****

**From**

**Helpful SAS tips**

**Jenny Lee**

**January 7, 2016**

Conversions

1 inch = .0254 meters

39.3701 inches = 1 meter My height in meters = 1.57

1 pound = .453592 kilograms

2.20462 pounds = 1 kilogram My weight in kilograms = 45.36

SAS code

1. **To import data**

**proc** **import** datafile="Desktop\cod.xlsx" out=work.cod dbms=xlsx replace; /\*dbms is where you put the file type\*/

**run**;

Data dataset name;

Proc import out=\_*libref.sas dataset*\_\_\_\_\_ [for example, work.cancer]

Datafile= “\_filepath\_\_\_\_\_”

Dbms=excel replace; [whatever file it is, for this example, excel]

Range=”\_\_\_\_\_”

Getnames= yes [or] no;

Usedate=yes;

Sheet= “\_\_\_\_”;

Run;

[There are other commands if necessary]

/\*LIST INPUT SPACE DELIMITED\*/

**DATA** WORK.club1;

INFILE "Y:\Documents\P8483\Spring2013\Data\P8483-Data-SP2013\club1.txt" MISSOVER;

INPUT IdNumber Name $ Team $ StartWeight EndWeight;

**RUN**;

**PROC** **PRINT** DATA = club1;**RUN**;

/\*LIST INPUT COMMA DELIMITED\*/

**DATA** WORK.club1comma;

INFILE "Y:\Documents\P8483\Spring2013\Data\P8483-Data-SP2013\club1comma.txt" DLM= "," MISSOVER;

INPUT IdNumber Name $ Team $ StartWeight EndWeight;

**RUN**;

**PROC** **PRINT** DATA = club1comma;**RUN**;

\*creating a SAS data set from a .XPT file;

LIBNAME ravens "Y:\Documents\P8483\EPIC2011\Data";

\*this libname specifies where the xpt files are located and also ;

LIBNAME xpt XPORT "Y:\Documents\P8483\EPIC2011\XPT\DEMO\_c.xpt";

**DATA** ravens.demo\_c;

SET xpt.demo\_c;

**RUN**;

NOTE: You can also use the drop-down menu, which may be easier.

1. **db**

**proc** **export** data=icd10

outfile="P:\Stafffolders\Jenny\Projects\NCI\exclusion\_icd10\_20161207.xlsx"

dbms=xlsx replace;

**run**;

Below is how you can get the labels to transfer over to SPSS

**proc** **export** data=rcope4

outfile="P:\Study Folders\REACH\Data Management\Data Requests\20180703\_Gaspar\_rcope\rcope.sav"

dbms=spss replace;

FMTLIB=myfmtlib.formats;

**run**;

1. **Libname, content, and print**

LIBNAME epic "Y:\Documents\P8483\EPIC2011\Data\";

**PROC** **CONTENTS** DATA= epic.demo;

**RUN**;

**PROC** **PRINT** DATA= epic.demo;

**RUN**;

**How to order by variable number**

Proc contents data=”File path” order=varnum;

Run;

1. **Merging datasets**

**PROC** **SORT** DATA = epic.startWeight; BY IdNumber; **RUN**;

**PROC** **SORT** DATA = epic.endWeight; BY IdNumber; **RUN**;

\*MERGE the two SAS data sets using a MERGE statement and BY statement in a DATA step;

\*this code merges together data from work.sw and work.ew and outputs the

merged data into a new SAS data set named work.allWt2;

**DATA** work.allWt;

MERGE epic.startWeight epic.endweight;

BY IdNumber;

**RUN**;

\*view the new SAS data set;

**PROC** **PRINT** DATA = work.allWt;

**RUN**;

\*using the "in=" option to only merge observations common to both data sets ;

\*first sort endweightshort - note that startweight doesn't need to be sorted because we sorted that above;

**PROC** **SORT** DATA = epic.endweightshort OUT=work.ewshort; BY idnumber; **RUN**;

**DATA** work.allWt3;

MERGE epic.startweight (in=insw) work.ewshort (in=inew);

BY IdNumber;

if insw and inew;

**RUN**;

\*Modify an existing data set using the SET statement;

**DATA** work.refineAllWt;

SET work.allWt2;

Weightchg = (endWeight - startWeight);

