

# Beautiful world maps in R with rworldmap

Andy South

[southandy@gmail.com](mailto:southandy@gmail.com)

[andysouth.co.uk](http://andysouth.co.uk)

LondonR, December 3<sup>rd</sup> 2013

# rworldmap

- Released 2010
- Aims *improve visualisation of global data*  
*easy to start, flexible to extend*  
*publication quality outputs*  
*aid interdisciplinary work*
- 4842 downloads via RStudio in 2013 (so far...)
- Contains up-to-date country boundaries
- RJournal article

## rworldmap: A New R package for Mapping Global Data

by *Andy South*

**Abstract** **rworldmap** is a relatively new package available on CRAN for the mapping and visualisation of global data. The vision is to make the display of global data easier, to facilitate understanding and communication. The initial focus is on data referenced by country or grid due to the frequency of use of such data in global assessments. Tools to link data referenced by country (either name or code) to a map, and then to display the map are provided as are functions to map global gridded data. Country and gridded functions accept the same arguments to specify the nature of categories and colour and how legends are formatted. This package builds on the functionality of existing packages, particularly **sp**, **maptools** and **fields**. Example code is provided to produce maps, to link with the packages **classInt**, **RColorBrewer** and **ncdf**, and to plot examples of publicly available country and gridded data.

There appears to be a gap in the market for free software tools that can be used across disciplinary boundaries to produce innovative, publication quality global visualisations. Within R there are great building blocks (particularly **sp**, **maptools** and **fields**) for spatial data but users previously had to go through a number of steps if they wanted to produce world maps of their own data. Experience has shown that difficulties with linking data and creating classifications, colour schemes and legends, currently constrains researchers' ability to view and display global data. We aim to reduce that constraint to allow researchers to spend more time on the more important issue of what they want to display. The vision for **rworldmap** is to produce a package to facilitate the visualisation and mapping of global data. Because the focus is on global data, the package can be more specialised than existing packages, making world mapping easier, partly because it doesn't have to deal with detailed local maps. Through **rworldmap** we aim to make it easy for R users to explore their global data and also to produce publication quality figures from their outputs.

# sp : what rworldmap is built on

- spatial classes
- Spatial Polygons Dataframe (sPDF)



Sovereign country	Algeria	DZA
Sovereign country	Ecuador	ECU
Sovereign country	Egypt	EGY
Sovereign country	Eritrea	ERI
Sovereign country	Spain	ESP
Sovereign country	Estonia	EST
Sovereign country	Ethiopia	ETH
County	Finland	FIN
Sovereign country	Fiji	FJI
Dependency	Falkland Islands	FLK
Country	France	FRA

# sp : what rworldmap is built on

- spatial classes
- Spatial Polygons Dataframe (sPDF)



Sovereign country	Algeria	DZA
Sovereign country	Ecuador	ECU
Sovereign country	Egypt	EGY
Sovereign country	Eritrea	ERI
Sovereign country	Spain	ESP
Sovereign country	Estonia	EST
Sovereign country	Ethiopia	ETH
County	Finland	FIN
Sovereign country	Fiji	FJI
Dependency	Falkland Islands	FLK
Country	France	FRA

# maps in rworldmap are spatial polygons dataframes

```
sPDF <- getMap()
```

```
class(sPDF)
```

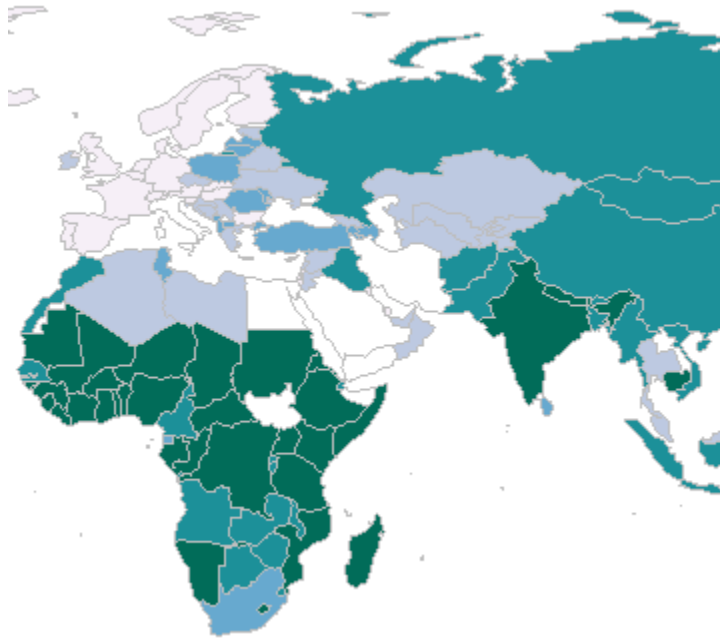
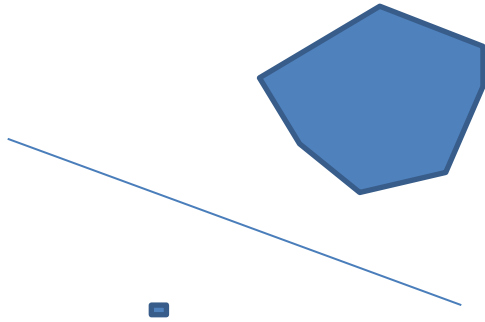
```
plot(sPDF)
```

```
#to get at the dataframe
```

