Six Years of Shiny in Research - Collaborative Development of Web Tools in R

by Peter Kasprzak, Lachlan Mitchell, Olena Kravchuk and Andy Timmins

Abstract The use of Shiny in research publications is investigated over the six and a half years since the appearance of this popular web application framework for R, which has been utilised in many varied research areas. While it is demonstrated that the complexity of Shiny applications is limited by the background architecture, and real security concerns exist for novice app developers, the collaborative benefits are worth attention from the wider research community. Shiny simplifies the use of complex methodologies for people of different specialities, at the level of proficiency appropriate for the end user. This enables a diverse community of users to interact efficiently, and utilise cutting edge methodologies. The literature reviewed demonstrates that complex methodologies can be put into practice without insisting on investment in professional training, for a comprehensive understanding from all participants. It appears that Shiny opens up concurrent benefits in communication between those who analyse data and other disciplines, that would enrich much of the peer-reviewed research.

Introduction

Data is the backbone of research. With the rise of automated data gathering tools, data size and complexity of analysis have driven a growing gap between research disciplines and the required data analysis. Another issue is the fact that different approaches to the same data can compromise validity, as seen in an analysis on effect sizes in observational studies, which found that varied methodological workflows could reverse conclusions regarding the studied intervention (Donoho, 2017). Collaborative learning which employs common task frameworks can help interpret, quantify, and possibly cap methodological variation across disciplines (Donoho, 2017). Software such as Matlab Moler and Mathworks (2012), Minitab Arend (2010), Genstat Payne et al. (2007) and SPSS Landau and Everitt (2004) have attempted to bridge this gap by creating more user friendly interfaces that either make coding more intuitive and easier to learn, or use drop down menus and radio button selection to bypass the command line. Current analytical software, such as those mentioned above, each have their own limitations which include non publication ready quality graphics, non-intuitive drop down menus, restrictive interfacing with other software, price point (including the cost of licensing the proprietary software) and the difficulties that inevitably occur when colleagues attempt to run code originating from other software on their preferred platform. Despite its own weaknesses, which include a very steep learning curve and non-intuitive

programming language, R Team (2019) has grown to become the most popular programming language for statistics and biological data analysis, spawning over 14,000 free to use packages over a wide range of subject material (Li et al., 2018).

While code of any language can be shared easily between users, general use requires a level of familiarity with the specific program. Transforming a piece of R code into a interactive app capable of use by a broad audience recently required either knowledge of other coding languages (Gunuganti, 2018), or consultation/collaboration with a computer scientist/app developer and the added time and cost justified. Specialist apps that benefit only a small number of people often do not meet cost/benefit benchmarks, which means that many useful advancements have stalled due to experience requirements with data analysis software, or an understanding of the underlying theory for day to day use, constituting a complexity barrier (DePalma, 2013). Shiny can generalise R code for all levels of users, bringing the latest advancements in methodology measurably closer to everyone. This does create new issues relating to data security, as novice app developers will now require a knowledge of web internet protocols for secure data transfers to be assured.

The increased use of technology, sensors and other data capture devices have brought an interesting issue to light. Researchers and practitioners without a background in data analysis now have the ability to gather large amounts of data (LaZerte et al., 2017). Limited options exist for those without data analysis training to correctly analyse data gathered from the field and experiments, which has arguably led to issues impacting experimental reproducibility. A Nature survey of 1,576 researchers from the disciplines of chemistry, physics and engineering, earth and environment, biology, medicine and other, found more than 50% surveyed believed that low statistical power or poor analysis was a strong contributor to irreproducibility (Baker, 2016). In the same survey more than 90% of respondents believed that a better understanding of statistics was required to drive reproducibility of research. Learning analytical methodologies and programs is a non-trivial task, and subcontracted analysis, even within house, generally comes with a wait for results. Purchasing proprietary software can be inflexible and often expensive which takes resources away from research, and open source software is dependent on a minimum level of computer literacy, and the ability to test the software to ensure correct results is essential (LaZerte et al., 2017).

Open source and free, Shiny has grown in popularity with the first Shiny Developer Conference held in January 2016 and a growing use in peer reviewed academic papers. While the number of papers has steadily increased each year, Shiny remains an incompletely explored topic, with the potential for Shiny to make a significant positive contribution to the general field of science not yet properly examined. To the best of our knowledge this is the first Shiny review.

The rest of this review is organised as follows. Section 2 details the literature search, the keywords and findings. Section 3 presents the technical aspects of Shiny, including hosting costs and security, along with restrictions. Section 4 discusses the use of Shiny in research with relevant examples from the literature, and finally section 5 gives the conclusion along with the authors opinion.

Algorithms/methods for literature search

A thorough search for Shiny results in the academic literature was undertaken to investigate the growth from 2012 - 2018 in research, and which publications and subject areas were represented. The focus of this paper is the use of Shiny to

bridge specialist academic and theoretical innovations, and its role in disseminating knowledge to government, industry and the general community, therefore, it is acknowledged that this is a non-exhaustive list of Shiny case uses. We acknowledge that the literature search is not fully comprehensive, as newspaper articles, blogs and other non-academic areas were filtered, which makes this review biased towards an academic standpoint. The search was conducted with four major data bases. Web of Science and Scopus were used due to their reputation as multi-disciplinary databases, with Google Scholar and the University of Adelaide (UofA) utilised as their algorithms search the entire document and all fields for keywords. The keywords used in all searches were of the form "Shiny Web Application" OR "Shiny Web App", with an exact search not suitable in this case, and "R" not included to avoid the inevitable non-related hits. The search was then filtered by year to span 2012 - 2018 and the document type was limited to Dissertations, Articles, Conference proceedings and Reviews (where allowed), to investigate the use of Shiny in the research literature only. Books were excluded from the search due to the small number of published materials. A separate search conducted for books showed that as of 2016, only two books were written on the use of Shiny, with both being structured as instructional manuals. Beeley (2013) takes the beginner from their first application and walks them through the major concepts to more complicated applications, while Moon (2016) uses Shiny to teach **ggplot2** (Wickham et al., 2018) graphics. As of 2018 a Google search for "Shiny Web Application Books" yielded seven results, including one second edition release.

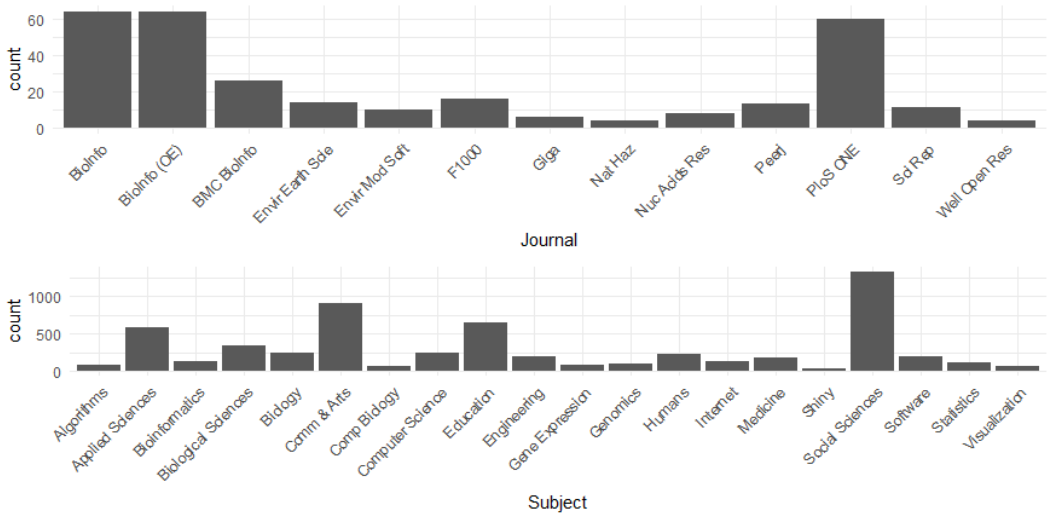
The UofA search engine is powered by ExLibris Primo, which includes all resources owned or subscribed to by the library and selected free and open access resources. It includes 345 databases, and links to the major collections of articles and eBooks totalling over 50 million items, which can expand out to 100s of millions. The UofA search was conducted with the terms "Shiny web app OR Shiny web application" and returned 5,251 results with 1,391 peer reviewed articles, 3,456 dissertations, 18 reviews and 119 conference proceedings, which are broken into results by journal title, subject tag and languages published in, displayed in Figure 1.

