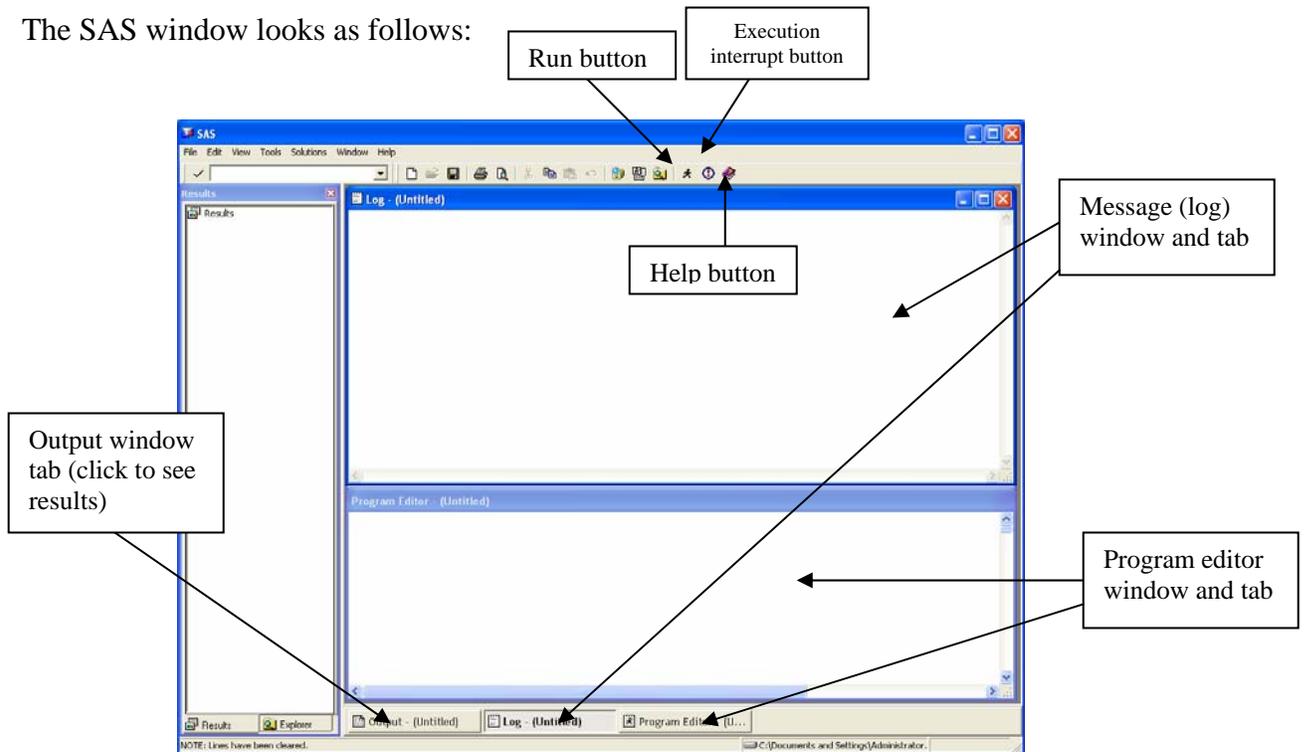


Session 1: Entering data into SAS

In this session we will “dissect” the DATA step of SAS

The SAS window looks as follows:



The DATA step is used to enter, edit or define a new data set in SAS. The syntax is as follows:

```
data dataname;          * Note that all lines in SAS end in ';' ;
```

Where *dataname* is a name for the new data set.

There are various ways to enter data into SAS. The simplest is by hand.

For example, consider entering the following data set of survival times (the sign “+” denotes a censored observation):

1, 2, 2, 2⁺ 3, 5, 6, 7⁺, 8, 16⁺, 17, 34⁺

We will enter these data into SAS manually, creating a variable (*nhltime*) for the survival time and a second (*fail*) for the censoring indicator. We will call this data set *nhodlymph*, for non-Hodskin’s lymphoma.

The SAS code is as follows:

```
data nhodlymph;
  input nhltime fail;
  datalines;
  1 1
  2 1
  2 1
  2 0
  3 1
  5 1
  6 1
  7 0
  8 1
  16 0
  17 1
  34 0
;
run;
```

Consider the components of this data step:

```
input nhltime fail;
```

This is the input statement that tells SAS that variables are to be inputted and their names.

After an input statement, there needs to be a statement declaring how the variables will be entered in the data. In our case, we will be entering them manually, so we write

```
datalines;
```

The values of the data points will follow. Note that they need to be entered in the order that were declared in the input statement. The first one will be a value for the variable `nhltime`, the second one will be associated with `fail`, the third with `nhltime`, the fourth with `fail` and so on. Unpredictable things will happen if SAS does not find something and goes looking for a value in the next line (for example, if there is no value for `fail` in the third line, the value 2 in the fourth line will become a value for `fail`, the value 0 a value for `nhltime` and so on (you don't want that!). If the values that you are entering are truly missing, then put a period (".") to signify that the value is missing and satisfy SAS (missing character values become more complicated, so we will not discuss them here).

Notice also that there are no semicolons (";") until the data entry stream is complete.

Every piece of code should end with the statement

```
run;
```

although this is not strictly necessary (as soon as SAS finds another data step or procedure it will execute the code; however it is a good habit to establish, especially if we run code

in segments as we will show later on).

