STAT 425 and STAT 625 Statistical Software

Lecture 4

More on Raw Data, Creating permanent SAS Data Sets

Saida Zardi

More on Reading Raw Data from External Files

Mixing Input Styles

- Mixing List Input, Column Input and Format input

Yellowstone	ID/MT/WY 1872	4,065,493
Everglades	Fl 1934	1,398,800
Yosemite	CA 1864	760,917
Great Smokey Mountains	NC/TN 1926	520,269
Wolf Trap Farm	VA 1966	130

```
* Creating a data file NatParks with mixing input styles;

Data NatParks;

   Infile 'J:\\CLASSES\STAT46\natparks.txt';
   Input parkname $1-22
         Sate $
         Year
         @40 Acreage COMMA9.;
Run;

proc print data=NatParks;
   Title 'Selected National Parks';

run;
```

parkname	column style Input
State and Year	List Input
Acreage	Formatted Input

- parkname → column style Input
- State and Year → List Input
- Acreage → Formatted Input

Selected National Parks

Obs	parkname	Sate	Year	Acreage
1	Yellowstone	ID/MT/wy	1872	4065493
2	Everglades	FI	1934	1398800
3	Yosemite	CA	1864	760917
4	Great Smokey Mountains	NC/TN	1926	520269
5	Wolf Trap Farm	VA	1966	130

Reading Messy Data

The data don’t line up in nice column or have predictable length

Example:

```
The Kraken School: ASU Time: 1:45.35
Koicrete School: UoA Time: 1:33.7
Max School: UCSD Time: 1:26.47
Prospector School: CPSLO Time: 1:12.08
```

Use @’character’ column pointer and the colon modifier

```
Data results;
  Infile 'J:\\\CLASSES\STAT46\competition.txt';
  Input @'School:'  School $
        @'Time:'    RaceTime :STIMER8.;
RUN;

PROC PRINT data = results;
    title 'Competition Results';
Run;
```

Output

Competition Results

Obs	School	RaceTime
1	ASU	105.35
2	UoA	93.70
3	UCSD	86.47
4	CPSLO	72.08

Creating Permanent Data Sets

Why create permanent data sets?

If you need to use your data set more than once it makes sense to create them permanent and reuse them each time you need, especially when the data set is relatively large.

Libref

When a data set is created by a data statement, for example:

Data Mydata;

SAS adds it to a temporary library: WORK

A SAS data set name has two parts:

libref.data-set-name

Example:

data Work.Mydata;

The Libname Statement

```
libname lecture4 'J:\\CLASSES\STAT46';

DATA lecture4.test_scores;
     length ID $ 3 Name $ 15;
     input ID $ Score1-Score3 Name $;
datalines;
1 90 95 85 Jean
2 95 85 75 Jill
3 75 65 70 Mona
4 95 95 95 Elena
;
run;
```

- The *Libname* statement needs to be reissued each time you open a new SAS session.
- The data set extension is namr-of-file.sas7bdat.

Example: test_scores.sas7bdat

The Descriptor Portion of a SAS Data Set Using Proc contents

- A SAS data set consists of two parts: a **Descriptor** portion and a **Data** portion.
- The descriptor portion: using Proc Content

```
title 'The Descriptor Portion of the Data Set Test_Scores';
proc contents data=lecture4.test_scores;
run;
```

The Descriptor Portion

- The output has three parts:
- The first one:

The Descriptor Portion of the Data Set Test_Scores

The CONTENTS Procedure

Data Set Name	LECTURE4.TEST_SCORES	Observations	4
Member Type	DATA	Variables	5
Engine	V9	Indexes	0
Created	09/13/2018 15:04:22	Observation Length	48
Last Modified	09/13/2018 15:04:22	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	WINDOWS_64		
Encoding	wlatin1 Western (Windows)		

The Descriptor Portion of a SAS Data Set Using Proc contents

Alphabetic List of Variables and Attributes			
#	Variable	Type	Len
1	ID	Char	3
2	Name	Char	15
3	Score1	Num	8
4	Score2	Num	8
5	Score3	Num	8

The Descriptor Portion of a SAS Data Set Using Proc contents

Engine/Host Dependent Information	
Data Set Page Size	65536
Number of Data Set Pages	1
First Data Page	1
Max Obs per Page	1361
Obs in First Data Page	4
Number of Data Set Repairs	0
ExtendObsCounter	YES
Filename	J:\CLASSES\STAT46\test_scores.sas7bdat
Release Created	9.0401M4
Host Created	X64_SR12R2
Owner Name	AMERICAN\zardi
File Size	128KB
File Size (bytes)	131072

