CMAT Newsletter: December 2004

Wolfgang M. Hartmann

December 2004

1 General Remarks

In 2004 I spent most of my time at SAS in Cary NC finishing up my new PROC SVM and starting a new PROC KPLS for the SAS Enterprise Miner product. In addition I spend some time for organizing a session and a talk at the PARA04 conference in Copenhagen (Denmark) and even as an actor in a documentary movie for a friend. Also, some time was spent on my small section for Rod McDonald's Festschrift and on a few reviews for JSS and so on. So, not much time was left for new functionality in CMAT and programming on my private Laptop is much more tedious than on my Desktop here in Heidelberg.

Some acknowledgments: The new `scalpha` function would not have been written without Prof. Maydeu-Olivares contribution and help with his Mathematica code. Unfortunately, I do not have a reference for his paper on this. The asymptotic standard errors for `factor` and `pca` were only possible by looking into the ROSEF Fortran routines of Prof. Ogasarawa. The new rotation methods were implemented after looking into the Bernaard & Jennrich software that available on their UCLA website. The `randisc` function was implemented during reviewing the Marsaglia, Tsang, & Wang (2004) paper for JSS.

2 Modifications of Features

2.1 Fixed Bugs

1. I found an incredible bug when multiplying a symmetric matrix **A** with a lower triangular matrix **B** (yes, Albert, like a factor loading matrix). I only muliplied by the diagonal of **B**. A way around would be to set the zero values of the lower triangular **B** with very small values. For the new version, however, the bug is fixed. (BTW, I have about 70 different methods of matrix multiplication implemented due to different forms, data types, and storage types of matrices. But I have much confidence in the hypothesis that they are all correct now.)

2. Another integer overflow was found und corrected at the transpose of a very large matrix. This algorithm is not trivial since it transposes a

rectangular matrix incore. (But now it seems to work for matrices like those of the Roche microarray data which is about 1000 by 30000 entries large.)

3. I found a number of pretty bad bugs in the `factor`, `frotate`, and `svm` functions.

2.2 Extensions to `reg()` Function

Only for least squares regression: If the `nobstat` (no observation statistics) option is not specified and `print` option is specified larger than 3, now an additional observationwise table is printed following the table of residuals and predicted values. The table contains a set of indicators for identifying outliers (see Neter et.al, 1990, chapter 11).
This additional table contains the following columns:

1. the diagonal values h_{ii} of the *hat matrix*

$$\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$$

2. the *studentized deleted residuals* d_i^* or *externally studendized residuals* with the probability of the $t(n-p-1)$ value

3. the *DFFITS* value for the influence of the fitted values $\hat{y}_i$. Cases are marked with an asterisk when the DFFITS value is larger than $2\sqrt{p/n}$.

4. Cook's distance D_i with probability of the $F(p, n-p)$ value. Cases are marked with an asterisk if the $F(p, n-p)$ probability is larger than 2α.

Here, the number of parameters is denoted by p and the number of cases by n. The following is the body fat example from Neter et.al. (1990, table 8.1):

```
print "Body Fat Example, N=20 (Table 8.1, p.272)";
body = [ 1  19.5  43.1  29.1  11.9 ,
         2  24.7  49.8  28.2  22.8 ,
         3  30.7  51.9  37.0  18.7 ,
         4  29.8  54.3  31.1  20.1 ,
         5  19.1  42.2  30.9  12.9 ,
         6  25.6  53.9  23.7  21.7 ,
         7  31.4  58.5  27.6  27.1 ,
         8  27.9  52.1  30.6  25.4 ,
         9  22.1  49.9  23.2  21.3 ,
        10  25.5  53.5  24.8  19.3 ,
        11  31.1  56.6  30.0  25.4 ,
        12  30.4  56.7  28.3  27.2 ,
        13  18.7  46.5  23.0  11.7 ,
        14  19.7  44.2  28.6  17.8 ,
        15  14.6  42.7  21.3  12.8 ,
```

```
       16  29.5  54.4  30.1  23.9 ,
       17  27.7  55.3  25.7  22.6 ,
       18  30.2  58.6  24.6  25.4 ,
       19  22.7  48.2  27.1  14.8 ,
       20  25.2  51.0  27.5  21.1 ];
vname = [ "Subject" "Triceps" "Thigh" "Midarm" "Body" ];
body = cname(body,vname);


/* Y = -19.174 + 0.2224 X[1] + 0.6594 X[2] */
modB = "5 = 2 3";
optn = [ "print"      4 ,
         "meth"    "xpx"];
< gof,parm > = reg(body,modB,"l_2",optn);


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

                Number Valid Observations   20
                Response Variable          Y[5]
                N Independend Variables      2
                Least-Squares (L_2) Regression
                Significance Level:  0.0500000
                Design Coding:  Rank-Deficient
                No Variable Selection Process


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

                     Intercept + X2 + X3


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

 Column  Nobs        Mean     Std Dev   Skewness   Kurtosis
 Y[5]      20   20.195000   5.1061856  -0.4248862  -1.0091843
 X[2]      20   25.305000   5.0232591  -0.5318842  -0.7945173
 X[3]      20   51.170000   5.2346115  -0.4174940  -0.9315992
```

****
Least-Squares (L_2) Regression
****

SSQ Model	385.4387097	SSQ Total	495.3895000
SSQ Error (SSE)	109.9507903	L2 Norm = sqrt(SSE)	10.48574224
Model DF	2	Error DF	17
Root MSE	2.543166048	Res.Err.Var. (MSE)	6.467693549
F Value	29.79723040	Probability>F	2.7742e-006
Dep. Var.	20.19500000	Coef. Var. (C.V.)	12.59304802
R Squared	0.778051835	Adj. R Squared	0.751940287
Akaike IC (AIC)	40.08601263	Bayesian IC (BIC)	43.08255242
Schwarz BC (SBC)	25.09881581	Mallows CP	3.000000000

Var	Estimate	AStdErr	T_Value	Prob	Low_W_CI	Upp_W_CI
Intercep	-19.174246	8.360641	-2.2934	0.0348	-35.56080	-2.787691
X1	0.2223526	0.303439	0.7328	0.4737	-0.372377	0.817082
X2	0.6594218	0.291187	2.2646	0.0369	0.088705	1.230138

Covariance Matrix of Parameter Estimates

1|	69.9		
2|	1.847	0.09208	
3|	-2.273	-0.08163	0.08479

Predicted Values and Residuals
****

Case	Observed	Residual	Predicted	Lower CI	Upper CI	ASE Pred
1	11.90000	-1.682709	13.58271	7.464765	19.70065	2.899753
2	22.80000	3.642931	19.15707	15.84551	22.46863	1.569596
3	18.70000	-3.175970	21.87597	13.55400	30.19794	3.944408
4	20.10000	-3.158465	23.25847	18.71342	27.80351	2.154238
5	12.90000	-2.9e-004	12.90029	6.104673	19.69590	3.220953
6	21.70000	-0.360816	22.06082	17.16707	26.95457	2.319515
7	27.10000	0.716199	26.38380	21.00255	31.76505	2.550580
8	25.40000	4.014733	21.38527	17.15099	25.61955	2.006943
9	21.30000	2.655106	18.64489	14.02477	23.26502	2.189824
10	19.30000	-2.474812	21.77481	17.24404	26.30558	2.147472

11	25.40000	0.335806	25.06419	20.33058	29.79781	2.243614
12	27.20000	2.225511	24.97449	20.46386	29.48512	2.137925
13	11.70000	-3.946861	15.64686	9.883588	21.41013	2.731648
14	17.80000	3.447456	14.35254	9.102840	19.60225	2.488229
15	12.80000	0.570587	12.22941	4.352532	20.10629	3.733445
16	23.90000	0.642298	23.25770	19.04570	27.46971	1.996385
17	22.60000	-0.850946	23.45095	19.01675	27.88514	2.101698
18	25.40000	-0.782920	26.18292	20.12953	32.23631	2.869155
19	14.80000	-2.857289	17.65729	14.12641	21.18817	1.673548
20	21.10000	1.040449	20.05955	17.00569	23.11341	1.447453

```
Diag of Hat Matrix and Cook's Distance
**************************************
```

Case	DiagHat	StudRes	tProb	DFFITS	CooksD	FProb
1	0.201013	-0.729985	0.2380	-0.366147	0.045951	0.0135
2	0.058895	1.534254	0.0723	0.383810	0.045481	0.0133
3	0.371933*	-1.654330	0.0588	-1.273067*	0.490157*	0.3063
4	0.110940	-1.348484	0.0981	-0.476348	0.072162	0.0259
5	0.248010	-1.3e-004	0.5000	-7.3e-005	1.9e-009	0.0000
6	0.128616	-0.147549	0.4423	-0.056687	1.1e-003	0.0001
7	0.155517	0.298128	0.3847	0.127937	5.8e-003	0.0006
8	0.096288	1.760092*	0.0487	0.574521	0.097939	0.0399
9	0.114636	1.117649	0.1401	0.402165	0.053134	0.0167
10	0.110244	-1.033728	0.1583	-0.363873	0.043957	0.0127
11	0.120337	0.136661	0.4465	0.050546	9.0e-004	0.0000
12	0.109266	0.923179	0.1848	0.323337	0.035154	0.0092
13	0.178382	-1.825903*	0.0433	-0.850781*	0.212150*	0.1134
14	0.148007	1.524763	0.0734	0.635514	0.124893	0.0559
15	0.333212*	0.267150	0.3964	0.188852	0.012575	0.0020
16	0.095277	0.258132	0.3998	0.083768	2.5e-003	0.0002
17	0.105595	-0.344509	0.3675	-0.118373	4.9e-003	0.0005
18	0.196793	-0.334408	0.3712	-0.165527	9.6e-003	0.0014
19	0.066954	-1.176171	0.1284	-0.315071	0.032360	0.0081
20	0.050085	0.409356	0.3439	0.093997	3.1e-003	0.0002

```
Threshold for DiagHat Values (2*p/n)               0.300000000
Studentized Residuals t(1.-alfa,n-p-1)             1.745883676
Threshold for DFFITS (2*sqrt(p/n))                 0.774596669
```

2.3 Extensions to `svm()` Function

Additional options for `svm`:

Option Name	Second Column	Meaning
"ktest"		test inertia of kernel matrix

The `ktest` option is recommended especially for the sigmoid kernel function which for certain parameters results in nonpositive definite kernel matrices. The `ktest` option is not of much use with the grid search and parameter tuning.

Not very visible for the user, but a highly efficient speedup in some applications resultet from the following extension of the code: during initialization of the `svm` call it is decided between three data storage schemes:

1. For applications with $N < n$ it is checked whether the $N \times N$ kernel matrix is not too large to fit incore. In this case, a flag is set and the optimization methods use this fact for speedier processing. The kernel matrix is only updated when either the observation or variable subset is changed, like for Loo or cross validation.
2. For applications with $N > n$ it is checked whether the entire $N \times n$ data matrix $\mathbf{X}$ fits incore. In this case a flag is set and the optimization methods use this fact for speedier processing.
3. If neither of the first two cases apply, the data are kept on file and are processed in large segments.

Note, that for the FQP method the kernel matrix must be always incore. For linear kernel and methods PSVM, LSVM, and ASVM it is commonly assumed that $N \geq n$ and the kernel matrix is never stored incore.

Another decomposition QP method similar to the SMO method was added which can solve the following four tasks

1. C Classification
2. ν Classification
3. ϵ Regression
4. ν Regression

This method is similar to that of LIBSVM by Hsu & Lin (1999) which is also part of the R statistical language ().

Task	Method	Shape of Data	Apply for Size
”clas”			
C Classification	FQP	$N > n$ and $N < n$	small lin. and nonlin.
	DQP	$N > n$ and $N < n$	large lin. and nonlin.
	LSSVM (Least Squares)	$N > n$	medium lin. and nonl.
	RSVM (Reduced)	$N > n$	very large nonlin.
	NSVM (Newton)	$N > n$	large lin., medium nonlin.
	LSVM (Lagrangean)	$N > n$	very large lin., medium nonlin.
	ASVM (Active Set)	$N > n$	very large lin., no nonlin.
	PSVM (Proximal)	$N > n$	very large lin., medium nonlin.
	SMO	$N > n$ and $N < n$	large lin. and nonlin.
”nuclas”			
ν Classification	FQP	$N > n$ and $N < n$	small lin. and nonlin.
	SMO	$N > n$ and $N < n$	large lin. and nonlin.
”regr”			
$L2$ Regression	FQP	$N > n$ and $N < n$	small lin. and nonlin.
	LSSVM (Least Squares)	$N > n$	medium lin. and nonlin.
”epsreg”			
ϵ Regression	FQP	$N > n$ and $N < n$	small linear and nonlin.
	SMO	$N > n$ and $N < n$	large linear and nonlin.
”nureg”			
ν Regression	FQP	$N > n$ and $N < n$	small lin and nonlin.
	SMO	$N > n$ and $N < n$	large linear and nonlin.
”fsclas”			
Feature Sel. Clas.	FSM	$N < n$	medium linear
”fsreg”			
Feature Sel. Regr.	FSM	$N < n$	medium linear
”sfsclas”			
Stepwise Feat. Sel.	SFSM	$N < n$	medium linear

2.4 Extensions to `frotate()` Function

A large number of rotation methods were added to the `frotate` function, as well as to the `factor` and the `sem` function. For that reason we list here the entire new document for the `frotate` function.

---

y = frotate(x) | < y,t,c > = frotate(x<,”sopt”<,par<,targ<,weig>>>>)

sopt= ”varmax”|”quamax”|”equmax”|”parmax”|”facpar”|”crafer”

sopt= ”tandm1”|”tandm2”|”mccamm”|”infmax”|”bentlr”|”minent”

sopt= ”simmax”|”oblmax”|”oblmin”|”quamin”|”biqmin”|”covmin”

sopt= ”geomin”|”promax”|”targt1”|”targt2”

Purpose: The `frotate()` function performs a variety of orthogonal and oblique rotations of factor patterns toward a so-called *simple structure*. For more details see [**?**].

