

# B2 - Geokodierung

Jan-Philipp Kolb

23 Oktober 2018

# Inhalt dieses Abschnitts

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

## 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

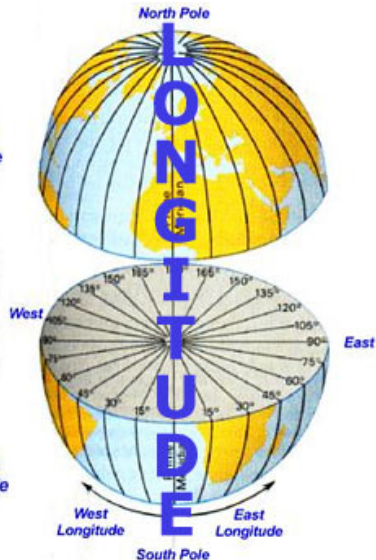
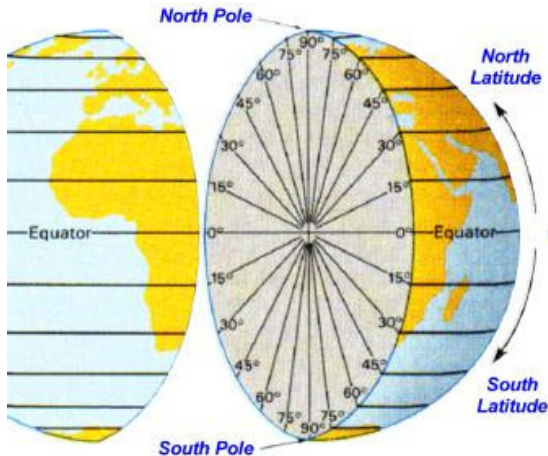
- Einer der ersten Ansätze Geokodierung mit R durchzuführen
- Wenn Geokodierung mit R durchgeführt wird dieses Paket wohl am häufigsten verwendet.
- Das führt auch dazu, dass im Internet zahlreiche Anwendungsbeispiele zu finden sind.

```
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

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

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

Eine andere Distanz bekommen

```
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")
```

# Koordinaten verschiedener Orte in Deutschland

```
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

# Geokodierung - verschiedene Punkte von Interesse

```
POI1 <- geocode_OSM("B2, 1 Mannheim")
POI2 <- geocode_OSM("Hbf Mannheim")
POI3 <- geocode_OSM("Mannheim, Friedrichsplatz")
ListPOI <- data.frame(rbind(POI1$coords, POI2$coords, POI3$coords))
ListPOI
```

# Punkte in der Karte

```
MA_map <- qmap("Mannheim")
```

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

# Punkte in der Karte

```
MA_map +  
geom_point(aes(x = x, y = y), col="red",  
data = ListPOI)
```

# 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

```
install.packages("geonames")
```

- 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)
```

# Wikipediaeinträge in der Nähe

elevation	feature	lng	distance	countryCode	rank	lang	title
102	city	8.46711	0.1738	DE	98	en	Qua
103	landmark	8.46212	0.1986	NA	90	en	Reis
103	landmark	8.4616	0.2423	DE	13	en	Klap
104	landmark	8.46294	0.3178	DE	84	en	GES
102	city	8.4691	0.3258	DE	100	en	Mar

- Login für Geonames
- Link um mit den Geodaten zu arbeiten
- Informationen über den Download

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

```
erg <- geonames::GNcities(49.649591,8.627236,
```

# 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.