

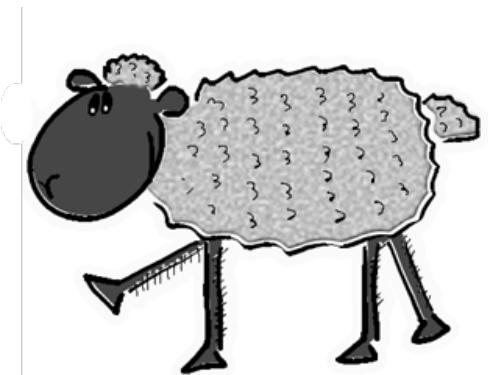


SASPAC

- SAS packageの作り方と事例紹介 -

仲家 諒 鳴澤 輝子

Takeda Pharmaceutical Company
Statistical & Quantitative Sciences Japan
2025 September 25



developer



Agenda



- 前半（作り方）
 - SAS packageとは
 - SAS packageの作成方法
 - SASPACer、SASPACerShiny
- 後半（事例紹介）
 - RWD Expressの紹介
 - SASインデックスで高速抽出
 - マクロ化→SASPACへ
 - 活用例① ②
 - SASPACやってみよう

SAS Packageとは？

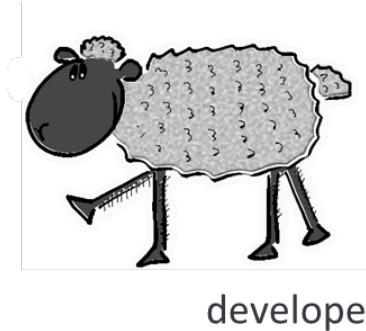
- SAS Packages Framework(SPF)を用いて、SASマクロ、関数、データなどを統一した様式でパッケージ化したもの
https://github.com/yabwon/SAS_PACKAGES
- Rのようにパッケージを作成・共有利用できる
- SASPAC (SAS Packages Archive) レポジトリ
<https://github.com/SASPAC>

SAS packagesはBartosz Jablonski氏*が2019年に開発し、近年利用者が増加している

*Takeda pharmaceutical, Warsaw University of Technology, yabwon



SAS Packageの作成方法



1. SPFの読み込み（セットアップ）
→ Back upスライド参照（3行実行のみ！）

1.5. ChatGPTなどで六角形ロゴを作る！！

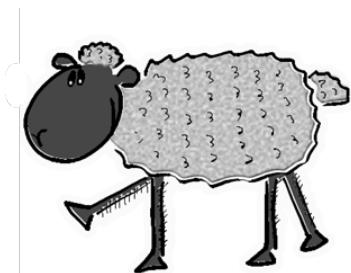
2. パッケージ用のソースファイル・フォルダの作成
→ 以降のスライドにて説明
3. %generatePackage()マクロの実行

```
%generatePackage(  
    filesLocation=C:\your\packages\folder, /*上記で作成したファイル・フォルダの場所*/  
    markdownDoc=1, /*パッケージ情報、マクロや関数などの情報のmarkdownファイル作成有無*/  
    easyArch=1           /*.zipと.mdのコピーファイル（名前にversion付き）の作成有無*/  
)
```

結果として、[packagename].zipファイル
(パッケージファイル) が作成される

その他、パッケージ作成時のテスト
などに関するパラメータあり

パッケージ用のソースファイル・フォルダ



- 標準的な構成

- 01_libname
- 02_formats
- 03_functions
- 04_data
- 05_lazydata
- 06_macro
- 07_test
- addcnt
- description
- license

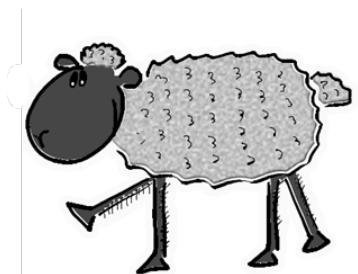
- 連番を付ける（パッケージ作成時の実行順のため）
- オブジェクトタイプ（macroなど）を後ろにつける
- Testはパッケージ作成時のテスト用コード
- 数字、 "_"、 小文字のみ
- .sasファイル以外は無視される（addcnt以外）
- Addcnt（Additional contents）はパッケージにそのまま含まれる
- description.sasは必須ファイル、 license.sasは作ったほうがよい（とのページで説明）

https://github.com/yabwon/SAS_PACKAGES/blob/628641a6cfb6c5cf900283e9db450c7fe0e386e9/SPF/Documentation/Paper_1079-2021/My%20First%20SAS%20Package%20-%20a%20How%20To.pdf

https://github.com/yabwon/SAS_PACKAGES/blob/main/SPF/Documentation/HelloWorldPackage.md

パッケージ用のソースファイル・フォルダ

- 各フォルダ内のsasファイルの中身の例



developer

- 01_libname
- 02_formats
- 03_functions
- 04_data
- 05_lazydata
- 06_macro
- 07_test
- addcnt
- description**
- license**

```
Type : Package
Package : testPackage
Title : Test package
Version : 0.0.1
Author : Taro Sasu(taro.sasu@mail.com)
Maintainer : Taro Sasu(taro.sasu@mail.com)
License : MIT
Encoding : UTF8
Required : "Base SAS Software"
ReqPackages : "List any SAS packages required here"

DESCRIPTION START:

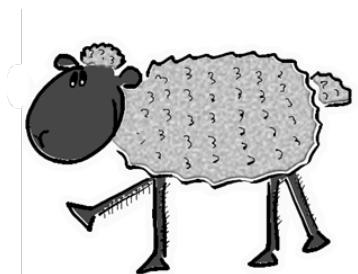
## The testPackage ##
The test package is to show how the package is created
You can describe package information here

DESCRIPTION END:
```

Typeなど、この形式でパッケージ情報を埋めていく

パッケージ用のソースファイル・フォルダ

- 各フォルダ内のsasファイルの中身の例



developer

- 01_libname
- 02_formats
- 03_functions
- 04_data
- 05_lazydata
- 06_macro
- 07_test
- addcnt
- description
- license

Copyright (c) [YEAR] [Owner Name]

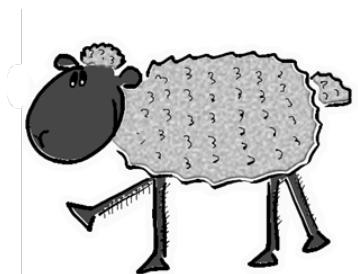
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

