harp verification toolbox

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

EWGLAM / SRNWP
30.9 – 3.10.2019
Sofia

5 years of Harp (v1 and v2)

HARP: continuous development since 2013:

Developing **EPS** and **spatial verification tools** in a common effort (**H**irlam-**A**ladin **R** & **P**ython tools for verification)

- Based on available R-packages
- Shell/Python scripts
- Deal with spatial data-formats (GRIB, FA, netcdf4, hdf5) and station data (SYNOP, local networks)
- Decoder for native data-formats (INCA, lightning data, local radar data, …) (projection, grid size and distance, …).
- Lots of scores for EPS, few spatial methods (FSS, SAL)
- Documentation on google docs

5 years of Harp (v1 and v2)

Quite sufficient tools … BUT

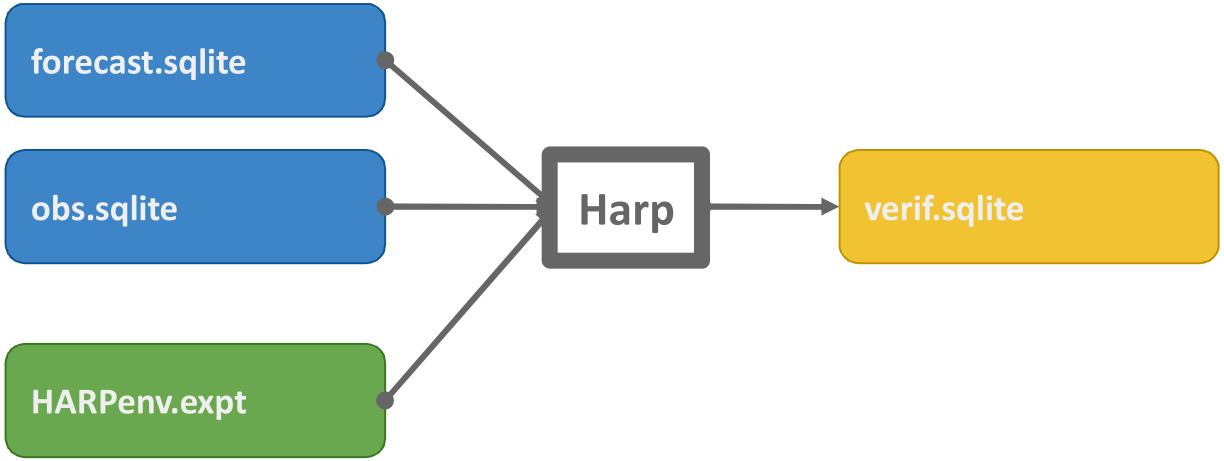
- Install R
- Install package dependencies
  - install system libraries
- Download and install harp
  - “in-house” R-packages
  - suite of shell scripts
  - Configuration files
- Edit configuration file(s)
- Script to interpolate forecasts to stations or observation grid
- Run script to convert observations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

5 years of Harp (v1 and v2)

courtesy of Andrew Singleton

Quite mighty tools … BUT

- 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)
- 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
- Visualize results



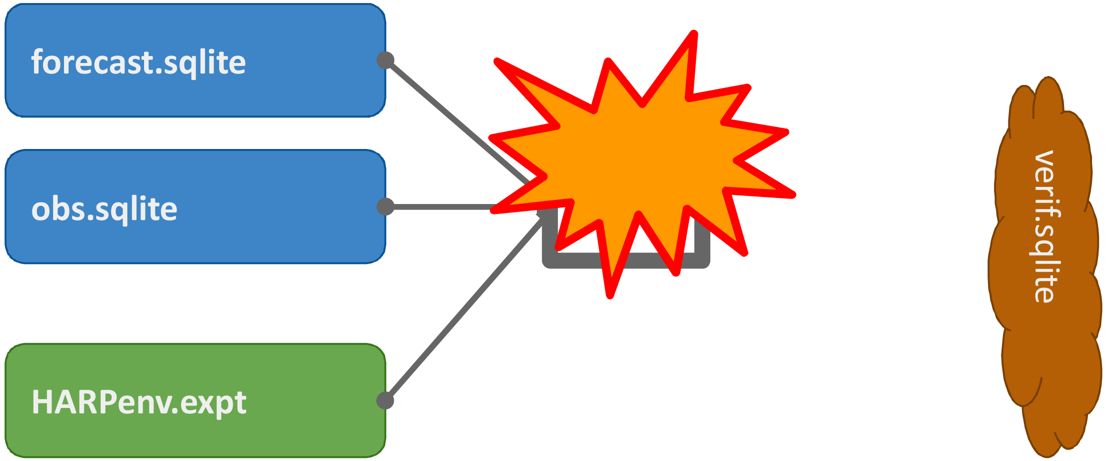
Harp v1 and v2: recap

courtesy of Andrew Singleton

BUT …

When Harp fails:

- Installation
- Scripts
- Platform
- Environment
- …



Frequent interaction between user and developers

towards harp (v3)

