Public



PharmaForest:

Building a Collaborative Open-Source SAS Package Ecosystem for Standardized Clinical Trial Reporting and Analytics

PharmaSUG Tokyo 2026 [2026-10-29]

Ryo Nakaya (Takeda Pharmaceutical Company Limited)
Yutaka Morioka (EPS Corporation)
Sharad Chhetri (Takeda Pharmaceuticals U.S.A., Inc.)
Hiroki Yamanobe (DAIICHI SANKYO COMPANY, LIMITED)

Public

Member (Company)

Ryo Nakaya – TAKEDA JAPAN



Yutaka Morioka – EPS



Sharad Chhetri– TAKEDA US



Hiroki Yamanobe – DAIICHI SANKYO

Speaker



Public



Disclaimer

- This presentation is intended to provide an overview of activities undertaken by PharmaForest. Any views or opinions expressed in this presentation are solely those of the individual presenters and do not necessarily reflect the official views, policies, or positions of PharmaForest or any other organization or group with which the presenters are affiliated.

Public

A Changing Clinical Programming Landscape

Open-source momentum

- Open-source R is gaining momentum
  - Reusable packages are increasingly used in clinical programming
  - Projects such as pharmaverse demonstrate the value of community-driven development
  - Shared code enables faster innovation, standardization, and reuse

Current reality

- SAS remains central to regulatory submission workflows
  - SAS is deeply embedded in established clinical development processes
  - Many organizations have substantial SAS codebases and expertise
  - Submission-ready workflows still rely heavily on SAS

Can we bring the benefits of open-source collaboration to the SAS ecosystem?

Public

The Challenge: Reusable SAS Assets Are Fragmented

- Reusable SAS Assets Are Still Fragmented
  - Clinical programmers often solve similar problems repeatedly —but reusable SAS tools are not always easy to share across projects and organizations.

1. Repeated Development

- Similar macros and utilities are developed independently across projects and companies.



2. Inconsistent Quality Processes

- Testing, documentation, and code review practices vary between organizations.



3. Limited Cross-company Reuse

- Useful SAS code often remains within individual teams or companies, limiting broader collaboration.



The opportunity is not to replace SAS
— but to improve how *SAS tools* are developed, shared, and maintained.

画像利用:
いらすとや - 車輪の再発明のイラスト https://www.irasutoya.com/2018/10/blog-post_58.html
いらすとや - 検品のイラスト(男性) https://www.irasutoya.com/2016/10/blog-post_532.html
いらすとや - 風通しが良い職場のイラスト https://www.irasutoya.com/2017/12/blog-post_785.html

Public

PharmaForest

- What's PharmaForest ?



- What is a SAS Package?
  - A reusable, self-contained bundle for sharing SAS code and resources
    - SAS code, macros, formats, functions, data, and documentation
    - Standardized structure based on SPF
    - Easy to install, load, share, and reuse
    - Designed for portability across projects and environments

To Use [OncoPlotter] package within PharmaForest, in an Environment with SPF Installed:

```
%installPackage(OncoPlotter, mirror=PharmaForest)
%loadPackage(OncoPlotter)
```

SPF: SAS Packages Framework

Public

SPF

- What is SPF?
  - SAS Packages Framework (SPF) provides a standardized way to create, install, load, and use SAS Packages.
    - SPF was developed by Bartosz Jablonski.
    - Defines a common structure for SAS Packages
    - Provides tools to install and load packages
    - Makes packages easier to share and reuse across projects

SAS Packages Framework
https://github.com/yabwon/SAS_PACKAGES

SPF setup requires only this much code

```
filename packages "C:¥Temp¥SAS_PACKAGES¥packages";
filename SPFinit url "https://raw.githubusercontent.com/yabwon/SAS_PACKAGES/main/SPF/SPFinit.sas";
%include SPFinit;
```