Input: **x** The first input argument specifies the $n_{var} \times n_{fact}$ factor pattern matrix $\mathbf{X}$.

sopt The second input argument is a string option specifying the method of rotation, i.e. the form of objective function optimized in the rotation process:

crafer Crawford-Ferguson family (specify γ): orthogonal (oblique)

varmax Varimax rotation (crafer with $\gamma = \frac{1}{n}$): orthogonal (oblique)

quamax Quartimax rotation (crafer with $\gamma = 0$): orthogonal (oblique)

equmax Equamax rotation (crafer with $\gamma = \frac{q}{2n}$): orthogonal (oblique)

parmax Parsimax rotation (crafer with $\gamma = \frac{q-1}{n+q-2}$): orthogonal (oblique)

facpar Factor Parsimony rotation (crafer with $\gamma = 1$): orthogonal (oblique)

minent orthogonal Minimum Entropy rotation

tandm1 orthogonal Tandem 1 rotation

tandm2 orthogonal Tandem 2 rotation

mccamm orthogonal McCammon rotation

oblmin Direct Oblimin family (specify γ): oblique (orthogonal)

quamin Direct Quartimin (oblmin with $\gamma = 0$): oblique (orthogonal)

biqmin Bi-Quartimin (oblmin with $\gamma = .5$): oblique (orthogonal)

covmin Covarimin (oblmin with $\gamma = 1$): oblique (orthogonal)

simmax Simplimax (specify number nonzero loadings γ): oblique (orthogonal)

bentlr Bentler rotation criterion: oblique (orthogonal)

infmax Infomax rotation: oblique (orthogonal)

targt1 Target rotation: oblique (orthogonal)

targt2 (partially specified) Target rotation: oblique (orthogonal)

oblmax oblique Oblimax

geomin oblique Geomin

promax oblique Promax (specify odd integer *gamma* power)

See Bernaard & Jennrich (2004) and Browne (2001) for details.

par The (optional) third argument may be used to specify additional options:

par[1] Specifies the amount of printed output. Default is `ipri=0` i.e. not output is printed.

par[2] Specifies whether the normed (`norm=1`) or the unnormed (`norm=1`) pattern matrix is rotated. Default is `norm=0`.

par[3] Specifies a parameter γ used for Crawford-Ferguson, Direct Oblimin, and Simplimax rotation. The default depends on the method of rotation specified in the second input argument.

par[4] Specifies the criterion for terminating the iteration. If the rotation angle is less than the range `tol` the iterative process is stopped. Default is `tol=1.e-4`.

par[5] Specifies the maximum number of iterations in the iterative rotation process. Default is `maxit=100`.

par[6] Specifies whether orthogonal (par[6]=1) or oblique rotation is used (see table below for defaults).

par[7] Specifies the estimation algorithm. =0: traditional polynomial solver =1: gradient projection by Bernaards & Jennrich (2004). The traditional solver is implemented only for the Ctrawford-Ferguson and the direct Oblimin families.

par[8] Specifies initialization of transformation matrix for the gradient projection algorithm. (=0: use identity matrix; =1: use uniform random orthogonal matrix)

targ The fourth argument is valid only for target and partially specified target rotation. This target matrix must have the same dimension as the first input argument (loadings).

targ The fifth argument is valid only for Bowne's partially specified target rotation. This weight matrix must have the same dimension as the first input argument (loadings).

The following table shows whether orthogonal or oblique rotation is the default and if the default can be changed by specifying `par[6]`:

spec	Rotation Method	orthogonal	oblique
"quamax"	Quartimax (Carroll, 1953)	default	yes
"varmax"	Varimax (Kaiser, 1958)	default	yes
"equmax"	Equamax	default	yes
"parmax"	Parsimax	default	yes
"facpar"	Factor Parsimony	default	yes
"crafer"	Crawford-Ferguson (1970)	default	yes
"minent"	Minimum Entropy (Jennrich)	default	no
"tandm1"	Tandem 1 (Comrey,1967)	default	no
"tandm2"	Tandem 2 (Comrey, 1967)	default	no
"mccamm"	McCammon (1966)	default	no
"quamin"	Quartimin (Carroll, 1953)	yes	default
"simmax"	Simplimax (Kiers, 1994)	yes	default
"oblmin"	Oblimin (Carroll, 1960)	yes	default
"biqmin"	Bi-Quartimin	yes	default
"covmin"	Covarimin	yes	default
"bentlr"	Bentler Criterion (1977)	yes	default
"infmax"	Infomax (McKeon,1968)	yes	default
"oblmax"	Oblimax (Saunders,1961)	no	default
"geomin"	Geomin (Yates, 1984)	no	default
"promax"	Promax (Hendrickson,White)	no	default
"targt1"	Target Rotation (Harman,1976)	yes	default
"targt2"	Weighted Target (Browne,2001)	yes	default

Output and Algorithms: **y** The first output argument gives the $n_{var} \times n_{fact}$ rotated factor pattern matrix **X**.

t The second output argument is different for orthogonal or oblique rotation:

- For orthogonal rotation `t` is the orthogonal $n_{fact} \times n_{fact}$ rotation matrix **T**, in $\mathbf{Y} = \mathbf{XT}$.
- For oblique rotation `t` is the covariance matrix of the rotated factors.

Restrictions: 1. The input factor pattern `x` cannot contain any string data or missing values.

2. The input input factor pattern `x` cannot contain any complex data.

Relationships:

Relationships: procrust()

Examples: 1. The following example shows the results of orthogonal Varimax, oblique Promax and direct Oblimin retations for the *Five Socioeconomic Variables* data set by Harman (1966) (see also SAS/STAT manual p. 805, with the exception of direct Quartimin):

(a) Orthogonal VARIMAX Rotation:

```
fa = [ .87899  -.15847,
       .74215  -.57806,
       .71447   .67936,
       .71370  -.55515,
       .62533   .76621 ];
par = cons(4,1,.);
par[2]= 1;
< pat,rot> = frotate(fa,"vmax",par);
print "Normalized Orthogonal Varimax Rotation";
print "Rotated Factor Pattern", pat;
print "Rotation Matrix", rot;
```

```
Normalized Orthogonal Varimax Rotation

Rotated Factor Pattern
   |         1         2
------------------------
1 |   0.79085   0.41508
2 |   0.94071  -0.00004
3 |   0.14624   0.97499
4 |   0.90419   0.00055
5 |   0.02255   0.98874

Rotation Matrix
  |         1         2
------------------------
1 |   0.78895   0.61446
2 |  -0.61446   0.78895
```

(b) Oblique PROMAX Rotation:

```
par = cons(4,1,.);
par[2]= 0; par[3] = 3.;
< pat,cor > = frotate(fa,"prom",par);
print "Promax (Hendricksen & White, 1964) Rotation: Power k=3";
print "Rotated Factor Pattern", pat;
print "Factor Correlations", cor;
```

```
Promax (Hendricksen & White, 1964) Rotation: Power k=3

Rotated Factor Pattern
  |         1         2
------------------------
1 |   0.75870   0.33305
2 |   0.95844  -0.10690
3 |   0.03848   0.97672
```

```
4 |   0.92116  -0.10215
5 |  -0.08910   1.00477

Factor Correlations
S |         1         2
------------------------
1 |   1.00000
2 |   0.22003   1.00000
```

(c) Oblique Quartimin Rotation:

```
par = cons(4,1,.);
par[2]= 0;
< pat,cor > = frotate(fa,"qmin",par);
print "Direct Quartimin (Jennrich & Sampson, 1966) Rotation";
print "Rotated Factor Pattern", pat;
print "Factor Correlations", cor;
```

```
Direct Quartimin (Jennrich & Sampson, 1966) Rotation
Rotated Factor Pattern
   |         1         2
------------------------
 1 |   0.76238   0.34779
 2 |   0.95242  -0.08626
 3 |   0.05806   0.97378
 4 |   0.91539  -0.08231
 5 |  -0.06844   0.99904

Factor Correlations
 S |         1         2
------------------------
 1 |   1.00000
 2 |   0.18020   1.00000
```

2. The following example shows the results of some orthogonal and oblique rotation methods for the 8 *Physical Variables* example by Harman(1976):

```
print "8 Physical Variables from Harman";
fa = [ .830 -.396,
       .818 -.469,
       .777 -.470,
       .798 -.401,
       .786  .500,
       .672  .458,
       .594  .444,
```

```
        .647  .333 ];

 rnam = [ "HEIGHT"  "ARM_SPAN"  "FOREARM"
          "LOW_LEG"  "WEIGHT"  "DIAMETER"
          "CH_GIRTH"   "CH_WIDTH" ];
 cnam = [ "Fact1" "Fact2" ];
 fa = rname(fa,rnam);
 fa = cname(fa,cnam);

/* Infomax: default: par[6] = 0; */
par = cons(8,1,.);
par[1] = 2; par[2]= 0; par[6] = 1; par[7] = 1;
< pat,rot> = frotate(fa,"infmax",par);
print "Unnorm. Orthogonal Infomax Rotation: PG", pat, rot;

       *** Orthogonal Rotation: Infomax (McKeon,1968) ***

              Iter     OptCrit      Termin        Alfa

                 1  0.68567526 -0.96494412  1.00000000
                 2  0.61899085 -0.30271627  2.00000000
                 3  0.24161248 -1.17514453  2.00000000
                 4  0.23905703 -2.02496477  1.00000000
                 5  0.23900717 -2.95650440  1.00000000
                 6  0.23900649 -3.90850496  1.00000000
                 7  0.23900648 -4.86308776  1.00000000
                 8  0.23900648 -5.81796151  1.00000000

                     Rotated Factor Loadings
                     ***********************

                         Dense Matrix fa

                       |          1         2
                     --------------------------
                     1 |   0.8873221  0.2416101
                     2 |   0.9258697  0.1784667
                     3 |   0.8954565  0.1509523
                     4 |   0.8663387  0.2169383
                     5 |   0.2692519  0.8917956
                     6 |   0.2102833  0.7855755
                     7 |   0.1603191  0.7240648
                     8 |   0.2729160  0.6745479
```

```
                 Orthogonal Transformation Matrix
                 ********************************

                      Dense Matrix (2 by 2)

                       |          1          2
                     --------------------------
                     1 |   0.7577009  0.6526020
                     2 |  -0.6526020  0.7577009
```

```
/* McCammon: default: par[6] = 1; */
par = cons(8,1,.);
par[1] = 2; par[2]= 0; par[6] = 1; par[7] = 1;
< pat,rot> = frotate(fa,"mccamm",par);
print "Unnorm. Orth. McCammon Rotation: PG", pat, rot;
```

```
          *** Orthogonal Rotation: McCammon (1966) ***

              Iter     OptCrit      Termin        Alfa

                 1  1.97760232 -1.26064372  1.00000000
                 2  1.95765625 -0.51934186  2.00000000
                 3  1.54483618 -1.87748814  4.00000000
                 4  1.54470609 -2.19864508  1.00000000
                 5  1.54467672 -2.53121979  1.00000000
                 6  1.54467040 -2.86987292  1.00000000
                 7  1.54466908 -3.21149581  1.00000000
                 8  1.54466880 -3.55451140  1.00000000
                 9  1.54466874 -3.89816772  1.00000000
                10  1.54466873 -4.24211624  1.00000000
                11  1.54466873 -4.58619748  1.00000000
                12  1.54466873 -4.93033890  1.00000000
                13  1.54466873 -5.27450757  1.00000000

                     Rotated Factor Loadings
                     ***********************

                         Dense Matrix fa

                       |          1          2
                     --------------------------
                     1 |   0.3224094  0.8612597
                     2 |   0.2630899  0.9054660
                     3 |   0.2328881  0.8777199
```

```
       4 |   0.2959076  0.8426409
       5 |   0.9128258  0.1858631
       6 |   0.8016203  0.1369414
       7 |   0.7357639  0.0928626
       8 |   0.6968417  0.2095462


   Orthogonal Transformation Matrix
   ********************************

        Dense Matrix (2 by 2)

         |           1          2
       --------------------------
       1 |   0.7196962 -0.6942891
       2 |  -0.6942891 -0.7196962
```

```
/* Oblique Simplimax: default: par[6] = 0; */
par = cons(8,1,.);
par[1] = 2; par[2]= 0; par[3] = 8;  par[6] = 0; par[7] = 1;
< pat,tran,fcor > = frotate(fa,"simmax",par);
print "Unnorm. Obl. Simplimax Rotation: PG", pat,tran,fcor;
```

```
        *** Oblique Rotation: Simplimax (Kiers, 1994) ***

              Iter     OptCrit      Termin        Alfa

                 1  1.52626700 -0.36141691  1.00000000
                 2  1.51771816 -0.67645799  0.06250000
                 3  1.51418619 -0.90355307  0.12500000
                 4  1.51155960 -1.04333793  0.25000000
                 5  1.50634705 -0.04917045  0.50000000
                 6  0.41769096  0.30199071  1.00000000
                 7  0.08244918 -0.09009243  0.12500000
                 8  0.02495677 -0.52057354  0.12500000
                 9  0.01786313 -1.15316655  0.12500000
                10  0.01750000 -1.91578437  0.12500000
                11  0.01748932 -2.71493573  0.12500000
                12  0.01748905 -3.52068241  0.12500000
                13  0.01748904 -4.32748282  0.12500000
                14  0.01748904 -5.13444762  0.12500000

                   Rotated Factor Loadings
                   ***********************
```

```
                    Dense Matrix fa

                |          1          2
               --------------------------
               1 |   0.8926798  0.0537386
               2 |   0.9550391 -0.0258904
               3 |   0.9306006 -0.0491396
               4 |   0.8776400  0.0313667
               5 |   0.0088048  0.9273067
               6 |  -0.0216501  0.8233801
               7 |  -0.0565768  0.7670330
               8 |   0.0823719  0.6846131


           Oblique Transformation Matrix
           *****************************

                 Dense Matrix (2 by 2)

                |          1          2
               --------------------------
               1 |   0.8792614  0.8392675
               2 |  -0.4763395  0.5437188


                  Factor Correlations
                  *******************

                 Dense Matrix (2 by 2)

                |          1          2
               --------------------------
               1 |   1.0000000  0.4789407
               2 |   0.4789407  1.0000000
```

```
/* Obl. Target 1: default: par[6] = 0; */
par = cons(8,1,.);
par[1] = 2; par[2]= 0; par[6] = 0; par[7] = 1;
targ = cons(8,2,1.);
< pat,tran,fcor > = frotate(fa,"targt1",par,targ);
print "Unnorm. Target 1 Rotation: Old", pat,tran,fcor;
```

```
     *** Oblique Rotation: Target Rotation (Harman,1976) ***
```

