Two Methods to Merge Data onto Every Observation in Another Dataset

Lisa Mendez, Ph.D., Knowesis, Inc., San Antonio , TX
Kim Brunnert, Ph.D., Elsevier, Houston, TX

ABSTRACT

There are times when you just can't seem to find a PROC that will do exactly what you want. We came across a scenario where we needed to calculate the mean of a student data file and then flag student observations that were more than three standard deviations from the mean. We came across two methods to do what we needed to do. One method uses a combination of Data steps and Procs, and utilizes the *If _N_ then set* method. Another method utilizes Proc SQL. This paper will outline both methods step-by-step and illustrates two different ways to do the same thing. Personal preference dictates which method to use.

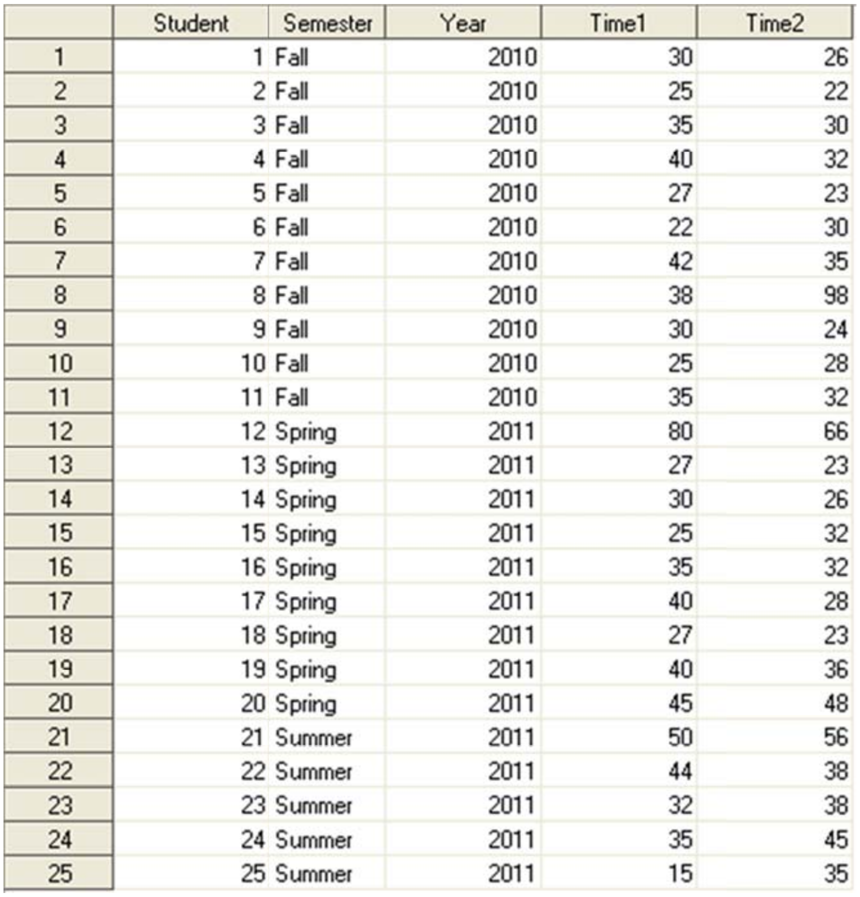
INTRODUCTION

We have a student data set that identifies students by ID number across two years. Each student has two time values (Time1 & Time2). We want to calculate the mean of all of the students' time values and flag those students whose time values fall above or below three standard deviations from the mean.

RAW DATA

The raw data set has five variables and 25 observations. The variables are: Student, Semester, Year, Time1, and Time2. Each student has a time value, which represents time in seconds.

Figure 1. Raw Data.

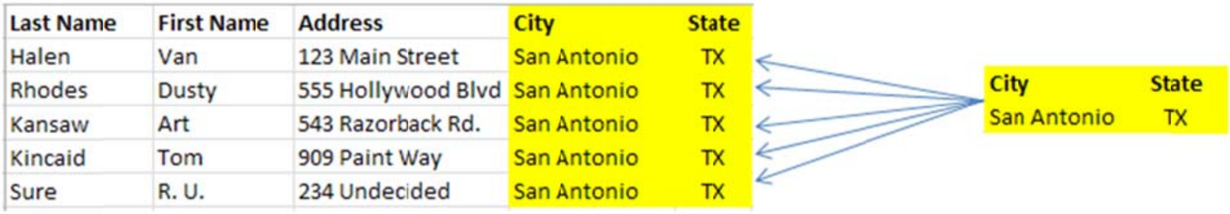


	Student	Semester	Year	Time1	Time2
1	1	Fall	2010	30	26
2	2	Fall	2010	25	22
3	3	Fall	2010	35	30
4	4	Fall	2010	40	32
5	5	Fall	2010	27	23
6	6	Fall	2010	22	30
7	7	Fall	2010	42	35
8	8	Fall	2010	38	98
9	9	Fall	2010	30	24
10	10	Fall	2010	25	28
11	11	Fall	2010	35	32
12	12	Spring	2011	80	66
13	13	Spring	2011	27	23
14	14	Spring	2011	30	26
15	15	Spring	2011	25	32
16	16	Spring	2011	35	32
17	17	Spring	2011	40	28
18	18	Spring	2011	27	23
19	19	Spring	2011	40	36
20	20	Spring	2011	45	48
21	21	Summer	2011	50	56
22	22	Summer	2011	44	38
23	23	Summer	2011	32	38
24	24	Summer	2011	35	45
25	25	Summer	2011	15	35

USING THE *IF _N_=1* SET METHOD

Essentially, the If _N_=1 Set method sets the values of a one-record dataset to all other observations of another dataset.

Figure 2. Illustration of how one observation is set to every observation in another dataset.

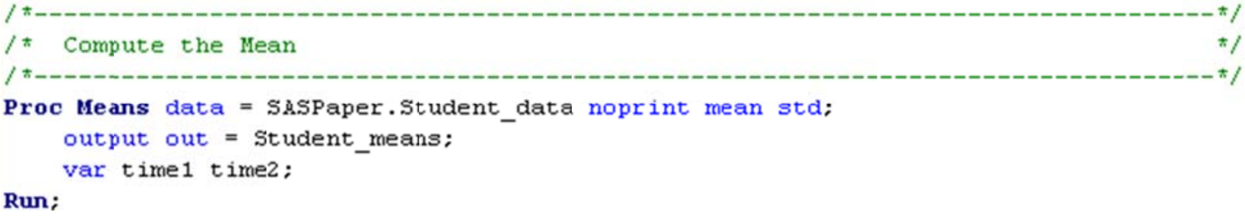


Last Name	First Name	Address	City	State
Halen	Van	123 Main Street	San Antonio	TX
Rhodes	Dusty	555 Hollywood Blvd	San Antonio	TX
Kansaw	Art	543 Razorback Rd.	San Antonio	TX
Kincaid	Tom	909 Paint Way	San Antonio	TX
Sure	R. U.	234 Undecided	San Antonio	TX

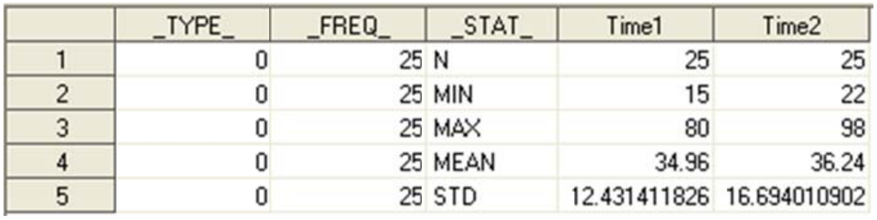
City	State
San Antonio	TX

To create the one-record dataset that we will use, we need to start by calculating the means of the student dataset. We want to calculate the mean for both the Time1 and Time2 variables. We will use the *output out* = statement to create an output dataset.

Sample SAS Code and Output Dataset



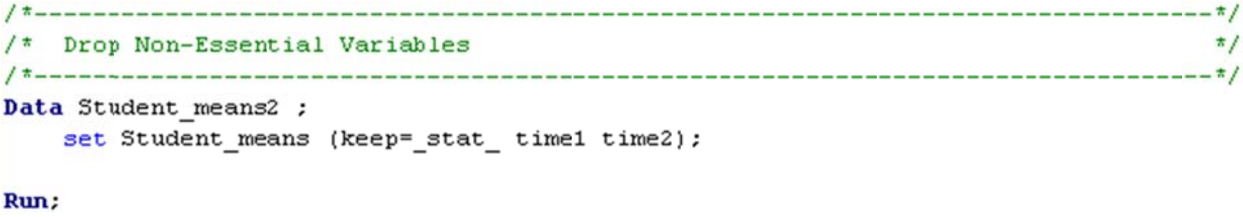
```
/*-------------------------------------------------------------------------*/
/*  Compute the Mean                                                       */
/*-------------------------------------------------------------------------*/
Proc Means data = SASPaper.Student_data noprint mean std;
    output out = Student_means;
    var time1 time2;
Run;
```



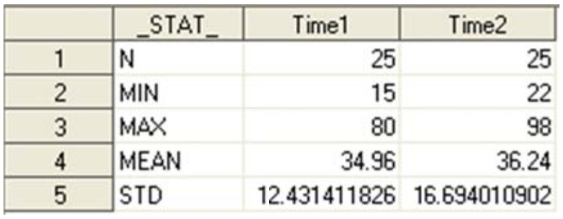
	TYPE	_FREQ_	_STAT_	Time1	Time2
1	0	25	N	25	25
2	0	25	MIN	15	22
3	0	25	MAX	80	98
4	0	25	MEAN	34.96	36.24
5	0	25	STD	12.431411826	16.694010902

Next we will drop the non-essential variables. We only the need the _STAT_ variables, so we drop the _TYPE_ and _FREQ_ variables.

