

For Higher
Customer Satisfaction,
We Bridge
the SAS System
Between
Customer's World.

SAS Technical News

AUTUMN 2010

40.00	-

意外と知らない!?

SAS Academic News ····· 08

- 事例紹介
- コラム「SAS四方山話」

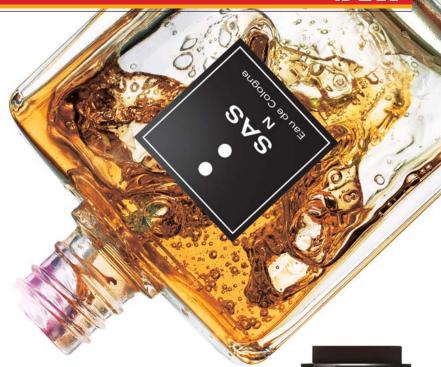
Q&A · · · · · · · · · · · · · · · · · · ·	14
SASトレーニングのお知らせ ·····	16

最新リリース情報 ・・・・・・・・・ 20



Mysterious Colon

SAS言語ではさまざまな記号が使われます。中でも、ステートメントの終りを示すためのセミコロン、語句を区切るためのコンマ、文字列を囲うためのクォーテーションはSASプログラムに不可欠な記号であり、それらの意味・機能はよくご存知だと思います。一方で、コロン(:)の機能について知っている方は非常に少ないように思います。事実、弊社テクニカルサポートへ送られてくるSASプログラムの中でもコロンを見かけることは殆どありません。また、弊社トレーニング、マニュアルの中でもコロンに特化した記述はありません。では、SASプログラムにおけるコロンとはどういうものなのでしょうか。



特集

意外と知らない!? コロンの活用術をご紹介

SAS言語においてコロンが持つ意味・機能を一概に説明することができません。なぜならば、コロンはさまざまなステートメントの中で使用することが可能で、それぞれの場合において意味・機能が異なるためです。では、実際にこのコロンにはどのような機能があるのか?このコロンはどのように使うと便利なのか?をご説明いたします。

コロンには以下のような機能があります。

- ・前方一致
- ・フォーマットモディファイヤ
- ・マクロ変数の作成
- ・SASログのキーワード強調
- ・ラベルの指定
- ・引数の指定 (SAS/AF)
- ・配列の上限と下限の指定

何だか謎めいた雰囲気を持つ記号ではありますが、その謎を解き明かす価値はあります。コロンの機能を理解し、プログラミングに活かせば、他の記号では行うことができない特殊な処理の実現や、プログラムの可読性アップを図ることができるためです。

本特集では、コロンの機能のうち特に利便性が高いものを3つ選び、 それぞれサンプルプログラムを交えてご紹介します。

1.前方一致

2.フォーマットモディファイヤ

3.マクロ変数の生成

前方一致

データの抽出を行うとき、先頭の一部分しか分からない場合、通常は SUBSTR関数やKSUBSTR関数を用います。このような場合、コロンに よる前方一致を用いることで、語句の先頭が一致する文字列だけを 処理することができます。例えば「ソフト」という語句に前方一致する 文字列を抽出する処理を選択した場合「ソフト」「ソフトウェア」「ソフトクリーム」などがヒットします。

コロンによる前方一致は、文字列の他に、変数名、データセット名、フォーマットなどに対しても適用可能です。また、前方一致に合わせて命名規則を工夫することで、プログラムの効率を上げることができます。以下、6つの具体例をご紹介します。

1.1 演算子として

コロンを演算子として使うと、変数var1の値が「日本」から始まるオブザベーションを選択するというようなプログラムを効率的に作成できます。 例えば、変数var1に格納されている値の初めの2文字を抽出して、その文字列が「東京」であれば処理を行う場合、KSUBSTR関数では次のような記述となります。

例 KSUBSTR関数を使った前方一致

```
DATA sample1a;
 INPUT var1 $ 16.;
 IF KSUBSTR(var1,1,2)='東京'
                          /* 東京に前方一致する場合 */
   THEN include=1;
   ELSE include=0;
CARDS:
東京都
東京特許許可局
東京スカイツリー
大阪駅
米国本社
RUN;
```

一方、コロンを使えば上記と同一の処理をより簡易な記述で行うことが できます。イコール演算子の後にコロンを記述することで、IFステートメント の条件が「完全一致」から「前方一致」へと変化するため、KSUBSTR 関数が不要となります。

例 コロンを使った前方一致

```
DATA sample1b;
 INPUT var1 $ 16.;
 IF var1=:'東京'
                          /* 東京に前方一致する場合 */
   THEN include=1;
   ELSE include=0;
CARDS:
東京都
東京特許許可局
東京スカイツリー
大阪駅
米国本社
RIIN .
```

もし、検索条件を「東京」もしくは「大阪」から始まる文字列へと変更した い場合、どのようにプログラムを書き換えればいいでしょうか? 関数を使う場合、次のようなプログラムが考えられます。

例 KSUBSTR関数を使った複数前方一致

```
DATA sample1c;
 INPUT var1 $16.;
                  /* 東京もしくは大阪に前方一致する場合 */
 IF KSUBSTR(var1,1,2)='東京' or KSUBSTR(var1,1,2)='大阪'
   THEN include=1;
   ELSE include=0;
CARDS;
東京都
東京特許許可局
東京スカイツリー
大阪駅
米国本社
RUN;
```

IN演算子の後にコロンを記述(IN:)すると、「〇〇」もしくは「××」から始 まる文字列だけを処理ができます。KSUBSTR関数やOR演算子を 使った記述が不要となり、プログラムが簡素化されます。

例 コロンを使った複数前方一致

```
DATA sample1d;
 INPUT var1 $16.;
  IF var1 IN: ('東京','大阪') /* 東京もしくは大阪に前方一致する場合 */
   THEN include=1:
   ELSE include=0:
CARDS:
東京都
東京特許許可局
東京スカイツリー
大阪駅
米国本社
RUN:
```

1.2 一時変数の一括削除

DATAステップでデータ加工を行うとき、最終的にはデータセットへ 出力しない変数(一時変数)を作成することがあります。

次のサンプルプログラムでは、変数pi、変数e、変数cを作成したものの、 データセットへの出力は不要なため、DROPステートメントでそれらを 指定しています。

例 変数の削除

```
DATA sample2a;
  pi=3.14;
  e=2.71:
  c=0.577:
  result=pi*e*c;
                                   /* 一時変数の削除 */
  DROP pi e c;
RUN;
```

今回のように一時変数の数が少ない場合は前ページのプログラムで 問題ありませんが、多くの一時変数を扱うプログラムの場合はそうは いきません。全体の変数の数が多くなると、一時変数とその他の変数 との区別が難しくなり、また、DROPステートメントやKEEPステートメント で変数名を記述する負担が大きくなります。

このような場合、一時変数名の先頭の文字列を統一することで、通常 の変数と区別できます。例えば先頭の文字列を「tmp」に統一すれば、 一時変数であることが一目瞭然です。

```
tmppi=3.14;
tmpe=2.71:
tmpc = 0.577:
```

さらに、この命名規則とコロンの前方一致の効果を組み合わせれば、 短い記述で変数名が「tmp」から始まる全ての変数を削除できます。

DROP tmp::

以下のプログラムでは、3つの一時変数は削除されるため、変数result だけがデータセットへ出力されます。

例 コロンを使った変数の一括削除

[特集] Mysterious Colon

```
DATA sample2b;

tmppi=3.14;

tmpe=2.71;

tmpc=0.577;

result=tmppi*tmpe*tmpc;

DROP tmp:; /* 一時変数の一括削除 */

RUN;
```

なお、コロンはKEEPステートメントの中でも同じように使うことができます。例えば「変数名がresから始まる全ての変数」を残す場合は次のように記述します。

KEEP res:;

1.3 データセットの一括削除

データ加工の過程で、多くの一時データセットを作成する場合があります。これらはプログラムの実行終了後は不要で、余分なディスクスペースを取るだけですので、削除するケースが多いと思います。このような処理もコロンを使うことで効率良く行うことができます。

まず、変数を一括削除したときと同じように、命名規則を設定します。 今回は、一時データセットであることを示すためにデータセット名をアン ダースコア(_)から始めます。今回は、マクロプログラムの終りに、コロン を用いて一時データセット全削除処理を記述します。

例 コロンを使ったデータセットの一括削除

```
%MACRO sample3;
DATA _test1 _test2 _test3 remain; /* 4つのデータセットを作成 */
SET sashelp.class;
RUN;
PROC DATASETS LIBRARY=work;
DELETE _:; /* アンダースコアから始まるデータセットを一括削除 */
QUIT;
%MEND sample3;
%sample3;
```