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

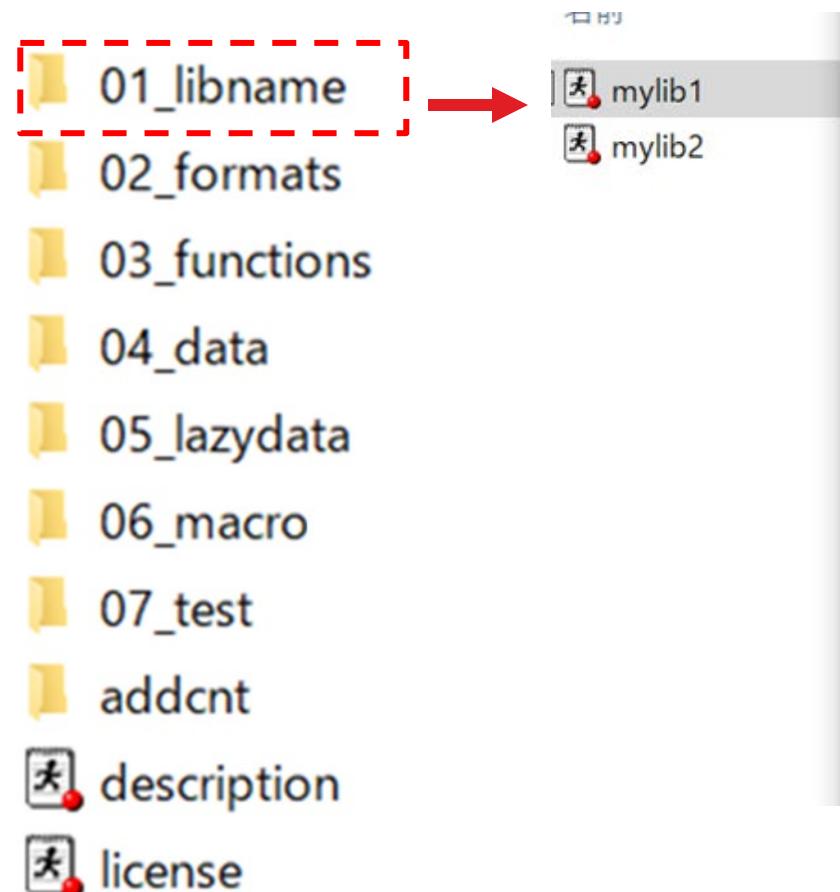
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MITライセンスの標準文
(license.sasファイルがない場合は
MITの標準文が自動設定される)

パッケージ用のソースファイル・フォルダ



- 各フォルダ内のsasファイルの中身の例



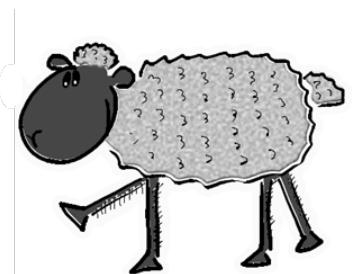
```
1 *** HELP START ***
2
3 Create mylib library under work directory.
4
5 /*** HELP END ***/
6
7 data _null_;
8 length rc0 $ 32767 rc1 rc2 8;
9 lib = "myLib1";
10 rc0 = DCREATE(lib, "%sysfunc(pathname(%work%))");
11 put rc0 =;
12 rc1 = LIBNAME(lib, "%sysfunc(pathname(%work%))" !! lib, "BASE");
13 rc2 = LIBREF(lib);
14 if rc2 NE 0 then rc1 = LIBNAME(lib, "%sysfunc(pathname(%work%))", "BASE");
15 run;
16
17 libname myLib1 LIST;
```

ヘルプ情報記載欄

ファイル名 = オブジェクト名
(1ファイル : 1オブジェクト)
(libname以外も原則として同様)

パッケージ用のソースファイル・フォルダ

- 各フォルダ内のsasファイルの中身の例



developer

A screenshot of a file explorer window. On the left, there's a tree-view navigation pane with the following folder structure:

- 01_libname
- 02_formats
- 03_functions
- 04_data
- 05_lazydata
- 06_macro
- 07_test
- addcnt
- description
- license

The "02_formats" folder is highlighted with a red dashed border and has a red arrow pointing to it from the text on the right. Inside the "02_formats" folder, there are two files: "fmtnum" and "infmtnum".

```
1  /*** HELP START ***/
2
3 This is a format.
4 Proc format ; and run ; are not needed within SASPAC framework.
5
6 /*** HELP END ***/
7
8   value fmtNum
9     low -< 0 = "negative"
10    0 = "zero"
11    0 <- high = "positive"
12    other = "missing"
13 .
```

ヘルプ情報記載欄

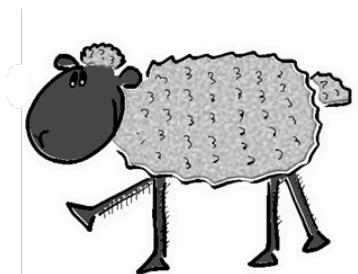
Proc format ; と run ; の記載不可
(中身のみ記載)

Formatsについては、1ファイル : 1オブジェクトでなくてよい

Note: format(sなし)というフォルダ構成も可能だが、formatフォルダは歴史的に先に作られたもので、proc format lib=work.&packageName.format ;とrun ;を記載する必要がある。（普段はformatsフォルダのほうが使い勝手がよい）

パッケージ用のソースファイル・フォルダ

- 各フォルダ内のsasファイルの中身の例



developer

01_libname

02_formats

03_functions

04_data

05_lazydata

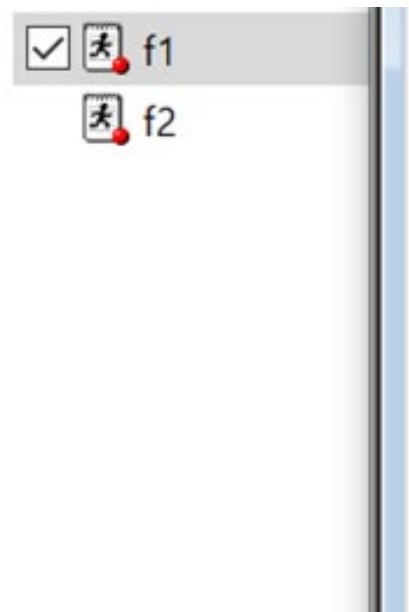
06_macro

07_test

addcnt

description

license



```
1  /*** HELP START ***/
2
3  F1 is an user-defined function to output +1 value.
4  Proc fcmp outlib=work.f.p ; and run ; are not needed in
5  options cmplib=work.f ; is required prior to f1 is used.
6
7  /*** HELP END ***/
8
9  function F1(n);
10    return (n+1);
11  endsub;
```

