

**SAS Introduction and Selected Textbook Examples by SAS Code for
“Survival Analysis Using S: Analysis of Time-to-Event Data by
Tableman and Kim”**

**Jong Sung Kim
Assistant Professor of Statistics
Department of Mathematics and Statistics
Portland State University**

SAS Introduction and Selected Textbook Examples by SAS Code for “Survival Analysis Using S: Analysis of Time-to-Event Data by Tableman and Kim”

Jong Sung Kim*
Assistant Professor of Statistics
Department of Mathematics and Statistics
Portland State University

This packet is prepared for students, practitioners, and researchers who are not familiar with SAS. As such, an instructor using the above text can prepare lectures based on this packet. The SAS code provided parallels major S functions to perform the analyses discussed in the text. However, I do not provide instructions on how to create SAS macros. This task is left to the instructor of the course. In Chapter 6, however, I use a couple of macros which are downloaded from www.mayo.edu/hsr/biostat.html. These macros are well written, but I added several additional comments to help new users of SAS to grasp contents easier. The instructor is then recommended to introduce SAS macro programming and ask students to compose simple macros when appropriate. For example, a good exercise would be to make some transformations of the output variables and construct confidence intervals. This would provide students with a good SAS programming experience without spending much of the class time for that purpose. There are a number of good SAS books available in the literature. I suggest readers interested in SAS programming to refer to Cody and Pass (1995) and Burlew (1998) among others for further details.

Introduction (the first chapter) focuses on how to create a library and import a text file and an excel file into SAS. I appreciate Aleksandra Sumic, a former Master of Public Health student at Oregon Health & Science University, for her help on this chapter. The rest covers main parts of the Chapters 2 – 7 of the text. I appreciate Solange Mongoue, a student in Ph.D. Mathematical Sciences program, for her reading materials for Chapters 2 – 7.

*Copyright(c) 2005 by Jong Sung Kim, all rights reserved. This packet may be freely shared among individuals, but it may not be reprinted in any medium without written consent from the author and the publisher.

Contents

Creating a Library in SAS	4
Importing Excel File into SAS	6
Importing Text File into SAS	12
SAS Textbook Examples for Chapter 2: Nonparametric Methods	17
SAS Textbook Examples for Chapter 3: Parametric Methods	24
SAS Textbook Examples for Chapter 4: Regression Models	29
SAS Textbook Examples for Chapter 5: Cox Proportional Hazards Models	31
SAS Textbook Examples for Chapter 6: Model Checking: Data Diagnostics	37
SAS Textbook Examples for Chapter 7: Additional Topics	61
References	68

Creating a Library in SAS

When working with SAS, we need to tell the program where is the data set that we want to use located. In SAS, this location is referred to as a *library*. This means that prior to importing or creating a data set in SAS, we need to make sure that we have a library where we will store the data. There are two types of libraries in SAS, the permanent and the temporary ones. The temporary library is called a “WORK Library”. We do not need to create this library as it appears under Libraries in the Explorer window every time we start the program. Unless instructed differently, SAS will look for and store the data sets in this location. If the data set that we want to use is not located in the Work library and we forget to instruct SAS otherwise, the error message will appear in the log window stating that a data set does not exist. In addition, keep in mind that everything stored in this temporary library will be erased at the end of each SAS session. In order to save the data for the future use, one should chose permanent location when either importing already existing or storing newly created data sets in SAS. The permanent library can be created in two different ways:

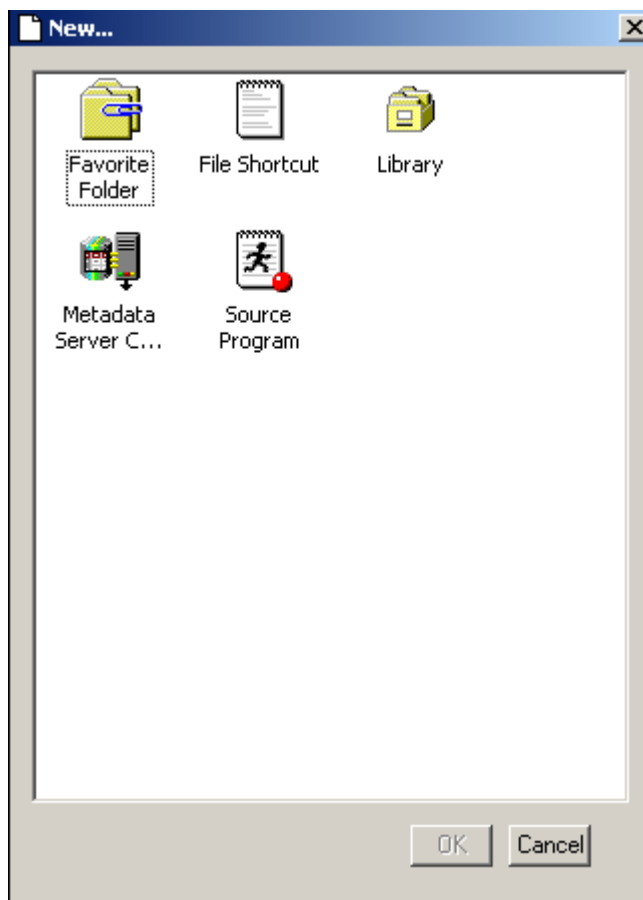
1) Using File menu:

Open SAS

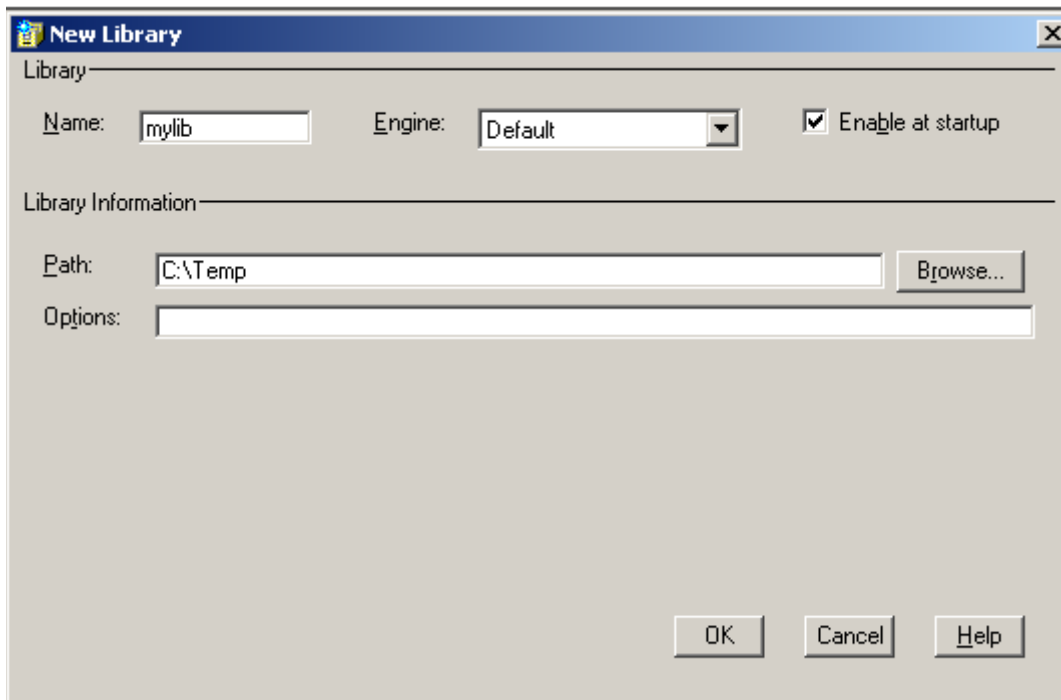
Select Explorer window

Go to File

Click New and a window similar to the one below will appear



Click Library and a window similar to the one below will appear



In the 'Name' window, type in the name you have chosen for your new library.

In the 'Path' window, type in (or locate using the Browse button) the path where your data set will be stored.

Select 'Enable at startup' box if you want your library defined every time SAS starts up.

Click OK.

2) Using LIBNAME statement:

In the editor window, type in the following command:

```
libname mylib 'C:\Temp';  
run;
```

After the libname statement, type in the name you are assigning to your new library (in this example *mylib*). This name is referred to as a library reference (libref). Following the library reference, type in the path for the location for your data set (in this example it is C:\Temp). Now run the code, and your library should be created.

Once you have created the library, you should be able to see it under “Libraries” in the explorer window. Given that you have already imported some data in SAS format, you should be able to see all the SAS data sets stored in the location you have specified for this library.

Now that you know how to create a library, you can proceed to import data into SAS. Depending on your needs (temporary vs. permanent data set) choose WORK or mylib library for the location. If you would like to create a permanent data set, be aware that you cannot create a new library from the Import Wizard environment. So make sure you create a library before you use the Import Wizard.

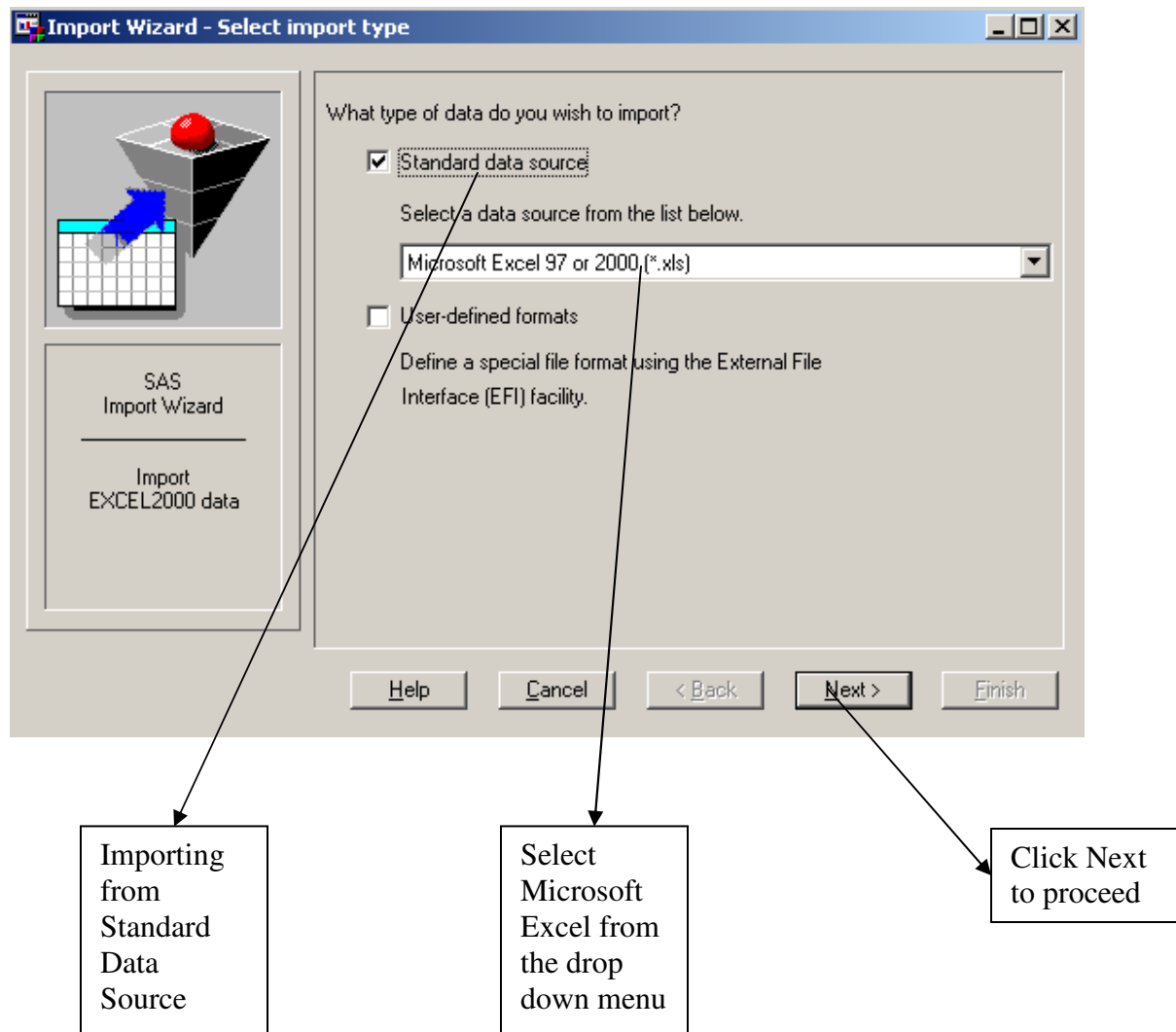
Importing Excel File into SAS

1)

The easiest way to import an Excel file into SAS is to use Import Wizard. To access the Import Wizard, start SAS and then, from the drop down menu, select

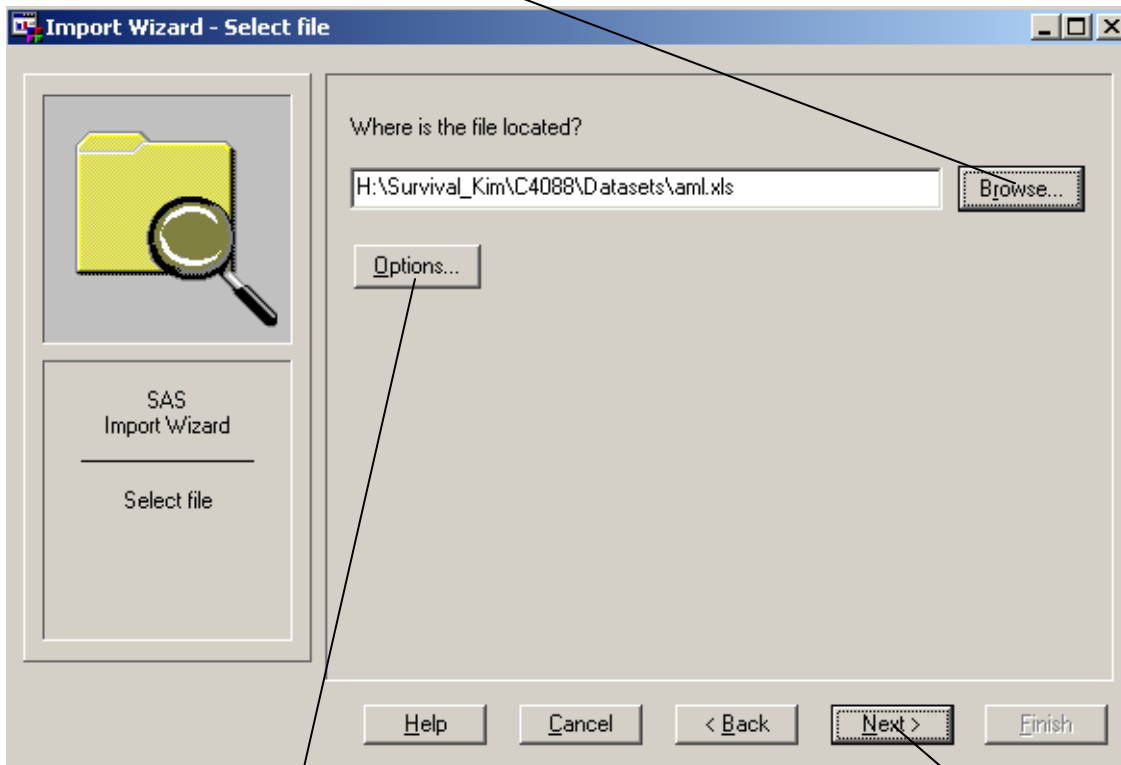
File → Import Data

This will take you to the Import Wizard as shown on the image below:



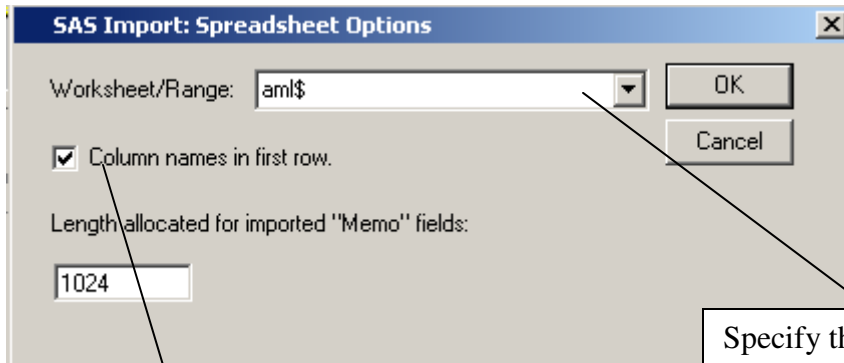
2)

In the next step **type in** the location of the file you want to import, or simply use **Browse** button to locate the file



If you have more than one worksheet in your file and if the variable names are included in the first row of the spreadsheet, click on the Options button. A window similar to the one below will appear. Fill in the required fields and then click OK to proceed.

Click Next to proceed

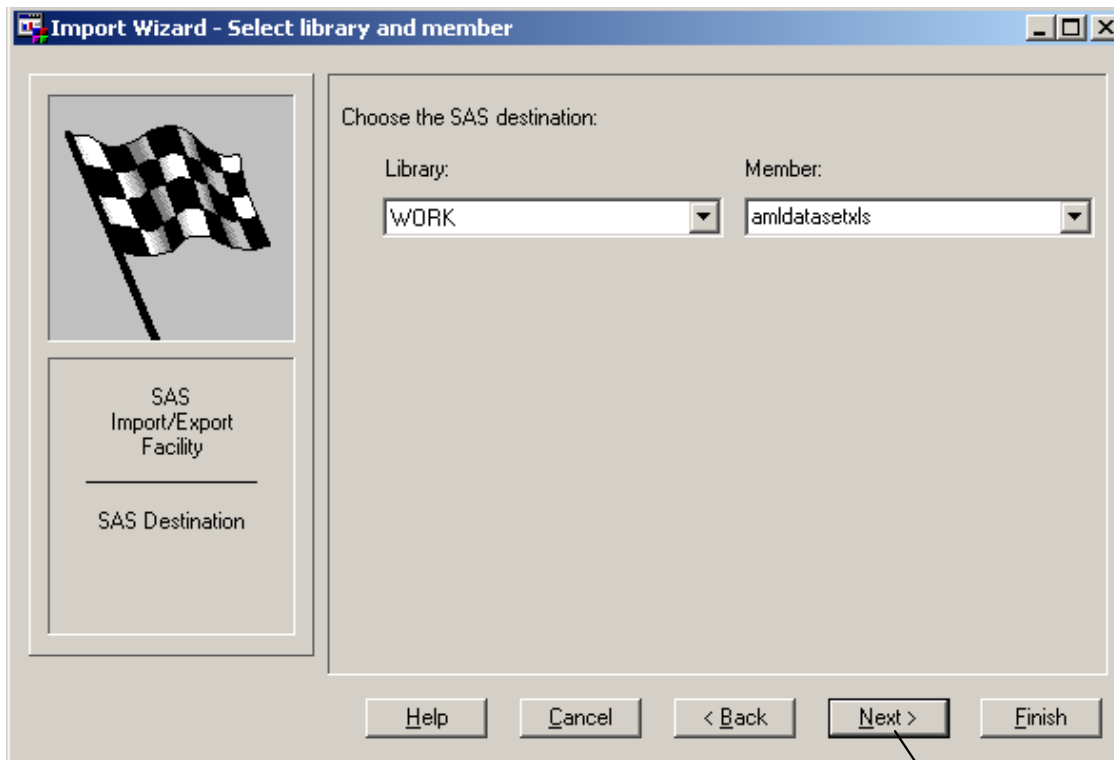


Select this box if variable names are included in the first row of the spreadsheet

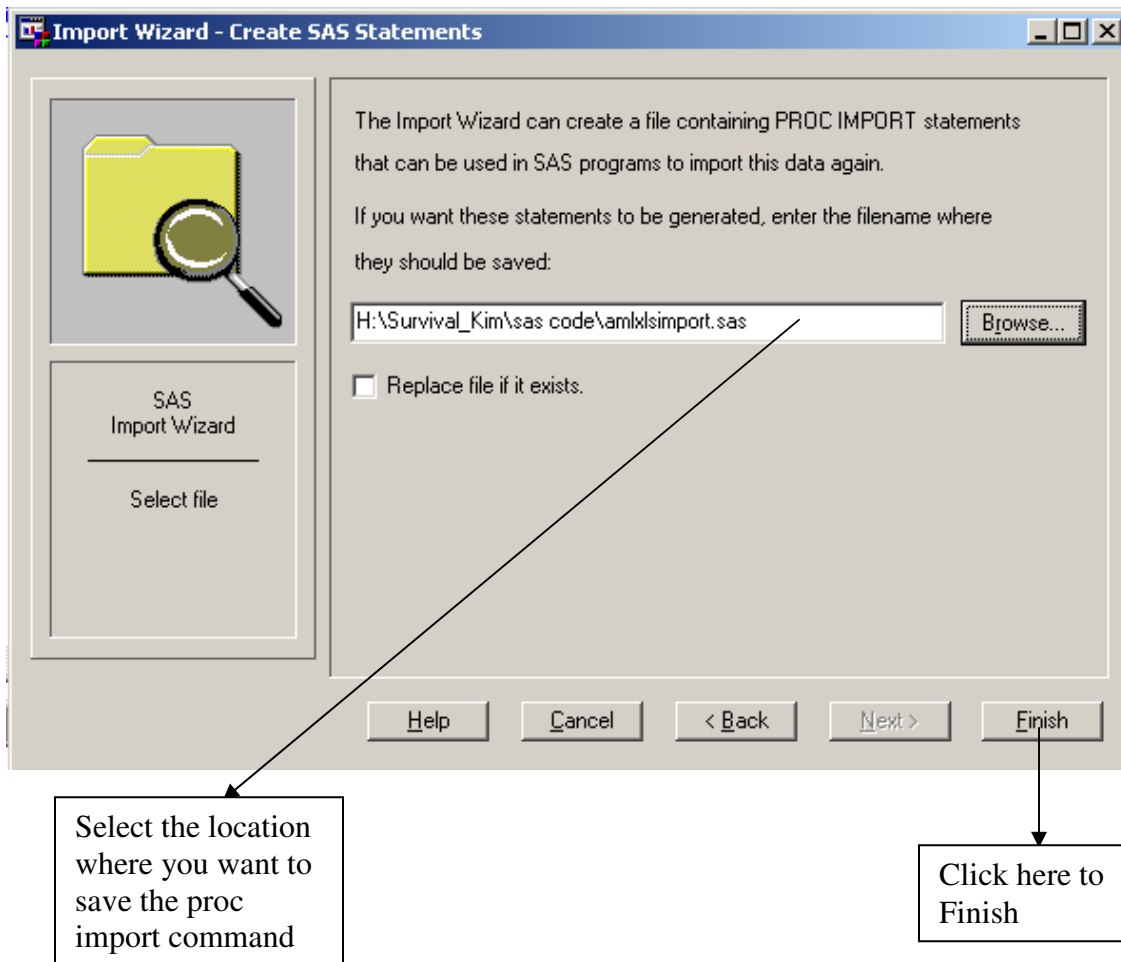
Specify the worksheet in the Excel file you want to import (necessary only if you have more than one in your file – our first assignment has one sheet only)

3)

Next, select the SAS library you want to import your file into (e.g. Library: Work) and type in the name of your dataset (e.g. Member: amldatasetxls)



