Paper AD17-2011

Application for Survival Analysis through Microsoft© Access GUI

Zhong Yan, i3, Indianapolis, IN
Jie Li, i3, Austin, Texas
Jiazheng (Steven) He, i3, San Francisco, California

Abstract

This paper describes the application we developed that allows users who have minimum SAS knowledge to perform survival analysis. This application was built on Microsoft Access using Visual Basic for Applications (VBA). It contains a graphical user interface (GUI) that divides the analysis tasks into different tabs. Under each tab, user can enter parameters such as dataset location, variable names for time and censor, censoring indicator, and comparison group variable name. After user hits certain button, it will trigger SAS to run Lifetest or Cox regression survival analysis. This paper will introduce the design of the GUI and VBA code to run survival analysis.

Introduction

There are cases that investigators would like to compare treatment effects on various survival endpoints within different subgroups of patients. It could be time consuming to send requests out and wait for the results being sent back. It is also inconvenient to think ahead of time for all of the needed analysis requests being sent out at one time.

The application Survival Analysis was developed for users with minimum SAS knowledge to perform survival analysis. Using this application, investigators can easily control which subpopulation within the time to event dataset he would like to analyze with minimum SAS knowledge.

The application contains a GUI for users to input needed information to customize the dynamic analysis. It was built on Microsoft Access using Visual Basic for Applications (VBA). This paper will focus on one of the analysis functions – Lifetest. The design of the GUI, the usage of the application through examples, and the VBA code to drive survival analysis with SAS will be introduced.

System Requirement

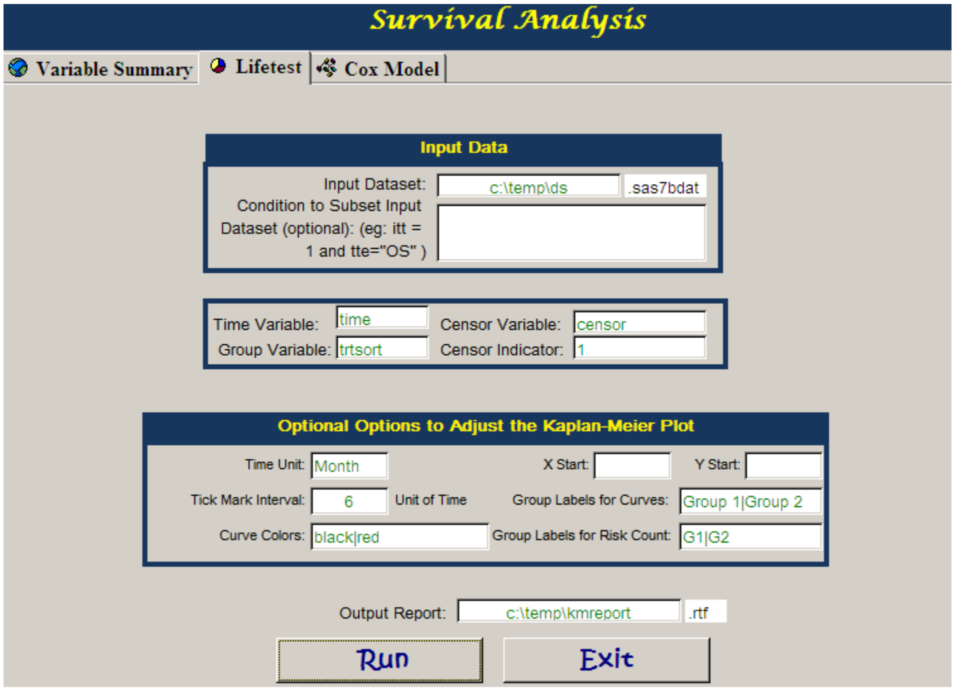
The Survival Analysis application was built on Microsoft Access 2007 and tested on SAS 9.0 and above (Windows based). The user needs to have Microsoft Access 2007 and SAS for Windows (version 9.0 or above) installed on the same computer. The application specific file “kmsurv.sas” needs to be stored in the same directory as the application.

