**In-Class Exercise #3**

**Merging datasets using SAS**

**Tasks**

1. Learn how to merge datasets using the merge statement
2. Know how to handle duplicate observations
3. Learn how to merge a dataset with a output from a proc step
4. Understand the basics of PROC SQL and know how to merge data sets using PROC SQL

**1. Merging datasets using the merge statement**

**sasprogram3.sas**

|  |
| --- |
| \* Sample SAS program showing how to merge two different datasets;  OPTIONS NOCENTER;  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;  \* one-to-one merge;  DATA dataset1;  INPUT id $ 1-8 score1 9-11 testtime $ 13-17;  CARDS;  R0000001 90 10:00  R0000002 80 10:00  R0000003 60 12:00  R0000004 70 10:00  R0000005100 12:00  R0000006 75 18:00  R0000007 80 18:00  R0000008 40 10:00  R0000009 55 18:00  R0000010 45 12:00  ;  TITLE 'Printout out of first test scores';  PROC PRINT DATA=dataset1; RUN;  DATA dataset2;  INPUT id $ 1-8 score2 9-11 testtime $ 13-17;  CARDS;  R0000005100 12:00  R0000001 75 10:00  R0000003 50 12:00  R0000004 70 10:00  R0000007 90 18:00  R0000006 70 18:00  R0000002 95 10:00  R0000010 . 12:00  R0000009 . 18:00  R0000008 . 10:00  ;  TITLE 'Printout out of second scores';  PROC PRINT DATA=dataset2; RUN;  PROC SORT DATA=dataset1; by ID;  PROC SORT DATA=dataset2; by ID;  \* one-to-one merge;  DATA datamerge12;  MERGE dataset1 dataset2; by id;  avg = (score1 + score2) / 2;  \*same result as above using proc sql;  /\*proc sql;  create table datamerge12 as  select a.\*, b.score2, (score1+score2)/2 as avg from dataset1 as a left join dataset2 as b on a.id = b.id;  quit;\*/  \* Delete students that dropped the class;  DATA datamerge12; SET datamerge12;  IF score2 eq . then delete;  /\*without VAR, by default it will show all variables\*/  TITLE 'Printout out of both test scores and average after one-to-one merge';  PROC PRINT DATA=datamerge12; VAR id testtime score1 score2 avg; RUN;  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;  \* one-to-many (or many-to-one) merge;  DATA dataset3;  INPUT testtime $ 1-5 curve 6-7;  CARDS;  10:00 3  12:00 2  14:00 4  16:00 1  ;  PROC SORT DATA=datamerge12; by testtime;  PROC SORT DATA=dataset3; by testtime;  TITLE 'Regular merge (no conditions given)';  DATA datamerge123;  MERGE datamerge12 dataset3; by testtime;  PROC PRINT DATA=datamerge123; VAR id testtime score1 score2 avg curve; RUN;  TITLE 'Inner join merge (must be in both datasets)';  DATA datamerge123;  MERGE datamerge12 (in=a) dataset3 (in=b); by testtime; if a and b;  PROC PRINT DATA=datamerge123; VAR id testtime score1 score2 avg curve; RUN;  TITLE 'Outer full join merge (same as regular merge - could be in either dataset)';  DATA datamerge123;  MERGE datamerge12 (in=a) dataset3 (in=b); by testtime; if a or b;  PROC PRINT DATA=datamerge123; VAR id testtime score1 score2 avg curve; RUN;  TITLE 'Outer left join merge (must be in the left dataset)';  DATA datamerge123;  MERGE datamerge12 (in=a) dataset3 (in=b); by testtime; if a;  PROC PRINT DATA=datamerge123; VAR id testtime score1 score2 avg curve; RUN;  TITLE 'Outer right join merge (must be in the right dataset)';  DATA datamerge123;  MERGE datamerge12 (in=a) dataset3 (in=b); by testtime; if b;  PROC PRINT DATA=datamerge123; VAR id testtime score1 score2 avg curve; RUN; |

