



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

^aSchool of Geography and the Environment, University of Oxford, OX1 3QY, UK

^bIRSTEA, HYCAR Research Unit, 1 rue Pierre-Gilles de Gennes, 92160 Antony, France

^cForecast Department, European Centre for Medium-Range Weather Forecasts (ECMWF), Shinfield Park, Reading, RG2 9AX, UK

^dSchool of Geography, Earth and Environmental Sciences, University of Birmingham, B15 2TT, UK

^eSchool of Architecture, Civil and Building Engineering, Loughborough University, Loughborough, UK

^fDepartment of Environmental Sciences, Informatics and Statistics, Ca' Foscari University of Venice, Venice, Italy

^gCentre 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 place in the hydrological sciences over the last decade, driven by the availability of diverse hydro-meteorological data archives and the development of open-source computational tools. The growth of R's usage in hydrology is reflected in the number of newly published hydrological packages, the strengthening of online user communities, and the popularity of training courses and events. In this paper, we explore the

- 5 benefits and advantages of R's usage in hydrology, such as: the democratization of data science and numerical literacy, the enhancement of reproducible research and open science, the access to statistical tools, the ease of connecting R to and from other languages, and the support provided by a growing community. This paper provides an overview of important packages at every step of the hydrological workflow, from the retrieval of hydro-meteorological data, to spatial analysis and cartography, hydrological modelling, statistics, and the design of static and dynamic visualizations, presentations and documents.
- 10 We discuss some of the challenges that arise when using R in hydrology and useful tools to overcome them, including the use of hydrological libraries, documentation and vignettes (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
- 15 hydrology in R, and the continued growth of the community via short courses and events.

1 Introduction: the rapid rise of R in hydrology

In recent decades, the hydrological sciences, like many other disciplines, have witnessed major changes due to the growth of diverse data archives and the development of computational resources. Hydrology has benefited from the increase in publicly-accessible data, including: (a) observational river flow archives (Hannah et al., 2011) such as the World Meteorological Or-





ganization's Global Runoff Data Centre, which currently includes more than 9,500 stations from 161 countries; (b) gridded reanalysis climate data products such as Copernicus's ERA-Interim (Dee et al., 2011) or ERA-5 (Hersbach et al., 2018; Copernicus Climate Change Service, 2018); (c) measurements from sensors and satellites, such as total water storage variations from the Gravity Recovery and Climate Experiment (GRACE, Tapley et al., 2004) or snow cover area from NSIDC MODIS (Hall and Riggs 2016); and (d) catchment attributes such as the Global Streamflow Indices and Metadata Archive (Do et al., 2018)

5 and Riggs, 2016); and (d) catchment attributes such as the Global Streamflow Indices and Metadata Archive (Do et al., 2018), or national datasets such as GAGESII (Falcone, 2011) and CAMELS (Addor et al., 2017). Together, these datasets have facilitated the investigation of many catchments as well as interdisciplinary research linking meteorology, climatology, hydrology and the earth sciences.

In addition to the availability of large-scale data archives, the increase in computational power and uptake of programming languages have also been a major driver of change in the discipline. Increasingly, hydrologists are using data science approaches to derive process insights from large and complex datasets (e.g. Guo, 2017). The ability to explore and mine these datasets has facilitated a move from in-depth experiments in single catchments towards large-sample studies (e.g. Villarini et al., 2009; Berghuijs et al., 2014; Slater et al., 2015; Archfield et al., 2016; Blöschl et al., 2017; Blum et al., 2017; Harrigan et al., 2018). By applying an analysis to many catchments, or grouping those with similar characteristics, hydrologists have been able to

15 test the broad applicability of hydrological theories and draw systematic insights about hydrological processes (e.g. Blöschl et al., 2013). Large-sample studies are increasingly being employed to develop novel (conceptual and/or physical) models that are applicable across diverse catchment conditions, thereby improving process understanding and leading to more robust predictions (e.g. Gupta et al., 2014). At the same time, a broad range of well-established lumped (e.g. Coron et al., 2017, 2018), distributed (e.g. Buytaert, 2018) and semi-distributed (e.g. Metcalfe et al., 2018) models of the hydrological cycle - many of

20 which were initially developed in older languages (e.g. Fortran; Beven, 1997; Clark et al., 2008) - have now been translated to or coded in R.

The uptake of empirical, conceptual and physically-based computational hydrological models as well as the growth of largesample studies have driven the emergence of a field that can be described as *computational hydrology* (e.g. Hutton et al., 2016; Melsen et al., 2017): a scientific paradigm that allows hydrologists to explore hydrological questions with a computational ap-

25 proach. Computational hydrology aims to develop reproducible hydrological analyses that can be applied to many catchments, with the aim of generating novel and systematic process insights that improve our physical understanding of the hydrological cycle. The field differs somewhat from both *geocomputation* (e.g. Openshaw and Abrahart, 2000; Lovelace et al., 2019), which uses similar computational-intensive techniques but with a focus on spatial data analysis, or *hydroinformatics* (e.g. Abbott et al., 1991), which has a greater focus on information and communications technologies and/or black box approaches to address mater analysis.

30 address water-related issues.

The growth of computational hydrology has been enhanced by the development of the open-source programming language R, invented by Ross Ihaka and Robert Gentleman in the 1990s (R Core Team, 2018) and supported by an enthusiastic and rapidly-growing online community. As a free multi-platform language, R is highly versatile and has a wide range of uses including: data acquisition and provisioning, manipulation, analysis, modelling, statistics, visualization, and even well-developed geospatial

35 and geographic information system (GIS) applications. R can be used for generating reports, interactive presentations for teach-





ing or conferences, or even to prototype dashboards and web applications. One of the greatest strengths of R is its extremely active community of users, who, in the past 25 years, have developed and released in the public domain more than 10,000 packages spanning many scientific disciplines. The Comprehensive R Archive Network (CRAN, https://cran.r-project.org) is the main repository for these packages, hosted by a network of ftp and web servers around the world. The abrupt increase in the number of hydrological package updates in 2018 exemplifies the growing importance of the language in hydrology and strengthening of the community (Figure 1). These developments have allowed the R language to fit comfortably in production-



the number of hydrological package updates in 2018 exemplifies the growing importance of the language in hydrology and strengthening of the community (Figure 1). These developments have allowed the R language to fit comfortably in production-ready ecosystems (e.g. web applications) and take advantage of cutting-edge technologies and tools for improving reproducibility, testing and continuous integration.



Figure 1. The number of R packages available on CRAN (1997-2018). Left: all packages; right: hydrology packages only (defined as those that include the string "hydro" within the package metadata; erroneous packages containing terms like hydrocarbon were removed). Bar colors indicate number of packages published based on (1) date of first release (blue bars; https://cran.r-project.org/src/contrib/archive); (2) date of last publication - i.e. most recent publication date of packages that are currently available on CRAN (red bars; https://cran.r-project.org/web/packages/available_packages_by_date.html). In both cases each package is counted only once; for example the package hydroTSM was first released in 2010 (blue) but its most recent update is 2017 (red). The script to produce this figure is provided as part of the supplementary materials (Supplement 1). Data were downloaded from CRAN on 18 January 2019.

10

This paper aims to provide a broad overview of the utility of R in the hydrological sciences and of important developments in recent years. In Section 2 we provide a summary of the many benefits and advantages of using R in hydrology. In Section 3 we describe some of the key hydrological packages that have been developed by the community, as well as general packages of broad application in the sciences. In Section 4 we discuss some of the challenges that the R-Hydro community faces, including tools and solutions to overcome these challenges. Finally, in section 5 we list some of the future directions to strengthen the hydrological computational community.





5

2 The benefits and advantages of using R in hydrology

2.1 Democratizing data science and numerical literacy

One of the principal advantages of R is its ease of use, resulting from typically detailed documentation, a large number of online resources, and the ability to freely view and learn from source code (under the open-source license). Further, R can be run on all major operating systems (i.e., Microsoft Windows, macOS, Linux), making it ideal for institutional or personal use. In contrast with compiled languages, such as C or Fortran, R is an interpreted language, which means that the code can be written and executed line-by-line. In practice this means that achieving a basic hydrological analysis can be as simple as writing a sequence of commands to read a file, clean the data, and plot a graph. These sequences of commands are typically collated in an "R script"; together, multiple scripts may constitute a hydrological workflow (see Section 3.1). A high level of

- 10 documentation and support exists for each of the tasks within a hydrological workflow, including a diverse range of packages (see Section 3). R-Users may search for R-relevant content (such as expressions, packages and functions) using the dedicated internet search engine, RSeek (https://rseek.org), as well as generalist search engines such as Google or Bing. For many R newcomers, the initial hurdle is finding the time required to understand how the R language works; but these efforts pay off rapidly as one learns how to find solutions on the Internet and to navigate the help files provided within R.
- 15 One of the advantages of writing a script is that it can be reused and improved incrementally over time, so the author never has to repeat a task manually (in contrast to "point-and-click" software that lack a programmatic interface). Moreover, the same script can be used repeatedly by different people to reproduce a given result (avoiding duplication of efforts) or to analyze different data: e.g. different catchments, different types of gridded data, or the same data at different points in time. Thus, scripts written in R (or other languages) have wide-ranging benefits, as they: facilitate the testing and quality-control
- 20 of the scientific workflow, can be shared and improved by a team of users, lessen the risk of making manual mistakes, and significantly enhance the speed with which analyses can be conducted.

Recent developments in R have contributed to enhancing numerical literacy in the hydrological sciences. R's ease of access and use has improved what might be termed "numerical or scientific computing literacy" within the hydrological community. Volunteer projects such as the Software Carpentry (Wilson, 2006, 2014) have been teaching basic computing skills to

- 25 researchers since 1998, and R now forms a central part of their training. Over the past decade or so, R has become one of the core tools for scientific computation in hydrology. Hosted instances of R allow the user to run R and RStudio[®] (an IDE, described in greater detail in section 4.2) in the cloud, i.e. in a web browser rather than locally on one's own computer. These hosted instances have made the language more accessible to non-specialists, due to the large range of pre-installed packages. Importantly, the RStudio Cloud (https://rstudio.cloud/) has recently been developed "to make it easy for professionals, hobby-
- 30 ists, trainers, teachers and students to do, share, teach and learn data science using R", and provides many learning materials, including interactive tutorials covering the basics of data science, cheatsheets for working with popular R packages, links to DataCamp[®] courses, and a guide to using RStudio Cloud. Such hosted instances of R remove the initial hurdle of installing R and the required packages (i.e. the technicalities), and ultimately make scientific work more accessible therein democratizing knowledge and advancement.





The hydrologic R-Users (R-Hydro) community has developed a number of platforms to share computational hydrology analyses. Code and results can be published via traditional media (e.g. as articles, supplemental material, scripts, packages, or computational environments) or on the web via blog posts, snippets or tutorial documents, allowing users to engage interactively. The "literate programming" paradigm (i.e., interspersing snippets of code and English within a unique document; see

- 5 Knuth, 1984) has also become increasingly popular due to the development of dashboards and online publishing platforms. Dashboards are interfaces that may include a group of related data visualizations such as charts, graphs, tables and maps (see e.g. Figure 5). They can be fixed (e.g. a non-interactive web page) or dynamic (e.g. allowing users to alter input values and see how outputs change virtually in real-time). Hydrologists have increasingly employed dashboards to display their analyses in an interactive, user-friendly manner, using packages such as plotly, a graphing library that makes publication-quality
- 10 graphs online (Sievert, 2018), the shiny web application framework (Chang et al., 2015), shinydashboard (Chang and Borges Ribeiro, 2018), or the flexdashboard package (Iannone et al., 2018). Additionally, free services such as RPubs (a free web-publishing platform, https://rpubs.com) and the aforementioned plotly enable the publishing of these static or interactive documents and visualizations online. In doing so, these publishing services facilitate immediate interaction and knowledge transfer.