GUI Design

The Survival Analysis contains an Access macro called “autoexec”. It acts as the springboard for the application. When the application is double clicked, this macro will be executed automatically. The “autoexec” contains an action called “OpenForm” to open the specified GUI. The GUI for Survival Analysis is essentially an Access form. The form contains 3 tabs: Variable Summary, Lifetest, and Cox Model. Clicking on each tab brings up specific interface to accept user’s input associated with the specific analysis task. Figure 1 shows the GUI for the tab “Lifetest”.

The GUI of the tab “Lifetest” contains Access components of (1) labels to describe the fields, (2) text boxes to accept user’s input, (3) rectangles to organize the GUI components for cosmetic effect, and (4) buttons to trigger either the analysis or exit application action. Some of the text boxes are associated with certain events, for example, “Got Focus” and “Lost Focus” events. In this application, “Got Focus” event unhides specified label, which displays instruction on how to use the text box. “Lost Focus” event can work in the opposite way – hiding the instruction label if the label exists. If the text box is supposed to accept input for a file path and name, “Lost Focus” event will check the input value for the text box. For example, if no input, the error message will pop up to remind the user of missing information. If the input path does not exist, the error message will tell you that the path does not exist. The default values for text boxes give user examples of the input format. The default values show up automatically when the analysis tab is clicked at the first time. The button “Run” triggers the analysis task using SAS. The button “Exit” triggers exit application event.

The tabs “Variable Summary” and “Cox Model” have the similar GUI design as the tab “Lifetest”. The only difference is that “Cox Model” also contains a Combo Box to allow user to select the tie handling method from the listed choices.



Input Data

Input Dataset: c:\temp\ds .sas7bdat

Condition to Subset Input Dataset (optional): (eg: itt = 1 and tte="OS")

Time Variable: time

Censor Variable: censor

Group Variable: trtsort

Censor Indicator: 1

Optional Options to Adjust the Kaplan-Meier Plot

Time Unit: Month

X Start:

Y Start:

Tick Mark Interval: 6 Unit of Time

Group Labels for Curves: Group 1|Group 2

Curve Colors: black|red

Group Labels for Risk Count: G1|G2

Output Report: c:\temp\kmreport .rtf

Figure 1. GUI of the Application Survival Analysis

Application Usage Examples

The tab “Lifetest” performs Kaplan-Meier survival analysis. The interface has fields to input SAS dataset path and name, the time variable and the censor variable names, as well as censor indicator value representing censored subjects. If a subpopulation needs to be analyzed, the user dataset subset condition can be entered, for example, the survival endpoint type and disease histology, providing the variables exist in the SAS dataset.

