

# Karten erstellen mit R

Jan-Philipp Kolb

11 Januar 2019

# Kleine Vorstellungsrunde

- Wie beurteilt Ihr Eure Fähigkeiten mit R?
- Habt Ihr Erfahrungen mit anderen Programmiersprachen / Statistiksoftware? Wenn ja welche?
- Was sind Eure Erwartungen für diesen Kurs?

# Disclaimer/ Informationen vorab

Normalerweise gibt es große Unterschiede bei Vorkenntnissen und Fähigkeiten - bitte gebt Bescheid, wenn es zu schnell oder zu langsam geht oder etwas unklar geblieben ist.

- Wenn es Fragen gibt - immer fragen
- In diesem Kurs gibt es viele **Übungen**, denn das Programmieren / die Nutzung von R lernt man am Ende nur allein.
- Ich habe viele **Beispiele** - probiert sie aus
- R macht mehr Spaß zusammen - arbeitet zusammen!

# Disclaimer

- Zum Import, zur Verarbeitung und Visualisierung gibt es bereits sehr viele Pakete.
- Das Gebiet entwickelt sich sehr schnell.
- Es ist nicht möglich alles davon in diesem Kurs vorzustellen.
- Ich möchte anhand einiger interessanter Beispiele einen Einblick darin geben, was alles möglich ist.

# Warum R?

# Gründe R zu nutzen...

- ... R ist eine **quelloffene Sprache**
- ... hervorragende **Grafiken, Grafiken, Grafiken**
- ... R kann in Kombination mit anderen Programmen verwendet werden - z.B. zur **Verknüpfung von Daten**
- ... R kann zur **Automatisierung** verwendet werden
- ... Breite und aktive Community - **Man kann die Intelligenz anderer Leute nutzen ;)**

# R kann in Kombination mit anderen Programmen genutzt werden...

Use R!

Richard M. Heiberger  
Erich Neuwirth

## R Through Excel



SASmixed



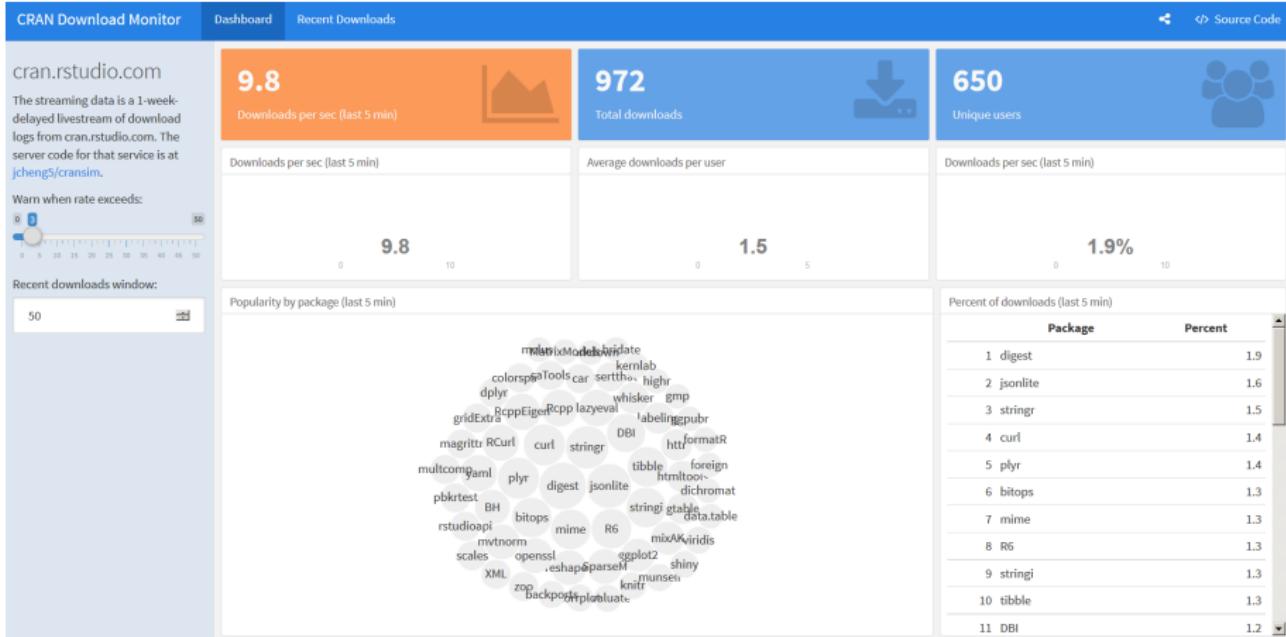
rPython R package

Statistics and Computing

Robert A. Muenchen · Joseph M. Hilbe

## R for Stata Users

# Die Beliebtheit von R-Paketen



# Download R:

<http://www.r-project.org/>



[CRAN](#)  
[Mirrors](#)  
[What's new?](#)  
[Task Views](#)  
[Search](#)

[About R](#)  
[R Homepage](#)  
[The R Journal](#)

[Software](#)  
[R Sources](#)  
[R Binaries](#)  
[Packages](#)  
[Other](#)

## The Comprehensive R Archive Network

### Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

### Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (Friday 2017-04-21, You Stupid Darkness)  
[R-3.4.0.tar.gz](#), read [what's new](#) in the latest version.

# Open Source Programm R

## Das ist das Basis-R:

The screenshot shows the RGui (32-bit) interface with two windows open. The top window is the R Editor, titled 'RGui (32-bit)', displaying the file path 'C:\Users\kolbjp\Documents\GitHub\IntroR\2017\rcode\HalloWelt.R - R Editor' and the code '# Hallo Welt' followed by 'a <- c(1,2,3)'. The bottom window is the R Console, titled 'R Console', displaying the R startup message in German and the command history: '> # Hallo Welt', '>', '> a <- c(1,2,3)', and '>'.

```
R Gui (32-bit)
Datei Bearbeiten Pakete Windows Hilfe
C:\Users\kolbjp\Documents\GitHub\IntroR\2017\rcode\HalloWelt.R - R Editor
# Hallo Welt
a <- c(1,2,3)

R Console
Tippen Sie 'demo()' für einige Demos, 'help()' für on-line Hilfe, oder
'help.start()' für eine HTML Browserschnittstelle zur Hilfe.
Tippen Sie 'q()', um R zu verlassen.

[Vorher gesicherter Workspace wiederhergestellt]

> # Hallo Welt
>
> a <- c(1,2,3)
> |
```

# Graphical user interface

Viele Leute nutzen ein **Graphical User Interface** (GUI) oder ein **Integrated Development Interface** (IDE).

Aus den folgenden Gründen:

- Syntax-Hervorhebung
- Auto-Vervollständigung
- Bessere Übersicht über Graphiken, Pakete, Dateien, ...

# RStudio

The screenshot shows the RStudio interface with the following components:

- Code Editor:** Displays R code for generating a presentation. The code includes sections for "Getting started", "How to get help", "Data import", "GESIS panel data", "Basic data analysis", "Survey package", "Graphics", "Linear regression", "Logistic regression", "Error messages", and "Hierarchical/Multilevel models". It also includes code for reading an Excel file, scheduling tasks, and using knitr to generate a PDF.
- File Browser:** Shows the project structure under "IntroDataAnalysis" with files like "A1\_General.Rmd", "A1\_General.html", and "A1\_General.pdf".
- Terminal:** Shows the command used to run pandoc to convert the R Markdown file into a PDF, including options for Beamer, theme, color, font, and structure.

```

42 c("A1 Getting started","A2 How to get help", "A3 Data import","A4 The GESIS panel data
43   data","A5 data export")
44
45 c("B1 Basic data analysis","B2 The use of the survey package","B3 Graphics","B4 linear
46   regression","B5 logistic regression")
47 c("C1 Understanding error messages","C2 Hierarchical/Multilevel models")
48
49
50 ```{r,echo=F}
51 library(knitr)
52 sched <- xlsx::read.xlsx2("../orga/schedule.xlsx")
53 dates <- sched[1:7,c(1,4,5)]
54 dates <- dates[which(dates$Part=="Break"),]
55
56 kable(dates, row.names = F)
57 ...
58
59
60 ## Overview - advantages of R
61
62
63 # Create a ...
64
65
66 ordinary text without R code
67
68 |.....| 96%
69 label: unnamed-chunk-13 (with options)
70 List of 1
71 $ eval: symbol F
72
73 |.....| 100%
74 ordinary text without R code
75
76
77 "D:/Programme/RStudio/bin/pandoc/pandoc" -o A2_How2gethelp.pdf --from markdown+autolink_bare_uris+asciidoc+identifiers+tex_math_single_backslash+implicit_figures
78 --output A2_How2gethelp.pdf --variable theme=CambridgeUS --variable colortheme=beaver --variable fonttheme=structurebold --highlight-style tango --pdf-engine pdflatex --self-contained
79 output file: A2_How2gethelp.knit.md
80
81
82 Output created: A2_How2gethelp.pdf

```

# Übung - Vorbereitung

- Schaue, ob R auf dem Computer installiert ist
- Wenn nicht, lade **R** herunter und installiere es.
- Prüfe ob Rstudio installiert ist.
- Wenn nicht - **installiere** Rstudio.
- Starte RStudio. Gehe in die Konsole (meistens Fenster unten links) und tippe
- Wenn noch kein Skript geöffnet im oberen linken Teil von Rstudio geöffnet ist, gehe zum Menü und öffne ein neues Skript. Checks das Datum mit `date()` und die R version mit `sessionInfo()`.

# Erste Schritte mit R

# R ist eine objektorientierte Sprache.

## Vektoren und Zuweisungen

- <- ist der Zuweisungsoperator

```
b <- c(1,2) # create an object with the numbers 1 and 2
```

- Auf dieses Objekt kann eine Funktion angewendet werden:

```
mean(b) # computes the mean
```

```
## [1] 1.5
```

Mit diesen Funktionen können wir etwas über die Eigenschaften des Objekts erfahren:

```
length(b) # b has the length 2
```

```
## [1] 2
```

## Objektstruktur

```
str(b) # b is a numeric vector
```

```
## num [1:2] 1 2
```

# Übung - Zuweisungen und Funktionen

Erstellen Sie einen Vektor  $b$  mit den Zahlen von 1 bis 5 und berechnen Sie....

