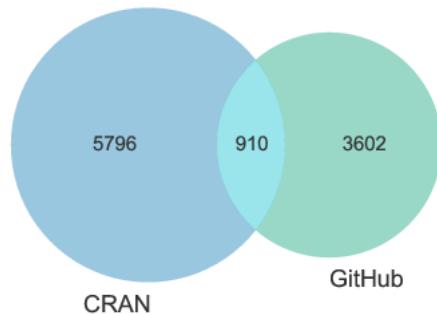


Figure 166:

Datensätze Suchfunktion

Japhilko / GeoData

Code Issues 1 Pull requests 0 Projects 0 Wiki Pulse Graphs Settings

Branch: master → GeoData / data / BLA_tab.csv Find file Copy path

Japhilko scraping 5ffce56 on 29 Mar 2016

1 contributor

18 lines (17 sloc) | 866 Bytes Raw Blame History

| | Land | Fläche (km ²) | Einwohner | Einw. je km ² | Hauptstadt |
|---|-------------------|---------------------------|-----------|--------------------------|------------|
| 2 | Baden-Württemberg | 35751.36 | 10569111 | 296 | Stuttgart |
| 3 | Bayern | 70550.23 | 12519571 | 177 | München |
| 8 | Hessen | 21114.93 | 6016481 | 285 | Wiesbaden |

Figure 167:

Git und Rstudio

Links

- Using github and rstudio
- How do I tell Git for Windows where to find my private RSA key
- Reset local repository branch to be just like remote repository HEAD
- How I Manage Data Analysis Projects with RStudio and Git - Part 1
- How do I force “git pull” to overwrite local files?

Version Control, File Sharing, and Collaboration Using GitHub and RStudio

January 29, 2017

By geraldbelton



(This article was first published on [R – Gerald Belton](#), and kindly contributed to [R-bloggers](#))

102 SHARES

f Share

t Tweet

This is Part 3 of our “Getting Started with R Programming” series. For previous articles in the series, click here: [Part 1](#), [Part 2](#).

This week, we are going to talk about using git and GitHub with RStudio to manage your projects.

Figure 168:

C++ Integration - Überblick über das Paket rcpp

Warum die Integration von c++

Robert Gentleman, in R Programming for Bioinformatics, 2008, about R's built-in C interfaces:

Since R is not compiled, in some situations its performance can be substantially improved by writing code in a compiled language. There are also reasons not to write code in other languages, and in particular we caution against premature optimization, prototyping in R is often cost effective. And in our experience very few routines need to be implemented in other languages for efficiency reasons. Another substantial reason not to use an implementation in some other language is increased complexity. The use of another language almost always results in higher maintenance costs and less stability. In addition, any extensions or enhancements of the code will require someone that is proficient in both R and the other language.

- Rcpp does make some of the above caution statements slightly less critical.

Warum und wann?

- Warum? - R wird langsam oder hat Probleme bei der Speicherverwaltung: zum Beispiel bei Schleifen, die nicht vektorisiert werden können.
- Wann? - wenn man es mit Rcode nicht besser hinbekommt und man den langsamen Code identifiziert hat.

Voraussetzung Compiler

Für Windows, Rtools

1. <http://cran.r-project.org/bin/windows/Rtools/>

2. <http://cran.r-project.org/doc/manuals/R-admin.html#The-Windows-toolset>

Für Mac, Xcode

3. http://cran.r-project.org/doc/manuals/R-admin.html#Installing-R-under-_0028Mac_0029-OS-X

4. <http://cran.r-project.org/doc/manuals/R-admin.html#Mac-OS-X>

Was wir nutzen werden

Wir werden die folgenden beiden Pakete nutzen:

- `inline` und die `cfunction` um Inline C code zu schreiben, der on-the-fly kompiliert wird (Es gibt auch eine `cxxfunction` für C++ Code).
- `Rcpp`, und die Nutzung der Funktion `cppFunction`

Rcpp ist das am meisten heruntergeladene Paket

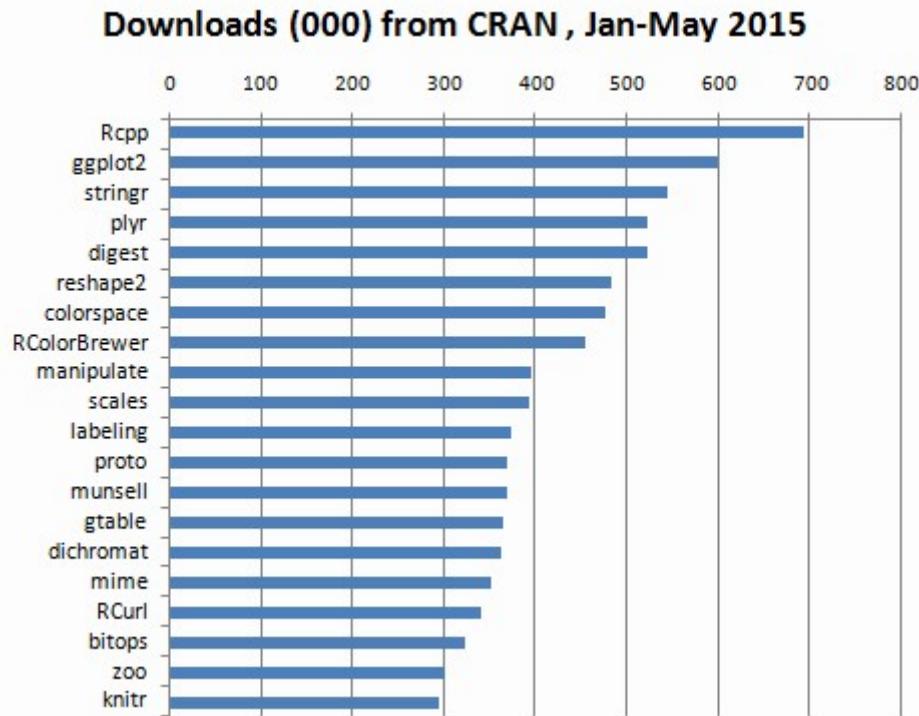


Figure 169:

Das Paket `inline`

```
install.packages("inline")
```

```
library(inline)
citation("inline")
```

```

## To cite package 'inline' in publications use:
##
##   Oleg Sklyar, Duncan Murdoch, Mike Smith, Dirk Eddelbuettel,
##   Romain Francois and Karline Soetaert (2015). inline: Functions
##   to Inline C, C++, Fortran Function Calls from R. R package
##   version 0.3.14. https://CRAN.R-project.org/package=inline
##
## A BibTeX entry for LaTeX users is
##
## @Manual{,
##   title = {inline: Functions to Inline C, C++, Fortran Function Calls from R},
##   author = {Oleg Sklyar and Duncan Murdoch and Mike Smith and Dirk Eddelbuettel and Romain Francois},
##   year = {2015},
##   note = {R package version 0.3.14},
##   url = {https://CRAN.R-project.org/package=inline},
## }
##
## ATTENTION: This citation information has been auto-generated from
## the package DESCRIPTION file and may need manual editing, see
## 'help("citation")'.

