1. Introduction to SAS programming

Use of SAS
March 2011

Basic concept of SAS programming

- Write a SAS program
  - A program is like a recipe: a series of instructions to be executed in a specified sequence
- Let SAS execute and interpret your program and do some statistical calculations
- SAS responds by giving results in some format

– We need to know the syntax and rules for the SAS language.

Typical use of SAS for statistical analysis

- You have data in some format
- Create SAS data (day 3)
- Get the data into SAS (libname statement)
- Look at the data using SAS
- Perhaps transform or select part of the data, i.e making the data ready for statistical analysis
- Use the appropriate SAS statistical procedure, called a ”proc”, e.g. `proc logistic` for logistic regression
- Get your results out
- Check that SAS did what you asked for (SAS does some ”logging” of your instructions)
- Interpret the SAS output

Basic structure of the SAS system

- SAS programing
  - data step
  - proc step
- Base SAS (50 different proc's)
  - Data manipulation: so-called "data-step", `proc sort`
  - Showing data: `proc contents`, `proc print`, `proc format`
  - Basic statistics: `proc means`, `proc univariate`, `proc freq`
- Special modules
  - SAS/STAT: `proc ttest`, `proc glm`, `proc logistic`, … (more than 65 proc's)
  - SAS/GRAPH: `proc gplot` (23 proc's)
  - other modules: SAS/ASSIST, QC, ETS, FSP, IML, …

SAS-data sets

Observations × variables :

sex	age	weight	name
1	8	25	John
2	5	17	Anna
2	13	48	Maria
⋮	⋮	⋮	⋮

- Variable names: sex age weight name (OBS: not æøå)
- Variable types: Numeric or Character

SAS Analyst

- Add-on to SAS (“application”)
- Menu/form-oriented
- Writes and runs programs for you
- Similar to SPSS

BUT:

- Does not cover everything
- Tedious to use in the long run
- Analysis will be difficult to reproduce

We do not use this

SAS Display Manager

Framework for program development and data handling

- Editor window: Writing SAS programs
- Log window: ”Logging”, i.e. what happen when and did it work?
- Output window: Results from proc’s
- SAS Explorer: Navigating the SAS system as you have seen
- Results: Structured output, designed for helping the user navigating in large amount of output

SAS-Explorer and SAS libraries

Similar to Windows Explorer, where you can access SAS-data stored in SAS libraries.

- SAS libraries contains:
  - SAS-data sets
  - SAS catalogs (user defined formats, options, setup info, etc.)
- Double-click Libraries in SAS-Explorer. SAS is born with some libraries of which two are important to know about:
  - WORK
  - SASUSER

Getting data into SAS: the libname statement

On your own drive in the library `p:\sas\data\other` you will find the SAS data set `fitness.sas7bdat`.

To get this into SAS, we turn `p:\sas\data\other` into a SAS library. This is done with a libname statement. Go to the Editor window. Type in the fragment below and click on “the running man” in toolbar or press function key F3.

```
libname new 'p:\sas\data\other';
```

We give the library `p:\sas\data\other` the nickname ’new’. The name ’new’ is something you decide. BUT you have to give the precise path of the library.

From now on we can refer to files in library using the name ’new’.

Now use SAS-Explorer to find the fitness data.

Viewtable

The data set is opened in a window called VIEWTABLE

- Note entries under **View**: Forms/Table mode and Column Names/Labels.
- Data can be in "Browse" mode or "Edit" mode. In Browse-mode, you can only navigate the data set, not change it, add new variables, etc.
- Default is Browse mode
- You switch modes with **Edit** ⟶ **Mode** ⟶ **Edit**
- We recommend NEVER to change data in viewtable. It is much better to write a program doing the changes.
- NB: Do not try to run programs that modify a data set which is open in Viewtable: Now, close Viewtable using a single-click in upper right corner.

Exercise: Reading in some ”nice” data

In this exercise you will be working with a data file in SAS format called `bissau.sas7bdat`. Data comes from rural Guinea-Bissau, West-Africa: 5273 children visited when being less than 7 months of age and followed for approximately six months. Registration of vaccination status, weight, etc at visit and deaths registered during follow-up.

