

統計解析システムSAS概説(3)

武富, 敬
九州大学大型計算機センター研究開発部

大賀, 豊満
九州大学大型計算機センター業務掛

石田, いつ子
九州大学大型計算機センター研究開発部

永井, 徳仁
九州大学大型計算機センター業務掛

他

<https://doi.org/10.15017/1474964>

出版情報 : 九州大学大型計算機センター広報. 14 (4), pp. 491-514, 1981-12-10. 九州大学大型計算機センター
バージョン :
権利関係 :

統計解析システム SAS 概説 (3)

武富 敬*, 大賀 豊満**, 石田 いつ子*, 永井 徳仁**, 遠矢 真知子*, 景川 耕宇*

今回、統計解析結果の手軽な図表出力、時系列解析の充実などを目的として、SASと関連した他のソフトウェアプロダクトであるSAS/GRAPH, SAS/ETS (Econometric and Time Series Library)を導入した。これによって、SASを含むデータ解析分野の環境が更に充実するものと期待される。その公開については、センターニュース等でお知らせするつもりなので、注意して頂きたい。また、SAS自体も、新機能(％INCLUDEによるステートメントの取り込み、Full-Screen Optionの機能など)の開発や新しい種々のオプション・SAS関数の追加等に見られるように、まだまだ発展し続けるシステムのようなのである。これらの様子は、SAS Institute社の発行する季刊の「SAS Communications」***等で窺い知ることができるが、これらについても随時お知らせしていくつもりである。

12では、SASデータセットの管理機能について述べる。SET, MERGE, UPDATEの各ステートメントの機能、付随したオプション、使用例などについて概説する。これらのステートメント(及びそれに続くプログラムステートメント)が与える、データの検索・抽出、分割・合成などの機能は、SASを単純な関係型(relational)データベースを扱うデータベース管理システムと見なすことができることを示す。13では、データ解析においては避けられない欠損値の取り扱いについて述べる。レポート作成などに使用されるFILE, PUTステートメントについては、次の機会に述べることにし、以上でDATAステップに関する概説は、一応終りとする[1, 2]。次章から、PROCステップ、すなわち各SASプロシジャの使用法の概説に移る。まず、14では、PROCステップについて概観する。15では、他の統計パッケージとのインターフェイスをとるためのCONVERTプロシジャについて述べる。ここでは、特にSPSSとのインターフェイス(SPSSシステムファイルのSASデータセットへの変換)に限って述べる。16では、SASユーティリティズについて概説する。一般的なユーティリティとして、SORT, PRINTの各プロシジャについて述べ、SASデータセット処理のためのユーティリティとして、CONTENTS, COPY, DELETE, DATASETSの各プロシジャについて述べる。

なお、SAS/GRAPHについて概要を知りたい方は、名大大型計算機センターニュースの解説[6]を参照されるとよい。

12. SASデータセット管理

実際のデータ解析にあたっては、様々の試行錯誤の繰り返しや、データの多面的解析のために、データのサブセット化、データセットの連結、多数のデータセットのマージ(併合)、データの更新な

* 九州大学大型計算機センター 研究開発部

** 同 業務掛

*** 「SAS ユーザズグループ」の会合で配布している資料の一つである。SAS, SAS/GRAPH, SAS/ETSの現在の状況についてのNewsletter。

どの作業が不可欠である。これらデータセットの操作のための機能を与えるのが、これから述べる SET, MERGE, UPDATE ステートメントである。

文献[3]で述べたように、SASで取り扱うデータは、原則として行方向にオブザベーション、列方向に変数を配列した行列構造（表の形式）をとるので、単純な関係モデルに属するデータを取り扱うものと見なしてもよい。それら単純な関係型データベースにおける、データの検索・抽出、分割・合成などの機能を与えるものが、ここに述べるステートメント（及びそれに続くプログラムステートメント）と考えることも可能である。こうして、SASを単純な関係型データベースを扱うデータベース管理システムとして捉える見方は、実際に、本センターの利用者のあるグループによる“医療情報データベース”構築へのSASの利用として応用され、一応の成果が得られている[4]。更に付け加えると、SASには、データベース保護（protection）の機能もある程度備わっていて、データセットの使用や消去に関する保護が可能である（マニュアル[5]のAppendix 10を参照）。また、この場合にはそのデータベースの解析には、アプリケーションプログラム作成などは必要なく、SASプロシジャを適用することにより容易にできる訳である。

1.2.1 概 要

SET, MERGE, UPDATE各ステートメントの機能の概略は、次のとおりである。

SET …… 1つ又はそれ以上のSASデータセットから1行ずつオブザベーションを入力する。

MERGE …… 2つ又はそれ以上のSASデータセットからのオブザベーションを併合する。

UPDATE …… マスターデータセットをトランザクションデータセットにより更新する。

ここで注意すべきことは、これらのステートメントの取り扱う対象がSASデータセットに限られることである。よって、OSデータセットのままのデータは、5で述べた方法によりまずSASデータセットに変換することが先決である。

次に、これらのステートメントに付随して使用されるオプション及びそれに伴う特有な変数について述べよう。これらの変数の値をプログラムステートメントで用いることにより、更に柔軟なデータセット操作が可能である。ただし、これら変数の値そのものは、SASデータセットに出力されることはない。

1) IN=オプション

形式は、以下のようである。

SASデータセット名 (IN=変数名) ;

INの後ろに指定された変数の値は、現在のオブザベーションがどのデータセットから来ているかを示し、INの前のSASデータセットに対応するものであれば1、そうでなければ0の値を、SASが自動的に設定する。データセットの同定などに使用できる（1.2.2の例3参照）。

2) END=オプション

形式は、以下のようである。

SASデータセット名 END = 変数名 ;

ENDの後ろに指定された変数の値は、SAS データセット全体に関するオブザベーションの終了、すなわちEOF(End Of File)を検出して知らせるものであり、EOFならば1、そうでなければ0の値を、SASが自動的に設定する(12.2例3参照)。入力終了時の処理(例えば、入力したある変数の総合計の出力など)に有効である。

3) BYステートメントに付随した2つの変数(FIRST変数、LAST変数)

BYステートメントは、サブセット毎の処理をしたい時にキーとなる変数の指定に用いる。例えば、SEXという変数が▼F▼と▼M▼の2つの値をとり、ソーティングによりサブセット化されているとする。この時、BY SEX;と指定することにより、▼F▼、▼M▼それぞれのサブセット毎の処理が可能である。このサブセットのことをBYグループとも呼ぶ。

SASでは、このBYステートメントが現れると、そこに指定された変数毎に、各BYグループの最初と最後を示す次の2つの変数が作成され、値が自動的に設定される。

FIRST. by 変数名各BYグループの最初のオブザベーションが現れるごとに1、それ以外は0の値をとる。

LAST. by 変数名各BYグループの最後のオブザベーションが現れるごとに1、それ以外は0の値をとる。

ここで、by 変数名とは、BYステートメントで指定した各々の変数名のことである(12.2の例4参照)。

12.2 SETステートメント

SETステートメントは、SAS データセットの入力による、データの追加、削除、サブセット化、SAS データセットの連結などのために用いられる。以下に形式を示す。

SET SAS データセット名[(IN = 変数名)] [**END** = 変数名] ;

これは、SPSSのGET FILEカードの機能に対応する。SAS データセットを複数指定した時に、この過程で作られる全体のSAS データセットは、指定した各々に含まれていたすべての変数を含むものが作成される。この時、対応するものがない変数の値は、欠損値として作成される。

例1.

<pre> alloc f(save) da(sasplib.data) old① ready sassort ready sas 1? data subcirc1 ; 2? set save.circle ;② 3? if sex = 'm' ;③ 4? avg = mean(x1,x2) ;④ 5? drop pulse x1 x2 x3 ;⑤ 6? proc print ;⑥ 7? title data set subcirc1 ; 8? run; </pre>	<pre> DATA SET CIRCLE OBS ID NAME AGE SEX WEIGHT PULSE X1 X2 X3 1 310 YAMAMOTO 27 F 62.9 64 36 . D 2 310 HARA 17 M 49.1 52 92 87 A 3 210 HAGASHI 19 F 58.5 68 100 96 A 4 110 TOYODA 45 M 90.6 72 73 79 B 5 210 KOHNO 13 F 28.4 64 45 60 D 6 110 YAMAKURA 25 M 50.2 62 69 58 C 7 110 SAITOH 66 M 68.4 68 40 . D 8 310 EHOTO 24 F 42.5 66 . 100 A 9 110 NAKAHATA 58 M 61.1 60 64 72 C 10 210 KOHATU 19 F 33.8 60 90 68 B 11 210 NAKAHISI 60 F 52.7 58 84 80 B 12 310 KAKEFU 42 M 49.2 68 60 . C 13 310 AWAGUTI 31 F 56.5 50 . 32 D 14 310 KOBAYASHI 51 M 58.6 50 81 58 C 15 110 MATUBARA 36 F 38.8 64 43 . D 16 310 HISINOTO 65 M 72.3 64 76 82 B 17 110 HATUOKA 33 F 42.7 64 49 24 D </pre>
--	--

