

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

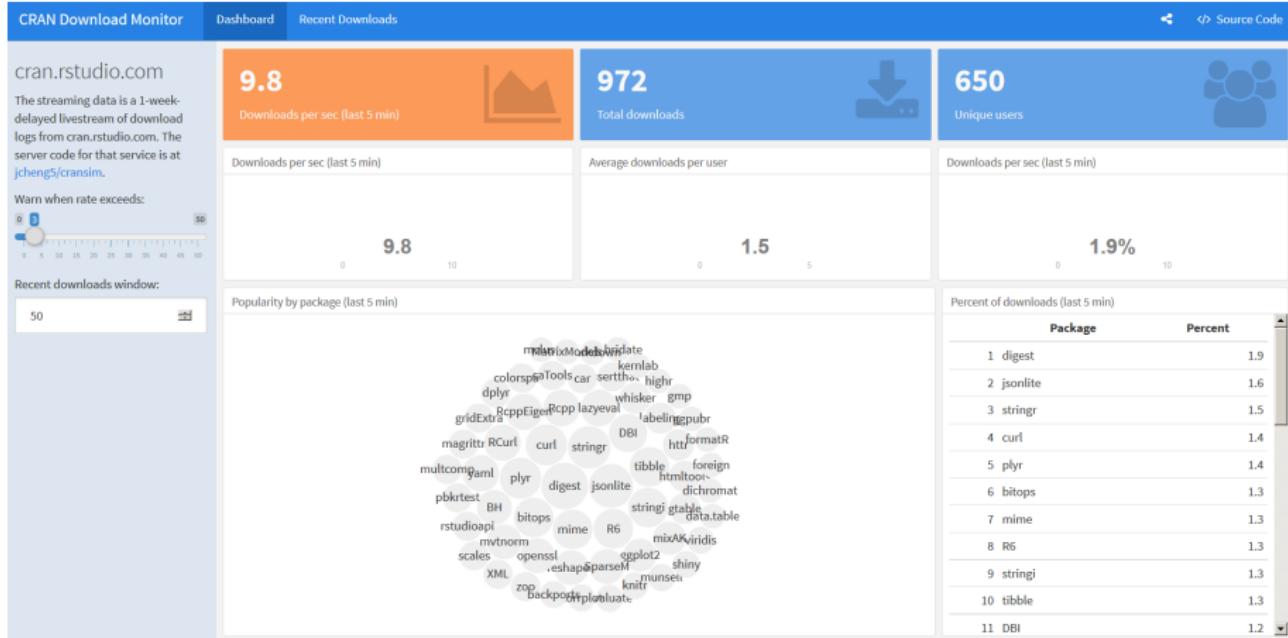
R for Stata Users

IBM SPSS Statistics Essentials for R

Users

- [Download IBM SPSS Statistics Essentials for R files](#)
- [Donate money](#)
- [Project detail and discuss](#)
- [Get support](#)

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.

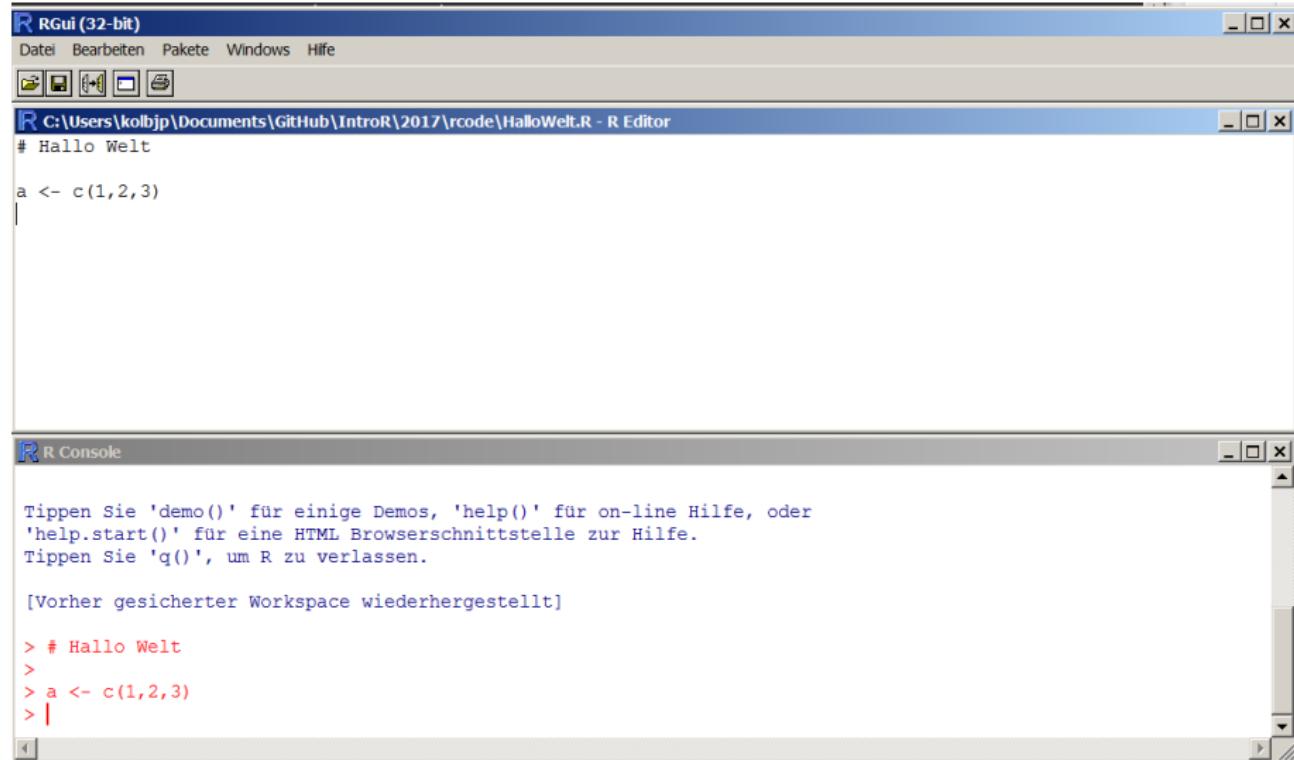
Source Code for all Platforms

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:



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, CambridgeUS font theme, and various output formats.

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
- Erzeuge ein neues R Script (Ctrl + Shift + N)
- Das Datum bekommt man mit der Funktion `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

Erstelle einen Vektor b mit den Zahlen von 1 bis 5 und berechne....

- 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.

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 a summary, a link to the full question page, and related tags like r, r-faq, ggplot2, dataframe, and plot. To the right, there's a sidebar for the R Language documentation with 22,187 frequent questions tagged. Other sections include "Related Tags" and a footer with links to help, privacy policy, and terms of service.

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.

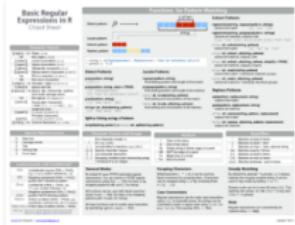
dplyr::select
Use a particular function from a package.

data(iris)
Load a built-in dataset into the environment.

Vectors			Programming	
Creating Vectors			For Loop	
Vector Functions			While Loop	
c(2, 4, 6)	2 4 6	Join elements into a vector	<pre>for (variable in sequence){ Do something }</pre>	Example
2:6	2 3 4 5 6	An integer sequence	<pre>for (i in 1:4){ j <- i + 10 print(j) }</pre>	<pre>while (i < 5){ print(i) i <- i + 1 }</pre>
seq(2, 3, by=0.5)	2.0 2.5 3.0	A complex sequence		Example
rep(1:2, times=3)	1 2 1 2 1 2	Repeat a vector		
rep(1:2, each=3)	1 1 1 2 2 2	Repeat elements of a vector		
Selecting Vector Elements			If Statements	
By Position			Functions	
x[4]	The fourth element.		<pre>function_name <- function(var){ Do something return(new_variable) }</pre>	Example
x[-4]	All but the fourth.		<pre>square <- function(x){ squared <- x*x return(squared) }</pre>	Example
x[2:4]	Elements two to four.			
x[-(2:4)]	All elements except two to four.			
x[c(1, 5)]	Elements one and five.			
Reading and Writing Data				
Input	Output		Description	
<code>df <- read.table('file.txt')</code>	<code>write.table(df, 'file.txt')</code>		Read and write a delimited text file.	

Mehr Schummelzettel

Regular Expressions



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

[DOWNLOAD](#)

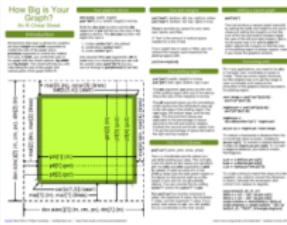
The leaflet package



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

[DOWNLOAD](#)

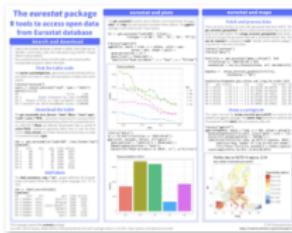
How big is your graph?



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

[DOWNLOAD](#)

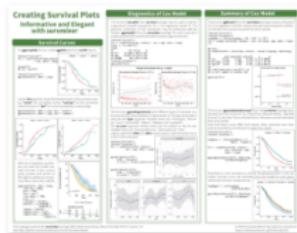
The eurostat package



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:

```
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
```

Getting Help

The Workspace

Input/Output

Packages

Graphic User Interfaces

Customizing Startup

Publication Quality Output

Batch Processing

Reusing Results

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 Fr 11 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 on the left contains R code to set the working directory.

