

R Schnittstellen - Internetressourcen und Schnittstellen nutzen

Jan-Philipp Kolb

9 Mai 2017

Internetressourcen und Schnittstellen nutzen

Was sind API's?

Eine Programmierschnittstelle, genauer Schnittstelle zur Anwendungsprogrammierung, häufig nur kurz **API** genannt (englisch application programming interface, wörtlich ‚Anwendungsprogrammierschnittstelle‘), ist ein Programmteil, der von einem Softwaresystem anderen Programmen zur Anbindung an das System zur Verfügung ...

Programmierschnittstelle – Wikipedia
<https://de.wikipedia.org/wiki/Programmierschnittstelle>

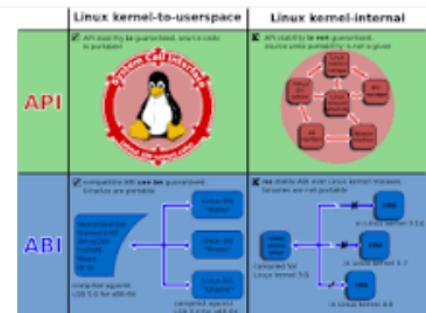


Figure 1:

Programmierschnittstellen

- Relevanz hat der Begriff der API vor allem durch seine Verwendung von Webdiensten erhalten.
- APIs dienen also zum Austausch und der Weiterverarbeitung von Daten und Inhalten zwischen verschiedenen Webseiten, Programmen und Content-Anbietern.

Beispiel

Über die YouTube-API haben Entwickler die Möglichkeit, nach Videos mit gewünschten Parametern, wie zum Beispiel Name oder Länge zu suchen. Die API schickt die Antwort in Form einer XML-Datei zurück. Diese kann dann nach einer Auswertung für die eigene Webseite verwendet werden.

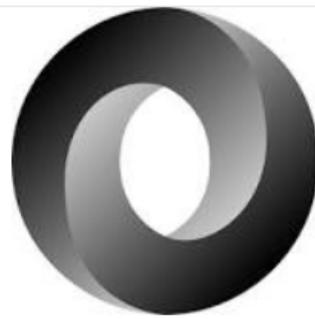
Figure 2:

Bedeutung

- es ist unter anderem möglich Daten von Programmierschnittstellen zu beziehen
- diese Daten sind allerdings nicht als .xlsx, .csv, .dta oder ähnliches abgespeichert sondern in einem der folgenden Formate: .json, .xml etc.

JavaScript Object Notation

Die JavaScript Object Notation, kurz **JSON** [ˈdʒeɪson], ist ein kompaktes Datenformat in einer einfach lesbaren Textform zum Zweck des Datenaustauschs zwischen Anwendungen. Jedes gültige **JSON**-Dokument soll ein gültiges JavaScript sein und per eval() interpretiert werden können.



JSON - Wikipedia

https://de.wikipedia.org/wiki/JavaScript_Object_Notation

Figure 3:

Das GeoJSON Format

- GeoJSON ist ein offenes Format um geografische Daten nach der Simple-Feature-Access-Spezifikation zu repräsentieren.
- Dafür wird die JavaScript Object Notation verwendet.

Die Struktur der Daten kann man sich mit einem JSON Viewer anschauen

GeoJSON

- GeoJSON ist ein offenes Format um geografische Daten nach der Simple-Feature-Access-Spezifikation zu repräsentieren.
- Simple Feature Access ist eine Spezifikation des Open Geospatial Consortium, welche eine allgemein gültige Architektur für geografische Daten und deren Geometrien definiert.
- Die Spezifikation beschreibt einerseits die Speicherung und den Zugriff auf Geometrien und andererseits verschiedene räumliche Operatoren.



Figure 4:

Leaflet for R

[Introduction](#)

[The Map Widget](#)

[Basemaps](#)

Working with GeoJSON & TopoJSON

For working with GeoJSON/TopoJSON data you have two options: either read it into `sp` objects; or use the `addGeoJSON()` and `addTopoJSON()` functions.

Figure 5:

Einfache Geometrien

Typ	Beispiel	
Point (Punkt)		<pre>{ "type": "Point", "coordinates": [30, 10] }</pre>
LineString (Linie)		<pre>{ "type": "LineString", "coordinates": [[30, 10], [10, 30], [40, 40]] }</pre>
Polygon		<pre>{ "type": "Polygon", "coordinates": [[[30, 10], [40, 40], [20, 40], [10, 20], [30, 10]]] }</pre>
		<pre>{ "type": "Polygon", "coordinates": [[[35, 10], [45, 45], [15, 40], [10, 20], [35, 10]], [[20, 30], [35, 35], [30, 20], [20, 30]]] }</pre>

Figure 6:

OpenStreetMap Daten

Beispiele für GeoJSON

Import von JSON-Objekten und XML Dateien

Import von JavaScript Object Notation (JSON)

- Jedes gültige JSON-Dokument soll ein gültiges JavaScript sein
- JSON wird zur Übertragung und zum Speichern von strukturierten Daten eingesetzt
- Insbesondere bei Webanwendungen und mobilen Apps wird es in Verbindung mit JavaScript, Ajax oder WebSockets zum Transfer von Daten zwischen dem Client und dem Server häufig genutzt.

```
{  
    "Herausgeber": "Xema",  
    "Nummer": "1234-5678-9012-3456",  
    "Deckung": 2e+6,  
    "Waehrung": "EURO",  
    "Inhaber":  
    {  
        "Name": "Mustermann",  
        "Vorname": "Max",  
        "maennlich": true,  
        "Hobbies": [ "Reiten", "Golfen", "Lesen" ],  
        "Alter": 42,  
        "Kinder": [],  
        "Partner": null  
    }  
}
```

Figure 7:

Download von Beispieldaten

- Overpass Turbo kann verwendet werden um Beispieldaten zu bekommen

<https://overpass-turbo.eu/>

Exkurs OpenStreetMap Daten

- Auf Overpass Turbo können Daten für Map Features exportiert werden
- Eine Liste der erhältlichen Map Features gibt es auf http://wiki.openstreetmap.org/wiki/DE:Map_Features

The screenshot shows a JSON viewer interface with the URL jsonviewer.stack.hu. The JSON structure is as follows:

```

{
  "type": "FeatureCollection",
  "generator": "overpass-turbo",
  "copyright": "The data included in this document is from www.openstreetmap.org. The data is made available under ODbL.",
  "timestamp": "2017-04-23T08:54:59Z",
  "features": [
    {
      "type": "Feature",
      "id": "node/246574149",
      "properties": {
        "@id": "node/246574149",
        "amenity": "drinking_water",
        "flow": "push-button",
        "type": "nasone"
      },
      "geometry": {
        "type": "Point"
      }
    }
  ]
}

```

Figure 8:

The screenshot shows the Overpass Turbo interface with the URL <https://overpass-turbo.eu>. The interface includes a toolbar with buttons for Ausführen, Teilen, Export, Wizard, Speichern, Laden, Einstellungen, and Hilfe. A search bar and a "Flatt this" button are also present. The main area has two panes: "Karte" (Map) and "Daten" (Data). The "Daten" pane displays a map of a city area with several blue circles highlighting locations. The "Karte" pane shows the same map with street names like "Viale Cesare Ceradini", "Viale del Monte Oppio", "Parco del Colle Oppio", and "Viale Serapide". The "Daten" pane also contains a code editor with the following Overpass query:

```

/*
This is an example Overpass query.
Try it out by pressing the Run button above!
You can find more examples with the Load tool.
*/
node
[amenity=drinking_water]
((bbox));
out;

```

Figure 9:

Schlüssel	Wert	Element	Kommentar	Darstellung	Foto
Verpflegung, Einkehr					
amenity	bar	<input type="radio"/> <input checked="" type="checkbox"/>	Bar, Nachtlokal . Es werden hauptsächlich alkoholische Getränke serviert. Siehe auch Beschreibung von amenity=bar und amenity=pub zur Unterscheidung von Bar und Pub (Kneipe).		
amenity	bbq	<input type="radio"/>	Grillplatz. Kann mit fuel=* (wood/gas/electric) kombiniert werden.		
amenity	biergarten	<input type="radio"/> <input checked="" type="checkbox"/>	Biergarten		

Figure 10:

Das Paket jsonlite

```
install.packages("jsonlite")

library(jsonlite)
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
##
## @Article{,
##   title = {The jsonlite Package: A Practical and Consistent Mapping Between JSON Data and R Objects},
##   author = {Jeroen Ooms},
##   journal = {arXiv:1403.2805 [stat.CO]},
##   year = {2014},
##   url = {https://arxiv.org/abs/1403.2805},
## }
```

JSON importieren

- die Daten für die Trinkstationen in Rom habe ich mit Overpass Turbo exportiert

```
library("jsonlite")
DRINKWATER <- fromJSON("data/RomDrinkingWater.geojson")

names(DRINKWATER)[1:3]

## [1] "type"      "generator"  "copyright"
names(DRINKWATER)[4:5]

## [1] "timestamp" "features"
```

Die Daten anschauen

```
head(DRINKWATER$features)

##      type          id properties.@id properties.amenity properties.flow
## 1 Feature node/246574149 node/246574149    drinking_water    push-button
## 2 Feature node/246574150 node/246574150    drinking_water           <NA>
## 3 Feature node/246574151 node/246574151    drinking_water           <NA>
## 4 Feature node/248743324 node/248743324    drinking_water           <NA>
## 5 Feature node/251773348 node/251773348    drinking_water           <NA>
## 6 Feature node/251773551 node/251773551    drinking_water           <NA>
## properties.type properties.name properties.name:fr properties.wheelchair
## 1           nasone           <NA>           <NA>           <NA>
## 2           <NA>           <NA>           <NA>           <NA>
## 3           <NA>           <NA>           <NA>           <NA>
```

```

## 4 <NA> <NA> <NA> <NA>
## 5 nasone <NA> <NA> <NA> <NA>
## 6 <NA> Acqua Marcia Eau potable yes
## properties.created_by properties.indoor geometry.type
## 1 <NA> <NA> Point
## 2 <NA> <NA> Point
## 3 <NA> <NA> Point
## 4 <NA> <NA> Point
## 5 <NA> <NA> Point
## 6 <NA> <NA> Point
## geometry.coordinates
## 1 12.49191, 41.89479
## 2 12.49095, 41.89489
## 3 12.48774, 41.89450
## 4 12.48773, 41.89354
## 5 12.48529, 41.88539
## 6 12.48386, 41.89332

```

Github JSON Daten

- Es lassen sich auch Dinge aus dem Web auslesen:

```

my_repos <- fromJSON("https://api.github.com/users/japhilko/repos")

head(my_repos)

##      id          name           full_name
## 1 29143362 2015-01-15-EMBLHeidelberg Japhilko/2015-01-15-EMBLHeidelberg
## 2 39427013          DataAnalysis       Japhilko/DataAnalysis
## 3 26485588        DataGeneration       Japhilko/DataGeneration
## 4 26164276          DLR_IntroR       Japhilko/DLR_IntroR
## 5 20760765          GeoData       Japhilko/GeoData
## 6 55756271         geosmdata       Japhilko/geosmdata
##   owner.login owner.id
## 1     Japhilko    7593396
## 2     Japhilko    7593396
## 3     Japhilko    7593396
## 4     Japhilko    7593396
## 5     Japhilko    7593396
## 6     Japhilko    7593396
##                               owner.avatar_url owner.gravatar_id
## 1 https://avatars2.githubusercontent.com/u/7593396?v=3
## 2 https://avatars2.githubusercontent.com/u/7593396?v=3
## 3 https://avatars2.githubusercontent.com/u/7593396?v=3
## 4 https://avatars2.githubusercontent.com/u/7593396?v=3
## 5 https://avatars2.githubusercontent.com/u/7593396?v=3
## 6 https://avatars2.githubusercontent.com/u/7593396?v=3
##                               owner.url          owner.html_url
## 1 https://api.github.com/users/Japhilko https://github.com/Japhilko
## 2 https://api.github.com/users/Japhilko https://github.com/Japhilko
## 3 https://api.github.com/users/Japhilko https://github.com/Japhilko
## 4 https://api.github.com/users/Japhilko https://github.com/Japhilko
## 5 https://api.github.com/users/Japhilko https://github.com/Japhilko
## 6 https://api.github.com/users/Japhilko https://github.com/Japhilko