Sample SAS Code and Output Dataset



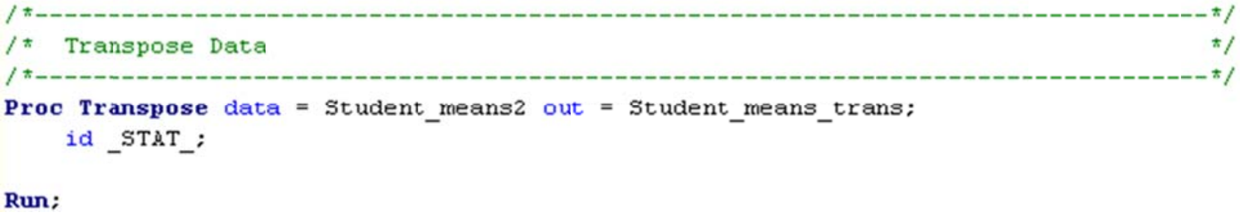
```
/*-------------------------------------------------------------------------*/
/*  Drop Non-Essential Variables                                           */
/*-------------------------------------------------------------------------*/
Data Student_means2 ;
    set Student_means (keep=_stat_ time1 time2);

Run;
```



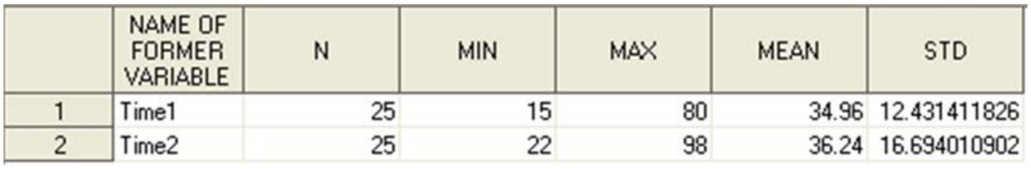
	STAT	Time1	Time2
1	N	25	25
2	MIN	15	22
3	MAX	80	98
4	MEAN	34.96	36.24
5	STD	12.431411826	16.694010902

Once we get the mean and standard deviations for both Time1 & Time2 we will need to transpose the data to get the data into a format that we can use to calculate three standard deviations above and below the means.

Sample SAS Code and Output Dataset



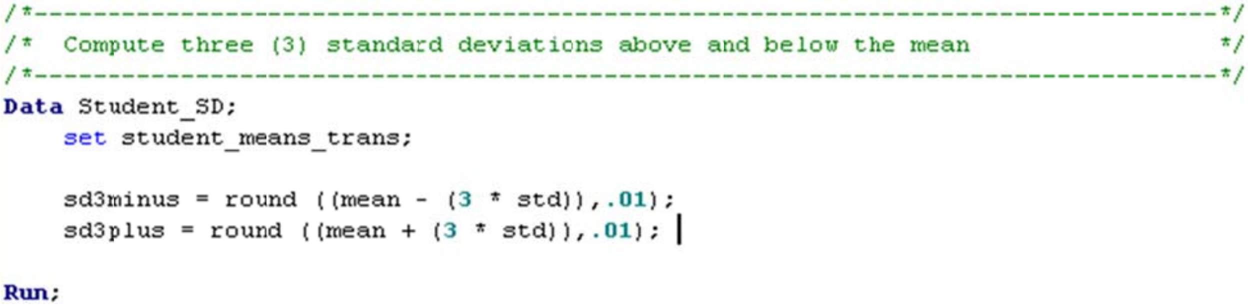
```
/*-------------------------------------------------------------------------------*/
/*  Transpose Data                                                               */
/*-------------------------------------------------------------------------------*/
Proc Transpose data = Student_means2 out = Student_means_trans;
    id _STAT_;

Run;
```



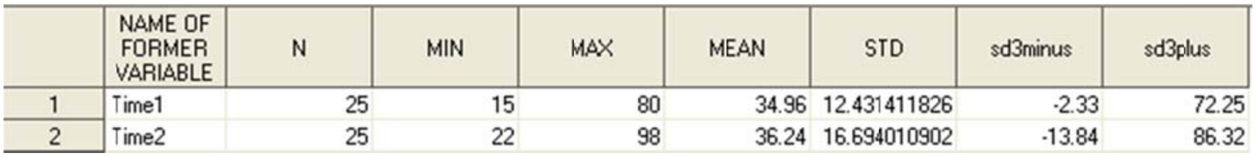
	NAME OF FORMER VARIABLE	N	MIN	MAX	MEAN	STD
1	Time1	25	15	80	34.96	12.431411826
2	Time2	25	22	98	36.24	16.694010902

At this point we can calculate three (3) standard deviations above and below the means.

Sample SAS Code and Output Dataset



```
/*-------------------------------------------------------------------------------*/
/*  Compute three (3) standard deviations above and below the mean               */
/*-------------------------------------------------------------------------------*/
Data Student_SD;
    set student_means_trans;

    sd3minus = round ((mean - (3 * std)),.01);
    sd3plus = round ((mean + (3 * std)),.01);

Run;
```

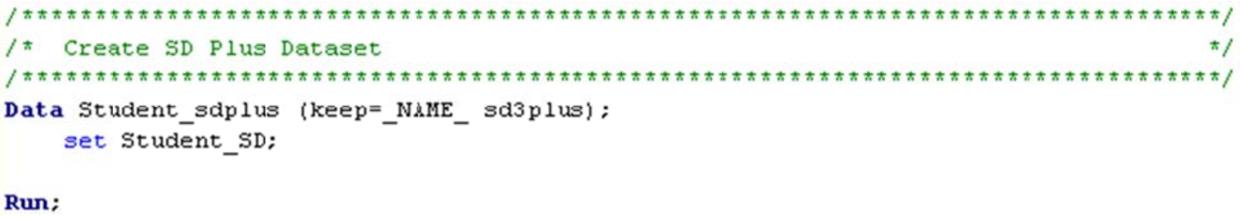


	NAME OF FORMER VARIABLE	N	MIN	MAX	MEAN	STD	sd3minus	sd3plus
1	Time1	25	15	80	34.96	12.431411826	-2.33	72.25
2	Time2	25	22	98	36.24	16.694010902	-13.84	86.32

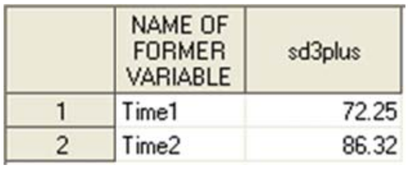
Now we have the four variables we need: sd3minus and sd3plus at Time1 and Time2. Now, to put those values in the original dataset we first split the new data into two datasets: one with 3SD ABOVE the mean for Time1 and Time2, and one with 3SD BELOW the mean for Time1 and Time2.

To create the 3SD dataset ABOVE the mean, we will create a dataset with only the Time variables and the 3SD ABOVE values, transpose the data, and then rename variables and drop non-essential variables.

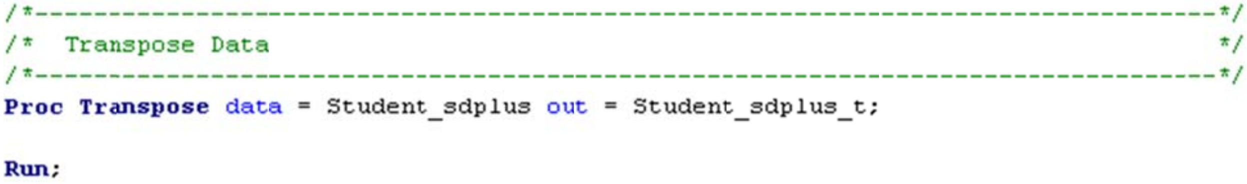
Sample SAS Code and Output Dataset



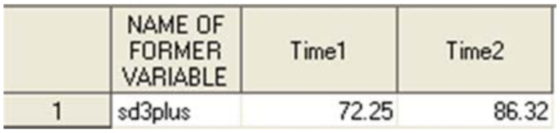
```
/*******************************************************************************/
/*  Create SD Plus Dataset                                                     */
/*******************************************************************************/
Data Student_sdplus (keep=_NAME_ sd3plus);
    set Student_SD;

Run;
```



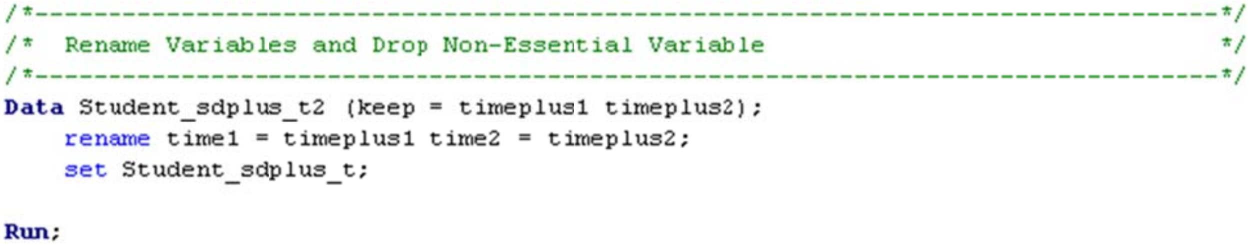
	NAME OF FORMER VARIABLE	sd3plus
1	Time1	72.25
2	Time2	86.32