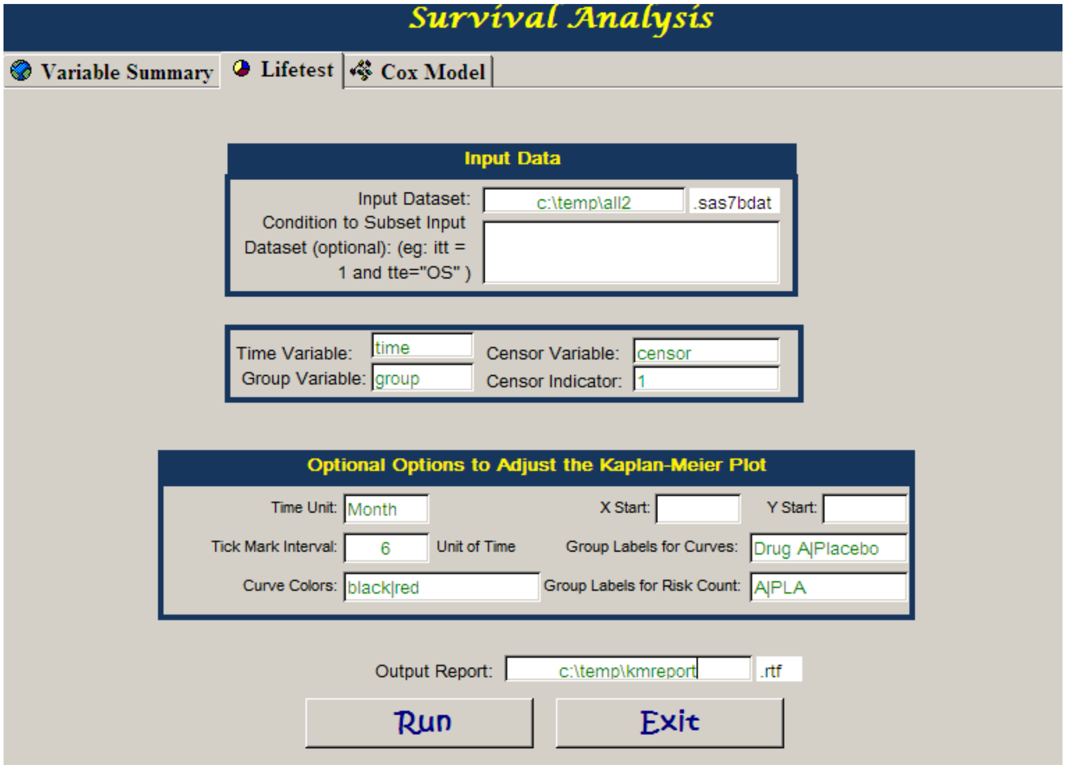
There is a format requirement for the comparison group variable (for example, variable name “treatment”) in the SAS dataset. This variable needs to have values of sequential numbers starting from 1. The values act as group ID. If the text box for the group variable is blank, the whole data will be treated as one group. The analysis result is Kaplan-Meier curves stored in a RTF file, one curve for each group. Some key summary statistics will also be summarized and shown on the graph.

For two group comparison, in addition to the output curves, the result also contains a summary table. Hazard ratio will be calculated and included in the summary table. It is important to make sure the interest group has group variable value 1 and the reference group with value 2 to ensure the correct interpretation of the hazard ratio on the resulting report. (For example, treatment=1 for group Drug A, treatment=2 for group Placebo).

To ease the usage of the application, instructions are often given through “Got Focus” event associated with the text box - when user clicks specific text box, instruction will show up. The major advantages of “Lifetest” analysis tool are: (1) it is convenient to customize the graph (for example, change the color of each curve, specify summary text location and the labels of the groups, test different time interval to calculate the number of patients at risk, enter dataset subset condition); (2) in addition to the graph, it generates complicated summary table for two group comparison.

Example1 – Create Kaplan-Meier Plot and Summary Table for Two Group Comparison

Figure 2 shows the screenshot of the user input to perform two group comparisons of the survival summary statistics. The input simulated SAS dataset name is all2.sas7bdat in this example. It contains two treatment groups. The data is stored in the path “c:\temp\”. The group variable has the name as “group” in this dataset with value 1 for drug A and 2 for placebo. After all needed information has been entered and the button “Run” has been clicked, three output files will be generated. The first output file is called “kmreport.rtf”, which stores the output summary table (see figure 3) and the Kaplan-Meier graph (see figure 4). The second output file is called “kmreport.log”, which contains the SAS log. The third output file is “kmreport.sas”, which is dynamically generated from MS Access VBA. This SAS file contains the information to customize the analysis based on the user’s input.



Input Dataset: c:\temp\all2 .sas7bdat

Condition to Subset Input Dataset (optional): (eg: itt = 1 and tte="OS")

Time Variable: time

Group Variable: group

Censor Variable: censor

Censor Indicator: 1

Time Unit: Month

X Start:

Y Start:

Tick Mark Interval: 6 Unit of Time

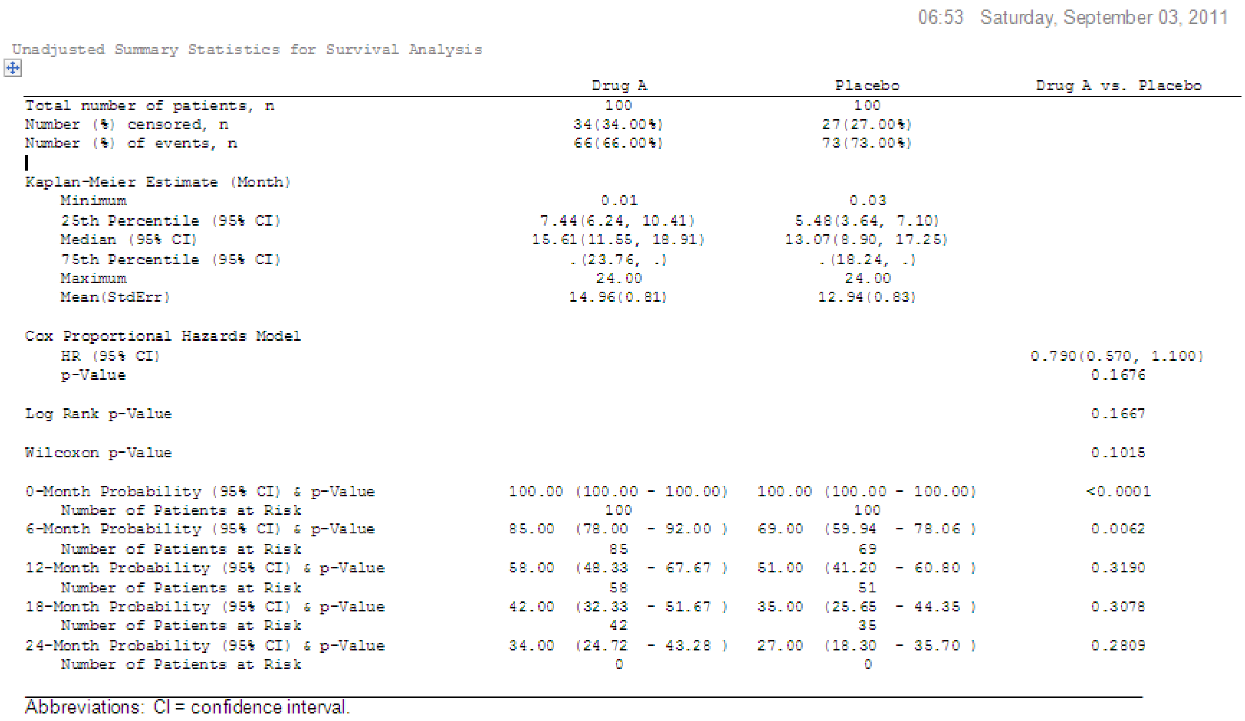
Group Labels for Curves: Drug A|Placebo

Curve Colors: black|red

Group Labels for Risk Count: A|PLA

Output Report: c:\temp\kmreport .rtf

Figure 2. User Input to Compare Two Treatment Groups for Summary Statistics of Survival Analysis



```
                                                                              06:53   Saturday, September 03, 2011
Unadjusted Summary Statistics for Survival Analysis

                                                  Drug A                  Placebo              Drug A vs. Placebo
Total number of patients, n                         100                      100
Number (%) censored, n                          34(34.00%)               27(27.00%)
Number (%) of events, n                         66(66.00%)               73(73.00%)

Kaplan-Meier Estimate (Month)
    Minimum                                        0.01                     0.03
    25th Percentile (95% CI)                  7.44(6.24, 10.41)        5.48(3.64, 7.10)
    Median (95% CI)                          15.61(11.55, 18.91)      13.07(8.90, 17.25)
    75th Percentile (95% CI)                   .(23.76, .)              .(18.24, .)
    Maximum                                        24.00                    24.00
    Mean(StdErr)                                 14.96(0.81)              12.94(0.83)

Cox Proportional Hazards Model
    HR (95% CI)                                                                            0.790(0.570, 1.100)
    p-Value                                                                                      0.1676

Log Rank p-Value                                                                                 0.1667

Wilcoxon p-Value                                                                                 0.1015

0-Month Probability (95% CI) & p-Value   100.00 (100.00 - 100.00)  100.00 (100.00 - 100.00)     <0.0001
    Number of Patients at Risk                       100                      100
6-Month Probability (95% CI) & p-Value   85.00  (78.00  - 92.00 )  69.00  (59.94  - 78.06 )      0.0062
    Number of Patients at Risk                       85                       69
12-Month Probability (95% CI) & p-Value  58.00  (48.33  - 67.67 )  51.00  (41.20  - 60.80 )      0.3190
    Number of Patients at Risk                       58                       51
18-Month Probability (95% CI) & p-Value  42.00  (32.33  - 51.67 )  35.00  (25.65  - 44.35 )      0.3078
    Number of Patients at Risk                       42                       35
24-Month Probability (95% CI) & p-Value  34.00  (24.72  - 43.28 )  27.00  (18.30  - 35.70 )      0.2809
    Number of Patients at Risk                       0                        0

Abbreviations: CI = confidence interval.
```