**RUN**;

**proc** **print** data = refineallwt;

**run**;

\*Keep and drop statement;

**DATA** work.demored;

SET epic.demo (keep = id age region rename=(id=idnumber));

**RUN**;

**proc** **print** data = demored;

**run**;

\*drop = ;

**DATA** work.demored;

SET epic.demo (drop = education examdate sex race rename=(id=idnumber));

**RUN**;

**proc** **print** data = demored;

**run**;

**SQL: Left join**

\*Left join appt1 table and clinic tables to add stepclinic id field;

**proc** **sql**;

create table appt2 as

select a.\*, b.StepClinic\_ID

from appt1 a left join clinic b on a.Epic\_Dept\_Num=b.Epic\_Dept\_Num;

**quit**;

1. **Reviewing data**

**DATA** work.child;

SET work.master;

WHERE age < **20** and sbp ^= **.**;

**RUN**;

**proc** **print** data = child;

var age sbp;

**run**;

**proc** **freq** data = child;

table age sbp;

**run**;

**proc means data=child;**

**var age ht wt;**

**run;**

1. **Cleaning data**

\*restrict to participants ge age 20 and remove observations missing height, weight or bp data using a WHERE statement;

**DATA** work.master1;

SET work.master;

WHERE age >= **20** and height ^= **.** and height < **888** and weight ^= **.** and weight < **888** and

sbp ^= **.** and sbp < **888** and dbp ^= **.** and dbp < **888** and bpmeds ^= **.** and bpmeds < **8**;

**RUN**;

Deleting duplicate entries

**proc** **sort** data=icd10 out=newdata nodupkey;

by exclusion exdescriptor;

**quit**;

/\*this combines records with same id and all values in the N data variables (as long as there is only 1 answer for each record\*/

**proc** **sql**;

create table othercond3 as

select nct\_id,

max(ECOG) as ECOG,

max(Karnofsky) as Karnofsky,

max(SWOG) as SWOG,

max(Zubrod) as Zubrod

from othercond2

group by nct\_id;

**quit**;

1. **Formats**
   1. To set length in characters

length AsAd03b\_HrtMedName\_01m $**55** PTSDDCA\_Mth01\_Q4B8 $550.;

1. **Dates and time**

Listed below are some formats for dates.

|  |  |
| --- | --- |
| Format | Displayed Value |
| MMDDYY10. | 04/12/2010 |
| Worddate. | April 12, 2015 |
| Weekdatx. | Monday, 12 April 2010 |
| Monyy | Apr10 |
| Monyy7. | Apr2010 |
| Worddate. | Monday |
| Month. | 4 |
| Year. | 2010 |

Proc freq data=work.dates;

Tables order\_date; [variable name]

Format order\_date mmddyy10.; [variable name format name]

Title “Order dates”;

Run;

***Code for time (via days) elapsed between 2 events***

Data work.dates;

Days = ship\_date - order\_date;

Run;

Proc print data=work.dates;

Var order\_date ship\_date days;

Format order\_date ship\_date mmddyy10.;

Title “number of days between order and ship date”;

Run;

NOTE: To give credit for the “day” when the first of the 2 events occurred, add 1.

Data work.dates;

Day = (ship\_date - order\_date) + 1;

Run;

For more info, visit [http://support.sas.com/documentation/cdl/en/lrcon/65287/HTML/default/viewer.htm#p0g056g35ez8son1sfavozh0lfb3.htm#n1d3z2e1f8u8drn1j70702vf4q3h](http://support.sas.com/documentation/cdl/en/lrcon/65287/HTML/default/viewer.htm#p0g056g35ez8son1sfavozh0lfb3.htm).