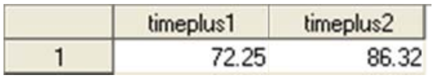
```
/*-----------------------------------------------------------------------------*/
/*  Transpose Data                                                             */
/*-----------------------------------------------------------------------------*/
Proc Transpose data = Student_sdplus out = Student_sdplus_t;

Run;
```



	NAME OF FORMER VARIABLE	Time1	Time2
1	sd3plus	72.25	86.32



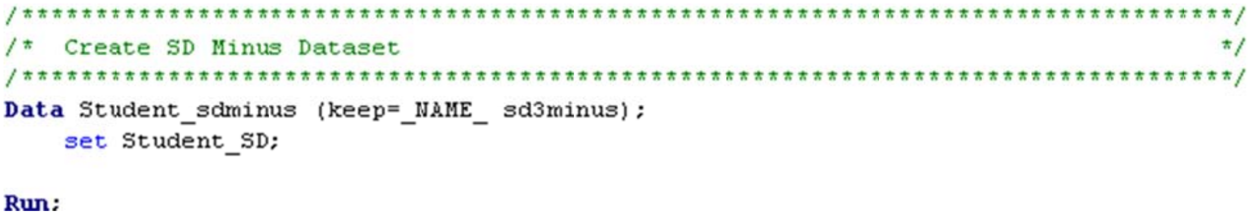
```
/*-----------------------------------------------------------------------------*/
/*  Rename Variables and Drop Non-Essential Variable                           */
/*-----------------------------------------------------------------------------*/
Data Student_sdplus_t2 (keep = timeplus1 timeplus2);
    rename time1 = timeplus1 time2 = timeplus2;
    set Student_sdplus_t;

Run;
```



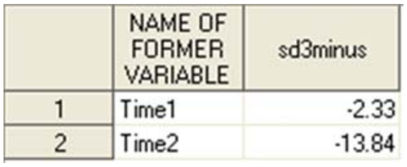
	timeplus1	timeplus2
1	72.25	86.32

We will do the same steps to create the 3SD BELOW dataset.

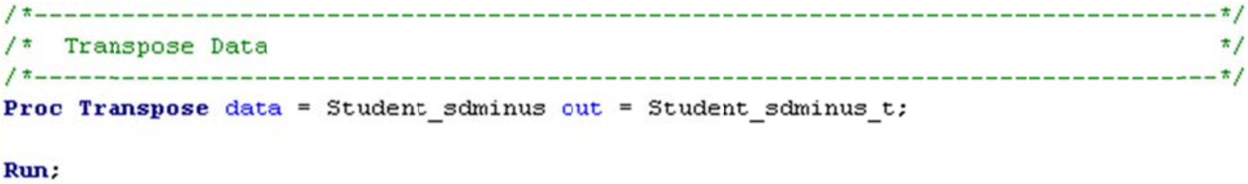
Sample SAS Code and Output Dataset



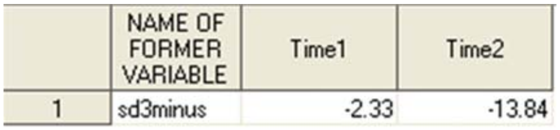
```
/*******************************************************************************/
/*  Create SD Minus Dataset                                                    */
/*******************************************************************************/
Data Student_sdminus (keep=_NAME_ sd3minus);
    set Student_SD;

Run;
```



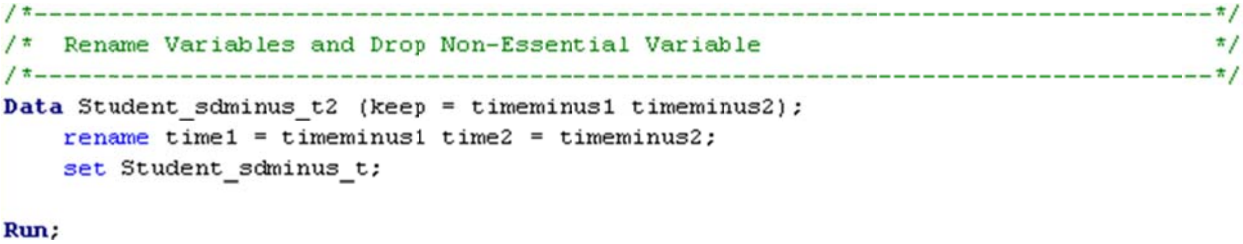
	NAME OF FORMER VARIABLE	sd3minus
1	Time1	-2.33
2	Time2	-13.84



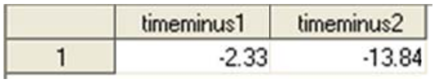
```
/*---------------------------------------------------------------------------------*/
/*  Transpose Data                                                                  */
/*---------------------------------------------------------------------------------*/
Proc Transpose data = Student_sdminus out = Student_sdminus_t;

Run;
```



	NAME OF FORMER VARIABLE	Time1	Time2
1	sd3minus	-2.33	-13.84



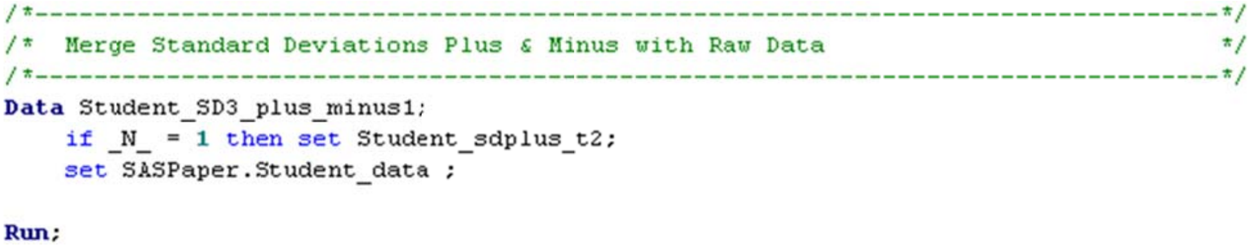
```
/*---------------------------------------------------------------------------------*/
/*  Rename Variables and Drop Non-Essential Variable                                */
/*---------------------------------------------------------------------------------*/
Data Student_sdminus_t2 (keep = timeminus1 timeminus2);
    rename time1 = timeminus1 time2 = timeminus2;
    set Student_sdminus_t;

Run;
```



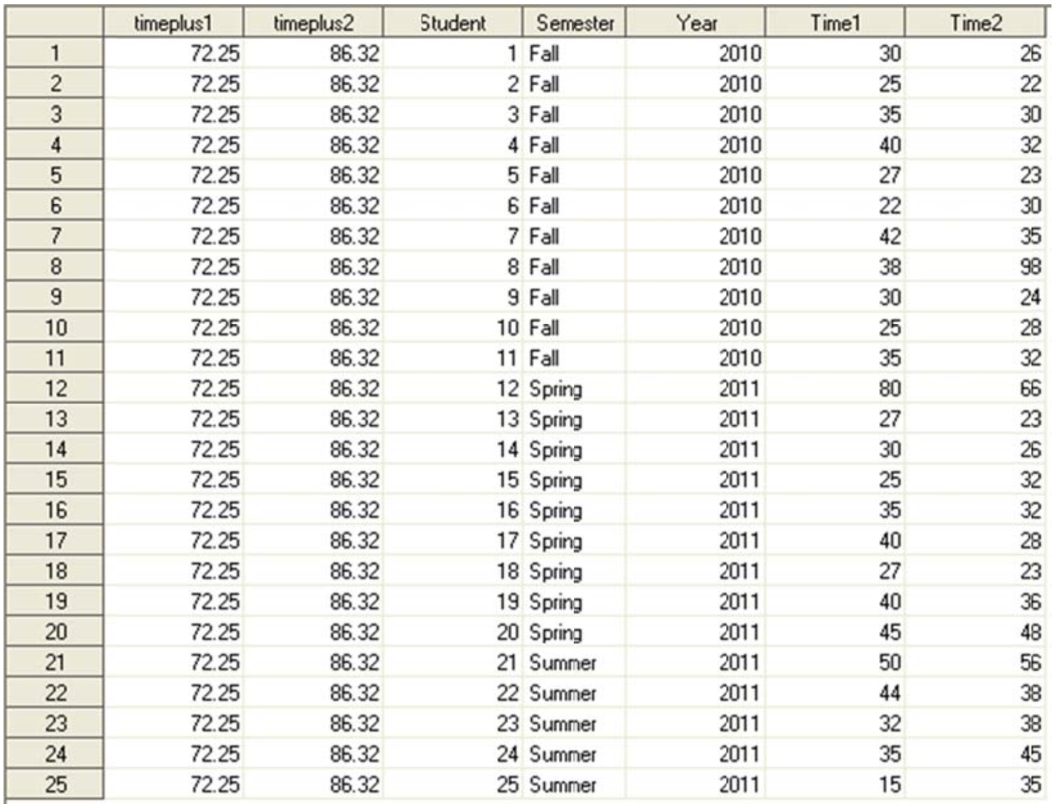
	timeminus1	timeminus2
1	-2.33	-13.84

Now we are ready to merge the data together by using the If _N_=1 Set syntax. Reading the syntax below we are stating: if the number of observations for the first dataset (Student_sdplus_t2) equals one (1) then set that observation with every observation in the Student_data dataset located in the SASPaper library, and output the file to a dataset named Student_SD3_plus_minus1.

