

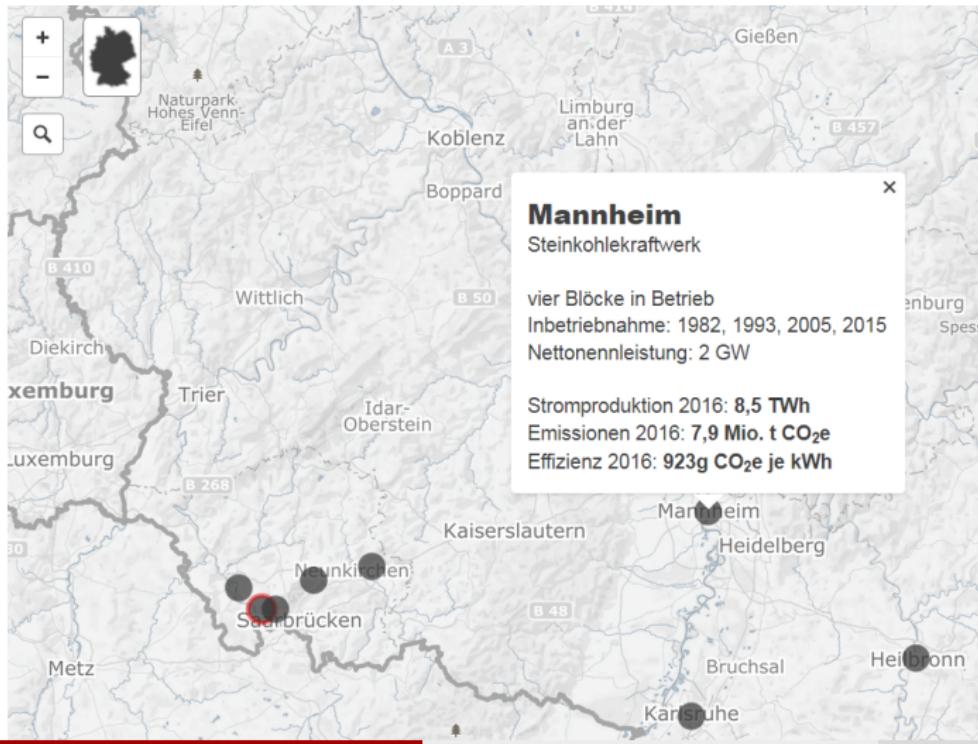
Thematische Karten erstellen mit R

Jan-Philipp Kolb

23 November 2017

Motivation - Deutschlands größte Klimasünder

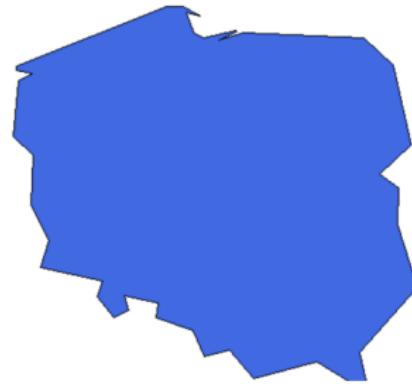
- Spiegel Artikel am 16.11.2017 aus Anlass der Jamaika Gespräche



Gliederung

- Quellen für Polygonzüge (Staaten, Gemeinden, PLZ Bereiche etc.)
- Quellen für inhaltliche Daten
- Pakete zur Erstellung thematischer Karten (bspw. Pakete `maptools`, `sp`, `tmap`)
- Verknüpfung von Daten
- Beispiele für die Darstellung in Karten

Quellen für Polygonzüge



Hello World

```
library(maps)  
map()
```



Das Paket `maps` - etwas detaillierter

Grenzen sind recht grob:

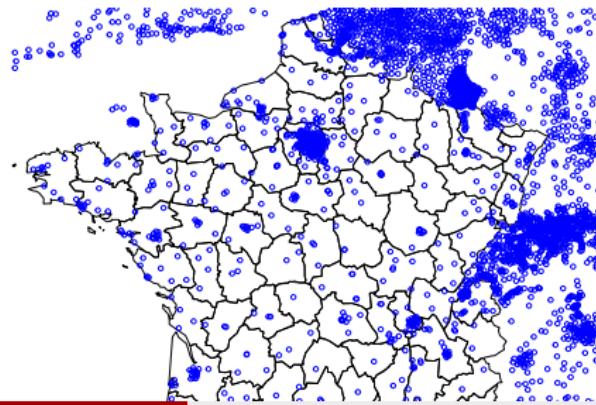
```
map("world", "Germany")
```



Das Paket maps - Mehr Information

- Nur für manche Staaten bekommt man Umkreise für Einheiten unterhalb der Staatsgrenze (bspw. Frankreich, USA).

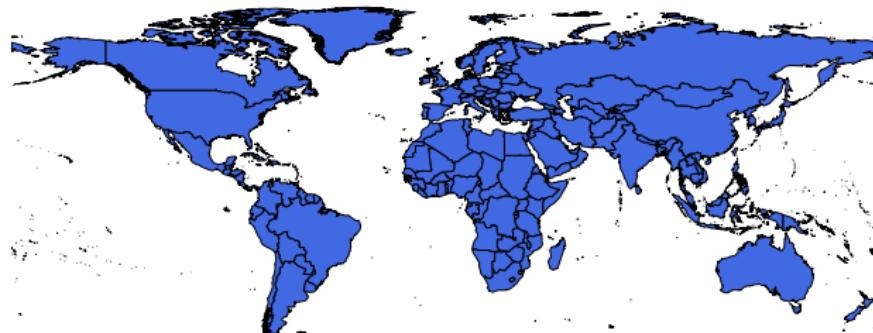
```
data(world.cities)
map("france")
map.cities(world.cities,col="blue")
```



Das Paket `maptools`

- Das Paket `maptools` hat intuitivere Bedienung, zudem können Shapefiles verarbeitet werden.

```
library(maptools)
data(wrld_simpl)
plot(wrld_simpl,col="royalblue")
```



Was sind shapefiles (.shp)?

- Das Dateiformat Shapefile ist ein ursprünglich für die Software ArcView der Firma ESRI entwickeltes Format für Geodaten. (Quelle: Wikipedia)

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

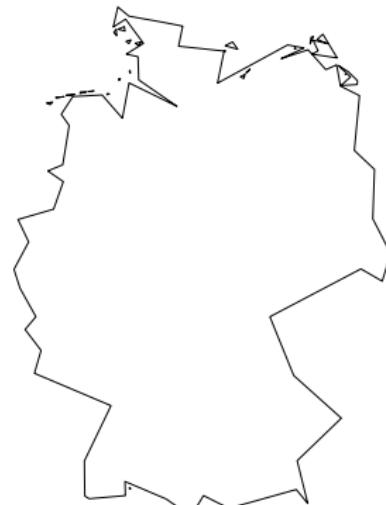
	FIPS	ISO2	ISO3	UN	NAME	AREA	POP2010
ATG	AC	AG	ATG	28	Antigua and Barbuda	44	830000
DZA	AG	DZ	DZA	12	Algeria	238174	328541000
AZE	AJ	AZ	AZE	31	Azerbaijan	8260	83520000
ALB	AL	AL	ALB	8	Albania	2740	31537000
ARM	AM	AM	ARM	51	Armenia	2820	30176000
AGO	AO	AO	AGO	24	Angola	124670	160952000

```
length(wrld_simpl)
```

Einzelne Elemente des Datensatzes plotten

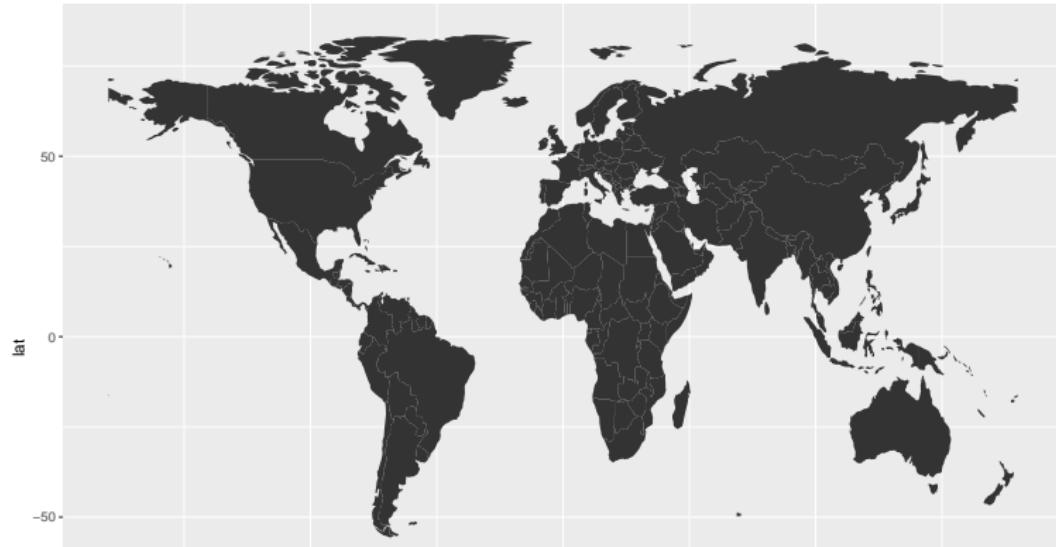
```
ind <- which(wrld_simpl$ISO3=="DEU")
```

```
plot(wrld_simpl[ind,])
```