ヘルプ情報記載欄

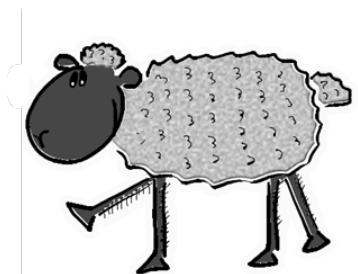
Proc fcmp ; と run ; の記載不可
(中身のみ記載)

Functionsについては、1ファイル：
1オブジェクトでなくてよい

Note: function(sなし)というフォルダ構成も可能だが、functionフォルダは歴史的に先に作られたもので、proc fcmp outlib=work.&packageName.fcmp.package;とrun ;を記載する必要がある。（普段はfunctionssフォルダのほうが使い勝手がよい）

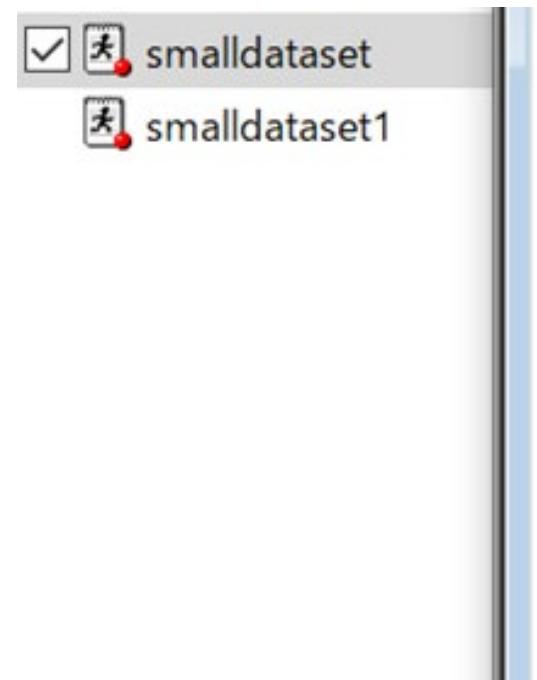
パッケージ用のソースファイル・フォルダ

- 各フォルダ内のsasファイルの中身の例



developer

01_libname
02_formats
03_functions
04_data
05_lazydata
06_macro
07_test
addcnt
description
license



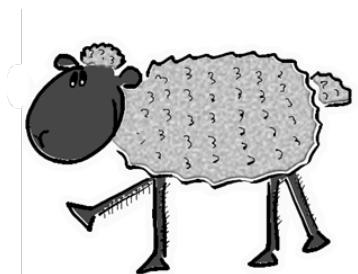
```
1 *** HELP START ***
2
3 This is a small dataset.
4
5 /***** HELP END ***
6
7 data myLib1.smallDataset;
8   do n = , -1, 0, 1;
9     m = put(n, fmtNum.);
10    output;
11  end;
12 run;
```

ヘルプ情報記載欄

01_libnameで定義したmylib1ライブラリと02_formatsで定義したfmtNumフォーマットを使用している（そのためフォルダに連番を振っている）

パッケージ用のソースファイル・フォルダ

- 各フォルダ内のsasファイルの中身の例



developer

01_libname
02_formats
03_functions
04_data
05_lazydata
06_macro
07_test
addcnt
description
license

biggerdataset
 biggerdataset1

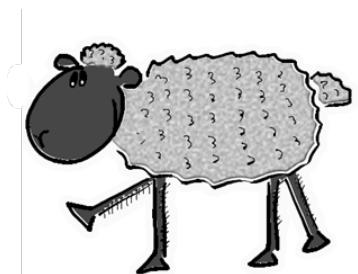
```
1 /*** HELP START ***/  
2  
3 This is a bigger dataset.  
4  
5 /*** HELP END ***/  
6  
7 data myLib1.biggerDataset;  
8   do i = , -1e6 to 1e6;  
9     j = put(i, fmtNum.);  
10    k = ranuni(17);  
11    output;  
12  end;  
13 run;
```

ヘルプ情報記載欄

%loadPackage()でlazyData=で
指定しないとロードされない
仕様になっている

パッケージ用のソースファイル・フォルダ

- 各フォルダ内のsasファイルの中身の例



developer

- 01_libname
- 02_formats
- 03_functions
- 04_data
- 05_lazydata
- 06_macro
- 07_test
- addcnt
- description
- license

The screenshot shows the SAS Studio interface. On the left, there's a file browser with several folders listed. The folder '06_macro' is highlighted with a red dashed border and has a red arrow pointing to it from the left. In the center, there's a code editor window displaying SAS code. The code defines a macro named 'mcrOne'. It starts with a help comment, followed by a description, and then the macro definition itself which includes setting a dataset, calling two functions, and outputting the results. The code editor has syntax highlighting for SAS keywords and comments.

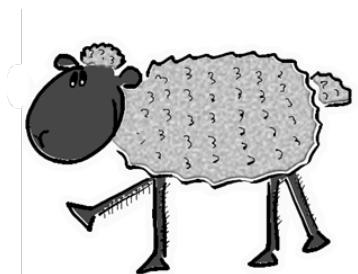
```
1 /**** HELP START ***//*/
2
3 This is mcrOne macro.
4
5 /*/*/* HELP END */
6
7 %macro mcrOne();
8   %put **Hi! This is macro &sysmacroname.**;
9   data _null_;
10   set myLib1.smallDataset;
11   p = f1(n);
12   p + f2(n);
13   put (n p) (= fmtNum.);
14   run;
15 %mend mcrOne;
16
```

ヘルプ情報記載欄

マクロはパッケージの主役!!

