



PLUSPROJECT
PARTNERSHIP

EMPLOYEE
OWNED
BIostatISTICS
CRO

Know Your Environment

Will Greenway

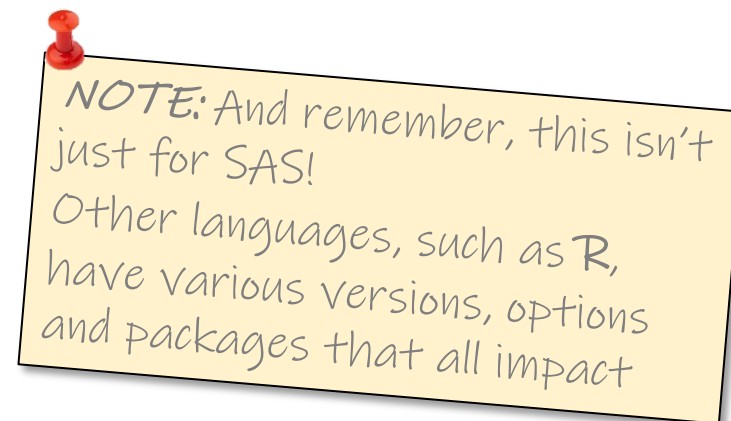
Principal Programming Specialist

Contents

Challenges Based on Real-life Scenarios

After this session, hopefully you'll be encouraged to **know your**:

- **Product & Versions**
- **Software Options**
- **Operating System**
 - Terminating Strings
 - Referencing Files



*NOTE: And remember, this isn't just for SAS!
Other languages, such as R, have various versions, options and packages that all impact*

Product & Versions

Scenario #1:

- Stats analysis is to be dual-programmed.
- The Procedure **proc power** is being used and the parameters are known up-front.
- Both Programmers are logged into slightly different Virtual Desktops, but both use Windows and have PC SAS v9.4 installed.
- Programmer 1 has no issues.
- Programmer 2 is receiving the following issues in their log:

```
1 proc power;  
2   twosamplefreq test =   FM_RR  
                           -----  
                           1  
WARNING 1-322: Assuming the symbol FM was misspelled as FM_RR.  
8     ...;  
9 run;  
ERROR: The NULLRELATIVERISK option is invalid for TEST=FM.  
ERROR: The GROUPPROPORTIONS and NULLRELATIVERISK options cannot be used simultaneously.
```

NOTE: ... in the code is for simplicity only, other parameters were correctly specified.

Product & Versions

Troubleshooting:

- Windows PC SAS v9.4 was confirmed as being used on both.
- BUT, **proc power** is part of the SAS/STAT sub-package.
- The following was run to check versions of SAS/STAT:

```
11 proc product_status;  
12 run;
```

- This highlighted differences:
 - Programmer 1 had SAS/STAT 14.3
 - Programmer 2 had SAS/STAT 13.2
- Documentation checked and confirmed that SAS/STAT 13.2 does not have FM_RR as a valid value of test=, which is why SAS was assuming FM_RR was instead a misspelling of FM

Product & Versions

Lessons:

- Version of BASE SAS should always be considered when programming.
- BUT that is not all! If using functions/procedures from a sub-products or package (e.g. SAS/STAT), it is worth checking which version(s) you are using – there are differences!
- Same is applicable for other packages, e.g. SAS/GRAPH, SAS/AF and SAS/ACCESS Interface for PC Files.

Remember: Your SAS version goes beyond the BASE SAS version. Learn what packages are installed and their respective versions.

Software Options

Scenario #2:

- External Excel file received. Columns headers have spaces and special characters.
- Initial SAS program successfully converts this to a permanent SAS dataset.
- Another SAS program reads in the permanent dataset to produce an output.
- Initial run of both programs was fine and no issues in the log.
- The next day re-running the 2nd program alone gave the following in the log:

```
28      data x;  
29          set util.titles;  
ERROR: The value 'REPORT NUMBER'n is not a valid SAS name.  
30      run;
```

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

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

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

NOTE: DATA statement used (Total process time):

real time 0.00 seconds

cpu time 0.01 seconds



But it worked yesterday???

Software Options

Troubleshooting:

- For the initial creation of the two programs, both were created by the same person during the same SAS session.
- The next day, only the second program needed re-running and this was done in a new SAS session.
- Checking the program that created the permanent dataset, the SAS option `validvarname` was changed to `any`.
- SAS system autoexec had set `validvarname=upcase`.
- Value of this option above confirmed in current erroring session by running:

```
16 proc options option=validvarname;  
17 run;
```

Software Options

Scenario #3:

- SDTM program written for DM/SUPPDM.
- Code was based on a previous study in the same compound for consistency.
- Running the program gave QNAM values in lowercase, however this was not an issue in the previous study.

