UPPSALA UNIVERSITET

Introduction to SAS

WELCOME!

SAS/STAT® Software

Inger Persson

UPPSALA UNIVERSITET

What is SAS?

- Developed in the early 1970s at North Carolina State University
- Originally intended for management and analysis of agricultural field experiments
- Now the most widely used statistical software
- Used to stand for “Statistical Analysis System”, now it is not an acronym for anything
- Pronounced “sass”, not spelled out as three letters.

UPPSALA UNIVERSITET

Overview of SAS Products

- Base SAS - data management and basic procedures
- SAS/STAT - statistical analysis
- SAS/GRAPH - presentation quality graphics
- SAS/OR - Operations research
- SAS/ETS - Econometrics and Time Series Analysis
- SAS/IML - interactive matrix language
- SAS/AF - applications facility (menus and interfaces)
- SAS/QC - quality control

SAS OnDemand for Academics

SAS version 9.4 is used in SAS OnDemand for Academics

SAS online documentation and support

The SAS documentation is available online, at https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.5/pgmsashome/home.htm.

Here, you can find links to the BASE SAS procedures guide, the SAS/STAT users guide, data step statement library (more on these later) and much more.

SAS Customer Support:
https://support.sas.com/en/support-home.html

List of keyboard shortcuts:
https://documentation.sas.com/doc/en/webeditorcdc/3.8/webeditorug/keyboardshortcuts.htm

UPPSALA UNIVERSITET

Basic Structure of SAS

Two main components to most SAS programs - the data step(s) and the procedure step(s).

The **data step** reads data from external sources, manipulates and combines it with other data sets, and prints reports. The data step is used to prepare your data for use by one of the procedures (often called “procs”).

SAS is very lenient about the format of its input - statements can be broken up across lines, multiple statements can appear on a single line, and blank spaces and lines can be added to make the program more readable.

Basic Structure of SAS

The **procedure steps** perform analysis on the data, and produce (often huge amounts of) output.

The most effective strategy for learning SAS is to concentrate on the details of the data step, and learn the details of each procedure as you have a need for them.

UPPSALA UNIVERSITET

Some Preliminary Concepts and Rules

- SAS variable names must be 32 characters or less, constructed of letters, digits and the underscore character.
- Don’t start variable names with an underscore (_), because special system variables are named that way.
- SAS is not case sensitive, except inside of quoted strings.
- Missing values are handled consistently in SAS, and are represented by a period (.).
- Each statement in SAS must end in a semicolon (;).

Structure of SAS programs

- Lines beginning with an asterisk (*) are treated as comments. Alternatively you can enclose comments between /* and */.
- You can combine as many data and proc steps in whatever order you want.
- Data steps begin with the word **data** and procedure steps begin with the word **proc**.
- The run; command signals to SAS that the previous commands can be executed.

Structure of SAS programs

- There are global options (like linesize and pagesize) as well as options specific to datasets and procedures.
- Informative messages are written to the SAS log - make sure you read it!

The Data Step: Basics

Each data step begins with the word **data** and optionally one or more data set names (and associated options) followed by a semicolon. The name(s) given on the data step are the names of data sets which will be created within the data step.

The **input** statement of SAS is used to read data from an external source, or from lines contained in your SAS program.

The **infile** statement names an external .txt-file from which to read the data.

Data Step: input Statement

There are different forms of the input statement, we'll use the simplest version: **List input.**

List input (free form) - data fields must be separated by at least one blank. List the names of the variables, follow the name with a dollar sign ($) for character data.

The ampersand (&) modifier tells SAS to use two whitespace characters to signal the end of a character variable, allowing embedded blanks to be read using list input.

proc import

For certain simple data files (e.g. CSV or Excel files), SAS can create a SAS data set directly using **proc import**.

The dbms= option informs SAS of the type of file to be read, and choices include csv (Comma-separated values), xlsx (Excel spreadsheets), dbf (Dbase files), dta (Stata files), sav (SPSS files), and tab (Tab-separated files).

proc import provides no options for formatting, and may not be successful with all types of data files.

The SET statement

When you wish to process an already created SAS data set, the **set** statement is used.

Each time SAS encounters a set statement, it inputs observation(s) from an existing data set, containing all the variables in the original data set along with any newly created variables.

The IF, THEN, and ELSE statements

The **if** statement can be used to perform an operation only for selected observations, together with the **then** statement.

If you wish to perform one operation for selected observations, and another for other observations, the if and then statements can be followed by the **else** statement.

To select variables in your data set

Sometimes you don’t need to use all of the variables in a data set for further processing.

To restrict the variables in an input data set, the data set option **drop=** or **keep=** can be used with a list of variable names.

To select observations in your data set

Sometimes you don’t want to include all of the observations in a data set for further processing.

To restrict the observations in an input data set, the **where** statement can be used with some criteria to be fulfilled.

