

Response to Reviewers

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

Previous responses from the online discussion are in blue; new responses after revision are in red.

Response to Editor

5 Editor Decision: Publish subject to revisions (further review by editor and referees) (24 Apr 2019) by Erwin Zehe

Comments to the Author:

Dear Dr. Slater,

After reading the reviews and your manuscript again, I share the generally very positive attitude of the reviewers. Overall I agree with Michael Stoezle that the manuscript would gain even more strength, when putting more emphasis on discussing

10 *best practice of using R code in hydrological research. This will be particularly valuable for R beginners.*

Best regards,

Erwin Zehe

We would like to thank the Editor for his positive comments on our manuscript. We have incorporated the vast majority of the Reviewers' suggestions. Throughout the manuscript, we have emphasized our discussion of best practice in using R code
15 for hydrological research. For example, we have discussed: setting up repositories and committing code online; following a standard "hydrological workflow" in R; sharing R code via the "literate programming paradigm", blog posts, tutorials; and designing hydrological vignettes to facilitate the uptake of R packages.

Response to Reviewer 1

20

We would like to thank Reviewer 1 for constructive comments on our manuscript. We feel that many of these comments are helpful and will improve the quality of our review paper. Below we provide the Reviewer's comments verbatim in black italic text, and our responses are immediately below each comment in blue text.

25 *"This manuscript represents a substantial contribution to scientific progress in hydrology in that it makes readers aware of the new task view in R for Hydrological Data and Modeling. Because there are almost 14,000 packages available in R and countless more in other repositories, it is difficult for analysts to keep up on packages potentially useful to them. Having a*

hydrology Task View is an important maturation in the use of R in the hydrology community, that many might not be aware of for quite some time without some targeted messaging to the hydrology community"

We thank the reviewer for the positive comments on the manuscript, and we agree that it was important to highlight the new Task View as part of this review paper.

5

"The discussion of many packages available to hydrologists is good. While the authors cannot discuss them all, they have a fitting selection of useful packages. For example, an analyst may be aware of the trend analysis functions in R in multiple packages, but not the spatial analysis/GIS functionality, and this article helps bring together diverse analysis methods in one discussion of the hydrology community.

10 Thank you. Yes, indeed, one aim of the paper was to provide a broad overview of all the different tools and techniques that are available to hydrologists in R.

"The mention of the traceability of package authors through their ORCIDs is another important, relatively recent addition to R. As an active member of the R community, I was unaware of this and found that many packages do not use this yet. However, this is another way to judge the background of package developers and find related articles excellent information to provide to the R Hydro community."

15

Yes, absolutely; we agree that adding ORCIDs to packages is a useful recent development in R!

"The scientific quality of this article is excellent, providing a brief history of software languages used in hydrology and a brief history of R, while making good arguments for the use of R in hydrology. Reproducible research benefits all scientific work and is an area with continued need for improvement in hydrology can help."

20

Thank you for the positive comments!

"The figures and graphics are appropriate for the article and are informative. The paper is well written and carefully edited prior to review (much appreciated by reviewers)."

25

Again, we thank the reviewer for the positive comments on the manuscript.

" The scientific resources listed were good. It could have even been better if some of the authors were from the US and able to describe additional developments, providing a more global view."

30

We appreciate the reviewer's opinion. The authors of this manuscript are from the UK, France, Italy, and Morocco; it originated from past contributions to the European Geosciences Union short course "Using R in Hydrology" in Vienna, which may explain why the authorship is mostly euro-centric. However, we have tried to provide a balanced, objective perspective throughout the manuscript, and we welcome suggestions if we have missed any additional developments in the manuscript.

"A suggested improvement to the article is in the inclusion of the useR! Conference (URL provided) in the discussion of scientific resources and courses. These annual conferences are not specific to hydrology and earth sciences, but usually have some environmental science or ecology sessions as well as sessions or short courses of general interest on learning R, best practices, spatial analysis and statistical methodology incorporated in various R packages."

- 5 Thank you, we will mention this conference in the manuscript. Many of these R conferences (RStudio, satuRdays or eRum) now stream the R talks and/or make the resources available after the event. We can certainly learn from these approaches too.

Added in section 5.6.

"Another suggested improvement is more discussion on setting up repositories. What are a few good options for hydrologists? Many subscribe to the workflow provided, but are still somewhat mystified by the repository step, that is actually rather easy once someone clears the initial hurdles of selection and setup."

- 10 This is a very useful comment. We entirely agree and will provide a succinct overview of how to set up a repository. We also link to the Software Carpentry course which provides valuable suggestions on how to set up and structure a project repository.

Added in section 3.1.

15

"On page 14, lines 8 and 9, the "(Mann-)Kendall testing" punctuation is a bit awkward, exacerbated by the line break in it. I suggest leaving it as Mann-Kendall testing because readers will know what that is referring to whether they know it as Kendall or Mann-Kendall."

Yes, thank you, we will alter this accordingly.

- 20 Done.

Response to Reviewer 2

We would like to thank Reviewer 2 for constructive comments on our manuscript. The comments are helpful and will certainly

25 improve the quality of our manuscript. Below we provide the Reviewer's comments verbatim in black italic text, and our responses are immediately below each comment in blue text.

"I enjoyed reading this paper, which is for sure useful to get an overview on what R can do for hydrologists and what hydrologists can do for R. I am an R user, and I have also developed packages, which places me at one side of the spectrum of potential readers of this paper. As such, I can benefit a lot from the information provided, especially regarding tools on subjects I am not an expert in. I wonder whether the great amount of information provided in the paper isn't overwhelming for an R beginner. It is unlikely that a R beginner will comment the paper in this discussion stage so, if I were one of the Authors, I would suggest to ask, e.g., a couple of master or starting PhD students to perform an internal review of the paper as non-expert readers (that's just an idea)."

30

We thank the Reviewer for their positive comments. We find that the comment about the diverse readership of this manuscript (in terms of beginners vs. experts) is very relevant. We have thus shared the manuscript with one MSc and one PhD candidate to ask for their feedback and they both said they found the paper straightforward to follow; in fact both mentioned that the workflow figure was useful. They also raised some minor points that we will use to improve the revised version of the paper.

- 5 The revised manuscript will aim to make it as easy as possible for hydrologists to navigate R and the incredible amount of R resources that exist.

"Besides the many advantages of using R (and open data and software in general), I would have expected a mention on possible drawbacks. A couple of examples follow:

- 10 - *The easy accessibility and usability of open tools results, sometimes, in applications without critical scrutiny on what one does. It happened to me to receive an email from a user of my package asking me to tell him/her how to apply the methods to her/his data. A few months later I've been asked to review the paper written by the same R user who, then I realised, had no clue about the meaning of the methods applied. In other words, how can the misuse of software be reduced within the open source community? (I am aware that this happens also with proprietary software)."*

- 15 We feel that the misuse of software is a legitimate concern but it is indeed equally applicable to any proprietary software, as the Reviewer mentions. Recognising that it is likely beyond the scope of our paper to provide real solutions to software misuse, we will nonetheless highlight in the manuscript that one solution is to make the documentation as clear as possible. We will encourage expert users to contribute vignettes and documentation to their packages, or to write blog-posts or tutorial-style papers which explain how to correctly implement a specific method. Ensuring that use-cases are perceived as a useful
20 contribution to software and the scientific discourse is a vital step forward.

Done in section 4.1.

- "- Everybody can make an R package available, even if not carefully tested. This may result in errors that are then propagated by users. That happened to me too. I had a bug in my code but, thanks to the open source, a user spotted the error and I could
25 correct it. When my package was first published, I received several feedback messages that helped me a lot to clean the code. The frequency has then decreased considerably meaning that, hopefully, the main errors have been solved (or that people do not use the package any more :-)"*

Having said this, I believe the paper is good and worth publishing in HESS, maybe after a minor revision."

- Thank you for the positive comment. We entirely agree with the Reviewer's point of view and will mention in our revised
30 manuscript both the pitfalls (risk of errors in new packages) and strengths (community review) of the open source approach. We will also place emphasis on the importance of including tests when developing an R package.

Done in section 5.1.

"Small things:

Page 2, line 31: I would mention here more specifically what R is, or better what R was, i.e., software for statistics (you say that in section 2.3). This is to better inform the non-expert reader. "

Thank you - yes indeed. We will include this information.

Done in the introduction.

5

"Figure 1: I am not a big fan of this kind of figures in review papers. The text describing it could be sufficient. "

We appreciate the reviewer's opinion but we feel this figure helps give a sense of the growth of R, providing more information than the text alone.

No change made.

10

"Page 5, line 18: I would say here that hydrologic research is criticized for the lack of reproducibility, rather than scientific research in general."

We are not sure that hydrology is worse (or better) than many other fields, however the original reference does indeed refer to computational hydrology specifically. We will thus provide a broader discussion about about lack of reproducibility, e.g.

15 <https://www.nature.com/collections/prbfkwmwvz> or <https://doi.org/10.1080/13697137.2018.1476968>.

After careful consideration, we find that this paragraph already provides a lengthy and sufficient discussion of reproducibility, and therefore we made minor changes only.

"Page 7, line 14: Maybe it is just the way I call things, but I would say that C and Fortran are very efficient in performing loop
20 tasks. R, Matlab etc. are very good in vector operations."

