

A6 - Shapefiles

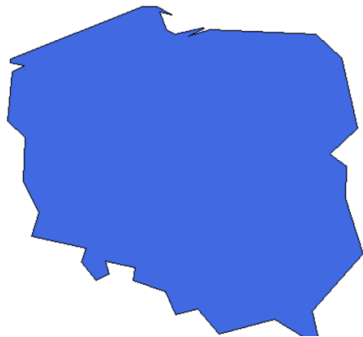
Jan-Philipp Kolb

22 Oktober 2018

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

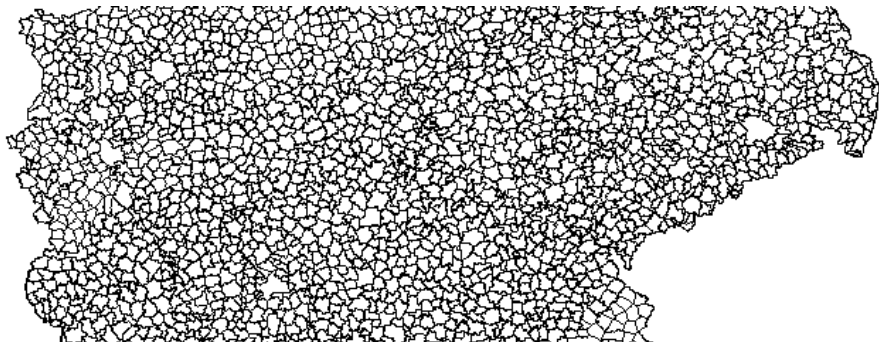
Quelle Ortsnetzbereiche: Bundesnetzagentur

- **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

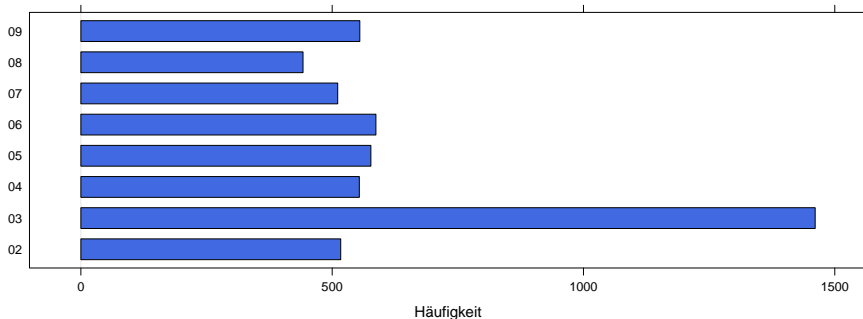
```
kable(head(onb@data))
```

	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

Einen Vorwahlbereich ausschneiden

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

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



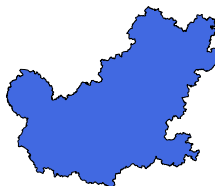
Vorwahlbereich ausschneiden

```
vw6 <- onb[vwb2=="06",]  
plot(vw6)
```



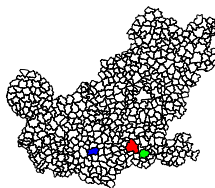
Shapefiles zusammenfassen

```
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")
```

```
## 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
```

A6A Übung - 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 Administrative 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
```

```
##           [,1]      [,2]  
## [1,] 6.026519 50.17767  
## [2,] 6.031361 50.16563  
## [3,] 6.035646 50.16410  
## [4,] 6.042747 50.16157  
## [5,] 6.043894 50.16116  
## [6,] 6.048243 50.16008
```

Der Datenslot

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

```
##      OBJECTID ID_0 ISO      NAME_0 ID_1      NAME_1 HASC_1 CCM
## 1           1  131 LUX Luxembourg  1      Diekirch  LU.DI
## 2           2  131 LUX Luxembourg  2 Grevenmacher  LU.GR
## 3           3  131 LUX Luxembourg  3      Luxembourg LU.LU
##      TYPE_1 ENGTYPE_1 NL_NAME_1      VARNAME_1
## 1 District  District      Dikrech|Dikkrich
## 2 District  District      Gréivemaacher
## 3 District  District      Lëtzebuerg|Luxemburg
```

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)
```



A6A Aufgabe: Download von Shapefiles für die Gemeinden Deutschlands

- Lade die Shapefiles Datei (UTM32 Kompakt) von **hier** herunter.
- Entpacke den zip-Ordner und importiere den Shapefile (VG250_F.shp) mit den Gemeinden, mit einer geeigneten Funktion.

