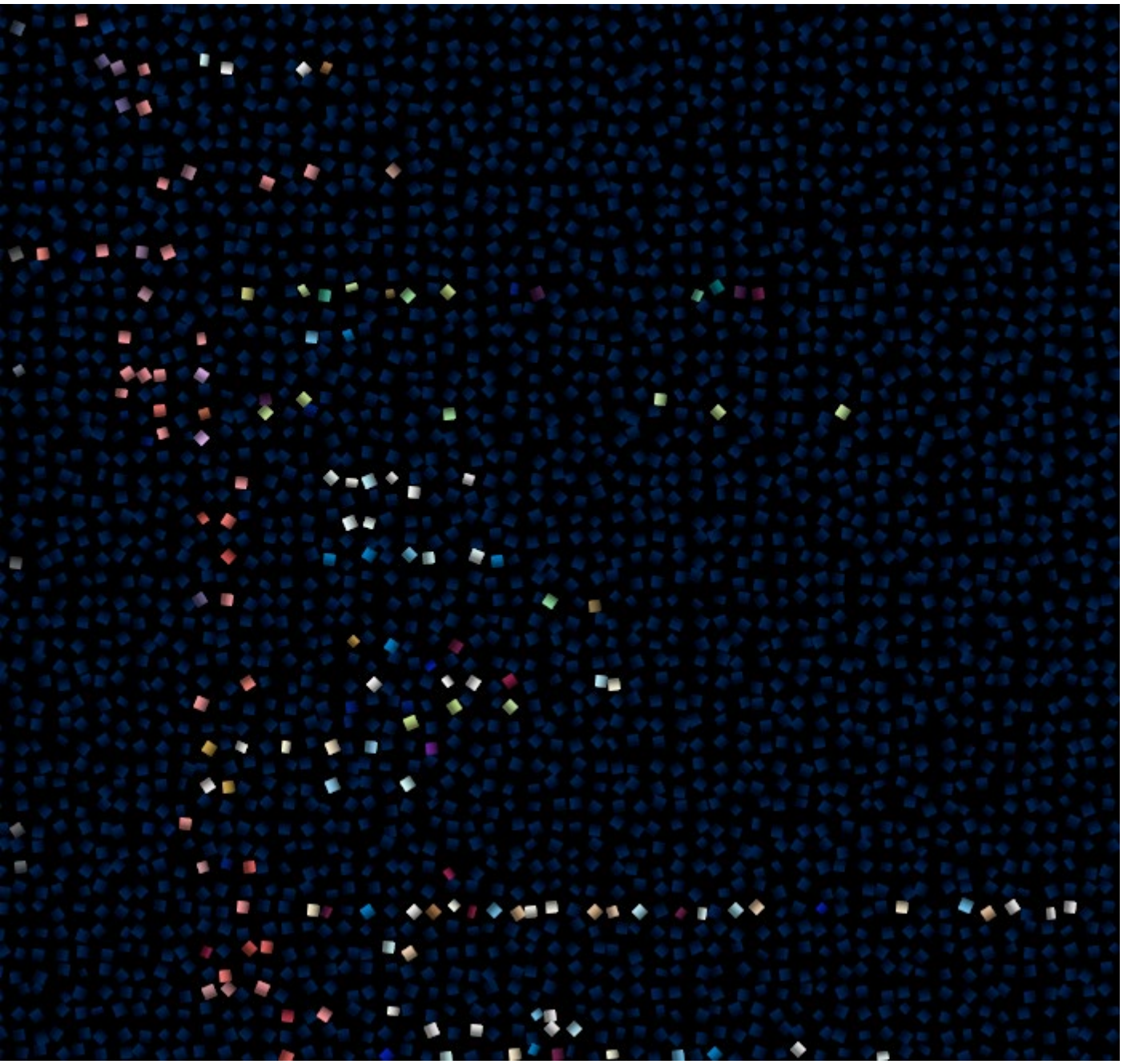
Statistical Open Source Software

Charter and Report



modernstats
by HLG - MOS

This PDF version created was created in February 2025.

A website rendition of this report is available at https://unece.github.io/OSS/

This work is in English.

modernistats
by HLG - MOS

Table of contents

Acknowledgements

This publication represents the main outcome of the Statistical Open Source Software project that was conducted over the 2024 period, and mandated by the UNECE High-Level Group for the Modernisation of Official Statistics (HLG-MOS) following its annual meeting in November 2023.

The project, led by Carlo Vaccari (UNECE Project Manager), was composed of around 30 experts, drawing from national statistics and other institutions as well as international organisations. The following experts kindly dedicated their time and contributed their knowledge, experience, and expertise to this project.

- Craig Lindenmayer (Australian Bureau of Statistics)
- Kate Burnett-Isaacs (Infrastructure Canada), who lead the Governance & Maintenance sub-team
- Mireille Paquette, Li Wang, Christie Glover, & Jonathan Wylie (Statistics Canada)
- Marcello D'Orazio, Lorenzo Asti, Francesco Isidori, Pierpaolo Massoli & Samanta Pietropaoli (Italian National Institute of Statistics)
- Akmaral Tokbergenova & Kairat Kipatov (Statistics Kazakhstan)
- Olav ten Bosch & Mark van der Loo (Statistics Netherlands)
- Pubudu Senanayake & Kevin Townend (Statistics New Zealand)
- Nevena Mitrovic, Aleksandra Skoko Despenic, Mira Nikic & Nikola Orlic (Statistical Office of the Republic of Serbia)
- Karl McKenzie, Martin Ralphs & Ken Rennoldson (UK ONS)
- Matyas Meszaros (Eurostat)
- Jonathan Challener (OECD)
- Iraj Namdarian (Council for Agricultural Research and Economics)
- InKyung Choi & Andrew Tait (UNECE)

The project team expresses its appreciation to SORS for having hosted the project sprint.

Executive summary

Open source software (OSS) has become a transformative force in various fields, and this holds true for official statistics. This publication aims to explore the topic, dispel myths, and provide practical guidance to national statistical offices (NSOs) interested in adopting or developing OSS.

The use of proprietary software can come with expensive per-user licences, and may result in vendor lock-in. Additionally, vendors do not have high incentive to develop for relatively niche communities such as official statistics. OSS, on the other hand, allows communities to develop their own solutions for problems they understand and to do so in the open, thereby building trust with their stakeholders.

At the heart of this publication is the OSS charter, a set of principles that are generally applicable to all organisations involved in the production of official statistics. These principles can help guide NSOs to develop, validate and share statistical methods openly while ensuring reproducibility of official statistics.

This publication provides several case studies which demonstrate the different ways OSS communities can be managed, how offices can adopt OSS, and how software can be developed in the open. These practical examples are intended to act as a guiding light, by clearly showing that working with OSS has real benefits and is a practical option for modern NSOs.

Major topics in OSS are explored and cover licensing, standards, culture, knowledge building, and security issues. Discussions of these topics are conducted with reference to the case studies in order to relate the concepts discussed to real world steps.

Finally, this publication contains three annexes. The first explores the connections between the OSS charter and other frameworks, and the second contains a SWOT analysis on OSS adoption by NSOs, which covers the strengths, weaknesses, threats, and opportunities of working with OSS with honest and open candour in the spirit of OSS. The final annex covers explorations between Official Statistics, OSS, and the field of AI.

List of abbreviations

Abbreviation	Meaning
AI	Artificial Intelligence
API	Application Programming Interface
CLA	Contributor Licence Agreement
CRAN	Comprehensive R Archive Network
DDI	Data Documentation Initiative
EUPL	European Union Public Licence
ESS	European Statistical System
HLG-MOS	High Level Group for the Modernisation of Official Statistics
GPL	General Public Licence
GSBPM	Generic Statistical Business Process Model
GSIM	Generic Statistical Information Model
IST	Istraživanje (Survey in Serbian language)
Istat	Istituto Nazionale di Statistica (Italian NSO)
JSON	JavaScript Object Notation
MIT	Massachusetts Institute of Technology
NSO	National Statistical Office (Organisation)
OECD	Organisation for Economic Co-operation and Development
OS	Open Source
OSI	Open Source Initiative
OSS	Open Source Software
SDMX	Statistical Data and Metadata eXchange
SIS-CC	Statistical Information System Collaboration Community
SORS	Statistical Office of the Republic of Serbia
SWOT	Strengths/Weaknesses/Opportunities/Threats
UNECE	United Nations Economic Commission for Europe
XML	eXtensible Markup Language

```
type: website
output-dir: docs

theme-navigation: true

site:
  title: "HLG-MOS Open Source Software"
  site-path: /oss/
  navbar:
    logo: "assets/img/oss_charter_logo.png"
    left:
      - href: index.qmd
        text: Home
      - href: charter.qmd
        text: "The Principles"
      - case_studies.qmd
      - href: major_topics.qmd
        text: "Major Topics"
      - background.qmd
      - resources.qmd
```

1. Introduction

1.1 What is open source?

There are different approaches that may be taken when developing software, which include developing in the open (i.e., documenting development in a publicly accessible place such as a GitHub repository), making code sharable, and allowing and encouraging collaboration between interested parties. However, what essentially makes software "open source" is whether the source code is freely available for sharing, and whether the code may be modified by users and derivative works created, although some restrictions may apply given the licence that the authors choose for the software.

In order for software to be open source, the authors must specify such by choosing an appropriate licence for the software that allows for free distribution and/or modification. Restrictions can apply with the licence as long as they do not arbitrarily limit the use and modification of the software (e.g., by limiting its use to only certain groups or fields of endeavour).

Further information on open source can be found on the Open Source Initiative website.

1.2 Strengths and weakness of open source for statistical organisations

Open source software (OSS) has emerged as a transformative force in various fields, offering various opportunities for innovation, collaboration. For statistical organisations, OSS presents a unique chance to modernise their processes and develop tailored solutions. Its open nature encourages shared development and community-driven improvements, making it a valuable resource in addressing the evolving needs of statistical work. They also serve as both enablers and catalysts for other technological changes such as cloud and data science that statistical organisations are striving to embrace.

However, the adoption of OSS is not without its challenges. It can be a huge transformation for statistical organisations that can impact their infrastructure, culture, and capacity. While it promises many strengths, such as transparency and flexibility, organisations must also grapple with weaknesses of OSS and external threats that could impact their operations.

In the following section, we highlight the key strengths, weaknesses, opportunities, and threats of OSS in the context of statistical organisations. More detailed description of this analysis can be found in Annex 2. It is important to note that occasionally the **same factor can be both strength and weakness**, which highlights the importance of understanding their duality for making informed decisions and managing trade-offs as an organisation.

1.2.1 Strengths and opportunities

- **Democratisation of development and agility:** OSS allows organisations, regardless of size, as well as any individuals in the organisations to develop software without the requirement of large upfront costs or access to restricted software or technology, fostering a more democratic and scalable development process.
- **Freedom to shape organisational future and meet needs**: OSS provides flexibility to customise software without vendor restrictions or limitations which enables organisations to shape their solutions to fit specific needs.
- **Trust through more transparency**: OSS promotes trust by making the codebase fully examinable and ensuring transparency in data handling and statistical processes.
- **Improvement of quality, interoperability, and standardisation**: Open source development encourages better quality solutions through community contributions, while also promoting interoperability and adherence to open standards.
- **Sense of community and communal development**: OSS fosters a collaborative environment where developers and users contribute to the public good, improving quality, innovation, and developer satisfaction.
- **Cost reduction**: OSS eliminates the need for costly proprietary software licenses, and its collaborative nature results in shared development costs and reduces overall expenses.
- **Alignment with job market trends**: Statistical organisations using OSS are better positioned to attract talent, as open source skills are increasingly available in the job market.

1.2.2 Weaknesses and threats

- **Maintenance and sustainability**: OSS may become a "single point of failure" if key contributors leave or organisations shift priorities, and this can lead to uncertainty in long-term support.
- **Governance complexities**: The absence of a clear governance framework can lead to confusion about roles and responsibilities, especially as projects grow and evolve.
- **Lack of legal expertise**: Statistical organisations may lack the legal expertise required to navigate OSS licensing and intellectual property (IP) issues, leading to potential legal risks.
- **Learning curve and cultural change:** OSS often requires staff to learn new programming languages and adopt a culture of collaborative development, which can be a significant challenge in traditional environments.
- **Hidden costs:** Transitioning to OSS can incur hidden costs related to capacity building, maintenance, and parallel use of legacy systems, as well as unpredictable support from community-driven resources.
- **IP issues:** The open nature of OSS can expose organisations to IP exploitation.
- **Security breaches:** The transparency of OSS may make the software vulnerable to malicious attacks, compromising data integrity and security.

1.3 Purpose of this document

This document is developed to establish the common understanding on Open Source Software (OSS) and facilitate the adoption of OSS in statistical organisations. Based on concrete experiences

from statistical organisations, it explores the common challenges and issues arising as they try to adopt OSS as a user of existing OSS but also as a developer and contributor to OSS.

The document is organised into the following substantive chapters:

- **Chapter 2. OSS Charter**: While the benefits of OSS are widely acknowledged, there is often a different understanding of what adopting and using OSS really entails in practice. This leads to a misalignment of expectations both within organisations and across the official statistics community which can hinder progress. This chapter outlines a set of OSS principles designed to foster a common understanding of the expected behaviour and practices when adopting OSS, which ultimately will help the statistical community to collaborate more effectively.
- **Chapter 3. OSS Case Studies**: This chapter presents a series of prominent case studies showcasing the adoption and development of OSS within national statistical offices (NSOs) and at an international level. These cases illustrate how various initiatives navigate challenges and leverage opportunities in their OSS journeys.
- **Chapter 4. Main topics in OSS and recommendations**: Building on insights from the case studies, this chapter discusses common challenges that arise during the OSS adoption journey such as choosing license, establishing organisation-wide knowledge and capability, and strengthening security measures, and offers practical recommendations.

