Introduction to SAS Programming

Daniel T Yehdego

The University of Texas at El Paso
Computational Science Program
Oct 29, 2013
What is SAS?

✓ Statistical software package
✓ Developed at North Carolina State University, 1966
  SAS Institute – 1976
✓ Point-and-click interface – version 9 in 2004
✓ Advanced analytics, BI, data management, predictive analytics.

❖ Insurance, public health, scientific research, finance, human resources, IT, utilities, retail, forecasting and decision-making.

❖ Standard to submit clinical pharmaceutical trials to FDA

http://en.wikipedia.org/wiki/SAS_(software)
Introduction

- Base SAS
- SAS/STAT
- SAS/GRAPH
- SAS/IML
- SAS/OR …
SAS Windowing Environment
Getting Help in SAS

![SAS Help Menu](image)

- **SAS System Help**
- **Using This Window**
- **Books and Training**
- **Getting Started with SAS Software**
- **SAS on the Web**
- **About SAS System**

**NOTE:**
SAS initialization used:
real time 11.70 seconds
data work.NewSalesEmps;
    length First_Name $ 12 Last_Name $ 18
        Job_Title $ 25;
    infile 'newemps.csv' dlm=',';
    input First_Name $ Last_Name $
        Job_Title $ Salary;
run;

proc print data=work.NewSalesEmps;
data work.NewSalesEmps;
  length First_Name $12 Last_Name $18
  Job_Title $25;
  infile 'newemps.csv' dlm=',';
  input First_Name $ Last_Name $
  Job_Title $ Salary;
run;

NOTE: The infile 'newemps.csv' is:
  File Name=S:\Workshop\newemps.csv,
  RECFM=V,LRECL=256

NOTE: 71 records were read from the infile 'newemps.csv'.
  The minimum record length was 28.
  The maximum record length was 47.
NOTE: The data set WORK.NEWSALESEMPS has 71 observations and 4 variables.

proc print data=work.NewSalesEmps;
run;

NOTE: There were 71 observations read from the data set WORK.NEWSALESEMPS.
## Partial Proc Print Output

<table>
<thead>
<tr>
<th>Obs</th>
<th>First_Name</th>
<th>Last_Name</th>
<th>Job_Title</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Satyakam</td>
<td>Denny</td>
<td>Sales Rep. II</td>
<td>26780</td>
</tr>
<tr>
<td>2</td>
<td>Monica</td>
<td>Kletschkus</td>
<td>Sales Rep. IV</td>
<td>30890</td>
</tr>
<tr>
<td>3</td>
<td>Kevin</td>
<td>Lyon</td>
<td>Sales Rep. I</td>
<td>26955</td>
</tr>
<tr>
<td>4</td>
<td>Petrea</td>
<td>Soltau</td>
<td>Sales Rep. II</td>
<td>27440</td>
</tr>
<tr>
<td>5</td>
<td>Marina</td>
<td>Iyengar</td>
<td>Sales Rep. III</td>
<td>29715</td>
</tr>
<tr>
<td>6</td>
<td>Shani</td>
<td>Duckett</td>
<td>Sales Rep. I</td>
<td>25795</td>
</tr>
<tr>
<td>7</td>
<td>Fang</td>
<td>Wilson</td>
<td>Sales Rep. II</td>
<td>26810</td>
</tr>
<tr>
<td>8</td>
<td>Michael</td>
<td>Minas</td>
<td>Sales Rep. I</td>
<td>26970</td>
</tr>
<tr>
<td>9</td>
<td>Amanda</td>
<td>Liebman</td>
<td>Sales Rep. II</td>
<td>27465</td>
</tr>
<tr>
<td>10</td>
<td>Vincent</td>
<td>Eastley</td>
<td>Sales Rep. III</td>
<td>29695</td>
</tr>
<tr>
<td>11</td>
<td>Viney</td>
<td>Barbis</td>
<td>Sales Rep. III</td>
<td>30265</td>
</tr>
<tr>
<td>12</td>
<td>Skev</td>
<td>Rusli</td>
<td>Sales Rep. II</td>
<td>26580</td>
</tr>
<tr>
<td>13</td>
<td>Narelle</td>
<td>James</td>
<td>Sales Rep. III</td>
<td>29990</td>
</tr>
<tr>
<td>14</td>
<td>Gerry</td>
<td>Snellings</td>
<td>Sales Rep. I</td>
<td>26445</td>
</tr>
<tr>
<td>15</td>
<td>Leonid</td>
<td>Karavdic</td>
<td>Sales Rep. II</td>
<td>27860</td>
</tr>
</tbody>
</table>
Components of a SAS program