STUDYID	RDOMAIN	USUBJID	IDVAR	IDVARVAL	QNAME	QLABEL	QVAL	QORIG	QEQAL
CDISC01	DM	CDISC01.100008			raceoth	Race, Other	EASTERN INDIAN	CRF	
CDISC01	DM	CDISC01.100008			rand	Randomized Population Flag	Y	CRF	
CDISC01	DM	CDISC01.100008			randno	Randomization Number	0004	CRF	
CDISC01	DM	CDISC01.100008			safety	Safety Population Flag	Y	Derived	CLINICAL STUDY SPONSOR

...but it worked on the last study, and the code is pretty much the same?!

Software Options

Lessons:

- SAS options that are changed are retained in the current SAS session.
- Changed SAS options are not retained into new SAS sessions.
- Changed SAS options should ideally be reverted after use (particularly in macros!).
- Consider all implications of changing the option (*in this case, creating a SAS dataset that was accurate to the Excel but not readable using the system's default options*).

 Remember: Different systems will have different defaults. To make code transportable, ideally program to the most restrictive (within reason...).

Software Options

Troubleshooting:

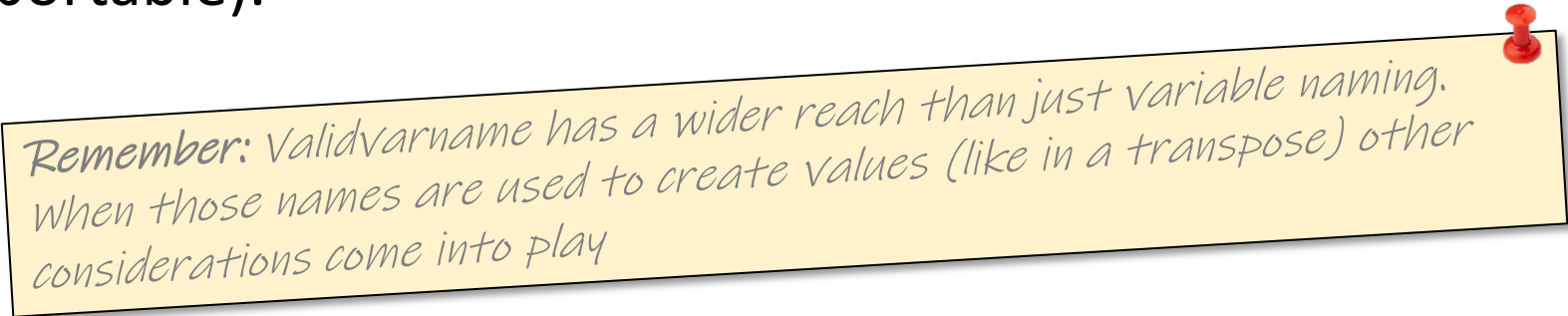
- Program was checked vs. the previous study, the code was (almost) the same.
- A **proc transpose** was used to create the SUPPDM.
- QNAM was created directly from _NAME_ without `upcase` function.
- Code creating the original variables were written in lowercase.
- Study-level autoexecs were present, the previous study used `validvarname=upcase`, while the newer study used `validvarname=v7`.
- Again, Value of this option above confirmed in current study by running:

```
16 proc options option=validvarname;  
17 run;
```

Software Options

Lessons:

- Casing for variable names often doesn't matter, but sometimes it does.
- Options like `validvarname` can have indirect impacts on the output of **`proc transpose`**, and the SASHELP (or dictionary) datasets.
- If you are relying on a certain option, set it or make sure it the value you expect.
- Account for sessions where it may not be true (adding the `upcase` function when creating QNAM would have made the code more transportable).



Remember: validvarname has a wider reach than just variable naming. When those names are used to create values (like in a transpose) other considerations come into play

Operating System (Terminating Strings)

Scenario #4:

- Unix SAS v9.4 is being used.
- External CSV (comma separated ASCII text) file received and needs to be read in.
- Code written using **proc import** to import and includes the option `getnames=yes`.
- Issues present in the last column:
 - Variable was oddly named (VAR3), losing the actual name of the last column.
 - Column should have been numeric but was being read in as character.

Operating System (Terminating Strings)

Troubleshooting:

- Changed to using data step `infile` statement to import, including header row.
- SAS `rank()` function used to look for non-standard ASCII characters in the last column.
- Using the above, it was identified that the last column always ended in a Carriage Return [*CR, byte 13*]
- The `infile` statement was adjusted to add the option `termstr=crlf`, which resulted in the issues resolving.



