SAS PROGRAMMING (Nominal Week 7 material)
* Arrays
* DO groups
* Statements: RETAIN, RENAME, LABEL, FORMAT, SUM
* Using formats in DATA steps
* Conditional execution
* More on missing values


ARRAYS

* look to use if writing the same set of code multiple times
* “arrays” can contain lists of variables
* “arrays” also good for restructuring data sets

Common example 1a: Recoding a set of variables

```sas
/*
Suppose you have a data set "old_data" containing
Variables: a_var, b_var, var3, var4, var5
(all numeric with missing values coded as -999)

Recode -999 as missing=. 
*/
```

```sas
data old_data;
```
input a_var b_var var3 var4 var5 @@;
datalines;
1 2 3 4 5 6 7 -999 8 9 10 11 12 -999 14;
run;

data recode_ex; set old_data;
array all[5] a_var b_var var3 var4 var5;
do ii=1 to 5;
    if all[ii] = -999 then all[ii]=.;
end;
drop ii;
/* can use either [], {}, () to reference array elements */

options nocenter nodate;
proc print;
    run;
run;

Obs  a_var  b_var  var3  var4  var5
1    1      2      3      4      5
2    6      7      .      8      9
3    10     11     12     .     14

/* alternative to get SAS to count array size &
dimension of array */
*

data recode_ex2; set old_data;
array all{*} a_var b_var var3 var4 var5;
do ii=1 to dim(all);
    if all{ii} = -999 then all{ii}=.;
end;
drop ii;

options nocenter nodate;
proc print;
    title 'Recode 2: Using SAS to count array elements';
run;

Recode 2: Using SAS to count array elements

<table>
<thead>
<tr>
<th>Obs</th>
<th>a_var</th>
<th>b_var</th>
<th>var3</th>
<th>var4</th>
<th>var5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>7</td>
<td>.</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>.</td>
<td>14</td>
</tr>
</tbody>
</table>

/* alternative to get SAS to select all numeric variables
   _NUMERIC_ = all numeric variables
   _CHARACTER_ = all character variables
   _ALL_ = all variables
*/

data old_data2;
  input char_var $ a_var b_var var3 var4 var5 @@;
datalines;
a 1 2 3 4 5 b 6 7 -999 8 9 c 10 11 12 -999 14
;run;

data recode_ex3; set old_data2;
  array all[*] _NUMERIC_;
  do ii=1 to dim(all);
    if all[ii] = -999 then all[ii]=.;
  end;
  drop ii;
proc print;
  title 'Recode 3: Using _NUMERIC_ to select elements';
run;

Recode 3: Using _NUMERIC_ to select elements

<table>
<thead>
<tr>
<th>Obs</th>
<th>char_var</th>
<th>a_var</th>
<th>b_var</th>
<th>var3</th>
<th>var4</th>
<th>var5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>b</td>
<td>6</td>
<td>7</td>
<td>.</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>c</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>.</td>
<td>14</td>
</tr>
</tbody>
</table>

/*
   Recoding both numeric and character values using arrays
*/

Data D5;
input name $ sex $ t1 t2 t3 t4 t5 time6 time_7;

ARRAY num_array{*} _NUMERIC_;
ARRAY char_array{*} _CHARACTER_;

/* recode the numeric variables */
DO inum = 1 to dim(num_array);
   if num_array{inum}=-999 then num_array{inum}=.;
END;

/* recode the character variables */
Do ichar = 1 to dim(char_array);
   if char_array{ichar}="-999" then char_array{ichar}=" ";
END;

drop inum ichar;
datalines;
MrSmith -999 6 5 5 5 4 3
-999 F 7 -999 4 4 3 -999 2
;
proc print;
title "Recoding missing values using Arrays using DO loop";
run;

Recoding missing values using Arrays using DO loop

<table>
<thead>
<tr>
<th>Obs</th>
<th>name</th>
<th>sex</th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
<th>t5</th>
<th>time6</th>
<th>time_7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MrSmith</td>
<td></td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>F</td>
<td>7</td>
<td></td>
<td>.</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>.</td>
<td>2</td>
</tr>
</tbody>
</table>

Common example 1b: Recoding a set of variables (with array initialized)

data temps;
   array tempF(4) tempF1-tempF4 (32,50,68,86);
   array tempC(4) tempC1-tempC4;

   do itemp = 1 to 4;
      tempC(itemp) = 5/9*(tempF(itemp)-32);
   end;
   drop itemp;

proc print;
   run;
Common example 2: Creating multiple observations from a single observation

data one;
  input x1 x2 x3 x4;
  datalines;
  60 62 64 68
  80 84 90 98
; 
data two; set one;
  array xx[4] x1-x4;
  do time=1 to 4;
    x=xx[time];
    output;
  end;
  drop x1-x4;
run;
proc print;
title 'Expand one record to multiple records';
run;

Expand one record to multiple records
Obs   time   x
 1     1     60
 2     2     62
 3     3     64
 4     4     68
 5     1     80
 6     2     84
 7     3     90
 8     4     98

Common example 3: Creating one observations from multiple observations

data multi;
  input id time heart_rate;
  datalines;
proc sort data=multi; by id time;
/*
data sorted by the variable "id"
FIRST.id = 1 if first occurrence of new by group variable
LAST.id = 1 if last occurrence of a by group variable
*/
data one; set multi;
   by id;
   array xx[4] x1-x4;
   retain x1-x4;   * values kept from previous observation;
   if FIRST.id=1 then do ii=1 to 4;
      xx[ii]=.;     * elements initialized to missing;
   end;
   xx[time]=heart_rate;
   if LAST.id=1 then output;
   keep id x1-x4;
run;

proc print;
title 'Condense multiple records to one record';
run;

Condense multiple records to one record
Obs  id  x1  x2  x3  x4
    1   1  60  62  64  68
Alternative using PROC TRANSPOSE

```sas
proc transpose data=multi out=one_tran(keep=id hr1-hr4) prefix=HR; by id;
  var heart_rate;
proc print data=one_tran;
title alternative using PROC TRANSPOSE;
run;
```

Alternative using PROC TRANSPOSE

<table>
<thead>
<tr>
<th>Obs</th>
<th>id</th>
<th>HR1</th>
<th>HR2</th>
<th>HR3</th>
<th>HR4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>60</td>
<td>62</td>
<td>64</td>
<td>68</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>80</td>
<td>84</td>
<td>90</td>
<td>98</td>
</tr>
</tbody>
</table>

Common example 4: Inputting values in ARRAY variable list

```sas
Data D1;
  ARRAY ADL{7} ADL1-ADL7;
  input ADL1-ADL7;
  datalines;
    6 6 5 5 5 4 3
  ;
```

or

```sas
Data D2;
  ARRAY ADL{*} ADL1-ADL7;
  input ADL1-ADL7;
  datalines;
    6 6 5 5 5 4 3
  ;
```

or

```sas
Data D3;
  ARRAY ADL{*} t1 t2 t3 t4 t5 time6 time_7;
  input t1 t2 t3 t4 t5 time6 time_7;
  datalines;
    6 6 5 5 5 4 3
  ;
```

A more complicated example: Randomization test for testing equality of 2 populations
use PLAN to generate a set of indices for the randomization test
and then use TRANSPOSE to package the output

