introduction to sas

Juan Shishido, D-Lab, UC Berkeley

Month Day, 2016

introduction

what is sas

SAS is an integrated system of software solutions

It enables:

- data management
- report generation
- plotting
- statistical and mathematical analyses
- and more

products

- Base SAS
- SAS/STAT
- SAS/ETS
- SAS Text Miner
- SAS Energy Forecasting
- and much, much more

Products & Solutions A-Z

base sas

base sas

Includes:

- a programming language
- a data management facility
- data analysis and reporting utilities

programming language

> *The SAS language contains statements, expressions, functions and CALL routines, options, formats, and informats*

There are two main components:

- data steps
- procedure steps

SAS programs—files ending in the `.sas` file extension—typically include several DATA and PROC steps

programming language

Example of a DATA step

```
data example;
    infile 'path/to/file';
    input x1 x2 x3;
run;
```

programming language

Syntax

One of the most important rules is that **SAS statements must end with a semicolon**

SAS statements can span multiple lines

Multiple SAS statements can appear on the same line, so long as each is separated by a semicolon

A `run;` statement, which creates a "step boundary," marking the end of a step, isn't required between steps in a program, but is recommended

programming language

SAS Names

Are used for data sets, variables, and other items

In general, these names must:

- contain only letters, numbers, or underscores (_)
- begin with a letter or underscore
- have a length betwen one and 32 characters
    - maximum length varies by name type (e.g., variable names versus library references)
- not contain blanks

Names are *not* case sensitive

data representation

In SAS, data is organized into rows and columns in what is called a SAS data set

x1	x2	x3
25	m	berkeley
26	f	san francisco
23	f	oakland
24	m	marin

Each row is sometimes called an “observation” and each column a “variable”

data step

DATA steps begin with the `data` statement and are typically used to create, modify, or replace SAS data sets

Data can either be read inline or from external sources, such as `.txt`, `.csv`, or `.sas7bdat` files

SAS data sets can either be temporary or permanent

Temporary data sets are stored in the `WORK` library and are deleted at the end of each SAS session

Permanent data sets are saved to disk

data step

SAS data sets are temporary, by default

In the code above, `example` is a temporary SAS data set

To read or write a *permanent* SAS data set, use dot notation such as `libref.dataset`

The `libref` is a name associated with a SAS library or directory location

It is possible to use `work.dataset` to be explicit about temporary data sets

data step

To set up a `libref` use the `libname` keyword

```
libname mylib 'path/to/dir';
```

In this example, `mylib` is a variable representing the `path/to/dir` location

Note that `libref` names can only be 8 character long

```
data mylib.example;
    ...
run;
```

In the code above, the data set `example` will be saved to the location associated with `mylib`

proc step

references

links

- http://www.stat.berkeley.edu/~spector/s100/sas.pdf
- http://www.ats.ucla.edu/stat/sas/library/SASRead_os.htm
- http://www2.sas.com/proceedings/sugi31/246-31.pdf
- https://www.ssc.wisc.edu/sscc/pubs/4-18.htm