The Rhine Case

A compact, end-to-end example comparing multiple models for the Rhine by running one integrated verification pipeline. Learn the core *veriflow* workflow and how to interpret the resulting verification metrics.

enabling delta

Rijkswaterstaat
Ministerie van Infrastructuur en Waterstaat

Deltares

VORTECH

SWFT RESPONSE

Forecast verification with *veriflow* notebooks

Crash course + hands-on exercise

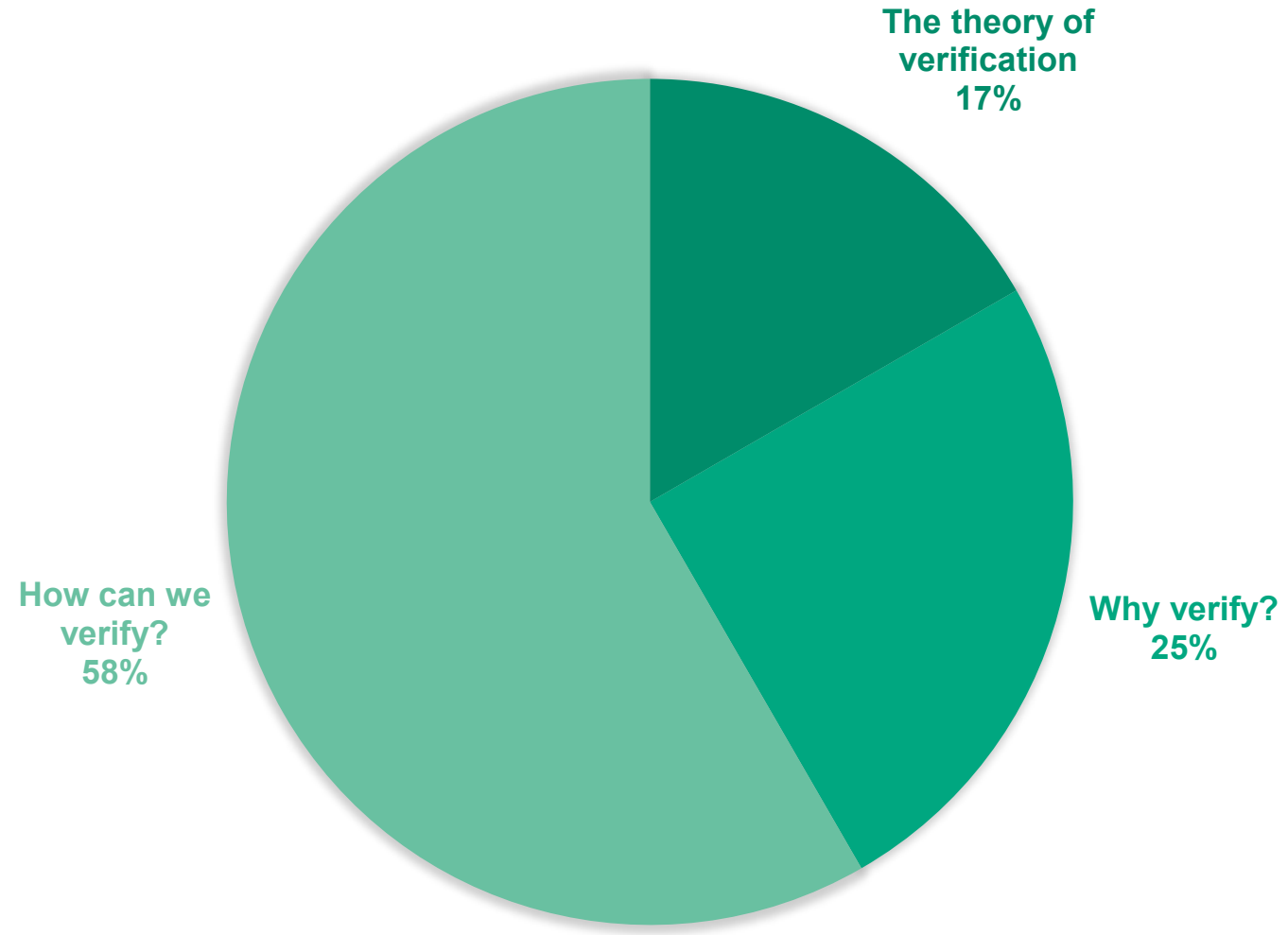
Maarten Smoorenburg, Jurian Beunk, Cees Voesenek, Aljen Uitbeierse, Dave Casson, Joost Driebergen

3 July 2026, Delft-FEWS Anwendertreffen @ LANUK Duisburg

Agenda

1. Two crash courses
   - Probabilistic forecasting
   - Forecast verification
2. Veriflow
3. Hands-on work
4. Discussions

The balance of this training



Focus on demo

You will each run your own verification pipelines

You will learn how to use Delft-FEWS for data access

You will each make your own visualisations

We will not: Spend a lot of time on in-depth theory of verification

Instead:

- We will cover some probabilistic scores
- We will share good references to theoretical background

By the end of this morning, you will be able to:

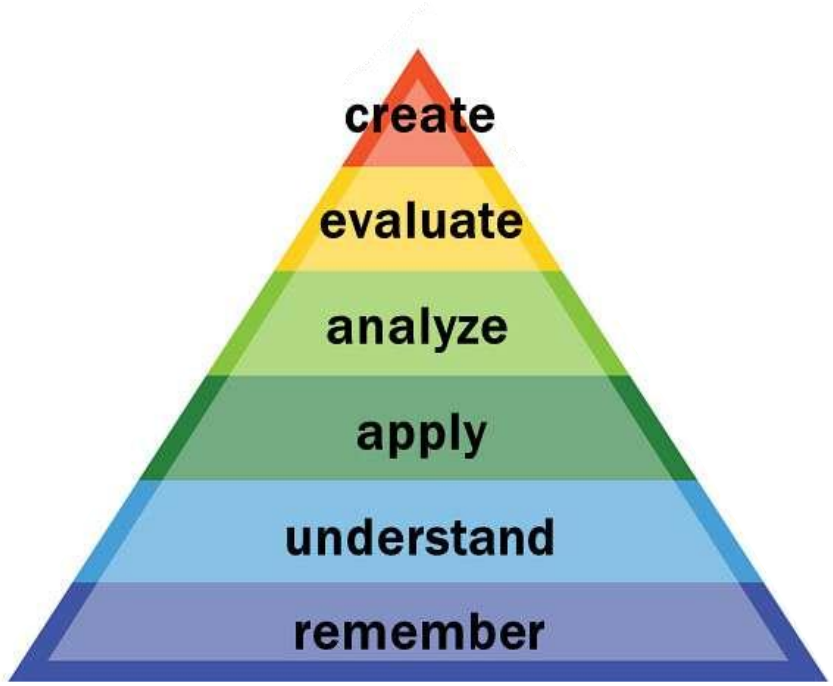
Understand the basics of verification and it's value

Understand how *veriflow* works and how it integrates with Delft-FEWS

Understand how *veriflow* can be applied in both research, model development and operations.

Run your own verification pipelines with *veriflow*

Make your own verification plots and visualisations



1 Crash Course: Probabilistic Forecasting

Why? What is it?

Learn more?

10

Nov

Probabilistic Forecasting - Short Course (DSD-INT 2026)

10 Nov 2026

09:00 - 16:30 (GMT+01:00)

Course (on-premises)

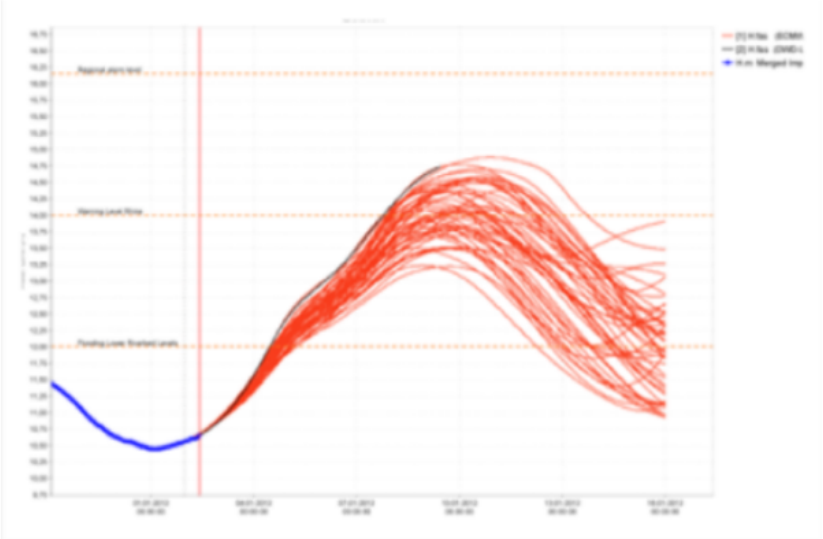
€ 629 (excluding 21% VAT)

Room : Yangtze River Delta

Remaining places : 15

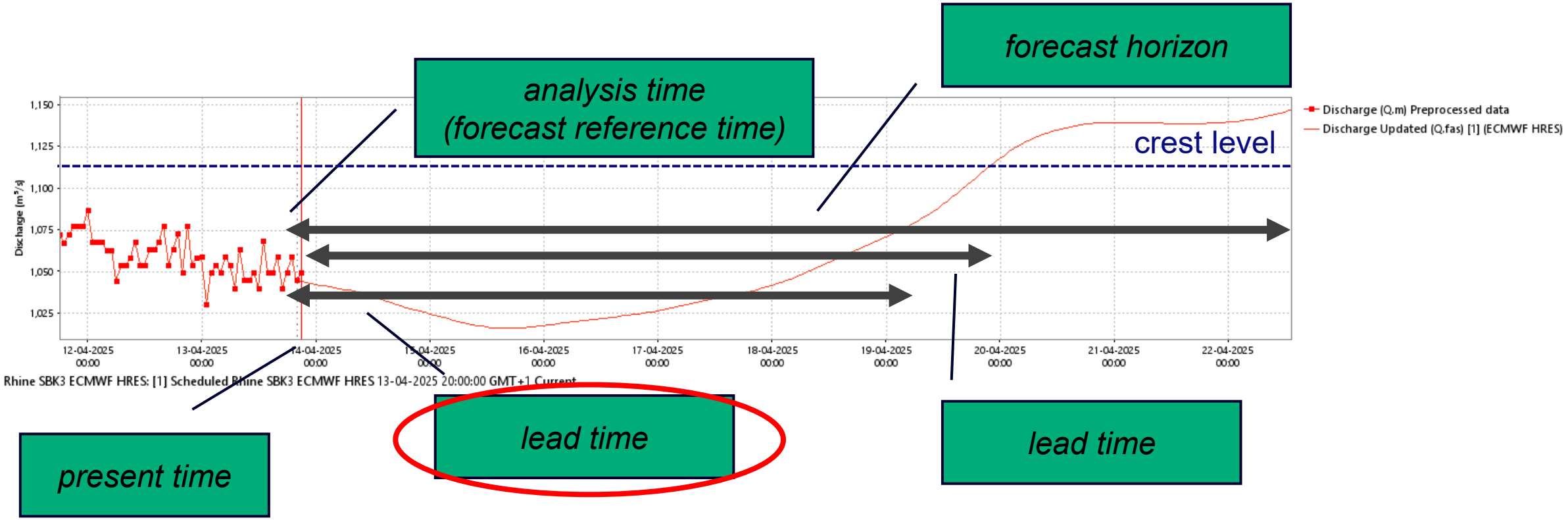
Academy

+ Add to my Calendar



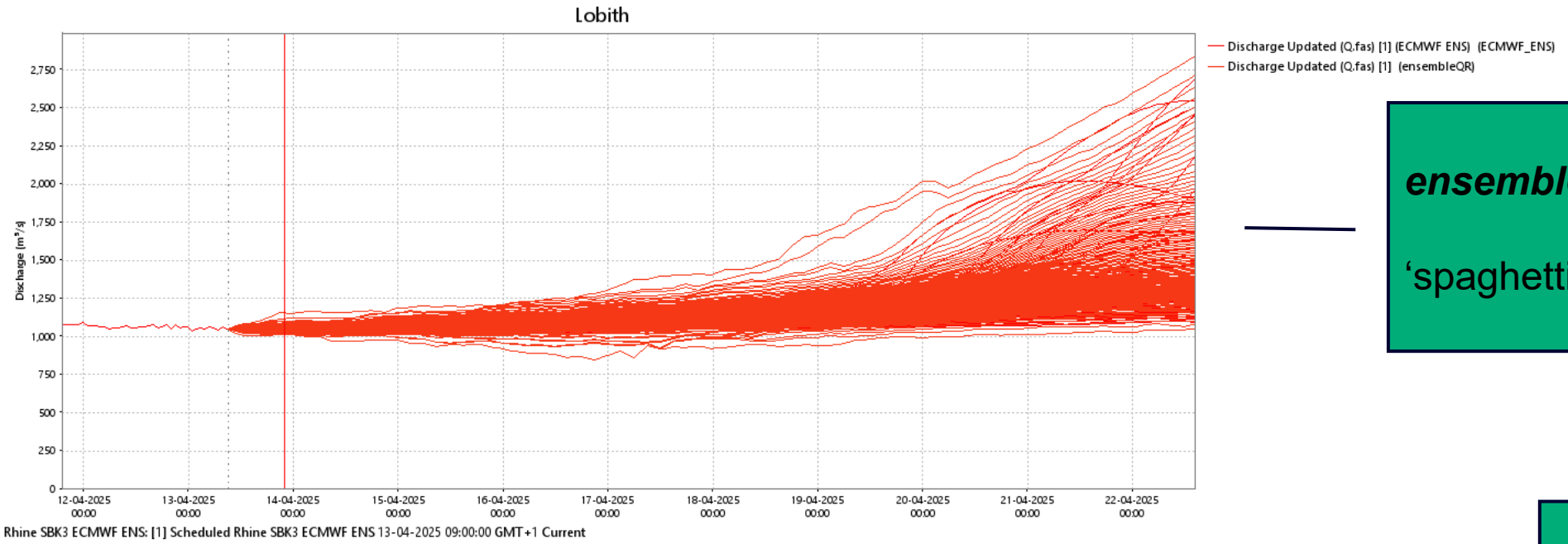
[Probabilistic Forecasting - Short Course (DSD-INT 2026) - Academy]

Definitions



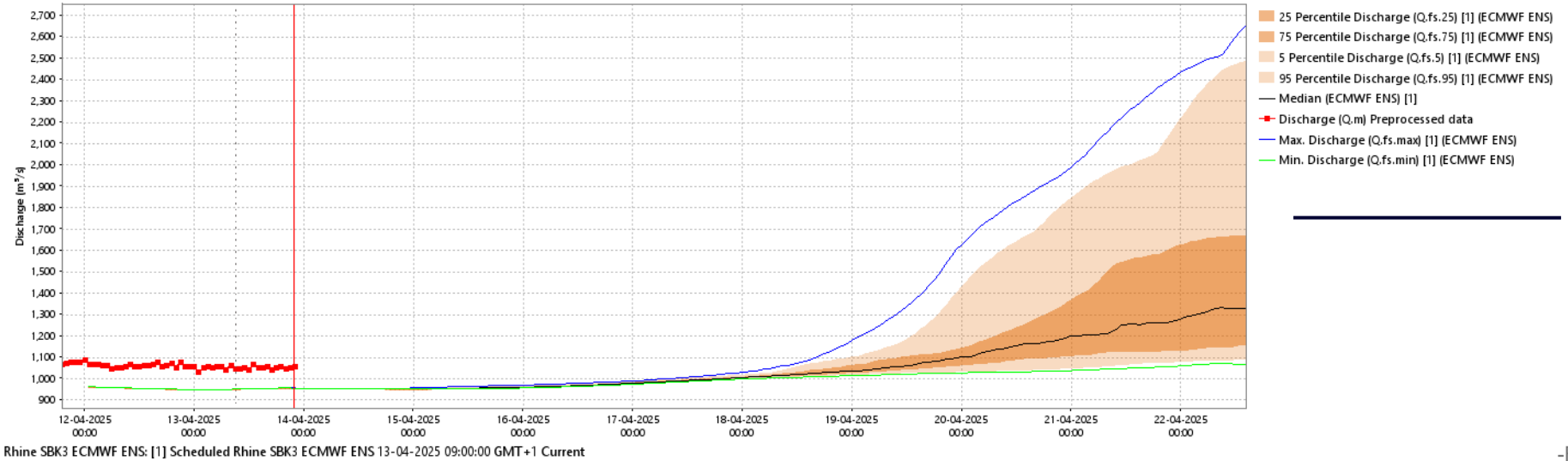
Other times:
Completion time: moment that forecast computation was finished
Issue time: time of publishing the (validated) forecast

Definitions



ensemble forecast

'spaghetti plot'



probabilistic forecast

with:
bands / intervals
quantiles/percentiles (of exceedance)

Why use probabilistic forecasts?

