R

Emanuel Huber

Februar 26, 2018

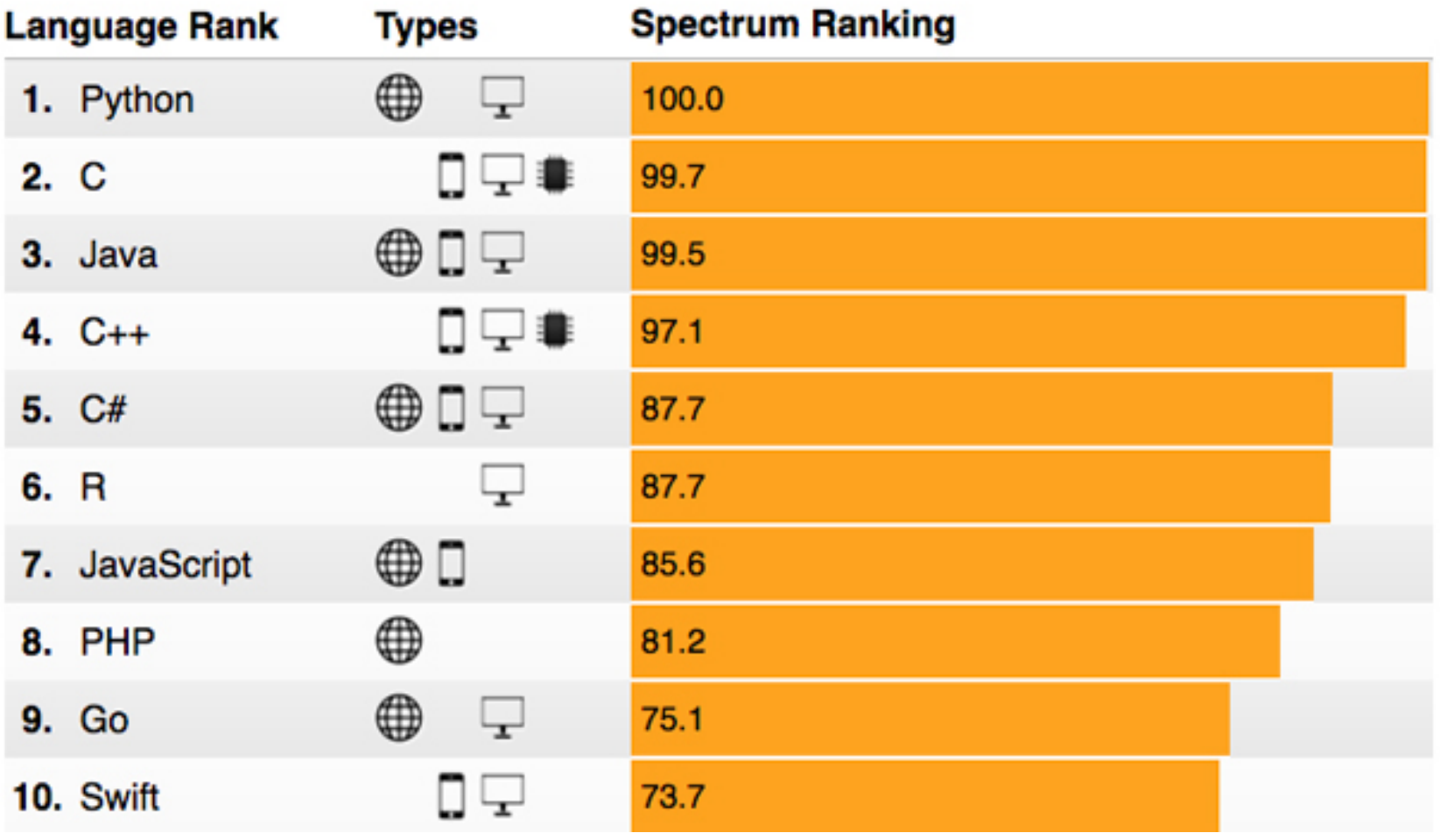
Structure

- Talk:
  1. Overview of R and potential use for AUG-group
  2. Some important concept on R language
- Resources
  - for each task (on https://emanuelhuber.github.io/Rcourse/
    - list of best R-package
    - cheat sheets
    - online book/course/tutorial
  - some tutorial on P:/RKurs (including data)
- What happens next?

Introduction

Programming language ranking by the IEEE (2017)

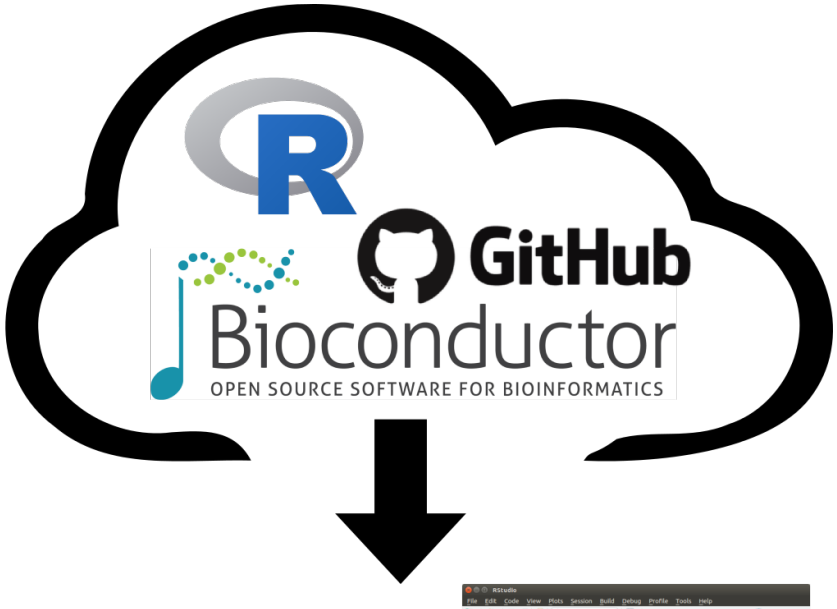
Institute of Electrical and Electronics Engineers ranking

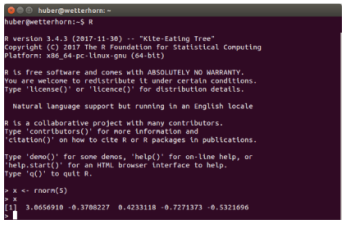


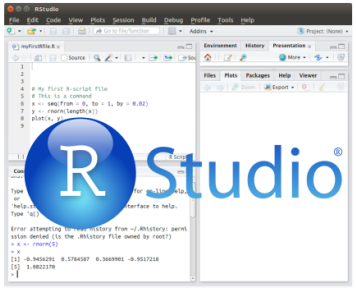
Why R is so popular

- free and open-source (no licence)
- runs on Linux, Windows and MacOS.
- large community
- many packages available that are documented (help files + vignette)
- can link C, C++, Fortran code
- excellent tools for data analysis (Google, Airbnb, Facebook, Microsoft...)
- high-quality graphics
- object-oriented programming

R Environment: GUI and packages







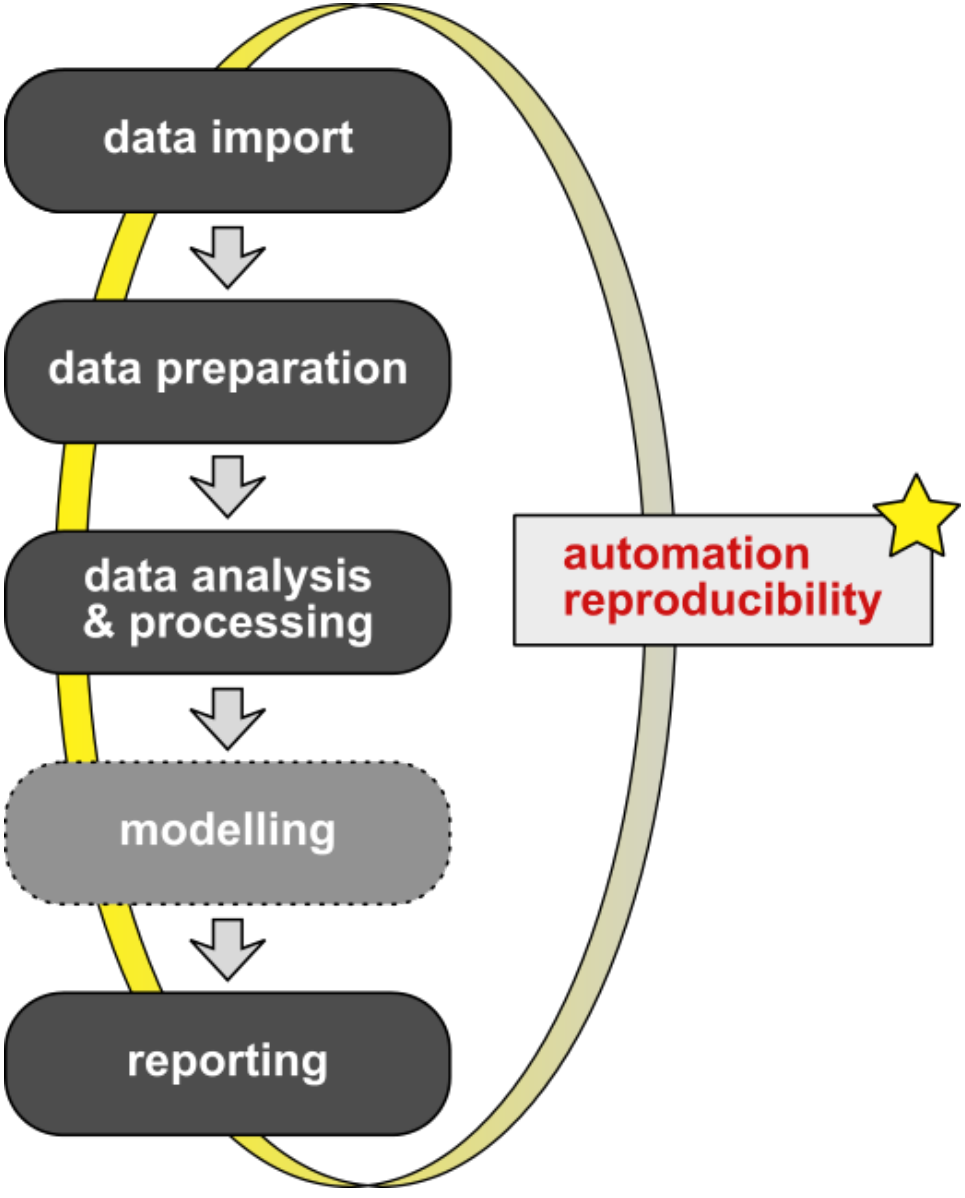
More than 12'000 packages!

CRAN Task Views

Bayesian	Bayesian Inference
ChemPhys	Chemometrics and Computational Physics
ClinicalTrials	Clinical Trial Design, Monitoring, and Analysis
Cluster	Cluster Analysis & Finite Mixture Models
DifferentialEquations	Differential Equations
Distributions	Probability Distributions
Econometrics	Econometrics
Environmetrics	Analysis of Ecological and Environmental Data
ExperimentalDesign	Design of Experiments (DoE) & Analysis of Experimental Data
ExtremeValue	Extreme Value Analysis
Finance	Empirical Finance
FunctionalData	Functional Data Analysis
Genetics	Statistical Genetics
Graphics	Graphic Displays & Dynamic Graphics & Graphic Devices & Visualization
HighPerformanceComputing	High-Performance and Parallel Computing with R
MachineLearning	Machine Learning & Statistical Learning
MedicalImaging	Medical Image Analysis
MetaAnalysis	Meta-Analysis
Multivariate	Multivariate Statistics
NaturalLanguageProcessing	Natural Language Processing
NumericalMathematics	Numerical Mathematics
OfficialStatistics	Official Statistics & Survey Methodology
Optimization	Optimization and Mathematical Programming
Pharmacokinetics	Analysis of Pharmacokinetic Data
Phylogenetics	Phylogenetics, Especially Comparative Methods
Psychometrics	Psychometric Models and Methods
ReproducibleResearch	Reproducible Research
Robust	Robust Statistical Methods
SocialSciences	Statistics for the Social Sciences
Spatial	Analysis of Spatial Data
SpatioTemporal	Handling and Analyzing Spatio-Temporal Data
Survival	Survival Analysis
TimeSeries	Time Series Analysis
WebTechnologies	Web Technologies and Services
gR	gRaphical Models in R

Potential use of R for AUG group

R strengths



- all-in-one tool
- no licence
- teaching, research, reporting
- automation and reproducible workflow

Data import/export

Resources - data import

Function	What It Does
`read.table()`	Reads any tabular data where the columns are separated `read.table(file = "filePath", sep = "\t", header = TRUE)`
`read.csv()`	A simplified version of read.table() to read CSV files. `read.csv(file = "filePath")`
`scan()`	Finer control over the read process when your data isn't tabular. `scan("filePath", skip = 1, nmax = 100)`
`readLines()`	Reads text from a text file one line at a time. `readLines("filePath")`
`read.fwf()`	Read a file with dates in fixed-width format. `read.fwf("filePath", widths = c(1, 2, 3)`
`readxl::read_excel()`	To read excel files (xls and xlsx), from `readxl` package `read_excel("filePath", sheet = "mtcars")`

Import/export

Data preparation

Resources - data cleaning

- **Package**
  - `dplyr`
- **Cheat sheet**
  - regular expression cheat sheet
  - `dplyr`: data transformation cheat sheet
- **Tutorials/book**
  - Tutorial: An introduction to data cleaning with R (53 p.)
  - Hands-On Data Science with R: Data Preparation
  - check `dplyr`-vignettes
  - Tutorial: Regular Expressions in R

Data cleaning 1

date	location	parameter	value
2011-01-12	RIVER	Head	2.31
2011-01-18	RIVER	nitrat	27.7
2011-01-18	GW_1	RIV_head	NA
2011-01-18	GW_1	nitrat	0.0083
2011-01-18	GW_1	sulfat	0
2011-01-18	GW_1	bromid	6.419
2011-01-18	GW_2	324/head	0.1226
2011-01-18	GW_2	nitrat	1.07
2011-01-18	GW_2	bromid	0
2011-03-02	RIVER	head	0.0194
2011-03-02	RIVER	nitrat	6.116
2011-03-02	RIVER	bromid	1.09
2011-03-02	GW_1	HEAD	0.0347