We tend to agree with the Reviewer: that is true in most cases (and is the reason why the apply functions loops are coded in C). We will remove the expression "when dealing with vector operations" from our initial sentence.

Done in section 2.4.

25 "Section 2.5: Just as a note, which may be ignored by the authors if not relevant, IAHS also organises R courses in its meetings, e.g., there will be a "Using R in Hydrology" workshop at the IUGG2019 conference in Montreal. Also, among others, the PUB summer course (URL provided) uses R intensively."

Thank you. We were not aware of these other R sessions but will definitely add this to the manuscript!

Done in section 2.5.

30

"Page 10, line 16: I would not use Twitter to discover packages, but to be exposed to communications about new packages or new features. "

Yes, we agree with this comment and will alter the text accordingly.

Done.

35

"Page 12, line 24: I normally use RgoogleMaps (URL provided) to interact with Google Maps."

Thank you, we were not aware of this but will mention it in the text!

Done.

5 "Page 13, line 15: Actually also TUWmodel can be used in a semi-distributed fashion, allowing for differentiating into elevation zones."

Good to know; we will include this information in the revised text. Although the functions describe a lumped model, we noticed in the examples that a semi-distributed application can be implemented. We will rephrase that sentence in the manuscript.

Done.

10

"Page 14, line 19: For regional frequency analysis I would also mention the lmomRFA package (URL provided) developed by the very father of the L-moment methodology J. R. M. Hosking."

Thank you, we will certainly include the lmomRFA package, as well as lmom, since Lmoments are important in hydrology.

Done.

15

Response to Reviewer 3

We would like to thank Michael Stoelzle for this constructive review and for his positive comments on our manuscript. We look forward to implementing these suggestions which will certainly improve the paper. Below we provide Michael's comments verbatim in italic black text.

25 *First of all, I really support the idea of the paper and I think the presented review is fully in the scope of HESS. I like how the relationship between hydrology and R is presented and the authors gave a broad overview of historical and recent developments in the RHydro community and sketch a interesting roadmap for the future of R in hydrology. The paper is from my perspective also a good example for a joint effort from young scientists in our hydrological community. However, since the first two reviews are full of praise I try to be a little bit more critical putting together some thoughts that hopefully improve the paper. My major "overall" suggestion is to revise the paper going a little bit away from just having a list of available packages or presenting potential possibilities in the R-Hydro universe towards a) more "best practices" for certain challenges and b) more comparisons of similar packages. However, I absolutely recommend to publish this paper in HESS. See more specific comments below.*

30 *Best regards, Michael*

Thank you for these positive comments on our manuscript. We are pleased to see this work described as a "good example of joint effort from young scientists in the community", and we appreciate the thoughtfulness in these constructive suggestions.

We will certainly revise the manuscript by emphasizing best practice and comparison of package functionalities, as suggested.

35 We have attempted to emphasize best practice and compare package functionalities throughout the manuscript.

Major comments

The authors give a lot of technical information about R in different section in the paper (e.g. GNU S background in sect 2.3, R script workflow in Sect. 2.1, R output format in the Introduction). As all the parts are really important to learn about the
5 fundamental principles of R for me it would be more helpful to have the technical and historical information about R in one separate section. This might be also helpful for readers who are new to R coming from other programming languages.

We agree it would be a good idea to move some of these elements into one separate paragraph in the introduction. However we feel that creating an entire section on the history of R might be a little misplaced, so instead we suggest to include a few general references in the introductory paragraph, such as (for example) <https://cran.r-project.org/doc/manuals/r-release/R-lang.html> or
10 https://cran.r-project.org/doc/FAQ/R-FAQ.html#What-is-R_003f

We have added references in this paragraph.

I miss more discussion about different philosophies how to program in R. From my perspective there is, for example, often the question whether you are doing your visualizations with ggplot2 or in base-R or doing data wrangling with dplyr or in
15 a complete different “style” with data.table. I don’t know if there are comparable examples for hydrological packages (doing more or less the same thing but with different techniques / packages). The paper could, however, be improved by presenting opposed approaches to solve problems in R. For example, data.table is known to be very fast in data wrangling but on the other side the dplyr package (and piping %>%) is known to support highly readable code. Here the trade-off between performance and communication could be discussed (i.e. Sect 4.3 and Sect. 2.1/2.2). My experience from teaching R in hydrology is that
20 students really like the concept of piping (%>% read as “and then do that”) and can do the step from base-R to tidyverse in a couple of sessions. Tidyverse and Pipes could also be mentioned in Sect. 5.3 as a good possibility to teach “highly readable code” in the classroom.

We agree that the different approaches to R programming (e.g. base-R versus the tidyverse) are definitely worth mentioning. It is often difficult to decide between different packages that offer similar options (e.g. many different packages for parallel
25 computing). The choice of approach is often quite a personal matter (i.e. depends on the individual) and many people often mix both types of approaches. Regarding which approach is easiest for beginners, we note that while the tidyverse style might seem easier at first, as soon as the analysis becomes more complex, dplyr on his own won’t suffice. For example, to handle out-of memory dplyr needs to be connected to a database, or you need to switch to bigmemory/data.table. Thus in many cases, using base-R is safer (e.g. for operational purposes, or for creating and maintaining packages) as it avoids issues with dependencies
30 and updates. Overall, we entirely agree that it is worth discussing these issues in the manuscript and providing a more explicit comparison of the different approaches - thank you for raising this point!

After considerable reflection on this point, and although we agree with Michael that the specific philosophies of R are interesting and valuable, we have come to the conclusion that they are beyond the scope of this manuscript, which aims to focus on the role of R in hydrology rather than the general use of R. Therefore, in the end we did not include this discussion, because
35 we felt it would dilute the manuscript.

Regarding Sect 2.2 it would be helpful to draft a way forward how developers of R-Hydro packages can develop better R packages with a clear description of input data types, a comprehensive documentation on package usage. For example, tidyverse is set of packages that work well together as all packages use the same data representations and API design. Is this also a valuable approach for, let's say, developer of R packages that include hydrological models? For example, is it possible to use the data retrieval packages from Table 1 in a coherent way or is the syntax between all packages different? Is there an effort to combine data retrieval packages from different regions with a consistent R syntax/usage? Using different hydrological R packages with the same syntax might be of great value for the community. A short discussion about this issue can draft a way forward in the R Hydrology community (as mentioned in Sect 4.1 where the need of good vignettes in R is highlighted).

Yes, we agree with the points made here. The existing hydrological data retrieval packages (e.g. rnrf, dataRetrieval...) are structured differently because they were set up independently, and because the underlying hydrological and hydrometric datasets differ from country to country. It is therefore difficult to coordinate hydrological data amongst regions. As more nations develop similar hydrological data acquisition packages in the future, we believe it would be worth implementing a common syntax and data output form. For example, it would be ideal to use consistent APIs and consistent output objects across packages. We will make some recommendations for future developers of these hydrological packages in the manuscript. Additionally, it is worth mentioning that there is currently no effort to combine data retrieval packages from different regions but this is certainly worth doing. Perhaps the World Meteorological Organization's Hydrological Observation System (WHOS) might consider implementing (<http://www.wmo.int/pages/prog/hwrp/chy/whos/index.php>) something like this - for example, (i) a meta-package with functions that convert hydrometric data from other packages to a standard format, or (ii) a meta-package for all hydrological models to be run within the same framework.

We have included this discussion in sections 5.1-5.3., including a discussion of data standards (e.g. Open Geospatial Consortium WaterML)

The introduction of section 3 and also Fig. 3 are really nice and well-conceived. I like the idea to split the data analysis into different parts and subsections in the paper (Sect. 3.x).

Thank you!

In general, the paper is full of valuable package, links, and corresponding citations. However, from my perspective often only the package names are given and more detailed information how the package is working or why it is especially useful/important for the hydrological community is missing. Of course, it is not possible to describe all the functionality of all the packages in detail but the authors could pick 2-3 cases where they really compare the advantages and disadvantages of using different packages for the same purpose / challenge. To be honest (and I also think that the author team is aware of that), you can relatively easy check on the internet if there is a package for a certain task and what are the technical details or data requirements of a single package. So, the great list of available packages cannot be the main message of this compelling paper! Comparing different packages among each other could really be of great value for the hydrological community and this paper is a good

place to do that. A good example is Sect 3.4 describing spatial data analysis and visualization in R. In this section a lot of important R packages are mentioned but the reader gets less information which package is better or has the same functionality as other packages etc. In Sect 3.5 the information about snow functionality in the hydrological models is a good example how this can be done.

- 5 Yes, it is true that in some cases we were quite brief - there is so much to say that this paper could be written as a book! However, we agree that it is worth comparing the advantages/disadvantages of similar packages when we revise the manuscript. We will provide some information about advantages of specific packages within the different sections, similarly to the snow functionality example.

- We have included discussion of the merits of specific packages at various points throughout the manuscript (for example, a comparison between `mgcv` and `gamlss` in the statistical section, or the discussion of different statistical modelling packages in the models section. We are unable to do this for all packages because this is already a very long manuscript and would turn into a book if we did!