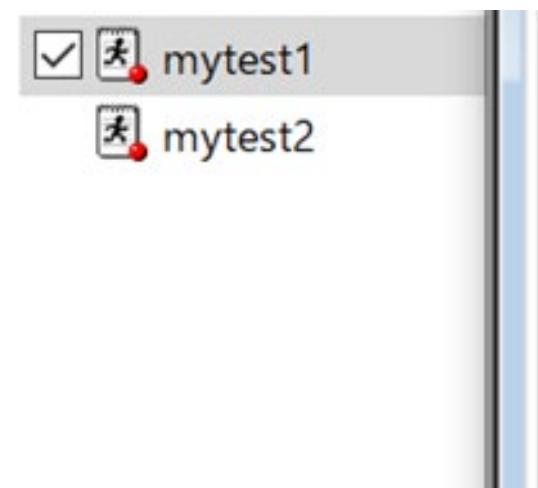
パッケージ用のソースファイル・フォルダ

- 各フォルダ内のsasファイルの中身の例



developer

- 01_libname
- 02_formats
- 03_functions
- 04_data
- 05_lazydata
- 06_macro
- 07_test
- addcnt
- description
- license

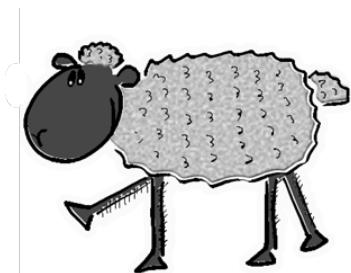


```
1 /* *** HELP START *** */
2
3 This is a test.
4
5 *///*** HELP END ***
6
7 %mcrTwo(m=mcrOne)
8
```

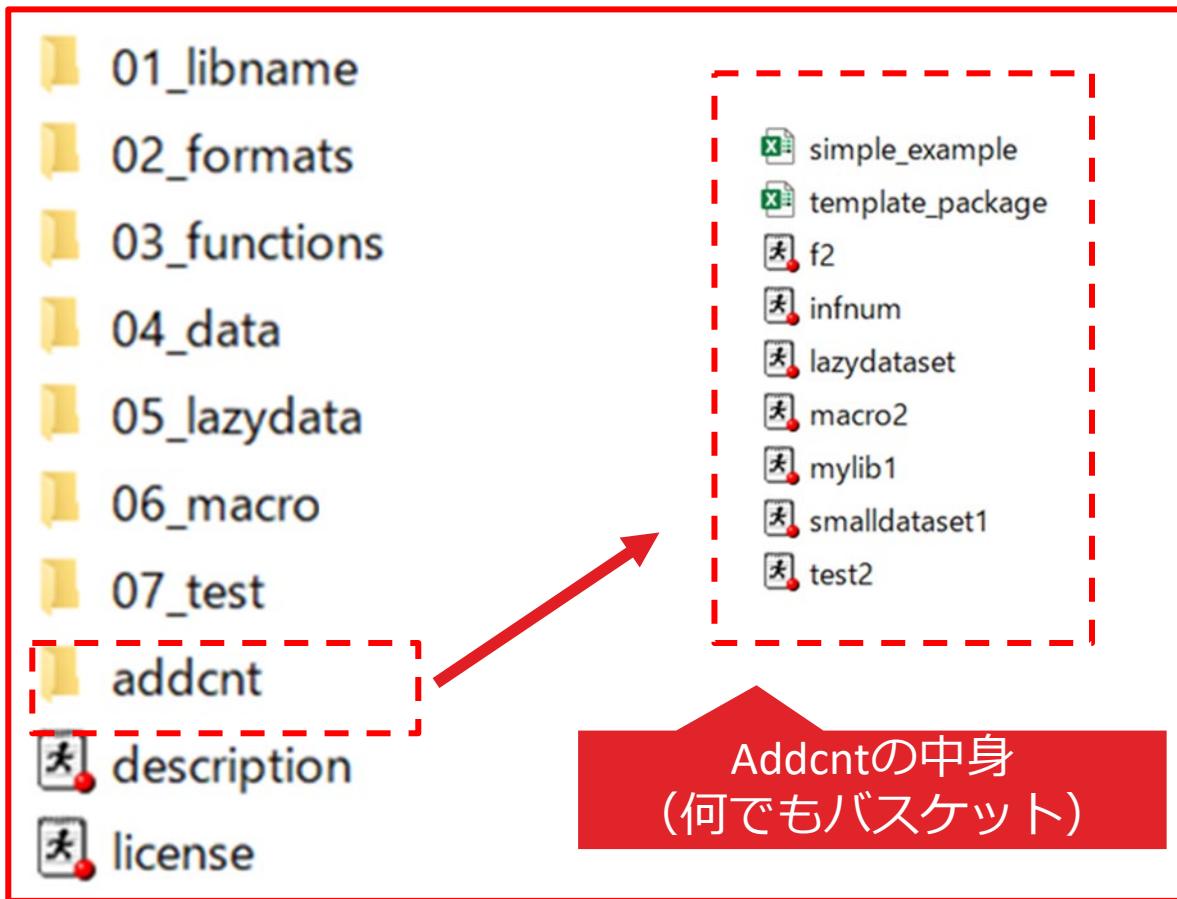
ヘルプ情報記載欄

%generatePackage()でtestPackage=Y
(デフォルト) とするとパッケージ
作成時にテスト実行

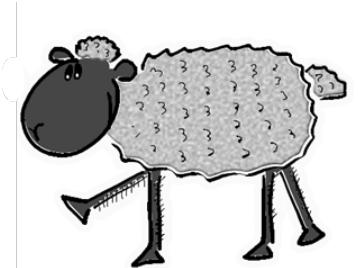
パッケージ用のソースファイル・フォルダ



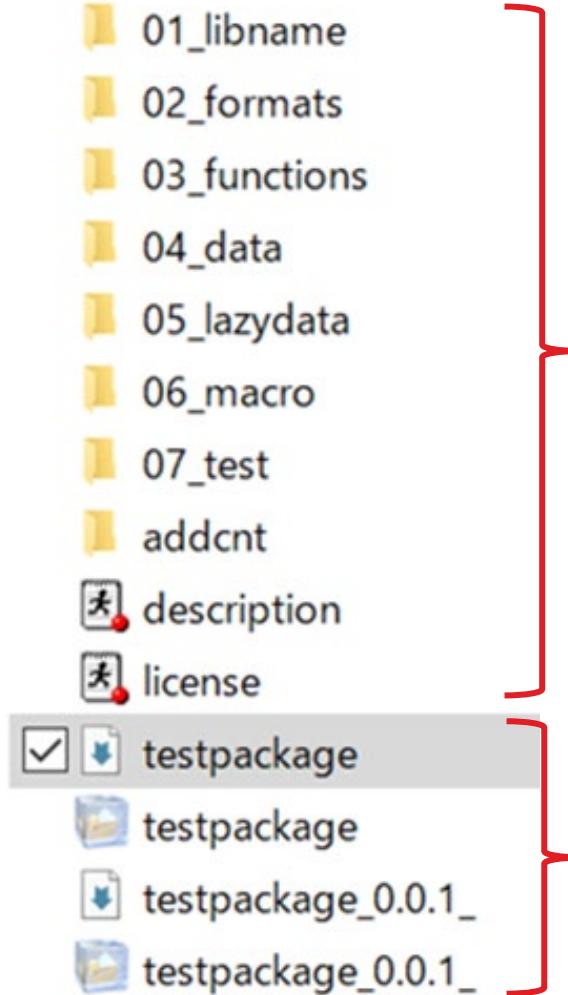
- Additional Contents(Addcnt)



