Plans for open-source packages to prepare rainfall data

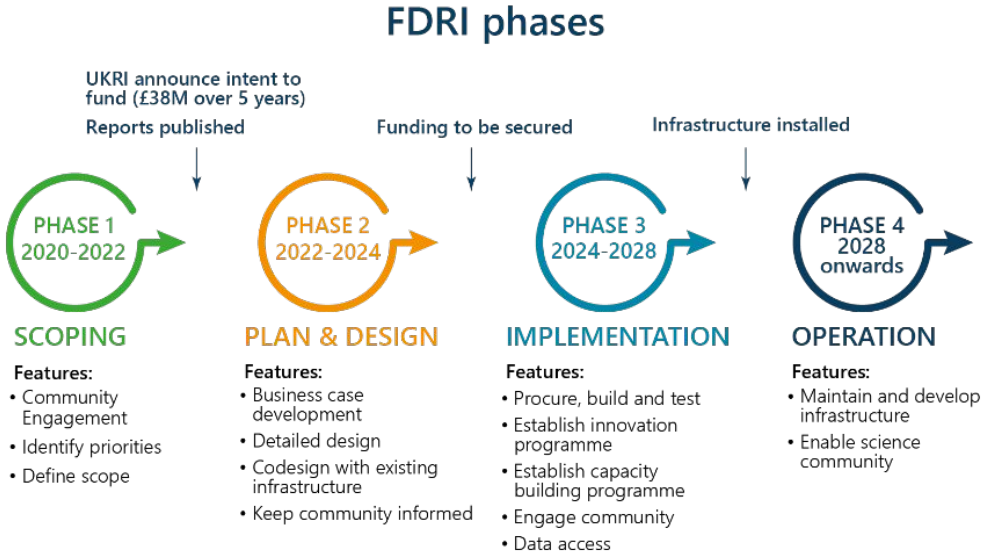
(including RainfallQC)

Dr. Tom Keel
Hydrological Data Analyst – UKCEH Wallingford

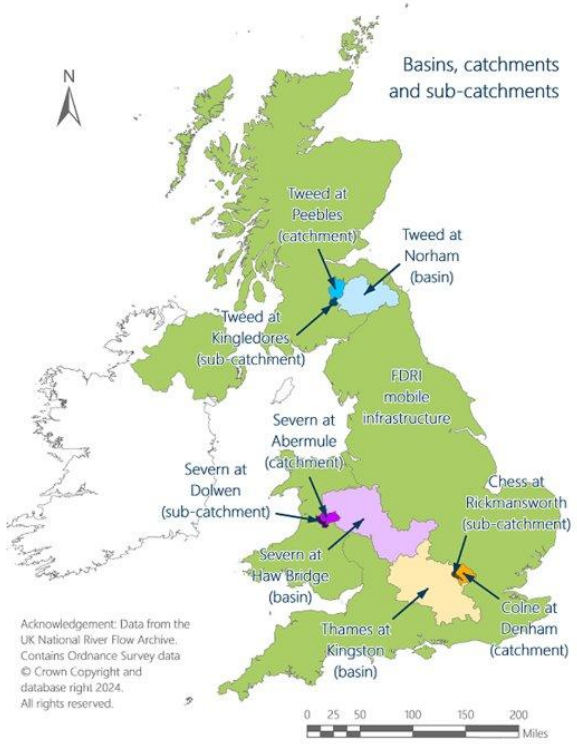
FDRI
UK Centre for Ecology & Hydrology
BGS British Geological Survey
IMPERIAL
University of BRISTOL
UKRI Natural Environment Research Council

Floods and Drought Research Infrastructure (FDRI)

- Multi-year NERC investment
- Based on new:
  - Intensive catchment monitoring
  - Digital infrastructure
- Supports:
  - Innovation in hydrological monitoring and data processing
  - Community and capacity building



FDRI plans



For catchment monitoring:

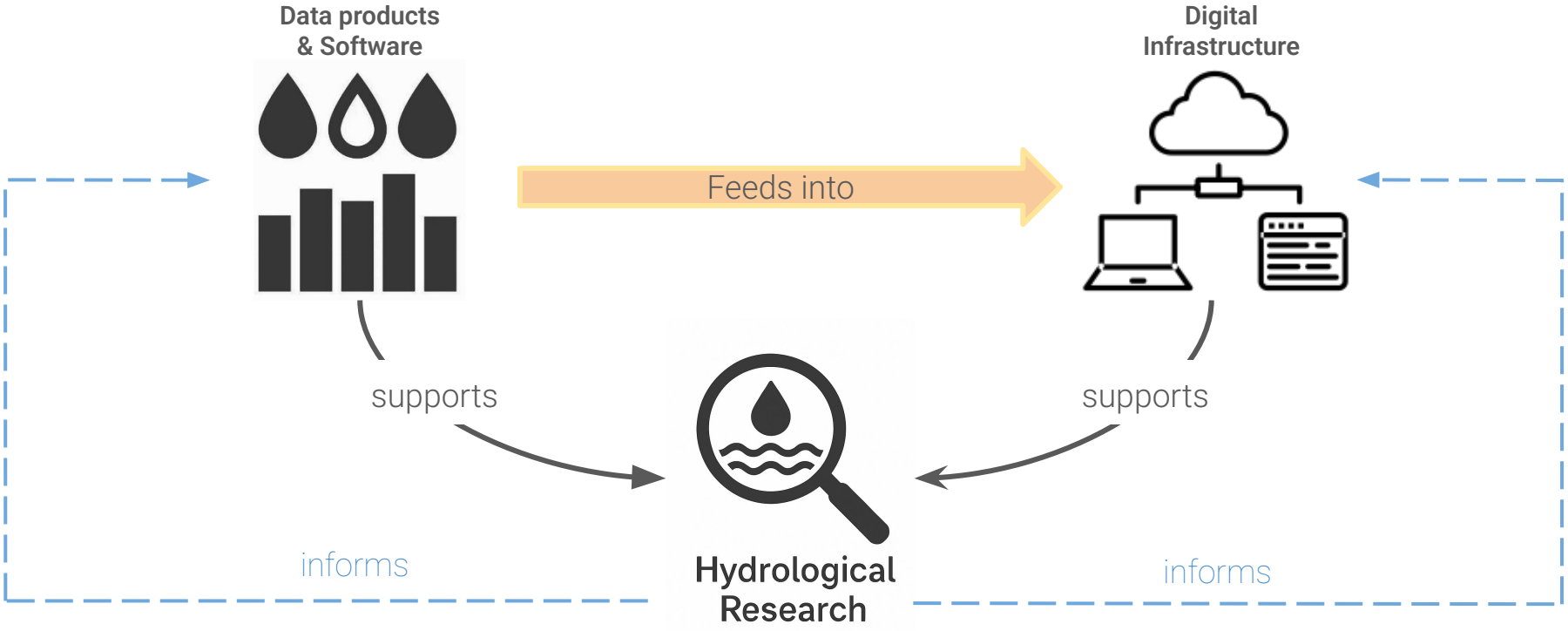
- Intensive real time monitoring of 3 catchments: **(1)** Upper Severn in Wales, **(2)** Upper Tweed in Scotland; **(3)** River Chess in England

For research infrastructure:

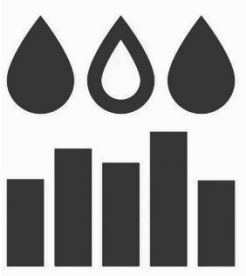
"FDRI will improve access to UK-wide data"

- UKFlow15 dataset of QC'd river flows for 1300+ gauging stations (Summer 2025)
- CAMELS-GB sub-daily (Autumn 2025)
- CEH-GEAR 15-min (Summer 2026)
- **New tools for data processing**

What is hydrological research infrastructure?



Plans for UK rainfall products within FDRI



Data & Software:

- Well maintained datasets e.g. *CEH-GEAR 15 min*
- Easy-to-use software tools for data processing for QC, flagging, gridding, metadata handling
- Workshops & teaching materials

Infrastructure:

- Processing pipelines for QC and flagging
- Object storage & Data catalogs serve as a place for up to date hydrological data
- Dashboards, Data portals & Web maps

I am working towards updating:

Sub-daily CEH-GEAR - A gridded rainfall product

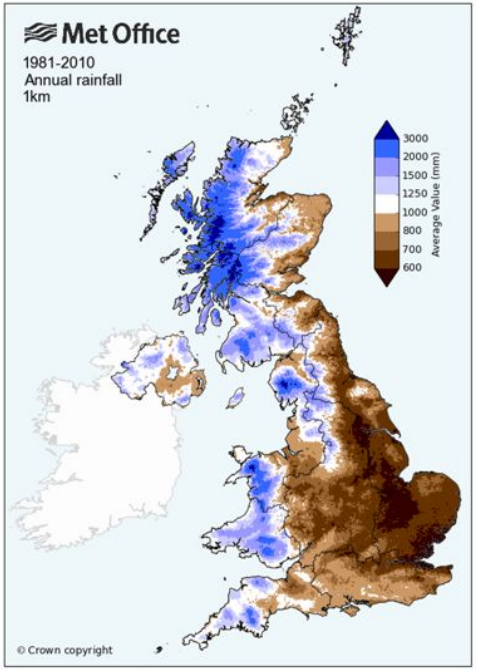
Lewis, E. et al *see all authors*

Gridded estimates of hourly areal rainfall for Great Britain 1990-2016 [CEH-GEAR1hr] v2

