## •• DESCRIPTION OF MACRO COMMANDS ••

This commands allow the user to perform fully automated survival analysis and to select,transform or recode the variables in the active data array. See the examples in the Survival Macros file to learn how to use this commands. You may use small letters, capital letters or both to define these commands. It also allows the computing of descriptive statistics for the data array in memory and the exact P value for normal, chi-square, Student and F probability distributions.

•BACKCOLOR ( 'white', 'black' ) sets the background color for graphical plots. You must put one of the arguments in the list above;otherwise you'll get an error message. The Type is STRING.

•BEEP; beeps for one second.

•CHIPROB(chi,df); computes the exact P value of ChiSquare distribution 'chi' with 'df' degrees of freedom.

•CATEGORIES(var1,var2,..option). Indentifies the categorical variables in the data array included in the model that have to be transformed into DEVIATION (option = 0) or INDICATOR (option = 1) variables.

•CHITEST(var1,var2); computes a Chi-Square Test for categories in the 'var1' column of the data array by the categories in the 'var2' column of the data array. Results are presented in a contingency table where the rows are the max number of categories in 'var1' and the columns the max number of categories in 'var2' The maximum number of categories (n) is seven and the categories should be enumerated from 1 ......to n categories.

•CLEAR; clears the active output window. This option is not available with the

Macro Command submenu if you use System 6.xx

•COMPUTE. This command is used to recode or transform variables in a speedy fashion. It has the following syntax's:

```
COMPUTE

    statement ;

    statement ;
```

```
etc.      ;

END;
```

See the examples in the Survival Macros file to learn how to use this command.

- COUNTIF( Boolean expresion ) count the number of cases in the active data array that match the conditions specified in the Boolean argument. Use the rDATA[ ] token to access the variables in the active data array. The result of the count command is displayed on the active Output Window.

Example:

```
COUNTIF((rData[1] > 50) and (rData[2] =3));
```

- COVARIATES (number, number ....) set covariate pattern values for the current survival test. The Type is REAL.

- CREATE ( filename, Var, Var, Var ... ) . Creates a new File filename with the variables specified in the argument list.

Example: CREATE ('HD80:Data:MyNewFile', 1,2,3,7,8);

- ESTIMATE; starts estimation of Mle's and survival functions. It has no arguments.
  You must always specify the column number of the status variable and time variable with SETSTATUS(n) and SETTIME(n) before issuing this command; otherwise you'll get an error message.

- EXIT; terminates macro execution.

- FIXDATA; makes permanente the changes in the data array. You need to reload your data file if you want your original data back.

- FORMAT(DataLength,DataDecimals) sets the data length and the number of figures after the decimal point for numerical output.

- FPROB(F,df1,df2); computes the exact P value of F distribution F with df1 and df2 degrees of freedom.

- FUNCTION . This command has the following syntax's:

```
FUNCTION

      statement ;

      statement ;

etc.    ;

END;
```

It must be only used to compute the value of the variables at each failure point when you chose a Time Dependent Analysis by pressing the Option Key while selecting the variables to be included in the model or when using the macro command VARCODE(number, number) within a Macro. See example in the Macro Time1 macro file to learn how to use this command.

- GET ( Rows, Columns, nVars ); gets the number of rows, number of columns and the number of selected vars of the active data array. The Type is INTEGER;

- GETDATA( 'filename' ) loads data from a binary file or a tab delimited text file. If the data file is in a different folder or volume you must specify the full path to access the file. If not, a dialog box is shown. The TYPE is STRING.

- GROUPLABEL(Val:'string'; Val: 'string' ....) puts labels to the groups for both numerical and graphical output. Val is the value of the categorial variable that identify the different groups in stratified analysis.

- INTPLOT(option); actives the plot of the Confidence Intervals for the current survival function. It must be used after the ESTIMATE command. option is 1 for 95% CI and 2 for 99% CI.

- LIST ( length, decimals ); Updates the output window and list the entire data array with the format given by the two arguments. The Type is INTEGER.

- LOGPLOT; activates the plot of Log -(Log) of Survival function, which is used for testing the proportionality assumption.It must be used after the ESTIMATE macro command.

- MACRO; command needed to identify a Macro procedure. The text in quotes after the word MACRO is added as a new command in the MACRO Menu.

