

SAS Technical News

Autumn 2003

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

CONTENTS

- 1 SASプログラムのエラー発見・解消テクニック
- 9 Q&A
- 13 SASトレーニングのお知らせ
- 14 新刊マニュアルのご紹介
- 16 最新リリース情報

特集

SASプログラムの エラー発見・解消テクニック

1. はじめに

SASに限らず、プログラムを書いたことがある方なら、プログラムがエラーになり意図したとおりに動かないといった経験が必ずあると思います。プログラムがエラーになった場合、避けて通れない道、それは『プログラムのデバッグ』です。そこで、本特集ではDATAステップを中心に、エラー発見に役立つテクニックをご紹介します。

2. ログとメッセージ

ご存知のように、SASはプログラムを実行するとログを生成します。これには、以下の情報が含まれています。

実行されたプログラム
作成されたSASデータセットに関する情報(変数やオブザベーション数など)
プログラム実行中に発生した警告・エラーメッセージ
プログラム実行に要した時間

プログラムが正常に実行されない場合(「WARNING(警告)」や「ERROR(エラーメッセージ)」が表示された場合)、それらのメッセージの内容を確認することにより、原因やプログラム修正のためのヒントを見つけることができます。また、プログラムの実行後は、意図したとおりに実行されたかどうかを検証するために、ログを確認する必要があります。

3. エラータイプ

エラーが発生したプログラムを修正するには、まず発生したエラーのタイプを知ることが非常に重要です。SASプログラムの実行時に発生するエラーには、以下の4つのタイプがあります。

構文エラー(Syntax error)

ステートメントがSASプログラムの構文規則に沿っていない場合に起こります。

セマンティックエラー(Semantic error)

ステートメントの引数として指定した文字列が、構文規則上は正しくても、その引数の使用意図としては間違っているような場合に起こります。

実行時エラー(Execution-time error)

コンパイル済みのプログラムを実行する場合に起こるエラーです。

(例: 0による除算)

データエラー(Data error)

ステートメントは正しいものの、データ値がステートメントに適さない場合に起こります。

それぞれ、サンプルプログラムとともに具体例を紹介します。

3.1 構文エラー(Syntax error)

以下のサンプルは、SASデータセットsashelp.classから男性のデータのみを抽出し、変数weightの出力形式(フォーマット)に8.2を指定しているプログラムです。しかし、FORMATステートメントに不要な「=」があるため、構文エラーが発生しています。

サンプルプログラム

```
DATA temp;
  SET sashelp.class(WHERE=(sex='M'));
  FORMAT=weight 8.2;
RUN;
```

エラーログ

```
28 DATA temp;
29 SET sashelp.class(WHERE=(sex='M'));
NOTE: SCL プログラムのテキスト
30 FORMAT=weight 8.2;
-----
22
ERROR 22-322: 構文エラーです。次の 1 つを指定してください :
, !, &, *, **, +, -, /, <, <=, >, =, >, >=,
AND, EQ, GE, GT,
LE, LT, MAX, MIN, NE, NG, NL, OR, ^=, |, ||, ~=.
```

3.2 セマンティックエラー(Semantic error)

下記は、SASデータセットsashelp.classから女性のデータのみを抽出して、SASデータセットgirlsを作成するプログラムです。SASプログラムの構文に間違いはなく、コンパイル時にエラーは発生しませんが、OUTPUTステートメントで指定しているSASデータセット名とDATAステートメントで指定しているデータセット名が一致しないために、セマンティックエラーが発生します。

サンプルプログラム

```
DATA girls;
  SET sashelp.class;
  IF sex='F' THEN OUTPUT women;
RUN;
```

エラーログ

```
48 DATA girls;
49 SET sashelp.class;
NOTE: SCL プログラムのテキスト
50 IF sex='F' THEN OUTPUT women;
-----
455
ERROR 455-185: DATA ステートメントにデータセット名が指定されていません。

51 RUN;
```

3.3 実行時エラー(Execution-time error)

下記は、購入商品、支払金額、購入数をSASデータセットpriceとして作成するプログラムです。3オブザベーション目にて0による除算が行なわれているため、「uprice=totprice/unit;」は実行されません(実行時エラーが発生します)。

サンプルプログラム

```
DATA price;
  INPUT item $ totprice unit;
  uprice=totprice/unit;
CARDS;
牛乳 890 5
たまご 1200 10
納豆 110 0
;
RUN;
```

エラーログ

```
78 DATA price;
79 INPUT item $ totprice unit;
80 uprice=totprice/unit;
81 CARDS;

NOTE: 行 80 カラム 20 で 0 による割り算がありました。
RULE: -----1-----2-----3-----4
84 納豆 110 0
item=納豆 totprice=110 unit=0 uprice=. _ERROR_=1 _N_=3
NOTE: 以下の箇所では演算式を計算できなかったため、結果を欠損値に
設定しました。
(回数) (行: カラム)
1 80:20
NOTE: データセット WORK.PRICE は 3 オブザベーション、
```

4 変数です。

NOTE: DATA ステートメント 処理 :

処理時間	0.07 秒
CPU 時間	0.07 秒

3.4 データエラー(Data error)

社員番号と誕生日をSASデータセットemp_febとして作成するプログラムです。存在しない日付(1961年2月29日)がデータに含まれているために、データエラーが発生します。

サンプルプログラム

```
DATA emp_feb;
  INPUT empno $ birthday;
  INFORMAT birthday yymmdd10.;
  FORMAT birthday yymmdd10.;
CARDS;
00070 1948/02/15
00100 1960/02/29
00116 1961/02/29
00249 1976/02/02
;
RUN;
```

エラーログ

```
12 DATA emp_feb;
13 INPUT empno $ birthday;
14 INFORMAT birthday yymmdd10.;
15 FORMAT birthday yymmdd10.;
16 CARDS;

NOTE: birthday に対して、無効なデータが行 19 カラム 7-16
      にあります。
RULE:  ---+---1-----2-----3-----4
19 00116 1961/02/29
empno=00116 birthday=. _ERROR_=1 _N_=3
NOTE: データセット WORK.EMP_FEB は 4 オブザーベーション、2
      変数です。
NOTE: DATA ステートメント 処理 :
      処理時間 0.05 秒
      CPU 時間 0.05 秒

21 ;
22 RUN;
```