https://doi.org/10.5285/fc9423d6-3d54-467f-bb2b-fc7357a3941f

Download ↓ | Supporting docs | Cite this dataset ❝❞

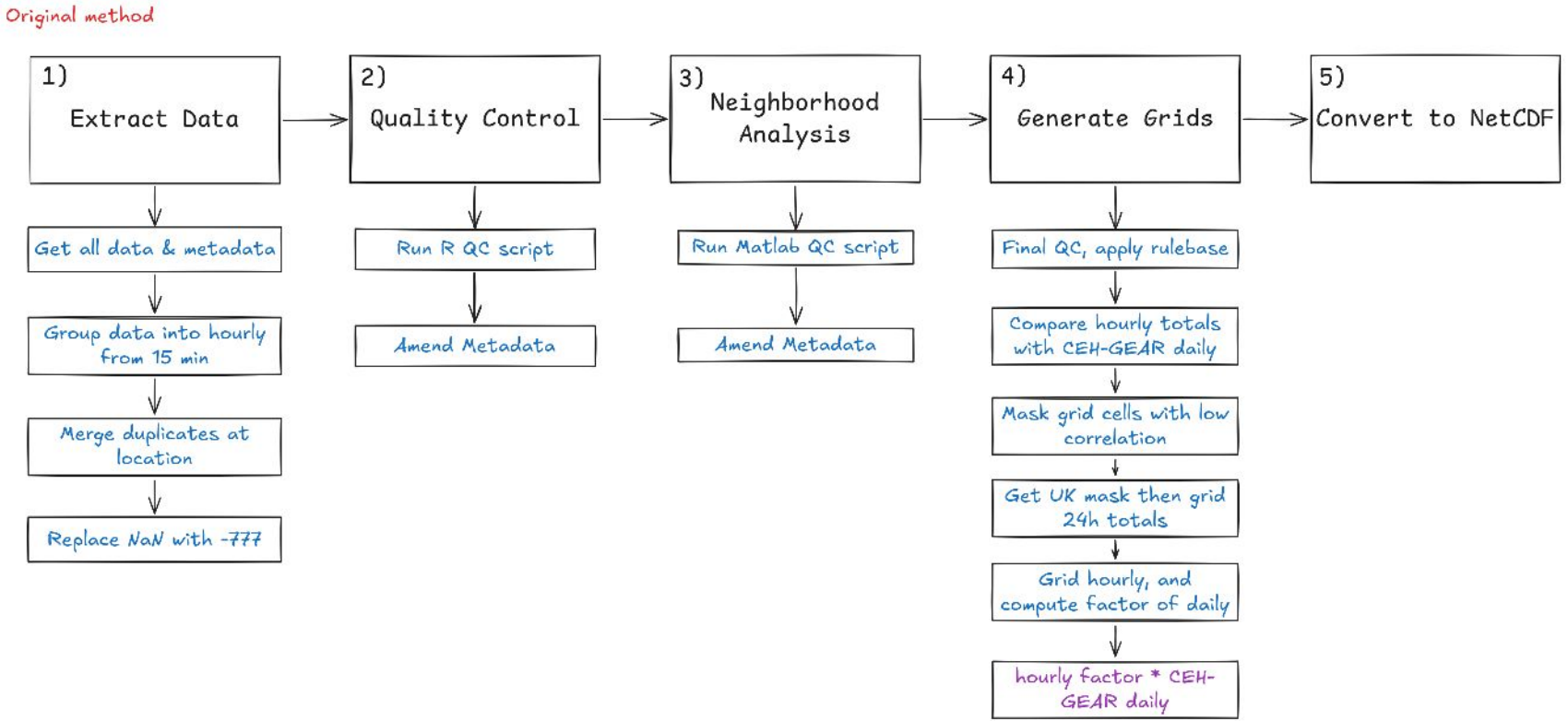
- 1 km grid resolution
- 15 mins & 1 hour products
- Compliments the HadUK-Grid daily product from the Met Office



