

Creative DATA SCIENCE using SAS and OPEN SOURCE (PP-19)

Stijn Rogiers – Matteo Landrò – Samiul Haque – Sherrine Eid - William Kuan – Antonio De Castro – Mark Wolff - Fadi Glor

BACKGROUND

COVID-19 has ACCELERATED the ongoing MODERNIZATION within CLINICAL DEVELOPMENT. The key industry drivers include:

- Hybrid DECENTRALIZED clinical trials
- The demand for SHORTER development TIMELINES
- The increased usage of REAL WORLD DATA
- DATA SCIENCE including AI (to make truly DATA DRIVEN decisions)
- OPENNESS

OBJECTIVES

We provide 3 Use Cases from within Clinical Development (Biometrics) how SAS embraces and extends OPENNESS.

Why OPENNESS

OPENNESS is not only adopting open-source languages, but also about PROVIDING MORE PEOPLE ACCESS to ANALYTICS, leveraging Low-Code/No-Code Visual Interfaces. The key question is HOW to best COMBINE all these options to speed up the ongoing transformation, while meeting CURRENT & FUTURE REGULATORY requirements. It's a decision of CHOICE and CONTROL.

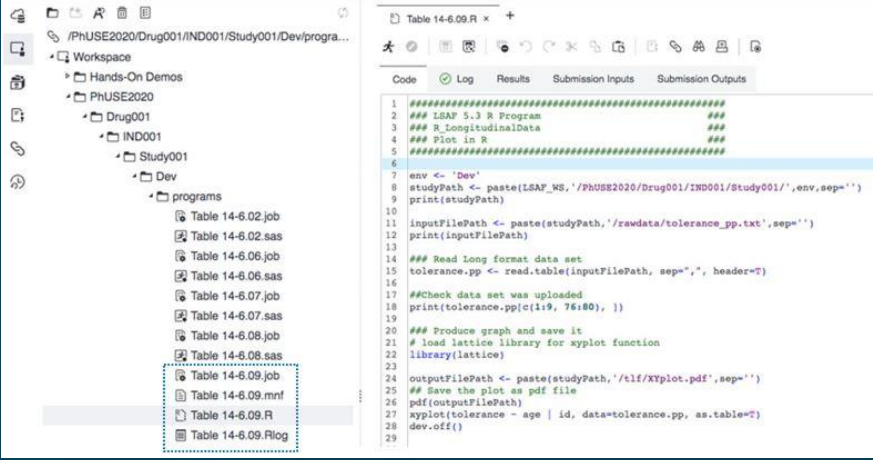
USE CASE 1 - CREATION OF TABLES, LISTINGS, FIGURES (TLF's) and Interim / Final Analysis

INDUSTRY SOLUTION APPROACH

New in **SAS Life Sciences Analytics Framework (LSAF) 5.3** (GA Nov 2020) is the integration of R. *Users can open a SAS session or an R session* and code program(s) for analysis. *The Program Editor supports the development of multiple programming languages* including SAS and R.; provides color coding and produces an R log (.Rlog).

Important to emphasize is *that R code is managed the same way than SAS code* in the environment with similar processes using jobs and manifests to *ensure traceability and reproducibility of each execution*. Users are also able to have SAS and R programs combined within the same Job.

PROs : Package Mgmt. and Execution of final SAS & R code within SAS LSAF; Auditability & Compliance; Enhanced Performance & Collaborations.



