Paper S1-01-2013 The Essence of DATA Step Programming Arthur Li, City of Hope National Medical Center, Duarte, CA

ABSTRACT

The fundamental of SAS[®] programming is DATA step programming. The essence of DATA step programming is to understand how SAS processes the data during the compilation and execution phases. In this paper, you will be exposed to what happens "behind the scenes" while creating a SAS dataset. You will learn how a new dataset is created, one observation at a time, from either a raw text file or an existing SAS dataset, to the program data vector (PDV) and from the PDV to the newly-created SAS dataset. Once you fully understand DATA step processing, learning the SUM and RETAIN statements will become easier to grasp. Relating to this topic, this paper will also cover BY-group processing.

INTRODUCTION

A common befuddlement often facing beginning SAS programmers is that the SAS dataset that they create is not what they intended to create; i.e. there are more or less observations than intended or the value of the newly-created variable was not retained correctly. These types of mistakes are most commonly committed because programming novices learn SAS language syntax without understanding the fundamental SAS programming concepts. The purpose of this paper is to guide you through how DATA step programming operates, step by step, by way of providing various examples.

DATA STEP PROCESSING OVERVIEW

A DATA step is processed in two-phase sequences: compilation and execution phases. In the compilation phase, each statement is scanned for syntax errors. If an error is found, SAS will stop processing. The execution phase only begins after the compilation phase ends. Both phases do not occur simultaneously.

In the execution phase, the DATA step works like a loop, repetitively executing statements to read data values and create observations one at a time. Each loop is called an iteration. We can refer to this type of loop as the implicit loop, which is different from the explicit loop, by using FOR or WHILE statements.

In Program 1 (below), you will see how DATA step processing works. Program 1 reads the raw data from a text file, *example1.txt*. There are two observations and three variables in this dataset, NAME (column 1 - 7), HEIGHT (column 9 - 10), and WEIGHT (column 12 - 14). Notice that the WEIGHT variable for the first observation is entered as "12D", which is a data entry error. Since each variable is occupied in a fixed field and the values for these variables are standard character or numerical values, the column input method is best used to read the raw dataset. You will also notice that a new variable, BMI, is created in this program.

Example1.txt:



Program 1: data ex1; infile 'C:\Users\Arthur\Documents\WUSS Proposal\Forms\example1.txt'; input name \$ 1-7 height 9-10 weight 12-14; BMI = 700*weight/(height*height); output; run;

COMPILATION PHASE

Since Program 1 reads raw datasets, the input buffer is created at the beginning of the compilation phase. The input buffer is used to hold raw data (Figure 1). However, if you read a SAS dataset instead of a raw dataset, the input buffer will not be created.

SAS also creates the PDV in the compilation phase (Figure 1). SAS uses the PDV, a memory area on your computer, to build the new dataset. There are two automatic variables, _N_ and _ERROR_, inside the PDV. _N_ equaling 1 indicates the first observation is being processed, _N_ equaling 2 indicates the second observation is being processed, and so on. The automatic variable _ERROR_ is an indicator variable with values of 1 or 0. _ERROR_ equaling 1 signals the data error of the currently-processed observation, such as reading the data with an incorrect data type. In addition to the two automatic variables, there is one space allocated for each of the variables that will be created from this DATA step. HEIGHT and WEIGHT are the variables that are read from the external raw dataset. BMI is the variable that is created based on HEIGHT and WEIGHT.

Notice that some of the variables in the PDV are marked with (D), which stands for "dropped," and some of the variables are marked with (K), which stands for "kept". The variables marked with (K) will be written to the output dataset; on the other hand, the variables marked with (D) will not. All the automatic variables will not be written to the output dataset.

During the compilation phase, SAS also checks for syntax errors, such as invalid variable names, options, punctuations, misspelled keywords, etc.

Once the compilation is finished, the descriptor portion of the SAS dataset is created, which includes dataset name, the number of observations, and the number, names, and attributes of variables.

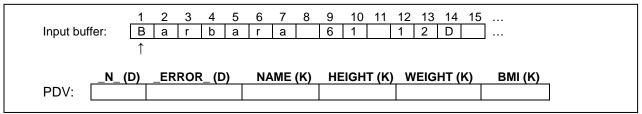


Figure1. Input buffer and the PDV.

EXECUTION PHASE

At the beginning of the execution phase, the automatic variable _N_ is initialized to 1, and _ERROR_ is initialized to 0 since there is no data error. The non-automatic variables are set to missing. Once the INFILE statement identifies the location of the input file, the first data line is read into the input buffer. Then the INPUT statement reads data values from the record in the input buffer according to instructions from the INPUT statement and writes them to the PDV. When the OUTPUT statement is executed, the values from the PDV are copied as a single observation to the SAS dataset *ex1*, but the values from the automatic variables and the variables that are marked (D) are not outputted. Please see Figure 2a to see a detailed explanation of the process of each step for the first iteration.

At the end of the DATA step, the SAS system returns to the beginning of the DATA step to begin the next iteration. The values of the variables in the PDV are reset to missing. The automatic variable _N_ is incremented to 2 and _ERROR_ is set to 0. The second data line is read into the input buffer. See the illustration in Figure 2b for details.

At the end of the DATA step of the second iteration, the SAS system again returns to the beginning of the DATA step to begin the next iteration (see Figure 2c). The values of the variables in the PDV are reset to missing. The automatic variable $_N_$ is incremented to 3. Since there are no more records to read, the SAS system goes to the next DATA or PROC step.

THE OUTPUT STATEMENT

In Program 1, the explicit OUTPUT statement was used, which tells SAS to write the current observation from the PDV to a SAS dataset immediately, not at the end of the DATA step. Without using the explicit OUTPUT statement, by default, every DATA step contains an implicit OUTPUT statement at the end of the DATA step that tells the SAS system to write observations to the dataset. However, placing an explicit OUTPUT statement in a DATA step overrides the implicit output; in other words, the SAS system adds an observation to a dataset only when an explicit OUTPUT statement is executed. Once an explicit OUTPUT statement is used to write an observation to a dataset, there is no longer an implicit OUTPUT statement at the end of the DATA step. Furthermore, more than one OUTPUT statement in the DATA step can be used. You will see an example of using multiple OUTPUT statements later in this paper.

ata ex1	ATION: ;												
	N(D)	_ER	ROR_(D)	NAME (K)	HEIGHT (K)	WE	GHT (K)	BM	I (K)	
PDV:	1		0			•			•			•	
	IATION: T -automatic					tialized to 1	and	I_ER	ROR_ is	s ini	tialize	d to 0.	
nfile ' Input buff	1	s\ Ar 2 a	thur\Doc 3 4 r b	5	nts\WUSS 6 7 8 r a	Proposal \ 9 10 1 6 1	1	rms\ 12 1 1 2	3 14		txt' 	;	
is read ir	nto the inpu	ut but	ffer. SAS	uses	the input po	he location o inter to read d at the beg	l da	ita fro	m the in	put	buffe		
nput na	me \$ 1-7 _N_(D)		ight 9-1 RROR_(D		eight 12- NAME (K)	14; HEIGHT (K)	WE	GHT (K)	вм	I (K)	
PDV:	1		1		Barbara	61			•			•	
	ot in the PD	DV. T	he input p	lues f	rom columr r now rests	is 1-7 from th in column 8,	he ii wh	nput l nich is	immedia	nd c atel	opies y afte	INPU them t r the la	o the st
value read input buffe but 12D is 1. Meanw	ot in the PE d. Then th er and assis s not a valio hile, an err 0*weight	DV. T e INF igns f d nur ror m ror m	The input p PUT stater them to HI neric value essage wi eight*he	ilues f pointe ment EIGH e; this ill be s eight	rom column r now rests instructs SA T. Next, SA causes Wil sent to the S	IS 1-7 from the in column 8, S to read van S attempts for the column 8, S attempts for the column strength of the	he ii wh lue to re nair catir	nput l hich is s fror ead v n mis ng the	ouffer an immedia n columr alues fro sing and locatior	nd c atel ns 9 om 0 L_E n of	st the opies y afte -10 fr colum RROI the d	INPU them to r the later from the ns 12- R_ is s ata err	to the ast ast 14 et to
value read input buffe but 12D is 1. Meanw	ot in the PE d. Then th er and assiss not a valie hile, an err	DV. T e INF igns f d nur ror m ror m	The input p PUT stater them to HI neric value essage wi	ilues f pointe ment EIGH e; this ill be s eight	rom columr r now rests instructs SA T. Next, SA s causes Wi sent to the S	is 1-7 from th in column 8, S to read va S attempts t EIGHT to rer	he ii wh lue to re nair catir	nput l hich is s fror ead v n mis ng the	ouffer an immedia n columr alues fro sing and	nd c atel ns 9 om 0 L_E n of	st the opies y afte -10 fr colum RROI the d	INPU them the r the later om the ns 12- R_ is s	to the ast ast 14 et to
value read input buff but 12D is 1. Meanw BMI = 70 PDV: [EXPLAN/ operations	ot in the PE d. Then th er and assis s not a vali- thile, an err 0*weight N_(D) 1 1	DV. T e INF igns t d nur ror m c/ (h _EF	The input p PUT stater them to HI neric value essage wi eight *he ROR_ (D 1 signment s	Ilues f pointe ment i EIGH e; this ill be s eight) stater	rom columr r now rests instructs SA T. Next, SA s causes Wi sent to the S t); NAME (K) Barbara	IS 1-7 from the second	he ii wh alue to re nair catir	nput l nich is s fror ead v n mis ng the WE	ouffer an immedia n columr alues fro sing and locatior GHT (K)	nd c atel ns { om (E n of)	st the opies ly afte -10 fr colum RROI the d BM	INPU ⁻ them to r the la oom the ns 12- R_ is s ata err I (K)	to the ast ast 14 et to
value read input buffd but 12D is 1. Meanw BMI = 70 PDV: [EXPLAN/ operations butput; Ex1:	ot in the PE d. Then th er and assis not a valiu hile, an err 0*weight _N_(D) 1 <u>ATION</u> : Th s on a miss	OV. T e INF igns f d nur ror m -/ (h 	The input p PUT stater them to HI neric value essage wi eight*he ROR_ (D 1 signment s value will r	Ilues f pointe ment i EIGH e; this ill be s ill be s eight) stater result	rom column r now rests instructs SA T. Next, SA s causes Wi sent to the S NAME (K) Barbara nent is exect in a missing	IS 1-7 from the second	he ii wh alue to re nair catir	nput l nich is s fror ead v n mis ng the WE	ouffer an immedia n columr alues fro sing and locatior GHT (K)	nd c atel ns { om (E n of)	st the opies ly afte -10 fr colum RROI the d BM	INPU ⁻ them to r the la oom the ns 12- R_ is s ata err I (K)	to the ast ast 14 et to
value read input buffd but 12D is 1. Meanw BMI = 70 PDV: [EXPLAN/ operations butput; Ex1: NAME	ot in the PE d. Then th er and assis s not a vali- thile, an err 0*weight 0*weight 1 <u>1</u> <u>ATION</u> : Th s on a missis	VV. T e INF igns f d nurr cor m 	The input p PUT stater them to HI neric value essage wi eight *he ROR_ (D 1 signment s	Ilues f pointe ment i EIGH e; this ill be s ill be s eight) stater result	rom column r now rests instructs SA T. Next, SA s causes Wi sent to the S NAME (K) Barbara nent is exec	IS 1-7 from the second	he ii wh alue to re nair catir	nput l nich is s fror ead v n mis ng the WE	ouffer an immedia n columr alues fro sing and locatior GHT (K)	nd c atel ns { om (E n of)	st the opies ly afte -10 fr colum RROI the d BM	INPU ⁻ them to r the la oom the ns 12- R_ is s ata err I (K)	to the ast ast 14 et to
value read input buffe but 12D is 1. Meanw BMI = 70 PDV: [EXPLANA operations butput; Ex1:	ot in the PE d. Then th er and assis s not a vali- thile, an err 0*weight 0*weight 1 <u>1</u> <u>ATION</u> : Th s on a missis	OV. T e INF igns f d nur ror m -/ (h 	The input p PUT stater them to HI neric value essage wi eight*he ROR_ (D 1 signment s value will r	Ilues f pointe ment i EIGH e; this ill be s ill be s eight) stater result	rom column r now rests instructs SA T. Next, SA s causes Wi sent to the S NAME (K) Barbara nent is exect in a missing	IS 1-7 from the second	he ii wh alue to re nair catir	nput l nich is s fror ead v n mis ng the WE	ouffer an immedia n columr alues fro sing and locatior GHT (K)	nd c atel ns { om (E n of)	st the opies ly afte -10 fr colum RROI the d BM	INPU ⁻ them to r the la oom the ns 12- R_ is s ata err I (K)	to the ast ast 14 et to
value read input buffe but 12D is 1. Meanw MI = 70 PDV: [EXPLAN/ operations putput; Ex1: NAME Barbara EXPLAN PDV are Cun; EXPLAN	ot in the PE d. Then th er and assis s not a vali- thile, an err 0*weight N_(D) 1 1 ATION: Th s on a miss HEIGH	DV. T e INF igns t d nur ror m	The input p PUT stater them to HI meric value essage wi eight *he ROR_ (D 1 signment s value will r WEIGHT the OUTF ngle obser	Ilues f pointe ment EIGH e; this ill be s eight) stater result - 	rom column r now rests instructs SA T. Next, SA s causes Wi sent to the S NAME (K) Barbara nent is exect in a missing BMI	IS 1-7 from the second	ne ii wh lue to re mair catir K) //I w	nput l hich is s fror ead v n mis ng the will rer vill rer	ouffer an immedia n columr alues fro sing and locatior GHT (K) • nain mis	ad c atel om of E on of E sin d w	st the opies y afte -10 fr colum RROI the d BM g sinc	INPU ⁻ them to r the la om the ns 12- R_ is s ata err I (K) • • • • •	o the ast 14 et to or.

