Übung 4: Komplexe Rasterdaten

CAS FAB: Räumliche Daten in R

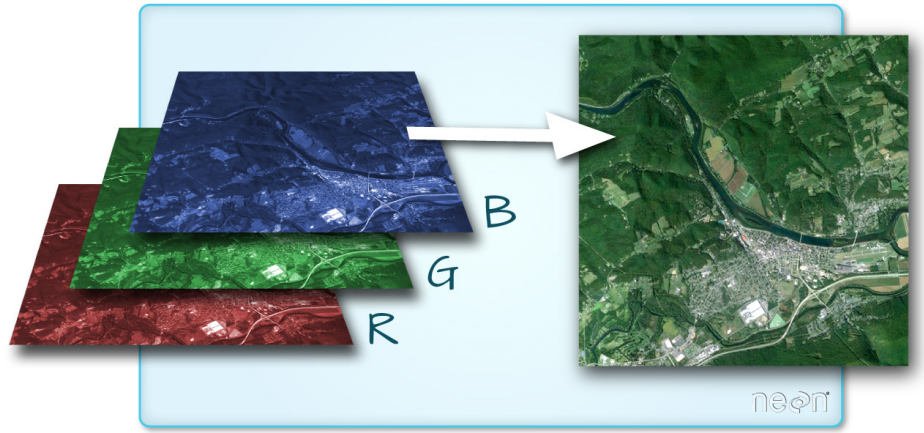
Nils Ratnaweera

Forschungsgruppe Geoinformatik

2021-11-30

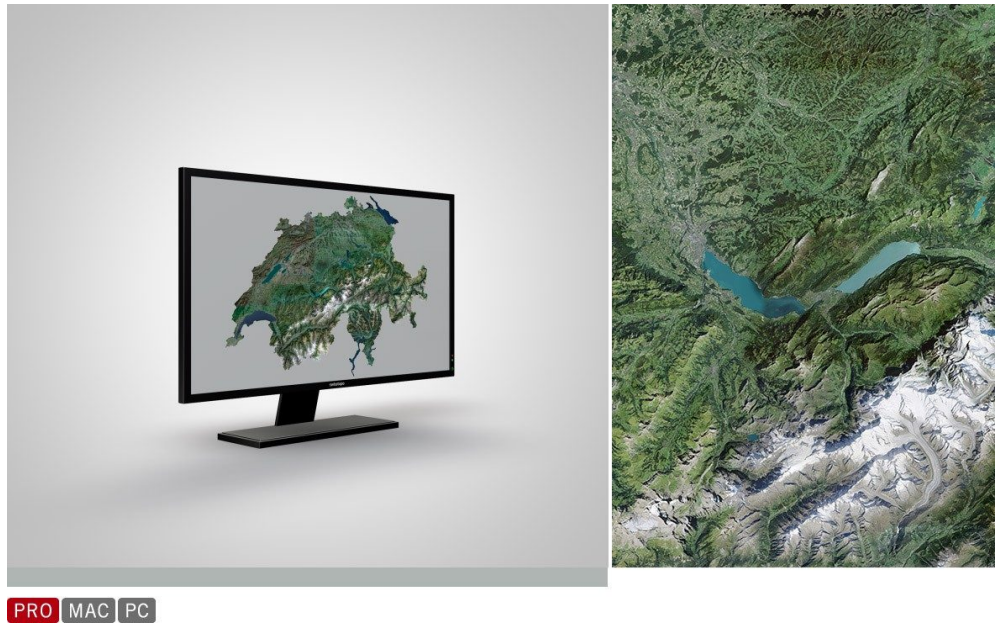
Input: Komplexe Rasterdaten

- Satelliten und Drohnen nehmen meist verschiedene Spektren von Elektromagnetischen Wellen auf
- diese Spektren werden in unterschiedlichen Datensätzen abgespeichert
- diese Datensätze müssen wieder zusammengefügt werden um ein Gesamtbild zu erhalten
- Beispiel: Rot, Grün und Blau werte fügen sich zu einem Farbluftbild zusammen