- ① den Mittelwert
- ② die Varianz
- ③ die Standardabweichung
- ④ die Quadratwurzel aus dem Mittelwert

# Hilfe bekommen

# Wie bekomme ich Hilfe?

- Um Hilfe im Allgemeinen zu bekommen:

```
help.start()
```

- Online-Dokumentation für die meisten Funktionen:

```
help(name)
```

- Benutze ?, um Hilfe zu bekommen

```
?mean
```

- example(lm) liefert ein Beispiel für die lineare Regression

```
example(lm)
```

# Vignetten

- Eine Vignette ist ein Papier, das die wichtigsten Funktionen eines Pakets darstellt.
- Sie enthalten viele reproduzierbare Beispiele.
- Vignetten sind ein neues Werkzeug, deshalb hat nicht jedes Paket eine Vignette.

```
browseVignettes()
```

- Um eine Vignette zu bekommen:

```
vignette("osmdata")
```

# Ein Beispiel für eine Vignette - Das Paket osmdata

| <https://cran.r-project.org/web/packages/osmdata/vignettes/osmdata.html>

## 1. Introduction

`osmdata` is an R package for downloading and using data from OpenStreetMap ([OSM](#)). OSM is a global open access mapping project, which is free and open under the [ODbL licence](#) [@OpenStreetMap]. This has many benefits, ensuring transparent data provenance and ownership, enabling real-time evolution of the database and, by allowing anyone to contribute, encouraging democratic decision making and citizen science [@johnson\_models\_2017]. See the [OSM wiki](#) to find out how to contribute to the world's open geographical data commons.

Unlike the [openStreetMap](#) package, which facilitates the download of raster tiles, `osmdata` provides access to the vector data underlying OSM.

`osmdata` can be installed from CRAN with

```
install.packages("osmdata")
```

and then loaded in the usual way:

```
library(osmdata)
```

```
## Data (c) OpenStreetMap contributors, ODbL 1.0. http://www.openstreetmap.org/copyright
```

The development version of `osmdata` can be installed with the `devtools` package using the following command:

```
devtools::install_github('osmdata/osmdata')
```

# Demos

- für manche Pakete gibt es Demos:

```
demo() # zeigt alle verfügbaren Demos
demo(package = "httr") # Zeigt alle Demos in einem Paket

# Ein spezifisches Demo laufen lassen:
demo("oauth1-twitter", package = "httr")
```

- Wenn ein Demo gestartet wird, ist der zugehörige Code in der Konsole sichtbar

```
demo(nlm)
```

```
> demo(nlm)
```

```
demo(nlm)
----- ~~~~
```

# Die Funktion apropos

- durchsucht alles über den angegebenen String:

```
apropos("lm")
```

```
## [1] ".colMeans"           ".lm.fit"          "colMeans"
## [4] "confint.lm"           "contr.helmert"    "dummy.coef.lm"
## [7] "getAllMethods"         "glm"              "glm.control"
## [10] "glm.fit"               "KalmanForecast"   "KalmanLike"
## [13] "KalmanRun"             "KalmanSmooth"     "kappa.lm"
## [16] "lm"                    "lm.fit"            "lm.influence"
## [19] "lm.wfit"               "model.matrix.lm"  "nlm"
## [22] "nlminb"                "predict.glm"      "predict.lm"
## [25] "residuals.glm"         "residuals.lm"     "summary.glm"
## [28] "summary.lm"
```

- Funktion kann auch mit **regulären Ausdrücken** verwendet werden...

```
?"regular expression"
```

# Suchmaschine für die R-Seite

RSiteSearch("glm")

## R Site Search

Query:   [\[How to search\]](#)

Display:  Description:  Sort:

### Target:

- Functions
- Task views

For problems WITH THIS PAGE (not with R) contact [baron@upenn.edu](mailto:baron@upenn.edu).

## Results:

### References:

- **views:** [ glm: 11 ]
- **vignettes:** [ (can't open the index) ]
- **functions:** [ glm: 4391 ]

Total 4402 documents matching your query.

# Nutzung von Suchmaschinen

- Ich nutze **duckduckgo.de**:

R-project + "was ich schon immer wissen wollte"

- das funktioniert natürlich für alle Suchmaschinen!



DuckDuckGo

R-project + "what I want to know" |



# Stackoverflow

- Für alle Fragen zum programmieren
- Ist nicht auf R fokussiert - aber es gibt **viele Diskussionen zu R-Fragen**
- Sehr detaillierte Diskussionen

The screenshot shows the StackOverflow homepage. At the top, there's a navigation bar with links for Questions, Jobs, Documentation (beta), Tags, and Users, along with a search bar containing the placeholder "[r]". Below the navigation bar, there's a section for "Tagged Questions" with a "frequent" tab selected. A featured question is displayed with the title "How to make a great R reproducible example?". The question has 1776 votes, 22 answers, and 147k views. It includes tags like r and r-faq. To the right, there's a sidebar for the R Language documentation with 22,187 frequent questions tagged. Below that, there's a "Related Tags" section listing ggplot2, dataframe, and plot.

**Tagged Questions**

Questions Jobs Documentation BETA Tags Users

[r]

Tagged Questions

info newest **frequent** votes active unanswered

R is a free, open-source programming language and software environment for statistical computing, bioinformatics, and graphics. Please supplement your question with a minimal reproducible example. Use dput() for data and specify all non-base packages with library calls. For statistical questions ...

learn more... top users synonyms (2) r jobs

**22,187**  
frequent questions tagged

r about »

**R Language**  
DOCUMENTATION

Find a request to handle or browse 121 topics.

**Related Tags**

ggplot2 × 2875  
dataframe × 1351  
plot × 1105

# Ein Schummelzettel für Basis R

<https://www.rstudio.com/resources/cheatsheets/>

## Base R Cheat Sheet

### Getting Help

#### Accessing the help files

?mean

Get help of a particular function.

help.search('weighted mean')

Search the help files for a word or phrase.

help(package = 'dplyr')

Find help for a package.

#### More about an object

str(iris)

Get a summary of an object's structure.

class(iris)

Find the class an object belongs to.

### Using Packages

install.packages('dplyr')

Download and install a package from CRAN.

library(dplyr)

Load the package into the session, making all its functions available to use.

dplyr::select

Use a particular function from a package.

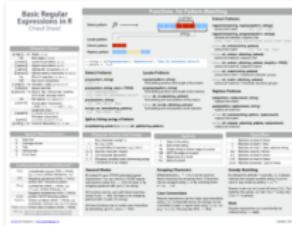
data(iris)

Load a built-in dataset into the environment.

Vectors	Programming
<b>Creating Vectors</b>	<b>For Loop</b>
<pre>c(2, 4, 6)      2 4 6      Join elements into a vector</pre>	<pre>for (variable in sequence){</pre>
<pre>2:6            2 3 4 5 6    An integer sequence</pre>	<pre>  Do something</pre>
<pre>seq(2, 3, by=0.5) 2.0 2.5 3.0  A complex sequence</pre>	<pre>}</pre>
<pre>rep(1:2, times=3) 1 2 1 2 1 2  Repeat a vector</pre>	<b>Example</b>
<pre>rep(1:2, each=3) 1 1 1 2 2 2  Repeat elements of a vector</pre>	<pre>for (i in 1:4){</pre>
	<pre>  j &lt;- i + 10</pre>
	<pre>  print(j)</pre>
	<pre>}</pre>
	<b>While Loop</b>
	<pre>while (condition){</pre>
	<pre>  Do something</pre>
	<pre>}</pre>
	<b>Example</b>
	<pre>while (i &lt; 5){</pre>
	<pre>  print(i)</pre>
	<pre>  i &lt;- i + 1</pre>
	<pre>}</pre>
<b>Vector Functions</b>	<b>If Statements</b>
<pre>sort(x)</pre>	<pre>if (condition){</pre>
<small>Return x sorted.</small>	<small>Do something</small>
<pre>rev(x)</pre>	<pre>} else {</pre>
<small>Return x reversed.</small>	<small>Do something different</small>
<pre>table(x)</pre>	
<small>See counts of values.</small>	
<pre>unique(x)</pre>	
<small>See unique values.</small>	
<b>Selecting Vector Elements</b>	<b>Functions</b>
<b>By Position</b>	
<pre>x[4]</pre>	<pre>function_name &lt;- function(var){</pre>
<small>The fourth element.</small>	<small>Do something</small>
<pre>x[-4]</pre>	<small>return(new_variable)</small>
<small>All but the fourth.</small>	
<pre>x[2:4]</pre>	
<small>Elements two to four.</small>	
<pre>x[-(2:4)]</pre>	
<small>All elements except two to four.</small>	
<pre>x[c(1, 5)]</pre>	
<small>Elements one and five.</small>	
	<b>Example</b>
	<pre>if (i &gt; 3){</pre>
	<pre>  print('Yes')</pre>
	<pre>} else {</pre>
	<pre>  print('No')</pre>
	<b>Example</b>
	<pre>square &lt;- function(x){</pre>
	<pre>  squared &lt;- x*x</pre>
	<pre>  return(squared)</pre>
	<b>Reading and Writing Data</b>
	<small>Also see the <a href="#">readr</a> package.</small>
<b>Input</b>	<b>Output</b>
<pre>df &lt;- read.table('file.txt')</pre>	<pre>write.table(df, 'file.txt')</pre>
	<small>Description</small>
	<small>Read and write a delimited text file.</small>

# Mehr Schummelzettel

## Regular Expressions



Basics of regular expressions and pattern matching in R by Ian Kopacka.  
Updated 09/16.

[DOWNLOAD](#)

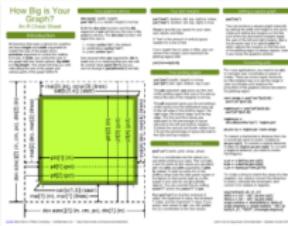
## The leaflet package



Interactive maps in R with leaflet, by Kejia Shi. Updated 05/17.

[DOWNLOAD](#)

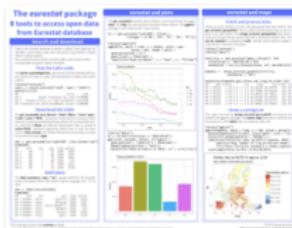
## How big is your graph?



Graph sizing with base R by by Stephen Simon. Updated 10/16.

[DOWNLOAD](#)

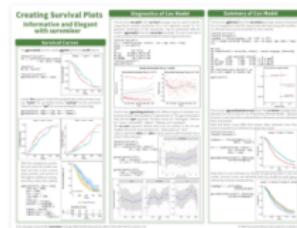
## The eurostat package



R tools to access the eurostat database, by rOpenGov. Updated 03/17.

Jan-Philipp Kolb

## The survminer package



Elegant survival plots, by Przemyslaw Biecek. Updated 03/17.

## The sjmisc package



dplyr friendly Data and Variable Transformation. by Daniel Lüdecke.

Karten erstellen mit R

# Quick R

- Immer mit vielen Beispielen und Hilfen bezüglich eines Themas
- Beispiel: **Quick R - Getting Help**



R Tutorial | R Interface | Data Input | Data Management | Statistics | Advanced Statistics | Graphs | Advanced Graphs

< R Interface

## Getting Help

Once R is installed, there is a comprehensive built-in help system. At the program's command prompt you can use any of the following:

Getting Help  
The Workspace  
Input/Output  
Packages  
Graphic User Interfaces  
Customizing Startup  
Publication Quality Output  
Batch Processing  
Reusing Results

```
help.start()    # general help
help(foo)       # help about function foo
?foo           # same thing
apropos("foo") # list all functions containing string foo
example(foo)   # show an example of function foo
```

# Weitere Links

- Überblick - wie bekommt man Hilfe in R



[Home]

Download

CRAN

## Getting Help with R

### Helping Yourself

Before asking others for help, it's generally a good idea for you to try to help yourself. R includes extensive facilities for accessing documentation and searching for help. There are also specialized search engines for accessing information about R on the internet, and general internet search engines can also prove useful ([see below](#)).