Iter	OptCrit	Termin	Alfa
1	10.1236490	1.08943629	1.00000000
2	2.78169521	0.97352318	0.06250000
3	1.80732532	0.55922460	0.01562500
4	1.47916720	0.51987546	0.03125000
5	1.37916082	0.45839900	0.01562500
6	1.29595793	0.38457848	0.01562500
7	1.23631844	0.32000873	0.01562500
8	1.18836847	0.24865657	0.01562500
9	1.15256594	0.18385055	0.01562500
10	1.11338729	0.33319038	0.03125000
11	1.09111195	-0.00695281	0.00781250
12	1.07691786	-0.06019947	0.01562500
13	1.05608255	-0.16475916	0.03125000
14	1.03694727	-0.07681719	0.06250000
15	1.03326521	-0.42420592	0.00781250
16	1.03142107	-0.48783638	0.01562500
17	1.02896539	-0.47446862	0.03125000
18	1.02797017	-0.57308907	0.01562500
19	1.02723418	-0.65685590	0.01562500
20	1.02666861	-0.73152094	0.01562500
...	...	...	...
80	1.02462046	-4.93072935	0.01562500
81	1.02462046	-4.84603631	0.03125000
82	1.02462046	-5.09361885	0.00781250

Rotated Factor Loadings
****

Dense Matrix fa

	1	2
1	1.3273144	0.8193223
2	1.3401719	0.7552502
3	1.2829759	0.7011371
4	1.2843913	0.7742885
5	0.9007520	1.3563699
6	0.7576849	1.1798909
7	0.6537977	1.0689186
8	0.7734462	1.0643744

```
                 Oblique Transformation Matrix
                 *****************************

                    Dense Matrix (2 by 2)

                      |          1          2
                   --------------------------
                   1 |   0.4535341  0.2783004
                   2 |  -0.8912389  0.9604941


                     Factor Correlations
                     *******************

                    Dense Matrix (2 by 2)

                      |          1          2
                   --------------------------
                   1 |   1.0000000 -0.7298110
                   2 |  -0.7298110  1.0000000
```

```
/* Obl. Target 2: default: par[6] = 0; */
par = cons(8,1,.);
par[1] = 2; par[2]= 0; par[6] = 0; par[7] = 1;
targ = cons(8,2,0.);
weig = [ 1 0, 1 0, 1 0, 1 0,
         0 1, 0 1, 0 1, 0 1 ];
< pat,tran,fcor > = frotate(fa,"targt2",par,targ,weig);
print "Unnorm. Target 2 Rotation: Old", pat,tran,fcor;
```

```
     *** Oblique Rotation: Weighted Target (Browne,2001) ***

              Iter     OptCrit      Termin        Alfa

                 1  3.36634600  0.56325057  1.00000000
                 2  0.69873847  0.41760539  0.25000000
                 3  0.10311111  0.02891224  0.12500000
                 4  0.01901169 -0.80349500  0.12500000
                 5  0.01785599 -1.10949756  0.06250000
                 6  0.01757565 -1.42350253  0.06250000
                 7  0.01750970 -1.73770979  0.06250000
                 8  0.01749414 -2.04661506  0.06250000
                 9  0.01749036 -2.34648736  0.06250000
                10  0.01748906 -3.25549633  0.12500000
                11  0.01748904 -4.10787020  0.12500000
```

```
12  0.01748904 -4.93188559  0.12500000
13  0.01748904 -5.74469096  0.12500000

     Rotated Factor Loadings
     ***********************

         Dense Matrix fa

       |           1          2
     --------------------------
     1 |   0.0537387  0.8926798
     2 |  -0.0258904  0.9550391
     3 |  -0.0491396  0.9306007
     4 |   0.0313667  0.8776400
     5 |   0.9273071  0.0088039
     6 |   0.8233804 -0.0216509
     7 |   0.7670333 -0.0565775
     8 |   0.6846134  0.0823713

  Oblique Transformation Matrix
  *****************************

      Dense Matrix (2 by 2)

       |           1          2
     --------------------------
     1 |   0.8392679 -0.8792614
     2 |  -0.5437181 -0.4763396

       Factor Correlations
       *******************

      Dense Matrix (2 by 2)

       |           1          2
     --------------------------
     1 |   1.0000000 -0.4789414
     2 |  -0.4789414  1.0000000
```

2.5 Extensions to `factor()` Function

1. The same rotation methods as in the `rotate` or the new `pca` function are now also available in the `factor()` function.

2. In addition, asymptotic standard errors and confidence intervals are now computed for rotated factor loadings, the unique variances, and the factor correlations (when oblique rotation is applied).
3. Also for oblique rotation, the *factor structure* is computed with asymptotic standard errors and confidence intervals.
4. The numerical methods of the scree test are added to the `nfact`, the `percent`, and `mineigen` criterion for estimating the significant number of dimensions.
5. A new algorithm which relates the Ogasarawa ASE approach for rotated factor loadings and the Bernaard-Jennrich gradient specification for a large variety of orthogonal and oblique rotation methods makes it possible to compute approximate standard errors for all of these methods.

The asymptotic standard errors and Wald confidence intervals of rotated factor loadings depend on:

1. the kind of rotation method appplied and especially the fact whether the rotation is orthogonal or oblique;
2. if the (unstandardized) covariance or the (standardized) correlation matrix is analysed. (Note, you may read in a correlation matrix, but if you specify the analysis option `"anal"` as `"cov"`, the results correspond to those of a covariance analysis.)
3. if Kaiser normalization of the loadings is applied for the specific rotation method.

Examples: 1. Covariance (UNStand.) UNNormal. Orthog. VARIMAX:

```
print "8 Physical Variables from Harman";
corr = [  1.0     .846    .805    .859    .473    .398    .301    .382,
          .846    1.0     .881    .826    .376    .326    .277    .415,
          .805    .881    1.0     .801    .380    .319    .237    .345,
          .859    .826    .801    1.0     .436    .329    .327    .365,
          .473    .376    .380    .436    1.0     .762    .730    .629,
          .398    .326    .319    .329    .762    1.0     .583    .577,
          .301    .277    .237    .327    .730    .583    1.0     .539,
          .382    .415    .345    .365    .629    .577    .539    1.0 ];
nobs = 305;
labl = [ "HEIGHT"  "ARM SPAN"  "LENGTH OF FOREARM"
         "LENGTH OF LOWER LEG"  "WEIGHT"  "BITROCHANTERIC DIAMETER"
         "CHEST GIRTH"   "CHEST WIDTH" ];
rnam = [ "HEIGHT"  "ARM_SPAN"  "FOREARM"
         "LOW_LEG"  "WEIGHT"  "DIAMETER"
         "CH_GIRTH"   "CH_WIDTH" ];
corr = corr;  corr = cname(corr,rnam);
```

```
optn = [ "data"      "cor" ,
         "anal"      "cov" ,
         "nobs"        305 ,
         "meth"       "ml" ,
         "vers"     "semt" ,
         "prio"      "smc" ,
         "heyw"            ,
         "nfac"          2 ,
         "tech"   "nrridg" ,
         "cl"       "wald" ,
         "rvers"    "comm" ,
         "frot"   "varmax" ,
         "prin"          3 ];
< gof, est > = factor(corr,optn);
print "GOF=", gof;
print "Est=", est;
```

The first part of the output is similar for all succeeding examples and will be dropped later:

```
    Partial Corrs, Prior Communality Est., Number of Factors
    --------------------------------------------------------

      Partial Correlations Controlling all other Variables

              HEIGHT   ARM_SPAN    FOREARM    LOW_LEG     WEIGHT
HEIGHT     1.0000000
ARM_SPAN   0.3463871  1.0000000
FOREARM    0.0716667  0.5837059  1.0000000
LOW_LEG    0.4788432  0.1790842  0.1877339  1.0000000
WEIGHT     0.1825491 -0.1959834  0.0999127  0.0561332  1.0000000
DIAMETER   0.1029166 -0.0046150  0.0269871 -0.1223006  0.4915225
CH_GIRTH  -0.1459097  0.0913517 -0.1161478  0.1305798  0.4912359
CH_WIDTH  -0.0861932  0.2479890 -0.0865968 -0.0247644  0.2379439


      Partial Correlations Controlling all other Variables

                        DIAMETER   CH_GIRTH   CH_WIDTH
           DIAMETER    1.0000000
           CH_GIRTH    0.0543009  1.0000000
           CH_WIDTH    0.1773298  0.1204839  1.0000000


Kaiser's Measure of Sampling Adequacy: Over-all MSA = 0.845461
```

HEIGHT	ARM_SPAN	FOREARM	LOW_LEG	WEIGHT
0.86431470	0.81629625	0.85765444	0.88666181	0.77961439
DIAMETER	CH_GIRTH	CH_WIDTH		
0.85112147	0.82404409	0.89846891		

Prior Communality Estimates: SMC

HEIGHT	ARM_SPAN	FOREARM	LOW_LEG	WEIGHT
0.81620091	0.84931371	0.80057816	0.78843663	0.74883068

DIAMETER	CH_GIRTH	CH_WIDTH
0.60414883	0.56219059	0.47779203

Preliminary Eigenvalues Based on Prior Communalities
Total = 24.5249   Average = 3.0656

	1	2	3	4
Eigenvalue	20.5304638	4.85135547	0.29909150	0.07596672
Difference	15.6791083	4.55226396	0.22312478	0.14074754
Proportion	0.83712855	0.19781376	0.01219544	0.00309754
Cumulative	0.83712855	1.03494230	1.04713774	1.05023528

	5	6	7	8
Eigenvalue	-0.06478081	-0.30152863	-0.36187687	-0.50382710
Difference	0.23674781	0.06034824	0.14195022	
Proportion	-0.00264143	-0.01229481	-0.01475551	-0.02054352
Cumulative	1.04759385	1.03529903	1.02054352	1.00000000

Results of Scree Test for alpha=0.1
****

Diag of Hat Matrix and Cook's Distance
(starting with two smallest eigenvalues adding next)
****

Case	DiagHat	StudRes	tProb	DFFITS	CooksD	FProb
3	0.833333	.	.	.	2.500000*	0.5918
4	0.700000	-2.005807	0.1472	-3.063921*	1.868828*	0.6514
5	0.600000	-0.474355	0.3410	-0.580964	0.227540	0.1909
6	0.523810	-1.415936	0.1259	-1.485046*	0.881285*	0.5182
7	0.464286	-63.86292*	0.0000	-59.45315*	2.164544*	0.7897

```
8  0.416667  -9.155964* 0.0001 -7.738202*  2.022244* 0.7869
```

Recommended Dimension Based on Studendized Residual : 2
Recommended Dimension Based on Cooks D : 3
Threshold for Cooks Distance: 0.25

Fit Linear Regression Through N Points (Cases)
****

N	Eval	MSE	Rsquared	Fval	Prob	BIC
8	0.4961729	0.000000	1.000000	.	.	.
7	0.6381231	0.000000	1.000000	.	.	.
6	0.6984714	1.1e-003	0.948553	18.43763	0.146	-17.70653
5	0.9352192	2.8e-003	0.944500	34.03605	0.028	-18.30311
4	1.0759667	2.1e-003	0.971601	102.6392	0.002	-25.68375
3	1.2990915	2.6e-003	0.977146	171.0259	0.000	-30.67697
2	5.8513555	2.112048	0.516284	5.336638	0.069	10.15830
1	21.530464	31.26946	0.493512	5.846292	0.052	32.35079

The Bayes inform. criterion shows a minimum at 3 eigenvalues.

Some Indices by Malinowski (1991) for Nobs=305
****

N	Eval	RE	IE	IND	SLperc
1	21.530464	0.0717607	0.0253712	0.0014645	3.5161910
2	5.8513555	0.0530133	0.0265066	0.0014726	11.548137
3	1.2990915	0.0502058	0.0307447	0.0020082	40.186077
4	1.0759667	0.0476324	0.0336812	0.0029770	42.913285
5	0.9352192	0.0447552	0.0353821	0.0049728	44.716900
6	0.6984714	0.0431219	0.0373447	0.0107805	51.563701
7	0.6381231	0.0403336	0.0377286	0.0403336	57.022970
8	0.4961729	.	.	.	.