- A SAS program is a sequence of steps that the user submits for execution

DATA steps: create SAS data sets

PROC steps: process SAS data sets

Raw Data

SAS Data Set

PROC Step

Report
A SAS program is a sequence of steps

```sas
data work.NewSalesEmps;
  length First_Name $ 12
       Last_Name $ 18 Job_Title $ 25;
  infile 'newemps.csv' dlm=',';
  input First_Name $ Last_Name $ 
    Job_Title $ Salary;
run;

proc print data=work.NewSalesEmps;
run;

proc means data=work.NewSalesEmps;
  class Job_Title;
  var Salary;
run;
```
SAS program

SAS statements characteristics

```sas
data work.NewSalesEmps;
  length First_Name $ 12
         Last_Name  $ 18 Job_Title  $ 25;
  infile 'newemps.csv' dlm=',';
  input First_Name $ Last_Name $
         Job_Title $ Salary;
run;

proc print data=work.NewSalesEmps;
r
run;

proc means data=work.NewSalesEmps;
  class Job_Title;
  var Salary;
run;
```
SAS Syntax rules

- SAS statements are free-format.
- One or more blanks or special characters can be used to separate words.
- They can begin and end in any column.
- A single statement can span multiple lines.
- Several statements can be on the same line.

```sas
data work.NewSalesEmps;
  length First_Name $ 12
  Last_Name $ 18 Job_Title $ 25;
  infile 'newemps.csv' dlm=',,';
  input First_Name $ Last_Name $
    Job_Title $ Salary;
run;
proc print data=work.NewSalesEmps; run;
  proc means data =work.NewSalesEmps;
  class Job_Title; var Salary;run;
```
* This program creates and uses the data set called work.NewSalesEmps. *

```sas
data work.NewSalesEmps;
    length First_Name $ 12 Last_Name $ 18
         Job_Title $ 25;
    infile 'newemps.csv' dlm=',';
    input First_Name $ Last_Name $
         Job_Title $ Salary /*numeric*/;
run;

/*
proc print data=work.NewSalesEmps;
run;
*/

proc means data=work.NewSalesEmps;
    *class Job_Title;
    var Salary;
run;
```
Syntax Errors

Examples of syntax errors:

✓ Misspelled keywords
✓ Missing semicolons
✓ Unmatched quotation marks
✓ Invalid options

```plaintext
daat work.NewSalesEmps;
  length First_Name $12
    Last_Name $18 Job_Title $25;
  infile 'newemps.csv' dlm=',';
  input First_Name $ Last_Name $
    Job_Title $ Salary;
run;

proc print data=work.NewSalesEmps
run;

proc means data=work.NewSalesEmps average max;
  class Job_Title;
  var Salary;
run;
```

```
ERROR 22-322: Syntax error, expecting one of the following:
  a name, a quoted string, (, /, ;, _DATA_, _LAST_, _NULL_).
```
SAS Data Set

- Define the components of a SAS data set.
- Define a SAS variable.
- Identify a missing value and a SAS date value.
- State the naming conventions for SAS data sets and variables.
- Browse the descriptor portion of SAS data sets by using the CONTENTS procedure.
- Browse the data portion of SAS data sets by using the PRINT procedure.
SAS Data Set

- A file that SAS creates and processes

<table>
<thead>
<tr>
<th>Data Set Name</th>
<th>WORK. NEWSALESEMP5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engine</td>
<td>V9</td>
</tr>
<tr>
<td>Created</td>
<td>Fri, Feb 08, 2008 01:40 PM</td>
</tr>
<tr>
<td>Observations</td>
<td>71</td>
</tr>
<tr>
<td>Variables</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>First_Name</th>
<th>Last_Name</th>
<th>Job_Title</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Satyakam</td>
<td>Denny</td>
<td>Sales Rep. II</td>
<td>26780</td>
</tr>
<tr>
<td>Monica</td>
<td>Kletschkus</td>
<td>Sales Rep. IV</td>
<td>30890</td>
</tr>
<tr>
<td>Kevin</td>
<td>Lyon</td>
<td>Sales Rep. I</td>
<td>26955</td>
</tr>
<tr>
<td>Petrea</td>
<td>Soltau</td>
<td>Sales Rep. II</td>
<td>27440</td>
</tr>
</tbody>
</table>
PROC CONTENTS DATA=SAS-data-set;
RUN;

The CONTENTS Procedure

<table>
<thead>
<tr>
<th>Data Set Name</th>
<th>WORK.NEWSALESEMP</th>
<th>Observations</th>
<th>71</th>
</tr>
</thead>
<tbody>
<tr>
<td>Member Type</td>
<td>DATA</td>
<td>Variables</td>
<td>4</td>
</tr>
<tr>
<td>Engine</td>
<td>V9</td>
<td>Indexes</td>
<td>0</td>
</tr>
<tr>
<td>Created</td>
<td>Wed, Jan 16, 2008</td>
<td>Observation Length</td>
<td>64</td>
</tr>
<tr>
<td>Last Modified</td>
<td>Wed, Jan 16, 2008</td>
<td>Deleted Observations</td>
<td>0</td>
</tr>
<tr>
<td>Protection</td>
<td></td>
<td>Compressed</td>
<td>NO</td>
</tr>
<tr>
<td>Data Set Type</td>
<td></td>
<td>Sorted</td>
<td>NO</td>
</tr>
<tr>
<td>Label</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Alphabetic List of Variables and Attributes

