関西SASユーザー会

ーSASを使った便利なTIPS＆テクニックー

SAS Institute Japan Ltd.
東　一成（KazunariAzuma）
Kazunari.Azuma@sas.com

The Power to Know.

本日のアジェンダ

- あるユーザーからの質問からサンプルを二つご紹介します。
(BaseSASだけではなくSAS/STATも必要ですが…)
  - SASでランダムサンプリングを行うにはどうすればよいですが？
    - 色々な手法を選んだり、件数や繰り返してサンプリングを行う簡単な方法はありませんか？
  - 主成分分析により、Prin1、Prin2、Prin3、、、と変数が作成されます。Eigenvalueが 1 以上の変数のみをKeepするには？

sas e-Intelligence

The Power to Know.

SASによるサンプリングについて

sas e-Intelligence

The Power to Know.

サンプリングのための方法

- V6の場合・・・
  - RANUNI関数を利用して、乱数を発生させてデータステップなどを利用して抽出する。
  - SQLプロシジャなどを用いて抽出する。
  - 層別抽出の場合はFREQプロシジャなどを組み合わせで実現をする。
  - BaseSASがあればOK。
- V8の場合・・・
  - SURVEYSELECTプロシジャを利用すれば数行で済む。
  - ただしSAS/STATが必要。

色々なサンプリング手法

- 例えば以下のようなサンプリング手法を考えます。
  - 単純無作為抽出
  - 系統抽出法（システムサンプリング）
  - 層別抽出
  - 復元抽出
  - 非復元抽出

  - 上記の抽出方法をいかにして行うかをご紹介させていただきます。

- V6による方法は単純無作為抽出と層別抽出の二つだけにしたいと思います。

それぞれのサンプリング手法について1

- 単純無作為抽出法
  - 単純無作為抽出とは、母集団のどの固体も同じ確率で取り出される抽出法。乱数表やコンピューターに乱数を発生をさせて、サンプルを抽出する場合が多い。
- 系統抽出法（システムサンプリング）
  - 系統抽出法は母集団構成員のリストで標本を等間隔に並んでいる個体を取り出していく方法。500人の母集団から50人のサンプルが必要だとすると、最初の1名は乱数などにより決定される。最初が5番だとするとそれ以降は、15、25、35･･･といった形で抽出されて行く方法。
- 層別抽出法
  - 母集団をいくつかの層に分け、それぞれの層から単純無作為抽出法などによって、サンプルを作成する方法。各層から抽出する場合、母集団の各層の人数に比例させる方法が多い。これは各母集団の内部で、いくつかの均質な層に分かれていると考え、各層別に標本を抽出してサンプルを作成した方が誤差が小さく、均質なサンプルを得ることができるという考え方。

それぞれのサンプリング手法について2

- 復元抽出
  - 母集団から標本を抽出する場合に、どの標本も同じ確率で取り出される状態である必要がある。このためには取り出した標本をもう一度母集団に戻して、次の標本を取り出す必要がある。ただ場合によっては重複されて抽出される標本が出てくる。
- 非復元抽出
  - 上記の復元抽出とは逆で、一度抽出された標本は母集団に戻さないで、次の標本を抽出する。
- 上記の二つの手法であるが、サンプルの標本数に比較して、母集団が十分に大きければ、重複が起こる可能性も低いので、差はほとんど無いと考えられる。

今回利用するデータについて

- 今回のサンプリングの例で利用するデータは以下のような形で、20,000件あります。

OBS	CUST_ID	AGE	GENDER	FLG	Sales	MARRIAGE	HOUSING
1	1	44	女性	0	1500	既婚	学生寮
2	2	44	女性	1	0	未婚	学生寮
3	3	37	女性	0	0	既婚	社宅
4	4	56	女性	0	0	既婚	マンション(賃)
5	5	32	女性	0	3400	既婚	社宅
6	6	62	女性	0	0	既婚	マンション(賃)
7	7	60	女性	0	0	未婚	マンション(賃)
8	8	40	男性	1	0	未婚	社宅
9	9	51	女性	1	0	未婚	社宅
10	10	27	女性	0	0	未婚	学生寮
11	11	44	女性	1	0	未婚	学生寮
12	12	28	不明	0	12000	既婚	マンション(持)
13	13	31	女性	0	0	既婚	社宅
14	14	34	女性	1	0	既婚	マンション(持)

sas e-Intelligence

The Power to Know.

