"Show me your code, and then I will trust your figures": Towards software-agnostic open algorithms in statistical production[1]

Jacopo Grazzini[a], Pierre Lamarche[b], Julien Gaffuri[a], and Jean-Marc Museux[a]

[a]DG ESTAT, European Commission (Eurostat) – *{first.last}@ec.europa.eu*

[b]National Institute of Statistics and Economic Studies (INSEE) – *pierre.lamarche@insee.fr*

Abstract

This contribution aims at further promoting the development and deployment of open, reproducible, reusable, verifiable, and collaborative computational resources in statistical offices regardless of the platform/software in use. Motivated by the consensus that data-driven evidence-based policymaking should be transparent, we argue that such approach is not only necessary for the practical implementation of statistical production systems, but also essential to reinforce the quality and trust of official statistics, especially in the context of a "post-truth" society. We discuss some practical requirements to gear the continuous and flexible development and deployment of software components in production environments. Together with the adoption of some best practices derived from the open source community and the integration of new technological solutions, we propose to unleash the social power of open algorithms so as to create new participatory models of interaction between produsers that can contribute to a more holistic and extensive approach to production systems. Overall, a greater transparency in designing production processes is expected to result in a better grip on the quality of the statistical processes involved in data-driven policy-making. We illustrate this flexible and agile approach with existing open, stand-alone software or source code used in statistical production environments at Eurostat.

Keywords: statistical production, open algorithms, smart and explainable statistics, openness, transparency and trustability, accountability and verifiability, reusability and reproducibility, control and maintenance, sharing and collaboration, open source and legacy software, software-agnostic development, microservice.

1. Introduction

With the advent of *Open Source Software* (OSS) in the scientific statistical community – not to mention other disrupting technologies and software emerging from data science and similar active communities – many statistical organisations, in particular National Statistical Institutes (NSIs), are nowadays considering to introduce – or migrate to – such solutions (especially, R and Python) in production environments so as to benefit from the many statistical libraries, advanced algorithms and innovative developments made available freely and openly. The debate about

[1] Supporting material developed by the authors is available at this page: https://github.com/eurostat.

actual and alleged risks related to the use of OSS (*e.g.*, in terms of warranties and liability, security and vulnerability, *etc…*) compared to those of proprietary/*commercial-off-the-shelf* (COTS) software still holds. However, it seems that the – cultural, rather than structural – OSS phobia in public administrations, linked to the potential exposure of flaws, or the supposed loss of relevance and value, has been overcome[2]. In parallel, the many calls for transparent and defensible evidence-based and data-informed policymaking further encourage the use of OSS in statistical organisations and NSIs, beyond all practical aspects and technological considerations (Grazzini & Pantisano, 2015; Höchtl *et al.*, 2016).

Several challenging issues accompany the practical implementation of statistical production systems, *e.g.* in modelling and programming, in handling and managing data, in running and monitoring of the execution of processes, to mention a few (Salgado, 2016). OSS may not only become a possible solution in production (Templ & Todorov, 2016), it could also reinforce the quality and trust of official statistics, especially in a context of *"post-truth"* society (Baldacci & Pelagalli, 2017; Association for Computing Machinery, 2017). Beyond OSS alone, we advocate for the adoption of an *Open Algorithm* (OA) strategy in NSIs. We discuss some practical requirements to gear the continuous and flexible development and deployment of statistical components and services in production environments, may they derive from OSS or COTS software. The generalised use of OAs supports the overall quality and trust, as well as the complete transparency, of the statistical processes involved in policy-driven data-informed evidence-based decision-making. Through the adoption of some best practices derived from the OSS community and the integration of new technological solutions, it can also help create new participatory models of interaction between *"produsers"* (statisticians, scientists and citizens) to support new modes of production of official statistics in a constantly evolving data ecosystem.