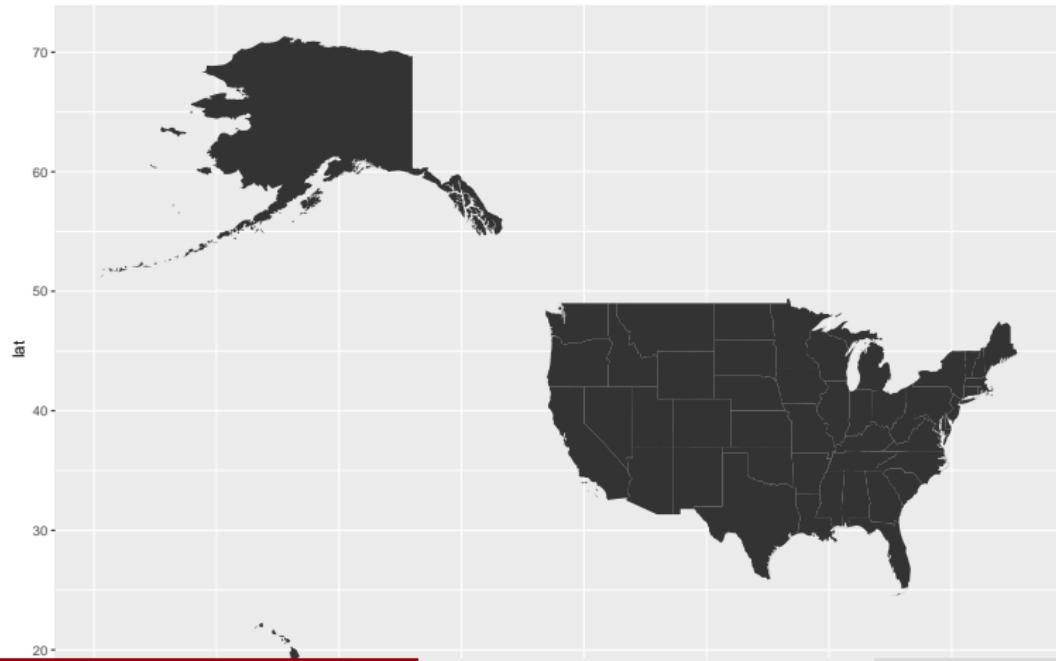
Das R-Paket choroplethrMaps

```
library(ggplot2);library(choroplethrMaps)
data(country.map)
ggplot(country.map, aes(long, lat, group=group))+
  geom_polygon()
```



Eine Karte für die USA

```
data(state.map)
ggplot(state.map,aes(long,lat,group=group))+geom_polygon()
```



Andere Quellen für Shapefiles - Das Paket raster

- 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)
```



Daten für das Luxembourg Beispiel

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

OBJECTID	ID_0	ISO	NAME_0	ID_1	NAME_1	HASC_1
1	131	LUX	Luxembourg	1	Diekirch	LU.DI
2	131	LUX	Luxembourg	2	Grevenmacher	LU.GR
3	131	LUX	Luxembourg	3	Luxembourg	LU.LU

Shapefiles bei Eurostat

 eurostat
Ihr Schlüssel zur europäischen Statistik

Anmelden | Registrieren 

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

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

Neugkeiten **Daten** **Veröffentlichungen** **Über Eurostat** **Hilfe**

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

ZENUS

Zensus

Bitte beachten Sie, dass die fraglichen Bestimmungen für die jeweiligen Datensätze einzuhalten. Das Herunterladen und die Verwendung dieser Daten ist abhängig von deren Abnahme.

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
	1:1 Mio	Europe	Polygon	Shapefile	2011	ETRS89	08/10/2015	CENSUS_UNITS_2011_RG_SH.zip

BKG - Quelle für Kreise in Deutschland

- Umrisse von 402 Kreisen in Deutschland
- Quelle: Bundesamt für Kartographie und Geodäsie (BKG)
- Karten gibt es auch für Bundesländer und Gemeinden

```
library(maptools)
krs <- readShapePoly("vg250_ebenen/vg250_krs.shp")
plot(krs)
```

```
head(krs@data$RS)
```

```
## [1] 03401 03458 09473 05962 10046 05916
## 402 Levels: 01001 01002 01003 01004 01051 01053 01054 01055
```

Die Kreise für Baden-Württemberg

- Systematik hinter dem Amtlichen Gemeindeschlüssel (AGS) bzw. Regionalschlüssel (RS) bei Wikipedia

```
BLA <- substr(krs@data$RS, 1, 2)
plot(krs[BLA=="08",])
```



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	Geometrie ²
PDF	DIN A1	Karte der Wahlkreise und der Landkreise und kreisfreien Städte mit Beschriftung und Legende (1,41 MB)	generalisiert ⁴

Ortsnetzbereiche

Quelle: Bundesnetzagentur

```
onb <- readShapePoly("onb_grenzen.shp")
```

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

	VORWAHL	NAME	KENNUNG
0	04651	Sylt	NA
1	04668	Klanxbüll	NA
2	04664	Neukirchen b Niebüll	NA
3	04663	Süderlügum	NA
4	04666	Ladelund	NA
5	04631	Glücksburg Ostsee	NA

Karte der Vorwahlbereiche



Einen größeren Vorwahlbereich ausschneiden

```
vwb <- as.character(onb@data$VORWAHL)
vwb1 <- substr(vwb, 1,2)
vwb7 <- onb[vwb1=="07",]
plot(vwb7)
```



Das Paket rgdal

- Postleitzahlensbereiche - <http://arnulf.us/PLZ>

```
library(rgdal)
```

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

```
library(rgdal)
PLZ <- readOGR ("post_pl.shp","post_pl")
```

PLZ-Bereiche in Stuttgart

```
SG <- PLZ[PLZ@data$PLZORT99=="Stuttgart",]  
plot(SG,col="chocolate1")
```



PLZ-Bereiche in Berlin

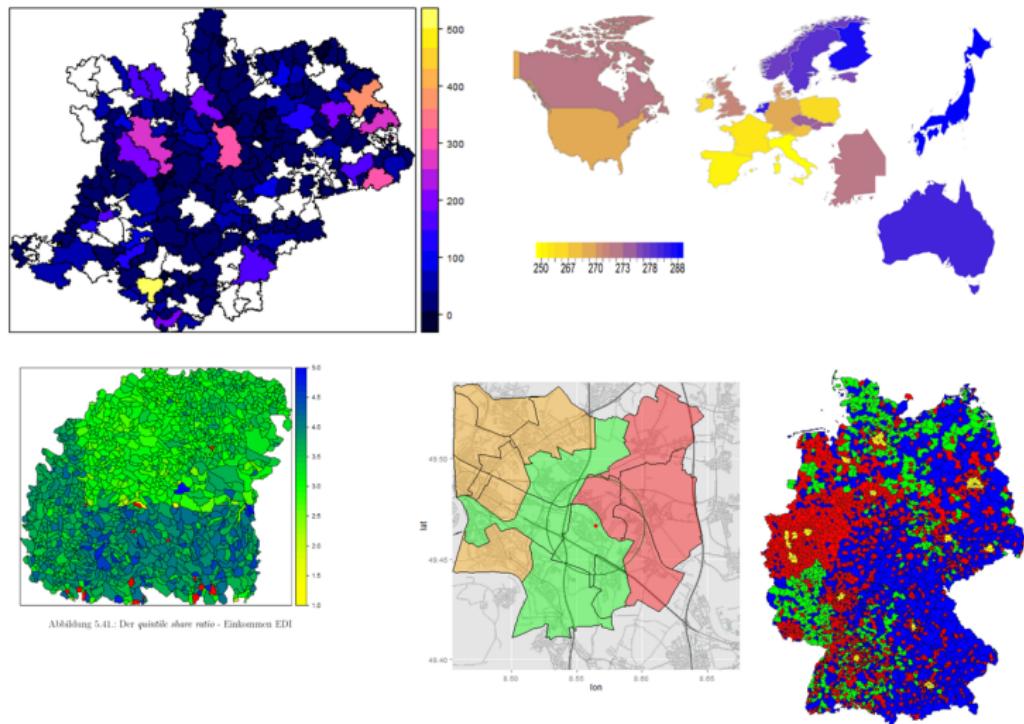
```
BE <- PLZ[PLZ@data$PLZORT99%in%c("Berlin-West",  
                                     "Berlin (östl. Stadtbezirke)"),]  
plot(BE,col="chocolate2",border="lightgray")
```



Zwischenfazit - Quellen für Polygonzüge

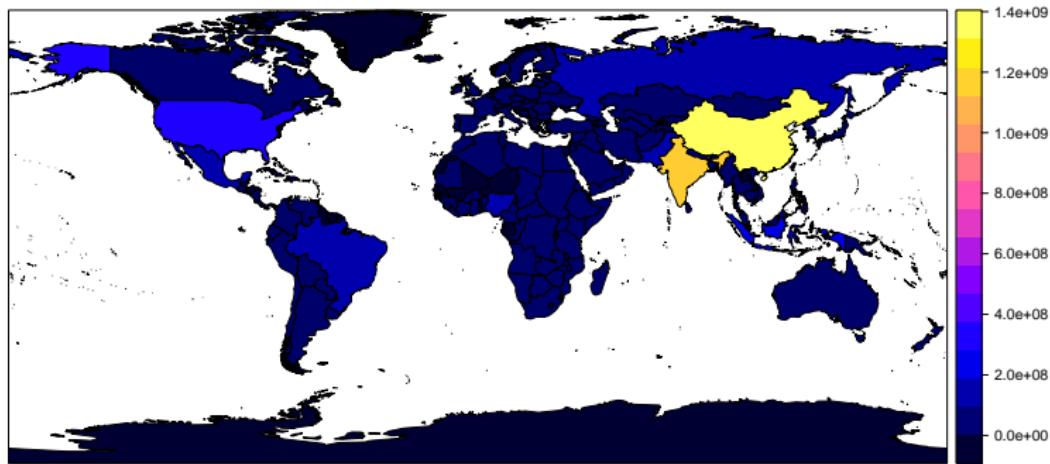
- In einigen R-Paketen sind Polygonzüge hinterlegt
- Umgang mit den Daten unterscheidet sich
- Am sinnvollsten ist das arbeiten mit Shapefiles (.shp)
- Neben den R-Paketen gibt es zahlreiche weitere Quellen für Polygonzüge.

Thematische Karten mit R erstellen



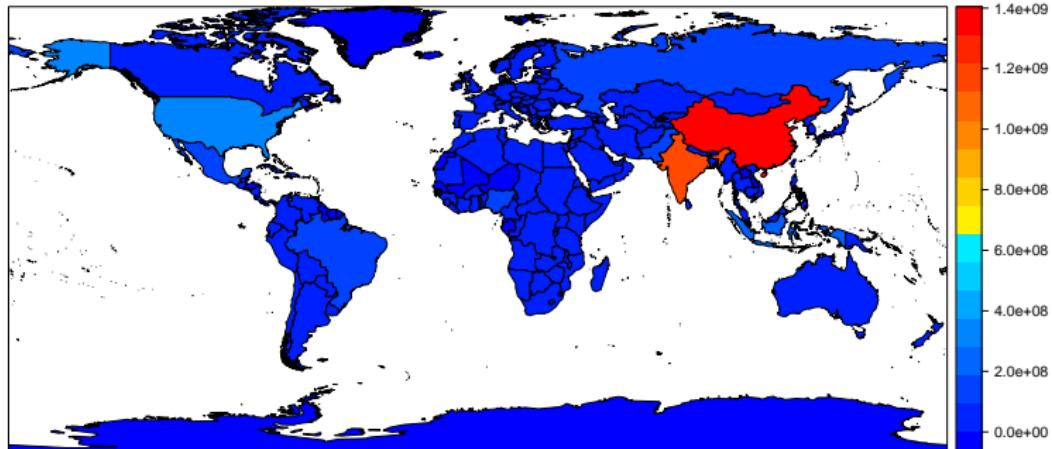
Thematische Karten erzeugen - das Paket sp