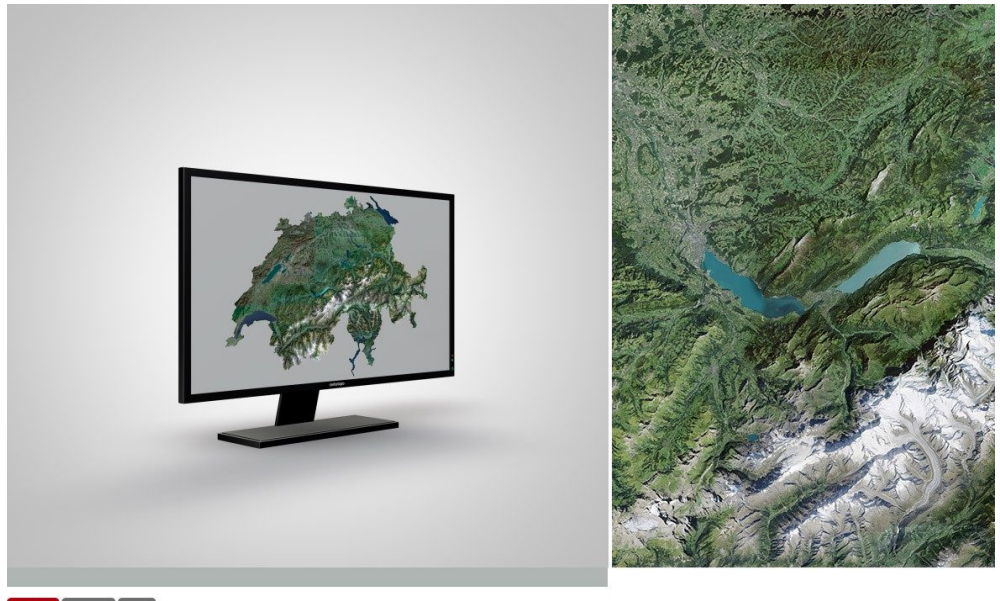
Übung 4.1

- Ladet euch den Datensatz `swissimage 25` von Swisstopo herunter
- Shortlink: https://bit.ly/3qGTjXN
- Enzipped den Inhalt in euer RStudio Projekt und schaut den Inhalt an
  - Was ist das Koordinatenbezugssystem? Wie hoch ist die räumliche Auflösung?



Übung 4.1

- Ladet euch den Datensatz `swissimage 25` von Swisstopo herunter
- Shortlink: https://bit.ly/3qGTjXN
- Enzipped den Inhalt in euer RStudio Projekt und schaut den Inhalt an
  - Was ist das Koordinatenbezugssystem? Wie hoch ist die räumliche Auflösung?



Lösung

1. Koordinatenbezugssystem: EPSG 2056
2. räumliche Auflösung: 25m

Übung 4.2

- Erstelle ein neues R Script mit dem Namen `Uebung_4.R`
- Lade die libraries `sf`, `tmap` und `terra`.
- Importiere den Swissimage Datensatz
- Weise dem importierten Datensatz das Korrekte Koordinatenbezugssystem zu
- Schau dir den Datensatz in der Konsole sowie mit `plot()` an

Übung 4.2

- Erstelle ein neues R Script mit dem Namen `Uebung_4.R`
- Lade die libraries `sf`, `tmap` und `terra`.
- Importiere den Swissimage Datensatz
- Weise dem importierten Datensatz das Korrekte Koordinatenbezugssystem zu
- Schau dir den Datensatz in der Konsole sowie mit `plot()` an

Lösung

```
library(sf)
library(tmap)
library(terra)

swissimage <- rast("_data/original/swissimage25/SWISSIMAGE25m/SI25-2012-2013-2014.tif")

crs(swissimage) <- "epsg: 2056"
```

Übung 4.2

```
plot(swissimage)
```

Übung 4.2

```
swissimage
```

```
## class       : SpatRaster
## dimensions  : 9480, 14000, 3  (nrow, ncol, nlyr)
## resolution  : 25, 25  (x, y)
## extent      : 2484375, 2834375, 1062000, 1299000  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=somerc +lat_0=46.9524055555556 +lon_0=7.43958333333333 +k_0=1 +x_0=2600000 +y_0=
## source      : SI25-2012-2013-2014.tif
## red-grn-blue: 1, 2, 3
## names       : SI2_1, SI2_2, SI2_3
```

Input: RGB Plots mit `tmap`

Input: RGB Plots mit `tmap`

- Um ein `rgb` Datensatz mit `tmap` zu plotten, verwenden wir nicht mehr `tm_raster()` sondern `tm_rgb`

Input: RGB Plots mit `tmap`

- Um ein `rgb` Datensatz mit `tmap` zu plotten, verwenden wir nicht mehr `tm_raster()` sondern `tm_rgb`

```
tm_shape(swissimage) +
  tm_rgb()
```

Input: Seltsamer bug mit `tmap`

Input: Seltsamer bug mit `tmap`

- Obwohl wir das Koordinatenbezugssystem korrekt gesetzt haben, kann `tmap` diese nicht interpretieren

Input: Seltsamer bug mit `tmap`

- Obwohl wir das Koordinatenbezugssystem korrekt gesetzt haben, kann `tmap` diese nicht interpretieren
- dies äussert sich in der folgenden Warnung:

Input: Seltsamer bug mit `tmap`

- Obwohl wir das Koordinatenbezugssystem korrekt gesetzt haben, kann `tmap` diese nicht interpretieren
- dies äussert sich in der folgenden Warnung:
- `Current projection of shape swissimage unknown and cannot be determined.`

