The challenge of porting scientific results to operational applications

Rebekka Posselt*, Rebecca Hiller, Mark A. Liniger

*rebekka.posselt@meteoswiss.ch
The challenge of porting scientific results to operational applications

Rebekka Posselt*, Rebecca Hiller, Mark A. Liniger
Thanks to work by: Christoph Frei, Sophie Fukutome

*rebekka.posselt@meteoswiss.ch
In the beginning

The task:

• Analyze the past climate
• Monitor the current climate
• Predict the future climate

Therefore, climate researchers

• Develop new methods
• Create new visualizations
to deliver climate products
• Understandable
• Usable

by the climate information customer.
And then?

• «Leave it on the shelf to gather dust»?
• Port it to operational applications
Report generation

- Direct, automatic output
- Uses Sweave (\LaTeX) with embedded R-Code

- Implementation of the official layout (LaTeX class)
- Statistical analysis via embedded R-Code
- Customizable (heading, text, contact info, ...)
- Automatic formatting (header, footer, ...)
Gridded Datasets

Monthly Mean Temperature (degC) Jan 2015

External
(Web)

Internal

Daily Mean Temperature (degC) 2015–10–19

http://www.meteoschweiz.admin.ch/home/klima/gegenwart/monats- und-jahreskarten.html

- User-dependent visualizations
  - Less information for Internet → Common user
  - Additional information for internal use (e.g., used for quality control) → Expert user
Climate Indices: Ongoing work

- Need for unification!
- Several tools @ MeteoSwiss used to calculate climate indices for station data, gridded data, climate scenarios

CRAN-package “climdex.pcic”: contribution and collaboration
And then?

- «Leave it on the shelf to gather dust»
- Port it to operational applications

???
Requirements on operational tools

- Reliable
- Reproducible
- Maintainable
- Automated
- User friendly
The solution @ MeteoSwiss

CATs → Climate Analysis Tools

CATs → collection of «R»-packages

= open-source statistical software

CATs provide the framework for

• Coordinated development
• Automatic and/or individual production
• Easy maintenance

of a wide variety of climate products.
Common structure

Programm call (→ argument names and formatting)
• Easy to use, understood one → understood all

CAT/Package structure (→ Input preparation, Data retrieval, Data analysis, Output)
• Enables coordinated and easy development
• Easy maintenance

General functionalities in CAT. Helper Libraries
• Basic functionality (data retrieval, color tables, labeling)
• Advanced, more scientific functionality (climate indices, Verification Skill scores)
Common Structure

- CATs
- CAT.Helper & External Libraries
- R-Base

DWH

Figures

Data

Tables

R-Base (http://www.r-project.org/)
Documentation

pheno.longts {phenopoly}

Description

Create a plot of long phenological records

Usage

pheno.longts(
  parameter = "bjaesc09",
  station = "FGE",
  datafile = "dwh",
  bgimage = "aesculus",
  start.year = 1800,
  end.year = "current",
  ylim = c(-6, 120),
  title = NULL,
  add.years=TRUE,
  filter.col = "red",
  filter.max.gap=5,
  line.col = "darkblue",
  outpath="current",
  languages = "G"
)

Arguments

character string parameter name as in dwh for text file data input, the parameter will be used in the output datafile
character string station name abbr as in dwh for text file data input, the station will be used in the output datafile
character string "dwh" to derive data from dwh or the path and datafile of the input file the file must contain two columns separated by white spaces or tabs the columns do not have a header the first column defines the observation year in the format YYYYmmddMMHHSS the second column the observation date in the format YYYY
character string datafile of the background image NULL to omit the background image "prunus" and "aesculus" for default pictures of the two plants provided as part of the package
numeric start year of the analysis YYYY e.g. start.year=2012
numeric end year of the analysis YYYY e.g. end.year=2012 or end.year="current" for the current year, or end.year="last" for the year preceding the current year
numeric vector of length 2, min and max value on the y-axis, xlim=NULL for autoscaling

• Comprehensive documentation required
• Meaningful examples and Tests necessary
• Recommended packages:
  • «roxygen» for Documentation
  • «testthat» for Testing
## GIT – A version control system

- **Change tracking:**
  - Who changed What, Where, and When
  - Supports collaboration between developers

- **Code archive**

- **Trigger for automatic installation on the servers**
Automated production & monitoring

- Job scheduling via Linux-intern «crontab»
  - Manage the automatic production for Internet, Intranet, Archive, …
  - ~20 CATs run as «cronjobs» at different times (daily, monthly, seasonal, yearly, special, …)

- Monitoring by capturing «errors»
  - Within the «cronjob» scripts
    → Trigger of Fail-Emails
Server architecture

• Development server
  → Reserved for developing CATs
• Interactive server
  → Accessible by all interested employees
  → Individual production
• Production server
  → Reserved for automated production
  → Redundant layout (Main + FailOver)

• All servers provide the same environment (OS, libraries)
• Maintenance and administration by IT-Department
Requirements on operational tools

Reliable
- Tested
- Redundant system
- Monitored

Reproducible
- Code archive
- Scriptable
- Documented

Maintainable
- Common structure
- Standardization
- Single implementation

Automated
- Scheduled
- Monitored

User friendly
- Documented
- Customizable
- Easy access