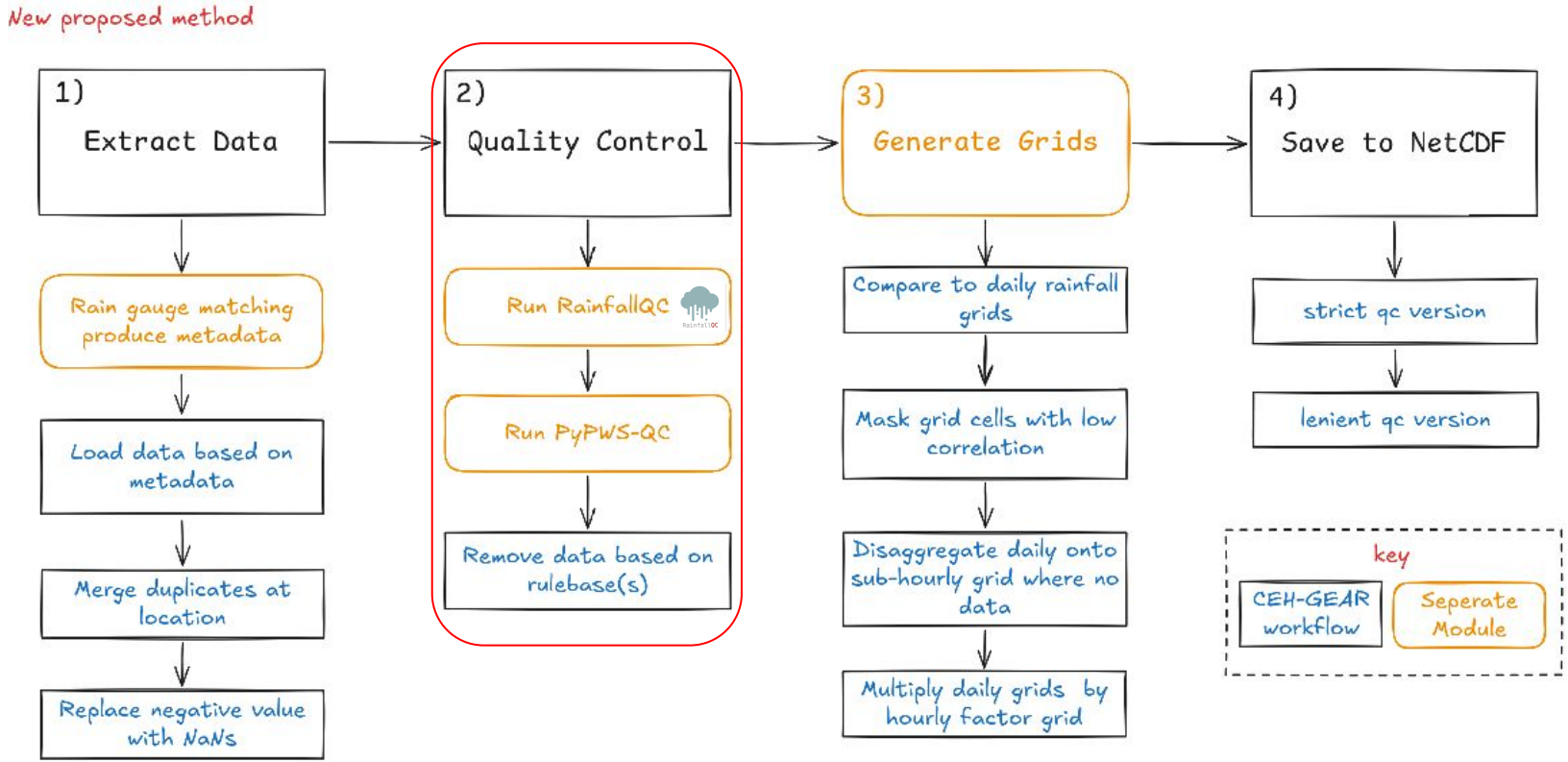
FDRI

HadUK-Grid: *Hollis et al. (2019),* https://doi.org/10.1002/gdj3.78

Existing CEH-GEAR 1h workflow



A new open-source workflow for CEH-GEAR 15min



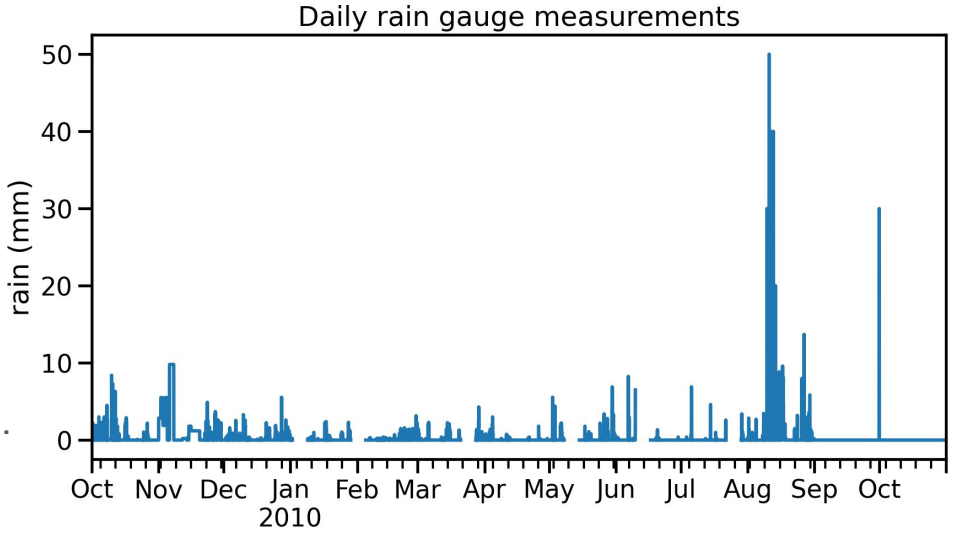
Lewis, E. et al (2022). **CEH-GEAR 1hr v2**. NERC EDS EIDC. https://doi.org/10.5285/fc9423d6-3d54-467f-bb2b-fc7357a3941f

Quality controlling rain gauge data

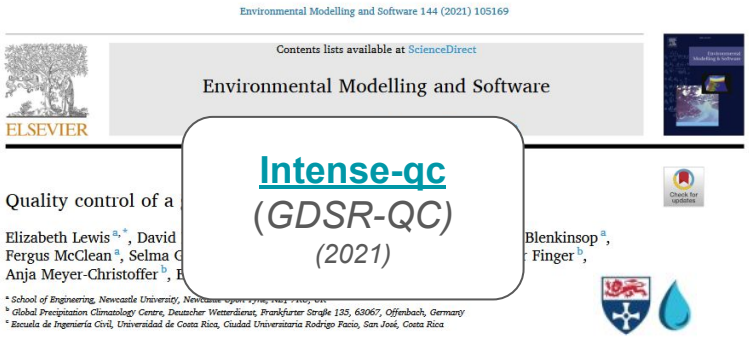
Various human and mechanical errors introduce artefacts into the rainfall record that may go undetected

Considerations:

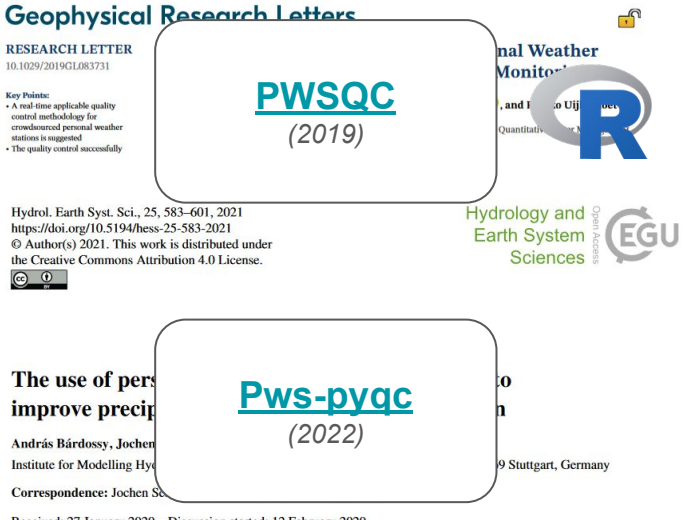
- QC is a retrospective process
- Whether to flag suspect data or remove it
- QC involves a **manual** and **automated** stage.



Automated QC Tools (3)



Intense-qc
(GDSR-QC)
(2021)



PWSQC
(2019)

Pws-pyqc
(2022)

- Provides modular QC checks
- Extended to sub-hourly by **SubHourlyQC**

- Provides filters and a QC framework
- Designed for whole networks of PWS and, ideally, a good reference dataset

Review of current QC protocols provided in: *El Hachem et al. (2024) DOI: 10.5194/hess-28-4715-2024*

Considerations for quality control tools

Possible issues with QC:

- Coupling to underlying data and thresholds
- Reliance on metadata and flagging protocols
- Technically complex with long compute times
- Tools are not maintained
- **Coupling to time windows/resolutions**

What is not covered by existing tools:

- What if we want to run a single check (a student might not want full QC)
- What defines streaks? What defines ‘extremes’
- Do we need the different checks for different temporal resolutions…