```

Das Rcpp Paket

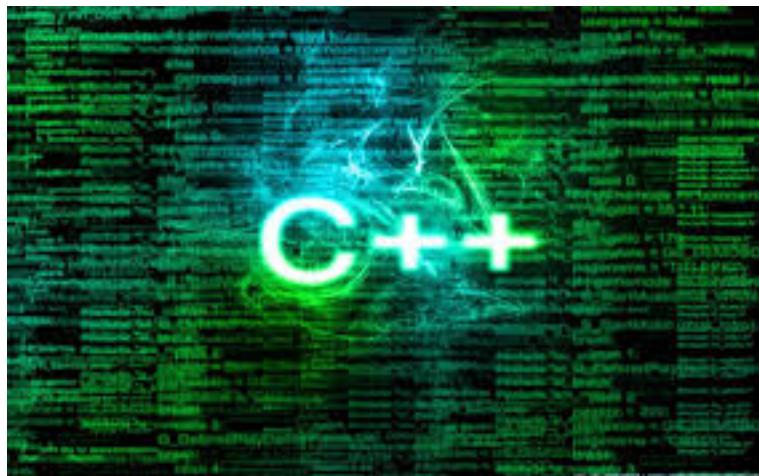


Figure 170:

- Dirk Eddelbuettel und Romain Francois, unter Mitwirkung von Douglas Bates, John Chambers und JJ Allaire.
- Eine flexible Umgebung das die Integration von R und C/C++ ermöglicht
- <http://www.rcpp.org/>
- Mit dabei ist die Dokumentation und Beispiele:

```
vignette(package = "Rcpp")
```

- Alle Basis R Typen sind als C++ Klassen integriert.
- Man muss sich keine Sorgen über garbage collection machen.

Was ist garbage collection

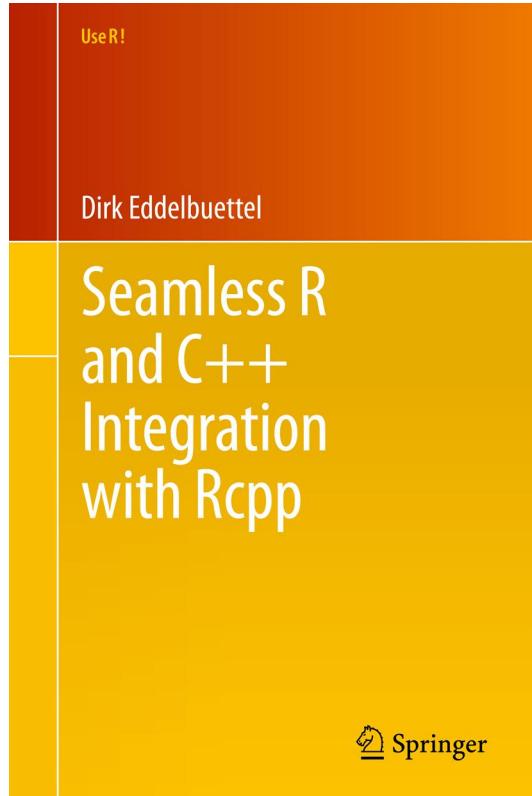
- Mit dem Befehl `gc()` werden Variablen gelöscht, die nicht mehr benötigt werden/auf die man keinen Zugriff mehr hat.
- Keine Variablen werden gelöscht, die man noch benötigt

Verwandte Pakete

- `RcppArmadillo` - Armadillo basiert auf einem C++ Paket für lineare Algebra.
- `RcppEigen` - hoch-performante Eigen lineare Algebra Bibliothek.
- `RInside` - R aus C++ heraus nutzen.

Einleitung

- R ist in C geschrieben



- Die Nutzung der Schnittstelle zu C liegt nahe

Das R-Paket CPP

- R Simulationsmodelle bis zu 20 mal schneller
- Hohe Performanz mit Rcpp

```
install.packages("Rcpp")
```

```
library(Rcpp)
cppFunction('int add(int x, int y, int z) {
  int sum = x + y + z;
}
```

```

    return sum;
}')
# add works like a regular R function
add

add(1, 2, 3)

```

Rcpp

Tutorial on Rcpp by Hadley Wickham

```
library(Rcpp)
```

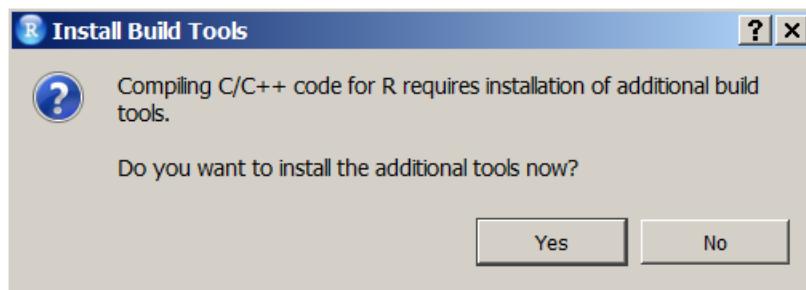


Figure 171:

```

cppFunction('int add(int x, int y, int z) {
  int sum = x + y + z;
  return sum;
}')

add(1, 2, 3)

```

Benchmarking

```

install.packages("microbenchmark")

library(microbenchmark)

```

- R-bloggers Artikel zu dem Paket `microbenchmark`

Das Paket `rbenchmark`

```

install.packages("rbenchmark")

library(rbenchmark)

```

Eine cpp Funktion zum aufsummieren

```

library(Rcpp)
cppFunction('

```

```

double sumC(NumericVector x) {
    int n = x.size();
    double total = 0;
    for(int i = 0; i < n; ++i) {
        total += x[i];
    }
    return total;
}

```

Ein erster Vergleich

```

y <- rnorm(10000000)
microbenchmark(
  sumC(y),
  sum(y)
)

> library(Rcpp)
> y <- rnorm(10000000)
> microbenchmark(
+   sumC(y),
+   sum(y)
+ )
Unit: milliseconds
expr      min       lq     mean   median      uq     max neval cld
sumC(y) 17.85742 18.34534 20.36847 18.98247 22.67203 28.78102    100    a
sum(y)  17.88387 19.35946 21.19280 19.61960 23.38337 29.48921    100    a

```

Figure 172:

Resourcen

Youtube Videos

- ... mit Dirk Edelbuettel
- ... R mit C++: Rcpp, RInside, und RProtobuf
- Oliver Heidmann - Programmieren in R - Rcpp
- Das Paket `cxxfunplus`

```
install.packages("cxxfunplus")
```

- Rcpp: Seamless R and C++ Integration
- Rcpp Honig
- Advanced R on Memory
- C++ sugar

Datenbanken und R

Was sind Datenbanken?

