PSTAT 130

SAS BASE PROGRAMMING

- Lecture 4 -

Dr. Julie Swenson

Objectives

- The CONTENTS Procedure
- The SORT Procedure
- Create SAS Data Sets

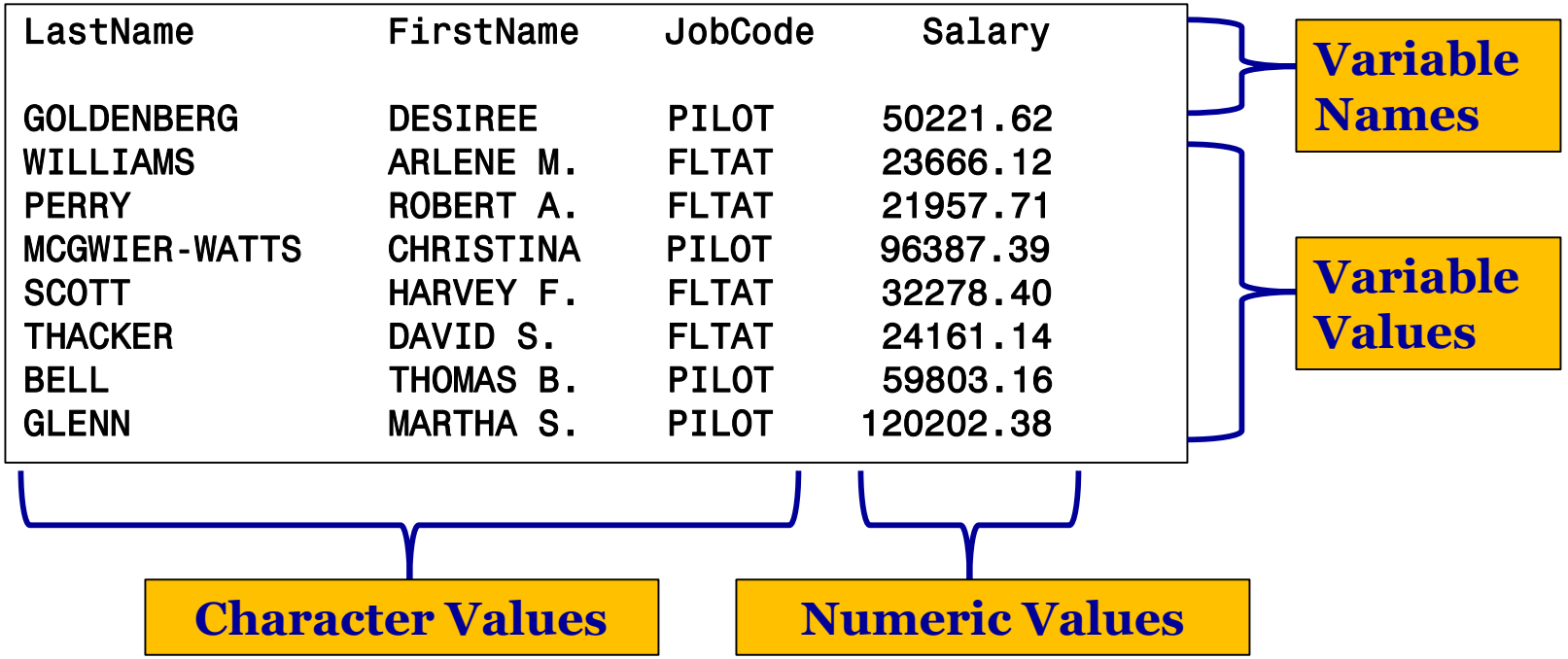
SAS Data Sets

- SAS data sets have a descriptor portion and a data portion

Descriptor Portion	General data set information * data set name * data set label * date/time created * storage information * number of observations Information for each variable * Name * Type * Length * Position * Format * Informat * Label
Data Portion	Observations for each variable Name Age Height Weight John 19 69 180 Mary 22 63 130 John 21 67 165

Data Portion

- The data portion of a SAS data set is a rectangular table of character and/or numeric data values



Note: Missing Data Values

- A value must exist for every variable for each observation. Missing values are valid values.