- For me a major deficiency of the paper is that the topic “colours” is completely missing. Hydrology is a multi-faceted discipline and hydrologists doing a lot of visualizations for posters, presentations and papers. Hydrology is also often based on multi-dimensional analyses with different variables in space and time. To visualize hydrological processes and hydrological change colour is often the first choice to compare data sets and to investigate relationships. As we all know, a sophisticated colour choice is often a huge challenge. I encourage the authors to implement a “colour section” where a short discussion about, for example, the basics of appropriate colour gradients for graphs and maps is given (one-colour gradient, two-colours gradient, discrete vs. continuous variables). In large-scale hydrology “rainbow” colour gradients are still state-of-the-art (to generate fancy and colorful world maps) but a lot of papers and blogpost have taught us that the “rainbow” isn’t perceptual uniform and not colourblind-safe.

- Indeed, there is a lot to say about data visualization - we agree this would be a useful addition. It is widely accepted that the rainbow color scheme is a poor choice for data visualization (e.g. <https://betterfigures.org/>, <https://www.climate-lab-book.ac.uk/2014/end-of-the-rainbow/>, or a recent R-bloggers post on this topic - <https://www.r-bloggers.com/at-the-end-of-the-rainbow/>). We will mention that R is strong in this department and will include a short section about colours, color-blind friendly palettes, and appropriate choices in hydrology. We will also mention some useful papers on this subject, e.g. ?.

Done in section 3.7.

- 30 *Minor comments*

P2L20 Give 1-2 examples of hydrological models coded in R.

Yes, we will do this.

Done. We discussed TOPMODEL, FUSE, TUWmodel, and various other models in section 3.5 rather than in the introduction.

P2L33-35 *It might be helpful for the reader to compare the fundamental principles of R against other programming languages (e.g. C, Java) to better describe the architecture of R (e.g. in terms of object-orientation, compilation of code*

Yes, we will include a short description of R's principles (as an object-oriented programming and interpreted language) in the technical section mentioned above.

5 Done in section 2.1. instead.

P3L4 *Are there any reasons for the large number of updates in R-Hydro packages (related to Fig.1)?*

We actually had a lengthy discussion about this among ourselves. One possibility is that the increase in updates in 2018 is related to documentation requirements on CRAN, as a couple of features that were okay before had to be modified (possibly
10 due to package dependencies). Alternatively it is possible that the rise in package updates was linked to the development and uptake of R-HUB (a web service that allows developers to test and debug R-packages on different operating systems to reproduce what CRAN does). We were not entirely sure of the answer and so we refrained from mentioning this in the manuscript.

No action taken.

15

P4L11 *The platform Stackoverflow (from Sect. 2.5) can also be mentioned here as there often also information on relevant R packages are given.*

Yes, we will do this.

Done in section 2.5. We were careful to avoid redundancies in the manuscript.

20

P5L31 *What is meant with the "standardized format"? Here it might be also valuable to add packages like roxygen2 that help to develop, create and document R packages.*

The sentence referred to is "Relying on well-established publishing platforms such as CRAN and GitHub has promoted a standardized format for developing and disseminating R code." Here, we were referring to the best practice in writing R code,

25 and will clarify this in the text. We will include the roxygen2 package.

Done.

P6L27 *It will be easy to find some justification for this statement (e.g. number of R topics on stackoverflow.com compared to other programming languages).*

30 The statement referred to is "R is still considered the most powerful language and environment for statistical computing". We will justify the statement in our revised manuscript.

Done.

P6L3-14 *The paper would be improved by adding various possibilities to share and publish code during and after paper
35 publication (e.g. github repository, gists or other tools and platforms). Is there a possibility to share R code with a DOI?*

We agree, this is a good idea. We will mention different options and platforms for sharing R code with a DOI such as GitHub, Zenodo, Figshare, and R Pubs/Plotly (also for dashboards and interactive plots). Another important data repository to mention is <https://www.pangaea.de/>, because it is tailored to Earth and Environmental Sciences. Some of these platforms are discussed in the journal *Geoscientific Model Development* https://www.geoscientific-model-development.net/about/code_and_data_policy.

5 [html](#)

Done (new section 5.1).

Sect 2.3 For me this section contributes less to the strength of the paper – what is the key message here? The specific statistical packages that are relevant for R-Hydro are given in Sect 3.6 (as mentioned). The link to CRAN is perhaps too unspecific here.

10 *Yes, R provides a vast number of statistical tools for hydrologists but then you have to explain in more detail why and what advantages R here has to offer (compared to other languages?).*

Yes, there is indeed some overlap between sections. We will remove this opening section (2.3. *Providing statistical tools for hydrology*) and strengthen section 3.6 (*Packages for hydrological statistics*) instead. We will move the description of CRAN to the new suggested section that provides a background on R.

15 *In the end we decided that there is merit in keeping the brief discussion of the importance of statistics in R in the opening sections of the manuscript, and keeping it separate from the discussion of statistical packages later in the manuscript. We did remove the description of CRAN.*

Sect 2.5 What are the rotating topics and why have they been chosen? The picture might be nice to see but the space could instead better be used to highlight the EHGU SC topics from the last years with a short information why the topics were of great interest for the community. It might be also important to highlight the possibility to raise issues for certain R packages on github.com to foster an online-documented discussion with the package developers (i.e. interaction in the community). At least a cross reference to Sect. 5.4 could be made here.

25 *The section Michael is referring to (Section 2.5.) is the section on *scientific resources and courses*. We will keep the picture and will include a list of the topics too, explaining why they are of interest. We also agree that it is worth highlighting in this section the ability to raise issues with the developers on GitHub (provided packages are developed/published on GitHub). We will encourage publication of packages on GitHub and will mention that the bugs report link can be added in the Description file so that package authors can specify how they prefer to receive bug reports.*

30 *We added the cross-reference to 5.6. Raising issues is mentioned in section 5.2. The short course topics are already listed in 5.6.*

Table 1 I am not sure on which basis you have choose the 11 packages? Are they the most important ones? Perhaps a thematic order would improve the table (sort by data type, spatial extend, multiple proposes or not, . . .)

35 *Yes, we agree that ordering these packages by data type (and perhaps providing a bit more information about the data) would be worthwhile.*

Done.

P11L5-9 Consider to mention the rio package here as it is really powerful. The tidyverse package is cited as a package that reads multiple file formats – I think this is not the main purpose of the tidyverse package.

- 5 Thank you for the suggestion about the `rio` package. We will indeed rephrase our description of `tidyverse`!

Done.

P11L10 Which netcdf package is the best? Or let's say, are there more information which package to use for a specific application.

- 10 We will discuss which package is most appropriate for different uses.

After reflection, they perform similar roles and we feel it is not our place to make that judgment, so have not added more information here.

P12L15 For all ggplot2 users it is really important to mention that the ggplot2 3.0 version offers support to visualize sf objects directly with a specific geom (geom_sf). So, if one is familiar with ggplot2 and want to visualize spatial data (raster, shapefiles etc.) this is a really easy way to do that? As the ggplot2 community is growing and growing this might be a really important information for some readers of the paper.

- 15

This is a great idea; thank you for the suggestion.

Done.

20

Table 2 Should be included in Sect 3.5

Yes; we will rectify this.

Actually it already was in section 3.5, but the LaTeX formatting made it look like it wasn't.

- 25 Sect. 3.6 Package “skimr” might be valuable for this section.

Good idea, thank you!

Done.

P15L10 I agree, one strength of ggplot2 and faceting is the ability to generate multiple views. This is especially important to avoid overplotting or – as the authors mentioned – to split a data set in different plots using meta information of this data set (i.e. season). If you have a nice example for faceting here would be a good place to show this principle for the readers.

- 30

We do have some good examples and will include one, thank you!

Done.

P19L17 Is it true that publishing a R package on CRAN is free of quality checks? However, a short comment on the differences during package publication (CRAN vs. GitHub) might be helpful here.

In our experience, CRAN only checks for code and documentation consistency, not code logic nor quality/coverage (CRAN does not run linter checks or test coverage). The point about the differences during package publication is important and
5 discussed briefly in section 5.2.

Done.

P19L21-26 I agree but the paragraph could be more specific on how we should develop R packages. Is there a good article or
blogpost describing the progress of publishing a R package? What are the requirements to generate good tutorials or readme
10 files? Might it be useful if journals like HESS recommend to publish paper code on github?

We will include (1) a description of the process, or provide a clear, straightforward reference on publishing packages; and (2) a
description of what a useful tutorial/readme should include for hydrology. Some references for package building include, for ex-
ample: (i) a popular minimal description of how to write a package <https://hilaryparker.com/2014/04/29/writing-an-r-package-from-scratch/>
(ii) helpful tutorials such as https://kbroman.org/pkg_primer/ or a more tidyverse oriented reference with a video: <https://www.rstudio.com/resources/videos/you-can-make-a-package-in-20-minutes/>. We will also suggest that journals should en-
15 courage authors to publish their code. For example, the journal *Geoscientific Model Development* already does this: https://www.geoscientific-model-development.net/about/code_and_data_policy.html.

Thank you for the helpful suggestions.

We have included the reference by Hadley Wickham, which is the clearest possible description of how to create an R package.
20 We have discussed vignettes in section 4.1. There is also already a discussion encouraging authors to publish their code.

References

Kelleher, C. and Wagener, T.: Ten guidelines for effective data visualization in scientific publications, *Environmental Modelling & Software*, 26, 822–827, 2011.