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

Environment is empty

Packages	Version
AER	1.2-2
arules	1.1-2
bitops	1.0-6
boot	1.3-11
brew	1.0-6
car	2.0-19
caTools	1.17
class	7.3-10
cluster	1.15.2
codetools	0.2-8
colorspace	1.2-4
compiler	3.1.0
DAAG	1.18

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.

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', 'Check for Updates', and a search icon. The main area displays a list of installed packages with their details:

Package	Description	Version	Action
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	

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

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

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

Warum die räumliche Perspektive

Warum die Darstellung in Karten

- Darstellung in Karten ermöglicht besseres Verständnis von sozialwissenschaftlicher Phänomene - Attraktiver Output
- Durch die INSPIRE Richtlinie und *Collaborative Mapping* wächst der verfügbare Bestand an Geodaten.
- Daten sind oft frei verfügbar im Internet (z.B. Nutzung von APIs)
- Allerdings sind die Daten oft wenig strukturiert, heterogen oder nicht zur räumlichen Visualisierung vorgesehen
- beinhalten aber implizit geographische Informationen (Web 2.0)
- Oftmals sind wenig oder keine Metadaten vorhanden

Das Thema Geodatenlandschaft

Georeferenzierung von Daten

Situation und Zukunft der Geodatenlandschaft in Deutschland

Herausgegeben vom Rat für Sozial- und Wirtschaftsdaten

RatSWD.

Rat für Sozial- und
WirtschaftsDaten

GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung

Was heißt das für diesen Kurs

Vorgestellt werden:

- Möglichkeiten für den Download, den Import, die Verarbeitung und die Visualisierung von Geodaten
- Quellen für Geodaten
- Eine Programmierschnittstelle (APIs) um die Daten zu bekommen
- R-Pakete um diese Daten zu verarbeiten und zu visualisieren

OpenStreetMap

Inhalt dieses Abschnitts

- Vorstellung des Openstreetmap (OSM) Projekts
- Welche OSM-Daten sind erhältlich und wie kann man diese bekommen?
- Präsentation des Key-Value Schemas, dass bei OSM Daten verwendet wird.

OpenStreetMap Projekt

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

OpenStreetMap.org ist ein im Jahre 2004 gegründetes internationales Projekt mit dem Ziel, eine freie Weltkarte zu erschaffen. Dafür sammeln wir weltweit Daten über Straßen, Eisenbahnen, Flüsse, Wälder, Häuser und vieles mehr.

OpenStreetMap

Wikipedia - OpenStreetMap

OpenStreetMap (OSM) ist ein kollaboratives Projekt um eine editierbare Weltkarte zu erzeugen.

[street](#) [png](#) [maps](#) [osm](#) [svg](#) [google maps](#) [screenshot](#) [tile](#) [cambridge](#) [openstreetmap.org](#) [api](#) [openstreet](#)



OpenStreetMap – Wikipedia
de.wikipedia.org



DE Browsing - OpenStreetMap Wiki
wiki.openstreetmap.org



OpenStreetMap - Wikipedia
en.wikipedia.org



Openstreetmap: Die free Weltkarte richtig nutzen...
insights.info



Why would you use OpenStreetMap i...
geoawesomeness.com



The map you see on OpenStreetMap ...
blog.openstreetmap.org



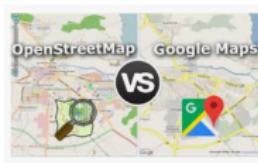
Vandalismusfall bei OpenStreetMap - Pro-Linux
pro-linux.de



OpenStreetMap - Wikipedia
en.wikipedia.org



Applications of OpenStreetMap - OpenStreetMap Wiki
wiki.openstreetmap.org



Google Maps vs OpenStreetMap: Which is the Best W...
monde-geospatial.com



OpenStreetMap Alternatives and Similar ...
alternativeo.net

Openstreetmap Tags



- Mit dem key wird eine Kategorie bezeichnet.
- Der value wird zur Beschreibung der jeweiligen Form verwendet.
- So gibt es bspw. zahlreiche OSM-Objekte mit dem key highway. Dabei kann es sich bspw. um einen Fußweg (value=pathway) oder um eine Autobahn (value=motorway) handeln.

OSM Map Features

Amenity

Used to map facilities used by visitors and residents. For example: toilets, telephones, banks, pharmacies, cafes, parking and schools. See the page [Amenities](#) for an introduction on its usage.

Key	Value	Element	Comment	Rendering	Photo
Sustenance					
amenity	bar		Bar is a purpose-built commercial establishment that sells alcoholic drinks to be consumed on the premises. They are characterised by a noisy and vibrant atmosphere, similar to a party and usually don't sell food. See also the description of the tags amenity=pub;bar;restaurant for a distinction between these.		
amenity	bbq		BBQ or Barbecue is a permanently built grill for cooking food, which is most typically used outdoors by the public. For example these may be found in city parks or at beaches. Use the tag fuel=* to specify the source of heating, such as fuel=wood;electric;charcoal . For mapping nearby table and chairs, see also the tag tourism=picnic_site . For mapping campfires and firepits, instead use the tag leisure=firepit .		
amenity	biergarten		Biergarten or beer garden is an open-air area where alcoholic beverages along with food is prepared and served. See also the description of the tags amenity=pub;bar;restaurant . A biergarten can commonly be found attached to a beer hall, pub, bar, or restaurant. In this case, you can use biergarten=yes additional to amenity=pub;bar;restaurant .		
amenity	cafe		Cafe is generally an informal place that offers casual meals and beverages, typically, the focus is on coffee or tea. Also known as a coffeehouse/shop , bistro or sidewalk cafe . The kind of food served may be mapped with the tags cuisine=* and diet=* . See also the tags amenity=restaurant;bar;fast_food .		
amenity	drinking_water		Drinking water is a place where humans can obtain potable water for consumption. Typically, the water is used for only drinking. Also known as a drinking fountain or water tap .		

Objekttypen in OSM

Es gibt prinzipiell drei verschiedene Objekttypen:

- Es gibt einfache nodes, also Punkte. Das kann bspw. eine Haltestelle des öffentlichen Nahverkehrs sein.
- Der zweite Objekttyp sind Wege. Dadurch wird beispielsweise der Verlauf von Straßen oder Flüssen beschrieben.
- Der dritte Objekttyp sind Relationen. Dabei handelt es sich um eine Gruppierung von Objekten, die in einem logischen Zusammenhang stehen.

Hippolyte Pruvost and Peter Mooney: Exploring Data Model Relations in OpenStreetMap

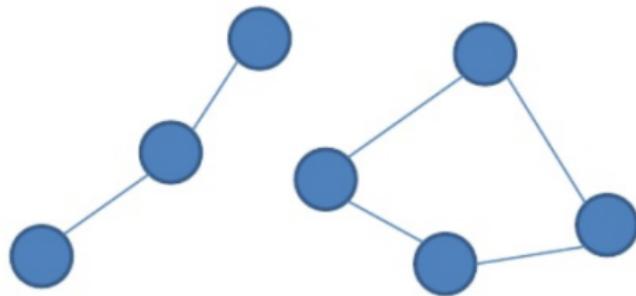
OpenStreetMap objects

Martijn van Exel - nodes and ways



Nodes

Points with (X,Y) and attributes

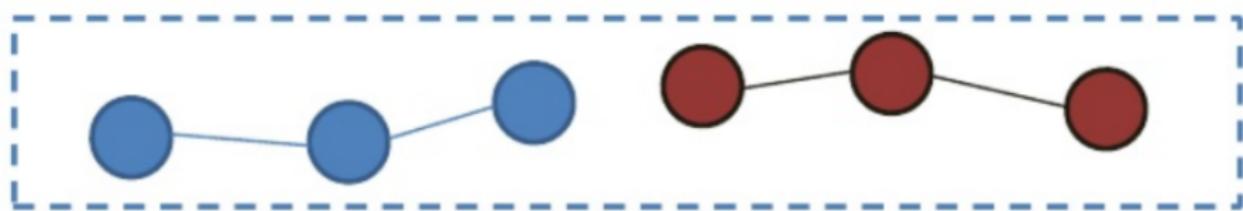


Ways

Ordered set of nodes. Polygons
are closed ways, not a separate
entity. Can have attributes.
)

OpenStreetMap objects

Relations



Relations

Groups of objects with a functional
or geographic relationship

)

Möglichkeiten die OSM-Daten zu bekommen

- Es gibt verschiedene Möglichkeiten die OSM-Daten zu bekommen.
- Metro Extracts bietet rechteckige Ausschnitte für eine Stadt oder Region von Interesse auf der ganzen Welt.
- Grundsätzlich werden die Rohdaten im *Protocolbuffer Binary Format* (PBF) oder im *Extensible Markup Language* (XML) Format angeboten.
- Die **Keyhole Markup Language** (KML) ist eine Auszeichnungssprache zur Beschreibung von Geodaten. Diese wurde bei Google Earth angewendet. KML befolgt die XML-Syntax.

