UCF STA4102 Lecture 1

Alexander V. Mantzaris

UCF | Department of Statistics

UNIVERSITY OF CENTRAL FLORIDA

Overview

1. an interesting read
2. Background on SAS (Statistical Analysis System)
3. Basics

-The apparent contrast between statistics and truth might be inferred from the book title. May I quote a remark I once overheard: 'There are three kinds of lies; the justifiable/innocent ones, common unjustifiable lies, and statistics.'
At the basis of all, lies the conviction that conclusions drawn from statistical considerations are at best uncertain and at worst misleading. I do not deny that a great deal of meaningless and unfounded talk is presented to the public in the name of statistics. But my purpose is to show that, starting from statistical observations and applying them a clear and precise concept of probability it is possible to arrive at conclusions which are just as reliable and 'truthfull' and quite as practically useful as those obtained in any other exact science. In order to achieve this purpose, I must ask you to follow me along a road which is often laborious and by paths which at first sight may appear unnecessarily winding.-
PROBABILITY, STATISTICS AND TRUTH by Richard von Mises
(This was before the age of BIG DATA)

current section

1. an interesting read
2. Background on SAS (Statistical Analysis System)
3. Basics

History

SAS was developed at North Carolina State University from 1966

- why is that important?
- At this stage it is not. Even towards the end of the course. It becomes more important to understand the history of the language when looking at the differences from other systems. The history of a system is often useful for understanding its **conventions** it has adopted. The reason you are forced to do things in one way rather than another, or why certain objects or fields are referred to in a manner which differs from other approaches.
- why mention the dates and not simply the information?
- Good question! Since software was invented to either store data, enhance communication, display data, or provide faster computational means; the different approaches in which 'human to computer instruction' were done influenced each other.

History

The programming language 'C' was developed afterwards in 1969

- That is suprising. 'C' is quite fundamental and well known. I would have expected that statistical software would come much later in history.
- Summary statistics have been done since systems using punch cards were around! (in high school I actually had to learn about punch cards). Statistics has been one of the biggest drivers of computation ever; since the start. They were always complaining about having to work through repetitive mindless tasks. Having to compute means, and variances on paper with the help of a calculator is an era you missed. If you were there you would understand.

History

'C' is called a language and SAS is called a system, why?

- In general programming languages aim at building flat structure of basic commands and contructs to instruct the computer what operations to take upon data structures. Their libraries/APIs etc written in the same language asprire to the same conventions and are usually written as a single application/executable in which all parts can be included. Now, SAS over so many years has produced a lot of software to use. Not all of it comes packaged with the standard install. The components can be loosely cooperative through the GUI terminal, and some require extra purchases. Some of the tailoring engineered into SAS allows companies to integrate it into their businesses quite easily.

History

I have heard of many ways to do statistical computation. There is MATLAB, R, SPSS, Python, and even in 'C'. Why do we choose SAS?

- You are very right in that there are many options to do the same thing and it is good to know the options. In the future you will probably have to use a couple of them as well and at least be acquainted with the rest.

- Is SAS then just a random choice to make of the selection? Why not choose one of the free options or MATLAB, or even Python which is quite strong?

- The choice of a platform (language or system) to do a statistical analysis can be complex. There are some basic considerations to take into account.

Why choose SAS?

Reasons

- The system has evolved largely influenced by corporations integrating the software into their business models. The Python language has evolved from academics. MATLAB has evolved originally from mathematicians, but that influence has decreased and the engineering firms now the primary customers (their Simulink system/product is one of a kind).
- SAS combines many spreadsheet features found in excel, which makes it accessible to even people coming from less rigorous backgrounds, and can feel familiar.
- A strong tradition in consultancy. The installation of components tailored for specific actions is appealing to those who don’t have the time for training but want expertise to solve immediate problems.

Why choose SAS?

The options such as Python seem more attractive because of all the people using it and the great tutorials on Youtube

When choosing a language/platform many recommend to join the community that comprises the development most similar to yourself. Ex. web developers steer the Ruby development and they style suites those applications (academics are not fond of the 'more than one way to do things' philosophy of ruby, and like Python's one correct way period). Nothing is perfect, and SAS is geared towards data analysts. It is not Mathematica for theoretical mathematicians, or Maple for those working on fundamental algebra. SAS works well for those who have data and are focused on drawing intelligent conclusions from it.