3.5 ロジックエラー

これまでご紹介した4タイプのエラー以外に、SASでは見つけられないエラータイプがあります。それがロジックエラーです。たとえば、「年齢が11歳以上の場合に変数flagに1をセットする」プログラムを作成したとします。しかし、下記のサンプルプログラムの条件式は、「年齢が11歳より大きい場合」と指定されています。構文エラーなどは発生しませんが、条件式が適切でないため、プログラムは意図したとおりに動きません。

サンプルプログラム

```
DATA logic;
  SET sashelp.class;
  IF age > 11 THEN FLAG=1;
RUN;
```

4 SASプログラムのエラーを発見する際に役立つテクニック

SASでは、次の4つのコンポーネントをデバッグに利用することができます。これらを組み合わせるにより、検証時のデータ損失の予防や、エラー発生個所の特定を容易にすることができます。

オプション(システムオプション・データセットオプション)
ステートメント
関数およびCALLルーチン
自動変数

サンプルプログラムを利用して、具体的に紹介します。

4.1 オプション(システムオプション・データセットオプション)を利用する

4.1.1 構文エラー(Syntax error)を見つける

OBS=オプションとNOREPLACEオプションを同時に利用すると、構文エラーを見つける際に非常に便利です。以下のプログラムは、SASデータセットtest.prdsaleからデータを読み込み、売り上げ達成率を変数salespct (salespct=predict/actual;)として追加して、その結果をSASデータセットtest.prdsale2とするものです。

サンプルプログラム

```
PROC CONTENTS DATA=test.prdsale;
RUN;

DATA test.prdsale2
  SETS test.prdsale;
  salespct=predict/actual;
  label salespct='売り上げ達成率';
RUN;

PROC CONTENTS DATA=test.prdsale;
RUN;
```

実行ログ

```

~ 省略 ~

3   DATA test.prdsale2           ①
4       SETS test.prdsale;       ②
5       salespct=predict/actual;
6       label salespct='売上げ達成率';
7   RUN;

NOTE: 変数 actual は初期化されていません。
NOTE: 変数 predict は初期化されていません。
NOTE: 欠損値を含んだ計算により、以下の箇所で欠損値が生成されました。
      ( 回数 )( 行 : カラム )
      1 5:20
NOTE: データセット TEST.PRDSALE2 は 1 オブザベーション、3 変数です。
NOTE: データセット WORK.SETS は 1 オブザベーション、3 変数です。
NOTE: データセット TEST.PRDSALE は 1 オブザベーション、3
      変数です。           ③

~ 省略 ~

```

実行ログ①のDATAステートメントを終了させる「;(セミコロン)」がなく、また、②のSETステートメントの記述が「SETS」になっているため、意図した結果が得られていません。このサンプルプログラムで最も注目すべき部分は③です。DATAステップ前後にあるCONTENTSプロシジャの結果を確認してみましょう(表1)。

表1 CONTENTSプロシジャの結果(抜粋)

	DATAステップ実行前	DATAステップ実行後
オブザベーション数	1440	1
変数数	10	3

DATAステップの構文エラーにより、データセットtest.prdsale が1オブザベーション、3変数のデータセットとして上書きされてしまっています。このプログラムを修正して再度実行するには、上書きしてしまったデータセットtest.prdsaleを再作成しなければなりません。このような現象を防ぐためには、OBS=0およびNOREPLACEオプションを同時に利用すると便利です。これらのオプションを指定すると、DATAステップは実行されるので、構文エラーがある場合にはそれがログに出力されますが、データセットや外部ファイルからデータを読み込むことはありません。また、永久ライブラリのデータセット上書きを防ぐことができます。

サンプルプログラム

```

OPTIONS OBS=0 NOREPLACE;          /* For Debug */

PROC CONTENTS DATA=test.prdsale;
RUN;

DATA test.prdsale2
  SETS test.prdsale;
  salespct=predict/actual;
  label salespct='売上げ達成率';
RUN;

```

```

PROC CONTENTS DATA=test.prdsale;
RUN;

```

実行ログ

```

~ 省略 ~

NOTE: データセット TEST.PRDSALE2 は 1 オブザベーション、
      3 変数です。
WARNING: NOREPLACEオプションのため、データセットTEST.PRDSALE2
を置き換えていません。
NOTE: データセット WORK.SETS は1オブザベーション、3 変数です。
NOTE: データセット TEST.PRDSALE は 1 オブザベーション、 3
      変数です。
WARNING: NOREPLACE オプションのため、データセット
      TEST.PRDSALE を置き換えていません。
NOTE: DATA ステートメント 処理 :
      処理時間           0.02 秒
      CPU 時間           0.02 秒

~ 省略 ~

```

表2 CONTENTSプロシジャの結果(抜粋)

	DATAステップ実行前	DATAステップ実行後
オブザベーション数	1440	1440
変数数	10	10

実行ログとCONTENTSプロシジャの結果から、今度はデータセットtest.prdsaleが上書きされていないことが確認できます(OBS=0の指定があっても、CONTENTSプロシジャやDATASETSプロシジャのようにオブザベーションを読み込まないプロシジャは、通常どおり実行されています)。なお、設定をデフォルトの状態に戻すには、「OPTIONS OBS=MAX REPLACE;」と記述します。

4.1.2 データをサブセット化して処理結果を確認

大量のデータを利用するプログラムを作成する場合、そのプログラムの移動確認にも必然的に時間がかかります。その際、FIRSTOBS=データセットオプション、およびOBS=データセットオプションでデータをサブセット化してからプログラムの実行結果を確認すれば、実行時間が短くなり、スムーズにプログラムの検証ができます。以下のプログラムは、合計ステートメントを利用して変数totactにactualの累計を格納するようになっています。このとき、SASはデータセットsashelp.prdsaleから10オブザベーション(11オブザベーション目から20オブザベーション目、同プログラム実行ログ①部分を参照)しか読み込まないため、処理は短時間で終了します。

サンプルプログラム

```

DATA prdsale;
  SET sashelp.prdsale(FIRSTOBS=11 OBS=20);
  totact+actual;
RUN;

PROC PRINT DATA=prdsale;
  VAR actual totact;

