

Check your code! Control your data!

October 2017

Kristina Zweier

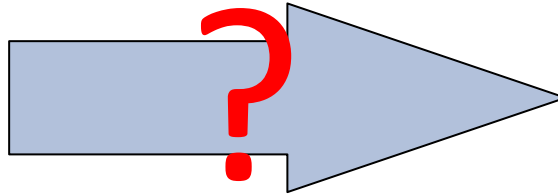


CLINIPACE
WORLDWIDE

A Beginner's Dilemma

Start:

- New to SAS
- New to content



Goal:

Correct Outputs

A Beginner's Dilemma

Start:

- New to SAS
- New to content



Experience



Goal:

Correct Outputs

Know your data attributes

- Content: What is your data about?
- Quantity: How many observations/variables?
- Quality: Is the data cleaned or uncleaned?
- Status: Is the data complete or incomplete?
- Structure: How is the data set constructed?

Visualize progress of data

- Get idea about data distribution and progression
- Spot entry errors

Macro %lplot

- Produces a line plot of the development of selected parameters over defined time points
- Define parameters for execution and the macro does the rest
- Premise: data set should be arranged in vertical order and contain a numeric time point variable

Macro %lplot

SUBJID	AVISITN	PARAMCD	AVAL
0004	1	HEIGHT	169,5
0004	1	WEIGHT	69,2
0004	1	PULSE	73
0004	2	HEIGHT	169,5
0004	2	WEIGHT	68,9
0004	2	PULSE	75

Macro %lplot

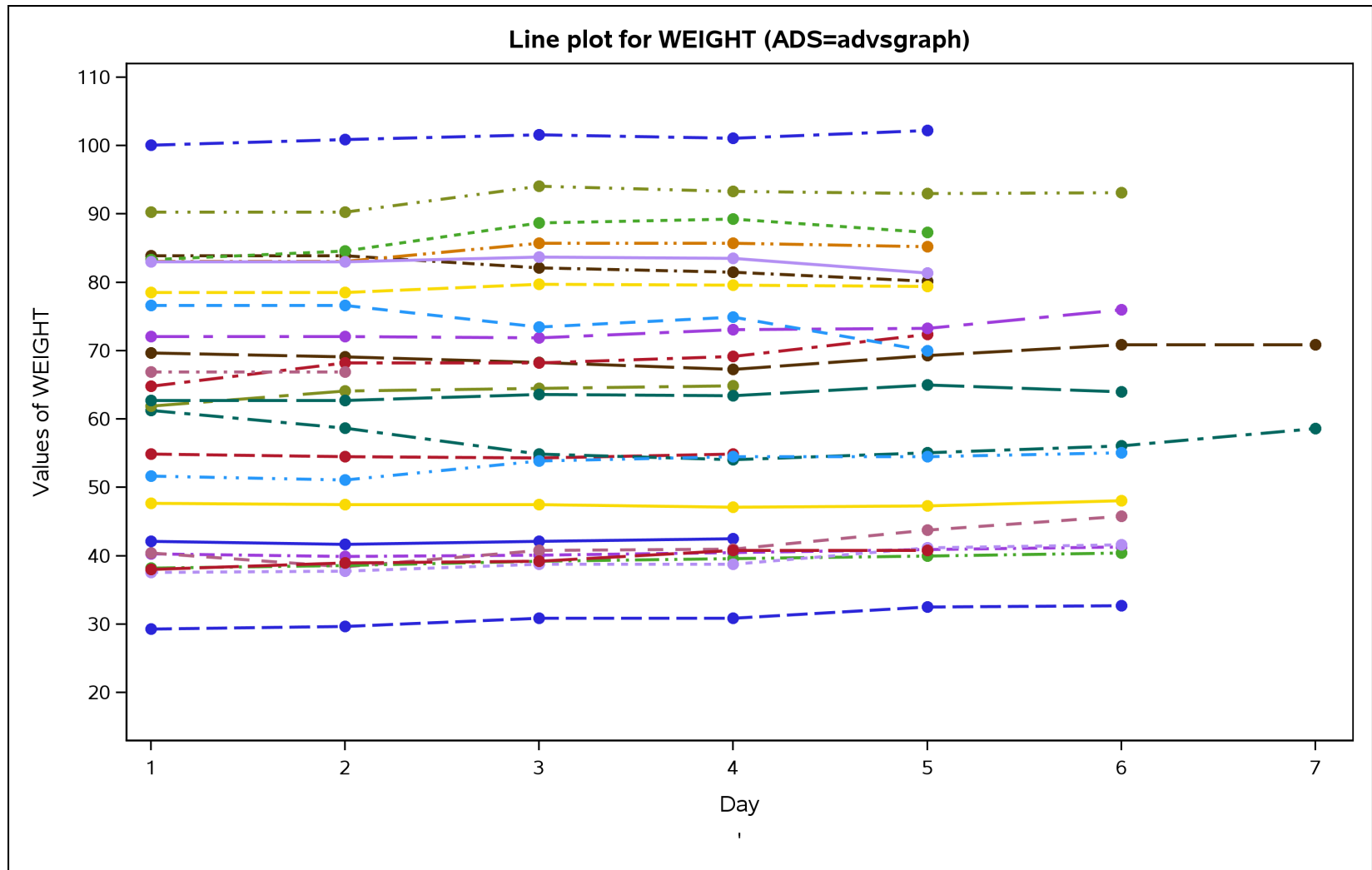
```
%MACRO lplot      ( /* MP = Mandatory parameter          */
    in             = /* MP Name of input data set          */
    ,_testvar     = /* MP Name of parameter name variable */
    ,aval         = /* MP Name of numeric parameter value variable */
    ,visit        = /* MP Name of chosen numeric time point variable */
    ,groupvar    = /* MP Name of grouping variable        */
    ,outpath     = /* MP Name of output file pathname     */
                    );
```

