Introduction to SAS Procedures

Shannon Pileggi

STAT 330

OUTLINE

Programming Basics

PROC CONTENTS

PROC PRINT

PROC MEANS

PROC IMPORT

Programming Tips

1. Create a 'game plan' before you start programming
   - this can be a mental list OR a plain English description of your plan (or pictures or diagrams - whatever you find most helpful)
   - this step is not required, but can save you hours of wasted time!
   - if you get stuck and ask for help, I will most likely ask you about your 'game plan'
2. Don't try to program the entire thing at once
   - work on one part at a time
   - if you spend more than an hour on something and you are still unable to figure it out, you should ask for help

SAS programs

A SAS program is executed

line by line

and

observation by observation.

Which of the following pseudo-code chunks would successfully create Z as the product of X and Y?

Program 1

```
z =  x * y ;
x = 5 ;
y = 3 ;
```

Program 1

Program 2

```
x = 5 ;
y = 3 ;
z =  x * y ;
```

Program 2

Errors and debugging

Common Errors:

- Statements that are out of order
- Misspelling a variable name or SAS key word
- Forgetting a semi-colon
- Not closing a comment or quote
- Not highlighting an entire DATA or PROC step and submitting it

Debugging methods:

- Check your log
- Use comments to stepwise hide/reveal your code until you identify the error
- Submit a single quote,or semi-colon, or `run;` by itself
- Save your code, **exit** SAS, and re-launch SAS to start from a clean slate.

Programming Basics

PROC CONTENTS

PROC PRINT

PROC MEANS

PROC IMPORT

PROC CONTENTS

SAS Code

```
PROC CONTENTS DATA = sashelp.cars;
RUN;
```

SAS Code

- `PROC CONTENTS` displays information about the SAS data set
  - data set name, number of observations, number of variables, date created
  - for each variable: type, length, formats, and informats
- By default, variables are listed in alphabetical order
- When working with a new data set, always use `PROC CONTENTS` first!

Labels and formats

Labels

- Variable names are typically short, but sometimes not informative
- Use **labels** to replace variables names in output
- Labels can be up to 256 characters long

Formats

- Variable *values* may not always be displayed in an easy to read manner (eg, SAS dates, money)
- **Formats** to change the display of variable values in output

Working with PROCS

All PROCS have *required* and *optional* statements.

SAS Code

```
PROC CONTENTS DATA = sashelp.cars VARNUM;
RUN;
```

SAS Code

Here, VARNUM is actually an *optional* statement. To see all optional statements, examine the help file for that procedure.

On your own: Examine the help file for PROC CONTENTS to determine what VARNUM does.

Titles

SAS Code

```
title "Contents of cars data, alphabetical variables";
PROC CONTENTS DATA=sashelp.cars;
RUN;
title;
```

SAS Code

Title statements

- can be *inside* or *outside* the PROC.
- will **carry through** for all future output.
- can be turned off by submitting `title;`.

Programming Basics

PROC CONTENTS

PROC PRINT

PROC MEANS

PROC IMPORT

PROC PRINT

SAS Code

```
PROC PRINT DATA = sashelp.cars;
RUN;
```

SAS Code

- PROC PRINT displays values of the SAS data set
- By default, all observations and variables are printed

PROC PRINT, optional statements

SAS Code

```
PROC PRINT DATA = sashelp.cars (OBS=10);
        VAR msrp enginesize;
RUN;
```

SAS Code

On your own: Predict what this will do before you submit it.

PROC PRINT, optional statements

SAS Code

```
PROC PRINT DATA = sashelp.cars (OBS=10) _______;
        VAR make model msrp enginesize;
RUN;
```

SAS Code

On your own: Use the help file for PROC PRINT to replace the blank with an option to display the variable label rather than the variable name as shown in the partial output.

Obs	Make	Model	MSRP	Engine Size (L)
1	Acura	MDX	$36,945	3.5
2	Acura	RSX Type S 2dr	$23,820	2.0
3	Acura	TSX 4dr	$26,990	2.4
4	Acura	TL 4dr	$33,195	3.2

More options in PROC PRINT

- NOOBS - suppresses observation numbers
- LABEL or L - Uses labels instead of variable names
- BY var_1 var_2 ... var_k
  - will start a new section for the different values of the specified variable(s); data *must* already be sorted according to the BY variables (see PROC SORT).
- ID var_1 var_2 ... var_k
  - Observation numbers are omitted, ID variable value is included instead
- VAR var_1 var_2 ... var_k
  - specifies which variables to print
- SUM var_1 var_2 ... var_k
  - prints sums for specified numeric variables

Programming Basics

PROC CONTENTS

PROC PRINT

PROC MEANS

PROC IMPORT

PROC MEANS

SAS Code

```
PROC MEANS DATA = sashelp.cars;
RUN;
```

SAS Code

- PROC MEANS produces basic summary statistics
- By default, *N*, mean, sd, min, and max are calculated for all *numeric* variables

PROC MEANS, optional statements

SAS Code

```
PROC MEANS DATA = sashelp.cars ;
        VAR msrp enginesize;
RUN;
```

SAS Code

On your own: Predict what this will do before you submit it.

