Statistical Journal of the IAOS
1–10
© The Author(s) 2026
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/18747655251411424
journals.sagepub.com/home/sji

Statistical open source software for official statistics: State of play and future directions

Olav ten Bosch[1] and Mark PJ van der Loo[1,2]

Abstract

Statistical organizations worldwide are increasingly adopting open source technologies for producing official statistics. This shift is motivated by the potential of open source tools to increase transparency, improve efficiency, and enhance reproducibility. Moreover, young professionals in statistics and data science enter the labour market with strong skills in open source tools. The adoption of open source software signifies a change in how statistical organizations operate and collaborate.

This paper provides an overview of the state of open source adoption in official statistics. It details the open source movement among statistical organizations, the experiences of Statistics Netherlands with open source adoption and the creation of R-packages implementing common statistical methods. It also describes the development and use of the "awesome list of official statistics software" and discusses a set of principles for open source in official statistics, derived from best practices across various organizations. These principles have been endorsed (June 2025) by the Conference of European Statisticians (CES).

Furthermore, it explores future directions for maturing this community, including metrics for assessing maturity, such as true independence of software modules, support for uncertainty propagation, and privacy by design. Moreover it presents ideas on redesigning the statistical open source landscape.

Keywords

Open-source, methodology, architecture, community, principles, strategy, collaboration

Received: 15 May 2025; accepted: 12 December 2025

1 Introduction

National Statistical Offices (NSOs) around the globe are increasingly moving their production processes towards methodologies based on open source technologies. This transition is fueled by the potential of open source tools to enhance transparency, improve efficiency, and foster reproducibility in statistical analysis. It is also driven by the fact that data scientists and statisticians nowadays arrive on the labour market with extensive knowledge of open source software tools.

In contrast to commercial off-the-shelf software (COTS), the development and use of free and open source software (FOSS) is not dominated by contracts and service level agreements with a single supplier. Rather, a community of users and contributors, often led by a small core team, develop and maintain the product. The adoption of open source software, for methodological and operational process steps, is therefore not merely a change in technology but signifies a deeper change in how statistical organizations operate and collaborate. It includes changes in technology, operations, competencies, and culture.

This paper aims to provide an overview of the state of play of open source software in official statistics and to progress thinking in future directions. Chapter 2 will detail on the current open source movement among statistical organizations, the open source adoption at Statistics Netherlands, the introduction and operation of the awesome list of official statistics and the leading principles for

[1] Methodology and Process Development, Statistics Netherlands, The Hague, The Netherlands
[2] LIACS, Leiden University, Leiden, The Netherlands

Corresponding author:
Olav ten Bosch, Methodology and Process Development, Statistics Netherlands, Henri Faasdreef 312, The Hague 2492JP, The Netherlands.
Email: o.tenbosch@cbs.nl

open source adoption derived from best practices in multiple organizations. Chapter 3 dives into future directions to mature this community, with sections on metrics for measuring open source software maturity, a strategy on truly independent software modules, challenges on propagation of uncertainty in an open source software chain and other development. It concludes with a section on redesigning the software landscape. Chapter 4 provides a conclusion.

2 State of play of open source in official statistics

Several NSOs have already made significant progress in integrating open source tools into their workflows. The Central Statistics Office (CSO) in Ireland has made a strategic decision to transition to R as its primary analytics tool, recognizing the necessity for a modern and sustainable analytics environment.[1] Statistics Norway is currently developing its own Data Platform ("DAPLA") in the public cloud, incorporating open source technologies like Python and R to modernize its statistical production processes.[2] Statistics Poland recently declared the use of Open Source software in the statistical production process as crucial.[3] Statistics Austria is an active player on the statistical open source marketplace with many methodological R packages used internally as well as at other NSOs.[4] Statistics Canada has published procedures to publish open source code on behalf of their organisation using a well-defined workflow.[5] The Office of National Statistics (ONS) in the UK transforms, as part of the wider analytical community in the UK government, the way they do analysis by moving to large-scale adoption of open source tools.[6] Statistics Netherlands has been utilizing R and Python for fifteen years and has established the awesome list of official statistics software to facilitate knowledge exchange among statistical organizations.[7] Many work on adopting open source policies to transform their traditional production systems into architectures composed of ready and trusted open source statistical building blocks.

A recent paper written under the UNECE high level group on modernisation of official statistics (HLG-MOS) on the future of National Statistical Offices[8] discusses the challenges and opportunities faced by NSOs. It emphasizes the need for NSOs to adapt to changes such as the rise of mis- and disinformation, competition from private sector data providers, declining survey response rates, and the emergence of new technologies. The document highlights that new innovations, including open source solutions, present major opportunities for NSOs. In particular, regarding open source, the paper notes that these solutions provide NSOs with options for collaboration, greater transparency into their practices, and easier and timelier dissemination of their data products.

Another HLG-MOS initiative is the Open-Source Software Charter for Official Statistics.[9] It recognises open source software as essential for modern statistical production, promoting transparency in methodology and fostering international collaboration in developing and supporting the production of official statistics. Moreover it lists 7 principles to be used to put open source into practice. We dive deeper in these principles in section 2.3.

The transition from closed to open technologies, however, presents several challenges for NSOs. These include the need to acquire new skill sets, integrate open source tools with existing systems, and establish robust governance and quality assurance frameworks. To address these challenges, NSOs are implementing strategies such as developing Reproducible Analytical Pipelines (RAP), fostering communities of practice, and creating comprehensive open source policies.

