The George Washington University
Department of Statistics

STAT 4197/6197 – Fall 2019

Week 15 - December 6, 2019

Major Topic: SASPy and Selected Python Applications

Detailed Topics:

1) SASpy Applications
2) Table Lookup Techniques
3) Best Practices in SAS Programming
4) Errors and Warnings in SAS Log
5) Efficiency in SAS Programming
6) Review for Final Exam

Acknowledgements: Texts are primarily adapted/obtained from SAS documentation and web sources.
Portions of SAS' copyrighted SAS course content are reproduced here with permission of SAS Institute Inc., Cary, NC, USA. SAS Institute Inc. makes no warranties with respect to these materials and disclaims all liability therefor. Do not copy or circulate the handouts.

SASPy

Utilization of Python in clinical study by SASPy (PhUSE Connect 2018)

A Basic Introduction to SASPy and Jupyter Notebooks (SAS Global Forum, 2018)

Introducing SASPy: Use Python code to access SAS

Coding in Python with SAS University Edition (SAS Blogs, February 5, 2018)

How to code in Python with SAS 9.4

Open Your Mind: Use Cases for SAS® and Open-Source Analytics (SAS Global Forum, 2017)

SAS® and Python: The Perfect Partners in Crime

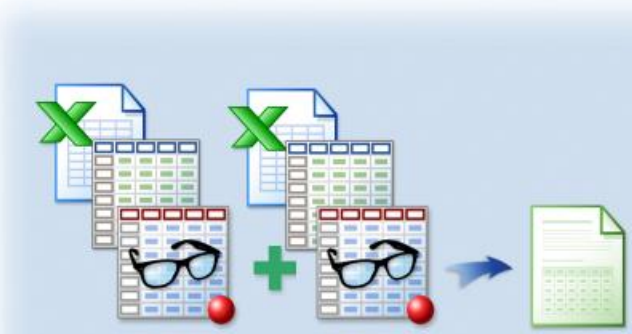
SAS vs. Python - How Do They Compare (SESUG, 2017)

Revolutionizing Statistical Computing in SAS® with the Jupyter Notebook

Acknowledgements: Texts are primarily adapted/obtained from SAS documentation and web sources.
Portions of SAS' copyrighted SAS course content are reproduced here with permission of SAS Institute Inc., Cary, NC, USA. SAS Institute Inc. makes no warranties with respect to these materials and disclaims all liability therefor. Do not copy or circulate the handouts.

Introduction

There is often a need to combine or look up data from multiple sources to create meaningful reports.



4

Introduction

When data sources do not share a common structure, a lookup table can match them.

1

Acknowledgements: Texts are primarily adapted/obtained from SAS documentation and web sources.
Portions of SAS' copyrighted SAS course content are reproduced here with permission of SAS Institute Inc., Cary, NC, USA. SAS Institute Inc. makes no warranties with respect to these materials and disclaims all liability therefor. Do not copy or circulate the handouts.

Lookup Table Data

You can use several techniques to look up data.

Continent ID	Continent Name
91	North America
93	Europe
94	Africa
95	Asia
96	Australia/Pacific

6

Acknowledgements: Texts are primarily adapted/obtained from SAS documentation and web sources.
Portions of SAS' copyrighted SAS course content are reproduced here with permission of SAS Institute Inc., Cary, NC, USA. SAS Institute Inc. makes no warranties with respect to these materials and disclaims all liability therefor. Do not copy or circulate the handouts.

IF-THEN/ELSE Statements

Lookup tables can be SAS programming statements.

```
data countryinfo;
   set orion.country;
   if ContinentID=91
      then Continent='North America';
   else if ContinentID=93
      then Continent='Europe';
   else if ContinentID=94
      then Continent='Africa';
   else if ContinentID=95
      then Continent='Asia';
   else if ContinentID=96
      then Continent='Australia/Pacific';
run;
```

7

DATA Step Merge

Lookup tables can be SAS data sets that are accessed during a DATA step merge.

```
data countryinfo;
   merge orion.country
         orion.Continent;
   by ContinentID;
run;
```

8

Acknowledgements: Texts are primarily adapted/obtained from SAS documentation and web sources.
Portions of SAS' copyrighted SAS course content are reproduced here with permission of SAS Institute Inc., Cary, NC, USA. SAS Institute Inc. makes no warranties with respect to these materials and disclaims all liability therefor. Do not copy or circulate the handouts.

