SESUG 2022 Paper 147

# Acing Technical Questions in SAS Programmer Interviewers

Lauren Rackley

## ABSTRACT

Part of the interview process for SAS programming positions often involves technical assessments or technical questions. Important topics to know are SQL joins, ways to create macro variables, by group processing, and correct use of the RETAIN statement. In an effort to help candidates getting started as a SAS programmer, this paper will provide guidance on how to tackle technical questions and technical assessments given during interviews.

## INTRODUCTION

When hiring programmers, managers may assess SAS ability in the interview if it is required to perform the day-to-day job duties. Interviews can take on several different formats now from being asked technical questions in a phone or panel interview, writing code in a Teams chat box, completing a data exercise and sending it back within a specified timeframe, writing code on a paper and sending a picture of it to the interviewer, and spotting errors in SAS code. Code samples may also be requested and the interviewer could ask a candidate to walk through the code sample in a call. Examples provided in this paper were based on actual interviews I had in the past few years in a variety of industries, such as pharmaceuticals to public sector. A caveat for pharmaceutical industry is to expect CDISC questions. This paper is intended to give guidance on how to answer such questions. As a disclaimer, this paper does not guarantee that these questions will crop up in an interview or that an offer will be extended. Of course, being able to answer technical questions correctly can demonstrate SAS proficiency.

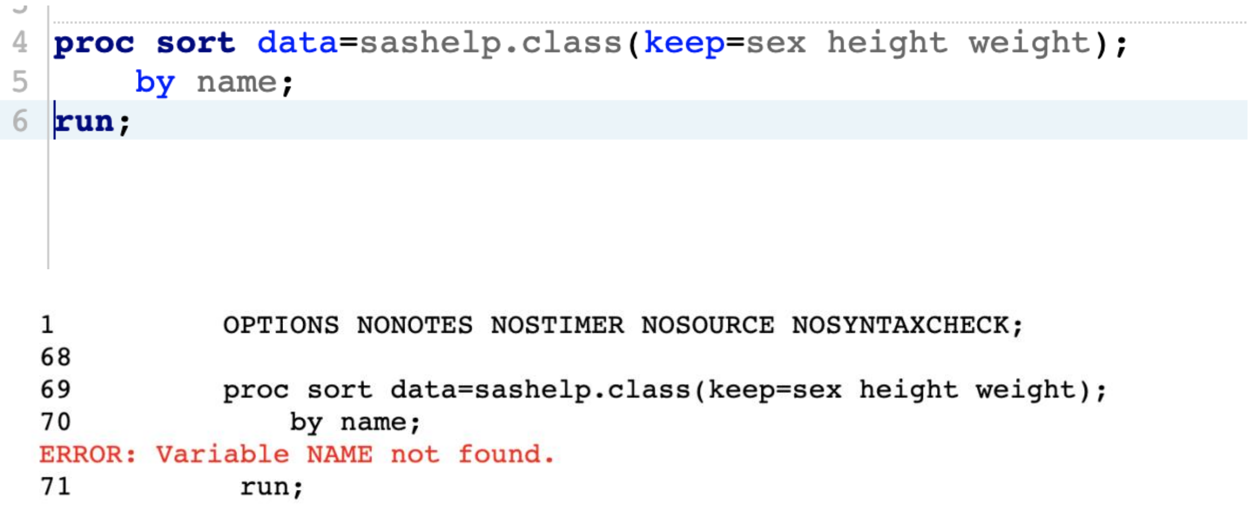
## COMMON INTERVIEW TOPICS

PROC SQL has come up in past interviews. Some people feel strongly about only using the DATA step. Some love PROC SQL. Interviewers may ask your preference or which one is best for a specific scenario. Questions about different joins are also common. If PROC SQL is mentioned in the job description, then be sure you can explain basic joins in SQL. You may also want to study the set operators.

The RETAIN statement is very powerful so don't be surprised if you get asked what it does. If the role you are interviewing for involves statistics, then you may get asked statistical questions. Examples include:

- What is a censor in survival analysis?
- What type of test do you use to get a p-value for a categorical value? Continuous variable?
- What type of response variable is used in logistic regression? What is the syntax of proc logistic?