2. Unleashing the *social power* of open algorithms

Openness is obviously one crucial and desired aspect of statistical software. The term *"openness"* shall however be interpreted according to the broad conceptual

[2] See for instance how France public administration made available the source code of tax and benefits calculators (material on Join-up platform) so as to increase transparency.

sense of OA (and *open source code*) rather than the sole technical meaning of *open source software*. Together with some best practices derived from the OSS community (Lahti *et al.*, 2017; Perez-Riverol *et al.*, 2016) – taking advantage of version control, automated unit tests, generic documentation, continuous integration, and collaborative development – sound principles[3] gear the practical development of OAs and ensure the transparency of the statistical processes used in production:

- Reproducibility contributes to consistency checks and quality assurance of the output statistical products by ensuring that the statistical processes and computational workflows can be replicated (Stodden, 2014).
- Reusability of software components ensures they become applicable in contexts/domains other than the ones for which they have been developed (Ince *et al.*, 2012). Instead, a critical barrier is when the source code used for a given application is closed (or, often, written in a way that defies its reusability).
- Verifiability and auditability shall be supported: given any input configuration, it is possible not only to regenerate the output of a statistical process, but also to fully test and validate the underlying model/methodology (Leek *et al.*, 2017).
- Sharing helps in-house adoption of efficient ways of working in production, avoiding developments from scratch every time. It can also foster collaboration beyond the statistical organisations so that external communities contribute to an increased robustness and additional features (Perez-Riverol *et al.*, 2016).

Overall, a greater transparency in designing production processes is expected to result in a better grip on the quality of the output statistical products. In an open environment, there is actually generally a set of positive incentives for useful modifications to be shared back to the benefit of the entire community. This can contribute to a more holistic and extensive approach to production systems in statistical offices with the development and deployment of high-quality statistical processes[4]. Such a collaborative approach also reduces both the risk of having no-one available to maintain or update the system and the cost of the necessary testing and audit procedures needed for high-reliability software.

[3] One shall bear in mind that *"the modernisation and industrialisation of official statistical production needs a unified combination of statistics and computer science in its very principles"* (Salgado, 2016).

[4] For instance, the ICW source code (available here) enables users to fully reproduce the experimental statistics on European households' income, consumption and wealth disseminated by Eurostat.

3. Becoming software *agnostic*

OSS – and more generally open source technologies – should be preferred over traditional proprietary commercial (COTS) technologies – closed-source, black-box – since they guarantee openness and adaptability, and presents very tangible benefits (Grazzini & Pantisano, 2015; Templ & Todorov, 2016). However, as previously mentioned, the actual implementation in-place in many statistical offices is often tied to legacy COTS software solutions. Since they are believed to guarantee the reliability of statistical processes that OSS may lack – not only from a technological standpoint, but also from legal and structural aspects – COTS software (*e.g.,* SAS software) are still in wide and prominent use in statistical organisations. In adopting any desired OSS solution, say, R for instance, there is supposedly a trade-off between the risk linked to business continuity and reliability against the need for efficiency, innovation and cost-effective solutions (Grazzini & Lamarche, 2017). Hence, in view of future migration, a strong control of the practical and effective implementation of statistical processes, methods and techniques is required so as to favour the continuity of operations, but also support their future evolution.

More generally, a key issue to consider is the comprehensiveness of the statistical information that should be achieved through clear and efficient (re)implementation – may that derive from OSS or COTS software. Statistical software packages often implement a broad range of techniques but do so in an ad-hoc manner, leaving users at a disadvantage since they may not understand all the implications of a given process, or how to test the validity of results produced by the software (Grazzini & Lamarche, 2017). The approach based on OAs should actually be adopted when developing free as well as proprietary software[5]. Ideally, it should be possible to implement a given statistical operation in whatever software/platform (*e.g.,* programming language) that fits best. The focus should not be on specific software solutions, instead they should complement each other, so as to overcome the drawbacks and risks of adopting a *one-size-fits-all* approach for software. Finally, the *agnostic* approach will leverage the opportunities that new technologies (in particular,

[5] The source code for proprietary software can also be made open with appropriate licensing. An example at Eurostat is for instance PING made openly available (here), though being mainly developed in proprietary SAS. Also note that the Journal of Open Source Software does commonly accept submissions that rely upon proprietary languages and development environments.