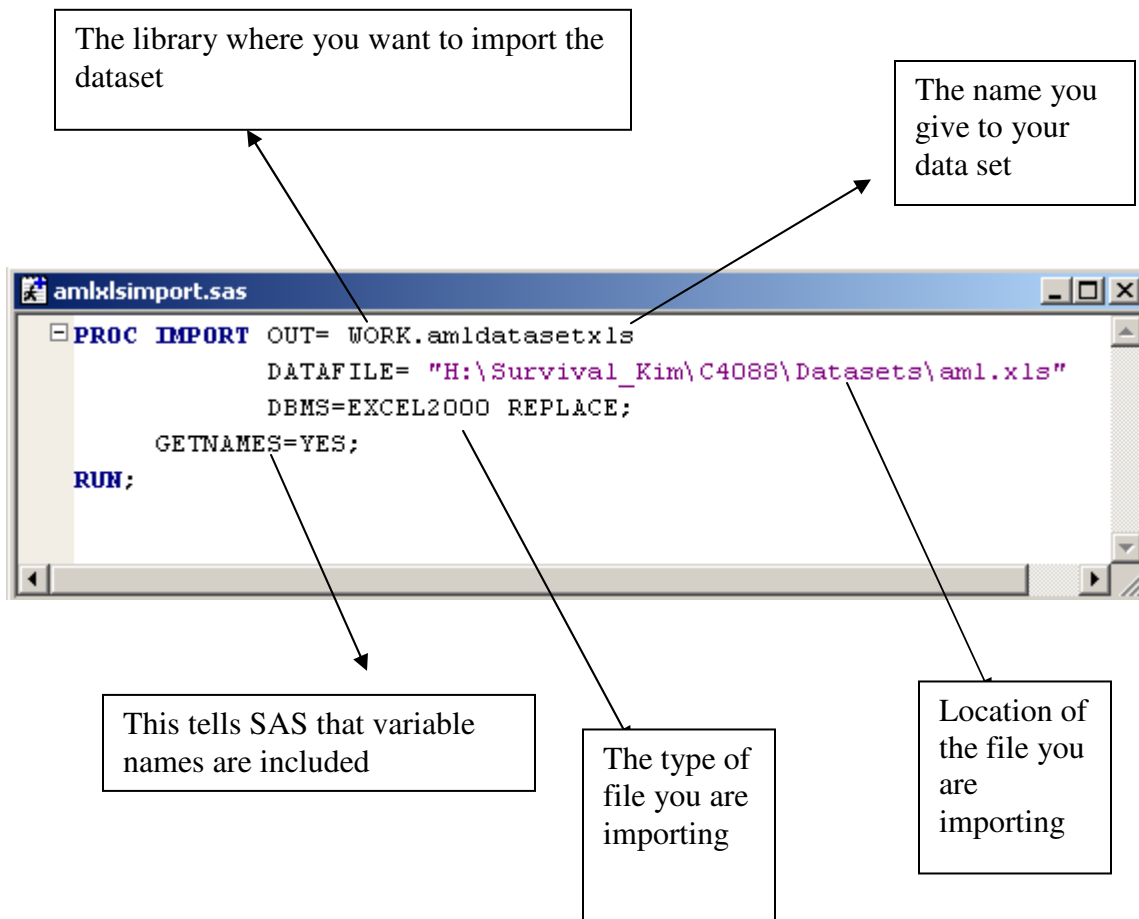
Another way to import a file into SAS is to start the program, and in the editor window, type in the code that tells it to import the Excel file (proc import command). The easy way to find out how this code looks like is to select **Next** button in this step of the import using the Import Wizard. A window similar to the one below will appear:



If you have done everything successfully, the statement “WORK.amldatasetxls was successfully created” should appear in your Log window. Go to the Explorer window. You should be able to find your data set in the Work library (or the library you have chosen for this assignment). Double click on it. You should now be able to see it.

4)

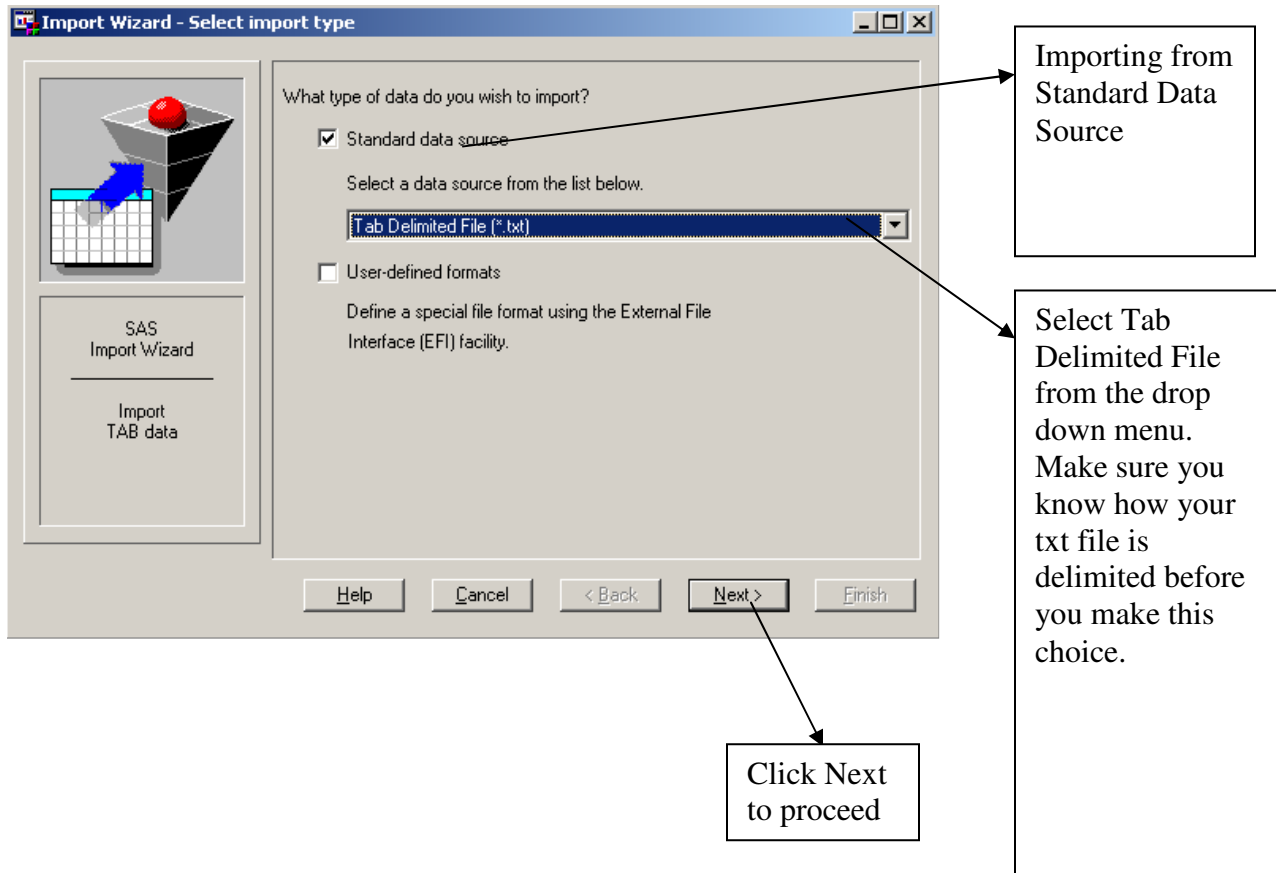
To look up the proc import command you have saved earlier, select the Editor window in SAS and open the code from the location you have chosen. You should be able to see the window with the following code:



Now try to import the Excel file using the proc import command! Open SAS as usual, and type the above code in. Make necessary adjustments if you are using different library, names, etc. You should get the same result as the one you got using the Import Wizard.

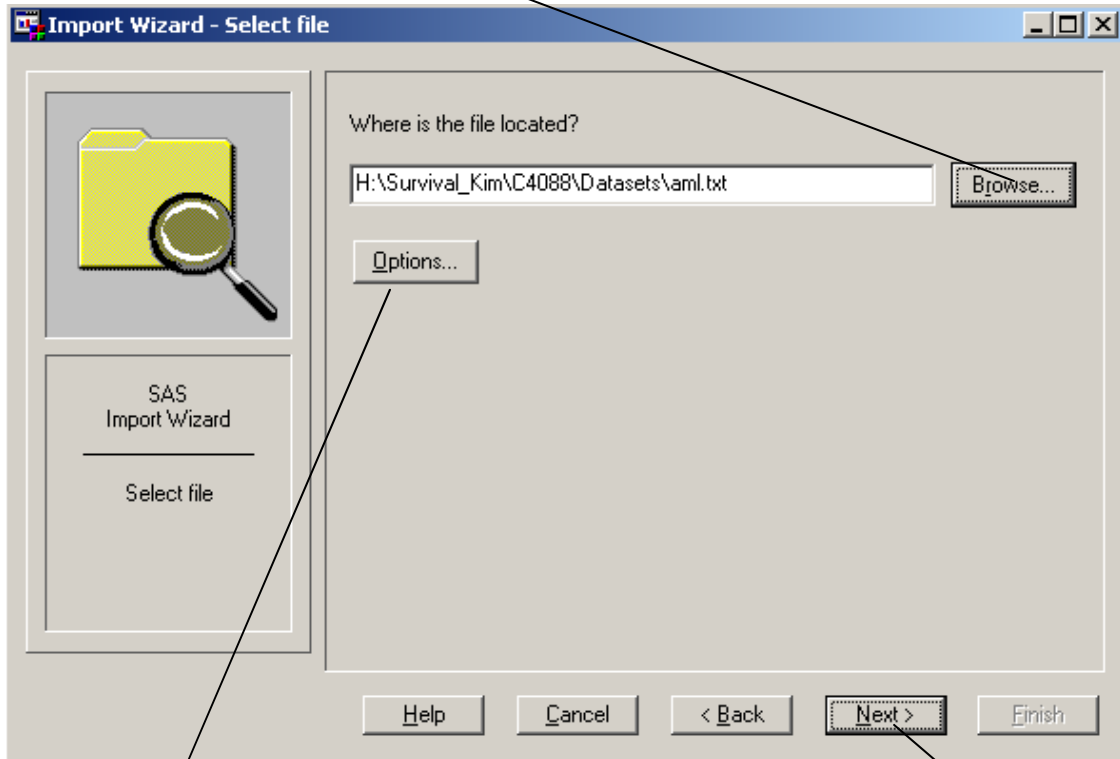
Importing Text File into SAS

1). The easiest way to import a Text file into SAS is to use Import Wizard. To access the Import Wizard, start SAS and then, from the drop down menu, select **File** → **Import Data**. This will take you to the Import Wizard as shown on the image below:



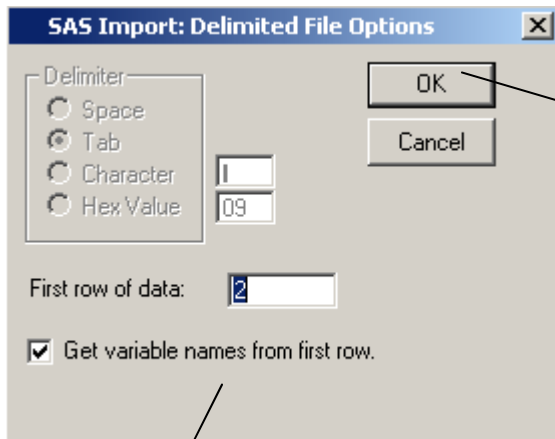
2)

In the next step **type in** the location of the file you want to import, or simply use **Browse** button to locate the file



Click on the options button

Click Next to proceed

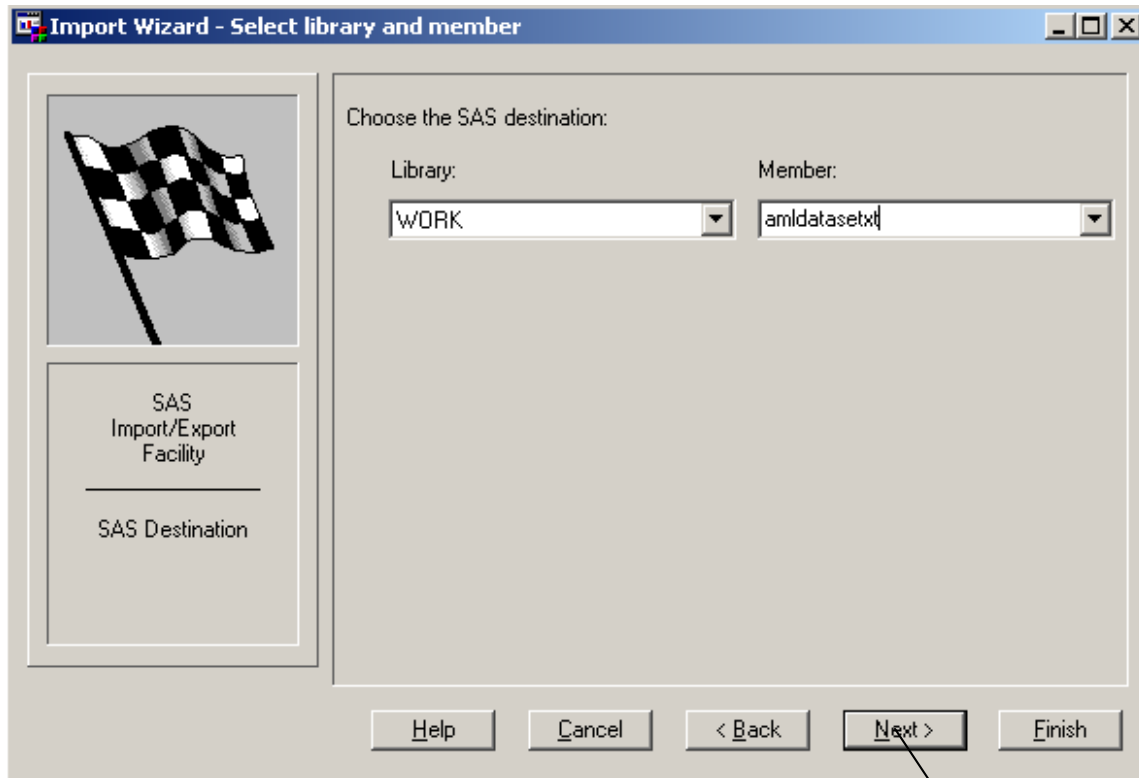


Click OK to proceed

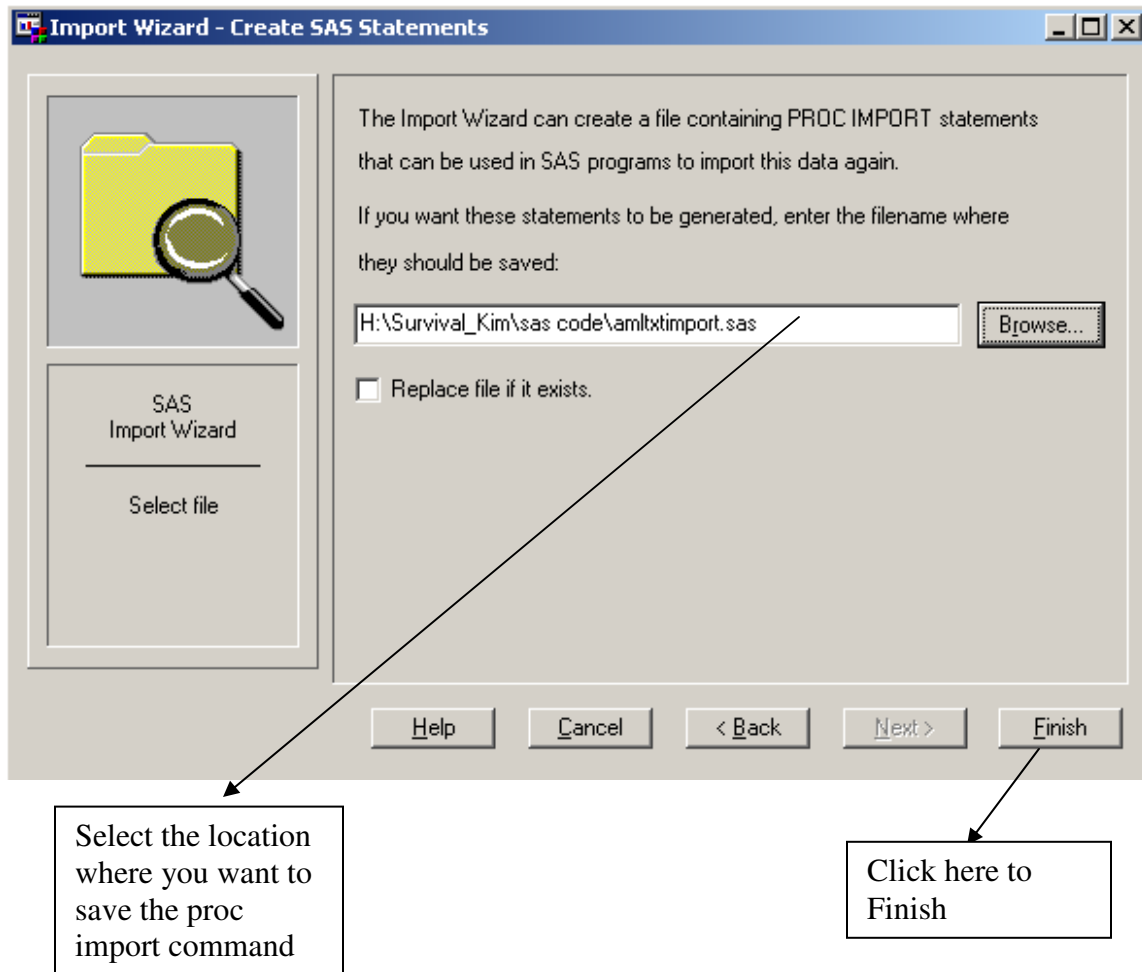
Select this box if variable names are included in the first row of the spreadsheet

3)

Next, select the SAS library you want to import your file to (e.g. Library: Work) and type in the name of your dataset (e.g. Member: amldatasetxt)



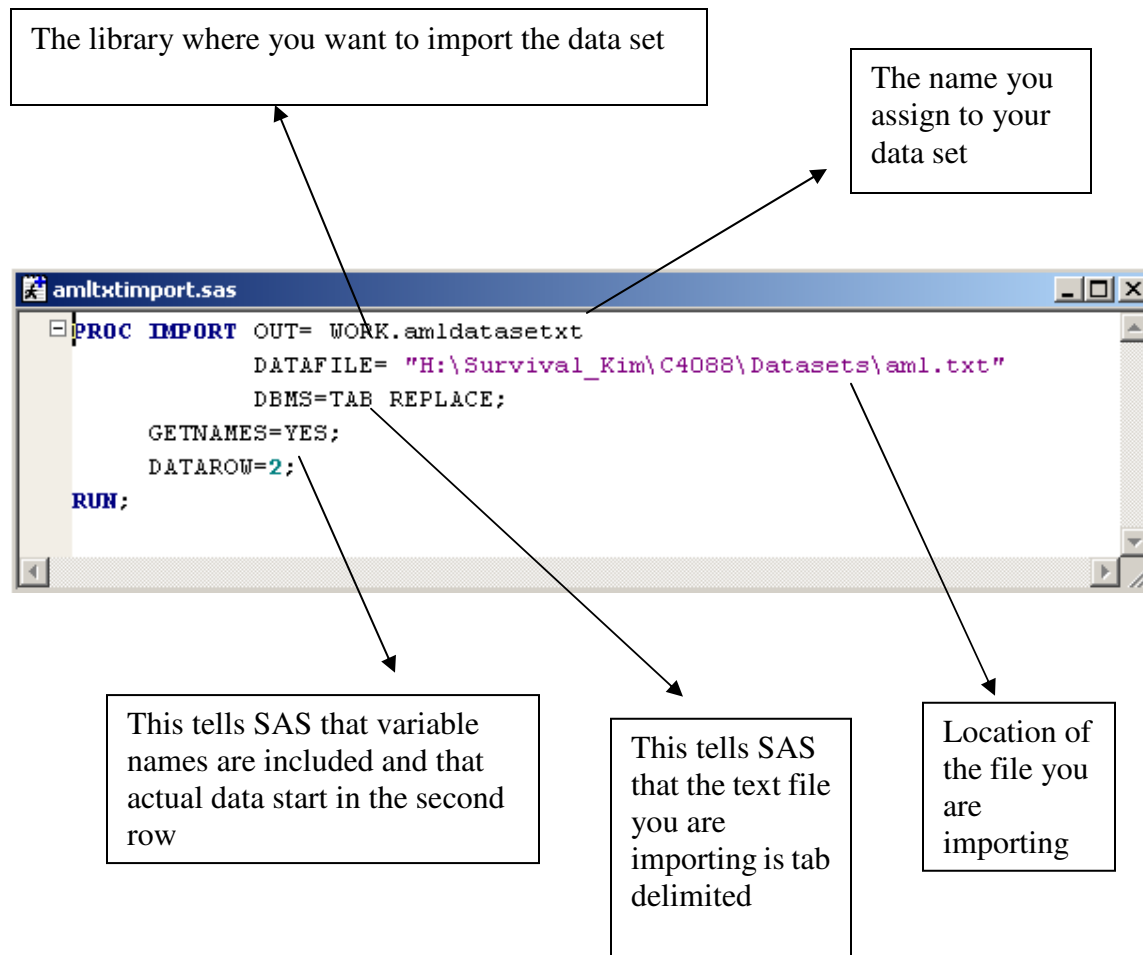
Another way to import a file into SAS is to start the program, and in the editor window, type in the code that tells it to import the txt file (proc import command). The easy way to find out how this code looks like is to select **Next** button in this step of the import using the Import Wizard. A window similar to the one below will appear:



If you have done everything successfully, the statement “WORK.amldataset.txt was successfully created” should appear in your Log window. Go to the Explorer window. You should be able to find your data set in the Work library (or the library you have chosen for this assignment). Double click on it. You should now be able to see it.

4)

To look up the proc import command you have saved earlier, select the Editor window in SAS and open the code from the location you have chosen. You should be able to see the window with the following code:



Now try to import the text file using the proc import command! Open SAS as usual, type the above code in. Make necessary adjustments if you are using different library, names, etc. You should get the same result as the one you got from the Import Wizard.