***More code for days to***

**data** readmiss1000;

set readmiss1000;

/\*Combined MACE (major acute cardiac event)/ACM variable \*/

/\*(yes/no, plus earliest date of event—that is, date of whichever event comes first)\*/

if **0**<=Date\_ACM-Date\_IndexDC<=**30** then dead30days=**1**;

else dead30days=**0**;

if dead30days=**1** | ACS\_Adjudicated\_Readmission\_1=**1** | ACS\_Adjudicated\_Readmission\_2=**1** | ACS\_Adjudicated\_Readmission\_3=**3** then MACE\_or\_ACM\_30d\_bin=**1**;

else MACE\_or\_ACM\_30d\_bin=**0**;

if dead30days=**1** & MACE\_or\_ACM\_30d\_bin=**1** then do;

if ACS\_Adjudicated\_Readmission\_1=**1** then MACE\_or\_ACM\_30d\_date=min(Date\_ACM,Date\_Readmission\_1);

else if ACS\_Adjudicated\_Readmission\_2=**1** then MACE\_or\_ACM\_30d\_date=min(Date\_ACM,Date\_Readmission\_2);

else if ACS\_Adjudicated\_Readmission\_3=**1** then MACE\_or\_ACM\_30d\_date=min(Date\_ACM,Date\_Readmission\_3);

else MACE\_or\_ACM\_30d\_date=Date\_ACM;

end;

if dead30days=**0** & EDorHosp\_30d\_daysto ne **.** & MACE\_or\_ACM\_30d\_bin=**1** then do; \*Only account for those visits within the first 30 days;

if ACS\_Adjudicated\_Readmission\_1=**1** then MACE\_or\_ACM\_30d\_date=Date\_Readmission\_1;

else if ACS\_Adjudicated\_Readmission\_2=**1** then MACE\_or\_ACM\_30d\_date=Date\_Readmission\_2;

else if ACS\_Adjudicated\_Readmission\_3=**1** then MACE\_or\_ACM\_30d\_date=Date\_Readmission\_3;

end;

format MACE\_or\_ACM\_30d\_date MMDDYY10.;

/\*Days to first MACE/ACM event within 30 days\*/

if MACE\_or\_ACM\_30d\_bin = **1** then

MACE\_or\_ACM\_30d\_daysto = MACE\_or\_ACM\_30d\_date-Date\_IndexDC;

***Code to reformat dates and time (from FM)***

IF CreationDate NE "" THEN DO;

CreationDate\_Yr = Substr(CreationDate,**1**,**4**);

CreationDate\_Mth = Substr(CreationDate,**6**,**2**);

CreationDate\_dy = Substr(CreationDate,**9**,**2**);

CreationDate2 =mdy(CreationDate\_Mth,CreationDate\_dy,CreationDate\_Yr);

END;

IF CHT\_B04\_ED\_Time NE "" THEN DO;

CHT\_B04\_ED\_Time\_HR = Substr(CHT\_B04\_ED\_Time,**1**,**2**);

CHT\_B04\_ED\_Time\_MIN = Substr(CHT\_B04\_ED\_Time,**4**,**2**);

CHT\_B04\_ED\_Time\_SEC = Substr(CHT\_B04\_ED\_Time,**7**,**2**);

CHT\_B04\_ED\_Time2 =HMS(CHT\_B04\_ED\_Time\_HR,CHT\_B04\_ED\_Time\_MIN,CHT\_B04\_ED\_Time\_SEC);

END;

***[To add format]***

format CHT\_B04\_ED\_Date2 CreationDate2 ModificationDate2 mmddyy10. CHT\_B04\_ED\_Time2 CreationTime2 ModificationTime2 TIME10. ;

1. **Recodes**

**data** work.master1;

set work.master;

WHERE age >= **20** and height ^= **.** and height < **888** and weight ^= **.** and weight < **888** and

sbp ^= **.** and sbp < **888** and dbp ^= **.** and dbp < **888** and bpmeds ^= **.** and bpmeds < **8**;

\*create a constant;

constant10 = **10**; \*the variable named "constant10" will have a value of 10 for all observations;

\*addition;

sbp2 = sbp + **2**;

dbp2 = dbp + **1**;

\*create bmi variable;

\*first need to convert height from cm to m using division;

heightm = height/**100**;

\*now create bmi;

bmi = weight/(heightm\*\***2**);

\*alt approach to BMI;

bmi2 = weight/((height/**100**)\*\***2**);

\*note that \*\* raises a variable to the specified power;

\*calculates bmi by using multiplication in the denominator;

bmi3 = weight/(heightm\*heightm);

\*ex, raise a variable to different powers;

power3 = constant10\*\***3**;

power4 = constant10\*\***4**;

\*logarithm;

log10v1 = log10(**100**);

log10bmi = log10(bmi);