- Eine Datenbank, auch Datenbanksystem (DBS) genannt, ist ein System zur elektronischen Datenverwaltung.
- Die wesentliche Aufgabe eines DBS ist es, große Datenmengen effizient, widerspruchsfrei und dauerhaft zu speichern.

Wann sollte man R um Datenbanken ergänzen?

- Wenn ein Datensatz in den Arbeitsspeicher passt, gibt es keinen Grund eine Datenbank zu nutzen.
- die Nutzung von Schnittstellen zu Datenbanken macht den Code nur komplizierter und langsamer

Man nutzt die Schnittstelle zu Datenbanken,...

- weil die Daten bereits in einer Datenbank vorgehalten werden
- oder weil der Datensatz nicht in den Arbeitsspeicher passt

Die drei großen Open-Source Datenbanken

- SQLite, MySQL und PostgreSQL
- für alle drei gibt es Anbindungen in R

```
install.packages("RSQLite")
install.packages("RMySQL")
install.packages("RpostgreSQL")
```

- in der Folge liegt der Fokus vor allem auf SQLite und PostgreSQL

SQLite

- SQLite - Open Source Programmzbibliothek mit relationalem Datenbanksystem
- SQLite ist eine schlanke Datenbank und man muss nichts weiter installieren um sie zu nutzen.
- SQLite ist schon in R und Python integriert

MySQL Datenbank

- Grundlage für dynamische Webauftritte
- am meisten verbreitetes Datenbanksystem
- allerdings schwieriger zu konfigurieren als die anderen SQL Datenbanken SQLite und PostgreSQL

PostgreSQL

- PostgreSQL ist in den meisten Linux-Distributionen enthalten.
- Bietet eine Schnittstellen zu vielen Programmiersprachen



Figure 173:

Vergleich zwischen MySQL und PostgreSQL

Beispiel zu relationalen Datenbanken

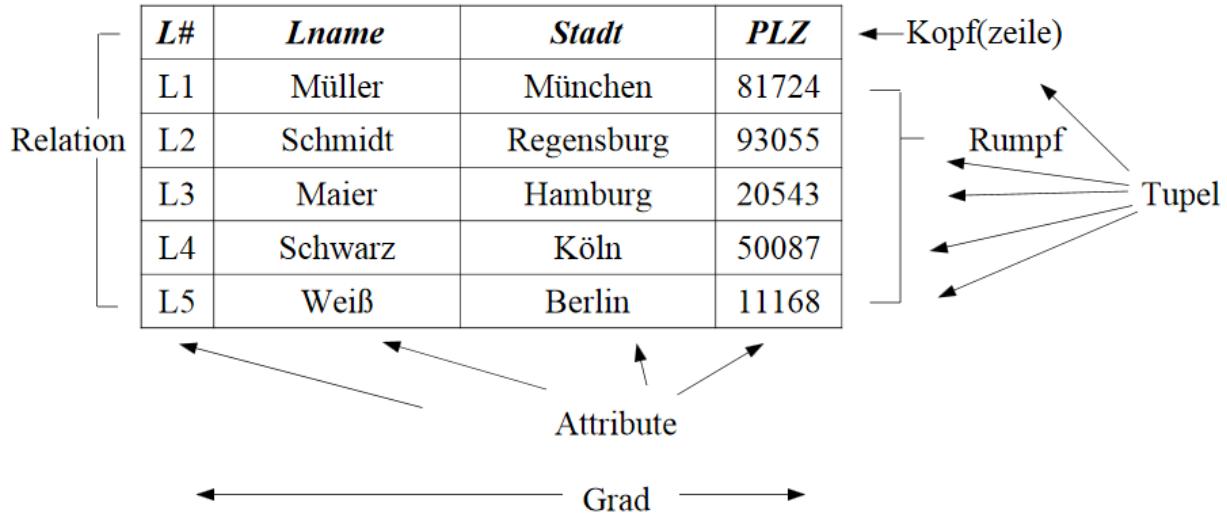


Figure 174:

Was ist der Unterschied zwischen SQL und NoSQL

- NoSQL steht für not only SQL
- Key-Value-Stores (bspw. CouchDB, MongoDB) und Speicherung unstrukturierter Daten wird durch Schema Evolution ermöglicht
- mit NoSQL lassen sich deutlich größere Datenmengen handeln

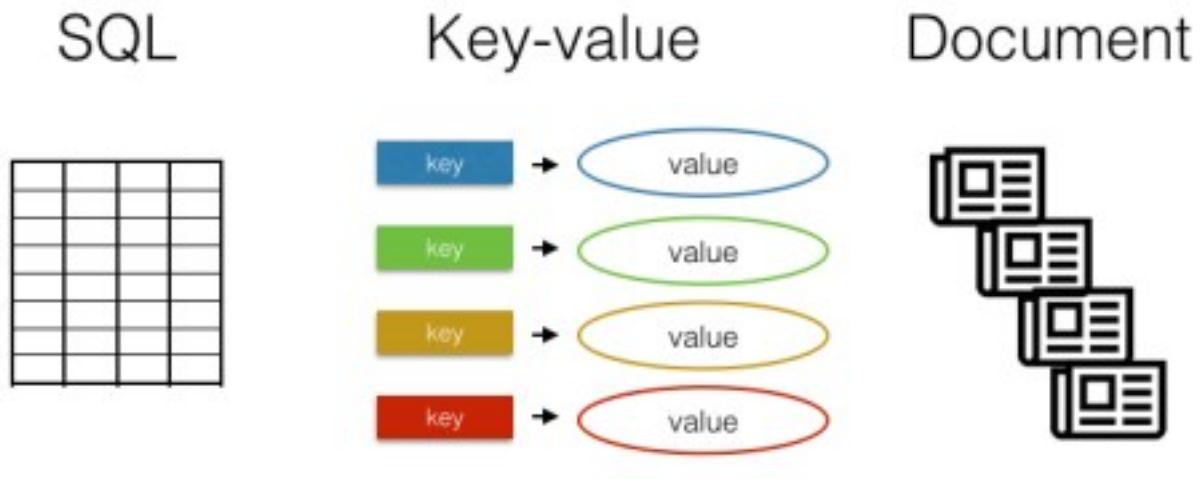


Figure 175:

- Horizontale Skalierbarkeit - wichtig bei Daten wie Video, Audio oder Bild-Dateien
- NoSQL-Bewegung ist nicht proprietär an einen Hersteller gebunden

MongoDB

- MongoDB - kann sehr schnell und einfach installiert und benutzt werden.
- Schema-freie, dokumentenorientierte NoSQL-Datenbank
- kann Sammlungen von JSON-ähnlichen Dokumenten verwalten

CouchDB

- dokumentenorientierte Datenbank
- zur Interaktion mit CouchDB kann das Paket `sofa` verwendet werden

Podcast zu CouchDB

Quick-R zur Integration von Datenbanken

SQL lernen...