無作為抽出の方法-SASV6の場合 1

```
/*件数を指定する場合-SQLプロシジャ*/
proc sql outobs=2000;
 create table sample1 as
 select ranuni(0) as random, * from ksug.customer order
   by random;
quit;
```

SQLプロシジャを利用して行う方法。この場合はRANUNI関数を利用して乱数を発生させ、乱数の値で上位2000件を取ってきている。
比較的単純に行う事が出きるので便利。

sas e-Intelligence

The Power to Know.

無作為抽出の方法-SASV6の場合 2

```
/*件数を指定する場合-データステップ*/
data sample2 ;
  set ksug.customer nobs = total ;
  if _sample_count_ <  2000 then do ;
    if ranuni(0) *(total + 1 - _N_ ) < = ( 2000 - _sample_count_) then do ;
          _sample_count_ + 1 ;
          output ;
    end ;
  end ;
  drop _sample_count_ ;
run ;
```

データステップを使って行う方法は上記の通り。
乱数を発生させる部分はSQLプロシジャの場合と同じであるが、上記の例では2,000件と指定してあるので、偏ったとり方をしないように工夫が行われている。流れとしては以下のとおりになる。

○例えば･･･20,000件のうち2,000件を抽出したいので、全体の約10%を抽出。
1件目　乱数0.5 * 20000+1 - 1 <= (2000 - 0) →抽出されない
2件目　乱数0.1 * 20000+1 - 2 <= (2000 - 0) →抽出される
3件目　乱数0.9 * 20000+1 - 3 <= (2000 - 1) →抽出されない
上記の処理を2000件になるまで行う。

無作為抽出の方法-SASV6の場合 3

```
/*おおよその割合で抽出する場合*/
data sample3;
   set ksug.customer;
   if ranuni(0)<=0.1;
run;
```

上記の場合も乱数を発生させる方法で同じですが、件数の指定をしていないので記述が簡単。ただ、20,000件のうち2,000件を抽出したいと思っても、数件～数十件程度のブレが見られる。

層別抽出の方法-SAS V6での場合

```
/*性別の[男性][女性]の各層から10%ずつ抽出*/
data sample4 (drop=n1--_cnt2) ;
   set ksug.customer;
   if gender='男性' then do;
      n1+1;
      if _cnt1_ <  820 then do ;
         if ranuni(0) *(8205 - n1 ) < = ( 820 - _cnt1_)  then do ;
        _cnt1_ + 1 ;output ;
         end ;
      end ;
    end ;
    if gender='女性' then do;
      n2+1;
      if _cnt2_ <  807 then do ;
         if ranuni(0) *(8069 - n2 ) < = ( 807 - _cnt2_) then do ;
        _cnt2_ + 1 ;output ;
         end ;
       end ;
    end ;
run;
```

データステップなどを用いて層別抽出を行うには、左の図のようになる。

性別の各層から10%ずつ抽出する処理であるが、まずFREQプロシジャなどで各層の件数を把握し、その内10%の件数を抽出できるようにロジックを組み合わせていく必要がある。

V8以降(SAS/STAT必要)でのサンプリング

- SAS V8からは新たにSURVEYSELECTプロシジャが追加され、様々なサンプリング手法を行う事ができるようになっている。
- 構文は以下のとおり。

```
PROC SURVEYSELECT options ;
        STRATA variables ;
        CONTROL variables ;
        SIZE variable ;
        ID variables ;
```

- 手法は11通り用意されているが、今回は以下の手法を紹介します。
  - 単純無作為抽出
  - システムサンプリング (系統抽出法)
  - 層別抽出
  - 復元抽出
  - 非復元抽出

単純無作為抽出 1

```
/*件数を指定して行う非復元抽出*/
PROC SURVEYSELECT DATA=ksug.customer
    METHOD=SRS
    N=2000
    OUT=sample5;
RUN;QUIT;
```

