R Consortium
2017 Community Survey

Selected questions summary

Mark Hornick, Oracle

Introduction

While the 2017 R Consortium Survey contains a rich set of structured responses, some of the most telling responses are free-text. This document summarizes three free-text responses to the 2017 R Consortium Survey, specifically:

- What is the best aspect of working with R?
- What is the worst aspect of working with R?
- What else do you think the R Consortium should be doing for you?

From this, a set of overall *opportunities* is extracted based on survey responses to these questions. Subsequently, each question's responses are characterized individually, first with a word cloud based on the TF-IDF ranking of words in the responses, and then followed by a summary of observations extracted from statements containing some of the top words.

Since any one perspective is bound to omit or emphasize content that others may not, this is viewed as a launch point for discussion with the lightly edited individual responses containing the "word of interest" also provided. (Note: only those responses matching the specific keywords searched are included and were reviewed for the summary.) While many responses within each question are repetitive, this repetition gives a feel for consensus among respondents. Other responses add significant insight or color to perceived needs of the R community and ecosystem beyond the provided summary, so browsing these may be useful.

Table of Contents

Introduction 2
Opportunities Summary 5
Question 27: What is the best aspect of working with R? - Observations 7
- General 7
- The community and ecosystem 7
- R Environment 8
- Packages 8
- Language 8
- Resources 8

Question 28: What is the worst aspect of working with R? - Observations 9
- General 9
- Performance 10
- Deploying R in a company 10
- Error messages 10
- Syntax and function naming 11
- Documentation 11
- Memory 11
- Enhancements 11

Question 31: What else do you think the R Consortium should be doing for you? - Observations 12
- General 12
- Events 13
- CRAN / Packages 13
- Development environment 13
- Documentation 13
- Standards 14
- Commercial / enterprise support 14
- Performance / scalability 14
- Code quality 14
- Python 14
- Usability 14
- Bugs 15
- Enhancements 15
- Education 15

Question 27: Supporting statements 16
- Community 16
- Flexibility 20
- ggplot 24
- tidyverse 25
- RStudio 28
- Statistical 31

Question 28: Supporting statements 36
- Cryptic 36
- Error message 37
- Learning curve 39
- Memory 43
- Parallel 47
- Syntax 48
- Slow 52
- Bug 57
- Install 59
- Big data 61
- Production 61

Question 31: Supporting statements 65
- Roadmap 65
- Commercial 65
- Enterprise 65

Error 65
Package 65
CRAN 67
Training 68
Education 68
Tutorial 69
Course 69
Fund 69
Internationalization 70
Community 70
Documentation 71
Project 72
Website 73
Pain 73
Know 73
Python 75
Support 75
Standard 77

Opportunities Summary

The following topics and suggestions are derived from the subsequent individual questions. While some of these may already be underway through the R Foundation or R Consortium, or perhaps even exist in the R ecosystem, they pose opportunities to communicate the availability of such capabilities to the broader R community in blog posts, a newsletter, and/or press releases. In other cases, some of these could result in, e.g., RC ISC projects or RC Marketing Committee activities.

- Roadmap and vision
    - Specify and circulate a roadmap for R that is visionary yet realistic
    - Address concern over sustainability of R
    - Explain how to transition technology to younger contributors, e.g., Fortran knowledge is waning
    - New or revised features and functionality, development environment and infrastructure
    - Devise, implement, and share a plan to promote R in companies, organizations, universities, high schools
- Produce an R newsletter
    - Identify areas/topics of interest to the R community
    - Solicit articles from the community
    - Communicate better what the R Consortium is about
        - Only 679 of 3618 felt they understood the difference between RC and R Foundation; 2089 did not understand the difference!
        - Only 903 of 3618 felt they understood what the RC does; 13029 did not; 1382 were uncertain
        - 2382 of 3618 have heard of the RC, but 862 have not
- Identify features missing from core R to maintain benefits over Python (see details later)
- Redesign the CRAN website to make it more modern
- Adopt tidyverse as main/core programming paradigm for R
- Packages
    - Put process in place for achieving consistency, simplification, consolidation, and quality control of packages and particularly package documentation
    - Address package search issues
        - Users' ability to find packages they need with indication of quality/currency of package
        - Provide logical grouping or tree of functionality beyond task views
        - Too many packages on CRAN / GitHub – how to organize better
    - Provide ability to download consistent set of packages
        - Relative to a specific R version
        - Relative to a specific point in time
        - Relative to a specific package version
    - Introduce certification of R packages along various criteria with "badges" posted on CRAN
    - Provide guidance on "best practices" to enable parallelism in CRAN packages, e.g., writing re-entrant code
    - Better sync of packages between github and CRAN – bug fixed versions not always or immediate propagated to CRAN – users sent to github for fixed version
- Address issues negatively affecting corporate use of R
    - Security – expand blog post into whitepaper to answer "How to implement R in a secure way?" "Why R is safe for secure environments?" "What safeguards are in place to promote R software safety?
    - Licensing
    - Package quality / testing certification
    - Ease of deployment
        - R itself
        - R scripts in production
        - Package installation
    - Introduce certification of R packages along various criteria, e.g., suitability for industrial or big data use

    - Highlight proprietary solutions of RC member companies that already extend R for corporate use – addressing various issues cited in this document
- Language consistency issues
    - Launch project to revise function and argument naming for consistency / unity
        - Issue recommended guidelines or standards going forward
        - Provide a plan to evolve select aspects of legacy R code to new standard
    - Deprecate outdated modes of programming
    - Launch project to provide best practices for using R
    - Explore whether the language should be "standardized" and what this would mean for the community
    - Provide a full language specification so other R implementations can work from a document and not introduce GPL licensing of R (Python has such a specification, and more lenient licensing)
- R performance and scalability
    - Rewrite algorithm packages to enable parallel and/or non-memory resident execution
    - Identify needed minimum performance for specific data size (# rows, # columns) for different algorithms / functionality
    - Enable better parallelism across core R – parallel package is viewed as unstable or unreliable
    - Provide support for multi-threading and GPU usage
    - Look at alternative R implementations that may have already addressed some of these issues
- Revise error messages to improve usefulness and clarity, and aid debugging more effectively
    - Across core R
    - Best practices for package developers
    - Solicit community for examples of poor error messages in R and correct them
- Documentation
    - Identify worst core R documentation and revise / republish
        - Define "worst": cryptic descriptions, examples too simple/complicated
        - Solicit community feedback to identify packages most in need of rewrite or improvement
    - Identify functionality that would most benefit from (additional) vignettes
    - Cite high quality material in a common place and issue awards (plaque with gift)
    - Launch a project to internationalize R documentation
        - Prioritize languages, perhaps starting with Spanish
- R Consortium
    - Solicit broader input from community on what projects should be funded
        - Introduce survey / voting
        - Not final determination, but input to the process
    - Provide lower level entry fees for small companies to join
    - Introduce more frequent grant cycles, perhaps quarterly
    - Greater funding for local user groups
    - Sponsor more conferences

Question 27: What is the best aspect of working with R? - Observations

General

- The community
- Open source and free
    - Easy to contribute to
- Collaboration
    - Share results via Rmarkdown
    - Easy to work as a team
- Tools
    - Breadth of tools
    - RStudio IDE and server
- RStudio

The community and ecosystem

- Very important to R users (more so than functionality, some of which is available elsewhere)
- Very active group
- Helps users learn and use R
- Quick response time to questions online
- Very supportive, welcoming, friendly, knowledgeable
- Meetups and RUGs
- Constant state of excitement and encouragement

R Environment

- Rich functionality
- Fast and easy to install and get started
- Frequent updates
- Easy to use
- Free
- Flexibility and the ability to go from start to finish for an entire project without leaving the R environment
- Work interactively with R, while also using it as a more formal programming language
- Stable environment
- Speed of code execution

Packages

- Breadth of functionality
    - Powerful and quality statistical and machine learning algorithms / methods
    - Innovative algorithms
    - Modern statistical methods are either initially or exclusively implemented, i.e., not in Python
    - Exploratory data analysis is very quick to perform
- Frequent addition of new and useful packages
- Tidyverse: ggplot2, dplyr, tidyr
    - "debugging can be a bear"
- Ease of data manipulation
- Visualization capabilities
    - Produce beautiful plots
- Rstan
- knitr

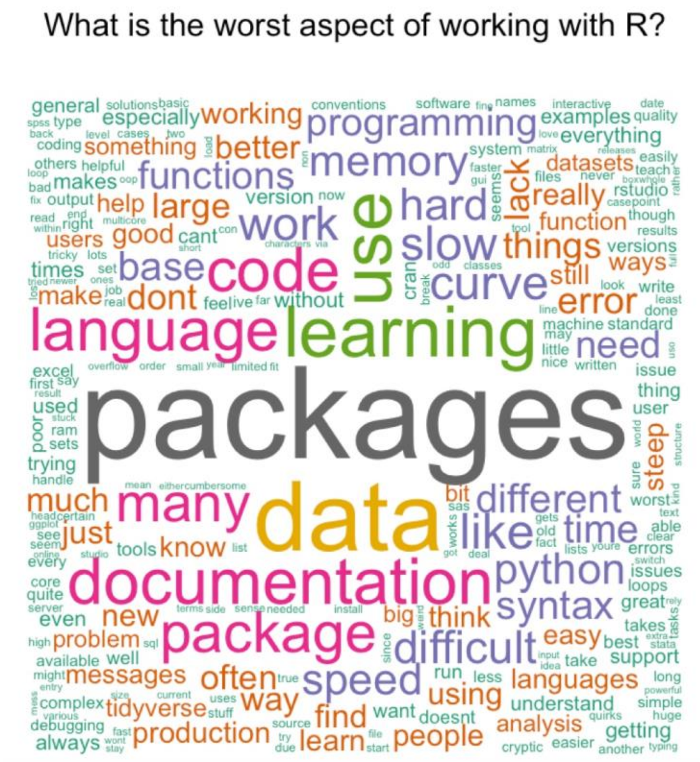
Language

- Powerful
- Flexible
    - Easy to customize
    - Extensible
    - Define your own functions and packages
- Ability to prototype quickly and easily
- Simple commands with good defaults to produce high quality results

Resources

- CRAN
- Bioconductor
- StackOverflow
- Ability to google an answer to most any question/problem in R
- Github
- RBloggers
- Forums
- #rstats

Question 28: What is the worst aspect of working with R? - Observations



General

- Hard to learn – steep learning curve
- Not a general programming language – non-statistical tasks are cumbersome
- Need an OpenRefine for R
- Worrying that the core language isn't developing fast enough and that R will eventually die / that it's losing to Python
    - It is hard for beginners to get into because of inconsistency
    - Concern that R isn't enough. Some R users learning Python, begrudgingly
    - Missing out on python's web programming capabilities
    - Difficult to handle large datasets in R (in comparison to Python)
    - Python is much more popular and easy to use with web data
    - Python has unified machine learning algorithms package, like sklearn
        - Consider official R version of common algorithms that receive hot fixes/patches
- Having to leave the tidyverse
- Security checks required for corporate use make using R for projects difficult
- Licensing of packages for corporate use makes using R for projects difficult or a non-starter
    - Must turn to Python for corporate projects
- Painful to work with dates
    - Having to get packages security-checked before they can be used on our desktops, and the security/privacy blocks from our corporate policy that means we can't use it in the cloud with our sensitive data
- Difficult to find the right package, or know which to use/trust

    - Many packages available but many use outdated data structures and are very slow on large data, requiring customization
    - Tons of functions that I have no idea how to use or why I would use. Number of packages on CRAN and GitHub also seems overwhelming.
- Sometimes, code runs in one environment (e.g., OS) but not in another
- Not being able to use Latex syntax and fonts directly into the generated plots. This should be possible and implemented like in Python's matplotlib
- Confusion with rmarkdown: Sometimes you code in LaTeX, other times the rmarkdown syntax, other times HTML can pop in there. Confusing to use multiple languages in one document without a clear delineation of what is going to work and when.
- It's annoying when a slightly old piece of code won't run because a minor change has been made to R core or a package
    - Lack of backward compatibility
    - "I'm even ok with the clunky syntax (relative to Python, for example), but the inconsistency of the clunkiness is a pain sometimes"
- Lack of integration with Cloud environments, (e.g. dynamoDB/AWSlambda)
- Bugginess of package interactions
- "Crippled developer environment (svn instead of git, who uses bugzilla that you cannot self register?)"

Performance

- Slow speed
    - Algorithm performance very slow for larger data
    - Especially compared to Python
    - Some users switch to Python due to algorithm performance
    - Better performance / support for looping – slower than Python or C
- Scalability for Big Data (count of respondent answers)
    - 1,000 or fewer records - 459
    - 1,001 to 10,000 records - 871
    - 10,001 to 100,000 records - 828
    - 100,000 to 1 Million records - 811
    - 1 to 10 Million records - 495
    - 10 Million to 100 Million records - 20
    - More than 100 Million records - 76
- Lack of parallelism in general operations
    - Non-existent support for multithreading, multicore, gpus
    - Terrible parallel processing support lacks a ML pipeline like scikit learn
    - The parallel package sometimes can make the whole R session crash
    - Hard to terminate a parallel job before it's finished
    - Parallelism (multicore) doesn't work by default

Deploying R in a company

- Difficult to put R solutions into production
- Installing a package may result in many more subpackages being installed, taking a long time
    - Text mining packages are a real challenge in this area
- "At Work it's a nightmare to install and maintain the packages without administrator rights. Generally, the IT department cannot help a lot."
- Requirement to update other packages when installing a new one may produce unwanted side effects in a robust, production environment
- Packrat is amazing, but it's very memory expensive, since it makes a full copy of R to the repository
- Need pre-built packages with easy version consistency among subpackages
- Total reproducibility is difficult due to many versions of packages and R itself
    - E.g., Bioconductor changes so often that it is near impossible to have reproducible research
- The need to reinstall all needed packages after installing a new version of R is painful
    - Automatically install packages from previous version of R, but with latest versions

Error messages

- Cryptic, obtuse, ambiguous, hard to understand

- Debugging is very difficult
- Lack of good debugging tools
- Traceback() often leads nowhere

Syntax and function naming

- Make more consistent in formatting
    - Base R differs from many popular packages, which makes it difficult for new users
    - Object-oriented vs. functional, tree vs. tabular structures
    - Enable more natural OOP as found in Python
- Syntax often cryptic or inconsistent – too many ways to do the same thing
    - Core R syntax is very inconsistent and fragmented. Makes learning R difficult
- Lack of consistency between packages
    - Each has a different way of working and using it
    - Too many syntax or function definition styles, need to standardize
- Too many assignment operators
- The worst aspect of R is layers of legacy functionality, kept for backward compatibility
- Missing syntactic sugar. A good list of these is provided by Jim Hester (https://rawgit.com/jimhester/presentations/master/2017_07_03-DSC2017-Syntax_Extensions_To_R/2017_07_03-DSC2017-Syntax_Extensions_To_R.html#/r-syntax-extensions)
- Many features would be better off with a traditional OOP implementation - e.g. date time objects (POSIXct). Odd syntax at times: assignment "<-", "." allowed in names, "$" as member access, "c()"."
- Function names are inconsistent: row.names, rownamesrownames, rowMeans, rowSums, rowsumbrowseURL, contrib.url, fixup.package.URLspackage.contents, packageStatusmahalanobis, TukeyHSDgetMethod, getS3methodread.csv and write.csvload and savereadRDS and saveRDSSys.time, system.timecumsum, colSumsFunction
- Arguments are inconsistent:aggregate(.., FUN=mean, ...)reorder(..., FUN=mean, ...)reorder.dendrogram(..., agglo.FUN=mean)reshape2::acast(..., fun.aggregate=mean, ...)promptData(, filename= ,)write.table(, file= ,)# Text rotation argument is las. Or maybe srt. Or maybe rot.

Documentation

- Examples are too simple or too complicated
- Cryptic descriptions
- "Unfriendly"
- Matlab and Python have much better documentation

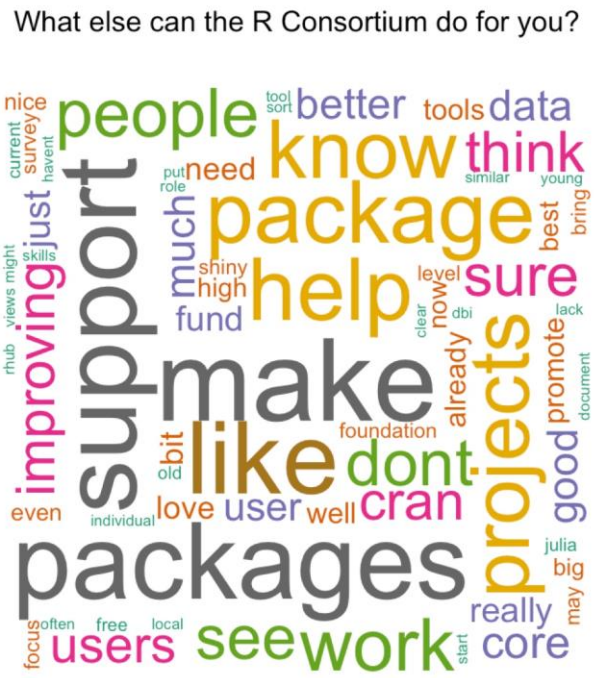
Memory

- Need standard way to deal with out-of-memory data / big data
- Better memory management
- Enable incremental loading/processing of data in chunks
- Crashes when runs out of memory instead of graceful error message
- Memory copies in loops.do.call() can use up all memory and grind to a halt

Enhancements

- Replace split-apply-combine with dplyr
    - "The split-apply-combine approach is absurdly hard when you compare it to something like a PivotTable in Excel. Thank the lord for dplyr."
    - As tidyverse overtakes many base functions, core / base r should reflect what people actually use and what is best for the future of the language
- Explain 'not run' in documentation
- Make vignettes more visible and encourage development and use
- Needs a good interactive plot designer
- Even heavily downloaded packages such as MSBVAR contains serious bugs that drive the results and nobody talks about it or fixes them
- Package support of R version – need time capsule configuration capability

Question 31: What else do you think the R Consortium should be doing for you? - Observations



General

- Publish a roadmap for R
    - Clear plan for future of the Core language (including a potential need for drastic changes if some historical decisions are seen to hinder progress)
    - Be "visionary" for R
    - Better mechanism to provide feedback to development community for future R development
- Promotion of R
    - Promote R in companies, organizations, school, universities
    - Send an R newsletter
- Focus on improving base R, not just adding new packages
- Concern over sustainability of R given demands on R Core members and CRAN volunteers
- Central website for RUGs, with wiki / bb to share ideas and experiences, interact with R Foundation and core group
- Perceived attitude of R Core Team – "nobody wanted to muck with the r-core team for fear of being ripleyed (not by brian, there are other ripleys now)"
- Solicit broader input from the community on what projects should get funded, not just the BoDs
    - Perhaps introduce a survey for non-binding community input / votes
- Fund greater support of data science capabilities in R
    - Deployment of R models
- Communicate funding opportunities that can be shared with various R communities, e.g., RUGs, Meetups, etc.
- Fund workshops for teaching undergraduates – small awards
- Start a project to facilitate R-related employment opportunities
- Best practices for maintaining a lively local R community
- Consider the role of a formal "community historian"
- Take a more active role in the global R community

- Foster more teamwork and collaboration across the R community
- Help promote R as a valid language (as python) to use in IT departments and in production
    - It is sometimes perceived as just a language for prototyping and for research
- Improve RC communication to the community on projects and priorities, better visibility
- Promote projects beyond those from RStudio
- Support for technology transfer from Academia to Business, for example, a research group helps companies using Statistical Process Control with R, the consortium could have a funding program for such short applied projects
- More frequent grant cycles
- Identify and fix pain points, gaps in R ecosystem
- R Consortium Membership
    - Provide lower level entry fees for small companies to join

Events

- User Groups
    - Provide funding to RUGs
    - Provide support in organizing events
    - Pay meetup fees for R user groups
    - Continue support for RLadies and other underrepresented groups
- Conferences
    - Support conferences outside of US and Europe
    - More targeted support for regional useR! Conferences - appear more grassroots and ad-hoc. Have more coordinated effort.
    - Sponsor more conferences
- Encourage a certification for packages along various dimensions
    - Licensing
    - Coding standard conformance
    - Software quality
    - Security
    - RC to play a greater role in validating more of R

CRAN / Packages

- Define a process for retiring outdated, non-maintained packages
- Reduce / prevent / manage package functionality duplication
- Ensure well-used packages have maintainer if current one moves on
- Need for consistency, simplification, consolidation, and possibly quality control of packages and particularly package documentation (by supporting CRAN)
- Develop a feature voting system, including for packages
- Fund development of good quality packages with needed functionality (competing with Python)
- Funding package maintainers to improve packages in areas where R's relevance is slipping, such as machine learning
- Easier way to search CRAN packages by topic
    - Provide a map to R packages, e.g., a grouping/tree
- Produce Linux package binaries
- Revamp submission process
- Redesign the website – modernize
    - "It's a bit embarrassing to have such an old system compared to NPM, Pypi, Rubygems, etc."

Development environment

- Host on GitHub for broader access of bugs and contribution
- Make it easier for non-R Foundation Core team to fix bugs listed in BugZilla
- How to transition technology to younger contributors, e.g., Fortran knowledge is waning

Documentation

- Improve quality and completeness of package documentation
    - Cite high quality material in a common place – give awards?

    - Especially base R documentation
    - Strive to reach audiences beyond statistically-minded users
    - Introduce peer-review of package documentation to improve quality, perhaps as part of submission process
- Improve documentation of old functions, especially commonly used ones. These are often written in insider language or to explain the problem code was written to solve, but not how it is actually used by other people. Terms used and shorthand commands, like ( ) to print, are not defined.
- Better documentation for markdown, knit, pandoc templates, parameters and option
- Better documentation format
    - Content of current help files is sufficient, but presentation is not
    - Many attempts, nothing changed - with the exception of vignettes/knitr/markdown support
    - pkgdown is a good concept, output format should be standard
- Internationalize documentation, driven by user demand / need
    - Start a project and get people to volunteer to translate documentation

Standards

- Make the R ecosystem more coherent
    - Standardize the R language, like what C Standards committee did, not C++
    - Produce a better language standard, like C++ 11 for C++
    - Unify R package and function syntax / naming, e.g., bioconductor vs. cran, tidyverse vs. data.table
    - Greater emphasis on standardizing R, e.g., like what was done with DBI
    - Unify the API – naming of functions, arguments
    - Homogenize R code naming conventions and structure
- R Consortium to publish recommended guidelines or best practices for the "correct" way to use R
    - How to make a great function or package?
    - How to use other programming languages with R
- Promote quality standards

Commercial / enterprise support

- Make R more commercial-friendly, e.g., by providing suitable licensing
- Integrate with other technologies that are important to industry, e.g., relational databases, SAS, Tableau, Oracle, OCR software,
- Provide support/guidance for enterprise-level use of R, e.g., deployed across an organization
- Provide a guide to industrial strength packages and verification for industry use

Performance / scalability

- Faster data management in R
- Improve R performance overall, perhaps with new base R implementation
- Improve ML algorithm performance
- Better support for larger data sets
- Support Big Data
    - Fund projects to scale CRAN packages on distributed platforms (Spark, Hadoop)

Code quality

- Require more tests for packages – report code coverage

Python

- Improve R where losing ground to Python, e.g., NLP with poor/slow implementation of word2vec, deep learning, reinforcement learning
- Ensure popular package functionality in Python is available in R
- 1595 of 3618 respondents also use Python for data analysis

Usability

- Improve R tooling for popular text editors (vs. code, atom, sublime)

- Improve error messages – better guidance on what is causing the problem
- Better support for package dependency, package versioning, project reproducibility, dependencies on packages – need R-sponsored "time capsule" for getting consistent package state at specific time
    - Make versions dependencies easier to manage to obtain reproducible results
- Something similar to ropensci where they have a steady outreach of how to use their packages together

Bugs

- Address language issues (shiny, decoding), timezone problems, integrate data.table into the base packages.

Enhancements

- Provide a solution for package identifiers masking and for handling simultaneous different package versions.
- Support for ML model deployment
- Remove language inconsistencies (such as list()<-NULL; NA factor level handling;) and fund in-core development of performance features which are addressed by bit, bit64 and ff (lean data types such as bit, INTEGER64 addressing instead of INTEGER32+DOUBLE, out of core data, leveraging sorted caches, move to a JIT compiler; or : merge efforts with Julia)
- Make hacks of the tidyverse un-hacky
- Optimizations in other R implementations (Renjin, etc.) should flow back into vanilla RSupport efforts to improve documentation
- Refurbish the RPub concept, add automatic tags, order the rmd, make it social and collaborative
- Support an upgraded version of opencpu with possibility of login
- Support next generation of R, e.g., with tidyverse as the default interface on a faster platform
- Support underlying computational infrastructure: more general multi-core parallel internals, gpu computing for rasters
- Package support for engineering applications, e.g., thermal analysis

Education

- Invest in teaching / learning resources
    - E.g., conferences, online webinars, learning platforms like Swirl, DataCamp, Coursera, etc.
    - Need for professional R developer training
    - Promote development of and consolidate training material
    - More on-boarding materials to get people stated with R
    - More online tutorials, e.g., video lectures, like MIT or Stanford
    - More online examples
    - How to develop packages
- Provide resources for R user to find / learn new R packages and system developments
- Provide a service to help users with package development and use best practices
- Develop a learning/teaching package to learn R in R
- Expand availability of R training beyond the large urban centers through webinars
- Promote R and training in economically deprived countries
- Provide support/funding for those who run R education projects, e.g., grant resources for education

Question 27: Supporting statements

Community

[1] "I know it's an unpopular opinion, but I think the language is very readable (vs. a full class-based OO language with methods to do everything). The community should not be discounted as a highly valuable aspect of the language."
[2] "Tidyverse and the community"
[3] "I would say the community around which is very active and the number of different things that you can do"
[4] "It is easy to write R code, I can get things up and running quickly, and if I have problems the community is great and it is easy to find help"
[5] "The community around it, the breadth of available packages."
[6] "The community, the availability of tools, the flexibility of open source."
[7] "R's flexibility and the community of users."
[8] "THE COMMUNITY."
[9] "The community"
[10] "The great packages being developed and the community to learn from."
[11] "The community is extremely supportive"
[12] "The community."
[13] "I love the community and the culture of sharing code. I also love the way that developers are continuously pushing the envelope of what can be done using R, particularly in the realm of reproducible research."
[14] "The community"
[15] "The community!"
[16] "1. The community2. CRAN"
[17] "The community"
[18] "The community. The investment people have made in making the packages amazing. Stack Overflow."
[19] "The community"
[20] "The community"
[21] "The community"
[22] "The community of people who are willing to share."
[23] "The community and the shiny package is wonderful compared to trying to make web apps in Python"
[24] "The community. If you have a problem, you get help immediately."
[25] "the community, the rich functionality, the flexibility"
[26] "The community."
[27] "The community. Availability of functionality (graphics, modeling, etc)"
[28] "The community and its willingness to share knowledge"
[29] "It's super expressive for doing data analysis work and the community of user is active and supportive."
[30] "The community"
[31] "The community. I love all the cool things that are being developed and if I have any problems I can usually find a solution without much trouble."
[32] "The community and the availability of so much \"best practice\"-resources. Hadley Wickham's packages - I owe that man so many hours."
[33] "The community! So helpful and welcoming. Power and flexibility also."
[34] "The community"
[35] "the community. IDE RStudio"
[36] "The community. Full stop.Packages, StackOverflow, mailing lists, conferences, user groups, etc, etc."
[37] "The community and the ecosystem"
[38] "You learn something everyday, the community is great"
[39] "the community. ease of use. open source. it is easy to get some first results"
[40] "There are many R people. I love the community."
[41] "CRAN, cutting edge statistical methods, the community."
[42] "the community"
[43] "Always getting help through the community; its ability to expand what you're capable of doing"
[44] "The community! It is supportive, open, welcoming, helpful, etc--there doesn't seem to be a lot of the eye-rolling snark that noobs get when they ask a question in other environments."
[45] "The community!"
[46] "The community."
[47] "Lots of packages with good support from the community"
[48] "Rstudio, the community, the packages"

[49] "How active the community is to provide help and improve R."
[50] "The community"
[51] "The community. There is a high likelihood that someone else has solved a problem that you're trying to solve."
[52] "The community - & the general sense that for any new problem, there is probably a package or some code on GitHub that'll get you half way there."
[53] "The community. I rarely have a problem that can't be fixed with some Googling and if I need some sort of analytic functionality it's generally available via CRAN."
[54] "Ease of finding answers to questions. While the R documentation is not always the best, the community around R is large enough that any question has been answered on a listserv or StackOverflow. And if it has not been, questions are usually answered quickly with good detail."
[55] "The wide array of packages available on CRAN is immensely useful, and one of the reasons I've stuck with R as my primary analysis tool: I have yet to reach an analysis that I *can't* do with R. (Some exceptions of course: still getting R to run properly on our HPC, use Python occasionally for GIS scripts.) The community is also very informative. I've actually never needed to post a question on stack overflow (also, I'm scared to do so) because someone else has already asked a similar enough question and the answers are super detailed, so I can adapt the code fix to my own scripts. The amount of information on stack overflow has saved me possibly DAYS of coding. Finally, people in my field (ecology) have embraced R, so being able to share scripts with contributors, get help, etc. is a huge plus."
[56] "The R community, by far. The language itself is extremely useful, but there are alternatives that could accomplish most if not all of what I do in R. But the community support is irreplaceable; it is rare that I can't get a quick, helpful answer from members of the R community."
[57] "The community, rarely do I encounter an R related issue that is unique. Someone has likely discussed the issue, found a solution, or if I am unable to find any resources between stackoverflow, r-bloggers, and twitter there is always someone willing to help."
[58] "The community and their involvement on other platforms such as Twitter"
[59] "There's a lovely feel to the language. It's broad because it's open-source. And the community is the best programming community out there."
[60] "The community."
[61] "THE COMMUNITY!"
[62] "The community"
[63] "The community. Nothing has encouraged me to continue to use R as my primary tool for data cleaning, visualization and analysis more than the myriad helpful souls who ask and answer questions about R on forums, email lists, stackoverflow and slacks. The amount of times I've discovered exciting new packages or new ways of solving old problems on Twitter through the #rstats hashtag keeps me enthused and interested. The open source nature of R and its generous community make it one of the most positive examples of how science can be a global cooperative endeavor rather than a method to make money behind closed doors."
[64] "The resources and help provided by the community."
[65] "The relentless advance of the language thanks to the community -- always new things to do and learn. R also has lots of great little data packages, which make the first part of analysis easy (getting the data). But the best thing about R is the open-source community built up around it."
[66] "Quick prototyping, easyness of data handling / munging, fun, social aspect of the community (github...)..."
[67] "It's kind of fun. Every day there's something new to try, or an improved package. Mostly it's the community. So much sharing."
[68] "The community is great."
[69] "The community and the wide range of packages."
[70] "Collaboration with others in the community"
[71] "Variety of packages, the community, visualization tools, RStudio"
[72] "The community is amazing. All R people I've met and interacted with on twitter and in real life are super helpful and fun. It's also given me a career! And makes my research possible."
[73] "It is quick and easy to get up and running, and the community around R can't be beat. If I don't know how to do something, I have a plethora of places I can go for help, and get assistance within minutes."
[74] "The community. Welcoming, friendly and helpful."
[75] "many improvements with package like dplyr and other and first the community"
[76] "The community and the written examples one can find on the internet"
[77] "The community behind it, there is always someone willing to help to be found on the internet."
[78] "The community aspect of R is without question the best aspect of R."
[79] "The community. The breadth, depth and generosity of the R contributors continues to amaze me."
[80] "The community."