```

```

##                               owner.followers_url
## 1 https://api.github.com/users/Japhilko/followers
## 2 https://api.github.com/users/Japhilko/followers
## 3 https://api.github.com/users/Japhilko/followers
## 4 https://api.github.com/users/Japhilko/followers
## 5 https://api.github.com/users/Japhilko/followers
## 6 https://api.github.com/users/Japhilko/followers
##                               owner.following_url
## 1 https://api.github.com/users/Japhilko/following{/other_user}
## 2 https://api.github.com/users/Japhilko/following{/other_user}
## 3 https://api.github.com/users/Japhilko/following{/other_user}
## 4 https://api.github.com/users/Japhilko/following{/other_user}
## 5 https://api.github.com/users/Japhilko/following{/other_user}
## 6 https://api.github.com/users/Japhilko/following{/other_user}
##                               owner.gists_url
## 1 https://api.github.com/users/Japhilko/gists{/gist_id}
## 2 https://api.github.com/users/Japhilko/gists{/gist_id}
## 3 https://api.github.com/users/Japhilko/gists{/gist_id}
## 4 https://api.github.com/users/Japhilko/gists{/gist_id}
## 5 https://api.github.com/users/Japhilko/gists{/gist_id}
## 6 https://api.github.com/users/Japhilko/gists{/gist_id}
##                               owner.starred_url
## 1 https://api.github.com/users/Japhilko/starred{/owner}{/repo}
## 2 https://api.github.com/users/Japhilko/starred{/owner}{/repo}
## 3 https://api.github.com/users/Japhilko/starred{/owner}{/repo}
## 4 https://api.github.com/users/Japhilko/starred{/owner}{/repo}
## 5 https://api.github.com/users/Japhilko/starred{/owner}{/repo}
## 6 https://api.github.com/users/Japhilko/starred{/owner}{/repo}
##                               owner.subscriptions_url
## 1 https://api.github.com/users/Japhilko/subscriptions
## 2 https://api.github.com/users/Japhilko/subscriptions
## 3 https://api.github.com/users/Japhilko/subscriptions
## 4 https://api.github.com/users/Japhilko/subscriptions
## 5 https://api.github.com/users/Japhilko/subscriptions
## 6 https://api.github.com/users/Japhilko/subscriptions
##                               owner.organizations_url
## 1 https://api.github.com/users/Japhilko/orgs
## 2 https://api.github.com/users/Japhilko/orgs
## 3 https://api.github.com/users/Japhilko/orgs
## 4 https://api.github.com/users/Japhilko/orgs
## 5 https://api.github.com/users/Japhilko/orgs
## 6 https://api.github.com/users/Japhilko/orgs
##                               owner.repos_url
## 1 https://api.github.com/users/Japhilko/repos
## 2 https://api.github.com/users/Japhilko/repos
## 3 https://api.github.com/users/Japhilko/repos
## 4 https://api.github.com/users/Japhilko/repos
## 5 https://api.github.com/users/Japhilko/repos
## 6 https://api.github.com/users/Japhilko/repos
##                               owner.events_url
## 1 https://api.github.com/users/Japhilko/events{/privacy}
## 2 https://api.github.com/users/Japhilko/events{/privacy}
## 3 https://api.github.com/users/Japhilko/events{/privacy}
## 4 https://api.github.com/users/Japhilko/events{/privacy}

```

```

## 5 https://api.github.com/users/Japhilko/events{/privacy}
## 6 https://api.github.com/users/Japhilko/events{/privacy}
##           owner.received_events_url owner.type
## 1 https://api.github.com/users/Japhilko/received_events      User
## 2 https://api.github.com/users/Japhilko/received_events      User
## 3 https://api.github.com/users/Japhilko/received_events      User
## 4 https://api.github.com/users/Japhilko/received_events      User
## 5 https://api.github.com/users/Japhilko/received_events      User
## 6 https://api.github.com/users/Japhilko/received_events      User
##   owner.site_admin private
## 1          FALSE  FALSE
## 2          FALSE  FALSE
## 3          FALSE  FALSE
## 4          FALSE  FALSE
## 5          FALSE  FALSE
## 6          FALSE  FALSE
##           html_url
## 1 https://github.com/Japhilko/2015-01-15-EMBLHeidelberg
## 2           https://github.com/Japhilko/DataAnalysis
## 3           https://github.com/Japhilko/DataGeneration
## 4           https://github.com/Japhilko/DLR_IntroR
## 5           https://github.com/Japhilko/GeoData
## 6           https://github.com/Japhilko/geosmdata
##           description fork
## 1 R programming and development (EMBL, Jan 2015)  TRUE
## 2           My research on data analysis FALSE
## 3           Rcode for generating synthatic data FALSE
## 4           Unterlagen für DLR Workshop FALSE
## 5           Research on statistics and geodata FALSE
## 6           package to import OpenstreetMap data FALSE
##           url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg
## 2           https://api.github.com/repos/Japhilko/DataAnalysis
## 3           https://api.github.com/repos/Japhilko/DataGeneration
## 4           https://api.github.com/repos/Japhilko/DLR_IntroR
## 5           https://api.github.com/repos/Japhilko/GeoData
## 6           https://api.github.com/repos/Japhilko/geosmdata
##           forks_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/forks
## 2           https://api.github.com/repos/Japhilko/DataAnalysis/forks
## 3           https://api.github.com/repos/Japhilko/DataGeneration/forks
## 4           https://api.github.com/repos/Japhilko/DLR_IntroR/forks
## 5           https://api.github.com/repos/Japhilko/GeoData/forks
## 6           https://api.github.com/repos/Japhilko/geosmdata/forks
##           keys_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/keys{/key_id}
## 2           https://api.github.com/repos/Japhilko/DataAnalysis/keys{/key_id}
## 3           https://api.github.com/repos/Japhilko/DataGeneration/keys{/key_id}
## 4           https://api.github.com/repos/Japhilko/DLR_IntroR/keys{/key_id}
## 5           https://api.github.com/repos/Japhilko/GeoData/keys{/key_id}
## 6           https://api.github.com/repos/Japhilko/geosmdata/keys{/key_id}
##           collaborators_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/collaborators{/collaborator}
## 2           https://api.github.com/repos/Japhilko/DataAnalysis/collaborators{/collaborator}

```

```

## 3      https://api.github.com/repos/Japhilko/DataGeneration/collaborators{/collaborator}
## 4      https://api.github.com/repos/Japhilko/DLR_IntroR/collaborators{/collaborator}
## 5      https://api.github.com/repos/Japhilko/GeoData/collaborators{/collaborator}
## 6      https://api.github.com/repos/Japhilko/geosmdata/collaborators{/collaborator}
##
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/teams
## 2      https://api.github.com/repos/Japhilko/DataAnalysis/teams
## 3      https://api.github.com/repos/Japhilko/DataGeneration/teams
## 4      https://api.github.com/repos/Japhilko/DLR_IntroR/teams
## 5      https://api.github.com/repos/Japhilko/GeoData/teams
## 6      https://api.github.com/repos/Japhilko/geosmdata/teams
##
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/hooks
## 2      https://api.github.com/repos/Japhilko/DataAnalysis/hooks
## 3      https://api.github.com/repos/Japhilko/DataGeneration/hooks
## 4      https://api.github.com/repos/Japhilko/DLR_IntroR/hooks
## 5      https://api.github.com/repos/Japhilko/GeoData/hooks
## 6      https://api.github.com/repos/Japhilko/geosmdata/hooks
##
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/issues/events{/number}
## 2      https://api.github.com/repos/Japhilko/DataAnalysis/issues/events{/number}
## 3      https://api.github.com/repos/Japhilko/DataGeneration/issues/events{/number}
## 4      https://api.github.com/repos/Japhilko/DLR_IntroR/issues/events{/number}
## 5      https://api.github.com/repos/Japhilko/GeoData/issues/events{/number}
## 6      https://api.github.com/repos/Japhilko/geosmdata/issues/events{/number}
##
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/events
## 2      https://api.github.com/repos/Japhilko/DataAnalysis/events
## 3      https://api.github.com/repos/Japhilko/DataGeneration/events
## 4      https://api.github.com/repos/Japhilko/DLR_IntroR/events
## 5      https://api.github.com/repos/Japhilko/GeoData/events
## 6      https://api.github.com/repos/Japhilko/geosmdata/events
##
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/assignees{/user}
## 2      https://api.github.com/repos/Japhilko/DataAnalysis/assignees{/user}
## 3      https://api.github.com/repos/Japhilko/DataGeneration/assignees{/user}
## 4      https://api.github.com/repos/Japhilko/DLR_IntroR/assignees{/user}
## 5      https://api.github.com/repos/Japhilko/GeoData/assignees{/user}
## 6      https://api.github.com/repos/Japhilko/geosmdata/assignees{/user}
##
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/branches{/branch}
## 2      https://api.github.com/repos/Japhilko/DataAnalysis/branches{/branch}
## 3      https://api.github.com/repos/Japhilko/DataGeneration/branches{/branch}
## 4      https://api.github.com/repos/Japhilko/DLR_IntroR/branches{/branch}
## 5      https://api.github.com/repos/Japhilko/GeoData/branches{/branch}
## 6      https://api.github.com/repos/Japhilko/geosmdata/branches{/branch}
##
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/tags
## 2      https://api.github.com/repos/Japhilko/DataAnalysis/tags
## 3      https://api.github.com/repos/Japhilko/DataGeneration/tags
## 4      https://api.github.com/repos/Japhilko/DLR_IntroR/tags
## 5      https://api.github.com/repos/Japhilko/GeoData/tags
## 6      https://api.github.com/repos/Japhilko/geosmdata/tags
##
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/blobs{/blob}

```

```

## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/git/blobs{/sha}
## 2           https://api.github.com/repos/Japhilko/DataAnalysis/git/blobs{/sha}
## 3           https://api.github.com/repos/Japhilko/DataGeneration/git/blobs{/sha}
## 4           https://api.github.com/repos/Japhilko/DLR_IntroR/git/blobs{/sha}
## 5           https://api.github.com/repos/Japhilko/GeoData/git/blobs{/sha}
## 6           https://api.github.com/repos/Japhilko/geosmdata/git/blobs{/sha}
##
##                                         git_tags_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/git/tags{/sha}
## 2           https://api.github.com/repos/Japhilko/DataAnalysis/git/tags{/sha}
## 3           https://api.github.com/repos/Japhilko/DataGeneration/git/tags{/sha}
## 4           https://api.github.com/repos/Japhilko/DLR_IntroR/git/tags{/sha}
## 5           https://api.github.com/repos/Japhilko/GeoData/git/tags{/sha}
## 6           https://api.github.com/repos/Japhilko/geosmdata/git/tags{/sha}
##
##                                         git_refs_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/git/refs{/sha}
## 2           https://api.github.com/repos/Japhilko/DataAnalysis/git/refs{/sha}
## 3           https://api.github.com/repos/Japhilko/DataGeneration/git/refs{/sha}
## 4           https://api.github.com/repos/Japhilko/DLR_IntroR/git/refs{/sha}
## 5           https://api.github.com/repos/Japhilko/GeoData/git/refs{/sha}
## 6           https://api.github.com/repos/Japhilko/geosmdata/git/refs{/sha}
##
##                                         trees_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/git/trees{/sha}
## 2           https://api.github.com/repos/Japhilko/DataAnalysis/git/trees{/sha}
## 3           https://api.github.com/repos/Japhilko/DataGeneration/git/trees{/sha}
## 4           https://api.github.com/repos/Japhilko/DLR_IntroR/git/trees{/sha}
## 5           https://api.github.com/repos/Japhilko/GeoData/git/trees{/sha}
## 6           https://api.github.com/repos/Japhilko/geosmdata/git/trees{/sha}
##
##                                         statuses_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/statuses/{sha}
## 2           https://api.github.com/repos/Japhilko/DataAnalysis/statuses/{sha}
## 3           https://api.github.com/repos/Japhilko/DataGeneration/statuses/{sha}
## 4           https://api.github.com/repos/Japhilko/DLR_IntroR/statuses/{sha}
## 5           https://api.github.com/repos/Japhilko/GeoData/statuses/{sha}
## 6           https://api.github.com/repos/Japhilko/geosmdata/statuses/{sha}
##
##                                         languages_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/languages
## 2           https://api.github.com/repos/Japhilko/DataAnalysis/languages
## 3           https://api.github.com/repos/Japhilko/DataGeneration/languages
## 4           https://api.github.com/repos/Japhilko/DLR_IntroR/languages
## 5           https://api.github.com/repos/Japhilko/GeoData/languages
## 6           https://api.github.com/repos/Japhilko/geosmdata/languages
##
##                                         stargazers_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/stargazers
## 2           https://api.github.com/repos/Japhilko/DataAnalysis/stargazers
## 3           https://api.github.com/repos/Japhilko/DataGeneration/stargazers
## 4           https://api.github.com/repos/Japhilko/DLR_IntroR/stargazers
## 5           https://api.github.com/repos/Japhilko/GeoData/stargazers
## 6           https://api.github.com/repos/Japhilko/geosmdata/stargazers
##
##                                         contributors_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/contributors
## 2           https://api.github.com/repos/Japhilko/DataAnalysis/contributors
## 3           https://api.github.com/repos/Japhilko/DataGeneration/contributors
## 4           https://api.github.com/repos/Japhilko/DLR_IntroR/contributors
## 5           https://api.github.com/repos/Japhilko/GeoData/contributors

