CMAT Newsletter: December 2006

Wolfgang M. Hartmann

December 2006

Contents

1 General Remarks

Some bugs are fixed - again. Among others, some related to the `glim` function and the ROC curve output.

1.1 New Functions

The following new functions are implemented:

byte transfers integers into characters using the ASCI table

branks computes tied and bivariate ranks

covlag computes autocovariance estimates for a vector time series

gee computes longitudinal generalized linear regression model using the method of GEE (generalized estimation of equations)

modtst testing the model statement

ranktie rank values of a vector averaging ranks with ties

roccomp equality test of the areas under ROC curves for different logistic models with binary response

scad Smoothly Clipped Absolute Deviation (Fan and Li, 2001, 2002)

smsvm (Structured) Multinomial SVM algorithms by Lee and Wahba

toeplitz generates (block) Toeplitz matrix

1.2 Fixed Bugs

1. The `glim()` function did not return the correct ROC curve content.
2. A rather serious bug was corrected concerning the comparison of string data. Statements like this

   ```
   i = (resp[3] == "A") ? 1 : 0;
   ```

   now return the correct value.
3. The format of matrix output was improved.

2 Modifications of Features

2.1 New Language Feature

For all functions that use a model string specification, the `at`, @, operator for restricting the level of interactions when using multiple `bar`, |, operators was implemented.

2.2 Extension to Various Functions

For the functions `svm`, `nlkpls`, `smsvm`, and `scad` the new option `"cval"` can be specified and have one of the following four string values:

"loo" for leave-one-out (Loo) cross validation

"blo(ck)" for block sampling cross validation

"spl(it)" for split sampling cross validation

"ran(dom)" for random sampling cross validation

"set" for validation data set cross validation (this option is valid only if in addition to the training data a test data set is specified for input)

2.3 Extensions to `glim()` Function

<gof,parm,sterr,conf,cov,typ1,typ3,yhat,roc> =

= glim(data,model<,optn<,class<,xini<,con>>>>)

A new output of the ROC curve is added and the area under the ROC curve is added to the `gof` return argument.

```
data = [ 0 14 25,
         0 20 35,
         0 19 45,
         7 18 55,
         6 12 65,
        17 17 75];
cnam = [ "disease" "n" "age" ];
data = cname(data,cnam);


/* [1] disease / [2] n = [3] age */
modl = "1/2 = 3";
optn = [ "print"           5,
         "link"      "logit",
         "dist"      "binom",
         "trial"           2,
         "tech"     "trureg" ];
< gof,parm,sterr,conf,cov,t1,t3,yhat,roc > = glim(data,modl,optn);


                      *****************
                      Model Information
                      *****************

                Number Valid Observations    6
```

```
Events/Trial Resp      [1]/[2]
N Independend Variables      1
Error Distribution    BINOMIAL
Link Function            LOGIT

Trial Variable Column        2
Significance Level:  0.0500000
Design Coding:       Full-Rank
Hessian Matx. for Optimization
Hessian Matx. for Covariance M
No Variable Selection Process
```

****
Model Effects
****

Intercept + X3

****
Simple Statistics
****

Column	Nobs	Mean	Std Dev	Skewness	Kurtosis
Y[1]	6	0.3148148	0.4016427	1.0873358	0.5039404
X[3]	6	50.000000	18.708287	0.0000000	-1.2000000

****
Parameter Information
****

Parameter	Meaning
1	Intercept
2	age

****
Events/Trials Response Variable
****

Value	Nobs	Proportion
Event	30	30.000000
NonEvent	70	70.000000

****
Goodness of Model Fit
****

Log Likelihood	-24.234022	Degrees of Freedom	4
Deviance	7.7755662	Pearson ChiSquare	6.6020386
SSE	8.2937938	MSE = SSE/nobs	1.3822990
AIC (Intercept)	124.17286	AIC (All Param.)	52.468044
SBC (Intercept)	126.77803	SBC (All Param.)	57.678384
-2logL (Intercept)	122.17286	-2logL (All Param.)	48.468044
-2logL (ChiSqu.)	73.704817	Pvalue (df= 1)	0.0000000
Score ChiSqu. Test	55.327354	Pvalue (df= 1)	0.0000000
Wald Test	23.347513	Pvalue (df= 1)	0.0000014

****
Analysis of Effects and Parameter Estimates
****

Parameter	DF	Estimate	Std_Error	WaldChiSq	Pr>ChiSq
Intercept	1	-12.50164	2.555528	23.931660	0.000001
age	1	0.206587	0.042755	23.347513	0.000001
Scale	0	1.000000	0.000000	.	.

Covariance Matrix and ASE Based on Hessian Matrix

****
Confidence Limits of Parameters
****

Parameter	Estimate	LowWaldCL	UppWaldCL
Intercept	-12.501642	-17.5103843	-7.49289886
age	0.2065871	0.12278962	0.29038459

Wald Confidence Intervals Based on Hessian Matrix

****
Odds Ratio and Standardized Estimates
****

Parameter	OddsRatio	StndEst	LowerCL	UpperCL
age	1.2294748	2.130826	1.13064653	1.33694156

****
Evaluation of Training Data Fit
****

Index	Value	StdErr
Absolute Classification Error	13	.
Classification Accuracy	87.00000000	.
Concordant Pairs	92.61904762	.
Discordant Pairs	2.000000000	.
Tied Pairs	5.380952381	.
Goodman-Kruskal Gamma	0.944517834	0.032751331
Kendall Tau_a	0.384444444	.
Kendall Tau_b	0.687698721	0.079996134
Stuart Tau_c	0.572000000	0.086546173
Somers D C|R	0.680952381	0.084157569
Somers D R|C	0.694511899	0.083126388

Classification Table

	Predicted	
Observed	NoEvt	Event
NoEvt	64	6
Event	7	23

Area under the ROC Curve (Disc.) = 0.953095
Area under the ROC Curve (Cont.) = 0.953095

Nobs	Yobs	Ypred	LowerCL	UpperCL	Diag(H)
1	0	6.506e-004	3.403e-005	0.01230171	0.02063943
2	0	0.00511173	5.989e-004	0.04219523	0.12225035
3	0	0.03896968	0.01001222	0.13984749	0.35720391
4	7	0.24243955	0.12544301	0.41657825	0.55424696
5	6	0.71636875	0.51309867	0.85822603	0.48496607
6	17	0.95222920	0.81451290	0.98906917	0.46069328

```
SSE=8.29379 MSE=1.3823 Misclassification=13

Pearson ChiSquare=6.60204 Deviance=7.77557 (df=4)
```

The following is new output:

```
************************************************************
Receiver-Operating-Characteristic (ROC) Curve (Area=0.953095)
************************************************************

   N  Obs  Posit  Negat  FalsP  FalsN   Sensit 1-Specif   Prob
   1    6     17     70      0     13 0.566667 0.000000 0.9522
   2    5     23     64      6      7 0.766667 0.085714 0.7164
   3    4     30     53     17      0 1.000000 0.242857 0.2424
   4    3     30     34     36      0 1.000000 0.514286 0.0390
   5    2     30     14     56      0 1.000000 0.800000 0.0051
   6    1     30      0     70      0 1.000000 1.000000 0.0007


            *********************************
            LR Statistics for Type 1 Analysis
            *********************************

    Effect              Deviance    DF   ChiSquare  Pr>Chi

    Intercept        81.48038288     0     .          .
    age              7.775566201     1  73.7048167  0.0000


            *********************************
            LR Statistics for Type 3 Analysis
            *********************************

         Effect              DF   ChiSquare  Pr>Chi

         age                  1  73.7048167  0.0000
```

And the following output shows some of the return arguments:

```
print "Yhat=", yhat;
print "ROC=", roc;
print "GOF=", gof;
```

```
Yhat=
   |   N_event   N_nonev     Y_hat     Resid
-------------------------------------------
 1 |    0.0000   14.0000    0.0007    0.0091
 2 |    0.0000   20.0000    0.0051    0.1022
 3 |    0.0000   19.0000    0.0390    0.7404
 4 |    7.0000   11.0000    0.2424    7.9698
 5 |    6.0000    6.0000    0.7164    6.0000
 6 |   17.0000    0.0000    0.9522    0.8121

   |     Y_low     Y_upp      H_ii
---------------------------------
 1 |    0.0000    0.0123    0.0206
 2 |    0.0006    0.0422    0.1223
 3 |    0.0100    0.1398    0.3572
 4 |    0.1254    0.4166    0.5542
 5 |    0.5131    0.8582    0.4850
 6 |    0.8145    0.9891    0.4607


ROC=
   |      _obs     _prob      _pos      _neg
-------------------------------------------
 1 |    6.0000    0.9522   17.0000   70.0000
 2 |    5.0000    0.7164   23.0000   64.0000
 3 |    4.0000    0.2424   30.0000   53.0000
 4 |    3.0000    0.0390   30.0000   34.0000
 5 |    2.0000    0.0051   30.0000   14.0000
 6 |    1.0000    0.0007   30.0000    0.0000

   |   falspos   falsneg    sensit    1mspec
-------------------------------------------
 1 |    0.0000   13.0000    0.5667    0.0000
 2 |    6.0000    7.0000    0.7667    0.0857
 3 |   17.0000    0.0000    1.0000    0.2429
 4 |   36.0000    0.0000    1.0000    0.5143
 5 |   56.0000    0.0000    1.0000    0.8000
 6 |   70.0000    0.0000    1.0000    1.0000
```

The `gof` vector now also contains the area under the ROC curve:

```
GOF=
          |         1
----------------------
  RetCode |    0.0000
```

N_obs	6.0000
N_par	1.0000
RDF	2.0000
EDF	4.0000
LDF	1.0000
Likel0	61.0864
Likeli	24.2340
Devianc	7.7756
Pearson	6.6020
AIC	52.4680
SBC	57.6784
SSE	8.2938
MSE	1.3823
Misclas	13.0000
SCOR	55.3274
Pscor	1e-013
CRAT	73.7048
Pcrat	9e-018
ROCarea	0.9531

3 New Developments

3.1 Function `branks`

b = branks(a)

Purpose: The `branks` function computes tied and bivariate ranks. The tied rank of an entry x_i of N vector x is defined as

$$R_i = \frac{1}{2} + \sum_{j=1}^{N} u(x_i - x_j)$$

where

$$u(t) = \begin{cases} 1 & \text{if} \quad t < 0 \\ \frac{1}{2} & \text{if} \quad t = 0 \\ 0 & \text{if} \quad t > 0 \end{cases}$$

The bivariate rank of pair (x_i, y_i) of two vectors x and y is defined as

$$Q_i = \frac{3}{4} + \sum_{j=1}^{N} u(x_i - x_j)u(y_i - y_j)$$

Input: The input object `a` must be a 2 column matrix with real or integer values.

Output: The result `b` is is a three column matrix containing:

1. in its first two columns the tied ranks of the first and second column of the input matrix
2. in its third column the bivariate ranks among first and second column.

Restrictions:
1. Missing values, complex, and string data are not permitted.
2. The input object `a` should have two columns.

Relationships: ranktie()

Examples:
1. Example from SAS/IML Manual:

```
a = [ 1 0, 4 2, 3 4, 5 3, 6 3 ];
b = branks(a);
print "BRANKS: ", b;
```

```
BRANKS:
   |         1         2         3
-----------------------------------
```

```
 1 |   1.00000   1.00000   1.00000
 2 |   3.00000   2.00000   2.00000
 3 |   2.00000   5.00000   2.00000
 4 |   4.00000   3.50000   3.00000
 5 |   5.00000   3.50000   3.50000
```

3.2 Function `byte`

b = byte(a)

Purpose: The `byte` function transfers integers into ASCI characters.

Input: The input object `a` can be scalar, vector or matrix and should have real or int values in $[0, 255]$.

Output: The result`b` is of the same dimension as the input argument `a` and contains characters.

Restrictions: 1. Missing values, complex, and string data are not permitted.

2. The input object `a` should have real or int values in $[0, 255]$.

Relationships: atof(), atoi()

Examples: 1. Example from SAS/IML Manual:

```
a = [ 97:122 ];
b = byte(a);
print "Byte: ", b;
```

```
Byte:
   |   1   2   3   4   5   6   7   8   9  10  11  12  13
---------------------------------------------------------
 1 |   a   b   c   d   e   f   g   h   i   j   k   l   m

   |  14  15  16  17  18  19  20  21  22  23  24  25  26
---------------------------------------------------------
 1 |   n   o   p   q   r   s   t   u   v   w   x   y   z
```

3.3 Function `covlag`

b = covlag(x,k)

Purpose: The `covlag` function computes a sequence of lagged cross product matrices. The function is useful for computing autocovariance sequences for scalar and vector time series. Assuming that the input `a` is a $nr \times nv$ matrix, then the result `b` is a $nv \times k * nv$ matrix containing a sequence of lagged cross products. The jth $nv \times nv$ block is

$k < 0$

$$\frac{1}{nr} \sum_{j=i}^{n} x_j^T x_{j-i+1}$$

$k > 0$

$$\frac{1}{nr} \sum_{j=i}^{n} (x_j^T - \bar{x})(x_{j-i+1} - \bar{x})$$

Input:
1. The first argument `x` must be a $nr \times nv$ matrix of (nonmissing) time series values.
2. The second argument k must be nonzero positive or negative scalar but with $0 < \|k\| < nr$. If k is positive, a mean correction is made, if k is negative, no mean correction is made.

Output: The result `b` is a $nv \times k * nv$ matrix containing a sequence of lagged cross products.

Restrictions:
1. Missing values and string data are not permitted.
2. The first argument x must be numeric vector or matrix, the second argument k must be integer.

Relationships:

Examples: 1. Example from SAS/IML Manual:

```
print "Example from IML Manual";
/* res=  33.0000   23.1000   13.6000    4.9000 */
vec = [ -9:2:9 ]; print "Input=", vec;
cov = covlag(vec,4);
print "COVLAG(,4)=",cov;
```

```
Example from IML Manual

Input=
   |   1   2   3   4   5   6   7   8   9  10
-----------------------------------------
 1 |  -9  -7  -5  -3  -1   1   3   5   7   9
```

```
COVLAG(,4)=
   |         1         2         3         4
-------------------------------------------
 1 |   33.0000   23.1000   13.6000    4.9000
```

2. Simple example with matrix argument:

```
a2 = [ 1 2, 3 4, 5 6 ];
b2 = covlag(a2,2);
print "B2=", b2;
```

```
B2=
   |         1         2         3         4
-------------------------------------------
 1 |   2.66667   2.66667   0.00000   0.00000
 2 |   2.66667   2.66667   0.00000   0.00000
```

```
a2 = [ 1 2, 3 4, 5 6 ];
b3 = covlag(a2,-2);
print "B2=", b3;
```

```
B2=
   |         1         2         3         4
-------------------------------------------
 1 |   11.6667   14.6667    6.0000    8.6667
 2 |   14.6667   18.6667    7.3333   10.6667
```

3.4 Function gee

< gof,parm,ncov,rcov,yhat > = gee(data,model<,optn,<,class<,rstr<,xini>>>>)

Purpose: The gee function computes the generalized linear regression model for data with grouped observations, i.e. subjects in clusters or longitudinal data for K time points. There are n_k observations in each group. The data must contain an ID variable indicating the grouping. The data in each group are assumed to be correlated rather than independent as in the comon Generalized LInear Model. It is assumed that each group has the same correlation structure. The correlation structure can be either specified as a symmetric matrix input argument or specified as one of a number of different choices. The following correlation structures can be specified:

1. specified input (fixed, kept constant and is not fittet)
2. unspecified (independent): assumes equal group sizes $n_1 = \ldots = n_K$, all off diagonal entries are estimated
3. stationary m dependent: assumes equal group sizes $n_1 = \ldots = n_K$ and and $m < n_k$
4. nonstationary m dependent: assumes $m < n_k$
5. exchangeable: all off diagonal entries have same value α
6. AR m dependent: assumes $m < n_k$

The link and distribution functions are specified similar to those of the `glim()` and `glmixd()` functions. The following link functions $g(\mu)$ can be specified:

1. Identity: $g(\mu) = \mu$
2. Logarithm: $g(\mu) = \log(\mu)$
3. Logit: $g(\mu) = \frac{\log(\mu)}{(1-\mu)}$
4. Probit: $g(\mu) = \Phi^{-1}(\mu)$
5. Complementary Log-Log: $g(\mu) = \log(-\log(1-\mu))$
6. Reciprocal: $g(\mu) = 1./\mu$

The following variance distribution functions can be specified:

1. Normal (Gaussian): $V(\mu) = 1$
2. Poisson: $V(\mu) = \mu$
3. Binomial: $V(\mu) = \mu(1-\mu)$
4. Gamma: $V(\mu) = \mu^2$

Input: There are at most six input arguments.

data The first argument `data` must be a $N \times n$ matrix containing the data of K groups of observations. The data matrix must contain an ID variable (column) specifying to which group the observation belongs. In some cases of assumed

model : The analysis model is specified in form of a string, e.g. `model= "3=1 2"`, containing column numbers for variables. The syntax of the `model` string argument is the same as for the `glmod()` function except for the additional *events / trial* response specification. **????**

optn This argument must be specified in form of a two column matrix where the first column defines the option as string value (in quotes) and the second column can be used for a numeric or string specification of the option. See table below for content.

class specifies which of the columns of the input data matrix $\mathbf{X}$ are nominally scaled CLASS variables. (This argument can be missing value.)

rstr This input argument specifies a fixed input correlation matrix. The matrix must be symmetric and cannot contain missing entries. The row and column number (size) of this matrix must be the same as the maximum group size n_k of the data.

xini This input argument specifies a starting vector for the iteration of the parameter estimates β. The number of entries of β is defined by the `model` and `class`. If the size of β is not obvious, the printed output of preliminary run using the same `model` and `class` input will show the meaning of the entries of β.

Option	Second Column	Meaning
"alpha"	real	default=0.05
"cstr"	string	correlation structure
	"arm"	auto regressive (AR) m structure
	"given"	structure is specified (fixed)
	"ident"	structure is identity matrix (fixed)
	"mstat"	stationary with m dependence
	"nonst"	nonstationary with m dependence
	"excha"	exchageable
	"unspe"	unspecified (independent)
"dep"	int	dependence, default=1 (¿ 0)
"dist"	string	variance distribution
	"nor"	normal (Gaussian)
	"poi"	Poisson
	"bin"	Binomial
	"gam"	Gamma
"freq"	int	column number of frequency variable
"idvar"	int	column number of ID variable
"link"	string	link function
	"cll"	complementary Log-Log
	"ide"	identity
	"log"	logarithm
	"logit"	logit
	"probit"	probit
	"rec"	reciprocal
"maxit"	int	maximum number of iterations (def=400)
"nobstat"		suppress observationwise output
"noint"		do not use intercept in linear model
"offset"	int	column number of offset variable
"pcov"	int	print covariance (¿0) and correlation (¿1) matrices
"popt"	int	print optimization history
"print"	int	controls printed output
"pres"		printed observationwise output
"trial"	int	column number of trial variable
"wgt"	int	column number of weight variable
"xtol"	real	relative x termination criterion (def=1.e-3)
"ysc"	string	type of scale of response variable
	'i'	interval
	'n'	nominal
	'o'	ordinal

Output: There are at most five return arguments.

gof returns a vector of scalar results.

parm returns a matrix with np rows and either two or four columns where the rows correspnd to the parameters in β and the columns have the following content:

1. containing the naive parameter estimates β
2. containing the "robust" parameter estimates z
3. containing the naive asymptotic standard errors
4. containing the robust asymptotic standard errors

ncov contains the $np \times np$ naive asymptotic variance matrix of parameter estimates

rcov contains the $np \times np$ robust asymptotic variance matrix of parameter estimates

yhat matrix with N rows and columns containing:

1. the observed values $y_i, i = 1, \ldots, N$ from the data input
2. the values of the frequency variable in the data
3. the predicted values $\hat{y}_i, i = 1, \ldots, N$ for each observation
4. the residuals $r_i, i = 1, \ldots, N$

Restrictions:

1. Note, that the dimensions of the `rstr` and `xini` arguments must be compatible with the other specifications.
2. The data must not be sorted by the ID variable indicating the groups. But sorted data may reduce the execution time.

