

CREATING MAPS

Jan-Philipp Kolb

25 Februar, 2020

THE PACKAGE TMAP

library(tmap)

cartography

earthquake

leaflet

bubble

legend

raster

shape

crimes

animated

plot

lines

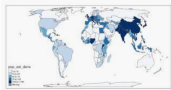
e walker

london

thematic

mapping

interactive



SpatialControl + tmap R package
spatcontrol.net



Mapping to a Y(map) | R-bloggers
r-bloggers.com



Sharon Machlis on Twitter: "New package tm..."
twitter.com



Packages to simplify mapping in R (Revolution Analytics)
blog.revolutionanalytics.com



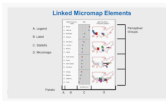
r - How to overlay point co-ordinate...
gis.stackexchange.com



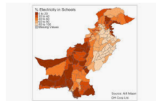
cartography package | R Documentation
rdocumentation.org



Using R to Visualize Spatial Data: R as GIS ...
slideshare.net



SFS GIS R | Micromap And Tmap
ryan-hill.github.io



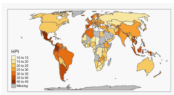
r - How to overlay point co-ordinates on tmap sh...
gis.stackexchange.com



GIS with R
patkilo.github.io



R tmap, legend formatting - displ...
stackoverflow.com



tmap: get started!
cran.r-project.org



Exploring and presenting maps with "tmap" | user...
channel5.mdn.com



mtennes/tmap R package for thematic maps b...
devhub.io



Data Viz 2: Mapping Areas and Context
alex.singelton.com

THE WORLD DATASET

NATURAL EARTH

- Dataset contains information from **Natural Earth**

`data(World)`



Natural Earth

Free vector and raster map data at
1:10m, 1:50m, and 1:110m scales

Search

Home

Features

Downloads

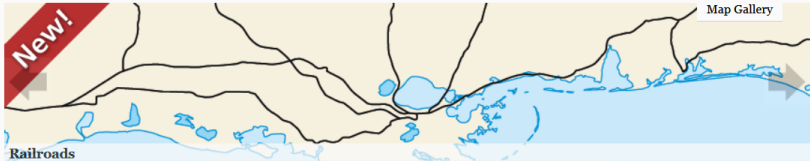
Blog

Forums

Corrections

About

Map Gallery



Natural Earth is a public domain map dataset available at 1:10m, 1:50m, and 1:110 million scales. Featuring tightly integrated vector and raster data, with Natural Earth you can make a variety of visually pleasing, well-crafted maps with cartography or GIS software.

THE QTM COMMAND FROM THE TMAP PACKAGE

FAST THEMATIC MAP

- With the command **qtm** you can create a fast thematic map
- Example from the **Vignette** for the tmap package

```
qtm(World)
```



THE WORLD-DATASET

The World Dataset in Package tmap

RPubs brought to you by Ethical

Show 10 entries

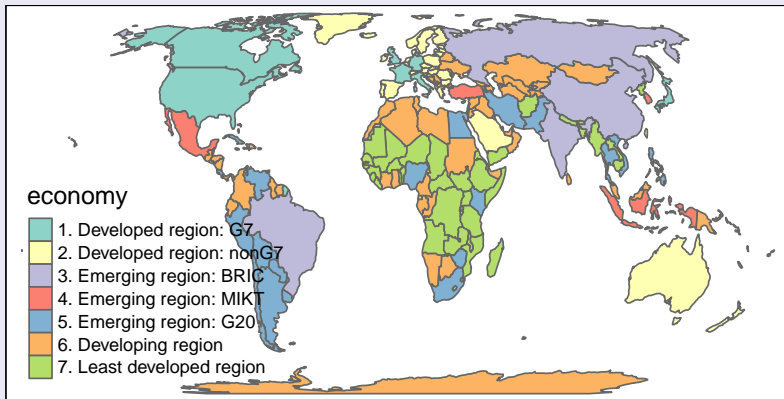
Search:

	iso_a3	name	sovereign	continent	subregion	area	pop_est	pop_est_dens	gdp_md_est	gdp_cap_est	economy	income_grp	life_exp	well_being	HPI
2	AFG	Afghanistan	Afghanistan	Asia	Southern Asia	652860	28400000	43.5009037159575	22270	784.154929577465	7. Least developed region	5. Low income	48.7	4.75838085759722	36.753657778004
3	AGO	Angola	Angola	Africa	Middle Africa	1246700	12799293	10.2665380604797	110300	8617.6634912569	7. Least developed region	3. Upper middle income	51.1	4.20609164016618	33.201432044336
5	ALB	Albania	Albania	Europe	Southern Europe	27400	3639453	132.826751824818	21810	5992.65878691111	6. Developing region	4. Lower middle income	76.9	5.26893660419411	54.051180370208
8	ARE	United Arab Emirates	United Arab Emirates	Asia	Western Asia	83600	4798491	57.3982177033493	184300	38407.907819554	6. Developing region	2. High income: nonOECD	76.5	7.1968030933638	31.778274185231
9	ARG	Argentina	Argentina	South America	South America	2736690	40913584	14.9500250302373	573900	14027.1260518267	5. Emerging region: G20	3. Upper middle income	75.9	6.44106720496824	54.0550416711541
10	ARM	Armenia	Armenia	Asia	Western Asia	28470	2967004	104.215103617843	18770	6326.24694809983	6. Developing region	4. Lower middle income	74.2	4.36781129220333	46.0031857989857
12	ATA	Antarctica	Antarctica	Antarctica	Antarctica	10866664.4069415	3802	0.000349877373370556			6. Developing region	2. High income: nonOECD			
14	ATF	Fr. S. Antarctic Lands	France	Seven seas (open ocean)	Seven seas (open ocean)	6187.20529564552	140	0.0226273403435523	16	114285.714285714	6. Developing region	2. High income: nonOECD			
16	AUS	Australia	Australia	Oceania	Australia and New Zealand	7682300	21262641	2.76774416515887	800200	37634.0831790369	2. Developed region: nonG7	1. High income: OECD	81.9	7.40561614869191	41.9798119494163
17	AUT	Austria	Austria	Europe	Western Europe	82409	8210281	99.6284507760075	329500	40132.6093467446	2. Developed region: nonG7	1. High income: OECD	80.9	7.34603595780621	47.0851352018778

TO GET MORE COLOR IN THE MAP

ECONOMIC DEVELOPMENT STATUS

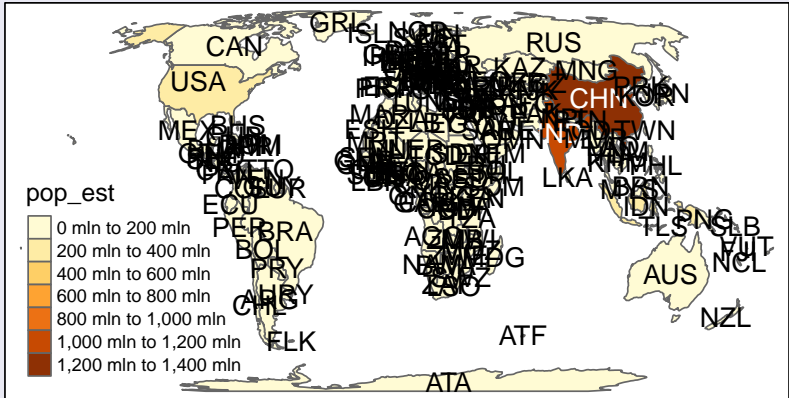
```
qtm(World, fill="economy")
```



A MAP WITH TEXT

POPULATION

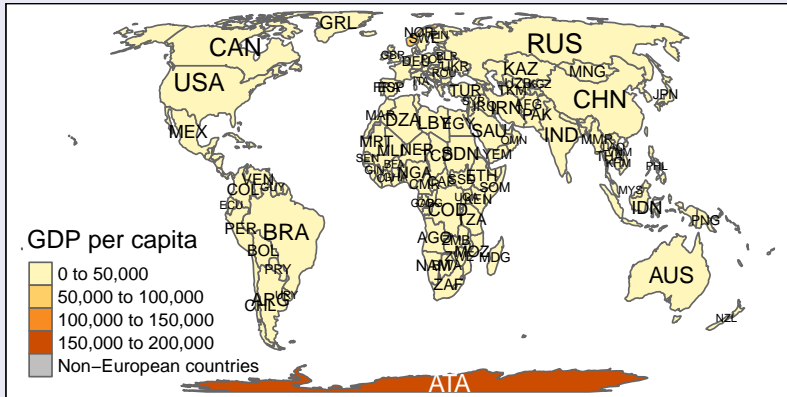
```
qtm(World, fill="pop_est", text="iso_a3")
```