LastName	FirstName	JobCode	Salary
GOLDENBERG	DESIREE	PILOT	50221.62
WILLIAMS	ARLENE M.	FLTAT	23666.12
PERRY	ROBERT A.	FLTAT	.
MCGWIER-WATTS	CHRISTINA	PILOT	96387.39
SCOTT	HARVEY F.		32278.40

A character missing value is displayed as a blank.

A numeric missing value is displayed as a period.

Descriptor Portion

- The descriptor portion of a SAS data set contains
  - General information about the SAS data set (such as data set name and number of observations)
  - Variable attributes (name, type, length, position, informat, format, label)

- The CONTENTS procedure (`PROC CONTENTS`) displays the descriptor portion of a SAS data set

The CONTENTS Procedure

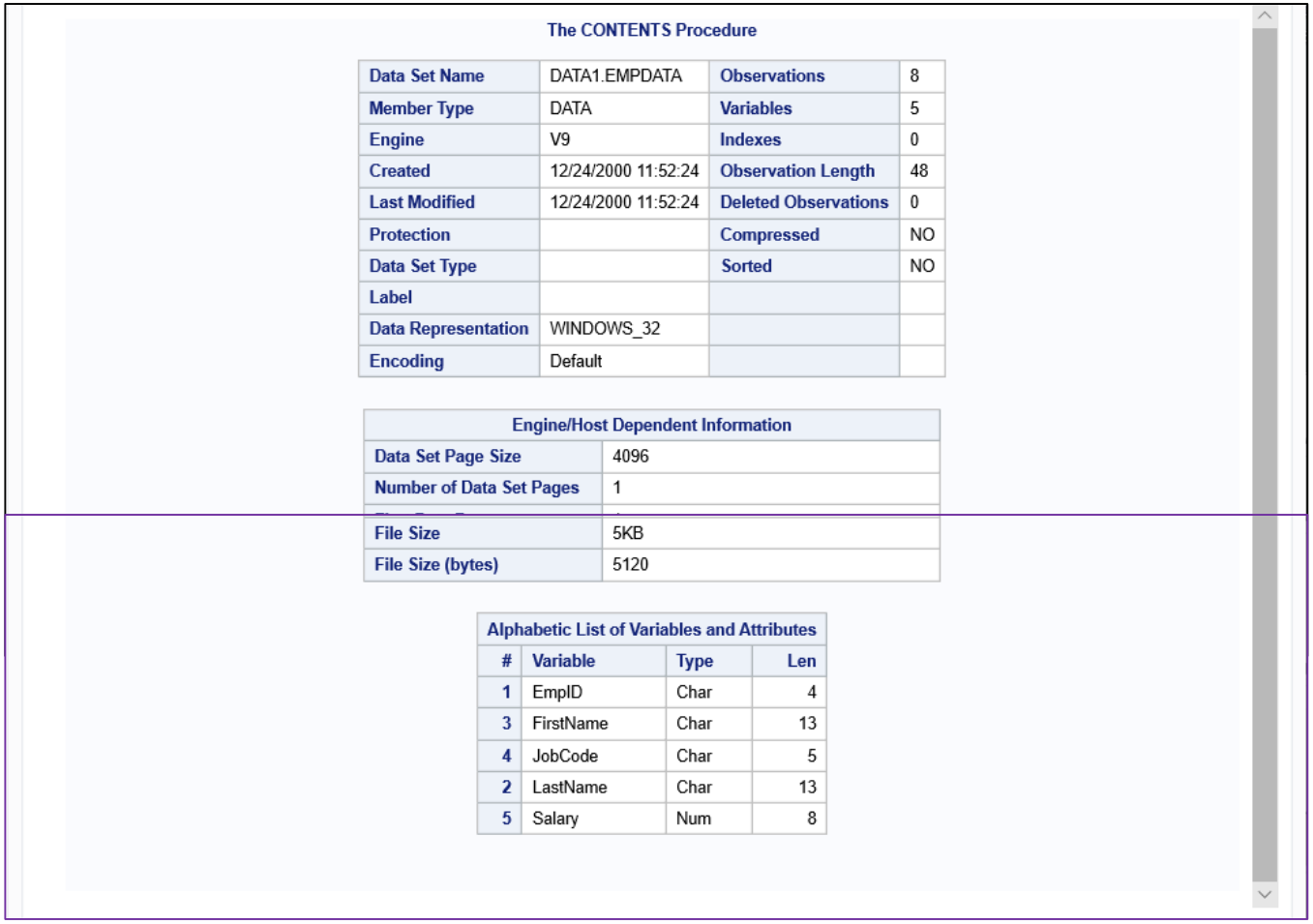
- General form of the CONTENTS procedure:

```
PROC CONTENTS DATA=<SAS-data-set>;
RUN;
```