Operating System (Terminating Strings)

Scenario #5:

- Windows SAS v9.4 is being used.
- External CSV (comma separated ASCII text) file received and needs to be read in.
- Code written using `infile` statement for import.
- Normally this was fine, but for one version there were issues:
 - Only the first row was read in and the last column of row 1 appeared concatenated with the first column of the next row.
 - Log showed Import issues:

```
Unable to sample external file, no data in first 5 records.  
ERROR: Import unsuccessful. See SAS Log for details.
```

Operating System (Terminating Strings)

Troubleshooting:

- Vendor was queried to find out if anything changed between versions.
 - It was reported that a different user created the file this time, using a Mac instead of a Windows PC.
- Based on previous Scenario, file was also inspected to check what terminating strings were being used.
- The `infile` statement was adjusted to add the option `termstr=lf`, which resulted in the issues resolving.

Operating System (Terminating Strings)

Lessons:

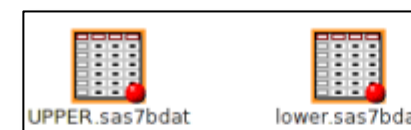
- Be aware that different OS have different defaults for specifying new lines (Windows: CR+LF, Unix: LF).
- The `infile`, `file` and `filename` statements allow you to specify the terminating (new line) string for files which can be adjusted to read files correctly.

*Remember: Consider which OS you are using, and which OS may have been used to create external files. If in doubt and you run into issues, have a play with the various options! **NO NEED TO CHANGE THE INPUT FILE***

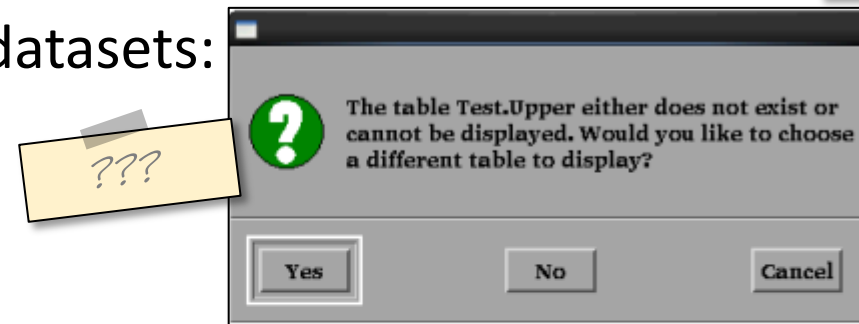
Operating System (Referencing Files)

Scenario #6:

- Unix SAS v9.2 is being used.
- External vendor created and sent .sas7bdat files:
- SAS `libname` created and Explorer Window shows all files present:
- However, SAS cannot open 1 of the datasets:



Contents of 'Test'	
Name	
	Lower
	Upper



- And produces error messages in the log when reading into a data step:

```

26  data test;
27    set test.lower test.upper;
ERROR: File TEST.UPPER.DATA does not exist.
28  run;

NOTE: The SAS System stopped processing this step because of errors.
WARNING: The data set WORK.TEST may be incomplete.  When this step was stopped there were 0 observations and 1 variables.
NOTE: DATA statement used (Total process time):
      real time           0.02 seconds
      cpu time            0.02 seconds
  
```



Operating System (Referencing Files)

Troubleshooting:

- The actual issue was less obvious than the example on the previous slide! There was only 1 uppercase letter part way through the filename.
- Various things were checked, including for non-standard ASCII characters, but these found no issues.
- Confirmed with Vendor how they created the files:
 - Their process included creating valid .sas7bdat files from SAS, but then renaming the files for export to Clients.
- Filename was shortened to 1 character, and it was read in fine!
- Researched for .sas7bdat filename restrictions and came across:

Usage Note 15989: In UNIX environments, the names of SAS® data sets and external files must contain only lowercase characters

- Confirmed that changing filename to all lowercase worked.
- Updated version using only lowercase naming requested from Vendor.

Operating System (Referencing Files)

Lessons:

- Filename conventions between different Operating Systems differ.
- But also SAS plays a part and can add further complexities!
- A “simple rename” may not be as simple as that if you are sending files elsewhere...
- SAS Support is an extremely useful resource, particularly for researching or confirming issues.

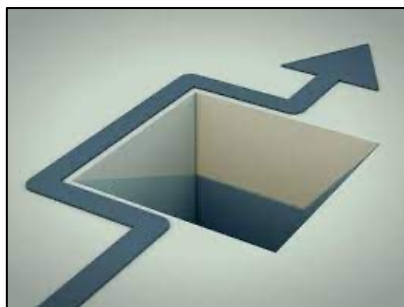


Remember: If in doubt, use lowercase filenames for everything (also this aligns with some regulatory requirements anyway)

Is that all?

