Spatial Visualisations

Further reading

Jan-Philipp Kolb

Monday, November 10, 2014

Outline

Get the right map

Adding color, points bubbles and lines

The sp-package

Further R-packages

Gliederung

Get the right map

Adding color, points bubbles and lines

The sp-package

Further R-packages

R-package `maptools` - Get country shapefiles

```r
library(maptools)
data(wrld_simpl)
```

Country codes (ISO2)

Get the country codes for Europe under the following web page:

http://www.countrycallingcodes.com/iso-country-codes/europe-codes.php

```
library(XML)
link <- "http://www.countrycallingcodes.com/iso-country-cod

tables <- readHTMLTable(link)
ISO2 <- as.character(tables[[2]][,2])
ISO2 <- ISO2[-(53:56)]
ISO2 <- ISO2[-(1:2)]
```

Subselecting Europe

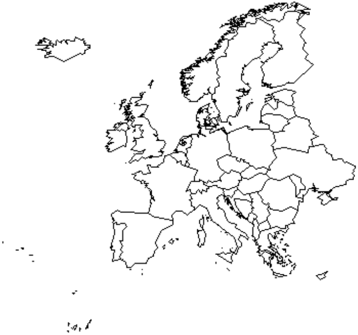
We are not displaying Russia, because it is so big:

```
ISO2 <- ISO2[-which(ISO2=="RU")]
```

```
ind <- match(ISO2,wrld_simpl$ISO2)
Europe <- wrld_simpl[ind,]
```

Plotting Europe

```
plot(Europe)
```



Gliederung

Get the right map

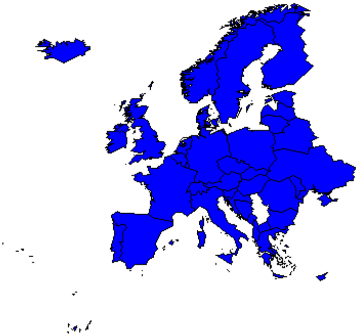
Adding color, points bubbles and lines

The sp-package

Further R-packages

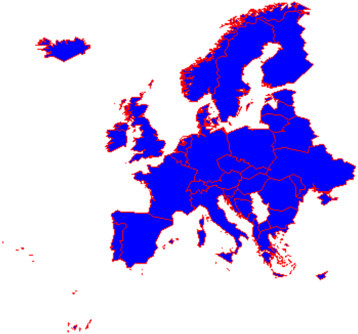
Plotting Europe

```
plot(Europe,col="blue")
```



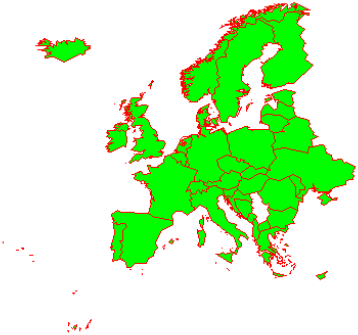
Plotting Europe

```
plot(Europe,col="blue",border="red")
```



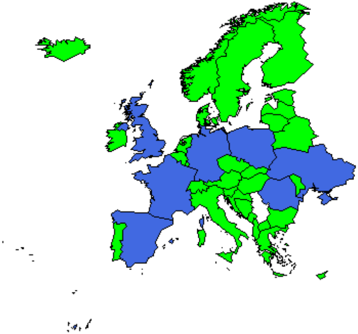
Plotting Europe

```
Europe$colors <- "green"
plot(Europe,col=Europe$colors,border="red")
```



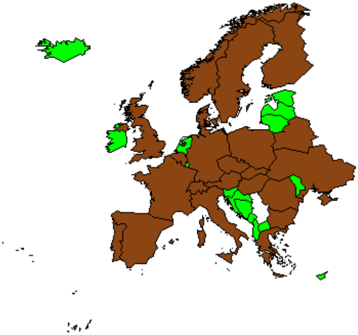
Plotting Europe

```
pop05 <- Europe$POP2005
Europe$colors[pop05>mean(pop05)] <- "royalblue"
plot(Europe,col=Europe$colors)
```