The IND function is monoton increasing showing no minimum.
Largest 1 Eigenvalue(s) with SLperc <= alfa= 10 %

****
Maximum Likelihood Factor Analysis
(Exploratory Factor Analysis)
****

Input Data. . . . . .Correlation Matrix

```
Analysis of . . . . . . . . . . . . . . . Covariance Matrix
Number of Items . . . . . . . . . . . . . . . . .         8
Number of Factors (NFactor Criterion) . . . . . .         2
Number of Subjects. . . . . . . . . . . . . . . .       305
Version . . . . . . . . . . . . . . . . . SEM-Type Matrix Model
Orthogonal Rotation Method. . . . . .Varimax (Kaiser, 1958)
Unnormed Rotation: Parameter. . . . . . . . . . . 1.0000000
ASE for Rotated Solution. . . . . . . . . . .Wald (Analyt.)
```

Name	Mean	Std Dev
HEIGHT	0.000000000	1.00000000
ARM_SPAN	0.000000000	1.00000000
FOREARM	0.000000000	1.00000000
LOW_LEG	0.000000000	1.00000000
WEIGHT	0.000000000	1.00000000
DIAMETER	0.000000000	1.00000000
CH_GIRTH	0.000000000	1.00000000
CH_WIDTH	0.000000000	1.00000000

Covariance Matrix

	HEIGHT	ARM_SPAN	FOREARM	LOW_LEG	WEIGHT
HEIGHT	1.0000000				
ARM_SPAN	0.8460000	1.0000000			
FOREARM	0.8050000	0.8810000	1.0000000		
LOW_LEG	0.8590000	0.8260000	0.8010000	1.0000000	
WEIGHT	0.4730000	0.3760000	0.3800000	0.4360000	1.0000000
DIAMETER	0.3980000	0.3260000	0.3190000	0.3290000	0.7620000
CH_GIRTH	0.3010000	0.2770000	0.2370000	0.3270000	0.7300000
CH_WIDTH	0.3820000	0.4150000	0.3450000	0.3650000	0.6290000

Covariance Matrix

	DIAMETER	CH_GIRTH	CH_WIDTH
DIAMETER	1.0000000		
CH_GIRTH	0.5830000	1.0000000	
CH_WIDTH	0.5770000	0.5390000	1.0000000

Determinant = 9.67398457e-004 (Ln = -6.9409e+000)

Predicted Model Matrix

	HEIGHT	ARM_SPAN	FOREARM	LOW_LEG	WEIGHT

HEIGHT	1.0000000				
ARM_SPAN	0.8540117	1.0000000			
FOREARM	0.8257605	0.8628643	1.0000000		
LOW_LEG	0.8147737	0.8418818	0.8138943	1.0000000	
WEIGHT	0.4671479	0.3837231	0.3745563	0.4331819	1.0000000
DIAMETER	0.3900251	0.3202526	0.3126076	0.3616351	0.7613073
CH_GIRTH	0.3317014	0.2604322	0.2547597	0.3044429	0.7273288
CH_WIDTH	0.4184390	0.3700118	0.3599727	0.3948788	0.6398503

Predicted Model Matrix

	DIAMETER	CH_GIRTH	CH_WIDTH
DIAMETER	1.0000000		
CH_GIRTH	0.6079248	1.0000000	
CH_WIDTH	0.5347187	0.5045987	1.0000000

Determinant = 1.00000000e+000 (Ln = 0.0000e+000)

Residual Matrix

	HEIGHT	ARM_SPAN	FOREARM	LOW_LEG	WEIGHT
HEIGHT	0.0000000				
ARM_SPAN	-0.0080117	0.0000000			
FOREARM	-0.0207605	0.0181357	0.0000000		
LOW_LEG	0.0442263	-0.0158818	-0.0128943	0.0000000	
WEIGHT	0.0058521	-0.0077231	0.0054437	0.0028181	0.0000000
DIAMETER	0.0079749	0.0057474	0.0063924	-0.0326351	6.93e-004
CH_GIRTH	-0.0307014	0.0165678	-0.0177597	0.0225571	0.0026712
CH_WIDTH	-0.0364390	0.0449882	-0.0149727	-0.0298788	-0.0108503

Residual Matrix

	DIAMETER	CH_GIRTH	CH_WIDTH
DIAMETER	0.0000000		
CH_GIRTH	-0.0249248	0.0000000	
CH_WIDTH	0.0422813	0.0344013	0.0000000

Average Off-diagonal Residual = 1.87208372e-002
Sum-of-Squares= 0.0147603

Rank Order of 10 Largest Residuals

CH_WIDTH,ARM_SPAN	LOW_LEG,HEIGHT	CH_WIDTH,DIAMETER
0.04498815473	0.04422634882	0.04228126272

CH_WIDTH,HEIGHT	CH_WIDTH,CH_GIRTH	DIAMETER,LOW_LEG
-0.03643900759	0.03440125602	-0.03263512800
CH_GIRTH,HEIGHT	CH_WIDTH,LOW_LEG	CH_GIRTH,DIAMETER
-0.03070144405	-0.02987880130	-0.02492479125
	CH_GIRTH,LOW_LEG	
	0.02255705987	

Distribution of Residuals
(Each * represents 1 residuals)

```
-0.0407136 - -0.0271424   4  11.11% | ****
-0.0271424 - -0.0135712   5  13.89% | *****
-0.0135712 -  0.0000000   4  11.11% | ****
 0.0000000 -  0.0135712  16  44.44% | ****************
 0.0135712 -  0.0271424   3   8.33% | ***
 0.0271424 -  0.0407136   1   2.78% | *
 0.0407136 -  0.0542848   3   8.33% | ***
```

Asymptotically Standardized Residuals

	HEIGHT	ARM_SPAN	FOREARM	LOW_LEG	WEIGHT
HEIGHT	0.0000000				
ARM_SPAN	-2.1748170	0.0000000			
FOREARM	-3.4755391	5.4990175	0.0000000		
LOW_LEG	5.9422214	-3.6528903	-1.8618633	0.0000000	
WEIGHT	1.7829437	-3.3855941	1.6928679	0.7668223	0.0000000
DIAMETER	0.6734393	0.6994809	0.5547220	-2.4835227	0.3095902
CH_GIRTH	-2.3947067	1.8689796	-1.4276909	1.5879800	0.9945315
CH_WIDTH	-2.4276347	4.3197244	-1.0266948	-1.7983091	-2.2431263

Asymptotically Standardized Residuals

	DIAMETER	CH_GIRTH	CH_WIDTH
DIAMETER	0.0000000		
CH_GIRTH	-1.7471159	0.0000000	
CH_WIDTH	2.0253387	1.4919427	0.0000000

Average Off-diagonal Residual = 2.15389668e+000
Sum-of-Squares= 182.808

```
        Rank Order of 10 Largest Residuals

  LOW_LEG,HEIGHT     FOREARM,ARM_SPAN    CH_WIDTH,ARM_SPAN
   5.94222144825       5.49901745652       4.31972442764

LOW_LEG,ARM_SPAN       FOREARM,HEIGHT      WEIGHT,ARM_SPAN
  -3.65289033083      -3.47553907380      -3.38559407921

DIAMETER,LOW_LEG      CH_WIDTH,HEIGHT      CH_GIRTH,HEIGHT
  -2.48352273910      -2.42763466883      -2.39470669132

                      CH_WIDTH,WEIGHT
                      -2.24312625059

Loadings and Unique Variances with ASE's and Wald CIs
-----------------------------------------------------

                          FAC_1                 FAC_2
HEIGHT     0.9111724  0.0442642  0.0000000  0.0000000
          [ 0.824416, 0.997929] [ 0.000000, 0.000000]

ARM_SPAN   0.9372669  0.0434627 -0.1202519  0.0330194
          [ 0.852082, 1.022452] [-0.184969,-0.055535]

FOREARM    0.9062615  0.0445519 -0.1118926  0.0353542
          [ 0.818941, 0.993582] [-0.181186,-0.042600]

LOW_LEG    0.8942036  0.0448724 -0.0313867  0.0364515
          [ 0.806255, 0.982152] [-0.102830, 0.040057]

WEIGHT     0.5126889  0.0578448  0.8050037  0.0429964
          [ 0.399315, 0.626062] [ 0.720732, 0.889275]

DIAMETER   0.4280475  0.0580745  0.6731050  0.0480620
          [ 0.314224, 0.541871] [ 0.578905, 0.767305]

CH_GIRTH   0.3640381  0.0588330  0.6716622  0.0495325
          [ 0.248728, 0.479349] [ 0.574580, 0.768744]

CH_WIDTH   0.4592314  0.0565938  0.5023671  0.0506518
          [ 0.348310, 0.570153] [ 0.403092, 0.601643]

                                     U_Var
           HEIGHT     0.1697649  0.0175229
                     [ 0.135421, 0.204109]
```

```
ARM_SPAN   0.1070702  0.0148859
          [ 0.077894, 0.136246]

FOREARM    0.1661701  0.0180289
          [ 0.130834, 0.201506]

LOW_LEG    0.1994147  0.0197322
          [ 0.160740, 0.238089]

WEIGHT     0.0891191  0.0291079
          [ 0.032069, 0.146170]

DIAMETER   0.3637049  0.0362058
          [ 0.292743, 0.434667]

CH_GIRTH   0.4163462  0.0398034
          [ 0.338333, 0.494359]

CH_WIDTH   0.5367337  0.0462854
          [ 0.446016, 0.627451]
```

```
(1) Standalone Fit Measures: -------------------------------
Fit criterion . . . . . . . . . . . . . . . . . .    0.2532
Normal Th. Chi-square (df = 13) . . . . . . . . . . . . . .
Bartlett Corrected Chi-square (df = 13) . . . . . . . . . .
Normal Theory Reweighted LS Chi-square  . . . . .  142.4181
Probability of Close Fit  . . . . . . . . . . . .    0.0000
Z-Test of Wilson & Hilferty (1931). . . . . . . .    6.3185
(2) Incremental Fit Measures: ------------------------------
Null Model Chi-square (df = 36) . . . . . . . .   2110.0336
RMSEA Estimate  . . . . .  0.1272  90%C.I.[ 0.1006,  0.1554]
ECVI Estimate . . . . . .  0.4091  90%C.I.[ 0.3294,  0.5142]
McDonald's (1989) Centrality. . . . . . . . . . .    0.9005
Tucker-Lewis Coefficient TLI. . . . . . . . . . .    0.9146
Bentler & Bonett's (1980) NFI . . . . . . . . . .    0.9635
Bentler's Comparative Fit Index . . . . . . . . .    0.9692
Parsimonious NFI (James, Mulaik, & Brett,1982). .    0.3479
Bollen (1986) Normed Index Rho1 . . . . . . . . .    0.8990
Bollen (1988) Non-normed Index Delta2 . . . . . .    0.9695
(3) Information Criteria: ----------------------------------
Akaike's Information Criterion. . . . . . . . . .   50.9612
Bozdogan's (1987) CAIC. . . . . . . . . . . . . .  -10.4029
Schwarz's Bayesian Criterion. . . . . . . . . . .    2.5971
```

```
(4) Other Fit Measures: ------------------------------------
Goodness of Fit Index (GFI) . . . . . . . . . . .    0.9370
Parsimonious GFI (Mulaik, 1989) . . . . . . . . .    0.3383
GFI Adjusted for Degrees of Freedom (AGFI). . . .    0.8254
Root Mean Square Residual (RMR) . . . . . . . . .    0.0202
Hoelter's (1983) Critical N . . . . . . . . . . .        90

        Rotated Factor Loadings with Standard Errors
        --------------------------------------------

                           FAC_1                 FAC_2
   HEIGHT     0.8712266  0.0443224  0.2668320  0.0345348
             [ 0.784356, 0.958097] [ 0.199145, 0.334519]

   ARM_SPAN   0.9313923  0.0431417  0.1594936  0.0313323
             [ 0.846836, 1.015948] [ 0.098083, 0.220904]

   FOREARM    0.8992982  0.0442926  0.1584067  0.0331098
             [ 0.812486, 0.986110] [ 0.093513, 0.223301]

   LOW_LEG    0.8641932  0.0449309  0.2318521  0.0350481
             [ 0.776130, 0.952256] [ 0.163159, 0.300545]

   WEIGHT     0.2544715  0.0360988  0.9198506  0.0450534
             [ 0.183719, 0.325224] [ 0.831548, 1.008154]

   DIAMETER   0.2121667  0.0418692  0.7689476  0.0493626
             [ 0.130105, 0.294229] [ 0.672199, 0.865696]

   CH_GIRTH   0.1513860  0.0427960  0.7488231  0.0503011
             [ 0.067507, 0.235265] [ 0.650235, 0.847412]

   CH_WIDTH   0.2919832  0.0478984  0.6148269  0.0519312
             [ 0.198104, 0.385862] [ 0.513044, 0.716610]

                                      U_Var
              HEIGHT     0.1697649  0.0175229
                        [ 0.135421, 0.204109]

              ARM_SPAN   0.1070702  0.0148859
                        [ 0.077894, 0.136246]

              FOREARM    0.1661701  0.0180289
                        [ 0.130834, 0.201506]
```

```
                LOW_LEG    0.1994147  0.0197322
                          [ 0.160740, 0.238089]

                WEIGHT     0.0891191  0.0291079
                          [ 0.032069, 0.146170]

                DIAMETER   0.3637049  0.0362058
                          [ 0.292743, 0.434667]

                CH_GIRTH   0.4163462  0.0398034
                          [ 0.338333, 0.494359]

                CH_WIDTH   0.5367337  0.0462854
                          [ 0.446016, 0.627451]
```

```
optn = [ "data"      "cor" ,
         "anal"      "cov" ,
         "nobs"        305 ,
         "meth"       "ml" ,
         "vers"     "semt" ,
         "prio"      "smc" ,
         "heyw"            ,
         "nfac"          2 ,
         "tech"   "nrridg" ,
         "cl"       "wald" ,
         "rvers"    "comm" ,
         "frot"   "varmax" ,
         "fidi"        "c" ,
         "prin"          3 ];
< gof, est > = factor(corr,optn);
print "GOF=", gof;
print "Est=", est;
```

```
               *********************************
               Maximum Likelihood Factor Analysis
                 (Exploratory Factor Analysis)
               *********************************

  Input Data. . . . . . . . . . . . . . . . .Correlation Matrix
  Analysis of . . . . . . . . . . . . . . . . Covariance Matrix
  Number of Items . . . . . . . . . . . . . . . . .          8
  Number of Factors (NFactor Criterion) . . . . . .          2
  Number of Subjects. . . . . . . . . . . . . . . .        305
  Version . . . . . . . . . . . . . . . . . SEM-Type Matrix Model
  Orthogonal Rotation Method. . . . . .Varimax (Kaiser, 1958)
```

```
Unnormed Rotation: Parameter. . . . . . . . . . . 1.0000000
ASE for Rotated Solution. . . . . . . . . . . Wald (FinDif)

        Rotated Factor Loadings with Standard Errors
        --------------------------------------------

                          FAC_1                 FAC_2
   HEIGHT     0.8712266  0.0443224  0.2668320  0.0345348
             [ 0.784356, 0.958097] [ 0.199145, 0.334519]

   ARM_SPAN   0.9313923  0.0431417  0.1594936  0.0313323
             [ 0.846836, 1.015948] [ 0.098083, 0.220904]

   FOREARM    0.8992982  0.0442926  0.1584067  0.0331098
             [ 0.812486, 0.986110] [ 0.093513, 0.223301]

   LOW_LEG    0.8641932  0.0449309  0.2318521  0.0350481
             [ 0.776130, 0.952256] [ 0.163159, 0.300545]

   WEIGHT     0.2544715  0.0360988  0.9198506  0.0450534
             [ 0.183719, 0.325224] [ 0.831548, 1.008154]

   DIAMETER   0.2121667  0.0418692  0.7689476  0.0493626
             [ 0.130105, 0.294229] [ 0.672199, 0.865696]

   CH_GIRTH   0.1513860  0.0427960  0.7488231  0.0503011
             [ 0.067507, 0.235265] [ 0.650235, 0.847412]

   CH_WIDTH   0.2919832  0.0478984  0.6148269  0.0519312
             [ 0.198104, 0.385862] [ 0.513044, 0.716610]


                                      U_Var
              HEIGHT     0.1697649  0.0175229
                        [ 0.135421, 0.204109]

              ARM_SPAN   0.1070702  0.0148859
                        [ 0.077894, 0.136246]

              FOREARM    0.1661701  0.0180289
                        [ 0.130834, 0.201506]

              LOW_LEG    0.1994147  0.0197322
                        [ 0.160740, 0.238089]

              WEIGHT     0.0891191  0.0291079
```

```
                     [ 0.032069, 0.146170]

          DIAMETER   0.3637049  0.0362058
                     [ 0.292743, 0.434667]

          CH_GIRTH   0.4163462  0.0398034
                     [ 0.338333, 0.494359]

          CH_WIDTH   0.5367337  0.0462854
                     [ 0.446016, 0.627451]
```

