初/中級者が陥りやすいプログラミングエラー

井上　貴博
（ノバルティス　ファーマ株式会社）

Programming errors that easily occur on beginner/intermediate SAS programmers

Takahiro Inoue
Data Sciences & Scientific Operations Department,
Novartis Pharma K.K.

要旨:

初/中級者には理解し難く陥りやすいプログラミングエラーは多数ある。エラー発見・解消テクニックを紹介し、浮動小数点の丸め誤差、PDVなど、理解していないと意図した結果が得られない事例を紹介する。

キーワード: プログラミング、テクニック、エラー、浮動小数点、丸め誤差、PDV 、Program Data Vector

Disclaimer

The opinions expressed in this presentation and on the following slides are solely those of the presenter and not necessarily those of Novartis. Novartis does not guarantee the accuracy or reliability of the information provided herein.

論文の構成

1. はじめに
2. エラー発見・解消テクニック
3. 浮動小数点の丸め誤差
4. Program Data Vector
5. おわりに

はじめに

- SAS OnDemand for Academicsを使用し作成しています。処理内容が再現ができるように論文にプログラムも記載していますが、他の動作環境を使用する場合、結果が異なる場合があります。
- 発表時間の関係上、割愛・省略している部分が多くあり、詳細は論文集をご確認ください。

エラー発見・解消テクニック

1. はじめに
2. エラー発見・解消テクニック
3. 浮動小数点の丸め誤差
4. Program Data Vector
5. おわりに

詳細は論文をご確認ください。

ログについて

- プログラムを実行した際、生成されるログには、以下の情報が含まれている。
  - 実行されたプログラム
  - 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)
  - プログラム実行中の発生した警告・エラーメッセージなど
  - プログラム実行に要した時間
- エラーの原因やプログラム修正のためのヒントは、ログから見つけることができ、ログを確認することが重要である。

詳細は論文をご確認ください。

ログの確認

- ERROR, WARNINGではないNOTEでも注意すべきメッセージが多数ある。
- ログに確認する際のキーワードの一例を以下に示す。

ERROR	WARNING	BY 値を繰り返すデータセット （バージョンにより「BY値」とスペースがないものもある）		
欠損値を含んだ	初期化され	処理を中止	範囲外	による割り算がありました
ゼロ割り	無効な	LOST CARD	変換	文字を超えています
実行を中止	切り捨て	表示が制限	ステートメントの余分な情報を無視します	
既にライブラリ	出力形式によって小数点がシフト			変数がありません
存在しません	演算式を計算できなかった			

詳細は論文をご確認ください。

エラーのタイプ

- 発生したエラーのタイプを知ることで、そのエラーのタイプに合わせたプログラム修正ができる。
  - 構文エラー(Syntax errors)
  - セマンティックエラー(Semantic errors)
  - 実行時エラー(Execution-time errors)
  - データエラー(Data errors)
  - マクロ関連エラー(Macro-related Errors)※
  - 論理エラー (Logic errors)

※本稿では紹介していない

詳細は論文をご確認ください。

エラー発見・解消テクニック

- これを組み合わせることにより、検証時のデータ損失の予防や、エラー発生箇所の特定を容易にすることができる。
  - オプション(システム/データセット)
  - ステートメント
  - 関数およびCALLルーチン
  - 自動変数

浮動小数点の丸め誤差

1. はじめに
2. エラー発見・解消テクニック
3. 浮動小数点の丸め誤差
4. Program Data Vector
5. おわりに

事例

変数fr1とfr2の値は？

```
data temp ;
    a = 3 * 0.1 ;
    if 0.3 < a  then fr1 = 1 ;
    if a <= 0.3 then fr2 = 1 ;
    put a= fr1= fr2= ;
run ;
```

0.3

FALSE

fr1=.

TRUE

fr2=1

実際の結果


```
61 data temp ;
62 a = 3 * 0.1 ;
63 if 0.3 < a then fr1 = 1 ;
64 if a <= 0.3 then fr2 = 1 ;
65 put a= fr1= fr2= ;
66 run ;
a=0.3 fr1=1 fr2=.
NOTE: データセットWORK
NOTE: DATAステートメント
```

a=0.3 fr1=1 fr2=.