Public

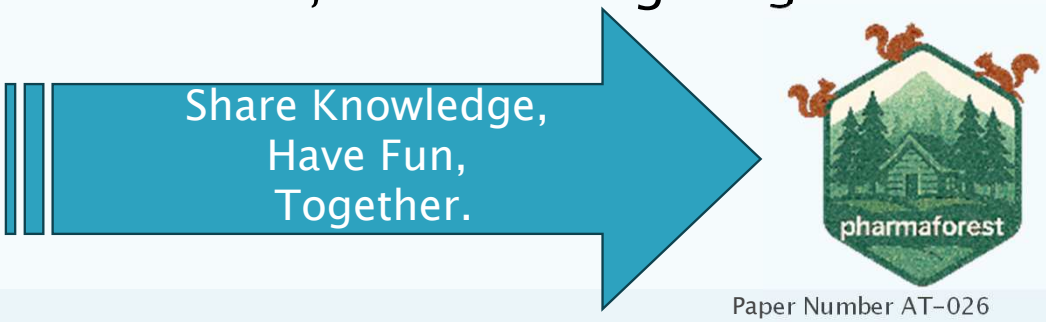
PharmaForest

- Goal

  Through SAS Packages, we want to actively encourage sharing of SAS know-how that has often stayed within individuals. By doing this, we aim to build up collective knowledge, boost productivity, ensure quality through standardization, and energize our community.

- Priority

  Our priority is to share openly—and get others to share as well—so that more people can join in. On that basis, we'll work on improving quality, driving standardization, and creating long-term value.

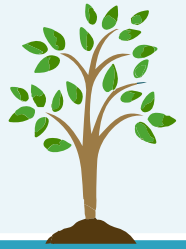


Public



Our Journey






2024: Learning Stage	2025: Launch Stage	2026: Growth Stage	2027: Expansion Stage	Future Vision
• Know and learn SAS packages and its framework • Imagine what can be done • 10th Osaka-SASBen • Of course, so many presentations by Bart as foundation of the initiative by then	• Start small to brainstorm idea and create • Formed sub WG in PHUSE Japan OST WG • Initiated PharmaForest in GitHub • 11th Osaka-SASBen • SAS Innovate Tokyo • SAS user conference Japan • PHUSE SDE Japan	• Let more people know about our idea and vision • Strategic planning and governance for sustainability • Hands-on workshop in Osaka • Approval Phuse WG • Phuse US Connect • PharmaSUG U.S. • KDSS 10th • SAS Innovate Tokyo • Today [PharmaSUG Tokyo]	• Expanding real-world usecases in clinical programming • More collaboration with people, companies, and even regulatory	• **Key part of SAS community's sharing ecosystem (enhancing collaboration)** • Supplementary but essential functions for users (less effort in programming)

Open-source expansion and AI/ML evolution is transforming our way of programming in clinical development and lead us somewhere.
The initiative aims to lead the discussion by shaping ideas into forms, and envision/pursue future state.

Public

SAS Packages Supporting the End-to-End Clinical Data Workflow



Osaka SAS Benkyo-kai 12th
https://sites.google.com/view/osakasasbenkyokai/%E3%83%9B%E3%83%BC%E3%83%A0

Public

Examples of Ongoing Development

- PharmaForest is developing reusable SAS components for common clinical programming tasks.

adamski package

- Adamski is a SAS package inspired by the R package {admiral}. It aims to bring the same flexible and modular ADaM derivation framework to the SAS environment. The package follows the {admiral} design principles while adapting to SAS syntax and workflows. It enables consistent, reproducible ADaM dataset creation in compliance with CDISC standards. Adamski serves as a bridge between open-source R implementations and traditional SAS programming.

- **ADaM Derivation Utilities**
  - Support frequently used ADaM derivation patterns
  - Promote consistency across analyses and studies
  - Encapsulate derivation logic into reusable components
  - Improve transparency through documentation and examples

Collaborators develop individual functions, which are integrated into a community-built package.



admiral
https://pharmaverse.github.io/admiral/cran-release/

Start with common programming problems, then turn proven solutions into reusable packages.

