SAS API for RSCORE Dynamic-Link Library (RSCORE DLL)
[version RSCR190A.dll]

This document describes the API for calling the RSCORE DLL from SAS 9.1.3. Familiarity with SAS programming in general and `PROC IML` in particular is assumed. The following table describes the argument list for calling the `RSCORE_SAS` subroutine that is exported by the DLL. The DLL was compiled with Lahey-Fujitsu Fortran 95 mixed-language options. To successfully call the DLL from your SAS program, you will also need the `RSCORE_20080411.tbl` and `RSCR190A.dll` files.

Arg	Variable	Description	Fortran Type	SAS Type
1	`normals`	rating frequencies for normals (array)	`REAL*8(101)`	`RB8.`
2	`abnormals`	rating frequencies for signals (array)	`REAL*8(101)`	`RB8.`
3	`catnames`	integer labels for categories (array)	`INTEGER*4(101)`	`IB4.`
4	`numcat`	number of categories (scalar)	`INTEGER*4`	`IB4.`
5	`numnorm`	sum of normal frequencies (scalar)	`REAL*8`	`RB8.`
6	`numabnorm`	sum of signal frequencies (scalar)	`REAL*8`	`RB8.`
7	`A`	y-intercept for RSCORE model (scalar)	`REAL*8`	`RB8.`
7	`B`	slope for RSCORE model (scalar)	`REAL*8`	`RB8.`
9	`area`	area under ROC curve (scalar)	`REAL*8`	`RB8.`
10	`cutpoints`	cutoffs/cutpoints/thresholds (array)	`REAL*8(101)`	`RB8.`
11	`sens`	sensitivity (scalar)	`REAL*8`	`RB8.`
12	`spec`	specificity (scalar)	`REAL*8`	`RB8.`
13	`retcode`	return code from RSCORE DLL (scalar)	`INTEGER*4`	`IB4.`
14	`debug`	turn on/off debugging (scalar)	`INTEGER*4`	`IB4.`

Notes:

1. All arguments are required. There are no optional arguments.

2. You must initialize **ALL** variables passed as arguments to the DLL (even if you are only interested in returned values). For example, you must initialize `A`, `B`, `auc`, `cutoffs`, and `retcode` to some value such as `0` for `long` variables and 0.0 for `double` variables prior to the call. **See a separate note below for initializing sens and spec.** It is not enough to declare and allocate the variables—they must be initialized. If you fail to do this, the DLL may crash or exhibit unpredictable behavior.

3. The `sens` & `spec` variables are used to choose a sensitivity [p(TP)] at a given specificity [1-p(FP)] or specificity and a given sensitivity analysis. These variables also return the sensitivity or specificity. The following rules should be followed:

   If you want only an area analysis and do not care about sensitivity or specificity, then set both variables to -1:
   `sens` = -1.0
   `spec` = -1.0

   If you want to compute sensitivity at a given specificity, then set `sens` to -1 and `spec` to the target specificity [1 – p(FP)]:
   `sens` = -1.0
   `spec` = target specificity value (range 0.0 < `spec` < 1.0)

If you want to compute sensitivity at a given specificity, then set `spec` to -1 and `sens` to the target sensitivity [p(TP)]
`sens` = target sensitivity value (range 0.0 < `sens` < 1.0)
`spec` = -1.0

Note that the above ranges specify "<", not "<="! Target values cannot equal 0.0 or 1.0.

If you set both sens and spec to values greater than -1, an error condition will result (`retcode` will be value greater than 0).

4. All arguments must be passed by reference.

5. If the DLL successfully computes values, the return code should be 0 (`retcode` = 0). Any value other than 0 indicates an error condition, so you should check `retcode` after each call to the DLL.

6. The debug variable is used to turn on logging for the DLL. Normally, you should set `debug` = 0. If you set `debug` = 1, the DLL will create a text log for each call. This log contains a listing of the input values to the DLL as well as any output values that were computed. This debug log will (usually) be created in the directory that contains the RSCORE DLL file.

7. You will need to place `RSCORE_20080411.tbl` in a directory that is specified in the `filename sascbtbl` statement in your SAS program. If you fail to do this, you will get erroneous output (for an example see the "`Output if TBL not found.lst`" file included with this package).

8. You will need to place `RSCR190A.dll` in a directory that is specified in your system PATH variable. If you fail to do this, you will get a warning message in the SAS log file and erroneous output (for an example see the "`Output if DLL not found.lst`" file included with this package).

Last Edit:
Kevin M. Schartz, Ph.D., M.C.S.
April 24, 2008