予想と逆の結果に

```
data temp ;
    a = 3 * 0.1 ;
    if 0.3 < a then fr1 = 1 ;
    if a <= 0.3 then fr2 = 1 ;
    put a= fr1= fr2= ;
run ;
```

0.3

FALSE

fr1=.

TRUE

fr2=1

2進数から10進数

1 1 0 0 1 (2進数)

2^4 2^3 2^2 2^1 2^0

$1 \times 16 + 1 \times 8 + 0 \times 4 + 0 \times 2 + 1 \times 1$

= 25 (10進数)

2進数から10進数

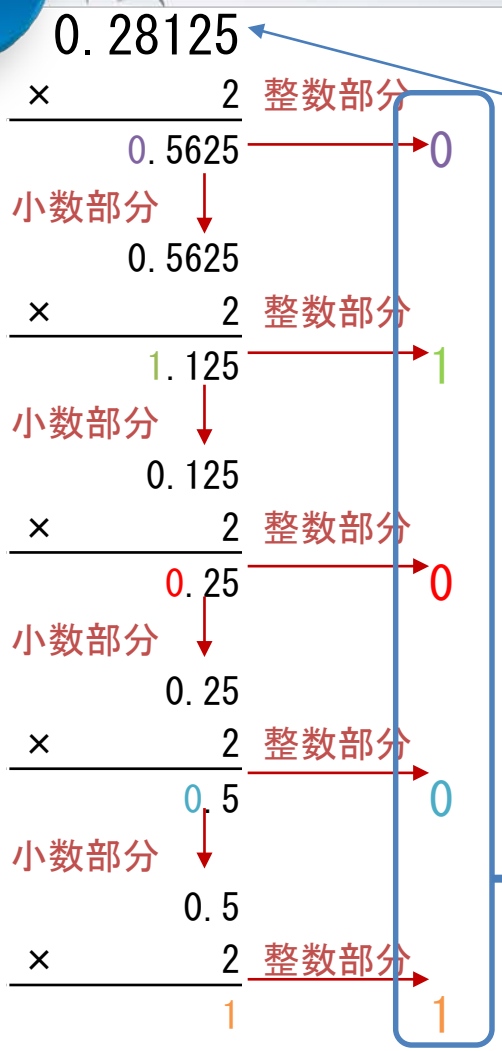
$0.0\,1\,0\,0\,1$ (2進数)

$2^{-1}\ 2^{-2}\ 2^{-3}\ 2^{-4}\ 2^{-5}$

$0 \times 0.5 + 1 \times 0.25 + 0 \times 0.125 + 0 \times 0.0625 + 1 \times 0.03125$

$= 0.28125$ (10進数)

