2. SAS プログラム

2.1 SAS プログラムの構造とプログラムの基本

SAS のプログラムは，データセット文とプロシージャ文の 2 つから構成されている

- データセット文 (データステップともいう)
  - ✧ データの入力，入力したデータの合成や削除，選択などをおこなう
- プロシージャ文 (プロシージャ (プロック; PROC) ステップ)
  - ✧ 統計的処理 (データの解析) やグラフ･表などの作成をする命令を記述する

```
data d1;
input a b;
cards;
1 2
3 4
5 6
;
run;
proc print;
run;
```

データセット文

プロシージャ文

左は，前回 SAS の基本操作で入力したプログラムである

これの
「data d1;」という行から，一つ目の「run;」
という行までがデータセット文で，
proc print;
run;
というのがプロシージャ文にあたる

それぞれについては，1.2 以降で詳しく紹介する

○ 基本的な留意点

- 半角英数字を用いる。また，SAS は大文字小文字の区別に対応していない
- 日本語など 2 バイト文字にはほとんど対応していないと考えた方がよい
  ディレクトリ (フォルダ) やファイル名，プログラムに至るまで，半角英数字のみを用いるのが賢明
- 単語と単語の間は，必ず半角 1 つ分スペースを空けること
- 行の終わりなどに区切りとして用いられるのは「;」セミコロン (忘れがちなので注意！)

では，次のようなタッチタイピングにおけるトレーニング効果の検証に関する (仮想) 実験データを使って，実際のデータセット文を説明しよう

- 実験参加者を 2 つのグループにわけ，英文を 1 分間に何文字タイピングできるかを測定 (1 回目)
- その後 1 つのグループの人はタッチタイピングのトレーニングを受ける
- 翌日，両方のグループの人に同じタイピングテストをしてもらって測定 (2 回目)

表 1. タッチタイピングにおけるトレーニング効果についての結果

参加者	性別	トレーニングの有無	タイピング文字数	
			1 回目	2 回目
1	男	あり	80	98
2	男	あり	75	100
3	女	あり	80	95
4	女	あり	96	120
5	男	なし	69	74
6	男	なし	100	105
7	女	なし	59	
8	女	なし	90	92

←注 データなし…①

○ SAS のプログラム文では，日本語は使えないので，「男」や「女」，「あり」や「なし」など
例の表中で日本語表記されたデータは，あらためて半角英数字表記にする必要がある

○ この例では，
- 性別は，男性を m(male)，女性を f (female)
- トレーニングの有無は，ありを y (yes)，なしを n (no) とコーディングしている

！注意！ データに名前をつける時は，後から名前を見ただけで内容がわかるようなものにしよう

```
1 m y 80 98
2 m y 75 100
3 f y 80 95
4 f y 96 120
5 m n 69 74
6 m n 100 105
7 f n 59 .
8 f n 90 92
```

上のデータを SAS で使えるように入力すると，左のようになる

- SAS では，1人分のデータのまとまりを**オブザベーション**とよぶ
- 1つのオブザベーションの中に，複数の種類のデータがある
- SAS では，1オブザベーションにつき1行を使用するのが原則
- 1オブザベーション内のデータは，半角スペース1つ分空けて入力する
- データの並びは，オブザベーション間で共通にする
  (左の場合では，1列目が被験者，2列目が性別……) というようになっている
- 欠損値がある場合は「.」(半角ピリオド) を入力する

2.2 データセット文

データセット文の構造は次のようになっている．この構造は，どのようなデータを入力する場合でも基本的には変わらない

```
data (データセット名);
input (変数リスト);          ←インプット文
<オプション>;
cards;
(データの入力)               ←データ入力開始

;                            ←データの終了のマーク……②
run;
```

図1．データセット文の構造

○ データセット名とは……SAS 上でのデータセット文全体の名前
アルファベットで始まる半角英数字で，5 文字以内が望ましい (アンダーバーは使用可) が，適当につけてしまってもあまり問題はない．複数のデータセットを同じプログラム内で使い分けるときに識別のために用いるが，単一のデータセットしか用いない場合は特にその後必要になることはない

