The best IDE for data science still needs to be built

Brian Carter

# Introduction

Data science involves many programming languages and each comes with its preferred integrated development environment (IDE). I know this because I’ve been doing it for over 20 years and I’ve used a lot of them. I started with SPSS and Stata in grad school before taking my first real job as a SAS programmer. I settled on R as my primary programming tool via Rstudio, but I’ve had jobs that required Python with Jupyter notebooks, VS Code, and others. I’m a multilingual data scientist because that is the reality of the career.

Today there are three pillars in data science: SAS, R, and Python. Many of us specialize in one or another, either due to preference or industry standard, but many of us work across all three. Unfortunately, there is no single IDE to serve us all. This makes any multilingual workflow cumbersome and inefficient because we must navigate data types, programming environments, and operating systems. Oftentimes we choose the one with which we are most comfortable, and [just argue about which is better](https://www.reddit.com/r/statistics/comments/1003gwv/d_how_popular_is_sas_compared_to_r_and_python/?utm_source=share&utm_medium=web3x&utm_name=web3xcss&utm_term=1&utm_content=share_button). But in reality, we simply don’t have a platform that will let us always use the best tool for the job.

I believe that the best IDE for data science will soon be available. This IDE will integrate these three dominant data science tools into a seamless workflow. We can take advantage of the best that each of these tools has to offer to more efficiently explore and understand our data.

## The three pillars of data science

SAS, R, and Python are very different but each is best suited for different tasks. In most cases, each can technically do the job of the rest, but they are not equally efficient.

### SAS

SAS is the best tool for the management and cleaning of tabular data. The SAS data step and PROC SQL are powerful tools for manipulating data and deriving new variables. SAS only includes two data types: numeric and character so the programmer rarely has to worry about type errors. SAS doesn’t have complicated rules for coding missing values. The IF - THEN - ELSE syntax is easy to understand and easy to read. SAS arrays and SAS macros provide a mechanism to automate repetitive tasks across similar datasets and variables. The SAS data step is memory efficient. Simply put, the SAS data step is easier to use and provides fewer opportunities for coding errors than any other data science tool.

That is where the advantages of SAS end. The SAS PROC system of statistical models is based on programming syntax from the 1970s. SAS is slow to incorporate the latest and greatest models. SAS PROC reliance on a tabular datasets makes advanced algorithms impossible. The ODS output system is difficult to navigate. And SAS graphics are difficult to program and generally look terrible.

### R

I fell in love with R because I could make beautiful plots. When [ggplot2](https://ggplot2.tidyverse.org) and [plotly](https://plotly.com/r/) came out, these beautiful graphics were even easier. As a statistical analyst, the basic modeling formula, y ~ x1 + x2 + x3 matches how I was taught to understand my first linear model. The reporting tools in R are powerful and flexible: [Quarto](https://quarto.org) makes it easy to create professional reports and presentations. The creativity and variety of the [20,000+ packages on CRAN](https://cran.r-project.org) constantly amazes and delights me. And I could argue that [Shiny](https://shiny.posit.co) is the coolest thing ever invented.

### Python

These days, Python is the most [popular programming](https://www.datacamp.com/blog/top-programming-languages-for-data-scientists-in-2022) language for data science. It is the *lingua franca* in computer science which allows data scientists to communicate among a multidisciplinary group of engineers and IT professionals. Python includes libraries like [Pytorch](https://pytorch.org) and [scikit-learn](https://scikit-learn.org/stable/) for high performance computing and machine learning. The bleeding edge of data science is developed in Python. Sure, there are packages in R that can do these things too, but these tasks are usually accomplished in Python.

## The IDEs

Given these tools, data scientists have three primary IDEs to choose from. The SAS language runs in the [SAS proprietary system](https://www.sas.com/en_us/home.html). Posit provides R and Python support in [Workbench](https://posit.co/products/enterprise/workbench/) and is by and large the perennial favorite in the R community. [VS Code](https://code.visualstudio.com) developed by Microsoft, is the [the most popular IDE](https://pypl.github.io/IDE.html#google_vignette) among all code developers and data scientists.