The adoption of open source technologies also brings significant benefits. Open source promotes transparency by making tools and code freely available, enhancing the verifiability and reproducibility of statistical processing. It improves efficiency by facilitating code sharing and reducing duplication of effort. Furthermore, open source fosters collaboration, both within and between organizations, driving innovation and knowledge exchange.

Open source adoption comes with a culture of community building. FOSS communities typically use standardized ways to ask questions and to contribute, where an important aspect is that users of all levels of experience are able to contribute something useful. Experienced programmers might solve bugs or implement features, while less experienced users can contribute for example by asking questions, testing preliminary versions, improving documentation, or creating worked examples for beginners. Open source adoption also means embracing a culture of collaboration within an organisation as much as between organisations – often in informal settings.

2.1 Open source adoption at Statistics Netherlands

In 2010, Statistics Netherlands approved the Free and Open Source (FOSS) tool R for production use, expanding its initial role as a research tool.[4] A bottom-up adoption of R by professionals led to user groups, courses, development standards, and application management. This coincided with the creation of Statistics Netherlands' FOSS policy. R gained popularity among statisticians, with at least 50% of heavy processing jobs now using R scripts. Moreover, the site dashboards.cbs.nl is fully developed with R Shiny.

Python was also approved for statistical production in 2012, following a similar bottom-up approach with user meetings. While Python's adoption has been slower than R's, likely due to R's closer alignment with statistical work compared to Python's appeal to later-arriving data scientists, R is now recommended for statistical data work, and

Python for orchestration and process management. Functionally, Python and R have converged. Python initially excelled in web service integration and machine learning/text mining, while R was strong in visualization, data processing, statistical libraries, and its package management system (CRAN). Today, they are largely equivalent in many areas, except for package management maturity.

In 2014, Git was adopted for version control for non-IT programmers, replacing SVN with an internal server based on Gitbucket. Git training is now part of the standard curriculum at the 'CBS Academy' while Gitea replaced Gitbucket in 2024.

Recognizing the need for production systems built with composable modules, CBS researchers began actively using and contributing to international open source ecosystems. A significant portion of these contributions are R packages focused on statistical data cleaning and processing. Over time, Statistics Netherlands and other statistical institutes have adopted these packages as core components. Within CBS, these popular R packages are integral to producing statistics across diverse domains, including social and economic indicators, agriculture, international trade, education, environmental data, emissions, income, shipping, short-term statistics, recreation, and museums, among others. Notably, statistical institutes in Iceland, Denmark, Italy, and Brazil, along with the US Department of Agriculture National Agricultural Statistical Service (USDA-NASS), which utilizes CBS R packages for validating and cleaning data from large national surveys of American farmers, have also adopted these tools.

Currently, this ecosystem consists of a number of packages that integrate seamlessly. Not only because there is a shared technical platform (*i.c.* R), but also because careful thought was put in pegging out, with formal precision, what each fundamental processing step entails. The main packages are:

- validate: Check validity of data based on user-defined rules[10,11]
- dcmodify: Adapt erroneous data based on user-defined rules[12]
- errorlocate: Find the minimal number of erroneous data points[13]
- simputation: Many different imputation methods with a single, easy to learn interface[14]
- rspa: Adjust numerical records to fit equality and inequality restrictions[15]
- deductive: Solve data errors, using the data and validation rules[16]
- validatetools: Find contradictions and redundancies in rule sets[17]
- accumulate: Grouped aggregation, where grouping is dynamic and data-dependent[18]
- lumberjack: Automatically track changes in data for logging purposes[19]
- reclin2: Join datasets based on (multiple) possibly inexact keys[20]
- rtrim: Estimate growth and decline of animal populations in the presence of missing data[21]
- tmap: Create (interactive) thematic maps in R such as choropleths and bubble maps[22]
- hbsae: Hierarchical Bayesian Small Area Estimation[23]

We describe two examples demonstrating the extensibility and power of this modular approach.

The first example concerns *data validation*. Until the 2010s, the act of checking data against domain knowledge, in the form of validation rules, was not recognized as a separate activity. In many systems this was either hard-coded by users or integrated in a larger data-editing system. Creating a separate package (validate) with the sole purpose of defining, manipulating and executing data validation rules yielded the possibility of monitoring the progress of data quality along multiple statistical value chains using a single piece of software. Moreover, the rule management system of the package is reused in packages for error localization (errorlocate), data correction (deductive) and aggregating based on dynamically defined data groupings (accumulate).

A second example is an imputation package (simputation) that allows users to compose popular imputation models in fall-through scenarios that are often used in economic statistics. The package allows for group-wise processing, where groups are statically defined. When the need arose to extend the functionality, it was possible to define a new add-on package (accumulate) that allows for grouping of data where the grouping is determined dynamically, depending on data circumstances. The fact that it was possible to add new, unanticipated functionality is a consequence of the careful design and separation of concerns when designing each individual module.

All packages have been developed in the open, by hosting the code on open version control repositories (GitHub), presenting the work at conferences, publishing in scientific journals, and promoting usage and feedback from (potential) users. The uptake of packages by non-CBS users is facilitated by releasing the packages on a standardized release platform (CRAN), using permissive licenses and paying attention to documentation. Feedback and contributions from users outside of Statistics Netherlands, and even from outside of the official statistics community has substantially helped improve and generalize the software. The fact that software can be easily downloaded and installed makes it trivial for R users to give the software a try and the open development platforms facilitate reporting of questions, issues, or even to contribute. It should in this respect be mentioned that contributions may range from things as simple as fixing typing errors in the documentation, to demonstrating new use cases, filing bug reports, or even fixing bugs or adding functionality.