図15. SAS データセットCIRCLEの内容

①で既存の SAS データライブラリ SASDLIB.
DATA を DD 名 SAVE で割り当てた。②で、その中に
含まれる SAS データセット CIRCLE を、SET ステ
ートメントにより入力した。図 15 に、その内容を示
す。③で、SEX の値が ▼M▼ のものだけ取り出し、
オブザベーションに関するサブセット化を行った。
④で、入力変数 X 1, X 2 の平均を計算し、新変数
AVG を生成して、変数の付加を行った。⑤で、
DROP ステートメントにより、変数 PULSE, X 1 から X 3 までを落し、変数に関するサブセット化
を行った。⑥で、データセット変容後の内容を出力した。それを図 16 に示す。図 15 と比較すれば、
データ操作の内容が読み取れるであろう。

例 2.

```
17 data sortcirc ;
27 set save.circle ;
37 proc sort ; .....①
47 by id ;
57 run ;
67 proc print ;
77 title data set sortcirc ;
87 run ;
```

例 1 と同じく、SAS データセット
CIRCLE を入力し、①で、変数 ID
をキーとしてソーティングを行った。
その結果を図 17 に示す。

DATA SET SUBCIRCL

OBS	ID	NAME	AGE	SEX	WEIGHT	AVG
1	310	HARA	17	M	49.1	89.5
2	110	TOYODA	45	M	90.6	76.0
3	110	YAMAKURA	25	M	50.2	63.5
4	110	SAITOH	66	M	68.4	40.0
5	110	NAKAHATA	58	M	61.1	68.0
6	310	KAKEFU	42	M	49.2	60.0
7	310	KOBAYASI	51	M	58.6	69.5
8	310	NISIMOTO	65	M	72.3	79.0

図 16. 例 1 により、行、列方向のサブセ
ット化などを行った後の SAS デー
タセット SUBCIRCL の内容

DATA SET SORTCIRC

OBS	ID	NAME	AGE	SEX	WEIGHT	PULSE	X1	X2	X3
1	110	TOYODA	45	M	90.6	72	73	79	B
2	110	YAMAKURA	25	M	50.2	62	69	58	C
3	110	SAITOH	66	M	68.4	68	40	.	D
4	110	NAKAHATA	58	M	61.1	60	64	72	C
5	110	HATUBARA	36	F	38.8	64	43	.	D
6	110	HATUOKA	33	F	42.7	64	49	24	D
7	210	HAGASIMA	19	F	58.5	68	100	96	A
8	210	KOHNO	13	F	28.4	64	45	60	D
9	210	KOHATU	19	F	33.8	60	90	68	B
10	210	NAKANISI	60	F	52.7	58	84	80	B
11	310	YAMAHOTO	27	F	62.9	64	36	.	D
12	310	HARA	17	M	49.1	52	92	87	A
13	310	EHOTO	24	F	42.5	66	.	100	A
14	310	KAKEFU	42	M	49.2	68	60	.	C
15	310	AWAGUTI	31	F	56.5	50	.	32	D
16	310	KOBAYASI	51	M	58.6	50	81	58	C
17	310	NISIMOTO	65	M	72.3	64	76	82	B

図 17. 図 15 で示した CIRCLE について、変数 ID でソ
ーティングした結果の SAS データセット SORTCIRC
の内容

```
97 data save.malcircl save.femcircl ; .....②
107 set sortcirc ;
117 if sex = 'm' then output save.malcircl ; .....③
127 if sex = 'f' then output save.femcircl ; .....④
137 drop x1 x2 x3 ;
147 proc print data=save.malcircl ;
157 title data set malcircl ;
167 run ;
187 proc print data=save.femcircl ;
197 title data set femcircl ; run ;
```

②で、前のステップで作成した SAS データ
セット SORTCIRC を入力した。③で、
OUTPUT ステートメントにより、SEX の値
ごとに、MALCIRCL および FEMCIRCL
という 2 つの SAS データセットに出力した。
この際、④で、DROP ステートメントにより、
不必要な変数は削除した。なお、これら 2 つ
のデータセットは保存したいため、2 レベル

DATA SET MALCIRCL

OBS	ID	NAME	AGE	SEX	WEIGHT	PULSE
1	110	TOYODA	45	M	90.6	72
2	110	YAMAKURA	25	M	50.2	62
3	110	SAITOH	66	M	68.4	68
4	110	NAKAHATA	58	M	61.1	60
5	310	HARA	17	M	49.1	52
6	310	KAKEFU	42	M	49.2	68
7	310	KOBAYASI	51	M	58.6	50
8	310	NISIMOTO	65	M	72.3	64

DATA SET FEMCIRCL

OBS	ID	NAME	AGE	SEX	WEIGHT	PULSE
1	110	HATUBARA	36	F	38.8	64
2	110	HATUOKA	33	F	42.7	64
3	210	HAGASIMA	19	F	58.5	68
4	210	KOHNO	13	F	28.4	64
5	210	KOHATU	19	F	33.8	60
6	210	NAKANISI	60	F	52.7	58
7	310	YAMAHOTO	27	F	62.9	64
8	310	EHOTO	24	F	42.5	66
9	310	AWAGUTI	31	F	56.5	50

図 18. 図 17 で示した SORTCIRC を、2 つの
SAS データセット MALCIRCL,
FEMCIRCL に分離した例

ネームの指定をしている。これらは共に、例1で割り当てたSASデータライブラリに出力されて保存される。

例3. SASデータセットの連結 (concatenation)

```
1? data totcircl ;
2?   set save.malcircl(in=in1)
3?     save.femcircl(in=in2) end=e1 ; .....①
4?   inn = in1 ; inf = in2 ; eof = e1 ; .....②
5? proc print ;
6?   title data set totcircl ;
7? run ;
```

①で、例2で作成した2つのSASデータセットMALCIRCLとFEMCIRCLを続けて指定して入力した。このように、SETステートメントに複数のSASデータセットを指定すると、SASデータセットの連結が行われる。この時、INオプションおよびENDオプションを指定している。前述したように、これらオプションに付随した変数の値はSASデータセットに出力されることはないで、それらの値がどうなっているか例示するために、②で、他の変数に置き換えた。

図19に、連結の結果を示す。INN, INFの値から分かるように、SETステートメントで指定した左から右へのデータセットの順序に連結される。

DATA SET TOTCIRCL									
OBS	ID	NAME	AGE	SEX	WEIGHT	PULSE	INN	INF	EOF
1	110	TOYODA	45	M	90.6	72	1	0	0
2	110	YAMAKURA	25	M	50.2	62	1	0	0
3	110	SAITOH	66	M	68.4	68	1	0	0
4	110	NAKAHATA	58	M	61.1	60	1	0	0
5	310	HARA	17	M	49.1	52	1	0	0
6	310	KAKEFU	42	M	49.2	68	1	0	0
7	310	KOBAYASI	51	M	58.6	50	1	0	0
8	310	NISINOTO	65	M	72.3	64	1	0	0
9	110	HIATUBARA	36	F	38.8	64	0	1	0
10	110	HATUOKA	33	F	42.7	64	0	1	0
11	210	NAGASIMA	19	F	58.5	68	0	1	0
12	210	KOHNO	13	F	28.4	64	0	1	0
13	210	KOHATU	19	F	33.8	60	0	1	0
14	210	NAKANISI	60	F	52.7	58	0	1	0
15	310	YAMANOJO	27	F	62.9	64	0	1	0
16	310	ENOTO	24	F	42.5	66	0	1	0
17	310	AWAGUTI	31	F	56.5	50	0	1	1

図19. 図18で示した2つのSASデータセットを連結した結果のSASデータセットTOTCIRCLの内容

例4. SASデータセットの差し込み (interleaving)

```
1? data intcircl ;
2?   set save.malcircl(in=in1)
3?     save.femcircl(in=in2) ; .....①
4?   by id ;
5?   inn = in1 ; inf = in2 ;
6?   first = first.id ; last = last.id ; .....②
8? proc print ;
9?   title data set intcircl ;
10? run ;
```

例3と同じように、①で2つのSASデータセットを入力したが、この際、更にBYステートメントに変数IDを指定している。このようにBYステートメント付きで、SETステートメントを実行すると、SASデータセットの差し込みが行われる。②で、このBYステートメントに付随したFIRSTおよびLAST変数の値を例示するために、INオプションの変数と共に、変数の置き換えを行った。図20に、差し込みの結果を示す(図19と比較のこと)。INN, INF, FIRST, LASTの値から分かるように、差し込みとは、BYステートメントで指定したキー変数についてのソーティングの順序を保

存するように行う一種の“連結”のことである。当然、SETにより入力する複数のSASデータセットは、BYステートメントに指定するキー変数により、前もってソーティングされていなければならない。

12.3 MERGE ステートメント

MERGE ステートメントは、複数のSASデータセットを入力して併合し、それらに含まれていた変数をすべて持つような1つの新しいSASデータセットを作成するのに用いる。おおまかには、SETステートメントによる連結が、オブザベーション方向にデータ（表の形式）を拡大するのに対し、MERGEステートメントによる併合は、変数方向に表を拡大するものと考えておいてよい。こうし

て、ある表には含まれていて、ある表にはないような情報は、このMERGEで合成することにより、望むすべての情報を含む表を完成することができる訳である。以下に形式を示す。

```

MERGE SASデータセット名[ (IN=変数名1) ]
      SASデータセット名[ (IN=変数名2) ]
      :
      [ END=変数名 ] ;

