

SAS® GLOBAL FORUM 2021

Paper 1072 - 2021

Open Source SAS Macros – What? Where? How?

Katja Glaß, Katja Glass Consulting

ABSTRACT

Open Source provides a lot of opportunities to work smarter. This presentation will give an overview about open-source solutions which are available for SAS. The presentation is going to briefly explain how open-source can be applied in companies and organizations, what licenses are available and what other important aspects to consider for example when it comes to maintenance, documentation and validation. The core of the presentation will be the overview of the various solutions to allow a brief overview of what is available. These are SAS macros, scripts and related tools.

INTRODUCTION

Why to create new source code for topics where there is already something available? Do we really need to re-invent everything? Using open source enables us reusing source for various purposes, to adopt to our problems, to learn, to share, to collaborate and specially to standardize through collaboration!

But what is open source?

Open Source software is software that can be freely accessed, used, changed, and shared (in modified or unmodified form) by anyone.

BASICS

The core point of open-source software is the availability of the source code. Looking for the SAS® Software, it could be SAS® macros or scripts or it could be related to SAS® and written in any other language. Open-source programs can be used very flexible in various areas. The license is a very important aspect to consider. If there is no license attached, the copyright can be assumed.

The main advantage is that software and source code can easily be used, modified and shared. Open source is very common in various business fields. It is heavily used in web development. Even complete operating systems like Unix and Android are open-source and developer environments like Eclipse are available. There are many applications, source codes and code snippets available in the public as open source. Unluckily there are just a few solutions available related to the SAS® software.

OPEN-SOURCE MOTIVATION

Typical open-source providers are single persons who create projects on their free time as they are highly motivated. As SAS® is mainly used in business area, there are fewer people and less use-cases for a private motivation. Sometimes people want to show their knowhow, which is why especially the macro solutions are made available by consultants or small consultant companies. More solutions are coming as a result of working groups and even in paper solutions are available – unluckily papers often miss a license statement.

Finally, there are also business models for open source – but in the software environment for SAS® this is hardly to find.

The motivation to create open source can be different, but typically is driven by the collaboration idea. The functionality is the core for open-source solutions. To develop a solution and show that it works is an integral part for software development. This is what is created with the highest priority and motivation. Documentation and also communication is somehow something which is quite necessary to gain publicly and findability. For this most open-source project contains documentation as well when the motivation is high enough. Nevertheless, there are also hidden gems where very valuable functionality as open-source is available, but the documentation is missing due to a lack of motivation.

Figure 1: Motivation line for open-source developers



Looking for other aspects of a software development, for example the training aspect, the motivation is typically not sufficient for this. The validation motivation is usually the lowest and for this only in rare cases tests or even validation documentation will be available for open source projects.

LICENSES

There are various open licenses available. If source code is available in posts or papers where no license is attached, then the copyright is applicable. To allow others to reuse code easily a license should ideally be applied. There are two major groups of open-source licenses – permissible and copyleft licenses.

PERMISSIBLE LICENSES

The permissible license group is the most flexible and straight forward license type. It can be compared to the BY-Attribution of the Creative Commons license for other works. The license itself typically explains how to mention the creator. The “unlicenses” license can be seen as public domain knowledge and can be used without any limitations – no notifications or the creator is required.

For other permissible licenses like the “MIT”, the source code can be used, but additional information must be kept. Some license types like the “Apache License 2.0” also specify how to document modifications. “BSD” additionally prohibits endorsements and promotions for derived products of the original creators or contributors. The “MIT” is the most common open-source license used. This license is very easy and slim, the only thing to do:

“The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.”

Meaning, when use, modify or share this or deriving code, just keep the complete copyright notice.

The derived code can use any kind of license. It could even be proprietary code or a software which can be sold. Just make sure to deliver the original copyright-notice as well.

COPYLEFT LICENSES

The copyleft category of licenses contains additional rules to allow users of derived code the same rights like received. This is called copyleft. When receiving software with the rights to use, share, modify, users of the newly derived source should get the same rights granted. The most common licenses used are those from the GNU-family. The core license is the GNU General Public License (GPL). All rights received must be provided the users of the derived work as well.

The Lesser General Public License (LGPL) states that the software license for the derived work must not be shared when using the LGPL parts only as libraries. This license is more open to support commercial software. The Affero General Public License (AGPL) in contrast is stricter and requires the grant to use, modify and share rights for derived work also when not the software is provided to the users, but software as a service (SaaS) is used.

REMARKS

When using open source only within a company or organization – for this no external users available - then the company or organization is considered as one group and not as other users. Finally, the derived source code from copyleft licenses can then only be kept within the companies without the need to further publish code.