Kreise eines Bundeslandes

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



A6B Aufgabe: Eine Karte für das Saarland erzeugen

- Schränke die Daten auf das Saarland ein, und zeichne eine Karte vom Saarland.
- Speichere den Datensatz in geeigneter Form ab.

Andere Quellen

- World Port Index

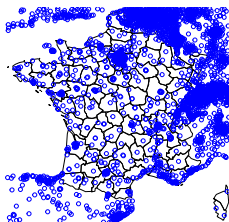
```
library(rgdal)
WPI <- readOGR ("WPI.shp", "WPI")
plot(WPI)
```



Das Paket maps - Mehr Information

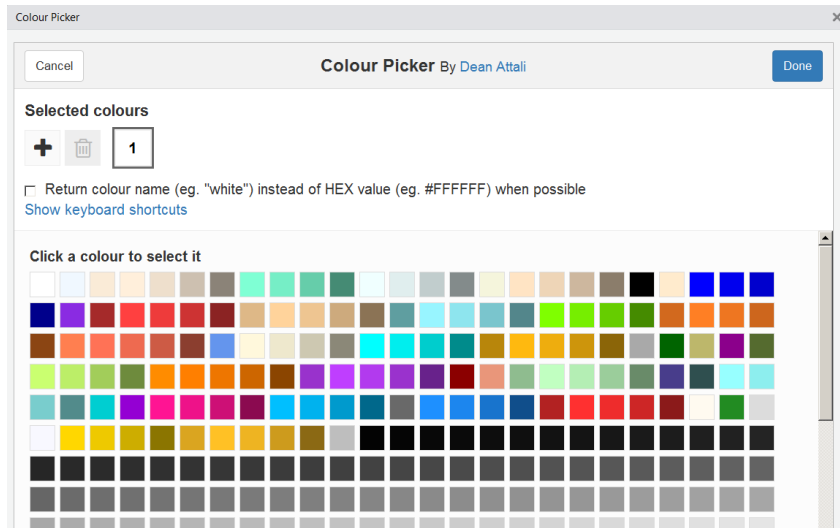
- Nur für manche Staaten bekommt man mit dem Paket maps Umkreise für Einheiten unterhalb der Staatsgrenze (bspw. Frankreich, USA).

```
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-Vermerkes ¹ ist es Ihnen gestattet, die Inhalte uneingeschränkt zu verwenden.

[Dateiformat](#)

[Bildformat](#)

[Inhalt](#)

[Geometrie ²](#)

Shapefiles bei Eurostat

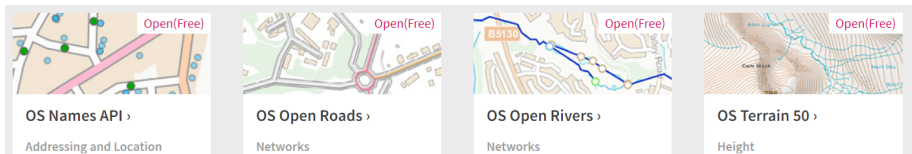
- **Eurostat Karten** - in der Regel die Europäischen Mitgliedsstaaten

The screenshot shows the Eurostat website interface. At the top, there's a navigation bar with the Eurostat logo, a search bar, and links for 'Anmelden', 'Registrieren', 'Rechtlicher Hinweis', 'RSS', 'Cookies', 'Links', 'Kontakt', and a language dropdown set to 'Deutsch'. Below the navigation bar, a blue banner contains the text 'eurostat Ihr Schlüssel zur europäischen Statistik'. A secondary navigation bar has links for 'Neuigkeiten', 'Daten', 'Veröffentlichungen', 'Über Eurostat', and 'Hilfe'. The main content area shows a breadcrumb trail: 'Europäische Kommission > Eurostat > GISCO: Geographische Informationen und Karten N... > Geografische Daten > Geobasisdaten > Verwaltungseinheiten / Statistische Einheiten > Zensus'. On the left, a sidebar lists 'Übersicht', '* Geografische Daten', '* GISCO-Aktivitäten', and 'Häufig gestellte Fragen'. The main content area is titled 'Zensus' and contains a table of census data. A note above the table states: '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

Weitere Quellen für Shapefiles

- **Open linked data - Ordnance Survey (GB)**



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