- ... mit W3School
- ... mit SQLzoo
- ... SQL lernen mit tutorialspoint
- ... mit Beispielen von sql tutorial

CRE125 CouchDB

Veröffentlicht am **10. Juni 2009**

Die moderne Key/Value-Datenbank lädt Entwickler zum Entspannen ein

CouchDB ist ein vergleichsweise neues Projekt im Bereich der Datenbanken, dass durch eine Reihe frischer Ansätze mit den etablierten Konzepten relationaler Datenbanken bricht. CouchDB speichert Daten dokumentenorientiert und ohne feste Struktur und orientiert sich in seinem Kommunikationsmodell am World Wide Web und nutzt dessen Standards wie z.B. HTTP. Im Gespräch mit Tim Pritlove



Figure 176:

Access to Database Management Systems (DBMS)

ODBC Interface

The **RODBC** package provides access to databases (including Microsoft Access and Microsoft SQL Server) through an ODBC interface.

Figure 177:

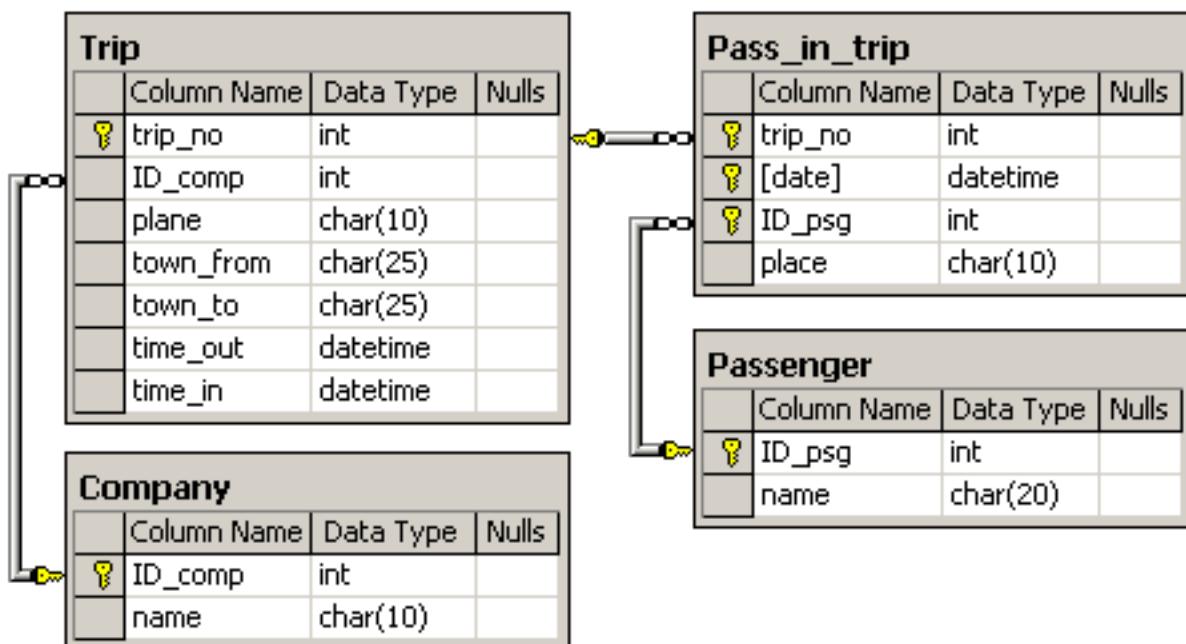


Figure 178:

essentialSQL Home About Programs

How to use SQL Union in Set Operations

In this puzzle, we're going to learn how to write a SQL UNION without using Set Operations. Set operations allow us to compare rows from two or more tables to arrive at a result. For several classes of problems, it is much easier to use a set operation, than join data. But can we [...]

[Continue reading](#)

Kris Wenzel
Puzzles
Tags ↓

Figure 179:

Weitergehendes Lernen

Weitere Ressourcen

- 10 einfache Schritte um SQL zu verstehen
- Abfrage planen
- Video um NoSQL zu verstehen

Das R-Paket dplyr

Das Paket dplyr

```
install.packages("nycflights13")

library(nycflights13)
dim(flights)

## [1] 336776      19

head(flights)

## # A tibble: 6 × 19
##   year month   day dep_time sched_dep_time dep_delay arr_time
##   <int> <int> <int>     <int>           <int>     <dbl>    <int>
## 1 2013     1     1      517            515        2     830
## 2 2013     1     1      533            529        4     850
## 3 2013     1     1      542            540        2     923
## 4 2013     1     1      544            545       -1    1004
## 5 2013     1     1      554            600       -6     812
## 6 2013     1     1      554            558       -4     740
## # ... with 12 more variables: sched_arr_time <int>, arr_delay <dbl>,
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <chr>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>,
## #   time_hour <dttm>
```

Die Reihen filtern mit filter()

```
library(dplyr)
head(filter(flights, month == 1, day == 1))

## # A tibble: 6 × 19
##   year month   day dep_time sched_dep_time dep_delay arr_time
##   <int> <int> <int>     <int>           <int>     <dbl>    <int>
## 1 2013     1     1      517            515        2     830
## 2 2013     1     1      533            529        4     850
## 3 2013     1     1      542            540        2     923
## 4 2013     1     1      544            545       -1    1004
## 5 2013     1     1      554            600       -6     812
## 6 2013     1     1      554            558       -4     740
## # ... with 12 more variables: sched_arr_time <int>, arr_delay <dbl>,
```

```
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <chr>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>,
## #   time_hour <dttm>
```

Das Paket downloader

```
install.packages("downloader")
library(downloader)
```

downloader: Download Files over HTTP and HTTPS

Provides a wrapper for the download.file function, making it possible to download files over HTTPS on Windows, Mac OS X, and other Unix-like platforms. The 'RCurl' package provides this functionality (and much more) but can be difficult to install because it must be compiled with external dependencies. This package has no external dependencies, so it is much easier to install.

| | |
|-------------|---------------------------------------|
| Version: | 0.4 |
| Imports: | utils, digest |
| Suggests: | testthat |
| Published: | 2015-07-09 |
| Author: | Winston Chang |
| Maintainer: | Winston Chang <winston at stdout.org> |

Figure 180:

Einen Beispieldatensatz herunterladen und importieren

```
url <- "https://raw.githubusercontent.com/genomicsclass/dagdata/master/inst/extdata/msleep_ggplot2.csv"
filename <- "msleep_ggplot2.csv"
```