上記プログラムを実行すると、アンダースコアから始まる3つのデータセットは削除され、データセットremainだけがWORKライブラリに残ります。

1.4 変数の一括フォーマット/インフォーマット

コロンをFORMATステートメントやINFORMATステートメントの中で使うことで、一括フォーマット/インフォーマットを行うことができます。 以下の例をご覧ください。3種類の日付変数に同一のインフォーマット、フォーマットを適用しています。

例 変数のフォーマット/インフォーマット

```
DATA sample4a;
INPUT birthdate startdate enddate;
INFORMAT birthdate startdate enddate DATE.; /* インフォーマットの適用 */
FORMAT birthdate startdate enddate MMDDYY9.; /* フォーマットの適用 */
CARDS;
21MAR1980 01JAN2000 05JAN2010
01FEB1984 01JAN2000 01FEB2004
12JAN1991 01JAN2000 23DEC2002
22APR1983 01JAN2000 11DEC2009
;
RUN;
```

このようなプログラムも、コロンを使うことで簡略化できます。 birthdate、startdate、enddate

という変数名の代わりに、

datebirth, datestart, dateend

というふうに変数名が「date」から始まるよう命名します。

こうすることで、FORMAT/INFORMATステートメントの中でコロンを使った前方一致を用いることができます。

例 コロンを使った変数の一括フォーマット/インフォーマット

```
DATA sample4b;

INPUT datebirth datestart dateend;

INFORMAT date: DATE. /* インフォーマットの一括適用 */

FORMAT date: MMDDYY9.; /* フォーマットの一括適用 */

CARDS;

21MAR1980 01JAN2000 05JAN2010

01FEB1984 01JAN2000 01FEB2004

12JAN1991 01JAN2000 23DEC2002

22APR1983 01JAN2000 11DEC2009

;

RUN;
```

1.5 要素が未知数である配列の扱い

コロンを配列(ARRAYステートメント)で使うと「総数の不明な変数を 配列でまとめて処理する」というようなプログラムを効率的に作成でき ます。

例えば、以下の処理を行うプログラムを書く場合を考えます。

- 1.1人の顧客(変数id)につき1つのオブザベーションとなるようデータを 転置する
- 2.商品の価格(変数price)が100以上の場合、その商品について10%の 割引を行う
- 3.割引を適用した後の合計購入額を計算する

例 元のデータ

OBS	id	price
1	00001	100
2	00001	50
3	00002	200
4	00003	100
5	00003	200
6	00003	150

例 求める結果

OBS	id	_NAME_	COL1	COL2	COL3	total
1	00001	price	90	50		140
2	00002	price	180			180
3	00003	price	90	180	135	405

次のプログラムのように、TRANSPOSEプロシジャを使ってデータを転 置すると、BYステートメントによって新しく作成された変数に対して COL1、COL2、COL3…という変数名が自動的に設定されます。これらの 変数を加工(10%の割引と合計購入額の算出)するには、配列を使う 必要がありますが、変数の数がデータに依存するため、配列の定義が 行えないという問題が生じます。

例 データセットの作成および転置

```
DATA sample5a;
 INPUT id $5. price;
CARDS;
00001 100
00001 50
00002 200
00003 100
00003 200
00003 150
RUN;
PROC TRANSPOSE DATA=sample5a OUT=sample5b;
  BY id:
  VAR price;
RUN:
```

変数の数を多めに見積り、以下のように定義することは可能です。 しかし、このようなプログラムは開発する上で好ましくありません。

```
ARRAY col[999] col1-col999:
```

```
DO i=1 TO DIM(col):
```

{… グループ処理…};

END;

ARRAYステートメントの変数定義にてコロンを記述すれば、名前が「col」 から始まる全ての変数を配列に格納することができます。

```
ARRAY col(*) col:;
```

DO i=1 TO DIM(col);

{… グループ処理…};

END:

例として次のようなプログラムが考えられます。

例 割引を適用後の合計購入額を計算

```
DATA sample5a;
  INPUT id $5. price;
CARDS;
00001 100
00001 50
00002 200
00003 100
00003 200
00003 150
RIIN:
PROC TRANSPOSE DATA=sample5a OUT=sample5b;
 BY id:
  VAR price;
RUN;
DATA sample5c;
 SET sample5b;
  BY id:
  ARRAY col{*} col:;
                                    /* アイテム数不明 */
  IF first.id
   THEN total=0:
  DO i=1 TO DIM(col);
   IF col(i) >= 100
                                            /* 割引 */
      THEN col(i) = col(i) *0.9;
    total+col(i);
  END;
  DROP i;
RUN;
```



1.6 その他の応用

プロシジャの中でもコロンによる短縮記述を使うことはできます。例えば、SORTプロシジャの中でコロンを使うと以下のようにBY変数の記述を簡略化できます。

例 コロンを使ったBY変数の一括指定

```
DATA sample6b;
INPUT skid skname $ gender $ country $; /* 接頭SKを付与する */
CARDS;
1 JOHN M USA
4 MIKE M USA
3 YAMADA F JPN
2 SUSAN F ENG
5 JULIA F SPA
;
RUN;
PROC SORT DATA=sample6b;
BY sk:; /* 前方一致により変数を指定 */
RUN;
```

注) SORTプロシジャ内でコロンによる前方一致を使った場合、その並び順はデータ 入力時の変数の並びに影響を受けます。変数の並びは、CONTENTSプロシジャを 使用してご確認ください。

その他、CLASSステートメントで指定する変数や、VARステートメントで 指定する変数も以下のようにコロンを使って記述することが可能です。

CLASS cla:; VAR var:;

前方一致の用法とメリットを理解した上で、ご自身のコードを見つめ直していただければ、コロンを活用できる箇所が見つかると思います。

フォーマットモディファイヤ

コロンをINPUTステートメントの中で使うと、区切り文字で区切られたデータを、任意のインフォーマットに沿って読み込むことが可能になります。このようにデータの読み込み時にオプションとして用いられるコロンを「フォーマットモディファイヤ」と呼びます。

例えば次のようなデータを通常のINPUTステートメントで読み込もうとすると、エラーが表示され、文字データの一部が正常に読み込まれません。

例 フォーマットモディファイヤを使わないデータ読み込み

```
DATA sample7a; /* 正しく読み込むことができない */
INPUT word $11. number; /* インフォーマットで変数長を指定 */
CARDS;
observation 11
colon 5
underscore 10
;
RUN;
```

例 読み込みエラーのログ

そこで、次のように変数名(word)とインフォーマット(\$11.)の間にコロンを記述します。間に記述されたコロンはフォーマットモディファイヤとして機能し、空白で区切られたデータを正しく読み取ることができるようになります。

例 フォーマットモディファイヤを使ったデータ読み込み

```
DATA sample7b; /* 正しく読み込める */
INPUT word : $11. number; /* フォーマットモディファイヤを使用 */
CARDS;
observation 11
colon 5
underscore 10
;
RUN;
```

マクロ変数の生成

データセットの検索を行う際、マクロ変数を利用することで柔軟な検索 処理が実現される場合があります。

非常に小さいデータセットに格納された値を、大きいデータセットから 検索する場合を考えます。まず、小さいデータセットの各オブザベーション の値をマクロ変数に格納します。その後、大きいデータセットをDATA ステップで読み込み、IFステートメント(もしくはWHEREステートメント) の条件にマクロ変数の値を指定することで検索を行うことができます。 SQLプロシジャの中でコロンを使うことで、上記のようなケースに適した 形のマクロ変数を生成することが可能です。具体的には、SELECT ステートメントのINTO句を使います。コロンはINTO句の構文の一要素 として、マクロ変数の前に記述されます。

例 INTO句の構文

コロンはSQLプロシジャの中でも使われます

PROC SQL;

SELECT 変数名 INTO : マクロ変数名 [SEPARATED BY 区切り文字] FROM データセット名もしくはテーブル名; QUIT;



INTO句を用いると、区切り文字を指定して、複数のデータ値をマクロ変数に格納できます。記号を含むマクロ変数のデータ加工は注意を要しますが、INTO句を使うとその必要がありません。