- Eine Liste mit HowTo's
- Eine Liste mit den wichtigsten R-Befehlen

**R ist modular**

# Wo man Routinen findet

- Viele Funktionen sind in Basis-R enthalten.
- Viele spezifische Funktionen sind in zusätzliche Bibliotheken integriert.
- R kann modular durch sogenannte Pakete oder Bibliotheken erweitert werden.
- Die wichtigsten Pakete, die auf CRAN gehostet werden (13640 at Do 10 Jan 2019)
- Weitere Pakete findet man z.B. unter **bioconductor**

# Übersicht R-Pakete



# Installation von Paketen

- Die Anführungszeichen um den Paketnamen herum sind für den Befehl `install.packages` notwendig.
- Sie sind optional für den Befehl `library`.
- Man kann auch `require` anstelle von `library` verwenden.

```
install.packages("raster")
```

```
library(raster)
```

# Installation von Paketen mit RStudio

The screenshot shows the RStudio interface. The top navigation bar includes tabs for Environment, History, and Presentation. The Environment pane displays the message "Environment is empty". The Packages pane lists various R packages with their versions and download links. The code editor pane shows a script named "paths.R" with the following content:

```
1 setwd("D:/Projekte/Rpackages/germanwebr/Rfunctions")
2
```

The R console at the bottom displays the message "R is a collaborative project with many contributors. Type 'contributors()' for more information and 'citation()' on how to cite R or R packages in publications."

Package	Description	Version
AER	Applied Econometrics with R	1.2-2
arules	Mining Association Rules and Frequent Itemsets	1.1-2
bitops	Bitwise Operations	1.0-6
boot	Bootstrap Functions (originally by Angelo Canty for S)	1.3-11
brew	Templating Framework for Report Generation	1.0-6
car	Companion to Applied Regression	2.0-19
caTools	Tools: moving window statistics, GIF, Base64, ROC AUC, etc.	1.17
class	Functions for Classification	7.3-10
cluster	Cluster Analysis Extended Rousseeuw et al.	1.15.2
codetools	Code Analysis Tools for R	0.2-8
colorspace	Color Space Manipulation	1.2-4
compiler	The R Compiler Package	3.1.0
DAAG	Data Analysis And Graphics data and functions	1.18

# Bestehende Pakete und Installation

The screenshot shows the RStudio interface with the 'Packages' tab selected in the top navigation bar. Below the navigation bar, there are buttons for 'Install Packages' and 'Check for Updates'. A search bar is also present. The main area displays a list of installed packages, each with a checkbox, the package name in blue, a brief description, and its version number. The packages listed are: AER, arules, bitops, boot, and brew.

checkbox	package	description	version	x
<input type="checkbox"/>	<a href="#">AER</a>	Applied Econometrics with R	1.2-2	
<input type="checkbox"/>	<a href="#">arules</a>	Mining Association Rules and Frequent Itemsets	1.1-2	
<input type="checkbox"/>	<a href="#">bitops</a>	Bitwise Operations	1.0-6	
<input type="checkbox"/>	<a href="#">boot</a>	Bootstrap Functions (originally by Angelo Canty for S)	1.3-11	
<input type="checkbox"/>	<a href="#">brew</a>	Templating Framework for Report Generation	1.0-6	

# Übersicht Pakete:

- Luhmann - **Übersicht mit vielen nützlichen Paketen**
- Mit dem Paket leaflet kann man interaktive Karten erstellen.
- Das Paket tmap zur Erstellung von thematischen Karten.
- **Paket maptools um Karten zu erzeugen**
- Das Paket sf - bietet Zugang zu **simple features**.



# Pakete aus verschiedenen Quellen installieren

## Pakete vom CRAN Server installieren

```
install.packages("lme4")
```

## Pakete vom Bioconductor Server installieren

```
source("https://bioconductor.org/biocLite.R")
biocLite(c("GenomicFeatures", "AnnotationDbi"))
```

## Pakete von Github installieren

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

install_github("hadley/maptools")
```

# Wie bekomme ich einen Überblick?

- Entdecke Pakete, die kürzlich auf den **CRAN** Server hochgeladen wurden
- Nutze eine Shiny Web-App, in der **Pakete angezeigt werden, die kürzlich von CRAN** heruntergeladen wurden.
- Werfe einen Blick auf eine **Quick-Liste nützlicher Pakete**
- ...., oder auf eine Liste mit den **besten Paketen für die Datenverarbeitung und -analyse**,.....
- ...., oder sieh Dir **die 50 meistgenutzten Pakete** an.

# CRAN Task Views

- Bezuglich mancher Themen gibt es einen Überblick über alle wichtigen Pakete - (**CRAN Task Views**)
- Momentan gibt es 35 Task Views.
- Alle Pakete einer Task-View können mit folgendem Befehl installiert werden: **command:**

```
install.packages("ctv")
library("ctv")
install.views("Spatial")
```

## CRAN Task Views

<a href="#">Bayesian</a>	Bayesian Inference
<a href="#">ChemPhys</a>	Chemometrics and Computational Physics
<a href="#">ClinicalTrials</a>	Clinical Trial Design, Monitoring, and Analysis
<a href="#">Cluster</a>	Cluster Analysis & Finite Mixture Models
<a href="#">DifferentialEquations</a>	Differential Equations
<a href="#">Distributions</a>	Probability Distributions
<a href="#">Econometrics</a>	Econometrics
<a href="#">Environmetrics</a>	Analysis of Ecological and Environmental Data
<a href="#">ExperimentalDesign</a>	Design of Experiments (DoE) & Analysis of Experimental Data

# Übung - zusätzliche Pakete

Geh bspw. auf <https://cran.r-project.org/> und suche nach Paketen...

- die sich für interaktive Karten eignen.
- mit denen man thematische Karten erstellen kann
- mit denen man die räumliche Distanz berechnen kann
- mit denen man eine Satellitenkarte bekommen kann

# Erste Karten

# Das Paket maptools

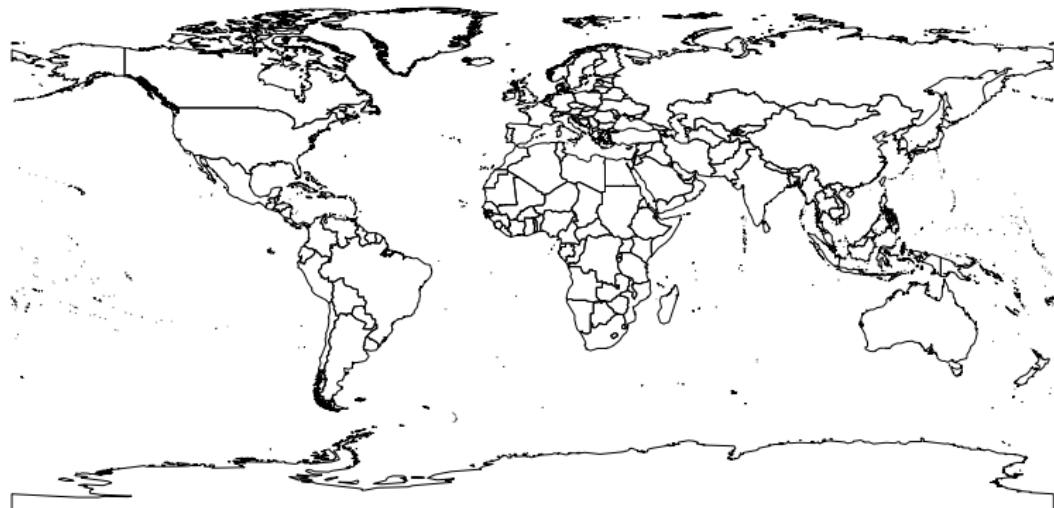
- Datensatz wrld\_simpl aus dem Paket maptools
- Polygone für fast alle Staaten der Erde

```
library(maptools)
data(wrld_simpl)
```

	FIPS	ISO2	ISO3	UN	NAME
ATG	AC	AG	ATG	28	Antigua and Barbuda
DZA	AG	DZ	DZA	12	Algeria
AZE	AJ	AZ	AZE	31	Azerbaijan
ALB	AL	AL	ALB	8	Albania

# Hello world

```
data(wrld_simpl)  
plot(wrld_simpl)
```



# Der shapefile

- Es handelt sich um einen shapefile

```
typeof(wrld_simpl)
```

```
## [1] "S4"
```

- Die Daten sind als S4 abgespeichert
- Es gibt verschiedene Slots
- In einem davon ist Information als data.frame gespeichert.

# Der Datensatz

```
head(wrld_simpl@data)
```

	FIPS	ISO2	ISO3	UN	NAME
ATG	AC	AG	ATG	28	Antigua and Barbuda
DZA	AG	DZ	DZA	12	Algeria
AZE	AJ	AZ	AZE	31	Azerbaijan
ALB	AL	AL	ALB	8	Albania

# Die Struktur der Daten

```
head(wrld_simpl@data$NAME)
```

```
## [1] Antigua and Barbuda Algeria Azerbaijan
## [4] Albania Armenia Angola
## 246 Levels: Aaland Islands Afghanistan Albania Algeria ...
```

```
head(wrld_simpl@data$ISO2)
```

```
## [1] AG DZ AZ AL AM AO
## 246 Levels: AD AE AF AG AI AL AM AN AQ AR AS AT AU AW AX
```

```
head(wrld_simpl@data$POP2005)
```

```
## [1] 83039 32854159 8352021 3153731 3017661 16095214
```

# Eine logische Abfrage

```
ind_SA <- wrld_simpl@data$NAME == "South Africa"  
head(ind_SA)
```

```
## [1] FALSE FALSE FALSE FALSE FALSE FALSE
```

```
table(ind_SA)
```

```
## ind_SA
```

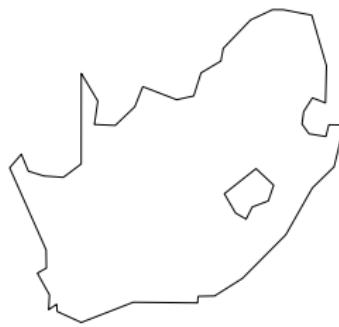
```
## FALSE TRUE
```