	RATION:		0 7 0	0 10 11	40 40 44 45	
Input buff	er: <u>J</u> c		6 7 8	9 10 11	<u>12 13 14 15</u> 1 7 5]
	↑					_
lata ex1;		RROR_ (D)	NAME (K)	HEIGHT (K)	WEIGHT (K)	BMI (K)
PDV:	2	0		•	•	•
positione	d at the begin	nning of the inp	out buffer. The	automatic vari	fer. The input p able _N_ is initia automatic variab	alized to 2 and
				-	rms\example1	.txt';
input na		eight 9-10 RROR_(D)	NAME (K)	14; HEIGHT (K)	WEIGHT (K)	BMI (K)
PDV:	2	0	John	62	175	•
			his observatio	<u></u>		
BMI = 70	-	height*heig RROR_(D)		HEIGHT (K)	WEIGHT (K)	BMI (K)
BMI = 70 PDV: [-	height*heig	ght);		WEIGHT (К) 175	BMI (K) 3.09573
PDV: [_N_(D) _E	height*heig ERROR_(D) 0	ght); NAME(K)	HEIGHT (K)	• •	
PDV: [EXPLAN	_N_(D) _E 2	height*heig ERROR_(D) 0	ght); NAME(K)	HEIGHT (K)	• •	
PDV: [EXPLAN output; Ex1: NAME Barbara	_N_(D) _E 2 ATION: BMI HEIGHT 61	height *heig RROR_(D) 0 is calculated. WEIGHT	ght); NAME (K) John	HEIGHT (K)	• •	
PDV: [EXPLAN output; Ex1: NAME	_N_ (D) _E 2 <u>ATION:</u> BMI HEIGHT	height*heig RROR_(D) 0 is calculated.	ght); NAME (K) John	HEIGHT (K)	• •	
PDV: [EXPLAN output; Ex1: NAME Barbara John EXPLAN EXPLAN	_N_(D) _E 2 ATION: BMI HEIGHT 61 62 ATION: The	height *heig RROR_(D) 0 is calculated. WEIGHT 175 second observ SAS system re	BMI 31.8678	HEIGHT (K) 62	• •	3.09573

HIRD ITERATION:				
ata ex1;				510 (10)
<u>_N_(D)</u> _ERR0	<u> </u>	HEIGHT (K)	WEIGHT (K)	BMI (K)
PDV: 3	0	•	•	•
EXPLANATION: The auto non-automatic variables ar				

Figure 2c. The third iteration of Program 1.

THE DIFFERENCE BETWEEN READING A RAW DATASET AND READING A SAS DATASET

When creating a SAS dataset based on a raw dataset, SAS sets each variable value in the PDV to missing at the beginning of each iteration of execution, except for the automatic variables, variables that are named in the RETAIN statement, variables created by the SUM statement, data elements in a _TEMPORARY_ array, and variables created in the options of the FILE/INFILE statement.

The dataset that is being created can be referred to as an *output* dataset. The output dataset is the one that follows the keyword DATA. Often the output dataset is created based on the existing SAS dataset instead of from the raw dataset. You can also call the existing SAS dataset that is used to create the output dataset the *input* dataset. The input dataset is the one that follows the keyword SET. When creating a SAS dataset based on an input SAS dataset, SAS sets each variable to missing in the PDV *only* before the first iteration of the execution. Variables will keep (retain) their values in the PDV until they are replaced by the new values from the input dataset. These variables exist in both the input and output datasets. Often when creating a new dataset based on the existing SAS dataset, you will create new variables based on existing variables; these new variables are not from the input dataset. These new variables will be set to missing in the PDV at the beginning of *every* iteration of the execution.

THE RETAIN AND SUM STATEMENTS

Suppose you would like to create a new variable that is based on values from previous observations, such as creating a variable that accumulates the values from other numeric variables. Consider the following SAS dataset, $ex2^{1}$. Based on ex2, suppose you would like to create a new variable, TOTAL, that is used to accumulate the SCORE variable.

Ex2:		
	ID	SCORE
1	A01	3
2	A02	
3	A03	4

In order to create an accumulator variable, TOTAL, you need to set the TOTAL to 0 at the first iteration of the execution. Then at each iteration of the execution, add the value from the SCORE variable to the TOTAL variable. Since TOTAL is a new variable that you want to create, TOTAL will be set to missing in the PDV at the beginning of *every* iteration of the execution. In order to accumulate the TOTAL variable, you need to retain the value of TOTAL at the beginning of each iteration of the execution. In this situation, you need to use the RETAIN statement.

THE RETAIN STATEMENT

The RETAIN statement has the following form:

RETAIN VARIABLE <VALUE>;

VARIABLE is the name of the variable that you will want to retain and VALUE is a numeric value that is used to initialize the VARIABLE *only* at the first iteration of the DATA step execution. If you do not specify an initial value, the retained variable is initialized as missing before the first execution of the DATA step. The RETAIN statement prevents the VARIABLE from being initialized each time the DATA step executes. Here is program to create the TOTAL variable by using the RETAIN statement.

```
Program 3:
data ex2_2;
    set ex2;
    retain total 0;
    total = sum(total, score);
run;
```

The execution phase begins immediately after the completion of the compilation phase. At the beginning of the execution phase, the variables ID and SCORE are set to missing (see Figure 3a); however, the variable TOTAL is initialized to 0 because of the RETAIN statement. Next, the SET statement copies the first observation from the dataset *ex2* to the PDV. The RETAIN statement is a compile-time only statement; it does not execute during the execution phase. Then the variable TOTAL is calculated. Finally, the DATA step execution reaches the final step. Since there is no explicit OUTPUT statement in this program, the implicit OUTPUT statement at the end of the DATA

¹ A SAS file has an extension of "sas7bdat", for example, ex2.sas7bdat. I will not write the extension in this paper for convenience purposes

	RATION							
x2:		<u>-</u>						
ID	SCORE							
A01		3 ← Readir	ng					
A02			-					
A03		4						
lata ex	2_2;							
	N(ID (K)	SCORE (K)	TOTAL (K)	7	
PDV:	1	0			•	0		
		l. N. io initio	lized to (1 and EDDC		to 0 The year	ichles ID and	·7
					R_ is initialized 0 because of the			
set ex	-							
	N (· / 1	ID (K)	SCORE (K)	TOTAL (K)	7	
PDV:	1	0		A01	3	0		
EXPLA	ANATION	I: The SET st	atement	copies the fir	st observation fr	rom <i>ex</i> 2 to the	PDV.]
retain EXPLA	total NATION	0; I: The RETAI	N statem		st observation fr			·
retain EXPLA	total ANATION = sum(t	0; I: The RETAI	N statem	nent does not	execute during	the execution		·
retain EXPLA total	total ANATION = sum(t	0; I: The RETAI cotal, sco D)ERROF	N statem re); R_(D)	nent does not ID (K)	execute during SCORE (K)	the execution]]
retain EXPLA	total ANATION = sum(t	0; I: The RETAI	N statem re); R_(D)	nent does not	execute during	the execution		·]
retain EXPLA total PDV:	total ANATION = sum(t _N_(0; I: The RETAI cotal, sco D)ERROF	N statem re); R_(D)	nent does not ID (K) A01	execute during SCORE (K)	the execution]]
retain EXPL4 total PDV: EXPL4	total ANATION = sum(t _N_(0; : The RETAI :otal, sco: D) _ERROF 0	N statem re); R_(D)	nent does not ID (K) A01	execute during SCORE (K)	the execution		·
retain EXPLA total PDV: EXPLA run;	total ANATION = sum(t _N_(t _1	0; : The RETAI :otal, sco: D) _ERROF 0	N statem re); R_(D)	nent does not ID (K) A01	execute during SCORE (K)	the execution]]
retain EXPLA total PDV: EXPLA run; Ex2_2:	total ANATION = sum(t _N_(1 ANATION	0; :: The RETAI :: total, sco: D) _ERROF 0 :: TOTAL is c	N statem re); R_(D) alculatec	nent does not ID (K) A01	execute during SCORE (K)	the execution		
retain EXPLA total PDV: EXPLA run; Ex2_2: ID	total ANATION = sum(t _N_(1 ANATION	0; I: The RETAI cotal, sco D) _ERROF 0 I: TOTAL is co SCORE	N statem re); R_(D) alculatec	ID (K) A01	execute during SCORE (K)	the execution		
retain EXPLA total PDV: EXPLA run; Ex2_2:	total ANATION = sum(t _N_(1 ANATION	0; :: The RETAI :: total, sco: D) _ERROF 0 :: TOTAL is c	N statem re); R_(D) alculatec	nent does not ID (K) A01	execute during SCORE (K)	the execution		·····
retain EXPLA total PDV: EXPLA run; Ex2_2: ID A01	total ANATION = sum(t _N_(1 ANATION	0; I: The RETAI cotal, sco: D)ERROF 0 I: TOTAL is c SCORE 3 I: The implicit	N statem	ID (K) A01 d. TAL 3 IT statement a	execute during SCORE (K) 3 at the end of the	the execution TOTAL (K) 3		

step tells the SAS system to write observations to the dataset. The SAS system returns to the beginning of the DATA step to begin the second iteration.

At the beginning of the second iteration, since data is read from an existing SAS dataset, instead of reading from the raw dataset, values in the PDV for variables ID and SCORE are retained from the previous iteration. The newly created variable TOTAL is also retained because the RETAIN statement is used. See Figures 3b and 3c for the process of the second and third iterations.

ID	SCORE					
01	3	_				
02		← Reading	g			
03	4					
lata ex) _ERROR		() SCORE (K)	TOTAL (K)	
PDV:	_N_(C		_(D) ID (H A0		<u>3</u>	
		J J				1
EXPLA	NATION	_N_ is increi	mented to 2. ID	and SCORE are reta	ained from the	previous iteration
becaus	e data ar	e read from a		lataset. TOTAL is a		
stateme	ent is use	d.				
set ex2	2; _N_(C) _ERROR	_(D) ID (H	() SCORE (K)	TOTAL (K)	
PDV:	2				<u>3</u>]
ι υ ν.		0	70		5	l
EXPLA	NATION	The SET sta	atement copies th	ne second observation	on from ex2 to	the PDV.
retain	total = sum(to); otal, scor	e);			the PDV.
retain cotal =	total = sum(to _N_(E); otal, scor) _ERROR	re); _(D) ID(H	() SCORE (K)	TOTAL (K)	the PDV.
retain	total = sum(to); otal, scor	e);	() SCORE (K)		the PDV.
PDV: EXPLA	total = sum(to _N_(E _2 NATION	0; ptal, scor) _ERROR 0	e); _(D) ID (P _ A0	() SCORE (K)	TOTAL (K) 3]
PDV: EXPLA the mis	total = sum(to _N_(E _2 NATION	D; Dtal, scor ERROR 0 TOTAL is ca	e); _(D) ID (P _ A0	K) SCORE (K) 2 .	TOTAL (K) 3]
PDV: EXPLA	total = sum(to _N_(E _2 NATION	D; Dtal, scor ERROR 0 TOTAL is ca	e); _(D) ID (P _ A0	K) SCORE (K) 2 .	TOTAL (K) 3]
PDV: EXPLA the mis Ex2_2: ID	total = sum(to _N_(E _2 NATION	D; Dtal, scor ERROR 0 TOTAL is ca e is treated a	e); _(D) ID (F A0 alculated. When t s 0.	K) SCORE (K) 2 .	TOTAL (K) 3]
PDV: EXPLA the mis cun; Ex2_2:	total = sum(to _N_(E _2 NATION	D; Dtal, scor ERROR 0 TOTAL is ca e is treated a SCORE	e); _(D) ID (P A0 alculated. When t s 0. TOTAL	K) SCORE (K) 2 .	TOTAL (K) 3]

THE SUM STATEMENT

The program above can be re-written by using the SUM statement instead of using the RETAIN statement. The SUM statement has the following form:

VARIABLE + EXPRESSION;

The SUM statement may seem unusual because it does not contain the equal sign. VARIABLE is the numeric accumulator variable that is to be created. VARIABLE is automatically set to 0 at the beginning of the first iteration of the DATA step execution and it is retained in following iterations. EXPRESSION is any SAS expression. In the situation where EXPRESSION is evaluated to a missing value, it is treated as 0. Here is an equivalent version of Program 3 using the SUM statement.

```
Program 4:
data ex2_3;
    set ex2;
    total + score;
run;
```