SAS Textbook Examples for Chapter 2: Nonparametric Methods

*SAS library is the location where your data set is stored. Before you use a data set you need to type in your own path for the location;

```
libname mylib 'H:\Survival_Kim\C4088\Datasets';
```

*Making a copy of the original data set is highly recommended, in particular for beginners. This prevents us from unwillingly altering the original data set. Note that a SAS data set name has two parts, for example: WORK.aml or mylib.aml. The first part of the name is library reference and the second part is the unique name you have assigned to your data set. Unless you specify library reference, SAS uses WORK library by default. The following code will create a copy of the aml data set called aml_copy. Both data sets are stored in the mylib library;

```
options ls=80 nocenter nodate; /* ls=80 specifies the maximum length of output lines
= 80, nocenter makes the output left-justified, and nodate requests that the date
should not be printed */
```

```
data mylib.aml_copy; set mylib.aml; run;
```

```
proc contents data=mylib.aml_copy; *Checking the contents of the aml_copy data set.
This line of code will output the list of all variables in the aml_copy data set
along with some additional information on the variables (e.g. whether they are
character or numeric);
run;
```

```
*Create a subset of the aml_copy data set where group=1 only and call it aml_1;
data mylib.aml_1; set mylib.aml_copy;
if group ne 1 then delete;
run;
```

```
*Check if the newly created subset is correct;
proc contents data=mylib.aml_1;
run;
```

*Fig 2.1, page 27;

*Get a survival curve only for group 1, considering all the observations as exact. This can be done by providing the time statement with only the time variable (weeks). Plots = (s) requests the survivor function;

```
proc lifetest plots=(s) data=mylib.aml_1;
time weeks;
title 'The Empirical Survivor Function of the AML Data ';
title2 'maintained group, assuming no censored data';
label weeks ='time until relapse (in weeks)';
run;
```

<Edited output>

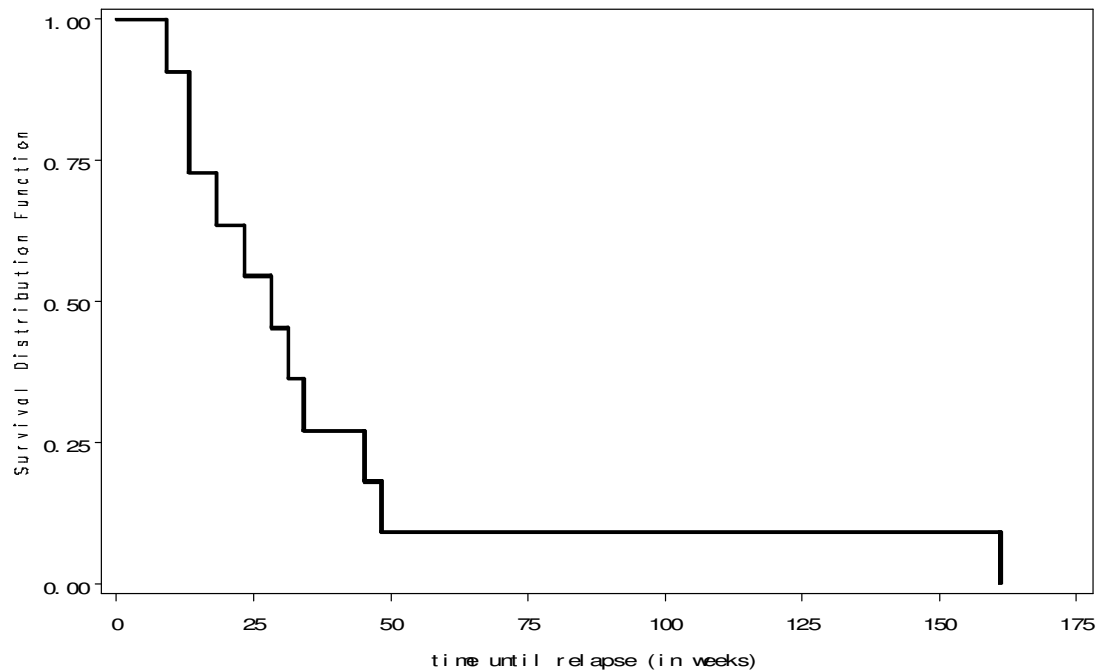
Product-Limit Survival Estimates

weeks	Survival	Failure	Survival Standard Error	Number Failed	Number Left
0.000	1.0000	0	0	0	11
9.000	0.9091	0.0909	0.0867	1	10
13.000	.	.	.	2	9
13.000	0.7273	0.2727	0.1343	3	8
18.000	0.6364	0.3636	0.1450	4	7
23.000	0.5455	0.4545	0.1501	5	6
28.000	0.4545	0.5455	0.1501	6	5
31.000	0.3636	0.6364	0.1450	7	4
34.000	0.2727	0.7273	0.1343	8	3
45.000	0.1818	0.8182	0.1163	9	2
48.000	0.0909	0.9091	0.0867	10	1
161.000	0	1.0000	0	11	0

Mean	Standard Error
38.455	12.848

The Empirical Survivor Function of the AML Data

naïve method group, assuming no censored data



***Fig 2.3, page 36;**

*Get a survival curve only for group 1, taking into account censored and uncensored observations. status(0) tells SAS that 0 represents right-censored and 1 exact observations;

```
proc lifetest plots=(s) data=mylib.aml_1;
time weeks*status(0);
title 'K-M Survival Curve for the AML Data';
title2 'maintained group';
label weeks ='time until relapse (in weeks)';
run;
```

<Edited output>

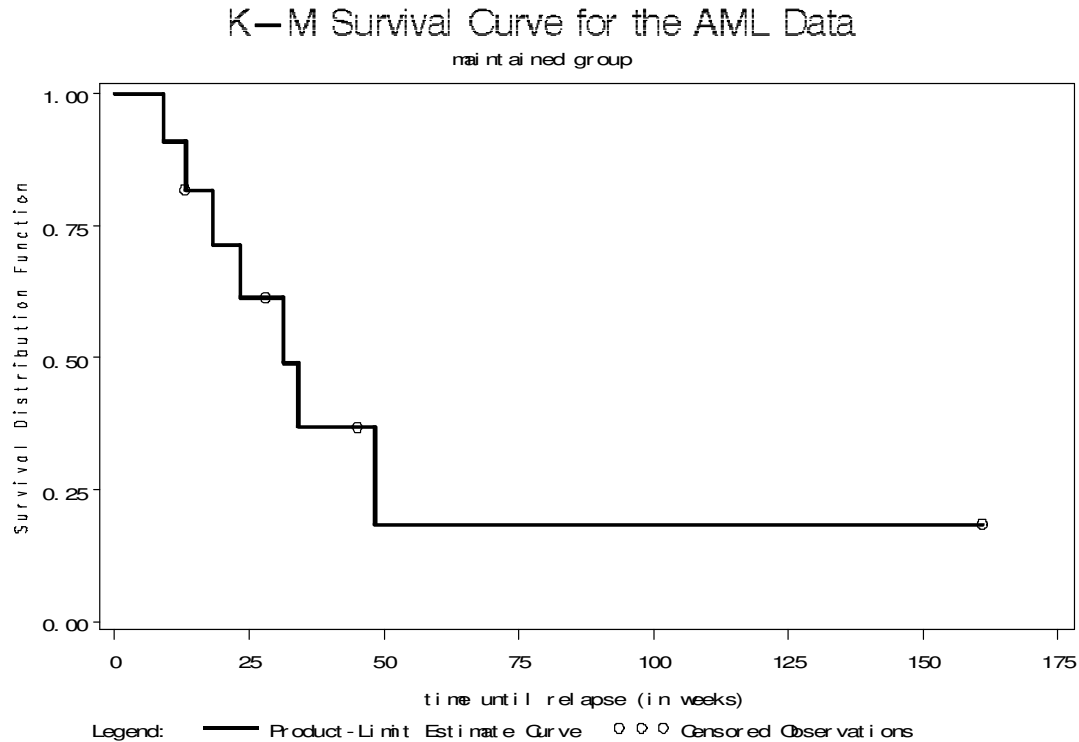
Product-Limit Survival Estimates

weeks	Survival	Failure	Survival Standard Error	Number Failed	Number Left
0.000	1.0000	0	0	0	11
9.000	0.9091	0.0909	0.0867	1	10
13.000	0.8182	0.1818	0.1163	2	9
13.000*	.	.	.	2	8
18.000	0.7159	0.2841	0.1397	3	7
23.000	0.6136	0.3864	0.1526	4	6
28.000*	.	.	.	4	5
31.000	0.4909	0.5091	0.1642	5	4
34.000	0.3682	0.6318	0.1627	6	3
45.000*	.	.	.	6	2
48.000	0.1841	0.8159	0.1535	7	1
161.000*	.	.	.	7	0

NOTE: The marked survival times are censored observations.

Mean	Standard Error
31.843	4.891

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.



***Fig 2.4, page 40 and second log-rank test on page 44;**

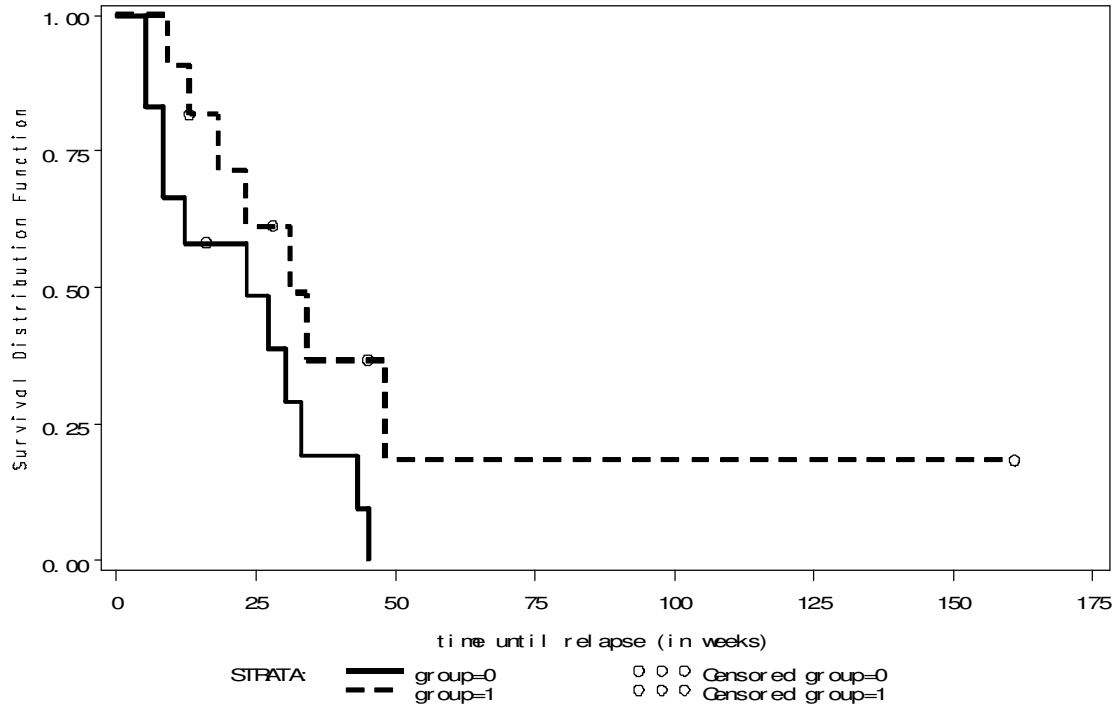
*Comparison of survival curves for group 1 and group 2. By default SAS produces two curves in different colors. If we are printing in black and white, we need to tell SAS to use different line patterns for the two curves. We do this by using symbol command where "l" describes the line pattern and "w" the width of the lines. The strata statement indicates that we are comparing different groups;

```

symbol1 c=black l=1 w=3;
symbol2 c=black l=3 w=3;
proc lifetest plots=(s) data=mylib.aml_copy;
time weeks*status(0);
strata group; *Specifies the set of variables defining the strata;
title 'Comparison of K-M Survival Curves for the AML data';
label weeks ='time until relapse (in weeks)';
run;

```

Comparison of K-M Survival Curves for the AML data



Summary of the Number of Censored and Uncensored Values

Stratum	group	Total	Failed	Censored	Percent		
					Censored	Mean	Standard Error
1	0	12	11	1	8.33	22.708	4.385
2	1	11	7	4	36.36	31.843	4.891

Test of Equality over Strata

Test	Chi-Square	DF	Pr >
			Chi-Square
Log-Rank	3.3964	1	0.0653

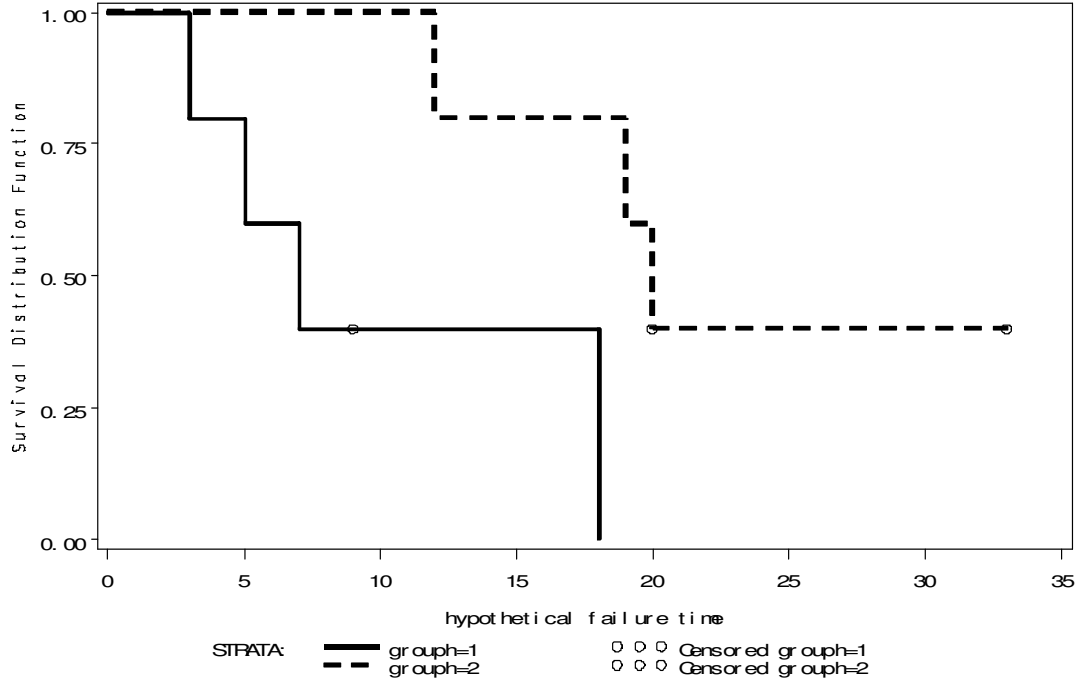
***Comparison of survival curves for group 1 and group 2 in the fictitious data;**

```
symbol1 c=black l=1 w=3;
symbol2 c=black l=3 w=3;
data mylib.fictxls_copy; set mylib.fictxls; run;
proc contents data=mylib.fictxls_copy;
run;
```

***First log-rank test on page 44;**

```
proc lifetest plots=(s) data=mylib.fictxls_copy;
time hypdata*cen (0);
strata group;
title 'K-M Survival Curves for group 1 and 2';
label hypdata ='hypothetical failure time';
run;
```

K-M Survival Curves for group 1 and 2



Summary of the Number of Censored and Uncensored Values

Stratum	group	Total	Failed	Censored	Percent		
					Censored	Mean	Standard Error
1	1	5	4	1	20.00	10.2000	3.3530
2	2	5	3	2	40.00	18.2000	1.7111

Total		10	7	3	30.00		

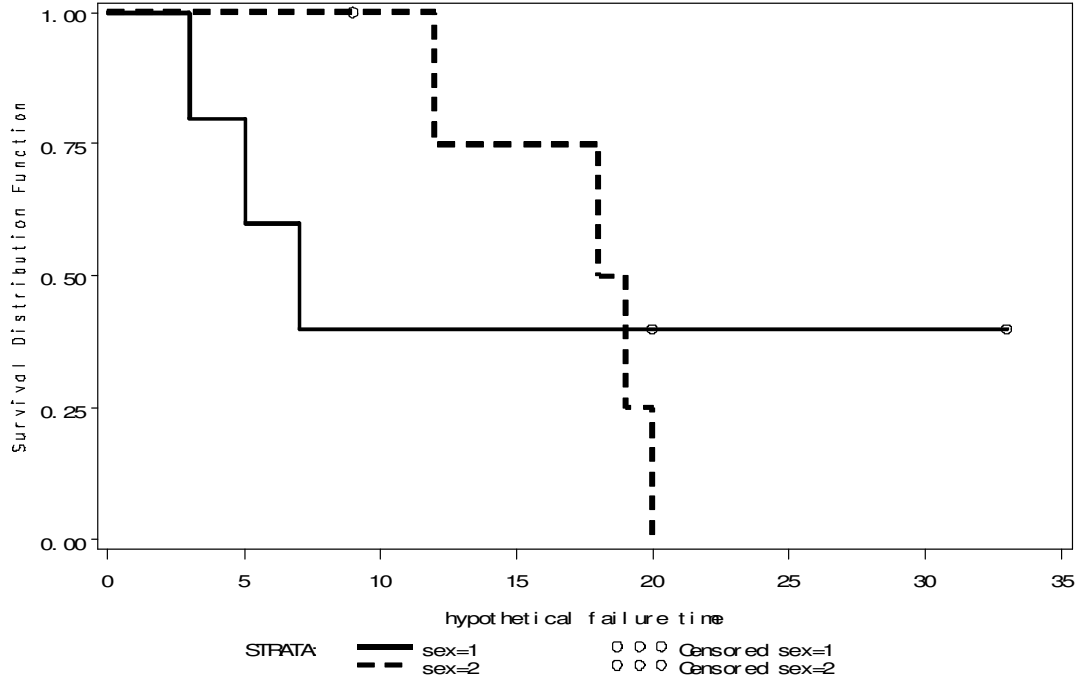
Test of Equality over Strata

Test	Chi-Square	DF	Pr > Chi-Square
Log-Rank	5.1972	1	0.0226

*Log-rank test on page 48;

```
proc lifetest plots=(s) data=mylib.fictxls_copy;
time hypdata*cen (0);
strata sex;
test group; /* testing the equality of survival curves over groups */
title 'K-M Survival Curves for group 1 and 2, stratified by sex';
label hypdata ='hypothetical failure time';
run;
```

K-M Survival Curves for group 1 and 2, stratified by sex



Variable	Test Statistic	Standard Deviation	Chi-Square	Pr > Chi-Square
grouph	1.9833	1.0592	3.5061	0.0611

SAS Textbook Examples for Chapter 3: Parametric Methods

*SAS library is the location where your data set is stored. Before you use a data set you need to type in your own path for the location;

```
libname mylib 'H:\Survival_Kim\C4088\Datasets';
```

```
options ls=80 nocenter nodate;
```

```
data mylib.aml_copy; set mylib.aml; run;
```

*Create a subset of the aml where group=1 only;

```
data mylib.aml_1; set mylib.aml_copy; if group ne 1 then delete; run;
```

*The following code produces SAS equivalent of exponential fit, page 75 – top of page 76;

```
proc lifereg data=mylib.aml_1; model weeks*status(0) = / d = exponential; run;
```

<Edited output>

Model Information

Data Set	mylib.AML_1	
Dependent Variable	Log(weeks)	weeks
Censoring Variable	status	status
Censoring Value(s)	0	
Number of Observations	11	
Noncensored Values	7	
Right Censored Values	4	
Left Censored Values	0	
Interval Censored Values	0	
Name of Distribution	Exponential	
Log Likelihood	-14.09064556	

Analysis of Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	4.1015	0.3780	3.3607	4.8423	117.75	<.0001
Weibull Scale	1	60.4286	22.8399	28.8083	126.7553		

*The following code produces SAS equivalent of Weibull fit, page 78;

```
proc lifereg data=mylib.aml_1;
```

```
model weeks*status(0) = / d = Weibull; /* Weibull is the default */
```

```
run;
```

<Edited output>

Model Information

```
Data Set          mylib.AML_1
Dependent Variable  Log(weeks)    weeks
Censoring Variable  status        status
Censoring Value(s)  0
Number of Observations  11
Noncensored Values  7
Right Censored Values  4
Left Censored Values  0
Interval Censored Values  0
Name of Distribution  Weibull
Log Likelihood      -14.08436767
```

Analysis of Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	4.0997	0.3665	3.3814	4.8180	125.14	<.0001
Scale	1	0.9691	0.2686	0.5630	1.6682		
Weibull Scale	1	60.3229	22.1072	29.4125	123.7177		
Weibull Shape	1	1.0319	0.2859	0.5994	1.7762		

***The following code produces SAS equivalent of log-logistic fit, page 79;**
proc lifereg data=mylib.aml_1; model weeks*status(0) = / d = llogistic; run;

<Edited output>

Model Information

```
Data Set          mylib.AML_1
Dependent Variable  Log(weeks)    weeks
Censoring Variable  status        status
Censoring Value(s)  0
Number of Observations  11
Noncensored Values  7
Right Censored Values  4
Left Censored Values  0
Interval Censored Values  0
Name of Distribution  LLogistic
Log Likelihood      -12.5040003
```

Analysis of Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	3.5146	0.3061	2.9146	4.1146	131.81	<.0001
Scale	1	0.5423	0.1722	0.2910	1.0104		

*The following code produces SAS equivalent of pp. 82 - 84;

```
*Model 1 on page 82;
proc lifereg data=mylib.aml_copy;
model weeks*status(0) = / d = Weibull;
run;
```

Model Information

```
Data Set          mylib.AML_COPY
Dependent Variable  Log(weeks)    weeks
Censoring Variable  status        status
Censoring Value(s)  0
Number of Observations  23
Noncensored Values  18
Right Censored Values  5
Left Censored Values  0
Interval Censored Values  0
Name of Distribution  Weibull
Log Likelihood      -30.7995164
```

Analysis of Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	3.6425	0.2171	3.2170	4.0680	281.56	<.0001
Scale	1	0.9119	0.1546	0.6542	1.2712		
Weibull Scale	1	38.1868	8.2895	24.9538	58.4372		
Weibull Shape	1	1.0966	0.1859	0.7867	1.5287		

```
*Model 2 on page 82;
proc lifereg data=mylib.aml_copy;
model weeks*status(0) = group/ d = Weibull;
run;
```

```
Name of Distribution  Weibull
Log Likelihood      -28.14249237
```

Type III Analysis of Effects

Effect	DF	Chi-Square	Pr > ChiSq
group	1	5.9031	0.0151

Analysis of Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	3.1797	0.2406	2.7082	3.6512	174.72	<.0001
group	1	0.9293	0.3825	0.1797	1.6790	5.90	0.0151
Scale	1	0.7910	0.1410	0.5578	1.1216		
Weibull Shape	1	1.2643	0.2253	0.8915	1.7929		

*Model 3 on page 83; *Sorting is necessary as the "by" option in the proc lifereg requires the group to be listed in the increasing order;

```
proc sort data=mylib.aml_copy;
by group; /* sorts the data by the increasing order of group values */
run;
proc lifereg data=mylib.aml_copy;
model weeks*status(0) = / d = Weibull;
by group;
run;
```

group=0

```
Data Set          mylib.AML_COPY
Dependent Variable      Log(weeks)   weeks
Censoring Variable      status     status
Censoring Value(s)      0
Number of Observations  12
Noncensored Values      11
Right Censored Values   1
Left Censored Values    0
Interval Censored Values 0
Name of Distribution     Weibull
Log Likelihood          -13.38464793
```

Analysis of Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	3.2222	0.1983	2.8335	3.6109	264.00	<.0001
Scale	1	0.6355	0.1551	0.3939	1.0253		
Weibull Scale	1	25.0832	4.9744	17.0051	36.9989		
Weibull Shape	1	1.5736	0.3841	0.9753	2.5390		

group=1

```
Data Set          mylib.AML_COPY
Dependent Variable      Log(weeks)   weeks
Censoring Variable      status     status
Censoring Value(s)      0
Number of Observations  11
Noncensored Values      7
Right Censored Values   4
Left Censored Values    0
Interval Censored Values 0
Name of Distribution     Weibull
Log Likelihood          -14.08436767
```

Analysis of Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	4.0997	0.3665	3.3814	4.8180	125.14	<.0001
Scale	1	0.9691	0.2686	0.5630	1.6682		
Weibull Scale	1	60.3229	22.1072	29.4125	123.7177		
Weibull Shape	1	1.0319	0.2859	0.5994	1.7762		

```
*log-logistic fit;
proc lifereg data=mylib.aml_copy;
model weeks*status(0) = group/ d = llogistic;
run;
```

```
Name of Distribution          LLogistic
Log Likelihood              -26.97351055
```