- Example

```
proc contents data=data1.empdata;
run;
```

Descriptor Portion



The CONTENTS Procedure

Data Set Name	DATA1.EMPDATA	Observations	8
Member Type	DATA	Variables	5
Engine	V9	Indexes	0
Created	12/24/2000 11:52:24	Observation Length	48
Last Modified	12/24/2000 11:52:24	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	WINDOWS_32		
Encoding	Default		

Engine/Host Dependent Information	
Data Set Page Size	4096
Number of Data Set Pages	1
File Size	5KB
File Size (bytes)	5120

Alphabetic List of Variables and Attributes

#	Variable	Type	Len
1	EmpID	Char	4
3	FirstName	Char	13
4	JobCode	Char	5
2	LastName	Char	13
5	Salary	Num	8

The CONTENTS Procedure

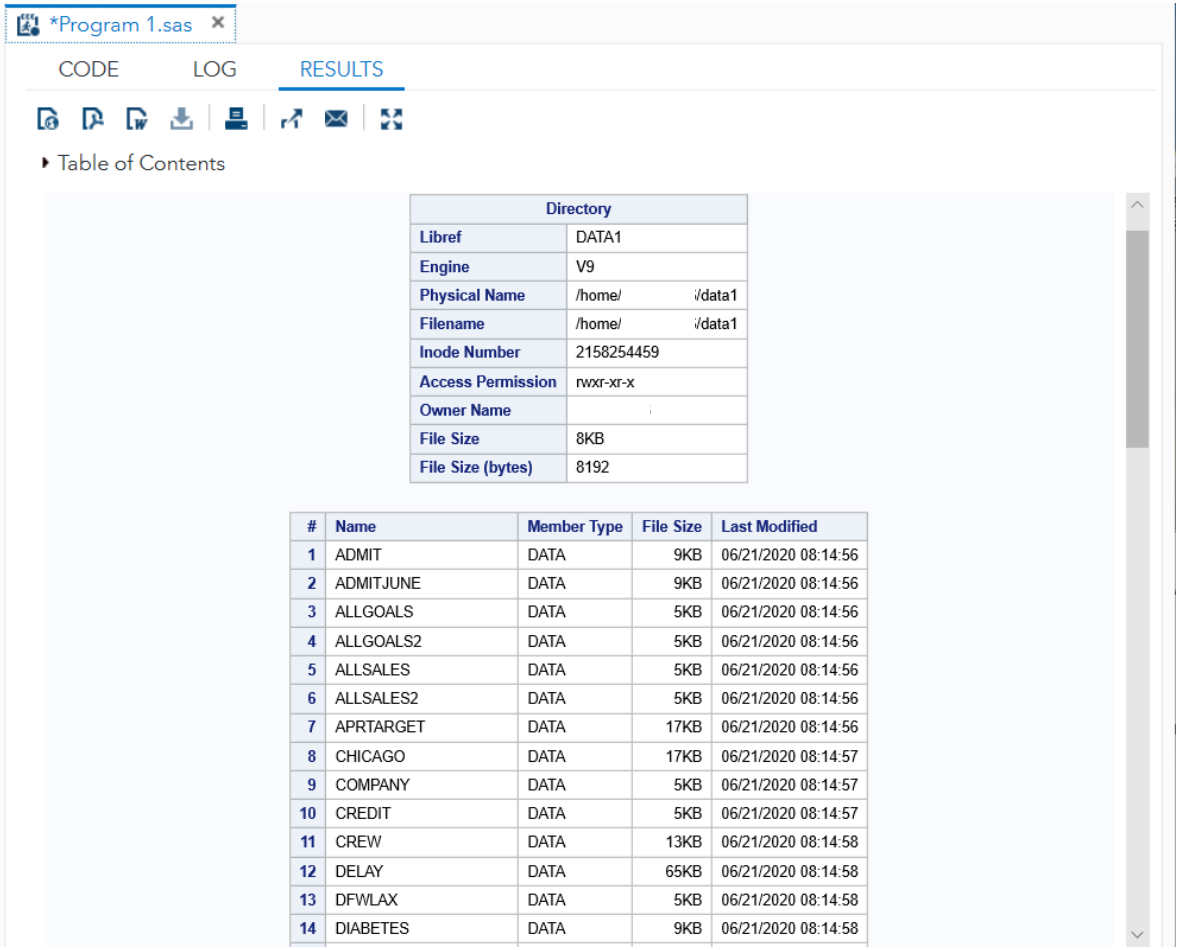
- Contents of entire library
  - Using the `_ALL_` keyword to list all the SAS files in the library and the `NODS` option to suppress the descriptor portions of the data sets.
  - General form of the `NODS` option

```
PROC CONTENTS DATA=libref._ALL_ NODS;
RUN;
```