PROC SQL Join

Lookup tables can be SAS data sets that are accessed during a PROC SQL join.

```
proc sql;
   create table countryinfo as
      select * from
        orion.country a, orion.Continent b
        where a.ContinentID=b.ContinentID;
quit;
```

9

User-Defined Formats

Lookup tables can be user-defined formats that are accessed with a FORMAT statement or PUT function.

```
proc format;
   value ContName
      91='North America'   93='Europe'
      94='Africa'          95='Asia'
      96='Australia/Pacific';
run;

proc print data=orion.country;
   format ContinentID ContName.;
run;

data countryinfo;
   set orion.country;
   Continent=put(ContinentID,ContName.);
run;
```

10

Acknowledgements: Texts are primarily adapted/obtained from SAS documentation and web sources.
Portions of SAS' copyrighted SAS course content are reproduced here with permission of SAS Institute Inc., Cary, NC, USA. SAS Institute Inc. makes no warranties with respect to these materials and disclaims all liability therefor. Do not copy or circulate the handouts.

Arrays

Lookup tables can be arrays.

```
data countryinfo;
   array ContName{91:96} $ 30 _temporary_
      ('North America',
         ' ',
      'Europe',
      'Africa',
      'Asia',
      'Australia/Pacific');
   set orion.country;
   Continent=ContName{ContinentID};
run;
```

11

Table Lookup Techniques

The technique that is used to perform a table lookup depends on the data.

Table Location	Technique	Lookup Table
DATA step	IF-THEN/ELSE statements	SAS programming statements
Disk	SQL join merge SET/SET KEY=	SAS data set
Memory	FORMAT statement PUT statement PUT function	user-defined format
	array reference	array
	FIND method	hash object

12

Acknowledgements: Texts are primarily adapted/obtained from SAS documentation and web sources.
Portions of SAS' copyrighted SAS course content are reproduced here with permission of SAS Institute Inc., Cary, NC, USA. SAS Institute Inc. makes no warranties with respect to these materials and disclaims all liability therefor. Do not copy or circulate the handouts.

Reviewing SAS Arrays

An array is declared with an ARRAY statement. Array elements can be SAS variables or temporary data elements.

```
array ContName{91:96} $ 30 _temporary_
          ('North America',
           ' ',
           'Europe',
           'Africa',
           'Asia',
           'Australia/Pacific');
```

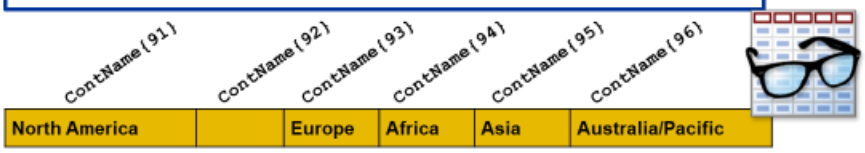
ARRAY *array-name* {*number-of-elements*} <$> <*length*> <*_temporary_*> <*list-of-variables*> <(*initial-values*)>;

24

Reviewing SAS Arrays

This ARRAY statement assigns an initial value to each corresponding element. SAS matches the elements and values by position, so the values must be listed in the order of the array elements.

```
array ContName{91:96} $ 30 _temporary_
          ('North America',
           ' ',
           'Europe',
           'Africa',
           'Asia',
           'Australia/Pacific');
```



25

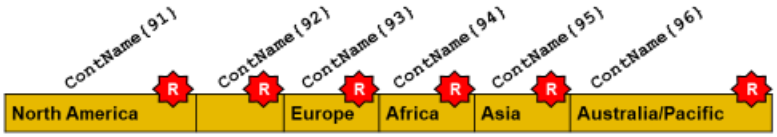
...

Acknowledgements: Texts are primarily adapted/obtained from SAS documentation and web sources.
Portions of SAS' copyrighted SAS course content are reproduced here with permission of SAS Institute Inc., Cary, NC, USA. SAS Institute Inc. makes no warranties with respect to these materials and disclaims all liability therefor. Do not copy or circulate the handouts.