%generatePackage()のアウトプット



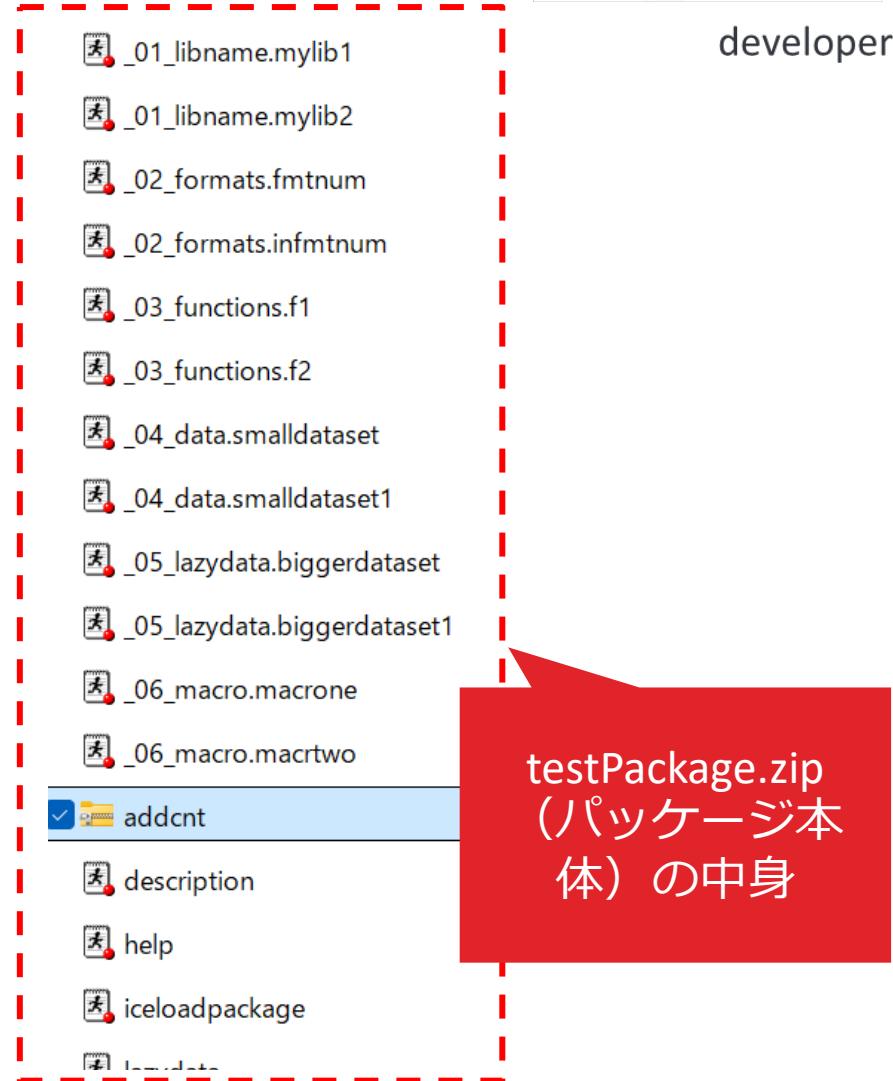
developer



ソースフォルダ・ファイル
(%generatePackage()に入る入力データ)

testPackage.md
(descriptionやヘルプ情報を基
にしたマークダウン文書)

パッケージ本体 (.zip) 、 説明書 (.md) 、
アーカイブ用コピー (_version)
(%generatePackage()の出力データ)



testPackage.zip
(パッケージ本
体) の中身

SASPACer



- **SAS packageをワンパンで作る**ためのパッケージ
(Sampleのエクセル見たら大体わかる仕組み)

<https://github.com/Nakaya-Ryo/SASPACer>

| | |
|-------------|---|
| Type | Package |
| Package | testPackage |
| Title | My first SAS package |
| Version | 0.0.1 |
| Author | John Smith(john.smith@mail.com) |
| Maintainer | John Smith(john.smith@mail.com) |
| License | MIT |
| Encoding | UTF8 |
| Required | "Base SAS Software" |
| ReqPackage | "Baseplus (2.1.0)" |
| Description | ## The myPackage ## The 'myPackage' is my first SAS package. ### References ### 1. Bartosz Jablonski. "My First SAS Package - a How To". SGF |
| | description license 01_libname 02_forma |

| name | help | body | location |
|--------|--|--|---------------------------|
| mcrone | This is mcrOne macro. (No need to write location column if content is written in body column.) | %macro mcrOne(); %put **Hi! This is macro &sysmacroname.**; data _null_; set myLib.smallDataset; p = f1(n); p + f2(n); put (n p) (= fmtNum.); run; %mend mcrOne; | |
| mcrtwo | This is mcrTwo macro. (No need to write body column if content is in a file written in location column, SASPACer reads the file.) | | C:\temp\addcnt\mcrtwo.sas |

%ex2pac(

excel_file=¥folder¥to¥excel_file.xlsx,
package_location=¥folder¥for¥output,
complete_generation=Y

)

エクセルを入力として、ソースフォルダ・
ファイル、パッケージファイルを出力する。
フォルダ・ファイルとか作る必要なし！



SASPACErShiny



- SAS packageをワンツーで作るためのRパッケージ (Shiny application)
<https://github.com/Nakaya-Ryo/SASPACErShiny>



C:\Temp\test - Shiny
http://127.0.0.1:4324 | Open in Browser | C

SASPACEr shiny

Set all in the main(right) panel and run:

Run

New Tab Name:
myTab

Add Tab

Instructions(3 steps)

step1: Set all fields in the right panel. You can add tabs and remove tabs and edit contents(by double click) in each tab.

step2: Run button creates and downloads zipped source folders/files of SAS package.

step3: You can unzip and run %generatePackage() in SAS Packages Framework(SPF) using SAS to generate SAS package file.