Papers/Presentations on this topic

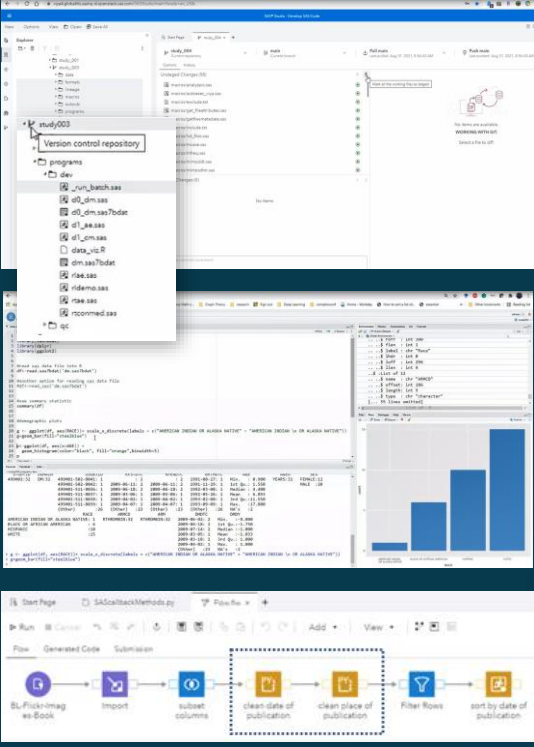
(1) Manage TLFs Development in LSAF using SAS and R (PharmaSUG 2021 Paper, Qualiance & SAS)
(2) PHUSE EMEA Autumn SDE 2021, DNR, R and LSAF - Challenges and Solutions

TECHNOLOGY APPROACH

Some of our life sciences customers are not focused on Commercial off-the-shelf platforms (or combine both approaches) and therefore investigate our Cloud Native **SAS Viya 4** analytical platform. The **SAS Viya 4** uses a *continuous integration, continuous delivery (CI/CD)* process that allows customers to choose their release intervals, so they gain access to the latest product innovations the moment they're ready or can incorporate updates into their own change-management schedules. This is especially interesting for exploratory environments where you want to make use of the latest and greatest of open-source packages. SAS Studio (on SAS Viya 4) is the latest development environment and provides an *out-of-the-box integration with GIT* (see picture), leveraging SAS Viya Compute (SAS9) or SAS Viya CAS engine. *R developers can work from within R Studio (on the SAS Viya platform) and connect to the same GIT repository.*

Presentations on this topic
PHUSE EU Connect 2021 – Leadership & Innovation stream:
The ongoing modernization journey at Orion Pharma (LI02)

Note: Since Oct 2021 (monthly CI/CD release) it is also possible to develop Python code directly within SAS Studio. The same request for R (already available in LSAF) has been added to the roadmap. See SAS Studio Flow example for SAS and Python code example.



USE CASE 2 - REAL WORLD EVIDENCE generation

Leveraging OHDSI (OMOP data) and HADES (R) packages.

HADES (formally known as the **OHDSI** Methods Library) is a *set of open-source R packages* for large scale analytics, including population characterization, population-level causal effect estimation, and patient-level prediction. In this specific Use Case (*demo purpose only*) we are looking *at patients that started using Nonsteroidal anti-inflammatory drug* (NSAIDs) for the first time, trying to predict who will develop a gastrointestinal (GI) bleed in the next year.

(1) **Import packages**

```
library(dplyr)
library(tidy)
library(randomforest)
library(caret)
library(lme4)
library(lmerTest)
library(lmerModel)
library(lmerModel)
```

(2) **Load Eunomia Data**

```
connectOHDSI <- Eunomia::getOHDSIConnectOHDSI()
eunomia <- createOHDSIConnectOHDSI()
```

(3) **Run LassoModel**

```
lassoModel <- setLassoModelRegression(seed = 0)
lassoResults <- runLassoModelRegression(population = population,
                                       alpha = alpha,
                                       modelSettings = lassoModel,
                                       testSet = "person",
                                       restriction = "RFE",
                                       nfold = 2,
                                       nrepeats = 10)
```

(4) **Promote table into CAS.**

We continue from here with making the data "analytics ready" (each patient having a single row and each column a specific covariate) so *we can load the data into CAS*.