Relationships: glmixd(), glim()

Examples:

1. Simple example from the Pinheiro website:

```
print "Simple Example found at Pinheiro";
simp = [ 1   10   1  1 , 1   22   2  1 , 1   33   3  1 , 1   39   4  1 ,
         1   49   5  1 , 1   63   6  1 , 2    9   1  1 , 2   17   2  1 ,
         2   31   3  1 , 2   44   4  1 , 2   53   5  1 , 2   59   6  1 ,
         3   11   1  1 , 3   19   2  1 , 3   29   3  1 , 3   41   4  1 ,
         3   52   5  1 , 3   59   6  1 , 4   15   1  2 , 4   24   2  2 ,
         4   36   3  2 , 4   44   4  2 , 4   55   5  2 , 4   67   6  2 ,
         5   15   1  2 , 5   27   2  2 , 5   34   3  2 , 5   47   4  2 ,
         5   56   5  2 , 5   66   6  2 , 6   14   1  2 , 6   27   2  2 ,
         6   35   3  2 , 6   47   4  2 , 6   55   5  2 , 6   68   6  2 ];

cnam = [ "sub" "y" "x" "trt" ];
simp = cname(simp,cnam); /* print simp; */
```

```
print "Correlation Struct=Given";
modl = "2 = 3 4";
optn = [ "print"         4 ,
         "ysc"         'i' ,
         "cstr"    "given" ,
         "link"    "ident" ,
         "dist"   "normal" ,
         "idvar"         1 ];
 ride = ide(6);
 < gof,parm,ncov,rcov,yhat > = gee(simp,modl,optn,.,ride);
```

****
Model Information
****

Number Valid Observations	36
Response Variable	Y[2]
N Independend Variables	2
Error Distribution	NORMAL
Link Function	IDENTITY
Corr Structure	Specified
Interval Response: 2 Categorie	
Subject Variable Column	1
Significance Level:	0.0500000
Design Coding:	Full-Rank

****
Model Effects
****

Intercept + X3 + X4

****
Class Level Information
****

Class	Level	Value
I[1]	6	1 2 3 4 5 6

****
Simple Statistics

****

Column	Nobs	Mean	Std Dev	Skewness	Kurtosis
Y[2]	36	38.111111	18.037703	-0.0122782	-1.1875351
X[3]	36	3.5000000	1.7320508	0.0000000	-1.2770053
X[4]	36	1.5000000	0.5070926	0.0000000	-2.1212121

****
Parameter Information
****

Number	Coefficient	Effect
1	Beta_1	Fixed_Intercept
2	Beta_2	x
3	Beta_3	trt

Data are sorted in groups of subject ID variable.
Maximum Cluster Size 6
Minimum Cluster Size 6

Initial Estimates

	1
Beta_1	-5.488888889
Beta_2	10.26666667
Beta_3	5.111111111

Start of Optimization: Maximum Iterations=100 Xtol=0.001
Succesful Optimization after 1 Iterations

Working Correlations

1	1				
2	0	1			
3	0	0	1		
4	0	0	0	1	
5	0	0	0	0	1
6	0	0	0	0	0

6	1

Scale Parameter 2.36049
Mean Squared Error 2.36049

****
Naive and Robust Estimates
****

N	Parameter	N_Estimate	Stderr	R_Estimate	Stderr
1	Beta_1	-5.4889	0.9649	-9.8238	0.5587
2	Beta_2	10.2667	0.1499	119.5321	0.0859
3	Beta_3	5.1111	0.5121	17.8157	0.2869

Naive Variance Matrix
****

Dense Symmetric Matrix (3 by 3)

S	Beta_1	Beta_2	Beta_3
Beta_1	0.9310837		
Beta_2	-0.0786831	0.0224809	
Beta_3	-0.3934156	-6.94e-017	0.2622771

Naive Correlation Matrix

	Beta_1	Beta_2	Beta_3
Beta_1	1.000000000		
Beta_2	-0.543851678	1.000000000	
Beta_3	-0.796117339	-9.0366e-016	1.000000000

Robust Variance Matrix
****

Dense Symmetric Matrix (3 by 3)

S	Beta_1	Beta_2	Beta_3
Beta_1	0.3121811		
Beta_2	-0.0289947	0.0073772	
Beta_3	-0.1288066	0.0021164	0.0823045

Robust Correlation Matrix

```
                 Beta_1       Beta_2       Beta_3
Beta_1   1.000000000
Beta_2  -0.604185760  1.000000000
Beta_3  -0.803568531  0.085889750  1.000000000


              ******************************
              Observed and Predicted Response
              ******************************

            Nobs        Yobs        Yhat    Residual
              1  10.0000000  9.88888889  0.11111111
              2  22.0000000  20.1555556  1.84444444
              3  33.0000000  30.4222222  2.57777778
              4  39.0000000  40.6888889 -1.68888889
              5  49.0000000  50.9555556 -1.95555556
              6  63.0000000  61.2222222  1.77777778
              7  9.00000000  9.88888889 -0.88888889
              8  17.0000000  20.1555556 -3.15555556
              9  31.0000000  30.4222222  0.57777778
             10  44.0000000  40.6888889  3.31111111
             ....................................
             35  55.0000000  56.0666667 -1.06666667
             36  68.0000000  66.3333333  1.66666667


 print "EXCHA: GOF=",gof;
 print "parm=",parm;
 print "ncov=",ncov;
 print "rcov=",rcov;
 print "yhat=",yhat;


print "Correlation Struct=Exchange";
modl = "2 = 3 4";
optn = [ "print"         4 ,
         "xtol"      1.e-3 ,
         "ysc"         'i' ,
         "cstr"    "excha" ,
         "link"    "ident" ,
         "dist"   "normal" ,
         "idvar"         1 ];
 < gof,parm,ncov,rcov,yhat > = gee(simp,modl,optn);


Working Correlations
--------------------
```

```
1|          1
2|    -0.1372          1
3|    -0.1372    -0.1372          1
4|    -0.1372    -0.1372    -0.1372          1
5|    -0.1372    -0.1372    -0.1372    -0.1372          1
6|    -0.1372    -0.1372    -0.1372    -0.1372    -0.1372

6|          1
```

Scale Parameter 2.36049
Mean Squared Error 2.36049

Naive and Robust Estimates

N	Parameter	N_Estimate	Stderr	R_Estimate	Stderr
1	Beta_1	-5.4889	0.7204	-9.8238	0.5587
2	Beta_2	10.2667	0.1599	119.5321	0.0859
3	Beta_3	5.1111	0.2869	17.8157	0.2869

Naive Variance Matrix

Dense Symmetric Matrix (3 by 3)

S	Beta_1	Beta_2	Beta_3
Beta_1	0.5189465		
Beta_2	-0.0894815	0.0255661	
Beta_3	-0.1234568	-8.33e-017	0.0823045

Naive Correlation Matrix

	Beta_1	Beta_2	Beta_3
Beta_1	1.000000000		
Beta_2	-0.776853829	1.000000000	
Beta_3	-0.597367823	-1.8152e-015	1.000000000

Robust Variance Matrix

```
                Dense Symmetric Matrix (3 by 3)

                S |      Beta_1     Beta_2     Beta_3
                ------------------------------------
           Beta_1 |   0.3121811
           Beta_2 |  -0.0289947  0.0073772
           Beta_3 |  -0.1288066  0.0021164  0.0823045


                  Robust Correlation Matrix

                      Beta_1       Beta_2       Beta_3
         Beta_1   1.000000000
         Beta_2  -0.604185760  1.000000000
         Beta_3  -0.803568531  0.085889750  1.000000000


 print "Correlation Struct=Stationary m=1";
 modl = "2 = 3 4";
 optn = [ "print"         4 ,
          "xtol"      1.e-3 ,
          "ysc"         'i' ,
          "cstr"    "mstat" ,
          "link"    "ident" ,
          "dist"   "normal" ,
          "idvar"         1 ];
  < gof,parm,ncov,rcov,yhat > = gee(simp,modl,optn);

Working Correlations
--------------------

   1|          1
   2|    -0.1087          1
   3|          0    -0.1087          1
   4|          0          0    -0.1087          1
   5|          0          0          0    -0.1087          1
   6|          0          0          0          0    -0.1087

   6|          1


                  Scale Parameter 2.36063
                Mean Squared Error 2.36063


                ************************
                Naive and Robust Estimates
```

```
                    *************************

    N   Parameter  N_Estimate      Stderr  R_Estimate      Stderr
    1      Beta_1     -5.4525      0.8829     -9.4693      0.5758
    2      Beta_2     10.2669      0.1412    111.0222      0.0925
    3      Beta_3      5.0893      0.4627     17.6139      0.2889

 print "Correlation Struct=AR M=1";
 modl = "2 = 3 4";
 optn = [ "print"         4 ,
          "ysc"         'i' ,
          "cstr"      "arm" ,
          "link"    "ident" ,
          "dist"   "normal" ,
          "idvar"         1 ];
  < gof,parm,ncov,rcov,yhat > = gee(simp,modl,optn);

Working Correlations
--------------------

   1|          1
   2|    -0.1086          1
   3|     0.0118    -0.1086          1
   4|  -0.001282     0.0118    -0.1086          1
   5|  0.0001392  -0.001282     0.0118    -0.1086          1
   6| -1.512e-005  0.0001392  -0.001282     0.0118    -0.1086

   6|          1


                  Scale Parameter 2.3606
                 Mean Squared Error 2.3606


                 *************************
                 Naive and Robust Estimates
                 *************************

    N   Parameter  N_Estimate      Stderr  R_Estimate      Stderr
    1      Beta_1     -5.4570      0.8887     -9.4546      0.5772
    2      Beta_2     10.2667      0.1413    111.0429      0.0925
    3      Beta_3      5.0924      0.4669     17.6310      0.2888

 print "Correlation Struct=Nonstationary";
 modl = "2 = 3 4";
```

```
optn = [ "print"         3 ,
         "pcov"          2 ,
         "popt"          2 ,
         "maxit"        20 ,
         "xtol"      1.e-3 ,
         "ysc"         'i' ,
         "cstr"    "nonst" ,
         "link"    "ident" ,
         "dist"   "normal" ,
         "idvar"         1 ];
 < gof,parm,ncov,rcov,yhat > = gee(simp,modl,optn);
```

The iteration is not successful:

```
Start of Optimization: Maximum Iterations=20 Xtol=0.001

*****************************************************
GEE Iteration History (Corr Structure:  Nonstationary)
*****************************************************

  Iter       Sigma        Crit        Diff     L1_Crit
     1  14.1629630  0.35566360           .  1.95219796
     2  19.4199285  0.26703591 -0.08862768  1.28849899
     3  14.7377326  0.28987700  0.02284109  1.39871159
     4  19.5768881  0.29256768  0.00269068  1.41706390
     5  14.6648906  0.29650448  0.00393680  1.43613198
     6  19.7013344  0.29763133  0.00112685  1.44390274
     7  14.6463351  0.29860150  9.702e-004  1.44860934
     8  19.7333471  0.29889785  2.963e-004  1.45065993
     9  14.6418583  0.29913155  2.337e-004  1.45179415
    10  19.7412802  0.29920538  7.383e-005  1.45230546
    11  14.6407701  0.29926232  5.694e-005  1.45258185
    12  19.7432312  0.29928050  1.818e-005  1.45270778
    13  14.6405040  0.29929444  1.394e-005  1.45277545
    14  19.7437101  0.29929890  4.463e-006  1.45280637
    15  14.6404387  0.29930232  3.418e-006  1.45282296
    16  19.7438276  0.29930342  1.095e-006  1.45283055
    17  14.6404227  0.29930425  8.384e-007  1.45283462
    18  19.7438564  0.29930452  2.686e-007  1.45283648
    19  14.6404188  0.29930473  2.056e-007  1.45283748
    20  19.7438635  0.29930479  6.590e-008  1.45283794
```

2. RATGEE example from the Pinheiro website:

```
/*--- [1] Control ---*/
```

```
con0 = [ 151.00    199.00    246.00    283.00    320.00,
         145.00    199.00    249.00    293.00    354.00,
         147.00    214.00    263.00    312.00    328.00,
         155.00    200.00    237.00    272.00    297.00,
         135.00    188.00    230.00    280.00    323.00,
         159.00    210.00    252.00    298.00    331.00,
         141.00    189.00    231.00    275.00    305.00,
         159.00    201.00    248.00    297.00    338.00,
         177.00    236.00    285.00    340.00    376.00,
         134.00    182.00    220.00    260.00    296.00,
         160.00    208.00    261.00    313.00    352.00,
         143.00    188.00    220.00    273.00    314.00,
         154.00    200.00    244.00    289.00    325.00,
         171.00    221.00    270.00    326.00    358.00,
         163.00    216.00    242.00    281.00    312.00,
         160.00    207.00    248.00    288.00    324.00,
         142.00    187.00    234.00    280.00    316.00,
         156.00    203.00    243.00    283.00    317.00,
         157.00    212.00    259.00    307.00    336.00,
         152.00    203.00    246.00    286.00    321.00,
         154.00    205.00    253.00    298.00    334.00,
         139.00    190.00    225.00    267.00    302.00,
         146.00    191.00    229.00    272.00    302.00,
         157.00    211.00    250.00    285.00    323.00,
         132.00    185.00    237.00    286.00    331.00,
         160.00    207.00    257.00    303.00    345.00,
         169.00    216.00    261.00    295.00    333.00,
         157.00    205.00    248.00    289.00    316.00,
         137.00    180.00    219.00    258.00    291.00,
         153.00    200.00    244.00    286.00    324.00 ];

nr = nrow(con0);
rat = [ 1 : nr ]‘;
trt = cons(nr,1,0.);
con1 = trt -> rat -> con0[,1] -> cons(nr,1, 8.);
con2 = trt -> rat -> con0[,2] -> cons(nr,1,15.);
con3 = trt -> rat -> con0[,3] -> cons(nr,1,22.);
con4 = trt -> rat -> con0[,4] -> cons(nr,1,29.);
con5 = trt -> rat -> con0[,5] -> cons(nr,1,36.);
cont = con1 |> con2 |> con3 |> con4 |> con5;
cnam = [ "trt" "rat" "wgt" "time" ];
con1 = cname(con1,cnam); print con1;

/*--- [2] Treatment ---*/
trt0 = [ 114.00    151.00    188.00    214.00    253.00,
```

```
          140.00    189.00    229.00    258.00    303.00,
          133.00    176.00    220.00    252.00    282.00,
          132.00    168.00    208.00    234.00    270.00,
          119.00    155.00    186.00    207.00    233.00,
          155.00    190.00    220.00    243.00    262.00,
          177.00    146.00    188.00    214.00    245.00,
          129.00    160.00    199.00    228.00    261.00,
          148.00    192.00    227.00    247.00    278.00,
          144.00    183.00    220.00    245.00    278.00,
          117.00    145.00    177.00    210.00    245.00,
          156.00    208.00    242.00    278.00    319.00,
          108.00    137.00    164.00    189.00    221.00,
          140.00    178.00    210.00    235.00    265.00,
          139.00    180.00    221.00    256.00    289.00,
          129.00    170.00    209.00    251.00    289.00,
          122.00    154.00    191.00    226.00    254.00,
          134.00    178.00    212.00    247.00    276.00,
          131.00    169.00    206.00    242.00    278.00,
          119.00    155.00    190.00    225.00    264.00,
          140.00    178.00    220.00    249.00    285.00,
          136.00    167.00    204.00    232.00    271.00,
          148.00    184.00    225.00    250.00    284.00,
          145.00    178.00    219.00    252.00    290.00,
          146.00    175.00    212.00    233.00    266.00,
          118.00    154.00    178.00    203.00    226.00,
          147.00    190.00    221.00    259.00    286.00,
          143.00    182.00    210.00    240.00    264.00,
          148.00    197.00    235.00    279.00    309.00,
          136.00    177.00    223.00    256.00    287.00 ];

nr = nrow(trt0);
rat = 100. + [ 1 : nr ]‘;
trt = cons(nr,1,1.);
trt1 = trt -> rat -> trt0[,1] -> cons(nr,1, 8.);
trt2 = trt -> rat -> trt0[,2] -> cons(nr,1,15.);
trt3 = trt -> rat -> trt0[,3] -> cons(nr,1,22.);
trt4 = trt -> rat -> trt0[,4] -> cons(nr,1,29.);
trt5 = trt -> rat -> trt0[,5] -> cons(nr,1,36.);
trea = trt1 |> trt2 |> trt3 |> trt4 |> trt5;
cnam = [ "trt" "rat" "wgt" "time" ];
trt1 = cname(trt1,cnam); print trt1;


ful0 = cont |> trea;
tim2 = (ful0[,4] - 22.) .** 2;
ttrt = ful0[,1] .* ful0[,4];
```

```
t2rt = ful0[,1] .* tim2;
full = ful0 -> tim2 -> ttrt -> t2rt;
cnam = [ "trt" "rat" "wgt" "time" "tim2" "timtrt" "tim2trt" ];
full = cname(full,cnam);  /* print "FULL=", full; */


print "LIANG-ZEGER MODELS: Model1: Exchangeable";
modl = "3 = 4 5";
optn = [ "print"         4 ,
         "cstr"    "excha" ,
         "link"    "ident" ,
         "dist"   "normal" ,
         "idvar"         2 ];
< gof,parm,ncov,rcov,yhat > = gee(full,modl,optn);
```

```
Working Correlations
--------------------

   1|          1
   2|     0.8224          1
   3|     0.8224     0.8224          1
   4|     0.8224     0.8224     0.8224          1
   5|     0.8224     0.8224     0.8224     0.8224          1

                  Scale Parameter 690.169
                Mean Squared Error 690.169

                 *************************
                 Naive and Robust Estimates
                 *************************

    N   Parameter  N_Estimate      Stderr  R_Estimate      Stderr
    1      Beta_1    106.0610      3.5313     54.4200      1.9489
    2      Beta_2      5.4876      0.0646     46.1127      0.1190
    3      Beta_3     -0.0289      0.0078     -6.2577      0.0046
```

```
print "GOF=",gof;
print "parm=",parm;
print "ncov=",ncov;
print "rcov=",rcov;
print "yhat=",yhat;


print "Model 2: LIANG-ZEGER MODELS: Model2: AR-M, mdep=1";
```

```
modl = "3 = 4 5";
optn = [ "print"         4 ,
         "nobstat"         ,
         "xtol"      1.e-3 ,
         "ysc"         'i' ,
         "cstr"      "arm" ,
         "dep"           1 ,
         "link"    "ident" ,
         "dist"   "normal" ,
         "idvar"         2 ];
< gof,parm,ncov,rcov,yhat > = gee(full,modl,optn);

                  Scale Parameter 690.17
                 Mean Squared Error 690.17

                 *************************
                 Naive and Robust Estimates
                 *************************

  N   Parameter  N_Estimate      Stderr  R_Estimate      Stderr
  1      Beta_1    106.0699      3.7853     53.0473      1.9995
  2      Beta_2      5.4875      0.0801     46.0506      0.1192
  3      Beta_3     -0.0289      0.0054     -5.8582      0.0049


print "Model 3: LIANG-ZEGER MODELS: Model3: Exchangeable";
modl = "3 = 4 5 1 6 7";
optn = [ "print"         4 ,
         "nobstat"         ,
         "xtol"      1.e-3 ,
         "ysc"         'i' ,
         "cstr"    "excha" ,
         "link"    "ident" ,
         "dist"   "normal" ,
         "idvar"         2 ];
< gof,parm,ncov,rcov,yhat > = gee(full,modl,optn);

                  Scale Parameter 295.353
                 Mean Squared Error 295.353

                 *************************
                 Naive and Robust Estimates
                 *************************
```

```
  N   Parameter  N_Estimate      Stderr  R_Estimate      Stderr
  1      Beta_1    110.4724      3.2917     43.6452      2.5311
  2      Beta_2      6.1810      0.0653     60.4876      0.1022
  3      Beta_3     -0.0395      0.0079     -7.5588      0.0052
  4      Beta_4     -8.8229      4.6551     -2.3668      3.7277
  5      Beta_5     -1.3867      0.0924     -8.8409      0.1568
  6      Beta_6      0.0212      0.0112      2.4047      0.0088
```

```
optn = [ "print"         4 ,
         "nobstat"         ,
         "xtol"      1.e-3 ,
         "ysc"         'i' ,
         "cstr"      "arm" ,
         "dep"           1 ,
         "link"    "ident" ,
         "dist"   "normal" ,
         "idvar"         2 ];
< gof,parm,ncov,rcov,yhat > = gee(full,modl,optn);

                 Scale Parameter 295.411
                Mean Squared Error 295.411

                 *************************
                 Naive and Robust Estimates
                 *************************

  N   Parameter  N_Estimate      Stderr  R_Estimate      Stderr
  1      Beta_1    111.0383      3.6951     42.9344      2.5862
  2      Beta_2      6.1658      0.0934     62.5770      0.0985
  3      Beta_3     -0.0419      0.0066     -7.7637      0.0054
  4      Beta_4     -9.9370      5.2257     -2.6248      3.7857
  5      Beta_5     -1.3567      0.1321     -8.3956      0.1616
  6      Beta_6      0.0258      0.0093      2.7818      0.0093
```

3.5 Function `modtst`

xmom = modtst(x,model<,class<,optn>>)

Purpose: The function `modtst` is used to test the correctness of the parsing of a model string argument.

Input: **x** specifies a $N \times n$ data matrix containing the N observations of n numerical and/or nominal scaled CLASS variables in its n columns.

model specifies a predictive (with response) or exploratory (without response) model in form of a string or a string vector which may contain the operators :, ∗, |, and @. (Note, a string stretching across line breaks must be separated by a \.)

class specifies which of the columns of the input data matrix **X** are nominally scaled CLASS variables. (This argument can be missing value.)

optn This argument must be specified in form of a two column matrix where the first column defines the option as string value (in quotes) and the second column can be used for a numeric or string specification of the option. See table below for content.

Output: The only return argument contains the means, standard deviations, minima, and maxima of the predictor and response variables.

Restrictions:

1. The input argument `x` must be a numeric matrix.
2. Missing values in `x` are not permitted.

Relationships:

Examples: The following two examples are by Fan and Li (2002), see also Benner (2006), and are also used in an application of the `scad` function. The first example shows a tedious way to generate polynomial effects from the original variables. The second example shows that the use of the | and the @ operator in the model argument can much easier arrive at the same results.

Example with explicitly formulated interactions:

1.
```
options NOECHO;
 %inc "C:\\tdata2\\ulrich\\scadch12.dat";
options ECHO;

 vnam = [ "Yresp" "a" "treatment" "gender" "married" "what" "status" ];
 dlta = 1 - scadch12[,7];
 c6  = scadch12[,6];
 h1 = (c6 .== 2); h2 = (c6 .== 4);
 h3 = (c6 .== 5); hh = h1 -> h2 -> h3; /* print hh; */
 c2  = scadch12[,2]; c3  = scadch12[,3];
 c4  = scadch12[,4]; c5  = scadch12[,5];
 c6  = scadch12[,6];
 c32  = c3 .* c2; c34 = c3 .* c4; c35 = c3 .* c5;
 c24  = c2 .* c4; c25 = c2 .* c5; c45 = c4 .* c5;
 iact = c32 -> c34 -> c35 -> c24 -> c25 -> c45;
 scad1 = scadch12[,1:5] -> hh -> iact -> dlta;
 vnam1 = [ "Yresp" "a" "treatment" "gender" "married"
           "h1" "h2" "h3" "c32" "c34" "c35"
           "c24" "c25"  "c45" "status" ];
```

```
scad1 = cname(scad1,vnam1);
model = "1 = 2:5 6:8 9:14";
class = [ 6:8 15 ];
xmom = modtst(scad1,model,class);
print "Xmoments=",xmom;
```

```
                    *****************
                    Model Information
                    *****************

                Number Valid Observations 1601
                Response Variable          Y[1]
                N Independend Variables     13

                Linear Model Without Intercept
                Design Coding:       Full-Rank

                     *************
                     Model Effects
                     *************

X2  + X3  + X4  + X5  + C6  + C7  + C8  + X9  + X10 + X11 + X12 +
                          X13 + X14

                   ***********************
                   Class Level Information
                   ***********************

Class  Level    Value

C[ 6]      2        0       1
C[ 7]      2        0       1
C[ 8]      2        0       1

                     *****************
                     Simple Statistics
                     *****************

Column   Nobs        Mean     Std Dev    Skewness    Kurtosis
Y[ 1]    1601   241.23111   275.27033   1.2579456   0.5915896
X[ 2]    1601   82.645222   7.7069160  -0.2190298  -0.4944203
X[ 3]    1601   0.4447220   0.4970902   0.2226843  -1.9528528
```

X[4]	1601	0.2642099	0.4410493	1.0705607	-0.8549693
X[5]	1601	0.1717676	0.3772962	1.7420949	1.0361874
X[9]	1601	36.765147	41.387209	0.2643487	-1.8795305
X[10]	1601	0.1093067	0.3121211	2.5066039	4.2884189
X[11]	1601	0.0712055	0.2572481	3.3378731	9.1528291
X[12]	1601	21.333542	35.837462	1.1129424	-0.6992959
X[13]	1601	13.652717	30.123733	1.7776706	1.2244007
X[14]	1601	0.1011868	0.3016701	2.6473428	5.0146870

****
Number of Observations for Class Levels
****

Variable	Value	Nobs	Proportion
C[6]	0	1258	78.575890
	1	343	21.424110
C[7]	0	1088	67.957527
	1	513	32.042473
C[8]	0	1432	89.444097
	1	169	10.555903

Xmoments=

	Eff_01	Eff_02	Eff_03	Eff_04	Eff_05
Mean	82.645	0.44472	0.26421	0.17177	0.21424
StdDev	7.7045	0.49693	0.44091	0.37718	0.41029
Minimum	65.000	0.00000	0.00000	0.00000	0.00000
Maximum	104.00	1.00000	1.00000	1.00000	1.00000

	Eff_06	Eff_07	Eff_08	Eff_09	Eff_10
Mean	0.32042	0.10556	36.765	0.10931	0.07
StdDev	0.46664	0.30727	41.374	0.31202	0.25717
Minimum	0.00000	0.00000	0.00000	0.00000	0.00000
Maximum	1.00000	1.00000	99.000	1.00000	1.00000

	Eff_11	Eff_12	Eff_13	Y_01
Mean	21.334	13.653	0.10119	241.23
StdDev	35.826	30.114	0.30158	275.18
Minimum	0.00000	0.00000	0.00000	0.00000

```
Maximum |   100.000    99.000   1.00000    1092.0
```

2. Example using | and @ operators:

```
scad2 = scadch12[,1:5] -> hh -> dlta;
vnam2 = [ "Yresp" "a" "treatment" "gender" "married"
          "h1" "h2" "h3" "status" ];
scad2 = cname(scad2,vnam2);
model = "1 = 2|3|4|5@2 6:8";
class = [ 6:9 ];
xmom = modtst(scad2,model,class);
print "Xmoments=",xmom;
```

```
                    *****************
                    Model Information
                    *****************

                Number Valid Observations 1601
                Response Variable          Y[1]
                N Independend Variables      7

                Linear Model Without Intercept
                Design Coding:       Full-Rank


                      *************
                      Model Effects
                      *************

X2 + X3 + X2 * X3 + X4 + X2 * X4 + X3 * X4 + X5 + X2 * X5 +
                X3 * X5 + X4 * X5 + C6 + C7 + C8


                   ***********************
                   Class Level Information
                   ***********************

  Class  Level    Value

   C[6]      2        0       1
   C[7]      2        0       1
   C[8]      2        0       1


                     *****************
                     Simple Statistics
```

****

Column	Nobs	Mean	Std Dev	Skewness	Kurtosis
Y[1]	1601	241.23111	275.27033	1.2579456	0.5915896
X[2]	1601	82.645222	7.7069160	-0.2190298	-0.4944203
X[3]	1601	0.4447220	0.4970902	0.2226843	-1.9528528
X[4]	1601	0.2642099	0.4410493	1.0705607	-0.8549693
X[5]	1601	0.1717676	0.3772962	1.7420949	1.0361874

****
Number of Observations for Class Levels
****

Variable	Value	Nobs	Proportion
C[6]	0	1258	78.575890
	1	343	21.424110
C[7]	0	1088	67.957527
	1	513	32.042473
C[8]	0	1432	89.444097
	1	169	10.555903

Xmoments=

	Eff_01	Eff_02	Eff_03	Eff_04	Eff_05
Mean	82.645	0.44472	36.765	0.26421	21.334
StdDev	7.7045	0.49693	41.374	0.44091	35.826
Minimum	65.000	0.00000	0.00000	0.00000	0.00000
Maximum	104.00	1.00000	99.000	1.00000	100.000

	Eff_06	Eff_07	Eff_08	Eff_09	Eff_10
Mean	0.10931	0.17177	13.653	0.07	0.10119
StdDev	0.31202	0.37718	30.114	0.25717	0.30158
Minimum	0.00000	0.00000	0.00000	0.00000	0.00000
Maximum	1.00000	1.00000	99.000	1.00000	1.00000

	Eff_11	Eff_12	Eff_13	Y_01
Mean	0.21424	0.32042	0.10556	241.23
StdDev	0.41029	0.46664	0.30727	275.18
Minimum	0.00000	0.00000	0.00000	0.00000

```
Maximum |   1.00000   1.00000   1.00000    1092.0
```

3.6 Function ranktie

b = ranktie(a)

Purpose: The `ranktie` function requires a numeric input vector `b` and creates a vector of the same dimension with the ranks of the entries of `a` and tied ranks are averaged.