/* nitrofen data
   concentrations 0 and 160 will be used to illustrate a randomization test
*/

libname class 'D:\baileraj\Classes\Fall 2003\sta402\data';

data test; set class.nitrofen;
   if conc=0 | conc=160;

proc ttest;
title NITROFEN: t-test of (0, 160) concentrations;
class conc;
   var total;
run;

/*
NITROFEN: t-test of (0, 160) concentrations
The TTEST Procedure

<table>
<thead>
<tr>
<th>Variable</th>
<th>conc</th>
<th>N</th>
<th>Mean</th>
<th>Lower CL</th>
<th>Upper CL</th>
<th>Lower CL</th>
<th>Upper CL</th>
</tr>
</thead>
<tbody>
<tr>
<td>total</td>
<td></td>
<td>10</td>
<td>28.827</td>
<td>26.612</td>
<td>30.988</td>
<td>2.4737</td>
<td>3.5963</td>
</tr>
<tr>
<td>total</td>
<td>160</td>
<td>10</td>
<td>26.612</td>
<td>25.311</td>
<td>29.988</td>
<td>1.6229</td>
<td>2.3594</td>
</tr>
<tr>
<td>total</td>
<td></td>
<td>10</td>
<td>0.2424</td>
<td>0.2424</td>
<td>0.2424</td>
<td>0.2424</td>
<td>0.2424</td>
</tr>
</tbody>
</table>

T-Tests

| Variable | Method      | Variances | DF | t Value | Pr > |t| |
|----------|-------------|-----------|----|---------|-------|
| total    | Pooled      | Equal     | 18| 2.28    | 0.0351|
| total    | Satterthwaite| Unequal   | 18| 2.28    | 0.0372|

Equality of Variances

<table>
<thead>
<tr>
<th>Variable</th>
<th>Method</th>
<th>Num DF</th>
<th>Den DF</th>
<th>F Value</th>
<th>Pr &gt; F</th>
</tr>
</thead>
<tbody>
<tr>
<td>total</td>
<td>Folded</td>
<td>9</td>
<td>9</td>
<td>2.32</td>
<td>0.2252</td>
</tr>
</tbody>
</table>

*/

proc print;
title NITROFEN: print of (0, 160) concentrations;
   var conc total;
run;

/*
NITROFEN: print of (0, 160) concentrations

<table>
<thead>
<tr>
<th>Obs</th>
<th>conc</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>27</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>32</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>34</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>33</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>36</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>34</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>33</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>24</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>31</td>
</tr>
<tr>
<td>11</td>
<td>160</td>
<td>29</td>
</tr>
</tbody>
</table>
9

12 160 29
13 160 23
14 160 27
15 160 30
16 160 31
17 160 30
18 160 26
19 160 29
20 160 29

/*
proc transpose data=test prefix=xx out=tran_out;
  var total;
run;

data obs_test; set tran_out;
  type = ‘O’;
run;

proc print data=obs_test;
title 'Randomization test:  observed data';
run;

/*
Randomization test:  observed data
__
| N | A | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | t |
| O | M | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | y |
| b | E | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | y |
| s | _ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | e |

1 total 27 32 34 33 36 34 33 30 24 31 29 29 23 27 30 31 30 26 29 29 0
*/

proc plan;
  factors test=4000 ordered in=20;
  output out=d_permut;
run;

proc transpose data=d_permut prefix=in out=out_permut(keep=in1-in20); by test;
run;

proc print data=out_permut;
run;

data _null_; set obs_test;
  file 'D:\baileraj\Classes\Fall 2003\sta402\SAS-programs\week7-perm.data';
  put type xx1-xx20;
run;

data _null_; set out_permut;
  type = 'P';  * permutation data;
  file 'D:\baileraj\Classes\Fall 2003\sta402\SAS-programs\week7-perm.data';
mod;   /* mod option adds lines to existing file */
  put type in1-in20;
run;

/*  week7-perm.data ...
O 27 32 34 33 36 34 33 30 24 31 29 29 23 27 30 31 30 26 29 29
data perm_data;
  array both(20) x1-x10 y1-y10; /* array for observed values */
  array ins(20) in1-in20;        /* index array */
  array perms(20) xp1-xp10 yp1-yp10; /* array for permuted values */
  infile 'D:\baileraj\Classes\Fall 2003\sta402\SAS-programs\week7-perm.data';
  input type $ @;
  if type='O' then do;
    input x1-x10 y1-y10;
    obs_diff = mean(of x1-x10) - mean(of y1-y10);
    retain obs_diff x1-x10 y1-y10;
  end;
  else do;
    input in1-in20;
    do ii = 1 to 20;
      perms{ii} = both{ ins[ii] };
    end;
    perm_diff = mean(of xp1-xp10) - mean(of yp1-yp10);
    perm_ge = (perm_diff >= obs_diff); /* 1-tailed; */
    perm_2tail = (abs(perm_diff) >= abs(obs_diff)); /* 2-tailed; */
    keep obs_diff perm_diff perm_ge perm_2tail;
  end;
  output;
end;
/
/*
proc print;
  run;
*/

proc freq data=perm_data;
title 'NITROFEN: randomization test -> upper tail P-value';
  table perm_ge perm_2tail;
run;

NITROFEN: randomization test -> upper tail P-value

The FREQ Procedure

<table>
<thead>
<tr>
<th>perm_ge</th>
<th>Frequency</th>
<th>Percent</th>
<th>Cumulative Frequency</th>
<th>Cumulative Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3927</td>
<td>98.18</td>
<td>3927</td>
<td>98.18</td>
</tr>
<tr>
<td>1</td>
<td>73</td>
<td>1.83</td>
<td>4000</td>
<td>100.00</td>
</tr>
</tbody>
</table>

< -- - P_{upper} = 0.0183>
/*

Constructing a test of spatial randomness versus regular or clustered patterns of response.

Problem:
- 4 trees were observed in a hypothetical square plot
- are these trees clustered in this plot? regularly spaced?
- how can you check?

Strategy:
1. Determine nearest-neighbor distances
2. Calculate the average NN distance
3. Generate a sample of observations that are randomly distributed in the region of interest
4. Calculate the average NN distance for this set
5. Repeat steps 3 and 4 a large number of times
6. P-values are the proportional of generated samples that were more extreme than observed

History:
First examined this problem (12apr95) in [-.classes.ies612]monte_spatial.sas (old VAX file)

*/

options ls=74;

data plot1;
  title plot1 assessment of pattern;
  array xobs xobs1-xobs4;
  array yobs yobs1-yobs4;
  array nnobs nnobs1-nnobs4;

  input xobs1-xobs4 yobs1-yobs4 @@;

  /* Determine the observed NN distance and average */
  sumnnobs = 0;
  do i=1 to 4;   * find NN distance for each point ;
nnobs(i) = 100;  * initialize distances to be large;

do j=1 to 4;     * compare the ith point to all others;
    d=sqrt( (xobs(i)-xobs(j))**2 + (yobs(i)-yobs(j))**2 );
    if (d<nnobs(i)) and (d>0) then nnobs(i)=d;
* output;          * output if debugging desired;
end;

sumnnobs=sumnnobs+nnobs(i);
end;

avgnnobs = sumnnobs/4;  * observed average NN distance;

datalines;
.25 .75 .25 .75 .75 .75 .25 .25
;

proc print;
data mccsr1; set plot1;
    array xobs xobs1-xobs4;
    array yobs yobs1-yobs4;
    array xsim xsim1-xsim4;
    array ysim ysim1-ysim4;
    array nnobs nnobs1-nnobs4;
    array nncsr nncsr1-nncsr4;
/*  Generate a large number of CSR plots with 4 trees  */
/*  CSR = completely spatially random                  */

