maps package

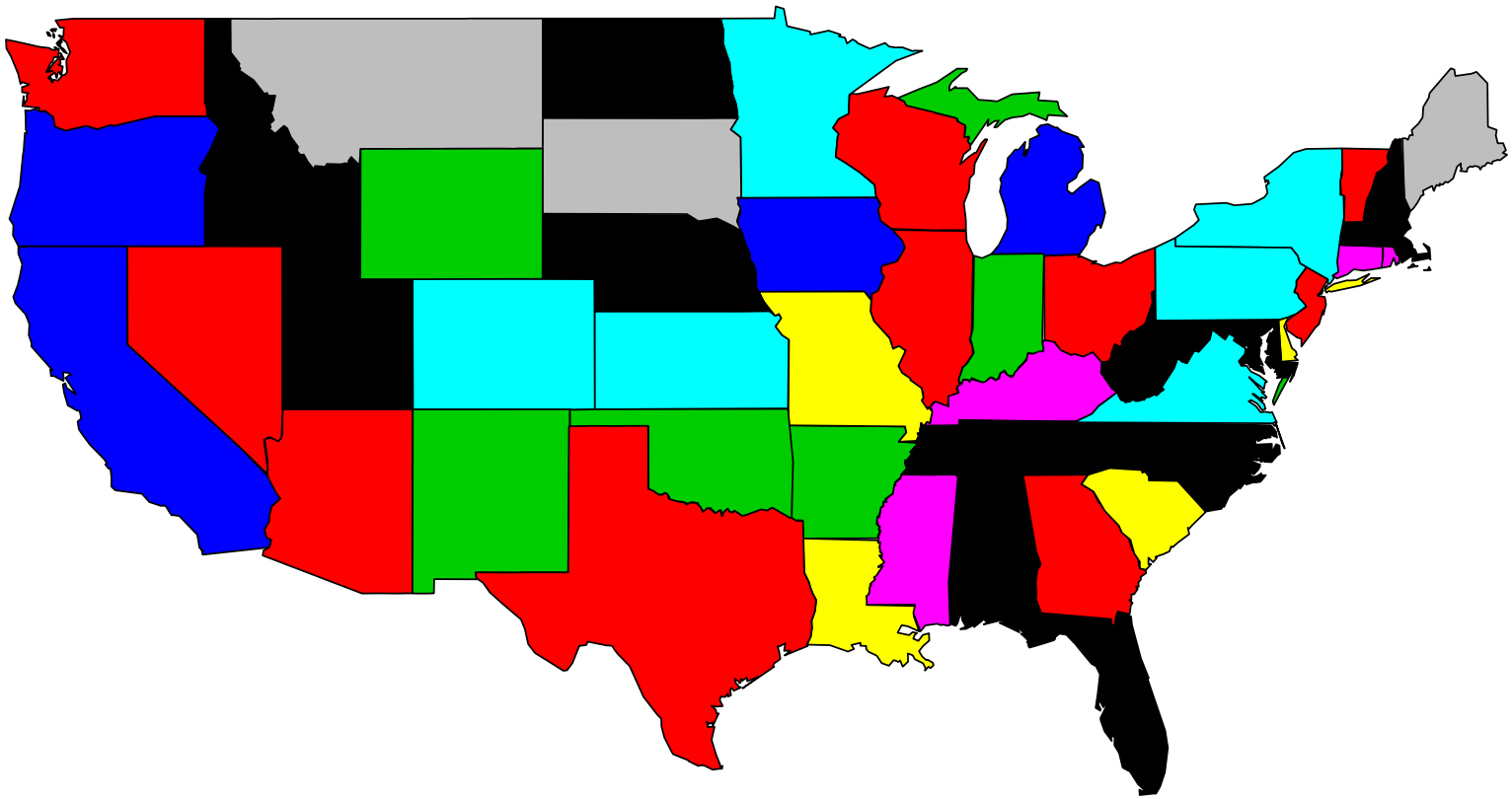
Jan-Philipp Kolb

Fri Sep 18 14:22:39 2015

Political Map:

- A political map does not show any topographic features.
- It focuses on the state and national boundaries of a place.
- Include locations of cities - both large and small, depending on the detail of the map.
- A common type of political map would be one showing the 50 U.S. states and their borders along with the United States' north and south international borders (map of the United States).