<table>
<thead>
<tr>
<th>#</th>
<th>Variable</th>
<th>Type</th>
<th>Len</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>First_Name</td>
<td>Char</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>Job_Title</td>
<td>Char</td>
<td>25</td>
</tr>
<tr>
<td>2</td>
<td>Last_Name</td>
<td>Char</td>
<td>18</td>
</tr>
<tr>
<td>4</td>
<td>Salary</td>
<td>Num</td>
<td>8</td>
</tr>
</tbody>
</table>
SAS data portion

- Organized as a table of observations (rows) and variables (columns).
- SAS stores date values as numeric values.
- Stored as the number of days between January 1, 1960, and a specific date.
- Missing values are valid values in a SAS data set.
- Variable names must start with a letter or underscore.

```
PROC PRINT DATA=SAS-data-set;
RUN;
```

Partial PROC PRINT Output

<table>
<thead>
<tr>
<th>Obs</th>
<th>First_Name</th>
<th>Last_Name</th>
<th>Job_Title</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Satyakam</td>
<td>Denny</td>
<td>Sales Rep. II</td>
<td>26780</td>
</tr>
<tr>
<td>2</td>
<td>Monica</td>
<td>Kletschkus</td>
<td>Sales Rep. IV</td>
<td>30890</td>
</tr>
</tbody>
</table>
SAS data portion

```sas
proc print data=work.NewSalesEmps noobs;
   var Last_Name First_Name Salary;
run;
```

Partial PROC PRINT Output

<table>
<thead>
<tr>
<th>Last_Name</th>
<th>First_Name</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Denny</td>
<td>Satyakam</td>
<td>26780</td>
</tr>
<tr>
<td>Kletschkus</td>
<td>Monica</td>
<td>30890</td>
</tr>
<tr>
<td>Lyon</td>
<td>Kevin</td>
<td>26955</td>
</tr>
<tr>
<td>Soltan</td>
<td>Petrea</td>
<td>27440</td>
</tr>
</tbody>
</table>
SAS Data Libraries

- Work – temporary library
- Sasuser – permanent library
- Can also create and access your own permanent libraries
- SAS file has a two-level name `libref.filename`

```
LIBNAME libref 'SAS-data-library' <options>;
```

```
libname orion 's:\workshop';
```
data NewSalesEmps;
  length First_Name $ 12
  Last_Name $ 18 Job_Title $ 25;
  infile 'newemps.csv' dlm=',';
  input First_Name $ Last_Name $
  Job_Title $ Salary;
run;

proc print data=work.NewSalesEmps;
run;

PROC CONTENTS DATA=libref._ALL_ NODS;
RUN;
Reading SAS Data Sets

LIBNAME libref 'SAS-data-library';

DATA output-SAS-data-set;
  SET input-SAS-data-set;
  WHERE where-expression;
  KEEP variable-list;
  LABEL variable = 'label';
  variable = 'label';
  variable = 'label';
  FORMAT variable(s) format;
RUN;
Reading SAS Data Sets

```
libname orion 's:\workshop';
data work.subset1;
   set orion.sales;
run;

proc print data=work.subset1;
run;
```

Both data sets contain 165 observations and 9 variables

```
data work.subset1;
set orion.sales;
run;
```

NOTE: There were 165 observations read from the data set ORION.SALES.
NOTE: The data set WORK.SUBSET1 has 165 observations and 9 variables.
Subset observations and variables

- Adding statements to the DATA and PROC step

```
WHERE where-expression;
```

Examples:
```
WHERE Gender = 'M';
WHERE Salary > 50000;
```
## Subset observations and variables