* initialize counters of nn avg dist le or ge than observed;
numle = 0;       numge = 0;

do isim = 1 to 1000;

    do ii = 1 to 4;
        xsim(ii) = ranuni(0);
        ysim(ii) = ranuni(0);
    end;

/*  Find NN distance for the simulated trees */

    sumnnncsr = 0;
    do i=1 to 4;
        nncsr(i) = 100;  * initialize;
    end;

    do j=1 to 4;
        d=sqrt( (xobs(i)-xobs(j))**2 + (yobs(i)-yobs(j))**2 );
        if (d<nncsr(i)) and (d>0) then nncsr(i)=d;
    end;

    sumnncsr = sumnncsr+nnncsr(i);
end;

avgnnncsr = sumnnncsr/4;  * observed average NN distance for CSR;
do j=1 to 4;
   d=sqrt( (xsim(i)-xsim(j))**2 + (ysim(i)-ysim(j))**2 );
   if (d<nncsr(i)) and (d>0) then nncsr(i)=d;
   *      output;   * debugging;
end;

   sumnncsr=sumnncsr+nncsr(i);
end;

   avgnncsr = sumnncsr/4;

   /*
    * Accumulate counts of patterns consistent with regularity/aggreg.
    */

   ile = (avgnncsr <= avgnnobs);
   ige = (avgnncsr >= avgnnobs);

   numle = numle + ile;
   numge = numge + ige;

   drop i j ii xobs1-xobs4 yobs1-yobs4 nnobs1-nnobs4
    sumnnobs sumnncsr;
   output;
end;      * if the isim - simulation loop;

* proc print;

proc freq;
  table ile ige;
run;

The FREQ Procedure

  ile     Frequency   Percent  Cumulative Frequency  Cumulative Percent
            0       46  4.60         46  4.60
           1      954 95.40        1000 100.00

  ige     Frequency   Percent  Cumulative Frequency  Cumulative Percent
            0      954 95.40         954 95.40
           1       46  4.60         1000 100.00

RETAIN

*  tough to perform calculations across observations
* SAS normally initializes each variable to missing
* RETAIN instructs system not to assign a missing value to a variable but to remember a different value

```sas
data retain_demo1;
  input dobs time x;
  retain subject 0;
  if time=1 then subject=subject+1;
  datalines;
  1 1 60
  2 2 62
  3 3 64
  4 4 68
  5 1 80
  6 2 84
  7 3 90
  8 4 98
;
proc print;
  id dobs;
run;
```

```sas
data retain_demo2;
  input dobs time x;
  if time=1 then subject+1; * implicitly retains values for calculations;
  datalines;
  1 1 60
  2 2 62
  3 3 64
  4 4 68
  5 1 80
  6 2 84
  7 3 90
  8 4 98
;
options nocenter;
proc print;
  title2 'implicitly retain with subject+1 statement';
  id dobs;
run;
```

```
implicitly retain with subject+1 statement
```

<table>
<thead>
<tr>
<th>dobs</th>
<th>time</th>
<th>x</th>
<th>subject</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>60</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>62</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>64</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>68</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>80</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>84</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>90</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>98</td>
<td>2</td>
</tr>
</tbody>
</table>
/*
   example: find the average weight by subject using
            DATA step programming
 */

/* STEP 1: read in the data file */
data diet;
  input id @3 date mmddyy8. weight;
  format date mmddyy8.;
datalines;
1 10/01/92 155
1 10/08/92 158
1 10/15/92 158
1 10/22/92 158
2 09/02/92 200
2 09/09/92 198
2 09/16/92 196
2 09/23/92 202;
proc print;
title 'diet data';
  run;

data diet2; set diet;

if _n_=1 then total=weight;
else if id = lag(id) then total=total+weight;
else if id NE lag(id) then total = weight;

proc print;
  run;

Obs  id   date    weight    total
  1   1  10/01/92   155      155
  2   1  10/08/92   158      313
  3   1  10/15/92   158      471
  4   1  10/22/92   158      629
  5   2  09/02/92   200      200
  6   2  09/09/92   198      398
  7   2  09/16/92   196      594
  8   2  09/23/92   202      796

/* STEP 2: 
accumulate the total weight measurements for an individual ID */
data diet3; set diet;
  retain total 0;
  if id = lag(id) then total=total+weight;
  else if id NE lag(id) then total = weight;
  proc print;
  run;

Obs  id   date    weight    total
  1   1  10/01/92   155      155
  2   1  10/08/92   158      313
  3   1  10/15/92   158      471
  4   1  10/22/92   158      629
  5   2  09/02/92   200      200
  6   2  09/09/92   198      398
  7   2  09/16/92   196      594
  8   2  09/23/92   202      796

/* STEP 3: 
Calculate the average weight and output the desired data set */
proc sort data=diet3; by id;
data diet4; set diet3; by id;
  if LAST.id;   * special variable LAST.id=1 if last value in BY;
  wt_avg = total/4;
  keep id wt_avg;
  proc print;
  run;

Obs  id  wt_avg
  1   1  157.25
  2   2  199.00

/*
STEPS 2 and 3 ALTERNATIVE:
*/
Accumulate cumulative weight and average of measurements
And then extract the last measurement for each ID
*/

data diet5; set diet;
retain total 0 count 0;
if id = lag(id) then do;
    total = total + weight;
    count = count + 1;
    wt_avg = total / count;
end;
else if id NE lag(id) then do;
    total = weight;
    count = 1;
    wt_avg = total / count;
end;

proc print;
run;

obs    id        date    weight    total    count    wt_avg
1      1    10/01/92      155      155       1      155.00
2      1    10/08/92      158      313       2      156.50
3      1    10/15/92      158      471       3      157.00
4      1    10/22/92      158      629       4      157.25
5      2    09/02/92      200      200       1      200.00
6      2    09/09/92      198      398       2      199.00
7      2    09/16/92      196      594       3      198.00
8      2    09/23/92      202      796       4      199.00

data diet6; set diet5; by id;
    if LAST.id;
    keep id wt_avg;
proc print;
run;

obs    id    wt_avg
1      1    157.25
2      2    199.00

/*
example: find the total time enrolled for each participant
    [motivated by an example where people may enroll/
        disenroll in a program during different quarters]
*/

options formdlim="-";

data test;
input id xstart xstop;
datalines;
1 15 25
2 10 12
2 18 22
proc print;
  run;

data test2; set test; by id;
array start(9) start1-start9;
array stop(9) stop1-stop9;
array times(9) times1-times9;

retain count 0;
retain start1-start9 stop1-stop9 times1-times9;

if FIRST.id=1 then do;  * initialize count and arrays with new ID;
  count = 0;
  do ii=1 to 9;
    start{ii} = .;
    stop{ii} = .;
    times{ii} = .;
  end;
end;
count = count + 1;
start{count} = xstart;
stop{count} = xstop;
times{count} = xstop - xstart;

if LAST.id=1 then output;  * output results if last obs for ID;
drop xstart xstop ii;
run;

data test3; set test2;
  total_time = sum(of times1-times9);
run;

proc print;
  run;

material from
C:\Documents and Settings\John Bailer\My
Documents\baileraj\Classes\Fall 2003\sta402\handouts\week8-
15oct03.doc

COMBINING AND MANAGING SAS DATA SETS (nominal week 8 material)
* SET statement for concatenation and interleaving
* MERGE statement for joining observations
* UPDATE statement for updating a master file (maybe)
* Special variables: IN, END, FIRST, and LAST
* Creating multiple data sets in one DATA step
* Reshaping data sets
* Managing data sets using PROC COPY and PROC DATASETS
* Transporting data sets between hosts

Reference:


<table>
<thead>
<tr>
<th>Temporary versus Permanent SAS data sets (Delwiche and Slaughter – Ch. 2.9)</th>
</tr>
</thead>
</table>

* if you use a data set more than once, it may be more efficient to save it as a permanent data set.

