

2024年10月18日



業務で使用しているハッシュオブジェクトの紹介

イーピーエス株式会社

リアルワールドエビデンス事業本部

RWE推進センター リアルワールドデータサイエンス部

坂尻 大樹

TEL. 06-7167-5518

目次

- ・はじめに
- ・ハッシュオブジェクトとは
- ・フラグ変数の作成
- ・他のデータセットから変数を取得
- ・業務実用例
- ・応用編：多対多の処理
- ・参考資料

はじめに

SAS9.1よりハッシュオブジェクト(メモリに展開されるデータ構造体)機能がBase SASで利用可能となった

ハッシュオブジェクトの構文は、見慣れない書き方…

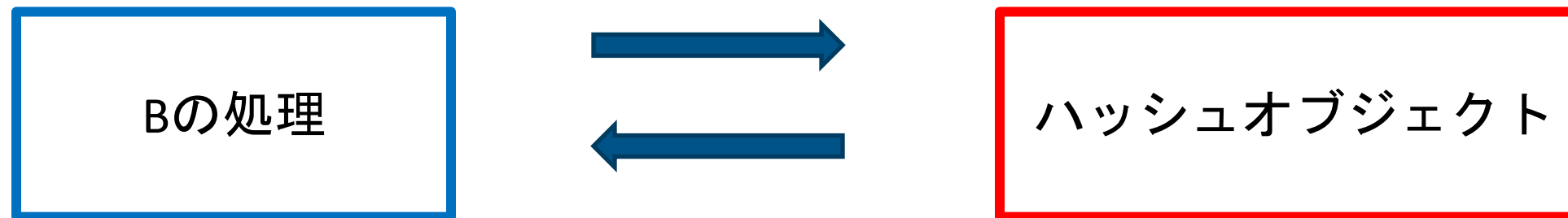
→発表者も初めて見た時、面を食らった

→勧められたマクロが便利だったため、勉強し始めた

ハッシュオブジェクトとは

データステップ中は、基本的にSET(MERGE)したデータセットについて処理を行う
→他のデータセットを参照するには、あらかじめ準備が必要

`data A;`
`set B;`



`run;`

そこで、参照したい他のデータセットをハッシュオブジェクトに格納することで、データのやり取りが可能になる

使用するマシンのスペックに左右されるが、メモリを使用する都合上、大容量データをハッシュオブジェクトに格納するとエラーになることがある
→条件で絞る等の工夫が必要になるケースも

業務実用例（フラグ変数の作成）

異なるデータセットの情報からフラグを作りたい

→例：有害事象のデータセットから重篤な有害事象を発現した症例を抽出し、フラグを作りたい
（でもデータセットのサイズも大きく、SORT→MERGEはしたくない…）

→ハッシュオブジェクトのCHECKメソッドとMULTIDATA: 'Y' を
使えば楽にできる!!

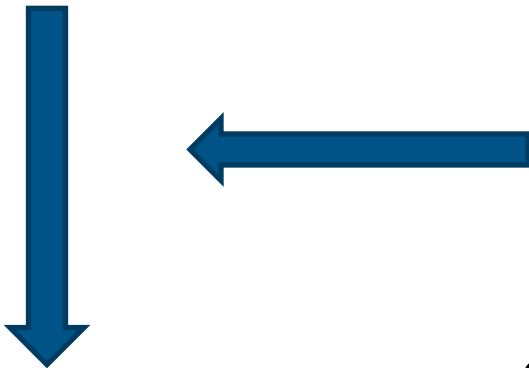
→マクロ作成

業務実用例（フラグ変数の作成）

```
data BACK;  
CASEID=1;output;  
CASEID=2;output;  
CASEID=3;output;  
run;
```

	CASEID
1	1
2	2
3	3

	CASEID	DOSE	ROOT
1	1	10	経口
2	1	10	経口
3	2	10	皮下
4	2	5	舌下
5	3	10	



	CASEID	ROOT1	ROOT2	ROOT3
1	1	1	.	.
2	2	.	1	1
3	3	.	.	.

