

B2 - Geokodierung

Jan-Philipp Kolb

23 Oktober 2018

Inhalt dieses Abschnitts

- Das Konzept der Geokoordinaten erklären
- Möglichkeiten vorstellen, die Geokodierung mit R durchzuführen
- Nutzung der Nominatim API

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 ...

Geokodierung mit dem Paket ggmap

- Einer der ersten Ansätze Geokodierung mit R durchzuführen
- Wenn Geokodierung mit R durchgeführt wird dieses Paket wohl am häufigsten verwendet.
- Das führt auch dazu, dass im Internet zahlreiche Anwendungsbeispiele zu finden sind.

```
library(ggmap)
geocode("Heidelberg")
```

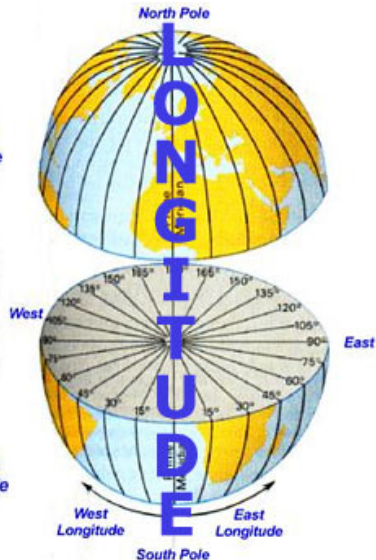
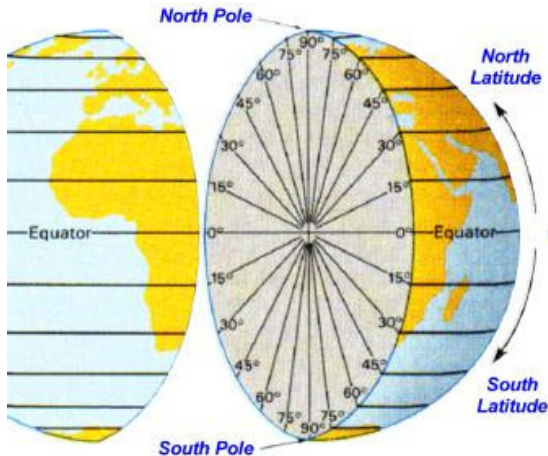
Information from URL : <http://maps.googleapis.com/maps/api/geocode>

	lon	lat
--	-----	-----

1	8.672434	49.39875
---	----------	----------

Latitude und Longitude

LATITUDE



Die Distanz zwischen zwei Punkten

Distanzen für verschiedene Verkehrsmittel

```
mapdist("Q1, 4 Mannheim", "B2, 1 Mannheim")
```

```
mapdist("Q1, 4 Mannheim", "B2, 1 Mannheim", mode="walking")
```

```
mapdist("Q1, 4 Mannheim", "B2, 1 Mannheim", mode="bicycling")
```

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
```

Koordinaten verschiedener Orte in Deutschland

Geokodierung mit einer Schleife

```
cities <- c("Hamburg", "Koeln", "Dresden", "Muenchen")

lat <- vector()
lon <- vector()
for (i in 1:length(cities)){
  gc <- geocode_OSM(cities[i])
  lat[i] <- gc$coords[1]
  lon[i] <- gc$coords[2]
}
```


Welche Koordinaten hat der Norden

```
Dat <- data.frame(cities,lon,lat)
kable(Dat)
```

cities	lon	lat
Hamburg	53.55034	10.000654
Koeln	50.93836	6.959974
Dresden	51.04933	13.738144
Muenchen	48.13711	11.575382

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))
```

Daten einlesen

- Hier wird ein Beispieldatensatz eingelesen, den ich über räumliche Stichproben und reverse geocoding erzeugt habe.

```
load("../data/addr_list_t_68239.RData")  
head(addr_list_t)
```

```
## [1] "Lilienstraße 32A, 68535 Edingen-Neckarhausen, Germany"  
## [2] "Waldspitze 6, 68239 Mannheim, Germany"  
## [3] "Holzweg 51, 68239 Mannheim, Germany"  
## [4] "Kloppenheimer Str. 247, 68239 Mannheim, Germany"  
## [5] "Mallaustraße 121, 68219 Mannheim, Germany"  
## [6] "Holzweg 33A, 68239 Mannheim, Germany"
```

Die erste Adressen geokodieren

```
geocode_OSM(addr_list_t[1])
```

```
## $query
```

```
## [1] "Lilienstraße 32A, 68535 Edingen-Neckarhausen, Germany"
```

```
##
```

```
## $coords
```

```
##           x           y
```

```
## 8.584601 49.445360
```

```
##
```

```
## $bbox
```

```
##           min           max
```

```
## x 8.584494 8.584708
```

```
## y 49.445276 49.445443
```

Alle Adressen geokodieren

```
gc_list <- list()

for (i in 1:length(addr_list_t)){
  gc_list[[i]] <- geocode_OSM(addr_list_t[i])
}
```

Geokodierung mit dem R-Paket `opencage`

- Um dieses Paket zu nutzen muss man sich vorher bei der API registrieren

```
library(opencage)
```

```
gc_info<-opencage_forward(placename =  
                        "Amsterdam, Van Woustraat")
```

- Hinweise, wie das Paket genutzt werden kann sind im **opencage Tutorial** zu finden.

Das Paket geonames

Nutzung des geonames Paketes

- Ein Account ist notwendig um die meisten Funktionen des Paketes geonames zu nutzen.

```
library(geonames)
```

```
options(geonamesUsername="myusername")
```

```
MAwiki<-GNfindNearbyWikipedia(postalcode=68239,country="DE",  
                               radius=10)
```

Beispiel Geonames

Wikipediaeinträge in der Nähe

- **Login** für die Nutzung des Web-Services Geonames.
- **Hier** kann man das Arbeiten mit dem Webservice starten.
- **Informationen zum Download bei Geonames**

Show entries

Search:

	elevation	feature	lng	distance	countryCode	rank	lang	title	lat	wikipediaUrl
1	102	city	8.46711	0.1738	DE	98	en	Quadratesstadt	49.48848	en.wikipedia.org/wiki/Quadratesstadt
2	103	landmark	8.46212	0.1986		90	en	Reiss Engelhorn Museum	49.48888	en.wikipedia.org/wiki/Reiss_Engelhorn_Museum
3	103	landmark	8.4616	0.2423	DE	13	en	Klappsmühl' am Rathaus	49.4891	en.wikipedia.org/wiki/Klappsmühl%27am_Rathaus
4	104	landmark	8.46294	0.3178	DE	84	en	GESIS – Leibniz Institute for the Social Sciences	49.485686	en.wikipedia.org/wiki/GESIS_%27Leibniz_Institute_for_the_Social_Sciences
5	102	city	8.4691	0.3258	DE	100	en	Mannheim	49.489	en.wikipedia.org/wiki/Mannheim

Eine Bounding Box erstellen

```
library(osmdata)
bbox <- getbb("Mannheim")

erg <- geonames::GNcities(49.649591, 8.627236,
                          49.329591, 8.307236)
```

Geokodieren mit der API Nominatim

```
library("RJSONIO")
con <- url("http://nominatim.openstreetmap.org/search?format=json&addressdetails=1&extratags=1&q=Amsterdam+Niederlande")
geoc <- fromJSON(paste(readLines(con, warn=F), collapse = ''))
close(con)
```

So sieht das Ergebnis aus

```
names(geoc[[1]])
```

```
## [1] "place_id"      "licence"        "osm_type"       "osm_id"
## [5] "boundingbox"   "lat"            "lon"            "display_
## [9] "class"         "type"           "importance"     "address"
## [13] "extratags"
```

```
geoc[[1]]$address
```

```
##      house_number      road      residential      su
##              "1"      "Rozengracht"      "Jordaan"      "Amster
## city_district      city      state      post
##      "Centrum"      "Amsterdam" "Noord-Holland"      "101
##      country      country_code
##      "Nederland"      "nl"
```

Das Paket jsonlite nutzen

```
con <- url("http://nominatim.openstreetmap.org/search?format=json&addressdetails=1&extratags=1&q=Amsterdam+Niederlande")
geoc2 <- jsonlite::fromJSON(con)

geoc2df <- with(geoc2, data.frame(osm_id, lat, lon))
geoc2df$house_number <- geoc2$address$house_number
save(geoc2df, file = "../data/Amsterdam_Rozengracht_geoc2df.RData")
```

osm_id	lat	lon	house_number
2721815875	52.3737223	4.8826404	1
2743624072	52.3719482	4.8755534	237-1
2721830930	52.3736673	4.8823914	7-1
2721827922	52.3734021	4.8813371	53-1
2721824637	52.372232	4.8767542	231-1
2721823434	52.3724786	4.8776618	187-1
2721820122	52.3727335	4.8786657	137-1

Das Paket googlaway

Accesses Google Maps APIs to Retrieve Data and Plot Maps

```
library(googlaway)
```

- Ein API Schlüssel ist notwendig um die meisten Funktionen des Paketes zu nutzen.

Das Paket bbox

- Das Paket bbox ist auf github zu finden.
- Beispieldatensatz laden:

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

- Rahmen für das räumliche Objekt bestimmen:

```
library(bbox)  
b_box(ddat)
```

```
## [1] 5.866286 47.273602 15.048632 55.058262
```

```
citation("bbox")
```

- Überblick von Jesse Sadler zur **Geokodierung mit R**
- Ein Schummelzettel für **ggmap**
- Die Vignette zum Paket tmap - **tmap: get started**
- **latlong.net** - eine Homepage um Koordinaten zu bestimmen.