



13-16 November  
ICC Belfast

EU 2022

The Clinical Data  
Science Conference

# SAS and Python Working Together

Jim Box

Principal Data Scientist, HLS R&D

SAS Institute

Coding Tips & Tricks CT13

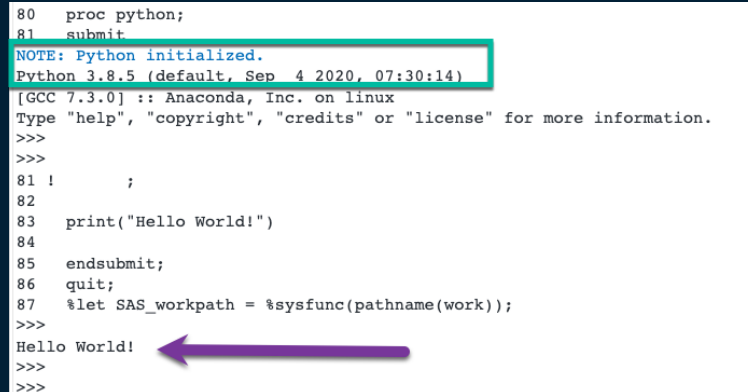
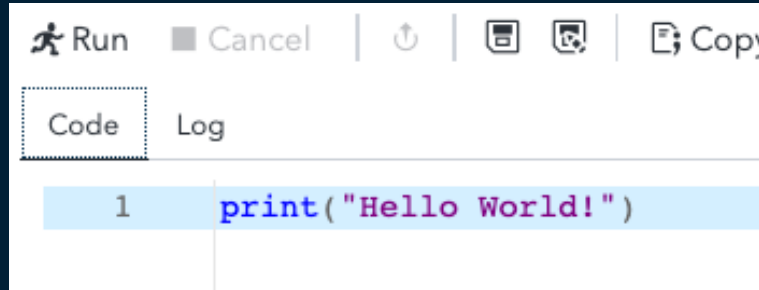
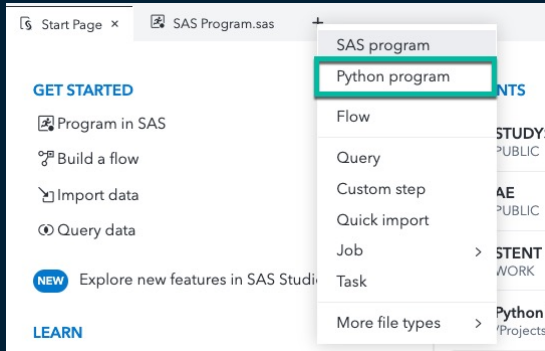


# SAS and Python

Using Python in a SAS Session

# SAS Viya and Python

## Write Python Programs in SAS Studio



# SAS Viya and Python

## PROC Python

- Run Python code within a SAS session
- Call most SAS functions within Python statements
- Submit SAS code from Python
- Move data between SAS datasets and Pandas dataframes
- Transfer values between SAS macro variables and Python variables

# SAS Viya and Python

## PROC Python – Run SAS in Python

```
1  /* Define a SAS macro variable in SAS code;
2  %let language = 'python';
3
4  proc python;
5  submit;
6
7  print("Python in the SAS Log:")
8
9  # /*use symget to read a SAS macro variable into a python variable;
10 lang = SAS.symget('language')
11 ver = 3.8
12
13 # /* Submit SAS code inside python, using python syntax. This dataset will live in WORK library;
14 SAS.submit("data work.test; language={}; version={}; run;".format(lang, ver))
15
16
```

# SAS Viya and Python

## PROC Python – Run SAS in Python

```
17  # -*- Execute SAS functions with sasfnc;
18  var3 = SAS.sasfnc("upcase", "hello world")
19  print( var3)
20
21  # -*-Use symput to assign the value of a Python variable to a SAS macro;
22  py_var = 'Inside python'
23  SAS.symput('macrovar', py_var)
24
25
26  endsubmit;
27  run;
28
29  -*- Show that the SAS macro variable persists and is populated;
30  %put &=macrovar;
31
```

# SAS Viya and Python

## PROC Python – Run SAS in Python

```
83  proc python;  
84  submit
```

**NOTE: Python initialized.**

Python 3.8.5 (default, Sep 4 2020, 07:30:14)

[GCC 7.3.0] :: Anaconda, Inc. on linux

Type "help", "copyright", "credits" or "license" for more information.

```
>>>
```

# SAS Viya and Python

## PROC Python – Run SAS in Python

```
108 data work.test; language='python'; version=3.8; run;  
NOTE: The data set WORK.TEST has 1 observations and 2 variables.  
NOTE: DATA statement used (Total process time):  
      real time           0.00 seconds  
      cpu time            0.00 seconds  
  
>>>
```