Now let's try to run the data. You do that by clicking on the icon  on the tool bar. Alternatively you can press the F8 key. SAS will produce the following comments:

```
39 data nhodlymph;
40     input nhltime censor;
41     datalines;

NOTE: The data set WORK.NHODLYMPH has 12 observations and 2 variables.
NOTE: DATA statement used:
      real time           0.03 seconds
      cpu time            0.03 seconds

54 ;
55 run;
```

You do not want to see any other color in the comments (especially red!). Warnings will be in green.

Now let's print the data set. SAS accomplishes all actions through "procedures" or PROCs. These have a standard syntax. Consider the simplest of all, the printing procedure. To print the data set nhodlymph we do the following:

```
proc print data=nhodlymph;
    title "Non-hodgkin's lymphoma data set";
run;
```

Now consider the components of this procedure:

```
proc print data=nhodlymph;
```

After the statement `proc` follows the name of the procedure (here `print`). Then follows the name of the data that will be processed. Here this is `nhodlymph`. The data set is recognized as the name following the statement `data=`. It is not strictly necessary to use this. SAS will use, by default, the last data set created. However, it is a good habit to *always* write the data set name, since, eventually, you will have created many data sets and you will not know which one is the "default" one. The `proc` statement can also include several options. It is ended by a semicolon.

Notice the title statement in the second line of the previous code:

```
title "Non-hodgkin's lymphoma data set";
```

This is a title that will be attached to the output. Titles in SAS remain active until superseded by another title. You can attach subtitles by numbering them. The first title is `title1` or simply `title`. The second subtitle is `title2` and so on.

Note: Higher level (lower number) titles remove lower-level titles but do not remove higher-level titles!

The text of a title is included between single or double quotes. To include an apostrophe (right single quote) in a title, you must use double quotes around the text and the apostrophe (as shown above).

The results are as follows:

Non-hodgkin's lymphoma data set			2
			06:07 Monday, December 8, 2003
Obs	nhltime	fail	
1	1	1	
2	2	1	
3	2	1	
4	2	0	
5	3	1	
6	5	1	
7	6	1	
8	7	0	
9	8	1	
10	16	0	
11	17	1	
12	34	0	

The SAS output includes a time/date stamp and a page stamp. These can be removed, but we will not concern ourselves with this at present.

Note that, by default, SAS prints an observation counter (under Obs above). If, for some reason, you want to remove it, type the previous print procedure with the NOOBS option as follows:

```
proc print data=nhodlymph noobs;  
  title "Non-hodgkin's lymphoma data set";  
run;
```

Now let's complete the laboratory session by carrying out a simple Kaplan-Meier analysis of this data set. This is accomplished in SAS with another procedure named `lifetest`. The code is as follows:

```
proc lifetest data=nhodlymph method=pl plots=(s);  
  time nhltime*fail(0);  
  title "Kaplan-Meier analysis of the non-Hodgkin's lymphoma data set";  
run;
```

Let's analyze this procedure.

```
proc lifetest data=nhodlymph method=pl plots=(s);
```

The syntax of the procedure is standard. There are however, two options that request specific analyses. The first is `method` and the second is `plot`. The `method` option tells SAS what method of estimation of the survival distribution to use. There are two

methods: The Kaplan-Meier method (denoted by the letters PL, i.e., product-limit, or KM, Kaplan-Meier estimator). These are entered following an equal sign “=”. The requested plots can be survival plots “(s)”, minus log-survival plots (i.e., $-\log S(t)$ versus time t) “(ls)”, log-minus-log survival plots ($\log[-\log S(t)]$ versus t) “(lls)”, hazard “(h)”, probability distribution function ($F(t)$) plots “(pdf)” and a plot of the censored observations “(c)”. You can request multiple plots by entering a list of plots separated by commas. The output of the previous commands is as follows:

Kaplan-Meier analysis of the non-Hodgkin's lymphoma data set					4
					06:07 Monday, December 8, 2003
The LIFETEST Procedure					
Product-Limit Survival Estimates					
nhltime	Survival	Failure	Survival Standard Error	Number Failed	Number Left
0.0000	1.0000	0	0	0	12
1.0000	0.9167	0.0833	0.0798	1	11
2.0000	.	.	.	2	10
2.0000	0.7500	0.2500	0.1250	3	9
2.0000*	.	.	.	3	8
3.0000	0.6563	0.3438	0.1402	4	7
5.0000	0.5625	0.4375	0.1482	5	6
6.0000	0.4688	0.5313	0.1503	6	5
7.0000*	.	.	.	6	4
8.0000	0.3516	0.6484	0.1517	7	3
16.0000*	.	.	.	7	2
17.0000	0.1758	0.8242	0.1456	8	1
34.0000*	.	.	.	8	0

NOTE: The marked survival times are censored observations.

Summary Statistics for Time Variable nhltime

Quartile Estimates

Percent	Point Estimate	95% Confidence Interval [Lower Upper)	
75	17.0000	6.0000	.
50	6.0000	3.0000	17.0000
25	2.5000	2.0000	8.0000

Mean Standard Error

8.6432 2.1132

NOTE: The mean survival time and its standard error were underestimated because the largest observation was censored and the estimation was restricted to the largest event time.

Summary of the Number of Censored and Uncensored Values

Total	Failed	Censored	Percent Censored
12	8	4	33.33

Kaplan-Meier analysis of the non-Hodgkin's lymphoma data set