15 2.2 Enhancing reproducible hydrological research and open science

Reproducibility is a key feature of the scientific method and can be broadly defined as the ability for the community to reproduce and verify previous findings. Encouraging reproducible practices helps reduce the likelihood of errors or falsification, while increasing the uptake of any positive developments within a discipline (Hutton et al., 2016). However, scientific research is increasingly under fire for its lack of reproducibility due to missing or inadequate methodology description (Ceola et al.,

20 2015) and model and data availability. True reproducibility requires more than the mere repeatability of results with the same computer code and data: one must also be able to reproduce a study's conclusions when testing the theory with different data or a different model setup (Melsen et al., 2017).

The open-source nature of R packages and the CRAN repository set-up are one key added value of R to reproducible research. The CRAN repository ensures the traceability of past analyses by archiving former versions of the packages compiled

- on any platform (https://cran.r-project.org). In addition, packages are citeable in reports and papers together with their version number, allowing the user to track which code was used. The package developers are also "traceable" on the CRAN via their ORCID number, which provides an indication of whether they also authored any corresponding scientific papers. The CRAN Task Views provide guides to the packages and tools that exist on CRAN for the different disciplines. Note that there are also many packages on the GitHub repository that never make it to CRAN for a range of reasons, but this does not
- 30 necessarily make them less reliable. The Task Views provide a list of tools which can enhance reproducibility in R (see https: //cran.r-project.org/web/views/ReproducibleResearch.html). Relying on well-established publishing platforms such as CRAN and GitHub has promoted a standardized format for developing and disseminating R code. Packages such as rhub (Csárdi, 2017) also provide code checks for packages on certain operating systems. Other initiatives, such as rOpenSci (carefully vetted





scientific R software tools, see https://ropensci.org), and their code peer-review, have facilitated the implementation of best practices whilst holding authors to scientific scrutiny.

Journals in the field of hydrology such as Hydrology and Earth System Sciences, Water Resources Research, and the ASCE journals Journal of Water Resources Planning and Management and Journal of Hydrologic Engineering now actively encour-

- age authors to publish the data and computer codes underlying the results presented in their papers. The journal *Nature* states 5 that a manuscript can be rejected if the code used to generate new analyses cannot be provided to the editors/reviewers. Despite the advantages of sharing hydrological code, few computational hydrologists do this because cleaning and annotating the code places an additional burden on the publishing timeframe. However, it is reasonable to assume that, as more journals require submission of codes with papers, the community of computational hydrologists (and associated fields) will continue to grow
- and strengthen the science. In addition to the sharing of open-source code, reproducibility experts also advocate the use of 10 software toolsets such as version control, scripting, container technology, and computational notebooks to enhance the reproducibility of scientific results (Perkel, 2018). Hydrological tutorials, vignettes or teaching documents increasingly implement literate programming (Section 2.1), where the code and results are described in plain English within one same document or webpage.
- 15 In the hydrological sciences, several ongoing open scientific initiatives can be noted, other than the R packages that are discussed in Section 3. These initiatives include the HydroShare web-based system for sharing hydrologic data and models, which allows hydrologists to visualize, analyze, and work with data and models on the HydroShare website (e.g. creating and publishing a Web App resource; Essawy et al., 2018). Additionally, the Sharing Water-related Information to Tackle Changes in the Hydrosphere for Operational Needs (SWITCH-ON) virtual laboratory aims to explore the potential of Open Data for water
- security and management (Ceola et al., 2015). The reproducibility agenda has benefited from strong political and financial 20 support, with European Union projects like FOSTER Plus (Fostering the practical implementation of Open Science in Horizon 2020 and beyond) that aim to encourage open science. The European Commission's "Science with and for Society" program, implemented as part of Horizon 2020 (European Commission, 2015, p. 17) similarly noted that the re-use of research data generated with public funding should have beneficial impacts for science, the economy and society. As such, open science is 25 expected to encourage interdisciplinary research, and is considered a key approach to tackle the grand challenges of our time.

2.3 Providing statistical tools for hydrology

There are many different types of software for statistical analysis, but R is still considered the most powerful language and environment for statistical computing. R is "GNU S", a language which can be described as the modern implementation of the S language (Chambers, 1998) and is specifically optimized for statistical computing. In addition to the standard statistical

30 techniques within base-R (i.e., the in-built basic functions that define R as a language), R also provides access to a large variety of advanced and recent statistical packages, which have been developed by its user community of statisticians and statisticallyminded scientists working in a range of research fields. When comparing R with other similar open-source languages, users often describe R's unique selling point as the vast number of statistical packages which liken it to a free, community-driven version of statistical software.





The statistical and graphical packages provided in R are particularly useful for the hydrological sciences, and include techniques such as linear and nonlinear modelling, statistical tests, time series analysis, classification, or clustering (see https://cran.r-project.org). A description of specific statistical packages relevant to hydrology is provided in section 3.6.

2.4 Connecting R to and from other languages

5 Many different programming languages are used in hydrology, including older languages such as Fortran, developed in the 1950s (e.g. Backus, 1978) and C in 1972 (e.g. Ritchie, 1993); proprietary commercial languages e.g. MATLAB[®], initially released in 1984 (MathWorks, 2018); and more recent open-source languages like Python, which appeared in 1990 (e.g. Sanner et al., 1999; Guttag, 2013), R in 1993 (e.g. R Core Team, 2018; Ihaka, 1998), and Julia in 2012 (Bezanson et al., 2012). Interconnections between R and these different languages can be very useful for taking advantage of each language's
10 strengths, or using pre-existing scripts that were originally written in other languages.

S and R were both built using algorithms implemented mostly in Fortran and sometimes in C (Chambers, 2016, p.55), which is why R can be connected natively to both languages. For example, airGR (Coron et al., 2017, 2018) makes use of Fortran codes for the hydrological models in order to take advantage of the CPU efficiency of Fortran when dealing with vector operations. R can also be connected to different languages using a range of packages, e.g. C++ (package Rcpp, Ed-

- 15 delbuettel and Balamuta, 2017) and Java (package rJava, Urbanek, 2018). Connections to Python can be achieved with the packages reticulate (Allaire et al., 2018b), rPython (Bellosta, 2015), rJython (Grothendieck and Bellosta, 2012), and XRPython (Chambers, 2017). Additionally, R can also be connected to Javascript (e.g. package V8, Ooms, 2017), Matlab (e.g. package R.matlab, Bengtsson, 2018), or Julia (packages JuliaCall and XRJulia, Chambers, 2018; Li, 2018). Conversely, connecting other languages to R is also possible. For example, rpy2 (Gautier, 2018) is the Python interface to
- 20 the R language and runs an embedded R, providing access from Python using R's own C-API. The Python library pyRserve (Heinkel, 2017) also connects Python with R using Rserve. The RCall package allows users to call R from Julia (Bates et al., 2015).

2.5 Interacting with the R-Hydro community: scientific resources and courses

- One of the major advantages of R is the extensive user community, which provides ample support to newcomers through various initiatives, and is growing at a fast pace. R-Hydro beginners are strongly encouraged to join the discussion on various R-related topics on social media (e.g. on Twitter using the hashtag "#rstats") to get a sense of the most widely-used tools, best practice and latest developments. StackOverflow (https://stackoverflow.com) is the go-to online discussion forum for any user, from beginners to expert developers, and provides code-snippets to solve a wide range of common problems. In addition, the R project also offers thematic mailing lists (https://www.r-project.org/mail.html), including lists relating to usage (R-help),
- 30 package development (R-package-devel) and language development (R-devel), in order to help others, report bugs, or propose solutions to fix them. More recent but already extremely active is the RStudio Community forum (https://community.rstudio. com), which includes users interested in RStudio-developed applications and packages.





5

A wide range of scientific resources, including online manuals and tutorials in several languages, have been developed by the community. The rOpenSci project (https://ropensci.org) brings together a community of volunteers who promote the open development of packages in a non-profit initiative, by reviewing scientific R packages before they are uploaded to CRAN. rOpenSci has contributed notably to the development of best practice by contributing free code reviews and targeted/invited blog posts. Resources can be found in many places online, such as the *Journal of Open Source Software (JOSS)*, which hosts a

wealth of short papers documenting R packages.

Short courses, i.e., voluntary training sessions run during conferences like the European Geosciences Union (EGU), have also grown in popularity in the hydrological community in recent years (Figure 2). Since the 2017 EGU General Assembly, the "Using R in Hydrology" short course has been organized in conjunction with the Young Hydrologic Society (YHS, https:

- 10 //younghs.com) as part of the annual Hydrological Sciences division program. The core aim of the short course is to bring together the R-Hydro community, including both students who are learning to program and advanced users who are either teaching computational hydrology or developing packages for hydrological analysis. Recent advances in R for hydrology applications are demonstrated during the short course and a platform for open discussion between guest speakers and course participants is provided. By rotating topics each year, a valuable repository of slides, code examples, and follow up material is
- 15 provided to the community on GitHub (https://github.com/hydrosoc).



Figure 2. EGU Short course "Using R in Hydrology" (https://meetingorganizer.copernicus.org/EGU2018/session/28914). Credit: A. Khouakhi, 11 April 2018.

Finally, it is worth mentioning that the continued growth of the R community is supported by the R Consortium (https: //www.r-consortium.org/about), a group with an open-source governance and foundation model that supports the worldwide community of users, maintainers and developers of R software. The R Consortium provides support to the community in multiple ways. One example of such support is the grant program run by the R Consortium's Infrastructure Steering Committee (ISC), which funds development of projects seeking to promote improvement of the R infrastructure and to achieve long term

20





3 R packages in hydrology

3.1 The hydrological workflow

R is an ever-growing environment, as can be seen in the number of R packages that are developed every year (Figure 1). There are now hydrological packages for every step of a standard hydrological workflow (Figure 3); we describe these steps in subsequent sections. A typical hydrological workflow might, for example: (1) Set up an online repository so that versioned code can be uploaded repeatedly, establishing version control (i.e. a reviewable history), then search for the most relevant packages (Section 3.2); (2) Retrieve hydrological data online, using hydrological data retrieval packages or web APIs (Section 3.3). (3) Manipulate, clean and tidy the data. This step may involve handling missing data, checking data completeness, reshaping and aggregating data, or converting strings to date format. We do not develop this section specifically because these are general

- tasks that are not specific to hydrology. (4) Extract driving meteorological/climatological data, for example by extracting catchment-averaged precipitation from gridded data held within netCDF files, and manipulating the spatial data (Section 3.4).
 (5) Conduct hydrological modelling by using the inputs prepared in previous steps (Section 3.5 and Table 2). (6) Conduct statistical analyses, visualize and verify the data (e.g. by quantifying the skill of the hydrological model simulations, assessing change and trends in the data, or extracting statistics on hydrological extremes; see Section 3.6). Exploratory Data Analyses
- 15 (EDA) may also occur much sooner in the workflow. (7) Visualize the data by producing high-quality figures and maps. This step may occur at any step of the hydrological workflow and is supported by many of the spatial packages discussed in Sections 3.4 and 3.7. (8) Publish results. Any resulting papers, reports, presentations, or dynamic interactive web-apps can also be written in R (Section 3.8). In an ideal workflow, the final code might be cleaned, annotated, and shared.



Figure 3. A typical hydrological workflow in R, containing eight steps (see Section 3.1). A selection of relevant R packages used within each script is indicated in coloured text; these packages are described in further detail in sections 3.3 to 3.8.





3.2 Finding the right package: the CRAN hydrology Task View and social media

CRAN Task Views were recently developed to provide thematic lists of the packages that are most relevant to specific disciplines. The hydrology Task View for "Hydrological Data and Modelling" (https://cran.r-project.org/web/views/Hydrology. html) was published to the CRAN in January 2019 and lists over a hundred packages under the following categories:

5

15

- Data retrieval: hydrological data sources (surface or groundwater, both quantity and quality); meteorological data (e.g. precipitation, radiation, temperature, both measurements and reanalysis).

- Data analysis: data tidying (e.g. gap-filling, data organization, quality control); hydrograph analysis (functions for working with streamflow data, including flow statistics, trends, biological indices); meteorology (functions for working with meteorological and climate data); spatial data processing.
- 10 - Modelling: process-based modelling (scripts for preparing inputs/outputs and running process-based models); statistical modelling (hydrology-related statistical models).

