

# R-Paket maps

Jan-Philipp Kolb

22 Februar 2017

# Politische Karte:

Paket maps

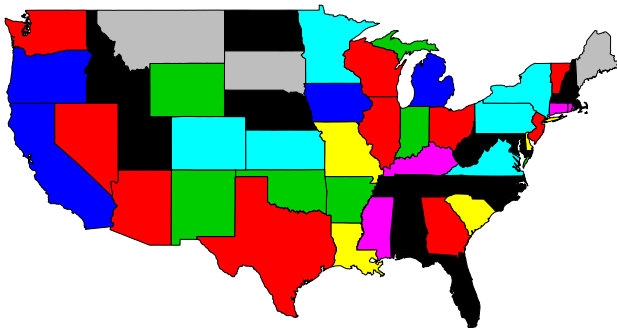
Jan-Philipp  
Kolb

- Eine politische Karte zeigt keine topographischen Informationen.
- Hauptsächlich sollen administrative Grenzen dargestellt werden.
- Teilweise werden auch Städte mit eingezeichnet
- Ein typisches Beispiel ist eine Karte der US-Staaten

# Politische Karte

Paket maps

Jan-Philipp  
Kolb



# Das R-Paket maps

Das R-Paket maps

Jan-Philipp  
Kolb

```
library(maps)
```

Wie man Hilfe bekommt:

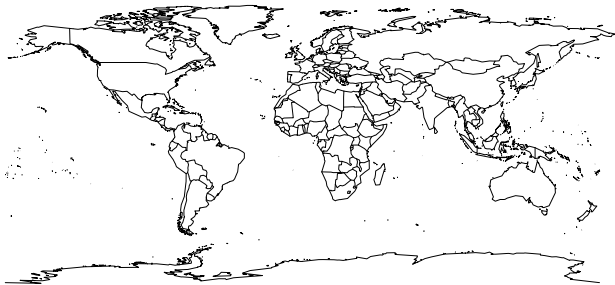
```
?map
```

# Hallo Welt

Paket maps

Jan-Philipp  
Kolb

`map()`



# Ein erstes Argument

Paket maps

Der gleiche Befehl mit einem Argument:

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

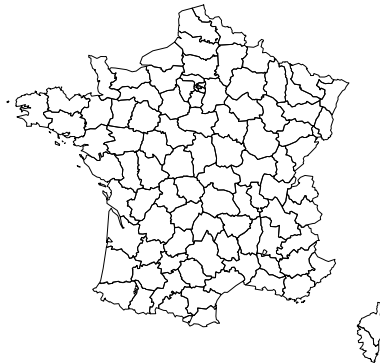


# Eine Frankreichkarte

Paket maps

Jan-Philipp  
Kolb

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



# Politische Karte - Italien

Grenzen in blau:

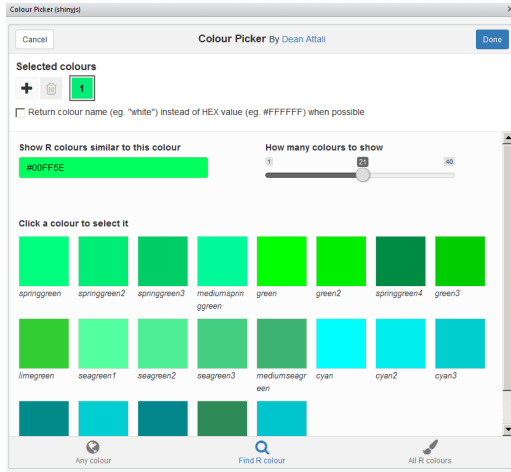
```
library ( maps )  
map ("italy", col = "blue")
```





# Add In: Colour Picker

- RStudio Version 0.99.893 oder höher ist notwendig
- Das Paket shinyjs muss installiert sein.



