



# Übung Open Data

## **BADEWETTER-INDEX, Gruppe 5**

**Abschlusspräsentation, 23. Mai 2019**

Benjamin Schüpbach und Christoph von Matt

Forschungsstelle Digitale Nachhaltigkeit  
Institut für Wirtschaftsinformatik  
Universität Bern

# Team

## > Benjamin Schüpbach

- Hauptfach: Geographie (Master)
- Semester: 4
- Bisherige Programmiererfahrung: Python
- Email: [benjamin.schuepbach@students.unibe.ch](mailto:benjamin.schuepbach@students.unibe.ch)

## > Christoph von Matt

- Hauptfach: Geographie (Master)
- Semester: 4
- Bisherige Programmiererfahrung: Python, R
- Email: [christoph.vonmatt@students.unibe.ch](mailto:christoph.vonmatt@students.unibe.ch)

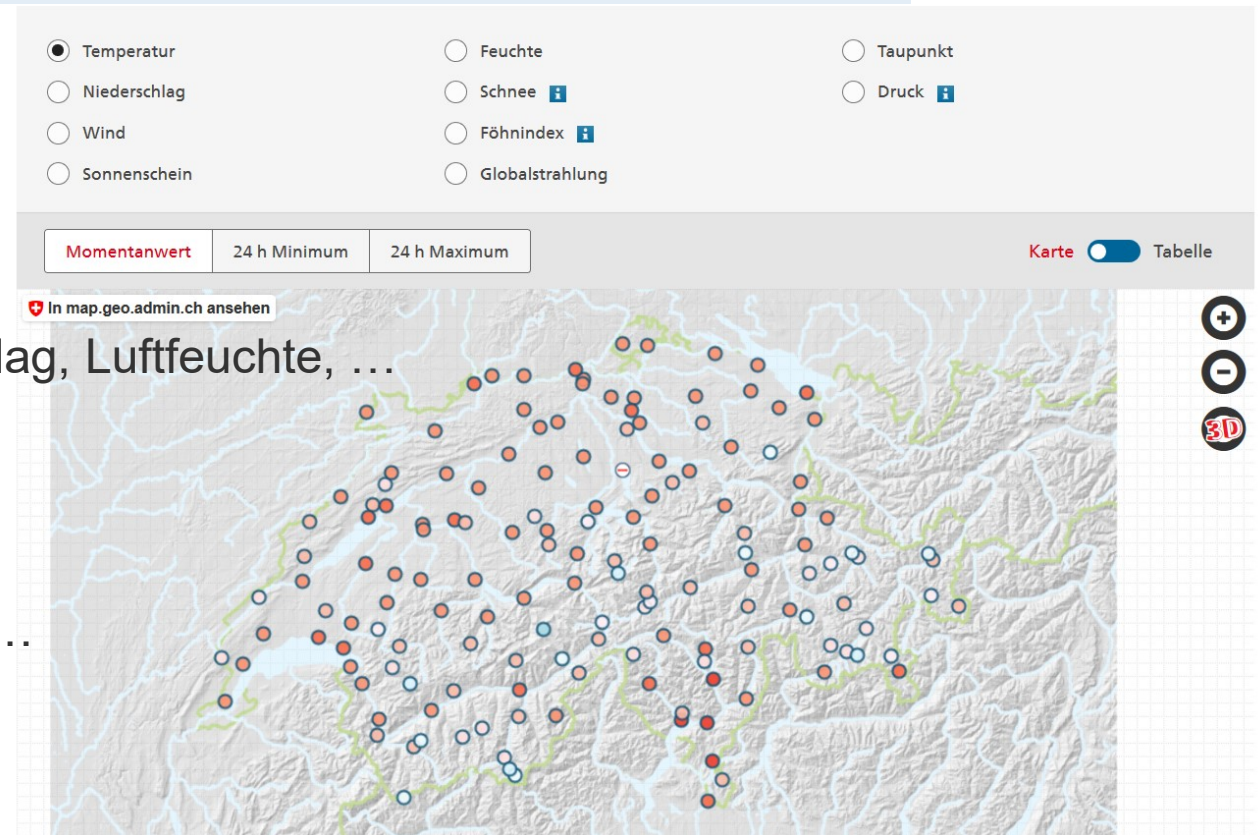
# Aufgabenstellung

- > Ausgangslage, Problemstellung:
  - Messwerte automatischer Wetterstationen (MeteoSchweiz)
  
- > Zielsetzung:
  - Aufzeigen des Potentials von OpenData anhand eines Badeindexes
  
- > Datenquelle, Data Coaches:
  - Daniel Wolf (MeteoSchweiz)
  - Joël Fisler (MeteoSchweiz)
  - Weitere Quellen: BAFU, BFS (komplementäre Daten)

# Daten

## > Datenformate:

- **CSV (Meteodaten, MCH)**  
→ StationsID, Temperatur, Niederschlag, Luftfeuchte, ...
- **.txt (Metadaten, MCH)**  
→ Stationshöhe, Stationsname, ...
- **JSON (Flusstemperaturen, BAFU)**  
→ Stationsname, Temperaturklasse, ...
- **Shapefile (Basisgeometrien, BFS)**  
→ Landes- und Kantonsgrenzen