Figure 3. The output Summary Table for Two Treatment Groups

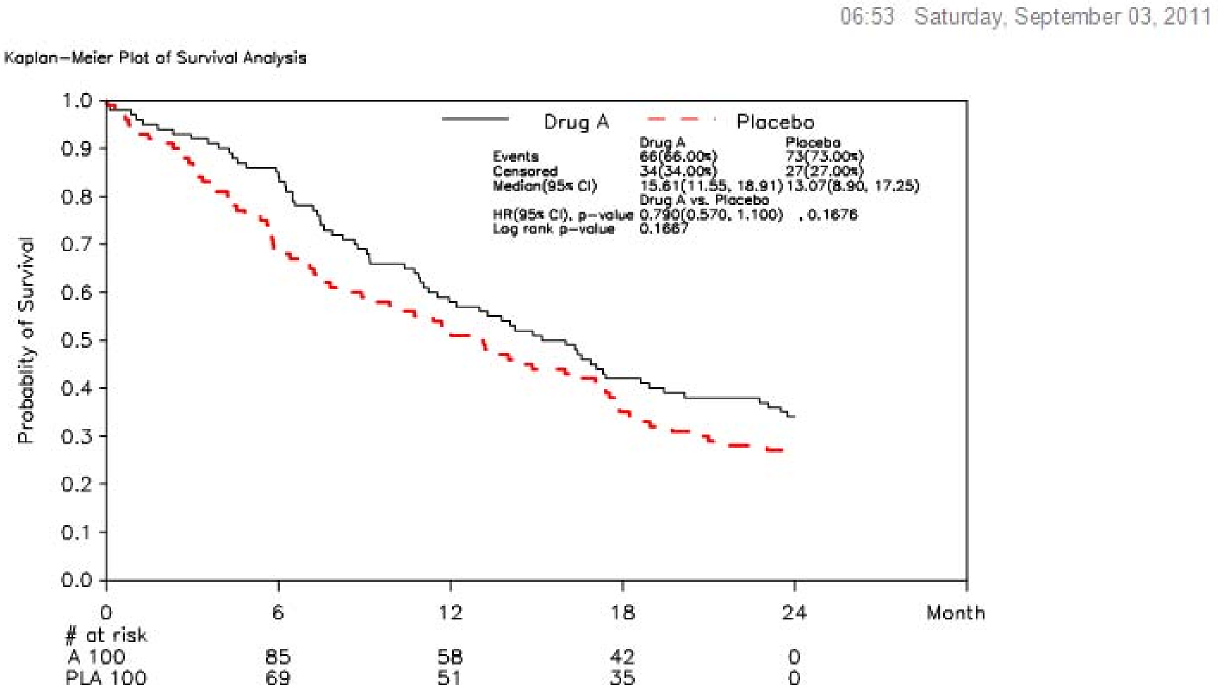
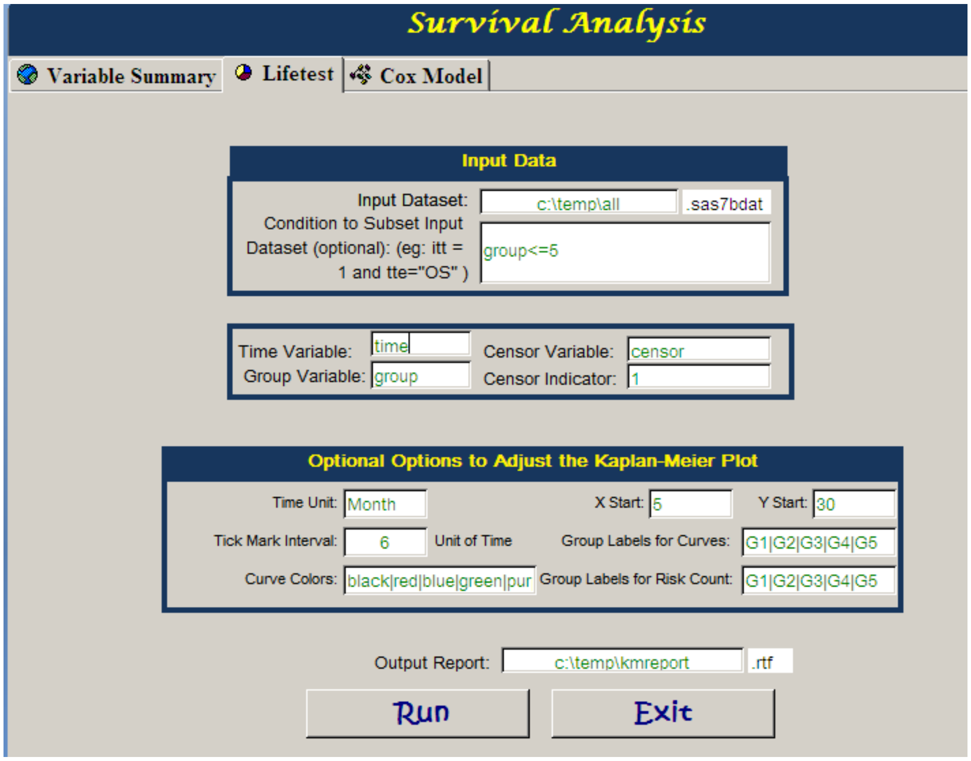


