Using R in Hydrology: recent developments and future directions

Louise J. Slater[a], Guillaume Thirel[b], Shaun Harrigan[c],
Olivier Delaigue[b], Alexander Hurley[d],
Abdou Khouakhi[e], Ilaria Prodoscimi[f],
Claudia Vitolo[c], and Katie Smith[g]
+ Paul Astagneau[b]

[a] School of Geography and the Environment, University of Oxford, OX1 3QY, UK
[b] IRSTEA, HYCAR Research Unit, 1 rue Pierre-Gilles de Gennes, 92160 Antony, France
[c] Forecast Department, European Centre for Medium-Range Weather Forecasts, Shinfield Park, Reading, RG2 9AX, UK
[d] School of Geography, Earth and Environmental Sciences, University of Birmingham, B15 2TT, UK
[e] School of Architecture, Civil and Building Engineering, Loughborough University, Loughborough, UK
[f] Department of Environmental Sciences, Informatics and Statistics, Ca' Foscari University of Venice, Venice, Italy
[g] Centre for Ecology & Hydrology, Maclean Building, Crowmarsh Gifford, Wallingford, OX10 8BB, UK

Many thanks to

Louise Slater

University of Oxford

Paul Astagneau

Irstea

The recent growth of R in hydrology

Driven by:

- The increase in **publicly-accessible data** (river flow, gridded reanalysis products, remote sensing data, catchment attributes)
- The increase in **computational power** and uptake of programming languages to explore and derive process insights from large and complex datasets
- An active and supportive **community of users**

Why use R in hydrology?

An open-source, multiplatform, versatile language with a wide range of uses:

- Data acquisition
- Data analysis, modelling and statistics
- Visualization
- Geospatial and GIS applications
- Generation of reports and presentations
- Interactive dashboards and web applications
- A useful teaching tool

Using R in Hydrology, *HESSD*, doi: 10.5194/hess-2019-50

R as a driver of hydrology

- More than 14,000 packages released in 25 years in many disciplines
- R combines cutting-edge technologies and tools and enables seamless integration of all the steps in a hydrological workflow

The recent (2019) CRAN Task View for Hydrological Data and Modeling (Zipper et al.)

CRAN Task View: Hydrological Data and Modeling

Maintainer:	Sam Zipper, Sam Albers, Ilaria Prosdocimi
Contact:	samuelczipper at gmail.com
Version:	2019-03-21
URL:	https://CRAN.R-project.org/view=Hydrology

This Task View contains information about packages broadly relevant to *hydrology* , defined as the movement, distribution and quality of water and water resources over a broad spatial scale of landscapes. Packages are broadly grouped according to their function; however, many have functionality that spans multiple categories. We also highlight other, existing resources that have related functions - for example, statistical analysis or spatial data processing. See also Riccardo Rigon's excellent list of hydrology-related R tools and resources.

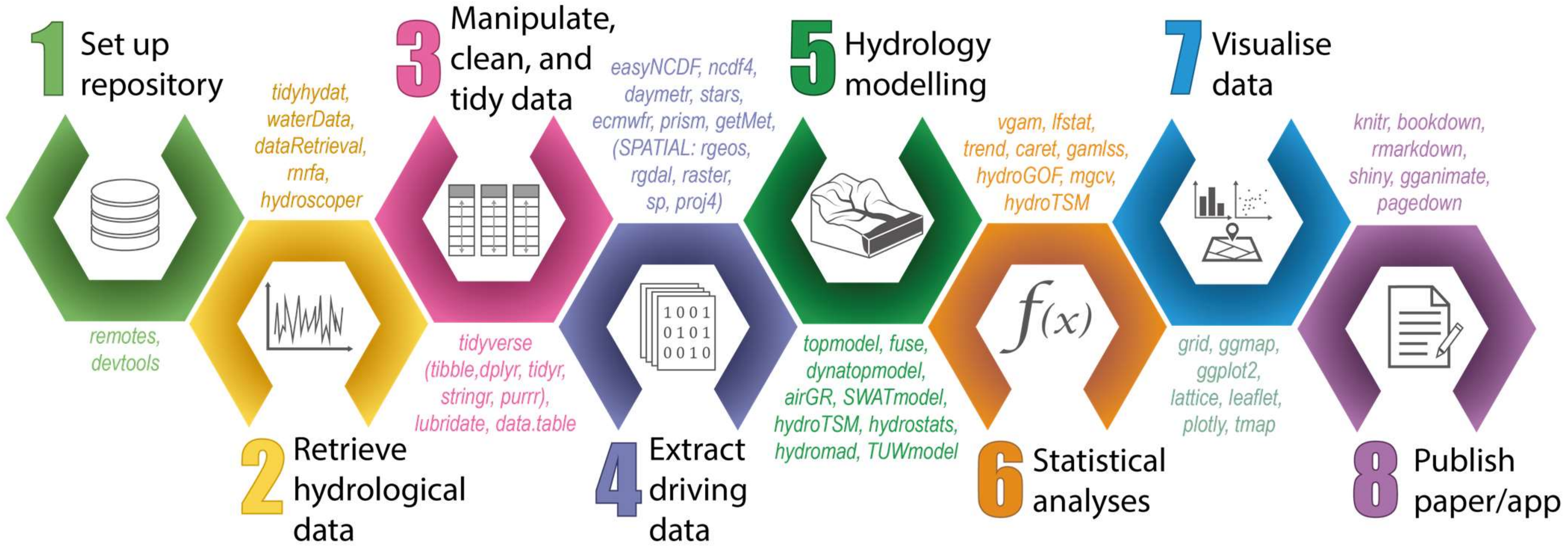
If you have any comments or suggestions for additions or improvements for this Task View, go to GitHub and submit an issue , or make some changes and submit a pull request . If you can't contribute on GitHub, send Sam Zipper an email . If you have an issue with one of the packages discussed below, please contact the maintainer of that package.

