SAS and Open Source Playing Nicely Together

Jim Box

Principal Data Scientist, HLS R&D

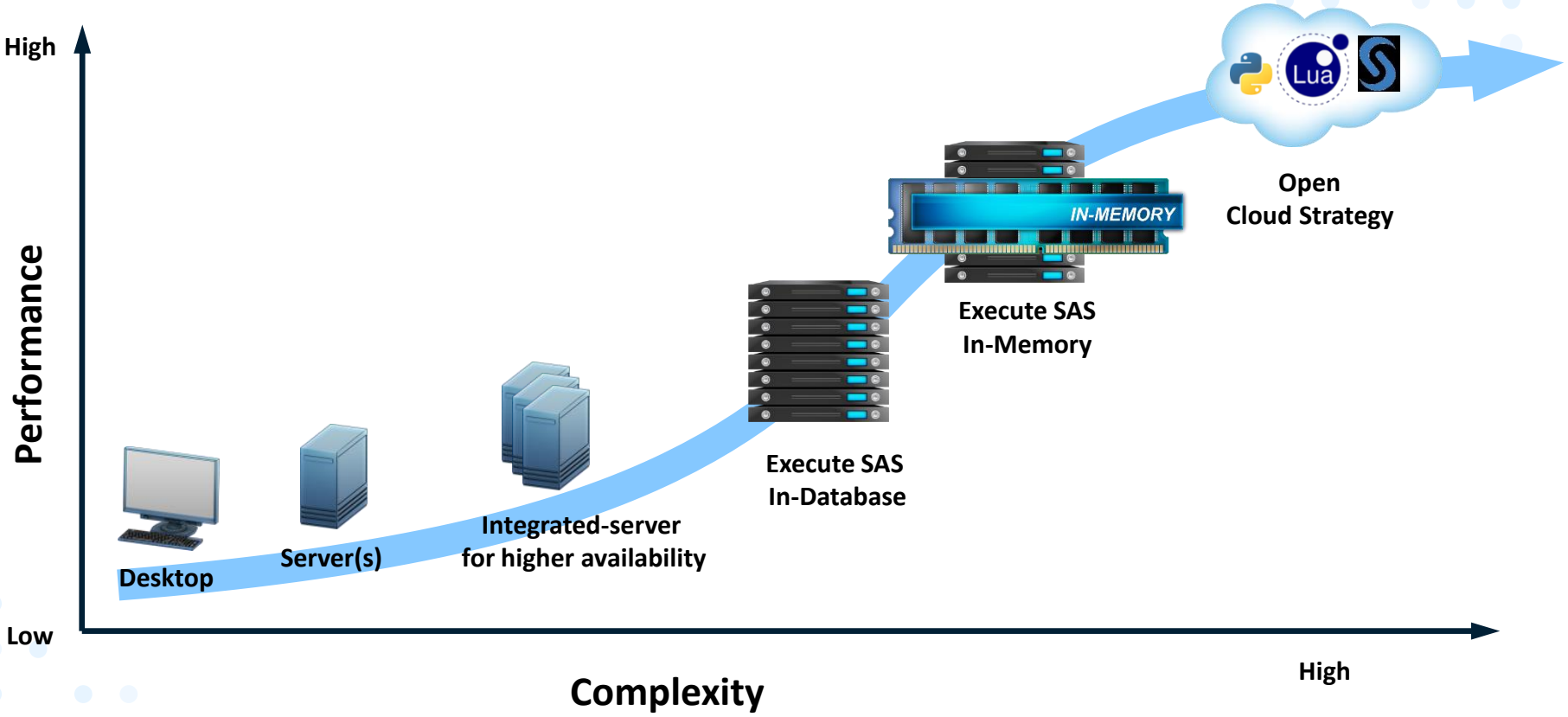
SAS Institute

Copyright © SAS Institute Inc. All rights reserved.