Figure 4. The Output Kaplan-Meier Curves for Two Treatment Groups

Example2 – Create Kaplan-Meier Plot for More Than Two Groups

Similar to two group comparison, the application can perform analysis for up to nine groups. Figure 5 and 6 show the usage example and the output graph.



Input Data

Input Dataset: c:\temp\all .sas7bdat

Condition to Subset Input Dataset (optional): (eg: itt = 1 and tte="OS") group<=5

Time Variable: time

Group Variable: group

Censor Variable: censor

Censor Indicator: 1

Optional Options to Adjust the Kaplan-Meier Plot

Time Unit: Month

X Start: 5

Y Start: 30

Tick Mark Interval: 6 Unit of Time

Group Labels for Curves: G1|G2|G3|G4|G5

Curve Colors: black|red|blue|green|pur

Group Labels for Risk Count: G1|G2|G3|G4|G5

Output Report: c:\temp\kmreport .rtf

Figure 5. User Input to Analyze Five Groups

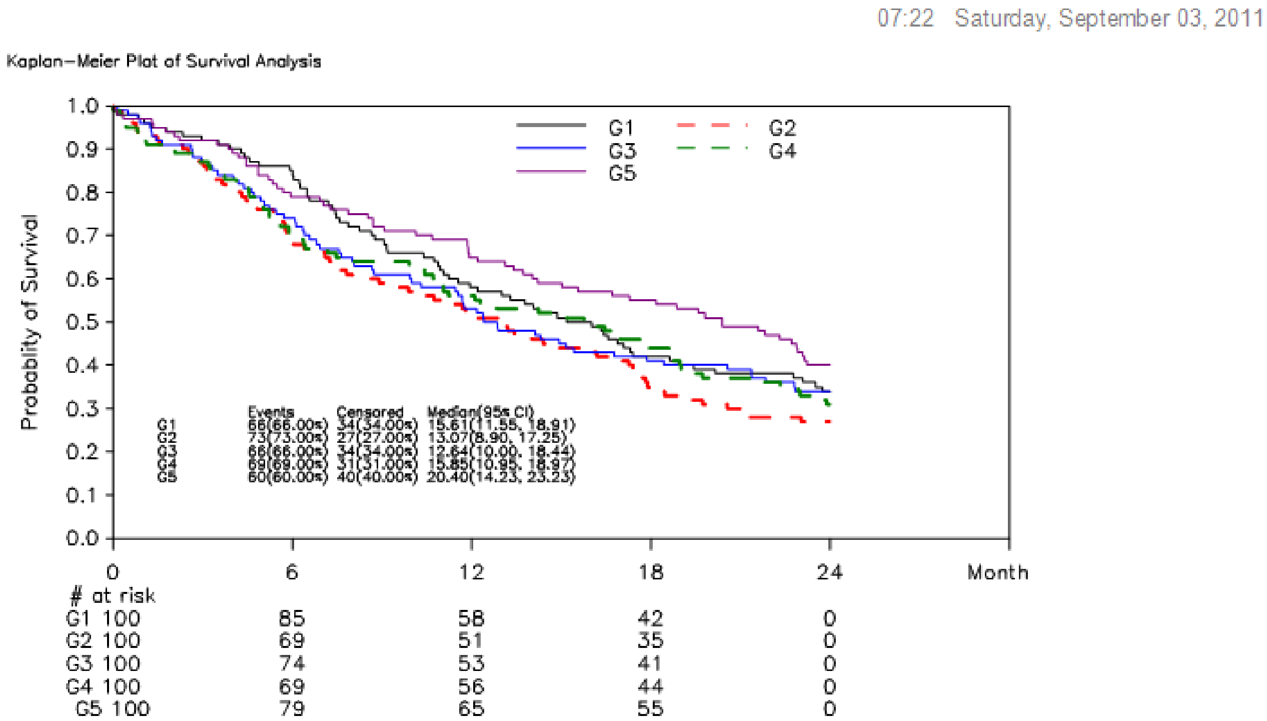


Figure 6. The Output Kaplan-Meier Curves for Five Groups

VBA Code to Drive Survival Analysis

After the button “Run” on the tab “Lifetest” is clicked, the application will check the input on this tab for the text boxes and extract needed information, for example, input dataset path and output path. The possible input errors will be handled properly, for example, missing information or path does not exist.

VBA Code for Text Box Events