```

SASでは、50個のSASデータセットのマージ（併合）まで許されている。

MERGEステートメントによる併合の仕方は、これに続いてBYステートメントが現れるか否かで、次の2種に分けられる。

i) 1対1マージ (one-to-one merge) ……後にBYステートメントを伴わない場合。この時は、両者のオブザベーションの順序（一方の1番目は、他方の1番目というように）に併合が行われる。そのため、両者のオブザベーションが厳密に1対1に対応していることが必要で、例えば、一方のオブザベーションが、他方にはない余分のオブザベーションや重複したオブザベーションを含んでいたりとすると、MERGEにより作成されたデータの整合性は完全に失われる。

ii) マッチマージ (match merge) ……必ず後にBYステートメントを伴う。この時、両者で併合の共通のキーとなる変数を、BYステートメントにより指定する、すなわち、そのような変数が存在しなければならない。また、この場合に入力されるSASデータセットは、その共通のキー変数により前もってソーティングされていなければならない。この場合の併合は、BYグループを意識して行われるので、余分のオブザベーションや重複したオブザベーションがあっても、データの整合性が失われることはない。

DATA SET INTCIRCL

OBS	ID	NAME	AGE	SEX	WEIGHT	PULSE	INM	INF	FIRST	LAST
1	110	TOYODA	45	M	90.6	72	1	0	1	0
2	110	YANAKURA	25	M	50.2	62	1	0	0	0
3	110	SAITOH	66	M	68.4	68	1	0	0	0
4	110	NAKAHATA	58	M	61.1	60	1	0	0	0
5	110	HATUBARA	36	F	38.8	64	0	1	0	0
6	110	HATUOKA	33	F	42.7	64	0	1	0	1
7	210	NAGASIMA	19	F	58.5	68	0	1	1	0
8	210	KOHNO	13	F	28.4	64	0	1	0	0
9	210	KOHATU	19	F	33.8	60	0	1	0	0
10	210	NAKANISI	60	F	52.7	58	0	1	0	1
11	310	HARA	17	M	49.1	52	1	0	1	0
12	310	KAKEFU	42	M	49.2	68	1	0	0	0
13	310	KOBAYASI	51	M	58.6	50	1	0	0	0
14	310	NISHIMOTO	65	M	72.3	64	1	0	0	0
15	310	YAMAHOTO	27	F	62.9	64	0	1	0	0
16	310	ENOTO	24	F	42.5	66	0	1	0	0
17	310	AWAGUTI	31	F	56.5	50	0	1	0	1

図2.0. 図1.8で示した2つのSASデータセットの差し込みを行った結果のSASデータセットINTCIRCLの内容

一般には、併合するデータセットの間で、余分なオブザベーションや重複したオブザベーションがあって、両者でオブザベーション数が異なるのが普通であろう。そこで、MERGE ステートメントは、後に BY ステートメントを伴うマッチマージで使うものと考えておいた方がいいであろう。なお、キー変数以外にも両者で共通の変数が存在する場合には、MERGE ステートメントに指定した最も右側のデータセットからの値が採られる（12.4 の UPDATE ステートメントと異なる）。

例 1. 1 対 1 マージ

```
17 data onemerge ;
27 merge onecircl twocircl ; ..... ①
37 proc print ;
47 title data set onemerge ;
57 run ;
```

①で、図 21 で示した SAS データセット

ONECIRCL と、図 22 で示した TWOCIRCL を 1 対 1 マージした。その結果を図 23 に示す。

この例では、併合される両者のオブザベーション数は等しくない（図 21、図 22）が、たまたま余分なオブザベーションが TWOCIRCL の最後にあったために、うまく併合できた。図 23 より、このように対応するオブザベーションがない変数の値（この場合は変数 AGE と SEX）は、欠損値として作成されることが分かる。

DATA SET ONECIRCL			DATA SET TWOCIRCL					DATA SET ONEMERGE									
OBS	NAME	AGE SEX	OBS	NAME	WEIGHT	PULSE	X1	X2	X3	OBS	NAME	AGE SEX	WEIGHT	PULSE	X1	X2	X3
1	YAMAHOTO	27 F	1	YAMAHOTO	62.9	64	36	.	D	1	YAMAHOTO	27 F	62.9	64	36	.	D
2	HARA	17 H	2	HARA	49.1	52	92	87	A	2	HARA	17 H	49.1	52	92	87	A
3	NAGASIMA	19 F	3	NAGASIMA	58.5	68	100	96	A	3	NAGASIMA	19 F	58.5	68	100	96	A
4	TOYODA	45 H	4	TOYODA	90.6	72	73	79	B	4	TOYODA	45 H	90.6	72	73	79	B
5	KOHNO	13 F	5	KOHNO	28.4	64	45	60	D	5	KOHNO	13 F	28.4	64	45	60	D
6	YAMAKURA	25 M	6	YAMAKURA	50.2	62	69	58	C	6	YAMAKURA	25 M	50.2	62	69	58	C
7	SAITOH	66 H	7	SAITOH	68.4	68	40	.	D	7	SAITOH	66 H	68.4	68	40	.	D
8	EHOTO	24 F	8	EHOTO	42.5	66	.	100	A	8	EHOTO	24 F	42.5	66	.	100	A
9	NAKAHATA	58 H	9	NAKAHATA	61.1	60	64	72	C	9	NAKAHATA	58 H	61.1	60	64	72	C
10	KOHATU	19 F	10	KOHATU	33.8	60	90	68	B	10	KOHATU	19 F	33.8	60	90	68	B
11	NAKANISI	60 F	11	NAKANISI	52.7	58	84	80	B	11	NAKANISI	60 F	52.7	58	84	80	B
12	KAKEFU	42 H	12	KAKEFU	49.2	68	60	.	C	12	KAKEFU	42 H	49.2	68	60	.	C
			13	AWAGUTI	56.5	50	.	32	D	13	AWAGUTI	.	56.5	50	.	32	D

図 21. SAS データセット
ONECIRCL の内容

図 22. SAS データセット
TWOCIRCL の内容

図 23. 図 21 の ONECIRCL と図 22 の
TWOCIRCL を 1 対 1 マージした
結果の SAS データセット
ONEMERGE の内容

例 2. マッチマージ

```
17 data matmerge ;
27 merge save.circlex(in=in1)
37 save.circley(in=in2) ; ..... ①
47 by name ;
57 inx = in1 ; iny = in2 ;
67 first = first.name ; ..... ②
77 last = last.name ;
87 run ;
97 proc print ;
107 title data set matmerge ;
117 run ;
```

①で、図 24 で示した保存 SAS データセット

CIRCLEX と、図 25 で示した CIRCLEY を、変数 NAME を併合の共通のキーとしてマッチマージした。前述したように、併合される両者は、共通のキー変数 NAME でソーティングされている。②で、IN オプションの変数および FIRST

LAST 変数の値を例示するために、変数の置き換えを行った。併合の結果を図 26 に示す。この場合も、対応するオブザベーションがない変数の値は、欠損値となっていることが分かる。

DATA SET CIRCLEX

OBS	NAME	PULSE	X1
1	HARA	52	92
2	HARA	61	84
3	HARA	68	42
4	KOHNO	64	45
5	NAGASIMA	68	100
6	TOYODA	72	73
7	TOYODA	57	65
8	YAMAKURA	62	69

DATA SET CIRCLEY

OBS	NAME	X2	X3
1	HARA	87	A
2	HARA	56	C
3	KOHNO	60	D
4	KOHATU	68	B
5	NAKAHATA	72	C
6	NAKANISI	80	B
7	YAMAKURA	58	C

DATA SET HATHERGE

OBS	NAME	PULSE	X1	X2	X3	INX	INY	FIRST	LAST
1	HARA	52	92	87	A	1	1	1	0
2	HARA	61	84	56	C	1	1	0	0
3	HARA	68	42	56	C	1	1	0	1
4	KOHNO	64	45	60	D	1	1	1	1
5	KOHATU	.	68	B	0	1	1	1	1
6	NAGASIMA	68	100	.	0	1	0	1	1
7	NAKAHATA	.	72	C	0	1	1	1	1
8	NAKANISI	.	80	B	0	1	1	1	1
9	TOYODA	72	73	.	1	0	1	1	0
10	TOYODA	57	65	.	1	0	0	1	1
11	YAMAKURA	62	69	58	C	1	1	1	1

図 24. SAS データセ
ット CIRCLEX
の内容図 25. SAS データセ
ット CIRCLEY
の内容図 26. 図 24 の CIRCLEX と 図 25 の CIRCLEY
をマッチマージした結果の SAS データセット
MATMERGE の内容

例 3. 米国の 1952 年から現在までの大統領選挙の結果を納めた SAS データセット ELECTION (図 27) と、各候補者の政党・出身の州を納めた SAS データセット CANDIDAT (図 28) がある。これから、この 2 つの表を併合して、大統領選の当選者の政党・出身州を求めることを考えよう。

```

17 proc sort data=save.election ;
27   by name ;
37   run ;
47 proc sort data=save.candidat ; .....①
57   by name ;
67   run ;

77 data wselect ;
87   merge save.election(in=in1)
97         save.candidat(in=in2) ; .....②
107  by name ;
117  ine = in1 ; inc = in2 ;
127  run ;

187 proc print ;
197   title data set matelect ;
207  run ;