\*ROUND statement;

bmir = ROUND(bmi,**1**);

bmir1 = ROUND(bmi,**.1**);

bmir2 = ROUND (bmi,**.01**);

bmir3 = ROUND(bmi,**10**);

\*RENAME statement is used to rename variables;

RENAME subscapSF = SSSF; \*existing variable name = new variable name;

\*subtraction - pulse pressure;

pulsepressure = sbp - dbp;

**RUN**;

**proc** **print** data = work.master1;

**run**;

**proc** **contents** data = work.master1;**run**;

array

**data** keyvars1;

set keyvars;

missing M N;

array life [\*]

LISMOKE\_01M LIDRINK\_01M LIANXITY\_01M LICLINIC\_01M LICOUNCL\_01M LIDIET\_01M LIPHYCAL\_01M LIREHAB\_01M LIAVG\_01M SF\_PCS\_01M SF\_MCS\_01M

LISMOKE\_06M LIDRINK\_06M LIANXITY\_06M LICLINIC\_06M LICOUNCL\_06M LIDIET\_06M LIPHYCAL\_06M LIREHAB\_06M LIAVG\_06M SF\_PCS\_06M SF\_MCS\_06M

LISMOKE\_12M LIDRINK\_12M LIANXITY\_12M LICLINIC\_12M LICOUNCL\_12M LIDIET\_12M LIPHYCAL\_12M LIREHAB\_12M LIAVG\_12M SF\_PCS\_12M SF\_MCS\_12M

Education\_years SEX ETHNIC RACE BLACK Partner\_Status NOT\_US\_BORN

ENGLISH ENGLISH\_FLUENT PCL\_C\_3TO7 PCL\_C\_01M\_;

do i = **1** to dim(life);

if life [i]= -**1** then life [i] = **.M**;

if life [i] = -**2** then life [i] = **.N**;

end;

**run**;

Converting numeric variables in character

Creating scales

**1st method (sum will show up as missing if there is a missing value):** ASI\_physical\_01m = ASI3\_Mth01\_Q04 + ASI3\_Mth01\_Q12 + ASI3\_Mth01\_Q08 + ASI3\_Mth01\_Q07 + ASI3\_Mth01\_Q15 + ASI3\_Mth01\_Q03;

OR

if nmiss(of IPCS03\_FindOutConcern--IPCS12\_TreatEqual IPCS01\_SpeakTooFast\_R--IPCS18\_NegativeAtt\_R) =**0** then IPC\_Sum\_im =

IPCS01\_SpeakTooFast\_R + IPCS02\_HardUnderstd\_R + IPCS03\_FindOutConcern + IPCS04\_SayImportant +

IPCS05\_TakeSeriously + IPCS06\_ExplainTestResults + IPCS07\_ExplainPEResults + IPCS08\_TxPlanToget +

IPCS09\_AskHelpTx + IPCS10\_ShowConcern + IPCS11\_Respect + IPCS12\_TreatEqual + IPCS13\_RaceLessAttent\_R +

IPCS14\_RaceDiscrimate\_R + IPCS15\_BehaveRudely\_R + IPCS16\_TalkDown\_R + IPCS17\_HardTime\_R + IPCS18\_NegativeAtt\_R;

**2nd method (this will generate sum even with missing values):** if nmiss(of PCL5\_01\_Memories\_01m--PCL5\_20\_Sleep\_01m)=**0** then PCL5\_Sum\_1m\_im=sum(of PCL5\_01\_Memories\_01m--PCL5\_20\_Sleep\_01m);

**Or variablename=sum(of test1 test2 test3);**

**Or variablename=sum(1,2,3)**

**CBCH hw**

Question 2. Compute bmi in kg/m squared;

**data** job.data2;

set job.data;

heightm=height/**100**;

bmi=(weight/heightm\*\***2**);

**run**;

title 'Comparing conversion of bmi';

**proc** **print** data=job.data2 (obs=**5**);

var height heightm weight bmi;

**run**;

\*Question 3. Score LOT scale.;