```
## 245 1
```

# Eine Karte für Süd Afrika

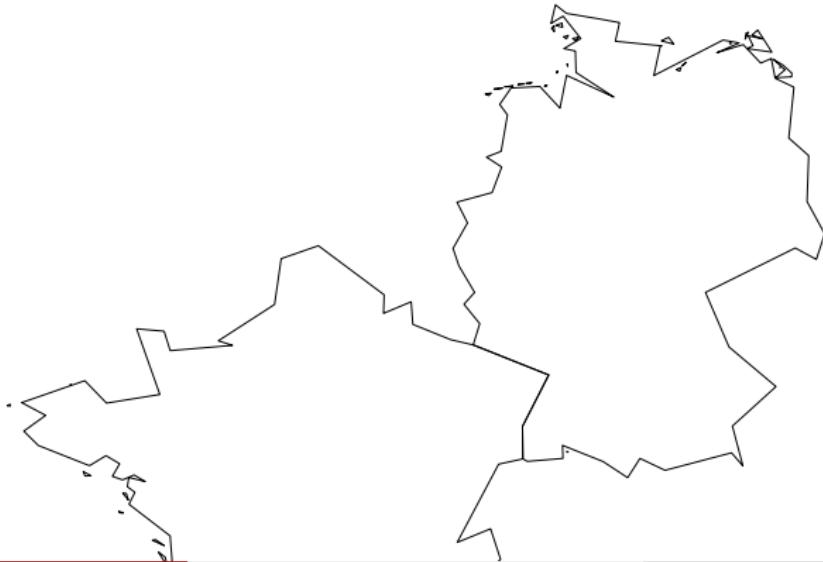
- Ein Land zeichnen

```
SouthAfrica <- wrld_simpl[ind_SA,]  
plot(SouthAfrica)
```



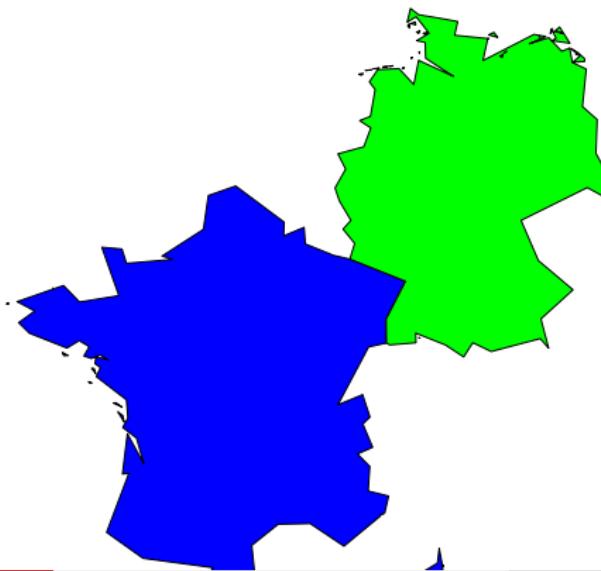
# Mehr als ein Land zeichnen

```
EuropeList <- c('Germany', 'France')
my_map <- wrld_simpl[wrld_simpl$NAME %in% EuropeList, ]
par(mai=c(0,0,0,0))
plot(my_map)
```



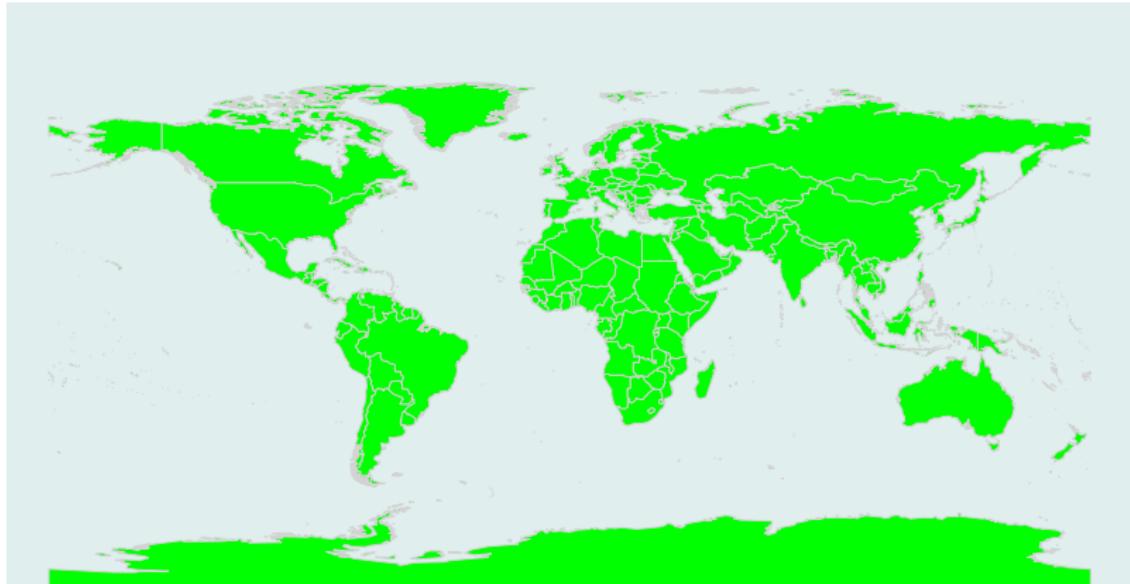
# Mehr Farbe

```
my_map@data$color <- c("blue","green")
plot(my_map,col=my_map@data$color)
```



# Mehr Farbe für die Welt

```
plot(wrld_simpl, bg='azure2', col='green',  
border='lightgray')
```



# Eine Karte für Europa

```
Europe <- wrld_simpl[wrld_simpl$REGION=="150",]  
plot(Europe,col="royalblue")
```



# Europa ohne Russland

```
ind <- which(Europe@data$NAME=="Russia")
EU <- Europe[-ind,]
plot(EU,col="blue",border="darkgray")
```



# Spielen Sie mit Farben

```
EU$colors <- "green"  
plot(EU,col=EU$colors,border="darkgray")
```



# Mehr über Farben

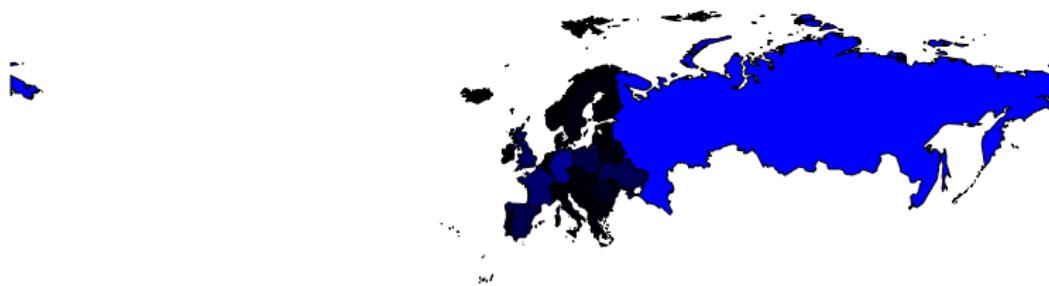
Colors in R

```
Europe$colors[pop05>median(pop05)] <- "chocolate4"  
plot(Europe,col=Europe$colors)
```



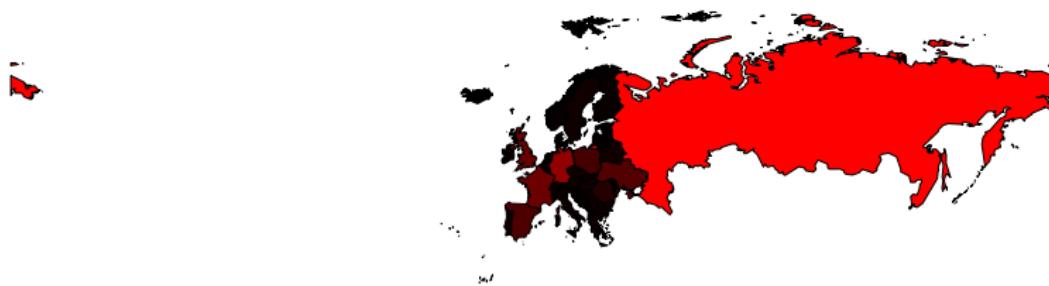
# Europa - Farbschattierung blau

```
val <- Europe$POP2005/max(Europe$POP2005)
plot(Europe,col=rgb(0,0,val))
```



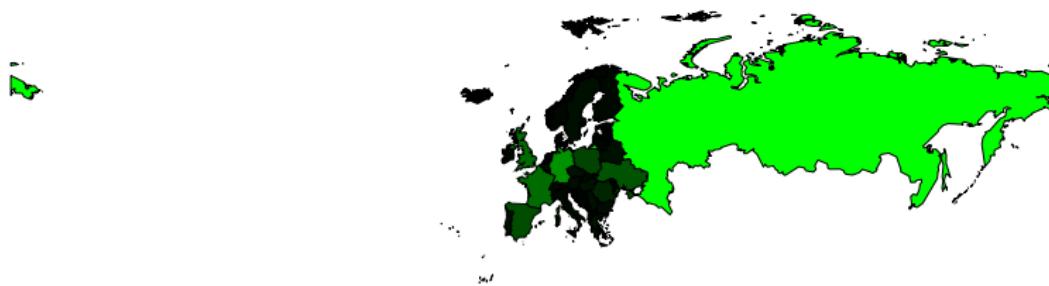
# Europa - Farbschattierung rot

```
val <- Europe$POP2005/max(Europe$POP2005)  
plot(Europe,col=rgb(val,0,0))
```



# Europa - Farbschattierung grün

```
val <- Europe$POP2005/max(Europe$POP2005)
plot(Europe,col=rgb(0,val,0))
```



# Europa - Farbschattierung grau

```
val <- Europe$POP2005/max(Europe$POP2005)
plot(Europe,col=rgb(val,val,val))
```



# Europa - zwei Graphiken nebeneinander

```
par(mfrow=c(1,2))
plot(Europe,col=rgb(val,0,val))
plot(Europe,col=rgb(val,val,0))
```



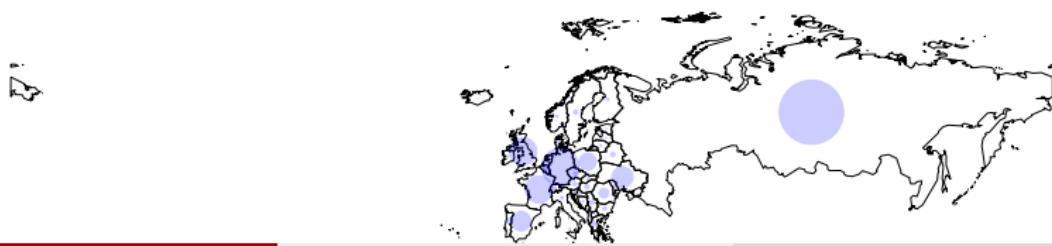
# Europa - Punkte hinzufügen

