SAS Overview Notes (Chapter 1)

It is probably more instructive to see the interface and language by example, but we will summarize a few main points for future reference.

The Interface

- Editor Window - where you write the code
- Log Window – where notes and errors from statements run are generated
- Results Window – this allows for navigating results from output
- Explorer Window – is like a Windows Explorer browser, so you can navigate through files (data files, program files, etc.)
- Output Window – where output is generated if the **ods** is set to **listing** (with the default **ods** settings in 9.3, this window will not be used)

The Language

Types of statements:

- **Data** step – define the data set (could be from raw data, a data file, or an existing SAS data set)
- **Proc** step – says what procedure to use
- Localized statements and options – set within **data** or **proc** steps
- Global statements and options – applied to all statements that follow

Steps of program (roughly speaking…):

- Start with a **data** or **proc** step
- End at the next **data**, **proc** or **run** statement
- Give names to data sets and variables in the data set (variable names are not case sensitive)
- Submit the program via **Run** menu or tool bar icons to run the program and generate results

The Output Delivery System (ODS)

Within **proc** statements we will generally want to include **ods exclude** or **ods select** statements to pick the particular tables and graphics we want to include in our reports. These are *local* statements within **proc** steps.

Notable Changes from SAS 9.2 to SAS 9.3

The required text and the recommended texts were written for SAS 9.2 or earlier. There are a couple of changes it is particularly important for us to note:

- The default for the Output Delivery System is to generate results in **html**.
- Graphics are turned on by default

ODS (Output Delivery System) in version 9.3

When you first start SAS 9.3 you will see a dialog that mentions changes to the output in version 9.3. It will also show you where you can change the settings to get the old output behavior. Prior to 9.3, output is generated in **listing** format (plain text) and printed to the **Output Window**.

In 9.3 and later, the default is **html** with **graphics** turned on and **listing** turned off (so the **Output Window** will be blank unless we turn the listing output on via the **Tools>Options>Preferences** menu item (the check boxes are in **Results** tab).

Note that we can turn the various output types (e.g. **listing**, **html**, **rtf, pdf**) **on** and **off** using **ods** statements. The ones that are turned on will generate results. The ones that are turned off won't.

Setting the type of output and turning graphics on or off can be done through *global* **ods** statements. Throughout the textbook and the program files that come with it, you will see the following statement to turn graphics on:

```
ods graphics on;
```

With SAS 9.3, we will not need this statement because the graphics are turned on by default.

The Data Step

The **data** step is how we define a data set in SAS. It will include:

- a name for the data set
- names and types for the input variables
- a data source
- and any additional processing code

The data source can be raw data entered manually, a data file, or an existing SAS data set.

We will see these types of data set creations via the following examples (roughly based on those in the text):

- **bodyfat** data set defined manually
- **slimmingclub** data set read in as **list** input
- **slimmingclub2** data set read in as **column** input
- **slimmingclub2** data set read in using **informats**
- addition of a variable to the **slimmingclub** data set
- creating a data set from an existing data set by selecting the women from the **bodyfat** data set

We will also see a very basic use of a procedure when we use **proc print** to see what is in these data sets.

The Proc Step

Proc stands for procedure. The procedures perform the analyses on our data. This could be some form of descriptive analysis, model fitting, visualization, etc. The simplest syntax (actually, we don't always need the data setting, but it's usually there…) is the following:

```
proc procedurename data=datasetname;
run;
```

In the **proc print** examples so far, **print** was ***procedurename***, ***datasetname*** was the actual name of the SAS data set being printed.

We can add additional statements which might specify things like the variables to use, the model being fitted, or a condition on the observations to use within the procedure as follows:

```
proc procedurename data=datasetname;
  statementname  <variables>;
run;
```

Here ***statementname*** would be a particular statement name (e.g. **var**, **where**, **by**, **model**,…) and ***variables*** would be the variables or other specification required for that statement. Available statements will vary depending on procedure.

Note: The **<>** are not part of the syntax. They are used to indicate that you might not need to give the values. This is a notation you will see throughout the SAS documentation.