```
library(sp)  
spplot(wrld_simpl, "POP2005")
```



Andere Einfärbungen wählen - Das Paket colorRamps

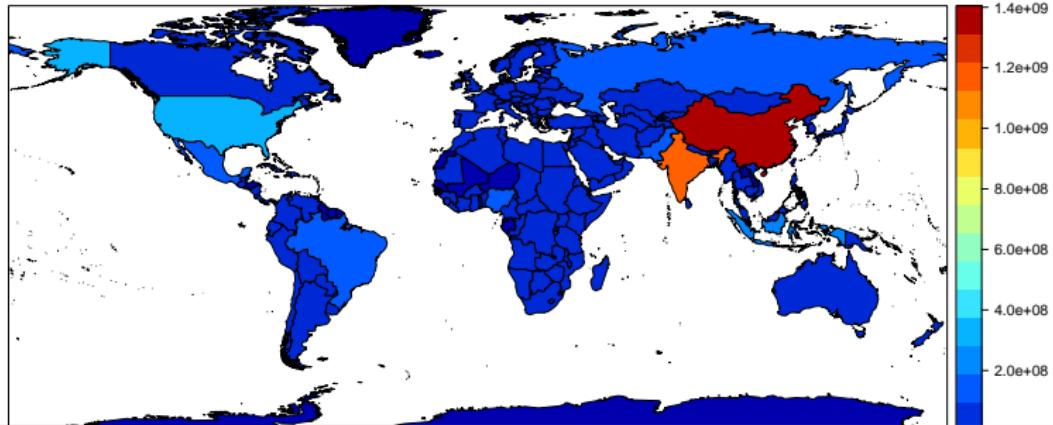
```
library(colorRamps)
spplot(wrld_simpl, "POP2005", col.regions=blue2red(100))
```



Es gibt auch noch mehr Farbverläufe im Paket colorRamps

- blue2green, blue2yellow

```
spplot(wrld_simpl, "POP2005", col.regions=matlab.like(100))
```



Thematische Karten mit dem Paket choroplethr

```
library(choroplethr)
```

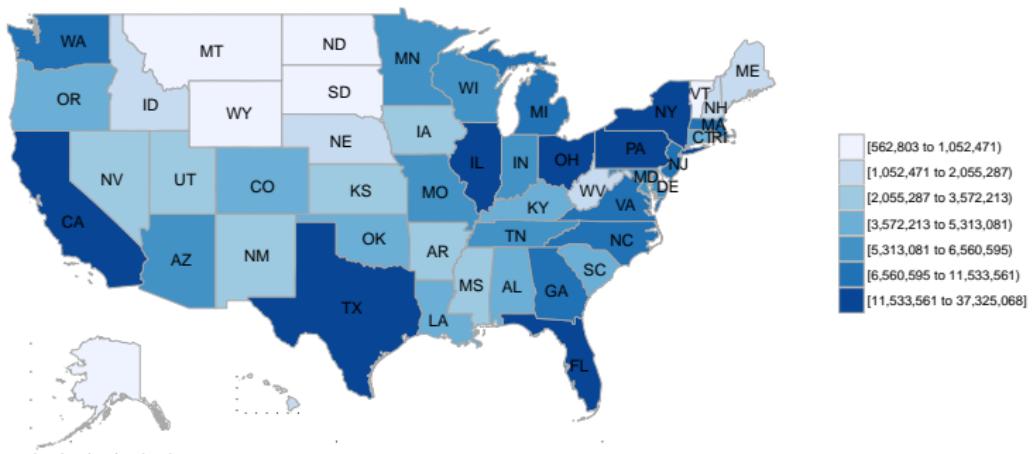
```
data(df_pop_state)
```

```
head(df_pop_state)
```

region	value
alabama	4777326
alaska	711139
arizona	6410979
arkansas	2916372
california	37325068
colorado	5042853

Eine thematische Karte mit choroplethr erstellen

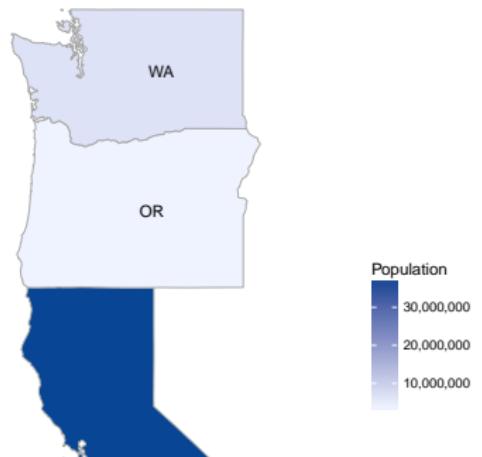
```
state_choropleth(df_pop_state)
```



Nur drei Staaten darstellen

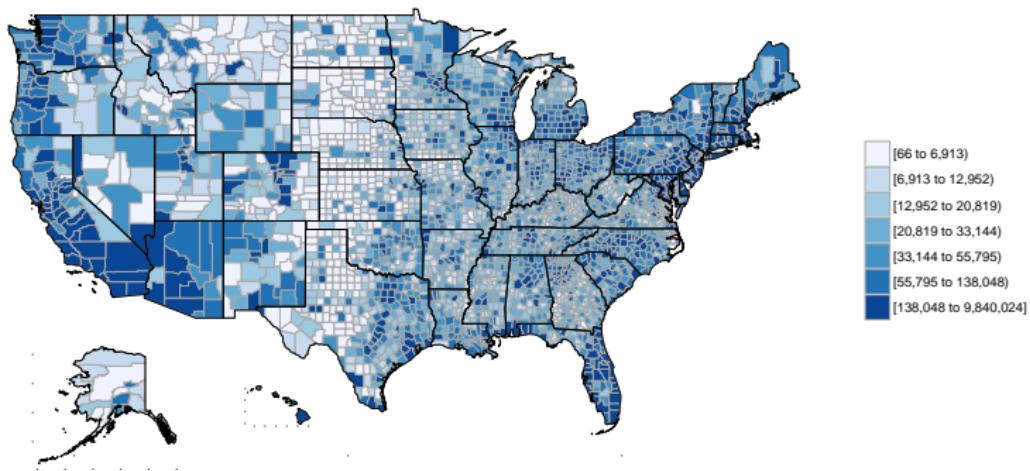
```
state_choropleth(df_pop_state,
                  title      = "2012 Population Estimates",
                  legend     = "Population", num_colors = 1,
                  zoom       = c("california", "washington",
                               "oregon"))
```

2012 Population Estimates



Eine Karte der US Counties

```
data(df_pop_county)  
county_choropleth(df_pop_county)
```



Choroplethen Länder

```
data(df_pop_country)
country_choropleth(df_pop_country,
                    title      = "2012 Population Estimates",
                    legend     = "Population", num_colors = 1,
                    zoom       = c("austria", "germany",
                                "poland", "switzerland"))
```

2012 Population Estimates



Weltbank Daten

- AG.AGR.TRAC.NO - Agricultural machinery, tractors

```
library(WDI)
WDI_dat <- WDI(country="all",
                 indicator=c("AG.AGR.TRAC.NO",
                            "TM.TAX.TCOM.BC.ZS"),
                 start=1990, end=2000)
```

- Es gibt auch eine Funktion WDIsearch mit der man nach Indikatoren suchen kann

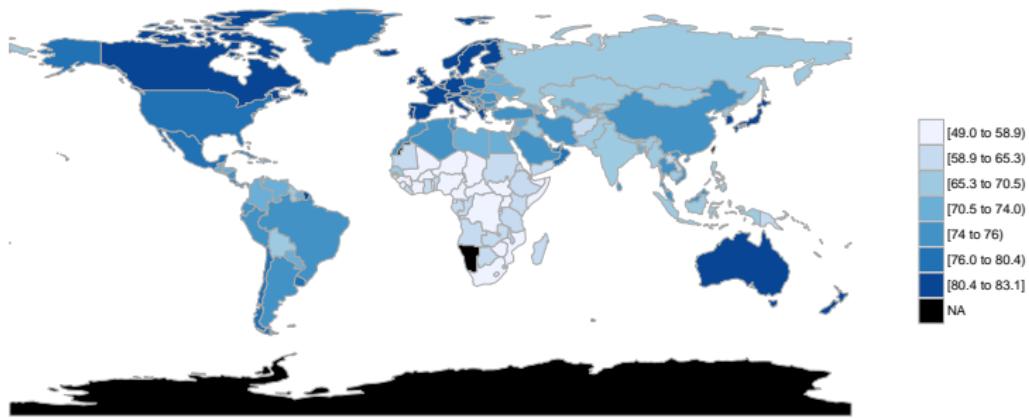
```
head(WDI_dat)
```

iso2c	country	year	AG.AGR.TRAC.NO	TM.TAX.TCOM.BC.ZS
1A	Arab World	1990	383795	NA
1A	Arab World	1991	402168	NA

Weltkarte mit den Weltbank Daten

```
choroplethr_wdi(code="SP.DYN.LE00.IN", year=2012,  
                 title="2012 Life Expectancy")
```

2012 Life Expectancy



Eurostat Daten

Sie können eine Statistik der Sparquote bei Eurostat downloaden.

<http://ec.europa.eu/eurostat/web/euro-indicators/peeis>

```
library(xlsx)
HHSR <- read.xlsx2("data/HHsavingRate.xls", 1)
```

geo	X2012Q3	X2012Q4	X2013Q1	X2013Q2	X2013Q3	X2013Q4
Euro area (19 countries)	9.82	11.86	11.37	16.28	10.0	10.0
EU (28 countries)	8.67	10.92	9.42	14.63	8.3	8.3
Belgium	12.52	9.33	13.99	19.03	12.0	12.0
Czech Republic	10.16	14.81	9.46	10.44	10.0	10.0
Denmark	7.04	2.56	10.15	8.44	5.4	5.4
Germany	15.06	14.77	19.45	16.07	15.0	15.0
Ireland	15.08	5.38	11.48	17.29	16.0	16.0
Spain	6.99	11.75	1.57	17.27	6.5	6.5

Zensus Ergebnisse

15

statistische Ämter stellen beim Zensus ein Team.



STARTSEITE

ZENSUS 2011

INFOTHEK

Ergebnisse des Zensus 2011 zum Download

Als Download bieten wir Ihnen auf dieser Seite zusätzlich zur Zensusdatenbank CSV- und teilweise Excel-Tabellen mit umfassenden Personen-, Haushalts- und Familien- sowie Gebäude- und Wohnungsmerkmalen. Die Ergebnisse liegen auf Bundes-, Länder-, Kreis- und Gemeindeebene vor. Außerdem sind einzelne Ergebnisse für Gitterzellen verfügbar.

Eine ausführliche methodische Beschreibung zu diesen Ergebnissen finden Sie in der Publikation

↳ [Zensus 2011 - Methoden und Verfahren \(PDF, 613KB, nicht barrierefrei\)](#).

Bevölkerung

Ergebnisse des Zensus 2011

- ↳ [Download-Tabelle "Bevölkerung" im Excel-Format \(xlsx, 10MB, nicht barrierefrei\)](#)
- ↳ [Download-Tabelle "Bevölkerung" im CSV-Format \(zip, 3MB, nicht barrierefrei\)](#)

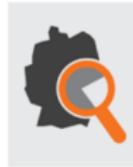
- ↳ [Bevölkerung nach Alter in Jahren und Geschlecht für Gemeinden \(zip, 6MB, nicht barrierefrei\)](#)
- ↳ [Bevölkerung nach Staat der Staatsangehörigkeit und Geschlecht für Gemeinden \(zip, 4MB, nicht barrierefrei\)](#)

[Zensus 2011 im Vergleich zur Bevölkerungsfortschreibung](#)

DAS HAT SICH GEÄNDERT

- Haushalte und Familien
- Migration
- Religion
- Erwerbstätigkeit
- Gebäude und Wohnungen

ZENSUSERGEBNISSE



Die Ergebnisse des Zensus 2011 finden Sie in der
↳ [Zensusdatenbank](#)

Zensus Atlas

<https://ergebnisse.zensus2011.de/>

Ergebnisse einfach und schnell	Ergebnisse dynamisch und individuell	Ergebnisse kartografisch und visuell
		

Zensus Gemeindeergebnisse

Gebiet	Bevölkerung mit Migrationshintergrund nach Regionen			
	Migrationshintergrund nach Regionen			
	EU27-Land	Sonstiges Europa	Sonstige Welt	Unbekanntes Ausland
7	144	145	146	147
Baden-Württemberg	1045490	1023790	602440	1620
Reg.-Bez. Stuttgart	439110	448380	218680	840
Stuttgart, Landeshauptstadt	87580	89330	47200	/
Stuttgart, Landeshauptstadt	87580	89330	47200	/
Böblingen	50560	44310	16510	/
Böblingen, Stadt	8690	8270	3030	/
Leonberg, Stadt	6480	3800	2270	/

Beispiel Zensus Daten