25

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^e, 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 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 a typical hydrological workflow based on reproducible principles and packages for retrieval of hydro-meteorological data, spatial analysis, hydrological modelling, statistics, and the design of static and dynamic visualizations and documents](#). 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 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 ERA5 (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), 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 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. GR4J, Perrin et al., 2003) and semi-distributed (e.g. TOPMODEL, Beven, 1997) models of the hydrological cycle - many of which were initially developed in older languages such as Fortran - have now been incorporated into R packages. Some of them have been translated into R and others remain in their initial languages, e.g. SWAT (Neitsch et al., 2005). The uptake of empirical, conceptual and physically-based computational hydrological models as well as the growth of large-sample 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 approach. 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 *hydroinformatics* (e.g. Abbott et al., 1991), which has a greater focus on information and communications technologies and/or black box approaches to address water-related issues, and *geocomputation*, (e.g. Openshaw and Abrahart, 2000; Lovelace et al., 2019), which uses similar computational techniques for spatial data analysis. [changed sentence structure]

The growth of computational hydrology has been enhanced by the development of the open-source programming language R, originally invented/developed for statistical computing for-statistics by Ross Ihaka and Robert Gentleman in the 1990s (Ihaka and Gentleman, 1996; 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 and geographic information system (GIS) applications. R can be used for generating reports, interactive presentations for teaching 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-ready ecosystems (e.g. web applications) and take advantage of cutting-edge technologies and tools for improving reproducibility, testing and continuous integration.

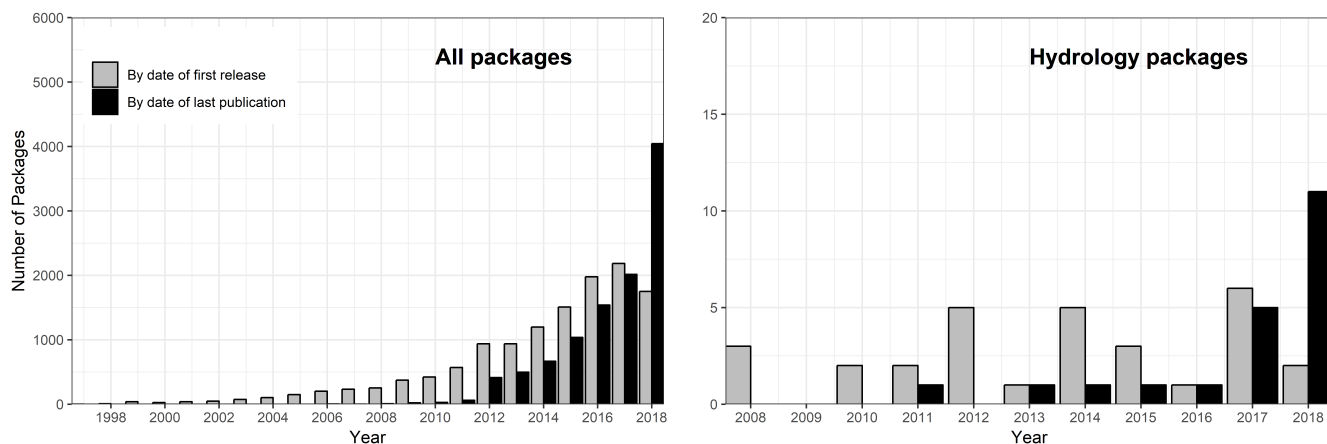


Figure 1. The number of R packages available on CRAN (1997-2018). Left: all packages; right: hydrology packages only (defined as those that include the string "hydro" within the package metadata; erroneous packages containing terms like hydrocarbon were removed). Bar colors indicate number of packages published based on (1) date of first release (bluegrey 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 (redblack 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 (bluegrey) but its most recent update is 2017 (redblack). The script to produce this figure is provided as part of the supplementary materials (Supplement 1). Data were downloaded from CRAN on 18 January 4 June 2019.

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 hydrology community.

2 The benefits and advantages of using R in hydrology

2.1 Democratizing open 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, object-oriented programming (the language is organized around objects with unique attributes), functional programming (the code can be written with functions to facilitate modularity and avoid changing-state data), and the availability of the 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 Figure 3). A high level of 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. For many R newcomers, the initial hurdle is finding the time required to understand-how-the-R-language-works gain basic understanding of R itself; but these efforts pay off rapidly as one learns how to find solutions on the Internet (see section 2.5) and to navigate help files provided within R.

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 re-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 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 and updated.

Recent developments in R have contributed to enhancing open science and numerical literacy in the hydrological sciences. R's ease of access and use has improved what might be termed "scientific computing literacy" within the hydrological community. Volunteer projects such as the Software Carpentry (Wilson, 2006, 2014) have been teaching basic computing skills to 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, hobbyists, 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, and code is increasingly being shared via repositories (see section 5.1). Code and results can be published viaas
5 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 written text within a unique document; see 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 6). They can be fixed (e.g. a non-
10 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 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 RPub (a free web-publishing platform, <https://rpubs.com>)
15 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.

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 (methods can be tested
20 by other researchers), 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., 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).

25 The open-source nature of R packages and the CRAN repository set-up are one key added value of R to reproducible geo-scientific research (Pebesma et al., 2012). 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
30 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~~ are submitted to on-line repositories such as GitHub that never make it to CRAN for a range of reasons, but this does not 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 devel-~~

opening and disseminating the use of best practice in writing R code. Packages such as `rhub` (Csárdi, 2017) ~~also provide code checks~~ can be used to verify code for packages on certain operating systems, while packages like `roxygen2` help users create documentation for their R packages (Wickham et al., 2018a). 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 encourage authors to publish the data and computer codes underlying the results presented in their papers. The journal *Nature* states 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~~ field. In addition to the sharing of open-source code, reproducibility experts also advocate the use of 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.

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 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 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 and popular language and environment for statistical computing. For example, a search for the word "statistics" on Stack Overflow, the question and answer website for programmers, on 18 May 2019 generated 12,339 results with the tag "R", 9,582 results with Python, 4,916 with Java, 2,623 with C#, and 48 with Fortran. R is "GNU S", a language which can be described as the modern implementation of the S language (Becker and Chambers, 1984; Chambers, 1998) ^[ref. added] and is specifically optimized

for statistical computing. In addition to the standard statistical 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 statistically-minded 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 [\[removed CRAN URL\]](#). A description of specific statistical packages relevant to hydrology is provided in section 3.6.

2.4 Connecting R to and from other languages

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; Guttig, 2013), R in 1993 (e.g. R Core Team, 2018; Ihaka, 1998), and Julia in 2012 (Bezanson et al., 2012).

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. Interconnections between R and other languages can be very useful for taking advantage of each language's strengths, or using pre-existing scripts that were originally written in other languages. For example, C and Fortran are both efficient in performing loop tasks. This is the reason why the `apply` functions loops are coded in C. The `airGR` hydrological models (Coron et al., 2017, 2018) also make use of Fortran in order to take advantage of its CPU efficiency when dealing with vector operations.

R can also be connected to different languages using a range of packages, e.g. C++ (package `Rcpp`, Eddebuettel 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 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. On Twitter, the "`#rstats`" community is particularly active, and users are exposed to communications about a range of new packages and recent developments. Stack Overflow (<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), 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.

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 through](#) 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 [for R users](#) 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://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~~[users](#) 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 provided to the community on GitHub (<https://github.com/hydrosoc>; [see section 5.6 for further information](#)). [It is also worth noting that a regular short course is also run at the EGU on R package development. In addition to the EGU, other regular conferences and meetings such as those run by the International Association of Hydrological Sciences \(IAHS\) and the Prediction in Ungauged Basins \(PUB Summer school\) also organise R courses during their meetings. The IAHS Statistical Hydrology \(STAHY\) commission regularly organizes workshops and short courses on various statistical topics which typically come with a computer-based component in R.](#)

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 stability of the R community.

3 R packages in ~~hydrology~~[a typical hydrological workflow](#)

note: [The sections have been altered so that they now follow the workflow in Figure 3.]



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

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 each of these steps in subsequent sections.

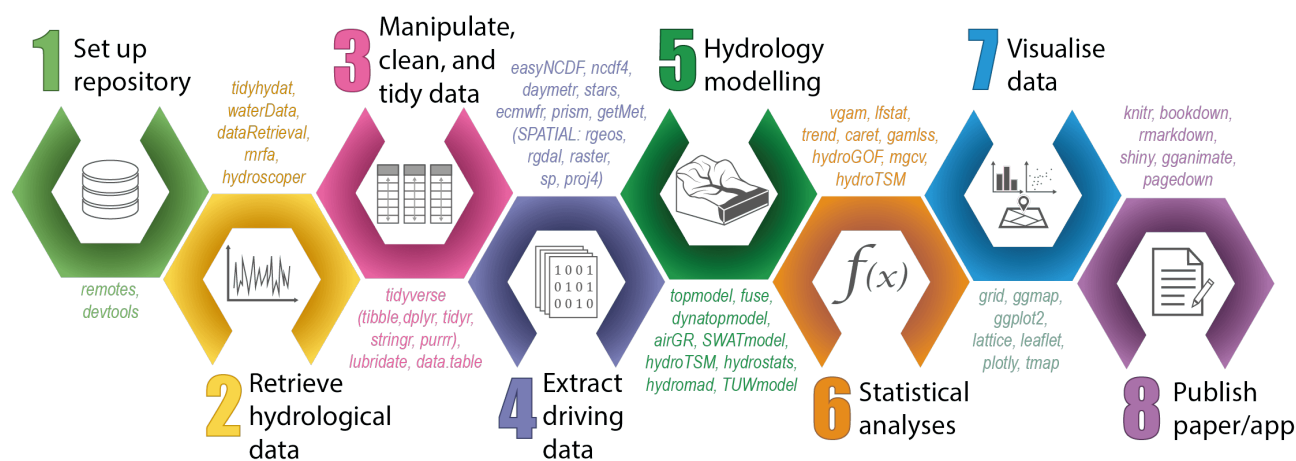


Figure 3. A typical hydrological workflow in R, containing eight steps (see Section). A selection of relevant R packages used within each script is indicated in coloured text; these packages are described in further detail in sections 3.2 to 3.8.