The search in Scopus used TITLE-ABS-KEY(Shiny AND web AND app*) AND PUBYEAR $>$ 2012 AND PUBYEAR $<$ 2019 as its search terms, with the same document limitations. The decision was made to only check the title, abstract and keywords as too many irrelevant results were being returned when including other fields, with a final result of 155 items. These were restricted once again to articles (114), conference papers (38), conference reviews (2), and reviews (1). These are broken into number of records published by year, journal title and subject tag displayed in Figure 2. There were 154 records published in English, with one record published in Spanish.

The Web of Science search returned 144 results using the search criteria ALL=(Shiny Web App*) and filtered to the same time frame. Choices of document criteria included Articles and Proceedings papers which resulted in 110 Articles and 34 Proceedings papers, with dissertations not returned in this search. These are once again broken into publications per year, journal title and subject tag displayed in Figure 3.

Again the predominate language was English with 142 records, one Spanish, and one Portuguese record found.

The Google Scholar search terms used first were [Shiny web app | application] which returned 16,400 results. A range of additional terms were used to narrow down results including, "security OR complexity OR architecture OR hosting", with "Shiny" being a required keyword, which returned approximately 10,400 results.



Language	English	French	German	Chinese	Portuguese	Spanish	Chamorro	Slavic	Italian	Norwegian
Count	5232	5	8	5	4	4	1	1	1	1

Figure 1: Summary of UofA search results partitioned into number of results by journal for records > 3 with abbreviations given in Table 1, results by subject tag with abbreviations given in Table 2, and published languages.

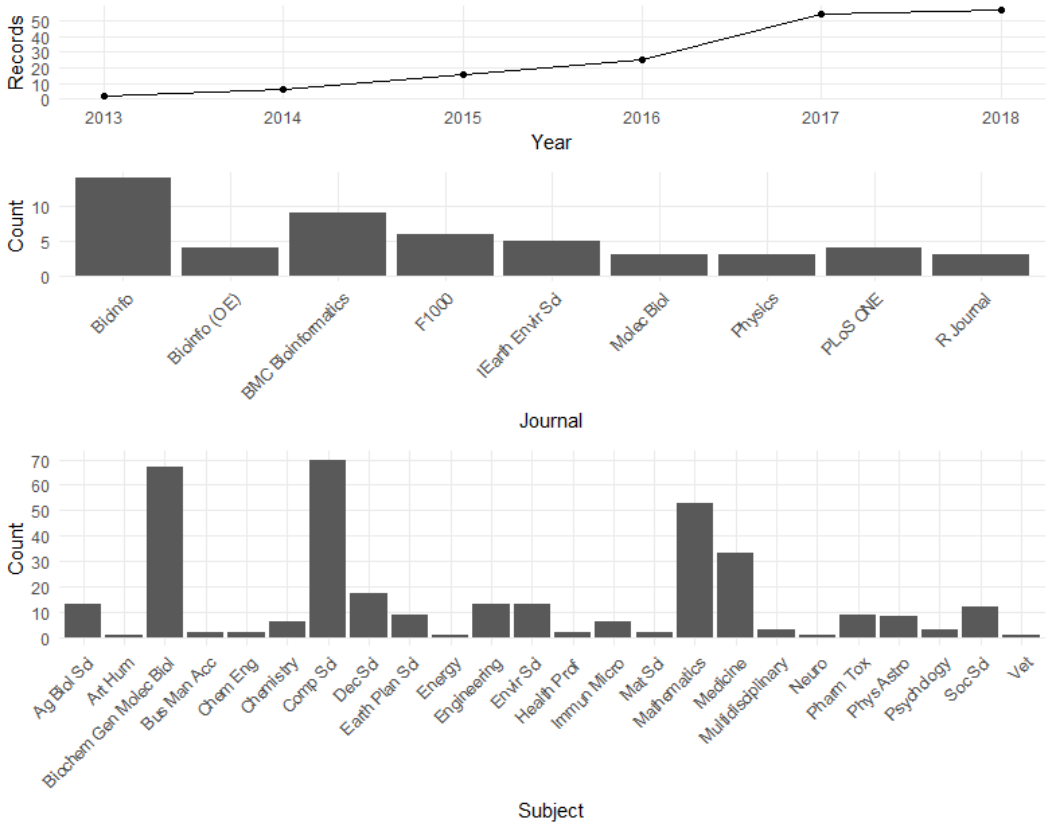


Figure 2: Summary of Scopus search results partitioned into number of published results by year, journal title for records > 2 with abbreviations given in Table 1 and results by subject tag with abbreviations given in Table 2.

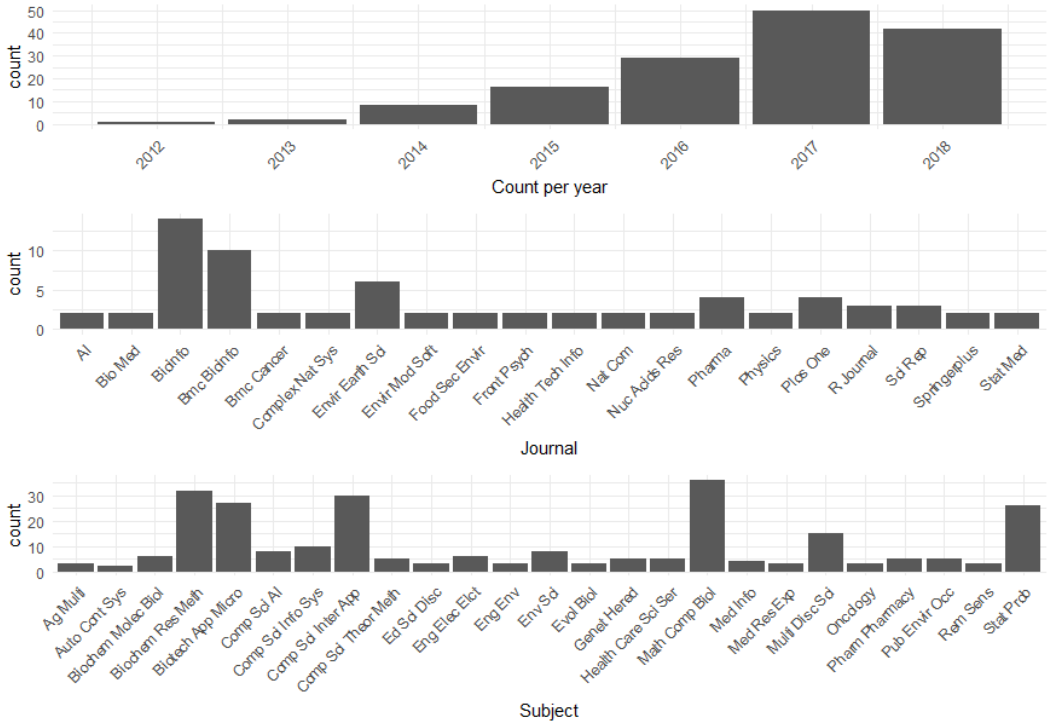


Figure 3: Summary of WOS search results partitioned into number of published results by year, journal title for records > 2 with abbreviations given in Table 1 and results by subject tag > 2 with abbreviations given in Table 2.

Unfortunately by record 135 irrelevant results were found that did not contain the required term "Shiny", which appeared to be an error in the algorithm. Given that the search returned over 100% more results than the UofA search, it was decided to not use these results to create this paper, as the UofA search utilised the Google Scholar databases.

