

**2018**

**WESTERN USERS OF  
SAS SOFTWARE**

**SACRAMENTO, CA.**



# Introduction to SAS Procedures: SAS Basics III

---

Susan J. Slaughter, Avocet Solutions

# SAS Essentials

Section for people new to SAS

Core presentations

1. How SAS Thinks
2. Introduction to DATA Step Programming
3. Introduction to SAS Procedures

We'll go fast

Slides are on my website

There will be a test

Do you have the handout?



# DATA versus PROC steps

Two basic parts of SAS programs

## DATA step

Begin with DATA statement

Input and modify data

Create SAS data set

Flexibility of programming

## PROC step

Begin with PROC statement

Perform analysis or task

Produce report

Like filling out a form

Susan says: This is a simplification



# SAS Procedures

In SAS 9.4 there are 234 procedures

Base SAS alone has 66 procedures

Procedures perform many tasks

- Reporting
- Statistical analysis
- Econometric / Time series
- Graphics
- Utilities



# SAS Procedures

I will focus on sorting and reporting

- PROC CONTENTS
- PROC SORT
- PROC FORMAT
- PROC PRINT
- PROC FREQ
- PROC MEANS
- PROC SQL



# SAS Procedures

Most procedures use the DATA= option in PROC statement

Example:

```
PROC PRINT DATA = students;
```

If you don't specify DATA=, SAS uses most recently created data set



# SAS Procedures

Every procedure is different, but there are similarities

Most procedures use these statements

- TITLE
- FOOTNOTE
- LABEL
- WHERE
- BY
- FORMAT



# TITLE statements

Can have up to 10 titles

```
TITLE 'This is a title';  
TITLE2 'This is another title';  
TITLE10 'Yet another title';
```

TITLE statements are global

- Stay in effect until you submit a new TITLE statement

To cancel, submit a null TITLE statement

```
TITLE;
```





# FOOTNOTE statements

Can have up to 10 footnotes

```
FOOTNOTE 'This is a footnote';  
FOOTNOTE2 'This is another footnote';  
FOOTNOTE10 'Yet another footnote';
```

FOOTNOTE statements are global

- Stay in effect until you submit a new FOOTNOTE statement

To cancel, submit a null FOOTNOTE statement

```
FOOTNOTE;
```



# LABEL statements

LABEL statements apply labels to variables

Labels are useful when variable names are too short

General form:

```
LABEL var = 'label' var = 'label';
```

Example:

```
LABEL Bdate = 'Birthdate' age = 'Age';
```

In a PROC step, labels apply only to that procedure.

In a DATA step, labels are saved in data set.



# WHERE statements

Tells SAS to use a subset

- Similar but not same as subsetting IF!

Can be used in DATA or PROC step

General form:

```
WHERE condition;
```

Examples:

```
WHERE Age >= 21;
```

```
WHERE Name = 'Wong' ;
```

```
WHERE Name = 'Wong' AND Age >= 21;
```



# Data for examples

```
* Input student enrollment data;
DATA students;
  INPUT ID $ Name $ AmtPaid Course $ New;
  DATALINES;
78374 Adam      350.00 597 1
75638 Michele  525.00 221 1
78634 Jacob     625.00 221 0
28746 .         .      597 2
58743 Zina      250.00 435 0
45378 Amy       250.00 435 0
87463 Angela    525.00 221 1
46732 Trevor    450.00 597 0
23867 Michael   450.00 597 0
;
RUN;
```



# PROC CONTENTS

SAS data sets have two portions

- data
- descriptor

PROC CONTENTS produces report of descriptor information

Example:

```
PROC CONTENTS DATA = students;  
RUN;
```



# PROC CONTENTS

## The SAS System

### The CONTENTS Procedure

<b>Data Set Name</b>	WORK.STUDENTS	<b>Observations</b>	9
<b>Member Type</b>	DATA	<b>Variables</b>	5
<b>Engine</b>	V9	<b>Indexes</b>	0
<b>Created</b>	08/31/2018 15:53:14	<b>Observation Length</b>	40
<b>Last Modified</b>	08/31/2018 15:53:14	<b>Deleted Observations</b>	0
<b>Protection</b>		<b>Compressed</b>	NO
<b>Data Set Type</b>		<b>Sorted</b>	NO
<b>Label</b>			
<b>Data Representation</b>	WINDOWS_64		
<b>Encoding</b>	wlatin1 Western (Windows)		



