MEANS、TABULATE、DATASETSプロシジャの機能紹介

SAS Institute Japan株式会社
カスタマーサービス本部
プロフェッショナルサービスグループ
渋谷 佳枝(Yoshie.Shibuya@sas.com)
迫田 奈緒子(Naoko.Sakota@sas.com)
檜皮 孝史(Takafumi.Hiwada@sas.com)

AGENDA

- MEANSプロシジャ
  - ーV8から新たに出力可能な統計量
  - ーAUTONAME/AUTOLABELオプション
  - ーCLASSDATAオプション
- TABULATEプロシジャ
  - ー分類変数に欠損値が含まれる場合
  - ー特定のセルに該当する値が欠損値の場合
- DATASETSプロシジャ
  - ーメンバの削除
  - ーー貫性制約
  - ーDATAステップとのパフォーマンス比較

1. MEANSプロシジャ

1-1.V8から新たに出力可能な統計量

MEANSプロシジャは、数値変数に対する単変量の要約記述統計量を生成します。以下の統計量がVersion8から新たに出力可能になりました。

MEDIAN (中央値)	P1 (1パーセント点)	P5 (5パーセント点)
P10 (10パーセント点)	P90 (90パーセント点)	P95 (95パーセント点)
Q1 (25パーセント点)	Q3 (75パーセント点)	QRANGE (Q1とQ3との差異)

1. MEANSプロシジャ

1-2.AUTONAME/AUTOLABELオプション

AUTONAME/AUTOLABELオプションを使用することにより、出力される統計量が元データの変数名＋統計量、元データのラベル＋統計量と自動的に作成されます。そのため、複数の統計量出力の際に手入力で記述する必要があった作業を軽減化することが可能となります。

基本的な構文

```
PROC MEANS DATA = データセット名 ;
   CLASS クラス変数 ;
   VAR 分析変数 ;
  OUTPUT OUT = 出力データセット名  出力したい統計量=  / AUTONAME AUTOLABEL ;
RUN ;
```

使用する分析変数の数と出力したい統計量の数が少ない場合は、それ程変わらないが、分析変数と出力する統計量が多くなった場合は非常に便利なオプション！

AUTONAME/AUTOLABELにより自動的に作成された変数名･ラベル

合計_Mean	満足度_Mean	合計_Sum	満足度_Sum
621.56862745	70.352941176	20600	3588
406.28571429	72.971428571	14220	2554
924.05405405	81.662162162	68380	6043
912.94117647	69.215686275	93120	7060
1350.3529412	70.917647059	114780	6029
549.61538462	78.173076923	28580	4065
502	71.833333333	15060	2155
390.52631579	71.368421053	7420	1356

1. MEANSプロシジャ

1-3.CLASSDATAオプション

CLASSDATAオプションを使用することにより、分析したい分類変数の組み合わせを含む2次データセットを指定することができます。そのため、元データの分類変数に存在しない値でも、CLASSDATAで指定したデータに存在していれば、MEANSプロシジャの結果に出力されるようになります。

	id	性別
1	1	女性
2	2	女性
3	3	女性
4	4	不明
5	5	女性

「gender」

	sex
1	男性
2	女性
3	不明

「sex」

	性別	kensu
1	女性	4
2	不明	1

	性別	kensu
1	女性	4
2	不明	1
3	男性	0

CLASSDATAにsexを指定

```
PROC MEANS DATA = gender  NWAY NOPRINT  CLASSDATA = sex  ORDER = freq;
  CLASS sex ;
  OUTPUT OUT = result(DROP=_type_ _freq_) N = kensu ;
RUN ;
```

2. TABULATE プロシジャ

欠損値の取扱について

TABULATEプロシジャは、データセットの変数を使用して、記述統計量を表形式で表示します。以下では、TABULATEプロシジャを用い集計表を作成する場合のデータに含まれる欠損値の取り扱いについて説明していきます。

	デフォルトの扱い	表示を変更するには
1. 分類変数に欠損値が含まれる場合	テーブルより除外する	TABULATEステートメント、もしくはCLASSステートメントにてMISSINGオプションを指定
2. 特定のセルに該当するオブザベーションの分析変数がすべて欠損値の場合	(NとNMISS以外の) すべての統計量において欠損値が表示される	TABLE ステートメントにてMISSTEXTオプションを指定
3. 特定の水準のデータが存在しない場合	結果テーブルに水準が表示されない	TABULATEステートメントにてCLASSDATAオプションを指定

2．TABULATE プロシジャ

2-1.MISSINGオプション

分類変数に欠損値が含まれる場合、その値は出力より除外されます。欠損値を一つの水準として集計を行いたい場合はMISSINGオプションを指定することにより、欠損値が一つの水準として扱われ、集計結果に反映されます。

分類変数に欠損値が含まれるデータ

VIEWTABLE: Work.Class

	Name	Sex	Age	Height	Weight
1	Alice	F	.	56.5	84
2	Barbara	F	13	65.3	98
3	Carol	F	14	62.8	102.5
4	Jane	F	12	59.8	84.5
5	Janet	F	15	62.5	112.5

```
PROC TABULATE DATA = class ;
  CLASS age sex / MISSING ;
  VAR height weight ;
  TABLES age * (height * sum weight * sum),sex ;
RUN ;
```