3.1 Setting up a repository and finding the right packages ~~the CRAN hydrology Task View and social media~~

- 5 Setting up a repository with version control (i.e. a reviewable and restorable history) at the start of a research project has many advantages. A repository is a structured set of files that will track edits that any team member makes to the project, similar to the "track changes" function in common word processors. In R, version control can be implemented quite simply by con-

necting RStudio with Git or Subversion, through hosting services like Bitbucket, GitHub, GitLab (which are like a DropBox for code). There are just a few initial steps to set up a repository. These include: (1) creating a local directory (folder) to host your (RStudio) project; (2) creating a Git or a Subversion repository online; and (3) linking the two. Beyond those three steps, all the user needs to do is regularly commit (i.e. save) their code, together with a very brief summary of the changes made.

- 5 These changes can then be uploaded, updating and synchronizing the online repository with labelled annotations (user name and timestamp). The repository can be public or private, with different levels of administrator access for users. Many tutorials on how to set up a repository in R can be found online. One example is the Software Carpentry course on 'R for Reproducible Scientific Analysis' (<https://swcarpentry.github.io/r-novice-gapminder/>) which also provides valuable suggestions on how to structure a project repository (e.g. with source data as 'read-only' and generated data as 'disposable', and folders such as 'scr' for scripts and 'doc' for documentation).

Once a project folder/repository has been set up, one might need to identify the most useful R packages and functions for the task at hand. 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:

- 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.
- 20 – 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>).

30 3.2 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 `rnrf` 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 import USGS daily hydrological data from the USGS web services, plot 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 scientific 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 5.4 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 ERA5 reanalysis are openly available (Hufkens, 2018).

Table 1. Examples of packages for hydrological and/or meteorological data retrieval. [See the Hydrology Task View for latest additions](https://cran.r-project.org/web/views/Hydrology.html) <https://cran.r-project.org/web/views/Hydrology.html>

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

3.3 Packages for reading, manipulating, and cleaning the data

note: [We added this section so the text follows the structure shown in Figure 3.]

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, that form part of the tidyverse package suite (Wickham, 2017), which facilitate reading a range of formats including `xls(x)`, `tsv`, `fwf` (e.g. package `readr`; Wickham et al., 2018b). The `rio` package facilitates data import and export by making assumptions about data structure from the file extension. (Chan et al., 2018).

For reading or writing `netcdfnetCDF` 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).

Observed hydro-climatological time series data typically need to be "cleaned" because they suffer from various data gaps and errors. For an overview of the different issues with hydrological data see Wilby et al. (2017). 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 in R. However, for overviews and tutorials on manipulating and cleaning hydrological data we point the reader to published resources from the "Using R in hydrology" workshop (see section 5.6).

3.4 Packages for extracting driving data, 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 and extracting information. 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; or package `sf` (Pebesma, 2018a), which provides support for simple features to encode spatial vector data. [Version 3.0 of ggplot2 offers support to visualize sf objects directly with a specific geom \(geom_sf\), which allows users to visualize spatial data \(raster, shapefiles\) easily.](#) 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` (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; [RgoogleMaps is used to interact with Google Maps](#); while the JavaScript `leaflet` (Cheng et al., 2018) library can be 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 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).

3.5 Packages for hydrological modelling

[The next step in a typical hydrological workflow is to conduct hydrological modelling by using the data inputs prepared in previous steps.](#) 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 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` (Metcalf et al., 2015, 2018); SWAT in `SWATmodel` (Fuka et al., 2014); and GR4J in `airGR` (Coron et al., 2017, 2018) and `hydromad` (Andrews et al., 2011; 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

Table 2. Examples of packages for hydrological modelling.[See the Hydrology Task View for latest additions: https://cran.r-project.org/web/views/Hydrology.html](https://cran.r-project.org/web/views/Hydrology.html)

Package	Description
<code>airGR</code>	Suite of GR Hydrological Models for Precipitation-Runoff Modelling (Coron et al., 2017, 2018)
<code>brook90</code>	Implementation of the BROOK90 hydrologic model (Federer, 1995; Schmidt-Walter, 2018; Kronenberg and Oehlschl�gel, 2019)
<code>dynatopmodel</code>	Implementation of the Dynamic TOPMODEL Hydrological Model (Metcalf� et al., 2015, 2018)
<code>Ecohydmod</code>	Ecohydrological Modelling (Souza, 2017)
<code>fuse</code>	Ensemble Hydrological Modelling (Vitolo et al., 2016b)
<code>hydromad</code>	Hydrological Model Assessment and Development (Andrews et al., 2011; Andrews and Guillaume, 2018)
<code>RHMS</code>	Hydrologic Modelling System for R Users (Arabzadeh and Araghinejad, 2018)
<code>topmodel</code>	Implementation of the Hydrological Model TOPMODEL in R (Buytaert, 2018)
<code>TUWmodel</code>	Lumped Hydrological Model for Education Purposes (Viglione and Parajka, 2014)

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 the interactions between the processes) models. [Some packages, such as `TUWmodel` and `topmodel`, allow a spatial distribution of the hydrological processes. `TUWmodel`, for instance, allows the user to implement](#)

5 [a semi-distributed application with differentiation into elevation zones, while `topmodel` allows the user to make topographic index classes from a topographic index map. The FUSE model \(Clark et al., 2008\) originally implemented in Fortran, is still actively developed and also has a dedicated R package called `fuse` \(Vitolo et al., 2016b\) that is a re-implementation of the FUSE 2011 model.](#) 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 [and the `hydromad` package is intended to provided a](#)

10 [flexible framework to assemble different soil moisture accounting and routing schemes.](#)

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 functions (e.g. `airGR` and `RHMS`). 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

15 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

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. [The `skimr`](#)

5 [package](#) provides compact and flexible summaries of data; can be used with pipes and displays nicely in the console. 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-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
10 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. [While `mgcv` can be useful for more complex smoothing approaches \(e.g. tensor product smooths or varying coefficient models\), the focus of `gamlss` is to develop flexible models for a large class of distributions which are not part of the exponential family of distributions, allowing for parameters other than the location to be modelled as a smooth function of the covariates of interest.](#)
15 [A data analyst might therefore choose which package to use depending on the most likely distribution of the data which are being analyzed and the complexity of the analysis carried out.](#) The `caret` package (Kuhn et al., 2018) [is more commonly used](#) for predictive modelling and machine learning, [as it](#) allows the user to develop an impressive range of predictive 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
20 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). [Lmoments can be implemented using the `lmom` Hosking \(2019b\) package, while the `lmomRFA` package Hosking \(2019a\) enables regional frequency analysis.](#) The `lfstat` (Koffler et al., 2016) package provides functions to compute
25 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 simulated and observed hydrological time series. In Ta-
30 ble 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>).

Table 3. Examples of packages for hydrological statistics.

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

3.7 Packages for static and dynamic hydrological data visualization

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 GRZGR-Z model (Becker and Chambers, 1977) [ref. added], 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), 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` 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 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. Figure 4 provides a hydrological example of faceting where a statistical model is visualised for different seasons (rows) and different peak-over-threshold quantities (columns).

Dynamic charts - where the user can, for instance, hover over one or multiple points to read the associated (meta)data (e.g. a hydrometric station number or the value of a point) - have also grown in popularity in hydrological analyses in recent