```

DATA SET ELECTION

OBS	YEAR	NAME	LOSER
1	1952	EISENHOWER	STEVENSON
2	1956	EISENHOWER	STEVENSON
3	1960	KENNEDY	NIXON
4	1964	JOHNSON	GOLDWATER
5	1968	NIXON	HUMPHREY
6	1972	NIXON	MCGOVERN
7	1976	CARTER	FORD
8	1980	REAGAN	CARTER

DATA SET CANDIDAT

OBS	NAME	PARTY	STATE
1	EISENHOWER	R	TEXAS
2	STEVENSON	D	ILL
3	NIXON	R	CALIF
4	GOLDWATER	R	ARIZ
5	KENNEDY	D	MASS
6	JOHNSON	D	TEXAS
7	HUMPHREY	D	MINN
8	MCGOVERN	D	S.DAK
9	FORD	R	MICH
10	CARTER	D	GA
11	REAGAN	R	CALIF

図 27. SAS データセ
ット ELECTION
の内容図 28. SAS データセ
ット CANDIDAT
の内容

まず、①でマージの前準備として、両者のデータセットを変数 NAME (併合の共通のキーとなる) でソーティングした。SORT プロシジャで OUT = オプションが指定されていないため、ソートされた結果がそのまま元のデータセットに格納され保存される (16.1.1 参照)。②で、これらソートした両者のデータセットを、変数 NAME をキーとしてマッチマージした。その結果を図 29 に示す。IN オプションの変数の値も例示されている。しかし、これでは大統領選の敗者も現れて、表としては見にくい。これをなくすためには、以下のようにすればよい。

DATA SET MATELECT

OBS	YEAR	NAME	LOSER	PARTY	STATE	INE	INC
1	1976	CARTER	FORD	D	GA	1	1
2	1952	EISENHOWER	STEVENSON	R	TEXAS	1	1
3	1956	EISENHOWER	STEVENSON	R	TEXAS	1	1
4	.	FORD	.	R	MICH	0	1
5	.	GOLDWATER	.	R	ARIZ	0	1
6	.	HUMPHREY	.	D	MINN	0	1
7	1964	JOHNSON	GOLDWATER	D	TEXAS	1	1
8	1960	KENNEDY	NIXON	D	MASS	1	1
9	.	MCGOVERN	.	D	S.DAK	0	1
10	1968	NIXON	HUMPHREY	R	CALIF	1	1
11	1972	NIXON	MCGOVERN	R	CALIF	1	1
12	1980	REAGAN	CARTER	R	CALIF	1	1
13	.	STEVENSON	.	D	ILL	0	.

図 29. マッチマージした結果の SAS データセ
ット MATELECT の内容

```

21? data velect ;
22?   merge save.election(in=ine)      .....③
23?       save.candidat ;
24?       by name ;
25?       if ine ;                    .....④
26? run ;
27? proc print ;
28?   title data set netelect ;
29? run ;

```

DATA SET NETELECT

OBS	YEAR	NAME	LOSER	PARTY	STATE
1	1976	CARTER	FORD	D	GA
2	1952	EISENHOWER	STEVENSON	R	TEXAS
3	1956	EISENHOWER	STEVENSON	R	TEXAS
4	1964	JOHNSON	GOLDWATER	D	TEXAS
5	1960	KENNEDY	NIXON	D	MASS
6	1968	NIXON	HUMPHREY	R	CALIF
7	1972	NIXON	HCGOVERN	R	CALIF
8	1980	REAGAN	CARTER	R	CALIF

図 30. 図 29 の MATELECT について、当選者のみ取り出したサブセット NETELECT の内容

③で、IN=オプションを指定してマッチマージした。④で、その中でデータセット ELECTION に含まれていたオブザベーションのみ取り出した (IF INE は、IF INE = 1 と同じ)。その結果を図 30 に示す。なお、これを YEAR についてソートすれば、図 31 に示す最終的な表が得られる。

DATA SET PRESIDEN

OBS	YEAR	NAME	LOSER	PARTY	STATE
1	1952	EISENHOWER	STEVENSON	R	TEXAS
2	1956	EISENHOWER	STEVENSON	R	TEXAS
3	1960	KENNEDY	NIXON	D	MASS
4	1964	JOHNSON	GOLDWATER	D	TEXAS
5	1968	NIXON	HUMPHREY	R	CALIF
6	1972	NIXON	HCGOVERN	R	CALIF
7	1976	CARTER	FORD	D	GA
8	1980	REAGAN	CARTER	R	CALIF

図 31. 図 30 の NETELECT を、YEAR についてソーティングした最終結果の表

12.4 UPDATE ステートメント

UPDATE ステートメントは、MERGE ステートメントの 1 つの特別な型を与えるものと見なされる。すなわち、マスターデータセット (以後、マスターと略称) とトランザクションデータセット (以後、トランザクションと略称) を併合して、データを更新したり、追加したり、削除したりするのに用いられる。以下に形式を示す。

UPDATE マスターデータセット名 [(IN = 変数名 1)] トランザクションデータセット名 [(IN = 変数名 2)] [END = 変数名] ;
--

UPDATE ステートメントは、必ず後に BY ステートメントを伴い、その BY ステートメントに併合の共通のキーとなる変数を指定する。そのため、マスターもトランザクションも、前もってそのキー変数でソーティングされている必要がある。

このように、UPDATE は MERGE と類似しているにも拘らず、1 つの独立したステートメントとして存在しているのは、以下の点が異なるからである。

i) MERGE では、50 個までの SAS データセットが対象となるのに対し、UPDATE では、マスターとトランザクションの 2 個の SAS データセットしか扱わない。これは、ベースとなる大量のマスターを、それに比較して少量であるトランザクションにより更新するという考え方から来ている。

ii) MERGE では、前述したように、キー変数以外の両者に共通な変数に対しては、最も右側のデータセットからの値 (たとえ、それが欠損値でも) が採られた。一方、UPDATE では、マスターの値が更新されるのは、トランザクションが欠損値以外の値を持つ時であり、トランザクションにある欠損値は、マスターの値をそのまま生かすために用いられる。

iii)MERGEでは、マッチマージの指定をする限り、オブザベーションに重複があるのは構わなかった。UPDATEでは、キー変数に対しマスターが重複した値を持つことは許されない。ただし、トランザクションの方は重複して構わない。それは、UPDATEにおけるマスターとトランザクションの結合の仕方が、以下のように、MERGEとは異なっているからである。キー変数に対して、トランザクションに重複したオブザベーションがあっても、UPDATEは、それらの値を順次マスターに適用して更新し、最終的にすべてを更新した結果の1オブザベーションしか出力しない(例参照)。一方、MERGEの場合は、重複したBYグループの数だけのオブザベーションが出力される。

UPDATEステートメントが特に有用なのは、定期的にトランザクションが産み出され、それでマスターを更新しなければならないような場合であろう。それ以外で、特にデータの修正・更新が少量の場合は、7で述べたEDITORプロシジャを用い、TSSでSASデータセットの編集を行うことで十分間に合うようである[4]。

例 マスターデータセットMASTCLEを、トランザクションデータセットTRANSCLにより更新する。

```
17 data newmast ;
27 update save.mastcle
37 save.transcl ; .....①
47 by name ;
57 run ;
67 proc print ;
77 title data set newmast ;
87 run ;
```

マスターMASTCLEの内容を図32に、トランザクションTRANSCLの内容を図33に示す。①で、これら両者を変数NAMEを共通のキーとして併合し、マスターを更新した。その結果を図34に示す。

トランザクションTRANSCLの最初のオブザベーション(NAME=▼AKAMATU▼)は、マスターに追加されている。このように、トランザクションにあってマスターにないオブザベーションは追加される。次の2, 3番目のオブザベーション(NAME=▼HARA▼)は、順次マスターに適用され、マスターを更新する。すなわち、2番目で変数X1, X2を更新し、3番目で変数WEIGHTを更新している。4番目のオブザベーション(NAME=▼NAGASIMA▼)は、下線(―)で示されている特別な欠損値(13.1参照)を含んでいる。トランザクションに含まれる通常の欠損値(数変数に対してはピリオド, 文字変数に対しては空白で表示されている)は、前述したようにマスターに何ら影響を与えないが、特別な欠損値は、マスターのその部分を通常の欠損値とする。5番目のオブザベーション(NAME=▼YAMAMOTO▼)は、マスターの変数X2の欠損値を更新して数値を与えるために使われている。

DATA SET MASTCLE

OBS	NAME	SEX	WEIGHT	X1	X2
1	HARA	M	49.1	92	87
2	KAKEFU	M	49.2	60	.
3	KOHNO	F	28.4	45	60
4	KOHATU	F	33.8	90	68
5	NAGASIMA	F	58.5	100	96
6	NAKABATA	M	61.1	64	72
7	TOYODA	M	90.6	73	79
8	YAMAKURA	M	50.2	69	58
9	YAMAMOTO	F	62.9	36	.