INTO句が活躍する具体例として、データセットが持つ変数の値(複数オブザベーション)を、別のデータセットのIN演算子の中で使う処理を考えます。

複数の変数値をマクロ変数に格納

データセットの(変数の)値 00001 00003 00005

上記のような処理を実現するには、変数の各オブザベーションの値を コンマ区切りでマクロ変数に格納し、後続のDATAステップでIN演算子 の中にマクロ変数を展開します。また、データが文字列の場合、QUOTE 関数で各値をダブルクォーテーションで囲う必要があります。

IN演算子でのマクロ変数の展開

マクロ変数展開前 IF id IN (&idkey); IF id IN ("00001","00003","00005");

このような場合はINTO句を用いることで、簡単にデータをコンマ区切りでマクロ変数に格納できます。以下のサンプルプログラムをご覧ください。データセットの変数値を、別のデータセットの中から検索し、一致したオブザベーションを抽出します。この場合、INTO句を使って変数idの値をコンマ区切りでマクロ変数idkeyに格納しています。

例 マクロ変数を使ったID検索

```
DATA sample8a;
 INPUT id : $5.;
                                 /* 検索を行うid番号 */
CARDS:
00001
00003
00005
RUN:
PROC SQL NOPRINT;
                   /* マクロ変数idkeyをコンマ区切りで作成 */
 SELECT QUOTE(id) INTO : idkey SEPARATED BY ','
   FROM sample8a;
QUIT;
DATA sample8b;
 INPUT id : $5. student : $10.;
 IF id IN (&idkey); /* マクロ変数idkeyに含まれる値を抽出 */
CARDS;
00001 TANAKA
00002 SUZUKI
00003 SATO
00004 YAMADA
00005 NAKAMURA
run;
```



本特集では、マイナーでミステリアスな記号、コロンにスポットライトを当て、その機能と活用法をご紹介しました。他のステートメントと併せて、工夫して使うことで強力な効果を発揮する記号であることがご理解いただけたと思います。SASプログラムを書かかれる際、本特集の内容を思い出してコロンをご活用いただければ幸いです。

[参考資料]

- 1. Colonoscopy for the SAS® Programmer, Haiping Luo http://www2.sas.com/proceedings/sugi29/054-29.pdf
- 2. That Mysterious Colon (:), William C. Murphy http://www2.sas.com/proceedings/sugi26/p073-26.pdf
- 3. Base SAS® 9.2 Procedures Guide SELECT Statement http://support.sas.com/documentation/cdl/en/proc/618 95/HTML/default/a002473678.htm
- 4. SAS® 9.2 Language Reference: Dictionary, Third Edition INPUT Statement, List

http://support.sas.com/documentation/cdl/en/lrdict/63 026/HTML/default/a000144370.htm#a000144378

5. SAS® 9.2 Language Reference: Concepts, Second EditionSAS Operators in Expressions

http://support.sas.com/documentation/cdl/en/lrcon/62 955/HTML/default/a000780367.htm#a000695002

6. SAS® 9.2 Language Reference: Concepts, Second Edition
- Data Set Lists

http://support.sas.com/documentation/cdl/en/lrcon/62 955/HTML/default/viewer.htm#/documentation/cdl/en/lrcon/62955/HTML/default/a003040446.htm **SAS Academic News**

SAS アカデミック・ニュース

Academic News

2010年夏号から内容を一新した事例紹介と四方山話の第2号、

事例紹介ではEnterprise GuideのGUI操作でExcelデータの取り込みから、データの編集、追加、

変数の結合等、データ加工、さらにそのデータを他のプロジェクトで使用できるようにするための

エクスポート機能まで、一連の流れでご紹介しています。

「SAS四方山話」では、SASバージョン6時代に販売していましたSASクイックリファレンスを、

インターネット・エクスプローラーの「お気に入り」の機能を活用しプロシジャのシンタックス(文法)や

サンプル(事例)をすぐに確認できる、現代版デジタルクイックリファレンスの作成方法をご紹介しています。

事例紹介

コラム 「SAS四方山話」

事例紹介

高柳 良太 國學院大學 経済学部および人間総合科学大学 人間科学部 兼任講師

第2章 ExcelデータとSASの連携

ExcelデータをEGに取り込む

Excelのデータ(ワークシート)をEGに読 み込ますことは、取り立てて難しいことは ありません。形式さえ正しければ、対象とな るファイルとシートを指定するだけでイン ポートしてくれます。

データを扱うソフトウェアというと、教育の 場においてはExcelの独壇場といった感じ です。Excel以外にも、いわゆる表計算ソフ トは何種類もありましたが、今ではExcelが 完全にスタンダードになった感があります。 大学などで学生にデータを準備させる場 合も、Excelを使わせるのが普通です。

意外なことに、高校を出たての学生達で もあまりExcelになじみがないことが多い です。ましてやデータ解析などとなると「そ れって何ですか?」と言われてしまいます。 高校の「情報C」という科目では、簡単な表 計算ソフトを使ってデータ解析を行うこと になっているはずです。高校の「情報」は AからCまであるので、よほど情報Cを選択 している高校が少ないのかもしれません。

もちろん、その学生の意欲や能力にも よるのですが、データ解析を行うにあたっ ては、もう一度Excelの使い方を指導した 方がいい場合が多々あります。それでも、 さすがに若い人はすばらしく、あっという間 に習熟していく学生も多いです。

EGにもデータエディタがあります。ただ、 EGを初めて使う人が、せっかくExcelに慣 れているのであれば、それを生かしていき たいと思います。ただでさえ、統計解析 ソフトに対する心のハードルが高い人には 「Excelの続きっぽいですよ!!」というアプ ローチは有効だと思います。

Excelは、今やPCを購入すると大抵つい てきます。学生のプライベートのPCにも 大抵は入っているのですから、学生に宿題 としてデータを用意させることもできます。 実際に大学等では、学生の自分のPCで データの作成をさせ、学校のPCで分析を させるということになります。

Excelのような表計算ソフトをデータ入力 に使う利点は、普及しているということの 他にもあります。データの可視化が手軽に できるということです。

私が大学で初めてデータ解析を始めた 頃は、データはテキストエディタ等で準備 し、同じくエディタで作成するプログラムに、 そのデータファイルを指定していました。 つまり、分析対象となるデータを見ながら 分析をするのは、少々面倒でもありました。 WindowsのようなGUIが普及してくると、 各社の統計解析アプリも、GUIのデータ エディタを搭載するようになります。データ を表示しながら、データの分析ができるの は画期的なことでした。今の学生達は、 見えてない(開いていない)データに対して 分析を行うということに、感覚的について 行けないような気もします。

せっかくExcelでデータの扱いに慣れた のなら、それを生かしてEGに移行してもら えるといいと思います。今回はそういった点 から、Excelデータとの連携を中心に、変数 やケースの追加、計算データの追加など を説明したいと思います。



Excelデータの形式

Excelデータをインポートするには、デー タの形式にいくつかの条件があります。 大きな条件は以下の3つです。

- ・行方向がケース、列方向が変数
- ・1行目を変数名にする
- ・1セルに1つのデータ

	Α	В	C	D	E	F	G	Н	I
1	No	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
2	101	1	3	4	1	55	2	2	1
3	102	1	3	6.5	2	30	4	3	2
4	103	1	3	2.5	5	40	2	2	3
5	104	1	2	5	3	40	2	1	1
6	105	1	2	5	3	40	2	2	1

条件どおりのExcelデータの例

Excelで実際に入力すると、図のようにな ります。以前からCSV形式のデータでやり とりをしている人にしてみれば、条件と呼べ るほどのことでもないかと思います。ただ、 テキストエディタはおろかWindowsのメモ帳 すら見たこともない学生が増えた昨今、こう いうところから確認をしていかないとうまく 指導ができないというのもまた事実です。

行列方向の指定

行方向がケース、列方向が変数というの は、エディタでデータ入力をしていた人間に とっては自明のことでした。なぜなら、 エディタでは通常、入力がない状態では縦 方向にカーソルを移動することができない からです。

Excelの場合は、行列どの方向にデータ を入れても、ピボットテーブルなどの機能を 使わない限りはあまり問題が起きません。 ただ、EGに限らずデータ解析のソフトウェ アは1行名が変数というのが定石です。 もし、行列が逆方向のデータシートを分析 する場合は、以下の方法があります。

1) Excelでデータをコピーし[形式を選択して張 り付け]で「行列を入れ替える」を使用する

形式を選択して貼り付け 貼り付け で使用してすべて貼り付け付 数式田 罫線を除くすべて⊗ (M(V) **SMEW** 式と数値の書式(日) ◎ 値と数値の書式(」) スナ規模的(N) Ltacvo) ◎ 乗算(M) ◎ 除算(I) () 就算(S) ✔ 行列を入れ替える(日) □空白セルを無視する(B) OK キャンセル