Below is the example of VBA code for Lost Focus event associated with text box “txtoutfile”. This text box accepts user’s input for output report path and file name, which is related to the place to store the analysis result.

```
Private Sub txtoutfile_LostFocus()

  Dim outpath As String, pos As Integer, inputtrim As String

  If Nz(Me.txtoutfile, "") & "" <> "" Then

    inputtrim = Trim(Me.txtoutfile)
    pos = InStrRev(inputtrim, "\")

    If Len(inputtrim) = 0 Then
       MsgBox "Error: blank report name."
    ElseIf pos = 0 Then
       outpath = "" & Application.CurrentProject.path
       outfile = inputtrim
    ElseIf pos = Len(inputtrim) Then MsgBox "Error: no report name provided, & _
    please enter report name for your output.", vbCritical, "Invalid information"
    Else
       outpath = Mid(inputtrim, 1, pos)
       If FolderExists(outpath) = False Then MsgBox "Error: output path " & _
       & outpath & " does not exist."
    End If
  Else:  MsgBox "Error: Blank output path and report name.", vbCritical, & _
  "Missing information"
  End If

End Sub
```

Another example of VBA Code for Got Focus and Lost Focus events associated with text box “txttimeunit” is given below. It controls when the label “lbltimeunit” should be visible or invisible to the user.

```
Private Sub txttimeunit_GotFocus()
  lbltimeunit.Visible = True
End Sub

Private Sub txttimeunit_LostFocus()
  lbltimeunit.Visible = False
End Sub
```