The **output** or **delete** statements can also be used.

To merge data sets

To merge data sets, simply use the **merge** statement.

There has to be at least one identification variable, included in all data sets that are to be merged.

The data sets have to be sorted, by the identification variable. This can be done using **proc sort**.

Temporary Data Sets

By default, the data sets you create with SAS are deleted at the end of your SAS session.

During your session, they are stored in a ‘work’ directory (Libraries - My Libraries - WORK).

UPPSALA UNIVERSITET

Permanent Data Sets

You can save your SAS data sets permanently by first specifying a directory/library to use with the **libname** statement, and then using a two level data set name in the data step.

In a later session, you could refer to the data set directly, without having to create it in a data step.

Operators in SAS

Arithmetic operators:

+	Addition	**	Exponentiation
-	Subtraction	sqrt()	Square root
*	Multiplication	log()	Logarithm (base *e*)
/	Division	log10()	Logaritm (base 10)

Comparison operators:

= or *eq*	equal to	^= or *ne*	not equal to
> or *gt*	greater than	>= or *ge*	greater than or equal to
< or *lt*	less than	<= or *le*	less than or equal to

UPPSALA UNIVERSITET

Operators in SAS

Boolean (true or false) operators:

& or *and*	and
| or *or*	or
^ or *not*	negation

Other operators:

><	minimum
<>	maximum

The **in** operator lets you test for equality to any of several constant values.
“x in(1,2,3)” is the same as “x=1 or x=2 or x=3”

SAS procedures

There are a large number of procedures available in SAS. They can all be found and described in the Online Documentation

proc means

The **means** procedure provides data summarization tools to compute descriptive statistics for variables across all observations and within groups of observations, e.g.:

- descriptive statistics based on moments
- quantiles (incl. median)
- confidence limits for the mean
- t-test
- etc…

proc univariate

You can use the **proc univariate** statement to request a variety of statistics for summarizing the data distribution of each analysis variable:

- sample moments
- basic measures of location and variability
- confidence intervals for the mean, standard deviation, and variance
- tests for location
- tests for normality
- frequency counts for observations
- missing values
- etc…

Procedure syntax

Available options can be found in the OnlineDoc

```
PROC UNIVARIATE < options > ;
    BY variables ;
    CLASS variable-1 <(v-options)> < variable-2 <(v-options)> >
          < / KEYLEVEL= value1 | ( value1 value2 ) >;
    FREQ variable ;
    HISTOGRAM < variables > < / options > ;
    ID variables ;
    INSET keyword-list < / options > ;
    OUTPUT < OUT=SAS-data-set >
          < keyword1=names...keywordk=names > < percentile-options >;
    PROBPLOT < variables > < / options > ;
    QQPLOT < variables > < / options > ;
    VAR variables ;
    WEIGHT variable ;
run;
```

or

Additional statements (explained in the OnlineDoc)

Anything within <> can be excluded

Exercise: input statement

Create a data set with the following information:

Patient	1	2	3	4	5	6	7	8	9	10
Day of diagnosis	5	8	12	16	17	19	23	24	30	32
Treatment	A	B	B	B	A	A	B	A	A	B
Treatment start day	9	8	14	17	20	20	24	28	32	33
Day of first effect	16	14	18	21	23	27	30	35	40	37

Five variables (columns) are needed: Patient no, Day of diagnosis, Treatment, Treatment start day, Day of first effect

Check that the data set contains the correct information.

Exercise: infile statement and proc import

Make use of the code in *SAS examples.pdf*
and do the following:

1) Create a dataset containing the data on the previous slide by reading in the file *patients.txt.*

2) Create a dataset containing the data above by importing the file *patients.xlsx*

Exercise: The data step

Use (one of) the data set(s) that you just created.

1) Create a new variable: Time from diagnosis to treatment start.

2) Create a new variable: Time from treatment to effect.

3) Create a new indicator variable, taking the value 1 if Time to effect <7 days and 0 otherwise.

4) Delete the indicator variable you just created.

5) Save the data set permanently.

Exercise: The data step, cont'd

6) Create a new data set, containing only the individuals where time from diagnosis to treatment start is max 3 days and time from treatment to effect is less than 7 days.

7) Create a new data set, containing the individuals where *either* time from diagnosis to treatment start is more than 3 days *or* time from treatment to effect is at least 7 days.

8) Create a new data set, merging the two data sets you created in steps 6) and 7). Remember to select an appropriate variable to merge by.

Exercise: SAS procedures

Use the data set containing the patient data that you just created.

1) Present the number of observations (frequencies), means and standard deviations for time from diagnosis to treatment start and time from treatment to effect, using proc means.

2) Present median and quartiles for the two time variables above (proc univariate).

3) Test if there is any difference between the two groups' average time from treatment start to effect using proc ttest and the CLASS statement (ignore the small sample size).