Additionally, many of the other 38 Task Views that were available in January 2019 were relevant for hydrology (https://cran. r-project.org/web/views). For example, the "EnvironMetrics" Task View contains a Hydrology and Oceanography section, and the "High-Performance and Parallel Computing with R" Task View can also be useful for hydrologists using large amounts of

data. Other relevant Task Views cover a range of topics such as time series analysis, machine learning, or spatial analysis. One last way of discovering relevant and useful hydrological packages is social media, such as Twitter, where many hydrologists share their most recent publications as well as links to useful resources and packages. Some of the Twitter handles that highlight relevant packages for computational hydrology include the USGS group supporting R scientific programming (https://twitter.com/USGS_R) or the rOpenSci page (https://twitter.com/ropensci).

20 3.3 Packages for retrieving hydro-meteorological data

One of the most useful computational advances in recent years has been the development of packages designed specifically to retrieve data from online hydrological archives. Different packages have been designed for importing hydrometric data from repositories such as dataRetrieval for the U.S. Geological Survey (USGS)'s National Water Information System (Hirsch and De Cicco, 2015), or rnrfa for the UK's National River Flow Archive (Vitolo et al., 2016a). Many of these packages provide vignettes that illustrate how to use the functions. For example, the USGS's waterData package allows the user to

25 import USGS daily hydrological data from the USGS web services, plot the time series data, fill in missing values, or visualize anomalies over a range of timescales (e.g. one year, 30 days or one day), in just a few lines of code (see the waterData package vignette on the CRAN; Ryberg and Vecchia, 2017). Other relevant packages are listed in Table 1.

Additionally, many data retrieval packages that are relevant for hydrological analyses have been developed by related scien-30 tific disciplines, such as meteorology and climatology. In the future it seems likely that most water and meteorological agencies around the world will facilitate access of these data via APIs and open-source packages (see Section 5.2 for further information on APIs and the possible future of hydro-meteorological data provision). Table 1 is far from exhaustive and there are many





other relevant packages that are available on CRAN. For example, the new ecmwfr package provides a straightforward interface to the public ECMWF API Web Services and the Copernicus Climate Data Store (CDS), where open datasets such as the ERA-5 reanalysis are openly available (Hufkens, 2018).

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)

 Table 1. Examples of packages for hydrological and/or meteorological data retrieval.

Once data have been retrieved or downloaded, a broad range of packages are available for reading different types of data and their associated metadata. However, these packages are not specifically hydrological and so will be discussed here with brevity. Note that in many cases, for example with the dataRetrieval package, the data are imported directly into the R workspace. Basic data formats such as *csv* and *txt* can be read with base-R (read.table function), or with additional packages from, for example, the tidyverse (Wickham, 2017) package suite, which facilitate reading a range of formats including *xls(x), tsv, fwf* (e.g. package readr; Wickham et al., 2018).

10 For reading or writing netcdf files, a number of packages like ncdf4 (Pierce, 2017), easyNCDF (BSC-CNS and Manubens, 2017), stars (Pebesma, 2018b), or raster (Hijmans, 2017) can be used. GRIB files can also be handled with packages such as raster or gribr (Wendt, 2018). There is still relatively minimal support for Open Geospatial Consortium (OGC) services in R, although there are some packages such as sos4R (of which a new version is expected to be released on CRAN soon), ows4R (Blondel, 2018b), or geometa (Blondel, 2018a).





3.4 Packages for spatial analysis and cartography

In the past, R may have been a less powerful alternative to the more established spatial software for processing large datasets. Now, however, R can be parallelised more easily than other software (harnessing the power of multiple processor cores to handle large datasets) and can integrate GIS analyses within a complete, automated hydrological workflow, which includes data processing steps (before or after any GIS analyses) and any subsequent statistical analyses. It is this integration of GIS as one step within the hydrological workflow that makes R extremely attractive. As a result, in recent years, R has become the "go to" for geocomputation and geostatistics and can now be used as a GIS in its own right. Multiple books have been published on the topic of spatial analysis and mapping with R (Brunsdon and Comber, 2015), or more broadly, geocomputation with R (Lovelace et al., 2019), which includes topics such as reading and writing geographic data and making maps in R.

- Many methods are now implemented within R for handling vectorial data, with packages such as sp (Pebesma and Bivand, 2005; Bivand et al., 2013) for plotting data as maps or for spatial selection; package rgdal (Pebesma and Bivand, 2005; Bivand et al., 2013), which provides bindings to the Geospatial Data Abstraction Library (GDAL) for reading/writing data and access to projection and transformation operations from the PROJ.4 (Urbanek, 2012) library; package rgeos (Bivand and Rundel, 2018), which provides an interface to the Geometry Engine Open Source (GEOS) library for geometrical operations;
- 15 or package sf (Pebesma, 2018a), which provides support for simple features to encode spatial vector data. There is also a range of packages for handling raster data, like the package raster (Hijmans, 2017), which can be used to read, write, manipulate, analyze and model gridded spatial data; and stars (Pebesma, 2018b) for reading, manipulating, writing and plotting spatiotemporal arrays. With all these packages, R now has full cartography and mapping functionality, and can produce sophisticated maps that are either static or interactive. Packages such as cartography (Giraud and Lambert, 2016) or tmap
- 20 (Tennekes, 2018) can both be used for creating symbols, choropleth or other types of maps, and for facilitating the visualisation of spatial data distributions in thematic maps. These two packages are quite similar in content but not in form: cartography uses the painter's model and while tmap uses the Grammar of Graphics and allows users to draw static and dynamic maps with the same code. Additionally, ggmap (Kahle and Wickham, 2013) is commonly used for visualizing, geolocating and routing spatial data on top of static maps such as Google Maps; while the JavaScript leaflet (Cheng et al., 2018) library can be
- 25 used for creating interactive web maps. R software can also be linked to a range of other GIS software to take advantage of specific capabilities. R can access ArcGIS geoprocessing tools (ESRI, 2018) by building an interface between R and the ArcPy Python side-package using the RPyGeo package (Brenning et al., 2018b). R can also call wrappers for GDAL/OGR utilities (GDAL Development Team, 2018) using gdalUtils (Greenberg and Mattiuzzi, 2018). There are also interpreted interfaces between the GRASS GIS (GRASS Development Team, 2019) and R from within the GRASS environment or from R, using the
- 30 packages spgrass6 (Bivand, 2016) or rgrass7 (Bivand, 2018). R can be integrated with QGIS (QGIS Development Team, 2018) using RQGIS (Muenchow et al., 2017), or with SAGA GIS (SAGA Development Team, 2008) using RSAGA (Brenning et al., 2018a).





 Table 2. Examples of packages for hydrological modelling.

Package	Description
airGR	Suite of GR Hydrological Models for Precipitation-Runoff Modelling (Coron et al., 2017, 2018)
dynatopmodel	Implementation of the Dynamic TOPMODEL Hydrological Model (Metcalfe et al., 2018)
Ecohydmod	Ecohydrological Modelling (Souza, 2017)
fuse	Ensemble Hydrological Modelling (Vitolo et al., 2016b)
hydromad	Hydrological Model Assessment and Development (Andrews and Guillaume, 2018)
RHMS	Hydrologic Modelling System for R Users (Arabzadeh and Araghinejad, 2018)
topmodel	Implementation of the Hydrological Model TOPMODEL in R (Buytaert, 2018)
TUWmodel	Lumped Hydrological Model for Education Purposes (Viglione and Parajka, 2014)

3.5 Packages for hydrological modelling

Hydrological modelling often proceeds by simplifying hydrological processes to test hypotheses about the water cycle. These models are useful to manage water resources, to reconstruct incomplete flow time series, to predict extreme events (floods or low flows), or to anticipate the effects of future climatic or anthropogenic changes. In Table 2 we highlight some of the key packages that facilitate the implementation of certain hydrological models in R. As R can be used for every step within the

5 packages that facilitate the implementation of certain hydrological models in R. As R can be used for every step within the hydrological modelling process, from importing and cleaning data, to exploratory analyses, data modelling, data analysis, and graphical visualization, it represents an ideal language for hydrological modellers.

Several well-known hydrological models are provided in these packages, such as the HBV model in the TUWmodel package (Viglione and Parajka, 2014); TOPMODEL in the packages topmodel (Buytaert, 2018) and dynatopmodel (Metcalfe

- 10 et al., 2018); SWAT in SWATmodel (Fuka et al., 2014); and GR4J in airGR (Coron et al., 2017, 2018) and hydromad (Andrews and Guillaume, 2018). Some packages also include degree-day snow models (e.g. airGR, hydromad and TUWmodel) that are used in nival basins to simulate the accumulation and melt of the snowpack, which greatly impacts the flow regimes. Many of the models included in these packages are lumped (i.e., consider the catchment as a single unit with area-averaged variables) or conceptual (i.e., provide a simplified representation of the physical processes with empirical equations describing
- 15 the interactions between the processes) models. Only the TOPMODEL packages include (semi-)distributed models. Additionally, the vast majority of the models in the above packages are deterministic; however, the fuse package can also provide ensembles of conceptual lumped hydrological models.

The above packages typically allow the user to run the hydrological models, and usually provide some sample input data, with executable examples. Some packages provide optimization algorithms, criteria calculation and dedicated plotting func-

20 tions (e.g. airGR and hydromad). Developing hydrological modelling packages in R is relatively straightforward because of





the language's flexibility; additionally, as discussed in Section 2.4, R can be connected to other programming languages. This property allows R-users to embed already existing model codes into R and provides a more friendly environment, as many packages ease the analysis of data and model simulations.

3.6 Packages for hydrological statistics

- 5 R was initially developed as a statistical computing language and is still the primary language in which novel statistical methods are coded and distributed. Statistical approaches are employed for an extremely wide range of tasks in hydrology and it is virtually impossible to give a complete coverage of all possible packages that might be useful to hydrologists. Many estimating procedures can be carried out using the base-R stats package, which includes for example correlation analysis and (Mann-)Kendall testing, linear regression, Poisson regression and Gamma regression. There are also generalist packages for non-
- 10 parametric trend tests such as trend (Pohlert, 2018), which includes a broad range of tests for trend detection, correlation, as well as detection of change-points or non-randomness. Many general modelling packages have been found to be useful by hydrologists, for example mgcv (Wood, 2017), VGAM (Yee et al., 2015) and gamlss (Rigby and Stasinopoulos, 2005), which implement generalized additive regression models for a large number of distributional families; or the caret package (Kuhn et al., 2018) for predictive modelling and machine learning, which allows the user to develop an impressive range of predictive
- 15 models (e.g. neural networks, deep learning and decision trees).

There are many packages available for common hydrological tasks. A comprehensive set of functions for carrying out Extreme Value Analysis (Coles, 2001) can be found in the extRemes package (Gilleland and Katz, 2016), and a more complete overview can be found in the CRAN "Extreme Value Analysis" Task View. nsRFA consists of a collection of statistical tools for objective (non-supervised) applications of the Regional Frequency Analysis methods in hydrology (Viglione, 2018). The

- 20 lfstat (Koffler et al., 2016) package provides functions to compute statistics and plots for low flows similar to those in the manual on low-flow estimation and prediction (WMO, 2009). Many of the standard model evaluation metrics are provided directly within these packages. The caret package, for instance, imports other packages, such as ModelMetrics (Hunt, 2018) to facilitate model evaluation directly. A range of packages have also been developed specifically for model evaluation. The hydroGOF package (Zambrano-Bigiarini, 2017a) provides goodness-of-fit functions specifically for comparison of sim-
- 25 ulated and observed hydrological time series. In Table 3 we present a brief overview of some other packages for hydrological statistics. For a broader overview of available packages and further examples, we point the reader to the CRAN Task View on machine learning and statistical learning (https://cran.r-project.org/web/views/MachineLearning.html).

3.7 Packages for static and dynamic visualizations

Data visualizations play an important role in hydrological analysis: R makes them straightforward to implement, and allows considerable flexibility. R includes three main families of graphics packages: a painter model, natively present in R and based on the S language's GRZ model, the Trellis graphs (e.g. the lattice package, Sarkar (2008)) initially implemented in S, and *The Grammar of Graphics* (Wilkinson, 1999). The last two families are both based on the grid package (R Core Team, 2018),





Table 3. Examples of packages for hydrological statistics.