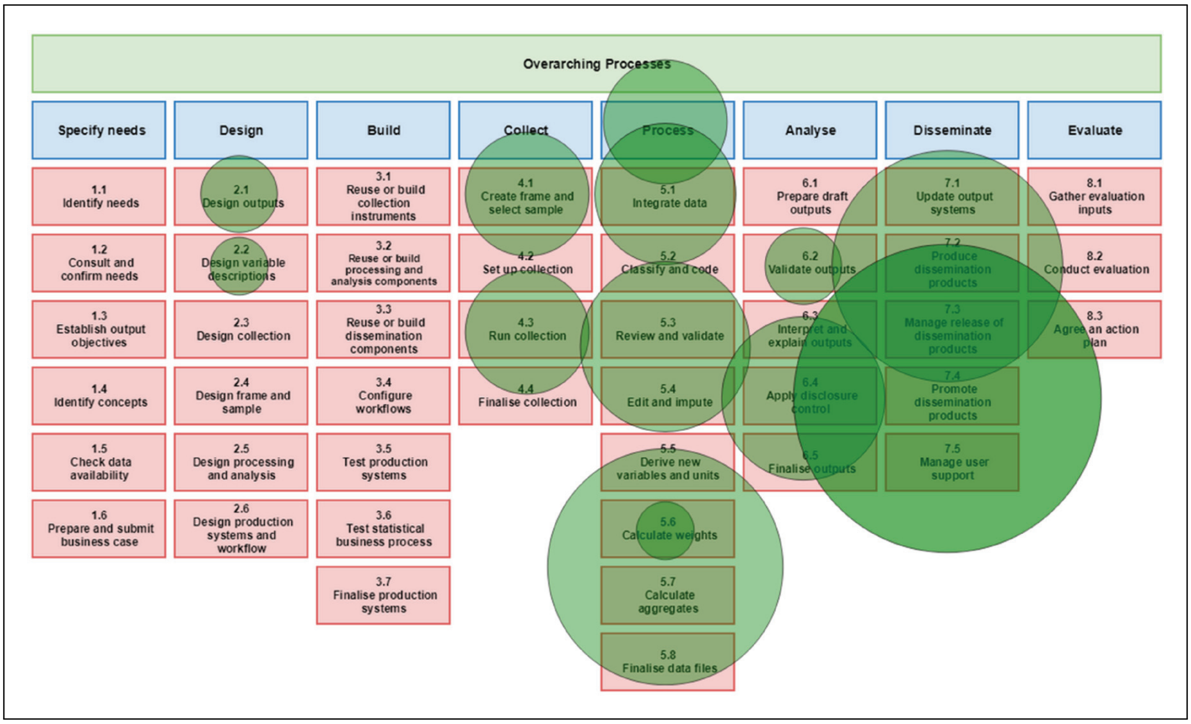


Figure 1. Awesome official statistics packages organized by GSBPM.

In 2018, Statistics Netherlands hosted the sixth installment of the *use of R in Official Statistics* conference (uRos) at the Hague. The conference sold out at 100 participants from over 40 countries. The event in The Hague marked the first occasion of uRos outside of its `home country' of Romania, and it has been traveling biannually since then, with the latest installment (2024) hosted by the Hellenic Statistical Authority (Greece).

Current and future developments at Statistics Netherlands heavily lean on open source. The main renewal programs have chosen open source tools R and Python as the standard tools, in combination with cloud infrastructure based on kubernetes. The innovation of the output domain is focusing on use of the free and open source .STAT suite.[24]

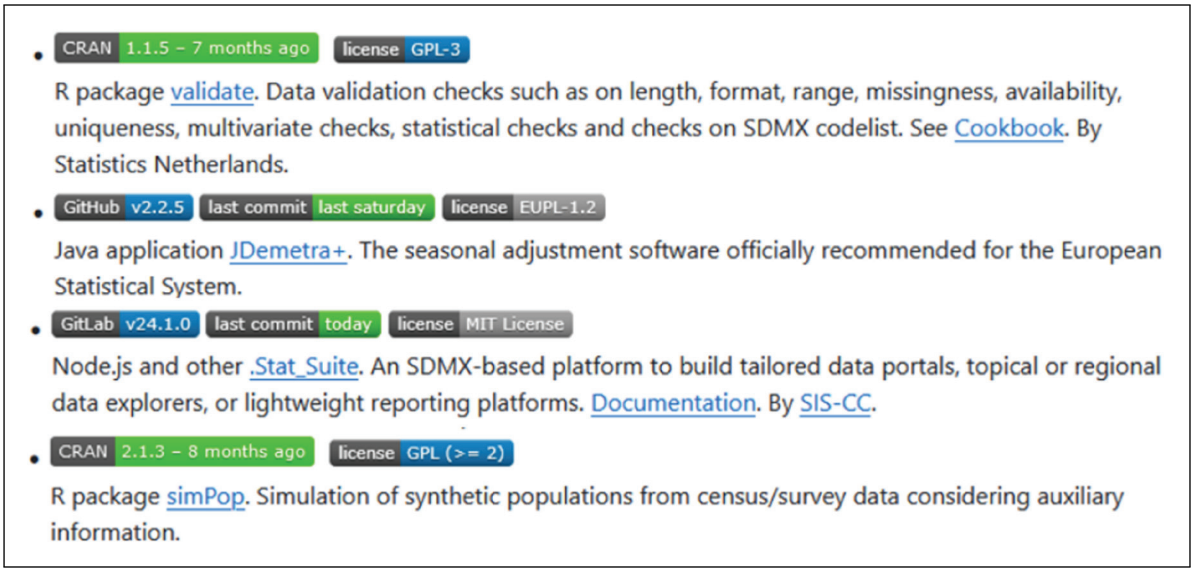
2.2 *The awesome list of official statistics software*