[81] "The community. And it is fun."
[82] "Its got to be the community and loads of resources available to learn and improve everyday."
[83] "The community willing to help one another."
[84] "Flexibility (packages, techniques etc.), ability to run on a server (with RStudio server) and the community: any question you have is mostly a google away."
[85] "The community on github and stackoverflow."
[86] "R is free and light to install so you can use everywhere, there is always new packages , the community is really great and you can learn lots of things from it. So I would say the best aspect is the community"
[87] "So many projects in so many different fields! The open source aspect allows this and it is great to follow the community."
[88] "Ease of use, the community (both \"industry\" and academic) and evolving ecosystem of packages as well as RStudio IDE."
[89] "The community of other users who share their packages and knowledge."
[90] "It's hard to pick just one thing. I guess the community? There are so many good packages on CRAN, so many good questions and answers on stackoverflow, so many good blog posts across the internet (and all in one place on RBloggers). R would not be what it is today without these contributions from the community."
[91] "The community, especially the opportunity to contribute tools for others to use."
[92] "The community"
[93] "The community"
[94] "The community and the frequent updates."
[95] "Everyone says it because it's true: the community of bloggers, helpers, and package developers makes it the best platform."
[96] "Familiarity customizability and the community support"
[97] "The community."
[98] "The community is very helpful, whether people asking questions on Stackoverflow (using data built into packages as examples), people answering those questions - who often give a deeper explanation than necessary that gives greater insight, the people writing blogs so I can learn more and the people writing the wide range of packages and documentation so I can expand/enhance my toolset."
[99] "The community on stack exchange and the RStudio folks."
[100] "The breadth of algorithms and the community - the BARUG talks and meetups that I have attended have been uniformly excellent, which reflects on the strengths of the community"
[101] "It is a mixture of enabling me to get my job done while also encouraging and enhancing my creativity and productivity. Time spent using R never feels like time wasted. The community also plays a huge role in this, I have found it to be very open and engaging."
[102] "That it's free, and the community of people using R."
[103] "The support fostered by the Community and the variety of packages."
[104] "The community"
[105] "Doing analysis and producing publication quality graphs in the one program. The range of analysis available and the help from the community."
[106] "The community"
[107] "The people involved in the community. Also the ease and convenience of most routine statistical data analysis."
[108] "the community"
[109] "The community."
[110] "The community. It feels more like an ecosystem than just a language."
[111] "I guess the community. R keeps getting better. And it wouldn't if it was just a brilliant one-off that never got honed by thousands of contributions, complaints, bug-fixes, and add-ons."
[112] "The community. There are so many people using R in data science, statistics, and applied fields that I can virtually always find packages to do complex tasks and support for bugs and coding issues. R is also the default platform for releases in my fields (statistics, sociology) so new methods seen in publications are typically ready for me to use. I teach a graduate level R programming course, and, I tell my students, any issue you encounter or problem you need to solve in R is probably solved and well documented online (if you know what to look for). That is a function of a large and mature user base."
[113] "the community and tidyverse"
[114] "The community"
[115] "Open source and free means I can use it at home, for study or for any of the jobs I've had in the past five years. I don't need to go through procurement to install it. I can run it at home if I need to work remotely. The language is very easy. The packages are very responsive to community demand. It's easy to share my code. The

online support from the community is amazing. But I would say the single best part is that it's an easy to use *open source freeware* program."
[116] "The community knowledge and the breadth of packages"
[117] "A 50/50 mixture of the sheer number of packages for everything & people in the community willing to be helpful and assist the weird problems that sometimes appear."
[118] "The flexibility of the language. The community and the support that that generates on forums."
[119] "Versatility, diversity of packages available, brevity to write complex analysis, the wide documentation and help from the community"
[120] "The community."
[121] "The community, the breadth of available tools, the ease and power of writing your own functions, but above all, the well-developed reproducible research tools"
[122] "The community of users and developers around it."
[123] "There are a lot. Its ability to do things that Excel can't. The reproducibility. The community. The hugeness of everything data related makes it feel like something I will be using for a very long time"
[124] "The community"
[125] "The community."
[126] "The fact that it can be turned into whatever specialty analysis too you need it to be. E.g. there are many people that argue that R is the best Geographical Information System nowadays. In my opinion this is because it allows the community to extend it in all sorts of directions and at the same time still use all the existing (non-domain) aspects. When you combine domain-specific behavior with non-domain-specific analysis tools/algorithms you can only win!"
[127] "The community and easy to use"
[128] "Powerful and easyish. The community."
[129] "I guess the community effort to always be teaching each other, documentation for me is the greatest strength of R."
[130] "The community"
[131] "The community an the availability of packages."
[132] "The community."
[133] "The community."
[134] "the community and how helpful people can be sometimes. There are so many packages it seems that every needs are covered, it's just a matter of finding the best package / approach for each task and some tutorial."
[135] "Packages and the community"
[136] "Breadth of the community"
[137] "the flexibility given by the packages and the community support"
[138] "I like the options that R offers, there is different packages for the same propose and the community always willing to help"
[139] "The community help, any question will be answered by someone if posted online. Also anything that I may need to do is covered by a package of either CRAN or Bioconductor or github."
[140] "I think is the community working always in new packages and functions to share and that R can be used to do a complete analysis from get and clean data until shared your conclusions with excellent ways,"
[141] "Once you get the hang of the syntax, it's very straight forward and flexible. I love all the new packages that are coming out in the tidyverse as they make managing and visualizing my data much much easier. I also love talking about R with other people and because the community is so large, it's so easy to find others that use R for lots of different reasons."
[142] "The community that wants to make working with R as easy as I feel it is becoming. And that moment where you find the stray comma after staring at your screen for hours."
[143] "To have everything you need for statistical things, but also web integration, etc., from the community"
[144] "The community, all the smart people that lean on each other and make it possible to anything with R"
[145] "Ease of use, the RStudio IDE, the community in forums, the manuals."
[146] "The community is large and there are a lot of resources online."
[147] "The community is amazing. They have created some extremely powerful and useful tools that make using and extending R infinitely easier than other environments."
[148] "The RStudio IDE is top-notch, and the community is fun, very open-minded."
[149] "The community and rapid pace of growth. R people = good people (a stolen but true sentiment). I keep saying I've got to move to Python to close some gap or other, but then I discover a package that does what I need in R. So far the 'wave' of development has stayed in front of me. A year ago I would have said I was at the edge, but the progress of the community seems to have accelerated."

[150] "The fact that the community is so large, which helps in two ways: 1) lots of libraries available to solve many different problems, and 2) active forums (especially Stackoverflow) where to obtain help when running into issues with R"
[151] "The community!! And the number of resources I can find on Twitter"
[152] "The community!!!!!! and Hadley."
[153] "The community."
[154] "The community."
[155] "The community!"
[156] "The community and upward trajectory of packages and overall popularity. "
[157] "The community"
[158] "The community."
[159] "The community support. Free and open-source nature. Helps with statistical analysis without investing in commercial software. Other language interfaces. Wide variety of packages. Visualization options."
[160] " Solutions proposed by the community"
[161] "Extensive tools and resources are available from the community. The package system makes it very responsive to new analyses and reproducibility."
[162] "The community of support never ceases to amaze me. There is a constant state of excitement as well as encouragement."
[163] "That you get so much support from Users all over the word - the community is unbelievable. Also the developers make a great job!"

Flexibility

[1] "The community, the availability of tools, the flexibility of opensource."
[2] "R's flexibility and the community of users."
[3] "Flexibility and the ability to go from start to finish for an entire project without leaving the R environment"
[4] "Its flexibility and tooling for ease of data manipulation"
[5] "Its flexibility--and its focus on posing and answering analytic questions about data."
[6] "Flexibility"
[7] "If I can use a single word: capabilities I can use two words: capabilities + flexibility Three words: capabilities + flexibility + community"
[8] "Flexibility and capability for end-to-end use for analysis and reporting."
[9] "Flexibility. Variety of packages. Easy to use high-level implementations."
[10] "Flexibility"
[11] "You can usually get exactly what you want and have to flexibility to develop and apply new ideas"
[12] "the community, the rich functionality, the flexibility"
[13] "Flexibility and breadth of capabilities."
[14] "The power and flexibility in how you do things."
[15] "Its flexibility: it tackles data manipulation/wrangling, visualization and statistical analysis. There is no need to jump from one software to another. It even generates the final report via markdown!"
[16] "The community! So helpful and welcoming. Power and flexibility also."
[17] "flexibility"
[18] "flexibility, ease of use of tidyverse!"
[19] "Flexibility"
[20] "* Ability to hold multiple heterogeneous datasets in memory gives great flexibility. Solution for almost anything can be found. Ability to view source empowers users. All the statistical algorithms you probably need. Great output options for results.(sorry, not one)"
[21] "Speed and flexibility of routine analysis."
[22] "Flexibility and robustness"
[23] "Flexibility, free (as in beer) removes a lot of barriers when moving between organizations."
[24] "flexibility, large community with lots of packages and solutions."
[25] "Flexibility"
[26] "Flexibility and breadth of packages/statistical techniques and approaches (including relatively new US Geological survey packages)."
[27] "Its flexibility and the availability of unique tools like shiny, ggplot2 and most of the tidyverse)"
[28] "The flexibility, and that it's easy to concisely express thoughts in the language."
[29] "Flexibility is a huge plus - but can also be a huge pain - especially for beginners."

[30] "Flexibility, power - the access to other peoples packages. Inspiring when you discover new things you can do."
[31] "The flexibility"
[32] "The flexibility"
[33] "Flexibility."
[34] "flexibility, availability of functions/packages, free!!"
[35] "Community, flexibility"
[36] "Flexibility. Despite its issues R still lives up to its rapid prototyping label. To be truly competitive in the future, there may need to be a breaking change to R."
[37] "Flexibility"
[38] "Flexibility, expressiveness."
[39] "Flexibility, working simultaneously with more datasets."
[40] "Flexibility and availability of packages."
[41] "Ease of manipulating data and the readability of the resulting code with dplyr, and ggplot's flexibility for plotting whatever I need."
[42] "Flexibility. I mostly do inference and data visualization in R. But I recently also took up agent-based modeling. And R has been shockingly effective for doing so."
[43] "Flexibility and ability to easily troubleshoot with online resources"
[44] "The flexibility/customization"
[45] "Flexibility."
[46] "Flexibility, reproducibility and the strong community behind R."
[47] "Breadth, flexibility and customization"
[48] "RStudio and the tidyverse has made it much easier to use R. The best aspect of R is its flexibility. It is very useful to be able to work with it interactively and as a more formal programming language."
[49] "Flexibility, its easy to generalize a specific code to a function."
[50] "Flexibility. Documentation/help on the web."
[51] "It's flexibility and new integration with markdown and document creation. It's a one stop shop"
[52] "Flexibility of the language and the broad/diverse community of contributors"
[53] "flexibility to analyze data, create vis, fit technically rigorous and innovative statistical algorithms, and share results via Rmarkdown, shiny, etc."
[54] "Flexibility"
[55] "Flexibility, community, packaging system."
[56] "Flexibility (many ways to do a task)"
[57] "Flexibility and there is always new clever stuff to learn"
[58] "Flexibility, easy of use, speed"
[59] "Flexibility (packages, techniques etc.), ability to run on a server (with RStudio server) and the community: any question you have is mostly a google away."
[60] "The flexibility of R and it's packages, but I'd also have to say the Stackoverflow support community."
[61] "It's flexibility, I do not need to learn a new language for data reading, cleaning, analysis and graphics, but can do each of these in R. This is in large part to the tidyverse combined with ggplot2"
[62] "Flexibility"
[63] "It's Flexibility."
[64] "Community, flexibility of the language & data pre-processing functionality and nice visualizations (except almost non-existing dynamic visualizations)."
[65] "Flexibility of R as it is programming language, and availability of packages, as it were a proprietary tool for statistics / ML"
[66] "Easy of use when installing packages and the flexibility of the language, leading to tools like dplyr."
[67] "The flexibility."
[68] "Flexibility, ease of getting under the hood, various resources for support"
[69] "Community and flexibility"
[70] "Flexibility."
[71] "The flexibility to handle almost any type of analysis is one of the best aspects, plus the ability to be able to write code at fairly high abstract levels."
[72] "The flexibility, the customization, its free, the robustness of packages. There is an R package to do just about anything you can think of for data science."
[73] "Flexibility and great scope of statistical methods"
[74] "Flexibility"
[75] "It's flexibility and repeatability"
[76] "flexibility and community"

[77] "Flexibility, ease of prototyping, expressiveness."
[78] "- Flexibility- Power- Adaptability: My code in R is completely different after the emergence of tidyverse. It tells a lot that such changes have been carried out from within R without any central planning. That is just not possible with commercial software"
[79] "Flexibility without having to worry much about the numerical analysis aspects of algorithms."
[80] "It has several best aspects (in arbitrary order): openness (as in open source), community, generosity, flexibility and wide scope."
[81] "Incredible flexibility."
[82] "The tremendous flexibility"
[83] "flexibility"
[84] "Flexibility, variety of packages, open source, no black box"
[85] "Endless possibilities, speed, flexibility, contributed packages"
[86] "Flexibility to do almost anything. Ability to process large volumes of data & share code"
[87] "The user community and the fact that it is open source. User community (via stackexchange) helps me find tips on how to solve code problems. Open source means that it is up to date and gives me access to the most advanced methods. Exposure to the cutting edge is pretty great. Also, having worked with other statistical platforms like STATA and SAS, R gives me the most flexibility. While there was a time cost associated with training, I find I get great return on my investment."
[88] "Flexibility."
[89] "Flexibility to achieve what I want in a reproducible way."
[90] "speed. and flexibility. can do amazing analysis quickly. and reproducibly."
[91] "Overall flexibility and breath of packages"
[92] "Flexibility"
[93] "The flexibility of the program and the many statistical functions."
[94] "Flexibility."
[95] "the flexibility in the language and the open source platform, which allows the R community and authors of packages to push new versions on a regular/frequent basis. It's open source!"
[96] "Flexibility and community"
[97] "The flexibility and ability to do things in few lines of code."
[98] "flexibility, functional programming style, options, vast scope of libraries, open source culture, support and community"
[99] "flexibility"
[100] "Flexibility"
[101] "flexibility"
[102] "The maximum flexibility one has and the power of the R and the tidy related packages to start with."
[103] "Flexibility"
[104] "Flexibility via packages"
[105] "Its flexibility."
[106] "flexibility, a lot of functions already written by other people"
[107] "Its flexibility and ease of finding useful examples online"
[108] "Flexibility"
[109] "The flexibility of the language. The community and the support that that generates on forums."
[110] "Flexibility and freedom."
[111] "Flexibility and easily reproducible."
[112] "Flexibility."
[113] "Its flexibility."
[114] "Wide range flexibility"
[115] "1. Access to R documentations and helps 2. It lets to create as many objects at a time3. Data visualizations4. Flexibility of writing commands 5. Of course, it is free."
[116] "Flexibility: I can make it do whatever I want."
[117] "Flexibility. I can code a machine learning algorithm for half of the day and work on a completely different visualization for another project using very similar constructs and packages."
[118] "I like the flexibility it provides - for maximum customization"
[119] "The flexibility of analysis."
[120] "Flexibility and availability of many packages that solve many different problems"
[121] "flexibility and power"
[122] "flexibility"
[123] "FlexibilityNo costs (comp to matlab)"
[124] "Flexibility, community, and indexing."

[125] "Its flexibility and breadth of applications."
[126] "The REPL aspect, the ability to use the language at different skill levels, the extreme flexibility"
[127] "Once the user masters its way of coding, the flexibility of code"
[128] "Flexibility, Easy of use, Easy to teach and work with team"
[129] "Flexibility and great data structure, especially for high-dimensional data"
[130] "Flexibility combined with data centricity."
[131] "Flexibility and community"
[132] "Performance & Flexibility"
[133] "Flexibility. Variety. Versatility."
[134] "Flexibility"
[135] "The flexibility."
[136] "Flexibility."
[137] "Flexibility. Whatever I want to do with data, R can do it."
[138] "the flexibility given by the packages and the community support"
[139] "Flexibility and ease of using it"
[140] "the flexibility"
[141] "Flexibility"
[142] "Flexibility"
[143] "Flexibility of R language"
[144] "Flexibility and ease of use."
[145] "The flexibility has allowed brilliant data scientists to evolve tailored grammars which closely fit the data analysis process. As such, the abstraction of using a statistical language generates minimum frictions between thinking and doing. In addition, R is a little specialized, and doesn't compete with general-purpose languages. Many programmers hate R. As such, I'm not worried about my job being moved to India."
[146] "I can't identify a single best aspect. There are multiple aspects of R that I greatly appreciate. I guess my top two would be: 1) Flexibility: I can use CRAN/github packages or write my own code2) I really love rmarkdown and the ability to generate reports easily. When I need to discuss specific issues with analyses or data cleaning/management with my team, this makes it super-easy to pull in examples of those issues, maybe add some illustrative plots, and kick it off for their review. Doing something similar in SAS (our team's other primary tool) would require exporting results to multiple documents and hand-editing."
[147] "Pack rat and sending my markup to other people. Flexibility and power to deal with relatively large data sets"
[148] "Flexibility."
[149] "Flexibility to do very custom applications and analyses. Constantly evolving, as long as you can keep up with the new packages, functionality, etc."
[150] "Stability, flexibility and the breadth of tools that it has."
[151] "Flexibility and immediate visuals."
[152] "Its flexibility, data manipulation and visualization capabilities."
[153] "Flexibility and user community."
[154] "Flexibility, if something does not exist, you can create it as it is a real programming language (other than SAS!). It's meant to be used for data analysis and some excellent packages to that end exist."
[155] "It's a powerful tool for sophisticated analysis of course in terms of the baked-in techniques, but especially when wedded to its flexibility as a general purpose language."
[156] "Flexibility, that I can write my own functions for the things I want to do, and that anybody in the world with access to the internet can use it. I also like the sharing spirit of the people who use R , reflected in the fact that they share their packages and their knowledge, when I want to know the answer to a problem while writing code, I just google my question , and most of the time somebody has written something about it. I love R."
[157] "Its flexibility of analyzing the data"
[158] "Its comprehensive nature and flexibility for building an analytical system that I can, eventually and hopefully, present to users to run and manage."
[159] "Flexibility, ease of implementation. From idea to rough prototype in very little time. "
[160] "Open source. Flexibility. Number of packages"
[161] "flexibility"
[162] "Open access, transparency of algorithms, flexibility, a platform for research and Teaching not only statistics but data analysis in any area of knowledge "
[163] "Flexibility. You can do everything you want, and implement quickly in a production environment"
[164] "Flexibility to do what I want how I want it."
[165] "Interactivity and flexibility of the language and primarily that I know it very well so don't need to think hard too accomplish most tasks."

[166] "Flexibility. A lot of difficult programming is done for you, which allows you to just explore the data freely."
[167] "Flexibility"
[168] "Its flexibility transform data as much as I need it"
[169] "The flexibility in handling various file formats and ability to interrogate data easily"
[170] "Flexibility."
[171] "It's free, there is support from the R community, flexibility "
[172] "Flexibility, powerful. Stability"
[173] "The flexibility of how it can be used"
[174] "Flexibility and availability of multiple packages"
[175] "Flexibility of getting the code to do what I want. "
[176] "Almost everything may be done. Flexibility. Difficult things may be put into functions that then are used repeatedly."
[177] "Flexibility. I am a better statistician and scientist because of R: more order in my analysis, reproducibility and documentation, have learnt about new methods which otherwise I would never have the time to find and read in a journal, etc."
[178] "flexibility and cost"
[179] "Flexibility, breadth of CRAN, tutorial material available online."
[180] "Flexibility. I can write functions to perform specialized analysis, create production-quality figures, and even entire reports. The fact that R is free is also a huge benefit."
[181] "Flexibility, if you are a statistician."
[182] "Flexibility and power of the language"
[183] "flexibility"

ggplot

[1] "ggplot2"
[2] "magrittr, data.table, and ggplot2"
[3] "Ease of use. tidyverse. ggplot. interactive viz. packages by ropensci team."
[4] "ggplot"
[5] "ggplot2, lme4, dplyr and stan are all standout implementations that are written by the best of the best. It's just so great to be able to use that. RStudio and Rmarkdown are also awesome."
[6] "Its flexibility and the availability of unique tools like shiny, ggplot2 and most of the tidyverse)"
[7] "With Dplyr and GGplot it's a very efficient analysis platform."
[8] "Producing nice figures with ggplot2"
[9] "Existence of Rstudio, shiny, tidyverse, ggplot. They are amazing tools."
[10] "Statistical algorithms and pretty graphs with ggplot2"
[11] "Making beautiful visualizations with ggplot"
[12] "Ease of manipulating data and the readability of the resulting code with dplyr, and ggplot's flexibility for plotting whatever I need."
[13] "Feeling of having data at my fingertips, just with a read_csv, dplyr and ggplot2"
[14] "Ggplot, dplyr"
[15] "Working with tools such as ggplot2 and dplyr to explore data."
[16] "data manipulation packages: dplyr, tidyr, ggplot2, etc."
[17] "It's flexibility, I do not need to learn a new language for data reading, cleaning, analysis and graphics, but can do each of these in R. This is in large part to the tidyverse combined with ggplot2"
[18] "RStudio , httr, ggplot2"
[19] "ggplot2"
[20] "dplyr, ggplot"
[21] "Vectorized languageBioconductorggplot2"
[22] "Simple commands with good defaults produce high quality results (e.g. graphs from ggplot2)"
[23] "Creating functions to perform customized operations on data and then clearly represent results with ggplot."
[24] "Ggplot2, stats functions"
[25] "Best visualization package (ggplot2), best data wrangling packages (tidyverse), good speed (so far as I can tell), great community."
[26] "R is a complete language with excellent packages for data manipulation (dplyr, tidyr,stringr) , visualization (ggplot2) with an active community."

[27] "Once again, I guess that it is the lingua franca in the Statistical Analysis community. It is very easy and fast to prototype something: going from raw data to initial plots to first model is quick (thanks to dplyr, tidyr and ggplot2. And maybe caret too!)"
[28] "ggplot2"
[29] "Variety of packages: ggplot2, devtools, testthat The apply family of functions"
[30] "i like Data Visualization with ggplot2"
[31] "Pretty plots with ggplot2, human-readable code with dplyr, having a reproducible record of my work, quick reports with Rmarkdown/knitr"
[32] "Ggplot2, la comunidad y lo especializado que es."
[33] "?????????ggplot????rmarkdown?shiny??????????"
[34] "That user-contributed packages like dplyr, tidyr, and ggplot completely obviate any need to use base R for data analysis, because base R is terrible. These packages don't cover more general use cases; they can't completely replace the R syntax. But if I just use R for data analysis and visualization, it's pretty great!"
[35] "statistical tools, dplyr, ggplot2, tidyr, ..."
[36] "Fast analysis and quick plots with data.table and ggplot. More gene rally the speed and modern statistical technique's"

tidyverse

[1] "Tidyverse and the community"
[2] "Everyone else at my company has a chance to see the tools that I use. I also love the workflow provided by R and the tidyverse."
[3] "tidyverse"
[4] "Community and tidyverse"
[5] "The tidyverse is one of the best aspects of working with R."
[6] "The tidyverse to be honest. It's some excellent sugar/syntax above the base language of R. I very rarely dive into writing 'vanilla' R code."
[7] "I feel like I can write code as I think; tidyverse helps making data science more human."
[8] "Ease of use. tidyverse. ggplot. Interactive viz. packages by ropensci team."
[9] "Tidyverse."
[10] "Tidyverse"
[11] "The power, with the Tidyverse, of loading, analyzing and visualizing data very quickly."
[12] "Thanks to the contributions of community (data scientists and statisticians) you have everything you need to import modeling and reporting with great quality and professionalism. Today you can explore the data, model it and finally create high quality applications (tidyverse, dplyr> = 0.7, shinny, rmarkdown, flexdasboard, etc.). And if you need a company to scale your work rstudio can meet expectations"
[13] "Flexibility, ease of use of tidyverse!"
[14] "The package ecosystem - especially the new generation of systematic infrastructure packages - e.g. tidyverse"
[15] "Helpful R community and there tidyverse makes everything very easy to understand"
[16] "Where do I begin? - R-Studio, Tidyverse, knitr & Rmarkdown. Integrations with Bitbucket"
[17] "Tidyverse"
[18] "Most problems I can think of have been encountered and solved elsewhere. It is regularly updated, many things I want to do have been implemented already and are simple to adapt to my circumstances. The tidyverse in particular has made using R 1000x better for me."
[19] "Of course it enables you to solve analytical problems but it also helps your thinking process because you have to think about your data especially when using the tidyverse approach/ packages in your workflow"
[20] "Its flexibility and the availability of unique tools like shiny, ggplot2 and most of the tidyverse)"
[21] "It is two-folds: quick implementations of basic statistical analyses and graphics, and the easiness of writing your own functions/packages to customize your own, fancier analyses and implement new methods. Also, the tidyverse approach is really what has been setting R apart from other statistical software in recent years."
[22] "the tidyverse"
[23] "Tidyverse is unmatched in terms of usefulness and vision"
[24] "'Flow'. I feel like I spend more time writing 'analysis' code than 'boilerplate' code. tidyverse is big part of this."
[25] "Tidyverse"
[26] "I feel totally comfortable in R. I know regardless of the task or challenge, given enough time I can figure out a solution in R. Plus, it's so easy to write - especially with the increased adoption of the tidyverse."