Example:    Macro 'My Macro';

- MAKEBINS ( VarIn,VarOut,Option,Bin1,Bin2,Bin3 ...... BinMax   ). This command

recodes a numerical variable into categories. VarIn   is the source column number. VarOut specifies the column number where to put the newly recoded variable. If VarOut   equals to VarIn , all data in the source column will be overwritten. Option  is a Boolean variable; when is set to true the first category of the new coded variable corresponds to those values of the numerical variables that   are less than the value specified in Bin1 and the last category to those values that are equal   or greater than the   value given in BinMax.
THE MAX NUMBER OF BINS THE COMMAND CAN CREATE   IS 30.

Example 1:   MAKEBINS (3,3,true, 40,50,60,70) will recode the variable in column 3 and put the result in the same column as follows:

Value

Category Nº

< 40

1

40 - 49

2

50 - 59

3

60 - 69

4

70 =>

5

Example 2: MAKEBINS(3,4,false,40,50,60,70) will codify the variable in column

3 and put the result in column 4 as follows:

| Value | Category Nº |
|---|---|
| 40 - 49 | 1 |
| 50 - 59 | 2 |
| 60 - 69 | 3 |

This second option should cover the whole value range of the variables. If not, you may get a combination of both categorical and numerical variables in the same column. If not, you're warned with a message.

•NEWVARS ( number, default ) . This command appends to the right of the active data array new number columns with the default value. It is used to create dummy variables to store, for example, the transformed values of other variables. Example NEWVARS( 2, 0);

•NORMPROB(Z); computes the exact P value of Normal distribution 'Z'

•OMITIF(Boolean argument) ; excludes from the active data array the cases that meet the conditions specified in the Boolean argument. Use the rDATA[ ] token to access the variables in the active data array.

•PARAMETRIC('ModelType', option1, option2, option3, option4, sigma);

"Model Type" is the the type of Parametric Model:

-Set 'ModelType' to 'exponential' to fit data to a exponential model

-Set 'ModelType' to 'weibull' to fit data to a two Parameter Weibull model

-Set 'ModelType' to 'lognormal' to fit data to a Log-Normall model

'option1', 'option2' and 'option3' are boolean expressions:

-Set 'option1' to true if you want to use a constant term. This option should always be true if you include covariates in the model

-Set 'option2' to true to center the covariates values around their mean value

-Set 'option3' to true if you want the iterations to be printed in the output window

-Set 'option4' to true if you want the Cumulative Distribution insted of the Survival Function to be plotted for the Survival Function adjusted for the effect of the covariables in the model.

'sigma' is the initial value for estimation of the model's sigma term. Should be set to 1 unless you have convergence problems

Example: parametric('weibull',true,false,false,true,1);

•RECODE ( Var,OldValue,NewValue, [Default Value] ). This comand easily recodes the OldValue for the variable number Var with NewValue. Default Value is optional; if you specify it, the variables with values different to OldValue take its value. Is the same as if you used the following Pascal statement within a standard procedure :

'If   OldValue   then NewValue  else DefaultValue'.

You may use either the name of a previously defined variable with the column number you want to transform, the column number itself or a matrix definition (rVar[column number] or rData[column number]).

Example: The following commands will render the same result:

```
RECODE(1:1:2;5:1:2:-1);

  grade:= 3; stage:= 5;

 RECODE(grade:1:2;stage:1:2:-1);

 Recode(rData[3]:1:2;rData[5]:1:2:-1);
```

You may use colons or commas as delimiter of the arguments. Use a semicolon as delimiter when you   recode different variables with a single command.

•RESET; this command set different flags and constants needed for the survival

algorithm. It has no arguments and should be the first command in a procedure that uses macro commands, before setting the parameters for the test in course.

•RESPLOT; activates the plot of the Cumulative Function for Residuals, which is used to Test the goodness of fit of data to model. It must also be used after the ESTIMATE macro command.

•REVERT; restores the values in the active data array to the values contained in the last loaded data file. It has no arguments. You'll appreciate this macro when you need to transform, recode or select your original data for a given task.

•RUNMACRO(number); excutes the macro located in the item number in the Macro Menu. Useful for linking different macros in complex automated survival analysis. Read carefully the macros in the 'Survival Macros' to learn how to use these command.

•SAVEDATA('Filename',['text','binary' ]) saves data in the active data array as a Tab delimited text file (text) or as a binary application file (binary). You must   put   one of the string arguments in the list above, otherwise you'll get an error message. If you omit the Filename a standard file dialog box is presented.