Statistics Netherlands was not alone in its adoption of open source solutions, with similar trends emerging in other statistical organizations. Recognizing this growing movement, the authors of this paper collaborated with international leaders to launch the “Awesome List of Official Statistics Software”.[25] This curated and living collection of tools for statistical production was initiated in 2017 during the UNECE conference on Statistical Data Editing (SDE) held in The Hague. By 2019, the list had grown to 100 contributions and currently features 153 Free and Open Source Software tools.

The tools on the awesome list need to fulfill a few criteria. They need to be free, open source, available for download, and used in the production of, or provides access to, official statistics. Moreover, tools should be easy to install and use, and actively maintained. Developed collaboratively in the open source spirit, the list benefits from numerous international contributions and is organized according to the Generic Statistical Business Process Model (GSBPM). Figure 1 illustrates the distribution of the software packages across the various GSBPM processes.

The main purpose of this list is to not replicate information that is already maintained elsewhere, but to simply point to the information provided by the respective open source developer(s). Hence, each entry on the list provides a link to the software download, a brief description, and essential metadata such as the latest version, last commit, and license, provided in a popular badges format as used in open source communities. Figure 2 showcases examples of entries on the list.

The information for each package is automatically extracted from the services where the packages have been published. In particular, data is gathered automatically from CRAN, PyPi, Gitlab and GitHub. Most of the work is done by connectors that can be extended to add new services, while some rare platforms require manual configuration. This approach allows for deriving statistical information on the open source software. As an example, Figure 3 shows the distribution of programming languages of items on the list. It becomes clear that the vast majority of items are



- CRAN 1.1.5 – 7 months ago license GPL-3

  R package validate. Data validation checks such as on length, format, range, missingness, availability, uniqueness, multivariate checks, statistical checks and checks on SDMX codelist. See Cookbook. By Statistics Netherlands.
- GitHub v2.2.5 last commit last saturday license EUPL-1.2

  Java application JDemetra+. The seasonal adjustment software officially recommended for the European Statistical System.
- GitLab v24.1.0 last commit today license MIT License

  Node.js and other .Stat_Suite. An SDMX-based platform to build tailored data portals, topical or regional data explorers, or lightweight reporting platforms. Documentation. By SIS-CC.
- CRAN 2.1.3 – 8 months ago license GPL (>= 2)

  R package simPop. Simulation of synthetic populations from census/survey data considering auxiliary information.

Figure 2. Examples of items on the awesome list of official statistics software.

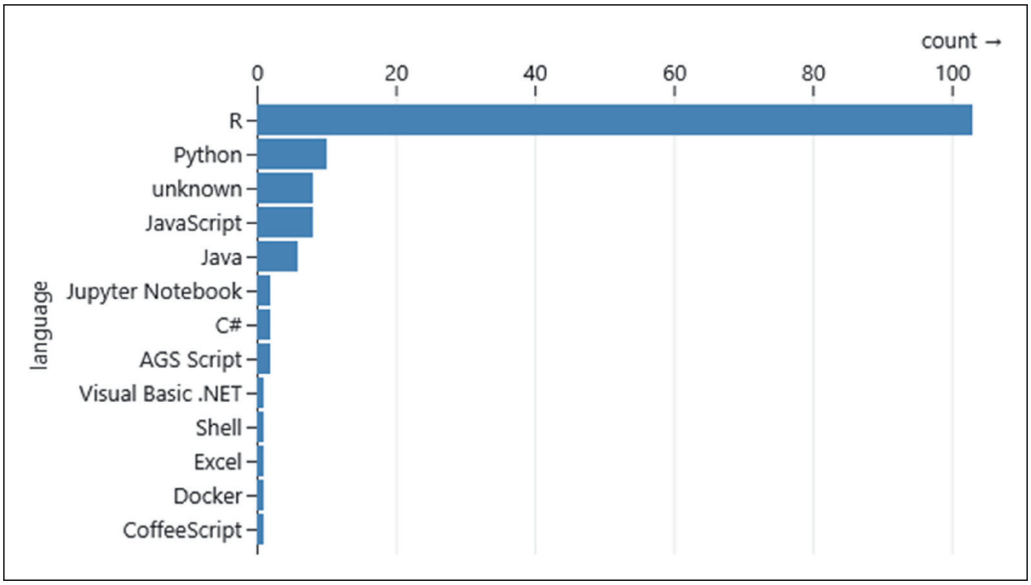


Figure 3. The most popular programming languages among items on the list.

written in R, which testifies to the excellent methods of software sharing in this community. Figure 4 shows license statistics. GPL is the most popular, followed by MIT and EUPL. The latest statistics on the list are always available from the visuals section on the list itself.