Type III Analysis of Effects

Effect	DF	Wald	
		Chi-Square	Pr > ChiSq
group	1	2.3622	0.1243

Analysis of Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	2.8985	0.2673	2.3747	3.4223	117.61	<.0001
group	1	0.6045	0.3933	-0.1664	1.3753	2.36	0.1243
Scale	1	0.5133	0.0983	0.3526	0.7472		

```
*log-normal fit;
proc lifereg data=mylib.aml_copy;
model weeks*status(0) = group/ d = lnormal;
run;
```

```
Name of Distribution          Lognormal
Log Likelihood              -26.54845961
```

Type III Analysis of Effects

Effect	DF	Wald	
		Chi-Square	Pr > ChiSq
group	1	3.6287	0.0568

Analysis of Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	2.8544	0.2539	2.3567	3.3521	126.37	<.0001
group	1	0.7245	0.3803	-0.0209	1.4699	3.63	0.0568
Scale	1	0.8647	0.1466	0.6203	1.2055		

SAS Textbook Examples for Chapter 4: Regression Models

```
libname mylib 'H:\Survival_Kim\C4088\Datasets';
options ls=80 nocenter nodate;
```

*The following code produces SAS equivalent of the results on pages 108, 109, and 112;

```
proc lifereg data=mylib.Motorette_copy;
model time*status(0) = x/ d = weibull;
run;
```

```
proc lifereg data=mylib.Motorette_copy;
model time*status(0) = x/ d = exponential;
run;
```

```
proc lifereg data=mylib.Motorette_copy;
model time*status(0) = x/ d = lnormal;
run;
```

```
proc lifereg data=mylib.Motorette_copy;
model time*status(0) = x/ d = llogistic;
run;
```

<Edited output>

Name of Distribution		Weibull					
Log Likelihood		-22.90155531					
Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	-11.8269	1.9567	-15.6621	-7.9918	36.53	<.0001
x	1	8.9971	0.9009	7.2314	10.7628	99.74	<.0001
Scale	1	0.3592	0.0793	0.2331	0.5536		
Weibull Shape	1	2.7838	0.6143	1.8063	4.2902		

Name of Distribution		Exponential					
Log Likelihood		-30.40796771					
Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	-8.9187	5.4911	-19.6811	1.8436	2.64	0.1043
x	1	7.7895	2.5362	2.8186	12.7604	9.43	0.0021
Scale	0	1.0000	0.0000	1.0000	1.0000		
Weibull Shape	0	1.0000	0.0000	1.0000	1.0000		

Name of Distribution		Lognormal					
Log Likelihood		-24.40199883					
Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	-10.4389	2.7461	-15.8212	-5.0566	14.45	0.0001
x	1	8.2972	1.2707	5.8067	10.7877	42.64	<.0001
Scale	1	0.6006	0.1102	0.4192	0.8605		

SAS Textbook Examples for Chapter 5: Cox Proportional Hazards Models

*SAS library is the location where your data set is stored. Before you use a data set you need to type in your own path for the location;

```
libname mylib 'H:\Survival_Kim\C4088\Datasets';
```

```
options ls=80 nocenter nodate;
```

```
data CNS2new;
```

```
set mylib.CNS2;
```

```
if AGE le 60 then AGE60 = 1;
```

```
else AGE60 = 0;
```

```
SEX_AGE60 = SEX*AGE60; /* Creation of interaction term */
```

```
GROUP_AGE60 = GROUP*AGE60;
```

```
LESSING_AGE60 = LESSING*AGE60;
```

```
run;
```

```
proc phreg data=CNS2new; /* cns2.coxint1 */
```

```
model B3TODEATH*STATUS(0) = KPS_PRE_ GROUP SEX AGE60 LESSING CHEMOPRIOR SEX_AGE60 LESSING_AGE60  
GROUP_AGE60;
```

```
run;
```

Variable	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio
KPS_PRE_	1	-0.04719	0.01403	11.3151	0.0008	0.954
GROUP	1	1.97374	0.70676	7.7989	0.0052	7.198
SEX	1	-3.27950	0.88833	13.6291	0.0002	0.038
AGE60	1	-0.37769	0.68572	0.3034	0.5818	0.685
LESSING	1	1.63824	0.67057	5.9684	0.0146	5.146
CHEMOPRIOR	1	1.00340	0.53760	3.4836	0.0620	2.728
SEX_AGE60	1	2.82729	0.92128	9.4179	0.0021	16.900
LESSING_AGE60	1	-1.58788	0.83736	3.5959	0.0579	0.204
GROUP_AGE60	1	-1.22499	0.83795	2.1371	0.1438	0.294

```
proc phreg data=CNS2new; /* cns2.coxint2 */
```

```
model B3TODEATH*STATUS(0) = KPS_PRE_ GROUP SEX LESSING CHEMOPRIOR;
```

```
run;
```

Variable	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio
KPS_PRE_	1	-0.03720	0.01236	9.0504	0.0026	0.963
GROUP	1	1.13168	0.43116	6.8894	0.0087	3.101
SEX	1	-2.58019	0.76703	11.3155	0.0008	0.076
AGE60	1	-1.34561	0.50931	6.9803	0.0082	0.260
LESSING	1	0.55786	0.40279	1.9182	0.1661	1.747
CHEMOPRIOR	1	0.85012	0.51696	2.7043	0.1001	2.340

```
proc phreg data=CNS2new; /* cns2.coxint3 */
```

```
model B3TODEATH*STATUS(0) = KPS_PRE_ GROUP SEX AGE60 LESSING CHEMOPRIOR SEX_AGE60 LESSING_AGE60;
```

```
run;
```

Variable	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio
----------	----	--------------------	----------------	------------	------------	--------------

KPS_PRE_	1	-0.04378	0.01343	10.6239	0.0011	0.957
GROUP	1	1.10956	0.43348	6.5518	0.0105	3.033
SEX	1	-2.73570	0.76331	12.8451	0.0003	0.065
AGE60	1	-0.88528	0.59559	2.2094	0.1372	0.413
LESSING	1	1.35413	0.63219	4.5881	0.0322	3.873
CHEMOPRIOR	1	0.86502	0.52515	2.7132	0.0995	2.375
SEX_AGE60	1	2.42482	0.87714	7.6423	0.0057	11.300
LESSING AGE60	1	-1.23835	0.80368	2.3742	0.1234	0.290

```
proc phreg data=CNS2new; /* cns2.coxint4 */
model B3TODEATH*STATUS(0) = KPS_PRE_ GROUP SEX AGE60 LESSING CHEMOPRIOR SEX_AGE60;
run;
```

Variable	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio
KPS_PRE_	1	-0.03720	0.01236	9.0504	0.0026	0.963
GROUP	1	1.13168	0.43116	6.8894	0.0087	3.101
SEX	1	-2.58019	0.76703	11.3155	0.0008	0.076
AGE60	1	-1.34561	0.50931	6.9803	0.0082	0.260
LESSING	1	0.55786	0.40279	1.9182	0.1661	1.747
CHEMOPRIOR	1	0.85012	0.51696	2.7043	0.1001	2.340
SEX_AGE60	1	2.31523	0.87511	6.9994	0.0082	10.127

```
proc phreg data=CNS2new; /* cns2.coxint5 */
model B3TODEATH*STATUS(0) = KPS_PRE_ GROUP SEX AGE60 CHEMOPRIOR SEX_AGE60;
run;
```

Variable	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio
KPS_PRE_	1	-0.04018	0.01216	10.9266	0.0009	0.961
GROUP	1	0.95465	0.40760	5.4856	0.0192	2.598
SEX	1	-2.46462	0.77044	10.2334	0.0014	0.085
AGE60	1	-1.08691	0.47091	5.3274	0.0210	0.337
CHEMOPRIOR	1	0.79308	0.51005	2.4177	0.1200	2.210
SEX_AGE60	1	2.16438	0.87098	6.1752	0.0130	8.709

```
proc phreg data=CNS2new;
model B3TODEATH*STATUS(0) = KPS_PRE_ GROUP SEX AGE60 SEX_AGE60; /* cns2.coxint6 */
run;
```

Variable	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio
KPS_PRE_	1	-0.03071	0.01029	8.9128	0.0028	0.970
GROUP	1	1.14262	0.37841	9.1174	0.0025	3.135
SEX	1	-2.09231	0.70097	8.9095	0.0028	0.123
AGE60	1	-1.02869	0.45546	5.1011	0.0239	0.357
SEX_AGE60	1	2.11076	0.85295	6.1240	0.0133	8.255

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	27.1678	5	<.0001
Score	27.8909	5	<.0001
Wald	24.2267	5	0.0002

```
proc phreg data=CNS2new;
```

```

model B3TODEATH*STATUS(0) = KPS_PRE_ GROUP SEX CHEMOPRIOR; /* cns2.cox1 */
run;

```

Variable	DF	Parameter		Chi-Square	Pr > ChiSq	Hazard Ratio
		Estimate	Standard Error			
KPS_PRE_	1	-0.04292	0.01164	13.6003	0.0002	0.958
GROUP	1	0.55414	0.38764	2.0435	0.1529	1.740
SEX	1	-1.07081	0.45586	5.5177	0.0188	0.343
CHEMOPRIOR	1	0.71548	0.47595	2.2598	0.1328	2.045

```

proc phreg data=CNS2new;
model B3TODEATH*STATUS(0) = KPS_PRE_ SEX CHEMOPRIOR; /* cns2.cox2 */
run;

```

Variable	DF	Parameter		Chi-Square	Pr > ChiSq	Hazard Ratio
		Estimate	Standard Error			
KPS_PRE_	1	-0.04873	0.01101	19.5897	<.0001	0.952
SEX	1	-1.19686	0.44617	7.1959	0.0073	0.302
CHEMOPRIOR	1	0.99490	0.43873	5.1424	0.0233	2.704

```

proc phreg data=CNS2new;
model B3TODEATH*STATUS(0) = KPS_PRE_ GROUP SEX; /* cns2.cox3 */
run;

```

Variable	DF	Parameter		Chi-Square	Pr > ChiSq	Hazard Ratio
		Estimate	Standard Error			
KPS_PRE_	1	-0.03454	0.01009	11.7273	0.0006	0.966
GROUP	1	0.77223	0.35427	4.7514	0.0293	2.165
SEX	1	-0.79580	0.40990	3.7692	0.0522	0.451

```

proc phreg data=CNS2new;
strata GROUP;
model B3TODEATH*STATUS(0) = KPS_PRE_ SEX AGE60 SEX_AGE60; /* cns2.coxint7 */
run;

```

Variable	DF	Parameter		Chi-Square	Pr > ChiSq	Hazard Ratio
		Estimate	Standard Error			
KPS_PRE_	1	-0.03249	0.01080	9.0436	0.0026	0.968
SEX	1	-2.16871	0.71795	9.1247	0.0025	0.114
AGE60	1	-1.11154	0.47607	5.4514	0.0196	0.329
SEX AGE60	1	2.22616	0.87614	6.4561	0.0111	9.264

```

proc phreg data=CNS2new;
strata SEX;
model B3TODEATH*STATUS(0) = KPS_PRE_ GROUP AGE60 SEX_AGE60; /* cns2.coxint8 */
run;

```

Variable	DF	Parameter		Chi-Square	Pr > ChiSq	Hazard Ratio
		Estimate	Standard Error			
KPS_PRE_	1	-0.03321	0.01042	10.1546	0.0014	0.967
GROUP	1	1.15156	0.38146	9.1134	0.0025	3.163
AGE60	1	-0.96317	0.45315	4.5177	0.0335	0.382
SEX AGE60	1	2.16109	0.87544	6.0939	0.0136	8.681

*The following is to plot survival graphs on one graph, similar to Figure 5.4;

```

proc sort data=CNS2new; /* sorting is necessary in order to use a BY statement
                           in later programs */
by GROUP;
run;

*BY Statement is used in the following proc phreg to obtain separate baseline survival
curves in groups defined by the BY variables;
proc phreg data=CNS2new;
model B3TODEATH*STATUS(0) = KPS_PRE_ SEX AGE60 SEX_AGE60;
baseline out=out survival=survival loglogs=lcumhaz; /* baseline statement creates a SAS
                                                         data set that contains output statistics for each stratum */
by GROUP;
run;

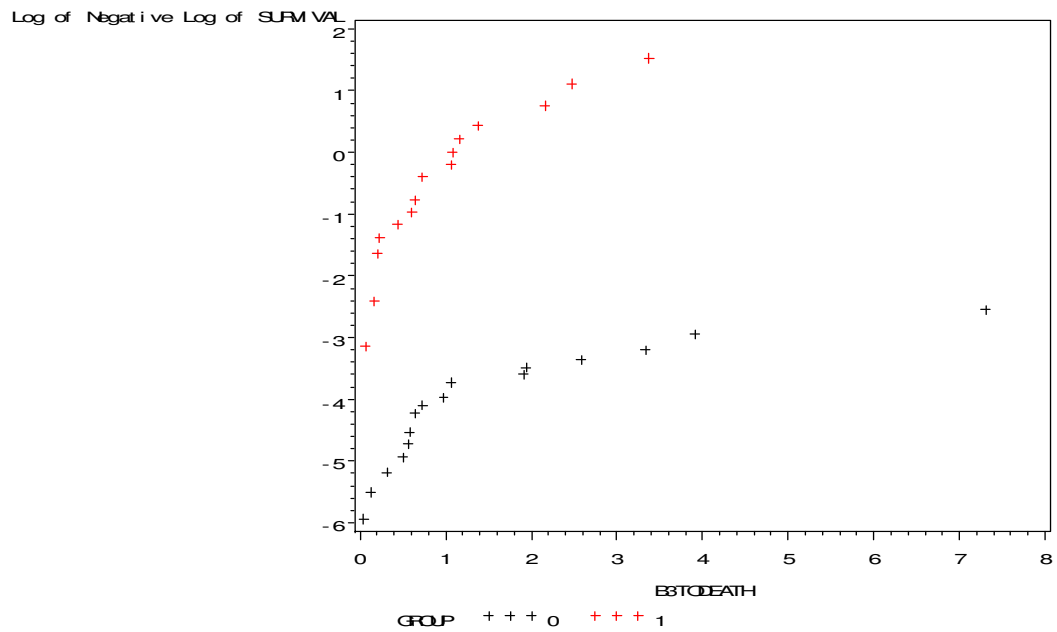
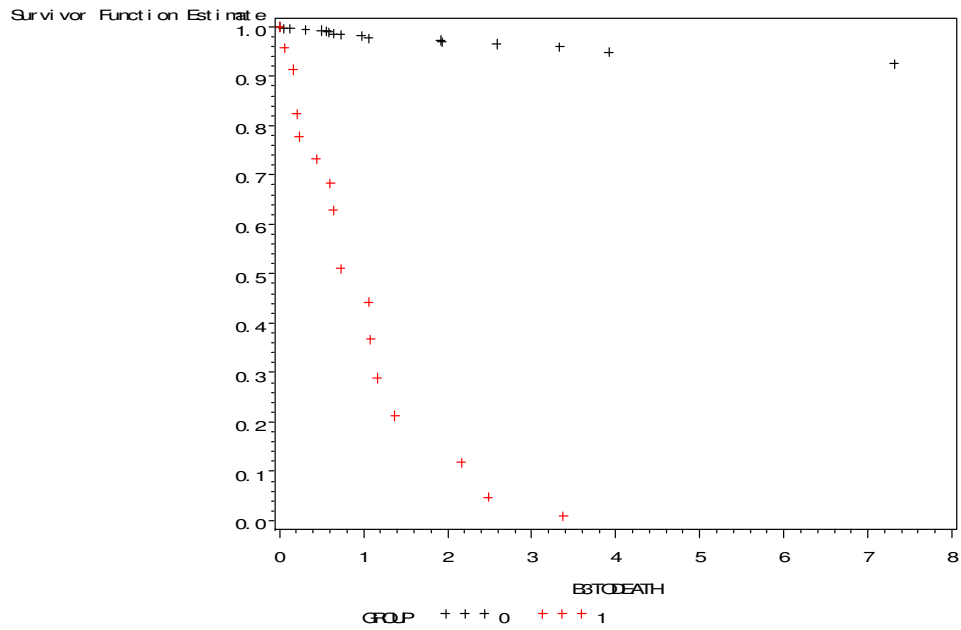
proc gplot data=out;
plot survival*B3TODEATH=GROUP; /* baseline survival curves for
                                 maintained and nonmaintained */
plot lcumhaz*B3TODEATH=GROUP;
run;

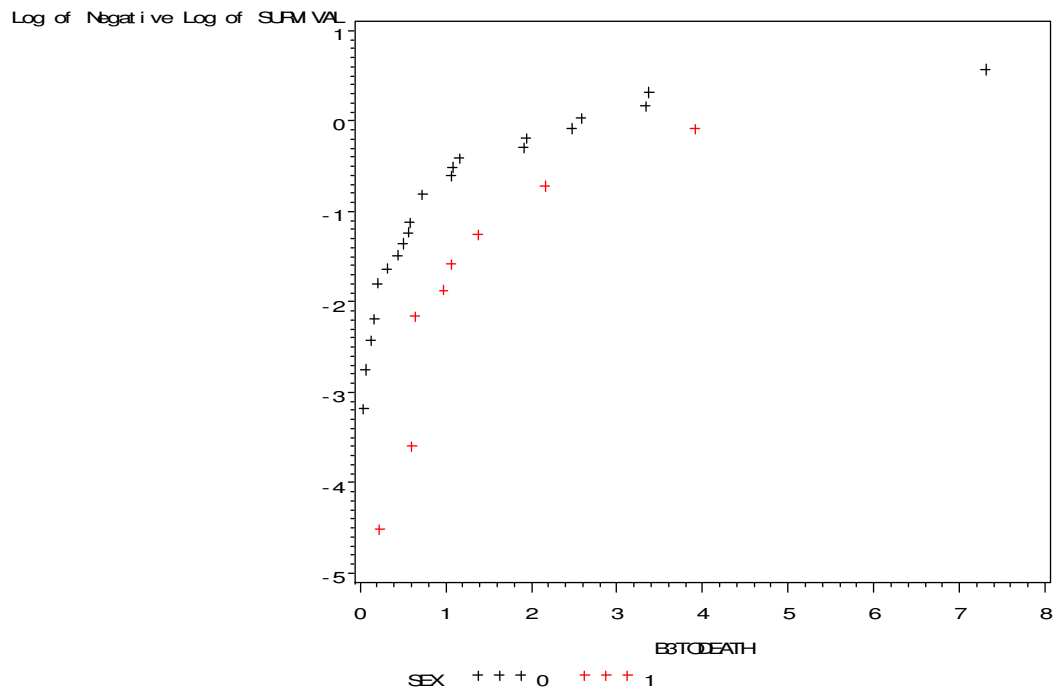
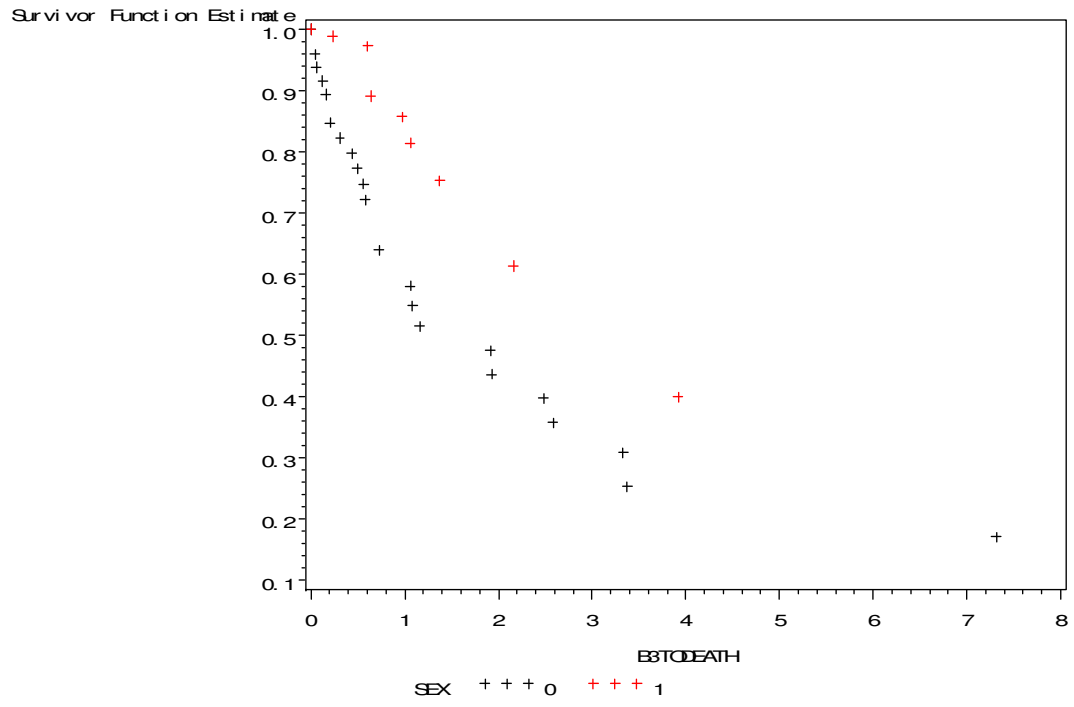
proc sort data=CNS2new;
by SEX;
run;

proc phreg data=CNS2new;
model B3TODEATH*STATUS(0) = KPS_PRE_ GROUP AGE60 SEX_AGE60;
baseline out=out1 survival=survival1 loglogs=lcumhaz1;
by SEX;
run;

proc gplot data=out1;
plot survival1*B3TODEATH=SEX; /* baseline survival curves for male and female */
plot lcumhaz1*B3TODEATH=SEX;
run;

```





SAS Textbook Examples for Chapter 6: Model Checking: Data Diagnostics

*SAS library is the location where your data set is stored. Before you use a data set you need to type in your own path for the location;

```
libname mylib 'H:\Survival_Kim\C4088\Datasets';
```

```
options ls=80 nocenter nodate;
```

```
data CNS2new;
```

```
set mylib.CNS2;
```

```
if AGE le 60 then AGE60 = 1;
```

```
else AGE60 = 0;
```

```
SEX_AGE60 = SEX*AGE60;
```

```
GROUP_AGE60 = GROUP*AGE60;
```

```
LESSING_AGE60 = LESSING*AGE60;
```

```
run;
```

*Figure 6.12 Martingale residuals to look for best functional form of the continuous covariate KPS.PRE.;

```
data temp;
```

```
set CNS2new;
```

```
dummy = 0;
```

```
run;
```

```
proc phreg data=temp noprint;
```

```
model B3TODEATH*STATUS(0) = GROUP SEX AGE60 SEX_AGE60 dummy;
```

```
output out=temp1 resmart=rr/order=data;
```

```
run;
```

```
data temp2;
```

```
merge temp temp1;
```

```
run;
```