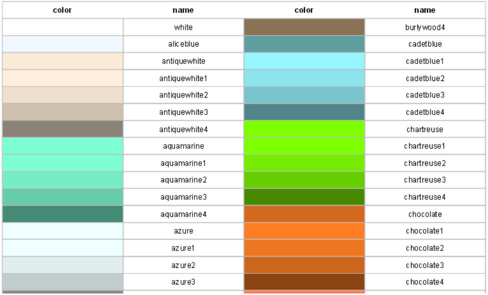
Plotting Europe

```
Europe$colors[pop05>median(pop05)] <- "chocolate4"
plot(Europe,col=Europe$colors)
```



Colors in R

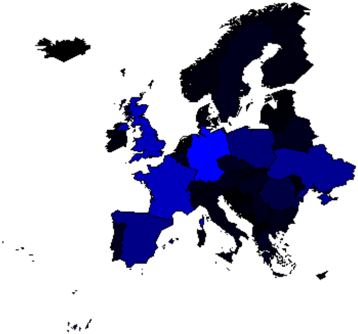
http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf



color	name	color	name
⬜	white	🟫	burlywood4
⬜	aliceblue	🟦	cadetblue
⬜	antiquewhite	🟦	cadetblue1
⬜	antiquewhite1	🟦	cadetblue2
⬜	antiquewhite2	🟦	cadetblue3
⬜	antiquewhite3	🟦	cadetblue4
🟫	antiquewhite4	🟩	chartreuse
🟩	aquamarine	🟩	chartreuse1
🟩	aquamarine1	🟩	chartreuse2
🟩	aquamarine2	🟩	chartreuse3
🟩	aquamarine3	🟩	chartreuse4
🟩	aquamarine4	🟧	chocolate
⬜	azure	🟧	chocolate1
⬜	azure1	🟧	chocolate2
⬜	azure2	🟧	chocolate3
⬜	azure3	🟫	chocolate4

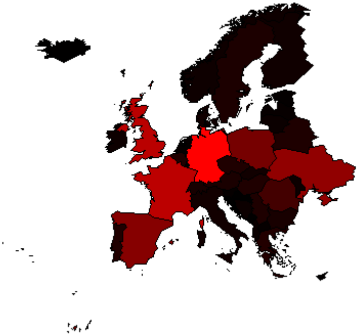
Plotting Europe - shading blue

```
val <- Europe$POP2005/max(Europe$POP2005)
plot(Europe,col=rgb(0,0,val))
```



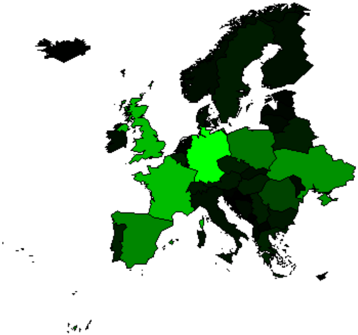
Plotting Europe - shading red

```
val <- Europe$POP2005/max(Europe$POP2005)
plot(Europe,col=rgb(val,0,0))
```



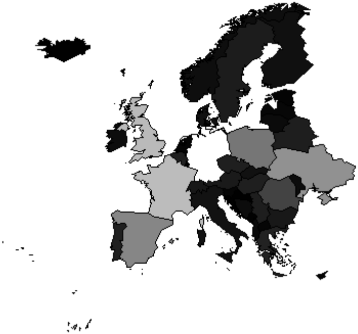
Plotting Europe - shading green

```
val <- Europe$POP2005/max(Europe$POP2005)
plot(Europe,col=rgb(0,val,0))
```



