Where to find SAS:
1. To use SAS with the department of Statistics computer system, you need first sign on to the server. You write your script in text file and save it with .sas file extension, e.g. script.sas. To run it you type

   > sas script.sas

   You will see two output files: first.log and first.lst.

2. SAS is available at UF computer labs such as the one in McCarty B.

3. You may purchase SAS from UF Software Licensing Services as a UF employee.

SAS has a lot of documentation/help easily accessible since it is not open source like R. See http://support.sas.com/documentation/index.html

A nice SAS tutorial that is available can be found at http://galsterhome.com/stats/Tutorial/sas_tutorial_contents.htm

SAS Lesson 1: Input Data by Hand

SAS is a comprehensive system which integrates utilities for storing, modifying, analyzing, and graphing data. Learning to write SAS programs is difficult at first, but doing so will enable you to perform the specific tasks which are most appropriate for your data. Unlike other packages which are easier to use, SAS offers the flexibility for you to make necessary modifications to standard analytic techniques or to select from several possible ways to analyze the same data.

In this lesson, you will write a simple SAS program in which data will be entered into the computer and printed. The instructions below are appropriate most SAS Versions, but for some computers, though all of the SAS statements will remain the same, there will be some differences in I/O or operational instructions.

The following example will show you how to enter data into SAS and obtain a printout. The data represent October 1997 estimates of yields of early-season oranges (e.g. navel) and late-season oranges (e.g. Valencia) in four U. S. states. Crop estimates are in units of millions of boxes. The SAS dataset will have three variables (state, early-season yield estimates, and late-season yield estimates) with four observations, where each observation corresponds to one state.
On any text editor, you write your SAS program. Here is a very simple one:

```
OPTION PS=55 LS=76 NOCENTER NODATE NONUMBER;
TITLE 'Citrus Data';
DATA oranges; INPUT state $ 1-10 early 12-14 late 16-18;
DATALINES;
Florida     130  90
California  37   26
Texas       1.3  .15
Arizona     .65  .85
PROC PRINT DATA=oranges;
RUN;
```

Let us start with the first line,

```
OPTION LS=76 PS=55 NOCENTER NODATE NONUMBER;
```

means horizontal printing lines = 76, vertical printing lines = 55 characters without centralization, date or page numbers.

```
DATA oranges;
```

The word DATA instructs SAS that you are going to provide data in following steps.

SAS has two main categories of operations. Data steps are used to input and modify datasets. Procedure steps, or PROCs, perform operations on the data. The word “oranges” gives a name to the dataset that will be created. Any name can be assigned to a dataset, as long as it follows these rules:

- It contains only the following characters: the letters A through Z, the digits 0 through 9, and the underscore.
- It contains 1 to 8 characters. (Although this rule may have been relaxed in newer versions, it is best to keep the names short.)
- The first character is not a digit.
- SAS is not letter case sensitive. Thus data oranges or DATA ORANGES means the same thing. I intend to use capital letters for SAS instructions and small letters for user assigned names.

Finally, the statement ends with a semicolon. This tells SAS that the current instruction has ended and to proceed with the next instruction. With a few exceptions, semicolons must be placed at the end of each statement in SAS. Many errors in SAS are caused by forgetting the semicolon.

In the next line, add the following:

```
INPUT state $ 1-10 early 12-14 late 16-18;
```

Three variables are named in this statement: STATE, EARLY, and LATE. The rules for naming variables are the same as the rules for naming datasets. The numbers refer to positions of the variables and are optional. For example, the data for STATE will be listed beginning in the leftmost column 1 and using up to 10 characters. The $ specifies that STATE should be regarded as a character variable and not a numeric variable. By default, SAS assumes that all variables are numeric.
The next line is

**DATALINES;**

This indicates that the actual data will appear next. The statement **CARDS;** is equivalent.

Next, enter the data. An optional single semicolon may put at the end of data with a line by itself after the last line of data.

```
Florida    130  90
California  37  26
Texas      1.3 .15
Arizona    .65 .85
;
```

Now, add the instructions for SAS to print the data. One of the procedures in SAS, PROC PRINT, can be used to provide a simple printout of the data.

```
PROC PRINT DATA=oranges;
RUN;
```

Finally, end with a RUN statement. This tells SAS to process the data and provide the information you have requested.

Here is a second example to illustrate some additional features including titles and labeling of variables

```
options nodate nonumber ps=55 ls=80;
title 'Entering Data';
title2 'Example 2';
data ex2;
input temp mat @@;
label temp='temperature C' mat='rate of maturation';
cards;
10 26.3  10 21.3  15 15.2  15 16.4  20 14.5  20 13.6  25 7.5
25 9.3  28 8.7  28 8.7  30 5.8  30 5.6  32 9.4  32 9.8
;
proc print data=ex2 label;
run;
```

Pay attention to the input line. Two variables “temp” and “mat” are to be read. The @@ statement allows us to enter data in lines. It specifies that the first number to be read is for temp, the second for mat, the third for temp again and so forth. Below is the output.
## Entering Data
### Example 2

<table>
<thead>
<tr>
<th>Obs</th>
<th>temperature</th>
<th>rate of maturation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>26.3</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>21.3</td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td>15.2</td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td>16.4</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>14.5</td>
</tr>
<tr>
<td>6</td>
<td>20</td>
<td>13.6</td>
</tr>
<tr>
<td>7</td>
<td>25</td>
<td>7.5</td>
</tr>
<tr>
<td>8</td>
<td>25</td>
<td>9.3</td>
</tr>
<tr>
<td>9</td>
<td>28</td>
<td>8.7</td>
</tr>
<tr>
<td>10</td>
<td>28</td>
<td>8.7</td>
</tr>
<tr>
<td>11</td>
<td>30</td>
<td>5.8</td>
</tr>
<tr>
<td>12</td>
<td>30</td>
<td>5.6</td>
</tr>
<tr>
<td>13</td>
<td>32</td>
<td>9.4</td>
</tr>
<tr>
<td>14</td>
<td>32</td>
<td>9.8</td>
</tr>
</tbody>
</table>