Meldeplattform

Dr. Till Straube
straube@geo.uni-frankfurt.de

Institut für Humangeographie
Goethe-Universität Frankfurt

2023-02-22

Welcome to Rmarkdown!

Rmarkdown documents are a nice way to combine text with code.

```
print("This is a code chunk.")
```

```
## [1] "This is a code chunk."
```

```
print("Inside this chunk, every character counts!")
```

```
## [1] "Inside this chunk, every character counts!"
```

Outside the chunk, you're safe! This text will not be executed as code. Feel free to edit and take your own notes here.

You can follow markdown conventions to produce beautiful documents: https://www.markdownguide.org/basic-syntax/

Load packages

Packages (or libraries, or modules) are pieces of code that can be used in your own code.

We will be relying on two packages today:

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(jsonlite)
```

```
##
## Attaching package: 'jsonlite'
##
## The following object is masked from 'package:purrr':
##
##     flatten
```

If you get an error while executing this, you will need to install the missing packages first.

Get data from API

```
url <- "https://www.meldeplattform-radverkehr.de/geoserver/geocms/ows?service=WFS&version=1.0&reques
response <- read_json(url)
```

Explore the response

The response is a „nested“ list: A list of lists (of lists)…

What entries are in the list?

```
names(response)
```

```
## [1] "type"          "totalFeatures" "features"      "crs"
```

Let’s explore the entries:

```
response$type
```

```
## [1] "FeatureCollection"
```

```
response$totalFeatures
```

```
## [1] 789
```

```
#response$features

#response$features[[112]]
```

What is some information that is visible here but not in the browser?

What is visible in the browser but obscured here?

Convert to tabular format

This next bit is some classic „data wrangling“.

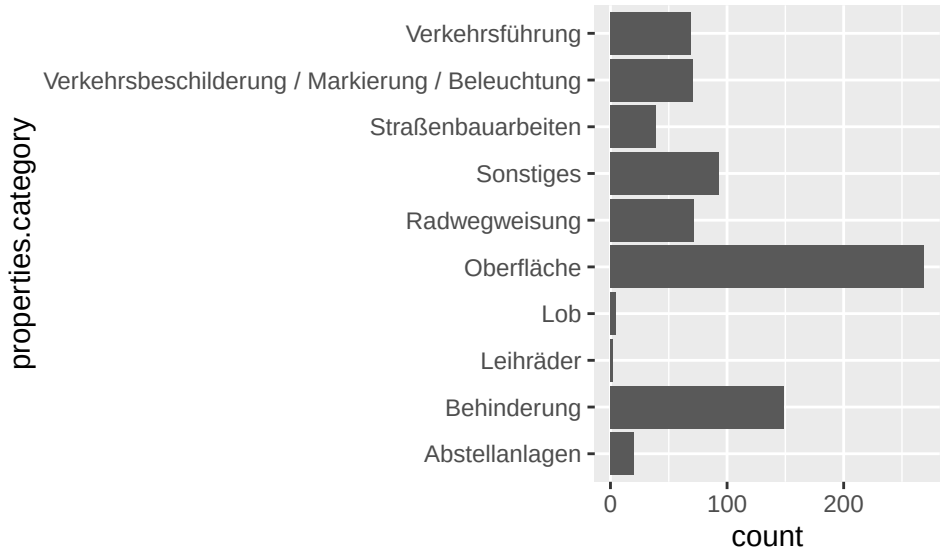
```
alerts <- response$features %>%
  map(\(f) { as.list(unlist(f)) }) %>%
  bind_rows()
```

Get a feeling for the dataset

```
View(alerts)
str(alerts)
summary(alerts)
names(alerts)
alerts$properties.comment[1]
```

How frequent are the categories?

```
ggplot(alerts, aes(x = properties.category)) +
  geom_bar() +
  coord_flip()
```



Who reports these things?

```
alerts %>%
  group_by(properties.reporter) %>%
  summarize(count = n()) %>%
  arrange(desc(count))
```

```
## # A tibble: 470 x 2
##    properties.reporter count
##    <chr>               <int>
##  1 <NA>                   32
##  2 Lemster, Ralf          13
##  3 Bandur, Maria          12
##  4 Schlösser, Ulrich      12
##  5 Abt, marcus             9
##  6 Kotek, Sebastian        9
##  7 Schmalhofer, Georg      9
##  8 Achilles, Sebastian     8
##  9 Häuser, Peter           7
## 10 Klein, Andreas          7
## # ... with 460 more rows
```

What about Kettenhofweg?

```
alerts %>%
  filter(str_detect(properties.addresssummary, "Kettenhofweg")) %>%
  .$properties.ticketdescription
```

```
## [1] "(Im Mängelmelder gibt es leider keine Kategorie Sonstiges)\n\nDer Glascontainer auf der Ecke
## [2] "Der Kettenhofweg ist jetzt teilweise Fahrradstraße.\n\nMeist auf der nördlichen Seite sind Pa
## [3] ""
```

Let's do some mapping!

We need more libraries for this.

```
library(sf)
```

```
## Linking to GEOS 3.11.0, GDAL 3.5.2, PROJ 9.0.1; sf_use_s2() is TRUE
```

```
library(leaflet)
```

Convert to Simple Features

```
response$crs
```

```
## $type
## [1] "name"
##
## $properties
## $properties$name
## [1] "urn:ogc:def:crs:EPSG::3857"
```

The coordinate reference system is an important piece of information to properly interpret geographical coordinates.

It can be identified by its EPSG code. Let's remember it for later:

```
crs <- 3857
```

```
alert_map <- alerts %>%
  st_as_sf(coords = c("geometry.coordinates1", "geometry.coordinates2")) %>%
  st_set_crs(crs) %>%
  st_transform(4326)
```

Map it

```
alert_map %>%
  filter(parse_number(properties.status) >= 12) %>%
  leaflet() %>%
  addTiles() %>%
  leaflet::addCircleMarkers()
```