10進数から2進数



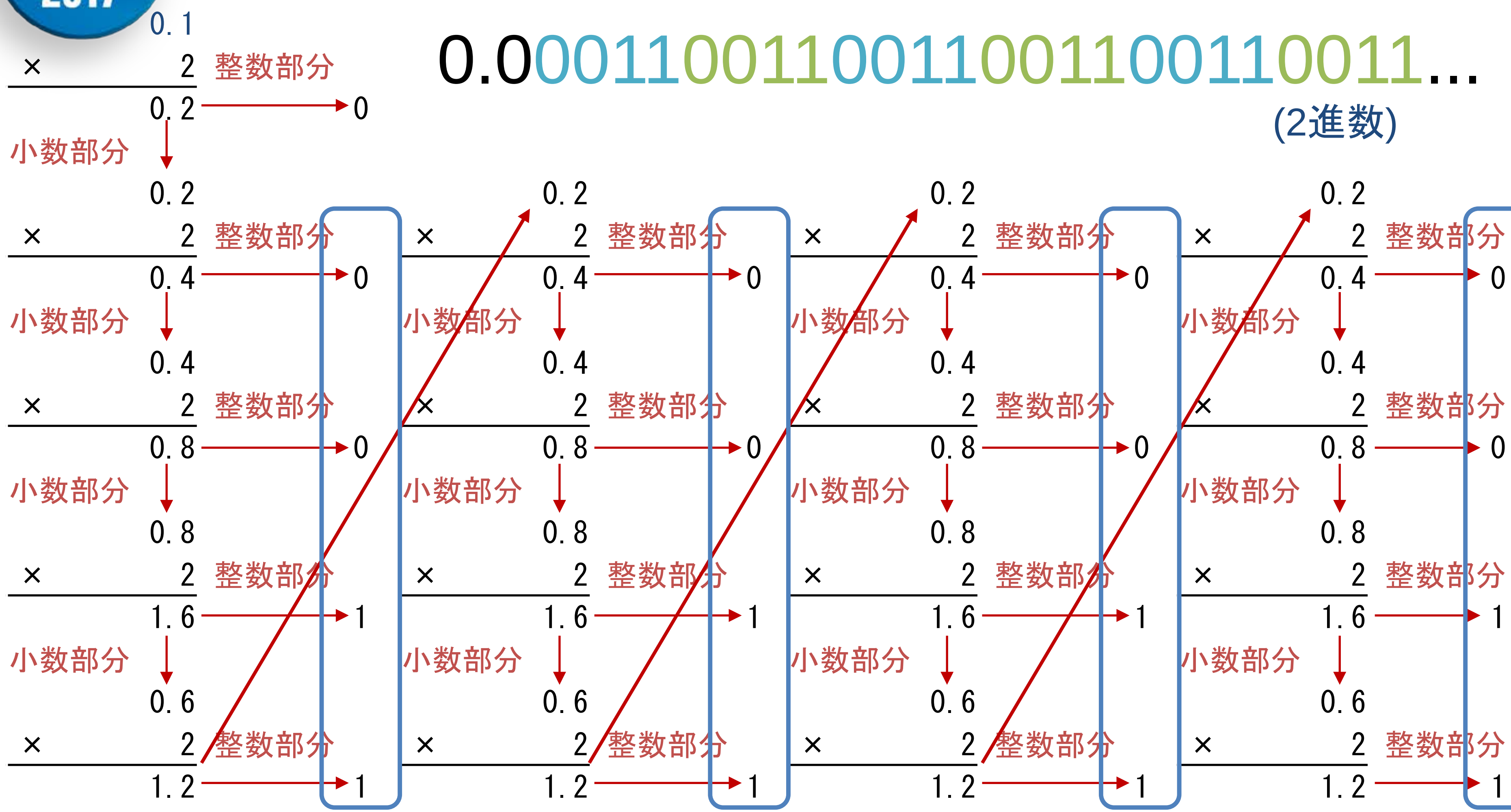
0.28125 (10進数)

0.01001 (2進数)