図32. マスターデータセットMASTCLEの内容

DATA SET TRANSCL

OBS	NAME	SEX	WEIGHT	X1	X2
1	AKAMATU	M	78.5	.	68
2	HARA	.	.	87	92
3	HARA	.	52.8	.	.
4	NAGASIMA	―	―	.	―
5	YAMAMOTO	.	.	.	59

図33. トランザクションデータセットTRANSCLの内容

DATA SET NEWMAST

OBS	NAME	SEX	WEIGHT	X1	X2
1	AKAMATU	M	78.5	.	68
2	HARA	M	52.8	87	92
3	KAKEFU	M	49.2	60	.
4	KOHNO	F	28.4	45	60
5	KOHATU	F	33.8	90	68
6	NAGASIMA	.	.	100	.
7	NAKABATA	M	61.1	64	72
8	TOYODA	M	90.6	73	79
9	YAMAKURA	M	50.2	69	58
10	YAMAMOTO	F	62.9	36	59

図34. 図32のMASTCLEを、図33のTRANSCLにより更新した結果のSASデータセットNEWMASTの内容

1.3. 欠損値の取り扱い

統計データのほとんどは、幾つかのオブザベーションについて、ある変数の値が欠けている、すなわち欠損値 (missing values) を持つのが普通である。以下に、この場合の指定の仕方およびその取り扱いなどについて述べる。

1.3.1 数変数に対する欠損値

数変数に対するデータを入力する際に、欠損値の指定をしたい場合は、そのデータの含まれるカラムにピリオド (.) を書くか、または空白としておけばよい。ただし、リスト入力の場合は、空白がデータの区切りを表すので、ピリオドで指定する必要がある。あるいは、その変数の取りうる範囲外の数値を入れておき、後でプログラムステートメントにより条件判定をして、欠損値として指定する方法もある (例 1 参照)。SAS 内部では、数変数に対する欠損値は負の無限大として表現されているので、SORT プロシジャで昇順にソーティングすると、欠損値を含むデータが最初に現れる。式の中で数変数に対する欠損値を表すには、ピリオドで指定すればよい (例 1 参照)。

数変数に対しては、特別な欠損値 (special missing values) の指定が可能である。この指定には、英字 (A~Z) と下線 (_) の 27 個の文字を使用することができる。これらは、欠損値の間で区別をしたい時に用いることができ、そのために 27 種までの欠損値の区別が可能である。ただし、この場合は、必ず次の形式で MISSING ステートメントにより欠損値とする文字を指定しておく必要がある。

MISSING 特別な欠損値を表す文字 ;

式の中で特別な欠損値を表すには、それを表している文字の前にピリオドを付加して指定しなければならない (例 2 参照)。

例 1. DATA EX1 ;

INPUT X Y Z ;

IF X= . THEN DO ;①

(X が欠損値の場合の処理)

END ;

IF Y=999 THEN Y= . ;②

S1=(Y+Z)/2 ;③

S2=MEAN(Y,Z) ;④

①で、入力変数 X が欠損値か否かのデータチェックを行い、その場合は特別な処理を施すようにしている。②では、Y の値を調べて、値が 9 9 9 ならば欠損値とした。③と④は、いずれも Y と Z の平均を求めるものである。しかし、③では、Y、Z いずれかが欠損値ならば、その結果の S 1 も欠損値となるのに対し、④では、欠損値は除いて処理するので、S 2 は欠損値とはならない。S 1 のように一度欠損値となったものを、以後のステートメントで次々にデータ加工していくと、それが次々に伝播する、すなわち欠損値の伝播 (propagation of missing values) が起こる。それを防ぐためには、④のように対応する SAS 関数を用いるか、6 で述べたサムステートメントを用いるか、あるいは、

欠損値の可能性のあるものに対しては事前に前処理を施しておくようにしなければならない。

例2. 特別な欠損値を用いて欠損値の区別をする例。例えば、あるアンケート調査で、回答者が回答を拒否したために起った欠損値と、誤って回答したために起った欠損値を区別して集計、解析したい場合。

```
DATA EX2 ;
MISSING R I ;                      .....①
INPUT X Y ;
IF X= . R THEN DO ;                .....②
    ( Xが特別な欠損値Rの場合の処理 )
END ;
IF X= . I THEN DO ;                .....③
    ( Xが特別な欠損値Iの場合の処理 )
END ;
```

①で、特別な欠損値を表す文字として、拒否 (Refuse) の回答の場合に R, 誤った (Invalid) 回答の場合に I を指定した。②で、入力変数 X が R の値をとるか否かを調べ、その場合は特別な処理を施すようにしている。③も同様である。

1.3.2 文字変数に対する欠損値

文字変数の値を欠損値として入力するには、そのデータの部分を空白としておけばよい。ただし、リスト入力の場合は、1.3.1 で述べた通りである。SAS 内部では、文字変数に対する欠損値は空白として表現されているので、昇順のソーティングの場合、欠損値を含むデータが最初に現れる。式の中で文字変数に対する欠損値を表すには、空白を引用符で囲み、▼ ▼ の形で使用する (例参照)。

```
例 DATA EX3 ;
INPUT NAME $ 1-4 ANSWER $ 11-20 ;
IF NAME=▼XXXX▼ THEN NAME=▼ ▼ ;    .....①
IF ANSWER=▼ ▼ THEN DO ;            .....②
    ( ANSWERが欠損値の場合の処理 )
END ;
```

①で、NAME の値を調べ、▼XXXX▼ ならば欠損値としている。②では、ANSWER の値が欠損値か否かのチェックを行い、欠損値ならば特別な処理を施すようにしている。

1.3.3 その他

以上の他、SAS 自身が処理の過程の中で欠損値を産み出す場合がある。例えば、ゼロ割り (zero divide) が起きた場合や、数変数として指定したデータの中に、数字以外の文字があって変換がうまく行かない場合などは、欠損値として処理し、その旨の警告メッセージを出力し、処理は続行される。また、SET や MERGE などに対応する変数の値がない個所は、欠損値とされることは前述した通りである。

実際の解析に際して、各 SAS プロシジャが欠損値を如何に取り扱うかについては、マニュアル [5] の第 6 章あるいは各プロシジャの説明の章を参照されたい。

14. PROC ステップ概説

PROC ステップは、DATA ステップで作成した SAS データセットを、種々の SAS プロシジャを呼び出すことにより解析・処理する過程である。以下に、この PROC ステップ全般にあてはまる事柄について概説する。

各 SAS プロシジャを呼び出す形式は、以下のようである。

PROC プロシジャ名 [オプション] [**DATA**=SAS データセット名] ;

この時、次の条件を満足するような状況では、オプション、データセット名の指定は必要なく、プロシジャ名の指定だけで、SAS が自動的に処理を行ってくれる。その条件は、次の 3 つである。

i) このプロシジャを呼び出す時点で最も近い、それ以前に作成された SAS データセットを処理する場合。そうでない時は、DATA = オプションで処理したい SAS データセット名を指定してやる必要がある。

ii) SAS データセットに含まれるすべての変数についての処理を行う場合。そうでない場合は、この PROC ステートメントに続いて、VAR (VARIABLES の省略形) ステートメントで処理したい変数リストを指定する必要がある。

iii) サブセット毎の処理ではなく、SAS データセットに含まれる全体のオブザベーションについての処理を行う場合。そうでない場合は、この PROC ステートメントに続いて、BY ステートメントでサブセット毎の処理を指示する必要がある。

なお、上に述べた VAR, BY ステートメント以外にも、PROC ステートメントに続いて指定できるステートメントがあるが、そのうち有用な幾つかを挙げる。ただし、これらの指定は、そのプロシジャ内だけしか有効でないことに注意する必要がある。

DROP ステートメント 処理から除外したい変数を指定する。

KEEP ステートメント 処理に組み入れたい変数を指定する。

LABEL ステートメント 解析結果を分かりやすく表示するために、変数にラベルを付加する。

FORMAT ステートメント 解析結果を分かりやすく表示するために、変数の値にラベルを付加する。

TITLE ステートメント 処理結果にタイトルを表示する。

次に、変数リストについて述べておく。変数リストとは、変数名の並びのことで、別に PROC ステップに限らず、DATA ステップにおいても指定可能である。表 14 に、変数リストの種類、指定の仕方、その意味などについてまとめた。ただし、指定例として

```
INPUT  A B $ C $ D E F $ G H $ I ;
```

の場合を考慮した。

表 1 4. 変数リストの指定一覧

指 定 例	意 味	備 考
VAR1-VAR100	変数VAR1からVAR100まで順に100個を指定する。	先頭が英字で、英字列とそれに続く数字列で変数を表わすのを、番号付き変数という。 この番号付き変数の時、最初と最後をハイフン（-）でつないで、その一群を指定できる。
A--G	変数Aから順にGまでを指定する。	番号付き変数でない場合にも、最初と最後を2つのハイフン（--）でつないで、その一群を指定できる。
A-NUMERIC-G	変数AからGまでの間の数変数だけを指定する。 すなわち、A, D, E, Gがあてはまる。	最初と最後の変数の間に単語NUMERICを入れ、それらの間をハイフンでつなぐと、それらの間の数変数だけを指定できる。
A-CHARACTER-G	変数AからGまでの間の文字変数だけを指定する。 すなわち、B, C, Fがあてはまる。	最初と最後の変数の間に単語CHARACTERを入れ、それらの間をハイフンでつなぐと、それらの間の文字変数だけを指定できる。
NUMERIC	SAS データセットに含まれるすべての数変数を指定する。 すなわち、A, D, E, G, Iがあてはまる。	キーワード_NUMERIC_を用いる。
CHARACTER	SAS データセットに含まれるすべての文字変数を指定する。 すなわち、B, C, F, Hがあてはまる。	キーワード_CHARACTER_を用いる。
ALL	SAS データセットに含まれるすべての変数を指定する。	キーワード_ALL_を用いる。