Why use probabilistic forecasts?

1. Future is uncertain → show this explicitly

2. Enable risk-based decision-making → risk = *probability* x *consequences*

3. Extending forecast lead time

4. Allows for separation of responsibilities between forecaster and decision-maker

2 Crash Course: Forecast Verification

Why? What is it? …. With a focus on probabilistic forecasting

Verification is about measuring quality of models and forecasts

Different attributes of quality exist

The quality of a simulation or forecast is often measured with respect to a “true” or “observed” value

Streamflow or waterlevel measurements
Raingauges and temperature measurements

And can also be expressed relative to another simulation or forecast

Why is verification important for the operational domain?

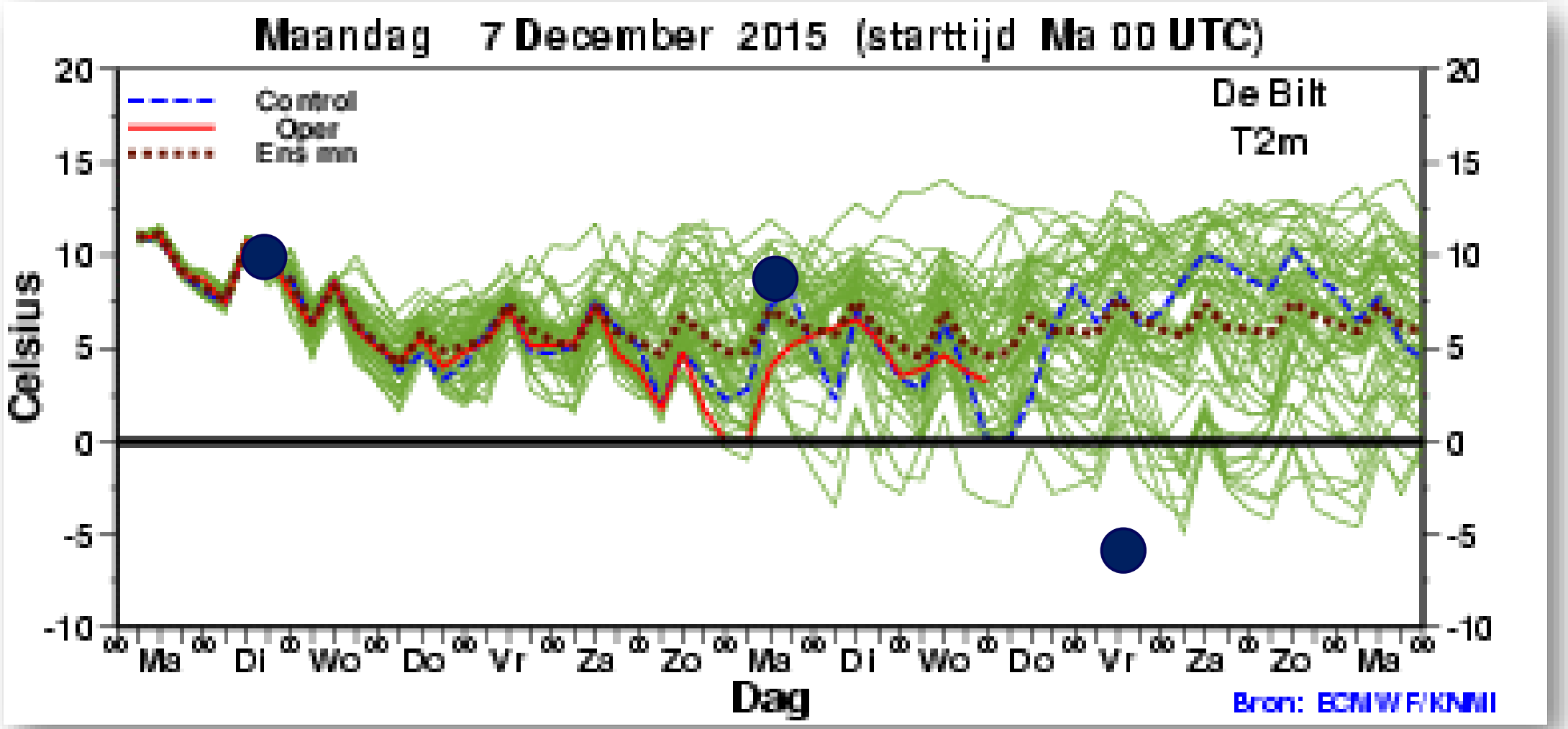
Real time monitoring	• Continuous monitoring of models to detect deteriorating model performance. Relevant for re-calibration and re-training of models.
Improving the forecast	• Continuous decision support in real-time forecasting
Model improvement	• Advanced insights in forecast performance and quality • Comparing model performance with different meteorological forcing
Reporting	• Reporting on model and forecast performance • Management KPIs

What is a good forecast?
Murphy, 1993

Quality	the correspondence between the forecasts and the matching observations
Value	the incremental economic and/or other benefits realized by decision makers through the use of the forecasts
Consistency	the correspondence between forecasters' judgments and their forecasts

Murphy, Allan H. "What is a good forecast? An essay on the nature of goodness in weather forecasting." *Weather and forecasting* 8.2 (1993): 281-293.

How good or bad is this forecast?



What to expect from a probability forecast?

Accuracy

- Average correspondence between observed and predicted values

Reliability

- Correspondence of predicted probabilities with observed relative frequencies

Sharpness

- Tendency to produce 0% and 100% probability forecasts
- A sharp ensemble forecast has a small width

“Sharpness, given reliability”

Attributes of forecast **quality**

TABLE 2. Short definitions and relevant distributions for various aspects of forecast quality.

Aspect	Definition	Relevant distribution(s)
Bias	Correspondence between mean forecast and mean observation	$p(f)$ and $p(x)$
Association	Overall strength of linear relationship between individual pairs of forecasts and observations	$p(f, x)$
Accuracy	Average correspondence between individual pairs of forecasts and observations	$p(f, x)$
Skill	Accuracy of forecasts of interest relative to accuracy of forecasts produced by standard of reference	$p(f, x)$
Reliability	Correspondence between conditional mean observation and conditioning forecast, averaged over all forecasts	$p(x\|f)$ and $p(f)$
Resolution	Difference between conditional mean observation and unconditional mean observation, averaged over all forecasts	$p(x\|f)$ and $p(f)$
Sharpness	Variability of forecasts as described by distribution of forecasts	$p(f)$
Discrimination 1	Correspondence between conditional mean forecast and conditioning observation, averaged over all observations	$p(f\|x)$ and $p(x)$
Discrimination 2	Difference between conditional mean forecast and unconditional mean forecast, averaged over all observations	$p(f\|x)$ and $p(x)$
Uncertainty	Variability of observations as described by distribution of observations	$p(x)$

Murphy, A.H. 'What Is a Good Forecast? An Essay on the Nature of Goodness in Weather Forecasting'. Weather and Forecasting 8, no. 2 (1993): 281–293.

Today we use CRPS and the Rank Histogram

… but won't go into the theory behind it

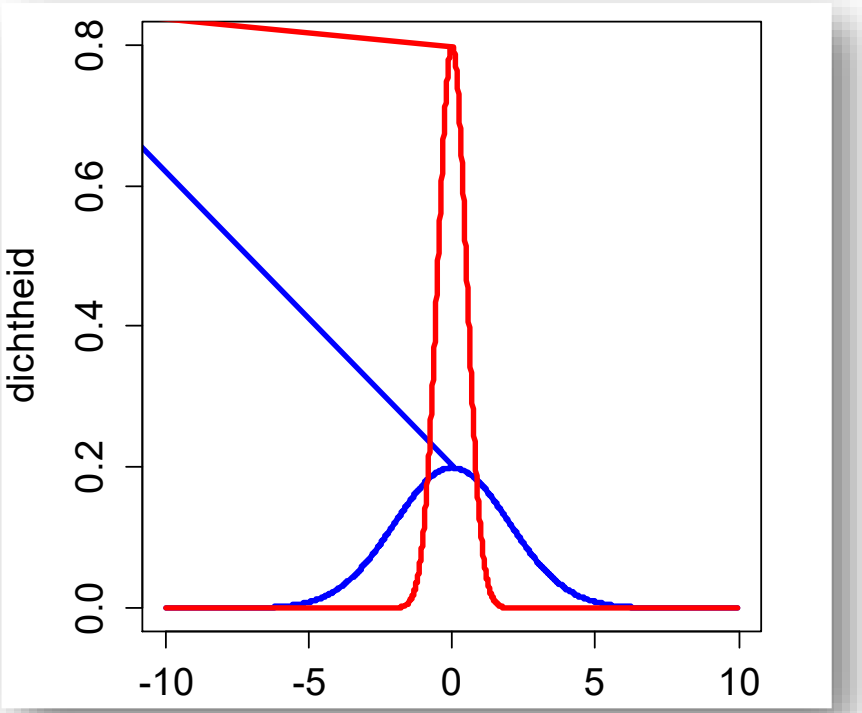
Rank histogram	Captures reliability `scores` tutorial
Continuous Ranked Probability Score	Captures accuracy, reliability and sharpness `scores` tutorial

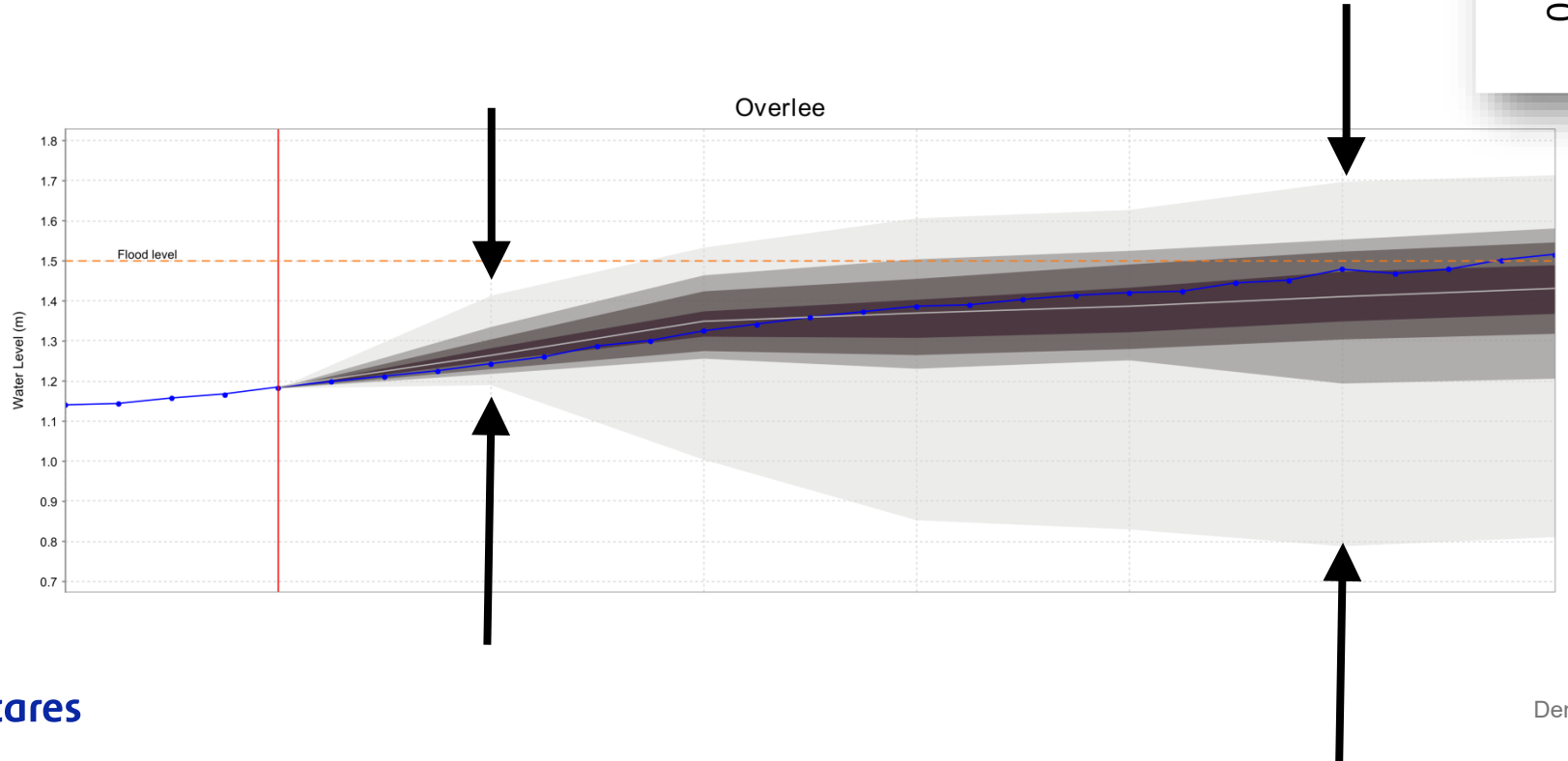
Sharpness

Tendency to produce 0% and 100% probability forecasts

→ measure of the width of a predictive distribution

What is ideal? Under which conditions?