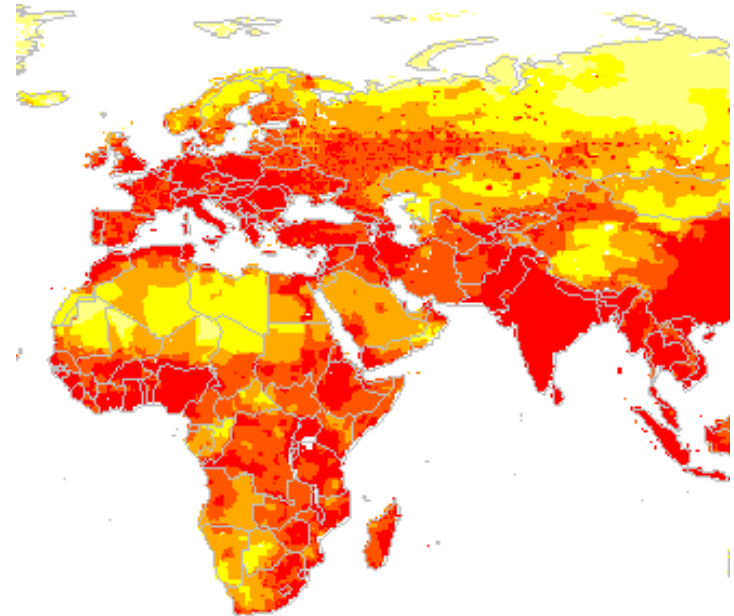
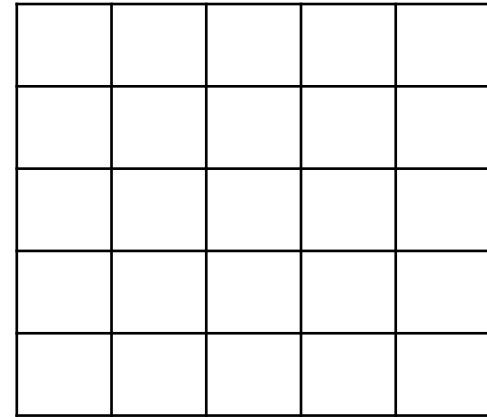
```
dF <- sPDF@data
```

```
str(dF)
```

# vector



# raster



# Joining 1

Your data

country	temperature
UK	cold
Spain	hot

Map data

Algeria	DZA
Ecuador	ECU
Egypt	EGY
Eritrea	ERI
Spain	ESP
Estonia	EST
Ethiopia	ETH
Finland	FIN
Fiji	FJI
Falkland Islands	FLK
France	FRA



Columns in common



# Joining 2

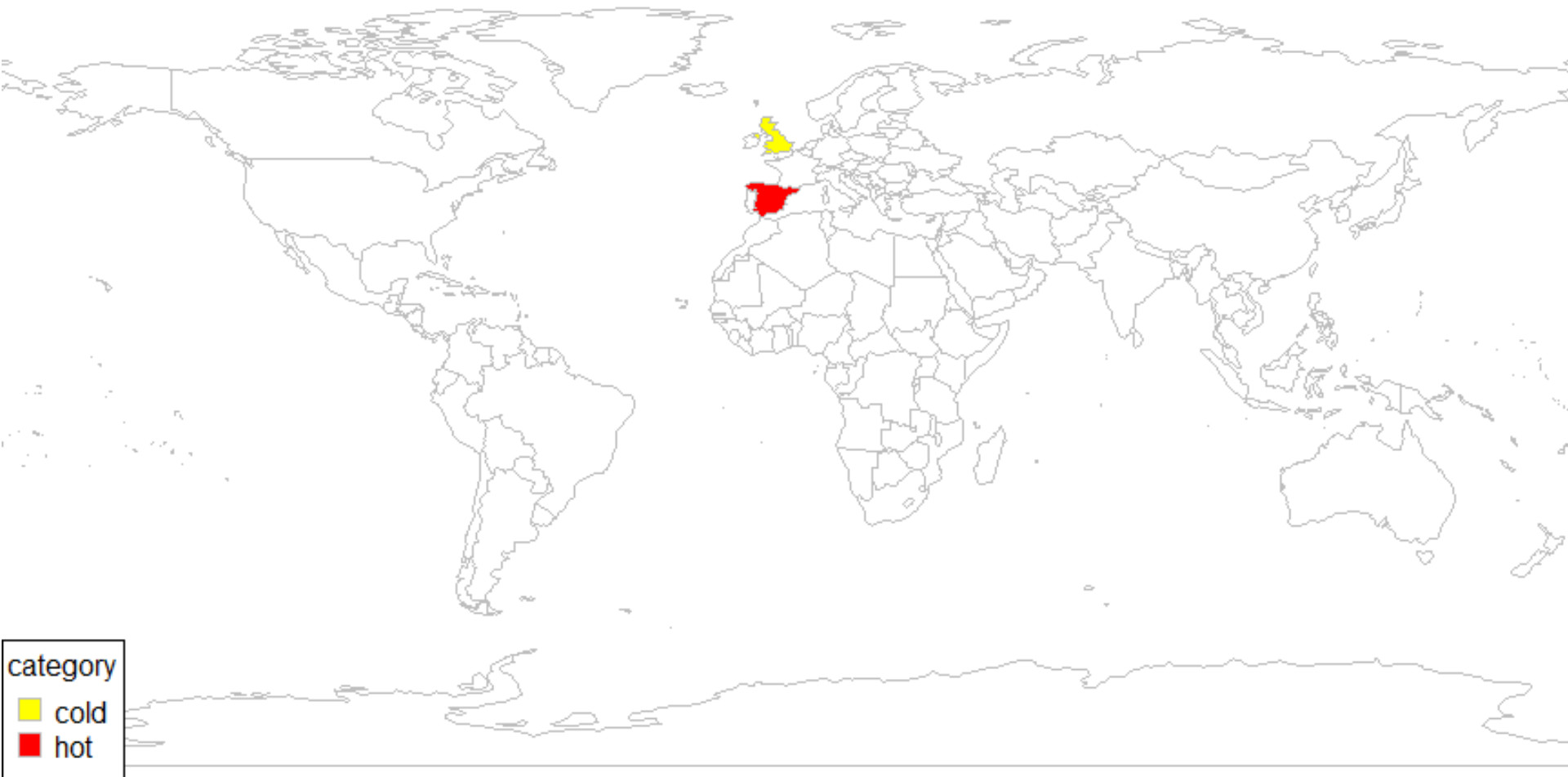
```
dF <- data.frame( country=c("spain", "uk")  
                  , weather=c("hot", "cold") )
```

```
sPDF <-
```

```
joinCountryData2Map( dF  
                     ,joinCode = "NAME"  
                     ,nameJoinColumn = "country")
```

```
mapCountryData(sPDF,nameColumnToPlot='weather')
```