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)

a low-level abstraction layer that allows the development of graphical packages with different philosophies, which means that the created graphics can be stored in objects to be updated and plotted again later on.

Base-R can produce publication-quality figures; it includes a series of functions and methods that allow the user to plot various types of visualizations and the output of statistical models. Visualizations are produced in base-R using the graphics

- 5 package (R Core Team, 2018), which relies on the philosophy of superimposing elements of the graphics (with no provision for deleting an element once it is drawn), and includes a range of functions to add straight lines, arrows, axes, boxes, grids, legends, or annotation to a plot. While visualizations produced via base-R are highly customizable, other packages provide more consistent interactions and a suite of highly useful functions to summarize, highlight and split/layer data with a minimal amount of code. A community favorite is the package ggplot2, created by Wickham (2016b), which allows a high level
- 10 of flexibility and "tuning" of the graphs depending on the users' needs. ggplot2 is an implementation of the Grammar of Graphics and allows users to build almost any type of graphic (see the R graph gallery: https://www.r-graph-gallery.com). ggplot2 also contains functions like "facets" (https://plot.ly/ggplot2/facet) which allow the user to split and plot the data by categories in many different rows or columns (e.g. to compare results for different seasons).
- Dynamic charts where the user can, for instance, hover over one or multiple points to read the associated (meta)data 15 (e.g. a hydrometric station number or the value of a point) - have also grown in popularity in hydrological analyses in recent years. These dynamic graphics are particularly useful to inspect data, such as outliers, or to explain an analysis when teaching hydrology in the classroom (e.g. by zooming in on different parts of a time series). Dynamic graphics including maps can be created by using the plotly package (Sievert, 2018), a graphic library that allows the user to make graphs interactive with minimal extra code. A number of JavaScript-based visualization libraries can also help achieve dynamic graphics. For
- 20 interactive time-series data requiring axis display features such as zooming/panning and highlighting of series/points, the dygraphs package (Vanderkam et al., 2018) provides an interface to the JavaScript charting library. The manipulate





10

package (Allaire, 2014) allows the user to easily add sliders and other control tools to otherwise static plots. Other JavaScriptbased packages include rbokeh (Hafen and Continuum Analytics, Inc., 2016), which provides an interface to Bokeh, allowing the user to create web-based plots with hover and dynamic functionalities. rAmCharts (Thieurmel et al., 2018) creates interactive charts based on the amcharts.js library, while the highcharter (Kunst, 2017) package is based on the Highcharts JavaScript graphics library. Examples of all of these implementations (including leaflet, dygraphs, plotly, rbokeh,

5 highcharter, and other visualisation libraries) can be found at https://www.htmlwidgets.org. We encourage the reader to explore these HTML widgets as they show how simple it is to implement dynamic graphics with just a line or two of R code. In addition to the above packages that allow the hydrologist to create their own graphics, many hydrological packages

have also been developed with built-in functions to plot hydrological data, such as the aforementioned waterData package (Ryberg and Vecchia, 2017).

3.8 Packages for creating presentations and documents

A vast array of packages have been developed in R for creating dynamic presentations and documents, which are particularly useful for illustrating hydrological concepts. Dynamic interfaces and web-based apps can be created with shiny (Chang et al., 2015), such as the airGRteaching (Delaigue et al., 2018a, b) interface (Figure 5, see Section 5.3). R can also be used to

generate animated GIF files for presentations, for example to highlight temporal changes in land cover or reservoir levels from 15 Earth Observation. GIFs can be created using for example gganimate (Pedersen, 2018) to animate ggplot2 and create videos and animated image files, or caTools (Tuszynski, 2018) for reading and writing GIF and ENVI binary files.

Additionally, various packages have been designed to produce interactive maps, such as leaflet (Cheng et al., 2018), leafletR (Graul, 2016), plotGoogleMaps (Kilibarda and Bajat, 2012), or googleVis (Gesmann and de Castillo,

2011). R also offers many tools for creating websites or blogposts, with packages like rmarkdown (Allaire et al., 2018c), 20 blogdown (Xie et al., 2017) or pkgdown (Wickham and Hesselberth, 2018).

Presentations, books, reports and documents can be generated in LaTeX, or Markdown, natively with Sweave functionalities, or with packages such as knitr (Xie, 2018), markdown (Allaire et al., 2018a), rmarkdown (Allaire et al., 2018c), pagedown (to Paginate the HTML output of R Markdown with CSS for printing; Xie and Lesur, 2019) or bookdown (Xie,

25 2016). These packages facilitate the automation of such documents (e.g. monthly reports), through in-line text tokens that are dynamically linked to data in R's workspace.

Challenges and solutions when using R in hydrology 4

Hydrological libraries, documentation, and vignettes 4.1

For most hydrologists who are new to R, the initial hurdle is understanding how to install libraries and use packages to explore their own data sets. The book *R packages* (Wickham, 2015) is freely available online and explains everything from the basic

30





5

installation of packages to the role of metadata, understanding documentation, the role of vignettes, and best practices on GitHub (one common collaboration and version control platform).

R packages centralized on CRAN are structured similarly, with a reference manual, source code, a license file, and other common elements. The code and documentation of all packages is verified before they are uploaded to CRAN. R packages ideally provide two forms of documentation: a short form (help pages) and a long form (vignettes), which are both complementary and serve different purposes (Wickham, 2015). The help pages explain what each function does, describe the required

input and the produced output, and usually include a section with executable examples. Vignettes, in contrast, are tutorials that

illustrate how R packages and their functions are used, often with discussion of the outputs. However, not all packages include a link to a vignette on the CRAN repository, as this is not compulsory. Developing clear and useful vignettes is one of the key
challenges in facilitating the uptake of new packages and methods!

4.2 Integrated Development Environments (IDEs): facilitating the use of R

IDEs are software applications that are used to facilitate coding by providing the code editor, compiler or interpreter and debugger within a single graphical user interface (GUI). There exist a range of IDEs for R, such as Eclipse/StatET, (an Eclipse based IDE; e.g. des Riviêres and Wiegand, 2004; Wahlbrink, 2016), Emacs Speaks Statistics (ESS, an add-on package for GNU

- 15 Emacs; Rossini et al., 2004), Microsoft's R Tools for Visual Studio (RTVS, a plug-in for the Microsoft Visual Studio IDE; e.g. Beard, 2016), RKWard (which integrates with the KDE desktop environment; Rödiger et al., 2012), Tinn-R (a replacement for the basic code editor designed to replace Rgui; Faria et al., 2008), or Vim/Neovim (a terminal-based working environment designed to facilitate working in a pure command-line interface; e.g. Neil, 2018).
- The RStudio IDE (RStudio Team, 2018) is the most popular of the IDEs. RStudio facilitates the uptake of R by hydrologists,
 by providing a helpful research tool and training environment to new and experienced programmers alike. RStudio is available in two editions: an open-source edition and an enterprise edition with a commercial license. Both can run on a desktop (RStudio Desktop) or server (RStudio Server) and can be installed on different platforms (Windows, macOS, and Linux). Most hydrologists use the free RStudio Desktop edition, although university departments and companies increasingly offer server editions too. RStudio's features include: a console, a syntax-highlighting editor that supports direct code execution, integrated
- 25 R help and documentation, support for version control systems, as well as tools for plotting, history, debugging and workspace management.

The RStudio environment makes it straightforward for hydrologists to conduct a range of tasks, e.g. visualise data, create dynamic graphs or web-apps with shiny, use version control, or develop and view Rmarkdown presentations e.g. in Latex, HTML or PowerPoint), all within the IDE. Other advantages of RStudio include the easy access to the help window, the ease

30 of package management and updating, code debugging, or even the easy development of packages in one location. All of these benefits contribute to facilitating open hydrological science. RStudio is currently in version 1.1, but version 1.2 will include many new features to help interact with other programming platforms. These new features include, for example, the integrated support for the reticulate (Allaire et al., 2018b) and r2d3 (Luraschi and Allaire, 2018) packages, making it easy to call





Python and D3 from within R sessions and R Notebooks. An exhaustive list of features can be found on the RStudio website (https://www.rstudio.com/products/RStudio).

4.3 Big Data and parallel computing challenges in hydrology

In the early years of R, the software was unable to handle large data files exceeding millions of rows or complex data formats.

- 5 However, both of these limitations have since been overcome. Some of the early packages for handling large data files include bigmemory (Kane et al., 2013) or biganalytics (Emerson and Kane, 2016), where matrices are allocated to shared memory and use memory-mapped files. More recently, the package data.table (Dowle and Srinivasan, 2018) has gained increasing popularity for enhanced data frames, allowing fast aggregation of large data (e.g. 100GB in RAM) or fast ordered joins. In hydrology, some of the spatial data handling packages are particularly relevant, such as the sf package (Pebesma,
- 10 2018a) for large shapefiles, or the feather (Wickham, 2016a) package for feather files, a lightweight binary columnar data store designed for maximum speed. Database connections can be established using a range of packages such as RPostgreSQL (Conway et al., 2017) or RSQLite (Müller et al., 2018). Distributed dataframes can be implemented using packages such as sparklyr (Luraschi et al., 2018).

As the volume of available data increases and hydrologists use a greater number of models and ensembles, parallel com-

- 15 puting where many calculations are carried out simultaneously, instead of sequentially has become an essential tool in computational hydrology and has sped up analyses. For instance, instead of using traditional for-loops (an approach were one action is carried out over a data set iteratively), the data may be broken down into groups (e.g. by year or by season) and functions can be applied to each group in parallel. The performance boosts that can be achieved by parallelizing the code (i.e. using more than one core at a time) are considerable. Even without access to a high-performance computer or cluster, it is possible to
- 20 perform hydrological tasks faster, since most local machines now have between four to 16 cores. R has multiple facilities and packages to enable the parallelization of code execution. At the most simple level, base-R functions like lapply and sapply can be used to apply a specific function to a vector/list input, which can speed up analyses considerably. For instance, the base-R parallel package for distributed computing provides methods to access additional computational power by allocating available processing cores into a cluster. The foreach (Microsoft and Weston, 2017) and doParallel (Microsoft Corporation and Weston, 2018) packages can also be used together to execute "for-each" loops in parallel.
 - Other packages that are widely used for parallel computing in hydrology and other areas include the snowfall (Knaus, 2015) package, which facilitates the development of parallel R programs, e.g. by including extended error checks. Additionally, some packages have parallel functionalities that are integrated directly within the package, such as the h20 package (LeDell et al., 2018) that provides an interface to H2O, a scalable platform offering parallelized implementations of supervised and unsupervised machine learning algorithms.
- 30 unsupervised machine learning algorithms.





5 A roadmap for the future of R in hydrology

As we have shown above, the development of R fosters progress in hydrology. In this section we discuss what we perceive as some of the future avenues for enhancing hydrological research or operational hydrological practice with R, and how we can achieve these as a community.

5 5.1 R packages as a driver of progress in hydrology

The consistent stream of new R packages has been a great driver for progress not only in hydrology, but even in science more broadly. Most importantly, hydrological packages favor the uptake and development of hydrological methods and acceleration of science. Sharing code increases the likelihood that an approach will be used by other scientists in their research. McKiernan et al. (2016) for instance indicate that "...open research is associated with increases in citations, media attention, potential

10 collaborators, job opportunities and funding opportunities. These findings are evidence that open research practices bring significant benefits to researchers relative to more traditional closed practices." Sharing code saves new users the trouble of "reinventing the wheel" and writing codes that have already been developed by others.

Additionally, the open-source nature of R packages means that different users can contribute feedback to R package developers and help enhance existing code. This feedback between users and developers is one key route to scientific progress.

- 15 Users can identify issues or suggest improvements by commenting on online collaboration platforms (e.g. GitHub or GitLab) or by emailing the maintainer. Most packages are hosted on the GitHub platform before becoming available via the CRAN archive (ensuring a certain standard and best practices are met). Although CRAN itself does not have any mechanism to check the quality or cleanness of the code, there is a suite of packages that are used to (i) ensure clean code that follows a widely accepted style-guide, such as the lintr package (Hester, 2018) and (ii) rigorously test the functions against known outcomes,
- 20 using for example the testthat package (Wickham, 2011).