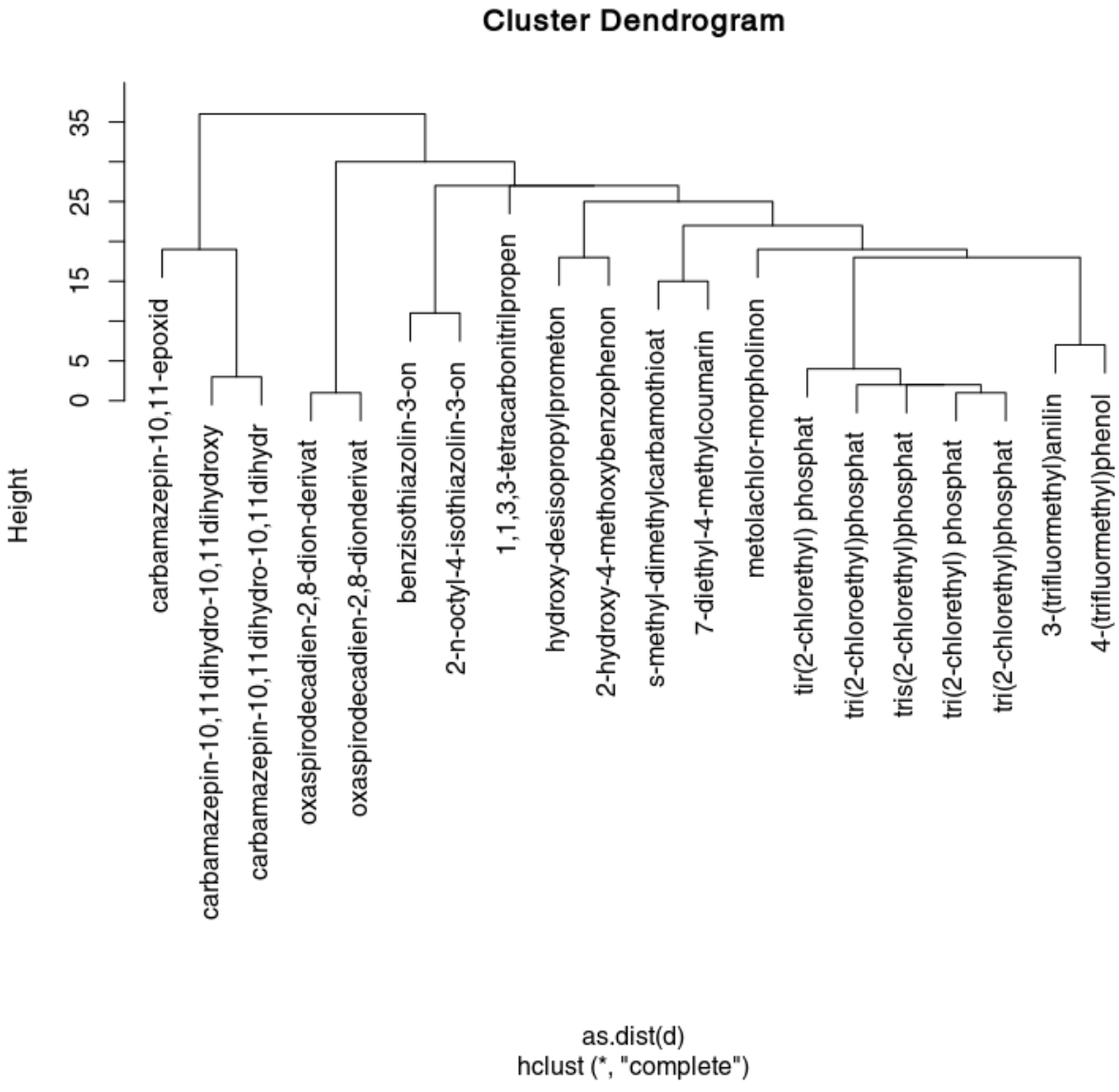
- remove rows/columns
- remove duplicates
- transform data
- correct for inconsistencies
- convert data type
- deal with missing values

- rename variables: `Head`, `RIV_head`, `324/head`, `head`, `HEAD` -> `head`

```
pattern <- "^RIV_|^[[:digit:]]+/"
x$parameter <- sub(pattern, "", x$parameter)
x$parameter <- tolower(x$parameter)
```

Data cleaning 2

detect typo errors > approximate string matching:



Resources - data shaping

- **Package**
  - `tidyr`
- **Cheat sheet**
  - `tidyr`: Tidy Data (see p. 2)
- **Tutorials/book**
  - Concepts of data tidying are well explained in the chapter Data Tidying from the book Data science with R.

Data shaping 1 (tidy data)

Tidy data satisfies three rules (see Data Science with R):

- Each variable in the data set is placed in its own column
- Each observation is placed in its own row
- Each value is placed in its own cell

Data shaping 2 (tidy data)

"Messy" data

date	location	parameter	value
2011-01-12	RIVER	head	2.31
2011-01-18	RIVER	nitrat	27.7
2011-01-18	GW_1	head	NA
2011-01-18	GW_1	nitrat	0.0083
2011-01-18	GW_1	sulfat	0
2011-01-18	GW_1	bromid	6.419
2011-01-18	GW_2	head	0.1226
2011-01-18	GW_2	nitrat	1.07
2011-01-18	GW_2	bromid	0
2011-03-02	RIVER	head	0.0194
2011-03-02	RIVER	nitrat	6.116
2011-03-02	RIVER	bromid	1.09
2011-03-02	GW_1	head	0.0347

Tidy data

date	location	head	nitrat	sulfat	bromid
2011-01-12	RIVER	2.31	27.7	NA	NA
2011-01-18	GW_1	NA	0.0083	0	6.419
2011-01-18	GW_2	0.1226	1.07	NA	0
2011-03-02	RIVER	0.0194	6.116	NA	1.09
2011-03-02	GW_1	0.0347	NA	NA	NA

```
x_tidy <- tidyr::spread(x, parameter, value)
```

Data analysis and processing

Data analysis

- data analysis
  - statistics: check
  - linear regression
  - boxplot
  - linear regression (p-value, R^2)
  - PCA (linear relationship)
  - cluster analysis
    - dendrogram nice tutorial, another tutorial
    - k-means algorithm

Sensitivity analysis

Time series - resources

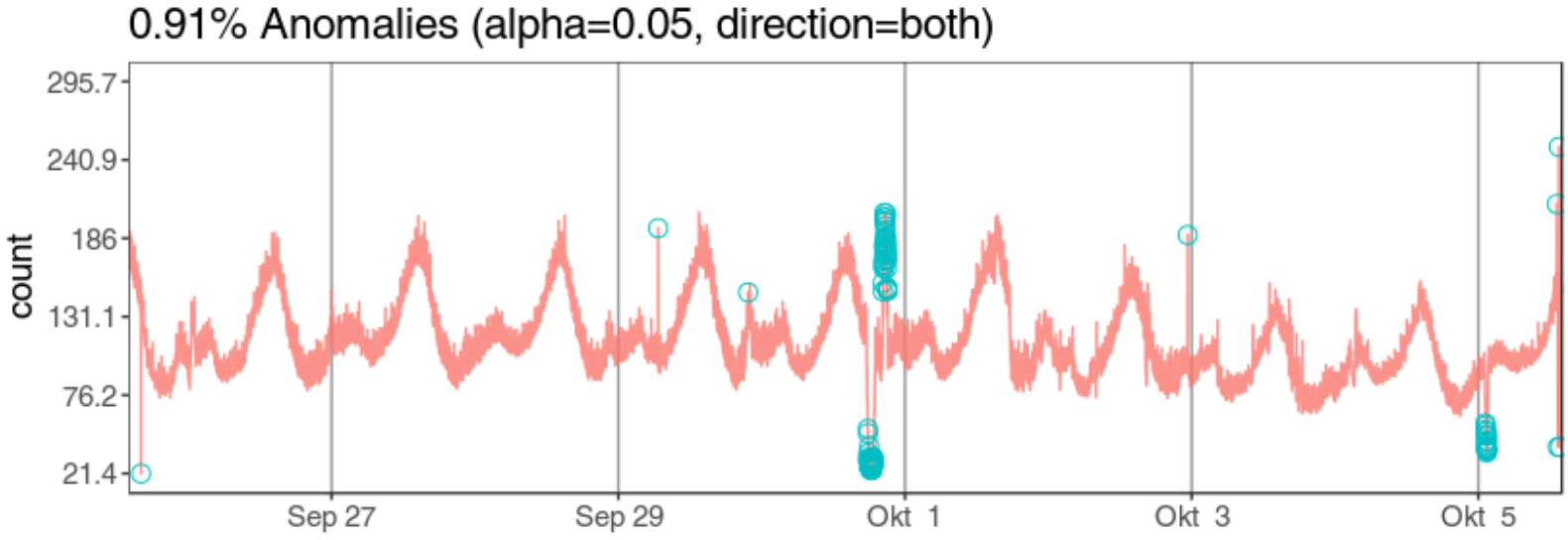
- **Package**
  - "eXtensible Time Series" `xts` that extends the `zoo` package
  - `zoo` for regular and irregular Time Series
  - `lubridate` to deal with date and time
- **Cheat sheet**
  - eXtensible Time Series: `xts`
  - How to deal with date and time: `lubridate` cheat sheet
- **Tutorials/book**
  - check `xts` vignettes
  - check `zoo` vignettes
  - Dates and Times Made Easy with `lubridate`

Check package for time-series processing:

- pracma
- signal

Time series - Import and preparation

- Import function (?)
- Detect and remove anomalies in time-series Application: remove anomalies and estimate true values

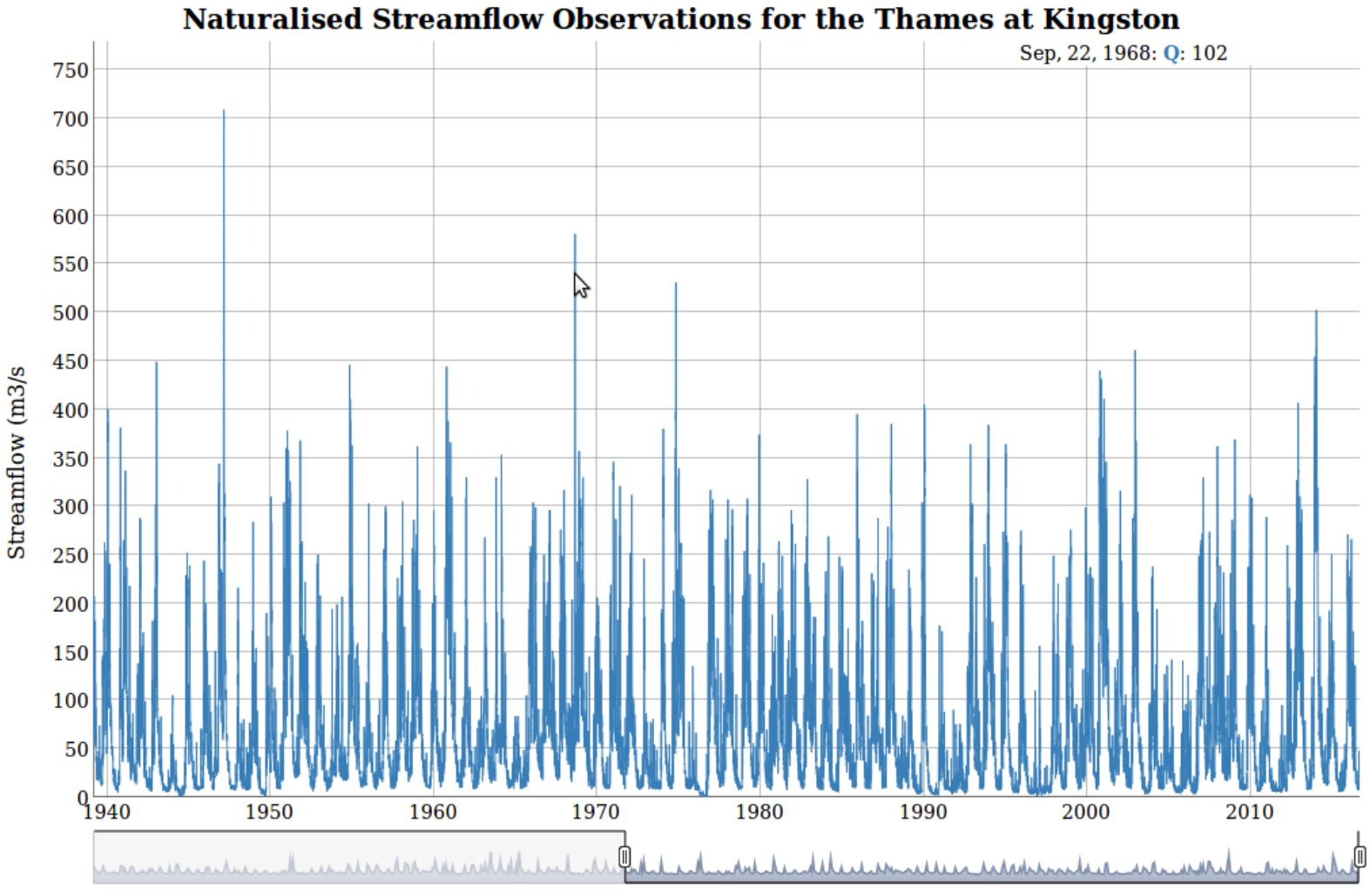


source

- fill in missing values (interpolation)
- regular time-step
  - smoothing
  - decimate

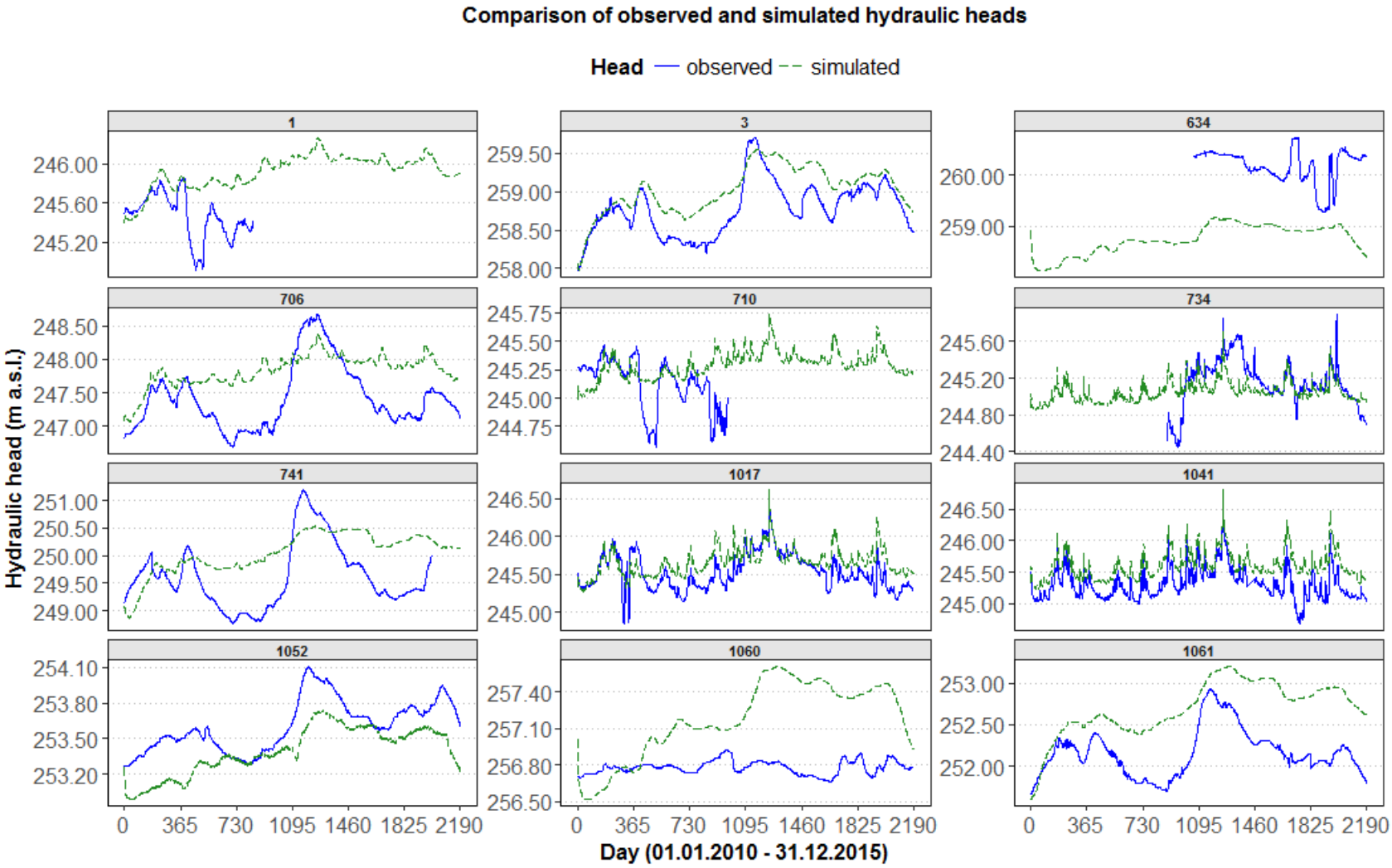
Time series - -visualizing (0)