Image: Second state state in the second state state is supported by the previous iterated in the second state state is supported by the previous iterated iterates iterat								
2 . 3 4 exe2_2; $N_(D)$ _ERROR_(D) _ID (K) _SCORE (K) _TOTAL (K) PDV: 3 0 $A02$ 3 EXPLANATION: _N_ is incremented to 3. ID and SCORE are retained from the previous iterat TOTAL is also retained. $A03$ 4 3 et ex2;	ID	SCORE						
3 4 \leftarrow Reading ata ex2_2; N_(D) _ERROR_(D) ID (K) SCORE (K) TOTAL (K) PDV: 3 0 A02 3 EXPLANATION: _N_ is incremented to 3. ID and SCORE are retained from the previous iterat TOTAL is also retained. at ex2; N_(D) _ERROR_(D) ID (K) SCORE (K) TOTAL (K) PDV: 3 0 A03 4 3 EXPLANATION: The SET statement copies the third observation from ex2 to the PDV. etain total 0; o A03 4 7 PDV: 3 0 A03 4 7 EXPLANATION: The SET statement copies the third observation from ex2 to the PDV. etain total 0; o A03 4 7 PDV: 3 0 A03 4 7 PDV: 3 0 A03 4 7 SCORE TOTAL is calculated.	1		3					
ata ex2_2; N_(D)_ERROR_(D) ID (K) SCORE (K) TOTAL (K) PDV: 3 0 A02 3 EXPLANATION: _N_ is incremented to 3. ID and SCORE are retained from the previous iterat TOTAL is also retained. ata ex2; et ex2; _N_(D)_ERROR_(D) ID (K) SCORE (K) TOTAL (K) PDV: 3 0 A03 4 3 EXPLANATION: The SET statement copies the third observation from ex2 to the PDV. etain total 0; o A03 4 7 etain total 0; 0 A03 4 7 EXPLANATION: The SET statement copies the third observation from ex2 to the PDV. ata a 0 A03 4 7 PDV: 3 0 A03 4 7 1	2							
PDV: 3 0 A02 3 EXPLANATION: _N_ is incremented to 3. ID and SCORE are retained from the previous iterat TOTAL is also retained. ID and SCORE are retained from the previous iterat TOTAL is also retained. et ex2;	03		4 ← Readin	g				
N_(D) ERROR_(D) ID (K) SCORE (K) TOTAL (K) PDV: 3 0 A02 . 3 EXPLANATION: N_ is incremented to 3. ID and SCORE are retained from the previous iterat TOTAL is also retained.								
PDV: 3 0 A02 3 EXPLANATION: _N_ is incremented to 3. ID and SCORE are retained from the previous iterat TOTAL is also retained. ID and SCORE are retained from the previous iterat TOTAL is also retained. et ex2;	lata ex			(D)		SCOPE (K)		
EXPLANATION: _N_ is incremented to 3. ID and SCORE are retained from the previous iterat TOTAL is also retained. et ex2; _N_(D) _ERROR_(D) ID (K) SCORE (K) TOTAL (K) PDV: 3 3 0 A03 4 3 0 A03 4 SCORE (K) TOTAL (K) PDV: 3 0 A03 EXPLANATION: The SET statement copies the third observation from ex2 to the PDV. etain total 0; otal = sum(total, score); _N_(D) _ERROR_(D) ID (K) SCORE (K) TOTAL (K) PDV: 3 0 A03 4 7 EXPLANATION: TOTAL is calculated.	י∕וחפ			_ (U)				7
TOTAL is also retained. et ex2; N_(D)_ERROR_(D) ID (K) SCORE (K) TOTAL (K) PDV: 3 0 A03 4 3 EXPLANATION: The SET statement copies the third observation from ex2 to the PDV. etain total 0; otal = sum(total, score); _N_(D)_ERROR_(D) ID (K) SCORE (K) TOTAL (K) PDV: 3 0 A03 4 7 EXPLANATION: TOTAL is calculated. :: ID SCORE TOTAL A03 4 7 EXPLANATION: TOTAL is calculated. :: ID SCORE TOTAL A01 3 3 3 A02	FDV.	5	0		AUZ	•	5	
TOTAL is also retained. et ex2; N_(D)_ERROR_(D) ID (K) SCORE (K) TOTAL (K) PDV: 3 0 A03 4 3 EXPLANATION: The SET statement copies the third observation from ex2 to the PDV. etain total 0; otal = sum(total, score); _N_(D)_ERROR_(D) ID (K) SCORE (K) TOTAL (K) PDV: 3 0 A03 4 7 EXPLANATION: TOTAL is calculated. :: ID SCORE TOTAL A03 4 7 EXPLANATION: TOTAL is calculated. :: ID SCORE TOTAL A01 3 3 3 A02	EXPLA		I: N is incre	mented to	o 3. ID and	SCORE are ret	ained from the	previous iteration
$\frac{N_{0}(D)_{eRROR_{0}}(D) ID_{0}(K) SCORE_{0}(K) TOTAL_{0}(K)}{3 0 A03 4 3}$ EXPLANATION: The SET statement copies the third observation from <i>ex2</i> to the PDV. etain total 0; otal = sum(total, score); $\frac{N_{0}(D)_{eRROR_{0}}(D) ID_{0}(K) SCORE_{0}(K) TOTAL_{0}(K)}{3 0 A03 4 7}$ EXPLANATION: TOTAL is calculated. :; Ex2_2: $\frac{ID SCORE TOTAL}{A01 3 3} \\ A02 A03 4 7$ EXPLANATION: The observation from the PDV is written to the dataset. The SAS system retur the beginning of the DATA step to begin the 4 th iteration.					0 01 12 0.10			
$\frac{N_{0}(D)_{eRROR_{0}}(D) ID_{0}(K) SCORE_{0}(K) TOTAL_{0}(K)}{3 0 A03 4 3}$ EXPLANATION: The SET statement copies the third observation from <i>ex2</i> to the PDV. etain total 0; otal = sum(total, score); $\frac{N_{0}(D)_{eRROR_{0}}(D) ID_{0}(K) SCORE_{0}(K) TOTAL_{0}(K)}{3 0 A03 4 7}$ EXPLANATION: TOTAL is calculated. :; Ex2_2: $\frac{ID SCORE TOTAL}{A01 3 3} \\ A02 A03 4 7$ EXPLANATION: The observation from the PDV is written to the dataset. The SAS system retur the beginning of the DATA step to begin the 4 th iteration.								
PDV: 3 0 A03 4 3 EXPLANATION: The SET statement copies the third observation from ex2 to the PDV. etain total 0; otal = sum(total, score); <u>N_(D)_ERROR_(D)</u> ID (K) SCORE (K) TOTAL (K) PDV: 3 0 A03 4 7 EXPLANATION: TOTAL is calculated. .; Ex2_2: ID SCORE TOTAL A01 3 3 A02 . 3 A03 4 7 EXPLANATION: The observation from the PDV is written to the dataset. The SAS system retur the beginning of the DATA step to begin the 4 th iteration.	set ex2	-						
EXPLANATION: The SET statement copies the third observation from ex2 to the PDV. etain total 0; otal = sum(total, score); $N_(D)$ _ERROR_(D) ID (K) SCORE (K) TOTAL (K) PDV: 3 0 A03 4 7 EXPLANATION: TOTAL is calculated.			, =	_(D)				7
etain total 0; otal = sum(total, score); $N_(D)$ ERROR_(D) ID (K) SCORE (K) TOTAL (K) PDV: 3 0 A03 4 7 EXPLANATION: TOTAL is calculated.	PDV:	3	0		A03	4	3	
etain total 0; otal = sum(total, score); $N_(D)$ ERROR_(D) ID (K) SCORE (K) TOTAL (K) PDV: 3 0 A03 4 7 EXPLANATION: TOTAL is calculated.								
otal = sum(total, score); $N_(D)$ _ERROR_(D) ID (K) SCORE (K) TOTAL (K)PDV:30A0347EXPLANATION: TOTAL is calculated.:;EXPLANATION: TOTAL is calculated.:;EX2_2: ID SCORE TOTAL A01A023A03.4.7EXPLANATION: The observation from the PDV is written to the dataset. The SAS system retur the beginning of the DATA step to begin the 4 th iteration.								
otal = sum(total, score); $N_(D)$ _ERROR_(D) ID (K) SCORE (K) TOTAL (K)PDV:30A0347EXPLANATION: TOTAL is calculated.:;EXPLANATION: TOTAL is calculated.:;EX2_2: ID SCORE TOTAL A01A023A03.4.7EXPLANATION: The observation from the PDV is written to the dataset. The SAS system retur the beginning of the DATA step to begin the 4 th iteration.	EXPLA	NATION	I: The SET sta	atement o	copies the th	ird observation	from ex2 to the	e PDV.
N(D) _ERROR_(D) ID (K) SCORE (K) TOTAL (K) PDV: 3 0 A03 4 7 EXPLANATION: TOTAL is calculated.				atement c	copies the th	ird observation	from ex2 to the	e PDV.
EXPLANATION: TOTAL is calculated. .; Ex2_2: ID SCORE A01 3 A02 . A03 4 TOTAL EXPLANATION: The observation from the PDV is written to the dataset. The SAS system retur the beginning of the DATA step to begin the 4 th iteration.	retain	total	0;		copies the th	ird observation	from <i>ex2</i> to th	e PDV.
Ex2_2: ID SCORE TOTAL A01 3 3 A02 . 3 A03 4 7 EXPLANATION: The observation from the PDV is written to the dataset. The SAS system retur the beginning of the DATA step to begin the 4 th iteration.	retain	total = sum(t	0; cotal, scor	e);				e PDV.
Ex2_2: ID SCORE TOTAL A01 3 3 A02 . 3 A03 4 7 EXPLANATION: The observation from the PDV is written to the dataset. The SAS system retur the beginning of the DATA step to begin the 4 th iteration.	etain	total = sum(t	0; cotal, scor D) _ERROR	e);	ID (K)	SCORE (K)	TOTAL (K)	e PDV.
Ex2_2: ID SCORE TOTAL A01 3 3 A02 . 3 A03 4 7 EXPLANATION: The observation from the PDV is written to the dataset. The SAS system retur the beginning of the DATA step to begin the 4 th iteration.	etain otal =	total = sum(t	0; cotal, scor D) _ERROR	e);	ID (K)	SCORE (K)	TOTAL (K)	e PDV.
Ex2_2: ID SCORE TOTAL A01 3 3 A02 . 3 A03 4 7 EXPLANATION: The observation from the PDV is written to the dataset. The SAS system retur the beginning of the DATA step to begin the 4 th iteration.	retain cotal = PDV:	total = sum(t _N_(3	0; cotal, scor D) _ERROR 0	re); _(D)	ID (К) А03	SCORE (K)	TOTAL (K)	e PDV.
ID SCORE TOTAL A01 3 3 A02 . 3 A03 4 7 EXPLANATION: The observation from the PDV is written to the dataset. The SAS system retur the beginning of the DATA step to begin the 4 th iteration.	PDV:	total = sum(t _N_(3	0; cotal, scor D) _ERROR 0	re); _(D)	ID (К) А03	SCORE (K)	TOTAL (K)	e PDV.
A01 3 3 A02 . . 3 A03 . . . EXPLANATION: The observation from the PDV is written to the dataset. The SAS system retur the beginning of the DATA step to begin the 4 th iteration.	PDV: EXPLA	total = sum(t _N_(3	0; cotal, scor D) _ERROR 0	re); _(D)	ID (К) А03	SCORE (K)	TOTAL (K)	e PDV.
A02 .	PDV: EXPLA	total = sum(t _ <u>N_(</u> _3	0; cotal, scor D) _ERROR 0 	e); _(D) culated.	ID (K) A03	SCORE (K)	TOTAL (K)	e PDV.
A03 4 7 EXPLANATION: The observation from the PDV is written to the dataset. The SAS system retur the beginning of the DATA step to begin the 4 th iteration.	PDV: EXPLA Ex2_2: ID	total = sum(t _ <u>N_(</u> _3	0; cotal, scor D) _ERROR 0 I: TOTAL is ca SCORE	e); _(D) culated.	ID (K) A03	SCORE (K)	TOTAL (K)	e PDV.
EXPLANATION: The observation from the PDV is written to the dataset. The SAS system retur the beginning of the DATA step to begin the 4 th iteration.	PDV: EXPLA b; Ex2_2: ID A01	total = sum(t _ <u>N_(</u> _3	0; cotal, scor D) _ERROR 0 I: TOTAL is ca SCORE	e); _(D) culated.	ID (K) A03	SCORE (K)	TOTAL (K)	e PDV.
the beginning of the DATA step to begin the 4 th iteration.	PDV: EXPLA F; Ex2_2: ID A01 A02	total = sum(t _ <u>N_(</u> _3	0; cotal, scor D) _ERROR 0 : TOTAL is ca SCORE 3	e); _(D) culated.	ID (K) A03	SCORE (K)	TOTAL (K)	e PDV.
EXPLANATION: The observation from the PDV is written to the dataset. The SAS system retur the beginning of the DATA step to begin the 4 th iteration.	PDV: EXPLA	total = sum(t _N_(3	0; cotal, scor D) _ERROR 0	re); _(D)	ID (К) А03	SCORE (K)	TOTAL (K)	e PDV.
the beginning of the DATA step to begin the 4 th iteration.	PDV: EXPLA ; Ex2_2: ID A01 A02	total = sum(t _ <u>N_(</u> _3	0; cotal, scor D) _ERROR 0 : TOTAL is ca SCORE 3	e); _(D) culated.	ID (K) A03	SCORE (K)	TOTAL (K)	e PDV.
	etain otal = PDV: EXPLA ; Ex2_2: ID A01 A02	total = sum(t _ <u>N_(</u> _3	0; cotal, scor D) _ERROR 0 : TOTAL is ca SCORE 3	e); _(D) culated.	ID (K) A03	SCORE (K)	TOTAL (K)	e PDV.
	Expla PDV: EXPLA ; Ex2_2: ID A01 A02 A03 EXPLA	total = sum(t N_(3	0; total, scor D) _ERROR 0 1: TOTAL is ca SCORE 3 4 J: The observa	e); _(D) alculated. TOT	ID (K) A03 AL 3 3 7	SCORE (K)	<u>TOTAL (K)</u> 7]
$h = 4h = 4\tilde{W}$ and $\theta = 1$ is an interval of $h = 100000000000000000000000000000000000$	ExPLA EXPLA F Ex2_2: ID A01 A02 A03 EXPLA	total = sum(t N_(3	0; total, scor D) _ERROR 0 1: TOTAL is ca SCORE 3 4 J: The observa	e); _(D) alculated. TOT	ID (K) A03 AL 3 3 7	SCORE (K)	<u>TOTAL (K)</u> 7]
In the 4 th and final iteration, there are no more observations to read; the SAS system goes to th next DATA or PROC step.	PDV: EXPLA r; Ex2_2: ID A01 A02 A03 EXPLA the beg	total sum(t <u>N_(</u> 3 ANATION ginning c	0; otal, scor D) _ERROR 0 1: TOTAL is can 3 SCORE 3 4 4 1: The observations of the DATA states of	e); _(D) alculated. TOT ation from ep to beg	ID (K) A03 AU AU AU AU AU AU AU AU AU AU AU AU AU	SCORE (K)	TOTAL (K) 7] AS system returns

Figure 3c. The third iteration of Program 3.