Data Retrieval

Hydrological data sources (surface water/groundwater quantity and quality)

- dataRetrieval: Collection of functions to help retrieve U.S. Geological Survey (USGS) and U.S. Environmental Protection Agency (EPA) water quality and hydrology data from web services.
- dbhydroR: Client for programmatic access to the South Florida Water Management District's DBHYDRO database , with functions for accessing hydrologic and water quality data.
- hddtools: Hydrological Data Discovery Tools. Facilitates discovery and handling of hydrological data, access to catalogues and databases.
- hydrolinks: Tools to link geographic data with hydrologic network, including lakes, streams and rivers. Includes automated download of U.S. National Hydrography Network and other hydrolayers.

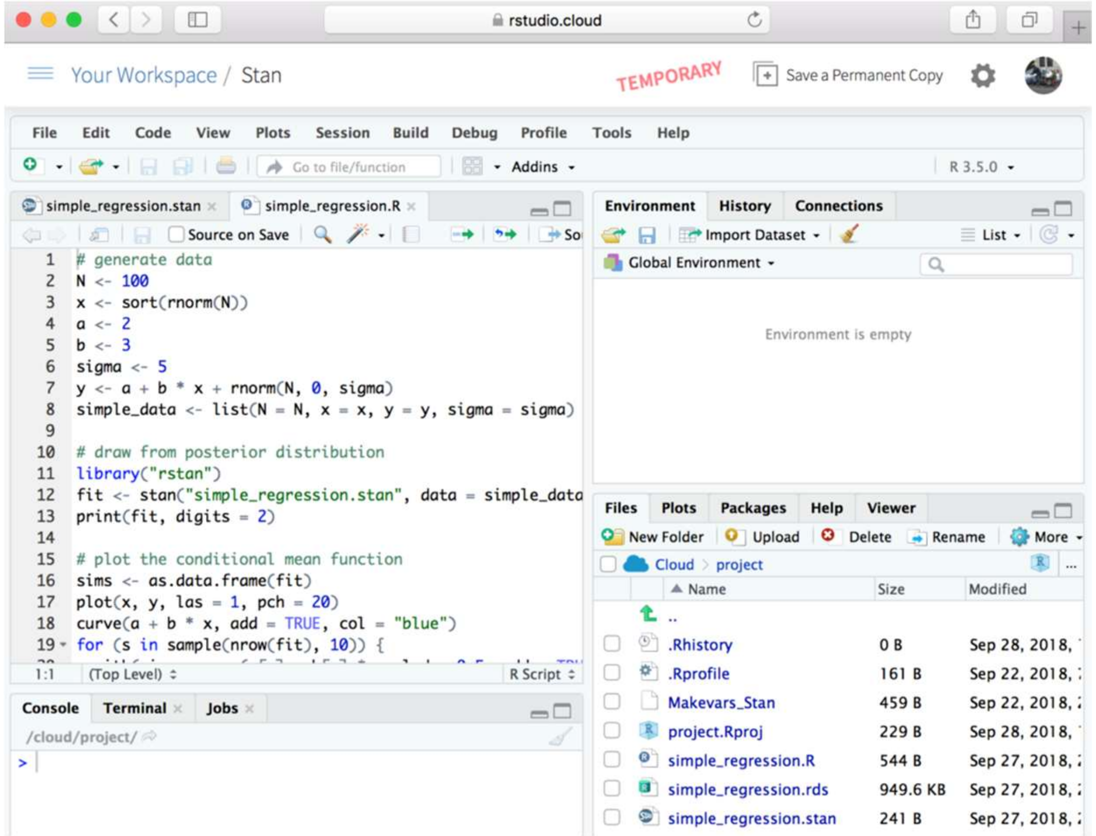
Using R in Hydrology



Advantages | The Hydro Workflow | Challenges | A Future Roadmap | Hydro modelling | Conclusions

Democratizing data science

- **Ease of use** (open source, interpreted language)
- **Documentation and support** (StackOverflow, Rseek.com)
- **Packages** for most hydrological tasks
- **Scripts** for incremental improvements and reuse/sharing
- **Hosted instances** of R (e.g. RStudio Cloud)



Rstudio cloud

Reproducible research

- **CRAN**: traceability by archiving past analyses and via **ORCID** numbers
- Standardized code format: CRAN, GitHub
- **rOpenSci** tools and code peer-review
- Publishing code via tutorials, **vignettes**
- All packages centralized on CRAN contain help pages

`dataRetrieval: Retrieval Functions for USGS and EPA Hydrologic and Water Quality Data`

Collection of functions to help retrieve U.S. Geological Survey (USGS) and U.S. Environmental Protection Agency (EPA) water quality and hydrology data from web services. USGS web services are discovered from National Water Information System (NWIS) <https://waterservices.usgs.gov/> and <https://waterdata.usgs.gov/nwis>. Both EPA and USGS water quality data are obtained from the Water Quality Portal <https://www.waterqualitydata.us/>.

Version:	2.7.5
Depends:	R (≥ 3.0)
Imports:	httr (≥ 1.0.0), curl, lubridate (≥ 1.5.0), stats, utils, xml2, readr (≥ 1.0.0), jsonlite
Suggests:	htmlTable, knitr, testthat
Published:	2019-06-05
Author:	Laura DeCicco [aut, cre], Robert Hirsch [aut], David Lorenz [aut], Jordan Read [ctb], Jordan Walker [ctb], Lindsay Carr [ctb], David Watkins [aut]
Maintainer:	Laura DeCicco <ldecicco at usgs.gov>
BugReports:	https://github.com/USGS-R/dataRetrieval/issues
License:	CC0
Copyright:	This software is in the public domain because it contains materials that originally came from the United States Geological Survey, an agency of the United States Department of Interior.
URL:	https://pubs.usgs.gov/tm/04/a10/
NeedsCompilation:	no
Citation:	dataRetrieval citation info
Materials:	NEWS
In views:	Hydrology
CRAN checks:	dataRetrieval results