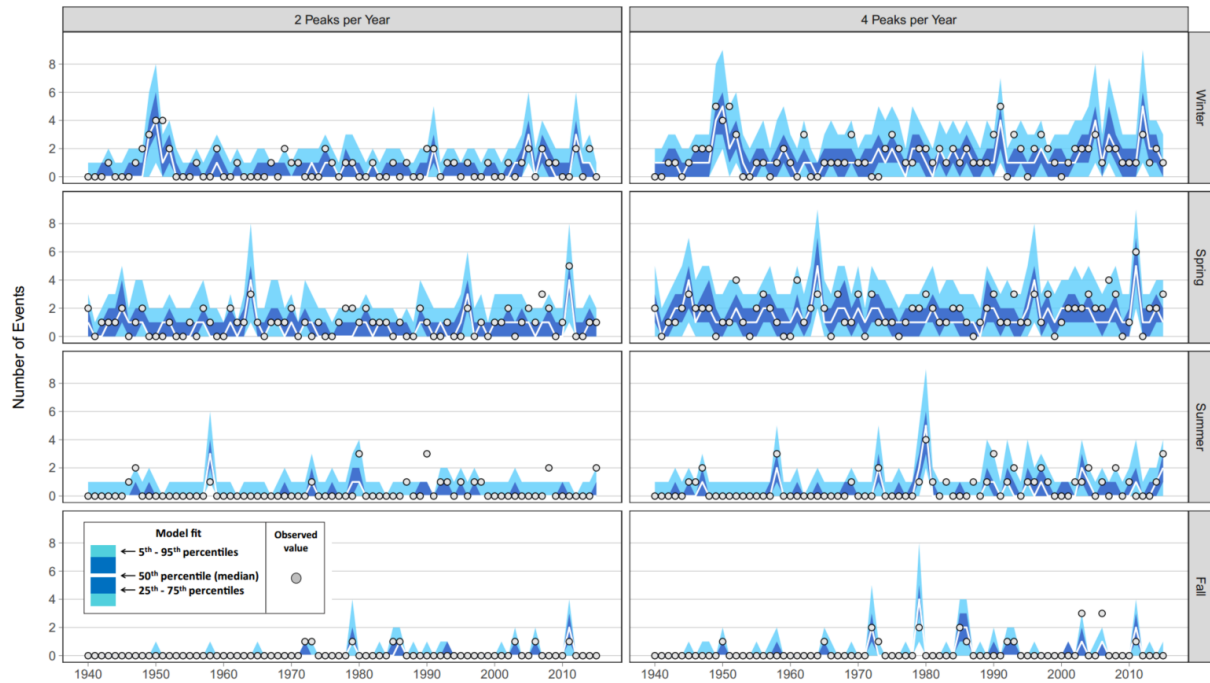


Figure 4. An example of a manuscript figure produced in R with the package `ggplot2` using the `facet_grid` function, which allows the user to split a dataset by categories (rows and columns). Here, a probabilistic time series model of flood frequency at one gauging station in Ohio, USA, is shown for four different seasons (rows), and two peak-over-threshold quantities (columns: 2-4 peaks per year). The probabilistic model is plotted in the background as color ribbons (using `geom_ribbon`) and the observed values are plotted in the foreground as grey circles (`geom_point`). Source: Neri et al. (2019).

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

5 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` package (Allaire, 2014) allows the user to easily add sliders and other control tools to otherwise static plots. Other JavaScript-based 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` (Thieurmél et al., 2018) creates

10 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`,

highchart, 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.

Choosing the appropriate colour gradients for hydrological graphs and maps is key. It is widely accepted now that certain color schemes, and notably the infamous rainbow color scale, are poor choices for data visualization (Light and Bartlein, 2004;

- 5 Stauffer et al., 2015). The rainbow scheme has been shown to distort perceptions of data, and alter meaning by creating false boundaries between values, and additionally is not colourblind-safe, and other alternatives like perceptually uniform color maps have been suggested (Crameri, 2018) . The R language is strong in the area of colour gradients and there are many colourblind-friendly palettes available for hydrology that follow effective data visualization guidelines (Kelleher and Wagener, 2011). There are both manually-defined and predefined palettes using packages such as RColorBrewer (Neuwirth, 2014), colorspace (Zeileis et al., 2019), colorRamps (Keitt, 2012) or viridis (Garnier, 2018). For example, the colorspace package can be used for hue chroma luminance (HCL) based color palettes (which are based on how humans perceive color, in contrast to the RGB color space, which is based on technical demands of TV/computer screens) in meteorological visualizations (Stauffer et al., 2015).

3.8 Packages for creating presentations and documents

- 15 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 6, see Section 5.5). 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 Earth Observation. GIFs can be created using for example `gganimate` (Pedersen, 2018) to animate `ggplot2` and create videos
- 20 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), `blogdown` (Xie et al., 2017) or `pkgdown` (Wickham and Hesselberth, 2018).

- 25 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, 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.

4 Challenges and solutions when using R in hydrology

4.1 Hydrological libraries, documentation, and vignettes

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 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; in fact these are key to reducing the misuse of software and ensuring the package users understand the methods. Use cases can be written as blog-posts or tutorial-style papers with explanation of how to correctly interpret and implement a specific method, and helping the community move forward.

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 Rivières and Wiegand, 2004; Wahlbrink, 2016), Emacs Speaks Statistics (ESS, an add-on package for GNU 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 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 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 [version 1.2 includes many new features](#) to help interact with other programming platforms, such as the `reticulate` (Allaire et al., 2018b) and `r2d3` (Luraschi and Allaire, 2018) packages, which make 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

- 10 In the early years of R, the software was unable to handle large data files exceeding millions of rows or complex data formats. 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
- 15 joins. In hydrology, some of the spatial data handling packages are particularly relevant, such as the `sf` package (Pebesma, 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).
- 20 As the volume of available data increases and hydrologists use a greater number of models and ensembles, parallel computing - 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 where 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
- 25 more than one core at a time) are considerable. Even without access to a high-performance computer or cluster, it is possible to 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
- 30 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 `h2o` package (LeDell

et al., 2018) that provides an interface to H2O, a scalable platform offering parallelized implementations of supervised and 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.

5.1 Sharing R code with the community

[Note: We have created a new section here, inserted some of the text from 5.2, restructured the paragraph and added information about sharing platforms following Reviewer 3's suggestion]

- 10 Open research practices bring significant benefits to researchers ~~relative to more traditional closed practices~~ (McKiernan et al., 2016), such as increases in citations, media attention, potential collaborators, job opportunities and funding opportunities. Most im-
importantly, sharing code increases the likelihood that an approach will be used by other scientists in their research, and saves
new users the trouble of "reinventing the wheel" and writing codes that have already been developed by others. There are
many different platforms for sharing and publishing R code such as GitHub, Figshare, and RPub/Plotly (also for dashboards
15 and interactive plots). To obtain a DOI, users may wish to use repositories such as ZENODO, a general purpose open-access
repository which allows researchers in the sciences and humanities to deposit data sets, research software, and reports. The
data publisher PANGAEA, additionally, is specifically tailored to archiving, publishing and re-usage of data in Earth and En-
vironmental sciences. For further guidance, some of these platforms are discussed in the code and data policy section of the
open-access journal *Geoscientific Model Development*.

20 5.2 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, as~~ packages favor the uptake and development of ~~hydrological~~ methods. Additionally, the open-source nature of R packages means that different users can contribute feedback to R package developers and help enhance existing code. R users can raise issues for certain packages directly on online platforms hosting the repositories (such
25 as GitHub or Gitlab) to foster an online-documented discussion with the package developers (generating interaction in the
community). Package authors can also add a bugs report link or an email address in their package Description file (a common
best-practice for developers) to specify how they prefer to receive bug reports. This feedback between users and developers is
one key route to scientific progress. Users can identify issues or suggest improvements by commenting on online collaboration
platforms (e.g. GitHub or GitLab) or by emailing the developer/maintainer. Most packages are hosted on ~~the GitHub platform~~ repos-
30 itory hosting platforms (Github, Gitlab) 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 and documentation that follows a widely accepted style-guide, such as the lintr package (Hester, 2018), goodpractice (Csardi and Frick, 2018) and hunspell (Ooms, 2018) packages, and (ii) rigorously test the functions against known pre-defined outcomes, using for example the testthat package (Wickham, 2011). In comparison with GitHub, CRAN is a specific organised repository which provides executables for several platforms, whereas
5 finding an hydrological R package on Github can be difficult. The comparison between the two is somewhat like visiting a curated art auction versus an antique store. Additionally, CRAN has more thorough checks and gives visibility/searchability to a package, also signalling that the author intends to maintain the package over the long term.

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
10 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-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, not a minor task, and~~ 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 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
20 reproducibility) and sometimes also include an additional journal paper reference. If both are available, then ideally both must be 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~~ support reproducibility via 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
25 (e.g. due to a lack of scientific scrutiny) can be alleviated through software peer review initiatives such as those provided by rOpenSci. It is worth noting that when writing new packages, the open-source approach has both pitfalls (risk of errors in new packages) and strengths (community review, and range of tests that can be implemented via the above-mentioned packages).

5.3 Harmonizing hydrological workflows

One of the main challenges today for hydrological workflow harmonization is to standardise calls to data which are structured
30 in different ways by data providers around the world. While the number of packages for hydrological data retrieval has grown considerably (see Table 1), the packages are structured differently because they were set up independently and because the underlying hydrological and hydrometric datasets differ from country to country. As more nations develop similar hydrological data acquisition packages, we believe it would be worth implementing a common syntax and data output form. For example, it would be ideal to use consistent APIs and output objects across packages. Suggesting the "ideal" format is beyond the scope

of this manuscript, but a potential task force might be set up to help draft a way forward for the community. There is currently no effort to combine data retrieval packages from different regions, but it might be worth implementing a meta-package with functions that convert hydrometric data from other packages to a standard format, or one for hydrological models to be run within the same framework. Additionally, the growing tendency towards data standardization at the global level may facilitate the development of consistent data retrieval packages. Examples of global data standards include WaterML, the Open Geospatial Consortium standards for hydrological time series, or TimeSeriesML, a more generic candidate standard currently under discussion. As more water agencies and data providers adopt such standards, it will become much easier to develop consistent data retrieval packages, and in theory we may no longer require different packages for data retrieval because the only component that should change is the server endpoint.