But when additional parties are involved like Clinical Research Organizations (CROs), academia or authorities, these become new users, so in accordance with the corresponding copyleft license, these users must receive the same rights and for this likely also the source code.

When user must be provided with the source code, this does not mean that the source code must be published in the internet to the public. It only means that the users must be able to get the source code. As the users have the same rights, they are also allowed to use, modify and share the source code, so finally source code can be shared through the internet to everyone by another user. But this is only a can, not a must.

AVAILABLE SOLUTIONS

Most open-source tools are using permissible licenses and are very easy to reuse. One core question still remains, this is what open-source solutions exist for specific tasks? It is very difficult to find open-source solutions. There is no global platform where open-source software can easily be announced. For the programming language R, there exist the CRAN system, which can easily be searched. Quite often presentations on conferences are used, but they typically also just have a limited range.

OPEN SOURCE PORTAL FOR CLINICAL STUDY EVALUATIONS

To overcome the issue of finding open-source solutions, Katja Glass Consulting has created an open-source portal for clinical study evaluations where various solutions in the clinical study area can be found. A lot of generic SAS solutions are available in this portal as well. Not all solutions are for clinical study evaluations, but can be used in any other context.

Figure 2: Screenshot of the Open Source Portal for clinical study evaluations



The portal provides an overview of tools available in the “Overview” tab. Many details and links to further information like the source locations, available presentations and more can be found. A table view can be used to search for specific keywords, a license, a working field or similar.

The “Programs” view allows to search for specific open-source macros or programs. By now nearly 700 are included from nine different sources. Sometimes similar macros are available from different sources, like compare macros. This can easily be investigated and found through the search.

When filtering the “Overview” for solutions related to SAS, there are 17 different open-source tools found:

- SAS Macros
 - FDA Jumpstart Scripts
 - RhoInc Plots
 - SASjs Core Macros
 - SMILE – Smart SAS Macros
 - SAS Macros by Scott Bass
 - Roland's SAS® Macros,
 - Chris's SAS Macros
 - ...

- SAS Scripts
 - PHUSE White Paper Central Tendencies Scripts
 - SAS® Blog
 - DefineXML SAS XMLMAP
- Tools
 - SASUnit
 - Reindeer – Render SAS Results into Word
 - StatTag