# Choroplethen - R-Paket maps

R-Paket maps

Wir wollen die Flächen blau einfärben:

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

Jan-Philipp  
Kolb



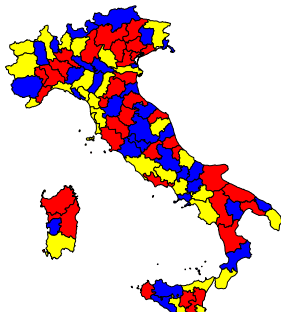
# Mehr Farben

Einen Vektor erzeugen:

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

nun nutzen wir den Vektor um die Farbe zu spezifizieren

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



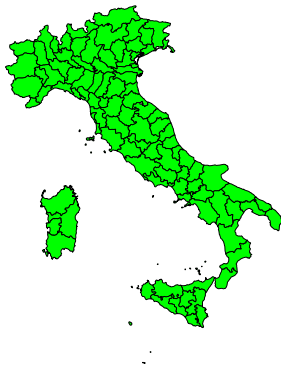
## Excurs: mehr Farben!

Paket maps

Jan-Philipp  
Kolb

Wir können auch den `rgb` Befehl nutzen um eigene Farben zu erzeugen:

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

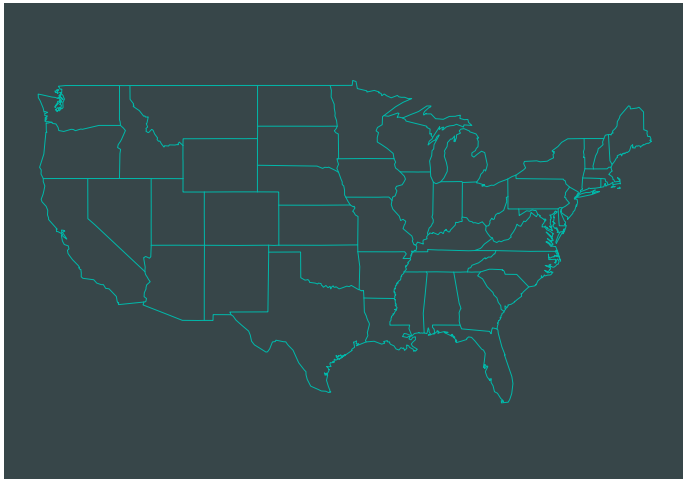


# Hintergrund

Plotting maps

Jan-Philipp  
Kolb

```
map("state", col="#01B8AA", bg="#374649")
```



## Auch ausprobieren:

```
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))
```



# Choroplethen - R-Paket maps

R-Paket maps

Jan-Philipp  
Kolb

Wenn man wissen will, welche Region sich wohinter verbirgt:

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

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

# Karten mit nur einem Argument

Paket maps

Jan-Philipp  
Kolb

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

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



# Eine Karte für Deutschland

Paket maps

Jan-Philipp  
Kolb

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



# R-Paket maps - zwei Länder

R-Paket maps

Jan-Philipp  
Kolb

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



## Zwei Länder mit mehr Farbe

Paket maps

Jan-Philipp  
Kolb

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

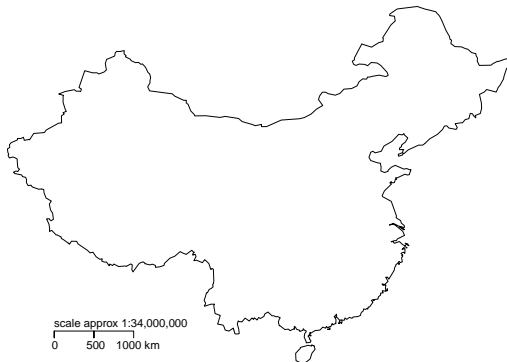


# R-Paket maps - mehr Feature

Paket maps

Jan-Philipp  
Kolb

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

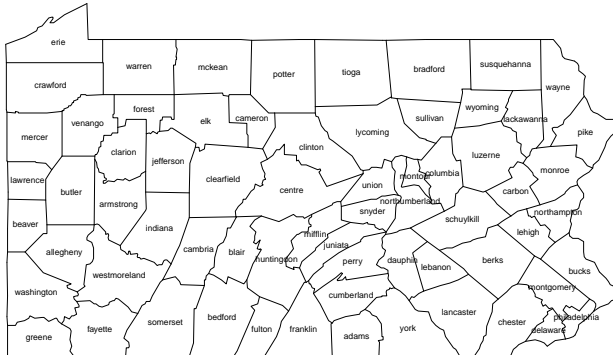


# R-Paket maps - mehr Feature

R-Paket maps

Wie map, aber mit Bezeichnungen für die Regionen:

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



Jan-Philipp  
Kolb

# R-Paket maps - Städte der Welt

R-Paket maps

Jan-Philipp  
Kolb

mit dem Befehl `data` kann man spezifische Datensätze einlesen.

