Using R in the Statistical Office: the experience of Statistics Netherlands and Statistics Austria

Alexander Kowarik
Statistics Austria

Mark van der Loo
Statistics Netherlands

ABSTRACT

Over the last decade the R language and statistical environment has received a surge in popularity. Indeed, R has become one of the central tools for modern statistics and data science. Slowly but certainly, statistical offices are introducing R as a valid tool for statistical production as well. The Austrian and Dutch offices were amongst the first national statistical institutes to approve R as a tool for production. The aim of this paper is to describe how R was introduced and currently used in our offices. On one hand, we focus on practical issues such as infrastructural considerations, the use of R-based software by non-R programmers, and update policies. On the other hand we describe the activities that were undertaken to educate new users in the use of R. We also discuss work that was contributed back to the R community in general and the official statistics community in particular. Finally, we discuss how collaboration and standardization take place in an open source environment.

Keywords*: R Software, Official Statistics*
JEL Classification*: C80, C88*

INTRODUCTION

The R language exists since 1993, and back then it was a new and exotic thing. Nowadays, at least for over a decade it is the dominant programming language and statistical software in academia in the area of Statistics. Indeed, IEEE (2017) now ranks R as the 6th most popular programming language worldwide. Similarly, the TIOBE (2017) index currently lists R as the number 15 language while the PYPL (2017) popularity index ranks R as the number 8 language. This is impressive especially when one considers that the special purpose language R competes in these rankings with general purpose languages such as Python, C, C# and Java. In each of these ranking, R has risen over the last years and has surpassed all other statistical languages, including SAS, in popularity.

It is natural to ask why R became so popular. Although formal research on this question appears to be lacking, a few properties and circumstances that make such a growth plausible readily present themselves. On one hand, R may be `riding the wave of data science' that is currently washing over industry, government and academia. R probably was simply there at the right place at the right time -- offering a wide range of statistical and data handling functionality with a convenient and programmable interface for free when data science took off as a field. Additionally, the choice to publish R as an open source tool was an important cornerstone for its success.

Before continuing however, it is worthwhile pointing out a few properties typical to R, its developers, and its community. First, R is developed by a dedicated core team with a strong focus on the software's correctness, stability, and backward compatibility. The R core team consists mostly academics working in the field of (computational) statistics --- a field that benefits greatly from the availability of such software. For users, this means that R can be relied upon both for functionality and API stability. Second, R is developed under the GNU general public license. This entails that anyone is permitted to copy, study, alter, and redistribute R and its source code. Third, R is extensible with code bundles ('R packages') that are easily exchanged. A valid R package is endowed with a description of its purpose, version, dependencies, and authorship. Every function exposed to the user must be described in a standardized manner for a package to be valid. Package authors can offer their work for publication on a central repository such as CRAN. At CRAN, packages are subjected to intense scrutiny, both prior to acceptance and afterwards. In particular, CRAN ensures that no package entering the repository breaks any other package that depends on it, directly or indirectly. Users can therefore rely on the fact that all of the 11.000+ packages currently on CRAN have a minimum amount of documentation and a certain amount of interface stability. Fourth, R has attracted attention from the industry, which lead to the development of many tools that facilitate the workflow from initial analyses to publishing results, deploying models, or sharing software/applications. Examples include integrated development environments, database connectivity tools, software test suites, visualisation tools, connectivity with web applications and more.

The growth of R has led to an active international (online) community of users and developers who efficiently exchange information via mailinglist, fora (stackoverflow), and social media (Twitter). There are several conference series devoted to the use of R. The most well-known include the useR!

conference, which is considered the main worldwide user meeting, the satRday initiative, the twice-a-year Effective Applications of the R Language (EARL) conference, and of course the 'use of R in official statistics' conference (uRos). R is widely supported by the industry with major software vendors offering an interfaces to and from R from their products. The software industry aims to advance R and its community by combining efforts through the R consortium, an organization that *'support[s] the worldwide community of users, maintainers and developers of R software'* (R consortium, 2015).