Developing an R package requires a structured approach, just like writing a scientific paper. There are many generalist resources for writing R packages, such as the R packages book (see http://r-pkgs.had.co.nz, Wickham, 2015), or a suite of video tutorials on YouTube. The main requirements for publishing a package are listed on the CRAN website. Providing documentation and tutorials, e.g. via blogposts or good *Readme* files, is essential. Package authors can also provide user-

25 friendly and modular functions to aid maintenance and future development (e.g. through external contributions or suggested changes on online platforms). The RHub service also provides a "build and check" service for R packages (Csárdi, 2017).

Authoring hydrology-based R packages that can stand up to scientific scrutiny and ensure user-friendliness is by no means a simple or brief task. Hence, such investment should be recognized within the community. Fueling the development and dissemination of new R-based methods is therefore the joint responsibility of developers, authors and journal editors. If developers

30 include digital object identifiers (DOI) as well as instructions for citation within their software, then authors can subsequently cite these packages. The adequate reference(s) can be obtained with the citation function for every package. The references generally comprise a reference to the package (with CRAN link and package version, which is key for encouraging reproducibility) and sometimes also include an additional journal paper reference. If both are available, then ideally both must be





5

cited to provide recognition of computational scientists' contribution to the hydrological community and to enhance the reproducibility of the research. Editors might also provide platforms for sharing computational approaches, such as special issues or sections for technical notes. Dedicated software journals such as the *Journal of Open Source Software* can be used to publish brief, technical descriptions of R packages. Any potential apprehensions for publishing such methods (e.g. due to a lack of scientific scrutiny) can be alleviated through software peer review initiatives such as those provided by rOpenSci.

5.2 APIs: hydrological data acquisition and provision

Application programming interfaces (APIs) play a key role in hydrological data acquisition and provision, and are likely to become increasingly important in the future. An API is a set of code which usually includes subroutine definitions, communication protocols, and tools for building software and interacting with different datasets. APIs are use-case specific by definition,

- 10 but the interface is often provided in HTTP (i.e., Web protocol), so requests and responses can be made and received by a wide range of languages or systems. For R-Hydro users, hydrological interaction with HTTP APIs usually comes in the form of data acquisition packages such as the aforementioned packages rnrfa (Vitolo et al., 2016a), tidyhydat (Albers, 2017), or daymetr (Hufkens et al., 2018). These packages provide means of posting requests via the R command line, and then return an R object for subsequent manipulation. Internally, the source code translates all inputs and outputs to and from HTTP. Pro-
- 15 grammatically accessing a variety of vast online resources (i.e. data) in this way has considerably advanced our understanding in the earth sciences.

Recent developments, however, have opened up a new, and arguably under-utilized, approach to APIs for the R community: rather than exploiting an existing interface, R users can now increasingly rely on a set of tools to develop and make accessible their own APIs for use by third parties via HTTP. Noteworthy projects here are OpenCPU (client and scalable cloud implemen-

20 tation, Ooms, 2014) and RStudio's plumbr package (Lawrence and Wickham, 2018). API development is likely to become a major avenue of future advances in scientific computing within the hydrological community, as well as interaction beyond the scientific realm. Such APIs can provide access to latest raw or analyses-ready data, methods, or even entire (statistical) models written in R.

The simplicity and ubiquitous implementation of HTTP has vast implications, of which we highlight four. 1) Common 25 issues with interoperability between languages can be overcome, and more attention can be afforded to gaining insights, rather than developing (often convoluted) language bridges (termed a "separation of concerns"; Council and Heineman, 2001). This means, for the first time, fully seamless interaction between research groups working in different languages is possible (e.g. between R, Python and MatLab). 2) Computational infrastructure can be scaled with comparatively little technical knowledge, increasing R's aptitude to a wider range of applications. 3) Outputs can be made accessible to third parties with or without a

30 deeper understanding of the underlying science. 4) Empirical data is rarely "clean" and scientific computing hence has to be adaptable: R source code can be altered to account for changing data, while maintaining a standardized, public-facing API. To highlight potential strengths of such APIs, we provide a brief, interdisciplinary use-case scenario rooted in natural hazard management and hydrological sciences in Figure 4.







Figure 4. Use-case scenario for implementation of APIs developed within hydrological sciences. Group A collates the most recent earth and ground-based observations of land cover, topography and climate data for an in-house project, yet provides an API to interact with an analyses-ready, spatially-explicit data set for their region of interest. Group B is interested in geomorphological processes - and has developed a model to predict likely occurrence of mass movements, which they also provide access to via an API. During a period of extreme rainfall over a scientifically unrelated region, both groups decide to adjust their methods and API to provide pertinent data (group A), which feed into predictions of landslide risk (group B). Finally, a disaster relief organization (group C) can swiftly act and use the outputs from the adapted API (A + B) to develop a simple web application with maps and warnings for use by the general public in the affected area.

5.3 Teaching hydrology in R

Due to its relative ease of use and open-source nature, R is increasingly being used as an interactive tool for teaching in the hydrological sciences. Many examples of typical hydrological analyses can be found online as tutorials (see for example, Hurley, 2018). The USGS have also published relevant examples on their hydrological blog at https://owi.usgs.gov/blog which may be of interest to the community.

```
5
```

R packages such as airGRteaching or TUWmodel have also recently been developed to facilitate teaching of hydrological modelling. The TUWmodel package (Viglione and Parajka, 2014) provides a lumped conceptual rainfall-runoff model, based on the well-known HBV model (Bergström and Forsman, 1973) and is designed for education purposes. The airGRteaching package (Delaigue et al., 2018a, b) provides access to a suite of GR rainfall-runoff models such as GR4J

10 for use by students with limited programming skills: only three basic functions are needed to prepare the data, calibrate a model and run/evaluate a simulation, and a single function can produce plots directly using objects from the three previously-





airGRteaching 🕵 👩 🔜 🔟 Interface 🛛 & Functionalities



Figure 5. The airGRteaching interface that is used for teaching hydrological modelling. Students can choose a catchment dataset, a hydrological model and activate a snow model on the top left panel of the interface. Modifying the parameters values (on the left) by using the sliding bars will automatically update the graphs and scores displayed (on the right). This interface highlights in an interactive manner the basic principle of the rainfall-runoff relationship and its description in hydrological models. On the bottom left of the interface, an objective function can be chosen and an automatic calibration procedure can be launched within the interface.

mentioned functions. In addition, a graphical shiny web application is included to explore data and model results with automatic updating of the plots and scores when the model parameters are modified by students; the app allows users to explore the internal fluxes and state variables of the models (Figure 5). Such tools can be used for understanding and illustrating the main hydrological processes and concepts and for relating them to hydrological models components and parameters.

- 5 There is now an increasing number of online applications that allow beginners to learn R in a sandbox, i.e., a virtual space for testing coding online. Sandboxes are particularly useful for introducing basic methods in computational hydrology, without having to master the technicalities of R. Examples of R sandboxes include the RStudio Cloud (https://rstudio.cloud), which can be used to learn, teach, or test R code. Another sandbox, initially called R-fiddle, is now called datacamp-light (https://github. com/datacamp/datacamp-light). Many other R sandboxes can be found online. Thus, rather than installing R on multiple PCs,
- 10 sandboxes can be used by educators/lecturers by asking students to connect directly to the web.





5



Figure 6. The R community. Left: the R meetup groups https://www.meetup.com/topics/r-project-for-statistical-computing; Right: the R-Ladies global organization https://rladies.org/about-us.

5.4 Developing the community: short courses, help desks and meet-up events

This paper reflects the strong collective desire to develop the community of computational hydrologists in R. As mentioned in section 2.5, the R-Hydro community has been meeting regularly in recent years at the EGU General Assembly during the Short Course "Using R in Hydrology". This course is run annually by the Young Hydrologic Society (YHS) and is typically attended by a wide range of hydrologists, ranging from beginners to more experienced users (Figure 2). The resources and teaching

- presentations from the short course are made available to the community online (see Boessenkool et al., 2017; Slater et al., 2018). These presentations include a range of topics such as: using R as a GIS; discharge time-series visualization; extreme value statistics; hydrological modelling; trend analysis; accessing hydrological data using web APIs; extracting gridded netCDF data; report generation; and other hydrological tasks in R.
- 10 The global reach of R meet-up groups has also grown rapidly, such as the official R-users group meet-up (http://r-users-group. meetup.com). Many local initiatives for R users have regular meet-ups and seminar series and can be found either on the meetup website or on social media webpages, such as Twitter (e.g. https://twitter.com/rusersoxford). There is also an ongoing push for greater inclusivity of under-represented groups in programming, as exemplified by the rapid growth of the global "RLadies" movement (Figure 6).
- 15 In addition, the computational community has been trying to provide support to other programmers by running "Coding Help Desks" such as at the American Geophysical Union (AGU) 2018 Fall meeting, in the Career Center. Geoscientist volunteers





ran the desk in 2018 to provide perspective and advice to other coders, with a range of short, 10-minute tutorials on topics such as keeping track of code versions with GitHub, making pretty plots, or using a new package/library. We anticipate that such help desks, short courses and meet-up sessions will continue to help grow the computational hydrology community in future years.

5 6 Conclusions

Over the last decade, the open-source programming language R has acquired a central role in hydrological research as well as in the operational practice of hydrology. With the rapidly increasing number of packages that are now available for every step of the hydrological workflow, R facilitates a broad range of hydrological analyses from start to finish. This paper provides an overview of the use of the open-source programming language R in hydrology, by describing these packages as well as the

10 influence of R on the discipline. Both the flexible nature of the language and the diverse range of computational, visualization, and modelling tools (physically-based and statistical) have facilitated the testing of hydrological theories over a range of spatial and temporal scales, as well as interactive teaching of hydrology within the classroom.

By encouraging others to use the language, to share their codes, propose new packages or contribute to the improvement of existing packages, we believe R will continue to facilitate further advances in hydrology, with wide-ranging improvements

- 15 of hydrological theory, models and tools. These new computational tools and approaches are essential to achieving long-term goals such as the International Association of Hydrological Sciences (IAHS)'s Science Plan for the decade 2013-2022, "*Panta Rhei*: Change in Hydrology and Society", which seeks to improve the assessment, attribution, and modelling of hydrological change. The rise of computational hydrology is also playing a key role in enhancing the reproducibility of science and the computational literacy of both scientists and practitioners. Within scientific research labs, we anticipate that committing code
- 20 to a repository will become standard practice, as will the submission and review of code along with manuscript text as part of the scientific publication process. Ten years from now, with the continued rise in the teaching of open-source programming in schools - and of computational modelling within university curricula - it is plausible to expect that R will play an increased role in hydrology.

Competing interests. The authors declare that they have no conflict of interest

25 *Acknowledgements.* The authors thank all those who have contributed to the R-Hydro community and the R community more broadly, whether by developing and sharing code, or teaching/helping others to use these tools. In particular, we thank the guest speakers who have participated in the "Using R in Hydrology" EGU short course.





References

Abbott, M. B. et al.: Hydroinformatics: information technology and the aquatic environment, Avebury Technical, Aldershot, 1991.

Addor, N., Newman, A. J., Mizukami, N., and Clark, M. P.: The CAMELS data set: catchment attributes and meteorology for large-sample studies, Hydrology and Earth System Sciences, 21, 5293–5313, https://doi.org/10.5194/hess-21-5293-2017, 2017.

- 5 Albers, S.: tidyhydat: Extract and Tidy Canadian Hydrometric Data, The Journal of Open Source Software, 2, https://doi.org/10.21105/joss.00511, 2017.
 - Allaire, J.: manipulate: Interactive Plots for RStudio, https://CRAN.R-project.org/package=manipulate, r package version 1.0.1, 2014.
 - Allaire, J., Horner, J., Xie, Y., Marti, V., and Porte, N.: markdown: 'Markdown' Rendering for R, https://CRAN.R-project.org/package= markdown, r package version 0.9, 2018a.
- 10 Allaire, J., Ushey, K., and Tang, Y.: reticulate: Interface to 'Python', https://CRAN.R-project.org/package=reticulate, R package version 1.10, 2018b.
 - Allaire, J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham, H., Cheng, J., and Chang, W.: rmarkdown: Dynamic Documents for R, https://CRAN.R-project.org/package=rmarkdown, R package version 1.10, 2018c.