Excelの[形式を選択して貼り付け]の[行列を入れ替える]

2) そのままの状態でEGにインポートし、 その後[データ]の[転置]を使用して新 しいデータテーブルを作成する



「データ]の[転置]

どちらでも結果的に同じように分析可能 のデータができあがります。ただ、この場合 は後述しますが1列目が変数名であること が必要となります。

いずれにしろ、後から編集作業が必要に なることは変わりないので、個人的には入 力時から行列方向を正しく設定しておいた 方がいいと考えます。

1行目が変数名

Excelで1行目が変数名の場合、インポー トのメニューで「範囲の先頭行にフィールド 名が挿入されている」のチェックをつけれ ば、1行目を変数名としてインポートしてくれ ます。



「範囲の先頭行にフィールド名が挿入されている」にチェック その下に「SAS命名規則に準拠するように列名を変更する」が あるので必要に応じてチェック

インポートしたデータをEGのみで使用 する場合は、変数名の文字をラベル化する ことに関する制限はあまりありません。 当然日本語(2バイト文字)も使用でき、 Base SASのような制限、つまり変数名の 先頭が数字でも問題ありません。また、 変数名の長さは半角32文字(全角16文字) までになっています。Excelから読み込む 場合、変数名は強制的に半角32文字以内 となりますが、ラベルにはそれを越える 長さも読み込まれます。

読み込んだデータをBase SAS等EG 以外のSASシステムで利用したい場合は、 インポート時に「SAS命名規則に準拠する ように列名を変更する」にチェックを入れ ます。この場合、変数名で使用できる文字 は半角英数字のみ、記号はアンダーバー (_)のみが可能となり、変数名の先頭に 数字を使用することが不可となります。

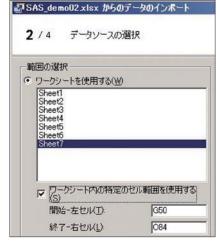
また、行列が逆になっているデータを インポートする場合、一度SAS形式にイン ポートして、それから前述の[転置]を使用 します。その場合、最初にインポートすると きには「範囲の先頭行にフィールド名が挿 入されている」のチェックをはずしてイン ポートします。そして、インポートしたデータ を転置する際に、変数名にしたい列(変数 名がインポートされている変数)を「新しい 列の名前」で指定します。



変数F1が本来の変数名なので、それを「タスクの役割」の 「新しい列の名前」に指定

Excelはどのセルからデータを入れても、 全く問題がありません。ひどい学生は、知ら ぬ間にマウスのホイール機能で50行目 ぐらいにアクティブセルが飛んでいるのに 気付かずに入力を始めてしまったりします。

ExcelのA1セルから変数名が始まって いないデータの場合でも、そこまでが空欄 の場合などはそのままでインポートでき ます。うまくいかない場合や、部分的にイン ポートしたい場合は「ワークシート内の特 定のセル範囲を使用する」をチェックして インポートします。チェックすると、開始位置 (左上)と終了位置(右下)を指定するよう になるので、してインポートします。



「ワークシート内の特定のセル範囲を使用する」に チェックすると、開始と終了位置が指定できる

この「ワークシート内の特定のセル範囲 を使用する」にチェックすると、例えば変数 名が複数行にわたっているようなファイル でも、変数として読み込ませたいセル位置 を指定すればEGにデータをインポートす ることができます。開始や終了位置がわか らない場合は、一度指定せずに読み込ま せ、プロセスフローで再指定をすれば、EG で位置を取得しているので編集を楽に することができます。

	>
- F- S	間(回)
(SAS_c	実行(B)
eet 🕟	データ(SAS_demo02nxlsx[Sheet6]) のインボート の変更(Y)
יערנט	データ (SAS_demo02nxlsx[Sheet6]) のインボート からの分岐を実行(B)
	入力データの選択(E)

プロセスフローで、インポートのアイコン上で右クリックをして 「データのインポートの変更」を選択する

EGにインポートするデータは、基本的に は変数名などでセル結合などのよけいな 装飾はしない方がいいでしょう。個人的に は、インポート用のシートを別途作成した 方がいいと思います。ちなみに、Excelには セル内の任意の場所でAltキーを押しな がらenterキーを押すと、セル内で改行が できますが、このセル内改行した変数名を EGにインポートしても、通常どおり読み込 まれます。ただし、EGのデータエディタには セル内改行の機能がないので、改行せず にそのままの表示となります。

1セル1データ

データ入力は1セルに1データとなり ます。これも、エディタでカラム形式での入 カや、CSVでの入力を経験してきた人には 当たり前のことです。ですが、Excelなどの 表計算ソフトの場合は、どんな入れ方でも できてしまうので、改めて注意が必要です。 特に複数回答(マルチアンサー)や、枝番 で分かれているような回答がある調査の 場合、まとめて入力してしまうと、それらは 文字型データとして扱われます。

よく、複数回答データで1セルに「1,2,3」 のように入力されているシートをみる機会 がありますが、このようなデータは事前に 編集しておく必要があります。

	A	В	
1	No	Q1	
2	101	4,5,7	
3	102		1
4	103		5
5	104	1,2,3,5,7	
6	105		6
7	106		5
8	107		5
9	108		5

上)ダメな複数回答の入力例 下) 正しい複数回答の入力例

	A	В	C	D	E	F	G	H
1	No	Q1-1	Q1-2	Q1-3	Q1-4	Q1-5	Q1-6	Q1-7
2	101				1	1		1
3	102	1						
4	103					1		
5	104	1	1	1		1		- 1
6	105						1	
7	106					1		
8	107					1		
9	108					1		

また、1セルを長くして、その内でカンマ区 切りをしている不思議なデータを見ること がたまにあります。一つのセル内が1カラ ムに該当するので、それはカンマのある文 字型データということになってしまいます。 この場合はテキストエディタに貼り付けるな どして、CSVのデータとしてインポートした 方がよいでしょう。

	A	В
1	No,Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8	
2	101,1,3,4,1,55,2,2,1	
3	102,1,3,6.5,2,30,4,3,2	
4	103,1,3,25,5,40,2,2,3	
5	104,1,2,5,3,40,2,1,1	
6	105,1,2,5,3,40,2,2,1	
7	106,1,3,5,2,60,3,2,1	
8	107,1,2,6,2,60,,2,1	
9	108,1,2,3,4,90,2,2,1	

	A	В	C	D	E	F	G	H	1
1	No	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
2	1.01	1	3	4	1	55	2	2	1
3	102	1	3	6.5	2	30	4	3	2
4	103	1	3	2.5	5	40	2	2	3
5	104	1	2	5	3	40	2	1	1
6	105	1	2	5	3	40	2	2	1
7	106	1	3	5	2	60	3	2	1
8	107	1	2	6	2	60		2	1
9	108	1	2	3	4	90	2	2	1

上) 全部文字列となってしまう入力例

下) 正しい入力例

Excelでは、データと同じシートに関数等 で統計値を出力することが可能です。この ようなExcelのシートをインポートした場 合、データ部分と統計値が接していると、 統計値もデータとしてインポートされます。 統計値が1行または1列以上離れていて、 データ部分と接していなければ、データとし てインポートされません。個人的には、集 計などを行ったExcelのシートをインポート する場合は、コピーするなどして集計の ないシートを別途作成した方がいいように 思います。

なお、Excelの関数値は値としてEGに インポートされます。ExcelのSUM関数等 で複数の変数を合計した変数を作成した 場合も、インポートをすれば単なる値となり ます。従って、インポートしたEG上で値を 変更しても再計算されません。

データの編集 ケースの追加(行方向のデータを増やしたい)

ケースを追加したい場合は、データテー ブルで[データ]メニューの[テーブルの追 加」を使います。

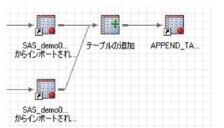


テーブルの追加

この機能を使用する場合は、双方の データがSASデータ形式でなくてはなり ません。Excelデータの場合は、あらかじめ インポートしてSAS形式にしておく必要が あります。



[テーブルの追加]で追加したい変数を選ぶ



追加された状態をプロセスフローで確認

なお、変数名と種類(データ型)は同一で ないと結合されないので注意が必要です。 変数名が違っている場合は別変数として 追加され、同一変数名で種類(データ型) が異なる場合はエラーとなり結合が実施 されません。

この方法を使えば、ファイルやシートに分 かれているデータを結合したい場合に使 用します。例えば男女別にシートを別けて しまったデータなどは、この方法で結合す ることができます。