**2. Duplicate observations**

**sasprogram3.sas - continued**

|  |
| --- |
| \* problems with duplicate data when merging;  DATA dataset3a;  INPUT testtime $ 1-5 curve 6-7;  CARDS;  10:00 3  12:00 2  14:00 4  16:00 1  18:00 5  10:00 0  12:00 2  ;  TITLE 'Print out all entries';  PROC SORT DATA=dataset3a; by testtime;  PROC PRINT DATA=dataset3a; RUN;  /\*빈도분석 결과를 하나의 데이터셋으로 저장\*/  TITLE 'Identify dupicate entries';  PROC FREQ DATA=dataset3a;  TABLES testtime / NOPRINT OUT=duplist;  PROC PRINT DATA=duplist; WHERE COUNT ge 2; RUN;  /\*nodup 과 nodupkey의 차이점\*/  TITLE 'Eliminating exact duplicates using nodup';  PROC SORT DATA=dataset3a nodup out=dataset3a\_nodup; by testtime;  PROC PRINT DATA=dataset3a\_nodup; RUN;  TITLE 'Eliminating duplicates based on the by variable using nodupkey';  PROC SORT DATA=dataset3a nodupkey out=dataset3a\_nodupkey; by testtime;  PROC PRINT DATA=dataset3a\_nodupkey; RUN;  \* for details, see http://www2.sas.com/proceedings/forum2007/069-2007.pdf; |

**3. Merging a dataset with output from a proc step**

**sasprogram3.sas - continued**

|  |
| --- |
| \* merging with output from a proc statement;  PROC SORT data=datamerge12; BY testtime;  TITLE 'Calculate means by testtime';  PROC MEANS data=datamerge12; BY testtime; VAR score1 score2 avg;  OUTPUT OUT=dataset4 MEAN = score1mean score2mean avgmean; RUN;  PROC SORT DATA=datamerge12; by testtime;  PROC SORT DATA=dataset4; by testtime;  DATA datamerge124;  MERGE datamerge12 dataset4; by testtime;  newcurve = 85 - avgmean;  curvedavg = avg + newcurve;  PROC SORT DATA=datamerge124; BY ID;  TITLE 'Print curved average score';  PROC PRINT DATA=datamerge124; VAR id testtime score1 score2 avg newcurve curvedavg; RUN; |

**4. The basics of PROC SQL and merging data sets using PROC SQL**

**sasprogram3.sas – continued**

|  |
| --- |
| \*merging datasets using proc sql. Much shorter and effective code to write;  TITLE 'Merging dataset1 and dataset2 in proc sql';  PROC SQL;  CREATE TABLE sql\_merged12 AS  SELECT \*  FROM dataset1 AS A INNER JOIN (SELECT \* FROM dataset2 WHERE score2 IS NOT MISSING) as B  ON A.ID=B.ID  ORDER BY ID;  QUIT;  PROC PRINT DATA=sql\_merged12; run;  DATA sql\_merged12; SET sql\_merged12;  avg = (score1 + score2) / 2;  TITLE 'Merging sql\_merged12 and dataset3 in proc sql';  PROC SQL;  CREATE TABLE sql\_merged123 AS  SELECT \*  FROM sql\_merged12 AS A LEFT JOIN dataset3 as B  ON A.testtime=B.testtime  ORDER BY ID;  QUIT;  PROC PRINT DATA=sql\_merged123; run;  TITLE 'Calculating mean values and merging in proc sql';  PROC SQL;  CREATE TABLE sql\_merged124 AS  SELECT \*  , MEAN(score1) AS score1mean  , MEAN(score2) AS score2mean  , MEAN(avg) AS avgmean  FROM sql\_merged12  GROUP BY TESTTIME  ORDER BY ID;  QUIT;  DATA sql\_merged124; SET sql\_merged124;  newcurve = 85 - avgmean;  curvedavg = avg + newcurve;  PROC PRINT DATA=sql\_merged124; run;  \* for some help on proc sql, see:  http://www2.sas.com/proceedings/sugi27/p070-27.pdf  http://www2.sas.com/proceedings/sugi29/042-29.pdf; |

#libname libname ‘C:\Users\samsung\Downloads’;

Proc sql;

Create table compu as

Select gvkey, at, ni, prcc\_f, avg(ni) as ni\_avg from libname.FSdatas

Group by gvkey;

Quit;