```
which(Europe$ISO2=="FR") # 10  
## [1] 10  
  
plot(Europe)  
points(Europe$LON[10],Europe$LAT[10],col="red",pch=20)
```



# Europa - Blasen hinzufügen

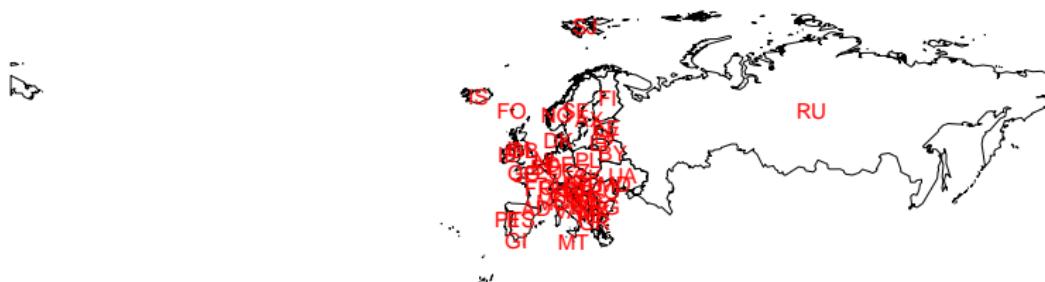
```
pop <- Europe$POP2005  
pop <- pop/max(pop)*10  
plot(Europe)  
points(Europe$LON,Europe$LAT,cex=pop,col=rgb(0,0,1,.2),  
pch=20)
```



# Europa - Text hinzufügen

```
plot(Europe)
```

```
text(Europe$LON,Europe$LAT,Europe$ISO2,col="red")
```



# Europa - Linien hinzufügen

```
which(Europe$ISO2=="FR") # 15  
which(Europe$ISO2=="DE") # 16
```

```
Dat <- cbind(Europe$LON[15:16],Europe$LAT[15:16])  
plot(Europe)  
lines(Dat,col="red",lwd=2)
```



# Geokodierung

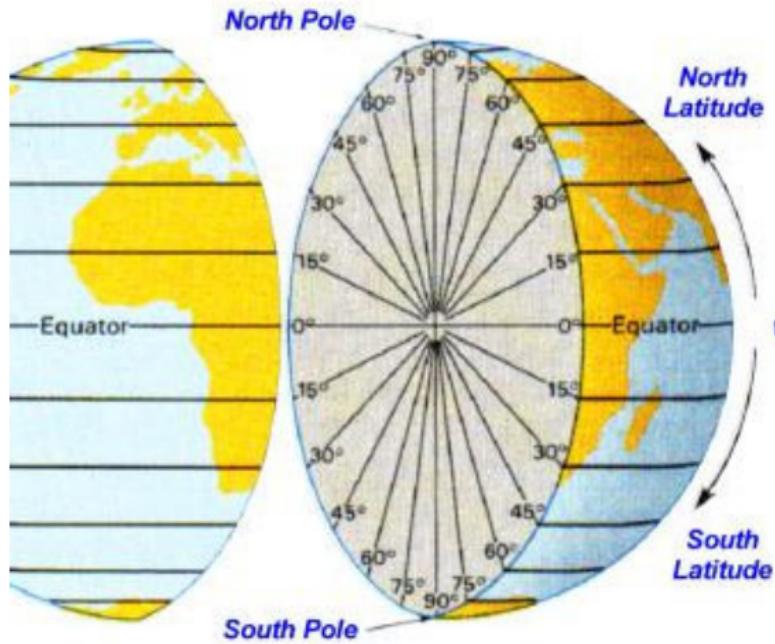
# Geokodierung

## Wikipedia - Geocoding

*Geocoding ( . . . ) uses a description of a location, most typically a postal address or place name, to find geographic coordinates from spatial reference data . . .*

# Latitude und Longitude

## LATITUDE



# Geokodierung mit dem Paket `tmaptools`

- Beim Paket `tmaptools` wird die Nominatim API zur Geokodierung verwendet.
- Diese Funktion hat den Vorteil, dass eine Projektion ausgewählt werden kann, in der die Geokodierungen zurück gegeben werden.

```
library("tmaptools")
```

```
?geocode_OSM
```

# Der Geocode für Schwäbisch-Gmünd

```
geocode_OSM("Schwäbisch-Gmünd")
```

```
## $query
## [1] "Schwäbisch-Gmünd"
##
## $coords
##           x           y
## 9.796353 48.800118
##
## $bbox
##       xmin      ymin      xmax      ymax
## 9.713714 48.714554 9.943580 48.844434
```

# Reverse Geokodierung

*Reverse geocoding is the process of back (reverse) coding of a point location (latitude, longitude) to a readable address or place name. This permits the identification of nearby street addresses, places, and/or areal subdivisions such as neighbourhoods, county, state, or country.*

Quelle: Wikipedia

# Eine Karte für einen bestimmten Ort bekommen

```
library("OpenStreetMap")  
  
map <- openmap(c(48.791510,9.809462),  
                 c(48.801510,9.829462),  
                 type="osm")  
plot(map)
```

## Die Karte für Schwäbisch Gmünd

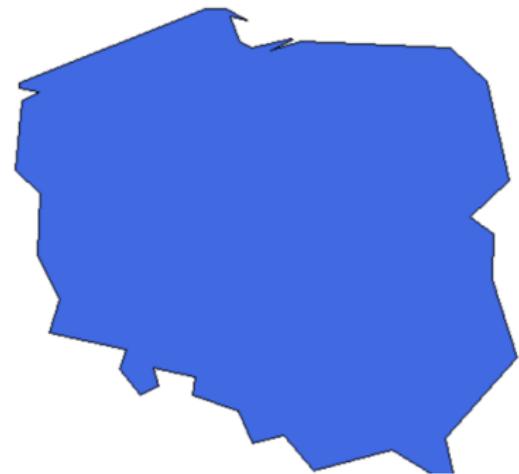


# Shapefiles bekommen, bearbeiten und visualisieren

# Worum geht es in diesem Abschnitt

- Was sind Shapefiles?
- Wie kann man Shapefiles (.shp) in R importieren?
- Der Import von Shapefiles wird anhand von Vorwahl- und PLZ-Bereichen gezeigt.
- Wie kann man einzelne Polygonzüge zusammenfassen?

# Ein kleines Quizz



# Das shapefile Format ...

- ... ist ein beliebtes Format räumlicher Vektordaten für geographisches Informationssysteme (GIS).
- Das Dateiformat Shapefile ist ein ursprünglich für die Software ArcView der Firma ESRI entwickeltes Format für Geodaten. (Quelle: [Wikipedia](#))
- Es wurde entwickelt und reguliert von **ESRI**
- (meist) offene Spezifikation um Daten Interoperabilität zwischen Esri und anderen Formaten zu sichern.
- Es können Punkte, Linien und Polygone beschrieben werden
- Jedes Element hat Attribute, wie bspw. Name oder Temperatur die es beschreiben.

Quelle: <https://en.wikipedia.org/wiki/Shapefile>

# Der R Befehl `readShapePoly`

Um Shape-Dateien zu lesen, ist es notwendig, die drei Dateien mit den folgenden Dateierweiterungen im gleichen Verzeichnis zu haben:

- .shp
- .dbf
- .shx

# Vorwahlbereiche in Deutschland

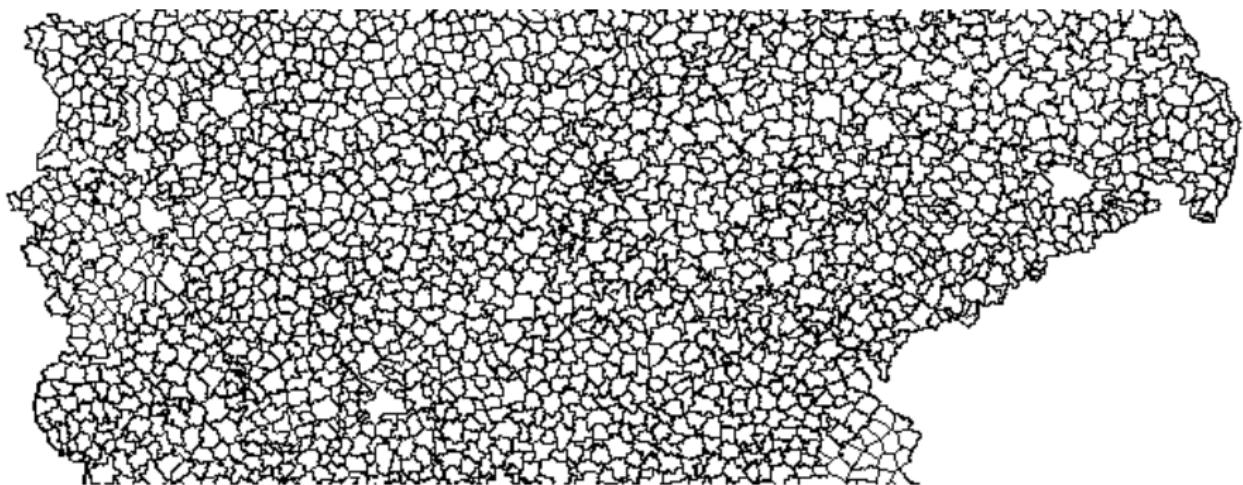
## Quelle Ortsnetzbereiche: Bundesnetzagentur

- **Download** ONB-Grenzen
- Wir verwenden das Paket `maptools` um die Daten einzulesen:

```
setwd(geodata_path)
library(maptools)
onb <- readShapePoly("onb_grenzen.shp")
```

# Die Karte zeichnen

```
plot(onb)
```



# Der Datenslot

```
kable(head(onb@data))
```

# Einen Vorwahlbereich ausschneiden

```
vwb <- onb@data$VORWAHL  
vwb2 <- substr(vwb, 1,2)
```

```
library(lattice)  
barchart(table(vwb2), col="royalblue",  
         xlab="Häufigkeit")
```

# Vorwahlbereich ausschneiden

```
vwb6 <- onb[vwb2=="06",]  
plot(vwb6)
```

# Shapefiles zusammenfassen

```
vwb6c <- unionSpatialPolygons(vwb6,  
                                rep(1,length(vwb6)))  
plot(vwb6c,col="royalblue")
```

# Wo ist Mannheim?

