R and Harp packages new data and enhanced tools

C. Zingerle, **A. Deckmyn, A. Singleton,** B. Sass

EWGLAM and SRNWP meeting
Salzburg, 1.-4.10.2018

Harp v1 and v2: recap

HARP: continuous development since 2013:

- Common development of **EPS and spatial verification tools** in Aladin and Hirlam:
  - Based on available R-packages (and shell/python scripts - **H**irlam-**A**ladin **R** & **P**ython tools for verification)
- First version for EPS only (v1, 2015), in 2017 (v2) including Harp spatial tools.
  - Deal with different spatial data-formats (GRIB, FA, netcdf4, hdf5) and station data (SYNOP, local networks)
    - For your own native data-formats (INCA, lightning data, local radar data, …) write a decoder (projection, grid size and distance, …).
  - Excessive number of scores for EPS available
  - Includes fuzzy (FSS, ETS), object oriented (SAL) and point verification scores calculated on the grid
  - Documentation on google docs

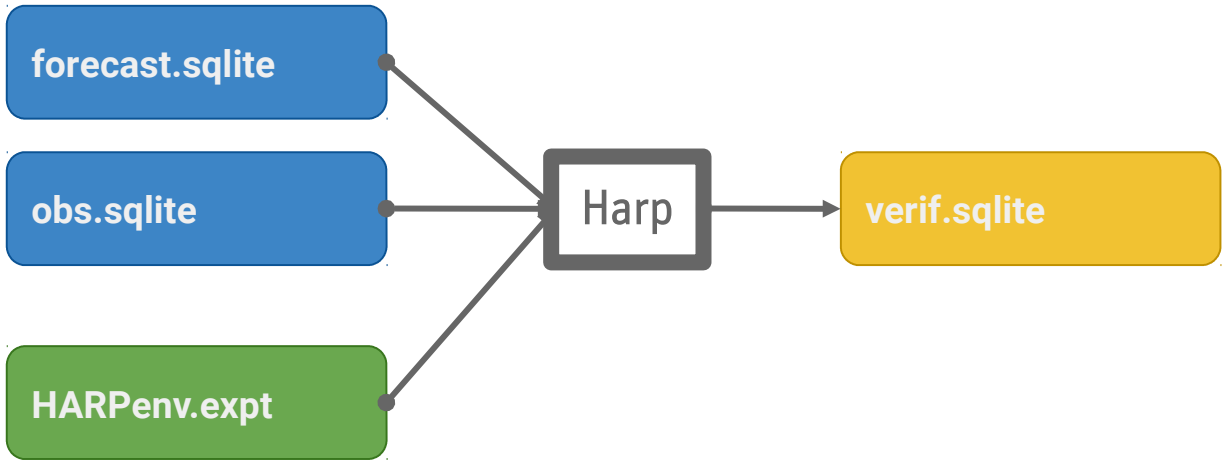
Harp v1 and v2: recap

- Install R
- Install package dependencies
  - install system libraries
- Download and install harp
  - “in-house” R-packages
  - suite of shell scripts
  - Configuration files
- Edit configuration file(s)
- Run script to interpolate forecasts to stations or observation grid
- Run script to convert observations to sqlite or observation to forecast grid
- Run script to compute verification scores
- Visualise results (interactive)

Harp v1 and v2: recap

courtesy of
Andrew Singleton

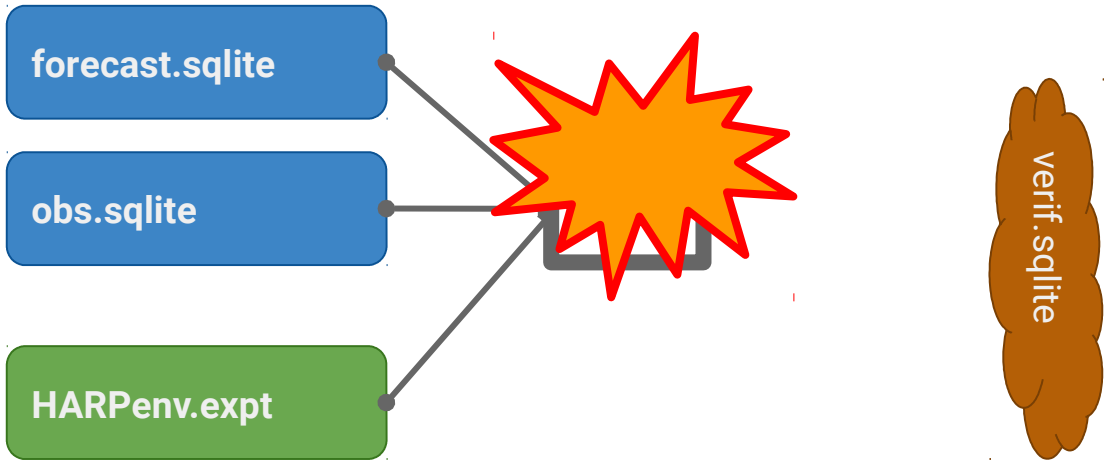
- Install R
- Install package dependencies
  - install system libraries
- Download and install harp
  - “in-house” R-packages
  - suite of shell scripts
  - Configuration files
- Edit configuration file(s)
- Run script to interpolate forecasts to stations or observation grid
- Run script to convert observations to sqlite or observation to forecast grid
- Run script to compute verification scores
- Visualise results (interactive)

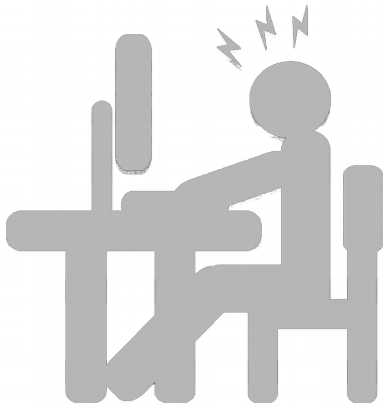


Harp v1 and v2: recap

courtesy of
Andrew Singleton

- Install R
- Install package dependencies
  - install system libraries
- Download and install harp
  - “in-house” R-packages
  - suite of shell scripts
  - Configuration files
- Edit configuration file(s)
- Run script to interpolate forecasts to stations or observation grid
- Run script to convert observations to sqlite or observation to forecast grid
- Run script to compute verification scores
- Visualise results (interactive)

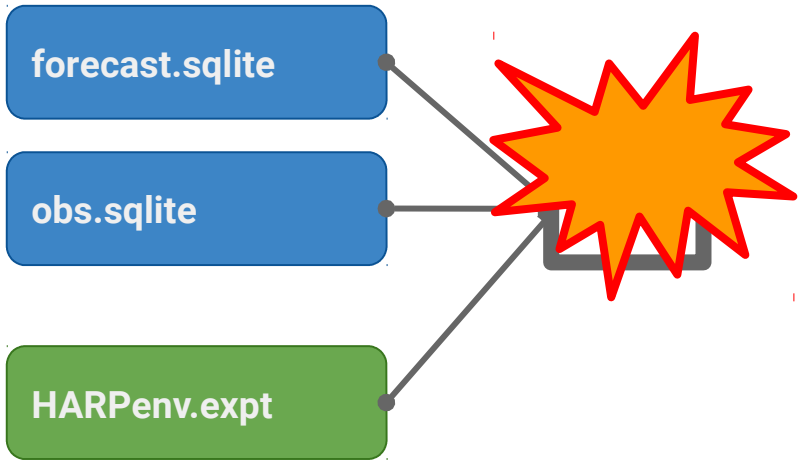


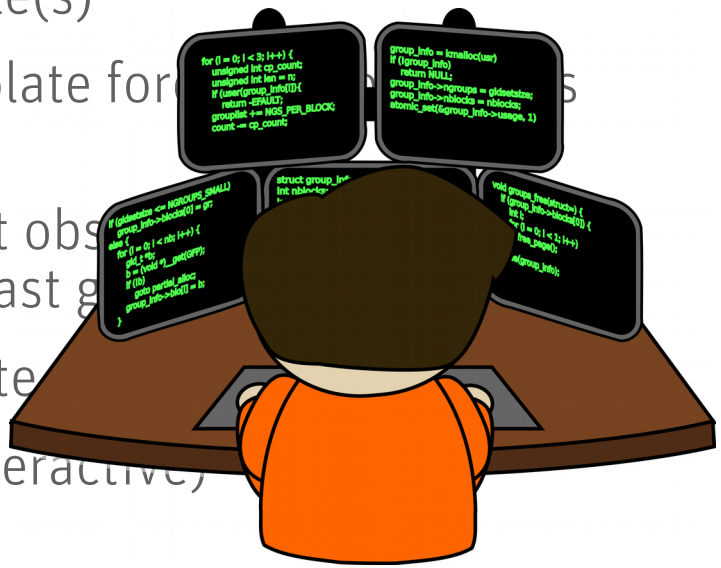


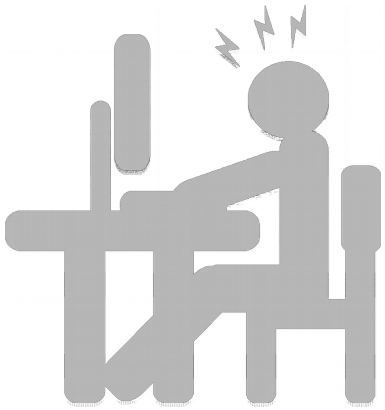
Harp v1 and v2: recap

courtesy of
Andrew Singleton

- Install R
- Install package dependencies
  - install system libraries
- Download and install harp
  - “in-house” R-packages
  - suite of shell scripts
  - Configuration files
- Edit configuration file(s)
- Run script to interpolate for[illegible]s or observation grid
- Run script to convert obs[illegible] observation to forecast g[illegible]
- Run script to compute[illegible]
- Visualise results (interactive)







