Transitioning from SAS to R: A Case Study from Customised Statistics

Emil Thranholm, *Statistics Denmark,* eth@dst.dk

Abstract

This paper presents a case study of the transition of Statistics Denmark's Data Consultancy team from SAS to R as its primary programming language. We examine the drivers behind this transition, the rationale for selecting R over Python, and lessons learned that may benefit other national statistical institutes (NSIs) facing similar digital transitions.

SAS has long dominated statistical programming in NSIs, but open-source alternatives such as R and Python have become increasingly viable. The Data Consultancy team produces customised statistics for external customers (e.g. ministries, municipalities and private companies) by retrieving and combining data from existing datasets. The team has recurring tasks with small-scale production systems, which offer advantages for digital transformation compared to larger, more complex statistical production systems with intertwined processes. We believe that experiences from this small-scale production can be generalised to the digital transition of larger systems.

Several factors motivated our transition, including maintaining relevance in the labour market, as recent university graduates typically possess R or Python skills rather than SAS skills. In addition, R skills enable greater international cooperation, as R skills are in higher demand within Statistics Denmark's strategic sector cooperation around the world. Furthermore, the transition creates better opportunities for automating processes in the future and adoption of emerging technologies (e.g. AI). While long-term efficiency and economic gains are expected, it is important to acknowledge that productivity declines during the transition. We sought to minimise this productivity loss by building a foundation prior to implementation, and the paper elaborates on the specific measures taken. We found R to be better suited than Python for our core tasks of data retrieval and statistical computation, a choice the paper examines in detail. Based on our experience, this paper offers practical recommendations for other NSIs undergoing similar transitions.

Keywords: Digital transition, R programming, Open-source software, SAS migration

1. Introduction

Many national statistical institutes (NSIs) are under pressure to reduce reliance on costly proprietary software and modernise their statistical production environments. At Statistics Denmark, SAS has been the dominant programming tool for decades, but the rise of open-source languages, in particular R and Python, has made the case for transition increasingly compelling. Open-source tools bring lower costs, stronger alignment with university training, and a more active international community of practice.

This paper is a case study of the Data Consultancy team at Statistics Denmark, which completed a transition from SAS to R in January 2026. The team's small-scale, well-defined production environment made it a suitable context for piloting this kind of change. This paper describes the key drivers for the transition, the rationale for choosing R over Python, the practical measures taken to manage implementation, and the lessons learned. The aim is to provide a reference for other NSIs considering a similar path.

2. Background

2.1 The Data Consultancy Team

The Data Consultancy team produces tailor-made statistics for paying customers, primarily ministries, government agencies, municipalities, and private-sector organisations such as NGOs, interest groups, and consultancy firms. Customers submit a data request and, where Statistics Denmark holds the relevant register data, receive a quoted offer. Upon acceptance, the team extracts and links data from the registers, applies any required transformations and statistical computations, performs statistical disclosure control in line with Statistics Denmark's Data Confidentiality Policy (Statistics Denmark, 2024), and delivers results, typically as an Excel file.

The team comprises around 14 members and handles approximately 250–300 of the aforementioned requests per year. Request/task sizes range from a few hours to over 200 hours, with the vast majority falling between 15 and 50 hours. Some tasks are one-off requests; others recur annually or quarterly. This recurring, modular structure, with relatively short and self-contained production pipelines, made it possible to make an immediate digital transition.

2.2 SAS at Statistics Denmark and the Data Consultancy Team

SAS has been the primary statistical programming tool at Statistics Denmark since the Data Consultancy team was founded 30 years ago. It remains widely used across the organisation, though it has been classified internally as legacy software. The team's codebase includes recurring tasks written in SAS that have grown complex and difficult to maintain over time, with bugs becoming progressively harder to isolate. This technical debt, combined with the broader institutional direction, created a clear motivation to find a modern alternative.

3. Drivers for Transition

3.1 Labour Market Relevance

SAS is no longer widely taught at Danish universities. The Data Consultancy team recruits primarily from the social sciences, where R (and to some extent Python) has become the standard tool for statistical programming. As a result, new graduates arrive with R (or Python) skills, not SAS. Requiring incoming staff to learn SAS represents an unnecessary onboarding burden and may deter qualified candidates from applying altogether. The transition to R aligns the team's tooling with the skills available in the labour market, widening the pool of viable candidates and reducing time-to-productivity for new hires.