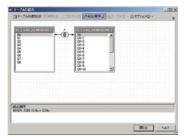
変数の結合 クエリビルダの使用

変数を追加したい場合は、クエリビルダ から行います。アクティブのデータに対して クエリビルダを起動し「テーブルの追加」で 追加したいテーブルを選択します。



[クエリビルダ]の[テーブルの追加]で追加する変数のあるファイルを選択

次に「テーブルの結合」で結合するキー 変数を指定します。同じ変数名なら自動的 に結合指定がされています。

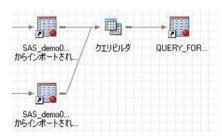


変数名と型が同じなら、自動的に結合のためのキー変数が指定される

ビルダでは新しいデータテーブルを作成 することになるので「データの選択」で新 しく作成されるデータテーブルで使用する 変数を指定します。



使用する変数の指定



追加された状態をプロセスフローで確認

計算した変数の追加

クエリビルダを使うと、例えば既存の複数 の変数の和を変数にしたい場合など、新た に計算した変数を作成することも可能です。 作成するには、クエリビルダの[新しい計算 列の追加]の「新規作成」を選択します。



[新] い計算列の追加]で「新規作成」

任意の計算を行いたい場合は「高度な 式」を選択します。



前述の変数の追加と同様、ビルダは新し いデータテーブルを作成することになるの で、新しいデータテーブルで必要となる変 数を選ぶ必要があります。

作成した変数のエクスポート

Excelからインポートしたファイルは、原 則このプロジェクト内での使用となります。 他のプロジェクトやBase SASで使用する 場合は、SASデータとしてエクスポートする 必要があります。

クエリビルダ等で編集したデータを、 Excel形式にエクスポートすることもできま す。ただし、EG4.2ではエクスポートできる のは2003形式のExcelファイル(拡張子で いえば.xls)までです。Excelファイルは、 2007(拡張子は.xlsx)から1シートのデータ 容量が広がりました。2003までは1シート 65535行×255列でしたが、2007では 1048576行×16384列になりました。EG自 体はメモリのある限りデータは無制限です が、Excelへエクスポートする場合は2003 までの形式(65535行×255列)より行列 方向のどちらかが大きなデータの場合は、 エクスポートできません。このような大きな データは、CSVでもエクスポートができな いことがあります。もしどうしてもエクスポー トしたい場合は、EG上でクエリビルダを使 用し、いくつかのファイルに分割しておく必 要があります。なお、ExcelからEGへのイン ポートは、Excelの2007形式(.xlsx形式)に 対応しています。

余談ではありますが、学生達ヘコン ピュータ操作を教える上でのワンポイント として、ファイルの拡張子が意外と重要な キーワードになります。つまり、今の大学生 はPC操作がある程度できるにもかかわら ず、拡張子の概念が全くなかったりして、 いったい高校の頃はどうしていたのだろう と思うことが多々あります。学校などでずっ と「拡張子非表示」の状態でPCを使用して いるからでしょうか、Macユーザーの学生 の方が、WindowsPCとのファイルのやり とりを経験しているので、よほど拡張子を 気にしていたりします。ファイルを保存する 際に、大学のPCが拡張子を表示する設定 になっていると、見慣れないものだからと 勝手に拡張子を消してしまい、アプリケー ション側から認識されなくしてしまう学生も 後を絶ちません。

エクスポートすると、当然拡張子変更が 行われますが、PC操作に慣れているから といって拡張子の概念があるとは限りま せん。拡張子によってファイル形式を判断 するというのは、当たり前のことではなかっ たりします。まさかとは思っても、慣れて いない人に教えるときは一応確認した方 が良さそうです。

今回はExcelからのインポートを中心に、 クエリビルダの話も少ししました。クエリ ビルダは、SAS形式のデータで使用する ものなので、インポートしたデータに限った 話ではありません。今回ご紹介したような 機能を使用すれば、比較的軽快にExcel とEGとの連携をすることができるでしょう。

次号では、EGの出力形式と、出力を他の ソフトウェアで使用する際のポイントなどに ついてご説明する予定です。

大橋 渉

コラム「SAS四方山話」

千葉大学医学部疾患プロテオミクス寄附研究部門 データベース・インフォマティクス担当 ヤンセンファーマ株式会社 研究開発本部 臨床統計部 統計解析グループ マネージャー 医学博士

第11回 身近な教材でSASを学ぶ ~ヘルプの使い方(2)~

どのようにHelpを利用するのか

前回(夏号)までに、検索の方法と PROCEDUREのFront Pageの各項目(The OO Procedure, Overview, Getting Started, Syntax, Details, Examples) (3 ついて説明させていただきました。今回 からは、これらを用いて学習するための方 法について、筆者なりの推奨方法などを ご紹介させていただきますが、どのレベル および段階までの学習が必要なのかには 個人差がございますので、あくまで「SAS 初心者の方がプログラムを学習や日常業 務を行うために」やってみてはいかがで しょうと思っている方法であることを、あら かじめお断りさせていただきます。もしも この中に使えそうな方法がございましたら、 是非とも積極利用いただければ幸いです。

お気に入りの活用

Helpの中身(説明)は全て英語ですが、 表示は日本語です。いわゆるInternet Explorerのブックマークと一緒で、一度見た ページを登録しておくことが可能です。図1 はFreqのFront Pageを登録した場合の事 例で、登録名も「The FREQ Procedure」の ままですが、もちろん別の名前で登録する ことも出来ます。ここで筆者のお勧めその ①ですが、一度検索したProcedureに 関しては、そのFront Pageをお気に入り 登録しておくことをお勧めします。後述 しますが、今後の業務において何度も参照 するであろう部分についても再度検索する 手間が省けますので、業務の効率化が 見込めるのは間違いありません。筆者の 経験では、LifetestやPhregのような マニアックなものよりも、実はFreqや Univariateのようなものを調べる回数の方 が圧倒的に多かったのです。あくまで筆者 の個人的な感覚ではありますが、比較的 単純と思われるProcedureほど調べる ことが多いように思えます。



図1:まずはProcedureのFront Pageをお気に入りに入れる

Syntax (文法)

実際の業務や研究において何度も足を 運ぶであろう場所がここになります。オプ ションの付与の仕方や順番はもちろんの こと、(筆者の場合は) スラッシュやセミコ ロンの位置までも確認が必要になることも あります。皆様の中にも「あれ、何でこのout の文字は青くならないのだろう」などと思っ ていたところ、スラッシュを入れたら青くなっ た…などというご経験をされた方もいらっ しゃるのではないでしょうか(それ以外にも、 知らずにプログラムを実行したら、エラー Logが示されて、何が足りないかはじめて 分かった…などという例もあるとは思いま すが…)。

図2はSASの中で比較的利用頻度が 高いProcedureの一つであるFregの例で すが、実は案外当たり前と思えることでも つい最近まで知らなかったことがありま した。例えば筆者の場合ですと、当時の先 輩社員からproc freq data=data1; tables col1*col2; run; を基本形として教えてもら いました。これは言わずと知れた変数col1 とcol2のクロス集計ですが、例えばこれを 男女別に行いたいときに、当時の筆者は まずdata1からif文を用いて男性だけ、およ び女性だけのSASデータセットを2つ作成 して、図3左側のプログラムをそれぞれの データセットに対して2回実行していまし た。byを用いる方法を知らなかった筆者 は、Helpを通じて存在を知るまでの約4年 間、本来は1本で済むプログラムを4本実 行していたのです。セミコロンやスラッシュ の位置確認のついででも構いませんの で、眺めてみると案外良いことがあるかも 知れません。参考までにby~weightまで はどの順番で記載しても構いませんが、 セミコロンの有無やスラッシュの位置には 是非ともご注意願います。



図2:各Statementをクリックしてみましょう

data male; set data1; if sex=1; run; data female; set data1; if sex=2; run; proc freq data=male; tables col1*col2; run; proc freq data=female; tables col1*col2; run;

proc freq data=data1; by sex; tables col1*col2; run;

図3:どっちが楽か?