This list has turned out to be a valuable way for different types of users in the European Statistical System (ESS) – like statistical method experts, statisticians, IT staff, and managers – to find out about open source software that's already available. Statistical organizations worldwide suggest changes or additions. Ideas for adding features like compatibility details, how well-tested the software is, and how many people use it are noted on the list's GitHub page. The UNECE has recommended this list to NSOs on LinkedIn. In conclusion, this list, which many people contribute to, has already helped statistical organizations to reuse software, and it will continue to do so if the community stays active in using and maintaining it. But it has the potential to be more than just a list. Chapter 3 will present ideas for adding metrics that could further improve the open source software situation for statistics.

2.3 Principles on open source software

In the latter half of 2022, statistical agencies from numerous countries, alongside Eurostat and the OECD,

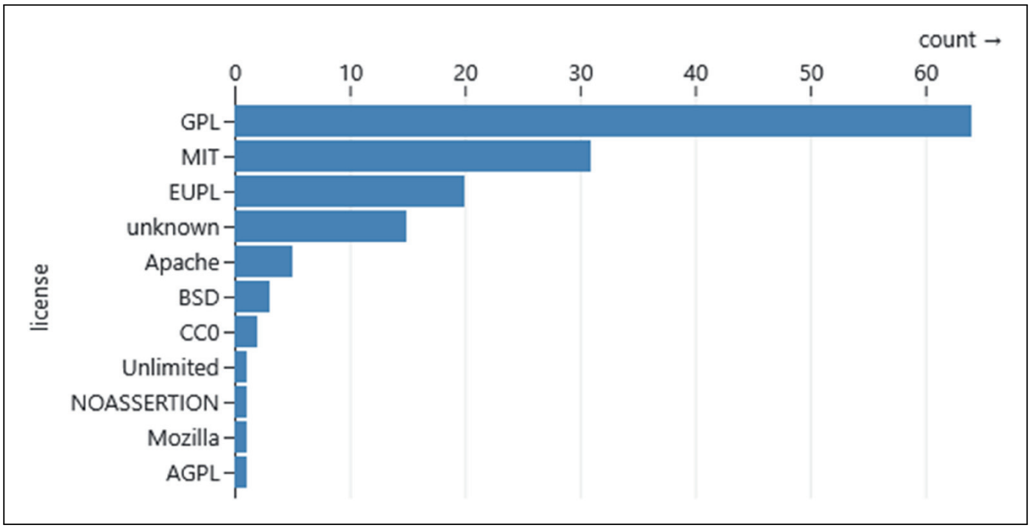


Figure 4. The most popular open source licenses used on the list.

established the 'Group on Open Source for Official Statistics (OS4OS)'. This group's mandate was to share and review current practices and lessons learned regarding the use of OSS for statistical purposes, and to explore potential collaborative work within the European Statistical System (ESS), including governance and technical tasks. Statistics Netherlands actively participated in this group, leading the subgroup focused on defining core principles.

These principles were subsequently adopted verbatim by the UNECE HLG-MOS group on open source, forming the basis of their software charter, available online.[9] This charter provides a one-liner summary, a brief explanation, the rationale, and implications for National Statistical Institutes (NSIs) and international organizations for each principle.

The seven key principles are:

1. **OSS by default**: In the production of official statistics we prefer the *use* of open source software solutions over closed software solutions. Moreover we *share* our software solutions as open source**.**
2. **Work in the open**: We start our projects in the open from the beginning and clearly mark maturity status.
3. **Improve and give back**: We rather *improve* existing open source solutions than decide to create new solutions and we *give* our improvements *back* to the respective open source community.
4. **Think generic statistical building blocks**: In our open source work we strive for re-usable *generic functional building blocks* that support well-defined methodology in statistical processes.
5. **Test, package and document**: We test, package and document our open source software for easy-re-use.
6. **Choose permissive**: We choose the most permissive OS license possible for sharing our software.
7. **Promote**: We invest in *promoting* new developments or improvements on our open source software within the official statistics community and where applicable in a wider context.

These seven principles are not arbitrary; they are grounded in extensive experience from collaborative open source projects within the official statistics community, many of which are featured on the Awesome list of official statistics software and discussed at conferences such as "The Use of R in Official Statistics" (2013–2025) and UNECE ModernStats World Workshops (2018–2022). Furthermore, they align with ESS and EU policies, including the ESS Code of Practice[26] and the EU-Open Source Strategy,[27] and build upon earlier work on best practices and strategies.

We believe these principles can serve as a cornerstone for the continued growth of the open source community for official statistics. Ideally, they can guide the creation of an environment that fosters the expansion and productivity of this community. For strategic and high-level management participating in the UNECE Conference of European Statisticians (CES), these principles have significant implications, such as: 1) supporting the primary selection of OSS over proprietary solutions; 2) establishing frameworks for employees to collaborate openly with experts from other (statistical) organizations on FOSS at all maturity levels; and 3) understanding and supporting employees in refactoring software to effectively contribute improvements back to the open source community. Broad support for these principles from high-level management within statistical organizations would send a clear message to the ESS OSS community that the open source work conducted over the years and planned for the future is not solely reliant on individual efforts but is an integral part of the intended operation and culture of the entire ESS. The UNECE HLG-MOS Statistical Open Source Software Guiding Principles, have been endorsed (June 2025) by the CES, after a broad consultation among the members, which turned out very positive.

2.4 Reflections on state of play

From the previous sections it is clear that the landscape of official statistics is undergoing a significant transformation, driven by the increasing adoption of open source technologies. National Statistical Offices (NSOs) are strategically embracing this shift to enhance transparency, efficiency, and reproducibility. Important ingredients in this shift is knowledge of mature statistical solutions as maintained in the awesome list of official statistics software, and the established principles for open source in official statistics put together by the Eurostat group on Open Source for Official Statistics (OS4OS) and adopted by the UNECE High Level group on modernisation of official statistics (HLG-MOS) in the Open source Charter.