We hope that the in-depth exploration of main topics in OSS, illustrative case studies, and recommendations help statisticians and IT specialists with the development, deployment, and maintenance of OSS solutions.

It is important to note that, as highlighted throughout the document, adoption of OSS is not just a technical issue of making the codebase open. It requires the engagement of diverse stakeholders (not just direct users of the software, but also business people whose work is influenced by the tools). The transition to OSS necessitates a cultural shift and change of mindset which are often significant barriers in organisations. Therefore, it is critical to secure the buy-in and practical understanding of OSS adoption of both high-level managers, who shape organisational strategy and long-term investment, and the middle managers, who are responsible with oversight of specific departments or projects.

In the field of official statistics, statistical organisations have a strong tradition of collaborating on innovations and modernisation efforts. They have successfully shared the experimental burden of emerging technologies and exchanged valuable experiences and lessons learned. The adoption of open source software relies even more heavily on cooperation and community-driven support, reflecting the ethos of openness, shared knowledge, and collective problem-solving in OSS. We hope this document provides a helpful guide for statistics organisations to conduct their own thorough analysis when evaluating the adoption of open source solutions and navigating their own OSS journey.

2. OSS charter

2.1 Opening statement

We recognise open source software (OSS) as essential for modern statistical production, promoting transparency in methodology and fostering international collaboration in developing and supporting the production of official statistics.

Open source solutions enable national statistical offices (NSOs) to develop, validate, and share statistical methods while ensuring reproducibility of official statistics. The transparent nature of open source software allows for peer review of statistical procedures, thereby strengthening the credibility of official statistics. Sharing software as open source software (OSS) is consistent with the transparency principle of the Fundamental Principles of Official Statistics (principle 3) and the National Quality Assurance Framework (principle 6).

Reuse of software assets across organisations in the statistical process chain is beneficial. Reducing duplication of efforts through co-investments increases efficiency, and sharing statistical code and tools between NSOs creates a collaborative ecosystem that accelerates innovation in official statistics, while facilitating methodological harmonisation across countries. These approaches ensure efficient use of public resources while maintaining the independence and scientific integrity of national statistical systems. Therefore, by adopting and developing open source tools, NSOs can build flexible, cost-effective statistical infrastructures that can adapt to new and emerging data sources and methodological innovations, while building trust through transparency in statistical production.

A statistical open source community is most effective and innovative if it works from a common understanding across statistical organisations of the underlying drivers for open source. For this reason, it is necessary to identify the basic principles underlying open source in official statistics.

We therefore endorse using the following Principles on Open Source Software in official statistics in both the production of software, and the adoption of software for statistical production.

2.2 The Principles

Introduction

This document lists a number of principles for OSS.[1] They are considered to be generally applicable to all organisations involved in the production chains of official statistics. This includes NSOs as well as international organisations that provide official statistics. These principles are kept as generic as possible, meaning that they are technology-agnostic (applicable to all platforms and programming languages) and domain-agnostic (applicable to software in all statistical

[1] The principles were initially developed as "ESS Principles on Open Source Software" (https://os4os.pages.code.europa.eu/pbbp/principles.html) by the group on Open Source for Official Statistics (OS4OS) and adjusted for the global context.

domains). Finally, they are formulated in general terms, meant as a starting point for the development of more concrete implementation details in the further development towards an effective and mature statistical open source community.

Principle 1. OSS by default

Statement

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.

Rationale

This principle contributes to the core values of official statistics, such as transparency and independence in the way we produce statistics and striving for high quality and reproducibility. Using and sharing open source software increases the transparency of our work and avoids black boxes in the implementation of official statistics.

Implications

This means that when implementing, redesigning or creating new processes, open source software solutions have preference. Only when no viable open source solutions exist should an NSO deviate from the standard OSS option. Likewise, sharing as open source is the default, but it is possible to deviate from this in justified cases. For NSOs this means that the methods used in the production of official statistics are not only described, but also that the code used to actually apply the method is shared as OSS. For international organisations this means openness about how international aggregates are computed via OSS solutions.

Principle 2. Work in the open

Statement

We start our projects in the open from the beginning and clearly mark maturity status.

Rationale

Many projects have the intention to publish results as open source but have difficulty deciding on the best time to do so. It might feel uncomfortable to put early ideas and rough implementation sketches on-line, but on the other hand sharing it too late prevents others from providing valuable comments and ideas or volunteering to work together on the project. To circumvent this dilemma, we start working in the open right from the beginning wherever possible and clearly mark and update our project's development phase over time.

Implications

This means that it is recommended and accepted to start development projects in the public domain. We clearly show the development status, which may vary from pre-alpha to stable and

proven by showing a public roadmap, public source code repository, a public backlog of features, issues, bugs etc.

Principle 3. Improve and give back

Statement

We improve existing open source solutions rather than decide to create new solutions and we give our improvements back to the respective open source community.

Rationale

There are cases where existing open source solutions do not cover one-to-one the functionality needed in official statistics. The quickest way to address this is to copy a solution, adapt it and use it. However, the improvements made in the original solution will not be merged into the copy and our improvements made to the copy will not be visible in a wider context. Therefore, we strive to give back our improvements to the open source community as change requests or suggestions even if it takes additional resources to do so. In the end, this is an investment in the effectiveness and efficiency of the official statistics community as a whole.

Implications

This means that statistical organisations actively search for solutions that can be reused instead of having to create new solutions themselves. Even if a solution does not exactly fit the required functionality, it can be examined for how it could be improved while keeping the intended functionality in mind, or even extending such functionality. This also applies for partial solutions such as code snippets and models (including machine learning models) that could be valuable for others. The changes or enhancements should be tested, documented, and returned to the respective community to decide on whether to integrate them into their solution.

Principle 4. Think generic statistical building blocks

Statement

In our open source work we strive for re-usable generic functional building blocks that support well-defined methodologies in statistical processes.

Rationale

Publishing source code as open source is not sufficient by itself for effective reuse in the global official statistics community. It is necessary to think about the design of what is to be shared and to identify generic statistical building blocks that can be used in different contexts. Therefore, we design the software from the point of view of the intended users and in such a way that it can be reused in as many statistical domains or organisations as possible. This helps maintain complex statistical processes and high-quality official statistics.

Implications

This means that monolithic applications are componentised as much as possible into generic configurable statistical building blocks. We put statistical functionality into code and make statistical expertise configurable. We make these components as generic as possible in time, across statistical domains and across statistical organisations. For individual NSOs this means that not just *its own* statistical production process should be kept in mind when developing tools, but also the possible *wider applicability*. International organisations should actively encourage the development and sharing of generic OSS solutions within their domain of expertise.

Principle 5. Test, package and document

Statement

We test, package and document our open source software for easy reuse.

Rationale

Re-using generic statistical software in the official statistics community is not always easy due to differences in statistical processes, technological environments, and ways of working. Testing our software for functionality and security and packaging our software with good documentation is of utmost importance as it improves the chances of reuse. General purpose package management systems offer versioning and documentation facilities to share generic statistical software. The use of such packaging systems helps to maintain complex statistical processes and ensure high-quality official statistics.

Implications

This means that we invest in testing, security scanning, packaging and documentation to enable reuse. Security patches are applied as soon as possible. Documentation is designed from the point of view of a statistical user, keeping it concise, understandable but also complete and covering at least the basic functionality and a complete API reference. Packaging is a key success criterion for open source projects. Larger projects should adopt modern approaches such as containerisation, automating as much as possible. Smaller projects may also follow these practices. Each package is downloadable without registration, can be installed with minimal effort, and has a minimal viable example that can be executed. Dependencies are managed and minimised as much as possible. Versioning is implemented according to the principles of the respective package exchange platform with a preference for semantic versioning. Security patches are implemented with priority. For individual NSOs this means that published OSS software is maintained and updated according to the policies of the relevant platforms, e.g., CRAN. International organisations should play an active role in sharing knowledge about testing and packaging, as well as documentation policies in their domain of expertise.

Principle 6. Choose permissive

Statement

We choose the most permissive OS licence possible for sharing our software.

Rationale

Re-using software is in the common interest of the official statistics community. Reuse of our software not only enhances efficiency but also improves the quality of the software by allowing the wider user community to contribute to its development and maintenance. To maximise reuse by others it is necessary to choose an OS licence that maximally allows reuse, and minimises conflicts with other licences. This is known as “permissive”. When choosing the appropriate OS licence we strive for maximum reuse.

Implications

This means that when sharing software we opt for a permissive licence (e.g. Apache 2.0/MIT) over a “copyleft” licence, taking into legal, organisational and societal considerations. Mandatory acknowledgement / attribution of sources and authors is a viable additional option.

Principle 7. Promote

Statement

We invest in promoting new developments or improvements of our open source software within the official statistics community, and where applicable in a wider context.

Rationale

Reuse of generic software will not happen if no one knows what can be reused. On the other hand, it is difficult to know beforehand what the value of our software is for others. The only way forward is to communicate, even if we have no idea whether it can be used in a wider context. We promote our software in an honest and concise way, mentioning its core functionality. We let the public know our plans for new developments and improvements and are open to suggestions for improvements.

Implications

This means working together on communication facilities targeted at the open source community. A community-driven approach of sharing knowledge, tackling possible OS building blocks, and the application of OS in statistical production should be preferred to centrally maintained repositories. A centrally maintained repository of software tools can quickly become outdated, and collecting and organising information from the community can be an immense effort. Therefore, such a repository should be maintained by the community as a whole. For individual NSOs this means actively participating in the OSS community by attending events, joining relevant forums, etc. International organisations should play an active role in the organisation of the statistical OSS community in their domains of expertise.

3. Case studies

In the evolving landscape of software development and data management, open source software (OSS) stands out as a cornerstone that embraces collaboration, innovation, and transparency. This chapter delves into a series of prominent case studies highlighting the adoption and development of OSS within national statistical offices (NSOs) and internationally, shedding light on how these initiatives navigate through various challenges and opportunities. Additionally, the case study on the Awesome List project showcases a different facet of OSS utility and community engagement.

The first part of our exploration focuses on how OSS has empowered NSOs through the adoption of sophisticated data cleaning tools and specialised R packages developed by Statistics Netherlands and Istat respectively. These tools not only streamline the data handling processes but also improve the accuracy and reliability of statistical outputs. Through these case studies, readers will gain insights into practical OSS deployment strategies and the consequent enhancements in data management within national frameworks.

We then examine two case studies of OSS development that exemplify successful international collaboration but at each end of the open source spectrum. The Statistical Information System Collaboration Community (SIS-CC), a mature open source community operating for several years, and the Statistical Office of the Republic of Slovenia (SORS), who are now taking the first steps from a closed community model for a piece of software to an open source one. Both illustrate how collaborative efforts can lead to robust solutions that serve multiple countries and cultural contexts, thereby optimising the functionality and reach of official statistics organisations.

In contrast to the other case studies, the Awesome List project represents a unique initiative. This endeavour involved curating a list of OSS tools, categorising them according to their utility and model of openness. This case study not only highlights the diversity and richness of OSS tools available but also serves as a vital resource for organisations keen on adopting open source solutions.

These case studies also demonstrate how the OSS principles outlined in Chapter 2 are played out in real world examples. The key aspects essential for the adoption of OSS in statistical organisations that commonly arise from these case studies and how to address associated challenges are discussed in Chapter 4.

3.1 Building a FOSS ecosystem for statistical data processing

Statistics Netherlands

3.1.1 Introduction

Statistics Netherlands (CBS) has embraced the use of Free and Open Source Software for statistical production for more than a decade. A significant step in that direction was taken in 2010 when R was adopted as a strategic tool for data processing.[2] R is both a software tool for statistical data processing and a programming language that is extensible through R packages. These packages are typically published on the Comprehensive R Archive Network, which enforces a strict quality policy on code quality, documentation, and interoperability of packages.

Recognising the need for production systems that are built out of composable and reusable (generic) modules, researchers at CBS started to both use and contribute packages to the R ecosystem. A major part of those contributions consist of R packages in the area of statistical data cleaning and data processing.[3,4]

Over time, these packages have been adopted both within Statistics Netherlands and outside. Within CBS, packages are used in the production of areas covering social and economic statistics, agriculture, international trade, education, environmental statistics, emissions, income, shipping, Short-Term-Statistics, recreation, museums, and many more. Outside statistics we have seen uptake of the packages by statistical institutes in countries such as Iceland, Denmark, Italy, and Brazil. It is noteworthy that the US Department of Agriculture National Agricultural Statistical Service (USDA-NASS)[5] is using CBS R packages to validate and clean data from large national surveys under American farmers.

3.1.2 The R based ecosystem

The current ecosystem (Table 1) consists of a number of packages that integrate seamlessly. Not only because there is a shared technical platform (i.e., R), but also because careful thought was put in pegging out, with formal precision, what each fundamental processing step entails. Below we describe two examples demonstrating the extensibility and power of this modular approach.

The first example concerns *data validation*. Until about 2024, the act of checking data against domain knowledge, in the form of validation rules, was not recognised as a separate activity. In almost any available system this was either hard-coded by users or integrated in a larger data-editing system. Creating a separate package (called *validate*) with the sole purpose of defining, manipulating and executing data validation rules yielded the possibility of monitoring the progress

[2] Alexander Kowarik, & Mark van der Loo (2018). Using R in the Statistical Office: the experience of Statistics Netherlands and Statistics Austria. *Romanian Statistical Review, 66(1), 15-29.*
[3] Jeroen Pannekoek, Sander Scholtus, & Mark Van der Loo (2013). Automated and Manual Data Editing: A View on Process Design and Methodology. *Journal of Official Statistics, 29(4), 511-537.*
[4] Jonge, E., & Loo, M. (2018). *Statistical data cleaning with applications in R*. John Wiley & Sons. ISBN: 978-1-118-89715-7
[5] USDA - National Agricultural Statistics Service (https://www.nass.usda.gov/)

of data quality along multiple statistical value chains using a single piece of software.[6,7] Moreover, the rule management system of the package is reused in packages for error localisation (*errorlocate*), data correction (*deductive*) and aggregating based on dynamically defined data groupings (*accumulate*).[8,9,10]

A second example is an imputation package (*simputation*) that allows users to combine (i.e., chain) a large number of popular imputation models in fall-through scenarios that are often used in economic statistics.[11] 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 and 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.