**data** job.data3;

set job.data2;

array forward [**5**] lot01\_bl lot04\_bl lot05\_bl lot10\_bl lot11\_bl;

do i=**1** to **5**;

if forward [i]=-**1** then forward [i]=**.**;

if forward [i]=**1** then forward [i]=**0**;

if forward [i]=**2** then forward [i]=**1**;

if forward [i]=**3** then forward [i]=**2**;

if forward [i]=**4** then forward [i]=**3**;

if forward [i]=**5** then forward [i]=**4**;

end; drop i;

array reverse [**4**] lot03\_bl lot08\_bl lot09\_bl lot12\_bl;

do i=**1** to **4**;

if reverse [i]=-**1** then reverse [i]=**.**;

if reverse [i]=**1** then reverse [i]=**4**;

if reverse [i]=**2** then reverse [i]=**3**;

if reverse [i]=**3** then reverse [i]=**2**;

if reverse [i]=**4** then reverse [i]=**1**;

if reverse [i]=**5** then reverse [i]=**0**;

end; drop i;

**run**;

title 'Comparing forward array';

**proc** **print** data=job.data3 (obs=**5**);

var newID lot01\_bl lot04\_bl lot05\_bl lot10\_bl lot11\_bl;

**run**;

title 'Comparing reverse array';

**proc** **print** data=job.data3 (obs=**5**);

var newID lot03\_bl lot08\_bl lot09\_bl lot12\_bl;

**run**;

**data** job.data4;

set job.data3;

lotscore= sum(lot01\_bl,lot03\_bl,lot04\_bl,lot05\_bl,lot08\_bl,lot09\_bl,lot11\_bl,lot12\_bl);

**run**;

title 'Total LOT score';

**proc** **print** data=job.data4 (obs=**5**);

var lot01\_bl lot03\_bl lot04\_bl lot05\_bl lot08\_bl lot09\_bl lot11\_bl lot12\_bl lotscore;

**run**;

1. **Proc print**

In csv file

ods csv file="P:\Study Folders\REACH\Data Management\1000 ID Cleaning\REACH Email Code\Missing Variable Codes\CSV Files/Missing\_Phase2.csv";

**proc** **print** data=missing\_p2;

var ID Phase\_2\_Missing;

**run**;

ods csv close;

1. **Descriptive statistics**

Histogram

Proc univariate data=*title;*

Id *id*;

var *variable\_name;*

histogram / normal;

run;

Proc freq

title 'Frequencies of newid, date\_bl, group, race, ethnicity, gender and all LOT variables';

**proc** **freq** data=job.data;

tables newid date\_bl group race ethnic gender lot01\_bl lot02\_bl lot03\_bl lot04\_bl lot05\_bl lot06\_bl lot07\_bl lot08\_bl lot09\_bl lot10\_bl lot11\_bl lot12\_bl;

**run**;

Proc means

title 'mean median of age, height and weight';

**proc** **means** data=job.data n mean median min max std;

var age height weight;

**run**;

\*Question 4. Run descriptive statistics on variables of interest.;

Chi-squares

title 'Demographics';

**proc** **freq** data=job.data4;

tables group\* (race ethnic gender) / chisq expected fisher cl;

**run**;

T-tests

title 'Age';

**proc** **ttest** data=job.data4;

class group;

var age;

**run**;

title 'BMI and LOT score';

**proc** **ttest** data=job.data4;

class group;

var bmi lotscore;

**run**;

title 'mean median of age bmi lotscore by group';

**proc** **means** data=job.data4 n mean median min max std;

class group;

var age bmi lotscore;

**run**;

/\*I decided to also create categories for bmi and lot scores since this information may be more meaningful than just looking at the means. \*/

**data** job.data5;

set job.data4;

/\*BMI categories: normal weight, overweight and obese\*/

If bmi ge **0** and bmi le **24.9** then bmicat=**0**; /\*normal\*/

Else if bmi gt **24.9** and bmi le **29.9** then bmicat=**1**; /\*overweight\*/

Else if bmi gt **29.9** then bmicat=**2**; /\*obese\*/

/\*categories for pessimistic, neutral and optimistic\*/

If lotscore ge **0** and lotscore lt **17** then lotcat=**0**; /\*pessimistic\*/

else if lotscore = **17** then lotcat=**1**; /\*neutral\*/

Else if lotscore gt **17** and lotscore le **32** then lotcat=**2**; /\*optimistic\*/

**Run**;

**proc** **print** data=job.data5 (obs=**5**);

var newid bmi bmicat lotscore lotcat;