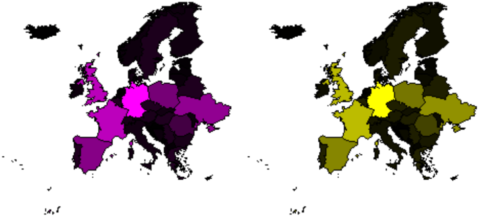
Plotting Europe - shading gray

```
val <- Europe$POP2005/max(Europe$POP2005)
plot(Europe,col=rgb(val,val,val))
```



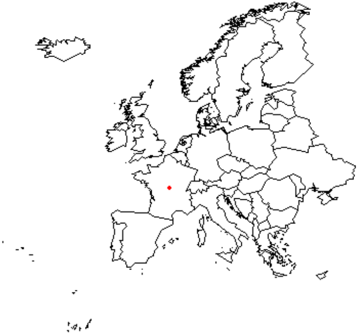
Plotting Europe - two plots beside

```
par(mfrow=c(1,2))
plot(Europe,col=rgb(val,0,val))
plot(Europe,col=rgb(val,val,0))
```



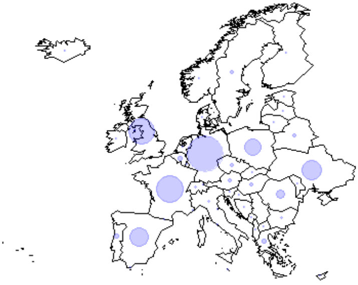
Plotting Europe - add points

```
which(Europe$ISO2=="FR") # 14
plot(Europe)
points(Europe$LON[14],Europe$LAT[14],col="red",pch=20)
```



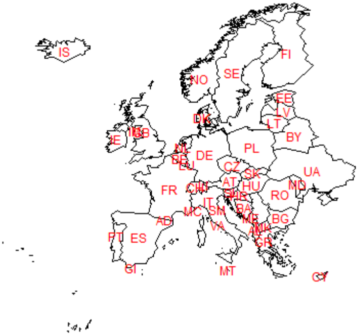
Plotting Europe - add bubbles

```
pop <- Europe$POP2005
pop <- pop/max(pop)*10
plot(Europe)
points(Europe$LON,Europe$LAT,cex=pop,col=rgb(0,0,1,.2),
pch=20)
```



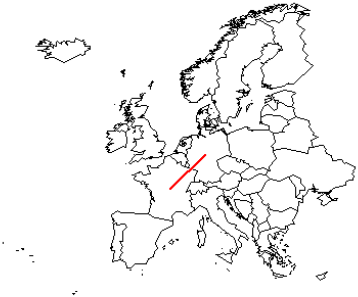
Plotting Europe - add text

```
plot(Europe)
text(Europe$LON,Europe$LAT,Europe$ISO2,col="red")
```



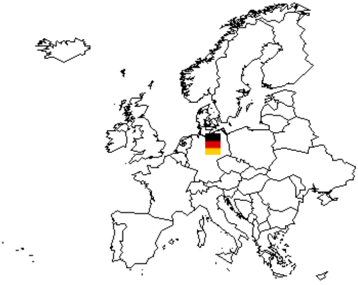
Plotting Europe - add lines

```
which(Europe$ISO2=="FR") # 15
which(Europe$ISO2=="DE") # 16
Dat <- cbind(Europe$LON[15:16],Europe$LAT[15:16])
plot(Europe)
lines(Dat,col="red",lwd=2)
```



Plotting Europe - add symbols

```
library(png)
fDEU <- readPNG("DEUflag.png")
fDEU<- as.raster(fDEU[,,1:3])
plot(Europe)
rasterImage(fDEU,Europe$LON[16],Europe$LAT[16],13,54)
```