PROC MEANS, optional statements

SAS Code

```
PROC MEANS DATA = sashelp.cars ;
        VAR make model msrp enginesize;
RUN;
```

SAS Code

On your own: Predict what this will do before you submit it.

PROC MEANS, optional statements

SAS Code

```
PROC MEANS DATA = sashelp.cars (OBS=10) _______;
        VAR msrp enginesize;
RUN;
```

SAS Code

On your own: Use the help file for PROC MEANS to replace the blank with *options* to display only the mean rounded to two decimal places as shown.

Variable	Label	Mean
MSRP		32774.86
EngineSize	Engine Size (L)	3.20

More options in PROC MEANS

- the CLASS statement defines subgroups for analysis
  - think results separated by a categorical variable, but condensed on a single page
- the BY produces separate results by subgroups for analysis
  - think results separated by a categorical variable, but separated in multiple tables
  - requires data to be sorted prior to execution

Programming Basics
PROC CONTENTS
PROC PRINT
PROC MEANS
PROC IMPORT

PROC IMPORT

SAS Code

```
PROC IMPORT
   DATAFILE = "Computer Location/mydata.ext"
   OUT = DataSetName
   DBMS = identifier
   REPLACE ;
RUN;
```

SAS Code

`DATAFILE=` takes computer location, data file name, and extension of the data file

`OUT=` specifies the SAS data set name you want to create

`DBMS=` specifies type of data (e.g., `CSV`, `TAB`, `DLM`)

`REPLACE` option overwrites an existing SAS data set called *DataSetName*

The American Community Survey (ACS)

- Each year since 2005, the U.S. Census Bureau has surveyed about 3.5 million households with the American Community Survey (ACS)
  http://www.census.gov/programs-surveys/acs/
- The data are used in government and policy decisions, and helps to determine the allocation of more than **$400 billion** in federal and state funds each year.
- In the spring of 2012 the House of Representatives voted to eliminate the survey. Daniel Webster (R):
  - *This is a program that intrudes on peoples lives, just like the Environmental Protection Agency or the bank regulators.*
  - *Were spending $70 per person to fill this out. Thats just not cost effective.*
  - *In the end, this is not a scientific survey. Its a random survey.*
- Senator Rand Paul sponsored a bill to make participation in the ACS voluntary.

The data

first few rows of acs.csv

```
Sex,Age,MarStat,Income,HoursWk,Race,USCitizen,HealthInsurance,Language
female,31,not married,60,40,white,citizen,yes,other
male,31,not married,0.36,12,black,citizen,yes,native English
male,75,not married,0,,white,citizen,yes,native English
female,80,not married,0,,white,citizen,yes,native English
```

first few rows of acs.csv

On your own: What features do you see in the data that might need to be addressed to import it?

Note: A data file **cannot** be open in Excel when you try to import it into SAS.

SAS Code

```
PROC IMPORT 1
   DATAFILE="X:/StatStudio/spileggi/Data Sets/acs.csv" 2
   OUT = acs 3
   DBMS = CSV 4
   REPLACE 5
RUN;
```

SAS Code

On your own:

1. Which numbered locations require a semi-colon?
2. Why is the `REPLACE` option useful?
3. Did we need to address any of the features we discussed?
4. Briefly compare the data in SAS to the data in the CSV file. What differences do you notice?

GUESSINGROWS

- `GUESSINGROWS` specifies the number of rows to scan to determine the appropriate *type* and *length* for variables
- the default value is 20
- On your own: What value of `GUESSINGROWS` do you recommend to correctly import the data?

SAS Code

```
PROC IMPORT
   DATAFILE="X:/StatStudio/spileggi/Data Sets/acs.csv"
   OUT = acs
   DBMS = CSV
   REPLACE ;
   GUESSINGROWS=?;
RUN;
```

SAS Code

The Import Wizard

- `File -> Import Data`
- `Select data source -> Next`
- `Browse ->` for file location `-> Next`
- `Library -> WORK, Member:` *DataSetName* `-> Next`
- `Browse ->` create `.sas` file that saves `PROC IMPORT` code
- `-> Finish`

Code generated by the import wizard

SAS Code

```
PROC IMPORT
   OUT= WORK.ACS
   DATAFILE= "X:/StatStudio/spileggi/Data Sets/acs.csv"
   DBMS=CSV REPLACE;
   GETNAMES=YES;
   DATAROW=2;
RUN;
```

SAS Code

On your own: What would you need to change about this code in order to correctly import the data?