

## STAT:5400

More on SAS Macros  
Some useful Character Functions in  
SAS

Lecture 27  
Nov. 12, 2018  
Kate Cowles  
374 SH

kate-cowles@uiowa.edu

## Iterative and conditional processing in SAS macro programs

- `mprint` option was used so that SAS log will show code actually generated by macro programs

```

164      %macro daily;
165          proc means data=books.ytdsales(where=(datesold=today()))
166              maxdec=2 sum;
167              title "Daily Sales Report for &sysdate";
168              class section;
169              var salepric;
170          run;
171          %if &sysday=Friday %then %do;
172              proc means data=books.ytdsales
173                  (where=(today()-6 le datesold le today()))
174                  sum maxdec=2;
175                  title "Weekly Sales Report Week Ending &sysdate";
176                  class section;
177                  var salepric;
178              run;
179          %end;
180      %mend daily;
181
182      %daily
MPRINT(DAILY):  proc means
                  data=books.ytdsales(where=(datesold=today())) maxdec=2 sum;
MPRINT(DAILY):  title "Daily Sales Report for 16JUL03";
MPRINT(DAILY):  class section;

```

```

MPRINT(DAILY):  var salepric;
MPRINT(DAILY):  run;

```

```

187      %macro makesets;
188          data
189              %do i=1 %to 12;
190                  month&i
191              %end;
192          ;
193          set books.ytdsales;
194          mosale=month(datesold);
195          if mosale=1 then output month1;
196          %do i=2 %to 12;
197              else if mosale=&i then output month&i;
198          %end;
199          run;
200      %mend makesets;
201
202      %makesets
MPRINT(MAKESETS):  data month1 month2 month3 month4 month5 month6
month7 month8 month9 month10 month11 month12 ;
MPRINT(MAKESETS):  set books.ytdsales;
MPRINT(MAKESETS):  mosale=month(datesold);
MPRINT(MAKESETS):  if mosale=1 then output month1;
MPRINT(MAKESETS):  else if mosale=2 then output month2;
MPRINT(MAKESETS):  else if mosale=3 then output month3;
MPRINT(MAKESETS):  else if mosale=4 then output month4;
MPRINT(MAKESETS):  else if mosale=5 then output month5;
MPRINT(MAKESETS):  else if mosale=6 then output month6;
MPRINT(MAKESETS):  else if mosale=7 then output month7;
MPRINT(MAKESETS):  else if mosale=8 then output month8;
MPRINT(MAKESETS):  else if mosale=9 then output month9;
MPRINT(MAKESETS):  else if mosale=10 then output month10;
MPRINT(MAKESETS):  else if mosale=11 then output month11;
MPRINT(MAKESETS):  else if mosale=12 then output month12;
MPRINT(MAKESETS):  run;

```

## Some useful character functions in SAS

- character functions operate on the values of character variables
- also called string functions

## The Compress function: removing selected characters

- **compress** function with single argument removes only blanks
- optional second argument is list of all characters to be removed (enclosed in single or double quotes)

```
options linesize = 72 ;

data phonebook ;
length first last $ 10 ;
input first last phone $ 21-33 ;
phone1 = compress(phone) ;
phone2 = compress(phone,'()- ') ;
datalines ;
Kate Cowles          (319)354-3684
Brendan Holly        3193543684
Mysterious Stranger  515 555 1212
;

proc print data = phonebook ;
run ;
```

Obs	first	last	phone	phone1	phone2
1	Kate	Cowles	(319)354-3684	(319)354-3684	3193543684
2	Brendan	Holly	3193543684	3193543684	3193543684
3	Mysterious	Stranger	515 555 1212	5155551212	5155551212

## SUBSTR: extract a part of a character variable

- arguments
  - character variable
  - which position to start at
  - how many characters to extract
- extracted sections will be padded out with blanks to length of original variable

## The concatenation operator: ||

- glues together pieces to make a complete character string

```
data phonebook ;
set phonebook ;
area = substr(phone2,1,3) ;
exchg = substr(phone2,4,3) ;
rest = substr(phone2,7,4) ;
phone3 = '(' || area || ')' || exchg || '-' || rest ;
run ;

proc print data = phonebook;
run ;
```

Obs	first	last	phone	phone1	phone2
1	Kate	Cowles	(319)354-3684	(319)354-3684	3193543684
2	Brendan	Holly	3193543684	3193543684	3193543684
3	Mysterious	Stranger	515 555 1212	5155551212	5155551212

```
Obs area exchg rest phone3
```

Obs	area	exchg	rest	phone3
1	319	354	3684	(319 )354 -3684
2	319	354	3684	(319 )354 -3684
3	515	555	1212	(515 )555 -1212

## Fixed version

```
data phonebook ;
set phonebook ;
area = substr(phone2,1,3) ;
exchg = substr(phone2,4,3) ;
rest = substr(phone2,7,4) ;
phone3 = '(' || compress(area) || ')' || compress(exchg) ||
        '-' || compress(rest) ;
run ;

proc print data = phonebook;
run ;
```

Obs	first	last	phone	phone1
1	Kate	Cowles	(319)354-3684	(319)354-3684
2	Brendan	Holly	3193543684	3193543684
3	Mysterious	Stranger	515 555 1212	5155551212

```
Obs phone2 area exchg rest phone3
```

Obs	phone2	area	exchg	rest	phone3
1	3193543684	319	354	3684	(319)354-3684
2	3193543684	319	354	3684	(319)354-3684
3	5155551212	515	555	1212	(515)555-1212

## Contents of phonebook dataset

```
The CONTENTS Procedure
```

Data Set Name:	WORK.PHONEBOOK	Observations:	3
Member Type:	DATA	Variables:	9
Engine:	V8	Indexes:	0
Created:	10:43 Wednesday, July 16, 2003	Observation Length:	140
Last Modified:	10:43 Wednesday, July 16, 2003	Deleted Observations:	0
Protection:		Compressed:	NO
Data Set Type:		Sorted:	NO
Label:			

#	Variable	Type	Len	Pos
6	area	Char	13	59
7	exchg	Char	13	72
1	first	Char	10	0
2	last	Char	10	10
3	phone	Char	13	20
4	phone1	Char	13	33
5	phone2	Char	13	46
9	phone3	Char	42	98
8	rest	Char	13	85

### -----Engine/Host Dependent Information-----

```
Data Set Page Size: 16384
Number of Data Set Pages: 1
First Data Page: 1
Max Obs per Page: 116
Obs in First Data Page: 3
Number of Data Set Repairs: 0
File Name: /usr/tmp/SAS_workF4D500005960_
mouse/phonebook.sas7bdat
Release Created: 8.0202M0
Host Created: HP-UX
Inode Number: 53106
Access Permission: rw-----
Owner Name: UNKNOWN
File Size (bytes): 24576
```

### -----Alphabetic List of Variables and Attributes-----