[27] "Conciseness and expressiveness tidyverse for data manipulation. Nothing I've worked with equals it. Downside: when there are errors in code, debugging can be a bear. I've found myself in a call stack where dplyr code appears to be doing some runtime expression evaluation, and struggling to figure out what I did wrong."
[28] "Ongoing improvements including tidyverse"
[29] "Tidyverse"
[30] "easy of use, especially with tidyverse"
[31] "Abundance of packages, strong community, ease of use of the language with tidyverse."
[32] "Flexible. Great learning environment- eg I am always finding new applications via R-blogs or other sources (Twitter #rstats or package author feeds) that I can apply to my research. RStudio. The development of RStudio and the tidyverse has made the use of R reachable for me and others that are not experienced programmers. The main advance is the consistency of the syntax (i.e., data argument first)."
[33] "Existence of RStudio, shiny, tidyverse, ggplot. They are amazing tools."
[34] "the tidyverse grammar provides a paradigm which greatly limited my degrees of freedom. Now that my data is tidy, I don't have to think about how to implement things, it's always grouping, mutating, filtering, and summarizing. It helped me simplify the workflow of working with a new data set. First, make it tidy, and then do the analysis. So much time saved."
[35] "The tidyverse/RStudio community and their commitment to consistency and usability"
[36] "It has amazing packages (e.g., the tidyverse) that make data analysis not only easy but also so much fun to use!"
[37] "Tidyverse"
[38] "Exploratory data analysis using tidyverse ecosystem"
[39] "RStudio, the tidyverse"
[40] "tidyverse, hands down. Also RStudio and rmarkdown and the twitter #rstats community"
[41] "Ease of use. Dplyr and the tidyverse make it very easy to get quick answers to questions"
[42] "RStudio - the IDE and also their software tools like Shiny and the tidyverse."
[43] "The tidyverse packages."
[44] "tidyverse"
[45] "Tidyverse. I can make pretty graphs from any set of data"
[46] "tidyverse"
[47] "The maturity of tools for statistical analysis and the beautiful API of tidyverse packages"
[48] "Tidyverse, dataframes, apply to abstract away for loops"
[49] "RStudio and the tidyverse has made it much easier to use R. The best aspect of R is it's flexibility. It is very useful to be able to work with it interactively and as a more formal programming language."
[50] "The \"tidyverse\" packages for fundamentals - without them, I'm not sure I'd love R like I do. And for any niche problem, someone has probably asked a question on StackOverflow or released a package on GitHub. There's also always something to learn, unlike Excel where you can plateau. I now have a package on CRAN and have learned about unit tests, version control, etc. - never thought that would be me."
[51] "Tidyverse"
[52] "Tidyverse"
[53] "tidyverse - easy to write readable/testable code"
[54] "The tidyverse! It really gives R an edge over its competitors."
[55] "Speed to produce decent results via RStudio workflows (knitr tidyverse etc)"
[56] "A language that facilitates data analysis above anything else. The rapid pace of change. Graphical capabilities. Tidyverse for data management."
[57] "Tidyverse."
[58] "Versatility, selection of available packages, tidyverse"
[59] "It's flexibility, I do not need to learn a new language for data reading, cleaning, analysis and graphics, but can do each of these in R. This is in large part to the tidyverse combined with ggplot2"
[60] "Ease of thought to code translation, especially with the tidyverse"
[61] "The tidyverse makes data manipulation, analysis, and visualization impossibly quick, intuitive, and easy."
[62] "Analysis and modeling is less tasking with R, especially with the packages that have evolved over the years - DT, Tidyverse, etc"
[63] "The syntax is very natural for interactively exploring data. Whether it's the dollar sign notation on a data.frame, the consistent use of brackets, or helpful packages like those in the tidyverse and others, R makes it pretty easy to quickly manipulate data without a lot of required code. The graphics are especially powerful for scripting, and RStudio is unmatched by IDEs in other languages for data analysis."
[64] "Magrittr, tidyverse and RStudio + Knitr"

[65] "- Flexibility- Power- Adaptability: My code in R is completely different after the emergence of tidyverse. It tells a lot that such changes have been carried out from within R without any central planning. That is just not possible with commercial software"
[66] "Extensions, constant improvements, tidyverse. Community. Reproducibility."
[67] "Community, packages, tidyverse, it's free."
[68] "it is kinda easy. tidyverse is superb, and helps me a lot to structure my thoughts about the data. and there is always a package for everything. i like RStudio a lot. Microsoft support is also fine. Statistics is just a click away."
[69] "tidyverse, large community adoption, and statistical analysis foundation."
[70] "tidyverse and powerful packages like rstan"
[71] "The packages. Data manipulation and exploratory analysis is super easy and fast to do thanks to the tidyverse"
[72] "Tidyverse"
[73] "Its constantly evolving and getting better. Eg tidyverse tools."
[74] "Tidyverse and Rstudio are brilliant. So well written!"
[75] "the community and tidyverse"
[76] "Reproducible research, tidyverse, shiny, rmarkdown, purrr, community"
[77] "With e.g. tidyverse could be very intuitive. Reasonable basic settings in many functions. If you want you have control, but do not have to control everything from start"
[78] "Easy to find solutions for the problems, tidyverse, Rstudio"
[79] "Easily create higher abstractions Interfacing with dbRcpp Tidyverse"
[80] "High number of existing packages: if you want to try an algorithm you don't know, it probably exists in some R package already, and you can evaluate it before spending more time in implementing this in production.The other great aspect are the RStudio and RMarkdown allowing me to gather all parts (code and results) of my analyses in one place. All packages of the tidyverse are also a great help for data manipulation tasks, which takes up most of my time."
[81] "Rstudio is great, Tidyverse is amazing"
[82] "Best visualization package (ggplot2), best data wrangling packages (tidyverse), good speed (so far as I can tell), great community."
[83] "Tidyverse packages."
[84] "Community, tidyverse"
[85] "Reproducible research (Org Mode, text-mode programming) + good data wrangling (tidyverse) + good analysis (Stan and friends; arm; ...) + the grammar of graphics"
[86] "The tidyverse."
[87] "The Tidyverse and RStudio"
[88] "Open source, huge amounts of libraries, and the Tidyverse"
[89] "The tidyverse."
[90] "The conduct of Rhelp was often been angry and unkind. Hadley and others have raised the level of discourse and always modeled civility, welcoming new users, etc. This is the most important thing. Yes the algorithms are lovely; yes RStudio has changed by life; yes the tidyverse is a welcome breath of fresh air, coherence etc. But most importantly the strident R culture of years ago is changing, put-downs and slams in on-line forums are discouraged and increasingly rare, and this is a tremendous boost for students and for us all."
[91] "A robust community willing to offer help and active development. The tidyverse has allowed for R to be more approachable for new users and to standardize certain approaches that make it easier to talk with other useRs"
[92] "The data manipulation tools especially with the tidyverse"
[93] "Pretty good community alignment on ways of working (tidyverse)"
[94] "Tidyverse and RStudio"
[95] "tidyverse makes learning it a lot easier."
[96] "tidyverse"
[97] "The ease of manipulating data with data frames and the Tidyverse."
[98] "The tremendous user community, RStudio, and the tidyverse packages. They have collectively *GREATLY* increased the quality of life for working with R, particularly in the last 5 years or so."
[99] "The tidyverse Ease of common operations"
[100] "Once you get the hang of the syntax, it's very straight forward and flexible. I love all the new packages that are coming out in the tidyverse as they make managing and visualizing my data much easier. I also love talking about R with other people and because the community is so large, it's so easy to find others that use R for lots of different reasons."
[101] "CRAN + RStudio + tidyverse + community."

[102] "The tidyverse suite of packages."
[103] "RStudio and tidyverse. Data IO Easy to try new packages"
[104] "tidyverse"
[105] "The tidyverse"
[106] "R Studio, Tidyverse, Shiny"
[107] "Tidyverse makes it very easy to write analysis routines that are easy to understand, share and adapt"
[108] "Consistency of tidyverse, new packages, RStudio, DataCamp"
[109] "The tidyverse."
[110] "Tidyverse. Installing packages is seamless."
[111] "It is language for doing statistics; it is created for that! Vectorized functions. Graphics. One can accomplish a lot with less typing. Thousand of packages, many brilliant ones: tidyverse, knitr, ..."
[112] "the Tidyverse and the SO/twitter community"

RStudio

[1] "RStudio, both the app and the incredible packages that they are putting out"
[2] "RStudio"
[3] "RStudio interface"
[4] "Makes data science work easier with various packages and Rstudio."
[5] "Functional programming, good community and the best developer team - RStudio"
[6] "The RStudio development capabilities that make easy to document and test code. Packages like testthat, roxygen2 and devtools make my work very easy and my code very robust."
[7] "the community. IDE RStudio"
[8] "Thanks to the contributions of community (data scientists and statisticians) you have everything you need to import modeling and reporting with great quality and professionalism. Today you can explore the data, model it and finally create high quality applications (tidyverse, dplyr> = 0.7, shinny, rmarkdown, flexdasboard, etc.). And if you need a company to scale your work RStudio can meet expectations"
[9] "The RStudio environment is very nice and intuitive for the most part"
[10] "Easy to use thanks RStudio environment and easy to deal with a data project from A to Z (data wrangling, data viz, modeling, Shiny app)"
[11] "ggplot2, lme4, dplyr and stan are all standout implementations that are written by the best of the best. It's just so great to be able to use that. RStudio and Rmarkdown are also awesome."
[12] "The RStudio IDE"
[13] "Variety of options. I can work in vim using tmux to send code to terminal. On a remote instantiation of R!Or I can sit in RStudio and enjoy modern integration with git and profiling tools."
[14] "RStudio, the community, the packages"
[15] "I can smoothly complete a data analysis without any breaks in the chain. Folks at RStudio with the development of the tidy suite of tools and reporting tools like R Markdown and Shiny have really made R the only statistical/programming tool I need."
[16] "Easy to debug. RStudio interface. Cran. Large community."
[17] "RStudio IDE"
[18] "Honestly, probably RStudio. It's an amazing IDE"
[19] "Flexible. Great learning environment- e.g., I am always finding new applications via R-blogs or other sources (Twitter #rstats or package author feeds) that I can apply to my research. RStudio. The development of RStudio and the tidyverse has made the use of R reachable for me and others that are not experienced programmers. The main advance is the consistency of the syntax (ie data argument first)."
[20] "Existence of RStudio, shiny, tidyverse, ggplot. They are amazing tools."
[21] "RStudio"
[22] "The tidyverse/RStudio community and their commitment to consistency and usability"
[23] "RStudio"
[24] "Packages!!!!!!The pain point is downloading source files and loading them in the server environment because most packages have a lot of dependencies and that can be a pain unlike the desktop environment when dependency es are added automatically when using RStudio"
[25] "RStudio, the tidyverse"
[26] "The work of the RStudio's guys"
[27] "tidyverse, hands down. also RStudio and rmarkdown and the twitter #rstats community"
[28] "RStudio - the IDE and also their software tools like Shiny and the tidyverse."
[29] "RStudio IDE. R gets the job done. Packages. Community."
[30] "Open, free, spark support, lots of packages, easy to use with Rstudio Server"

[31] "Love RStudio and Rmarkdown! So easy to merge code, data, analysis, and final document all together. No cut and paste!"
[32] "Rstudio"
[33] "Variety of packages, the community, visualization tools, RStudio"
[34] "RStudio and the tidyverse has made it much easier to use R. The best aspect of R is it's flexibility. It is very useful to be able to work with it interactively and as a more formal programming language."
[35] "The easy to do data analysis with r + RStudio"
[36] "it's a very flexible language, tons of packages, RStudio is pretty good"
[37] "Easy to use, and the RStudio environment is great to work in. Once you learn the nuances, an easy language to work in."
[38] "Rmarkdown allows me to use the RStudio ide as my basic work routine"
[39] "RStudio is a great environment. Exploratory analysis is very easy in R, then with RMarkdown I can make a finished report."
[40] "Speed to produce decent results via RStudio workflows (knitr tidyverse etc)"
[41] "R is a very flexible language.data.frame as a fundamental type is an advantage with respect to python. So many more packages use data.frames in R vs pandas in python just because data frames are fundamental to R.Rstudio is a very polished IDE.There is a large number of packages in CRAN"
[42] "Flexibility (packages, techniques etc.), ability to run on a server (with Rstudio server) and the community: any question you have is mostly a google away."
[43] "Ease of use, the community (both "industry" and academic) and evolving ecosystem of packages as well as RStudio IDE."
[44] "Great results with little code. Easily extendable (even by me)Nice IDEs (RStudio)Get to meet really smart people"
[45] "My experience of R is influenced by the ease of use of using RStudio, which is a very polished interface to R and allows me to use markup to produce quick and sophisticated reports."
[46] "The syntax is very natural for interactively exploring data. Whether it's the dollar sign notation on a data.frame, the consistent use of brackets, or helpful packages like those in the tidyverse and others, R makes it pretty easy to quickly manipulate data without a lot of required code. The graphics are especially powerful for scripting, and RStudio is unmatched by IDEs in other languages for data analysis."
[47] "I like it. It works well for me. Using R through RStudio is absolutely critical in making the language useful (in my opinion)."
[48] "1. Integration with Latex and/or Rmarkdown for reproducible research, 2. the conveniences of RStudio, 3. the quality of visualizations, 4. that it is free, 5. discovery and application of new tools/packages/techniques from user-written packages."
[49] "Magrittr, tidyverse and RStudio + Knitr"
[50] "RStudio , httr, ggplot2"
[51] "Ease of use, many packages available, good datavis capabilities, RStudio"
[52] "The community on stack exchange and the RStudio folks."
[53] "RStudio!"
[54] "The RStudio IDE and great tools from RStudio."
[55] "The new features such as Rmarkdown and Shiny along with RStudio integration with git have made it so much easier to share code/results/reports. It is more fun than it used to be."
[56] "The high quality design: the language makes sense, RStudio looks and feels great, and all the functions I need are available to me."
[57] "It's easy to write scripts and also have a REPL open to get immediate feedback about what the commands are doing. The REPL is well designed for interactive sessions like this, and when integrated into Emacs or RStudio it makes for a productive working environment."
[58] "It is easy to solve most data manipulation and analysis issues. I have recently moved to RStudio to make it easier to use tools like knitR."
[59] "it is kinda easy. tidyverse is superb, and helps me a lot to structure my thoughts about the data. and there is always a package for everything. i like RStudio a lot. Microsoft support is also fine. Statistics is just a click away."
[60] "Excellent breadth of packages; object-oriented oriented programming lends well to statistical analysis; general purpose utility of R means that projects can often be contained entirely within R; fast pace of innovation on packages; excellent GUI (RStudio); well documented."
[61] "Many people have lots of data but they are not doing much with it outside of basic dashboards. With R I can whip up a summary analysis, building up to more detailed analytics. And I can do it all from RStudio. Resulting in a word or PDF document or slides. This amazes people. They are doubly amazed when I can update the analysis with new data super fast."

[62] "RStudio integration"
[63] "Community, Functional, RStudio IDE"
[64] "great packages for different purposes, with RStudio provides a great environment to run whole analytics pipeline from analysis to presentation"
[65] "Its just fun. Its uncomplicated and functional, the workflow is nice, RStudio and rmarkdown are awesome."
[66] "Very integrated statistical programming, the amount of online resources through especially stack overflow, interaction with RStudio, active development."
[67] "RStudio"
[68] "Tidyverse and RStudio are brilliant. So well written!"
[69] "the syntax is so simple. also RStudio is such a great program"
[70] "RStudio gives me a very effective environment in which I can do all my work."
[71] "Easy to find solutions for the problems, tidyverse, RStudio"
[72] "It's actually fun to code in RStudio."
[73] "High number of existing packages: if you want to try an algorithm you don't know, it probably exists in some R package already, and you can evaluate it before spending more time in implementing this in production. The other great aspects are the RStudio and RMarkdown allowing me to gather all parts (code and results) of my analyses in one place. All packages of the tidyverse are also a great help for data manipulation tasks, which takes up most of my time."
[74] "It facilitates learning-by-doing, which is extremely relevant for students. They here of a method, trial it with the example, read through the rest of the help-file (or vignette or web page) and start toying with the settings. R-functions rarely crash (although some do regularly), typically are fast and cover a HUGE range of statistical procedures. I want to switch to Python but haven't found any reason yet. (Maybe with RStudio starting to dominate through the glossy and ridiculous shiny stuff I will at some point be annoyed enough to switch. Currently, I simply use Rgui (on mac).)"
[75] "Rstudio, the great existent packages and the great community."
[76] "It's easy, intuitive, RStudio is very clean and efficient"
[77] "To be honest, it is RStudio"
[78] "Spectrum of statistical tools, machine learning algorithms, visualization tools, and RStudio is just amazing."
[79] "Speed. Construction of reports that would take all day in our enterprise report writing platform can be done in RStudio in under an hour. Execution of complex reports that take 30 minutes in our enterprise solution, but only seconds to run in R. When an analyst like me is studying data in an iterative fashion, speed to get me from one iteration to the next and then to the one after that is critical."
[80] "RStudio is great, Tidyverse is amazing"
[81] "Ease of use, interactivity, good environment to develop in (RStudio, StatET plugin, ...), good visualization routines."
[82] "The Tidyverse and RStudio"
[83] "RStudio"
[84] "The conduct of Rhelp was often been angry and unkind. Hadley and others have raised the level of discourse and always modeled civility, welcoming new users, etc. This is the most important thing. Yes the algorithms are lovely; yes RStudio has changed by life; yes the tidyverse is a welcome breath of fresh air, coherence etc. But most importantly the strident R culture of years ago is changing, put-downs and slams in on-line forums are discouraged and increasingly rare, and this is a tremendous boost for students and for us all."
[85] "RStudio with markdown"
[86] "community. RStudio's products."
[87] "RStudio. The best IDE hands downs."
[88] "The power it gives me in working with data. Compared to my colleagues and clients who mostly try to get by using Excel, I feel like I've got a power saw and they're using a hatchet. I should also mention reusability. I use RStudio and have scripts that provide a record of what I've done, and if a client makes a tweak in a data file, I can usually incorporate the new data in no time."
[89] "Tidyverse and RStudio"
[90] "- Packages for everything- RStudio environment- CRAN and MRAN- R-Bloggers.com- Shiny"
[91] "Great community of people that is willing to teach and share their work. I think RStudio in particular needs to be thanked for this because they do invest so much money and manpower in educating and informing the R community of new and better features in the packages they support."
[92] "Interpretable, not compiled. Easy to try different things in the RStudio interface."
[93] "RStudio"

[94] "RStudio is tremendously useful tool that makes debugging and validating my findings very easy. I can't imagine I would appreciate R as much without being able to use it."
[95] "The tremendous user community, RStudio, and the tidyverse packages. They have collectively *GREATLY* increased the quality of life for working with R, particularly in the last 5 years or so."
[96] "Cran + RStudio + tidyverse + community."
[97] "RStudio, packages, open source"
[98] "Ease of use, the RStudio IDE, the community in forums, the manuals."
[99] "RStudio and tidyverse. Data IO Easy to try new packages"
[100] "The RStudio IDE is top-notch, and the community is fun, very open-minded."
[101] "RStudio. Nice community."
[102] "En realidad es el entorno RStudio lo que hace de un lugar c?modo para analizar datos"
[103] "For me, it is fun, really really fun. You can do anything with R nowadays (from a simple linear model to write a book or create a webpage) and it is challenging (in a good way) everyday. I never get bored when open the RStudio/R console, you are always learning new and amazing things"
[104] "RStudio"
[105] "????rstudio???dplyr???????"
[106] "Consistent documentation, solid IDE (rstudio), good support of reproducible research through rmarkdown."
[107] "La comunidad es lo primero. Creo que la comunidad de R es de las mejores de Open Source que hay. Por otro lado, los paquetes y RStudio son muy completos y consiguen que tengas un buen ritmo de trabajo."
[108] "RStudio"
[109] "Disponibilidad de paquetes, entorno grafico atractivo (RStudio), fcil de usar."
[110] "Que uno construye el mundo que desea y no aquel que una herramienta de alto nivel le permitir?a. Su gran comunidad y su apertura para apoyar a cualquiera miembro ante sus inquietudesSu gran nmero de paquetesLa existencia de RstudioLas nuevas posibilidades de visualizaci?n y de reportes."
[111] "consistency of tidyverse, new packages, RStudio, DataCamp"
[112] "Ease of use, community, number of packages, R-bloggers, RStudio, community, community, community, R is life"
[113] "RStudio and Rmarkdown!!"
[114] "(i) modeling and data visualization capabilities for flexible investigations and nonstandard analyses, (ii) growing ecosystem of support software (RStudio, knitr, etc.) for adopting and institutionalizing Reproducible Research approaches, (iii) cost (I work for a government agency in a field that is perennially underfunded), (iv) large, active community continually developing new tools and making new methods available for (relatively) easy implementation."
[115] "I use git as version control tool, mixture of git, r, rstudio and rmarkdown, is unbelievable ... I cannot forget how my scientific life was harder without these tools "
[116] "RStudio and CRAN"
[117] "Great discussions on stack overflow, Nice graphics, absolutely love notebooks in RStudio. "

Statistical

[1] "Iteration, automation, use of sophisticated statistical tools, replication, and error control."
[2] "Modern statistical methods are either initially or exclusively implemented in it."
[3] "Easy of use. Powerful machine learning and statistical packages"
[4] "Its designed for people with a statistical/data analysis bent: vectorization, indexing from 1, data frames as a first-class citizen, functional programming."
[5] "For all available statistical languages, r offers the widest selection"
[6] "The quality of the statistical analyses which can be conducted. High power analyses can be conducted on large datasets"
[7] "The possibility to use an open source tool to do advanced statistical analysis. And combined with RStudio, one can do reproducible research (with R Markdown)."
[8] "Flexibility in writing as compared with other statistical software. It is superior for statistics over general purpose language given focus on data, data operations and building model on it."
[9] "Complete control over the workflow, from data ingestion to visualization, including data manipulation, wrangling, statistical analysis"
[10] "Very easy to resolve an error by googling. Simple syntax. A lot of statistical techniques are already written by someone else."
[11] "Its flexibility: it tackles data manipulation/wrangling, visualization and statistical analysis. There is no need to jump from one software to another. It even generates the final report via markdown!"

[12] "Access to a lot of statistical packages, especially for statistical test and time series modeling. These are not usually covered in Python."
[13] "The statistical models."
[14] "Ease of use. One R to meet all Statistical requirement"
[15] "All relevant statistical methods are available in R; it's open-source, so easy to customize and contribute."
[16] "The combination of interactivity for rapid exploration or guesswork, with the other tools in the ecosystem dedicated to making robust and reproducible analysis. It goes into version control readily (a major problem with Mathematica notebooks), and supports every statistical question I might have."
[17] "Availability of statistical computing algorithms"
[18] "Statistical programming language"
[19] "CRAN, cutting edge statistical methods, the community."
[20] "It allows for a reusable workflow for statistical analysis."
[21] "Ability to hold multiple heterogeneous datasets in memory gives great flexibility. Solution for almost anything can be found. Ability to view source empowers users. All the statistical algorithms you probably need. Great output options for results.(sorry, not one)"
[22] "The fact that it is a free program that I can use for almost every single statistical need that I might ever have."
[23] "Flexibility and breadth of packages/statistical techniques and approaches (including relatively new US Geological survey packages)."
[24] "It is two-folds: quick implementations of basic statistical analyses and graphics, and the easiness of writing your own functions/packages to customize your own, fancier analyses and implement new methods. Also, the tidyverse approach is really what has been setting R apart from other statistical software in recent years."
[25] "Simple, many statistical methods,"
[26] "I can smoothly complete a data analysis without any breaks in the chain. Folks at RStudio with the development of the tidy suite of tools and reporting tools like R Markdown and Shiny have really made R the only statistical/programming tool I need."
[27] "That the large, supportive community means that packages (and StackOverflow answers) exist to handle most anything you want to do from statistical analysis to just about anything."
[28] "Reproducibility and advanced statistical analysis"
[29] "The statistical analysis."
[30] "Statistical algorithms and pretty graphs with ggplot2"
[31] "The breadth of statistical functions built in."
[32] "You can concentrate on the statistical part because reading data, performing EDA, building statistical models and checking them is much easier and reproducible than in other languages."
[33] "The open source community. R is very transparent relative to most statistical software programs. Also, writing your own functions. So satisfying!"
[34] "Using built in functions to complete statistical work"
[35] "I have yet to run into a data or statistical problem that R can't help me solve; it's extremely versatile!"
[36] "The maturity of tools for statistical analysis and the beautiful API of tidyverse packages"
[37] "High quality for the best possible price. As a university faculty, I'm able to recommend my students an excellent statistical package which can be used for free"
[38] "The breadth of available packages that make most statistical analysis possible"
[39] "You can do really any data related task that will fit into memory. I can always craft a unique solution if one is needed, and yet there are also very high-quality implementations of standard statistical methods as well."
[40] "Flexibility to analyze data, create visualizations, fit technically rigorous and innovative statistical algorithms, and share results via Rmarkdown, shiny, etc."
[41] "Very easy to manipulate data and visualize along with access to a ridiculous number of statistical packages makes it one of the best mathematical computation software."
[42] "Ability to import, clean and manipulate data, do statistical analysis and then visualize all in one place. R "culture" makes you feel you are not alone in learning and exploring, even if I only lurk and am not proficient enough to contribute."
[43] "R is a combination of a programming language and a statistical software package. I think the balance is just about perfect."
[44] "MANY. Data manipulation, statistical analysis combined with nearly all-purpose programming languages, web interfaces (shiny), easy, great and very customizable plots, interactions with PPT/EXCEL, easy multicore support etc."
[45] "A lot of statistical functions."
[46] "It is tied very closely to statistical analyses, and so that makes it relatively easy to use with students."

[47] "Two things jump out at users defecting from SPSS or another \"major\" commercial software package. One, R output can be customized to include only the needed statistics. I have generated 100-page \"canned\" analysis outputs in SPSS when the relevant statistical information could have been contained in 10 pages or less. Two, R graphics are an order of magnitude better than SPSS, SAS, or Stata graphics. Publication-quality plots can be obtained directly within R, without any post-processing with standalone graphics packages."
[48] "Flexibility and great scope of statistical methods"
[49] "It's a statistical programming language whose general programming features don't feel like they're "second-class""
[50] "Open source, free, state of the art statistical analysis. What else would I need? :-) I wish I knew the existence of R when I was on my undergraduate studies back in 2004. Instead we were using the expensive Eviews for our Econometrics courses."
[51] "The language is forgiving (despite having that introducing quirks, which proponents of strict syntax just don't see the value). It has a large, growing, and agile community... asking for help is easy. Where R is best suited (statistical programming and data analysis), there are really no peers with this community strength."
[52] "Vector/ matrix orientation. Very fast prototyping of statistical/ mathematical ideas."
[53] "Well, of course, it is free. I like command line statistical packages. I always find an option for my analysis."
[54] "It's like a second language to me. It is written for statistical use, which is my primary area of work."
[55] "I love how many statistical tools are available. I very rarely have to write a test from first principals"
[56] "I love that R was designed at the fundamental level with statistical data in mind."
[57] "Easy statistical tests, easy data wrangling and easy visualizations"
[58] "Perfect symbiosis statistical tools and a programming language with a simple syntax. All in one friendly environment"
[59] "power of general purpose programming with statistical modeling"
[60] "Speed (over basic stats packages; I'm in the social sciences, so SPSS for example), autonomy (to create one's own scripts) and control over what statistical analyses I plan to entertain. I also believe R offers the most comprehensive set of tools for statistical analysis, so I wouldn't even consider switching to a different tool (since I'm also quite familiar with R already)."
[61] "tidyverse, large community adoption, and statistical analysis foundation."
[62] "Excellent breadth of packages; object-oriented oriented programming lends well to statistical analysis; general purpose utility of R means that projects can often be contained entirely within R; fast pace of innovation on packages; excellent GUI (RStudio); well documented."
[63] "The people involved in the community. Also the ease and convenience of most routine statistical data analysis."
[64] "The user community and the fact that it is open source. User community (via stackexchange) helps me find tips on how to solve code problems. Open source means that it is up to date and gives me access to the most advanced methods. Exposure to the cutting edge is pretty great. Also, having worked with other statistical platforms like STATA and SAS, R gives me the most flexibility. While there was a time cost associated with training, I find I get great return on my investment."
[65] "Ease of use, most run-of-the-mill analyses are included, breadth of statistical and machine learning packages available, beauty of plotting."
[66] "Very integrated statistical programming, the amount of online resources through especially stack overflow, interaction with RStudio, active development."
[67] "The flexibility of the program and the many statistical functions."
[68] "Work everything in one place. Produce statistical results, beautiful outputs"
[69] "Combination of ease to work with data and availability of statistical methods."
[70] "It's a real general purpose programming language, differently from other statistical analysis tools. It's quite easy to write and to read. There are a lot of packages available. It's well documented and there are a lot of guides available on the Internet."
[71] "Results of statistical analysis are objects which can be used for further programming"
[72] "Ease of applying statistical analysis and also the ability to directly include them in presentations"
[73] "It facilitates learning-by-doing, which is extremely relevant for students. They here of a method, trial it with the example, read through the rest of the help-file (or vignette or web page) and start toying with the settings. R-functions rarely crash (although some do regularly), typically are fast and cover a HUGE range of statistical procedures. I want to switch to Python but haven't found any reason yet. (Maybe with RStudio starting to dominate through the glossy and ridiculous shiny stuff I will at some point be annoyed enough to switch. Currently, I simply use Rgui (on mac).)"
[74] "Greatly fit to statistical studies. R doesn't need the complicated concepts like OOP. R gives plentiful statistical graphic tools, functions and testing. It's the most famous statistical program and really works."