Other questions may center around how to merge data or create macro variables. Merging datasets and creating macro variables are essential skills that every programmer should have. An example of testing whether someone can merge data is presenting screenshots of sample data and asking the candidate to write out the steps of how to combine the datasets. You may also be asked about differences between combining data horizontally and vertically. Merge statement is used to combine datasets horizontally. Set statement within a dataset will stack datasets on top of each other. Data could also be combined with joins or set operators in PROC SQL. In my experience, I have been asked more than once what ways there are to create macro variables in SAS. There are plenty of SAS papers that discuss all the ways to create macro variables.



```
proc sort data=sashelp.class(keep=sex height weight);
    by name;
run;
```

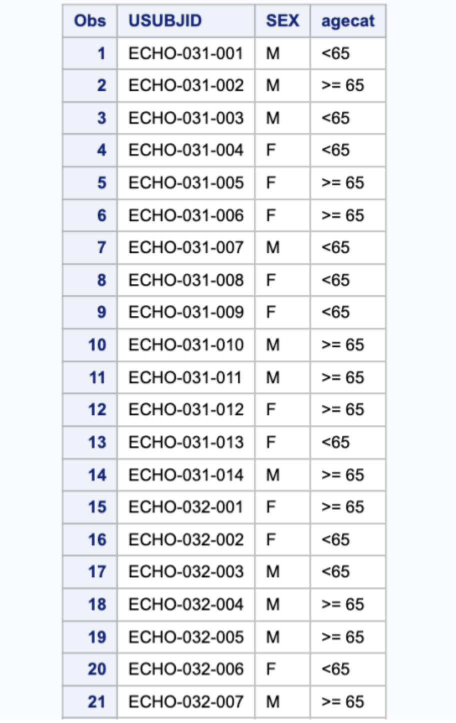
```
1          OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;
68
69         proc sort data=sashelp.class(keep=sex height weight);
70             by name;
ERROR: Variable NAME not found.
71          run;
```

**Figure 1. Spotting the error in a SAS program**

One type of question you may come across is spotting the error in a SAS program. Example code is given and a portion of the log is displayed. In this case, the error is generated from the name variable being left out of the keep statement. To remove the error, name variable would have to be added to the keep statement.

## SAMPLE QUESTION FROM A DATA EXERCISE

A dataset may be provided and you may be asked to provide the code to accomplish a programming task.



| Obs | USUBJID | SEX | agecat |
|---|---|---|---|
| 1 | ECHO-031-001 | M | <65 |
| 2 | ECHO-031-002 | M | >= 65 |
| 3 | ECHO-031-003 | M | <65 |
| 4 | ECHO-031-004 | F | <65 |
| 5 | ECHO-031-005 | F | >= 65 |
| 6 | ECHO-031-006 | F | >= 65 |
| 7 | ECHO-031-007 | M | <65 |
| 8 | ECHO-031-008 | F | <65 |
| 9 | ECHO-031-009 | F | <65 |
| 10 | ECHO-031-010 | M | >= 65 |
| 11 | ECHO-031-011 | M | >= 65 |
| 12 | ECHO-031-012 | F | >= 65 |
| 13 | ECHO-031-013 | F | <65 |
| 14 | ECHO-031-014 | M | >= 65 |
| 15 | ECHO-032-001 | F | >= 65 |
| 16 | ECHO-032-002 | F | <65 |
| 17 | ECHO-032-003 | M | <65 |
| 18 | ECHO-032-004 | M | >= 65 |
| 19 | ECHO-032-005 | M | >= 65 |
| 20 | ECHO-032-006 | F | <65 |
| 21 | ECHO-032-007 | M | >= 65 |