15. インターフェイス

ここでは、他の統計パッケージとのインターフェイスをとるためのCONVERTプロシジャについて述べる。このプロシジャは、Data-text, OSIRIS, BMDP, SPSS等で作成したデータセットを、SAS データセットに変換する機能を持つ。しかし、ここでは、本センターで使用頻度の高いSPSSとのインターフェイスに限って述べる。従来SPSSを使用してデータ解析を行ってきた利用者も、このCONVERTプロシジャでSAS データセットを作成することにより、容易にSASに乗り移れる。以下に形式を示す。

PROC CONVERT

オプション ;

ここで、関連するオプションのみ以下に挙げる。

- i) **SPSS**=DD名 ……入力するSPSSのシステムファイルをDD名で指定する。そのため、そのファイルを前もってそのDD名で割り当てておく必要がある。
- ii) **OUT**=SAS データセット名 ……変換結果を格納する出力SASデータセット名を指定する。そこで、以下の例に示すように、そのSASデータセットを格納するSASデータライブラリを、前もって割り当てておく必要がある。

また、この変換に際して、ある変数を落したり、変数名を変更したりするために、14で述べたKEEP, DROP, RENAME, FORMATステートメントが使用できる。

ただ一つ、このCONVERTで変換できないものがある。それは、変数の値のラベル(value labels)である。これについては、後述するFORMATプロシジャにより、新たに自分で定義するしかない。また、SPSSでは、文字変数の値は4バイトまでなので、SASに変換された後の文字変数も4バイトの長さに設定される。

例

```

alloc f(insps) da(spssds.data) sh .....①
ready
alloc f(outsas) da(sasds.data) ne ca t rel sp(20 20) .....②
ready
sas

17 proc convert spss=insps out=outsas.test ; .....③
27 run ;
37 proc contents data=outsas.test ; .....④
47 run ;

```

①で、入力するSPSSのシステムファイルSPSSDS. DATAを、DD名INSPSSで割り当てた。②で、変換結果を格納するSASデータライブラリSASDS. DATAを、DD名OUTSASで割り当てた。③で、CONVERTプロシジャを呼び出し、変換結果をSASデータセットTESTとして作成した。④では、16.2.1で述べるCONTENTSプロシジャにより、新たに作成したSASデータセットTESTの内容記述部(ドキュメンテーション部)を出力している。このように、SASデータセットの作成や更新のたびに、その内容記述部を出力しておくことは、後々の管理のために大いに役立つ。

16. SAS ユーティリティズ

SAS ユーティリティズに含まれるプロシジャを、以下の2つに大別して概説する。

- i) 一般的なユーティリティズ……SORT, PRINT, FORMAT, EDITOR
- ii) SAS データセット全体に関わる処理のためのユーティリティズ……CONTENTS, COPY, DELETE, DATASETS

このうち、i)のEDITORについては、すでに7で述べたので省略する。また、FORMATについては、次の機会に譲る。

16.1 一般的なユーティリティズ

16.1.1 SORTプロシジャ

SORTプロシジャは、SASデータセットのソーティング、すなわち、オブザベーションの並び換

えによるサブセット化のために用いる。前述したように、MERGE, UPDATE ステートメントを用いる場合、あるいはBYステートメントによりサブセット毎の処理をしたい場合には、このSORT プロシジャで、前もってソーティングしておく必要がある。以下に形式を示す。

```
PROC SORT [DATA=SAS データセット名] [OUT=SAS データセット名];
```

このプロシジャは、必ずその後にBYステートメントを伴い、このステートメントにソートのキーとなる変数のリストを指定する。DATA=, OUT=オプションは、次のように指定する。

i) DATA=SAS データセット名……ソートの対象となるSAS データセット名を指定する。省略すれば、この時点に最も近い、以前に作成されたSAS データセットを指定したことになる。

ii) OUT=SAS データセット名……ソートした結果を格納するSAS データセット名を指定する。省略すれば、ソートの対象とした元のデータセットに出力される。この場合、もう少し詳しく言うと次のようになる。ソートした結果は、新しいデータセットに格納され、それに元のデータセット名が付加されて、その後、元のデータセットは消去される。

文字変数の値については、小さい方から大きい方への値の順序、すなわち文字の昇順 (ascending order) の順序は、次のようである。

```
blank  ¢ .<(+ | &!$*); ¯ - / , % _ > ? : # @ ' = "
abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789
```

また、数変数の値については、昇順に次のようになる。

```
missing values (. _ . . A to .Z)
numeric values
```

なお、降順 (descending order) にソートしたい時には、BYステートメントで、キー変数の前にキーワードDESCENDINGを指定してやればよい (例参照)。

例

```
1? proc sort data=save.circle
2?   out=dessort ;
3?   by descending sex ;
4?   descending weight ;
5?   run ;
6? proc print ;
7?   title data set dessort ;
8?   run ;
```

①で、図35に示したSAS データセットCIRCLEを、まず変数SEXで降順にソートし、その結果のサブセットを更に変数WEIGHTで降順にソートした。その結果を図36に示す。この場合、ソートの対象としたCIRCLEもそのまま保存し

ておきたいので、OUT=オプションを指定し、別の一時的SAS データセットDESSORTに出力している。

DATA SET CIRCLE

OBS	ID	NAME	AGE	SEX	WEIGHT	PULSE	X1	X2	X3
1	310	YAHAMOTO	27	F	62.9	64	36	.	D
2	310	HARA	17	M	49.1	52	92	87	A
3	210	NAGASIMA	19	F	58.5	68	100	96	A
4	110	TOYODA	45	M	90.6	72	73	79	B
5	210	KOHNO	13	F	28.4	64	45	60	D
6	110	YAHAKURA	25	M	50.2	62	69	58	C
7	110	SAITOH	66	M	68.4	68	40	.	D
8	310	EMOTO	24	F	42.5	66	.	100	A
9	110	NAKAHATA	58	M	61.1	60	64	72	C
10	210	KOHATU	19	F	33.8	60	90	68	B
11	210	NAKANISI	60	F	52.7	58	84	80	B
12	310	KAKEFU	42	M	49.2	68	60	.	C
13	310	AMAGUTI	31	F	56.5	50	.	32	D
14	310	KOBAYASI	51	M	58.6	50	81	58	C
15	110	MATUBARA	36	F	38.8	64	43	.	D
16	310	NISHIMOTO	65	M	72.3	64	76	82	B
17	110	HATUOKA	33	F	42.7	64	49	24	D

図 35. SAS データセット CIRCLE の内容

DATA SET DESSORT

OBS	ID	NAME	AGE	SEX	WEIGHT	PULSE	X1	X2	X3
1	110	TOYODA	45	M	90.6	72	73	79	B
2	310	NISHIMOTO	65	M	72.3	64	76	82	B
3	110	SAITOH	66	M	68.4	68	40	.	D
4	110	NAKAHATA	58	M	61.1	60	64	72	C
5	310	KOBAYASI	51	M	58.6	50	81	58	C
6	110	YAHAKURA	25	M	50.2	62	69	58	C
7	310	KAKEFU	42	M	49.2	68	60	.	C
8	310	HARA	17	M	49.1	52	92	87	A
9	310	YAHAMOTO	27	F	62.9	64	36	.	D
10	210	NAGASIMA	19	F	58.5	68	100	96	A
11	310	AMAGUTI	31	F	56.5	50	.	32	D
12	210	NAKANISI	60	F	52.7	58	84	80	B
13	110	HATUOKA	33	F	42.7	64	49	24	D
14	310	EMOTO	24	F	42.5	66	.	100	A
15	110	MATUBARA	36	F	38.8	64	43	.	D
16	210	KOHATU	19	F	33.8	60	90	68	B
17	210	KOHNO	13	F	28.4	64	45	60	D

図 36. 図 35 の CIRCLE を、変数 SEX で降順に、次いで変数 WEIGHT で降順にソーティングした結果の SAS データセット DESSORT の内容

16.1.2 PRINT プロシジャ

PRINT プロシジャは、SAS データセットの内容（データ部の各オブザベーションのデータ値）の出力や、簡単なレポート作成のために用いる。以下に形式を示す。

```
PROC PRINT    [ オプション ] [ DATA = SAS データセット名 ] ;
```

オプションについては、マニュアル [5] を参照されたい。この PRINT プロシジャに続いて指定できるステートメントのうち、有用なものを以下に挙げる。

- i) **BY** 変数リスト；……各 BY グループ毎に分離して出力させる場合に、そのキーとなる変数を指定する。当然、対象となる SAS データセットは、前もってそのキー変数でソーティングされていなければならない（例参照）。
- ii) **VAR** 変数リスト；……出力したい変数およびその順序を、変数リストに指定する。省略すれば、すべての変数の値が出力される。
- iii) **ID** 変数名；……これで指定した変数の値が、各オブザベーションの行の最初に出力される。省略すれば、そこにはオブザベーションの番号が出力される。
- iv) **SUM** 変数リスト；……これで指定した変数の合計が出力される。

例

```
17 proc sort data=save.circle
27   out=prtsort ;          .....①
37   by sex ;
47   run ;
57 proc print ; by sex ;
67   id name ;              .....②
77   var age weight pulse ;
87   sum weight pulse ;
97   run ;
```