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

`head` - den ersten Teil eines Objekts zurück geben

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

# Überblick über Datensatz Städte der Welt

Paket maps

Jan-Philipp  
Kolb

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

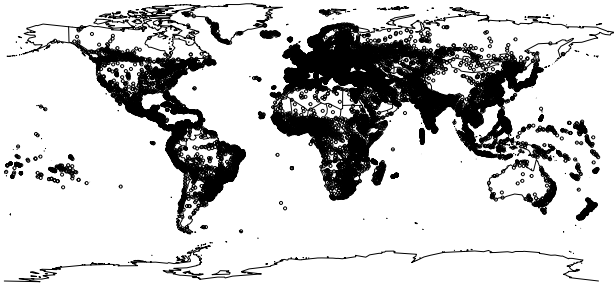
# Die Städte der Welt kartieren

Plotting maps

Jan-Philipp  
Kolb

```
map()
```

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



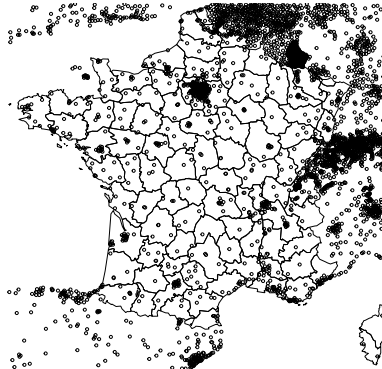


# Frankreichs Städte

Plot maps

Jan-Philipp  
Kolb

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



# Nur französische Städte

Paket maps

Jan-Philipp  
Kolb

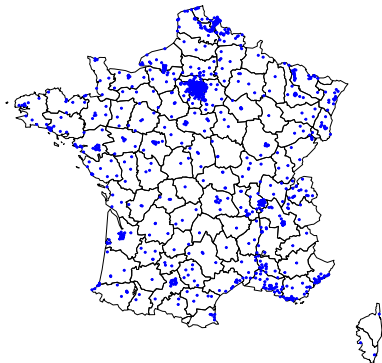
```
FrenchCity <- world.cities$country.etc=="France"  
FCit <- world.cities[FrenchCity,]
```

	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

# Nur französische Städte

Jetzt ist es möglich nur die französischen Städte zu kartieren:

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

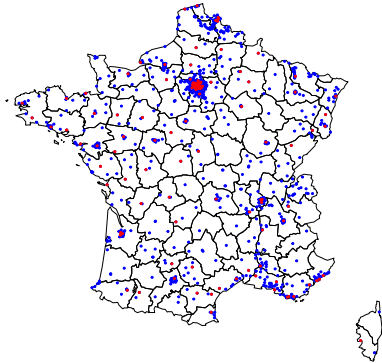


# Verschiedene Größen - verschiedene Farben

Plotting maps

Jan-Philipp  
Kolb

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



# Beispiel für Fortgeschrittene

Paket maps

Jan-Philipp  
Kolb

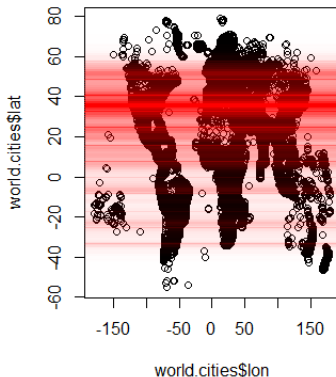
- Arthur Charpentier auf R-bloggers - Where people live

```
library(maps)
data("world.cities")
X <- world.cities[,c("lat", "pop")]
plot(world.cities$lon, world.cities$lat)
for(i in 1:length(X$lat)) {
  abline(h=X$lat[i], col=rgb(1,0,0,.001), lwd=2.7)
}
```