Sample SAS Code and Output Dataset



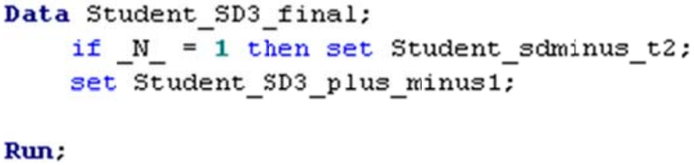
```
/*---------------------------------------------------------------------------------*/
/*  Merge Standard Deviations Plus & Minus with Raw Data                            */
/*---------------------------------------------------------------------------------*/
Data Student_SD3_plus_minus1;
    if _N_ = 1 then set Student_sdplus_t2;
    set SASPaper.Student_data ;

Run;
```



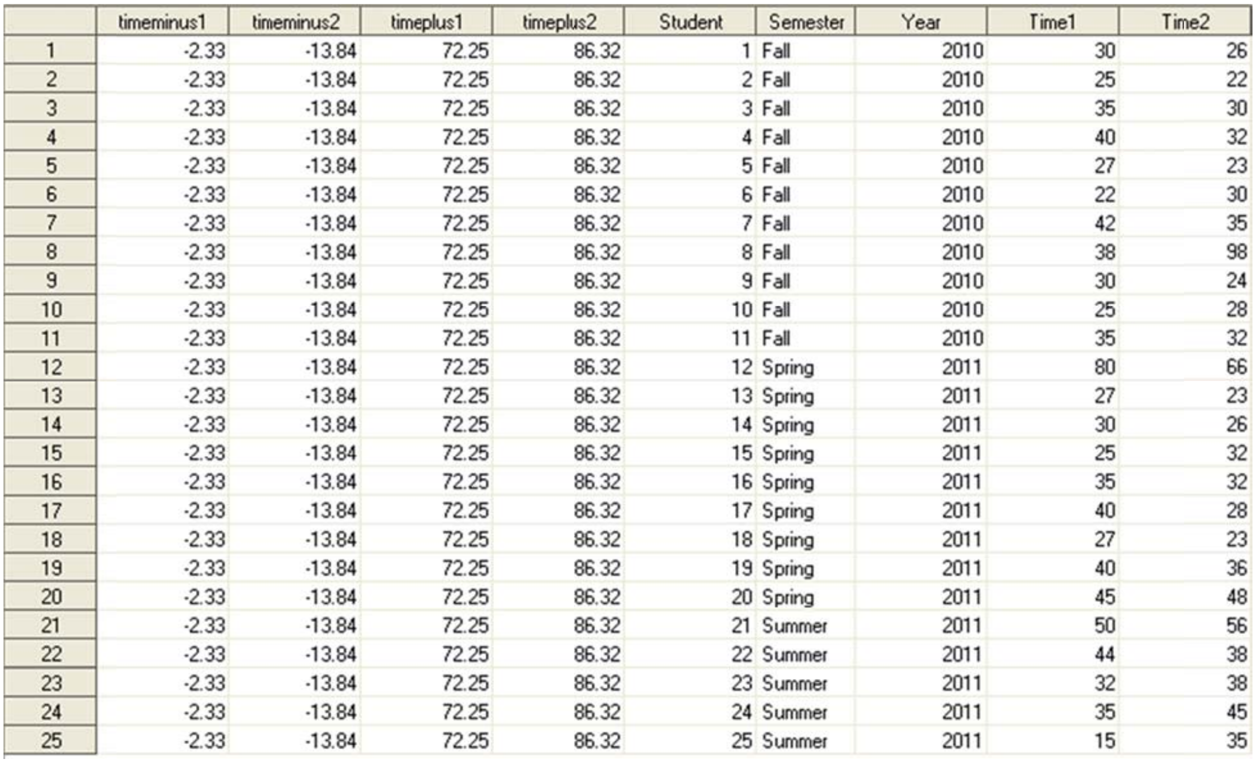
	timeplus1	timeplus2	Student	Semester	Year	Time1	Time2
1	72.25	86.32	1	Fall	2010	30	26
2	72.25	86.32	2	Fall	2010	25	22
3	72.25	86.32	3	Fall	2010	35	30
4	72.25	86.32	4	Fall	2010	40	32
5	72.25	86.32	5	Fall	2010	27	23
6	72.25	86.32	6	Fall	2010	22	30
7	72.25	86.32	7	Fall	2010	42	35
8	72.25	86.32	8	Fall	2010	38	98
9	72.25	86.32	9	Fall	2010	30	24
10	72.25	86.32	10	Fall	2010	25	28
11	72.25	86.32	11	Fall	2010	35	32
12	72.25	86.32	12	Spring	2011	80	66
13	72.25	86.32	13	Spring	2011	27	23
14	72.25	86.32	14	Spring	2011	30	26
15	72.25	86.32	15	Spring	2011	25	32
16	72.25	86.32	16	Spring	2011	35	32
17	72.25	86.32	17	Spring	2011	40	28
18	72.25	86.32	18	Spring	2011	27	23
19	72.25	86.32	19	Spring	2011	40	36
20	72.25	86.32	20	Spring	2011	45	48
21	72.25	86.32	21	Summer	2011	50	56
22	72.25	86.32	22	Summer	2011	44	38
23	72.25	86.32	23	Summer	2011	32	38
24	72.25	86.32	24	Summer	2011	35	45
25	72.25	86.32	25	Summer	2011	15	35

Notice that every observation in the original Student_Data dataset has the same 3SD Above values for Time1 and Time2 merged with it.

We will now merge the 3SD Below values.



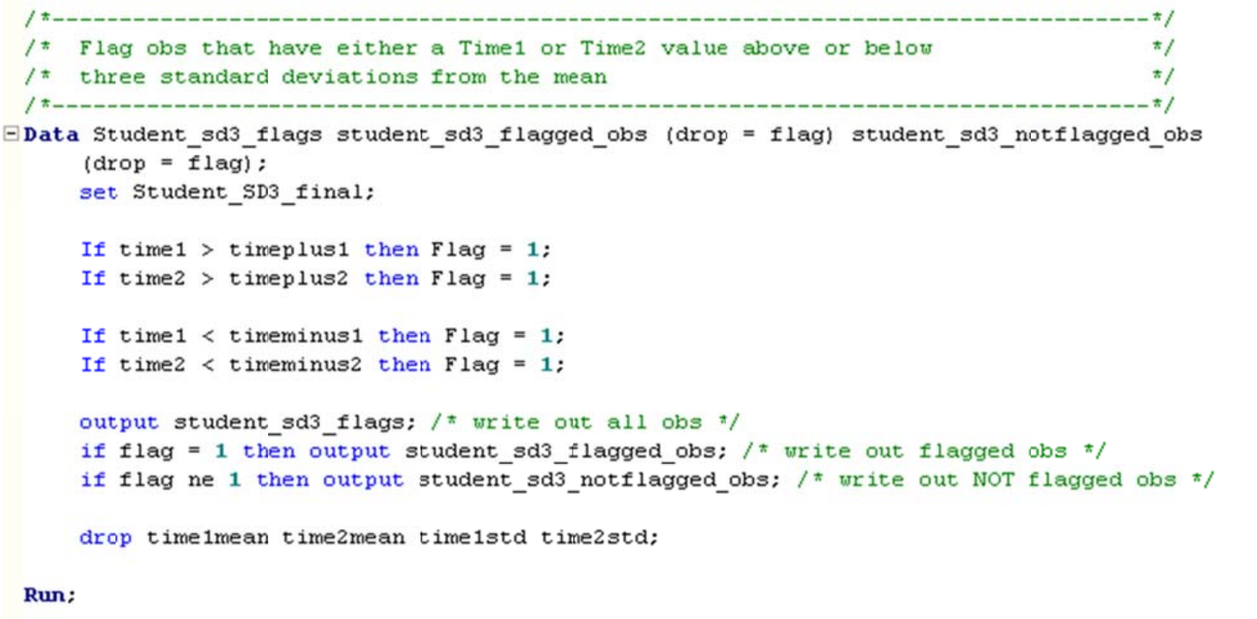
```
Data Student_SD3_final;
     if _N_ = 1 then set Student_sdminus_t2;
     set Student_SD3_plus_minus1;

Run;
```



	timeminus1	timeminus2	timeplus1	timeplus2	Student	Semester	Year	Time1	Time2
1	-2.33	-13.84	72.25	86.32	1	Fall	2010	30	26
2	-2.33	-13.84	72.25	86.32	2	Fall	2010	25	22
3	-2.33	-13.84	72.25	86.32	3	Fall	2010	35	30
4	-2.33	-13.84	72.25	86.32	4	Fall	2010	40	32
5	-2.33	-13.84	72.25	86.32	5	Fall	2010	27	23
6	-2.33	-13.84	72.25	86.32	6	Fall	2010	22	30
7	-2.33	-13.84	72.25	86.32	7	Fall	2010	42	35
8	-2.33	-13.84	72.25	86.32	8	Fall	2010	38	98
9	-2.33	-13.84	72.25	86.32	9	Fall	2010	30	24
10	-2.33	-13.84	72.25	86.32	10	Fall	2010	25	28
11	-2.33	-13.84	72.25	86.32	11	Fall	2010	35	32
12	-2.33	-13.84	72.25	86.32	12	Spring	2011	80	66
13	-2.33	-13.84	72.25	86.32	13	Spring	2011	27	23
14	-2.33	-13.84	72.25	86.32	14	Spring	2011	30	26
15	-2.33	-13.84	72.25	86.32	15	Spring	2011	25	32
16	-2.33	-13.84	72.25	86.32	16	Spring	2011	35	32
17	-2.33	-13.84	72.25	86.32	17	Spring	2011	40	28
18	-2.33	-13.84	72.25	86.32	18	Spring	2011	27	23
19	-2.33	-13.84	72.25	86.32	19	Spring	2011	40	36
20	-2.33	-13.84	72.25	86.32	20	Spring	2011	45	48
21	-2.33	-13.84	72.25	86.32	21	Summer	2011	50	56
22	-2.33	-13.84	72.25	86.32	22	Summer	2011	44	38
23	-2.33	-13.84	72.25	86.32	23	Summer	2011	32	38
24	-2.33	-13.84	72.25	86.32	24	Summer	2011	35	45
25	-2.33	-13.84	72.25	86.32	25	Summer	2011	15	35