10進数「0.1」の2進数

0.00011001100110011001100110011…

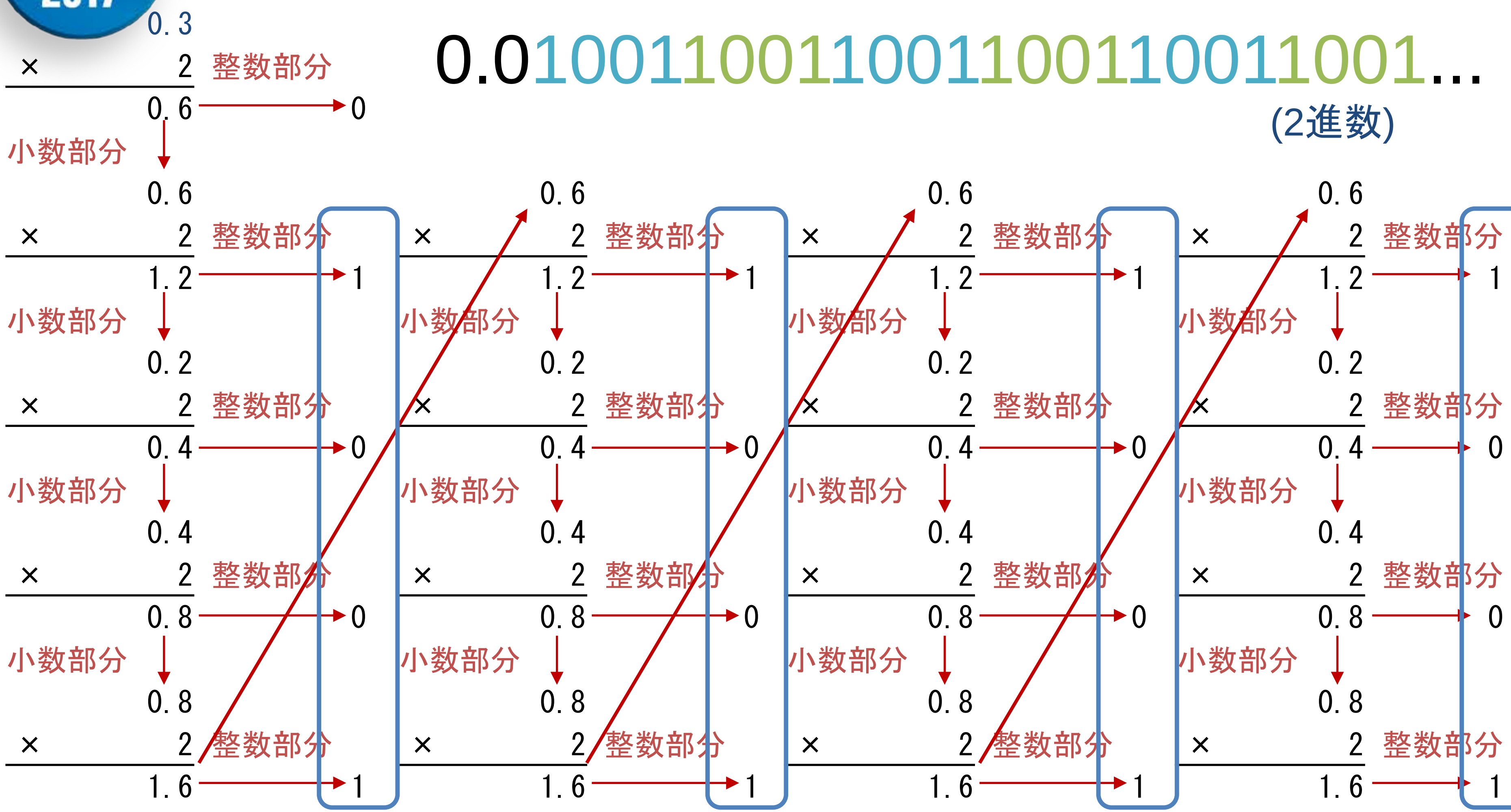
(2進数)



10進数「0.3」の2進数

0.0100110011001100110011001...

(2進数)



2進数での無限小数点(循環小数)

- 10進数では有限小数点であっても、
  2進数では無限小数点(循環小数)となる場合がある
- コンピュータ内部では、物理的な格納となり、
  つまり有限桁に変換され、精度が失われる
- SASはIEEE 準拠の浮動小数点表現 (倍精度浮動小数点表現)を使用している。

原因


```
60 data temp ;
61 a = 3 * 0.1 ;
62 b = 0.3 ;
63 c = a - b ;
64 put a= b= c= ;
65 run ;
a=0.3 b=0.3 c=5.551115E-17
NOTE: データセットWORK.TEM
NOTE: DATAステートメント処理
```

a=0.3 b=0.3 c=5.551115E-17
同じ値に見えるが、差異がある

```
data temp ;
  a = 3 * 0.1 ;
  if 0.3 <  a then fr1 = 1 ;
  if a <= 0.3 then fr2 = 1 ;
  put a= fr1= fr2= ;
run ;
```

0.3

FALSE fr1=.