```

```

## 6 https://api.github.com/repos/Japhilko/geosmdata/contributors
##                                         subscribers_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/subscribers
## 2 https://api.github.com/repos/Japhilko/DataAnalysis/subscribers
## 3 https://api.github.com/repos/Japhilko/DataGeneration/subscribers
## 4 https://api.github.com/repos/Japhilko/DLR_IntroR/subscribers
## 5 https://api.github.com/repos/Japhilko/GeoData/subscribers
## 6 https://api.github.com/repos/Japhilko/geosmdata/subscribers
##                                         subscription_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/subscription
## 2 https://api.github.com/repos/Japhilko/DataAnalysis/subscription
## 3 https://api.github.com/repos/Japhilko/DataGeneration/subscription
## 4 https://api.github.com/repos/Japhilko/DLR_IntroR/subscription
## 5 https://api.github.com/repos/Japhilko/GeoData/subscription
## 6 https://api.github.com/repos/Japhilko/geosmdata/subscription
##                                         commits_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/commits{/sha}
## 2 https://api.github.com/repos/Japhilko/DataAnalysis/commits{/sha}
## 3 https://api.github.com/repos/Japhilko/DataGeneration/commits{/sha}
## 4 https://api.github.com/repos/Japhilko/DLR_IntroR/commits{/sha}
## 5 https://api.github.com/repos/Japhilko/GeoData/commits{/sha}
## 6 https://api.github.com/repos/Japhilko/geosmdata/commits{/sha}
##                                         git_commits_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/git/commits{/sha}
## 2 https://api.github.com/repos/Japhilko/DataAnalysis/git/commits{/sha}
## 3 https://api.github.com/repos/Japhilko/DataGeneration/git/commits{/sha}
## 4 https://api.github.com/repos/Japhilko/DLR_IntroR/git/commits{/sha}
## 5 https://api.github.com/repos/Japhilko/GeoData/git/commits{/sha}
## 6 https://api.github.com/repos/Japhilko/geosmdata/git/commits{/sha}
##                                         comments_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/comments{/number}
## 2 https://api.github.com/repos/Japhilko/DataAnalysis/comments{/number}
## 3 https://api.github.com/repos/Japhilko/DataGeneration/comments{/number}
## 4 https://api.github.com/repos/Japhilko/DLR_IntroR/comments{/number}
## 5 https://api.github.com/repos/Japhilko/GeoData/comments{/number}
## 6 https://api.github.com/repos/Japhilko/geosmdata/comments{/number}
##                                         issue_comment_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/issues/comments{/number}
## 2 https://api.github.com/repos/Japhilko/DataAnalysis/issues/comments{/number}
## 3 https://api.github.com/repos/Japhilko/DataGeneration/issues/comments{/number}
## 4 https://api.github.com/repos/Japhilko/DLR_IntroR/issues/comments{/number}
## 5 https://api.github.com/repos/Japhilko/GeoData/issues/comments{/number}
## 6 https://api.github.com/repos/Japhilko/geosmdata/issues/comments{/number}
##                                         contents_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/contents/{+path}
## 2 https://api.github.com/repos/Japhilko/DataAnalysis/contents/{+path}
## 3 https://api.github.com/repos/Japhilko/DataGeneration/contents/{+path}
## 4 https://api.github.com/repos/Japhilko/DLR_IntroR/contents/{+path}
## 5 https://api.github.com/repos/Japhilko/GeoData/contents/{+path}
## 6 https://api.github.com/repos/Japhilko/geosmdata/contents/{+path}
##                                         compare_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/compare/{base}...{head}
## 2 https://api.github.com/repos/Japhilko/DataAnalysis/compare/{base}...{head}
## 3 https://api.github.com/repos/Japhilko/DataGeneration/compare/{base}...{head}

```

```

## 4 https://api.github.com/repos/Japhilko/DLR_IntroR/compare/{base}...{head}
## 5 https://api.github.com/repos/Japhilko/GeoData/compare/{base}...{head}
## 6 https://api.github.com/repos/Japhilko/geosmdata/compare/{base}...{head}
##
## merges_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/merges
## 2 https://api.github.com/repos/Japhilko/DataAnalysis/merges
## 3 https://api.github.com/repos/Japhilko/DataGeneration/merges
## 4 https://api.github.com/repos/Japhilko/DLR_IntroR/merges
## 5 https://api.github.com/repos/Japhilko/GeoData/merges
## 6 https://api.github.com/repos/Japhilko/geosmdata/merges
##
## archive_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/{archive_format}{/ref}
## 2 https://api.github.com/repos/Japhilko/DataAnalysis/{archive_format}{/ref}
## 3 https://api.github.com/repos/Japhilko/DataGeneration/{archive_format}{/ref}
## 4 https://api.github.com/repos/Japhilko/DLR_IntroR/{archive_format}{/ref}
## 5 https://api.github.com/repos/Japhilko/GeoData/{archive_format}{/ref}
## 6 https://api.github.com/repos/Japhilko/geosmdata/{archive_format}{/ref}
##
## downloads_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/downloads
## 2 https://api.github.com/repos/Japhilko/DataAnalysis/downloads
## 3 https://api.github.com/repos/Japhilko/DataGeneration/downloads
## 4 https://api.github.com/repos/Japhilko/DLR_IntroR/downloads
## 5 https://api.github.com/repos/Japhilko/GeoData/downloads
## 6 https://api.github.com/repos/Japhilko/geosmdata/downloads
##
## issues_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/issues{/number}
## 2 https://api.github.com/repos/Japhilko/DataAnalysis/issues{/number}
## 3 https://api.github.com/repos/Japhilko/DataGeneration/issues{/number}
## 4 https://api.github.com/repos/Japhilko/DLR_IntroR/issues{/number}
## 5 https://api.github.com/repos/Japhilko/GeoData/issues{/number}
## 6 https://api.github.com/repos/Japhilko/geosmdata/issues{/number}
##
## pulls_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/pulls{/number}
## 2 https://api.github.com/repos/Japhilko/DataAnalysis/pulls{/number}
## 3 https://api.github.com/repos/Japhilko/DataGeneration/pulls{/number}
## 4 https://api.github.com/repos/Japhilko/DLR_IntroR/pulls{/number}
## 5 https://api.github.com/repos/Japhilko/GeoData/pulls{/number}
## 6 https://api.github.com/repos/Japhilko/geosmdata/pulls{/number}
##
## milestones_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/milestones{/number}
## 2 https://api.github.com/repos/Japhilko/DataAnalysis/milestones{/number}
## 3 https://api.github.com/repos/Japhilko/DataGeneration/milestones{/number}
## 4 https://api.github.com/repos/Japhilko/DLR_IntroR/milestones{/number}
## 5 https://api.github.com/repos/Japhilko/GeoData/milestones{/number}
## 6 https://api.github.com/repos/Japhilko/geosmdata/milestones{/number}
##
## notifications
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/notifications{?since,all,participants}
## 2 https://api.github.com/repos/Japhilko/DataAnalysis/notifications{?since,all,participants}
## 3 https://api.github.com/repos/Japhilko/DataGeneration/notifications{?since,all,participants}
## 4 https://api.github.com/repos/Japhilko/DLR_IntroR/notifications{?since,all,participants}
## 5 https://api.github.com/repos/Japhilko/GeoData/notifications{?since,all,participants}
## 6 https://api.github.com/repos/Japhilko/geosmdata/notifications{?since,all,participants}
##
## labels_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/labels{/name}

```

```

## 2           https://api.github.com/repos/Japhilko/DataAnalysis/labels{/name}
## 3           https://api.github.com/repos/Japhilko/DataGeneration/labels{/name}
## 4           https://api.github.com/repos/Japhilko/DLR_IntroR/labels{/name}
## 5           https://api.github.com/repos/Japhilko/GeoData/labels{/name}
## 6           https://api.github.com/repos/Japhilko/geosmdata/labels{/name}
##
##           releases_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/releases{/id}
## 2           https://api.github.com/repos/Japhilko/DataAnalysis/releases{/id}
## 3           https://api.github.com/repos/Japhilko/DataGeneration/releases{/id}
## 4           https://api.github.com/repos/Japhilko/DLR_IntroR/releases{/id}
## 5           https://api.github.com/repos/Japhilko/GeoData/releases{/id}
## 6           https://api.github.com/repos/Japhilko/geosmdata/releases{/id}
##
##           deployments_url
## 1 https://api.github.com/repos/Japhilko/2015-01-15-EMBLHeidelberg/deployments
## 2           https://api.github.com/repos/Japhilko/DataAnalysis/deployments
## 3           https://api.github.com/repos/Japhilko/DataGeneration/deployments
## 4           https://api.github.com/repos/Japhilko/DLR_IntroR/deployments
## 5           https://api.github.com/repos/Japhilko/GeoData/deployments
## 6           https://api.github.com/repos/Japhilko/geosmdata/deployments
##
##           created_at      updated_at      pushed_at
## 1 2015-01-12T15:59:33Z 2015-01-12T15:59:34Z 2015-01-10T22:26:12Z
## 2 2015-07-21T06:00:37Z 2016-02-04T13:01:54Z 2017-04-24T14:20:11Z
## 3 2014-11-11T13:14:01Z 2015-04-21T14:51:01Z 2015-07-27T13:59:39Z
## 4 2014-11-04T10:34:17Z 2016-07-26T08:22:47Z 2016-08-11T13:23:54Z
## 5 2014-06-12T08:51:41Z 2017-03-23T06:00:42Z 2017-03-23T15:31:16Z
## 6 2016-04-08T06:35:45Z 2016-06-06T10:36:01Z 2016-06-08T11:06:58Z
##
##           git_url
## 1 git://github.com/Japhilko/2015-01-15-EMBLHeidelberg.git
## 2           git://github.com/Japhilko/DataAnalysis.git
## 3           git://github.com/Japhilko/DataGeneration.git
## 4           git://github.com/Japhilko/DLR_IntroR.git
## 5           git://github.com/Japhilko/GeoData.git
## 6           git://github.com/Japhilko/geosmdata.git
##
##           ssh_url
## 1 git@github.com:Japhilko/2015-01-15-EMBLHeidelberg.git
## 2           git@github.com:Japhilko/DataAnalysis.git
## 3           git@github.com:Japhilko/DataGeneration.git
## 4           git@github.com:Japhilko/DLR_IntroR.git
## 5           git@github.com:Japhilko/GeoData.git
## 6           git@github.com:Japhilko/geosmdata.git
##
##           clone_url
## 1 https://github.com/Japhilko/2015-01-15-EMBLHeidelberg.git
## 2           https://github.com/Japhilko/DataAnalysis.git
## 3           https://github.com/Japhilko/DataGeneration.git
## 4           https://github.com/Japhilko/DLR_IntroR.git
## 5           https://github.com/Japhilko/GeoData.git
## 6           https://github.com/Japhilko/geosmdata.git
##
##           svn_url homepage    size
## 1 https://github.com/Japhilko/2015-01-15-EMBLHeidelberg <NA> 5667
## 2           https://github.com/Japhilko/DataAnalysis <NA> 55636
## 3           https://github.com/Japhilko/DataGeneration <NA> 336
## 4           https://github.com/Japhilko/DLR_IntroR <NA> 32546
## 5           https://github.com/Japhilko/GeoData <NA> 1589706
## 6           https://github.com/Japhilko/geosmdata <NA> 19931