Political Map



The maps package

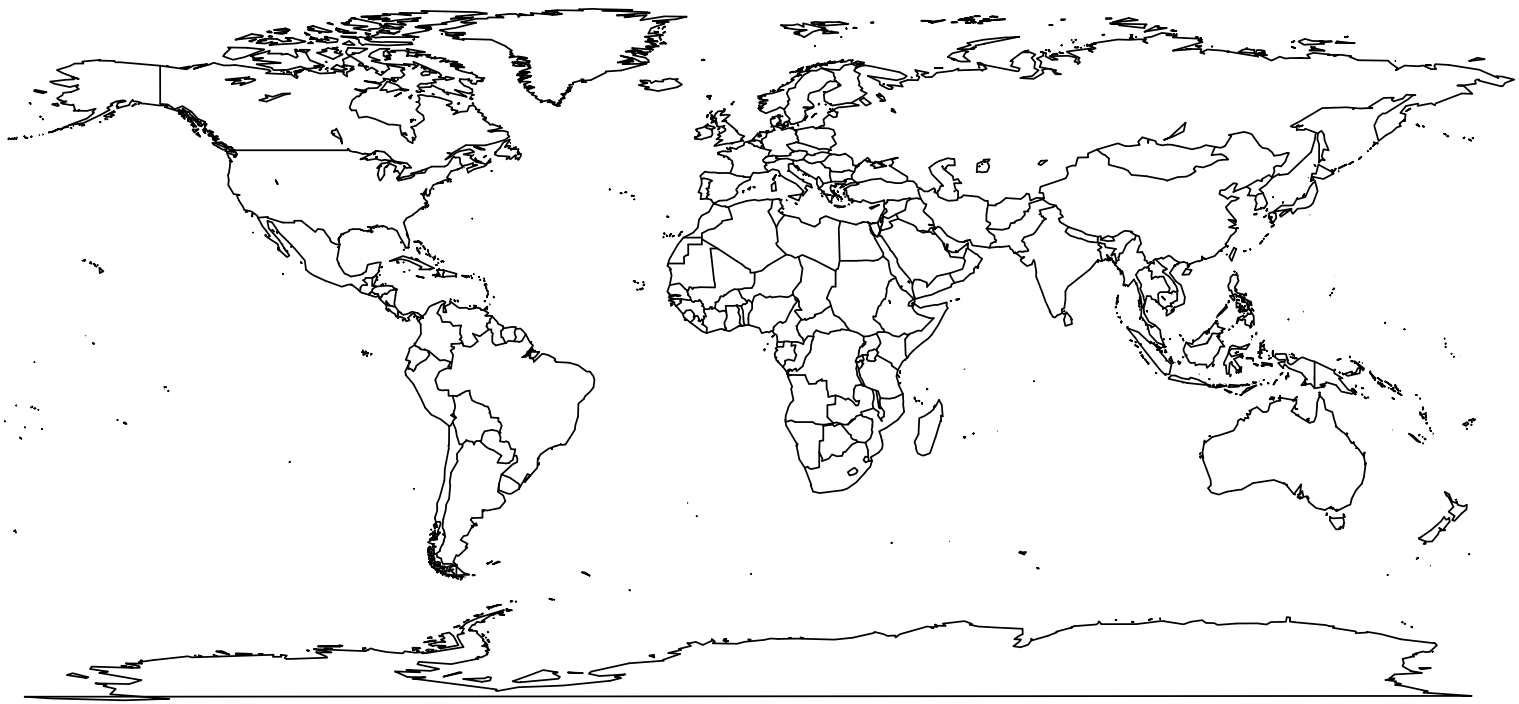
```
library(maps)
```

How to get help

```
?map
```

A first map

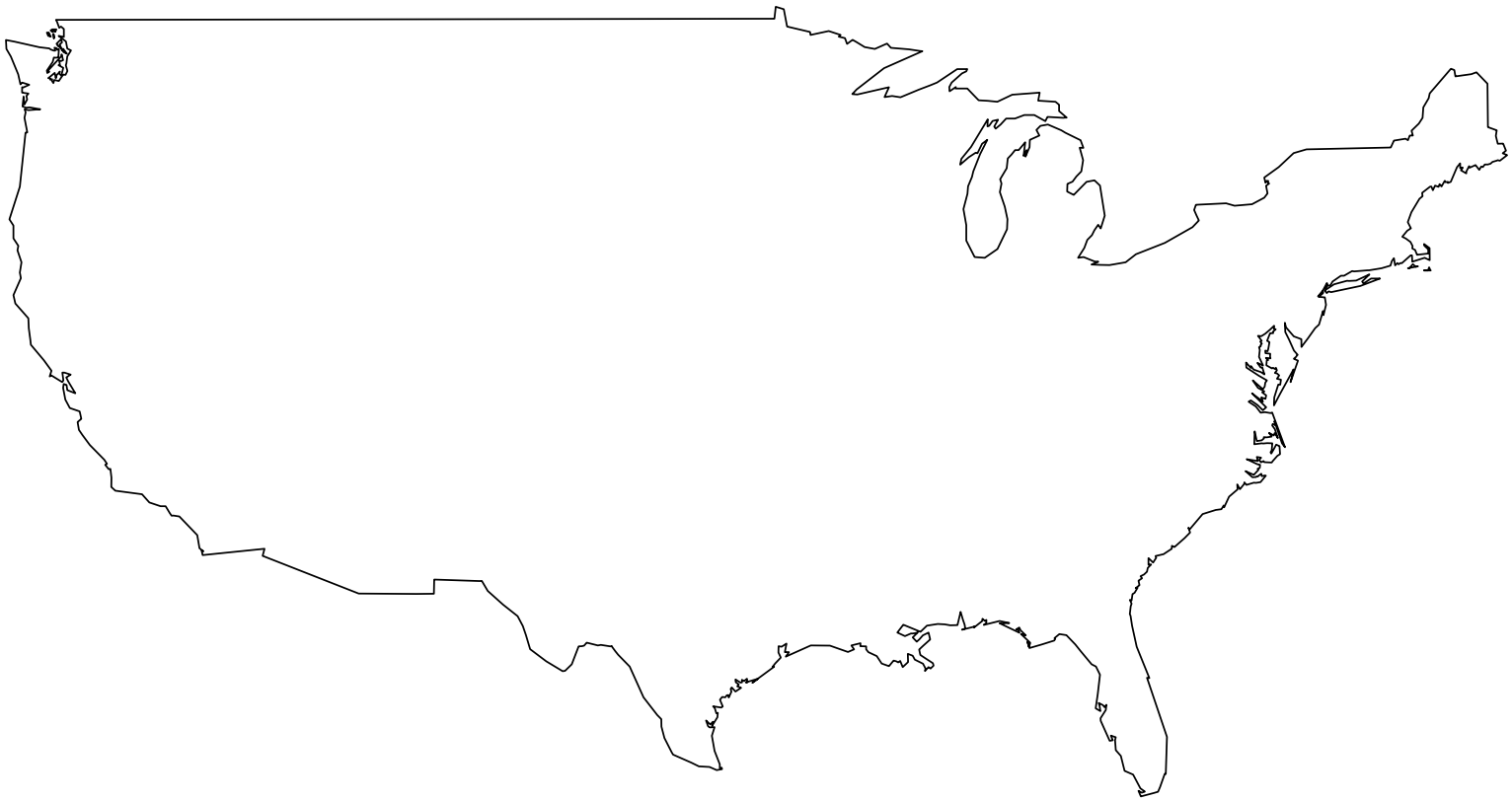
```
map()
```



A first argument

The same command with an argument:

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



A map of france

```
map("france")
```

Choropleths - package maps

Get the borders in blue color:

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

Choropleths - package maps

If we want the areas in blue:

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

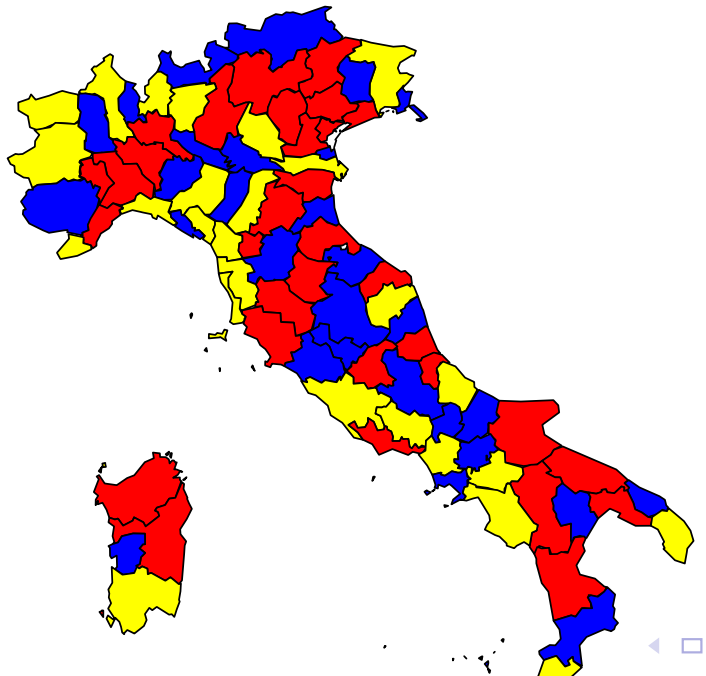
More colors

Create a vector:

```
colors <- c("blue","red","yellow")
```

and use the vector to specify the color

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



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)
```

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

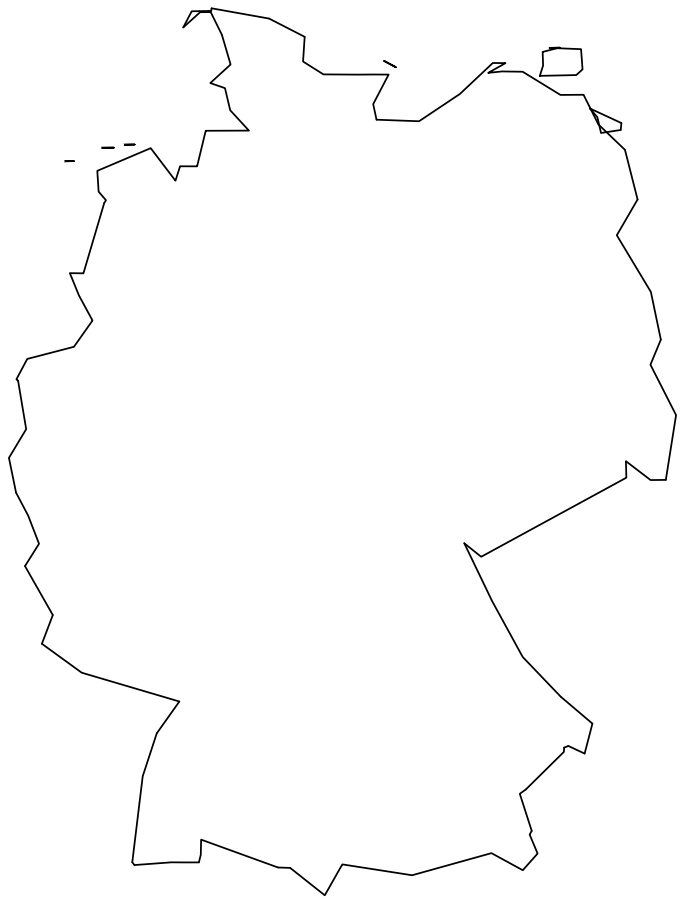
Maps with only one argument