- Using WINDOWS explorer, find the SAS-data set `bissau.sas7bdat` on your personal drive in the directory `p:\sas\data\africa`.
- Link this directory to a SAS library called `afrika` using a `libname` statement.
- Take a look at the data using VIEWTABLE.

More about the Editor window

- Here you write programs and ”submit” them to SAS
- Standard free text editing
- SAS is case-INsentitive (SEX, sex, Sex all refer to the same)
- Load and save editor contents (called `.sas` files, e.g. `henrik.sas`)
- Line numbers, Colours, indentation, and separators
- Type in the fragment below and click on “the running man” in toolbar or press function key F3:

```
proc print data=new.fitness;
run;
```

PROC PRINT

```
Obs    age    weight    runtime    rstpulse    runpulse    maxpulse    oxygen    group

  1     57     73.37     12.63        58          174         176      39.407      2
  2     54     79.38     11.17        62          156         165      46.080      2
  3     52     76.32      9.63        48          164         166      45.441      2
  4     50     70.87      8.92        48          146         155      54.625      2
  5     51     67.25     11.08        48          172         172      45.118      2
  6     54     91.63     12.88        44          168         172      39.203      2
  7     51     73.71     10.47        59          186         188      45.790      2
  8     57     59.08      9.93        49          148         155      50.545      2
  9     49     76.32      9.40        56          186         188      48.673      2
 10     48     61.24     11.50        52          170         176      47.920      2
 11     52     82.78     10.50        53          170         172      47.467      2
 12     44     73.03     10.13        45          168         168      50.541      1
 13     45     87.66     14.03        56          186         192      37.388      1
 14     45     66.45     11.12        51          176         176      44.754      1
 15     47     79.15     10.60        47          162         164      47.273      1

      .                                                                      .
      .                                                                      .
      .                                                                      .

 26     38     89.02      9.22        55          178         180      49.874      0
 27     47     77.45     11.63        58          176         176      44.811      0
 28     40     75.98     11.95        70          176         180      45.681      0
 29     43     81.19     10.85        64          162         170      49.091      0
 30     44     81.42     13.08        63          174         176      39.442      0
 31     38     81.87      8.63        48          170         186      60.055      0
```

Now try the procedure CONTENTS which will display the so-called header for the data set:

```
proc contents data=new.fitness;
run;
```

PROC CONTENTS

```
                                 The CONTENTS Procedure

        Data Set Name          NEW.FITNESS                          Observations          31
        Member Type            DATA                                 Variables             8
        Engine                 V9                                   Indexes               0
        Created                12. januar 2006 torsdag 23:28:07     Observation Length    64
        Last Modified          12. januar 2006 torsdag 23:28:07     Deleted Observations  0
        Protection                                                  Compressed            NO
        Data Set Type                                               Sorted                NO
        Label                  Exercise/fitness study data set
        Data Representation    WINDOWS_32
        Encoding               wlatin1  Western (Windows)

                            Alphabetic List of Variables and Attributes

                     #    Variable    Type    Len    Label

                     1    age         Num       8    Age in years
                     8    group       Num       8    Experimental group
                     6    maxpulse    Num       8    Maximum heart rate
                     7    oxygen      Num       8    Oxygen consumption
                     4    rstpulse    Num       8    Heart rate while resting
                     5    runpulse    Num       8    Heart rate while running
                     3    runtime     Num       8    Min. to run 1.5 miles
                     2    weight      Num       8    Weight in kg
```

Log window

In the Editor, add the option `position`, but you should misspell it as `poTition`

```
proc contents data=new.fitness poTition;
run;
```

Well, something happend, inspite of the error. Try to look in the log window: Press F6 or use your mouse. SAS does some guessing - very helpful. The option `position` will add a list where the variables are listed in order of their ”position” in the data set. Now try to write `politi` instead. Now we get an error - look in the log.