Still, the official statistics branch and especially national statistical institutes seem quite hesitant to make the switch from a traditional, often commercial, solution to an open-source alternative. Of course any switch of software product will demand resources to handle legacy code, staff (re) education, and so on. Reasons, particular to R or other open source products, seem to include the lack of formal procedures to handle free software with open source licenses and the perceived risk of working with software that is apparently not backed by commercial support.

The purpose of this paper is to discuss the experience of introducing and using R at the statistical office, from the perspective of two statistical institutes. Specific production steps or statistical products are not described in this paper. In Templ and Todorov (2016) you find an overview of R packages aligned to the general statistical business process model.

In the next section, we discuss how R was introduced into our offices as a valid tool for statistical production. After this we describe some developments in the infrastructure hosting the internal R distribution. In the fourth section we focus on how R packages are used at the statistical office and we provide a brief overview of the packages that have been contributed to the community. A special section is devoted the modes of collaboration that are common in a community-driven tool such as R. We conclude with a short summary and conclusion.

REMARKS ABOUT ADOPTING R AT THE STATISTICAL OFFICE

Austria

The process of adopting R as a standard tool at STAT was cumbersome and long, but we are now at stage with a lot of expertise and using R in different areas of the production process. Since 2004 it was used "unofficially" by one user at the methods unit, but new members of the methods unit soon joined him in using (and loving) the R environment. Basically, it was a grassroots movement and merely tolerated by the IT department, where still SAS was the only strategic statistical software. Soon after it was used to build our first

R package sdcMicro which was released to CRAN in May 2007 (VIM also in 2007 and sdcTable followed in 2009). Between 2004 and 2009 it was more or less tolerated that the methods unit used R and developed new methods with it, but it was not be used in production.

A wider audience got a first glimpse during the methodological training courses, where R was used since 2005 as a computer assisted training tool for visualizations, animations and to ask simple questions about the course content (see Dinges et al., 2011). The usage of the tool required no R knowledge but sparked the interest in several course participants.

In 2009 we got the official approval to use R, but only within the methods unit and we got our first server to use R on bigger problems, especially in the area of statistical disclosure control. Finally in 2013, R was approved as a standard tool for everyone at STAT and it could be used where deemed fit. In the same year a training programme for R was established. At first it was a week long course with everything in it from basic data manipulation, graphics to generating dynamic reports. The course programme was steadily improved and now consists of 6 modules. Over 100 participants took at least one of these courses. In addition to the training programme a wiki was set up, however it never really became lively and is not used anymore. To foster the exchange of knowledge and ideas we now have regular user group meetings (about twice a year). During these meetings a number of R users present their projects. About 40 employees use R at least once a week at the moment.

The methods unit, which was among the first user of R at STAT and has a lot of experience in using it, is providing support to all R users at STAT. The support requests are handled via the issue tracking system Jira. For IT-related issues, the methods unit interacts with the IT department through another Jira project, e.g. when a new version of the desktop R package is generated or some administrative tasks are necessary on the servers.

Netherlands

R was accepted as a standard (approved) tool for statistical production in 2010. The incentive for introducing the tool was the lack of a main tool or language for statistical programming at the office. At that time, tools such as Matlab, Stata, Ox, and Splus, were used in several niche places in the office while many production systems would rely on SPSS, standard office software, SN's own software modules (Bascula, Blaise), and custom-build software components for statistical production. After a preselection, R and Matlab were chosen as two of the main candidates to provide a scientific programming language. A detailed comparison of needs and features surfaced R as the preferred language. The fact that R is open source was one of the deciding factors.

Once it was decided that R would be introduced, a project was set up to decide on deployment, develop programming guides, and a wiki for R users. The fact that introduction of R was an `official project' with a dedicated project leader helped a tremendous amount in overcoming some organisational issues. For example, processes and contracts at the IT department were completely geared towards commercial products and vendors. There simply was no procedure to download and install an open source product for production. The fact that it was necessary to download R, CRAN and an interface separately (this was before RStudio, and Notepad++ was used with an R-interface) was also new. Finding solutions to such issues require some time and resources.