# PROC CONTENTS

## The SAS System

### The CONTENTS Procedure

Engine/Host Dependent Information	
Data Set Page Size	65536
Number of Data Set Pages	1
First Data Page	1
Max Obs per Page	1632
Obs in First Data Page	9
Number of Data Set Repairs	0
ExtendObsCounter	YES
Filename	C:\Users\Slaughter\Documents\My SAS Files\9.4\students.sas7bdat
Release Created	9.0401M3
Host Created	X64_8HOME



# PROC CONTENTS

## The SAS System

### The CONTENTS Procedure

Alphabetic List of Variables and Attributes			
#	Variable	Type	Len
3	AmtPaid	Num	8
4	Course	Char	8
1	ID	Char	8
2	Name	Char	8
5	New	Num	8





# PROC SORT

General form:

```
PROC SORT DATA=data-set-1 OUT=data-set-2;  
    BY variable-1 variable-2 .. variable-n;
```

If no OUT= then replaces data set

Default order ascending

To reverse use DESCENDING option

```
    BY DESCENDING variable;
```

Missing is always smallest (first in ascending order)



# PROC SORT

Example:

```
PROC SORT DATA = students OUT = studentsort;  
  BY Course Name;  
RUN;
```

	ID	Name	AmtPaid	Course	New
1	87463	Angela	525	221	1
2	78634	Jacob	625	221	0
3	75638	Michele	525	221	1
4	45378	Amy	250	435	0
5	58743	Zina	250	435	0
6	28746		.	597	2
7	78374	Adam	350	597	1
8	23867	Michael	450	597	0
9	46732	Trevor	450	597	0



# PROC SORT

Always check the SAS log!

```
76 PROC SORT DATA = students OUT = studentsort;  
77     BY Course Name;  
78 RUN;
```

NOTE: There were 9 observations read from the data set WORK.STUDENTS.

NOTE: The data set WORK.STUDENTSORT has 9 observations and 5 variables.

NOTE: PROCEDURE SORT used (Total process time):

real time	0.03 seconds
cpu time	0.01 seconds



# SAS Formats

Formats tell SAS how to display variables  
SAS has 100s of built-in formats

	<u>General form</u>	<u>Data</u>	<u>Format</u>	<u>Result</u>
Character	<i>\$formatw.</i>	alaska	\$UPCASE6.	ALASKA
Numeric	<i>formatw.d</i>	1000	COMMA8.2	1,000.00

Specify formats with a FORMAT statement



# FORMAT statements

Tells SAS how to display a variable

General form:

```
FORMAT var-list format. var-list format.;
```

Example:

```
FORMAT Price DOLLAR8.2 Date MMDDYY10.;
```

In a PROC step, formats apply only to that procedure.

In a DATA step, formats are saved in data set.



# PROC FORMAT

Create your own “user-defined” formats

General form:

```
PROC FORMAT;  
    VALUE name range-1 = 'formatted-text-1'  
           range-2 = 'formatted-text-2'  
           range-n = 'formatted-text-n' ;
```

Example:

```
PROC FORMAT;  
    VALUE newstu 1 = 'yes'  
           0 = 'no'  
    OTHER = '?' ;
```



# PROC PRINT

Produces detail reports

General form:

```
PROC PRINT options;  
VAR variable-list;
```

Options for PROC PRINT statement:

**NOOBS** removes observation numbers

**LABEL** use labels instead of variable names

Optional statements:

```
SUM variable-list; prints sums
```



# PROC PRINT

Example:

```
PROC PRINT DATA = studentsort;  
  VAR Course Name ID New AmtPaid;  
RUN;
```

The SAS System

Obs	Course	Name	ID	New	AmtPaid
1	221	Angela	87463	1	525
2	221	Jacob	78634	0	625
3	221	Michele	75638	1	525
4	435	Amy	45378	0	250
5	435	Zina	58743	0	250
6	597		28746	2	.
7	597	Adam	78374	1	350
8	597	Michael	23867	0	450
9	597	Trevor	46732	0	450