description
05_test
license
01_libname
02_formats
03_functions
04_macro
Type:
Package
Package:
myPackage
Title:
My Package
Version:
0.0.1
Author:
Your Name(yourname@mail.com)
Maintainer:
Your Name(yourname@mail.com)

GUIでソースフォルダ・ファイルを設定 & ダウンロードし、%generatePackage()を別途実行

%generatePackage(
filesLocation=¥path¥to¥source¥package¥folder,
markdownDoc=1,
easyArch=1)

SAS Package Lady



- SAS Packageを対話型で作ってくれるおばちゃん
<https://chatgpt.com/g/g-68be12f679a88191866ef1e9b35be3c4-sas-package-lady>
- サポート内容
 - パッケージ情報を聴取してくれる
 - マクロファイルなどをアップロードしたらヘルプ情報をしてくれる
 - ソースパッケージ（フォルダ・ファイル構成）を出力してくれる
 - ロゴ作成も手伝ってくれる
- chatGPTへのサインアップ（無料版OK）が必要

おばちゃんがソースパッケージを作ってやるけん、ダウンロードして%generatePackage()を別途実行しちゃうね



```
%generatePackage(  
  filesLocation=%path%to%source%package%folder,  
  markdownDoc=1,  
  easyArch=1)
```

SAS Packageの共有・利用



SASPAC / PharmaForest on GitHub



<https://github.com/SASPAC>



<https://github.com/PharmaForest>

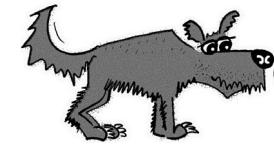
Individual repositories on GitHub



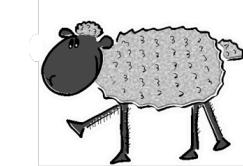
Intranet
(e.g. QMsf36scoring (Takeda's internal))*

Local

あなたは何パック?
(様々な共有レベルが可能)



インストール・使用
(User)



公開・共有
(Developer)



Agenda



- 前半（作り方）
 - SAS packageとは
 - SAS packageの作成方法
 - SASPACer
- 後半（事例紹介）
 - RWD Expressの紹介
 - SASインデックスで高速抽出
 - マクロ化→SASPACへ
 - 活用例① ②
 - SASPACやってみよう



パッケージ紹介 RWD Express

Real World Data 高速化ツール

鳴澤 輝子 (Narusawa Teruko)

仲家 諒 (Nakaya Ryo)

Takeda Pharmaceutical Company

Statistical and Quantitative Science

25 Sep 2025 @ SASユーザ会



- Real World Dataの効率的なハンドリングを可能にするツール

Real World Dataのプログラミングは、実行時間が長くなり非効率になるという課題がある。単純なデータ抽出であれば、SASインデックスを活用することで非常に高速な抽出が可能になることが分かったが、SASインデックスはプログラマにとって馴染みが薄く、使用される機会は限られている。

そのため、大規模データの扱いに不慣れなプログラマでも使用できるよう、インデックス化をマクロの中で実装した

SAS インデックスとは？



- データベース内のレコードを特定のキー変数（インデックス）を使って直接アクセスできるようにする仕組み

簡単に言うと...辞書の索引のようなもの
索引が付いていれば他のページを探しに行く必要
がなく、その分処理が速くなる

- SAS dataset に Indexが付いてるとどうなるの？

- ✓ ソートが不要になる
e.g: Merge、PROC MEANSなどの byステートメント
- ✓ SQLやWhereによる抽出が超高速になる
- ✓ Data/Proc ステップ + Byステートメント
- ✓ Match Merging



- ✓ ソートをするとインデックスが崩れるため、ソートは禁止
- ✓ If 条件文使用時の抽出は遅い
- ✓ 複雑な抽出には弱い
or より in, Substr() index()などを使うと遅いなど

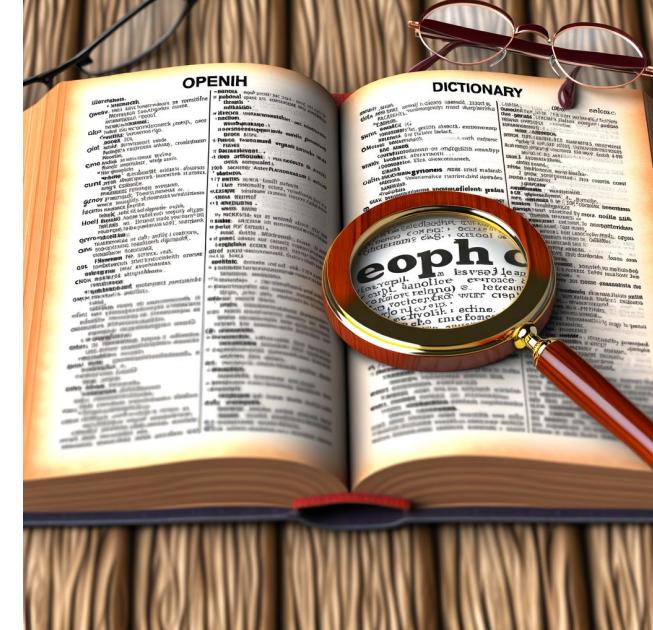
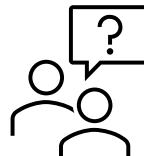
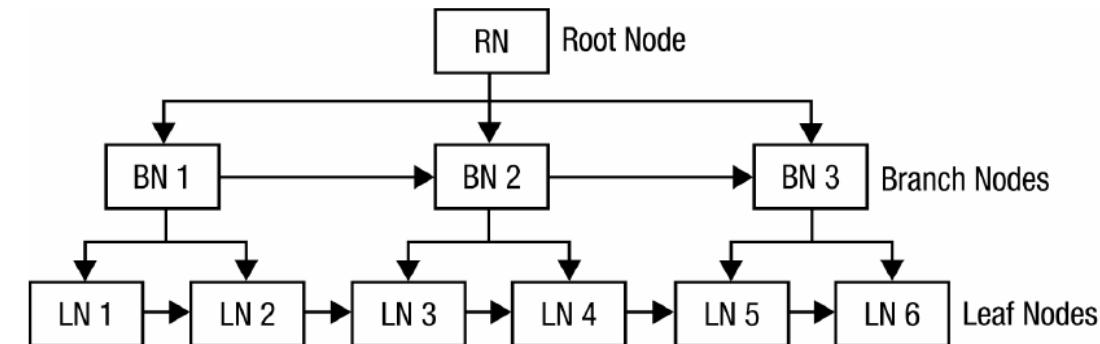