* SAS data set names all have two levels – the first level is its LIBREF (SAS data library referenced) and the second is the MEMBER name that identifies the data set within the library.

* LIBREF points to a particular location – often a physical location (e.g. disk) or a logical notation (e.g. directory).

/* example */
libname class 'D:\baileraj\Classes\Fall 2003\sta402\data';

data nitrofen_A; set class.nitrofen;
   brood=1; count=brood1; conc=conc; output;
   brood=2; count=brood2; conc=conc; output;
   brood=3; count=brood3; conc=conc; output;
   keep brood count conc;

* in the example above, two data sets are defined/referenced - LIBREF = “class” and MEMBER = “nitrofen” (where the “class” LIBREF refers to directory D:\baileraj\Classes\Fall 2003\sta402\data)

LIBREF = “WORK” (default – reserved for temporary SAS data sets) and MEMBER = “nitrofen_A” [to make this permanent]
data class.nitrofen_A; set class.nitrofen;
   brood=1; count=brood1; conc=conc; output;
   brood=2; count=brood2; conc=conc; output;
   brood=3; count=brood3; conc=conc; output;
keep brood count conc;

Create data sets for illustration by
Creating multiple data sets in one DATA step

libname class 'D:\baileraj\Classes\Fall 2003\sta402\data';

data dbrood1 dbrood2 dbrood3; set class.nitrofen;
   animal = _n_;    * create animal ID number;
   brood=1; count=brood1; conc=conc; output dbrood1;
   brood=2; count=brood2; conc=conc; output dbrood2;
   brood=3; count=brood3; conc=conc; output dbrood3;
keep animal brood count conc;

ODS RTF file='D:\baileraj\Classes\Fall 2003\sta402\SAS-programs\week6-tab1.rtf';

proc print data=dbrood1;
   title 'Nitrofen Brood 1 data';
   id animal;
run;
proc print data=dbrood2;
   title 'Nitrofen Brood 2 data';
   id animal;
run;
proc print data=dbrood3;
   title 'Nitrofen Brood 3 data';
   id animal;
run;

ODS RTF close;

<table>
<thead>
<tr>
<th>Obs</th>
<th>animal</th>
<th>conc</th>
<th>brood</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>47</td>
<td>47</td>
<td>310</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>48</td>
<td>48</td>
<td>310</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>49</td>
<td>49</td>
<td>310</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>50</td>
<td>50</td>
<td>310</td>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>
SET statement for concatenation and interleaving – essentially adds observations to a data set

```
data all3_stack; set dbrood1 dbrood2 dbrood3;
title 'Concatenate/stack all 3 brood-specific files into one file';

ODS RTF file='D:\baileraj\Classes\Fall 2003\sta402\SAS-programs\week8-prt2.rtf';

proc print;	id animal;
run;

ODS RTF close;
```
animal conc brood count
--- ---- --- ----- 
50 310 1 5
1 0 2 14
2 0 2 12
... ... ...
47 310 2 0
48 310 2 0
49 310 2 0
50 310 2 0
1 0 3 10
2 0 3 15
... ... ...
47 310 3 0
48 310 3 0
49 310 3 0
50 310 3 0

data all3_interleave;
  set dbrood1 dbrood2 dbrood3;
  by animal;
/*
 - assumes data sorted by animal ID
 - if not, then use PROC SORT first
*/
title 'Interleave all 3 brood-specific files into one file';

ODS RTF file='D:\baileraj\Classes\Fall 2003\sta402\SAS-programs\week8-prt3.rtf';

proc print;
  id animal;
  run;

ODS RTF close;
animal | conc | brood | Count |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>47</td>
<td>310</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>47</td>
<td>310</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>47</td>
<td>310</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>48</td>
<td>310</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>48</td>
<td>310</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>48</td>
<td>310</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>49</td>
<td>310</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>49</td>
<td>310</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>49</td>
<td>310</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>50</td>
<td>310</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>50</td>
<td>310</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>50</td>
<td>310</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>

**MERGE statement for joining observations – essentially adds variables to a data set**

```sas
/* rename count prior to merging data sets */
data dbrood1; set dbrood1;
   rename count=brood1;
drop brood;
data dbrood2; set dbrood2;
   rename count=brood2;
drop brood;
data dbrood3; set dbrood3;
   rename count=brood3;
drop brood;

data all3_merge;
   merge dbrood1 dbrood2 dbrood3;
   by animal;
/* assumes data sorted by animal ID */
title 'Merge all 3 brood-specific files into one file';
proc print data=dbrood1; run;

ods rtf file='D:\baileraj\Classes\Fall 2003\sta402\SAS-programs\week8-prt4.rtf';
proc print;
id animal;
run;
ods rtf close;
```
IN= data set option is used with the SET, MERGE, MODIFY, and UPDATE statements to create and name a variable that indicates whether the data set contributed data to the current observation.

/*
Based on Cody and Pass (Ch 3, Examples 5 and 6)
*/
options nodate nocenter;
data demo;
  input ID_num sex $ state $ @@;
datalines;
  1 M NY 2 F NJ 3 F NJ 4 M NY 5 M NY
;
proc print data=demo;
title ID_Num/Sex/State information;
  run;

data dept;
  input id dept_name $ salary @@;
datalines;
  1 PARTS 21000 2 SALES 45000 3 PARTS 20000 5 SALES 35000
;
proc print data=dept;
title ID/dept_name/salary;
run;

ID_Num/Sex/State information

<table>
<thead>
<tr>
<th>Obs</th>
<th>ID_num</th>
<th>sex</th>
<th>state</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>M</td>
<td>NY</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>F</td>
<td>NJ</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>F</td>
<td>NJ</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>M</td>
<td>NY</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>M</td>
<td>NY</td>
</tr>
</tbody>
</table>

ID/dept_name/salary

<table>
<thead>
<tr>
<th>dept_</th>
<th>Obs</th>
<th>id</th>
<th>name</th>
<th>salary</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>PARTS</td>
<td>21000</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>SALES</td>
<td>45000</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3</td>
<td>PARTS</td>
<td>20000</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>5</td>
<td>SALES</td>
<td>35000</td>
</tr>
</tbody>
</table>

options nodate nocenter formdlim="-";
data combo;
    merge demo(rename=(ID_Num=id)) dept;
    by id;
proc print data=combo;
    id id;
title ‘Both demographic and salary data combined’;
    run;

Both demographic and salary data combined

<table>
<thead>
<tr>
<th>dept_</th>
<th>id</th>
<th>sex</th>
<th>state</th>
<th>name</th>
<th>salary</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>M</td>
<td>NY</td>
<td>PARTS</td>
<td>21000</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>F</td>
<td>NJ</td>
<td>SALES</td>
<td>45000</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>F</td>
<td>NJ</td>
<td>PARTS</td>
<td>20000</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>M</td>
<td>NY</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>M</td>
<td>NY</td>
<td>SALES</td>
<td>35000</td>
</tr>
</tbody>
</table>
Suppose you didn’t want to include the observation for the ID with no Dept_name and Salary information? The IN= construction is a handy tool for this task.

```sas
data combo2;
  merge demo(rename=(ID_Num=id) IN=demo_used)
    dept (IN=dept_used);
  by id;

  put ' Observation = ' _n_;    * PUTs used to illustrate IN=;
  put '   demo_used = ' demo_used;
  put '   dept_used = ' dept_used;
  put '--------------------------';

  if demo_used=1 AND dept_used=1;

proc print data=combo2;
  id id;
  title 'Both demographic and salary data combined';
  title2 'only data with matched records included';
  run;
```

From the SAS LOG . . .