SAS Environments

9.4 vs Viya

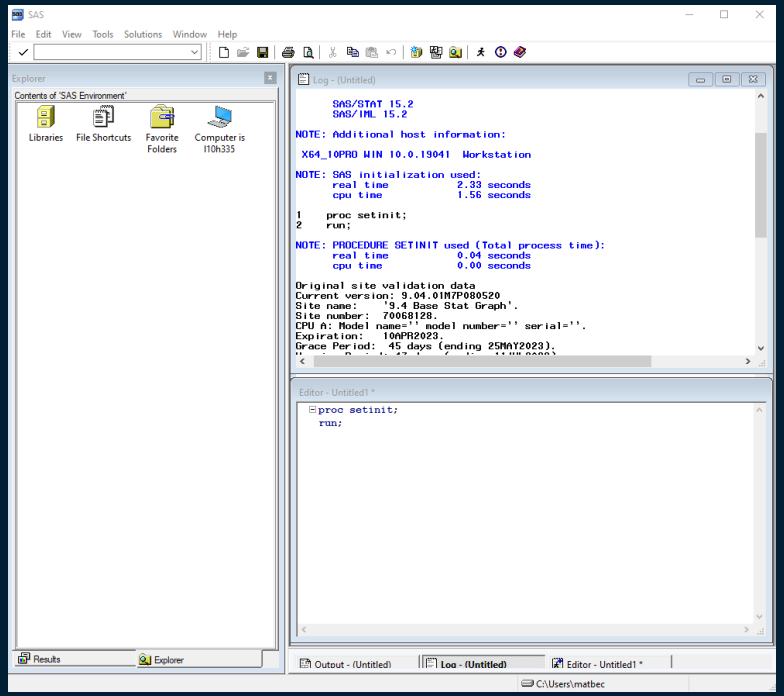
Copyright © SAS Institute Inc. All rights reserved.

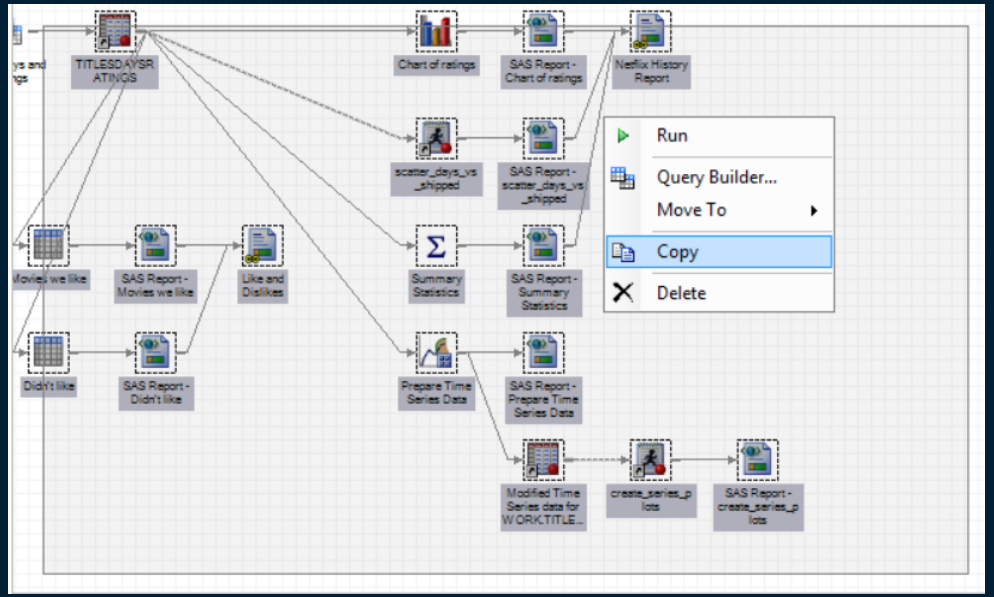


Copyright © SAS Institute Inc. All rights reserved.