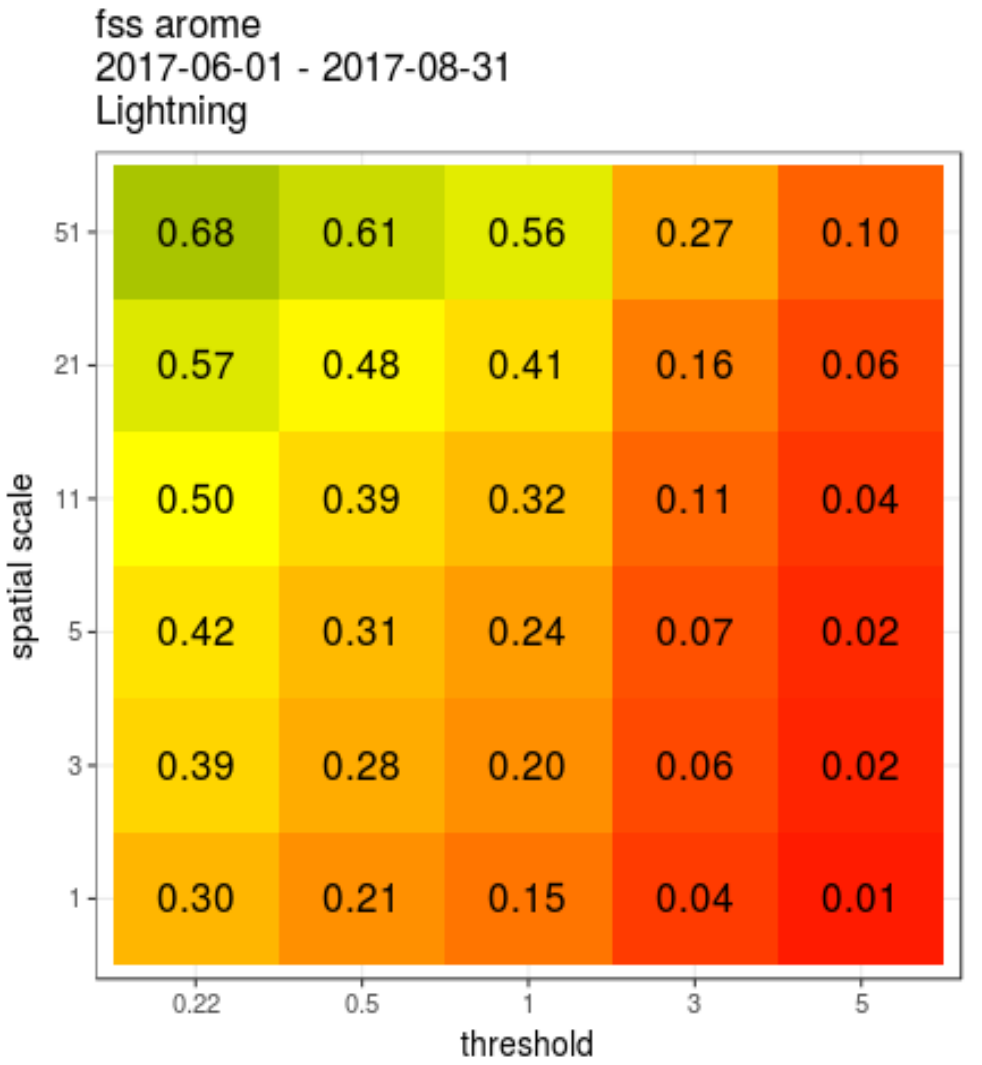
Harp v1 and v2: recap

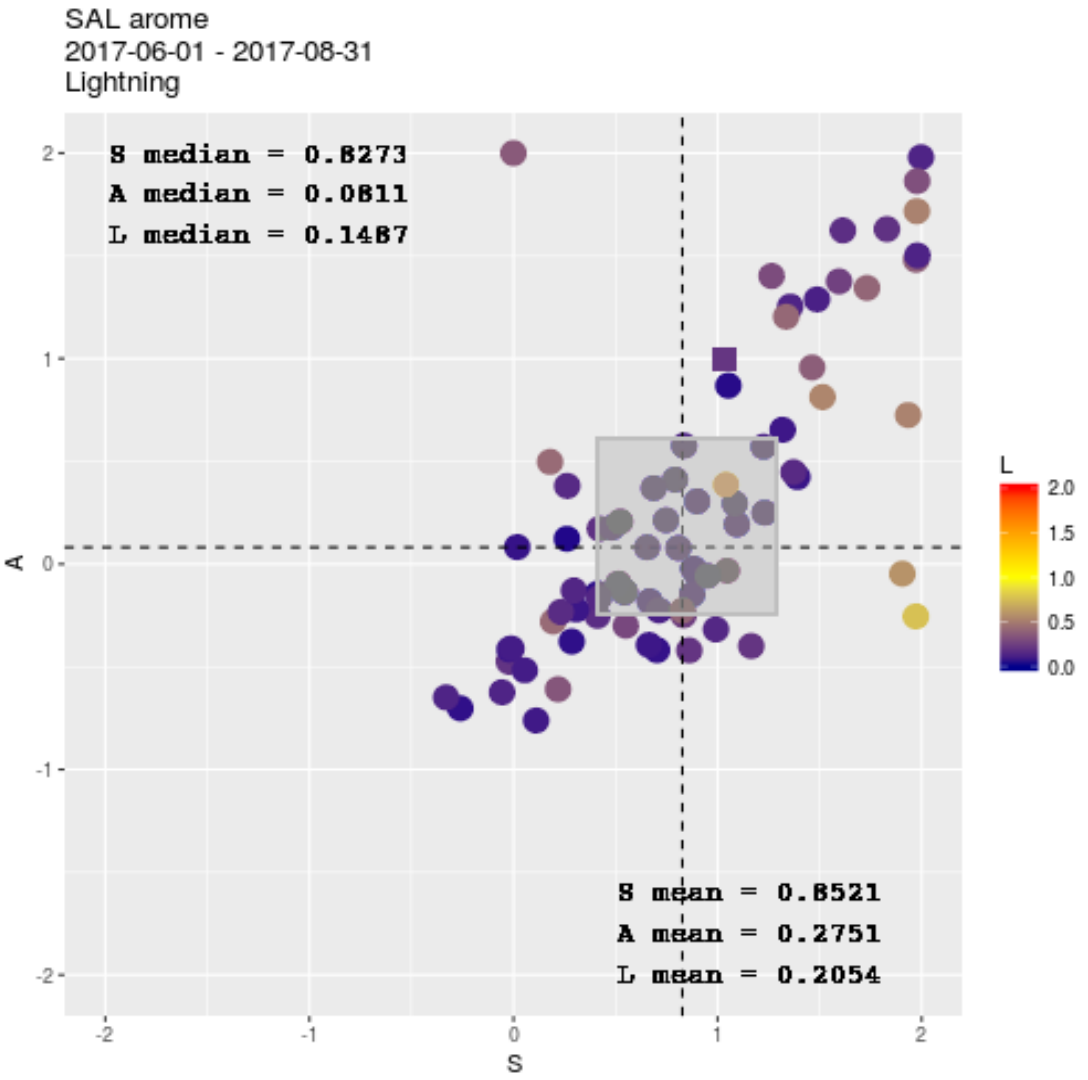
First official versions of the Harp tools

- Fully functional verification routines as R-scripts able to deal with main (spatial) data-formats and and some local formats
- Shiny (R) app for visualization included
- A lot of interaction between users and developers needed (local system, configuration, data, shell and R scripts, ...)
- Usefulness of spatial scores is (mainly) a question of:
  - Quality and availability of spatial observation data
  - Application and selection of spatial method

Harp v1 and v2: recap

Verification of lightning density forecast (summer, afternoon – night thunderstorms)





towards harp 3.0

Enhance usability

- Code organization as R-packages >>> **harpIO / harpPoint / harpSpatial / harpVis**
  - Stricter rules of use of funcitons
  - No interaction of users at code level
  - Enhanced portability
  - Enhanced package documentation necessary

towards harp 3.0

Enhance usability

- Executing harp interactively
  - Follow the harp workflow
  - Visualize forecasts and observations
  - In line documentation of your verification work
  - Conditional verification

towards harp 3.0

Enhance usability

- READERS
  - Extended documentation on readers
    - How to get observation and forecast data into harp
    - Example data
- Examples of setup and configuration files >>> vignettes
  - Extend harp online documentation
- Intensify research on available spatial observation data (common RWP)
  - Satellite radiance/BT, aircraft data, any grided analysis, …

towards harp 3.0

Enhance usability

- Docker (containers)
  - Run harp in a container on any system
  - Allows high grade of portability of harp setups in different environments
    - Develop setup for your operational environment
    - Static set of R, R-libraries, compilers, system libraries ...

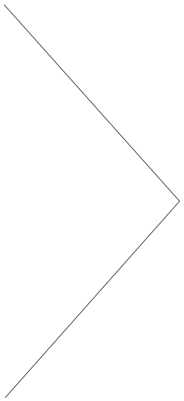
harp 3.0

- From R-scripts to **R-package**(s)
- Total rewrite of EPS and big part rewrite of spatial verification
    - Interactive in R (Rstudio, markdown)
    - Tidy verification data (in the spirit of tidyverse)
    - Inclusion of deterministic point verification
- New structure
- New visualisation functions
- New IO functions

harp 3.0

R packages for harp:

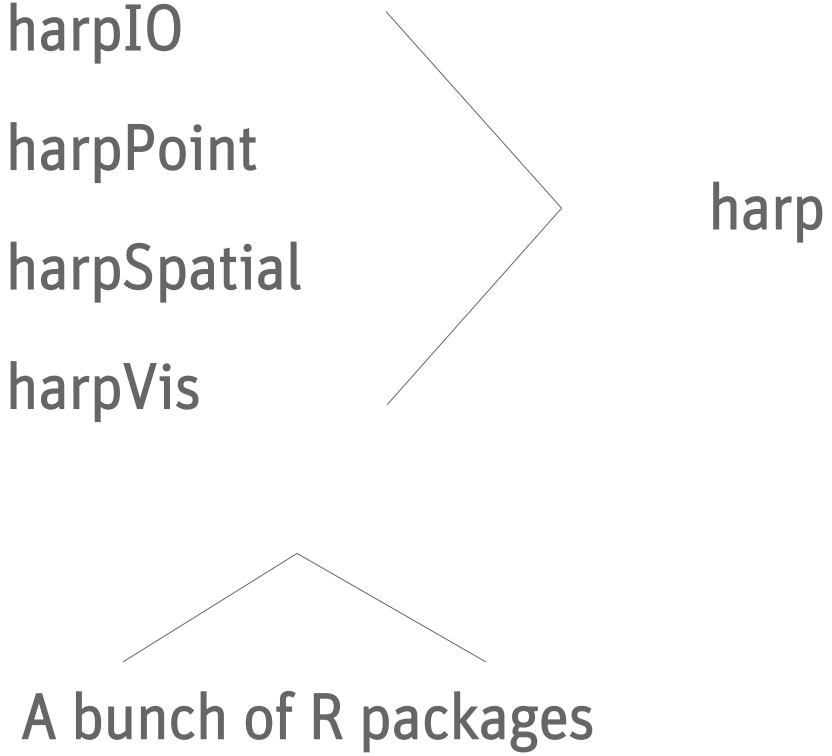
- harpIO
- harpPoint
- harpSpatial
- harpVis