THIS SCHEME IS BETTER:

GDP

```
qtm(World, fill="gdp_cap_est", text="iso_a3",  
    text.size="AREA", root=5, fill.title="GDP per capita",  
    fill.textNA="Non-European countries", theme="Europe")
```



TOPICS OF THE WORLD DATASET

AVAILABLE VARIABLES IN THE DATA SET

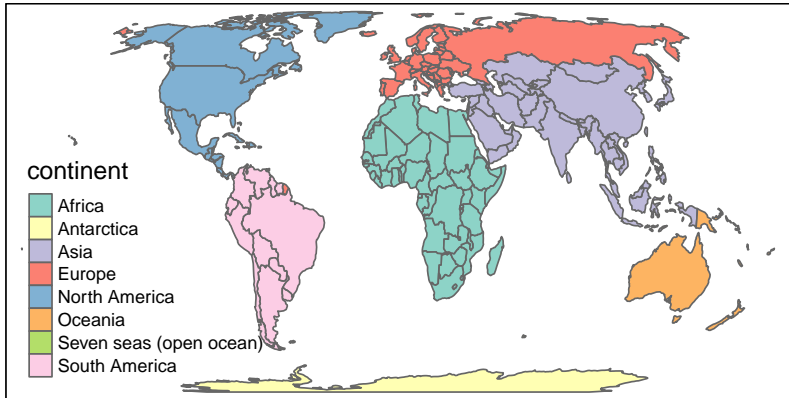
- **ISO classification**
- country name
- Area, population, population density,
- **Gross Domestic Product**
- Gross domestic product **at purchasing power parities**
- Economy, income group

THE WORLD DATASET - VARIABLES AND WHAT'S BEHIND

iso_a3	name	sovereign	continent
AFG	Afghanistan	Afghanistan	Asia
AGO	Angola	Angola	Africa
ALB	Albania	Albania	Europe
ARE	United Arab Emirates	United Arab Emirates	Asia
ARG	Argentina	Argentina	South America
ARM	Armenia	Armenia	Asia
ATA	Antarctica	Antarctica	Antarctica
ATF	Fr. S. Antarctic Lands	France	Seven seas (open oce

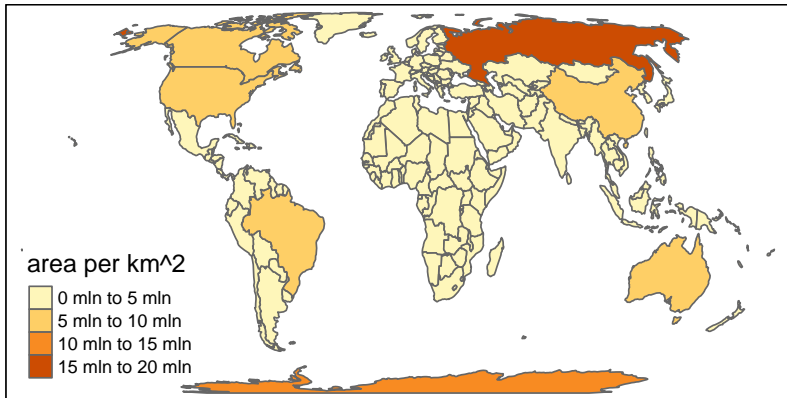
THE VARIABLE CONTINENT

```
qtm(World, fill="continent")
```