Enhance usability … for developers

- From R-scripts **to R-packages** >>> **harpIO / harpPoint / harpSpatial / harpVis**
  - Stricter rules of use of functions
  - Reduce interaction of users at code level (packages)
  - Portability between platforms
  - Documentation at package / function level
  - Tutorials and examples
  - Making packages easily available (GitHub)

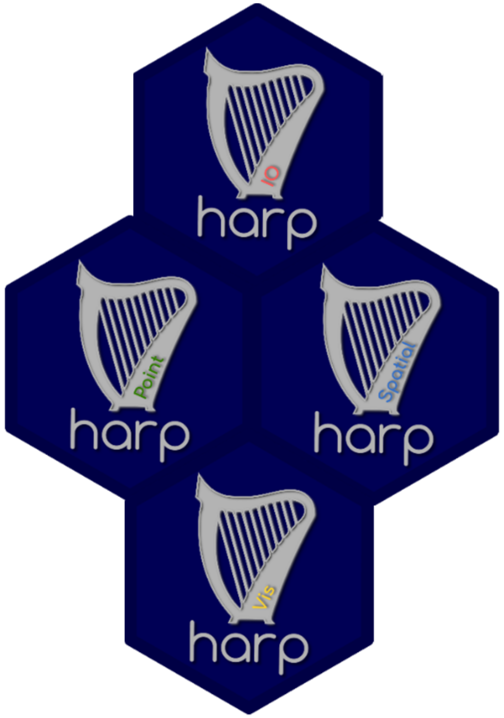
towards harp (v3)

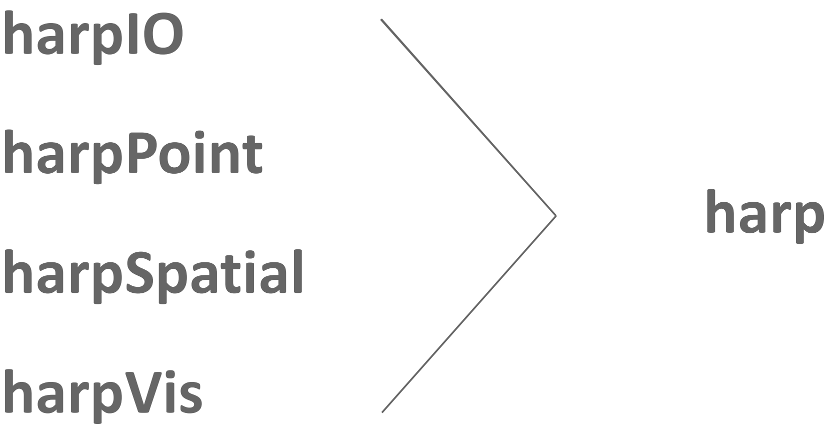
Enhance usability … for users

- Installation handled by R (mainly packages)
  - Get it from GitHub
- No complex scripts …
- Executing harp interactively
  - Follow the harp workflow
  - Visualize verification results AND forecasts and observations
  - In line documentation of your verification work (markdown)

harp

R packages for harp:





harpIO

harpPoint

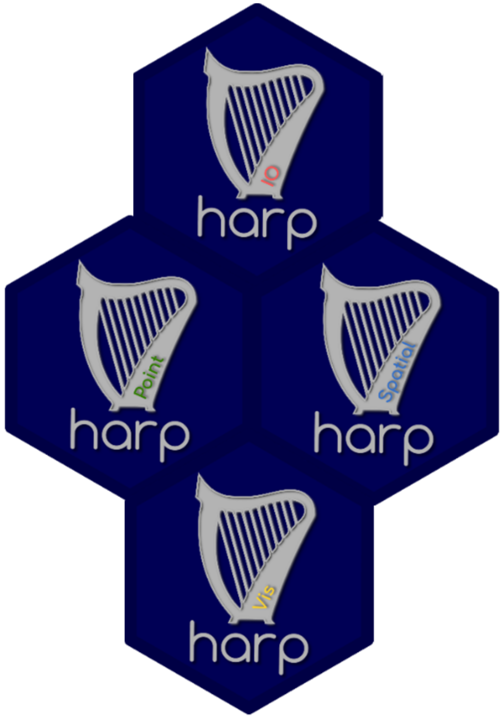
harpSpatial

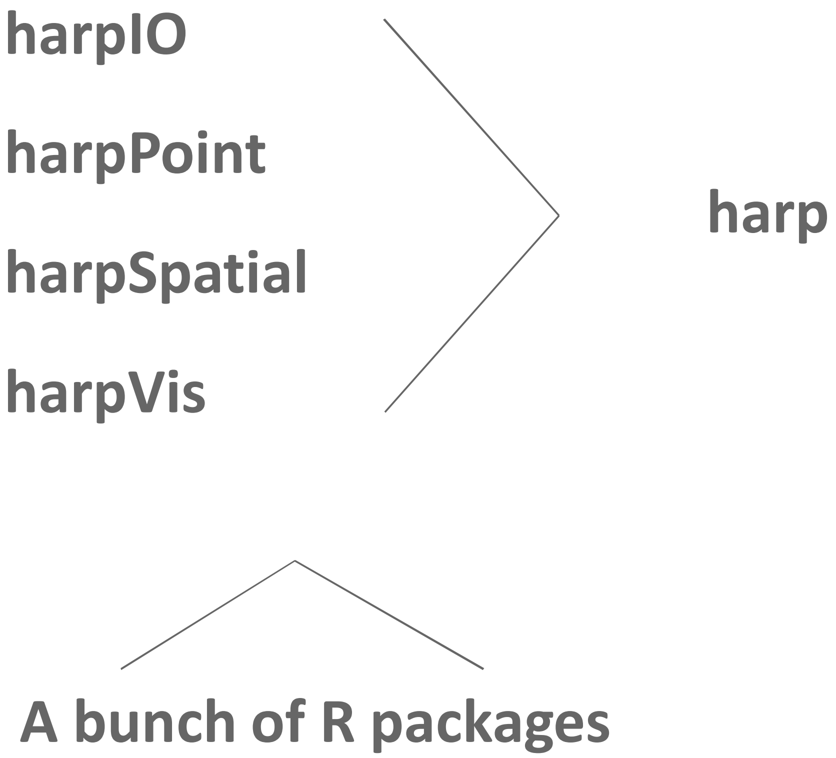
harpVis

harp

harp

R packages for harp:





harpIO

harpPoint

harpSpatial

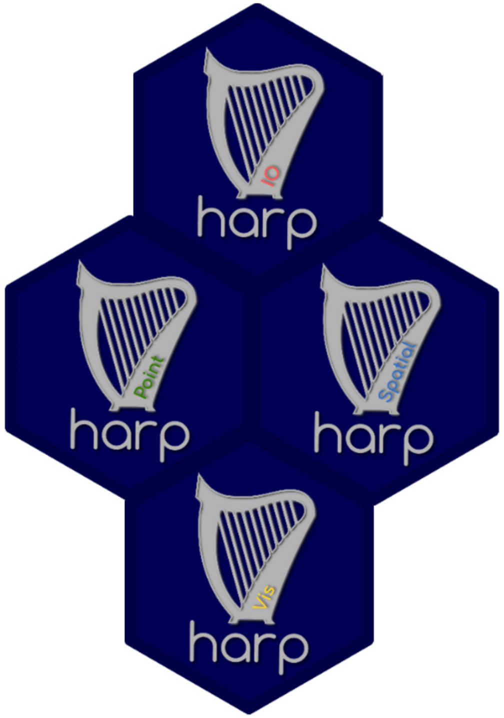
harpVis

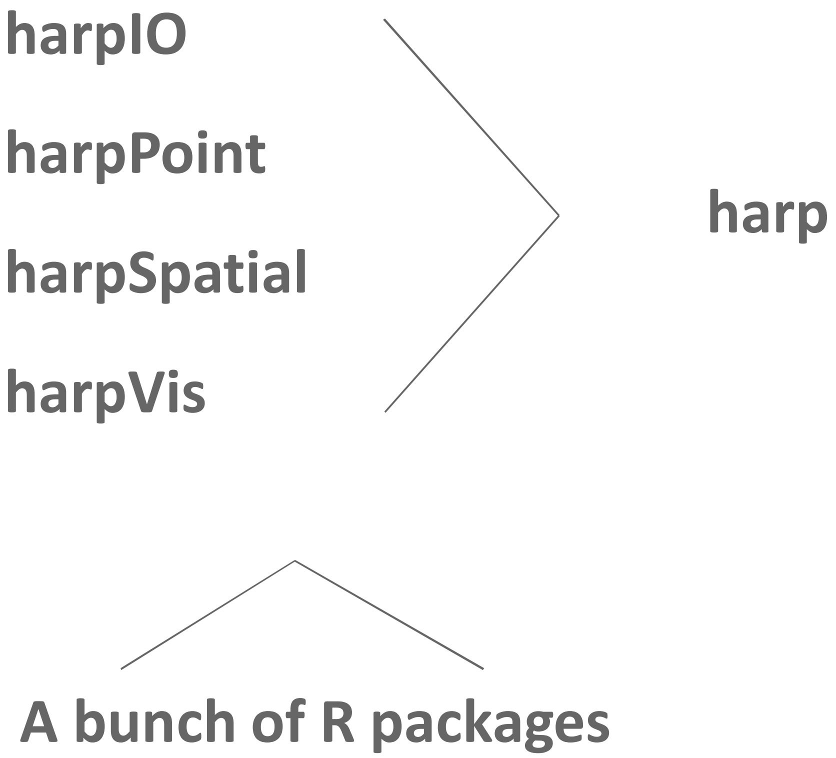
harp

A bunch of R packages

harp

R packages for harp:





harpIO

harpPoint

harpSpatial

harpVis

harp

A bunch of R packages

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

harp

Main task during last months:

- From R-scripts to **R-package**(s)
  - Complete re-organization of
    - EPS (harpPoint) and
    - Spatial verification (harpSpatial)
  - Tidy verification data (tidyverse)
  - Inclusion of deterministic verification (explicit functions)
- Turn IO-scripts into R-package with IO-functions (harpIO)
- Update of visualization (harpIO)
  - Universial plotting functions for scores and observations / forecasts

harpIO

Functions to **read** (& interpolate) meteorological data

- grib
- netcdf
- FA
- vfld / vobs

Functions to **write** station & verification data

- Sqlite database files
- R Data files

```
read_obs_convert(...) ##For point observations
read_point_obs(...) / read_profile_obs(...)
read_eps_interpolate(...) ##For eps forecast data
```

harp

harpPoint

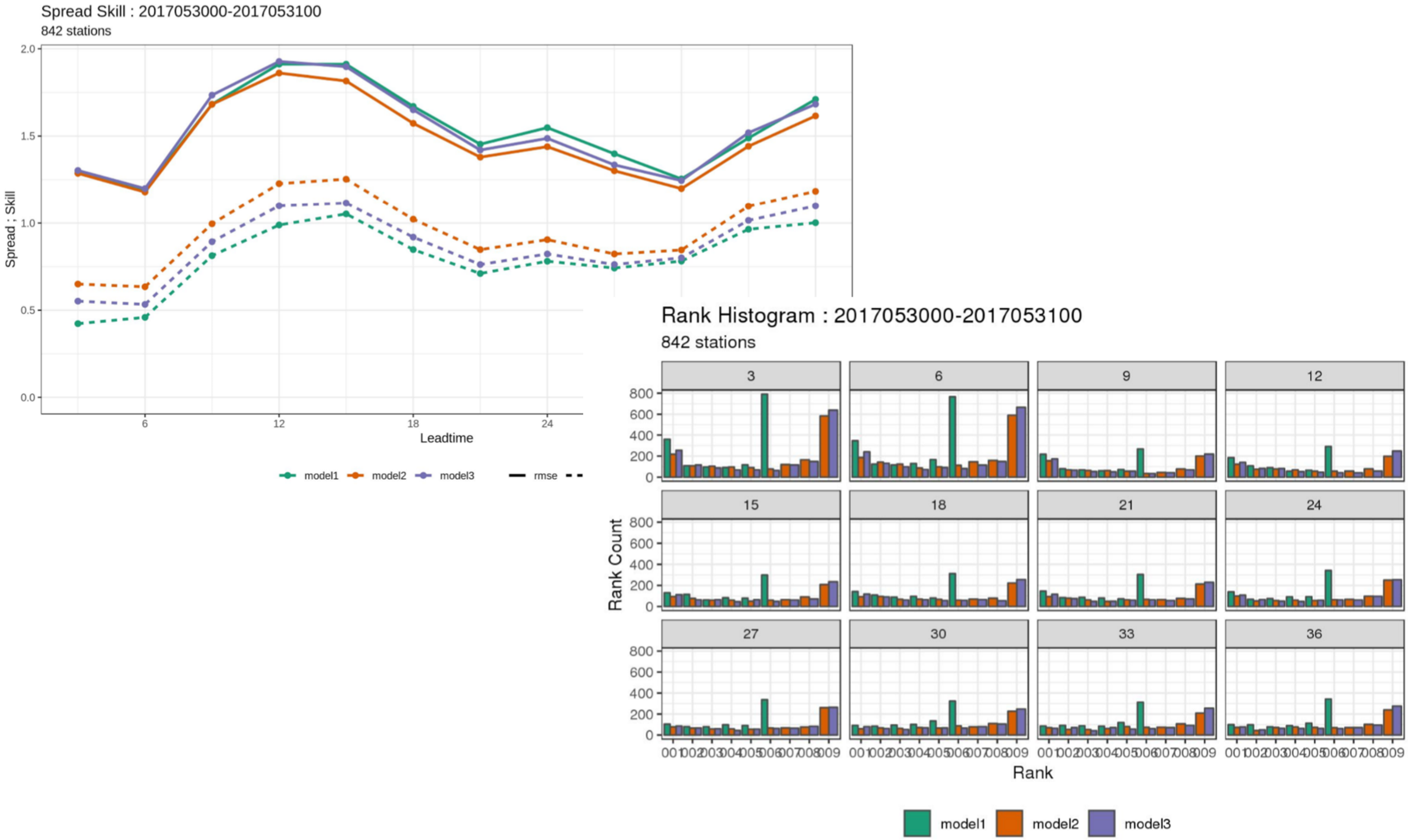
Functions for **point** verification

- EPS scores
- Deterministic scores
- Statistical tests

```
ens_verify(...) ## for ensemble verification
det_verify(...) ## for deterministic verification

ens_verify(..., thresholds = seq(-10, 20, 5))
det_verify(..., groupings = c("leadtime", "fcst_cycle", "SID"))
```