Shiny is a relatively novel tool with the total number of papers found quite small in comparison to larger bodies of work. Assuming the UofA library search completely covers the other 3 databases (which it is advertised to do), there is an approximate total of 5,000 unique peer reviewed pieces of work utilising Shiny since 2012, an average of over 700 papers per year. All searches showed that Bioinformatics journals published the largest number of Shiny papers, however, the vast majority of papers were published by a diverse range of titles, in a diverse range of fields. This indicates that Shiny is a flexible tool and not area specific. Only the Scopus search returned slightly different information with Computer Science, Biochemistry, Mathematics, and Other subjects tags registering the largest number of relevant hits. On closer inspection, while Bioinformatics did not register as a subject heading, the journal that published the greatest number of papers was Bioinformatics, followed by BMC Bioinformatics, which suggests that there is simply a difference in subject labelling. Far more papers were found by keyword searches in the body of the document, as evidenced by the total numbers of papers found by the library search from the UofA. This suggests that Shiny has been utilised as a general tool, and not as a new discovery in the later years. The vast majority of all papers were written in English, with some European countries represented, but very few Chinese papers.

The results of the search algorithms are reasonably reproducible, with some fluctuation occurring depending on the sources of publications, and performance of

the search engine. In our experience the fluctuation is less than 10%. Google scholar significantly alters the number of found papers depending on sorting. If sorting by relevance is checked then 127,000 results are found. Sorting by date reduces this number to what is stated above.

A subset of 600 papers were chosen for thorough reading to inform this report. These were the top 600 results returned by the UofA records search when sorted via relevance. The relevance ranking employed by ExLibris Primo is comprised of four main criteria.

1. Degree of match: Fields such as title, author and subject field are given a higher ranking, along with order of the query terms and completeness of phrases.
2. Academic significance: Citations and journal impact factor.
3. Type of search: Primo infers if the search is broad-topic or specific-topic, with broad topic searches amplifying overview material such as reference articles.
4. Publication date: Newer material is given preference.

The papers that discussed Shiny generally had "Shiny" in the title and/or the subject fields, increasing their relevance score. Earlier papers were more likely to discuss Shiny, with newer papers more likely to mention Shiny in the text only. The relevance search yielded a high number of the older papers as high relevance, along with a very broad range of use cases. The limit of 600 papers was an empirical cut off point, as this was the stage that papers had ceased discussing Shiny, and were only stating its use. It was decided that enough use cases had been examined to make comments regarding Shiny's relatively widespread use in the academic work. 445 original applications were introduced in these papers, which utilised 373 unique R packages. 229 unique peer reviewed journals were represented with 55 published in Bioinformatics, 31 published in PLoS ONE, and 21 published in BMC Bioinformatics. The final subset of papers that most thoroughly discussed the implementation of Shiny were chosen to create this report, and are given as references.

Technical aspects

Architectural overview

A Web application framework for R, Shiny was conceptualised by RStudio's CTO Joe Cheng and announced at the Joint Statistical Meeting conference in July of 2012 as a tool designed to help R programmers create interactive web applications, reports and analysis without the need to know HTML, CSS, or JavaScript (Chang et al., 2018).

The power of Shiny comes from the ability for an R user to quickly and simply code a reactive framework. A reactive framework allows objects to be updated when a source is changed, along with all connected objects. For example, in an imperative programming paradigm such as the R language, setting the line

$$c = a + b$$

means that c is assigned the sum of previously defined terms a, b and will not change when the values of a, b are changed without the variable c being re-evaluated.

Reactive programming allows the value of *c* to be updated almost instantaneously, including all other variables and outputs dependent on *c*, whenever *a* or *b* is changed. R completes this task with information travelling from input to output in a pull fashion. A pull fashion is when *c* learns of the new value of *a* or *b* when *c* is called. Shiny creates a system of alerts which flag changed expressions and the server re-evaluates all flags in an event known as a *flush* (Grolemund, 2015). Using two object classes called reactive values, such as a = reactive(), and observers, such as b = plot(), Shiny creates a *reactive context* between the two objects known as a *call-back* which is a command to re-evaluate the observer. Multiple observers can be linked to the same *reactive value* and the server will queue up all *call-backs* and run each *call-back* in the event of a *flush* (Grolemund, 2015).

This reactive framework allows user inputs to be evaluated via a UI (user interface) with a series of easily coded widgets such as text boxes, radio buttons and drop down menus from pre-programmed R code. Shiny then seamlessly updates outputs of tables, plots and summaries. A non R user can change the values of *a* and *b* via the user interface and explore the pre-coded results dependent on *c*.

A Shiny application has two main parts. A user interface object and a server function. The user interface contains code for the layout and appearance of the app, with default choices restricted in appearance. Layouts can be customised and changes to the appearance can be made if the programmer has some knowledge of HTML or CSS. For standard applications simple commands suffice and a knowledge of HTML or CSS language is not required for tweaks. The server function houses all the code that drives functionality of the application and can utilise all the built in programs available to R and RStudio users.

Hosting

For a small number of applications and limited run hours the cost of hosting a Shiny application is free, but it can become expensive quickly. Hosting on shinyapps.io requires no system administration knowledge and comes with layers of security and is supported by Shiny's IT team. According to the RStudio pricing website (Core Team, 2012), the platform is free for 5 applications and 25 active hours which increases to $39 AUD a month for unlimited applications and 500 active hours, to the top tier of $299 AUD a month, which allows for unlimited applications and 10,000 active hours. Shiny also has the option of Shiny server, Shiny Server Pro or RStudio Connect. These require a level of system administration knowledge, and also requires the apps to be hosted on a physical or virtual machine. RStudio Server Pro costs $9,995 AUD per year (Core Team, 2012). RStudio connect allows installation of software on a server behind your existing firewall and costs between $14,995 AUD per year ($62 AUD per user/month) to $75,995 AUD per year ($6.25 AUD per user/month) for a larger, specified number of named users (Core Team, 2012). Shiny server prices were not available. For those with an in-depth knowledge of internet security, it is possible, and more economical, to host the application independently by their own means.

Security

As Shiny is primarily a web technology, a very strong focus on application security must be adhered to, with novices in computer science more likely to make critical mistakes (Charpentier, 2013). A well-known concept in cryptography and web

security is *unknown unknowns* (Charpentier, 2013). Put simply, this refers to the fact that a developer cannot build defences for attack vectors they are unfamiliar with. For this reason, it is generally wise to leave the specifics of data security to experts in the field, with the end-developer instead relying on the vetted work that has been done for them.

For users of Shiny who elect to use shinyapps.io by RStudio, this is essentially what happens. Once uploaded, the application is secured behind best practices (Core Team, 2012). Unfortunately, this service is prohibitively expensive when compared to hosting the server on a cloud platform like Amazon Web Services (AWS) (Amazon, 2019) or Microsoft Azure (Microsoft, 2019), which requires application security to be taken into the app creator's hands. Certificates need to be created and kept up-to-date, servers need to be configured for HTTPS amongst other security protocols (Charpentier, 2013). Due to the local nature of R, this is likely to be a new issue, requiring a new set of skills, for many data analysts operating on the platform.

Architectural issues

Curiously there are only a small number of papers that explicitly mentioned concerns and limitations with respect to the use of Shiny to develop research focused apps. A paper by Dwivedi and Kowalski (2018) was the first to include a limitations section, emphasising the requirement for a fast internet connection when dealing with large data sets. This could be mitigated with the use of cloud based resources to store the data and host the app, with potentially faster network and processing speeds available with respect to local connections. Guo (2018) found R package updates a legitimate concern, as updates can occur without warning and crash an application. A less serious issue, the lack of flexibility of the dashboard is born from the simplification of creation, with Shiny's dashboard not being as flexible as one created in Java (Ge et al., 2018). There are however challenges with the use of Shiny, with one of them being the background architecture.