THE SUBSETTING IF STATEMENT

You can use the IF statement to continue processing only the observations that meet the condition of the specified expression. The IF statement has the following form:

IF EXPRESSION;

The EXPRESSION in the IF statement can be any SAS expression. If the EXPRESSION is true for the observation, SAS continues to execute statements in the DATA step and includes the current observation in the data set. The resulting SAS data set contains a subset of the external file or SAS data set.

On the other hand, if the EXPRESSION is false, then no further statements are processed for that observation and SAS immediately returns to the beginning of the DATA step. That is to say, the remaining program statements in the DATA step are not executed and the current observation is not written to the output data set. An example will be shown in the following section.

THE BY-GROUP PROCESSING IN THE DATA STEP

The examples that have been presented so far only contain one observation per subject. Sometimes you will also deal with data with multiple observations per subject. This type of data often results from repeated measures for each subject and is often called longitudinal data. For longitudinal data, sometimes it is useful to be able to identify the beginning or end of measurement for each subject. This can be accomplished by using the BY-group processing method. SAS locates the beginning and end of a BY-group by creating two temporary indicator variables for each BY variable: FIRST.VARIABLE and LAST.VARIABLE. For example, suppose that the ID variable is the BY variable; FIRST.ID and LAST.ID will be created. When FIRST.ID equals 1, SAS reads the first observation in an ID group. Similarly, LAST.ID equals 1 indicates SAS reads the last observation in an ID group.

Since FIRST.VARIABLE and LAST.VARIABLE are temporary variables, they are not being output to the output dataset. Ron Cody's book Longitudinal Data and SAS[®] – A Programmer's Guide provides practical examples for managing and/or manipulating this type of data. Some of the examples in this paper are adapted from his book.

Consider the following SAS dataset, ex3. Each subject has multiple numbers of observations. The number of observations for each subject differs. Suppose that you would like to calculate the total scores for each subject; see Program 5 below. In this program, the SUM statement is used to accumulate the total score. For this problem, the accumulation is completed for each subject, not for all of the subjects combined. Solving this problem can be achieved in three steps: initializing the TOTAL to 0 when starting to read the first observation of each subject; creating TOTAL by accumulating SCORE for each subject; outputting the TOTAL score when reading the last observation of each subject. Therefore, utilize the BY-group processing and the ID will be the BY variable. To use the BY-group processing in the DATA step, It is important to sort the data by the BY variable (ID) first.

Ev2.	
LAU.	

	ID	SCORE
1	A01	3
2	A01	4
3	A01	2
4	A02	4
5	A02	2

Program 5:

```
proc sort data=ex3;
    by id;
run;
data ex3_1 (drop=score);
    set ex3;
    by id;
    if first.id = 1 then total = 0;
    total + score;
    if last.id = 1;
run;
```

Since the BY statement was used after the SET statement in the DATA step, two automatic variables, FIRST.ID and LAST.ID, are created in the PDV. Both FIRST.ID and LAST.ID are initialized to 1 before the fist iteration of the DATA step execution (see Figure 4a). ID and SCORE variables are set to missing, but TOTAL is set to 0 since TOTAL is created by the SUM statement. When the SET statement executes, SAS copies the first observation from the sorted ex3 to the PDV. Since this is the first observation for the A01 subject, FIRST.ID is set to 1. The LAST.ID is set to 0 since this is not the last observation. Next, TOTAL is assigned to 0 because this is the first observation for ID A01 (SAS statement: if first.id = 1 then total = 0). The SUM statement accumulates the TOTAL variable. Because the subsetting IF statement is evaluated to be false (LAST.ID does not equal 1), SAS immediately returns to the beginning of the DATA step. That means the contents of the PDV are not output to the SAS dataset ex3_1.

The second iteration (see Figure 4b) is similar to the first iteration. During this iteration, both FIRST.ID and LAST.ID are set to 0. TOTAL is then accumulated. But the PDV contents are not outputted to the SAS dataset either since this is not the last observation for A01.

In the third iteration (see Figure 4c), FIRST.ID is set to 0 and LAST.ID is set to 1. TOTAL is accumulated. The subsetting IF statement is evaluated to be true. Then SAS reaches the end of the DATA step and the implicit OUTPUT statement copies the contents from the PDV (ID and TOTAL) to the SAS dataset $ex3_1$ (SCORE variable is dropped in the DATA statement). The rest of the iterations are similar to the iterations explained above. See Figures 4d- 4e for details.

:3: ID	ERATION: SCORE						
A01	3CORE	← Reading					
401	3	← Reading					
A01	4						
.02	4						
.02	2						
.02	-						
ata ex	3_1;						
	N (D)	_ERROR_ (I		LAST.ID (D)	ID (K)	SCORE (D)	
PDV:	1	0	1	1		•	0
			and LAST.ID are t to 0 since TOTA				ables are
361 10 1	missing, but	TOTAL 13 36		IL 13 Cleated by			
set ex	:3;						
							TOTAL (1/)
	N(D)	_ERROR_ (I	, , ,	LAST.ID (D)	ID (K)	SCORE (D)	TOTAL (K)
	ANATION: S	0 AS copies th	e first observation	0 from the sorte	A01 d ex3 to the	3 PDV. Since t	0 this is the
EXPLA first ob last ob	ANATION: S oservation fo oservation fo	AS copies th r the A01 sub r A01.	l 1 e first observation oject, FIRST.ID is	0 from the sorte set to 1. The L	A01 d ex3 to the AST.ID is s	PDV. Since thet to 0 since the	0 this is the is is not the
EXPLA first ob last ob	ANATION: S oservation fo	AS copies th r the A01 sub r A01.	i first observation oject, FIRST.ID is	0 from the sorte	A01 d ex3 to the	3 PDV. Since t	0 this is the
EXPLA first ob last ob by id; if fir PDV: EXPLA	ANATION: Soservation for servation for \mathbf{r}_{1} and \mathbf{r}_{2} and \mathbf{r}_{3} and \mathbf{r}_{4}	AS copies th r the A01 sub r A01. then tota _ERROR_ (I	1 e first observation oject, FIRST.ID is a1 = 0; D) FIRST.ID (D) 1 gned to 0.	0 from the sorte set to 1. The L LAST.ID (D) 0	A01 d ex3 to the AST.ID is so ID (K) A01	PDV. Since t et to 0 since th SCORE (D)	0 this is the is is not the TOTAL (K) 0
EXPLA first ob last ob by id; if fir PDV: EXPLA	ANATION: S poservation for st.id = 1 $\underline{N}(D)$ ANATION: T + score;	AS copies th r the A01 sub r A01. then tota _ERROR_ (I 0 OTAL is assi	1 e first observation oject, FIRST.ID is a1 = 0; D) FIRST.ID (D) 1 gned to 0.	0 from the sorte set to 1. The L LAST.ID (D)	A01 d ex3 to the AST.ID is si	3 PDV. Since thet to 0 since the SCORE (D) 3	0 this is the is is not the TOTAL (K)
EXPLA first ob last ob by id; if fir PDV: EXPLA total PDV:	ANATION: S Servation for Sst.id = 1 $\underline{N}(D)$ ANATION: T ANATION: T + score; $\underline{N}(D)$ 1	AS copies th if the A01 sub r A01. then tota ERROR_(I OTAL is assis _ERROR_(I	1 e first observatior oject, FIRST.ID is a1 = 0; D) FIRST.ID (D) 1 gned to 0. D) FIRST.ID (D) 1	0 from the sorte set to 1. The L LAST.ID (D) 0 LAST.ID (D)	A01 d ex3 to the AST.ID is so ID (K) A01 ID (K)	3 PDV. Since thet to 0 since the SCORE (D) 3 SCORE (D)	0 this is the is is not the TOTAL (K) TOTAL (K)

Figure 4a. First iteration for Program 5.

	ITERATION	<u>:</u>					
3: ID	SCORE						
A01	3						
401	4	← Reading					
401		Circading					
402	4						
402	2						
1							
data (ex3_1;						
	N(D)	_ERROR_ (D)	FIRST.ID (D)	LAST.ID (D)	ID (K) A01	SCORE (D)	<u>TOTAL (K)</u> 3
PDV:	2	0	I	0	AUT	3	3
EXPL	ANATION	N_ is increment	red to 2 The va	lues for the res	t of the vari	ables are retai	ned
L <u>=:::</u>							
set ex	· · · · · ·						
	N(D)	_ERROR_ (D)	FIRST.ID (D)	LAST.ID (D)	ID (K)	SCORE (D)	TOTAL (K)
	ANATION: S	0 SAS copies the s	0 second observation	0 tion from the so	A01	4 the PDV. Sind	3 ce this is
EXPL/ not the by id;	ANATION: S	0 GAS copies the s vation or the last	0 second observa observation for	0 tion from the so	A01	4 the PDV. Sind	3 ce this is
EXPL/ not the by id;	ANATION: S	0 SAS copies the s	0 second observa observation for	0 tion from the so	A01	4 the PDV. Sind	3 ce this is
EXPL/ not the by id;	ANATION: S e first observ	0 SAS copies the s vation or the last L then total	0 second observa observation for = 0;	0 tion from the so A01, both FIRS	A01 orted ex3 to ST.ID and L	4 the PDV. Sin AST.ID are se	3 ce this is t to 0.
EXPL/ not the by id; if fir PDV: EXPL/	ANATION: S e first observer st.id = 1 _N_(D) _2 ANATION: F	0 SAS copies the s vation or the last L then total _ERROR_(D)	0 second observation for observation for = 0; FIRST.ID (D) 0	0 tion from the so A01, both FIRS LAST.ID (D) 0	A01 orted ex3 to ST.ID and L ID (K)	4 the PDV. Sind AST.ID are se SCORE (D)	3 ce this is t to 0. TOTAL (K)
EXPL/ not the by id; if fir PDV: EXPL/	ANATION: S e first observ st.id = 1 _N_(D) 2	0 SAS copies the s vation or the last L then total _ERROR_(D) 0	0 second observation for observation for = 0; FIRST.ID (D) 0 there is no exe	0 tion from the so A01, both FIRS LAST.ID (D) 0	A01 wrted ex3 to ST.ID and L ID (K) A01	4 the PDV. Sind AST.ID are se SCORE (D) 4	3 ce this is t to 0. TOTAL (K) 3
EXPL/ not the by id; if fir PDV: EXPL/	ANATION: S e first observ st.id = 1 _N_(D) 2 ANATION: F + score;	0 SAS copies the s vation or the last L then total _ERROR_(D) 0 FIRST.ID ≠ 1, so	0 second observation for observation for = 0; FIRST.ID (D) 0	0 tion from the so A01, both FIRS LAST.ID (D) 0	A01 orted ex3 to ST.ID and L ID (K)	4 the PDV. Sind AST.ID are se SCORE (D)	3 ce this is t to 0. TOTAL (K)
EXPL/ not the by id; if fir PDV: EXPL/ total PDV:	ANATION: S e first observ sst.id = 1 N_(D) 2 ANATION: F + score; N_(D) 2	0 SAS copies the s vation or the last L then total _ERROR_(D) 0 TIRST.ID ≠ 1, so _ERROR_(D) 0	0 second observation for = 0; FIRST.ID (D) 0 there is no exe FIRST.ID (D) 0	0 tion from the so A01, both FIRS LAST.ID (D) 0 cution.	A01 wrted ex3 to ST.ID and L ID (K) A01 ID (K)	4 the PDV. Sind AST.ID are se SCORE (D) 4 SCORE (D)	3 ce this is t to 0. TOTAL (K) 3 TOTAL (K)
EXPL/ not the by id; if fir PDV: EXPL/ total PDV:	ANATION: S e first observ sst.id = 1 N_(D) 2 ANATION: F + score; N_(D) 2	0 SAS copies the s vation or the last L then total _ERROR_(D) 0 FIRST.ID ≠ 1, so _ERROR_(D)	0 second observation for = 0; FIRST.ID (D) 0 there is no exe FIRST.ID (D) 0	0 tion from the so A01, both FIRS LAST.ID (D) 0 cution.	A01 wrted ex3 to ST.ID and L ID (K) A01 ID (K)	4 the PDV. Sind AST.ID are se SCORE (D) 4 SCORE (D)	3 ce this is t to 0. TOTAL (K) 3 TOTAL (K)
EXPL/ not the by id; if fir PDV: EXPL/ total PDV: EXPL/	ANATION: S e first observ sst.id = 1 N_(D) 2 ANATION: F + score; N_(D) 2	0 SAS copies the s vation or the last L then total _ERROR_(D) 0 FIRST.ID ≠ 1, so _ERROR_(D) 0	0 second observation for = 0; FIRST.ID (D) 0 there is no exe FIRST.ID (D) 0	0 tion from the so A01, both FIRS LAST.ID (D) 0 cution.	A01 wrted ex3 to ST.ID and L ID (K) A01 ID (K)	4 the PDV. Sind AST.ID are se SCORE (D) 4 SCORE (D)	3 ce this is t to 0. TOTAL (K) 3 TOTAL (K)
EXPL/ not the by id; if fir PDV: EXPL/ total PDV: EXPL/ if las	ANATION: S e first observer st.id = 1 _N_(D) _2 ANATION: F + score; _N_(D) _2 ANATION: 1 st.id = 1;	0 CAS copies the solution or the last L then total _ERROR_(D) 0 IRST.ID ≠ 1, so _ERROR_(D) 0 TOTAL is calcula	0 second observation for observation for = 0; FIRST.ID (D) 0 there is no exe FIRST.ID (D) 0	0 tion from the so A01, both FIRS LAST.ID (D) 0 cution.	A01 wrted ex3 to ST.ID and L ID (K) A01 ID (K) A01	4 the PDV. Sinc AST.ID are se SCORE (D) 4 SCORE (D) 4	3 ce this is t to 0. TOTAL (K) 3 TOTAL (K) 7
EXPL/ not the by id; if fir PDV: EXPL/ total PDV: EXPL/ if las	ANATION: S e first observer st.id = 1 _N_(D) 2 ANATION: F + score; _N_(D) 2 ANATION: 1 st.id = 1; ANATION: 1	0 SAS copies the s vation or the last L then total _ERROR_(D) 0 FIRST.ID ≠ 1, so _ERROR_(D) 0	0 second observation for observation for = 0; FIRST.ID (D) 0 there is no exe FIRST.ID (D) 0 ated	0 tion from the so A01, both FIRS LAST.ID (D) 0 cution. LAST.ID (D) 0	A01 wrted ex3 to ST.ID and L ID (K) A01 ID (K) A01 FALSE beca	4 the PDV. Sinc AST.ID are se SCORE (D) 4 SCORE (D) 4 ause LAST.ID	3 ce this is t to 0. TOTAL (K) 3 TOTAL (K) 7

Figure 4b. Second iteration for Program 5.