Andrews, F. and Guillaume, J.: hydromad: Hydrological Model Assessment and Development, http://hydromad.catchment.org, R package

15 version 0.9-26, 2018.

Arabzadeh, R. and Araghinejad, S.: RHMS: Hydrologic Modelling System for R Users, https://CRAN.R-project.org/package=RHMS, R package version 1.4, 2018.

Archfield, S. A., Hirsch, R. M., Viglione, A., and Blöschl, G.: Fragmented patterns of flood change across the United States, Geophysical research letters, 43, https://doi.org/10.1002/2016GL070590, 2016.

20 Backus, J.: The history of Fortran I, II, and III, ACM Sigplan Notices, 13, 165–180, 1978.

Bates, D., Lai, R., Byrne, S., and contributors: RCall package, https://github.com/JuliaInterop/RCall.jl, 2015.

Beard, B.: Setup and Installation of R Tools for Visual Studio, in: Beginning SQL Server R Services, pp. 33-71, Springer, 2016.

- Bellosta, C. J. G.: rPython: Package Allowing R to Call Python, https://CRAN.R-project.org/package=rPython, r package version 0.0-6, 2015.
- 25 Bengtsson, H.: R.matlab: Read and Write MAT Files and Call MATLAB from Within R, https://CRAN.R-project.org/package=R.matlab, r package version 3.6.2, 2018.
 - Berghuijs, W., Woods, R., and Hrachowitz, M.: A precipitation shift from snow towards rain leads to a decrease in streamflow, Nature Climate Change, 4, 583, https://doi.org/10.1038/nclimate2246, 2014.
 - Bergström, S. and Forsman, A.: Development of a conceptual deterministic rainfall-runoff model, Nordic Hydrol., 4, 147-170,
- 30 https://doi.org/10.2166/nh.1973.0012, 1973.
 - Beven, K.: TOPMODEL: a critique, Hydrological processes, 11, 1069–1085, https://doi.org/10.1002/(SICI)1099-1085(199707)11:9<1069::AID-HYP545>3.0.CO;2-O, 1997.
 - Bezanson, J., Karpinski, S., Shah, V. B., and Edelman, A.: Julia: A fast dynamic language for technical computing, arXiv preprint arXiv:1209.5145, 2012.
- 35 Bivand, R.: spgrass6: Interface between GRASS 6 and R, http://CRAN.R-project.org/package=spgrass6, R package version 0.8-9, 2016. Bivand, R.: rgrass7: Interface Between GRASS 7 Geographical Information System and R, https://CRAN.R-project.org/package=rgrass7, R package version 0.1-12, 2018.





15

- Bivand, R. and Rundel, C.: rgeos: Interface to Geometry Engine Open Source ('GEOS'), https://CRAN.R-project.org/package=rgeos, R package version 0.4-2, 2018.
- Bivand, R., Pebesma, E., and Gomez-Rubio, V.: Applied spatial data analysis with R, Use R!, Springer, New York, second edn., http://www.asdar-book.org, 2013.
- 5 Blondel, E.: geometa: Tools for Reading and Writing ISO/OGC Geographic Metadata, https://CRAN.R-project.org/package=geometa, R package version 0.4-0, 2018a.

Blondel, E.: ows4R: Interface to OGC Web-Services (OWS), https://CRAN.R-project.org/package=ows4R, R package version 0.1-0, 2018b.

- Blöschl, G., Sivapalan, M., Savenije, H., Wagener, T., and Viglione, A.: Runoff prediction in ungauged basins: synthesis across processes, places and scales, Cambridge University Press, Cambridge, 2013.
- 10 Blöschl, G., Hall, J., Parajka, J., Perdigão, R. A., Merz, B., Arheimer, B., Aronica, G. T., Bilibashi, A., Bonacci, O., Borga, M., et al.: Changing climate shifts timing of European floods, Science, 357, 588–590, https://doi.org/10.1126/science.aan2506, 2017.

Blum, A. G., Archfield, S. A., and Vogel, R. M.: On the probability distribution of daily streamflow in the United States, Hydrology and Earth System Sciences, 21, 3093–3103, https://doi.org/10.5194/hess-21-3093-2017, 2017.

Boessenkool, B.: berryFunctions: Function Collection Related to Plotting and Hydrology, https://CRAN.R-project.org/package= berryFunctions, R package version 1.17.0, 2018.

Boessenkool, B., Harrigan, S., Smith, K., and Klotz, D.: Using R in Hydrology, EGU2017 Short Course. Zenodo, http://doi.org/10.5281/ zenodo.2554019, 2017.

- 20 Brenning, A., Bangs, D., and Becker, M.: RSAGA: SAGA Geoprocessing and Terrain Analysis, https://CRAN.R-project.org/package= RSAGA, R package version 1.3.0, 2018a.
 - Brenning, A., Polakowski, F., and Becker, M.: RPyGeo: ArcGIS Geoprocessing via Python, https://CRAN.R-project.org/package=RPyGeo, R package version 1.0.0, 2018b.

Brunsdon, C. and Comber, L.: An introduction to R for spatial analysis and mapping, Sage, Los Angeles, https://uk.sagepub.com/en-gb/eur/

- an-introduction-to-r-for-spatial-analysis-and-mapping/book241031, 2015.
 BSC-CNS and Manubens, N.: easyNCDF: Tools to Easily Read/Write NetCDF Files into/from Multidimensional R Arrays, https://CRAN. R-project.org/package=easyNCDF, R package version 0.0.4, 2017.
 - Buytaert, W.: topmodel: Implementation of the Hydrological Model TOPMODEL in R, https://CRAN.R-project.org/package=topmodel, R package version 0.7.3, 2018.
- 30 Ceola, S., Arheimer, B., Baratti, E., Blöschl, G., Capell, R., Castellarin, A., Freer, J., Han, D., Hrachowitz, M., Hundecha, Y., Hutton, C., Lindström, G., Montanari, A., Nijzink, R., Parajka, J., Toth, E., Viglione, A., and Wagener, T.: Virtual laboratories: new opportunities for collaborative water science, Hydrology and Earth System Sciences, 19, 2101–2117, https://doi.org/10.5194/hess-19-2101-2015, 2015. Chamberlain, S.: rnoaa: 'NOAA' Weather Data from R, https://CRAN.R-project.org/package=rnoaa, r package version 0.8.4, 2019. Chambers, J. M.: Programming with data: A guide to the S language, Springer Science & Business Media, 1998.
- 35 Chambers, J. M.: Extending R, The R series, Chapman and Hall/CRC Press, Boca Raton, London, New York, 2016. Chambers, J. M.: XRPython: Structured Interface to 'Python', https://CRAN.R-project.org/package=XRPython, r package version 0.8, 2017. Chambers, J. M.: XRJulia: Structured Interface to Julia, https://CRAN.R-project.org/package=XRJulia, R package version 0.7.7, 2018.

Bond, N.: hydrostats: Hydrologic Indices for Daily Time Series Data, https://CRAN.R-project.org/package=hydrostats, R package version 0.2.6, 2018.





15

Chang, W. and Borges Ribeiro, B.: shinydashboard: Create Dashboards with 'Shiny', https://CRAN.R-project.org/package=shinydashboard, r package version 0.7.1, 2018.

Chang, W., Cheng, J., Allaire, J., Xie, Y., and McPherson, J.: Shiny: web application framework for R, 2015.

Cheng, J., Karambelkar, B., and Xie, Y.: leaflet: Create Interactive Web Maps with the JavaScript 'Leaflet' Library, https://CRAN.R-project.

5 org/package=leaflet, R package version 2.0.2, 2018.

Clark, M. P., Rupp, D. E., Woods, R. A., Zheng, X., Ibbitt, R. P., Slater, A. G., Schmidt, J., and Uddstrom, M. J.: Hydrological data assimilation with the ensemble Kalman filter: Use of streamflow observations to update states in a distributed hydrological model, Advances in water resources, 31, 1309–1324, https://doi.org/10.1016/j.advwatres.2008.06.005, 2008.

Coles, S.: An Introduction to Statistical Modeling of Extreme Values, Springer Series in Statistics, Springer-Verlag, London, //www.springer.

10 com/gp/book/9781852334598, 2001.

Conway, J., Eddelbuettel, D., Nishiyama, T., Prayaga, S. K., and Tiffin, N.: RPostgreSQL: R Interface to the 'PostgreSQL' Database System, https://CRAN.R-project.org/package=RPostgreSQL, R package version 0.6-2, 2017.

Copernicus Climate Change Service: Climate Data Store, https://cds.climate.copernicus.eu, 2018.

Coron, L., Thirel, G., Delaigue, O., Perrin, C., and Andréassian, V.: The Suite of Lumped GR Hydrological Models in an R package, Environmental Modelling and Software, 94, 166–171, https://doi.org/10.1016/j.envsoft.2017.05.002, 2017.

Coron, L., Perrin, C., Delaigue, O., Thirel, G., and Michel, C.: airGR: Suite of GR Hydrological Models for Precipitation-Runoff Modelling, https://CRAN.R-project.org/package=airGR, R package version 1.0.10.11, 2018.

Council, W. and Heineman, G.: Component-based Software Engineering Putting the Pieces Together, Addison Weysley, 2001.

Csárdi, G.: rhub: Connect to 'R-hub', https://CRAN.R-project.org/package=rhub, R package version 1.0.2, 2017.

20 Dee, D. P., Uppala, S. M., Simmons, A., Berrisford, P., Poli, P., Kobayashi, S., Andrae, U., Balmaseda, M., Balsamo, G., Bauer, d. P., et al.: The ERA-Interim reanalysis: Configuration and performance of the data assimilation system, Quarterly Journal of the royal meteorological society, 137, 553–597, https://doi.org/10.1002/qj.828, 2011.

Delaigue, O., Coron, L., and Brigode, P.: airGRteaching: Teaching Hydrological Modelling with GR (Shiny Interface Included), https: //CRAN.R-project.org/package=airGRteaching, R package version 0.2.3.2, 2018a.

25 Delaigue, O., Thirel, G., Coron, L., and Brigode, P.: airGR and airGRteaching: Two Open-Source Tools for Rainfall-Runoff Modeling and Teaching Hydrology, in: HIC 2018. 13th International Conference on Hydroinformatics, edited by Loggia, G. L., Freni, G., Puleo, V., and Marchis, M. D., vol. 3 of *EPiC Series in Engineering*, pp. 541–548, EasyChair, https://doi.org/10.29007/qsqj, https://easychair.org/ publications/paper/VQ52, 2018b.

des Riviêres, J. and Wiegand, J.: Eclipse: A platform for integrating development tools., IBM Systems Journal, 43, 371-383, 2004.

Do, H. X., Gudmundsson, L., Leonard, M., Westra, S., and Grabs, W.: The Global Streamflow Indices and Metadata Archive (GSIM)-Part
 1: The production of a daily streamflow archive and metadata, Earth System Science Data, 10, https://doi.org/10.5194/essd-10-765-2018, 2018.

Dowle, M. and Srinivasan, A.: data.table: Extension of 'data.frame', https://CRAN.R-project.org/package=data.table, R package version 1.11.6, 2018.

- 35 Eddelbuettel, D. and Balamuta, J. J.: Extending extitR with extitC++: A Brief Introduction to extitRcpp, PeerJ Preprints, 5, e3188v1, https://doi.org/10.7287/peerj.preprints.3188v1, 2017.
 - Emerson, J. W. and Kane, M. J.: biganalytics: Utilities for 'big.matrix' Objects from Package 'bigmemory', https://CRAN.R-project.org/ package=biganalytics, R package version 1.1.14, 2016.





15

35

ESRI: ArcGIS Desktop Release 10.6. Redlands, CA, Environmental Systems Research Institute, ESRI Inc., 2018.