While Shiny has many benefits, the architecture of Shiny will be a limiting factor when building complex applications. Previously this concerned dismissed with Joe Cheng stating more recently:

> In the past, we've responded rather glibly to these requests: "Just use functions!" (Cheung et al., 2017)

As of 2017 Shiny has made moves to address this issue with the creation of modulisation (Cheng, 2017), however, the more involved use cases would be handled better by other computing languages, for the reasons detailed below. An analogous way of conceptualising this would be in the difference between applets and applications. Applets are generally small, discrete, and of low complexity, and are developed to perform a small number of functions for a highly specific purpose. Most Shiny products would fit this description quite well, while applications on the other hand, are generally more complex (Fayram, 2011). They are built for a number of different use cases, and tend to have relatively large codebases. Well established and popular web application frameworks such as Angular and React exist to fit these situations, containing much more general functionality than Shiny with much less specific functionality (such as functions related to data visualization) (Mitchell, 2018). None of this is to say that complex applications can not be created with Shiny, just that it may not be the most mature solution for the task.

While Shiny will undoubtedly continue to evolve in much the same way as R has, and many issues today will be gone tomorrow, a number of well-established software development paradigms must be diverged from:

- Shiny actively encourages the use of single-file applications, generally referring to this singular file as `app.R` (Core Team, 2017). Defining everything in a singular file works well for prototypes, but quickly falls apart as an application grows and increases in complexity. In general, code is compartmentalised into files which contain the logic for a single component. By allowing a single file to grow monolithic in size, code readability and re-usability is challenged, consequently making it harder to add additional components in the future (Fayram, 2011).
- Shiny insists on a reactive data-driven model over the more traditional and common event-driven model. While not necessarily a flaw in and of itself, many novice developers consider reactivity in programming to be a non-trivial concept (Fayram, 2011). Considering that Shiny, by nature, is aimed towards data analysts rather than computer scientists, it can increase the initial difficulty hurdle that beginners have to overcome. To further this issue, a bug has existed in RStudio since at least February 2018 that prevents automatic reloading from working with sourced files. When using multiple files like this, the server needs to be manually stopped and restarted between every change, making for a tedious development cycle. Concerns on the subject have not been addressed by either the RStudio or Shiny core developers (Hansen, 2018).

Both of the above points begin to cause major issues when put together. Encouragement of singular source files results in code quickly becoming unruly, threatening flexibility. This heightened complexity of source code will invariably be replicated within the reactive dependency graph, Shiny's internal mapping of reactive nodes and their relationships. In the event that something is not working as expected, RStudio provides little to no internal tools for debugging this graph. A new addition to CRAN in the form of **reactlog** (Schloerke and Cheng, 2019) is a first attempt to address this issue, which usually forces the developer to painstakingly debug the graph by hand. As the application becomes increasingly complex, this process gets closer and closer to impossible. Many of these cases are not as yet documented due to Shiny being a burgeoning technology, and to the best of our knowledge, this is the most in-depth look at the challenges in the peer reviewed literature.

A package, **ShinyTester** (Kohli, 2017), was added to CRAN `https://cran.r-project.org/` early in 2017. While it provides a promising first approach to debugging tools for Shiny (such as the inclusion of a dependency graph visualiser), it unfortunately seems to have been abandoned. Tools like this would likely alleviate the above outlined concerns.

Data size

Shiny is designed foremost as a server technology, with applications intended to be used remotely with a stable internet connection (Core Team, 2017). Shiny applications must be built with upload and download requirements in the fore. While Shiny applications can be run locally, doing so requires a base level of knowledge of R that may make it a sub-optimal approach, and limits accessibility. This is comparable to how mobile applications are generally shipped as pre-compiled binaries, rather than as raw source that the user would need to compile and install manually. One of the

largest issues with this inherent reliance on connectivity is the need for data to be uploaded and downloaded. Since Shiny has no inbuilt data streaming functionality, it is not possible to work with parts of data while waiting for the rest to upload (Core Team, 2017). An entire transfer must be completed before the dataset is made available to the application. This forces the application to require pre-partitioned uploads, which may not be possible for all types of datasets.

It is quite common to see a dataset approaching gigabytes in size, especially prevalent in areas such as genomic sequencing, which is generally technically unrealistic for datasets of this size to be worked with remotely, and would include extra data costs. If a large amount of bandwidth was made available to a single user, this could open up your service to potential denial-of-service attacks by malicious entities (Cloudflare, 2019). Furthermore, it may be legally unrealistic in terms of data ownership. Users are often uncomfortable providing sensitive data to unknown receivers, as there is no way for a Shiny app to prove that its not storing uploaded information permanently for the developer's own academic or financial gain (Kacha and Zitouni, 2018).

Literature analysis

Complexity barrier

The pattern of peer reviewed work, as shown in Figure 1, Figure 2, and Figure 3 shows that Bioinformatics is a popular and growing area for Shiny apps. Areas that traditionally have a lower focus on data analysis skills for researchers, such as Biological Sciences, Education and Index Medicus, appear to have higher usage levels. In the current literature Shiny is primarily used as a delivery/visualisation tool, and is not the focus with many papers referencing the use of Shiny but not discussing the merits. This trend becomes obvious in more recent papers, with much of the best discussion occurring in earlier papers.

To investigate the uptake of Shiny we must first understand some of the factors that determine the uptake of innovation. These are stated by Rogers (2001) as: (a) relative advantage, (b) compatibility, (c) complexity, (d) trialability and (e) observability. Rogers (2001) defined complexity as

> *...the degree to which an innovation is perceived as difficult to understand and use.*

Analysis methods are a non-trivial skill and the complexity of new methodologies in data analysis are a major hurdle for their uptake in fields such as biology and agriculture (DePalma et al., 2017). To drive innovation and uptake, tools must be accessible and usable by all interested parties (Jahanshiri and Shariff, 2014; Klein et al., 2017). Moraga (2017) noted in the area of public health, that while there had been progress in methodology and analysis

> *...these methods are still inaccessible for many researchers lacking the adequate programming skills to effectively use the required software.*

The first peer reviewed Shiny publications appeared in 2013, with the first two dissertations contributing the most to this discussion, as they give a glimpse to the vast potential for Shiny. The first dissertation using Shiny was published by

DePalma (2013) which allowed non-computer literate clinicians the ability to harness powerful statistical methodologies in a robust framework, to conduct antimicrobial susceptibility tests which determine an unknown pathogens susceptibility to various antibiotics. DePalma (2013) noted that previously new methods have not been adopted due to

> *...various computational difficulties and an absence of easy to use software for clinicians.*

Complex methodologies were able to be immediately used by end users without an assumption of computational skills, to inform important medical checks. This direct transfer of method is a concrete example of how Shiny is able to make complex research available to all interested parties, regardless of knowledge level. Specialised applications such as this would be difficult and costly to create without Shiny, and without general use software the advanced methodology would stall in uptake due to complexity barriers. This was later followed up with dBETS (diffusion Breakpoint Estimation Testing Software) by DePalma et al. (2017) who once again acknowledged

> *...the computational complexities associated with these new approaches has been a significant barrier for clinicians.*

Shiny is a potential solution to the barrier of complexity for the uptake of new methodologies.

Cross collaboration and dialogue

Cross collaboration between researchers and the easy dissemination of results is key to external validity (Munafò et al., 2017). Shiny promotes collaboration by allowing people with varying skill levels access to more complex methodologies. This has a flow on benefit to promote the collaboration of practitioners with researchers, or field researches with theorists, in order to create specialised, fit for purpose applications. This is illustrated in a paper by Wages and Petroni (2018) which designs and conducts Phase 1 dose finding trials using the continual reassessment method, and was noted to

> *...facilitate more efficient collaborations within study teams.*

Klein et al. (2017) underscores the requirement that

> *...facilitating the deployment of web applications for data analysis is important to promote collaboration within the scientific community and between scientists and stakeholders.*

Further examples of Shiny being used to open discussion by using apps to bring relevant parties with differing skills sets into collaboration include Díaz-Gay et al. (2018) who stated

> *...analysis of somatic mutational signatures remains currently inaccessible for a substantial proportion of the scientific community.*

As well as Whateley et al. (2015) who noted the knowledge gap between relevant parties and

> *...demonstrates the use of the Shiny web framework to bridge that gap, allowing for collaborative development of web tools that can be coded in the widely-used and free R statistical computing language.*

The ability to bridge the gap between researchers and the tools required for their data analysis was mentioned by Chen et al. (2018) in the context of environmental DNA. eDNA is becoming an essential tool in ecology and conservation biology and is utilised by a range of people with varying skill levels with Kandlikar et al. (2018) stating

> *Results from eDNA analyses can engage and educate natural resource managers, students, community scientists and naturalists, but without significant training in bioinformatics, it can be difficult for this diverse audience to interact with eDNA results.*

Shiny allows discipline specialists outside of computer science to code their own apps, bridging the skill gap for other researchers (Niu, 2017). This was demonstrated by an app called Armadillo Mapper (Feng et al., 2017), which was designed specifically to decrease the time between synthesising distributional knowledge on a computer and carrying out conservation efforts in the field. This encourages those without the resources to conduct their own analysis, to closely collaborate with analysts to create specialist applications. Rather than sending final data to an analyst for analysis, discussion and collaboration is encouraged at the beginning of an experiment. This enables low quality data due to issues such as pseudo-replication, low power and confounding variables to be avoided at the design stage rather than the analysis stage.