```

```
RUN;
```

実行ログ

```
1 DATA prdsale;
2 SET sashelp.prdsale(FIRSTOBS=11 OBS=20);
3 totact+actual;
4 RUN;
```

NOTE: データセット SASHELP.PRDSALE から 10 オブザベーションを読み込みました。 ①

NOTE: データセット WORK.PRDSALE は 10 オブザベーション、11 変数です。

NOTE: DATA ステートメント 処理 :

処理時間	0.59 秒
CPU 時間	0.03 秒

```
5
6 PROC PRINT DATA=prdsale;
7 VAR actual totact;
8 RUN;
```

NOTE: データセット WORK.PRDSALE から 10 オブザベーションを読み込みました。

NOTE: PROCEDURE PRINT 処理 :

処理時間	0.34 秒
CPU 時間	0.01 秒

4.2 ステートメントを利用する

4.2.1 データエラーを判別し、ログをカスタマイズする

プログラムは正しくても、不正なデータによって処理が悪影響を受ける場合があります。このようなことを防ぐために、データを判別してログに任意のエラーメッセージを出力することができます。以下のプログラムは、2月生まれの従業員のデータをSASデータセットemp_febとして作成するプログラムです。この際、不正なデータがあった場合はERRORステートメントを利用してログにメッセージを出力し、そのデータを削除します。

サンプルプログラム

```
DATA emp_feb;
  INPUT empno $ birthday;
  INFORMAT birthday yymmdd10.;
  FORMAT birthday yymmdd10.;
  IF MONTH(birthday)^=2 THEN DO;
    ERROR 'ERROR: 以下の不正データを削除します';
    DELETE;
  END;
CARDS;
00116 1948/02/15
00100 1960/02/29
00116 1961/03/01
00249 1976/02/02
;
RUN;
```

実行ログ

```
1 DATA emp_feb;
2 INPUT empno $ birthday;
3 INFORMAT birthday yymmdd10.;
4 FORMAT birthday yymmdd10.;
5 IF MONTH(birthday)^=2 THEN DO;
6 ERROR 'ERROR: 以下の不正データを削除します';
7 DELETE;
8 END;
9 CARDS;
```

ERROR: 以下の不正データを削除します

RULE: ----+----1-----+----2-----+----3-----+----4

9 00116 1961/03/01

empno=00116 birthday=1961-03-01 _ERROR_=1 _N_=3

NOTE: データセット WORK.EMP_FEB は 3 オブザベーション、2 変数です。

~ 省略 ~

4.2.2 ロジックエラーを見つける

プログラムの構文は正しくても、ロジックにエラーがあった場合、当然プログラムは意図したとおりに動きません。このような時はPUTステートメントを利用して実際の値を確認します。以下のプログラムは、本の貸出日と名前が階層構造になっているデータを読み込みます。貸出日は、その行の先頭が「D」、本の名前は「B」となっています。

サンプルプログラム

```
DATA booksout;
  FORMAT bookname $40.
         outdate yymmdd10.
         duedate yymmdd10.
;
  INPUT rectype $1 @;
  IF rectype='D' THEN INPUT @3 outdate yymmdd10.;
  ELSE IF rectype='B' THEN DO;
    INPUT bookname $ 3-43;
    duedate=outdate+3;
  OUTPUT;
  END;
  DROP rectype;
CARDS;
D 2002/10/08
B SASランゲージ リファレンス
B UNIX版SASシステム使用の手引き
D 2002/10/09
B SAS/GRAPHソフトウェア 初級リファレンス
B SASプロシジャ リファレンス
;
RUN;
```

作成されたデータセットは以下ようになってしまい、意図した結果が得られていません。

OBS	bookname	outdate	duedate
1	SASランゲージリファレンス	.	.
2	UNIX版SASシステム使用の手引き	.	.
3	SAS/GRAPHソフトウェア 初級リファレンス	.	.
4	SAS プロシジャ リファレンス	.	.

このプログラムでは、2つのIFステートメントがあり、それぞれ変数outdate(貸出日)およびduedate(返却期限日)を作成しているので、プログラムの問題点は、これらのIFステートメントを実行すれば確認できます。大量のデータを利用するプログラムを新規作成した場合は、稼働確認にも時間がかかりますが、このサンプルプログラムでは、OBS= INFILEステートメントオプションを指定して、2行目のデータを読み込めば2つのIFステートメントが実行されるようになっています。必要以上にデータを読み込まないので、実行時間が短くなります。また、PUTステートメントを追加すれば各変数にどのような値が格納されているのかがログに出力されるので、ロジックエラーの原因を見つけることができます。

OBS= INFILEステートメントオプションおよびPUTステートメントを追加したサンプルプログラム

```
DATA booksout;
  INFILE CARDS OBS=2;
  FORMAT bookname $40.
         outdate yymmdd10.
         duedate yymmdd10.
;
INPUT rectype $1 @;
IF rectype='D' THEN DO;
  INPUT @3 outdate yymmdd10.;
  PUT 'Record D: ' bookname= outdate= duedate=;
END;
ELSE IF rectype='B' THEN DO;
  INPUT bookname $ 3-43;
  duedate=outdate+3;
  PUT 'Record B: ' bookname= outdate= duedate=;
  OUTPUT;
END;
DROP rectype;

~ 省略 ~
```

実行ログ

```
~ 省略 ~

Record D: bookname= outdate=2002-10-08 duedate=.
Record B: bookname=SASランゲージ リファレンス outdate=. duedate=.
NOTE: 欠損値を含んだ計算により、以下の箇所欠損値が生成されました。
      (回数)(行:カラム)

~ 省略 ~
```

PUTステートメントのログから、行の先頭が「D」のデータを読み込んだ際には貸出日は正しく読み込まれていることがわかります。しかし、行の先頭が「B」のデータを読み込んだ際には、本の名前は正しく読み込まれているものの、貸出日に欠損値が格納されてしまいます。さらに、代入ステートメントで作成される変数duedateにも欠損値が格納されていることがわかります。つまり、このプログラムでは、本の貸出日が保持されていないという問題が発生していることがわかります。変数値を保持するためには、RETAINステートメントを利用します。プログラムを下記のように書き換えることで、本来の目的どおりのプログラムが完成します。