harp



Spread Skill : 2017053000-2017053100
842 stations
Spread ; Skill
2.0
1.5
1.0
0.5
0.0
6
12
18
24
Leadtime
model1
model2
model3
rmse
Rank Histogram : 2017053000-2017053100
842 stations
3
6
9
12
15
18
21
24
27
30
33
36
Rank Count
800
600
400
200
0
001 002 003 004 005 006 007 008 009
Rank
model1
model2
model3

harp

harpSpatial

Functions for **spatial** verification

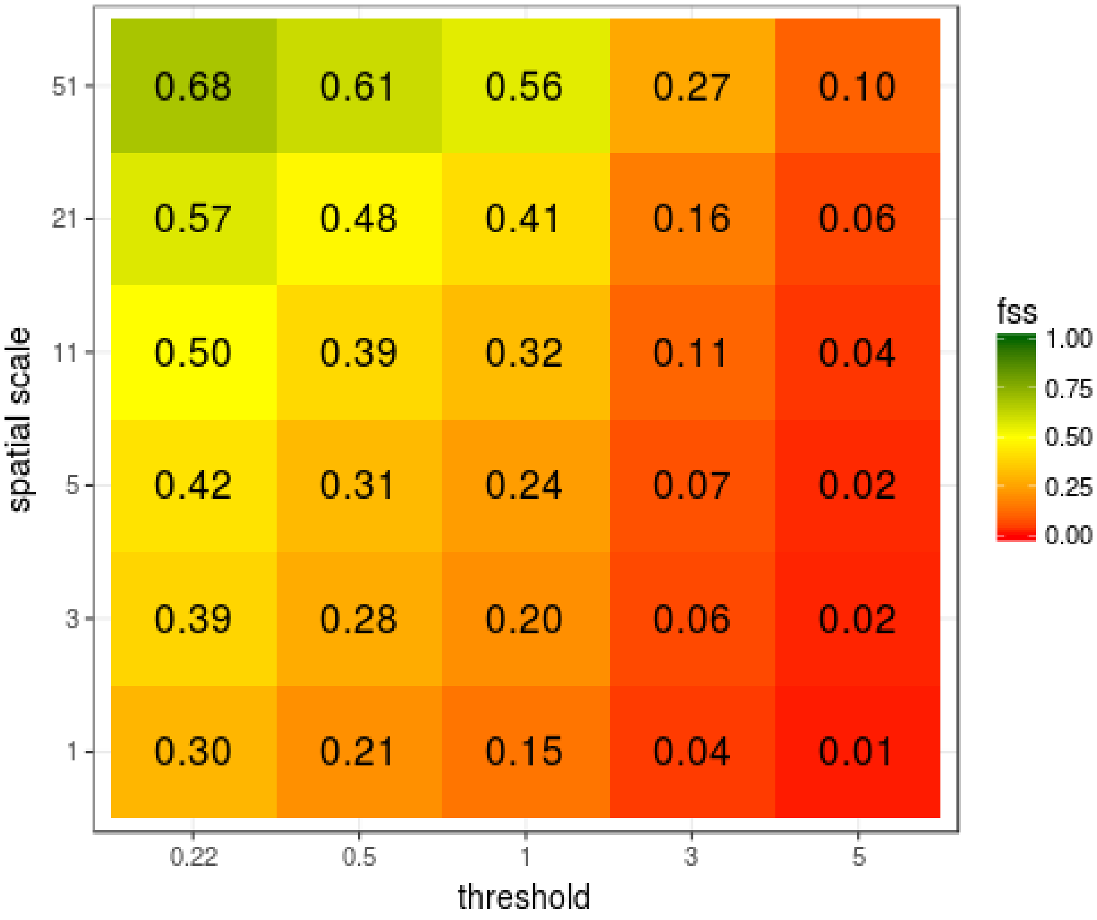
- FFS
- SAL

```
spat_verify(...) ## for spatial verification

spat_verify(..., FSS, thresholds = c(0,1,5,10,20,50,100))
```