## weather



Note that these data are categorical & rworldmap copes.

# HOWEVER ...

You should join using country codes instead of names if at all possible.

# LONDON2012

[HOME](#)[PARALYMPICS](#)[SCHEDULE](#)[MEDALS](#)[SPORT GUIDES](#)[PICTURE GALLERIES](#)[VENUE](#)

**HOT TOPICS:** [Paralympic magic moments](#) | [Paralympic medal map](#) | [Classifications explained](#) | [50 best moments](#)

## London 2012 Olympics: North Koreans furious as organisers blunder by displaying South Korean flag

The opening day of competition in the 2012 Games ended with the first diplomatic incident of the Games as organisers were forced to apologise to North Korea for displaying the flag of bitter enemies South Korea before their women's football match against Colombia.



- North Korea

Korea, Democratic People's Republic of  
Democratic People's Republic of Korea  
Korea, Dem. Rep.

- South Korea

Korea, Republic of  
Republic of Korea  
Korea



# Country Codes - ISO 3166

## What is ISO 3166?

ISO 3166 is the International Standard for country codes and codes for their subdivisions. The purpose of ISO 3166 is to establish internationally recognised codes for the representation of names of countries, territories or areas of geographical interest, and their subdivisions. However, ISO 3166 does not establish the names of countries, only the codes that represent them.

The country names in ISO 3166 come from United Nations sources. New names and codes are added automatically when the United Nations publishes new names in either the Terminology Bulletin Country Names or in the Country and Region Codes for Statistical Use maintained by the United Nations Statistics Divisions. Names for subdivisions are taken from relevant official national information sources.

ISO 3166 was first published in 1974 as a single standard to establish the country codes. It was expanded into three parts in 1997 to include the codes for subdivisions and the codes for names of countries that are no longer in use. Of the three parts, Part 1, ISO 3166-1 is generally used the most often.

# Joining data using country codes

```
#example data in rworldmap
```

```
data(countryExData)
```

```
sPDF <- joinCountryData2Map(countryExData
```

```
  , joinCode = "ISO3"
```

```
  , nameJoinColumn = "ISO3V10")
```

```
mapParams <- mapCountryData(sPDF
```

```
  , nameColumnToPlot="EPI")
```

# Examples with world bank indicators

```
library(WDI)
```

```
# http://data.worldbank.org/indicator/  
# Improved sanitation facilities (% of popn)
```

```
indicator <- "SH.STA.ACSN"
```

```
dFsanitation <- WDI( indicator=indicator  
                    , start=2005  
                    , end=2005 )
```