In the beginning, three different internal 'R distributions' were developed, distinguishing use for production, analyses and development. This practice that was eventually simplified into a single centralized distribution (to be discussed later). At the same time the initiative was taken to set up a local R user community. A few of the 'expert' programmers took it upon themselves to organise informal biweekly user meetings, set up trainings (twice per year) and to be available to help with operational questions. The driving aim was to have an internal community that mimics the international R community, in terms of openness and (informal) information exchange.

Currently, about two hundred of Statistics Netherlands' employees have followed the internal training course. Initially the course was aimed at statisticians who need to program a statistical production system but gradually the course was adapted to more ad-hoc users (analysts) and people who need to run and adapt scripts rather than develop them from scratch. More advanced (optional) workshops are taught occasionally as well to still serve the more advanced user community. A wiki was set up as well. It never took off in the sense of users contributing a lot of content. The wiki is still useful to collect answers to questions, workarounds, information on upgrade procedures, package installations and so on.

Statistics Netherlands has been decentralizing development of statistical systems over the past few years. In this context, R was only the first tool that has been introduced. Other tools that have been introduced since R include Python (2012-13), git (for version control, 2014-15) and Spark (2016). In all these cases, the importance of setting up a community in the way that was pioneered during the introduction of R was recognized. Explicit attention was paid to set up user meetings and other communication channels such as a community wiki during introduction of all these tools.

INFRASTRUCTURE

Austria

In the future R will be only available to users via dedicated R servers and the interface will be RStudio Server Pro. This setup allows for central maintenance of the R installation and for a very flexible approach to resource management. The users can interact with the server through the familiar interface of RStudio in the browser, so the training efforts are minimal when switching from RStudio Desktop.

Every user is limited in available resources, e.g. a user will begin with 4 GB of memory available, but this could be extended easily on a per user basis. Since the servers are virtualized it is also possible to allocate additional resources to the server even for just a short amount of time when a peak usage is predicted.

The key feature of the commercial version of RStudio Server is the possibility to have multiple sessions per user (with the open source version only one session per user is possible). Additionally, there are a number of enterprise features implemented in the commercial version of RStudio Server. At the moment the setup of a new more powerful R server with Rstudio Server Pro installed is under development and when ready users will be transferred to this server incrementally.

STATs current approach is described in the following, however this approach is deprecated and will probably be abandoned in 2018:

Through the centralised IT service everybody is able to request an installation packages of R, this installation package contains the following:

- R
- A predefined RProfile which gives information on startup about getting support and sets the CRAN mirror to the Austrian mirror, which is defined as an exception in the firewall.
- A “curated” collection of R packages. This list is curated by the methods unit and additional packages can be request by every R user. However, every R user can install additional R packages in a user specific library.
- RStudio
- MikTex (TeX distribution for Windows)
- Slik-Subversion (SVN command line tool for Windows)

The TeX installation is needed to be able to generate PDF documents from R Markdown files or R Sweave files and the SVN tools are necessary to access the internal SVN server, which can be used as version control for R projects.

About twice a year the R installation package is updated. This update is prepared in cooperation between the IT department and the methods unit.

Currently, a Linux server with R, RStudio Server and all necessary tools is available to a limited number of employees with the need for more memory (or for parallel processing). The server has 128 GB of memory and 16 cores and is used for some heavy-lifting tasks in R.

Netherlands

R is made available for common (desktop) use by installing it in a read-only central directory. Users of R and RStudio need to run a small script that sets up a shortcut to the centrally installed versions. The main advantage of this approach is that one user can write a script for a second (non-R) user that need not install anything. The R engine can be called from any virtual machine with access to the central directory. This means almost any VM in practice and it also includes the batch environment where (R) scripts can be scheduled to run.

Users also have access to a MikTeX installation so pdf documents can be generated from RStudio with ease. The methods department provides LaTeX, Wordsm, and HTML templates for internal reports and (externally published) discussion papers in CBS house style. These templates are made available from RStudio by default.