HIRD ITE	ERATION:						
ID	SCORE						
A01	3						
A01 A01	4	← Reading					
A02	4	(Redding					
A02	2						
data	ex3_1;						
	N(D)	_ERROR_ (D)	FIRST.ID (D)	LAST.ID (D)	ID (K)	SCORE (D)	TOTAL (K)
PDV:	3	0	0	0	A01	4	7
EXPL	ANATION: _	N_ is increment	ed to 3. The va	alues for the res	t of the vari	ables are retai	ned.
set ex	3.						
bee en	N_ (D)	_ERROR_ (D)	FIRST.ID (D)	LAST.ID (D)	ID (K)	SCORE (D)	TOTAL (K)
PDV:	3	0	0	1	A01	2	7
by id;	st.id = 1	then total					
PDV:	_N_(D)	_ERROR_ (D)	FIRST.ID (D)	LAST.ID (D)	ID (K) A01	SCORE (D)	TOTAL (K) 7
L	ANATION: F + score; _N_(D)	IRST.ID ≠ 1, so _ERROR_ (D)	there is no exe	cution.	ID (K)	SCORE (D)	TOTAL (K)
PDV:	3	0	0	1	A01	2	9
if las	st.id = 1;	OTAL is calcula		valuated to be	TRUE since	LAST.ID equa	als 1.
run; Ex3_1: II A01		TOTAL 9					
and To are ma	OTAL from the	AS reaches the he PDV to the d)). The SAS sy re 4d)	ataset ex3_1.	The remaining v	variables ar	e not copied si	nce they

Figure 4c. Third iteration for Program 5.

$\frac{N}{ D } = \frac{SCORE}{A01}$ $\frac{N}{A01} = \frac{A01}{4}$ $\frac{A01}{A02} = \frac{A02}{4}$ $\frac{A02}{A02} = \frac{A02}{4}$ $\frac{A02}{4} = \frac{A02}{4}$ $\frac{A02}{4} = \frac{A01}{2} = \frac{A02}{4}$ $\frac{A02}{4} = \frac{A01}{2} = \frac{A02}{4}$ $\frac{A02}{4} = \frac{A02}{4}$ $\frac{A02}{4$	Ex3: ID		-					
A013A014A012A024A022data ex3_1; $N_(D)$ _ERROR_(D) FIRST.ID (D) LAST.ID (D) ID (K) SCORE (D) TOTAL (K)PDV:4001A0129EXPLANATION: N_ is incremented to 4. The values for the remaining variables are retained.set ex3; $N_(D)$ _ERROR_(D) FIRST.ID (D) LAST.ID (D) ID (K) SCORE (D) TOTAL (K)PDV:4010A0249EXPLANATION: SAS copies the fourth observation from the sorted ex3 to the PDV. FIRST.ID is setto 1 since this is the first observation for A02. However, LAST.ID is set to 0.by id;if first.id = 1 then total = 0; $N_(D)$ _ERROR_(D) FIRST.ID (D) LAST.ID (D) ID (K) SCORE (D) TOTAL (K)PDV:4010A024010A024010A024010A024010A024010A024010A024010A024010A024010A024010A02401040	ID							
A014A012A024A022data ex3_1; PDV: $\mathbf{N}_{(D)}$ ERROR_(D) FIRST.ID (D) LAST.ID (D) ID (K) SCORE (D) TOTAL (K) PDV:PDV:4001A0129EXPLANATION: N_ is incremented to 4. The values for the remaining variables are retained.set ex3; PDV: $\mathbf{N}_{(D)}$ ERROR_(D) FIRST.ID (D) LAST.ID (D) ID (K) SCORE (D) TOTAL (K) 		SCORE						
A012A024A022data ex3_1; $=N_(D)$ _ERROR_(D) FIRST.ID (D) LAST.ID (D) ID (K) SCORE (D) TOTAL (K) $PDV:$ PDV:4001A0129EXPLANATION: _N_ is incremented to 4. The values for the remaining variables are retained.set ex3; $=N_(D)$ _ERROR_(D) FIRST.ID (D) LAST.ID (D) ID (K) SCORE (D) TOTAL (K) PDV:PDV:4010A0249EXPLANATION: SAS copies the fourth observation from the sorted ex3 to the PDV. FIRST.ID is set to 1 since this is the first observation for A02. However, LAST.ID is set to 0.by id; iff first.id = 1 then total = 0; $=N_(D)$ _ERROR_(D) FIRST.ID (D) LAST.ID (D) ID (K) SCORE (D) TOTAL (K) PDV:PDV:4010A024010A024010A024010A024010A0240104010A024010A024010A024010A024010401040104040404	A01	3						
A024A022dataex3_1; $N_(D)$ ERROR_(D)FIRST.ID (D)LAST.ID (D)ID (K)SCORE (D)TOTAL (K)PDV:4001A0129EXPLANATION:N_ is incremented to 4. The values for the remaining variables are retained.set ex3;N_(D)ERROR_(D)FIRST.ID (D)LAST.ID (D)ID (K)SCORE (D)TOTAL (K)PDV:4010A0249EXPLANATION: SAS copies the fourth observation from the sorted ex3 to the PDV.FIRST.ID is set to 1 since this is the first observation for A02. However, LAST.ID is set to 0.by id;if first.id = 1 then total = 0;N_(D)ERROR_(D)FIRST.ID (D)LAST.ID (D)ID (K)SCORE (D)TOTAL (K)PDV:4010A02400it first.id = 1 then total = 0;N_(D)ERROR_(D)FIRST.ID (D)LAST.ID (D)ID (K)SCORE (D)TOTAL (K)PDV:4010A0240itotal + score;N_(D)ERROR_(D)FIRST.ID (D)LAST.ID (D)ID (K)SCORE (D)TOTAL (K)PDV:4010A0244	A01	4						
A022dataex3_1; $N_{-}(D)$ _ERROR_(D) FIRST.ID (D) LAST.ID (D) ID (K) SCORE (D) TOTAL (K)PDV:4001A0129EXPLANATION: _N_ is incremented to 4. The values for the remaining variables are retained.set ex3; PDV: $N_{-}(D)$ _ERROR_(D) FIRST.ID (D) LAST.ID (D) ID (K) SCORE (D) TOTAL (K)PDV:4010A0249EXPLANATION: SAS copies the fourth observation from the sorted ex3 to the PDV. FIRST.ID is set to 1 since this is the first observation for A02. However, LAST.ID is set to 0.by id; if first.id = 1 then total = 0; $N_{-}(D)$ _ERROR_(D) FIRST.ID (D) LAST.ID (D) ID (K) SCORE (D) TOTAL (K) PDV:4010A0240total + score; $N_{-}(D)$ _ERROR_(D) FIRST.ID (D) LAST.ID (D) ID (K) SCORE (D) TOTAL (K) PDV:4010A0240	A01	2						
dataex3_1; N_(D) _ERROR_(D) FIRST.ID (D) LAST.ID (D) ID (K) SCORE (D) TOTAL (K) PDV:PDV:4001A0129[EXPLANATION: _N_ is incremented to 4. The values for the remaining variables are retained.set ex3; PDV: $N_(D)$ _ERROR_(D) FIRST.ID (D) LAST.ID (D) ID (K) SCORE (D) TOTAL (K) PDV:4010A0249[EXPLANATION: SAS copies the fourth observation from the sorted ex3 to the PDV. FIRST.ID is set to 1 since this is the first observation for A02. However, LAST.ID is set to 0.by id; if first.id = 1 then total = 0; $N_(D)$ _ERROR_(D) FIRST.ID (D) LAST.ID (D) ID (K) SCORE (D) TOTAL (K) PDV:4010A0240total + score; N_(D) _ERROR_(D) FIRST.ID (D) LAST.ID (D) ID (K) SCORE (D) TOTAL (K) PDV:4010A0244	A02	4	← Reading					
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	A02	2						
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $								
PDV:4001A0129EXPLANATION:N_ is incremented to 4. The values for the remaining variables are retained.set ex3; $N_(D)$ ERROR_(D)FIRST.ID (D)LAST.ID (D)ID (K)SCORE (D)TOTAL (K)PDV:4010A0249EXPLANATION:SAS copies the fourth observation from the sorted ex3 to the PDV.FIRST.ID is setto 1 since this is the first observation for A02.However, LAST.ID is set to 0.by id;if first.id = 1 then total = 0; $N_(D)$ ERROR_(D)FIRST.ID (D)LAST.ID (D)ID (K)PDV:4010A024total + score; $N_(D)$ ERROR_(D)FIRST.ID (D)LAST.ID (D)ID (K)SCORE (D)TOTAL (K)PDV:4010A0244	data	ex3_1;						
EXPLANATION: N is incremented to 4. The values for the remaining variables are retained. set ex3; N (D) ERROR (D) FIRST.ID (D) LAST.ID (D) ID (K) SCORE (D) TOTAL (K) PDV: 4 0 1 0 A02 4 9 EXPLANATION: SAS copies the fourth observation from the sorted ex3 to the PDV. FIRST.ID is set to 1 since this is the first observation for A02. However, LAST.ID is set to 0. by id; if first.id = 1 then total = 0; N (D) ERROR (D) FIRST.ID (D) LAST.ID (D) ID (K) SCORE (D) TOTAL (K) PDV: 4 0 1 0 A02 4 0 EXPLANATION: Since FIRST.ID = 1, TOTAL is set to 0. 1 0 A02 4 0 total + score; N (D) ERROR (D) FIRST.ID (D) LAST.ID (D) ID (K) SCORE (D) TOTAL (K) PDV: 4 0 1 0 A02 4 4								
set ex3; $N_(D)$ ERROR_(D)FIRST.ID (D)LAST.ID (D)ID (K)SCORE (D)TOTAL (K)PDV:4010A0249EXPLANATION: SAS copies the fourth observation from the sorted ex3 to the PDV. FIRST.ID is set to 1 since this is the first observation for A02. However, LAST.ID is set to 0.by id; if first.id = 1 then total = 0; $N_(D)$ $N_(D)$ $ERROR_(D)$ $FIRST.ID (D)$ $LAST.ID (D)$ $ID (K)$ $SCORE (D)$ $TOTAL (K)$ PDV:4010A0240total + score; $N_(D)$ $N_(D)$ $ERROR_(D)$ $FIRST.ID (D)$ $LAST.ID (D)$ $ID (K)$ $SCORE (D)$ $TOTAL (K)$ PDV:4010A0244	PDV:	4	0	0	1	A01	2	9
set ex3; $N_(D)$ ERROR_(D)FIRST.ID (D)LAST.ID (D)ID (K)SCORE (D)TOTAL (K)PDV:4010A0249EXPLANATION: SAS copies the fourth observation from the sorted ex3 to the PDV. FIRST.ID is set to 1 since this is the first observation for A02. However, LAST.ID is set to 0.by id; if first.id = 1 then total = 0; $N_(D)$ $N_(D)$ $ERROR_(D)$ $FIRST.ID (D)$ $LAST.ID (D)$ $ID (K)$ $SCORE (D)$ $TOTAL (K)$ PDV:4010A0240total + score; $N_(D)$ $N_(D)$ $ERROR_(D)$ $FIRST.ID (D)$ $LAST.ID (D)$ $ID (K)$ $SCORE (D)$ $TOTAL (K)$ PDV:4010A0244								
$\begin{array}{c c c c c c c c c c c c c c c c c c c $	EXPL	ANATION: _	in_ is increment	ted to 4. The va	liues for the ren	naining varia	ables are retail	nea.
PDV:4010A0249EXPLANATION: SAS copies the fourth observation from the sorted ex3 to the PDV. FIRST.ID is set to 1 since this is the first observation for A02. However, LAST.ID is set to 0.by id; if first.id = 1 then total = 0; $N_{-}(D)$ _ERROR_(D) FIRST.ID (D) LAST.ID (D) ID (K) SCORE (D) TOTAL (K) PDV:PDV:4010A02A0240EXPLANATION: Since FIRST.ID = 1, TOTAL is set to 0.total + score; $N_{-}(D)$ _ERROR_(D) FIRST.ID (D) LAST.ID (D) ID (K) SCORE (D) TOTAL (K) PDV:PDV:4010A024010A02440100A024010A0244010A0244010A0244010A0244010A0244010A0244010A0244010A0244010A024441404040404040	set ex	3;						
EXPLANATION: SAS copies the fourth observation from the sorted ex3 to the PDV. FIRST.ID is set to 1 since this is the first observation for A02. However, LAST.ID is set to 0.by id; if first.id = 1 then total = 0; $N_(D)$ _ERROR_(D) FIRST.ID (D) LAST.ID (D) ID (K) SCORE (D) TOTAL (K) PDV: 4 0 1 0 A02 4 0PDV: 4 0 1Colspan="2">COLSPan="2">COLSPan="2">COLSPan="2">COLSPan="2">COLSPan="2">COLSPan="2">COLSPan="2">COLSPan="2">COLSPan="2">COLSPan="2">COLSPan="2">COLSPan="2">COLSPan="2">COLSPan="2">COLSPan= 2000by id; if first.id = 1 then total = 0; $N_(D)$ _ERROR_(D) FIRST.ID (D) LAST.ID (D) ID (K) SCORE (D) TOTAL (K) PDV: 4 0 1PDV: 4 0Colspan="2">COLSPan= 2000 COLSPan=		-	_ERROR_ (D)	FIRST.ID (D)	LAST.ID (D)	ID (K)	SCORE (D)	TOTAL (K)
to 1 since this is the first observation for A02. However, LAST.ID is set to 0.by id;if first.id = 1 then total = 0; $N_{(D)}$ _ERROR_(D) FIRST.ID (D) LAST.ID (D) ID (K) SCORE (D) TOTAL (K)PDV:4010A0240EXPLANATION: Since FIRST.ID = 1, TOTAL is set to 0.total + score;N_(D) _ERROR_(D) FIRST.ID (D) LAST.ID (D) ID (K) SCORE (D) TOTAL (K)PDV:4010A0240	PDV:	4	0	1	0	A02	4	9
PDV: 4 0 1 0 A02 4 0 EXPLANATION: Since FIRST.ID = 1, TOTAL is set to 0. Image: Constraint of the set to 0. total + score;	to 1 SI	ince this is th	ie first observati	on for AUZ. Ho	wever, LAST.IL	J is set to U .		
EXPLANATION: Since FIRST.ID = 1, TOTAL is set to 0. total + score; N_(D)ERROR_(D) FIRST.ID (D) LAST.ID (D) ID (K) SCORE (D) TOTAL (K) PDV: 4 0 1 0 A02 4 4		st.id = 1						j
total + score;	if fir	st.id = 1 N_(D)	_ERROR_ (D)	FIRST.ID (D)		ID (K)		
N(D) _ERROR_(D) FIRST.ID (D) LAST.ID (D) ID (K) SCORE (D) TOTAL (K) PDV: 4 0 1 0 A02 4 4	if fir	st.id = 1 N_(D)	_ERROR_ (D)	FIRST.ID (D)		ID (K)		
PDV: 4 0 1 0 A02 4 4	if fir PDV:	rst.id = 1 _N_(D) _4	_ ERROR_ (D) 0	FIRST.ID (D) 1	0	ID (K)		
	if fir PDV:	<pre>st.id = 1N_(D)4</pre>	_ERROR_ (D) 0	FIRST.ID (D) 1 = 1, TOTAL is s	0 et to 0.	ID (К) А02	4	0
EXPLANATION: TOTAL is calculated.	if fir PDV: EXPL total	ANATION: S - score; - N_ (D) - N_ (D) - score;	_ERROR_ (D) 0 Since FIRST.ID	FIRST.ID (D) 1 = 1, TOTAL is s FIRST.ID (D)	0 et to 0. LAST.ID (D)	ID (K) A02 ID (K)	4 SCORE (D)	0
	if fir PDV: EXPL total	ANATION: S - score; - N_ (D) - N_ (D) - score;	_ERROR_ (D) 0 Since FIRST.ID	FIRST.ID (D) 1 = 1, TOTAL is s FIRST.ID (D)	0 et to 0. LAST.ID (D)	ID (K) A02 ID (K)	4 SCORE (D)	0
if last.id = 1;	if fir PDV: EXPL total PDV:	<pre>st.id = 1N_(D)4</pre>	_ERROR_ (D) 0 Since FIRST.ID ERROR_ (D) 0	FIRST.ID (D) 1 = 1, TOTAL is s FIRST.ID (D) 1	0 et to 0. LAST.ID (D)	ID (K) A02 ID (K)	4 SCORE (D)	0
EXPLANATION: The subsetting IF statement is evaluated to be FALSE because LAST.ID \neq 1.	if fir PDV: EXPL total PDV: EXPL	ANATION: S - <u>N_(D)</u> 4 ANATION: S + score; <u>N_(D)</u> 4 ANATION: T	_ERROR_ (D) 0 Since FIRST.ID _ERROR_ (D) 0 OTAL is calcula	FIRST.ID (D) 1 = 1, TOTAL is s FIRST.ID (D) 1	0 et to 0. LAST.ID (D)	ID (K) A02 ID (K)	4 SCORE (D)	0
SAS returns to the beginning of the DATA step to begin the 5^{th} iteration (see Figure 4e)	if fir PDV: EXPL total PDV: EXPL if las	<pre>st.id = 1 _N_(D) 4 ANATION: S + score; _N_(D) 4 ANATION: T ANATION: T st.id = 1;</pre>	_ERROR_ (D) 0 Since FIRST.ID ERROR_ (D) 0 OTAL is calcula	FIRST.ID (D) 1 = 1, TOTAL is s FIRST.ID (D) 1 ated.	0 et to 0. LAST.ID (D) 0	ID (К) A02 ID (К) A02	4 SCORE (D) 4	0 TOTAL (K) 4
	if fir PDV: EXPL total PDV: EXPL if las	<pre>st.id = 1 _N_(D) 4 ANATION: S + score; _N_(D) 4 ANATION: T st.id = 1; ANATION: T</pre>	_ERROR_ (D) 0 Since FIRST.ID ERROR_ (D) 0 OTAL is calcula	FIRST.ID (D) 1 = 1, TOTAL is s FIRST.ID (D) 1 ated. = statement is e	0 et to 0. LAST.ID (D) 0	ID (K) A02 ID (K) A02 FALSE beca	4 SCORE (D) 4	0 TOTAL (K) 4