```
zen <- read.csv2("data/Zensus_extract.csv")
# Personen mit eigener Migrationserfahrung
# mit beidseitigem Migrationshintergrund
zen2 <- data.frame(Personen_Mig=zen[,which(zen[9,]==128)],
                    Personen_Mig_bs=zen[,which(zen[9,]==133)])
```



The easy way to get, use and share data

[Datasets](#)[Organizations](#)[About](#)[Blog](#)[Help](#)

[Home](#) / Organizations / Global / UNESCO World Heritage Sites / **List of UNESCO World ...**

List of UNESCO World Heritage Sites (Excel)

[Go to resource](#)

URL: <http://whc.unesco.org/en/list/xls/>

From the dataset abstract

The list of UNESCO World Heritage Sites, available for syndication in a number of formats. Use of this data requires permission from the publisher, see the website for details.

Source: [UNESCO World Heritage Sites](#)

 [Data Explorer](#)

Weltkulturerbestätten

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

	name_en	date_inscribed	longitude	latitude	area_hectares
4	Butrint	1992	20.02611	39.75111	NA
6	M'Zab Valley	1982	3.68333	32.48333	665.03
7	DjÃ©milâ	1982	5.73667	36.32056	30.60
8	Timgad	1982	6.63333	35.45000	90.54
27	Fraser Island	1992	153.13333	-25.21667	184000.00

Exkurs - OpenStreetMap Projekt

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.

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



Export von OpenStreetMap Daten

<www.openstreetmap.org/export>

The screenshot shows the OpenStreetMap export interface. At the top, there's a navigation bar with links for 'Bearbeiten', 'Chronik', 'Export', 'GPS-Tracks', 'Benutzer-Blogs', 'Urheberrecht', 'Hilfe', 'Über', 'Anmelden', and 'Registrieren'. Below the navigation bar is a search bar with placeholder text 'Suchen' and a dropdown menu showing 'Relation: Berlin (62422)'. To the right of the search bar are buttons for 'Wiederholen', 'Log', and a magnifying glass icon. The main area is a map of Berlin and its surroundings, featuring a large orange polygon highlighting a specific administrative region. The map includes various geographical features like roads, rivers, and parks, along with place names such as 'Falkensee', 'Panketal', 'Altlandsberg', and 'Schöneiche bei Berlin'. A legend on the left side of the map provides information about the map symbols. On the far left, there's a sidebar titled 'Attribute' containing a table with detailed data for the highlighted region. The table includes columns for 'ISO3166-2' (DE-BE), 'TMC cid_58 tabcd_1: Class' (Area), 'TMC cid_58 tabcd_1: LCLVersion' (12.0), 'TMC cid_58 tabcd_1: LocationCode' (266), 'admin_level' (4), 'at_name_v1' (BfB-In), 'boundary' (administrative), 'capital' (yes), 'contact:facebook' (<http://www.facebook.com/Berlin>), 'contact:website' (<http://www.berlin.de>), 'de:amtlicher_gemeindeschlussel' (11000000), 'de:place' (city), 'de:place:note' (Kreisfreie Stadt), 'de:regionalchirurgie' (110000000000), and 'geographical_region' (Berlin;Berliner Urstromtal;Teltow;Nau einer Platte). The bottom of the map has scale bars for 3 km and 2 mi.

ISO3166-2	DE-BE
TMC cid_58 tabcd_1: Class	Area
TMC cid_58 tabcd_1: LCLVersion	12.0
TMC cid_58 tabcd_1: LocationCode	266
admin_level	4
at_name_v1	BfB-In
boundary	administrative
capital	yes
contact:facebook	http://www.facebook.com/Berlin
contact:website	http://www.berlin.de
de:amtlicher_gemeindeschlussel	11000000
de:place	city
de:place:note	Kreisfreie Stadt
de:regionalchirurgie	110000000000
geographical_region	Barnim;Berliner Urstromtal;Teltow;Nau einer Platte

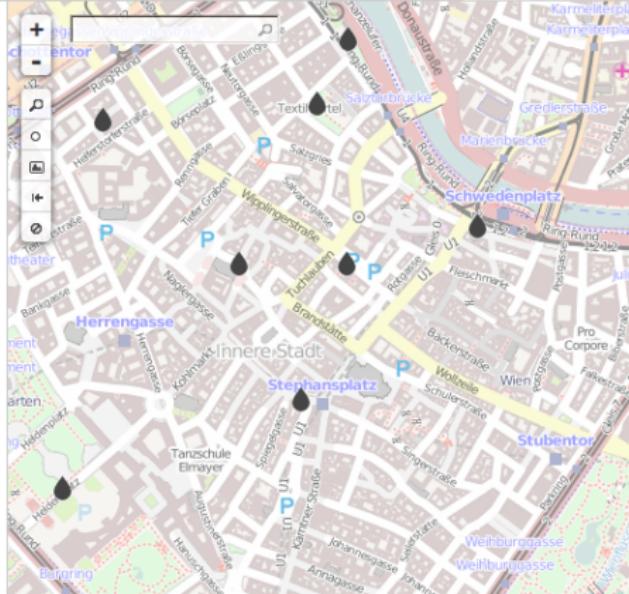
OpenStreetMap - Map Features

Roads					
These are the principal tags for the road network. They range from the most to least important.					
highway	motorway		A restricted access major divided highway, normally with 2 or more running lanes plus emergency hard shoulder. Equivalent to the Freeway, Autobahn, etc..		
highway	trunk		The most important roads in a country's system that aren't motorways. (Need not necessarily be a divided highway.)		
highway	primary		The next most important roads in a country's system. (Often link larger towns.)		
highway	secondary		The next most important roads in a country's system. (Often link towns.)		

Overpass Turbo

Ausführen Teilen Export Wizard Speichern Laden Einstellungen Hilfe overpass turbo ⓘ Karte Daten

```
1 <!--
2 This has been generated by the overpass-turbo wizard.
3 The original search was:
4 "Trinkbrunnen"
5 -->
6 <osm-script output="json" timeout="25">
7   <!-- gather results -->
8   <union>
9     <!-- query part for: "Trinkbrunnen" -->
10    <query type="node">
11      <has-kv k="amenity" v="drinking_water"/>
12      <bbox-query {bbox}>/>
13    </query>
14  </union>
15  <!-- print results -->
16  <print mode="body"/>
17  <recurse type="down"/>
18  <print mode="skeleton" order="quadtile"/>
19 </osm-script>
20
21 {{style:
22 node[amenity=drinking_water] {
23   icon-image: url("icons/maki/water-24.png");
24   icon-width: 24;
25 }
26 }}
```



Query Overpass

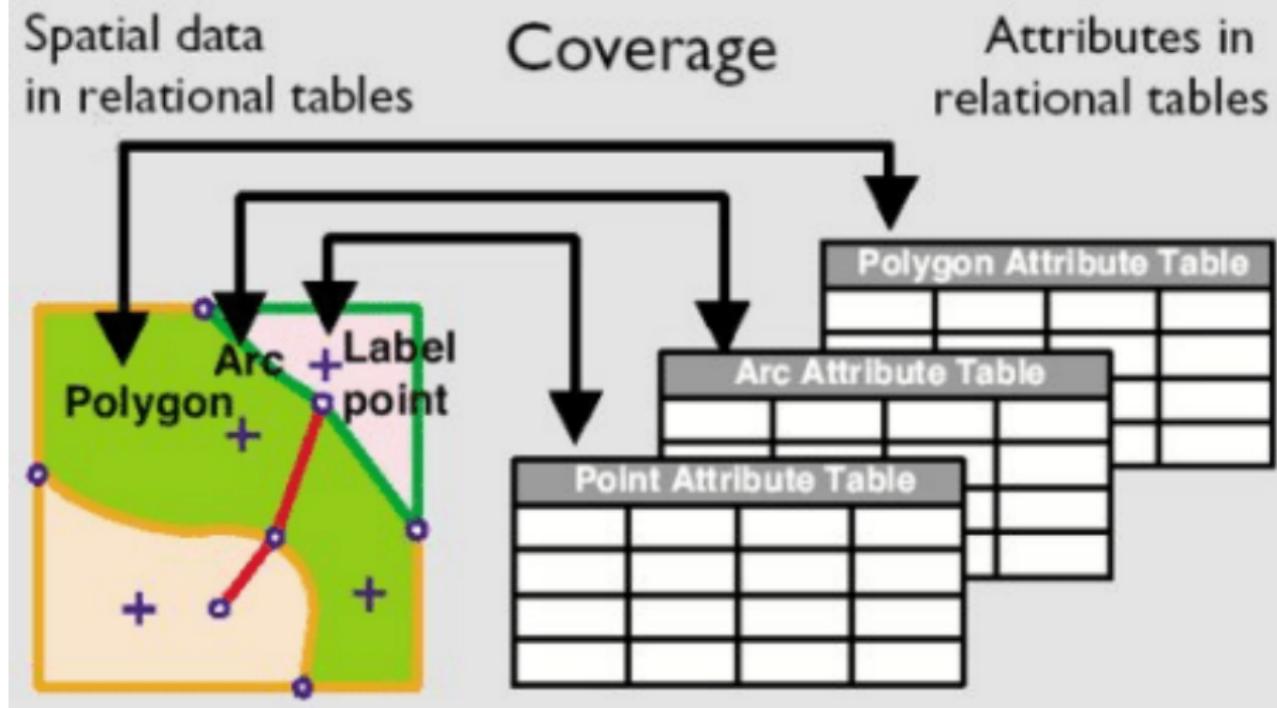
- Overpass hat eine eigene Abfragesprache

```
node
  [amenity=bar]
  ({{bbox}});
out;
```

Zwischenfazit - Quellen für inhaltliche Daten

- Gerade auf Staatenebene gibt es sehr viele Daten
- Oftmals hat man aber eigene Daten, die man in einer Karte darstellen möchte
- Frage ist wie sich die inhaltlichen Daten mit den Polygonzügen verbinden lassen

Verknüpfung von Daten



Quelle: Geographic Information Systems and Remote Sensing

Daten verbinden - Beispiel Eurostat Daten

- Wir hatten vorhin schon die Haushaltssparrate geladen
- Dabei handelt es sich um einen von den Ausgewählten Wichtigsten Europäischen Wirtschaftsindikatoren bei Eurostat
- Der Datensatz `wrld_simpl` aus dem Paket `maptools` wird verwendet
- Die Daten werden mit Hilfe des Ländernamens miteinander verknüpft

```
ind <- match(HHsr$geo, wrld_simpl@data$NAME)
ind <- ind[-which(is.na(ind))]
```

- Der `wrld_simpl` Datensatz wird auf Europa eingeschränkt

```
EUR <- wrld_simpl[ind, ]
```

Informationen zur Haushaltssparrate

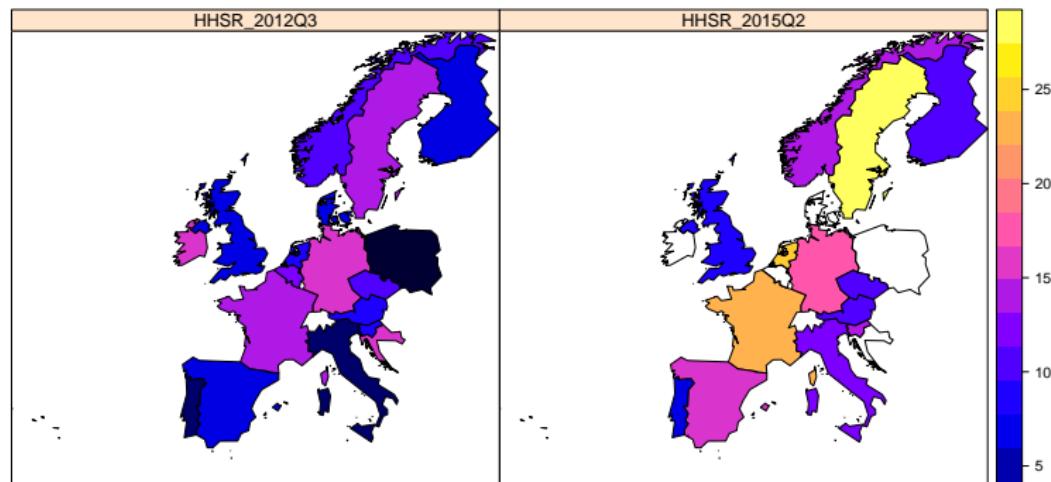
- Informationen zur Haushaltssparrate werden hinzugefügt