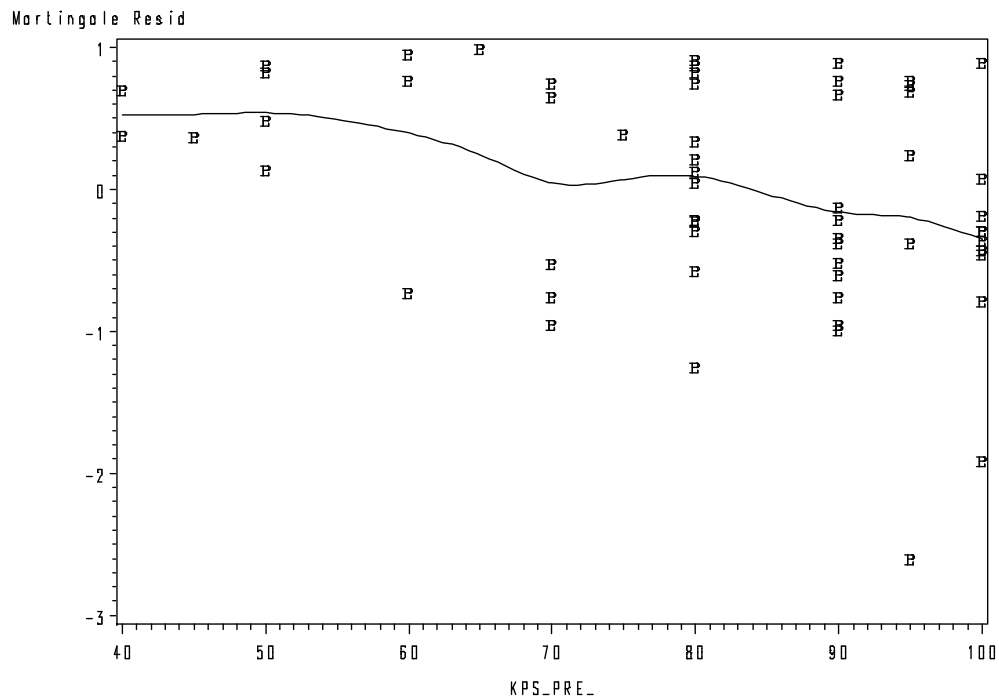
```
proc gplot;
```

```
plot rr*KPS_PRE_;
```

```
symbol i=sm60s v=j font=special;
```

```
label rr='Martingale Resid' KPS_PRE_ = 'KPS_PRE_';
```

```
run;
```



*Figure 6.13 Deviance residuals to check for outliers and Figure 6.16 The dfbetas to detect influential observations;

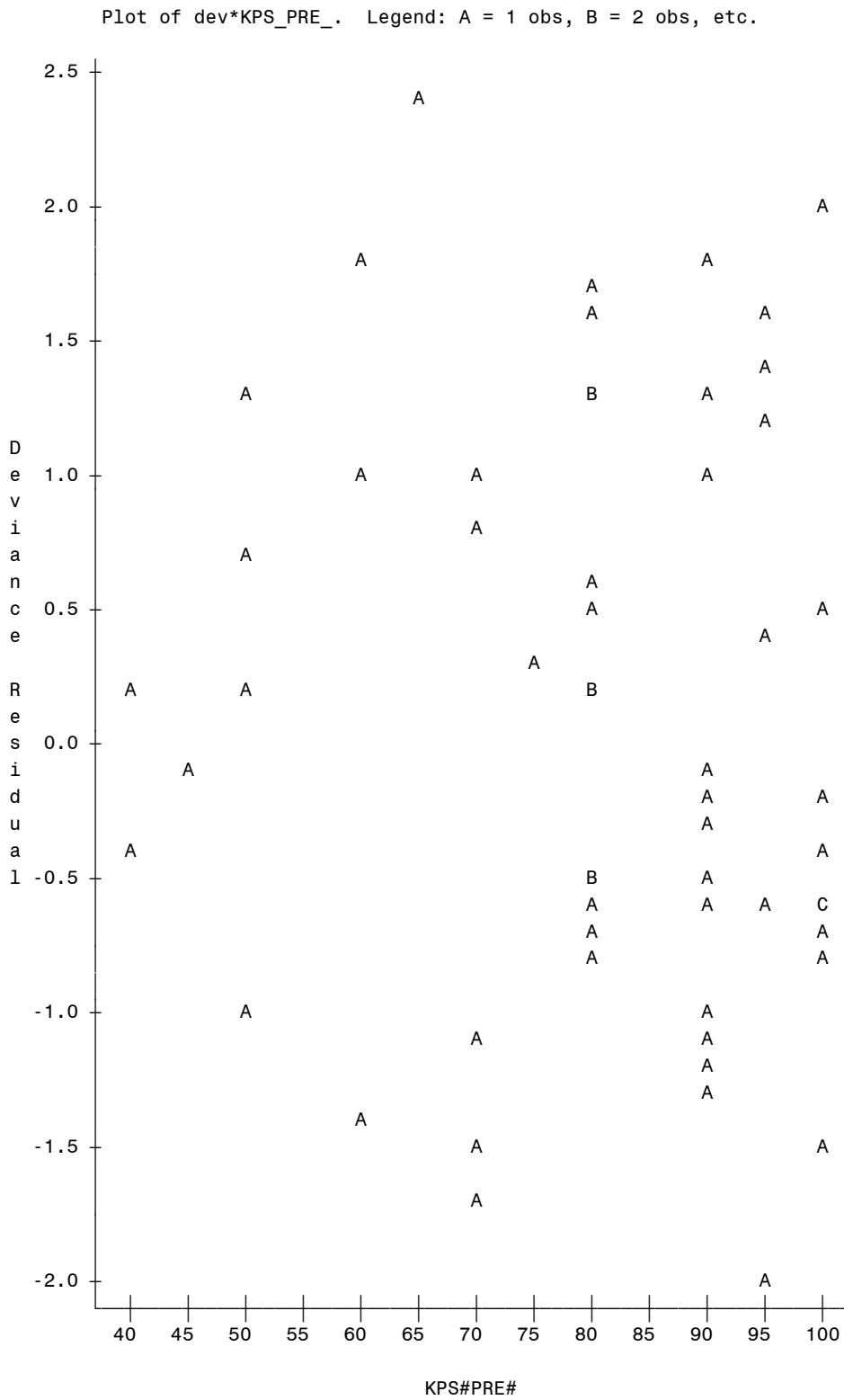
```

proc phreg data=CNS2new;
model B3TODEATH*STATUS(0) = KPS_PRE_ GROUP SEX AGE60 SEX_AGE60/ties=efron;
output out=dfbetatemp resdev=dev dfbeta = KPS_inf GROUP_inf SEX_inf AGE60_inf
SEX_AGE60_inf; /* Output data set dfbetatemp contains
                deviance residuals and dfbetas */
id PT_NUMBER;
run;

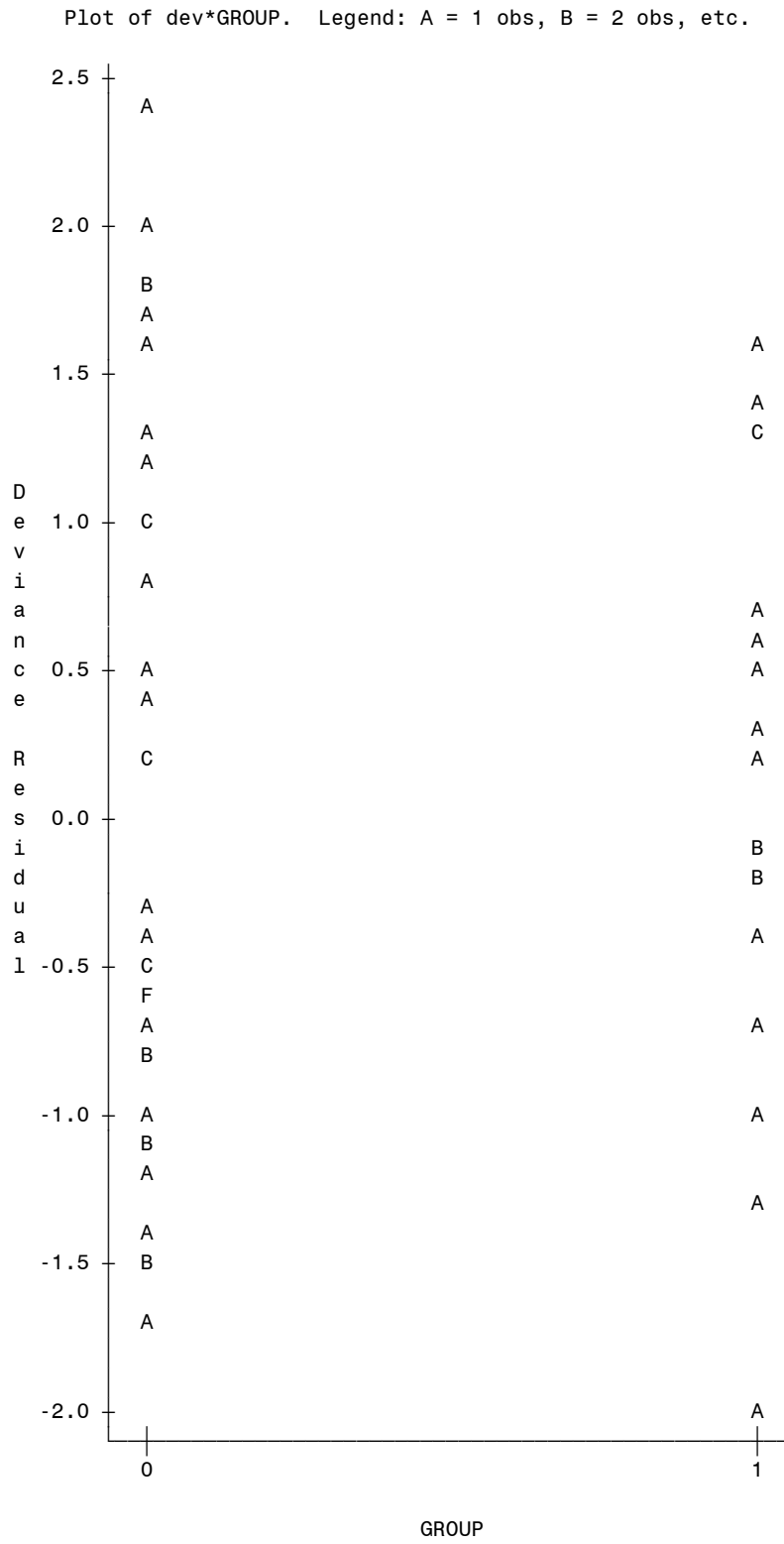
proc plot data=dfbetatemp; /* Figure 6.13 */
plot dev*KPS_PRE_;
plot dev*GROUP;
plot dev*SEX;
plot dev*AGE60;
plot dev*B3TODEATH;
run;

```

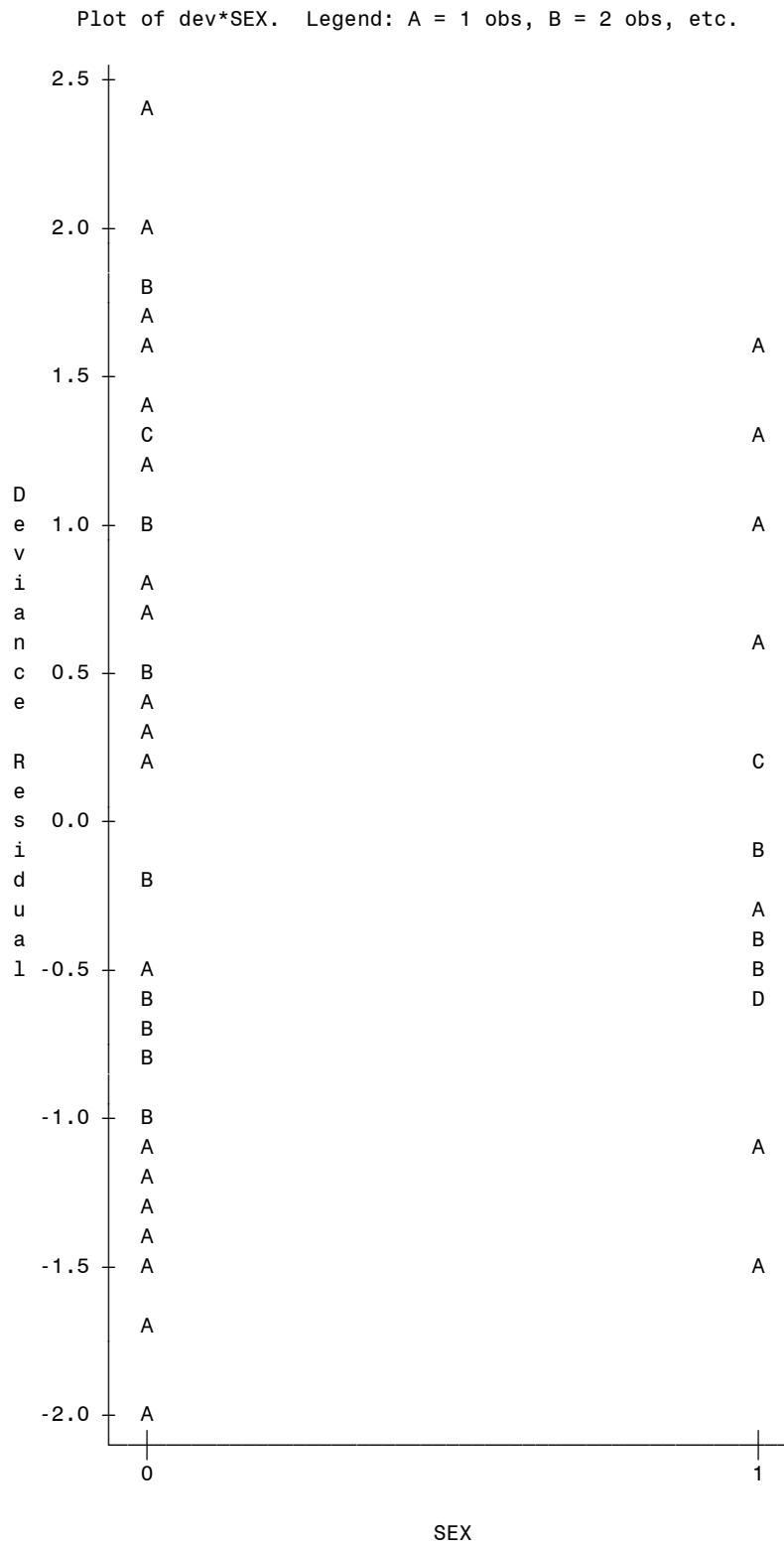
*Figure 6.13;



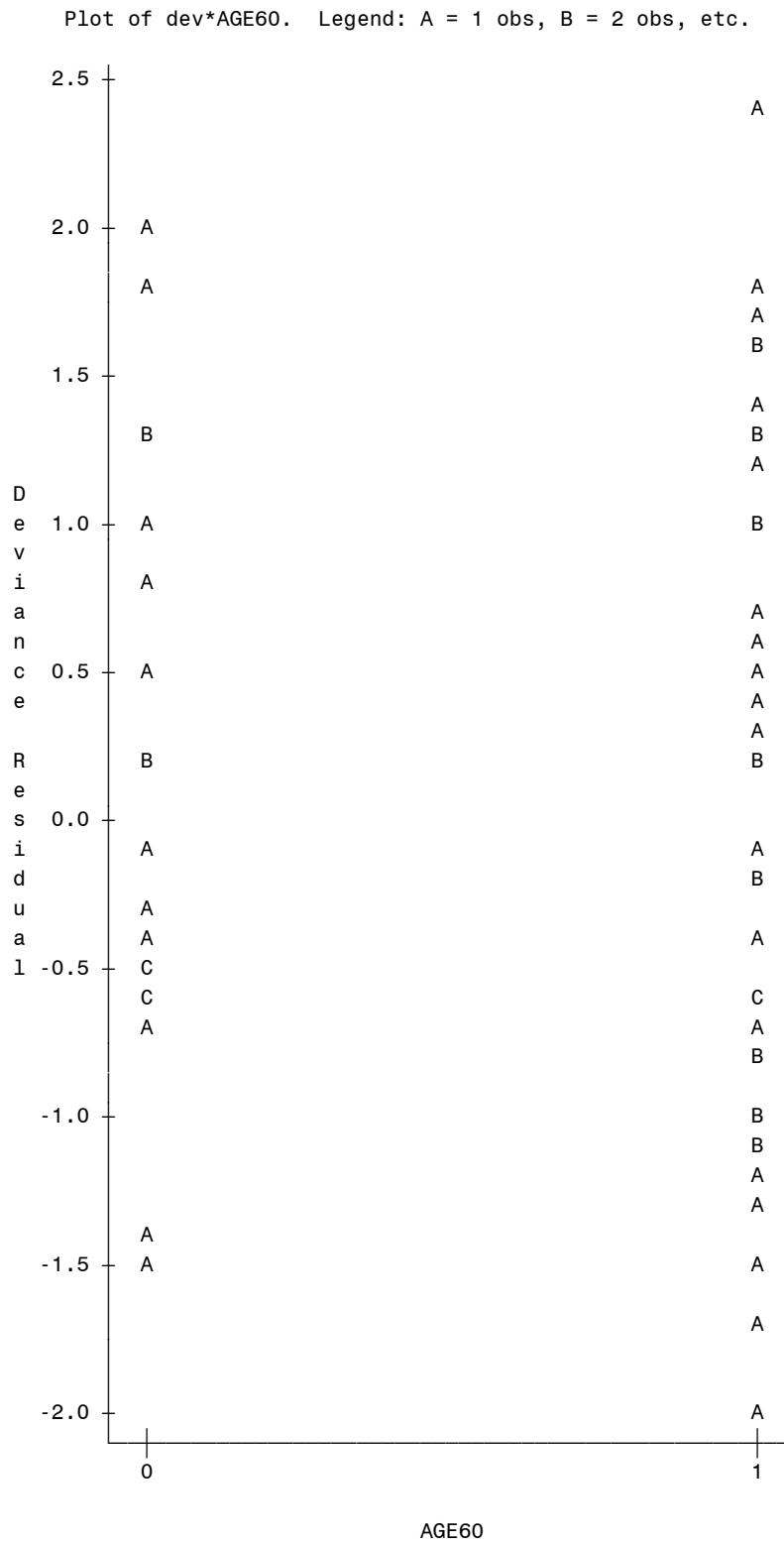
*Figure 6.13;



*Figure 6.13;

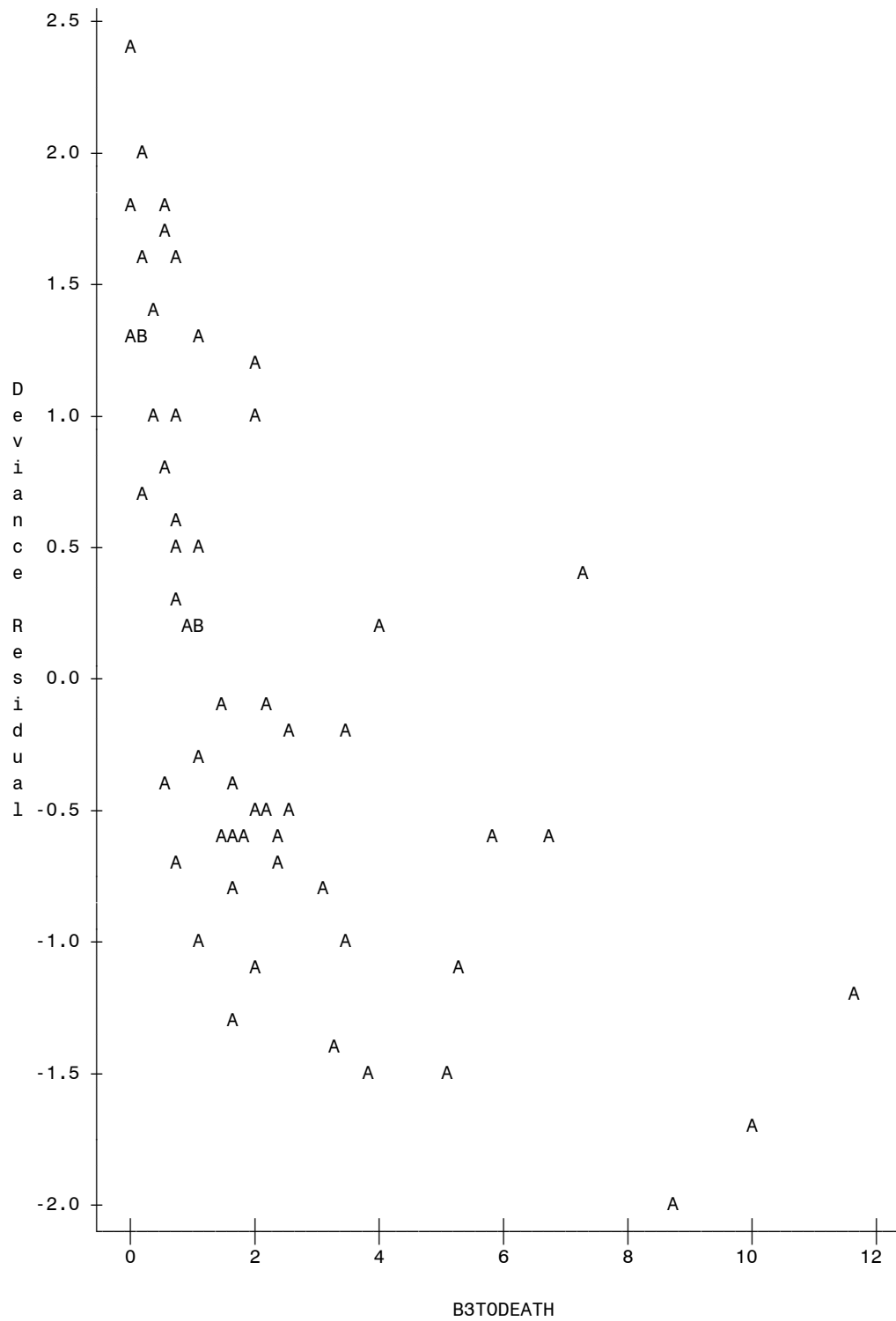


*Figure 6.13;



*Figure 6.13;

