

Interaktive Karten

Jan-Philipp Kolb

12 Juni 2017

Die Daten - Weltkulturerbe

- die Daten einlesen:

```
url <- "https://raw.githubusercontent.com/Japhilko/GeoData/master/2015/data/whcSites.csv"
```

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

- die Daten werden eingeschränkt:

```
whcSitesDat <- with(whcSites, data.frame(name_en,
                                         category))
```

Eine Tabelle erzeugen mit knitr

```
library(knitr)
kable(head(whcSitesDat))
```

name_en	category
Cultural Landscape and Archaeological Remains of the Bamiyan Valley	Cultural
Minaret and Archaeological Remains of Jam	Cultural
Historic Centres of Berat and Gjirokastra	Cultural
Butrint	Cultural
Al Qal'a of Beni Hammad	Cultural
M'Zab Valley	Cultural

Eine erste interaktive Tabelle - Das Paket DT

```
install.packages("DT")
```

DT: An R interface to the DataTables library

The R package **DT** provides an R interface to the JavaScript library **DataTables**. R data objects (matrices or data frames) can be displayed as tables on HTML pages, and **DataTables** provides filtering, pagination, sorting, and many other features in the tables.

You may install the stable version from CRAN, or the development version using `devtools::install_github('rstudio/DT')` if necessary (this website reflects the development version of **DT**):

Figure 1:

Weitere Variablen WHC Datensatz

```
whcSitesDat2 <- with(whcSites, data.frame(name_en, category,
                                           longitude, latitude, date_inscribed, area_hectares, danger_list))
```

- mit dem Befehl `datatable` kann man eine erste interaktive Tabelle erstellen:

```
library('DT')
datatable(whcSitesDat2)
```

Das Ergebnis bei Rpubs

<http://rpubs.com/Japhilko82/WHCdata>

RPubs brought to you by RStudio

Show entries Search:

	name_en	category	longitude	latitude	date_inscribed	area_hectares	danger_list
1	Cultural Landscape and Archaeological Remains of the Bamiyan Valley	Cultural	67.82525	34.84694	2003	158.9265	Y 2003
2	Minaret and Archaeological Remains of Jam	Cultural	64.51605556	34.39655556	2002	70	Y 2002
3	Historic Centres of Berat and Gjirokastra	Cultural	20.13333333	40.06944444	2005	58.9	
4	Butrint	Cultural	20.02611111	39.75111111	1992		
5	Al Qal'a of Beni Hammad	Cultural	4.78684	35.81844	1980	150	
6	M'Zab Valley	Cultural	3.68333	32.48333	1982	665.03	
7	DjÃ©mila	Cultural	5.73667	36.32056	1982	30.6	
8	Timgad	Cultural	6.63333	35.45	1982	90.54	
9	Kasbah of Algiers	Cultural	3.06028	36.78333	1992	60	
10	Madriu-Perafita-Claror Valley	Cultural	1.59555556	42.49472222	2004	4247	

Showing 1 to 10 of 1,031 entries Previous 2 3 4 5 ... 104 Next

Figure 2:

Das Paket `magrittr`

- `magrittr` - für den Pipe Operator in R:

```
install.packages("magrittr")
```

```
library("magrittr")
```

Simpler R coding with pipes > the present and future of the magrittr package



Tal Galili

August 5, 2014

Guest Post, R, R programming

0
SHARES

f Share

🐦 Tweet

✉ Subscribe

This is a guest post by *Stefan Milton*, the author of the [magrittr](#) package which introduces the %>% operator to R programming.

Figure 3:

Die Pipes nutzen

```
library(magrittr)

str1 <- "Hallo Welt"
str1 %>% substr(1,5)

## [1] "Hallo"

str1 %>% substr(1,5) %>% toupper()

## [1] "HALLO"
```

Das Paket leaflet

- leaflet - um interaktive Karten mit der JavaScript Bibliothek leaflet zu erzeugen

```
install.packages("leaflet")
```

```
library("leaflet")
```

- Bei leaflet wird mit so genannten Tiles gearbeitet.
- Robin Lovelace - The leaflet package for online mapping in R

Was sind Tiles?

- Die Übersetzung aus dem englischen ist Fliese und dieses Bild erklärt es eigentlich ganz gut.
- Es geht um Kachelgrafiken.
- Es ist eine Grafik bezeichnet, die mosaikartig zusammengesetzt ein vielfach größeres Gesamtbild ergibt.