```
EUR@data$HHSR_2012Q3 <- as.numeric(as.character(  
  HHsr [-(1:2),2]))  
EUR@data$HHSR_2015Q2 <- as.numeric(as.character(  
  HHsr [-(1:2),13]))
```

Karte mit Eurostat Indikator Household Saving Rate

- Mit dem Befehl `spplot` aus dem Paket `sp` ist es möglich zwei Karten nebeneinander darzustellen:

```
spplot(EUR,c("HHSR_2012Q3","HHSR_2015Q2"))
```



Daten verbinden - Beispiel Bäckereien in Berlin

- Quelle für die folgenden Daten ist:

Geodaten von **OpenStreetMap**



die freie Weltkarte

OpenStreetMap.org

Lizenz



OSM als Datenquelle

- Zum Download habe ich die Overpass API verwendet

```
(load("data/info_bar_Berlin.RData"))
```

```
## [1] "info"
```

	addr.postcode	addr.street	name	
79675952	13405	Scharnweberstraße	Albert's	52
86005430	NA	NA	Newton Bar	52
111644760	NA	NA	No Limit Shishabar	52
149607257	NA	NA	en passant	52
248651127	10115	Bergstraße	Z-Bar	52
267780050	10405	Christburger Straße	Immertreu	52

Verwendung des Pakets gosmd

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

```
library("gosmd")
pg_MA <- get_osm_nodes(object="leisure=playground", "Mannheim")
pg_MA <- extract_osm_nodes(pg_MA, value='playground')
```

Matching

```
tab_plz <- table(info_be$addr.postcode)
```

```
ind <- match(BE@data$PLZ99_N, names(tab_plz))  
ind
```

```
## [1] 1 2 3 4 5 6 7 8 NA 9 NA NA NA NA NA NA 10 11 12  
## [24] 17 18 19 20 21 22 23 24 25 NA 26 27 28 29 NA NA NA NA  
## [47] 34 35 NA NA 36 37 38 39 40 41 42 43 44 45 46 47 48 49  
## [70] NA 54 55 NA NA NA 56 57 58 59 60 NA NA NA NA NA NA 61 NA  
## [93] NA NA NA NA NA NA NA 63 NA NA 64 NA 65 NA NA NA 66 NA  
## [116] NA 68 NA  
## [139] NA 69 70 NA 71 72 73 74 75 NA 76 NA NA NA NA NA NA NA  
## [162] 77 NA 78 79 NA NA NA NA 80 NA NA NA NA 81 NA 82 83 84  
## [185] NA NA NA 85 NA NA
```

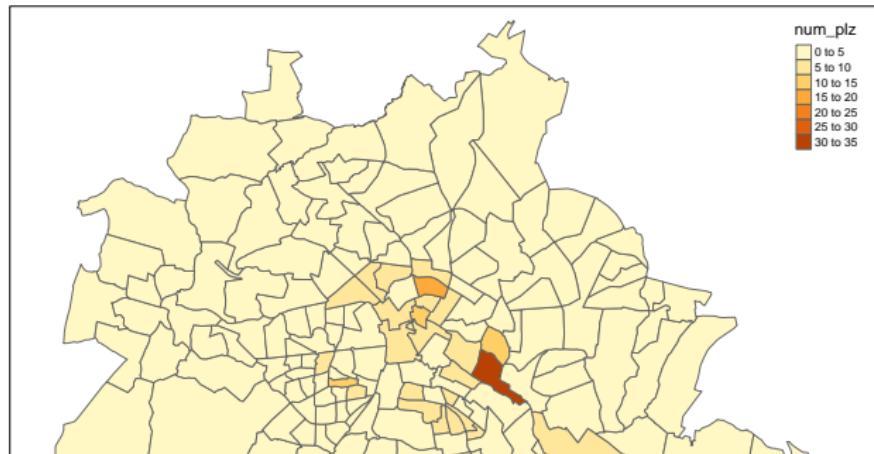
Daten anspielen

Das Paket tmap

```
library(tmap)
```

- mit qtm kann man schnell eine thematische Karte erzeugen

```
BE@data$num_plz[is.na(BE@data$num_plz)] <- 0  
qtm(BE, fill = "num_plz")
```



Mehr Informationen einbinden

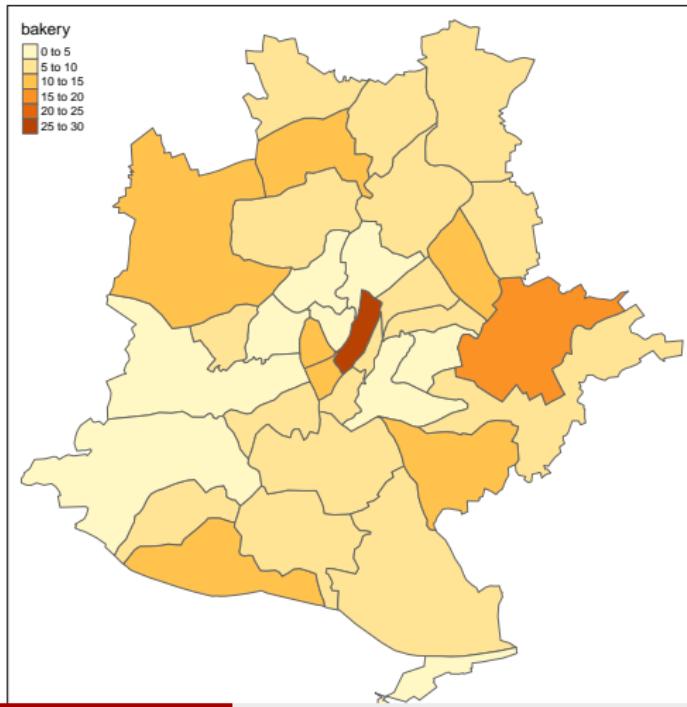
- Der folgende Datensatz ist eine Kombination aus den vorgestellten PLZ-Shapefiles und OSM-Daten die über Overpass heruntergeladen wurden:

```
load("data/osmsa_PLZ_14.RData")
```

	PLZ99	PLZ99_N	PLZORT99	nname	EWZ_gem	are
0	01067	1067	Dresden	Dresden, Stadt	512354	0.0008
1	01069	1069	Dresden	Dresden, Stadt	512354	0.0006
2	01097	1097	Dresden	Dresden, Stadt	512354	0.0004
3	01099	1099	Dresden	Dresden, Stadt	512354	0.0067
4	01109	1109	Dresden	Dresden, Stadt	512354	0.0034
5	01127	1127	Dresden	Dresden, Stadt	512354	0.0003

OSM-Daten - Bäckereien in Stuttgart

```
qtm(PLZ_SG,fill="bakery")
```



In welchem PLZ Bereich sind die meisten Bäckereien

```
kable(PLZ_SG@data[which.max(PLZ_SG$bakery),  
c("PLZ99","lat","lon","bakery")])
```

	PLZ99	lat	lon	bakery
4964	70173	48.7784485	9.1800132	30

Das R-Paket RDSTK

DATA SCIENCE TOOLKIT

Welcome to the Data Science Toolkit

Truly open tools for data.

Incorporates the city-level TwoFishes geocoder written by David Blackman at Foursquare.

Notice: Recent Outages

DSTK has been receiving a large amount of traffic lately, exceeding the limits of our servers. If you are using DSTK for business-critical applications, we strongly advise you host your own server so that you are not affected by outages.

- Data Science Toolkit API