# PROC PRINT

Example:

```
PROC PRINT DATA = studentsort LABEL NOOBS ;  
  VAR Course Name ID New AmtPaid;  
  SUM AmtPaid;  
  WHERE AmtPaid NE .;  
  TITLE 'Fall Quarter Registrations';  
  FOOTNOTE 'Paid registrations only';  
  LABEL AmtPaid = 'Amount Paid'  
        ID = 'Student ID' New = 'New Student';  
  FORMAT AmtPaid DOLLAR9.2 New newstu. ;  
RUN;
```



# PROC PRINT

## Fall Quarter Registrations

Course	Name	Student ID	New Student	Amount Paid
221	Angela	87463	yes	\$525.00
221	Jacob	78634	no	\$625.00
221	Michele	75638	yes	\$525.00
435	Amy	45378	no	\$250.00
435	Zina	58743	no	\$250.00
597	Adam	78374	yes	\$350.00
597	Michael	23867	no	\$450.00
597	Trevor	46732	no	\$450.00
				<b>\$3,425.00</b>

Paid registrations only



# PROC FREQ

Produces frequencies/cross-tabulations/counts

General form:

```
PROC FREQ;  
    TABLES variable-combinations / options;
```

Options for TABLES statement:

**LIST** Prints results as a list rather than a table

**MISSING** Includes missing values in tabulations



# PROC FREQ

Example:

```
PROC FREQ DATA = students;  
  TABLES Course * New;  
RUN;
```

The SAS System  
The FREQ Procedure

Table of Course by New				
Course	New			
Frequency Percent Row Pct Col Pct	0	1	2	Total
221	1 11.11 33.33 20.00	2 22.22 66.67 66.67	0 0.00 0.00 0.00	3 33.33
435	2 22.22 100.0 0 40.00	0 0.00 0.00 0.00	0 0.00 0.00	2 22.22
597	2 22.22 50.00 40.00	1 11.11 25.00 33.33	1 11.11 25.00 100.00	4 44.44
Total	5 55.56	3 33.33	1 11.11	9 100.0 0



# PROC FREQ

Example:

```
PROC FREQ DATA = students;  
  TABLES Course * New / LIST;  
  WHERE AmtPaid NE .;  
  TITLE 'Fall Quarter Registrations';  
  FOOTNOTE 'Paid registrations only';  
  FORMAT New newstu. ;  
RUN;
```



# PROC FREQ

## Fall Quarter Registrations

The FREQ Procedure

Course	New	Frequency	Percent	Cumulative Frequency	Cumulative Percent
221	no	1	12.50	1	12.50
221	yes	2	25.00	3	37.50
435	no	2	25.00	5	62.50
597	no	2	25.00	7	87.50
597	yes	1	12.50	8	100.00

Paid registrations only



# PROC MEANS

Produces descriptive summary statistics

- Often used to output summary data sets
- Alias for PROC SUMMARY

General form:

```
PROC MEANS options;  
VAR variable-list;
```

Optional statements:

```
CLASS variable-list; Like BY, but data can be unsorted
```

Options for PROC MEANS statement:

```
MISSING MAXDEC = n (number decimal places)  
MAX, MIN, MEAN, MEDIAN, MODE, N, SUM
```



# PROC MEANS

Example:

```
PROC MEANS DATA = students;  
  VAR AmtPaid;  
RUN;
```

The SAS System  
The MEANS Procedure

Analysis Variable : AmtPaid				
N	Mean	Std Dev	Minimum	Maximum
8	428.1250000	135.2494389	250.0000000	625.0000000





# PROC MEANS

Example:

```
PROC MEANS DATA = students
    MAXDEC = 2 MIN MAX MEAN SUM ;
VAR AmtPaid;
CLASS Course;
TITLE 'Fall Quarter Registrations';
FOOTNOTE 'Paid registrations only';
LABEL AmtPaid = 'Amount Paid';
RUN;
```



# PROC MEANS

## Fall Quarter Registrations

The MEANS Procedure

Analysis Variable : AmtPaid Amount Paid					
Course	N Obs	Minimum	Maximum	Mean	Sum
221	3	525.00	625.00	558.33	1675.00
435	2	250.00	250.00	250.00	500.00
597	4	350.00	450.00	416.67	1250.00