Macro %lplot

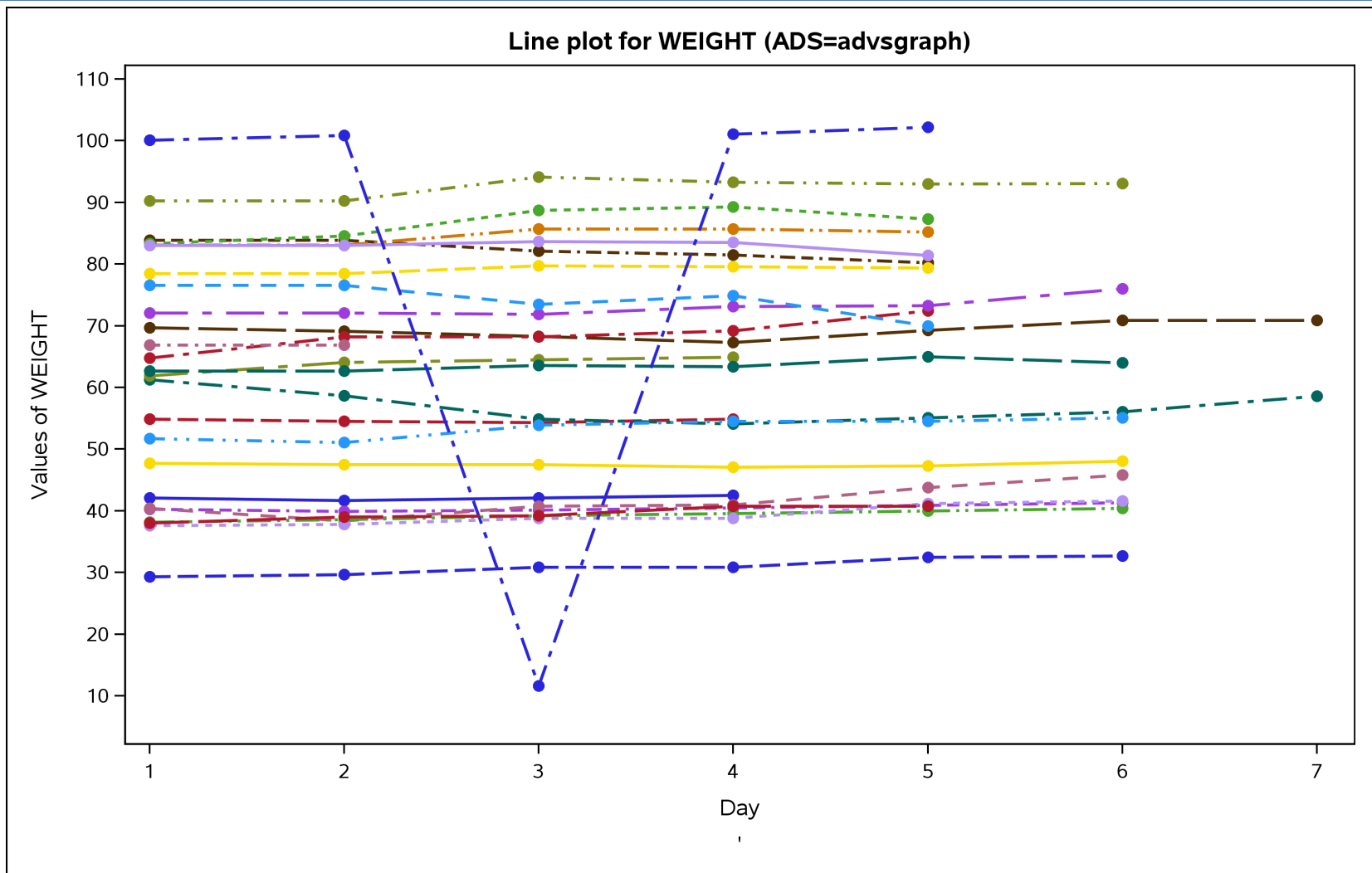
%lplot (in = advsgraph^❶ , _testvar = paramcd^❷ , aval = aval^❸ ,
visit = avisitn^❹ , groupvar = subjid^❺ ,outpath =
C:/SAS/mywork/out^❻);