```

```

##   stargazers_count watchers_count      language has_issues has_projects
## 1              0            0          TeX     FALSE      TRUE
## 2              0            0         HTML      TRUE      TRUE
## 3              0            0           R      TRUE      TRUE
## 4              2            2           R      TRUE      TRUE
## 5              6            6         HTML      TRUE      TRUE
## 6              0            0 ActionScript    TRUE      TRUE
##   has_downloads has_wiki has_pages forks_count mirror_url
## 1      TRUE     TRUE    FALSE            0        NA
## 2      TRUE     TRUE    FALSE            1        NA
## 3      TRUE     TRUE    TRUE            0        NA
## 4      TRUE     TRUE   FALSE            0        NA
## 5      TRUE     TRUE    TRUE            1        NA
## 6      TRUE     TRUE   FALSE            0        NA
##   open_issues_count forks open_issues watchers default_branch
## 1              0     0            0        0      master
## 2              0     1            0        0      master
## 3              0     0            0        0      master
## 4              0     0            0        2      master
## 5              1     1            1        6      master
## 6              0     0            0        0      master

```

Weiteres Beispiel für JSON Daten

- Die Ergast Developer API ist ein experimenteller Web Service, der eine historische Aufzeichnung von Motorsportdaten liefert.



Figure 11:

Ergast Daten lesen

```

library(knitr)
library(jsonlite)
res <- fromJSON('http://ergast.com/api/f1/2004/1/results.json')
drivers <- res$MRData$RaceTable$Races$Results[[1]]$Driver
kable(head(drivers))

```

driverId	code	url	givenName	familyName	dateOfB
michael_schumacher	MSC	http://en.wikipedia.org/wiki/Michael_Schumacher	Michael	Schumacher	1969-
barrichello	BAR	http://en.wikipedia.org/wiki/Rubens_Barrichello	Rubens	Barrichello	1972-
alonso	ALO	http://en.wikipedia.org/wiki/Fernando_Alonso	Fernando	Alonso	1981-
ralf_schumacher	SCH	http://en.wikipedia.org/wiki/Ralf_Schumacher	Ralf	Schumacher	1975-
montoya	MON	http://en.wikipedia.org/wiki/Juan_Pablo_Montoya	Juan	Pablo Montoya	1975-
button	BUT	http://en.wikipedia.org/wiki/Jenson_Button	Jenson	Button	1980-

Daten der New York Times

- Die New York Times hat mehrere APIs als Teil des NYT-Entwickler-Netzwerks.
- Es ist eine Schnittstelle zu Daten aus verschiedenen Abteilungen, wie Nachrichtenartikel, Buchbesprechungen, Immobilien, etc.
- Registrierung ist erforderlich (aber kostenlos) und ein Schlüssel kann hier erhalten werden.

New York Times Beispiel

```
article_key <- "&api-key=c2fede7bd9aea57c898f538e5ec0a1ee:6:68700045"
url <- "http://api.nytimes.com/svc/search/v2/articlesearch.json?q=obamacare+socialism"
req <- fromJSON(paste0(url, article_key))
articles <- req$response$docs
# kable(head(articles))
```

XML Dateien einlesen

```
<?xml version="1.0" encoding="UTF-8"?>
- <osm generator="Overpass API" version="0.6">
  <note>The data included in this document is from www.openstreetmap.org. The data is made available under ODbL.</note>
  <meta areas="2017-05-07T07:22:03Z" osm_base="2017-05-07T08:19:03Z"/>
  - <node lon="8.5028074" lat="49.5190994" id="30560755">
    <tag v="Potlatch 0.5d" k="created_by"/>
    <tag v="playground" k="leisure"/>
  </node>
  - <node lon="8.5393963" lat="49.4963345" id="76468450">
    <tag v="Potlatch 0.4a" k="created_by"/>
    <tag v="playground" k="leisure"/>
    <tag v="Rutsche, Schaukel, großer Sandkasten, Tischtennis" k="note"/>
  </node>
  - <node lon="8.5529589" lat="49.4967807" id="76468534">
    <tag v="playground" k="leisure"/>
  </node>
  - <node lon="8.5487501" lat="49.4923030" id="76468535">
    <tag v="playground" k="leisure"/>
  </node>
  - <node lon="8.5481399" lat="49.5024260" id="76468536">
    <tag v="playground" k="leisure"/>
    <tag v="Schaukel, Rutsche, Sandkasten, Spielhäuser, Tischtennis" k="note"/>
  </node>
  - <node lon="8.5420356" lat="49.4975937" id="76468558">
    <tag v="playground" k="leisure"/>
  </node>
```

Figure 12:

Import von XML Dateien

```
install.packages("XML")
library(XML)
citation("XML")

##
## To cite package 'XML' in publications use:
##
## Duncan Temple Lang and the CRAN Team (2016). XML: Tools for
## Parsing and Generating XML Within R and S-Plus. R package
## version 3.98-1.5. https://CRAN.R-project.org/package=XML
```

```

##' A BibTeX entry for LaTeX users is
##'
##' @Manual{,
##'   title = {XML: Tools for Parsing and Generating XML Within R and S-Plus},
##'   author = {Duncan Temple Lang and the CRAN Team},
##'   year = {2016},
##'   note = {R package version 3.98-1.5},
##'   url = {https://CRAN.R-project.org/package=XML},
##' }
##'
##' ATTENTION: This citation information has been auto-generated from
##' the package DESCRIPTION file and may need manual editing, see
##' 'help("citation")'.

```

Das R-Paket XML - Gaston Sanchez

```
library("XML")
```



Figure 13: Gaston Sanchez - Dataflow

Seine Arbeit sieht man hier.

Das Arbeiten mit XML Daten

Funktionen im XML Paket

Function	Description
xmlName()	name of the node

Function	Description
xmlSize()	number of subnodes
xmlAttrs()	named character vector of all attributes
xmlGetAttr()	value of a single attribute
xmlValue()	contents of a leaf node
xmlParent()	name of parent node
xmlAncestors()	name of ancestor nodes
getSibling()	siblings to the right or to the left
xmlNamespace()	the namespace (if there's one)

Das neuere xml2 Paket

```
install.packages("xml2")
library(xml2)
citation("xml2")

##
## To cite package 'xml2' in publications use:
##
##   Hadley Wickham, James Hester and Jeroen Ooms (2017). xml2: Parse
##   XML. R package version 1.1.0.
##   https://CRAN.R-project.org/package=xml2
##
## A BibTeX entry for LaTeX users is
##
## @Manual{,
##   title = {xml2: Parse XML},
##   author = {Hadley Wickham and James Hester and Jeroen Ooms},
##   year = {2017},
##   note = {R package version 1.1.0},
##   url = {https://CRAN.R-project.org/package=xml2},
## }
```

Beispiel Daten - die OpenStreetMap API

- OpenStreetMap hat eine Editier-API welche zum Lesen und Speichern von Rohen Geodaten von bzw. auf die OpenStreetMap Datenbank benutzt werden kann.

Die OpenStreetMap ID herausfinden

Einzelne Objekte finden

<www.openstreetmap.org/export>

OSM Ausschnitte herunterladen

<www.openstreetmap.org/export>

Getting Data from the Web with R

Part 4: Parsing XML/HTML Content

Gaston Sanchez

April-May 2014

Content licensed under [CC BY-NC-SA 4.0](#)

Figure 14: Gaston Sanchez - Webdaten bekommen

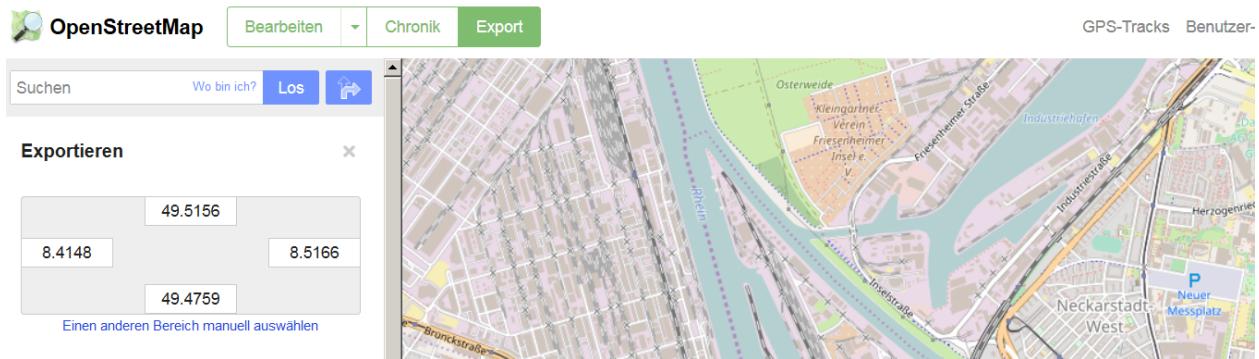


Figure 15:

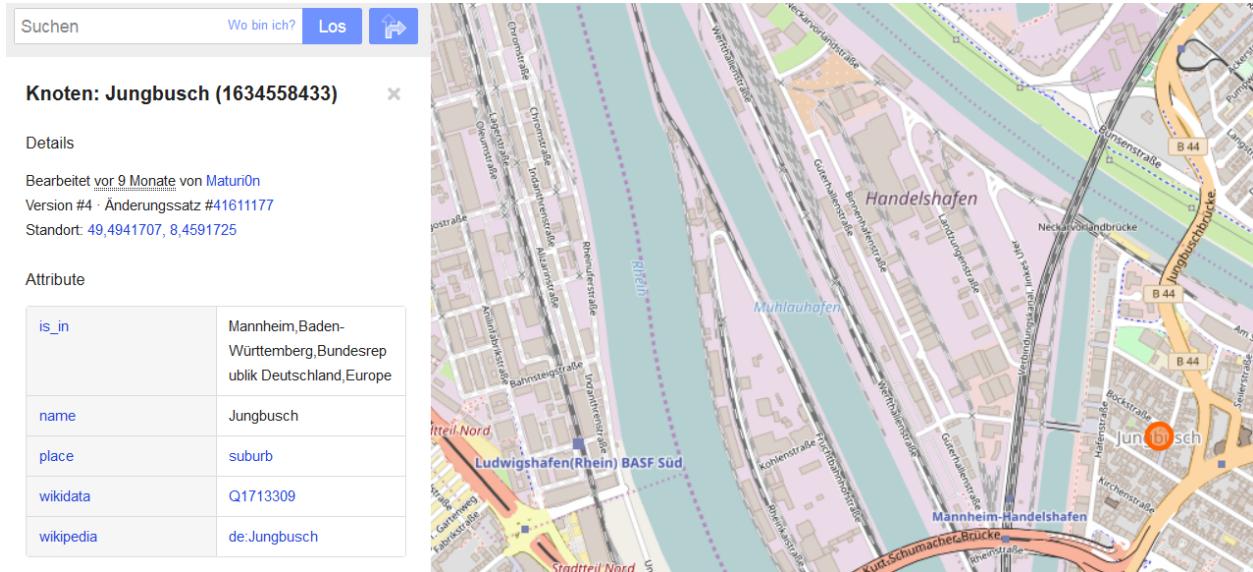


Figure 16:

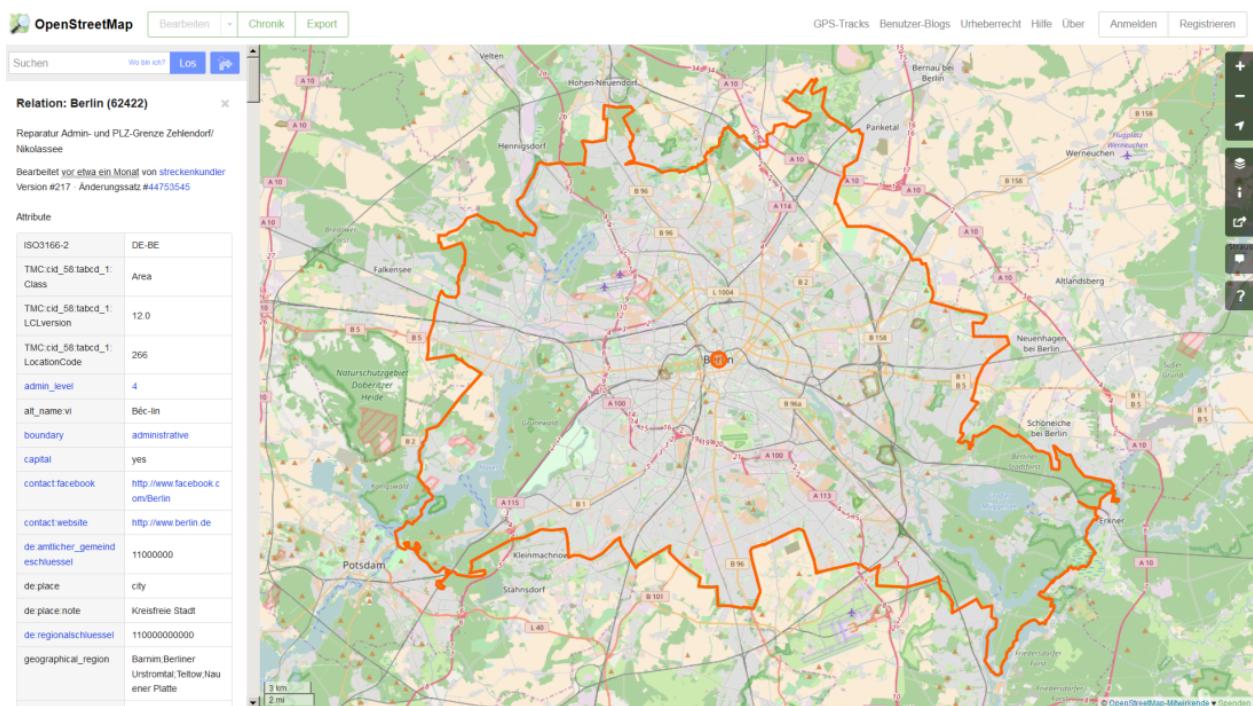


Figure 17: osm export

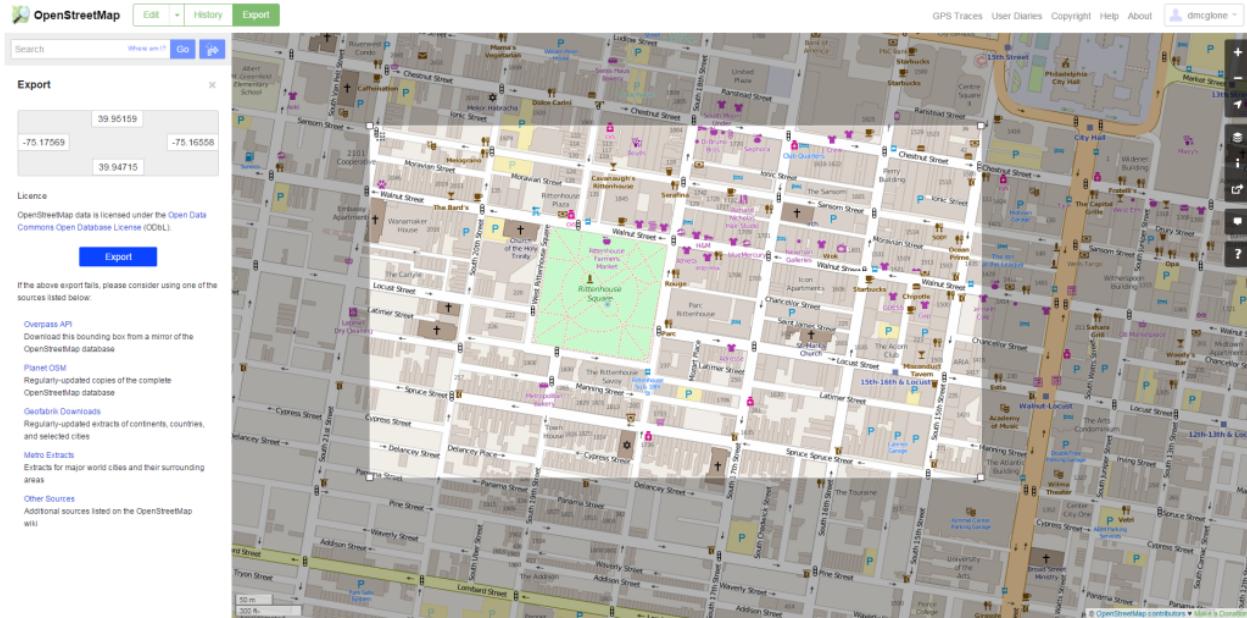


Figure 18: osm export

Erstes Beispiel

```
url <- "http://api.openstreetmap.org/api/0.6/
relation/62422"

library(xml2)
BE <- xmlParse(url)
```

```
<osm version="0.6" generator="CGImap 0.4.0 (19884 thorn-03.openstreetmap.org)" copyright="OpenStreetMap and contributors" attribution="http://www.openstreetmap.org/copyright"
license="http://opendatacommons.org/licenses/odbl/1-0/">
<relation id="62422" visible="true" version="209" changeset="36072269" timestamp="2015-12-20T19:49:52Z" user="tbicr" uid="278800">
<member type="node" ref="240109189" role="admin_centre"/>
<member type="way" ref="50291800" role="outer"/>
<member type="way" ref="7791336" role="outer"/>
<member type="way" ref="315222039" role="outer"/>
<member type="way" ref="77487568" role="outer"/>
<member type="way" ref="315222038" role="outer"/>
<member type="way" ref="98035898" role="outer"/>
<member type="way" ref="77501737" role="outer"/>
```

Figure 19: Administrative Grenzen Berlin

Das XML analysieren

- Tobi Bosede - Working with XML Data in R

```
xmltop = xmlRoot(BE)
class(xmltop)

## [1] "XMLInternalElementNode" "XMLInternalNode"
## [3] "XMLAbstractNode"

xmlSize(xmltop)

## [1] 1
```

```
xmlSize(xmltop[[1]])
```

```
## [1] 328
```

Nutzung von Xpath

Xpath, the XML Path Language, is a query language for selecting nodes from an XML document.

```
xpathApply(BE, "//tag[@k = 'source:population'])")
```

```
## [[1]]
```

```
## <tag k="source:population" v="http://www.statistik-berlin-brandenburg.de/Publikationen/Stat_Berichte"##  
## attr("class")  
## [1] "XMLNodeSet"
```

Beispiel: administrative Grenzen Berlin

Administrative Grenzen für Deutschland

```
url <- "http://api.openstreetmap.org/api/0.6/relation/62422"
```

```
BE <- xmlParse(url)
```

```
<osm version="0.6" generator="CGImap 0.4.0 (19884 thorn-03.openstreetmap.org)" copyright="OpenStreetMap and contributors" attribution="http://www.openstreetmap.org/copyright"  
license="http://opendatacommons.org/licenses/odbl/1-0/">  
-<relation id="62422" visible="true" version="209" changeset="36072269" timestamp="2015-12-20T19:49:52Z" user="tbicr" uid="278800">  
-<member type="node" ref="240109189" role="admin_centre"/>  
<member type="way" ref="50291800" role="outer"/>  
<member type="way" ref="77913336" role="outer"/>  
<member type="way" ref="315222039" role="outer"/>  
<member type="way" ref="77487568" role="outer"/>  
<member type="way" ref="315222038" role="outer"/>  
<member type="way" ref="98035898" role="outer"/>  
<member type="way" ref="77501737" role="outer"/>
```

Figure 20: Administrative Grenzen Berlin

Quelle für die Bevölkerungsgröße

```
xpathApply(BE, "//tag[@k = 'source:population'])")
```

```
## [[1]]
```

```
## <tag k="source:population" v="http://www.statistik-berlin-brandenburg.de/Publikationen/Stat_Berichte"##  
## attr("class")  
## [1] "XMLNodeSet"
```

-Statistik Berlin Brandenburg

Etwas überraschend:

```
xpathApply(BE, "//tag[@k = 'name:ta'])")
```

```
## [[1]]
## <tag k="name:ta" v="<U+OBAA><U+OBC6><U+OBBO><U+OBCT><U+OB2><U+OBFF><U+OBA9><U+OBCT>"/>
##
## attr(,"class")
## [1] "XMLNodeSet"
```

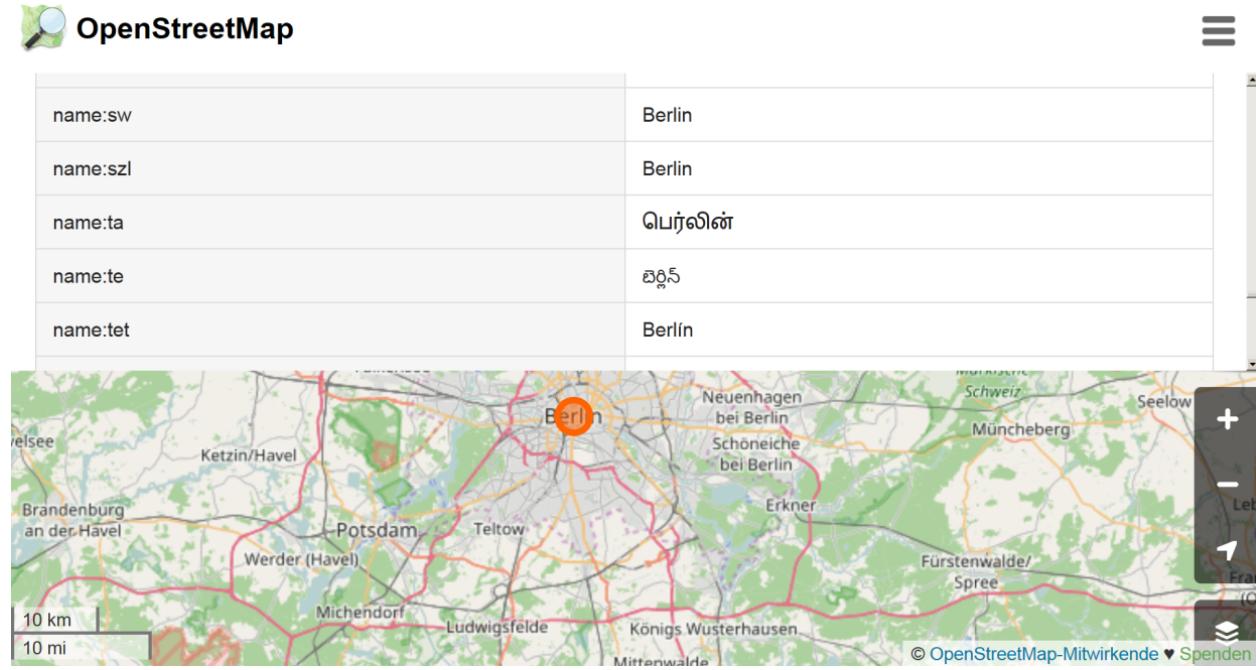


Figure 21:

Geographische Region

```
region <- xpathApply(BE,
  "//tag[@k = 'geographical_region']")
# regular expressions
region[[1]]
```



```
## <tag k="geographical_region" v="Barnim;Berliner Urstromtal;Teltow;Nauener Platte"/>
<tag k="geographical_region"
  v="Barnim;Berliner Urstromtal;
  Teltow;Nauener Platte"/>
```

Landkreis

Weiteres Beispiel

```
url2<-http://api.openstreetmap.org/api/0.6/node/25113879
obj2<-xmlParse(url2)
obj_amenity<-xpathApply(obj2, "//tag[@k = 'amenity']")[[1]]
obj_amenity
```



Figure 22: Barnim

```
## <tag k="amenity" v="university"/>
```

Wikipedia Artikel

```
xpathApply(obj2,"//tag[@k = 'wikipedia']")[[1]]  
  
## <tag k="wikipedia" v="de:Universität Mannheim"/>  
xpathApply(obj2,"//tag[@k = 'wheelchair']")[[1]]  
  
## <tag k="wheelchair" v="limited"/>  
xpathApply(obj2,"//tag[@k = 'name']")[[1]]  
  
## <tag k="name" v="Universität Mannheim"/>
```

Das C und das A

```
url3<-"http://api.openstreetmap.org/api/0.6/node/303550876"  
obj3 <- xmlParse(url3)  
xpathApply(obj3,"//tag[@k = 'opening_hours']")[[1]]  
  
## <tag k="opening_hours" v="Mo-Sa 09:00-20:00; Su,PH off"/>
```

Nur Fliegen ist schöner

```
url5<-"http://api.openstreetmap.org/api/0.6/way/162149882"  
obj5<-xmlParse(url5)  
xpathApply(obj5,"//tag[@k = 'name']")[[1]]  
  
## <tag k="name" v="City-Airport Mannheim"/>  
xpathApply(obj5,"//tag[@k = 'website']")[[1]]  
  
## <tag k="website" v="http://www.flugplatz-mannheim.de"/>  
xpathApply(obj5,"//tag[@k = 'iata']")[[1]]  
  
## <tag k="iata" v="MHG"/>
```

Einen Punkt parsen

```
url2 <- "http://api.openstreetmap.org/api/0.6/node/2923760808"  
RennesBa <- xmlParse(url2)  
RennesBa  
  
## <?xml version="1.0" encoding="UTF-8"?>  
## <osm version="0.6" generator="CGImap 0.6.0 (14801 thorn-01.openstreetmap.org)" copyright="OpenStreet  
##   <node id="2923760808" visible="true" version="7" changeset="47392918" timestamp="2017-04-02T20:42:  
##     <tag k="addr:city" v="Rennes"/>  
##     <tag k="addr:country" v="FR"/>  
##     <tag k="addr:housenumber" v="25"/>  
##     <tag k="addr:postcode" v="35000"/>
```

```

##      <tag k="addr:street" v="Avenue Jean Janvier"/>
##      <tag k="amenity" v="restaurant"/>
##      <tag k="capacity" v="90"/>
##      <tag k="name" v="Il Basilico"/>
##      <tag k="source:addr:housenumber" v="Rennes MÃ©tropole"/>
##      <tag k="source:addr:housenumber:ref" v="66075"/>
##      <tag k="source:addr:housenumber:version" v="2013-04-02"/>
##      <tag k="website" v="http://ilbasilico.fr"/>
##      <tag k="wheelchair" v="limited"/>
##      <tag k="wheelchair:description" v="Aucune sonnette pour indiquer sa prÃ©sence mais une rampe d'accÃ©s est installÃ©e au niveau de l'Ã©tablissement."/>
##    </node>
## </osm>
##

```

Einen Weg parsen

```

url3 <- "http://api.openstreetmap.org/api/0.6/way/72799743"
MadCalle <- xmlParse(url3)
MadCalle

```

```

## <?xml version="1.0" encoding="UTF-8"?>
## <osm version="0.6" generator="CGImap 0.6.0 (20705 thorn-03.openstreetmap.org)" copyright="OpenStreetMap contributors">
##   <way id="72799743" visible="true" version="5" changeset="11915713" timestamp="2012-06-16T14:49:40Z">
##     <nd ref="869268876"/>
##     <nd ref="1790008568"/>
##     <nd ref="864117544"/>
##     <nd ref="1790008571"/>
##     <nd ref="1790008601"/>
##     <nd ref="864117511"/>
##     <nd ref="1790008612"/>
##     <nd ref="1790008618"/>
##     <nd ref="864117819"/>
##     <tag k="highway" v="residential"/>
##     <tag k="name" v="Calle Alfonso Ercilla"/>
##     <tag k="oneway" v="yes"/>
##     <tag k="surface" v="asphalt"/>
##   </way>
## </osm>
##

```

The Overpass API

The Overpass API is a read-only API that serves up custom selected parts of the OSM map data.
http://wiki.openstreetmap.org/wiki/Overpass_API

Wichtige Information

http://wiki.openstreetmap.org/wiki/Map_Features



Figure 23: Logo Overpass API

Key	Value	Element	Comment	Rendering	Photo
leisure	beach_resort	<input type="checkbox"/> <input checked="" type="checkbox"/>	To mark the boundary of a managed beach.		
leisure	bird_hide	<input type="checkbox"/> <input checked="" type="checkbox"/>	A place to observe birds.		
leisure	common	<input type="checkbox"/> <input checked="" type="checkbox"/>	An area where the public can walk anywhere (UK)		
leisure	dance	<input type="checkbox"/> <input checked="" type="checkbox"/>	A place to go dancing.		
leisure	dog_park	<input type="checkbox"/> <input checked="" type="checkbox"/>	Designated area, with or without a fenced boundary, where dog-owners are permitted to exercise their pets unrestrained		
leisure	firepit	<input type="checkbox"/> <input checked="" type="checkbox"/>	A permanent location or structure to host campfires, bonfires, etc		
leisure	fishing	<input type="checkbox"/> <input checked="" type="checkbox"/>			

Figure 24: osm map features

Beispiel: Nutzung der Overpass API

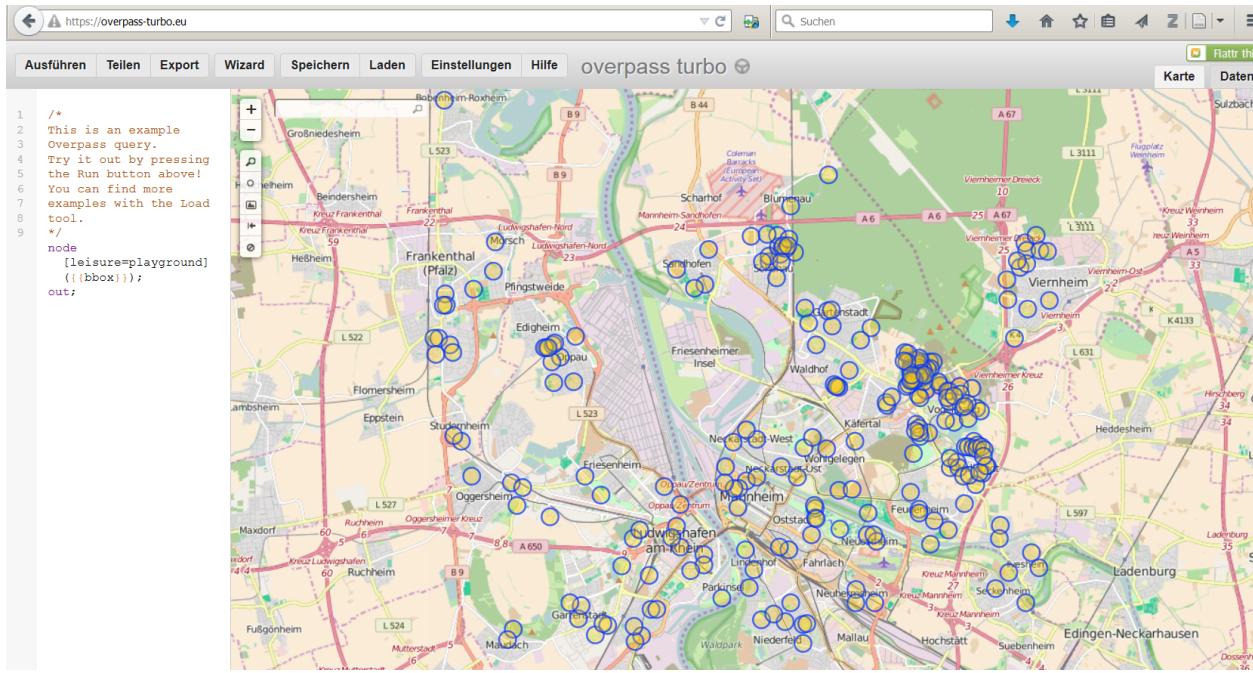


Figure 25: Spielplätze Mannheim

Export der Rohdaten

Import von der Overpass API zu R

```

library(XML)
place <- "Mannheim"
type_obj <- "node"
object <- "leisure=playground"

InfoList <- xmlParse(paste(Link1,place,"\"],",
type_obj,"(area)[",object,"];out;","sep=""))

```

XML Output

Das Arbeiten mit XML Daten (xpath)

Die Liste der ID's mit dem Wert *playground*:

```

node_id <- xpathApply(InfoList,
"/@v='playground']/parent::node/@id")
## node_id[[1]]

```

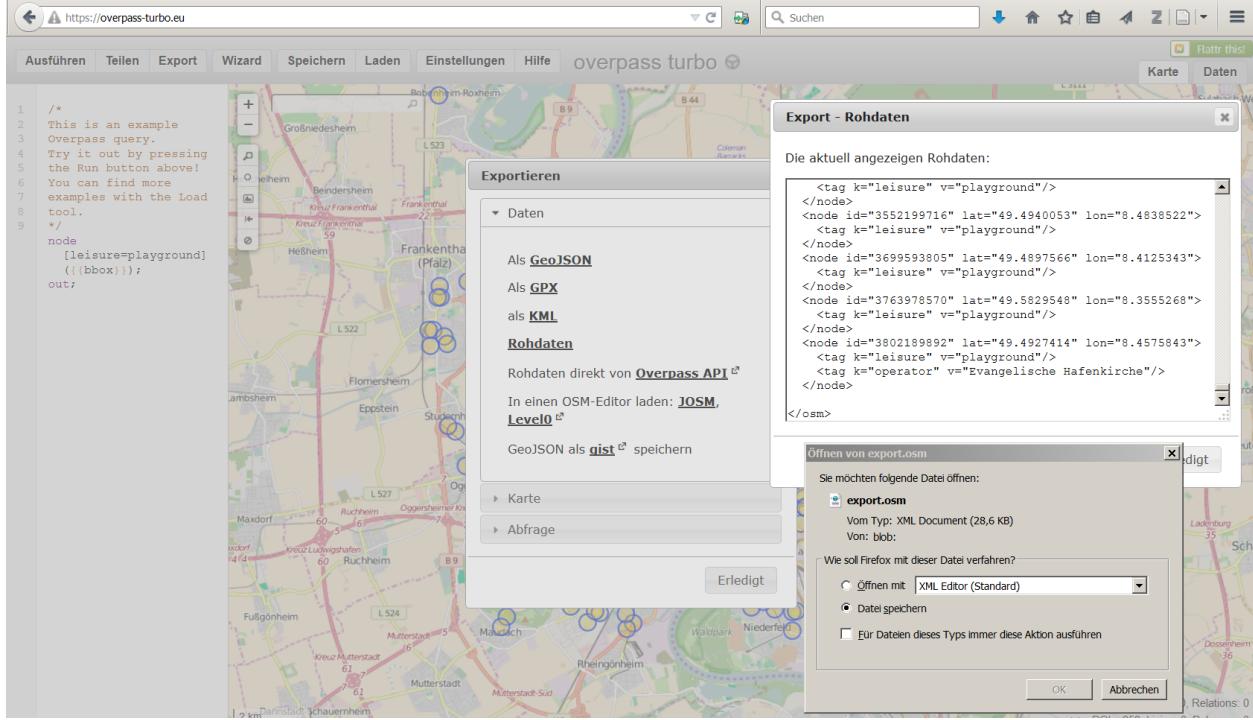


Figure 26: Export Rohdaten

```

<?xml version="1.0" encoding="UTF-8"?>
- <osm generator="Overpass API" version="0.6">
  <note>The data included in this document is from www.openstreetmap.org. The data is made available under ODbL.</note>
  <meta areas="2017-02-06T06:35:03Z" osm_base="2017-02-06T06:48:02Z"/>
  - <node lon="8.5028074" lat="49.5190994" id="30560755">
    <tag v="Potlatch 0.5d" k="created_by"/>
    <tag v="playground" k="leisure"/>
  </node>
  - <node lon="8.5393963" lat="49.4963345" id="76468450">
    <tag v="Potlatch 0.4a" k="created_by"/>
    <tag v="playground" k="leisure"/>
    <tag v="Rutsche, Schaukel, großer Sandkasten, Tischtennis" k="note"/>
  </node>
  - <node lon="8.5529589" lat="49.4967807" id="76468534">
    <tag v="playground" k="leisure"/>
  </node>
  - <node lon="8.5487501" lat="49.4923030" id="76468535">
    <tag v="playground" k="leisure"/>
  </node>