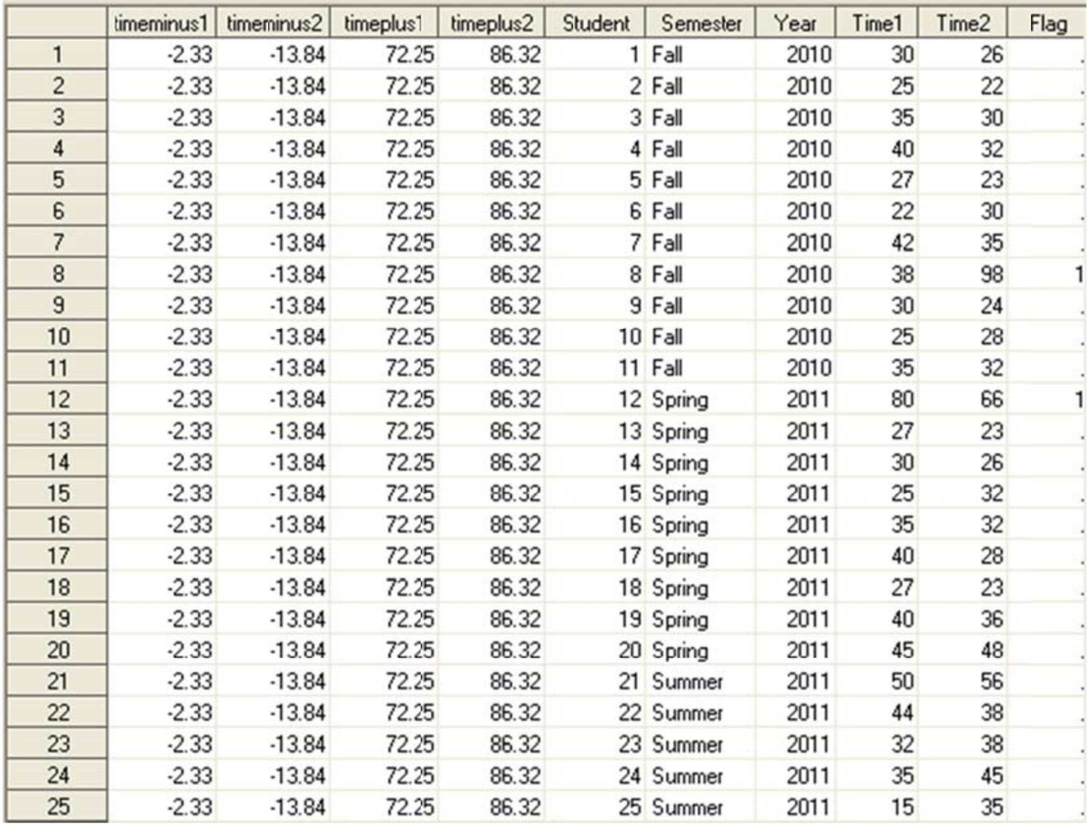
The final file has both the 3SD values Above and 3SD values Below for the Time1 and Time2 variables. Now we can flag the observations that have either a Time1 or Time2 value above or below three standard deviations from the mean. We can create a dataset with all of the observations and flags, but we can also create a dataset with only flagged observations and another dataset with un-flagged observations.

Sample SAS Code and Output Dataset



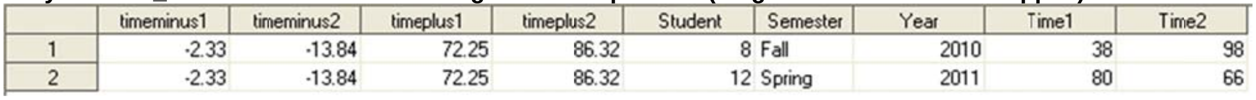
```
/*--------------------------------------------------------------------------------*/
/*  Flag obs that have either a Time1 or Time2 value above or below              */
/*  three standard deviations from the mean                                      */
/*--------------------------------------------------------------------------------*/
Data Student_sd3_flags student_sd3_flagged_obs (drop = flag) student_sd3_notflagged_obs
    (drop = flag);
    set Student_SD3_final;

    If time1 > timeplus1 then Flag = 1;
    If time2 > timeplus2 then Flag = 1;

    If time1 < timeminus1 then Flag = 1;
    If time2 < timeminus2 then Flag = 1;

    output student_sd3_flags; /* write out all obs */
    if flag = 1 then output student_sd3_flagged_obs; /* write out flagged obs */
    if flag ne 1 then output student_sd3_notflagged_obs; /* write out NOT flagged obs */

    drop time1mean time2mean time1std time2std;

Run;
```

All Student_Data observations with Flag variable



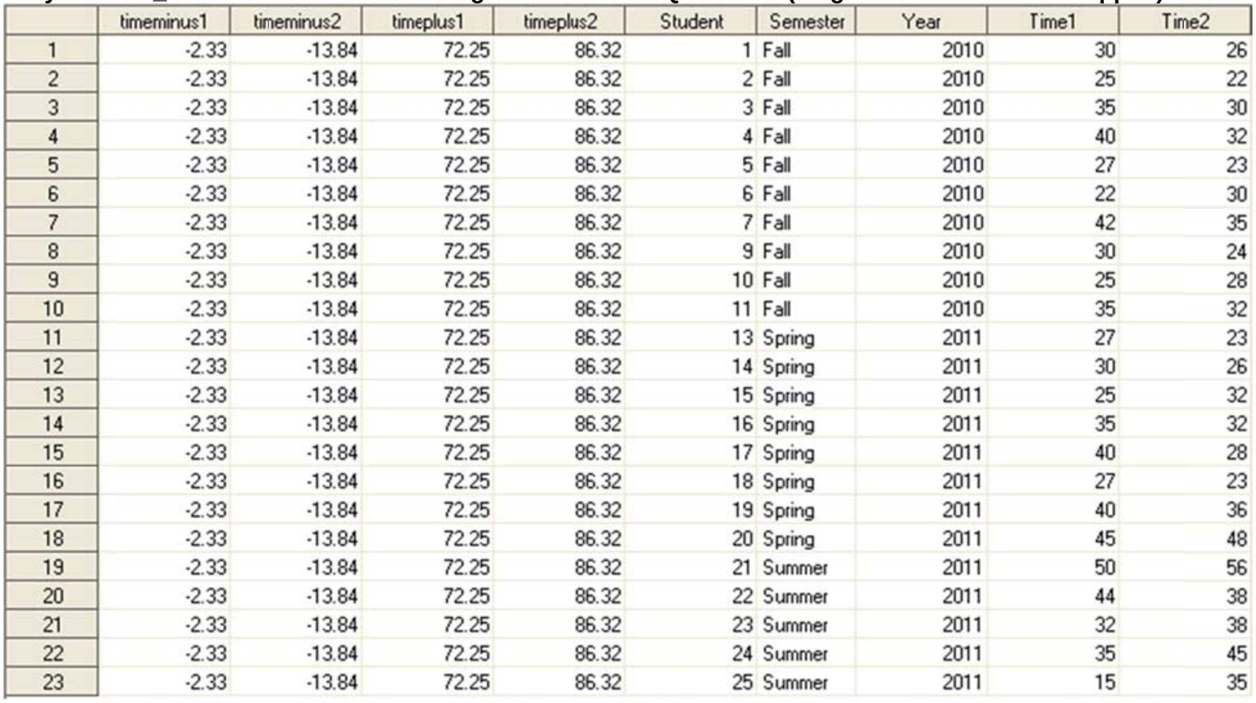
	timeminus1	timeminus2	timeplus1	timeplus2	Student	Semester	Year	Time1	Time2	Flag
1	-2.33	-13.84	72.25	86.32	1	Fall	2010	30	26	.
2	-2.33	-13.84	72.25	86.32	2	Fall	2010	25	22	.
3	-2.33	-13.84	72.25	86.32	3	Fall	2010	35	30	.
4	-2.33	-13.84	72.25	86.32	4	Fall	2010	40	32	.
5	-2.33	-13.84	72.25	86.32	5	Fall	2010	27	23	.
6	-2.33	-13.84	72.25	86.32	6	Fall	2010	22	30	.
7	-2.33	-13.84	72.25	86.32	7	Fall	2010	42	35	.
8	-2.33	-13.84	72.25	86.32	8	Fall	2010	38	98	1
9	-2.33	-13.84	72.25	86.32	9	Fall	2010	30	24	.
10	-2.33	-13.84	72.25	86.32	10	Fall	2010	25	28	.
11	-2.33	-13.84	72.25	86.32	11	Fall	2010	35	32	.
12	-2.33	-13.84	72.25	86.32	12	Spring	2011	80	66	1
13	-2.33	-13.84	72.25	86.32	13	Spring	2011	27	23	.
14	-2.33	-13.84	72.25	86.32	14	Spring	2011	30	26	.
15	-2.33	-13.84	72.25	86.32	15	Spring	2011	25	32	.
16	-2.33	-13.84	72.25	86.32	16	Spring	2011	35	32	.
17	-2.33	-13.84	72.25	86.32	17	Spring	2011	40	28	.
18	-2.33	-13.84	72.25	86.32	18	Spring	2011	27	23	.
19	-2.33	-13.84	72.25	86.32	19	Spring	2011	40	36	.
20	-2.33	-13.84	72.25	86.32	20	Spring	2011	45	48	.
21	-2.33	-13.84	72.25	86.32	21	Summer	2011	50	56	.
22	-2.33	-13.84	72.25	86.32	22	Summer	2011	44	38	.
23	-2.33	-13.84	72.25	86.32	23	Summer	2011	32	38	.
24	-2.33	-13.84	72.25	86.32	24	Summer	2011	35	45	.
25	-2.33	-13.84	72.25	86.32	25	Summer	2011	15	35	.

