

R Schnittstellen - Interaktive Darstellung mit Javascript und R

Jan-Philipp Kolb

8 Mai 2017

Interaktive Karten mit dem Javascript Paket leaflet

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))
```

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 10 entries

Search:

	name_en	category	longitude	latitude	date_inscribed	area_hectares	danger_list
101	Angkor	Cultural	103.8333333	13.43333333	1992	40100	P 1992-2004
96	Srebarna Nature Reserve	Natural	27.07806	44.11444	1983	638	P 1992-2003
184	Plitvice Lakes National Park	Natural	15.61444	44.87778	1979	29482	P 1992-1997
182	Old City of Dubrovnik	Cultural	18.09139	42.65056	1979	96.7	P 1991-1998

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)

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

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

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



Figure 4:

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

Die Karte mit mehr Farbe

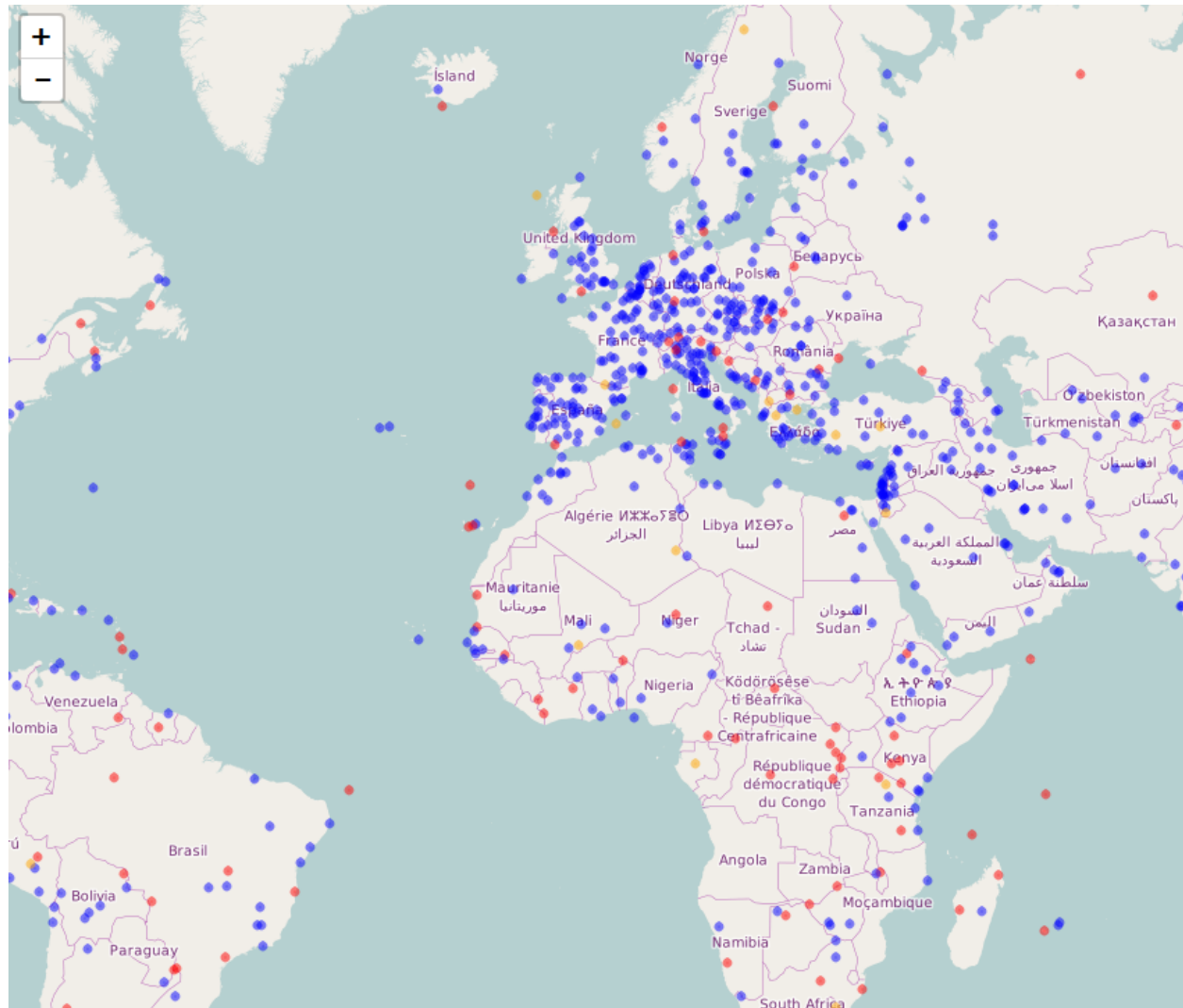


Figure 5: Weltkulturerbe

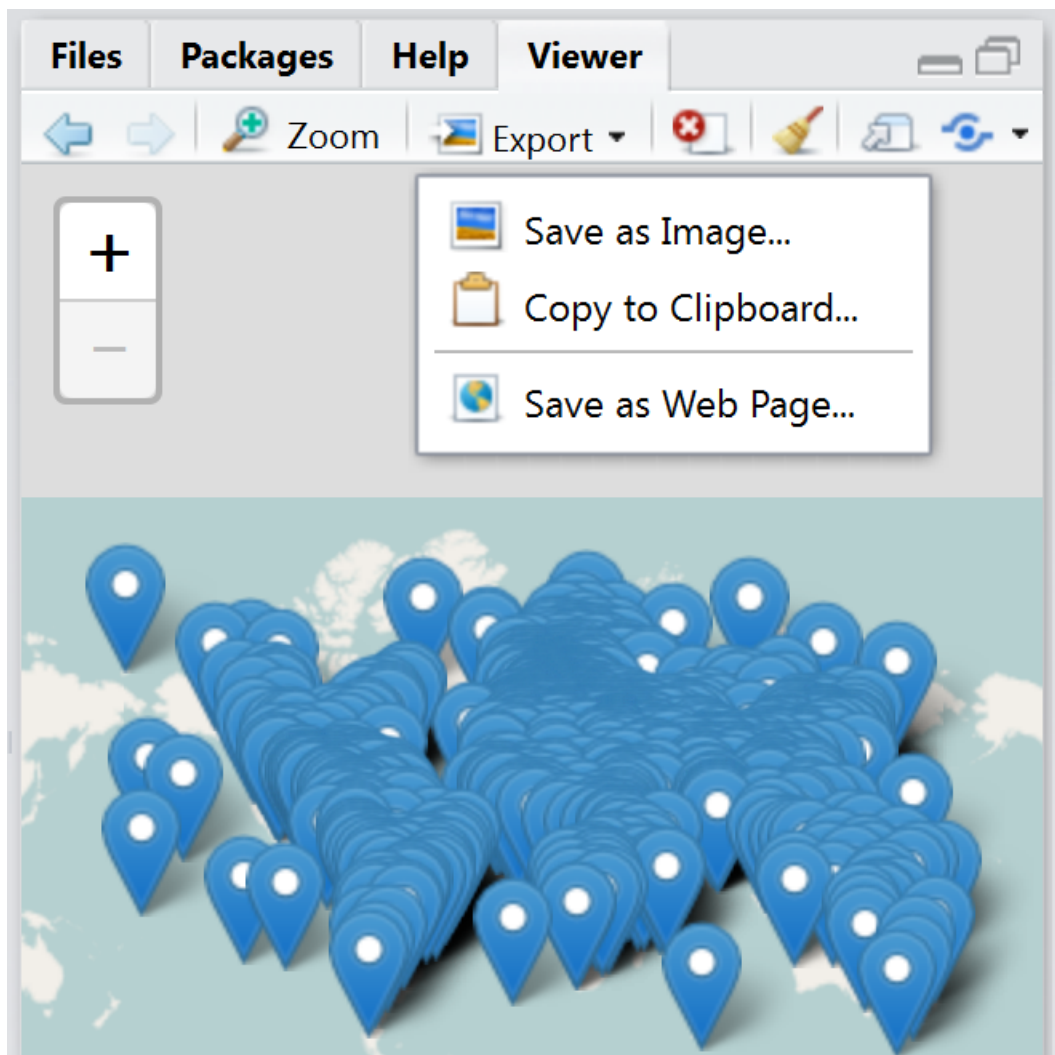


Figure 6:

Die Karte abspeichern

Layers ein- und ausblenden