○ インプット文とは……変数がいくつかあるかと，その種類を指定し，読み込ませる文
入力するデータの並びと，それが何であるかを読み込ませるために必要
インプット文の変数リストには，SAS で解析したいデータの変数名を，入力した順に記述する．つまり入力した変数の数だけ変数名を入力することになる
表1のデータでは，被験者，性別，トレーニングの有無，(タイピング文字数) 1回目，2回目，を変数リストに含む必要がある

○ インプット文における変数名入力の規則
- 変数名も，アルファベットで始まる半角英数字で入力する
- 変数名は内容がわかるようにする方が望ましく，また半角8文字以下にしておく方がよい．変数と変数の間は，半角スペース 1 文字分を入れておくこと

- 演算をおこなわないデータのことを文字型データとよぶ (これに対して，計算可能なデータを「数値型データ」とよぶ) の場合は，変数名のあとに，半角で一文字分スペースを空けた後に「$ (半角のドルマーク)」を挿入して文字型データであることを宣言する必要がある

表1のデータでは実験参加者，性別，トレーニングの有無が文字型データで，(タイピング文字数) 1回目，2回目は数値型データであるから，

```
data d1;
input sub $ sex $ train $ before after;
cards;
```

実験参加者を sub，性別を sex，トレーニングの有無を train，1回目を before，2回目を after とするとデータセット文の最初の3行は次のように書くことになる

図2．プログラム例文

○ 実験参加者 (sub) は1，2，3，4……と数字がついているが，これは参加者相互を識別するための番号で，その数字を足したり引いたりできるものではない．そこで，ここでは実験参加者という変数は文字型データとして扱い，sub の後ろに「$ (ドルマーク)」を入力している (ただし，演算をおこなうことが無意味だと本人が理解できていれば，数値型にしても特に問題は生じない)

○ インプット文の意味
インプット文に「input a b c;」と入力されていたら，データ入力欄での数字や文字の並びが左から，a，b，c，に対応しているということを示している

○ データの入力
- インプット文で変数名の指定をしたら，次はデータの入力！
- インプット文の一行下の「cards;」が，これはデータの開始を示すコマンドである
- インプット文で指定した順でデータを入力し，1オブザベーション (1人分) ごとに改行する
- 全データの最後に「; (半角セミコロン)」を入力し，全データの終了点を示す (②のこと)
- その後ろに「run;」を入力する (実行命令のコマンド)

今回の例では，インプット文で「input sub $ sex $ train $ before after;」と入力したので
「1 m y 80 98」で1セットのオブザベーションとなり，左から順に実験参加者，性別，トレーニングの有無，1回目，2回目の点数をあらわす数値である
※注 m は男 (male)，y はトレーニング有り (yes) を示している

○ データ入力での注意点：欠損値について
実際のデータには，表1「注～①」のようにデータのない場合 (＝欠損値) がある
欠損値があった場合には，その部分に「. (半角ピリオド)」を入力する
- 「.」を入力しなければ，次に書いてあるデータが (欠損していた) 変数のデータと見なされ，以降すべてがずれて読み込まれてしまうので注意すること

プログラム2

```
data d1;
input sub $ sex $ train $ before after;
cards;
1 m y 80 98
2 m y 75 100
3 f y 80 95
4 f y 96 120
5 m n 69 74
6 m n 100 105
7 f n 59 .
8 f n 90 92
;
run;
```

表1のデータでデータセット文を書いたものが左となる

！確認！

- m／f は male/female で男女を示している
- y／n はトレーニングの有無を yes/no で示している

- 実験参加者7は，2回目のデータが欠損しているので，「.」を挿入している

2.3 データセット文のオプション

図1の＜オプション＞のところでは，インプット文で読み込んだ変数を，算術演算子，関数，条件文などで加工して (つまり，四則演算や条件わけなどをして)，新たな変数を作り出すことができる
例えば，タイピング文字数の1回目と2回目の差を取りたいとき，あらかじめ自分で差を計算しておいて，データを入力することもできるが，オプションを使用すれば，その作業を SAS にさせることができる