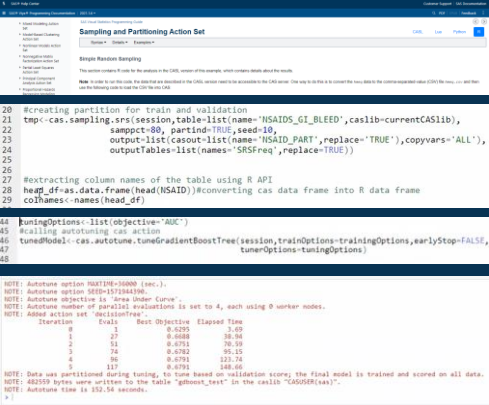
```
realTimeCasTable(cas, cas_id, casOut = list(casName="HADES_GI_BLEED",casOut="HADES_GI_BLEED"))
```

(5) **Loading SAS CAS Actions**

As we are interested in the modeling part we first need to load different Action Sets. We want to leverage SAS Viya *Auto-Tuning; Sampling* (to partition our data) *and decisionTree* (as we want to use 'Gradient Boosting' model that is part of The Decision Tree action set.)

```
15 #loading cas actions
16 loadActionSet(session, 'autotune')
17 loadActionSet(session, 'sampling')
18 loadActionSet(session, 'decisionTree')
```

As always, extensive SAS documentation is available for your guidance (see example 'Sampling and Partitioning Action Set' – **R section**).

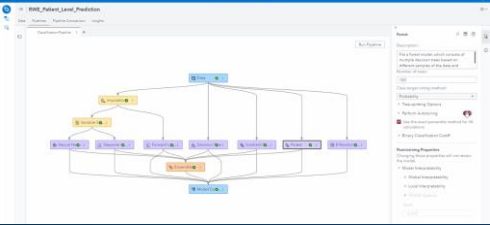


(6) SCORING the validation set

```
library(caret)
library(tidy)
library(randomforest)
library(caret)
library(lme4)
library(lmerTest)
library(lmerModel)
library(lmerModel)
```

(7) **SAS & Open-Source Model Comparison**

One other great capability is to build pipelines (drag & drop) *comparing out of the box SAS models* (6 in example below) *as well as open-source models* (1 in example below – in this case we selected an R Random Forest).



Note: In other use cases where python would be used (instead of R) users can leverage **Python-SWAT**. For additional resources for developers on SAS and open-source, see also **DEVELOPER.SAS.COM**. For more information on Use Cases in Clinical Development (e.g., Trial Feasibility, Synthetic Control Arms, ...) please also attend the RWE Hands-On Workshop at PHUSE EU Connect 2021 or listen to the recorded session.

For more info on Model Management and Model Performance over time - for both SAS and open-source models – see Use Case 3.

USE CASE 3 - HYBRID DECENTRALIZED CLINICAL TRIALS

HOW DIGITAL HEALTH IS TRANSFORMING CLINICAL DEVELOPMENT: THE ROLE OF DIGITAL ENDPOINTS, BIOMARKERS AND SaMD

Life Sciences companies have indicated that *decentralized trials* will be a part of their model for study conduct going forward, 75% have indicated an expectation that trials will be "hybrid" with a blend of "virtual" approaches. Critical to these opportunities will be digital biomarkers and endpoints, as well as advanced analytics including AI/ML. **Digital Biomarkers** are derivative of physiological data collected by means of digital devices such as wearables / sensors for example.

CRITICAL COMPONENTS TO BE INTEGRATED

- ✓ Sensor (sensing and communication hardware)
- ✓ Software (data acquisition and processing)
- ✓ Advanced Analytics (incl. AI/ML)
- ✓ Network (secure and reliable communication)

In this specific use case (*demo purpose only*) we demonstrate the identification of novel digital Biomarkers using sensor technologies such as accelerometers and gyroscopes and the application of machine learning techniques to *develop algorithms and models that describe patients balance and stability*. These models are in effect quantifying new clinical endpoints that then are used to assess disease state and therapeutic efficiency. The **Berg Balance Scale** involves qualitative judgment from a trained clinical observer to assess each patient based on several required behaviors. Our objective is to not only *digitize this assessment* by removing the human, and introducing a *more quantitative sensor based approach*, but to also *reduce the complexity* of the protocol using more sensitive digital biomarkers. As stated earlier, for this poster we are only *focusing on the 'openness' aspects of the Analytical Life Cycle* supporting this Use Case.

