POLITICAL RESEARCH EXCHANGE
2025, VOL. 7, 2549079
https://doi.org/10.1080/2474736X.2025.2549079

RESEARCH NOTES

OPEN ACCESS Check for updates

Automating awareness, digitizing dissemination: the radical right research robot as a template for (political) science communication in a fragmented social media landscape

Kai Arzheimer

Department of Political Science, University of Mainz, Mainz, Germany

ABSTRACT

Over the last 15 years, social media have become an integral part of the science infrastructure. The emergence of 'Science Twitter', the collective of scholars active on the platform now known as 'X', is a particularly prominent example of this trend. Within such communities, automated services ('bots') can play an important role by raising public and internal awareness of publications in the field, fostering connections amongst researchers, and stimulating serendipitous insights. In this note, I discuss the architecture of one such service: the Radical Right Research Robot (RRRR), written in the R programming language, which posts references and links to contributions from the vast literature on radical right/far-right parties and their voters. I also describe the various R packages and other open source tools on which RRRR relies. This description could serve as a template for similar bots in other fields. Finally, I discuss the challenges resulting from Elon Musk's takeover of Twitter and the subsequent decline of Science Twitter. RRRR has already branched out to two alternative text-based networks (Mastodon and Bluesky) and may eventually disengage from Twitter. While the technical difficulties involved are relatively easy to solve, fundamental issues related to platform ownership and governance persist.

ARTICLE HISTORY
Received 6 December 2024
Accepted 12 August 2025

KEYWORDS
Social media; science twitter; Twitterbot; radical right research

Introduction

Over the last 15 years, social media have become an important part of the science infrastructure. Twitter (now X) in particular emerged as the most important social medium for both internal and external science communication, with many scientists actively using the platform (Insall 2023). Since the 2010s, this constellation has been known as 'Science Twitter': a virtual space to share the latest papers, alongside chit-chat, rumours, and pictures of cats.

Part of Twitter's allure are 'bots': small programs that provide an automated information service to their 'followers', ranging from the obscure to the essential. One of

CONTACT Kai Arzheimer arzheimer@politik.uni-mainz.de

these is the Radical Right Research Robot (RRRR). RRRR regularly posts references and links to contributions from the vast literature on radical right/far-right parties and their voters. These references are pulled from a large bibliography for the field that is freely available on the internet (Arzheimer 2025). RRRR sent its first experimental post on February 4, 2018. Since February 27, it has posted regularly, apart from some outages caused by changes at Twitter.

To add value for its followers, RRRR provides context by highlighting the specific topic that a given text addresses, or flagging up the parties and countries that feature in the respective contribution. The ultimate aims of RRRR are (1) to raise public and internal awareness of the large body of knowledge that the field has accumulated, (2) to foster connections amongst researchers, and (3) to stimulate serendipitous insights, all by posting six references per day, every day.

RRRR is an academic side project that is made possible by free and open software. While its design principles are far from optimal, it could serve as a template for similar bots in other fields. In the remainder of this note, I will therefore present a general outline of RRRR's architecture and describe the packages on which it is built.

However, Science Twitter, and with it RRRR, face significant challenges. First, from 2022 Twitter has made access to its API more expensive and more difficult, which created considerable problems for non-commercial projects such as RRRR. More importantly, with Twitter's takeover by Elon Musk and the controversial changes to platform governance and moderation that followed, Science Twitter began to fragment and decline. While scientists remain active on Twitter or at least use the medium passively, many institutions, individual researchers and even whole epistemic communities have moved to other platforms, amongst which Bluesky and Mastodon are the most prominent.

To address these challenges, which I discuss in the last section, RRRR is branching out to these alternative networks. At the time of writing (July 2025), RRRR still has 1642 followers on Twitter, but also 355 followers on Mastodon and 996 followers on Bluesky. These figures partly reflect the consequences of the latest 'X-odus': while Mastondon is mostly stable, the number of RRRR's followers on Twitter is slowly decreasing whereas its audience on Bluesky has almost doubled in the final quarter of 2024 and almost doubled again over the following two quarters.