Download von OpenStreetMap Daten - Metro extracts

- Ausschnitte von OpenStreetMap für einzelne Städte (**metro extracts**)

metro extracts

Each week, Metro Extracts automatically creates snapshots of OpenStreetMap data into manageable, metro-area files in a variety of formats for you to use. Download an extract from the list of popular extracts below to get started right away, or request a new extract of anywhere in the world!

[Your Custom Extracts](#) | [Documentation](#) | [Tutorial](#) | [File Format Guide](#)

SEARCH FOR A CITY OR REGION

SEARCH

POPULAR EXTRACTS:

ABU DHABI

Dubai

ALGERIA

Algiers

ANGOLA

Luanda

ARGENTINA

Buenos Aires

AUSTRALIA

Adelaide

Brisbane



Download von OpenStreetMap Daten - Geofabrik

Geofabrik

- Eine Alternative zum Download von großen OSM-Ausschnitten ist die **Geofabrik**-Seite.
- Hier lassen sich ebenfalls aktuelle Ausschnitte aber auch Shapefiles herunterladen.



GEOFABRIK // Start
geofabrik.de



GEOFABRIK // Start
geofabrik.de



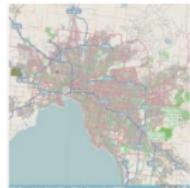
Geofabrik on Twitter: "Free Geofabrik...
[@geofabrik](https://twitter.com/geofabrik)



GEOFABRIK // Routing
geofabrik.de



LearnOSM
learnosm.org



GEOFABRIK // OpenStreetMap-Ges...
geofabrik.de

)

OSM Planet file

Datenbanklösungen

- Bei den eben vorgestellten Möglichkeiten geht es vor allem um das Herunterladen kleiner Ausschnitte.
- Wenn größere Datenmengen benötigt werden, sollte man eine Datenbanklösung nutzen.
- **PostgreSQL** hat den Vorteil, dass es Open-Source ist.

Download PostgreSQL

- Hier ist eine Einführung in PostgreSQL zu finden

The screenshot shows the Aqua Data Studio 15.0.0 interface. On the left, there's a tree view of a PostgreSQL 8.4.1 database named 'ivan'. Under 'Databases', it lists 'BI', 'datatype', 'information_schema', 'ivan', 'm1', 'pg_catalog', and 'public'. Under 'Tables' for the 'public' schema, it lists various tables like 'amay_1', 'bi_table', etc., and specifically highlights 'orders'. The main window displays the 'Alter: Table: orders' dialog. The 'General' tab is selected, showing the database is 'ivan', schema is 'public', and the table name is 'orders'. The table structure is listed in a table:

Column Name	Data Type	Length	Allow Nulls
1 orderid	int4		✓
2 customerid	char	5	✓
3 employeeid	int4		✓
4 orderdate	timestamp		✓
5 requireddate	timestamp		✓
6 shippeddate	timestamp		✓
7 shipvia	int4		✓
8 freight	money	40	✓
9 shipname	path	200	✓
10 shipaddress	point		✓

A dropdown menu for 'Data Type' is open over the 'money' entry, showing options like 'path', 'point', 'polygon', 'serial', 'smallint', 'text', 'time', and 'timetz'. Below the table, there are tabs for 'Constraints', 'Storage', 'Comments', 'Permissions', 'Preview SQL', and 'Messages'.

pgAdmin

- Sehr empfehlenswert: Arbeiten mit **pgAdmin**
- Beispiel: um Verknüpfung zu einer Datenbank herzustellen - Doppelklick auf den Server in pgAdmin



PostGIS für PostgreSQL

- **Installieren** der PostGIS Erweiterung:

```
CREATE EXTENSION postgis;
```



Programm zum Import der OSM Daten in PostgreSQL - osm2pgsql

- Läuft unter Linux deutlich besser
- Man kann entweder das Planet file oder Ausschnitte (im Beispiel für Berlin) herunterladen und direkt in die PostgreSQL Datenbank importieren.
- so könnte bspw. ein Import aussehen:

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

Verbindung zwischen R und Postgresql

Github Verzeichnis zum Paket

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

```
library("RPostgreSQL")
```

Nutzung von Quantum GIS

QGIS - Ein freies Open-Source-Geographisches-Informationssystem

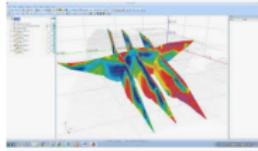
- Erstellen, bearbeiten, anzeigen, analysieren räumlicher Information unter Windows, Mac, Linux, BSD (bald auch Android)
- Mit QGIS kann bspw auch der WMS/WCS oder WFS-Dienst des IÖR-Monitors genutzt werden.



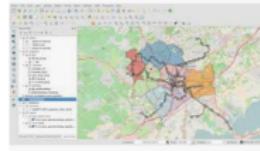
Boundless Releases QGIS 2.14.1 Open Source
boundlessgeo.com



Datei QGIS Logo.png – Wikipedia
de.wikipedia.org



Using GRASS GIS/QGIS to georeference TIFFs in 3...
gis.stackexchange.com



Prepare a QGIS project for Web
docs.3liz.com

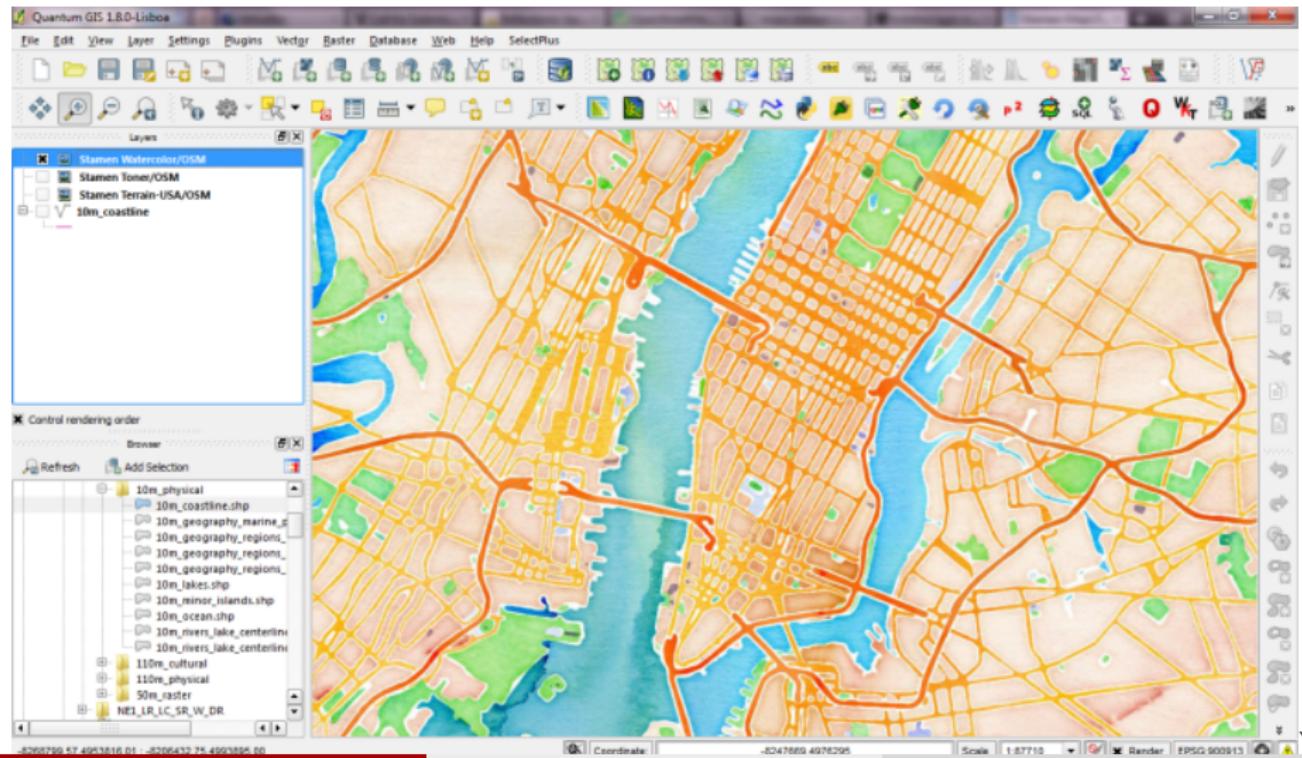


QGIS Online – rollApp
rollapp.com

)

Nutze bspw. QGIS um Shapefiles zu extrahieren

- Plugin OpenLayers



Links

- **Wiki zum Download** von Openstreetmap Daten
- Auf dem **Openstreetmap Blog** gibt es wöchentlich neue Nachrichten zu OSM.
- Liste möglicher Datenquellen für räumliche Analysen (weltweit und in **Deutschland**)
- Daten zu Administrative Grenzen kann man bei *Second Administrative Level Boundaries (SALB)* bekommen

