

Spatial Visualisations

First maps in R

Jan-Philipp Kolb

Monday, October 20, 2014



Outline

Preparations

Package maps

Package maptools

Package dismo

R-package ggmap

Preparations

- ▶ Create a new directory - for this course
- ▶ Start Rstudio
- ▶ Open a new script and save it in your directory

Preparations

Write a header like this one:

```
1 - #####  
2 # Spatial Mannheim  
3 # Jan-Philipp Kolb  
4 # 17.10.2014  
5 #  
6 # Choropleths  
7 #  
8 - #####  
9
```

Preparations

- ▶ Copy the adress of your new directory to your script
- ▶ Change the backslashes to slashes
- ▶ And write the following:

```
graph.path <- "C:/Users/Kolb/Spatial"
```

Outline

Preparations

Package maps

Package maptools

Package dismo

R-package ggmap

R-package maps

Package ‘maps’

September 22, 2014

Title Draw Geographical Maps

Version 2.3-9

Date 2014-09-22

Author Original S code by Richard A. Becker and Allan R. Wilks.
R version by Ray Brownrigg <Ray.Brownrigg@ecs.vuw.ac.nz>.
Enhancements by Thomas P Minka <tpminka@media.mit.edu>

Description Display of maps. Projection code and larger maps are in separate packages (mapproj and mapdata).

Depends R (>= 2.10.0)

LazyLoad yes

Suggests mapproj (>= 1.2-0)

License GPL-2

Maintainer Ray Brownrigg <Ray.Brownrigg@ecs.vuw.ac.nz>

<http://cran.r-project.org/web/packages/maps/maps.pdf>

R-package maps

Low resolution map of the world:

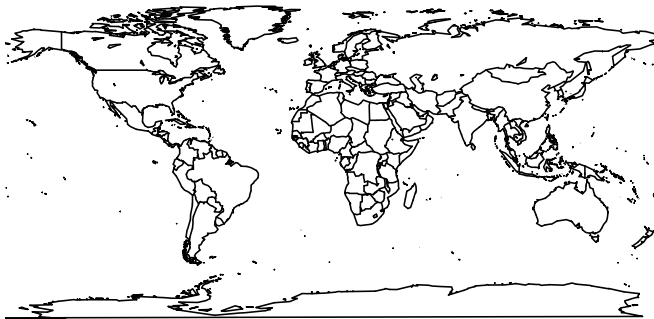
First you have to load the library:

```
library(maps)
```

Then the first command without arguments:

```
map()
```


R-package maps



How to get help

You get help for every command:

```
?map
```

map {maps}

R Documentation

Draw Geographical Maps

Description

Draw lines and polygons as specified by a map database.

R-package maps

National boundaries:

You need to do that only once:

```
library(maps)
```

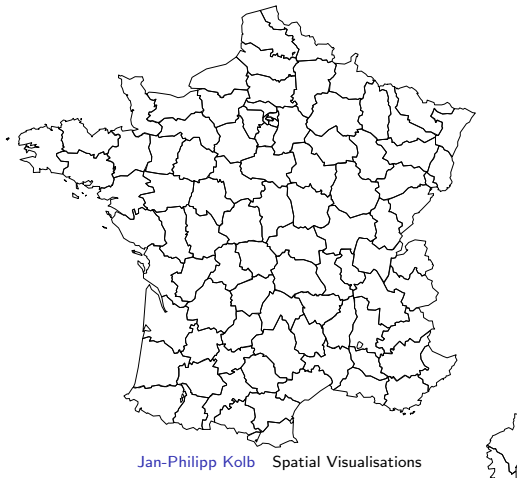
The same command with an argument:

```
map("usa")
```

R-package maps



R-package maps - France



Choropleths - package maps

There is also a map for Italy:

```
library(maps)  
map("italy", col = "blue")
```

Get the borders in blue color:



Choropleths - package maps

If we want the areas in blue:

```
map("italy", fill=T, col = "blue")
```



Choropleths - package maps

We can also have different colors:

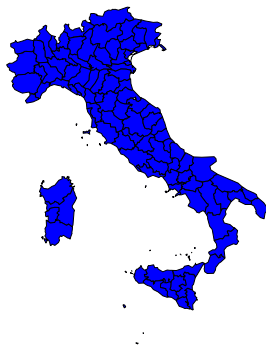
```
map("italy", fill=T, col = 1:10)
```



Choropleths - package maps

If you only type 4 you get blue:

```
map("italy", fill=T, col = 4)
```



Excursus: more colors!