2. Correlation (Standard.) UNNormal. Orthog. VARIMAX:

```
optn = [ "data"      "cor" ,
         "anal"      "cor" ,
         "nobs"        305 ,
         "meth"       "ml" ,
         "vers"     "semt" ,
         "prio"      "smc" ,
         "heyw"            ,
         "nfac"          2 ,
         "tech"   "nrridg" ,
         "cl"       "wald" ,
         "rvers"    "comm" ,
         "frot"   "varmax" ,
         "prin"          3 ];
< gof, est > = factor(corr,optn);
```

```
               *********************************
               Maximum Likelihood Factor Analysis
                 (Exploratory Factor Analysis)
               *********************************

  Input Data. . . . . . . . . . . . . . . . .Correlation Matrix
  Analysis of . . . . . . . . . . . . . . . .Correlation Matrix
  Number of Items . . . . . . . . . . . . . . . . .          8
  Number of Factors (NFactor Criterion) . . . . . .          2
  Number of Subjects. . . . . . . . . . . . . . . .        305
  Version . . . . . . . . . . . . . . . . . SEM-Type Matrix Model
  Orthogonal Rotation Method. . . . . .Varimax (Kaiser, 1958)
  Unnormed Rotation: Parameter. . . . . . . . . . . 1.0000000
  ASE for Rotated Solution. . . . . . . . . . .Wald (Analyt.)
```

```
     Rotated Factor Loadings with Standard Errors
     --------------------------------------------

                          FAC_1                 FAC_2
HEIGHT     0.8712266  0.0141437  0.2668320  0.0300971
          [ 0.843506, 0.898948] [ 0.207843, 0.325821]

ARM_SPAN   0.9313923  0.0099879  0.1594936  0.0290540
          [ 0.911816, 0.950968] [ 0.102549, 0.216438]

FOREARM    0.8992982  0.0124617  0.1584067  0.0310294
          [ 0.874874, 0.923723] [ 0.097590, 0.219223]

LOW_LEG    0.8641932  0.0150131  0.2318521  0.0315357
          [ 0.834768, 0.893618] [ 0.170043, 0.293661]

WEIGHT     0.2544715  0.0315743  0.9198506  0.0190010
          [ 0.192587, 0.316356] [ 0.882609, 0.957092]

DIAMETER   0.2121667  0.0392023  0.7689476  0.0264383
          [ 0.135332, 0.289002] [ 0.717129, 0.820766]

CH_GIRTH   0.1513860  0.0413044  0.7488231  0.0284028
          [ 0.070431, 0.232341] [ 0.693155, 0.804492]

CH_WIDTH   0.2919832  0.0437368  0.6148269  0.0362967
          [ 0.206261, 0.377706] [ 0.543687, 0.685967]


                                 U_Var
           HEIGHT     0.1697649  0.0405554
                     [ 0.090278, 0.249252]

           ARM_SPAN   0.1070702  0.0405554
                     [ 0.027583, 0.186557]

           FOREARM    0.1661701  0.0405554
                     [ 0.086683, 0.245657]

           LOW_LEG    0.1994147  0.0405554
                     [ 0.119928, 0.278902]

           WEIGHT     0.0891191  0.0405554
                     [ 9.6e-003, 0.168606]

           DIAMETER   0.3637049  0.0405554
```

```
                          [ 0.284218, 0.443192]

                CH_GIRTH   0.4163462  0.0405554
                          [ 0.336859, 0.495833]

                CH_WIDTH   0.5367337  0.0405554
                          [ 0.457247, 0.616221]
```

```
optn = [ "data"      "cor" ,
         "anal"      "cor" ,
         "nobs"        305 ,
         "meth"       "ml" ,
         "vers"     "semt" ,
         "prio"      "smc" ,
         "heyw"            ,
         "nfac"          2 ,
         "tech"   "nrridg" ,
         "cl"       "wald" ,
         "rvers"    "comm" ,
         "fidi"        "c" ,
         "frot"   "varmax" ,
         "prin"          3 ];
< gof, est > = factor(corr,optn);
```

```
               **********************************
               Maximum Likelihood Factor Analysis
                 (Exploratory Factor Analysis)
               **********************************

  Input Data. . . . . . . . . . . . . . . . .Correlation Matrix
  Analysis of . . . . . . . . . . . . . . . .Correlation Matrix
  Number of Items . . . . . . . . . . . . . . . . .          8
  Number of Factors (NFactor Criterion) . . . . . .          2
  Number of Subjects. . . . . . . . . . . . . . . .        305
  Version . . . . . . . . . . . . . . . . . SEM-Type Matrix Model
  Orthogonal Rotation Method. . . . . .Varimax (Kaiser, 1958)
  Unnormed Rotation: Parameter. . . . . . . . . . . 1.0000000
  ASE for Rotated Solution. . . . . . . . . . . Wald (FinDif)


          Rotated Factor Loadings with Standard Errors
          --------------------------------------------

                            FAC_1                 FAC_2
     HEIGHT     0.8712266  0.0141437  0.2668320  0.0300971
```

```
          [ 0.843506, 0.898948] [ 0.207843, 0.325821]

ARM_SPAN   0.9313923  0.0099879  0.1594936  0.0290540
          [ 0.911816, 0.950968] [ 0.102549, 0.216438]

FOREARM    0.8992982  0.0124617  0.1584067  0.0310294
          [ 0.874874, 0.923723] [ 0.097590, 0.219223]

LOW_LEG    0.8641932  0.0150131  0.2318521  0.0315357
          [ 0.834768, 0.893618] [ 0.170043, 0.293661]

WEIGHT     0.2544715  0.0315743  0.9198506  0.0190010
          [ 0.192587, 0.316356] [ 0.882609, 0.957092]

DIAMETER   0.2121667  0.0392023  0.7689476  0.0264383
          [ 0.135332, 0.289002] [ 0.717129, 0.820766]

CH_GIRTH   0.1513860  0.0413044  0.7488231  0.0284028
          [ 0.070431, 0.232341] [ 0.693155, 0.804492]

CH_WIDTH   0.2919832  0.0437368  0.6148269  0.0362967
          [ 0.206261, 0.377706] [ 0.543687, 0.685967]


                                 U_Var
           HEIGHT     0.1697649  0.0405554
                     [ 0.090278, 0.249252]

           ARM_SPAN   0.1070702  0.0405554
                     [ 0.027583, 0.186557]

           FOREARM    0.1661701  0.0405554
                     [ 0.086683, 0.245657]

           LOW_LEG    0.1994147  0.0405554
                     [ 0.119928, 0.278902]

           WEIGHT     0.0891191  0.0405554
                     [ 9.6e-003, 0.168606]

           DIAMETER   0.3637049  0.0405554
                     [ 0.284218, 0.443192]

           CH_GIRTH   0.4163462  0.0405554
                     [ 0.336859, 0.495833]
```

```
                CH_WIDTH   0.5367337  0.0405554
                          [ 0.457247, 0.616221]
```

