

A Brief Comparison of the SAS and Python Languages

Vince DelGobbo

Questions & Answers from BASUG Webinar April 20, 2022

There was not sufficient time during the webinar for Vince to answer all of the submitted questions. He has provided written responses to the unanswered questions below.

Q: What made you want to become an independent consultant?

A: I wanted to take on a variety of short-term projects to get exposed to different things. But currently I'm engaged with something that might be a long-term project.

Q: Number of guessing rows: does Python have the 'guessingrows' problem as in proc import 'xlsx' and get the wrong variable type when performing automatic variable creation for list imports?

A: Documentation for pandas read_excel is here:

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_excel.html

It is unclear what the `nrows` argument does, and that would be a good thing for you to experiment with in your Python journey. It looks like you can use the `dtype` argument to explicitly specify data types.

Q: Slide 15, what is the benefit of having "quit;" after each "run;"? Thanks!

A: SAS implemented run-group processing in Version 6. Procedures that support run-group processing require a QUIT statement to end the execution. At first only a few procedures supported this, but the number has grown over the years. Because I can't remember which procedures require a QUIT statement, I always use one. I know you were expecting to learn a secret "SAS gem", but sorry ;-)

Q: Each Python statement is parsed by a carriage return?

A: Yes

Q: On page 19: how does Python maintain alignment between the x & y dataframes?

A: I'm not sure what the question is, but the answer might be, "The same way that SAS does". The "y" and "X" DataFrames both contain the same number of rows but different columns. The Python code to create the "y" DataFrame is similar to this SAS code:

```
data work.y ;  
    set work.reaction ;  
    keep ReactionPercentage ;  
run ;
```

The SAS code to create the "X" DataFrame is similar, with the difference being the variables specified in the KEEP statement.

Q: Anything Python can do and SAS cannot?

A: Maybe.

Q: Vince, do you offer any online training courses? Which individuals can be pay to join? Are you teaching any courses prior to sas conference at austin? etc

A: I don't offer training courses. I can refer you to Russ Lavery's Python training course offered by

BASUG next month, and Kirk Lafler for SAS topics. Here is a list of pre- and post-conference training sessions at PharmaSUG 2022 in Austin:

<https://www.pharmasug.org/us/2022/training.html>

Q: Does PYTHON run faster than SAS?

A: I haven't performed a performance analysis so I can't comment on this. If you have Python installed on the same machine as SAS, then you can do the analysis. You can probably find some performance benchmark SAS programs on lexjansen.com, run the code in SAS, and then convert the code to Python and run it. What a great way to learn Python, and maybe you can give a talk on your findings.

Q: How close the results generated by SAS models vs Python models?

A: I didn't perform an in-depth comparison, but they looked comparable. I have provided the data, SAS code, and Python code, so you can do the analysis yourself and maybe present your findings in a future meeting.

Q: Are there some things that can be done in SAS, but not in Python?

A: Maybe.

Q: Is it possible to customize the result output

A: Sorry, I don't know.