  - `NODS` must be used in conjunction with the keyword _ALL_

```
proc contents data=data1._all_ nods;
run;
```

The CONTENTS Procedure



Directory	
Libref	DATA1
Engine	V9
Physical Name	/home/ i/data1
Filename	/home/ i/data1
Inode Number	2158254459
Access Permission	rwxr-xr-x
Owner Name	
File Size	8KB
File Size (bytes)	8192

#	Name	Member Type	File Size	Last Modified
1	ADMIT	DATA	9KB	06/21/2020 08:14:56
2	ADMITJUNE	DATA	9KB	06/21/2020 08:14:56
3	ALLGOALS	DATA	5KB	06/21/2020 08:14:56
4	ALLGOALS2	DATA	5KB	06/21/2020 08:14:56
5	ALLSALES	DATA	5KB	06/21/2020 08:14:56
6	ALLSALES2	DATA	5KB	06/21/2020 08:14:56
7	APRTARGET	DATA	17KB	06/21/2020 08:14:56
8	CHICAGO	DATA	17KB	06/21/2020 08:14:57
9	COMPANY	DATA	5KB	06/21/2020 08:14:57
10	CREDIT	DATA	5KB	06/21/2020 08:14:57
11	CREW	DATA	13KB	06/21/2020 08:14:58
12	DELAY	DATA	65KB	06/21/2020 08:14:58
13	DFWLAX	DATA	5KB	06/21/2020 08:14:58
14	DIABETES	DATA	9KB	06/21/2020 08:14:58

Class Exercise 1

- Use the **`staff`** data set in the data1 folder.
- Output the data portion of the data set.
- Output the descriptor portion of the data set.

The SORT Procedure

- Sequencing and grouping observations
  - Sequence (sort) observations in a SAS data set
  - Group observations in a list report
  - Print column subtotals in a list report
  - Control page breaks for subgroups

Sort a SAS Data Set

- To request subgroup totals in `PROC PRINT`, the observations in the data set must be grouped

- The SORT procedure
  - Rearranges the observations in a SAS data set
  - Can create a new SAS data set containing the rearranged observations
  - Can sort on multiple variables
  - Can sort in ASCENDING (default) or DESCENDING order
  - Does not generate printed output
  - Treats missing values as the smallest possible value

Sort a SAS Data Set

- General form of the `PROC SORT` step

```
PROC SORT DATA=input-SAS-data-set
          <OUT=output-SAS-data-set>;
   BY <DESCENDING> by-variable(s);
RUN;
```

- Examples (sorts the data by `Salary`)

```
proc sort data=data1.empdata;
   by Salary;
run;
```

CAUTION!
Overwrites the original data set with the **sorted** data set

```
proc sort data=data1.empdata out=work.jobsal;
   by Salary;
run;
```

Saves the sorted data set in **new**_data

Sort a SAS Data Set

- When you include more than one SORT variable in the BY statement
  - SAS sorts the data set by the first variable listed
  - Then sorts by the second variable WITHIN the values of the first variable
  - Then sorts by the third variable WITHIN the values of the second variable, etc.

- By default, SAS sorts in ascending order. The keyword DESCENDING applies to the following variable.