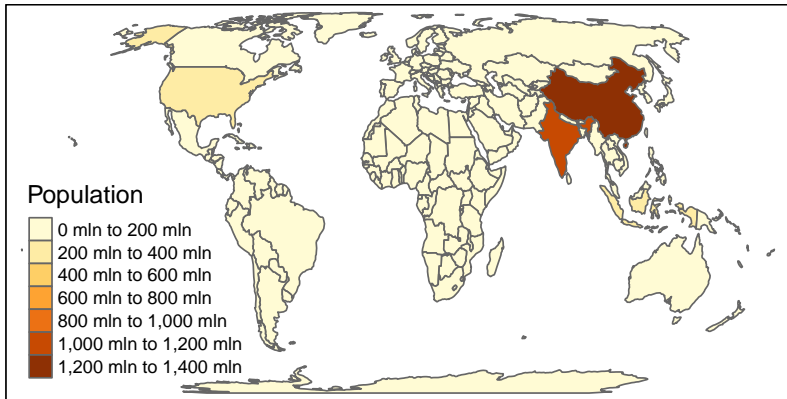
THE VARIABLE AREA

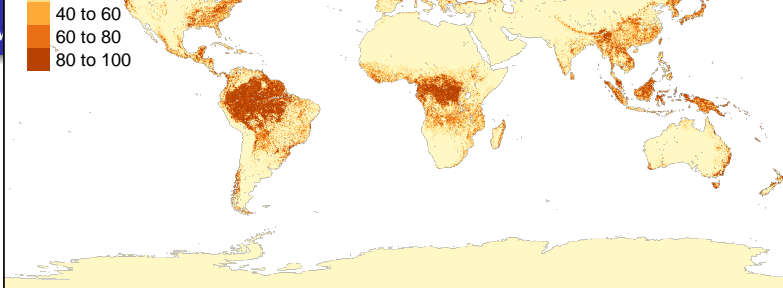
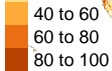
```
qtm(World, fill="area") # Russia is huge
```



POPULATION

```
qtm(World, fill="pop_est", fill.title="Population")
```

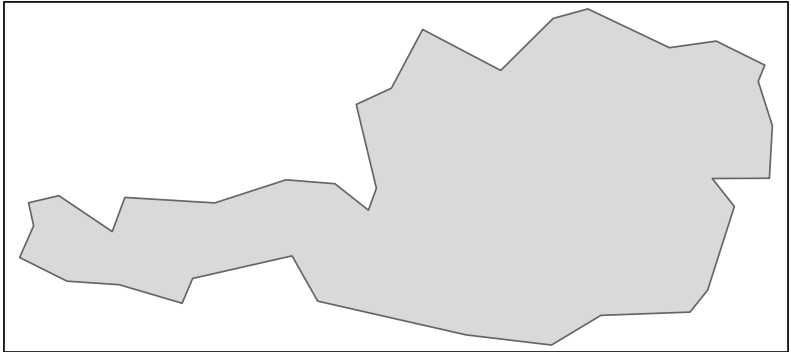




POPULATION AND LEVEL OF DEVELOPMENT

VISUALIZE ONLY ONE COUNTRY

```
tm_shape(World[World$name=="Austria", ]) +  
  tm_polygons()
```



LOAD EXAMPLE DATA

DATA SOURCE EUROSTAT

- Data about unemployment in Europe

```
url <- "https://raw.githubusercontent.com/Japhilko/GeoData/master"
```

```
Unemp <- read.csv(url)
```


AN OVERVIEW OF THE DATA

X	GEO	Val2007M12	Val2013M01
9316	EU28	6.9	10.9
9325	EU27	6.9	10.9
9334	EU25	6.9	11.0
9343	EU15	6.9	11.1
9352	EA	7.3	12.0
9361	EA19	7.3	12.0
9370	EA18	7.4	12.0
9379	EA17	7.4	12.0
9388	EA16	7.4	12.0
9397	EA15	7.3	12.0

EXCURSUS: THE COMMAND MATCH

CREATE TWO EXAMPLE VECTORS

```
vec_a <- c("A",2,6,1,"C")  
vec_b <- c(1,"C",2)
```

BRINGING THE TWO VECTORS TOGETHER

- With the function `match` you can see which element of the first vector matches the second vector.

```
match(vec_a,vec_b)
```

```
## [1] NA  3 NA  1  2
```