Paid registrations only



# PROC SQL

PROC SQL allows you to run SQL statements

General form:

```
PROC SQL;  
    sql-statement;  
QUIT;
```

PROC SQL is an "interactive" procedure

- Can submit more SQL statements
- SQL statements execute immediately (no RUN needed)
- Step ends with QUIT statement (or DATA or PROC)



# PROC SQL

Example:

```
PROC SQL;  
    TITLE 'Students in Course 221';  
    SELECT * FROM students  
        WHERE Course = '221';  
QUIT;
```

**Students in Course 221**

ID	Name	AmtPaid	Course	New
75638	Michele	525	221	1
78634	Jacob	625	221	0
87463	Angela	525	221	1



# Output Delivery System

ODS handles all procedure output

Output formats are called destinations

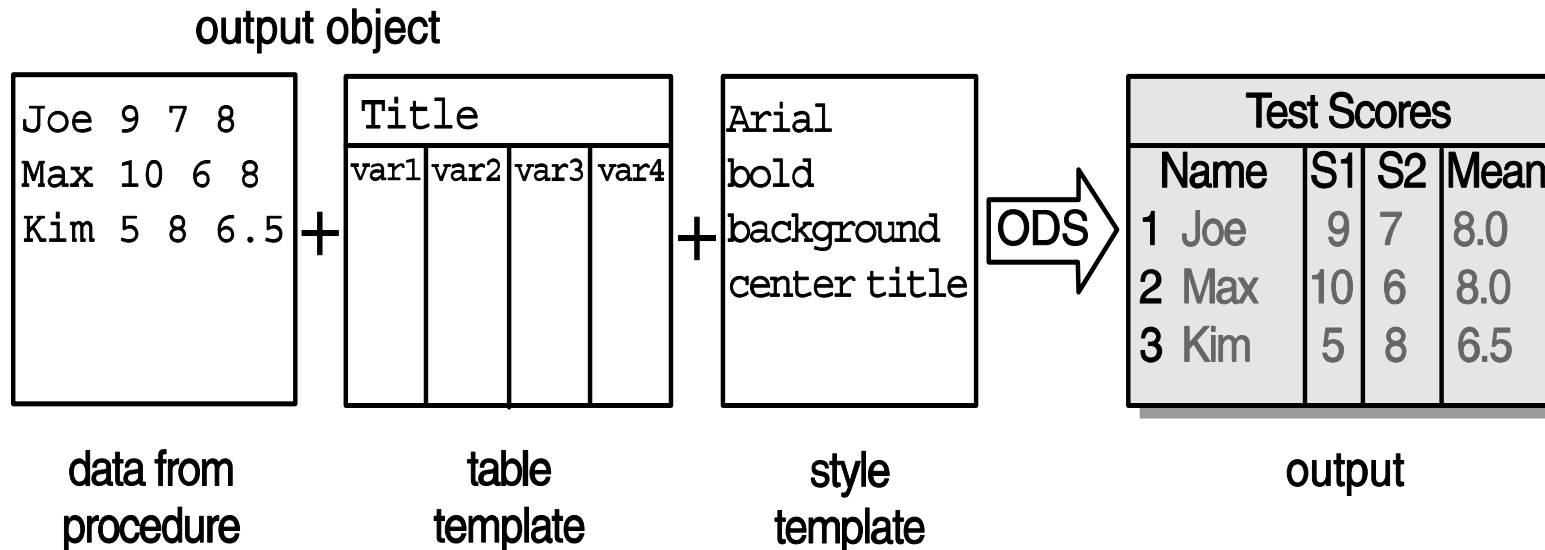
Many destinations

- HTML (default starting SAS 9.3)
- LISTING (text, default SAS 9.2 and earlier)
- PDF
- RTF
- POWERPOINT
- OUTPUT (SAS data set)