However, the continued evolution of the open source community in official statistics requires further elaboration. A key challenge is how to foster the development of a mature and efficient community capable of creating and maintaining a set of essential open source building blocks for use by any NSO. While resources like the awesome list of official statistics software are very valuable and should be continued, there might be extensions or other mechanisms to encourage responsible growth. For instance, adding information about the popularity of each building block or the supported data formats could be beneficial. However, requiring too much metadata could make the list difficult to maintain. The challenge lies in finding the right balance of metadata — one that is sustainable and still offers sufficient value to the official statistics community. In the following chapter, we delve into strategies to achieve this goal.

3 Future directions and open questions

Before discussing open questions for future FOSS development, it is worth briefly discussing current ideas on the architecture of statistical production systems. Many NSOs are currently moving from stove-pipe systems towards standardized and modular systems characterized by a few steady states of well-defined data products. Typically, processing data from one steady state to the next involves many steps that require extensive use of methodology and domain knowledge.

In order to make tools reusable it is necessary to externalize domain knowledge as configuration parameters. A single tool connecting two steady states would limit its reusability because of the many substeps it would contain. For the same reason it would render configuration of such a solution overly complex. We therefore argue that the reusable components implementing this processing need to perform smaller tasks than processing data from one steady state to the next. These smaller software components need to be both compatible, so they can be easily combined, and independent such that they can be separately parameterized.

3.1 Truly independent software modules

The idea of building statistical production systems from reusable components is not new. For example, in the Common Statistical Production Architecture[28] it was proposed to wrap new or existing software components so that they can exchange data and be controlled in a uniform manner. Such an approach depends mainly on agreeing on technical interfaces and data formats.

However, such a design does not necessarily create fully independent components that can be added in any order, free of emergent effects or interaction. The reason is that when we combine components, the way one component is parameterized may affect how other components need to be parameterized. For example, it is not uncommon to use a fall-through scheme when imputing many variables. In such a scheme one attempts to impute a target variable by estimating a model based on auxiliary information, and if that information is not available, an alternative method is chosen that uses another choice of auxiliary information. Here, the choice of auxiliary information that should be chosen in the second method, is not independent from the choice made in the first method. In statistical production systems, one easily finds more interdependencies that depend on domain knowledge. For example, the imputation method used will influence computation of uncertainty down the line and the type of data validation rules that are used as input for automated data cleaning restrict what methods can be used for imputation and adjustment.

This problem can be expressed more formally as follows. Let D be the domain of a variable or set of variables. We let P and Q be configurable software components that do some data processing, but leave the domain intact – *e.g.,* a rectangular data set is transformed into a rectangular data set with the same variables and dimensions. In notation this means we can write $P : \Omega \to D \to D$ and $Q : \Delta \to D \to D$, where Ω and Δ are the possible parametrizations of P and Q. In this notation, $P(\omega)$, $\omega \in \Omega$, is a function $D \to D$ and similar for Q (i.e., we model the act of configuring a component as *currying*). In the imputation example, P and Q could be different imputation models and Ω and Δ are the possible values for auxiliary information or of hyperparameters needed to specify the model. Applying Q after P can in principle be written as the formal composition $Q \circ P : \Delta \times \Omega \to D \to D$ but we saw in the example that not every combination of parameters in $\Delta \times \Omega$ makes sense. The valid subset of $\Delta \times \Omega$ depends on subject matter knowledge. We therefore would like to make progress on several questions.

First of all, it is interesting to learn under what conditions such subject matter dependencies arise, and whether these circumstances can be formalized or detected automatically. We could probably learn something from methods and tools that study interdependent data validation rules or data manipulation rules. Second, there is the question on the granularity of the basic building blocks. One solution to the problem sketched here, is to build large components that contain all parameters as input. However, this is not user-friendly as they require a lot of configuration, even for simple tasks. It is also undesirable from a software engineering perspective. As building blocks get smaller, they are easier to use and maintain, but the number of subject matter dependencies probably increases. At the moment it is unclear where the best trade-off between these two effects lies.

Besides the technical considerations stated above, there are also cultural aspects to take into account. Designing truly independent and hence composable and interchangeable components is typically not in the interest of commercial off the shelf (COTS) software providers. Indeed, for such providers it is more interesting to offer a software ecosystem that can only be extended with components of the same provider, binding users through a vendor lock-in. This driver in the COTS environment stands in contrast with open source culture, where reuse, collaboration, and co-creation is the norm. An overview of truly independent statistical operations, or a method for determining whether a certain operation is in some sense truly independent has therefore its most natural place in FOSS culture.

3.2 *Propagation of uncertainty in statistical pipelines*

The essence of statistics is to quantify the uncertainty of results. In traditional statistics, the uncertainty associated with an estimator is usually expressed in terms of the expected difference between the estimate and the true (population) value (bias) and as the variation over repeated experiments (variance). For relatively simple estimators, mathematical expressions can be derived that compute, approximate, or limit these parameters.

A statistical production system can be interpreted as a complex sequence of processing steps that together form an estimator of a parameter of interest. Each individual step in principle contributes to bias and variance in a non-trivial (non-linear) way. At the moment there is little known of the joint effect of combining processing steps on the uncertainty in outcome. Let alone what the effect is when one or more intermediate steps are altered, replaced, added, or removed. Some results in this direction, based on Eve's law of conditional variance and bootstrapping procedures, have been reported.[29] However, this approach is hard to scale, especially to production systems that include manual interventions.