①で、図 35 で示した SAS データセット CIRCLE を、変数 SEX でソートし、サブセット化した。②で、BY ステートメントに、その SEX を指定し、サブセット毎の出力をしている。この時、ID ステートメントで、変数 NAME の値を各行の最初に出力し、VAR ステートメントで、変数 AGE, WEIGHT, PULSE のみの値をこの順に出力し、後者 2 つの変数については、SUM ステートメントでその合計も出力している。出力結果を図 37 に示す。

SEX=F				
NAME	AGE	WEIGHT	PULSE	
YAHAMOTO	27	62.9	64	
NAGASIMA	19	58.5	68	
KOHNO	13	28.4	64	
ENOTO	24	42.5	66	
KOHATU	19	33.8	60	
NAKANISI	60	52.7	58	
AWAGUTI	31	56.5	50	
NATUBARA	36	38.8	64	
MATUOKA	33	42.7	64	
-----			---	
SEX		416.8	558	
SEX=M				
NAME	AGE	WEIGHT	PULSE	
HARA	17	49.1	52	
TOYODA	45	90.6	72	
YANAKURA	25	50.2	62	
SAITOH	66	68.4	68	
NAKAHATA	58	61.1	60	
KAKEFU	42	49.2	68	
KOBAYASI	51	58.6	50	
NISIMOTO	65	72.3	64	
-----			---	
SEX		499.5	496	
		-----	----	
		916.3	1054	

図 37. 図 35 の CIRCLE を、変数 SEX のサブセット毎に出力した例

1.6.2 SAS データセット処理のためのユーティリティズ

1.6.2.1 CONTENTS プロシジャ

CONTENTS プロシジャは、SAS データライブラリのディレクトリの出力、各 SAS データセットの内容記述（データセット属性、作成日付、データセットを作成するために用いた SAS ステートメント、変数名、変数の型、長さ、フォーマット、それに付加したラベルなど）や履歴情報の出力のために用いる。SAS データセットの作成や更新のたびに、これらの情報を出力し、ドキュメンテーションとして残しておけば、後々の管理は大いに楽になるであろう。以下に形式を示す。

PROC CONTENTS [オプション] [**DATA**=SAS データセット名] ;

DATA = オプションで、含まれるすべての SAS データセットを指定するのに、キーワード **ALL** が使用できる（例 1 参照）。以下に、有用なオプションの幾つかを挙げる。

- i) **DIRECTORY** SAS データライブラリ（OS データセット）のディレクトリ、すなわち、それに含まれるすべての SAS データセットに関する情報を出力する。
- ii) **HISTORY** SAS データセットの履歴情報を出力する。現在の世代を # 1 とし、それより古い順に順次 # 2, # 3 と名付けられている。通常、5 世代までの情報が格納されている。
- iii) **NODS** DATA = オプションにキーワード **ALL** を指定した時、各 SAS データセットについての内容記述の出力は抑止したい時指定する。含まれる SAS データセットが多い時は、これを指定した方がよい。

iv) **NOSOURCE** …… SAS データセットを作成するために用いた SAS ステートメントの出力を抑止する。

例 1. SAS データライブラリのディレクトリの出力

```
alloc f(save) da(sasdlb.data) sh          .....①
ready
sas
17 proc contents data=save._all_nods ; .....②
27. run ;
```

①で、SAS データライブラリ SASDLI
B. DATA を DD 名 SAVE で割り当てた。
②で、データセット名の指定にキーワード **_ALL_** を指定したが、**NODS** オプシ

ョンも同時に指定して、各 SAS データセットの内容記述は出力しないようにした。こうすると、ディレクトリ出力のみが得られる。出力結果を図 38 に示す。OS データセットの属性や、各 SAS データセット名、それに含まれるオブザベーションの数、それらが占めているトラック数などの情報を知ることができる。②は、オプション **DIRECTORY** を追加して指定しても同じ結果が得られる。

```
CONTENTS PROCEDURE
PHYSICAL CHARACTERISTICS OF OS DATA SET
DSNAME=F0037.SASDLB.DATA UNIT=SYSDA VOL=SER=PUB023 SPACE=(TRK,(2,20))
DISP=OLD DEVICE= 3350 DISK TRACKSIZE= 19069 BYTES
CREATED TUESDAY, APRIL 21, 1981 23 TRACKS ALLOCATED IN 4 EXTENTS
COST IF DATA STORED ONLINE= $0.2760/DAY
COST IF DATA STORED OFFLINE= $0.0920/DAY
SAS DATA SET DIRECTORY
NAME # OBS TRACKS SUBEXTENTS
CIRCLE 17 1 1
CIRCLEIN 17 1 1
HALCIRCL 8 1 1
FEMCIRCL 9 1 1
CIRCLEAD 21 1 1
CIRCLEX 8 1 1
CIRCLEY 7 1 1
ELECTION 8 1 1
CANDIDAT 11 1 1
TRANSCL 5 1 1
MASTCLE 9 1 1
TOTAL TRACKS USED: 12
HIGH TRACK USED: 13
```

図 38. SAS データライブラリのディレクトリ出力の例

例 2. SAS データセットの内容記述、履歴情報の出力

```
37 proc contents data=save.mastcle history ; .....①
47 run ;
```

①で、例 1 と同じ SAS データライブラリに含まれていた SAS データセット **MASTCLE** (図 32 参照) の内容記述を出力している。この結果を図 39 に示す。この時、オプション **HISTORY** を指定したので、履歴情報も出力される。それを図 40 に示す。これを見れば、SAS データセット **MASTCLE** がどのような経過のもとに作成されてきたか一目瞭然であろう。

16.2.2 COPY プロシジャ

COPY プロシジャは、SAS データライブラリの複写のために用いる。SAS データライブラリは、データセット編成が **DA** で、通常の **TSS** の **COPY** コマンドでは複写できないため、このプロシジャを使う必要がある。また、SAS データライブラリをディスクから磁気テープへバックアップするのにも使用できる (例 2 参照)。以下に形式を示す。

PROC COPY IN=DD名1 OUT=DD名2 ;

```

CONTENTS OF SAS DATA SET SAVE.MASTCLE
TRACKS USED=1 SUBEXTENTS=1 OBSERVATIONS=9 CREATED BY JOB F0037#
AT 14:29 MONDAY, OCTOBER 26, 1981 BY SAS RELEASE 79.3B
DSNAME=F0037.SASDLIB.DATA BLKSIZE=2048 LRECL=44 OBSERVATIONS PER TRACK=368
GENERATED BY DATA
ALPHABETIC LIST OF VARIABLES
# VARIABLE TYPE LENGTH POSITION FORMAT INFORMAT
  LABEL
1 NAME CHAR 8 4
2 SEX CHAR 8 12
3 WEIGHT NUM 8 20
4 X1 NUM 8 28
5 X2 NUM 8 36

+----- SOURCE STATEMENTS -----+
|DATA SAVE.MASTCLE ;
| SET SAVE.MASTCL ;
| IF NAME='EHOTO' THEN DELETE ;
| IF NAME='NAKANISI' THEN DELETE ;
| IF NAME='SAITOH' THEN DELETE ;
+-----+

```

図 39. 図 32 に示した SAS データセット MASTCLE の内容記述の出力例

```

.....HISTORY.....
SAS DATA SET #2 IN GENERATION 1 WAS USED IN CREATING DATA SET #1
CREATED BY JOB F0037# AT 14:29 MONDAY, OCTOBER 26, 1981
BY SAS RELEASE 79.3B DSNAME=F0037.SASDLIB.DATA BLKSIZE=2048 LRECL=44
OBSERVATIONS PER TRACK=368 GENERATED BY PROC SORT
.....HISTORY.....
SAS DATA SET #3 IN GENERATION 2 WAS USED IN CREATING DATA SET #2
CREATED BY JOB F0037# AT 14:29 MONDAY, OCTOBER 26, 1981
BY SAS RELEASE 79.3B DSNAME=SYS81299.T142907.SV000.F0037#.R0000003
BLKSIZE=2048 LRECL=44 OBSERVATIONS PER TRACK=368 GENERATED BY DATA

+----- SOURCE STATEMENTS -----+
|DATA ABC ;
| SET SAVE.CIRCLE ;
| IF _N_ GT 12 THEN DELETE ;
| DROP ID AGE PULSE X3 ;
+-----+
.....HISTORY.....
SAS DATA SET #4 IN GENERATION 3 WAS USED IN CREATING DATA SET #3
CREATED BY JOB F0037# AT 14:33 TUESDAY, APRIL 21, 1981
BY SAS RELEASE 79.3B DSNAME=F0037.SASDLIB.DATA INFILE(DSN=F0037.SASDT.DATA
VOL=SER=PU027) BLKSIZE=2048 LRECL=76 OBSERVATIONS PER TRACK=216
GENERATED BY DATA
+----- SOURCE STATEMENTS -----+
|DATA SAVE.CIRCLE ;
| INFILE IN ;
| INPUT ID NAME $ AGE SEX $ WEIGHT PULSE X1 X2 X3 $ ;
+-----+

```

図 40. 図 32 に示した SAS データセット MASTCLE の履歴情報の出力例

IN=オプションでは、複写される入力SASデータライブラリをDD名で指定する。OUT=オプションでは、複写先の出力SASデータライブラリをDD名で指定する。そのため、いずれのデータライブラリも、前もってそれらのDD名で割り当てられている必要がある。以下に、このプロシジャに続いて指定できるステートメントについて述べる。