```
Com <- vwb6@data$NAME  
plot(vwb6)  
plot(vwb6[Com=="Mannheim",], col="red", add=T)  
plot(vwb6[Com=="Heidelberg",], col="green", add=T)  
plot(vwb6[Com=="Kaiserslautern",], col="blue", add=T)
```

# Shapefiles für PLZ-Bereiche

## Quelle für PLZ Shapefiles

### Download

The post code areas for Germany are available in Shape format in the coordinate system WGS84 (EPSG:4326) as a zipped tar file (10MB) for download at:

<http://www.metaspatial.net/download/plz.tar.gz>

# Paket rgdal - PLZ Datensatz einlesen

```
library(rgdal)
```

```
setwd(data_path)
plz <- readOGR ("post_pl.shp","post_pl")
```

```
setwd("D:/GESIS/data")
plz <- readOGR ("post_pl.shp","post_pl")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "D:\GESIS\data\post_pl.shp", layer: "post_pl"
## with 8270 features
## It has 3 fields
```

# Die Daten plotten

```
plzbereich <- substr(plz@data$PLZ99, 1, 2)  
plot(plz[plzbereich=="68",])
```



# Die Grenze von Mannheim

```
ma_map <- plz[plz$PLZORT99=="Mannheim",]  
plot(ma_map)
```



# Die PLZ-Bereiche von Mannheim zusammenfassen

- Wir nutzen den Befehl `unionSpatialPolygons` im Paket `maptools`

```
library(maptools)
ma_map2 <- unionSpatialPolygons(SpP = ma_map,
                                 IDs = rep(1,length(ma_map)))
plot(ma_map2)
```



# Exkurs: der Befehl agrep

```
agrep("Freiburg",plz@data$PLZORT99)
```

```
## [1] 363 660 661 1349 5074 5798 5799 5800 5801 5802 5803  
## [15] 5807 5808 5809
```

```
agrep("Freiburg",plz@data$PLZORT99,value=T)
```

```
## [1] "Freyburg/ Unstrut"      "Freiberg"           "Freiber  
## [4] "Freiburg (Elbe)"       "Freiberg am Neckar"   "Freibur  
## [7] "Freiburg im Breisgau"  "Freiburg im Breisgau"  "Freibur  
## [10] "Freiburg im Breisgau" "Freiburg im Breisgau"  "Freibur  
## [13] "Freiburg im Breisgau" "Freiburg im Breisgau"  "Freibur  
## [16] "Freiburg im Breisgau" "Freiburg im Breisgau"
```

# Die Funktion grep

## Der exakte match

```
grep("Freiburg",plz@data$PLZORT99,value=T)
```

```
## [1] "Freiburg (Elbe)"      "Freiburg im Breisgau" "Freibur  
## [4] "Freiburg im Breisgau" "Freiburg im Breisgau" "Freibur  
## [7] "Freiburg im Breisgau" "Freiburg im Breisgau" "Freibur  
## [10] "Freiburg im Breisgau" "Freiburg im Breisgau" "Freibur  
## [13] "Freiburg im Breisgau"
```

```
agrep("Freiburg",plz@data$PLZORT99,value=T,  
      max.distance = 0.2)
```

```
## [1] "Frohburg"           "Freyburg/ Unstrut"   "Freiber  
## [4] "Freiberg"            "Freiburg (Elbe)"    "Ehrenbu  
## [7] "Gnarrenburg"         "Bad Driburg"       "Derenbu  
## [10] "Freiberg am Neckar" "Freiburg im Breisgau" "Freibur  
## [13] "Freiburg im Breisgau" "Freiburg im Breisgau" "Freibur  
## [16] "Freiburg im Breisgau" "Freiburg im Breisgau" "Freibur
```

# Aufgabe: PLZ Bereiche herunterladen

- Lade den Shapefile mit den PLZ-Bereichen **hier** herunter.
- Importiere den Shapefile in R mit einem geeigneten Befehl.
- Erzeuge einen Datensatz mit den PLZ-Bereichen von Berlin.
- Speichere den Datensatz als .RData Datei ab.

# Global Administraive Boundaries - GADM - NUTS level 1

- Für Polygonzüge unterhalb der Staatsgrenzen ist **Global Administrative Boundaries** eine gute Quelle.
- Vor allem wegen API, die man Paket raster nutzen kann.

```
library(raster)
LUX1 <- getData('GADM', country='LUX', level=1)
plot(LUX1)
```



# Ein Blick auf die Daten

Koordinaten im polygon slot

```
LUX1@polygons[[1]]@Polygons[[1]]@coords
```

```
##           x         y
## [1,] 6.238343 49.78491
## [2,] 6.238727 49.78969
## [3,] 6.238657 49.79102
## [4,] 6.238348 49.79232
## [5,] 6.238039 49.79295
## [6,] 6.237128 49.79403
```

# Der Datenslot

```
head(LUX1@data)
```

```
##      GID_0      NAME_0      GID_1      NAME_1      VARNAME_
## 1    LUX Luxembourg LUX.1_1 Diekirch      Dikrech|Dikkri
## 2    LUX Luxembourg LUX.2_1 Grevenmacher  Gréivemaache
## 3    LUX Luxembourg LUX.3_1 Luxembourg Létzebuerg|Luxembu
##      TYPE_1 ENGTYPEn_1 CC_1 HASC_1
## 1 District   District <NA>  LU.DI
## 2 District   District <NA>  LU.GR
## 3 District   District <NA>  LU.LU
```

# GADM- NUTS level 3

```
LUX3 <- getData('GADM', country='LUX', level=3)
plot(LUX3)
```



# GADM- NUTS level 4

```
LUX4 <- getData('GADM', country='LUX', level=4)
plot(LUX4)
```



# GADM- NUTS level 3

```
DEU3 <- getData('GADM', country='DEU', level=3)  
plot(DEU3)
```



# Gemeinden in Deutschland

Bundesamt für Kartographie und Geodäsie (BKG)

```
krs <- maptools::readShapePoly("vg250_krs.shp")
plot(krs)
```



# Aufgabe: Download von Shapefiles für die Schweiz

- Lade die Umrisse der NUTS2 Regionen für die Schweiz herunter.
- Zeichne eine Karte für die Kreise im Kanton Aarau .

# Kreise eines Bundeslandes

```
fds <- substr(krs@data$AGS, 1, 2)
plot(krs[fds=="05",])
```



# Das Paket maps - Mehr Information

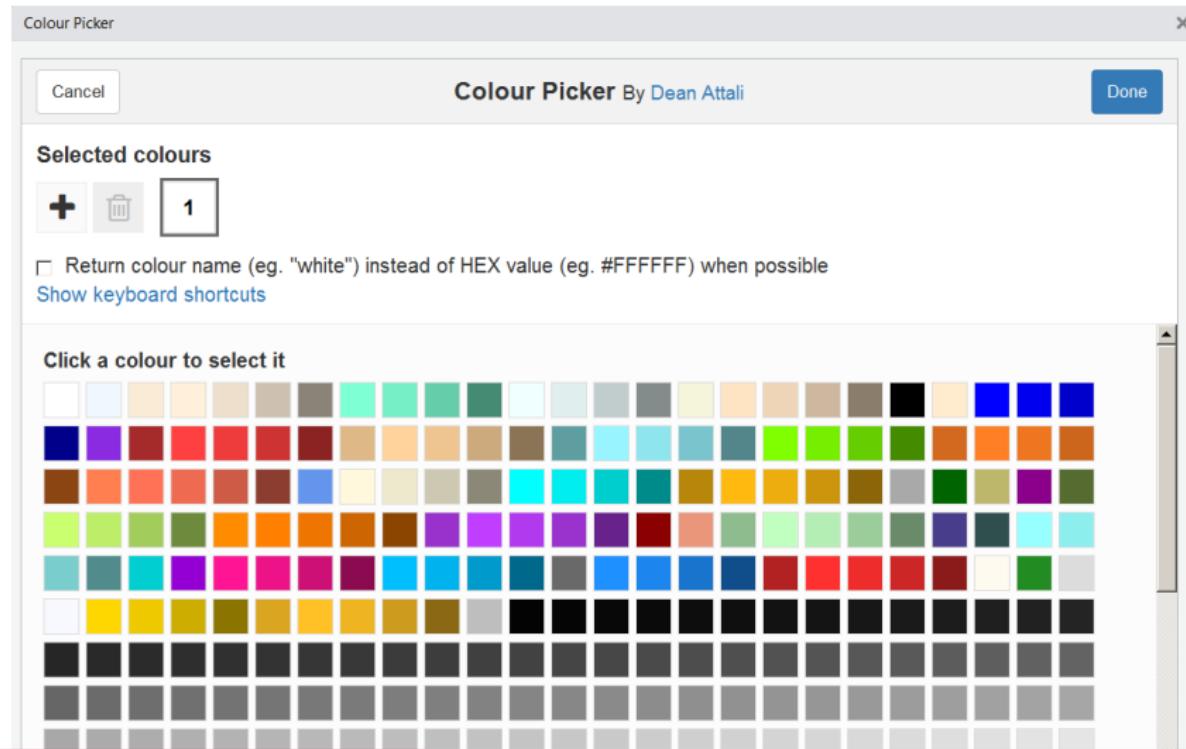
- Nur für manche Staaten bekommt man mit dem Paket `maps` Umkreise für Einheiten unterhalb der Staatsgrenze (bspw. Frankreich, USA).

```
library(maps)
data(world.cities)
map("france")
map.cities(world.cities,col="blue")
```



# Das Rstudio Addin colourpicker

```
install.packages("colourpicker")
```



# Weitere Quelle - Shapefiles für Wahlkreise

Der  
Bundeswahlleiter[English](#) |  [Leichte Sprache](#) |  [Gebärdensprache](#) | [PDF](#)[Bundestagswahl](#)[Europawahl](#)

## Bundestagswahl 2017

[Startseite](#) > [Bundestagswahl 2017](#) > [Wahlkreise](#) > [Downloads](#)

## Karte der Wahlkreise zum Download

Die Karte mit der Wahlkreisgeometrie zur aktuellen Bundestagswahl steht Ihnen in verschiedenen Formaten zum Download zur Verfügung. Unter Berücksichtigung des unten angegebenen Copyright-Vermerkes<sup>1</sup> ist es Ihnen gestattet, die Inhalte uneingeschränkt zu verwenden.

[Dateiformat](#)[Bildformat](#)[Inhalt](#)[Geometrie<sup>2</sup>](#)

# Shapefiles bei Eurostat

- **Eurostat Karten** - in der Regel die Europäischen Mitgliedsstaaten

 eurostat  
Ihr Schlüssel zur europäischen Statistik

Anmelden | Registrieren 

Rechtlicher Hinweis | RSS | Cookies | Links | Kontakt | Deutsch

Geben Sie ein Stichwort, einen Kode, einen Titel, ... ein 

**Neuigkeiten**      **Daten**      **Veröffentlichungen**      **Über Eurostat**      **Hilfe**

[Europäische Kommission](#) > Eurostat > GISCO: Geographische Informationen und Karten N... > Geografische Daten > Geobasisdaten  
 > Verwaltungseinheiten / Statistische Einheiten > Zensus

GISCO: GEOGRAPHISCHE INFORMATIONEN UND KARTEN									
ZENSUS									
Zensus									
Übersicht									
+ Geografische Daten	Bitte beachten Sie, dass die fraglichen Bestimmungen für die jeweiligen Datensätze einzuhalten. Das Herunterladen und die Verwendung dieser Daten ist abhängig von deren Abnahme.								
+ GISCO-Aktivitäten									
Häufig gestellte Fragen									
Verwaltungs-einheiten/ Statistische Einheiten	Maßstab	Erhebungs- umfang	Objektart	Format	Zeitraum	Koordinaten-referenz- system	Versionsdatum	Datei zum herunterladen	
Geometrischer Schwerpunkt der Zensus 2011	Alle	Europa	Point	Personal GDB	2011	ETRS89	08/10/2015	<a href="#">CENSUS_UNITS_PT_2011.zip</a>	
	Alle	Europa	Point	Shapefile	2011	ETRS89	08/10/2015	<a href="#">CENSUS_UNITS_PT_2011_SH.zip</a>	
Zensus-Einheiten 2011	1:1 Mio	Europe	Polygon	Personal GDB	2011	ETRS89	08/10/2015	<a href="#">CENSUS_UNITS_RG_01M_2011.zip</a>	

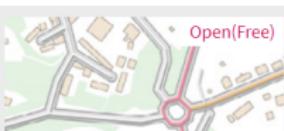
# Weitere Quellen für Shapefiles

- Open linked data - Ordnance Survey (GB)



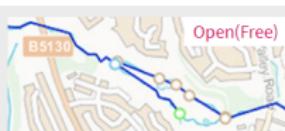
OS Names API ›

Addressing and Location



OS Open Roads ›

Networks



OS Open Rivers ›

Networks



OS Terrain 50 ›

Height

- World Borders Datensatz
- National Historical Information System
- Freie Polygon-Daten für die USA
- Überblick über - Spatial Data in R

# Interaktive Karten erstellen

# Beispiel zu Campingplätzen

- Die Daten stammen von:

<http://www.openstreetmap.de/>

- Dabei wird die Overpass API genutzt:

[http://wiki.openstreetmap.org/wiki/Overpass\\_API](http://wiki.openstreetmap.org/wiki/Overpass_API)

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