Reliability

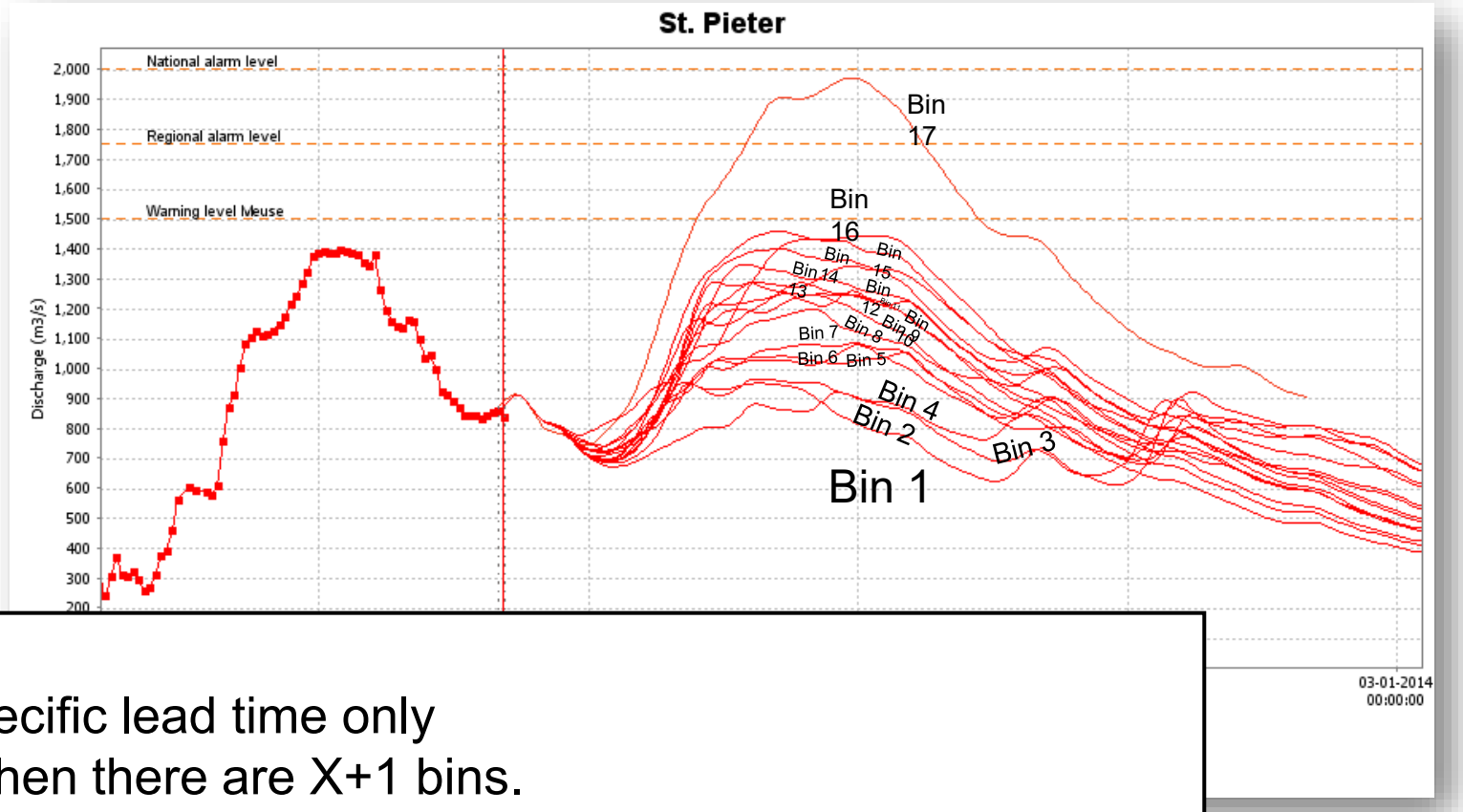
Measure with:

- **Rank Histogram (PIT diagram for probabilistic forecasts)**
- Reliability plot

Rank Histogram

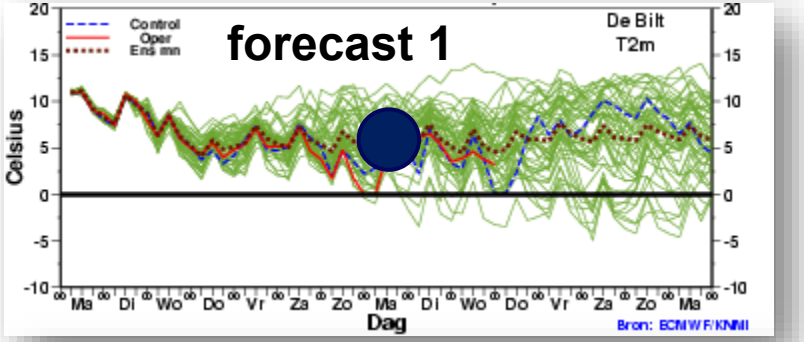
How often does the observation “land”…

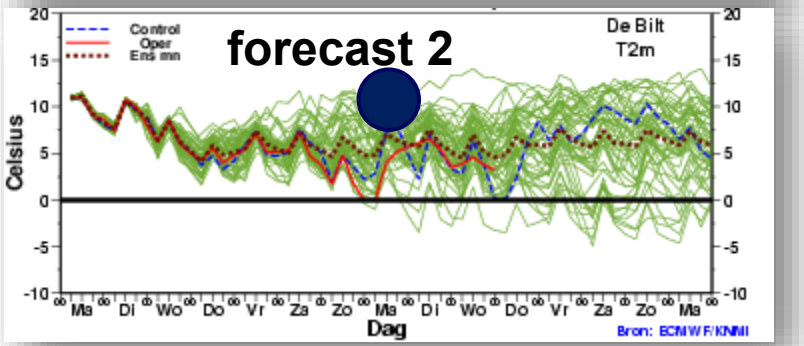
- … below the lowest member?
- … between …
- the lowest and second-lowest member?
- the second-lowest and third-lowest?
- the third-lowest and fourth-lowest?
- …
- … above the highest member?

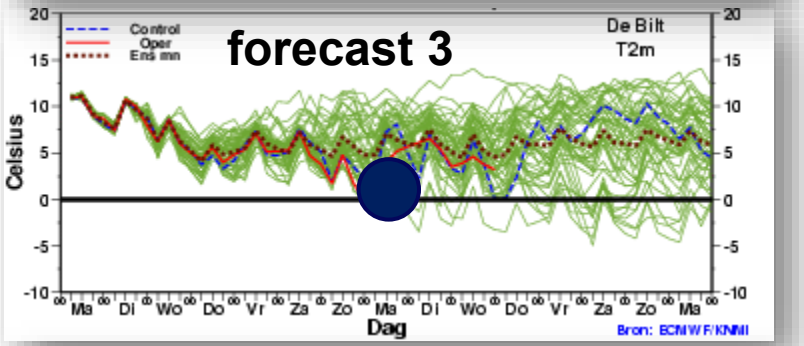


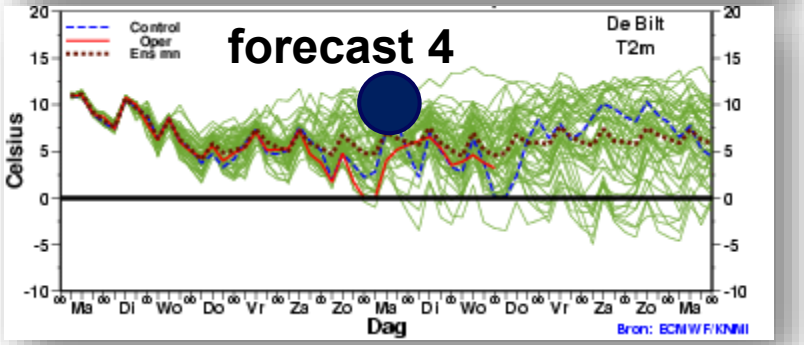
Nota Bene:

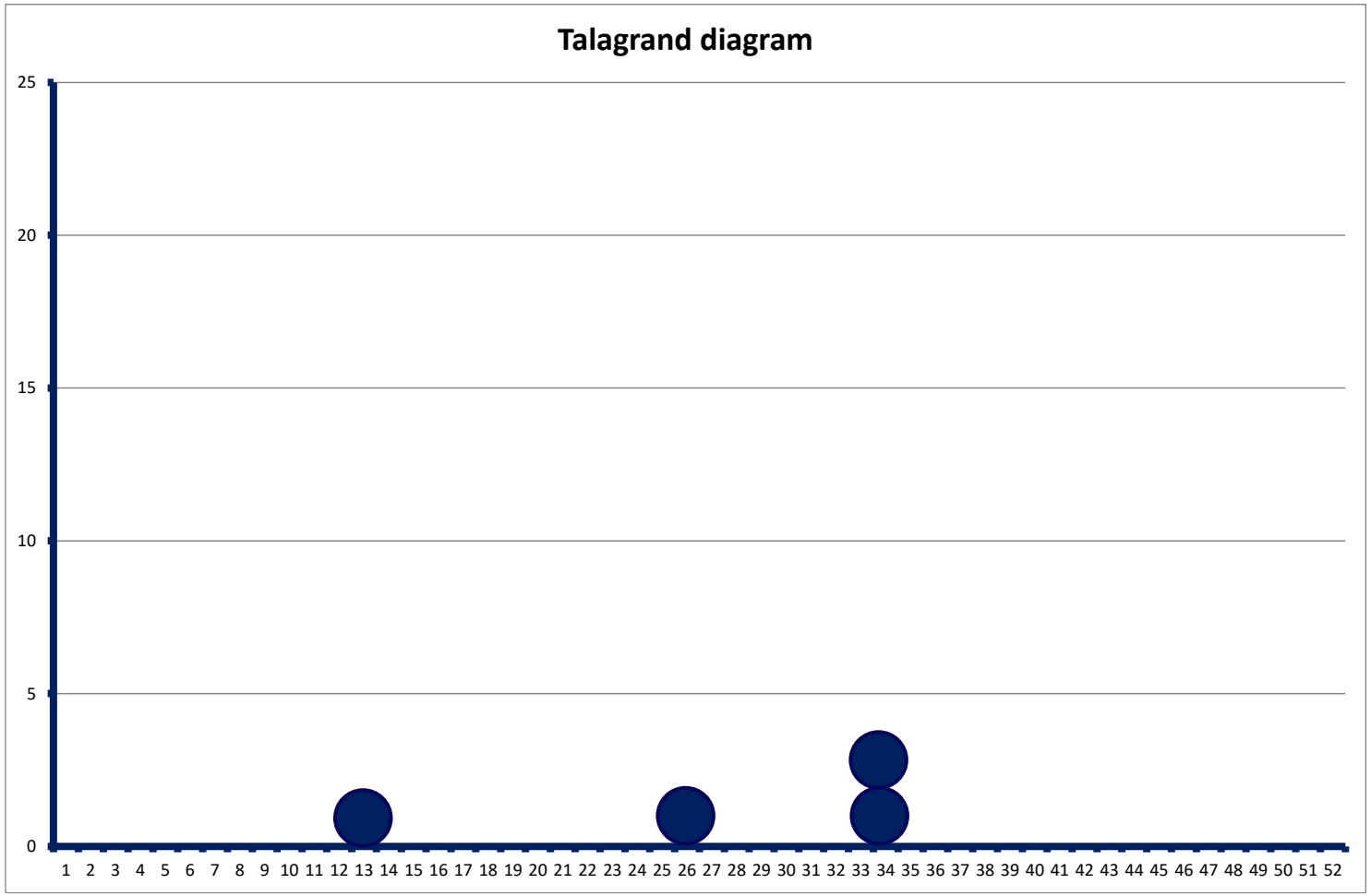
(1) Rank histograms are valid for a specific lead time only
(2) If your ensemble has X members then there are X+1 bins.
(3) Bins are not necessarily of equal width