**run**;

title 'BMI and lotscore groups';

**proc** **freq** data=job.data5;

tables group\* (bmicat lotcat) / chisq expected fisher cl;

**run**;

1. **Regression**

**Logistic regression**

/\*logistic regression of seeing a dr (LICLINIC\_01M)\*/

title "See a dr at 1 month logistic regression";

ods graphics on;

**proc** **logistic** data=keyvars2;

class sex\_re (ref="0") hisp\_race (ref="0") blackaa (ref="0") other (ref="0") / param=ref;;

model LICLINIC\_01M (event="1") = age age sex\_re SF\_PCS\_01M SF\_MCS\_01M grace\_ hisp\_race blackaa other / clodds=pl;

**run**;

**quit**;

**Linear regression**

/\*regression using lifestyle\_01m with race/ethnicity- use this! with insurance and \*/

title "Regression lifestyle score at 1 month by race/ethnicity and covariates";

ods graphics on;

**proc** **reg** data=keyvars2

plots(label only)= (cooksd rstudentbypredicted dffits dfbetas);

id obs;

model lifescore\_01m = age sex\_re SF\_PCS\_01M SF\_MCS\_01M grace\_ /\*Education\_years\*/ hisp\_race blackaa other / stb clb /\*stb tol vif influence\*/;

output out=reg\_lifesytle\_01m

p=yhati r=ei h=hati rstudent=jacki student=studentizedi;

**run**;

**quit**;

title;

1. **Transpose data**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Transposition 1 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Transpose the readmission dates for the first thirty days\*/

**proc** **sort** data=hosp\_30; by id Hosp\_03a\_AdmissionDate; **run**;

**proc** **transpose** data=hosp\_30

(Rename=(Hosp\_03a\_AdmissionDate=Admin1))

PREFIX=Date\_Readmission\_

out=hospital\_trans;

VAR Admin1;

BY ID;

**run**;

/\*drop unnecessary variable\*/

**data** hospital\_trans1;

set hospital\_trans;

drop \_Name\_ \_label\_;

**run**;

**proc** **freq** data=hospital\_trans1;

table Date\_Readmission\_1 Date\_Readmission\_2 Date\_Readmission\_3 / list missing;

**run**;

/\* Merge with overall dataset\*/

**proc** **sort** data=hospital\_trans1;by id;**run**;

**proc** **sort** data=keyvars1000;by id;**run**;

**data** keyvars1000;

merge keyvars1000 (in=a) hospital\_trans1;

by id;

if a;

**run**;

1. **Proc compare**

ods excel file="&saveto\chronic\_cond\_brief.xlsx";

**proc** **compare** base=study\_a1 compare=study\_b1 brief;

by nct\_id;

id nct\_id;

**run**;

ods excel close;

**OR**

ods excel file="P:\Staff Folders\David T\Reach QA\Output Files\H5Patch\_G5Patch.xlsx"

options(

sheet\_name="Compare"

sheet\_interval="none"

);

**proc** **compare** base = newestPatch compare = OldPatch;

Id ID;

**run**;

ods excel close;

1. **How to merge count variable to master dataset**
   1. Run a proc freq and output new dataset (out= “data file name”)
   2. Then merge with master dataset

proc freq;

tables A A\*B / out=D;

run;

data final;

merge original D;

by id;

run;

1. **Labels**

**DATA auto2;**

**SET auto;**

**LABEL rep78 ="1978 Repair Record"**

**mpg ="Miles Per Gallon"**

**foreign="Where Car Was Made";**

**RUN;**

1. **Computing age**

/\*Compute age\*/

age=yrdif(DOB, enrollment\_date, 'AGE');

if dob = **.** then age = yrdif(DOB\_self, enrollment\_date, 'AGE');

age2 = floor (age);

1. **Generate sequence number**

Data out;

Set in;

Seqno = \_N\_;

Run;

Other option: <https://www.listendata.com/2015/10/retain-statement-in-sas.html>

*data aaa;  
set abcd;  
retain z 0;  
z = z + 1;  
run;*

1. **Parsing string variable into 2 variables (STRIP function)**

**Example 1:**

data split;

set test;

length var1-var3 $10.;

array var(3) $;

do i = 1 to dim(var);

var[i]=scan(row,i,'|');

end;

run;