Gliederung

Get the right map

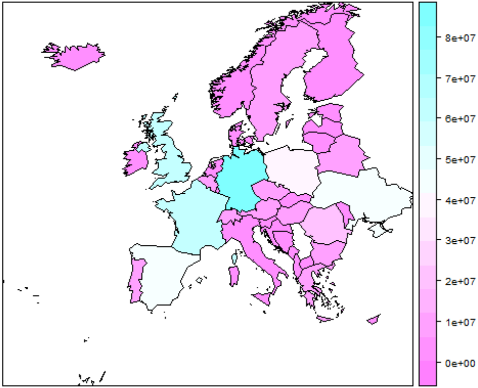
Adding color, points bubbles and lines

The sp-package

Further R-packages

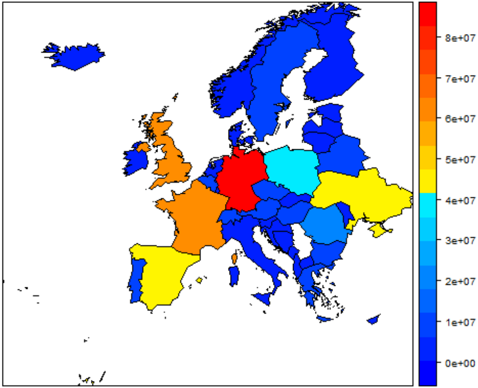
Plotting Europe - R-package sp

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



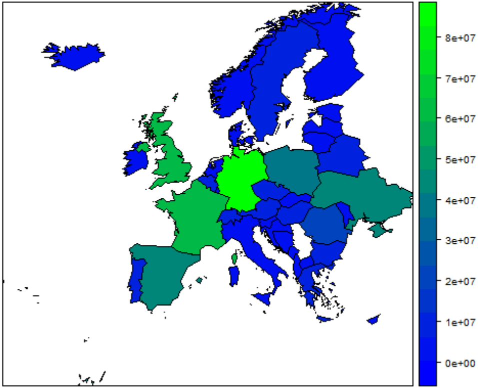
Plotting Europe - R-package sp

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



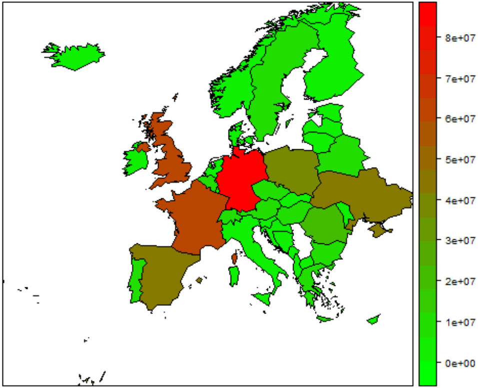
Plotting Europe - R-package sp

```
library(colorRamps)
spplot(Europe,"POP2005",col.regions=blue2green(100))
```



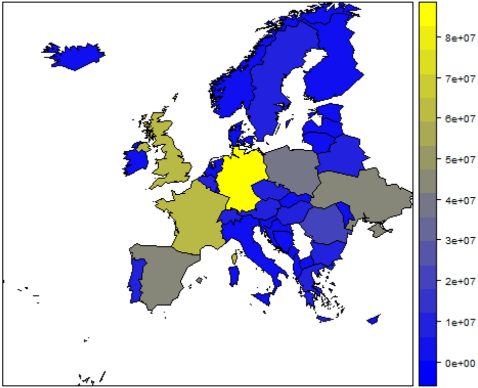
Plotting Europe - R-package sp

```
library(colorRamps)
spplot(Europe,"POP2005",col.regions=green2red(100))
```



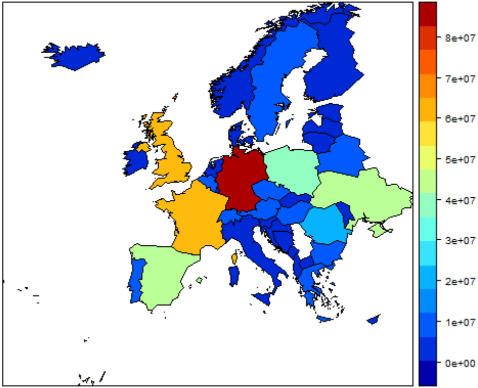
Plotting Europe - R-package sp

```
library(colorRamps)
spplot(Europe,"POP2005",col.regions=blue2yellow(100))
```