You can also use the `rgb()` command to create your own colors:

```
map("italy", fill=T, col = rgb(0,1,0))
```

Try also:

```
map("italy", fill=T, col = rgb(1,0,0))  
map("italy", fill=T, col = rgb(1,1,1))  
map("italy", fill=T, col = rgb(1,0.5,0.4))
```

Choropleths - package maps

If you want to know, which region is at which place:

```
italy <- map("italy", plot = F)
head(italy$names)
```

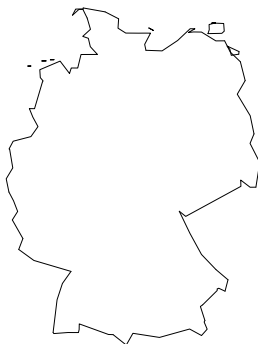
And you get the following results:

```
[1] "Bolzano-Bozen" "Belluno" "Udine" "Sondrio"
[5] "Trento" "Novara"
```

http://blog.lib.umn.edu/moor0554/canoemoore/R_Workshop_Spatial_032609.pdf

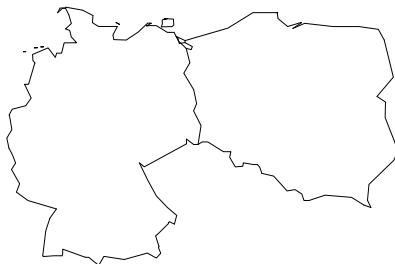
Choropleths

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



Choropleths - package maps - two countries

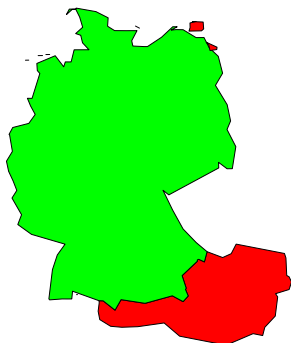
```
map("world", c("Germany", "Poland"))
```



Choropleths - package maps - two countries

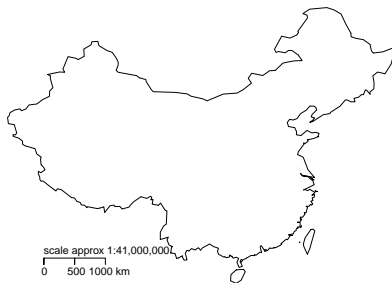
Something similar with color:

```
map("world", c("Germany", "Austria"), fill=T,  
col=c("red", "green"))
```



Choropleths - package maps - additional features

```
map("world", "China")  
map.scale()
```

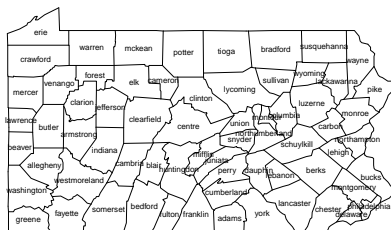


Choropleths - package maps - additional features

Like `map`, but labels the regions:

```
map.text("county", "penn")
```

Pennsylvania counties:



R-package maps - World cities

`data()` loads specified data sets, or lists the available data sets.

```
data(world.cities)
```

`head()` - Return the First Part of an Object

```
head(world.cities)
```

R-package maps - World cities

	name	country.etc	pop	lat	long	capital
1	'Abasan al-Jadidah	Palestine	5629	31.31	34.34	0
2	'Abasan al-Kabirah	Palestine	18999	31.32	34.35	0
3	'Abdul Hakim	Pakistan	47788	30.55	72.11	0
4	'Abdullah-as-Salam	Kuwait	21817	29.36	47.98	0
5	'Abud	Palestine	2456	32.03	35.07	0
6	'Abwein	Palestine	3434	32.03	35.20	0