Flexibility to link other software

Shiny has the flexibility to bridge the gap between specialised data gathering tools and available software. A Shiny app accompanying the R package **rHyperSpec** (Laney, 2013) was created to take complex data generated by hyperspectral cameras and link the data to available software packages in response to the problem of

> *...few free, open-source software packages that enable researchers to easily process and analyse such data in a manner that maximizes inter-comparison between studies.*

This showcases the flexibility of Shiny applications being able to upload information in various formats, make appropriate changes, and output the data in a form usable by another, completely independent piece of equipment/software. Previously there were precious few options to link independent software/equipment, especially without breaching warranty restrictions. Shiny shows tremendous flexibility in working with existing infrastructure to help decrease costs, especially when technologies are in their infancy.

Shiny gains flexibility and customisability directly from R. One of Shiny's most useful abilities is to wrap existing, or new, R packages for general consumption. Beck (2014) created an app called Seed which bundled several R packages together and used Shiny to host them on the web allowing

> *...user's access to powerful R based functions and libraries through a simple user interface.*

In the Precision Agriculture (PA) space, farmers have access to a multitude of proprietary sensors, few of which can be linked directly to analysis tools (Jayaraman et al., 2016). Shiny's highly customisable framework facilitates the linking of several pieces of independent software, and can avoid manual data wrangling and transfer. As an example, Jahanshiri and Shariff (2014) took data from existing PA sensors and utilised R functions for its analysis and visualisation of results. Shiny has proved more than useful in the results visualising area, with packages such as **ShinyStan** (Gabry et al., 2018) created in order to visualise modelling parameters and results from MCMC simulations.

Generalising complex methodologies

R packages can be thought of as a level of abstraction down from the mathematical theory, as the packages can be used by those without a need to have full understanding of the methodology. Shiny can be thought of as another level of abstraction down again, as R packages can be utilised, without needing an understanding of R itself. This ability to generalise analysis methodologies makes Shiny available to any interested party, and is the mechanism that drives flexibility, dialogue and cross collaboration.

There is an overarching requirement when making tools available to a broader audience to ensure correct methodology. The first example of using Shiny to guide and educate the user came from Assaad et al. (2014) who created two Shiny apps intended to allow Microsoft Word users access to One Way Anova analysis and post hoc tests. The app gave instructions to guide users through the process, which greatly simplified the common statistical test, whilst promoting proper statistical methodology. A real world example of protecting the end user comes from Hsu et al. (2018), who created an app for proper randomisation when allocating participants to a three-armed, double-blinded, randomized controlled trial (RCT) for depression. One critical characteristic of the app was to ensure mistakes were not made when properly balancing strata. A fail safe against experimental error was employed by not allowing participants to have their experimental ID overwritten, which means that any accidental changes after treatment has begun would not impact on the treatment received.

Generalised applications must be flexible to differing individual parameters. Shiny makes it a trivial task to allow parameters of a methodology to be changed depending on individual circumstances. Shiny wrapped simulations were used to explore humanitarian response and financial institution resiliency for earthquake risk in Indonesia, with Hartell (2014) allowing the simulation to be tweaked by individuals so that adjustments to calibration parameters could be made based on specific interests or circumstances. Other apps that allowed the user to specify parameters were created by Zhou et al. (2014) for detecting differential expression in RNA sequencing and Yin (2014) who utilised Bayesian statistical modelling to investigate the networks of epidemics transmission.

Shiny makes complex methodologies accessible to those who would previously not be part of the conversation, most likely due to a lack of theoretical study, or lack of familiarity with coding or analysis programs. Shiny was explicitly noted to help increase engagement by LaZerte et al. (2017), who created FeedR in order to record and visualise RFID data from ecological studies. The huge amount of data from RFID quickly becomes overwhelming and requires specialist methods to cope. The FeedR Shiny app was created to wrap the paired R package in order that

> *...this framework will become a meeting point for science, education and community awareness...we aim to inspire citizen engagement while simultaneously enabling robust scientific analysis (LaZerte et al., 2017).*

Responsible and open research

Reproducibility of research is a critical cornerstone of responsible research practices. Studies, such as Munafò et al. (2017), have indicated that reproducibility is not at high enough levels, with results of a survey conducted by Baker (2016) and published in Nature found

> *...more than 70% of researchers have tried and failed to reproduce another scientist's experiments and more than half have failed to reproduce their own experiments.*

Eight practices are argued for by Munafò et al. (2017), which includes promoting transparency and open science to increase reproducibility. Open source software such as Shiny can aid these objectives by creating a vessel to preserve code, and allow a greater number of interested parties to critically evaluate methodologies and results.

One benefit of Shiny wrapped code is that methodology comparisons become much easier to conduct. Methodologies wrapped in Shiny applications can be compared on a known data set under various conditions by the end user. This is a powerful tool in the advancement of reproducible research. Shiny was explicitly used in a dissertation by Parvandeh (2018) as a vessel to show the strategy and to create reproducibility of results enhancing responsible and reproducible research goals.

A Shiny app, or at least the code behind it is enduring. A paper from Sieriebriennikov et al. (2014) included a Shiny application named Nematode Indicator Joint Analysis (NINJA) 2.0, to automate manual calculations previously carried out using spreadsheet software, which is time consuming and prone to errors. The aim for NINJA to remain freely accessible was validated when it was later used by Burkhardt et al. (2019) to aid nematode calculations in semi-arid wheat systems, 5 years after its release. This suggests that maintaining a Shiny application is not overly difficult. The benefit of Shiny's easy maintenance and updating was mentioned for the first time in a dissertation by Niu (2017), which highlighted the fact that only the source code requires changing without having to download patches or modify individual applications.

Shiny also appeared in conjunction with machine learning to explore early phase drug discovery processes (Korkmaz et al., 2015), with Wojciechowski et al. (2015) noticing the power of Shiny to disseminate the results of research, stating

> *Interactive applications, developed using Shiny for the R programming language, have the potential to revolutionize the sharing and communication of pharmacometric model simulations.*

Free and open source software is ideally suited to disseminating the products of research (LaZerte et al., 2017), which drives collaboration and was noted to encourage local and direct monitoring of environmental data in Kenya (Mose et al., 2017). LaZerte et al. (2017) also found that Shiny's open source nature has another important benefit which

> *...reduces financial barriers to its use and the open-source aspect permits and encourages collaboration which can result in better, more powerful software.*

Cross collaboration and use of open source Shiny will hopefully also help drive data sharing. Yi et al. (2017) noted the utility and importance of data sharing promoted by Shiny applications, which is also one of the key recommendations by Munafò et al. (2017) in order to drive transparency and openness, and is currently a policy by *Science* and *Springer Nature* journals.