最も単純なサンプリングを行っている。
手法は「METHOD=SRS」で指定している。SRSはSimple Random Samplingの事。

結果は指定した通りに2000件のデータセットがサンプルとして出力。

単純無作為抽出 2

```
/*件数を指定。復元抽出で重複行を残さない*/
PROC SURVEYSELECT DATA=ksug.customer
     METHOD=URS
     N=2000
     OUT=sample6;
RUN;QUIT;
```

今回のプログラムは、手法が「METHOD=URS」であることが大きな違い。URSはUnrestricted Random Samplingの事。

復元抽出を行っているため、何件かは複数回抽出されている。しかし何度も同じ標本が選ばれると、結果が歪んでしまうことも考えられるので、今回はデフォルト設定のまま複数回選択されたものは1件だけ残して削除する方法を取っている。

何件重複しているかは、新たに追加される変数「NumberHits」を見ると分かる。
よって、このプログラムの結果は何行か削除されているので、2000件と指定したが実際には1917件しか抽出されていない。

単純無作為抽出 3

```
/*件数を指定。復元抽出で重複行を残す*/
PROC SURVEYSELECT DATA=ksug.customer
     METHOD=URS
     N=2000
     OUT=sample7
     OUTHITS;
RUN;QUIT;
```

今回のプログラムも同じように手法が「METHOD=URS」である。ただ今回は「OUTHITS」オプションを付けている。

復元抽出で同じ標本が何度抽出されても、削除はせずにサンプルとして出力する。何件重複しているかは、新たに追加される変数「NumberHits」を見ると分かる。

よって、このプログラムの結果は重複しても削除される事が無いので、指定した2000件が出力されている。

単純無作為抽出 4

```
/*割合を指定。「N=」を「RATE=」に変更*/
PROC SURVEYSELECT DATA=ksug.customer
  METHOD=SRS
  RATE=0.1
  OUT=sample8;
RUN;
QUIT;
```

単純無作為抽出をする時に、そのサンプルサイズを「N=2000」といった方法ではなく、全体の10%といった割合を指定して抽出を行う方法。

「N=」ではなくて、「RATE=0.1」と指定すると10%サンプリングを行える。

系統抽出(等間隔抽出法)

```
/*件数を指定*/
PROC SURVEYSELECT DATA=ksug.customer
   METHOD=SYS
   N=2000
   OUT=sample9;
RUN;
QUIT;
```

系統抽出を行う際には、「METHOD=SYS」と指定を行う。これはSystematic Random Samplingの略となっている。

全件で20,000件のデータから2,000件のサンプルを作るが、等間隔で抽出するので、200000／2000＝10という計算を行い、10件ずつ抽出してくる。初期値はシステム時間などを利用して、決定する。

データがランダムに並んでいる、もしくは北海道～沖縄県といった形で並んでいる場合はこれを反映することができる。ただし、何か特定のルールで並んでいる場合は、初期値によって全く傾向が違ったサンプルが抽出される恐れがあるので注意が必要。

層別抽出法 1

```
/*性別別に1000件ずつ持ってくる場合。
    [男性][女性][不明]なので合計3000名*/
PROC SORT DATA=ksug.customer;
    BY gender;
RUN;
PROC SURVEYSELECT DATA=ksug.customer
    METHOD=SRS N=1000 OUT=sample10;
    STRATA gender;
RUN;QUIT;
```

性別				
GENDER	度数	パーセント	累積度数	累積パーセント
女性	1000	33.33	1000	33.33
男性	1000	33.33	2000	66.67
不明	1000	33.33	3000	100.00

上記のプログラムでは、性別別に1000件ずつ抽出し、サンプルの作成を行っている。
「STRATAステートメント」で変数「GENDER」を指定する事により、変数「GENDER」内の水準を利用して、重み付けを行い同じ件数で抽出を行っている。
出力されたデータには重み付けの結果の変数などが追加されている。

またデータは「STRATAステートメント」で指定する変数でソートを行う、もしくはINDEXを作成しておく必要がある。

層別抽出法 2

```
/*性別別に10% 件ずつ持ってくる場合*/
PROC SORT DATA=ksug.customer;
    BY gender;
RUN;
PROC SURVEYSELECT DATA=ksug.customer
    METHOD=SRS
    RATE=0.1
    OUT=sample11;
    STRATA gender;
RUN;quit;
```