`Downloads:`

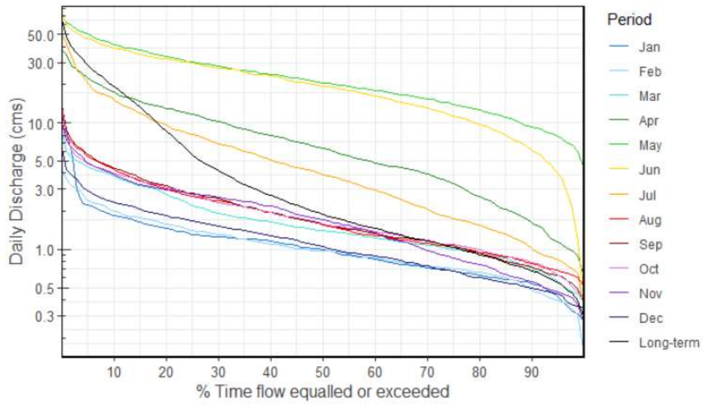
Reference manual:	dataRetrieval.pdf
Vignettes:	Introduction to the dataRetrieval package
Package source:	dataRetrieval_2.7.5.tar.gz
Windows binaries:	r-devel: dataRetrieval_2.7.5.zip, r-release: dataRetrieval_2.7.5.zip, r-oldrel: dataRetrieval_2.7.5.zip
OS X binaries:	r-release: dataRetrieval_2.7.5.tgz, r-oldrel: dataRetrieval_2.7.5.tgz

Statistical tools

- A language **designed** for statistical computing
- R is “GNU S”, a modern implementation of the S language (Bell Labs)
- **Base-R**: built-in basic functions
- Advanced statistical **packages**
- Many techniques for hydrology including linear and nonlinear modelling, statistical tests, time series analysis, classification or clustering

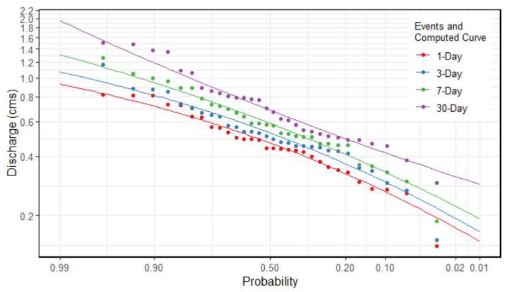
Flow duration curves can be produced using the `plot_flow_duration()` function.

```
plot_flow_duration(station_number = "08NM116",
                   start_year = 1981,
                   end_year = 2010)
#> $Flow_Duration
```



The probabilty of observed extreme events can also be plotted (using selected plotting position) along with the computed quantiles curve for comparison.

```
freq_results <- compute_annual_frequencies(station_number = "08NM116",
                                          start_year = 1981,
                                          end_year = 2010,
                                          roll_days = c(1,3,7,30))
freq_results$Freq_Plot
```



Source: Examples from the `fasttr` vignette

Connecting to other languages

- Many different languages are used in hydrology: interconnections can be very useful for taking advantage of each language's strengths or pre-existing scripts.
- **FORTRAN** and **C**: natively
- **C++**: Rcpp, Rcpp11, RcppArmadillo, RcppEigen
- **Python**: reticulate, rPython, rJython, XPython
- **Javascript**: V8
- **Matlab**: R.matlab
- **Julia**: JuliaCall, XRJulia

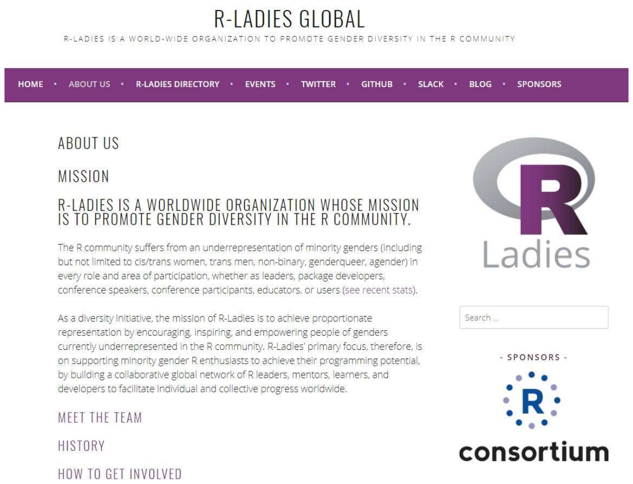
Source: code from the `airGR` package

```
##Call_fortran
  RESULTS <- .Fortran("frun_GR4J",PACKAGE="airGR",
                 ##inputs
                     LInputs=LInputSeries,                           ### length of input and outp
                     InputsPrecip=InputsModel$Precip[IndPeriod1],      ### input series of total pr
                     InputsPE=InputsModel$PotEvap[IndPeriod1],         ### input series potential e
                     NParam=as.integer(length(Param)),                ### number of model paramete
                     Param=Param,                                      ### parameter set
                     NStates=as.integer(length(RunOptions$IniStates)), ### number of state variable
                     StateStart=RunOptions$IniStates,                  ### state variables used whe
                     NOutputs=as.integer(length(IndOutputs)),          ### number of output series
                     IndOutputs=IndOutputs,                            ### indices of output series
                 ##outputs
                     Outputs=matrix(as.double(-999.999),nrow=LInputSeries,ncol=length(IndOutputs)),
                     StateEnd=rep(as.double(-999.999),length(RunOptions$IniStates))
                     )
  RESULTS$Outputs[ round(RESULTS$Outputs ,3)==(-999.999)] <- NA;
  RESULTS$StateEnd[round(RESULTS$StateEnd,3)==(-999.999)] <- NA;
```

- Connecting *other languages to R* is also possible: rpy2, pyRserve (Python), RCall (Julia)

Community interaction

- An extensive **user community** provides ample support to newcomers
- **Social media**: Twitter #rstats, Facebook ‘Hydrology in R’ group
- **StackOverflow** discussion forum
- **Mailing lists** for R usage or development (www.r-project.org/mail.html)
- **rOpenSci** non-profit community reviews scientific R packages
- **Short courses**: Using R in Hydrology (with the Young Hydrologic Society); see https://github.com/hydrosoc for EGU shared resources
- **R Consortium** supports users, maintainers and developers (e.g. grant program funds projects)
- Conferences: useR!, Rstudio, satuRdays, eRum
- Coding Help Desks at scientific conferences



Set up online repository so that versioned code can be uploaded repeatedly (version control)

