

B3 Javascript Bibliotheken nutzen

Jan-Philipp Kolb

23 Oktober 2018

Beispiel zu Campingplätzen

- Die Daten stammen von:

<http://www.openstreetmap.de/>

- Dabei wird die Overpass API genutzt:

http://wiki.openstreetmap.org/wiki/Overpass_API

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

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

Überblick über Daten zu Campingplätzen

X	name	tourism	website
1	Campingplatz Winkelbachtal	camp_site	http://www.gruibirg.de
2	Radler-Zeltplatz	camp_site	NA
3	Campingplatz des Naturfreundehauses	camp_site	NA
4	Campingplatz Am Aichstruter Stausee	camp_site	NA
5	NA	camp_site	NA
6	Kandern	camp_site	NA
7	Campingplatz Baiersbronn-Obertal	camp_site	NA
8	Campingplatz Schwabenmälz	camp_site	NA

Notwendige Pakete

magrittr - für den Pipe Operator in R:

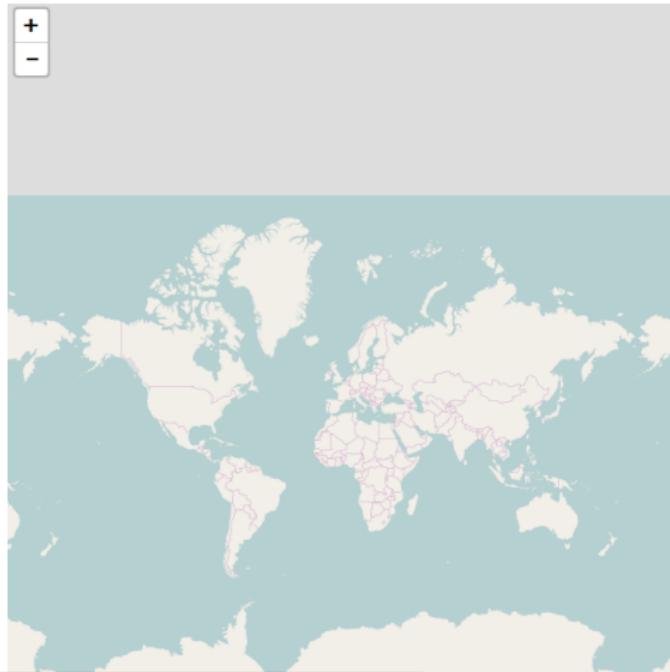
```
library("magrittr")
```

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

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

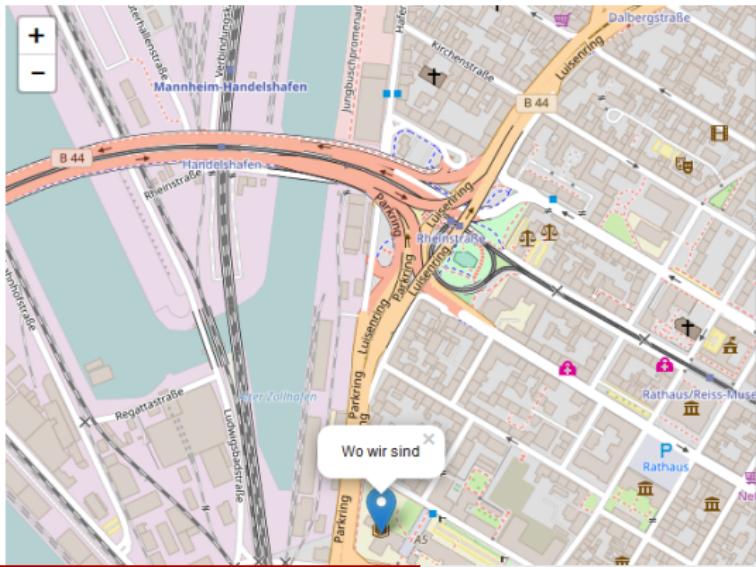
Eine erste interaktive Karte

```
leaflet()%>%  
  addTiles()
```



Auf eine Stadt zoomen

```
leaflet() %>%  
  addTiles() %>%  
  addMarkers(lng=8.456597, lat=49.48738,  
            popup="Wo wir sind")
```



Eine interaktive Karte

```
m <- leaflet() %>%  
  addTiles() %>%  
  addMarkers(lng=CampSites$lon,  
             lat=CampSites$lat,  
             popup=CampSites$name)  
  
m
```