The Descriptor Portion of a SAS Data Set Using Proc contents

- The VARNUM option: List the variables in the data set order

```
proc contents data=lecture4.test_scores varnum;
run;
```

Variables in Creation Order

#	Variable	Type	Len
1	ID	Char	3
2	Name	Char	15
3	Score1	Num	8
4	Score2	Num	8
5	Score3	Num	8

The Descriptor Portion of a SAS Data Set Using Proc contents

- New Libname at a new SAS session:

The libname can be changed when you open a new SAS session but the folder name must be the same to retrieve the data set.

```
Libname session4 'J:\\CLASSES\STAT46';
title 'The Descriptor Portion of the Data Set Test_Scores';
proc contents data=session4.test_scores varnum;
run;
```

The Descriptor Portion of a SAS Data Set Using Proc contents

- The POSITION option:

```
proc contents data=lecture4.test_scores  position;
run;
```

Both lists of varaibles are displayed:

Alphabetic List of Variables and Attributes

#	Variable	Type	Len
1	ID	Char	3
2	Name	Char	15
3	Score1	Num	8
4	Score2	Num	8
5	Score3	Num	8

Variables in Creation Order

#	Variable	Type	Len
1	ID	Char	3
2	Name	Char	15
3	Score1	Num	8
4	Score2	Num	8
5	Score3	Num	8

The Descriptor Portion of a SAS Data Set Using Proc contents

- List the Names of all data sets in a SAS Library

```
title 'Posting All the SAS Data Sets in a Library';
proc contents data=lecture4._all_ nods;
run;
```

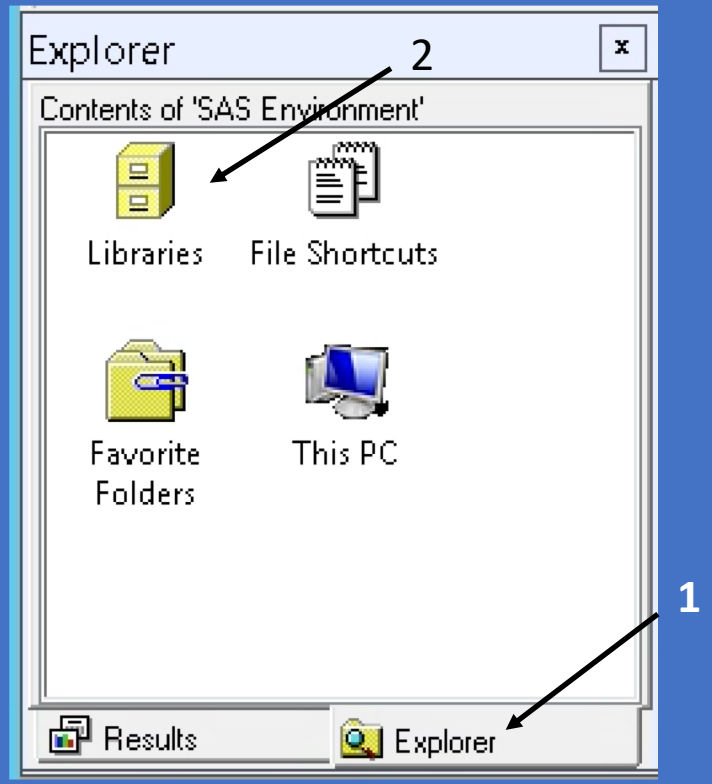
Posting All the SAS Data Sets in a Library