Table 1. R-based open source ecosystem for statistical data processing

R package	Description
validate	Check validity of data based on user-defined rules.
dcmodify	Adapt erroneous data based on user-defined rules.
errorlocate	Find the minimal number of erroneous data points.
simputation	Many different imputation methods with a single easy to learn interface.
rspa	Adjust numerical records to fit equality and inequality restrictions.
deductive	Solve data errors, using the data and validation rules.

[6] van der Loo, M. P. J., & de Jonge, E. (2021). Data Validation Infrastructure for R. *Journal of Statistical Software*, *97*(10), 1–31. https://doi.org/10.18637/jss.v097.i10
[7] van der Loo, M. P. J., & de Jonge, E. (2020). *Data Validation*. https://doi.org/10.1002/9781118445112
[8] van der Loo, M. P. J., & de Jonge, E. (2023). *errorlocate: Locate Errors with Validation Rules*. R package version 1.1.1, https://CRAN.R-project.org/package=errorlocate
[9] van der Loo, M. P. J., & de Jonge, E. (2021). *deductive: Data Correction and Imputation Using Deductive Methods*. R package version 1.0.0, https://CRAN.R-project.org/package=deductive
[10] van der Loo, M. P. J. (2024). Split-Apply-Combine with Dynamic Grouping. *Journal of Statistical Software (Accepted for publication)*
[11] van der Loo, M. P. J. (2022). *simputation: Simple Imputation.* R package version 0.2.8, https://CRAN.R-project.org/package=simputation

validatetools	Find contradictions and redundancies in rule sets.
accumulate	Grouped aggregation, where grouping is dynamic and data-dependent.
lumberjack	Automatically track changes in data for logging purposes.[12,13]
reclin2	Join datasets based on (multiple) possibly inexact keys.[14]

All packages have been developed in the open, by hosting the code on open version control repositories (i.e., 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 standardised release platform (i.e., CRAN), using permissive licences 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 generalise 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 contributing. 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.

3.1.3 How working in FOSS contributes to the success of the ecosystem

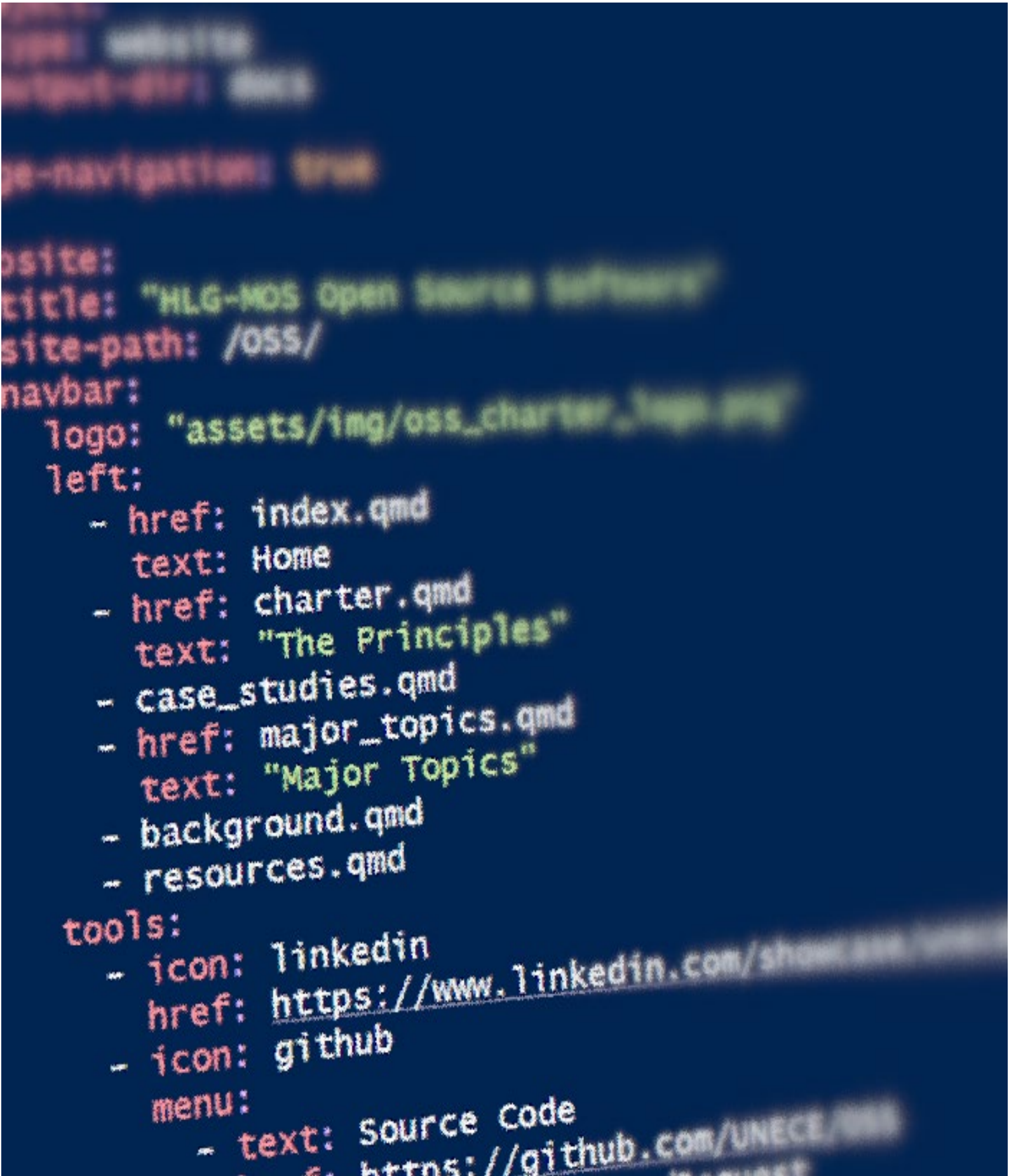
The fact that the whole ecosystem has been developed in the open, as an open source project, has contributed crucially to the success on several levels.

In the first place there is a large FOSS community in R that provides help, a packaging system, and also an infrastructure for developing, testing, documenting and publishing packages. The fact that this infrastructure, including the presence of a global informal and helpful community, exists has been extremely helpful during design and development of the packages. The usefulness of being able to directly contact the people who develop the infrastructure one is using is hard to overstate.

Secondly, publishing the packages in open source immediately gives a large audience the opportunity to try the product: *given enough eyeballs, all bugs are shallow.*[15] Enthusiastic users are

[12] van der Loo, M. P. J. (2021). Monitoring data in R with the lumberjack package. *Journal of Statistical Software 98, 1—11*
[13] van der Loo, M. P. J. (2021). A Method for Deriving Information from Running R Code. *The R Journal 13, 42-52*
[14] van der Laan, D. J. (2022). reclin2: a Toolkit for Record Linkage and Deduplication. *The R Journal, 14(2), 325-333*
[15] Linus's Law: https://en.wikipedia.org/wiki/Linus%27s_law

able to contribute new use cases, issues, and documentation, which overall has contributed to creating more general and more robust software.



3.2 Adoption of OSS and governance efforts in Istat

Italian National Institute of Statistics (Istat)

3.2.1 Introduction

Outside of statistical software, Istat has a long experience of using open source software and already in 2004 an official working group on the use of open source software was formed. The group, made up of statistical and IT experts, studied the current uses of OS software and its potential developments. Under the impetus of the group, open source software spread throughout Istat, from Linux servers to connectivity software and various modules to support group work.

As for the statistical software, in the past Istat's statistical work was carried out exclusively with SAS software, but for specific tasks some additional SAS tools had to be developed (e.g., MAUSS for deciding the optimal sample size and its allocation in multi-purpose surveys; GENESEES for weight calibration and estimation in sample surveys). For data processing and imputation, it was decided to use the SAS-based commercial tool BANFF, developed by Statistics Canada. Obviously, it was difficult to disseminate SAS-based tools to the members of the Italian National Statistical System (NSS) because, in most cases, they did not use SAS.

The introduction of R took place shortly after the year 2000, mainly in the Methodology Directorate, to test new methods; initially an informal group of R experts was created and met once a month; this group started to give workshops and occasional internal training (regular training started in 2007). These activities led to the identification of R as the preferred environment for developing new packages, also to replace tools developed in SAS or as stand-alone applications; the decision was also taken in response to a ministerial directive in April 2003, inviting government agencies to adopt open source software tools and avoid dependence on a single commercial software tool.

The first R packages were released around 2010 (ReGenesees for weights calibration and estimation; SamplingStrata for stratification and optimum allocation; SeleMix for selective editing; StatMatch for statistical matching), others followed later (e.g. FS4 for stratification, R2BEAT for determining optimal sample size and its allocation). In other cases, such as the RELAIS record linkage system, R became the engine for statistical computations. These packages were developed in the Methodology Directorate, and to facilitate their dissemination within the NSS and also externally, it was decided to create a repository on the corporate website.[16] Packages to be included in the repository had to pass a rigorous approval procedure, which required the availability of clear documentation, a presentation to a committee and also the adoption of an EUPL licence, as recommended by the Istat Legal Office. Later, the licence requirements were relaxed and the GPL licence was also accepted, being the most popular in the R community. Some of the R packages are also distributed on CRAN and are easily accessible in the “Official Statistics” task view.[17]

[16] https://www.istat.it/en/classifications-and-tools/methods-and-software-of-the-statistical-process/
[17] https://cloud.r-project.org/web/views/OfficialStatistics.html

Today, the activities related to R are focused on the maintenance of existing packages, which have recently experienced some disruptions due to the retirement of their developers/maintainers, and on the testing of already existing external packages (instead of developing new ones). For example, Istat is testing the R packages for data editing and imputation (validate, validatetools, errorlocate, simputation[18] and VIM) with the aim of adopting them to replace our obsolete standalone tool CONCORDJava. Similar tests are underway for disclosure control (R packages ptable and cellKey). Finally, we are working on extending the functionality of the R package RJDemetra, which provides the R interface to JDemetra+, the officially recommended seasonal adjustment software within the European Statistical System.

R training for staff is provided regularly by in-house trainers; there are two "core" courses ("Base R" and "Intermediate R") offered twice a year, and a number of short courses on specific topics/packages (e.g., the ggplot2 package) offered once a year. In addition, R tools are used in statistical courses (sampling, data integration, data processing and imputation, etc.) to demonstrate the application of methods. More generally, the developed packages and their improvements are promoted through presentations and tutorials at conferences/workshops, including the uRos[19] annual conference.

Recently, research aimed at investigating the potential use of statistical learning and more generally machine learning techniques for official statistics, also with the aim of exploiting alternative data sources such as big data, led to the adoption of Python. As with R, the approach is bottom-up, as Python is mainly used in the Methodological Directorate. Python is currently being used to produce some experimental statistics: the Social Mood on Economy Index (SMEI) and the imporT ExpoRt networK Analysis (TERRA), a tool for exploratory analysis of Eurostat data on international trade. Other ongoing projects are quite diverse: web mining to integrate and validate information from the Statistical Business Register; estimation of road accidents using big data; estimation of urban greenery using remote sensing images; imputation of education levels in the Register of Persons; estimation of shipping routes, etc., although it is already being used to produce some experimental statistics. A few years ago, it was decided to organise Python courses (basic and advanced) for staff, offered once a year.

Today, R and Python are the main languages used in the Methodology Directorate, but SAS remains the tool used in some technical units involved in the production of statistics. This is due to a number of factors: ageing staff in production units well trained in SAS and unwilling to lean and move to R; reduction of staff in production due to the inability to replace retired staff and consequently limited resources to ensure a complete migration from SAS to R; the fear of production managers of disrupting the publication of official statistics because of the introduction of new tools with limited support; and an absence of a structure ensuring support on R, contrary to what happens for SAS.

[18] https://github.com/data-cleaning
[19] https://r-project.ro/conferences.html

3.2.2 Governance

For these reasons, it was recently decided to increase efforts to ease the adoption of open source statistical tools in production directorates. The informal network of R experts will be expanded to include methodologists and subject matter experts with a solid knowledge of R and will become the official support structure for R from 2025 onwards. This network will also support the implementation of innovative methodologies (including the development of new packages) and the increase of training opportunities for staff. It should also contribute to the maintenance of packages already developed, also to avoid the problems recently experienced (i.e., the retirement of maintainers).

A first part of the Istat policy for the governance of the statistical open source software tools is expected to be published by the end of 2024-early 2025. It will mainly deal with statistical-methodological aspects and provide some basic indications on the IT infrastructure, as the two elements are closely linked. The second part, more focused on IT infrastructure and IT requirements, will be published later.

The first part of the governance of statistical open source software will provide Istat researchers with a set of guidelines and recommended practices for the development of new tools or the adoption (and adaptation) of existing ones; finally, it will redesign the procedures for their approval and dissemination (including guidance on licensing) and maintenance over time.

The guidelines for the development of new open source statistical software will be published in late 2024 and will include recommendations for writing R code and documenting it according to international standards to facilitate code sharing and reuse. A second level of documentation, tailored for internal purposes only, will document the use of the tools in the specific production process to facilitate modifications/adaptations to specific circumstances encountered in a subsequent replication of the process.