An educational tool

The strengths shown by Shiny seems to fit very well in the educational sector and it was no surprise that Shiny has been used as a teaching aid in order to get complex ideas across to students (Williams and Williams, 2018). Educational tools such as those by Arnholt (2018) help teach the concept of power in hypothesis tests, with Williams and Williams (2018) creating a similar application for confidence intervals, and an app by Courtney and Chang (2018) which normalises large datasets and allows students to explore the results of differing transformations. There are other benefits to using Shiny in the education sector. Kandlikar et al. (2018) created the Shiny app ranacapa and found that

> *A key benefit of using ranacapa was that despite having no prior bioinformatics experience, students could begin exploring the biodiversity in their samples in a matter of minutes by using the online instance of the Shiny app.*

This had the flow on effect of allowing teachers to focus more on the theory instead of the inevitable problems when teaching new, more complex software and provided a useful aid to self-learning (Kandlikar et al., 2018).

Conclusion

This review examined Shiny in peer reviewed publications from 2012 to 2018 and mapped the growth through various research fields. A subset of 600 papers were used to inform the bulk of the paper, with the authors personal experiences of Shiny included. While Shiny is not a *silver bullet* solution to issues in the research field, it confers the ability for specialised applications to be created cheaply and easily, such that any level of end user maybe included, no matter the complexity level of the methodology. This primary benefit creates a direct pathway for new findings to be rapidly incorporated into established work flows. The flexibility of Shiny means that apps can be tailored to exact specifications in all regards, with changes and maintenance of the app made relatively easy as an ongoing product of consultation further promoting collaboration. If an app is considered worthwhile adopting to an existing work flow, widespread adoption across an entire workplace is as simple as sharing the web address. This will have the inevitable knock on effects of allowing fewer people to do more, which will necessarily mean existing jobs have the potential of becoming obsolete. The argument that other jobs will be created is true, but not necessarily within the same sector, or for the people whose job has become obsolete. As we progress further into this technological world, this argument will require mature debate and nuance to be resolved.

In the current literature, Shiny has been used primarily as a visualisation and dissemination tool, with many papers not exploring the concurrent benefits and

challenges mentioned in this review. One benefit identified in the literature is the opportunity to increase high value dialogue between people with different skill sets. For example, field researchers, primary producers or marketers are able to sit down with theoretical researchers/consultants to create highly customised applications for up-coming experiments or daily work. Code published as a Shiny application has the useful attribute of making methodology comparisons easy, which promotes reproducible research and best practice standards.

With the ability to accelerate access to data analysis techniques comes the paramount issue of data security for those not familiar with web protocols. It is essential for those who host web based applications to become knowledgeable in this area. Web security protocols are likely to be a new skill set for many R programmers, and a non-trivial task potentially constituting a bottle neck for widespread Shiny uptake.

While the use of Shiny apps require minimal experience with computers, the creation of a Shiny application is a different story. The lack of debugging tools, the encouragement of single file applications, and the current implementation of the reactive data-driven model will limit the complexity of future applications.

Other open source and proprietary options are currently available, however, Shiny's flexibility, customisability, and low cost is highly desirable. Open source software comes with a minimum knowledge requirement barrier to entry, and proprietary software can be expensive and inflexible to changing situations and circumstances. Maintenance is required with Shiny, although it is limited to updating code when R packages or dependencies change, and can be done via the source code for all users.

Shiny is one of the better tools available if one is an existing R programmer given its inherited scope from R. It helps promotes open and reproducible research, and offers a real pathway to making complicated methodologies usable to those outside of research. The ability to provide an avenue to increase high value collaboration and dialogue between interested parties with differing skills sets make Shiny a tool worth exploring.

Acknowledgements

We would like to gratefully acknowledge the scholarship for the M.Phil program of the first author from the Grains Research and Development Corporation (GRDC) Australia.

Bibliography

Amazon. Amazon EC2 Pricing. https://aws.amazon.com/ec2/pricing/, 2019. [p8]

D. Arend. Minitab 17 Statistical Software. Minitab, Inc., 2010. [p1]

A. T. Arnholt. Using a Shiny app to teach the concept of power. *Teaching Statistics*, Mar. 2018. ISSN 1467-9639. doi: 10.1111/test.12186. [p15]

H. I. Assaad, L. Zhou, R. J. Carroll, and G. Wu. Rapid publication-ready MS-Word tables for one-way ANOVA. *SpringerPlus*, **3**(1):474, Aug. 2014. ISSN 2193-1801. doi: 10.1186/2193-1801-3-474. [p13]

M. Baker. 1,500 scientists lift the lid on reproducibility. *Nature News*, **533**(7604):452, May 2016. doi: 10.1038/533452a. [p2, 14]

D. Beck. *Investigating the Use of Classification Models to Study Microbial Community Associations with Bacterial Vaginosis*. Ph.D., University of Idaho, United States – Idaho, 2014. [p12]

C. Beeley. *Web Application Development with R Using Shiny*. Olton: Packt Publishing Ltd, first edition, 2013. [p3]

A. Burkhardt, S. S. Briar, J. M. Martin, P. M. Carr, J. Lachowiec, C. Zabinski, D. W. Roberts, P. Miller, and J. Sherman. Perennial crop legacy effects on nematode community structure in semi-arid wheat systems. *Applied Soil Ecology*, Jan. 2019. ISSN 0929-1393. doi: 10.1016/j.apsoil.2018.12.020. [p14]

W. Chang, J. Cheng, J. J. Allaire, Y. Xie, and J. McPherson. **Shiny**: Web Application Framework for R, 2018. [p6]

J. Charpentier. Web application Security. Technical Report Network Project, 7.5 hp, Halmstad University, Halmstad, Sweeden, Jan. 2013. [p7, 8]

Z. Chen, Y. Zheng, Z. Wang, M. Kutner, W. J. Curran, and J. Kowalski. Interactive calculator for operating characteristics of phase I cancer clinical trials using standard 3+3 designs. *Contemporary Clinical Trials Communications*, **12**:145–153, Nov. 2018. ISSN 2451-8654. doi: 10.1016/j.conctc.2018.10.006. [p12]

J. Cheng. Shiny - Modularizing Shiny app code. https://shiny.rstudio.com/articles/modules.html, June 2017. [p8]

F. Cheung, G. Fantoni, M. Conner, B. A. Sellers, Y. Kotliarov, J. Candia, K. Stagliano, and A. Biancotto. Web Tool for Navigating and Plotting SomaLogic ADAT Files. *Journal of Open Research Software*, **5**(1), Sept. 2017. ISSN 2049-9647. [p8]

Cloudflare. What Is a Distributed Denial-of-Service (DDoS) Attack? https://www.cloudflare.com/en-au/learning/ddos/what-is-a-ddos-attack/, 2019. [p10]

R. Core Team. RStudio Pricing. https://www.rstudio.com/pricing/, Nov. 2012. [p7, 8]

R. Core Team. Shiny Welcome to Shiny. https://shiny.rstudio.com/tutorial/written-tutorial/lesson1/, 2017. [p9, 10]

M. G. R. Courtney and K. C. Chang. Dealing with non-normality: An introduction and step-by-step guide using R. *Teaching Statistics*, **40**(2):51–59, 2018. ISSN 1467-9639. doi: 10.1111/test.12154. [p15]

G. DePalma. *Disk Diffusion Breakpoint Determination Using a Bayesian Nonparametric Variation of the Errors-in-Variables Model*. Ph.D., Purdue University, United States – Indiana, 2013. [p2, 11]

G. DePalma, J. Turnidge, and B. A. Craig. Determination of disk diffusion susceptibility testing interpretive criteria using model-based analysis: Development and implementation. *Diagnostic Microbiology and Infectious Disease*, **87**(2):143–149, Feb. 2017. ISSN 0732-8893. doi: 10.1016/j.diagmicrobio.2016.03.004. [p10, 11]

M. Díaz-Gay, M. Vila-Casadesús, S. Franch-Expósito, E. Hernández-Illán, J. J. Lozano, and S. Castellví-Bel. Mutational Signatures in Cancer (**MuSiCa**): A web application to implement mutational signatures analysis in cancer samples. *BMC Bioinformatics*, **19**(1):224, June 2018. ISSN 1471-2105. doi: 10.1186/s12859-018-2234-y. [p11]