Yet, only one of these will execute code from all three pillars of data science, and that is SAS through its [SAS Viya](https://www.sas.com/en_us/software/viya.html?utm_source=google&utm_medium=cpc&utm_campaign=brand-global-sas-gen-gbc-gmd-global&utm_content=PLN_12707&dclid=&gbraid=0AAAAAD_WZgD2XRqmkSPKba-G9jjatIFbc&gclid=Cj0KCQjwxeyxBhC7ARIsAC7dS38cwswKCIWNcHoUXDZBVfTm3_KMyMS_n0UrOHJX9klyZYZss6dSo5kaAjtFEALw_wcB) product. SAS Viya is a cloud-based system that runs the SAS language, R, and Python. SAS understood that open source analytics was a threat to their business model for the reasons outlined above. [After initial efforts to squash the upstarts](https://www.nytimes.com/2009/01/07/technology/business-computing/07program.html) SAS decided to embrace multilingual data science and incorporate R and Python into their ecosystem.

But it is still SAS. It is still running on a closed system. It is still very expensive. Using it does not provide me the same efficient experience I’ve come to expect from Posit and VS Code. It was built by a company that has been around for 50 years and has a reputation for being slow to innovate. It is not the best IDE for data science, but it is the only tool that can do all three.

## The ideal IDE is ready to be built

Enter a dark horse: the [Altair SLC](https://altair.com/altair-slc) a SAS compiler released from Altair. This compiler can take SAS code and return SAS results without SAS. Users can run their data steps, PROC SQL, and even run models, without paying the high cost of locking into the SAS ecosystem. I’ve used their compiler and it works well. I will tell my clients that this is a viable alternative to expensive SAS licences.

Having a SAS-independent SAS compiler finally gives data scientists a mechanism to build a truly efficient data science platform that will natively run SAS, R, and Python. In a single project, I could include a SAS data step for easily manipulating and cleaning my data. Quickly jump to an R file in the next tab for running my initial models and creating a beautiful plot. Then switch to a Python file for implementing a machine learning pipeline. All this in one interface that maximizes the value of each of these tools.

Everyone wants this kind of deep integration. Posit has made an admirable investment in optimizing their platform for Python developers. VS Code has R extensions that could bring R developers into a common IDE with everyone else. SAS has delivered all three languages in Viya, but it is still SAS.

My company is currently discussing opportunities to develop a new tool for all three languages. Incorporating the Altair SLC compiler into an IDE plugin would immediately set that tool apart from the competition. Personally, I think this would be an excellent extension to the Posit platform, but it is only a matter of time before Microsoft adds a similar feature into VS Code to fully complete its offerings. One way or another, integrating Altair’s SLC into either of these IDEs would make it a truly killer data science platform delivering the preferred tools for every task.

## The future of multilingual data science

In a world where we seamlessly drift between programming languages, incorporating generative AI will become a vital tool. [I’ve discussed](https://procogia.com/ai-tools-for-easy-sas-to-r-migration/) using AI tools as a code migrator: [GitHub Copilot](https://github.com/features/copilot) is fully integrated into VS Code and Posit Workbench, and other LLMs are available as a Rstudio plugin through the [chattr package](https://cran.r-project.org/web/packages/chattr/index.html). These tools are already helpful in generating a good first-pass code migration, but incorporating a back end testing and evaluation service is a logical next step. It would require an IDE that could run all three languages in parallel and a back-end algorithm for testing and evaluating results. Statisticians trained in SAS could request their GLM procedures in SAS return an equivalent glm() call in R, and rest comfortably knowing their models were tested and approved prior to deployment.

Our team at ProCogia is working on just such a tool. Many of us are more comfortable programming in our preferred language; AI will provide us with just the little bit of help we need transition to a more effective and efficient workflow. Early results are very promising and we are refining a series of general-purpose prompts for finer instructions to the LLM.

## Conclusion

It is time to build the best IDE for data science. The tools are there and we can do it with an open-source ethos. The Altair SLC compiler is the key to unlocking this opportunity and I am excited to start developing this integration with our community.

I will be at the [R Stats conference](https://rstats.ai) representing the R Consortium in New York City, May 2024. I would love a chance to talk with you and your team about this exciting opportunity to integrate all our tools into a single master application. I am looking forward to hearing from you soon.