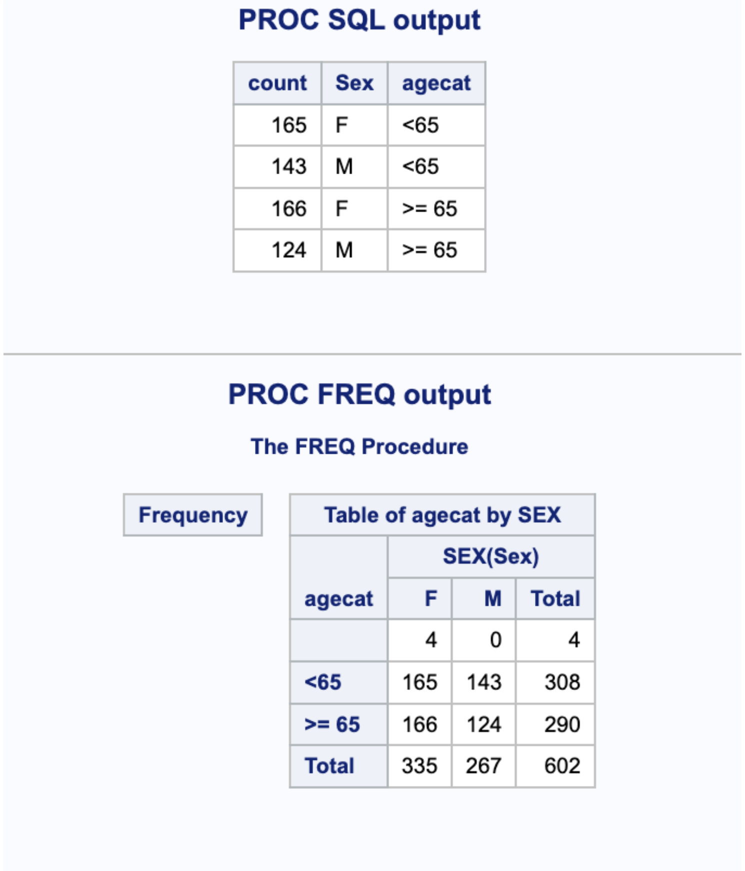
**Figure 2 Sample data for programming task**

A question could be: Find the number of subjects by age group and sex. The following code blocks present 2 ways to do that:

```
proc sql;
  select count(distinct usubjid) as count, sex, agecat
    from age
   where missing(agecat)=0
   group by agecat,sex;
 quit;

proc freq data=age;
  tables agecat*sex / norow nocol nopercent missing;
run;
```

Sample output from the above code blocks will be provided below:



**PROC SQL output**

| count | Sex | agecat |
|---|---|---|
| 165 | F | <65 |
| 143 | M | <65 |
| 166 | F | >= 65 |
| 124 | M | >= 65 |

**PROC FREQ output**

**The FREQ Procedure**

| Frequency |
|---|

| Table of agecat by SEX | | | |
|---|---|---|---|
| | SEX(Sex) | | |
| agecat | F | M | Total |
| | 4 | 0 | 4 |
| <65 | 165 | 143 | 308 |
| >= 65 | 166 | 124 | 290 |
| Total | 335 | 267 | 602 |

**Figure 3 Sample output for programming task**

Both code approaches found the number of subjects by age group and sex. Make note of the missing statement used in the proc freq approach. An interviewer could ask if you could proc freq can handle missing values. Being able to generate counts is an important task for summary reports. If you end up being a statistical programmer, you will often have to get n (%) in your tables or other summary statistics. You may be asked how you would create a certain table, such as which procs you would use or there could even be a test where datasets are given and you have to write a SAS program to produce the table itself and send it to the interview. Many companies may have proprietary macros for table generation but PROC MEANS, PROC UNIVARIATE, PROC SQL, PROC FREQ could all work for tables involving n (%) or summary statistics, such as Mean (SD), Median, Min, Max, Q1, Q3, etc.

Expect questions to range in difficulty depending on what position you are applying for. Most of the questions presented in this paper have been in interviews for someone with 3 years of experience or less.

## CONCLUSION

SAS technical interviews are nothing to be scared of. Make sure you know basic SAS syntax and exude confidence. Think carefully before answering. Do not panic if you can't finish a timed test or data exercise that has to be completed within a specified timeframe. In these situations, the thought process or proposed programming approach will still be valued. Also remember, that if you were rejected from a company, it does not mean you were terrible. You can be rejected and still receive positive feedback. It is not a reflection of you.

## REFERENCES

McNichol, C. 2018. "Are You Ready for It? Preparing for Your Next Technical Interview." *Proceedings of the PharmaSUG 2018 Conference*, Seattle, WA.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Lauren Rackley
Lauren.g.rackley@gmail.com