# Wo leben viele Menschen

Paket: maps

Jan-Philipp  
Kolb



**Figure 2:** Where People Live

# Thematische Karten mit dem R-Paket maps

R-Paket maps

Jan-Philipp  
Kolb

## Datenbasis - 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.*

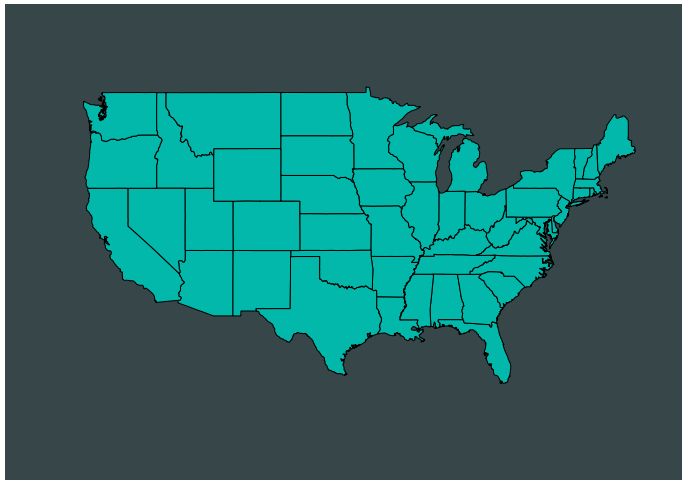
# Verschiedene Projektionen

Paket maps

Jan-Philipp  
Kolb

```
library(mapproj)
```

```
map("state", col="#01B8AA", bg="#374649", fill=TRUE,
```



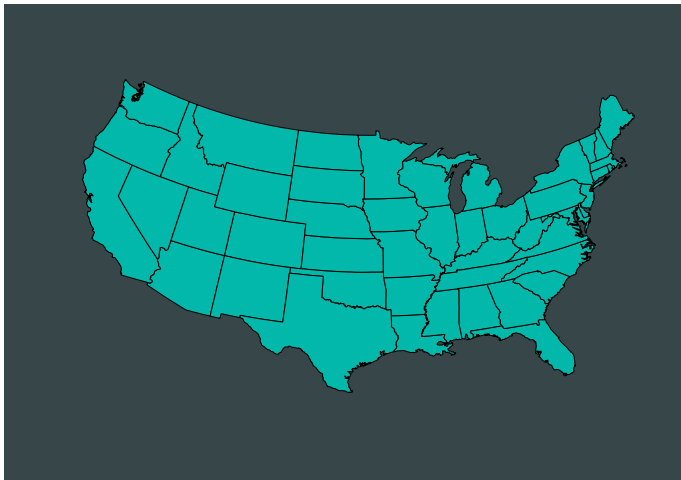


# Eine andere Projektion

Paket maps

Jan-Philipp  
Kolb

```
map("state", col="#01B8AA", bg="#374649", fill=TRUE,
```

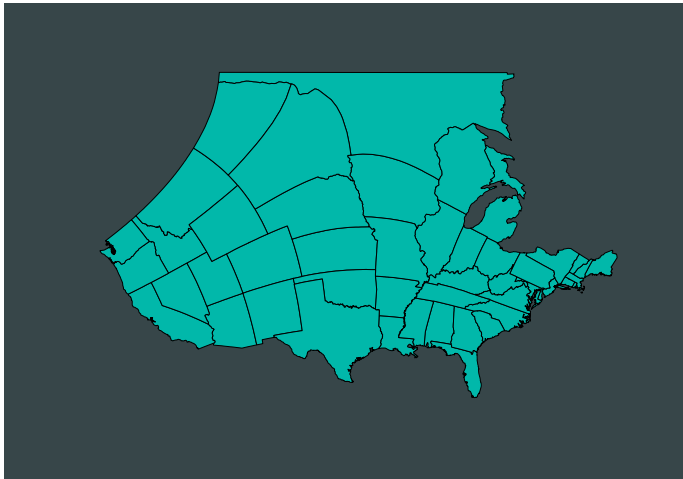


# Eine weitere Projektion

Paket maps

Jan-Philipp  
Kolb

```
map("state", col="#01B8AA", bg="#374649", fill=TRUE,
```



# Mehr Links und Quellen

Packet maps

Jan-Philipp  
Kolb

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