INFORMATION IN THE LOG WINDOW IS IMPORTANT TO CONSULT EVEN IF YOU BELIEVE EVERYTHING IS OKAY...

Results window

Results window got folders added

Print: The SAS system

Contents: The SAS system

Double-click the nodes inside these to see its contents

More about SAS libraries

- Structure containing datasets (and SAS catalogs)
- Actually just a folder on the computer disk
- See Properties in the Explorer for precise location
- The library WORK is a *temporary* library, i.e., it disappears (with everything inside it) when SAS is shut down. Used for intermediate results.
- SASUSER is permanent. Datasets here are still available the next time you open SAS.
- More than one library can refer to the same folder on the disk.

More about SAS-data sets

- Specified like `new.fitness`
- First part (before the dot) is the SAS library
- Second part is the name of the data set
- If first part is omitted, WORK is assumed
- Seen from Windows, data are in files with extension `sas7bdat`
- data sets have two logical parts, the header (a descriptive part) and the actual data
  - PROC CONTENTS respectively PROC PRINT

More about the DATA step

Submit the following data-step

```
data fitness;
  set new.fitness;
run;
```

The first line: creates a new data set. It is called fitness and is placed in the WORK library. So far it is empty.

The second line: copies the fitness data from the new-library into data set in WORK.

Go to the WORK library and find this version of the fitness data.

Saving a SAS program, Output window, Log window

To save what you have been producing in the Editor, you act as in e.g. Word:

File ⟶ **Save as...**

You save the Output and Log window in the same manner.

File extensions are:

`*.sas` Program files

`*.log` Log files

`*.lst` Output files

Keys

Shortcut keys for navigating the SAS system, press F9 to see. Important ones are:

- F5: Editor
- F6: Log
- F7: Output
- F3: ”Submit”, i.e. run the SAS program in the Editor

Close down the Keys window using either your mouse or the standard Windows ”Ctrl+F4”. Press F5 to go to the Editor.

NB: If you by accident closed the Editor you can get another one using the menu **View** ⟶ **Enhanced Editor** (the View menu also have a ”program editor” which is an older less featured version of the editor).

Advantages/disadvantages in SAS

+ Handles large amounts of data

+ Flexible data handling

+ Cover many (many) statistical models/methods

+ Many platforms

+ De facto standard, also large user-groups

– A bit old-fashioned

– Difficult to learn

– Unflexible in advanced programming

– Hard to make good-looking graphics (?)

The help system

- Via the small book icon on the toolbar or F1 – but does not work here
- Usually, you'll need to use the index
- You generally need to know roughly what you are looking for. It is not a textbook.
- SAS OnlineDoc which contains material that used to be found in large manuals (you can still buy those):

`http://support.sas.com/onlinedoc/913/docMainpage.jsp`

- "The Little SAS Book"

`http://support.sas.com/publishing/bbu/companion_site/56649.html`

How to get organized

- Interactive runs are convenient, but also dangerous
- Remember to save and keep program code.
- Collect the fragments into coherent `.sas` files that can be run from scratch.
- Test your programs in a freshly started SAS session.

Exercise: Bissau data

- How many observations and variables are in the data set? Use e.g. `proc contents`.
- Make a copy of data `afrika.bissau` into `work.bissau`. Use the 'set'-statement.
- Make a SAS program that makes a `proc print` of the Bissau data. Save the program somewhere on your personal drive.
- Restart SAS (closing and starting SAS).
- Open your SAS program and submit it. Did it work? If not, why not?