OSS) offer for working more effectively and it supports software (and hardware) independence, thereby alleviating the potential burden of legacy. One should only ensure that the statistical processes implemented using different software/technologies are consistent and robust[6].

4. Adopting a *plug and play* design

As evidenced by Salgado (2016), the development and deployment of a production system is not a trivial task. Grazzini & Lamarche (2017) underlined the need to reach certain standards in terms of interoperability, accessibility, effectiveness and robustness, scalability and timeliness, flexibility and extensibility. Considering that production systems will most likely change during their lifetime, long-term maintenance and extension may reveal burdensome due to legacy issues, *e.g.*, software components with customised design have changed or become obsolete. In this context, a *"plug and play"* design, based on modular and customisable software components which are easy to assemble into a reliable processing system, needs to be adopted. First, this design makes possible the integration of already existing processes, operations and components, as well as new ones. Second, the modular design further supports the *agnostic* approach by releasing the constraint on the choice of the software/platform used for the implementation.

Obviously, such design has a cost since it introduces further constraint on the deployment within the enterprise architecture: it is essential to guarantee that different statistical processes – *e.g.,* through service-oriented architecture – fit seamlessly together and can communicate between each other inside the *statistical production ecosystem* (Grazzini & Lamarche, 2017). When designing a modular production process, special attention needs to be paid to the specification of the statistical methods and the deployment of the software modules in which the statistical methods are encapsulated. There is actually a trade-off between the scope of application (flexibility), the ease of application (reusability), the efficiency, and the simplicity of the modular components. Still, the modular design is a way to share

[6] The project quantile (available here) proposes, for instance, a canonical implementation of a simple algorithm (quantile estimates) which is consistent throughout various programming languages.

validated statistical methods and OAs, while the benefit of these components will increase with each new statistical process that will be designed to (re)use them.

5. Discussion

OA, together with *open data*, can increase efficiency of statistical processes and timeliness of products and services[7]. They can be beneficial to gear the continuous and flexible development and deployment of available – open and free, but proprietary as well – software components in production. Besides the call for OSS solutions, one should ensure in practice that all these components are fully documented[8], tested and, ultimately, shared. In principle, OAs offer additional control since they guarantee not only the validity of the statistical products, but also the methodological choices made for the implementation (Ince *et al.*, 2012).

Still, the requirement of openness and transparency is not automatically equivalent to publicising or accessing the code of the algorithm. In particular, algorithmic decision-making may become a key element: algorithms can be applied to a wide range of processes and can underpin fully automated or semi-automated decision-making. The trend towards algorithmic decision-making – and automated data processing techniques – raises methodological and empirical difficulties which can lead to important biases that may be hard to identify, not to mention ethical issues (Guidotti *et al.*, 2018). For instance, decision-making criteria can be programmed-in through relatively simple instructions, or can be learned by the algorithm through machine learning approaches and analytics techniques (Höchtl *et al.*, 2016). Yet, a general implication of the basic functioning of machine learning – *e.g.,* through the design and training of a model on massive datasets – is that opening the algorithm may not be enough to understand and explain the – often opaque and rather unpredictable – decision[9]. More meaningful solutions will certainly be needed to ensure accountability

[7] Though one may argue they might do little at the end: *"it is not the data or the technology that is at the heart of the problem, but their application in a bureaucratic environment"* (Höchtl *et al.*, 2016).

[8] The code udoxy (available here) provides generic guidelines and material to automatically generate documentation from source code implemented in different programming languages.

[9] There is a common misconception that algorithms automatically result in unbiased decisions since they are well-defined procedures for processing data automatically; instead there remain many problems with the data they process and the type/nature of the algorithms themselves.

and fairness and a clearer understanding of effective openness and transparency (Association for Computing Machinery, 2017; Guidotti *et al.*, 2018).