•SAVEMODEL; Saves the parameters of the current estimated model for further use with the Survival Chart option of the Survival Menu. It has no arguments.

•SAVEPICT('name') saves the picture displayed in the active plot window to disk in the same folder than the application. This command should be used after the SURVPLOT, INTPLOT, LOGPLOT and RESPLOT commands, if you pretend to make different survival analysis with a single macro and want to save the graphical output.This macro also closes the active Plot window after saving to the disk.
If you use an empty string as the name argument, until you start a new analysis the graphic plots ar saved as Kaplan-Meier.PICTn, Model.PICTn, LogLog.PICTn, Intervals.PICTn and Residuals.PICTn, where 'n' is automatically increased for each saved file of the same graphic type.
Use a Path if you want the pictures saved in a different folder
This option is not available with the Macro Command submenu

•SAVETEST('filename'). Saves in the file filename the numerical results of the

current Test in the active Output Window. You should use this macro when the output window exceeds 37000 characters, the maximum allowed for text files.

•SELECTIF ( Boolean argument ). Selects from the active data array the cases that meet the conditions specified in the Boolean argument. As with the OMITIF command, use the rDATA[i] token to access the variables in the active data array.

•SELECTVAR(number, number, ..); this command selects the variables to be included in the regression model. Number specifies the column number of the data file where variables are located. The Type is INTEGER.

•SET( Rows,Columns,nVars ); sets the new number of Rows (cases), Columns (variables) and the number of selected vars (nVars), if any, for the active data array.The Type is INTEGER.

•SETCUTPOINT(number, varnumber). Sets cut point value number.
Varnumber is the column number of the variable. The Type is REAL for the first argument and NTEGER for the second argument.

•SETFAILCODE (number). Sets the code identifying the terminal event. The Type is INTEGER

•SETITER (number) sets number of max iterations for the Newton-Raphson algorithm. The Type is INTEGER.

•SORTBY(column1,column2).Sorts data array in ascending order by column1 and column2. If you put a zero for the value in column2 the data array is only sorted by the ascending order of values in column1.

- SURVPLOT. Activates the plot of both Kaplan-Meier and Model Survival functions.
  It must be used before the ESTIMATE macro command. This macro substitutes the GRAPHPLOT('true','false') command of previous versions.

- VARLABEL(Var:'string'; Var: 'string' ....) puts labels to the selected variables.
  Var is the column number of the variable in the active DATA ARRAY.

- SETOUTPUT('all', 'none', 'basal', model'). Selects numerical output to be shown on the active output window. You must put one of the arguments in the list above; otherwise you'll get an error message. The Type is STRING.

- SETPRECISION( number). Sets the level of precision for estimation of the coefficients of the regression model. The Type is REAL.

- SETSTATUS(number); sets the column number where the status variable is locate in the active data array. The Type is INTEGER.

- SETSTRATA (number, stratavar) .Sets the number of strata and the column number of the strata variable stratavar to perform stratified analysis.
  The Type is INTEGER.

- SETTIME( number ).Sets the column number where the survival times are located in the active data array. The Type is INTEGER.

- SETUNIT( 'days', 'weeks', 'months', 'years', 'user' ).Selects Time unit for numerical and graphical output. You must put one of the arguments in the list above; otherwise you'll get an error message. The Type is STRING.

- STAT(var,var, ...).Computes descriptive statistics (average, SD, SE, Min,Max and Range) for the specified var. If you use an asterix sign as argument (*) the statistics for all variables in the active data array are computed.

- TITLE ('My Title' ).Puts a custom title to the survival graph plot. The Type is STRING.

- TPROB(T,df).Computes the exact P value of Student T distribution T with df degrees of freedom.

- TTEST(var1,var2,[Var3]);computes a Student T-Test betwen var1 and var2.
  If you use the same value for var1 and var2 and use var3, which is optional, a T-Test is computed for the groups in var3. The number of groups must be two (use the RECODE or COMPUTE MACRO if necessary) otherwise you'll get an error message.

•VARCODE (number, number..) identifyes type of variable for time-dependent

analysis. number is 1 for fixed covariates and 2 for time-dependent covariates.
The order should be the same than the selected variables of the model.
Use this command only within a macro to activate Time-Dependent analysis.
You don't need to use this command in a macro if you employ only FIXED covariates.