dynamic time series visualisation

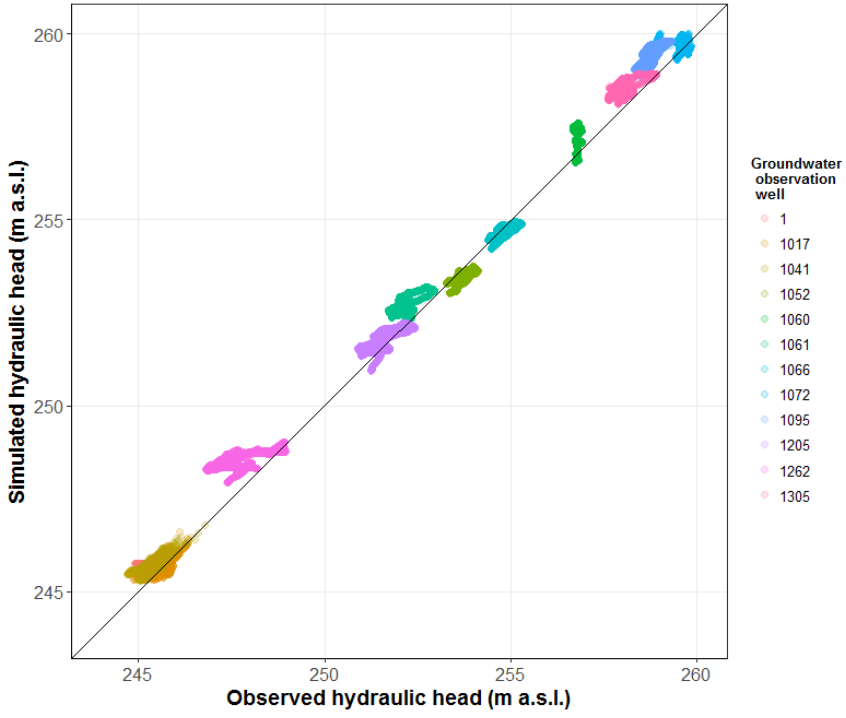


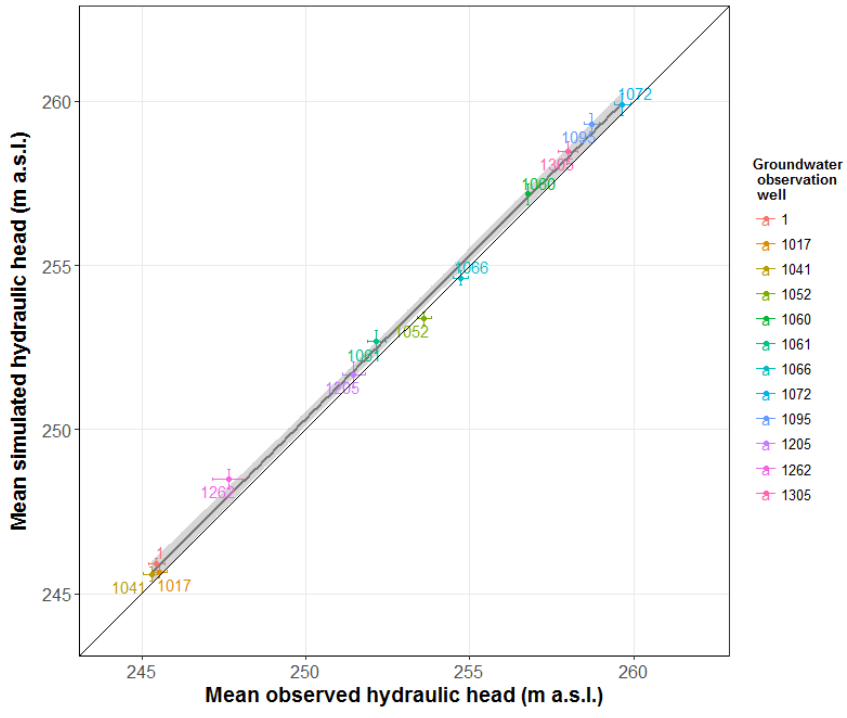
Time series - -visualizing (1)

Example: Groundwater head data from FEFLOW (merci MM) `P:/RKurs/...`

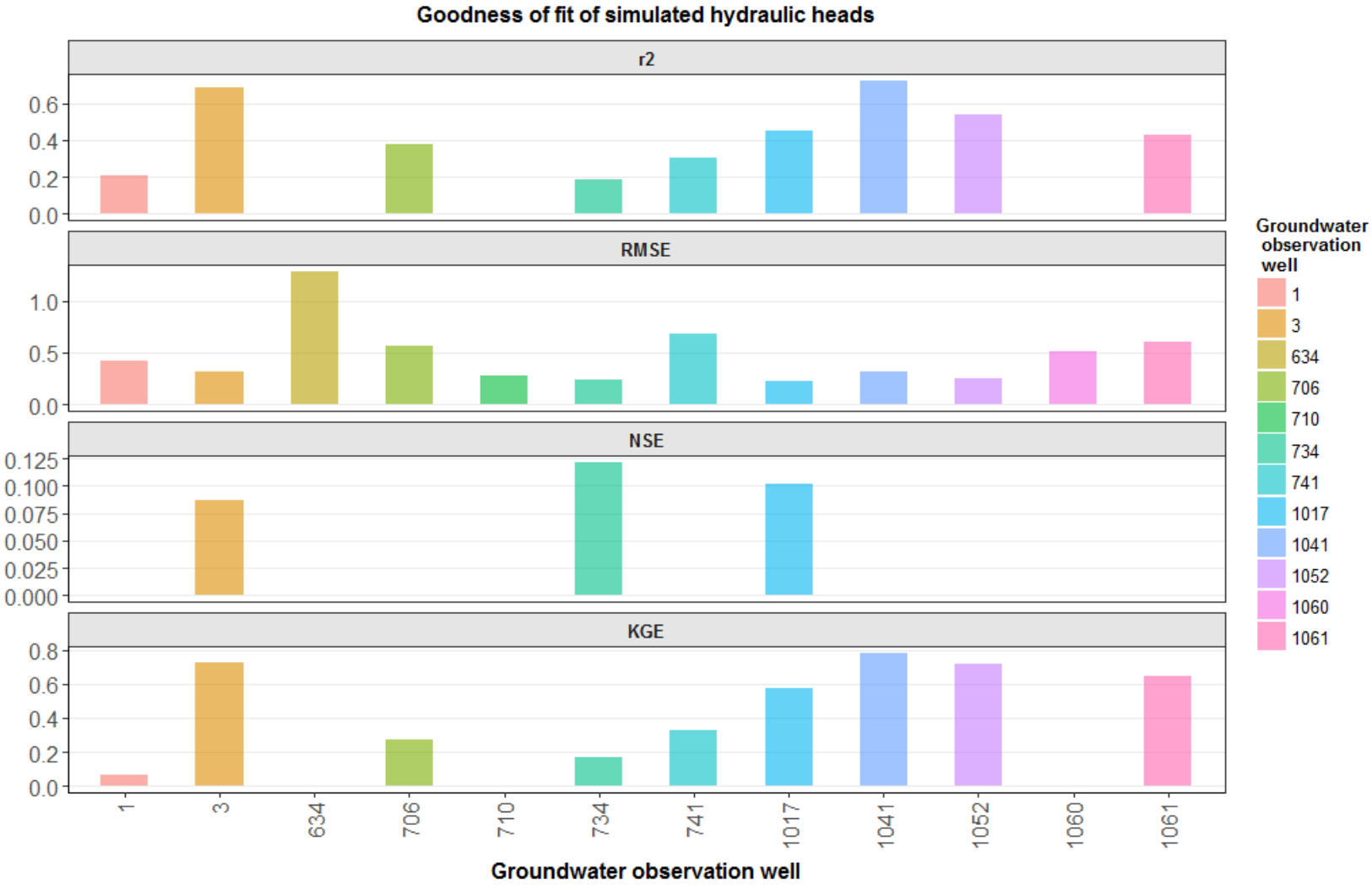


Time series - -visualizing (2)

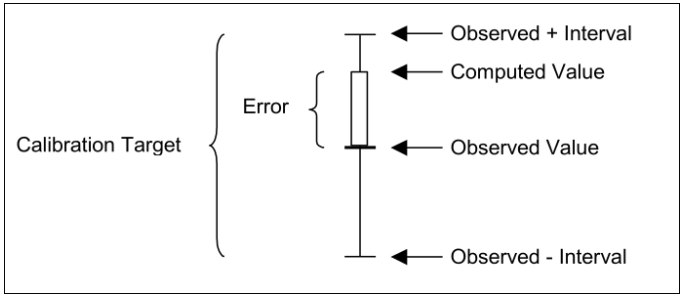


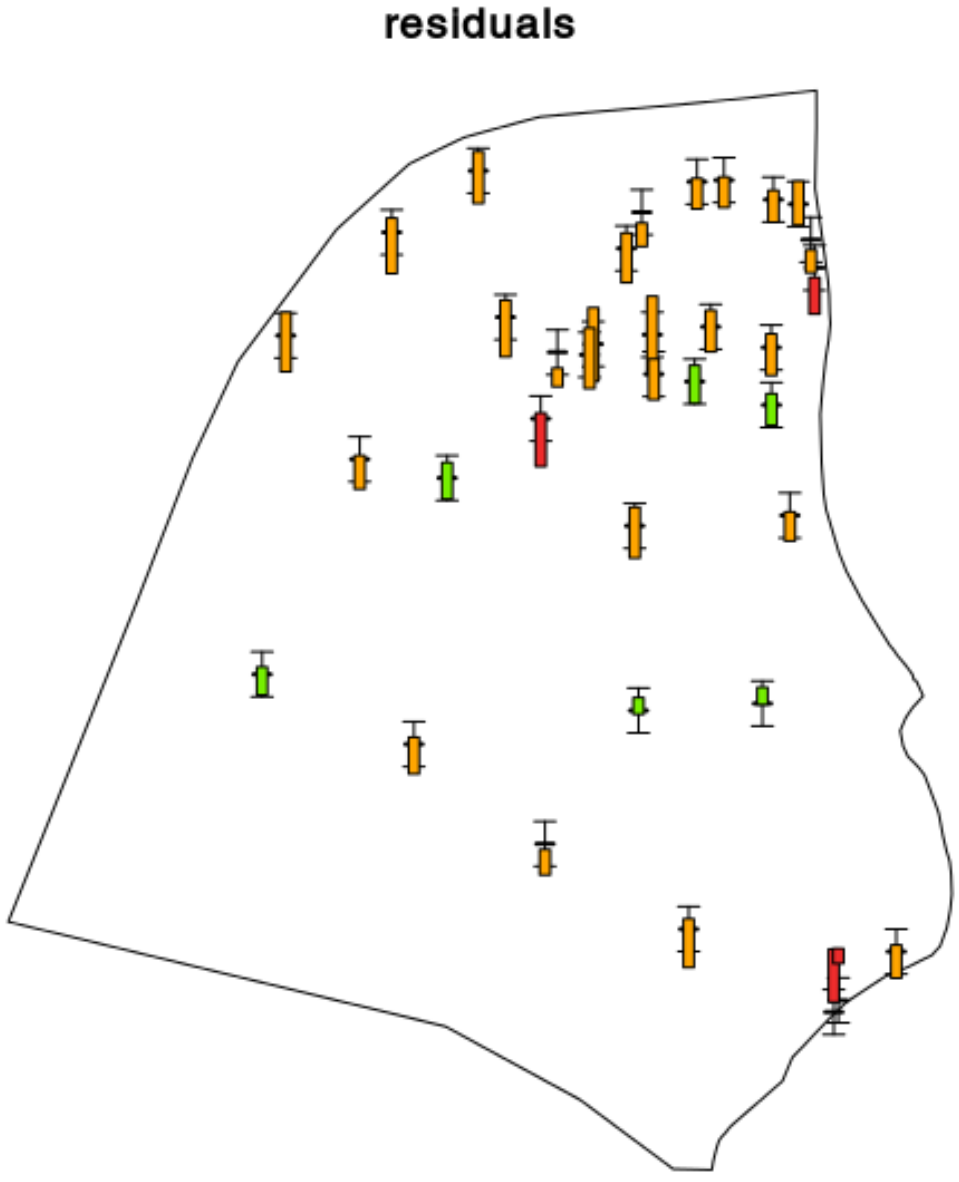


Time series - -visualizing (4)

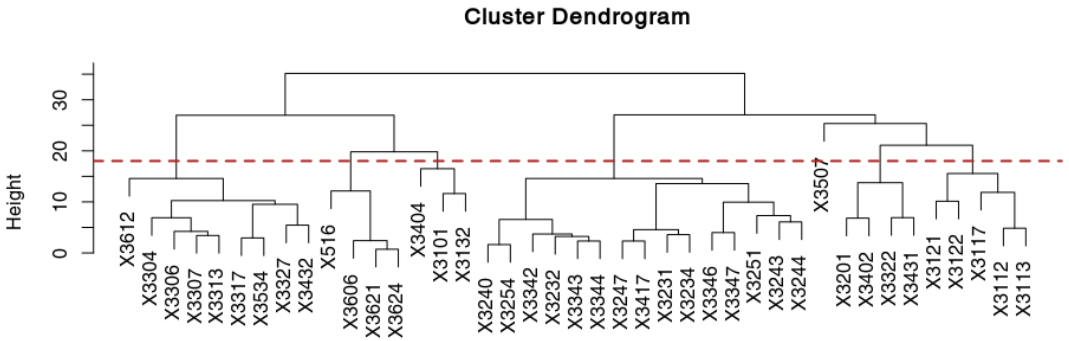


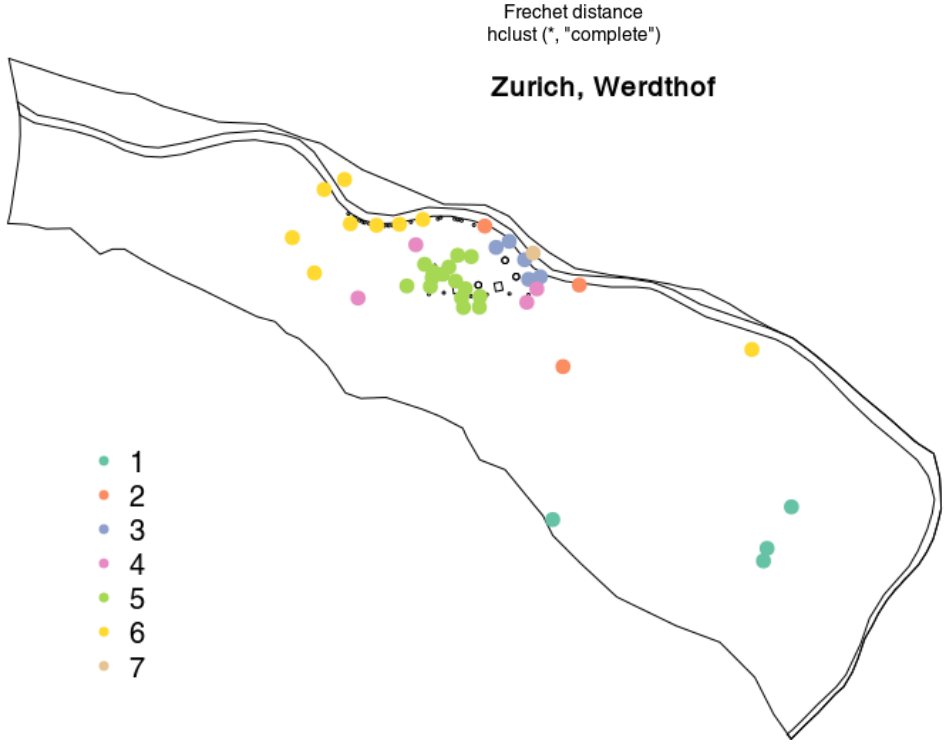
Time series - visualizing (5)