In the development of the code, much attention is paid to dependency issues in order to limit problems related to changes/disruptions in the maintenance of the packages on which the code depends. This issue will be even more relevant for the adoption and possible adaptation of externally developed open source statistical tools; guidance will be provided on the procedure to be followed: criteria for selection among different potential candidates; preliminary checks to be performed (availability and clarity of documentation; maturity, frequency of updates and possible bug fixes; limitations and dependencies on other tools; available support, etc.); and testing procedures of the selected tools in case studies of increasing complexity. For both developed and acquired/adapted existing tools, the governance policy will provide guidance on their endorsement, dissemination and promotion internally and externally.

Governance will also provide recommendations for user support, maintenance and updating of the approved tools over time. All defined procedures should be complemented by the identification and establishment of a set of governance bodies with clearly defined roles and responsibilities.

3.3 Building a community-driven OSS: The SIS-CC experience

Organisation for Economic Co-operation and Development (OECD)

The statistical information system collaboration community (SIS-CC) is **a reference open source community for official statistics**, focusing on product excellence and delivering concrete solutions to common problems through **co-investment and co-innovation**.

3.3.1 Licences

SIS-CC's adherence to the principles of openness and collaboration is embodied in its strategic choice to use the MIT licence for its open source tools. MIT is a permissive free software licence and places minimal restrictions on the reuse of code, thereby maximising flexibility and fostering an environment where innovation can thrive.

The decision to select MIT over alternative open source licences, such as Apache 2.0, is rooted in the desire for simplicity and minimal legal complexity. While both licences are permissive and encourage open contribution, there are differences that made MIT a better choice for SIS-CC. For example, the concise language and straightforward terms avoid legal jargon, making it very accessible for users to understand and implement correctly. There are fewer restrictions on the redistribution of software compared to Apache 2.0, which requires explicit attribution and changes documentation among other stipulations. With its minimalist approach, MIT is broadly compatible with other licences, allowing for greater interoperability of code across various projects and jurisdictions. It enables easier participation from the community because contributors do not need to worry about complex licence compliance, which can be more challenging with Apache 2.0. By opting for the MIT licence, the SIS-CC reduces the barriers to entry for users and contributors, thereby encouraging widespread adoption and collaboration on the open source project.

While embracing the simplicity of MIT, SIS-CC also recognises the importance of managing contributions effectively. One mechanism for doing this is through Contributor Licence Agreements (CLAs) that clarifies the terms under which a contributor submits code or content to a project, protecting both the contributor and the organisation by ensuring that the intellectual property is appropriately managed. However, for SIS-CC, it was deemed too complex as it would add an unnecessary overhead to the contribution process, with the potential to deter casual contributors. Instead, the SIS-CC opted for a more automated and centrally controlled review and merge process whereby source code, reuse of libraries, and other components, are checked and validated for potential breaks in the licence chain. So far this has served the SIS-CC well and facilitated a number of contributions from outside of the core maintainers of the project.

3.3.2 Standards

The SIS-CC plays a pivotal role in driving and promoting the adoption of global open standards within the statistical community, specifically focusing on the Statistical Data and Metadata Exchange (SDMX) and the Generic Statistical Business Process Model (GSBPM). These standards are essential for streamlined and accurate data management across diverse systems and organisations. Through the adoption of these standards, the SIS-CC has enhanced the efficiency, accuracy, and comparability of statistical data through standardised practices, fundamentally

altering how data is managed and exchanged globally. Adopting these standards has increased **efficiency and cost-effectiveness** through streamlined data processes, **fostered consistency and comparability** in data across various systems, enhancing overall data integrity, and **facilitated collaboration** within a standardised framework that has simplified data sharing and collaboration among statistical organisations. The adoption of SDMX and GSBPM has prepared organisations to meet future data challenges and integrate into the global statistical ecosystem effectively. The SIS-CC's robust initiative to standardised statistical practices has not only enhanced operational efficiencies within member organisations but also strengthened the global statistical community's capability to handle modern data demands, fostering a more connected and resilient statistical landscape. SDMX's role in facilitating a seamless communication and collaboration across not just statistical teams but also IT within organisations highlights its capacity to catalyse multidisciplinary teamwork, by automating data flows and enhancing user-friendly data exploration, which lays the groundwork for an efficient data-sharing environment.

Powered by SDMX, the .Stat Suite, being the SIS-CC flagship product, has revolutionised how data is managed, processed, and disseminated from end-to-end, making it more accessible and easier for researchers and analysts to combine and connect in analytical work. The SIS-CC has already started to explore the integration of Artificial Intelligence (AI) with SDMX and the .Stat Suite which promises to unlock even more potential. As AI capabilities evolve, SDMX's robust semantic framework can serve as a foundation for intelligent, automated data flows, and fostering innovations.

3.3.3 Knowledge Building

The commitment of SIS-CC to enhancing data skills and knowledge led to the establishment of the .Stat Academy. This initiative represents a comprehensive effort to democratise access to self-paced online training, with a focus on enhancing the knowledge of data practitioners in data modelling and SDMX, as well as data toolers in the technologies and tools needed to support the statistical lifecycle. Through a diverse array of free online courses and resources, the .Stat Academy is empowering data professionals worldwide. It leverages a blended learning approach of online courses, hands-on workshops, and collaborative projects to facilitate learning. This multifaceted approach ensures that participants gain practical experience alongside theoretical knowledge. By offering courses on a wide range of topics, the .Stat Academy caters to varying levels of expertise and professional needs, fostering a vibrant, global community of data practitioners who share insights, challenges, and solutions, thus collectively advancing the field of statistical information systems.

3.3.4 Culture

The journey from a closed community software development to an open source model represented a profound cultural shift for the SIS-CC. This evolution demanded a paradigm change, where openness, transparency, and collective engagement became the cornerstones of development. As SIS-CC members transitioned to open source practices, they embraced a culture that championed collaboration beyond organisational confines, paving the way for more innovative solutions. This shift confronted traditional viewpoints which emphasised proprietary control, urging a reorientation towards shared stewardship and a belief that pooling resources can lead to better outcomes.

Adopting an open source culture necessitated overcoming numerous challenges. Developers had to recalibrate their approaches to software development—acknowledging that the broader community can contribute valuable insights and code improvements that no single entity could achieve alone. It required rethinking strategies around intellectual property, where inclusivity in innovation assumes priority over exclusivity. The shift to DevSecOps demanded a cultural overhaul where the team worked collaboratively across the entire development cycle, breaking down traditional silos. It required nurturing a mindset that places equal emphasis on speed and security, embedding security considerations from the onset of development rather than being an afterthought. To deal with the resistance that such a profound change brings demanded a concerted effort in training, change management, and the establishment of new norms.

3.3.5 Governance

The SIS-CC operates within a multi-tier community ecosystem, underpinning a sustainable business model that promotes co-innovation and co-investment. This multifaceted governance structure is essential for managing the dynamics of collaboration among diverse organisations and partners. Central to its governance is the multi-tier community ecosystem, meticulously designed to foster balanced user growth while ensuring the retention of agility and the maintenance of product excellence. This structure categorises members into different tiers based on the nature and extent of their contribution. **Tier 1 organisations** are pivotal, providing financial and in-kind contributions. This tier has the potential for receiving additional grants, reflecting a deep investment in the community and product advancement. **Tier 2 organisations** benefit either through commercial avenues or through institutional backing, like that provided by the ILO LMIS[20] project. This ensures a diverse range of organisational types and resources are contributing to and benefitting from the community. **Tier 3 organisations** access the Community products through self-service tools, such as Gitlab, documentation, the .Stat Academy, and issue tickets, enabling broader, more accessible participation.

The governance philosophy of the SIS-CC champions an inclusive strategy, laying its foundations on the Community Foundations: Community Driven Dynamics; Open source Delivery; Full Data Lifecycle; Componentised Architecture; and Systematic User Research. These foundations are critical in a shared journey towards achieving mutual objectives, enhancing collaboration, and solidifying our collective value proposition.

[20] https://ilostat.ilo.org/resources/labour-market-information-systems/

3.4 Transforming a software into OSS at SORS

Statistical Office of the Republic of Serbia (SORS)

3.4.1 Introduction

The Statistical Office of the Republic of Serbia (SORS) has over the past 15 years developed the Istraživanje (IST),[21] a metadata driven statistical collection and production solution aligned with GSBPM, to meet the growing needs of statistical data processing in a rapidly evolving technological environment. The IST platform was developed in response to the complexity of statistical data handling, which requires a flexible, modular, and metadata-driven approach to ensure efficiency across the statistical lifecycle.

Since its launch in 2006, IST has been deployed in several NSOs, including those in Kyrgyzstan, Montenegro, Bosnia and Herzegovina, and Albania. The platform was shared through Memorandums of Understanding (MoUs) and provided to partner NSOs as free software. As part of these agreements, SORS supplied the full source code and ongoing support, further demonstrating its commitment to international cooperation and statistical innovation. Over time, IST evolved through continuous feedback from these partners, further refining its features and capabilities.

Collaborative Effort

Although IST was entirely developed by SORS, its expansion and refinement were facilitated by collaborations with other NSOs. These partnerships enabled SORS to adapt IST to different country-specific contexts while ensuring that the core system remained robust, flexible, and adaptable. The software's success lies in support for various file formats, languages, its modular design (which allows for integration with additional tools widespread among NSOs), and its comprehensive metadata management features that guide the statistical process from data collection to dissemination.

Key Features of IST

IST is a metadata-driven system designed to ensure seamless data management throughout the statistical process. Its architecture allows for real-time data entry, validation, and processing, making it an essential tool for modern statistical operations. The key features of IST include:

- **Metadata-Driven Architecture**: IST leverages metadata to control data processing workflows. This design ensures consistency and efficiency across different stages of data collection, validation, and analysis.
- **Modular Design**: The platform's architecture allows independent modules to be added or modified without affecting the entire system. This flexibility makes IST scalable and adaptable to various statistical requirements.

[21] http://istportal.net

- **Advanced Reporting Capabilities**: IST supports real-time reporting in multiple formats, such as Excel, JSON, CSV, TXT, and XML, facilitating easy data dissemination.

3.4.2 Steps Toward Open Source

In recent years, SORS has recognised the importance of promoting open source solutions to foster collaboration and enhance the usability of IST across multiple regions. As part of this strategy, SORS has initiated a comprehensive plan to release IST as an open source product, adhering to widely accepted licensing and contribution practices.

IST Productisation Steps

The transition to open source requires a structured and phased approach. In collaboration with international partners and legal experts, SORS is devising the plans for making IST open source:

1. **Defining Governance and Contribution Models**: A key aspect of the open source transition is the establishment of clear governance models based on the foundations of openness, collaboration, and sustainability. SORS aims to implement a Contributor Licence Agreement (CLA) to ensure that all contributions are legally bound to be licenced back to SORS, preserving control over the project's direction (lead by the Project Management Committee) while encouraging community contributions.
2. **Technical Documentation and Support**: IST's open source version will be accompanied by comprehensive documentation, including user and developer manuals. SORS will continue to provide remote support to partner NSOs to ensure smooth implementation and adoption of the open source version.
3. **Licensing Strategy**: Following a thorough review of different open source licences, SORS decided to use the Apache licence 2.0 with additional clauses to restrict the commercial redistribution of IST. This decision ensures that IST remains freely available for statistical offices and developers, while retaining legal protections over its intellectual property. The final decision will be made after thorough consideration by legal experts.
4. **Publishing and Maintenance**: IST's source code will be published on an open source platform, such as GitHub / GitLab, allowing for easy access and collaboration. The platform will also serve as a space for tracking issues, managing contributions, and providing updates.

3.4.3 Licences

Two open source licences were considered for IST's transition to open source: the **MIT licence** and the **Apache licence 2.0**. After careful evaluation, SORS opted for the **Apache licence 2.0**, which offers both permissive use and patent protection. This licence allows IST to be widely adopted while safeguarding SORS's intellectual property rights.

Key factors influencing this decision include:

- **Patent Protection**: Apache 2.0 includes an explicit patent grant, offering legal protection against patent-related disputes.

- **Broad Usage and Modification Rights**: The licence allows other statistical offices and developers to use, modify, and distribute IST, while ensuring that all modifications are licenced back to SORS.
- **Attribution**: All derivative works and redistributions must include an attribution notice recognizing SORS as the original developer.

Contributor Licence Agreement

To manage contributions effectively, SORS plans to implement a Contributor Licence Agreement (CLA). The CLA will ensure that all contributions to IST are licenced back to SORS, allowing the organisation to maintain control over the software's development while promoting collaboration with the wider open source community.

The key elements of the CLA include:

- **Modification Rights**: Contributors can modify IST for their internal use.
- **No Redistribution**: Contributors are not allowed to redistribute IST or its modified versions to third parties.
- **No Commercial Use**: IST and its derivatives cannot be used for commercial gain without explicit consent from SORS.

3.4.4 Conclusion

The transition of IST to an open source product represents a significant move by SORS to promote innovation and collaboration in the field of statistical data processing. While the specific licensing model for IST is still under consideration, SORS is carefully evaluating options, including the Apache licence 2.0, MIT and other permissive licence, to ensure the best balance between fostering global collaboration and protecting the integrity of the software.

By developing a flexible CLA, SORS aims to create a structured and collaborative environment where contributions from the broader statistical community, including universities, researchers, and developers, can enhance IST while ensuring its continued alignment with SORS's goals. As the transition progresses, SORS remains committed to maintaining IST's reputation as a modern, scalable, and secure solution for statistical data management, ensuring that it continues to meet the needs of NSOs worldwide.

3.5 Sharing OSS across communities - The Awesome List for Official Statistics Software

Statistics Netherlands

3.5.1 Introduction

Open source software offers significant benefits for producing official statistics, including cost savings, improved quality, greater flexibility, and the potential to foster standardisation. However, navigating the vast landscape of open source software packages already available within the statistical community can be challenging. Understanding which software exists, its maturity level, and its suitability for specific tasks is crucial for the reuse of statistical building blocks.