TRUE fr2=1

コンピュータは、有限精度の2進算術演算を使用
厳密な2進表現がない数値を扱う場合、コンピュータによって生成される
結果は、10進算術演算で生成される結果とわずかに異なる

回避策

ROUND関数を使用して丸め処理を行う

```
data temp ;
     a = 3 * 0.1 ;
     if 0.3 < round(a, 1e-8) then fr1 = 1 ;
     if round(a, 1e-8) <= 0.3 then fr2 = 1 ;
     put a= fr1= fr2= ;
run ;
```

ROUND関数の注意点

通常、ROUND関数は、結果の有効桁数が9以下で、次のどの条件にも該当しない場合、10進算術演算で期待される結果を生成する。

- 丸めの単位が整数である。
- 丸め単位が1e-15以上の10のべき乗である。
  (丸め単位が1よりも小さい場合、ROUNDは、丸め単位の逆数が10のべき乗との違いが最下位から第3または第4ビッドまででであれば、丸め単位を10のべき乗として扱う)
- 10進算術演算で期待される結果が小数第4位以下である。

Program Data Vector

1. はじめに
2. エラー発見・解消テクニック
3. 浮動小数点の丸め誤差
4. Program Data Vector
5. おわりに

事例①

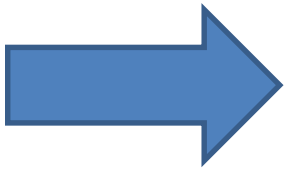
このプログラムを実行した際の結果は？

[Dataset: A]

id	var1
a1	1
a2	2

[Dataset: B]

id	var1	var2
b1	1	101
b2	2	202



[Dataset: C]

id	var1	var2
a1	1	.
a2	2	.
b1	1	101
b2	2	202

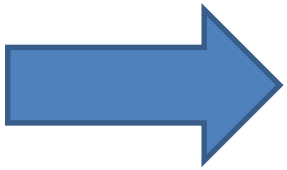
```
/* プログラム */
data C ;
    set A B ;
run ;
```

事例②

このプログラムを実行した際の結果は？

[Dataset: C]

id	var1	var2
a1	1	.
a2	2	.
b1	1	101
b2	2	202



[Dataset: D]

id	var1	var2	var3
a1	1	0	0
a2	2	.	.
b1	1	0	0
b2	2	202	.

```
/* プログラム */
data D ;
    set C ;
    if var1 = 1 then var2 = 0 ;
    if var1 = 1 then var3 = 0 ;
run ;
```

事例③

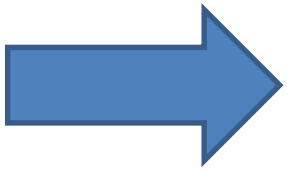
このプログラムを実行した際の結果は？

[Dataset: A]

id	var1
a1	1
a2	2

[Dataset: B]

id	var1	var2
b1	1	101
b2	2	202



[Dataset: E]

id	var1	var2	var3
a1	1	0	0
a2	2	0	.
b1	1	0	0
b2	2	202	.

```
/* プログラム */
data E ;
    set A B ;
    if var1 = 1 then var2 = 0 ;
    if var1 = 1 then var3 = 0 ;
run ;
```

二つのプログラムの違い

[Dataset: C]

id	var1	var2
a1	1	.
a2	2	.
b1	1	101
b2	2	202

```
data D ;
    set C ;
    if var1 = 1 then var2 = 0 ;
    if var1 = 1 then var3 = 0 ;
run ;
```

[Dataset: D]

id	var1	var2	var3
a1	1	0	0
a2	2	.	.
b1	1	0	0
b2	2	202	.

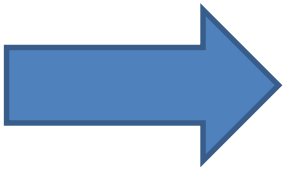
```
data E ;
    set A B ;
    if var1 = 1 then var2 = 0 ;
    if var1 = 1 then var3 = 0 ;
run ;
```

[Dataset: A]

id	var1
a1	1
a2	2

[Dataset: B]

id	var1	var2
b1	1	101
b2	2	202



[Dataset: E]

id	var1	var2	var3
a1	1	0	0
a2	2	0	.
b1	1	0	0
b2	2	202	.

Program Data Vector (PDV)

SASは、PDVに一度新しいデータセットを作成する。

[Dataset: C]

id	var1	var2
a1	1	.
a2	2	.
b1	1	101
b2	2	202

[Dataset: D]

id	var1	var2	var3
a1	1	0	0
a2	2	.	.
b1	1	0	0
b2	2	202	.