Time series - clustering





- remove obs with too many `NA`'s
- interpolate missing data `NA`
- compute distances between obs
- cut dendrogram –> classes
- plot onto map

Time series - statistics

- Daily, monthly, yearly values (cf. book: Analysis of ecological data with R)
- Auto-correlation function (ACF) and Cross-correlation function (CCF) See ARIMA model tutorial, tutorial2
- empirical cumulative distribution function

Time series - spectral analysis

- Fourier
- SSA
- ?
- wavelet analysis

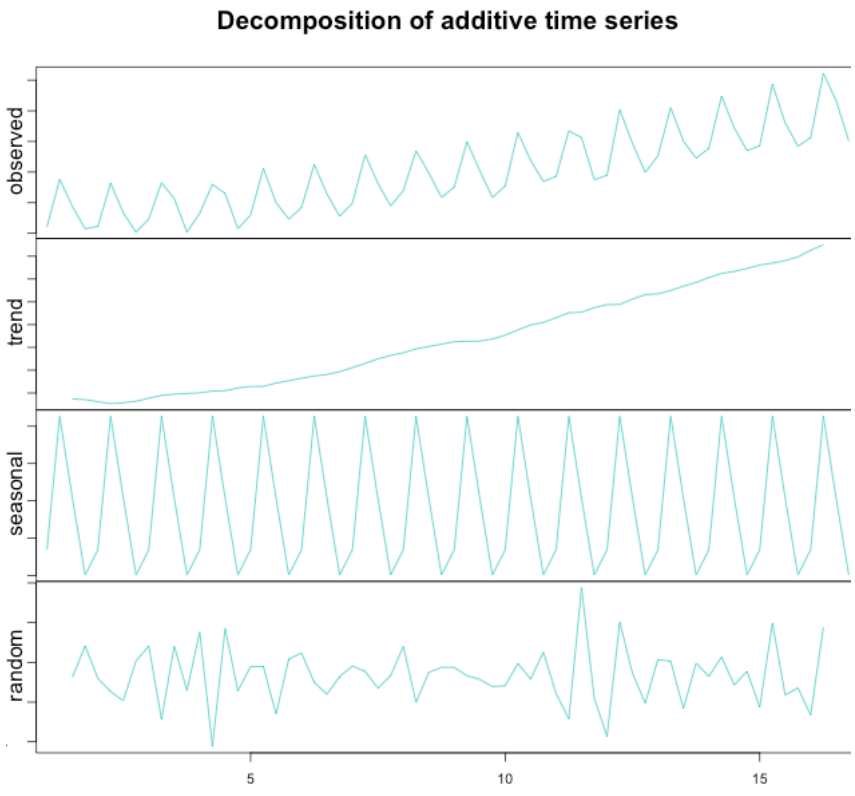
Time series - impulse response function

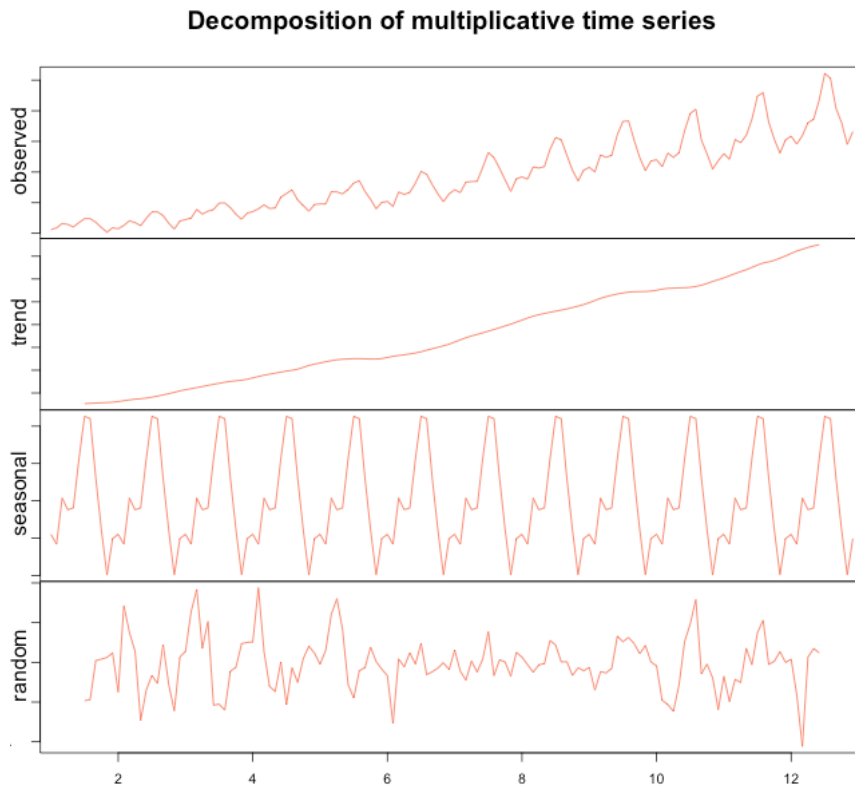
Check RRAWFLOW package rrawflow website

Time series - regression

- Regression

Time series - trend and seasonal components






```
x_dcp_add <- decompose(x, "additive")
x_dcp_mul <- decompose(x, "multiplicative")
```

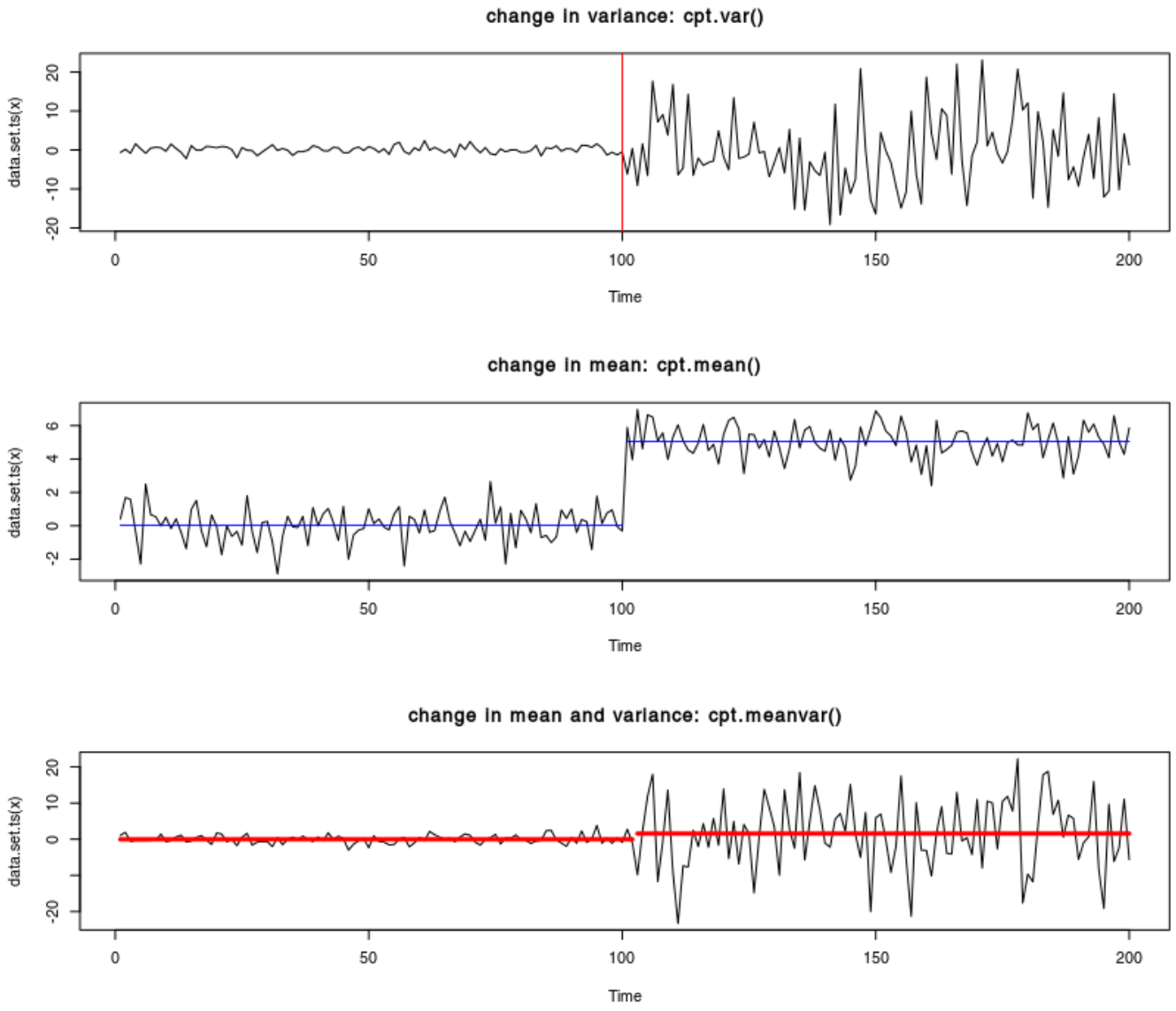
online tutorial

Time series - mining and clustering

Time series mining and clustering

- Compute distance between time-series
- Clustering
- Vizualisation (e.g., MDS)

Time series - change detection



package `changepoint`

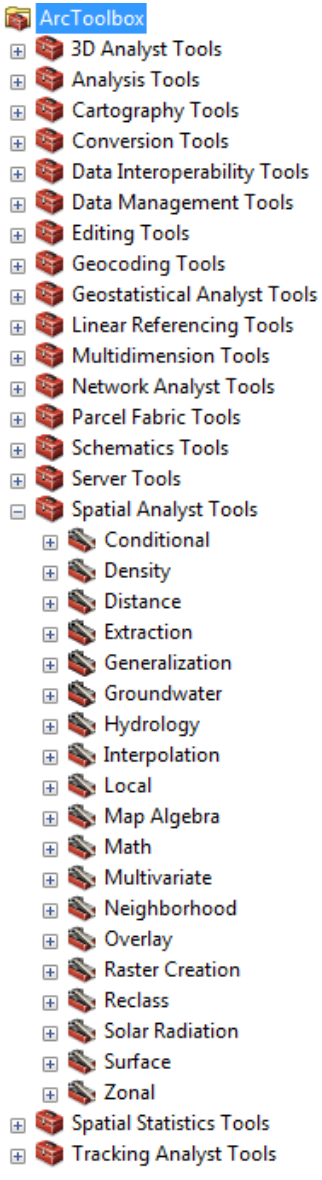
- Example: NF-Fisch Plot CP1 against CP2 and CP3: ellipses
- CD-Experiment

R GIS - resources

R is a GIS software (sp, rgeos, raster, rgdal) see also `sf` : chjeck

- **Package**
  - `sf` package for spatial vector data. Binds to GDAL for reading and writing data, to GEOS for geometrical operations, and to Proj.4 for projection conversions and datum (*will remplace sp/rgeos/rgdal in the long term* link)
  - `raster` package for raster, multi-band raster, ...
  - lidR for airborne LiDAR data manipulation and visualization for forestry application
  - packages to interact with existing GIS software link
    - `spgrass6`: Provides an interface between R and GRASS 6+. Allows for running R from within GRASS as well as running GRASS from within R.
    - `rgrass7`: Same as spgrass6, but for the latest version of GRASS, GRASS 7.
    - `RPyGeo`: A wrapper for accessing ArcGIS from R. Utilizes intermediate python scripts to fire up ArcGIS. Hasn't been updated in some time.
    - `RSAGA`: R interface to the command line version of SAGA GIS.
- **Cheat sheet**
  - eXtensible Time Series: `xts`
  - How to deal with date and time: `lubridate` cheat sheet
- **Tutorials/book**
  - Book: geocomputation with R
  - check `zoo` vignettes
  - Dates and Times Made Easy with `lubridate`

R GIS



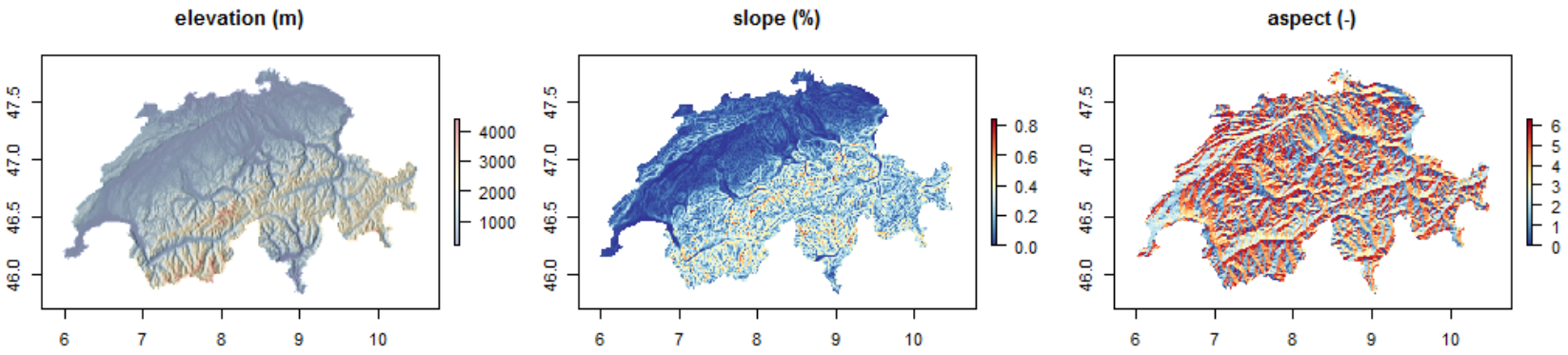
Tool

- manipulation raster
- manipulation feature
- raster interpolation:
  - book: R_Applied Spatial Data Analysis with R.pdf (chap. 8, 8.11)