Use Case Geomedizin

OpenStreetMap



DE:Key:healthcare

Wert	Beschreibung
healthcare=alternative	Jemand, der alternative Medizin anbietet, aber kein zugelassener Arzt mit entsprechendem Fachstudium ist. Bei Angeboten, die in keine der hier aufgeführten Kategorien passen, sollte besser die generische Kennzeichnung healthcare=yes benutzt werden.
healthcare=audiologist	Ein Hörgeräteakustiker . Für Geschäfte, die Hörhilfen verkaufen, sollte besser shop=hearing_aids verwendet werden.
healthcare=birthing_center	Ein von Hebammen betriebenes, außerklinisches Geburtshaus .
healthcare=blood_bank	Eine Blutbank zur Lagerung von Blutkonserven.
healthcare=blood_donation	Eine Einrichtung zur Entnahme von Blut- und Plasmaspenden. Auch Stammzellproben können an manchen Stellen entnommen werden.
healthcare=centre	Mehrere medizinische Einrichtungen an einem Ort, z.B. Gemeinschaftspraxen mit unterschiedlichen Schwerpunkten der praktizierenden Mediziner.
healthcare=clinic	Eine Tagesklinik zur ambulanten oder kurzzeitig stationären Behandlung. Sollte zur Zeit zusammen mit amenity=clinic verwendet werden!
healthcare=dentist	Eine Zahnarztpraxis . Sollte zur Zeit zusammen mit amenity=dentist verwendet werden!
healthcare=doctor	Eine Arztpraxis . Allgemein auch als Doktor bezeichnet. Sollte zur Zeit zusammen mit amenity=doctors verwendet werden!.
healthcare=hospital	Ein Krankenhaus . Sollte zur Zeit zusammen mit amenity=hospital verwendet werden!
healthcare=laboratory	Ein medizinisches Labor welches medizinische Analysen durchführt

Orte für die Blutspende

healthcare	blood_donation		Einrichtung zur Blutspende	
------------	----------------	---	----------------------------	---

Die Nutzung von Overpass

```
node
[healthcare=blood_donation]
({{bbox}});
out;
```

The screenshot shows the Overpass Turbo interface. On the left, there is a code editor window containing an Overpass query:

```
/*
This is an example Overpass query.
Try it out by pressing the Run button above!
You can find more examples with the Load
tool.
*/
node
[healthcare=blood_donation]
({{bbox}});
out;
```

The main area displays a map of the region around Frankfurt, Mannheim, and Stuttgart, with several locations marked with blue circles. The map includes place names like Frankfurt am Main, Würzburg, Mannheim, Heidelberg, Kaiserslautern, Stuttgart, and Baden-Baden. The Overpass Turbo interface has a top navigation bar with links for Ausführen, Teilen, Export, Wizard, Speichern, Laden, Einstellungen, Hilfe, and a search bar. Below the toolbar is a tab labeled "Karte" which is currently selected.

Alternativmediziner

node

```
[healthcare=alternative]
({{bbox}});
out;
```

The screenshot shows the Overpass Turbo web interface. The top navigation bar includes links for Ausführen, Teilen, Export, Wizard, Speichern, Laden, and Einstellungen. The main area contains a code editor with an example query and a map view. The code editor displays:

```

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

```

The map view shows a region around Pforzheim with several yellow circular markers indicating the locations of alternative medicine providers. To the right of the map is an "Exportieren" sidebar with options for saving the results.

Exportieren

- Daten
 - [Speichere/Kopiere als GeoJSON](#)
 - [Speichere/Kopiere als GPX](#)
 - [Speichere/Kopiere als KML](#)
 - [Speichere/Kopiere als OSM Rohdaten](#)
 - [Rohdaten direkt von Overpass API](#)
 - [In einen OSM-Editor laden: JOSM, Leveldo](#)
 - [GeoJSON als gist speichern](#)

Erste Karten

Das Paket maptools

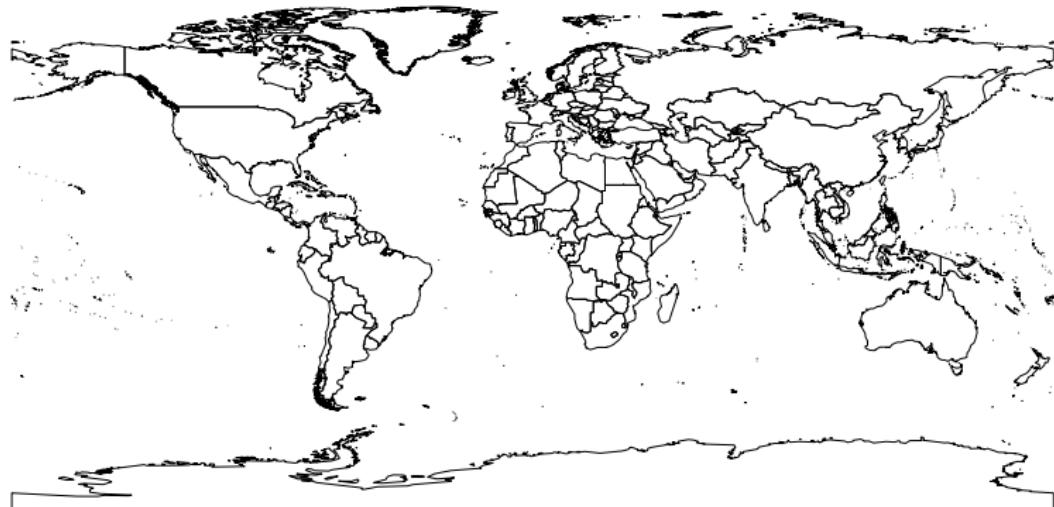
- Datensatz wrld_simpl aus dem Paket maptools
- Polygone für fast alle Staaten der Erde

```
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
```

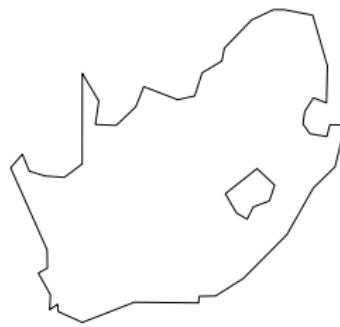
```
table(ind_SA)
```

```
## ind_SA  
## FALSE TRUE  
## 245 1
```