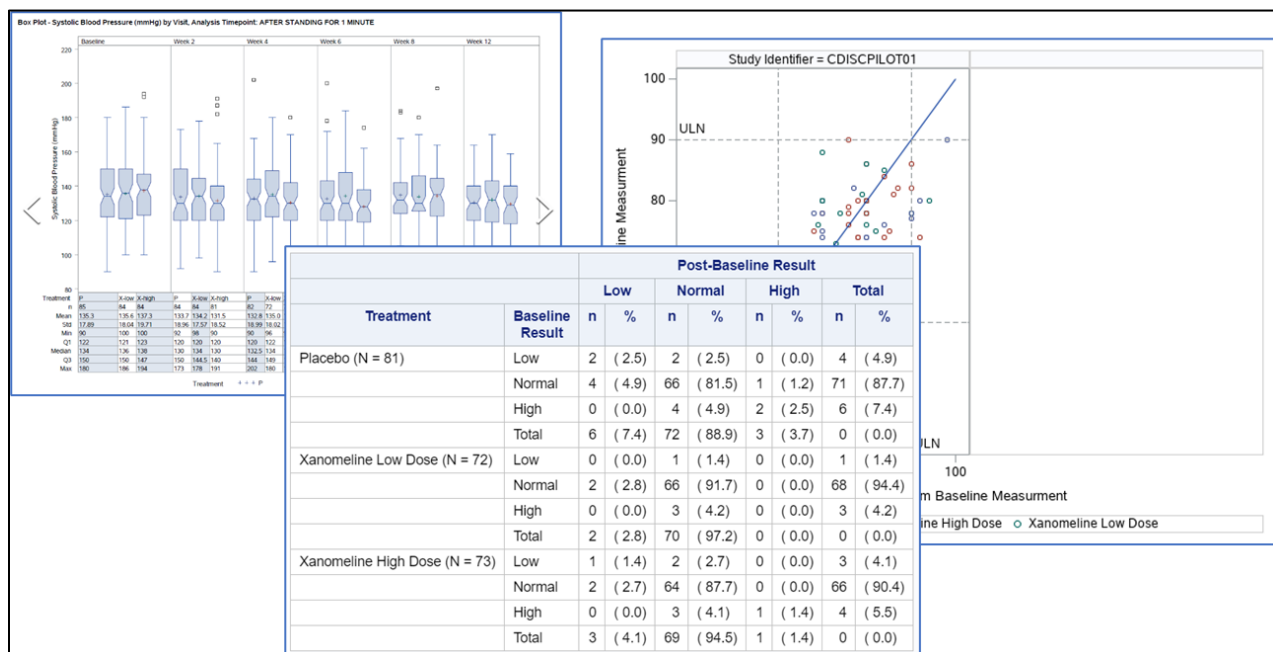
OPEN-SOURCE SCRIPTS & MACRO SOLUTIONS

There are very valuable scripts and macros available using the programming language SAS. PHUSE as a huge membership organization for the pharmaceutical industry has various working group projects, where the results are made available as open source for everyone. The first example is the PHUSE White Paper Scripts.

PHUSE White Paper Scripts

A working group has worked out various standard tables and figures and collected them in a “White Paper”. Some of these tables has then be programmed and the source code is made available in the PHUSE Script Repository in GitHub which is using the permissible MIT license.

Figure 3: Examples from PHUSE White Paper Scripts



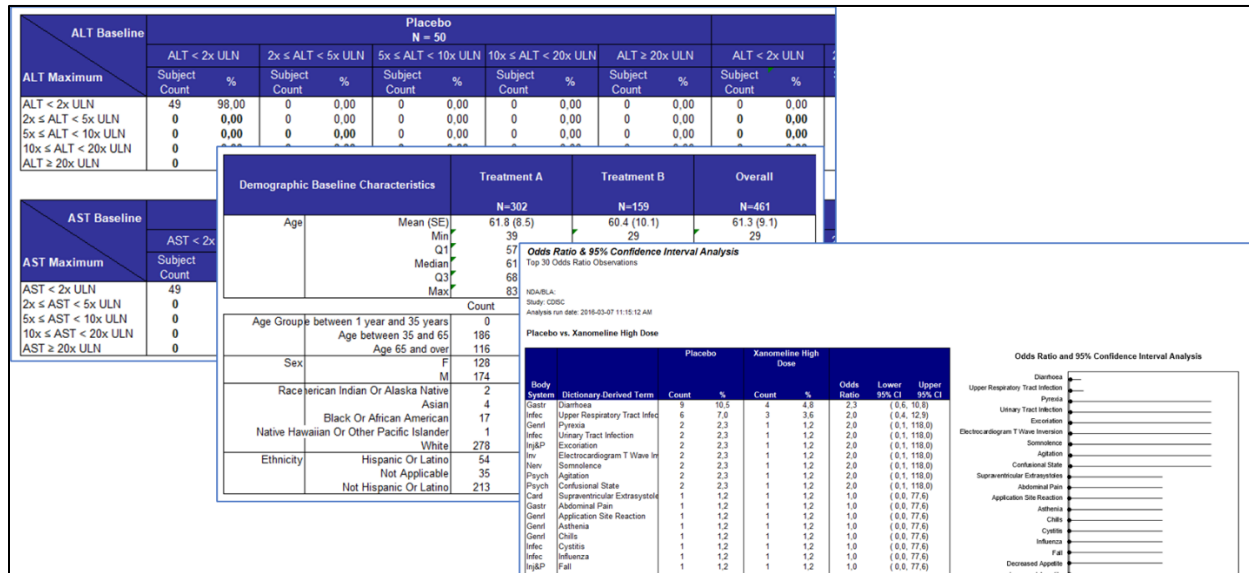
There are various examples available of complex graphics and tables. The great thing is that the source code can be used to learn how to create the required data modifications and the statements to create these outputs. The code can easily be used in own projects to adopt for other data.

FDA Jumpstart Scripts

The PHUSE organization also received a complex macro framework from the FDA for their open-source initiative. These macros and scripts are created around 2011 and create standard outputs based on the clinical data standard SDTM. The outputs are created in Excel format using the EXCEL libname engine which is only available for Windows versions together with Excel Templates.

Documentation and even specifications are available which are very detailed. A PHUSE working group had also been working on qualifications and made these available in the PHUSE script repository on GitHub as well. This complex framework can be used as basis to create standard outputs and graphics. By using the MIT License, this toolset can easily be used and modified.

Figure 4: Example outputs from FDA Jumpstart Scripts

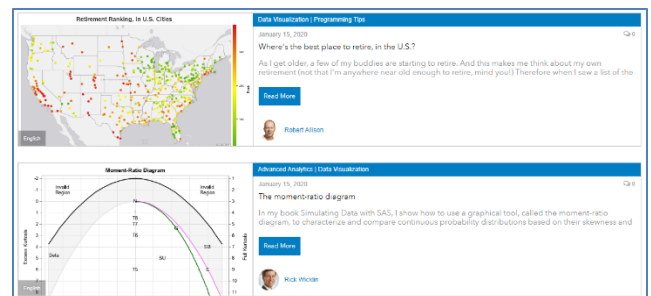


Due to the complexity updating the macros might be a bit tricky. The Excel Engine is also a bit challenging as this is not available on SAS-Unix environments. But an implementation is possible. Especially when the outputs should be used in different formats than excel, for example creating ODS outputs or graphics, the required update steps are feasible.