When upgrading, an old R version is not replaced but rather a new version is placed next to it. That way, production scripts need not to be updated and retested against the new distribution. Users are advised to upgrade scripts to the new R version during regular maintenance (e.g. updating for new data sets). This procedure is mainly developed to guarantee a degree of independence between updating R and applications of it. After a few years, older R versions can be removed as they are no longer used.

Statistics Netherlands has recently implemented virtual remote desktop servers with 64GB memory and 8 cores each. Any user can request access to such machines to run heavier applications or to work comfortably with larger datasets. These machines use the same centrally installed R distribution with a locally installed version of RStudio.

R users have access to a curated standard set of packages which are installed centrally as well. At each upgrade, the R expert group revisits the current set of packages, deprecating obsolete packages and adding new ones. For example, in the earliest versions, RGTK was the preferred tool to build graphical user interfaces while Shiny is recommended currently.

If so desired users can install packages in a personal library, using a local copy of CRAN that was downloaded at the same time as the central

R version. Since Statistical Departments have the ultimate responsibility for quality of statistics, users are warned that usage is at their own responsibility. R experts frequently help choosing or testing packages for specialized purposes.

PACKAGES

Since the number of available packages is huge, there is a need for some curated information with direct usage for NSIs. There are two main sources for this: The CRAN Task View Official Statistics and Survey Methodology (Templ, 2017) and the Official Statisitcs Awesome Software List (Ten Bosch and Van der Loo, 2017). The main difference between the two lists is that the CRAN Task View focusses on R packages, while the Awesome List includes any open source software that can be directly downloaded, is used at an NSI for production of statistics, and is free (in the sense of freedom). Both lists are open for suggestions by users.

Journals, conferences and workshops are other obvious ways to stay in touch. Interesting conferences for official statisticians are the 'use of R in Official Statistics' conference (uRos), the main *useR!* conference series, the new European R user meeting (eRum) and the biennial EARL conference (Effective Applications of the R Language). Additionally, there is a quite active R community on twitter, so it might be beneficial to follow people with similar interest there. A starting point for that could be to look at the followers of the two authors (@alexkowa and @markvdloo) and the accounts they follow. Most conferences have twitter handles as well and by searching for the relevant hashtag (e.g. #useR2017), useful information often surfaces.

Public Packages

An R package may be thought of as `the fundamental unit of R code' (Wickham 2014). Because packages contain both code and documentation, they are ideal containers for implementing novel methodologies in an organisation. Publishing R packages on CRAN has the additional advantage of possibly reaching more users, allowing or greater opportunity to find bugs in code or documentation. If the code is developed through an open environment such as github or or gitlab, other developers can contribute code or documentation as well.

Both Statistics Netherlands and Statistics Austria have published packages on the Comprehensive R Archive Network (CRAN). In the following, a concise overview of this work is provided.

Austria

Employees from STAT (and more specifically the method unit) have (co-)developed several packages which are available for download from CRAN, the most relevant are (see Templ et al., 2014 for details):

- sdcMicro (Templ et al., 2015) , sdcTable (Meindl, 2017) for statistical disclosure control for micro and tabular data (supported by the CENEX statistical disclosure control)
- simPop (Temp et al., 2017) for the generation of synthetic data sets
- VIM (Kowarik et al., 2016) for visualization and imputation of missing data
- x12 (Kowarik et al., 2014) for applying X13-ARIMA-SEATS to time series in R

These packages are already used in production in several NSIs and the packages for statistical disclosure control are under governance of the centre of excellence for statistical disclosure control which is partly funded by Eurostat.

All the public packages are hosted on Github, so it is easy to file an issue or to contribute to the development of any of the packages.

At the moment employees at STAT are working on a package that provides an interface to the EU GIS service GISCO, so it will provide the functionality of geocoding, reverse geocoding and routing. As STAT is part of the NETSILC3 project a package for bootstrapping and estimation of indicators and corresponding standard errors is also under development.