```

Figure 27: Spielplätze in Mannheim

```

> node_id[[1]]
      id
"30560755"
attr(,"class")
[1] "XMLAttributeValue"
> |

```

Figure 28: Erste node id

latitude und longitude bekommen

```
lat_x <- xpathApply(InfoList,
  "//tag[@v= 'playground']/parent::node/@ lat")
# lat_x[[1]];lat_x[[2]]

lat_x <- xpathApply(InfoList,
  "//tag[@v= 'playground']/parent::node/@ lon")
```

```
          lat
"49.5190994"
attr(,"class")
[1] "XMLAttributeValue"
          lat
"49.4963345"
attr(,"class")
[1] "XMLAttributeValue"
```

Figure 29: Latitude Koordinate

Paket auf Github

```
library(devtools)
install_github("Japhilko/gosmd")

library(gosmd)
pg_MA <- get_osm_nodes(object="leisure=playground",
                        "Mannheim")
info <- extract_osm_nodes(OSM.Data=pg_MA,
                           value="playground")
```

Ausschnitt der Ergebnisse

	leisure	lat	lon	note
30560755	playground	49.51910	8.502807	NA
76468450	playground	49.49633	8.539396	Rutsche, Schaukel, großer Sandkasten, Tischtennis
76468534	playground	49.49678	8.552959	NA
76468535	playground	49.49230	8.548750	NA
76468536	playground	49.50243	8.548140	Schaukel, Rutsche, Sandkasten, Spielhaus, Tischtennis
76468558	playground	49.49759	8.542036	NA

Link

- Tutorial zur Nutzung der Overpass API
- Vignette xml2

Mehr Beispiele, wie man mit XML Daten umgeht:

- Deborah Nolan - Extracting data from XML
- Duncan Temple Lang - A Short Introduction to the XML package for R

Noch mehr Informationen

- Web Daten manipulieren
- Tutorial zu xquery
- R und das Web (für Anfänger), Teil II: XML und R
- String Manipulation
- Nutzung, Vor- und Nachteile OSM
- Forschungsprojekte im Zusammenhang mit OpenStreetMap
- XML parsen - Stackoverflow
- Processing of GeoJson data in R

Die Pakete rvest und RCurl

Das Paket rvest

```
install.packages("rvest")
```

- rvest - Easily Harvest (Scrape) Web Pages

```
library(rvest)
```

```
ht <- read_html('https://www.google.co.in/search?q=guitar+repair+workshop')
links <- ht %>% html_nodes(xpath='//h3/a') %>% html_attr('href')
gsub('/url\\?q=', '', sapply(strsplit(links[as.vector(grep('url', links))]), split='&'), '[', 1))

## [1] "http://theguitarrepairworkshop.com/"
## [2] "http://www.guitarservices.com/"
## [3] "http://www.guitarrepairbench.com/guitar-building-projects/guitar-workshop/guitar-workshop-project"
## [4] "https://www.taylorguitars.com/dealer/guitar-repair-workshop-ltd"
## [5] "https://www.facebook.com/The-Guitar-Repair-Workshop-847517635259712/"
## [6] "http://www.laweekly.com/music/10-best-guitar-repair-shops-in-los-angeles-4647166"
## [7] "http://guitarworkshopglasgow.com/pages/repairs-1"
## [8] "https://www.justdial.com/Mumbai/Guitar-Repair-Services/nct-10988623"
## [9] "https://www.justdial.com/Delhi-NCR/Guitar-Repair-Services/nct-10988623"
```

Hin und weg

```
library(XML)
url4<-"http://api.openstreetmap.org/api/0.6/node/25439439"
obj4 <- xmlParse(url4)
xpathApply(obj4,"//tag[@k = 'railway:station_category']")[[1]]
```

<tag k="railway:station_category" v="2"/>

- Wikipedia Artikel Bahnhofskategorien

Stufe	Bahnsteigkanten	Bahnsteilänge	Reisende/Tag	Zughalte/Tag
6	1	bis 90 m	bis 49	bis 10
5	2	> 90 bis 140 m	50 bis 299	11 bis 50
4	3 bis 4	> 140 bis 170 m	300 bis 999	51 bis 100
3	5 bis 9	> 170 bis 210 m	1000 bis 9999	101 bis 500
2	10 bis 14	> 210 bis 280 m	10.000 bis 49.999	501 bis 1000
1	ab 15	> 280 m	ab 50.000	ab 1001

Prozent	Kategorie
> 90 %	1
> 80 bis 90 %	2
> 60 bis 80 %	3
> 50 bis 60 %	4
> 40 bis 50 %	5
> 25 bis 40 %	6
bis 25 %	7

Figure 30:

Exkurs: Bahnhofskategorien

- rvest: Easily Harvest (Scrape) Web Pages

```
library(rvest)
bhfkat<-read_html(
  "https://de.wikipedia.org/wiki/Bahnhofskategorie")
df_html_bhfkat<-html_table(
  html_nodes(bhfkat, "table")[[1]], fill = TRUE)
```

Bahnhofskategorien Übersicht

Stufe	Bahnsteigkanten	Bahnsteilänge	Reisende/Tag	Zughalte/Tag
6	01	> 000 bis 090 m	00000 bis 00049	000 bis 0010
5	02	> 090 bis 140 m	00050 bis 00299	011 bis 0050
4	03 bis 04	> 140 bis 170 m	00300 bis 00999	051 bis 0100
3	05 bis 09	> 170 bis 210 m	01000 bis 09999	101 bis 0500
2	10 bis 14	> 210 bis 280 m	10.000 bis 49.999	501 bis 1000
1	00i ab 15	> 280 m	00000i ab 50.000	000i ab 1001

Links

- Scraping CRAN with rvest

Webscraping

Notwendige Pakete

```
install.packages("tidyverse")
library(tidyverse)
```

The screenshot shows a blog post on the RStudio Blog. The title is "tidyverse 1.0.0" and the date is September 15, 2016. The content discusses the tidyverse as a set of packages that work in harmony because they share common data representations and API design. It mentions the tidyverse package and its purpose of making it easy to install and load core packages from the tidyverse in a single command. The text also notes that the best place to learn about all the packages in the tidyverse is R for Data Science, and it anticipates improvements to package websites, citation easier, and providing a common home for discussions about data analysis with the tidyverse.

Figure 31:

- R für DataScience

Weitere benötigte Pakete

- Das Paket `stringr`

```
library(stringr)
library(forcats)
library(ggmap)
library(rvest)
```

Daten von Wikipedia einsammeln

```
html.world_ports <- read_html("https://en.wikipedia.org/wiki/List_of_busiest_container_ports")
df.world_ports <- html_table(html_nodes(html.world_ports, "table")[[2]], fill = TRUE)

library(DT)
datatable(df.world_ports)
```

Die Daten anschauen

```
glimpse(df.world_ports)

## # Observations: 50
## # Variables: 15
## # $ Rank      <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16...
## # $ Port      <chr> "Shanghai", "Singapore", "Shenzhen", "Ningbo-Zhoushan...
## # $ Economy   <chr> "China", "Singapore", "China", "China", "Hong Kong", ...
## # $ 2015[1]   <chr> "36,516", "30,922", "24,142", "20,636", "20,073", "19...
## # $ 2014[2]   <chr> "35,268", "33,869", "23,798", "19,450", "22,374", "18...
## # $ 2013[3]   <chr> "33,617", "32,240", "23,280", "17,351", "22,352", "17...
## # $ 2012[4]   <chr> "32,529", "31,649", "22,940", "16,670", "23,117", "17...
## # $ 2011[5]   <chr> "31,700", "29,937", "22,570", "14,686", "24,384", "16...
## # $ 2010[6]   <chr> "29,069", "28,431", "22,510", "13,144", "23,532", "14...
## # $ 2009[7]   <chr> "25,002", "25,866", "18,250", "10,502", "20,983", "11...
## # $ 2008[8]   <chr> "27,980", "29,918", "21,414", "11,226", "24,248", "13...
## # $ 2007[9]   <chr> "26,150", "27,932", "21,099", "9,349", "23,881", "13,...
## # $ 2006[10]  <chr> "21,710", "24,792", "18,469", "7,068", "23,539", "12,...
## # $ 2005[11]  <chr> "18,084", "23,192", "16,197", "5,208", "22,427", "11,...
## # $ 2004[12]  <chr> "14,557", "21,329", "13,615", "4,006", "21,984", "11,...
```

Das Paket rvest

```
library(rvest)
ht <- read_html('https://www.google.co.in/search?q=guitar+repair+workshop')
links <- ht %>% html_nodes(xpath='//h3/a') %>% html_attr('href')
gsub('/url\?q=', '', sapply(strsplit(links[as.vector(grep('url', links))]), split='&'), '[', 1))

## [1] "http://theguitarrepairworkshop.com/"
## [2] "http://www.guitarservices.com/"
## [3] "http://www.guitarrepairbench.com/guitar-building-projects/guitar-workshop/guitar-workshop-project"
## [4] "https://www.taylorguitars.com/dealer/guitar-repair-workshop-ltd"
## [5] "https://www.facebook.com/The-Guitar-Repair-Workshop-847517635259712/"
## [6] "http://www.laweekly.com/music/10-best-guitar-repair-shops-in-los-angeles-4647166"
## [7] "http://guitarworkshopglasgow.com/pages/repairs-1"
## [8] "https://www.justdial.com/Mumbai/Guitar-Repair-Services/nct-10988623"
## [9] "https://www.justdial.com/Delhi-NCR/Guitar-Repair-Services/nct-10988623"
```

Links

- How to really do an analysis in R (part 1, data manipulation)
- Read CSV From The Web
- Scraping CRAN with rvest

Beispiel Scraping Wikipedia

Einleitung

Im Folgenden werde ich zeigen, wie man Textinformationen aus Wikipedia herunterladen, verarbeiten und analysieren kann.

```
install.packages("NLP")
install.packages("tm")
install.packages("FactoMineR")
```

Die verwendeten Pakete

- Das R-Paket `stringi` von Marek Gagolewski and Bartek Tartanus bietet Möglichkeiten zur String Verarbeitung.

```
library("stringi")
```

- `tm` ist ein R-Paket um Text Mining zu realisieren. Es wurde von Ingo Feinerer, Kurt Hornik, und David Meyer geschrieben.

```
library("tm")
```

- Und schließlich brauchen wir das `FactoMineR`-Paket, das von Sébastien Le, Julie Josse und François Husson zur Durchführung der Hauptkomponentenanalyse erstellt wurde.

```
library("FactoMineR")
```

Die Text Daten herunterladen

- Als Beispiel verwenden wir Daten zu verschiedenen Krankheiten.
- In diesem Fall habe ich 7 deutsche Webseiten für Infektionskrankheiten ausgewählt.

```
wiki <- "http://de.wikipedia.org/wiki/"

titles <- c("Zika-Virus", "Influenza-A-Virus_H1N1",
          "Spanische_Grippe", "Influenzavirus",
          "Vogelgrippe_H5N1",
          "Legionellose-Ausbruch_in_Warstein_2013",
          "Legionellose-Ausbruch_in_Jülich_2014")
```

Das Herunterladen der Seiten

- Zunächst wird ein Container erstellt um die Ergebnisse abzuspeichern
- Dann wird der Text für jeden Artikel heruntergeladen und in dem Container gespeichert.

```
articles <- character(length(titles))

for (i in 1:length(titles)){
  articles[i] <- stri_flatten(
    readLines(stri_paste(wiki, titles[i])), col = " ")
}
```

```
docs <- Corpus(VectorSource(articles))
```

Die Daten vorbereiten

Das Folgende basiert auf einem Blogpost von Norbert Ryciak über die automatische Kategorisierung von Wikipedia-Artikeln.

