

# Shine on R – Geospatial data processing the /Ahh/R way

Stefan Kütke, [Stefan.kuethe@manserv.de](mailto:Stefan.kuethe@manserv.de), [stefan@fdkii.de](mailto:stefan@fdkii.de)

- “R is not just a statistics package, it’s a language.
- R is designed to operate the way that problems are thought about.
- R is both flexible and powerful.” (*Patrick Burns*)
- Furthermore, R offers a lot of geospatial packages and a simple way to write web apps based on leaflet.

Dieser „Talk“ gibt einen Überblick über die Möglichkeiten der Spatialdatenverarbeitung mit der Open Source Programmiersprache R. Dabei wird zuerst eine kurze Einführung in R (vor allem die Art und Weise wie R mit Daten umgeht) gegeben. Danach werden die gängigsten R-Pakete zum Umgang mit Spatialdaten anhand von Beispielen vorgestellt. Abschließend wird die Verwendung von Javascript Bibliotheken wie *leaflet* in R zur Implementierung interaktiver (Geo)-Webapplikationen in wenigen Zeilen Code und die Möglichkeiten der Einbindung des R-Ökosystems in QGIS und *docker*-Umgebungen gezeigt.

R wurde vor einigen Dekaden von Statistikern entwickelt, um in erster Linie statistische Fragestellungen zu bearbeiten bzw. zu lösen. Dabei ist R nicht nur ein „Statistik-Werkzeug“, sondern eine flexible und mächtige Programmiersprache, die auf allen gängigen Betriebssystem (Unix, OS-X, Windows) verfügbar ist und durch eine stetig wachsende Anzahl von Paketen aus allen Bereichen erweitert werden kann. Diese werden in der Regel über *das Comprehensive R Archive Network (CRAN)*<sup>1</sup> zur Verfügung gestellt. So gibt es auch eine Vielzahl von Paketen, um Spatialdaten in R zu lesen, zu analysieren, zu manipulieren, zu visualisieren und zu schreiben. Durch die Pakete *rgdal* und das sich in aktiver Entwicklung befindende *sf*<sup>2</sup> kann R alle von *gdal* unterstützten Formate lesen und schreiben und bietet darüber hinaus R-Datenstrukturen an, die ein R-typisches und flexibles Datahandling ermöglichen. Weiterhin besitzt R wie Python eine interaktive Konsole, in der man mit wenigen Befehlen beliebige (Geo-)Daten direkt einlesen, bearbeiten und visualisieren kann. So liest folgendes Beispiel das Shapefile *world.shp* ein, extrahiert alle Features mit Attribut *CONTINENT == Africa* und visualisiert die Ausgabe in einem Plotfenster:

```
#> shape <- st_read(„world.shp“)
#> africa <- shape[shape$CONTINENT == „Africa“, ]
#> plot(africa)
```

Die meisten Pakete zur Spatialdatenverarbeitung in R hängen bisher von dem Paket *sp* ab, in dem unter anderem R-Strukturen für die verschiedenen Geometriotypen (Line, Polygon et cetera) implementiert sind. Mit dem Paket *rgeos* steht darüber hinaus in R ein Wrapper für die *GEOS* Bibliothek zur Verfügung. Weitere Pakete wie *gdalUtils* und *rmaphaper* greifen auf die „command line tools“ von *ogr* und *mapshaper* zu. Da das Paket *sp* schon ein wenig „in die Jahre gekommen ist“ und bisher in R noch keine wirkliche Unterstützung für *simple features* oder *simple feature access* vorhanden ist, befindet sich das Paket *sf* (vom gleichen Autor wie *sp*) in aktiver Entwicklung, um einerseits diese Lücke zu schließen, einige der bisher in den anderen Paketen zur

---

<sup>1</sup> <https://cran.r-project.org/>

<sup>2</sup> <https://github.com/edzer/sf>

Verfügung stehenden Methoden performanter zu gestalten sowie die Basis-Datenstruktur *data.frame* von R für die Repräsentation der Spatialobjekte zu nutzen. In Zukunft könnte *sp* durch *sf* abgelöst werden. Bisher besteht die Möglichkeiten die Datenstrukturen in beide Richtungen zu konvertieren.