Based on the information in the profiles, its remaining remaining audience on Twitter chiefly comprises extremism researchers based at universities and think tanks, (PhD) students, and journalists. The picture on Bluesky is quite similar, apart from a higher share of left-leaning activists who presumably moved away from Twitter. The audience on Mastondon resembles the one on Bluesky, although here the share of computer scientist appears to be somewhat higher. The number of likes, reshares, and comments that the bot's posts receive is fairly limited on Mastodon (about one every other day) and Twitter (about one per day). Interaction is more frequent on Bluesky, with between two and four reactions on an average day, often by scientists commenting on the respective reference. The bot itself makes no attempts to programmatically respond to these reactions.

The bibliography on the far/extreme/radical right

The bibliography behind the bot has been available as a public online resource since 2010. It originated from the extensive (391 titles) reference section of a monograph

(Arzheimer 2008). Since then, it has grown through scholars' generous donations of their personal reference collections. An annual campaign on social media also invites nominations for new additions, highlighting the need to include more work by women and members of marginalized groups. Additionally, scholars frequently share their relevant work with me, and I incorporate pertinent references that I encounter during my day-to-day research.

To ensure comprehensiveness, I also monitor the RSS feeds of the most important political and social science journals using keywords such as 'radical right' and have set up automatic content alerts on Semantic Scholar. Furthermore, I regularly review the 'publications alert' section in the ECPR's Standing Group on Extremism & Democracy newsletter. Finally, as almost all journal articles appear 'online first' nowadays, I also aim to update entries once they have been assigned to an issue. Although this process may seem labor intensive for a side project, it aligns with my primary research focus on the radical right and hence involves very little extra work.

The bibliography is therefore shaped both by my own research interests and by input from the research community. Since 2010, the bibliography has accrued 1231 additional entries, averaging 82 per year. However, most of this growth has occurred since 2017, when the bibliography comprised only 615 titles, reflecting the field's recent expansion and the rising interest in the bibliography, to which the bot has contributed.

Implementation of the bot

RRRR is mostly written in the R programming language (R Core Team 2024). Its full source code is available from https://gitlab.rlp.net/arzheim/rrresrobot. All line numbers below refer to version 1.0.9, which can be accessed here: https://gitlab.rlp.net/arzheim/rrresrobot/-/tree/version-1-0-9?ref_type=tags

To achieve better performance and reliability, the program is split in two parts that are installed on two different computers: a server that is always connected to the internet and an ordinary PC that holds the repository on which RRRR is based. Significant preparatory work is done on the PC, while the final posts are assembled on the server.

Server side

A single script (`RRResRobot.R`) that resides on a cheap Virtual Private Server (VPS) with a fairly minimal R installation is responsible for the actual posting. All information about the references is stored within four 'tibbles' (upgraded data frames, see Wickham et al. 2019) that the script reads from disk at startup (see lines 19-24). The most important of these tibbles contains a record for each entry in the bibliography. Each record in turn consists of a unique (BibTeX) key, a pre-formatted version of the reference as a string, a variable that holds the year of publication, and, optionally, links to the locations of up to three PNG images (the typeset abstract, a wordcloud, and a typeset single-sentence summary of the text) on disk. The secondary tibbles link the BibTeX key to (optional) lists of relevant countries, parties, and topics, which are required to put the reference into context.

After reading the tibbles, the script pulls a random record from the list of bibliographic references. The current version of the bot uses sampling weights proportional to $\frac{1}{\ln(\text{age}+1.5)}$, where age is simply the difference between the current and the publication year (line

246), so that more recent research has a higher probability of being selected. Once per year, after the campaign to solicit new references, an additional filter that selects only titles published in the current and the previous year is applied for a couple of weeks (line 29–32).

Next, `sample()`, alongside a series of `if`/`else if` statements, is used to randomly select a *type* of post (from line 36). Probabilities are chosen so that a small fraction of posts simply contain general information about the bibliography and its bot. Other posts contextualize the selected bibliographic reference with information on the countries, parties, or more general topics discussed in the respective text, or, in the case of publication anniversaries, highlight the fact that it was published 10, 15, or 25 years ago. Where available, the script also attaches one of the aforementioned images. Throughout the script, further calls to `sample()`, in conjunction with the *glue* package (Hester and Bryan 2024), provide additional variation.