# SAS Viya and Python

## PROC Python – Run SAS in Python

```
>>>  
Python in the SAS Log:  
HELLO WORLD  
>>>
```

```
>>>  
NOTE: PROCEDURE PYTHON used (Total process time):  
      real time           5.44 seconds  
      cpu time            0.02 seconds
```

# SAS Viya and Python

## PROC Python – Run SAS in Python

```
110  /* Show that the SAS macro variable persists and is populated;  
111  %put &=macrovar;  
    MACROVAR=Inside python
```

# SAS Viya and Python

## PROC Python – run Python in SAS

```
proc python;  
submit;  
  
import pandas as pd  
URL = "https://en.wikipedia.org/wiki/List_of_Super_Bowl_champions"  
  
tables = pd.read_html(URL, attrs = {'class' : 'wikitable sortable'})  
sb = tables[0]  
sb.head()  
  
ds = SAS.df2sd(sb, 'work.sb')  
  
endsubmit;  
run;
```

Use Pandas to read in a table from wiki  
page

Output to SAS dataset

# SAS and Open Source

## Use Case: Pull Data from Wikipedia

| Super Bowl championships |                                         |                                                            |         |                                                 |                                      |                                                  |              |                   |         |
|--------------------------|-----------------------------------------|------------------------------------------------------------|---------|-------------------------------------------------|--------------------------------------|--------------------------------------------------|--------------|-------------------|---------|
| Game ↕                   | Date/Season ↕                           | Winning team ↕                                             | Score ↕ | Losing team ↕                                   | Venue ↕                              | City ↕                                           | Attendance ↕ | Referee ↕         | Ref     |
| I<br>[sb 1]              | January 15, 1967 (1966<br>AFL/1966 NFL) | Green Bay Packers <sup>n</sup><br>(1, 1–0)                 | 35–10   | Kansas City<br>Chiefs <sup>a</sup><br>(1, 0–1)  | Los Angeles Memorial<br>Coliseum     | Los Angeles,<br>California <sup>[sb 2]</sup>     | 61,946       | Norm<br>Schachter | [7][8]  |
| II<br>[sb 1]             | January 14, 1968 (1967<br>AFL/1967 NFL) | Green Bay Packers <sup>n</sup><br>(2, 2–0)                 | 33–14   | Oakland Raiders <sup>a</sup><br>(1, 0–1)        | Miami Orange Bowl                    | Miami, Florida <sup>[sb 3]</sup>                 | 75,546       | Jack Vest         | [9][8]  |
| III<br>[sb 1]            | January 12, 1969 (1968<br>AFL/1968 NFL) | New York Jets <sup>a</sup><br>(1, 1–0)                     | 16–7    | Baltimore Colts <sup>n</sup><br>(1, 0–1)        | Miami Orange Bowl (2)                | Miami, Florida (2) <sup>[sb 3]</sup>             | 75,389       | Tom Bell          | [10][8] |
| IV<br>[sb 1]             | January 11, 1970 (1969<br>AFL/1969 NFL) | Kansas City Chiefs <sup>a</sup><br>(2, 1–1) <sup>[S]</sup> | 23–7    | Minnesota<br>Vikings <sup>n</sup><br>(1, 0–1)   | Tulane Stadium                       | New Orleans, Louisiana                           | 80,562       | John<br>McDonough | [11][8] |
| V                        | January 17, 1971 (1970)                 | Baltimore Colts <sup>A</sup><br>(2, 1–1)                   | 16–13   | Dallas Cowboys <sup>N</sup><br>(1, 0–1)         | Miami Orange Bowl (3)                | Miami, Florida (3) <sup>[sb 3]</sup>             | 79,204       | Norm<br>Schachter | [12][8] |
| VI                       | January 16, 1972 (1971)                 | Dallas Cowboys <sup>N</sup><br>(2, 1–1)                    | 24–3    | Miami Dolphins <sup>A</sup><br>(1, 0–1)         | Tulane Stadium (2)                   | New Orleans, Louisiana<br>(2)                    | 81,023       | Jim Tunney        | [13][8] |
| VII                      | January 14, 1973 (1972)                 | Miami Dolphins <sup>A</sup><br>(2, 1–1)                    | 14–7    | Washington<br>Redskins <sup>N</sup><br>(1, 0–1) | Los Angeles Memorial<br>Coliseum (2) | Los Angeles, California<br>(2) <sup>[sb 2]</sup> | 90,182       | Tom Bell          | [14][8] |

[https://en.wikipedia.org/wiki/List\\_of\\_Super\\_Bowl\\_champions](https://en.wikipedia.org/wiki/List_of_Super_Bowl_champions)