# i. choropleth map

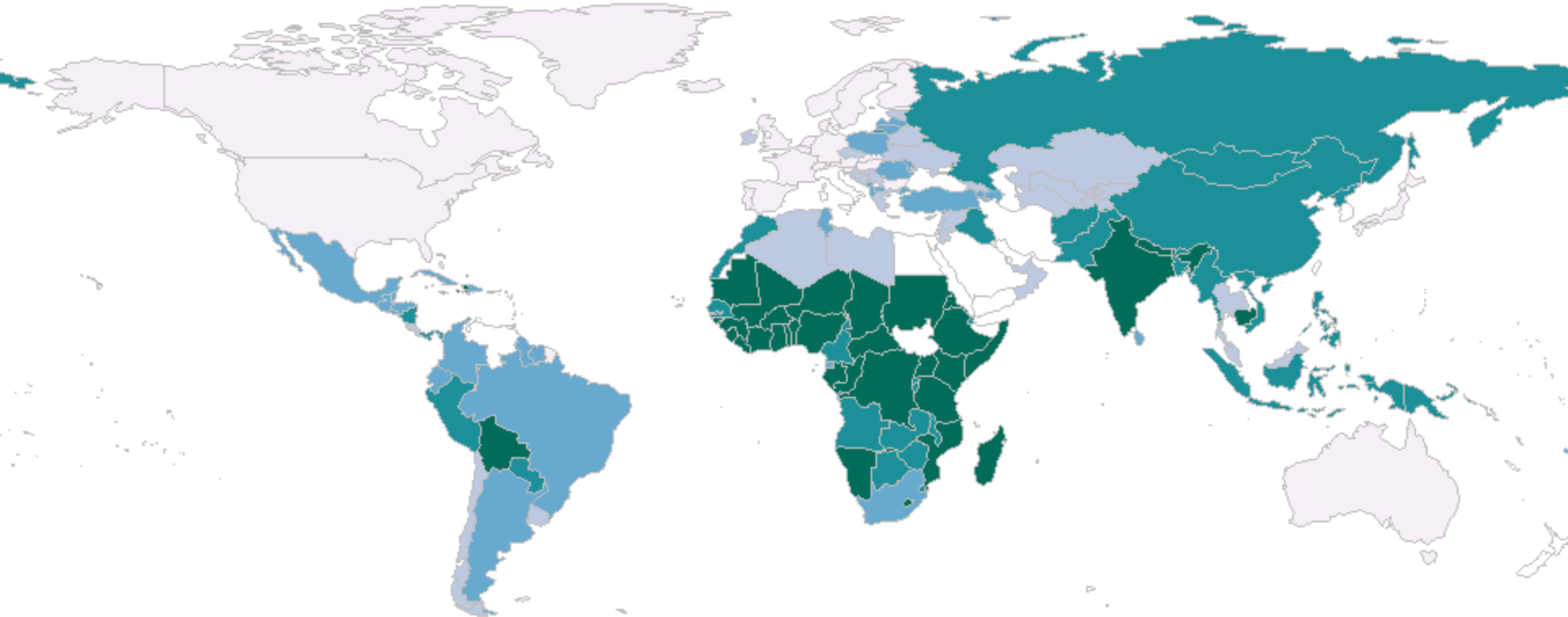
```
library(rworldmap)

sPDFsanitation <-
joinCountryData2Map( dFsanitation
                     , nameJoinColumn='iso2c'
                     , joinCode='ISO2')

library(RColorBrewer)
numCats <- 5
colourPalette <- rev(brewer.pal(numCats, "PuBuGn"))

mapCountryData( sPDFsanitation
                , nameColumnToPlot=indicator