Only Student_Data observations with Flag variable equal to 1 (Flag variable has been dropped)



	timeminus1	timeminus2	timeplus1	timeplus2	Student	Semester	Year	Time1	Time2
1	-2.33	-13.84	72.25	86.32	8	Fall	2010	38	98
2	-2.33	-13.84	72.25	86.32	12	Spring	2011	80	66

Only Student_Data observations with Flag variable NOT EQUAL to 1 (Flag variable has been dropped)



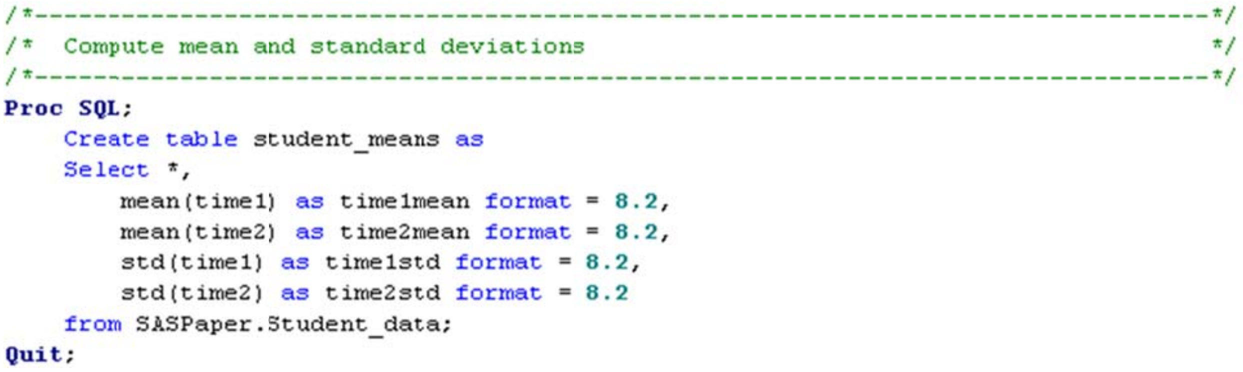
	timeminus1	timeminus2	timeplus1	timeplus2	Student	Semester	Year	Time1	Time2
1	-2.33	-13.84	72.25	86.32	1	Fall	2010	30	26
2	-2.33	-13.84	72.25	86.32	2	Fall	2010	25	22
3	-2.33	-13.84	72.25	86.32	3	Fall	2010	35	30
4	-2.33	-13.84	72.25	86.32	4	Fall	2010	40	32
5	-2.33	-13.84	72.25	86.32	5	Fall	2010	27	23
6	-2.33	-13.84	72.25	86.32	6	Fall	2010	22	30
7	-2.33	-13.84	72.25	86.32	7	Fall	2010	42	35
8	-2.33	-13.84	72.25	86.32	9	Fall	2010	30	24
9	-2.33	-13.84	72.25	86.32	10	Fall	2010	25	28
10	-2.33	-13.84	72.25	86.32	11	Fall	2010	35	32
11	-2.33	-13.84	72.25	86.32	13	Spring	2011	27	23
12	-2.33	-13.84	72.25	86.32	14	Spring	2011	30	26
13	-2.33	-13.84	72.25	86.32	15	Spring	2011	25	32
14	-2.33	-13.84	72.25	86.32	16	Spring	2011	35	32
15	-2.33	-13.84	72.25	86.32	17	Spring	2011	40	28
16	-2.33	-13.84	72.25	86.32	18	Spring	2011	27	23
17	-2.33	-13.84	72.25	86.32	19	Spring	2011	40	36
18	-2.33	-13.84	72.25	86.32	20	Spring	2011	45	48
19	-2.33	-13.84	72.25	86.32	21	Summer	2011	50	56
20	-2.33	-13.84	72.25	86.32	22	Summer	2011	44	38
21	-2.33	-13.84	72.25	86.32	23	Summer	2011	32	38
22	-2.33	-13.84	72.25	86.32	24	Summer	2011	35	45
23	-2.33	-13.84	72.25	86.32	25	Summer	2011	15	35

USING PROC SQL

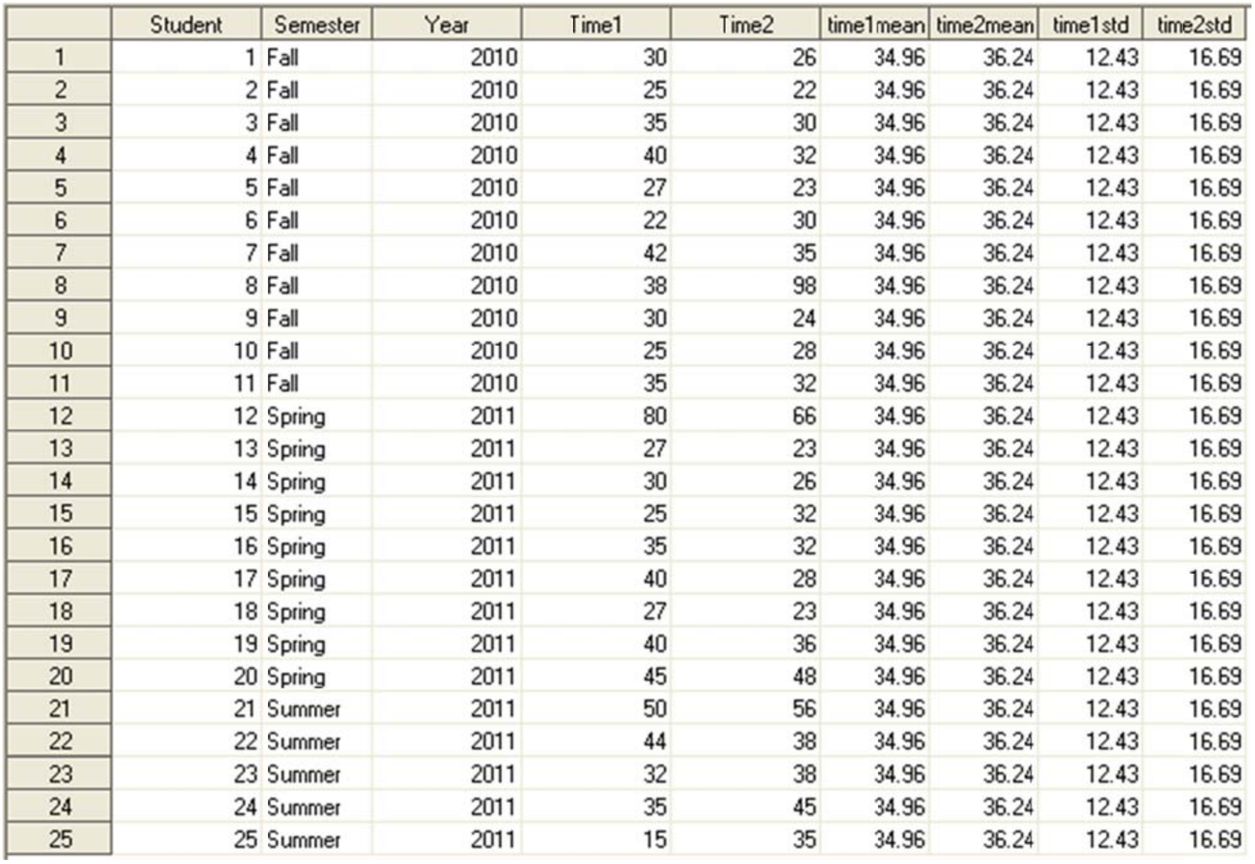
Using PROC SQL to achieve the same results can eliminate multiple steps; however, many people are reluctant to use PROC SQL if they are unfamiliar with it.