Observation = 1
  demo_used = 1
  dept_used = 1
--------------------------
Observation = 2
  demo_used = 1
  dept_used = 1
--------------------------
Observation = 3
  demo_used = 1
  dept_used = 1
--------------------------
Observation = 4
  demo_used = 1
  dept_used = 0
--------------------------
Observation = 5
  demo_used = 1
  dept_used = 1
--------------------------

From the OUTPUT . . .
Both demographic and salary data combined only data with matched records included

<table>
<thead>
<tr>
<th>id</th>
<th>sex</th>
<th>state</th>
<th>name</th>
<th>salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>M</td>
<td>NY</td>
<td>PARTS</td>
<td>21000</td>
</tr>
<tr>
<td>2</td>
<td>F</td>
<td>NJ</td>
<td>SALES</td>
<td>45000</td>
</tr>
<tr>
<td>3</td>
<td>F</td>
<td>NJ</td>
<td>PARTS</td>
<td>20000</td>
</tr>
<tr>
<td>5</td>
<td>M</td>
<td>NY</td>
<td>SALES</td>
<td>35000</td>
</tr>
</tbody>
</table>

**UPDATE statement for updating a master file (maybe)**

Suppose the data set COMBO2 from the last example was a master file that was to have records updated based upon the input of another file.

```sas
proc print data=combo2;
  title 'Master data file';
  run;

See above . . .

data newstuff;
  input id salary;
  datalines;
  2 22500
  5 39000
;

data new_master_file;
  update combo2 newstuff;
  by id;

proc print data=new_master_file;
  id id;
  title 'Updated master file - raises included 14oct03';
  run;
```

Updated master file - raises included 14oct03

<table>
<thead>
<tr>
<th>id</th>
<th>sex</th>
<th>state</th>
<th>name</th>
<th>salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>M</td>
<td>NY</td>
<td>PARTS</td>
<td>21000</td>
</tr>
<tr>
<td>2</td>
<td>F</td>
<td>NJ</td>
<td>SALES</td>
<td>22500</td>
</tr>
</tbody>
</table>
Reshaping data sets

* see week 7 discussion of using ARRAYS and OUTPUT statements to construct files with multiple records from one record (or one record from multiple records)

* see use of PROC TRANSPOSE to change data rows to columns

Managing data sets using PROC COPY and PROC DATASETS

Ref: Cody and Pass Ch 14 (Efficiency chapter)

Can use PROC DATASETS to modify/add variables, labels, formats, etc.

```
libname class 'D:\baileraj\Classes\Fall 2003\sta402\data';

proc print data=class.meat;
title 'MEAT data BEFORE PROC DATASETS manipulation';
  run;

PROC DATASETS LIBRARY=class;
  MODIFY meat;
  label logcount = 'log10(Bacterial count)';
  label condition = 'Meat Packaging Condition';
  count = 10**(logcount);
  label count = 'Raw Bacterial count';
  run;

proc print data=class.meat;
title 'MEAT data after PROC DATASETS manipulation';
  run;
```

* efficient for large data sets since PROC DATASETS works on entire data set and not on individual observations (so can be faster and more efficient)

* can also be used to rename datasets . . .

```
PROC DATASETS LIBRARY=class;
  CHANGE old_name = new_name;
  run;
```
**Transporting data sets between hosts**