Why choose SAS?

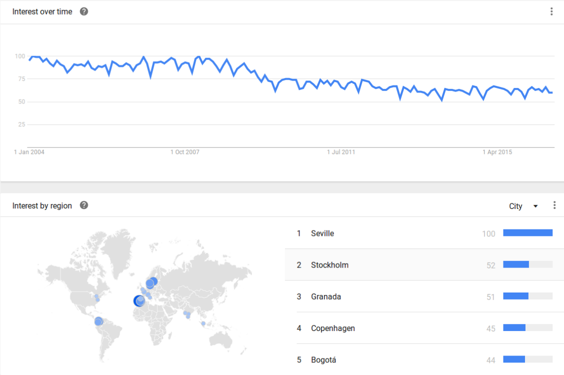
So who uses it is academics love Python, webmasters Ruby, engineers MATLAB, mathematicians Mathematica and developers C/C++?

When choosing a language/platform many recommend to join the community that comprises the development most similar to yourself. Ex. web developers steer the Ruby development and they style suites those applications (academics are not fond of the ’more than one way to do things’ philosophy of ruby, and like Python’s one correct way period). Nothing is perfect, and SAS is geared towards data analysts. It is not Mathematica for theoretical mathematicians, or Maple for those working on fundamental algebra. SAS works well for those who have data and are focused on drawing intelligent conclusions from it.

Why choose SAS?

That is not quite an exact image you are giving me

- Banks (eg. Sainsbury's bank)
- Biotech industry
- Goverment agencies



1	Seville	100
2	Stockholm	52
3	Granada	51
4	Copenhagen	45
5	Bogotá	44

Why choose SAS?

I don't want to pester but why not choose R? It is free as well. What does SAS do that R does not?

R has lots of libraries and functions to use and do almost any statistical analysis on it, so you are right to ask and make sure you are not wasting your time with redundant pursuits. You can write R code to perform an analysis, but it will frequently require alterations when the data changes and that is the strength of SAS. The components are easily adjustable with minimal effort. The writers of SAS work to eliminate users getting stuck. • This might not be exactly what you are looking for right now, but it is very useful to know. • If you consider it outdated, remember till this day 'legacy' code runds most of our lives still, COBOL!

Why choose SAS?

You won't be able to learn every langauge/platform, but as someone interested in data analysis this is undeniably useful.

OK

Between us, the interface reminds me of the rustic computing days of Windows95.

Does that mean that the consultancy tradition is still strong for SAS?

Yes.

An important point to remember if you are not familiar or accustomed to switching between languages/platforms/systems; many times **CONVENTIONS WILL NOT MAKE SENSE IMMEDIATELY**. *You have to adopt the convention to appreciate it over time*. The choices made are sometimes arbitrary and was chosen by the 'community' responsible for them given options.

current section

1. an interesting read

2. Background on SAS (Statistical Analysis System)

3. Basics

SAS fundamentals

SAS programming has the paradigm to organise code so that you have commands for the **DATA steps** and then for the **PROCEDURE steps**. Then an **OUTPUT step** can be used for result presentation.

• The data step refers to the information/data/tables/csv's etc that you use as input and the modifications you need to make on it, or even selections.

• The procedure step performs the computational analysis and the outputs you choose to produce.

SAS fundamentals

The Data Step

- Reading in data
- Reorganising the data(structure)
- Displaying data to the user

If you have worked with other languages this is essentially preconfigured ’for-loop’ situations where the majority of cases are configured and packaged into a ’step’. A basic view:

```
procedure Data Step skeleton
    data newdata;
    set mydata;
    -more statements-
    run;
end procedure
```

SAS fundamentals

First of all, input data. That is the first thing on my mind

The Data Step

infile names an external file to read from

```
procedure DATA STEP SKELETON
    data one;
    infile “input.data“;
    input a b c;
    run;
end procedure
```

datalines reads from inline data you write manually (or CARDS)

```
procedure DATA STEP SKELETON
    data one;
    input a b c;
    datalines;
    1 2 3
    ;
    run;
end procedure
```

SAS fundamentals

- You can get SAS to include your SAS commands in a file using **include**.

```
procedure Read commands externally and then return
    proc print data=one;
    run;
end procedure
```

This seems like random commands thrown at me.