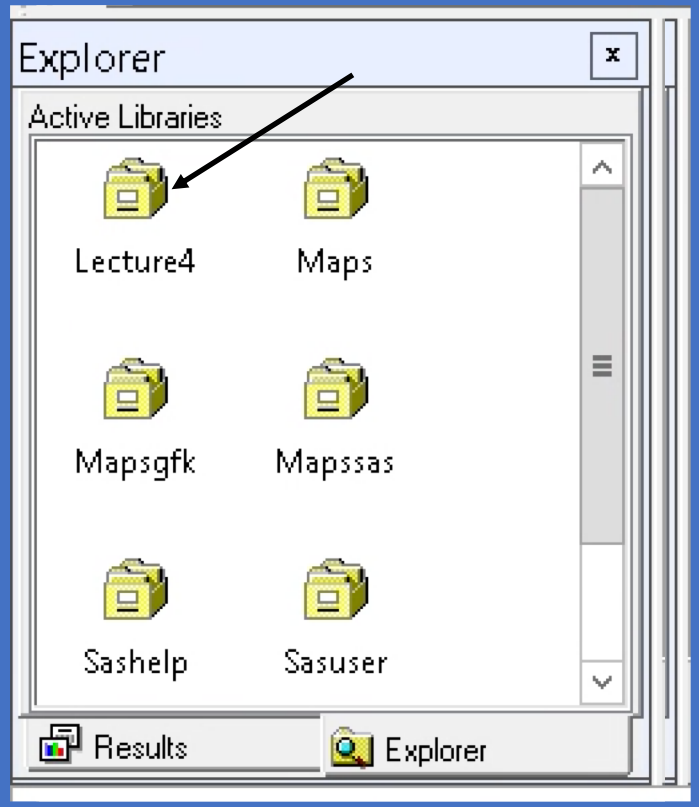
The CONTENTS Procedure

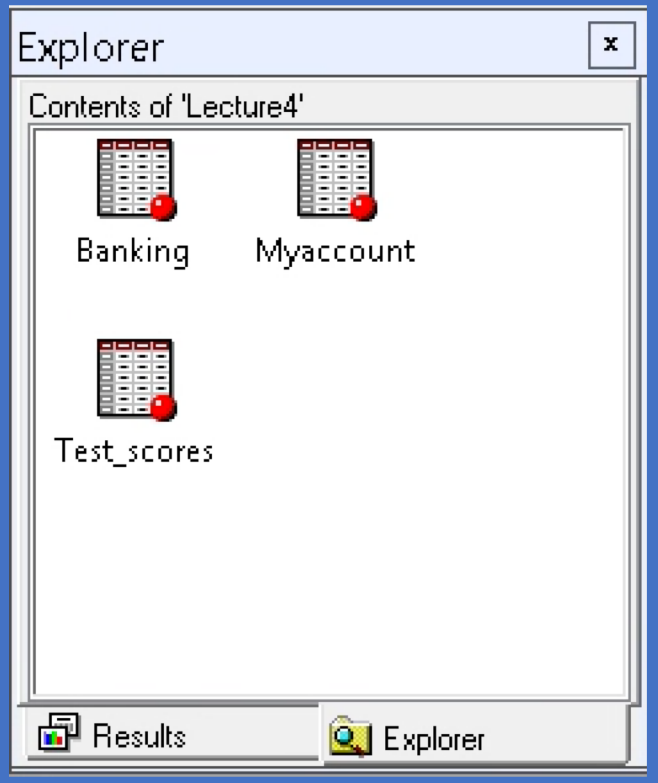
Directory	
Libref	LECTURE4
Engine	V9
Physical Name	J:\\CLASSES\STAT46
Filename	J:\\CLASSES\STAT46
Owner Name	AMERICAN\ahenning-adm
File Size	4KB
File Size (bytes)	4096

#	Name	Member Type	File Size	Last Modified
1	BANKING	DATA	128KB	09/10/2018 16:06:56
2	MYACCOUNT	DATA	128KB	09/10/2018 16:37:20
3	TEST_SCORES	DATA	128KB	09/13/2018 15:04:23