Introducing RainfallQC

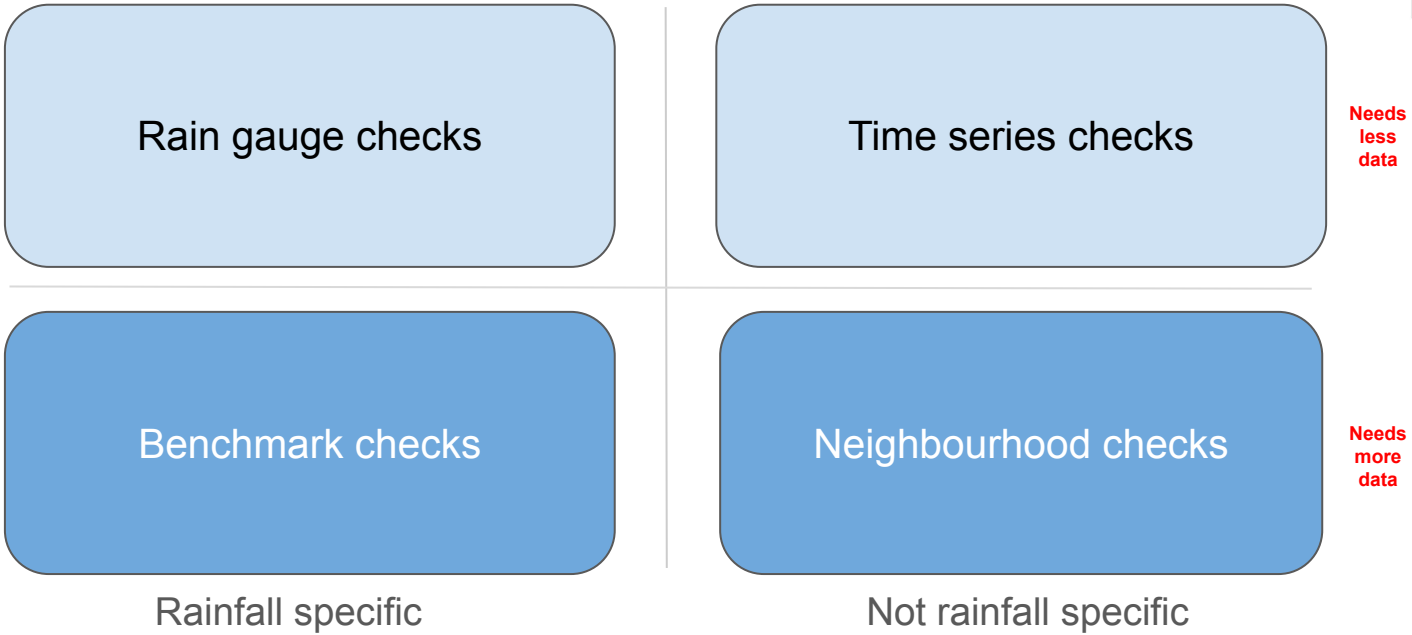
- An open-source Python package for customisable QC workflows
- Implements 25 QC checks from GSDR-QC (Lewis *et al.* 2021)
- Built on top of Polars

Installed with: *pip install rainfallqc*

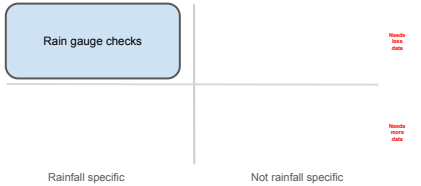
Link for ReadTheDocs

Source code: https://github.com/NERC-CEH/RainfallQC

QC checks in RainfallQC



Rain gauge checks



Detecting abnormalities in summary and descriptive statistics.

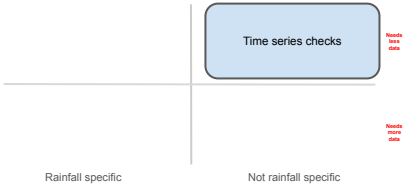
Examples:

- ➔ Resolution changes
- ➔ Temporal biases and drift
- ➔ Significant breakpoints
- ➔ Periods with low or zero values

Considerations:

- ❑ Requires metadata for context/setting
- ❑ Effective checking requires a sufficient historical record
- ❑ What if using data from different sources?