Plot of dev*B3TODEATH. Legend: A = 1 obs, B = 2 obs, etc.



```

proc sort data=dfbetatemp; by PT_NUMBER; run;

proc means noprint data=dfbetatemp;
  var KPS_inf GROUP_inf SEX_inf AGE60_inf SEX_AGE60_inf;
  by PT_NUMBER;
  output out=dfbetatemp1 sum=KPS_inf GROUP_inf SEX_inf AGE60_inf SEX_AGE60_inf;
run;

data dfbetatemp2; merge dfbetatemp1 CNS2new;
  by PT_NUMBER;
  if (first.PT_NUMBER=1); /* first.variable = 1 whenever SAS
                          encounters a distinct value from the previously
                          observed values from the specified variable, 0,
                          otherwise. */
  observation = _n_; /* Indicates the number of times SAS has
                     encountered distinct values from the specified
                     variable. Note that this is not necessarily equal
                     to the sample size */

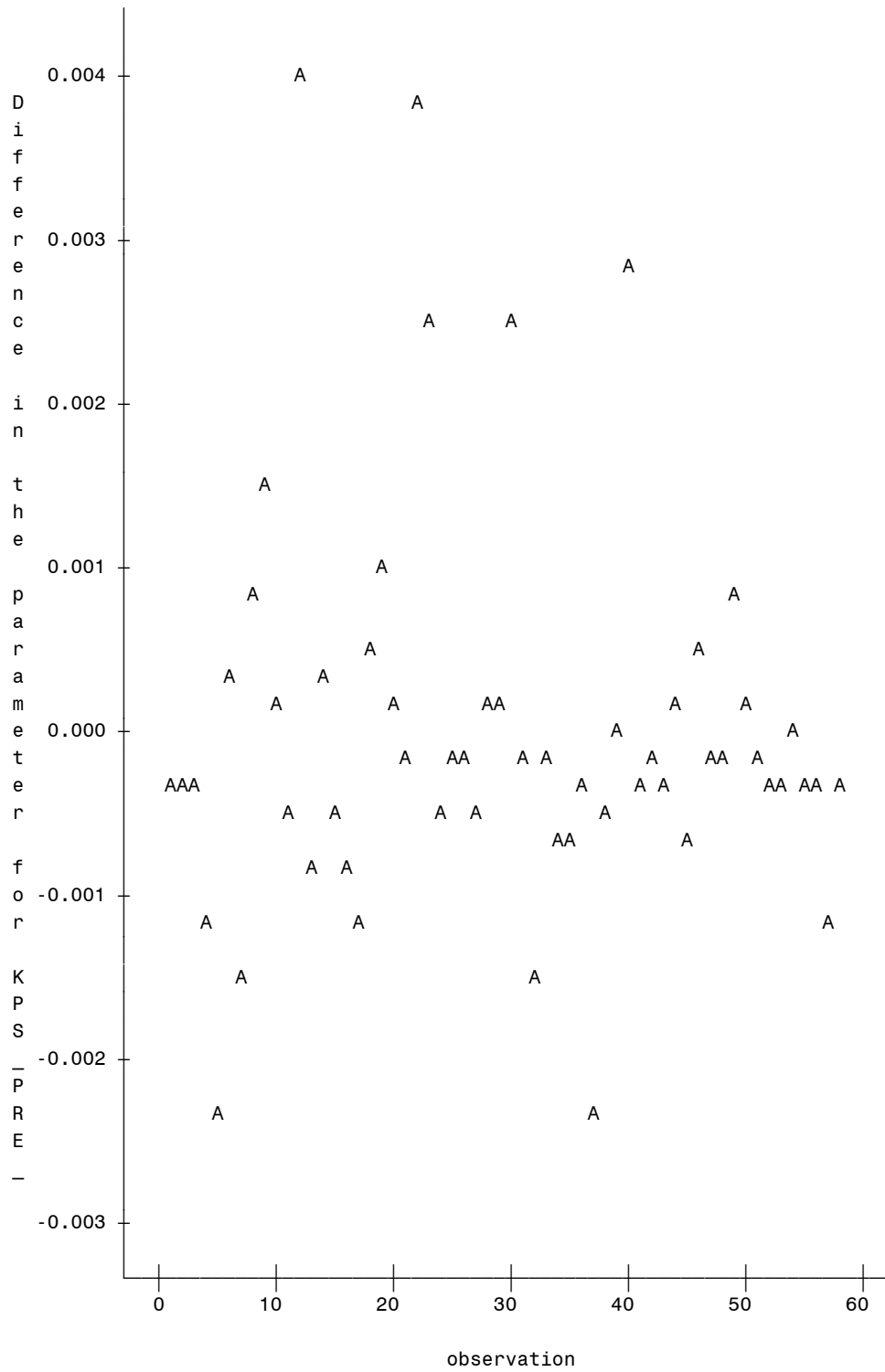
run;

proc plot; /* Figure 6.16 */
  plot KPS_inf*observation;
  plot GROUP_inf*observation;
  plot SEX_inf*observation;
  plot AGE60_inf*observation;
  plot SEX_AGE60_inf*observation;
run;

```

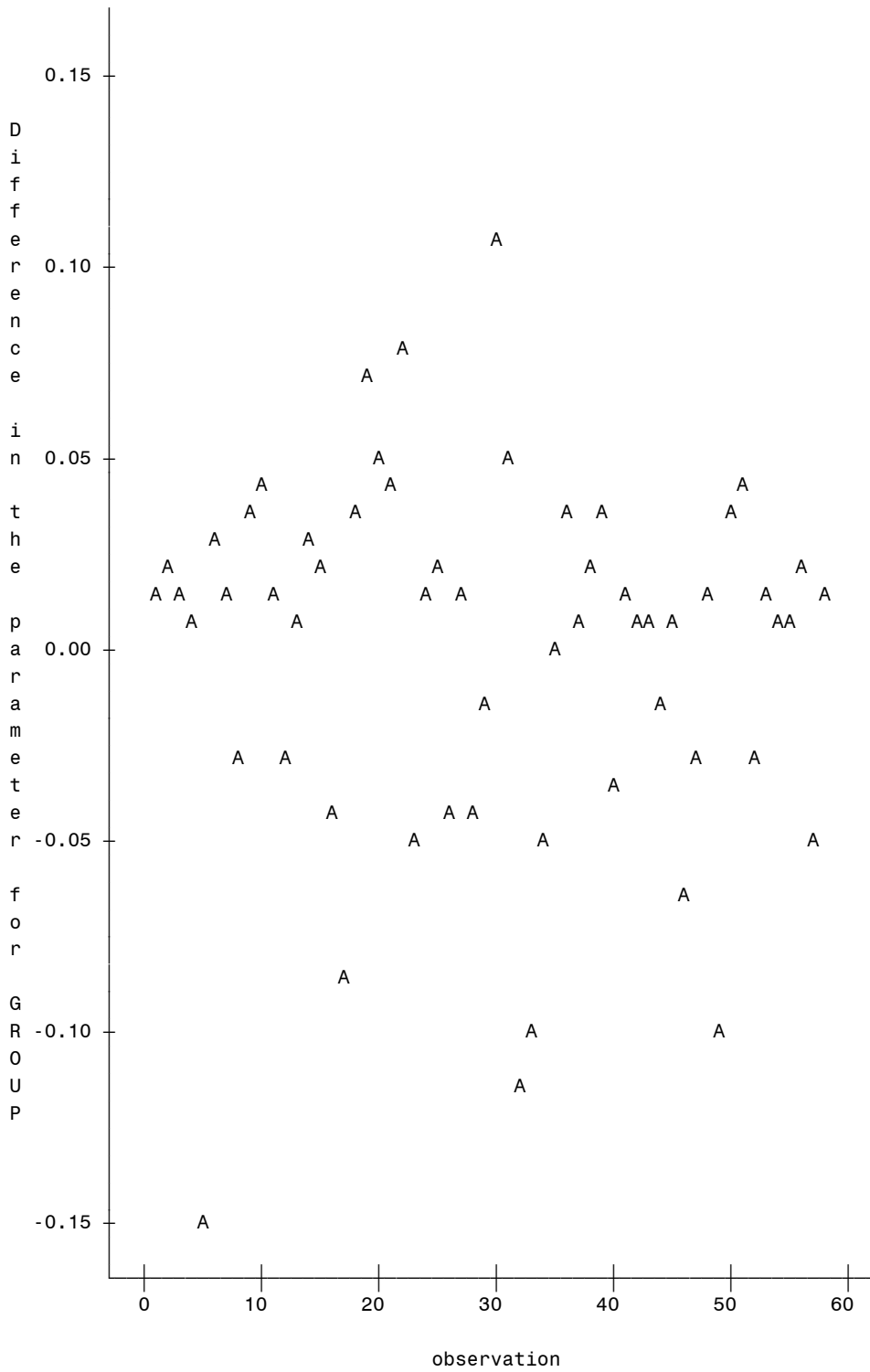
*Figure 6.16;

Plot of KPS_inf*observation. Legend: A = 1 obs, B = 2 obs, etc.



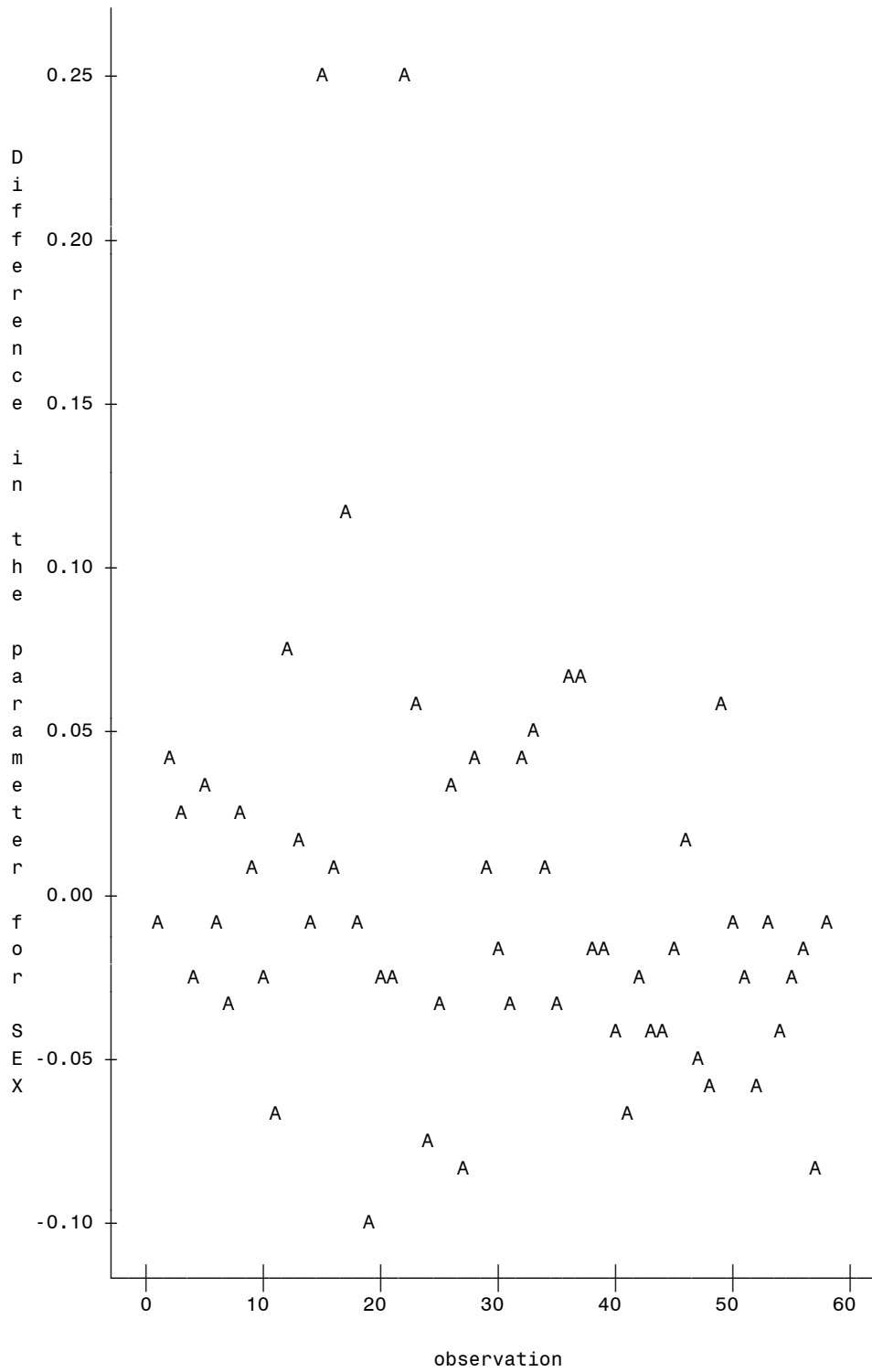
*Figure 6.16;

Plot of GROUP_inf*observation. Legend: A = 1 obs, B = 2 obs, etc.



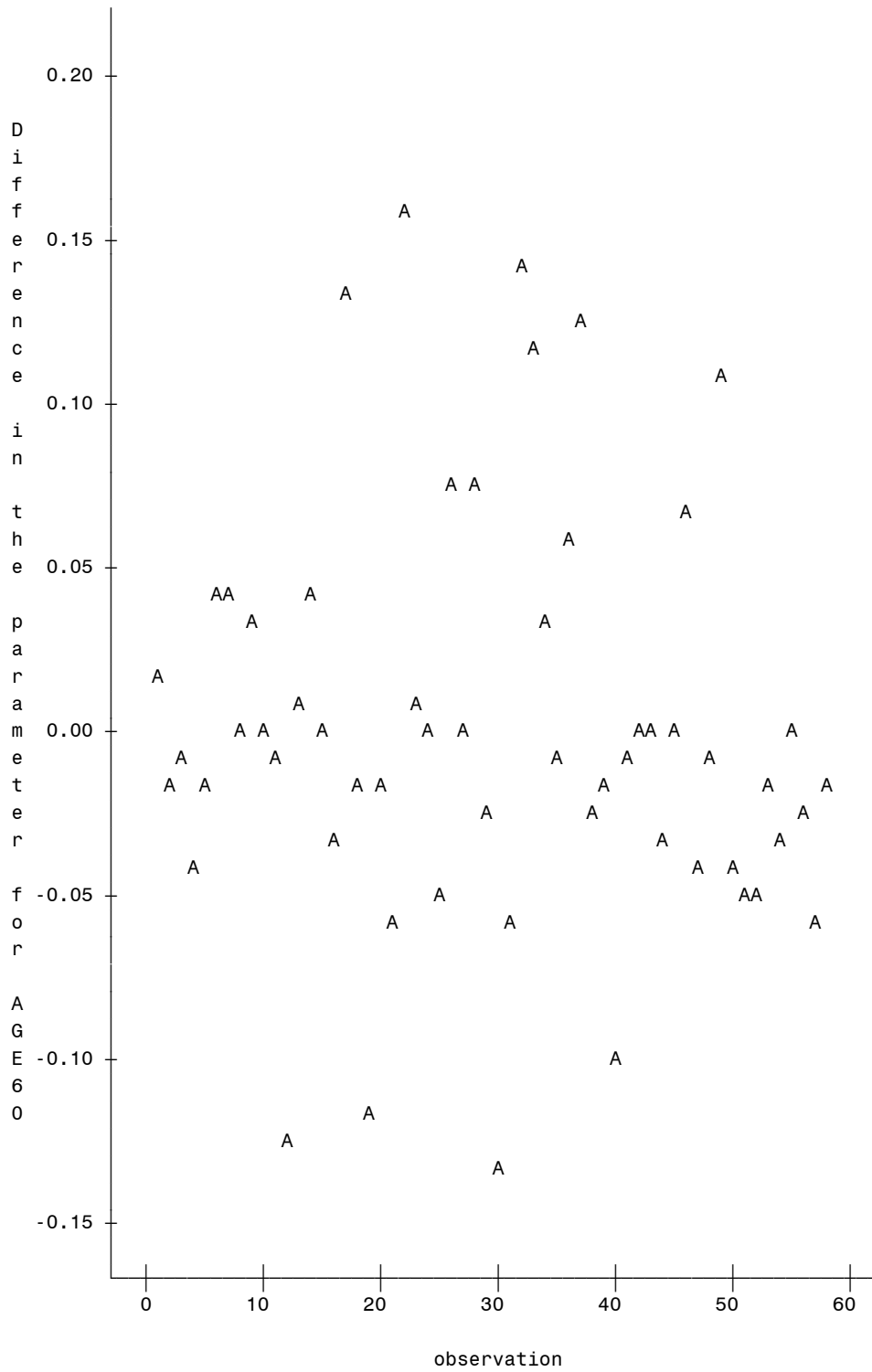
*Figure 6.16;

Plot of SEX_inf*observation. Legend: A = 1 obs, B = 2 obs, etc.



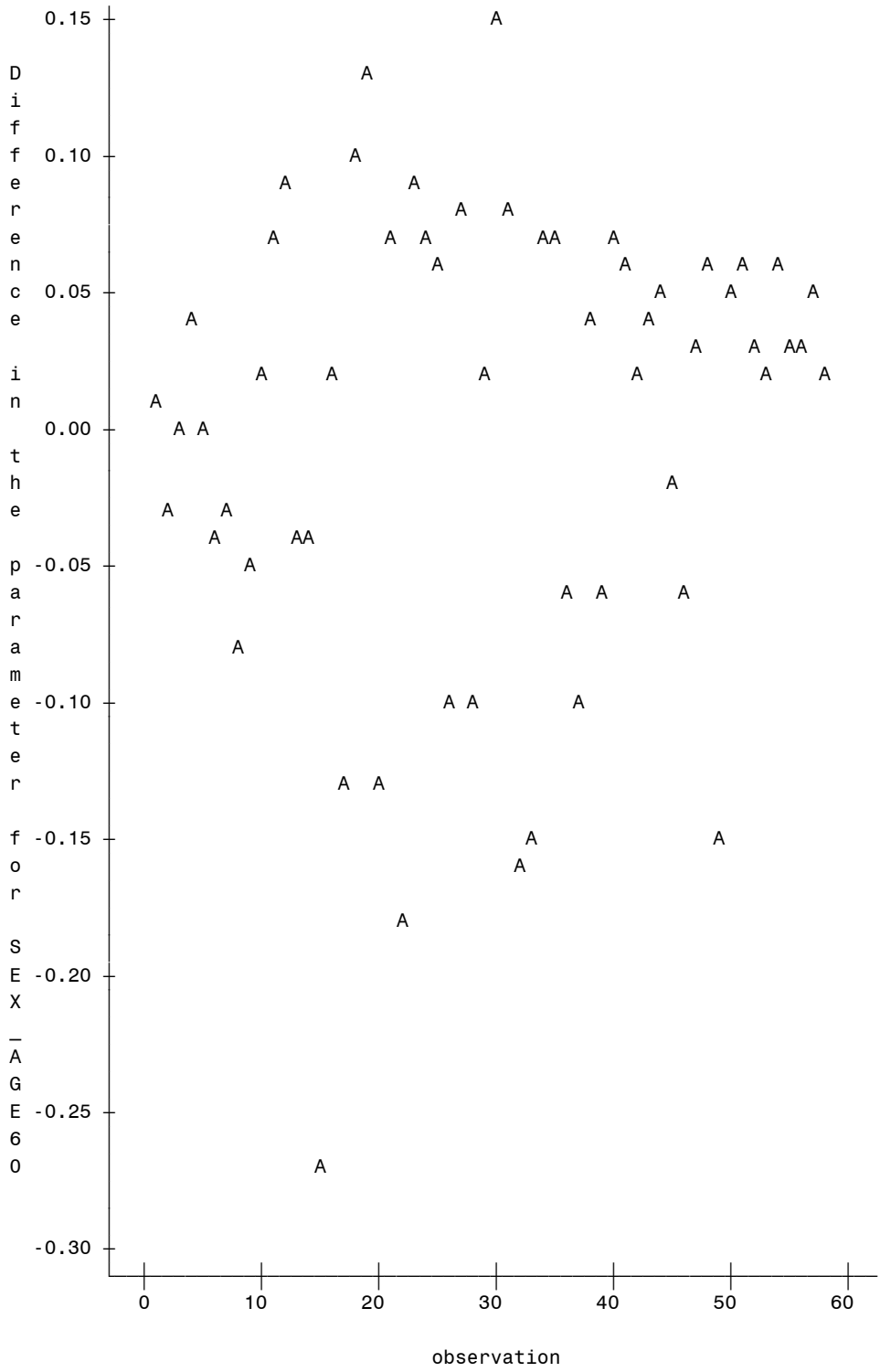
*Figure 6.16;

Plot of AGE60_inf*observation. Legend: A = 1 obs, B = 2 obs, etc.



*Figure 6.16;

Plot of SEX_AGE60_inf*observation. Legend: A = 1 obs, B = 2 obs, etc.



*Figure 6.15 Diagnostic plots of the constancy of the coefficients in cns2.coxint6.;

*Here we briefly describe what a macro is, and reproduce Figure 6.15 by applying macros, schoen and DASPLINE, downloaded from www.mayo.edu/hsr/biostat.html;

*SAS macro code consists of a couple of basic parts: macro and macro variables. The names of macro variables are prefixed and the names of macros are prefixed with a %. A macro variable is similar to a usual data variable except that it doesn't belong to a data set, and its value is always a character. This value could be a variable name, a numeral, or any text that one wants to substitute in his or her program. A macro is a larger piece of program that may contain complex logic and macro statements such as %do, %end, %if, %then, etc.;

```
%macro schoen(data=_last_,time=,event=,xvars=,vref=yes,
              strata=,outsch=schr,outbt=schbt,plot=r,points=yes,
              df=4,pvars=,alpha=.05,rug=no,ties=efron); /* A macro starts
                                                    with %macro */

footnote"schoen macro: event=&event time=&time strata=&strata";
footnote2"Xvars= &xvars";
%let bad=0; /* "let" defines a macro variable. */
%if &time= %then %do;
    %put ERROR: NO TIME VARIABLE GIVEN.;
    %let bad=1;
%end;
%if &event= %then %do;
    %put ERROR: NO EVENT VARIABLE GIVEN.; /* "put" is used to write to the
        SAS Log window the value of the variable of interest */
    %let bad=1;
%end;
%if &xvars= %then %do;
    %put ERROR: NO XVARS GIVEN.;
    %let bad=1;
%end;
%if &df<3 | &df>7 %then %do;
    %put ERROR: DF MUST BE BETWEEN 3 AND 7.;
    %let bad=1;
%end;
%if %upcase(&points)^=YES & %upcase(&points)^=NO %then %do; /* upcase
    converts all the letters in the argument to uppercase */
    %put ERROR: POINTS MUST BE YES OR NO.;
    %let bad=1;
%end;
%if %upcase(&rug)^=YES & %upcase(&rug)^=NO %then %do;
    %put ERROR: RUG MUST BE YES OR NO.;
    %let bad=1;
%end;
%if %upcase(&vref)^=YES & %upcase(&vref)^=NO %then %do;
    %put ERROR: VREF MUST BE YES OR NO.;
    %let bad=1;
%end;
%if %upcase(&plot)^=T & %upcase(&plot)^=R & %upcase(&plot)^=K &
    %upcase(&plot)^=N %then %do;
    %put ERROR: PLOT MUST BE T, R, K, OR N.;
    %let bad=1;
```

```

%end;
%if %upcase(&ties)^=EFRON & %upcase(&ties)^=BRESLOW
& %upcase(&ties)^=DISCRETE & %upcase(&ties)^=EXACT %then %do;
  %put ERROR: TIES MUST BE EFRON, BRESLOW, DISCRETE, EXACT.;
  %let bad=1;
%end;
%let badalpha=0;
data _null_;
  a=symget('alpha');
  if a<0 | a>=1 then call symput('badalpha',1); /* call symput takes a
                                                value and assigns it to a macro variable */
  if a=0 then doalpha=0;
  else doalpha=1;
  call symput('doalpha',doalpha);
run;
%if &badalpha=1 %then %do;
  %put ERROR: ALPHA MUST BE BETWEEN 0 AND 1;
  %let bad=1;
%end;
%let nxs=0; /*count the number of x vars;
%do %until(%scan(&xvars,&nxs+1,' ')= ); /* scan takes a string and an
integer, and return the ith word in the string */
  %let nxs=%eval(&nxs+1); /* eval evaluates arithmetic and logical
expressions */
%end;
%if &pvars^= %then %let plotvars=&pvars;
%else %let plotvars=&xvars;
%let nplotvar=0;
%do %until(%scan(&plotvars,&nplotvar+1,' ')= );
  %let nplotvar=%eval(&nplotvar+1);
%end;
%do i=1 %to &nplotvar;
  %let pv=%scan(&plotvars,&i,' ');
  %let found=0;
  %do j=1 %to &nxs;
    %if &pv=%scan(&xvars,&j,' ') %then %let found=1;
  %end;
  %if &found=0 %then %do;
    %put ERROR: PLOT VARIABLE &PV NOT ON XVARS LIST.;
    %let bad=1;
  %end;
%end;
%end;

%macro xlist(prefix); /*set-up var list code;
  %do j=1 %to &nxs;
    &prefix&j
  %end;
%mend xlist;
%if &bad=0 %then %do;
  data _setup; set &data; /*delete obs with missing values;
  xmiss=0;
  %do i=1 %to &nxs;
    xx=%scan(&xvars,&i);
    if xx=. then xmiss=1;
  %end;
  if &event=. or &time=. or xmiss=1 then delete;