SUBJID ^❺	AVISITN ^❹	PARAMCD ^❷	AVAL ^❸
0004	1	HEIGHT	169,5
0004	1	WEIGHT	69,2
0004	1	PULSE	73
0004	2	HEIGHT	169,5
0004	2	WEIGHT	68,9
0004	2	PULSE	75

Output



Output with data entry error



The Source of all that is good - GPP

Good programming practice: Apply it!

Custom Warnings & Errors

- SAS provides a possibility to write custom warning and error messages into the log file
 - ➔ use it to make it a habit to check your data regularly
- Good way to:
 - familiarize yourself with your data
 - ensure their quality
 - gain a better understanding of the mode of operation of SAS

Custom Warnings & Errors

- Include custom warnings and/or errors when using IF/THEN/ELSE statements
- Get used to creating custom warnings and/or errors by checking for missing or unexpected data

Example

Data set SASHELP.CLASS

OBS	NAME	SEX	AGE	HEIGHT	WEIGHT
1	Alfred	M	14	69	112.5
2	Alice	F	13	56.5	84
3	Barbara	F	13	65.3	98
4	Carol	F	14	62.8	102.5
5	Henry	M	14	63.5	102.5
6	James	M	12	57.3	83
7	Jane	F	12	59.8	84.5
8	Janet	F	15	62.5	112.5
9	Jeffrey	M	13	62.5	84
10	John	M	12	59	99.5
11	Joyce		11	51.3	50.5
12	Judy	F	14	64.3	90
13	Louise	F	12	56.3	77
14	Mary	F	15	66.5	112
15	Philip	M	16	72	150
16	Robert	M	12	64.8	128
17	Ronald	M	15	67	133
18	Thomas		11	57.5	85
19	William	M	15	66.5	112

*/*Split data set based on sex*/*

DATA girls boys;

SET class;

IF sex eq "F" **THEN OUTPUT** girls;

ELSE IF sex eq "M" **THEN OUTPUT** boys;

RUN;

NOTE: There were 19 observations read from the data set WORK.CLASS.

NOTE: The data set WORK.GIRLS has 8 observations and 5 variables.

+

NOTE: The data set WORK.BOYS has 9 observations and 5 variables.

= 17

Example

```
/*Split data set based on sex*/
```

```
DATA girls boys;
```

```
SET class;
```

```
IF sex eq "F" THEN OUTPUT girls;
```

```
ELSE IF sex eq "M" THEN OUTPUT boys;
```

```
ELSE PUT "WAR" "NING: missing or unexpected value for SEX. " /
```

```
name= sex=;
```

```
RUN;
```

WARNING: missing or unexpected value for SEX.

NAME=Joyce SEX=

WARNING: missing or unexpected value for SEX.

NAME=Thomas SEX=

Do what I want, not what I told you to do

- It is possible to create incorrect output without any warnings or errors in the log file
 - ➔ The program code was correct – just not the correct one for the desired output

Be(a)ware of the Program Data Vector

- Combining data sets by the SET and MERGE statements will be one of the first things beginners learn to achieve with SAS
- It is crucial to understand how SAS operates and the role of the Program Data Vector (PDV) to generate correct assembled data sets

Example

Data set TRT1



Obs	Subject Identifier	Planned Treatment	Actual Treatment	Time point (N)	ITT Population Flag	Safety Population Flag
1	1	1	1	201	Y	Y
2	1	1	1	202	Y	Y
3	1	1	1	203	Y	Y
4	2	2	.	201	Y	N
5	2	2	.	202	Y	N
6	2	2	.	203	Y	N

Data set TRT2

Obs	Subject Identifier	Planned Treatment	Time point (N)	ITT Population Flag	Safety Population Flag
1	101	1	201	Y	Y
2	101	1	202	Y	Y
3	101	1	203	Y	Y
4	102	2	201	Y	N
5	102	2	202	Y	N
6	102	2	203	Y	N