Public

Examples – adamski

derive_vars_dy

- derive_vars_dy() derives study day variables (e.g., ADY) from dates using a reference date such as TRTSDT.

```
# admiral
derive_vars_dy(
    datain,
    reference_date = TRTSDTM,
    source_vars = exprs(TRTSDTM, ASTDTM, AENDT)
)
```

引用: https://pharmaverse.github.io/admiral/cran-release/reference/derive_vars_dy.html

```
** adamski;
data ADAE;
    merge SDTM.AE ADSL;
    by USUBJID;
    %derive_vars_dy(
            reference_date = TRTSDT,
            source_vars = AESTDT AEENDT );
run;
```

引用:https://github.com/PharmaForest/adamski

derive_var_base

- derive_var_base() derives a baseline variable (e.g., BASE) from the record flagged as baseline within each group.

```
# admiral
derive_var_base(
    dataset,
    by_vars,
    source_var = AVAL,
    new_var = BASE,
    filter = ABLFL == "Y"
)
```

引用: https://pharmaverse.github.io/admiral/cran-release/reference/derive_var_base.html

```
**Derive BASE from AVAL;
%derive_var_base(
    dataset=adlb2,
    by_vars=USUBJID PARAMCD,
    source_var=AVAL,
    new_var=BASE);
```

引用:https://github.com/PharmaForest/adamski

Public

Package Validation

Packages are validated using the PharmaForest “valivali” validation toolbox.

- Valivali
  - Reusable assertion macros for SAS package testing
  - Dataset comparison and consistency checks
  - Automated generation of validation reports
  - Designed to support reproducible package verification



- Valivali loads {sasjscore} package developed by Allan Bowe when valivali is loaded and strongly influenced and powered by sasjscore. You need to install {sasjscore} to use the package.

Public

CDISC ARS-Related Packages

- Several SAS packages have been developed to support the CDISC Analysis Results Standard(ARS):
  - sARDen
  - sARDenX
  - sARDenMTS
- One of these packages has received COSA recognition, and the ARS-related package ecosystem is expected to continue evolving.

sARDen(SAS Analysis Results Display engine) is a SAS toolkit for producing CDISC ARS–aligned ARD. Inspired by the R pharmaverse cards schema, it respectfully mirrors ARD conventions for cross-language workflows.

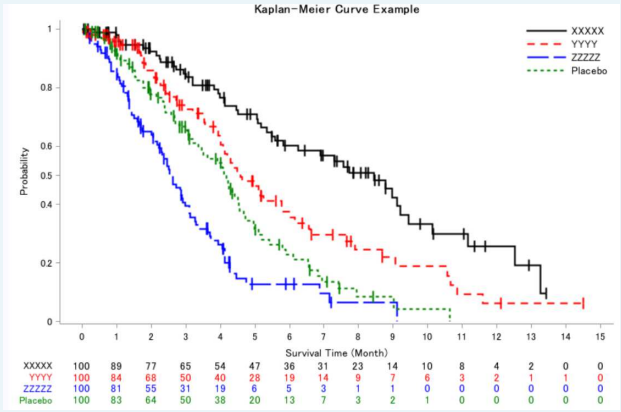
In particular, sARDenX is planned for further enhancement and expansion.