Prior to designing truly *"explainable algorithms"* – thus, "*explainable statistics"* – and beyond just opening algorithms, it is already possible to provide the public with further insights into the workings of algorithmic decision-making systems by opening and sharing fully reproducible computational workflows (Beaulieu-Jones & Greene, 2017). Actually, today's technological solutions, *e.g.* flexible Application Programming Interface, lightweight virtualised container platforms and versatile interactive notebooks, support an approach where algorithms are delivered, together with data, as portable, scalable, standardised and encapsulated (micro)services[10]. Reproducible computational workflows will not only enable *produsers* – *i.e.,* advanced users like statisticians, scientists and citizens – to fully reproduce experiments (Stodden, 2014), but also allow them to *"judge for themselves if they agree with the analytical choices, possibly identify innocent mistakes and try other routes"* (Leek *et al.*, 2017).

5. Conclusion

Although the requirements for official statistics in terms of transparency, privacy and ethics, quality and robustness, and timeliness seem not to change, the rise of new technologies and trends in sharing, handling, processing and analysing data calls for an upgrade of practices in statistical offices. It is believed that software solutions based upon OAs provide with an open, flexible and agile approach to immediate needs and legacy issues, as well as long-term problems and potential future software requirements for statistical production. Altogether, the benefits of OA in supporting the overall quality as well as the complete transparency of the statistical processes involved in policy-driven data-informed evidence-based decision-making are recognised.

[10] For instance: docker software enables users to develop, deploy and run distributed applications regardless of the infrastructure; Jupyter notebooks are web-based interactive computing platforms. When "combined" together, the Jupyter Notebook Data Science Stack provides a portable environment with interactive computing tools and (agnostically) supports different programming languages. This approach is adopted for the implementation of the happyGISCO (available here) computing interface on top of Eurostat GISCO web-services.

References

Association for Computing Machinery (2017), Statement on algorithmic transparency and accountability, US Public Policy Council.

Baldacci, E. & Pelagalli, F. (2017), Communication of statistics in post-truth society: the good, the bad and the ugly, Publications Office of the European Union, doi:10.2785/553958.

Beaulieu-Jones, B.K. & Greene, C.S. (2017), Reproducibility of computational workflows is automated using continuous analysis, Nature Biotechnology, 35: 342–346, doi:10.1038/nbt.3780.

Guidotti, R. *et al.* (2018), A survey of methods for explaining black box models, arXiv: 1802.01933.

Grazzini, J. & Lamarche, P. (2017), Production of social statistics...goes social!, in Proc. New Techniques and Technology in Statistics.

Grazzini, J. & Pantisano, F. (2015), Guidelines for scientific evidence provision for policy support based on Big Data and open technologies, Publications Office of the European Union, doi:10.2788/329540.

Höchtl, J. *et al.* (2016), Big data in the policy cycle: Policy decision making in the digital era, Journal of Organizational Computing and Electronic Commerce, 26(1-2):147-169, doi:10.1080/10919392.2015.1125187.

Ince, D.C. *et al.* (2012), The case for open computer programs, Nature, 482:485-488, doi:10.1038/nature10836.

Lahti, L. *et al.* (2017), Retrieval and analysis of Eurostat open data with the eurostat package, The R Journal, 9(1):385-392.

Leek, J. *et al.* (2017): Five ways to fix statistics, Nature 551:557-559, doi:10.1038/d41586-017-07522-z.

Perez-Riverol, Y. *et al.* (2016), Ten simple rules for taking advantage of *git* and *github*, PLOS Computational Biology, 12(7):e1004947, doi:10.1371/journal.pcbi.1004947.

Salgado, D. (2016), A modern vision of official statistical production, Instituto Nacional de Estadistica.

Stodden, V. (2014), The reproducible research movement in statistics, Statistical Journal of the IAOS, doi:10.3233/SJI-140818.

Templ, M. & Todorov, V. (2016), The software environment R for official statistics and survey methodology, Austrian Journal of Statistics, 45:97–124, doi:10.17713/ajs.v45i1.100.