Figure 4d. Fourth iteration for Program 5.

Ex3: ID A01 A01 A01 A02 A02	ERATION: SCORE 3 4 2 4 2 ex3_1;	← Reading					
PDV:	_N_(D)	ERROR_ (D) 0	FIRST.ID (D)	LAST.ID (D) 0	ID (K) A02	SCORE (D) 4	TOTAL (K)
			·····				
EXPL	ANATION:	_N_ is increment	ed to 5. The va	alues for the ren	naining vari	ables are retai	ned.
set ez	· · · · · · ·						
PDV:	_N_(D)	ERROR_ (D) 0	FIRST.ID (D) 0	LAST.ID (D)	ID (K) A02	SCORE (D) 2	TOTAL (K) 4
		<u> </u>	~	•			·
	ST.ID is set	SAS copies the find the find the find the find the find the first second			d <i>ex3</i> to the	PDV. FIRST	ID is set to
if fir		1 then total					
PDV:	_N_(D)	ERROR_ (D) 0	FIRST.ID (D)	LAST.ID (D)	<u>ID (K)</u> A02	SCORE (D) 2	TOTAL (K)
	+ score;	FIRST.ID ≠ 1, so]
PDV:	_N_ (D)	ERROR_ (D)	FIRST.ID (D)	LAST.ID (D)	ID (K) A02	SCORE (D) 2	TOTAL (K) 6
FDV.	5	0	0	I	AUZ	2	0
if las	st.id = 1	TOTAL is calcula ; The subsetting IF		valuated to be	TRUE since	LAST.ID equa	als 1.
run; Ex3_1 A01 A02	: D	TOTAL 9 6					
and T	OTAL from	SAS reaches the the PDV to the d jin the 6 th iteratio	ataset ex3 1.				
	there are no or PROC s	o more observati tep.	ons to read in t	he 6 th iteration,	the SAS sy	stem goes to t	ne next

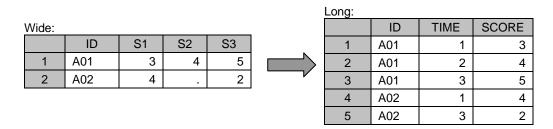
Figure 4e. Fifth iteration for Program 5.

RESTRUCTURING DATASETS

Restructuring datasets denotes transforming data from one observation per subject (the *wide* format) to multiple observations per subject (the *long* format) or transforming data from the *long* format to data in the *wide* format. The purpose of the transformation to different formats is to suit the data format requirement for different types of statistical procedures. This type of data transformation can be easily done by using more advanced programming techniques, such as using ARRAY or using PROC TRANSPOSE. However, this can also be accomplished without advanced techniques for more simple cases.

FROM WIDE FORMAT TO LONG FORMAT

Suppose that you are transforming data from the *wide* format to the *long* format such as the example below. Notice that data in the *long* format has a variable, TIME, that is used to distinguish the different measurements for each subject in the *wide* format. The original variables in the *wide* format, S1 - S3, become the variable SCORE in the *long* format.



Since only two observations need to be read from the *wide* dataset, there will be only two iterations for the DATA step processing. That means you need to generate the output three times for each iteration. Also, any missing values in variables S1 - S3 will not be outputted in the *long* dataset. Here is a solution for using multiple OUTPUT statements in one DATA step.

```
Program 6:
data long (drop=s1-s3);
   set wide;
   time = 1;
   score = s1;
   if not missing(score) then output;
   time = 2;
   score = s2;
   if not missing(score) then output;
   time = 3;
   score = s3;
   if not missing(score) then output;
run;
```

In Program 6, immediately after the SET statement, the TIME variable is set to 1. Next, the value from S1 is assigned to the SCORE variable. Now all the elements for the first observation in the *long* dataset are ready for outputting. Before outputting, check whether the SCORE value is missing or not; if it is not missing, use the explicit OUTPUT statement to create the first observation for the *long* dataset. Next, assign value 2 to the TIME variable and assign the value from S2 to SCORE and output the dataset again. Similar processes are repeated; assign 3 to TIME, assign S3 to SCORE, and output. Within the first iteration of the DATA step processing, the values for TIME and SCORE are being replaced three times. Once they are replaced, they are outputted to the final dataset. See Figure 5a for more details.

The second iteration (see Figure 5b) is similar to the first iteration. The only difference is that S2 is missing. After the value for S2 is assigned to SCORE, the contents in the PDV are not copied to the final dataset because SCORE equals missing.