その下のEXACTステートメントですが、 注釈のstatistic optionsは「統計手法の オプション」、computation optionsは「演 算手法のオプション」です。図4のように 集計表に関連する検定手法などの実行の ためのオプションと簡単な説明が一覧表 になっておりますが、さらに詳細を学習 したい方は、Front PageからDetailsを クリックしてみましょう(個人的には、この 表からDetailsの各項目にリンクが張られ ていれば非常に嬉しいと思っております)。 なお、こちらに記載があるあらゆるoptions は、文法的間違いさえなければ記載時に は青色に変化いたします。

Statistic-Option Exact Statistics Comput r s test (for 2×2 tables), simple kappa coefficient test weighted kappa coefficient test BINOMIAL portion tests for one-way tables chi-square goodness-of-fit test for one-way tables; Pearson chi-square, likelihood-ratio chi-square, and CHISO

図4:Optionの簡単な説明

Examples (事例)

Examples: FREQ Procedure

- . Output Data Set of Frequencies
- Frequency Dot Plots
- · Chi-Square Goodness-of-Fit Tests
- Binomial Proportions
- Analysis of a 2x2 Contingency Table
- . Output Data Set of Chi-Square Statistics
- · Cochran-Mantel-Haenszel Statistics
- Cochran-Armitage Trend Test
- . Friedman's Chi-Square Test
- · Cochran's Q Test

Example 3.1 Output Data Set of Frequencies

The eye and hair color of children from two different regions of Europe are re-recorded as cell counts, where the variable Gount contains the number of c-include missing combinations.

The following DATA step statements create the SAS data set Color.

```
Joint, John Strain Stra
```

図5:タイトルをクリックすると実行プログラムを確認できる

ここにはプログラム例が多数掲載されて いますので、自己学習のためには是非とも ご利用いただきたい場所です。図5の左側 の各項目をクリックすると、演習用のデー タ作成のプログラムと集計・解析のための プログラムが表示されますので、プログラ ムの部分をコピーしてエディタに貼り付け て実行してみましょう。皆様の要求に完全 に合致したプログラム例があるかは定か ではありませんが、各Procedureとも基本 的なものから結構難易度の高いものまで 用意されておりますので、全ての項目につ いて、是非とも一度は実行してみることを お勧めいたします。完全に合致したものが 見つからなくとも、Syntax等を参照しなが らご自身の目的に合致したプログラムを作 成するのも、実は大変良い学習方法のひ とつだったりします。英語の意味が分から ないときは迷わず実行して、日本語出力か ら目的に近づけていく…の繰り返しもまた 勉強になります(そこまでのヒマはない方 が大半ではあると存じますが…)。

図5の右側のリンク先には、元々実行 結果やoutputの見方も解説されておりま すが、英語が苦手という方は、プログラム を実行することで日本語の出力による outputを見ることができます。実際に筆者 はこれらExamplesを用いた授業を行い、 元々の解説と対比することにより、学生の 統計的英語表現の学習に用いました。 英語表現を身につけることを目的として いたわけではなく、単純にSASの学習の ための例題として用いただけでしたが、 実は筆者も予期しない効果があったと、 その時に改めて思った次第です。

英語は苦手…でも

まずはExamplesのプログラムを実行する ことで、各Procedureの基本的機能が見え て来ます。見たこともなかったような解析 方法でも、とりあえず実行してみることで理 解につながることもあると思います。英語 が苦手でなかなかHelpは使っていなかっ た…という方でも、実際には実行結果を日 本語で確認して見ると案外すんなりと理解 できたりもしています。英語が苦手…を自認 されていらっしゃる方こそ、Help学習をして みるのもよろしいのではないでしょうか。

次号、さらに別のProcedureに関する お話を予定しております。

(SASユーザー総会から) EGに関する考察

筆者の夏号における「Enterprise Guide (以下EG)を活用しましょう」という提案に つきまして、先日、ユーザー会でお会いさ せていただいた方と、筆者の著書に関する 話をさせていただいたのですが、ある方か ら「初心者の教育にはEGが有効ではない か」とのご意見を頂きました。また、その場 の別の方からは「いや、やはりプログラム あってこそのSASであり、安易にEGの利用 を推奨するべきではない!」とのご意見も 頂きました。「やはりプログラムこそSAS」と いうようなお話は以前にも頂いており、どう やらSASユーザーの中にはEGの利用に 否定的な方々もいらっしゃるようですので、 (本論から乖離してしまいますが)ここで EGについて少しお話しさせてください。

結論から申しますと、プログラムがあって こそのSASであるという部分に関して全く

異論はございません。実際に筆者も拙著 (「統計を知らない人のためのSAS入門 (オーム社)」)の中で「EGではSASの全て の機能を表現できない」ことは記載して おりますし、プログラムの組み合わせに より表現できる解析方法(例:一部の多重 比較法など)も少なくないことは理解して いるつもりです…が、プログラミング以外 は認めないという姿勢には問題があると 思えてならないのです。

筆者以外にも、たとえば國學院大学の 高柳良太先生はEGの教育利用について 同じ夏号で、筆者の職場の先輩である ヤンセンファーマの森美知代氏は、2010 年6月のライフサイエンスフォーラムにて演 題「新薬開発プロセスのパラダイムシフト」 の中で、EGを利用した業務の効率化・ 標準化について語っておられます。それ ぞれ、教育現場においてはプログラミング への足がかりとして(高柳先生)、業務に おいては工数の削減や標準化のために (森氏)、EGの利用は有意義であることを 主張されておられますが、これらのお考え には筆者も深く同意致します。どのような 場面であっても、EGの利用がプログラ ミングよりも優れている場面があれば、 積極的に取り入れられても良いと思われ ますし、必ずしも全ての場面において、プロ グラミングにこだわり続ける必要性はない のではないでしょうか。

「プログラミング以外は認めない…」。 筆者の感覚ですが、実は多くの方々がEGを 使った経験がないまま、そのような主張を されているのではないでしょうか。「EGよりは プログラミングに慣れているからその方が 早い。今更EGを覚える必要はない」「EGは プログラミングができないユーザーのため のもの」などという主張も一部ではあるよう ですが、必ずしもプログラミングだけが SASではありません。価値観も利用の可否 もそれぞれではありますが、実際に多くの 皆さまからEGの実際的な効果も示されて おりますので、例えば熟練のプログラマー がEGに目を向けて、適材適所で利用して みても良いのではないかと思います。

そろそろ…プログラム至上主義からの パラダイムシフトも必要です。

Q & A



-) デンドログラム、樹形図の表示に関して
- マクロ変数の値により、別のマクロ変数に異なる値を 格納する方法
- 半角カタカナを全角カタカナへ変換する
- プログラム内のLRECLオプションを一括で設定する方法
- バッチ処理でエラーが生じた後の、再実行について





TREEプロシジャにてデンドログラム、樹形図を描画しま したが、オブザベーションに対するラベルが重なって います。重ならないように表示できますか?

分析に含まれているオブザベーションの数が多い場合、 ラベルが重なり、見づらい表示となることがあります。この場 合、PROC TREEステートメントにて、HPAGES= VPAGES=

オプションを指定し、2ページ以上にてグラフを表示します。また、 GOPTIONSステートメントでHTEXT=オプションを用い、フォントの大き さを変更します。

例

GOPTIONS HTEXT=.8; /* フォントのサイズを指定 */ PROC TREE DATA=treedata HPAGES=2 VAPGES=2; /* 複数ページにて表示 */ RUN:

また、グラフのサイズを大きく設定した上で、外部ファイルに出力する ことも有用です。

例

/* 出力先を指定 */ FILENAME sastemp 'C:\forall test.gif'; /* XMAX=、YMAX= にてグラフサイズを大きく指定 */ GOPTIONS RESET=ALL DEV=gif GSFNAME=sastemp XMAX=50cm YMAX=50cm: PROC TREE DATA=treedata; RUN:



マクロ変数の値により、別のマクロ変数に異なる値を格納 したいと考えています。

マクロ変数 macv に 0 が格納されていれば、マクロ変数 judge に False を格納 マクロ変数 macv に 1 が格納されていれば、マクロ変数 judge に True を格納

マクロ変数の作成のみであるため、できるだけ簡易に値を格納したい のですが、何かよい方法はありませんか?



今回のようなケースでは、SAS 9.1.3より新しく追加された IFC関数と、既存の%SYSFUNCマクロ関数を使用いただく ことで、プログラムを簡素化することができます。

例

%LET macv=0;

%LET judge=%SYSFUNC(IFC(&macv=1,TRUE,FALSE)); /* IFC 関数 */ %PUT &judge;



半角カタカナを全角カタカナへ変換する方法はありま すか?