•WRITE( arguments, arguments, ..) writes the arguments to the active Output Window. The arguments may be a string, a variable, a number, a formatting string(eg. '#F6:2'), a Tabulator ('tab') or Carriage Return ('cr');

Example:

```
TimeAverage:= 560; you may compute this value
        Location:= 2
```

```
WRITE( 'The average survival time is','#F8:2',TimeAverage,' and is located in
        Column', '#F2:0',location, 'cr');
```

--------------------------------------------------------------------------

You may acces the numerical results of the Statistical Tests for further processing within a standard pascal procedure or macro with the TEMP[] Token as follows:

```
=================================================================
==
```

| | Temp[1] | Temp[2] | Temp[3] | Temp[4] | Temp[5] | Temp[6] |
|---|---|---|---|---|---|---|
| -CHITEST | Chi-Value | P-value | df | | | |
| -TTEST | T-Value | P-Value | df | | | |
| -CHIPROB | Chi-Value | P-Value | df | | | |
| -NORMPROB | Z-Value | P-Value | | | | |
| -TPROB | T-Value | P-Value | df | | | |
| -FPROB | F-Value | P-Value | df1 | df2 | | |
| -STAT(x) | Mean(x) | SD(x) | SE(x) | Min(x) | Max(x) | Range(x) |

----------------------------------------------------------------------------

Example

```
if Temp[2] < 0.05 then RUNMACRO(2) else EXIT;
```

•Standard Pascal Commands:

-Procedure

-Begin...end

-For ...To...Do

-Repeat Until

-While...Do

-If...Then...Else

-Var

•Operators:

, ,+, -, *, /, DIV, MOD, :=, =, >=, <=, < >, =

•Logical Operators:

-AND

-OR

-NOT

•Pascal Types:

-Integer; internally stored in extended precision

-Real; internally stored in extended precision

-Boolean: true,false

•Pascal Standard Functions

-trunc(x)

converts real (x) to integer with truncation

-round(x)

converts x to integer with rounding

-random

generates a random number between 0 and 1

-odd(x)

returns true if integer x is odd

-abs(x)

absolute value of x

-sqr(x);

square of x

-sqrt(x)

square root of x

-sin(x)

sine of x

-cos(x)

cosine of x

-arctan(x)

arctangent of x

-ln(x)

natural logarithm of x

-exp(x)

exponential of x

•• CUSTOM ARRAYS ••

Six different arrays are internally defined for data manipulation;

1) rVAR[k] ; vector of length MaxVars (30). Used to temporarily store the value

of the kth selected variable for a given test. You must use this token to access the VAR values within the COMPUTE ,FUNCTION, SELECTI, OMITIF, RECODE and COUNTIF custom procedures.

2) rDATA [k] ; vector of length MaxVars (30). Used to temporarily store the value of the kth variable in the active data array. You must use this token to access the VAR values within the above mentioned custom procedures.

3) gVAR[offset]; the same as rVAR but it must be used within a macro or a standard Pascal procedure.

4) gDATA[offset] ;the same as rDATA but it must be used within a macro or a standard Pascal procedure. The Type of these arrays is REAL.

As Total Data and Selected Data are stored in a one-column array of length (rows * columns) to access the ith row of the kth column through the gVAR and gDATA tokens with a standardd Pascal procedure you must use the routine listed below. Notice that the structure of the routine follows the Pascal convention for a standard procedure.

```
macro 'My Macro'
  var

rows,columns,nvars,index,i:integer;
  myValue:real;

begin

GET(rows,columns,nvars); get number of rows, columns and selected vars

for i:= 0 to rows -1 do

  begin

     index:= i * columns + 3;access the value in column 3 of the ith row
                   myValue:= gDATA[index];

      etc;


   end;

end;
```

5) rMEAN [k]; vector of REAL Type and MaxVars(30) length which stores the vector of average values for the kth selected variable of the ith individual. Currently used with the Time Dependent option.

6) TEMP [k]; Vector of REAL Type and MaxVars(30) length which stores the code of the kth selected variable for the model of the ith individual or case. This value is = 1 for Fixed Covariates and = 2 for Time Dependent Covariates. You may use TEMP[k] for the temporary storage of any value or variable within a standard Pascal procedure.