                , colourPalette=colourPalette
                , numCats=numCats )
```

# Access to Sanitation 2005



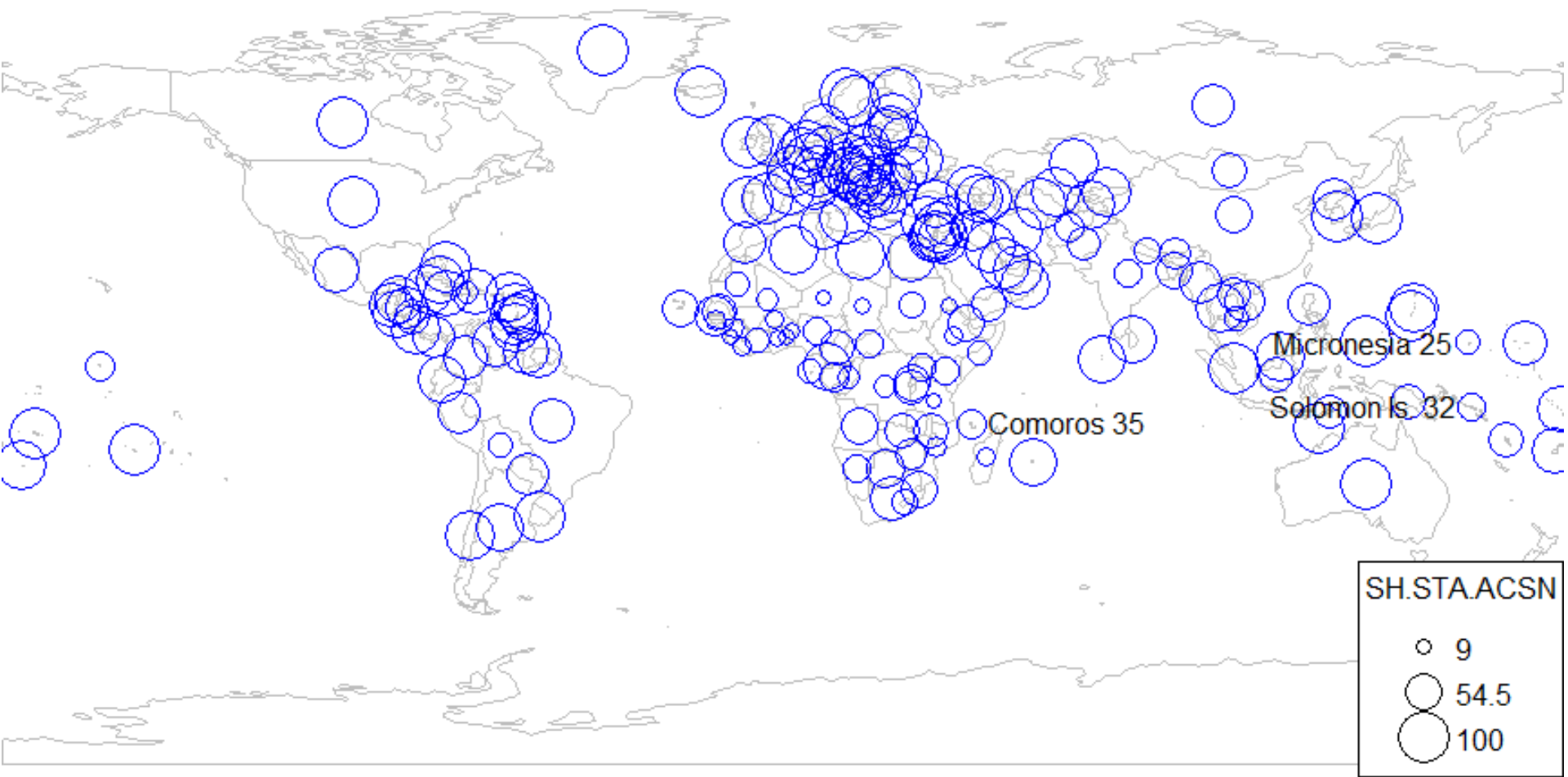
## ii. bubble map

```
library(rworldmap)
```

```
mapBubbles( sPDFsanitation  
            , nameZSize = indicator  
            , nameZColour = 'blue'  
            , fill = FALSE )
```