# Output Delivery System

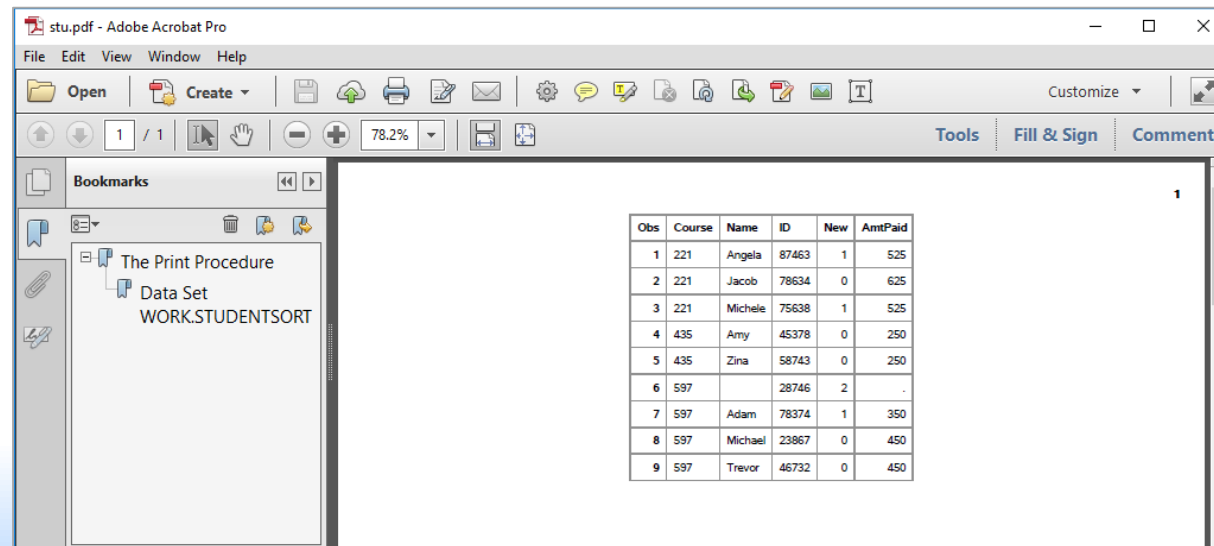
How ODS works:



# Changing ODS destination

Example:

```
ODS PDF FILE = 'c:\MyPDF\stu.pdf';  
PROC PRINT DATA = studentsort;  
  VAR Course Name ID New AmtPaid;  
RUN;  
ODS PDF CLOSE;
```



# ODS styles

SAS comes with many built-in styles including

- HTMLBLUE (default for HTML output)
- SASWEB
- BARRETTSBLUE
- ANALYSIS
- STATISTICAL
- JOURNAL

It is possible to create custom styles. The best way is in the Style Manager in Enterprise Guide.





# Changing ODS styles

Example:

```
ODS HTML STYLE = SASWEB FILE = 'c:\MyHTML\stu.html';  
PROC PRINT DATA = studentsort;  
  VAR Course Name ID New AmtPaid;  
RUN;  
ODS HTML CLOSE;
```

The SAS System

Obs	Course	Name	ID	New	AmtPaid
1	221	Angela	87463	1	525
2	221	Jacob	78634	0	625
3	221	Michele	75638	1	525
4	435	Amy	45378	0	250
5	435	Zina	58743	0	250
6	597		28746	2	.
7	597	Adam	78374	1	350
8	597	Michael	23867	0	450
9	597	Trevor	46732	0	450



# Pop quiz

1) What one statement is required by all procedures?

PROC statement

2) What data set will SAS procedures use by default if you do not specify a data set?

Most recently created data set

3) What does the LABEL statement do?

It applies labels to variables



# Pop quiz

- 4) Write a WHERE statement to keep only observations where the variable AmtPaid equals 525.

**WHERE AmtPaid = 525;**

- 5) List three pieces of information you could find in output from PROC CONTENTS.

Data set name, number of obs, number of vars, var names, var types, var lengths....

- 6) Which one procedure requires a BY statement?

**PROC SORT**



# Pop quiz

7) What value is always smallest in sort order?

Missing

8) Name a procedure you could use to produce counts.

PROC FREQ or MEANS (also TABULATE or REPORT)

9) What does the acronym ODS stand for?

Output Delivery System

10) Write a statement to change the style for HTML output to ANALYSIS.

```
ODS HTML STYLE = ANALYSIS;
```



# Other presentations

Beginner's Techniques

Thursday 8:30-2:30 in Big Sur



# Thank you!

## Enjoy the conference!

Susan Slaughter  
Avocet Solutions

Can download slides from [www.avocetsolutions.com](http://www.avocetsolutions.com)