Data Visualizations – SAS Blog

SAS posts regularly new blogs, one area is the blog for data visualizations. Here various developers show fantastic graphics which can be created with SAS. Often a detailed description and the source code is available. Unluckily there is no license mentioned, so the copyright is applied. I would expect that the source code might still be used, as this is coming from SAS directly. To be sure the author must be contacted to ask for permission for reusing the code. An alternative would be to use the ideas from the source code and perform an own implementation.

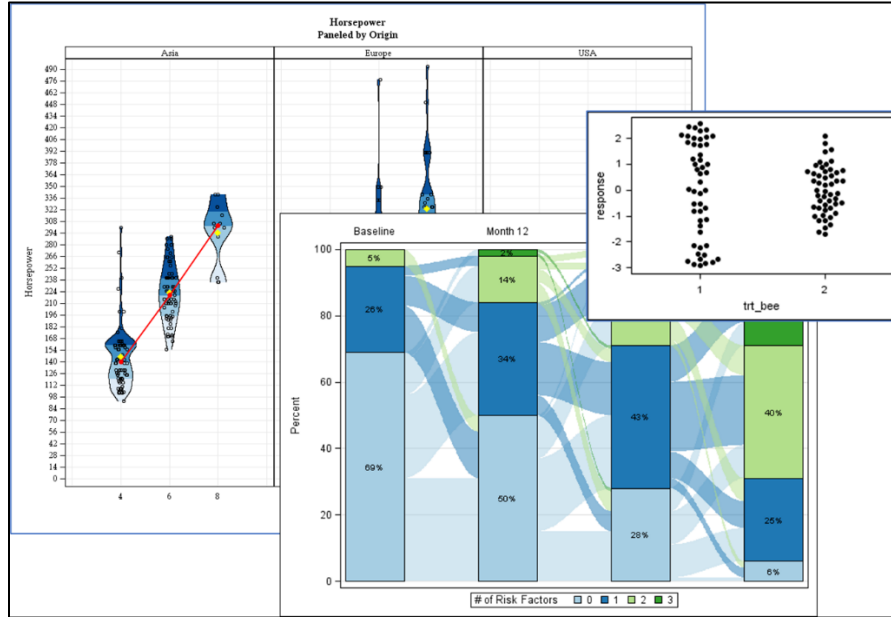
Figure 5: Example for SAS Blog



Rholnc Plots

Rho is a company which has released graphic macros as open-source. There are three graphic macros to create specific charts: Violin Plot, Bee Swarm Chart and Sankey Bar Chart. These macros are very generic and flexible and can be applied very easily to any kind of data. To each of these graphics a paper exists describing many details.

Figure 6: Example Plots



SASjs Core Macros

SASjs is an open-source framework to create various web applications connected with SAS. Next to the framework suite, SASjs contains many macros which are very valuable for various purposes. These macros contain prefixes which likely do not have naming conflicts with other macros like corporate macros.

There are for example various macros available to receive information, like to get the file size, get values, get the variable type and similar. Many check macros are available, like checking whether a dataset exist, a variable exist in a dataset. There are macros available for complex tasks like to zip and unzip files, search a text in a library, list files and directories, perform a binary copy and many more.

The SASjs kit has special macros available which are useful to organize and manage SAS Viya® as well as the SAS Metadata® server. These actively maintained macros are made available using the MIT license and can be used in a very generic way. Allan Bowe is the main contributor and maintainer of this package. A detailed macro documentation is available. Furthermore, the SASjs web framework contains a detailed documentation including examples and even videos.