We can add additional statements, by just adding additional statements:

```
proc procedurename data=datasetname;
  statementname1  <variables1>;
  statementname2  <variables2>;
 …
run;
```

We can specify options to the various statements as follows:

```
proc procedurename data=datasetname <procoptions>;
  statementname1  <variables1> </ s1options>;
  statementname2  <variables2> </ s2options>;
 …
run;
```

Note that the **/** is not used for options specified for the **proc** statement, but is for other statements. Technically, **data=** is an option to the **proc** statement, but it has been included throughout the example code here because in most examples that we work with, we will include the **data=** option.

Options can specify types of analyses to perform, methods to use, formatting for graphics, and so on.

A Few Common Statements

The text mentions a few fairly common statements that we will see in many procedures. We will note them here for reference.

- **var** – defines the variables to be used in the procedure
- **class** – states which variables are classification variables (e.g. nominal categorical variables like gender or group name which do not have a meaningful ordering) rather than continuous variables
- **where** – used for stating conditions, so we can get or operate on subsets of the data
- **by** – used to perform an operation based on a variable (e.g. sort by a variable in the data set, or perform an analysis for each value of a classification variable separately)

Note that when performing a procedure by a classification variable, we will usually need to sort by that variable first so data entries are already grouped by the classification variable.

ODS Statements

We will make use of **ods** statements both globally and locally (within procedures).

Some Global Usages

We can modify the way results are generated in general. By default results are generated in html and new results are appended so you can wind up with a lot of output to scroll through. The general issue and solution is described here:

http://support.sas.com/kb/43/911.html

If we run the following three lines of code:

```
ods html close;
ods preferences;
ods html newfile=proc;
```

this will tell SAS to generate a new html document each time code is run, so only the results of the last code selection run will display in the viewer window. You can still click on previous results in the Results window to see previous results. A downside of this setting is that lots of html files are generated.

We can change the output type by using the **ods** command to turn off the types we don't want and turn on the types we do want. To turn an output type on, we can use a statement of the form

ods *type*;

where ***type*** is a setting such as **html**, **listing**, or **rtf**.

To turn off that type of output we can use

ods *type* close;

We also need to use a close statement if we want to edit some files in other applications (e.g. editing a resulting rtf file in Word).

A common usage for us will be to write code in SAS to generate the results we want for a report, then add

ods rtf file='*filename*';

to the top of the program file, and

ods rtf close;

to the end of the code, then rerun the file to generate an rtf file named ***filename*** that can be edited in Word.

Note: The **close** statement at the end is necessary. The **ods rtf file=…** statement opens up a stream to write to rtf file. As long as that stream is open, SAS is in charge of writing to that file and you won't be able to edit it outside of SAS. The **close** statement closes the stream, and you can then edit the file in some other application (usually Word).

Some Local Usages

SAS procedures generate a bunch of tables and graphics by default. Usually, we don't need all of them, and we will want to just pick out the relevant pieces to include in our reports. To do this, we can add **ods** statements within our procedures. The **ods exclude** statement will not generate the results we specify to it, and the **ods select** statement will only generate the results we specify to it. These statements can be used globally, but often it is more convenient and safer to use them locally.

Results generated are tables or graphics. We can see the tables and graphics available for a procedure by looking at the documentation for that procedure. The names and descriptions can be found at the bottom of the **Details** tab for the procedure.

The Help System

Documentation and learning how to navigate that documentation are crucial to using software. Features you use regularly you will just remember and know how to use. Features you haven't used in a while you may need to look up. When there are lots of procedures with lots of features, you may also start with a small set of features and want to use more for analyses as needed later.

Within the documentation for the individual procedures, there are a number of tabs we will refer to often:

- **Overview** – gives a general overview of the features of the procedure
- **Getting Started** – gives some short examples
- **Syntax** – gives the statements, options, and defaults for a procedure

- **Details** – gives various details about features (including available tables and graphics), results, and theory
- **Examples** – generally gives more advanced examples

In class we will look at one particular example, but we will use the Help often so it is good to get accustomed to navigating it.