Desired Output

Data set Treatment

Obs	Subject Identifier	Planned Treatment	Actual Treatment	Time point (N)	ITT Population Flag	Safety Population Flag
1	1	1	1	201	Y	Y
2	1	1	1	202	Y	Y
3	1	1	1	203	Y	Y
4	2	2	.	201	Y	N
5	2	2	.	202	Y	N
6	2	2	.	203	Y	N
7	101	1	1	201	Y	Y
8	101	1	1	202	Y	Y
9	101	1	1	203	Y	Y
10	102	2	.	201	Y	N
11	102	2	.	202	Y	N
12	102	2	.	203	Y	N

Seemingly logical, albeit wrong approach

DATA treatment;

SET trt1

trt2 (IN=intrt2);

*) Actual treatment will be set to planned treatment for the safety patients;

IF intrt2 and saffl eq 'Y' THEN trta = trtp;

RUN;

Log file

NOTE: There were 6 observations read from the data set WORK.TRT1.

NOTE: There were 6 observations read from the data set WORK.TRT2.

NOTE: The data set WORK.TREATMENT has 12 observations and 6 variables.

Output

Obs	Subject Identifier	Planned Treatment	Actual Treatment	Time point (N)	ITT Population Flag	Safety Population Flag
1	1	1	1	201	Y	Y
2	1	1	1	202	Y	Y
3	1	1	1	203	Y	Y
4	2	2	.	201	Y	N
5	2	2	.	202	Y	N
6	2	2	.	203	Y	N
7	101	1	1	201	Y	Y
8	101	1	1	202	Y	Y
9	101	1	1	203	Y	Y
10	102	?	← 1	201	Y	N
11	102		← 1	202	Y	N
12	102		← 1	203	Y	N

What happened?

- The creation of a new data set through a DATA STEP takes place in two phases: compilation and execution
- In the compilation phase, the PDV is created
- PDV: storage area in memory, which contains the data that are read in with DATA STEP statements such as SET, MERGE or INPUT
- At the end of a DATA STEP, the content of the PDV is written into the output data set

Creation of a data set

- SAS creates one observation at a time; the DATA STEP with all its statements is executed for each observation
- At first, the variables in the PDV are set to missing; then SAS reads the input data into the PDV and executes DATA STEP statements
- After the DATA STEP has reached its end, SAS returns to the top of the DATA STEP and executes it for the next observation

SET & MERGE

Variables are NOT set to MISSING
Values are automatically retained

Output

Obs	Subject Identifier		Planned Treatment		Actual Treatment		Time point (N)		ITT Population Flag	Safety Population Flag	
1		1		1		1		201	Y		Y
2		1		1		1		202	Y		Y
3		1		1		1		203	Y		Y
4		2		2		.		201	Y		N
5		2		2		.		202	Y		N
6		2		2		.		203	Y		N
7		101		1	→	1		201	Y		Y
8		101		1	→	1		202	Y		Y
9		101		1	→	1		203	Y		Y
10		102		2		1		201	Y		N
11		102		2	→	1		202	Y		N
12		102		2	→	1		203	Y		N

DATA treatment;

SET trt1

trt2 (IN=intrt2);

***)** Actual treatment will be set to planned treatment for the safety patients;

IF intrt2 and saffl eq 'Y' **THEN** trta = trtp;

RUN;



Correct programming code

```
DATA treatment;  
  SET trt1  
    trt2 (IN=intrt2);
```

*) Actual treatment will be set to planned treatment for the safety patients of data set TRT2 and will be set missing otherwise;

```
IF intrt2 THEN DO;  
  IF saffl eq "Y" THEN trta = trtp;  
  ELSE CALL missing (trta);  
END;  
RUN;
```

Desired Output

Data set Treatment

Obs	Subject Identifier	Planned Treatment	Actual Treatment	Time point (N)	ITT Population Flag	Safety Population Flag
1	1	1	1	201	Y	Y
2	1	1	1	202	Y	Y
3	1	1	1	203	Y	Y
4	2	2	.	201	Y	N
5	2	2	.	202	Y	N
6	2	2	.	203	Y	N
7	101	1	1	201	Y	Y
8	101	1	1	202	Y	Y
9	101	1	1	203	Y	Y
10	102	2	.	201	Y	N
11	102	2	.	202	Y	N
12	102	2	.	203	Y	N

Conclusion/Summary

1. Know your data
2. Visualize your data if possible
3. Apply GPP
4. Check your log file and create custom warnings and errors
5. Learn the modus operandi of SAS to avoid "invisible" errors

Kristina Zweier
Statistical Programmer I
+49-6196-7709-320
kzweier@clinipace.com
www.clinipace.com

THANK YOU FOR YOUR
ATTENTION!



CLINIPACE
WORLDWIDE