The CRAN Hydrology Task View

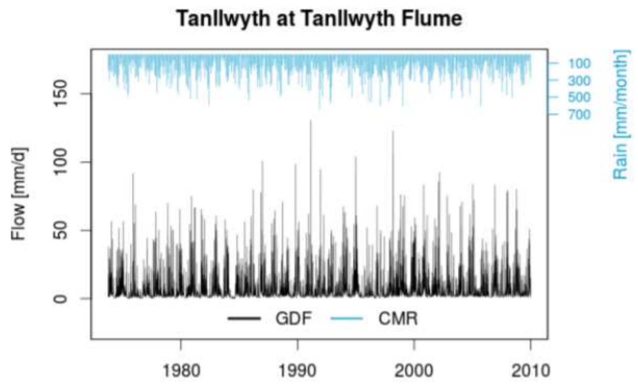
(maintainers: S.Zipper, S.Albers, I.Prosdocimi)

Published in January 2019 https://cran.r-project.org/web/views/Hydrology.html; lists over 100 hydrological packages in 3 categories:

- **Data retrieval**: **hydrological data** (surface or groundwater, both quantity and quality); **meteorological data** (e.g. precipitation, radiation, temperature)
- **Data analysis**: **data tidying** (e.g. gap-filling, data organization, quality control); **hydrograph analysis** (e.g. flow statistics, trends, biological indices); **spatial data processing**
- **Modelling**: **process-based** (scripts for preparing inputs/outputs and running process-based models); **statistical modelling** (hydrology-related statistical models)

Package	Description
`dataRetrieval`	Retrieve USGS and EPA hydrologic and water quality data (Hirsch and De Cicco, 2015)
`daymetr`	Interface to the Daymet web services: NASA daily surface weather and climatological summaries over North America, Hawaii, Puerto Rico (Hufkens et al., 2018)
`ecmwfr`	Interface to the public ECMWF API web services (Hufkens, 2018)
`getMet`	Get meteorological data for hydrologic models (Sommerlot et al., 2016)
`hddtools`	Hydrological data discovery tools (Vitolo, 2017)
`hydroscoper`	Interface to the Greek national data bank for hydrometeorological information (Vantas, 2018)
`prism`	Access the Oregon State Prism climate data using the web service API data (Hart and Bell, 2015)
`rnoaa`	Interface to NOAA weather data (Chamberlain, 2019)
`rnrfa`	Retrieve, filter and visualize data from the UK National River Flow Archive (Vitolo et al., 2016a)
`tidyhydat`	Extract and tidy Canadian hydrometric data (Albers, 2017)
`waterData`	Retrieve, analyze, and calculate anomalies of daily hydrologic time series data (Ryberg and Vecchia, 2017)

- One of the most useful computational advances in recent years: **hydrological data acquisition** via R packages.
- Many provide **vignettes** (tutorials) showing how to plot, manipulate, or visualize the data in a few lines of code
- Meteorological and climatological data access is also facilitated



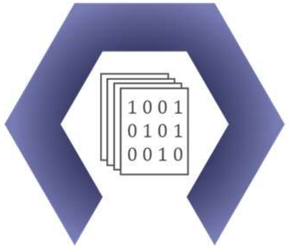
Source: Vitolo et al. 2016

tidyverse packages
(source www.dicook.org)

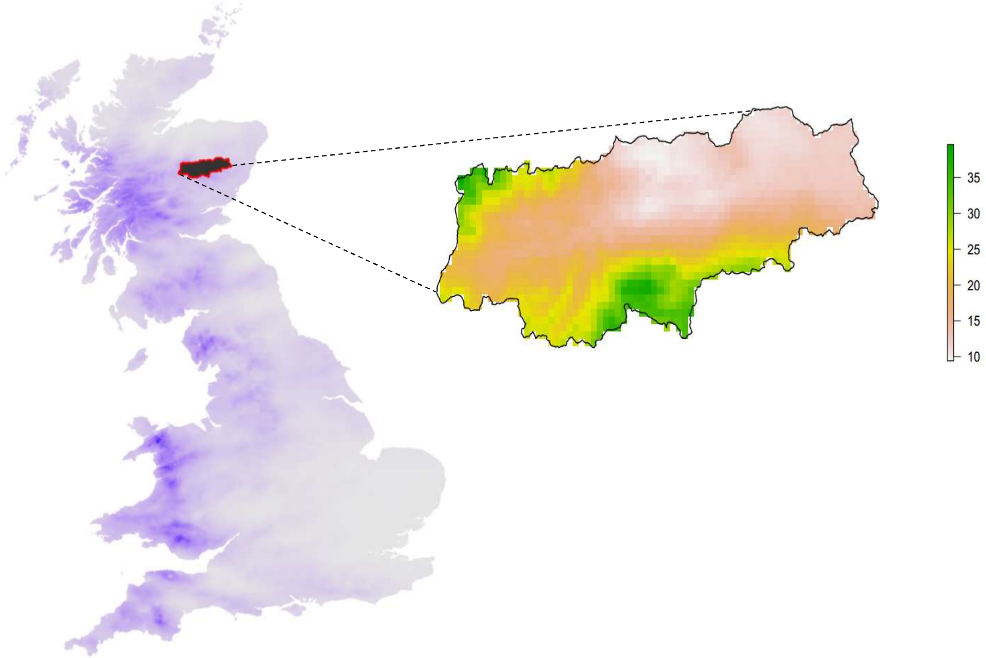
- R handles basic (csv, text with base-R) through to specific data formats:
  - xls(x), fwf (`tidyverse`, `readr`)
  - Netcdf (`ncdf4`, `easyNCDF`, `stars`, `raster`)
  - GRIB (`raster`, `gribr`)
  - Open Geospatial Consortium services (`sos4R`, `ows4R`, `geometa`)
- … and the tidyverse provides a range of packages for reading, cleaning and tidying different types of data (though not specifically hydrological)

Using R in Hydrology, *HESSD*, doi: 10.5194/hess-2019-50

easyNCDF, ncdf4, daymetr, stars, ecmwfr, prism, getMet, (SPATIAL: rgeos, rgdal, raster, sp, proj4)