VBA Code to Generate SAS Program

VBA code can be used to write text stream to an external file. The VBA code below shows how the application generates the output file with the extension of “.sas”. The file is stored in the same directory as the “.rtf” file entered by the user to store the analysis result. “ofile” in the code stores the extracted output path and file name without file extension from the GUI. “myfile” adds the file extension “.sas” to “ofile”. The “print” statement writes the text stream line by line to the specified “myfile”. The VBA variables (dspath, ds, filter, vartime, varcensor, strsetgrp, censoriind, timeuit, tick,grplbl, risklbl, clor, x0, y0, ofile, and outpath) used in the code below are assigned values based on the user’s input from the GUI. The system might assign some optional fields to default values when they are blank.

```
Dim myfile As String
myfile = ofile & ".sas"
fnum = FreeFile()

Open myfile For Output As fnum

Print #fnum, "%include '" & curdir & "kmsurv.sas'; " & vbCrLf
Print #fnum, "libname db '" & dspath & "'; " & vbCrLf
Print #fnum, "data kmdata; set db." & ds & "; if " & filter & "; time=" & _
& vartime & "; censor=" & varcensor & "; run;" & vbCrLf
Print #fnum, strsetgrp & vbCrLf
Print #fnum, "%kmsurv(ds=kmdata, timevar=time, censorvar=censor, showhr= & _
Y, showlogrankp=Y, " & vbCrLf & "censorflagind=" & censorind & ", timeunit & _
=" & timeunit & ", timeatick=" & tick & ", annolabel = " & grplbl
Print #fnum, ",risklabel = " & risklbl & ", curve_color = " & color & _
& ", anno_xstart= " & x0 & ", anno_ystart = " & y0
Print #fnum, ", outrpt= " & ofile & ".rtf, temppath= " & outpath
Print #fnum, ", titlefontsize=0.8, title1=Kaplan-Meier Plot of Survival Analysis); "
Close #fnum
```

Example SAS Program File Generated from the VBA Code

```
%include 'C:\temp\kmsurv.sas';

libname db 'c:\temp\';

data kmdata; set db.all2; if 1; time=time; censor=censor; run;

data kmdata; set kmdata; figgroup =group; keep time censor figgroup;run;

%kmsurv(ds=kmdata, timevar=time, censorvar=censor, showhr= Y, showlogrankp=Y,
censorflagind=1,timeunit=Month, timeatick=6, annolabel = Drug A|Placebo,
risklabel = A|PLA, curve_color = black|red, anno_xstart= 45, anno_ystart= 92,
outrpt= c:\temp\kmreport.rtf, temppath= c:\temp\,
titlefontsize=0.8, title1=Kaplan-Meier Plot of Survival Analysis);
```

VBA Code to Run SAS Program

VBA code below first stores SAS executable (sas.exe) including its path in a string variable “sas”. Then it calls the function “RunSAS” to launch SAS executable to run “.sas” program generated by VBA. The function “RunSAS” uses VBA function “shell” to launch SAS application from MS Access. Based on RunSAS call, log file will be generated. The output of RTF file will also been generated based on the SAS program.

```
Dim mylog As String
mylog = ofile & ".log"

Dim sas1 As String, sas2 As String, sas3 As String, sas As String
sas1 = "c:\program files\sas\sasfoundation\9.2\sas.exe"
sas2 = "c:\program files\sas\sas 9.1\sas.exe"
sas3 = "c:\program files (x86)\sas\sas system\9.0\sas.exe"

If FileExists(sas1) Then
  sas = sas1
ElseIf FileExists(sas2) Then
  sas = sas2
ElseIf FileExists(sas3) Then
  sas = sas3
Else
  sas = InputBox("Input your SAS executible file sas.exe path and name:")
End If

Dim retVal
retVal = RunSAS(sas & " " & myfile & " -log " & mylog, 1)
```

Conclusion

This paper introduced the application Survival Analysis including the design of the GUI, VBA code to drive the analysis, and how to use the application through examples. There are a few points that the application can be improved in the future. For example, more text fields can be added in the tab to allow more customization choices for the analysis. Another improvement that can be achieved very easily is, if the SAS executable is not among the three paths (sas1, sas2, and sas3) the application defined, it is more convenient to store the user’s input SAS executable in the application so that the user does not need to re-input SAS executable path next time unless the path has been updated.

Acknowledgements

I’d like to thank Baoguang Han for his valuable suggestions on this application.

Contact Information

Your comments and questions are valued and encouraged. Contact the author at:

Zhong Yan
i3
Office: (317) 564-2858
E-mail: ohzhong@yahoo.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies.