



# Hands-on Workshop: SAS<sup>®</sup> Program Debugging Challenge

Course Notes

*Hands-on Workshop: SAS® Program Debugging Challenge Course Notes* was developed by Peter Styliadis. Editing and production support was provided by the Curriculum Development and Support Department.

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. Other brand and product names are trademarks of their respective companies.

**Hands-on Workshop: SAS® Program Debugging Challenge Course Notes**

Copyright © 2018 SAS Institute Inc. Cary, NC, USA. All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

---

Book code E71221, course code S18HPDC, prepared date 17Apr2018.

S18HPDC\_001

# Table of Contents

|   |            |
|---|------------|
| To learn more.....                                      | iv         |
| <b>Chapter 1 SAS® Program Debugging Challenge .....</b> | <b>1-1</b> |
| 1.1 Introduction.....                                   | 1-3        |
| 1.2 Challenge Issues .....                              | 1-4        |
| 1.3 Challenge Hints .....                               | 1-6        |
| 1.4 Suggested Answer.....                               | 1-9        |

## To learn more...



For information about other courses in the curriculum, contact the SAS Education Division at 1-800-333-7660, or send e-mail to [training@sas.com](mailto:training@sas.com). You can also find this information on the web at <http://support.sas.com/training/> as well as in the Training Course Catalog.

For a list of SAS books (including e-books) that relate to the topics covered in this course notes, visit <https://www.sas.com/sas/books.html> or call 1-800-727-0025. US customers receive free shipping to US addresses.

# Chapter 1 SAS<sup>®</sup> Program Debugging Challenge

|     |                        |     |
|-----|------------------------|-----|
| 1.1 | Introduction .....     | 1-3 |
| 1.2 | Challenge Issues ..... | 1-4 |
| 1.3 | Challenge Hints .....  | 1-6 |
| 1.4 | Suggested Answer.....  | 1-9 |



# 1.1 Introduction

---

## Scenario

You are an analyst at the National Hurricane Center. Your department has been working on a project to create a program that will enable the department to increase efficiency when analyzing and reporting storms.

Currently, this process is done manually, but a previous analyst attempted to streamline the process using SAS. The analyst was close to finishing but has been pulled off the project.

With your SAS experience, your manager has tasked you with debugging the SAS program that the previous analyst wrote. However, this program contains logic and syntax errors. Lucky for you, the previous analyst has documented many of the issues.

## Resources

All the resources that you need for the challenge are in C:\Workshop\Challenge\ProgramDebug. The folder contains the following:

- **Storm Challenge** SAS program
- **Storm Challenge Backup** SAS program
- **Storm\_Details** SAS data set
- **Storm\_Lookup** SAS data set
- **Kahoot!** trivia link

## 1.2 Challenge Issues

The following issues have been documented throughout the creation of the **Storm Challenge.sas** program. Your tasks are to investigate, debug and address the issues in the program. It is recommended you follow the documented issues below to complete the challenge on time. If you are stuck, you can refer to the Challenge Hints section of the document for assistance.

### DATA Step Issues

1. The **path** macro variable was added to be used in the LIBNAME statement, but it returns a syntax error.

#### Partial Error Message

```

49
ERROR: Libref DATA is not assigned.
ERROR: Error in the LIBNAME statement.
NOTE 49-169: The meaning of an identifier after a quoted string might change in a future SAS release. Inserting white space
          between a quoted string and the succeeding identifier is recommended.
ERROR 22-7: Invalid option name C.

```

2. Review the format to ensure that there are no overlapping ranges.
3. There is a syntax error when using the format to create the **Category** variable.

#### Partial Error Message

```

46          Category=input(MaxWindSpeed,Stormcategory.);
          _____
          485
NOTE 485-185: Informat STORMCATEGORY was not found or could not be loaded.

```

4. The variable **last** needs to be a temporary variable that detects the end of the file, not a variable included in the final data set.
5. The **EndingYear** macro variable might not resolve (depending on how #4 was solved).
6. Data values for **MaxWindSpeed** and **MaxStatus** are not as expected in the data set. (Multiple steps are needed to fix the data values.)