5.4 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, 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 `rnrf` (Vitolo et al., 2016a), `tidyhyd` (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. Programmatically 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 implementation, Ooms, 2014) and RStudio's `plumber` 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 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 deeper understanding of the underlying science. 4) Empirical data is rarely "clean" and scientific computing hence has to be

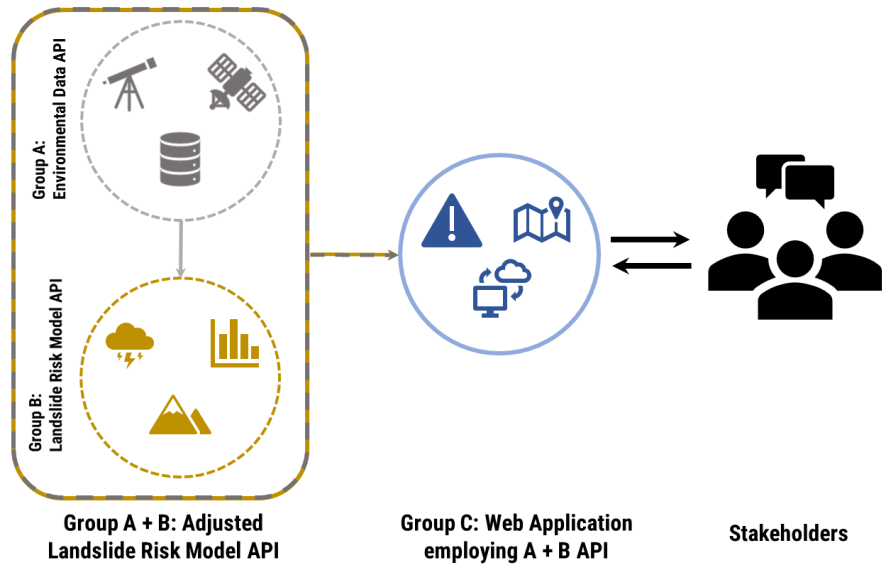


Figure 5. Use-case scenario for implementation of APIs developed within hydrological sciences. Group A collates the most recent earth and ground-based observations of land cover, topography and climate data for an in-house project, yet provides an API to interact with an analyses-ready, spatially-explicit 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.

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 5.

5.5 Teaching hydrology in R

- 5 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.

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



Figure 6. 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.

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-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 6). 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.

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, sandboxes can be used by educators/lecturers by asking students to connect directly to the web.

5.6 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 [on the YHS GitHub pages](https://github.com/hydrosoc/) (<https://github.com/hydrosoc/>) and [archived on the Zenodo repository](#) (see Boessenkool et al., 2017; Slater et al., 2018; Hurley et al., 2019). [added 2019 repository] These presentations include a range of topics such as: [obtaining, cleaning and visualizing hydrological data with R](#); [parallel and high performance computing for hydrologists](#); [automating tasks](#); [using R Shiny with hydrological data](#); [modelling the hydrological cycle in snow-dominated catchments](#); [community-led initiatives](#); 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.

[Many other R conferences will be of relevance to the readers, including the *useR!* conference, *RStudio*, *satuRdays*, or *eRum*. These conferences are not specific to hydrology and earth sciences, but usually have some discipline-specific sessions as well as sessions and short courses of general interest for learning R, including best practice, spatial analysis, statistical methods. Many of these conferences also stream the talks and/or make resources available after the event.](#)

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 7).

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 high-quality](#) 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.

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 influence of R on the discipline. Both the flexible nature of the language and the diverse range of computational, visualization,

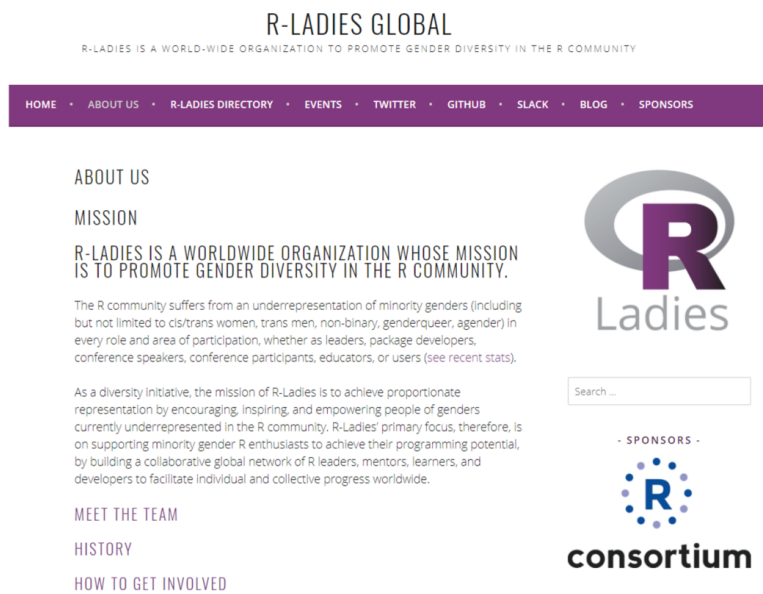


Figure 7. The R community. Left: the R meetup groups <https://www.meetup.com/topics/r-project-for-statistical-computing>; Right: the R-Ladies global organization <https://rladies.org/about-us>.

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 of hydrological theory, models and tools. These new computational tools and approaches are essential to achieving long-term goals such as the 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 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

Acknowledgements. We thank two anonymous reviewers and Michael Stoezle for comments that improved the manuscript. We also ~~The~~ ~~authors~~ thank all those who have contributed to the R-Hydro community and the R community more broadly, whether by developing and sharing their code, or helping and teaching others to use these tools. In particular, we thank the guest speakers who have participated in the "Using R in Hydrology" EGU short course. Finally, we also thank Paul Astagneau, MSc at Irstea, for the discussions on modelling packages

5 that helped to improve the manuscript.

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=](https://CRAN.R-project.org/package=markdown)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.
- Andrews, F., Croke, B., and Jakeman, A.: An open software environment for hydrological model assessment and development, *Environmental Modelling & Software*, 26, 1171 – 1185, <https://doi.org/https://doi.org/10.1016/j.envsoft.2011.04.006>, 2011.
- 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.
- 20 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.
- 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.
- 25 Becker, R. A. and Chambers, J. M.: GR-Z: A System of Graphical Subroutines for Data Analysis, in: *Computer Science and Statistics*, pp. 409–415, U.S. Dept. of Commerce, National Bureau of Standards : for sale by the Supt. of Docs., U.S. Govt. Print. Off., Gaithersburg, Maryland, 1977.
- Becker, R. A. and Chambers, J. M.: S: an interactive environment for data analysis and graphics, *The Wadsworth statistics/probability series*, Wadsworth Advanced Book Program, Belmont, Calif, 1984.
- 30 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.
- 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.
- 35 Bergström, S. and Forsman, A.: Development of a conceptual deterministic rainfall-runoff model, *Nordic Hydrol.*, 4, 147–170, <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](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.
- 5 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.
- 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.
- 10 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.
- 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.
- 15 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.
- 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.
- 20 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.
- 25 Bond, N.: hydrostats: Hydrologic Indices for Daily Time Series Data, <https://CRAN.R-project.org/package=hydrostats>, R package version 0.2.6, 2018.
- 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.
- 30 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.
- 35 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.

- 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.
- 5 Chambers, J. M.: *Programming with data: A guide to the S language*, Springer Science & Business Media, 1998.
- 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.
- Chan, C.-h., Chan, G. C., Leeper, T. J., and Becker, J.: rio: A Swiss-army knife for data file I/O, r package version 0.5.16, 2018.
- 10 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.org/package=leaflet>, R package version 2.0.2, 2018.
- 15 Clark, M. P., Slater, A. G., Rupp, D. E., Woods, R. A., Vrugt, J. A., Gupta, H. V., Wagener, T., and Hay, L. E.: Framework for Understanding Structural Errors (FUSE): A modular framework to diagnose differences between hydrological models, *Water Resources Research*, 44, 2008.
- Coles, S.: *An Introduction to Statistical Modeling of Extreme Values*, Springer Series in Statistics, Springer-Verlag, London, <http://www.springer.com/gp/book/9781852334598>, 2001.
- 20 Conway, J., Eddebuettel, 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.
- 25 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 Wesley, 2001.
- Cramer, F.: Geodynamic diagnostics, scientific visualisation and StagLab 3.0, *Geoscientific Model Development*, 11, 2541–2562, 2018.
- Csardi, G. and Frick, H.: goodpractice: Advice on R Package Building, <https://CRAN.R-project.org/package=goodpractice>, r package version 1.0.2, 2018.
- 30 Csárdi, G.: rhub: Connect to 'R-hub', <https://CRAN.R-project.org/package=rhub>, R package version 1.0.2, 2017.
- 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.
- 35 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.
- 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 Rivières, 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
5 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.
- Eddelbuettel, D. and Balamuta, J. J.: Extending extitR with extitC++: A Brief Introduction to extitRcpp, *PeerJ Preprints*, 5, e3188v1,
10 <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.
- 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
15 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.
- 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>,
20 2011.
- Faria, J., Grosjean, P., and Jelihovschi, E.: Tinn-R-GUI/Editor for R language and environment statistical computing, Disponivel em:<
<http://sourceforge.net/projects/tinn-r>, 2008.
- Federer, C.: BROOK90: a simulation model for evaporation, soil water and streamflow, 1995.
- 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.
25
- Garnier, S.: viridis: Default Color Maps from ‘matplotlib’, <https://CRAN.R-project.org/package=viridis>, r package version 0.5.1, 2018.
- 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.
- 30 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.
- Giraud, T. and Lambert, N.: cartography: Create and Integrate Maps in your R Workflow, *JOSS*, 1, <https://doi.org/10.21105/joss.00054>, 2016.
- 35 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.