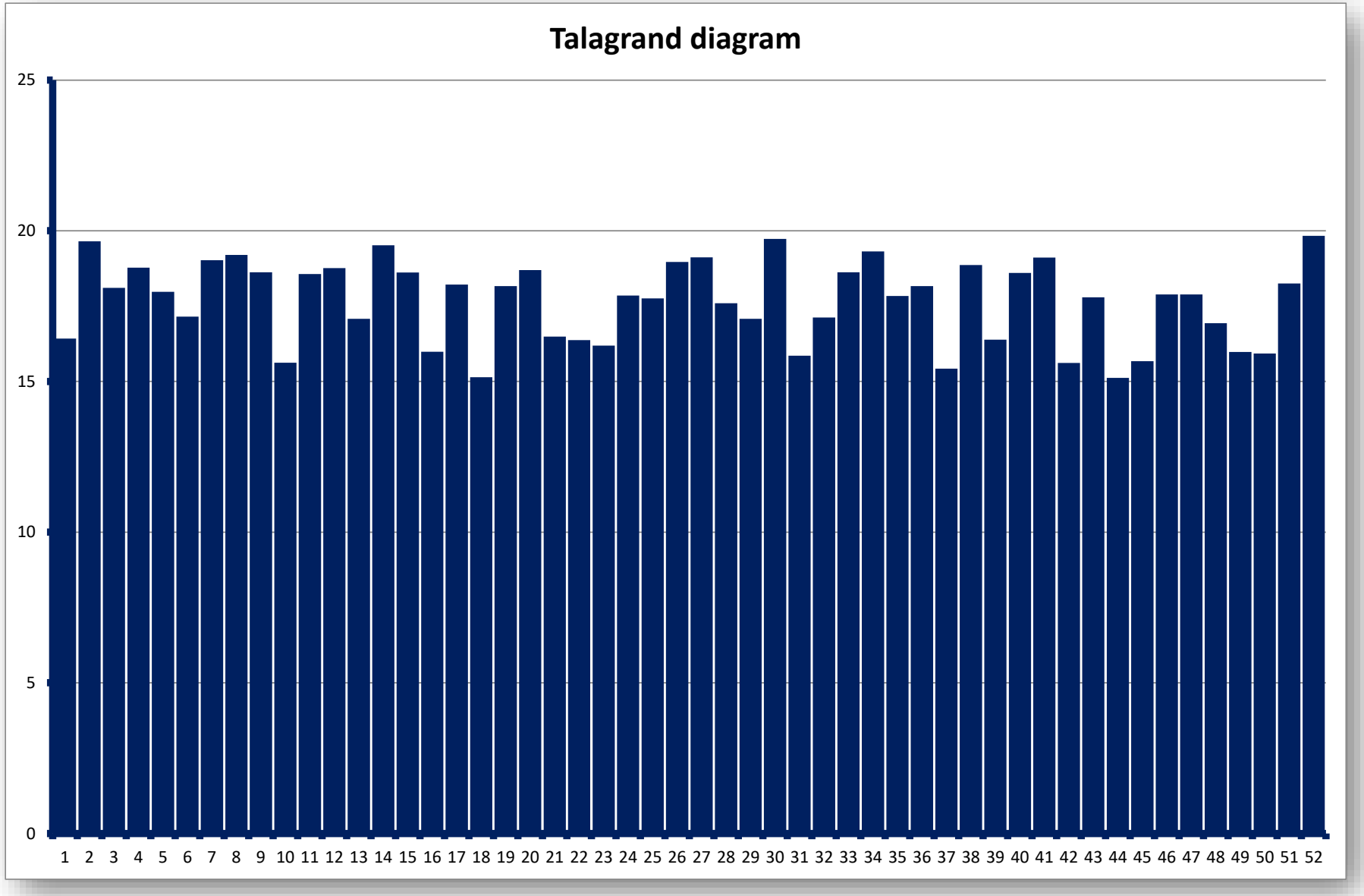

forecast 1
De Bilt
T2m
Control
Oper
Ens mn
Celsius
Dag
Bron: ECMWF/KNMI


forecast 2
De Bilt
T2m
Control
Oper
Ens mn
Celsius
Dag
Bron: ECMWF/KNMI


forecast 3
De Bilt
T2m
Control
Oper
Ens mn
Celsius
Dag
Bron: ECMWF/KNMI

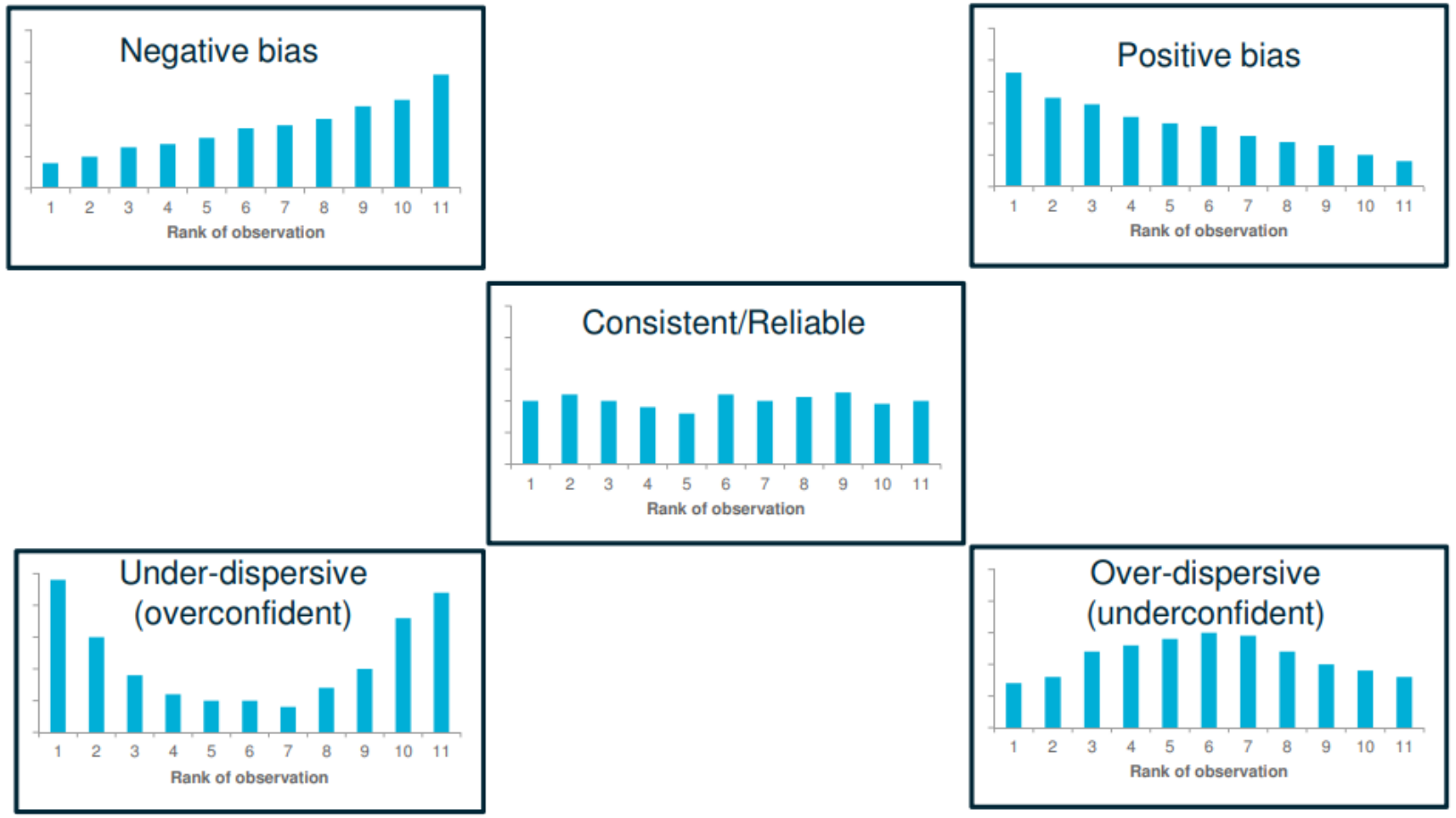

forecast 4
De Bilt
T2m
Control
Oper
Ens mn
Celsius
Dag
Bron: ECMWF/KNMI


Talagrand diagram
25
20
15
10
5
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52


Talagrand diagram
25
20
15
10
5
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52

Australian Government
Bureau of Meteorology

Rank histogram

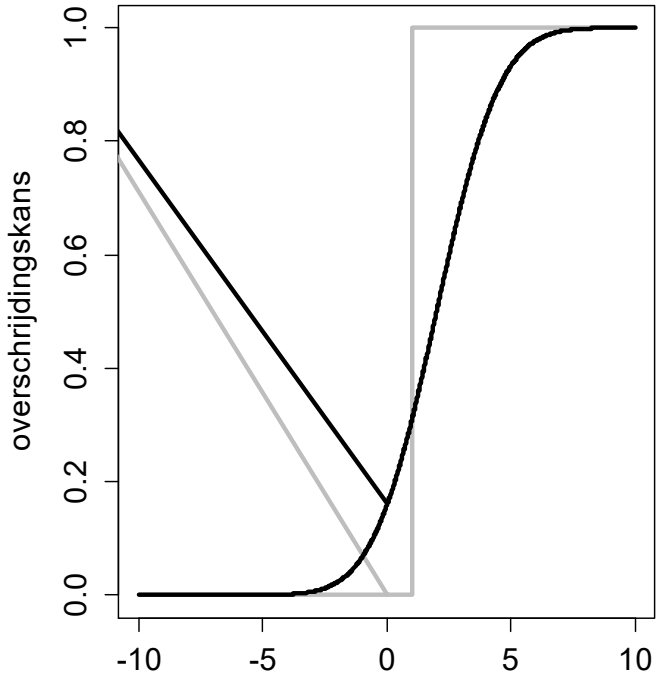


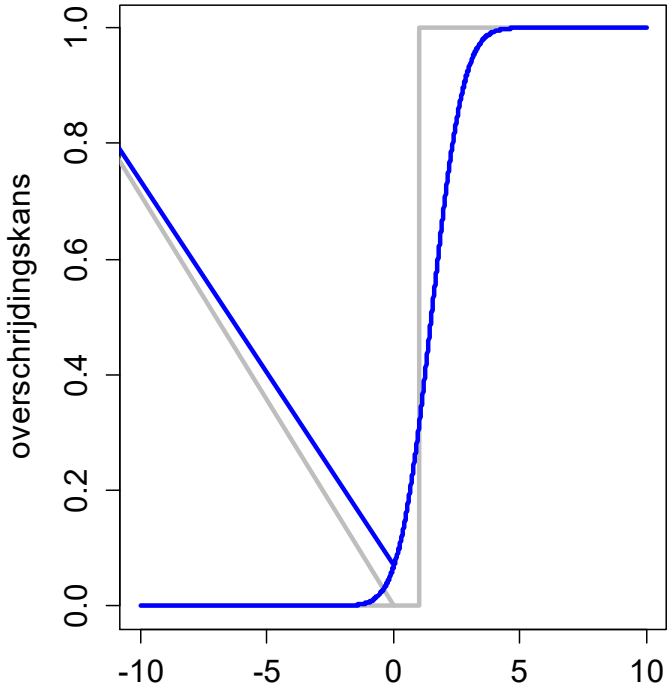
Example courtesy of Schroedter-Homscheidt et al., 2016.
(DOI: 10.13140/RG.2.2.10959.94883)

Continuous Ranked Probability Score (CRPS)

- Measure for *accuracy*, *reliability* and *sharpness*
- Area of the difference between the forecast and observed CDF
- Generalization of Mean Absolute Error for probabilistic forecasts

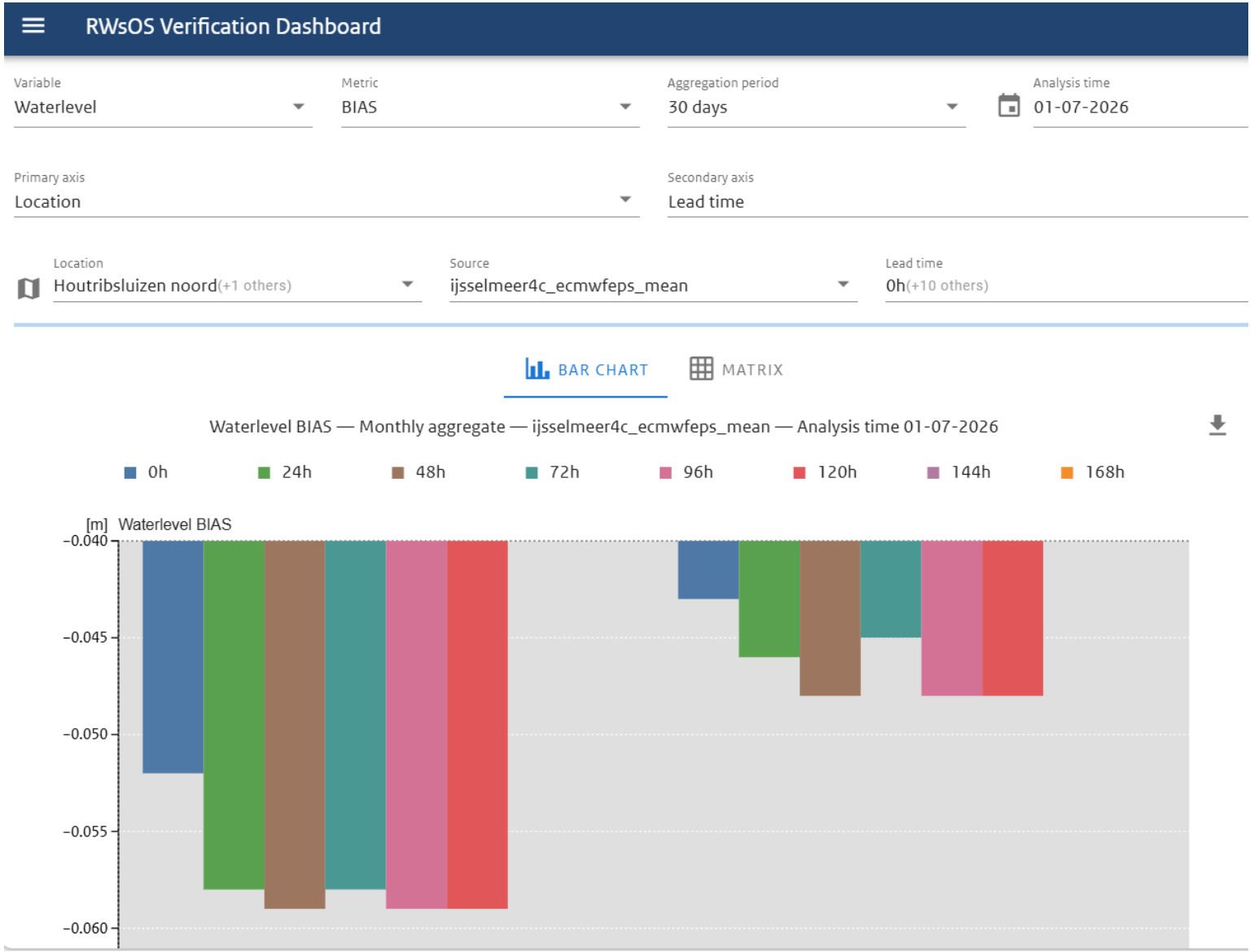
Simple metric for comparing forecasts
→ skill measure with CRPSS





In the operational forecasting domain, there’s large potential for improving forecast verification practices