性別				
GENDER	度数	パーセント	累積度数	累積パーセント
女性	807	40.33	807	40.33
男性	821	41.03	1628	81.36
不明	373	18.64	2001	100.00

上記のプログラムでは、性別別に10%件ずつ抽出し、サンプルの作成を行っている。結果的に20,000件のうち2001件のサンプルを抽出している。
この場合も前ページと同じように、出力されたデータには重み付けの結果の変数などが追加されている。またデータは「STRATAステートメント」で指定する変数でソートを行う、もしくはINDEXを作成しておく必要がある。

The Power to Know.

繰り返してサンプリングを行うには？

```
/*100オブザベーションの10のサンプルを作成するには？*/
PROC SURVEYSELECT DATA=ksug.customer
    METHOD=SRS
    N=100
    OUT=sample12
    REP=10;
RUN;QUIT;
```

左記は母集団のデータから無作為に、100オブザベーションの10サンプルを作成する方法となっています。

プログラムとしては普通の単純無作為抽出の場合と同じですが、「REP」オプションに値を設定する事により、その回数分サンプリングを行います。

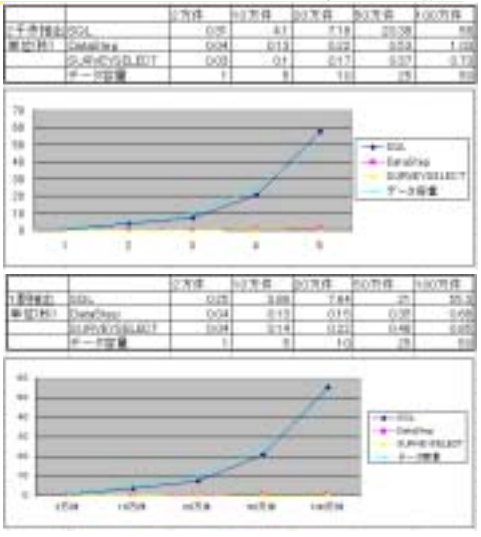
この処理結果は別々のデータセットではなく、一つのデータセットとして出力されるので、変数「Replicate」に繰り返しの回数が保存されます。

The Power to Know.

パフォーマンスの比較

－SQLプロシジャ、Dataステップ、SURVEYSELECTプロシジャでの比較－

- テスト結果に関して
  - テストパターンは2通りを実施した。
  - 1つ目は2万～100万件まで母集団を増やしていき、その中で2000件を抽出する時間をSQLプロシジャ、Dataステップ、SURVEYSELECTプロシジャで比較した。
  - ふたつ目は上記とほぼ同じであるが、サンプルサイズを母集団の10%と指定を行った。
- データステップとSURVEYSELECTプロシジャのスピードが目立つが、記述の簡単さ、手法の多さを考えるとSURVEYSELECTプロシジャは便利に利用する事ができる。



SASによる主成分分析について

頂いた質問の背景

- ユーザーの方はEnterpriseMinerを利用しており回帰分析等の手法を使っている。
- データには数値項目の変数を多く保持しており、回帰分析を行う時に変数間の共線性の問題がある。それぞれの変数を幾つかのまとめて、その結果を利用して回帰分析を行いたいということであった(主成分回帰)。
- その際に主成分分析の結果、主成分スコアの変数で有効なもの(固有値>=1)だけを自動的に残すようなマクロがあると便利なんだけど‥‥。
- ということでサンプルを作成をしました。
- ちなみに今回のデータと質問を頂いた会社とは全く関係ありません。

