

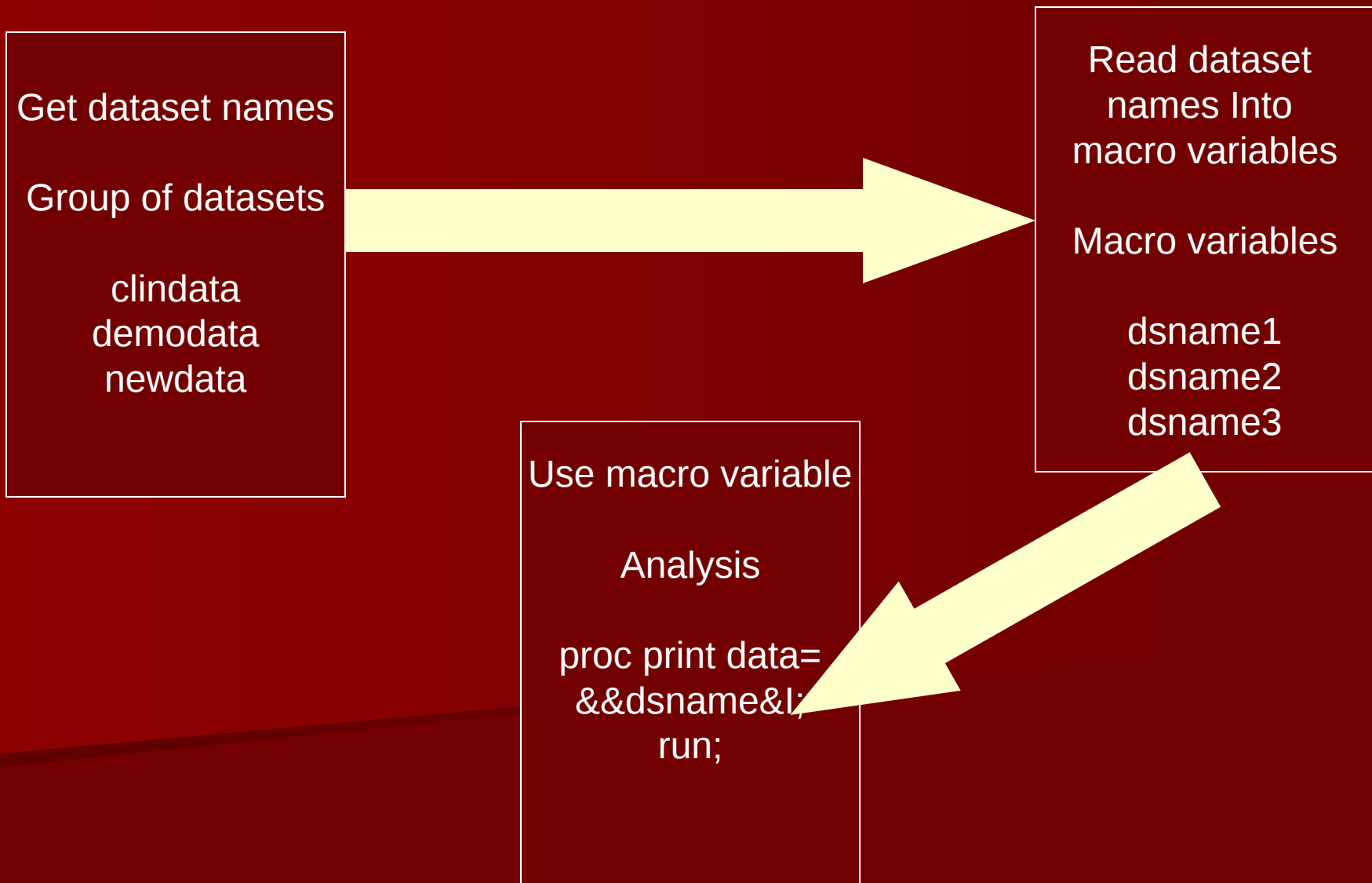
How to Run SAS Procedures on a Group of Datasets

Diana Ventura
Center for Biostatistics in AIDS Research (CBAR)
BASUG 2008, Q1 Meeting
March 28, 2008

Conceptually: How does this work?

- A macro variable is created from the names of SAS datasets in single directory.
- A macro variable that represents the names of datasets allows SAS procedures to be run on a group of datasets without specifying the names of each individual dataset.