Stamen als Hintergrundkarte

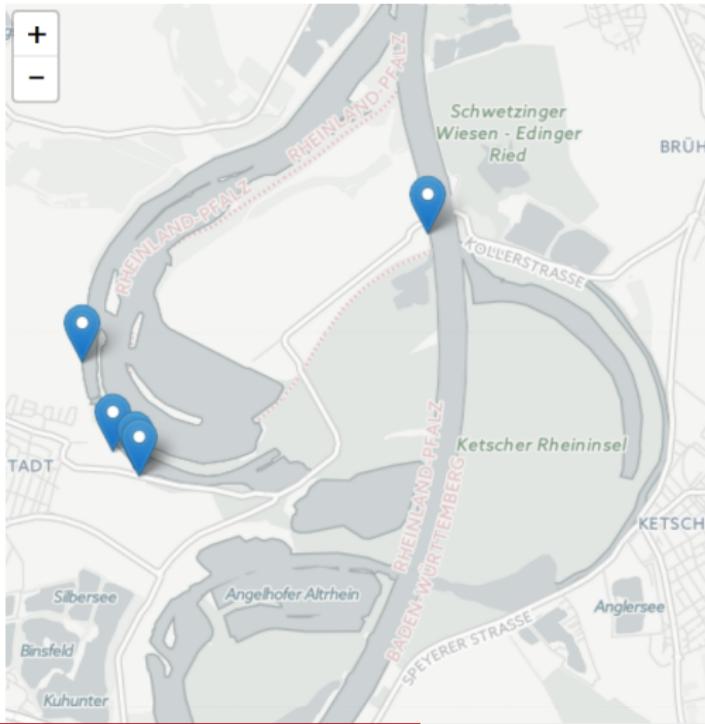
```
m %>% addProviderTiles("Stamen.Toner")
```



Leaflet | © OpenStreetMap contributors, CC-BY-SA, Map tiles by Stamen Design, CC BY 3.0 —

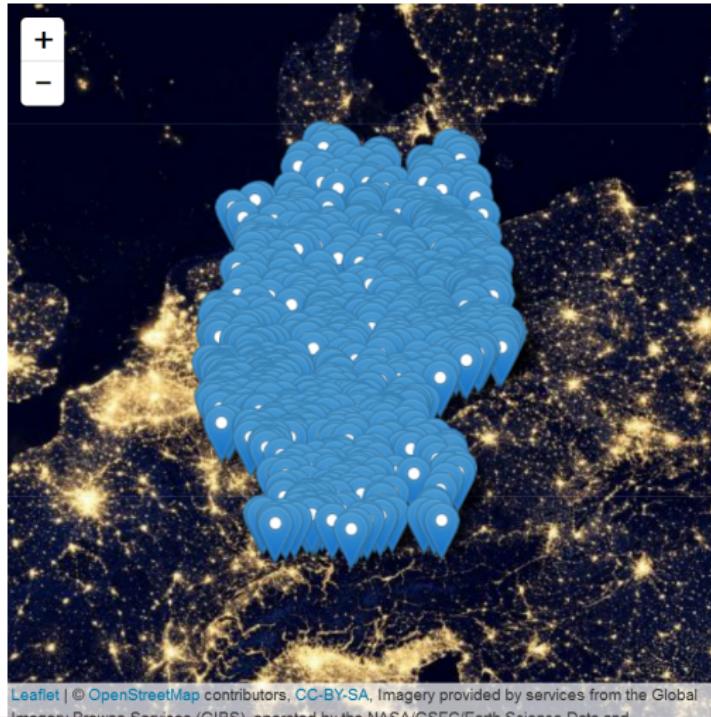
CartoDB als Hintergrund

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



Mehr Hintergründe

```
m %>% addProviderTiles("NASAGIBS.ViirsEarthAtNight2012")
```



Leaflet | © OpenStreetMap contributors, CC-BY-SA, Imagery provided by services from the Global Imagery Browse Services (GIBS) operated by the NASA GSFC Earth Science Data and

Mehr Informationen hinzufügen

```
popupInfo <- paste(CampSites$name, "\n", CampSites$website)
```