Eine Karte für Süd Afrika

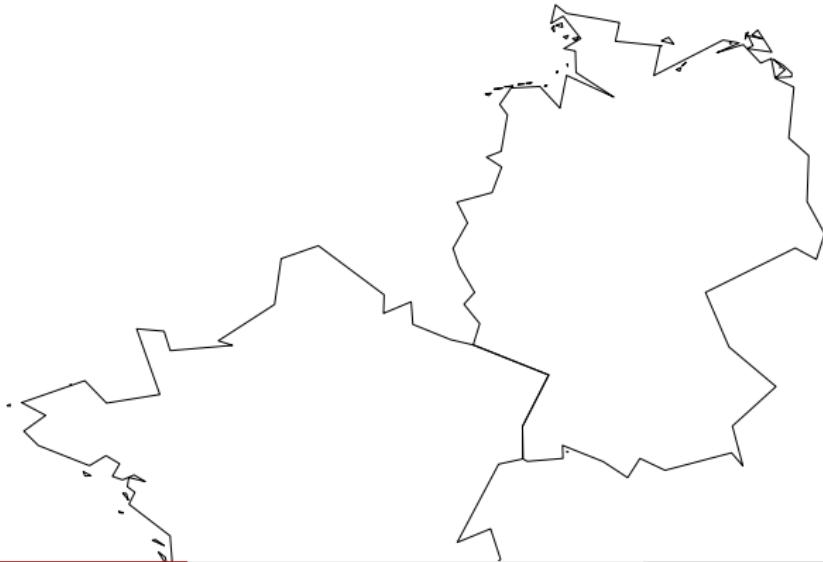
- Ein Land zeichnen

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



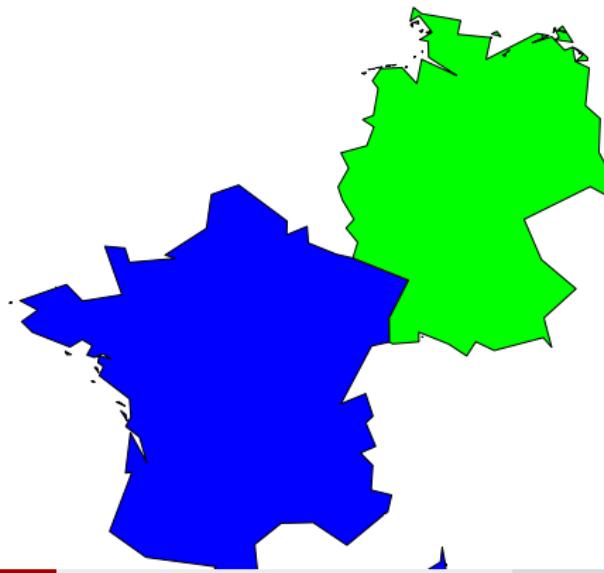
Mehr als ein Land zeichnen

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



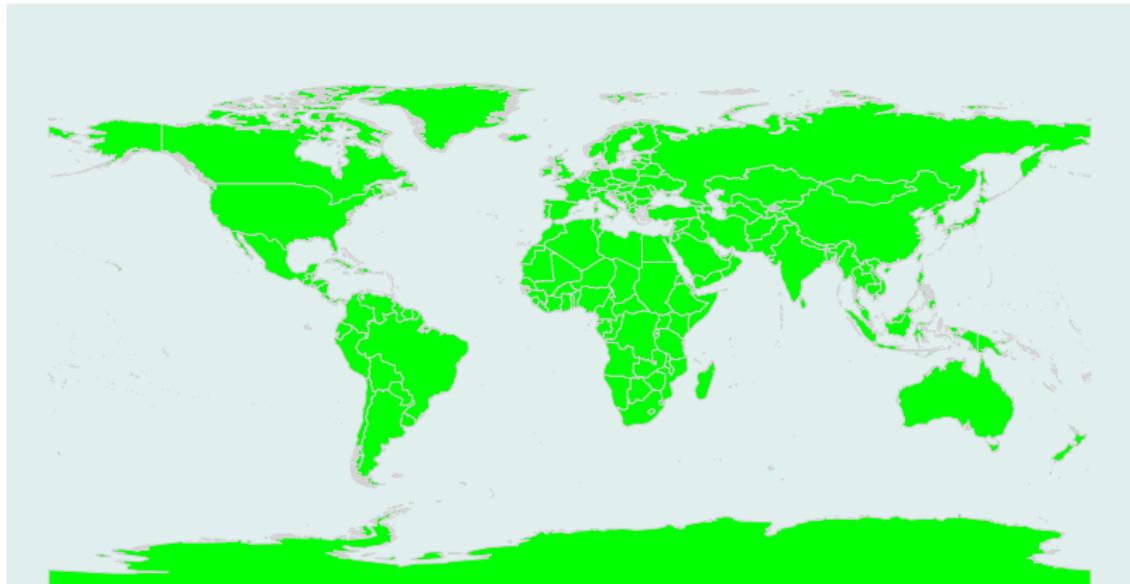
Mehr Farbe

```
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

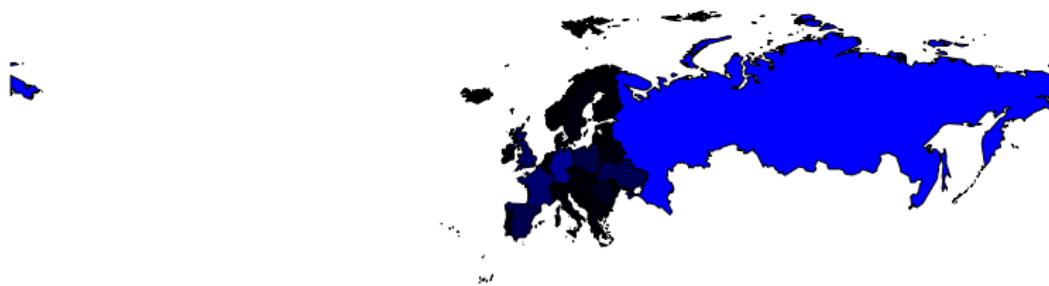
Colors in R

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



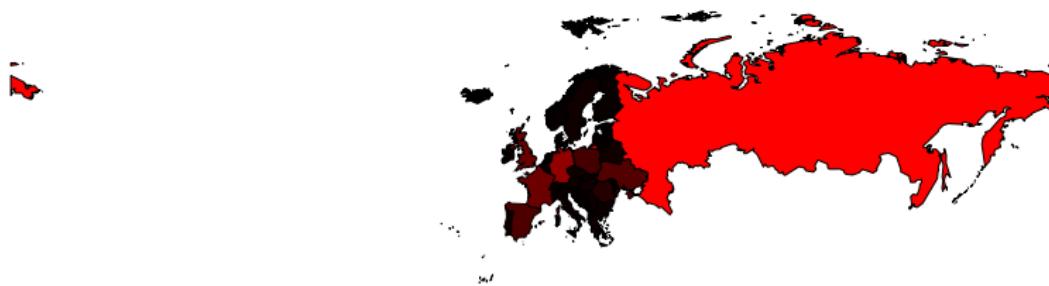
Europa - Farbschattierung blau

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



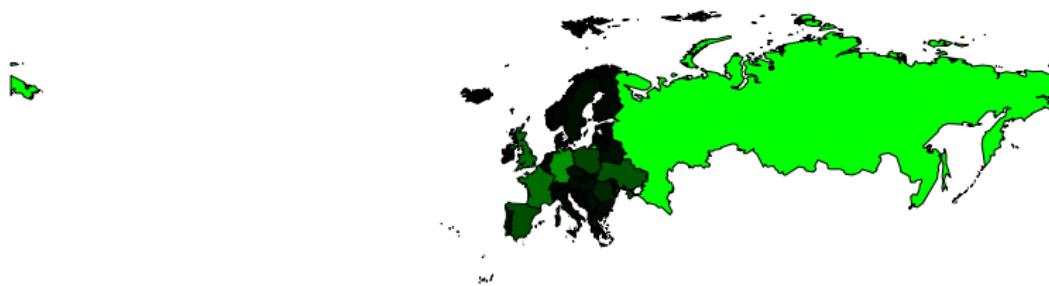
Europa - Farbschattierung rot

```
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))
```



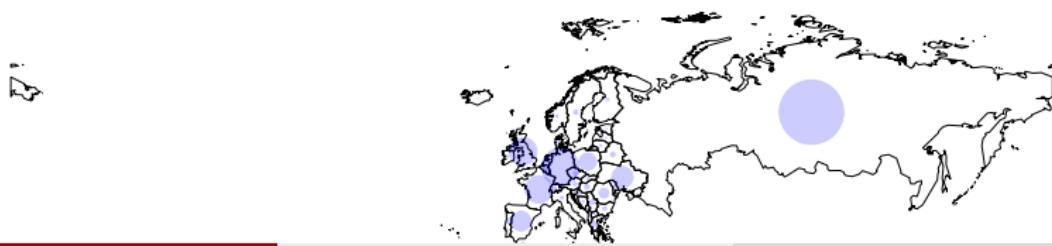
Europa - Punkte hinzufügen

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



Europa - Blasen hinzufügen

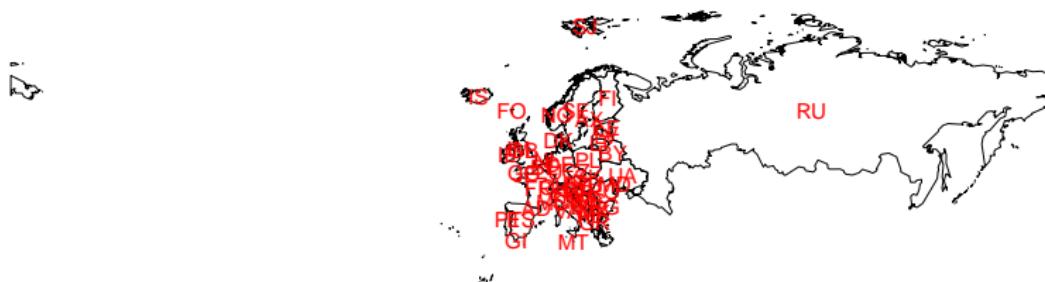
```
pop <- Europe$POP2005  
pop <- pop/max(pop)*10  
plot(Europe)  
points(Europe$LON,Europe$LAT,cex=pop,col=rgb(0,0,1,.2),  
pch=20)
```



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

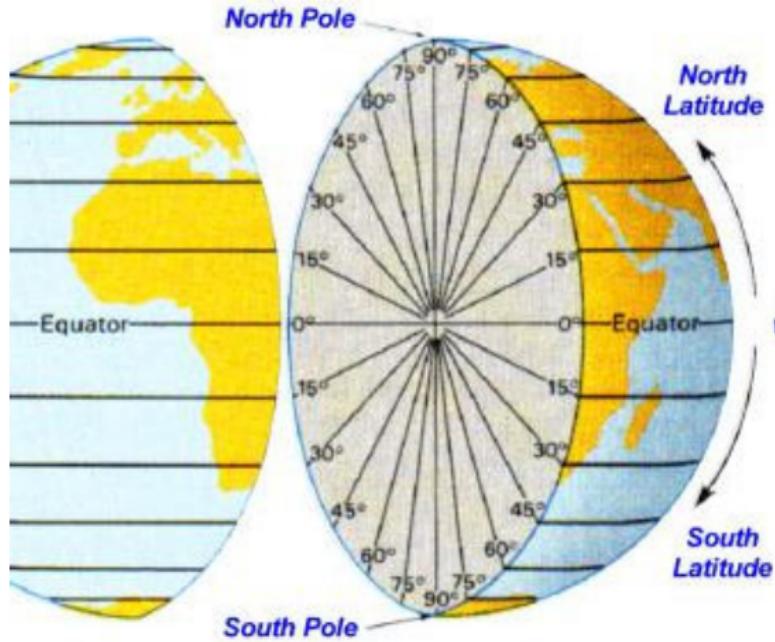
Geokodierung

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

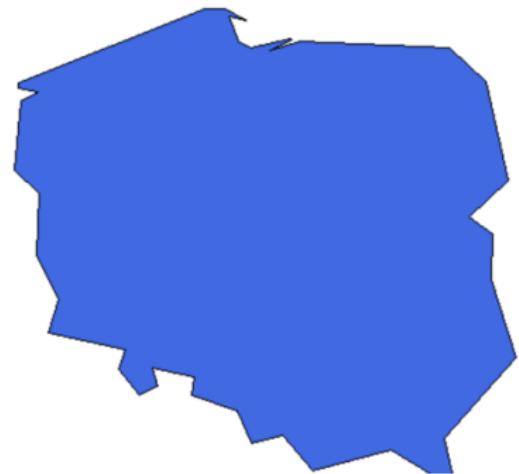


Shapefiles bekommen, bearbeiten und visualisieren

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

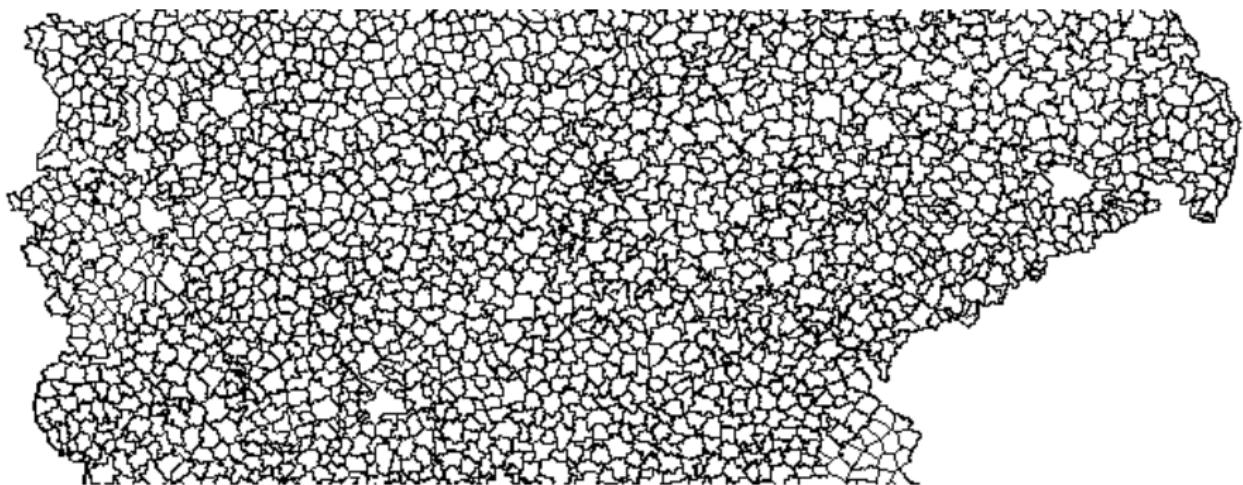
Quelle Ortsnetzbereiche: Bundesnetzagentur

- **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
```