Eine interaktive Karte erstellen

```
m <- leaflet() %>%  
  addTiles() %>% # Add default OpenStreetMap map tiles  
  addMarkers(lng=whcSites$lon,  
             lat=whcSites$lat,  
             popup=whcSites$name_en)  
m
```

Die Karte zeigen



Figure 4:

Farbe hinzu

- die unterschiedlichen Kategorien farblich einfärben

```
whcSites$color <- "red"  
whcSites$color[whcSites$category=="Cultural"] <- "blue"  
whcSites$color[whcSites$category=="Mixed"] <- "orange"
```

Eine Karte mit Farbe erzeugen

```
m1 <- leaflet() %>%
  addTiles() %>%
  addCircles(lng=whcSites$lon,
             lat=whcSites$lat,
             popup=whcSites$name_en,
             color=whcSites$color)
```

Die Karte mit mehr Farbe

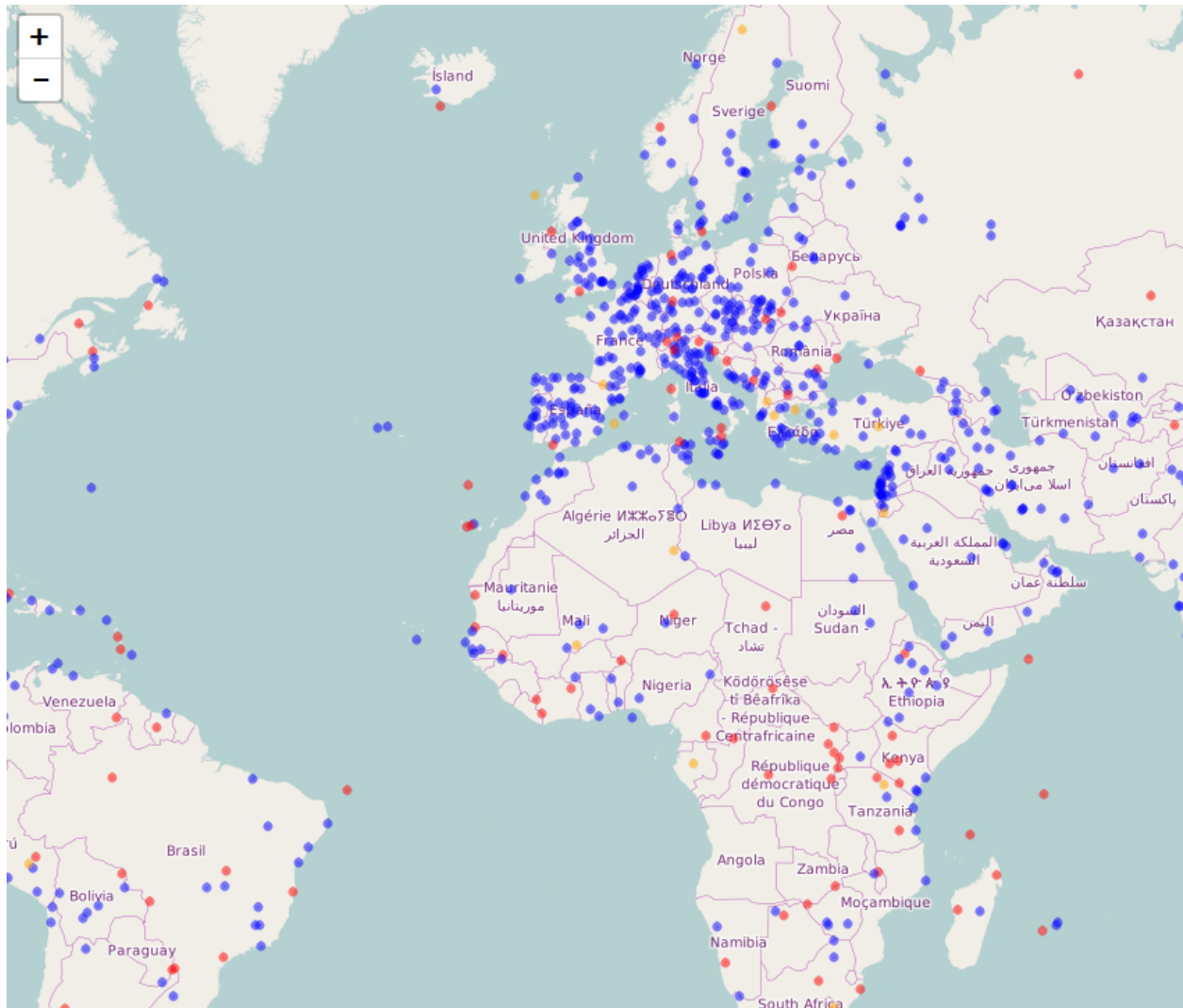


Figure 5: Weltkulturerbe

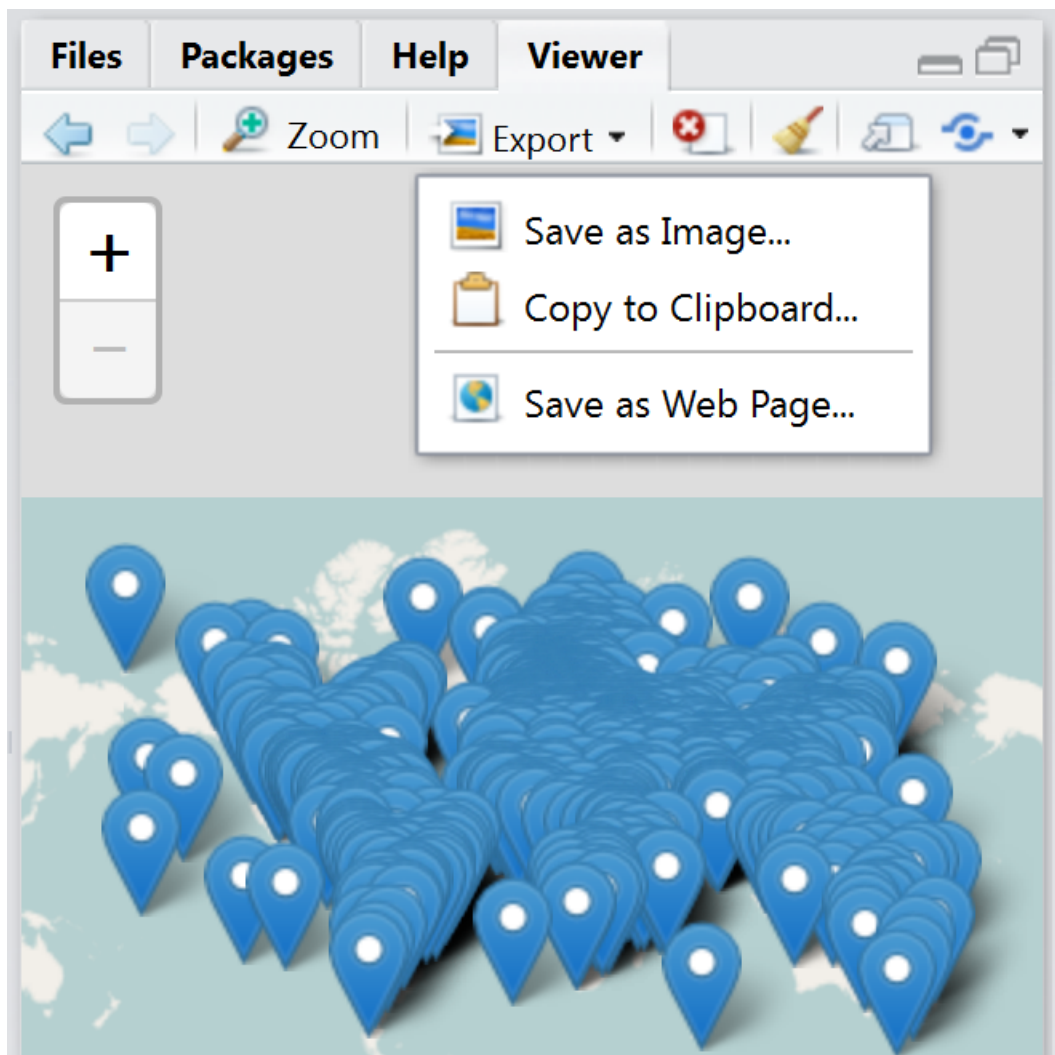


Figure 6:

Die Karte abspeichern

Beispiel US Staaten

```
library(maps)
mapStates = map("state", fill = TRUE, plot = FALSE)
leaflet(data = mapStates) %>% addTiles() %>%
  addPolygons(fillColor = topo.colors(10, alpha = NULL), stroke = FALSE)
```

Der Befehl `setView`

- mit `setView` kann man bestimmen welchen Ausschnitt man für die Hintergrundkarte haben möchte
- dazu muss man die latitude und Longitude Koordinaten und ein zoom Level angeben
- dabei kann man nur ganze Zahlen angeben
- je kleiner die Zahl, desto größer ist der Kartenausschnitt:
- level 3 - Kontinent
- level 10 - Stadt
- level 21 - Gebäude

Die Basiskarte ändern

- Neben der Default Basiskarte kann man auch andere Hintergründe aktivieren

```
m <- leaflet() %>% setView(lng = -71.0589, lat = 42.3601, zoom = 12)
m %>% addTiles()
m %>% addProviderTiles("Stamen.Toner")
```

Basiskarte - CartoDB

```
m %>% addProviderTiles("CartoDB.Positron")
```

Esri.NatGeoWorldMap

```
m %>% addProviderTiles("Esri.NatGeoWorldMap")
```

OpenTopoMap

```
m %>% addProviderTiles("OpenTopoMap")
```

Thunderforest.OpenCycleMap

```
m %>% addProviderTiles("Thunderforest.OpenCycleMap")
```

WMS Tiles hinzufügen

```
leaflet() %>% addTiles() %>% setView(-93.65, 42.0285, zoom = 4) %>%  
  addWMSTiles(  
    "http://mesonet.agron.iastate.edu/cgi-bin/wms/nexrad/n0r.cgi",  
    layers = "nexrad-n0r-900913",  
    options = WMSTileOptions(format = "image/png", transparent = TRUE),  
    attribution = "Weather data © 2012 IEM Nexrad"  
  )
```

Mehrere Layer miteinander kombinieren

```
m %>% addProviderTiles("MtbMap") %>%  
  addProviderTiles("Stamen.TonerLines",  
    options = providerTileOptions(opacity = 0.35)) %>%  
  addProviderTiles("Stamen.TonerLabels")
```

Andere Marker benutzen

```
greenLeafIcon <- makeIcon(  
  iconUrl = "http://leafletjs.com/examples/custom-icons/leaf-green.png",  
  iconWidth = 38, iconHeight = 95,  
  iconAnchorX = 22, iconAnchorY = 94,  
  shadowUrl = "http://leafletjs.com/examples/custom-icons/leaf-shadow.png",  
  shadowWidth = 50, shadowHeight = 64,  
  shadowAnchorX = 4, shadowAnchorY = 62  
)  
  
leaflet(data = quakes[1:4,]) %>% addTiles() %>%  
  addMarkers(~long, ~lat, icon = greenLeafIcon)
```

Andere Icons einfügen

- es lassen sich alle möglichen Icons einfügen

```
menIcon <- makeIcon("https://img.clipartfest.com/707b339dc88f57bbd5d88377891131e3_bean-people-clipart-c",  
  iconWidth = 38, iconHeight = 95,  
  iconAnchorX = 22, iconAnchorY = 94)  
  
leaflet(data = quakes[1:4,]) %>% addTiles() %>%  
  addMarkers(~long, ~lat, icon = menIcon)
```

Cluster Optionen für Marker

```
leaflet(quakes) %>% addTiles() %>% addMarkers(  
  clusterOptions = markerClusterOptions()  
)
```


Ein Rechteck hinzufügen

```
leaflet() %>% addTiles() %>%  
  addRectangles(  
    lng1=-118.456554, lat1=34.078039,  
    lng2=-118.436383, lat2=34.062717,  
    fillColor = "transparent"  
  )
```

Links und Quellen

- 4 Tricks zum Arbeiten mit Leaflet
- <http://www.r-bloggers.com/the-leaflet-package-for-online-mapping-in-r/>
- <https://rstudio.github.io/leaflet/>

Aufgabe leaflet

- Verwenden Sie die Adresse, die Sie zuvor geokodiert haben, um eine interaktive Karte um diesen Punkt herum zu erstellen.