SAS Platforms

SAS 9.4 – Where you are now

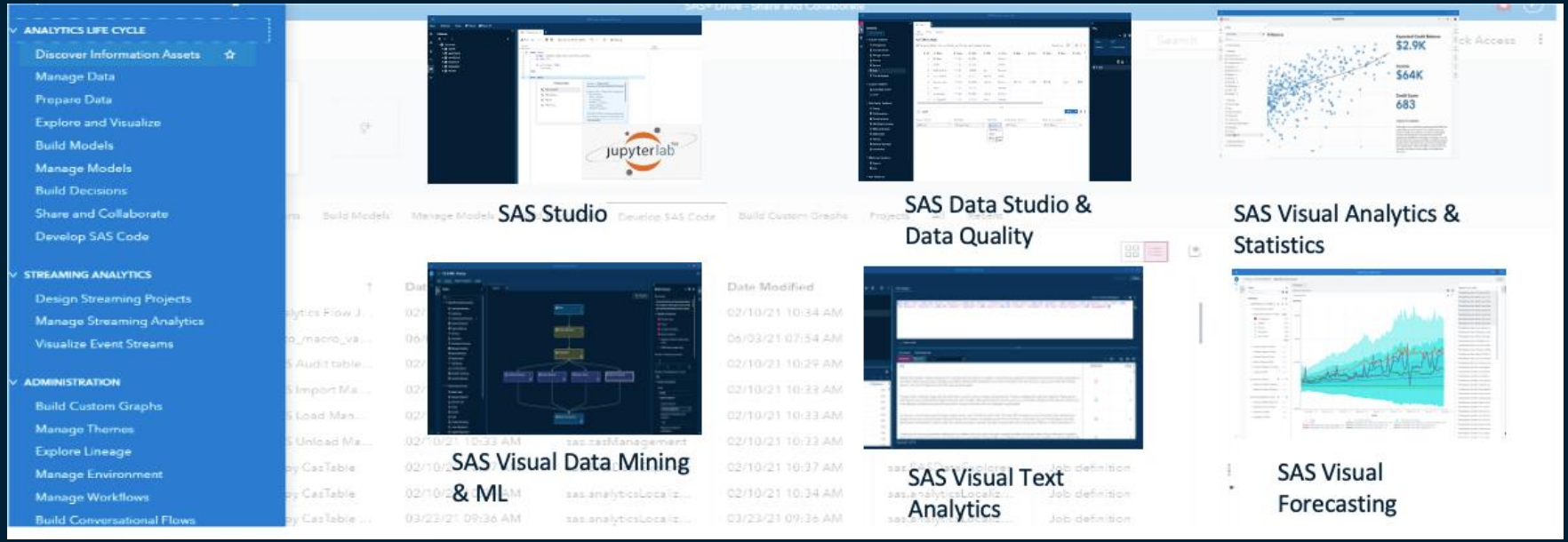




Copyright © SAS Institute Inc. All rights reserved.

SAS Platforms

SAS Viya – Where we are now



Copyright © SAS Institute Inc. All rights reserved.

SAS and R

A long-term relationship

Copyright © SAS Institute Inc. All rights reserved.

SAS 9.4 and R

PROC IML

- PROC IML (Interactive Matrix Language)
- Code structure:
  - call ExportDataSetToR(<"*SAS libname.dataset*">,"<*R data frame*>")
  - Submit / R;
  - *All your R code*
  - Endsubmit;
  - call ImportDataSetFromR("<*SAS dataset to write*>","<*R data frame*>")
  - Quit;

Copyright © SAS Institute Inc. All rights reserved.

SAS 9.4 and R

PROC IML

```
1  PROC IML;
2  call ExportDataSetToR("Sashelp.Heart", "df" );
3
4  submit / R;
5     summary(df)
6  endsubmit;
7
8  QUIT;
9
```

Copyright © SAS Institute Inc. All rights reserved.