#optional interactive labelling of countries

```
identifyCountries( sPDFsanitation  
                  , nameColumnToPlot = indicator )
```

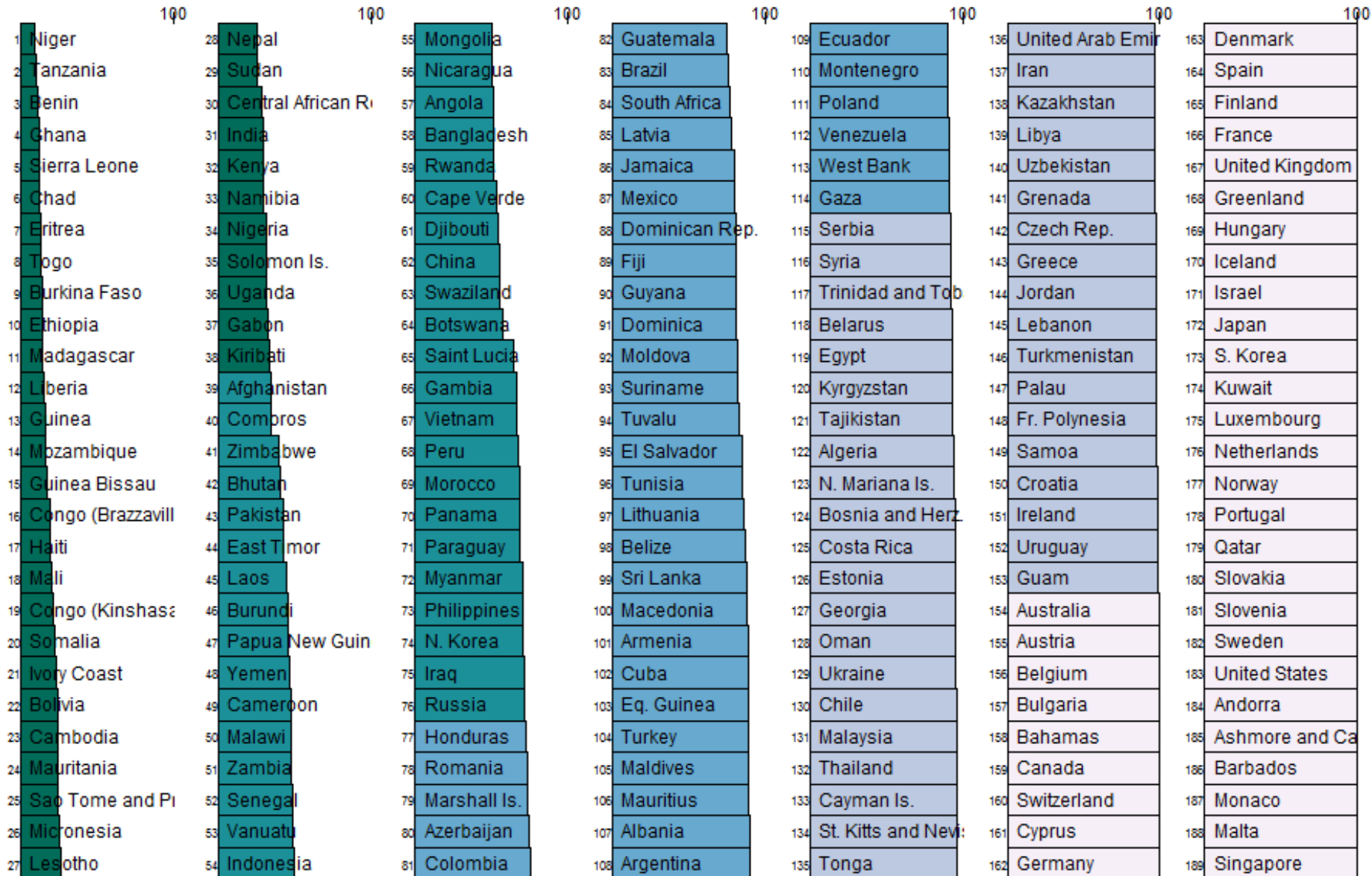


### iii. Ranked bar plot

```
library(rworldmap)
```

```
barplotCountryData( sPDFsanitation  
  , nameColumnToPlot=indicator  
  , nameCountryColumn = "NAME"  
  , numCats = numCats  
  , colourPalette = colourPalette  
  , na.last = NA  
  , decreasing = FALSE  
  , scaleSameInPanels = TRUE  
  , numPanels = 7  
  , cex = 1.1 )
```

# Access to sanitation



## non world maps

```
## US states map downloaded from :
```

```
## http://www2.census.gov/cgi-bin/shapefiles2009/national-files
```

```
inFile <- 'tl_2009_us_state.shp'
```

```
sPDF <- readShapePoly(inFile)
```

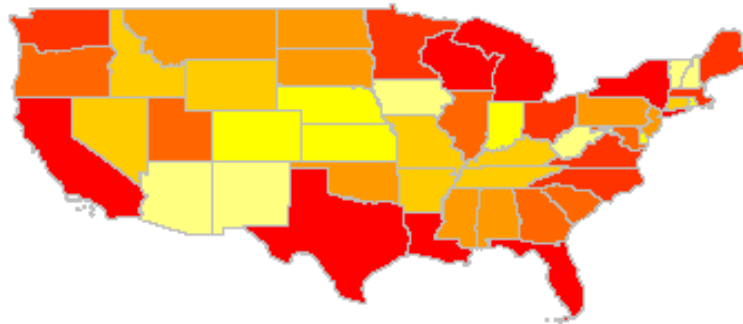
```
str(sPDF@data)
```

```
## use mapPolys to map the sPDF
```

```
mapPolys(sPDF, nameColumnToPlot =  
"AWATER", mapRegion='North America')
```