We can not expect that the uncertainty associated with a complex sequence of steps is a linear function of uncertainty associated with the individual steps. Creating *a priori* statements about the final uncertainty associated with a system at design time or when updating it will probably be prohibitively difficult. However it is important to investigate whether propagated uncertainty can be automatically, or systematically derived so the effect of design decisions can be properly judged.

The problem of propagating uncertainty in complex data processing systems poses both mathematical and technical challenges. This includes questions such as: how can we properly compute uncertainty throughout a statistical pipeline? How is uncertainty propagated in complex, multi-stream processing tasks, such as cases where multiple sources are used, or the output of intermediate steps is reused at multiple instances? Can we meaningfully standardize the definition and measurement of uncertainty for multiple processing steps and transport it throughout the system?

3.3 *Privacy by design*

Modern statistical production systems are required to follow the 'privacy by design' principle. Meaning that from the design stage onwards, privacy and risk of disclosure are considered from the beginning. This is especially interesting for systems that cross organizational boundaries, like in Edge Computing.

This leads to several open questions such as: what requirements can be imposed on software components in such a way that the privacy by design principle is guaranteed? A very practical example is that components should not write potentially sensitive data to temporary storage that is not sufficiently secured. Or what is potentially exposed in the logging of these tools? More generally it means that every open source component in the process chain should be able to be trusted on privacy aspects. In an statistical process composed of multiple independent open source building blocks this is a shared responsibility among all developers of all elements, who thus need to agree on a minimum level of privacy by design principles. From a statistical viewpoint this raises methodological challenges, such as how do any privacy-related issues propagate through a system that consists of multiple components? Under what conditions can statistical privacy problems arise or be prevented? How do we model these thoughts in concrete guidelines or best practices for open source developers in official statistics?

3.4 *Redesigning the landscape, guided by metrics*

In the previous sections we raised ideas and questions on a future open source landscape that supports truly independent software modules, propagation of uncertainty, and supports privacy by design. In this section we ask ourselves how the official statistics open source community could mature from the current state of play to a landscape satisfying these features.

As an example of such possible landscape redesign, we take a closer view at the category “access to official statistics” on the awesome list. This category consists of software components either built by a statistical organization to facilitate access or - in many cases - data scientists outside of the organization that want to ease data access for themselves and others. It provides a convenience layer for access to official statistics in general. This category of packages is interesting because there are a large number of packages (42 at the time of writing), which are developed independently. As a result, package developers have made different choices in design, features, and the data provider(s) they give access to.

In[30] this part of the landscape has been analyzed in more detail. This work includes, amongst others an overview of access points per package and a comparison of features per package . Although all packages principally have the same goal: access official statistics directly from data analysis software, it was found that the available user-facing functionality varies significantly across different implementations. Moreover, the interfaces differ across packages, forcing users to learn multiple interfaces. This led to the idea that the community of developers in this category should strive towards a uniform set of features for accessing any statistical data from any statistical provider.

For every part of the awesome list comparable strategies could be defined. To guide open source developers to collectively mature the official statistics software landscape, metrics should be defined. There is a large body of literature on software quality, focusing on aspects pertaining to the software itself, to maturity of organizations using it, and to the organization building and maintaining it.[31,32] For this goal, we need metrics targeted at the needs of each category.[33]

These metrics could be based on the thinking in the previous sections, namely:

- truly independency
- propagation of uncertainty
- privacy by design

Other ideas, specific for official statistics, include:

- Uniqueness (is there not something else already that covers it partly)
- alignment with official statistical standards and data standards
- alignment with steady states architecture
- domain neutrality (supports many statistics)
- domain knowledge configurability
- The right granularity level: small enough to be treated as a simple black box operation, and large enough to perform a statistical function.

In addition, common metrics on software quality could be added, such as:

- ease of use and learnability
- minimal dependencies
- stable interfaces over time
- test coverage
- documentation quality
- user satisfaction

A major challenge will be to operationalize these aspects into concrete metrics, preferably in a way such that they can be derived automatically. An awesome list with these metrics in place would have several benefits. It could play an important role in pushing the official statistics FOSS landscape towards increased maturity by stimulating developers to improve on these metrics. It would help potential users and contributors to make informed choices. In the ideal situation, it would help to change the software landscape towards more standardization and a limited number of high-quality software components offering maximum functionality to any statistical organization.

4 Summary and conclusion

It is widely recognized that open source is an invaluable resource for modernizing official statistics. This pertains as much to the technical side of using and contributing to FOSS, as to the shift towards an open source culture, where transparency, collaboration, and community are the norm.

In this paper we have highlighted the current state of play internationally and in particular for statistics Netherlands. We have discussed several current international and collaborative projects, including the awesome list of official statistics software and the principles on open source software. These principles are currently considered for endorsement within the international statistical community.

We have identified several strands of work that will help the official statistics community moving forward. These include research on truly independent software components that allow for propagating uncertainty and that support privacy by design.

Finally, we feel that the introduction of a specific set of metrics could help push the official statistics FOSS community towards more uniformity where a small number of composable high-quality software components offer maximum functionality.