```
library("RDSTK")
```

Die Daten für Stuttgart

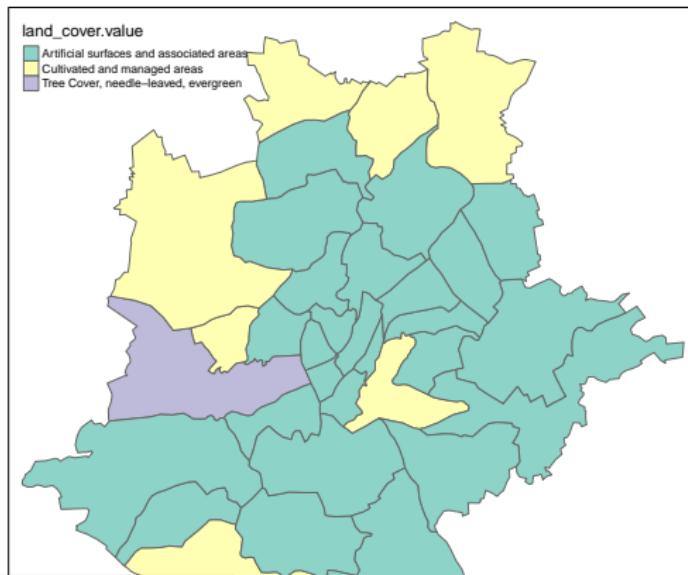
```
PLZ_SG <- PLZ[PLZ@data$PLZORT99=="Stuttgart",]
```

Type_landcover	Freq
Artificial surfaces and associated areas	26
Cultivated and managed areas	8
Tree Cover, needle-leaved, evergreen	1

Eine Karte der Flächenbedeckung erstellen

- Daten von European Commission Land Resource Management Unit Global Land Cover 2000.

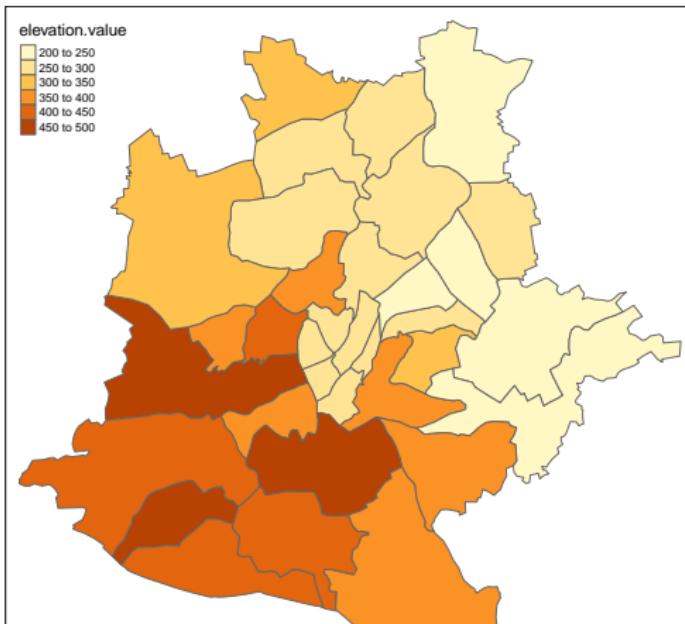
```
qtm(PLZ_SG,fill="land_cover.value")
```



Die Höhe in Stuttgart

- Daten von NASA and the CGIAR Consortium for Spatial Information .

```
qtm(PLZ_SG,fill="elevation.value")
```



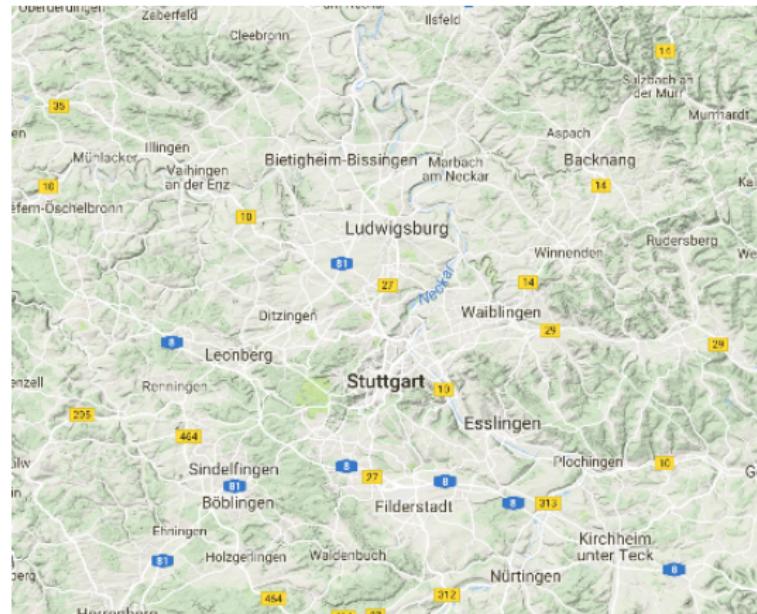
Graphiken Stadtleben Stuttgart - das Paket ggmap

```
devtools::install_github("dkahle/ggmap")
install.packages("ggmap")
```

Eine erste Karte mit ggmap erzeugen

```
library(ggmap)
```

```
qmap("Stuttgart")
```



Karte für einen ganzen Staat

`qmap("Germany")`



Ein anderes *zoom level*

- level 3 - Kontinent
- level 10 - Stadt
- level 21 - Gebäude

```
qmap("Germany", zoom = 6)
```



Karte für eine Sehenswürdigkeit

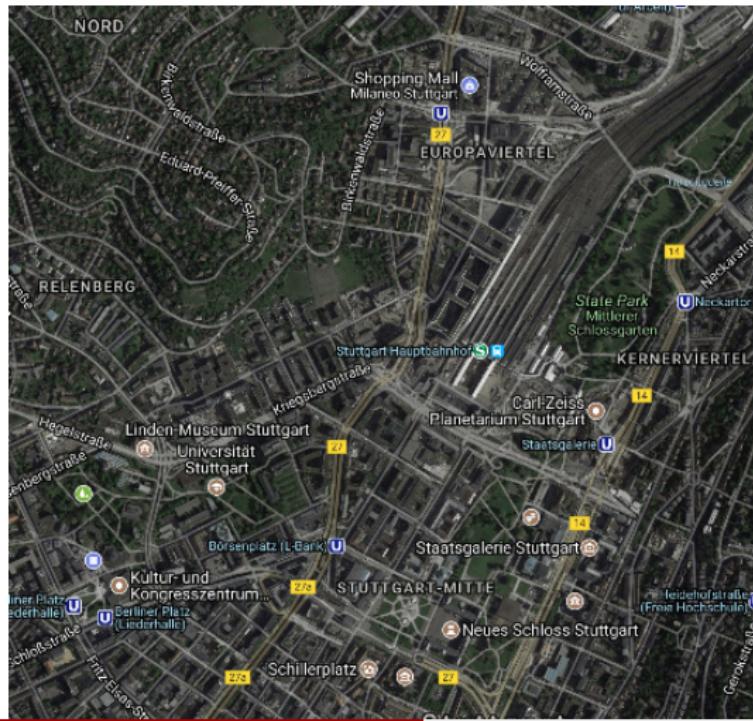
- ggmap - maptype satellite

```
WIL <- qmap("Wilhelma", zoom=20, maptype="satellite")  
WIL
```



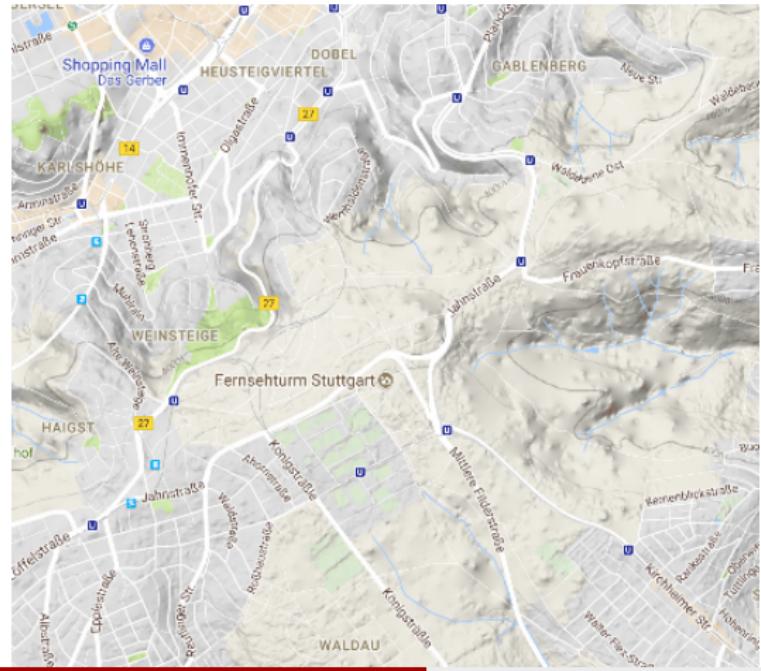
ggmap - maptype satellite zoom 20

```
qmap('Stuttgart Hauptbahnhof', zoom = 15, maptype="hybrid")
```



Eine terrain Karte

```
qmap('Stuttgart Fernsehturm', zoom = 14,  
      maptype="terrain")
```



ggmap - maptype watercolor

```
qmap('Stuttgart', zoom = 14,  
      maptype="watercolor",source="stamen")
```



ggmap - source stamen

```
qmap('Stuttgart', zoom = 14,  
      maptype="toner",source="stamen")
```



ggmap - maptype toner-lite

```
qmap('Stuttgart', zoom = 14,  
      maptype="toner-lite",source="stamen")
```



ggmap - maptype terrain-lines

```
qmap('Stuttgart', zoom = 14,  
      maptype="terrain-lines",source="stamen")
```



Geokodierung

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

Wikipedia - Geocoding

```
library(ggmap)  
geocode("Stuttgart")
```

```
##           lon         lat  
## 1 9.182932 48.77585
```

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

```
revgeocode(c(48,8))
```

```
## [1] "Unnamed Road, Somalia"
```

Die Distanz zwischen zwei Punkten

```
mapdist("Marienplatz Stuttgart", "Hauptbahnhof Stuttgart")
```

```
##                               from                      to      m      km
## 1 Marienplatz Stuttgart Hauptbahnhof Stuttgart 3136 3.136 1
##   minutes     hours
## 1 8.133333 0.1355556
```

```
mapdist("Marienplatz Stuttgart", "Hauptbahnhof Stuttgart",
        mode="walking")
```