3. Covariance (UNStand.) UNNormal. Oblique VARIMAX:

```
optn = [ "data"      "cor" ,
         "anal"      "cov" ,
         "nobs"        305 ,
         "meth"       "ml" ,
         "vers"     "semt" ,
         "prio"      "smc" ,
         "heyw"            ,
         "nfac"          2 ,
         "tech"   "nrridg" ,
         "cl"       "wald" ,
         "rvers"    "comm" ,
         "frot"   "varmax" ,
         "obl"             ,
         "prin"          3 ];
< gof, est > = factor(corr,optn);
```

```
               **********************************
               Maximum Likelihood Factor Analysis
                 (Exploratory Factor Analysis)
               **********************************

  Input Data. . . . . . . . . . . . . . . . .Correlation Matrix
  Analysis of . . . . . . . . . . . . . . . . Covariance Matrix
  Number of Items . . . . . . . . . . . . . . . . . .         8
  Number of Factors (NFactor Criterion) . . . . . .           2
  Number of Subjects. . . . . . . . . . . . . . . . .       305
  Version . . . . . . . . . . . . . . . . . SEM-Type Matrix Model
  Oblique Rotation Method . . . . . . . .Varimax (Kaiser, 1958)
  Unnormed Rotation: Parameter. . . . . . . . . . . . 1.0000000
  ASE for Rotated Solution. . . . . . . . . . .Wald (Analyt.)


        Rotated Factor Loadings with Standard Errors
        --------------------------------------------

                           FAC_1                 FAC_2
   HEIGHT     0.8665651  0.0391719  0.1163976  0.0294724
             [ 0.789790, 0.943341] [ 0.058633, 0.174162]
```

```
ARM_SPAN   0.9470597  0.0399299 -0.0065359  0.0256653
          [ 0.868799, 1.025321] [-0.056839, 0.043767]

FOREARM    0.9137018  0.0413638 -0.0017191  0.0280751
          [ 0.832630, 0.994773] [-0.056745, 0.053307]

LOW_LEG    0.8649593  0.0407346  0.0812733  0.0303856
          [ 0.785121, 0.944798] [ 0.021719, 0.140828]

WEIGHT     0.1148670  0.0241759  0.9107634  0.0451476
          [ 0.067483, 0.162251] [ 0.822276, 0.999251]

DIAMETER   0.0954394  0.0334479  0.7614546  0.0497878
          [ 0.029883, 0.160996] [ 0.663872, 0.859037]

CH_GIRTH   0.0352317  0.0352612  0.7517627  0.0510242
          [-0.033879, 0.104342] [ 0.651757, 0.851768]

CH_WIDTH   0.2041496  0.0432818  0.5861600  0.0527210
          [ 0.119319, 0.288980] [ 0.482829, 0.689491]


                                 U_Var
           HEIGHT     0.1697649  0.0175229
                     [ 0.135421, 0.204109]

           ARM_SPAN   0.1070702  0.0148859
                     [ 0.077894, 0.136246]

           FOREARM    0.1661701  0.0180289
                     [ 0.130834, 0.201506]

           LOW_LEG    0.1994147  0.0197322
                     [ 0.160740, 0.238089]

           WEIGHT     0.0891191  0.0291079
                     [ 0.032069, 0.146170]

           DIAMETER   0.3637049  0.0362058
                     [ 0.292743, 0.434667]

           CH_GIRTH   0.4163462  0.0398034
                     [ 0.338333, 0.494359]

           CH_WIDTH   0.5367337  0.0462854
                     [ 0.446016, 0.627451]
```

```
                Factor Correlations with Standard Errors
                ----------------------------------------

                                FAC_1                 FAC_2
          FAC_1   1.0000000  0.0000000  0.3259352  0.0420027
                 [ 1.000000, 1.000000] [ 0.243611, 0.408259]

          FAC_2   0.3259352  0.0420027  1.0000000  0.0000000
                 [ 0.243611, 0.408259] [ 1.000000, 1.000000]


                            Factor Structure
                            ----------------

                                  FAC_1                 FAC_2
         HEIGHT     0.9045032  0.0431616  0.3988417  0.0602162
                   [ 0.819908, 0.989098] [ 0.280820, 0.516863]

         ARM_SPAN   0.9449294  0.0432253  0.3021442  0.0619942
                   [ 0.860209, 1.029649] [ 0.180638, 0.423651]

         FOREARM    0.9131414  0.0443032  0.2960884  0.0618693
                   [ 0.826309, 0.999974] [ 0.174827, 0.417350]

         LOW_LEG    0.8914492  0.0440635  0.3631940  0.0606990
                   [ 0.805086, 0.977812] [ 0.244226, 0.482162]

         WEIGHT     0.4117168  0.0467606  0.9482026  0.0453545
                   [ 0.320068, 0.503366] [ 0.859309, 1.037096]

         DIAMETER   0.3436243  0.0501345  0.7925617  0.0493925
                   [ 0.245362, 0.441886] [ 0.695754, 0.889369]

         CH_GIRTH   0.2802576  0.0515099  0.7632460  0.0501906
                   [ 0.179300, 0.381215] [ 0.664874, 0.861618]

         CH_WIDTH   0.3951998  0.0515157  0.6526996  0.0528764
                   [ 0.294231, 0.496169] [ 0.549064, 0.756335]


optn = [ "data"      "cor" ,
         "anal"      "cov" ,
         "nobs"        305 ,
         "meth"       "ml" ,
```

```
         "vers"     "semt" ,
         "prio"      "smc" ,
         "heyw"            ,
         "nfac"          2 ,
         "tech"   "nrridg" ,
         "cl"       "wald" ,
         "rvers"    "comm" ,
         "frot"   "varmax" ,
         "obl"             ,
         "fidi"        "c" ,
         "prin"          3 ];
< gof, est > = factor(corr,optn);
```

```
               **********************************
               Maximum Likelihood Factor Analysis
                 (Exploratory Factor Analysis)
               **********************************

  Input Data. . . . . . . . . . . . . . . . .Correlation Matrix
  Analysis of . . . . . . . . . . . . . . . . Covariance Matrix
  Number of Items . . . . . . . . . . . . . . . .            8
  Number of Factors (NFactor Criterion) . . . . . .          2
  Number of Subjects. . . . . . . . . . . . . . . .        305
  Version . . . . . . . . . . . . . . . . . SEM-Type Matrix Model
  Oblique Rotation Method . . . . . . . .Varimax (Kaiser, 1958)
  Unnormed Rotation: Parameter. . . . . . . . . . . 1.0000000
  ASE for Rotated Solution. . . . . . . . . . . . Wald (FinDif)

          Rotated Factor Loadings with Standard Errors
          --------------------------------------------

                             FAC_1                  FAC_2
     HEIGHT     0.8665651  0.0391215  0.1163976  0.0298072
               [ 0.789888, 0.943242] [ 0.057977, 0.174819]

     ARM_SPAN   0.9470597  0.0398938 -0.0065359  0.0260289
               [ 0.868869, 1.025250] [-0.057552, 0.044480]

     FOREARM    0.9137018  0.0413331 -0.0017191  0.0284134
               [ 0.832690, 0.994713] [-0.057408, 0.053970]

     LOW_LEG    0.8649593  0.0406932  0.0812733  0.0307053
               [ 0.785202, 0.944717] [ 0.021092, 0.141455]

     WEIGHT     0.1148670  0.0243525  0.9107634  0.0452862
```

```
          [ 0.067137, 0.162597] [ 0.822004, 0.999523]

DIAMETER   0.0954394  0.0333746  0.7614546  0.0498565
          [ 0.030026, 0.160853] [ 0.663738, 0.859172]

CH_GIRTH   0.0352317  0.0351579  0.7517627  0.0510650
          [-0.033677, 0.104140] [ 0.651677, 0.851848]

CH_WIDTH   0.2041496  0.0432256  0.5861600  0.0527940
          [ 0.119429, 0.288870] [ 0.482686, 0.689634]


                                 U_Var
           HEIGHT     0.1697649  0.0175229
                     [ 0.135421, 0.204109]

           ARM_SPAN   0.1070702  0.0148859
                     [ 0.077894, 0.136246]

           FOREARM    0.1661701  0.0180289
                     [ 0.130834, 0.201506]

           LOW_LEG    0.1994147  0.0197322
                     [ 0.160740, 0.238089]

           WEIGHT     0.0891191  0.0291079
                     [ 0.032069, 0.146170]

           DIAMETER   0.3637049  0.0362058
                     [ 0.292743, 0.434667]

           CH_GIRTH   0.4163462  0.0398034
                     [ 0.338333, 0.494359]

           CH_WIDTH   0.5367337  0.0462854
                     [ 0.446016, 0.627451]

       Factor Correlations with Standard Errors
       ----------------------------------------

                         FAC_1                 FAC_2
 FAC_1   1.0000000  0.0000000  0.3259352  0.0425414
        [ 1.000000, 1.000000] [ 0.242556, 0.409315]

 FAC_2   0.3259352  0.0425414  1.0000000  0.0000000
        [ 0.242556, 0.409315] [ 1.000000, 1.000000]
```

```
                         Factor Structure
                         ----------------

                              FAC_1               FAC_2
HEIGHT     0.9045032  0.0431349  0.3988417  0.0611188
          [ 0.819960, 0.989046] [ 0.279051, 0.518632]

ARM_SPAN   0.9449294  0.0432263  0.3021442  0.0630045
          [ 0.860207, 1.029652] [ 0.178658, 0.425631]

FOREARM    0.9131414  0.0443035  0.2960884  0.0628295
          [ 0.826308, 0.999975] [ 0.172945, 0.419232]

LOW_LEG    0.8914492  0.0440471  0.3631940  0.0615933
          [ 0.805118, 0.977780] [ 0.242473, 0.483915]

WEIGHT     0.4117168  0.0462543  0.9482026  0.0454053
          [ 0.321060, 0.502374] [ 0.859210, 1.037195]

DIAMETER   0.3436243  0.0497604  0.7925617  0.0494233
          [ 0.246096, 0.441153] [ 0.695694, 0.889430]

CH_GIRTH   0.2802576  0.0511508  0.7632460  0.0501978
          [ 0.180004, 0.380511] [ 0.664860, 0.861632]

CH_WIDTH   0.3951998  0.0512847  0.6526996  0.0529644
          [ 0.294684, 0.495716] [ 0.548891, 0.756508]
```

4. Correlation (Standard.) UNNormal. Oblique VARIMAX:

```
optn = [ "data"      "cor" ,
         "anal"      "cor" ,
         "nobs"        305 ,
         "meth"       "ml" ,
         "vers"     "semt" ,
         "prio"      "smc" ,
         "heyw"            ,
         "nfac"          2 ,
         "tech"   "nrridg" ,
         "cl"       "wald" ,
         "rvers"    "comm" ,
         "frot"   "varmax" ,
         "obl"             ,
```

```
        "fidi"        "c" ,
        "prin"          3 ];
< gof, est > = factor(corr,optn);

              *********************************
              Maximum Likelihood Factor Analysis
                (Exploratory Factor Analysis)
              *********************************

  Input Data. . . . . . . . . . . . . . . . .Correlation Matrix
  Analysis of . . . . . . . . . . . . . . . .Correlation Matrix
  Number of Items . . . . . . . . . . . . . . . . .         8
  Number of Factors (NFactor Criterion) . . . . . .         2
  Number of Subjects. . . . . . . . . . . . . . . .       305
  Version . . . . . . . . . . . . . . . . SEM-Type Matrix Model
  Oblique Rotation Method . . . . . . .Varimax (Kaiser, 1958)
  Unnormed Rotation: Parameter. . . . . . . . . . . 1.0000000
  ASE for Rotated Solution. . . . . . . . . . .Wald (Analyt.)

          Rotated Factor Loadings with Standard Errors
          --------------------------------------------

                          FAC_1                   FAC_2
     HEIGHT     0.8665651  0.0168243  0.1163976  0.0257846
               [ 0.833590, 0.899540] [ 0.065861, 0.166934]

     ARM_SPAN   0.9470597  0.0111557 -0.0065359  0.0222945
               [ 0.925195, 0.968925] [-0.050232, 0.037161]

     FOREARM    0.9137018  0.0137099 -0.0017191  0.0252029
               [ 0.886831, 0.940573] [-0.051116, 0.047678]

     LOW_LEG    0.8649593  0.0172310  0.0812733  0.0272056
               [ 0.831187, 0.898731] [ 0.027951, 0.134595]

     WEIGHT     0.1148670  0.0218171  0.9107634  0.0208081
               [ 0.072106, 0.157628] [ 0.869980, 0.951547]

     DIAMETER   0.0954394  0.0321562  0.7614546  0.0285576
               [ 0.032414, 0.158464] [ 0.705483, 0.817427]

     CH_GIRTH   0.0352317  0.0338198  0.7517627  0.0301565
               [-0.031054, 0.101517] [ 0.692657, 0.810868]

     CH_WIDTH   0.2041496  0.0424601  0.5861600  0.0396482
```

```
             [ 0.120929, 0.287370] [ 0.508451, 0.663869]

                                    U_Var
              HEIGHT      0.1697649  0.0405554
                         [ 0.090278, 0.249252]

              ARM_SPAN    0.1070702  0.0405554
                         [ 0.027583, 0.186557]

              FOREARM     0.1661701  0.0405554
                         [ 0.086683, 0.245657]

              LOW_LEG     0.1994147  0.0405554
                         [ 0.119928, 0.278902]

              WEIGHT      0.0891191  0.0405554
                         [ 9.6e-003, 0.168606]

              DIAMETER    0.3637049  0.0405554
                         [ 0.284218, 0.443192]

              CH_GIRTH    0.4163462  0.0405554
                         [ 0.336859, 0.495833]

              CH_WIDTH    0.5367337  0.0405554
                         [ 0.457247, 0.616221]

            Factor Correlations with Standard Errors
            ----------------------------------------

                          FAC_1                 FAC_2
      FAC_1   1.0000000  0.0000000  0.3259352  0.0400536
             [ 1.000000, 1.000000] [ 0.247432, 0.404439]

      FAC_2   0.3259352  0.0400536  1.0000000  0.0000000
             [ 0.247432, 0.404439] [ 1.000000, 1.000000]

                        Factor Structure
                        ----------------

                            FAC_1                 FAC_2
    HEIGHT      0.9045032  0.0117189  0.3988417  0.0475449
               [ 0.881535, 0.927472] [ 0.305655, 0.492028]
```

```
ARM_SPAN   0.9449294  0.0088729  0.3021442  0.0515997
          [ 0.927539, 0.962320] [ 0.201011, 0.403278]

FOREARM    0.9131414  0.0115709  0.2960884  0.0520719
          [ 0.890463, 0.935820] [ 0.194029, 0.398148]

LOW_LEG    0.8914492  0.0131606  0.3631940  0.0493346
          [ 0.865655, 0.917244] [ 0.266500, 0.459888]

WEIGHT     0.4117168  0.0393174  0.9482026  0.0166195
          [ 0.334656, 0.488778] [ 0.915629, 0.980776]

DIAMETER   0.3436243  0.0450198  0.7925617  0.0250918
          [ 0.255387, 0.431861] [ 0.743383, 0.841741]

CH_GIRTH   0.2802576  0.0478013  0.7632460  0.0276188
          [ 0.186569, 0.373946] [ 0.709114, 0.817378]

CH_WIDTH   0.3951998  0.0454400  0.6526996  0.0350593
          [ 0.306139, 0.484261] [ 0.583985, 0.721415]
```

5. Covariance (UNStand.) UNNormal. Orthog. SIMPLIMAX:

```
optn = [ "data"      "cor" ,
         "anal"      "cov" ,
         "nobs"        305 ,
         "meth"       "ml" ,
         "vers"     "semt" ,
         "prio"      "smc" ,
         "heyw"            ,
         "nfac"          2 ,
         "tech"   "nrridg" ,
         "cl"       "wald" ,
         "rvers"    "jenn" ,
         "frot"   "simmax" ,
         "rotp"          8 ,
         "ort"             ,
         "fidi"        "c" ,
         "phis"            ,
         "prin"          3 ];
< gof, est > = factor(corr,optn);
```

```
          **********************************
          Maximum Likelihood Factor Analysis
            (Exploratory Factor Analysis)
```

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

Input Data. . . . . . . . . . . . . . . . .Correlation Matrix
Analysis of . . . . . . . . . . . . . . . . Covariance Matrix
Number of Items . . . . . . . . . . . . . . . . . .         8
Number of Factors (NFactor Criterion) . . . . . .           2
Number of Subjects. . . . . . . . . . . . . . . . .       305
Version . . . . . . . . . . . . . . . . SEM-Type Matrix Model
Orthogonal Rotation Method. . . . . Simplimax (Kiers, 1994)
Unnormed Rotation: Parameter. . . . . . . . . . . . 8.0000000
ASE for Rotated Solution. . . . . . . . . . . Wald (FinDif)

        Rotated Factor Loadings with Standard Errors
        --------------------------------------------

                             FAC_1                  FAC_2
   HEIGHT     0.8729492  0.0442534  0.2611412  0.0336199
             [ 0.786214, 0.959684] [ 0.195247, 0.327035]

   ARM_SPAN   0.9324132  0.0431556  0.1534125  0.0310791
             [ 0.847830, 1.016997] [ 0.092499, 0.214326]

   FOREARM    0.9003127  0.0443221  0.1525351  0.0327716
             [ 0.813443, 0.987183] [ 0.088304, 0.216766]

   LOW_LEG    0.8656877  0.0449078  0.2262080  0.0342187
             [ 0.777670, 0.953705] [ 0.159141, 0.293275]

   WEIGHT     0.2604685  0.0372572  0.9181705  0.0447987
             [ 0.187446, 0.333491] [ 0.830367, 1.005974]

   DIAMETER   0.2171798  0.0424197  0.7675467  0.0493917
             [ 0.134039, 0.300321] [ 0.670741, 0.864353]

   CH_GIRTH   0.1562691  0.0433846  0.7478193  0.0504320
             [ 0.071237, 0.241301] [ 0.648974, 0.846664]

   CH_WIDTH   0.2959890  0.0470669  0.6129085  0.0521255
             [ 0.203740, 0.388238] [ 0.510744, 0.715073]

                                        U_Var
              HEIGHT     0.1697649  0.0175229
                        [ 0.135421, 0.204109]

              ARM_SPAN   0.1070702  0.0148859
```

```
                    [ 0.077894, 0.136246]

          FOREARM    0.1661701  0.0180289
                    [ 0.130834, 0.201506]

          LOW_LEG    0.1994147  0.0197322
                    [ 0.160740, 0.238089]