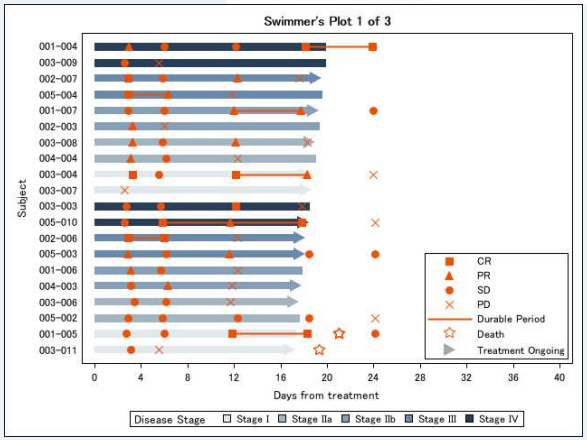
Public

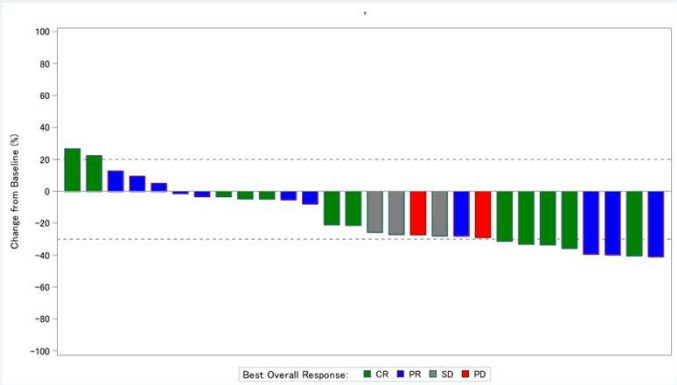
Plot for Oncology

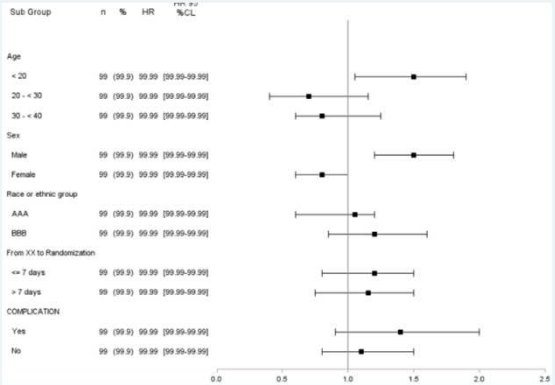
A SAS package for creating commonly used figures in oncology studies

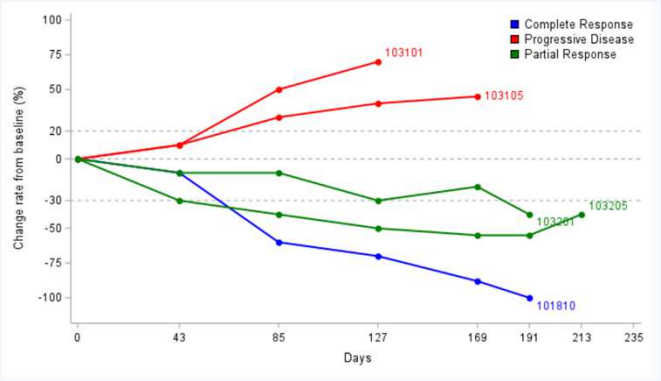
- OncoPlotter provides reusable SAS macros for generating key figures used in oncology clinical trials.
  - Kaplan-Meier Plot
  - Swimmer Plot
  - Waterfall Plot
  - Forest Plot
  - Spider Plot











Public

A Reporting Package Example

- **Reporting & Visualization**

- **Reporting with SAS Packages**
  - Generate RTF tables from analysis or listing datasets
  - Reusable and parameterized reporting logic
  - Version-controlled for consistent use across projects

- **Looking Ahead**
  - Visualization packages supporting ARD are also planned

```
%rtfCreator(DS=sashelp.class
,COLNUM  =2
,VARLST  =name sex
,JUSTLST =Left Center
,WIDTHLST=300 150
,TBLHEAD=%str(Table Head)
,TBLFOOT=%str(Table Foot)
);
```

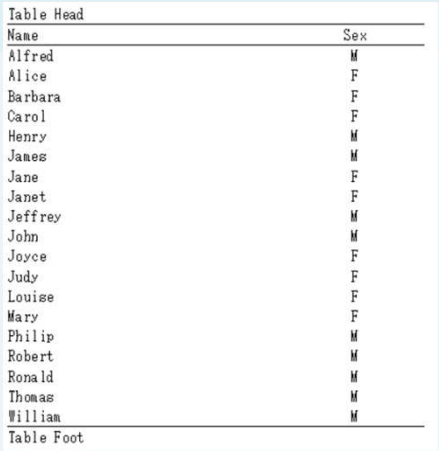


Table Head

Name	Sex
Alfred	M
Alice	F
Barbara	F
Carol	F
Henry	M
James	M
Jane	F
Janet	F
Jeffrey	M
John	M
Joyce	F
Judy	F
Louise	F
Mary	F
Philip	M
Robert	M
Ronald	M
Thomas	M
William	M

Table Foot

Public

PharmaForest Ecosystem Map









Public

Who Benefits?

- PharmaForest is designed for anyone who develops, reviews, or manages SAS-based clinical programming.