**Bringing data sets into SAS that were created in other software (e.g. Excel, etc.)**

Spreadsheet from Aaron Roberts

<table>
<thead>
<tr>
<th>Individual</th>
<th>Treatment (Hours)</th>
<th>% Mass Change</th>
<th>Gill TBAR</th>
<th>Liver TBAR</th>
<th>Gill Mt</th>
<th>Liver Mt</th>
</tr>
</thead>
<tbody>
<tr>
<td>386</td>
<td>0</td>
<td>0.99089467</td>
<td>3.0257</td>
<td>1.542</td>
<td>0.8452</td>
<td>0.485465</td>
</tr>
<tr>
<td>387</td>
<td>0</td>
<td>1.1202381</td>
<td>4.2563</td>
<td>1.425</td>
<td>0.9563</td>
<td>0.123555</td>
</tr>
<tr>
<td>389</td>
<td>0</td>
<td>1.06237624</td>
<td>2.5157</td>
<td>1.358</td>
<td>0.7542</td>
<td>0.986561</td>
</tr>
<tr>
<td>390</td>
<td>0</td>
<td>1.0382716</td>
<td>3.5457</td>
<td>1.398</td>
<td>0.6524</td>
<td>1.25</td>
</tr>
<tr>
<td>391</td>
<td>0</td>
<td>1.07125749</td>
<td>2.0253</td>
<td>1.252</td>
<td>0.8564</td>
<td>0.546547</td>
</tr>
<tr>
<td>350</td>
<td>12</td>
<td>1.11851852</td>
<td>3.6644</td>
<td>2.33243</td>
<td>2.17365269</td>
<td>1.82</td>
</tr>
<tr>
<td>356</td>
<td>12</td>
<td>1.06725664</td>
<td>2.8228</td>
<td>2.11679</td>
<td>1.87894737</td>
<td>0.123565</td>
</tr>
<tr>
<td>366</td>
<td>12</td>
<td>1.03030303</td>
<td>10.0213</td>
<td>1.68132</td>
<td>1.58741259</td>
<td>2.2123</td>
</tr>
<tr>
<td>371</td>
<td>12</td>
<td>1.06329114</td>
<td>7.3597</td>
<td>1.65815</td>
<td>2.21090909</td>
<td>2.564523</td>
</tr>
<tr>
<td>383</td>
<td>12</td>
<td>1.05410256</td>
<td>3.6596</td>
<td>2.18386</td>
<td>5</td>
<td>1.985654</td>
</tr>
<tr>
<td>363</td>
<td>24</td>
<td>1.16489362</td>
<td>2.6482</td>
<td>2.39454</td>
<td>1.5270936</td>
<td>2.65569</td>
</tr>
<tr>
<td>367</td>
<td>24</td>
<td>1.14606742</td>
<td>1.9737</td>
<td>1.34201</td>
<td>1.45525292</td>
<td>1.155652</td>
</tr>
<tr>
<td>368</td>
<td>24</td>
<td>1.15</td>
<td>1.1539</td>
<td>1.68515</td>
<td>2.21090909</td>
<td>2.564523</td>
</tr>
<tr>
<td>377</td>
<td>24</td>
<td>1.15263158</td>
<td>2.0995</td>
<td>2.23705</td>
<td>1.59183673</td>
<td>1.985625</td>
</tr>
<tr>
<td>340</td>
<td>24</td>
<td>1.09111111</td>
<td>1.5814</td>
<td>2.58947</td>
<td>1.16286645</td>
<td>1.545856</td>
</tr>
<tr>
<td>359</td>
<td>48</td>
<td>15.365</td>
<td>1.31079</td>
<td>1.7330167</td>
<td>5.12213</td>
<td></td>
</tr>
<tr>
<td>365</td>
<td>48</td>
<td>1.07831325</td>
<td>15.7033</td>
<td>1.46148</td>
<td>1.1553981</td>
<td>2.233124</td>
</tr>
<tr>
<td>370</td>
<td>48</td>
<td>1.08888889</td>
<td>16.7038</td>
<td>1.5632</td>
<td>0.8831003</td>
<td>3.654565</td>
</tr>
<tr>
<td>378</td>
<td>48</td>
<td>1.06666667</td>
<td>20.7736</td>
<td>2.04189</td>
<td>1.21146953</td>
<td>1.255465</td>
</tr>
<tr>
<td>382</td>
<td>48</td>
<td>1.14</td>
<td>14.2439</td>
<td>1.4526</td>
<td>3.11363636</td>
<td>4.256456</td>
</tr>
<tr>
<td>343</td>
<td>168</td>
<td>1.04123711</td>
<td>16.4728</td>
<td>1.69528</td>
<td>1.93333333</td>
<td>14.5</td>
</tr>
<tr>
<td>349</td>
<td>168</td>
<td>1.02439024</td>
<td>11.0246</td>
<td>1.74338</td>
<td>1.50996016</td>
<td>7.83</td>
</tr>
<tr>
<td>357</td>
<td>168</td>
<td>15.7033</td>
<td>1.46148</td>
<td>1.1553981</td>
<td>2.233124</td>
<td></td>
</tr>
<tr>
<td>373</td>
<td>168</td>
<td>1.05</td>
<td>12.4249</td>
<td>2.04496</td>
<td>1.37540453</td>
<td>5.25623</td>
</tr>
<tr>
<td>375</td>
<td>168</td>
<td>16.1007</td>
<td>18.2306</td>
<td>1.9745229</td>
<td>9.584662</td>
<td></td>
</tr>
<tr>
<td>376</td>
<td>336</td>
<td>10.5593</td>
<td>1.6243</td>
<td>3.33834586</td>
<td>10.25656</td>
<td></td>
</tr>
<tr>
<td>381</td>
<td>336</td>
<td>13.4313</td>
<td>1.1452</td>
<td>1.20972644</td>
<td>4.27</td>
<td></td>
</tr>
<tr>
<td>384</td>
<td>336</td>
<td>12.6583</td>
<td>1.1871</td>
<td>0.2948914</td>
<td>22.26266</td>
<td></td>
</tr>
<tr>
<td>393</td>
<td>336</td>
<td>11.9653</td>
<td>1.2565</td>
<td>1.52635698</td>
<td>9.582623</td>
<td></td>
</tr>
<tr>
<td>394</td>
<td>336</td>
<td>8.9309</td>
<td>1.02719</td>
<td>2.07421875</td>
<td>12.25466</td>
<td></td>
</tr>
<tr>
<td>342</td>
<td>336</td>
<td>9.4486</td>
<td>1.1895</td>
<td>1.99090909</td>
<td>8.256265</td>
<td></td>
</tr>
<tr>
<td>347</td>
<td>504</td>
<td>4.8959</td>
<td>1.3565</td>
<td>1.8956466</td>
<td>8.254565</td>
<td></td>
</tr>
<tr>
<td>392</td>
<td>504</td>
<td>1.5335</td>
<td>1.325</td>
<td>1.7525646</td>
<td>9.256486</td>
<td></td>
</tr>
<tr>
<td>343</td>
<td>504</td>
<td>3.5363</td>
<td>1.456</td>
<td>2.23564</td>
<td>14.25846</td>
<td></td>
</tr>
<tr>
<td>341</td>
<td>504</td>
<td>5.1245</td>
<td>1.2958</td>
<td>3.1256</td>
<td>16.25615</td>
<td></td>
</tr>
<tr>
<td>364</td>
<td>672</td>
<td>4.2543</td>
<td>1.1545</td>
<td>2.26563</td>
<td>12.02216</td>
<td></td>
</tr>
<tr>
<td>401</td>
<td>672</td>
<td>3.8569</td>
<td>1.2696</td>
<td>1.84586</td>
<td>6.253123</td>
<td></td>
</tr>
</tbody>
</table>
* use MENUs to import Excel spreadsheet …

FILE > IMPORT DATA (to start the SAS import wizard)

* Browse for location of the data file (can specify options – e.g. names in row 1, cells requested, etc.)

* all that is produced is a note on the LOG window …

NOTE: WORK.ROBERTS was successfully created.

* can save code produced by the IMPORT wizard

PROC IMPORT OUT= WORK.roberts
   DATAFILE= "D:\baileraj\Classes\Fall 2003\sta402\data\Roberts Data.xls"
   DBMS=EXCEL2000 REPLACE;
   GETNAMES=YES;
RUN;

* you should check to see if the data set is correctly structured and entered (don’t always trust wizards!) …

options nocenter nodate;

ODS RTF file='D:\baileraj\Classes\Fall 2003\sta402\SAS-programs\week8-prtc.rtf';
proc contents data=Roberts;
title ‘Roberts Excel spreadsheet imported: CONTENTS’;
run;
proc print data=Roberts;
title ‘Roberts Excel spreadsheet imported: PRINTED’;
run;
ODS RTF Close;

<table>
<thead>
<tr>
<th>Data Set Name:</th>
<th>WORK.ROBERTS</th>
<th>Observations:</th>
<th>37</th>
</tr>
</thead>
<tbody>
<tr>
<td>Member Type:</td>
<td>DATA</td>
<td>Variables:</td>
<td>11</td>
</tr>
<tr>
<td>Engine:</td>
<td>V8</td>
<td>Indexes:</td>
<td>0</td>
</tr>
<tr>
<td>Created:</td>
<td>8:18 Wednesday, October 15, 2003</td>
<td>Observation Length:</td>
<td>88</td>
</tr>
<tr>
<td>Last Modified:</td>
<td>8:18 Wednesday, October 15, 2003</td>
<td>Deleted Observations:</td>
<td>0</td>
</tr>
<tr>
<td>Protection:</td>
<td></td>
<td>Compressed:</td>
<td>NO</td>
</tr>
<tr>
<td>Data Set Type:</td>
<td></td>
<td>Sorted:</td>
<td>NO</td>
</tr>
<tr>
<td>Label:</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
-----Engine/Host Dependent Information-----