```
m2 <- leaflet() %>%
  addTiles(group = "OSM (default)") %>%
  addProviderTiles("Stamen.Toner", group = "Toner") %>%
  addProviderTiles("Stamen.TonerLite", group = "Toner Lite") %>%

  addCircles(lng=whcSites$lon,
             lat=whcSites$lat,
             popup=whcSites$name_en) %>%

  addLayersControl(
    baseGroups = c("OSM (default)", "Toner", "Toner Lite"),
    options = layersControlOptions(collapsed = FALSE)
  )
m2
```

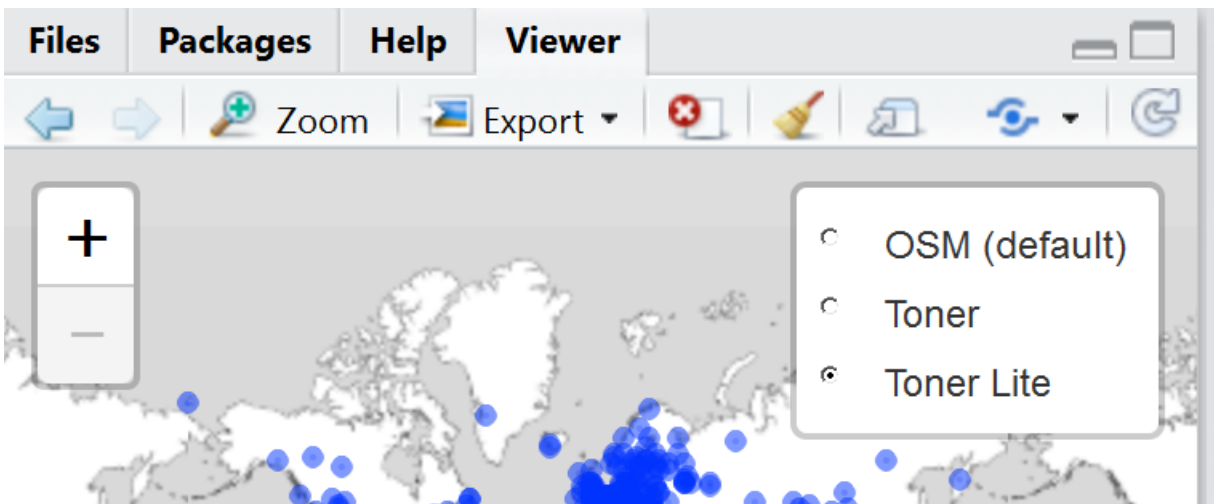


Figure 7:

Ein weiteres Beispiel mit Erdbebendaten

```
outline <- quakes[chull(quakes$long, quakes$lat),]