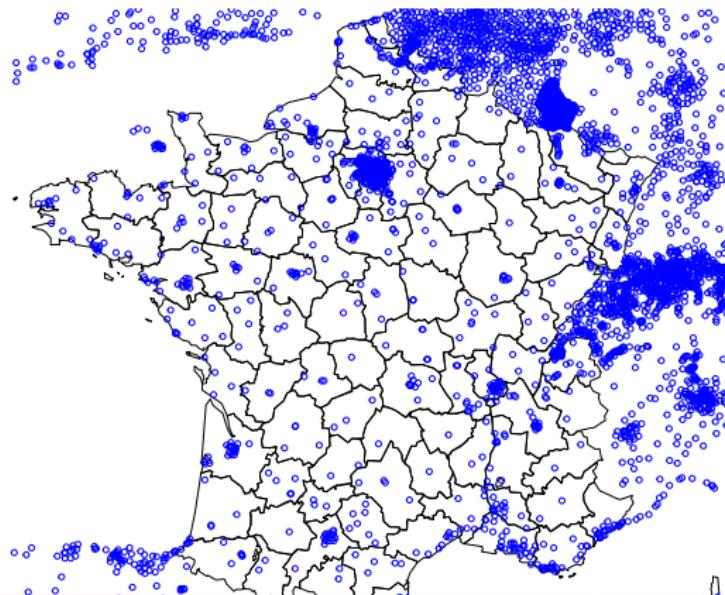
Die Daten plotten

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



Die Grenze von Mannheim

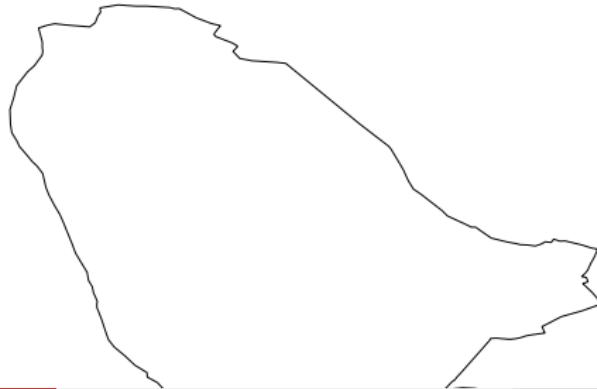
```
ma_map <- plz[plz$PLZORT99=="Mannheim",]  
plot(ma_map)
```



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
```

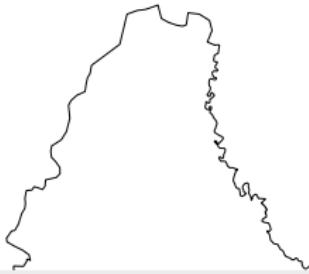
Aufgabe: PLZ Bereiche herunterladen

- 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

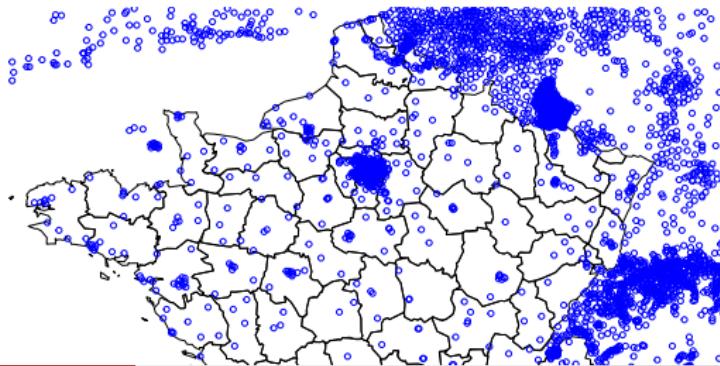
```
fds <- substr(krs@data$AGS, 1, 2)
plot(krs[fds=="05",])
```



Das Paket maps - Mehr Information

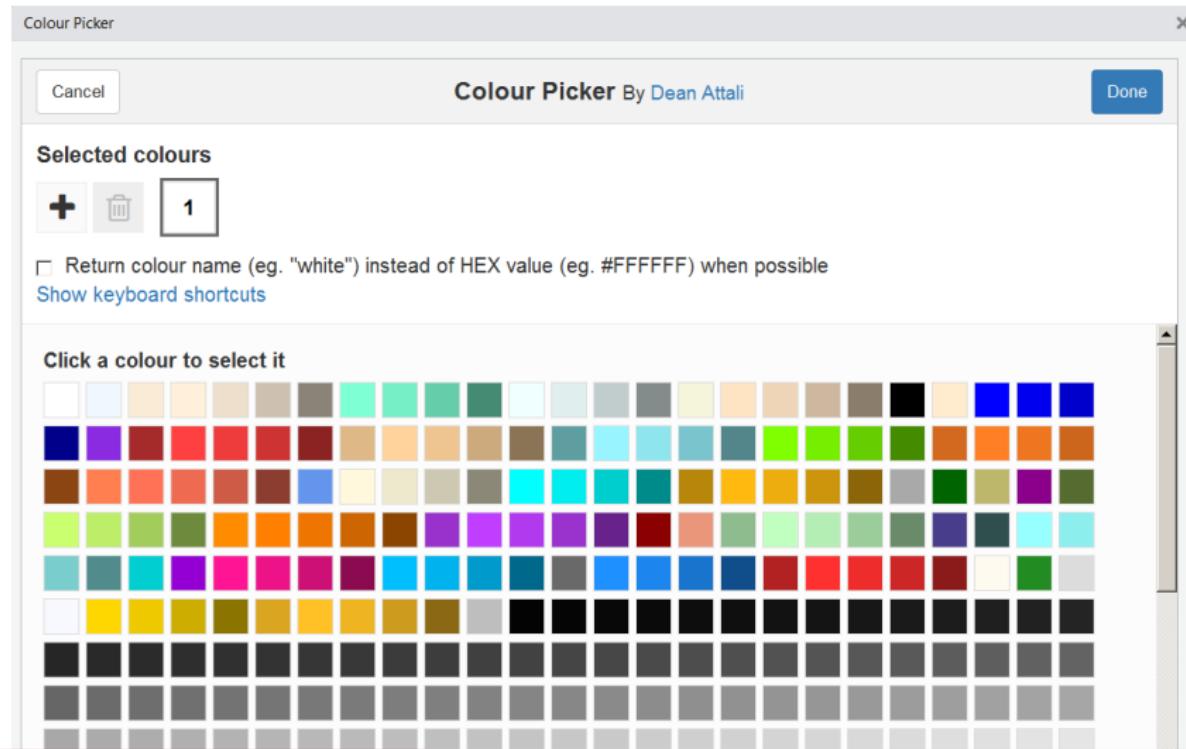
- Nur für manche Staaten bekommt man mit dem Paket `maps` Umkreise für Einheiten unterhalb der Staatsgrenze (bspw. Frankreich, USA).

```
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](#) | [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-Vermerkes¹ ist es Ihnen gestattet, die Inhalte uneingeschränkt zu verwenden.