```
map("county")
```

Argument	What
county	US Counties
france	France
italy	Italy
nz	New Zealand
state	US States
usa	USA
world	Countries of the world

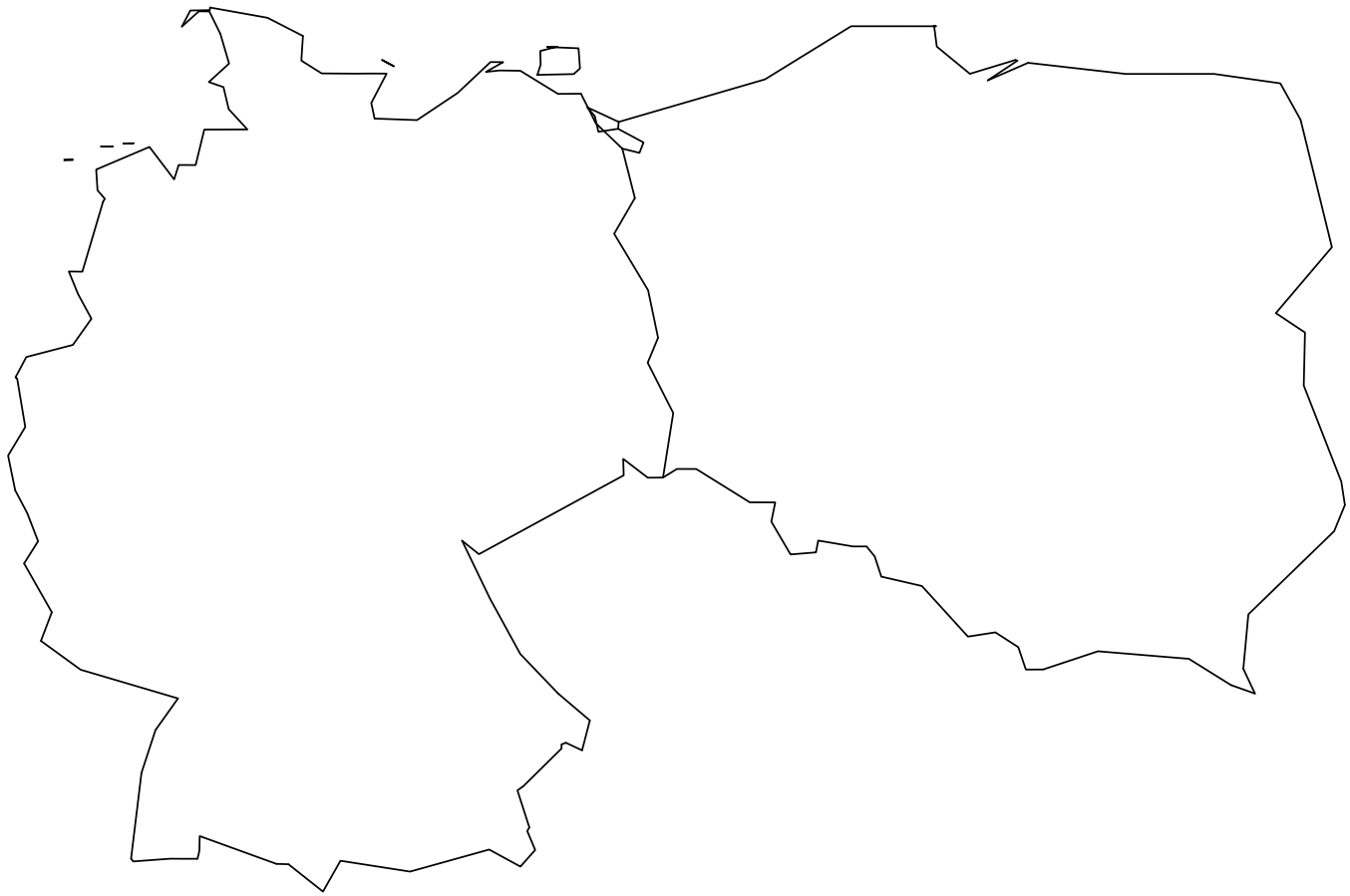
A map for Germany

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



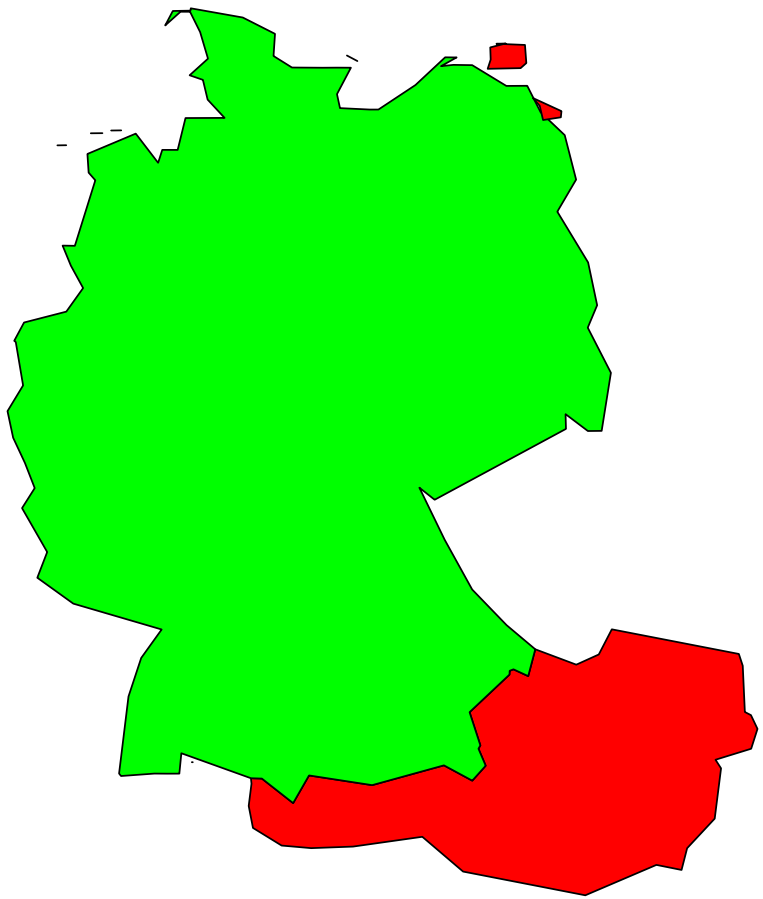
Choropleths - package maps - two countries

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



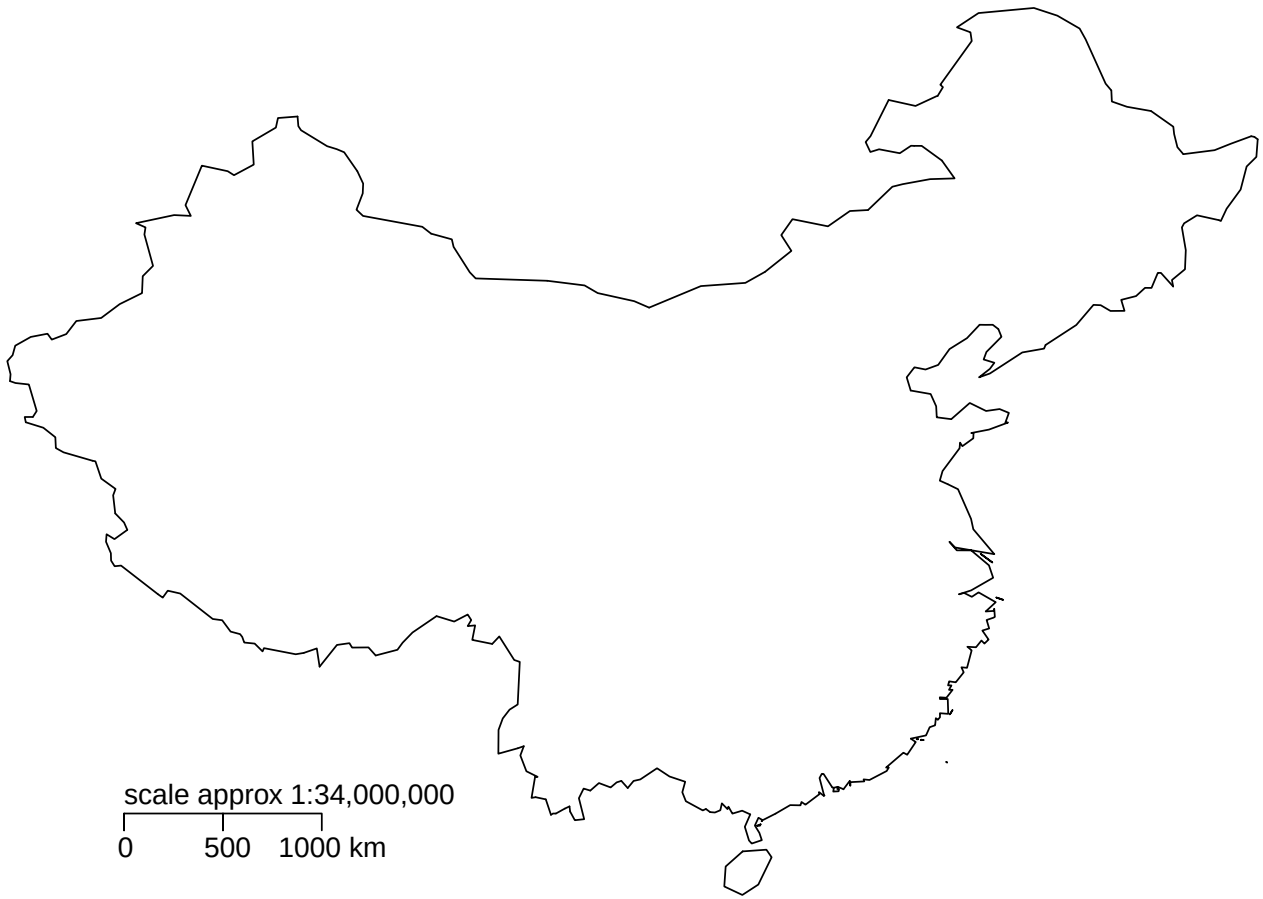
2 countries with color

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