The quick brush is an important step. I just want you to see the style of commands, it will sink in. *Multipass repitition is the best method to get a feel for it.* (I prefer this to a thorough single pass)

SAS fundamentals

```
procedure SAS DATA SET FROM RAW INPUT
    DATA TEMP;
    INPUT C;
    DATALINES;
    1
    2
    3
    4
    5
    ;
    RUN;
end procedure
```

- Every command line ends with a semicolon (';').

SAS fundamentals

Log Window display

```
NOTE: The data set WORK.TEMP has 5 observations and 1
variables.
NOTE: DATA statement used (Total progress time):
      real time       0.08 seconds
      cpu time        0.03 seconds
```

The observations are 1 through to 5, and the single variable is 'C'. All kept in the 'TEMP'.

SAS fundamentals

```
procedure The MEANS Procedure
    PROC MEANS DATA=TEMP;
    RUN;
end procedure
```

Results Viewer - SAS Output window in the GUI, is a tab. This window produces HTML whose source can be viewed and be directly included into a webpage or database for web usage.

1. Analysis Variable : C
2. N - Mean - Std Dev - Minimum - Maximum
3. 5 - 3.0000 - 1.5811 - 1.0000 - 5

SAS fundamentals

```
procedure SAS DATA SET FROM RAW INPUT
    DATA EXAMPLE1;
    INPUT NAME $ SEX $ AGE INCOME;
    CARDS;
    Alex M 18 1000
    Bob M 19 2000
    Catherine F 20 3000
    Dorothy F 21 4000
    Eliza F 22 5000
    Felix M 23 6000
end procedure
```

• Every command line ends with a semicolon (’;’). • The dollar sign (’$’) after a variable name indicate that they are character variables and not numbers.

SAS fundamentals

```
procedure Basics
    PROC PRINT DATA=EXAMPLE1;
     VAR AGE INCOME;
    RUN;
end procedure
```

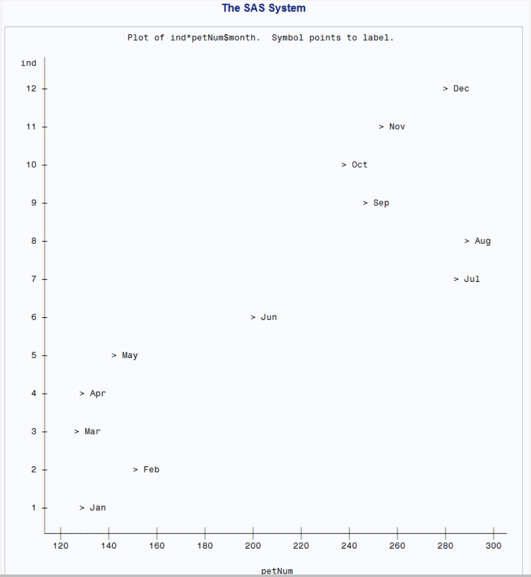
```
procedure Basics
    PROC MEANS DATA=EXAMPLE1;
     VAR AGE INCOME;
    RUN;
end procedure
```

King County, Washington PetData 2016 Pet Adoptions

```
data one;
input ind month $ petNum;
datalines;
1 Jan 129
2 Feb 151
3 Mar 126
4 Apr 128
5 May 143
6 Jun 200
7 Jul 285
8 Aug 288
9 Sep 247
10 Oct 238
11 Nov 253
12 Dec 279
;
run;
```

```
proc plot data = one;
plot ind*petNum $month;
by month; run;
```

output of pet plot


The SAS System

Plot of ind*petNum$month. Symbol points to label.

Remember

- SAS commands/statements begin with DATA/PROC or another keyword
- They end with a semicolon ';'

This is the pattern of how we pass instructions to operate on the data?

Yes, and more to follo wfrom next time.