To address this challenge, in 2017 a number of conference participants including members of Statistics Netherlands started the *awesome list of official statistics software*.[22] It is a community approach to facilitate knowledge sharing among statistical organisations and to promote the adoption of open source solutions. The list quickly grew from the software the initiators were involved in into an extensive catalogue of mature open source solutions in the ESS.[23]

3.5.2 Explanation of the awesome list of statistical software

Sharing statistical software among institutes has been a valuable practice for years, particularly in areas like disclosure control, data editing, collection, and dissemination. While a few well-established solutions have been widely adopted, the current software landscape for official statistics is far more complex and dynamic than in former years. Numerous specialised packages are continually being developed, making it challenging to maintain a comprehensive overview and increasing the risk of redundant development.

To address this issue, the awesome list of official statistics software was created, inspired by the popular concept of community-driven knowledge sharing in the 'awesome list concept'.[24] This list serves as a valuable resource for discovering and utilising generic official statistics software. The list has grown significantly since its inception, now featuring over 130 open source packages that are readily available, well-maintained, and actively used by statistical offices worldwide. It includes packages for automated access to official statistics output and is itself developed and maintained in an open source spirit.

The concept of this list is not to replicate information but as much as possible to link to the information maintained 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

[22] https://github.com/SNStatComp/awesome-official-statistics-software
[23] Olav ten Bosch, Mark van der Loo, Alexander Kowarik, (2020). *The awesome list of official statistical software: 100 ... and counting.* The Use of R in Official Statistics - uRos202
[24] Awesome list concept by Sindre Sorhus: https://github.com/sindresorhus/awesome

latest version, last commit, and licence. This information is automatically extracted from the packaging system metadata, ensuring consistency and ease of use.

- GitHub v.2.1 | last commit june 2021 | license GPL-3.0
  Python Social-Media-Presence. A script for detecting social media presence on enterprises websites. By Statistics Poland.
- 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 1: Open source software on the awesome list.

To provide users with a clear understanding of the software's applications on the list, it is organised according to the Generic Statistical Business Process Model (GSBPM). Figure 2 illustrates the distribution of the 135 items across the various GSBPM processes.

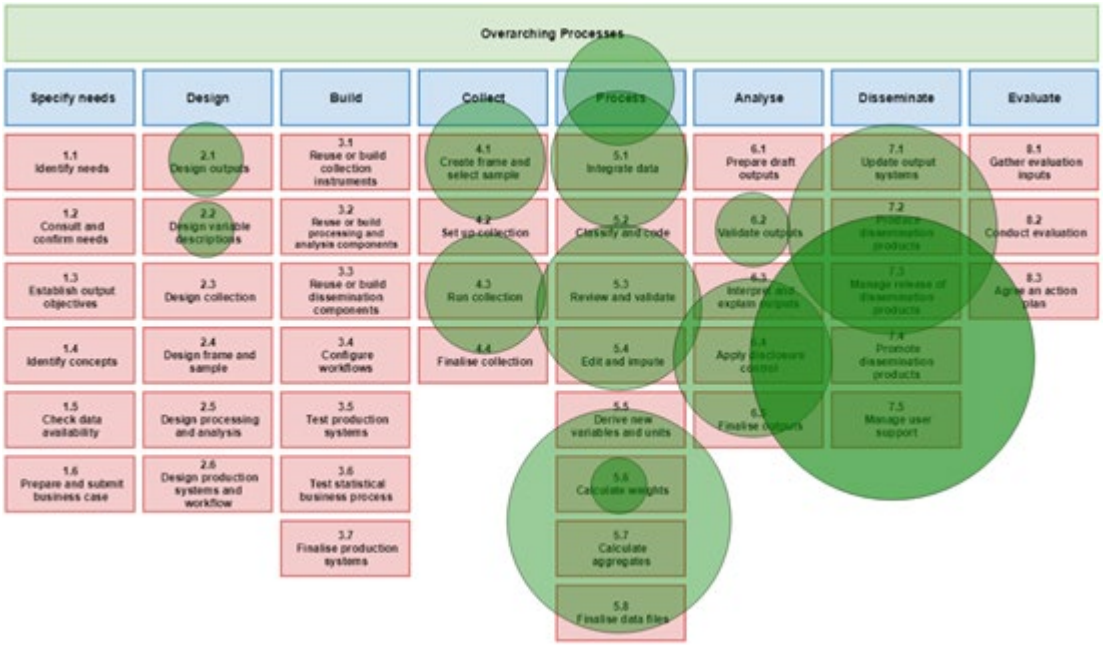


Figure 2: Software on the list by GSBPM

Figure 3 shows the distribution of programming languages of items on the list. The vast majority of items are written in R, which shows the excellent software sharing methods in this community. Figure 4 shows the licences used on the list. GPL is the most popular licence followed by MIT and EUPL.

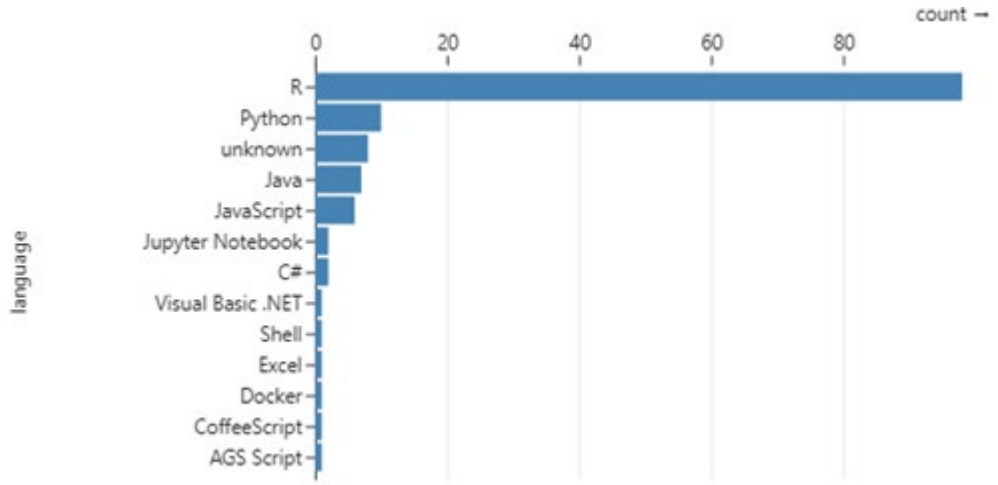


Figure 3: The most popular programming languages among items on the list.

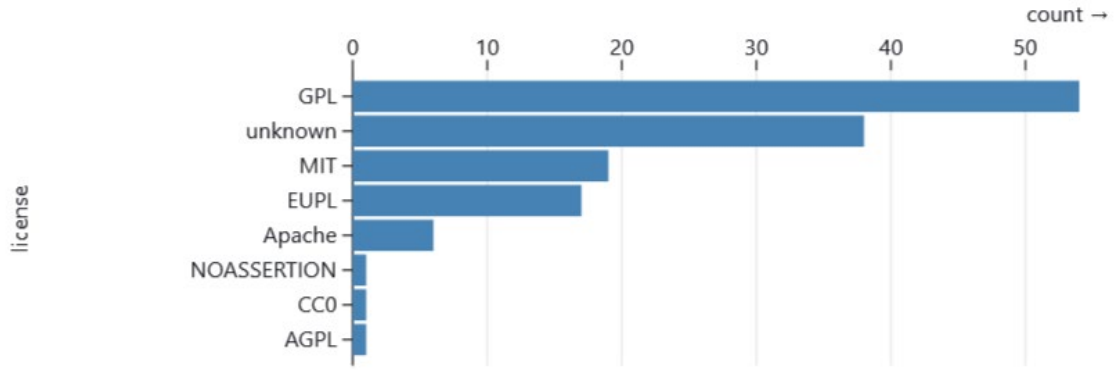


Figure 4: Open source licences of software on the list.

3.5.3 How the awesome list contributes to the FOSS in official statistics

The list has proven to be a valuable tool for sharing knowledge about existing open source solutions among different user communities in the ESS, either methodologists, statisticians, IT specialists, or management.[25] Suggestions for changes or additions come from statistical organisations around the world and functionality suggestions for adding compatibility information, maturity indicators, and popularity metrics are documented on the GitHub repository itself. Concluding we can say that this list, maintained by many, has already helped software reuse

[25] van der Loo, M., & ten Bosch, O. (7 2023). Free and Open Source Software at Statistics Netherlands. *Conference of European Statisticians, Seventy-First Plenary Session, Geneva, 22-23 June 2023*

among statistical organisations and as long as the community actively uses and maintains it, it will continue this important role.

4. Major topics in OS and recommendations

In this chapter, we dive into the core aspects of the open source ecosystem, which is not only about software but also involves a rich tapestry of legal, operational, and cultural dimensions. We explore six main topics:

- **License considerations** provide the backbone of open source, defining how software can be used, modified, and shared.
- **Standards** that ensure software integration and innovation, but also critical for data consistency, accuracy, and interoperability in statistical processes.
- **Culture** that explores the misconceptions and concerns about OSS.
- **Knowledge building** being the communal learning and sharing process that lies at the heart of open source projects.
- **Governance** that provides the structures and practices overseeing decision-making.
- **Security** addressing the strategies and challenges in protecting open source software against potential vulnerabilities.

4.1 Licences considerations

The Open Source Initiative (https://opensource.org/licences) provides detailed information about all the officially approved open source licences. However, choosing an open source licence among these can still be tricky because it involves balancing multiple competing interests and considerations such as:

1. **Long-term implications**: Your choice is often permanent and affects how your project can be used, modified, and distributed.
2. **Legal complexity**: Different licences have subtle but important differences in terms of rights, obligations, and protections they provide.
3. **Trade-offs**: You need to balance:
    - Project adoption vs. control.
    - Commercial potential vs. open source principles.
    - Protection of your rights vs. user freedom.
4. **Ecosystem considerations**: Your choice needs to be compatible with:
    - Dependencies you're using (components and software used).
    - Platforms you are targeting (Operating Systems, Clouds, CRAN, GitHub, …).
    - Community expectations.
    - Local/regional considerations (like EUPL for European countries).

One important grouping is between “viral” and “permissive” licences. Viral/copyleft licences like GPL and EUPL require any derivative work to also be open source under the same licence, thus the licence requirements cascade throughout the entire project and its derivatives. Permissive licence like MIT and Apache on the other hand allow the code to be used in any way, including in closed-source projects, thus each user can choose their own licence. This implies different business impacts; viral licences can make commercial adoption more challenging and ensure ongoing

openness while permissive licences are more business-friendly and allow others to create closed-source versions if desired.

The choice of open source license varies within the official statistical community, as highlighted by the license usage analysis[26] of tools in the Awesome list (Chapter 3.5). The most popular are GPL (approx. 50%), MIT (approx. 20%) and EUPL (approx. 15%). The choice of individual organisation is influenced by multiple factors.

For example, Istat mainly use the EUPL 1.1 licence (for some R tools developed by Istat are available under GPL license) for the following reasons:

1. The EUPL licence is the only licence translated in Italian.
2. EUPL is the only licence "approved" and sponsored by European regulations and therefore legally valid in Italy.

SIS-CC releases .STAT suite (Chapter 3.3) under the MIT licence. They chose a permissive licence for following reasons:

1. **International collaboration**: The MIT licence facilitates easy sharing and collaboration between statistical offices and organisations across different countries.
2. **Public sector alignment**: Many statistical organisations are public sector entities, and the MIT licence aligns well with government open source policies.
3. **Integration flexibility**: Statistical systems often need to integrate with various existing tools and systems - the MIT licence makes this legally straightforward.
4. **Community building**: The permissive nature of the MIT licence encourages participation from a wide range of contributors.

SORS at the current time is defining its open source policy (Chapter 3.4); they will use a permissive licence like Apache. As SORS wants to manage a community of NSOs working on its software tools, the reasons for choosing a permissive licence are similar to those of SIS-CC.

4.2 Standards

Open source software (OSS) has strong connections with the adoption of open standards in different fields. The presence of established standards facilitates the adoption of OSS, while the use of OSS, in turn, promotes the broader adoption and development of those standards. Here some of standards that play important roles:

4.2.1 Programming languages:

- **Open source programming languages, in particular R,** have been instrumental in the adoption of open source policies within national statistical offices (NSOs) by providing a trusted, versatile platform specifically tailored for statistical analysis and data processing. The ecosystem of R packages[27] supports a wide range of statistical and data science tasks,

[26] https://observablehq.com/@olavtenbosch/visualizing-awesomeofficialstatistics-org
[27] The best example is the software repository CRAN at https://cran.r-project.org/

making it possible for NSOs to replace or complement proprietary software with reliable, community-validated tools. Additionally, the R community's emphasis on reproducible research fosters a standardised approach to analysis and reporting, which enhances collaboration and long-term sustainability in NSOs' statistical workflows.

- **Consistency and Compatibility**: Standardising on widely-used programming languages such as R and Python[28] ensures compatibility across various statistical and data science tools. These languages have extensive libraries and frameworks that simplify the integration of different tools and applications, making it easier for organisations to share and collaborate on code.
- **Community and Ecosystem Support**: Popular open source languages have large, active communities that continually expand and improve their ecosystems. This community support ensures regular updates, a wealth of resources, and broad compatibility, which is particularly valuable for organisations relying on open source tools.
- **Training and Knowledge Sharing**: Adopting standardised programming languages streamlines training for new staff and facilitates knowledge sharing within and between organisations. Staff trained using Python or R can more easily adapt to similar environments, promoting a more unified skill set across NSOs, academia and other public agencies.