ORCID iDs

Olav ten Bosch https://orcid.org/0000-0002-1943-7558
Mark PJ van der Loo https://orcid.org/0000-0002-9807-4686

Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

Declaration of conflicting interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

1. The Journey to Using R – Experience of Central Statistics Office (Ireland), Conference of European Statistics (CES), Geneva, 2023.
2. Sandberg L and Thindberg C. DAPLA – a cloud based statistical production system and its implications for Statistics Norway. In: Conference of European Statistics (CES), Geneva, 2023.
3. Statistics Poland. Open Source, Conference of European Statistics (CES), Geneva, 2023.
4. Kowarik A and van der Loo M. Using R in the Statistical Office: the experience of Statistics Netherlands and Statistics Austria. *Romanian Statistical Review* 2018; 66: 15–29.
5. Publishing Open Source Code at HICC (Statistics Canada). https://canada-ca.github.io/open-source-logiciel-libre/en/open-source-standards.html
6. Ralphs M. Transforming statistical workflows to use open-source technology at the UK Office for National Statistics. In: Conference of European Statistics (CES), Geneva, 2023.
7. Van der Loo M and Ten Bosch O. Free and open source software at Statistics Netherlands. In: Conference of European Statistics (CES), Geneva, 2023.
8. Rahman O (executive ed.) The future of national statistics offices: A call to action. United Nations Economic Commission for Europe (UNECE), 2025 May.
9. Statistical Open Source Software Charter and Report, 2025. https://unece.github.io/OSS/HLG-MOS%20OSS%20Charter.pdf
10. van der Loo MPJ and de Jonge E. Data validation infrastructure for R. *J Stat Softw* 2021; 97: 1–31.
11. van der Loo MPJ and de Jonge E. Data validation. In: *Wiley StatsRef: statistics reference online*. American Cancer Society, 2020, pp. 1–7. doi:10.1002/9781118445112.stat08255
12. van der Loo M and de Jonge E. dcmodify: Modify data using externally defined modification rules. 2024. https://doi.org/10.32614/CRAN.package.dcmodify, R package version 0.9.0, https://CRAN.R-project.org/package=dcmodify.
13. de Jonge E and van der Loo M. errorlocate: Locate errors with validation rules. 2023. https://doi.org/10.32614/CRAN.package.errorlocate, R package version 1.1.1, https://CRAN.R-project.org/package=errorlocate.
14. van der Loo M. simputation: Simple imputation. 2024. https://doi.org/10.32614/CRAN.package.simputation, R package version 0.2.9, https://CRAN.R-project.org/package=simputation.
15. van der Loo M and de Jonge E.rspa: Adapt numerical records to fit (in)equality restrictions. 2022. https://doi.org/10.32614/CRAN.package.rspa, R package version 0.2.8, https://cran.r-project.org/package=rspa.
16. van der Loo M and de Jonge E. deductive: Data correction and imputation using deductive methods. 2025. https://doi.org/10.32614/CRAN.package.deductive, R package version 1.0.1, https://CRAN.R-project.org/package=deductive.
17. de Jonge E and van der Loo M. validatetools: Checking and simplifying validation rule sets. 2023. https://doi.org/10.32614/CRAN.package.validatetools, R package version 0.5.2, https://CRAN.R-project.org/package=validatetools.
18. van der Loo MPJ. Split-apply-combine with dynamic grouping. *J Stat Softw* 2025; 112: 1–21.
19. van der Loo MPJ. Monitoring data in R with the lumberjack package. *J Stat Softw* 2021; 98: 1–13.
20. van der Laan DJ. Reclin2: a toolkit for record linkage and deduplication. *R J* 2022; 14: 320–328. ISSN: 2073-4859.
21. Bogaart P, van der Loo M and Pannekoek J. _rtrim: Trends and indices for monitoring data_. 2024. https://doi.org/10.32614/CRAN.package.rtrim, R package version 2.3.0, https://CRAN.R-project.org/package=rtrim.
22. Tennekes M. Tmap: thematic maps in R. *J Stat Softw* 2018; 84: 1–39.
23. Boonstra H. hbsae: Hierarchical Bayesian small area estimation. 2022. https://doi.org/10.32614/CRAN.package.hbsae, R package version 1.2, https://CRAN.R-project.org/package=hbsae.
24. https://siscc.org/
25. ten Bosch O, van der Loo M and Kowarik A. The awesome list of official statistical software: 100…and counting. In: The Use of R in Official Statistics - uRos202, 2020.
26. European Statistics Code of Practice - Quality - Eurostat
27. https://ec.europa.eu/info/departments/informatics/open-source-software-strategy_en
28. CSPA 2.0 (2023) common statistical production architecture: https://statswiki.unece.org/spaces/CSPA/overview (accessed 08 April 2025).
29. van der Loo MPJ, Pannekoek J and Rijnveld L. *Computational Estimates of data editing related variance.* UNECE Work Session on Statistical Data Editing (The Hague, Netherlands), 2017.
30. ten Bosch O, de Jonge E and Laloli E. To be FAIR, what is missing in Official Statistics?. In: Conference On Smart Metadata for Official Statistics (COSMOS 2024), Paris, 2024.
31. Umm-e-Laila A, Zahoor , Mehboob K, et al. Comparison of open source maturity models. *Procedia Comput Sci* 2017; 111: 348–354.
32. Adewumi A, Misra S and Omoregbe N. A review of models for evaluating quality in open source software. *IERI Proc* 2013; 4: 88–92.
33. Van der Loo MPJ. Computing in the statistical office. *Stat J IAOS* 2021; 37: 1023–1036.