4 Extract driving data



- R is a GIS in its own right (e.g. Lovelace et al. 2019)
  - Can handle **multiple GIS formats**: vectorial data and maps (`sp`, `sf`), geometrical operations (`rgeos`), GDAL (`rgdal`), GEOS (`rgeos`), raster data (`raster`), spatiotemporal arrays (`stars`)
  - **Linkages with other GISs**: ArcGIS (`RPyGeo`), GRASS (`spgrass6`, `rgrass7`), or QGIS (`RQGIS` or `RSAGA`)
- R facilitates easy integration of GIS within the hydrological workflow

A focus on rainfall-runoff modelling

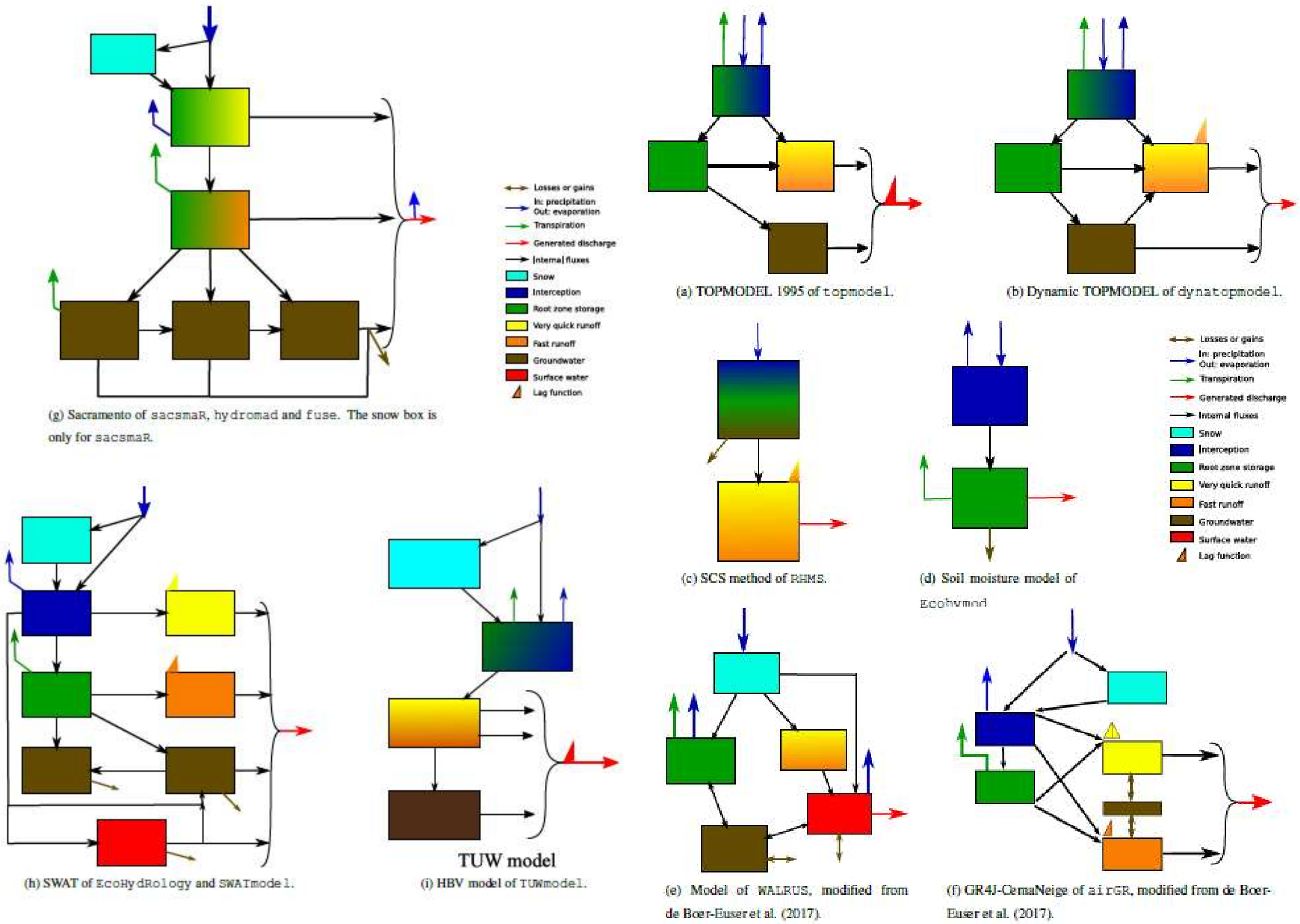
Many models are available in R!

- Simple to complex models
- Lumped to distributed models
- Sometimes include a snow routine or optimization algorithms

Package	Models
airGR	GR models (GR4J, etc.)
airGRteaching	GR models (GR4J, etc.)
dynatopmodel	Dynamic topmodel
EcoHydRology	SWAT
fuse	Modular structure
hydromad	Modular structure
RHMS	SCS-CN
sacsmaR	Sacramento
SWATmodel	SWAT
topmodel	Topmodel
TUWmodel	HBV
WALRUS	WALRUS