- 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.
- 5 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.
- Gutttag, J. V.: Introduction to computation and programming using Python, Mit Press, 2013.
- 10 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.
- 15 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.
- 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.
- 20 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 Boissésou, 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.
- 25 Hester, J.: lintr: A 'Linter' for R Code, <https://CRAN.R-project.org/package=lintr>, R package version 1.0.3, 2018.
- 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.
- 30 Hosking, J. R. M.: Regional Frequency Analysis using L-Moments, <https://CRAN.R-project.org/package=lmomRFA>, R package, version 3.2, 2019a.
- Hosking, J. R. M.: L-Moments, R package, version 2.8, 2019b.
- Hufkens, K.: ecmwfr: Interface to the public 'ECMWF' Web Services, <https://github.com/khufkens/ecmwfr>, R package version 0.0.2, 2018.
- 35 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, 2018.

- Hurley, A.: Processing, modelling and visualizing hydrological data in R (Version 1.1.1) Zenodo, <http://doi.org/10.5281/zenodo.2553998>, 2018.
- Hurley, A., Slater, L., Thirel, G., Barker, L., Vitolo, C., Prosdocimi, I., and Harrigan, S.: Using R in Hydrology at EGU2019, <https://doi.org/10.5281/zenodo.3236979>, <https://doi.org/10.5281/zenodo.3236979>, 2019.
- 5 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.
- 10 Ihaka, R. and Gentleman, R.: R: a language for data analysis and graphics, *Journal of computational and graphical statistics*, 5, 299–314, 1996.
- 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.
- 15 Keitt, T.: colorRamps: Builds color tables, <https://CRAN.R-project.org/package=colorRamps>, r package version 2.3, 2012.
- Kelleher, C. and Wagener, T.: Ten guidelines for effective data visualization in scientific publications, *Environmental Modelling & Software*, 26, 822–827, 2011.
- Kilibarda, M. and Bajat, B.: plotGoogleMaps: the R-based web-mapping tool for thematic spatial data, *GEOMATICA*, 66, 37–49, 2012.
- 20 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.
- 25 Kronenberg, R. and Oehlschlägel, L. M.: Implementation of the BROOK90 model in R, https://github.com/rkronen/Brook90_R, 2019.
- 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://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, 30 2017.
- Lawrence, M. and Wickham, H.: plumb: Mutable and dynamic data models, RStudio, R package, version 0.6.9, <http://CRAN.R-project.org/package=plumb>, 2018.
- 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.
- 35 Li, C.: JuliaCall: Seamless Integration Between R and 'Julia', <https://CRAN.R-project.org/package=JuliaCall>, R package version 0.16.0, 2018.
- Light, A. and Bartlein, P. J.: The end of the rainbow? Color schemes for improved data graphics, *Eos, Transactions American Geophysical Union*, 85, 385–391, 2004.

- 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.pdf>, 2018.
- McKiernan, E., Bourne, P., Brown, C., Buck, S., Kenall, A., Lin, J., McDougall, D., Nosek, B., Ram, K., Soderberg, C., Spies, J., Thaney, K., Updegrave, A., Woo, K., and Yarkoni, T.: *How open science helps researchers succeed*, *eLife*, 5, <https://doi.org/10.7554/eLife.16800>, 2016.
- 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.
- Metcalfe, P., Beven, K., and Freer, J.: *Dynamic TOPMODEL: A new implementation in R and its sensitivity to time and space steps*, *Environmental Modelling & Software*, 72, 155 – 172, <https://doi.org/10.1016/j.envsoft.2015.06.010>, 2015.
- 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.
- 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>, R package version 2.1.1, 2018.
- Neil, D.: *Modern Vim: Craft Your Development Environment with Vim 8 and Neovim*, Pragmatic Bookshelf, 2018.
- Neitsch, S. L., Arnold, J. G., Kiniry, J. R., and Williams, J. R.: *"Soil Water Assessment Tool Theoretical Documentation (version 2005)"*, <https://swat.tamu.edu/media/1290/swat2000theory.pdf>, Grassland Soil and Water Research Laboratory, Temple, Texas, 2005.
- Neri, A., Villarini, G., Slater, L. J., and Napolitano, F.: *On the statistical attribution of the frequency of flood events across the US Midwest*, *Advances in Water Resources*, 2019.
- Neuwirth, E.: *RColorBrewer: ColorBrewer Palettes*, <https://CRAN.R-project.org/package=RColorBrewer>, r package version 1.1-2, 2014.
- Ooms, J.: *The OpenCPU system: Towards a universal interface for scientific computing through separation of concerns*, *arXiv preprint arXiv:1406.4806*, 2014.
- Ooms, J.: *V8: Embedded JavaScript Engine for R*, <https://CRAN.R-project.org/package=V8>, 2017.
- Ooms, J.: *hunspell: High-Performance Stemmer, Tokenizer, and Spell Checker*, <https://CRAN.R-project.org/package=hunspell>, r package version 3.0, 2018.
- 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.

- 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.
- Pebesma, E., Nüst, D., and Bivand, R.: The R software environment in reproducible geoscientific research, *Eos, Transactions American Geophysical Union*, 93, 163–163, 2012.
- 5 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.
- Perrin, C., Michel, C., and Andréassian, V.: Improvement of a parsimonious model for streamflow simulation, *Journal of Hydrology*, 279, 275 – 289, [https://doi.org/https://doi.org/10.1016/S0022-1694\(03\)00225-7](https://doi.org/https://doi.org/10.1016/S0022-1694(03)00225-7), 2003.
- Pierce, D.: ncdf4: Interface to Unidata netCDF (Version 4 or Earlier) Format Data Files, <https://CRAN.R-project.org/package=ncdf4>, R
10 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://www.R-project.org>, 2018.
15
- 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-
20 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://CRAN.R-project.org/package=waterData>, R package version 1.0.8, 2017.
25
- 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://lmdvr.r-forge.r-project.org>, 2008.
- Schmidt-Walter, P.: brook90r, <https://doi.org/10.5281/zenodo.1433677>, <https://doi.org/10.5281/zenodo.1433677>, 2018.
- 30 Sievert, C.: plotly for R, <https://plotly-book.cpsievert.me>, 2018.
- Slater, L., Harrigan, S., Gauster, T., Hurley, A., Thirel, G., and Vitolo, C.: Using R in Hydrology, EGU2018 Short Course. Zenodo, <https://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.
- 35 Sommerlot, A., Fuka, D., and Easton, Z.: getMet: Get Meteorological Data for Hydrologic Models, <https://CRAN.R-project.org/package=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.

- Stauffer, R., Mayr, G. J., Dabernig, M., and Zeileis, A.: Somewhere over the rainbow: How to make effective use of colors in meteorological visualizations, *Bulletin of the American Meteorological Society*, 96, 203–216, 2015.
- 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.
- 5 Tennekes, M.: tmap: Thematic Maps in R, *Journal of Statistical Software*, 84, 1–39, <https://doi.org/10.18637/jss.v084.i06>, 2018.
- Thieurmél, 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 package version 1.17.1.1, 2018.
- 10 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 Thieurmél, 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.
- 15 Vantas, K.: hydroscooper: 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.
- Viglione, A. and Parajka, J.: TUWmodel: Lumped Hydrological Model for Education Purposes. R package version 0.1-4, 2014.
- 20 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.
- Vitolo, C., Fry, M., and Buytaert, W.: rnrf: 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.
- 25 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.
- 30 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.
- Wickham, H.: ggplot2: Elegant Graphics for Data Analysis, Use R!, Springer, New York, <http://ggplot2.org>, 2016b.
- 35 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.

- Wickham, H., Danenberg, P., and Eugster, M.: roxygen2: In-Line Documentation for R, <https://CRAN.R-project.org/package=roxygen2>, r package version 6.1.1, 2018a.
- 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, 2018b.
- 5 Wilby, R. L., Clifford, N. J., De Luca, P., Harrigan, S., Hillier, J. K., Hodgkins, R., Johnson, M. F., Matthews, T. K., Murphy, C., Noone, S. J., et al.: The ‘dirty dozen’ of freshwater science: detecting then reconciling hydrological data biases and errors, *Wiley Interdisciplinary Reviews: Water*, 4, e1209, 2017.
- Wilkinson, L.: *The Grammar of Graphics, Statistics and Computing*, Springer, New York, first edn., 1999.
- Wilson, G.: Software carpentry: getting scientists to write better code by making them more productive, *Computing in Science & Engineering*, 8, 66–69, 2006.
- 10 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.
- 15 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.
- 20 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 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.
- 25 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.
- 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.
- 30 Zeileis, A., Fisher, J. C., Hornik, K., Ihaka, R., McWhite, C. D., Murrell, P., Stauffer, R., and Wilke, C. O.: colorspace: A Toolbox for Manipulating and Assessing Colors and Palettes, arXiv 1903.06490, arXiv.org E-Print Archive, <http://arxiv.org/abs/1903.06490>, 2019.