SAS 9.4 and R

PROC IML

Code Log Results

```
  Status                          DeathCause     AgeCHDdiag       Sex
Alive:3218   Cancer                    : 539   Min.   :32.0   Female:2873
Dead :1991   Cerebral Vascular Disease: 378   1st Qu.:57.0   Male  :2336
             Coronary Heart Disease    : 605   Median :63.0
             Other                     : 357   Mean   :63.3
             Unknown                   : 112   3rd Qu.:70.0
             NA's                      :3218   Max.   :90.0
                                               NA's   :3760
  AgeAtStart        Height          Weight        Diastolic
Min.   :28.00   Min.   :51.50   Min.   : 67.0   Min.   : 50.00
1st Qu.:37.00   1st Qu.:62.25   1st Qu.:132.0   1st Qu.: 76.00
Median :43.00   Median :64.50   Median :150.0   Median : 84.00
Mean   :44.07   Mean   :64.81   Mean   :153.1   Mean   : 85.36
3rd Qu.:51.00   3rd Qu.:67.50   3rd Qu.:172.0   3rd Qu.: 92.00
Max.   :62.00   Max.   :76.50   Max.   :300.0   Max.   :160.00
                NA's   :6       NA's   :6
   Systolic          MRW          Smoking         AgeAtDeath     Cholesterol
Min.   : 82.0   Min.   : 67   Min.   : 0.000   Min.   :36.00   Min.   : 96.0
1st Qu.:120.0   1st Qu.:106   1st Qu.: 0.000   1st Qu.:63.00   1st Qu.:196.0
Median :132.0   Median :118   Median : 1.000   Median :71.00   Median :223.0
Mean   :136.9   Mean   :120   Mean   : 9.367   Mean   :70.54   Mean   :227.4
3rd Qu.:148.0   3rd Qu.:131   3rd Qu.:20.000   3rd Qu.:79.00   3rd Qu.:255.0
Max.   :300.0   Max.   :268   Max.   :60.000   Max.   :93.00   Max.   :568.0
                NA's   :6     NA's   :36       NA's   :3218    NA's   :152
    Chol_Status     BP_Status        Weight_Status            Smoking_Status
Borderline:1861   High   :2267   Normal     :1472   Heavy (16-25)     :1046
Desirable :1405   Normal :2143   Overweight :3550   Light (1-5)       : 579
High      :1791   Optimal: 799   Underweight: 181   Moderate (6-15)   : 576
NA's      : 152                  NA's       :   6   Non-smoker         :2501
                                                    Very Heavy (> 25): 471
                                                    NA's              :  36
```

SAS 9.4 and R

PROC IML

- Text output will be written to the Results window
- Graphical output needs to be pointed to a specific location – it will not (currently) show up in the SAS interface

```
119  p <- ggplot(teams, aes(x=Outcome, y=Score, color=Outcome)) + geom_boxplot()
120
121  png(file="/data/compute-landingzone/Projects/Open Source/boxplot.png",
122  width=600, height=350)
123  p
124  dev.off()
```

Copyright © SAS Institute Inc. All rights reserved.

SAS 9.4 and R

Reading SAS Data into R

- R library: sas7bdat
- Command: df=read.sas7bdat(“heart.sas7bdat”)

- R library: haven:
- Command: df= read_sas(“heart.sas7bdat”)

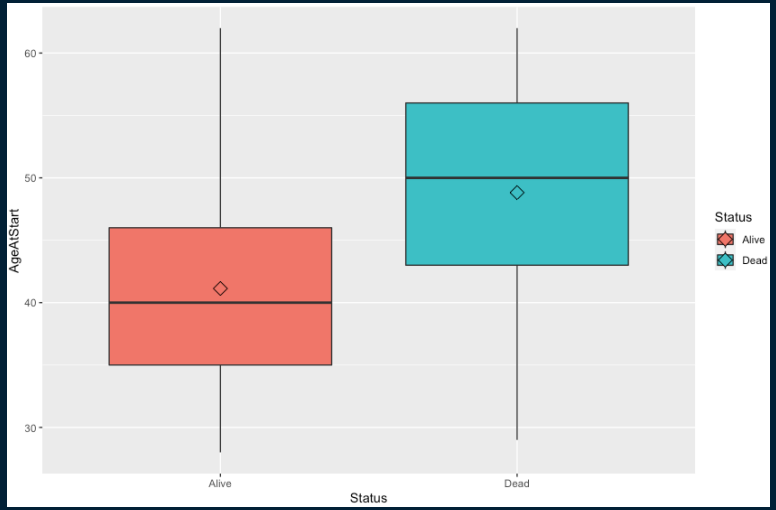
- Haven also has a write_sas command to output datasets when done

Copyright © SAS Institute Inc. All rights reserved.

SAS 9.4 and R

Reading SAS Data into R

```
1  library(ggplot2)
2  library(sas7bdat)
3
4  df=read.sas7bdat("heart.sas7bdat")
5
6  plt =  ggplot(df, aes(x=Status, y=AgeAtStart,fill=Status)) +
7    geom_boxplot(outlier.colour="red", outlier.shape=8,
8                 outlier.size=4)
9
10 plt + stat_summary(fun=mean, geom='point',shape=23,size=4)
11
```



Copyright © SAS Institute Inc. All rights reserved.

SAS Viya and R

PROC IML

- Works exactly the same as SAS 9.4. (runs in the compute engine)

Copyright © SAS Institute Inc. All rights reserved.

SAS Viya and R

R Studio connection & SWAT

```
1  library(swat)
2
3  Sys.setenv(CAS_CLIENT_SSL_CA_LIST='/etc/pki/tls/certs/ca.crt')
4
5  username <- rstudioapi::askForPassword("username")
6  password <- rstudioapi::askForPassword("password")
7
8  session <- swat::CAS('https://viya4.globalhls.sashq-d.openstack.sas.com/cas-shared-default-http', 443, protocol='https',
9
10 currentCASlib<-'PUBLIC'
11
12 #list available tables in caslib
13 sites <- defCasTable(session,caslib=currentCASlib,"StudySites")
14 head(sites)
15 dim(sites)
16
```

Copyright © SAS Institute Inc. All rights reserved.

SAS Viya and R

R Studio connection & SWAT

```
> head(sites)
  Replication                          Site Start_Flag StartUp Cost StartDelay CountryDelay FirstPatient LastPatient
1           1      Chenango Memorial Hospital          1   22886 5500         60           32        22891        23298
2           1           Derry Medical Center          1   22890 4500         64           36        22894        23300
3           1         Lassen General Hospital         1   22891 6000         65           37        22895        23295
4           1           Sacred Heart Hospital         1   22894 5000         68           40        22898        23298
5           1 Shasta Regional Medical Center          1   22879 5500         53           25        22883        23297
6           1              St Eligus Hospital         1   22897 7500         71           43        22902        23295
  N_Screened N_Enroll     P_Fail N_Complete    P_Comp StudyVisits ScreenCost VisitCost TotalCost
1         93       84 0.09677419         68 0.8095238         236      62775    206516    269291
2        110       97 0.11818182         86 0.8865979         277      82500    275200    357700
3         93       80 0.13978495         64 0.8000000         223      60450    187200    247650
4        118      108 0.08474576         90 0.8333333         309      85550    293580    379130
5        100       81 0.19000000         64 0.7901235         230      70000    201600    271600
6         81       69 0.14814815         51 0.7391304         194      53460    151470    204930
```

Copyright © SAS Institute Inc. All rights reserved.

SAS Viya and R

R Studio connection & SWAT

```
site51 <- cas.dataStep.runCode(session,
                                code="
        data Site51;
        set PUBLIC.StudySites;
        if Replication = ' 51';
run;"
)
results<-cas.table.fetch(session,
                         table=list(name="site51")
)
results
```

Copyright © SAS Institute Inc. All rights reserved.

SAS Viya and R

R Studio connection & SWAT

	Index	Replication	Site	Start_Flag	StartUp	Cost	StartDelay	CountryDelay	FirstPatient	LastPatient
1	1	51	Chenango Memorial Hospital	1	22903	5500	77	37	22908	23320
2	2	51	Derry Medical Center	1	22895	4500	69	29	22899	23322
3	3	51	Lassen General Hospital	1	22902	6000	76	36	22906	23318
4	4	51	Sacred Heart Hospital	1	22895	5000	69	29	22899	23322
5	5	51	Shasta Regional Medical Center	1	22899	5500	73	33	22903	23322
6	6	51	St Eligus Hospital	1	22907	7500	81	41	22912	23323
7	7	51	Tower Medical Group	1	22898	6550	72	32	22904	23319
8	8	51	Twin Pines Medical Center	1	22899	7000	73	33	22905	23320
9	9	51	Western Regional Hospital	1	22897	8000	71	31	22901	23325
10	10	51	Wexler Medical Center	1	22897	4250	71	31	22901	23325

Copyright © SAS Institute Inc. All rights reserved.