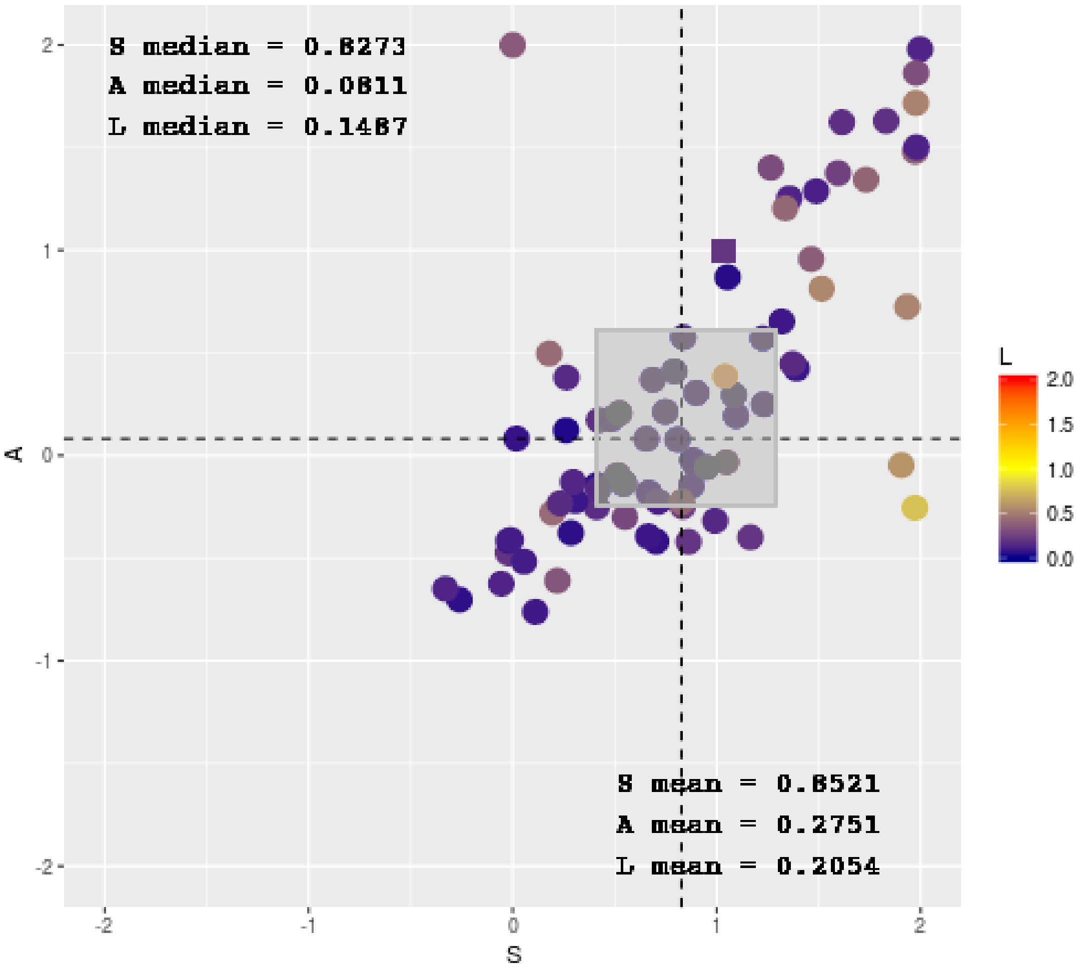
harp

fss arome
2017-06-01 - 2017-08-31
Lightning



spatial scale \ threshold	0.22	0.5	1	3	5
51	0.68	0.61	0.56	0.27	0.10
21	0.57	0.48	0.41	0.16	0.06
11	0.50	0.39	0.32	0.11	0.04
5	0.42	0.31	0.24	0.07	0.02
3	0.39	0.28	0.20	0.06	0.02
1	0.30	0.21	0.15	0.04	0.01