- folgender Code sagt, dass das File nur heruntergeladen wird, wenn es noch nicht existiert

```
setwd("data/")
if (!file.exists(filename)) download(url,filename)
msleep <- read.csv("msleep_ggplot2.csv")
```

Den Datensatz anschauen

- die ersten Zeilen des Datensatzes anschauen

```
head(msleep)
```

| | | name | genus | vore | order | conservation |
|------|----------------------------|------------|-------|--------------|--------------|--------------|
| ## 1 | Cheetah | Acinonyx | carni | Carnivora | lc | |
| ## 2 | Owl monkey | Aotus | omni | Primates | <NA> | |
| ## 3 | Mountain beaver | Aplodontia | herbi | Rodentia | nt | |
| ## 4 | Greater short-tailed shrew | Blarina | omni | Soricomorpha | lc | |
| ## 5 | Cow | Bos | herbi | Artiodactyla | domesticated | |
| ## 6 | Three-toed sloth | Bradypus | herbi | Pilosa | <NA> | |

```

##   sleep_total sleep_rem sleep_cycle awake brainwt bodywt
## 1      12.1       NA        NA  11.9      NA 50.000
## 2      17.0       1.8        NA   7.0 0.01550  0.480
## 3      14.4       2.4        NA   9.6      NA 1.350
## 4      14.9       2.3 0.1333333  9.1 0.00029  0.019
## 5       4.0       0.7 0.6666667 20.0 0.42300 600.000
## 6      14.4       2.2 0.7666667  9.6      NA 3.850

```

Eine erste Auswahl treffen

- die Befehle erinnern schon an die SQL Sprache

```

sleepData <- select(msleep, name, sleep_total)
head(sleepData)

```

```

##           name sleep_total
## 1      Cheetah      12.1
## 2 Owl monkey      17.0
## 3 Mountain beaver 14.4
## 4 Greater short-tailed shrew 14.9
## 5 Cow              4.0
## 6 Three-toed sloth 14.4

```

Was bedeuten die Spalten

| column name | Description |
|--------------|---------------------------------------|
| name | common name |
| genus | taxonomic rank |
| vore | carnivore, omnivore or herbivore? |
| order | taxonomic rank |
| conservation | the conservation status of the mammal |
| sleep_total | total amount of sleep, in hours |
| sleep_rem | rem sleep, in hours |
| sleep_cycle | length of sleep cycle, in hours |
| awake | amount of time spent awake, in hours |
| brainwt | brain weight in kilograms |
| bodywt | body weight in kilograms |

Figure 181:

Die verschiedenen dplyr Befehle

Spalten auswählen

| dplyr verbs | Description |
|--------------------------|--|
| <code>select()</code> | select columns |
| <code>filter()</code> | filter rows |
| <code>arrange()</code> | re-order or arrange rows |
| <code>mutate()</code> | create new columns |
| <code>summarise()</code> | summarise values |
| <code>group_by()</code> | allows for group operations in the “split-apply-combine” concept |

Figure 182:

```
sleepData <- select(msleep, name, sleep_total)
head(sleepData)
```

```
##           name sleep_total
## 1          Cheetah     12.1
## 2      Owl monkey     17.0
## 3 Mountain beaver     14.4
## 4 Greater short-tailed shrew     14.9
## 5            Cow      4.0
## 6 Three-toed sloth     14.4
```

Das Gegenteil

- mit dem '-' Zeichen kann man sich alle Spalten bis auf die entsprechende anzeigen lassen

```
head(select(msleep, -name))
```

```
##      genus vore      order conservation sleep_total sleep_rem
## 1  Acinonyx carni Carnivora         lc      12.1       NA
## 2     Aotus omni   Primates        <NA>     17.0      1.8
## 3  Aplodontia herbi Rodentia        nt      14.4      2.4
## 4    Blarina omni Soricomorpha      lc      14.9      2.3
## 5      Bos herbi Artiodactyla domesticated      4.0      0.7
## 6   Bradypus herbi    Pilosa        <NA>     14.4      2.2
##   sleep_cycle awake brainwt bodywt
## 1        NA  11.9      NA 50.000
## 2        NA    7.0 0.01550   0.480
## 3        NA    9.6      NA  1.350
## 4  0.1333333  9.1 0.00029   0.019
## 5  0.6666667 20.0 0.42300 600.000
## 6  0.7666667    9.6      NA   3.850
```

Auswahl treffen

- alle Spalten anzeigen lassen, die mit einer Kombination von Buchstaben anfangen

```
head(select(msleep, starts_with("sl")))
```

```
##   sleep_total sleep_rem sleep_cycle
```

```

## 1      12.1      NA      NA
## 2      17.0      1.8      NA
## 3     14.4      2.4      NA
## 4     14.9      2.3  0.1333333
## 5      4.0      0.7  0.6666667
## 6     14.4      2.2  0.7666667

```

1. `ends_with()` = Select columns that end with a character string
2. `contains()` = Select columns that contain a character string
3. `matches()` = Select columns that match a regular expression
4. `one_of()` = Select columns names that are from a group of names

Figure 183:

Zeilen auswählen

```
filter(msleep, sleep_total >= 16)
```

```

##           name   genus  vore      order conservation
## 1      Owl monkey   Aotus  omni  Primates      <NA>
## 2 Long-nosed armadillo   Dasypus  carni  Cingulata      lc
## 3 North American Opossum   Didelphis  omni Didelphimorphia      lc
## 4      Big brown bat   Eptesicus insecti Chiroptera      lc
## 5 Thick-tailed opossum   Lutreolina  carni Didelphimorphia      lc
## 6      Little brown bat   Myotis insecti Chiroptera      <NA>
## 7      Giant armadillo   Priodontes insecti  Cingulata      en
## 8 Arctic ground squirrel Spermophilus  herbi  Rodentia      lc
##   sleep_total sleep_rem sleep_cycle awake brainwt bodywt
## 1      17.0      1.8      NA    7.0  0.01550  0.480
## 2      17.4      3.1  0.3833333    6.6  0.01080  3.500
## 3      18.0      4.9  0.3333333    6.0  0.00630  1.700
## 4      19.7      3.9  0.1166667    4.3  0.00030  0.023
## 5      19.4      6.6      NA    4.6      NA  0.370
## 6      19.9      2.0  0.2000000    4.1  0.00025  0.010
## 7      18.1      6.1      NA    5.9  0.08100 60.000
## 8      16.6      NA      NA    7.4  0.00570  0.920