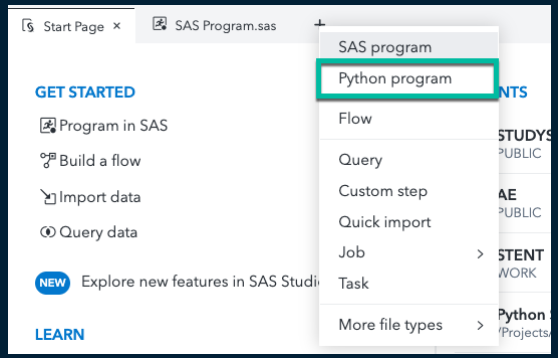
SAS and Python

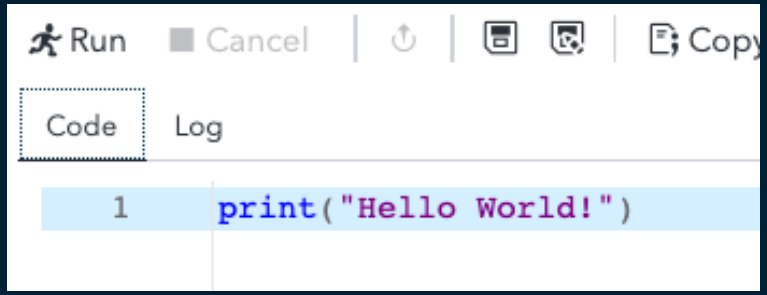
A big focus for SAS

Copyright © SAS Institute Inc. All rights reserved.

SAS Viya and Python

Write Python Programs in SAS Studio






```
80   proc python;
81   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.
>>>
>>>
81 !       ;
82
83   print("Hello World!")
84
85   endsubmit;
86   quit;
87   %let SAS_workpath = %sysfunc(pathname(work));
>>>
Hello World!
>>>
>>>
```

Copyright © SAS Institute Inc. All rights reserved.

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

Copyright © SAS Institute Inc. All rights reserved.

SAS Viya and Python

PROC Python – Run SAS in Python

```
%let language = 'python';

proc python;
submit;
lang = SAS.symget('language')
ver = 3.8
SAS.submit("data work.test; language={}; version={}; run;".format(lang,ver))

var3 = SAS.sasfnc("upcase","hello world")
print( var3)

py_var = 'Inside python'
SAS.symput('macrovar', py_var)


endsubmit;
run;

%put &=macrovar;


proc print data=test;
run;
```

Copyright © SAS Institute Inc. All rights reserved.

SAS Viya and Python

PROC Python – Run SAS in Python

```
95    endsubmit;
96    run;
97    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            0.01 seconds

>>>
HELLO WORLD
>>>
>>>
NOTE: PROCEDURE PYTHON used (Total process time):
      real time           0.00 seconds
      cpu time            0.01 seconds

98
99    %put &=macrovar;
MACROVAR=Inside python
100
101
```

Copyright © SAS Institute Inc. All rights reserved.

SAS Viya and Python

PROC Python – Run SAS in Python

Obs	language	version
1	python	3.8

Copyright © SAS Institute Inc. All rights reserved.

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

Copyright © SAS Institute Inc. All rights reserved.

SAS and Open Source

Use Cases

- Sometimes a Python method or an R library can make your work easier
- PROC PYTHON and PROC IML make it easy to leverage Python and R, but still be able to use SAS when it makes sense
- SAS Viya makes it easy to mix and match programming languages to solve problems in the most efficient way possible

Copyright © SAS Institute Inc. All rights reserved.

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

https://en.wikipedia.org/wiki/List_of_Super_Bowl_champions

Copyright © SAS Institute Inc. All rights reserved.

SAS and Open Source

Use Case: Python grabs data

```
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;
```

Copyright © SAS Institute Inc. All rights reserved.

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;
```

Copyright © SAS Institute Inc. All rights reserved.

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;
```

Copyright © SAS Institute Inc. All rights reserved.

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

Copyright © SAS Institute Inc. All rights reserved.

Conclusion

SAS & Open Source

- SAS has been open source friendly for years with R in PROC IML
- Python integration is a key goal
- SAS Viya makes it easy to mix and match programming languages to solve problems in the most efficient way possible
- There are several other integration points that are more for modelers and data scientists
  - Saspy in Python
  - Open Source nodes in Viya Modelling pipelines

Copyright © SAS Institute Inc. All rights reserved.

Thanks

sas.com

Copyright © SAS Institute Inc. All rights reserved.