主成分分析とは？

- 主成分分析とは？
  - 主成分分析とは、幾つかの変数から主成分というお互いに無相関な合成変数を作るための手法。これにより複数の特徴を表す総合的な指標を作成する事が出きる。
- 特徴は？
  - 生成された主成分スコアはお互いに無相関である。
  - 主成分スコアは平均0、分散1に標準化されている。
  - 生成された主成分スコアは複数生成されるが、いくつ利用するかは以下のような判断基準がある。解釈を考えると3～7つが限度。(目安程度とお考えを・・・)
    - 固有値を目安にする方法・・・基準は1以上
    - 累積寄与率を目安とする方法・・・基準は70%～80%程度
    - スクリープロットを目安にする方法・・・固有値をパレート図などを用いて大きい順に並べ、急激に落ち込む手前の主成分。
  - 主成分をいかにしてに解釈するかが重要。(意見は色々とあると思いますが・・・)
  - 他の手法と同様だが、色々なグラフや手法と組み合わせると有効な結果が出てくると考えられる。

今回利用するデータは？

#	変数	ラベル
1	COMPANY	会社名
2	BUNRUI	業種大分類
3	CHUU	業種中分類
4	SISAN_RYUDO	流動資産合計
5	SISAN_TANAOROSHI	棚卸資産合計
6	SISAN_YUKEIKOTEI	有形固定資産合計
7	SISAN_MUKEIKOTEI	無形固定資産合計
8	SISAN_TOSHI_SONOTA	投資・その他資産合計
9	FUSAI_RYUDO	流動負債合計
10	FUSAI_KOTEI	固定負債合計
11	SIHON	資本合計
12	URIAGE	売上高
13	KOKOKU_SENDENHI	広告費・宣伝費
14	HANKANHI	販売費及び一般管理費
15	RIEKI_EIGYO	営業利益
16	EIGYOGAI_SYUEKI	営業外収益合計
17	EIGYOGAI_HIYO	営業外費用合計
18	RIEKI_KEIJO	経常利益
19	TOKUBETSU_RIEKI	特別利益合計
20	TOKUBETSU_SONSHITSU	特別損失合計
21	HAITOKIN	年間支払配当金
22	RIEKI_TOKI	当期利益

- 左図のような変数の構成をしているデータを使って、一番下の当期利益を予測できるようなモデルを作成したい。
- 一般的に当期利益を算出するためのロジックは決まっているため、ある一定の勘定項目が分かっていれば算出できてしまう。また金額といった値だけではなく、「流動比率」「売上高対総資産回転率」等の財務指標や格付けのデータなどがあれば面白い。
- 多くの数値変数があり、かつ微妙に絡み合っているが、業種によっては大きく構成が違ってくる。
- この辺りを主成分分析を利用して解析するサンプルをご紹介し、質問に出てきた主成分回帰を行えるデータセットを作成する。

主成分分析の実行と結果 1

```
proc princomp data=KSUG.ENTERPRISE
   out=OUT_PRIN;
   var SISAN_RYUDO--HAITOKIN;
run;quit;
```

Eigenvalues of the Correlation Matrix

	Eigenvalue	Difference	Proportion	Cumulative
1	6.38819538	5.64881958	0.7985	0.7985
2	2.73937581	1.02188073	0.3424	1.1409
3	1.71749508	1.47015618	0.2147	1.3556
4	1.24733890	0.43503702	0.1559	1.5115
5	0.81230187	0.05298808	0.1015	1.6130
6	0.75931379	0.07078058	0.0949	1.7079
7	0.68853321	0.13188575	0.0861	1.7940
8	0.55664746	0.11588883	0.0696	1.8636
9	0.44075863	0.04194242	0.0551	1.9187
10	0.39881621	0.03884478	0.0499	1.9686
11	0.35997143	0.04488141	0.0450	2.0136
12	0.31509002	0.04241023	0.0394	2.0530
13	0.27267979	0.02318274	0.0341	2.0871
14	0.24949705	0.00563018	0.0312	2.1183
15	0.24386687	0.01758593	0.0305	2.1488
16	0.22628095		0.0283	2.1771