4.2.2 Data formats

- **Interoperability Across Systems**: Standard data formats like CSV,[29] JSON,[30] and XML[31] enable data to be shared and processed by multiple systems without extensive reformatting. Using open, well-documented data formats ensures that datasets remain accessible and usable across different platforms and applications, fostering smooth data exchange.
- **Long-Term Accessibility and Data Preservation**: Open data formats reduce the risk of data obsolescence and vendor lock-in. CSV and JSON, for instance, can be easily accessed and parsed by any software, which is crucial for long-term data usability. Open formats also ensure that future tools can read and interpret historical datasets without conversion issues.
- **Data Quality and Validation**: Standardised data formats often come with tools or built-in capabilities for data validation, helping organisations maintain high data quality and consistency. For example, JSON and XML formats allow schema definitions that can be used to enforce data structure and integrity.

4.2.3 Exchange protocols

- **Global Compatibility and Integration**: Exchange protocols such as SDMX are specifically designed for statistical data, ensuring interoperability across statistical organisations

[28] A good third-party software repository is the Python Project Index (PyPI) at https://pypi.org/
[29] An awesome list: https://github.com/secretGeek/AwesomeCSV
[30] JSON resources can be found at https://github.com/burningtree/awesome-json?tab=readme-ov-file , https://ajv.js.org/ and https://www.json.org/json-en.html
[31] An awesome list: https://github.com/StanimirIglev/awesome-xml

globally. By adhering to these protocols, statistical offices can integrate their data with international platforms and share information that are compatible with other countries and organisations, facilitating global data collaboration and analysis. In the SDMX website there is a rich list of software tools[32] to support SDMX[33] implementers and developers. Also authentication and authorisation standard protocols like Oauth[34] can support the integration of systems.

- **Efficient Data Sharing and Real-Time Access**: Standard data exchange protocols, such as SDMX, RESTful APIs,[35] and OData (Open Data Protocol),[36] allow data to be accessed, shared, and updated in real time. These protocols make it easy for systems to communicate and exchange data quickly, which is essential for large-scale data dissemination and integration across different statistical platforms.

4.2.4 ModernStats standards

- By adhering to UNECE ModernStats standards (e.g., GSBPM, GSIM), NSOs can build modular, compatible open source solutions that support efficient, standardised workflows and foster collaboration within the global statistical community.
- Open source tools designed to align with GSBPM can fit seamlessly into existing workflows, as they adhere to recognised standards for data processing, validation, and analysis. The use of a common process model facilitates the adoption of compatible levels of granularity between different statistical packages.
- The use of GSIM as common reference for metadata platforms increases the compatibility and the possibility of integration between software packages used in the different phases of the statistical process.

4.3 Culture

Transition to open source software requires cultural shifts in organisations, particularly if closed source/proprietary offerings have been the norm for a long period of time. In addition, the adoption of open-source principles demands a certain ethos and culture of work to be present within the organisation.

Several common misconceptions about OSS contribute to hesitancy and resistance within organisations:

- OSS is of a lower quality as it is free (with the rationale that things that are paid for must be of higher quality or value).

[32] Tools for SDMX implementers and developers https://sdmx.org/?page_id=4500
[33] The Bank for International Settlements publishes another website with SDMX resources at https://www.sdmx.io/
[34] https://oauth.net/2/
[35] An Awesome list is available at https://github.com/marmelab/awesome-rest
[36] Tools and resources can be found at https://www.odata.org/

- OSS is not seen to have the same level of support as that provided by vendors of proprietary software, who are responsible for ensuring their products quality and providing support if problems arise.
- OSS is changeable by anyone so there is no reliability or consistency in the tools.
- OSS is vulnerable to security threats, whereas proprietary software is safer given the commercial concerns of vendors (even though many organisations already use tools and software that are already built on OSS such as cloud computing solutions, security software).

These misconceptions, combined with existing organisational practices, contribute to cultural barriers and concerns that arise during the transition to OSS, such as:

- Risk aversion: If there are existing tools and software, changes are perceived as risky, especially by long standing staff and operational management, where the production of outputs may be considered to be at risk.
- Resource concerns: NSOs may face challenges in allocating budgets for implementation, integration, and ongoing maintenance, underestimating the advantages that can come from shared development with other developers. A change in general requires resourcing, and an active decision to pursue. It is easier to allow passive indecision to propagate the status quo.
- Inexperience with the wider OSS community, and the large-scale adoption of OSS across technical spheres.
- Trepidation about the capability required to maintain an internal code base, or tooling of versions adopted from OSS, or maintain a tool itself.
- Preference for stability and control: The dependency on proprietary tools may create a sense of comfort. The focus of open source tools is often on evolving features and flexibility, which can seem less stable or controlled to NSOs.

Addressing these concerns is critical for NSOs to build a sustainable OSS culture across organisations. In the subsequent chapters, we discuss some of key challenges and strategies to address them through: knowledge building (Chapter 4.4), establishing governance (Chapter 4.5) and adopting security practices (Chapter 4.6).

4.4 Knowledge building

Below are general recommendations for national statistical offices (NSOs) about knowledge sharing, based on the use cases provided and focusing on capacity building, documentation, and training:

Capacity building through participating in collaborative networks:

- Encourage open source familiarity: Promote the adoption of widely-used open source tools like R and Python by integrating them into the organisation's workflows. Use structured training programs to make the staff familiar with these tools and their applications in statistical work.

- Establish centres of excellence: Create dedicated teams or units within the NSO to specialise in open source tools and methodologies. These teams can act as internal consultants, providing technical support and sharing expertise across departments.
- Leverage community collaboration: Actively participate in international and regional communities such as SIS-CC or open source forums to build institutional knowledge by learning from other NSOs' experiences. Collaborative knowledge sharing accelerates capacity building and reduces redundant efforts.
- Leverage partnerships with academic institutions and international organisations to enhance knowledge transfer, ensuring access to the latest methodologies and technologies.

Example: SIS-CC facilitates capacity building by aligning members around shared tools and standards like SDMX, promoting collective development and mutual learning.

Comprehensive and accessible documentation:

- Develop and maintain comprehensive and consistent documentation for all statistical processes, methodologies, and software implementations. Use frameworks like GSIM to ensure structured documentation, enhancing clarity and usability for internal and external stakeholders.
- Where possible, share documentation publicly to foster transparency and collaboration. This includes sharing metadata standards, data schemas, and technical manuals for open source tools used or developed by the NSO.
- Regularly update technical manuals and process documentation to ensure continuity, especially when staff turnover occurs. NSOs should prioritise institutionalising knowledge over reliance on individual expertise.
- Use open metadata standards such as DDI and SDMX to document statistical processes and data workflows comprehensively, enabling consistent understanding and reuse across organisations.

Example: IST metadata-driven system by SORS supports standardised documentation, enhancing internal and regional knowledge sharing for effective statistical production.

Ongoing training and skill development:

- Offer regular training programs to build staff capacity in open source tools and modern statistical methodologies, ensuring teams remain adept at using and maintaining evolving technologies and methodologies.
- Design training to address varied skill levels, from foundational workshops for beginners to advanced sessions for specialists that emphasises practical applications in statistical workflows. Modular training ensures staff across departments can develop relevant skills at their own pace.
- For NSOs transitioning from proprietary to open source systems, implement mandatory training to ensure staff can effectively use and adapt to new tools. Istat's systematic training for R users is an effective example.
- Collaborate with international organisations, universities, or private sector experts to access specialised training and knowledge-sharing opportunities.

Promote a culture of peer-to-peer knowledge exchange:

- Encourage knowledge sharing between employees through internal forums, mentorship programs, or cross-departmental collaborations as well as organising workshops, hackathons, or collaborative projects where staff can work together on open source solutions, sharing insights and expertise. Such initiatives help disseminate expertise and bridge skill gaps across the organisation.
- Create opportunities for staff to present their innovations and solutions internally and externally, fostering an environment where shared contributions are valued and rewarded.

Example: The Data Clean ecosystem in Statistics Netherlands thrives on community contributions, enabling the exchange of modular, reusable tools that expand collective expertise.

Develop centralised knowledge repositories:

- Implement centralised knowledge repositories to store and share documentation, training materials, FAQs, case studies and user feedback, making these resources readily accessible to all staff and external collaborators. This ensures institutional knowledge is preserved.
- Use platforms like wikis, shared drives, or open source platforms to manage and distribute this information, ensuring continuity even when staff turnover occurs.

Example: The Awesome List of Official Statistics Software acts as a shared resource for NSOs, providing a curated repository of open source tools and their applications.

Support open knowledge sharing practices externally:

- Actively contribute to open source communities by sharing code, documentation, and lessons learned as well as by participating in relevant international, regional and national expert meetings (e.g., *The Use of R in Official Statistics* conferences).[37] This practice not only helps the broader community but also positions the NSO as a leader in statistical innovation.

Example: The SIS-CC's .Stat Academy engages in sharing knowledge through online training, focusing on data modelling and SDMX. The availability of free training resources to support the statistical lifecycle acts as a powerful stimulus for the diffusion of open standards and software tools in the official statistics community.

[37] http://r-project.ro/conferences.html

4.5 Governance

4.5.1 The two governance models

In the case-studies chapter we saw different governance models used by the projects: we could analyse them following the *Cathedral* and *Bazaar* models, defined by Eric S. Raymond in his influential essay, The Cathedral and the Bazaar.

The **Cathedral** model is a structured, centralised approach where a core team of developers maintains the control over the project, releasing updates after extensive internal testing to ensure stability and polish. The process is more closed, with limited involvement from the broader community until a version is ready for public release. This model emphasises planning, longer development cycles, and **top-down** decision-making.

In contrast, the **Bazaar** model is a decentralised, open approach that thrives on community participation and rapid iteration. Code is developed transparently, with contributors working simultaneously, often releasing small updates frequently. Decisions are made collaboratively, encouraging contributions from anyone, which allows the project to evolve quickly. This model embraces an agile, **bottom-up** development style that prioritises community feedback and adaptability over centralised control.

Coming to our use-cases, from one side we have the SIS-CC community (Chapter 3.3), organised with Strategic Level Group, Management Level Group and Architecture Task Force, where the NSOs can choose among three different levels of participation and the development is supported by a defined vision and by a set of training courses available on the web.

At the other extreme we have communities like the Awesome List or the Data Cleaning communities (Chapter 3.1 and Chapter 3.5), where users propose their own packages, there are no predefined strategies or developments, each user is free to test the products and possibly participate in their development. The community is held together only by the interest in software that can be used in the processes of official statistics organisations.

Also at the level of the NSOs, we can see two different governance models: the Serbian one (Chapter 3.4) is more “structured”, the software released is controlled by the “official” structure and the developments are managed by the IT department in collaboration with the users. In Istat (Chapter 3.2), the packages released are managed by a group of methodologists that publish their packages on public platforms and follow their development with a sort of volunteering.

4.5.2 Recommendations on governance of open source software

Below we list some recommendations on governance that are valid for all models:

Define clear objectives for governance:

- Establish a governance framework tailored to the NSO's specific needs and objectives, ensuring alignment with statistical priorities such as data quality, security, and compliance.

- Define which aspects of software development require centralised control (Cathedral) and which can benefit from community-driven contributions (Bazaar).

Encourage stakeholder engagement:

- Develop mechanisms to engage statisticians, IT experts, and external contributors in governance discussions.
- Use forums, workshops, and collaborative platforms to gather input on governance policies and software priorities.
- Establish a supportive environment for external contributors by providing clear guidelines, documentation, and mentorship opportunities.

Apply governance models differently as needed:

- Apply the Cathedral model to critical components such as data security, metadata management, and compliance tools while using the Bazaar model for auxiliary tools and add-ons.
- Distributed autonomy: delegate governance of less critical projects or extensions to trusted community members while retaining oversight over core functions.

Establish quality assurance processes:

- Introduce robust quality assurance mechanisms to ensure all software contributions meet high standards for statistical accuracy, usability, documentation, and security.
- Require peer reviews for contributions to Bazaar-style projects and maintain a centralised team for final validation.

Risk management:

- Develop a comprehensive risk management strategy to address potential issues such as untested contributions, security vulnerabilities, and dependency risks.
- Regularly audit community-driven code for vulnerabilities and integrate automated testing tools into the development pipeline.

Knowledge building and sharing:

- Provide training for all NSO staff on OSS governance, development practices, and community engagement.
- Facilitate the exchange of experiences, challenges, and solutions among NSOs to improve OSS governance.
- Create shared resources, such as guidelines for adopting and managing OSS, tailored to the needs of statistical organisations.

Transparency in governance:

- Publish on the web governance policies, decision-making processes, and project roadmaps to ensure transparency and trust.

- Clearly communicate the criteria for adopting, rejecting, or modifying community contributions.

Leverage international standards and guidelines:

- Ensure that OSS governance aligns with international statistical standards, such as GSBPM, GSIM, and SDMX, to maintain consistency and interoperability.
- Collaborate with international organisations to standardise governance approaches and share collective expertise.

4.6 Security

Security by obfuscation refers to the practice of relying on secrecy or obscurity of system details, such as source code or algorithms, to protect software or systems from threats. This practice provides a false sense of security and does not address underlying vulnerabilities. It can be easily circumvented by skilled attackers who can reverse-engineer the code.

Open source software takes a different approach to security. The source code is publicly available, allowing anyone to inspect, modify, and enhance it. This transparency can lead to more robust security for several reasons:

- Many eyes principle: The idea that "*given enough eyeballs, all bugs are shallow*" (called Linus's Law) suggests that the more people who review the code, the more likely vulnerabilities will be discovered and fixed. This collaborative effort can lead to more secure software.
- Community contributions: Open source projects benefit from contributions from a global community of developers. This diverse input can lead to more innovative security solutions and faster identification of vulnerabilities.
- Transparency: Open source software promotes transparency, which builds trust among users. Users can verify the security of the software themselves or rely on community reviews and audits.
- Rapid response: Open source communities can quickly respond to security issues. Once a vulnerability is identified, patches and updates can be developed and distributed rapidly.
- Peer review: Open source projects often undergo rigorous peer review processes. Code changes are scrutinised by multiple developers, reducing the likelihood of introducing new vulnerabilities.