#### Partial Answer **work.storms**

|   | StormKey    | Name    | MaxStatus | MaxWindSpeed | Start_Date | End_Date  | LengthofStor... | Category   |
|---|-------------|---------|-----------|--------------|------------|-----------|-----------------|------------|
| 1 | 1950ABLE    | ABLE    | HU        | 110          | 12AUG1950  | 24AUG1950 | 13              | Category 2 |
| 2 | 1950BAKER   | BAKER   | HU        | 90           | 18AUG1950  | 01SEP1950 | 15              | Category 1 |
| 3 | 1950CHARLIE | CHARLIE | HU        | 95           | 21AUG1950  | 05SEP1950 | 16              | Category 1 |
| 4 | 1950DOG     | DOG     | HU        | 125          | 30AUG1950  | 18SEP1950 | 20              | Category 3 |
| 5 | 1950EASY    | EASY    | HU        | 105          | 01SEP1950  | 09SEP1950 | 9               | Category 2 |

7. **MaxStatus** data values might be truncated (depending on how #6 was solved).



## Join Issue

1. Make sure to include all observations from the **work.storms** data set in the newly created **work.storm\_answer** data set.

Partial Answer **work.storm\_answer**

| StormKey  | Name  | MaxStatus | MaxWind | Category   | Start_Date | End_Date  | LengthofStorm |
|-----------|-------|-----------|---------|------------|------------|-----------|---------------|
| 1950ABLE  | ABLE  | Hurricane | 110     | Category 2 | 12AUG1950  | 24AUG1950 | 13            |
| 1950HIKI  | HIKI  | Hurricane | 75      | Category 1 | 12AUG1950  | 21AUG1950 | 10            |
| 1950BAKER | BAKER | Hurricane | 90      | Category 1 | 18AUG1950  | 01SEP1950 | 15            |

## Report Issues

1. The macro variables in the TITLE statement will not resolve.

Desired Title Example

Storms from 1950 to 2016

2. The footnote is not writing to output the exact time and date that the report is being run.

Desired Footnote Example (should be exact date and time that you created the report)

Report Generated on: January 22, 2018 at 15:36:19

3. Remove the warning regarding the units' value for left and right margin options.

## Validate your Results

To validate your results, run the validation code at the bottom of the **Storm Challenge.sas** program when you are finished. After you run the code, check the results and fill in the answers below. These validation results will be used in the end-of-class trivia.

1. How many total storms are in your final data set? \_\_\_\_\_
2. How many storms did not have a status description in the final data set (if any)?  
\_\_\_\_\_
3. How many category 3 storms were in the final data set? \_\_\_\_\_
4. What is the total average wind speed? \_\_\_\_\_
5. What was the issue with the footnote?  
\_\_\_\_\_

## The Challenge on SAS Support Communities

Not enough time to finish the challenge? Want to try the challenge again? Please visit the SAS Support Communities for the workshop files and challenge updates!

## 1.3 Challenge Hints

---

### DATA Step Issues

1. The **path** macro variable was added to be used in the LIBNAME statement, but it returns a syntax error.

```
/*Remove the quotes from the macro variable*/
%let path=C:\Workshop\Challenge\ProgramDebug;
```

2. Review the format to ensure that there are no overlapping ranges.

```
/*Make sure to fix the overlapping numeric ranges*/

proc format;
  value StormCategory /*Using MPH*/
    74 - 95 = 'Category 1'
    96 - 110 = 'Category 2'
    111 - 129 = 'Category 3'
    130 - 156 = 'Category 4'
    157 - high = 'Category 5'
    other = 'N/A';
run;
```

3. There is a syntax error when using the format to create the **Category** variable.

```
/*Replace the INPUT function with a PUT function*/
Category=put(MaxWindSpeed,Stormcategory.);
```

4. The variable **last** needs to be a temporary variable that detects the end of the file, not a variable included in the final data set.

```
/*Add the END= option in the SET statement*/
set data.Storm_Details end=last;
```

5. The **EndingYear** macro variable might not resolve (depending on how #4 was solved).

```
/*If you fix issue 4 by using the code above, this issue will be
resolved. The statement will use last as the end of file*/
if last then call symputx('EndingYear',year(date));
```

6. Data values for **MaxWindSpeed** and **MaxStatus** are not as expected in the data set. (Multiple steps are needed to fix the data values.)