          WEIGHT     0.0891191  0.0291079
                    [ 0.032069, 0.146170]

          DIAMETER   0.3637049  0.0362058
                    [ 0.292743, 0.434667]

          CH_GIRTH   0.4163462  0.0398034
                    [ 0.338333, 0.494359]

          CH_WIDTH   0.5367337  0.0462854
                    [ 0.446016, 0.627451]
```

6. Correlation (Standard.) UNNormal. Orthog. SIMPLIMAX:

```
optn = [ "data"      "cor" ,
         "anal"      "cor" ,
         "nobs"        305 ,
         "meth"       "ml" ,
         "vers"     "semt" ,
         "prio"      "smc" ,
         "heyw"            ,
         "nfac"          2 ,
         "tech"   "nrridg" ,
         "cl"       "wald" ,
         "rvers"    "jenn" ,
         "frot"   "simmax" ,
         "rotp"          8 ,
         "ort"             ,
         "fidi"        "c" ,
         "phis"            ,
         "prin"          3 ];
< gof, est > = factor(corr,optn);
```

```
                 *********************************
                 Maximum Likelihood Factor Analysis
                   (Exploratory Factor Analysis)
                 *********************************
```

```
Input Data. . . . . . . . . . . . . . . . .Correlation Matrix
Analysis of . . . . . . . . . . . . . . . .Correlation Matrix
Number of Items . . . . . . . . . . . . . . . . . .         8
Number of Factors (NFactor Criterion) . . . . . .         2
Number of Subjects. . . . . . . . . . . . . . . . .       305
Version . . . . . . . . . . . . . . . . SEM-Type Matrix Model
Orthogonal Rotation Method. . . . . Simplimax (Kiers, 1994)
Unnormed Rotation: Parameter. . . . . . . . . . . 8.0000000
ASE for Rotated Solution. . . . . . . . . . . Wald (FinDif)

        Rotated Factor Loadings with Standard Errors
        --------------------------------------------

                             FAC_1                 FAC_2
   HEIGHT     0.8729492  0.0138241  0.2611412  0.0293193
             [ 0.845855, 0.900044] [ 0.203676, 0.318606]

   ARM_SPAN   0.9324132  0.0099057  0.1534125  0.0290041
             [ 0.912998, 0.951828] [ 0.096565, 0.210260]

   FOREARM    0.9003127  0.0123842  0.1525351  0.0308976
             [ 0.876040, 0.924585] [ 0.091977, 0.213093]

   LOW_LEG    0.8656877  0.0147915  0.2262080  0.0308785
             [ 0.836697, 0.894678] [ 0.165687, 0.286729]

   WEIGHT     0.2604685  0.0324764  0.9181705  0.0190410
             [ 0.196816, 0.324121] [ 0.880851, 0.955490]

   DIAMETER   0.2171798  0.0396368  0.7675467  0.0266840
             [ 0.139493, 0.294867] [ 0.715247, 0.819846]

   CH_GIRTH   0.1562691  0.0418593  0.7478193  0.0286421
             [ 0.074226, 0.238312] [ 0.691682, 0.803957]

   CH_WIDTH   0.2959890  0.0429156  0.6129085  0.0365067
             [ 0.211876, 0.380102] [ 0.541357, 0.684460]


                                   U_Var
              HEIGHT     0.1697649  0.0405554
                        [ 0.090278, 0.249252]

              ARM_SPAN   0.1070702  0.0405554
```

```
                     [ 0.027583, 0.186557]

          FOREARM     0.1661701  0.0405554
                     [ 0.086683, 0.245657]

          LOW_LEG     0.1994147  0.0405554
                     [ 0.119928, 0.278902]

          WEIGHT      0.0891191  0.0405554
                     [ 9.6e-003, 0.168606]

          DIAMETER    0.3637049  0.0405554
                     [ 0.284218, 0.443192]

          CH_GIRTH    0.4163462  0.0405554
                     [ 0.336859, 0.495833]

          CH_WIDTH    0.5367337  0.0405554
                     [ 0.457247, 0.616221]
```

7. Covariance (UNStand.) UNNormal. Oblique SIMPLIMAX:

```
optn = [ "data"      "cor" ,
         "anal"      "cov" ,
         "nobs"        305 ,
         "meth"       "ml" ,
         "vers"     "semt" ,
         "prio"      "smc" ,
         "heyw"            ,
         "nfac"          2 ,
         "tech"   "nrridg" ,
         "cl"       "wald" ,
         "rvers"    "jenn" ,
         "frot"   "simmax" ,
         "rotp"          8 ,
         "obl"             ,
         "fidi"        "c" ,
         "phis"            ,
         "prin"          3 ];
< gof, est > = factor(corr,optn);
```

```
               *********************************
               Maximum Likelihood Factor Analysis
                 (Exploratory Factor Analysis)
               *********************************
```

```
Input Data. . . . . . . . . . . . . . . . .Correlation Matrix
Analysis of . . . . . . . . . . . . . . . . Covariance Matrix
Number of Items . . . . . . . . . . . . . . . . . .         8
Number of Factors (NFactor Criterion) . . . . . .           2
Number of Subjects. . . . . . . . . . . . . . . . .       305
Version . . . . . . . . . . . . . . . . SEM-Type Matrix Model
Oblique Rotation Method . . . . . . Simplimax (Kiers, 1994)
Unnormed Rotation: Parameter. . . . . . . . . . . 8.0000000
ASE for Rotated Solution. . . . . . . . . . . Wald (FinDif)

        Rotated Factor Loadings with Standard Errors
        --------------------------------------------

                             FAC_1                 FAC_2
   HEIGHT     0.8701877  0.0448571  0.0799964  0.0252513
             [ 0.782269, 0.958106] [ 0.030505, 0.129488]

   ARM_SPAN   0.9697016  0.0452501 -0.0542468  0.0209397
             [ 0.881013, 1.058390] [-0.095288,-0.013206]

   FOREARM    0.9349055  0.0465094 -0.0474777  0.0238682
             [ 0.843749, 1.026062] [-0.094259,-7.0e-004]

   LOW_LEG    0.8734516  0.0462013  0.0428702  0.0264398
             [ 0.782899, 0.964005] [-9.0e-003, 0.094691]

   WEIGHT    -0.0097215  0.0264140  0.9590137  0.0480592
             [-0.061492, 0.042049] [ 0.864819, 1.053208]

   DIAMETER  -0.0087380  0.0347440  0.8018247  0.0527139
             [-0.076835, 0.059359] [ 0.698507, 0.905142]

   CH_GIRTH  -0.0689732  0.0361491  0.7945667  0.0541488
             [-0.139824, 1.9e-003] [ 0.688437, 0.900696]

   CH_WIDTH   0.1269532  0.0453483  0.6107065  0.0559698
             [ 0.038072, 0.215834] [ 0.501008, 0.720405]


                                        U_Var
              HEIGHT     0.1697649  0.0175229
                        [ 0.135421, 0.204109]

              ARM_SPAN   0.1070702  0.0148859
                        [ 0.077894, 0.136246]
```

```
FOREARM    0.1661701  0.0180289
          [ 0.130834, 0.201506]

LOW_LEG    0.1994147  0.0197322
          [ 0.160740, 0.238089]

WEIGHT     0.0891191  0.0291079
          [ 0.032069, 0.146170]

DIAMETER   0.3637049  0.0362058
          [ 0.292743, 0.434667]

CH_GIRTH   0.4163462  0.0398034
          [ 0.338333, 0.494359]

CH_WIDTH   0.5367337  0.0462854
          [ 0.446016, 0.627451]
```

```
    Factor Correlations with Standard Errors
    ----------------------------------------

                     FAC_1                 FAC_2
FAC_1   1.0000000  0.0000000  0.4784319  0.0484444
       [ 1.000000, 1.000000] [ 0.383483, 0.573381]

FAC_2   0.4784319  0.0484444  1.0000000  0.0000000
       [ 0.383483, 0.573381] [ 1.000000, 1.000000]
```

```
                Factor Structure
                ----------------

                       FAC_1                 FAC_2
HEIGHT      0.9084605  0.0444015  0.4963219  0.0558471
           [ 0.821435, 0.995486] [ 0.386864, 0.605780]

ARM_SPAN    0.9437482  0.0431838  0.4096894  0.0576528
           [ 0.859110, 1.028387] [ 0.296692, 0.522687]

FOREARM     0.9121906  0.0443101  0.3998109  0.0577414
           [ 0.825345, 0.999037] [ 0.286640, 0.512982]

LOW_LEG     0.8939620  0.0449089  0.4607573  0.0564958
           [ 0.805942, 0.981982] [ 0.350028, 0.571487]
```

```
WEIGHT     0.4491012  0.0553706  0.9543626  0.0448059
          [ 0.340577, 0.557626] [ 0.866545, 1.042181]

DIAMETER   0.3748805  0.0563848  0.7976442  0.0491406
          [ 0.264368, 0.485393] [ 0.701330, 0.893958]

CH_GIRTH   0.3111728  0.0573476  0.7615677  0.0501945
          [ 0.198774, 0.423572] [ 0.663188, 0.859947]

CH_WIDTH   0.4191347  0.0552409  0.6714449  0.0522279
          [ 0.310864, 0.527405] [ 0.569080, 0.773810]
```

8. Correlation (Stand.) UNNormal. Oblique SIMPLIMAX:

```
optn = [ "data"      "cor" ,
         "anal"      "cor" ,
         "nobs"        305 ,
         "meth"       "ml" ,
         "vers"     "semt" ,
         "prio"      "smc" ,
         "heyw"            ,
         "nfac"          2 ,
         "tech"   "nrridg" ,
         "cl"       "wald" ,
         "rvers"    "jenn" ,
         "frot"   "simmax" ,
         "rotp"          8 ,
         "obl"             ,
         "fidi"        "c" ,
         "phis"            ,
         "prin"          3 ];
< gof, est > = factor(corr,optn);
```

```
               *********************************
               Maximum Likelihood Factor Analysis
                 (Exploratory Factor Analysis)
               *********************************

  Input Data. . . . . . . . . . . . . . . . .Correlation Matrix
  Analysis of . . . . . . . . . . . . . . . .Correlation Matrix
  Number of Items . . . . . . . . . . . . . . . . . .         8
  Number of Factors (NFactor Criterion) . . . . . .         2
  Number of Subjects. . . . . . . . . . . . . . . . .       305
  Version . . . . . . . . . . . . . . . . . SEM-Type Matrix Model
```

```
Oblique Rotation Method . . . . . . Simplimax (Kiers, 1994)
Unnormed Rotation: Parameter. . . . . . . . . . . 8.0000000
ASE for Rotated Solution. . . . . . . . . . . Wald (FinDif)

        Rotated Factor Loadings with Standard Errors
        --------------------------------------------

                          FAC_1                 FAC_2
   HEIGHT     0.8701877  0.0185070  0.0799964  0.0252409
             [ 0.833915, 0.906461] [ 0.030525, 0.129468]

   ARM_SPAN   0.9697016  0.0138460 -0.0542468  0.0209601
             [ 0.942564, 0.996839] [-0.095328,-0.013166]

   FOREARM    0.9349055  0.0164221 -0.0474777  0.0238676
             [ 0.902719, 0.967092] [-0.094257,-7.0e-004]

   LOW_LEG    0.8734516  0.0193746  0.0428702  0.0264145
             [ 0.835478, 0.911425] [-8.9e-003, 0.094642]

   WEIGHT    -0.0097215  0.0262218  0.9590137  0.0230930
             [-0.061115, 0.041672] [ 0.913752, 1.004275]

   DIAMETER  -0.0087380  0.0346419  0.8018247  0.0309988
             [-0.076635, 0.059159] [ 0.741068, 0.862581]

   CH_GIRTH  -0.0689732  0.0360589  0.7945667  0.0329026
             [-0.139647, 1.7e-003] [ 0.730079, 0.859055]

   CH_WIDTH   0.1269532  0.0450999  0.6107065  0.0427735
             [ 0.038559, 0.215347] [ 0.526872, 0.694541]


                                    U_Var
              HEIGHT     0.1697649  0.0405554
                        [ 0.090278, 0.249252]

              ARM_SPAN   0.1070702  0.0405554
                        [ 0.027583, 0.186557]

              FOREARM    0.1661701  0.0405554
                        [ 0.086683, 0.245657]

              LOW_LEG    0.1994147  0.0405554
                        [ 0.119928, 0.278902]
```

```
WEIGHT     0.0891191  0.0405554
          [ 9.6e-003, 0.168606]

DIAMETER   0.3637049  0.0405554
          [ 0.284218, 0.443192]

CH_GIRTH   0.4163462  0.0405554
          [ 0.336859, 0.495833]

CH_WIDTH   0.5367337  0.0405554
          [ 0.457247, 0.616221]
```

```
    Factor Correlations with Standard Errors
    ----------------------------------------

                     FAC_1                 FAC_2
 FAC_1   1.0000000  0.0000000  0.4784319  0.0480774
        [ 1.000000, 1.000000] [ 0.384202, 0.572662]

 FAC_2   0.4784319  0.0480774  1.0000000  0.0000000
        [ 0.384202, 0.572662] [ 1.000000, 1.000000]
```

```
                Factor Structure
                ----------------

                       FAC_1                 FAC_2
HEIGHT     0.9084605  0.0117344  0.4963219  0.0457528
          [ 0.885461, 0.931460] [ 0.406648, 0.585996]

ARM_SPAN   0.9437482  0.0088421  0.4096894  0.0507042
          [ 0.926418, 0.961078] [ 0.310311, 0.509068]

FOREARM    0.9121906  0.0115667  0.3998109  0.0511214
          [ 0.889520, 0.934861] [ 0.299615, 0.500007]

LOW_LEG    0.8939620  0.0131842  0.4607573  0.0477599
          [ 0.868122, 0.919802] [ 0.367149, 0.554365]

WEIGHT     0.4491012  0.0467569  0.9543626  0.0155619
          [ 0.357459, 0.540743] [ 0.923862, 0.984863]