R-package maps - World cities

```
map()  
map.cities(world.cities)
```

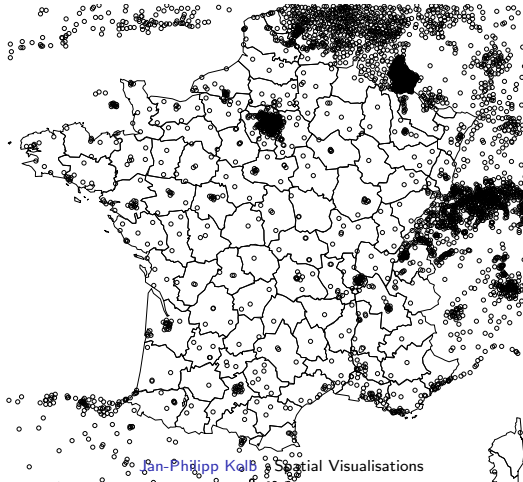
R-package maps - World cities



R-package maps - Cities france

```
data(world.cities)
map("france")
map.cities(world.cities)
```

R-package maps - Cities france



R-package maps - Cities france

We only want to have the french cities:

```
FrenchCity <- world.cities$country.etc=="France"
```

```
FCit <- world.cities[FrenchCity,]
```

```
head(FCit)
```

R-package maps - Cities france

	name	country.etc	pop	lat	long	capital
195	Abbeville	France	26656	50.12	1.83	0
318	Acheres	France	23219	48.97	2.06	0
477	Agde	France	23477	43.33	3.46	0
479	Agen	France	34742	44.20	0.62	0
643	Aire-sur-la-Lys	France	10470	50.64	2.39	0
648	Aix-en-Provence	France	148622	43.53	5.44	0

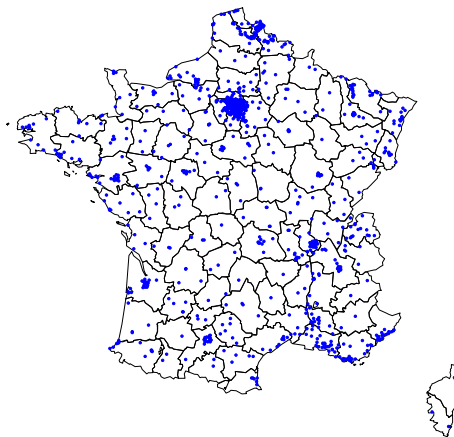
R-package maps - Cities france

Now it is possible to map only the french cities:

```
map("france")  
map.cities(FCit, col="blue", pch=20)
```

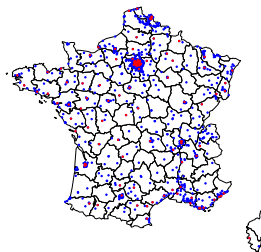
`pch` - plotting character, i.e., symbol to use.

R-package maps - Cities france



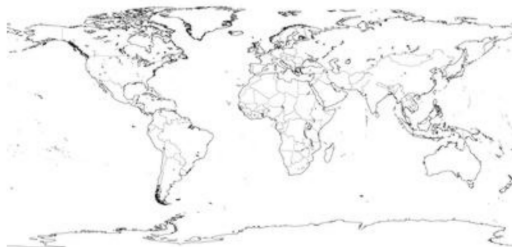
R-package maps - Cities france

```
map("france")  
map.cities(FCit,col="blue",pch=20)  
map.cities(FCit_Bc,col="red",pch=20)
```



R-package maps - data basis - CIA World DataBank II

CIA World DataBank II



The CIA World DataBank is a collection of world map data, consisting of vector descriptions of land outlines, rivers, / political boundaries. It was created by U.S. government in the 1980s.

<http://www.evl.uic.edu/pape/data/WDB/>

Outline

Preparations

Package `maps`

Package `maptools`

Package `dismo`

R-package `ggmap`

R-package maptools

Package ‘maptools’

July 2, 2014

Version 0.8-30

Date 2014-06-05

Title Tools for reading and handling spatial objects

Encoding UTF-8

Depends R (>= 2.10), sp (>= 1.0-11)

Imports foreign (>= 0.8), methods, grid, lattice, stats

Suggests rgeos (>= 0.1-8), spatstat (>= 1.18), PBSmapping, RColorBrewer

Enhances maps, gplib, RArcInfo

Description Set of tools for manipulating and reading geographic data, in particular **ESRI shape-files**; C code used from shapelib. It includes binary access to GSHHG shoreline files. The package also provides interface wrappers for exchanging spatial objects with packages such as PB-Smapping, spatstat, maps, RArcInfo, Stata tmap, WinBUGS, Mondrian, and others.