[75] "R provides a connection point between more 'theoretical/fundamental' research (e.g. probability,...) and practitioners. It shortcuts the distance (not so many environmental scientists are able to implement statistical theory straight from the printed paper) by making it possible for those practitioners to use and test statistical practices it in their applications."
[76] "A lot of things are already done for statistical problems."
[77] "That I can have a look at how a function works and understand what it is doing not relying on something cryptic from e.g. Stata that takes for ever to read. Often learn statistical procedures by looking at the code producing the results."
[78] "The ability to produce high quality documents showing where data come from, R code, figures, and statistical output in one place."
[79] "Spectrum of statistical tools, machine learning algorithms, visualization tools, and RStudio is just amazing."
[80] "Statistical tools and active developer community."
[81] "Friendly environment for statistical computing and good support for reproducible reports/research."
[82] "The easiness of doing statistical analysis and the visualization APIs."
[83] "R is ideal for statistical computing, but for me the aspect that one can do numerical mathematics and optimization with R is also important."
[84] "There are complex statistical algorithms already implemented"
[85] "How flexible it is and the ability to do almost any kind of statistical analysis I can think of."
[86] "Free, and it suits my work (statistical analysis of fairly big datasets)."
[87] "Great statistical modeling and graphical visualizations."
[88] "Easy to do complex statistical analysis."
[89] "Loading a manipulation data, performing statistical tests."
[90] "Availability of a large number of statistical algorithms. Plotting facilities."
[91] "I know it is capable of doing all the things I want it to, even if I can't figure out how. When I do figure out how, I can clearly see how the data have been analyzed. By contrast, a lot of other software is simply limited in functionality and the statistical procedures are inadequately documented."
[92] "Robust language that does very sophisticated data manipulation, statistical analysis and machine learning in an efficient and intuitive way"
[93] "That it is useful for anything statistical, and it's free."
[94] "It is generally straight forward and has a lot of advanced statistical packages I need."
[95] "Reproducible research, statistical packages, visualization"
[96] "It's like a combination of a general purpose programming language and a statistical software."
[97] "Statistical algorithms and data wrangling"
[98] "The breadth of statistical algorithms available."
[99] "It save time for reproducible pieces of work, it's easy to document if someone else needs to dive into the analysis, it's got tools for everything I do (connecting to a database, statistical analysis, ML algorithms, visualization)"
[100] "The flexibility has allowed brilliant data scientists to evolve tailored grammars which closely fit the data analysis process. As such, the abstraction of using a statistical language generates minimum frictions between thinking and doing. In addition, R is a little specialized, and doesn't compete with general purpose languages. Many programmers hate R. As such, I'm not worried about my job being moved to India."
[101] "the documentation - although I have friends that found it awful and have stayed permanently in STATA - for me it has been always clear what it means across packages, another thing that i like is that the errors are quite 'clear', namely you can tell where you went wrong and how, i have problem understanding the errors of other statistical packages, like stata and spss"
[102] "Compared to e.g. Python, explorative data analysis is very quick to perform. Also the selection of ready-made implementations of statistical algorithms."
[103] "The availability of packages to build complete data processing solutions, i.e. data import from a database, data wrangling in dplyr, statistical and predictive modeling, data visualization, building (interactive) reports, shiny interfaces, etc."
[104] "The statistical based language, the new graph libraries and working with R in the cloud with Jupyter, Zeppelin, etc."
[105] "Once again, I guess that it is the lingua franca in the Statistical Analysis community. It is very easy and fast to prototype something: going from raw data to initial plots to first model is quick (thanks to dplyr, tidyr and ggplot2. And maybe caret too!)"
[106] "To have everything you need for statistical things, but also web integration, etc., from the community"
[107] "I enjoy the functional programming features in R. They mesh really well with my statistical background."
[108] "The statistical analysis options"

[109] "Extensive set of statistical and machine learning algorithms"
[110] "The breadth of modern statistical tools. The quality of visualizations."
[111] "It's made for statistical analysis and trying to be everything to everyone. This allows concepts to match implementations much easily - e.g., a function in a formula is a function in R, as opposed to being an abstract/public interface type function from like a OO language."
[112] "Statistical Accuracy and for quick visualization."
[113] "It's always updated, not just the core, but also the statistical packages."
[114] "On the edge of statistical development. Most if not all, new statistical development are translated in R"
[115] "Statistical analysis tooling comes for free. This sounds small but it's pretty great to drop into an interactive stats shell."
[116] "Statistical packages"
[117] "The amount of work that statisticians and computer scientists have put into it: this shows in the extensive documentation, the thoughtfulness of many data structures, and the online communities with help not only for base R but for many specialized packages that already have answers to many questions. Compared to other languages, R (well, technically, S) being built around statistical analysis really shows in how statistical operations are tremendously easy and integrated with everything. Doing even simple modeling in other languages requires overhead and preparation, let alone complex modeling. The open-source nature of R also helps; more than a few times, I have looked at source code: to debug, make sure a function is doing what I think it is doing and want it to do, or make variants. I don't look at the core implementation, though, and packages and scripts in Matlab are equally accessible, but I am comforted by knowing that there are computer scientists out there who can and do look at the core implementation to check it over and make improvements, something not possible for Matlab. "
[118] "Easy of use for Statistical Computing "
[119] "data table, Hadley's packages, S3, formulas, and the large body of obscure statistical research available as R code"
[120] "Comprehensive set of statistical algorithms and if I need to I can change the existing code."
[121] "The breadth of available statistical packages"
[122] "The best aspect of R is its analytic and statistical packages. "
[123] "The community support. Free and open-source nature. Helps with statistical analysis without investing in commercial software. Other language interfaces. Wide variety of packages. Visualization options."
[124] "statistical tools, dplyr, ggplot2, tidyr, ..."
[125] "Statistical methods available, either via packages or through core functions. Graphics."
[126] "At it's best, R makes difficult statistical analyses easy and seemingly impossible Ines do-able. "
[127] "All of the statistical packages."
[128] "Keeping up with the larger community of people working on statistical packages."
[129] "Fast analysis and quick plots with data.table and ggplot. More gene rally the speed and modern statistical technique's"
[130] "Community support, quality of statistical functions"
[131] "It's a statistical standard"

Question 28: Supporting statements

Cryptic

[1] "Cryptic error messages. Inconsistencies in function naming schemes. Examples in the documentation that are either too simple or too complicated."
[2] "Cryptic error messages"
[3] "Cryptic documentation for multiple important base functions"
[4] "Error messages are incredibly cryptic-- doubly so if using an unfamiliar package. With time, I have learned to understand directly what the 'meaning' of error messages are, but in my experience teaching or showing newcomers R code and trying to help them with it, I have noticed that it is a big hurdle."
[5] "Error messages can be really cryptic."
[6] "Struggling with cryptic documentation that doesn't have good examples to follow."
[7] "Sometimes cryptic, Steep learning curve"
[8] "Pretty steep learning curve, especially if you're using base R. The split-apply-combine approach is absurdly hard when you compare it to something like a PivotTable in Excel. Thank the lord for dplyr. And the error messages can be cryptic for newbies: “objects of type closure are not subsettable” hehe - sure, that's not too bad to an advanced R user, but that is pull-your-hair-out stuff for a beginner."
[9] "As someone who had never coded, at all, I have found that the multiple ways of doing everything was confusing. I spent time learning base plots until someone said, learn ggplot. I have come increasingly to try to use the tidyverse packages for everything. In short, when you learn on your own you don't know how to judge all the possible functions and packages. Also, as everyone says, the error messages are often cryptic: only someone who wouldn't make the mistake can understand them."
[10] "The syntax is sometimes cryptic or inconsistent, and every once a while something produces unexpected results. The flip side of R's flexibility is that there are usually many ways to accomplish something, which can be confusing. Sometimes I teach my students one way to do something, and they find another way to do the same thing on the web, and get confused. I believe that is an unavoidable price to pay for the flexibility."
[11] "Cryptic error messages."
[12] "Sometimes, the packages literature are rather cryptic."
[13] "Sometimes unexpected behavior without a warning which makes debugging more time-consuming than necessary. Default behavior should throw more warnings, e.g., with vector recycling or hist(1:4). In general also error messages are cryptic for beginners"
[14] "Error messages are often cryptic and not helpful"
[15] "Documentation is somewhat cryptic"
[16] "Cryptic error messages cost a lot of nerves"
[17] "Slow speed. Also I consider the Rd documentation system very creaky. It puts developers in a box. It encourages cryptic explanations. It discourages examples by forcing them to be run when checking packages ('not run' is one of the most cryptic elements for users; only developers realize it's a CRAN-relevant annotation.) It discourages holistic user manuals that actually explain how to use a package in a coherent workflow, along with relevant technical details of algorithms. The vignette system helps but is misnamed, under-used and under-visible. Required documentation and vignettes have been great features; they're creaky only due to R's success. Static code analysis by CRAN checking is too invasive and attempts to be too smart. Drop Solaris or explain to the community who still uses it."
[18] "It is not a general programming language: non-statistical tasks are cumbersome. It is difficult to define complex data structures (and S4 classes make things worse).Non-trivial algorithms (requiring loops or recursion)are difficult to write and very slow. Vectorizing computations sometimes leads to cryptic code."
[19] "Steep learning curve. Some cryptic error messages"
[20] "It's hard to understand some really cryptic packages. But I persist."
[21] "The cryptic documentation which I (with a PhD) often can't decode."
[22] "Poor documentation - often need to Google. Not as fast as MATLAB. Errors can be quite cryptic sometimes."
[23] "Sometimes the package documentation is more cryptic than I would like for beginner/intermediates not strictly in the academic or scientific world. Some might call it intellectual laziness but some users need to get the job done. Somewhere between spoon feeding and outright full blown tutorials would be good as a standardization."
[24] "It can be cryptic and difficult to figure out why one is getting an error. "

Error message

[1] "Cryptic error messages. Inconsistencies in function naming schemes. Examples in the documentation that are either too simple or too complicated."
[2] "Error messages (sometimes not as helpful as they should be)"
[3] "R really needs to improve the error messages. Sometimes I really don't know what happened based on the error message."
[4] "Error messages, crashing, etc."
[5] "Error messages and matching issues between packages and versions."
[6] "Cryptic error messages"
[7] "Insanely opaque error messages. And traceback() frequently leads nowhere."
[8] "1. Documentation2. Error messages"
[9] "Error messages by far. Easily the hardest thing to work through, as sometimes it seems impossible to find out what caused it, and how to fix."
[10] "Error messages are incredibly cryptic-- doubly so if using an unfamiliar package. With time, I have learned to understand directly what the 'meaning' of error messages are, but in my experience teaching or showing newcomers R code and trying to help them with it, I have noticed that it is a big hurdle."
[11] "Ambiguous error messages."
[12] "Error messages can be really cryptic."
[13] "Could be more useful error messages and list viewing in RStudio"
[14] "Obscure error messages."
[15] "Trouble shooting can be difficult. Error messages are often obtuse and package interactions can sometimes be difficult to determine."
[16] "Occasionally running into error messages that make no sense"
[17] "Spotting typos when code won't execute (bad workman always blames his tools...) Error messages can be extremely vague"
[18] "Error message is difficult to understand."
[19] "Error messages are perplexing."
[20] "Many error messages are not very informative about the cause of the error & some packages have a really poor and insufficient documentation."
[21] "Pretty steep learning curve, especially if you're using base R. The split-apply-combine approach is absurdly hard when you compare it to something like a PivotTable in Excel. Thank the lord for dplyr. And the error messages can be cryptic for newbies: "objects of type closure are not subsettable" hehe - sure, that's not too bad to an advanced R user, but that is pull-your-hair-out stuff for a beginner."
[22] "Confusing error messages."
[23] "It's a steep learning curve. I feel I still have much to learn. I enjoy expanding my skill set in this manner, but it can be a bit daunting at times. Sometimes it can be difficult to work out why something isn't working because the error messages can sometimes be a bit ambiguous. Googling can help with this, but it's not simple to pick up in the way that Excel is."
[24] "Sometimes the error messages are really a mess."
[25] "I feel the error messages are terrible, I frequently wonder if it could be a collective project to catch errors and display them in relatively human readable form."
[26] "The error messages can sometimes be a bit opaque"
[27] "Error messages in R are frequently useless or even worse, misleading. I know that error documentation is not high on any package developer's to-do list, but discovering the true cause of failure in syntax code is frequently frustrating to the end user. I have said nasty things in this survey about SPSS, SAS, and Stata, but I will say that their error message reporting is superior to R."
[28] "Lots of ways to implement the same task, so it can be difficult to find the simplest way to solve a problem. Error messages can sometimes be tricky to decipher, especially when lists are involved, and especially due to NSE (which itself is a very powerful tool). Some of the defaults in base R require some trial-and-error to figure out."
[29] "The unfriendliness of the documentation and error messages. It is incredibly frustrating having to guess what went wrong when you're given no examples of the commands and the error messages don't point you to your errors."
[30] "Cryptic error messages."
[31] "The error messages. It is also sometimes hard to stop a process when I did something wrong and it takes a long time to compute."
[32] "Not all error messages are necessarily easily interpreted by novices, which gives the language a greater learning curve than is probably necessary."

[33] "Dealing with odd errors or bugs that don't have helpful error messages."
[34] "Sometimes the error messages can be obscure, though this is exacerbated when reviewing code written by intermediate-level users because they know enough to get into trouble but not enough to avoid pitfalls."
[35] "Steep learning curve and error messages can be hard to understand at times. Core R code can be hard to parse (but Hadley's grammar gets around that, I just need to learn it)"
[36] "Unhelpful error messages/warnings (i.e., not detailed enough to understand where the issue is happening)"
[37] "Sometimes the documentation is a bit opaque and error messages not so helpful."
[38] "odd hard to interpret error messages. thank god for stackoverflow!"
[39] "Sometimes unexpected behavior without a warning which makes debugging more time-consuming than necessary. Default behavior should throw more warnings, e.g with vector recycling or hist(1:4). In general also error messages are cryptic for beginners"
[40] "Obtuse error messages, incomprehensible help files, examples in help files that are designed to show how clever the author is rather than how the software works, the culture of bullying and tagging along with bullies."
[41] "Error messages are often cryptic and not helpful"
[42] "bad documentation in base R, horribly non-informative error messages, no trace list in error messages by default"
[43] "Oh dear, quite a bit. The documentation quite often, the lofty and sometimes hostile style on r-help (though this has improved a bit and I can sympathize as so many questions are problematical). However, the real challenge is that the learning curve is, mostly inevitably I think, steep and goes on being steep. That isn't helped by real inconsistencies in style across packages and in the emergence of the tidyverse and magrittr conventions which sometimes seem to be taking R into being something so different from “old R” that they might be a new language/system. There are also some “gotchas”, particularly around factors, but more widely, that can be really irritating and sometimes the get me with warnings or error messages that are way, way less than helpful, sometimes avoidably I think. I think it might need a radical rethink of the warning/error system to make it easier to, and to encourage authors, to write much more helpful error messages. I have a continued beef that R base neglected text output, except for those who used TeX in a way that I think was blatantly about keeping “ordinary” and social sciences from adopting R as early as they could have done and was completely inconsistent with the superb attention to graphic output. A funny “verse” of knitr, RStudio and a number of related packages, is starting to address this and so much so that I have, with some regrets, abandoned my beloved Emacs/ESS use of R for RStudio. However, there is much that is still a mess in table output and easy preparation of “from data to submitted paper” reproducible research."
[44] "obscure error messages"
[45] "Sometimes frustrating - error messages, speed"
[46] "R error that breaks the r Environment while computation without any error message"
[47] "Sometimes I wish the error messages were more descriptive. Can be a little overwhelming when getting started."
[48] "Understanding the error messages."
[49] "cryptic error messages cost a lot of nerves"
[50] "Understanding the error messages that I receive and initial learning of the language."
[51] "Error messages too technical"
[52] "Steep learning curve. Some cryptic error messages"
[53] "Can not understand error messages."
[54] "Somewhat obtuse syntax and unhelpful error messages at times"
[55] "The help and specifically the examples in the help seem very complicated, more so than the task at hand. Some inexplicable error messages on installation of packages ."
[56] "Inconsistency among base R functions and impenetrable error messages."
[57] "Find some of the error messages can lack in detail, making it tricky to troubleshoot"
[58] "Ambiguous error messages."
[59] "Error messages not always very helpful"
[60] "Error messages are often incredibly difficult to understand. Sometimes everything work because R finds solutions to incompatibilities between data formats, but the results are wrong."
[61] "Interacting with some of the base functionality can be frustrating, e.g. less helpful error messages"
[62] "Obscure error messages!"
[63] "The error messages are almost always opaque. Thank god for traceback()."
[64] "VERY poor and unhelpful error messages."
[65] "When I lose connection to the work server, and hence my files, I get error messages every time I type something, even in the RStudio text editor."
[66] "The error messages and built in help"

Learning curve

[1] "The learning curve is very steep"
[2] "Language is a bit odd and can have a steep learning curve compared to more C-like languages"
[3] "Idiots who describe it as “written by nerds”. People who say it has a “steep learning curve” having spent a few minutes looking at it. The lack of corporate management board-level awareness of its capabilities."
[4] "Steep learning curve. Programming language not very rich featured and somehow odd"
[5] "Getting started, steep learning curve"
[6] "Steep learning curve for those who don't know it."
[7] "Steep learning curve. I feel stupid all the time."
[8] "the initial learning curve"
[9] "Steep learning curve (when you have no programming experience), sometimes different package versions produce different results, large data sets are difficult to handle because R reads the data into memory. I know there are workarounds for that and also some nice packages, but I feel that this aspect could be improved."
[10] "Sometimes cryptic. Steep learning curve"
[11] "It's a steep learning curve. R is my go-to data analysis software. I would argue it's the best go-to data analysis software. But convincing colleagues (and students) to use it can be a challenge because it can be so difficult to get up to speed."
[12] "Learning curve re: understanding data structures"
[13] "The initial learning curve is definitely steep."
[14] "steep learning curve for beginners (but worth it!!)"
[15] "The learning curve, combined with terse or non-user-friendly documentation of lots of functions in base R. I've now shepherded a couple of co-workers through learning R, and have observed that there is usually a 3-month period of head-exploding, swearing frustration before the language really starts making sense. It may be inevitable, but I feel like better access to informative examples would help."
[16] "The learning curve can be steep, and there are lots of intricacies and details to learn. I.E... Capitalization matters (this is helped with the right GUI!), do I need a comma when filtering my data.frame rows? Are these dates in the right format? Are these numbers numeric?, Do I need double brackets or single brackets? These are all things that can be remedied by getting used to them, learning them, or with the aid of a good GUI such as RStudio, though."
[17] "Steep learning curve."
[18] "Steeper learning curve than tools that have a more graphical use instead of command based"
[19] "Steep learning curve. Getting colleagues to use R."
[20] "Worrying that the core language isn't developing fast enough and that R will eventually die / that it's losing to Python / that it is really hard for beginners to get into because of inconsistency. Having to leave the tidyverse. knowing that it'll take fellow students at least a year or two to be able to do anything reasonably complex in R because the learning curve is high -- mostly I think this has to do with starting in the wrong place and a lack of cohesion on the whole."
[21] "Steep learning curve for biology undergrads"
[22] "Steep learning curve at beginning"
[23] "Learning curve for newcomers or domain specific researchers"
[24] "Learning curve."
[25] "Steep beginners learning curve"
[26] "Learning curve for beginners."
[27] "steep learning curve, trillions of possibilities..."
[28] "it's error prone; steep learning curve"
[29] "Learning curve"
[30] "Having to get packages security-checked before they can be used on our desktops, and the security/privacy blocks from our corporate policy that means we can't use it in the cloud with our sensitive data. Also wish debugging was easier, and the learning curve easier."
[31] "The learning curve for those not experienced with programming, but there's not much that can be done about that apart from sticking with it in the early (frustrating) stages or attending R workshops."
[32] "Steep learning curve, because there are so many different ways to do the same thing."
[33] "The learning curve. Working with string or otherwise heterogeneous data."
[34] "Learning curve"
[35] "The learning curve was steep but was worthwhile. Sometimes the documentation isn't especially easy to get through, though the help function is useful."
[36] "The steep learning curve. Makes it hard to convince people to switch over from GUI programs. Also some of the community (generally straight white males!) can be very hostile to beginners."

[37] "Learning curve is steep"
[38] "Pretty steep learning curve, especially if you're using base R. The split-apply-combine approach is absurdly hard when you compare it to something like a PivotTable in Excel. Thank the lord for dplyr. And the error messages can be cryptic for newbies: “objects of type closure are not subsettable” hehe - sure, that's not too bad to an advanced R user, but that is pull-your-hair-out stuff for a beginner."
[39] "Steep(!) learning curve, especially for folks who are new to programming (like me)"
[40] "Steep learning curve made worse by poor documentation."
[41] "As a non-programmer, it has had a steep learning curve. I still have trouble sometimes understanding programmer-speak."
[42] "(1) Steep learning curve/hard to do things that seem like they should be easy.(2) Memory issues. Although there are many current methods/tools for dealing with this - in today's day and age (to make R the truly preferred tool), Base R and R packages should be able to better handle significantly larger data sets/computations “out of the box” (that is, you should be able to use the same R code, packages, etc. whether your data is small with thousands of records or larger with millions+ of records). (3) Fragmented ecosystem - many tools to do many things (which makes R so strong) BUT these tools don't always work well together (or tools to do the same thing work very differently) and can sometimes cause resulting complications."
[43] "Learning curve. Runtime"
[44] "Learning curve"
[45] "Learning curve. Maintaining scripts with esoteric packages."
[46] "Steep learning curve which is exacerbated by lack of rules on style and syntax. R may be too flexible."
[47] "It's a steep learning curve. I feel I still have much to learn. I enjoy expanding my skill set in this manner, but it can be a bit daunting at times. Sometimes it can be difficult to work out why something isn't working because the error messages can sometimes be a bit ambiguous. Googling can help with this, but it's not simple to pick up in the way that Excel is."
[48] "Steep learning curve, inconsistencies in language."
[49] "Learning curve is steep initially"
[50] "the hard learning curve at the beginning, getting data into R, learning the different kinds of variables"
[51] "Steep Learning curve."
[52] "The learning curve of having to use new packages or new features."
[53] "Learning curve for CS newbies, cannot handle large datasets (biggest limitation Vs Python)"
[54] "it was the learning curve"
[55] "High learning curve."
[56] "Coming from the spreadsheet world it can be difficult at times to hold in my head how the data is being manipulated. I find myself using str(), head(), and view() pretty frequently to make sure I know what happened to the columns. That and translating tasks (like splitting strings) from excel functions to R functions, but both of these are just part of my learning curve and not structural to the language or environment."
[57] "Steep learning curve."
[58] "Learning curve."
[59] "learning curve, quirks"
[60] "Trying to onboard new hires who have never used R ... the steep learning curve."
[61] "My biggest problem is usually that there are so many packages that its difficult to remember which function I want to use and the particular quirks/limitations of each command. In addition, R sometimes evolves so quickly -- especially with developments in the “Tidyverse” -- that it’s hard to keep up. While the “Tidyverse” contributions to R are excellent, and I use them constantly, and have enormously improved the usability of R, the fact that they evolve fairly quickly means that I feel like I'm constantly starting over on the R coding learning curve. That is, learning R while the most useful aspects of R are constantly changing is frustratingly Sisyphean. I don't think there's much that can be done about this; R is in a transition phase. But I can't wait until things mature and settle down into greater continuity and equilibrium."
[62] "Learning curve can be steep and frustrating at times. Some of the R help documentation takes too much for granted from the typical user."
[63] "The learning curve was pretty high. It took me being a TA in a structural equation-modeling course for me to finally feel comfortable enough in R, but I still have so much to learn. Data cleaning is a daunting task in R right now, so I usually do that in Excel and/or SPSS because that's very easy for me."
[64] "Not all error messages are necessarily easily interpreted by novices, which gives the language a greater learning curve than is probably necessary."
[65] "I can't think of anything offhand. If forced to come up with something, I think the worst part would be just starting off, I think that the hugeness of R and the learning curve may be intimidating for some."
[66] "The learning curve."
[67] "The learning curve to get other people to use it"

[68] "The not-so-consistent syntax, which may get worse as packages are updated (e.g., ggplot). It's annoying when a slightly old piece of code won't run because a tiny little detail has been updated. I'm ok with the learning curve, and I'm even ok with the clunky syntax (relative to Python, for example), but the inconsistency of the clunkiness is a pain sometimes."
[69] "syntax has a huge learning curve"
[70] "steep learning curve"
[71] "Steep learning curve and error messages can be hard to understand at times. Core R code can be hard to parse (but Hadley's grammar gets around that, I just need to learn it)"
[72] "Steep learning curve to write code. Must know exactly what the errors mean to fix them, there's not enough guidance on how to improve your code"
[73] "Learning curve for newcomers? Most languages have these, but particularly the learning curve for statistics students who just want to do some EDA/basic modeling."
[74] "The learning curve. Errors that are hard to interpret."
[75] "I had a steep learning curve. Part of the problem was being led in "wrong directions", at least for me. I wrote a R Quick Start document for a presentation that suggests using dplyr, ggplot, etc and not xts (unless you know why you need it)."
[76] "Excluding using the Tidy family, R has a huge learning curve."
[77] "The initial learning curve"
[78] "The learning curve for just starting out, even if you just want to infile data and manipulate data and output data. this has gotten better in the last several years. Documentation of the language and packages could be written better for the layperson, especially with better examples. Instead of just having snippets of code to copy and past i would rather see the snippets of code and the output that statement output. Some package vignettes are better than others, but in general could be improves"
[79] "Hard learning curve for functional programming. Difficulty is transferring OO patterns to R."
[80] "Learning curve"
[81] "The learning curve for new packages or new techniques can be a bit steep but it's not especially bad. Sometimes also my work computer doesn't like R installing packages or updates without admin rights, so I have to do my analysis without any packages other than the base installation (or packages installed in a temp directory)"
[82] "Steepness of the learning curve"
[83] "Learning curve"
[84] "Oh dear, quite a bit. The documentation quite often, the lofty and sometimes hostile style on r-help (though this has improved a bit and I can sympathize as so many questions are problematical). However, the real challenge is that the learning curve is, mostly inevitably I think, steep and goes on being steep. That isn't helped by real inconsistencies in style across packages and in the emergence of the tidyverse and magrittr conventions which sometimes seem to be taking R into being something so different from "old R" that they might be a new language/system. There are also some "gotchas", particularly around factors, but more widely, that can be really irritating and sometimes the get me with warnings or error messages that are way, way less than helpful, sometimes avoidably I think. I think it might need a radical rethink of the warning/error system to make it easier to, and to encourage authors, to write much more helpful error messages. I have a continued beef that R base neglected text output, except for those who used TeX in a way that I think was blatantly about keeping "ordinary" and social sciences from adopting R as early as they could have done and was completely inconsistent with the superb attention to graphic output. A funny "verse" of knitr, RStudio and a number of related packages, is starting to address this and so much so that I have, with some regrets, abandoned my beloved Emacs/ESS use of R for RStudio. However, there is much that is still a mess in table output and easy preparation of "from data to submitted paper" reproducible research."
[85] "The learning curve is steep. Inconsistency in syntax, requirements and performance between packages."
[86] "Steep learning curve, memorize functions names..."
[87] "The learning curve"
[88] "Having to invest time to learn and get work done. Getting through the learning curve."
[89] "The steep learning curve of having new packages come out all of the time. You really have to stay on top of R and force yourself to code every day to be able to maintain vs. something like Visual Basic which largely stays the same and is definitely more standard."
[90] "There is a steep code learning curve. I've been using R for about 4 years and I feel like I've just scratched the surface. The package documents are not always super clear and I find my self Googling a lot."
[91] "Steep learning curve. To friends just getting started with data science / machine learning / programming, I'm much more likely to recommend python than R, even though I use R much more frequently than python"
[92] "R has a steep learning curve. CRAN needs to have its packages be segmented by scope, by adding up to 3 category labels in the package parameters so that one can browse for packages by topic. There is a bit of a birds

nest of packages. class based architecture needs improving, but I see that work is being done about this. Need support for scripted workflows, i.e. scripts that get the job done, illustrating the use of packages in CRAN or bio conductor. Need better management of example data, example data is not easy to explore/find."
 [93] "Learning curve"
 [94] "The learning curve can be high at the beginning"
 [95] "R has a VERY steep learning curve."
 [96] "Learning curve is steep"
 [97] "There's a steep learning curve, and the syntax feels unforgiving to beginners. In the days pre-tidyverse (year 10 ante Hadley) it took me three dedicated attempts to get warm with R."
 [98] "Steep learning curve. Some cryptic error messages"
 [99] "It can be a steep learning curve. An irregular user like me needs to refer back to a cheat sheet very often."
[100] "There is a steep learning curve. Well worth it though."
[101] "Learning curve for non-programmers..."
[102] "Steep learning curve"
[103] "Steep learning curve and insufficient vignettes and accessible examples. I would have never been able to use R to do anything without Stack Overflow and Google."
[104] "The learning curve is steep for most beginners."
[105] "Learning curve and the variety of programming styles."
[106] "Sometimes there are speed issues and the learning curve is pretty steep"
[107] "Steep learning curve"
[108] "The slope of the learning curve is large. Getting to understand different styles such as dplyr and data.table apart from the base R may get you some time."
[109] "Learning curve, having to go through with a fine tooth and comb when ether are errors in code, poor visualizations"
[110] "The sheer number of functions makes the learning curve fairly steep. The class system is somewhat obscure."
[111] "Learning curve"
[112] "I started working with R several years ago when there had not been much development of simplifying packages. The learning curve was really awful. I didn't have time to learn the language properly and only used it when I needed it for a given task, and (of course) had a lot of frustration. It's still not exactly simple; the variable scoping can be confusing when developing my own functions. And sometimes packages throw errors for no discernable reason. Also, the package documentation needs to have more and better examples, and better descriptive content. They seem to often take a bare-bones approach that is not sufficient for users who don't live and breathe R. Thank god for blogs and StackOverflow, honestly. Without those resources, I'd never have been able to become a more consistent user. But it shouldn't be necessary to crowd-source documentation development."
[113] "Learning curve"
[114] "There are many learning curves. Can take all day to do something simple. But once you get that working, it is typically resolved. Also - updates can affect previous code/packages and if packages end up being 'discontinued'."
[115] "Steep learning curve. Lack of coding standards - every programmer has a different style, so no consistency in approach to problem solving. Can be hard to read another's code."
[116] "Steep learning curve."
[117] "The learning curve"
[118] "Learning curve"
[119] "Relatively steep learning curve."
[120] "Learning curve and changing how you think about analysis. The transition from a point and click interface was difficult and I imagine that is one of the major issues for people starting out. The lack of "standards" in terms of how to do things (e.g. lots of ways to do something as simple as rename a column in a dataset) can be problematic. Also, basic editing of data is very difficult. I tend to do lots of my cleaning in Excel because it is just simpler to edit a value in a cell there vs doing it in R."
[121] "Learning curve, not clear trace for inherited objects to know what other parameters we can apply from parents (i.e. when there is ...)"
[122] "steep learning curve for novices (but this is probably unavoidable)"
[123] "Its documentation. Though the learning curve is usually a feared aspect about R (for non-programmers), it wouldn't be an issue if the documentation were more clear."
[124] "The learning curve at the beginning"
[125] "Steep learning curve."
[126] "Learning curve"

[127] "Steep learning curve for those without any prior stats and/or programming experience. Discovering and learning the right set of packages for your own requirements requires time and experimentation."
[128] "Learning curve for non-savvy users "
[129] "Initial learning curve, sometimes TOO MANY ways to do something"
[130] "It is a harder learning curve than a pure GUI type environment but once learned it is very valuable for reproducible research."
[131] "Initial learning curve for command line."
[132] "The steep learning curve. The language has odd quirks that are easy to run into (like the fact that if is not vectorized), and the documentation for the base functions is often very difficult to read for newcomers."
[133] "It has a steep learning curve"
[134] "Learning curve"
[135] "Having to teach it to people who aren't willing to take the initiative to figure it out themselves. When I began using R, people told me there was a steep learning curve that paid off after you got over it, and I found this to be true. But I had the time and resources to figure out how to start using R, whereas a lot of the other scientists I work with are “too busy”/too far into their careers/too set in their ways to bother taking the time to learn, which is a real shame."
[136] "Learning curve, syntax, importing awkward data."
[137] "Steep learning curve, difficult base graphics customization, lack of a centralized resource to submit package bug fixes and new features"
[138] "Slow computations and steep learning curve"
[139] "Steep learning curve for beginners. "
[140] "Steep learning curve"
[141] "The learning curve, and the lack of examples particularly in adding notation and text to graphs. You have to experiment to get what you want. "
[142] "Learning curve. Sadly I think that the learning curve is the cost of a powerful language. "
[143] "Learning curve for non-R savvy collaborators, variation in documentation depth and utility, packages w/o vignettes"
[144] "Steep learning curve for newbies ... Packages and approaches like tidyverse by Wickham makes it more easier ..."
[145] "Learning curve."
[146] "The learning curve to learn the syntax was tough, and the documentation is not as clear cut as a beginner would like. Ideally, I think video tutorials would be the best."