RETAINステートメントを追加したサンプルプログラム

```
DATA booksout;
  FORMAT bookname $40.
         outdate yymmdd10.
         duedate yymmdd10.
;
  RETAIN outdate;
  INPUT rectype $1 @;

~ 省略 ~
```

4.2.3 SASデータセットにダイレクトアクセスしデータをサブセットする
大量のデータを利用するプログラムを作成すると、稼働確認に非常に時間がかかるため、データのサブセット化で時間を節約する方法を4.1.2でご紹介しました。しかしこの方法をソート済みデータセットに対して実行した場合、サブセット化されたサンプルデータに偏りが発生することがあります。このようなときには、SASデータセットの各オブザベーションにダイレクトアクセスし、適当な間隔でデータを抽出してサブセット化します。以下のプログラムは、SASデータセットsashelp.prdsaleから1オブザベーション目から100オブザベーションごとにデータを抽出してサブセット化します。

サンプルプログラム

```
DATA direct1;
  DO pickup=1 to totobs by 100;
    SET sashelp.prdsale POINT=pickup NOBS=totobs;
    OUTPUT;
  END;
  STOP;
RUN;
```

実行ログ

```

1  DATA direct1;
2      DO pickup=1 to totobs by 100;
3          SET sashelp.prdsale POINT=pickup NOBS=totobs;
4          OUTPUT;
5      END;
6      STOP;
7  RUN;

```

NOTE: データセット WORK.DIRECT1 は 15 オブザベーション、10 変数です。

NOTE: DATA ステートメント 処理 :

処理時間	0.08 秒
CPU 時間	0.06 秒

また、サブセットするオブザベーション数をパーセンテージで指定することもできます。以下のサンプルプログラムでは、全データの20%を抽出してサブセット化しています。

サンプルプログラム

```

DATA direct2(drop=samppct);
    samppct=20;
    DO pickup=1 to totobs by (totobs/(totobs*(samppct/100)));
        SET sashelp.prdsale POINT=pickup NOBS=totobs;
        OUTPUT;
    END;
    STOP;
RUN;

```

4.3 関数を利用する

4.3.1 データセットの存在を確認する

関数を利用して、データセットの存在を確認することができます。以下のプログラムは、データセットwork.existの存在を確認し、存在しない場合にはERRORフラグを立てます。

サンプルプログラム

```

DATA exist2;
    SET sashelp.class;
RUN;

DATA _null_;
    dsexist=EXIST('WORK.EXIST');
    IF dsexist=1 THEN
        PUT 'NOTE: データセットWORK.SAMPLEは存在します';
    ELSE DO;
        ERROR 'ERROR: データセットWORK.SAMPLEは存在しません';
    END;
RUN;

```

4.3.2 システムオプションの設定値を取得する

システムオプションの設定値によっては、プログラムが意図した結果にならない場合があります。このような場合、GETOPTION関数を利用して設定値を取得します。以下のプログラムは、乳児のデータをデータセットnewbornとして作成するプログラムです。誕生日のデータは西暦が二桁なので、YEARCUTOFFシステムオプションの設定値が1900の場合、正確な日付をSAS日付に変換することができず、データセットnewbornが作成されません。

サンプルプログラム

```

%let cutoff=%SYSFUNC(GETOPTION(yearcutoff));
OPTIONS yearcutoff=1900;

DATA newborn;
    IF GETOPTION('YEARCUTOFF') >= '1920' THEN DO;
        INPUT name $ sex $ birthday:yymmdd8.;
        FORMAT birthday yymmdd10.;
    END;
    ELSE DO;
        PUT 'NOTE: YEARCUTOFFオプションの設定値を確認してください';
        STOP;
    END;
CARDS;
FUUKA F 02/01/28
TERUAKI M 02/09/03
TAISEI M 03/01/15
;
RUN;
OPTIONS yearcutoff=&cutoff;

```

実行ログ

~ 省略 ~

NOTE: YEARCUTOFFオプションの設定値を確認してください

NOTE: データセット WORK.NEWBORN は 0 オブザベーション、3 変数です。

NOTE: DATA ステートメント 処理 :

処理時間	0.07 秒
CPU 時間	0.07 秒

4.4 自動変数を利用する

4.4.1 何行目のデータを処理しているのか取得する

以下のプログラムは、2月生まれの従業員のデータをデータセット emp_febとして作成するプログラムです。この際、不正なデータがあった場合はERRORステートメントを利用して何行目のデータが不正であるかをログに出力し、そのデータを削除します。

サンプルプログラム

```
DATA emp_feb;
  INPUT empno $ birthday;
  INFORMAT birthday yymmdd10.;
  FORMAT birthday yymmdd10.;
  IF MONTH(birthday)^=2 THEN DO;
    ERROR 'ERROR: ' _n_ '行目のデータは不正です 削除します';
    DELETE;
  END;
CARDS;
00116 1948/02/15
00100 1960/02/29
00116 1961/03/01
00249 1976/02/02
;
RUN;
```

実行ログ

```
~ 省略 ~
ERROR: 3 行目のデータは不正です 削除します
RULE:      ----+-----1----+-----2----+-----3----+-----4----
12         00116 1961/03/01
empno=00116 birthday=1961-03-01 _ERROR_=1 _N_=3
NOTE: データセット WORK.EMP_FEB は 3 オブザベーション、2 変数です。
~ 省略 ~
```

5 おわりに

今回の特集では、SASプログラムのエラーを発見する際に役立つテクニックをご紹介しました。プログラムを意図したとおりに動かすためには、注意深くその動きに注目しなければなりません。「急がば回れ」というように、遠回りでもエラーを発見・確認し、的確に修正することが、結局は早く確実にプログラムを動かすことにつながるのではないのでしょうか。本特集でご紹介した情報が、SASプログラマの方がプログラムの作成・運用を行なう上でのささやかな杖となれば幸いです。



