

# 複雑なデータを解析しやすくする ためのSASによるデータ加工

## 繰り返しを利用したプログラム

(株)ACRONET 林 行和

# 高田さんの発表を早速実践

「そうかぁ、**ARRAY**ってああやって使うんだね。  
便利だとは聞いてはいたけど。  
早速、使ってみよう！」

・・・プログラム中・・・

「よし！できた！実行！！」

「...あれえ～～？？？」

# 高田さんの発表を早速断念

「やっぱりARRAYは難しいなあ。  
今までどおり、IF文、SET、MERGEで  
やればいいか。。。」

・・・ARRAYも頑張って覚えて欲しいけど・・・

だったら、そのプログラム、せめてミスを防ぐ  
技を使ってみませんか？

# 本日の内容

- 本日表示プログラムの目的(高田さんと同じ)
- 単純に組むと
  - ミスを生みやすい冗長的なプログラム例
- マクロを使ってみよう
  - 時点部分をマクロに
  - 項目部分もマクロに

# 本日示すプログラムの目的(高田さんと同じ)

タイプ1:1症例1オブザベーション

ID	TC1	HDL1	TC2	HDL2	TC3	HDL3
1	212	50	224	64	204	73
2	206	58	208	63	212	58
3	221	47	236	70	242	38
4	251	60	239	37	236	42



加工

タイプ4:症例ごと項目ごと時点ごとの繰り返し

ID	LBCD	LB	VISIT	VAL
1	1	TC	1	212
1	1	TC	2	224
1	1	TC	3	204
1	2	HDL	1	50
1	2	HDL	2	73
1	2	HDL	3	73
2	1	TC	1	206
2	1	TC	2	208
2	1	TC	3	212
2	2	HDL	1	58
2	2	HDL	2	58
2	2	HDL	3	58
3	1	TC	1	221
3	1	TC	2	236

- 測定値が横並び
- 検査項目名と時点が併合された変数名

- 測定値が縦並び
- 検査項目名(LB)と時点(VISIT)が別々

# 単純に考えると...

タイプ1:1症例1オブザベーション

ID	TC1	HDL1	TC2	HDL2	TC3	HDL3
1	212	50	224	64	204	73
2	206	58	208	63	212	58
3	221	47	236	70	242	38
4	251	60	239	37	236	42

タイプ1のTCの1時点目をタイプ4の形にすると

加工

タイプ4:症例ごと項目ごと時点ごとの繰り返し

ID	LBCD	LB	VISIT	VAL
1	1	TC	1	212
1	1	TC	2	224
1	1	TC	3	204
1	2	HDL	1	50
1	2	HDL	2	64
1	2	HDL	3	73
2	1	TC	1	206
2	1	TC	2	208
2	1	TC	3	212
2	2	HDL	1	58
2	2	HDL	2	63
2	2	HDL	3	58
3	1	TC	1	221
3	1	TC	2	236

```
data TC1; set TEMP1;
  VAL=TC1;
  LB= " TC " ;
  LBCD=1;
  VISIT=1;
  keep ID LBCD LB VISIT VAL;
run;
```

## 単純に考えると...

```
data TC1; set TEMP1;  
  VAL=TC1;  
  LB= " TC " ;  
  LBCD=1;  
  VISIT=1;  
  keep ID LBCD LB VISIT VAL;  
run;
```



```
data WRK;  
  set TC1 TC2 TC3 HDL1 HDL2 HDL3;  
run;  
  
proc sort data=WRK;  
  by ID LBCD VISIT;  
run;
```

同様に  
TC2, TC3, HDL1, HDL2, HDL3  
を作成し、

単純な繰り返しはミスの元!!