SAS programming

- A program is a “recipe”: A series of instructions to be executed in a specified sequence
- Notice: SAS is not a spreadsheet. Output is output, and does not change automatically if data are changed
- Some rules and conventions are necessary for SAS to be able to interpret its instructions
- Separators (semicolon, /)
- Parentheses and quote symbols must be matched

A simple SAS program

```
data new;                                  |
  set original;                            |
  age=1997-birthyr;                        |  Data Step
  bmi=weight/(height*height);              |
run;                                       |



proc print data=new;                       |
  var id age bmi;                          |
run;                                       |
                                           |  Proc Steps
proc means data=new;                       |
  var age bmi;                             |
run;                                       |
```

DATA steps and PROC steps

- Roughly speaking, SAS programs consist of two kinds of *steps* (= blocks of instructions):
- DATA steps define datasets. E.g. by reading raw data, computing transformed variables, selecting cases, etc.
- PROC steps contain standard procedures that operate *on* datasets. You can't, e.g., transform variables in a PROC step.
- Normal arrangement of a SAS program is to put DATA steps at the beginning, but they can occur intermixed
- There are a a few SAS *statements* in addition to the DATA and PROC steps. They typically set up definitions for later use: LIBNAME, OPTIONS, AXIS, and SYMBOL statements are the most common ones

Basic things about the SAS language

- Almost everything starts with a keyword and ends with semicolon (exception being that there is no keyword before computations in a DATA step)
- **Statements** are pieces of code separated by semicolon

```
OPTIONS ls=80;
PROC GPLOT data=sasuser.fitness;
   PLOT maxpulse * age;
RUN;
PROC GLM data=sasuser.fitness;
   MODEL maxpulse = age / solution;
RUN;QUIT;
```

- Keywords: OPTIONS PROC PLOT MODEL RUN QUIT
- Some statements belong together in blocks (**steps**).

Things to notice

```
OPTIONS ls=80;
PROC GPLOT data=sasuser.fitness;
   PLOT maxpulse * age;
RUN;
PROC GLM data=sasuser.fitness;
   MODEL maxpulse = age / solution;
RUN;QUIT;
```

- The slash symbol (/) is often used to introduce options for a statement
- Separators like semicolons and slashes are necessary to avoid ambiguity: `solution` is not a variable name, `run` is not an option.
- SAS detects the end of a step when there is a new DATA or PROC statement (RUN is not always needed).

Formatting of code

- SAS generally doesn't care about whitespace and line breaks

```
data
        work.cohort;
        set course.males98;
run;
```

- is the same as

```
data work.cohort; set course.males98; run;
```

- Good practice is to have at most one statement per line.

Indentation

- Enhances readability considerably. (You *will* have to read your own old code!)
- DATA and PROC steps are entered starting at the left edge. Likewise `OPTIONS` statements and `RUN` and `QUIT`.
- Any subordinate statements are indented by 2-4 blanks
- In statements which do not fit on one line, subsequent lines are also indented.
- This creates visual groups, so that you can easily see where one thing ends and the next begins.

Example of good indentation

```
data new;
  set original;
  age=1997-birthyr;
  bmi=weight/(height*height);
run;



proc print data=new;
  var id age bmi;
run;

proc means data=new;
  var age bmi;
run;
```

Ingredients of a DATA step

```
data new;
  set original;
  age=1997-birthyr;
  bmi=weight/(height*height);
run;
```

- Specification line (name of new data set)
- Data source (here: name of old SAS data set)
- Computation
- Assignment

Variables

- Columns of a dataset
- Can be numerical (usual case)
- – or character (text strings)
- Values of a character variable are given in quotes: `’male’` or `"male"`
- A dot (.) denotes a missing value for a numerical variable and is the lowest number in SAS.
- Calculations involving a missing will result in a missing (most of the times)
- A ”” or ” denotes a missing value for a character variable.

Names of variables

- SAS is case-insensitive (`SEX`, `sex`, `Sex` all refer to the same variable)
- Names can be up to 32 characters long (older SAS: max 8)
- Names can consist of (english) letters, numbers and underscore (`_`)
  - but can *not* start with a number

Comments

Two ways of making comments in SAS programs:

```
/* Comments */

 * Comments ;
```

Example:

```
/* Here I make a new data
with new variables age and bmi*/
data new;
  set original;
  age=1997-birthyr;
  bmi=weight/(height*height);
run;

*Here I print age and bmi;
proc print data=new;
```

```
  var id age bmi;
run;

*Here I calculate means;
proc means data=new;
  var age bmi;
run;
```