3.2 International Cooperation

Open-source languages substantially lower the barriers to international collaboration. Packages and functions can be freely shared and reused across organisations, something the proprietary SAS ecosystem does not easily support. The Data Consultancy team directly benefits from this: statistical disclosure control in our workflows relies on the *GaussSuppression* R package (Langsrud et al., 2025), developed by colleagues at Statistics Norway. The broader community has produced a growing catalogue of R packages for official statistics, documented in the awesome official statistics software list (SNStatComp, 2026).

3.3 Future Automation and Emerging Technologies

Open-source languages integrate more readily with modern development toolchains, version control systems, CI/CD pipelines, and AI tooling. While SAS does offer some

AI-adjacent products, the team currently uses base SAS, meaning any such extension would itself require a separate transition. R's open ecosystem allows the team to adopt emerging tools incrementally, without waiting for a commercial vendor to bundle and release equivalent functionality.

4. Why R Over Python?

Both R and Python are credible alternatives to SAS for the team's core tasks, and most things achievable in one can be done in the other. The choice of R was based on a combination of fit-for-purpose assessment and practical considerations.

R was designed by statisticians for statistical work. This heritage is reflected throughout the language and its ecosystem: statistical methods are first-class citizens, data structures are built around vectors and data frames, and standard output from modelling functions is formatted for direct interpretation. Python is a general-purpose language that has developed strong data science libraries over time, but statistical reasoning is not embedded in its design in the same way. For a team whose core activity is statistical computation on register data, this distinction matters in day-to-day practice.

A more concrete advantage is the *tidyverse* package collection, which provides a consistent, expressive grammar for data manipulation (Wickham et al., 2019). Chaining operations with the pipe operator produces code that reads closely to the analytical steps it performs, which makes it easier to write, review, and maintain. Python's *pandas* or *polars* libraries cover similar ground and are powerful in their own right, but their syntax is often more verbose for the kind of iterative, exploratory data work that the team performs: filtering, joining, reshaping, and summarising register data.

The interactive, data-first workflow in commonly used R integrated development environments (IDEs) also suits the team's working style. After a transformation step, particularly joins across registers, the result is often visually inspected: checking for unexpected missing values, duplicate rows, or shifts in the row count. This kind of step-by-step validation is possible in Python too, but it requires more deliberate setup and is less naturally embedded in the default workflow and commonly used IDEs.

For the specific domain of official statistics, R also has a stronger tooling footprint. The awesome official statistics software list (SNStatComp, 2026) contains a substantial majority of R packages. Equivalent Python tooling in this space is more limited albeit growing.

A further practical factor is that several team members already had strong R skills prior to the transition. This created internal capacity to support colleagues during upskilling, an advantage that would not have existed had Python been chosen. That said, R and Python are not mutually exclusive. The team's default is R, but Python remains available for tasks where it offers a clear advantage. The decision may also be revisited as the landscape evolves: if Python's share of official-statistics tooling grows and university graduates arrive with predominantly Python backgrounds, a subsequent transition may follow in time.

5. Implementation

5.1 Preparation Phase

The transition had been discussed informally for several years before becoming a formal project, which meant that some groundwork was already in place. The core of the preparation was mapping the team's key SAS workflows and building equivalent R infrastructure. Widely used internal SAS macros were identified, and their functionality was either replicated in an internal R package or substituted with an existing R package offering an equivalent workflow. Furthermore, we focused on sticking to commonly known and well-maintained packages to avoid niche packages that may have bugs in their code, which can produce incorrect results. This is, of course, a risk for all packages, and generally a greater risk with open-source software compared to commercially licensed software. By using well-established and well-documented packages, we have not found this to be an issue (yet).

During the preparation stage, team members with existing R skills piloted the new setup on tasks over the course of a year. This exposed practical issues, missing functionality, edge cases, and integration gaps that had to be resolved before they affected the broader team. The preparation phase also included guidance on coding and a shared set of recommended packages to avoid fragmentation in how the team uses R.

5.2 Production Phase and Managing the Productivity Dip

From 1 January 2026, all new tasks are coded in R. This was announced a few months in advance, once the key workflows were in place. As expected, productivity declined in the initial period: team members were learning a new language while continuing to deliver for the customers. This dip is a normal and foreseeable consequence of any such transition, and it is important to acknowledge this.