<table>
<thead>
<tr>
<th>Data Set Page Size:</th>
<th>8192</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Data Set Pages:</td>
<td>1</td>
</tr>
<tr>
<td>First Data Page:</td>
<td>1</td>
</tr>
<tr>
<td>Max Obs per Page:</td>
<td>92</td>
</tr>
<tr>
<td>Obs in First Data Page:</td>
<td>37</td>
</tr>
<tr>
<td>Number of Data Set Repairs:</td>
<td>0</td>
</tr>
<tr>
<td>File Name:</td>
<td>C:\DOCUME<del>1\baileraj\LOCALS</del>1\Temp\SAS Temporary Files_TD608\roberts.sas7bdat</td>
</tr>
<tr>
<td>Release Created:</td>
<td>8.0202M0</td>
</tr>
<tr>
<td>Host Created:</td>
<td>WIN_PRO</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>#</th>
<th>Variable</th>
<th>Type</th>
<th>Len</th>
<th>Pos</th>
<th>Format</th>
<th>Informat</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>F8</td>
<td>Char</td>
<td>1</td>
<td>80</td>
<td>$</td>
<td>$.</td>
<td>F8</td>
</tr>
<tr>
<td>6</td>
<td>Gill_Mt</td>
<td>Num</td>
<td>8</td>
<td>40</td>
<td>4</td>
<td>$.</td>
<td>Gill Mt</td>
</tr>
<tr>
<td>10</td>
<td>Gill_SOD</td>
<td>Num</td>
<td>8</td>
<td>64</td>
<td>6</td>
<td>$.</td>
<td>Gill SOD</td>
</tr>
<tr>
<td>4</td>
<td>Gill_TBAR</td>
<td>Num</td>
<td>8</td>
<td>24</td>
<td>4</td>
<td>$.</td>
<td>Gill TBAR</td>
</tr>
<tr>
<td>1</td>
<td>Individual</td>
<td>Num</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>$.</td>
<td>Individual</td>
</tr>
<tr>
<td>7</td>
<td>Liver_Mt</td>
<td>Num</td>
<td>8</td>
<td>48</td>
<td>4</td>
<td>$.</td>
<td>Liver Mt</td>
</tr>
<tr>
<td>11</td>
<td>Liver_SOD</td>
<td>Num</td>
<td>8</td>
<td>72</td>
<td>7</td>
<td>$.</td>
<td>Liver SOD</td>
</tr>
<tr>
<td>5</td>
<td>Liver_TBAR</td>
<td>Num</td>
<td>8</td>
<td>32</td>
<td>3</td>
<td>$.</td>
<td>Liver TBAR</td>
</tr>
<tr>
<td>9</td>
<td>SOD_Treatment</td>
<td>Num</td>
<td>8</td>
<td>56</td>
<td>5</td>
<td>$.</td>
<td>SOD Treatment</td>
</tr>
<tr>
<td>2</td>
<td>Treatment__Hours_</td>
<td>Num</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>$.</td>
<td>Treatment (Hours)</td>
</tr>
<tr>
<td>3</td>
<td>__Mass_Change</td>
<td>Num</td>
<td>8</td>
<td>16</td>
<td>16</td>
<td>$.</td>
<td>% Mass Change</td>
</tr>
<tr>
<td>Obs</td>
<td>Individual</td>
<td>Treatment_Hours</td>
<td>_Mass_Change</td>
<td>Gill_TBAR</td>
<td>Liver_TBAR</td>
<td>Gill_Mt</td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>------------</td>
<td>----------------</td>
<td>--------------</td>
<td>-----------</td>
<td>------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>386</td>
<td>0</td>
<td>0.99085</td>
<td>3.0257</td>
<td>1.54200</td>
<td>0.84520</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>387</td>
<td>0</td>
<td>1.12024</td>
<td>4.2563</td>
<td>1.42500</td>
<td>0.95630</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>389</td>
<td>0</td>
<td>1.06238</td>
<td>2.5157</td>
<td>1.35800</td>
<td>0.75420</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>390</td>
<td>0</td>
<td>1.03827</td>
<td>3.5457</td>
<td>1.39800</td>
<td>0.65240</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>391</td>
<td>0</td>
<td>1.07126</td>
<td>2.0253</td>
<td>1.25200</td>
<td>0.85640</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>350</td>
<td>12</td>
<td>1.11852</td>
<td>3.6644</td>
<td>2.33243</td>
<td>2.17365</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>356</td>
<td>12</td>
<td>1.06726</td>
<td>2.8228</td>
<td>2.11679</td>
<td>1.87895</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>366</td>
<td>12</td>
<td>1.03030</td>
<td>10.0213</td>
<td>1.68132</td>
<td>1.58741</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>371</td>
<td>12</td>
<td>1.06329</td>
<td>7.3597</td>
<td>2.25639</td>
<td>0.76780</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>383</td>
<td>12</td>
<td>1.06410</td>
<td>3.6596</td>
<td>2.18386</td>
<td>5.00000</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>363</td>
<td>24</td>
<td>1.16489</td>
<td>2.6482</td>
<td>2.39454</td>
<td>1.52709</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>367</td>
<td>24</td>
<td>1.14607</td>
<td>1.9737</td>
<td>1.34201</td>
<td>1.45525</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>368</td>
<td>24</td>
<td>1.15000</td>
<td>1.1539</td>
<td>1.68515</td>
<td>2.21091</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>377</td>
<td>24</td>
<td>1.15263</td>
<td>2.0995</td>
<td>2.23705</td>
<td>1.59184</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>340</td>
<td>24</td>
<td>1.09111</td>
<td>1.5814</td>
<td>2.58947</td>
<td>1.16287</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>359</td>
<td>48</td>
<td>1.00000</td>
<td>15.3650</td>
<td>1.31079</td>
<td>1.73303</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>365</td>
<td>48</td>
<td>1.07831</td>
<td>15.7033</td>
<td>1.46148</td>
<td>1.15534</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>370</td>
<td>48</td>
<td>1.08889</td>
<td>16.7038</td>
<td>1.56320</td>
<td>0.88331</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>378</td>
<td>48</td>
<td>1.06667</td>
<td>20.7736</td>
<td>2.04189</td>
<td>1.21147</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>382</td>
<td>48</td>
<td>1.14000</td>
<td>14.2439</td>
<td>1.45260</td>
<td>3.11364</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>343</td>
<td>168</td>
<td>1.04124</td>
<td>16.4728</td>
<td>1.69528</td>
<td>1.93333</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>349</td>
<td>168</td>
<td>1.02439</td>
<td>11.0246</td>
<td>1.74338</td>
<td>1.50996</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>357</td>
<td>168</td>
<td>1.05000</td>
<td>12.4249</td>
<td>2.04496</td>
<td>1.37540</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>373</td>
<td>168</td>
<td>0.97345</td>
<td>16.1007</td>
<td>1.82306</td>
<td>1.97452</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>375</td>
<td>168</td>
<td>1.07042</td>
<td>10.5593</td>
<td>1.62430</td>
<td>3.33835</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>376</td>
<td>336</td>
<td>1.04646</td>
<td>13.4313</td>
<td>1.14520</td>
<td>1.20973</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>381</td>
<td>336</td>
<td>1.05256</td>
<td>12.6583</td>
<td>1.18710</td>
<td>0.29489</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>384</td>