Netherlands

All published packages have been developed at the methods department. Most of them are a result of the annually updated research programme. Packages produced as a result of the research programme can be divided into packages focused on data editing, data visualisation, packages for handling large datasets and others.

In the area of data editing, the validate package (Van der Loo and de Jonge, 2017b) provides functionality to read, apply, investigate and maintain sets of edit rules. The results of confronting data with a set of edits can be conveniently summarized, exported or visualized. The validate package is fundamental to other data cleaning functionality. In particular, the errorlocate package (De Jonge and van der Loo, 2017) implements error localization functionality based on the paradigm of Fellegi and Holt (1976). A package for detection and removal of redundancies and inconsistencies in rule sets based on the validate package is currently under development and will be published soon under the name validatetools. (De Jonge and Van der Loo, 2017). These three packages replace the editrules package (De Jonge and Van der Loo) which combined the functionality of rule application, management and manipulation. Work on a package that can export data validation results in the new ESS validation report standard (Van der Loo and Ten Bosch, 2017) is underway.

Several rule-driven data cleaning methods developed at SN have been implemented in the deductive package (van der Loo and De Jonge, 2017). This package provides methods for deductive imputation, finding typing errors in numerical data under linear (in)equality constraints. The package uses rule sets defined in the validate package. Package simputation (Van der Loo, 2017) offers a simple interface to many different imputation (model-based) methods, while rspa offers functionality to force imputed numerical values to satisfy linear (in)equality constraints. The recently published lumberjack package (Van der Loo, 2017b) can be used to trace (log) changes in a data set as it gets processed during data editing. A thorough discussion of both theory and application of data cleaning in R can be found in Van der Loo and De Jonge (2017c).

In the area of data visualisation, the tabplot package (Tennekes and De Jonge, 2017) implements a method for visualizing large, multivariate datasets while tabplotd3 (De Jonge and Tennekes, 2013) implements a web-based interactive version. The treemap package of Tennekes (2017) implements tree plots -- a space-filling visualisation of hierarchical data structures. Finally, packages tmap and tmaptools provide functionality for creating thematic maps (Tennekes, 2017b, 2017c).

The package hbsae of Boonstra (2012) implements hierarchical bayesian methods for small area estimation.

Besides packages that are developed as part of the research programme, several employees at Statistics Netherlands have developed infrastructural packages for working with large data, either for other projects and/or in their free time. Worth mentioning are the LaF package (Van der Laan 2017) for handling large ASCII files, ffbase (De Jonge *et al*, 2016) for working with large binary files on disk (in ff format), and the lvec package by Van der Laan (2017b), which implements a disk-based representation of standard R vectors.

Finally, we note some packages developed by employees of statistics netherlands that have been mostly personal interest and free time projects. The widely used whisker package of De Jonge (2013) implements the moustache standard for logicless templating, the stringdist package of Van der Loo implements a range of string distance metrics, and docopt (De Jonge, 2016) package facilitates documenting and parsing of command line options when running an R script from the shell.

Internal Packages

To get R code ready for the statistical production process it is advisable to wrap it in an R package. The distribution and the development process (versioning) of an R package is well defined and well manageable. Therefore, there are specific R packages for specific tasks in the process.

Austria

The methods unit develops internal R packages to facilitate repetitive tasks. One example is drawing a sample of household or persons, for that the R package sampSTAT was developed, the centrally stored rich frame can be easily downloaded from within R without SQL knowledge or even knowing how the database containing the frame looks like. Additional functionality is drawing samples with different designs and auxiliary functions such as telephone number search or accessing the address register to get the most current postal information.

Previously, many price-indices at STAT have been calculated using Microsoft

Excel. Since this should be avoided, the package RPI allows defining a hierarchical structured price index and applying many useful methods such as switching the base period, splicing an index (e.g combining indices with different base periods), applying quality adjustments or automated reporting/ plotting on such index objects.