The actual process of posting is trivial but has been made much more difficult by the changes introduced after Musk's takeover of the platform. RRRR previously relied on package *twitteR* (Gentry 2015), which became obsolete with the new API and the more restrictive new free tier. However, under the new 'Free' plan, it is (currently) still possible to register a single app to gain limited access to newer versions of the Twitter API. The current developer's version (2.0.0.9000, available from https://rdrr.io/github/ropensci/rtweet/) of package *rtweet* (Kearney 2019) is then able to post programmatically using stored credentials – something that was previously taken for granted but temporarily became impossible.[1] Posting text with *rtweet* requires three steps (from line 326):

```
client_id <- Sys.getenv("client_id")
client_secret <- Sys.getenv("client_secret")

client <- rtweet_client(app = "RRRRobot",
                        client_id=client_id,client_secret=client_secret)
client_as(client)
```

using the credentials received[2] when registering the app on Twitter generates a client object. This object is then set as the default client for future posts with `client_as(client)`. A call to `tweet_post(text="my tweet")` finally posts the text.

However, `tweet_post()` on its own cannot post tweets to which images are attached. Posting images, which used to be trivial, now requires auxiliary access to the *old* Twitter API via two additional packages, *httr* (Wickham 2023a) and *httr2* (Wickham 2023b), as well as function `create_token()` from *rtweet*, which is marked as deprecated. Also required is package *base64enc* (Urbanek 2015) for encoding images. The following stanza (starting at line 352), which is adapted from code found at https://github.com/ropensci/rtweet/issues/778, currently works:

```
# try to create token for 1.0$
your_token <- create_token(
  app = app,
  consumer_key = consumer_key,
  consumer_secret = consumer_secret,
  access_token = access_token,
  access_secret = access_token_secret)
```

```
# Read and encode image
image_data <- readBin(media, "raw", file.info(media)$size)
image_base64 <- base64encode(image_data)

# Twitter API endpoint for media upload
media_upload_url <-
  "https://upload.twitter.com/1.1/media/upload.json"

# Create the API request
response <- POST(
  url = media_upload_url,
  config(token = your_token),
  body = list(media = upload_file(media)),
  encode = "multipart"
)

# Parse the response
parsed_response <- content(response, "parsed")

# Extract media ID from the response
media_id <- parsed_response$media_id_string

# Now post the tweet
tweet_post(text="my tweet",
  media=list(media_ids = list(as.character(media_id)))
)
```

The cron facility on the VPS starts the server-side script every four hours. A single run of the script typically completes in under five seconds.

Repository side

The repository with which RRRR works comprises a single large (approximately 21,000 lines of text) BibTeX-file containing 1622 bibliographic references, and 944 PDF files. It is stored on an office PC that is backed up regularly.

The bot's repository side is a motley collection of scripts, most of them also written in R. The most important one (`builddataset.R`) is quite short and simple. From the BibTeX-file, it constructs the main tibble that is used by the server-side script. For this, it relies heavily on the *RefManageR* (McLean 2017) package, which provides powerful functions for reading, writing, formatting, and managing BibTeX-files. The `ReadBib()` function converts the flat BibTeX-file into an R object, from which the tibble is built. Using the `saveRDS()` function, this tibble is then serialized and written to disk for subsequent fast access by the server side script.

A second R script (`extract-countries-topics.R`) builds the lookup tables which store contextual information about the references. The *quanteda* package (Benoit et al. 2018) provides the robust infrastructure that is required for dealing with textual data. For each reference, the script first merges the title with the abstract (where available) into a single string. After filtering out references in other languages than English and removing common stop words, these strings are then stored in a corpus structure for further processing.

Identifying relevant countries is straightforward with the `tokens_lookup()` function and a dictionary of country names and demonyms: all strings are split into tokens, and if one or more of these tokens (words) appear in the dictionary, the reference is labelled with the respective country name. For roughly half of the contributions, this method generates a list of one or more countries that can be prefixed to a tweet.

> The #RadicalRight in Denmark: M. H. Nicolaisen. 'From Toleration to Recognition: Explaining Change and Stability In Party Responses to the Danish People's Party. In: Comparative European Politics 21.6 (2023), pp. 799–816. http://dx.doi.org/10.1057/s41295-023-00338-4'

is a typical example. The same procedure is repeated with a different dictionary that contains names and acronyms of radical right parties, resulting in tweets like this one:

> Are you interested in the #AfD? A. Heinze and M. Weisskircher. 'How Political Parties Respond to Pariah Street Protest: The Case of Anti-Corona Mobilisation in Germany'. In: German Politics 32.3 (2023), pp. 563–584.