Using R in Hydrology, *HESSD*, doi: 10.5194/hess-2019-50

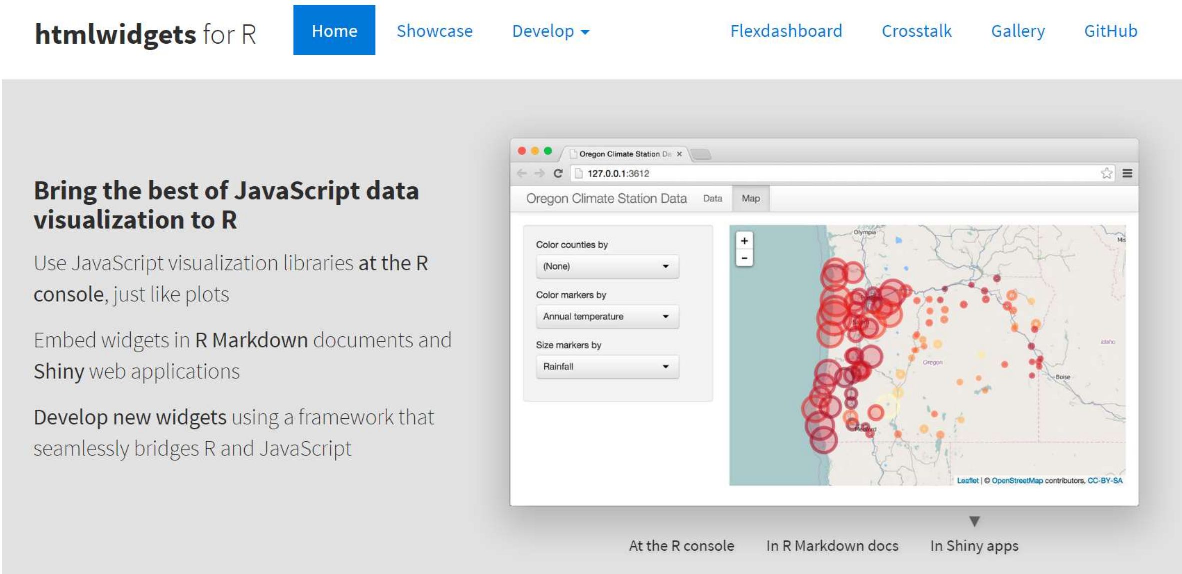
Overview of the available models



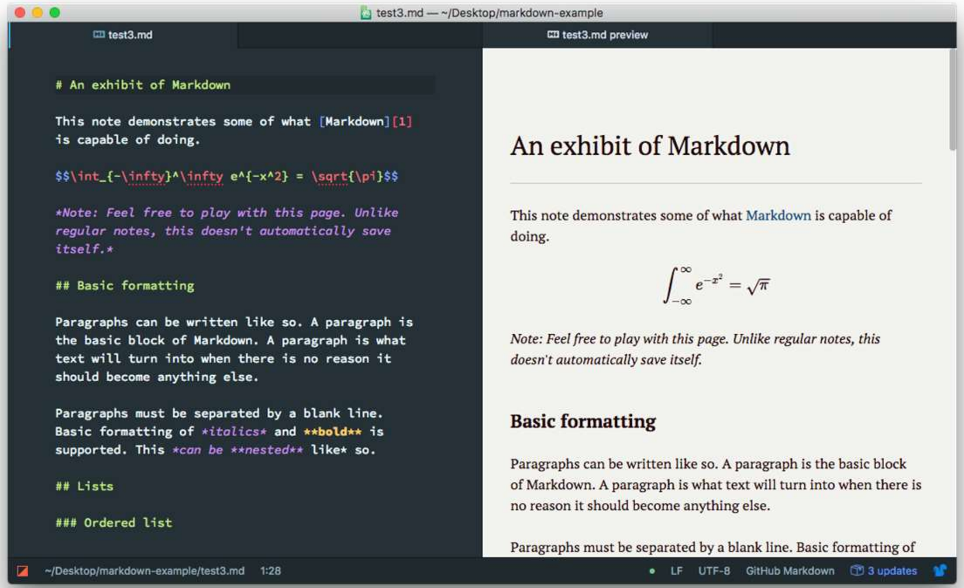
Package	Description
`berryFunctions`	Function Collection Related to Plotting and Hydrology (Boessenkool, 2018)
`hydroGOF`	Goodness-of-Fit Functions for Comparison of Simulated and Observed Hydrological Time Series (Zambrano-Bigiarini, 2017a)
`hydrolinks`	Hydrologic Network Linking Data and Tools (Winslow et al., In Prep)
`hydrostats`	Hydrologic Indices for Daily Time Series Data (Bond, 2018)
`hydroTSM`	Time Series Management, Analysis and Interpolation for Hydrological Modelling (Zambrano-Bigiarini, 2017b)
`lfstat`	Calculation of Low Flow Statistics for Daily Stream Flow Data (Koffler et al., 2016)

- R was initially developed as a statistical computing language – and still is
- Estimating procedures in base-R **`stats`** package (correlation, regression…)
- Generalist packages, e.g. for parametric trend tests, change-points, non-randomness (e.g. `trend`)
- Modelling packages useful for hydrologists: `mgcv`, `VGAM`, `gamlss`, `caret`
- Extreme Value Analysis (e.g. `extremes`; see the dedicated Task View)
- Regional frequency analysis: (`nsRFA, lmomRFA`); low-flow statistics (`lfstat`)
- Model evaluation metrics: `caret`, `ModelMetrics`, `hydroGOF`

Using R in Hydrology, *HESSD*, doi: 10.5194/hess-2019-50



- Hydrological data visualizations are straightforward and flexible in R
- Simple, customizable plots with base-R graphics package; vs. greater tuning to build almost any type of graphic with **ggplot2**
- **Facets** to split and plot data easily by categories (e.g. different months)
- Dynamic **JavaScript**-based libraries with zooming/panning or highlighting graphs or maps (`dygraphs`, `plotly`, `leaflet`, `plotGoogleMaps`, `googleVIS`…)
- Generate animated GIF files (`gganimate`, `caTools`)