Based on the public R package for statistical disclosure control for tabular data sdcTable, the R packages anonLSE and anonKJE provide customized function for structural business survey and short term statistics. For the anonymization process, it is also taken into account that there are differences in classification of Austrian national publications versus European requirements and a coherent suppression pattern is computed.

Netherlands

At Statistics Netherlands, projects involving methodological advice sometimes include (co)writing an R package for a specific internal purpose. Examples include a package for statistical disclosure control in a specific area, a package for survey sampling capable of excluding units that were included earlier, a package for time series analysis, and most recently, a package for accessing and analyzing supply-and-demand tables for National Accounts. Some packages are (co-)developed by the methods department as a service to the statistical domain. In this case ownership is usually transferred when development is finished. In other cases, staff at the statistical domain develop the packages for local use.

COLLABORATION

Over the last years, R has surged in popularity both within and outside the Statistical Office. The rise of R (and other open source data science tools) went hand in hand with the rise of a very open and collaborative culture amongst a generation of developers in general and data scientists in particular. Statistics Netherlands and Statistics Austria have profited tremendously from the R community, both on- and offline. Our offices have also contributed back to the R community, for example by contributing packages. The informal way of collaborating in the open source community differs from the formalized collaboration instruments that are typical in the official statistics community. It is therefore worthwhile to highlight a few aspects of the way the open source community typically operates.

There are many ways in which one can choose to work together in an open source community. First of all, one can simply try or use an open source product. Each time software is used, it is in a way tested. In that sense every run contributes to its reliability. It is considered good (or nice) practice to acknowledge the author(s) of a package, for example by 'liking' or following the project site. Second, one can collaborate by advocating the software packages that have shown to be useful. Tools are both abundant and rapidly evolving, so even just pointing out a package on Twitter may help someone to discover the tool they need. Other typical ways to collaborate indirectly include answering questions on Q&A sites (stackoverflow is the most popular one) or mailing lists, write blog posts on applications, write a tutorial, or give talks at user meetings. A simple way to directly collaborate in an open source project is to provide feedback. One can post reports on performance, bugs in documentation or code, or discuss new features. Finally, the most direct way to collaborate is to contribute code or documentation to a project.

As for Statistics Austria and Statistics Netherlands: we are indeed using each other's packages, in production as well as in research. Also, bug reports and documentation suggestions have gone back and forth between package authors of our offices. Besides that, both Statistics Netherlands and Statistics Austria have received bug reports, suggestions and comments from (official) statisticians worldwide. In the authors experience, such contributions significantly enhance the quality of published R packages.

These more or less informal collaboration modes stand in contrast with the often highly regulated environment that is international collaboration in official statistics. There, international collaboration projects are proposed, defined, tendered, funded, executed, deployed and finally evaluated. It also stands in contrast with the way things are standardized. Where in the official statistics community standards are often designed and chosen in a top-down manner, in the open source community standards evolve by survival of the fittest: popular, useful ideas, standards, and software survive while others wither and die. Perhaps paradoxically, the freedom found in the open source community almost automatically induces a level of standardization. To illustrate, consider the example of IDE's (integrated development environments) developed for R. Before RStudio was published in 2011 there were many different solutions in use: Notepad++ with NpptoR, Tinn-R, Eclipse with the StatEt plugin and more. All have their advantages and disadvantages, but once RStudio was published it took over a large chunk of users almost immediately. Something similar happened in the field of report generation. Before the 'knitr' package was first published in 2012, numerous developers worked on R packages allowing users to inject results of computations into a report. The *useR!*2012 conference even had a tutorial session and a presentation session demonstrating several solutions. Examples include Sweave (coming standard with R), odfWeave (Kuhn, 2014), and several others. Once the knitr package of Xie (2015) came out (it was first released in 2012), it quickly took over the 'market' for report generation because it filled a need in a way that appealed to many users. Both RStudio and knitr can currently be considered *de facto* open source standards and both are the result of bottom-up approaches to a particular problem.