Missingオプションなし

			Sex	
			F	M
Age				
11	Height	Sum	51.30	57.50
	Weight	Sum	50.50	85.00
12	Height	Sum	116.10	101.10
	Weight	Sum	161.50	310.50
13	Height	Sum	65.30	62.50
	Weight	Sum	98.00	84.00
14	Height	Sum	127.10	132.50

Missingオプションあり

			Sex	
			F	M
Age				
.	Height	Sum	56.50	
	Weight	Sum	84.00	
11	Height	Sum	51.30	57.50
	Weight	Sum	50.50	85
12	Height	Sum	116.10	101
	Weight	Sum	161.50	310
13	Height	Sum	65.30	62.50
	Weight	Sum	98.00	84.00
14	Height	Sum	127.10	132.50

欠損値を一つの水準として表示

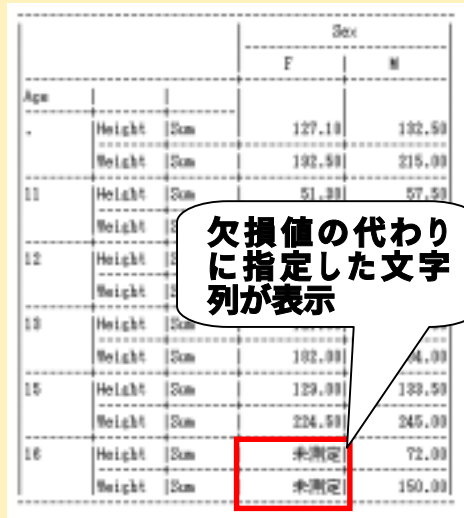
2．TABULATE プロシジャ

2-2.MISSTEXT オプション

特定のセルに該当する分析変数の値がすべて欠損値だった場合や該当するレコードが無かった場合は、すべての統計量（NとNMISS以外）に欠損値が表示されます。結果表示に欠損値以外の値を使用したい場合は、MISSTEXTオプションを使用することにより、指定した値が欠損値の代わりに表示されます。

例）特定のセルに分類されたオブザベーションの身長、体重の値が欠損値だった場合、欠損値の代わりに“未測定”と表示させたい。

```
PROC TABULATE DATA = class ;
  CLASS age sex / MISSING ;
  VAR height weight ;
  TABLES age * (height * sum weight * sum) ,sex
                / MISSTEXT = "未測定" ;
RUN;
```



3．DATASETSプロシジャ

3-1.メンバの削除

DATASETSプロシジャは、SASデータライブラリ中のSASファイルの一覧の作成、名前の変更、コピーや削除などを行う、対話型のプロシジャで柔軟なファイル操作が可能です。以下のプログラム構文では、ライブラリ内のメンバを削除します。

ライブラリ内のメンバを全て削除

```
PROC DATASETS LIB = ライブラリ名 KILL ;
QUIT ;
```

ライブラリ内の指定したメンバ以外の全てを削除

```
PROC DATASETS LIB = ライブラリ名 ;
  SAVE 削除しないメンバ ;
QUIT ;
```

3．DATASETSプロシジャ

3-2.一貫性制約の作成

DATASETSプロシジャのIC CREATEステートメントを使用することで、Version8から一貫性制約を作成することが可能になりました。また、MESSAGE=オプションとの併用でエラー時のメッセージの設定も行うこともできます。一貫性制約を使用することにより、データの矛盾や間違いを未然に防ぐことができ、よりデータの整合性が保たれます。

基本的な構文

```
PROC DATASETS LIB = ライブラリ名 ;
  MODIFY データセット名 ;
  IC CREATE 一貫性制約名称 = 制約のタイプ
      MESSAGE ='制約違反時にログに書き込みエラーメッセージ' ;
QUIT ;
```

一貫性制約はSQLプロシジャ、DATASETSプロシジャ、SCLでのみ生成･追加･削除が可能であり、DATAステップでは取り扱うことはできません。

3．DATASETSプロシジャ

3-3.DATAステップとのパフォーマンス比較

同一データ（件数約370万件）を対象にDATAステップとDATASETSプロシジャを使用してフォーマットを割り当てる処理を行った際のパフォーマンスを比較します。

DATAステップによるフォーマット処理

```
DATA sample ;
  SET sample ;
  FORMAT birthday YYMMDD8.;
RUN ;
```

```
NOTE: DATA ステートメント 処理 :
      処理時間                4:57.16
      CPU 時間                12.00 秒
```

DATASETSプロシジャによるフォーマット処理

```
PROC DATASETS LIB = work NOLIST ;
  MODIFY sample ;
  FORMAT birthday YYMMDD8. ;
QUIT ;
```

```
NOTE: PROCEDURE DATASETS 処理 :
      処理時間                0.03 秒
      CPU 時間                0.02 秒
```

DATAステップはフォーマットを割り当てる際にも1オブザベーションずつデータを読み込んでいくが、DATASETSプロシジャはディスクリプタ部の情報を読み込み、書き換えるだけなので、処理時間が大幅に短縮できます。

まとめ

- Version8への拡張に伴い、プロシジャに対しても様々な拡張が施されました。
- SASを使用して間もないユーザにとっては、今回紹介したプロシジャ・オプションを使用することにより、コーディングの簡略化とパフォーマンス面の向上が望めます。
- Version8の拡張点は、情報配信や分散オブジェクトのサポートなどに注目されがちだが、これまでのSASの言語体系に対しても様々な拡張が施され、依然として強力なものとなっています。

The Power to Know®