

Proc luaを初めて使ってみた
-SASでの処理を条件に応じて変える-

浅井 友紀
(エイツーヘルスケア株式会社)

I tried PROC LUA for the first time

Tomoki Asai
A2 Healthcare Corporation

要旨：

実行されるコードを分岐、繰り返すためにはSASマクロが用いられてきた。

本発表ではSASマクロではなくProc Luaを用い、実行されるコードの分岐や繰り返しをするプログラムを紹介する

キーワード：proc lua ,SASマクロ

Luaとは

- 実行速度が速く、C言語に組み込みやすいスクリプト言語
- LuaからC言語の呼び出し、C言語からLuaの呼び出しが容易にできる
- ポルトガル語で月を意味する
- 最近ではゲームの開発に使用されていることもある

言語自体が軽量のため、IoTの際に使用されることも
(無線LAN対応の某SDカードではLuaの動作が保証されている)

Luaの基本構文

```
proc lua;
```

```
  submit;
```

```
  --luaの変数を作成する
```

```
  local test = "変数を"
```

```
  local test2 = "作るよ"
```

```
  --作成した変数を表示
```

```
  print( "test=", test )
```

```
  --SASの関数はsas.xxの形式で使うことが出来る
```

```
  print( sas.catx("",test , test2) )
```

```
  --ここにSASコードを書く
```

```
  sas.submit([[
```

```
    data test;
```

```
      a = 1; output;
```

```
    run;
```

```
  ]])
```

```
  endsubmit;
```

```
run;
```

コメントは一行の場合は「 --コメント 」、
複数行の場合は「 --[[コメント]] 」

Luaのsubmitからendsubmitまでの間は、
文末にセミコロンが不要

Luaによる実行コードの分岐の例

```
proc lua ;
  submit;
  --データセットの有無を取得(有=1,無し=0)
  local ds = sas.exist("DAT")

  --データセットのオブザベーション数を取得
  local ds = sas.open("DAT2")      --""で囲むとそのデータセット名を指す
  local obs = sas.nobs(ds)         --""で囲まないとluaのテーブルを指す
  sas.close(ds)                    --sas.openで開いたデータセットは必ず閉じる

  if ds == 0 then
    print( "データセットがありません" )
  elseif obs == 0 then
    print( "データセットDAT2のレコードがありません" )
  elseif sas.submit([[ data DAT3; set DAT DAT2; run; ]])
  end;
endsubmit;
run;
```

Luaによる繰り返し処理の例

```
proc lua ;  
  submit;  
  
  --for [ループ変数] = [初期値], [終わり値], [増え幅]  
  --増え幅を省略すると1つつ増える  
  for i = 1 , 5 do;  
    print("今", i, "です")  
  
    if i == 4 then  
      break  
    end;  
  end;  
  
  print("BREAK文によりループを抜けました")  
  
  endsubmit;  
run;
```

今回作成したプログラムの概要

- Excelの情報をSASに読みこむ
- Excelから読みこんだ情報に基づいて、SASのコードを分岐させる
- Excelのシート(今回は2シート)分プログラムを繰り返し実行する

以下のようなExcelを読みこんで、Name2の値を変数名に、Labelの値を変数ラベルとする。Excel内の行数分の変数を持ち、0obsのデータセットを作成するプログラム

Excelの例

Name1	Name2	Label	Type	Format	r k
A	ID	ID	text		
A	A1	label 1	text		
A	A2	label 2	text		
A	A3	label 3	text		
A	A4	label 4	num	YYMMDD10.	
A	A4_1	label 41	num		
A	A5	label 5	text		
A	A6	label 6	text		
A	A7	label 7	num		
A	A8	label 8	text		