Open source is no more a security risk than proprietary software, and due to the code being open to inspection there are grounds for arguing it is in fact more secure. However, what it does have in common with proprietary software is a potential vulnerability to supply chain attacks. This is where rather than attacking the software itself, other packages (or services) imported or used are attacked instead. These may be proprietary or open source themselves. Defending against such attacks is too large a topic for discussion in this document, but no further security measures other than those that an organisation should be following regardless are required when using OSS in lieu of proprietary software.

Below are general recommendations for NSOs regarding security issues in adopting and developing open source software, with some references to the use cases presented. While these recommendations are interconnected, we present them separately to provide clarity and focus on specific aspects of security. Further discussion of security related issues can be found in Annex 2: SWOT analysis of OS adoption in NSOs.

4.6.1 Secure Adoption of Open Source Software Recommendations

- **Security assessments:**
  - Perform regular security audits of open source software before adoption. Assess for vulnerabilities in dependencies, outdated libraries, and compliance with data protection laws (e.g., GDPR) or standards (e.g., ISO/IEC 27001). Where possible, automate package vulnerability scans in a CI/CD pipeline or internal package repository.
  - Tools: In the R-based ecosystem, managing third-party dependencies is critical; ensure packages are well-maintained and secure. Tools like SonarQube for static analysis or Snyk for dependency scanning can help identify vulnerabilities early.
- **Establish secure deployment practices:**
  - Use containerisation tools like Docker to isolate open source applications and ensure consistent and secure environments.
  - Provide guidance on best practices for securely configuring and maintaining systems, as highlighted by the Awesome List use case.
  - Consider adopting Configuration as Code (CaC) tools such as Ansible or Terraform to automate and enforce secure configurations.
- **Ensure data confidentiality:**
  - Implement anonymisation and pseudonymisation techniques for sensitive data before processing it with open source tools. This is particularly important for platforms handling confidential statistical data.
  - Tools: Use tools like ARX for data anonymisation or OpenDP for implementing differential privacy to ensure data confidentiality.
- **Leverage open source trustworthiness:**
  - Use software with permissive and well-documented licences (e.g., Apache 2.0 or MIT), as chosen by SORS and SIS-CC in the related case studies, ensuring no hidden restrictions or licensing issues compromise data security.
- **International cooperation among NSOs:**
  - Enhance open source security by enabling shared expertise in code reviews, vulnerability identification, and the implementation of best practices. The collective effort ensures thorough scrutiny of software, reduces security risks, and promotes alignment with international standards.
  - By pooling resources and collaborating, NSOs build more robust and resilient open source solutions that benefit the global statistical community.
  - Tools: Collaboration platforms like Common Vulnerabilities and Exposures (CVE) or shared vulnerability databases can be leveraged to track and mitigate security threats across borders.

4.6.2 Secure Development of Open Source Software Recommendations

- **Embed security in the development lifecycle:**
  - Adopt practices, like DevSecOps, to integrate security at every stage of the software development lifecycle. This ensures that vulnerabilities are addressed proactively rather than reactively, as emphasised by SIS-CC's centralised review and merge process.
  - Utilise tools like Jenkins with security plugins, and integrate automated security scanning in CI/CD pipelines.
- **Implement contributor governance:**
  - Use Contributor Licence Agreements (CLAs) or similar governance tools to ensure all contributions to open source projects are secure, vetted, and legally compliant. This strategy is effective for ensuring that external contributions to tools like IST remain secure.
- **Limit dependency risks and conduct supply chain security:**
  - While open source software presents numerous security benefits, it is susceptible to supply chain attacks. This involves attackers targeting external dependencies rather than the software itself.
  - Minimise reliance on external libraries or ensure dependency management by selecting only well-maintained and frequently updated packages (Istat's focus on dependency stability for R packages is an example of such pre-emptive security in development).
  - Mitigation strategies includes:
    - Regularly assess dependencies for security patches and updates.
    - Regularly scan for vulnerabilities in dependencies using tools like OWASP Dependency-Check or GitHub Dependabot.
    - Use trusted repositories and verify cryptographic signatures for all third-party packages.
    - Employ strategies such as dependency pinning and careful management of version upgrades.
- **Monitor threats and leverage community vetting:**
  - Continuously monitor for emerging threats within the open source ecosystem. Leverage tools like Threat Intelligence Platforms (TIPs) or community-driven security alerts to stay informed.
  - Encourage open community feedback and peer review to identify and fix security vulnerabilities. The collaborative nature of the Data Clean ecosystem and the Awesome List ensures wider scrutiny and faster resolution of potential security issues.
- **Establish a clear incident response plan for addressing security vulnerabilities in open source software:**
  - This should include guidelines for quickly patching vulnerabilities, communicating risks, and coordinating with affected stakeholders. Consider automated patch management systems to minimise response time.

5. Concluding remarks

Open source software at its heart is about freedom, flexibility, and community. It allows NSOs to use, adapt, and develop technical solutions that they require in the manner that is best suited to their needs. The problem with freedom is that it is daunting. Using OSS means that NSOs have to make choices concerning governance models of their software, they must determine how to foster the right culture in themselves for OSS to flourish, they need to consider how best to document development and share knowledge, and so on. Put simply, everything is in their own hands.

An important aspect when using and developing OSS is that this is not a journey that needs to be taken alone. Many NSOs have already spent many years working with OSS and their work and documentation, and through their conferences, online lists, and repositories, one can find like-minded individuals and offices, and learn from and collaborate with them.

As trust in data becomes an increasingly prominent issue in an age of social media and disinformation, working in the open presents an opportunity to build trust in official statistics.

Ultimately, what an office can get out of OSS depends on the effort they put in, but the possibilities are limitless.

Finally, it should be noted that there are growing connections between the OS field and the AI community. The principles outlined in the charter of this document can also be applied when using and developing OS AI systems, thereby assisting with the mitigation of security and trust concerns by the public, and allowing NSOs to take advantage of this technology in an open and transparent manner. Further information on this topic can be found in Annex 3.

Annex 1: Mapping between the OSS charter and other frameworks

Principle	Fundamental Principles of Official Statistics	UN NQAF Quality Principles 2019	EU-Open source strategy, 2020
1. OSS by default	2: Professional Standards, Scientific Principles, and Professional Ethics; 3: Accountability and transparency; 10: International Cooperation	5: Assuring impartiality and objectivity, 6: Assuring transparency, 8: Assuring commitment to quality, 10: Assuring methodological soundness, 18: Assuring coherence and comparability	5.1 Think open, 5.2 Transform, 5.3 Share, 5.4 Contribute
2. Work in the open	3: Accountability and transparency; 8: National coordination	6: Assuring transparency, 11: Assuring cost-effectiveness (quality principle requirement 11.4)	5.2 Transform, 5.5 Secure, 5.6 Stay in control
3. Improve and give back	2: Professional Standards, Scientific Principles, and Professional Ethics; 8: National coordination; 10: International cooperation	3: Managing statistical standards, 8: Assuring commitment to Quality, 10: Assuring methodological soundness, 11: Assuring cost-effectiveness (11.6)	5.1 Think open, 5.2 Transform, 5.3 Share, 5.4 Contribute
4. Think generic statistical building blocks	2: Professional standards and ethics; 9: Use of international standards; 10: International cooperation	3: Managing statistical standards, 10: Assuring methodological soundness, 11: Assuring cost-effectiveness (11.6)	5.2 Transform, 5.5 Secure, 5.6 Stay in control

5. Test, package and document	3: Accountability and transparency; 10: International cooperation	6: Assuring transparency, 8: Assuring commitment to quality, 10: Assuring methodological soundness (10.5), 12: Assuring appropriate statistical procedures (12.1), 19: Managing metadata	5.3 Share, 5.5 Secure, 5.6 Stay in control
6. Choose permissive	10: International Cooperation	3: Managing statistical standards	5.1 Think open, 5.2 Transform, 5.3 Share, 5.4 Contribute
7. Promote	10: International cooperation	3: Managing statistical standards (3.1), 8: Assuring commitment to quality (8.2)	5.1 Think open, 5.3 Share

Annex 2: SWOT analysis on OS adoption in NSOs

This SWOT analysis is based on a preliminary SWOT exercise conducted by the project team as part of its sprint in September 2024 and subsequent in-depth evaluation of the identified strengths, weaknesses, opportunities, and threats of open source software (OSS). Within each category, we have extracted key themes, and outlined the various aspects of the use of open source software in a statistical organisation through those themes.

Note that while open source development is driven by a collaborative community, the focus of this analysis is on individual organisations as the decision of open source adoption is made at the level of the organisation.

Strengths and opportunities

In this section, we discuss the strengths of OSS - positive attributes and resources that provide a competitive advantage in its use - as well as the opportunities it presents for statistical organisations.

In analysing the strengths, we recognise that they can both be elements that are currently present, or elements that are yet to be realised, which may require the leveraging of various opportunities to achieve.

Strengths and opportunities fall into the following key themes:

- Freedom to shape organisational future and meet needs.
- Democratisation of development and agility.
- Transparency.
- Improvement of quality, interoperability, and standardisation.
- A sense of community and communal development (for a public good).
- Cost reduction.
- Alignment with job market trends.

There is also overlap and interaction between these themes (e.g., transparency can lead to improvements in quality), which we note below.

Finally, it is important to note that to realise the strengths and opportunities outlined, the talent capability required within an organisation is different than if off-the-shelf priority solutions are used, or solutions contracted out for development and used in-house. The fostering of in-house open source talent is vital to open source strategy and organisational resilience.

Democratisation of development

Because OSS does not require proprietary licenses or large upfront costs, any actor is able to begin a project or development with existing tools, software, and code without limits of access to restricted technology. This means that such development is available to any organisation (or indeed individual), large or small, even if their financial resources are limited.

Thus, OSS development is a more democratic venture by these virtues. In addition, this lends itself to agility of development, and scalability. Essentially, a project can start with one person, and remain small, or grow into a large community, depending on the utility and ease-of-use of the developed software. Conversely, with proprietary tools, this is essentially impossible due to restricted access (particularly to source code).

In addition, **the clearly established licensing models** can protect users and developers alike, thus reducing the liability risks to development, further reducing barriers, and allowing a broader community to contribute to software development without fear of legal issues.

We note however, OSS in and of itself does not remove all resourcing barriers to development, for example, time, capacity, and capability driven barriers will remain regardless of the openness of the software in question.

Freedom to shape organisational future and meet needs

With OSS, there are no restrictions driven by vendor policies, nor limitations imposed based on external decisions by the maintainers of the proprietary tools. For example, if a feature set is removed from a proprietary tool, one has no recourse, apart from negotiations with the vendor. Such limitations do not exist with OSS, as one is free to create a fork of a version with the desired properties.

Critically, OSS also avoids **capture by vendors** (who often do not have public good as their core mission) and **vendor lock-in**. This increases the flexibility and decision space available to an organisation to shape its own future. This also reduces the risk of unsupported software becoming a large technical debt (as vendors withdraw support, or feature sets), as with OSS a new maintainer can simply take over, or form a new clone or fork if desired.

In addition, because large commercial developers of proprietary software do not have a general interest in investment and development of solutions for niche markets (such as official statistics) NSOs using these types of software are forced to adopt their practices and compromise best practice to fit what is available, rather than having a tool that is fully fit-for-purpose, thus limiting the flexibility to operate in the most optimal manner to achieve the missions of the organisation. An internal capability in the use of OSS, and investment therein provides a mitigation against such.

Transparency

Trust in official statistics is a critical requirement for the successful use and uptake of the outputs and insights produced by NSOs as well as other organisations dealing with official statistics. Without trust, decisions based on these statistics, and the statistics themselves can be called into question, decreasing their value and utility.

Transparency in the sourcing, transforming, and analysing of data, through to the production of outputs (i.e., for all activities within the GSBPM) is a critical element of building trust.

While OSS alone is not sufficient to build such transparency, it forms a necessary foundation, because all elements of the process (in theory) are fully examinable for steps that are undertaken in any transformation, analysis, modelling, or dissemination conducted with OSS.

If an organisation uses OSS in their processes, and also adheres to the principles of OSS by openly publishing their codebase, anyone in the public can interrogate the code, suggest improvements, and report on any issues. This demonstrates that the processes are not hidden, and can be vetted. Further, this allows for alignment with open publishing, open access, and open science standards.

Improvement of quality, interoperability, and standardisation

The transparency achieved by the examinability of published code (see above) can lead to better quality solutions, both because a developer is more likely to take extra care in their work if it is public, and, the community can suggest improvements and raise issues as they are discovered.

Interoperability and standardisation, while important in and of themselves, can also be thought of as contributing to quality, since these can reduce duplication of solutions, increase efficiency in adoption and development for organisations, propagate best practices, avoid compatibility issues, and reduce the risk of errors or failures.

Developing to open standards organically drives standardisation across the solution space and tool kits. As a consequence of that adherence to open standards leads to better integrability of various solutions and other systems adhering to the same standards. This means that implementations of common methodologies, techniques, and solutions can be replicable, and easily adopted to a given organisation's needs, effortlessly achieving a baseline of quality. This kind of flexibility in terms of standardisation and quality improvement is not possible with proprietary software, as it can only be driven by the vendor.

Improved interoperability and standardisation can also enhance collaboration with external stakeholders, such as other government agencies and academic institutions, by enabling them to adopt the same open source software for similar purposes within their organisations. This shared usage fosters cooperation across diverse entities and facilitates joint projects.