Q&A

マクロ変数の存在チェック

コロンモディファイヤを使用したワードパターンの作成

ユーザIDの取得方法

ODS出力のタイトルに上付き文字を利用したい

UNIX版におけるダブルクリックの速度調整

「XX時XX分」を読み込むフォーマット

Microsoft Excelからのインポートで文字型属性の数値を読み込む方法

カテゴリ変数に対してダミー変数を作成する方法

データ間に対応のある2群の比較



Q

マクロ変数が定義されているかどうかをチェックする方法はありますか？

A

現在定義されているマクロ変数は、SASHELP.VMACROビューを参照して確認できます。チェックするマクロ変数名と変数NAMEの値が同じオブザベーションがあれば、マクロ変数が定義されています。

例1

```
%macro check(mvar);
  %* mvar: マクロ変数名;
  %local i tmp;
  %let dsid=%sysfunc(open(sashelp.vmacro));
  %let num=%sysfunc(varnum(&dsid,name));
  %do %until(&ob = -1);
    %let i=%eval(&i+1);
    %let ob=%sysfunc(fetchobs(&dsid,&i));
    %let val=%sysfunc(getvarc(&dsid,&num));
    %if &val = %upcase(&mvar) %then %do;
      %let ob = -1;
      %let tmp=yes;
    %end;
    %else %do;
      %let tmp=no;
    %end;
    %if &ob=-1 %then %do;
      &tmp
    %end;
  %end;
  %let rc=%sysfunc(close(&dsid));
%mend check;

%put Does the macro variable exist: %check(abc);
```

例2

```
%macro check2(mvar);
  %* mvar: マクロ変数名;
  %global _exist;
  %let _exist=;
  proc sql noprint;
    select count(*) into :_chk from sashelp.vmacro
      where name = "%upcase(&mvar)";
  quit;
  %if &_chk = 0 %then %let _exist=no;
  %else %let _exist=yes;
%mend check2;

%check2(abc);
%put &_exist;
```

Q

条件文を使用して、文字列中の1文字目に特定の値が入っていた場合にのみ、そのオブザベーションを抽出するという処理を考えています。WHEREステートメントでLIKE演算子を使用すれば可能ですが、入力データがSASデータセットである必要があります。読み込むファイルがブランク区切りやカンマ区切り(CSV)ファイルの場合、どのようにプログラムを記述すれば良いのでしょうか？

A

入力データセットがSASデータセットであれば、WHEREステートメントとLIKE演算子を併用することで、文字列を抽出するためのワードパターンを作成することが可能です。下記の例は、変数stringに含まれる文字列の値が、「B」で始まるオブザベーションのみ抽出されます。

```
WHERE string LIKE "B%";
```

しかし、入力データがブランク区切りやカンマ区切りなどである場合、WHEREステートメントは使用できません。このようなときは、サブセット化IFステートメントでSUBSTR関数、またはKSUBSTR関数を使用する方法が考えられますが、「コロンモディファイヤ」を使用する方がより簡便です。なお、コロンモディファイヤは前方一致のみに対応し、中間一致や後方一致には対応していません。次ページに、変数hanteiの値が「不」から始まるオブザベーションのみを抽出するサンプルプログラムを示します。

```

data test;
  input kamoku $ hantei $;
  *if KSUBSTR(hantei,1,1)="不"; /* KSUBSTR関数を使用する場合の記述*/
  if hantei = ":";"不"; /* コロンモディファイヤを使用する場合の記述 */
cards;
英語 合格
国語 合格
数学 合格
理科 不合格
社会 合格
;
run;

proc print data=test noobs;
run;

```

上記サンプルプログラムの出力結果は、下記ようになります。

```

kamoku hantei

理科 不合格

```

Q SASを実行しているOSのユーザIDをプログラム中で取得したいのですが、SYSGET関数を使用する場合、WindowsとUNIXでは環境変数の名前が異なるのでそれぞれのOS用にプログラムを書かなければなりません。プラットフォームに依存しない、ユーザIDを取得する関数のようなものはありますか。

A SAS System 8で追加された自動マクロ変数SYSUSERIDを使用することで、プラットフォームに依存せずにユーザIDを取得できます。なお、SAS System 8でも、これまで通りSYSGET関数を使用してユーザIDを取得できます。

SYSGET関数 Windowsの場合

SYSGET関数	Windowsの場合 (デフォルトの環境変数)	%put %SYSGET(USERNAME);
	UNIXの場合 (デフォルトの環境変数)	%put %SYSGET(LOGNAME);
SYSUSERID自動マクロ変数	OSに依存しない	%put &SYSUSERID;

Q カイ2乗検定の結果をHTMLファイルに出力しています。このとき、HTMLファイルのタイトルに、「2乗」を上付き文字として出力したいのですが、可能でしょうか。

A SAS 8.2でODS機能に追加されたESCAPECHAR=ステートメントを指定することで、TITLEステートメントに特殊文字を使用できるようになりました。この機能を利用して、特殊文字に「^」を割り当て、HTMLタグの「²」を記述すれば、タイトルに上付き文字を出力できます。

```

data test; /* サンプルデータ作成 */
  input x y freq;
cards;
1 1 12
1 2 21
2 1 19
2 2 31
;
run;

ODS ESCAPECHAR='^'; /* 特殊文字を出力 */
ODS HTML FILE='c:\test.html';

TITLE '_^{super 2}検定'; /* カイ2乗検定 */
proc freq data=test;
  tables x*y/chisq;
  weight freq;
run;

ODS HTML CLOSE;

```

なお、下付文字を出力する場合は、同じ特殊文字を用いて「^{sub 2}」のように記述します。

Q UNIX版SASをDMSで利用しています。マウスのダブルクリックの速度を調整することは可能でしょうか？

A Xリソースを利用して調整できます。

```
SAS.*MultiClickTime: 数値(ミリ秒単位)
```

具体的には、ご利用のXリソースファイルに次のように追記するか、コマンドラインなどのオプションとして個別に指定することも可能です。

Xリソースファイルへの追記例

```
SAS.*MultiClickTime:400
```

コマンドラインからの指定例

```
sas_ja -xrm "SAS.*MultiClickTime:400"
```

なお、ダブルクリック値のデフォルト値は、200(ミリ秒)です。数値を大きくすることで、ダブルクリックの速度が遅くても反応するように調整できます。

Q

「XX時XX分」を読み込むフォーマットはありますか？

A

残念ですがそのようなフォーマットは用意されていません。しかし、いったん文字変数として読み込み、KTRANSLATE関数を使用すれば、SAS時間値として取り込むことが可能です。

```
/* "時"を":"(半角)に変換し、"分"を削除して時間値として読み込む */
data _null_;
  A="11時30分";
  B=input(KTRANSLATE(A,":","時","","分"),TIME5.);
  put B= TIME5.;
run;
```