```
##                               from                      to      m      km
## 1 Marienplatz Stuttgart Hauptbahnhof Stuttgart 2505 2.505 1
##   minutes     hours
## 1 31.23333 0.5205556
```

Eine andere Distanz bekommen

```
mapdist("Marienplatz Stuttgart", "Hauptbahnhof Stuttgart",
        mode="bicycling")
```

```
##                               from                      to      m      km
## 1 Marienplatz Stuttgart Hauptbahnhof Stuttgart 2722 2.722 1
##   minutes hours
## 1     8.1 0.135
```

Geokodierung - verschiedene Punkte von Interesse

```
POI1 <- geocode("B2, 1 Mannheim",source="google")
POI2 <- geocode("Hbf Mannheim",source="google")
POI3 <- geocode("Mannheim, Friedrichsplatz",source="google")
ListPOI <- rbind(POI1,POI2,POI3)
POI1;POI2;POI3
```

```
##           lon      lat
## 1 8.462844 49.48569
```

```
##           lon      lat
## 1 8.469879 49.47972
```

```
##   lon lat
## 1  NA  NA
```

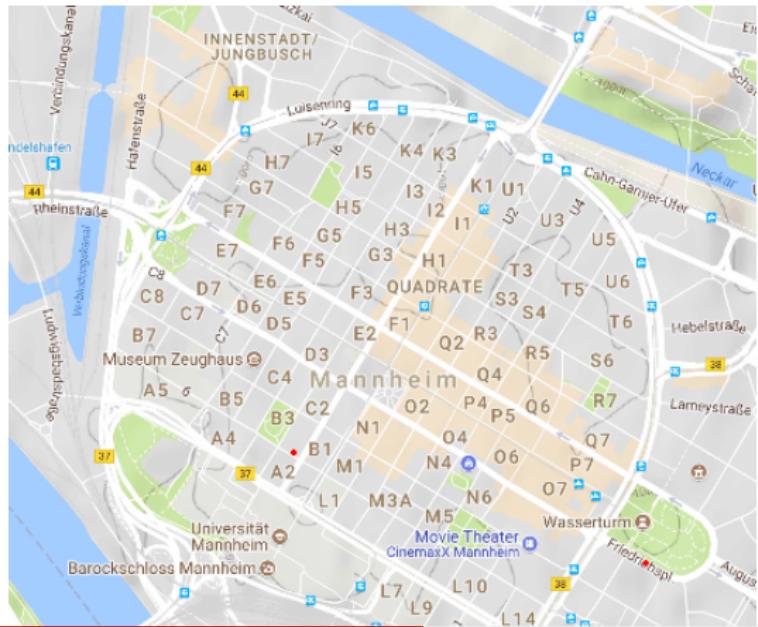
Punkte in der Karte

```
MA_map +
  geom_point(aes(x = lon, y = lat),
  data = ListPOI)
```

Punkte in der Karte

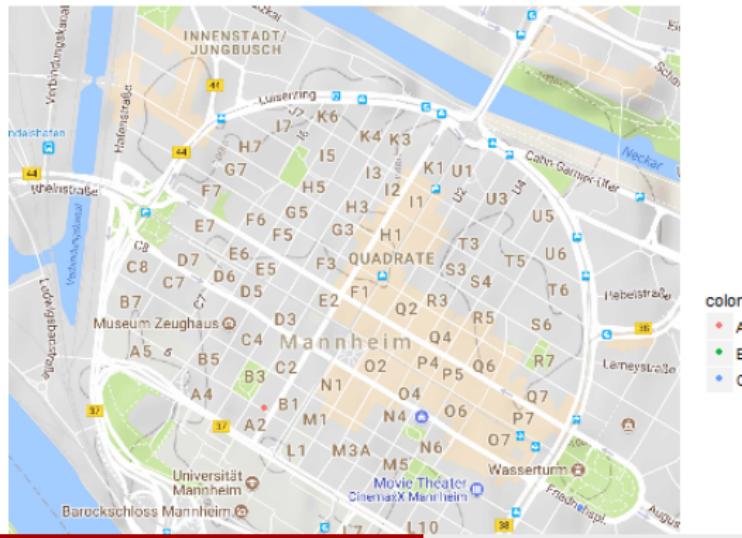
MA_map +

```
geom_point(aes(x = lon, y = lat), col="red",  
data = ListPOI)
```



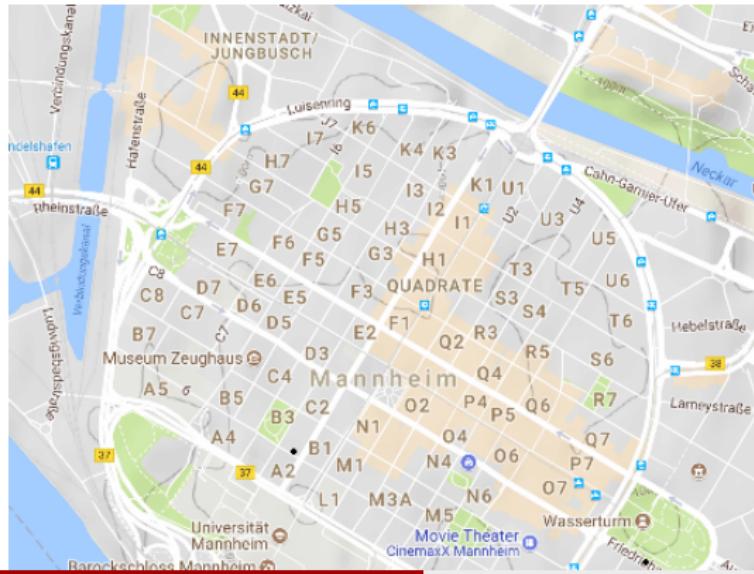
ggmap - verschiedene Farben

```
ListPOI$color <- c("A","B","C")  
MA_map +  
  geom_point(aes(x = lon, y = lat, col=color),  
  data = ListPOI)
```



ggmap - größere Punkte

```
ListPOI$size <- c(10,20,30)  
MA_map +  
  geom_point(aes(x = lon, y = lat, col=color, size=size),  
  data = ListPOI)
```



Eine Route von Google maps bekommen

```
from <- "Mannheim Hbf"  
to <- "Mannheim B2 , 1"  
route_df <- route(from, to, structure = "route")
```

Mehr Information

<http://rpackages.ianhowson.com/cran/ggmap/man/route.html>

Eine Karte mit dieser Information zeichnen

```
qmap("Mannheim Hbf", zoom = 14) +  
  geom_path(  
    aes(x = lon, y = lat), colour = "red", size = 1.5,  
    data = route_df, lineend = "round"  
)
```

Wie fügt man Punkte hinzu

- Nutzung von geom_point
- Question on stackoverflow

<http://i.stack.imgur.com>

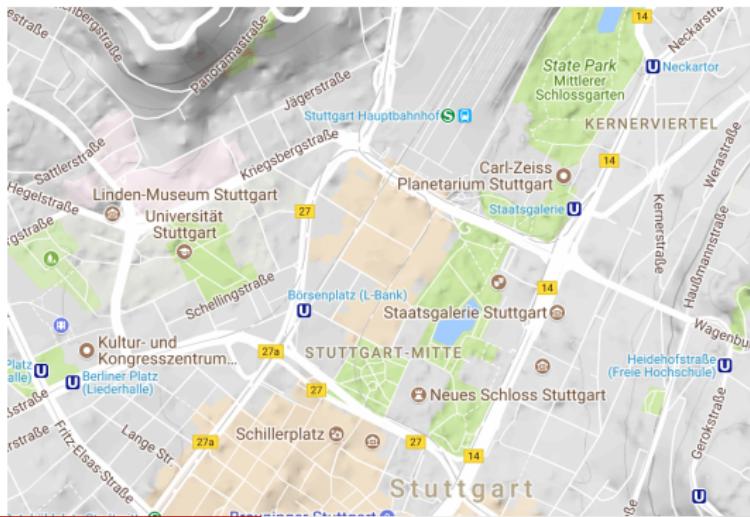
Cheatsheet

- Cheatsheet zu data visualisation

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

Das Paket ggmap

```
library(ggmap)
lon_plz <- PLZ_SG@data[which.max(PLZ_SG$bakery), "lon"]
lat_plz <- PLZ_SG@data[which.max(PLZ_SG$bakery), "lat"]
mp_plz <- as.numeric(c(lon_plz, lat_plz))
qmap(location = mp_plz, zoom=15)
```



Das Paket osmar

```
library(osmar)

src <- osmsource_api()
gc <- geocode("Stuttgart-Degerloch")
bb <- center_bbox(gc$lon, gc$lat, 800, 800)
ua <- get_osm(bb, source = src)
plot(ua)
```



Gebäude in diesem Ausschnitt plotten

```
bg_ids <- find(ua, way(tags(k=="building")))
bg_ids <- find_down(ua, way(bg_ids))
bg <- subset(ua, ids = bg_ids)
bg_poly <- as_sp(bg, "polygons")
plot(bg_poly)
```



Gebäude und Straßen im Ausschnitt



Schlussfolie

Vielen Dank für die Aufmerksamkeit

Erweiterungsmöglichkeiten

- Interaktive Karten mit dem Paket leaflet

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

- Die Nutzung der Overpass API mit R

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

Karte auf Carto zur Energieerzeugung