```

Mehrere logische Abfragen um Zeilen auszuwählen

```
filter(msleep, sleep_total >= 16, bodywt >= 1)
```

```

##           name   genus  vore      order conservation
## 1 Long-nosed armadillo   Dasypus  carni  Cingulata      lc
## 2 North American Opossum   Didelphis  omni Didelphimorphia      lc
## 3      Giant armadillo   Priodontes insecti  Cingulata      en
##   sleep_total sleep_rem sleep_cycle awake brainwt bodywt
## 1      17.4      3.1  0.3833333    6.6  0.0108    3.5
## 2      18.0      4.9  0.3333333    6.0  0.0063    1.7
## 3      18.1      6.1      NA    5.9  0.0810 60.000

```

Neue Spalten erzeugen mit `mutate`

```

msleep %>%
  mutate(rem_proportion = sleep_rem / sleep_total) %>%
  head

## # A tibble: 6 x 8
##   name      genus vore    order conservation
##   <fct>     <fct> <fct>   <fct>    <fct>
## 1 Cheetah   Acinonyx carni Carnivora lc
## 2 Owl monkey Aotus    omni    Primates <NA>
## 3 Mountain beaver Aplodontia herbi Rodentia nt
## 4 Greater short-tailed shrew Blarina  omni  Soricomorpha lc
## 5 Cow        Bos     herbi Artiodactyla domesticated
## 6 Three-toed sloth Bradypus herbi Pilosa <NA>
## #> # A tibble: 6 x 8
## #>   sleep_total sleep_rem sleep_cycle awake brainwt bodywt rem_proportion
## #>   <dbl>       <dbl>      <dbl>    <dbl>    <dbl>    <dbl>          <dbl>
## #> 1 12.1        NA         NA      11.9     NA    50.000          NA
## #> 2 17.0        1.8        NA      7.0     0.01550    0.480  0.1058824
## #> 3 14.4        2.4        NA      9.6     NA    1.350  0.1666667
## #> 4 14.9        2.3      0.1333333  9.1     0.00029    0.019  0.1543624
## #> 5  4.0        0.7      0.6666667 20.0     0.42300  600.000  0.1750000
## #> 6 14.4        2.2      0.7666667  9.6     NA    3.850  0.1527778

```

Die Anweisung group_by

```

msleep %>%
  group_by(order) %>%
  summarise(avg_sleep = mean(sleep_total),
            min_sleep = min(sleep_total),
            max_sleep = max(sleep_total),
            total = n())
## # A tibble: 19 × 5
##   order avg_sleep min_sleep max_sleep total
##   <fctr>     <dbl>      <dbl>      <dbl> <int>
## 1 Afrosoricida 15.600000    15.6       15.6    1
## 2 Artiodactyla  4.516667    1.9        9.1     6
## 3 Carnivora    10.116667   3.5       15.8    12
## 4 Cetacea       4.500000    2.7        5.6     3
## 5 Chiroptera   19.800000   19.7       19.9    2
## 6 Cingulata    17.750000   17.4       18.1    2
## 7 Didelphimorphia 18.700000  18.0       19.4    2
## 8 Diprotodontia 12.400000   11.1       13.7    2
## 9 Erinaceomorpha 10.200000   10.1       10.3    2
## 10 Hyracoidea   5.666667    5.3        6.3     3
## 11 Lagomorpha   8.400000    8.4        8.4     1
## 12 Monotremata  8.600000    8.6        8.6     1
## 13 Perissodactyla 3.466667  2.9        4.4     3
## 14 Pilosa       14.400000   14.4       14.4    1
## 15 Primates     10.500000   8.0        17.0    12
## 16 Proboscidea  3.600000    3.3        3.9     2
## 17 Rodentia     12.468182   7.0        16.6    22
## 18 Scandentia   8.900000    8.9        8.9     1
## 19 Soricomorpha 11.100000   8.4        14.9    5

```

Databases

2016-06-23

As well as working with local in-memory data like data frames and data tables, dplyr also works with remote on-disk data stored in databases. Generally, if your data fits in memory there is no advantage to putting it in a database: it will only be slower and more hassle. The reason you'd want to use dplyr with a database is because either your data is already in a database (and you don't want to work with static csv files that someone else has dumped out for you), or you have so much data that it does not fit in memory and you have to use a database. Currently dplyr supports the three most popular open source databases (sqlite, mysql and postgresql), and google's bigquery.

Figure 184:

Am Besten funktioniert mit SQLite

- alles was man braucht wird quasi schon mit R mitgeliefert
- der Befehl `src_sqlite` kann genutzt werden um sich mit einer Datenbank zu verbinden
- bei Verwendung von `create=T` wird eine neue Datenbank erzeugt
- bei `create=F` muss man den Pfad zur Datenbank angeben

```
library(dplyr)
setwd("data")
my_db <- src_sqlite("my_db.sqlite3", create = T)
```

Erste Datenbank mit Beispieldatensatz befüllen

```
library(nycflights13)
flights_sqlite <- copy_to(my_db, flights, temporary = FALSE, indexes = list(
  c("year", "month", "day"), "carrier", "tailnum"))
```

Den Datensatz wieder heraus bekommen

- mit dem Befehl `tbl` kann man sich mit Tabellen innerhalb einer Datenbank verbinden

```
flights_sqlite <- tbl(nycflights13_sqlite(), "flights")
```

- das gleiche Ergebnis:

```
tbl(my_db, sql("SELECT * FROM flights"))
```

Eine weitere Abfrage

```
tbl(my_db, sql("SELECT * FROM flights WHERE month = 1 AND dep_time = 517"))
```

Integration von PostgreSQL mit dem Paket

RPostgreSQL

Die Nutzung von RPostgreSQL



Figure 185:

PostgreSQL installieren

- Installation Windows
- Installation Linux

pgAdmin

pgAdmin ist eine [Open-Source-Software](#) zur Entwicklung und Administration von PostgreSQL-Datenbanken und davon abgeleiteten Datenbanken wie EnterpriseDB Postgres Plus Advanced Server oder Greenplum. Eine [graphische Benutzeroberfläche](#) erleichtert die Administration von Datenbanken. Der Editor für SQL-Abfragen enthält ein graphisches EXPLAIN, mit dessen Hilfe sich performantere Abfragen erstellen lassen. Durch eine native Anbindung an PostgreSQL ermöglicht das [GUI](#) den Zugriff auf die gesamte PostgreSQL-Funktionalität.