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Mnemonic</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>EQ</td>
<td>equal to</td>
</tr>
<tr>
<td>^=</td>
<td>NE</td>
<td>not equal to</td>
</tr>
<tr>
<td>^=</td>
<td>NE</td>
<td>not equal to</td>
</tr>
<tr>
<td>^=</td>
<td>NE</td>
<td>not equal to</td>
</tr>
<tr>
<td>&gt;</td>
<td>GT</td>
<td>greater than</td>
</tr>
<tr>
<td>&lt;</td>
<td>LT</td>
<td>less than</td>
</tr>
<tr>
<td>&gt;=</td>
<td>GE</td>
<td>greater than or equal</td>
</tr>
<tr>
<td>&lt;=</td>
<td>LE</td>
<td>less than or equal</td>
</tr>
<tr>
<td></td>
<td>IN</td>
<td>equal to one of a list</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>**</td>
<td>exponentiation</td>
</tr>
<tr>
<td>*</td>
<td>multiplication</td>
</tr>
<tr>
<td>/</td>
<td>division</td>
</tr>
<tr>
<td>+</td>
<td>addition</td>
</tr>
<tr>
<td>-</td>
<td>subtraction</td>
</tr>
</tbody>
</table>

```
where Gender ne 'M' and Salary >=50000;

where Gender ne 'M' or Salary >= 50000;

where Country = 'AU' or Country = 'US';

where Country not in ('AU' 'US');
```
Subset observations and variables

```
where Job_Title contains 'Rep';
where Employee_ID is missing;
where Employee_ID is null;
```

The DROP and KEEP Statements

```
data work.subset1;
   set orion.sales;
   where Country='AU' and
        Job_Title contains 'Rep';
   keep First_Name Last_Name Salary
        Job_Title Hire_Date;
run;
```
Producing Summary Reports

```sas
proc freq data=orion.sales;
  tables Gender Country;
run;
```

```
proc freq data=orion.sales;
run;
```

**The FREQ Procedure**

<table>
<thead>
<tr>
<th>Gender</th>
<th>Frequency</th>
<th>Percent</th>
<th>Cumulative Frequency</th>
<th>Cumulative Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>68</td>
<td>41.21</td>
<td>68</td>
<td>41.21</td>
</tr>
<tr>
<td>M</td>
<td>97</td>
<td>58.79</td>
<td>165</td>
<td>100.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Country</th>
<th>Frequency</th>
<th>Percent</th>
<th>Cumulative Frequency</th>
<th>Cumulative Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>AU</td>
<td>63</td>
<td>38.18</td>
<td>63</td>
<td>38.18</td>
</tr>
<tr>
<td>US</td>
<td>102</td>
<td>61.82</td>
<td>165</td>
<td>100.00</td>
</tr>
</tbody>
</table>

**one-way frequency tables**
Producing Summary Reports

```sql
proc freq data=orion.sales;
  tables Gender*Country;
run;
```

two-way frequency table

The FREQ Procedure

Table of Gender by Country

<table>
<thead>
<tr>
<th>Gender</th>
<th>Country</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AU</td>
<td>US</td>
</tr>
<tr>
<td>F</td>
<td>27</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>16.36</td>
<td>24.85</td>
</tr>
<tr>
<td></td>
<td>39.71</td>
<td>60.29</td>
</tr>
<tr>
<td></td>
<td>42.86</td>
<td>40.20</td>
</tr>
<tr>
<td>M</td>
<td>36</td>
<td>61</td>
</tr>
<tr>
<td></td>
<td>21.82</td>
<td>36.97</td>
</tr>
<tr>
<td></td>
<td>37.11</td>
<td>62.89</td>
</tr>
<tr>
<td></td>
<td>57.14</td>
<td>59.80</td>
</tr>
<tr>
<td>Total</td>
<td>63</td>
<td>102</td>
</tr>
<tr>
<td></td>
<td>38.18</td>
<td>61.82</td>
</tr>
</tbody>
</table>
Additional statements can be added to enhance the report.

```sas
proc format;
    value $ctryfmt 'AU'='Australia'
            'US'='United States';
run;

options nodate pageno=1;
ods html file='p112d01.html';
proc freq data=orion.sales;
    tables Gender*Country;
    where Job_Title contains 'Rep';
    format Country $ctryfmt.;
    title 'Sales Rep Frequency Report';
run;
ods html close;
```
### Sales Rep Frequency Report

#### The FREQ Procedure

<table>
<thead>
<tr>
<th>Gender</th>
<th>Australia</th>
<th>United States</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>27</td>
<td>40</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td>16.98</td>
<td>25.16</td>
<td>42.14</td>
</tr>
<tr>
<td></td>
<td>40.30</td>
<td>59.70</td>
<td></td>
</tr>
<tr>
<td></td>
<td>44.26</td>
<td>40.82</td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>34</td>
<td>58</td>
<td>92</td>
</tr>
<tr>
<td></td>
<td>21.38</td>
<td>36.48</td>
<td>57.86</td>
</tr>
<tr>
<td></td>
<td>36.96</td>
<td>63.04</td>
<td></td>
</tr>
<tr>
<td></td>
<td>55.74</td>
<td>59.18</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>61</td>
<td>98</td>
<td>159</td>
</tr>
<tr>
<td></td>
<td>38.36</td>
<td>61.64</td>
<td>100.00</td>
</tr>
</tbody>
</table>
Thanks !!

Questions ?