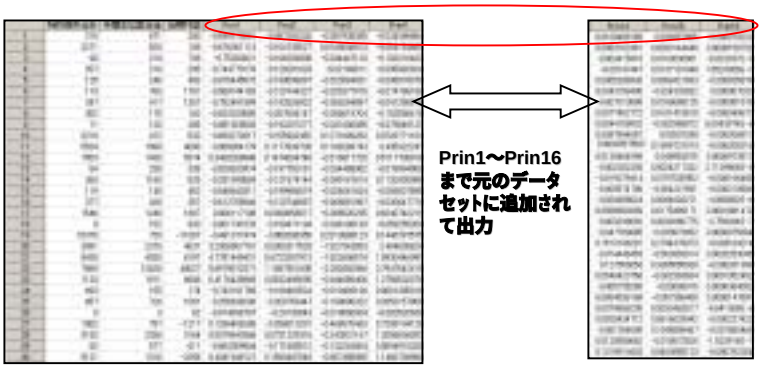
データを利用して主成分分析を行う。
今回は目的変数以外の変数を利用。
出力結果としては、変数の基礎統計量と相関行列、固有値、固有ベクトルなどが出力される。
固有値 (Eigenvalue) の出力結果を表示している。
　先ほどの主成分の残す基準である、固有値が1以上、累積寄与率(Cumulative)が80%と考えると第3主成分、もしくは第4主成分まで利用すればデータの説明はできそう。

「OUT=」オプションで主成分スコアのついたデータを書き出している。

主成分分析の実行と結果 2

- 前ページのプログラムにより、アウトプットとして入力データセットに主成分スコアが付けられたデータセットが作成される。この主成分スコアの変数には、プログラム中「var」オプションで指定された分だけ作成される。
- 今回は16変数追加されている。データ件数が多かったり、もっと変数が多い場合などはもう一度プログラムを作って、前ページのアウトプット画面を見ながら、DROPする変数を決めてデータセットを作り直すのも面倒となってくる。

データセット「OUT_PRIN」



Prin1～Prin16まで元のデータセットに追加されて出力

データ可能のためのマクロ紹介

```
%macro prin_dat1(indata,outdata,input,eigen);
ods trace on;
ods output Eigenvalues=Eigenvalues;
            proc princomp data=&indata out=&outdata;                    ①
                          var &input;
            run;quit;
ods output close;
data Eigenvalues;
   set Eigenvalues;
   if eigenvalue < &eigen then do ;                                     ②
      n+1;
      call symput('drop_prin' || left(put(n,best.)),
                          compress(left('prin'||number)));
      call symput('n_data',n);
   end;
run;
data &outdata;
  set &outdata(drop = %do i=1 %to &n_data; &&drop_prin&i %end;) ;       ③
run;
%mend;
%prin_dat1(KSUG.ENTERPRISE,OUT_PRIN, SISAN_RYUDO--HAITOKIN,1);
```

29

- ここまで説明を行ってきた事を行おうとした場合、有効な主成分を残すためには、出力結果を見ながら、その後に回帰分析時に主成分を選択する必要がある。また不必要な変数もあることからデータ容量が大きくなってしまう。
- そこで指定した固有値以上の変数だけを残すようにする。
- こうするとPRINCOMPプロシジャのOUTオプションに指定されたデーアセットには必要な主成分スコアと元にあった変数だけが出力される。
- マクロ全体の流れとしては、可変となるインプットのデータセット、出力のデータセット、分析変数、固有値の閾値を設定すれば自動的に実行されるようになっている。
- プログラムの①～③に関して説明を加えていく。

① 主成分分析のアウトプットとデータセット化

```
ods trace on;
ods output Eigenvalues=Eigenvalues;
     proc princomp data=&indata out=&outdata;
          var &input;
     run;quit;
ods output close;
```

```
名前 :          Eigenvalues
ラベル :        Eigenvalues of the Correlation Matrix
テンプレート :  Stat.PrinComp.Eigenvalues
データラベル :  Eigenvalues
パス :          Princomp.Eigenvalues
```

	Number	Eigenvalue	Difference	Proportion	Cumulative
1	1	8.2208	5.4429	0.8221	0.8221
2	2	2.7779	1.8271	0.2778	1.0000
3	3	0.9508	0.4678	0.0951	1.0951
4	4	0.4830	0.3952	0.0483	1.1434
5	5	0.0878	0.0252	0.0088	1.1522
6	6	0.0626	0.0178	0.0063	1.1585
7	7	0.0448	0.0119	0.0045	1.1630
8	8	0.0329	0.0074	0.0033	1.1663
9	9	0.0255	0.0041	0.0026	1.1688
10	10	0.0214	0.0063	0.0021	1.1709
11	11	0.0151	0.0049	0.0015	1.1724
12	12	0.0102	0.0042	0.0010	1.1734
13	13	0.0060	0.0021	0.0006	1.1740
14	14	0.0039	0.0008	0.0004	1.1744
15	15	0.0031	0.0010	0.0003	1.1747
16	16	0.0021	0.0000	0.0002	1.1749