Figure 186:

PG admin installieren

- PGadmin
- Tutorial zur Nutzung von PGadmin



Figure 187:

Neue Datenbank anlegen



Figure 188:

Eine neue Datenbank

- Unter Linux kann man auch in der Kommandozeile einen neuen Nutzer anlegen:

```
sudo -u postgres createuser Japhilko
```

- und auch eine neue Datenbank anlegen:

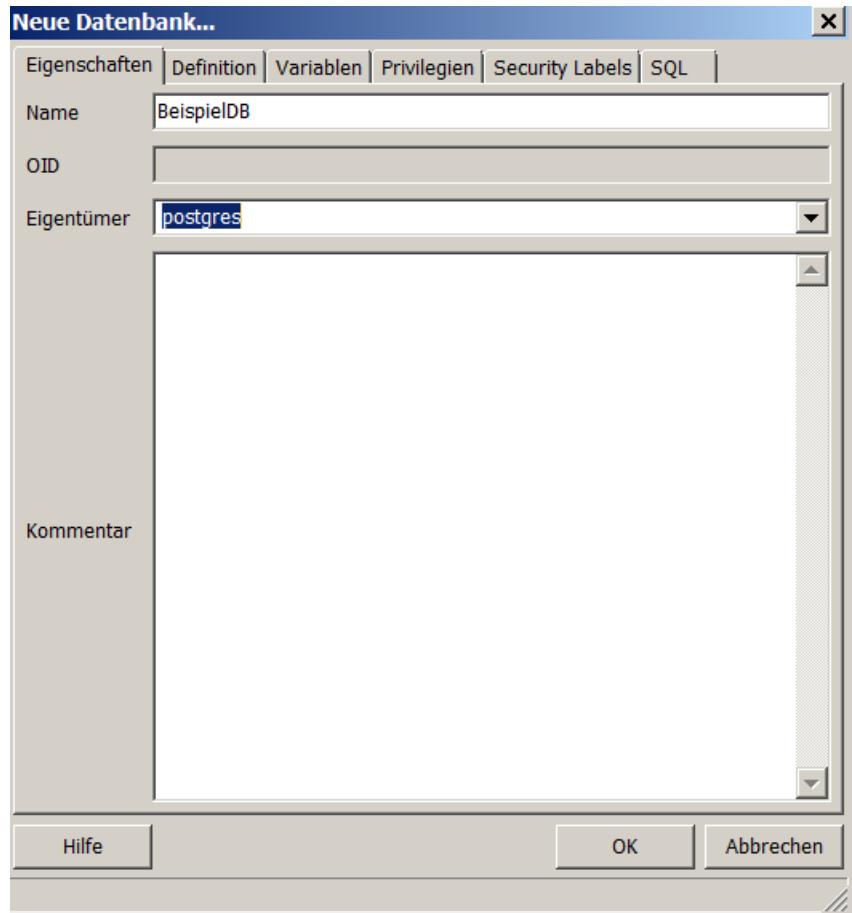


Figure 189:

```
sudo -u postgres createdb -E UTF8 -O Japhilko offlgeoc
```

Wie bekomme ich Daten in die Datenbank

```
install.packages("RPostgreSQL")  
  
library("RPostgreSQL")  
  
citation("RPostgreSQL")  
  
##  
## To cite package 'RPostgreSQL' in publications use:  
##  
##   Joe Conway, Dirk Eddelbuettel, Tomoaki Nishiyama, Sameer Kumar  
##   Prayaga and Neil Tiffin (2016). RPostgreSQL: R interface to the  
##   PostgreSQL database system. R package version 0.4-1.  
##   https://CRAN.R-project.org/package=RPostgreSQL  
##  
## A BibTeX entry for LaTeX users is  
##  
##   @Manual{,  
##     title = {RPostgreSQL: R interface to the PostgreSQL database system},  
##     author = {Joe Conway and Dirk Eddelbuettel and Tomoaki Nishiyama and Sameer Kumar Prayaga and Ne  
##     year = {2016},  
##     note = {R package version 0.4-1},  
##     url = {https://CRAN.R-project.org/package=RPostgreSQL},  
##   }  
##  
## ATTENTION: This citation information has been auto-generated from  
## the package DESCRIPTION file and may need manual editing, see  
## 'help("citation")'.
```

Datenbank mit R verbinden

```
pw <- {"1234"}  
drv <- dbDriver("PostgreSQL")  
con <- dbConnect(drv, dbname = "BeispielDB",  
                  host = "localhost", port = 5432,  
                  user = "postgres", password = pw)  
rm(pw) # removes the password  
  
dbExistsTable(con, "BeispielDB")
```

Daten an Datenbank schicken

```
data(mtcars)  
df <- data.frame(carname = rownames(mtcars),  
                  mtcars,  
                  row.names = NULL)  
df$carname <- as.character(df$carname)
```

```
rm(mtcars)

dbWriteTable(con, "cartable",
            value = df, append = TRUE, row.names = FALSE)
```

- eine Abfrage machen:

```
df_postgres <- dbGetQuery(con, "SELECT * from cartable")
```

- die beiden Tabellen müssten gleich sein

```
identical(df, df_postgres)
```

Anwendung - Geodaten in die Datenbank migrieren

- Zunächst muss für die Datenbank die postgis Erweiterung installiert werden:

```
CREATE EXTENSION postgis;
```

- Der Anfang mit PostGIS
- PostGIS und R

Programm zum Import der OpenStreetMap Daten in PostgreSQL- osm2pgsql

- Ausschnitte der OpenStreetMap Daten können bei der Geofabrik heruntergeladen werden
- Nutzung von osm2pgsql
- läuft unter Linux deutlich besser
- so könnte bspw. ein Import in PostgreSQL aussehen:

```
osm2pgsql -c -d osmBerlin --slim -C -k berlin-latest.osm.pbf
```

```
osm2pgsql -s -U postgres -d offlgeoc /home/kolb/Forschung/osmData/data/saarland-latest.osm.pbf
```

```
osm2pgsql -s -U postgres -d offlgeocRLP -o gazetteer /home/kolb/Forschung/osmData/data/rheinland-pfalz-latest.osm.pbf
```

Mögliche Abfragen

So bekommt man alle administrativen Grenzen:

```
SELECT name FROM planet_osm_polygon WHERE boundary='administrative'
```

- mehr als eine Spalte auswählen

```
df_postgres <- dbGetQuery(con, "SELECT name, admin_level FROM planet_osm_polygon WHERE boundary='administrative'")
```

Eine Abfrage zu administrativen Grenzen (ein spezielles Level)

```
df_adm8 <- dbGetQuery(con, "SELECT name, admin_level FROM planet_osm_polygon WHERE boundary='administrative' AND admin_level=8")
```

Mögliche Abfragen

```
df_hnr <- dbGetQuery(con, "SELECT * FROM planet_osm_line, planet_osm_point  
WHERE planet_osm_line.name='Nordring' AND planet_osm_line.highway IN ('motorway','trunk','primary')  
AND planet_osm_point.name='Ludwigshafen' AND planet_osm_point.place IN ('city', 'town')  
ORDER BY ST_Distance(planet_osm_line.way, planet_osm_point.way)")  
  
df_hnr <- dbGetQuery(con, "SELECT * FROM planet_osm_line, planet_osm_point  
WHERE planet_osm_line.name='Nordring' AND planet_osm_point.name='Ludwigshafen'  
ORDER BY ST_Distance(planet_osm_line.way, planet_osm_point.way)")  
head(df_hnr)  
  
df_ <- dbGetQuery(con, "SELECT * FROM planet_osm_line, planet_osm_point  
WHERE planet_osm_line.name='Nordring' AND planet_osm_point.name='Ludwigshafen'  
ORDER BY ST_Distance(planet_osm_line.way, planet_osm_point.way)")  
head(df_)  
  
colnames(df_)  
  
table(df_$name)
```