Memory

[1] "- Lack of the standard way to deal with out-of-memory data.- Inconsistent naming conventions in base R packages."
[2] "Poor memory management."
[3] "Occasionally memory issues with larger data sizes. Also how to work models into production systems easily"
[4] "Memory issues .."
[5] "Anything attempting recursion. It occasionally becomes a memory hog, but that’s usually my fault at the end of the day."
[6] "Memory and the lack of"
[7] "Data must fit into memory"
[8] "If you have large indexes (datasets) that do not fit into memory, R may be not be the best choice."
[9] "Memory issues. For instance, SAS can read huge data files partially, chunk by a chunk. R cannot, it reads everything to a memory."
[10] "Speed and memory management"
[11] "Slow...taking a lot of memory;"
[12] "Running out of memory."
[13] "Definitely deploying R. In my company we have a R package with 10 dependencies. But in the end the total dependency tree have 75 packages. It takes so long to install. In addition, there isn't a easy way to install packages in the required versions. Packages always install the latest version of their dependencies. Packrat is amazing, but it's very memory expensive, since it make a full copy of R to the repository. Pre built packages with easy version specification would be amazing!!"
[14] "Can be slow. Holds too much in memory to munge big data without taking extra steps to reduce data set as much as possible. For loops also take a really long time to run."
[15] "memory based"

[16] "Performance wise, R does not handle loops very well. It requires all data to be loaded into memory. On the programming side, the biggest annoyance is that types can change unexpectedly (e.g. by subscripting when you forget drop=F)."
[17] "Memory limits"
[18] "The memory limit."
[19] "The well documented memory issues."
[20] "Steep learning curve (when you have no programming experience), sometimes different package versions produce different results, large data sets are difficult to handle because R reads the data into memory. I know there are workarounds for that and also some nice packages, but I feel that this aspect could be improved."
[21] "Speed and memory"
[22] "It's hard to think of an answer that doesn't just make me think I should study more and get better at writing code, but I'd say memory management is something that isn't very clearly communicated and can be problematic."
[23] "Memory management and the like are not particularly intuitive to folks without a CS background."
[24] "Historically the memory constraints, but easy access to more memory makes that less of an issue"
[25] "Reading large datasets into memory. My skills aren't sophisticated enough yet to handle “big data” so large datasets can sometimes be a problem when memory space is at a premium."
[26] "Memory. But it forces you to find alternative solution sometimes faster"
[27] "I've occasionally run into speed/memory issues in R, especially when working with datasets > 1 million rows. It's not enough to not run, but analysis is noticeably slower and occasionally crashes R."
[28] "R still struggles with very large datasets, even on a system with large amounts of memory. Integration with SQL is improving but still cumbersome. Need better support for parallel processing."
[29] "Limited parallelization and memory limitations to size of data"
[30] "Not memory efficient."
[31] "Memory capacity and single core ML calculations"
[32] "Dealing with big data, memory issue, and not super great ML algorithms."
[33] "Out of memory"
[34] "Small memory machines"
[35] "Slow in some aspects. Having to load data into memory."
[36] "Handling big (>16GB) objects in memory."
[37] "Memory issues"
[38] "The documentation is useless for beginners. Without Stack Overflow, it would be impossible to learn it from scratch. Matlab and Python have much better documentation. Also speed and memory usage should be improved."
[39] "Memory management"
[40] "Can't load data set larger than memory"
[41] "Performance with large datasets, difficulty parallelizing operations. Packages help, but complicated. Requirement to fit all data in memory is serious constraint at times. Even with that limitation, could handle memory more efficiently (better garbage collection, etc.)."
[42] "Memory use"
[43] "It can be slow and a memory hog"
[44] "Difficult to build a proper mental model of how R functions work without reading source code. The disconnect between learning how to do things in the tidyverse vs base R. The lack of careful memory control when working with very large data sets."
[45] "(1) Steep learning curve/hard to do things that seem like they should be easy.(2) Memory issues. Although there are many current methods/tools for dealing with this - in today's day and age (to make R the truly preferred tool), Base R and R packages should be able to better handle significantly larger data sets/computations “out of the box” (that is, you should be able to use the same R code, packages, etc. whether your data is small with thousands of records or larger with millions+ of records). (3) Fragmented ecosystem - many tools to do many things (which makes R so strong) BUT these tools don't always work well together (or tools to do the same thing work very differently) and can sometimes cause resulting complications."
[46] "R crashes instead of throwing an error when I run out of memory."
[47] "Memory management for huge datasets"
[48] "Memory handling and speed."
[49] "The way it works with in memory datasets"
[50] "Speed. memory management."
[51] "Memory limitations, especially with large datasets or complex models."

[52] "Using R in production is not the easiest. Do-able but requires some good software development expertise, which might be hard to find from people with mathematical/statistical backgrounds. Also, single threaded by default and limitations of memory can be a bug bear. Comments apply to the open source version on CRAN."
[53] "Restrictions in memory usage. When you reach the borders of what's there, it not just gets "harder" it somehow gets impossible, without going directly into CPP and implementing stuff by yourself. Please change how Restrictions on DLLs are managed. We live in the time of even more complicated packages. I wasn't to choose, how many packages I use and need for my analysis."
[54] "Use a Lot of memory, don't remove objects after rm Command, loops are slower than C or Python"
[55] "Struggling with processing, modeling, and memory limitations on large datasets."
[56] "Speed for anything not fitting in memory."
[57] "Inconsistent order of arguments of functions (core and packages). Sometimes memory usage, gc doesn't always clean up. Wish it could use a swap space or page file - willing to take the performance hit (solid state drives are pretty common now)."
[58] "Damned slow. Tries to put everything into memory. Parallelization is cool with the snow-like interprocess communication and also the fork-like with mclapply(), but more multicore support would be great."
[59] "Memory restrictions"
[60] "Memory limitation Parallel processing"
[61] "Memory overflow when work with large data. Python is better!"
[62] "Memory handling with having to (usually) load everything into RAM and subsequently not releasing all of the RAM when e.g. removing an object."
[63] "Memory usage and CPU usage"
[64] "Data size, memory, and processing limitations."
[65] "Working in memory for bug dataset issues"
[66] "Slow at times and memory-limitation. No tools available to write production grade code, to my knowledge"
[67] "Lack of speed and performance. It should use all the resources of the computer when possible, including: 1) Using all cores when possible for a given calculation without needing to explicitly give it the instructions2) If there's a part of the code which don't depend on previous code it should start calculating both branches in parallel on different cores to win some time3) If it runs out of of memory, use a portion of the SSD as memory(By the way Alteryx does all of this and is super fast (a lot faster than R), if Alteryx is such a high level, user friendly, drag and drop style, how come it outperforms raw code in R (defeats the purpose of coding, the tradeoff of performance vs. ease of use doesn't work here)"
[68] "Working in Windows, facing issues with file encodings. Running it memory limits."
[69] "The worst aspect of working with R is being single-threaded and having memory limitations. These two aspects are very damaging to scaling web applications with Shiny and analyzing datasets at web scale without having specialized knowledge of distributed computing with AWS or other systems."
[70] "Memory management, inconsistency across packages (for instance, recovering estimated coefficients from the models in different packages)."
[71] "The memory limits"
[72] "Interoperability with some datasets / data format. Memory needs for large datasets"
[73] "Memory management"
[74] "Memory"
[75] "The "everything in memory" thing, Interactive visualization is not all that good in R, sometimes edits using point and click are easier (trying different colors...)"
[76] "- Speed when you really need it.- Some memory issues (the usual object duplication etc. as a result of pass by copy, etc.).- Not necessarily a good language, pedagogically speaking (yeap, I miss a scheme or racket)"
[77] "Memory and data structures"
[78] "Performance and memory restrictions"
[79] "Its too slow. Loads everything into memory and my computer crashes. Python and C++ are so much faster. The solution is to make it easier to work with python and C++ code from R and RStudio, so you can use them when you need real speed. Also, the documentation is pretty bad. Why do I have to go into the ?par documentation to find parameters that should be in ?plot. Just include everything where it should be! This drives me crazy. R docs and rmarkdown need to take a leaf out of jupyter notebooks and python documentation."
[80] "strict adherence to in memory data.some algorithms are slow. Ordering of axes in ggplot is a pain in my arse. dplyr does some funny stuff sometimes. Documentation is too sparse and incomplete much more often than i'd like."
[81] "Memory handling could be better, more efficient (i.e. less copying, less mutability of types)"
[82] "Memory management with large datasets"
[83] "Speed, memory"

[84] "Object in memory handling. Memory is rarely enough, especially with larger sets."
[85] "Some things are confusing, usually there are many ways to achieve a goal, which don't always combine. And the performance isn't great, sometimes just slow, eats a lot of memory..."
[86] "One is the limit of memory of the size of the arrays. Another it is that sometimes it results complicated to control all the elements on a figure. There are many different packages to make maps and plots, but they don't have the control that I expect over a graphic. In this sense I think python is more complete."
[87] "Memory limit and need to change code to parallelize tasks"
[88] "Speed of loops, memory limitations"
[89] "Memory performance, horrifying garbage cleaning, and the language as a general purpose language."
[90] "Slow and memory hungry."
[91] "The loops and memory management."
[92] "R is sometimes slow and is not that easy to work with large datasets as R keeps everything in memory."
[93] "Speed, memory limits, non existent support for multithreading, multicore, gpus."
[94] "Memory problems"
[95] "Memory issues. I have problems with some machine learning algorithms which work better in Python"
[96] "The worst aspect of R is layers of legacy functionality, kept for backward compatibility. Poorly written documentation. Slow loops. If used improperly, memory copies in loops.do.call() can use up all memory and grind to a halt."
[97] "Memory. I struggle when trying to work with large datasets. I wish it were easier to read in parts of csv files instead of reading in the entire thing then dropping cases/columns. It's also been tough for me to learn how to take advantage of R's vectorization (e.g., using lapply commands instead of for loops), but that's not so much of an issue as the memory."
[98] "Convincing people that the memory problem is not a problem"
[99] "The memory required to run big data. For Kaggle competition sometimes I had problem to upload 1 million of dataset rows. Time lost to run machine Learning. There are others platform performing better and this is a pity."
[100] "Dealing huge database (due to in-memory analysis) slow vague and confusing coding grammar"
[101] "Reading in files that don't fit into memory. Hitting memory limits is a pain."
[102] "Memory constraints. For many problems, R just can't handle the data anymore, and I need to switch to Python."
[103] "Use of memory to store data"
[104] "Speed and memory"
[105] "Out of memory error, not all CPU used, could have a better read.table function that does more automatically"
[106] "Uses lot of memory"
[107] "Base R is often slow and memory management us not great"
[108] "Bigger files years ago when ram memory wasn't so cheap... For production keeping a server updated could be painful. Sometimes you need to stick to a version and then look for compatible packages and compatible dependencies."
[109] "You have to load the data in memory, which limits its size."
[110] "Speed, memory limitations"
[111] "Sometimes it is slow or uses too much memory."
[112] "Only use RAM memory"
[113] "The heterogeneity of the syntaxes because of the different package logics. Some are written with S3, S4, OO, functional logic etc. and the mixed logic of parameter definitions. Maybe it would be better if the packages were more standardized. The memory issues, ad-hoc crashes, badly written inefficient packages."
[114] "Memory handling and efficiency, although some efficiency issues are alleviated by Rcpp. While there is improvement in handling large data files (such as big.memory and ff), there can be improvements in r/w speed."
[115] "It not able to work with data larger than memory transparently for the user. Only some trivial functions are implemented in some very esoteric and difficult to use packages. Working with loops is very slow, you need to vectorize your code, sometimes it's a problem. It doesn't have a good interactive spreadsheet-like GUI. It's difficult to find a function among so many packages, each working in a different way and many times incompatible. It lacks of a good interactive plots designer. And many other things. Now I'm considering moving to other tools such as Tableau, Julia or Spark."
[116] "Since most of the algorithms work in a single data set in a non-distributed fashion, the inherent in-memory nature can be extremely slow with large datasets."
[117] "Memory dependency!"
[118] "In-memory data makes process of massive datasets slow in the standard hardware that we use in emerging countries."

[119] "Debugging. Sometimes, I'll need to step through lines in a package or compiled Java or C library. Sometimes, code runs in one environment but not in another. Sometimes I get wacky memory errors, freezes but no usable info."
[120] "R is limited to data types that can fit into memory. Yes there are special packages that try to get around this, and not every problem can be broken down into smaller parts and regrouped. Sometimes you just need to deal with gigs of data in memory. R requires you to have a lot of memory to deal with that. I wish R would have data types that could work off the hard drive instead."
[121] "Big memory data. Data.table rocks for speed but requires a different syntax. "
[122] "Memory management and the duplication of objects in ram. This becomes troublesome when dealing with large files"
[123] "The documentation is difficult for newbies. Package documentation quality is not consistent across packages. Dealing with CRAN maintainer are sometimes difficult. Recreating some of the tests in CRAN such as memtests in GCC and CLANG is very difficult. R-hub seems to be promising. Handling large datasets in R is also difficult due to memory limitations."
[124] "Establishing ODBC connections to large databases seems very difficult. The necessity of reading flat files into memory is often frustrating."
[125] "Running out of memory ... "
[126] "It wastes memory. This can be annoying when dealing with large data sets."
[127] "Ram memory"
[128] "Memory. It can't work with very large data sets - so one is always dependent on RAM size"
[129] "Everything is kept in memory"
[130] "it is memory intensive language, the syntax is sometimes counter intuitive when compared to other languages such as Python"
[131] "memory efficiency when working with large data sets"

Parallel

[1] "Parallel computations can be tricky."
[2] "Multithreading/working in parallel"
[3] "Single thread (there are solutions / openR/ parallel ect)"
[4] "Improving performance through parallel computation."
[5] "R still struggles with very large datasets, even on a system with large amounts of memory. Integration with SQL is improving but still cumbersome. Need better support for parallel processing."
[6] "Limited parallelization and memory limitations to size of data"
[7] "Terrible parallel processing support lacks a ML pipeline like scikit learn."
[8] "Tracking down errors in parallel backend is a bitch"
[9] "Not easy to productionize or parallelize easily as python."
[10] "Lack of native parallelism"
[11] "Performance with large datasets, difficulty parallelizing operations. Packages help, but complicated. Requirement to fit all data in memory is serious constraint at times. Even with that limitation, could handle memory more efficiently (better garbage collection, etc.)."
[12] "Sometimes I have to use loop to do some calculation and I don't want to write a compiled code with C++, it can be very slow. The parallel package sometimes can make the whole R session crash. And the parallel job is very hard to terminate before the job is finished."
[13] "Can be slow and parallelization is quite underdeveloped."
[14] "Working with dbs and big data sets -lack of parallel version of functions or algos"
[15] "Runtime is slow, parallel processing is difficult to manage"
[16] "The parallel existence of a “base R” and a “tidyverse” dialect."
[17] "Damned slow. Tries to put everything into memory. Parallelization is cool with the snow-like inter-process communication and also the fork-like with mclapply(), but more multicore support would be great."
[18] "Slow computation times in some situations, R base should have more parallelized functions. BLAS could be improved."
[19] "Parallelization."
[20] "Memory limitation Parallel processing"
[21] "Lack of speed and performance. It should use all the resources of the computer when possible, including: 1) Using all cores when possible for a given calculation without needing to explicitly give it the instructions2) If there’s a part of the code which don't depend on previous code it should start calculating both branches in parallel on different cores to win some time3) If it runs out of of memory, use a portion of the SSD as memory(By the way Alteryx does all of this and is super fast (a lot faster than R), if Alteryx is such a high level, user friendly,

drag and drop style, how come it outperforms raw code in R (defeats the purpose of coding, the tradeoff of performance vs. ease of use doesn't work here)"
[22] "Having to really think about parallelization."
[23] "This might not be true now, but being able to easily parallelize jobs"
[24] "Each package is different (has different way of working). Parallelism (multicore) doesn't work by default"
[25] "parallel or using multicore"
[26] "Implementing parallelization"
[27] "Parallelization in R is very tricky"
[28] "Memory limit and need to change code to parallelize tasks"
[29] "Occasional irregular syntax. Some external packages can be slow, so simulations can take 1000s of cpu hours, 100s of clock hours via doParallel."
[30] "Speed and parallelization"
[31] "Speed. I find very annoying being able to only work with one processor core or setting up parallelism in R."
[32] "1. low speed compared with C/C++ and needs to convert for speed up 2. Need to help from other platform to do distributed parallel process such as h2o"
[33] "Slowness. Lack of capacity for big data. Lack of easy interface with big data tech. No/little parallelization."
[34] "Some times a lot of coding is needed for simple tasks. Speed and parallelism. "

Syntax

 [1] "The different fiefdoms result in some inconsistencies in syntax and while most of the community is great, there are enough occasions of snarky responses to help requests that it can be intimidating enough to turn off new users."
 [2] "Too many syntax styles"
 [3] "As I mentioned, not being able to use Latex syntax and fonts directly into the generated plots. This should be possible and implemented like in Pythons matplotlib"
 [4] "The lack of consistency in naming conventions and syntax. Every package uses slightly different syntax which makes remembering how to do things harder (though autocomplete in RStudio helps a lot here)."
 [5] "Syntax, documentation, speed and execution are horrible"
 [6] "Debugging in R is a mess. Syntax is sometimes absurd."
 [7] "Inconsistent and unconventional syntax. Inconsistent and unconventional data structures"
 [8] "Teaching the crazy syntax to new users, e.g., [], [[]], $. Trying to explain data.frame vs data.table vs DataTable vs data_tbl to new users. S4 is opaque and cumbersome, even for experienced R users, but there's no escaping it because I need bioconductor functionality."
 [9] "Syntax can be daunting"
[10] "Syntax differences between different base packages and tidyverse - prefer tidyverse. So many packages to choose from it can be hard to know which ones to use - or even ones that would be useful exist."
[11] "Learning the syntax..."
[12] "The lack of consistent syntax in base R packages (the `as.Date()` vs `as.character()` problem). R also lacks any tools that can compete with Python when it comes to web scraping. rvest is good, but I have not found it to be as powerful as scrapy."
[13] "Syntax can be painfully arcane. Would much rather use basic for loop structure than remember which apply function works best or the purrr .x .y .f ~ ..."
[14] "The syntax very hard to get used to"
[15] "Programming for the web feels like a hack. could use stronger type system. syntax isn't very consistent."
[16] "I've had difficulty converting others to R due to its reliance on outside packages that may use different syntax (such as learning to use strange operators like “%>%” for dplyr)"
[17] "The inconsistency of syntax across the whole ecosystem, e.g., the requirement for quotes when selecting variables in some situations and not in others. This often creates confusion - though arguably it often creates flexibility also."
[18] "Trying to teach others how to use it. Having to explain how to do something in base R and it being completely different syntax for dplyr and ggplot etc."
[19] "The syntax is ugly, it's very hard to write good code in R."
[20] "Inconsistent syntax once you get far away from Tidyverse"
[21] "The syntax is showing its age -- lambda functions are so verbose, NSE is mad and wouldn't have to be with modern syntax."
[22] "There are some performance issues, especially with the basic loop constructs. I'd like to see an LLVM back-end. I'd like to see more consistency in quality on CRAN. I'd like to see more models use the tidyverse syntax and approach. FACTORS. OH MY GOD ANYTHING INVOLVING FACTORS. I'd like to see more consistency

in programmatic approach on CRAN. The difference in base assumptions between (for example) markovchain and tidyverse is awful, they could be in different languages. Too many assignment operators. Dynamic typing. Working with dates is painful."
[23] "For those with little/no programming (or basic command line interface) experience, there is a barrier to entry. The statistical analysis output is not "built-in." The advent of rmarkdown has helped me immensely. Output of summary data to tables is still not easy, especially for beginners. The advent of multiple table-related packages is helping, though. I find the `htmlTable` package to be the most useful. The base R syntax is not consistent in terms of order/naming of arguments. This is not clear to beginners and took awhile to get used to. It wasn't until after learning dplyr flavor syntax that I realized how inconsistent it is. I feel that experienced R programmers don't realize how inconsistent the syntax is, and just live with it."
[24] "I'm not on the job market yet (about to start last year of PhD), and I'm worried knowing R isn't enough. Hence learning Python, begrudgingly. The worst part might be the confusion of rmarkdown. Sometimes you code in LaTeX, other times the rmarkdown syntax, other times HTML can pop in there. I dunno, I find it confusing using a bunch of different languages in one document without a clear delineation of what is going to work and when."
[25] "Different syntax/naming schemes for different packages--no strictly adhered to set of rules that is followed among all packages"
[26] "Missing syntactic sugar. A good list of these is provided by Jim Hester (https://rawgit.com/jimhester/presentations/master/2017_07_03-DSC2017-Syntax_Extensions_To_R/2017_07_03-DSC2017-Syntax_Extensions_To_R.html#/r-syntax-extensions)."
[27] "How terrible so much of base R is. Weird syntax, functions with inconsistent return types, stringsAsFactors = TRUE, etc."
[28] "Speed and scalability is an issue with production code and the syntax can be a bit strange."
[29] "Inconsistent syntax and formulas. As a hobbyist programmer, I had trouble getting much use from it before I started using dplyr / tidyr / ggplot2."
[30] "The syntax is not intuitive, and running operations over a dataset is staggeringly complex compared to any modern language I've tried."
[31] "The lack of consistent syntax between packages, but it's not a major issue for me."
[32] "Haha syntax. Scope. R is a terrible programming language. But an amazing data analysis language. Try attaching dplyr then biobase and use select()."
[33] "I never really know if I'm "doing it right." I'm completely self-taught in R, and it's my first programming language. I know that there are tutorials out there, and I do often look through them, but I still never really know if I have the correct syntax, etc. For as long as I've been using R, I would like to be more savvy and knowledgeable about it, but I'm not even sure I could teach it to someone else."
[34] "There are many ways to do something and sometime it is not very clear which is better syntax is not always consistent between package"
[35] "Sometimes just get caught up in the language itself - syntax, some libraries play better with others... - when I'd like to be more focused on getting results"
[36] "Inconsistent syntax; inconsistent documentation quality; many ways to do the same thing."
[37] "Steep learning curve which is exacerbated by lack of rules on style and syntax. R may be too flexible."
[38] "The syntax is sometimes cryptic or inconsistent, and every once a while something produces unexpected results. The flip side of R's flexibility is that there are usually many ways to accomplish something, which can be confusing. Sometimes I teach my students one way to do something, and they find another way to do the same thing on the web, and get confused. I believe that is an unavoidable price to pay for the flexibility."
[39] "Core R syntax is very inconsistent and fragmented. Makes learning R difficult."
[40] "Syntax sometimes is a bit not so intuitive. Especially for someone transitioning from another language."
[41] "Arcane syntax, every package doing everything a different way (Twidlr helps a lot with adding formula-notation to many packages)."
[42] "Sometimes the syntax for one function is not consistent with other functions which are otherwise similar."
[43] "Documentation of packages. Have you ever tried to find the right syntax for ggplot2 calls from the help texts? Awfull! From all the languages I have peaked into, working with strings/characters is least fun in R."
[44] "Error messages in R are frequently useless or even worse, misleading. I know that error documentation is not high on any package developer's to-do list, but discovering the true cause of failure in syntax code is frequently frustrating to the end user. I have said nasty things in this survey about SPSS, SAS, and Stata, but I will say that their error message reporting is superior to R."
[45] "Remember specific syntax rarely used"
[46] "The fact that the different packages use different syntax for their methods. Some use package.method, some method(data) etc. Also, the fact that the language does not force you to follow a specific syntax. In contrast, Python forces you to indent the code, otherwise it will not run."

[47] "Syntax and can be slow at times when dealing with data at scale."
[48] "Becoming accustomed to syntax."
[49] "Learning R syntax."
[50] "Often obtuse syntax"
[51] "Inconsistent syntax, but tidyverse is recent help with that"
[52] "I think new users are often befuddled by the inconsistent syntax (eg: plot(x,y) vs plot(y~x))."
[53] "Often weird, unintuitive syntax. Things that are trivially easy in other languages can be difficult in R. User community used to be really snobbish (StackOverflow is a vast improvement over old R lists)."
[54] "The not-so-consistent syntax, which may get worse as packages are updated (e.g., ggplot). It's annoying when a slightly old piece of code won't run because a tiny little detail has been updated. I'm ok with the learning curve, and I'm even ok with the clunky syntax (relative to Python, for example), but the inconsistency of the clunkiness is a pain sometimes."
[55] "Syntax can be ugly and the various DSLs can make collaboration challenging (ie tidyverse vs base vs data.table)"
[56] "syntax has a huge learning curve"
[57] "Quirks of data types and syntax can be somewhat hard to parse"
[58] "Sometimes the syntax can get confusing"
[59] "the inconsistency of the syntax across packages can be frustrating at times."
[60] "Quirky and unusual syntax features (especially for beginners) -- two ways of assigning values, massive chaining operator and so on."
[61] "Learning new packages can sometimes be like learning new languages since there is often syntax that is specific to it. Also, sometimes R will do weird things to your data types without you realizing which can affect downstream processes. With more experience this doesn't affect me as much but it used to impact my work A LOT"
[62] "syntax! some of the byzantine syntax still does my head in after working almost exclusively in R for 8+ yrs."
[63] "R has quite a few inconsistencies that can make the language feel hard to grok as a new user. This is not improved by splintering domain specific syntax, inconsistent interfaces and the dizzying number of OO systems. All of this can make “on-boarding” difficult."
[64] "The picky syntax."
[65] "Fragmentation and inconsistent development. Because R packages are products of the community, there are a bajillion of them, their quality is uneven, and development can stop abruptly at any time. Sometimes there are a dozen different packages available that handle the same types of problem and it their methods for doing so are very different. Choosing one approach can sometimes lock you in, which is unfortunate if you discover you need some unimplemented functionality down the road. Sometimes great packages have nonsensical syntax and poor documentation. Sometimes a great package hasn't been updated in years and there is not a solid replacement for it."
[66] "The learning curve is steep. Inconsistency in syntax, requirements and performance between packages."
[67] "Inconsistent syntax across packages learn curve"
[68] "The syntax is rather quirky - most newbies nowadays would come from a background of C/C++, Python or Java - the syntax of R looks a bit alien / old fashioned in comparison. For instance most tutorials and examples use <- instead of the equivalent and more familiar =. Also functions and classes seem to be an afterthought."
[69] "Scaling. Having the overview over packages Syntax gap between packages No sklearn-like all in one toolbox for machine learning"
[70] "Inconsistent language syntax."
[71] "Many features would be better off with a traditional OOP implementation - e.g. date time objects (POSIXct). Odd syntax at times: assignment “<-”, “.” allowed in names, “$” as member access, “c()”."
[72] "Getting used to the syntax. I've programmed in some other languages, and it's definitely harder learning R, because it's so different."
[73] "non-intuitive syntax"
[74] "more and more packages come up but sometimes their syntax seems a bit confusing to original r base code"
[75] "Sometimes confusing syntax, limited documentation, and the frustration of having to hunt among 10,000 packages to find the functions I need."
[76] "Slow for loops, inconsistent syntax in the base core functions."
[77] "There’s a steep learning curve, and the syntax feels unforgiving to beginners. In the days pre-tidyverse (year 10 ante Hadley) it took me three dedicated attempts to get warm with R."
[78] "Quirky syntax."

[79] "Slow in certain use cases, very messy function call syntax, dislike for tidyverse syntax (I know I'm a minority here!)"
[80] "Occasional irregular syntax. Some external packages can be slow, so simulations can take 1000s of cpu hours, 100s of clock hours via doParallel."
[81] "Data structures are odd. Documentation is patchy. Syntax is non-uniform. Quite a lot of functional redundancy - some fast, other cumbersome and slow."
[82] "Frustration when using new syntax and not being able to go through it."
[83] "Time needed to learn language/syntax/function"
[84] "Inconsistencies in the language, sometimes unintuitive baseR syntax, some general purpose things are easier in other languages (web apps with Python & Flask, web scraping and browser steering, deep learning)"
[85] "syntax sometimes opaque, so difficult to remember what needs coding to say something simple, but this is probably similar to any language at the level at which I use it!"
[86] "Ugly syntax. Inconvenient documentation."
[87] "Extremely inconsistent syntax."
[88] "Somewhat obtuse syntax and unhelpful error messages at times"
[89] "Learning the basic logic and syntax."
[90] "Inconsistency in syntax"
[91] "Learning the syntax of every package. Also when some things are out of date and then nothing works"
[92] "Hard to remember syntax. Methodology used by different packages makes it difficult to remember everything."
[93] "sometimes odd syntax"
[94] "Not intuitive, as packages are different in syntax and output."
[95] "The heterogeneity of the syntaxes because of the different package logics. Some are written with S3, S4, OO, functional logic etc. and the mixed logic of parameter definitions. Maybe it would be better if the packages were more standardized. The memory issues, ad-hoc crashes, badly written unefficient packages."
[96] "The language has been split into those who use the tidyverse and those who don't, when the syntax is so different it literally prevents R programmers working together on the same project. It's quite deeply concerning that R has put higher value from upon people getting results quickly at the cost of bad coding practices being adopted. CRAN is now saturated with packages that do exactly the same thing. I fear that R has to answer how it will help it's developers build production level coding standards and stop focusing on short term syntactical sugar to woo new users. Failing to address some of these significant problems will hurt the longevity of the language."
[97] "Lack of type system. Sometimes different syntax."
[98] "Different syntax for several packages, sometimes non-intuitive"
[99] "Have to google everything and rely on stack overflow, don't feel like the syntax is easy to remember replicate (even for tidyverse, though that's certainly better), has the problem of any programming language that it's reliant on me to fix typos, etc. Would love for R to be a bit more like OpenRefine, where there are workflow options that are coded into a GUI and that can be reapplied easily on new datasets. I'm working on a big data analysis project, and every time I realize there's something else I want to capture, I'm stuck going back through all of my code, manually changing variable and file names, etc. Sure, I can right my own functions, and I'm working up to that, but I wish there was a better development environment for beginners that produced the "best practice" code, more like Stata or maybe SPSS. Maybe Exploratory.io is the answer, but I haven't tried it much yet."
[100] "It is not easy to find which is the "right" way of doing a task. By "right", I mean the best standard way and some hint to more specific tuning and more Advanced used. Moreover, sometimes, performance could be a drawback. Also, R is very permissive and writing working script is easy even if it is kind of dirty. Writing code with good practice, and standard way is not so easy (comparing to python syntax for example, or module). it is a force that user can get to their aim very easily but it is painful when you have to put this code into production or to share this code. (comments with a company insider point of view)"
[101] "The more complicated the routine, the more evident it becomes that I am trying to tie together the reasoning of hundreds of different people, each with their own ideas of what makes a good function interface, or good syntax, or realistic practice. The beautiful communal aspect of the language, the broad contributions from highly diverse sources with particular expertise and needs, is by some turns a miracle but by others an anthropological nightmare. It's the zoo-like quality that for me limits R to work outside of production. Somewhat related: it's also very difficult to proof code well. Error messaging is often inscrutable, a problem amplified by how frequently package updates seem to introduce breaking changes."
[102] "I often have trouble with the syntax and struggle to do things I know are simple to do in SAS like re-format data."
[103] "learning syntax"

[104] "Syntax is not always logical across packages and several packages have taken off with quite different language and usage (such as dplyr etc.). I am self-taught in base and have not been able to get my head around these other packages in a good way."
[105] "As with any other foreign language you need to use it. If you work with R just occasionally you can perceive difficulties with vocabulary nad syntax. But this is solved by excellent documentation and convenient R-help"
[106] "Sometimes the R syntax is horrible"
[107] "Some times the scripts are a lot ...and always need to look up at the syntax of the function "
[108] "data.frame syntax"
[109] "It can be slow. There also some counter-intuitive aspects to the programming/syntax."
[110] "The language itself: syntax is difficult to learn and it is somewhat hard to write performant code."
[111] "The base syntax and everything that makes it tedious and frustrating as a general-purpose programming language."
[112] "Big memory data. Data.table rocks for speed but requires a different syntax. "
[113] "Learning curve, syntax, importing awkward data."
[114] "Don't like syntax"
[115] "Syntax can be inconsistent, and so I need to do a lot of cutting and pasting. Certain commands produce strange output (e.g., bwplot in mosaic package). Rarely, my students have bugs that are hard for me to fix. The need to use dataframe$variable to denote variables in many commands in base R is annoying, and attach/detach is no better. "
[116] "it is memory intensive language, the syntax is sometimes counter intuitive when compared to other languages such as Python"
[117] "No dictionary/hash data structure. Syntax varies too much from package to package. "
[118] "The learning curve to learn the syntax was tough, and the documentation is not as clear cut as a beginner would like. Ideally, I think video tutorials would be the best."