Statistical Programmers
- Reuse proven SAS components
- Reduce repetitive programming
- Learn from code developed outside their own organization

Managers & Team Leads
- Improve consistency and maintainability
- Reduce duplicated development effort
- Encourage knowledge sharing across teams



Data Scientists & Developers
- Contribute to modular SAS tooling
- Apply modern open-source development practices
- Collaborate across technical communities

Organizations & the Wider Community
- Share solutions to common industry problems
- Benefit from broader peer review
- Build common tools that no single company needs to maintain alone

PharmaForest turns individual solutions into shared community assets.

Public



Demo1 - adamski programs

```
** Setting [packages folder];
filename packages "C:¥temp¥sas¥packages";

** For the 1st run, install SPFinit;
filename SPFinit url
"https://raw.githubusercontent.com/yabwon/SAS_PACKAGES/main/SPF/SPFinit.sas";
%include  SPFinit;
%installPackage(SPFinit);

** From the 2nd run onward, load SPFinit;
/*%include  packages(SPFinit);*/

** install & load SASPAC;
%installPackage(adamski  , mirror=3);
%installPackage(sas_faker, mirror=3);
%loadPackage(adamski);
%loadPackage(sas_faker);

** create dummy data;
%sas_faker(n_groups=3,
           n_per_group=100,
           output_lib=WORK)
```

```
** merge SDTM_AE amd ADSL;
data DT1;
  merge AE   (in=a obs=10)
        ADSL(in=b)
        ;
  by USUBJID;
  if a and b;
run;

** ASTDT, ASTDY;
data DT2;
  set DT1(keep=TRTSDT AESTDTC);

  format ASTDT yymmdd10.;
  ASTDT=input(AESTDTC,yymmdd10.);

  %derive_vars_dy(
       reference_date = TRTSDT,
       source_vars    = ASTDT
   );
   put _all_;
run;
```

Public

Demo1 – adamski results

```
%derive_vars_dy(
      reference_date = TRTSDT,
      source_vars    = ASTDT)
```

```
AESTDTC=2026-01-26 TRTSDT=2026-01-21 ASTDT=2026-01-26 ASTDY=6 _ERROR_=0 _N_=1
AESTDTC=2026-01-28 TRTSDT=2026-01-21 ASTDT=2026-01-28 ASTDY=8 _ERROR_=0 _N_=2
AESTDTC=2026-01-30 TRTSDT=2026-01-21 ASTDT=2026-01-30 ASTDY=10 _ERROR_=0 _N_=3
AESTDTC=2026-02-08 TRTSDT=2026-01-21 ASTDT=2026-02-08 ASTDY=19 _ERROR_=0 _N_=4
AESTDTC=2026-02-08 TRTSDT=2026-01-21 ASTDT=2026-02-08 ASTDY=19 _ERROR_=0 _N_=5
AESTDTC=2026-02-08 TRTSDT=2026-01-21 ASTDT=2026-02-08 ASTDY=19 _ERROR_=0 _N_=6
AESTDTC=2026-02-12 TRTSDT=2026-01-21 ASTDT=2026-02-12 ASTDY=23 _ERROR_=0 _N_=7
AESTDTC=2026-02-15 TRTSDT=2026-01-21 ASTDT=2026-02-15 ASTDY=26 _ERROR_=0 _N_=8
AESTDTC=2026-02-24 TRTSDT=2026-01-21 ASTDT=2026-02-24 ASTDY=35 _ERROR_=0 _N_=9
AESTDTC=2026-02-27 TRTSDT=2026-01-21 ASTDT=2026-02-27 ASTDY=38 _ERROR_=0 _N_=10
NOTE: There were 10 observations read from the data set WORK.DT1.
NOTE: The data set WORK.DT2 has 10 observations and 4 variables.
NOTE: DATA statement used (Total process time):
      real time           0.00 seconds
      cpu time            0.00 seconds
```