harp

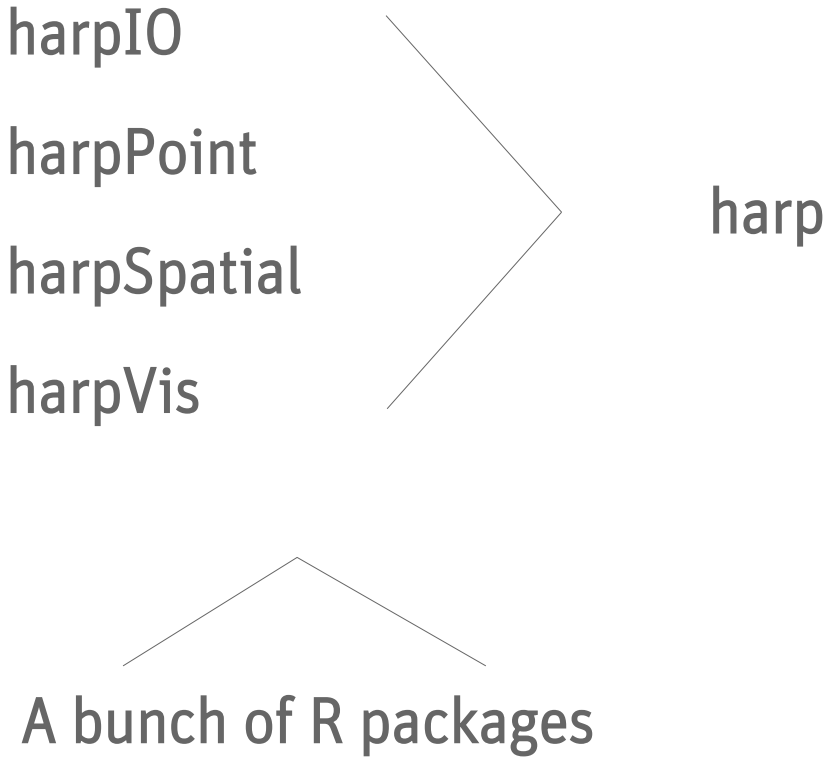
harp 3.0

R packages for harp:



harp 3.0

R packages for harp:



- Independence of platform
- Documentation inside R
- Interactive use of R packages within Rstudio, markdown, ...

harp 3.0

harpIO

Functions to **read** (& interpolate) meteorological data
- grib
- netcdf
- FA
- hdf5
- vfld / vobs
- Expandable to more formats

Functions to **write** station & verification data
- sqlite
- rds

harp 3.0

harpPoint

Functions for point verification
- EPS scores
- Deterministic scores
- (Score cards)
- (Statistical tests)

Functions for plotting point verification scores
- Universal plot function
- Interactive shiny app

harp 3.0

harpSpatial

Functions for spatial verification
- FFS
- SAL
- (MODE)
- more...?

Functions for plotting spatial verification scores
- Universal plot function (?)
- Interactive shiny app

harp 3.0

harpVis Functions for plotting meteorological data

- Maps
- "Meteograms"
- Experimental probabilistic visualistions
- Profiles (Skew T - log P)
- Cross sections
- Interactive shiny app(s)

harp 3.0

harpVis Functions for plotting meteorological data

- Maps
- "Meteograms"
- Experimental probabilistic visualistions
- Profiles (Skew T - log P)
- Cross sections
- Interactive shiny app(s)

Available on Github?

harp 3.0

Documentation

- Each harp package will be accompanied with at least one vignette giving a worked example
- Every function will include inline documentation with workable examples
- Plan for blog giving examples of different tasks that could be done with harp
- Rmarkdown skeletons for workflows
- Snippets for Rstudio
- Plan for open source ...

harp 3.0

New flexibility in harp

Interactivity -> ownership of the verification process

Conditional verification

Analysis of observation errors - e.g. perturb observations

Confidence intervals

Read and write many data formats

Non verification related data wrangling and plotting

Growing platform indenpendence

towards harp v3

Enhance usability

- Code organization as R-packages >>> **harpIO / harpPoint / harpSpatial / harpVis**
  - Stricter rules of use of funcitons
  - No interaction of users at code level
  - Enhanced portability
  - Enhanced package documentation necessary

towards harp v3

Enhance usability

- Executing harp interactively
  - Follow the harp workflow
  - Visualize forecasts and observations
  - In line documentation of your verification work
  - Conditional verification

towards harp v3

Enhance usability

- Docker (containers)
  - Run harp in a container on any system
  - Allows high grade of portability of harp setups in different environments
    - Develop setup for your operational environment
    - Static set of R, R-libraries, compilers, system libraries ...

towards harp v3

Enhance usability

- Decoders >>> READERS
  - Extended documentation on readers
    - How to get information about your data into R
    - Example data
- Examples of setup and configuration files >>> vignettes
  - Extend harp online documentation

install.packages("harp")

C. Zingerle, **A. Deckmyn, A. Singleton,** B. Sass

41 EWGLAM and 26 SRNWP meeting

Thanks!