IRST ITERATIO	<u>N:</u>		-					
Wide: ID	S1 S2	S3						
A01	3 4	5	•	- Reading				
A02	4	2						
data long (d	rop=s1-s3	;						
N_(ID (K)	S1 (D)	S2 (D)	S3 (D)	TIME (K)	SCORE (K)
PDV: 1	C			•	•	•	•	•
EXPLANATIO	N: _N_ is set	to 1 an	d_ERROF	R_ is set to	0. Other v	/ariables a	re set to mis	sing.
set wide;								
_ <u>N_(</u>		R_ (D)	ID (K)	S1 (D)	S2 (D)	S3 (D)	TIME (K)	SCORE (K)
PDV: 1	C		A01	3	4	5	•	•
EXPLANATIO	N: The first o	oservat	ion from th	e wide data	aset is cop	ied to the I	PDV.	
time = 1; sc	ore = s1;							
N (D) _ERRO	R (D)	ID (K)	S1 (D)	S2 (D)	S3 (D)	TIME (K)	SCORE (K)
PDV: 1	C		A01	3	4	5	1	3
EXPLANATIO	N: TIME is se	t to 1 a	nd SCORE	E is set to 3	3 (from the	value of S	 1).	
if not missi	ng(score)	Luen	output;					
Long:							d SCORE a	
ID	TIME	SC	CORE	copied to	o dataset <i>l</i> a		d SCORE a	
U	TIME 1	SC	CORE 3		o dataset <i>l</i> a			
ID	1	SC		copied to	o dataset <i>l</i> a			
ID A01 time = 2; sc N_(1 ore = s2;		3 ID (K)	copied to missing. S1 (D)	o dataset <i>la</i> S2 (D)	ong since S	SCORE is no	
ID A01 time = 2; sc N_(PDV:1	1 ore = s2; D) _ERRO	R_ (D)	3 ID (K) A01	copied to missing. S1 (D) 3	o dataset <i>lo</i> S2 (D) 4	ong since S S3 (D) 5	SCORE is no TIME (K) 2	ot
ID A01 time = 2; sc N_(1 ore = s2; D) _ERRO	R_ (D)	3 ID (K) A01	copied to missing. S1 (D) 3	o dataset <i>lo</i> S2 (D) 4	ong since S S3 (D) 5	SCORE is no TIME (K) 2	ot
ID A01 time = 2; sc N_(PDV:1	1 ore = s2; D) _ERRO 0 N: TIME is se	R_ (D) It to 2 a	3 ID (K) A01 nd SCORE	copied to missing. S1 (D) 3	o dataset <i>lo</i> S2 (D) 4	ong since S S3 (D) 5	SCORE is no TIME (K) 2	ot
ID A01 time = 2; sc N_(PDV:1 EXPLANATION if not missi	1 ore = s2; D) _ERRO 0 N: TIME is se	R_ (D) It to 2 a	3 ID (K) A01 nd SCORE	copied to missing. S1 (D) 3 is set to 4	o dataset <i>la</i> S2 (D) 4 I (from the	S3 (D) 5 value of S	SCORE is no TIME (K) 2 2).	SCORE (K)
ID A01 time = 2; sc N_(PDV:1 EXPLANATIO	1 ore = s2; D) _ERRO 0 N: TIME is se	R_ (D) tt to 2 a then	3 ID (K) A01 nd SCORE	S1 (D) 3 is set to 4	S2 (D) 4 I (from the	S3 (D) 5 value of Si 0, TIME, an	SCORE is no TIME (K) 2 2). Ind SCORE a	SCORE (K)
ID A01 time = 2; sc PDV: 1 EXPLANATION if not missi Long:	1 ore = s2; D) _ERRO 	R_ (D) tt to 2 a then	3 ID (K) A01 nd SCORE output;	S1 (D) 3 is set to 4	S2 (D) 4 I (from the	S3 (D) 5 value of Si 0, TIME, an	SCORE is no TIME (K) 2 2).	SCORE (K)
ID A01 time = 2; sc N_(PDV:1 EXPLANATION if not missi Long: ID	1 ore = s2; D) _ERRO 0 N: TIME is se ng(score) TIME	R_ (D) tt to 2 a then	3 ID (K) A01 nd SCORE output;	S1 (D) 3 is set to 4	S2 (D) 4 I (from the	S3 (D) 5 value of Si 0, TIME, an	SCORE is no TIME (K) 2 2). Ind SCORE a	SCORE (K)
ID A01 time = 2; sc PDV: 1 EXPLANATION if not missi Long: ID A01 A01	1 ore = s2; D) _ERRO (C) N: TIME is se ng(score) TIME 1 2	R_ (D) tt to 2 a then	3 ID (K) A01 nd SCORE output; CORE 3	S1 (D) 3 is set to 4	S2 (D) 4 I (from the	S3 (D) 5 value of Si 0, TIME, an	SCORE is no TIME (K) 2 2). Ind SCORE a	SCORE (K)
ID A01 time = 2; sc N_(PDV:1 EXPLANATION if not missi Long: ID A01	1 ore = s2; D) _ERRO (C) N: TIME is set ng(score) TIME 1 2 ore = s3;	R_(D) t to 2 a then	3 ID (K) A01 nd SCORE output; CORE 3	S1 (D) 3 is set to 4	S2 (D) 4 I (from the	S3 (D) 5 value of Si 0, TIME, an	SCORE is no TIME (K) 2 2). Ind SCORE a	SCORE (K)
ID A01 time = 2; sc PDV: 1 EXPLANATION if not missi Long: ID A01 time = 3; sc	1 ore = s2; D) _ERRO (C) N: TIME is second ng(score) TIME 1 2 ore = s3;	R_ (D) t to 2 a then SC R_ (D)	3 ID (K) A01 nd SCORE 0 CORE 3 4	S1 (D) 3 is set to 4 EXPLAN to datase	S2 (D) 4 (from the IATION: ID et <i>long</i> sinc	since S S3 (D) 5 value of S value of S value of S value of S	SCORE is no TIME (K) 2 2). Id SCORE a is not missir	score (K) 4 re copied
ID A01 time = 2; sc N_(PDV: 1 EXPLANATIO if not missi Long: ID A01 time = 3; sc PDV: PDV:	1 pre = s2; D) _ERRO (N: TIME is se ng(score) TIME 1 2 pre = s3; D) _ERRO (R_ (D) t to 2 a then S(R_ (D)	3 ID (K) A01 nd SCORE 0 0 0 0 0 0 0 0 1 0 (K) A01 1 0 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1	S1 (D) 3 is set to 4 EXPLAN to datase S1 (D) 3	S2 (D) 4 (from the IATION: ID et <i>long</i> sinc S2 (D) 4	S3 (D) 5 value of S value of S value of S S3 (D) 5	SCORE is no TIME (K) 2 2). d SCORE a is not missir TIME (K) 3	SCORE (K) 4 re copied ng. SCORE (K)
ID A01 time = 2; sc PDV: 1 EXPLANATION if not missi Long: ID A01 time = 3; sc PDV: PDV: 1 EXPLANATION	1 ore = s2; D) _ERRO O N: TIME is se ng(score) TIME 1 2 ore = s3; D) _ERRO O N: TIME is se	R_ (D) t to 2 a then R_ (D) t to 3 a	3 ID (K) A01 nd SCORE 0 0 0 0 0 0 0 1 0 (K) A01 nd SCORE 0 1 0 (K) 0 1 1 1 1 1 1 1 1 1 1 1 1 1	S1 (D) 3 is set to 4 EXPLAN to datase S1 (D) 3	S2 (D) 4 (from the IATION: ID et <i>long</i> sinc S2 (D) 4	S3 (D) 5 value of S value of S value of S S3 (D) 5	SCORE is no TIME (K) 2 2). d SCORE a is not missir TIME (K) 3	SCORE (K) 4 re copied ng. SCORE (K)
ID A01 time = 2; sc N_(PDV: 1 EXPLANATIO if not missi Long: ID A01 A01 ID A01 ID A01 ID A01 ID ID ID A01 IT EXPLANATIO If not missi if not missi	1 ore = s2; D) _ERRO O N: TIME is se ng(score) TIME 1 2 ore = s3; D) _ERRO O N: TIME is se	R_ (D) t to 2 a then R_ (D) t to 3 a	3 ID (K) A01 nd SCORE 0 0 0 0 0 0 0 1 0 (K) A01 nd SCORE 0 1 0 (K) 0 1 1 1 1 1 1 1 1 1 1 1 1 1	S1 (D) 3 is set to 4 EXPLAN to datase S1 (D) 3	S2 (D) 4 (from the IATION: ID et <i>long</i> sinc S2 (D) 4	S3 (D) 5 value of S value of S value of S S3 (D) 5	SCORE is no TIME (K) 2 2). d SCORE a is not missir TIME (K) 3	SCORE (K) 4 re copied ng. SCORE (K)
ID A01 time = 2; sc N_(PDV: 1 if not missi Long: ID A01 A01 if not missi Long: ID A01 A01 ID A01 ID ID	1 ore = s2; D) _ERRO O N: TIME is se ng(score) TIME 1 2 ore = s3; D) _ERRO O N: TIME is se ng(score)	R_(D) t to 2 a then R_(D) t to 3 a then	3 ID (K) A01 nd SCORE 3 4 ID (K) A01 nd SCORE output;	S1 (D) 3 is set to 4 EXPLAN to datase S1 (D) 3 is set to 5	S2 (D) 4 4 (from the IATION: ID et <i>long</i> sinc S2 (D) 4 5 (from the	S3 (D) 5 value of S value of S c, TIME, ar ce SCORE S3 (D) 5 value of S	SCORE is no TIME (K) 2 2). d SCORE a is not missir TIME (K) 3 3).	score (K) 4 re copied ng. SCORE (K) 5
ID A01 time = 2; sc N_(PDV: 1 if not missi Long: ID A01 A01 if not missi Long: ID A01 A01 ID ID ID ID ID ID ID IT EXPLANATION if not missi Long: ID ID	1 ore = s2; D) _ERRO O N: TIME is se ng(score) TIME 1 2 ore = s3; D) _ERRO O N: TIME is se ng(score) TIME	R_(D) t to 2 a then R_(D) t to 3 a then	3 ID (K) A01 nd SCORE 3 4 ID (K) A01 output; CORE output;	S1 (D) 3 is set to 4 EXPLAN to datase S1 (D) 3 is set to 5 EXPLAN	S2 (D) 4 4 4 (from the IATION: ID 5 (from the IATION: ID	5 S3 (D) 5 value of S value of S 0, TIME, ar 5 value of S value of S value of S	SCORE is no TIME (K) 2 2). IN SCORE a is not missin TIME (K) 3 3). IN SCORE a	score (K) 4 re copied ng. Score (K) 5 re copied
ID A01 time = 2; sc PDV: 1 if not missi Long: ID A01 A01 if not missi Long: ID A01 ID A01 ID A01 if not missi PDV: 1 EXPLANATIOI if not missi Long: ID A01	1 ore = s2; D) _ERRO O N: TIME is se ng(score) TIME 1 2 ore = s3; D) _ERRO O N: TIME is se ng(score) TIME 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	R_(D) t to 2 a then R_(D) t to 3 a then	3 ID (K) A01 nd SCORE 3 4 ID (K) A01 nd SCORE output;	S1 (D) 3 is set to 4 EXPLAN to datase S1 (D) 3 is set to 5 EXPLAN	S2 (D) 4 4 4 (from the IATION: ID 5 (from the IATION: ID	5 S3 (D) 5 value of S value of S 0, TIME, ar 5 value of S value of S value of S	SCORE is no TIME (K) 2 2). d SCORE a is not missir TIME (K) 3 3).	score (K) 4 re copied ng. Score (K) 5 re copied
ID A01 time = 2; sc N_(PDV: 1 if not missi Long: ID A01 A01 if not missi Long: ID A01 A01 ID ID ID ID ID ID ID IT EXPLANATION if not missi Long: ID ID	1 ore = s2; D) _ERRO OR N: TIME is se ng(score) TIME 1 2 ore = s3; D) _ERRO OR N: TIME is se ng(score) TIME	R_(D) t to 2 a then R_(D) t to 3 a then	3 ID (K) A01 nd SCORE 3 4 ID (K) A01 output; CORE 0 3 4 CORE 3 5 CORE 3 5 CORE CORE 5 CORE CORE CORE CORE CORE CORE CORE CORE CORE CORE CORE CORE CORE COR	S1 (D) 3 is set to 4 EXPLAN to datase S1 (D) 3 is set to 5 EXPLAN	S2 (D) 4 4 4 (from the IATION: ID 5 (from the IATION: ID	5 S3 (D) 5 value of S value of S 0, TIME, ar 5 value of S value of S value of S	SCORE is no TIME (K) 2 2). IN SCORE a is not missin TIME (K) 3 3). IN SCORE a	score (K) 4 re copied ng. Score (K) 5 re copied

Figure 5a. First iteration for Program 6.