Plotting Europe - R-package sp

```
library(colorRamps)
spplot(Europe,"POP2005",col.regions=matlab_like(100))
```



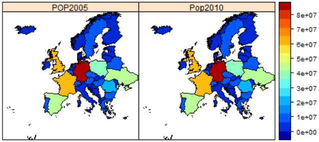
Plotting Europe - R-package sp

Generating synthetic data (Population 2010

```
Europe$Pop2010 <- Europe$POP2005 +
runif(length(Europe),-10000,10000)
```

And plotting two variables:

```
spplot(Europe,c("POP2005","Pop2010"),
       col.regions=matlab.like(100))
```



Outline

Get the right map

Adding color, points bubbles and lines

The sp-package

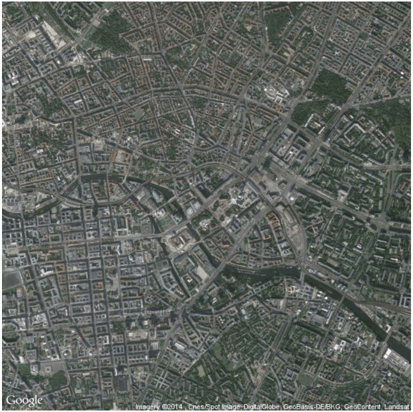
Further R-packages

R-package RgoogleMaps

```
library(RgoogleMaps)
newmap <- GetMap(center = c(52.52001,13.40495), zoom = 14,
                 destfile = "Berlin.png",
                 maptype = "satellite")
```

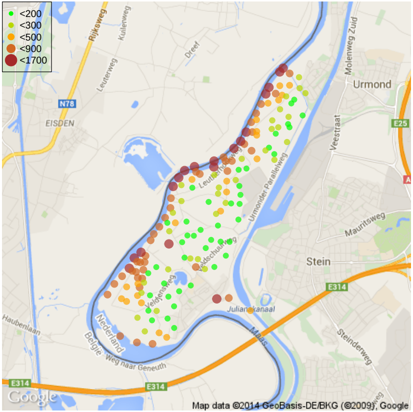
R-package RgoogleMaps

```
PlotOnStaticMap(newmap)
```


Google
Imagery ©2014 , Cnes/Spot Image, DigitalGlobe, GeoBasis-DE/BKG, GeoContent, Landsat

R-package RgoogleMaps

```
? bubbleMap
```

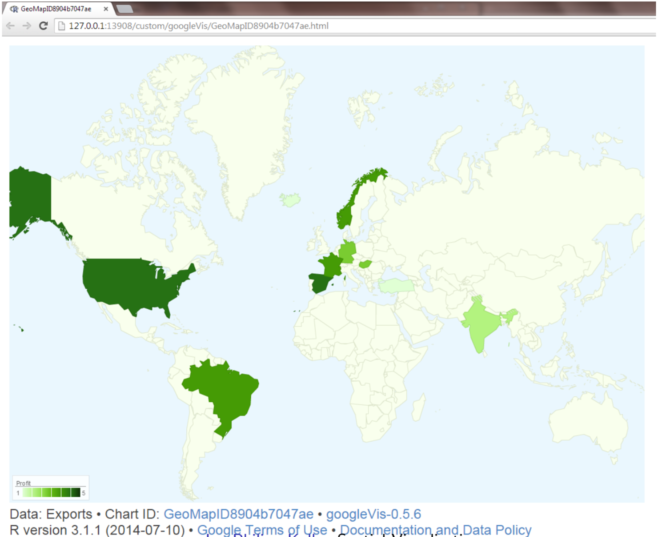


R-package googleVis

```
library(googleVis)

data(Exports)    # a simple data frame
Geo <- gvisGeoMap(Exports, locationvar="Country",
numvar="Profit",options=list(height=400,
dataMode=’regions’))
plot(Geo)
```