Still missing

- TIN
- watershed but
  - check that
  - check how to compute flow accumulation
  - idea: use local structure tensor to compute direction and accumulation....
- download tiles webgis of Italy region

R GIS - DEM processing



R GIS - Feature projection

check graticules sf class

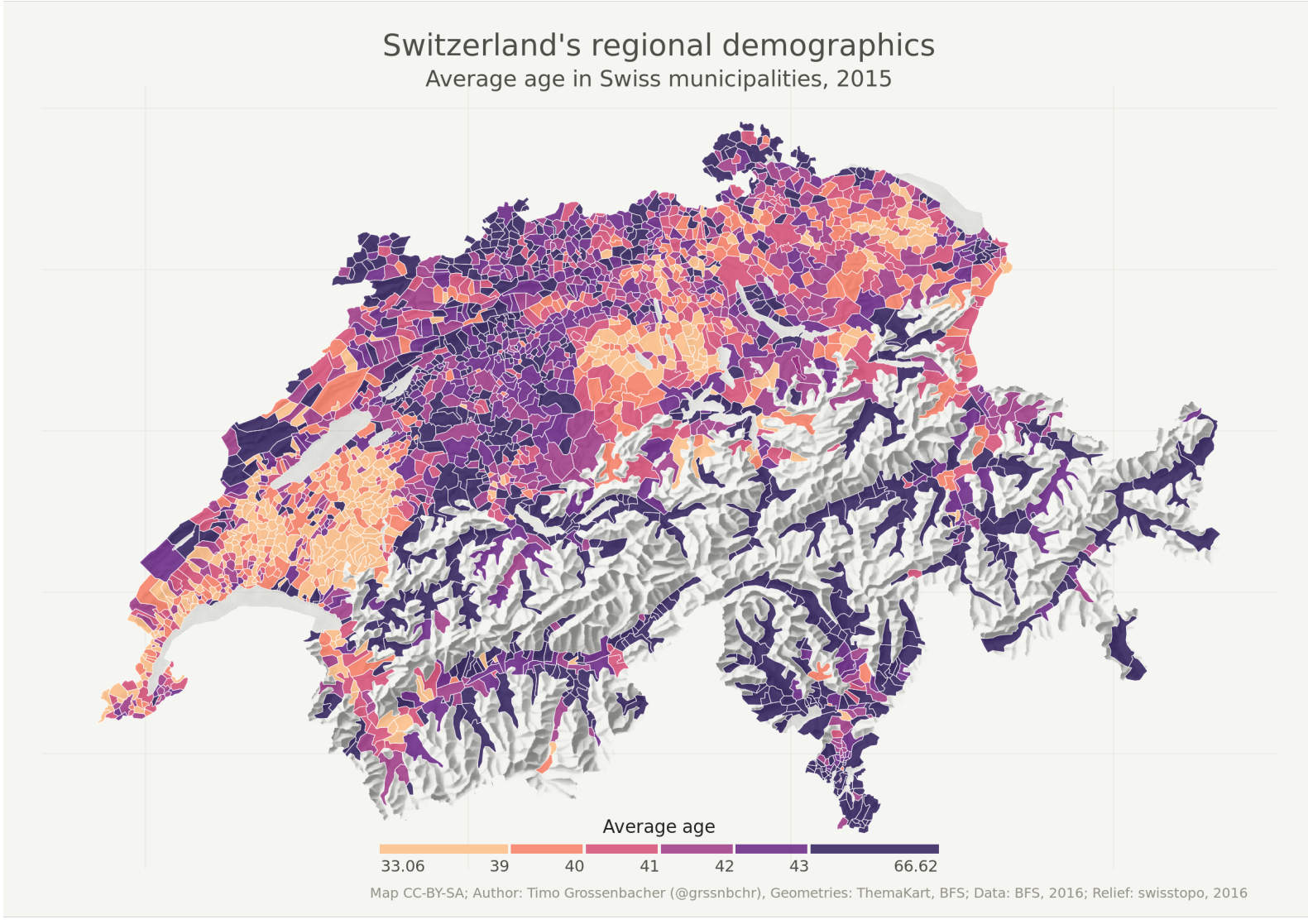
Mercator projection

Lambert Azimuthal Equal Area Projection

Lambert Conformal Conic Projection

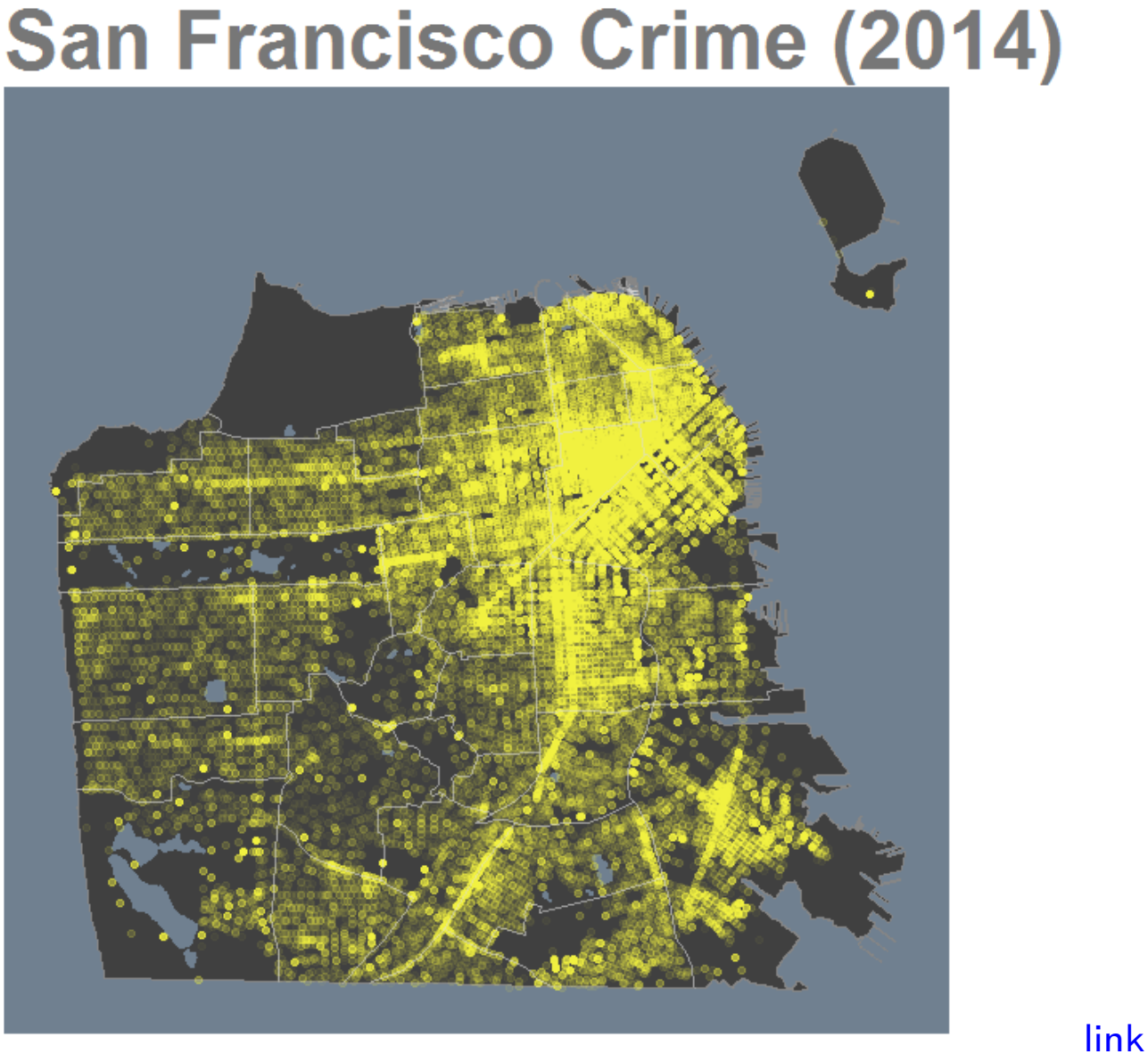
Albers Equal Area Conic

R GIS - Maps (1)



link

R GIS - Map (2)



link

R GIS - Spatial interpolation

- spatial interpolation
- gstat
- gstat
- ipdw
- automap package
- check that

R GIS - LiDAR

R package for airborne LiDAR data manipulation and visualisation for forestry application lidR

R GIS - making and using bathymetric maps in R with `marmap`

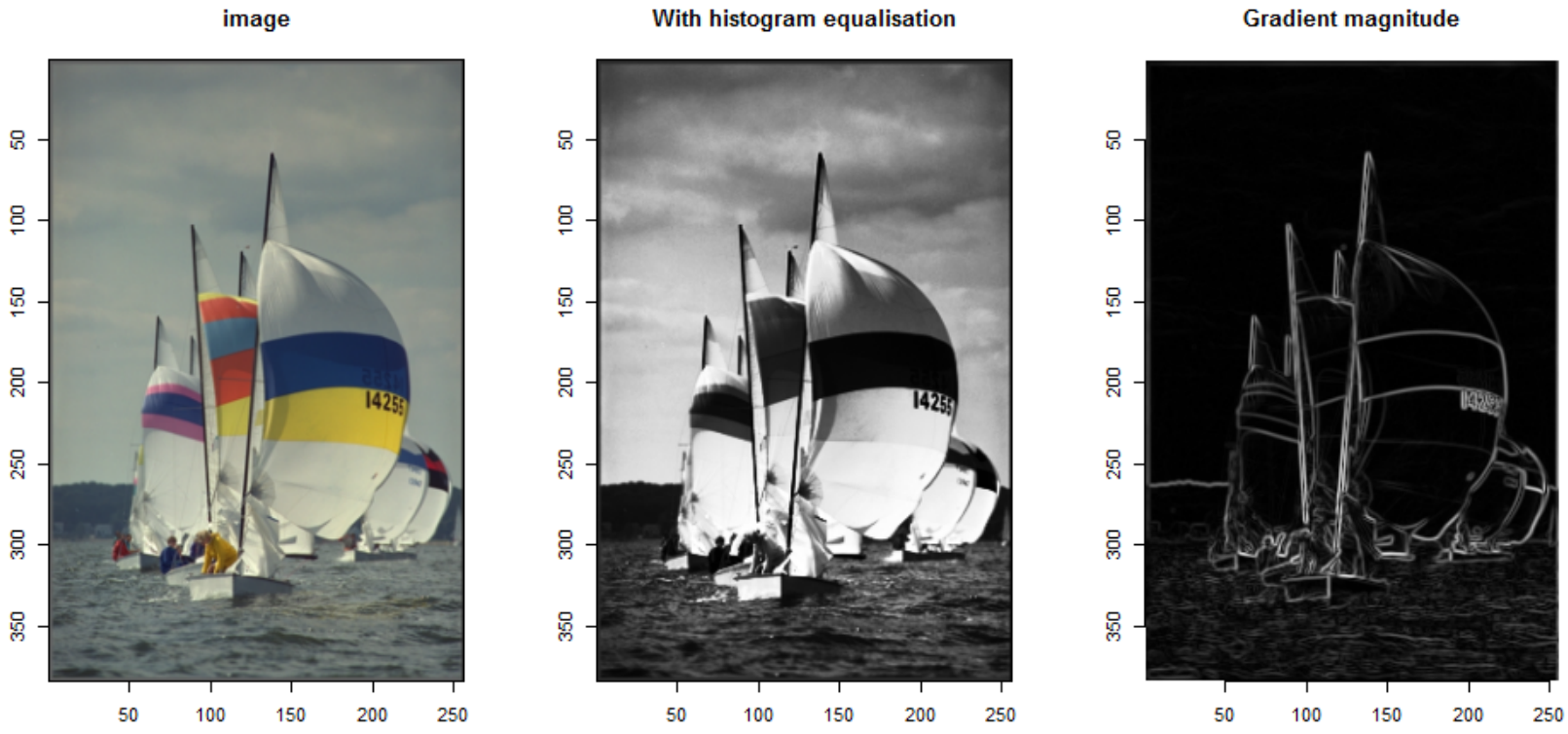
link

Image analysis and processing - resources

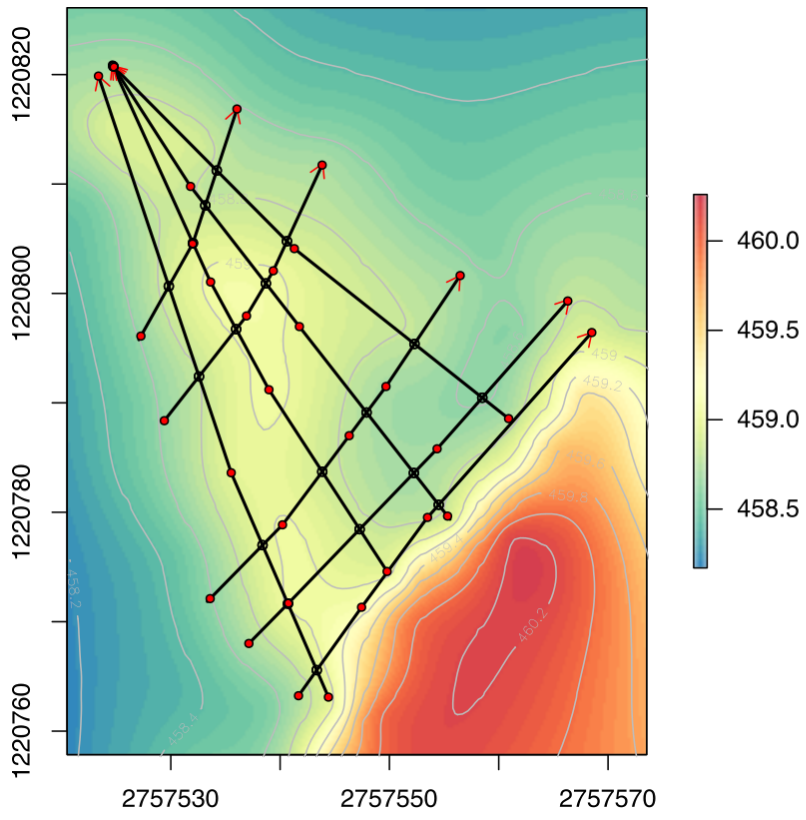
- **Package**
  - `imager`
  - `EBImage`
- **Tutorials/book**
  - check `imager` vignettes
  - `imager` project website
  - check `EBImage` vignettes