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

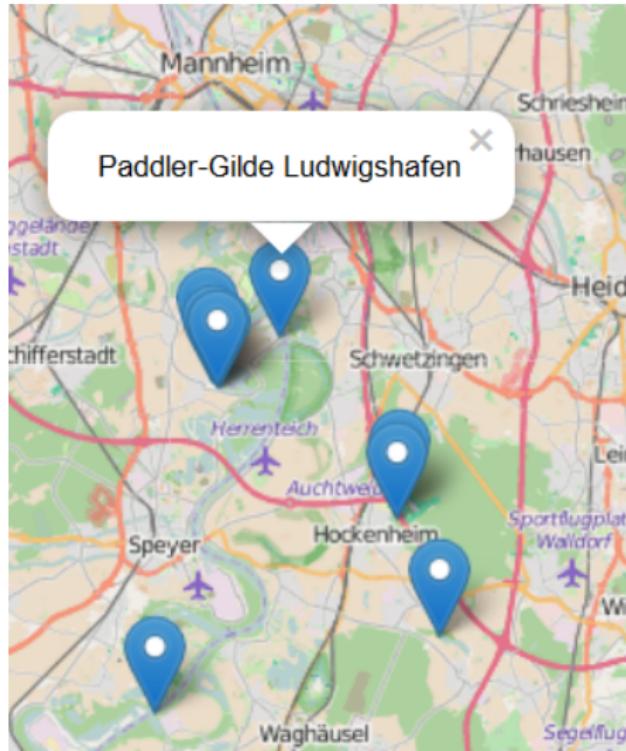
Das Ergebnis ist hier:

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

Die resultierende Karte

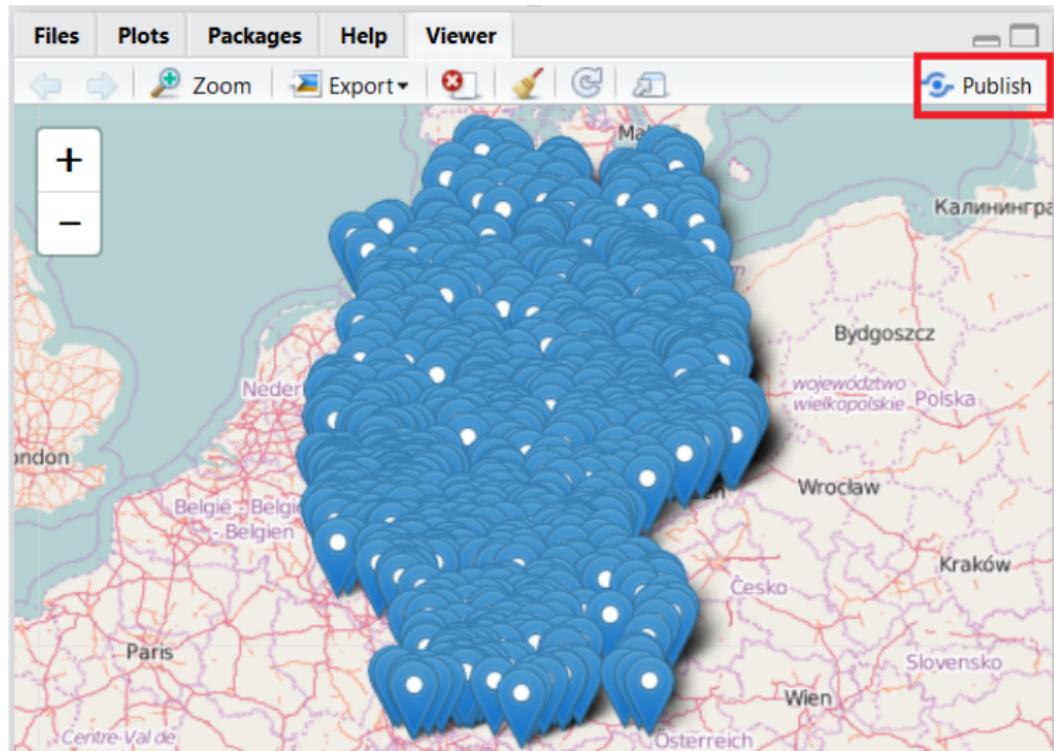


Popups in einer interaktiven Karte



Ich hab die Ergebnisse hochgeladen:

Wie man auf Rpubs publizieren kann



Ein weiteres Beispiel - Weltkulturerbe

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

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



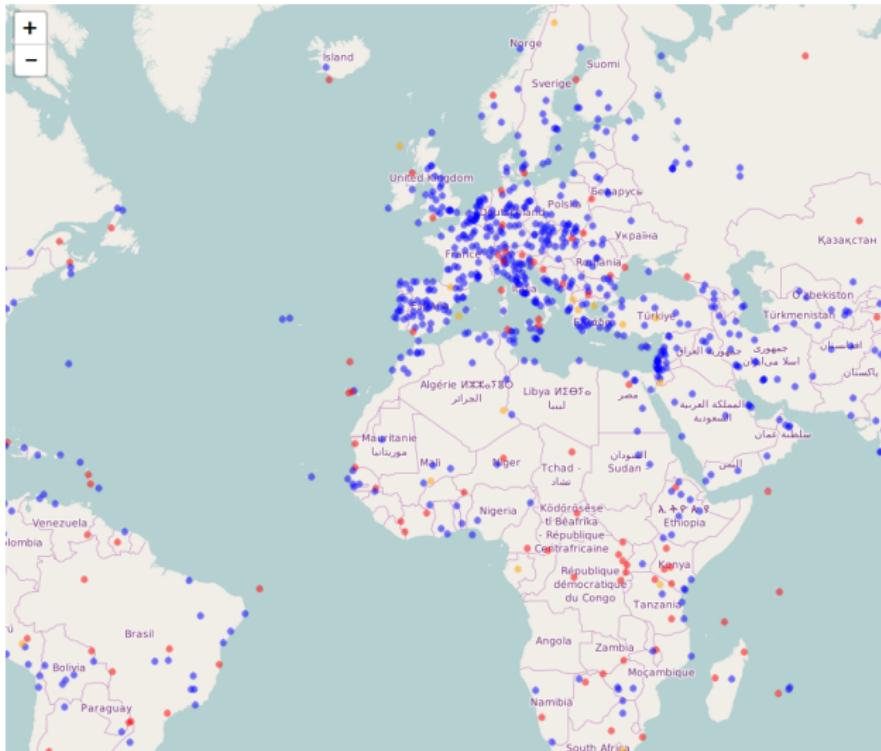
Farbe hinzufügen

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

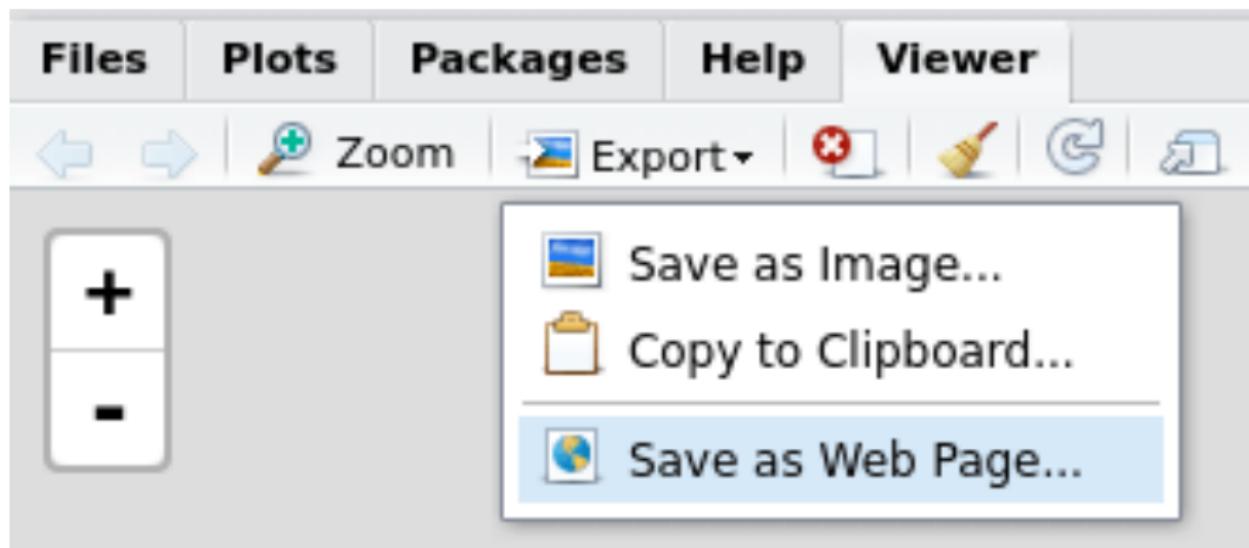
Eine Karte mit Farbe erzeugen

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

Die Karte zeigen



Die Karte abspeichern



Das Paket `mapview` - Beispieldatensatz Franken

```
library(mapview)
```

```
mapview(franconia)
```