A sense of community and communal development (for a public good)

A less tangible, yet important, strength of OSS development is that both users and developers can have a sense of community. Where dealings with vendors of proprietary software are often purely transactional, because OSS is a more collaborative effort, usually driven by people's needs, passions, and even voluntary efforts, this can develop a sense of belonging to something bigger, the open source community (for a particular interest).

Such communal development and collaboration, coupled with the transparency aspects above can also improve quality and innovation, since contributors outside of the project may bring new ideas and perspectives to the work.

The sense of making a greater contribution, and the acknowledgement received from your peers in the community can drive developer satisfaction, which has other benefits in terms of morale, wider contributions to the organisation, and talent retention.

Cost reduction

A major strength of OSS is the large realisable cost reductions in both development and operations. There are several direct drivers of this:

- No need for expensive per-user licenses, or other software usage fees,
- No risk of escalating licensing costs (often for no changes in the underlying software),
- No need to buy add-on proprietary features (either as they are released, or locked behind paywalls),
- No need to buy into an expensive wider ecosystem of vendor-locked software to be able to use specific tools.

In addition to this there are indirect drivers of cost savings, specifically:

- A multiplied return on investment (ROI) due to co-investment (where many organisations can work together on development, realising the benefits, without having to duplicate investment or fees),
- Usage of existing codebases and adapting to flexible requirements (see section on Freedom), leading to more optimal resource allocation and operations,
- Once developed, the software can be used by many, so for downstream users this is a major reduction in cost, and for upstream developers, the contributions back upstream can improve their product at no extra cost.

Alignment with job market trends

Finally, it should be noted that there is a growing shift toward open source culture and the adoption of open source programming languages across various fields such as research and the private sector. Many universities now offer curricula focused on open source languages such as R and Python. Statistical organisations that actively use open source software not only are better positioned to attract these candidates, presenting themselves as forward-thinking and aligned with current industry standards but also can avoid the need to train recruits in outdated skills. Additionally, as the job market for statisticians, data scientists, and software engineers increasingly comprises individuals with open source expertise, the talent pool for statistical organisations will continue to expand, providing more opportunities for recruitment.

As noted earlier, the various strengths of OSS overlap and are intertwined. It is these overlapping strengths that lead to a significant amount of the cost reductions. Because of that, in assessing the costs, particularly when the choice is between proprietary solutions and open source solutions, a wider perspective on costs, services, and implications must be taken. However, even a direct comparison is likely to yield lower costs with the use of OSS due to the direct drivers above.

Example: MySQL and MariaDB

MySQL is a relational database management system first released by the Swedish company MySQL AB in 1995. In a few years MySQL became the most popular open source database management system. It was standard (using the standard language SQL), portable (Windows, Linux, Unix, MacOS), fast, and easy to manage and integrate with other tools. In 2008, Sun Microsystems, the developer of the Java language and a strong supporter of open systems like Unix, acquired MySQL AB.

In 2009 Oracle corporation proposed a takeover of Sun Microsystems. In 2010, the European Commission approved Oracle's acquisition of Sun, on the condition that Oracle continue to invest in MySQL and keep it openly licensed.

The day Oracle announced the purchase of Sun, the main author of MySQL and founder of MySQL AB, Michael Widenius, forked MySQL, launching MariaDB; many MySQL developers followed him. MariaDB was created with the target to remain *free* and *open source* under the GNU GPL licence. Over many years MariaDB has grown as a product, maintaining compatibility with MySQL, while developing many new features and new engines, offering a fully managed cloud database service.

MariaDB became the new "M" in the LAMP stack, and today is used by millions of users. It is the default database in most Linux distributions, and is available in every cloud.

History shows that open-source licenses can ensure continuity of supply and service even in the presence of disruptive corporate events. MariaDB is a success story that has increased user confidence in the open-source world.

BOX A

Weakness and threats

This section discusses inherent **weakness** of open source software or limitation, lack of capability within a statistical organisation that makes it difficult to adopt open source software. It also includes **threats,** external factors and influences that could impose risks to the organisations when using open source.

Maintenance and sustainability

One of the major weaknesses associated with OSS is around long-term maintenance and sustainability. Software used in statistical organisations, especially those in production, require a certain level of sustainability as they can have a significant impact on comparability of the statistics produced, which is a crucial quality of official statistics.

When OSS depends heavily on individual contributors or single organisations, it can lead to a "single point of failure" (unless community support is created). If these key individuals become unavailable (e.g., retirement, transfer) or the organisation who maintained the OSS shifts their

priorities and discontinues the support, it becomes difficult to sustain the development and maintenance of the software. Open source licences by nature do not guarantee assistance (which are typically established via formal service level agreements or terms and conditions for proprietary software from private companies) and this adds uncertainty, which in turn creates fear in the users and organisations who try to adopt OSS.

Governance

Complexities around governance present a significant challenge in the adoption of open source software. The lack of a clear governance framework creates a "governance maze," where roles, responsibilities, and processes such as deciding who does what, when, and where are poorly defined. As the number of project clones and spinoffs grows, determining which version to adopt, support and keep becomes complicated (“clones hell”).

Governance becomes even more complex at the international level as it is difficult to establish structure and enforce policies. For larger-scale projects, relying solely on voluntary contributions may be insufficient. These projects would require systematic and sustained support from organisations to ensure proper governance.

Lack of legal expertise

As demonstrated in Chapter 3: Case studies, open source software adoption requires legal considerations, particularly regarding licensing when developing and releasing the software or repurposing existing software. Legal expertise in statistical organisations has been typically focused on areas such as statistical law and data access, and expertise in open source license and managing issues arising from license is limited.

Learning curve and lack of culture

From a user-perspective, open source software may present a steeper learning curve compared to proprietary software that provides user-friendly interfaces and customer service as part of the package. Also, open source tools are often based on open source programming languages such as R and Python, which represent new skill sets for many staff members. While many new recruits are already familiar with these languages, much of the workforce in statistical organisations were trained and worked with traditional programming languages.

This also incurs massive cultural change. For example, developers often have a habit of working independently, with limited experience on code-sharing. This lack of a cooperative culture can create barriers to open source software adoption.

Integration

The incorporation of OSS into existing systems and workflows can be a serious challenge. Some proprietary software allows for the incorporation of scripts written in languages like R and Python,

while others are more restrictive, and it is possible that multiple separate workflows must be run in order to produce desired results.

Similarly, tools that are developed with OSS that rely on proprietary software for part of the production process, can run into issues where the proprietary software is updated, and the pipeline built with OSS is resultantly broken. One example is the R package *Pagedown* which is used to create reports written in R and structured in page format using JS and CSS, which relies on Chrome for the rendering of documents into PDF formatting. Updates of Chrome have in the past resulted in the rendering process of the package to break. Thus, the desired flexibility and power of OSS is thus hamstrung by the reliance of proprietary software.

Further discussion on integration and related interoperability issues can be found in SIS-CC's excellent and detailed article, "Enhancing SDMX tools interoperability for improved organisational efficiency".[38]

Hidden and double cost

As highlighted by all factors mentioned above, open source software is far from "free". Organisations must invest significantly in capacity building, maintenance and governance. Additionally, during the transition period, they may need to maintain traditional software in parallel which results in double costs for license, support and infrastructure.

There is also a cost surrounding uncertainty. From a user perspective, the lack of guaranteed customer service can create operational risks, while reliance on community-driven support from an open source developer perspective may introduce unpredictability. These can pose a significant financial and operational burden for organisations adopting open source software.

Potential IP issues and security breaches from outside

The fundamental premise of OSS that source code is open and anyone can freely use it can present potential threats related to legal, intellectual property (IP) issues, and security breaches. One major concern is the possibility of outside entities taking over the software, modifying it, and exploiting it for commercial gains. Improvement is always welcome, but not all actors may follow the terms in license. This could result in IP complications, which can be particularly daunting for statistical organisations, where there is lack of legal expertise in OSS licensing and IP issues.

Also, the open nature of OSS exposes it to risks of malicious actors exploiting it for attacks, compromising the software's integrity as well as corrupted outputs, loss of sensitive data, or diminished trust in the system. With the use of open data expanding, open source codes may lead to further increase of privacy risk, e.g., through membership inference attack on ML models. This threat is particularly critical for statistical organisations, where reliability and confidentiality are paramount.

[38] https://siscc.org/enhancing-sdmx-tools-interoperability-for-improved-organisational-efficiency/

Annex 3: Open source and AI

The intersection of open source and AI promotes a dynamic ecosystem where transparency, collaboration and innovation drive the development of accessible, trustworthy and cutting-edge AI technologies.

As data, models and algorithms fuel AI development, making sure that they are open is critical for NSOs not least to secure reproducibility, replicability and traceability of outputs. These are preconditions of any credible, accurate and trustworthy official statistics.

The OSI document

In the fall 2024, the Open Source Initiative (OSI) released the final version of "The Open Source AI definition" which outlines the principles and requirements for defining Open Source Artificial Intelligence (AI).

OSI defines as "Open Source AI" a system that must allow users to:

- **Use** the system for any purpose and without having to ask for permission.
- **Study** how the system works and inspect its components.
- **Modify** the system for any purpose, including to change its output.
- **Share** the system for others to use with or without modifications, for any purpose.

To modify a machine-learning system, the following elements must be available:

- **Data Information**: Detailed information about the data used to train the system, including its provenance, scope, characteristics, and processing methods.
- **Code**: Complete source code used to train and run the system, including data processing, training, validation, testing, and inference code.
- **Parameters**: Model parameters such as weights or configuration settings, including checkpoints and optimiser states.

Concerning the licensing requirements, applied licenses should adhere to OSI standards, ensuring the system and its components remain accessible and modifiable. In some cases (viral licenses) conditions may require modified versions to be released under the same terms as the original, preserving openness.

As machine learning systems are composed of AI **models (**including the architecture, parameters, and inference code) and AI **weights (**the learned parameters that produce outputs from inputs), both models and weights must provide data information and code for reproducibility. More fundamentally, disclosing the underlying data, algorithms and models used by AI systems is critical to ensure the traceability of inputs to outputs and outputs to inputs.

Open source AI for NSOs

National statistical offices (NSOs) can significantly benefit from adopting Open Source AI, not least because of its alignment with the principles of transparency, efficiency, and public trust, which are central to the mission of NSOs. There are a number of key reasons why Open Source AI can be fruitful for NSOs:

- **Transparency and accountability**:
  - Auditable systems: Open Source AI systems can be inspected and audited, ensuring that the methods used for data analysis and reporting are transparent and accountable.
  - Reproducibility: Open source AI allows external parties to reproduce results, validating the integrity and reliability of the statistical outputs produced by NSOs.
- **Customisation and flexibility**:
  - Tailored solutions: NSOs can customise Open Source AI tools to meet their specific needs and requirements, such as national data processing requirements, regional language models, or specific statistical methodologies. At the same time, open models can easily be "portable", integrating easily in similar environments.
  - Adaptability: Open Source AI allows NSOs to adapt and improve their AI systems as statistical methods and data sources evolve, ensuring long-term relevance and effectiveness.
- **Collaboration, sharing and accessibility**:
  - Community support: NSOs can benefit from the support of a global community of developers and experts (NSOs, international organisations, academic institutions, …), including to freely share code, models and training data. Community can provide assistance, updates and improvements, while also reducing duplication of efforts.
  - Sharing and capacity building: collaboration with other statistical organisations and the Open Source community can lead to the sharing of best practices and innovative solutions, increasing compatibility and standardisation. It can also foster the sharing of training and knowledge.
  - Wider accessibility: Open Source AI tools enable smaller NSOs or those with limited budgets to access advanced statistical and machine-learning capabilities
- **Innovation**:
  - Access to cutting-edge technology: Open Source AI provides NSOs with free access to state-of-the-art tools and models, enabling them to adopt the latest advancements in data analysis, forecasting, and machine learning.
  - Scalability: NSOs can scale their AI solutions easily by leveraging the collective efforts of the community, ensuring that their technologies can grow and adapt as their data collection and analysis needs expand.
- **Ethical considerations**:
  - Ethical AI: by disclosing training data, algorithms and models, Open Source AI allows NSOs to ensure that their AI systems are developed and used ethically, addressing concerns about bias, privacy, and FAIRness.
  - Compliance: The transparency of Open Source AI systems enables NSOs to comply with ethical standards and regulatory requirements, ensuring that their statistical work is conducted responsibly.

- **Strengthening security**:
    - Control over data: Open Source AI systems allow NSOs to retain full control over their data, minimizing risks associated with external vendors or proprietary black-box solutions.
    - Community-verified security: The transparency of Open Source tools means that vulnerabilities can be quickly identified and patched by the community, enhancing overall security.
    - Quality assurance: Open Source AI tools can be analysed and validated by the community, ensuring that they meet high standards of data integrity and security.

Open Source AI offers NSOs a path to greater transparency, collaboration, and innovation while fostering trust and accountability. By embracing these tools, NSOs can modernise their operations, improve statistical outputs, and better serve the public and policymakers. This approach aligns with global trends in open data and open government initiatives, ensuring that NSOs remain leaders in statistical innovation and integrity.

OSI released in December 2024 a document with the aim of applying the principles of the Open Source Definition in the domain of AI, trying to merge in Open Source AI different "openness" principles interacting with each other: Open Data with Open Source with Open Science and Open Knowledge.

Statistical Open Source Software

Charter and Report

Open source solutions enable national statistical offices (NSOs) to develop, validate, and share statistical methods while ensuring reproducibility of official statistics. The transparent nature of open source software allows for peer review of statistical procedures, thereby strengthening the credibility of official statistics.

This report and the charter of principles herein are intended to guide NSOs in adopting and developing open source tools so they may build flexible, cost-effective statistical infrastructures that can adapt to new and emerging data sources and methodological innovations, while building trust through transparency in statistical production.

modernstats
by HLG - MOS