○ オプション入力の規則
オプションを使いたい時には，プログラムの＜オプション＞とある部分で
・(新しい変数名) ＝ 演算式 or 関数名 ；(半角セミコロン)
の順で入力する (セミコロンを忘れないこと！)

1回目と2回目との差 (変数名は diff としている) を示す新しい変数を作りたいときには
「diff = after – before」
となる．これをプログラム2に付け加えたのがプログラムである

オプションでは，変数間の四則演算をおこなうだけでなく，さまざまな演算が可能である

オプションで用いることができるのは，大きく分けて
① 算術演算子
② 関数 (算術･統計･丸め･数学)
③ 条件文 (if～then 文)

である．一番よく使うのは，変数の四則演算をおこなう算術演算子である
条件文については，分析のためのプロシージャ文を学習する段階で改めて解説する

プログラム3

```
data d1;
input sub $ sex $ train $ before after;
diff = after – before ;
cards;
1 m y 80 98
2 m y 75 100
3 f y 80 95
4 f y 96 120
5 m n 69 74
6 m n 100 105
7 f n 59 .
8 f n 90 92
;
run;
```

2.4 算術演算子

(インプット文で，a，bという変数が読み込まれ，それらから d という新しい変数を作る場合)

- 和「+」 d=a+b;
- 差「- (マイナス)」 d=a-b;
- 割り算「/ (斜め線)」 d=a/b;
- 掛け算「* (アスタリスク)」 d=a*b;
- 累乗「 ** n (アスタリスク2つ＋n 乗)」 d=a ** 2; (これは2乗の場合，3乗なら d=a ** 3;

演算式の場合は，変数と演算子の間にスペースを入れてもかまわない (入れなくてもよい)
これらを組み合わせることも可能．また数式のように「( ) 括弧」も有効である
例： d = (a – b) / c; といった演算も可能

2.5 その他の演算子

比較演算子

＝ (EQ) 等しい(EQUAL)	^＝ (NE) 等しくない(NOT EQUAL)
＞ (GT) より大きい (GREATER THAN)	＞＝ (GE) 以上(GREATER or EQUAL)
＜ (NG) より大きくない(NOT GREATER)	＜(LT) より小さい(LESS THAN)
＜＝(LE) 以下(LESS or EQUAL)	^＜(NL) より小さくない(NOT LESS THAN)

論理演算子

& (AND) かつ ! (OR) または ^(NOT) 否定

これらの演算子は，条件文と組み合わせて新しい変数を作る場合に使うことが多いので，具体的な使用法は後述する

2.6 関数

SAS はさまざまな関数ももっているので，Excel などと同様の感覚で利用することができる
先ほど作った新しい変数 diff と同様に，Input 文の直後に記述することによって，これらの関数を用いて別の変数を作成することができる

算術関数

ABS(x)	x の絶対値		
MAX(x,y,…)	x,y,…の中の最大値	MIN(x,y,…)	x,y,…の中の最小値
MOD(x,y)	x を y で割ったときの剰余	SQRT(x)	x の平方根

丸め関数

INT(x)	x の整数部分を取り出す	ROUND(x,y)	x を y の単位で四捨五入

数字関数

LOG(x)	自然対数 (底が e の x の対数)
LOG2(x)	2 を底とする x の対数
LOG10(x)	10 を底とする x の対数

三角関数

COS(x), SIN(x), TAN(x)	x のサイン・コサイン・タンジェント
ARSIN(x)	x の逆正弦変換 (角変換)

たとえば変数 x の自然対数をとり，その値を新しい変数 logx として定義したいときは，input 文の直後に，

```
logx = log(x);
```