PRINCOMPプロシジャなどの統計量を活用する場合は、「OUT」オプション、「OUTSTAT」オプションなどが利用できるが、今回は固有値だけのデータが欲しいのでODSの機能を利用する。

ods trace on;

PRINCOMPプロシジャでどのようなアウトプットが出せて、そのアウトプットをデータセット化するための名前を知る必要がある。その場合には「ods trace on;」と記述する事により、ログにどのような統計量が出力されるか結果が分かる。

ods output Eigenvalues=Eigenvalues;

今回は固有値のデータセットを作成したいので、「ods output Eigenvalues」という部分で、固有値をEigenvaluesというデータセットに出力する指定をしている。

出力されるデータセットを見てみると、主成分の固有値で１未満の変数は「Prin5」以降だということが分かるので、自動的この変数をセットして削除することを行う。

30

② 削除する変数の決定と変数へのセット

```
data _NULL_;
   set Eigenvalues;
   if eigenvalue < &eigen then do ;
      n+1;
      call symput('drop_prin' || left(put(n,best.)),
          compress(left('prin'||number)));
      call symput('n_data',n);
   end;
run;
```

マクロ変数名	値
drop_prin1	Prin5
drop_prin2	Prin6
drop_prin3	Prin7
drop_prin4	Prin8
drop_prin5	Prin9
drop_prin6	Prin10
drop_prin7	Prin11
drop_prin8	Prin12
drop_prin9	Prin13
drop_prin10	Prin14
drop_prin11	Prin15
drop_prin12	Prin16

マクロ変数名	値
n_data	12

前ページのデータセット「 Eigenvalues 」を読込んで、固有値が指定した値以下の物だけマクロ変数に格納をしていき、その件数もセットをしておく。今回は固有値を1にセットしているので第5主成分以下が格納されている。

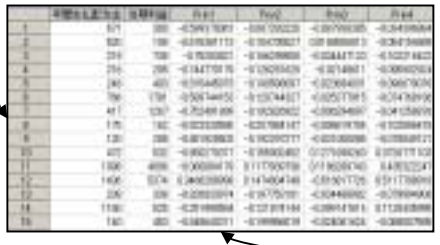
マクロ変数には右の表のような形で格納されている。

③ 余分な主成分スコア変数の削除

```
data &outdata;
   set &outdata(drop =
          %do i=1 %to &n_data;
                &&drop_prin&i
          %end;) ;
run;
```

マクロ変数名	値
n_data	12

マクロ変数名	値
drop_prin1	Prin5
drop_prin2	Prin6
drop_prin3	Prin7
drop_prin4	Prin8
drop_prin5	Prin9
drop_prin6	Prin10
drop_prin7	Prin11
drop_prin8	Prin12
drop_prin9	Prin13
drop_prin10	Prin14
drop_prin11	Prin15
drop_prin12	Prin16



最後にPRINCOMPプロシジャで出力され、まだ多くの主成分スコア変数が残されているデータを読込む。

マクロに格納されている固有値が1未満の変数を呼出して、「SET」ステートメントの「DROP」オプションにセットをしていき、いらない変数を削除する。そして最終的なデータセットが書き出される。

これにより必要な変数だけが残されたデータセットが完成する。

他には何か出来ないか？

- 今回はサンプリングと主成分分析を題材にしたプロシジャの結果のデータ二次加工についてご紹介しました。
- 一般的に企業で使われるデータの分布は非常歪んでおり、このような場合にモデルの妥当性を評価する必要があり、有名な方法としてジャックナイフ法、ブートストラップ法などがある。
- ふたつのテーマを組み合わせる事により、データからリサンプリングを行い、主成分分析を繰り返して行って、結果の安定性やバラツキを知る事ができるような処理も組み込めると考えられる。

sas e-Intelligence

The Power to Know™