Figure 1.2 The Index Tree Structure

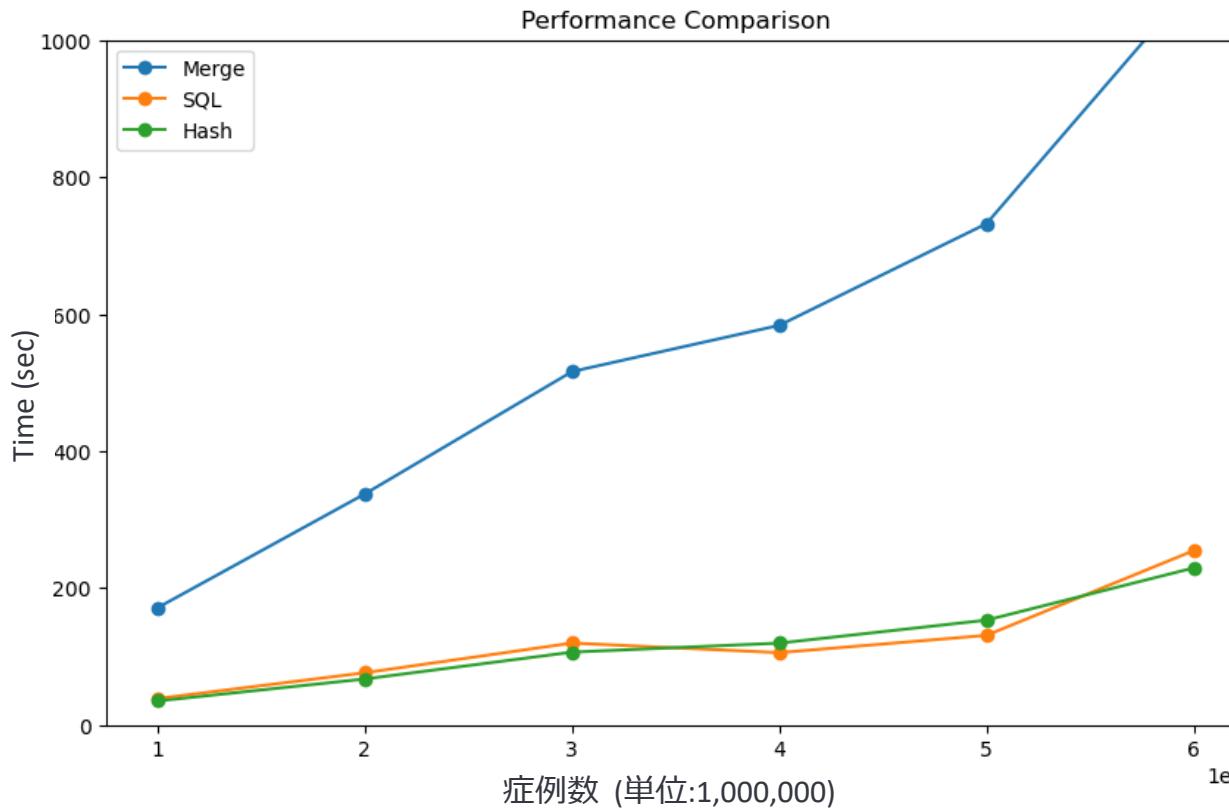


インデックス化による処理時間の違い Merge/Hash/SQL



Before

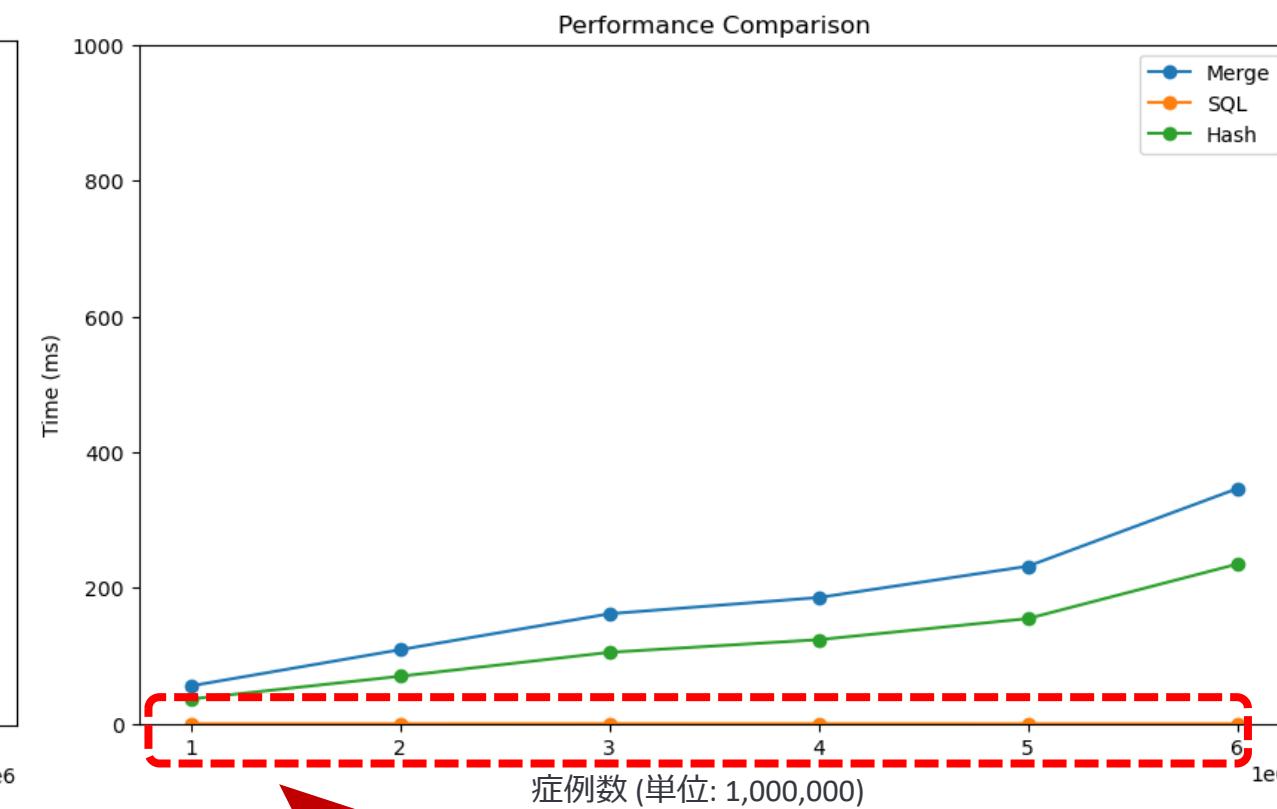
→ After



[条件]

A: 大きなデータセット（症例数 : x軸×500） B: 100症例 (1症例1レコード)

大きなデータセットAから、1症例1レコードのデータセットBに入っている100症例分のレコードを抽出する時間を計測



SQLまたはWhere文による抽出は0.1秒以下！！

RWD Express SASPACまでの道のり...