SAS 9.2より新しく追加されたKPROPCASE関数を 使うことによって、半角カタカナを全角カタカナに 変換することができます。今回の場合は、第2引数に

"HALF-KATAKANA、FULL-KATAKANA"オプションを指定します。 その他のオプションについては以下のドキュメントをご覧ください。

SAS 9.2 NLS Reference Guide: KPROPCASE Function

http://support.sas.com/documentation/cdl/en/nlsref/61893 /HTML/default/a003284797.htm

例

```
DATA sample;
 LENGTH word1 word2 $50;
 INPUT word1 $;
                 /* 半角カタカナ を 全角カタカナへ変換 */
 word2 = KPROPCASE(word1, 'HALF-KATAKANA,
FULL-KATAKANA');
DATALINES:
サスインスティチュートジャパン
テクニカルサポート
ビジネス・アナリティクス
RUN:
```



使用しているプログラムでは、外部ファイルの入出力が多 数あります。全てのFILE、INFILEステートメントに適切な LRECLオプションを指定するのが大変なのですが、一括

して余裕を持たせた長さのLRECLオプションを指定する方法はあり ますか?





SAS 9.2より機能拡張としてLRECLシステムオプションが追 加されました。LRECLオプションのデフォルト値は256バイト ですが、LRECLシステムオプションを指定することにより、

実行するプログラム内のLRECLオプションの値が全て変更されます。

[構文]

LRECL = N

N:処理するファイルの最大レコード長を指定します。 指定できる値は、1から32767までとなり、デフォルト値は256です。

OPTIONS LRECL = 1000;

LRECLシステムオプションの詳細に関しましては、次のオンラインヘルプ をご参照下さい。

[LRECL= System Option]

http://support.sas.com/documentation/cdl/en/Irdict/63026 /HTML/default/a003160472.htm



非常に多数のステップからなるSASプログラムをバッチ で実行しています。エラーが発生し再実行した場合、 プログラムの初めから処理されるため、デバッグに時間

を要してしまいます。なにか効率よくデバッグするような方法はある でしょうか?



SAS 9.2からバッチ処理でエラーが発生した際、エラー が発生したステップから再実行する'Checkpoint'、 'Restart'という機能が新たに追加されました。

Checkpoint機能は、エラーが発生したステップの情報をSASが 内部で保持するSASライブラリへ保存します。次にRestart機能 では、エラーが発生したステップから再実行します。

Checkpoint機能を有効にするには、STEPCHKPTオプション、 Restart機能を有効にするにはSTEPRESTARTをSAS起動時に 指定します。ただし、STEPCHKPT、STEPRESTARTオプションを有 効するには、併せてNOWORKTERM、NOWORKINITオプションの 指定が必須となります。

また、ERRORCHECK STRICTオプションを指定することで検知する ERRORの種類を増やし、ERRORABENDオプションを指定する ことでERROR時に処理を中断させます。

sas -sysin 'c:\mysas\myprogram.sas' -stepchkpt -steprestart -noworkterm -noworkinit -errorcheck strict -errorabend

なお、この例ではSASが内部で保持するSASライブラリを使用する方 法ですが、保存先を任意の恒久SASライブラリを使用こともできます。 使用及び設定方法の詳細は、次のオンラインヘルプをご参照ください。

[Checkpoint Mode and Restart Mode]

http://support.sas.com/documentation/cdl/en/Ircon/62955/ HTML/default/a000993436.htm#a003133413



SAS 9.2でのMetadata Server環境のバックアップを 検討しています。ディスクやディレクトリのバックアップは 行いますが、SASで用意しているバックアップ機能は

ありますか?



最新のSAS 9.2ではSAS管理コンソールからのGUI操 作により、バックアップやリストアの実行、バックアップ 処理のプログラムの作成/入手が可能です。

次の手順にてバックアップウィザードを起動し、ご確認ください。

- 1. 管理コンソールに無制限ユーザーなど管理者権限のあるIDで ログインします
- 2. [環境管理] => [メタデータマネージャ] から [メタデータユーティ リティ] =>[バックアップ & リストア]を選択します
- 3. [アクション] => [新しい定義]を選択すると、バックアップウィザー ドが起動します
 - ※各画面の詳細説明については画面右下の[ヘルプ]を押す ことで確認できます。
 - ※この際、実際のサーバー上で有効なIDとパスワードの入力が 必要となる場合があります。

バックアップに関する詳細については、次のマニュアルにてご確認 願います。

PRODUCT DOCUMENTATION SAS Intelligence Platform

http://support.sas.com/documentation/onlinedoc/intellpla tform/index.html

[Administration Documentation] => [System Administration Guide] [PART4 Backing Up and Restoring Your System]

SAS 9.1.3では、%OMABAKUPマクロを利用したプログラムを独自 に作成いただくことでバックアップを取得することが可能です。 こちらはSAS Technical News 2005年夏号で特集しております。

STN

SAS Training

SAS トレーニングのお知らせ

SAS® Base Programming for SAS® 9と SAS® Advanced Programming for SAS® 9の日本語版認定試験

2010年9月1日よりSASのプログラミング系認定試験である、SAS Base Programming for SAS 9(以下:Base試験)とSAS Advanced Programming for SAS 9(以下:Advanced試験)が日本語で受験できるようになりました。下記Webページよりコンテンツ(試験範囲)、対応コース、試験予約の方法などの詳細情報にアクセス可能です。

www.sas.com/jp/training/certify/index.html

今回は受験の心得とBase試験の練習問題を掲載します。

概要

Base試験、Advanced試験共にBase SASを対象とした試験です。 試験はコンピューター上で受験し、問題形式は4択式です。受験の順番は決まっていませんが、Base試験は1年程度のSAS経験、Advanced試験は3年程度のSAS経験を目安としています。以下はSAS未経験から短期間でBase試験に合格した、とある勤勉な米

国SAS Institute社員(愛称:まーくん)のレポートをもとに5つの受験の心得を紹介します。まーくんはSAS研修受講以外にも、勤務時間外の不断の努力(毎週10時間×5週間の勉強)により3カ月で合格という偉業を達成しました。皆様も努力やコツで準備期間は短縮可能と思います。



まーくん推奨!受験の心得

① 試験範囲の熟知

上記のWebページより、試験範囲をご確認ください。Base試験、Advanced試験共に試験範囲は広いです。ご自身の経験上あまり使用していない部分が明確に分かっていると勉強の計画を立てやすいと思います。対象となる研修コース(後述)の受講により、全体的な範囲がつかめます。

② 構文の理解

まずは試験範囲に該当するプロシジャ、ステートメント、オプションの基本的な構文を再確認する必要があります。ま一くんは単語カードを使って構文を覚えました。Base試験であればポイント解説講座(後述)にて、基本構文をまとめています。

③ 内部動作の理解

DATAステップであれば、コンパイルフェーズと実行フェーズ、入力 バッファやプログラムデータベクトル(PDV)の動作など、内部構造を 理解しましょう。試験合格には内部動作の理解は必須と言えます。

④ プログラミングの経験

そして、内部動作の理解にはSASプログラミング経験が必要です。データエラー発生時などは、入力バッファやPDVの内容がログに出力されるため、正しい記述/間違った記述を行った際のログやアウトプットなどを確認することで理解が深まります。

⑤ 期日の決定

受験勉強はある程度の期限を決めていないと継続的な勉強を行いづらいと思います。思い立ったが吉日、1年間有効な受験チケット(後述)を購入して期限を決めておくのも有効かもしれません。2010年9月現在、1枚より割引を行っています。

練習問題

下記のプログラムをサブミットするとERRORメッセージが出力されます(コードの左は行番号)。

```
data x;
       name = 'yamada';
                           * 文字変数;
       ageC = '30';
3
                           * 文字変数:
       ageN = 30;
                           * 数值変数;
5
       result1 = ageC + 1;
       result2 = name !! ageN;
6
       result3 = ageC > 20;
8
    run;
9
10
    data y;
11
       set x:
12
       where ageC > 20;
13
    run:
```

ERRORは何行目で発生しますか。

A. 5行目 :result1 = ageC + 1; B. 6行目 :result2 = name !! ageN; C. 7行目 :result3 = ageC > 20; D. 12行目 :where ageC > 20;

解説

これは変数タイプの自動変換に関する問題です。

SAS変数のタイプは数値と文字の2種類のみです。一般的な2種類 の使い分けは、「記述統計量を求めるなど、算術演算を行いたい かどうか?」で数値変数にするかを決定します。

また、DATAステップでは、ステップの先頭から見て初めてその変数 が登場する段階で変数タイプと長さが決定します。LENGTHステー トメントなどでタイプと長さを明示していない場合は文脈に応じて 自動的に決まります。