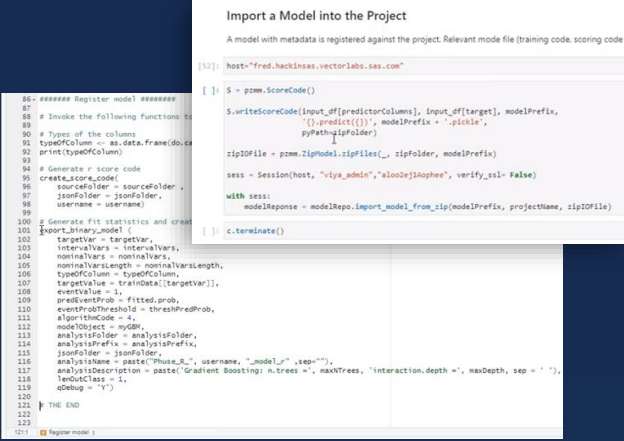
If you are interested to get the full story, please check out the material from [the PHUSE EU Connect 2020 Workshop](#) or contact the authors on this poster to organize a specific workshop for your company.

(1) **Develop & Train R & Python models**

The 2 screenshots on the right hand side illustrate the flexibility of *developing and training models in R or Python natively* on the SAS Viya platform, while still connecting to the *data loaded in CAS*. As explained in Use Case 2 there is the possibility from within R or Python to leverage CAS Action Sets where relevant in your R/Python code.

(2) **Import R & Python models into SAS**

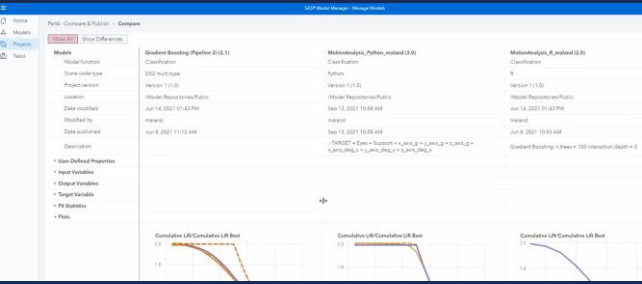
As a next step we import/load these trained models in SAS (model management) in order to automatically *compare SAS, R and Python models to select the best performing or/and most explainable model*.



(3) **Automated Diagnostics & Model Evaluation (out of the box)**

Once your models that you created in R/Python and loaded into SAS are available, you can compare automatically these models with each other or/and with SAS models. Moreover, you can also compare with open-source models you created as part of the model pipelines (open-source Nodes) within SAS – see screenshot Use Case 2 above. There is a lot of flexibility to accommodate the preference and skillset of each user.

Below you can find an example of such a comparison report. Finally we also have the option to run Performance Reports to follow up on models over time. Very important when considering **SaMD**.



Finally, we want to brief you that SAS organized in 2020 a first big **HACK-IN-SAS** (one month hackathon) on the latest SAS Viya platform on MSFT Azure. *What is exciting is that nearly all use cases within the Life Sciences and Health Care track (i.e., HEALTH TRACK) are demonstrating the EMBRACE and EXTEND message on SAS and open-source; Coding & Visual Interfaces*. One of these use cases has been build upon our balance and stability prototype.

Check out team: **Red Hot Chili Steppers**. Enjoy the video and stay tuned for more Hack-a-Thon opportunities. The next Hack-a-Thon (Q1 2022) is in planning phase.