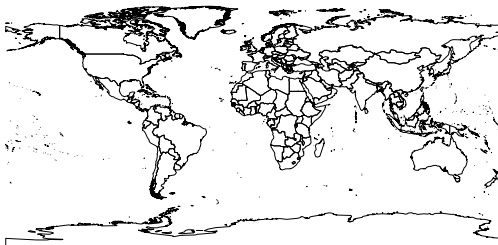
R-package maptools - World

Many roads lead to Rome....

```
data(wrld_simpl)
plot(wrld_simpl)
```

http://www.innet-project.eu/sites/default/files/Bibiko_HowToDrawAMapmap_New.pdf

R-package maptools - World



R-package `maptools`

- ▶ Looks like `maps`-package
- ▶ But there is much more information included
- ▶ The information is organized in a different way:

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

R-package maptools - South Africa

```
SouthAfrica <- wrld_simpl[  
wrld_simpl@data$NAME == "South Africa",]  
plot(SouthAfrica)
```



R-package maptools - Poland

We create an indicator:

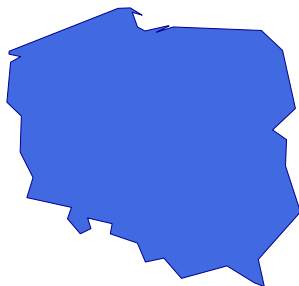
```
Pol <- wrld_simpl@data$NAME=="Poland"  
table(Pol)
```

With `table()` we get the following result:

```
Pol  
FALSE  TRUE  
  245     1
```

R-package maptools - Poland

```
Poland <- wrld_simpl[Pol==T,]  
plot(Poland,col="royalblue",border="darkblue")
```



R-package maptools - Europe

Create a map with the European Countries:

```
EuropeList <- c('Germany', 'France')  
my_map <- wrld_simpl[wrld_simpl$NAME %in% EuropeList, ]
```

R-package maptools - Parts of the world

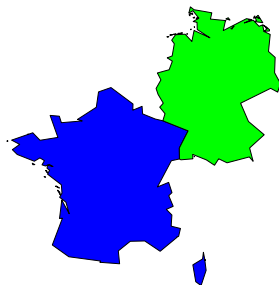
Draw only parts of the world

```
my_map <- wrld_simpl[wrld_simpl$NAME %in% c('Germany',  
'France'),]  
plot(my_map)
```

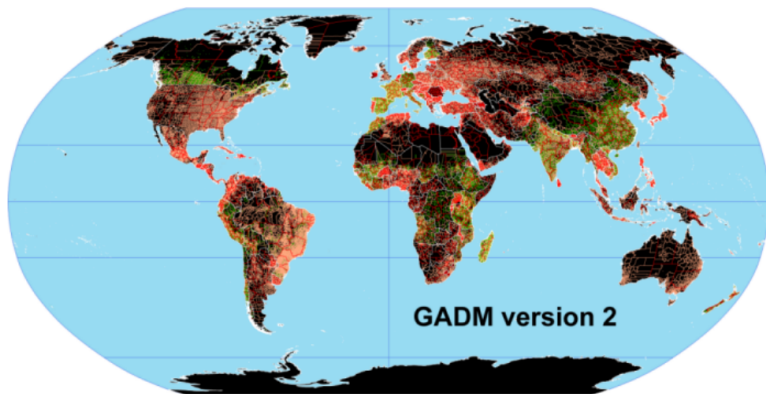


R-package maptools - More info

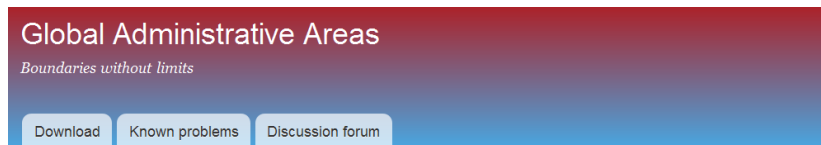
```
my_map@data$color <- c("blue","green")  
plot(my_map,col=my_map@data$color)
```



GADM database of Global Administrative Areas



GADM database of Global Administrative Areas



GADM database of Global Administrative Areas

GADM is a spatial database of the location of the world's administrative areas (or administrative boundaries) for use in [GIS](#) and similar software. Administrative areas in this database are countries and lower level subdivisions such as provinces, departments, bibhag, bundeslander, daerah istimewa, fivondronana, krong, landsvæðun, opština, sous-préfectures, counties, and thana. GADM describes where these administrative areas are (the "spatial features"), and for each area it provides some attributes, such as the name and variant names.