プログラム

```
proc lua terminate;
  submit ;
  --読み込み対象のexcelシート名をテーブルに
  dataset = {}
  dataset[1] = "Sheet3"
  dataset[2] = "Sheet1"

  for i, tag in ipairs(dataset) do      --テーブルを要素番号順にtagに格納する
    sas.submit([[
      filename xls dde "EXCEL|@tag@!R2C2:R500C5" ;
      data @tag@DEF ;
      infile xls dsd notab dlm='09'x missover lrecl=32767 ;
      length _NAME _LABEL _TYPE _FORMAT _LENGTH $80;
      input  _NAME $ _LABEL $ _TYPE $ _FORMAT $ ;

      if ( _NAME="" ) then delete ;
      if _TYPE = "num" then _LENGTH = "8.";
      if _TYPE = "text" then _LENGTH = "$200";
      run ;
      filename xls clear;      ]])
```



```

sas.submit([[
  data @tag@DEF10 ;
    set @tag@DEF end=_EOF ;
    ATT_LABEL = cats("label= " , _LABEL , "");
    if _TYPE = "text" then ATT_LENGTH = cats("length= ",_LENGTH);
    if _TYPE = "num" then ATT_LENGTH = cats("length= ",_LENGTH);
    if _TYPE = "num" and _FORMAT ^= "" then ATT_FORMAT = cats("format= ",_FORMAT);
run ;

```

*--変数の定義確認;

*----- 重複チェック 重複する変数名のみを残す;

```

proc freq data = @tag@DEF10 noprint;
  tables _NAME / out = @tag@_VAL_DUP( where = (COUNT > 1) drop = percent ) ;
run;

```

*----- lengthチェック 変数名が8文字以上、ラベルが40字以上のものを残す;

```

data @tag@_VAL_LENGTH ;
  set @tag@DEF10;
  CHLENG_NAME = length(_NAME);
  CHLENG_LABEL = length(_LABEL);
  if (CHLENG_NAME > 8) or (CHLENG_LABEL > 40) ;
run;
]])

```

--重複チェック

```
local _valdup = dataset[i].." _VAL_DUP"  
local dsdup = sas.open(_valdup)  
local dupobs = sas.nobs(dsdup)  
sas.close(dsdup)
```

--lengthチェック

```
local _vallen = dataset[i].." _VAL_LENGTH"  
local dslen = sas.open(_vallen)  
local lenobs = sas.nobs(dslen)  
sas.close(dslen)
```

--処理対象のデータがあるかのチェック

```
local _valds = dataset[i].."DEF10"  
local ds = sas.exist(_valds)
```

if ds == 0 then

```
  print( sas.catx("","E","RROR:変数情報が格納されたデータセットがありません") )
```

elseif dupobs > 0 and lenobs > 0 then

```
  print( sas.catx("","E","RROR:変数名が重複し、文字数が規定以上です") )
```

elseif dupobs > 0 then

```
  print( sas.catx("","E","RROR:同じ変数名が複数有ります") )
```

elseif lenobs > 0 then

```
  print( sas.catx("","E","RROR:変数名、ラベル名が規定文字数を超過しています") )
```

```

else
  sas.submit([[
    data _NULL_;
      set @tag@DEF10 end=_EOF ;
      if ( _N_=1 ) then call execute("data @tag@ ;" );
      call execute("attrib "||trim(_NAME)) ;
      call execute(" " ||trim(ATT_label) ) ;
      call execute(" " ||trim(ATT_length) ) ;
      call execute(" " ||trim(ATT_format) ) ;
      call execute(" ;" );
      call execute("call missing("||trim(_NAME)||") ;" );
      call execute("stop ;");

      if ( _EOF ) then call execute("run ;" );
    run ;
    *----- 作成したものを保存;
      data OUT.@tag@ ; set @tag@ ; run;
  ]])

  end
end;
endsubmit;
run;

```

参考資料

- ・Base SAS 9.4 プロシジャガイド 第4版
(<http://www.sas.com/offices/asiapacific/japan/service/help/pdf/v94/proc.pdf>)
- ・Base SAS 9.4 Procedures Guide, Seventh Edition
(<https://support.sas.com/documentation/cdl/en/proc/70377/HTML/default/viewer.htm#n1w8nl91tml15dn1mw9p5l8oj6hy.htm>)