Conceptually: How does this work?



Why create a macro variable that represents the datasets?

- Saves time, effort and typing.
- The code in this program is re-usable code.
- A different group of datasets can be analyzed quickly and easily by changing libname for the group of datasets.
- SAS procedures used to analyze data can be adjusted easily.

Get the Dataset Names

- PROC Datasets is used to create a variable containing all the datasets names in a directory.
- PROC SQL can also be used create the variable with the dataset names.
- Since the SQL does not provide a standard ANSI approach to renaming a table, PROC Datasets is the recommended method for renaming.

Get the Dataset Names

- PROC Datasets is used to create a variable containing all the datasets names in a directory.
- The `lib=dsndir` specifies the directory of the group of datasets analysis will be run on.

```
proc datasets memtype=data  
lib=dsndir; contents  
data=dsndir._all_  
out=cout.alldsn; run; quit;
```

Get the Dataset Names

```
proc datasets memtype=data  
lib=dsndir; contents  
data=dsndir._all_  
out=cout.alldsn; run; quit;
```

- The contents DATA=dsndir._ALL_ specification tells PROC Datasets to provide a listing of contents for all datasets in the directory.
- When the _ALL_ keyword is specified you need read access to all read-protected SAS datasets in the directory.

Get the Dataset Names

```
proc datasets memtype=data  
lib=dsndir; contents data=dsndir._all_  
out=cout.alldsn; run; quit;
```

- The contents dataset alldsn is put into a separate directory.
- If the alldsn is put into the directory where the group of analysis datasets is and read into a macro variable, SAS would try to perform analysis on the alldsn dataset as well.
- This would likely produce “ERROR: Variable not found.”

Get the Dataset Names

- The `alldsn` dataset must be a permanent dataset such as `out=cout.alldsn`.
- When a temporary dataset is specified, for example `out=alldsn`, PROC Datasets automatically puts the `alldsn` dataset into the directory of the analysis datasets.

Get the Dataset Names

```
proc sort data=cout.alldsn  
out=values nodupkey; by memname; run;
```

- MEMNAME is the variable created by the PROC Datasets procedure, it contains the names of the all the datasets in the directory.
- PROC Sort is used to unduplicate the variable MEMNAME in the dataset cout.alldsn because Proc Datasets creates a contents by variable name and not MEMNAME.

Read the Dataset Names into Macro Variables

- The MEMNAME variable from the values dataset is used to create two macro variables.
 - count is the total number of datasets in the directory.
 - dsname1-dsnameX contains the names of each of the datasets.

```
data _null_;  
set values end=last;  
call symput('dsname' || left(_n_), trim(memname));  
if last then call symput('count', _n_);  
run;
```

Read the Dataset Names into Macro Variables

```
call symput('dsname' || left(_n_),  
trim(memname)); run;
```

SYMBOL TABLE	
Macro Variable	Actual Name
dsname1	clindata
dsname2	demodata
dsname3	newdata

Use Macro Variables for Analysis

```
%do I=1 %to &count;  
proc print data=dsndir.&&dsname&I;  
var id startdt; run;  
%end;
```

- %do I=1 %to &count; Resolves to I=1 to 3
- proc print data=dsndir.&&dsname&I;
Resolves to PROC Print data=dsndir.clindata;
PROC Print data=dsndir.demodata;
PROC Print data=dsndir.newdata;

*** PROC DATASETS Method of capturing MEMNAME;**

```
libname dsndir '/home/mydir';
```

```
libname cout '/home/cout';
```

*** Get dataset name;**

```
proc datasets memtype=data lib=dsndir;
```

```
    contents data=dsndir._all_ out=cout.alldsn; run; quit;
```

```
proc sort data=cout.alldsn out=values nodupkey;
```

```
by memname; run;
```

*** Put dataset name into macro variables ;**

```
%macro group;
```

```
%local i;
```

```
data _null_;
```

```
set values end=last;
```

```
if last then call symput('count',_n_);
```

```
call symput('dsname' || left(_n_),trim(memname)); run;
```

*** Use macro variable for analysis;**

```
%do I=1 %to &count;
```

```
proc print data=dsndir.&&dsname&I; var id startdt; run;
```

```
%end;
```

```
%mend group;
```

```
%group;
```

Got Questions?

- You may reach me at
dventura@sdac.harvard.edu