The macros are packed into a single file and can easily be used in any SAS session having internet connection using the following two commands:

```
FILENAME mc url  
"https://raw.githubusercontent.com/sasjs/core/main/all.sas";  
%INCLUDE mc;
```

Figure 7: SASjs Macro Core documentation

%MACRO CORE (Production Ready Macros for SAS Application Developers) <https://github.com/sasjs/core>

MacroCore

- File List
 - base
 - lua
 - meta
 - metax
 - viya
 - mv_createfolder.sas**
 - mv_createwebservice.sas
 - mv_deletefoldermember.sas
 - mv_deletejes.sas
 - mv_deleteviyafolder.sas
 - mv_getaccesstoken.sas
 - mv_getapptoken.sas
 - mv_getclients.sas
 - mv_getfoldermembers.sas
 - mv_getgroupmembers.sas
 - mv_getgroups.sas
 - mv_getjobcode.sas
 - mv_getjoblog.sas
 - mv_getjobstate.sas

mv_createfolder.sas File Reference

Creates a viya folder if that folder does not already exist. [More...](#)

[Go to the source code of this file.](#)

Detailed Description

Expects oauth token in a global macro variable (default ACCESS_TOKEN).

```
options mprint;
%mv_createfolder(path=/Public)
```

Parameters

- path=** The full path of the folder to be created
- access_token_var=** The global macro variable to contain the access token
- grant_type=** valid values are "password" or "authorization_code" (unquoted). The default is authorization_code.

Version

VIYA V.03.04

SMILE – Smart SAS Macros

SMILE contains Smart SAS Macros – an intuitive library extension. Next to useful macros, example programs are available which are explained in detail on the documentation page where also the macros are documented.

Currently the most important macros are dealing with PDF documents. There is a macro to work with ODS DOCUMENTS which are modified to create flat bookmarks. For this case many examples are made available. Another nice macro can merge multiple PDF documents into one document. The (currently) last PDF related macro reads in the bookmarks including the level and page link into a SAS dataset. This makes it easier to program quality checks.

Figure 8: SMILE - Flat navigation PDF output example

ods_document_flat1.pdf - Adobe Reader

Datei Bearbeiten Anzeige Fenster Hilfe

Öffnen

1 / 10 125%

Lesezeichen

- Table 1: By Group Report about shoes
- Table 2: Table Class Output
- Table 3: Multiple outputs - Cars for make = Acura
- Table 4: Multiple outputs - Cars for make = Audi
- Table 5: Multiple outputs - Cars for make = BMW
- Table 6: Different label
- Figure 1: Class graphic

Table 1: By Group Report about shoes

Region=Canada

Product	Total Sales
Boot	\$17,720
Boot	\$40,213
Boot	\$7,892

As the SASjs core macros, the macro names have a prefix to avoid naming conflicts. Furthermore, these MIT licensed macros can also be included in a SAS session simply with the following statements:


```

FILENAME mc url "
https://raw.githubusercontent.com/KatjaGlassConsulting/SMILE-
SmartSASMacros/main/all.sas";
%INCLUDE mc;

```

SAS Macros by Scott Bass

Another valuable macro package is made available by Scott Bass who is an Australian consultant who made his macro available using the unlicensed license. When embedding and modifying these macros, not even a mentioning would be required.

He is still very active so from time to time there are updates and new macros. The macros are mainly supporting macros and the documentation is available in the header. Macros of this library are for example "RunAll" to allow for asynchrony program starts, a macro to align numbers according decimal place, export macros to Excel, CSV and more, a log parse macro to gain performance statistics from the log and many more.

One of my personal favorites is the compare macro which can easily compare two libraries and provides a nice summary dataset about the findings additionally to the detailed PROC COMPARE outputs.

Figure 9: Example Output Dataset of "compare" of two libraries

Library comparison report between adam and adam_mod libraries									
Flags			Base			Compare			
Base	Compare	Matched?	Libname	Memname	# of Obs	Libname	Memname	# of Obs	Obs Diff?
1	1	MATCHED	ADAM	ADAE	1,191	ADAM_MOD	ADAE	1,191	
1	1	MATCHED	ADAM	ADSL	254	ADAM_MOD	ADSL	254	
1	1	MATCHED	ADAM	ADTTE	254	ADAM_MOD	ADTTE	241	<<<
1	1	MATCHED	ADAM	ADVS	32,139	ADAM_MOD	ADVS	32,139	
0	1	NO MATCH			.	ADAM_MOD	NEW	1	<<<
1	0	NO MATCH	ADAM	ADADAS	12,463			.	<<<
1	0	NO MATCH	ADAM	ADCIBC	730			.	<<<

Rolands SAS Macros

Roland Rashleigh-Berry published many macros and a complex evaluation framework for legacy data some years ago. The focus of these macros was mainly to use them on his own and for his huge toolset. For this the intension for documentation was not too high, but information about the macros is to be found in the header as well as having speaking macro names. Unluckily he faded away, so there are no updates.

Some of the macros are:

- age (Age from Date)
- allunique (Unique over a library)
- char2num (Conversion)
- combine (Combine data sets)
- delmac (Remove macros from SASMACR)
- dslabel (Get dataset label)
- flatten (Reduce dataset to 1 per BY)
- ... (243 Utility Macros)

OPEN-SOURCE TOOLS

Next to scripts and macros more complex tools are available to support various SAS processes.