Existing recurring SAS tasks were not immediately migrated. Well-functioning, well-maintained SAS code continues to run. Where existing SAS code had become difficult to maintain, characterised by growing complexity and recurring bugs, it was prioritised for migration to R.

The productivity dip also hits team members asynchronously. A team member who has worked with SAS for a long time needs more time to adapt and learn R compared to a team member who had the skillset from university or participated in preparation for the transition. It is important to acknowledge and give support to those building the R skill-set from zero.

6. Lessons Learned and Recommendations

Based on the experience so far, several lessons stand out for others considering a similar transition.

Build the foundation before the switch. Identifying key workflows and having equivalent functionality ready in R was the most effective way to reduce friction in the transition.

Run pilots in production, if possible. Having willing team members solve real tasks in R, while the rest of the team continues in SAS, ensures that problems are identified in a way that a controlled test environment will not. It also creates internal experts who can support colleagues once the full transition begins.

Do not migrate everything at once. Prioritise migration of code that is already problem-atic. Well-maintained, functioning code can continue running until a natural moment to replace it arises. This reduces risk and spreads the workload.

Leverage internal skills. Having a few team members already with strong R expertise helped a lot. Having team members who were able to drive the transition and help others was a key part of the transition. Therefore, consider the starting point of the

team: does it need to build the skills from zero, or can it leverage an already existing skillset?

You can always prepare more. A concrete example of where we could have been better prepared was working with large datasets in R. Compared to R (and Python) in their base form, SAS is a powerhouse for handling large datasets. In production when working with large activity-based datasets, where one person can have multiple observations, we have faced some issues using R, simply because it became slow and crashed altogether. However, several R packages specifically target this issue, so we are now exploring the *data.table* package (Barrett et al., 2026) and the *fastverse* (Krantz, 2026), which is a package collection specifically targeting this issue. In hindsight, we could have been prepared for this issue, but it is the nature of a transition that issues like this occur.

7. Conclusion

The Data Consultancy team's transition from SAS to R reflects a broader shift in how NSIs approach statistical production. Open-source languages offer genuine advantages: lower costs, better alignment with the current labour market, stronger international community support, and greater flexibility to adopt emerging technologies. For a team working primarily with rectangular register data and relatively short production pipelines, R proved well-suited to the task, both because of its statistical heritage and because of the maturity of its ecosystem for official statistics work.

The transition was not without difficulty. Productivity fell as expected, and some aspects of the SAS environment still have no perfect R equivalent. However, careful preparation, building a shared R package of common workflows, and piloting with real tasks substantially reduced disruption. By focusing early migration effort on the most problematic legacy code rather than attempting a wholesale replacement, the team maintained production continuity throughout.

Appendix: List of References

Barrett T, Dowle M, Srinivasan A, Gorecki J, Chirico M, Hocking T, Schwendinger B, Krylov I (2026). *data.table: Extension of data.frame*. R package version 1.18.99, https://r-datatable.com.

Krantz S (2026). *fastverse: A Suite of High-Performance Packages for Statistics and Data Manipulation*. R package version 0.3.4, https://fastverse.github.io/fastverse/.

Langsrud, Ø., Lupp, D., Bøvelstad, H., Klungre, V. N., & Lindblad, J. (2025). *GaussSuppression: Tabular Data Suppression using Gaussian Elimination*. R package version 1.2.0. Statistics Norway. https://CRAN.R-project.org/package=GaussSuppression

SNStatComp (2026). Awesome Official Statistics Software. https://github.com/SNStatComp/awesome-official-statistics-software

Statistics Denmark (2024). *Data Confidentiality Policy*. https://www.dst.dk/Site/Dst/SingleFiles/GetArchiveFile.aspx?fi=formid&fo=dataconfidentiality--pdf

Wickham H, Averick M, Bryan J, Chang W, McGowan LD, François R, Grolemund G, Hayes A, Henry L, Hester J, Kuhn M, Pedersen TL, Miller E, Bache SM, Müller K, Ooms J, Robinson D, Seidel DP, Spinu V, Takahashi K, Vaughan D, Wilke C, Woo K, Yutani H (2019). “Welcome to the tidyverse.” *Journal of Open Source Software*, **4**(43), 1686. doi:10.21105/joss.01686.