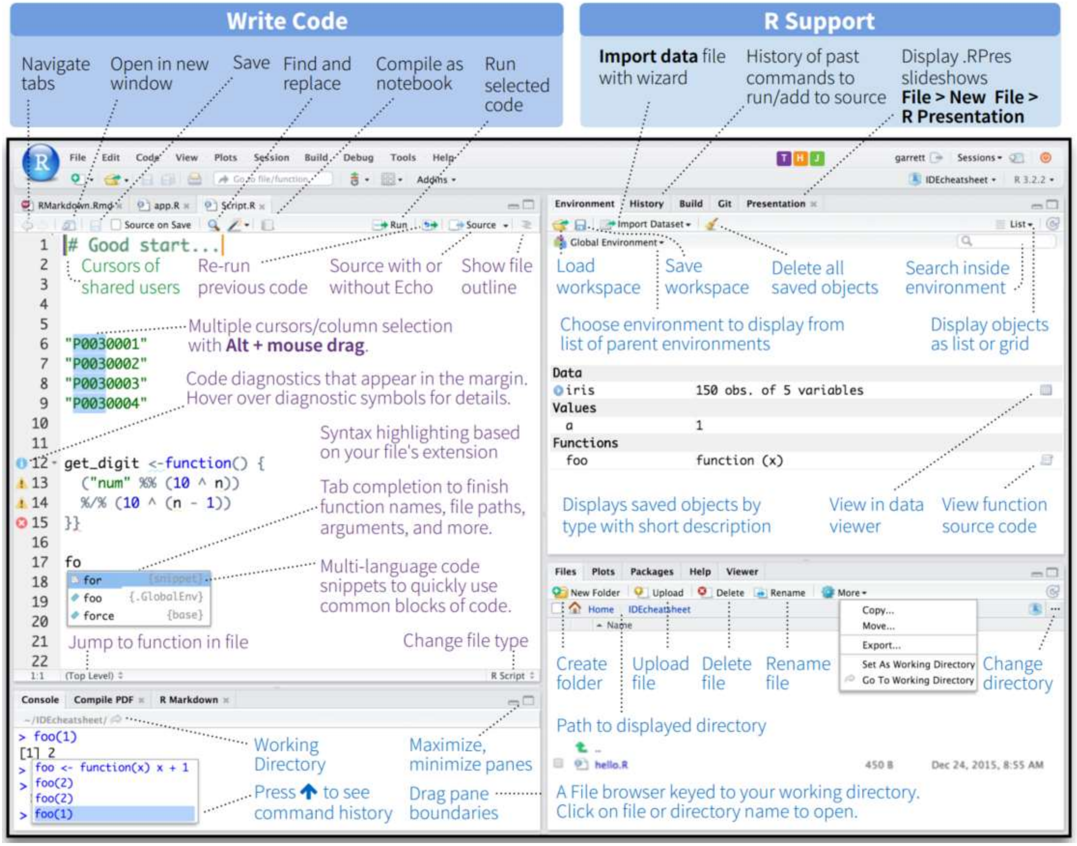
Using markdown to generate reports

(source: University of Illinois/NCSA)

- Many useful packages for creating hydrological presentations and documents
- Dynamic interfaces and web-based apps with `shiny` (e.g. `airGRteaching`)
- Create websites and blogposts (`rmarkdown`, `blogdown`, `pkgdown`)
- Generate presentations, books, reports (`knitr`, `markdown`, `rmarkdown`, `pagedown`, `bookdown`)

Integrated Development Environments (IDEs)

- Software applications used to **facilitate coding** (code editor, compiler, debugger, plot viewer)
- Many IDEs for R: Eclipse/StatET, ESS, RKWardm Tinn-R, Vim/Neovim
- The **RStudio** IDE is the most popular (allows visualizations, dynamic graphs/apps, version control, Rmarkdown presentations…)



The Rstudio IDE. Source: rstudio-IDE-cheatsheet.pdf

Big Data and Parallel Computing

Parallelize using: foreach and doParallel

The normal `for` loop in R looks like:

```
for (i in 1:3) {
  print(sqrt(i))
}
```

```
## [1] 1
## [1] 1.414214
## [1] 1.732051
```

The `foreach` method is similar, but uses the sequential `%do%` operator to indicate an expression to run. Note the difference in the returned data structure.

```
library(foreach)
foreach (i=1:3) %do% {
  sqrt(i)
}
```

```
## [[1]]
## [1] 1
##
## [[2]]
## [1] 1.414214
##
## [[3]]
## [1] 1.732051
```

In addition, `foreach` supports a parallelizable operator `%dopar%` from the `doParallel` package. This allows each iteration through the loop to use different cores or different machines in a cluster. Here, we demonstrate with using all the cores on the current machine:

```
library(foreach)
library(doParallel)
```

```
## Loading required package: iterators
```

```
registerDoParallel(numCores)  # use multicore, set to the number of our cores
foreach (i=1:3) %dopar% {
  sqrt(i)
}
```

```
## [[1]]
## [1] 1
##
## [[2]]
## [1] 1.414214
##
## [[3]]
## [1] 1.732051
```

Source: Jones 2017

- R is able to handle very large data files (packages `bigmemory`, `biganalytics`, or `data.table` for fast aggregation, e.g. 100GB in RAM)
- Many of the spatial data handling packages are relevant in hydrology, eg. for large shapefiles (`sf`) Database connections can be established (`RPostgreSQL`, `RSQLite`)
- Considerable performance boosts (time) achieved by parallelizing code (e.g. packages `parallel`, `foreach`, `doParallel`, `snowfall`, or `h2o`)

Packages as a driver of hydrological progress

rOpenSci Software Peer Review

build passing

Thank you for considering submitting your package to the rOpenSci suite. All the packages contributed by community members go through a process of open peer review to ensure a consistent level of quality for our users. This process also allows us to ensure that your package meets our guidelines and provides opportunity for discussion where exceptions are requested.

This README is a short intro to Software Peer Review for you as a potential author or reviewer. For more information, consult our gitbook "rOpenSci Packages: Development, Maintenance, and Peer Review".

Our code of conduct is mandatory for everyone involved in our review process.

- Why and how submit your package to rOpenSci?
- Why and how review for rOpenSci?
- Further resources
- Editors and reviewers

Why and how submit your package to rOpenSci?

- First, and foremost, we hope you submit your package for review **because you value the feedback**. We aim to provide useful feedback to package authors and for our review process to be open, non-adversarial, and focused on improving software quality.
- Once aboard, your package will continue to receive **support from rOpenSci members**. You'll retain ownership and control of of your package, but we can help with ongoing maintenance issues such as those associated with updates to R and dependencies and CRAN policies.
- rOpenSci will **promote your package** through our web page, blog, and social media. Packages in our suite are also distributed via our drat repository and Docker images, and listed in our task views.
- rOpenSci **packages can be cross-listed** with other repositories such as CRAN and BioConductor.
- rOpenSci packages that contain a short accompanying paper can, after review, be automatically submitted to the Journal of Open-Source Software for fast-tracked publication.

If you want to submit a package, read our guide for authors before opening a submission issue in this repository.

Why and how to review packages for rOpenSci?