従って、2行目と3行目は文字定数を割り当てているため、変数 nameとageCは文字変数、4行目は数値定数を割り当てているため ageNは数値変数となります。

これ以降選択肢として登場するステートメントを検証していきます。

result1 = ageC + 1; 6 result2 = name !! ageN; result3 = ageC > 20;

5行目では文字変数ageCを数値定数1で加算し、変数result1に 割り当てています。

DATAステップ内の算術演算で、演算対象が文字であった場合は 自動的に数値に変換を行います。入力形式を明示しなくても数値と して判定可能な、いわゆる「標準数値データ」の場合は数値への自 動変換が問題なく行われ、算術演算が行われます。従って、result1 には数値31が格納されます。

ちなみに算術演算子により自動変換が行われると、

NOTE: 以下の箇所で文字値を数値に変換しました。

という形式のNOTEメッセージが発生します。

6行目のように、文字列の連結を行う演算子(||や!!)で、数値を対象と した場合は、数値を文字に変換する自動変換が働きます。この場合、

NOTE: 以下の箇所で数値を文字値に変換しました。

というNOTEメッセージが出力されます。

7行目も5行目と同じです。つまり、比較演算子 > は数値を対象とした 演算子であるため、文字変数ageCの値は数値に自動変換されて比 較式が評価されます。SASの比較式の実行結果は、真の場合は1、 偽の場合は0という数値になります。従って変数result3は数値変数 で、ここでもERRORは発生しません。

where ageC > 20;

12行目も7行目と同じ「ageC > 20」の比較を行っているため、自動変換 が働くように思うかもしれません。しかしPDVにデータを読み込む前 に評価される、WHEREステートメントでは演算子を使用しても自動変 換は行われません。以下のERRORメッセージが出力されます。

ERROR: WHERE 式の演算子には互換性のあるタイプの変数が必要です。



正解:D



おわりに

今回はBase試験の試験範囲から問題をピックアップしました。 SAS研修では単なる記述方法の解説だけでなく、内部動作や 仕組みも含めた解説を多く行っているため、皆様のSASスキルの 向上に役立つものと自負しております。

今回日本語化された認定試験が、SASユーザーの皆様のスキル マップ、キャリアパスを考える上での一つのよい目標となればと 思います。

また、今後はPlatform系の3試験(Platform Administrator、Data Integration Developer、BI Content Developer)も日本語化を 予定しています。随時Webページには最新情報を更新していきます ので、SAS Enterprise BI ServerやSAS Data Integration Serverの環境を利用されている方もご期待ください。

関連コース・受験チケットのご案内

今回の練習問題(Base試験)の対象コースはこちらです。

- 1. SASプログラミング1:必須要素
- 2. SASプログラミング2: データ加工テクニック

「SAS認定プロフェッショナル バリュー・パッケージ」

試験合格に必要な、上記コースをセットにしたお得な割引プランで す。年内に2コース受講いただくと、受験チケット(18,900円(税込) 相当)を差し上げます。

www.sas.com/jp/training/certset.html



「SAS認定プロフェッショナル SAS Base Programmer for SAS 9ポイント解説講座」

SASプログラミング経験者向けの、資格取得への総仕上げ講座で す。模試問題とその解説を通じて、重要な技術トピックを復習でき、 効率的に受験準備できます。

今回日本語試験の提供開催を記念し、下記の開催日に限り、受験 チケット(18,900円(税込)相当)を付け、15%OFFの特別価格で ご提供します。

開催日: 2010年11月26日(金) 特別価格: 65.152円(税込)

(定価:76,650円(税込)より15%OFF)

www.sas.com/jp/training/course/cert_seminar.html



「受験チケット」

受験チケットは、18.900円(税込)で提供するSAS認定資格すべて に利用でき、購入後1年間有効です。また、支払い処理が購入時に 完了しますので、受験毎の支払いが不要となり、手間の削減にも なります。

また、組織内で資格取得に挑戦する場合は、今期の予算で購入 し、来年度4月からの人材育成にお役立ていただくことができます。 日本語試験の提供開催を記念し、購入単位1枚から割引を設定 しました。この機会に、ぜひ、受験チケットの購入をご検討ください。

www.sas.com/jp/training/certify/order.html



特別トレーニング・コースの開催のご案内

● 「医学研究におけるベイズ統計学: SAS9.2による アプローチ」 コース (2日間)

[日 程]

2010年11月18日(木)~19日(金) 10:00 ~ 17:00 (東京会場)

147,000円(税込)/※チケットのお取り扱いはありません。 [受講対象]

ベイズ統計を臨床試験のデザインに用いる方でSASを用いてベイズ 流データ解析を実施したい方

[学習内容]

ベイズ統計の基礎を解説し、ベイズ統計の実際の臨床試験への 適用事例を紹介します。

SASによる実際の解析手順、結果の解釈を説明します。

[Web Page]

www.sas.com/jp/training/course/bayesian_statistics_sas9.2.html

● 「SASによる観察研究データの統計解析 【後編】 」 コース (2日間)

[日 程]

2010年12月13日(月)~14日(火) 10:00 ~ 17:00 (東京会場)

「価格]

147,000円(税込)/※チケットのお取り扱いはありません。 【注意】本コースは、英文書籍を使用し日本語で説明を行います。 また、前編と後編に分かれていますので内容をお確かめ下さい。 [受講対象]

SASシステムによる臨床・疫学研究データの統計解析に従事している方(上級者)

[学習内容]

ランダム化をともなわない臨床・疫学研究データの統計解析では、 群間の比較可能性が崩れているために生じる「交絡」と呼ばれる 現象を調整することが常に要求されます。交絡調整法としては、 層別解析と回帰モデルに基づく方法の2つが古くから知られていますが、近年、傾向スコア(propensity score)に基づく方法やいくつかの因果モデルに基づく方法も応用例が増えてきています。上記テキストは、傾向スコアを用いた交絡調整法、周辺構造モデル(marginal structural model)や構造ネストモデル(structural nest model)といった因果モデルに基づく交絡調整法などについて、SASによる実際の解析手順、結果の解釈を主眼として書かれています。

【後編】コースでは、下記テキストの5章、9章、10章を学習する予定です。5章は欠測データの解析、9章は周辺構造モデル、10章は構造ネストモデルによる解析です。

[テキスト]

『Analysis of Observational Health Care Data Using SAS』(SAS Press)
※受講申込者は、トレーニング申込とは別に事前にご購入ください。
(上記書籍を既にお持ちの方は除きます。)

[書籍購入方法]

Amazon.co.jp(http://www.amazon.co.jp/)でご購入下さい。 [Web Page]

www.sas.com/jp/training/course/k_stat_2.html

SAS Institute Japan株式会社では、今後も多岐にわたったトレーニングコースを追加していく予定です。

コース内容・日程等の詳細は、順次弊社Webサイトに公開しますので、以下のURLをご参照ください。

http://www.sas.com/jp/training/

その他、トレーニングに関する情報については、上記のURLをご参照 いただくか、下記トレーニング担当までお問い合わせください。

トレーニング担当

T E L: 03-6434-3690 F A X: 03-6434-3691

E-mail: JPNTraining@sas.com

Latest Releases

最新リリース情報

PCプラットフォーム

Windows版 SAS 9.1.3 / 9.2 64-bit Windows(Itanium)版 SAS 9.1.3 / 9.2

メインフレームプラットフォーム

IBM版(OS/390,z/OS) SAS 9.1.3 / 9.2

UNIXプラットフォーム

SunOS/Solaris版 SAS 9.1.3 / 9.2 HP-UX版 SAS 9.1.3 / 9.2 HP-UX(Itanium)版 SAS 9.1.3 / 9.2 SAS 9.1.3 / 9.2 AIX版 Linux(Intel) 版 SAS 9.1.3 / 9.2 IBM版(OS/390,z/OS) SAS 9.1.3 / 9.2



SAS Technical News入手

SAS Technical Newsは、右記のURLから入手できます。

http://www.sas.com/jp/periodicals/technews/index.html

発行:SAS Institute Japan株式会社



■テクニカルニュースに関するお問い合わせ先

TEL:03-6434-3680 FAX:03-6434-3681



POWER

SAS Institute Japan株式会社

本社 〒106-6111 東京都港区六本木6-10-1 六本木ヒルズ森タワー 11F Tel 03(6434)3000 Fax 03(6434)3001

大阪支店 〒530-0004 大阪市北区堂島浜1-4-16 アクア堂島西館 12F Tel 06(6345)5700 Fax 06(6345)5655

www.sas.com/jp