Seite von MeteoSchweiz: Messdaten

# Vorgehen

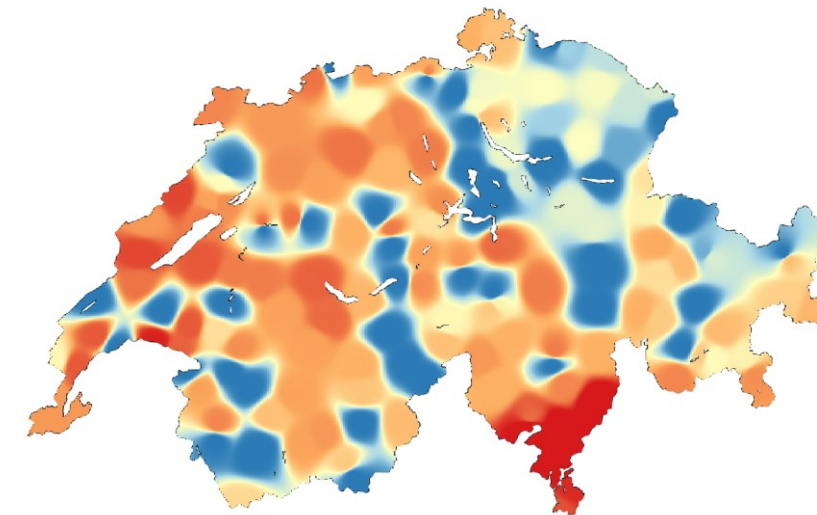
- > Datenbeschaffung:
  - OpenData Plattform (direkt per R-Skript)
  - Webseite Bundesamt für Statistik (BFS)
- > Datentransformation:
  - Konvertierung GeoJson / TopoJson
  - Indexberechnung (R-Skript),  
räumliche Interpolation (QGIS)
- > Technologien, Libraries etc.:
  - D3.js, Bootstrap, R, QGIS, jQuery,  
Google Fonts

```

181
182 #####
183 # Section 3: Joining Data-Tables together by "Station"
184 joined <- right_join(meta, badewetter_subset, by="Station")
185
186 joined <- as.data.frame(joined)
187
188 #####
189 # Section 4: INDEX CALCULATION
190 # get dimensions r-rows, c-cols
191 dims <- dim(joined)
192 r <- dims[1]
193 c <- dims[2]
194 # Badewetter-Index Calculation
195 index <- rep(NA, r)
196
197 cols <- colnames(joined)
198
199 for(i in 1:r){
200   temp <- as.numeric(joined[i,grep("Temp+", cols)])
201   print(temp)
202   prec <- as.numeric(joined[i,grep("Nieder+", cols)])
203   sun <- as.numeric(joined[i,grep("Sonnen+", cols)])
204   glob <- as.numeric(joined[i,grep("Global+", cols)])
205   feu <- as.numeric(joined[i,grep("Luftfeu+", cols)])
206   wind <- as.numeric(joined[i,grep("Windgesch+", cols)])
207
208   if(!is.na(temp) || is.na(prec) || is.na(sun))
209
210     #index[i] <- 0.4*temp + 0.2*prec + 0.05*sun
211
212   index[i] <- temp_cont(temp, temp_wgt, temp_
213     glob_cont(glob, glob_wgt, glob_max) + feu
214 }

```

R-Skript



QGIS-Raster

# Resultat

> Link: <http://bs14v076.opendata.iwi.unibe.ch>



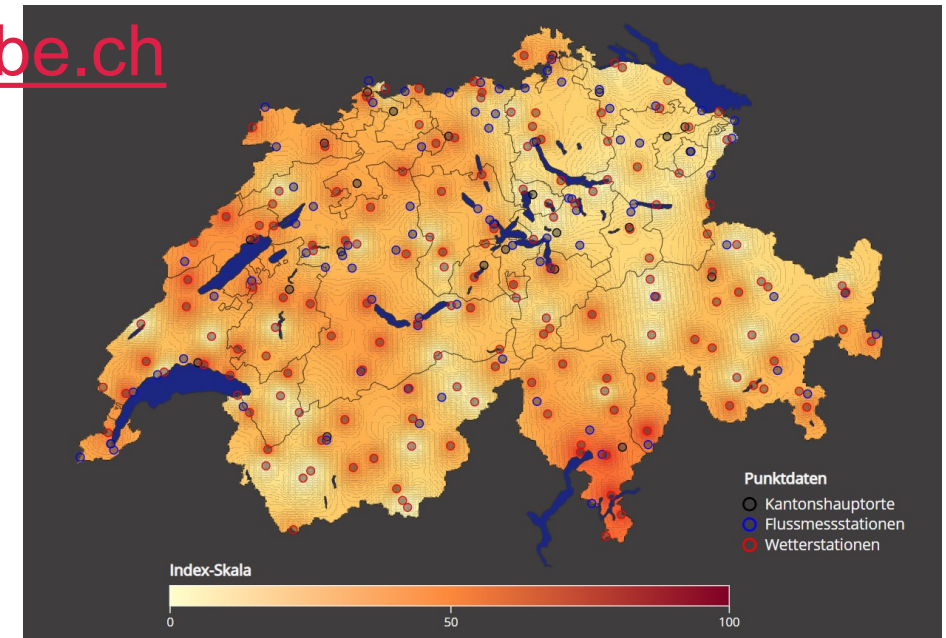
GitHub: @taetscher, @codicolus

> Visualisierungsart:

— Karte

> Funktionalitäten:

- Variable Interpolationsgenauigkeit
- Benutzerdefinierte, dynamische Berechnung des Index
- Anzeige aktueller Messdaten per Tooltip



# Fazit

- > Was haben wir gelernt?
  - Programmieren von Web-Applikationen
  
- > Was würden wir nächstes Mal anders machen?
  - Ressourceneffizienz auf Benutzerseite steigern (lange Ladezeiten von Rasterdaten)
  
- > Was bringt uns diese Übung?
  - Erweiterung unserer Programmierfähigkeiten
  - Verbindung von Geospatial Analysis Tools und web-basierten Visualisierungsmöglichkeiten

