

# HLG-MOS Statistical Open Source Software

## Guiding Principles for Official Statistics

### Table of contents

Opening Statement . . . . .	1
Principles . . . . .	2
1. OSS by default . . . . .	2
2. Work in the open . . . . .	3
3. Improve and give back . . . . .	3
4. Think generic statistical building blocks . . . . .	4
5. Test, package and document . . . . .	5
6. Choose permissive . . . . .	5
7. Promote . . . . .	6

**i** Note

The Guiding Principles (previously referred to as “The Charter”) were [endorsed](#) by the [Conference of European Statisticians \(CES\)](#) at the Seventy-third plenary session in Geneva, Switzerland, 16–18 June 2025.

### 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.

At the same time, OSS adoption and development should take into account operational needs, institutional priorities, security considerations, existing IT infrastructure and other relevant capabilities.

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.<sup>1</sup>

## Principles

### 1. OSS by default

#### Statement

In the production of official statistics, we prefer the use of open source software solutions over closed software solutions, taking into account business and technical requirements as well as legal and confidentiality considerations. 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 strive 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.

---

<sup>1</sup>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 (<https://os4os.pages.code.europa.eu/pbbp>) and adjusted for the global context.

## **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 it is possible to derogate from the default OSS option. The same applies to sharing: sharing as open source is the default, but it is possible to derogate 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.

## **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 with deciding on the best moment to do this. 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 projects' development phase over time to help manage expectations, especially for users engaging with early-stage tools. Clear communication of a project's maturity is essential for building confidence and trust in open-source software.

### **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. Sharing open-source code should be guided by project-specific risk assessments, particularly when sensitive data is involved.

## **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 exactly cover the functionality needed in official statistics. The quickest way to cope with 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 that. 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 re-used instead of creating new solutions. Even if a solution does not exactly fit the required functionality, it is examined how it could be improved while keeping the intended functionality in mind or even extending it. This also applies for partial solutions such as code snippets and (machine learning) models that could be valuable for others. The changes or enhancements are tested, documented, and returned to the respective community to decide on possible integration into their solution. The scale and timing of such contributions may need to take into account project-specific needs, operational deadlines, and available resources.

## **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 for effective re-use 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 away that it can be re-used 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 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.

## **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 way 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 re-use. 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 re-use. 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, automate as much as possible, and smaller projects can 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, packaging and documentation policies in their domain of expertise.

## **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. Re-use 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 re-use by others it is necessary to choose an OS licence that maximally allows re-use, and minimises conflicts with other licences. This is known as “permissive”. When choosing the appropriate OS licence we strive for maximum re-use.

## **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.

## **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**

Re-use of generic software will not happen if no one knows what can be re-used. On the other hand it is difficult to know beforehand what the value of our software is for others. The only way out is to communicate, even if we have no clue whether it can be used in a wider context. We promote our software in an honest, concise way, mentioning its core functionality. We let the public know our plans for new developments and improvements and be 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, possible OS building blocks and its application in the statistical production should be preferred to centrally maintained repositories. A centrally maintained repository of software tools can quickly become outdated and collecting information from the community could be a big effort. Therefore, such a repository should be maintained by the whole community. 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 domain of expertise.