Choropleths - package maps - additional features

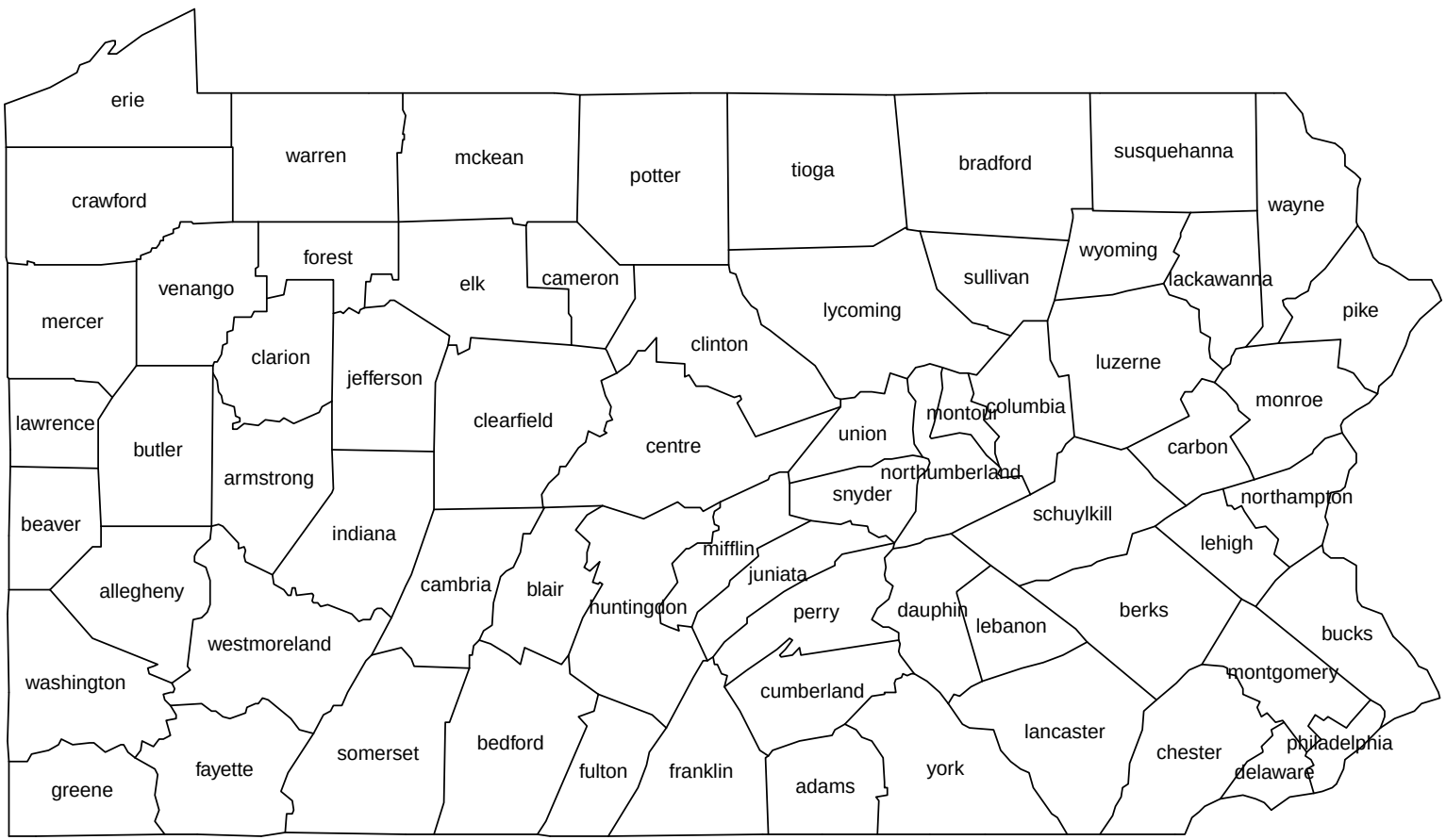
```
map("world", "China")
map.scale()
```



Choropleths - package maps - additional features

Like `map`, but labels the regions:

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



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

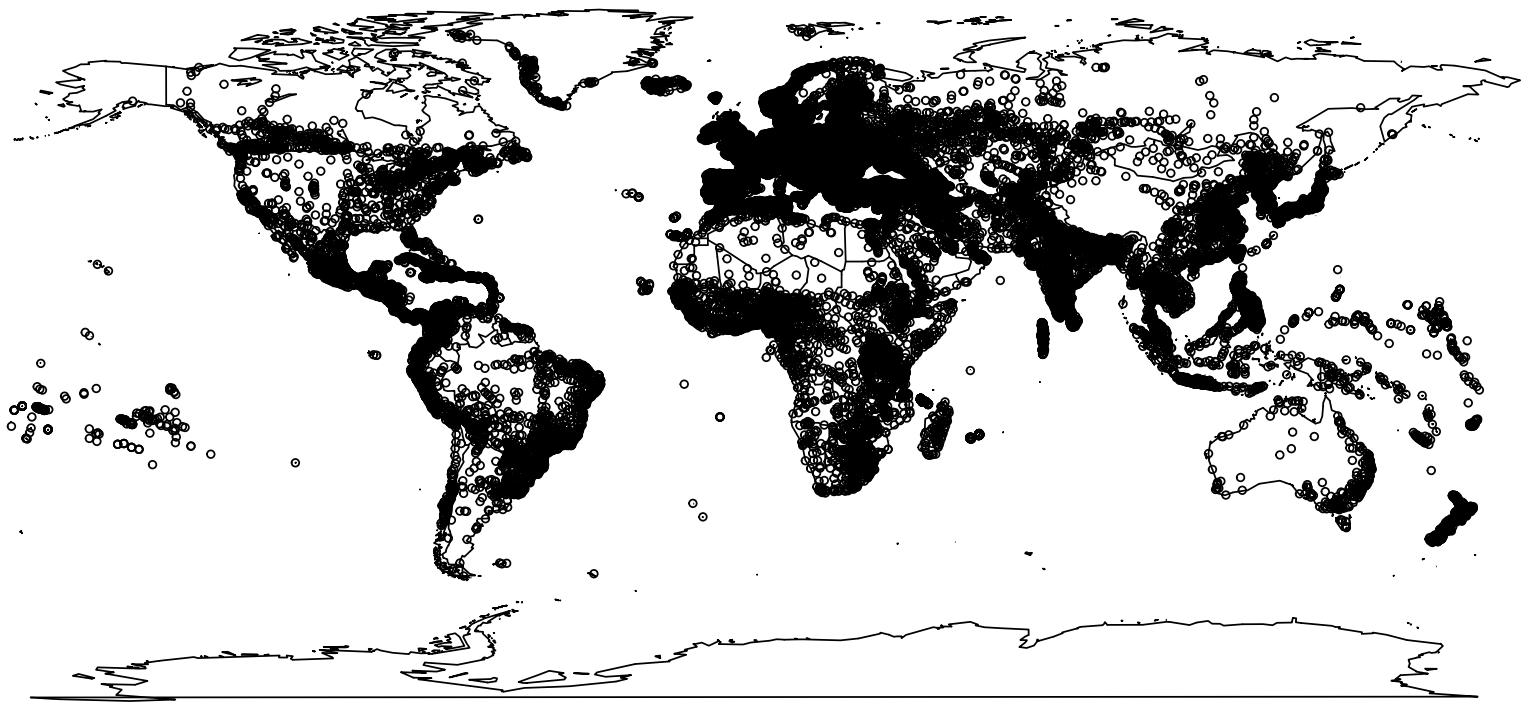
```
head(world.cities)
```

Overview of world cities data

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

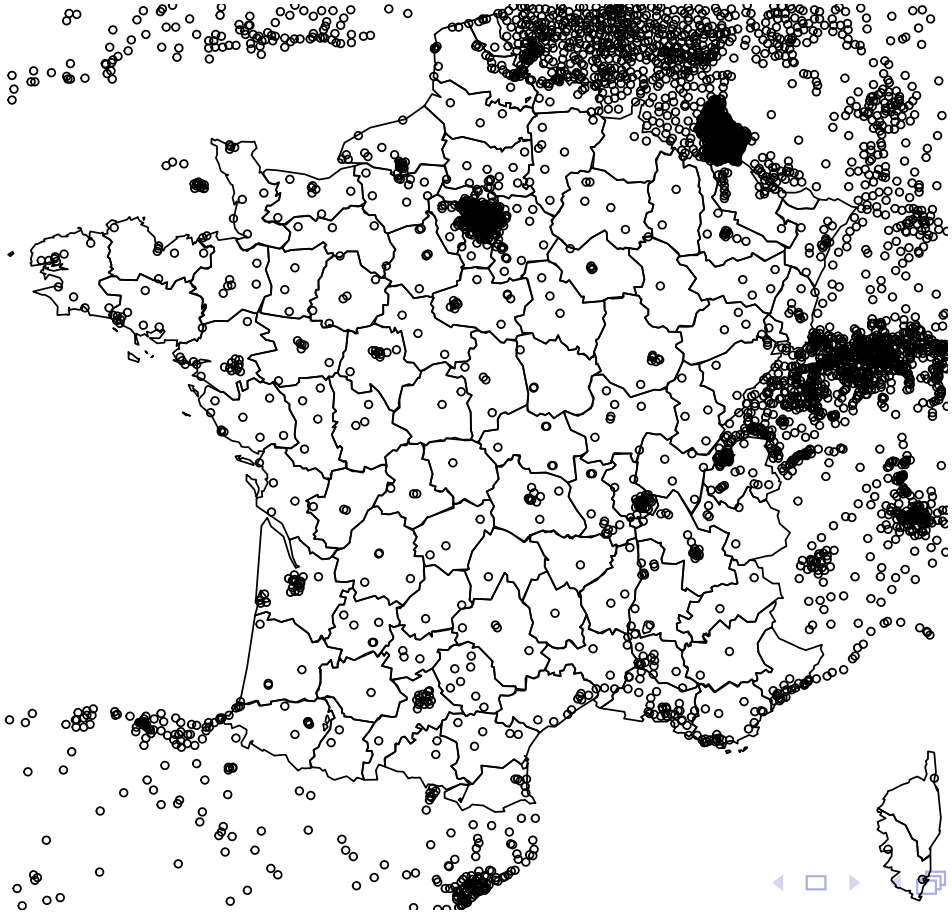
Map the cities of the world