Q

Microsoft Excelからのデータをインポートすると、本来存在するはずの数値データのうち、ところどころ欠損値となる部分があります。正しく読み込むためにはどうしたらよいでしょうか。

A

ご質問の状況は、Microsoft Excel上で数値型であるべきデータが文字型として入力されているために、インポート時に欠損値となっていることが考えられます。このような場合は、SASへデータをインポートする前に、Microsoft Excel上で文字型として入力されている各フィールドを、本来の数値型に修正してからインポートするか、またはいったんCSVなどの形式に出力後に再度インポートすることで、回避可能です。

Q

たとえば比例ハザードモデルを扱っている場合、PHREGプロシジャではCLASSステートメントがないため、カテゴリ変数を何らかの方法でダミー変数に展開する必要があります。現在はDATAステップで上記の処理をしていますが、何か他に良い方法がありますか？

A

カテゴリ変数が少なく、また水準数も少ない場合には、DATAステップを利用するのが一番簡単ですが、モデルが複雑なケースでは煩雑になってしまいます。そのようなときには、下記のSAS/STATのプロシジャを用いると、比較的容易にダミー変数を作成することが可能です。

GLMMODプロシジャ

GLMプロシジャと同じ展開方法でダミー変数を作成します。展開されたカテゴリ変数をデータセットへ出力するためには、OUTDESIGN=オプションを指定する、またはODS OUTPUTステートメントを使用します。なお、MODELステートメントに存在するもののCLASSステートメントでは指定されていない変数は、そのままデータセットへ出力されます。

```
/*OUTDESIGN=オプションを用いて、データセットout1に出力*/
proc glmmod data=a outdesign=out1 noprint;
  class drug disease;
  model y=drug disease drug*disease;
run;

proc print data=out1;
run;

ods listing close;
/*ODS OUTPUTステートメントを用いて、データセットout2へ出力*/
ods output DesignPoints=out2;
proc glmmod data=a;
  class drug disease;
  model y=drug disease drug*disease;
run;
ods listing;

proc print data=out2;
run;
```

TRANSREGプロシジャ

変数に対してさまざまな変換を施してから回帰を行なうTRANSREGプロシジャでも、DESIGNオプションを指定してダミー変数を作成し、またOUTPUTステートメントを併せて用いてデータセットへ出力することができます。GLMMODプロシジャと比べると、やや難解なプログラムとなりますが、従属変数を必ずしも指定しなくても良いこと、また複数の展開方法がサポートされているなどのメリットもあります。

```
proc transreg data=a design;
  model class(drug|disease);
  output out=out3;
run;

proc print data=out3;
run;

proc transreg data=a design;
  model class(drug|disease / effect);
  /*EFFECTコーディング*/
  output out=out4;
run;

proc print data=out4;
run;
```

GLMMODプロシジャとTRANSREGプロシジャの比較

	TRANSREGプロシジャ	GLMMODプロシジャ
変数数	変数と値の組合せ	OUTDESIGN=を使用するとCOL1,COL2,...,COLn
ダミー変数への展開方法	オプションの指定によってカスタマイズが可能	GLMプロシジャと同じ
従属変数	不要	従属変数の指定が必要
欠損値	欠損値として出力される	欠損地を含むオブザベーションごと除外される
処理速度	GLMMODプロシジャの方が速い傾向にある (TRANSREGプロシジャは、入力データの値から変数を作成する)	
その他の機能	マクロ変数に出力変数名が格納される	OUTPARM= オプションで変数名の情報を出力できる

なお、SAS/IMLにもダミー変数に展開する関数が用意されています。また、次期バージョン SAS System 9 以降では、LOGISTICプロシジャなどにも同様の機能が加わる予定です。

Q

データ間に対応がある場合、2群の比較を行なうにはどのようにすればいいですか。

A

2群の比較を行なうには、通常t検定が用いられますが、薬剤投与前後の観測値のように、データ間に対応がある場合には適していません。このような場合、対応を考慮したt検定を行なう必要があります。SAS System 8以降では、TTESTプロシジャにPAIREDステートメントが追加されたので、以下のサンプルプログラムのように指定することで、対応のあるt検定を行なうことができます。

```
data test;                /* サンプルデータの作成 */
  input before after@@;
cards;
10 13  14 15  13 12  15 18  21 17  15 19
;
run;

proc ttest data=test;
  paired after*before;
run;
```

また、同様の検定は、DATAステップでオブザベーションごとに変数間の差を求め、その差に対してUNIVARIATEプロシジャを実行することによって求めることができます。この方法では、ノンパラメトリック手法であるWilcoxonの符号付き順位と検定の結果も併せて得ることができます。具体的な方法に関しては右期のサンプルプログラムを参考にしてください。

```
data test2;              /* オブザベーション間の差を算出 */
  set test;
  diff=after-before;
run;

proc univariate data=test2;
  /* 変数DIFFに対する検定統計量を算出 */
  var diff;
run;
```

検定の結果は、「位置の検定: $\mu_0=0$ 」の表に表示されます。この表では1行目には対応のあるt検定、3行目にはWilcoxonの符号付き順位と検定に対応する検定統計量とp値が表示されます。



SAS Training

SASトレーニングのお知らせ

SASトレーニングのご案内

2003年10月より、すべてのトレーニングコースについて、料金の改定を行います。また、これに併せてチケット捺印数についても改定を行います。詳細は、以下のURLをご参照ください。