USE THE PACKAGE TMAP WITH YOUR DATA

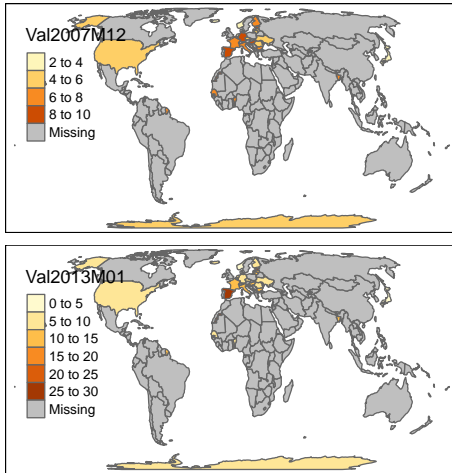
```
library("tmap")
```

MATCH THE DATA

```
iso_a2<- substr(World$iso_a3,1,2)  
ind <- match(iso_a2,Unemp$GEO)  
World$Val2007M12 <- Unemp$Val2007M12[ind]  
World$Val2013M01 <- Unemp$Val2013M01[ind]
```

PLOT A MAP

```
qtm(World,c("Val2007M12","Val2013M01"))
```



EXERCISE: VISUALISATION OF EUROSTAT DATA

FIRST PART - PLOT A MAP

- Download and import the data `unemprate_by_sex.csv` from ILIAS.
- Link the data with map data .
- Visualise the linked data in a map.

IF YOU HAVE THAT:

- Search for example **here** for datasets containing the country name and visualize the data with `tmap`.

THE FIRST LAW OF GEOGRAPHY (TFLG)

“All things are related, but nearby things are more related than distant things” [Tobler, 1970]

A MAP OF AFRICA

```
library(maptools)
data(wrld_simpl)
Africa <- wrld_simpl[wrld_simpl@data$REGION==2,]
plot(Africa)
```



THE CENTER OF A POLYGON

```
library(sp)
Af <- coordinates(Africa)
plot(Africa)
points(x=Af[1,1],y=Af[1,2],col="red",pch=20)
```



FIND THE NEAREST NEIGHBOURS

```
library(spdep)  
Af_nb <- tri2nb(Af)
```

Neighbours for the first country:

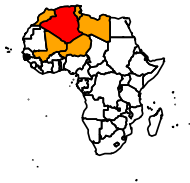
```
Af_nb[1]
```

```
## [[1]]
```

```
## [1] 24 26 27 32 48
```

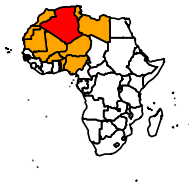
FIND THE NEIGHBOURS

```
plot(Africa)
plot(Africa[1,],col="red",add=T)
plot(Africa[Af_nb[1][[1]],],col="orange",add=T)
```



FIND TEN NEXT NEIGHBOURS

```
IDs <- row.names(as(Africa, "data.frame"))
Af10_nb <- knn2nb(knearneigh(Af, k = 10), row.names = IDs)
plot(Africa)
plot(Africa[1,], col="red", add=T)
plot(Africa[Af10_nb[1][[1]],], col="orange", add=T)
```



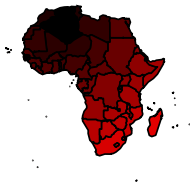
COMPUTE THE DISTANCE

```
Af <- coordinates(Africa) # get centroid  
library(raster)  
pointDistance(Af[1:4,], lonlat=TRUE) # compute distance
```

##	[,1]	[,2]	[,3]	[,4]
## [1,]	0	NA	NA	NA
## [2,]	4763231	0	NA	NA
## [3,]	2055609	2954497	0	NA
## [4,]	3484053	1295173	1839191	0

CALCULATE/DRAW A DISTANCE MATRIX

```
Dist_Af <- pointDistance(Af, lonlat=TRUE)
Af_color <- Dist_Af[,1]
Af_color <- Af_color/max(Af_color)
Af_color <- rgb(Af_color,0,0)
plot(Africa,col=Af_color)
```



- Raster, CMSAF and solaR

<https://procomun.wordpress.com/2011/06/17/raster-cmsaf-and-solar/>

- Getting rasters into shape from R

<https://johnbaumgartner.wordpress.com/2012/07/26/getting-rasters-into-shape-from-r/>