PDV内の処理

Program Data Vector

[Dataset: A]

id	var1
a1	1
a2	2

[Dataset: B]

id	var1	var2
b1	1	101
b2	2	202

```
data E ;
    set A B ;
    if var1 = 1 then var2 = 0 ;
    if var1 = 1 then var3 = 0 ;
run ;
```

[PDV]

id	var1	var2	var3
a2	2	0	.
既存			新規

[Dataset: E]

id	var1	var2	var3
a1	1	0	0
a2	2	0	.
b1	1	0	0
b2	2	202	.

前オブザベーションのデータが残った状態

Mergeの事例

[Dataset: F]

id	var1
1	XXXX
1	XXXX
2	YYYY
2	YYYY

[Dataset: H]

id	var1	var2
1	ZZZZ	7777
1	XXXX	7777
2	ZZZZ	9999
2	YYYY	9999

[Dataset: G]

id	var1	var2
1	ZZZZ	7777
2	ZZZZ	9999

```
/* プログラム */
data H ;
    merge F G ;
    by id ;
run ;
```

PDVに対する対策

Program Data Vector

基本的には変数の上書きを行わない。

上書きする場合は、PDVの処理を考慮して変数を初期化する

```
data E_ ;
     set A(in = A) B ;
     if var1 = 1 then var2 = 0 ;
       /* var2がないdataset:Aはvar2を初期化 */
       else if A then var2 = . ;
     if var1 = 1 then var3 = 0 ;
run ;
```

変数が重複する状態でのMERGE

基本的には変数が重複した状態でMergeしない。

「options msglevel = i ;」を指定することにより、
ログに重複した変数の情報を表示できる。

```
79          options msglevel=i;
80          data H ;
81          merge F G ;
82          by id ;
83          run ;

INFO:  変数 var1 (データセット WORK.F) はデータセットWORK.Gによって上書きされます。
NOTE: データセットWORK.Fから4オブザベーションを読み込みました。
NOTE: データセットWORK.Gから2オブザベーションを読み込みました。
```

おわりに

1. はじめに
2. エラー発見・解消テクニック
3. 浮動小数点の丸め誤差
4. Program Data Vector
5. おわりに

おわりに

SAS OnDemand for Academicsなどをはじめとする最新のSASでは、エラー、警告などの数をカテゴリで表示できるようになった。

しかし、本稿で紹介した事例にようにログのNOTEでも注意すべきメッセージでも多くある。
意図した結果を得るには、まずはログの確認が重要である。

おわりに

SAS初心者は、構文を学習することに重点を置くことが多い。
しかし、SASが、どのようにデータを処理するかを理解せずに
プログラミングすると意図した結果を得られないことがある。
正しい結果を作成できるように本稿で紹介したようなSAS内部
のデータ処理に対しても、理解を深めていただきたい。

この論文が、皆さまのSAS学習の一助になれば幸いです。

参考文献

1. SAS 9.4 Language Reference: Concepts, Sixth Edition
< https://support.sas.com/documentation/cdl/en/lrcon/69852/HTML/default/viewer.htm#titlepage.htm>
(参照: 2017-6-19)
2. SAS Technical News
< http://www.sas.com/offices/asiapacific/japan/periodicals/technews/>
(参照: 2017-6-19)
3. SAS 9.4関数とCALLルーチン リファレンス, 第4版
<http://support.sas.com/documentation/cdl_alternate/ja/lefunctionsref/67960/HTML/default/titlepage.htm>
(参照: 2017-6-19)
4. SAS FAQ
<https://www.sas.com/ja_jp/support/technical/faq.html>
(参照: 2017-6-19)
5. Essentials of the Program Data Vector (PDV): Directing the Aim to Understanding the DATA Step!
Arthur Xuejun Li, City of Hope National Medical Center, Duarte, CA
SAS Global Forum 2013, Paper 125-2013
6. The Use and Abuse of the Program Data Vector
Jim Johnson, Ephicacy Corporation, North Wales, PA, USA
SAS Global Forum 2012, Paper 255-2012

ご清聴ありがとうございました。

連絡先:

井上 貴博 (ノバルティス ファーマ株式会社)

takahiro.inoue@novartis.com