☺ 処方データの変数ROOTの「経口」「皮下」「舌下」を参照して、BACKにフラグを立てたい…

業務実用例（フラグ変数の作成）

```
/*=====
FLGMAKEマクロ

-----
データステップ中に挿入し、マスター(MASTER=)で指定したデータセット
から指定したキー(KEY=)に一致するレコードについてフラグ立てするマクロ。

MASTER          :データを持てきたいマスターとなるデータセット
KEY              :ルックアップするためのキー変数(複数指定可能)
FLG              :フラグを立てる変数名を設定
WH              :条件を設定(WH=%STR(AEFLG="Y"))
TYPE            :該当/非該当 TYPE=1(1/2) TYPE=2(1/0) TYPE=3(Y/N) TYPE=4(1/.)(デフォルトはTYPE=1)
MISSWARN        :元データのKEY欠損を許容するか否か(デフォルトはN(欠損はWARNINGとして出ます))

-----
・SAS 9.13以上の環境

オリジナル:20181105 坂尻 大樹

-----
【注意 点】
・IF文の中にネストしての使用は不可能です
  [NG例] DATA A;
              SET B;
              IF XX = 1 THEN DO;
                  %LOOKUP(MASTER=MASTER , KEY =ID,VAR= A B);
              END;
              RUN;

-----
【記載 例】
DATA AD$;
  SET HEAD;
  %FLGMAKE(MASTER=AE,KEY=STUDY SUBJID,FLG=AEUMU,WH=%STR(AEFLG="Y"));
RUN;
=====*/
```

```
%MACRO FLGMAKE(MASTER=,KEY=,FLG=,WH=,TYPE=1,MISSWARN=N);
/*設定*/
%LOCAL NAME QKEY KEYNUM KEY WH;
%LET KEY=%SYSFUNC(COMPBL(&KEY.));
%LET NAME = &SYSINDEX.;
%LET QKEY = %SYSFUNC( TRANWRD( %STR("&KEY.") , %STR( ) , %STR(",") ));
%LET KEYNUM = %SYSFUNC( COUNT( &KEY., %STR( ) ));

/*参照元データ準備*/
IF 0 THEN SET &MASTER(KEEP= &KEY);

/*IF文にネストして使われた際、_N_=1では危険なので念のため*/
RETAIN _N_&NAME 1;

/*ハッシュ設定*/
IF _N_&NAME = 1 THEN DO;

    %IF %LENGTH(&WH) NE 0 %THEN %DO;
        DECLARE HASH H&NAME.(DATASET:"&MASTER( WHERE = (" || %SYSFUNC(QUOTE(&WH.)) || ")" , MULTIDATA:'Y');
    %END;
    %ELSE %DO;
        DECLARE HASH H&NAME.(DATASET:"&MASTER.(KEEP= &KEY.)", MULTIDATA:'Y');
    %END;

    H&NAME..DEFINEKEY(&QKEY.);
    H&NAME..DEFINEDONE();
    _N_&NAME = 0 ;

    DROP _N_&NAME.;
END;

/*フラグの変数を初期化*/
%IF %LENGTH(&KEY.) NE 0 %THEN %DO;
    CALL MISSING(OF &FLG.);
%END;

/*フラグ立て*/
%IF %UPCASE(&TYPE.) EQ 1 %THEN %DO;
    &FLG. = IFN(H&NAME..CHECK()=0,1,2);
%END;
%IF %UPCASE(&TYPE.) EQ 2 %THEN %DO;
    &FLG. = IFN(H&NAME..CHECK()=0,1,0);
%END;
%IF %UPCASE(&TYPE.) EQ 3 %THEN %DO;
    &FLG. = IFC(H&NAME..CHECK()=0,"Y","N");
%END;
%IF %UPCASE(&TYPE.) EQ 4 %THEN %DO;
    &FLG. = IFN(H&NAME..CHECK()=0,1,.);
%END;

/*元データのKEY欠損を許容するか否か*/
%IF %UPCASE(&MISSWARN.)= N %THEN %DO;
    IF CMISS(OF &KEY.)>0 THEN DO;
        PUT LOG "WARNING:元データ内にKEY欠損があります。" +2 (&KEY.) (=);
        CALL MISSING(OF &FLG.);
    END;
%END;
%MEND FLGMAKE;
```

業務実用例（フラグ変数の作成）

```
data ADSL;  
  set BACK;  
  %FLGMAKE (MASTER=ADEX, KEY=CASEID, FLG=ROOT1, WH=%STR(ROOT="経口"), TYPE=4);  
  %FLGMAKE (MASTER=ADEX, KEY=CASEID, FLG=ROOT2, WH=%STR(ROOT="皮下"), TYPE=4);  
  %FLGMAKE (MASTER=ADEX, KEY=CASEID, FLG=ROOT3, WH=%STR(ROOT="舌下"), TYPE=4);  
run;
```

マクロの引数について

```
%FLGMAKE (MASTER=, KEY=, FLG=, WH=, TYPE=1, MISSWARN=N);
```

MASTER :データを持ってきたいマスターとなるデータセット

KEY :ルックアップするためのキー変数（複数指定可能）

FLG :フラグを立てる変数名を設定

WH :条件を設定(WH=%STR(AEFLG="Y"))

TYPE :該当/非該当（デフォルトはTYPE=1）

TYPE=1(1/2) TYPE=2(1/0) TYPE=3(Y/N) TYPE=4(1/.)