SASUnit

SASUnit is a validation and quality check framework for SAS programs, macros and processes. This toolset supports the validation development process as well as validation execution and automatic documentation. It has been available since 2010 and is continuously enhanced, so by now there is excessive functionality and documentation available. The tool is open source using the GPLv3 license.

Figure 10: SASUnit example validation report

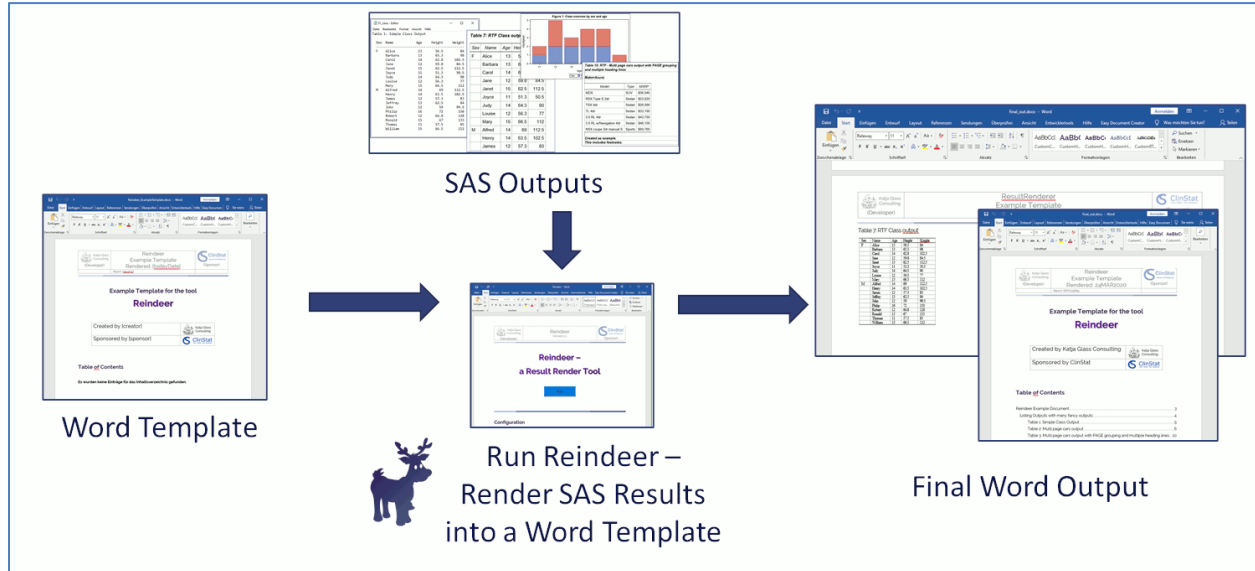
SASUnit Examples		No.	010		
<ul style="list-style-type: none"> ▼ Scenarios ▶ saspgm/assertexternal_example_test.sas ▶ saspgm/assertimage_example_test.sas ▶ saspgm/asserttext_example_test.sas ▶ saspgm/boxplot_test.sas ▶ saspgm/comparison_test.sas ▶ saspgm/crossreference_test.sas ▶ saspgm/database_test.sas ▶ saspgm/generate_test.sas ▶ saspgm/getvars_test.sas ▼ saspgm/nobs_test.sas <ul style="list-style-type: none"> ▶ 001 ▶ 002 ▶ 003 ▶ 004 ▶ 005 ▶ 006 ▶ saspgm/programdocumentation_test.sas ▶ saspgm/regression_test.sas ▶ Units under Test ▶ Program documentation 	Scenario Tests for nobs.sas - has to fail! Program saspgm/nobs_test.sas Last Run 12JUL2017:10:19:13 Duration 2.5 s				
		Test Cases			
No.	Test Case	Unit under Test	Last Run	Duration	Result
001	simple example with sashelp.class	nobs.sas	12JUL2017:10:19:14	0.1 s	
002	failed test - must be red!	nobs.sas	12JUL2017:10:19:15	0.1 s	
003	example with big dataset	nobs.sas	12JUL2017:10:19:15	0.3 s	
004	example with empty dataset	nobs.sas	12JUL2017:10:19:15	0.1 s	
005	dataset not specified	nobs.sas	12JUL2017:10:19:15	0.1 s	
006	invalid dataset	nobs.sas	12JUL2017:10:19:16	0.1 s	

Reindeer – Render SAS Results

Reindeer is a tool created in VBA for Microsoft Word® and can render various SAS results like listing, RTF, TAGSETS.RTF output and graphics into generic flexible Word templates. This tool is using the MIT license and was sponsored by ClinStat GmbH.