Here, the relevant party is only mentioned in the abstract, but that is enough for the algorithm to apply the hashtag.

Many texts in the bibliography are comparative and explicitly mention more than one country/party. This is not a problem, because the script stores this information in lists. If many country or party names are mentioned in a text, RRRR tags only a random selection of these to keep the post reasonably pithy.

Of the 1622 references, 263 are currently tagged with 'Germany', 142 with 'France', 122 with 'Netherlands', and 116 with 'Italy'. In total, 35 countries are recognized. The list of parties which are tagged reflects this geographic composition: the Front/Rassemblement National and the AfD lead the field with 100 publications each, followed by the Italian Lega, the Sweden Democrats, and 23 others.

This dictionary-based approach works well for countries and parties, because they are associated with a small set of very specific words. Identifying the primary substantive focus for each text using just the words contained in the title and abstract (if one exists) is a more complex task. To solve it, the script makes use of a version of the Newsmap algorithm (Watanabe 2018) that is implemented in *quanteda*.

Newsmap is a semi-supervised procedure that was originally developed for geographically classifying tokenized news articles on the basis of words that signal probable locations. Tokens are first labelled with the help of a dictionary of place names. For instance, the dictionary entry "FR': [France, French*, Paris]' will label the tokens 'France', 'Paris', and any token beginning with 'French' (e.g. 'Frenchmen') with the ISO code for France. The algorithm will then use this information to 'learn' about additional tokens (e.g. 'Macron' or 'Élysée') that frequently appear together with the seeds and are hence also associated with France, and will use this information for a relatively accurate classification of texts that do not need to contain the seed tokens. Because each classification is based on multiple tokens and the associated probabilities, the algorithm can usually identify the primary location even if other countries are also mentioned in the text (e.g. because a foreign head of state is visiting the Élysée).

But Newsmap *also* works surprisingly well for the classification of short texts into categories that have nothing to do with geography. RRRR currently distinguishes 14 different

topics, using seeds such as '[econom*, unemploy*, recess*, austeri*]' to identify e.g. texts that chiefly discuss the effect of economic factors on the radical right vote. Similarly, the seeds '[women, gender, men]' are used to find contributions which primarily discuss gender issues in radical right support. In the latter case, Newsmap's estimates show that additional tokens such as 'gendered', 'sexism', 'masculinity', 'sexuality', 'men's', 'männerparteien' and 'antidemocratic' are also indicative of a high probability of membership in this second cluster. While 'antidemocratic' appears quite generic, the other tokens are obviously closely related to the subject, even though they were selected without human intervention.

Compared to unsupervised procedures (topic modelling), Newsmap has the advantage of incorporating domain knowledge, as the number and content of topics are specified by the user. At the same time, the inductive use of additional tokens makes Newsmap much more powerful than a purely dictionary-based approach. In RRRR's current setup, 60 seed tokens (about four per topic) are enough to assign all references to a primary topic in a way that is generally plausible. 'Populism' (473), 'immigration' (215), and 'attitudes and personality' (171) currently top the list.

Because of the dictionary's small size, it is very easy to amend and expand this set of seeds to accommodate new topics or correct obvious misclassifications. However, an inductive approach based on large language models that may be able to identify new and emerging research topics could complement or even replace Newsmap in the future.

Besides the two main scripts, there are several auxiliary scripts which produce the images that can be attached to posts. One script uses the *RefManageR* and *purrr* (Wickham and Henry 2023) packages to extract the abstracts from the master BibTeX-file and export them as a large set of plain text files. A shell script then uses LaTeX to typeset these as a series of PDF-files, which are subsequently cropped and converted to the PNG format using the *ImageMagick* suite of tools.[3] Yet another script uses the *lexRankr* package for R (Spannbauer and White 2019) to generate the one-sentence summaries from the abstracts, which are then also typeset and converted. For articles and chapters, their PDF-files (where available) are converted to ASCII with the *pdftotext* utility available on Unix systems. From these, an R script creates wordclouds (also stored as PNG-files) with *quanteda*'s `textplot_wordcloud()` function after applying tokenisation and removing stop words.