**Example 2**

data middle;

set have ;

length term $10 tag $3 ;

do i=1 by 1 until (term=' ');

term=strip(scan(list,i,','));

tag=scan(term,2,'[]');

if term ne ' ' then output;

end;

run;

1. Input dates

Visit\_date=mdy(months,day,year);

format Visit\_date mmddyy10.;

**SQL**

1. Merging multiple datasets

/\*Merge ALL PHS SAQ7 Data\*/

**Proc** **sql**;

create table SAQ7\_ALL\_PHSa as

Select distinct A.ID,B.\*, C.\*, D.\*

from Phs\_ids as A Left Join SAQ\_P2a as B

on A.id=b.id

Left Join SAQ\_1moa as C

on A.id=C.id

Left Join SAQ\_12moa as D

on A.id=D.id;

**quit**;/\*N=903\*/

1. Other sample code

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* PHS Status Data Cleaning \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Created by Jenny Lee on 7/14/2023 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*Step 1: Creating datasets\*/

libname r "P:\Study Folders\REACH\Sean\SAS\Production";

libname i "P:\Study Folders\Edmondson\_PHS\_2015\Data Management\Key Vars Prep\IDs";

/\*Macro to set the multiple chart tables\*/

options mprint; /\*This option prints how the %MACRO is processing the in the LOG window\*/

**data** table\_id;

input table $12.;

CARDS;

visit\_mth01

visit\_mth06

visit\_mth12

visit\_bl

;

**run**;

**%MACRO** SETDATA();

%do i = **1** %TO **4**;

data \_null\_;

set table\_id (firstobs=&i obs=&i);

call symput ("name", table);

run;

data &name. (drop= EA\_Createdby

EA\_Modifiedby

EA\_CreationDate

EA\_ModificationDate

EA\_CreationTime

EA\_ModificationTime

EA\_Unique\_Id);

set r.&name.;

run;

proc contents data=&name. order=varnum; run;

%END;

**%MEND**;

%***SETDATA***();

/\*PHS ids dataset\*/

**data** phs\_id;

set i.phs\_ids;

**run**;

**proc** **sql**;

create table visit\_all as

select distinct a.ID, b.ID, b.Calc\_studystatus, b.Enrollment\_Date, b.Phase2\_ActualDate,

b.phase2\_fu\_status, b.WithdrawalDate, c.ID, c.Month1\_FU\_Status, c.Month1\_Date,

d.ID, d.Month6\_FU\_Status, d.Month6\_Date, e.ID, e.Month12\_FU\_Status, e.Month12\_Date

from phs\_id as A Left Join visit\_bl as B

on A.ID=B.ID

Left Join visit\_mth01 as C

on A.ID=C.ID

Left Join visit\_mth06 as D

on A.ID=D.ID

Left Join visit\_mth12 as E

on A.ID=E.ID;

**quit**;

/\*Step 2: QA\*/

**proc** **freq** data=visit\_all;

table phase2\_fu\_status Month1\_FU\_Status Month6\_FU\_Status Month12\_FU\_Status / list missing;

**run**;

**proc** **print** data=visit\_all;

where ((phase2\_fu\_status="Phase 2 Completed" or phase2\_fu\_status="Phase 2 partially completed Filemaker" or

phase2\_fu\_status="Phase 2 partially completed filemaker") and Phase2\_ActualDate=**.**) or

((Month1\_FU\_Status="1 month Completed" or Month1\_FU\_Status="1 month completed" or

Month1\_FU\_Status="1 month partially completed Filemaker")and Month1\_Date=**.**) or

((Month6\_FU\_Status="6 month Completed" or Month6\_FU\_Status="6 month completed" or

Month6\_FU\_Status="6 month partially completed Filemaker" or Month6\_FU\_Status="6 month partially completed filemaker")

and Month6\_Date=**.**) or ((Month12\_FU\_Status="12 month Completed" or Month12\_FU\_Status="12 month completed" or

Month12\_FU\_Status="12 month partially completed Filemaker" or Month12\_FU\_Status="12 month partially completed qualtrics")

and Month12\_Date=**.**);

**run**;

\*Only 2 obs missing phase2 date: id 60359 and 60388. JL fixed in FM using contact log notes. 7/14/2023;