[Dateiformat](#)[Bildformat](#)[Inhalt](#)[Geometrie²](#)

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									
+ 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	CENSUS_UNITS_PT_2011.zip	
	Alle	Europa	Point	Shapefile	2011	ETRS89	08/10/2015	CENSUS_UNITS_PT_2011_SH.zip	
Zensus-Einheiten 2011	1:1 Mio	Europe	Polygon	Personal GDB	2011	ETRS89	08/10/2015	CENSUS_UNITS_RG_01M_2011.zip	

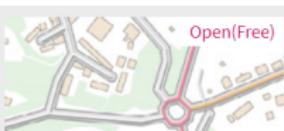
Weitere Quellen für Shapefiles

- Open linked data - Ordnance Survey (GB)



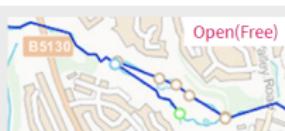
OS Names API ›

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

```
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	http://www.gruibirg.de
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")
```

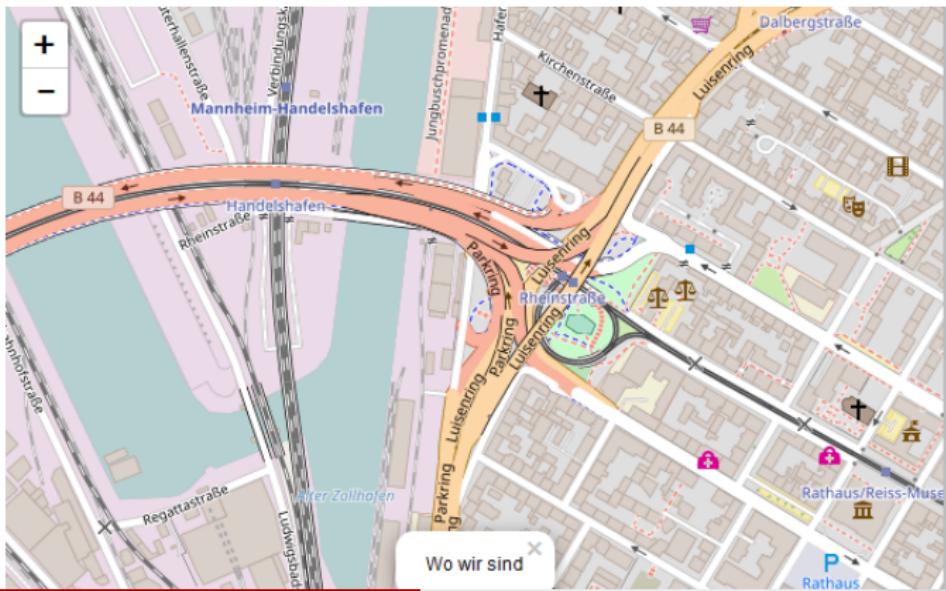
Eine erste interaktive Karte

```
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
```

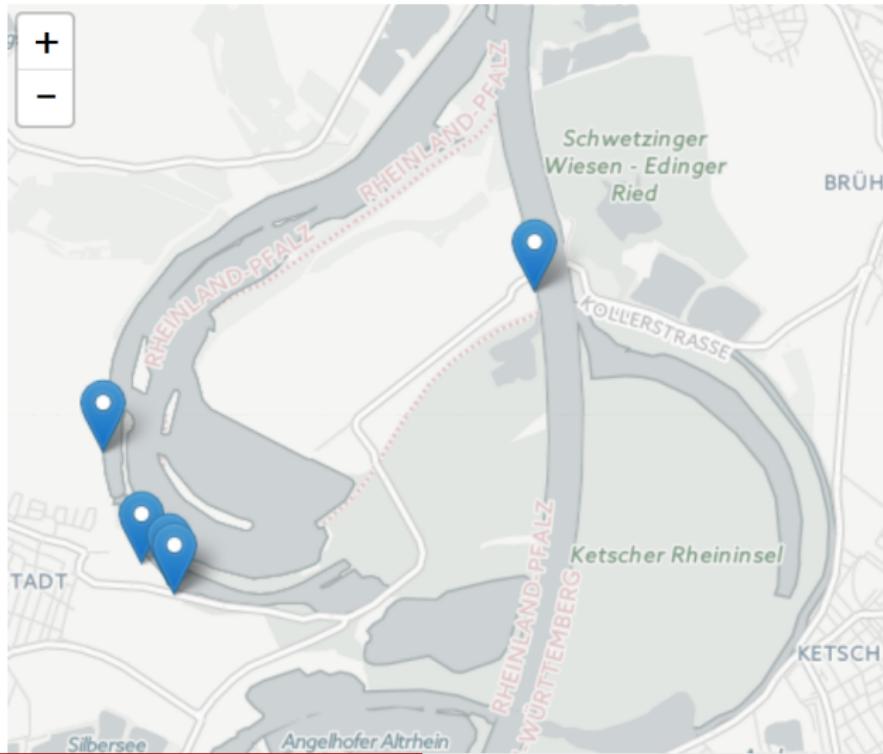
Stamen als Hintergrundkarte

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



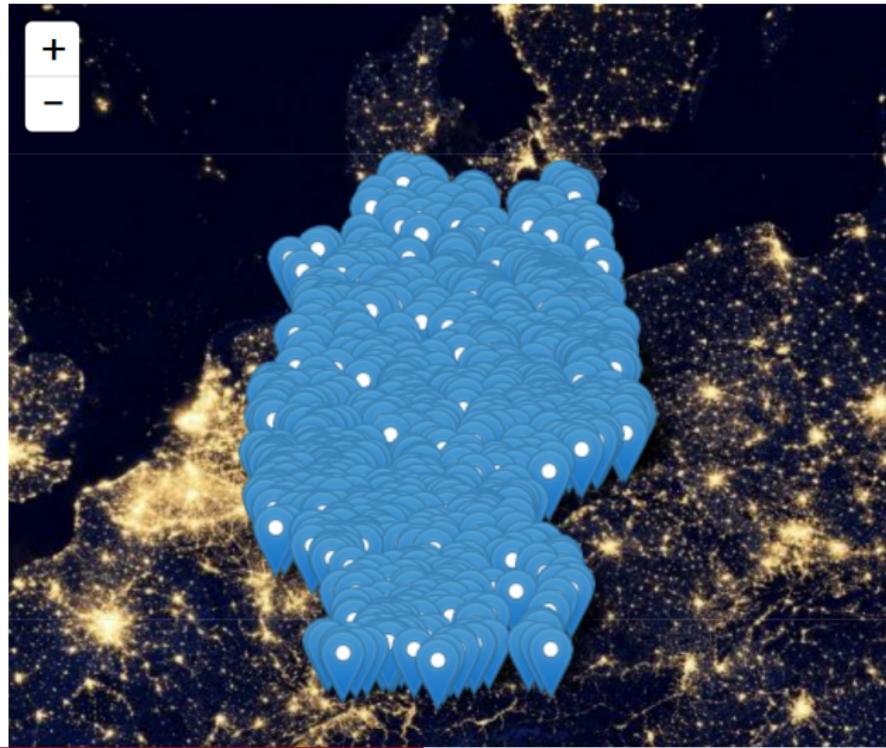
CartoDB als Hintergrund

```
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
```

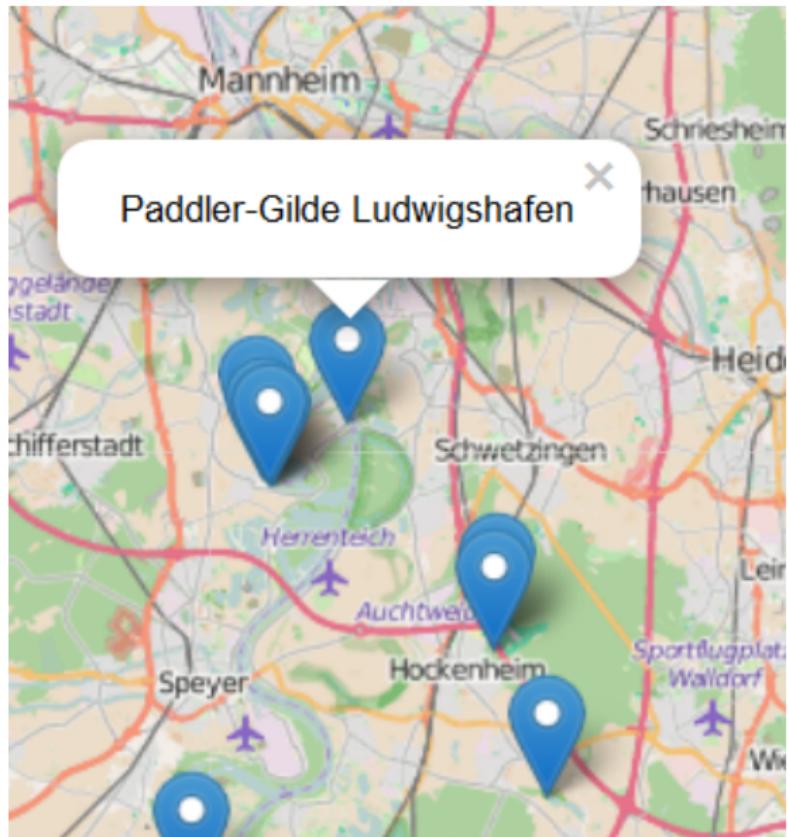
Das Ergebnis ist hier:

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

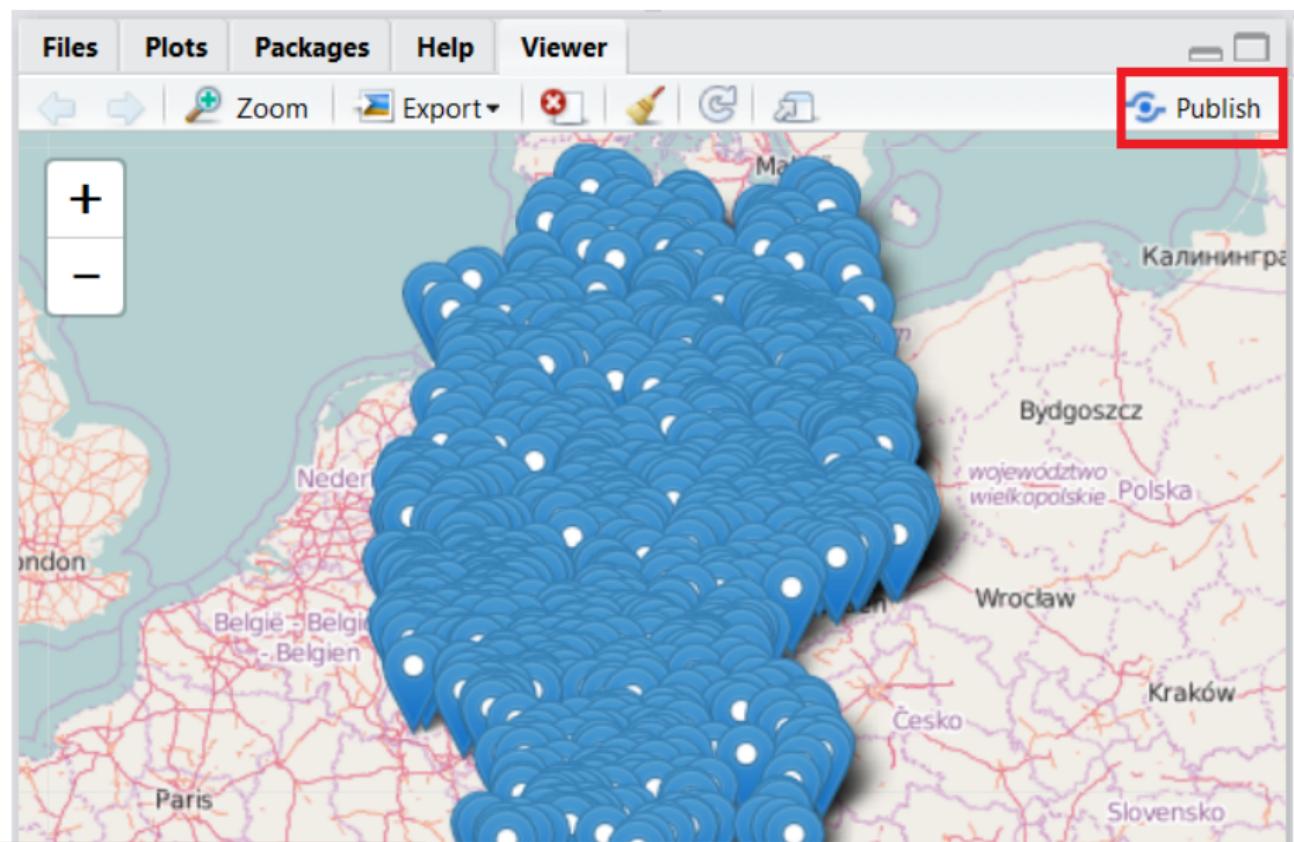
Die resultierende Karte



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

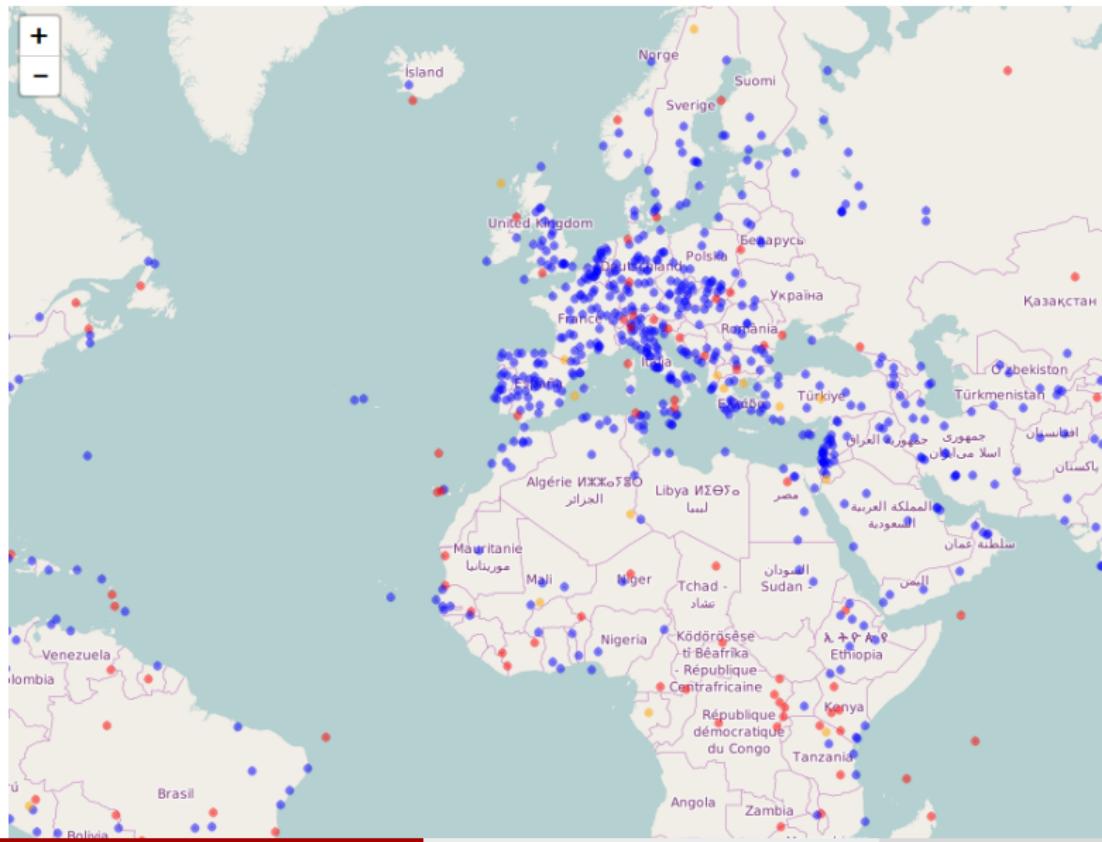
```
whcSites$color <- "red"  
whcSites$color[whcSites$category=="Cultural"] <- "blue"  
whcSites$color[whcSites$category=="Mixed"] <- "orange"
```

Eine Karte mit Farbe erzeugen

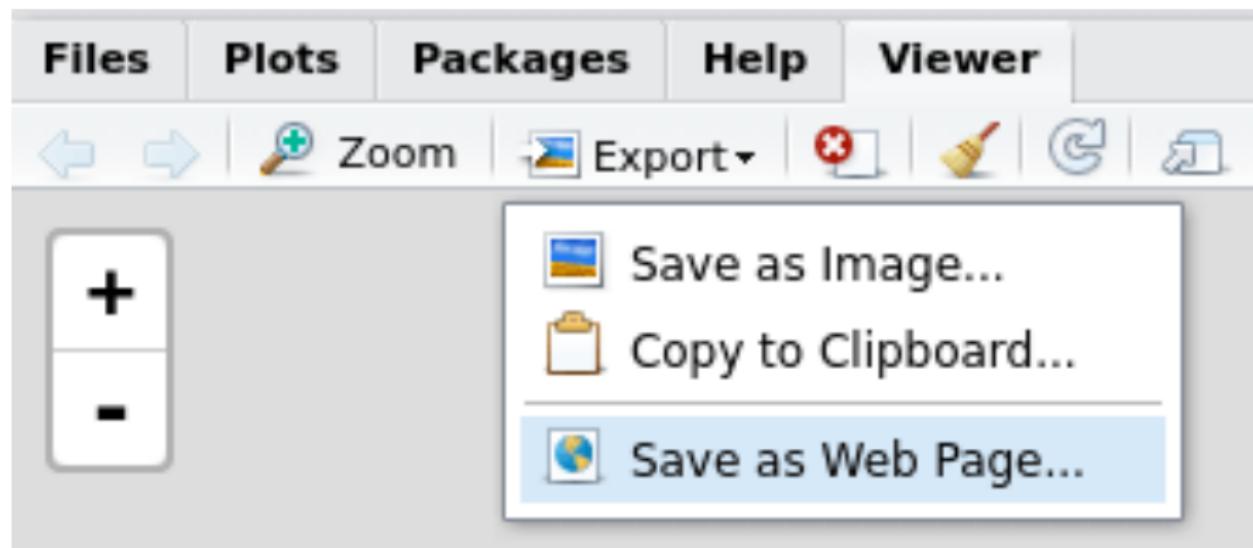
```
m1 <- leaflet() %>%
  addTiles() %>%
  addCircles(lng=whcSites$lon,
             lat=whcSites$lat,
             popup=whcSites$name_en,
             color=whcSites$color)
```

```
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**