Class Exercise 2

- Use the **admit** data set in the data1 folder, which contains hospital admitting information on patients
- Sort the **admit** data set into a NEW data set called **work.temp_admit**
  - Sort it by Actlevel Sex
  - Print the new data set
- Now sort by Sex Actlevel
  - Print the new data set

Create Subgroups

- The BY statement (in `PROC PRINT`) allows you to
  - Create a separation between listings of several subgroups
  - Identify each subgroup by the top of its listing
  - But, data set must be sorted on the BY variable first
- General form of the BY statement

```
BY variable(s);
```

- Example

```
proc print data=data1.admit;
    BY Sex;
run;
```

Sex=F

Obs	ID	Name	Age	Date	Height	Weight	ActLevel	Fee
1	2462	Almers, C	34	3	66	152	HIGH	124.80
2	2501	Bonaventure, T	31	17	61	123	LOW	149.75
3	2523	Johnson, R	43	31	63	137	MOD	149.75
4	2552	Reberson, P	32	9	67	151	MOD	149.75
5	2568	Eberhardt, S	49	27	64	172	LOW	124.80
6	2571	Nunnelly, A	44	19	66	140	HIGH	149.75
7	2572	Oberon, M	28	17	62	118	LOW	85.20
8	2575	Quigley, M	40	8	69	163	HIGH	124.80
9	2584	Takahashi, Y	43	29	65	123	MOD	124.80
10	2588	Ivan, H	22	20	63	139	LOW	85.20
11	2589	Wilcox, E	41	16	67	141	HIGH	149.75

Sex=M

Obs	ID	Name	Age	Date	Height	Weight	ActLevel	Fee
12	2458	Murray, W	27	1	72	168	HIGH	85.20
13	2539	LaMance, K	51	4	71	158	LOW	124.80
14	2544	Jones, M	29	6	76	193	HIGH	124.80
15	2555	King, E	35	13	70	173	MOD	149.75
16	2563	Pitts, D	34	22	73	154	LOW	124.80
17	2574	Peterson, V	30	6	69	147	MOD	149.75
18	2578	Cameron, L	47	5	72	173	MOD	124.80
19	2579	Underwood, K	60	22	71	191	LOW	149.75
20	2586	Derber, B	25	23	75	188	HIGH	85.20
21	2595	Warren, C	54	7	71	183	MOD	149.75

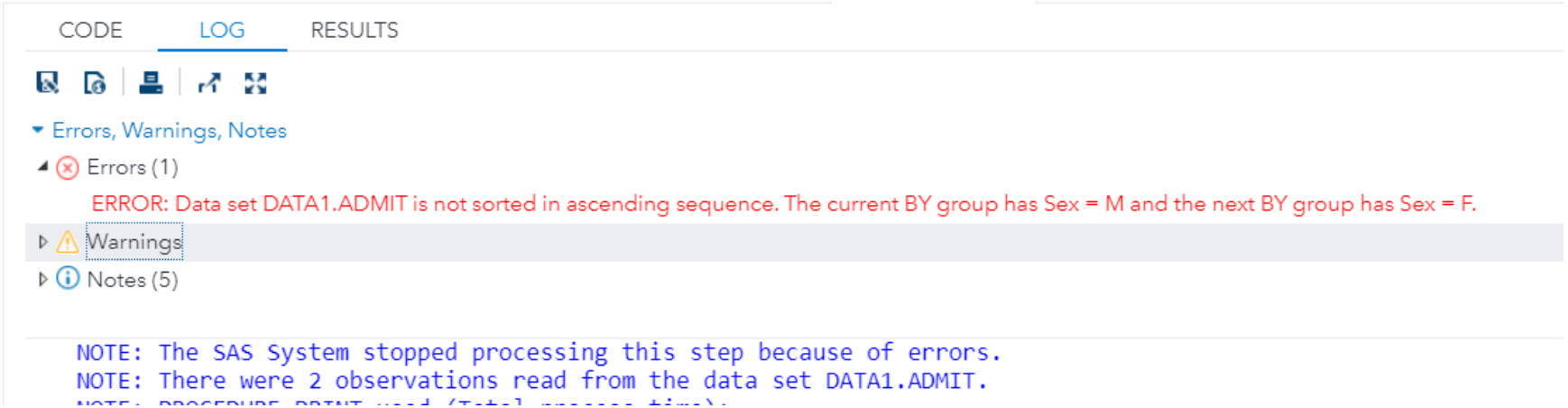
Print Sorted Observations

- If you attempt to use the BY statement on an unsorted data set, you will get some interesting results:

Sex=M

Obs	ID	Name	Age	Date	Height	Weight	ActLevel	Fee
1	2458	Murray, W	27	1	72	168	HIGH	85.20

- A look at the log file shows an error:



Print Sub & Grand Totals

- Using a BY statement and a SUM statement together in a `PROC PRINT` step produces subtotals and grand totals

- Print the data set grouped by `ActLevel` with a subtotal for the `Fee` column for each `ActLevel`

```
proc sort data=data1.admit
          out=work.admit;
   by ActLevel;
run;
proc print data=work.admit;
   by ActLevel;
   sum Fee;
run;
```

ActLevel=LOW

Obs	ID	Name	Sex	Age	Date	Height	Weight	Fee
8	2501	Bonaventure, T	F	31	17	61	123	149.75
9	2539	LaMance, K	M	51	4	71	158	124.80
10	2563	Pitts, D	M	34	22	73	154	124.80
11	2568	Eberhardt, S	F	49	27	64	172	124.80
12	2572	Oberon, M	F	28	17	62	118	85.20
13	2579	Underwood, K	M	60	22	71	191	149.75
14	2588	Ivan, H	F	22	20	63	139	85.20
ActLevel								844.30

ActLevel=MOD

Obs	ID	Name	Sex	Age	Date	Height	Weight	Fee
15	2523	Johnson, R	F	43	31	63	137	149.75
16	2552	Reberson, P	F	32	9	67	151	149.75
17	2555	King, E	M	35	13	70	173	149.75
18	2574	Peterson, V	M	30	6	69	147	149.75
19	2578	Cameron, L	M	47	5	72	173	124.80
20	2584	Takahashi, Y	F	43	29	65	123	124.80
21	2595	Warren, C	M	54	7	71	183	149.75
ActLevel								998.35
								2687.0

Page Breaks

- Use the PAGEBY statement to put each subgroup on a separate page
- General form of the PAGEBY statement

```
PAGEBY by-variable;
```

- Example

```
proc print data=work.admit;
   by ActLevel;
   pageby ActLevel;
   sum Fee;
run;
```

- The PAGEBY statement must be used with a BY statement

Identify Observations

- The ID statement allows you to
  - Suppress the Obs column in the report
  - Specify which variable(s) should replace the Obs column
- General form of the ID statement

```
ID variable(s);
```

- Example

```
proc print data=data1.empdata;
    id JobCode;
    var EmpID Salary;
run;
```

Special By-Group Format

- When the ID and BY statements specify the same variable
  - the Obs column is suppressed
  - the BY line is suppressed
  - the ID/BY variable prints in the leftmost column
  - each ID/BY value only prints at the start of each BY group (and on the subtotal line, if a SUM statement is used)

Special By-Group Format

- Specify JobCode in the BY and ID statements to change the report format.

```
proc sort data=data1.empdata out=work.empdata;
   by JobCode;
run;
proc print data=work.empdata;
   by JobCode;
   id JobCode;
   sum Salary;
run;
```