Most of these tasks can run in parallel. Moreover, most tasks are related to one and only one reference. This is important, because changes to the bibliography are always incremental. If references are added to or amended in the master BibTeX-file, there should be no need to re-generate the images pertaining to all the *other* references that were *not* modified. The process is hence similar to compilation projects in software development, and so SCons, a build automation tool, is used to manage all updates to the repository.[4]

Whether this is always significantly more efficient than a monolithic script that would simply rebuild everything unconditionally is a different question: even on a machine with enough memory, solid-state disks, and multiple cores, starting many interpreted scripts consecutively incurs substantial overheads. Moreover, RRRR is not updated more than once per month. Re-generating everything from scratch with a modest degree of parallelization currently takes a few hours, but the process could easily be run in the background or overnight.

Most importantly, however, attempts to describe the dependencies amongst its different parts in SCons-syntax have become somewhat convoluted as the system grew, resulting in some runs that could potentially be avoided. Decluttering the build process therefore remains a medium-term goal for RRRR's development.

Currently, the various scripts create 3872 PNG files, which occupy almost 400 MB of disk space. At the end of each update, the *rsync* utility is used to transfer them and the tibbles to the VPS in an efficient manner, as *rsync* only copies files that are new or have changed since the last upload.

Challenges

Every one-person project raises questions about its sustainability. However, the bibliography is very much a byproduct of my regular research, and RRRR's computational and financial costs are fairly minimal: in principle, not even the VPS is strictly necessary as the server-side script could also run on the office PC if that were left switched on overnight.

A much bigger concern are the vagaries of the social media landscape. While the core of RRRR's code base has essentially remained unchanged since December 2018, the aforementioned changes to Twitter's API meant that RRRR did not work reliably for several months in 2023 and 2024. Even worse, Twitter could completely cut off hobbyist projects such as RRRR at any moment, or introduce new API changes that would require updates to the R packages on which RRRR depends.

More generally, Science Twitter is rapidly declining and may soon disappear completely. When the Musk-led consortium bought Twitter in 2022, a significant number of scientists moved to Mastodon, a decentralized and non-commercial alternative to Twitter with a somewhat quirky, early-internet ethos. While this was enough of a disruption to leave scientific communities fragmented, Mastodon use never became a mass phenomenon. Similarly, Bluesky, another alternative to Twitter, initially grew even slower than Mastodon, as it operated on an invitation-only basis for a year.

When this changed in February 2024, Bluesky began to attract a mix of scientists, journalists, politicians, and practitioners reminiscent of old Twitter. Later that year, Musk's intervention in the US election campaign resulted in yet another, much larger migration from Twitter to Bluesky. Currently (July 2025), Bluesky has 36 million users, twelve times more than when it dropped the invitation requirement, and many of the personal networks that made up Science Twitter have re-appeared here. These developments are encouraging, and Bluesky claims to be committed to good governance and decentralization. However, Bluesky's status as a privately owned company leaves it open to external acquisition and unpredictable policy changes.

Nonetheless, in response to (Science) Twitter's deterioration, RRRR set up profiles on Mastodon in November 2022 and on Bluesky one year later. Currently these accounts simply mirror the posts of RRRR's Twitter profile, via a self-hosted installation of *touitomamout*, a cross-poster implemented in NodeJS.[5] While cross-posting works well most of the time, it is not clear how long I can (and want) to sustain RRRR's presence on Twitter. One of the goals for RRRR's near future is therefore to switch over to direct access to the Bluesky and Mastodon APIs via the *atrrr* (Gruber, Guinaudeau, and Votta 2024) and *rtoot* (Schoch and Hong Chan 2023) packages.

Summary and outlook

For better or worse, social media has become a well-established avenue for science communication. This includes both communication within and across epistemic communities, and communication between scientists and a broader public. By showcasing contributions to a field of inquiry, well-intended and well-behaved bots such as RRRR can play an important role in both spheres. Ideally, they raise awareness of the existing body of knowledge, help to provide context, and foster serendipitous insights and new connections amongst researchers.

Scripting languages like R or Python are ideally suited for programming such bots, as they provide powerful packages for text processing and access to the APIs of text-based social networks. Moreover, because many scientists are already familiar with these languages and because they require only elementary coding skills, interested researchers can focus on their domain expertise and knowledge of the relevant communities when setting up a bot.