#### Part 1

```
/*Add an IF-THEN/DO statement, and set MaxWindSpeed = . and
MaxStatus=" " the first time StormKey is encountered*/

if first.StormKey then do;
  Start_Date=date;
  MaxWindSpeed=.;
  MaxStatus=" ";
end;
```

#### Part 2

```
/*The IF statement needs a THEN-DO statement because you have
multiple assignment statements*/

if MaxWind > MaxWindSpeed then do;
  MaxWindSpeed=MaxWind;
  MaxStatus=Status;
end;
```

7. **MaxStatus** data values might be truncated (depending on how #6 was solved).

```
/*Depending on how you fixed issue 6, you might have truncated
values for MaxStatus. One way to fix the values is to use the
LENGTH statement below the SET statement*/

set data.Storm_Details end=last;
length MaxStatus $2;
```

## Join Issue

1. Make sure to include all observations from the **work.storms** data set in the newly created **work.storm\_answer** data set.

```
/*Change the inner join to a left join*/

from Storms as s left join data.Storm_lookup as l
```

## Report Issues

1. The macro variables in the TITLE statement will not resolve.

```
/*1. With macro variables, be sure to use double quotes*/  
/*2. Must also fix &EndingYear macro variable in the DATA Step  
Issues section*/  
  
title "Storms from &BeginningYear to &EndingYear";
```

2. The footnote is not writing to output the exact time and date that the report is being run.

```
/*The &sysdate9 and &systemtime automatic macro variables will  
resolve to the date and time the session was started. You will  
need to add two %sysfunc macro functions*/  
  
footnote "Report Generated on: %sysfunc(today()),worddate.) at  
%sysfunc(time()),time.)";
```

3. Remove the warning regarding the units' value for left and right margin options.

```
/*Add the units 'in' to both leftmargin and rightmargin*/  
  
options orientation=landscape nodate pageno=1 leftmargin=.1in  
rightmargin=.1in;
```

# 1.4 Suggested Answer

---

```

%let path=C:\Workshop\Challenge\ProgramDebug;
libname data "&path";

/*****
Saffir-Simpson Hurricane wind scale:
https://www.nhc.noaa.gov/aboutsshws.php
74-95 mph = Category 1
96-110 mph = Category 2
111-129 mph = Category 3
130-156 mph = Category 4
157+ mph = Category 5
All other storms = 'N/A'
*****/
proc format;
  value StormCategory /*Using MPH*/
    74 - 95 = 'Category 1'
    96 - 110 = 'Category 2'
    111 - 129 = 'Category 3'
    130 - 156 = 'Category 4'
    157 - high = 'Category 5'
    other = 'N/A';
run;

/*****
Fix the DATA Step
*****/
data Storms;
  set data.Storm_Details end=last;
  length MaxStatus $2;
  retain MaxWindSpeed MaxStatus Start_Date;
  by StormKey;
  if _n_=1 then call symputx('BeginningYear',year(date));
  if first.StormKey then do;
    Start_Date=date;
    MaxWindSpeed=.;
    MaxStatus=" ";
  end;
  if MaxWind > MaxWindSpeed then do;
    MaxWindSpeed=MaxWind;
    MaxStatus=Status;
  end;
  if last.StormKey then do;
    End_Date=Date;
    LengthofStorm=(End_Date - Start_Date)+1;
    Category=put(MaxWindSpeed,Stormcategory.);
  end;

```

```

        output;
    end;
    if last then call symputx('EndingYear',year(date));
    format start_date end_date date9.;
    drop Date Status MaxWind;
run;

/*Test macro variables*/
%put &=BeginningYear;
%put &=EndingYear;

/*****
Join the Storms data set with data.Storm_Lookup data set and create
the final data set.
*****/
proc sql;
create table Storm_Answer as
select StormKey, Name, l.StatusDescription as MaxStatus,
       MaxWindSpeed as MaxWind, Category, Start_Date,
       End_Date, LengthofStorm
  from Storms as s left join data.Storm_Lookup as l
     on s.MaxStatus = l.Status
  order by Start_Date;
quit;

/*****
Print the report.
*****/
options orientation=landscape nodate pageno=1 leftmargin=.1in
        rightmargin=.1in;

ods pdf file="&path\FinalReport.pdf";

title "Storms from &BeginningYear to &EndingYear";
footnote "Report Generated on:%sysfunc(today()),worddate.) at
%sysfunc(time()),time.)";

proc print data=Storm_Answer noobs;
run;

title;
footnote;
ods pdf close;

```

```

/*****
Validation Answers
*****/
1.    How many total storms are in your final data set?

1612

2.    How many storms did not have a status description in the final
data set (if any)?

1

3.    How many category 3 storms were in the final data set?

147

4.    What is the total average wind speed?

73.3

5.    What was the issue with the footnote?

&sysdate and &systemtime display session start time.

```