<td>336</td>
<td>0.85000</td>
<td>11.9653</td>
<td>1.25650</td>
<td>1.52636</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>393</td>
<td>336</td>
<td>0.95732</td>
<td>8.9309</td>
<td>1.02719</td>
<td>2.07422</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>394</td>
<td>336</td>
<td>0.96860</td>
<td>13.4222</td>
<td>1.29587</td>
<td>1.69565</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>342</td>
<td>336</td>
<td>0.91264</td>
<td>9.4486</td>
<td>1.18950</td>
<td>1.99099</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>347</td>
<td>504</td>
<td>0.95833</td>
<td>4.8959</td>
<td>1.35650</td>
<td>1.89565</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>392</td>
<td>504</td>
<td>0.87879</td>
<td>1.5335</td>
<td>1.32500</td>
<td>1.75256</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>343</td>
<td>504</td>
<td>0.92500</td>
<td>3.5363</td>
<td>1.45600</td>
<td>2.23564</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>341</td>
<td>504</td>
<td>0.91525</td>
<td>5.1245</td>
<td>1.29580</td>
<td>3.12560</td>
<td></td>
</tr>
<tr>
<td>Obs</td>
<td>Individual</td>
<td>Treatment_Hours</td>
<td>__Mass_Change</td>
<td>Gill_TBAR</td>
<td>Liver_TBAR</td>
<td>Gill_Mt</td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>------------</td>
<td>----------------</td>
<td>---------------</td>
<td>-----------</td>
<td>-----------</td>
<td>----------</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>364</td>
<td>672</td>
<td>1.08065</td>
<td>4.2543</td>
<td>1.15450</td>
<td>2.26563</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>401</td>
<td>672</td>
<td>0.64339</td>
<td>3.8569</td>
<td>1.26960</td>
<td>1.84586</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obs</th>
<th>Liver_Mt</th>
<th>F8</th>
<th>SOD_Treatment</th>
<th>Gill_SOD</th>
<th>Liver_SOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.4855</td>
<td>0</td>
<td>0</td>
<td>0.48775</td>
<td>0.7898</td>
</tr>
<tr>
<td>2</td>
<td>0.1236</td>
<td>0</td>
<td>0</td>
<td>1.88183</td>
<td>0.6843</td>
</tr>
<tr>
<td>3</td>
<td>0.9866</td>
<td>0</td>
<td>0</td>
<td>0.37731</td>
<td>1.1780</td>
</tr>
<tr>
<td>4</td>
<td>1.2500</td>
<td>0</td>
<td>0</td>
<td>0.21848</td>
<td>0.9785</td>
</tr>
<tr>
<td>5</td>
<td>0.5465</td>
<td>0</td>
<td>0</td>
<td>0.81323</td>
<td>0.9458</td>
</tr>
<tr>
<td>6</td>
<td>1.8200</td>
<td>12</td>
<td>2</td>
<td>0.84684</td>
<td>7.8490</td>
</tr>
<tr>
<td>7</td>
<td>0.1236</td>
<td>12</td>
<td>0</td>
<td>0.99774</td>
<td>9.0841</td>
</tr>
<tr>
<td>8</td>
<td>2.2123</td>
<td>12</td>
<td>0</td>
<td>2.33190</td>
<td>9.5429</td>
</tr>
<tr>
<td>9</td>
<td>0.9857</td>
<td>12</td>
<td>0</td>
<td>0.32480</td>
<td>2.9936</td>
</tr>
<tr>
<td>10</td>
<td>1.9857</td>
<td>12</td>
<td>0</td>
<td>0.84846</td>
<td>6.7903</td>
</tr>
<tr>
<td>11</td>
<td>2.6557</td>
<td>24</td>
<td>0</td>
<td>0.39234</td>
<td>7.2840</td>
</tr>
<tr>
<td>12</td>
<td>1.1557</td>
<td>24</td>
<td>0</td>
<td>0.99819</td>
<td>9.3823</td>
</tr>
<tr>
<td>13</td>
<td>2.5645</td>
<td>24</td>
<td>0</td>
<td>2.47703</td>
<td>8.4390</td>
</tr>
<tr>
<td>14</td>
<td>1.9856</td>
<td>24</td>
<td>0</td>
<td>1.85319</td>
<td>9.7677</td>
</tr>
<tr>
<td>15</td>
<td>1.5459</td>
<td>24</td>
<td>0</td>
<td>0.32345</td>
<td>7.6877</td>
</tr>
<tr>
<td>16</td>
<td>5.1221</td>
<td>48</td>
<td>0</td>
<td>0.18996</td>
<td>9.5429</td>
</tr>
<tr>
<td>17</td>
<td>2.2331</td>
<td>48</td>
<td>0</td>
<td>0.18968</td>
<td>10.0287</td>
</tr>
<tr>
<td>18</td>
<td>3.6546</td>
<td>48</td>
<td>0</td>
<td>1.34344</td>
<td>10.5438</td>
</tr>
<tr>
<td>19</td>
<td>1.2555</td>
<td>48</td>
<td>0</td>
<td>0.84844</td>
<td>10.5438</td>
</tr>
<tr>
<td>20</td>
<td>4.2565</td>
<td>48</td>
<td>0</td>
<td>0.38496</td>
<td>4.5882</td>
</tr>
<tr>
<td>21</td>
<td>14.5000</td>
<td>168</td>
<td>0</td>
<td>3.23469</td>
<td>14.3481</td>
</tr>
<tr>
<td>22</td>
<td>7.8300</td>
<td>168</td>
<td>0</td>
<td>0.36846</td>
<td>12.6384</td>
</tr>
<tr>
<td>23</td>
<td>5.2562</td>
<td>168</td>
<td>0</td>
<td>0.81981</td>
<td>11.1848</td>
</tr>
<tr>
<td>24</td>
<td>9.5847</td>
<td>168</td>
<td>0</td>
<td>0.68482</td>
<td>12.3819</td>
</tr>
<tr>
<td>25</td>
<td>10.2566</td>
<td>168</td>
<td>0</td>
<td>0.93848</td>
<td>11.2823</td>
</tr>
<tr>
<td>26</td>
<td>4.2700</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>27</td>
<td>22.2627</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>28</td>
<td>9.5826</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>29</td>
<td>12.2547</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>30</td>
<td>14.2567</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>31</td>
<td>8.2563</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>Obs</td>
<td>Liver_Mt</td>
<td>F8</td>
<td>SOD_Treatment</td>
<td>Gill_SOD</td>
<td>Liver_SOD</td>
</tr>
<tr>
<td>-----</td>
<td>----------</td>
<td>----</td>
<td>---------------</td>
<td>----------</td>
<td>-----------</td>
</tr>
<tr>
<td>32</td>
<td>8.2546</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>33</td>
<td>9.2565</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>34</td>
<td>14.2585</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>35</td>
<td>16.2561</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>36</td>
<td>12.0222</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>37</td>
<td>6.2531</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>

* you may need to rename variables and modify the data set;*

```plaintext
data roberts2; set Roberts;
   rename __Mass_Change = pct_mass_change  Treatment__Hours_ = Treatment_Hours;
drop f8;   /* empty column in the origin spreadsheet  */
proc print;
run;
```

Roberts Excel spreadsheet imported: PRINTED
* you can also EXPORT SAS data sets for use in other packages

* FILE > EXPORT DATA (and select options and browse file locations)

NOTE: D:\baileraj\Classes\Fall 2003\sta402\data\roberts2-data-15oct03.xls was successfully created.

* can save code produced by the EXPORT wizard

```
PROC EXPORT DATA= WORK.ROBERTS2
  OUTFILE= "D:\baileraj\Classes\Fall 2003\sta402\data\roberts2-data-15oct03.xls"
  DBMS=EXCEL2000 REPLACE;
RUN;
```

* check out HELP for IMPORT or EXPORT for more information on these activities