Essawy, B., Goodall, J., Zell, W., Voce, D., Morsy, M., Sadler, J., Yuan, Z., and Malik, T.: Integrating scientific cyberinfrastructures to improve reproducibility in computational hydrology: Example for HydroShare and GeoTrust, Environmental Modelling and Software, 105, 217–229, https://doi.org/10.1016/j.envsoft.2018.03.025, 2018.

5 European Commission: Science with and for Society, http://ec.europa.eu/research/participants/data/ref/h2020/wp/2018-2020/main/ h2020-wp1820-swfs_en.pdf, 2015.

Falcone, J.: GAGES-II: Geospatial Attributes of Gages for Evaluating Streamflow, US Geological Survey., https://doi.org/10.3133/70046617, 2011.

Faria, J., Grosjean, P., and Jelihovschi, E.: Tinn-R-GUI/Editor for R language and environment statistical computing, Disponivel em:<

10 http://sourceforge.net/projects/tinn-r, 2008.

Fuka, DR, Walter, MT, Steenhuis, TS, Easton, and ZM: SWATmodel: A multi-OS implementation of the TAMU SWAT model., https://CRAN.R-project.org/package=SWATmodel, r package version 0.5.9, 2014.

Gautier, L.: rpy2: Python interface to the R language (embedded R), https://pypi.org/project/rpy2, python library version 2.9.5, 2018.

GDAL Development Team: GDAL - Geospatial Data Abstraction Library, Version 2.4.0, Open Source Geospatial Foundation, https://www.gdal.org/, 2018.

Gesmann, M. and de Castillo, D.: googleVis: Interface between R and the Google Visualisation API, The R Journal, 3, 40–44, https://journal. r-project.org/archive/2011-2/RJournal_2011-2_Gesmann+de~Castillo.pdf, 2011.

Gilleland, E. and Katz, R. W.: extRemes 2.0: An Extreme Value Analysis Package in R, Journal of Statistical Software, 72, 1–39, https://doi.org/10.18637/jss.v072.i08, 2016.

20 Giraud, T. and Lambert, N.: cartography: Create and Integrate Maps in your R Workflow, JOSS, 1, https://doi.org/10.21105/joss.00054, 2016. GRASS Development Team: Geographic Resources Analysis Support System (GRASS GIS) Software, Open Source Geospatial Foundation, USA, http://grass.osgeo.org/, 2019.

Graul, C.: leafletR: Interactive Web-Maps Based on the Leaflet JavaScript Library, http://cran.r-project.org/package=leafletR, R package version 0.4-0, 2016.

25 Greenberg, J. A. and Mattiuzzi, M.: gdalUtils: Wrappers for the Geospatial Data Abstraction Library (GDAL) Utilities, https://CRAN. R-project.org/package=gdalUtils, R package version 2.0.1.14, 2018.

Grothendieck, G. and Bellosta, C. J. G.: rJython: R interface to Python via Jython, https://CRAN.R-project.org/package=rJython, r package version 0.0-4, 2012.

- Guo, H.: Big Earth data: A new frontier in Earth and information sciences, Big Earth Data, 1, 4–20,
 https://doi.org/10.1080/20964471.2017.1403062, 2017.
 - Gupta, H. V., Perrin, C., Blöschl, G., Montanari, A., Kumar, R., Clark, M., and Andréassian, V.: Large-sample hydrology: a need to balance depth with breadth, Hydrology and Earth System Sciences, 18, 463–477, https://doi.org/10.5194/hess-18-463-2014, 2014.

Guttag, J. V.: Introduction to computation and programming using Python, Mit Press, 2013.

Hafen, R. and Continuum Analytics, Inc.: rbokeh: R Interface for Bokeh, https://CRAN.R-project.org/package=rbokeh, R package version 0.5.0, 2016.

Hall, D. K. and Riggs, G. A.: MODIS/Terra Snow Cover Daily L3 Global 0.05Deg CMG, Version 6., NASA National Snow and Ice Data Center Distributed Active Archive Center, Boulder, Colorado USA, https://doi.org/10.5067/MODIS/MOD10C1.006, 2016.





- Hannah, D., Demuth, S., van Lanen, H., Looser, U., Prudhomme, C., Rees, G., Stahl, K., and Tallaksen, L.: Large-scale river flow archives: importance, current status and future needs, Hydrological processes, 25, 1191–1200, https://doi.org/10.1002/hyp.7794, 2011.
- Harrigan, S., Hannaford, J., Muchan, K., and Marsh, T. J.: Designation and trend analysis of the updated UK Benchmark Network of river flow stations: The UKBN2 dataset, Hydrology Research, 49, 552–567, 2018.
- 5 Hart, E. M. and Bell, K.: prism: Download data from the Oregon prism project, https://doi.org/10.5281/zenodo.33663, http://github.com/ ropensci/prism, R package version 0.0.6, 2015.
 - Heinkel, R.: pyRserve: A Python client to remotely access the R statistic package via network, https://pypi.org/project/pyRserve, python library version 0.9.1, 2017.
- Hersbach, H., de Rosnay, P., Bell, B., Schepers, D., Simmons, A., Soci, C., Abdalla, S., Alonso-Balmaseda, M., Balsamo, G., Bechtold, P.,
 Berrisford, P., Bidlot, J.-R., de Boisséson, E., Bonavita, M., Browne, P., Buizza, R., Dahlgren, P., Dee, D., Dragani, R., Diamantakis, M.,
 Flemming, J., Forbes, R., Geer, A. J., Haiden, T., Hólm, E., Haimberger, L., Hogan, R., Horányi, A., Janiskova, M., Laloyaux, P., Lopez,
 P., Munoz-Sabater, J., Peubey, C., Radu, R., Richardson, D., Thépaut, J.-N., Vitart, F., Yang, X., Zsótér, E., and Zuo, H.: Operational
 global reanalysis: progress, future directions and synergies with NWP, ERA Report Series, https://www.ecmwf.int/node/18765, 2018.
 Hester, J.: lintr: A 'Linter' for R Code, https://CRAN.R-project.org/package=lintr, R package version 1.0.3, 2018.
- 15 Hijmans, R. J.: raster: Geographic Data Analysis and Modeling, https://CRAN.R-project.org/package=raster, r package version 2.6-7, 2017. Hirsch, R. M. and De Cicco, L. A.: User guide to Exploration and Graphics for RivEr Trends (EGRET) and dataRetrieval: R packages for hydrologic data, chap. A10, U.S. Geological Survey, Reston, VA, https://pubs.usgs.gov/tm/04/a10, 2015. Hufkens, K.: ecmwfr: Interface to the public 'ECMWF' Web Services, https://github.com/khufkens/ecmwfr, R package version 0.0.2, 2018.
- Hufkens, K., Basler, D., Milliman, T., Melaas, E. K., and Richardson, A. D.: An integrated phenology modelling framework in R: modelling
 vegetation phenology with phenor, Methods in Ecology & Evolution, 9, 1–10, https://doi.org/10.1111/2041-210X.12970, 2018.
 Hunt, T.: ModelMetrics: Rapid Calculation of Model Metrics, https://CRAN.R-project.org/package=ModelMetrics, r package version 1.2.2,
 - Hunt, I.: ModelMetrics: Rapid Calculation of Model Metrics, https://CRAN.R-project.org/package=ModelMetrics, r package version 1.2.2, 2018.

Hurley, A.: Processing, modelling and visualizing hydrological data in R (Version 1.1.1) Zenodo, http://doi.org/10.5281/zenodo.2553998, 2018.

- 25 Hutton, C., Wagener, T., Freer, J., Han, D., Duffy, C., and Arheimer, B.: Most computational hydrology is not reproducible, so is it really science?, Water Resources Research, 52, 7548–7555, https://doi.org/10.1002/2016WR019285, 2016.
 - Iannone, R., Allaire, J., and Borges, B.: flexdashboard: R Markdown Format for Flexible Dashboards, https://CRAN.R-project.org/package= flexdashboard, R package version 0.5.1.1, 2018.
 - Ihaka, R.: R: Past and future history, Computing Science and Statistics, 392396, 1998.
- 30 Kahle, D. and Wickham, H.: ggmap: Spatial Visualization with ggplot2, The R Journal, 5, 144–161, http://journal.r-project.org/archive/ 2013-1/kahle-wickham.pdf, 2013.
 - Kane, M. J., Emerson, J., and Weston, S.: Scalable Strategies for Computing with Massive Data, Journal of Statistical Software, 55, 1–19, http://www.jstatsoft.org/v55/i14, 2013.

Kilibarda, M. and Bajat, B.: plotGoogleMaps: the R-based web-mapping tool for thematic spatial data, GEOMATICA, 66, 37-49, 2012.

35 Knaus, J.: snowfall: Easier cluster computing (based on snow)., https://CRAN.R-project.org/package=snowfall, R package version 1.84-6.1, 2015.

Knuth, D. E.: Literate programming, The Computer Journal, 27, 97-111, 1984.





Koffler, D., Gauster, T., and Laaha, G.: lfstat: Calculation of Low Flow Statistics for Daily Stream Flow Data, https://CRAN.R-project.org/ package=lfstat, R package version 0.9.4, 2016.

Kuhn, M., Wing, J., Weston, S., Williams, A., Keefer, C., Engelhardt, A., Cooper, T., Mayer, Z., Kenkel, B., the R Core Team, Benesty, M., Lescarbeau, R., Ziem, A., Scrucca, L., Tang, Y., Candan, C., and Hunt., T.: caret: Classification and Regression Training, https://www.com/actionalescontexpansion/com/actionactionalescontexpansion/com/actionalescontexpansion/co

5 //CRAN.R-project.org/package=caret, R package version 6.0-81, 2018.

Kunst, J.: highcharter: A Wrapper for the 'Highcharts' Library, https://CRAN.R-project.org/package=highcharter, R package version 0.5.0, 2017.

Lawrence, M. and Wickham, H.: plumbr: Mutable and dynamic data models, RStudio, R package, version 0.6.9, http://CRAN.R-project.org/ package=plumbr, 2018.

10 LeDell, E., Gill, N., Aiello, S., Fu, A., Candel, A., Click, C., Kraljevic, T., Nykodym, T., Aboyoun, P., Kurka, M., and Malohlava, M.: h2o: R Interface for 'H2O', https://CRAN.R-project.org/package=h2o, R package version 3.20.0.8, 2018.

Li, C.: JuliaCall: Seamless Integration Between R and 'Julia', https://CRAN.R-project.org/package=JuliaCall, R package version 0.16.0, 2018.

Lovelace, R., Nowosad, J., and Münchow, J.: Geocomputation with R, Chapman and Hall/CRC Press, Boca Raton, London, New York, https://geocompr.robinlovelace.net/, 2019.

Luraschi, J. and Allaire, J.: r2d3: Interface to 'D3' Visualizations, https://CRAN.R-project.org/package=r2d3, R package version 0.2.3, 2018. Luraschi, J., Kuo, K., Ushey, K., Allaire, J., and The Apache Software Foundation: sparklyr: R Interface to Apache Spark, https://CRAN. R-project.org/package=sparklyr, R package version 0.9.3, 2018.

MathWorks: Company Overview, https://www.mathworks.com/content/dam/mathworks/tag-team/Objects/c/company-fact-sheet-8282v18.

20 pdf, 2018.

25

15

McKiernan, E., Bourne, P., Brown, C., Buck, S., Kenall, A., Lin, J., McDougall, D., Nosek, B., Ram, K., Soderberg, C., Spies, J., Thaney, K., Updegrove, A., Woo, K., and Yarkoni, T.: How open science helps researchers succeed, eLife, 5, https://doi.org/10.7554/eLife.16800, 2016.

Metcalfe, P., Beven, K., and Freer, J.: dynatopmodel: Implementation of the Dynamic TOPMODEL Hydrological Model, https://CRAN. R-project.org/package=dynatopmodel, R package version 1.2.1, 2018.

Microsoft and Weston, S.: foreach: Provides Foreach Looping Construct for R, https://CRAN.R-project.org/package=foreach, R package version 1.4.4, 2017.