By now the tool supports batch processing and can even store the final result as PDF and not only as Word documents. Excessive documentation is available in the tool itself, which is a Word document with the embedded VBA macro.

Figure 11: Process overview for tool Reindeer



The content specification – for example which files should be created, which inputs should be used, general settings– can either be done in the Word file itself or in configuration text files which could easily be created in SAS together with the outputs itself.

Figure 12: Example for Reindeer - Render SAS Results

The figure shows the configuration and rendering of a Word document. The left side displays the configuration text in a text editor, including metadata and table references. The right side shows the rendered document, which includes a header with logos, the title 'Example Template for the tool Reindeer', and a 'Table of Contents' section.

Model	Type	MSRP
MDX	SUV	\$36,945
R5X Type S 2dr	Sedan	\$23,820
T5X 4dr	Sedan	\$26,990
TL 4dr	Sedan	\$33,195
3.5 RL 4dr	Sedan	\$43,755
3.5 RL w/Navigation 4dr	Sedan	\$46,100
NSX coupe 2dr manual S	Sports	\$89,765

Table of Contents

- Reindeer Example Document 2
- Listing Outputs with many fancy outputs 3
- Table 1: Simple Class Output 4
- Table 2: Multi page cars output 5
- Table 3: Multi page cars output with PAGE grouping and multiple heading lines ... 9

The table of contents, a navigation bar and similar can easily be created with the Word functionality. Word is also extremely flexible with respect to design other contents like a cover page, different portrait and landscape formats and for this might be easier to create designed documents.

StatTag – Plugin for Microsoft Word®

StatTag is another open-source tool using the MIT license to support SAS processes within Word. It is a free plug-in for conducting reproducible research and creating dynamic documents using Microsoft Word with the R, Stata and SAS statistical packages.

StatTag allows users to embed outputs like values, tables and figures directly within Word including the source code to create these. It provides an interface to edit code directly from within Word. Then the code execution can be triggered and the results are stored in the corresponding places in the Word document.

This tool is a great opportunity to design final documents with fluent text and updating the numbers, tables and figures by button click.

Figure 13: StatTag Toolbar Overview

The StatTag Word Toolbar

The StatTag toolbar has two groups of buttons which provide quick access to key functions

Manage Functions

	Code Files	Opens the Code Files Manager. See "Code Files Manager" below in Section 3.3 .
	Tags	Opens the Tag Manager. See "Tag Manager" below in Section 3.3 .
	Document Properties	Controls default formatting behavior applied to the active document, See "Document Properties" below.

Figure 14: StatTag Example Content

Education and anthropometrics in US adults: National Health and Nutrition Examination Study 2013-2014

Introduction: Education has been shown to be associated with body mass index (BMI). Gender, race, marital status and metabolic characteristics may alter this association.

Methods: This study included adult (≥ 30 years) participants from the 2013-2014 National Health and Nutrition Examination Study (NHANES). Demographics were self-reported and anthropometric measurements were taken by study personnel. Education was dichotomized based on matriculation to post-secondary education. Associations were estimated using T-tests or Wilcoxon rank sum tests for continuous data and chi-squared tests for categorical data. We examined the association between BMI and education level using multivariate linear regression.

Results: Among 38114808 participants, 20322649 (5.35%) self-reported any post-secondary education. Participants with post-secondary education had lower BMI (Beta: -0.3-0.6, 95% CI: -0.8-1.0 to 0.1-0.2) than those without post-secondary education. After adjusting for gender, race, age, marital status, fasting glucose and total cholesterol, post-secondary education no longer had significantly lower BMI (Beta: 0.2-0.2, 95% CI: -0.4-0.8 to 0.80-4).

Table 1. Association of Education with Participant Characteristics among 2013-2014 NHANES Participants.

Characteristic	No Post-Secondary (n=17792159)	Post-Secondary (n=20322649)	P-value
Male (%)	873 (49.1)1079 (50.0)	931 (45.8)1208 (45.6)	0.048 0.003

ASPECTS TO CONSIDER

The most important aspect to consider is to check the **license**, so what can be done with the tool. Typically, open-source tools use common licenses like MIT, GPL and similar which are very easy and open to use. Some tools use very custom licenses with restricted rights.

Please check them out accordingly. There are webpages providing a nice overview of the common licenses and what needs to be considered for these.

Open-source tools does not provide any **warranty**. This is an important aspect for the developers as otherwise it would be difficult to provide source-code as any code might contain bugs and open-source developers must not be at risk as otherwise they would not offer open-source. As validation needs a lot of effort, typically also this aspect is not covered by the majority of tools.