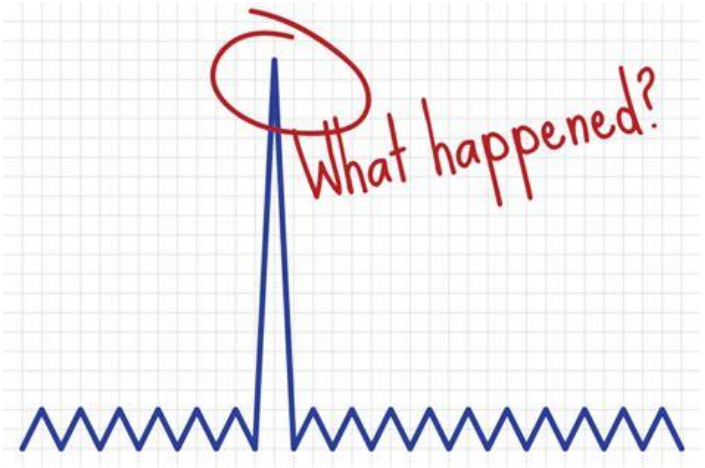
Time series checks



Detecting abnormalities in patterns of the data record.

Examples:

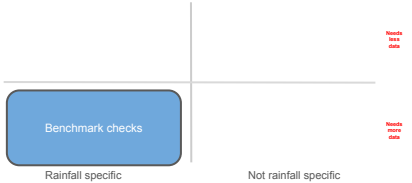
- ➔ Large spike checks
- ➔ Accumulations
- ➔ Streaks or other time series patterns



Considerations:

- ❑ Which thresholds to use?
- ❑ Is problem a human or mechanical one?

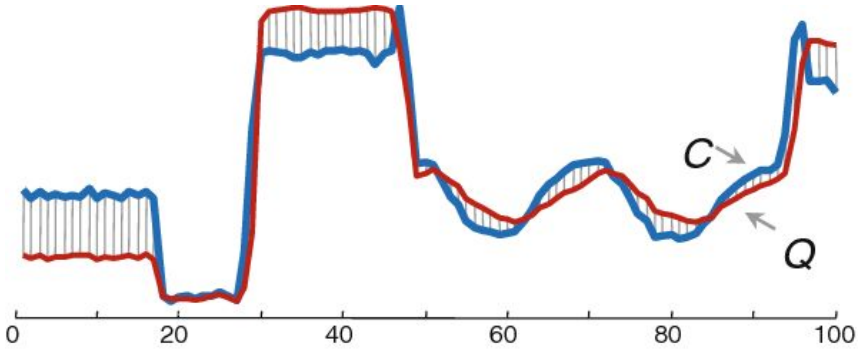
Benchmark checks



Detecting abnormalities based on benchmarks.

Examples:

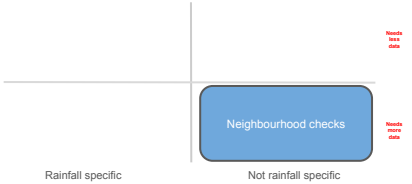
- ➔ World record rainfall
- ➔ Dry spell records for location
- ➔ Spatial and temporal offsets



Considerations:

- ❑ Needs reliable benchmark data
- ❑ Is location specific

Neighbourhood checks



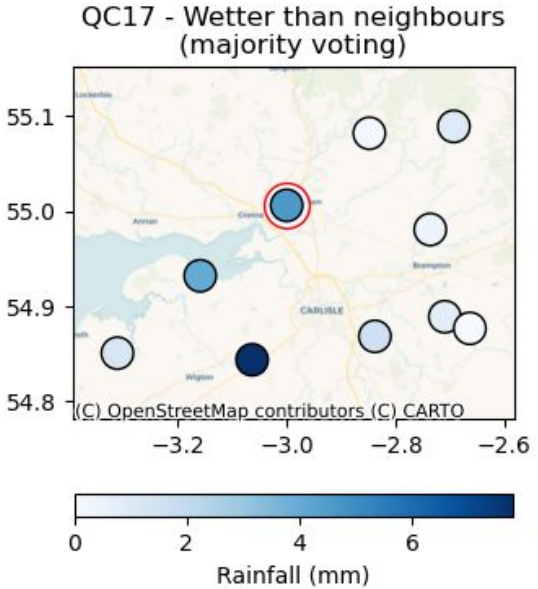
Detecting abnormalities based on measurements in neighbouring gauges.

Examples:

- ➔ Relatively extreme values vs neighbours
- ➔ Correlation & affinity indices
- ➔ Factor differences