D. Donoho. 50 Years of Data Science. *Journal of Computational and Graphical Statistics*, 26(4):745–766, Oct. 2017. ISSN 1061-8600. doi: 10.1080/10618600.2017.1384734. [p1]

B. Dwivedi and J. Kowalski. **shinyGISPA**: A web application for characterizing phenotype by gene sets using multiple omics data combinations. *PLoS ONE*, **13** (2), Feb. 2018. ISSN 1932-6203. doi: 10.1371/journal.pone.0192563. [p8]

D. Fayram. Functional Programming Is Hard, That's Why It's Good, Aug. 2011. [p8, 9]

X. Feng, M. C. Castro, E. Linde, and M. Papeş. **Armadillo Mapper**: A Case Study of an Online Application to Update Estimates of Species' Potential Distributions. *Tropical Conservation Science*, **10**, Jan. 2017. ISSN 1940-0829. doi: 10.1177/1940082917724133. [p12]

J. Gabry, S. D. Team, M. Andreae, M. Betancourt, B. Carpenter, Y. Gao, A. Gelman, B. Goodrich, D. Lee, D. Song, and R. Trangucci. Shinystan: Interactive Visual and Numerical Diagnostics and Posterior Analysis for Bayesian Models, May 2018. [p13]

S. X. Ge, E. W. Son, and R. Yao. **iDEP**: An integrated web application for differential expression and pathway analysis of RNA-Seq data. *BMC Bioinformatics*, **19**, Dec. 2018. ISSN 1471-2105. doi: 10.1186/s12859-018-2486-6. [p8]

G. Grolemund. Shiny - How to understand reactivity in R. https://shiny.rstudio.com/articles/understanding-reactivity.html, May 2015. [p7]

A. Gunuganti. Application Development Framework for R/Shiny. In *PharmaSUG 2018 Conference Proceedings*, volume AD-24, page 9, Seattle, Apr. 2018. PharmaSUG. [p2]

J. Guo. *Developing a Visualization Tool for Unsupervised Machine Learning Techniques on *Omics Data*. Master's, University of Washington, United States – Washington, 2018. [p8]

K. Hansen. Rstudio - Reload Shiny App when using source'ed modules without restart. https://stackoverflow.com/questions/50169896/reload-shiny-app-when-using-sourceed-modules-without-restart, May 2018. [p9]

J. Hartell. *Earthquake Risk in Indonesia: Parametric Contingent Claims for Humanitarian Response and Financial Institution Resiliency*. Ph.D., University of Kentucky, United States – Kentucky, 2014. [p13]

K. J. Hsu, K. Caffey, D. Pisner, J. Shumake, S. Risom, K. L. Ray, J. A. J. Smits, D. M. Schnyer, and C. G. Beevers. Attentional bias modification treatment for depression: Study protocol for a randomized controlled trial. *Contemporary Clinical Trials*, **75**: 59–66, Dec. 2018. ISSN 1551-7144. doi: 10.1016/j.cct.2018.10.014. [p13]

E. Jahanshiri and A. R. M. Shariff. Developing web-based data analysis tools for precision farming using R and Shiny. *IOP Conference Series: Earth and Environmental Science*, **20**(1), 2014. ISSN 1755-1315. doi: 10.1088/1755-1315/20/1/012014. [p10, 13]

P. P. Jayaraman, A. Yavari, D. Georgakopoulos, A. Morshed, and A. Zaslavsky. Internet of Things Platform for Smart Farming: Experiences and Lessons Learnt. *Sensors*, **16**(11):1884, Nov. 2016. doi: 10.3390/s16111884. [p13]

L. Kacha and A. Zitouni. An Overview on Data Security in Cloud Computing. pages 250–261, Sept. 2018. ISBN 978-3-319-67617-3. doi: 10.1007/978-3-319-67618-0_23. [p10]

G. S. Kandlikar, Z. J. Gold, M. C. Cowen, R. S. Meyer, A. C. Freise, N. J. Kraft, J. Moberg-Parker, J. Sprague, D. J. Kushner, and E. E. Curd. **Ranacapa**: An R package and Shiny web app to explore environmental DNA data with exploratory statistics and interactive visualizations. *F1000Research*, **7**, Nov. 2018. ISSN 2046-1402. doi: 10.12688/f1000research.16680.1. [p12, 15]

T. Klein, A. Samourkasidis, I. N. Athanasiadis, G. Bellocchi, and P. Calanca. **webXTREME**: R-based web tool for calculating agroclimatic indices of extreme events. *Computers and Electronics in Agriculture*, **136**:111–116, Apr. 2017. ISSN 0168-1699. doi: 10.1016/j.compag.2017.03.002. [p10, 11]

A. Kohli. **ShinyTester**:Functions to Minimize Bonehead Moves While Working with 'shiny', Feb. 2017. [p9]

S. Korkmaz, G. Zararsiz, and D. Goksuluk. **MLViS**: A Web Tool for Machine Learning-Based Virtual Screening in Early-Phase of Drug Discovery and Development. *PLoS One; San Francisco*, **10**(4), Apr. 2015. doi: http://dx.doi.org.proxy.library.adelaide.edu.au/10.1371/journal.pone.0124600. [p14]

S. Landau and B. Everitt. *A Handbook of Statistical Analyses Using SPSS*. Chapman & Hall/CRC, Boca Raton, 2004. ISBN 978-1-58488-369-2. [p1]

C. M. Laney. *Toward New Data and Information Management Solutions for Data-Intensive Ecological Research*. Ph.D., The University of Texas at El Paso, United States – Texas, Dec. 2013. [p12]

S. E. LaZerte, M. W. Reudink, K. A. Otter, J. Kusack, J. M. Bailey, A. Woolverton, M. Paetkau, A. de Jong, and D. J. Hill. Feedr and animalnexus.ca: A paired R package and user-friendly Web application for transforming and visualizing animal movement data from static stations. *Ecology and Evolution*, **7**(19):7884–7896, 2017. ISSN 2045-7758. doi: 10.1002/ece3.3240. [p2, 13, 14]

J. Li, B. Cui, Y. Dai, L. Bai, and J. Huang. BioInstaller: A comprehensive R package to construct interactive and reproducible biological data analysis applications based on the R platform. *PeerJ*, **6**, Oct. 2018. ISSN 2167-8359. doi: 10.7717/peerj.5853. [p2]

Microsoft. Pricing – Linux Virtual Machines | Microsoft Azure. https://azure.microsoft.com/en-au/pricing/details/virtual-machines/linux/, 2019. [p8]

E. Mitchell. Shiny applications without Shiny, Aug. 2018. [p8]

C. Moler and Mathworks. MATLAB 8.0 and Statistics Toolbox 8.1. The MathWorks, Inc., 2012. [p1]

K.-M. Moon. *Learn Ggplot2 Using Shiny App*. Number 2197-5736 in Use R! Springer, 2016. [p3]

P. Moraga. SpatialEpiApp: A Shiny web application for the analysis of spatial and spatio-temporal disease data. *Spatial and Spatio-temporal Epidemiology*, **23**:47–57, Nov. 2017. ISSN 1877-5845. doi: 10.1016/j.sste.2017.08.001. [p10]

V. N. Mose, D. Western, and P. Tyrrell. Application of open source tools for biodiversity conservation and natural resource management in East Africa. *Ecological Informatics*, Sept. 2017. ISSN 1574-9541. doi: 10.1016/j.ecoinf.2017.09.006. [p14]

M. R. Munafò, B. A. Nosek, D. V. M. Bishop, K. S. Button, C. D. Chambers, N. Percie du Sert, U. Simonsohn, E.-J. Wagenmakers, J. J. Ware, and J. P. A. Ioannidis. A manifesto for reproducible science. *Nature Human Behaviour*, **1**(1), Jan. 2017. ISSN 2397-3374. doi: 10.1038/s41562-016-0021. [p11, 14, 15]