```

```

                                /* run phreg and grab output datasets;
proc phreg data=_setup covout outest=_est ;
  model &time*&event(0)= &xvars / ties=&ties ;
  %if &strata ^= %then %do;
    strata &strata;
  %end;
  output out=_sch xbeta=xbeta ressch=rschl-rsch&nxs
    wtressch=wrschl-wrsch&nxs;

data _sch1; set _sch(keep=&strata &time &event rschl-rsch&nxs);
  drop &event;
  if &event=1;
  rename
  %do i=1 %to &nxs;
    %let thisx=%scan(&xvars,&i,' ');
    rsch&i=&thisx
  %end;
  ;
proc sort; by &time;

data &outsch; set _sch1;
proc sort; by &strata &time;
                                ** calculate overall Kaplan-Meier;
proc lifetest noprint data=_setup outs=_km;
  time &time*&event(0);
data _km2; set _km;
  keep &time probevt;
  if _censor_=0; **keep event times;
  probevt=1-survival;
  label probevt='1-Overall Kaplan-Meier';

data _null_;
  set _est;
  %if %substr(&sysver,1,1)=6 %then %do; /* %substr picks out a
                                subcomponent of a string variable with the
                                start position of the substring being 1 and
                                length of the substring 1 */
    if _type_='PARMS' & _name_='ESTIMATE';
  %end;
  %if %substr(&sysver,1,1)=8 %then %do;
    if _type_='PARMS' & upcase(_name_)=upcase("&time") ;
  %end;
  %do i=1 %to &nxs;
    %let thisx=%scan(&xvars,&i,' ');
    call symput("est&i",&thisx);
  %end;
data _sch2;
  set _sch(keep=&strata &time &event wrsch1-wrsch&nxs);
  keep &strata &time &xvars;
  if &event=1;
  %do i=1 %to &nxs;
    %let thisx=%scan(&xvars,&i,' ');
    &thisx=&&est&i + wrsch&i;
  %end;
proc sort; by &time;
data _bt;
  merge _sch2(in=ins) _km2(keep=&time probevt);

```

```

by &time;
  if ins;
proc sort; by &strata &time;
run;

symbol1 i=join l=1 c=black;
symbol2 i=join l=2 c=black;
symbol3 i=join l=2 c=black;
symbol4 i=none v=dot h=.1 cm c=black;
%macro dovars(tvar);
  %if %length(&tvar)=8 %then %let t=%substr(&tvar,1,7);
  %else %let t=&tvar;
  %daspline(&tvar,nk=&df,data=&outbt)
  %do i=1 %to &nplotvar;
    %let x=%scan(&plotvars,&i,' ');
    data __temp1;
      set &outbt;
        &&_&t
  %global _print_;
  %let _print_=off;
  %let ndummies=%eval(&df-2);
  %if %substr(&sysver,1,1)=8 %then %do;
    ods listing close;
  %end;
proc genmod;
  model &x=&tvar &t.1-&t.&ndummies/dist=normal obstats
  %if &doalpha=1 %then %do;
    alpha=&alpha
  %end;
  ;
  make 'OBSTATS' out=__temp2;
run;
%if %substr(&sysver,1,1)=8 %then %do;
  ods listing;
%end;
data __temp3;
  merge __temp1 __temp2;
/* lower=xbeta-2*std;
   upper=xbeta+2*std; */
%if %upcase(&rug)=YES %then %do;
  data __anno;
    set __temp3;
      x=&tvar;
      y=0;
      xsys='2';
      ysys='1';
      function='move';
      output;
      y=5;
      function='draw';
      output;
  %end;
proc sort data=__temp3; by &tvar;
proc gplot data=__temp3 gout=gschbt
%if %upcase(&rug)=YES %then %do;
  annotate=__anno
%end;

```

```

;
%if &doalpha=1 %then %do;
  plot xbeta*&tvar=1
        lower*&tvar=2
        upper*&tvar=3
        %if %upcase(&points)=YES %then %do;
          &x*&tvar=4
        %end;
  /overlay vaxis=axis1 haxis=axis2
  %if %upcase(&vref)=YES %then %do;
    vref=0 lvref=3
  %end;
;
%end;
%if &doalpha=0 %then %do;
  plot xbeta*&tvar=1
  %if %upcase(&points)=YES %then %do;
    &x*&tvar=4
  %end;
  /overlay vaxis=axis1 haxis=axis2
  %if %upcase(&vref)=YES %then %do;
    vref=0 lvref=3
  %end;
;
%end;
axis1 label=(r=0 a=90 "smooth(&x) df=&df");
%if &tvar=&time %then %do;
  axis2 label=("&tvar");
%end;
%if &tvar=rtime %then %do;
  axis2 label=("Rank for Variable &time");
%end;
%if &tvar=probevt %then %do;
  axis2 label=('1-Overall Kaplan-Meier');
%end;
run; quit;
%end;
%mend dovars;
title2'Scaled residuals(Bt) as a fcn of time.';
proc rank data=_bt out=&outbt; var &time; ranks rtime; /* computes
      ranks for one or more numeric variables */
proc corr pearson data=&outbt; /* computes Pearson correlation
      coefficient */
  with &xvars; var &time rtime probevt;
  label probevt='1 - Overall Kaplan-Meier';

%if %upcase(&plot)=T %then %do;
  %dovars(&time)
%end;

%if %upcase(&plot)=R %then %do;
  %dovars(rtime)
%end;

%if %upcase(&plot)=K %then %do;
  %dovars(probevt)
%end;

```

```

run;
quit;
footnote; symbol; title2;
proc datasets nolist;
  delete _setup _est _sch _sch1 _sch2 _km _km2 _bt
  %if %upcase(&rug)=YES %then %do;
    __anno
  %end;
  __temp1 __temp2 __temp3;
run;
quit;
%end;
%mend schoen; /* macro ends with %mend */

%MACRO DASPLINE(x,nk=4,knot1=,knot2=,knot3=,knot4=,knot5=,knot6=,knot7=,
knot8=,knot9=,knot10=,knot11=,knot12=,knot13=,knot14=,knot15=,
knot16=,knot17=,knot18=,knot19=,knot20=,
norm=2,data=_LAST_);
%LOCAL i j needknot nx lastds v v7 k tk tk1 t t1 k2 low hi slow shi kd;
/* "Local" creates macro variables that are available only
when the macro is executed. */
%LOCAL _e_1 _e_2 _e_3 _e_4 _e_5 _e_6 _e_7 _e_8 _e_9;
%LOCAL _lastds_;
*%DSHIDE;
%LET x=%SCAN(&x,1,''); /* "scan" searches for a word that is
specified by a number */
%LET needknot=0; %LET nx=0;
*%Strip off quotes in KNOTs and see if any KNOTS unspecified;
%DO i=1 %TO 20;
%IF %SCAN(&x,&i)= %THEN %GOTO nomorex;
%LET nx=%EVAL(&nx+1); /* "eval" evaluates arithmetic and logical
expressions using integer arithmetic */
%IF %QUOTE(&&knot&i)= %THEN %LET needknot=%EVAL(&needknot+1);
%ELSE %LET knot&i=%SCAN(&&knot&i,1,''); /* "quote" is used to mark
special characters and mnemonic operators in a
resolved value at macro execution */
%END;
%nomorex:
%IF &needknot^=0 %THEN %DO;
%LET lastds=&sysdsn; %IF &lastds^=_NULL_ %THEN
%LET lastds=%SCAN(&sysdsn,1).%SCAN(&sysdsn,2);
RUN;OPTIONS NONOTES;
PROC UNIVARIATE DATA=&data NOPRINT;VAR
%DO i=1 %TO &nx;
%IF %QUOTE(&&knot&i)= %THEN %SCAN(&x,&i);
%END;
; OUTPUT OUT=_stats_ pctlpts=
%IF &nk=3 %THEN 5 50 95;
%IF &nk=4 %THEN 5 35 65 95;
%IF &nk=5 %THEN 5 27.5 50 72.5 95;
%IF &nk=6 %THEN 5 23 41 59 77 95;
%IF &nk=7 %THEN 2.5 18.3333 34.1667 50 65.8333 81.6667 97.5;
PCTLPRE=x1-x&needknot PCTLNAME=p1-p&nk;
DATA _NULL_;SET _stats_;
*%For knot points close to zero, set to zero;
ARRAY _kp_ _NUMERIC_;DO OVER _kp_;IF ABS(_kp_)<1E-9 THEN _kp_=0;END;

```

```

%LET j=0;
%DO i=1 %TO &nx;
%IF %QUOTE(&&knot&i)= %THEN %DO;
  %LET j=%EVAL(&j+1);
  CALL SYMPUT("knot&i",
  TRIM(LEFT(x&j.p1)) /* "left" left-aligns an argument by removing
  leading blanks */
  %DO k=2 %TO &nk;
  ||" "||TRIM(LEFT(x&j.p&k)) /* "trim" trims trailing blanks */
  %END;
);
%END;
%END;
RUN; OPTIONS NOTES _LAST_=&lastds;
%END;
%DO i=1 %TO &nx;
%GLOBAL _knot&i._; %LET _knot&i._=&&knot&i;
%PUT Knots for %SCAN(&x,&i):&&knot&i;
%END;
%*Generate code for calculating dummy variables;
%DO i=1 %TO &nx;
%LET v=%SCAN(&x,&i); %IF %LENGTH(&v)=8 %THEN %LET v7=%SUBSTR(&v,1,7);
%ELSE %LET v7=&v; /* "length" returns the length of a string */
%GLOBAL _&v7; %LET _&v7=;
%*Get no. knots, last knot, next to last knot;
%DO k=1 %TO 99;
%IF %QUOTE(%SCAN(&&knot&i,&k,%STR( )))= %THEN %GOTO nomorek;
%END;
%nomorek: %LET k=%EVAL(&k-1); %LET k2=%EVAL(&k-2);
%LET tk=%SCAN(&&knot&i,&k,%STR( ));
%LET tk1=%SCAN(&&knot&i,%EVAL(&k-1),%STR( ));
%LET t1=%SCAN(&&knot&i,1,%STR( ));
%IF &norm=0 %THEN %LET kd=1;
%ELSE %IF &norm=1 %THEN %LET kd=(&tk - &tk1);
%ELSE %LET kd=((&tk - &t1)**.666666666666666);
%LET _e_1=; %LET _e_2=; %LET _e_3=; %LET _e_4=; %LET _e_5=;
%LET _e_6=; %LET _e_7=; %LET _e_8=; %LET _e_9=;
%DO j=1 %TO &k2;
%LET t=%SCAN(&&knot&i,&j,%STR( ));
%LET _e_&j=&v7&j=max((&v-&t)/&kd,0)**3+((&tk1-&t)*max((&v-&tk)/&kd,0)**3
-(&tk-&t)*max((&v-&tk1)/&kd,0)**3)/(&tk-&tk1)%STR(;);
%END;
%LET _&v7=&_e_1 &_e_2 &_e_3 &_e_4 &_e_5 &_e_6 &_e_7 &_e_8 &_e_9;
%END;
%OPTIONS _LAST_ = &_amp;lastds_;
%MEND;

```

```

%schoen(data=CNS2new, time=B3TODEATH, event=STATUS, xvars= KPS_PRE_ GROUP SEX
AGE60 SEX_AGE60, plot=r); /* Figure 6.15 */

```

schoen macro: event=STATUS time=B3TODEATH strata=
 Xvars= KPS_PRE_ GROUP SEX AGE60 SEX_AGE60

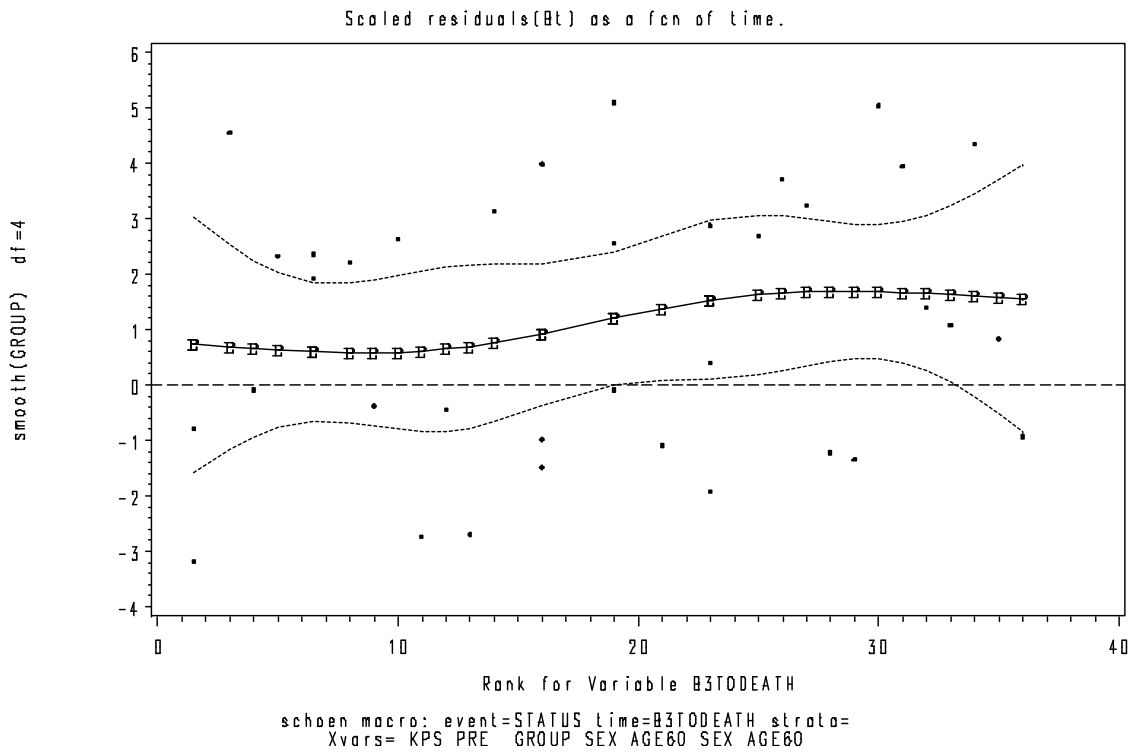
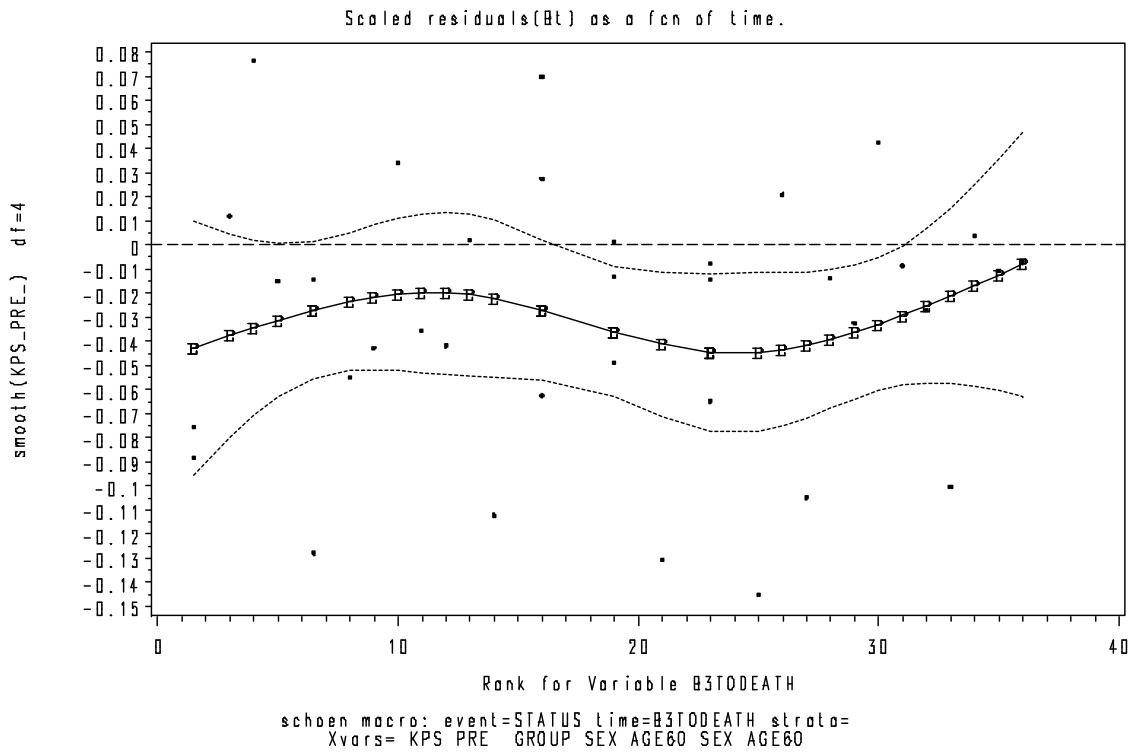
Scaled residuals(Bt) as a fcn of time.

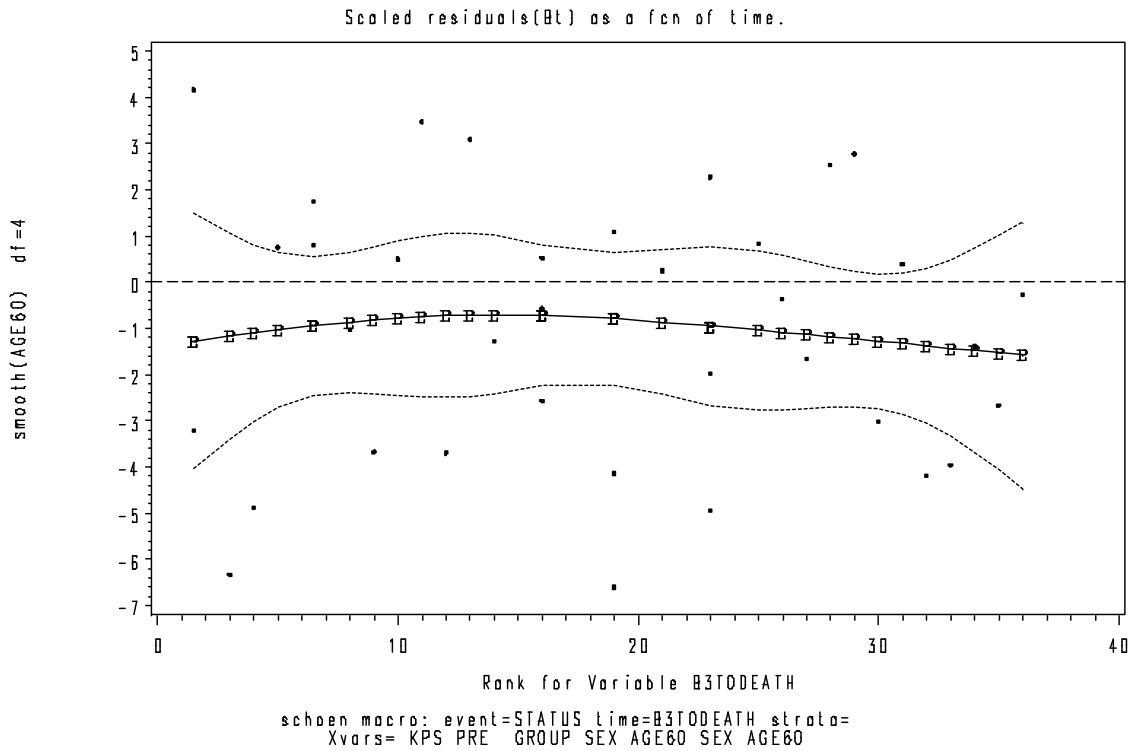
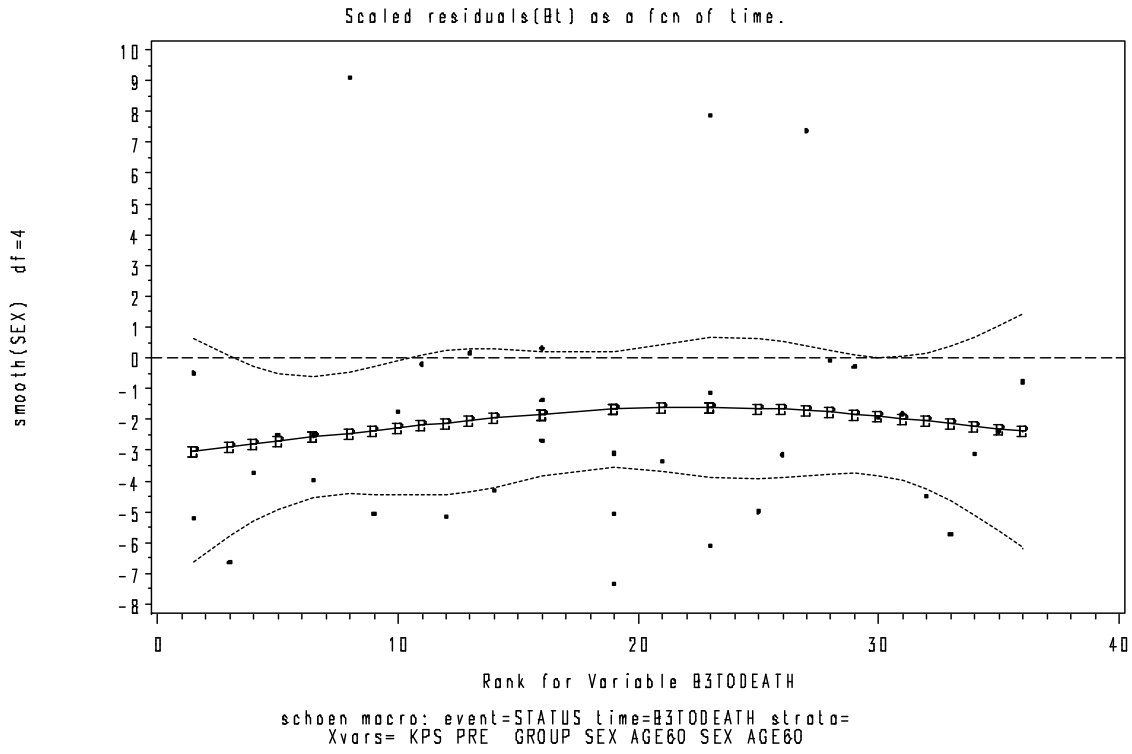
Simple Statistics