Input: The only argument `a` must be a numeric N vector without missing values.

Output: The only return argument is the vector of (tied) ranks of the input values.

Restrictions: 1. Missing values and string data are not permitted.

2. The input argument `a` must be a numeric vector.

Relationships: operator <:, branks()

Examples: 1. SAS/IML Manual, page 397:

```
a = [ 2 2 1 0 5 ];
b = ranktie(a);
print "Ranktie: ", b;
```

```
Ranktie:
   |         1         2         3         4         5
---------------------------------------------------------
 1 |   3.50000   3.50000   2.00000   1.00000   5.00000
```

3.7 Function roccomp

< gof,area,diff,cova,covd > = roccomp(yvec,xmat<,contr<,optn>>)

Purpose: The `roccomp` function implements the method by DeLong et al (1988) for the comparison of the area of the ROC curve. The Man-Whitney test with contrasts specifying the parametrization of different models is used for computing a χ^2 value and corresponding p value.

Input: **yvec** specifies N response vector $\mathbf{Y}$ of a logistic model; note, `yvec` must contain only binary $(0, 1)$ and maybe missing values.

xmat specifies $N \times nv$ data set $\mathbf{X}$ of a set of predictors in a variety of models

cont specifies $nc \times nv$ matrix of contrasts is optional

optn options vector with the following entries:

1. int specifying the amount of printed output (def=2)
2. real specifying probability α (def=.05)

Output: **gof** Some scalar results like joint χ^2 value with p values and degrees of freedom.

area The area under the ROC curve with asymptotic standard error and lower and upper confidence limits.

diff The estimates of test differences w.r.t. to the specified contrasts with asymptotic standard errors and confidence limits.

cova The asymptotic variance-covariance matrix for the areas under the ROC curve.

covd The asymptotic variance-covariance matrix for the test differences.

Restrictions: 1. The input data may contain missing data, but should not contain string or complex data.

2. The number of rows of input arguments 1 and 2 and the number of columns of input arguments 2 and 3 must be the same.

Relationships: glim()

Examples: 1. Example by DeLong et al. (1988):

```
iroc = [ 3.0 5.8 10 0,  3.2 6.3  5 1,  3.9 6.8  3 1,  2.8 4.8  6 0,
         3.2 5.8  3 1,  0.9 4.0  5 0,  2.5 5.7  8 0,  1.6 5.6  5 1,
         3.8 5.7  5 1,  3.7 6.7  6 1,   .   .   6 1,  3.2 5.4  4 1,
         3.8 6.6  6 1,  4.1 6.6  5 1,  3.6 5.7  5 1,  4.3 7.0  4 1,
         3.6 6.7  4 0,  2.3 4.4  6 1,  4.2 7.6  4 0,  4.0 6.6  6 0,
         3.5 5.8  6 1,  3.8 6.8  7 1,  3.0 4.7  8 0,  4.5 7.4  5 1,
         3.7 7.4  5 1,  3.1 6.6  6 1,  4.1 8.2  6 1,  4.3 7.0  5 1,
         4.3 6.5  4 1,  3.2 5.1  5 1,  2.6 4.7  6 1,  3.3 6.8  6 0,
         1.7 4.0  7 0,   .   .   6 1,  3.7 6.1  5 1,  3.3 6.3  7 1,
         4.2 7.7  6 1,  3.5 6.2  5 1,  2.9 5.7  9 0,  2.1 4.8  7 1,
          .   .   8 1,  2.8 6.2  8 0,   .   .   7 1,   .   .   7 1,
         4.0 7.0  7 1,  3.3 5.7  6 1,  3.7 6.9  5 1,  2.0  .   7 1,
         3.6 6.6  5 1];

iroc[,3] = 10 - iroc[,3];
cnam = [ "alb" "tp" "totscore" "popind" ];
iroc = cname(iroc,cnam);
```

```
yvec = iroc[,4]; ind = [ 3 1 2 ];
xmat = iroc[ ,ind ];
vnam = cnam[ ind ];
xmat = cname(xmat,vnam);
contr = [ 1 -1  0 ,
          1  0 -1 ];
optn = [   3 ,    /* ipri */
         .05 ];   /* alfa */
< gof,area,diff,cova,covd > = roccomp(yvec,xmat,contr,optn);
```

Contrast Test: ChiSquare=1.63143 df=2 pval= 0.4423

Pairwise Deletion Man-Whitney Coefficients

	totscore	alb	tp
totscore	0.686936937	0.717447917	0.725806452
alb	0.718750000	0.718750000	0.736559140
tp	0.647849462	0.647849462	0.647849462

ROC Curve Areas and 0.05 Confidence Intervals

****

Dense Matrix (3 by 4)

	Estimate	AStdErr	LowCL	UppCL
totscore	0.6869369	0.1051074	0.4809302	0.8929437
alb	0.7187500	0.0928336	0.5367996	0.9007004
tp	0.6478495	0.1000474	0.4517602	0.8439387

Estimated Variance-Covariance Matrix

****

Dense Symmetric Matrix (3 by 3)

S	totscore	alb	tp
totscore	0.0110476		
alb	0.0050274	0.0086181	
tp	0.0048193	0.0077445	0.0100095

Estimated Correlation Matrix

```
                totscore          alb           tp
totscore   1.000000000  0.515234232  0.458294492
alb        0.515234232  1.000000000  0.833843124
tp         0.458294492  0.833843124  1.000000000


                Contrast Coefficients
                **********************

                  Dense Matrix contr

              |    totscore        alb         tp
            ----------------------------------------
        Con_1 |   1.0000000 -1.0000000  0.0000000
        Con_2 |   1.0000000  0.0000000 -1.0000000


    Tests and 0.05 Confidence Intervals for Contrast Rows
    *****************************************************

                  Dense Matrix (2 by 6)

      |    Estimate    AStdErr      LowCL      UppCL     ChiSqu
    ------------------------------------------------------------
Con_1 |  -0.0318131  0.0980349 -0.2239580  0.1603319  0.1053051
Con_2 |   0.0390875  0.1068572 -0.1703487  0.2485237  0.1338037

      |      Pvalue
    ---------------
Con_1 |   0.7455539
Con_2 |   0.7145204


               Variance Estimates of Contrast
               ******************************

              Dense Symmetric Matrix (2 by 2)

                  S |       Con_1      Con_2
                  --------------------------
               Con_1 |   0.0096108
               Con_2 |   0.0089454  0.0114185


print "GOF=",  gof;
print "Area=", area;
print "CTest=",diff;
```

```
  print "COVA=", cova;
  print "COVD=", covd;
```

GOF=

	1
Failure	0.00000
ChiSqu	1.63143
Pvalue	0.44232
DF	2.00000
unused	.
unused	.
unused	.
unused	.
unused	.
unused	.

Area=

	Estimate	AStdErr	LowCL	UppCL
totscore	0.68694	0.10511	0.48093	0.89294
alb	0.71875	0.09283	0.53680	0.90070
tp	0.64785	0.10005	0.45176	0.84394

CTest=

	Estimate	AStdErr	LowCL
Con_1	-0.03181	0.09803	-0.22396
Con_2	0.03909	0.10686	-0.17035

	UppCL	ChiSqu	Pvalue
Con_1	0.16033	0.10531	0.74555
Con_2	0.24852	0.13380	0.71452

COVA=

SYM	totscore	alb	tp
totscore	0.01105		
alb	0.00503	0.00862	
tp	0.00482	0.00774	0.01001

COVD=

SYM	Con_1	Con_2
Con_1	0.00961	
Con_2	0.00895	0.01142

3.8 Function `scad`

gof = scad(meth,xtrn,lamb,frac,modl,optn<,class<,xtst>>)

<gof,parm,gcvtab,cov,restrn,reststst> = scad(meth,xtrn,lamb,frac,model,optn,...)

Purpose: The `scad` function stands for "smoothly clipped absolute deviations" and implements penalized sparse estimation for the following three parameter estimation methods

"lse" linear least squares regression

"phr" proportional hazards regression

"svm" support vector machines regression

In each of the three modeling methods the objective function of the optimization problem is extended by a penalty term which enforces sparsity in the vector of the parameter estimates θ. See Fan and Li (2001, 2002) for more details. Note, the method is sensitive toward local optima. Also note, that the original SCAD method is useful only for small and medium sized number n of variables since it requires to solve a linear system $\mathbf{Q}x = p$, where the symmetric matrix $\mathbf{Q} \in \mathcal{R}^{n \times n}$ is large and dense for a large number n of variables. For the case $n \gg N$ where the number n of variables is much larger than the number N of observations a modification of SCAD is used (Benner, 2006; Hastie & Tibshirani, 2002) which is using a preliminary singular value decomposition for analyzing a $n \times n$ matrix $\mathbf{V}$. This approach, however, reduces the content of original data information by introducing a specific linear scalar product model.

Input: **meth** specifies the estimation method and must be one of the following three string values:

"lse" linear least squares regression

"phr" proportional hazards regression

"svm" support vector machines classification (for binary binary response/target)

xtrn must be a $N \times n$ training data set

lamb must be a numeric scalar or vector specifying the parameter λ of the penalty term

frac must be a numeric scalar or vector specifying an error fraction δ for cross classification table. This argument is used in two different ways:

- relative fraction: option `cdelta` is not specified: a sensitivity measure of parameter estimates is computed in each (outer) iteration, e.g. the asymptotic standard error (from the diagonal entries of the asymptotic covariance matrix); the lower bound

of nonzero estimates is defined by the product of the specified fraction value with the vector of threshold values; this approach is recommended for linear LS regression and Cox PH regression.

- absolute bound: option `cdelta` is specified: estimates smaller in magnitude than the value in `frac` are set to zero; this approach is recommended for linear SVM classification.

model : The analysis model is specified in form of a string, e.g. `model= "3=1 2"`, containing column numbers for variables. The syntax of the `model` string argument is the same as for the `glmod()` function except for the additional *events / trial* response specification. **????**

optn This argument must be specified in form of a two column matrix where the first column defines the option as string value (in quotes) and the second column can be used for a numeric or string specification of the option. See table below for content.

class specifies which of the columns of the input data matrix **X** are nominally scaled CLASS variables. For this function. at least the response variable is nominal and must be specified as CLASS variable.

xtst specifies a $Nt \times n$ validation data set used for cross classification

Content of the Options Matrix :

Option	Second Column	Meaning
`"aparm"`	real	value of SCAD a parameter, default: $a = 3.7$
`"alpha"`	real	significance level for confidence intervals, default is $\alpha = .05$
"block"	int	performs k-fold block cross validation
"cdelta"		the input values `frac` are used as absolute constants (and not relative to sensitivity) for comparing estimates to be zeroed
"cen"	int	column number for censor variable
"cloo"		Loo (leave-one-out) validation
`"cval"`	string	type of cross validation (needs `fold`)
	"loo"	leave-one-out cross validation, requires `fold=1` or `fold=N`;
	"blo"	block cross validation (biased); typical `fold` values are 4 or 10;
	"spl"	split cross validation; typical `fold` values are 4 or 10;
	"ran"	random cross validation (specify `seed`); typical `fold` values are 4 or 10;
	"set"	test set validation (`xtst` must be specified)
`"fold"`	int	fold number for cross validation
`"gtol"`	real	relative gradient termination for optimization
`"initer"`	int	maximum number iterations for starting solution; default is 10
`"maxi"`	int	maximum number iterations for final solution; default is 400
"nobstat"		do not print observationwise statistics
"nomis"		skip observations with missing values
"nopr"		suppress all printed output
"nosmw"		use Hastie - Tibshirani version for $Nobs > nvar$ the Sherman, Morrison, Woodbury formula is default
"perzer"		percentage of zeroed coefficients in each outer iteration of first stage for $Nobs > nvar$
"plamb"		
"pthet"		
"pmat"		
"ppar"		
"ppred"		
"popt"	int	> 0: print optimization history
"pinit"	int	> 0: print input information
"phist"	int	> 0: print history for all pairs (λ, δ)
"ptab"	int	> 0: print cross validation table for all pairs (λ, δ)
"print"	int	> 0: prints more output for increasing values
"pall"		all printed output, corresponds to `print=10`
"psum"		only summary output, corresponds to `print=1`
"psho"		only short output, corresponds to `print=2`
"rand"	int	performs k-fold random cross validation
`"ridg"`	real	≥ 0 ridge value for ridged regression; useful only for rankdeficient $X^T X$ matrix, e.g. $n > N$
"split"	int	performs k-fold split cross validation
`"seed"`	int	SEED for random generator; default is time of day
`"sing"`	real	singularity threshold; default is $1.e-8$
`"techsc"`	string	type of SCAD algorithm
	"ori"	original algorithm: tries to zero small values dur-

Output: **gof** vector of scalar results

1. indicates sucess (=0) or failure (¿0)
2. computation time in seconds
3. best crossvalidation value
4. best function (likelihood) value
5. best λ value
6. best δ value
7. best ridge value
8. best number selected variables
9. shrink value w.r.t. best estimates

parm returns n *times*4 (usually sparse) matrix with parameter estimates in first column, corresponding asymptotic standard errors in column2, and lower and upper confidence in columns 3 and 4.

gcvtab returns the results of the cross validation for all (λ, δ) pairs: the first n_λ rows contain the GC values, the next the likelihood values, and the last the numbers of selected variables; the columns correspond to the δ values;

cov the $n \times n$ asymptotic covariance matrix of the parameter estimates

restrn not used at the time (planned for residuals of traing data)

reststst not used at the time (planned for residuals of test/validation data)

Restrictions:

1. Currently, the `xtrn` data matrix may not contain missing values.
2. For the PHReg method the column of a binary censor variable must be specified.

Relationships: lars(), lrforw(), garotte(), svm(), smsvm(), varsel()

Examples: 1. SCAD for linear least squares regression: Original Algorithm Data: Middle-Aged Men in Health Fitness Club: SAS/STAT: Linnerud, NC State University:

```
fit = [ 191  36  50   5  162   60,   189  37  52   2  110   60,
        193  38  58  12  101  101,   162  35  62  12  105   37,
        189  35  46  13  155   58,   182  36  56   4  101   42,
        211  38  56   8  101   38,   167  34  60   6  125   40,
        176  31  74  15  200   40,   154  33  56  17  251  250,
        169  34  50  17  120   38,   166  33  52  13  210  115,
        154  34  64  14  215  105,   247  46  50   1   50   50,
        193  36  46   6   70   31,   202  37  62  12  210  120,
        176  37  54   4   60   25,   157  32  52  11  230   80,
        156  33  54  15  225   73,   138  33  68   2  110   43 ];
```

```
 cnam = [ "weight" "waist" "pulse" "chins" "situps" "jumps" ];
 fit = cname(fit,cnam);


 d = 4. / 9.; loglam = [ -1. : d : 3. ];
 lamb = 10 ** loglam; print "Lambda", lamb;
 frac = [.5 : .25 : 2. ]; print "Error Fractions=", frac;
 model = "4 = 1:3";


optn = [ "aparm"     3.7 ,
         "xtol"    1.e-4 ,
         "ridg"       0. ,
         "techsc"  "ori" ,
         "pinit"       2 ,
         "ptab"        2 ,
         "phist"       2 ,
         "print"       3 ];
< gof,parm,gcvtab,cov,yprd > = scad("lse",fit,lamb,frac,model,optn);

                      *****************
                      Model Information
                      *****************

                Number Valid Observations   20
                Response Variable          Y[4]
                N Independend Variables      3
                Original LSREG SCAD (Benner)
                Ridge Lambda         0.0000000
                Use Unscaled Predictor
                Linear Model Without Intercept


                       *************
                       Model Effects
                       *************

                        X1 + X2 + X3


                      *****************
                      Simple Statistics
                      *****************

 Column  Nobs        Mean     Std Dev    Skewness    Kurtosis
 Y[4]      20   9.4500000   5.2862782  -0.1930288  -1.4135210
```

```
   X[1]      20   178.60000   24.690505   0.9698740   1.8023463
   X[2]      20   35.400000   3.2019731   1.8721345   5.6620990
   X[3]      20   56.100000   7.2103726   0.8460998   0.6069130

Input Error Fraction Values
---------------------------

  1 :         0.5       0.75          1       1.25        1.5
  6 :        1.75          2

Input SCAD Lambda Values
------------------------

  1 :         0.1     0.2783     0.7743      2.154      5.995
  6 :       16.68      46.42      129.2      359.4       1000
       Initial Values Computed with 2 Initial Iterations


        Initial Parameter Estimates and Standard Errors
        ***********************************************

                     N    Estimate    AsStdErr
                     1  0.03378733  0.01865903
                     2 -0.34690359  0.11454985
                     3  0.27698303  0.02438803


                  Cross Validation Grid Search
                  ****************************

   I   ErrFract   J     Lambda  Nsel  Nitr        Plik       GCVal
       Iter   Nsel  Dwn        Crit       Xdiff        Gmax
          1      3    0  251.061803  0.00735490  2.00000000
          2      3    0  251.080258  8.912e-004  0.06691847
          3      3    0  251.082706  1.106e-004  0.00849913
          4      3    0  251.083013  1.376e-005  0.00105987
   1  0.5000000   1  0.1000000     3     4  250.746698  17.2894487


       Iter   Nsel  Dwn        Crit       Xdiff        Gmax
          1      3    0  254.227666  0.07031560  5.56511880
          2      3    0  254.084400  0.01786045  1.71901509
          3      3    0  254.063859  0.00447770  0.66258200
          4      3    0  254.062520  0.00121047  0.19063699
          5      3    0  254.062421  3.339e-004  0.05349581
          6      3    0  254.062414  9.260e-005  0.01490876
```

```
          2  0.2782559     3     6  251.084770  16.7918974

Iter   Nsel  Dwn        Crit       Xdiff        Gmax
   1      3    0  260.928294  0.15100674  15.4852737
   2      3    0  258.756980  0.04526859  10.0992913
   3      2    1  258.451022  0.01970498  9.12852290
   4      2    0  259.422181  0.01151980  361.804036
   5      2    0  258.330698  0.00153023  1.36253239
   6      2    0  258.329431  2.538e-004  0.19845288
   7      2    0  258.329396  4.260e-005  0.03334405
          3  0.7742637     2     7  252.790025  15.3153568

Iter   Nsel  Dwn        Crit       Xdiff        Gmax
   1      2    1  279.082344  0.24010184  43.0886938
   2      2    0  267.820080  0.02924351  90.9730279
   3      2    0  267.027124  0.00975117  11.7981662
   4      2    0  266.930099  0.00414408  5.41741406
   5      2    0  266.911168  0.00193464  2.63339552
   6      2    0  266.906884  9.433e-004  1.30941104
   7      2    0  266.905846  4.697e-004  0.65845388
   8      2    0  266.905587  2.364e-004  0.33299046
   9      2    0  266.905521  1.196e-004  0.16888063
  10      2    0  266.905504  6.067e-005  0.08577432
          4  2.1544347     2    10  255.532495  14.9145250

Iter   Nsel  Dwn        Crit       Xdiff        Gmax
   1      3    0  329.597068  0.30762254  119.896850
   2      3    0  285.317987  0.03039389  106.320530
   3      3    0  281.801195  0.00682939  92.7707242
   4      3    0  281.128781  0.00157706  92.1354748
   5      3    0  280.980227  4.365e-004  91.8879490
   6      3    0  280.946005  1.757e-004  91.6974609
   7      3    0  280.937980  8.612e-005  91.5674971
          5  5.9948425     3     7  259.581704  14.9687203

Iter   Nsel  Dwn        Crit       Xdiff        Gmax
   1      1    2  470.157286  0.11502549  333.620107
   2      1    0  316.119128  3.654e-004  24.1106129
   3      1    0  316.114586  1.141e-005  0.75260746
          6  16.681005     1     3  261.960404  14.4655761

Iter   Nsel  Dwn        Crit       Xdiff        Gmax
   1      3    0  861.274443  0.37133561  928.317767
```

```
   2      3    0  413.170066  0.01615460  862.937421
   3      3    0  396.956697  0.01434796  613.809211
   4      3    0  388.912672  0.01317708  646.153463
   5      3    0  384.637812  0.01067571  671.052686
   6      3    0  382.268047  0.00838079  682.314021
   7      3    0  380.835985  0.00660070  687.267780
   8      3    0  379.903932  0.00527036  689.491606
   9      3    0  379.264099  0.00427561  690.459676
  10      3    0  378.807593  0.00352252  690.811270
  ..................................................
  55      2    0  377.084166  1.089e-004  11.2430152
  56      2    0  377.082935  1.047e-004  10.9388195
  57      2    0  377.081784  1.008e-004  10.6461086
  58      2    0  377.080705  9.706e-005  10.3642818
          7  46.415888     2    58  325.800193  18.0441472

Iter   Nsel  Dwn        Crit       Xdiff        Gmax
   1      3    0  1949.58117  0.38266658  2583.09933
   2      3    0  635.371818  0.02763656  2316.80239
   3      3    0  548.975894  0.02067005  1750.71062
   4      3    0  505.119810  0.01444421  1846.22418
   5      3    0  482.117891  0.00886600  1934.19999
   6      3    0  470.276020  0.00515792  1974.69495
   7      3    0  463.969846  0.00294184  1992.70854
   8      3    0  460.518351  0.00166458  2001.37132
   9      3    0  458.601822  9.385e-004  2005.83194
  10      3    0  457.530600  5.282e-004  2008.22446
  11      3    0  456.930164  2.970e-004  2009.53600
  12      2    1  456.593216  1.669e-004  2010.26318
  13      2    0  456.404028  9.377e-005  1132.01831
          8  129.15497     2    13  335.910146  18.4569184

Iter   Nsel  Dwn        Crit       Xdiff        Gmax
   1      3    0  4977.85927  0.38471281  7187.62733
   2      3    0  1157.23640  0.03197359  6404.24372
   3      3    0  852.481323  0.01602010  5234.72119
   4      3    0  737.340973  0.00831917  5304.99324
   5      3    0  686.807162  0.00382429  5505.71870
   6      3    0  666.700691  0.00162073  5623.72921
   7      3    0  658.936002  6.617e-004  5675.76285
   8      3    0  655.921683  2.661e-004  5696.65181
   9      3    0  654.740235  1.064e-004  5704.82014
  10      3    0  654.273984  4.243e-005  5708.00575
          9  359.38137     3    10  370.741716  20.0736139
```

```
      Iter   Nsel  Dwn        Crit       Xdiff        Gmax
         1      3    0  13404.2230  0.37920767  20000.0000
         2      3    0  2255.21628  0.03131396  18137.5762
         3      3    0  1332.11357  0.00970885  16161.7923
         4      3    0  1125.96537  0.00345334  15769.2026
         5      3    0  1064.04892  0.00125033  15741.7928
         6      3    0  1043.46338  6.390e-004  15799.1126
         7      3    0  1036.37505  4.322e-004  15861.8697
         8      3    0  1033.91060  2.804e-004  15910.6046
         9      3    0  1033.05310  1.774e-004  15944.2057
        10      3    0  1032.75518  1.106e-004  15966.1158
        11      2    1  1032.65184  6.837e-005  15979.9910
                10  1000.0000     2    11  642.331232  33.3995743

      Iter   Nsel  Dwn        Crit       Xdiff        Gmax
         1      3    0  251.061803  0.00735490  2.00000000
         2      3    0  251.080258  8.912e-004  0.06691847
         3      3    0  251.082706  1.106e-004  0.00849913
         4      3    0  251.083013  1.376e-005  0.00105987
  2  0.7500000   1  0.1000000     3     4  250.746698  17.2894487


   ..................................................
```

Now validate the last feasible pair $(\lambda, \delta) = (1000., 2.0)$:

```
      Iter   Nsel  Dwn        Crit       Xdiff        Gmax
         1      2    0  13099.2297  0.39451293  24316.5345
         2      2    0  2688.12285  0.03737396  17255.1830
         3      2    0  1617.25525  0.01412662  14881.7425
         4      2    0  1338.65777  0.00651959  14105.7915
         5      2    0  1238.77734  0.00325390  13785.3380
         6      2    0  1196.94690  0.00168240  13637.3847
         7      2    0  1177.73683  8.849e-004  13564.7416
         8      2    0  1168.38680  4.695e-004  13527.7804
         9      2    0  1163.66521  2.502e-004  13508.5755
        10      2    0  1161.22540  1.336e-004  13498.4724
        11      2    0  1159.94667  7.148e-005  13493.1183
                10  1000.0000     2    11  1156.84465  57.8450848


 Best GCValue Ridge=0 : ErrorFraction=1.75 Lambda=5.99484 Nsel=1

 Best Estimates
 --------------
```

```
1 :           0          0     0.1657
```

Iter	Nsel	Dwn	Crit	Xdiff	Gmax
1	2	1	329.597068	0.30762254	119.896850
2	2	0	289.940220	0.02836886	816.620082
3	1	1	282.202546	0.00577972	78.2151665
4	1	0	281.337744	0.00290129	187.548348
5	1	0	281.062685	3.191e-005	2.06306312

Total Processing Time: 0
The number of selected features out of 3 : 1
The average shrinkage factor: 0.055226

Parameter Estimates
****

Dense Matrix (3 by 4)

	Estimate	AsStdErr	LowerCI	UpperCI
Var_1	0.0000000	0.0000000	0.0000000	0.0000000
Var_2	0.0000000	0.0000000	0.0000000	0.0000000
Var_3	0.1656780	0.0039107	0.1580132	0.1733428

CV Table of GCValues

	Lamb_01	Lamb_02	Lamb_03	Lamb_04
Frac_1	17.28944866	16.79189740	15.31535676	14.91452501
Frac_2	17.28944866	16.79189740	15.31537263	14.91452501
Frac_3	17.28944866	16.79189740	15.31537263	14.91452501
Frac_4	17.28944866	16.79189740	15.31535217	14.91452501
Frac_5	17.28944866	15.48000214	15.31535217	14.91452501
Frac_6	17.28944866	15.48000219	15.31535217	14.91452501
Frac_7	15.55199697	15.48000242	15.31535890	14.91451831

CV Table of GCValues

	Lamb_05	Lamb_06	Lamb_07	Lamb_08
Frac_1	14.96872027	14.46557605	18.04414716	18.45691838
Frac_2	14.96872027	14.46557605	18.04414716	18.45691838
Frac_3	14.45402407	14.46557605	18.04414716	18.45691838
Frac_4	14.45402407	14.46557605	18.04414716	18.45691838
Frac_5	14.45402407	14.46557605	18.04414716	18.45691838

Frac_6	14.45358356	14.46557605	18.04414716	18.45691838
Frac_7	14.45378413	14.46518648	14.70550594	16.92844921

CV Table of GCValues

	Lamb_09	Lamb_10
Frac_1	20.07361386	33.39957428
Frac_2	20.07361386	33.39957428
Frac_3	20.07361386	33.39957428
Frac_4	20.07361386	33.39957428
Frac_5	20.07361386	33.39957428
Frac_6	20.07361386	33.39957428
Frac_7	34.45408851	57.84508478

CV Table of Likelihoods

	Lamb_01	Lamb_02	Lamb_03	Lamb_04
Frac_1	250.7466984	251.0847704	252.7900252	255.5324954
Frac_2	250.7466984	251.0847704	252.7885497	255.5324954
Frac_3	250.7466984	251.0847704	252.7885497	255.5324954
Frac_4	250.7466984	251.0847704	252.7904540	255.5324954
Frac_5	250.7466984	252.4360560	252.7904540	255.5324954
Frac_6	250.7466984	252.4360474	252.7904540	255.5324954
Frac_7	252.3863785	252.4360129	252.7898259	255.5327169

CV Table of Likelihoods

	Lamb_05	Lamb_06	Lamb_07	Lamb_08
Frac_1	259.5817038	261.9604041	325.8001931	335.9101459
Frac_2	259.5817038	261.9604041	325.8001931	335.9101459
Frac_3	261.1904216	261.9604041	325.8001931	335.9101459
Frac_4	261.1904216	261.9604041	325.8001931	335.9101459
Frac_5	261.1904216	261.9604041	325.8001931	335.9101459
Frac_6	261.1945880	261.9604041	325.8001931	335.9101459
Frac_7	261.1982695	261.9532393	267.8578445	313.3654027

CV Table of Likelihoods

	Lamb_09	Lamb_10
Frac_1	370.7417155	642.3312317
Frac_2	370.7417155	642.3312317
Frac_3	370.7417155	642.3312317
Frac_4	370.7417155	642.3312317

Frac_5	370.7417155	642.3312317
Frac_6	370.7417155	642.3312317
Frac_7	666.6620737	1156.844650

CV Table of N Select

	Lamb_01	Lamb_02	Lamb_03	Lamb_04
Frac_1	3.000000000	3.000000000	2.000000000	2.000000000
Frac_2	3.000000000	3.000000000	2.000000000	2.000000000
Frac_3	3.000000000	3.000000000	2.000000000	2.000000000
Frac_4	3.000000000	3.000000000	2.000000000	2.000000000
Frac_5	3.000000000	2.000000000	2.000000000	2.000000000
Frac_6	3.000000000	2.000000000	2.000000000	2.000000000
Frac_7	2.000000000	2.000000000	2.000000000	2.000000000

CV Table of N Select

	Lamb_05	Lamb_06	Lamb_07	Lamb_08
Frac_1	3.000000000	1.000000000	2.000000000	2.000000000
Frac_2	3.000000000	1.000000000	2.000000000	2.000000000
Frac_3	2.000000000	1.000000000	2.000000000	2.000000000
Frac_4	2.000000000	1.000000000	2.000000000	2.000000000
Frac_5	2.000000000	1.000000000	2.000000000	2.000000000
Frac_6	1.000000000	1.000000000	2.000000000	2.000000000
Frac_7	1.000000000	1.000000000	2.000000000	2.000000000

CV Table of N Select

	Lamb_09	Lamb_10
Frac_1	3.000000000	2.000000000
Frac_2	3.000000000	2.000000000
Frac_3	3.000000000	2.000000000
Frac_4	3.000000000	2.000000000
Frac_5	3.000000000	2.000000000
Frac_6	3.000000000	2.000000000
Frac_7	2.000000000	2.000000000

Approximate Covariance Matrix of Parameter Estimates

****

Dense Symmetric Matrix (3 by 3)

S	Var_1	Var_2	Var_3

```
          ------------------------------------
Var_1 |   1.53e-005
Var_2 |  -4.27e-006  1.75e-005
Var_3 |  -5.45e-005 -9.60e-005  2.41e-004
```

2. SCAD for linear least squares regression: Inner Iteration Algorithm Data: Middle-Aged Men in Health Fitness Club: SAS/STAT: Linnerud, NC State University:

   The alternative algorithm is specified by setting the `techsc` option to "inner":

```
optn = [ "aparm"     3.7 ,
         "xtol"    1.e-4 ,
         "ridg"       0. ,
         "techsc"  "inn" ,
         "pinit"       2 ,
         "ptab"        2 ,
         "phist"       2 ,
         "print"       3 ];
 < gof,parm,gcvtab,cov,yprd > = scad("lse",fit,lamb,frac,model,optn);
```

```
      Initial Parameter Estimates and Standard Errors
      ***********************************************

                  N    Estimate    AsStdErr
                  1  0.03378733  0.01865903
                  2 -0.34690359  0.11454985
                  3  0.27698303  0.02438803
```

The iterations look differently:

```
                Cross Validation Grid Search
                ****************************

I   ErrFract   J     Lambda  Nsel  Nitr        Plik       GCVal
    Iter   Nsel  Dwn        Crit       Xdiff        Gmax
       1             251.061803  0.00735490  2.00000000
       1             251.080258  8.912e-004  0.06691847
       1             251.082706  1.106e-004  0.00849913
       1             251.083013  1.376e-005  0.00105987
       1      3    0  251.083013  1.376e-005  0.00105987
1  0.5000000   1  0.1000000     3     1  250.746698  17.2894487


    Iter   Nsel  Dwn        Crit       Xdiff        Gmax
```

```
   1              254.227666  0.07031560  5.56511880
   1              254.084400  0.01786045  1.71901509
   1              254.063859  0.00447770  0.66258200
   1              254.062520  0.00121047  0.19063699
   1              254.062421  3.339e-004  0.05349581
   1              254.062414  9.260e-005  0.01490876
   1      3    0  254.062414  9.260e-005  0.01490876
          2  0.2782559     3     1  251.084770  16.7918974

Iter   Nsel  Dwn        Crit       Xdiff        Gmax
   1              260.928294  0.15100674  15.4852737
   1              258.756980  0.04526859  10.0992913
   1              258.451022  0.01970498  9.12852290
   1              258.366763  0.00909099  9.65074511
   1              258.338399  0.00338805  11.3924767
   1              258.330322  7.401e-004  13.9308895
   1              258.329414  1.132e-004  15.2203022
   1              258.329395  1.893e-005  14.7334662
   1      2    1  258.329395  1.893e-005  14.7334662
   2              258.329395  3.179e-006  0.00221996
   2      2    0  258.329395  3.179e-006  0.00221996
          3  0.7742637     2     2  252.791214  15.3153440

Iter   Nsel  Dwn        Crit       Xdiff        Gmax
   1              279.082344  0.24010184  43.0886938
   1              267.766006  0.03325951  42.4782785
   1              266.921243  0.01621428  49.6064113
   1              266.628523  0.01545301  68.7047305
   1              266.290323  0.01406869  48.2466985
   1              265.999165  0.01012402  22.9856480
   1              265.813045  0.00655937  15.6882247
   1              265.700229  0.00413416  15.9841097
   1              265.631402  0.00259350  16.0152681
   1              265.588993  0.00162807  15.9901494
   1              265.562654  0.00102354  15.9609188
   1              265.546204  6.443e-004  15.9381511
   1              265.535892  4.059e-004  15.9222834
   1              265.529412  2.559e-004  15.9117211
   1              265.525333  1.614e-004  15.9048478
   1              265.522764  1.018e-004  15.9004298
   1              265.521143  6.425e-005  15.8976102
   1      3    0  265.521143  6.425e-005  15.8976102
          4  2.1544347     3     1  255.878723  15.5316609
```

```
Iter   Nsel  Dwn        Crit       Xdiff        Gmax
   1              329.597068  0.30762254  119.896850
   1              285.317987  0.03039389  106.320530
   1              281.801195  0.00682939  92.7707242
   1              281.128781  0.00157706  92.1354748
   1              280.980227  4.365e-004  91.8879490
   1              280.946005  1.757e-004  91.6974609
   1              280.937980  8.612e-005  91.5674971
   1      3    0  280.937980  8.612e-005  91.5674971
          5  5.9948425     3     1  259.581704  14.9687203
```

```
Iter   Nsel  Dwn        Crit       Xdiff        Gmax
   1              470.157286  0.34805673  333.620107
   1              316.479680  0.00205997  332.511120
   1              316.100556  5.696e-004  249.445232
   1              316.025739  7.296e-004  248.336919
   1              315.988623  8.550e-004  249.618261
   1              315.960908  9.348e-004  251.032112
   1              315.936364  9.724e-004  252.500859
   1              315.914410  9.675e-004  253.968231
   1              315.895590  9.238e-004  255.371176
   1              315.880303  8.498e-004  256.657119
   1              315.868522  7.564e-004  257.792611
   1              315.859861  6.548e-004  258.764340
   1              315.853748  5.538e-004  259.575245
   1              315.849578  4.596e-004  260.238821
   1              315.846810  3.756e-004  260.773828
   1              315.845016  3.032e-004  261.200424
   1              315.843872  2.424e-004  261.537810
   1              315.843154  1.923e-004  261.803045
   1              315.842708  1.517e-004  262.010644
   1              315.842433  1.191e-004  262.172605
   1              315.842265  9.319e-005  262.298661
   1      2    1  315.842265  9.319e-005  262.298661
   2              315.842163  7.270e-005  2.35878010
   2      2    0  315.842163  7.270e-005  2.35878010
          6  16.681005     2     2  264.660582  14.9722522
```

```
Iter   Nsel  Dwn        Crit       Xdiff        Gmax
   1              861.274443  0.37133561  928.317767
   1              413.170066  0.01615460  862.937421
   1              396.956697  0.01434796  613.809211
   1              388.912672  0.01317708  646.153463
   ..............................................
   1              377.081784  1.008e-004  685.302808
```

```
   1              377.080705  9.706e-005  685.268006
   1      2    1  377.080705  9.706e-005  685.268006
   2              377.079694  9.351e-005  10.0927790
   2      2    0  377.079694  9.351e-005  10.0927790
           7  46.415888     2     2  325.861625  18.0470522

Iter   Nsel  Dwn        Crit       Xdiff        Gmax
   1              1949.58117  0.38266658  2583.09933
   1              635.371818  0.02763656  2316.80239
   1              548.975894  0.02067005  1750.71062
   1              505.119810  0.01444421  1846.22418
   1              482.117891  0.00886600  1934.19999
   1              470.276020  0.00515792  1974.69495
   1              463.969846  0.00294184  1992.70854
   1              460.518351  0.00166458  2001.37132
   1              458.601822  9.385e-004  2005.83194
   1              457.530600  5.282e-004  2008.22446
   1              456.930164  2.970e-004  2009.53600
   1              456.593216  1.669e-004  2010.26318
   1              456.404039  9.377e-005  2010.66877
   1      2    1  456.404039  9.377e-005  2010.66877
   2              456.297803  5.267e-005  1132.16053
   2      2    0  456.297803  5.267e-005  1132.16053
           8  129.15497     2     2  335.966530  18.4598815

Iter   Nsel  Dwn        Crit       Xdiff        Gmax
   1              4977.85927  0.38471281  7187.62733
   1              1157.23640  0.03197359  6404.24372
   1              852.481323  0.01602010  5234.72119
   1              737.340973  0.00831917  5304.99324
   1              686.807162  0.00382429  5505.71870
   1              666.700691  0.00162073  5623.72921
   1              658.936002  6.617e-004  5675.76285
   1              655.921683  2.661e-004  5696.65181
   1              654.740235  1.064e-004  5704.82014
   1              654.273984  4.243e-005  5708.00575
   1      3    0  654.273984  4.243e-005  5708.00575
           9  359.38137     3     1  370.741716  20.0736139

Iter   Nsel  Dwn        Crit       Xdiff        Gmax
   1              13404.2230  0.37920767  20000.0000
   1              2255.21628  0.03131396  18137.5762
   1              1332.11357  0.00970885  16161.7923
   1              1125.96537  0.00345334  15769.2026
```

```
      1              1064.04892  0.00125033  15741.7928
      1              1043.46338  6.390e-004  15799.1126
      1              1036.37505  4.322e-004  15861.8697
      1              1033.91060  2.804e-004  15910.6046
      1              1033.05310  1.774e-004  15944.2057
      1              1032.75518  1.106e-004  15966.1158
      1              1032.65184  6.837e-005  15979.9910
      1      2    1  1032.65184  6.837e-005  15979.9910
      2              1032.61597  4.203e-005  13187.6717
      2      2    0  1032.61597  4.203e-005  13187.6717
             10  1000.0000     2     2  640.972725  33.3317497


   Iter   Nsel  Dwn        Crit       Xdiff        Gmax
      1              251.061803  0.00735490  2.00000000
      1              251.080258  8.912e-004  0.06691847
      1              251.082706  1.106e-004  0.00849913
      1              251.083013  1.376e-005  0.00105987
      1      3    0  251.083013  1.376e-005  0.00105987
2  0.7500000   1  0.1000000     3     1  250.746698  17.2894487
```

However, the result looks similar:

```
 Best GCValue Ridge=0 : ErrorFraction=1.5 Lambda=5.99484 Nsel=2

Best Estimates
--------------

  1 :           0 -8.45e-007     0.1657


   Iter   Nsel  Dwn        Crit       Xdiff        Gmax
      1              329.597068  0.30762254  119.896850
      1              285.317987  0.03039389  106.320530
      1              281.801195  0.00682939  92.7707242
      1              281.128781  0.00157706  92.1354748
      1              280.980227  4.365e-004  91.8879490
      1              280.946005  1.757e-004  91.6974609
      1              280.937980  8.612e-005  91.5674971
      1      2    1  280.937980  8.612e-005  91.5674971
      2              283.836531  0.00921816  595.496575
      2              281.063152  9.865e-005  74.1199443
      2      2    0  281.063152  9.865e-005  74.1199443


                Total Processing Time: 0
          The number of selected features out of 3 : 2
```

```
The average shrinkage factor: 0.0552261

                Parameter Estimates
                *******************

                Dense Matrix (3 by 4)

       |    Estimate   AsStdErr    LowerCI    UpperCI
-------------------------------------------------------
Var_1 |   0.0000000  0.0000000  0.0000000  0.0000000
Var_2 |  -8.45e-007  5.62e-007 -1.95e-006  2.56e-007
Var_3 |   0.1656793  0.0039107  0.1580145  0.1733441
```

Here we only show the number of selected variables:

```
                CV Table of N Select

           Lamb_01      Lamb_02      Lamb_03      Lamb_04
Frac_1   3.000000000  3.000000000  2.000000000  3.000000000
Frac_2   3.000000000  3.000000000  2.000000000  3.000000000
Frac_3   3.000000000  3.000000000  2.000000000  3.000000000
Frac_4   3.000000000  3.000000000  2.000000000  3.000000000
Frac_5   3.000000000  2.000000000  2.000000000  3.000000000
Frac_6   3.000000000  2.000000000  2.000000000  3.000000000
Frac_7   2.000000000  2.000000000  2.000000000  2.000000000

                CV Table of N Select

           Lamb_05      Lamb_06      Lamb_07      Lamb_08
Frac_1   3.000000000  2.000000000  2.000000000  2.000000000
Frac_2   3.000000000  2.000000000  2.000000000  2.000000000
Frac_3   3.000000000  2.000000000  2.000000000  2.000000000
Frac_4   3.000000000  2.000000000  2.000000000  2.000000000
Frac_5   2.000000000  2.000000000  2.000000000  2.000000000
Frac_6   1.000000000  2.000000000  2.000000000  2.000000000
Frac_7   1.000000000  2.000000000  2.000000000  2.000000000

                CV Table of N Select

           Lamb_09      Lamb_10
Frac_1   3.000000000  2.000000000
Frac_2   3.000000000  2.000000000
Frac_3   3.000000000  2.000000000
Frac_4   3.000000000  2.000000000
```

```
Frac_5   3.000000000  2.000000000
Frac_6   3.000000000  2.000000000
Frac_7   2.000000000  2.000000000


Approximate Covariance Matrix of Parameter Estimates
****************************************************

          Dense Symmetric Matrix (3 by 3)

          S |       Var_1      Var_2      Var_3
          -------------------------------------
      Var_1 |   3.16e-013
      Var_2 |   1.38e-010  1.53e-005
      Var_3 |  -1.54e-005 -1.98e-008  6.13e-005
```

3. SCAD for Proportional Hazards Regression: Example of Fan and Li (2002) see also: Benner (2006)

```
options NOECHO;
 %inc "C:\\tdata2\\ulrich\\scadch12.dat";
options ECHO;

 vnam = [ "Yresp" "a" "treatment" "gender" "married" "what" "status" ];
 dlta = 1 - scadch12[,7];
 c6  = scadch12[,6];
 h1 = (c6 .== 2); h2 = (c6 .== 4);
 h3 = (c6 .== 5); hh = h1 -> h2 -> h3; /* print hh; */
 d = 4. / 9.; loglam = [ -1. : d : 3. ];
 lamb = 10 ** loglam; print "Lambda", lamb;
 frac = [.5 : .25 : 2. ]; print "Error Fractions=", frac;


 /*---------------------------------------------------------------*/
 /* Case 2: column order for easy model statement:
            interactions are with | operator */
 scad2 = scadch12[,1:5] -> hh -> dlta;
 model = "1 = 2|3|4|5@2 6:8";
 class = [ 6:9 ];


 optn = [ "aparm"     3.7 ,
          "cen"         9 ,  /* column for censor variable */
          "xtol"    1.e-4 ,
          "ridg"       0. ,
          "techsc"  "ori" ,
          "pinit"       2 ,
```

```
          "ptab"        2 ,
          "phist"       2 ,
          "print"       3 ];
  < gof,parm,gcvtab,cov,restrn > =
        scad("phr",scad2,lamb,frac,model,optn,class);
print "GOF=", gof;
print "PARM=", parm;
print "GCVTAB=", gcvtab;
print "COVmat=", cov;
```

****
Model Information
****

Number Valid Observations	1601
Response Variable	Y[1]
N Independend Variables	7
Original PHREG SCAD (Benner)	
Ridge Lambda	0.0000000
Use Unscaled Predictor	
Linear Model Without Intercept	
Censor Variable Column	9

****
Model Effects
****

```
X2 + X3 + X2 * X3 + X4 + X2 * X4 + X3 * X4 + X5 + X2 * X5 +
                X3 * X5 + X4 * X5 + C6 + C7 + C8
```

****
Class Level Information
****

Class	Level	Value
C[6]	2	0 1
C[7]	2	0 1
C[8]	2	0 1

****
Simple Statistics
****

Column	Nobs	Mean	Std Dev	Skewness	Kurtosis
Y[1]	1601	241.23111	275.27033	1.2579456	0.5915896
X[2]	1601	82.645222	7.7069160	-0.2190298	-0.4944203
X[3]	1601	0.4447220	0.4970902	0.2226843	-1.9528528
X[4]	1601	0.2642099	0.4410493	1.0705607	-0.8549693
X[5]	1601	0.1717676	0.3772962	1.7420949	1.0361874

****
Censor Variable
****

Value	Nobs	Proportion
Alive	322	20.112430
Dead	1279	79.887570

****
Number of Observations for Class Levels
****

Variable	Value	Nobs	Proportion
C[6]	0	1258	78.575890
	1	343	21.424110
C[7]	0	1088	67.957527
	1	513	32.042473
C[8]	0	1432	89.444097
	1	169	10.555903

Input Error Fraction Values
---

1 :	0.5	0.75	1	1.25	1.5
6 :	1.75	2			

Input SCAD Lambda Values
---

1 :	0.1	0.2783	0.7743	2.154	5.995
6 :	16.68	46.42	129.2	359.4	1000

Initial Values Computed with 4 Initial Iterations

Initial Parameter Estimates and Standard Errors

N	Estimate	AsStdErr
1	-0.01619046	0.00588448
2	-0.75634621	0.63635585
3	0.00860702	0.00757399
4	-1.25442431	0.68513058
5	0.02041417	0.00827005
6	-0.10379271	0.13180239
7	-0.79010107	0.85190365
8	0.01225658	0.01064355
9	-0.00408940	0.15754487
10	-0.07460442	0.15504546
11	0.03119433	0.07818168
12	0.24465941	0.06840640
13	0.56719302	0.09595737

Cross Validation Grid Search

I	ErrFract	J	Lambda	Nsel	Nitr	Plik	GCVal
	Iter	Nsel	Dwn	Crit	Xdiff	Gmax	
	1	10	0	-8626.19700	0.50770950	922.542844	
	2	8	2	-8630.54313	0.30769954	152.926131	
	3	8	0	-8602.93289	0.09479186	150.157847	
	4	8	0	-8597.14039	0.04770059	147.245988	
	5	8	0	-8595.98030	0.02578439	147.079480	
	6	8	0	-8595.64764	0.01444450	147.073098	
	7	7	1	-8595.53494	0.00824928	143.758791	
	8	7	0	-8595.49441	0.00476042	143.775746	
	9	6	1	-8595.47955	0.00276242	100.923224	
	10	6	0	-8595.47407	0.00160771	100.909421	
	11	6	0	-8595.47205	9.371e-004	100.904179	
	12	6	0	-8595.47131	5.467e-004	100.902159	
	13	6	0	-8595.47103	3.190e-004	100.901365	
	14	6	0	-8595.47093	1.862e-004	100.901043	
	15	6	0	-8595.47089	1.087e-004	100.900908	
	16	6	0	-8595.47088	6.345e-005	100.900848	
1	0.5000000	1	0.1000000	6	16	-8592.86704	5.39464772

Iter	Nsel	Dwn	Crit	Xdiff	Gmax
1	7	3	-9153.46948	0.96989826	1207.93060
2	6	1	-8785.79625	0.08669561	435.057171
3	6	0	-8615.32430	0.00962061	435.820409

```
   4      6    0 -8610.61225  0.00108413  436.915175
   5      5    1 -8610.06163  2.744e-004  402.466223
   6      5    0 -8609.98254  1.868e-004  402.547547
   7      5    0 -8609.96341  1.352e-004  402.603486
   8      4    1 -8609.95493  1.023e-004  394.984940
   9      3    1 -8609.94984  8.003e-005  45.1764030
          2  0.2782559     3     9 -8606.79250  5.38861987

Iter   Nsel  Dwn        Crit       Xdiff        Gmax
   1      7    3 -12896.0698  0.79101496  2002.03900
   2      6    1 -8659.16554  0.02451090  1238.69286
   3      6    0 -8616.92976  9.942e-004  1228.35909
   4      5    1 -8614.90799  2.374e-004  1196.51016
   5      5    0 -8614.66787  9.417e-005  1196.55556
          3  0.7742637     5     5 -8607.79149  5.38594511

Iter   Nsel  Dwn        Crit       Xdiff        Gmax
   1      8    2 -21599.8463  1.25675735  4211.69278
   2      7    1 -8687.86271  0.00938111  3443.97841
   3      7    0 -8626.09767  3.280e-004  3425.45641
   4      4    3 -8624.26187  1.806e-004  3399.13018
   5      3    1 -8623.97762  1.164e-004  2848.53440
   6      3    0 -8623.87669  8.211e-005  2841.72974
          4  2.1544347     3     6 -8612.37414  5.38404061

Iter   Nsel  Dwn        Crit       Xdiff        Gmax
   1      7    3 -44804.1660  1.25689392  10360.1857
   2      7    0 -8742.46269  0.00343520  9583.88376
   3      7    0 -8640.30628  6.818e-004  9558.83489
   4      3    4 -8634.67093  3.629e-004  8903.79027
   5      3    0 -8632.76197  2.229e-004  8881.67915
   6      2    1 -8631.86326  1.488e-004  5403.58464
   7      2    0 -8631.37555  1.048e-004  5324.81797
   8      2    0 -8631.08366  7.663e-005  5275.48303
          5  5.9948425     2     8 -8627.64973  5.38925166

Iter   Nsel  Dwn        Crit       Xdiff        Gmax
   1      8    2 -109371.564  1.25637098  27468.7324
   2      8    0 -8838.46299  0.00197370  26698.7842
   3      4    4 -8648.17938  5.004e-004  26623.6534
   4      2    2 -8634.75106  1.409e-004  22347.0603
   5      2    0 -8631.55354  4.103e-005  22271.6121
          6  16.681005     2     5 -8629.87894  5.39032713
```

```
      Iter   Nsel  Dwn        Crit       Xdiff        Gmax
         1      9    1 -289034.185  1.25558301  75074.2801
         2      9    0 -8958.37952  0.00140538  74293.1909
         3      3    6 -8643.96535  1.457e-004  73564.4398
         4      2    1 -8631.64768  1.541e-005  69859.5268
                7  46.415888     2     4 -8630.21010  5.39051463

      Iter   Nsel  Dwn        Crit       Xdiff        Gmax
         1     10    0 -788956.098  1.25496418  207539.544
         2      8    2 -9052.52422  6.996e-004  206732.132
         3      2    6 -8637.13759  2.660e-005  202332.210
                8  129.15497     2     3 -8630.11226  5.39045259

      Iter   Nsel  Dwn        Crit       Xdiff        Gmax
         1     10    0 -2180018.52  1.25464116  576132.010
         2      5    5 -9098.80911  2.876e-004  575279.462
         3      1    4 -8633.12526  3.939e-006  567493.047
                9  359.38137     1     3 -8630.32001  5.39058097

      Iter   Nsel  Dwn        Crit       Xdiff        Gmax
         1     10    0 -6050732.34  1.25450567  1601762.44
         2      2    8 -9118.19181  1.088e-004  1596619.56

      Iter   Nsel  Dwn        Crit       Xdiff        Gmax
         1     10    0 -8626.19700  0.50770950  922.542844
         2      8    2 -8630.54313  0.30769954  152.926131
         3      8    0 -8602.93289  0.09479186  150.157847
         4      8    0 -8597.14039  0.04770059  147.245988
         5      8    0 -8595.98030  0.02578439  147.079480
         6      8    0 -8595.64764  0.01444450  147.073098
         7      7    1 -8595.53494  0.00824928  143.758791
         8      7    0 -8595.49441  0.00476042  143.775746
         9      6    1 -8595.47955  0.00276242  100.923224
        10      6    0 -8595.47407  0.00160771  100.909421
        11      6    0 -8595.47205  9.371e-004  100.904179
        12      6    0 -8595.47131  5.467e-004  100.902159
        13      6    0 -8595.47103  3.190e-004  100.901365
        14      6    0 -8595.47093  1.862e-004  100.901043
        15      6    0 -8595.47089  1.087e-004  100.900908
        16      6    0 -8595.47088  6.345e-005  100.900848
  2  0.7500000   1  0.1000000     6    16 -8592.86704  5.39464772

     ..............................................
```

```
   Best GCValue Ridge=0 : ErrorFraction=1.75 Lambda=0.1 Nsel=3

 Best Estimates
 --------------

   1 :           0           0           0           0     0.00503
   6 :           0           0           0           0           0
  11 :           0  3.443e-005       0.451
```

Reiterate for best $(\lambda, \delta) = (0.1, 1.75)$:

Iter	Nsel	Dwn	Crit	Xdiff	Gmax
1	5	0	-8611.72819	0.51363747	681.746187
2	4	1	-8609.86457	0.07310553	121.567513
3	3	1	-8644.81470	0.02341925	12874.5545
4	3	0	-8600.04885	0.00833665	696.367087
5	3	0	-8599.04914	0.00302775	101.532835
6	3	0	-8598.74278	0.00111870	100.784012
7	3	0	-8598.63019	4.160e-004	100.508416
8	3	0	-8598.58840	1.550e-004	100.405998
9	3	0	-8598.57284	5.784e-005	100.367831

```
          Total Processing Time: 127
   The number of selected features out of 13 : 3
      The average shrinkage factor: 0.0350794
```

Parameter Estimates
****

Dense Matrix (13 by 4)

	Estimate	AsStdErr	LowerCI	UpperCI
Var_01	0.0000000	0.0000000	0.0000000	0.0000000
Var_02	0.0000000	0.0000000	0.0000000	0.0000000
Var_03	0.0000000	0.0000000	0.0000000	0.0000000
Var_04	0.0000000	0.0000000	0.0000000	0.0000000
Var_05	0.0050295	7.49e-004	0.0035610	0.0064981
Var_06	0.0000000	0.0000000	0.0000000	0.0000000
Var_07	0.0000000	0.0000000	0.0000000	0.0000000
Var_08	0.0000000	0.0000000	0.0000000	0.0000000
Var_09	0.0000000	0.0000000	0.0000000	0.0000000
Var_10	0.0000000	0.0000000	0.0000000	0.0000000

Var_11	0.0000000	0.0000000	0.0000000	0.0000000
Var_12	3.44e-005	9.04e-006	1.67e-005	5.22e-005
Var_13	0.4509683	0.0866009	0.2812336	0.6207029

CV Table of GCValues

	Lamb_01	Lamb_02	Lamb_03	Lamb_04
Frac_1	5.394647717	5.388619874	5.385945114	5.384040610
Frac_2	5.394647717	5.388619874	5.385945114	5.384040607
Frac_3	5.394647712	5.388656613	5.385974741	5.384071246
Frac_4	5.390247731	5.385232860	5.383838335	5.383836968
Frac_5	5.387160570	5.385232860	5.383838335	5.383836968
Frac_6	5.383559904	5.385232860	5.383838335	5.383836968
Frac_7	5.383559939	5.384219204	5.383822511	5.383825213

CV Table of GCValues

	Lamb_05	Lamb_06	Lamb_07	Lamb_08
Frac_1	5.389251665	5.390327135	5.390514629	5.390452589
Frac_2	5.389251665	5.390327135	5.390514631	5.390452649
Frac_3	5.389246323	5.390297007	5.390496166	5.390409048
Frac_4	5.389204135	5.390327599	5.390519414	5.390469495
Frac_5	5.389204135	5.390327599	5.390519414	5.390469495
Frac_6	5.389204135	5.390327599	5.390519414	5.390469495
Frac_7	5.389175938	5.390304452	5.390508071	5.390447189

CV Table of GCValues

	Lamb_09	Lamb_10
Frac_1	5.390580970	.
Frac_2	5.390580979	.
Frac_3	5.390574114	.
Frac_4	5.390583583	.
Frac_5	5.390583583	.
Frac_6	5.390583583	.
Frac_7	5.390580053	.

CV Table of Likelihoods

	Lamb_01	Lamb_02	Lamb_03	Lamb_04
Frac_1	-8592.867044	-8606.792501	-8607.791491	-8612.374138
Frac_2	-8592.867044	-8606.792501	-8607.791491	-8612.374131
Frac_3	-8592.867050	-8606.768703	-8607.767502	-8612.470389

Frac_4	-8595.432322	-8609.104968	-8610.261741	-8613.227129
Frac_5	-8596.599304	-8609.104968	-8610.261741	-8613.227129
Frac_6	-8597.752840	-8609.104968	-8610.261741	-8613.227129
Frac_7	-8597.752919	-8609.926926	-8610.305211	-8613.201367

CV Table of Likelihoods

	Lamb_05	Lamb_06	Lamb_07	Lamb_08
Frac_1	-8627.649726	-8629.878941	-8630.210103	-8630.112257
Frac_2	-8627.649726	-8629.878941	-8630.210106	-8630.112368
Frac_3	-8627.637508	-8629.826583	-8630.179695	-8630.041843
Frac_4	-8627.557277	-8629.880137	-8630.218030	-8630.139652
Frac_5	-8627.557277	-8629.880137	-8630.218030	-8630.139652
Frac_6	-8627.557277	-8629.880137	-8630.218030	-8630.139652
Frac_7	-8627.500901	-8629.840173	-8630.199386	-8630.103603

CV Table of Likelihoods

	Lamb_09	Lamb_10
Frac_1	-8630.320012	.
Frac_2	-8630.320028	.
Frac_3	-8630.308996	.
Frac_4	-8630.324214	.
Frac_5	-8630.324214	.
Frac_6	-8630.324214	.
Frac_7	-8630.318543	.

CV Table of N Select

	Lamb_01	Lamb_02	Lamb_03	Lamb_04
Frac_1	6.000000000	3.000000000	5.000000000	3.000000000
Frac_2	6.000000000	3.000000000	5.000000000	3.000000000
Frac_3	6.000000000	4.000000000	5.000000000	3.000000000
Frac_4	5.000000000	3.000000000	2.000000000	2.000000000
Frac_5	4.000000000	3.000000000	2.000000000	2.000000000
Frac_6	3.000000000	3.000000000	2.000000000	2.000000000
Frac_7	3.000000000	3.000000000	3.000000000	1.000000000

CV Table of N Select

	Lamb_05	Lamb_06	Lamb_07	Lamb_08
Frac_1	2.000000000	2.000000000	2.000000000	2.000000000
Frac_2	2.000000000	2.000000000	2.000000000	2.000000000

Frac_3	2.000000000	2.000000000	2.000000000	2.000000000
Frac_4	1.000000000	1.000000000	1.000000000	1.000000000
Frac_5	1.000000000	1.000000000	1.000000000	1.000000000
Frac_6	1.000000000	1.000000000	1.000000000	1.000000000
Frac_7	1.000000000	1.000000000	1.000000000	1.000000000

CV Table of N Select

	Lamb_09	Lamb_10
Frac_1	1.000000000	0.000000000
Frac_2	1.000000000	0.000000000
Frac_3	2.000000000	0.000000000
Frac_4	1.000000000	0.000000000
Frac_5	1.000000000	0.000000000
Frac_6	1.000000000	0.000000000
Frac_7	1.000000000	0.000000000

Approximate Covariance Matrix of Parameter Estimates

****

Dense Symmetric Matrix (13 by 13)

S	Var_01	Var_02	Var_03	Var_04	Var_05
Var_01	-5.61e-007				
Var_02	-2.11e-012	-8.17e-011			
Var_03	-1.86e-006	-3.36e-011	-0.0074997		
Var_04	-1.70e-005	-2.41e-006	-4.11e-006	-0.0076378	
Var_05	-1.22e-005	0.0012981	-1.79e-005	-3.23e-005	-0.0075601
Var_06	7.26e-021	1.33e-021	2.25e-019	1.12e-020	1.68e-021
Var_07	-5.70e-013	-6.01e-013	-7.50e-013	-1.53e-012	-2.49e-012
Var_08	9.27e-013	-4.10e-012	-3.43e-012	3.09e-012	4.79e-012
Var_09	-4.92e-013	-4.86e-013	-6.36e-013	-8.16e-013	-1.48e-012
Var_10	8.08e-013	-4.80e-012	-4.09e-012	1.61e-012	2.84e-012
Var_11	-1.68e-005	-0.0024625	2.70e-005	-9.36e-004	9.14e-006
Var_12	-3.30e-006	-0.0012372	1.72e-005	-0.0014474	1.81e-005
Var_13	1.85e-005	0.0049140	-4.86e-005	0.0033748	-4.07e-005

S	Var_06	Var_07	Var_08	Var_09	Var_10
Var_06	-1.28e-020				
Var_07	-1.25e-012	-6.70e-012			
Var_08	8.83e-013	2.79e-012	-1.66e-011		
Var_09	-1.70e-014	-5.19e-013	-7.92e-013	-6.64e-012	

```
Var_10 |  -1.08e-013  2.17e-013  4.69e-013  2.77e-012 -1.68e-011
Var_11 |   4.12e-004 -4.00e-004  1.31e-006 -2.50e-004  2.15e-004
Var_12 |  -7.68e-005 -0.0013426  1.50e-005 -3.96e-006  3.56e-004
Var_13 |   2.81e-005  0.0022207 -1.82e-005 -7.62e-004 -9.01e-004


     S |      Var_11     Var_12     Var_13
     -------------------------------------
Var_11 |  -0.0061124
Var_12 |  -0.0022777 -0.0046794
Var_13 |  -0.0022190 -0.0022695 -0.0092078
```

4. SCAD for binary support vector classification: Heart Datafrom Statlog Collection with n=14 variables and first N=60 observations:

```
data = rspfile("..\\tdata\\heart_60.dat");

d = 2. / 9.; loglam = [ -2. : d : 0. ];
lamb = 10 ** loglam; print "Lambda", lamb;
frac = [.001 : .001 : .006 ]; print "Error Fractions=", frac;
modl = "1 = 2:14";
class = 1;
```

Here we prefer to use the option `cdelta` for constant thresholds specified with the `frac` argument.

```
optn = [ "aparm"     3.7 ,
         "initer"    200 ,
         "xtol"    1.e-4 ,
         "ridg"       0. ,
         "cdelta"        , /* constant delta */
         "techsc"  "ori" ,
         "pinit"       2 ,
         "ptab"        2 ,
         "phist"       2 ,
         "print"       3 ];
< gof,parm,gcvtab,cov,yprd > =
      scad("svm",data,lamb,frac,modl,optn,class);
```

```
                    *****************
                    Model Information
                    *****************

                Number Valid Observations   60
```

```
Response Variable          Y[1]
N Independend Variables     13
Original SV Clas SCAD (Benner)
SCAD a               3.7000000
Constant Delta Values
Ridge Lambda         0.0000000
Use Unscaled Predictor


             *************
             Model Effects
             *************

Intercept + X2  + X3  + X4  + X5  + X6  + X7  + X8  + X9  + X10 +
              X11 + X12 + X13 + X14


            ***********************
            Class Level Information
            ***********************

Class  Level    Value

Y[ 1]      2       -1       1


             *****************
             Simple Statistics
             *****************

Column   Nobs        Mean     Std Dev    Skewness    Kurtosis
X[ 2]      60   0.1013889   0.3874933  -0.1204735  -0.9971807
X[ 3]      60   0.3666667   0.9382036  -0.8085886  -1.3938014
X[ 4]      60   0.4444444   0.6397582  -0.8222727  -0.4477859
X[ 5]      60  -0.3040880   0.2949943   0.5847253   0.3538571
X[ 6]      60  -0.4478691   0.3054052   2.2533417   8.1842160
X[ 7]      60  -0.6333333   0.7804099   1.6790470   0.8462940
X[ 8]      60   0.1666667   0.9943342  -0.3467924  -1.9457610
X[ 9]      60   0.1653944   0.3549292  -0.1804149  -0.8035486
X[10]      60  -0.3000000   0.9619895   0.6452146  -1.6395099
X[11]      60  -0.7107527   0.3498565   1.1575831   0.4200400
X[12]      60  -0.4833333   0.5672314   0.5092276  -0.7363578
X[13]      60  -0.5222222   0.6388745   1.2037471   0.4230041
X[14]      60  -0.1083333   0.9702859   0.1872150  -1.9813204

*************************************
```

```
            Number of Observations for Class Levels
            ***************************************

            Variable       Value    Nobs  Proportion
            Y[ 1]             -1      33   55.000000
                               1      27   45.000000


Input Error Fraction Values
---------------------------

  1 :       0.001      0.002      0.003      0.004      0.005
  6 :       0.006


Input SCAD Lambda Values
------------------------

  1 :        0.01    0.01668    0.02783    0.04642    0.07743
  6 :      0.1292     0.2154     0.3594     0.5995
```

The gmax value in the iteration history refers to the largest value of the right hand side P vector of the linear system. Since P is based (Zhang et.al., 2006) on the reciprocal residuals it is obvious that the values become very large and the linear system numerically rather unstable.

```
            Initial SCAD Iteration (p < N): Ridge=0
            ***************************************

            Iter        Crit       Xdiff        Gmax
               1  0.00000000  1.66103543  0.47500000
               2  1.36819298  0.14038975  1.27941241
               3  1.39061334  0.12008566  2.64993050
               4  1.40943892  0.10871498  17.1871440
               5  1.42944044  0.11030190  1114.24540
               6  1.45204990  0.10852487  3728.51267
               7  1.47702378  0.09896228  28115.3627
               8  1.50296101  0.08057033  208817.184
               9  1.52767821  0.07034316  626320.729
              10  1.55347033  0.07016766  2114172.90


             100  2.33802961  0.03075101  13859534.8
             101  2.34823162  0.03006414  861853.957
             102  2.35813500  0.02896716  18844909.2
             103  2.36756814  0.02725800  5595959.58
```

```
       104  2.37629400  0.02460720  13242371.3
       105  2.38398299  0.02046443  18742372.3
       106  2.39017581  0.01406369  20643196.1
       107  2.39427660  0.00553908  1740424.78
       108  2.39585529  7.672e-005  7503290.27

  Initial Values Computed with 108 Initial Iterations


  Initial Parameter Estimates and Thresholds: Ridge=0
  ***************************************************

                N    Estimate   Threshold
                1  3.50729265  1.54692295
                2 -3.56129028  0.20289707
                3  0.62861810  0.01723217
                4  0.42504932  0.06206437
                5  2.27208195  0.22923720
                6  0.49649260  0.34297116
                7  0.11575410  1.47666720
                8  0.38506571  0.01244077
                9 -1.66041244  0.31074645
               10 -0.32740919  0.03805963
               11 -0.00507776  0.02976426
               12  1.01701755  0.07861504
               13  3.69688051  0.17785721
               14  0.58254549  0.02675542


             Cross Validation Grid Search
             ****************************

I   ErrFract   J     Lambda  Nsel  Nitr        Plik       GCVal
    Iter   Nsel  Dwn        Crit       Xdiff        Gmax
       1     14    0  2.39876441  5.130e-005  59046682.6


1  0.0010000   1  0.0100000    14     1  2.39589363  0.06793583
    Iter   Nsel  Dwn        Crit       Xdiff        Gmax
       1     14    0  2.40382515  6.007e-005  59046682.6


           2  0.0166810    14     1  2.39589363  0.06793538
    Iter   Nsel  Dwn        Crit       Xdiff        Gmax
       1     14    0  2.41786916  7.470e-005  59046682.6
```

```
           3  0.0278256    14     1  2.39589363  0.06793464
```

Iter	Nsel	Dwn	Crit	Xdiff	Gmax
1	14	0	2.45560356	0.03248998	59046682.6
2	14	0	2.45810519	0.03837087	9295936.82
3	14	0	2.46097627	0.03053864	3522489.52
4	14	0	2.46260422	0.01231209	2763788.80
5	13	0	2.46322000	2.338e-004	4500790.39
6	13	0	2.46321804	5.471e-004	1542.81693
7	13	0	2.46306297	0.00237612	2141.48911
8	13	0	2.46231526	0.00431228	42263.1430
9	12	0	2.46105557	0.00220714	144811.531
10	12	0	2.46075951	0.00361358	114.008874
11	12	0	2.46212327	0.00335881	403.318315
12	12	0	2.46349152	0.00386718	3802.94552
13	12	0	2.46500098	0.00444070	37409.1130
14	12	0	2.46666624	0.00492035	292009.669
15	12	0	2.46846996	0.00531933	1839116.07
16	12	0	2.47039794	0.00565693	46285472.9
17	12	0	2.47243866	0.00594654	263932.069
18	12	0	2.47458142	0.00619795	1009062.66
19	12	0	2.47681594	0.00641789	11347252.5
20	12	0	2.47913269	0.00661231	3080961.49
165	12	0	2.83100971	1.505e-004	69422184.1
166	12	0	2.83105195	1.304e-004	20964277.6
167	12	0	2.83108855	1.124e-004	31805231.5
168	12	0	2.83112009	9.639e-005	139728702

```
             4  0.0464159    12   168  2.77542795  0.07227677
```

For space reasons only the first part of the iteration history is shown here. The best solution shows a very sparse vector of estimates where only the intercept and the variable 13 has nonzero loadings:

```
Best GCValue=0.0144258 : ErrorFraction=0.004 Lambda=0.359381 Nsel=2

 Best Estimates
 --------------

   1 :      0.8762          0          0          0          0
   6 :           0          0          0          0          0
  11 :           0          0      1.876          0
```

Iter	Nsel	Dwn	Crit	Xdiff	Gmax

1	14	0	4.75248878	0.10063687	59046682.6
2	13	0	4.71999645	0.30931310	876705.600
3	12	0	4.55006579	0.26730380	40.2975027
4	11	0	4.26895428	0.95042422	6.28127073
5	11	0	3.56659027	0.69095783	4.55712929
6	11	0	2.97893603	0.48346354	15.9433073
7	11	0	2.49475691	0.36419531	11.4848480
8	11	0	2.19825444	0.53723198	10.3043894
9	8	0	1.80801843	0.23970770	32.2060820
10	6	0	1.50044253	0.07017532	16.8513340
11	5	0	1.35508809	0.05860038	11.7087700
12	5	0	1.26858335	0.04482169	45.2573811
13	5	0	1.21190931	0.03319466	95.2346108
14	5	0	1.17941934	0.02394910	215.840594
15	4	0	1.15772234	0.01699589	518.076223
16	3	0	1.14177671	0.01197604	4237.83187
17	3	0	1.13091267	0.00863612	48.9275092
18	3	0	1.12592322	0.00565870	1395.29048
19	3	0	1.12183129	0.00391790	26900.9501
20	3	0	1.11902601	0.00270255	526262.105
21	3	0	1.11709394	0.00186175	10136323.3
22	2	0	1.11576386	4.118e-004	113118490
23	2	0	1.11345897	0.00168398	74.0514983
24	2	0	1.11261992	4.730e-004	263.309695
25	2	0	1.11238423	1.325e-004	939.229566
26	2	0	1.11231819	3.708e-005	3353.22792

****
Evaluation of Training Data Fit
****

Index	Value	StdErr
Absolute Classification Error	16	.
Classification Accuracy	73.33333333	.
Concordant Pairs	53.31088664	.
Discordant Pairs	7.182940516	.
Tied Pairs	39.50617284	.
Goodman-Kruskal Gamma	0.762523191	0.122501031
Kendall Tau_a	0.232203390	.
Kendall Tau_b	0.461279461	0.115104733
Stuart Tau_c	0.456666667	0.114733185
Somers D C|R	0.461279461	0.115272264
Somers D R|C	0.461279461	0.115272264

Classification Table

```
--------------------

          |    Predicted
 Observed |       -1       1
 ---------|-----------------
       -1 |       25       8
        1 |        8      19


Total Processing Time: 1
The number of selected features out of 14 : 2
The average shrinkage factor: 0.196598
Training: Hinge=0.566669 Misclassifications=16 Precision=73.3333 %


Parameter Estimates
*******************

Dense Column Vector (nrow=14)

        C |      Var_01     Var_02     Var_03     Var_04     Var_05
Estimate  |   0.8761749  0.0000000  0.0000000  0.0000000  0.0000000

        C |      Var_06     Var_07     Var_08     Var_09     Var_10
Estimate  |   0.0000000  0.0000000  0.0000000  0.0000000  0.0000000

        C |      Var_11     Var_12     Var_13     Var_14
Estimate  |   0.0000000  0.0000000  1.8761923  0.0000000


CV Table of GCValues

              Lamb_1       Lamb_2       Lamb_3       Lamb_4
   Frac_1  0.067935828  0.067935383  0.067934643  0.072276770
   Frac_2  0.067935828  0.067935383  0.067934643  0.072276762
   Frac_3  0.067935828  0.067935383  0.067934643  0.072276681
   Frac_4  0.067935828  0.067935383  0.067934643  0.072276681
   Frac_5  0.067935828  0.067935383  0.067934643  0.072276681
   Frac_6  0.072278642  0.072277772  0.072277077  0.072276918


CV Table of GCValues

              Lamb_5       Lamb_6       Lamb_7       Lamb_8
   Frac_1  0.072280889  0.056452232  0.032271230  0.016857149
   Frac_2  0.072281026  0.056456164  0.032272860  0.015685474
   Frac_3  0.072281026  0.056452336  0.032272694  0.015637276
   Frac_4  0.072281026  0.056456963  0.041509815  0.014425755
```

```
Frac_5   0.072281026  0.056532545  0.032271598  0.018156386
Frac_6   0.076453389  0.056454497  0.032271305  0.019006875
```

CV Table of GCValues

```
          Lamb_9
Frac_1       .
Frac_2       .
Frac_3       .
Frac_4       .
Frac_5       .
Frac_6       .
```

CV Table of Hinge Values

```
            Lamb_1       Lamb_2       Lamb_3       Lamb_4
Frac_1   2.395893628  2.395893628  2.395893628  2.775427950
Frac_2   2.395893628  2.395893628  2.395893628  2.775427657
Frac_3   2.395893628  2.395893628  2.395893628  2.775424568
Frac_4   2.395893628  2.395893628  2.395893628  2.775424568
Frac_5   2.395893628  2.395893628  2.395893628  2.775424568
Frac_6   2.775499868  2.775466463  2.775439766  2.775433664
```

CV Table of Hinge Values

```
            Lamb_5       Lamb_6       Lamb_7       Lamb_8
Frac_1   2.775592464  2.544113932  1.568381773  0.945124134
Frac_2   2.775603413  2.544291107  1.568460995  0.879432238
Frac_3   2.775603413  2.544118622  1.568452904  0.876729955
Frac_4   2.775603413  2.544327152  1.943352420  0.808804022
Frac_5   2.775603413  2.547733365  1.568399677  1.017968028
Frac_6   3.059409798  2.544215995  1.568385409  1.065652107
```

CV Table of Hinge Values

```
          Lamb_9
Frac_1       .
Frac_2       .
Frac_3       .
Frac_4       .
Frac_5       .
Frac_6       .
```

The number of selected variables obviously decreases with larger values of the penalty parameter λ:

CV Table of N Select

	Lamb_1	Lamb_2	Lamb_3	Lamb_4
Frac_1	14.00000000	14.00000000	14.00000000	12.00000000
Frac_2	14.00000000	14.00000000	14.00000000	12.00000000
Frac_3	14.00000000	14.00000000	14.00000000	12.00000000
Frac_4	14.00000000	14.00000000	14.00000000	12.00000000
Frac_5	14.00000000	14.00000000	14.00000000	12.00000000
Frac_6	12.00000000	12.00000000	12.00000000	12.00000000

CV Table of N Select

	Lamb_5	Lamb_6	Lamb_7	Lamb_8
Frac_1	12.00000000	8.000000000	6.000000000	2.000000000
Frac_2	12.00000000	8.000000000	6.000000000	2.000000000
Frac_3	12.00000000	8.000000000	6.000000000	2.000000000
Frac_4	12.00000000	8.000000000	7.000000000	2.000000000
Frac_5	12.00000000	8.000000000	6.000000000	2.000000000
Frac_6	11.00000000	8.000000000	6.000000000	2.000000000

CV Table of N Select

	Lamb_9
Frac_1	0.000000000
Frac_2	0.000000000
Frac_3	0.000000000
Frac_4	0.000000000
Frac_5	0.000000000
Frac_6	0.000000000

Approximate Covariance Matrix of Parameter Estimates

****

Dense Symmetric Matrix (14 by 14)

S	Var_01	Var_02	Var_03	Var_04	Var_05
Var_01	3.1854307				
Var_02	3.1853801	3.1856276			
Var_03	3.1960592	0.0011144	3.1971737		
Var_04	3.2314824	0.0084965	-0.0121992	3.3979041	
Var_05	-8.90e-004	5.20e-005	-0.0015376	-0.0077320	0.0071683
Var_06	1.6874422	-0.0182070	0.1329017	0.0389198	-0.0099840
Var_07	21.289719	-0.9344202	2.5648954	-0.0692622	0.0156852
Var_08	-3.4664439	0.1293400	-0.4167297	0.0082990	-0.0031447

```
Var_09 |   0.0736616  0.0602366  0.0062971 -0.0038137 -0.0065652
Var_10 |   2.0833604 -0.0576600  0.2646474  0.0041602  0.0123009
Var_11 |   0.0330421 -0.0367242  0.0061499  0.0022085  0.0026163
Var_12 |   2.5295879 -2.7892284 -0.0338416  0.1214339  1.9489605
Var_13 |   1998.6687 -739.11551  41.099446 -242.97152  857.74260
Var_14 |   0.0031013 -0.0010631  5.77e-005 -3.97e-004  0.0012721

     S |      Var_06     Var_07     Var_08     Var_09     Var_10
     ---------------------------------------------------------
Var_06 |   0.1200802
Var_07 |   0.0481650  2.0147940
Var_08 |  -0.0069359 -0.2940353  0.0525805
Var_09 |   0.0075436 -0.0760954  0.0065705  0.1801686
Var_10 |   0.0067299  0.1508623 -0.0153957 -0.0375123  2.5230264
Var_11 |  -0.0028447  0.0527966 -6.92e-005  0.0140820  0.0244104
Var_12 |   0.0921436  0.0339309 -1.0177064 -0.3845280 -0.0073448
Var_13 |   1343.6826 -4.5991250 -49.971609 -1234.9256 -149.40972
Var_14 |   0.0021410 -3.96e-006 -8.11e-005 -0.0019419 -2.28e-004

     S |      Var_11     Var_12     Var_13     Var_14
     ---------------------------------------------
Var_11 |   0.0763023
Var_12 |   0.9399163  1.9400469
Var_13 |  -81.002132  295.68711  645.54024
Var_14 |  -1.05e-004  4.25e-004  9.41e-004  1.46e-004


         Predicted Values and Residuals: Training Data
         *********************************************

               Obs       Yobs      Ypred    Residual

                 1  1.0000000  2.7523672 -1.75236721
                 2 -1.0000000 -1.0000174  1.740e-005
                 3  1.0000000 -1.0000174  2.00001740
                 4 -1.0000000  0.2507781 -1.25077810
                 5 -1.0000000  0.2507781 -1.25077810
                 6 -1.0000000 -1.0000174  1.740e-005
                 7  1.0000000  0.2507781  0.74922190
                 8  1.0000000  0.2507781  0.74922190
                 9  1.0000000  1.5015717 -0.50157172
                10  1.0000000  2.7523672 -1.75236721
                11 -1.0000000 -1.0000174  1.740e-005
                12 -1.0000000 -1.0000174  1.740e-005
                13 -1.0000000 -1.0000174  1.740e-005
                14  1.0000000  1.5015717 -0.50157172
```

```
15 -1.0000000  0.2507781 -1.25077810
16 -1.0000000 -1.0000174  1.740e-005
17  1.0000000  1.5015717 -0.50157172
18  1.0000000 -1.0000174  2.00001740
19 -1.0000000 -1.0000174  1.740e-005
20 -1.0000000 -1.0000174  1.740e-005
21  1.0000000  1.5015717 -0.50157172
22 -1.0000000 -1.0000174  1.740e-005
23 -1.0000000 -1.0000174  1.740e-005
24 -1.0000000 -1.0000174  1.740e-005
25 -1.0000000  0.2507781 -1.25077810
26 -1.0000000 -1.0000174  1.740e-005
27 -1.0000000 -1.0000174  1.740e-005
28 -1.0000000 -1.0000174  1.740e-005
29  1.0000000  0.2507781  0.74922190
30 -1.0000000  0.2507781 -1.25077810
31  1.0000000  0.2507781  0.74922190
32 -1.0000000 -1.0000174  1.740e-005
33 -1.0000000 -1.0000174  1.740e-005
34  1.0000000 -1.0000174  2.00001740
35  1.0000000 -1.0000174  2.00001740
36  1.0000000  1.5015717 -0.50157172
37  1.0000000  0.2507781  0.74922190
38  1.0000000 -1.0000174  2.00001740
39 -1.0000000 -1.0000174  1.740e-005
40 -1.0000000 -1.0000174  1.740e-005
41  1.0000000 -1.0000174  2.00001740
42 -1.0000000 -1.0000174  1.740e-005
43 -1.0000000 -1.0000174  1.740e-005
44 -1.0000000 -1.0000174  1.740e-005
45  1.0000000  0.2507781  0.74922190
46 -1.0000000 -1.0000174  1.740e-005
47  1.0000000  2.7523672 -1.75236721
48  1.0000000  0.2507781  0.74922190
49  1.0000000  2.7523672 -1.75236721
50  1.0000000  2.7523672 -1.75236721
51  1.0000000 -1.0000174  2.00001740
52 -1.0000000 -1.0000174  1.740e-005
53 -1.0000000  0.2507781 -1.25077810
54 -1.0000000  1.5015717 -2.50157172
55 -1.0000000 -1.0000174  1.740e-005
56 -1.0000000  0.2507781 -1.25077810
57  1.0000000  0.2507781  0.74922190
58 -1.0000000 -1.0000174  1.740e-005
59  1.0000000 -1.0000174  2.00001740
60  1.0000000  0.2507781  0.74922190
```

3.9 Function `smsvm`

gof = smsvm(task,trn,lamb,model,optn,class<,tst<,t0<,x0<,kfun>>>>)

<gof,theta,cmat,bvec,yprd,ftrn,yptt,ftst> = smsvm(task,trn,lamb,model,optn,...)

Purpose: Assume a training data $(\mathbf{X}, \mathbf{Y})$ set has N observations and n predictor variables $\mathbf{x}_{ik}$, $i = 1, \ldots, N, k = 1, \ldots, n$ and the categorical (Class) response Y has values $y_i, i = 1, \ldots, N$ with $nylev \geq 2$ levels. The `smsvm` function performs two support vector machine (SVM) algorithms specified by the `task` string input argument:

"msvm" MSVM: multicategory (multinomial) SVM:

The approach to SVM is described in a number of papers by Y. Lee and G. Wahba. For categorial targets with more than 2 categories, $nylev \geq 2$, this approach tries to optimize a single large QP with stacked kernel matrices of dimension $npar = (nylev-1)*N$ to obtain the optimal parameters $C[N, nylev]$ and $b[nylev]$. This approach is different from that used in function `svm`, where $nylev$ smaller QPs of dimension $npar = N$ are optimized.

For the binary classification problem and a regularization parameter $\lambda > 0$, the following minimization problem is solved for data $(\mathbf{X}, \mathbf{Y})$, $(\mathbf{x}_i, y_i), i = 1, \ldots, N$, and a specific kernel function $h \in H_K$ with $f(\mathbf{x}) = h(\mathbf{x}) + b$:

$$\frac{1}{N}\sum_{i=1}^{N}(1 - y_i f(\mathbf{x}_i))_+ + \lambda\|h\|^2_{H_K}$$

where $(x)_+ = \max(x, 0)$ and $\|h\|^2_{H_K}$ is the square norm of function h in the reproducing Hilbert space with reproducing kernel function $K(.,.)$. For linear kernel $h(x) = \mathbf{w} \cdot \mathbf{x}$ there is $\|h\|^2_{H_K} = \|\mathbf{w}\|^2$. For the binary response problem the $N + 1$ variables of the optimization problem are the N weight values $\mathbf{w}$ and the additive constant b. For the general multinomial response problem, $\mathbf{w} \in \mathcal{R}^{\setminus\dagger\updownarrow\rceil\sqsubseteq,\mathcal{N}}$ and $b \in \mathcal{R}^{\setminus\dagger\updownarrow\rceil\sqsubseteq}$. Here, the regularization parameter λ corresponds to the regularization value C used in the `svm` function.

"ssvm","csvm" sequential and combined SMSVM: structured (multicategory) SVM:

Here the MSVM approach is extended to the problem of variable selection by imposing an additional $L1$ penalty term like in Lasso. For that the kernel formulation is expressed in terms of the n predictor variables X_ν with weights $Theta_\nu$

$$K(\mathbf{x}_i, \mathbf{x}_j) = \sum_{\nu=1}^{n} \Theta_\nu K_\nu(\mathbf{x}_i, \mathbf{x}_j), \quad i, j, = 1, \ldots, N.$$

For ANOVA Kernels in addition to the n variable terms also all $n(n-1)/2$ interactions $\mathbf{X}_k \cdot \mathbf{X}_l$, $k, l = 1 \ldots, n$, can be added and the Θ vector of weights now becomes $d = n(n+1)/2$ number of entries Θ_ν .

Including the penalty term and the additional d weights Θ_ν into the model, the binary classification problem now becomes

$$\frac{1}{N}\sum_{i=1}^{N}(1 - y_i f(\mathbf{x}_i))_+ + \lambda \sum_{\nu=1}^{d} \Theta_\nu^{-1} \|h\|_{H_K}^2 + \lambda_\Theta \sum_{\nu=1}^{d} \Theta_\nu$$

subject to $\Theta_\nu \in [0, 1]$. For $\Theta_\nu \equiv 1$ the SMSVM model reduces to the MSVM model.

For a specified number of hyper parameters λ and λ_Θ a grid search is performed and the "best" solution is selected. For evaluation, k-fold cross validation or test validation can be used.

For SSVM two versions of the iterative algorithm for estimating the Θ values are implemented:

1. iteration with sequential estimation:
2. iteration with combined estimation:

Dimension d of Θ for kernel functions:

Kernel	d	nkp
spli(ne-)t1	n	0
spli(ne-)t2	n*(n+1)/2	0
splin(e-)1	2n	0
splin(e-2)2	2n+4*n*(n-1)	0
line(ar)	n	0
poly(nomial)	n	1
rbf	n	1

Input: The first six input arguments must be specified, the remaining arguments are optional.

task must be a string specifying the estimation task. The specified string value must be one of the following

`"msvm"` for common Multicategory SVM,

`"ssvm"` for sequentially structured MSVM,

`"ssvm"` for combined structured MSVM .

trn The training data set with N rows containing the response $\mathbf{Y}$ and the n predictor variable $\mathbf{X}$. The columns are selected by the `model` specification.

lamb For MSVM this should be a scalar or vector of integers, which are used as the exponents of 2 for the regularization value. For SMSVM this should be a 2-row matrix, where the first row contains a set of λ values and the second row contains a set of λ_Θ values. This is important: for different number of λ and λ_Θ values the unused entries in the 2-row matrix should be missing and not zero, since zero is a valid input.

model : The analysis model is specified in form of a string, e.g. `model= "3=1 2"`, containing column numbers for variables. The syntax of the `model` string argument is the same as for the `glmod()` function except for the additional *events / trial* response specification. `????`

optn This argument must be specified in form of a two column matrix where the first column defines the option as string value (in quotes) and the second column can be used for a numeric or string specification of the option. See table below for content.

class specifies which of the columns of the input data matrix $\mathbf{X}$ are nominally scaled CLASS variables. For this function. at least the response variable is nominal and must be specified as CLASS variable.

tst The test data set with Nt rows containing the response $\mathbf{Y}$ and the n predictor variable $\mathbf{X}$. The columns must be comaptible with those of the training set and the `model` and `class` specifications.

t0 Must be either missing or a d vector of values in $[0, 1]$. For MSVM, specifies a Θ vector used in weighting the d kernel matrices. For SSVM, specifies a Θ vector as an initial solution for the iteration process.

x0 If not missing this must be spefied either as a $(nylev - 1) * N$ vector or a $(nylev - 1) \times N$ matrix of starting values for the QP problem. If specified, the values should be nonnegative and satisfy the $nylev - 1$ linear equality contraints.

kfun optional string argument specifying a user defined function for kernel evaluation. The function definition must be of the form: `function kernel(vi,vj)` where the input arguments `vi` and `vj` are observation vectors of the data and the function must be a real scalar. Note, that Mercer's condition must be satisfied by the functions kernel definition, i.e. the resulting matrices $\mathbf{Q}$ of the QP must be positive (semi)definite.

Content of the Options Matrix :

Option	Second Column	Meaning
"block"		block cross validation
"cloo"		Loo validation
`"epsi"`	real	
`"fitcr"`	string	fit criterion
	"class"	misclassification rate
	"hinge"	hinge criterion
`"fold"`	int	fold number for cross validation
`"ftol"`	real	function termination criterion
`"gtol"`	real	gradient termination criterion
`"kern"`	string	link function, default is linear
	"line"	linear function
	"poly"	polynomial function
	"rbf"	Gaussian radial basis function (def. scaling $[0, 1]$)
`"kfp1"`	real	first parameter of kernel function
`"kfp2"`	real	second parameter of kernel function
`"kfp3"`	real	third parameter of kernel function
`"lphis"`	int	print history of LP iterations (def=)
`"lpmaxi"`	int	maximum number of LP iterations (def=)
`"nobstat"`		print observation statistics
`"nomis"`		do not use observations that contain missing values in one of the variables used in analysis
`"nopr"`		no printed output in ...txt (but warnings or errors are in ...log)
`"noscal"`		no scaling of predictor variables
`"ntiter"`	int	maximum number Theta iterations

Option	Second Column	Meaning
"order"		order categories of class variables
	"dat"	in order as they appear in the rows of data matrix
	"asc"	sorted in ascending order
	"des"	sorted in descending order
"pall"		is equivalent to "print" 10
"pinit"		print input information and model setup
"plamb"		print table of fit values for different lambdas
"pthet"		print table of theta values in history
"qpnusp"		null space QP technique
"qprasp"		range space QP technique
"qphis"	int	print history of QP iterations (def=)
"qpmaxi"	int	maximum number of QP iterations (def=)
"random"	int	fold number for random cross validation
"split"	int	fold number for split cross validation
"start"	string or int	method for obtaining starting values for QP
	"zero"	set to zero, same as 0
	"one"	set to one, same as 1
	"rand"	uniformly random, same as 2
"scale"	int	if not zero, perform scaling of predictor variables; this is default for several kernel functions
"seed"	int	seed for random generator; default is time of day; affects the results of random cross validation: affects results of RSVM and random starting values
"sing"	real	singularity criterion, default is 1.e-8
"trm2"	real	L2 termination criterion for Theta iteration
"trm1"	real	L1 termination criterion for Theta iteration
"trmi"	real	Loo termination criterion for Theta iteration
"tolop"	real	termination criterion for LP by PCx
"tolpr"	real	termination criterion for LP by PCx
"toldu"	real	termination criterion for LP by PCx
"tech"	string	specifies optimization for FQP and DQP
	"qpnusp"	null space QP technique
	"qprasp"	range space QP technique

Output: gof returns a vector of scalar results with the following entries

1. integer indicating some kind of failure when nonzero
2. computation time in seconds
3. best value of λ in terms of exponent of two
4. only for MSVM: best value of fit criterion
5. only for SMSVM: best value of λ_Θ in terms of exponent of two
6. only for SMSVM: number of selected variables, i.e. nonzero values in Θ
7. only for SMSVM: shrinking value: $\frac{1}{d}\sum_{j=1}^{d}\Theta_j$

8. only for SMSVM: L2 change in successive Θ estimates during iteration of $(\Theta, \mathbf{C}, \mathbf{b})$
9. only for SMSVM: L2 change in successive $\mathbf{C}$ estimates during iteration of $(\Theta, \mathbf{C}, \mathbf{b})$
10. only for SMSVM: L2 change in successive $\mathbf{b}$ estimates during iteration of $(\Theta, \mathbf{C}, \mathbf{b})$
11. number of Θ iterations executed

theta returns the estimates of the d vector Θ; for MSVM, these is either the `t0` input argument or a vector of ones;

cmat returns the $nylev \times N$ matrix $\mathbf{C}$ of estimated parameters for scoring

bvec returns the $nylev$ vector $\mathbf{b}$ of estimated constant intercepts for the model

yprd returns the N vector of predicted values $\hat{y}_i$ for the training data

ftrn returns the $nylev \times N$ matrix of predicted fit values for the training data

yptt returns the Nt vector of predicted values $\hat{y}_i$ for the test data

ftst returns the $nylev \times Nt$ matrix of predicted fit values for the test data

Restrictions: 1. Currently no missing values are permitted in `trn` training and `tst` test input matrix.

2. The dimensions of some input parameters must be compatible.

Relationships: svm(), lars(), qp(), lp()

Examples: 1. MSVM Example by Lee, Kim, Lee & Koo (2005): Simple MSVM Training

```
x = [ 0.528872284 0.569614776, 0.700185976 0.321923548, 0.462571064 0.470884898,
      0.513362291 0.806581861, 0.949215549 0.362976732, 0.264136137 0.154079203,
      0.604948994 0.951229039, 0.064013089 0.012931984, 0.749241452 0.996598753,
      0.494394564 0.533025743, 0.159963265 0.238321322, 0.778785072 0.974283966,
      0.464785946 0.845194487, 0.485296134 0.331759901, 0.316437662 0.956105723,
      0.159774920 0.735288704, 0.115898914 0.002054487, 0.214555487 0.265701185,
      0.843991904 0.013981205, 0.479514559 0.275019907, 0.079768412 0.671501640,
      0.939785768 0.181861395, 0.972153088 0.816082471, 0.303241238 0.510639912,
      0.938063862 0.421412485, 0.966934781 0.523864536, 0.620233059 0.246350058,
      0.181040596 0.517738702, 0.475137423 0.301242052, 0.206606589 0.530617075,
      0.323157200 0.993319286, 0.298947780 0.173359120, 0.691527047 0.549043990,
      0.989061426 0.738280592, 0.716703556 0.130655373, 0.838137786 0.535779711,
      0.850532592 0.131447349, 0.979186682 0.514502136, 0.282877193 0.742405009,
      0.004088614 0.013303000, 0.043562030 0.962760791, 0.043928142 0.637958958,
      0.009877956 0.115461618, 0.267268998 0.481439149, 0.355037130 0.406802560,
      0.382213150 0.001416346, 0.312186666 0.115521625, 0.109576124 0.343046696,
      0.790772599 0.364299313, 0.485656525 0.242482759, 0.531536482 0.386606219,
```

```
      0.407007085 0.970674523, 0.811286981 0.488186524, 0.643866451 0.141177355,
      0.568715947 0.738062642, 0.118313203 0.750450021, 0.967430168 0.464891446,
      0.262222449 0.735098338, 0.935817645 0.993227892, 0.780918555 0.507945463 ]
nr = nrow(x); nc = ncol(x); print "nr,nc=",nr,nc;



y = [ 2 2 2 2 3 2 1 2 2 2 1 3 1 3 2 1 1 2 3 2
      1 3 3 2 3 3 2 3 2 1 1 2 3 3 3 2 2 3 1 1
      1 2 1 2 2 2 1 1 3 1 3 2 2 2 3 2 3 1 3 3 ];
trn = x -> y';


modl = "3 = 1 2"; clas = 3;
lamb = [ -16 : -8 ]; print "lambda=",lamb;


lambda=
   |    1    2    3    4    5    6    7    8    9
-------------------------------------------------
 1 |  -16  -15  -14  -13  -12  -11  -10   -9   -8


optn = [ "print"          3 ,
         "noscal"           ,
         "kern"     "split1" ,
         "kfp1"          1. ,
         "tech"       "qpra" ,
         "start"          2 ,
         "seed"           5 ];
< gof,theta,cmat,bvec,yprd,ftrn > = smsvm("msvm",trn,lamb,modl,optn,clas);


                      *****************
                      Model Information
                      *****************

                Number Valid Observations   60
                Response Variable          Y[3]
                N Independend Variables      2
                MSVM
                QP Start: Random
                Kernel Function     Spline-t 1
                Use Unscaled Predictor


                       *************
```

Model Effects
****

X1 + X2

****
Class Level Information
****

Class	Level	Value
Y[3]	3	1 2 3

****
Simple Statistics
****

Column	Nobs	Mean	Std Dev	Skewness	Kurtosis
X[1]	60	0.4972387	0.3051313	0.1141473	-1.2177564
X[2]	60	0.4818079	0.2977871	0.1796648	-0.9598966

MSVM: Training HINGE Loss for Set of Lambdas
****

Ind	Log2L	Lambda	ClassEr	HingeEr	QP	LP
1	-16	1.526e-005	0.31666667	0.53426960	4	0
2	-15	3.052e-005	0.35000000	0.54486737	4	0
3	-14	6.104e-005	0.33333333	0.55366514	4	0
4	-13	1.221e-004	0.33333333	0.56084147	4	0
5	-12	2.441e-004	0.33333333	0.56561755	4	0
6	-11	4.883e-004	0.33333333	0.57856445	4	0
7	-10	9.766e-004	0.35000000	0.60644669	4	0
8	-9	0.00195313	0.40000000	0.65298646	4	0
9	-8	0.00390625	0.43333333	0.69792435	4	0

The optimal lambda.theta on log2 scale: -16

Model Intercepts b for 3 Ylevels
****

Dense Column Vector (nrow=3)

C	Ylev_1	Ylev_2	Ylev_3

-0.1271586 0.1708907 -0.0437321

Model Weights C for 3 Ylevels
****

Dense Matrix (3 by 60)

	1	2	3	4	5
Ylev_1	-364.08889	-364.08889	0.0000000	-364.08889	0.0000000
Ylev_2	728.17778	728.17778	0.0000000	728.17778	0.0000000
Ylev_3	-364.08889	-364.08889	0.0000000	-364.08889	0.0000000

	6	7	8	9	10
Ylev_1	-573.29253	728.17778	-520.56471	37.385694	257.75154
Ylev_2	518.97414	-364.08889	571.70196	527.44049	257.75154
Ylev_3	54.318394	-364.08889	-51.137256	-564.82618	-515.50308

	11	12	13	14	15
Ylev_1	436.35170	-364.08889	484.85231	57.257762	-438.14433
Ylev_2	-655.91497	-364.08889	-607.41435	-574.76221	654.12234
Ylev_3	219.56327	728.17778	122.56204	517.50445	-215.97801

	16	17	18	19	20
Ylev_1	710.07596	728.17778	-501.52538	-291.08367	0.0000000
Ylev_2	-327.88526	-364.08889	590.74128	-400.59150	0.0000000
Ylev_3	-382.19070	-364.08889	-89.215898	691.67517	0.0000000

	21	22	23	24	25
Ylev_1	29.903629	66.615092	-295.85685	-728.17778	0.0000000
Ylev_2	29.903629	-133.23018	147.92843	364.08889	0.0000000
Ylev_3	-59.807258	66.615092	147.92843	364.08889	0.0000000

	26	27	28	29	30
Ylev_1	0.0000000	-198.13983	-364.08889	176.89793	601.48588
Ylev_2	0.0000000	645.20325	-364.08889	176.89793	-490.78079
Ylev_3	0.0000000	-447.06342	728.17778	-353.79587	-110.70509

| | 31 | 32 | 33 | 34 | 35 |

```
       |          1          2          3          4          5
-------------------------------------------------------------------
Ylev_1 |   497.32622 -724.33948 -130.56923  1.26e-015  32.188785
Ylev_2 |  -594.94045  367.92719 -480.84872 -2.53e-015 -562.22773
Ylev_3 |   97.614229  356.41230  611.41795  1.26e-015  530.03894

       |          6          7          8          9         10
-------------------------------------------------------------------
Ylev_1 |   364.08889  364.08889 -6.32e-016  695.26747  114.84253
Ylev_2 |   364.08889  364.08889  3.16e-016 -396.99920  114.84253
Ylev_3 |  -728.17778 -728.17778  3.16e-016 -298.26826 -229.68506


       |         11         12         13         14         15
-------------------------------------------------------------------
Ylev_1 |   15.480417 -728.17778  149.07212 -570.60708 -4.5139544
Ylev_2 |   15.480417  364.08889 -298.14424  521.65958  523.12025
Ylev_3 |  -30.960833  364.08889  149.07212  48.947497 -518.60630

       |         16         17         18         19         20
-------------------------------------------------------------------
Ylev_1 |  -200.88843  454.51811  728.17778 -364.08889  425.87602
Ylev_2 |   203.78459 -637.74856 -364.08889 -364.08889 -666.39064
Ylev_3 |  -2.8961634  183.23045 -364.08889  728.17778  240.51462


       |         21         22         23         24         25
-------------------------------------------------------------------
Ylev_1 |   207.89926 -673.90863  364.08889  24.838507 -364.08889
Ylev_2 |  -650.08297  336.95432  364.08889  533.71408 -364.08889
Ylev_3 |   442.18370  336.95432 -728.17778 -558.55259  728.17778

       |         26         27         28         29         30
-------------------------------------------------------------------
Ylev_1 |  -716.54172  0.0000000  728.17778  0.0000000  364.08889
Ylev_2 |   375.72494  0.0000000 -364.08889  0.0000000 -728.17778
Ylev_3 |   340.81678  0.0000000 -364.08889  0.0000000  364.08889


         Predicted Values and Residuals: Tarining Data
         *********************************************

              Obs       Yobs      Ypred    Residual

                1  2.0000000  2.0000000  0.00000000
                2  2.0000000  2.0000000  0.00000000
                3  2.0000000  2.0000000  0.00000000
                4  2.0000000  2.0000000  0.00000000
```

5	3.0000000	3.0000000	0.00000000
6	2.0000000	2.0000000	0.00000000
7	1.0000000	2.0000000	-1.00000000
8	2.0000000	1.0000000	1.00000000
9	2.0000000	3.0000000	-1.00000000
10	2.0000000	2.0000000	0.00000000
11	1.0000000	1.0000000	0.00000000
12	3.0000000	3.0000000	0.00000000
13	1.0000000	2.0000000	-1.00000000
14	3.0000000	2.0000000	1.00000000
15	2.0000000	2.0000000	0.00000000
16	1.0000000	1.0000000	0.00000000
17	1.0000000	1.0000000	0.00000000
18	2.0000000	2.0000000	0.00000000
19	3.0000000	3.0000000	0.00000000
20	2.0000000	2.0000000	0.00000000
21	1.0000000	1.0000000	0.00000000
22	3.0000000	3.0000000	0.00000000
23	3.0000000	3.0000000	0.00000000
24	2.0000000	2.0000000	0.00000000
25	3.0000000	3.0000000	0.00000000
26	3.0000000	3.0000000	0.00000000
27	2.0000000	2.0000000	0.00000000
28	3.0000000	1.0000000	2.00000000
29	2.0000000	2.0000000	0.00000000
30	1.0000000	1.0000000	0.00000000
31	1.0000000	2.0000000	-1.00000000
32	2.0000000	2.0000000	0.00000000
33	3.0000000	2.0000000	1.00000000
34	3.0000000	3.0000000	0.00000000
35	3.0000000	2.0000000	1.00000000
36	2.0000000	3.0000000	-1.00000000
37	2.0000000	3.0000000	-1.00000000
38	3.0000000	3.0000000	0.00000000
39	1.0000000	2.0000000	-1.00000000
40	1.0000000	1.0000000	0.00000000

41	1.0000000	1.0000000	0.00000000
42	2.0000000	1.0000000	1.00000000
43	1.0000000	1.0000000	0.00000000
44	2.0000000	2.0000000	0.00000000
45	2.0000000	2.0000000	0.00000000
46	2.0000000	2.0000000	0.00000000
47	1.0000000	2.0000000	-1.00000000
48	1.0000000	1.0000000	0.00000000

```
                  49  3.0000000  3.0000000  0.00000000
                  50  1.0000000  2.0000000 -1.00000000
                  51  3.0000000  2.0000000  1.00000000
                  52  2.0000000  2.0000000  0.00000000
                  53  2.0000000  3.0000000 -1.00000000
                  54  2.0000000  2.0000000  0.00000000
                  55  3.0000000  2.0000000  1.00000000
                  56  2.0000000  1.0000000  1.00000000
                  57  3.0000000  3.0000000  0.00000000
                  58  1.0000000  1.0000000  0.00000000
                  59  3.0000000  3.0000000  0.00000000
                  60  3.0000000  3.0000000  0.00000000


                  Total Number of Kernel Calls: 1
                     Time for Optimization: 1
                     Total Processing Time: 1
```

```
  print "GOF=", gof;
  print "Theta=",theta;
  print "Cmat=",cmat;
  print "bvec=",bvec;
  print "Yprd=",yprd;
```

```
GOF=
            |  g-of-fit
------------------------
    Failure |     0.0000
   CompTime |     1.0000
  OptLambda |   -16.0000
    OptCrit |     0.5343
     unused |      .
     unused |      .
     unused |      .
     unused |      .
     unused |      .
     unused |      .
     unused |      .
     unused |      .
```

```
Theta=
   |     Dim_1     Dim_2
------------------------
 1 |   1.00000   1.00000
```

```
bvec=
         |         1
---------------------
  Ylev_1 |  -0.12716
  Ylev_2 |   0.17089
  Ylev_3 |  -0.04373
```

```
Yprd=
   |         1
--------------
 1 |   2.00000
 2 |   2.00000
 3 |   2.00000
 4 |   2.00000
 5 |   3.00000
 6 |   2.00000
 7 |   2.00000
 8 |   1.00000
 9 |   3.00000
10 |   2.00000
11 |   1.00000
12 |   3.00000
13 |   2.00000
14 |   2.00000
15 |   2.00000
16 |   1.00000
17 |   1.00000
18 |   2.00000
19 |   3.00000
20 |   2.00000
21 |   1.00000
22 |   3.00000
23 |   3.00000
24 |   2.00000
25 |   3.00000
26 |   3.00000
27 |   2.00000
28 |   1.00000
29 |   2.00000
30 |   1.00000
```

```
   |         1
--------------
31 |   2.00000
32 |   2.00000
33 |   2.00000
34 |   3.00000
35 |   2.00000
36 |   3.00000
37 |   3.00000
38 |   3.00000
39 |   2.00000
40 |   1.00000
41 |   1.00000
42 |   1.00000
43 |   1.00000
44 |   2.00000
45 |   2.00000
46 |   2.00000
47 |   2.00000
48 |   1.00000
49 |   3.00000
50 |   2.00000
51 |   2.00000
52 |   2.00000
53 |   3.00000
54 |   2.00000
55 |   2.00000
56 |   1.00000
57 |   3.00000
58 |   1.00000
59 |   3.00000
60 |   3.00000
```

2. MSVM Example by Lee, Kim, Lee & Koo (2005): Training with Test Validation

```
xtst = [ 0.916099340 0.82352774, 0.265586914 0.20559270, 0.096223712 0.17834181,
         0.338829069 0.48444123, 0.927028025 0.23831105, 0.809974386 0.54002736,
```

```
        0.915073710 0.50952501, 0.142475475 0.48073932, 0.396088460 0.90315165,
        0.521394802 0.27580360, 0.898900394 0.24095776, 0.951605172 0.01104448,
        0.075719937 0.77528725, 0.423623245 0.20548662, 0.859307121 0.34653661,
        0.810216786 0.67709408, 0.591530875 0.77640365, 0.246214617 0.54793864,
        0.001322537 0.60670532, 0.582305357 0.86138444, 0.501910063 0.24775436,
        0.535084397 0.91941108, 0.547103156 0.43485872, 0.673724107 0.70238004,
        0.508723245 0.46773255, 0.905968030 0.14092784, 0.483178575 0.88288055,
        0.426759451 0.22901014, 0.343323067 0.33785071, 0.805856677 0.09076315 ];

ytst = [ 3 1 1 2 3 3 3 2 2 3 3 3 1 1 3 2 2 2 1 1
         2 3 3 3 2 3 2 2 1 3 ];
 tst = xtst -> ytst';



 print "MSVM with Test Set validation: Nullspace: SPLIT1";
 modl = "3 = 1 2"; clas = 3;
 lamb = [ -16 : -8 ]; print "lambda=",lamb;
 optn = [ "print"           3 ,
          "noscal"            ,
          "kern"     "split1" ,
          "kfp1"           1. ,
          "tech"       "qpnu" ,
          "start"          2 ,
          "seed"         123 ];
 < gof,theta,cmat,bvec,yprd,ftrn,yptt,ftst > =
        smsvm("msvm",trn,lamb,modl,optn,clas,tst);

          MSVM: Validation HINGE Loss for Set of Lambdas
          **********************************************

       Ind   Log2L      Lambda     ClassEr     HingeEr QP LP
         1     -16  1.526e-005  0.33333333  0.94258011  6  0
         2     -15  3.052e-005  0.33333333  0.91580370  6  0
         3     -14  6.104e-005  0.33333333  0.90059494  6  0
         4     -13  1.221e-004  0.33333333  0.88784828  6  0
         5     -12  2.441e-004  0.33333333  0.88468780  6  0
         6     -11  4.883e-004  0.33333333  0.89239390  6  0
         7     -10  9.766e-004  0.33333333  0.93518363  6  0
         8      -9  0.00195313  0.36666667  0.92468131  6  0
         9      -8  0.00390625  0.36666667  0.92346110  6  0

           The optimal lambda.theta on log2 scale: -12


                 Total Number of Kernel Calls: 2
```

```
Time for Optimization: 2
Total Processing Time: 2
```

Model Intercepts b for 3 Ylevels

Dense Column Vector (nrow=3)

C	Ylev_1	Ylev_2	Ylev_3
	-0.1752226	0.2948687	-0.1196461

Model Weights C for 3 Ylevels

Dense Matrix (3 by 60)

	1	2	3	4	5
Ylev_1	8.7872597	22.755556	-26.936676	-22.755556	0.0000000
Ylev_2	29.739703	22.755556	41.329991	45.511111	0.0000000
Ylev_3	-38.526963	-45.511111	-14.393315	-22.755556	0.0000000

	6	7	8	9	10
Ylev_1	-45.511111	45.511111	-22.755556	-22.755556	22.755556
Ylev_2	22.755556	-22.755556	45.511111	45.511111	22.755556
Ylev_3	22.755556	-22.755556	-22.755556	-22.755556	-45.511111

	11	12	13	14	15
Ylev_1	22.755556	-22.755556	34.400649	-7.9468478	-32.847044
Ylev_2	-45.511111	-22.755556	-33.866017	-30.159909	35.419622
Ylev_3	22.755556	45.511111	-0.5346319	38.106757	-2.5725779

	16	17	18	19	20
Ylev_1	45.511111	45.511111	-45.511111	21.120399	-22.755556
Ylev_2	-22.755556	-22.755556	22.755556	-44.693533	45.511111
Ylev_3	-22.755556	-22.755556	22.755556	23.573134	-22.755556

	21	22	23	24	25
Ylev_1	22.755556	7.8534669	-45.511111	-45.511111	22.755556

```
Ylev_2 |  -45.511111 -15.706934  22.755556  22.755556 -45.511111
Ylev_3 |   22.755556  7.8534669  22.755556  22.755556  22.755556

       |          26         27         28         29         30
-------------------------------------------------------------------
Ylev_1 |   0.0000000  22.755556 -22.755556 -24.669196  45.511111
Ylev_2 |   0.0000000  22.755556 -22.755556  43.597471 -22.755556
Ylev_3 |   0.0000000 -45.511111  45.511111 -18.928276 -22.755556

       |          31         32         33         34         35
-------------------------------------------------------------------
Ylev_1 |   39.458700 -45.511111 -22.755556 -1.8105758 -17.986610
Ylev_2 |  -28.807967  22.755556 -22.755556  0.9052879 -25.140028
Ylev_3 |  -10.650733  22.755556  45.511111  0.9052879  43.126638

       |          36         37         38         39         40
-------------------------------------------------------------------
Ylev_1 |  -11.245914  22.755556  0.0000000  34.750467  0.8444261
Ylev_2 |   39.756290  22.755556  0.0000000 -33.516200  0.8444261
Ylev_3 |  -28.510377 -45.511111  0.0000000 -1.2342665 -1.6888522

       |          41         42         43         44         45
-------------------------------------------------------------------
Ylev_1 |   39.963475 -45.511111  0.0000000 -22.755556 -45.511111
Ylev_2 |  -28.303192  22.755556  0.0000000  45.511111  22.755556
Ylev_3 |  -11.660283  22.755556  0.0000000 -22.755556  22.755556

       |          46         47         48         49         50
-------------------------------------------------------------------
Ylev_1 |  -45.511111  22.755556  35.158604  17.660477  45.511111
Ylev_2 |   22.755556 -45.511111 -33.108063 -42.963572 -22.755556
Ylev_3 |   22.755556  22.755556 -2.0505415  25.303095 -22.755556

       |          51         52         53         54         55
-------------------------------------------------------------------
Ylev_1 |   22.755556 -45.511111  22.755556  22.755556 -22.755556
Ylev_2 |  -45.511111  22.755556  22.755556  22.755556 -22.755556
Ylev_3 |   22.755556  22.755556 -45.511111 -45.511111  45.511111

       |          56         57         58         59         60
-------------------------------------------------------------------
Ylev_1 |  -24.003272  7.58e-015  45.511111 -7.6806618  6.1510941
Ylev_2 |   44.263395 -1.52e-014 -22.755556  3.8403309 -37.208880
Ylev_3 |  -20.260124  7.58e-015 -22.755556  3.8403309  31.057786
```

Predicted Values and Residuals: Tarining Data

****

Obs	Yobs	Ypred	Residual
1	2.0000000	2.0000000	0.00000000
2	2.0000000	2.0000000	0.00000000
3	2.0000000	2.0000000	0.00000000
4	2.0000000	2.0000000	0.00000000
5	3.0000000	3.0000000	0.00000000
6	2.0000000	2.0000000	0.00000000
7	1.0000000	2.0000000	-1.00000000
8	2.0000000	1.0000000	1.00000000
9	2.0000000	2.0000000	0.00000000
10	2.0000000	2.0000000	0.00000000
11	1.0000000	1.0000000	0.00000000
12	3.0000000	3.0000000	0.00000000
13	1.0000000	2.0000000	-1.00000000
14	3.0000000	2.0000000	1.00000000
15	2.0000000	2.0000000	0.00000000
16	1.0000000	1.0000000	0.00000000
17	1.0000000	1.0000000	0.00000000
18	2.0000000	2.0000000	0.00000000
19	3.0000000	3.0000000	0.00000000
20	2.0000000	2.0000000	0.00000000
21	1.0000000	1.0000000	0.00000000
22	3.0000000	3.0000000	0.00000000
23	3.0000000	3.0000000	0.00000000
24	2.0000000	2.0000000	0.00000000
25	3.0000000	3.0000000	0.00000000
26	3.0000000	3.0000000	0.00000000
27	2.0000000	2.0000000	0.00000000
28	3.0000000	1.0000000	2.00000000
29	2.0000000	2.0000000	0.00000000
30	1.0000000	2.0000000	-1.00000000
31	1.0000000	2.0000000	-1.00000000
32	2.0000000	2.0000000	0.00000000
33	3.0000000	2.0000000	1.00000000
34	3.0000000	3.0000000	0.00000000
35	3.0000000	2.0000000	1.00000000
36	2.0000000	3.0000000	-1.00000000
37	2.0000000	3.0000000	-1.00000000
38	3.0000000	3.0000000	0.00000000
39	1.0000000	2.0000000	-1.00000000

40	1.0000000	1.0000000	0.00000000

41	1.0000000	1.0000000	0.00000000
42	2.0000000	1.0000000	1.00000000
43	1.0000000	1.0000000	0.00000000
44	2.0000000	2.0000000	0.00000000
45	2.0000000	2.0000000	0.00000000
46	2.0000000	2.0000000	0.00000000
47	1.0000000	2.0000000	-1.00000000
48	1.0000000	1.0000000	0.00000000
49	3.0000000	3.0000000	0.00000000
50	1.0000000	2.0000000	-1.00000000
51	3.0000000	2.0000000	1.00000000
52	2.0000000	2.0000000	0.00000000
53	2.0000000	3.0000000	-1.00000000
54	2.0000000	2.0000000	0.00000000
55	3.0000000	2.0000000	1.00000000
56	2.0000000	1.0000000	1.00000000
57	3.0000000	3.0000000	0.00000000
58	1.0000000	2.0000000	-1.00000000
59	3.0000000	3.0000000	0.00000000
60	3.0000000	3.0000000	0.00000000

Predicted Values and Residuals: Test Data

****

Obs	Yobs	Ypred	Residual
1	3.0000000	3.0000000	0.00000000
2	1.0000000	2.0000000	-1.00000000
3	1.0000000	2.0000000	-1.00000000
4	2.0000000	2.0000000	0.00000000
5	3.0000000	3.0000000	0.00000000
6	3.0000000	3.0000000	0.00000000
7	3.0000000	3.0000000	0.00000000
8	2.0000000	2.0000000	0.00000000
9	2.0000000	2.0000000	0.00000000
10	3.0000000	2.0000000	1.00000000
11	3.0000000	3.0000000	0.00000000
12	3.0000000	3.0000000	0.00000000
13	1.0000000	2.0000000	-1.00000000
14	1.0000000	2.0000000	-1.00000000
15	3.0000000	3.0000000	0.00000000
16	2.0000000	3.0000000	-1.00000000
17	2.0000000	2.0000000	0.00000000

```
 18  2.0000000  2.0000000  0.00000000
 19  1.0000000  2.0000000 -1.00000000
 20  1.0000000  2.0000000 -1.00000000
 21  2.0000000  2.0000000  0.00000000
 22  3.0000000  2.0000000  1.00000000
 23  3.0000000  2.0000000  1.00000000
 24  3.0000000  3.0000000  0.00000000
 25  2.0000000  2.0000000  0.00000000
 26  3.0000000  3.0000000  0.00000000
 27  2.0000000  2.0000000  0.00000000
 28  2.0000000  2.0000000  0.00000000
 29  1.0000000  2.0000000 -1.00000000
 30  3.0000000  3.0000000  0.00000000
```

3. SMSVM Example by Lee, Kim, Lee & Koo (2005)

```
modl = "3 = 1 2"; clas = 3;
lamb = [ -16 : -8 . . ,
         -10 : 0     ];  print "lambda=",lamb;
```

```
lambda=
   |         1         2         3         4         5         6
---------------------------------------------------------------------
 1 |  -16.0000  -15.0000  -14.0000  -13.0000  -12.0000  -11.0000
 2 |  -10.0000   -9.0000   -8.0000   -7.0000   -6.0000   -5.0000

   |         7         8         9        10        11
-----------------------------------------------------
 1 |  -10.0000   -9.0000   -8.0000     .         .
 2 |   -4.0000   -3.0000   -2.0000   -1.0000    0.0000
```

```
optn = [ "print"           3 ,
         "noscal"            ,
         "ntiter"          1 ,
         "kern"     "split1" ,
         "kfp1"           1. ,
         "tech"       "qpnu" ,
         "start"           2 ,
         "seed"            5 ];
< gof,theta,cmat,bvec,yprd,ftrn > = smsvm("ssvm",trn,lamb,modl,optn,clas);
```

```
                         *****************
                         Model Information
```

```
*****************

Number Valid Observations   60
Response Variable          Y[3]
N Independend Variables      2
SMSVM sequential
QP Start: Random
Kernel Function     Spline-t 1
Use Unscaled Predictor
```

Some printed output, similar to that of the first example, is skipped for space reasons. The following shows the history of the initial Cstep, where for given $\boldsymbol{\Theta}$, the model matrix $\mathbf{C}$ and the vector $\mathbf{b}$ are estimated:

```
Cstep: Training HINGE Loss for Set of Lambdas
*********************************************
```

Ind	Log2L	Lambda	ClassEr	HingeEr	QP	LP
1	-16	1.526e-005	0.31666667	0.53426933	6	0
2	-15	3.052e-005	0.35000000	0.54486737	6	0
3	-14	6.104e-005	0.33333333	0.55366514	6	0
4	-13	1.221e-004	0.33333333	0.56084147	6	0
5	-12	2.441e-004	0.33333333	0.56561755	6	0
6	-11	4.883e-004	0.33333333	0.57856445	6	0
7	-10	9.766e-004	0.35000000	0.60644669	6	0
8	-9	0.00195313	0.40000000	0.65298646	6	0
9	-8	0.00390625	0.43333333	0.69792435	6	0

```
The optimal lambda.theta on log2 scale: -16
```

The iteration start with the so-called Theta-step where for given model matrix $\mathbf{C}$ and vector $\mathbf{b}$ the $\boldsymbol{\Theta}$ vector ist estimated:

```
Tstep: Training HINGE Loss for Set of Lambdas
*********************************************
```

Ind	Log2L	Lambda	ClassEr	HingeEr	QP	LP
1	-10	9.766e-004	0.31666667	0.53426933	6	0
2	-9	0.00195313	0.31666667	0.53426933	6	0
3	-8	0.00390625	0.31666667	0.53426933	6	0
4	-7	0.00781250	0.31666667	0.53426933	6	0
5	-6	0.01562500	0.33333333	0.53617580	6	0
6	-5	0.03125000	0.35000000	0.54516738	6	0
7	-4	0.06250000	0.33333333	0.56660218	6	0

8	-3	0.12500000	0.33333333	0.56726872	6	0
9	-2	0.25000000	0.33333333	0.59859459	6	0
10	-1	0.50000000	0.58333333	0.97655451	6	0
11	0	1.00000000	0.58333333	0.97655451	6	0

The optimal lambda.theta on log2 scale: -10

The following ar the Theta vectors for the specified Λ_Θ values:

Table of Theta Values for all LambdaTheta

	Lambda	1	2
1	9.8e-004	1.00000000	1.00000000
2	2.0e-003	1.00000000	0.99999999
3	3.9e-003	1.00000000	1.00000000
4	7.8e-003	1.00000000	0.99999977
5	0.015625	1.00000000	0.87794335
6	0.031250	1.00000000	0.57434605
7	0.062500	1.00000000	2.434e-014
8	0.125000	0.99461075	1.818e-009
9	0.250000	0.85756332	2.412e-011
10	0.500000	5.389e-009	2.277e-011
11	1.000000	1.861e-009	2.573e-010

The number of selected features out of 2 : 2
The average shrinkage factor: 1

For the sequential (and not so for the combined) method each Θ iteration also takes a Cstep for estimating $\mathbf{C}$ and $\mathbf{b}$:

Cstep: Training HINGE Loss for Set of Lambdas

Ind	Log2L	Lambda	ClassEr	HingeEr	QP	LP
1	-16	1.526e-005	0.31666667	0.53426933	6	0
2	-15	3.052e-005	0.35000000	0.54486737	6	0
3	-14	6.104e-005	0.33333333	0.55366514	6	0
4	-13	1.221e-004	0.33333333	0.56084147	6	0
5	-12	2.441e-004	0.33333333	0.56561755	6	0
6	-11	4.883e-004	0.33333333	0.57856445	6	0
7	-10	9.766e-004	0.35000000	0.60644669	6	0
8	-9	0.00195313	0.40000000	0.65298646	6	0
9	-8	0.00390625	0.43333333	0.69792435	6	0

```
The optimal lambda.theta on log2 scale: -16
```

The Θ iteration ist terminated due to small differences in the successive $\mathbf{\Theta}$ values:

```
*** Iter=1 : SSQ(Th)=3.671e-009 SSQ(b)=2.11e-008 SSQ(C)=3.55e-006
         *** L_1(Th)=4.086e-009 L_oo(Th)=3.645e-009 ***


               Total Number of Kernel Calls: 27
                   Time for Optimization: 3
                   Total Processing Time: 4


print "GOF=", gof;
print "Theta=",theta;
print "Cmat=",cmat;
print "bvec=",bvec;
print "Yprd=",yprd;


                 Theta Weights for 2 Dimensions
                 ******************************

                   Dense Row Vector (ncol=2)

                   R |       Dim_1      Dim_2
                         1.0000000  1.0000000


                Model Intercepts b for 3 Ylevels
                ********************************

                  Dense Column Vector (nrow=3)

               C |      Ylev_1     Ylev_2     Ylev_3
                    -0.1271594  0.1708751 -0.0437157


                 Model Weights C for 3 Ylevels
                 *****************************

                     Dense Matrix (3 by 60)

       |          1          2          3          4          5
     ---------------------------------------------------------------
Ylev_1 |  -364.08889 -364.08889  0.0000000 -364.08889  0.0000000
```

```
Ylev_2 |   728.17778  728.17778  0.0000000  728.17778  0.0000000
Ylev_3 |  -364.08889 -364.08889  0.0000000 -364.08889  0.0000000

       |          6          7          8          9         10
     ------------------------------------------------------------
Ylev_1 |  -573.13437  728.17778 -519.67204  37.450855  257.78956
Ylev_2 |   519.13230 -364.08889  572.59463  527.40791  257.78956
Ylev_3 |   54.002067 -364.08889 -52.922591 -564.85876 -515.57912

       |         11         12         13         14         15
     ------------------------------------------------------------
Ylev_1 |   436.65312 -364.08889  484.79812  57.220761 -437.92367
Ylev_2 |  -655.61355 -364.08889 -607.46854 -574.74371  654.34300
Ylev_3 |   218.96043  728.17778  122.67042  517.52295 -216.41934

       |         16         17         18         19         20
     ------------------------------------------------------------
Ylev_1 |   710.08457  726.99604 -501.50649 -291.09166  0.0000000
Ylev_2 |  -327.90247 -365.27063  590.76017 -400.58750  0.0000000
Ylev_3 |  -382.18210 -361.72540 -89.253677  691.67916  0.0000000

       |         21         22         23         24         25
     ------------------------------------------------------------
Ylev_1 |   29.799862  66.609602 -295.85319 -728.17778 -4.85e-013
Ylev_2 |   29.799862 -133.21920  147.92659  364.08889  2.43e-013
Ylev_3 |  -59.599725  66.609602  147.92659  364.08889  2.43e-013

       |         26         27         28         29         30
     ------------------------------------------------------------
Ylev_1 |   0.0000000 -198.20368 -364.08889  176.93245  601.51253
Ylev_2 |   0.0000000  645.23517 -364.08889  176.93245 -490.75413
Ylev_3 |   0.0000000 -447.03149  728.17778 -353.86489 -110.75840

       |         31         32         33         34         35
     ------------------------------------------------------------
Ylev_1 |   497.39495 -724.56740 -130.61938 -5.05e-015  32.153910
Ylev_2 |  -594.87172  367.69926 -480.82365  2.53e-015 -562.21029
Ylev_3 |   97.476771  356.86814  611.44302  2.53e-015  530.05638

       |         36         37         38         39         40
     ------------------------------------------------------------
Ylev_1 |   364.08889  364.08889  2.43e-013  694.96247  114.78129
Ylev_2 |   364.08889  364.08889 -4.85e-013 -397.30420  114.78129
Ylev_3 |  -728.17778 -728.17778  2.43e-013 -297.65827 -229.56259
```

	41	42	43	44	45
Ylev_1	15.191551	-728.17778	149.07044	-570.49570	-4.7224749
Ylev_2	15.191551	364.08889	-298.14089	521.77096	522.90044
Ylev_3	-30.383102	364.08889	149.07044	48.724740	-518.17796

	46	47	48	49	50
Ylev_1	-200.44295	454.39304	728.17778	-364.08889	425.71062
Ylev_2	204.21297	-637.87363	-364.08889	-364.08889	-666.55605
Ylev_3	-3.7700142	183.48059	-364.08889	728.17778	240.84543

	51	52	53	54	55
Ylev_1	208.00603	-673.98322	364.08889	24.909566	-364.08889
Ylev_2	-650.13635	336.99161	364.08889	533.67855	-364.08889
Ylev_3	442.13032	336.99161	-728.17778	-558.58812	728.17778

	56	57	58	59	60
Ylev_1	-716.11621	0.0000000	728.17778	0.0000000	364.08889
Ylev_2	376.15045	0.0000000	-364.08889	0.0000000	-728.17778
Ylev_3	339.96576	0.0000000	-364.08889	0.0000000	364.08889

Predicted Values and Residuals: Tarining Data

****

Obs	Yobs	Ypred	Residual
1	2.0000000	2.0000000	0.00000000
2	2.0000000	2.0000000	0.00000000
3	2.0000000	2.0000000	0.00000000
4	2.0000000	2.0000000	0.00000000
5	3.0000000	3.0000000	0.00000000
6	2.0000000	2.0000000	0.00000000
7	1.0000000	2.0000000	-1.00000000
8	2.0000000	1.0000000	1.00000000
9	2.0000000	3.0000000	-1.00000000
10	2.0000000	2.0000000	0.00000000
11	1.0000000	1.0000000	0.00000000
12	3.0000000	3.0000000	0.00000000
13	1.0000000	2.0000000	-1.00000000
14	3.0000000	2.0000000	1.00000000

15	2.0000000	2.0000000	0.00000000
16	1.0000000	1.0000000	0.00000000
17	1.0000000	1.0000000	0.00000000
18	2.0000000	2.0000000	0.00000000
19	3.0000000	3.0000000	0.00000000
20	2.0000000	2.0000000	0.00000000
21	1.0000000	1.0000000	0.00000000
22	3.0000000	3.0000000	0.00000000
23	3.0000000	3.0000000	0.00000000
24	2.0000000	2.0000000	0.00000000
25	3.0000000	3.0000000	0.00000000
26	3.0000000	3.0000000	0.00000000
27	2.0000000	2.0000000	0.00000000
28	3.0000000	1.0000000	2.00000000
29	2.0000000	2.0000000	0.00000000
30	1.0000000	1.0000000	0.00000000
31	1.0000000	2.0000000	-1.00000000
32	2.0000000	2.0000000	0.00000000
33	3.0000000	2.0000000	1.00000000
34	3.0000000	3.0000000	0.00000000
35	3.0000000	2.0000000	1.00000000
36	2.0000000	3.0000000	-1.00000000
37	2.0000000	3.0000000	-1.00000000
38	3.0000000	3.0000000	0.00000000
39	1.0000000	2.0000000	-1.00000000
40	1.0000000	1.0000000	0.00000000

41	1.0000000	1.0000000	0.00000000
42	2.0000000	1.0000000	1.00000000
43	1.0000000	1.0000000	0.00000000
44	2.0000000	2.0000000	0.00000000
45	2.0000000	2.0000000	0.00000000
46	2.0000000	2.0000000	0.00000000
47	1.0000000	2.0000000	-1.00000000
48	1.0000000	1.0000000	0.00000000
49	3.0000000	3.0000000	0.00000000
50	1.0000000	2.0000000	-1.00000000
51	3.0000000	2.0000000	1.00000000
52	2.0000000	2.0000000	0.00000000
53	2.0000000	3.0000000	-1.00000000
54	2.0000000	2.0000000	0.00000000
55	3.0000000	2.0000000	1.00000000
56	2.0000000	1.0000000	1.00000000
57	3.0000000	3.0000000	0.00000000
58	1.0000000	1.0000000	0.00000000

59	3.0000000	3.0000000	0.00000000
60	3.0000000	3.0000000	0.00000000

GOF=

	g-of-fit
Failure	0.0000
CompTime	5.0000
OptLambda	-16.0000
OptLambTh	-10.0000
Nselected	2.0000
Shrinking	1.0000
SSQ(Theta)	4e-009
SSQ(b)	2e-008
SSQ(C)	4e-006
ThetaIter	1.0000
unused	.
unused	.

Theta=

	Dim_1	Dim_2
1	1.00000	1.00000

bvec=

	1
Ylev_1	-0.12716
Ylev_2	0.17088
Ylev_3	-0.04372

```
Yprd=
   |         1
---------------
 1 |   2.00000
 2 |   2.00000
 3 |   2.00000
 4 |   2.00000
 5 |   3.00000
 6 |   2.00000
 7 |   2.00000
 8 |   1.00000
 9 |   3.00000
10 |   2.00000
11 |   1.00000
12 |   3.00000
13 |   2.00000
14 |   2.00000
15 |   2.00000
16 |   1.00000
17 |   1.00000
18 |   2.00000
19 |   3.00000
20 |   2.00000
21 |   1.00000
22 |   3.00000
23 |   3.00000
24 |   2.00000
25 |   3.00000
26 |   3.00000
27 |   2.00000
28 |   1.00000
29 |   2.00000
30 |   1.00000
```

```
   |         1
---------------
31 |   2.00000
32 |   2.00000
33 |   2.00000
34 |   3.00000
35 |   2.00000
36 |   3.00000
37 |   3.00000
38 |   3.00000
39 |   2.00000
40 |   1.00000
41 |   1.00000
42 |   1.00000
43 |   1.00000
44 |   2.00000
45 |   2.00000
46 |   2.00000
47 |   2.00000
48 |   1.00000
49 |   3.00000
50 |   2.00000
51 |   2.00000
52 |   2.00000
53 |   3.00000
54 |   2.00000
55 |   2.00000
56 |   1.00000
57 |   3.00000
58 |   1.00000
59 |   3.00000
60 |   3.00000
```

3.10 Function `toeplitz`

b = toeplitz(a)

Purpose: The `toeplitz` function computes a Toeplitz matrix from an input vector a block Toeplitz matrix from an $nr \times nc$ input matrix. One dimension of the input argument must be a (int) multiple of the other dimension, i.e. either $nr = k * nc$ or $nc = k * nr$.

nr = k * nc Be $\mathbf{A}$ of form

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \mathbf{A}_3 \\ \vdots \\ \mathbf{A}_n \end{bmatrix}$$

then

$$\mathbf{B} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2^T & \mathbf{A}_3^T & \cdots & \mathbf{A}_n^T \\ \mathbf{A}_2 & \mathbf{A}_1 & \mathbf{A}_2^T & \cdots & \mathbf{A}_{n-1}^T \\ \mathbf{A}_3 & \mathbf{A}_2 & \mathbf{A}_1 & \cdots & \mathbf{A}_{n-2}^T \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{A}_n & \mathbf{A}_{n-1} & \mathbf{A}_{n-2} & \cdots & \mathbf{A}_1 \end{bmatrix}$$

nc = k * nr Be $\mathbf{A}$ of form

$$\mathbf{A} = [\mathbf{A}_1|\mathbf{A}_2|\mathbf{A}_3|\cdots|\mathbf{A}_n]$$

then

$$\mathbf{B} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{A}_3 & \cdots & \mathbf{A}_n \\ \mathbf{A}_2^T & \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_{n-1} \\ \mathbf{A}_3^T & \mathbf{A}_2^T & \mathbf{A}_1 & \cdots & \mathbf{A}_{n-2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{A}_n^T & \mathbf{A}_{n-1}^T & \mathbf{A}_{n-2}^T & \cdots & \mathbf{A}_1 \end{bmatrix}$$

Input: The only argument `a` must be a either a numeric vector or matrix of (nonmissing) values.

Output: The output is the $\max(nr, nc) \times \max(nr, nc)$ square matrix $\mathbf{B}$ which is symmetric when submatrix $\mathbf{A}_1$ is symmetric.

Restrictions: 1. Missing values and string data are not permitted.

2. One dimension of the input argument must be a (int) multiple of the other dimension, i.e. either $nr = k * nc$ or $nc = k * nr$.

Relationships:

Examples: 1. Example from SAS/IML Manual:

```
/* input row vector */
vec = [ 1:5 ];
toe = toeplitz(vec);
print "Toeplitz1=",toe;
```

```
Toeplitz1=
 S |         1         2         3         4         5
-------------------------------------------------------
 1 |   1.00000
 2 |   2.00000   1.00000
 3 |   3.00000   2.00000   1.00000
 4 |   4.00000   3.00000   2.00000   1.00000
 5 |   5.00000   4.00000   3.00000   2.00000   1.00000
```

```
   /* input column vector */
   vec = [ 1:5 ]‘;
   toe = toeplitz(vec);
   print "Toeplitz2=",toe;
```

```
Toeplitz2=
 S |         1         2         3         4         5
-------------------------------------------------------
 1 |   1.00000
 2 |   2.00000   1.00000
 3 |   3.00000   2.00000   1.00000
 4 |   4.00000   3.00000   2.00000   1.00000
 5 |   5.00000   4.00000   3.00000   2.00000   1.00000
```

```
   /*--- [2] Matrix ---*/
   mat = [ 1 2,  3 4 , 5 6 , 7 8 ];
   toe = toeplitz(mat);
   print "Toeplitz3=",toe;
```

```
Toeplitz3=
   |         1         2         3         4
-------------------------------------------
 1 |   1.00000   2.00000   5.00000   7.00000
 2 |   3.00000   4.00000   6.00000   8.00000
 3 |   5.00000   6.00000   1.00000   2.00000
 4 |   7.00000   8.00000   3.00000   4.00000
```

```
   mat = [ 1 2 3 4, 5 6 7 8 ];
   toe = toeplitz(mat);
   print "Toeplitz4=",toe;
```

```
Toeplitz4=
   |         1         2         3         4
```

```
-----------------------------------------
 1 |   1.00000   2.00000   3.00000   4.00000
 2 |   5.00000   6.00000   7.00000   8.00000
 3 |   3.00000   7.00000   1.00000   2.00000
 4 |   4.00000   8.00000   5.00000   6.00000
```