Special By-Group Format

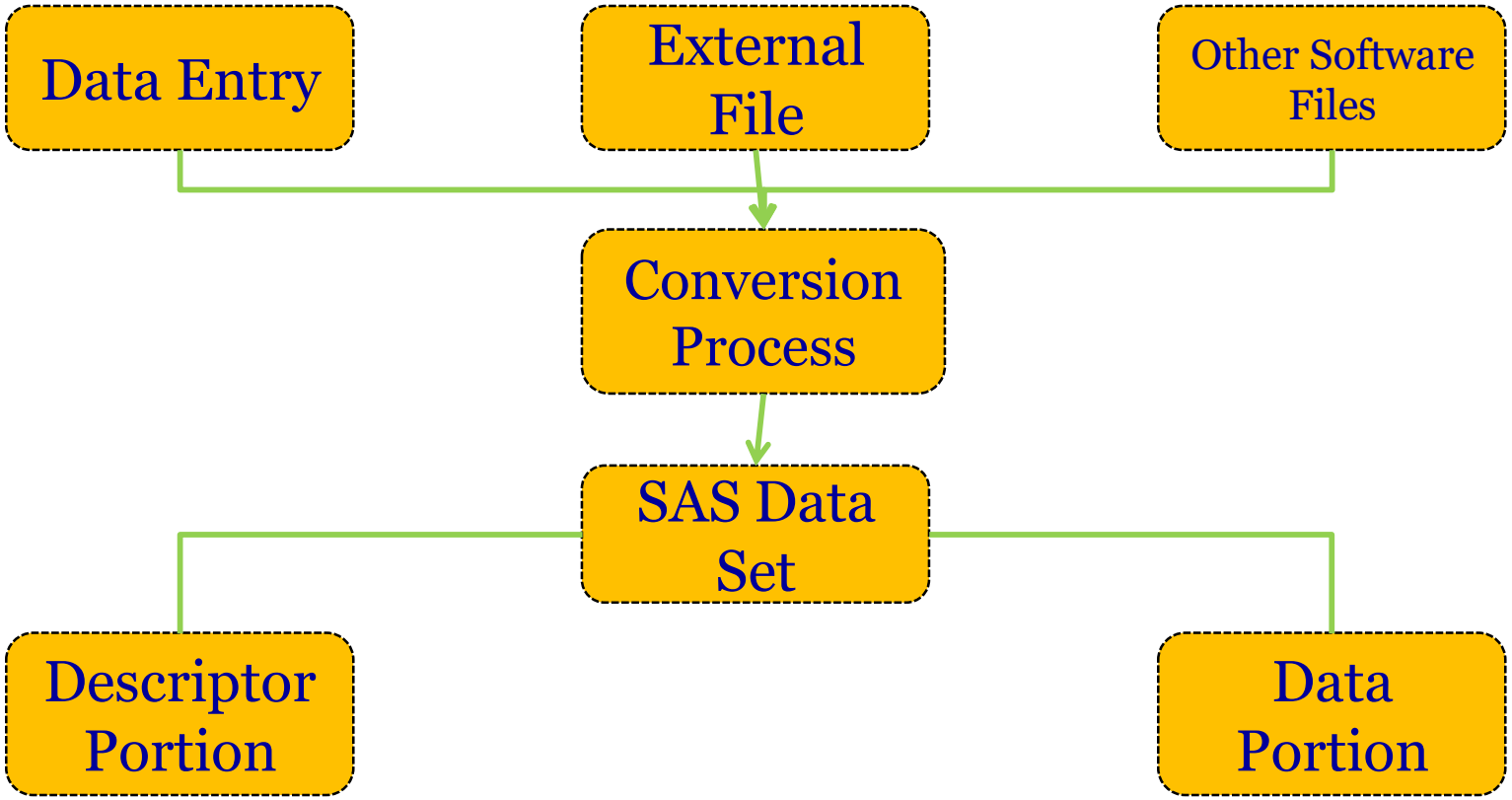
JobCode	EmpID	LastName	FirstName	Salary
FLTAT	0040	WILLIAMS	ARLENE M.	23666.12
	0071	PERRY	ROBERT A.	21957.71
	0091	SCOTT	HARVEY F.	32278.40
	0106	THACKER	DAVID S.	24161.14
FLTAT				102063.37

JobCode	EmpID	LastName	FirstName	Salary
PILOT	0031	GOLDENBERG	DESIREE	50221.62
	0082	MCGWIER-WATTS	CHRISTINA	96387.39
	0355	BELL	THOMAS B.	59803.16
	0366	GLENN	MARTHA S.	120202.38
PILOT				326614.55
				428677.92

The DATA Step

- Now, let's go back to the data step
- What is one of the most important things we need to do in this step?