Not by a long shot! This is just a *tiny glimpse* of the numerous things that can impact programs.

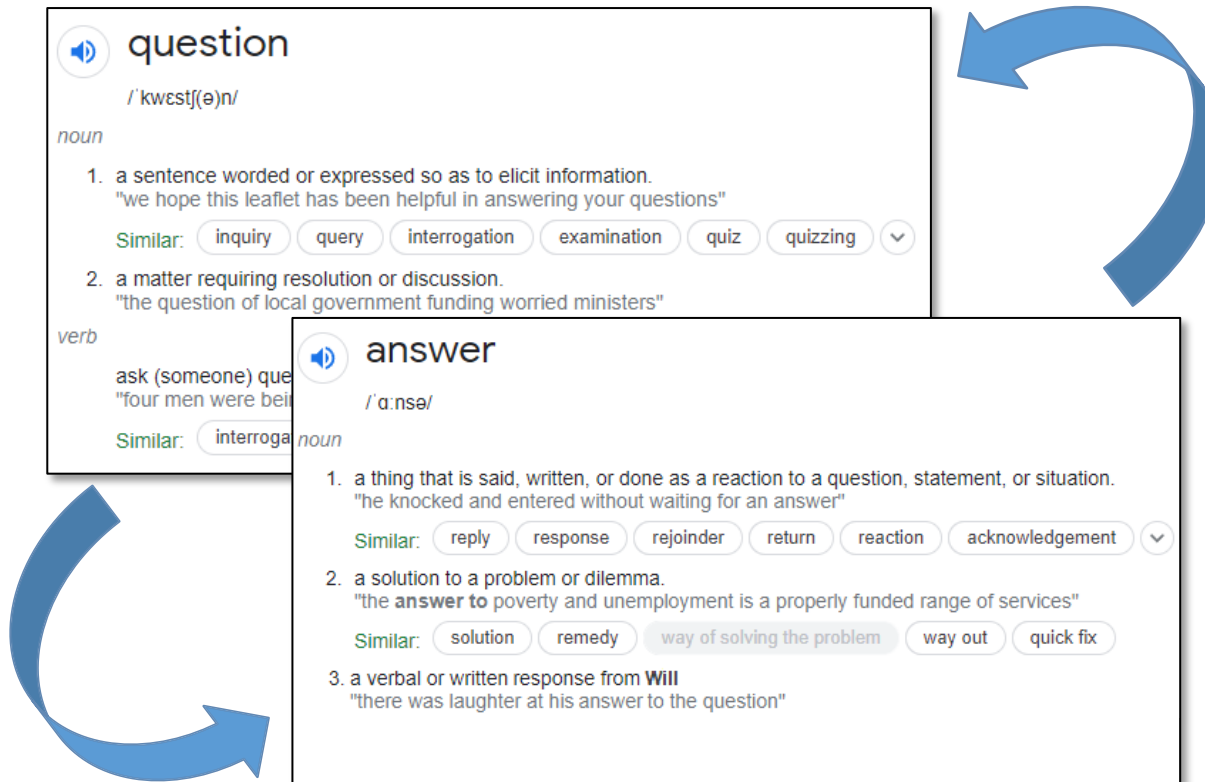
- There are MANY other SAS system options, I've only mentioned 1!
- Encoding differences (*e.g. wlatin1 vs UTF-8*)
- Will `dm` or `window` work?
- Is `x` enabled? Which commands work? Alternate method needed?



The more you understand about your environment and difference that can occur, the faster you can debug, or **proactively code to avoid**.

Remember: *Know your environment*

Questions?



question
/ˈkwɛstʃ(ə)n/
noun

1. a sentence worded or expressed so as to elicit information.
"we hope this leaflet has been helpful in answering your questions"

Similar: inquiry query interrogation examination quiz quizzing

verb

ask (someone) question
"four men were being asked questions"

Similar: interrogate

answer
/ˈɑːnsə/
noun

1. a thing that is said, written, or done as a reaction to a question, statement, or situation.
"he knocked and entered without waiting for an answer"

Similar: reply response rejoinder return reaction acknowledgement

2. a solution to a problem or dilemma.
"the **answer** to poverty and unemployment is a properly funded range of services"

Similar: solution remedy way of solving the problem way out quick fix

3. a verbal or written response from **Will**
"there was laughter at his answer to the question"



**PLUSPROJECT
PARTNERSHIP** | EMPLOYEE OWNED
BIOSTATISTICS
CRO

Will Greenway
Principal Programming Specialist

 willgreenway
 will.greenway@plus-project.co.uk
 www.plus-project.co.uk

MANCHESTER • EDINBURGH • LEEDS • WARSAW • KIEV