# SAS and Open Source

## Use Case: SAS cleans data

```
data SB2;
set SB;
  SB = _N_;
  Season = SB + 1965;
  if Attendance in ('TBD', 'Attendance') then delete;
  fans = input(Attendance,8.);

  Winner = substr('Winning team'n,1,index('Winning team'n,'(')-2);
  Loser = substr('Losing team'n,1,index('Losing team'n,'(')-2);
  WS = input(substr(Score,1,2),2.);
  points = trim(left(scan(score,1,' ')));
  LS = input(substr(points,6,2),2.);

  OT = index(Score,"OT")>0;
  paren = index(City,'(');
  bracket = index(City,'[');
  if paren then locale = substr(City,1,paren-1);
  else if bracket then locale = substr(City,1,bracket-1);
  else locale = trim(left(City));

  City1 = scan(locale,1,',');
  State = scan(locale,2,',');

run;
```

# SAS and Open Source

## Use Case: SAS Reports Output

```
Proc SQL;

Select SB as Superbowl "SuperBowl Number"
      ,Season "Season"
      ,City1 as City "SB City"
      ,State "SB State"
      ,fans as Attendance "Attendance" format = comma9.
      ,Winner "Winning Team"
      ,Loser "Losing Team"
      ,points as Score "Score"
      ,WS "Winning Score"
      ,LS "Losing Score"
      ,OT "Overtime"

from SB2;

QUIT;
```

# SAS and Open Source

## Use Case: SAS Reports Output

| SuperBowl Number | Season | SB City     | SB State   | Attendance | Winning Team        | Losing Team         | Score | Winning Score | Losing Score | Overtime |
|------------------|--------|-------------|------------|------------|---------------------|---------------------|-------|---------------|--------------|----------|
| 1                | 1966   | Los Angeles | California | 61,946     | Green Bay Packers   | Kansas City Chiefs  | 35-10 | 35            | 10           | 0        |
| 2                | 1967   | Miami       | Florida    | 75,546     | Green Bay Packers   | Oakland Raiders     | 33-14 | 33            | 14           | 0        |
| 3                | 1968   | Miami       | Florida    | 75,389     | New York Jets       | Baltimore Colts     | 16-7  | 16            | 7            | 0        |
| 4                | 1969   | New Orleans | Louisiana  | 80,562     | Kansas City Chiefs  | Minnesota Vikings   | 23-7  | 23            | 7            | 0        |
| 5                | 1970   | Miami       | Florida    | 79,204     | Baltimore Colts     | Dallas Cowboys      | 16-13 | 16            | 13           | 0        |
| 6                | 1971   | New Orleans | Louisiana  | 81,023     | Dallas Cowboys      | Miami Dolphins      | 24-3  | 24            | 3            | 0        |
| 7                | 1972   | Los Angeles | California | 90,182     | Miami Dolphins      | Washington Redskins | 14-7  | 14            | 7            | 0        |
| 8                | 1973   | Houston     | Texas      | 71,882     | Miami Dolphins      | Minnesota Vikings   | 24-7  | 24            | 7            | 0        |
| 9                | 1974   | New Orleans | Louisiana  | 80,997     | Pittsburgh Steelers | Minnesota Vikings   | 16-6  | 16            | 6            | 0        |
| 10               | 1975   | Miami       | Florida    | 80,187     | Pittsburgh Steelers | Dallas Cowboys      | 21-17 | 21            | 17           | 0        |
| 11               | 1976   | Pasadena    | California | 103,438    | Oakland Raiders     | Minnesota Vikings   | 32-14 | 32            | 14           | 0        |
| 12               | 1977   | New Orleans | Louisiana  | 76,400     | Dallas Cowboys      | Denver Broncos      | 27-10 | 27            | 10           | 0        |

# SAS and Python

Using SAS in a Python Session

# SASPy

<https://github.com/sassoftware/saspy>



## Overview

This module provides interfaces between Python and SAS. This module enables a Python developer to create mixed Python/SAS workflows to leverage the powers of both SAS and Python, by connecting a Python process to any of a variety of SAS deployments, where it will run SAS code. The SAS code is generated by the SASPy object and methods or explicitly user written. Results from SAS are returned as text, HTML5 documents (via SAS ODS), or as Pandas Data Frames. This module supports running analytics and returning the resulting graphics and result data to the Python process. It can convert data representations between SAS Data Sets and Pandas Data Frames.

# Thanks



**Jim Box**  
Principal Data Scientist, Life Sciences at SAS  
Durham, North Carolina, United States · [Contact info](#)



**Samiul Haque** · 1st  
Transforming data into intelligence through cross-disciplinary collaboration | Machine Learning | Data Science | Analytics  
Cary, North Carolina, United States · [Contact info](#)

sas.com