MISSWARN :元データのKEY欠損を許容するか否か

(デフォルトはN (欠損の場合、WARNINGとして出ます))

業務実用例（他のデータセットから変数を取得）

異なるデータセットの変数を取得したい

→例：患者背景のデータセットから年齢、性別等の背景情報を取得したい
（でもデータセットのサイズも大きく、SORT→MERGEはしたくない…）

→ハッシュオブジェクトのDEFINEDONEメソッドを使えば楽にできる!!

→マクロ化

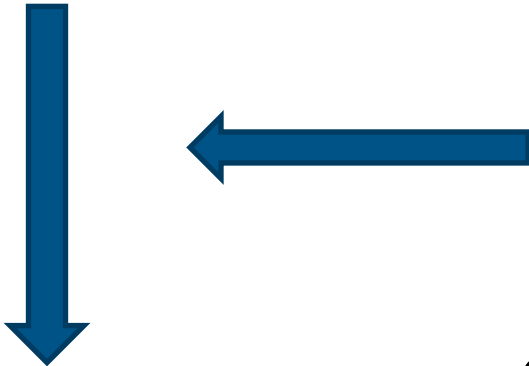
イーピーエス 森岡さんのマクロをお借りしました。
ありがとうございます。

業務実用例（他のデータセットから変数を取得））

```
data BACK;  
CASEID=1;output;  
CASEID=2;output;  
CASEID=3;output;  
run;
```

	CASEID
1	1
2	2
3	3

	CASEID	SEX	AGE
1	1	1	10
2	2	2	20
3	3	1	30
4	4	2	40



	CASEID	SEX	AGE
1	1	1	10
2	2	2	20
3	3	1	30

😊 患者背景データの年齢、性別を取得したい…

業務実用例（他のデータセットから変数を取得）

```
/*=====
lookupマクロ

-----
データステップ中に挿入し、マスター(master=)で指定したデータセット
から指定したキー(key=)に一致するレコードについて
必要な変数(var=)をマスターから持ってきて結合するマクロ。

master          :データを持てきたいマスターとなるデータセット(keyで一意の必要あり)
key              :ルックアップするためのキー変数(複数指定可能)
var              :マスターから持てきたい変数(複数指定可能)
warn             :マスターに該当キー値が見つからなかった場合のログへのWARNINGの出力(Y/N)
flg              :マスターに該当キーが見つかった場合に1を返す変数名を設定
-----
・SAS 9.13以上の環境

オリジナル:20170301 森岡 裕

-----
【注意点】
・指定したキー変数でユニークになっているマスターを使用してください。
  以下のエラーがでる場合は、master=で指定したデータセットがkey=で指定した
  変数でユニークになっていないことが考えられます。
      ERROR: データセット master(keep= XX) を 行 X カラム XXにロードする際に、キーの重複が見つかりました。
      ERROR: ハッシュデータセットのロードは、行 X カラム XXで失敗しました。
      ERROR: DATA STEP コンポーネントオブジェクトエラーEXECUTION フェーズ時に中止されました。
・付与する変数がすでにデータセットに存在する場合、マスターの値で上書きし、
  アンマッチのレコードはブランクで上書きされます。

・if文の中にネストしての使用は不可能です
  [NG例] data A;
      set B;
      if XX = 1 then do;
          %Lookup(master=master , key =ID,var= a b);
      end;
  run;

【使用例①】
データセットmasterからIDをキーとしてaとbをrawに結合する

data raw;
    set raw;
    %Lookup(master=master , key =ID,var= a b)

run;

【使用例②】
データセットmasterにIDの値をキーとして同値があるかを判定し、ある場合はF1に1を返す

data raw;
    set raw;
    %Lookup(master=master , key =ID,flg=F1)

run;

※var=を指定してデータを取りつつ、フラグを設定することも可能
data raw;
    set raw;
    %Lookup(master=master , key =ID,var= a b ,flg=F1)

run;

=====*/
```

```
%macro Lookup(master=,key=,var=,warn=N,flg=);
    %local name qkey qvar keynum key var;
    %let key=%sysfunc(compbl(&key));
    %if %length(&var) ne 0 %then %do;
        %let var=%sysfunc(compbl(&var));
    %end;
    if 0 then set &master(keep= &key &var);
    %let name = &sysindex;
    retain _N_&name 1;
    if _N_&name = 1 then do;
        %let qkey = %sysfunc( tranwrd( %str("&key") , %str( ) , %str(",") ) );
        %let keynum = %sysfunc( count( &key, %str( ) ) );
        declare hash h&name.(dataset:"&master(keep= &key &var)" , duplicate:'E');
        h&name..definekey(&qkey);
        %if %length(&var) ne 0 %then %do;
            h&name..definedata(all:'Y');
        %end;
        h&name..definedone();
        _N_&name = 0 ;
    end;
    drop _N_&name ;
    if h&name..find() ne 0 then do;
        %if %length(&var) ne 0 %then %do;
            call missing(of &var &flg);
        %end;
        %if %upcase( &warn )= Y %then %do;
            if cmiss(of &key) ne &keynum +1 then putlog "WARNING:マスターに存在しないキーのレコードです" +2 (&key.) (=);
        %end;
    end;
    %if %length(&flg) ne 0 %then %do;
        else &flg = 1;
    %end;
%mend Lookup;
```