As with the If _N_=1 Set method, we must first begin by calculating the means and standard deviation for the Time1 and Time2 variables. In the SAS Code example below, note that the format statement is not necessary, but it helps to make the values more readable.

Sample SAS Code and Output Dataset



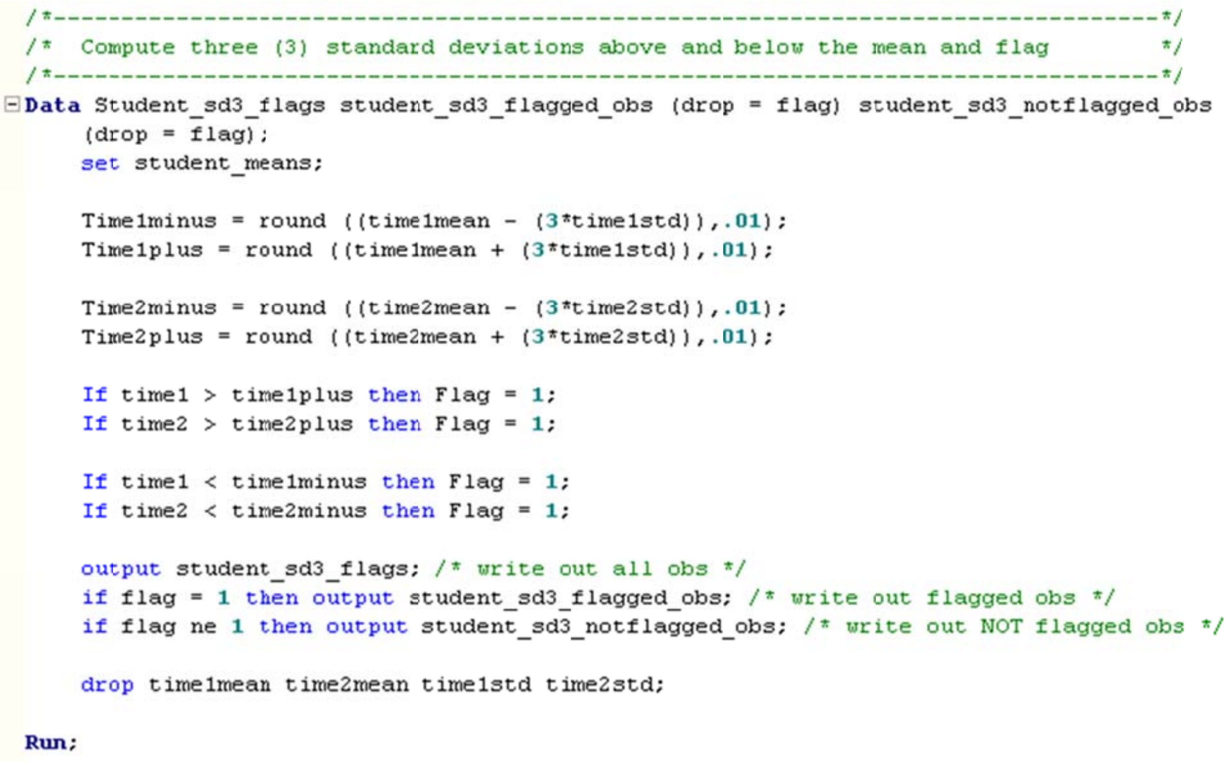
```
/*------------------------------------------------------------------------------*/
/*  Compute mean and standard deviations                                        */
/*------------------------------------------------------------------------------*/
Proc SQL;
    Create table student_means as
    Select *,
        mean(time1) as time1mean format = 8.2,
        mean(time2) as time2mean format = 8.2,
        std(time1) as time1std format = 8.2,
        std(time2) as time2std format = 8.2
    from SASPaper.Student_data;
Quit;
```



	Student	Semester	Year	Time1	Time2	time1mean	time2mean	time1std	time2std
1	1	Fall	2010	30	26	34.96	36.24	12.43	16.69
2	2	Fall	2010	25	22	34.96	36.24	12.43	16.69
3	3	Fall	2010	35	30	34.96	36.24	12.43	16.69
4	4	Fall	2010	40	32	34.96	36.24	12.43	16.69
5	5	Fall	2010	27	23	34.96	36.24	12.43	16.69
6	6	Fall	2010	22	30	34.96	36.24	12.43	16.69
7	7	Fall	2010	42	35	34.96	36.24	12.43	16.69
8	8	Fall	2010	38	98	34.96	36.24	12.43	16.69
9	9	Fall	2010	30	24	34.96	36.24	12.43	16.69
10	10	Fall	2010	25	28	34.96	36.24	12.43	16.69
11	11	Fall	2010	35	32	34.96	36.24	12.43	16.69
12	12	Spring	2011	80	66	34.96	36.24	12.43	16.69
13	13	Spring	2011	27	23	34.96	36.24	12.43	16.69
14	14	Spring	2011	30	26	34.96	36.24	12.43	16.69
15	15	Spring	2011	25	32	34.96	36.24	12.43	16.69
16	16	Spring	2011	35	32	34.96	36.24	12.43	16.69
17	17	Spring	2011	40	28	34.96	36.24	12.43	16.69
18	18	Spring	2011	27	23	34.96	36.24	12.43	16.69
19	19	Spring	2011	40	36	34.96	36.24	12.43	16.69
20	20	Spring	2011	45	48	34.96	36.24	12.43	16.69
21	21	Summer	2011	50	56	34.96	36.24	12.43	16.69
22	22	Summer	2011	44	38	34.96	36.24	12.43	16.69
23	23	Summer	2011	32	38	34.96	36.24	12.43	16.69
24	24	Summer	2011	35	45	34.96	36.24	12.43	16.69
25	25	Summer	2011	15	35	34.96	36.24	12.43	16.69

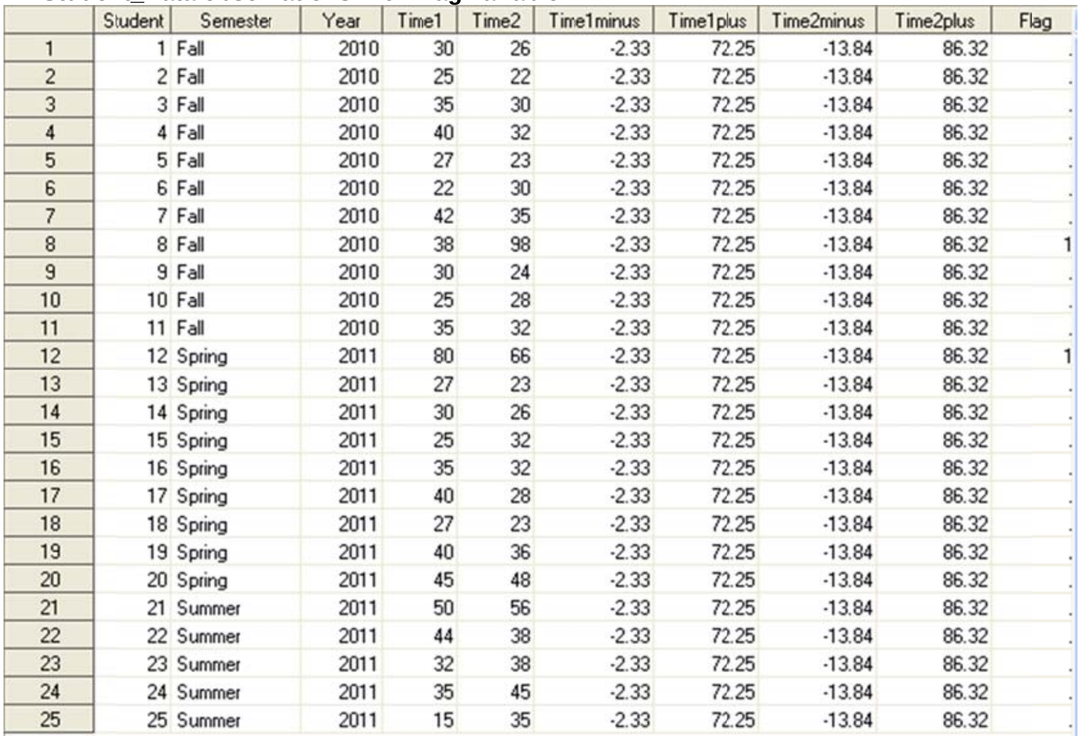
Next we will use a data step to compute three (3) standard deviations above and below the means, flag the observations, and write out the three different datasets. We will also eliminate non-essential variables for each output dataset specified.