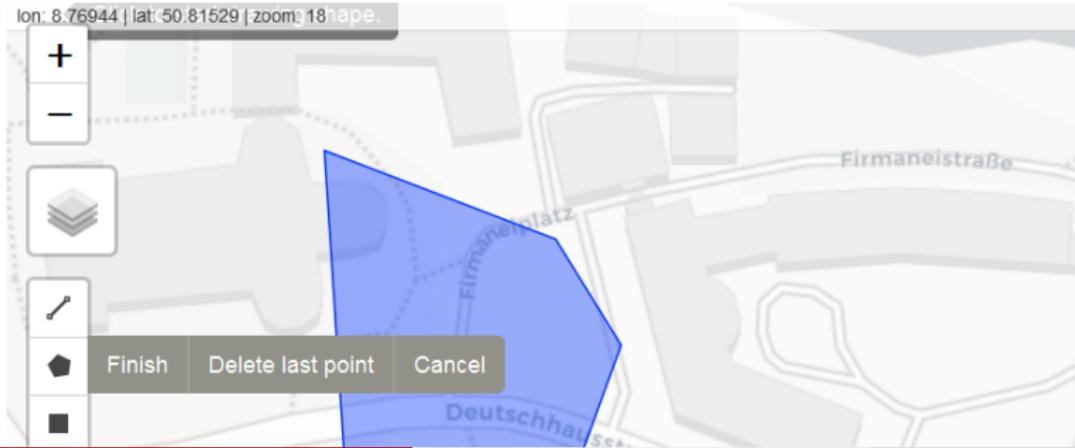
GADM und mapview

```
mapview(leaflet::gadmCHE)
```

Das Paket mapview

```
library(mapedit)  
library(magrittr)
```

```
lf <- mapview()  
drawing <- lf %>%  
  editMap()
```



Das Paket mapview

```
load("../data/spatsamp_68239.RData")
```

```
library(mapview)  
mapview(spatsamp)
```

Das Paket leaflet

```
library("tmaptools")
gc_tma <- geocode_OSM("Mannheim, GESIS")
```

```
library(leaflet)
library(magrittr)
m <- leaflet() %>%
  addTiles() %>%
  addMarkers(lng=8.463061 , lat=49.485736 , popup="GESIS Mannhei
m
```

Das Paket geojsonR

- JavaScript Object Notation

```
install.packages("geojsonR")
citation("geojsonR")
```

Wo bekomme ich ein geojson

- Ein **OSM map feature** heraus suchen
- z.B. key=highway, value=bus_stop
- Auf **Overpass Turbo** gehen und das Objekt herunterladen

```
bus_stops <- geojsonio::geojson_read("../data/Amsterdam_bus_st
what = "sp")
```

Die Punkte plotten

```
sp::plot(bus_stops)
```



Das Paket lawn

- **Vignette** für das Paket lawn
- Mit dem Paket `lawn` kann die Javascript-Bibliothek `turf.js` in R eingebunden werden.
- Weitere genutzte Javascript Bibliotheken (`geojson-random` und `geojsonhint`), werden verwendet um GeoJSON-Objekte zufällig zu erzeugen bzw. um die GeoJSON Objekte einzufärben.

```
install.packages("lawn")
citation("lawn")
```

```
library(lawn)
```

Ein zufälliges Beispiel Objekt erstellen

- Mit der Funktion gr_polygon kann ein Beispielobjekt erzeugt werden.
- Anschließend kann man sich das Objekt mit der generischen Funktion view plotten.

```
a <- gr_polygon(n = 1, vertices = 5, max_radial_length = 5)
view(a)
```

```
b <- gr_polygon(n = 1)
view(b)
```

Interaktive Deutschland Karte

```
gcs <- geojsonio::geojson_read("../data/ddat.geojson")
view(gcs)
```

Das Paket jsonlite

- A Robust, High Performance JSON Parser and Generator for R

```
library(jsonlite)
geoc <- read_json("../data/ddat.geojson")
```

```
citation("jsonlite")
```

```
##
## To cite jsonlite in publications use:
##
##   Jeroen Ooms (2014). The jsonlite Package: A Practical and
##   Consistent Mapping Between JSON Data and R Objects.
##   arXiv:1403.2805 [stat.CO] URL https://arxiv.org/abs/1403.2805
##
## A BibTeX entry for LaTeX users is
##
```

Das Paket RJSONIO

```
library("RJSONIO")
con <- url("http://nominatim.openstreetmap.org/search?format=json&addressdetails=1&extratags=1&q=Amsterdam+Niederlande+Rozengracht")
geoc <- fromJSON(paste(readLines(con, warn=F),
                      collapse = ''))
close(con)
```

Links und Quellen

- Rbloggers Artikel zu Leaflet
- Einführung in Leaflet für R
- Offline Karten mit RgoogleMaps und leaflet
- github Ordner für das lwan Paket