# 時点部分をマクロ化しよう！

時点に関わるのは、この部分

```
data TC1; set TEMP1;  
  VAL=TC1;  
  LB= " TC " ;  
  LBCD=1;  
  VISIT=1;  
  keep ID LBCD LB VISIT VAL;  
run;
```

TC1の構造

ID	LBCD	LB	VISIT	VAL
1	1	TC	1	212
2	1	TC	1	206
3	1	TC	1	221
4	1	TC	1	251

これが1～3まで繰り返される。



# 時点部分をマクロ化しよう！

```
%macro loopTC;
```

```
  %do i=1 %to 3;
```

```
    data TC&i.; set TEMP1;
```

```
      VAL=TC&i.;
```

```
      LB= " TC " ;
```

```
      LBCD=1;
```

```
      VISIT=&i.;
```

```
      keep ID LBCD LB VISIT VAL;
```

```
    run;
```

```
  %end;
```

```
%mend loopTC;
```

```
%loopTC;
```

添え字iを1から3まで回します

# 時点部分をマクロ化しよう！

時点の繰り返しは、マクロ化できました。  
一つのプログラムで3時点分のデータができます。

でも、これだと、%loopHDLというマクロも必要になりますね。

もう一度、プログラムを見てみましょう。

# 項目部分もマクロ化しよう！

項目に関わるのは、この部分

```
data TC1; set TEMP1;  
  VAL=TC1;  
  LB= "TC";  
  LBCD=1;  
  VISIT=1;  
  keep ID LBCD LB VISIT VAL;  
run;
```

# 項目部分もマクロ化しよう！

```
%macro loop(var, varnum);  
  data &var.1; set TEMP1;  
    VAL=&var.1;  
    LB= " &var. ";  
    LBCD=&varnum.;  
    VISIT=1;  
  keep ID LBCD LB VISIT VAL;  
run;  
%mend loop;  
%loop(TC, 1);  
%loop(HDL, 2);
```

文字型の変数にマクロ引数を代入する時は、  
ではなく、  
” ” じゃないとダメ！！

このプログラムで  
TC1, HDL1  
が出来上がり！

# 時点部分と項目部分を合わせると...

```
%macro loop(var, varnum);  
  %do i=1 %to 3;  
    data &var.&i.; set TEMP1;  
      VAL=&var.&i.;  
      LB= "&var. ";  
      LBCD=&varnum.;  
      VISIT=&i.;  
      keep ID LBCD LB VISIT VAL;  
    run;  
  %end;  
%mend loop;  
%loop(TC, 1);  
%loop(HDL, 2);
```

# 時点部分と項目部分を合わせると...

このプログラムでデータセット

TC1 TC2 TC3 HDL1 HDL2 HDL3

が出来上がりました。

```
data WRK; set TC1 TC2 TC3 HDL1 HDL2 HDL3;  
run;  
proc sort data=WRK;  
  by ID LBCD VISIT;  
run;
```

これで完成！

でも...

**臨検項目ってもっとたくさんありませんか？**

# もっと手間を減らそう！

```
data WRK;  
  delete;  
run;  
%macro loop(var, varnum);  
  %do i=1 %to 3;  
    data WK1; set TEMP1;  
    (中略)  
    run;  
    data WRK; set WRK WK1;  
    run;  
  %end;  
%mend loop;  
%loop(TC, 1);  
%loop(HDL, 2);
```

マクロ前に空のデータ  
セットを用意

マクロが一度回る毎に出  
来上がったWK1の内容が  
WRKにストックされていく

# イメージとしてはこんな感じに...

ID	LBCD	LB	VISIT	VAL	
1	1	TC	1	212	WRK1
2	1	TC	1	206	
3	1	TC	1	221	
4	1	TC	1	251	
1	1	TC	2	224	
2	1	TC	2	208	
3	1	TC	2	236	
4	1	TC	2	239	
1	1	TC	3	204	WK1
2	1	TC	3	212	
3	1	TC	3	242	
4	1	TC	3	236	



# もっと手間を減らそう！(MERGE編)

```
data WRK2;  
  ID = .;  
  delete;  
run;  
%macro loop;  
  data WK1; set WRK; (中略); run;  
  proc sort data=WK1; by ID; run;  
  data WRK2; merge WRK2 WK1;  
    by ID;  
  run;  
%mend loop;
```

あらかじめmerge keyになる変数を同じ属性で作っておく必要あり!!

# イメージとしては... (MERGE編)

ID	TC1	TC2	TC3
1	212	212	212
2	206	206	206
3	221	221	221
4	251	251	251

WRK2

WK1

# マクロを使いこなせるとこんなに便利

## マクロ使用の事例紹介

いくつか紹介できるので、ここで分量の調整をします。