業務実用例（他のデータセットから変数を取得）

```
data ADSL;  
  set BACK;  
  %LOOKUP (MASTER=BASE, KEY=CASEID, VAR=AGE SEX);  
run;
```

マクロの引数について

```
%LOOKUP (MASTER=, KEY=, VAR=, WARN=, FLG=);
```

MASTER	: データを持てきたいマスターとなるデータセット (KEYで一意的必要あり)
KEY	: ルックアップするためのキー変数（複数指定可能）
VAR	: マスターから持てきたい変数（複数指定可能）
WARN	: マスターに該当キー値が見つからなかった場合のログへのWARNINGの出力 (Y/N)
FLG	: マスターに該当キーが見つかった場合に1を返す変数名を設定

業務実用例（実例）

1つのデータステップ内で、異なるデータセットから何回でも結合できる（可読性も良くなります）

```
date ADSL;  
set BACK;
```

```
%LOOKUP(MASTER=SUBSET1 ,KEY=CASEID ,VAR=SAF SAF1 SAF2 EFF EFF1 EFF2);  
%LOOKUP(MASTER=SUBSET2 ,KEY=CASEID ,VAR=AESTDATE AEEDDATE);  
%LOOKUP(MASTER=SUBSET3 ,KEY=CASEID ,VAR=DOSEDATE DOSEID);
```

```
if DOSEDATE^=. then TRTSTDATE=DOSEDATE;  
if DOSEDATE^=. and DOSEID in (1,2) then TRTID=1;
```

```
%FLGMAKE(MASTER=SUBSET4, KEY=CASEID, FLG=DOSEFLG1, WH=%STR(200<=DOSE));  
%FLGMAKE(MASTER=SUBSET4, KEY=CASEID, FLG=DOSEFLG2, WH=%STR(<DOSE<200));
```

```
run;
```

業務実用例～応用編～ （多対多の処理）

%LOOKUPはキー変数が1対1、%FLGMAKEは1対1・1対多を想定

→多対多の処理はなんとかならないの？

→例：薬剤の処方日と併存疾患の発現日と比較してフラグを立てたい…
（どちらも1症例複数レコード…）

→マクロ化検討中…（次回発表できたら…）

業務実用例～応用編～（多対多の処理）

```
data DOSE;  
  format CASEID 8. DOSEDATE YYMMDD10.;  
  CASEID=1; DOSEDATE=INPUT("2024/1/1",YYMMDD10.); output;  
  CASEID=1; DOSEDATE=INPUT("2024/2/1",YYMMDD10.); output;  
  CASEID=2; DOSEDATE=INPUT("2024/1/1",YYMMDD10.); output;  
  CASEID=2; DOSEDATE=INPUT("2024/4/1",YYMMDD10.); output;  
run;
```