Example - Usage of GADM

sp is a package that provides classes and methods for spatial data.

```
library(sp)
```

Download information from gadm.org:

```
con <- url("http://gadm.org/data/rda/CHE_adm1.RData")  
print(load(con))  
close(con)
```

Example - Usage of GADM

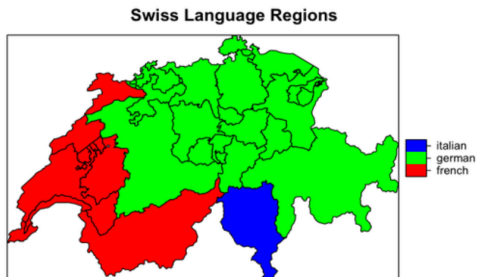
```
language <- c(rep("german",6),rep("french",2),  
rep("german",2),rep("french",2),"german","french",  
rep("german",7),  
"italian","german","french","french","german","german")  
gadm$language <- as.factor(language)
```

Define colours and plot the map with spplot

```
col = rainbow(length(levels(gadm$language)))  
spplot(gadm, "language", col.regions=col, main="Swiss  
Language Regions")
```

[http://blog.revolutionanalytics.com/2009/10/
geographic-maps-in-r.html](http://blog.revolutionanalytics.com/2009/10/geographic-maps-in-r.html)

Example - Usage of GADM



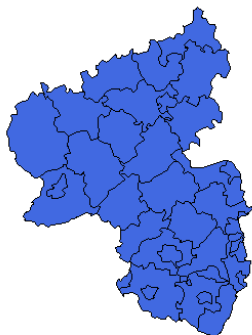
R-package rworldmap

```
library(rworldmap)
newmap <- getMap(resolution = "low")
plot(newmap)
```

<http://www.milanor.net/blog/?p=534>

Package maptools

```
D.KRS <- readShapePoly("vg2500_krs.shp")  
BLA <- substr(D.KRS@data$RS,1,2)  
plot(D.KRS[BLA=="07",],col="royalblue")
```



Outline

Preparations

Package maps

Package maptools

Package dismo

R-package ggmap

R-package dismo

```
library(dismo)
mymap <- gmap("France") # choose whatever country
plot(mymap)
```

<http://pakillo.github.io/R-GIS-tutorial/#gmap>

R-package dismo - France

R-package dismo - Mannheim

```
MAMap <- gmap("Mannheim")  
plot(MAMap)
```

R-package dismo - Mannheim

R-package dismo

```
mymap <- gmap("France", type = "satellite")  
plot(mymap)
```

<http://pakillo.github.io/R-GIS-tutorial/#gmap>

R-package ggmap

How can I draw a map of MODERN Europe?

```
library(ggmap)
library(mapproj)
map <- get_map(location = 'Europe', zoom = 4)
ggmap(map)
```

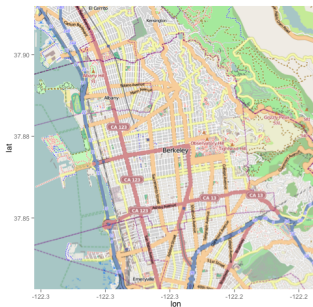
R-package ggmap

```
map3 <- get_map(location = "Berkeley, California",  
zoom = 13, source = "osm")
```

http://rstudio-pubs-static.s3.amazonaws.com/10823_e15ce99b55424ac9ad57c2ca11bf636c.html

R-package ggmap

```
map3 <- get_map(location = "Berkeley, California",  
zoom = 13, source = "osm")  
ggmap(map3)
```



http://rstudio-pubs-static.s3.amazonaws.com/10823_e15ce99b55424ac9ad57c2ca11bf636c.html

R-package ggmap

```
qmap("Redlands, California", zoom = 10,  
source = "stamen", maptype = "watercolor")
```



http://rstudio-pubs-static.s3.amazonaws.com/10823_e15ce99b55424ac9ad57c2ca11bf636c.html

R-package ggmap

```
qmap("Redlands, California", zoom = 13,  
source = "stamen", maptype = "toner")
```



http://rstudio-pubs-static.s3.amazonaws.com/10823_e15ce99b55424ac9ad57c2ca11bf636c.html