30 Microsoft Corporation and Weston, S.: doParallel: Foreach Parallel Adaptor for the 'parallel' Package, https://CRAN.R-project.org/package= doParallel, R package version 1.0.14, 2018.

Muenchow, J., Schratz, P., and Brenning, A.: RQGIS: Integrating R with QGIS for Statistical Geocomputing, The R Journal, 9, 409–428, https://rjournal.github.io/archive/2017/RJ-2017-067/RJ-2017-067.pdf, accepted for publication on 2017-12-04, 2017.

Müller, K., Wickham, H., James, D. A., and Falcon, S.: RSQLite: 'SQLite' Interface for R, https://CRAN.R-project.org/package=RSQLite,

35 R package version 2.1.1, 2018.

Neil, D.: Modern Vim: Craft Your Development Environment with Vim 8 and Neovim, Pragmatic Bookshelf, 2018.

Ooms, J.: The OpenCPU system: Towards a universal interface for scientific computing through separation of concerns, arXiv preprint arXiv:1406.4806, 2014.

Melsen, L., Torfs, P., Uijlenhoet, R., and Teuling, A.: Comment on "Most computational hydrology is not reproducible, so is it really science?" by Christopher Hutton et al., Water Resources Research, 53, 2568–2569, https://doi.org/10.1002/2016WR020208, 2017.





Ooms, J.: V8: Embedded JavaScript Engine for R, https://CRAN.R-project.org/package=V8, 2017. Openshaw, S. and Abrahart, R. J.: GeoComputation, vol. 24, Taylor & Francis, London, New York, 2000. Pebesma, E.: Simple Features for R: Standardized Support for Spatial Vector Data, The R Journal, https://journal.r-project.org/archive/2018/ RJ-2018-009/index.html, 2018a.

5 Pebesma, E.: stars: Scalable, Spatiotemporal Tidy Arrays, https://CRAN.R-project.org/package=stars, R package version 0.2-0, 2018b. Pebesma, E. and Bivand, R.: Classes and methods for spatial data in R, R News, 5, 9–13, https://CRAN.R-project.org/doc/Rnews, 2005. Pedersen, T. L.: gganimate: Create easy animations with ggplot2, https://github.com/thomasp85/gganimate, 2018. Perkel, J. M.: A toolkit for data transparency takes shape, Nature, 560, 513–515, https://doi.org/10.1038/d41586-018-05990-5, 2018.

Pierce, D.: ncdf4: Interface to Unidata netCDF (Version 4 or Earlier) Format Data Files, https://CRAN.R-project.org/package=ncdf4, R
package version 1.16, 2017.

Pohlert, T.: trend: Non-Parametric Trend Tests and Change-Point Detection, https://CRAN.R-project.org/package=trend, R package version 1.1.1, 2018.

QGIS Development Team: QGIS Geographic Information System, Open Source Geospatial Foundation Project, http://qgis.osgeo.org, 2018. R Core Team: R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, http:

- 15 //www.R-project.org, 2018.
 - Rigby, R. A. and Stasinopoulos, D. M.: Generalized additive models for location, scale and shape,(with discussion), Applied Statistics, 54, 507–554, 2005.

Ritchie, D. M.: The development of the C language, ACM Sigplan Notices, 28, 201-208, 1993.

Rödiger, S., Friedrichsmeier, T., Kapat, P., Michalke, M., et al.: RKWard: a comprehensive graphical user interface and integrated develop-

ment environment for statistical analysis with R, Journal of Statistical Software, 49, 1–34, 2012.
 Rossini, A. J., Heiberger, R. M., Sparapani, R. A., Maechler, M., and Hornik, K.: Emacs speaks statistics: A multiplatform, multipackage development environment for statistical analysis, Journal of Computational and Graphical Statistics, 13, 247–261, 2004.
 RStudio Team: RStudio: Integrated Development Environment for R, RStudio, Inc., Boston, MA, http://www.rstudio.com/, 2018.
 Ryberg, K. R. and Vecchia, A. V.: waterData: Retrieval, Analysis, and Anomaly Calculation of Daily Hydrologic Time Series Data, https:

- 25 //CRAN.R-project.org/package=waterData, R package version 1.0.8, 2017. SAGA Development Team: System for Automated Geoscientific Analyses (SAGA GIS), Germany, http://www.saga-gis.org/, 2008. Sanner, M. F. et al.: Python: a programming language for software integration and development, J Mol Graph Model, 17, 57–61, 1999. Sarkar, D.: Lattice: Multivariate Data Visualization with R, Use R!, Springer, New York, http://Imdvr.r-forge.r-project.org, 2008. Sievert, C.: plotly for R, https://plotly-book.cpsievert.me, 2018.
- 30 Slater, L., Harrigan, S., Gauster, T., Hurley, A., Thirel, G., and Vitolo, C.: Using R in Hydrology, EGU2018 Short Course. Zenodo, http: //doi.org/10.5281/zenodo.2554009, 2018.

Slater, L. J., Singer, M. B., and Kirchner, J. W.: Hydrologic versus geomorphic drivers of trends in flood hazard, Geophysical Research Letters, 42, 370–376, https://doi.org/10.1002/2014GL062482, 2015.

Sommerlot, A., Fuka, D., and Easton, Z.: getMet: Get Meteorological Data for Hydrologic Models, https://CRAN.R-project.org/package=

35 getMet, R package version 0.3.2, 2016.

Souza, R.: Ecohydmod: Ecohydrological Modelling, https://CRAN.R-project.org/package=Ecohydmod, R package version 1.0.0, 2017.

Tapley, B. D., Bettadpur, S., Ries, J. C., Thompson, P. F., and Watkins, M. M.: GRACE measurements of mass variability in the Earth system, Science, 305, 503–505, https://doi.org/10.1126/science.1099192, 2004.





10

Tennekes, M.: tmap: Thematic Maps in R, Journal of Statistical Software, 84, 1–39, https://doi.org/10.18637/jss.v084.i06, 2018.

Thieurmel, B., Marcelionis, A., Petit, J., Salette, E., and Robert, T.: rAmCharts: JavaScript Charts Tool, https://CRAN.R-project.org/ package=rAmCharts, R package version 2.1.9, 2018.

Tuszynski, J.: caTools: Tools: moving window statistics, GIF, Base64, ROC AUC, etc., https://CRAN.R-project.org/package=caTools, R

5 package version 1.17.1.1, 2018.

Urbanek, S.: proj4: A simple interface to the PROJ.4 cartographic projections library, https://CRAN.R-project.org/package=proj4, R package version 1.0-8, 2012.

Urbanek, S.: rJava: Low-Level R to Java Interface, https://CRAN.R-project.org/package=rJava, R package version 0.9-10, 2018.

Vanderkam, D., Allaire, J., Owen, J., Gromer, D., and Thieurmel, B.: dygraphs: Interface to 'Dygraphs' Interactive Time Series Charting Library, https://CRAN.R-project.org/package=dygraphs, R package version 1.1.1.6, 2018.

Vantas, K.: hydroscoper: R interface to the Greek National Data Bank for Hydrological and Meteorological Information, The Journal of Open Source Software, 2, https://doi.org/10.21105/joss.00625, 2018.

Viglione, A.: nsRFA: Non-Supervised Regional Frequency Analysis, https://CRAN.R-project.org/package=nsRFA, r package version 0.7-14, 2018.

15 Viglione, A. and Parajka, J.: TUWmodel: Lumped Hydrological Model for Education Purposes. R package version 0.1-4, 2014. Villarini, G., Serinaldi, F., Smith, J. A., and Krajewski, W. F.: On the stationarity of annual flood peaks in the continental United States during the 20th century, Water Resources Research, 45, 2009.

Vitolo, C.: hddtools: Hydrological Data Discovery Tools, The Journal of Open Source Software, 2, https://doi.org/10.21105/joss.00056, 2017.

20 Vitolo, C., Fry, M., and Buytaert, W.: rnrfa: An R package to Retrieve, Filter and Visualize Data from the UK National River Flow Archive, The R Journal, 8, 102–116, https://journal.r-project.org/archive/2016/RJ-2016-036, 2016a.

Vitolo, C., Wells, P., Dobias, M., and Buytaert, W.: fuse: An R package for ensemble Hydrological Modelling, The Journal of Open Source Software, 1, https://doi.org/10.21105/joss.00052, 2016b.

Wahlbrink, S.: Eclipse Plug-In for R: StatET [Software], 2016.

Wendt, N.: gribr: R interface for GRIB files using ECMWF ecCodes, https://github.com/nawendt/gribr, r package version 1.2.2, 2018.
 Wickham, H.: testthat: Get Started with Testing, The R Journal, 3, 5–10, https://journal.r-project.org/archive/2011-1/RJournal_2011-1_Wickham.pdf, 2011.

Wickham, H.: R packages: organize, test, document, and share your code, O'Reilly Media, Sebastopol, http://r-pkgs.had.co.nz, 2015.

Wickham, H.: feather: R Bindings to the Feather 'API', https://CRAN.R-project.org/package=feather, R package version 0.3.1, 2016a.

- 30 Wickham, H.: ggplot2: Elegant Graphics for Data Analysis, Use R!, Springer, New York, http://ggplot2.org, 2016b.
 - Wickham, H.: tidyverse: Easily Install and Load the 'Tidyverse', https://CRAN.R-project.org/package=tidyverse, R package version 1.2.1, 2017.

Wickham, H. and Hesselberth, J.: pkgdown: Make Static HTML Documentation for a Package, https://CRAN.R-project.org/package= pkgdown, R package version 1.3.0, 2018.

35 Wickham, H., Hester, J., and Francois, R.: readr: Read Rectangular Text Data, https://CRAN.R-project.org/package=readr, r package version 1.3.1, 2018.

Wilkinson, L.: The Grammar of Graphics, Statistics and Computing, Springer, New York, first edn., 1999.





5

10

Wilson, G.: Software carpentry: getting scientists to write better code by making them more productive, Computing in Science & Engineering, 8, 66–69, 2006.
Wilson, G.: Software Carpentry: lessons learned, F1000Research, 3, https://software-carpentry.org/lessons, 2014.
Winslow, L. A., Hahn, T. H., Princiotta, S. D., Leach, T. H., and Rose, K. C.: hydrolinks: A new tool to link macroscale to inland water bodies, In Prep/Review, In Prep.
WMO: Manual on low flow estimation and prediction, 2009.
Wood, S. N.: Generalized additive models: an introduction with R, Texts in statistical science, Chapman and Hall/CRC Press, Boca Raton, London, New York, second edn., 2017.
Xie, Y.: bookdown: Authoring Books and Technical Documents with R Markdown, The R series, Chapman and Hall/CRC Press, Boca Raton, London, New York, https://github.com/rstudio/bookdown, 2016.
Xie, Y.: knitr: A General-Purpose Package for Dynamic Report Generation in R, https://yihui.name/knitr, R package version 1.20, 2018.
Xie, Y. and Lesur, R.: pagedown: Paginate the HTML Output of R Markdown with CSS for Print, https://CRAN.R-project.org/package= pagedown, r package version 0.1, 2019.

Xie, Y., Hill, A. P., and Thomas, A.: blogdown: Creating Websites with R Markdown, The R series, Chapman and Hall/CRC Press, Boca

- 15 Raton, London, New York, https://github.com/rstudio/blogdown, 2017.
 - Yee, T. W., Stoklosa, J., and Huggins, R. M.: The VGAM Package for Capture-Recapture Data Using the Conditional Likelihood, Journal of Statistical Software, 65, 1–33, https://doi.org/10.18637/jss.v065.i05, 2015.
 - Zambrano-Bigiarini, M.: hydroGOF: Goodness-of-fit functions for comparison of simulated and observed hydrological time series, https://doi.org/10.5281/zenodo.840087, http://hzambran.github.io/hydroGOF, R package version 0.3-10, 2017a.
- 20 Zambrano-Bigiarini, M.: hydroTSM: Time Series Management, Analysis and Interpolation for Hydrological Modelling, https://doi.org/10.5281/zenodo.83964, https://github.com/hzambran/hydroTSM, R package version 0.5-1, 2017b.