Operational dashboard for detection and analysis



Rijkswaterstaat
Ministerie van Infrastructuur en Waterstaat

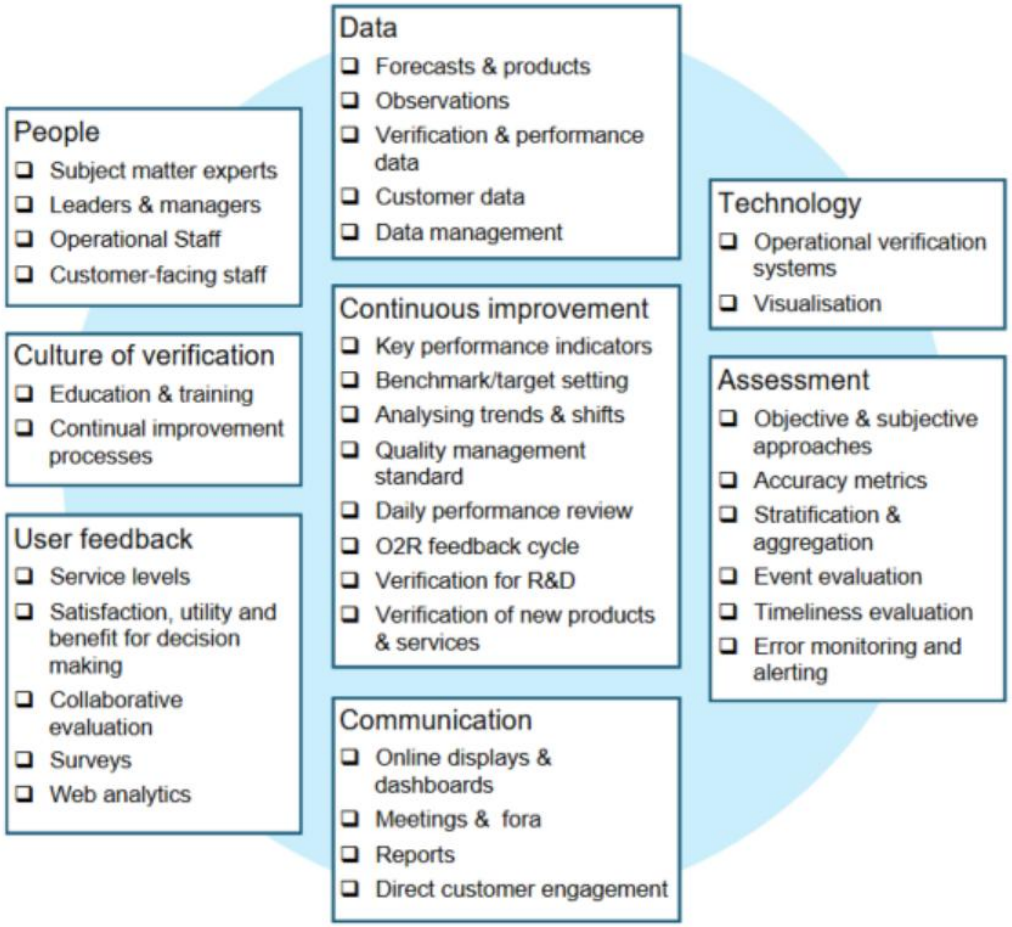
Deltares

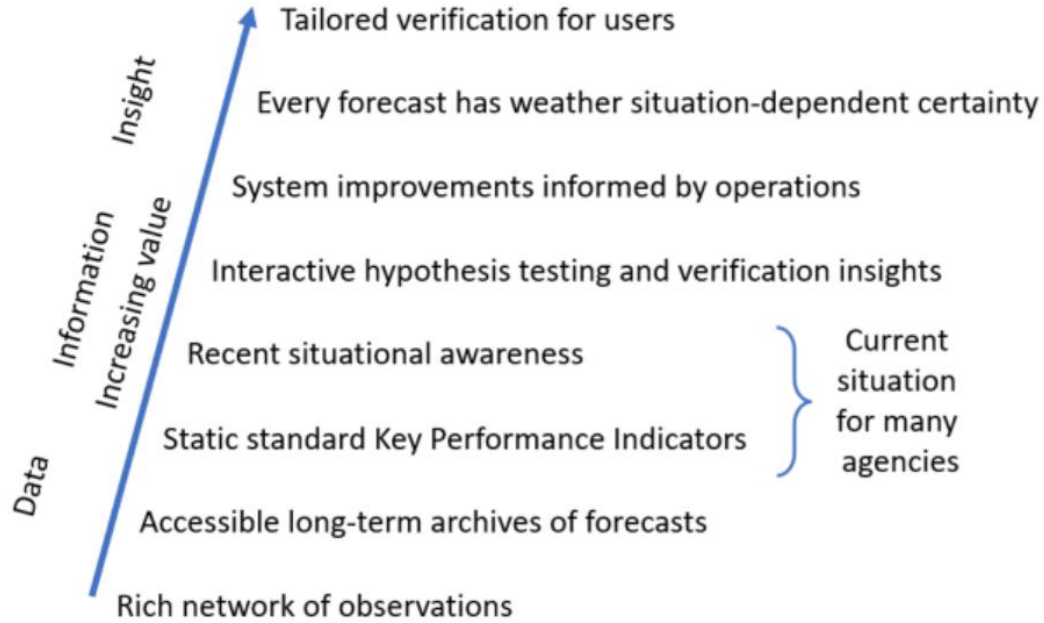
Operational dashboard for detection and analysis

Rijkswaterstaat
Ministerie van Infrastructuur en Waterstaat

Deltares

Maturity of forecast verification at your organization?





e Verification Information Value Ladder showing the foundations for forecast verification at the bottom and increasing value of agency moves up the ladder. Reprinted from Pagano et al. (2024). with permission © American Meteorological Society.

Pagano, T.C., Ebert, E.E. and Khanarmuei, M. (2025), Enhancing Forecast Verification in National Meteorological and Hydrological Services. Meteorol Appl, 32: e70051. https://doi.org/10.1002/met.70051

3 Veriflow - background

Why? What is it?

Outline

Why?

What is it?

What makes it special?

Future work!

Why we started with veriflow?

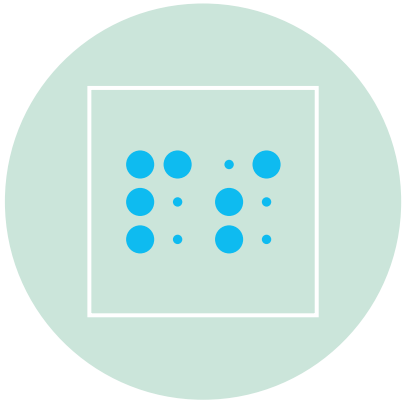
Question:

“How can we further establish solid verification practices in model development and operations?”

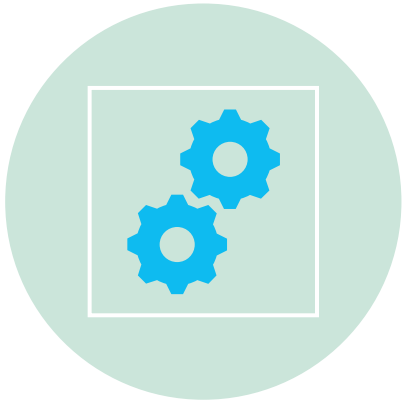
Answer:

“A robust and reproducible verification pipeline for model developers and operational forecasters”

Challenges when integrating verification in practice



Translating user-needs to a verification strategy



Technical challenges

Organisational integration

General technical challenges

Handling large data volumes

Transforming and matching data

Computing complex metrics

Current practices in the operational domain

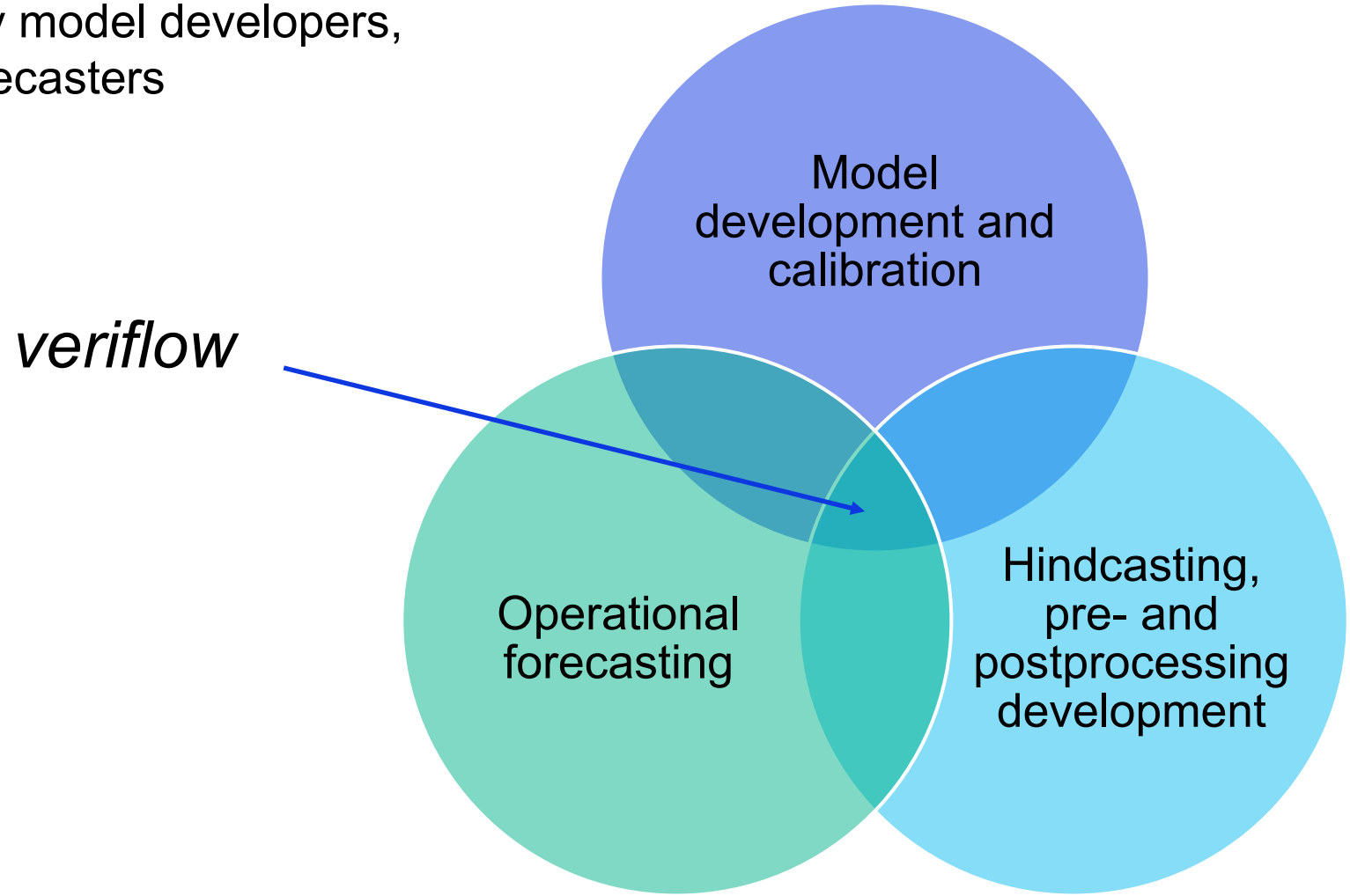
Manual and fragmented	different users applied their own scripts, spreadsheets, or tools
Use-case specific	solutions built for one use-case may have limited value for others
Time-consuming	data extraction, processing, and comparison required many manual steps
Hard to reproduce	methods and assumptions weren’t consistently documented

So we started developing *veriflow*

To enable Delft-FEWS users to perform **verification and skill assessmen**t of operational forecasts in an **efficient, robust and reproducible way** — for all types of forecasts (deterministic, ensemble, and probabilistic)

And to standardize verification practices between research, development and operational application

So it can be used by model developers, researchers and forecasters



The story of *veriflow*

Deltares started development in 2024
Collaboration with Rijkswaterstaat

In collaboration with Vortech (scientific software engineers)
- Code review
- Architecture

A Python package

It's open source (MIT)

Tested and documented
- Docs: https://deltares.github.io/veriflow/
- GitHub: https://github.com/Deltares/veriflow

README Contributing MIT license

pypi v0.4.1 | docs latest | codecov 82% | CI failing | docs build passing

Veriflow

A reproducible verification pipeline for evaluating model outputs and forecasts.

- Fetching data
- Computing scores
- Writing results

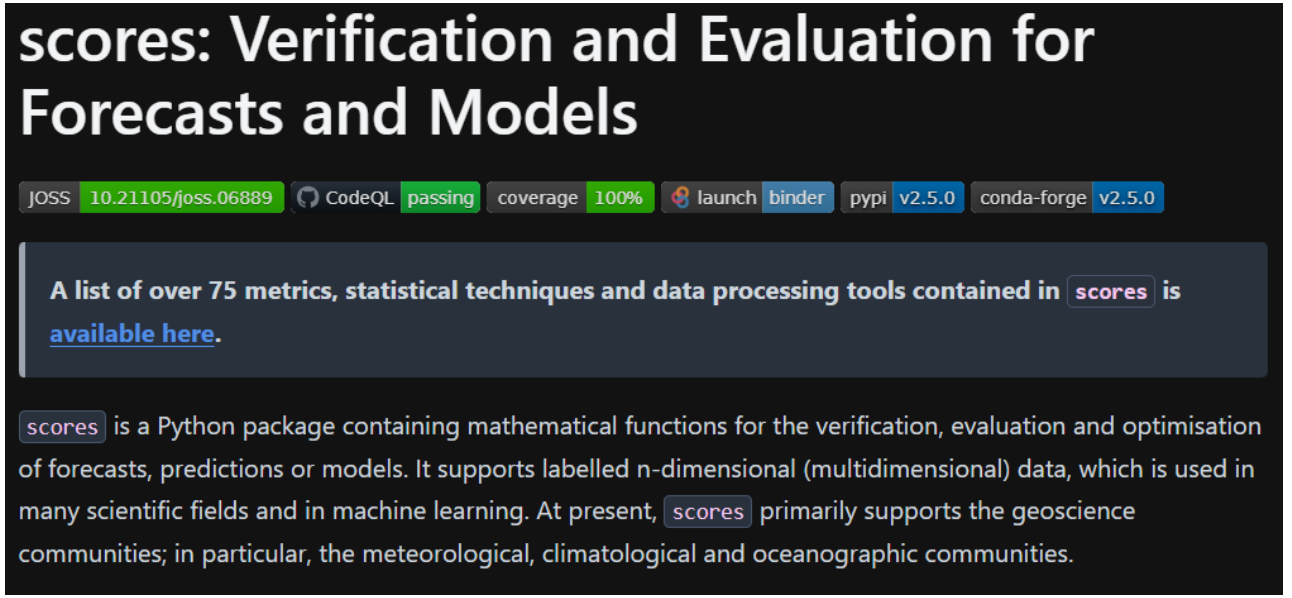
Key features

- ✅ Full control over the verification pipeline via configuration
- ✅ Native integration with Delft-FEWS
- ✅ Native integration with local and remote (S3) Zarr
- ✅ Builds on scores for computation of scores. This package has extensive functionality, and it's documentation is world-class.
- ✅ Extensible with your own (private) datasources, scores and datasinks
- ✅ Optimized internal datamodel for efficient computation

We make use of *scores* for computation of metrics

- Developed by the BoM (Bureau of Meteorology, Australia)
- Supports continuous, categorical, probabilistic and spatial verification scores

→ World-class documentation and tutorials
→ Active development
→ Fast computation
→ R&D on new scores