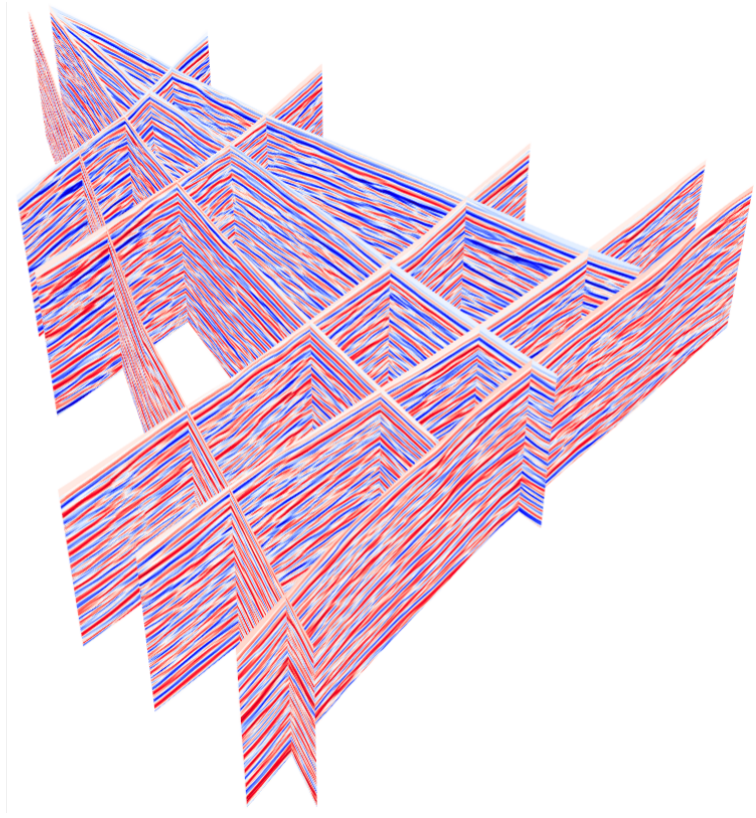
Image analysis and processing

Example: package `imager`

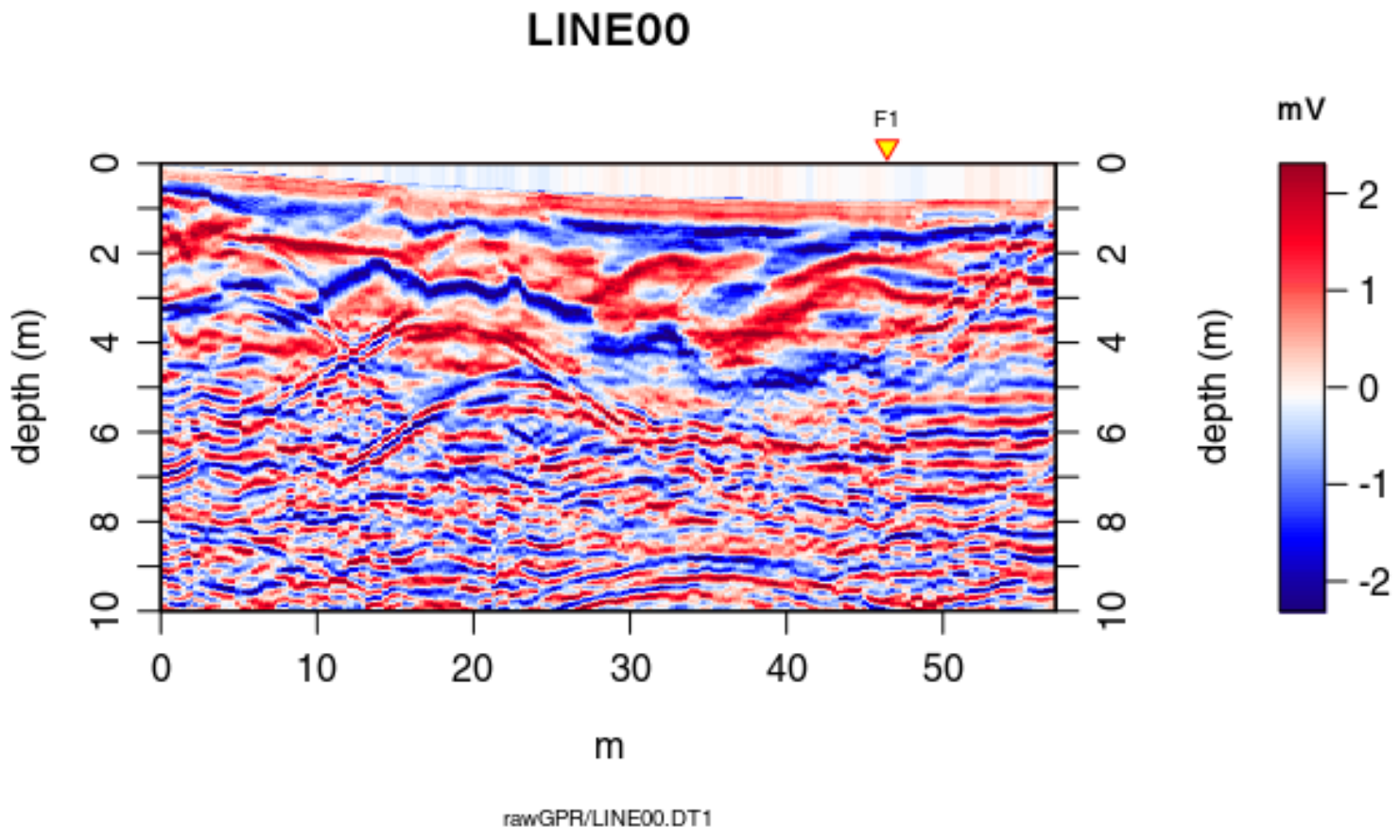


GPR data processing RGPR (1)





GPR data processing RGPR (2)



tutorial

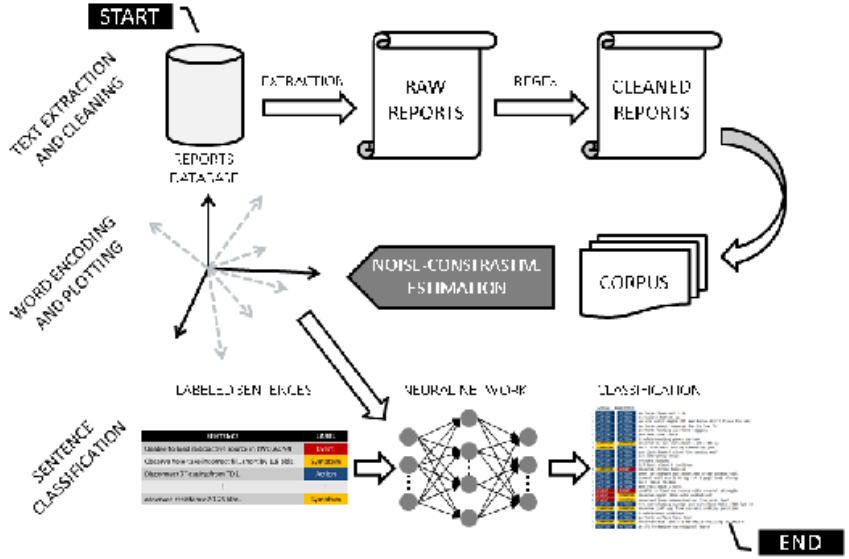
package

Seismic data

- seimic data analysis and processing
- eisis package
  - talk

Database

- quality control: detect “outliers”, bad quality data, errors
- statistics, plots, reporting
- text analytic meet geosciences



from arxiv.org/abs/1712.01476

- Regli et al. (2002) Interpretation of drill core and georadar data of coarse gravel deposits doi:10.1016/S0022-1694(01)00531-5

Misc

- Package `rioja`: Functions for the analysis of Quaternary science data, including constrained clustering, WA, WAPLS, IKFA, MLRC and MAT transfer functions, and stratigraphic diagrams.

Modeling/simulations

Statistical/stochastic models

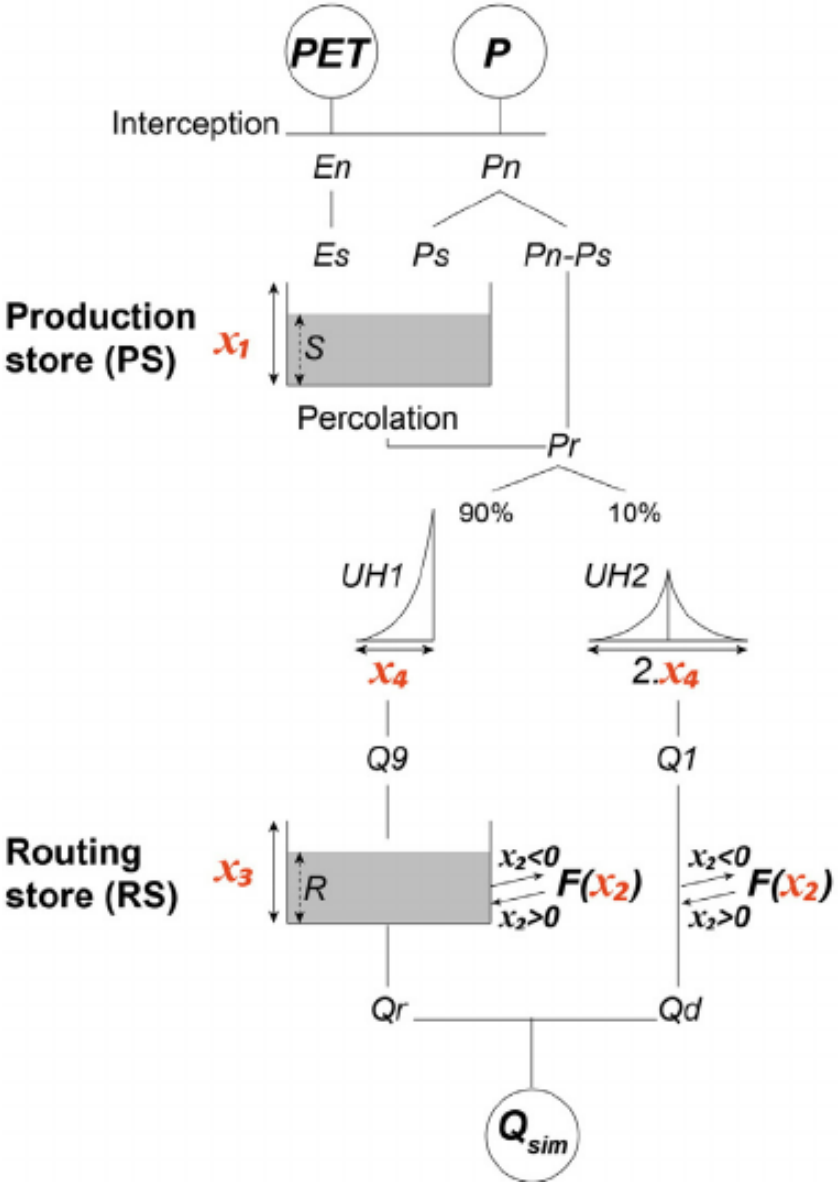
- probabilty distribution
- Gaussian process/Kriging (`RandomFields`): account for correlation
  - 3D subsurface heterogeneity model
- point process and marked point process
  - faults, `CBRDM`

Catchment flow simulation

- `RRAWFLOW`: Rainfall-Response Aquifer and Watershed Flow Model
- `topmodel` and `dynatopmodel` packages

Catchment flow simulation (1)

Package `airGR`, see tutorial on companion website



Simulation variables

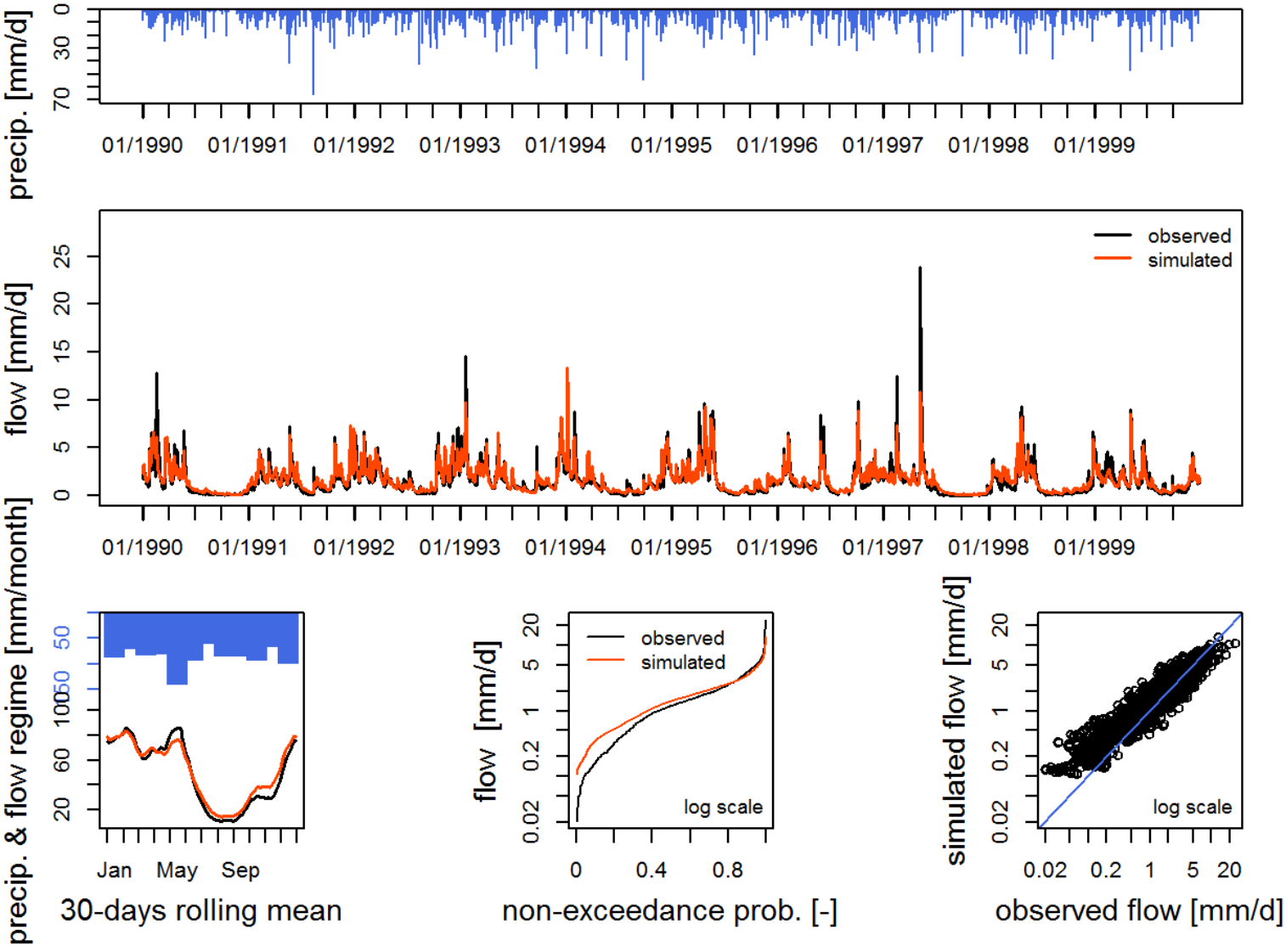
P: Precipitation
PET: Potential evapotranspiration
En: Net evapotranspiration capacity
Pn: Net precipitation
Es: Evaporation from the production store
Ps: Water filling the production store
S: Level in the production store
Pr: Water filling the routing store
UH1: Unit hydrograph 1
UH2: Unit hydrograph 2
Q9: Output of UH1
R: Level in the routing store
Q1: Output of UH2
Qr: Outflow of the routing store
F: Groundwater exchange term
Qd: Flow component from Q1 and F
Q_{sim}: Total simulated streamflow