SAS Data Sets



SAS Data Sets

Descriptor Portion	General data set information * data set name * data set label * date/time created * storage information * number of observations Information for each variable * Name * Type * Length * Position * Format * Informat * Label
Data Portion	Observations for each variable Name Age Height Weight John 19 69 180 Mary 22 63 130 John 21 67 165

Create a SAS Data Set

Data Source	KEYWORD/Method
Data in a permanent SAS dataset or created by another SAS program	SET
Data created in another software program (e.g., Excel)	Import (interactive)
Data Entry	Table Editor
Raw Data in a SAS Program	DATALINES
Raw Data in an external text file	INFILE

Use an Existing SAS Data Set

- To create a SAS data set from an existing SAS data set, you need to
  - Use a DATA statement to name the output SAS data set
  - Use the SET statement (in the DATA step) to read the existing SAS data set
  - Example

```
data work.temp_empdata;
    set data1.empdata;
run;
```

  - SET can refer to an existing SAS data set, temporary, or permanent

Name a SAS Data Set

- General form of the DATA statement
  - This DATA statement creates a temporary SAS data set

```
data work.dfwlax;
   set data1.dfwlax;
run;
```

  - This DATA statement creates a permanent SAS data set

```
libname home '/home/user';

data home.dfwlax;
   set data1.dfwlax;
run;
```

SAS 9.4: Import Excel File

- If you are reading in an Excel file:
  - Use the IMPORT option in the File menu to read it in and create a new SAS data set
  - You will need to tell SAS
    - What format the data are in
    - Where the file is stored
    - What character is used to separate data values (i.e. the “delimiter”)
    - Save your SAS data set in an existing library or create a new library.

SAS Studio: Import Excel File

- If you are reading in an Excel file:
  - Upload the Excel file to your folder.
  - Double click on the Excel file.

  - Run the code. By default, the output SAS dataset is work.import.

SAS 9.4: Table Editor

- Create a SAS data set using the Table Editor
  - Go to Menu → Tools → Table Editor
  - Enter data values into the table
  - Set the name, type, and width for each variable/column using Data Menu – Column Attributes (or right-click on each column heading)
  - Save the data set into a library (you can create a library at this time, if desired)
  - You can now use this SAS data set in your program

Use Raw Data

- To create a SAS data set from raw data, you need to
  - Use a DATA statement to name the output SAS data set
  - Use an INFILE or DATALINES statement to identify where to find the raw data
  - Use an INPUT statement to identify the variables in the raw data file

Raw Data Within a SAS Program

- If the raw data is contained in the SAS program
  - Use the DATALINES keyword, followed by the raw data lines

- Example:

```
data work.sample;
   input firstname $ gender $ age;
   datalines;
   John Male 22
   Jane Female 19
   ;
run;
```

Remember that this method works well for small amounts of data

Point to a Raw Data File

- If the raw data is contained in a separate text file
  - Use the INFILE statement, followed by a file specification.

- Example

```
data work.sample;
    infile '/home/user/sample.txt';
    input name $ gender $ age;
run;
```

  - This method is much more common than DATALINES
  - The file specification **does not use a library reference**
  - The INFILE statement comes before the input statement.

Datalines vs. Infile

- The DATALINES method
  - Allows you to see your data directly
  - Is good for small amounts of data
  - Is often used to create “test” data sets
- The INFILE method
  - Is more common
  - Is necessary if your data comes from an outside source (i.e. client, download, website, etc.)
  - Is preferred for large data sets
  - Allows you to easily re-run your programs on updated data

Class Exercise 3

- Use the **`admit`** data set located in the data1 folder
- Create the following report:
  - Set firstobs=5 and obs=15
  - Has the title: 'Admittance Report'
  - Is sorted by Sex and Name
  - Groups the patients with a different Sex on each page
  - Produces subtotals and a grand total for the Fee
  - Suppresses the Obs column
  - Displays subtotal and total number of observations

Class Exercise 3 - continued

- How many observations are on each page?
  - Is this what you expected?
- How would we change the program such that the output displays observations 5-15 from the sorted data set?

Class Exercise 4

- Create a data set called `temp_staff` in the work library
  - Read in the data set `staff` in the data1 folder
- Create a listing report that:
  - Sorts the data by WageCategory and then by Name
  - Groups the values of WageCategory on separate pages
  - Uses WageCategory as an identification variable
  - Only displays WageCategory, Name, WageRate, Bonus
  - Has an appropriate title

Class Exercise 4 - continued

- Do not display the date or page numbers
- Replace the column headings with
  - ‘Wage Category*_____’
  - ‘Full Name*_____’
  - ‘Wage Rate*_____’
  - ‘Bonus Amount*_____’;
- Split the labels on *.

- Aside: do we need to specifically select WageCategory for display?