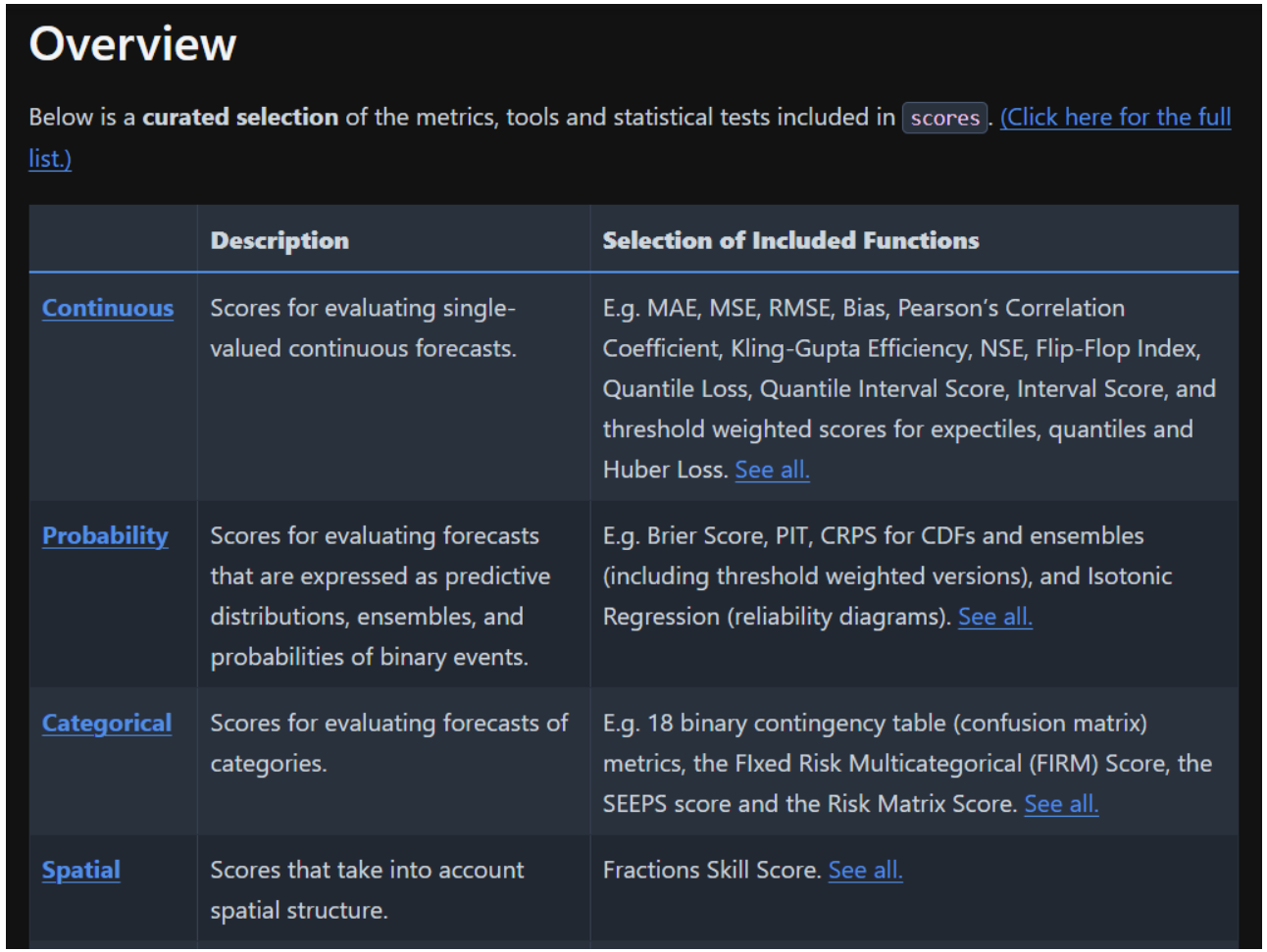


Leeuwenburg, T., Loveday, N., Ebert, E. E., Cook, H., Khanarmuei, M., Taggart, R. J., Ramanathan, N., Carroll, M., Chong, S., Griffiths, A., & Sharples, J. (2024). scores: A Python package for verifying and evaluating models and predictions with xarray. Journal of Open Source Software, 9(99), 6889. https://doi.org/10.21105/joss.06889

The scores documentation is world-class

4 important categories of verification scores exist

- Continuous
- Probabiliy
- Categorical
- Spatial



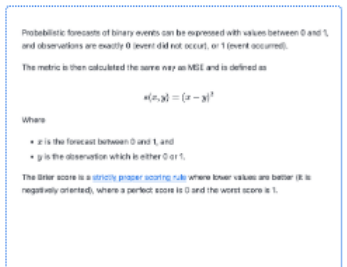
Overview

Below is a **curated selection** of the metrics, tools and statistical tests included in `scores`. (Click here for the full list.)

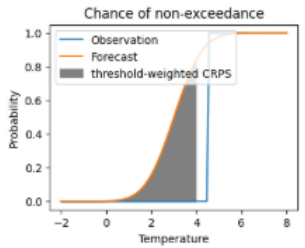
	Description	Selection of Included Functions
Continuous	Scores for evaluating single-valued continuous forecasts.	E.g. MAE, MSE, RMSE, Bias, Pearson's Correlation Coefficient, Kling-Gupta Efficiency, NSE, Flip-Flop Index, Quantile Loss, Quantile Interval Score, Interval Score, and threshold weighted scores for expectiles, quantiles and Huber Loss. See all.
Probability	Scores for evaluating forecasts that are expressed as predictive distributions, ensembles, and probabilities of binary events.	E.g. Brier Score, PIT, CRPS for CDFs and ensembles (including threshold weighted versions), and Isotonic Regression (reliability diagrams). See all.
Categorical	Scores for evaluating forecasts of categories.	E.g. 18 binary contingency table (confusion matrix) metrics, the FIxed Risk Multicategorical (FIRM) Score, the SEEPS score and the Risk Matrix Score. See all.
Spatial	Scores that take into account spatial structure.	Fractions Skill Score. See all.

Scores tutorials

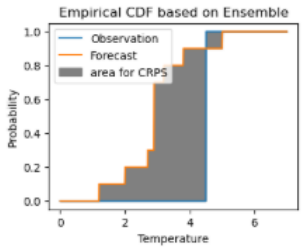
Probability



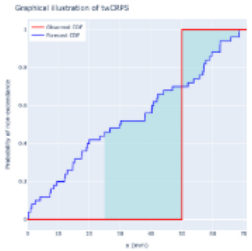
[Brier Score]



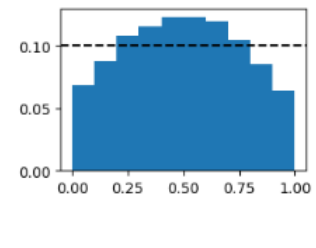
[CRPS for Forecasts Expressed as CDFs]



[CRPS for Ensemble Forecasts]

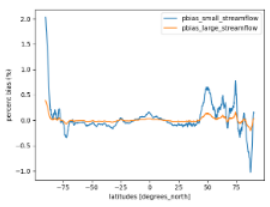


[twCRPS for Ensemble Forecasts]

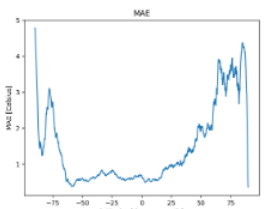


[Probability Integral Transform (PIT)]

Continuous

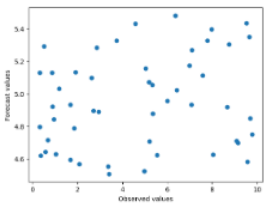


[Additive Bias, Multiplicative Bias and Percent Bias]

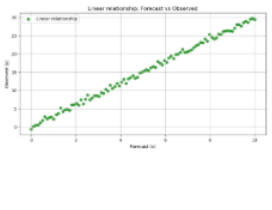


[MAE]

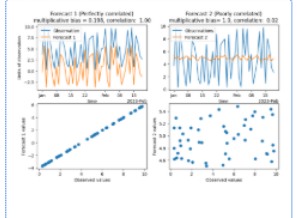
[Nash–Sutcliffe Model Efficiency Coefficient (NSE)]



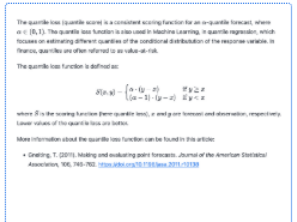
[Pearson's Correlation]



[Spearman's Correlation]



[Kling-Gupta Efficiency]



[Quantile Loss]

https://scores.readthedocs.io/en/stable/tutorials/Tutorial_Gallery.html

Easy use in Jupyter Notebooks (stand alond or server)

Inspirations from BoM's Jive system

(Loveday et al., 2024)

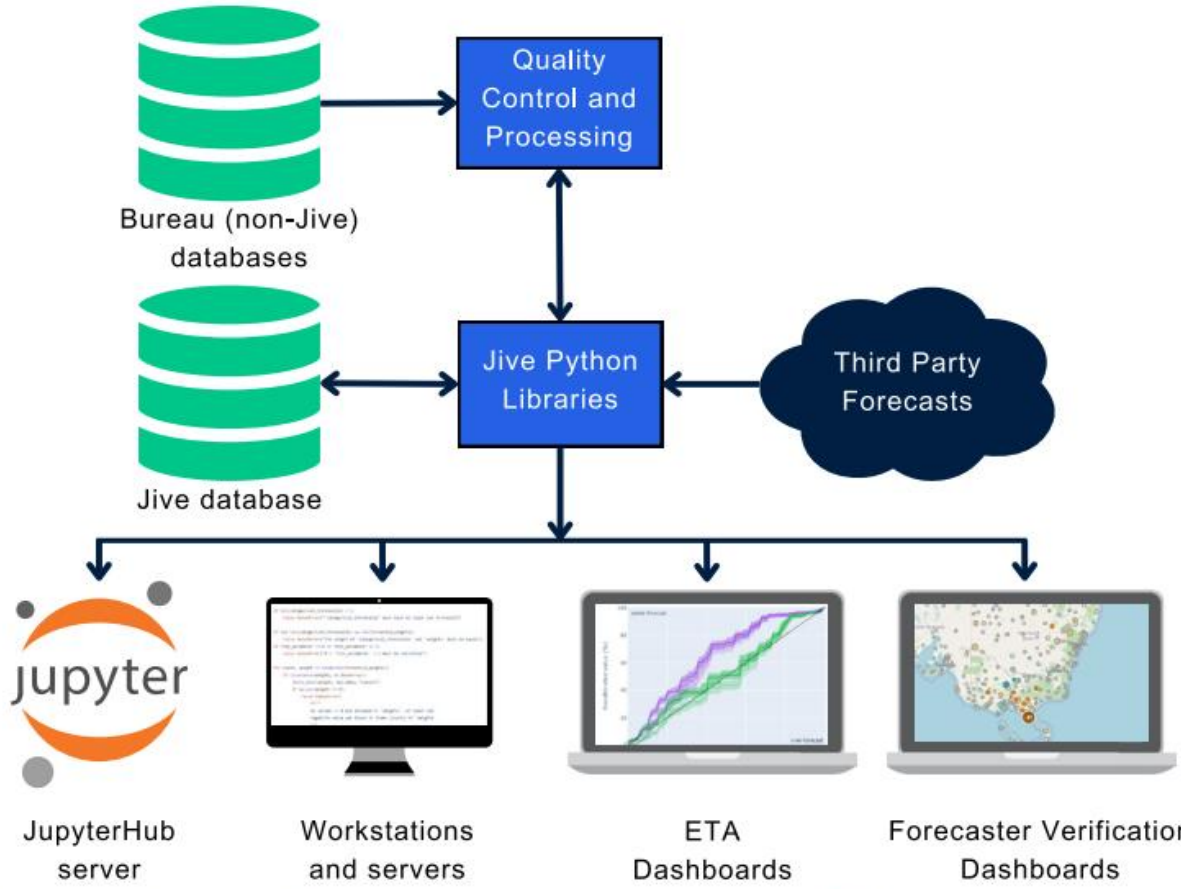


FIG. 2. Schematic diagram of the Jive system. Bureau databases supply forecasts and raw observations. Integrated Jive Python libraries do everything from data retrieval, processing, and storage, as well as generate statistics and visualize outputs on dashboards.

So what can *veriflow* do for you?

- Veriflow provides you with **flexible and reproducible verification pipeline**
- **Lowers the technical barrier** for verification
- Frees up experts to focus on **interpreting results** instead of managing data
- Builds trust and accountability by making verification **configurable and reproducible**
- Enables **consistent and comparable** verification across models, locations and times

Just like Delft-FEWS, veriflow is a community-driven effort…

We welcome contributions and hope to work together to further establish good verification practices in the operational domain

Veriflow is a reproducible verification pipeline tool

Connects to any Delft-FEWS system for data retrieval

Automates the verification workflow — from data retrieval to computation to result storage

Uses a simple configuration — specify what to verify and how

Supports all forecast types — deterministic, ensemble, and probabilistic

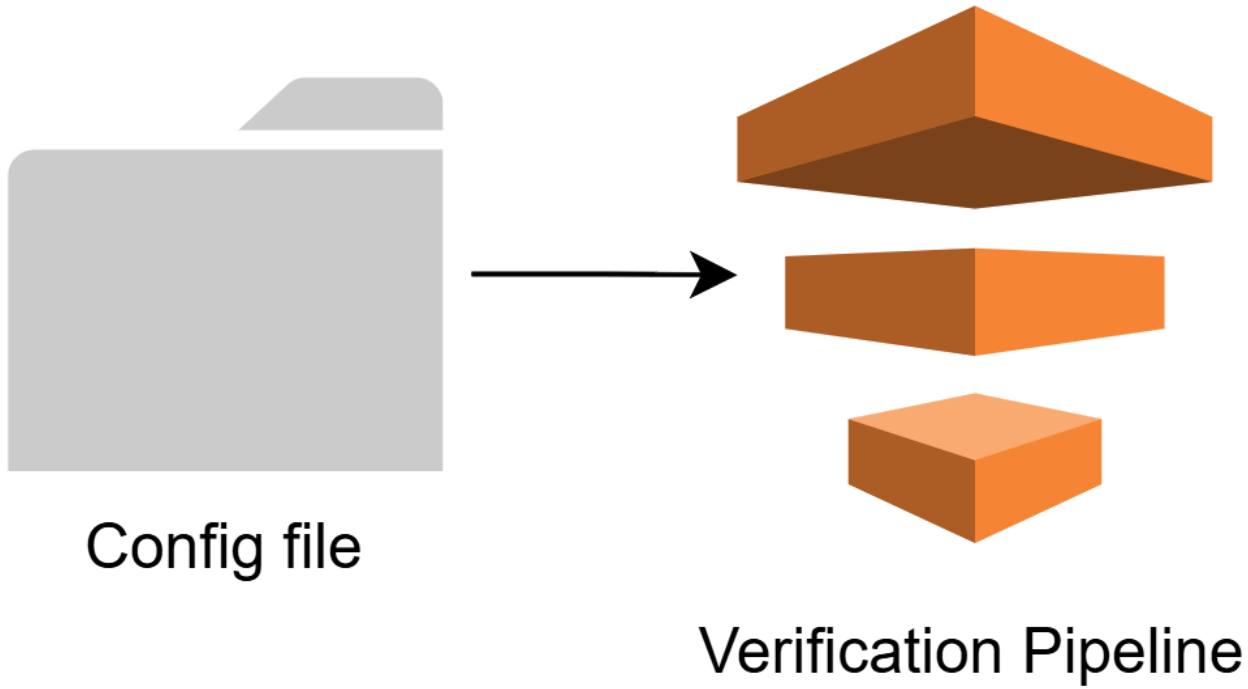
Has flexible inputs — allows you to connect to a private database

Has flexible scores — allows you implement your own custom scores

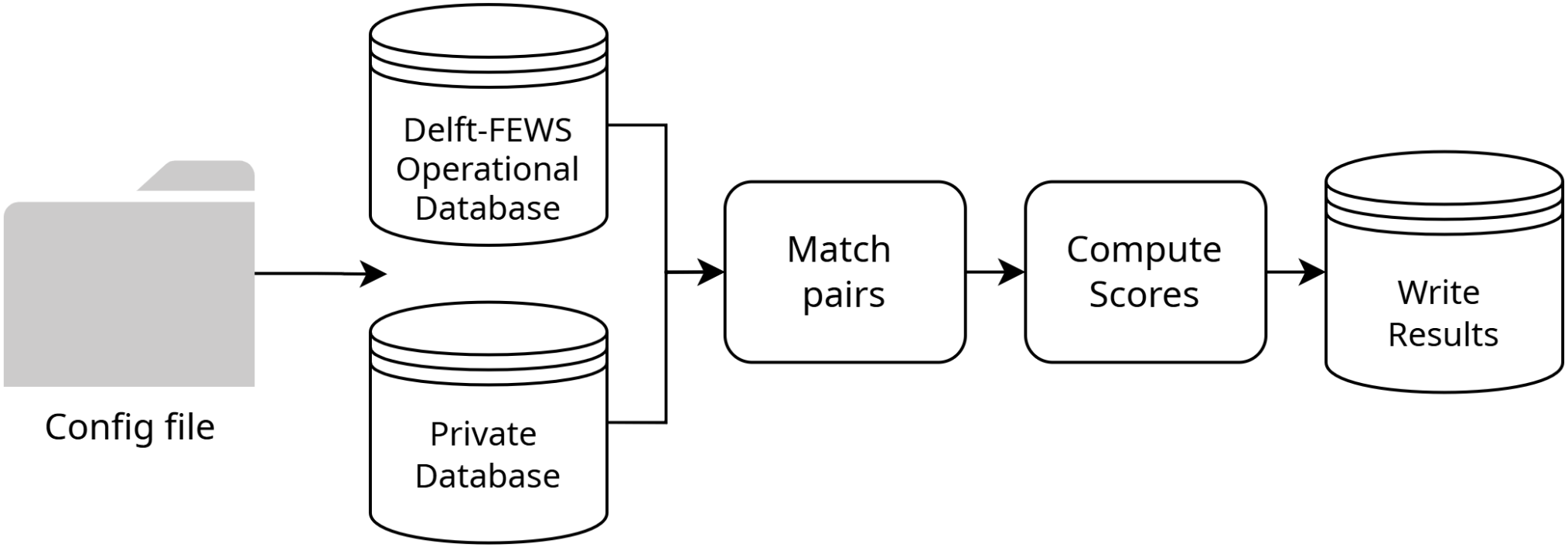
Has flexible outputs — results can be visualized

- In custom dashboards
- In Delft-FEWS (planned 2026)
- In any other format

Alright, so how does it work **exactly**?