- As in any peer-review process, we hope you choose to review **to give back to the rOpenSci and scientific communities.** Our mission to expand access to scientific data and promote a culture of reproducible research is only possible through the volunteer efforts of community members like you.
- Review is a two-way conversation. By reviewing packages, you'll have the chance to **continue to learn development practices from authors and other reviewers**.

https://github.com/ropensci/software-review

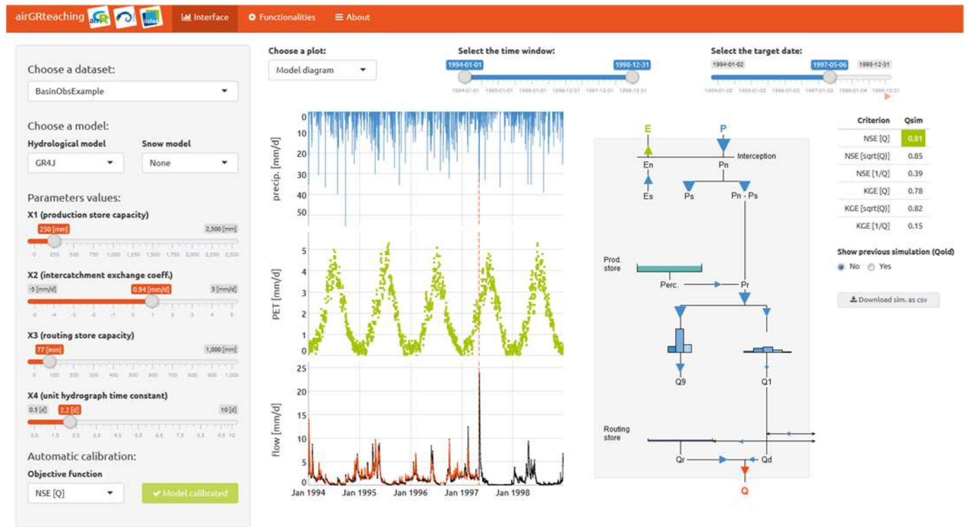
- Hydrological packages favor the **uptake and development** of hydrological methods
- Users can contribute **feedback** and help enhance existing code by commenting on collaboration platforms (e.g. GitHub or GitLab) or emailing the package maintainers
- It is crucial that we recognize the efforts that go into package development by **citing packages** and any accompanying journal papers
- Software **peer review** initiatives such as rOpenSci aim to help developers. Methods can be **published** in journals such as the *Journal of Open Source Software*

Teaching hydrology

- R provides an interactive tool for teaching hydrology
- Many examples of hydrological analyses can be found online as **tutorials** (see https://owi.usgs.gov/blog)
- R packages have been developed specifically to teach hydrology:

  `airGRteaching`: a suite of GR rainfall-runoff models such as GR4J for use by students with limited programming skills.

<- Talk in Room 513C at 5 pm



A shiny web application is included in `airGRteaching` package. Students can choose a catchment dataset, a hydrological model and a snow model and then explore both the data and model results.

Conclusions

- The open-source programming language R has acquired a **central role** in hydrological research and operational practice
- Both the **flexible** nature of the language and the diverse range of computational, visualization and modelling **tools** have facilitated the testing of hydrological theories and interactive teaching
- Within scientific research labs, we anticipate that committing code to a **repository** will become standard practice, as will the submission and **review of code** along with manuscript text
- We believe R will continue to facilitate further **advances** and will continue to play an increased role in hydrology

Have we missed something? Any specific question? If you have any suggestions please email:
louise.slater@ouce.ox.ac.uk or guillaume.thirel@irstea.fr

Hydrol. Earth Syst. Sci. Discuss., https://doi.org/10.5194/hess-2019-50
Manuscript under review for journal Hydrol. Earth Syst. Sci.
Discussion started: 18 February 2019

Hydrology and Earth System Sciences
Discussions
Open Access
EGU

Using R in hydrology: a review of recent developments and future directions

Louise J. Slater[a], Guillaume Thirel[b], Shaun Harrigan[c], Olivier Delaigue[b], Alexander Hurley[d], Abdou Khouakhi[e], Ilaria Prodoscimi[f], Claudia Vitolo[c], and Katie Smith[g]

[a]School of Geography and the Environment, University of Oxford, OX1 3QY, UK
[b]IRSTEA, HYCAR Research Unit, 1 rue Pierre-Gilles de Gennes, 92160 Antony, France
[c]Forecast Department, European Centre for Medium-Range Weather Forecasts (ECMWF), Shinfield Park, Reading, RG2 9AX, UK
[d]School of Geography, Earth and Environmental Sciences, University of Birmingham, B15 2TT, UK
[e]School of Architecture, Civil and Building Engineering, Loughborough University, Loughborough, UK
[f]Department of Environmental Sciences, Informatics and Statistics, Ca' Foscari University of Venice, Venice, Italy
[g]Centre for Ecology & Hydrology, Maclean Building, Crowmarsh Gifford, Wallingford, OX10 8BB, UK

Correspondence: Louise J. Slater (louise.slater@ouce.ox.ac.uk)

Abstract. The open-source programming language R has gained a central p… he hydrological sciences over the last decade, driven by the availability of diverse hydro-meteorological data ar… evelopment of open-source computational tools. The growth of R's usage in hydrology is reflected ir… blished hydrological packages, the strengthening of online user communities, and the populari… events. In this paper, we explore the benefits and advantages of R's usage in hydrology, such … data science and numerical literacy, the enhancement of reproducible research and open scie… al tools, the ease of connecting R to and from other languages, and the support provided by a … paper provides an overview of important packages at every step of the hydrological workflo… ydro-meteorological data, to spatial analysis and cartography, hydrological modelling, statis… ic and dynamic visualizations, presentations and documents. We discuss some of the challenges that … R in hydrology and useful tools to overcome them, including the use of hydrological libraries, documentatio… ettes (long-form guides that illustrate how to use packages); the role of Integrated Development Environments (IDEs); and the challenges of Big Data and parallel computing in hydrology. Last, this paper provides a roadmap for R's future within hydrology, with R packages as a driver of progress in the hydrological sciences, Application Programming Interfaces (APIs) providing new avenues for data acquisition and provision, enhanced teaching of hydrology in R, and the continued growth of the community via short courses and events.

Accepted!