Scientists need to be mindful, however, that they are usually just users and unpaid 'content creators', but not owners of social media platforms.[6] While platforms may be useful for a while, they can change their terms, fall out of fashion, or simply shut down without warning. Bot developers must therefore be ready to adapt to change, yet ought not invest *too* much of their time in such projects. And above all, they must not take their creations too seriously.

Notes

1. With slightly older versions of *rtweet* available from CRAN, users on the 'Free' plan are required to manually re-authenticate in a browser every couple of hours, which is obviously impractical for a bot.
2. Note that for security reasons, credentials are stored in the environment and not in the script itself.
3. See https://github.com/ImageMagick/ImageMagick.
4. See https://github.com/SCons/scons.
5. See https://github.com/louisgrasset/touitomamout.
6. In recognition of this, some institutions have set up their own Mastodon instances.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

Arzheimer K. 2025. "Bibliography on the Extreme Right in Western Europe." Accessed June 30, 2025. https://www.kai-arzheimer.com/extreme-right-western-europe-bibliography/.

Benoit K., K. Watanabe, H. Wang, P. Nulty, A. Obeng, S. Müller, and A. Matsuo. 2018. "quanteda: An R Package for the Quantitative Analysis of Textual Data." *Journal of Open Source Software* 3 (30): 774. https://doi.org/10.21105/joss.00774.

Gentry J. 2015. *twitteR: R Based Twitter Client*. R Package Version 1.1.9. https://CRAN.R-project.org/package=twitteR.

Gruber J. B., B. Guinaudeau, and F. Votta. 2024. *ATRRR: Wrapper for the 'AT' Protocol behind 'Bluesky'*. R Package Version 0.0.4. https://CRAN.R-project.org/package=atrrr.

Hester J., and Bryan J.. 2024. *glue: Interpreted String Literals*. R Package Version 1.7.0. https://CRAN.R-project.org/package=glue.
Insall R. 2023. "Science Twitter – Navigating Change in Science Communication." *Nature Reviews Molecular Cell Biology* 24 (5): 305–306. https://doi.org/10.1038/s41580-023-00581-3.https://doi.org/10.1038/s41580-023-00581-3.
Kearney M. W. 2019. "rtweet: Collecting and Analyzing Twitter Data." *Journal of Open Source Software*4 (42): 1829. https://doi.org/10.21105/joss.01829.
McLean M. W. 2017. "RefManageR: Import and Manage BibTeX and BibLaTeX References in R." *The Journal of Open Source Software* 2 (16): 338. https://doi.org/10.21105/joss.00338.
Arzheimer K. 2008. *Die Wähler der Extremen Rechten 1980–2002*. Wiesbaden: VS Verlag für Sozialwissenschaften.
R Core Team. 2024. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.
Schoch D., and C. Hong Chan. 2023. "Rtoot: Collecting and Analyzing Mastodon Data." *Mobile Media & Communication* 11 (3): 575–578. https://doi.org/10.1177/20501579231176678.
Spannbauer A., and B. White. 2019. *lexRankr: Extractive Summarization of Text with the LexRank Algorithm*. R Package Version 0.5.2. https://CRAN.R-project.org/package=lexRankr.
Urbanek S. 2015. *base64enc: Tools for Base64 Encoding*. R Package Version 0.1-3. https://CRAN.R-project.org/package=base64enc.
Watanabe K. 2018. "Newsmap." Digital Journalism 6 (3): 294–309. https://doi.org/10.1080/21670811.2017.1293487.
Wickham H. 2023a. *httr: Tools for Working with URLs and HTTP*. R Package Version 1.4.7. https://CRAN.R-project.org/package=httr.
Wickham H. 2023b. *httr2: Perform HTTP Requests and Process the Responses*. R Package Version 1.0.0. https://CRAN.R-project.org/package=httr2.
Wickham H., M. Averick, J. Bryan, W. Chang, L. D'Agostino McGowan, R. François, G. Grolemund, et al. 2019. "Welcome to the Tidyverse." *Journal of Open Source Software* 4 (43): 1686. https://doi.org/10.21105/joss.01686.
Wickham H., and L. Henry. 2023. *purrr: Functional Programming Tools*. R Package Version 1.0.2. https://CRAN.R-project.org/package=purrr.