<http://www.sas.com/japan/training/>

コース名	期間 / 価格(サービスチケット捺印数)	会場	開催日程
「SASプログラミング:Basic I」コース	2日間 90,000円()	東京	10/6・7、10/23・24、11/4・5、11/17・18、12/1・2、12/11・12
		大阪	10/16・17
「SASプログラミング:Basic II」コース	2日間 90,000円()	東京	10/14・15、10/28・29、11/10・11、11/27・28、12/8・9、12/18・19
		大阪	10/27・28
「実践データハンドリング」コース	1日間 45,000円()	東京	12/5
		大阪	11/21
「マクロ言語入門」コース	1日間 45,000円()	東京	10/31、11/25、12/15
		大阪	11/10
「マクロ言語上級」コース	1日間 45,000円()	東京	11/14
		大阪	12/5
「スタッフ向けデータ活用」コース	1日間 45,000円()	東京	11/19
「中間管理職向けデータ活用」コース	半日間 30,000円()	東京	11/12(13:30 ~ 16:30)
「統計概論」コース	1日間 45,000円()	東京	10/3、11/7
「SASによるデータ解析入門」コース	2日間 90,000円()	東京	10/9・10、11/13・14
		大阪	個別トレーニングとして承ります
「医薬向けSASによるデータ解析入門」コース	2日間 90,000円()	東京	10/27・28、12/8・9
		大阪	個別トレーニングとして承ります
「医薬向け分散分析」コース	1日間 45,000円()	東京	12/2
		大阪	個別トレーニングとして承ります
「医薬向けカテゴリカルデータ解析1」コース	2日間 90,000円()	東京	10/30・31
「医薬向けカテゴリカルデータ解析2」コース	2日間 90,000円()	東京	11/17・18
「SASによる回帰分析」コース	2日間 90,000円()	東京	11/20・21
「多変量解析」コース	2日間 90,000円()	東京	12/4・5
「ロジスティック回帰分析」コース	2日間 90,000円()	東京	11/27・28
「SAS/IML入門」コース	1日間 45,000円()	東京	12/17
「臨床データマネジメント」コース	2日間 90,000円()	東京	11/10・11
		大阪	個別トレーニングとして承ります
「集計解析表作成」コース	2日間 90,000円()	東京	11/25・26
		大阪	個別トレーニングとして承ります

New Publication

新刊マニュアルのご紹介

「第22回 日本SASユーザー会総会および研究発表会論文集」

注文番号:12003

価格:4,000円(税抜)

本書は、今年開催された「第22回 日本SASユーザー会総会および研究発表会論文集(SUGI-J2003)」で発表された論文をまとめたものです。本書に掲載されている論文一覧は、下記のとおりです。

口頭発表論文

医薬品開発

「臨床試験の早期中止の検討におけるベイズ流予測確率と条件付き検出力の利用について」

堺 伸也(イービーエス株式会社/東京理科大学)

菅波 秀規(興和株式会社/東京理科大学)

「Group Sequential計画のためのパワーシミュレータの開発」

本田 圭一、田崎 武信(塩野義製薬株式会社)

太田 裕二、佐賀野 修一(住商情報システム株式会社)

「投与期間別の副作用発現率を解釈するために」

古川 雅史、片山 和夫、田崎 武信(塩野義製薬株式会社)

「SAS Integration Technologies +

ASPによる解析帳票作成Webシステム構築の試み」

岩本 光司(武田薬品工業株式会社)

矢野 尚(株式会社富士通ビー・エス・シー)

「2値および計量値のシグモイド曲線

- 曲線の推定および逆推定と95%信頼区間 - 」

杉山 公仁(昭和薬品化工株式会社)

馬場 淳(明治製菓株式会社)

天竺桂 裕一朗(興和株式会社)

高橋 行雄(中外製薬株式会社)

「陰性および陽性対照があるシグモイド曲線

- ダミー変数を持つ非線型回帰モデルの応用 - 」

山田 雅之(キッセイ薬品工業株式会社)

吉田 光宏(グラクソ・スミスクライン株式会社)

高橋 行雄(中外製薬株式会社)

「計量値のシグモイド用量反応曲線の同時推定

- 効力比とその95%信頼区間 - 」

高橋 行雄(中外製薬株式会社)

チュートリアル

「生存時間解析における症例数設計」

浜田 知久馬、藤井 陽介(東京理科大学)

統計解析

「区間打ち切り生存時間データのセミパラメトリックな解析法のSASプログラムの紹介 - ギブス・サンプラーを利用した周辺尤度アプローチ - 」

西山 智(アベンティス ファーマ株式会社/東京理科大学)

吉村 健一(東京大学)

「イベント発生確率推定時における連続変数のカテゴリー化、およびカテゴリ変数の実数化」

上條 史夫、川崎 章弘(株式会社数理技研)

「SASによる生存時間の多重イベントの解析 - 糖尿病合併症を例に - 」

広本 篤、金子 徹治、大橋 靖雄(東京大学)

「再発事象に対するモデルを用いた解析方法の検討」

中 牧子、大橋 靖雄(東京大学)

「MIXEDプロシジャを用いた線形混合効果モデルの交互作用の指定方法」

寒水 孝司(東京理科大学)

菅波 秀規(興和株式会社/東京理科大学)

「要因配置実験の効果成分の表示から生じる不定性」

柴山 忠雄(前:名古屋市工業研究所)

統計教育

「CROにおけるSASプログラムの育成教育」

竹田 眞、佐藤 智美(株式会社CRCソリューションズ)

現所属:株式会社ACRONET

システム

「CALL EXECUTEを用いたマクロの再帰呼び出しと統計計算への応用」

伊藤 要二(アストラゼネカ株式会社)

「SAS未経験者をSAS内部構造を理解したDATAステップ

SASプログラマに短期間で育成するカリキュラムの紹介」

山田 大志、小澤 康彦、宮浦 千香子(アストラゼネカ株式会社)

「Microsoft AccessとSASによるデータマネジメントシステム」

中村 竜児、松沢 享(メディカル統計株式会社)

「SAS® Metadata, Authorization and Management Services

- Working Together for You SASによるメタデータマネジメント」

Michelle Ryals(SAS Institute Inc.)