- i) **SELECT** SAS データセットリスト；……入力SASデータライブラリに含まれる幾つかのSAS データセットを選択して複写する時、そのデータセット名のリストを指定する。この時、前方一致文字列を指定するために、コロン（:）が使用できる。例えば、データセット名としてABC:と指定すると、ABCで始まるすべてのデータセット名（ABC,ABCDE,ABCFYなど）を指定したと同じである。このステートメントを省略すれば、すべてのSAS データセットが複写される。
- ii) **EXCLUDE** SAS データセットリスト；……複写から除外したいSAS データセット名のリストを指定する。この場合も、前方一致文字列の指定のためにコロンが使用できる。
- SELECTとEXCLUDEは、排他的なステートメントなので、同時に指定してはならない。

例 1.

```
alloc f(from) da(sasdlb.data) sh          ..... ①
READY
alloc f(to) da(sasds.data) ne ca t rel sp(20 20) ..... ②
READY
sas
1? proc copy in=from out=to ;              ..... ③
2?     select circle: ;
3?     run ;
4? proc contents data=to._all_ nodes ;      ..... ④
5?     run ;
```

①で、入力側のSASデータライブラリSASDLIB.DATAをDD名FROMで割り当てた。その中には、図38で示した11個のSASデータセットが含まれている。②で、複写先の出力SASデータライブラリをDD名TOで割り当てた。③で、COPYプロシジャを呼び出し複写しているが、この時SELECTステートメントに、前方一致文字列指定の:を用いているため、CIRCLEで始まる名前を持つSASデータセットがすべて選択されて複写される。④で、その出力SASデータライブラリのディレクトリを出力した。その結果を図41に示す。図38と比較すれば、CIRCLEで始まる名前を持つ5個のSASデータセットが複写されたことが分かるであろう。

```
CONTENTS PROCEDURE
PHYSICAL CHARACTERISTICS OF OS DATA SET
DSNAME=F0037.SASDS.DATA UNIT=SYSDA VOL=SER=PUB035 SPACE=(TRK,(20,20))
DISP=NEW DEVICE= 3350 DISK TRACKSIZE= 19069 BYTES
CREATED THURSDAY, OCTOBER 29, 1981 6 TRACKS ALLOCATED IN 1 EXTENTS
COST IF DATA STORED ONLINE= $0.0720/DAY
COST IF DATA STORED OFFLINE= $0.0240/DAY
SAS DATA SET DIRECTORY
NAME          # OBS   TRACKS  SUBEXTENTS
CIRCLE        17      1        1
CIRCLEIN      17      1        1
CIRCLEAD      21      1        1
CIRCLEX       8       1        1
CIRCLEY       7       1        1
TOTAL TRACKS USED: 6
HIGH TRACK USED: 6
```

図 41. 図 38 の SAS データライブラリ SASDLIB. DATA から、複写により作成された SAS データライブラリ SASDS. DATA のディレクトリの出力

例2. ディスク上のSASデータライブラリSASDLIB. DATAを、磁気テープにデータセット名SASBACKUPでバックアップする。

```
// EXEC SAS
//FROM DD DSN=F9999. SASDLIB. DATA, DISP=SHR
//TO DD DSN=SASBACKUP, DISP=(NEW,KEEP),
// LABEL=(1,SL),VOL=SER=ABCDEF,UNIT=OPNMTA
//SYSIN DD *
PROC COPY IN=FROM OUT=TO ;
//
```

16.2.3 DELETE プロシジャ

DELETE プロシジャは、SAS データセットの消去のために用いる。消去された部分は空領域となってしまうが、その後作成されるSAS データセットに対して、なるべくデータライブラリの先頭に近い空領域から埋めることをSASが自動的に行うため、それらを気にする必要はない。このプロシジャのように、SAS データライブラリの中味の変更を伴う処置に対しては、DISP パラメータにOLDと指定しないと、SASはエラーメッセージを出力し処理を中止してしまうので注意が肝要である。以下に形式を示す。

PROC DELETE [DATA=SAS データセットリスト] ;

例

```
alloc f(del) da(sasds.data) old .....①
READY
sas
17 proc delete data=del.circlex del.circley ; .....②
27 run ;
37 proc contents data=del._all_ nodes ; .....③
47 run ;
```

①で、図41に示したSASデータライブラリSASDS. DATAをDD名DELで割り当てた。OLDオペランドを指定していることに注意。②で、その中に含まれるSASデータセットCIRCLEXとCIRCLEYを消去している。③で、消去の確認のためにディレクトリを出力している。その結果を図42に示す。

```
CONTENTS PROCEDURE
PHYSICAL CHARACTERISTICS OF OS DATA SET
DSNAME=F0037.SASDS.DATA UNIT=SYSDA VOL=SER=PUB035 SPACE=(TRK,(6,20))
DISP=OLD DEVICE= 3350 DISK TRACKSIZE= 19069 BYTES
CREATED THURSDAY, OCTOBER 29, 1981 6 TRACKS ALLOCATED IN 1 EXTENTS
COST IF DATA STORED ONLINE= $0.0720/DAY
COST IF DATA STORED OFFLINE= $0.0240/DAY
SAS DATA SET DIRECTORY
NAME # OBS TRACKS SUBEXTENTS
CIRCLE 17 1 1
CIRCLEIN 17 1 1
CIRCLEAD 21 1 1
TOTAL TRACKS USED: 4
HIGH TRACK USED:
```

図42. 図41のSASデータライブラリSASDS. DATAから、SASデータセット2個を消去した例

1.6.2.4 DATASETS プロシジャ

DATASETS プロシジャは、SAS データセットのリスト（一覧）、消去、SAS データセット名の変更（rename）などのために用いる。この場合も、DELETE プロシジャと同様、DISP パラメータに OLD と指定する必要がある。以下に形式を示す。

PROC DATASETS DDNAME =DD名 [オプション] ;

DDNAME = オプションには、処理したい SAS データセットの格納されている SAS データライブラリの DD 名を指定する。そのため、その SAS データライブラリは前もって割り当てられている必要がある。SAS データセット名のリストを出力したくない時は、オプションに NOLIST と指定する。以下に、このプロシジャに続いて指定できるステートメントについて述べる。

- i) **DELETE** SAS データセットリスト ; 消去したい SAS データセットのリストを指定する。DDNAME = オプションに DD 名を指定しているので、1 レベルネームの指定でよい。以下のステートメントも同様。
- ii) **SAVE** SAS データセットリスト ; 保存したい SAS データセットのリストを指定する。このリストに指定されなかった SAS データセットはすべて消去される。
- iii) **CHANGE** 旧 SAS データセット名 = 新データセット名 ; SAS データセット名を変更したい時指定する。
- iv) **EXCHANGE** SAS データセット名 = 他の SAS データセット名 ; DDNAME = オプションで指定した SAS データライブラリ中の SAS データセット間で、データセット名を交換したい時指定する。

その他、周期的に作成されるような多数の SAS データセットについて自動リネームの機能を与える AGE ステートメントがあるが、これについてはマニュアル [5] を参照されたい。

例

```
.alloc f(save) da(sasds.data) old          .....①
READY
sas
1? proc datasets ddname=save ;              .....②
2?   change circlein=circlebc ;
3?   save circlead circlebc ;
4?   run ;
```

①で、図 4.2 に示した SAS データライブラリ SASDS. DATA を DD 名 SAVE で割り当てた。②で、その中に含まれる SAS データセット CIRCLEIN を CIRCLEBC に名前変更をし、CIRCLEAD と CIRCLEBC のみ保存しようとしている。②の実行結果を図 4.3 に示す。変更前後のディレクトリが出力され、処理の内容が一目で分かる。

```

DSNAME=F0037.SASDS.DATA
LIST OF DATA SETS BEFORE UPDATE OF DIRECTORY.
NAME          OBS TRACKS PROT
CIRCLE        17          1
CIRCLEAD      21          1
CIRCLEIN      17          1
LIST OF DATA SETS AFTER UPDATE OF DIRECTORY.
NAME          OBS TRACKS PROT
CIRCLEAD      21          1
CIRCLEBC      17          1
  3 TRACKS USED
  6 TRACKS ALLOCATED
  4 HIGH TRACK USED
  2 TRACKS THAT CAN BE RELEASED FROM OS DATA SET
  1 EXTENTS

```

図43. 図42のSASデータライブラリSASDS. DATAについて、
SASデータセット名の変更、消去を行った例

参考文献

1. 武富, 大賀, 平野, 石田, 景川 統計解析システムSAS概説(1), 九大大型計算機センター広報, 14, 2, 1981, 149-171.
2. 武富, 大賀, 石田, 永井, 景川 統計解析システムSAS概説(2), 九大大型計算機センター広報, 14, 3, 1981, 366-387.
3. 武富, 大賀, 平野, 石田 センターにおける統計解析プログラムパッケージSASへの案内を兼ねて, 九大大型計算機センター広報, 14, 1, 1981, 6-30.
4. 廣田, 竹下, 上田, 喜久村, 志方, 藤井, 蓮尾, 竹下 長期追跡研究用データベースの試作(第3報) SASの利用, 第9回日本行動計量学会予稿集, 1981, 100-101.
5. SAS User's Guide 1979 Edition, SAS Institute Inc.
6. 秦野, 岡部, 津田, 福田, 角三, 芹沢 統計解析システムSAS, 図形表示プログラムSAS/GRAPHの紹介(その1), 名大大型計算機センターニュース, 12, 3, 1981, 310-343.