	CASEID	DOSEDATE
1	1	2024-01-01
2	1	2024-02-01
3	2	2024-01-01
4	2	2024-04-01

```
data MH;  
  format CASEID 8. MHSTARTDATE MHENDDATE YYMMDD10. ;  
  CASEID=1; MHSTARTDATE=INPUT("2024/1/1",YYMMDD10.); MHENDDATE=INPUT("2024/1/31",YYMMDD10.); output;  
  CASEID=2; MHSTARTDATE=INPUT("2024/3/1",YYMMDD10.); MHENDDATE=INPUT("2024/5/31",YYMMDD10.); output;  
  CASEID=2; MHSTARTDATE=INPUT("2024/3/1",YYMMDD10.); MHENDDATE=INPUT("2024/4/30",YYMMDD10.); output;  
run;
```

	CASEID	MHSTARTDATE	MHENDDATE
1	1	2024-01-01	2024-01-31
2	2	2024-03-01	2024-05-31
3	2	2024-03-01	2024-04-30

業務実用例～応用編～（多対多の処理）

	CASEID	DOSEDATE
1	1	2024-01-01
2	1	2024-02-01
3	2	2024-01-01
4	2	2024-04-01

	CASEID	MHSTARTDATE	MHENDDATE
1	1	2024-01-01	2024-01-31
2	2	2024-03-01	2024-05-31
3	2	2024-03-01	2024-04-30

	CASEID	DOSEDATE	FLG
1	1	2024-01-01	1
2	1	2024-02-01	0
3	2	2024-01-01	0
4	2	2024-04-01	1

業務実用例～応用編～（多対多の処理）

```
data SUBSET;
set DOSE;
*****ハッシュオブジェクトを定義;
if 0 then set MH;
if _n_=1 then do;
  dcl hash hs (dataset:"MH",multidata:"Y");
  hs.definekey ("CASEID");
  hs.definedata("CASEID","MHSTARTDATE","MHENDDATE");
  hs.definedone();
end;
*****ハッシュオブジェクト処理;
do while(hs.do_over()=0);
  if .<MHSTARTDATE<=DOSEDATE<=MHENDDATE then FLG=1;
  else FLG=0;
end;
*****ハッシュオブジェクト処理終了;

if FLG ne 1 then FLG=0;

keep CASEID DOSEDATE FLG;
run;
```

do while、 do_overメソッドの組み合わせ

ループさせているため、elseでFLG=0を設定すると、ループ1週目で1が立たったとしても、ループ2週目以降で該当しなければ0になる場合もある

（同一変数内で分岐させると1→0→1→0のように値が変動する可能性がある）



参考文献

(1) データステップ100万回 SAS新手一生 ([データステップ100万回 SAS新手一生 \(sas-tumesas.blogspot.com\)](http://sas-tumesas.blogspot.com))

「ハッシュオブジェクトの世界①」

「ハッシュオブジェクトの世界③ checkメソッドと、メソッドの戻り値をそのままIFC IFN関数に使う話」

「ハッシュオブジェクトの世界⑥ outputメソッド ordered」

「大半のソートは百害あって一利無しという話」

(2) SAS備忘録 ([SAS忘備録 \(sas-boubi.blogspot.com\)](http://sas-boubi.blogspot.com))

「Hashオブジェクトは、CDISCに係るSASプログラミングと相性バッチリ」

(3) SASユーザー総会2014 ([スライド1 \(sas.com\)](http://sas.com))

「SAS ハッシュオブジェクトを利用して医薬品開発に使用するプログラムを効率化する – 有害事象と併用薬、臨床検査値と途中変更のある施設基準値のマッチングからSASプログラムコードの分析まで」

森岡裕

(4) SAS ユーザー総会2018 論文集 ([セッション | SAS](http://sas.com))

「DOSUBL関数内で生成したビューをハッシュオブジェクトに格納することによりPROC DS2の ハッシュパッケージの挙動を擬似的に再現する方法」

森岡裕

経営理念

●基本理念

価値あるソリューションの創出を通じて、健康産業の発展に貢献します

●組織理念

日々新たに、また日に新たなり
Ever Progressing System

●行動指針

顧客志向 私たちは、お客様のことを第一に考え、お客様の価値創造に貢献します

ビジネス志向 私たちは、ビジネスの持続的な発展を通じて、社会の発展に貢献します

人間志向 私たちは、仕事を通じて成長し、すべてのステークホルダーの
QOL (Quality of Life) の向上に貢献します



イーピーエス株式会社