と記述すればよい

演算子や関数については，例えばこのページが詳しい
http://www.kudpc.kyoto-u.ac.jp/Service/Application/SAS/text/func.html

2.7 プロシージャ文

プロシージャ文は，入力したデータに対して，どのような解析をおこないたいかを指定するものである
解析の種類によっては，別の文と組み合わせて使うこともある

```
proc (プロシージャ名);
run;
```

プロシージャ文の基本構造は左のようになる
もっとも簡単なプロシージャは print で，入力したデータセットの中身を表示する命令である．データが正しく読み込まれているかどうかの確認によく用いる

```
proc print;
run;
```

print 文は，左のようになる．このプロシージャで「読み込まれたデータセットのすべて表示する」という命令 (解析) が実行される

上記で紹介したプログラム3に proc print をつけて完成させると右 (プログラム4) のようになる

では，実際に右のプログラムを動かしてみよう

プログラムを入力すると時間がかかるので，講義 Wiki の本日付情報からリンクされている SAS プログラムを，各自 Web ブラウザで読み込んで利用するとよい

プログラム4

```
data d1;
input sub $ sex $ train $ before after;
diff = after - before ;
cards;
1 m y 80 98
2 m y 75 100
3 f y 80 95
4 f y 96 120
5 m n 69 74
6 m n 100 105
7 f n 59 .
8 f n 90 92
;
run;

proc print;
run;
```

● プログラムの利用方法
index.html にプログラム4が表示されているので，プログラム部分をドラッグして選択し，メニュー→編集→コピーor 右クリックでコピーする．次に g-edit などのテキストエディターを起動して，新しいファイルとして貼り付ける．名前はなんでもかまわないが，内容のわかるようなもの (例 test.sas) にして，拡張子を「. sas」にして保存すること

○ 保存したら，program editor のメニューバー「ファイル」＞「開く」で，そのファイルを選択する
○ program editor にプログラム4と同じものが読み込まれるはずである

○ さあ，実行 (サブミット) してみよう！
SAS で作成したプログラムを実行 (サブミット) すると，output ウィンドウに入力したデータの一覧表が表示される．欠損値などが正しく入力されてない場合は，ずれて表示されてしまう

実行して，図8のようになれば，プログラムは正しく実行されているということになる

OBS	SUB	SEX	TRAIN	BEFORE	AFTER	DIFF
1	1	m	y	80	98	18
2	2	m	y	75	100	25
3	3	f	y	80	95	15
4	4	f	y	96	120	24
5	5	m	n	69	74	5
6	6	m	n	100	105	5
7	7	f	n	59	.	.
8	8	f	n	90	92	2

図3．プログラム4の実行結果

```
proc print;
var sex train diff;
run;
```

もし，一部の変数のみについてデータの一覧を表示したい時は，print プロシージャに新たな命令（ステートメント）を付け加えればよい．もし，性別，トレーニング，得点差だけを一覧に表示したいときは，var（変数：variable の意味）ステートメント（2行目）を使用して，左のように記述する

●うまく実行できない！

プログラムのどこかにミスがあって，うまく実行されていない場合は，OUTPUT ウィンドウに結果が表示されなかったり，表示されていたとしても一部だったりする．このようなときは log ウィンドウを表示させてみるとよい．なんらかのエラーメッセージが出ているはずである．それをよく読んで，プログラムのどこがおかしいのか，その原因を自分自身で突き止め，修正しなければならない（慣れるまで，これが結構大変．また，ピントはずれのエラーメッセージが出ていることもままある）

課題 1

人間科学部生 10 名を対象として，ある質問紙調査を実施した．調査項目と学生の回答状況を講義 Wiki の本日付情報からリンクしてある．この質問紙調査データを SAS で分析したい．適切なデータセット文を作成し，データセットの中身すべてを表示するプロシージャを実行できるようなプログラムを作成せよ

講義 Wiki: http://www.team1mile.com/asarin/hus/ip-hus/index.html