Alright, so how does it work exactly?



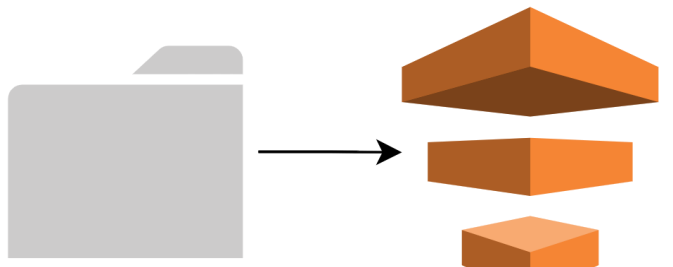
We define the verification pipeline in a simple config file

ConfigSchema (config.schema.json) | DaveCasson, 6 hours ago | 2 authors (You and one other)

```yaml
# yaml-language-server: $schema=https://deltares.github.io/veriflow/v0/config.schema.json
version: v0
general:
  verification_period:
    dimension: forecast_reference_time
    start: "2025-05-20T00:00:00Z"
    end: "2026-06-01T00:00:00Z"
  verification_pairs:
    - id: QR_GEPS
      obs: observed
      sim: simulated_GEPS
      variable: discharge
    - id: QR_GEFS
      obs: observed
      sim: simulated_GEFS
      variable: discharge
    - id: QR_IFS
      obs: observed
      sim: simulated_IFS
      variable: discharge
  lead_times:
    unit: D
    values: [1, 2, 3, 4, 5, 6]
datasources: ···
scores:
  - score_adapter: crps_for_ensemble
  - score_adapter: rank_histogram
    reduce_dims:
      - forecast_reference_time
```

In just 70 lines of YAML config,
We define

- What period to verify
- What models to verify
- What lead times to verify
- Where to get the data (Delft-FEWS)
- What scores to compute



Config file

Verification Pipeline

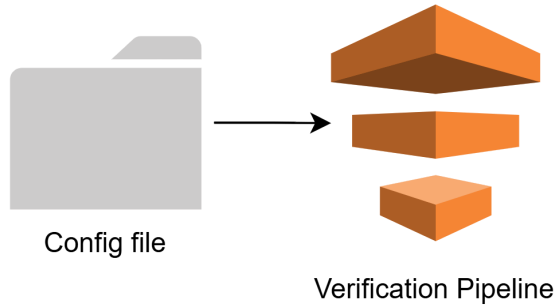
We run the *veriflow* pipeline in just one line of code

In just **one line of code**, we run the veriflow pipeline

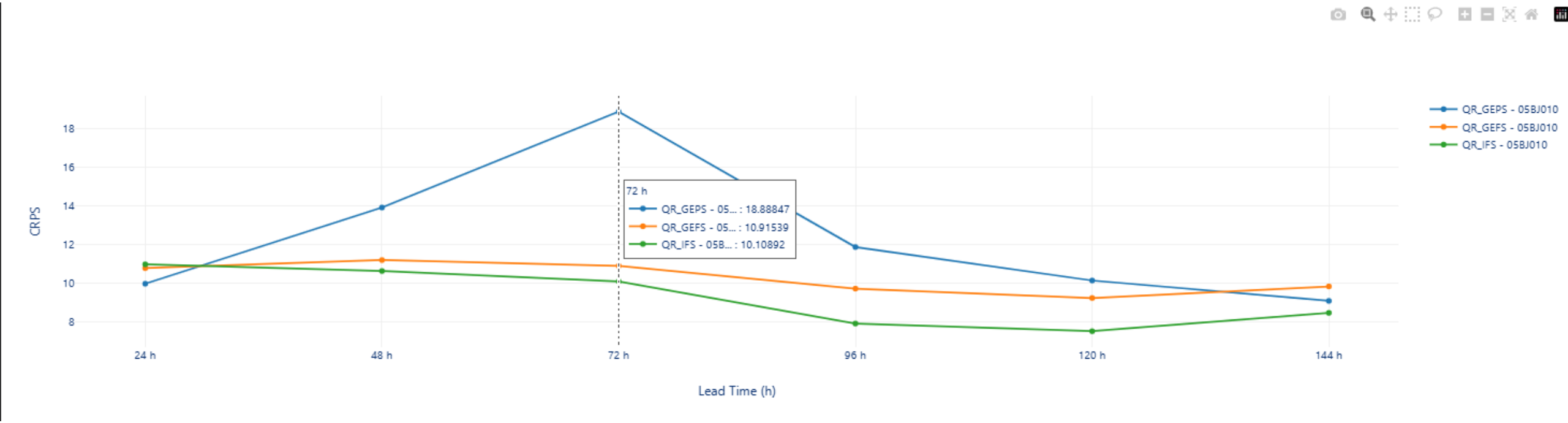
Running the *veriflow* pipeline

```
config_file_path = base_config / "Raven_Elbow_Discharge_Ensemble.yaml"
ods = run_pipeline(config=(config_file_path, "yaml"))
```

```
2026-06-12 16:38:53,599 - INFO - Successfully initialized the configuration.
        verification_period_start = 2025-05-20 00:00:00
        verification_period_end = 2026-06-01 00:00:00
2026-06-12 16:38:53,603 - INFO - Start getting data from FewsWebservice.
2026-06-12 16:38:54,109 - INFO - Successfully got data from FewsWebservice.
2026-06-12 16:38:54,110 - INFO - Start getting data from FewsWebservice.
2026-06-12 16:38:54,660 - INFO - Successfully got data from FewsWebservice.
2026-06-12 16:38:54,660 - INFO - Start getting data from FewsWebservice.
2026-06-12 16:38:55,203 - INFO - Successfully got data from FewsWebservice.
2026-06-12 16:38:55,203 - INFO - Start getting data from FewsWebservice.
2026-06-12 16:38:55,794 - INFO - Successfully got data from FewsWebservice.
2026-06-12 16:38:55,797 - INFO - Successfully loaded all data from sources.
2026-06-12 16:38:55,831 - INFO - Successfully computed CrpsForEnsemble for verification pair QR_GEPS.
2026-06-12 16:38:55,853 - INFO - Successfully computed CrpsForEnsemble for verification pair QR_GEFS.
2026-06-12 16:38:55,880 - INFO - Successfully computed CrpsForEnsemble for verification pair QR_IFS.
2026-06-12 16:38:55,898 - INFO - Successfully computed RankHistogram for verification pair QR_GEPS.
2026-06-12 16:38:55,916 - INFO - Successfully computed RankHistogram for verification pair QR_GEFS.
2026-06-12 16:38:55,935 - INFO - Successfully computed RankHistogram for verification pair QR_IFS.
2026-06-12 16:38:55,982 - INFO - Successfully wrote results of verification pair QR_GEPS to CFCompliantNetCDF.
2026-06-12 16:38:56,019 - INFO - Successfully wrote results of verification pair QR_GEFS to CFCompliantNetCDF.
2026-06-12 16:38:56,061 - INFO - Successfully wrote results of verification pair QR_IFS to CFCompliantNetCDF.
2026-06-12 16:38:56,062 - INFO - Verification pipeline completed successfully.
```



An example of CRPS results for the 3 NWP producst at various lead times



So *veriflow* can be easily applied to Delft-FEWS

Lowering the technical barrier for verification

Future

2026

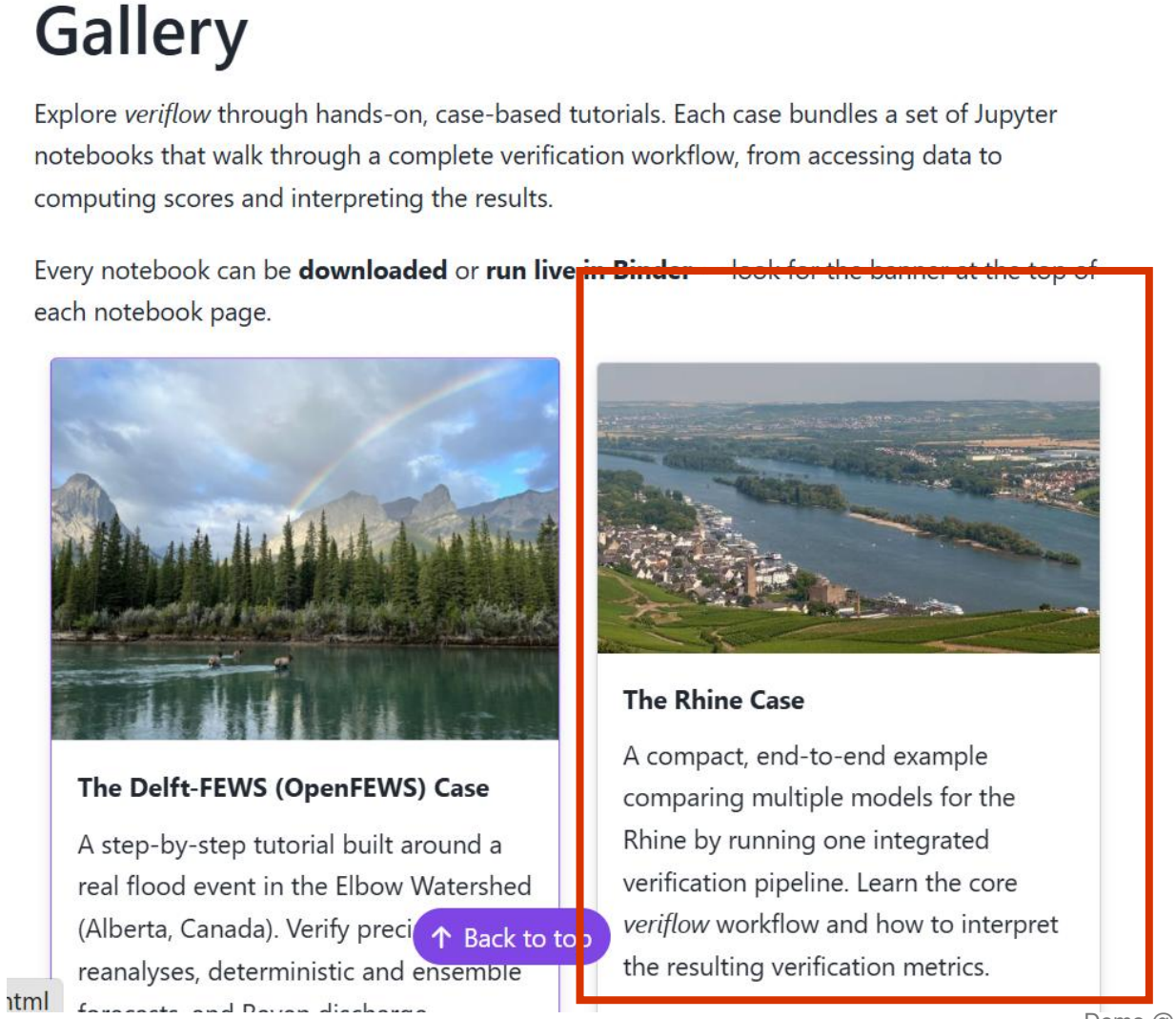
- Streamline filtering:
  a. In time
  b. Co-variables
  c. According to climatology
  d. Thresholds
- Skills scores
- More on detection: e.g. hit and false alarm rates
- Further integration in real-time FEWS

  2026: Design and testing, more development in 2027
- Dashboards stand alone and FEWS

4 Veriflow – trying it out

Exercises from demo gallery

Tutorial gallery: https://deltares.github.io/veriflow/gallery.html



Hindcast,
Deterministic
Ensemble
FEWS webservice

Raw and Postprocessed Precip Ensemble
ZARR archive

The Rhine case

A compact, end-to-end example that runs a single integrated verification pipeline to compare multiple models for the Rhine. Start with the **basics** to learn the core *veriflow* workflow, then move on to **interpretation** to make sense of the resulting verification metrics.

Each notebook page includes a banner to **download** it or **open it in Binder**.

1a · Basics

Configure a verification experiment, run the *veriflow* pipeline, inspect the returned `OutputDataset`, and explore results with the interactive app.

1b · Interpretation

Interpret the verification results: how forecast quality changes with lead time, how well the ensemble captures uncertainty, and the effect of post-processing.

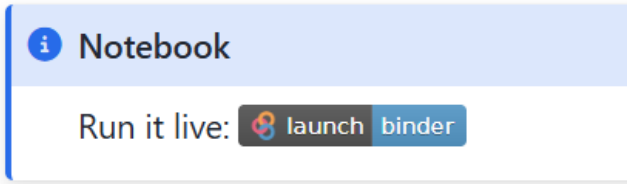
How will you do this?