Slow

[1] "Slow data, inconsistent command styles, and large numbers of functions reserved for unnecessary compatibility"
[2] "Unstable, slow, typically poor code documentation/low test coverage - has more to do with programmers than with programming language. There's no culture for test-driven development in R community yet"
[3] "By default, stringsAsFactors and drop are TRUE. Loops are still slower than in other languages. S3 and S4 classes are awkward and do not follow all of the traditional OOP paradigm."
[4] "Some very bizarre errors, particularly the explanation of them. Can be very slow, need to sort out the issue with for loop (I think it does a garbage collection). Also, wonder if it might be possible to have float32 as that could speed things up."
[5] "Sometimes its a little slow and function writing is kind of annoying in comparison to python"
[6] "Managing package dependencies and back-ups. It's hard to make a project fully reproducible because of difficulties managing package versions. Microsoft's checkpoint is unsatisfying because it means we have to rely on Microsoft (eww). Packrat is very slow and buggy."
[7] "Slower to perform analysis than in Excel"
[8] "That I am a slow learner and I get stuck sometimes."
[9] "It can be slow - while there are a number of code optimizations you can do to speed it up, once you know what you're doing and how R works internally, for new users these tricks aren't always obvious."
[10] "Learning it- the curve rises very slowly"
[11] "Slow, especially on the older machines I have to use at work."
[12] "It is slow. Mean very slow if i need to process large datasets. I use rcpp to gain speed but it would be very nice to have agility of accessing to variables by reference. I think this makes it much more faster. Right?"
[13] "Can sometimes be slow"
[14] "Slow, inconsistent function calls"
[15] "Speed. Sometimes it is too slow for the more computationally expensive algorithms I use. I would prefer to use python over c++ to increase efficiency."
[16] "Slow...taking a lot of memory;"
[17] "It can be slow. I wish it was multi threaded by default."
[18] "Missing out on python's web programming capabilities slow + inadequate development of the language itself (64 integers, windows build system using very old gcc, documentation that needs an improvements, closed development where instead of direct git pull request -not related to github per se- one has submit patches to mailing list...."

[19] "Can be slow. Holds too much in memory to munge big data without taking extra steps to reduce data set as much as possible. For loops also take a really long time to run."
[20] "Can be slow. Often there's a fast way and a slower way to do things, and the difference is dependent on R internals and hard to know in advance."
[21] "Slow, cannot handle massive data"
[22] "Two complaints. First, R can be terribly slow for certain operations that I do (Bayesian models or large micro econometric datasets with 5 Million observations). Second, in my area (empirical macroeconomics) there are surprisingly few advanced packages on CRAN or in general code available. Most of the time I have to rely on matlab toolboxes and code for things like structural Bayesian vector autoregression analysis (there are related packages but their functionality and usability is far less advanced than what is available in matlab or even Python)."
[23] "Can be slow, visualizations can be better (Matlab is better, ggplot can be hard to use)."
[24] "Can be slow"
[25] "Sometimes could be slow and it is difficult to handle large dataset (in comparison with Python)"
[26] "It can be slow. It also seems to be unable to fit some models that SAS can with ease. For example linear mixed models with large amounts of data (on the order of 100,000 observations) can be fit in SAS but crash or do not converge in lme4."
[27] "Very slow at times."
[28] "Slow performance and kinks of base r, needing to know what to avoid and supplement with better packages. Occasionally quirky, inconsistent behavior. Lack of pure functional aspects."
[29] "Slow in some aspects compared to Python, but that is about it."
[30] "It's slow and I hate having to write in other languages to speed things up. Also fragmentation- r packages such as tidyverse are overtaking many base functions for getting stuff done. I think that core / base r should reflect what people actually use and what is best for the future of the language. Discovering new packages and functions is very difficult. Lots of active r developer interact on twitter and GitHub, but it is difficult for users (rather than developers) to find new and well developed packages."
[31] "Slow implementation of algorithms for working with large Datasets."
[32] "Attempting to create procedural programming in Base R. This is a fact of R, but most loop structures in base R are very slow."
[33] "Slow performance depending on how I write the code"
[34] "I've occasionally run into speed/memory issues in R, especially when working with datasets > 1 million rows. It's not enough to not run, but analysis is noticeably slower and occasionally crashes R."
[35] "It gets slow with big datasets."
[36] "Non-vectorized code can be very slow."
[37] "Skeptics claiming that R is slow and that Python is better :)"
[38] "It is slow. It is a domain specific language."
[39] "Slow server side (http) and the fact that it is not used by others in the team. Yet????"
[40] "It's a bit slow with big data"
[41] "Slowness"
[42] "Slow speed"
[43] "Slow and it's a weird language"
[44] "Slow in some aspects. Having to load data into memory."
[45] "Slow Graphics"
[46] "It can be slow at times. Frustrating when you have to run C++ through R and use a new language for complex models. Also difficult to find out optimal ways of coding to increase speed. While the documentation is there for how to do stuff it's more spares for improving and speeding up your code."
[47] "Lack of Type checking, slow speeds. *apply functions. They are easy to write and take up less space, but terrible to read."
[48] "Some machine learning algorithms are noticeably slower than their Python counterparts"
[49] "R/shiny doesn't run on windows and is generally slower than d3. I switched to d3 for viz because of that."
[50] "Sometimes I have to use loop to do some calculation and I don't want to write a compiled code with C++, it can be very slow. The parallel package sometimes can make the whole R session crash. And the parallel job is very hard to terminate before the job is finished."
[51] "Can be slow and parallelization is quite underdeveloped."
[52] "In some situations quite slow but probably I have to improve my code as well ;-)"
[53] "Somewhat slow, so it is necessary to learn the C++ programming for example."
[54] "Runtime is slow, parallel processing is difficult to manage"
[55] "It can be slow and a memory hog"
[56] "Slow speed and arcane language."

[57] "It is slow"
[58] "Use a Lot of memory, don't remove objects after rm Command, loops are slower than C or Python"
[59] "Some bugs, a little slow sometimes..."
[60] "Slow performance with running few algorithms"
[61] "Base R is slow and there seems to be no reason for it. There isn't much innovation in Base R and there is little consistency among output formats. Packages like Broom help, but this should be something that Base R handles."
[62] "Damned slow. Tries to put everything into memory. Parallelization is cool with the snow-like interprocess communication and also the fork-like with mclapply(), but more multicore support would be great."
[63] "Slow computation times in some situations, R base should have more parallelized functions. BLAS could be improved."
[64] "Run times can be a bit slow."
[65] "Runtime is inevitably slower than compiled typed approaches. But the worst is that R will never become multithreaded itself"
[66] "Slow loops, etc."
[67] "Too slow, crashes sometimes"
[68] "Sometimes slow computations times"
[69] "Slow run times on large datasets"
[70] "It can become very slow with large data. A lot of this has to do with the servers we are running the code on. Package versions are constantly changing and our company doesn't have infrastructure to appropriately handle a production and development R environment. Difficulties writing production R code that the company can ingest."
[71] "Slow with large datasets"
[72] "Syntax and can be slow at times when dealing with data at scale."
[73] "Slow at times and memory-limitation. No tools available to write production grade code, to my knowledge"
[74] "Slow processing times"
[75] "I think the debugging tools of R are not well known, and I don't know if they are very usable. I think I tried to use them a few times and they did not stick. Another big concern for me - and this is controversial I think - is that I think the influence of RStudio is not great for the language. There are many great things that people can do with the R language that require the use of RStudio, and the RStudio team has so much control over some of the most popular packages in R that I worry that the language itself might slowly become inseparable from RStudio or weaker without it."
[76] "Slow performance, ancient kernel (bad Big Data capability, RAM-limited, no native multicore)"
[77] "Sometimes it is hard to find packages. Sometimes R code is slow unless I know tricks or other languages or specific best practices to make it faster. Often code breaks when packages are updated, so using something like packrat is important but not everyone does this for every project."
[78] "R can be slow because it is not compiled (if it's too slow I rewrite it in Python)"
[79] "The slow startup time, maybe? I realize I could fix that, but the fact that it takes longer to get RStudio started up and data loaded into it (reading from CSV or from a DB) keeps me from using it for a lot of tasks."
[80] "Slowness"
[81] "Managing packages and versions. Slowness."
[82] "Slow"
[83] "Can be slow"
[84] "Can be slow."
[85] "Not ideal for OOP Overhead of r environment can make it slow compared to python/java. Many packages available but many use outdated data structures and are very slow on large data, requiring customization Lack of integration in cloud environments (e.g. dynamoDB/AWSlambda)"
[86] "Tons of functions that I have no idea how to use or why I would use. Number of packages on CRAN and GitHub also seems overwhelming. It is also a data hog and at times - slow."
[87] "Its quirks as a programming language, S3 and S4 classes confusing for beginners (and most people who don't know OO programming), [] and [[]] being different and not obvious, = and <- being different is similar, but harder to screw up on, lists can be so useful, but I only slowly discovered all their features because they were confusing and did not encourage use. Speed of the language, not noticeable when you have you have a 100 by 10K table. A lot more noticeable with 100K rows."
[88] "Its too slow. Loads everything into memory and my computer crashes. Python and C++ are so much faster. The solution is to make it easier to work with python and C++ code from R and RStudio, so you can use them when you need real speed. Also, the documentation is pretty bad. Why do I have to go into the ?par

documentation to find parameters that should be in ?plot. Just include everything where it should be! This drives me crazy. R docs and rmarkdown need to take a leaf out of Jupyter notebooks and python documentation."
[89] "Some times is slow and limited to use large datasets. Another point is the interaction with other languages. Python is much more popular and easy to use with web data..."
[90] "Strict adherence to in memory data. Some algorithms are slow. Ordering of axes in ggplot is a pain in my arse. dplyr does some funny stuff sometimes. Documentation is too sparse and incomplete much more often than I'd like."
[91] "Some of my datasets are getting too large to fit onto RAM, and I find the existing tools for working with very large datasets to be not particularly straightforward to use. Also, some data analysis functions are too slow to realistically use in large scale simulations."
[92] "Slow to work to scan through files and perform computations without loading the entire dataset."
[93] "Slow on a regular PC with millions lines of record. A big speed gap with Python"
[94] "Slow for non vectorial operations. No static typing. Clumsy OOP implementation."
[95] "A bit slow"
[96] "Figuring out how to do everything with vectors and avoid slow looping can sometimes be challenging."
[97] "Minor: Slow loops. Major: in my field (Engineering, Computational fluid dynamics) not so many people know it."
[98] "Slowness"
[99] "Some things are confusing, usually there are many ways to achieve a goal, which don't always combine. And the performance isn't great, sometimes just slow, eats a lot of memory..."
[100] "Not using for cycle because it is too slow"
[101] "Readability of code. It would be cool to have more options for structuring pure-text files, e.g. different types of comment levels. I'm also not so fond of Rmd for that purpose, because it feels too slow. I don't want to re-compile the output just for changing some letters. The code too easily clutters up with side-lines and trail-and-error-code from earlier attempts that I (and others) fail to remove afterwards. A second editor accessing the same workspace would be cool (but then I probably wouldn't use it much, and it would take away screen-space). I have no solution, apart from using different combinations of #%%%%%%%- symbols and forcing myself to better in-line documentation of my code."
[102] "The speed. I've run the Bayesian updates and poker simulation. It contains 10^7 random number generating. It takes several, more than 10, minutes. Comparing the other languages like Python which is said very slow interpreting language, R is really slow. Some benchmarks shows the slowness of R."
[103] "Sometimes it is slow to read in a large dataset. Also data manipulation is more difficult than the SAS data step"
[104] "The slow speed of R and the difficulty in using R as a general programming language to develop products"
[105] "Loops run really slow!"
[106] "Slow"
[107] "Slow"
[108] "The fact that loops are very slow and that many people I work with are not familiar with R."
[109] "It is slow sometimes and freeze my laptop."
[110] "Slow speed. Also I consider the Rd documentation system very creaky. It puts developers in a box. It encourages cryptic explanations. It discourages examples by forcing them to be run when checking packages ('not run' is one of the most cryptic elements for users; only developers realize it's a CRAN-relevant annotation.) It discourages holistic user manuals that actually explain how to use a package in a coherent workflow, along with relevant technical details of algorithms. The vignette system helps but is misnamed, under-used and under-visible. Required documentation and vignettes have been great features; they're creaky only due to R's success. Static code analysis by CRAN checking is too invasive and attempts to be too smart. Drop Solaris or explain to the community who still uses it."
[111] "Slow and memory hungry."
[112] "Code can be quite slow to run"
[113] "Slowish"
[114] "R is sometimes slow and is not that easy to work with large datasets as R keeps everything in memory."
[115] "Very slow or hangs on large data sets"
[116] "Slow, not really a language."
[117] "Speed is too slow comparing with other languages, especially when running loops."
[118] "It's a bit slow and sometimes slightly annoying."
[119] "Slow graphics when working w large data sets Slow interactive plotting / frustrating when data sets are large as needs replotting"
[120] "Slow performance when doing multi level or spatial models. Using BUGS platforms is sometimes frustrating"

[121] "Slow!"
[122] "Slow for loops, inconsistent syntax in the base core functions."
[123] "The worst aspect of R is layers of legacy functionality, kept for backward compatibility. Poorly written documentation. Slow loops. If used improperly, memory copies in loops.do.call() can use up all memory and grind to a halt."
[124] "Slow in certain use cases, very messy function call syntax, dislike for tidyverse syntax (I know I'm a minority here!)"
[125] "Occasional irregular syntax. Some external packages can be slow, so simulations can take 1000s of cpu hours, 100s of clock hours via doParallel."
[126] "It is not a general programming language: non-statistical tasks are cumbersome. It is difficult to define complex data structures (and S4 classes make things worse). Non-trivial algorithms (requiring loops or recursion) are difficult to write and very slow. Vectorizing computations sometimes leads to cryptic code."
[127] "Data structures are odd. Documentation is patchy. Syntax is non-uniform. Quite a lot of functional redundancy - some fast, other cumbersome and slow."
[128] "Slow for large calculations."
[129] "Sometimes slow - especially with large genomics data - some multivariate model selection tools."
[130] "Dealing huge database (due to in-memory analysis) slow vague and confusing coding grammar"
[131] "Slow"
[132] "Slow"
[133] "Sometimes it gets slow"
[134] "Quite slow code. Too much actualizations that sometimes make older code not runnable."
[135] "Slow loop interaction"
[136] "Base R is often slow and memory management us not great"
[137] "A times when you have a lot of data the processing is a bit slow, at times tending to crush forcing one to restart R."
[138] "Way slower than classic programming languages (c++, java)"
[139] "Slow, hard to productionize"
[140] "Sometimes it is slow or uses too much memory."
[141] "Working in government that tends to move considerably slower than “ANY” pace, never mind the pace of technology - the updates and resulting variances on code can become quite frustrating. It changes so fast."
[142] "Somewhat slow (single threaded). Obscure code."
[143] "Tends to be slow when working with large data or performing many operations."
[144] "Slowness. Lack of capacity for big data. Lack of easy interface with big data tech. No/little parallelization."
[145] "Choice overload, can be slow."
[146] "It’s slow"
[147] "Slow on big data set"
[148] "It can be really slow when dealing with a big data set"
[149] "Slow and not (very) efficient with large data sets"
[150] "Sometimes obvious things do not work and are very slow"
[151] "It not able to work with data larger than memory transparently for the user. Only some trivial functions are implemented in some very esoteric and difficult to use packages. Working with loops is very slow, you need to vectorize your code, and sometimes it's a problem. It doesn't have a good interactive spreadsheet-like GUI. It's difficult to find a function among so many packages, each working in a different way and many times incompatible. It lacks of a good interactive plots designer. And many other things. Now I'm considering moving to other tools such as Tableau, Julia or Spark."
[152] "Since most of the algorithms work in a single data set in a non-distributed fashion, the inherent in-memory nature can be extremely slow with large datasets ."
[153] "The wrong things in the base lib are still slow and excessively complex. `readr` contains many examples of functions that should probably be pulled into base R during the next major version release."
[154] "Can be slow for some operations and large data. "
[155] "It runs slowly and can crash unexpectedly."
[156] "Sometimes slow, code is ugly."
[157] "Sometimes slow. A decent GUI we can use at work would be helpful (RStudio licensing is not acceptable for a commercial environment and we are very cost constrained so can't buy it). "
[158] "In-memory data makes process of massive datasets slow in the standard hardware that we use in emerging countries."
[159] "Slow, slow"
[160] "It can be slow. There also some counter-intuitive aspects to the programming/syntax."

[161] "Slow unless you know what you're doing. Recoding things from loops (which make a lot more sense when developing code) to make stuff faster sucks."
[162] "Occasionally slow in reading data, some defunct packages, occasional bugs in RStudio"
[163] "Sometimes it can be slow if I'm working with a large dataset. "
[164] "It's a bit slow, which is why I use it mostly for statistics and visualization of results, produced by the image processing and intense data analysis performed in C++."
[165] "Slow computations and steep learning curve"
[166] "Writing efficient R code is fairly hard. Simple stuff is painfully slow."
[167] "Slow machine learning. Switch to python for some of those tasks"
[168] "It is slow, and often does not work well with large data sets. Also the packages are often out of date or not fully debugged."
[169] "The slow speed of “for loops” and not being able to create stand-along applications. "
[170] "Slow. Not so easy to use or learn"

Bug

[1] "Thinking I have found a bug in a package, and not being able to pull out a minimal reproducible example - because I cannot, the bug does not exist. To be fair, this is not particular to R."
[2] "RStudio has been very buggy of late. Not strictly an R problem but an inescapable problem working with R."
[3] "Hard to trace bugs, often caused by poor design decisions - like drop = TRUE as a default, so that the structure of the output of subscripting can vary."
[4] "Managing package dependencies and back-ups. It’s hard to make a project fully reproducible because of difficulties managing package versions. Microsoft’s checkpoint is unsatisfying because it means we have to rely on Microsoft (eww). Packrat is very slow and buggy."
[5] "In some cases parts of the community are not so supportive and seem to take the view that you should just know what you're doing and if you don't, bugger off"
[6] "Some of the gotcha's like StringsAsFactors and drop can be very hard to debug. Documentation for 5 functions in one help, with almost no useful information, and the lack of readme's or vignettes for many packages to get an overview of basic functionality"
[7] "Debugging in R is a mess. Syntax is sometimes absurd."
[8] "Bugs"
[9] "R is painfully easy to break, with processes hanging frequently without a reason provided. We need better error debugging/management than just tryCatch."
[10] "Somewhat of a high barrier of entry and debugging is still harder than it should be"
[11] "Onboarding new users coming from other languages, debugging and dynamic typing, also speed can be cumbersome"
[12] "Bugs in packages"
[13] "Debugging"
[14] "Debugging seems like being back in the 80s, and a real rock-solid IDE with Intellisense would be awesome (RStudio is nice, but still far behind other languages)"
[15] "I find debugging in RStudio unpleasant."
[16] "More than other languages, I feel like I have to run & debug to have any sense of whether my code will work (i.e., very limited intellisense, little “compile time” feedback). IDEs are very rudimentary compared to other environments. It feels like I'm working 10 years in the past. I say that as someone who's been a software developer for 20+ years."
[17] "Some package design so complicated, eval stack is dozen or more levels high on errors, making debugging difficult. Several packages do this; most well known is tidyverse packages"
[18] "I think I still haven't grokked all the apply functions (sapply, dapply, lapply, etc.) Any time i find myself wanting to do something to every element of a vector, it gets hard. Sometimes it's painful that tibbles are *almost* data.tables, but then i deal with a nasty bug where it worked on my toy data.table example, and then when i load a data set it no longer works. I'd almost prefer there to be only tibbles, except sometimes i need to use rownames to then convert a data.table to a matrix for making heatmaps."
[19] "Random bugs that make R feel like an old language. Annoying character encodings, unhelpful errors, weirdness with booleans and NAs..."
[20] "Having to get packages security-checked before they can be used on our desktops, and the security/privacy blocks from our corporate policy that means we can't use it in the cloud with our sensitive data. Also wish debugging was easier, and the learning curve easier."
[21] "IDE: hard debugging, function definition changes (with every change we should run it again to update the definition), error handlings"

[22] "The worst aspect of R is that sometimes error reporting can be poor and debugging can be difficult at times (although I'm aware that a lot of progress has been made in this area)."
[23] "Packages take time to be updated, bugs can occur when new R version comes out and packages are not updated. I remember last year, ggplot became buggy after an R update, I had to revert back to an older version of R."
[24] "Debugging. It is hard."
[25] "There are a couple of things that bug me. - too many ways of doing the same thing, and little guidance to indicate which is best- sometimes I struggle to get the apply/mapply methods to work for me, I still think in for loops I guess- too many object specifications"
[26] "The quality of some packages as well as the redundancy of functions/packages. Even heavily downloaded packages such as MSBVAR contains serious bugs that drive the results and nobody talks about it or fixes them. The fact that i spend an hour the find out with of the five routines that do essentially the same just to figure out which one is the best is really a nuisance. Some sort of referee process for packages could help. Some functions of some packages are could be written with a more general focus, i.e. some very useful functions are only functioning within the package but hard to use with output from other packages, i.e. the vars package. This in turn also leads to redundancy."
[27] "Not so much a problem with R, but debugging R code in an Rmarkdown document has not been easy to figure out and can be a pain in the back-end (could be some tricks I still don't know about)."
[28] "Using R in production is not the easiest. Do-able but requires some good software development expertise, which might be hard to find from people with mathematical/statistical backgrounds. Also, single threaded by default and limitations of memory can be a bug bear. Comments apply to the open source version on CRAN."
[29] "Lack of standardization. There have been a number of times where a common function is re-written in a newer package that changes the interface with no way of knowing. (e.g. lag() in stats and dpylr) this has wasted many hours debugging strange errors in formerly stable code."
[30] "Some bugs, a little slow sometimes..."
[31] "Lack of paid support; bugginess of package interactions; and the lack of built in tools for package management"
[32] "Debugging."
[33] "Odd bugs, and often language conventions that differ from the ones I previously used (dots in variables, 1-indexing...)"
[34] "Debugging and debugging integration (e.g. breakpoints, conditional breakpoints etc. - standard in other programming languages)"
[35] "Working in memory for bug dataset issues"
[36] "I think the debugging tools of R are not well known, and I don't know if they are very usable. I think I tried to use them a few times and they did not stick. Another big concern for me - and this is controversial I think - is that I think the influence of RStudio is not great for the language. There are many great things that people can do with the R language that require the use of RStudio, and the RStudio team has so much control over some of the most popular packages in R that I worry that the language itself might slowly become inseparable from RStudio or weaker without it."
[37] "Bugs from RStudio"
[38] "Dealing with odd errors or bugs that don't have helpful error messages."
[39] "Sometimes unexpected behavior without a warning which makes debugging more time-consuming than necessary. Default behavior should throw more warnings, e.g. with vector recycling or hist(1:4). In general also error messages are cryptic for beginners"
[40] "Clunky interface to most graphics packages, lack of interactive graphics facility by default, too many minor bugs in RStudio."
[41] "Poor documentation. Some arcane language features that limit performance. Chaotic package environment. Debugging somewhat difficult"
[42] "Some of the R packages are buggy, and those that are not under such active development can fall into compatibility problems with the later releases of R."
[43] "Some bugs, like need to restart R from time to time, problems with spaces of variable names, with class of variables..."
[44] "The language, the chaos ... but the debugger could be better as well"
[45] "Uneven documentation and new packages that don't have a lot of users and debugging"
[46] "Bugging"
[47] "Crippled developer environment (svn instead of git, who uses bugzilla that you cannot self register?)"
[48] "It could be a bit buggy at times."
[49] "Incompatibilities between some packages and R version and other bugs that delay the work."
[50] "Maybe debugging."