There are various mechanisms for automated tests in repositories and for various programming languages like Java there are typically also tests published together with source code. But this really depends on the scope of the open-source project and is for those related to SAS hardly to find.

The core functionality that tools are created is typically no **challenge**. Due to motivation this is what the open-source developer wants to provide to the community. But when it comes to communication, that's where the challenges start. Maintenance and fixes cannot be assumed when developer spend their free time to provide open source. But typically, many developers are motivated to support and update their tools and scripts when bugs are found and reported. Whether enhancements and documentation are available strongly depends on the motivation of the developer. The motivation for validation and quality checks is typically the lowest and for this hard to find in open-source solutions.

But how to overcome these challenges and how to **enable more open-source**? The following core aspects should be supported to support open-source:

- Allow employees & contractors to publish open source
- Join open collaborations
- Invest in open source
- Apply license to papers, blogs

Especially investments can support the motivation to create open-source including valuable documentation.

To **create open-source** is easy and a great experience. The SAS® OnDemand for Academics (successor for SAS University Edition) can be used for non-commercial educational purposes, so can easily be used to create open-source. To allow anyone to use the source, a license should be applied. The most common and very easy license is the MIT which I personally would recommend. When commercialization of source-code reuse should be avoided, then the GNU licenses might fit better. Various platforms exist to publish open-source projects. The most successful one is GitHub, but also SourceForge, GitLab and other platforms exist to deploy source code to the world. Finally, people should get aware of the solutions. The communication is very important. Typically, platforms are social media, conferences and also the open source portal for clinical study evaluations.

In Summary, the following steps can be done:

- Create open-source with SAS® OnDemand for Academics
- Apply a license to the code, for example MIT
- Publish source via GitHub
- Communicate via social media, at conferences and in the open source portal

CONCLUSION

Open-source provide a high potential to simplify our work. There are quite some solutions already available and more are coming. Check out the open source portal for clinical study evaluations to find them.

It's on us to exploit the potential!

Furthermore, it's on us to enhance the potential!

Allow your employees and contractors to publish source code. Join open collaborations which create open-source. And a very important point: invest in open source! Then more and more solutions and adaptations will rise and enhance our daily work dramatically!

REFERENCES

Website – Open Source Portal for Clinical Study Evaluations. Accessed April 9, 2021. <https://www.glacon.eu/portal>.

Website - Open Source Initiative - FAQ. "What is "Open Source" software" Accessed February 5, 2021. <https://opensource.org/faq#osd>.

Website – Choose a license – overview of open source licenses. "What is "Open Source" software" Accessed February 5, 2021. <https://opensource.org/faq#osd>.

Website – Open Source Statistics. Accessed May 10, 2020. <https://resources.whitesourcesoftware.com/blog-whitesource/open-source-licenses-trends-and-predictions>

Website – Licenses – an overview of common open source licenses. Accessed April 9, 2021. <https://choosealicense.com/licenses/>

Website – PHUSE Whitepaper Scripts. Accessed April 9, 2021. <https://github.com/phuse-org/phuse-scripts/tree/master/whitepapers/WPCT>

Website – FDA Jumpstart Scripts. Accessed April 9, 2021. <https://github.com/phuse-org/phuse-scripts/tree/master/tested/SAS>

Website – SAS Blogs – Data Visualizations. Accessed April 9, 2021. <https://blogs.sas.com/content/topic/data-visualization/>

Website – RhoInc Plots. Accessed April 9, 2021. <https://github.com/RhoInc?language=sas>

Website – SASjs core macros. Accessed April 9, 2021. <https://github.com/sasjs/core>

Website – SMILE – Smart SAS Macros. Accessed April 9, 2021. <https://github.com/KatjaGlassConsulting/SMILE-SmartSASMacros>

Website – Scott Bass SAS Macros. Accessed April 9, 2021. <https://github.com/scottbass/SAS/tree/master/Macro>

Website – Chris SAS Macros. Accessed April 9, 2021. <https://github.com/chris-swenson/sasmacros>

Website – Rolands SAS Macros. Accessed April 9, 2021. <http://www.datasavantconsulting.com/roland/Spectre/download.html>

Website – SASUnit. Accessed April 9, 2021. <https://sourceforge.net/projects/sasunit/>

Website – Reindeer – Render SAS Results into Word. Accessed April 9, 2021. <https://github.com/KatjaGlassConsulting/reindeer>

Website – StatTag. Accessed April 9, 2021. <https://github.com/StatTag/StatTag>

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Katja Glaß
Katja Glass Consulting
info@glacon.eu
www.glacon.eu
www.glacon.eu/portal