Input: Seltsamer bug mit `tmap`

- Obwohl wir das Koordinatenbezugssystem korrekt gesetzt haben, kann `tmap` diese nicht interpretieren
- dies äussert sich in der folgenden Warnung:
- `Current projection of shape swissimage unknown and cannot be determined.`
- Hier kann ich euch nur folgenden (unbefriedigenden) Workaround anbieten

Input: Seltsamer bug mit `tmap`

- Obwohl wir das Koordinatenbezugssystem korrekt gesetzt haben, kann `tmap` diese nicht interpretieren
- dies äussert sich in der folgenden Warnung:
- `Current projection of shape swissimage unknown and cannot be determined.`
- Hier kann ich euch nur folgenden (unbefriedigenden) Workaround anbieten

```
swissimage <- project(swissimage, "epsg: 2056")
```

```
# zwar meldet `tmap` nun "Discarded datum CH1903+ in Proj4 definition"
# dies muss uns nicht kümmern
tm_shape(swissimage) +
  tm_rgb()
```

```
## stars_proxy object shown at 1216 by 823 cells.
```

Übung 4.3

Exportiert `swissimage` als `tif`-File

Übung 4.3

Exportiert `swissimage` als `tif`-File

Lösung

```
terra::writeRaster(swissimage, "_data/processed/swissimage.tif", overwrite = TRUE)
```

Input

Input

- Heute haben wir das Höhenmodell `dhm200` importiert

Input

- Heute haben wir das Höhenmodell `dhm200` importiert
- Höhenmodell mit 200m Auflösung (→ grob!)

Input

- Heute haben wir das Höhenmodell `dhm200` importiert
- Höhenmodell mit 200m Auflösung (→ grob!)
- swisstopo stellt zusätzlich das `dhm25` mit 25m Auflösung zur Verfügung (https://bit.ly/3kFgZrF)

Input

- Heute haben wir das Höhenmodell `dhm200` importiert
- Höhenmodell mit 200m Auflösung (→ grob!)
- swisstopo stellt zusätzlich das `dhm25` mit 25m Auflösung zur Verfügung (https://bit.ly/3kFgZrF)
- durch die höhere Auflösung dauert das transformieren in ein neues Koordinatensystem etwas länger

Übung 4.4

- Ladet euch das das `dhm25` mit 25m Auflösung herunter (https://bit.ly/3kFgZrF)
- importiert es in R
- setzt das korrekte CRS
- transformiert es in EPSG 2056 und verwendet dabei folgende Optionen:
  - mit `filename =` den Output direkt in ein File speichern
  - mit `progress = TRUE` den Fortschritt anzeigen lassen
- visualisiert es mit `tmap`

Übung 4.4

- Ladet euch das das `dhm25` mit 25m Auflösung herunter (https://bit.ly/3kFgZrF)
- importiert es in R
- setzt das korrekte CRS
- transformiert es in EPSG 2056 und verwendet dabei folgende Optionen:
  - mit `filename =` den Output direkt in ein File speichern
  - mit `progress = TRUE` den Fortschritt anzeigen lassen
- visualisiert es mit `tmap`

Lösung

```
dhm25 <- rast("_data/original/DHM25_MM_ASCII_GRID/ASCII_GRID_1part/dhm25_grid_raster.asc")
crs(dhm25) <- "epsg: 21781"

terra::project(dhm25, "epsg: 2056", filename = "_data/dhm25_2056.tif", progress = TRUE)
```

Übung 4.5 (Optional und Open End)

Suche dir auf den gängigen Portalen (s.u.) einen spannenden Datensatz und visualisiere diesen

- opendata.swiss
- map.geo.admin.ch
- swisstopo.admin.ch

Übung 4.6 (Optional und Open End)

- Lade dir swissimage daten in der Auflösung von 2m herunter und importiere sie in `R`
- Achtung! Sehr anspruchsvoll!!
- Tipps: du brauchst dazu:
  - `list.files()`
  - `lapply`
  - `do.call`
  - `mosaic`

Lösung

Daten herunterladen:

```
links <- read.csv("_data/original/swissimage_2m_landquart/swissimage_2m_urls.csv",header = FALS

links <- links$V1

filenames <- basename(links)

for (i in seq_along(links)) {
  download.file(links[i], file.path("_data/original/swissimage_2m_landquart/", filenames[i]))
}
```

Daten importieren:

```
swissimage_paths <- list.files("_data/original/swissimage_2m_landquart",full.names = TRUE)

swissimage_list <- lapply(swissimage_paths, function(x){rast(x)})

swiss_mosaic <- do.call(terra::mosaic, swissimage_list)

writeRaster(swiss_mosaic, "_data/processed/swissimage_2m_landquart.tif", overwrite = TRUE)
```