	TERAT	ION:								
Wide:	ID	S1	S2	S3						
	A01	3	4	5						
	A02	4	•	2		← Reading				
lata lo	ng (dı	rop=s	s1-s3);	;						
	N (I	D) _	ERROR	(D)	ID (K)	S1 (D)	S2 (D)	S3 (D)	TIME (K)	SCORE (K)
PDV:	2		0		A01	3	4	5	•	•
EXPLA	NATION	N: _N_	is incre	mente	d to 2. ID	and S1-S3	are retain	ed from the	e previous it	eration. The
newly-ci	reated v	variab	les, TIM	E and	SCORE,	are set to n	nissing.			
et wid	e;									
	N_ (D) _	ERROR	L_ (D)	ID (K)	S1 (D)	S2 (D)	S3 (D)	TIME (K)	SCORE (K)
PDV:	2		0		A02	4	•	2	•	•
EXPLA	NATION	N: The	second	obser	vation fro	m the wide	dataset is o	copied to t	he PDV.	
ime =	1; sco	ore =	= s1;							
	N (D)	ERROR	(D)	ID (K)	S1 (D)	S2 (D)	S3 (D)	TIME (K)	SCORE (K)
PDV:	2		0	_ /	A02	4	•	2	1	4
EXPLA	NATION	I: TIM	IE is set	to 1 ar	nd SCOR	E is set to 4	(from the	value of S ^r	1).	
					output;					
	missii	.19 (SC	JOIE) (oucpuc,					
<u>ong:</u> ID		TIN	ME	20	ORE					
10 \01		111	1	30	3				d SCORE a	
401			2		4		o dataset lo	ng since S	SCORE is no	ot
401			3		_	i missing.				
101			3		5	·				
			3 1		5 4	<u> </u>				
402	2; sco	ore =	1			·				
\02 ime =	2; sco _N_(1	2_ (D)	4 ID (K)	S1 (D)	S2 (D)	S3 (D)	TIME (K)	SCORE (K)
102 ime = PDV: [_ N_ (2	D) _	1 = s2; ERROR 0		4 ID (K) A02	S1 (D)	•	2	2	SCORE (K)
A02 ime = PDV: [_ N_ (2	D) _	1 = s2; ERROR 0		4 ID (K) A02		•	2	2	SCORE (K)
A02 ime = PDV: [EXPLAN	_ N_ (I 2 NATION	D) N: TIM	1 = s2; ERROR 0 IE is set	to 2 a	4 ID (K) A02	S1 (D)	•	2	2	SCORE (K) •
A02 ime = PDV: [EXPLAN f not 1	_N_(2 NATION missir	D)	1 ERROR 0 IE is set core) t	to 2 ar	4 ID (K) A02 nd SCOR	S1 (D) 4 E is set to n	• nissing (fro	2 m the valu	2	SCORE (K)
A02 ime = PDV: [EXPLAN f not 1 EXPLAN	_N_(2 NATION missin	D) N: TIM ng (sc N: No	1 ERROR 0 IE is set core) t	to 2 ar	4 ID (K) A02 nd SCOR	S1 (D)	• nissing (fro	2 m the valu	2	SCORE (K) •
A02 ime = PDV: [EXPLAN f not 1 EXPLAN	_N_(2 NATION missin NATION 3; sco	D) N: TIM ng (sc N: No pre =	1 ERROR 0 IE is set core) t output is = \$3;	to 2 ar	4 ID (K) A02 nd SCOR putput; rated sinc	S1 (D) 4 E is set to n e SCORE 6	• nissing (fro equals miss	2 m the valu	2 e of S2).	•
A02 ime = PDV: [EXPLAN f not 1 EXPLAN ime =	_N_(2 NATION missin NATION 3; sco _N_(D) N: TIM ng (sc N: No pre =	1 ERROR 0 IE is set core) t output is = s3; ERROR	to 2 ar	4 ID (K) A02 nd SCOR putput; rated sinc ID (K)	S1 (D) 4 E is set to n e SCORE e S1 (D)	• nissing (fro	2 m the valu ing. S3 (D)	2 e of S2). TIME (K)	• SCORE (K)
A02 ime = PDV: [EXPLAN f not p EXPLAN ime = PDV: [_N_(2 NATION missin NATION 3; scc _N_(2	D) N: TIM ng (sc N: No bre = D)	1 ERROR 0 IE is set core) t output is = \$3; ERROR 0	to 2 ar	4 ID (K) A02 and SCOR output; rated sinc ID (K) A02	S1 (D) 4 E is set to n e SCORE e S1 (D) 4	• nissing (fro equals miss S2 (D) •	2 m the valu ing. S3 (D) 2	2 e of S2). TIME (K) 3	•
A02 ime = PDV: [EXPLAN f not 1 EXPLAN ime = PDV: [EXPLAN	_N_(2 NATION missin NATION 3; scc _N_(2 NATION	D)	1 ERROR 0 IE is set core) t output is s3; ERROR 0 IE is set	to 2 ar	4 ID (K) A02 nd SCOR output; rated sinc ID (K) A02 nd SCOR	S1 (D) 4 E is set to n e SCORE e S1 (D)	• nissing (fro equals miss S2 (D) •	2 m the valu ing. S3 (D) 2	2 e of S2). TIME (K) 3	• SCORE (K)
A02 ime = PDV: [EXPLAN f not 1 EXPLAN ime = PDV: [EXPLAN	_N_(2 NATION missin NATION 3; scc _N_(2 NATION	D)	1 ERROR 0 IE is set core) t output is s3; ERROR 0 IE is set	to 2 ar	4 ID (K) A02 and SCOR output; rated sinc ID (K) A02	S1 (D) 4 E is set to n e SCORE e S1 (D) 4	• nissing (fro equals miss S2 (D) •	2 m the valu ing. S3 (D) 2	2 e of S2). TIME (K) 3	• SCORE (K)
A02 ime = PDV: [EXPLAN ime = PDV: [EXPLAN f not n f not n cong:	_N_(2 NATION missin NATION 3; scc _N_(2 NATION	D)	1 ERROR 0 IE is set core) t output is ERROR 0 IE is set core) t	to 2 ar to 2 ar gener gener to 3 ar	4 ID (K) A02 nd SCOR output; rated sinc ID (K) A02 nd SCOR output;	S1 (D) 4 E is set to n e SCORE e S1 (D) 4	• nissing (fro equals miss S2 (D) •	2 m the valu ing. S3 (D) 2	2 e of S2). TIME (K) 3	• SCORE (K)
A02 ime = PDV: [EXPLAN ime = PDV: [EXPLAN f not : _ong: ID	_N_(2 NATION missin NATION 3; scc _N_(2 NATION	D)	1 ERROR 0 IE is set core) t output is s3; ERROR 0 IE is set core) t	to 2 ar to 2 ar gener gener to 3 ar	4 ID (K) A02 output; rated sinc ID (K) A02 nd SCOR output; CORE	S1 (D) 4 E is set to n e SCORE e S1 (D) 4 E is set to 2	• nissing (fro equals miss S2 (D) •	2 m the valu ing. S3 (D) 2 value of S3	2 e of S2). TIME (K) 3 3).	• SCORE (K) 2
A02 ime = PDV: [EXPLAN ime = PDV: [EXPLAN f not p Long: ID A01	_N_(2 NATION missin NATION 3; scc _N_(2 NATION	D)	1 = s2; ERROR 0 IE is set core) t output is = s3; ERROR 0 IE is set core) t ME 1	to 2 ar to 2 ar gener gener to 3 ar	4 ID (K) A02 nd SCOR output; rated sinc ID (K) A02 nd SCOR output; CORE 3	S1 (D) 4 E is set to n e SCORE e S1 (D) 4 E is set to 2 E is set to 2	• nissing (fro equals miss S2 (D) • (from the ATION: ID	2 m the valu ing. S3 (D) 2 value of S3	2 e of S2). TIME (K) 3 3). d SCORE a	• SCORE (K) 2 re copied
A02 ime = PDV: [EXPLAN f not 1 EXPLAN ime = PDV: [EXPLAN f not 1 Long: ID A01 A01	_N_(2 NATION missin NATION 3; scc _N_(2 NATION	D)	1 ERROR 0 IE is set core) t output is s3; ERROR 0 IE is set core) t	to 2 ar to 2 ar gener gener to 3 ar	4 ID (K) A02 output; rated sinc ID (K) A02 nd SCOR output; CORE	S1 (D) 4 E is set to n e SCORE e S1 (D) 4 E is set to 2 E is set to 2	• nissing (fro equals miss S2 (D) • (from the ATION: ID	2 m the valu ing. S3 (D) 2 value of S3	2 e of S2). TIME (K) 3 3).	• SCORE (K) 2 re copied
A02 ime = PDV: [EXPLAN f not	_N_(2 NATION missin NATION 3; scc _N_(2 NATION	D)	1 = s2; ERROR 0 IE is set core) t output is = s3; ERROR 0 IE is set core) t ME 1 2	to 2 ar to 2 ar gener gener to 3 ar	4 ID (K) A02 nd SCOR output; rated sinc ID (K) A02 nd SCOR output; CORE 3 4	S1 (D) 4 E is set to n e SCORE e S1 (D) 4 E is set to 2 E is set to 2	• nissing (fro equals miss S2 (D) • (from the ATION: ID	2 m the valu ing. S3 (D) 2 value of S3	2 e of S2). TIME (K) 3 3). d SCORE a	• SCORE (K) 2 re copied
A02 ime = PDV: [EXPLAN f not 1 EXPLAN ime = PDV: [EXPLAN f not 1 Long: ID A01 A01 A01 A02	_N_(2 NATION missin NATION 3; scc _N_(2 NATION	D)	1 = s2; ERROR 0 IE is set core) t output is = s3; ERROR 0 IE is set core) t ME 1 2 3	to 2 ar to 2 ar gener gener to 3 ar	4 ID (K) A02 nd SCOR putput; rated sinc ID (K) A02 nd SCOR output; CORE 3 4 5	S1 (D) 4 E is set to n e SCORE e S1 (D) 4 E is set to 2 E is set to 2	• nissing (fro equals miss S2 (D) • (from the ATION: ID	2 m the valu ing. S3 (D) 2 value of S3	2 e of S2). TIME (K) 3 3). d SCORE a	• SCORE (K) 2 re copied
A02 ime = PDV: [EXPLAN ime = PDV: [EXPLAN f not n Long:	_N_(2 NATION missin NATION 3; scc _N_(2 NATION	D)	1 = s2; ERROR 0 IE is set core) t output is = s3; ERROR 0 IE is set core) t ME 1 2 3 1	to 2 ar to 2 ar gener gener to 3 ar	4 ID (K) A02 nd SCOR putput; rated sinc ID (K) A02 nd SCOR a02 nd SCOR core 3 4 5 4	S1 (D) 4 E is set to n e SCORE e S1 (D) 4 E is set to 2 E is set to 2	• nissing (fro equals miss S2 (D) • (from the ATION: ID	2 m the valu ing. S3 (D) 2 value of S3	2 e of S2). TIME (K) 3 3). d SCORE a	• SCORE (K) 2 re copied
A02 ime = EXPLAN f not p EXPLAN ime = PDV: [EXPLAN f not p EXPLAN f not p Long: ID A01 A01 A01 A01 A02 A02 un;	_N_((2 NATION missin NATION 3; scc _N_((2 NATION missin	D)	1 = \$2; ERROR 0 IE is set core) t core) t ERROR 0 IE is set core) t ME 1 2 3 1 3	to 2 ar chen of gener to 3 ar chen of SC	4 ID (K) A02 nd SCOR output; rated sinc ID (K) A02 nd SCOR add SCOR	S1 (D) 4 E is set to n e SCORE e S1 (D) 4 E is set to 2 EXPLAN to datase	• nissing (fro equals miss S2 (D) • (from the ATION: ID et <i>long</i> sinc	2 m the valu ing. S3 (D) 2 value of S: value of S:	2 e of S2). TIME (K) 3 3).	• SCORE (K) 2 re copied
A02 ime = EXPLAN f not p EXPLAN ime = PDV: EXPLAN f not p EXPLAN f not p Long: ID A01 A01 A01 A01 A02 A02 un; EXPLAN	_N_((2 NATION MISSIN 3; scc _N_((2 NATION MISSIN	D) N: TIM ng (sc N: No Dre = D) N: TIM ng (sc TIM ng (sc TIM	1 = \$2; ERROR 0 IE is set core) t core) t ERROR 0 IE is set core) t ME 1 2 3 1 3 SAS sy	to 2 ar chen of gener gener to 3 ar chen of SC	4 ID (K) A02 nd SCOR output; rated sinc ID (K) A02 nd SCOR add SCOR	S1 (D) 4 E is set to n e SCORE e S1 (D) 4 E is set to 2 EXPLAN to datase	• nissing (fro equals miss S2 (D) • (from the ATION: ID et <i>long</i> sinc	2 m the valu ing. S3 (D) 2 value of S: value of S: value of S:	2 e of S2). TIME (K) 3 3). d SCORE a	• SCORE (K) 2 re copied ng. 3 rd iteration.

Figure 5b. Second iteration for Program 6.

FROM LONG FORMAT TO WIDE FORMAT

Transforming data from the *long* format to the *wide* format is a little more complicated. Details for each single step will not be covered. However, if you attempt to draw the PDV yourself, you will be able to figure-out the contents of the PDV within each iteration of the DATA step processing. The SAS code is listed below:

Long:			
	ID	TIME	SCORE
1	A01	1	3
2	A01	2	4
3	A01	3	5
4	A02	1	4
5	A02	3	2

	Wide:				
\		ID	S1	S2	S3
	1	A01	3	4	5
	2	A02	4		2
		-			

Program 7:

```
proc sort data=long;
    by id time;
run;
data wide (drop=time score);
    set long;
    by id;
    retain s1-s3;
    if first.id then do;
        s1 = .; s2 = .; s3 = .;
    end;
    if time = 1 then s1 = score;
    else if time = 2 then s2 = score;
    else s3 = score;
    if last.id = 1;
run;
```

Program 7 begins by sorting the *long* dataset by ID and TIME. Sorting the variable TIME within each ID is important because it ensures the horizontal order of S1 – S3 in the *wide* dataset for each subject can be matched correctly with the vertical order of SCORE in the *long* dataset.

Since you are reading five observations from the *long* dataset but only creating two observations, it means that you are *not* copying data from the PDV to the final dataset at each iteration. As a matter of fact, you only need to generate one observation once all the observations for each subject have been processed. That means that the newly- created variables S1 - S3 in the final dataset need to retain their values; otherwise S1 - S3 will be initialized to missing at the beginning of each iteration of the DATA step processing.

Notice that subject A02 is missing one observation for TIME equaling 2. The value of S2 from the previous subject (A01) will be copied to the dataset *wide* for the subject A02 instead of a missing value because S2 is being retained. To avoid this problem, initialize S1 – S3 to missing when processing the first observation for each subject.

USING THE PUT STATEMENT TO OBSERVE THE CONTENTS OF THE PDV DURING EACH STEP

If you are not sure what the contents of the PDV are during each step of the DATA step processing, use the PUT statement inside the DATA step, which will generate the contents of each variable in the PDV on the SAS log. For example,

```
Program 8:
data ex2_2;
    put "1ST PUT" _all_;
    set ex2;
    put "2ND PUT" _all_;
    retain total 0;
    put "3RD PUT" _all_;
    total = sum(total, score);
    put "4TH PUT" _all_;
run;
```

The PUT statement can combine text strings in quotations with the contents of the variable on the SAS log. The keyword _ALL_ means that all the variables, including the automatic variables, will be output to the SAS log. Here is the corresponding log from Program 8.

SAS log:

```
289
    data ex2 2;
        put "1ST PUT" _all_;
290
291
         set ex2;
292
         put "2ND PUT" _all_;
293
         retain total 0;
294
         put "3RD PUT" _all_;
295
         total = sum(total, score);
296
         put "4TH PUT" all ;
297 run;
1ST PUTid= score=. total=0 _ERROR_=0 _N_=1
2ND PUTid=A01 score=3 total=0 _ERROR_=0 _N =1
3RD PUTid=A01 score=3 total=0 _ERROR_=0 _N_=1
4TH PUTid=A01 score=3 total=3 _ERROR_=0 _N_=1
1ST PUTid=A01 score=3 total=3 _ERROR_=0 _N_=2
2ND PUTid=A02 score=. total=3 _ERROR_=0 _N_=2
3RD PUTid=A02 score=. total=3 ERROR =0 N =2
4TH PUTid=A02 score=. total=3 ERROR =0 N =2
1ST PUTid=A02 score=. total=3 _ERROR_=0 _N_=3
2ND PUTid=A03 score=4 total=3 _ERROR_=0 _N_=3
3RD PUTid=A03 score=4 total=3 _ERROR_=0 _N_=3
4TH PUTid=A03 score=4 total=7 _ERROR_=0 _N_=3
1ST PUTid=A03 score=4 total=7 ERROR =0 N =4
```

CONCLUSION

To be a successful SAS programmer, you must be able to thoroughly comprehend how DATA steps are processed. The most important part of DATA step processing is to understand how data is transformed to the PDV and how data is copied from the PDV to a new dataset.

REFERENCES

Li, Arthur. 2013. Handbook of SAS[®] DATA Step Programming. Chapman and Hall/CRC.

CONTACT INFORMATION

Arthur Li City of Hope National Medical Center Department of Information Science 1500 East Duarte Road Duarte, CA 91010 - 3000 Work Phone: (626) 256-4673 ext. 65121 Fax: (626) 471-7106 E-mail: arthurli@coh.org

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.