[51] "Slow performance when doing multi level or spatial models. Using BUGS platforms is sometimes frustrating"
[52] "Data type conversions not systematically automatic or not, not well predictable, requiring much error handling / checking, different by package, hard to debug"
[53] "Debugging errors sometimes is a problem"
[54] "Sometimes performance, code debugging"
[55] "Debugging complex code is still a bit of mystery; especially setting watchpoints. E.g., debug stop when the variable `i` takes the value of 10. Although there are some project outline structures, e.g., ProjectTemplate and R Package, people seem to organize their code without any consistency."
[56] "- Debugging- Automatic installation of own R-based solutions on client computers together with the required R runtime and packages- no easy GUI development of non-HTML GUIs (for “fat clients” as stand-alone applications without using an application serv"
[57] "?????????????IDE???debug??"
[58] "???bug????"
[59] "Debugging. Sometimes, I'll need to step through lines in a package or compiled Java or C library. Sometimes, code runs in one environment but not in another. Sometimes I get wacky memory errors, freezes but no usable info."
[60] "???????????python??RCpp?C++????debug??"
[61] "Time consuming when debugging. Package support of R version. Updating R."
[62] "Occasionally slow in reading data, some defunct packages, occasional bugs in RStudio"
[63] "Steep learning curve, difficult base graphics customization, lack of a centralized resource to submit package bug fixes and new features"
[64] "Shiny applications are difficult to debug."
[65] "writing/debugging code"
[66] "It is slow, and often does not work well with large data sets. Also the packages are often out of date or not fully debugged."
[67] "Debugging code. "
[68] "Rjava configuration / bugs, lots of ugliness from S is still around (e.g. underscores are an operator) and R is very conservative about cleaning it up. I wish R 4.0 cleaned up much of it."
[69] "Syntax can be inconsistent, and so I need to do a lot of cutting and pasting. Certain commands produce strange output (e.g., bwplot in mosaic package). Rarely, my students have bugs that are hard for me to fix. The need to use dataframe$variable to denote variables in many commands in base R is annoying, and attach/detach is no better. "
[70] "debugging"

Install

[1] "At Work it's a nightmare to install and maintain the packages without administrator rights. Generally, the IT department cannot help a lot."
[2] "the forced updating of packages when installing a new packages -> this is sometimes an unwanted side effect in a robust, production environment"
[3] "Utterly inconstant language. Function names and arguments in R are an inconsistent mess: row.names, rownamesrownames, rowMeans, rowSums, rowsumbrowseURL, contrib.url, fixup.package.URLspackage.contents, packageStatusmahalanobis, TukeyHSDgetMethod, getS3methodread.csv and write.csvload and savereadRDS and saveRDSSys.time, system.timecumsum, colSumsFunction agruments are an inconsistent mess:aggregate(.., FUN=mean, ...)reorder(..., FUN=mean, ...)reorder.dendrogram(..., agglo.FUN=mean)reshape2::acast(..., fun.aggregate=mean, ...)promptData(, filename= ,)write.table(, file= ,)# Text rotation argument is las. Or maybe srt. Or maybe rot. Why cannot library(pkg) install packages on the fly? Latex can."
[4] "Definitely deploying R. In my company we have a R package with 10 dependencies. But in the end the total dependency tree have 75 packages. It takes so long to install. In addition, there isn't a easy way to install packages in the required versions. Packages always install the latest version of their dependencies. Packrat is amazing, but it's very memory expensive, since it makes a full copy of R to the repository. Pre built packages with easy version specification would be amazing!!"
[5] "The complexity. installing the packages I have some problems when installing certain packages like go RQDA which I felt was but it would be very helpful for a qualitative data analysis. To this day I'm still struggling to install it and i still have problems in using it. And so sometimes i get the packages worked perfectly from the first time sometimes it causes a lot of problems."
[6] "Installing packages and dependencies"

[7] "One thing that is still kind of annoying is keeping up with versioning. Packages and R are constantly updating and I find it tough to try to keep everything up to date and constantly am getting 'this was built under old R version X!' type warnings. If there were a way to make this more automated it would be great. I just always want the latest versions of everything. If it breaks my code, it breaks my code and I fix it. Also I always am finding I have to reinstall my packages when I update R which sucks. I'm sure there is a better way to do it that would avoid these problems but I want to not think about these things."
[8] "Package management. Conda makes it a lot easier for packages they distribute, and it's good enough installing CRAN packages in a Conda environment; haven't had many difficulties since adopting it. But I haven't had good success with packrat (admittedly some time ago). IMO npm sets the bar for per-project package management, and it would be nice to have something that good for R."
[9] "Not easy to update installed packages when upgrading R version."
[10] "Running and installing the packages!"
[11] "Inconsistent interfaces. Having to install lots of other tools to use packages (latex, particular versions of Java, etc.) Having to reformat output in Word and Excel to make it look professional before sharing it with others."
[12] "Newcomers have a hard time learning the basic idioms of R programming. Whether or not they have experience with other programming languages just changes what misconceptions they'll start with. Similarly, because the packages provided in the basic installation of R encourage different styles (object-oriented vs functional, tree vs tabular structures), there's often very different semantics among user extensions."
[13] "Ramping up is a bit tough, also dealing with package dependencies which can take forever to download, if there are a lot of them. I remember spending over 2 hours downloading text-mining related libraries, which was just painful to watch. Also - is it possible to set it up to uninstall libraries and the dependencies? Right now it's a real pain to do this, and it would be very helpful to get rid of unnecessary libraries/packages that aren't needed - I don't like having more turds in my system."
[14] "Installing on Linux OS is a nightmare specially in a corporate OS image where it has outdated OD packages and integrating with the new R versions with updated packages which requires newer OS version is a pain."
[15] "The learning curve for new packages or new techniques can be a bit steep but it's not especially bad. Sometimes also my work computer doesn't like R installing packages or updates without admin rights, so I have to do my analysis without any packages other than the base installation (or packages installed in a temp directory)"
[16] "too many changes to the base and R studio. If we install new version the old versions create issues. The packages need to sync when newer base versions are announced."
[17] "The process of installing a new version"
[18] "Whenever we upgrade a new version of R, need more information on how to retain the existing packages and move to the upgraded version without any problems. Currently whenever I upgrade R version, I have to reinstall all the packages that I use in my current work"
[19] "I would like to be able to upgrade to a new release more seamlessly, i.e., the if preferred the installer should just migrate the whole environment, all installed packages included to the new release. So, yes, the current release migration is a bit of a pain, and I tend to put it off too long, because of that."
[20] "This is a tough one. I honestly can't think of something that I dislike about R. Sometimes getting packages to install on (other people's) Windows machines can be a problem. But I use Linux and don't have issues myself."
[21] "Installing packages that are not on CRAN."
[22] "The help and specifically the examples in the help seem very complicated, more so than the task at hand. Some inexplicable error messages on installation of packages."
[23] "Panel Data (Time series - cross section) analysis is still not well covered in R. The only reason why Stata is still installed on my machine."
[24] "I had to install 4 versions of R because some packages work only with some versions. And the IBM SPSS also requires another version of R."
[25] "Having to install my regular packages each time I start a session in R. My regular packages are ggplot2 & dplyr."
[26] "Total reproducibility is difficult due to many versions of packages and R itself. I have had code that worked a year ago break due to package updates, or package incompatibilities with the latest version of R. I have several versions of R installed and often have to switch between them in order to get code to run."
[27] "- Debugging- Automatic installation of own R-based solutions on client computers together with the required R runtime and packages- no easy GUI development of non-HTML GUIs (for “fat clients” as stand-alone applications without using an application serv"
[28] "Too many choices sometimes and it is hard to evaluate which is the best option for a given situation. Bioconductor changes so often that it is near impossible to have reproducible research. Bioconductor also

makes it difficult to install packages at an institutional level and has lots of interdependencies so it takes more effort to get everything installed properly. "
[29] "It should have more packages included in the base distribution. The big ones, like ggplot2, plyr, reshape. Every time I update R, I have to jump through hoops to use the packages I used in the previous version. Moving \\library folders is hardly a convenient way. The installer (at least for OS X and Windows) should be more intelligent. "

Big data

[1] "Escalabilidad real a grandes volmenes de informaci?n y conectividad con clusters big data limitada (aunque en un crecimiento constante)"
[2] "Loading big data and crashing R studio"
[3] "Deal big data"
[4] "Can be slow. Holds too much in memory to munge big data without taking extra steps to reduce data set as much as possible. For loops also take a really long time to run."
[5] "Reading large datasets into memory. My skills aren't sophisticated enough yet to handle “big data” so large datasets can sometimes be a problem when memory space is at a premium."
[6] "Loops, not easy to manipulate big data"
[7] "It gets slow with big datasets."
[8] "It's a bit slow with big data"
[9] "trying to understand how best to work with big data."
[10] "Dealing with big data, memory issue, and not super great ML algorithms."
[11] "Working with dbs and big data sets -lack of parallel version of functions or algos"
[12] "Working with big data sets and GUI for data manipulation / cleaning."
[13] "Big data used to be challenging, but it's getting better now."
[14] "Speed, big data - it's designed for a laptop, not a multiCPU server."
[15] "Slow performance, ancient kernel (bad Big Data capability, RAM-limited, no native multicore)"
[16] "Currently not mature enough wrt big data, deep learning, and productionization of ML models"
[17] "Cannot handle large amount of data or big data!"
[18] "The memory required to run big data. For Kaggle competition sometimes I had problem to upload 1 million of dataset rows. Time lost to run machine Learning. There are others platform performing better and this is a pity."
[19] "Problems handling big data."
[20] "Needs more functionality with big data"
[21] "Poor capacity of dealing with Big Data (bigger than your RAM)"
[22] "Slowness. Lack of capacity for big data. Lack of easy interface with big data tech. No/little parallelization."
[23] "Have to google everything and rely on stack overflow, don't feel like the syntax is easy to remember replicate (even for tidyverse, though that's certainly better), has the problem of any programming language that it's reliant on me to fix typos, etc. Would love for R to be a bit more like OpenRefine, where there are workflow options that are coded into a GUI and that can be reapplied easily on new datasets. I'm working on a big data analysis project, and every time I realize there's something else I want to capture, I'm stuck going back through all of my code, manually changing variable and file names, etc. Sure, I can right my own functions, and I'm working up to that, but I wish there was a better development environment for beginners that produced the “best practice” code, more like Stata or maybe SPSS. Maybe Exploratory.io is the answer, but I haven't tried it much yet."
[24] "Can't handle really big data."
[25] "using big data technologies"
[26] "slow on big data set"
[27] "support with Big Data Analytics"
[28] "It can be really slow when dealing with a big data set"
[29] "How it manages poorly big datasets. On Linux, if I accidentally go beyond RAM, it effectively blocks my laptop. That's painful. There must be a better way. "
[30] "Working with big data, and sometimes try using the loops. "
[31] "It is not fast enough, especially when dealing with big datasets."
[32] "the performance in big data sets, maintenance"

Production

[1] "Database write support is crumby. Very little support for working in a production environment."

[2] "Occasionally memory issues with larger data sizes. Also how to work models into production systems easily"
[3] "Hard to convince management that R can be a production language instead of just an exploration and experimentation language"
[4] "Since things are being developed so quickly, using R in production can be difficult. Theres the desire to use the best implementation but it could break your code."
[5] "Dependency management in production. Honorable mention: Readability of code can be poor if less experienced users write code. Conventions are not well known. Stability is an issue, R being single threaded limits its web aspirations for production."
[6] "Concerns over production issues with R"
[7] "Sometimes the speed in production is lacking, package maintainability and testing can be a mess, submitting to CRAN can be a black box sometimes"
[8] "Bring R models in Production"
[9] "Managing projects. Packrat, checkpoint, docker, ... everything nice. But I would prefer an out of the box solution (or just something) that is really robust and just works in production without a lot of setup. Also It's very hard to compare Shiny against Tableau, PowerBI etc (when these self-service BI platforms integrate better statistical ie R support, there is somepoint where Shiny might not be valuable in many situations)."
[10] "Would love to see it evolve to more production ready general purpose language like python."
[11] "Not easy to get in production"
[12] "the forced updating of packages when installing a new packages -> this is sometimes an unwanted side effect in a robust, production environment"
[13] "Some of the old things that are hard to teach others- i.e. functions only returning the last argument so people can be sloppy with their coding. As well, I am not really sure on how to schedule batch analysis to happen regularly or to run non-shiny production analysis"
[14] "Breaking changes from critical packages breaking other packages downstream, especially for code that is in production. It would be nice if there was a method load packages by version (I know, I know: that's what packrat is for; I just need to teach myself how to integrate that into our pipeline)"
[15] "Looks more academic than professional, so if you are looking for production solution I doubt how much can be done"
[16] "Not so easy to incorporate in a production environment where speed is a crucial aspect."
[17] "Production, data input"
[18] "Data import from non-CSV files is an issue. Poor documentation of basically most if not all packages. And poorly formatted output of important package functions (see previous comments about party package implementation of random forest). And difficult interface with other languages. R is unfortunately not a production language, and would need to better interface with, say Python, which does not have as nice visualizations."
[19] "Production code and general programming"
[20] "It doesn't feel connected to the production tech stack anywhere I work"
[21] "Production capacity"
[22] "Managing dependencies for production has been annoying. Partly due to the immaturity of certain some key packages in the ecosystem."
[23] "Not as easy to deploy/productionise as python."
[24] "Oh my god please make it so R projects embed packages in them and updating the packages of one project doesn't break another. Pack rat needs to be built into R! Also there are tons of things like better error handling and edge cases with things like 0 row vectors that make R difficult to use in production and should be fixable."
[25] "Not easy to productionize or parallelise easily as python."
[26] "I would like a simple way to productionize models using a serverless environment."
[27] "R is not as strongly object oriented as Python and doesn't scale up in a way that allows it to be used in production on large scale data sets (e.g. pyspark)."
[28] "Speed and scalability is an issue with production code and the syntax can be a bit strange."
[29] "Anything around productionizing methods and code for use in larger scale software systems."
[30] "Adoption and support among engineering community is very poor. Designing algorithms in R, testing and verifying only to have to then work with engineering to rework the code into java for production gets frustrating."
[31] "It is still challenging to integrate into production websites."
[32] "Using it in production is a headache - requires a lot of thought."
[33] "There isn't enough strictness with data types and that's a huge problem for production. Not everything should be a data frame. But lists and vectors are awkward without list and dict comprehension and unpacking."
[34] "R struggles to be productionalized. Not as many people creating efficient modern learning methods"

[35] "Evangelizing. Convincing that it is production ready."
[36] "Bringing things into production"
[37] "Using R in production is not the easiest. Do-able but requires some good software development expertise which might be hard to find from people with mathematical/statistical backgrounds. Also, single threaded by default and limitations of memory can be a bug bear. Comments apply to the open source version on CRAN."
[38] "Not production-ready yet."
[39] "Productionizing R models is not trivial for real-time applications. Preferred approach is usually to port to another language"
[40] "Current difficulties on getting into production - lack of alternatives/competition for e.g. Shinyapps.io or the pricey Shiny Server Pro"
[41] "It's not so easy to build a production / industrial environment"
[42] "Hard to get code into production. No “easy compiler”, you need to use other solutions such as shiny / RInside / R gem for ruby / server with script scheduling / other solutions."
[43] "It requires a lot of time to master. It is very complicated to write optimised code. It's a nightmare to use in production (the first shot almost never works)"
[44] "Not generally accepted in industry to use to put code in production"
[45] "More support for use as production code."
[46] "It can become very slow with large data. A lot of this has to do with the servers we are running the code on. Package versions are constantly changing and our company doesn't have infrastructure to appropriately handle a production and development R environment. Difficulties writing production R code that the company can ingest."
[47] "Production and large computations."
[48] "Heavy load of productions designs are not that well supported (or I just not yet familiar with better practices)."
[49] "Slow at times and memory-limitation. No tools available to write production grade code, to my knowledge"
[50] "The inconsistencies in naming even in base R.I also dislike the use of . in names, since in most other languages that has a special meaning. For production I also think it would be good if R core would work with something like PEPs (from the Python world) and announce up front what will be tackled in the next release. Similarly, the community should do more to make packages break less often. Package developers should be educated to first deprecate functionality and inform users when the deprecated function will be removed and what do do differently from now on."
[51] "Im not aware of a good way to put code into production."
[52] "Very hard to go from prototype to production platform many cloud environments do not support R so we are often forced to switch to python and it's usually a one way street."
[53] "Putting algorithms into production Integrations of other kind with another languages."
[54] "Graphical tools. It requires just too much effort and knowledge as to be used in production/business environments. It would be great to have a rich UI like Tableau's (or even Excel's), to create visualizations by dragging-n-dropping/WYSIWYG approach, and that generates the necessary code in R under the covers."
[55] "Speed is sometimes an issue with larger datasets or heavier computations. But the larger problem is getting the “solution/model” from R into production."
[56] "Lack of confidence in organizations for use in a production environment"
[57] "Guys, I gotta tell you that the web hookups are not great. Db integration could be better, too. Generally the rule seems to be “prototype in r, production in python.”"
[58] "Understanding how to move it into production in our company."
[59] "Deploying to production. There isn't any best practice or guidelines"
[60] "Maybe it would be difficult to learn to use it in production, because not many people use it that way"
[61] "No production environment for machine learning models into production"
[62] "currently not mature enough wrt big data, deep learning, and productionization of ML models"
[63] "It is not that friendly for production of other than data analysis"
[64] "Using it in production environments"
[65] "Production! I can't stress that enough. while I do use R in my work environment, its always for internal use, I worked for (and currently employed for) major leading international companies, and yet whenever I suggest to do something that should be implemented in production the decision is to use different tools. I even sometimes write an entire project in R, and than 're-factor' it to python. I didn't had the chance to work with it on the new 'sql server 2016'.I suspect that R is not common enough in CS academy & community and thus it is not adopted later on when they make decisions concerning technologies to be used in their environment. Also, CV domain is not developed enough in R (in my opinion), and latest academy courses I participated in required either python or matlab which is a shame. I suggest to try and have it widely adopted among math & CS community."

[66] "It is a somewhat hard language to learn, specially for beginners. Even if you are already familiar with it, it is sometimes difficult to convince someone to work with you using R. The other main drawback IMO is its performance. I do not feel safe to deploy it on production."
[67] "Sometimes speed. Also convincing other people it can work in production."
[68] "Hard to deploy to production. People always trying to get me to use python"
[69] "Currently unable to scale for production"
[70] "Putting it into production"
[71] "I've not ventured into using R for production -- our SaaS application is in Ruby and often I've wanted to take ideas to a production-level to lower the barriers to productizing but I haven't seen much in using R well within production where it is a secondary or supplementary language"
[72] "naming conventions, R into production"
[73] "Language doesn't follow many of conventions (dollar sign, matrix starting with 1). Error handling in functions and packages for production harder to implement. Often difficult to determine the optimal package to use when many implement the same functions."
[74] "handling large datasets or as a commercial grade production code"
[75] "as i learnt it by myself, or some specific courses; im not quite aware of the production workflow environment ; how to structure a proper project, good practices on managing dependencies, or chaining really long works. will be really cool to have some tutorials on that way , as everything focus on the " quick get by data " ; and for more " big/ real" stuff you need order, not just a one lienr solve it all for me."
[76] "speed and productionalising the code"
[77] "Bigger files years ago when ram memory wasn't so cheap... For production keeping a server updated could be painful. Sometimes you need to stick to a version and then look for compatible packages and compatible dependencies."
[78] "- Licenses! GPL2 and GPL3 produce problems, limitations and additional effort when putting R Code into production"
[79] "At the moment, the worst aspects are the speed and the lack of tools for image analysis. R has also some problems when it comes to production even if it is improving ."
[80] "slow, hard to productionize"
[81] "The language has been split into those who use the Hadleyverse and those who don't, when the syntax is so different it literally prevents R programmers working together on the same project. It's quite deeply concerning that R has put higher value from upon people getting results quickly at the cost of bad coding practices being adopted. CRAN is now saturated with packages that do exactly the same thing. I fear that R has to answer how it will help it's developers build production level coding standards and stop focusing on short term syntactical sugar to woo new users. Failing to address some of these significant problems will hurt the longevity of the language."
[82] "It is not easy to find which is the "right" way of doing a task. By "right", I mean the best standard way and some hint to more specific tuning and more Advanced used. Moreover, sometimes, performance could be a drawback. Also, R is very permissive and writing working script is easy even if it is kind of dirty. Writing code with good practice, and standard way is not so easy (comparing to python syntax for example, or module). it is a force that user can get to their aim very easily but it is painful when you have to put this code into production or to share this code. (comments with a company insider point of view)"
[83] "The more complicated the routine, the more evident it becomes that I am trying to tie together the reasoning of hundreds of different people, each with their own ideas of what makes a good function interface, or good syntax, or realistic practice. The beautiful communal aspect of the language, the broad contributions from highly diverse sources with particular expertise and needs, is by some turns a miracle but by others an anthropological nightmare. It's the zoo-like quality that for me limits R to work outside of production. Somewhat related: it's also very difficult to proof code well. Error messaging is often inscrutable, a problem amplified by how frequently package updates seem to introduce breaking changes."
[84] "To little focus on production quality"
[85] "It not being considered "production ready" code."
[86] "Convincing devs to let you use it in production "
[87] "it's good for analysis and testing, but less so for overall production and very large datasets. And while the vast availability of packages may be an advantage, the overall heterogeneity, "continuity" and maturity of them may be a disadvantage for long term development aspects (less so for ad hoc analysis)."
[88] "Difficulty to take a prototype and implement it in production"
[89] "Help files semi-useful to dive into new topics, vignettes sometimes lacking depth, missing naming conventions, difficult to fit into a production landscape"

Question 31: Supporting statements

Roadmap

[1] "Get the R foundation to publish a roadmap. Support the speed up of R (manage more data faster). Improve activities in areas where R is loosing it, compared e.g. to python:. NLP (e.g. poor / slow implementation of word2vec). Deep learning. Reinforcement Learning"

Commercial

[1] "Keep driving commercial support of R and integration with other technology so that it stays relevant in industry, and keep investing in teaching and learning resources (e.g. conferences, online webinars, learning platforms like Swirl, DataCamp, Coursera, etc.) so that R users can keep up to date with R packages & developments."
[2] "Work toward easier integration with commercial analytics, especially SAS and Tableau, and relational databases."
[3] "I would like to see more investment in improving R IDE's and environments. Although I also understand that is tricky given that the two best options are both commercial products to one degree or another. Improvements in R tooling for popular text editors (vs code, atom, sublime) would help."
[4] "make R more commercial friendly"
[5] "Make R an option for commercial projects of big players by providing suitable licensing"

Enterprise

[1] "I work in a public hospital and am a bit of a lone ranger when it comes to my use of R in my organization. I would love to be able to implement an enterprise solution that could replace our reliance on sharing data via Excel spreadsheets and PowerPoint slides but I lack the technical know-how to scale my skills. My hospital is a Windows shop I have found it impossible to get buy in from our IT department to see the benefits of setting up an R environment that can be accessed by users across the organization. Simplifying enterprise level implementation that is cost effective for budget conscience public health facilities would be extremely well received."
[2] "Continue to ensure that R is seen as being enterprise compatible, without intervening in its open-source culture."

Error

[1] "Improving error messages."
[2] "Helping to build the language to be a bit more thorough. I think starting with making the language hosted on GitHub would help, so others can view all the Bugs in R, and try to help build it to make Base R better. Currently we cannot, which is frustrating to look at BugZilla, and see all the errors that exist that we can't really interact with."
[3] "Helping to facilitate projects to make the R Core more reasonable. The two suggestions I have are having a project to improve the documentation of old functions especially commonly used ones. These are often written in insider language ("a wrapper for noobs") or to explain the problem code was written to solve, but not how it is actually used by other people. Terms used are not defined, short hand link () to execute is used, which beginners may not know what is being done if they haven't seen that before. I've often wondered what would happen if I tried to submit updates. Also a project on error catching would be great. Lack of catching (and in general lack of checking returns) is a huge problem in R packages."

Package

[1] "Solving package dependency, package versioning, project reproducibility, dependencies on packages."
[2] "Keep driving commercial support of R and integration with other technology so that it stays relevant in industry, and keep investing in teaching and learning resources (e.g. conferences, online webinars, learning platforms like Swirl, DataCamp, Coursera etc) so that R users can keep up to date with R packages & developments."
[3] "I think there should be a service to help people with packages best practices. R Open Sci has something like this."
[4] "I would like to see more unification of R packages and their syntax. For instance, bioconductor vs cran, tidyverse vs data.table."

[5] "Something similar to ropensci where they have a steady outreach of how to use their packages together. In general, R Consortium can have a way to be more opinionated about what is the “correct” R way to do things."
[6] "Helping more people get started with package development."
[7] "Helping to improve some package documentation would be useful"
[8] "I would like to see more support for people doing econometrics with R. But I understand that this is not the main objective for the R consortium but rather the R package contributors."
[9] "I would the Consortium to found Duncan Temple Lang RLLVM package, it will make R way faster (like Julia). Generally, it is great that they fund great project and build a coherent R ecosystem."
[10] "Improve R package to manipulate large dataset and increase R machine learning strength"
[11] "Develop a learning/teaching package to learn R in R."
[12] "I like that rOpenSci makes lots of packages to access obscure data sources. Maybe R Consortium could find that sort of thing too."
[13] "Continue to promote quality standards, prevent too much overlap and duplication of efforts across packages."
[14] "Develop a feature voting system, including for packages."
[15] "Prioritizing packages that get to CRAN"
[16] "more house cleaning of out of date packages and homogenizing the R code grammars"
[17] "Lobbying to move R forward. Please, I don't mean more unvetted packages on CRAN. I mean base R."
[18] "I believe it is doing a good job already. Providing new packages gives a lot more visibility than improving the existing ones, and fortunately I know you already do help to improve existing packages (e.g. with DBI). Continue improving what already exists as well."
[19] "Helping to facilitate projects to make the R Core more reasonable. The two suggestions I have are having a project to improve the documentation of old functions especially commonly used ones. These are often written in insider language (“a wrapper for noobs”) or to explain the problem code was written to solve, but not how it is actually used by other people. Terms used are not defined, short hand link () to execute is used, which beginners may not know what is being done if they haven't seen that before. I've often wondered what would happen if I tried to submit updates. Also a project on error catching would be great. Lack of catching (and in general lack of checking returns) is a huge problem in R packages."
[20] "it's good like that, fund more package maybe"
[21] "Make sure that relatively much used packages will have an author / maintainer after the current author quits working on it."
[22] "Just investing in high quality packages and R development to ensure that R thrives into the future and continues to adapt to new technologies"
[23] "Focus on development of good quality software/packages."
[24] "Funding package maintainers to improve packages in areas where R's relevance is slipping, such as machine learning."
[25] "I think that the R Consortium is potentially the only body that can support “meta” package efforts to make the R ecosystem more coherent, because this requires real resources. One of R's biggest problems is highly inconsistent APIs and poorly defined/overlapping package scope. Supporting efforts to make the package ecosystem more holistic, well-defined, and consistent would be a worthy goal, and certainly make my life easier."
[26] "Provide a user-friendly way to search CRAN packages by topic. The task views are good, but only cover a few themes."
[27] "Funding R and packages"
[28] "Working towards consistency, simplification, consolidation, and possibly quality control of packages and particularly package documentation (by supporting CRAN)."
[29] "Taking care of language issues (shiny, decoding), timezone problems, integrate data.table into the base packages. Better documentation for markdown, knit, pandoc templates, parameters and option"
[30] "Creating high quality documentation / discussion about R (How to make a great function, package,) that won't be accepted on SO or elsewhere because it is non objective. Review packages too"
[31] "R Consortium should1. Promote R, at the workplace, at schools, etc. 2. Work with R Core Team to address any gray areas in the language, i.e., to codify R language, in the sense of what the C Standard committee did, not what the C++ Standard Committee did. For example, codify and document how best to write R packages using native languages such as C.3. To further document and clarify the gray areas in the language."
[32] "More substantial funding of important package development"
[33] "Bring package systems together, make versions dependencies easier to manage."
[34] "Keep funding R infrastructure! Documentation, checkpoint/pack rat package style improvements to specifying specific package versions in scripts and projects (this is another major pain point), more on-boarding material that, statistics teaching material, so much could be done!"

[35] "playing a more active role in improving base R, rather than making improvements through packages. improving vanilla R is how you'll reach all users"
[36] "foster package development and education"
[37] "More packages!"
[38] "linux package binaries!!pay meetup fees for R user groups + central website for RUGs including a place to share ideas + experiences start discussion with R foundation + core group whether to unify API (e.g. naming of functions, arguments etc) or even completely rewrite baseR with the knowledge we now have"
[39] "Come up with a solution for package identifiers masking and for handling simultaneous different package versions."
[40] "Better documentation format. The content of the current help files is good enough, the presentation is not. Many attempts, nothing changed - with the exception of vignettes/knitr/markdown support. pkgdown is a good concept, output format should be standard. Work that there are more tests, mainly in standard packages. When 3.4.0 came out, I was flooded by warnings about partial matching in my old reports, partially from standard packages such as RODBC. For this one, I even submitted an almost trivial pull request, https://github.com/cran/RODBC/pulls/dmenne, but nothing happened. At user2017, I spoke about the issue with several people, but nobody wanted to muck with the r-core team for fear of being ripleyed (not by brian, there are other ripleys now)."
[41] "The R Consortium should focus on investing in people and teams who are ready to develop R packages that the community needs the most."
[42] "Supporting more foundational projects like database infrastructure to make R more robust and provide a foundation for other package developers to build on."
[43] "Provide a map to R packages. For example a grouping/tree"
[44] "package management and improving base R"
[45] "Trying to align how packages work a bit more. Not to say a heavy hand should apply here. Maybe packages that meet certain standards get a certified sticker"
[46] "I'll likely approach other workers to see how some streamlining and consolidation of packages might be accomplished and in conjunction with others possibly make a proposal to the Consortium. I'm also looking at knowledge transfer to younger workers, as many of the underlying computational methods are inaccessible to those who haven't worked with the older computational environments, esp. Fortran. That may also be something the Consortium will want to support, though it has application much wider than R."
[47] "- Streamlining of open source licenses used in R packages- Support further development of opencpu"
[48] "Funding more package development."
[49] "Improve package and base function documentation and examples for beginners. Even as an experienced user, the documentation pages (especially examples) are often incomprehensible to me."
[50] "guide to industrial strength packages and verification for industry use"
[51] "funding Projects which help to Scale CRAN-Packages on distributed Platforms (Spark, Hadoop).I know, this is a HUGE effort, but it's well worth it"
[52] "A single recommended/blessed system for package version management would be helpful. "
[53] "Revamping CRAN and the submission process. Some sort of basis peer-review of package documentation. Speeding up and optimizing R so that it can handle larger datasets. A curated documentation for popular packages. Develop packages similar to shiny to create simple GUI."
[54] "Funding expert developing work to improve main R packages, to make them more stable, standardized and so on. Being visionary about R."

CRAN

[1] "Helping to ensure that the development and improvement of R itself is sustainable in the future. I'm concerned that R Core members and CRAN volunteers are being asked to do too much work with no compensation, and that this is unsustainable."
[2] "I would like to see more unification of R packages and their syntax. For instance, bioconductor vs cran, tidyverse vs data.table."
[3] "Getting more serious software engineers and developers interested in joining the ecosystem, however that is interpreted. Creating a better website for CRAN would be a nice start."
[4] "Making clear plans for the future of the Core language (including a potential need for drastic changes if some historical decisions are seen to hinder progress), as well as CRAN, and just as importantly, sharing them with the community."
[5] "Helping fund core R activities (e.g., modernizing the CRAN website, helping manage the bug tracker, maybe more documentation work)."

[6] "Prevent hostile takeover by Microsoft. Interact with CRAN and R-Core to make a even greater R experience."
[7] "Modernize R-Forge and CRAN"
[8] "Prioritizing packages that get to CRAN"
[9] "Modernize CRAN - it's a bit embarrassing to have such an old system compared to NPM, Pypi, Rubygems, etc. IMO, we don't really need humans maintaining submissions."
[10] "Lobbying to move R forward. Please, I don't mean more unvetted packages on CRAN. I mean base R."
[11] "Provide a user-friendly way to search CRAN packages by topic. The task views are good, but only cover a few themes."
[12] "Working towards consistency, simplification, consolidation, and possibly quality control of packages and particularly package documentation (by supporting CRAN)."
[13] "CRAN views and cataloging packages with a more effective search approach,"
[14] "I'm thankful to CRAN , stack overflow, Kaggle and the dozens of really cool sites , free books and other resources people spend their valuable time and effort to produce and distribute, I hope the consortium makes R integration with other products better (for me that would be Oracle, Tableau, OCR software...)"
[15] "Develop more quick commands (like MCRAN does) organize parts of R development and help disseminate into big companies."
[16] "Better documentation format. The content of the current help files is good enough, the presentation is not. Many attempts, nothing changed - with the exception of vignettes/knitr/markdown support. pkgdown is a good concept, output format should be standard. Work that there are more tests, mainly in standard packages. When 3.4.0 came out, I was flooded by warnings about partial matching in my old reports, partially from standard packages such as RODBC. For this one, I even submitted an almost trivial pull request, https://github.com/cran/RODBC/pulls/dmenne, but nothing happened. At user2017, I spoke about the issue with several people, but nobody wanted to muck with the r-core team for fear of being ripleyed (not by brian, there are other ripleys now)."
[17] "Look for more ways to reduce the causes of frustration experienced by the R Core team and the CRAN team. Focus on tasks that will ensure the continued relevance of R and the sustainability of the work that goes into it."
[18] "funding Projects which help to Scale CRAN-Packages on distributed Platforms (Spark, Hadoop).I know, this is a HUGE effort, but it's well worth it"
[19] "Conferencias grandes de R en espaol; Mejor contenido y documentaci?n. Un ejemplo es que los CRAN taskviews son HORRIBLES. Se puede hacer mejor."
[20] "Revamping CRAN and the submission process. Some sort of basis peer-review of package documentation. Speeding up and optimising R so that it can handle larger datasets. A curated documentation for popular packages. Develop packages similar to shiny to create simple GUI."

