SESUG Paper 109-2019

"I'M WARNING YOU!"

THE MOST COMMON SAS CODING MISTAKES

Christine Whitaker

Winston-Salem State University

OVERVIEW

It has come to my attention that there are some very common coding mistakes that even the most seasoned programmers tend to make sometimes. This paper will briefly cover the ones that I have found to be the most common.

#1: MISSING SEMICOLONS: By far the most common error, in my opinion, is the missing semicolon. A missing semicolon causes SAS to misinterpret not only the statement where the semicolon is missing, but the statements that follow. Let's review the code below:

```
data student
  infile 'h:\WHERE\ncair146.txt' lrecl=294 pad missover;
    input @001 banrid    $09. /* Student Id */;


proc print data=student
  var banrid ;
```

In this Data step where a semicolon was left off the DATA statement, SAS did not detect the error. This statement is correct except for the missing semicolon. In this example, the missing semicolon results in the INFILE statement being assumed to be part of the DATA statement. The missing semicolon on the Proc Print statement causes SAS to read the 2 statements as a

single statement. Resulting in the VAR statement being read as an option to the procedure. Since there is no VAR option in Proc Print, the program fails and the log says.

ERROR 202-322:

The option or parameter is not recognized.

The SAS System stopped processing this step because of errors.

ERROR 76-322:

Syntax error, Statement will be ignored.

The SAS System stopped processing this step because of errors.

#2: MISSPELLED KEYWORDS:

Sometimes SAS will correct spelling mistakes for you by making its best guess at what you meant to do. When this happens, SAS will continue execution and issue a warning explaining the assumption it has made as in the example below.

```
dat student;
  infile 'h:\WHERE\ncair146.txt' lrecl=294 pad missover;
    input @001 item01    $09. /* Social Security Number */
          @010 item02     6. /* Fice Code */
          @027 item06    $01. /* Enrollment Status */ ;
```

SAS LOG

- **15363 dat student;**
- **---**
- **14**
- **WARNING 14-169: Assuming the symbol DATA was misspelled as data**

#3: MISSPELLED DATASET NAMES:

```
data sdf01;
infile 'h:\CGWDATA\NCAIR_TESTDATA_071515.TXT' lrecl=214 pad missover;
    input @001 item01   $01.
proc sort data=sdfO1 out=sort01;
  by item80;
```

SAS LOG RESULTS:

ERROR: File WORK.SDFO1.DATA does not exist.

865 by item80;

NOTE: The SAS System stopped processing this step because of errors.

WARNING: The data set WORK.SORT01 may be incomplete. When this step was stopped there were 0 observations and 0 variables.

WARNING: Data set WORK.SORT01 was not replaced because this step was stopped.

NOTE: PROCEDURE SORT used (Total process time):

real time 0.12 seconds

cpu time 0.00 seconds

#4: UNMATCHED QUOTES: SAS keeps a running total of the number of quotation marks in the code. In the windowing environment, an odd number of quotation marks causes the program to "hang".

```
data student;
  infile 'h:\WHERE\ncair146.txt lrecl=294 pad missover;
    input @001 banrid   $09. /* Student Id */;
```

Because the ending quotation mark is missing on the INFILE Statement, SAS just keeps running and ultimately produces the SAS log below.

NOTE: SAS STOPPED PROCESSING THIS STEP BECAUSE OF ERRORS.

NOTE: SAS WENT TO A NEW LINE WHEN INPUT STATEMENT REACHED PAST THE END OF A LINE.

- NOTE: The SAS System stopped processing this step because of errors.
- WARNING: The data set WORK.SDF01 may be incomplete. When this step was stopped there were 0 observations and 7 variables.
- WARNING: Data set WORK.SDF01 was not replaced because this step was stopped.
- NOTE: DATA statement used (Total process time):

#5 CODING TIP: When using the IF/THEN/ELSE statement, programmers frequently forget to consider the possibility of bad data. In the example belowItem10 may contain values other than M or F. Allowing for bad data, we tend to use poor logic like this example:

PROBLEM

IF ITEM10='F' THEN GENDER ='FEMALE';

IF ITEM10='M' THEN GENDER ='MALE';

ELSE GENDER='UNKNOWN';

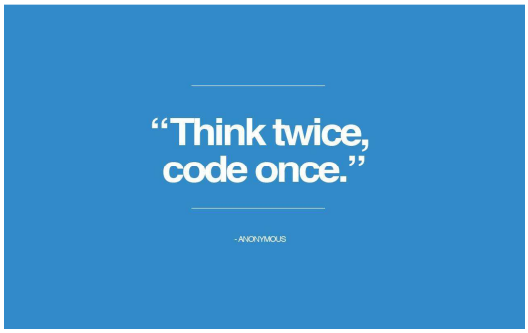
Since this program contains only one ELSE statement, Item10 will never be Female. In addition, any values of Female will get changed to Unknown.

SOLUTION

IF ITEM10='F' THEN GENDER ='FEMALE';

ELSE IF ITEM10='M' THEN GENDER ='MALE';

ELSE GENDER='UNKNOWN';



REFERENCES

http://support.sas.com/documentation/onlinedoc/91pdf/

ACKNOWLEDGEMENTS

Thanks to SAS online support for answering all my questions.

CONTACT INFORMATION

I welcome all comments and questions!

Christine Whitaker

Winston-Salem State University

Winston-Salem, NC 27110

(336)750-2115

whitakerc@wssu.edu