The power of the bottom-up approach was recently recognized by the establishment of the R Consortium (2015). One of the main activities of this industry-backed group is to provide funds for project grants, where anyone from `the R community at large' can write a proposal and apply for a grant. Over the last two years, eighteen projects have been funded. An example is the highly successful r-hub project (Csardi, 2015): a cloud service where package authors can run automated tests under different operating systems and build versions of R before submitting to CRAN. At the time of writing, the service has been running for two almost two years on an industry-sponsored cloud platform. Designing and building the infrastructure took an $80.000 grant and about one year time (R consortium, 2017).

In summary, the R community offers a wide range of collaboration possibilities allowing users and organizations with many levels of experience, skills, and needs to contribute. Since there are little or no formalities to take care of starting to collaborate is almost effortless.

SUMMARY AND CONCLUSION

In this paper, we have discussed the introduction of the open source tool R in the statistical office from the perspective of Statistics Austria and Statistics Netherlands. We have put particular emphasis on properties of R as an open source tool which sets it apart from many tools used traditionally in statistical offices. Although moving from a Commercial off the Shelve (COTS) approach to an open source product did pose some organisational and perhaps cultural problems, our experiences have been overwhelmingly positive. The community-driven approach to collaboration and standardization that is common in the international data science community in general and in the R community in particular, is something that both authors would highly recommend to anyone working in the field of official statistics.

References

1. **Boonstra, H.J.**, 2012, hbsae: Hierarchical Bayesian Small Area Estimation. R package version 1.0. https://CRAN.R-project.org/package=hbsae
2. **Csárdi, G.**, 2015, r-hub: the everything-builder the R community needs. Proposal to the R-consortium https://github.com/r-hub/proposal
3. **De Jonge, E, M van der Loo**, 2016, errorlocate: Locate Errors with Validation Rules. R package version 0.1.2. https://CRAN.R-project.org/package=errorlocate
4. **De Jonge, E**, 2013, whisker: {{mustache}} for R, logicless templating. R package version 0.3-2. https://CRAN.R-project.org/package=whiske
5. **De Jonge, E**, 2016, docopt: Command-Line Interface Specification Language. R package version 0.4.5. https://CRAN.R-project.org/package=docopt
6. **De Jonge, E**, **Van der Loo, M**, 2017, validatetools: Tools for Manipulating and Simplifying Validation Rule Sets. R package (to be published) https://github.com/data-cleaning/validatetools
7. **De Jonge, E, Van der Loo, M**, 2015, editrules: Parsing, Applying, and Manipulating Data Cleaning Rules. R package version 2.9.0. https://CRAN.R-project.org/package=editrules
8. **Edwin de Jonge, Martijn Tennekes**, 2013, tabplotd3: Tabplotd3, interactive inspection of large data. R package version 0.3.3. https://CRAN.R-project.org/package=tabplotd3
9. **De Jonge, E. J Wijffels, J Van der Laan**, 2016, ffbase: Basic Statistical Functions for Package 'ff'. R package version 0.12.3. https://CRAN.R-project.org/package=ffbase
10. **Dinges, G., Kowarik, A., Meindl, B., & Templ, M.**, 2011, *An open source approach for modern teaching methods: The interactive tgui system*. Journal of Statistical Software, 39(i07).
11. **IEEE**, 2017, *The 2017 top programming languages*. https://spectrum.ieee.org/computing/software/the-2017-top-programming-languages
12. **Kowarik, A., & Templ, M.**, 2016, *Imputation with R package VIM*. Journal of Statistical Software, 74(7), 1-16.
13. **Kowarik, A., Meraner, A., Templ, M., & Schopfhauser, D.**, 2014, *Seasonal Adjustment with the R Packages x12 and x12GUI*. Journal of Statistical Software, 62(1), 1-21.
14. **M. Kuhn**, 2014, odfWeave: Sweave processing of Open Document Format (ODF) files. R package version 0.8.4 https://CRAN.R-project.org/package=odfWeave.
15. **Meindl, B.**, 2017, sdcTable: Methods for SDC (statistical disclosure control) in tabular data, URL http://cran. r-project. org/package= sdcTable. R package version 0.22.8.
16. **PYPL**, 2017, *Popularity of Programming Language*. http://pypl.github.io/PYPL.html
17. **R consortium**, 2015, https://www.r-consortium.org/about
18. **R consortium**, 2017, Awarded projects. https://www.r-consortium.org/projects/awarded-projects