Training

[1] "Expanding the availability of R training beyond the large urban centres through webinars etc"
[2] "- I think R Consortium should support people to organize more conferences in countries other than US and Europe. I am from India and I want an R conference in my country.- Fund more initiatives such as Software Carpentry R Instructor Training in different regions"
[3] "Promoting R and funding training is economically deprived countries."
[4] "Professional R developer training"
[5] "To promote and consolidate training material. And to promote the research to work with BigData more easily."
[6] "May be more coordination for R training?"
[7] "More free training and working to get more R support by proprietary software, such as complete integration into Tableau. The ability to directly read Excel. With Microsoft being part of the Consortium I find it odd that R doesn't directly read and write to the entire Office suite."
[8] "Supporting user groups and sponsoring more training opportunities"

Education

[1] "I would love to have a more formalized mechanism for support for the R education projects I run."
[2] "I do not want to say what the consortium should or should not be doing, but some sort of grant resources for education would be awesome."
[3] "foster package development and education"

Tutorial

[1] "More online tutorials and examples"
[2] "Provide open-course tutorials (video lectures), like MIT or Stanford."
[3] "As a community, we will be building tutorials together."

Course

[1] "Keep driving commercial support of R and integration with other technology so that it stays relevant in industry, and keep investing in teaching and learning resources (e.g. conferences, online webinars, learning platforms like Swirl, DataCamp, Coursera, etc.) so that R users can keep up to date with R packages & developments."
[2] "Lobby the Pope for consideration of Hadley Wickham for Sainthood (of course, after waving the requirement that he be dead for five years)."
[3] "Provide open-course tutorials (video lectures), like MIT or Stanford."
[4] "Desarrollar su propio curso tipo edX o Coursera. "

Fund

[1] "Funding of projects serves the community as a whole and should continue."
[2] "- I think R Consortium should support people to organize more conferences in countries other than US and Europe. I am from India and I want an R conference in my country.- Fund more initiatives such as Software Carpentry R Instructor Training in differe"
[3] "Funding conferences and meetings to promote the community"
[4] "Helping fund R User Groups"
[5] "I would the Consortium to found Duncan Temple Lang RLLVM package, it will make R way faster (like Julia). Generally, it is great that they fund great project and build a coherent R ecosystem."
[6] "Continue funding great work like sf, r-hub."
[7] "Helping fund core R activities (e.g., modernizing the CRAN website, helping manage the bug tracker, maybe more documentation work)."
[8] "Promoting R and funding training is economically deprived countries."
[9] "Engaging more directly to ask the community what they want to be funded rather than having a small group of people decide what the community wants to be funded. (I guess that's exactly what this question is for though...)"
[10] "I don't know. It's still relatively new. If I've learned anything from 20 years of work with W3C, keeping the major vendors from taking over all of the key decisions would probably be the best thing it can do for the community (but I guess that's the R Foundation's work anyway). Trust me--the vendors, at the end of day, have only their best interests at heart (despite what they say and despite what they might fund in the short-run). And that's coming from a “business professor” (sigh)."
[11] "it's good like that, fund more package maybe"
[12] "Keep on funding useful projects!"
[13] "Funding package maintainers to improve packages in areas where R's relevance is slipping, such as machine learning."
[14] "Keep funding new strategic efforts to progress R for data science."
[15] "I would love to see a project funded for the deployment of R models. I know 100 people that can train a good model in R and 0 people that can put one into “production”. I would love to see this process made easier. Microsoft's R on SQL Server has done a lot (for me and my company) in this regard, but I would love to see an open source framework for deploying R models as APIs, microservices, containers, etc."
[16] "Funding R and packages"
[17] "Passing along funding opportunities that I can share with my meetup."
[18] "More substantial funding of important package development"
[19] "Keep funding R infrastructure! Documentation, checkpoint/pack rat package style improvements to specifying specific package versions in scripts and projects (this is another major pain point), more on-boarding material that, statistics teaching material, so much could be done!"
[20] "- keep on funding projects that close critical gaps in the R ecosystem"
[21] "Funding me :)It's a great idea but must be careful to stay open and not just solidify currently favored companies and approaches."
[22] "Easier ways to get funding to projects"
[23] "The consortium should fund me to enable me attend conferences for R users and get more exposure in the use of R language."

[24] "Could we get more funds (from Microsoft and philanthropists) to support some persons around the world (like myself) . I am a bit older, 4 degrees , experienced in a scientific area and want to teach- How do I get to do that - within universities- they are not open to new ways of teaching yet, and the workload is too high . (I am moving out of a university- as I have more potential to effect the changes that I can see are needed). The university system our crushes our creativity, and does not allow the new learning methods to be integrated. At my university we have awful systems like Blackboard to contend with. "
[25] "Funding more package development."
[26] "Maybe specific support for technology transfer from Academia to Business. For example my research group help companies using Statistical Process Control with R, the consortium could have a funding program for such short applied projects."
[27] "funding Projects which help to Scale CRAN-Packages on distributed Platforms (Spark, Hadoop).I know, this is a HUGE effort, but it's well worth it"
[28] "Get more funding, fund more cool projects. More frequent grant cycles, perhaps"
[29] "Make an effort to remove language inconsistencies (such as list()<-NULL; NA factor level handling;) and fund in-core development of performance features which are addressed by bit, bit64 and ff (lean data types such as bit, INTEGER64 addressing instead of INTEGER32+DOUBLE, out of core data, leveraging sorted caches, move to a JIT compiler; or : merge efforts with Julia)"
[30] "More low level awards to fund workshops for teaching undergraduates. "
[31] "Funding expert developing work to improve main R packages, to make them more stable, standardized and so on. Being visionary about R."

Internationalization

[1] "Try to document the user base. This survey is a great start. Possibly need a unique URI always hosting the currently available (or most recent) survey, so we can always direct people to a single place and build coverage with time. A mechanism to propose questions of general interest (example native language of users + preferred language for documentation, to prioritize internationalization efforts) to include in these surveys (maybe a proposal system)"
[2] "Working seriously on internationalization. As a professor who teaches research methodology I think it's one of the biggest barriers to widespread adoption in non English speaking countries. I would happily volunteer in a project like that, but I don't know how. Make it easy and people will show up to help!"

Community

[1] "Funding of projects serves the community as a whole and should continue."
[2] "Integrate community resources, speed up the optimization of R language, especially clean up some legacy code, and make the command style consistent as much as possible"
[3] "Keep up the good work! Anything that supports community building, particularly for those currently under-represented."
[4] "Having folks outside of companies decide what is good for the community."
[5] "Because of the large corporate sponsors and high visibility, projects to facilitate R-related employment opportunities would likely be very useful to the r community."
[6] "Funding conferences and meetings to promote the community"
[7] "Some guidelines to build and maintain a lively local community"
[8] "Making clear plans for the future of the Core language (including a potential need for drastic changes if some historical decisions are seen to hinder progress), as well as CRAN, and just as importantly, sharing them with the community."
[9] "Continue to support community projects to keep improving R libraries."
[10] "Keep doing what it's doing. Hopefully we are in this for the long haul. Maybe there's a role for a formal community historian."
[11] "They shouldn't do anything for me, but rather do things for R and the broader R community."
[12] "Engaging more directly to ask the community what they want to be funded rather than having a small group of people decide what the community wants to be funded. (I guess that's exactly what this question is for though...)"
[13] "Supporting participation of underrepresented minorities in the community. R-ladies is a great initiative and should be enhanced and continued but there doesn't seem to anywhere near similar support for people of color and other underrepresented groups."
[14] "More active role in global community."
[15] "I don't know. It's still relatively new. If I've learned anything from 20 years of work with W3C, keeping the major vendors from taking over all of the key decisions would probably be the best thing it can do for the

community (but I guess that's the R Foundation's work anyway). Trust me--the vendors, at the end of day, have only their best interests at heart (despite what they say and despite what they might fund in the short-run). And that's coming from a “business professor” (sigh)."
[16] "A web page for our community would be great. Support in organizing events"
[17] "Focus on supporting and expanding R community outside North America and Europe."
[18] "Make more local R events a reality. I live in Riga, Latvia, and there is a very thin R community here (from what I have witnessed so far). I believe that it is far wider, just that we are busy on our jobs and aren't reaching out to each other."
[19] "Push the directions and future standards of the R project. Consolidate the community and spread the word."
[20] "Expand our community, work on making hacks of the tidyverse un-hacky, support our volunteer developers."
[21] "Initiatives towards improving the quality of the R community to make R more accessible to the masses."
[22] "Fostering collaboration and teamwork across the community."
[23] "Feeding R user feedback to development community for future R development"
[24] "Identify the pain points in R and fix them with the community; know better the R users (this survey is a great step into that direction)."
[25] "The R Consortium should focus on investing in people and teams who are ready to develop R packages that the community needs the most."
[26] "As a community, we will be building tutorials together."
[27] "Working on making the R language more useful to the community"
[28] "Being a space for communication and contributions of R community."
[29] "I think it should help people (and group) around the world to share a common way of teaching, promoting and speaking about R in order to more rapidly close the gap between the “old way” and the “new way”. Moreover, it should help promoting R as a valid language (as python) to use in IT department and in production. It is not just a language for prototyping and for research. I think it could help a lot the community."
[30] "steer R language using community feedback"
[31] "I want to be connected with community members without any charge."

Documentation

[1] "Optimizations in other R implementations (Renjin, ...) should flow back into vanilla RSupport efforts to improve documentation"
[2] "Helping to improve some package documentation would be useful"
[3] "Documentation improvements in UX"
[4] "Helping fund core R activities (e.g., modernizing the CRAN website, helping manage the bug tracker, maybe more documentation work)."
[5] "Continuing to invest in the R infrastructure seems great. Perhaps also improving the quality and availability of documentation for common R tasks. Or collecting and endorsing existing high quality material."
[6] "Try to document the user base. This survey is a great start. Possibly need a unique URI always hosting the currently available (or most recent) survey, so we can always direct people to a single place and build coverage with time. A mechanism to propose questions of general interest (example native language of users + preferred language for documentation, to prioritize internationalization efforts) to include in these surveys (maybe a proposal system)"
[7] "I know this is already underway, but improved documentation / ways to contribute to documentation would be very useful."
[8] "Right now I'd like to see how the documentation project goes."
[9] "Helping to facilitate projects to make the R Core more reasonable. The two suggestions I have are having a project to improve the documentation of old functions especially commonly used ones. These are often written in insider language (“a wrapper for noobs”) or to explain the problem code was written to solve, but not how it is actually used by other people. Terms used are not defined, short hand link () to execute is used, which beginners may not know what is being done if they haven't seen that before. I've often wondered what would happen if I tried to submit updates. Also a project on error catching would be great. Lack of catching (and in general lack of checking returns) is a huge problem in R packages."
[10] "Improving base R documentation."
[11] "Working towards consistency, simplification, consolidation, and possibly quality control of packages and particularly package documentation (by supporting CRAN)."
[12] "Taking care of language issues (shiny, decoding), timezone problems, integrate data.table into the base packages. Better documentation for markdown, knit, pandoc templates, parameters and option"

[13] "Creating high quality documentation / discussion about R (How to make a great function, package,) that won't be accepted on SO or elsewhere because it is non objective. Review packages too"
[14] "Improving the R documentation."
[15] "Keep funding R infrastructure! Documentation, checkpoint/pack rat package style improvements to specifying specific package versions in scripts and projects (this is another major pain point), more on-boarding material that, statistics teaching material, so much could be done!"
[16] "Better documentation format. The content of the current help files is good enough, the presentation is not. Many attempts, nothing changed - with the exception of vignettes/knitr/markdown support. pkgdown is a good concept, output format should be standard. Work that there are more tests, mainly in standard packages. When 3.4.0 came out, I was flooded by warnings about partial matching in my old reports, partially from standard packages such as RODBC. For this one, I even submitted an almost trivial pull request, https://github.com/cran/RODBC/pulls/dmenne, but nothing happened. At user2017, I spoke about the issue with several people, but nobody wanted to muck with the r-core team for fear of being ripleyed (not by brian, there are other ripleys now)."
[17] "It could create a Task Force to improve the documentation of R to reach audiences beyond the statistically oriented users."
[18] "Improve package and base function documentation and examples for beginners. Even as an experienced user, the documentation pages (especially examples) are often incomprehensible to me."
[19] "Undertake a massive documentation rewrite effort."
[20] "Revamping CRAN and the submission process. Some sort of basis peer-review of package documentation. Speeding up and optimizing R so that it can handle larger datasets. A curated documentation for popular packages. Develop packages similar to shiny to create simple GUI."

Project

[1] "Funding of projects serves the community as a whole and should continue."
[2] "Solving package dependency, package versioning, project reproducibility, dependencies on packages."
[3] "Because of the large corporate sponsors and high visibility, projects to facilitate R-related employment opportunities would likely be very useful to the r community."
[4] "I'm not exactly clear what their projects or priorities are. Not sure I have seen much communication about that, so improving the visibility"
[5] "Stop exclusively promoting RStudio affiliated projects."
[6] "I would the Consortium to found Duncan Temple Lang RLLVM package, it will make R way faster (like Julia). Generally, it is great that they fund great project and build a coherent R ecosystem."
[7] "I think the projects are great! Just continue doing that."
[8] "Continue to support community projects to keep improving R libraries."
[9] "Continue projects related to databases and web services connectivity. Maybe also JS and V8 standard interface..."
[10] "I would love to have a more formalized mechanism for support for the R education projects I run."
[11] "Supporting projects"
[12] "Right now I'd like to see how the documentation project goes."
[13] "Investing in more projects that will make R code deployable in production directly."
[14] "Working seriously on internationalization. As a professor who teaches research methodology I think it's one of the biggest barriers to widespread adoption in non English speaking countries. I would happily volunteer in a project like that, but I don't know how. Make it easy and people will show up to help!"
[15] "Helping to facilitate projects to make the R Core more reasonable. The two suggestions I have are having a project to improve the documentation of old functions especially commonly used ones. These are often written in insider language ("a wrapper for noobs") or to explain the problem code was written to solve, but not how it is actually used by other people. Terms used are not defined, short hand link () to execute is used, which beginners may not know what is being done if they haven't seen that before. I've often wondered what would happen if I tried to submit updates. Also a project on error catching would be great. Lack of catching (and in general lack of checking returns) is a huge problem in R packages."
[16] "Keep on funding useful projects!"
[17] "Push the directions and future standards of the R project. Consolidate the community and spread the word."
[18] "I hope the Distributed Computing project takes off soon."
[19] "Idea of project:1 - Refurbish the RPub concept, adds automatic tags, order the rmd, make it social and collaborative. 2 - Support an upgraded version of opencpu with possibility of login and stuffs."

[20] "I would love to see a project funded for the deployment of R models. I know 100 people that can train a good model in R and 0 people that can put one into “production”. I would love to see this process made easier. Microsoft's R on SQL Server has done a lot (for me and my company) in this regard, but I would love to see an open source framework for deploying R models as APIs, microservices, containers, etc."
[21] "Keep funding R infrastructure! Documentation, checkpoint/pack rat package style improvements to specifying specific package versions in scripts and projects (this is another major pain point), more on-boarding material that, statistics teaching material, so much could be done!"
[22] "- keep on funding projects that close critical gaps in the R ecosystem"
[23] "Sponsoring more projects to increase the quality of critical standards and projects. For example, supporting development of the DBI standard."
[24] "Supporting more foundational projects like database infrastructure to make R more robust and provide a foundation for other package developers to build on."
[25] "Easier ways to get funding to projects"
[26] "I'll wait till already supported projects close, those look nice already."
[27] "Continue to pioneer projects that ensure the future of R, making it accessible, more powerful and useful."
[28] "Making sure that Gabor Csardi has enough time to develop igraph alongside all of his other brilliant projects (I work a lot with network data)."
[29] "Maybe specific support for technology transfer from Academia to Business. For example my research group help companies using Statistical Process Control with R, the consortium could have a funding program for such short applied projects."
[30] "Continue to support projects that help consolidate the tools that 80% of users use 80% of the time. Let other research teams support fringe projects."
[31] "funding Projects which help to Scale CRAN-Packages on distributed Platforms (Spark, Hadoop).I know, this is a HUGE effort, but it's well worth it"
[32] "Get more funding, fund more cool projects. More frequent grant cycles, perhaps"
[33] "Make R an option for commercial projects of big players by providing suitable licensing"

Website

[1] "Finally overhaul the design of the R website ????"
[2] "Getting more serious software engineers and developers interested in joining the ecosystem, however that is interpreted. Creating a better website for CRAN would be a nice start."
[3] "Helping fund core R activities (e.g., modernizing the CRAN website, helping manage the bug tracker, maybe more documentation work)."
[4] "linux package binaries!!pay meetup fees for R user groups + central website for RUGs incl place to share ideas + experiences start discussion with R foundation + core group whether to unify API (e.g. naming of functions, arguments etc.) or even completely rewrite baseR with the knowledge we now have"

Pain

[1] "Make sure R stays relevant with all the movement around it. Keep fixing the pain points"
[2] "Keep funding R infrastructure! Documentation, checkpoint/pack rat package style improvements to specifying specific package versions in scripts and projects (this is another major pain point), more on-boarding material that, statistics teaching material, so much could be done!"
[3] "Identify the pain points in R and fix them with the community; know better the R users (this survey is a great step into that direction)."

Know

[1] "I work with companies that are paranoid about using non-validated pieces of software (stupid I know). It would be nice if the R Consortium provided a lead on this (and not just plugging Mangos “validated R”).Also as I small business that uses R, I would like to join the Consortium, but the entry level sign-up fees are fairly high. Perhaps bronze level or startup fees?"
[2] "I work in a public hospital and am a bit of a lone ranger when it comes to my use of R in my organization. I would love to be able to implement an enterprise solution that could replace our reliance on sharing data via Excel spreadsheets and PowerPoint slides but I lack the technical know-how to scale my skills. My hospital is a Windows shop I have found it impossible to get buy in from our IT department to see the benefits of setting up an R environment that can be accessed by users across the organization. Simplifying enterprise level implementation that is cost effective for budget conscience public health facilities would be extremely well received."
[3] "I don't know."

[4] "I don't know."
[5] "Do not know"
[6] "Don't know"
[7] "Also don't know"
[8] "I don't know. It's still relatively new. If I've learned anything from 20 years of work with W3C, keeping the major vendors from taking over all of the key decisions would probably be the best thing it can do for the community (but I guess that's the R Foundation's work anyway). Trust me--the vendors, at the end of day, have only their best interests at heart (despite what they say and despite what they might fund in the short-run). And that's coming from a "business professor" (sigh)."
[9] "I know this is already underway, but improved documentation / ways to contribute to documentation would be very useful."
[10] "Unknown"
[11] "Working seriously on internationalization. As a professor who teaches research methodology I think it's one of the biggest barriers to widespread adoption in non English speaking countries. I would happily volunteer in a project like that, but I don't know how. Make it easy and people will show up to help!"
[12] "I believe it is doing a good job already. Providing new packages gives a lot more visibility than improving the existing ones, and fortunately I know you already do help to improve existing packages (e.g. with DBI). Continue improving what already exists as well."
[13] "Helping to facilitate projects to make the R Core more reasonable. The two suggestions I have are having a project to improve the documentation of old functions especially commonly used ones. These are often written in insider language ("a wrapper for noobs") or to explain the problem code was written to solve, but not how it is actually used by other people. Terms used are not defined, short hand link () to execute is used, which beginners may not know what is being done if they haven't seen that before. I've often wondered what would happen if I tried to submit updates. Also a project on error catching would be great. Lack of catching (and in general lack of checking returns) is a huge problem in R packages."
[14] "don't Know"
[15] "Who knows"
[16] "I would love to see a project funded for the deployment of R models. I know 100 people that can train a good model in R and 0 people that can put one into "production". I would love to see this process made easier. Microsoft's R on SQL Server has done a lot (for me and my company) in this regard, but I would love to see an open source framework for deploying R models as APIs, microservices, containers, etc."
[17] "I don't know"
[18] "dont know"
[19] "Help companies to know the benefit of R"
[20] "Doing more to destroy the Excel - PowerPoint vortex of evil. I know RStudio has development ongoing that will help with this, but those are really, really (really!) not good tools these days."
[21] "linux package binaries!!pay meetup fees for R user groups + central website for RUGs including place to share ideas + experiences start discussion with R foundation + core group whether to unify API (e.g. naming of functions, arguments etc.) or even completely rewrite baseR with the knowledge we now have"
[22] "Customers don't know what they want. I am a customer. Satisfy me."
[23] "I do not know"
[24] "Identify the pain points in R and fix them with the community; know better the R users (this survey is a great step into that direction)."
[25] "Know more about R users through surveys like this perhaps, so users' voice can be heard and the language can be improved"
[26] "I don't know"
[27] "Increase the support to RLadies. Come to our region to know all the R fans."
[28] "Don't know"
[29] "n/a, I trust they know much better than I do :)"
[30] "I'll likely approach other workers to see how some streamlining and consolidation of packages might be accomplished and in conjunction with others possibly make a proposal to the Consortium. I'm also looking at knowledge transfer to younger workers, as many of the underlying computational methods are inaccessible to those who haven't worked with the older computational environments, esp. Fortran. That may also be something the Consortium will want to support, though it has application much wider than R."
[31] "I don't know"
[32] "funding Projects which help to Scale CRAN-Packages on distributed Platforms (Spark, Hadoop).I know, this is a HUGE effort, but it's well worth it"

[33] "As a data scientist working mainly with R, I would like to see the platform in more success cases of implementations in production. That will help the users' possibilities to specialize in the language having a good job market to deploy their knowledge."

Python

[1] "Figuring out how to make R more production friendly and rival Python's stronger presence in machine learning"
[2] "Develop initiatives for engineering applications, such as thermal analysis. However, that might be better suited for Python or Octave."
[3] "R Consortium should work to make R competitive with other general purposes languages such as Python and Java."
[4] "Get the R foundation to publish a roadmap. Support the speed up of R (manage more data faster). Improve activities in areas where R is loosing it, compared e.g. to python:. NLP (e.g. poor / slow implementation of word2vec). Deep learning. Reinforcement Learning"
[5] "I think it should help people (and group) around the world to share a common way of teaching, promoting and speaking about R in order to more rapidly close the gap between the “old way” and the “new way”. Moreover, it should help promoting R as a valid langage (as python) to use in IT department and in production. It is not just a langage for prototyping and for research. I think it could help a lot the community."

Support

[1] "Keep up the good work! Anything that supports community building, particularly for those currently under-represented."
[2] "Optimizations in other R implementations (Renjin, ...) should flow back into vanilla RSupport efforts to improve documentation"
[3] "I’d like to see more targeted support for regional useR! conferences. At this point they seem more grassroots and ad-hoc. Be nice to have it a bit more coordinated."
[4] "Addressing needs for production R like supporting Rocker, MRAN, maybe threading, richer database writing"
[5] "Keep driving commercial support of R and integration with other technology so that it stays relevant in industry, and keep investing in teaching and learning resources (e.g. conferences, online webinars, learning platforms like Swirl, DataCamp, Coursera etc) so that R users can keep up to date with R packages & developments."
[6] "- I think R Consortium should support people to organize more conferences in countries other than US and Europe. I am from India and I want an R conference in my country.- Fund more initiatives such as Software Carpentry R Instructor Training in different"
[7] "I would like to see more support for people doing econometrics with R. But I understand that this is not the main objective for the R consortium but rather the R package contributors."
[8] "Supporting R as a language for general computing will help every R user in the end"
[9] "Continue to support community projects to keep improving R libraries."
[10] "Be more inclusive. As much as I love Hadley, for example, I do not want to see R become R-tidy. Be careful not to become too much of a walled garden between R-Foundation, R-Consortium, and RStudio. Also, please support WIndows more. I've been on R-devel/bugzilla for years trying to get simple things fixed or enhanced for Windows users, and we are always the bastard stepchild of the R infrastructure."
[11] "I would love to have a more formalized mechanism for support for the R education projects I run."
[12] "Support for R users & developers in the 3rd world"
[13] "Supporting projects"
[14] "Supporting participation of underrepresented minorities in the community. R-ladies is a great initiative and should be enhanced and continued but there doesn't seem to anywhere near similar support for people of color and other underrepresented groups."
[15] "A web page for our community would be great. Support in organizing events"
[16] "Focus on supporting and expanding R community outside North America and Europe."
[17] "fixing some basic R issues such as not having support for UNC in windows supporting a next generation of R, e.g., with tidyverse as the default, and on a much faster platform."
[18] "I think that the R Consortium is potentially the only body that can support “meta” package efforts to make the R ecosystem more coherent, because this requires real resources. One of R's biggest problems is highly inconsistent APIs and poorly defined/overlapping package scope. Supporting efforts to make the package ecosystem more holistic, well-defined, and consistent would be a worthy goal, and certainly make my life easier."

[19] "Expand our community, work on making hacks of the tidyverse un-hacky, support our volunteer developers."
[20] "Idea of project:1 - Refurbish the RPub concept, adds automatic tags, order the rmd, make it social and collaborative. 2 - Support an upgraded version of opencpu with possibility of login and stuffs."
[21] "More free training and working to get more R support by proprietary software, such as complete integration into Tableau. The ability to directly read Excel. With Microsoft being part of the Consortium I find it odd that R doesn't directly read and write to the entire Office suite."
[22] "Working towards consistency, simplification, consolidation, and possibly quality control of packages and particularly package documentation (by supporting CRAN)."
[23] "Keep supporting R"
[24] "Provide more support to the R Foundation."
[25] "Get Microsoft to support SQL Server in DBI better"
[26] "Better documentation format. The content of the current help files is good enough, the presentation is not. Many attempts, nothing changed - with the exception of vignettes/knitr/markdown support. pkgdown is a good concept, output format should be standard. Work that there are more tests, mainly in standard packages. When 3.4.0 came out, I was flooded by warnings about partial matching in my old reports, partially from standard packages such as RODBC. For this one, I even submitted an almost trivial pull request, https://github.com/cran/RODBC/pulls/dmenne, but nothing happened. At user2017, I spoke about the issue with several people, but nobody wanted to muck with the r-core team for fear of being ripleyed (not by brian, there are other ripleys now)."
[27] "Support more individual grants"
[28] "Support development of base R."
[29] "Sponsoring more projects to increase the quality of critical standards and projects. For example, supporting development of the DBI standard."
[30] "Supporting more foundational projects like database infrastructure to make R more robust and provide a foundation for other package developers to build on."
[31] "Get the R foundation to publish a roadmap. Support the speed up of R (manage more data faster). Improve activities in areas where R is loosing it, compared e.g. to python:. NLP (e.g. poor / slow implementation of word2vec). Deep learning. Reinforcement Learning"
[32] "support underlying computational infrastructure: more general multi-core parallel internals, gpu computing for rasters"
[33] "Increase the supportive to RLadies. Come to our region to know all the R fans."
[34] "Provide support in places which depends on R repos far from their locations"
[35] "I'll likely approach other workers to see how some streamlining and consolidation of packages might be accomplished and in conjunction with others possibly make a proposal to the Consortium. I'm also looking at knowledge transfer to younger workers, as many of the underlying computational methods are inaccessible to those who haven't worked with the older computational environments, esp. Fortran. That may also be something the Consortium will want to support, though it has application much wider than R."
[36] "- Streamlining of open source licenses used in R packages- Support further development of opencpu"
[37] "support"
[38] "I'll wait till already supported projects close, those look nice already."
[39] "Even more support for the idea of versioning/reproducibility."
[40] "Supporting user groups and sponsoring more training opportunities"
[41] "Could we get more funds (from Microsoft and philanthropists) to support some persons around the world (like myself) . I am a bit older, 4 degrees , experienced in a scientific area and want to teach- How do I get to do that - within universities- they are not open to new ways of teaching yet, and the workload is too high . (I am moving out of a university- as I have more potential to effect the changes that I can see are needed). The university system our crushes our creativity, and does not allow the new learning methods to be integrated. At my university we have awful systems like Blackboard to contend with. "
[42] "Support enterprise needs"
[43] "Maybe specific support for technology transfer from Academia to Business. For example my research group help companies using Statistical Process Control with R, the consortium could have a funding program for such short applied projects."
[44] "Continue to support projects that help consolidate the tools that 80% of users use 80% of the time. Let other research teams support fringe projects."
[45] "Continue what is already doing and also try to make the R support OOPS"
[46] "Support user groups, support developers "

[47] "1. Add more Spotfire or Tableau kind of dynamic controls in R/Shiny so we do not need to rely on Spotfire or Tableau. 2. Make R more useful to data processing so we do not need to rely on SAS. 3. Add native support of parallel computing in base R to take advantage of multi-core CPUs."

Standard

[1] "To design a better language standard, like C++ 11 for C++"
[2] "Continue projects related to databases and web services connectivity. Maybe also JS and V8 standard interface..."
[3] "Continue to promote quality standards, prevent too much overlap and duplication of efforts across packages."
[4] "More active R development or standardization of methods and techniques."
[5] "Getting R approved by the FDA as a standard way to deliver datasets."
[6] "Push the directions and future standards of the R project. Consolidate the community and spread the word."
[7] "R Consortium should1. Promote R, at the workplace, at schools, etc. 2. Work with R Core Team to address any gray areas in the language, i.e., to codify R language, in the sense of what the C Standard committee did, not what the C++ Standard Committee did. For example, codify and document how best to write R packages using native languages such as C.3. To further document and clarify the gray areas in the language."
[8] "I'd love to see more standardization with commonly used API's. RODBC is a successful example here."
[9] "Avoid parallel standards, assuming a strong position on some aspects of R that need to be strictly standardized."
[10] "Better documentation format. The content of the current help files is good enough, the presentation is not. Many attempts, nothing changed - with the exception of vignettes/knitr/markdown support. pkgdown is a good concept, output format should be standard.Work that there are more tests, mainly in standard packages. When 3.4.0 came out, I was flooded by warnings about partial matching in my old reports, partially from standard packages such as RODBC. For this one, I even submitted an almost trivial pull request, https://github.com/cran/RODBC/pulls/dmenne, but nothing happened. At user2017, I spoke about the issue with several people, but nobody wanted to muck with the r-core team for fear of being ripleyed (not by brian, there are other ripleys now)."
[11] "Sponsoring more projects to increase the quality of critical standards and projects. For example, supporting development of the DBI standard."
[12] "Trying to align how packages work a bit more. Not to say a heavy hand should apply here. Maybe packages that meet certain standards get a certified sticker"