Adresse in einem Ort

```
df_sipp <- dbGetQuery(con, "SELECT * FROM planet_osm_line, planet_osm_point  
WHERE planet_osm_line.name='Rechweg' AND planet_osm_point.name='Sippersfeld'  
ORDER BY ST_Distance(planet_osm_line.way, planet_osm_point.way)")  
head(df_sipp)
```

PostgreSQL and Leaflet

```
install.packages("plot3D")  
  
library(plot3D)  
library(RPostgreSQL)
```

Links

- osm2pgsql
- Andrew Whitby - Roll-your-own geocoding with OpenStreetMap Nominatim on Amazon EC2
- OpenStreetMap Nominatim Server for Geocoding
- Getting Started With PostGIS
- PostGIS geocode
- Nominatim installation
- Wie bekommt man OSM Daten
- PostgreSQL

Nutzung von weiteren Datenbanken (MongoDB, MySQL)

MySQL Eine erste Verbindung einrichten

- Was ist MySQL
- MySQL installieren

2.3 Installing MySQL on Microsoft Windows

- 2.3.1 MySQL Installation Layout on Microsoft Windows
- 2.3.2 Choosing An Installation Package
- 2.3.3 MySQL Installer for Windows
- 2.3.4 MySQL Notifier
- 2.3.5 Installing MySQL on Microsoft Windows Using a noinstall Zip Archive
- 2.3.6 Troubleshooting a Microsoft Windows MySQL Server Installation
- 2.3.7 Windows Postinstallation Procedures
- 2.3.8 Upgrading MySQL on Windows

Figure 190:

Der Start mit MySQL

- – PC Magazin - MySQL Datenbankserver

RMySQL

- Das Paket RMySQL auf dem CRAN Server

```
install.packages("RMySQL")  
  
library(RMySQL)  
mydb = dbConnect(MySQL(), user='user', password='password', dbname='database_name', host='host')
```

- Das Paket RMySQL auf Github
- Auf MySQL mit R zugreifen

Weitere Möglichkeit die Schnittstelle zwischen R und MySQL Datenbank nutzen

```
library(DBI)  
con <- dbConnect(RMySQL::MySQL(), group = "my-db")
```

- Youtube Video zu PHP & MySQL - Einführung und Installation

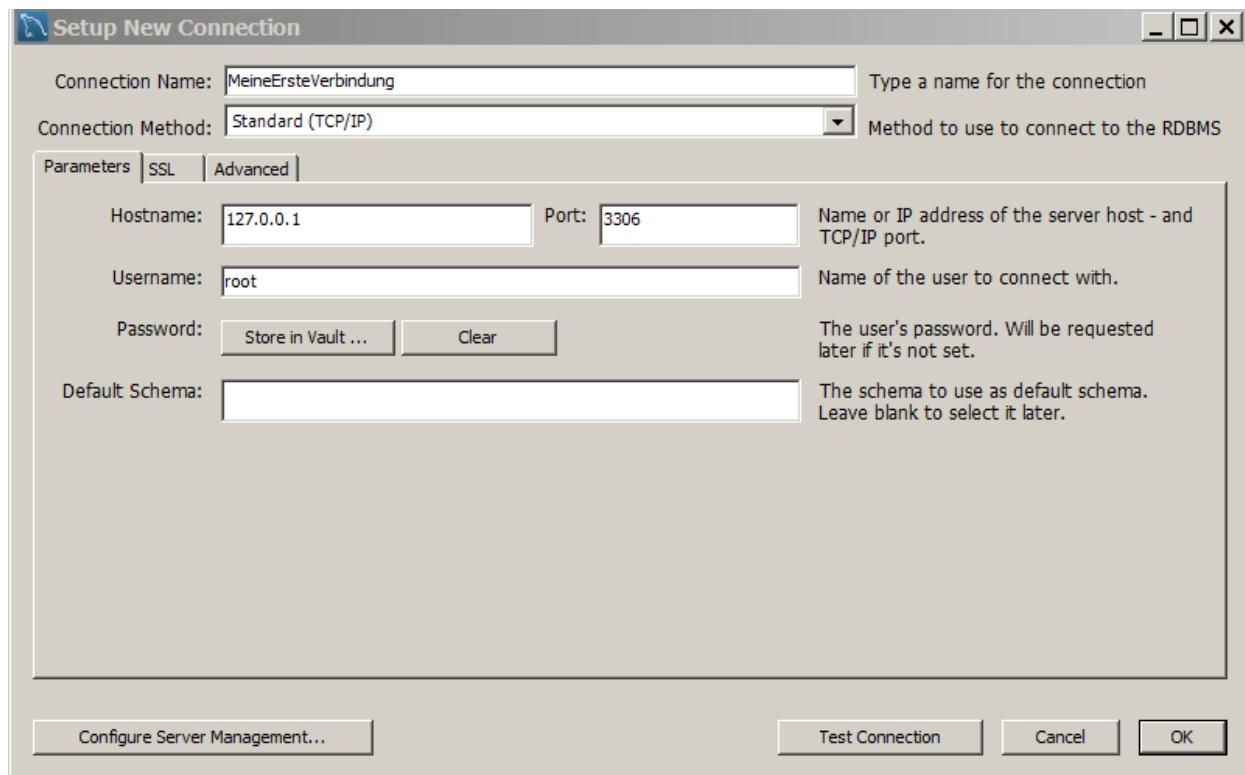


Figure 191:

Mongodb

- Installation von MongoDB unter Windows

Hello World

- Getting started with MongoDB in R

```
install.packages("mongolite")
library(mongolite)
m <- mongo(collection = "diamonds")
```

- R und Mongodb
- Cheatsheet zur Nutzung von Rmongodb
- Rpubs - mongoDB

CouchDB

- Installation von CouchDB auf Windows
- zur Interaktion mit CouchDB kann das Paket sofa verwendet werden

```
install.packages("jsonlite")
devtools::install_github("ropensci/sofa")
```

- CouchDB und OpenStreetMap

```
library("sofa")
```

Links

- Eine Liste aller NoSQL Datenbanken
- Azure in R nutzen
- Datenbanken in R
- Grundlagen Datenbanken - `dplyr` und `DBI`
- Frühe Entwicklung zu Integration von Datenbanken in R
- Full SSL Support 2nd link