# non world maps

**AWATER**



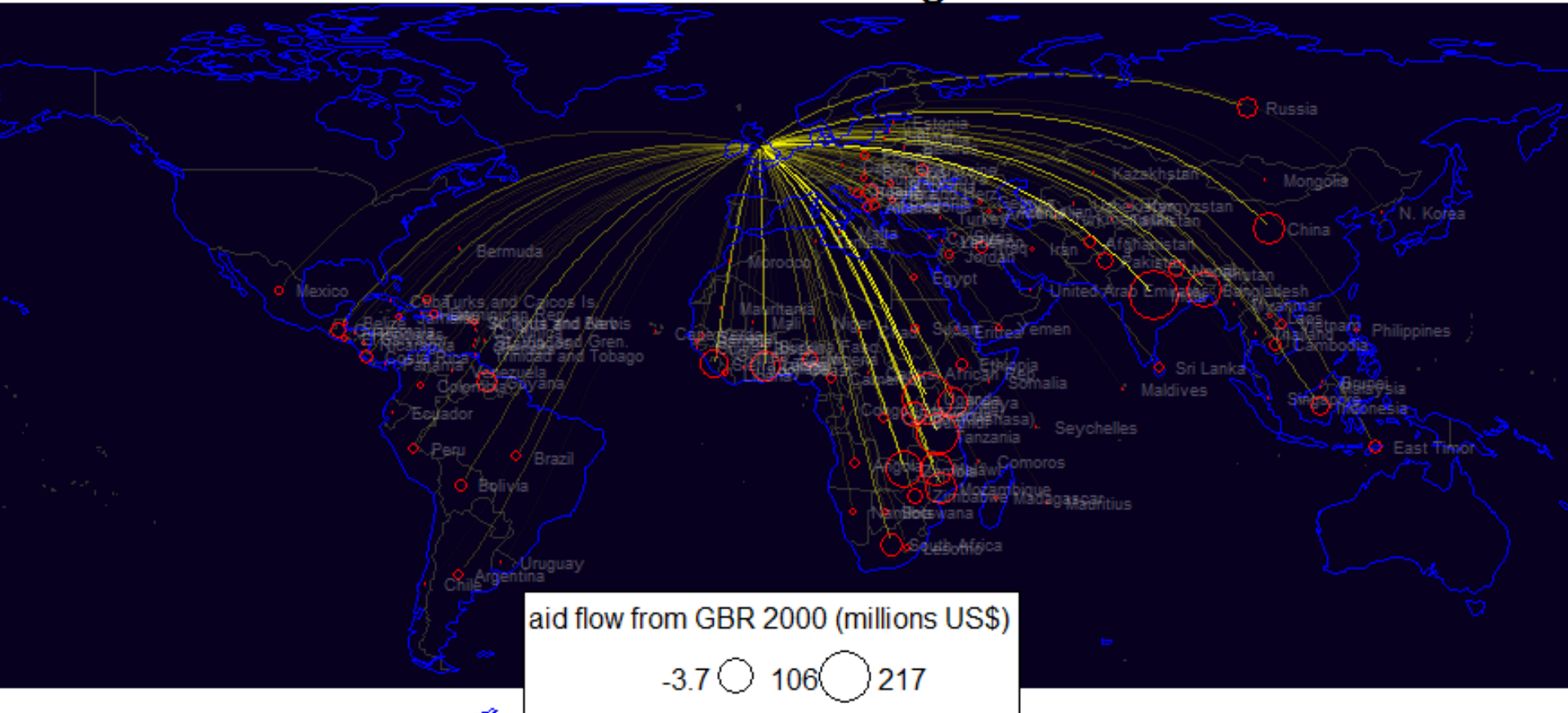
18880000

2.45e+11

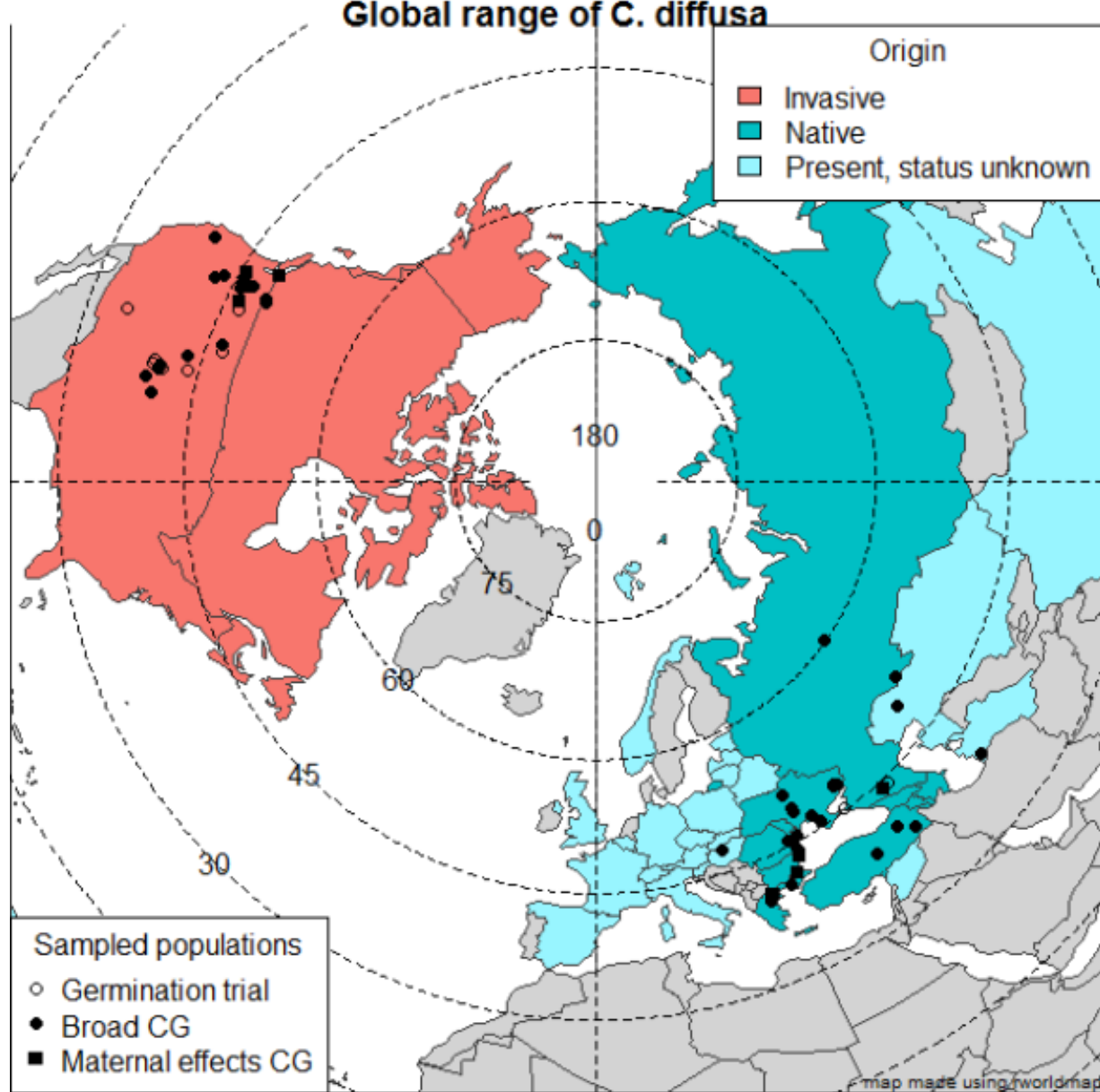


- These are the basics of rworldmap
- Add bits together to make more interesting plots ...