Calibration parameters

x_1: maximum capacity of the PS (mm)
x_2: groundwater exchange coefficient (mm)
x_3: maximum capacity of the RS (mm)
x_4: time base of unit hydrograph UH1 (d)

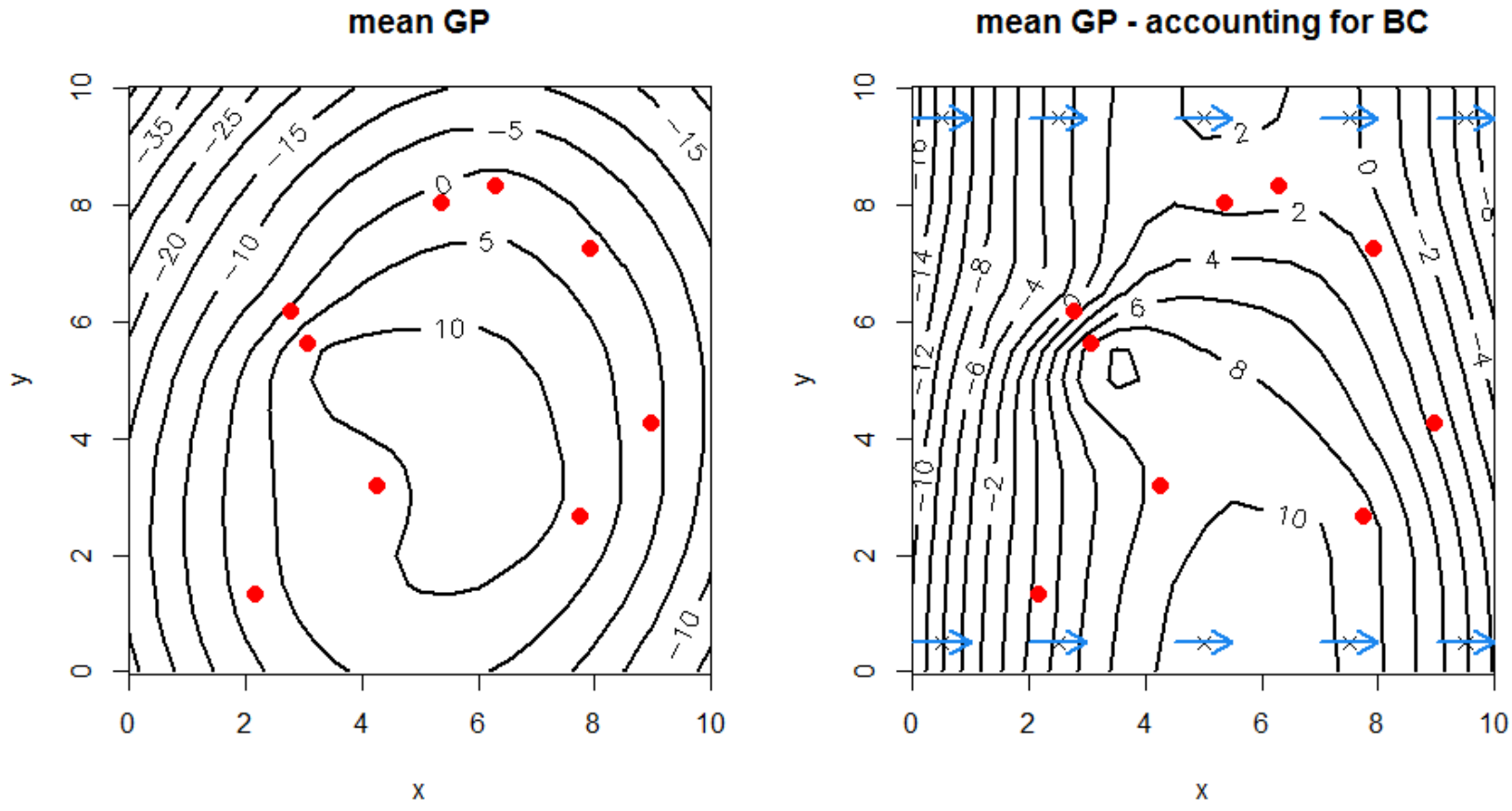
Catchment flow simulation (2)

Package `airGR`, see companion website



Groundwater head interpolation

package `GauProMod`: Gaussian process (Kriging) with constraints on boundaries



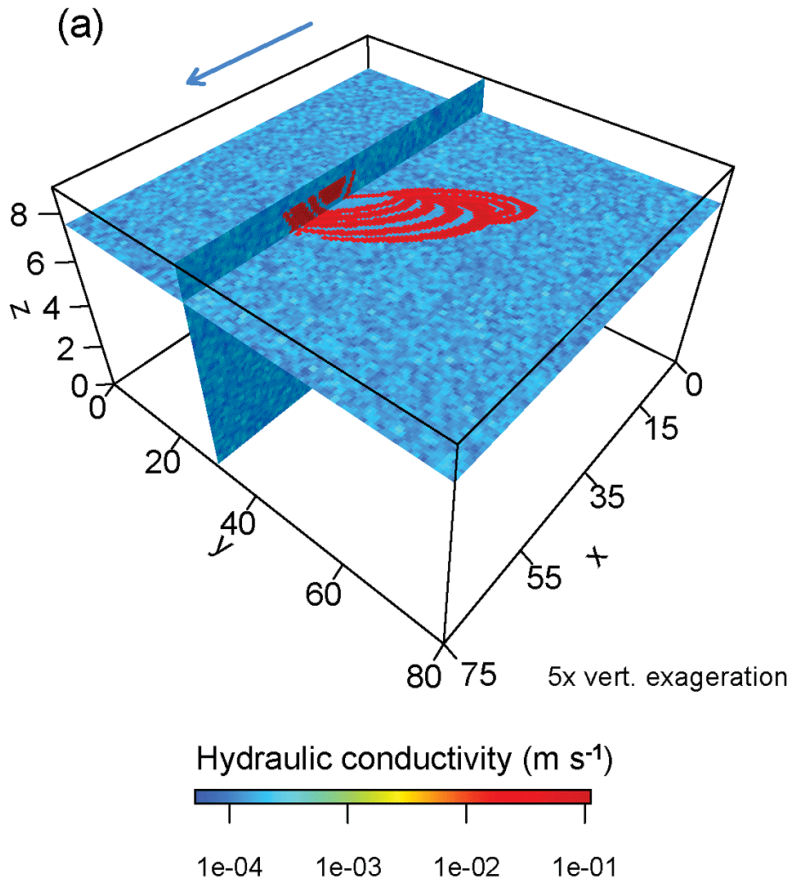
After Kuhlman and Igusquiza (2010)

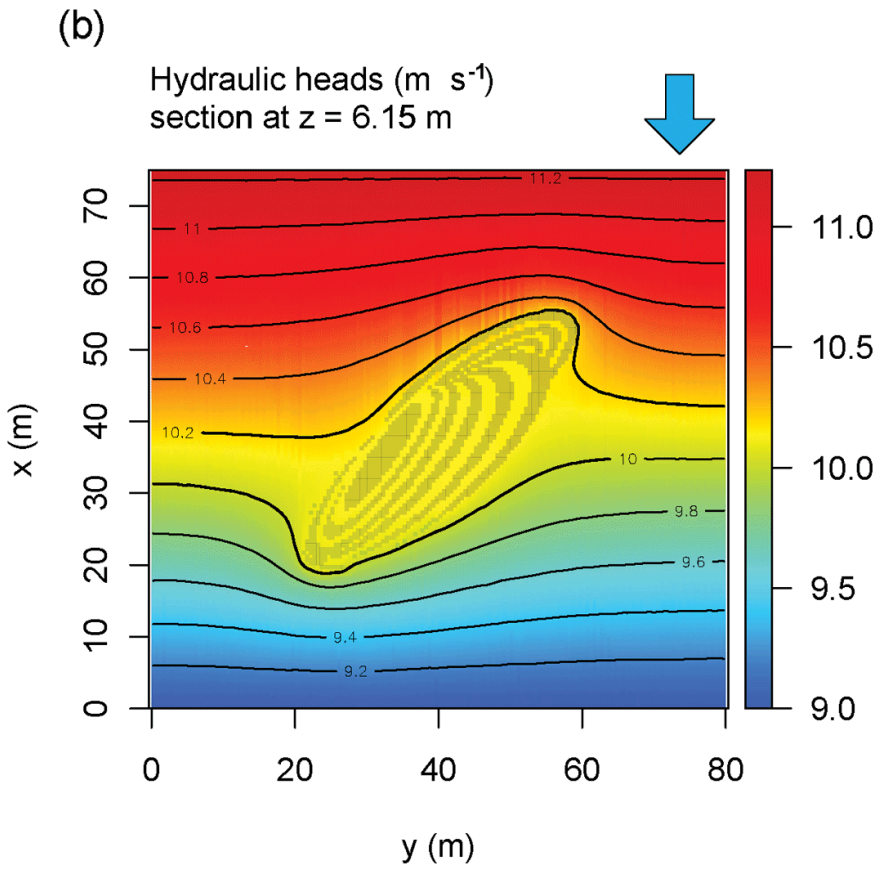
Modflow - US-GS reproducible report

github.com/USGS-R/wrv

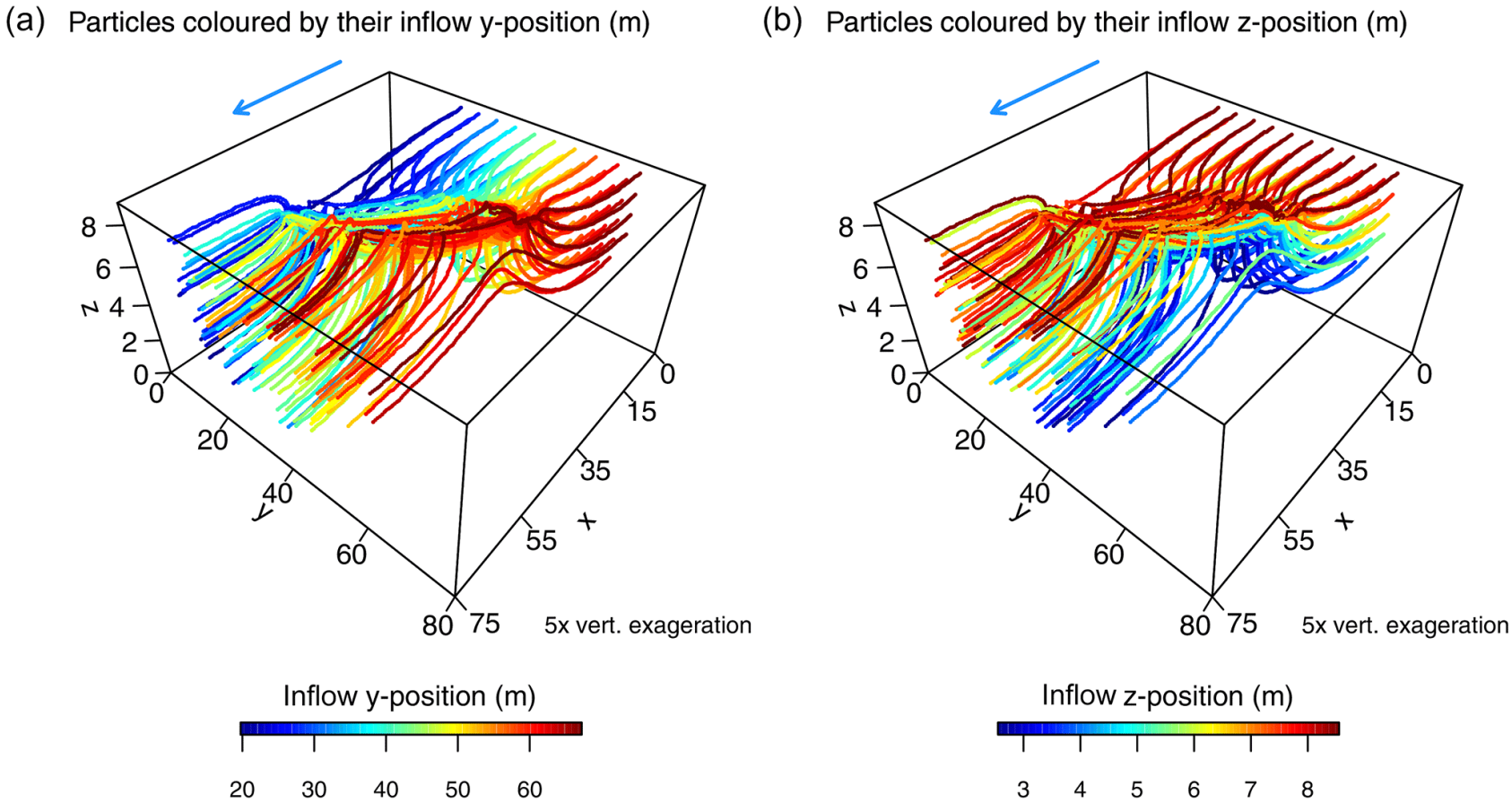
Modflow - subsurface flow mixing (1)

personal code based github.com/USGS-R/wrv

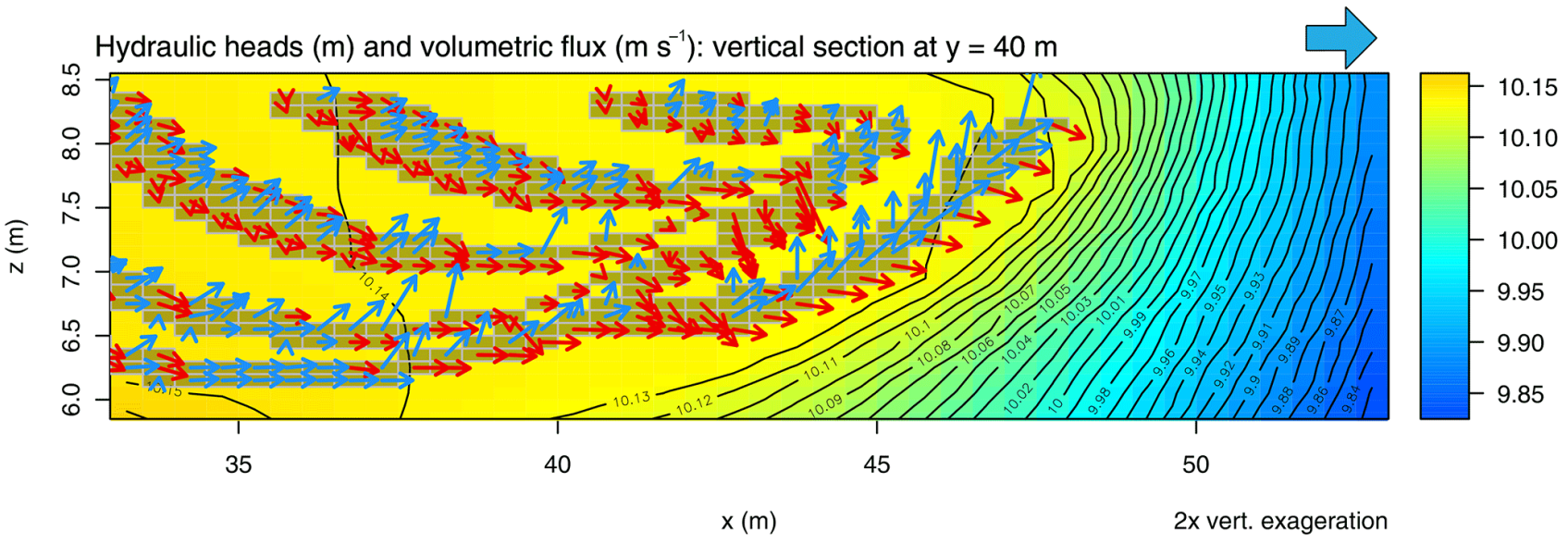




Modflow - subsurface flow mixing (2)