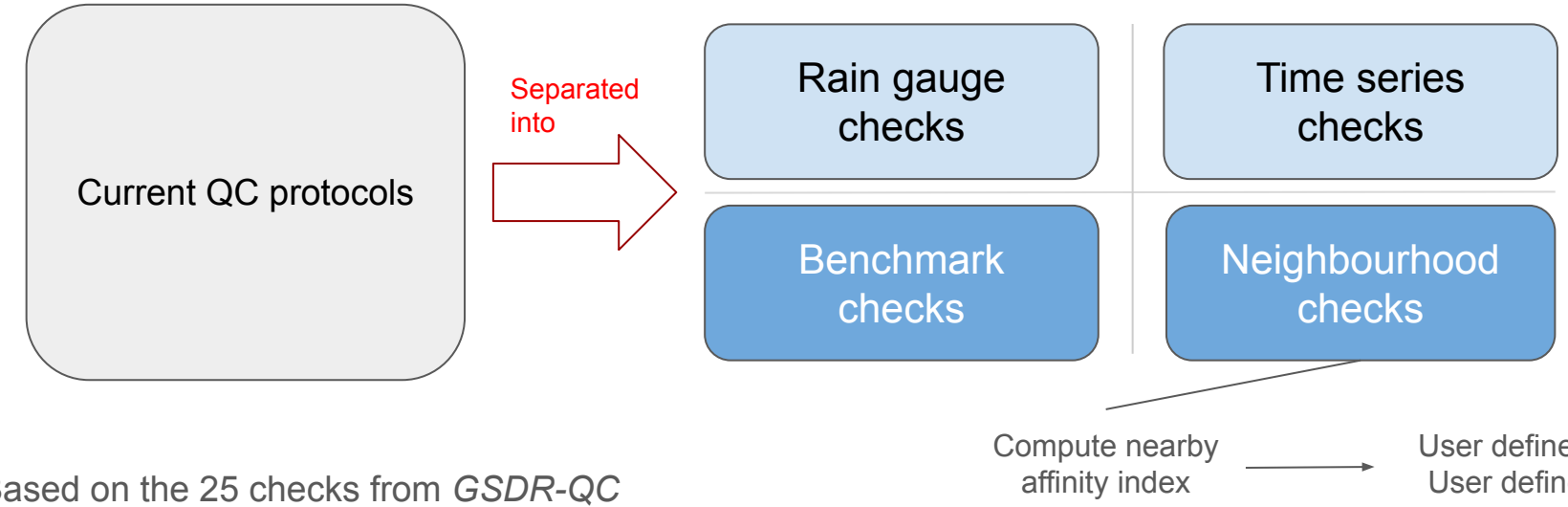
Considerations:

- ❑ Needs a reliable rain gauge network
- ❑ Which distances considered?
- ❑ Uses majority voting of neighbours



RainfallQC

Makes the QC procedure modular



Based on the 25 checks from *GSDR-QC*

GSDR-QC detailed in: *Lewis et al. (2021),* https://doi.org/10.1016/j.envsoft.2021.105169

RainfallQC v0.4.1

- Open-source Python package for running QC available via PyPI

```python
qc_to_run = ["QC1", "QC7", "QC14"]

qc_kwargs = {
    "QC1": {"quantile": 5},
    "QC7": {"expected_min_val": 0.1},
    "QC14": {"accumulation_multiplying_factor": 2.0}
}

results = apply_qc_framework.run_qc_framework(data,
                                              qc_framework='IntenseQC',
                                              qc_methods_to_run=qc_to_run,
                                              qc_kwargs=qc_kwargs)
```

Link for ReadTheDocs
[work in progres]

Designing your own QC framework

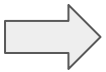
RainfallQC

```python
from rainfallqc import gauge_checks
from rainfallqc.qc_frameworks import apply_qc_framework

# Define your custom framework
custom_framework = {
    "min_val_01": {
        "function": gauge_checks.check_min_val_change,
    },
    "min_val_02": {
        "function": gauge_checks.check_min_val_change,
    },
    "user_qc_check": {
        "function": user_defined_qc_check
    },
}

qc_methods_to_run = ["min_val_01", "min_val_02", "user_qc_check"]

qc_kwargs = {
    "min_val_01": {"expected_min_val": 0.1},
    "min_val_02": {"expected_min_val": 0.2},
    "shared": {"target_gauge_col": f"rain_gauge_1"},
}
```



```python
import polars as pl

network_data = pl.read_csv("hourly_rain_gauge_network.csv")

# Run custom framework
result = apply_qc_framework.run_qc_framework(
    network_data,
    qc_framework="custom", ## Set the user defined QC framework
    qc_methods_to_run=qc_methods_to_run,
    qc_kwargs=qc_kwargs,
    user_defined_framework=custom_framework,
)
```

Designed for sensitivity analysis

Next steps

- Prepare initial version of CEH-GEAR 15min (ETA: before EGU26)
- Write docs for RainfallGridder and RainGaugeMatcha repos, making them open-source tools

FDRI is looking for community feedback

Getting involved with FDRI:

- Follow and find FDRI's projects @ https://github.com/NERC-CEH
- Sign up for FDRI newsletter

Sign up for the FDRI newsletter