B. Niu. *Mass Spectrometry-Based Structural Proteomics: Methodology and Application of Fast Photochemical Oxidation of Proteins (FPOP)*. Ph.D., Washington University in St. Louis, United States – Missouri, 2017. [p12, 14]

S. Parvandeh. *Epistasis Network and Machine Learning Methods for the Analysis of Biological Large Data*. Ph.D., The University of Tulsa, United States – Oklahoma, 2018. [p14]

R. Payne, D. Murray, S. Harding, D. Baird, and D. Soutar. GenStat. VSN International, 2007. [p1]

M. Rogers. Evolution: Diffusion of Innovations. In N. J. Smelser and P. B. Baltes, editors, *International Encyclopedia of the Social & Behavioral Sciences*, pages 4982–4986. Pergamon, Oxford, Jan. 2001. ISBN 978-0-08-043076-8. doi: 10.1016/B0-08-043076-7/03094-1. [p10]

B. Schloerke and J. Cheng. **Reactlog**: Reactivity Visualizer for 'shiny', Mar. 2019. [p9]

B. Sieriebriennikov, H. Ferris, and R. G. M. de Goede. NINJA: An automated calculation system for nematode-based biological monitoring. *European Journal of Soil Biology*, **61**:90–93, Mar. 2014. ISSN 1164-5563. doi: 10.1016/j.ejsobi.2014.02.004. [p14]

R. C. Team. ***R**: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019. [p2]

N. A. Wages and G. R. Petroni. A web tool for designing and conducting phase I trials using the continual reassessment method. *BMC Cancer*, **18**, Feb. 2018. ISSN 1471-2407. doi: 10.1186/s12885-018-4038-x. [p11]

S. Whateley, J. D. Walker, and C. Brown. A web-based screening model for climate risk to water supply systems in the northeastern United States. *Environmental Modelling & Software*, **73**:64–75, Nov. 2015. ISSN 1364-8152. doi: 10.1016/j.envsoft.2015.08.001. [p11]

H. Wickham, W. Chang, L. Henry, T. L. Pedersen, K. Takahashi, C. Wilke, and K. Woo. **Ggplot2**: Create Elegant Data Visualisations Using the Grammar of Graphics, 2018. [p3]

I. J. Williams and K. K. Williams. Using an R shiny to enhance the learning experience of confidence intervals. *Teaching Statistics*, **40**(1):24–28, Mar. 2018. ISSN 1467-9639. doi: 10.1111/test.12145. [p15]

J. Wojciechowski, A. M. Hopkins, and R. N. Upton. Interactive Pharmacometric Applications Using R and the Shiny Package. *CPT: Pharmacometrics & Systems Pharmacology*, **4**(3):146–159, Mar. 2015. ISSN 2163-8306. doi: 10.1002/psp4.21. [p14]

L. Yi, H. Pimentel, and L. Pachter. Zika infection of neural progenitor cells perturbs transcription in neurodevelopmental pathways. *PLoS One; San Francisco*, **12**(4), Apr. 2017. doi: http://dx.doi.org.proxy.library.adelaide.edu.au/10.1371/journal.pone.0175744. [p15]

J. Yin. *Bayesian Statistical Modeling in Epidemics and the Contact Networks That Transmit Them*. Ph.D., The University of Iowa, United States – Iowa, May 2014. [p13]

X. Zhou, H. Lindsay, and M. D. Robinson. Robustly detecting differential expression in RNA sequencing data using observation weights. *Nucleic Acids Research*, **42**(11), June 2014. ISSN 0305-1048. doi: 10.1093/nar/gku310. [p13]

Appendix

This appendix contains the table of abbreviations for journal titles and subject headings.

Table 1: Table of journal abbreviations

Journal	Abbreviation
2nd Symposium On Lapan Ipb Satellite Lisat For Food Security And Environmental Monitoring	Food Sec Envir
Bioinformatics	Bioinfo
Bioinformatics (Oxford England)	Bioinfo (OE)
Bmc Bioinformatics	BMC Bioinfo
Bmc Cancer	BMC Cancer
Environmental Earth Sciences	Envir Earth Sci
Environmental Modelling And Software	Envir Mod Soft
F1000research	F1000
Frontiers In Psychology	Front Psych
Gigascience	Giga
Iop Conference Series Earth And Environmental Science	Earth Envir Sci
Journal Of Pharmacokinetics And Pharmacodynamics	Pharma
Journal Of Physics Conference Series	Physics
Lecture Notes In Artificial Intelligence	AI
Natural Hazards	Nat Haz
Nature Communications	Nat Com
Nucleic Acids Research	Nuc Acids Res
Peerj	Peerj
PloS ONE	PloS ONE
Procedia Environmental Sciences	Envir Sci
R Journal	R Journal
Scientific Reports	Sci Rep
Source Code For Biology And Medicine	Bio Med
Springerplus	Springerplus
Statistics In Medicine	Stat Med
Studies In Health Technology And Informatics	Health Tech Info
Wellcome Open Research	Well Open Res
Workshop And International Seminar On Science Of Complex Natural Systems	Complex Nat Sys

Table 2: Table of abbreviations for subject tag

Subject	Abbreviation
Agricultural Biological Sciences	Ag Biol Sci
Agriculture Multidisciplinary	Ag Multi
Arts Humanities	Art Hum
Automation Control Systems	Auto Cont Sys
Biochemical Research Methods	Biochem Res Meth
Biochemistry Genetics Molecular Biology	Biochem Gen Molec Biol
Biochemistry Molecular Biology	Biochem Molec Biol
Biotechnology Applied Microbiology	Biotech App Micro
Business Management Accounting	Bus Man Acc
Chemical Engineering	Chem Eng
Chemistry	Chemistry
Communication and the Arts	Comm & Arts
Computational Biology	Comp Biology
Computer Science	Comp Sci
Computer Science Artificial Intelligence	Comp Sci AI
Computer Science Information Systems	Comp Sci Info Sys
Computer Science Interdisciplinary Applications	Comp Sci Inter App
Computer Science Theory Methods	Comp Sci Theor Meth
Decision Sciences	Dec Sci
Earth Planetary Sciences	Earth Plan Sci
Education Scientific Disciplines	Ed Sci Disc
Energy	Energy
Engineering	Engineering
Engineering Electrical Electronic	Eng Elec Elct
Engineering Environmental	Eng Env
Environmental Science	Envir Sci
Environmental Sciences	Env Sci
Evolutionary Biology	Evol Biol
Genetics Heredity	Genet Hered
Health Care Sciences Services	Health Care Sci Ser
Health Professions	Health Prof
Immunology Microbiology	Immun Micro
Materials Science	Mat Sci
Mathematical Computational Biology	Math Comp Biol
Mathematics	Mathematics
Medical Informatics	Med Info
Medicine	Medicine
Medicine Research Experimental	Med Res Exp
Multidisciplinary	Multidisciplinary
Multidisciplinary Sciences	Multi Disc Sci
Neuroscience	Neuro
Oncology	Oncology
Pharmacology Pharmacy	Pharm Pharmacy
Pharmacology Toxicology Pharmaceutics	Pharm Tox
Physics Astronomy	Phys Astro
Psychology	Psychology
Public Environmental Occupational Health	Pub Envir Occ
Remote Sensing	Rem Sens
Social Sciences	Soc Sci
Statistics Probability	Stat Prob
Veterinary	Vet

Peter Kasprzak
University of Adelaide
School of Agriculture Food and Wine, PMB 1, Glen Osmond, SA 5064
Australia
peter.kasprzak@adelaide.edu.au

Lachlan Mitchell
University of Adelaide
School of Agriculture Food and Wine, PMB 1, Glen Osmond, SA 5064
Australia
lachlan.mitchell@icloud.com

Olena Kravchuk
University of Adelaide
School of Agriculture Food and Wine, PMB 1, Glen Osmond, SA 5064
Australia
olena.kravchuk@adelaide.edu.au

Andy Timmins
University of Adelaide
School of Agriculture Food and Wine, PMB 1, Glen Osmond, SA 5064
Australia
andy.timmins@adelaide.edu.au