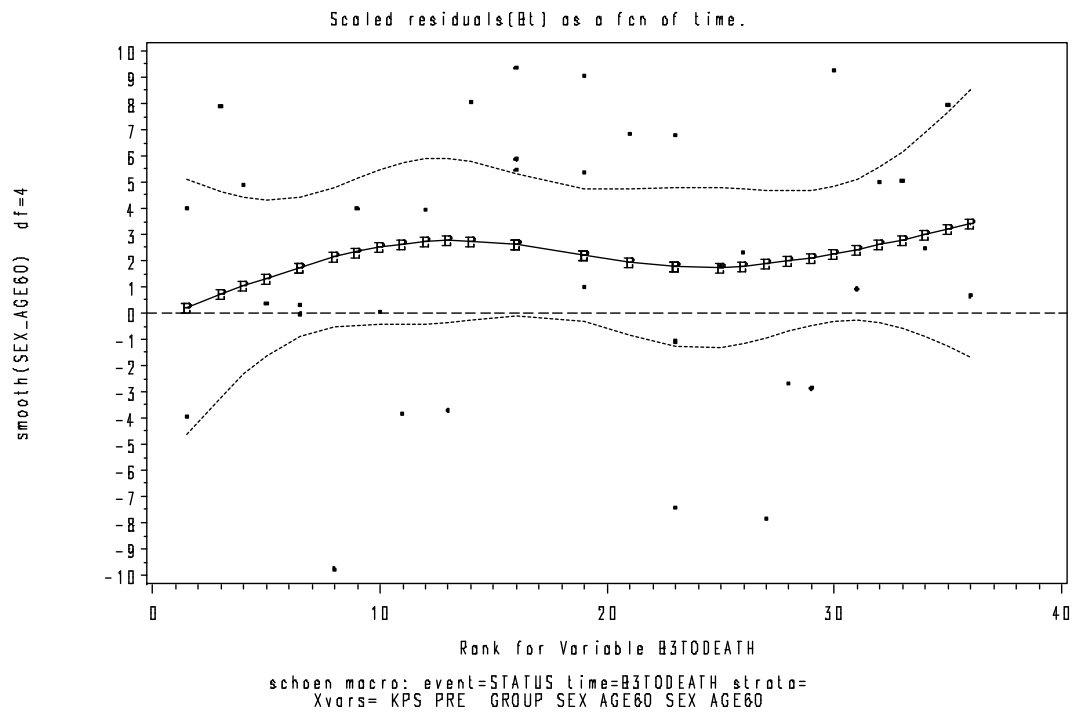
Variable Label
 KPS_PRE_
 GROUP
 SEX
 AGE60
 SEX_AGE60
 B3TODEATH B3TODEATH
 rtime Rank for Variable B3TODEATH
 probevt 1 - Overall Kaplan-Meier

Pearson Correlation Coefficients, N = 36
 Prob > |r| under H0: Rho=0

	B3TODEATH	rtime	probevt
KPS_PRE_	0.07923	0.02286	0.03977
	0.6460	0.8947	0.8178
GROUP	0.02529	0.17533	0.15704
	0.8836	0.3064	0.3603
SEX	0.02652	0.06953	0.05611
	0.8780	0.6870	0.7452
AGE60	-0.04098	-0.04854	-0.06107
	0.8124	0.7786	0.7235
SEX_AGE60	0.07042	0.08673	0.10076
	0.6832	0.6150	0.5588







SAS Textbook Examples for Chapter 7: Additional Topics

```

libname mylib 'H:\Survival_Kim\C4088\Datasets';
PROC IMPORT OUT= mylib.ADDICTS
            DATAFILE= "H:\Survival_Kim\C4088\Datasets\ADDICTS.xls"
            DBMS=EXCEL2000 REPLACE;
            RANGE="Addicts1$";
            GETNAMES=YES;
RUN;
options ls=80 nocenter nodate;
data ADDICTSnew;
set mylib.ADDICTS;
run;

*Part I: pp. 187 - 189 including Figure 7.2;

symbol1 c=black l = 1 w = 3;
symbol2 c=black l = 3 w = 3;
proc lifetest plots = (s) data=ADDICTSnew;
time Days_survival*STATUS(0);
strata Clinic;
run;

```

Stratum 1: Clinic = 1

Quartile Estimates

Percent	Point Estimate	95% Confidence Interval [Lower Upper)	
75	652.00	560.00	755.00
50	428.00	341.00	512.00
25	192.00	160.00	244.00

Mean	Standard Error
431.47	22.51

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.

Stratum 2: Clinic = 2

Quartile Estimates

Percent	Point Estimate	95% Confidence Interval [Lower Upper)	
75	.	.	.
50	.	661.00	.
25	280.00	190.00	540.00

Mean	Standard Error
629.82	39.34

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.

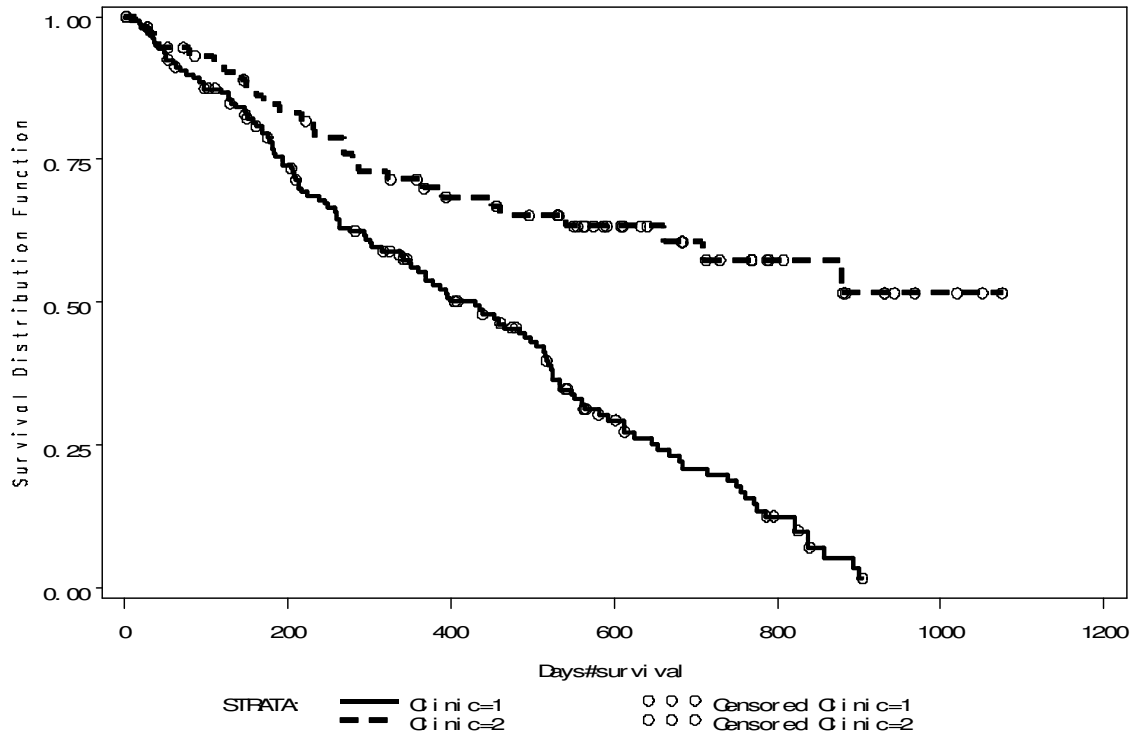
Summary of the Number of Censored and Uncensored Values

Stratum	Clinic	Total	Failed	Censored	Percent Censored
1	1	163	122	41	25.15
2	2	75	28	47	62.67
Total		238	150	88	36.97

Test of Equality over Strata

Test	Chi-Square	DF	Pr > Chi-Square
Log-Rank	27.8927	1	<.0001

Figure 7.2 K-M curves for ADDICTS not adjusted for covariates



*Part II: pp. 188 - 191;

```
proc phreg data=ADDICTSnew;
model Days_survival*Status(0) = Clinic Prison Dose;
run;
```

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	64.5190	3	<.0001
Score	56.2730	3	<.0001
Wald	54.0906	3	<.0001

Analysis of Maximum Likelihood Estimates

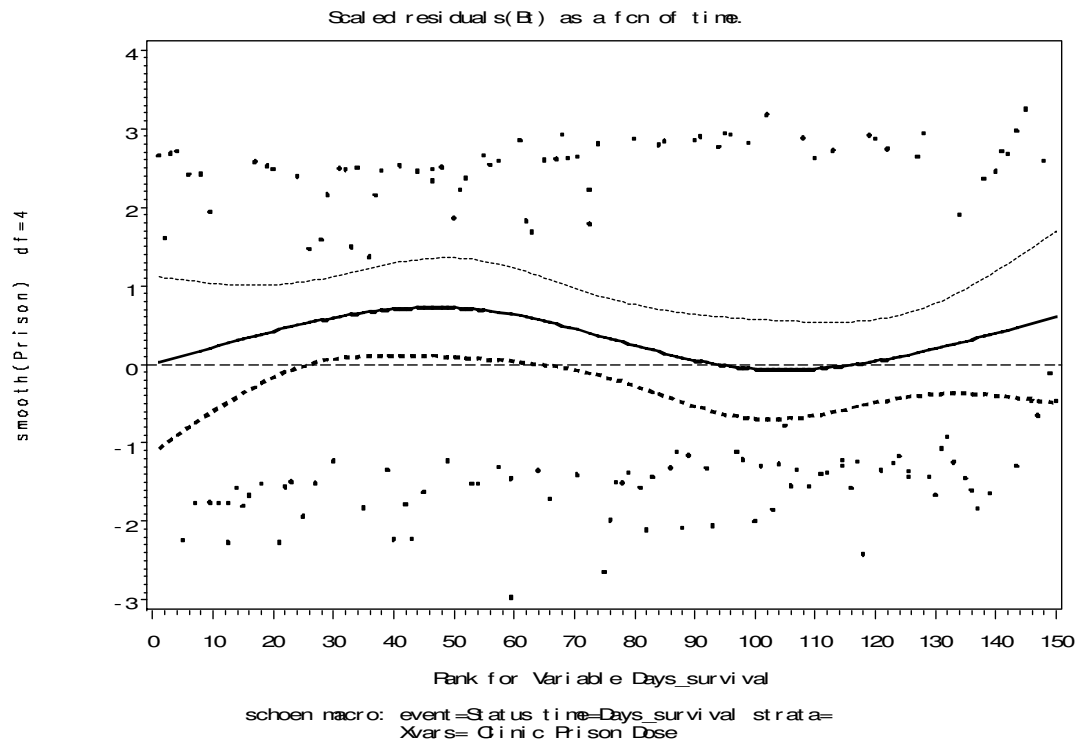
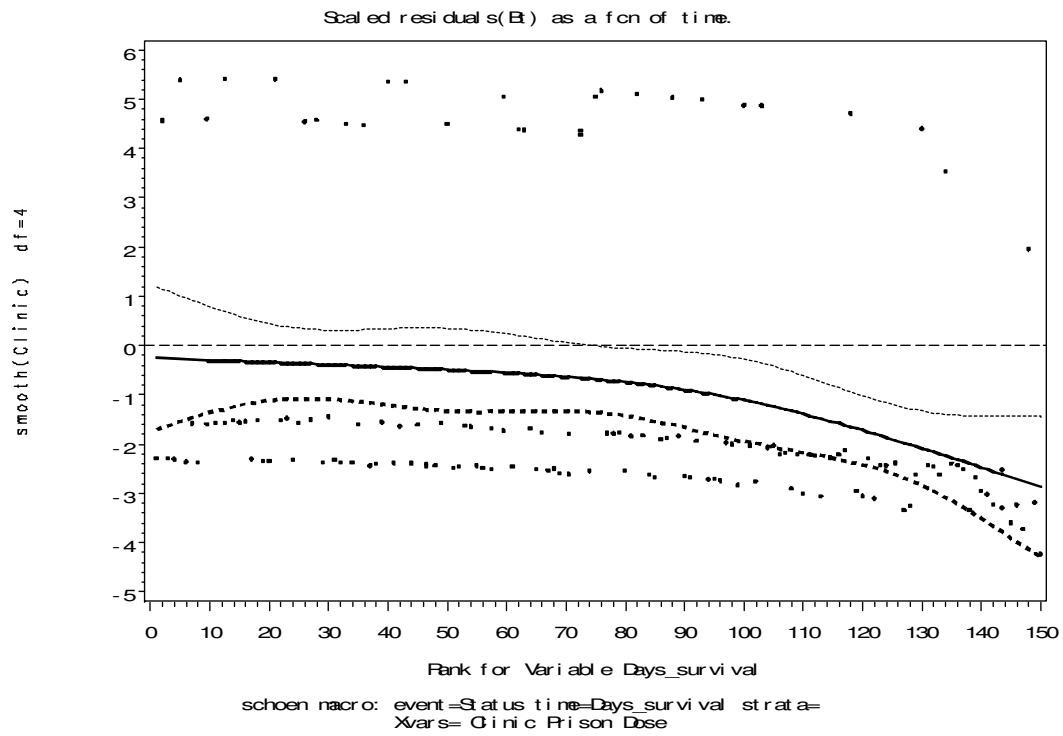
Variable	DF	Parameter		Chi-Square	Pr > ChiSq
		Estimate	Standard Error		
Clinic	1	-1.00876	0.21486	22.0419	<.0001
Prison	1	0.32650	0.16722	3.8123	0.0509
Dose	1	-0.03540	0.00638	30.7844	<.0001

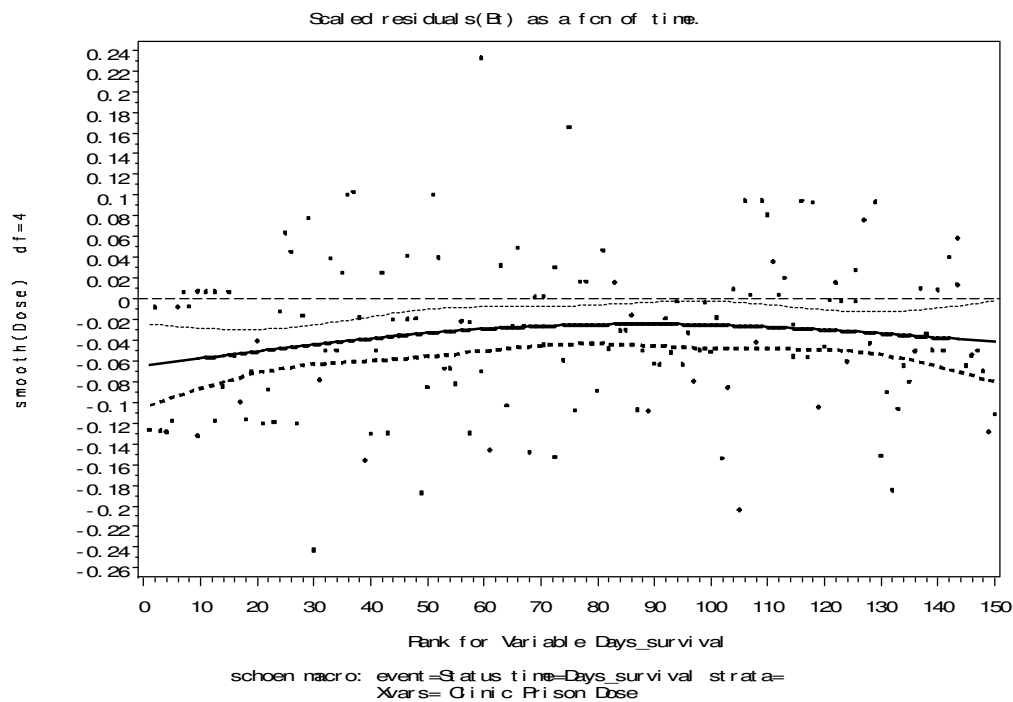
```
%schoen(data=ADDICTSnew, time=Days_survival, event=Status, xvars= Clinic Prison
Dose, plot=r); /* Figure 7.3 */
```

schoen macro: event=Status time=Days_survival strata= Xvars= Clinic Prison Dose

Pearson Correlation Coefficients, N = 150
Prob > |r| under H0: Rho=0

	Days_ survival	rtime	probevt
Clinic	-0.26342 0.0011	-0.24926 0.0021	-0.25735 0.0015
Prison	-0.03646 0.6578	-0.04637 0.5731	-0.03839 0.6409
Dose	0.06201 0.4510	0.08990 0.2739	0.07180 0.3826





*Part III: Stratified Cox model, pp. 191 - 192;

```
*output;
proc phreg data=ADDICTSnew;
strata Clinic;
model Days_survival*Status(0) = Prison Dose;
run;
```

Testing Global Null Hypothesis: BETA=0

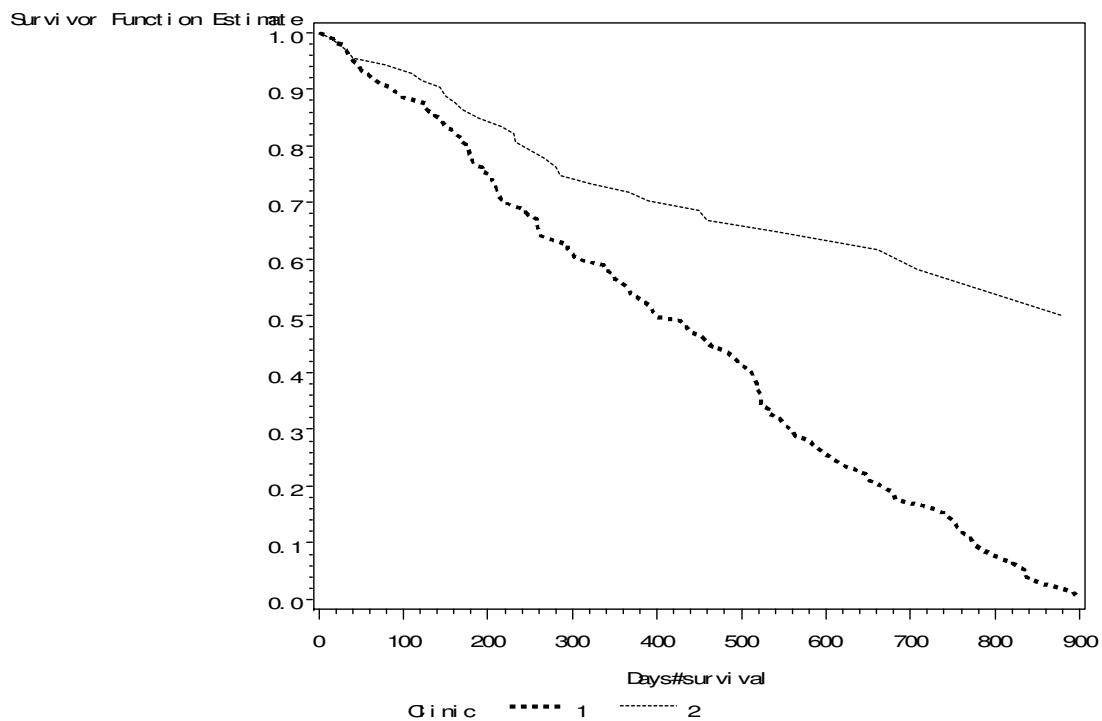
Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	33.9393	2	<.0001
Score	33.3628	2	<.0001
Wald	32.6858	2	<.0001

Analysis of Maximum Likelihood Estimates

Variable	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq
Prison	1	0.38877	0.16892	5.2974	0.0214
Dose	1	-0.03514	0.00647	29.5471	<.0001

***Figure 7.4;**

```
proc sort data=ADDICTSnew; by Clinic; run;
proc phreg data=ADDICTSnew;
model Days_survival*Status(0) = Prison Dose;
baseline out=out survival=survival loglogs=lcumhaz;
by Clinic;
run;
proc gplot data=out;
plot survival*Days_survival=Clinic; /* baseline survival curves for */
run; /* Clinic 1 and Clinic 2 */
```



***Part IV: An extended Cox model analysis, pp. 192 - 194;**

```
data extended;
set ADDICTSnew; if Clinic=2 then Clinic=0; run;
proc phreg data=extended;
model Days_survival*Status(0) = Prison Dose Clinicg1t Clinicg2t;
Clinicg1t=0;
if Days_survival < 365 then Clinicg1t=Clinic;
Clinicg2t=0;
if Days_survival >= 365 then Clinicg2t=Clinic;
run;
```

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	74.1749	4	<.0001
Score	64.1371	4	<.0001
Wald	57.4441	4	<.0001

Analysis of Maximum Likelihood Estimates

Variable	DF	Parameter		Chi-Square	Pr > ChiSq
		Estimate	Standard Error		
Prison	1	0.37770	0.16840	5.0304	0.0249
Dose	1	-0.03551	0.00644	30.4503	<.0001
Clinicg1t	1	0.45956	0.25529	3.2405	0.0718
Clinicg2t	1	1.82823	0.38595	22.4392	<.0001

*Left-truncated and right-censored data: pp. 210 - 211;

```

PROC IMPORT OUT= mylib.psychi
  DATAFILE= "H:\Survival_Kim\C4088\Datasets\psychi.xls"
  DBMS=EXCEL2000 REPLACE;
  RANGE="psychi$";
  GETNAMES=YES;
RUN;

/* **** phreg for left-truncated and right-censored data ****
** *****
** This requires two time variables in the model statement.
** "entry" is the delayed entry time into the study,
** that is, left-truncated time.
** "fuptime" is the failure time, either exact or right-censored.
** The rest of the model statement is the same as that for
** usual right-censored data.
** ***** */

data psychinew; /* converting a character variable into a numeric */
                /* one as an explanatory variable must be numeric. */
set mylib.psychi;
if SEX = 'M' then gender = 1;
if SEX = 'F' then gender = 0;
run;

proc phreg data=psychinew;
model (entry, futime)*status(0) = gender;
run;

```

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	0.6912	1	0.4058
Score	0.6649	1	0.4148
Wald	0.6525	1	0.4192

Analysis of Maximum Likelihood Estimates

Variable	DF	Parameter	Standard	Chi-Square	Pr > ChiSq
		Estimate	Error		
gender	1	-0.48526	0.60075	0.6525	0.4192

***Mistreating left-truncated and right-censored data as right-censored data;**

```
proc phreg data=psychinew;
model futime*status(0) = gender;
run;
```

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	0.4922	1	0.4829
Score	0.4745	1	0.4909
Wald	0.4681	1	0.4939

Analysis of Maximum Likelihood Estimates

Variable	DF	Parameter	Standard	Chi-Square	Pr > ChiSq
		Estimate	Error		
gender	1	-0.40982	0.59901	0.4681	0.4939

References

- Burlew, Michele M. (1998). *SAS Macro Programming Made Easy*. SAS Publishing.
- Cantor, Alan B. (2003). *SAS Survival Analysis Techniques for Medical Research 2nd edition*. SAS Publishing.
- Cody, Ronald and Pass, Ray. (1995). *SAS Programming by Example*. SAS Publishing.
- Harrell, Jr., Frank E. (2004). SAS macro *daspline*. Mayo Clinic College of Medicine.
- Hosmer, D. W. and Lemeshow, S. (1999). *Applied Survival Analysis: Regression Modeling of Time to Event Data*. New York: John Wiley and Sons.
- SAS OnlineDoc, V8. (1999). SAS Institute Inc., Cary, NC, USA.
- SAS (r) Proprietary Software Release 8.2 (TS2M0). SAS Institute Inc., Cary, NC, USA.
- Tableman, Mara and Kim, Jong S. (2004). *Survival Analysis Using S: Analysis of Time-to-Event Data*. Chapman & Hall/CRC, Boca Raton.
- Therneau, Terry and Bergstralh, Erik. (2004). SAS macro *schoen*. Mayo Clinic College of Medicine.
- Therneau, T. M. and Grambsch, P. M. (2000). *Modeling Survival Data: Extending the Cox Model*. New York: Springer-Verlag.