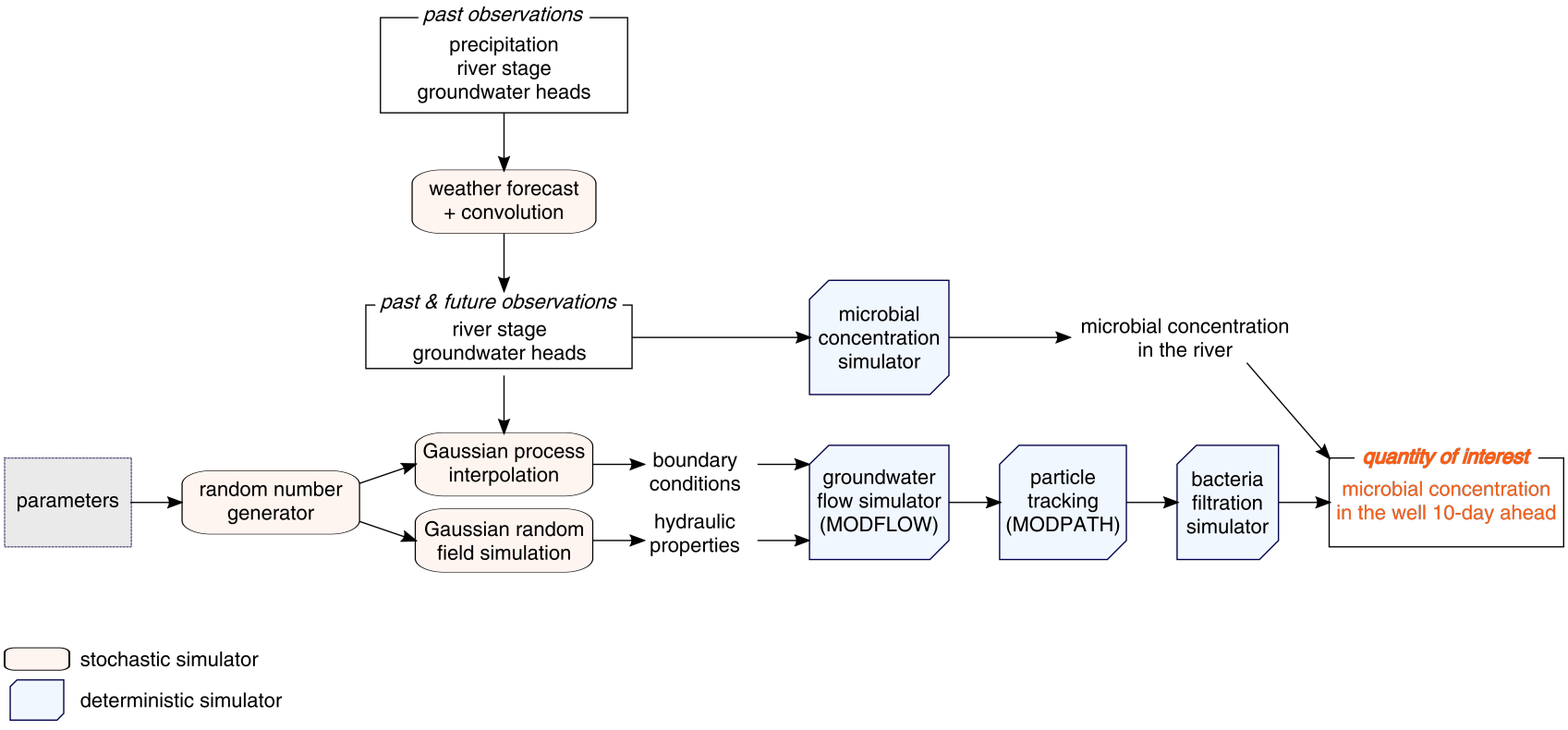
Modflow - subsurface flow mixing (3)



Modflow -stochastic simulation (1)

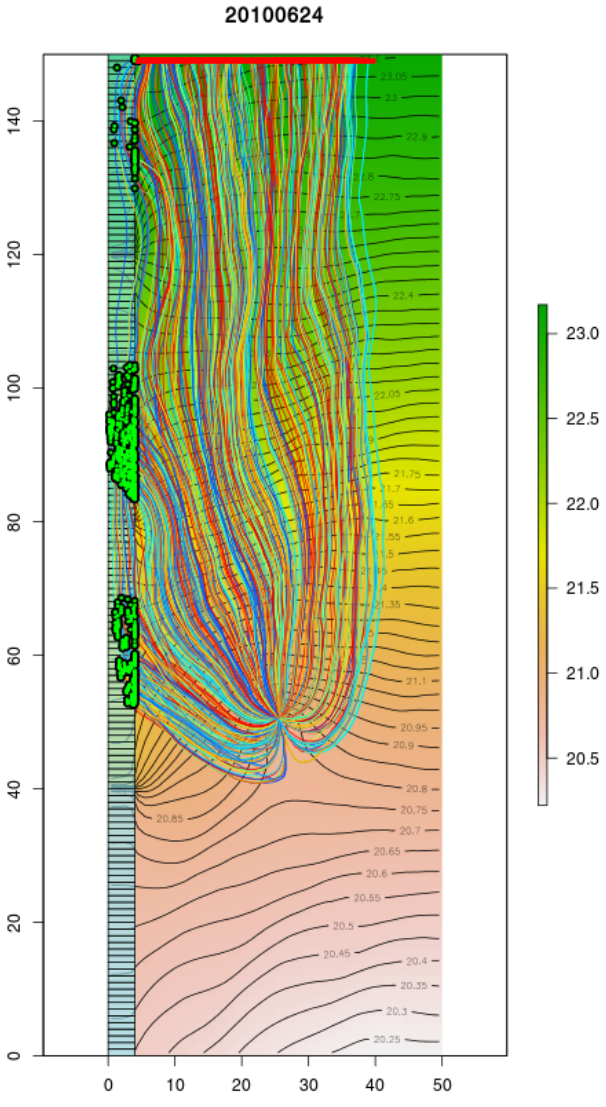
(`gwModBac` personal code based on github.com/USGS-R/wrv Groundwater flow simulation and particle tracking to forecast microbial concentration in a drinking water extraction well.

Workflow

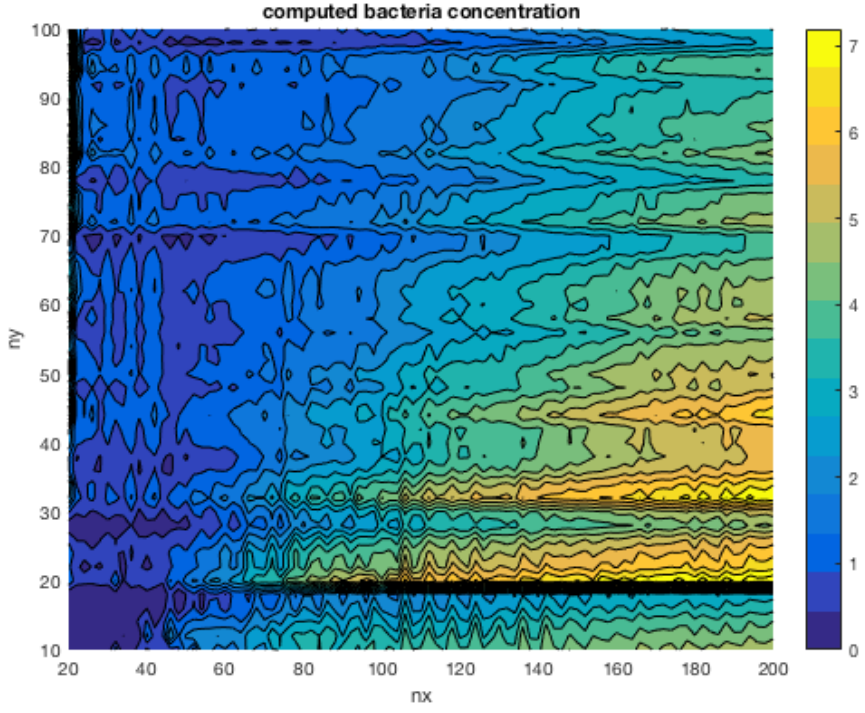


Modflow -stochastic simulation (2)

Bacteria pathways



Bacteria concentration as a function of grid size



physically based lumped model

simulate groundwater fluctuations in response to precipitation and pumping: package ambhasGW Sekhar et al. (2017)

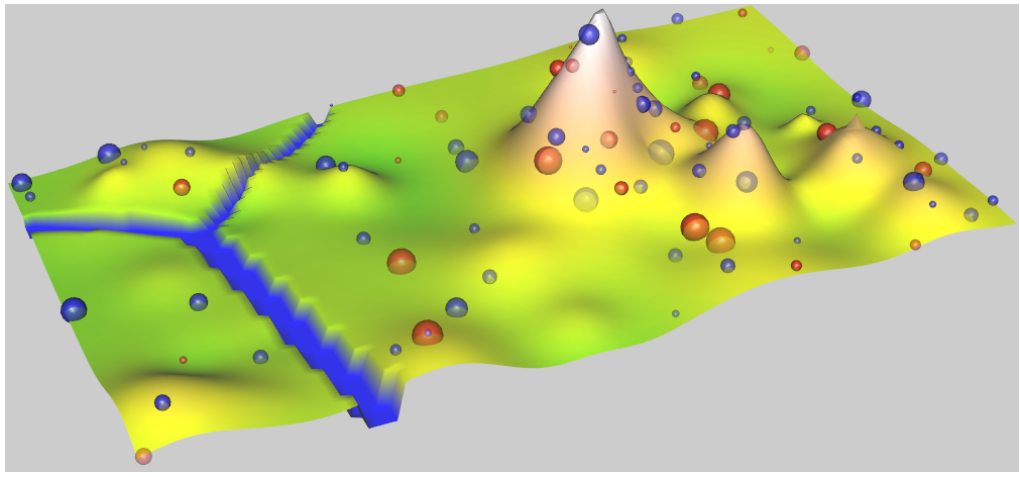
Reporting

publication-quality figures (see also package ggplot2)

Either:

- ggplot and ggplot2 and theirs cousins
- or base graphics, plot3D
- 3D interactive plot with rgl (based on OpenGL)

```
library(rgl)
demo(abundance)
```



- Report/presentation/book: HTML/PDF (packages RMarkdown and knitr)
- interactive web apps (package shiny)
- R package (code and/or data)
- R can create interactive teaching modules: You can do it in the console with swirl or on the web with Datamind.

Check

check

In short, pair-wise x-y plots were computed between the average baselines of all FCM and sensor parameters for a first visual observation of any apparent correlations. The pair-wise correlations between the full base- lines of all parameters were then quantified by the computation of Pearson's correlation coefficients (PCC, linear relationship) and Spearman's rank correlation coefficients (monotonic rela- tionship) after standardization of the FCM and sensor data sets (see Section 8 in supplementary information). The significance of the correlations was assessed by the computation of p-values at 95% confidence level. The pair-wise coefficients were displayed in a heat map for efficient representation of the gradients in pos- itive and inverse correlations between parameters, and for rapid identification of the pre-dominant correlations. In this heat map, the parameters were reordered by hierarchical clustering using the Ward algorithm (see Section 8 in supplementary information). The additional R packages Vegan (Oksanen et al., 2009), Heatplus (Ploner, 2011), and Heatmap.plus (Day, 2007) were used to these ends. TP1 Basellandschaft

Online flow cytometry measurements were linearly interpolated and then sampled at equal time intervals (15-minutes) using the "approx()" command of the statistical software R39. This was to adjust for minor deviations from the 15-minutes sampling interval in the original data set. . .

. . . The interpolated time series was then decomposed into a "trend component" (aperiodic dynamic), a "seasonality component" (periodic dynamic; R-code based terminology not related to seasons of the year), and a

Great tool for coding, reporting and code maintaining

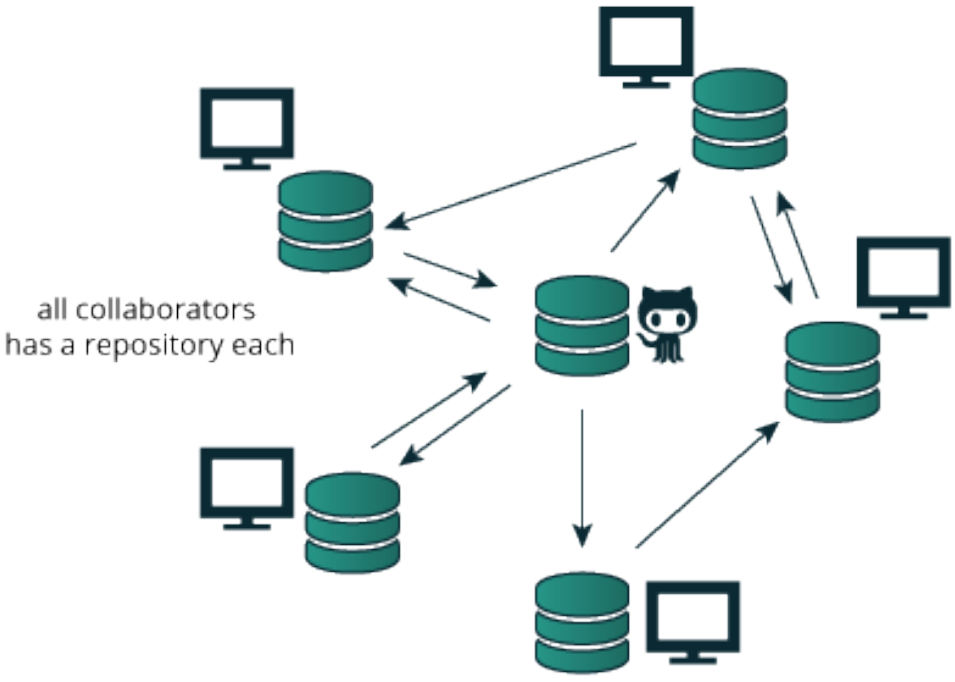
RStudio

- package creation and development
- reporting with Rmarkdown

git & GitHub

git = a free and open source distributed version control system

github = a web-based hosting service for version control using git



Open source means everyone can see my stupid mistakes (Karl Broman)

Version control means everyone can see every stupid mistake I've ever made (Karl Broman)

Reproducible research (1)

Karl – this is very interesting , however you used an old version of the data (n=143 rather than n=226).
I'm really sorry you did all that work on the incomplete dataset.
Bruce

from Karl Broman

Reproducible research (2)

from Karl Broman

0. Separate the raw data from everything (and don't modify them)
1. Organize your data & code
2. Everything with a script
3. Automate the process (GNU Make)
4. Turn scripts into reproducible reports
5. Turn repeated code into functions
6. Create a package/module
7. Use version control (git/GitHub), no more "really_true_final_2EH5b.doc"
8. Pick a license, any license