Reviewing SAS Arrays

When an initial value list is specified, all array elements behave as if they were named in a RETAIN statement.

```
array ContName{91:96} $ 30 _temporary_
          ('North America',
           ' ',
           'Europe',
           'Africa',
           'Asia',
           'Australia/Pacific');
```



26

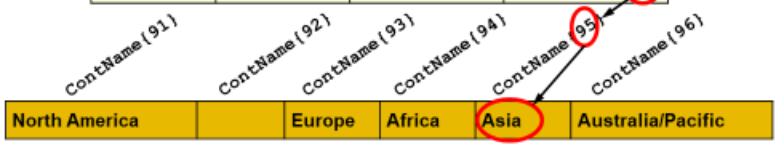
Acknowledgements: Texts are primarily adapted/obtained from SAS documentation and web sources.
Portions of SAS' copyrighted SAS course content are reproduced here with permission of SAS Institute Inc., Cary, NC, USA. SAS Institute Inc. makes no warranties with respect to these materials and disclaims all liability therefor. Do not copy or circulate the handouts.

Example: Looking Up Continent Names Using an Array

A one-dimensional array can serve as a lookup table with the names of the continents.

Partial **orion.country**

Country	Country Name	Population	ContinentID
AU	Australia	20,000,000	96
CA	Canada	.	91
DE	Germany	80,000,000	93
IL	Israel	5,000,000	95
TR	Turkey	70,000,000	95



29

Example: Looking Up Continent Names Using an Array

```
data countryinfo;
   array ContName{91:96} $ 30 _temporary_
         ('North America',
          ' ',
          'Europe',
          'Africa',
          'Asia',
          'Australia/Pacific');
   set orion.country;
   Continent=ContName{ContinentID};
run;
```

30

Acknowledgements: Texts are primarily adapted/obtained from SAS documentation and web sources.
Portions of SAS' copyrighted SAS course content are reproduced here with permission of SAS Institute Inc., Cary, NC, USA. SAS Institute Inc. makes no warranties with respect to these materials and disclaims all liability therefor. Do not copy or circulate the handouts.

Using a Table Lookup

Combine the data with a ***table lookup***.

Partial **orion.salarystats**

Statistic	Yr1978	Yr1979	Yr1980
MedianSalary	30025	29442.5	30020
StdSalary	28551.9	9918.35	22356.91
SumSalary	2393860	132150	235030
AvgSalary	39243.61	33037.5	39171.67

Partial **orion.employeepayroll**

YEAR(*SAS-date-value*)

EmployeeID	Salary	BirthDate	EmployeeHireDate
120101	163040	18AUG1980	01JUL2007
120102	108255	11AUG1973	01JUN1993
120103	87975	22JAN1953	01JAN1978
120104	46230	11MAY1958	01JAN1985
120105	27110	21DEC1978	01MAY2003

36

Acknowledgements: Texts are primarily adapted/obtained from SAS documentation and web sources.
Portions of SAS' copyrighted SAS course content are reproduced here with permission of SAS Institute Inc., Cary, NC, USA. SAS Institute Inc. makes no warranties with respect to these materials and disclaims all liability therefor. Do not copy or circulate the handouts.

Using a One-Dimensional Array

```
data compare;
   keep EmployeeID YearHired Salary
        Average SalaryDif;
   format Salary Average SalaryDif dollar12.2;
   array yr{1978:2011} Yr1978-Yr2011;
   if _n_=1 then set orion.salarystats
     (where=(Statistic='AvgSalary'));
   set orion.employeepayroll
       (keep=EmployeeID EmployeeHireDate Salary);
   YearHired=year(EmployeeHireDate);
   Average=yr{YearHired};
   SalaryDif=Salary-Average;
run;
```

37

p305d01

...

Using a One-Dimensional Array

```
data compare;
   keep EmployeeID YearHired Salary
        Average SalaryDif;
   format Salary Average SalaryDif dollar12.2;
   array yr{1978:2011} Yr1978-Yr2011;
   if _n_=1 then set orion.salarystats
     (where=(Statistic='AvgSalary'));
   set orion.employeepayroll
       (keep=EmployeeID EmployeeHireDate Salary);
   YearHired=year(EmployeeHireDate);
   Average=yr{YearHired};
   SalaryDif=Salary-Average;
run;
```

38

p305d01

Acknowledgements: Texts are primarily adapted/obtained from SAS documentation and web sources.
Portions of SAS' copyrighted SAS course content are reproduced here with permission of SAS Institute Inc., Cary, NC, USA. SAS Institute Inc. makes no warranties with respect to these materials and disclaims all liability therefor. Do not copy or circulate the handouts.

Acknowledgements: Texts are primarily adapted/obtained from SAS documentation and web sources.
Portions of SAS' copyrighted SAS course content are reproduced here with permission of SAS Institute Inc., Cary, NC, USA. SAS Institute Inc. makes no warranties with respect to these materials and disclaims all liability therefor. Do not copy or circulate the handouts.