Auch können R-Skripte direkt (analog zu Python) in QGIS eingebunden werden, so dass in QGIS auf die statistischen Algorithmen von R zugegriffen werden kann. Auf der anderen Seite besteht mit Hilfe des Pakets *rqgis* die Möglichkeit aus R direkt auf in QGIS verfügbare Algorithmen zuzugreifen.

Vor allen in den letzten Jahren wurde R um viele „Features“ bzw. Pakete und die Open Source Entwicklungsumgebung *RStudio*<sup>3</sup> erweitert, die zum einen die Arbeit mit R erleichtern und zum anderen die Möglichkeit bieten in wenigen Zeilen Code flexible Web-Applikationen (R-Paket *shiny*) zu implementieren und zu hosten (*Shiny Server*).

Mit dem Paket *htmlwidgets*<sup>4</sup> können javascript-Bibliotheken „gewrappt“ werden, wodurch diese direkt von der R-Konsole aus bzw. in *RStudio* angesprochen und zur interaktiven Visualisierung in R genutzt werden können. So lassen sich durch das R-Paket *leaflet* beispielsweise auf einfache Art und Weise interaktive Karten basierend auf dem *leaflet*-javascript-Framework erzeugen. Folgende Codezeilen erzeugen beispielsweise eine interaktive Webapplikation, die in einem *leaflet*-Layer Erdbeben aus dem in R enthaltenen *quakes* Datensatz darstellen:

```
# view <- fluidPage(  
#   h1("quakes"),  
#   leafletOutput("map")  
#)  
# controller <- function(input, output){  
#   output$map <- renderLeaflet(  
#     leaflet(quakes) %>% addTiles() %>%  
#     addMarkers(  
#       data = quakes,  
#       popup = ~ as.character(mag),  
#       clusterOptions = markerClusterOptions()  
#     )  
#   )  
# }  
app <- shinyApp(view, controller)
```

---

<sup>3</sup> <https://www.rstudio.com/>

<sup>4</sup> <http://www.htmlwidgets.org/>

Darüber hinaus können sowohl *RStudio* als auch der *Shiny* Server problemlos in *docker*-Containern<sup>5</sup> betrieben werden, so dass z. B. auch ein „R-Spatial-Data-Processing“- Container inklusive Entwicklungsumgebung „gebaut“ werden kann.

Da es sich sowohl bei R als auch bei der für die Spatialdatenverarbeitung weit verbreiteten Programmiersprache Python, die ebenfalls auf allen gängigen Betriebssystemen zur Verfügung steht, um Skriptsprachen handelt, lassen sich diese sehr gut in Skripten miteinander verknüpfen und so für automatisierte Prozesse nutzen. Dieser Ansatz findet sich auch indirekt in der Open Source Data Science Plattform *Anaconda*<sup>6</sup> wieder. Mit Hilfe von *rmarkdown*<sup>7</sup> kann R auch als flexibles automatisierbares Reporting-Tool eingesetzt werden.

Zusammenfassend lässt sich sagen, dass R vielseitige Möglichkeiten zur Spatialdatenverarbeitung und -visualisierung bietet, die durch eine Verknüpfung mit Python und anderen Skriptsprachen wie *nodejs*, noch ausgeweitet werden können.

Übersicht gängiger R-Packages zur Spatialdatenverarbeitung und –visualisierung, die in dem „Talk“ besprochen werden:

- *sf*
- *sp*
- *rgdal*
- *raster*
- *rgeos*
- *maptools*
- *gdalUtils*
- *rmapshaper*
- *spatstat*
- *rqgis*
- *leaflet*

---

<sup>5</sup> <https://hub.docker.com/r/rocker/rstudio/> und <https://hub.docker.com/r/rocker/shiny/>

<sup>6</sup> <https://www.continuum.io>

<sup>7</sup> <http://rmarkdown.rstudio.com/>