19. **Templ, M.**, 2017, *CRAN Task View of official statistics software. A curated list of R packages, useful for official statistics*. https://cran.r-project.org/web/views/Official-Statistics.html
20. **Templ, M., Meindl, B., Kowarik, A., & Dupriez, O.**, 2017, *Simulation of synthetic complex data: The R-package simPop*. Journal of Statistical Software, 79(i10).
21. **Templ, M., Kowarik, A., & Meindl, B.,** 2015, *Statistical disclosure control for microdata using the R package sdcMicro*. Journal of Statistical Software, 67(1), 1-36.
22. **Templ, M. & Todorov, V.**, 2016, *The Software Environment R for Official Statistics and Survey Methodology*. Austrian Journal of Statistics, 45, 97-124
23. **Templ, M., Kowarik, A., & Meindl, B.**, 2014, *Development and Current Practice in Using R at Statistics Austria*. Revista Română de Statistică nr. 2, 173.
24. **TIOBE**, 2017, *TIOBE Index for October 2017*. https://www.tiobe.com/tiobe-index/
25. **Ten Bosch, O.** and **M. van der Loo**, 2017, *Awesome official statistics software. A curated list of official statistics software that is free (libre), downloadable, open source and used at statistical offices*. http://www.awesomeofficialstatistics.org
26. **Tennekes, M** and **E de Jonge**, 2017, tabplot: Tableplot, a Visualization of Large Datasets. R package version 1.3-1. https://CRAN.R-project.org/package=tabplot
27. **Tennekes, M**, 2017, treemap: Treemap Visualization. R package version 2.4-2. https://CRAN.R-project.org/package=treemap
28. **Tennekes, M**, 2017b, tmap: Thematic Maps. R package version 1.1.0. https://CRAN.R-project.org/package=tmap
29. **Tennekes, M**, 2017c, tmaptools: Thematic Map Tools. R package version 1.2-1. https://CRAN.R-project.org/package=treemap
30. **Van der Laan, J**, 2017, LaF: Fast Access to Large ASCII Files. R package version 0.7.1. https://CRAN.R-project.org/package=LaF
31. **Jan van der Laan**, 2017b, lvec: Out of Memory Vectors. R package version 0.1.1. https://CRAN.R-project.org/package=lvec
32. **Van der Loo M**, 2014, *The stringdist package for approximate string matching*. The R Journal, 6, pp. 111-122. https://CRAN.R-project.org/package=stringdist.
33. **Van der Loo, M** and **E de Jonge**, 2017,. deductive: Data Correction and Imputation Using Deductive Methods. R package version 0.1.3. https://github.com/data-cleaning/deductive
34. **Van der Loo, M** and **E de Jonge**, 2017b, validate: Data Validation Infrastructure. R package version 0.2.0. https://CRAN.R-project.org/package=validate
35. **Van der Loo, M** and **E de Jonge,** 2017c, *Statistical data cleaning with applications in R*. Wiley Inc (in press).
36. **Van der Loo, M** and **Ten Bosch**, 2017, *Design of a machine-readable validation report structure*. Deliverable of Tasks 14/2 and 15/2 of the ESSnet Validat Integration. https://goo.gl/hEGdbo
37. **Van der Loo, M**, 2017, simputation: Simple Imputation. R package version 0.2.2. https://github.com/markvanderloo/simputation
38. **Van der Loo, M**, 2017, lumberjack: Track Changes in Data. R package version 0.2.0. https://github.com/markvanderloo/lumberjack
39. **Xie, Y.,** 2015, *Dynamic Documents with R and knitr*. 2nd edition. Chapman and Hall/CRC. ISBN 978-1498716963