SAL arome
2017-06-01 - 2017-08-31
Lightning



harp

harpVis

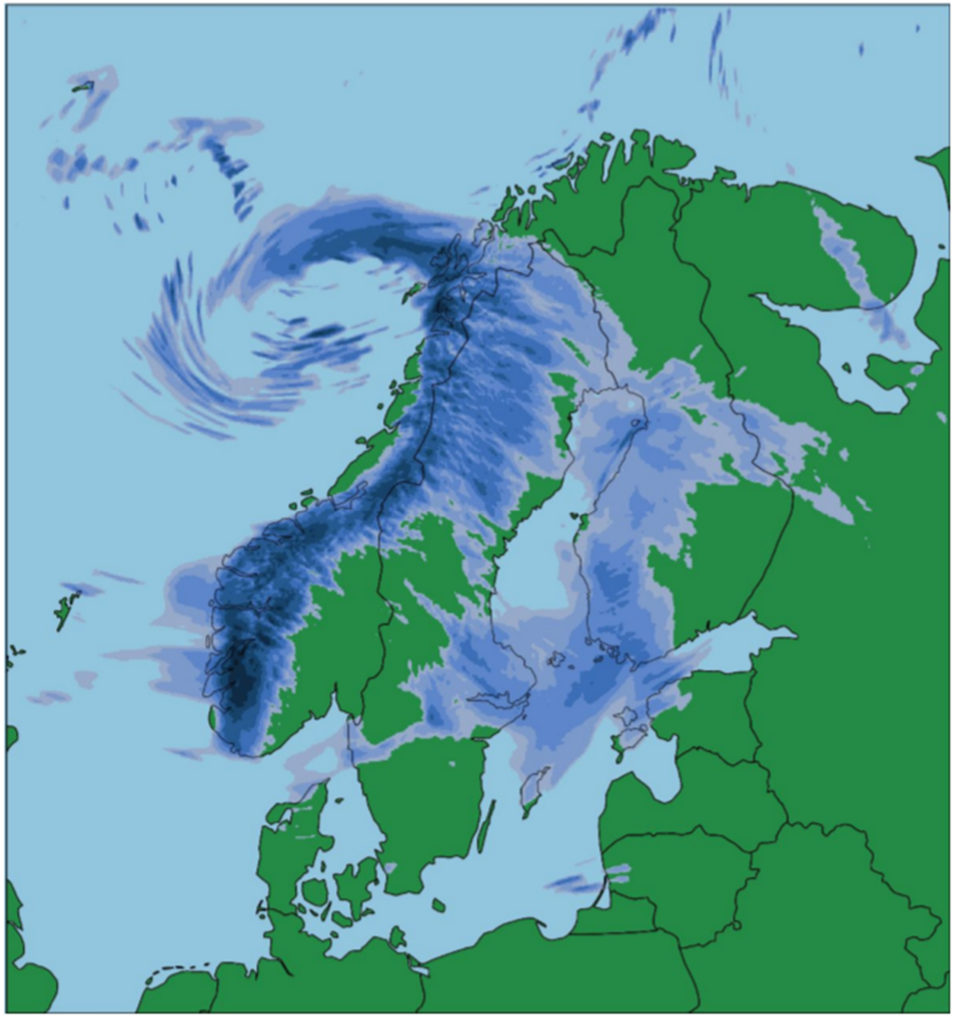
Functions for **potting** scores

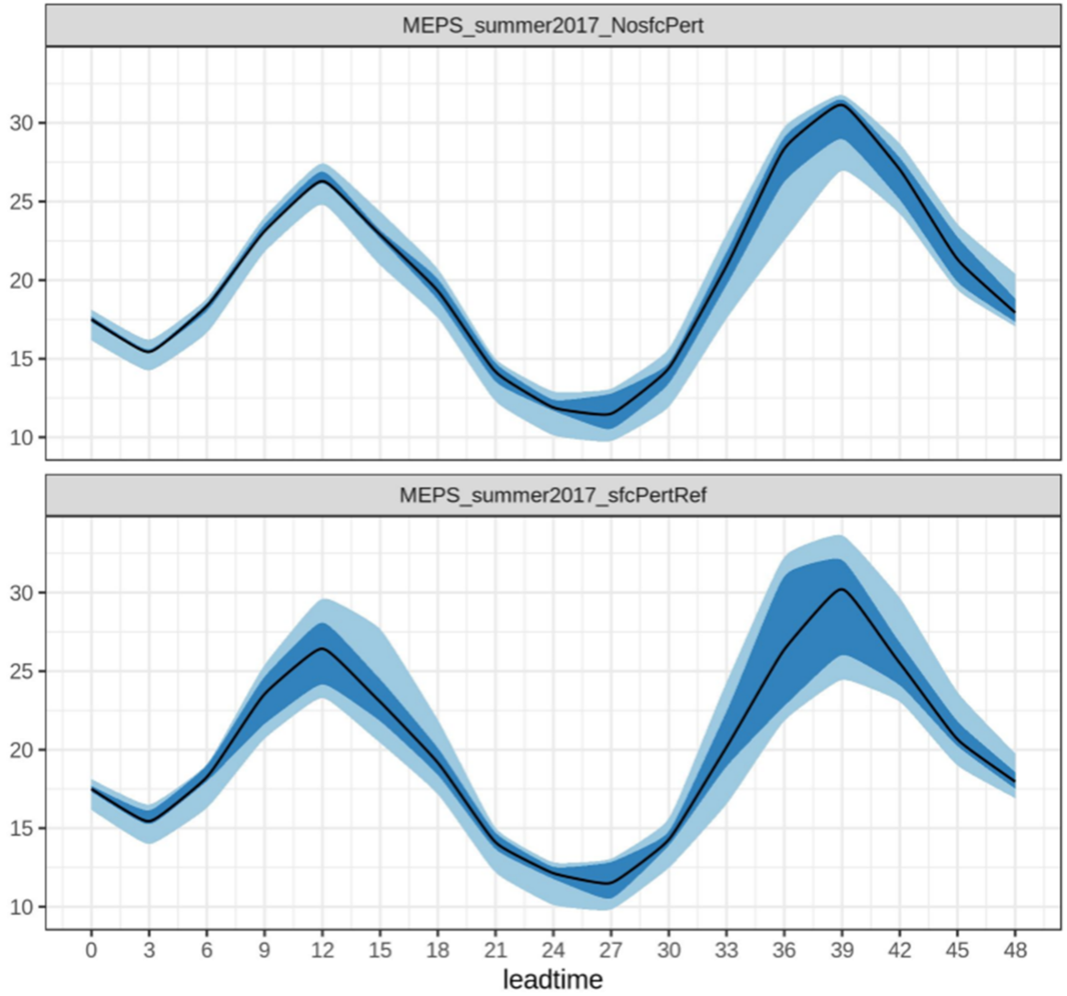
Functions for **plotting** meteorological data