map <- leaflet(quakes) %>%
  # Base groups
  addTiles(group = "OSM (default)") %>%
  addProviderTiles("Stamen.Toner", group = "Toner") %>%
  addProviderTiles("Stamen.TonerLite", group = "Toner Lite") %>%
  # Overlay groups
  addCircles(~long, ~lat, ~10^mag/5, stroke = F, group = "Quakes") %>%
  addPolygons(data = outline, lng = ~long, lat = ~lat,
             fill = F, weight = 2, color = "#FFFFCC", group = "Outline") %>%
  # Layers control
```

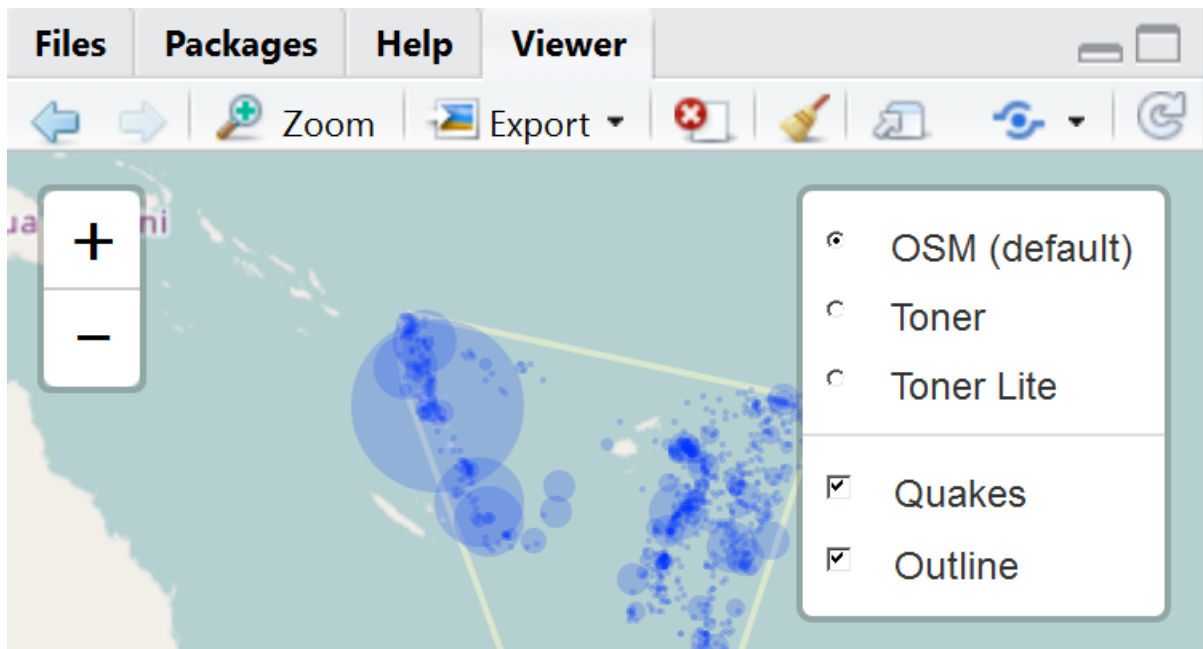


Figure 8:

```
addLayersControl(
  baseGroups = c("OSM (default)", "Toner", "Toner Lite"),
  overlayGroups = c("Quakes", "Outline"),
  options = layersControlOptions(collapsed = FALSE)
)
map
```

Karte mit Polygonen erzeugen

```
library(sp)
Sr1 = Polygon(cbind(c(2, 4, 4, 1, 2), c(2, 3, 5, 4, 2)))
Sr2 = Polygon(cbind(c(5, 4, 2, 5), c(2, 3, 2, 2)))
Sr3 = Polygon(cbind(c(4, 4, 5, 10, 4), c(5, 3, 2, 5, 5)))
Sr4 = Polygon(cbind(c(5, 6, 6, 5, 5), c(4, 4, 3, 3, 4)), hole = TRUE)
Srs1 = Polygons(list(Sr1), "s1")
Srs2 = Polygons(list(Sr2), "s2")
Srs3 = Polygons(list(Sr4, Sr3), "s3/4")
SpP = SpatialPolygons(list(Srs1, Srs2, Srs3), 1:3)
```

- so wird die Karte erzeugt:

```
leaflet(height = "300px") %>% addPolygons(data = SpP)
```

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/>

Interaktive Tabellen mit DataTables

R und die Javascript Data-Driven Documents (D3)

Interaktive Graphiken mit D3 und plotly

Netzwerkgraphiken mit vis.js

Das Erstellen von Ablaufdiagrammen mit mermaid

Addins für RStudio