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



Cities of France

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



Only cities of France

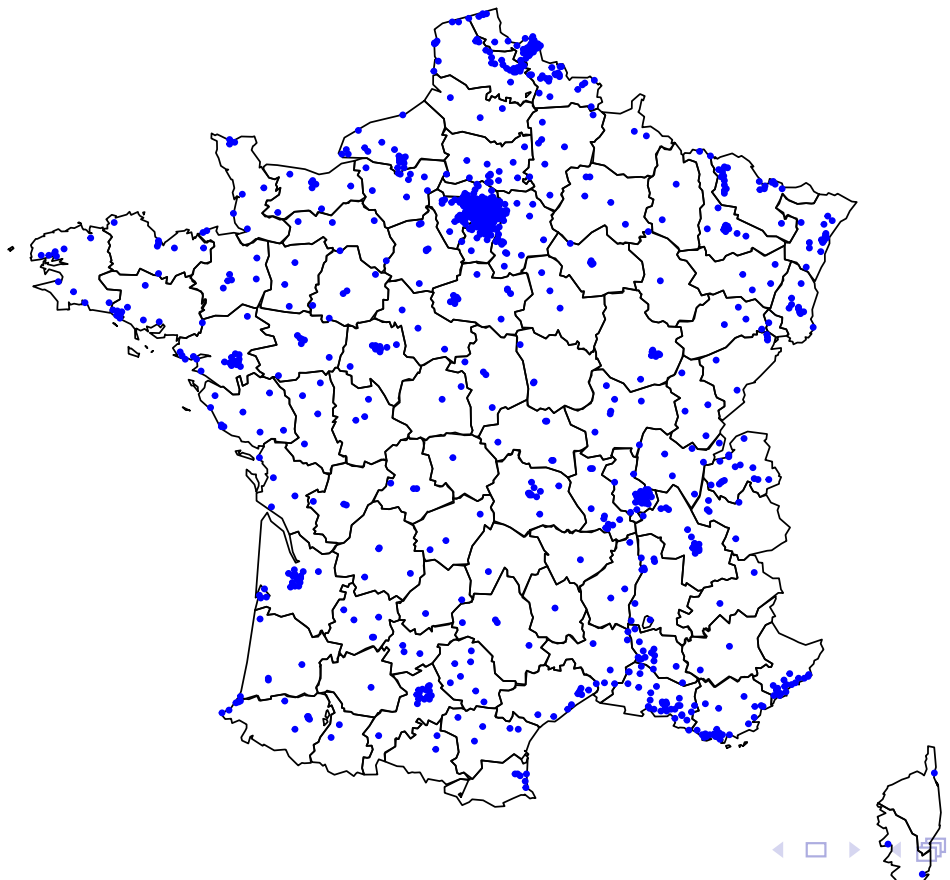
```
FrenchCity <- world.cities$country.etc=="France"
FCit <- world.cities[FrenchCity,]
```

	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

Only cities of France

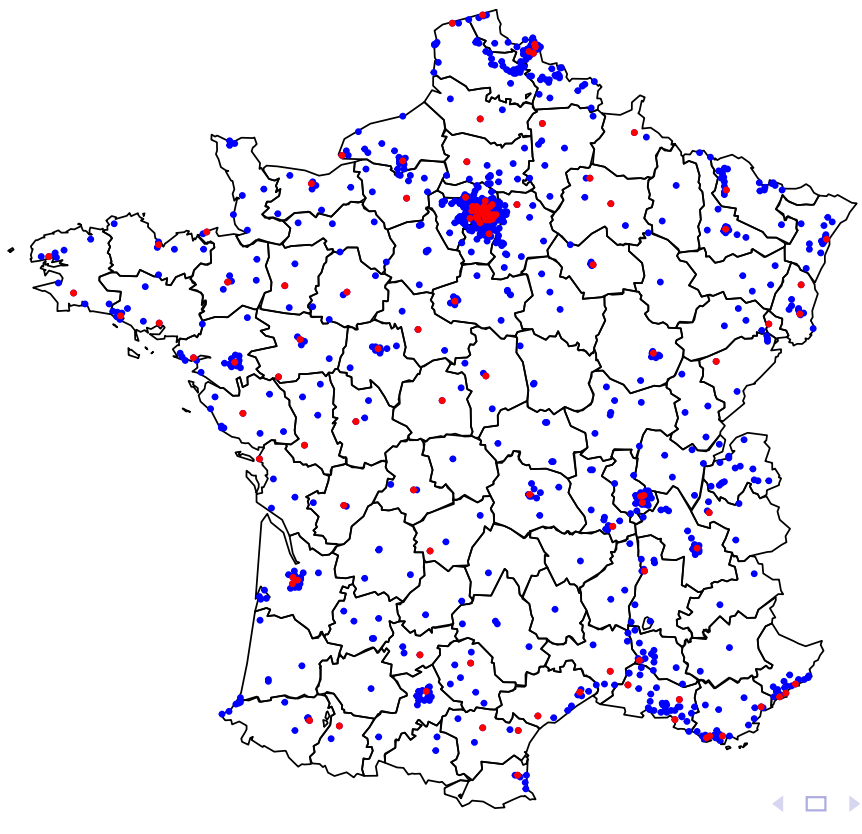
Now it is possible to map only the french cities:

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



Different sizes - different colors

