

```

/*#####
MATH 449 Statistical Data Management
Groupwork 8
performed by Inna Williams
#####
*/
/*1. Below we have a file containing family id, father's
name and income. We also have a file containing income
information for three years. Write a SAS code that
merge the files together so we have the dads observation
on the same line with the faminc observation based on
the key variable famid. */
data dads;
    input famid name $ income ;
cards;
2 Smith 22000
1 Bill 30000
3 Paul 25000
;
run;

data faminc;
    input famid faminc12 faminc13 faminc14 ;
cards;
3 75000 76000 77000
1 40000 40500 41000
2 45000 45400 45800
;
run;

PROC SQL;
TITLE "Dads and Famic Merged using proc sql";
CREATE TABLE Merged AS
SELECT D.famid, D.name, D.income,
       F.faminc12, F.faminc13, F.faminc14
from dads D
FULL JOIN faminc F ON F.famid=D.famid;
QUIT;

proc print data=Merged;
run;

```

Dads and Famic Merged using proc sql

Obs	famid	name	income	faminc12	faminc13	faminc14
1	1	Bill	30000	40000	40500	41000
2	2	Smith	22000	45000	45400	45800
3	3	Paul	25000	75000	76000	77000

```

/* Using SAS code with data step*/
proc sort data=dads;
by famid;
run;
proc sort data=faminc;
by famid;
run;

data combined;
merge dads faminc;
run;

proc print data=combined noobs;
TITLE "Dads and Famic Merged using data step";
run;

```

Dads and Famic Merged using data step

famid	name	income	faminc12	faminc13	faminc14
1	Bill	30000	40000	40500	41000
2	Smith	22000	45000	45400	45800
3	Paul	25000	75000	76000	77000

/*2. After merging the files together, change the names of the variables faminc12 , faminc13 and faminc14 to totalinc2012, totalinc2013 and totalinc2014, respectively.*/

```

PROC SQL;
TITLE1 "change the names of the variables";
TITLE2 "faminc12 , faminc13 and faminc14 to";
TITLE3 "totalinc2012, totalinc2013 and totalinc2014";
CREATE TABLE Merged AS
SELECT  faminc12 AS totalinc2012,
        faminc13 AS totalinc2013,
        faminc14 AS totalinc2014
FROM    Merged;
QUIT;
run;
proc print data =Merged ;
run;

```

change the names of the variables
faminc12 , faminc13 and faminc14 to
totalinc2012, totalinc2013 and totalinc2014

Obs	totalinc2012	totalinc2013	totalinc2014
1	40000	40500	41000
2	45000	45400	45800
3	75000	76000	77000

/*3. If we had a file with kids where a dad could have more than one kid. Matching up the "dads" with the "kids" is called a "one-to-many" merge since you are matching one dad observation to possibly many kids records. The dads and kids records are shown below. Notice here we have variable fid in the first data set and famid in the second. These are the variables that we want to match. When we merge the two using proc sql, we don't have to rename them, since we can use data set name identifier. */

```
DATA dads;
  INPUT famid name $ income ;
CARDS;
2 Smith 22000
1 Bill 30000
3 Paul 25000
;
RUN;
```

```
DATA kids;
INPUT fid kidname $ birth age wt sex $ ;
CARDS;
1 Beth 1 9 60 f
1 Bob 2 6 40 m
1 Barb 3 3 20 f
2 Andy 1 8 80 m
2 Al 2 6 50 m
2 Ann 3 2 20 f
3 Pete 1 6 60 m
3 Pam 2 4 40 f
3 Phil 3 2 20 m
; RUN;
```

```
PROC SQL;
TITLE "Match Dads and Kids";
CREATE TABLE MergedDadsAndKids AS
SELECT D.famid, D.name, D.income,
K.kidname, K.birth, K.age, K.wt, K.sex
from dads D
INNER JOIN kids K ON K.fid=D.famid;
QUIT;
```

```
PROC PRINT DATA=MergedDadsAndKids;
RUN;
```

Match Dads and Kids

Obs	famid	name	income	kidname	birth	age	wt	sex
1	1	Bill	30000	Beth	1	9	60	f
2	1	Bill	30000	Bob	2	6	40	m
3	1	Bill	30000	Barb	3	3	20	f
4	2	Smith	22000	Andy	1	8	80	m
5	2	Smith	22000	Al	2	6	50	m
6	2	Smith	22000	Ann	3	2	20	f
7	3	Paul	25000	Pete	1	6	60	m
8	3	Paul	25000	Pam	2	4	40	f
9	3	Paul	25000	Phil	3	2	20	m

```
/*4. title3 'Use Aliases, Calculated Columns and CASE Expressions';
options pageno=1;*/
```

```
DATA clinical;
INPUT patient $ gender $ asian $ wtLb htIn;
DATALINES;
Jack M no 205 66
Jock M yes 198 71
Jane F no 143 68
Joe M no 167 68
Jenny F no 98 65
Jackson M yes 221 65
Horton M no 314 70
Jill F yes 121 63
;
```

```
PROC SQL;
TITLE "Use Aliases, Calculated Columns and CASE Expressions";
select patient ,
       wtLb as Weight_lb,
       htIn 'Height (in.)',
       weight_lb*703/htIn**2
       as BMI format=4.1,
       case asian
         when 'yes' then calculated BMI / 23
         else calculated BMI / 25
       end

       as BMIP label='BMI Prime' format=4.2,
       case
         when calculated bmip LT 0.66 then 'Severely underweight'
         when calculated bmip LT 0.74 then 'Underweight'
         when calculated bmip LT 1.00 then 'Normal'
         when calculated bmip LT 1.21 then 'Overweight'
         when calculated bmip LT 1.41 then 'Obese Class I'
         when calculated bmip LT 1.60 then 'Obese Class II'
         else 'Obese Class III'
       end
       label='Weight Category',
       case
         when asian='yes' then 'Asian'
         else ''
       end
       label='Remark'
from clinical;
QUIT;
```

Use Aliases, Calculated Columns and CASE Expressions

patient	Weight_lb	Height (in.)	BMI	BMI Prime	Weight Category	Remark
Jack	205	66	33.1	1.32	Obese Class I	
Jock	198	71	27.6	1.20	Overweight	Asian
Jane	143	68	21.7	0.87	Normal	
Joe	167	68	25.4	1.02	Overweight	
Jenny	98	65	16.3	0.65	Severely underweight	
Jackson	221	65	36.8	1.60	Obese Class II	Asian
Horton	314	70	45.0	1.80	Obese Class III	
Jill	121	63	21.4	0.93	Normal	Asian