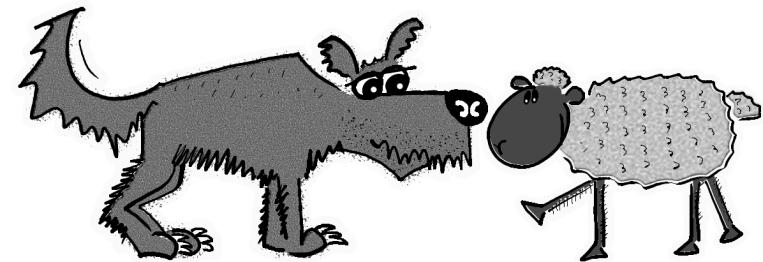
1: Real World Database研究の、解析プログラミング作業にSASインデックスを活用
活用例

- Feasibility Analysis
- 個別症例の抜き取りチェック

2: 汎用的に使えるようのマクロ化

3: タケダ社内のRWDプロジェクトで利用

4: **SASPACer** を使用し、GitHubにてリリース



活用例①：プログラミング用のSmall World Dataを作る



- <https://github.com/Narusawa-T/RWDExpress>

- %index_single_key
- %small_world

1. RWDライブラリのデータセットをまとめてインデックス化（←時間掛かります）

```
%index_single_key(inlib=rwd, outlib=rwdx, indexkey=patid);
```

2. インデックス化されたRWDから、特定の1000患者のみを抽出する

```
%small_world(inlib=rwdx, outlib=swd,  
            subject_level_ds=patient,  
            subject_id_var=patid, no_sub=1000);
```

※ 上記例では、patientデータの最初の1000患者を抽出

切り出されたSmall World Dataは、RWDと同じ構造のため、
これを使用してプログラミングを実施することで、実行時間を短縮できる。



活用例②：データセットを分割する



- <https://github.com/Narusawa-T/RWDExpress>
 - %split_world

100万レコードのactデータを、10万レコード毎に分割する

```
%split_world(inlib= rwd, indata= act, nperBlock=100000);
```

分割データセット act01, act02 ... act10ができる。

分割したデータセットを順番に処理し、最後に1つにまとめる

Block番号の指定も可能※

```
%split_world(inlib= rwd, indata= act, nperBlock=100000  
blockstart=1, blockend=2);
```

※以下のようなケースに使用

とりあえず最初の1つだけ作成したい...

23個目まで出来たところでTimeoutしてしまったので、24個目から...



- SAS Package を作成

- SASPACer を使用することで、初めてのハードルが大きく下がります。
 - エクセルに入力するだけで完了するため、専門的な知識がなくてもPackage作成可能
 - R Shiny版では、GUIからの入力でも実施できるため、より直感的に作業が可能
 - SAS Package Ladyでの、おばちゃんとのパッケージ作りは、もはや楽しい😊

- GitHubにアップロード

アカウントを作成し、SAS Package をアップロード

- 最後に...

皆さんもぜひ、お気に入りのマクロを、1つアップロードしてみてください。

SAS Packageを通じ、SASをより多くの人に親しんでもらい、SASのコミュニティが広がるのはとても嬉しいことです。コミュニティへの第一歩！を踏み出してください。





Back up

SPFのセットアップ

パッケージ用のフォルダを作ったうえで、下記を実行することで準備完了

```
filename packages "¥Your¥Folder¥"; 注) Packagesは固定の名前
filename SPFinit url
"https://raw.githubusercontent.com/yabwon/SAS_PACKAGES/main/SPF/SPFinit.sas";
%include SPFinit;
%installPackage(SPFinit)
-----
| filename packages "¥Your¥Folder¥";
| %include packages(SPFinit);
|
-----
```

一度SPFinit.sasを上記%installPackage(SPFinit)でpackagesの場所にダウンロードすれば、以降は下記実行でOK

https://github.com/yabwon/SAS_PACKAGES/blob/main/SPF/Documentation/HelloWorldPackage.md

インストール、ロードの例

```
%installPackage(baseplus)
%installPackage(SASPACer, sourcePath=https://github.com/Nakaya-Ryo/SASPACer/raw/main/)
%installPackage(RWDExpress(0.0.1), github=Narusawa-T)
```

SASPACレポジトリ上のものはsourcePath=は
不要でシンプルに記載可能

```
%loadPackage(baseplus)
%loadPackage(SASPACer)
```

%loadPackage()はRでのlibrary()のようなイメージ

References



- SAS Packages: The Way to Share (a How To), SAS Global Forum 2020, Bartosz Jablonski
<https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2020/4725-2020.pdf>
- SASPAC(SAS Packages Repository)
<https://github.com/SASPAC>
- Modern graph package “SAS plotter”, SAS User Group Conference Japan 2024, Kosuke Tsutsugo
<https://github.com/SASPAC/sasplotter>
- Integration of SAS GRID environment and SF-36 Health Survey scoring API with SAS Packages, PharmaSUG 2024, Bartosz Jablonski
<https://pharmasug.org/wp-content/uploads/2024/06/PharmaSUG-2024-SD-262.pdf>
- SASPAC -あるGithubアーカイブのご紹介-, 第10回大阪SAS勉強会 2024, Ryo Nakaya,
<https://sites.google.com/view/osakasasbenkyokai/%E7%AC%AC10%E5%9B%9E>
- SASPAC2 -Real World Data 高速化ツールの検討-, 第11回大阪SAS勉強会 2025, Teruko Narusawa,
<https://sites.google.com/view/osakasasbenkyokai/%E7%AC%AC11%E5%9B%9E>
- PharmaForest
<https://github.com/PharmaForest>
- SASPACer
<https://github.com/Nakaya-Ryo/SASPACer>
- SASPACerShiny
<https://github.com/Nakaya-Ryo/SASPACerShiny>
- SAS Package Lady
<https://chatgpt.com/g/g-68be12f679a88191866ef1e9b35be3c4-sas-package-lady>
- RWDExpress
<https://github.com/Narusawa-T/RWDExpress>