R-package googleVis



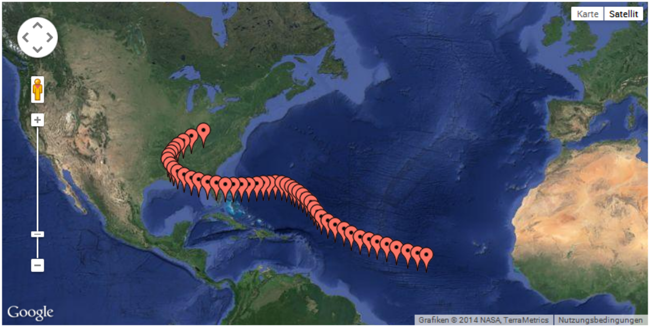
Data: Exports • Chart ID: GeoMapID8904b7047ae • googleVis-0.5.6
R version 3.1.1 (2014-07-10) • Google Terms of Use • Documentation and Data Policy

R-package googleVis

Example: Plotting point data onto a google map (internet)

```r
data(Andrew)
M1 <- gvisMap(Andrew, "LatLong", "Tip",
              options=list(showTip=TRUE, showLine=F,
enableScrollWheel=TRUE, mapType=’satellite’,
useMapTypeControl=TRUE, width=800,height=400))
plot(M1)
```

R-package googleVis



`http://pakillo.github.io/R-GIS-tutorial/#googlevis`

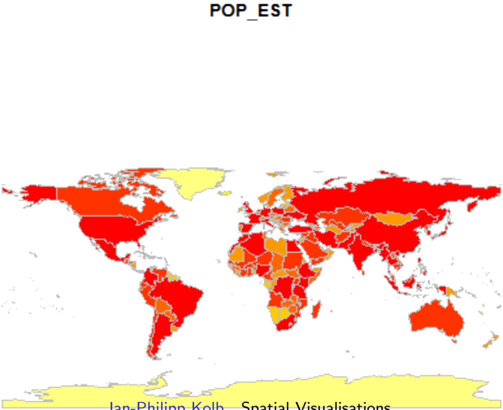
R-package rworldmap

```r
library(rworldmap)
# different resolutions available
newmap <- getMap(resolution = "coarse")
plot(newmap)
```

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

R-package rworldmap

```
vignette('rworldmap')
mapCountryData()
```

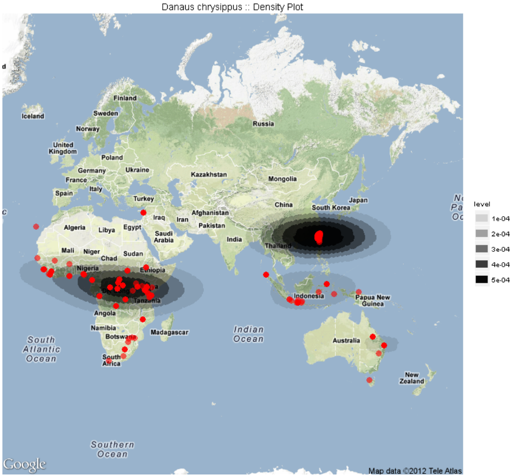


R-package dismo

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

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

R-package dismo

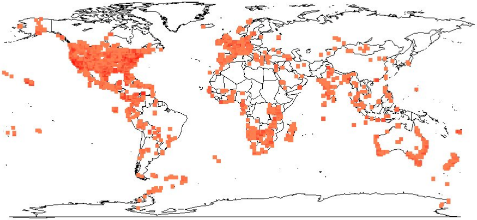


http://vijaybarve.files.wordpress.com/2012/07/dan_chr3.png

R-package dismo - Mannheim

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

R-package dismo



https://vijaybarve.files.wordpress.com/2013/04/world_density1.jpg

R-package dismo

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

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