- Eine Fehlermeldung ist aufgetreten, als ich den Code ausgewertet habe.
- Es war möglich, dieses Problem mit Hinweisen aus einer Diskussion auf Stackoverflow zu lösen.

```
docs2 <- tm_map(docs, function(x) stri_replace_all_regex(  
  x, "<.+?>", " "))  
docs3 <- tm_map(docs2, function(x) stri_replace_all_fixed(  
  x, "\t", " "))
```

Den Text weiterverarbeiten

```
docs4 <- tm_map(docs3, PlainTextDocument)  
docs5 <- tm_map(docs4, stripWhitespace)  
docs6 <- tm_map(docs5, removeWords, stopwords("german"))  
docs7 <- tm_map(docs6, removePunctuation)  
docs8 <- tm_map(docs7, tolower)  
# docs8 <- tm_map(docs8, PlainTextDocument)  
  
dtm <- DocumentTermMatrix(docs8)
```

Principal Component Analysis

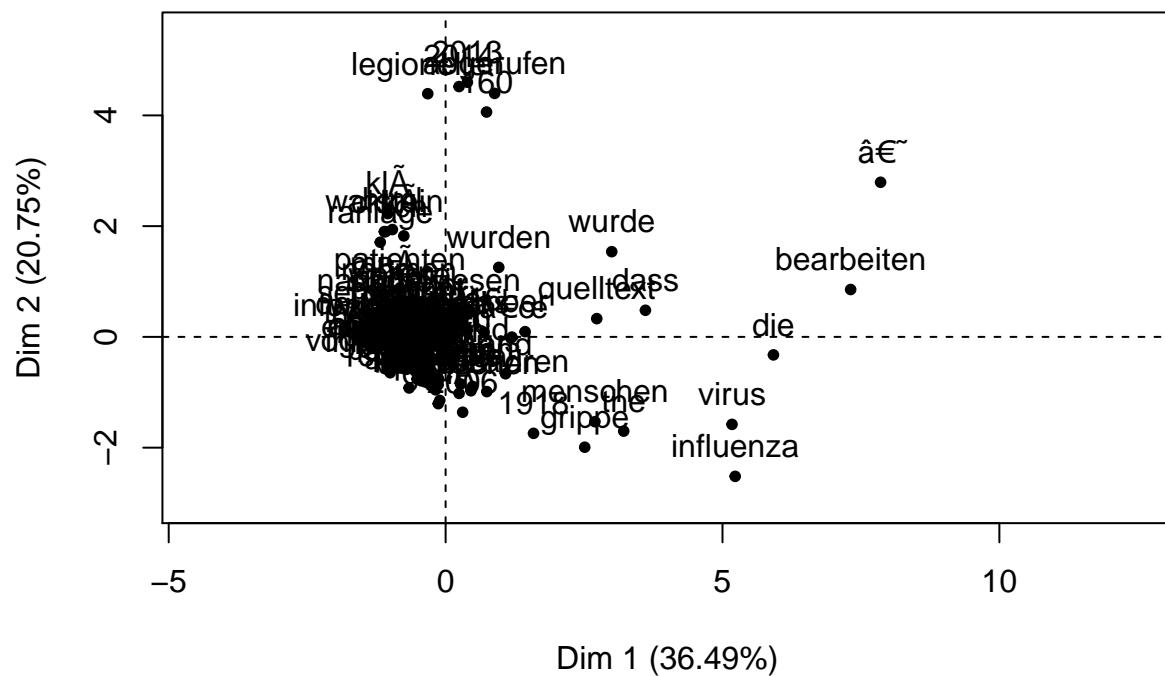
- Der folgende Code ist auf einem Blog post von Arthur Charpentier über das Mining von Wikipedia basiert.

```
dtm2 <- as.matrix(dtm)  
frequency <- colSums(dtm2)  
frequency <- sort(frequency, decreasing=TRUE)  
words <- frequency[frequency>20]  
s <- dtm2[,which(colnames(dtm2) %in% names(words))]  
  
for(i in 2:nrow(dtm2)){  
  s <- cbind(s, dtm2[i, which(colnames(dtm2) %in%  
    names(words))])  
}  
  
colnames(s) <- titles
```

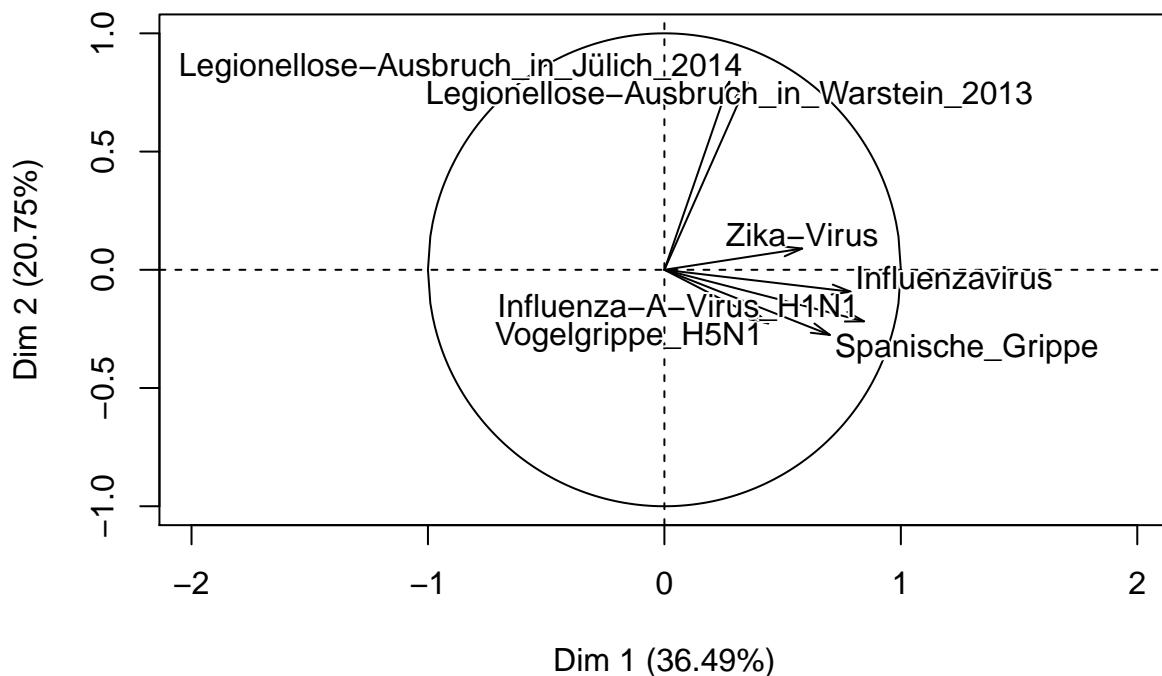
Ergebnis

```
PCA(s)
```

Individuals factor map (PCA)



Variables factor map (PCA)



```
## **Results for the Principal Component Analysis (PCA)**
## The analysis was performed on 125 individuals, described by 7 variables
## *The results are available in the following objects:
##
##      name           description
## 1  "$eig"          "eigenvalues"
## 2  "$var"          "results for the variables"
## 3  "$var$coord"    "coord. for the variables"
## 4  "$var$cor"       "correlations variables - dimensions"
## 5  "$var$cos2"     "cos2 for the variables"
## 6  "$var$contrib"   "contributions of the variables"
## 7  "$ind"          "results for the individuals"
## 8  "$ind$coord"    "coord. for the individuals"
## 9  "$ind$cos2"     "cos2 for the individuals"
## 10 "$ind$contrib"   "contributions of the individuals"
## 11 "$call"          "summary statistics"
## 12 "$call$centre"   "mean of the variables"
## 13 "$call$ecart.type" "standard error of the variables"
## 14 "$call$row.w"    "weights for the individuals"
## 15 "$call$col.w"    "weights for the variables"
```

Ergebnis

- In der Factor Map sehen wir das erwartete Ergebnis.
- Die Seiten zur Legionellen Krankheit sind sehr nah beianander, während die Seiten zur Influenza in

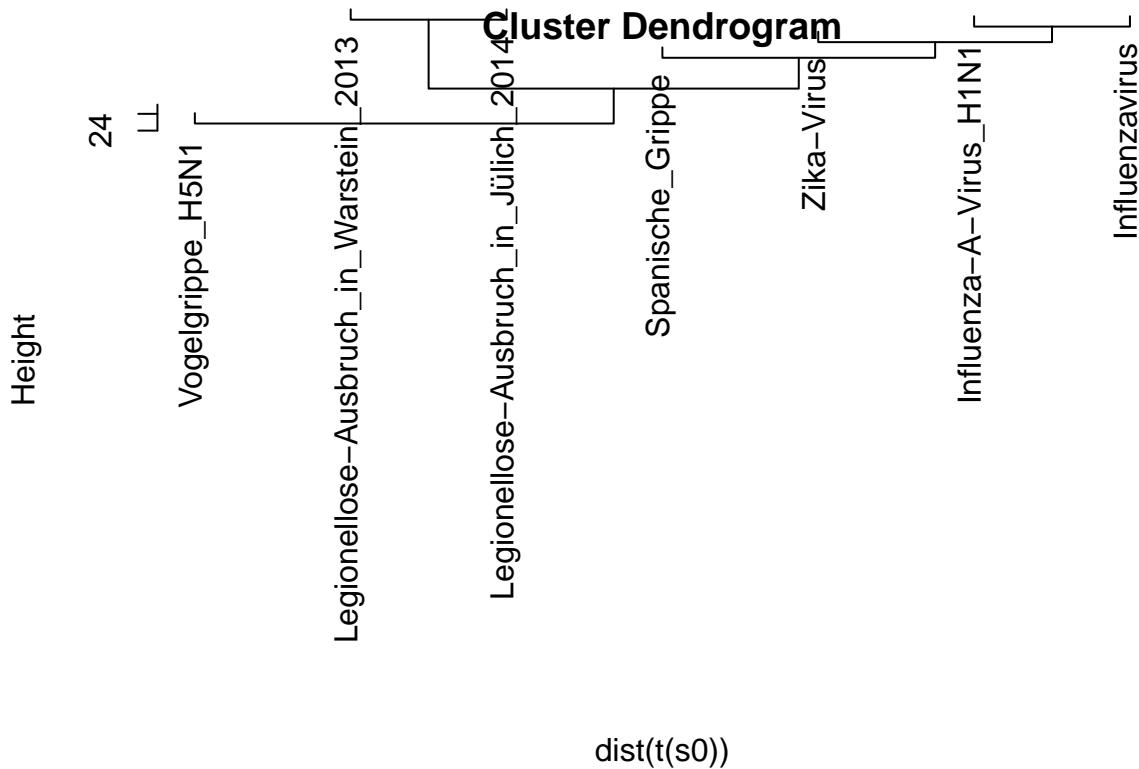
einem anderen Teil sind.

Das Dendrogramm

- Im Folgenden wird die Normalisierung durchgeführt und die Ergebnisse werden geplottet.

```
s0 <- s/apply(s,1,sd)
h <- hclust(dist(t(s0)), method = "ward")

plot(h, labels = titles, sub = "")
```



`dist(t(s0))`

- Youtube Video zu Text Mining