```
FCit_Bc<-FCit[FCit$pop>50000,]
map("france")
map.cities(FCit,col="blue",pch=20)
map.cities(FCit_Bc,col="red",pch=20)
```



Data basis - 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.

Example: US Unemployment

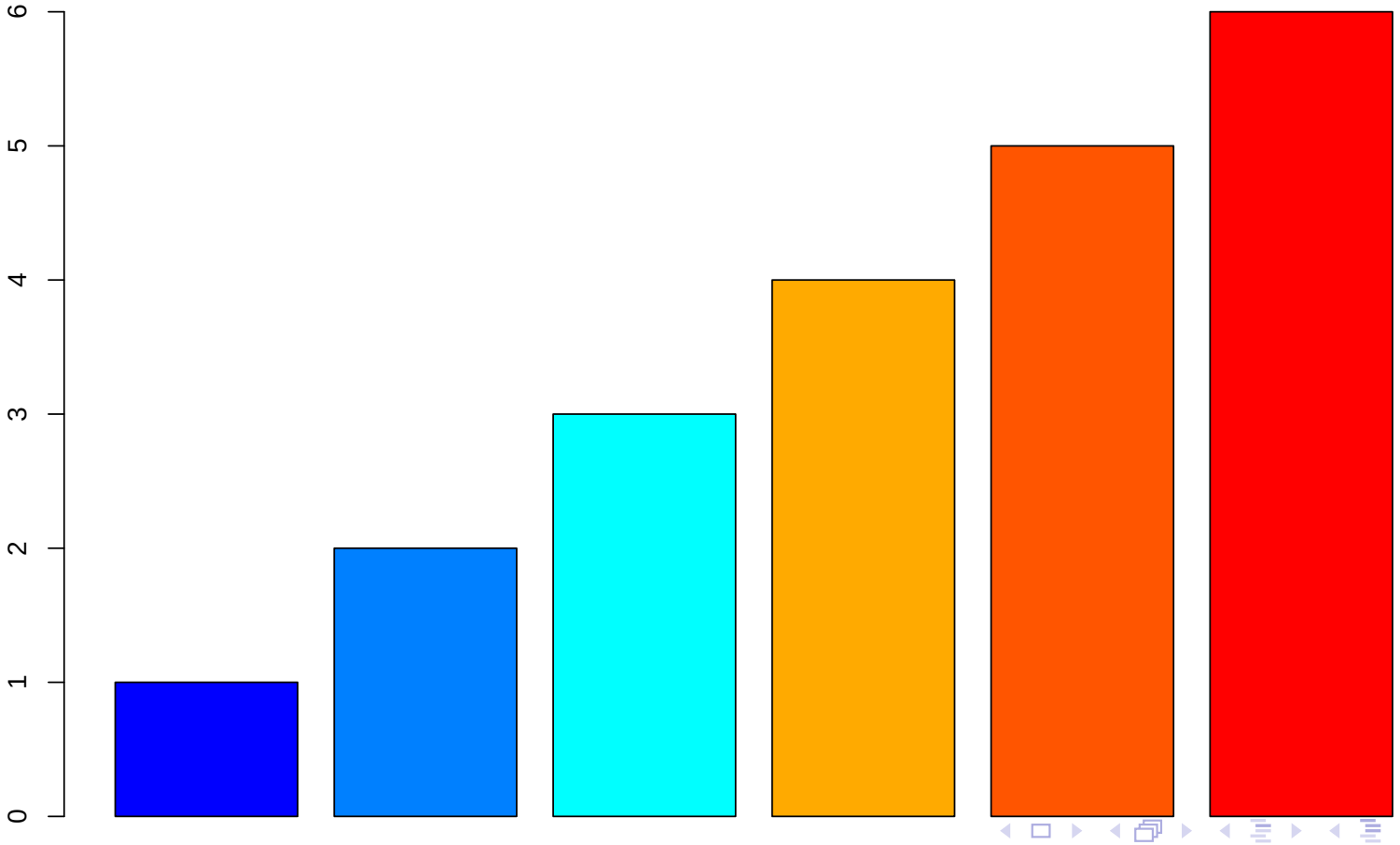
- More about the usage of package maps

Get the data:

```
library(maps)
data(unemp)
data(county.fips)
```

Colour gradients

```
library(colorRamps)
colors <- blue2red(6)
barplot(1:6,col=colors)
```

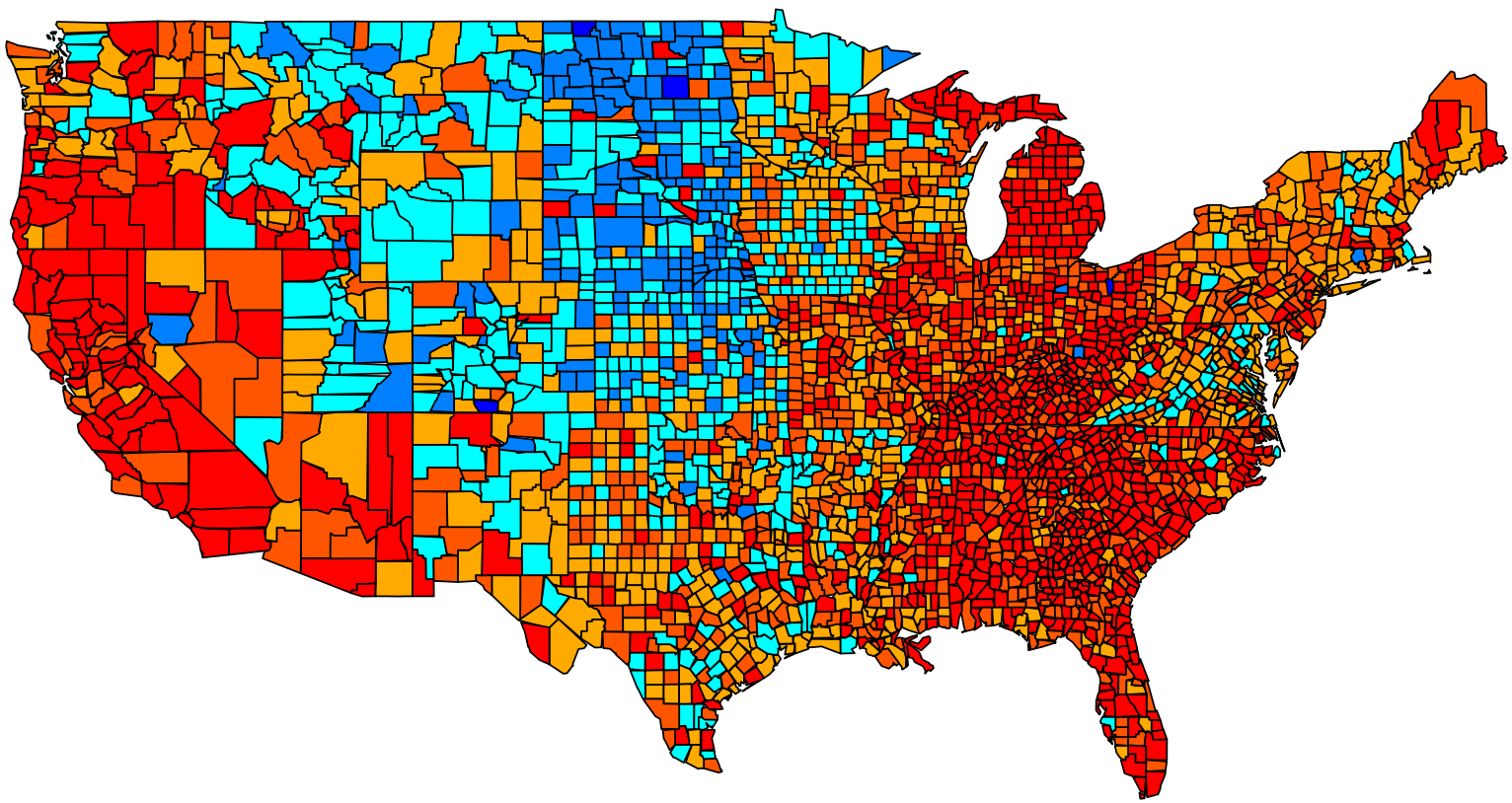


Example: US Unemployment - color coding

```
unemp$colorSteps <- cut(unemp$unemp,
          c(0, 2, 4, 6, 8,10, 100))
colorsmatch <- unemp$colorSteps[match(county.fips$fips,
          unemp$fips)]
```

Example: US Unemployment

```
map("county", col = colors[colorsmatch],
    fill = TRUE)
```



Further links and resources

- Using R — Working with Geospatial Data
- Robin Lovelace, James Cheshire - Introduction to visualising spatial data in R
- Maps in R: Introduction - Drawing the map of Europe