```
CampSites <- read.csv(url)
```

# Überblick über Daten zu Campingplätzen

X	name	tourism	website
1	Campingplatz Winkelbachtal	camp_site	<a href="http://www.gruibirg.de">http://www.gruibirg.de</a>
2	Radler-Zeltplatz	camp_site	NA
3	Campingplatz des Naturfreundehauses	camp_site	NA
4	Campingplatz Am Aichstruter Stausee	camp_site	NA
5	NA	camp_site	NA
6	Kandern	camp_site	NA
7	Campingplatz Baiersbronn-Obertal	camp_site	NA
8	Campingplatz Schwabenmühle	camp_site	NA

# Notwendige Pakete

magrittr - für den Pipe Operator in R:

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

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

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

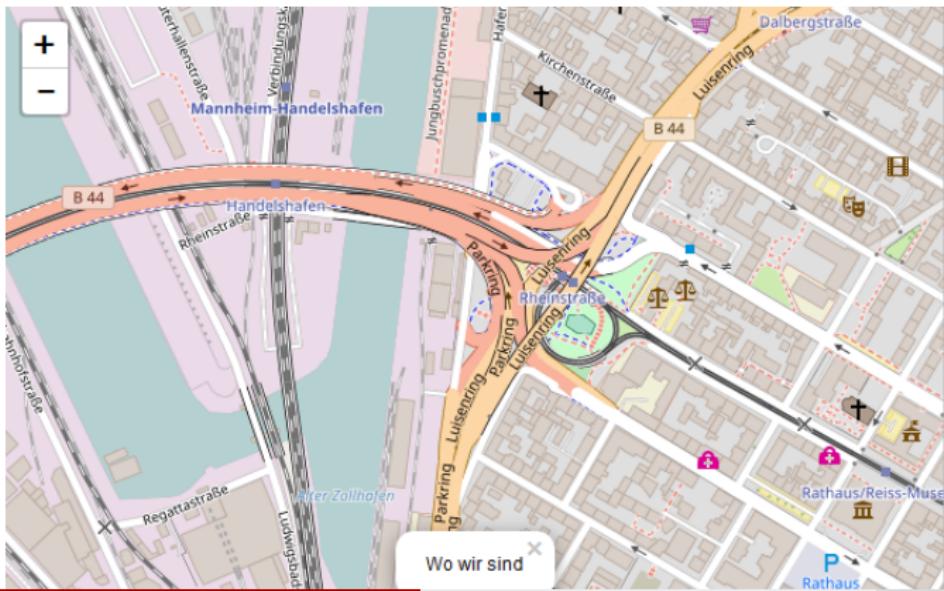
# Eine erste interaktive Karte

```
leaflet()%>%  
  addTiles()
```



# Auf eine Stadt zoomen

```
leaflet() %>%  
  addTiles() %>%  
  addMarkers(lng=8.456597, lat=49.48738,  
             popup="Wo wir sind")
```



# Eine interaktive Karte

```
m <- leaflet() %>%
  addTiles() %>%
  addMarkers(lng=CampSites$lon,
             lat=CampSites$lat,
             popup=CampSites$name)
m
```

# Das Paket leaflet - Visualisierung von Geokodierung

```
library("tmaptools")
gc_tma <- geocode_OSM("Mannheim, GESIS")
```

```
library(leaflet)
library(magrittr)
m <- leaflet() %>%
addTiles() %>%
addMarkers(lng=8.463061 , lat=49.485736 ,
           popup="GESIS Mannheim")
m
```

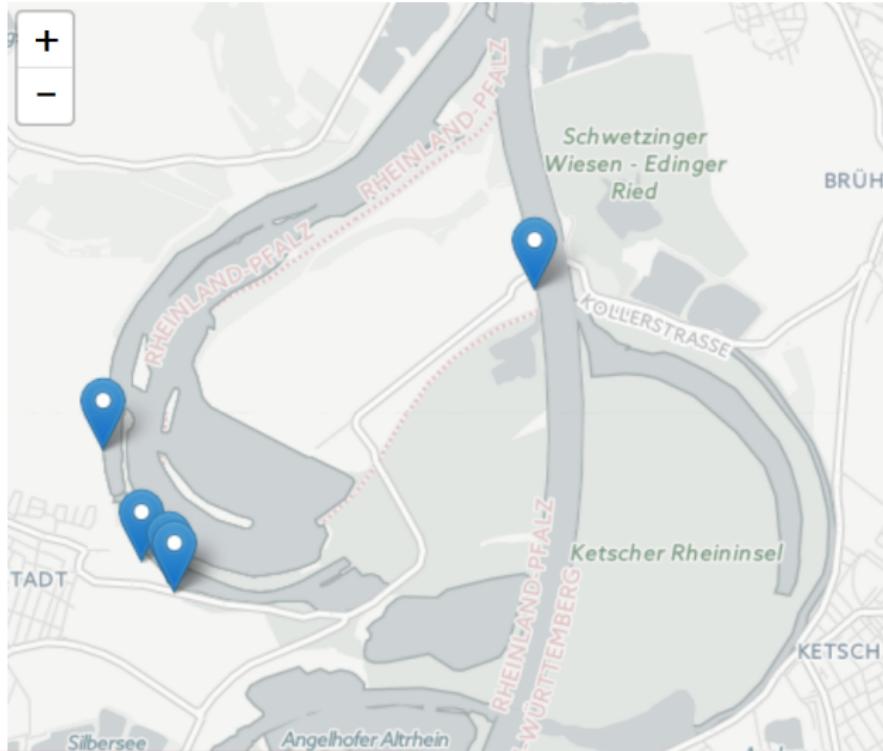
# Stamen als Hintergrundkarte

```
m %>% addProviderTiles("Stamen.Toner")
```



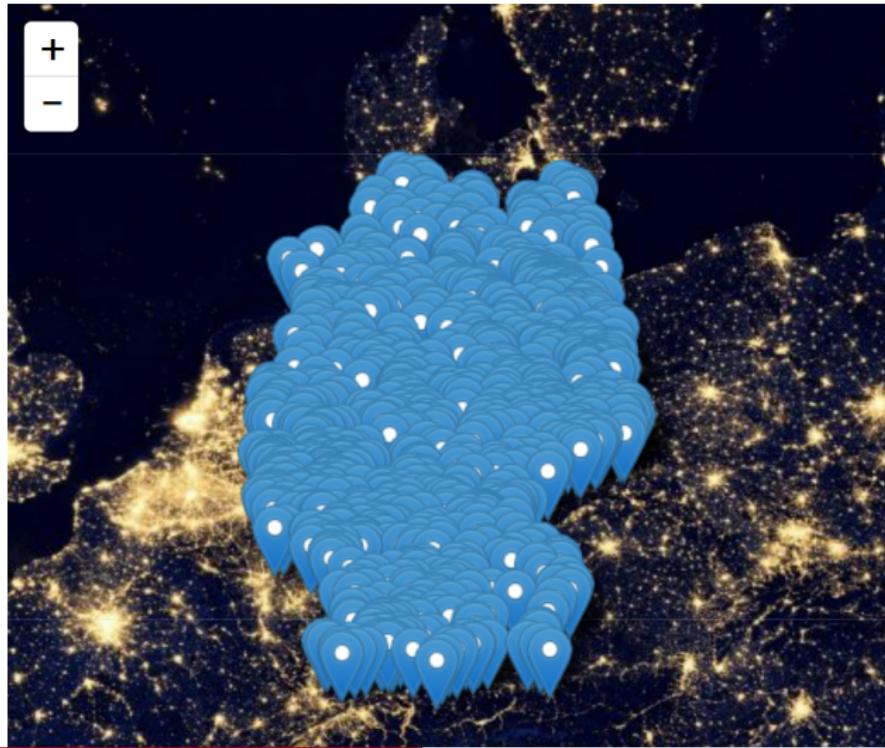
# CartoDB als Hintergrund

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



# Mehr Hintergründe

```
m %>% addProviderTiles("NASAGIBS.ViirsEarthAtNight2012")
```



# Mehr Informationen hinzufügen