Public

Demo2 – sARDen programs

```
** Setting [packages folder];
filename packages "C:¥temp¥sas¥packages";

** For the 1st run, install SPFinit;
filename SPFinit url
"https://raw.githubusercontent.com/yabwon/SAS_PACKAGES/main/SPF/SPFinit.sas";
%include  SPFinit;
%installPackage(SPFinit);

** From the 2nd run onward, load SPFinit;
/*%include  packages(SPFinit);*/

** install & load SASPAC;
%installPackage(aARDen, mirror=3);
%installPackage(sas_faker, mirror=3);
%loadPackage(aARDen);
%loadPackage(sas_faker);

** create dummy data;
%sas_faker(n_groups=3,
           n_per_group=100,
           output_lib=WORK)
```

```
** summary;
  %sard_summary(
    data=ADSL,
    by=TRT01P,
    variable=AGE WEIGHTBL,
    statistic=N MEDIAN MIN MAX MEAN SD,
    out=sard_summary_mean
  );
```

Public

Demo2 – sARDen results

VIEWTABLE: Work.Sard_summary_mean

	group1	group1_level	variable	context	stat_name	stat_label	stat	fmt_fun
1	TRT01P	Group1	AGE	summary	n	N	100	0
2	TRT01P	Group1	AGE	summary	median	Median	51.5	1
3	TRT01P	Group1	AGE	summary	min	Min	20	0
4	TRT01P	Group1	AGE	summary	max	Max	80	0
5	TRT01P	Group1	AGE	summary	mean	Mean	50.38	1
6	TRT01P	Group1	AGE	summary	sd	SD	18.3456047	1
7	TRT01P	Group1	WEIGHTBL	summary	n	N	100	0
8	TRT01P	Group1	WEIGHTBL	summary	median	Median	67.65	2
9	TRT01P	Group1	WEIGHTBL	summary	min	Min	37.7	1
10	TRT01P	Group1	WEIGHTBL	summary	max	Max	113.3	1
11	TRT01P	Group1	WEIGHTBL	summary	mean	Mean	68.916	2
12	TRT01P	Group1	WEIGHTBL	summary	sd	SD	16.5997975	2
13	TRT01P	Group2	AGE	summary	n	N	100	0

```
** summary;
  %sard_summary(
    data=ADSL,
    by=TRT01P,
    variable=AGE WEIGHTBL,
    statistic=N MEDIAN MIN MAX MEAN SD,
    out=sard_summary_mean
  );
```

Public

Reference

Reference

- Mancini E, Bundfuss S, Bearham M, et al. *admiral: ADaM in R Asset Library*. R package version 1.4.2. 2026.[accessed: 2026-09-14] https://pharmaverse.github.io/admiral/
- Yutaka Morioka.「PharmaForestについて」Osaka SAS Benkyo-kai 12th. https://sites.google.com/view/osakasasbenkyokai/%E3%83%9B%E3%83%BC%E3%83%A0
- PharmaForst. *Pharmaforest* GitHub. [accessed: 2026-09-14] https://github.com/PharmaForest/
- PharmaForest. *valivali*. GitHub. [accessed: 2026-09-14] https://github.com/PharmaForest/valivali
- PharmaForest. *adamski*. GitHub. [accessed: 2026-09-14] https://github.com/PharmaForest/adamski
- PharmaForest. *sARDen*. GitHub. [accessed: 2026-09-14] https://github.com/PharmaForest/sARDen/
- PharmaForest. *rtfCreator*. GitHub. [accessed: 2026-09-14] https://github.com/PharmaForest/rtfcreator/
- PharmaForest. *OncoPlotter*. GitHub. [accessed: 2026-09-14] https://github.com/PharmaForest/OncoPlotter/

image

- いらすとや. 車輪の再発明のイラスト [accessed: 2026-09-14] https://www.irasutoya.com/2018/10/blog-post_58.html
- いらすとや. 検品のイラスト（男性） [accessed: 2026-09-14] https://www.irasutoya.com/2016/10/blog-post_532.html
- いらすとや. 風通しが良い職場のイラスト [accessed: 2026-09-14] https://www.irasutoya.com/2017/12/blog-post_785.html

Public

Thank you for attention



Thank you for Your Attention

Contact: Hiroki Yamanobe (mail: hiroki.yamanobe@daiichisankyo.com)