View

Just look at website

See the notebook and results

Browser with Binder



Private session

Closed after inactive for 5-10min

Local installation

Info at: https://github.com/Deltares/veriflow

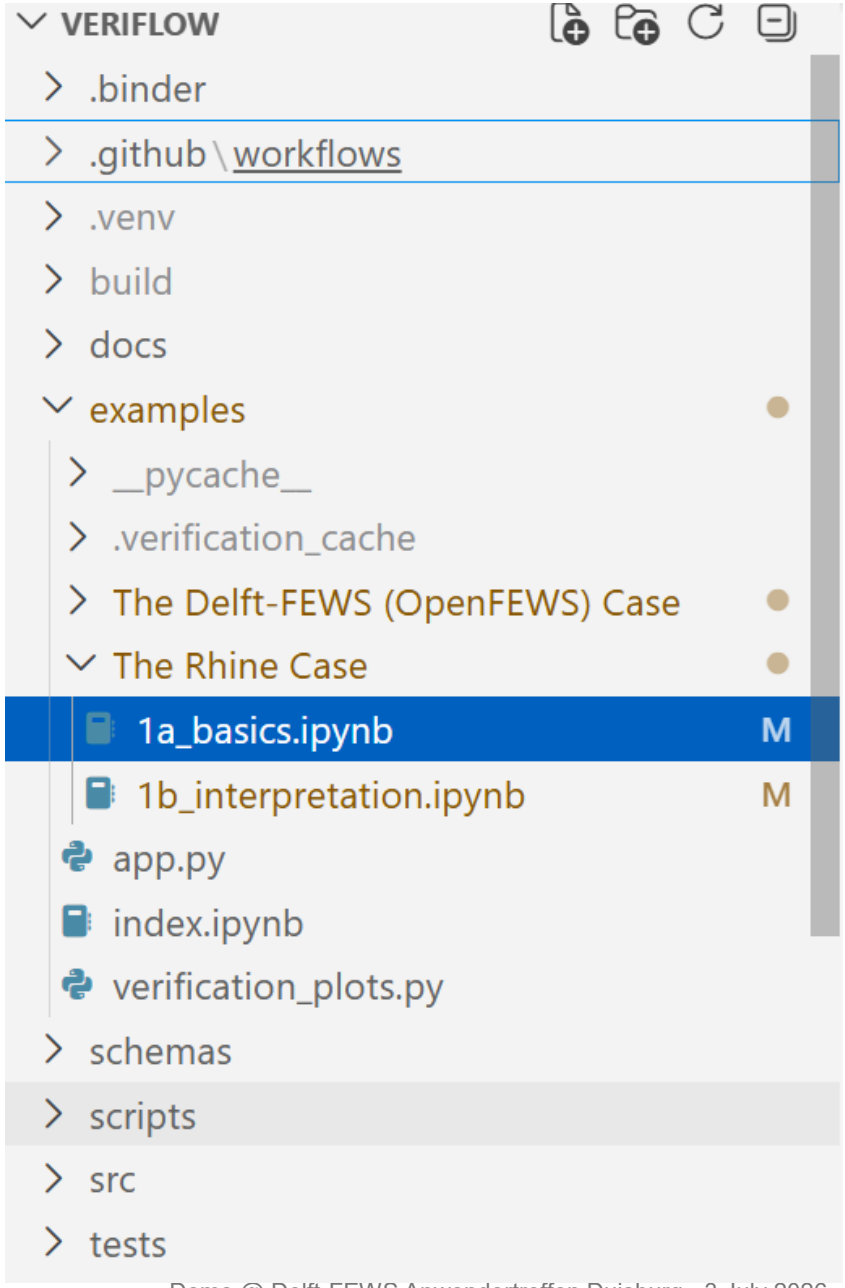
Python >=3.11

Install and download tutorial separately, or clone the repo.

The code

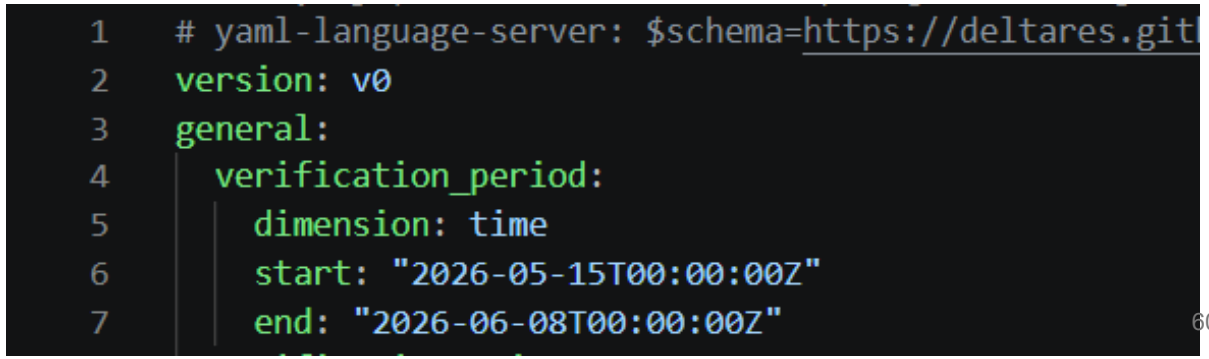
Rhine tutorial notebook uses matplotlib for plotting

```
pip install matplotlib
```



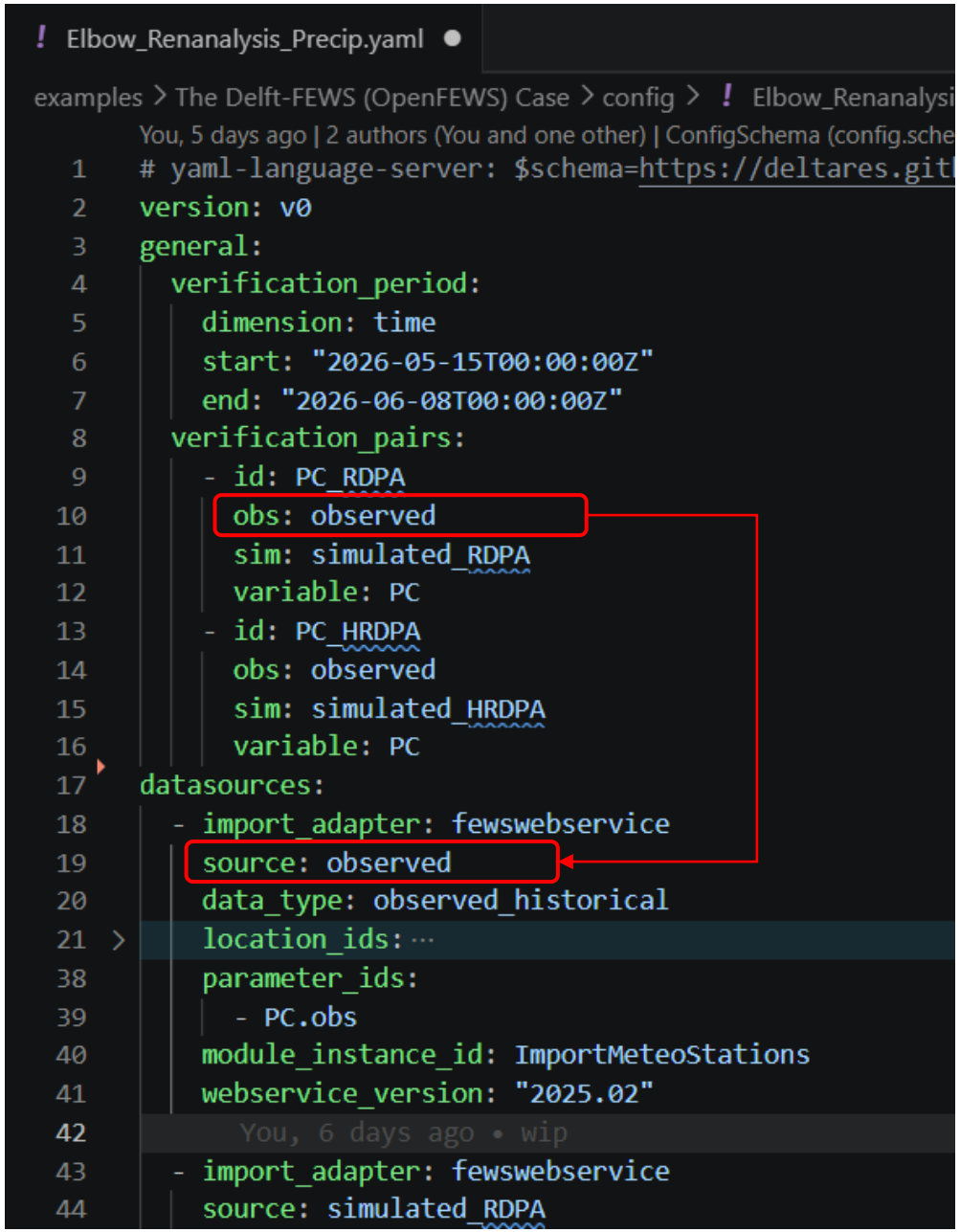
The verification period “dimension”

- The “dimension” in the verification_period allows you to define the dimension to use when defining your “verification period”
- Time:
  - define start and end along “time”
  - → used for historical simulations
  - → when used with forecasts, will mask all forecast values outside a given window
- Forecast_reference_time:
  - define start and end along “forecast_reference_time”
  - → standard for forecast verification: uses complete forecasts with a forecast_reference_time between start and stop



The verification pair

- The verification pair defines allows you to define what data to compare.
  - id: used as metadata in output
  - obs → reference your observed datasource by it's "source" property
  - sim → same concept as obs
  - variable: the variable to select from the datasource



Tips for new configurators

1

Use YAML language support to validate config against a schema

2

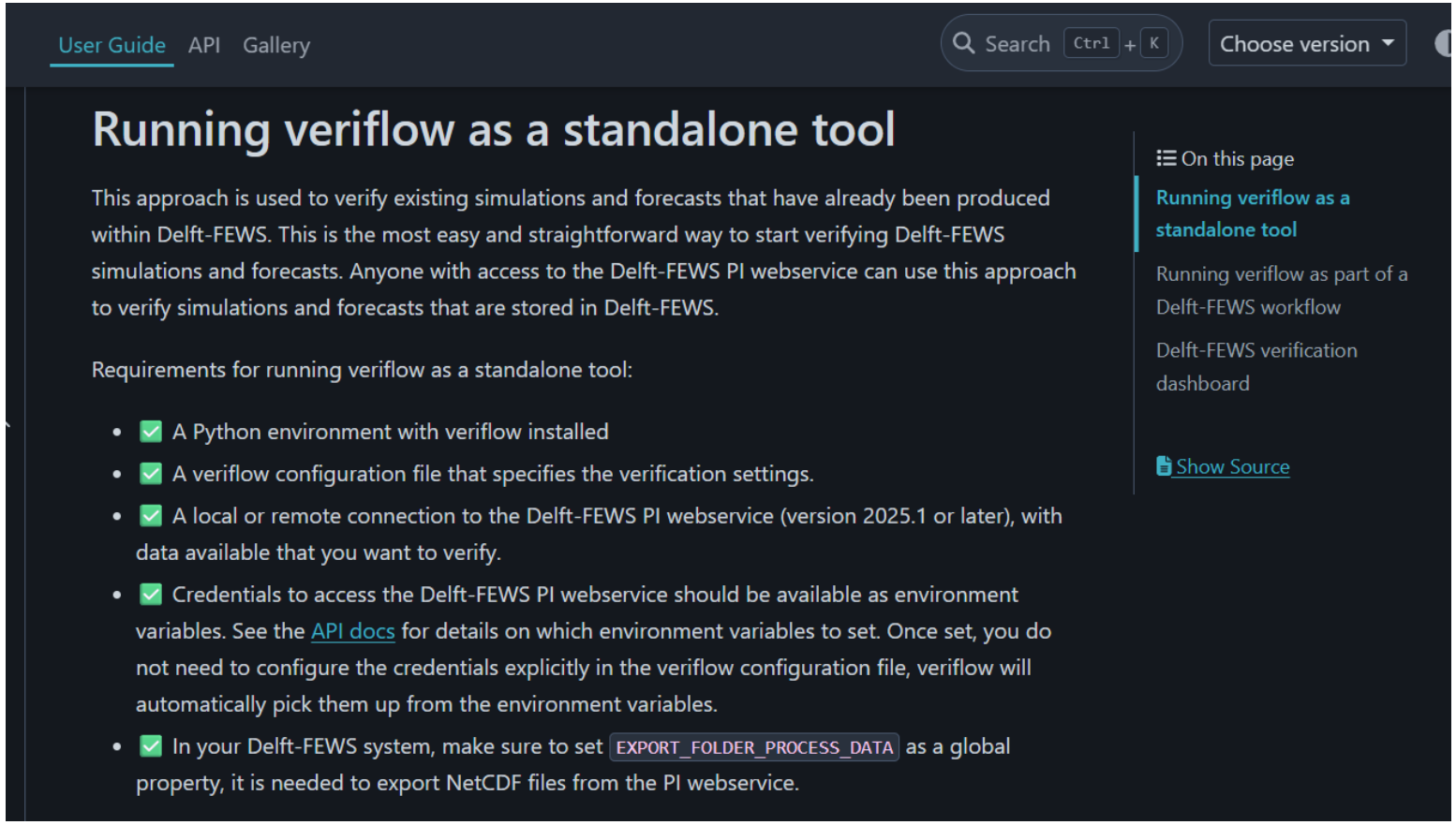
Use YAML language support to get useful hints on what is expected in certain fields

3

Use the documentation pages for concepts behind configuration

References

https://deltares.github.io/veriflow/user_guide/best_practices/delft_fews.html



Useful references

Software

Veriflow GitHub: https://github.com/Deltares/veriflow

Veriflow Documentation: https://deltares.github.io/veriflow/

Scores Documentation: https://scores.readthedocs.io/en/stable/

Papers

Murphy, Allan H. "What is a good forecast? An essay on the nature of goodness in weather forecasting." *Weather and forecasting* 8.2 (1993): 281-293.

Pagano, T.C., Ebert, E.E. and Khanarmuei, M. (2025), Enhancing Forecast Verification in National Meteorological and Hydrological Services. Meteorol Appl, 32: e70051. https://doi.org/10.1002/met.70051

Loveday, Nicholas, et al., 2024. “The Jive Verification System and its Transformative Impact on Weather Forecasting Operations”. Bull. Am. Met. Soc. Sept. 11. https://doi.org/10.1175/BAMS-D-23-0267.1

Books

Wilks, Daniel S. *Statistical methods in the atmospheric sciences*. Vol. 100. Academic press, 2011.

TIGGE workshop ECMWF (Beth Ebert, BoM); https://www.ecmwf.int/sites/default/files/elibrary/2005/15865-verification-ensembles.pdf

Guidelines on the Verification of Hydrological Forecasts (WMO); https://library.wmo.int/records/item/69478-guidelines-on-the-verification-of-hydrological-forecasts

References

Veriflow (Deltares); https://github.com/Deltares/veriflow

TIGGE workshop ECMWF (Beth Ebert, BoM); https://www.ecmwf.int/sites/default/files/elibrary/2005/15865-verification-ensembles.pdf

Guidelines on the Verification of Hydrological Forecasts (WMO); https://library.wmo.int/records/item/69478-guidelines-on-the-verification-of-hydrological-forecasts

5 Discussion

Veriflow… and forecast verification in general

Would you want to do more on forecast verification?

Is there a role for veriflow there?

How do you see the role of the Delft-FEWS suite?

Stay in touch

Maarten Smoorenburg

Maarten.Smoorenburg@deltares.nl

https://github.com/Deltares/veriflow

www.deltares.nl

info@deltares.nl

linkedin.com
/company
/deltares

@deltares

facebook.com
/deltaresNL