```
popupInfo <- paste(CampSites$name, "\n", CampSites$website)
```

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

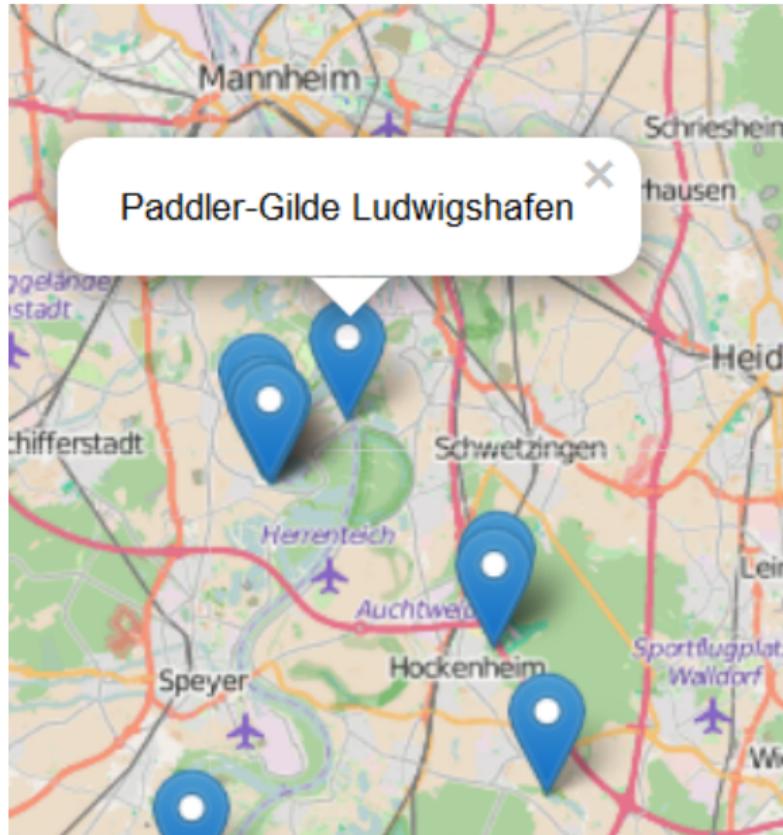
Das Ergebnis ist hier:

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

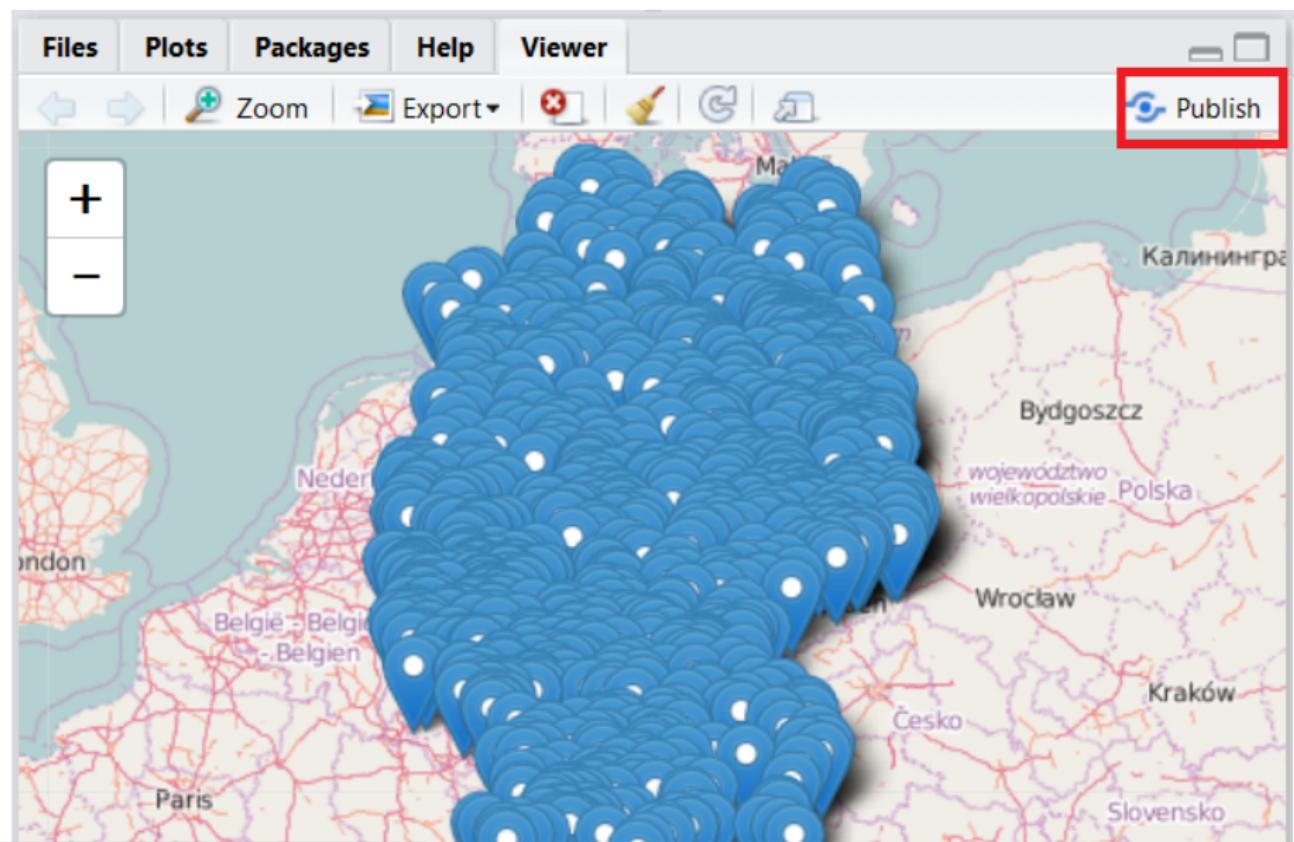
# Die resultierende Karte



# Popups in einer interaktiven Karte



# Wie man auf Rpubs publizieren kann



# Ein weiteres Beispiel - Weltkulturerbe

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

# Eine interaktive Karte erstellen

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

# Die Karte zeigen



# Farbe hinzufügen

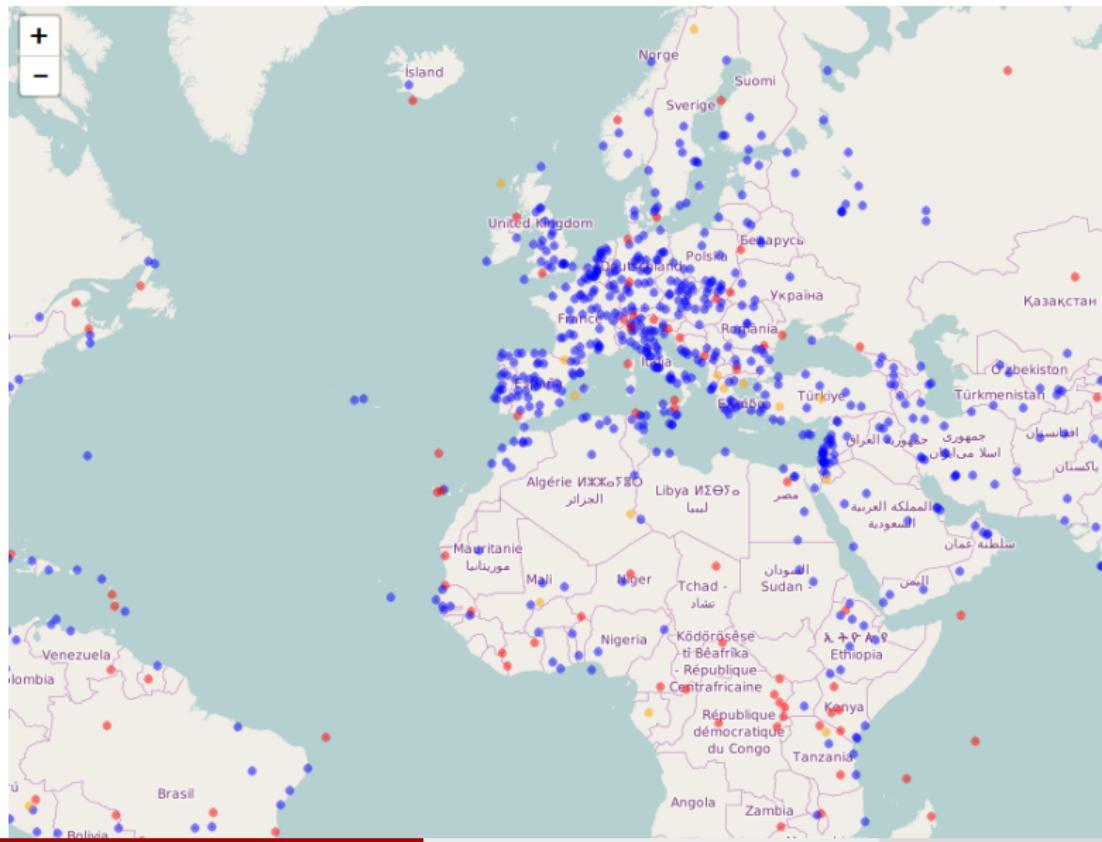
```
whcSites$color <- "red"  
whcSites$color[whcSites$category=="Cultural"] <- "blue"  
whcSites$color[whcSites$category=="Mixed"] <- "orange"
```

# Eine Karte mit Farbe erzeugen

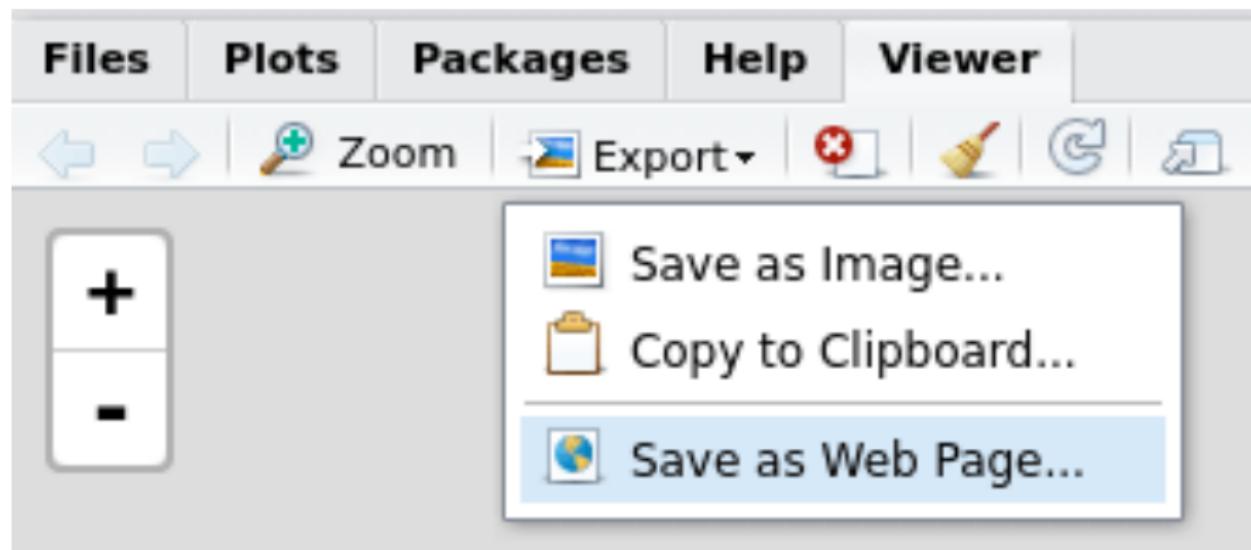
```
m1 <- leaflet() %>%
  addTiles() %>%
  addCircles(lng=whcSites$lon,
             lat=whcSites$lat,
             popup=whcSites$name_en,
             color=whcSites$color)
```

```
m1
```

# Die Karte zeigen



# Die Karte abspeichern



# Links und Quellen

- **R-bloggers** Artikel zu Leaflet
- **Einführung in Leaflet für R**
- **Offline Karten mit RgoogleMaps und leaflet**