翻訳:鹿渡 圭二郎、李 錦実、江口 英男(SAS Institute Japan株式会社)

「Enterprise Guide 2.0によるadd-in機能について」

木下 貴文(SAS Institute Japan株式会社)

「SAS/SHAREサーバーアクセスログの分析」

中村 崇文(SAS Institute Japan株式会社)

「簡易 運用入門」

弘田 貴(SAS Institute Japan株式会社)

「MEANS、TABULATE、DATASETSプロシジャの機能紹介」

檜皮 孝史、渋谷 佳枝、迫田 奈緒子(SAS Institute Japan株式会社)

経営・経済

「SASソフトウェアを利用したCIR++モデルのパラメータ推定と金利パス生成」
岸田 則生(株式会社CRCソリューションズ)

「コンシューマ・クレジット業の利益指向の新与信モデル」
小野 潔(株式会社UFJ銀行)

「非補償型ロジットモデルを用いた企業倒産確率の予測モデル
- NLP Procedureによる非補償型ロジットモデルに対するパラメータ推定 -」
坂巻 英一(株式会社金融工学研究所 / 東京工業大学)

「SAS Risk Dimensionsによる統合リスク分析のご紹介」
嘉陽 亜希子、鬼頭 拓郎、尾高 雅代、田中 愛(SAS Institute Japan株式会社)

調査・マーケティング

「建築生産における建築物の耐久性確保に関する実務者の意識と実態」
小島 隆矢(独立行政法人建築研究所)
小野 久美子、植木 暁司(国土交通省国土技術政策総合研究所)

「看護師のセクシャルハラスメントに対する意識について」
田久 浩志(中部学院大学)
岩本 晋(NPO福祉法人 OIEMASE)

「Life Time Valueを基準とした施策の最適化方法
- 遺伝的アルゴリズムによる解析事例 -」
小谷田 知行、堀 彰男(株式会社浜銀総合研究所)

「Bioinformaticsの手法を活用した
クレジットカード取引履歴データの途上審査モデルへの適用事例」
堀 彰男、小谷田 知行(株式会社浜銀総合研究所)

SASソリューション

「ゲノム創薬向け統合ソリューション
SAS Scientific Discovery Solutionsの紹介」
段谷 高章(SAS Institute Japan株式会社)

ポスターセッション論文**統計解析**

「一般化推定方程式およびSASの解析ツール」
王 露萍、野口 知雄(アベンティス ファーマ株式会社)
高田 康行(持田製薬株式会社)

「NLMIXEDプロシジャーを用いた
ItemResponseModelのシミュレーション」
板東 説也(有限会社電助システムズ)
宮岡 悦良、緑川 修一、高原 佳奈(東京理科大学)

「変量効果モデルによるメタ・アナリシス
DerSimonian-Laird法のSASマクロの作成」
中西 豊支、浜田 知久馬(東京理科大学)

「メタ・アナリシスにおける公表バイアスの評価
trim-and-fill法のSASマクロの作成」
松岡 伸篤、浜田 知久馬(東京理科大学)

統計教育

「看護系大学における疫学・生物統計学教育の実態調査」
田中 司朗(東京大学)

システム

「SASを用いたXMLデータの作成 - ODM ver. 1.1対応 -」
岡下 邦博、進藤 三富子(株式会社日本アルトマーク)

「SASデータセットのエクスポート」
羽田野 実(SAS Institute Japan株式会社)

経営・経済

「労働市場の時系列分析 - JMPを利用して -」
浦澤 浩一(株式会社八千代銀行 / 青山学院大学)

「アジルなSupply Chainを実現する予測プロセスの自動化
- SAS® High-Performance Forecastingのご紹介 -」
松舘 学(SAS Institute Japan株式会社)

調査・マーケティング

「地方における実演芸術鑑賞の実態
- 県民芸術劇場(兵庫県)の来場者調査より -」
有馬 昌宏(神戸商科大学)

「青年期女性の自意識と完全主義傾向の関連」
中村 晃士、牛島 定信、縣 俊彦、清水 英佑(東京慈恵会医科大学)

「個人レベルの嗜好を基にしたクラスタリング」
河崎 一益(株式会社日本アルトマーク)
松沢 利繁(株式会社インターナショナル・クリエイティブ・マーケティング)

「患者参加型医療情報交換システムのニーズ調査」
義澤 宣明、船曳 淳(株式会社三菱総合研究所)
小山 博史(東京大学)

グラフィック・統計教育

「SAS/GRAPH入門 - 社内における教育研修事例 -」
林 行和、畑中 雄介、小出 起美雅、山口 孝一(株式会社CRCソリューションズ)
現所属:株式会社ACRONET

グラフィック・レポート

「SASグラフによる動く万華鏡の作成」
岸本 容司(神戸商科大学)

上記のほか、過去の論文集(1993~1995年、1997年~2002年度版)についても随時販売していますので、ご利用ください。ただし在庫がなくなり次第、販売終了とさせていただきますのでご了承ください(1996年度版が終了しています)。

マニュアル販売係

T E L 03-3533-3835
F A X 03-3533-3781
E-mail JPNBooksale@sas.com

SASマニュアル注文用紙、および最新のPublication Catalog(マニュアル案内パンフレット)は弊社ホームページ(<http://www.sas.com/japan/manual/>)にて公開しておりますので、併せてご利用ください。

Latest Releases

最新リリース情報

PCプラットフォーム

Windows版	SAS 8.2	TS2M0
----------	---------	-------

ミニコンピュータプラットフォーム

OpenVMS AXP版	SAS 6.12	TS020
OpenVMS VAX版	SAS 6.08	TS407

UNIXプラットフォーム

Tru64版	SAS 8.2	TS2M0
ABI+版	SAS 6.11	TS040
SunOS/Solaris版	SAS 8.2	TS2M0
HP-UX版	SAS 8.2	TS2M0
AIX版	SAS 8.2	TS2M0
Linux版	SAS 8.2	TS2M0

メインフレームプラットフォーム

IBM版(OS/390, z/OS)	SAS 8.2	TS2M0
富士通版(F4, MSP)	SAS 6.09E	TS470
日立版(VOS3)	SAS 6.09E	TS470
CMS版	SAS 6.08	TS410

SAS Technical News Autumn 2003

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: 03-3533-3877

FAX: 03-3533-3781

E-mail: technews@jpn.sas.com



SAS Institute Japan株式会社

www.sas.com/japan/

東京本社
〒104-0054
東京都中央区勝どき1-13-1
イヌイビル・カチドキ 8F
Tel 03 (3533) 6921
Fax 03 (3533) 6927

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