Sample SAS Code and Output Dataset



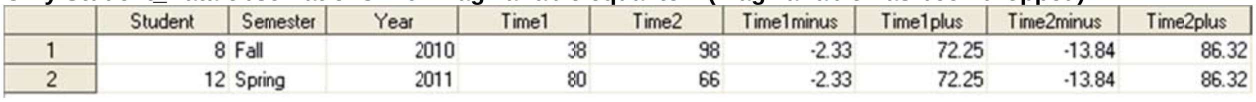
```
/*-------------------------------------------------------------------------*/
/*  Compute three (3) standard deviations above and below the mean and flag  */
/*-------------------------------------------------------------------------*/
Data Student_sd3_flags student_sd3_flagged_obs (drop = flag) student_sd3_notflagged_obs
     (drop = flag);
     set student_means;

     Time1minus = round ((time1mean - (3*time1std)),.01);
     Time1plus = round ((time1mean + (3*time1std)),.01);

     Time2minus = round ((time2mean - (3*time2std)),.01);
     Time2plus = round ((time2mean + (3*time2std)),.01);

     If time1 > time1plus then Flag = 1;
     If time2 > time2plus then Flag = 1;

     If time1 < time1minus then Flag = 1;
     If time2 < time2minus then Flag = 1;

     output student_sd3_flags; /* write out all obs */
     if flag = 1 then output student_sd3_flagged_obs; /* write out flagged obs */
     if flag ne 1 then output student_sd3_notflagged_obs; /* write out NOT flagged obs */

     drop time1mean time2mean time1std time2std;

Run;
```

All Student_Data observations with Flag variable



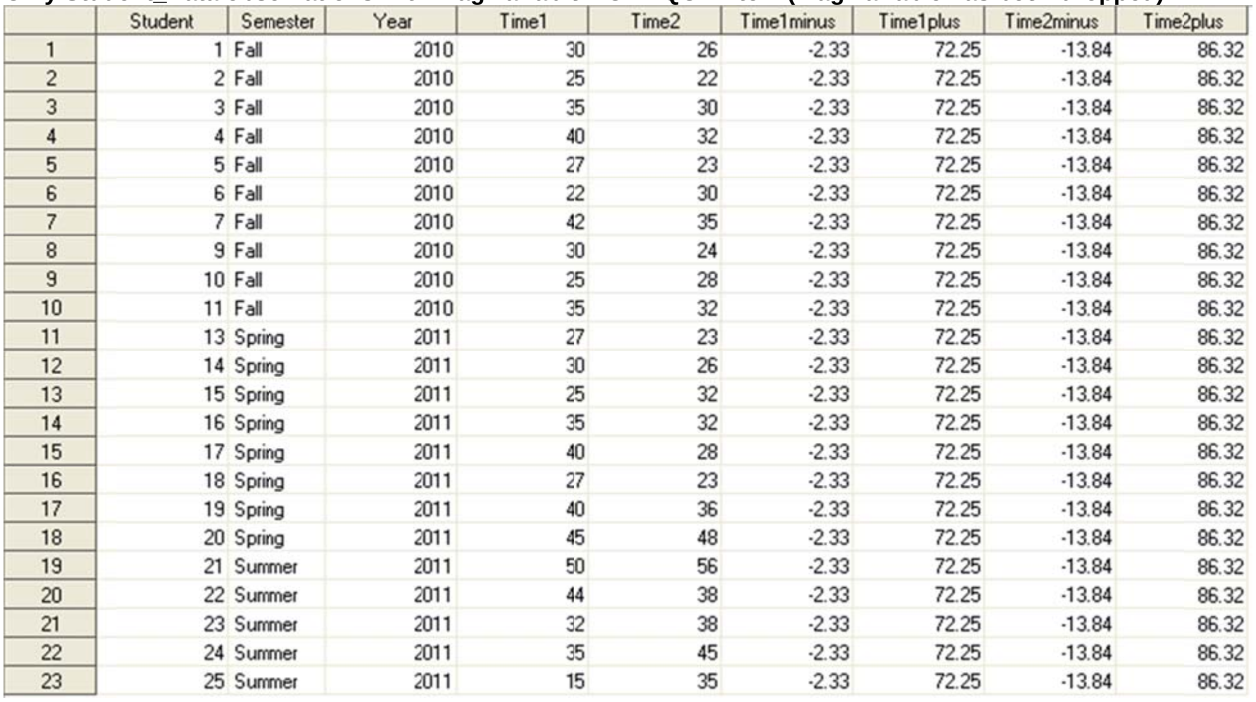
	Student	Semester	Year	Time1	Time2	Time1minus	Time1plus	Time2minus	Time2plus	Flag
1	1	Fall	2010	30	26	-2.33	72.25	-13.84	86.32	.
2	2	Fall	2010	25	22	-2.33	72.25	-13.84	86.32	.
3	3	Fall	2010	35	30	-2.33	72.25	-13.84	86.32	.
4	4	Fall	2010	40	32	-2.33	72.25	-13.84	86.32	.
5	5	Fall	2010	27	23	-2.33	72.25	-13.84	86.32	.
6	6	Fall	2010	22	30	-2.33	72.25	-13.84	86.32	.
7	7	Fall	2010	42	35	-2.33	72.25	-13.84	86.32	.
8	8	Fall	2010	38	98	-2.33	72.25	-13.84	86.32	1
9	9	Fall	2010	30	24	-2.33	72.25	-13.84	86.32	.
10	10	Fall	2010	25	28	-2.33	72.25	-13.84	86.32	.
11	11	Fall	2010	35	32	-2.33	72.25	-13.84	86.32	.
12	12	Spring	2011	80	66	-2.33	72.25	-13.84	86.32	1
13	13	Spring	2011	27	23	-2.33	72.25	-13.84	86.32	.
14	14	Spring	2011	30	26	-2.33	72.25	-13.84	86.32	.
15	15	Spring	2011	25	32	-2.33	72.25	-13.84	86.32	.
16	16	Spring	2011	35	32	-2.33	72.25	-13.84	86.32	.
17	17	Spring	2011	40	28	-2.33	72.25	-13.84	86.32	.
18	18	Spring	2011	27	23	-2.33	72.25	-13.84	86.32	.
19	19	Spring	2011	40	36	-2.33	72.25	-13.84	86.32	.
20	20	Spring	2011	45	48	-2.33	72.25	-13.84	86.32	.
21	21	Summer	2011	50	56	-2.33	72.25	-13.84	86.32	.
22	22	Summer	2011	44	38	-2.33	72.25	-13.84	86.32	.
23	23	Summer	2011	32	38	-2.33	72.25	-13.84	86.32	.
24	24	Summer	2011	35	45	-2.33	72.25	-13.84	86.32	.
25	25	Summer	2011	15	35	-2.33	72.25	-13.84	86.32	.

Only Student_Data observations with Flag variable equal to 1 (Flag variable has been dropped)



	Student	Semester	Year	Time1	Time2	Time1minus	Time1plus	Time2minus	Time2plus
1	8	Fall	2010	38	98	-2.33	72.25	-13.84	86.32
2	12	Spring	2011	80	66	-2.33	72.25	-13.84	86.32

Only Student_Data observations with Flag variable NOT EQUAL to 1 (Flag variable has been dropped)



	Student	Semester	Year	Time1	Time2	Time1minus	Time1plus	Time2minus	Time2plus
1	1	Fall	2010	30	26	-2.33	72.25	-13.84	86.32
2	2	Fall	2010	25	22	-2.33	72.25	-13.84	86.32
3	3	Fall	2010	35	30	-2.33	72.25	-13.84	86.32
4	4	Fall	2010	40	32	-2.33	72.25	-13.84	86.32
5	5	Fall	2010	27	23	-2.33	72.25	-13.84	86.32
6	6	Fall	2010	22	30	-2.33	72.25	-13.84	86.32
7	7	Fall	2010	42	35	-2.33	72.25	-13.84	86.32
8	9	Fall	2010	30	24	-2.33	72.25	-13.84	86.32
9	10	Fall	2010	25	28	-2.33	72.25	-13.84	86.32
10	11	Fall	2010	35	32	-2.33	72.25	-13.84	86.32
11	13	Spring	2011	27	23	-2.33	72.25	-13.84	86.32
12	14	Spring	2011	30	26	-2.33	72.25	-13.84	86.32
13	15	Spring	2011	25	32	-2.33	72.25	-13.84	86.32
14	16	Spring	2011	35	32	-2.33	72.25	-13.84	86.32
15	17	Spring	2011	40	28	-2.33	72.25	-13.84	86.32
16	18	Spring	2011	27	23	-2.33	72.25	-13.84	86.32
17	19	Spring	2011	40	36	-2.33	72.25	-13.84	86.32
18	20	Spring	2011	45	48	-2.33	72.25	-13.84	86.32
19	21	Summer	2011	50	56	-2.33	72.25	-13.84	86.32
20	22	Summer	2011	44	38	-2.33	72.25	-13.84	86.32
21	23	Summer	2011	32	38	-2.33	72.25	-13.84	86.32
22	24	Summer	2011	35	45	-2.33	72.25	-13.84	86.32
23	25	Summer	2011	15	35	-2.33	72.25	-13.84	86.32

Each method yields the exact same data sets and results. It is up to the user to determine which method he or she wants to use. Neither one method is better or worse than the other. Some people may be more comfortable using the data steps and PROCs, while others are more comfortable and familiar with PROC SQL.

CONCLUSION

It is always difficult to figure out ways to do things that are not common. We were able to figure out how to utilize the IF _N_=1 Set method by using the data step and PROCs that we were familiar with. We decided to tackle the same issue using PROC SQL and illustrate both ways to accommodate many SAS users. Each method will yield the same results. As long as the results are accurate, either method will work.

REFERENCES

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Kim Brunnert
Elsevier
11011 Richmond Ave, Ste 450
Houston, TX 77042
Phone: 713-346-6984
E-mail: k.brunnert@elsevier.com

Lisa Mendez
Knowesis, Inc.
San Antonio, TX
Phone: 202-709-8932 ext 231
E-mail: lmendez@knowesis-inc.com