# Aid flow from United Kingdom 2000



stackoverflow



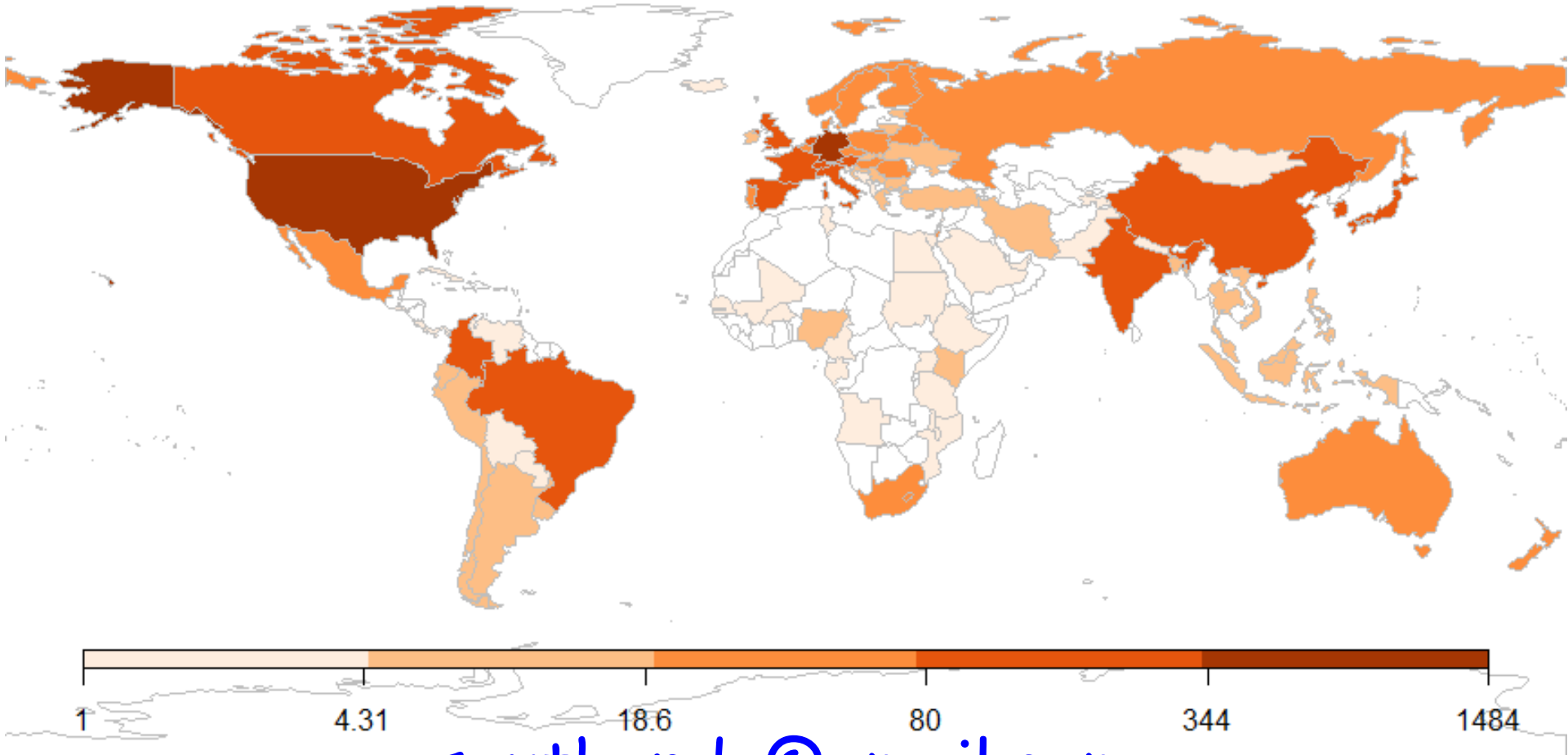
<https://alienplantation.wordpress.com/2013/06/18/nuts-and-bolts-modern-maps-of-eurasia-in-r/><sup>27</sup>

# Things to remember maybe

- What a spatial polygons dataframe is
- Joining country data to maps
- Use country codes
- Thanks to the developers of the packages on which rworldmap is built (sp, classInt, ...)

# Meta-mapping (using rworldmap to map the downloads of ...)

rworldmap downloads in 2013



[southandy@gmail.com](mailto:southandy@gmail.com)

andysouth.co.uk

Xtra bits after here ...

# modifying map legends

```
data(countryExData)
sPDF <- joinCountryData2Map(countryExData
  , joinCode = "ISO3"
  , nameJoinColumn = "ISO3V10")
```

```
mapParams <- mapCountryData(sPDF
  , nameColumnToPlot="EPI"
  , addLegend=FALSE )
```

```
do.call( addMapLegend
  , c(mapParams
  , legendLabels="all"
  , legendWidth=0.5 ))
```

# adding multiple elements to a plot

```
mapCountryData()
```

```
mapBubbles( getMap()  
            , nameZSize="GDP_MD_EST"  
            , nameZColour="black"  
            , add=TRUE )
```



# interactive identification & labelling

```
mapCountryData(nameColumnToPlot="POP_EST")  
identifyCountries(nameColumnToPlot="POP_EST")
```

Then click close to the centre of countries to bring up country labels.

# Resources

<http://andysouth.co.uk/softwareworldmap/>

<https://groups.google.com/group/rworldmap>

## rworldmap downloads in 2013