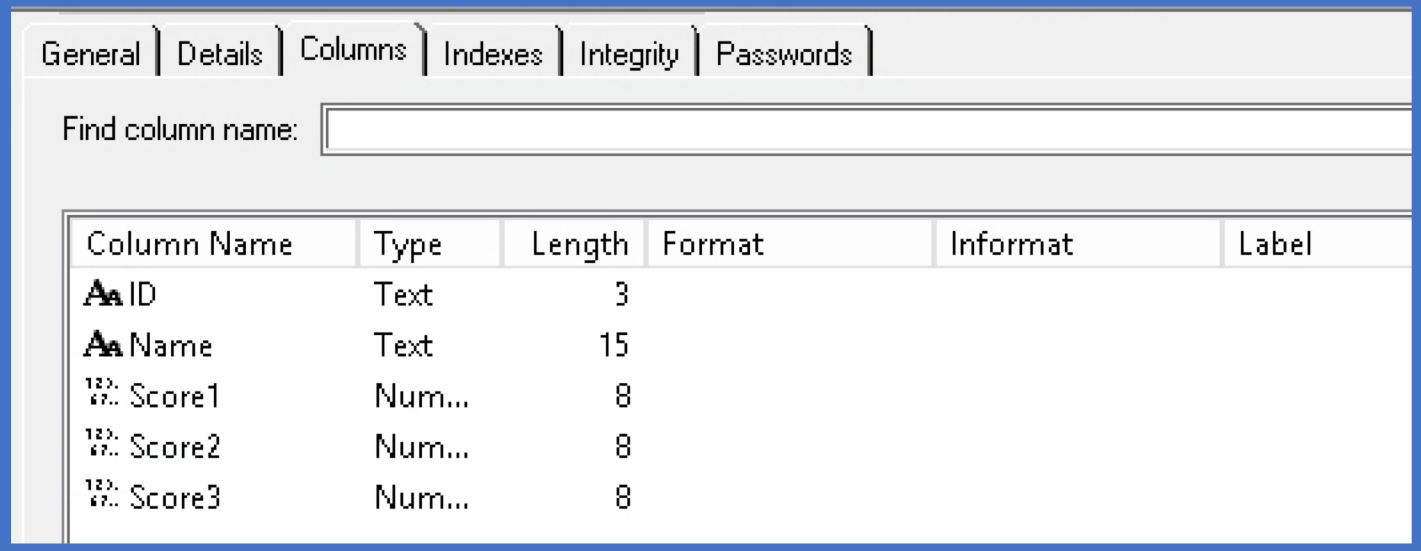
The Descriptor Portion of a SAS Data Set Using Point-and-Click approach







The Descriptor Portion of a SAS Data Set Using Point-and-Click approach



Right click

	ID	Name	Score1	Score2	Score3
1	1	Jean	90	95	85
2	2	Jill	95	85	75
3	3	Mona	75	65	70
4	4	Elena	95	95	95

Left click

The DATA Portion of a SAS Data Set

```
title 'Listing of the Data Set Test_Scores';
proc print data = lecture4.test_scores;
run;
```

Listing of the Data Set Test_Scores

Obs	ID	Name	Score1	Score2	Score3
1	1	Jean	90	95	85
2	2	Jill	95	85	75
3	3	Mona	75	65	70
4	4	Elena	95	95	95

Using a Data Set as Input to a Data Step

- Using a SAS data set as an input in a DATA step using the statement SET
- Example:

```
libname Lecture4 'J:\CLASSES\STAT46';

Data New;
   set Lecture4.Test_Scores;
   AvScore= mean(of score1-score3);
run;

title "Listing of Data Set New";

proc print data=New;
  var ID Score1-Score3 AvScore;
run;
```

Listing of Data Set New

Obs	ID	Score1	Score2	Score3	AvScore
1	1	90	95	85	90
2	2	95	85	75	85
3	3	75	65	70	70
4	4	95	95	95	95

DATA _NULL_: A Data Set That isn't

- Use Data set Name for applications where you want to process observations in a SAS data set, to print out errors or to produce a report, without the need to create a new data set.
- The reserved name _NULL_ tells SAS not to create a data set.

DATA _NULL_: A Data Set That isn't An example

```
Title 'Score Greater Than or Equal To 95';
Data _NULL_;
   set Lecture4.Test_Scores;
   if Score1 ge 95 or Score2 ge 95 or Score3 ge 95
   then put ID= Score1= Score2= Score3= ;
run;
```

Log - (Untitled)

```
32    Title 'Score Greater Than or Equal To 95';
33    Data _NULL_;
34       set Lecture4.Test_Scores;
35       if Score1 ge 95 or Score2 ge 95 or Score3 ge 95
         then put ID= Score1= Score2= Score3= ;
         ;

         re1=90 Score2=95 Score3=85
         re1=95 Score2=85 Score3=75
         re1=95 Score2=95 Score3=95
         ere were 4 observations read from the data set LECTURE4.TEST_SCORES.
NOTE: DATA statement used (Total process time):
      real time           0.04 seconds
      cpu time            0.00 seconds
```

DATA _NULL_: A Data Set That isn't

- If you want to send the output to a file, you need to place a file statement before the PUT statement:

  File 'J:\Classes\STAT46\Highscores.txt';

- A file statement si similar to an Infile statement: it tells SAS the destination of the text you're outputting.
- To write the results, of the PUT statement, on the output device you can use the reserved file reference PRINT:

  file print;

Data_NULL_ are sometimes used to create custom reports.