- Maps
- "Meteograms"
- Experimental probabilistic visualizations

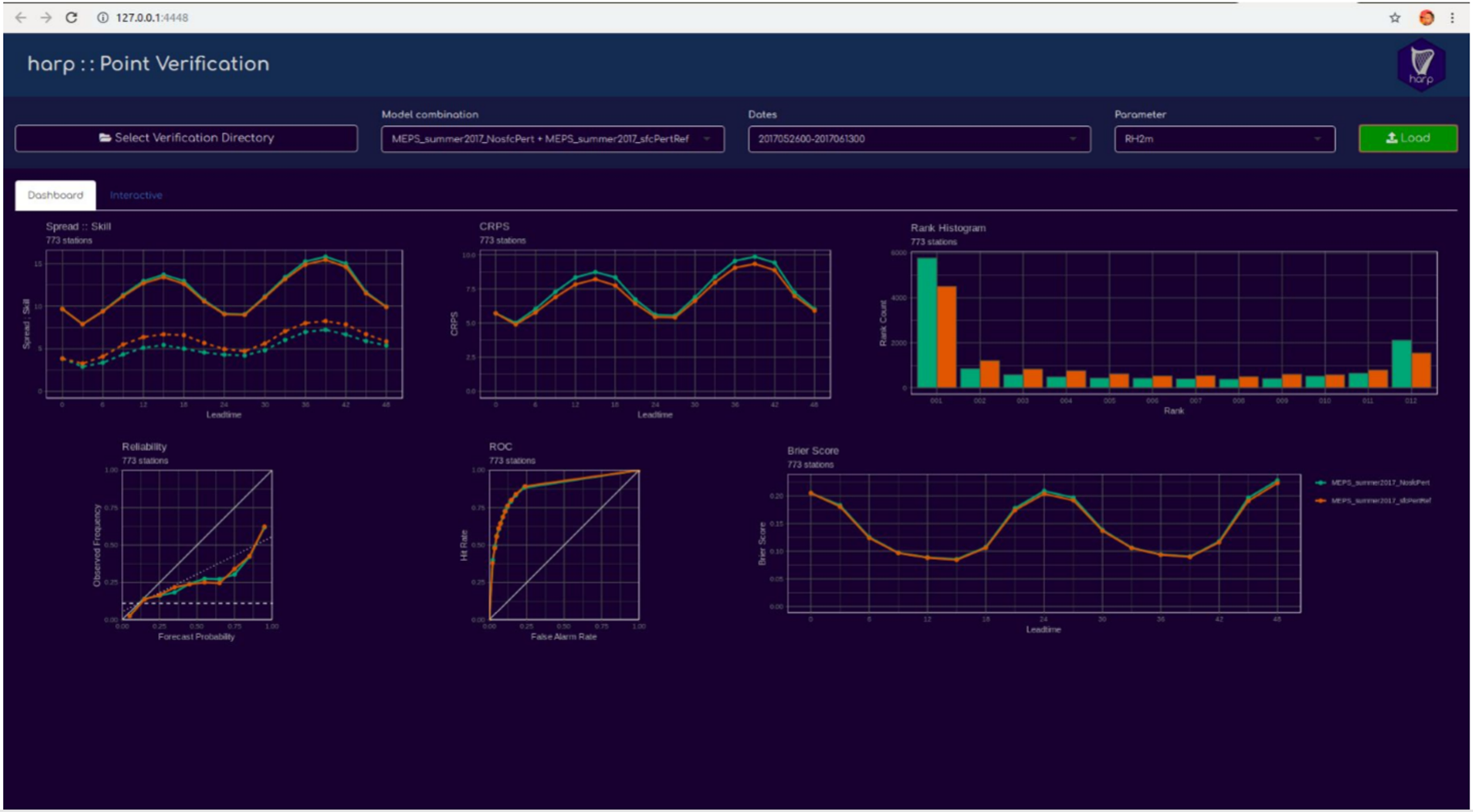
Interactive **shiny** app(s)

harp




MEPS_summer2017_NosfcPert
MEPS_summer2017_sfcPertRef
10
15
20
25
30
0
3
6
9
12
15
18
21
24
27
30
33
36
39
42
45
48
leadtime

harp



127.0.0.1:4448

harp :: Point Verification

Model combination: MEPS_summer2017_NosfcPert + MEPS_summer2017_sfcPertRef

Dates: 2017052600-2017061300

Parameter: RH2m

harp (v3)

What is still wanted

- Cleaning of packages
  - Improve and finish inline documentation
  - Bug-fixing
- Extend tutorials
- Add examples
- Have a user workshop (**14 – 17 October**)
- Add / develop more methods for spatial verification

```
install.packages(“harp”)
```

Thanks!