DIAMETER   0.3748805  0.0503732  0.7976442  0.0246346
          [ 0.276151, 0.473610] [ 0.749361, 0.845927]
```

```
CH_GIRTH   0.3111728  0.0531869  0.7615677  0.0275268
         [ 0.206928, 0.415417] [ 0.707616, 0.815519]

CH_WIDTH   0.4191347  0.0477870  0.6714449  0.0339897
         [ 0.325474, 0.512796] [ 0.604826, 0.738063]
```

2.6 New Functions

New functions are implemented:

generead not finished yet.

lrforw linear regression with $R2$ forward selection for data with many variables

pca computes the common principal component analysis for large scale applications (very large nvar or nobs) with normal ASEs for unrotated and rotated component loadings

randisc implements some generators for discrete random variates (Marsaglia, Tsang, & Wang, 2004).

scalpha computes the sample coefficient alpha with normal and nonnormal ASE and CIs

screetst scree test for the number of nonzero eigenvalues of a covariance matrix

split regression trees with binary response (target) for very large data sets

varclus cluster analyis of variables similar to PROC VARCLUS in SAS/STAT (Harman, 1976).

3 New Developments

3.1 Function `generead`

Not finished yet.

3.2 Function `lrforw`

```
<gof,parm,yptr,yptt> = lrforw(trn,model,optn<,class<,aov16<,grpv<,tst>>>>)
```

Purpose: The `lrforw` function implements a forward stepwise linear regression algorithm. The univariate response should be either interval or binary. The algorithm is based on Miller (2002) and Foster & Stine (2004). There are three major parts of the algorithm:

1. Ranking all predictor variables (including the derived AOV16 and Group variables by descending univariate R^2 value.
2. A stepwise forward selection algorithm with the linear model. At this time only two selection criteria are implemented, the R^2 and the χ^2 value. Other criteria such as information criteria and Mallows Cp will be added in the near future.
3. Only when the response is a binary variable, an approximate logistic model selection (with the estimation of two additional parameters) which results in binary predicted values with a 2 by 2 classification table. Note that the two parameters of the logistic model are estimated for given (fixed) linear coefficients resulting from step 2.

There are two types of algorithms for the stepwise forward selection implemented:

1. a fast algorithm that first stores the entire $n \times n$ matrix $\mathbf{X}^T\mathbf{X}$ incore and sweeps the entire matrix.
2. a slightly slower algorithm which usees only the $nr \times n$ part of the $\mathbf{X}^T\mathbf{X}$ matrix where nr is the number of rows which corresponds to selected effects.

For $nr \ll n$ the second algorithm can enable many applications in chemometrics and analysis of microarray data which cannnot be solved by the first algorithm.

Input: **trn** the $N \times M$ input data set where the M columns contain the $N \times n$ predictor matrix $\mathbf{X}$ and the N response vector $\mathbf{Y}$. Which data columns are selected for $\mathbf{X}$ and $\mathbf{Y}$ must be specified with the model string. Before computing the *garotte* estimates, the data $(\mathbf{X}, Y)$ will be mean centered.

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 : The option argument is specified in form of a simple vector.

Index	Meaning
[1]	amount of printed output
[2]	lower R2 threshold for selection this raw (unpartial) R^2 value from first step: only variables (effects) with larger R^2 values are candidates for selection in step 2.
[3]	lower threshold for R^2 or upper threshold for χ^2 for terminating the selection process; the R^2 or χ^2 value is now based on the partial X'X matrix.
[4]	upper range for number of rows in $\mathbf{X}^T\mathbf{X}$ matrix (def=4000);
[5]	upper threshhold for number selected vars in step 2; another criterion for terminating the selction process.
[6]	cutoff for accuracy table (def=.5); valid only for binary response.
[7]	=1: do not use AOV vars (def=1)
[8]	=1: do not use GROUP vars (def=1)
[9]	=0: use R2 selection criterion, =1: use chisquare criterion
[10]	print observationwise stat: predicted y, residuals.

If the number of rows needed for storing the $\mathbf{X}^T\mathbf{X}$ is larger than the value specified in `optn[4]` the fast first (default) algorithm switches to the slightly slower but more memory efficient second algorithm.

class : This optional argument should be an integer scalar or vector of integer scalars naming the number of columns which are considered categorical (nominal scaled) variables.

aov16 a list of column numbers of (non Class) predictor variables which are made to ordinal variables with 16 categories

grpv a list of column numbers of multinomial Class variables

test specifies a matrix of test data which is not used for modeling but for predicted values are computed (scored) that can be returned with the last output argument `yptt`.

Output: **gof** a vector of goodness-of-fit indicators

parm the parameter vector for the linear model

yptr the precited values $\hat{y}_i$ of the training data

yptt the precited values $\hat{y}_i$ of the test data

Restrictions: 1. Missing values in the data are replaced by the variable means.

2. The response cannot be multinomial.

Relationships: reg(), glim(), varsel()

Examples: 1. Australian: nobs=690, nvar=15:

```
form = fo = " %g";
for (j = 2; j <= 15; j++) form = strcat(form,fo);
fid = fopen("..\\tdata\\australian.dat","r");
data = fscanf(fid,form,.,15);
nr = nrow(data);
print "Observations of Australian.dat:",nr;
x = data[,2:15]; y = data[,1];
modl = "1 = 2 : 15";
clas = [ 1 8 9 10 11 15 ];

/* use R2 */
optn = cons(10,1,.); optn[1]= 3;
< gof,parm,yptr > = lrforw(data,modl,optn,clas);
```

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

                 Number Valid Observations  690
                 Response Variable         Y[1]
                 N Independend Variables     14
                 Number Parameters           42
                 Design Coding:  Rank-Deficient
                 Use R2 Selection Criterion
                 Without AOV16 Variables
                 Without Group Variables
                 Max Selection Steps          0
                 Lower R2 Threshold   1.00e-005
                 Lower Threshhold for 5.00e-005
                 Max Variables in Model    3000

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

Intercept + X2  + X3  + X4  + X5  + X6  + X7  + C8  + C9  + C10 +
```

```
C11 + X12 + X13 + X14 + C15

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

Class  Level    Value

Y[ 1]      2        0       1
C[ 8]      2        0       1
C[ 9]      2        0       1
C[10]     23        0       1       2       3       4
                    5       6       7       8       9
                   10      11      12      13      14
                   15      16      17      19      20
                   23      40      67
C[11]      2        0       1
C[15]      2        0       1

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

Column   Nobs        Mean     Std Dev    Skewness    Kurtosis
X[ 2]     690   31.568203   11.853273   1.1559350   1.1920586
X[ 3]     690   4.7587246   4.9781632   1.4888131   2.2740217
X[ 4]     690   1.7666667   0.4300628  -1.1534550  -0.3382696
X[ 5]     690   7.3724638   3.6832648  -0.0691905  -0.8490426
X[ 6]     690   4.6927536   1.9923161   0.4684118  -0.1781323
X[ 7]     690   2.2234058   3.3465134   2.8913304   11.200192
X[12]     690   1.9289855   0.2988131  -1.9447255   6.7188876
X[13]     690   184.01449   172.15927   2.7499117   19.926698
X[14]     690   1018.3855   5210.1026   13.140655   214.66997

         ********************************
         Rank Order of Rsquare of Effects
         ********************************

       Rank     Effect     Rsquared        df
          1      9 C10  0.030637506        22
          2      6  X7  0.009410823         1
          3      3  X4  0.004035827         1
          4      8  C9  0.004001295         1
          5     11 X12  0.003588089         1
```

6	12	X13	0.003433426	1
7	10	C11	0.002914307	1
8	5	X6	0.002772838	1
9	1	X2	0.001343905	1
10	2	X3	0.000793343	1
11	14	C15	0.000193129	1
12	4	X5	2.8317e-005	1
13	13	X14	1.3073e-005	1
14	7	C8	8.4314e-007	1

****
History of Forward Variable Selection
****

Step	Effect	DF	Rsquared	F_value	pval	TotSSQ	BIC
1	C10	22	0.0306	0.9582313	0.5168	4.6132092	-1020.0872
2	X7	1	0.0087	6.0379591	0.0143	1.3113913	-1024.1649
3	X4	1	0.0041	2.8604514	0.0913	0.6195341	-1024.9703
4	X13	1	0.0036	2.4799784	0.1158	0.5359362	-1025.3799
5	X12	1	0.0018	1.2545402	0.2631	0.2710088	-1024.5150
6	X3	1	0.0013	0.8817743	0.3481	0.1905169	-1023.2577
7	X6	1	0.0013	0.8949754	0.3445	0.1933999	-1022.0090
8	X5	1	0.0004	0.2458114	0.6202	0.0531793	-1020.0769
9	C11	1	0.0003	0.2273436	0.6337	0.0492416	-1015.9170
10	X14	1	0.0003	0.1823700	0.6695	0.0395496	-1013.8992
11	X2	1	0.0001	0.0928957	0.7606	0.0201736	-1011.7811

****
Final ANOVA Table for Target: Y[1]
****

Effect	DF	Rsquared	Sum-of-Squares
Model	32	0.05244694	7.897140596121
Error	657	.	142.6767724474
Total	689	.	150.5739130435

MSE	0.21782713	SBC	-1316.30001
AIC	-1017.51580	BIC	-1011.78105
F	1.13294262	Prb	0.28321169
CP	35.0000000	AR2	0.00326098

```
**********************************
Effects not chosen for Target: Y[1]
**********************************
```

Effect	DF	Rsquared	F_value	pval	Sum-of-Squares
C8	1	8.431e-007	0.00000000	0.0000	0.000126954516
C9	0	0.00000000	0.00000000	1.0000	0.000000000000
C15	1	6.630e-006	0.00458971	0.9460	0.000998231300

```
*****************************************
History of 2 Parameter Logistic Estimation
*****************************************
```

Iter	Alpha	Beta	Fiterror
1	-2.00247109	4.00372883	20.0397350
2	-2.58606133	4.96213985	1.25912926
3	-2.62803482	5.02840933	0.00615342
4	-2.62820382	5.02867320	9.819e-008

Logistic Regression: Intercept=-2.6282 Slope=5.02867

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

Index	Value	StdErr
Absolute Classification Error	215	.
Classification Accuracy	68.84057971	.
Concordant Pairs	8.339108339	.
Discordant Pairs	2.344652345	.
Tied Pairs	89.31623932	.
Goodman-Kruskal Gamma	0.561081081	0.129591139
Kendall Tau_b	0.135184107	0.041118411
Stuart Tau_c	0.052325142	0.017651916
Somers D C|R	0.059944560	0.020147192

```
Classification Table
--------------------

         |    Predicted
Observed |        0       1
---------|-----------------
```

```
                         0 |       19     203
                         1 |       12     456

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

                 Dense Row Vector (ncol=25)

R |   ModelNpar  SSQ_model  SSQ_error  SSQ_total   DF_model
      41.000000  7.8971406  142.67677  150.57391  32.000000

R |    DF_error   DF_total        MSE    Rsquare        SBC
      657.00000  689.00000  0.2178271  0.0524469 -1316.3000

R |         AIC        BIC     Fvalue      Fprob   MallowCp
     -1017.5158 -1011.7811  1.1329426  0.2832117  35.000000

R |       AdjR2   LogInter   LogSlope        N00        N01
      0.0032610  0.0000000  0.0000000  19.000000  203.00000

R |         N10        N11   ClassErr   Accuracy     unused
      12.000000  456.00000  215.00000  68.840580  0.0000000

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

                 Dense Row Vector (ncol=41)

R |   Intercept         X2         X3         X4         X5
      1.2804019  5.18e-004 -0.0042563 -0.0740905 -0.0023277

R |          X6         X7        C81        C82        C91
      0.0104565  0.0136187  0.0000000  0.0000000  0.0000000

R |         C92       C101       C102       C103       C104
      0.0000000 -0.3875943 -0.4462286 -0.4321149 -0.4070560

R |        C105       C106       C107       C108       C109
     -0.6603249 -0.4567209 -0.4077902 -0.4705896 -0.3328616

R |       C1010      C1011      C1012      C1013      C1014
     -0.2948545 -0.5615332 -0.4018717 -0.3672660 -1.1099980

R |       C1015      C1016      C1017      C1018      C1019
     -0.5965075 -0.3533042 -0.3984886 -0.1112957 -1.0611414
```

```
R |      C1020      C1021      C1022      C1023       C111
    -0.1454050 -1.1975104 -0.3727809  1.30e-014 -0.0174097

R |       C112        X12        X13        X14       C151
    -2.60e-015 -0.0687673  1.20e-004  1.54e-006  0.0000000

R |       C152
     0.0000000
```

```
                *********************************
                Effect Mapping in Parameter Object
                *********************************

                    Effect Type InMod Column Ncol
                        X2    V     Y      1    1
                        X3    V     Y      2    1
                        X4    V     Y      3    1
                        X5    V     Y      4    1
                        X6    V     Y      5    1
                        X7    V     Y      6    1
                        C8    C     N      7    2
                        C9    C     N      9    2
                       C10    C     Y     11   23
                       C11    C     Y     34    2
                       X12    V     Y     36    1
                       X13    V     Y     37    1
                       X14    V     Y     38    1
                       C15    C     N     39    2

                    Total Processing Time: 0
```

```
/* use pvalue of F: optn[2]=r2min, optn[3]=r2stop */
optn = cons(10,1,.);
optn[1]= 3; optn[2] = 1.e-4; optn[3]= .2; optn[9]= 1;
< gof,parm,yptr > = lrforw(data,modl,optn,clas);
```

```
                 *******************************
                 Rank Order of Rsquare of Effects
                 *******************************

               Rank     Effect     Rsquared        df
                  1       9 C10  0.030637506        22
                  2       6  X7  0.009410823         1
```

```
       3      3  X4  0.004035827         1
       4      8  C9  0.004001295         1
       5     11 X12  0.003588089         1
       6     12 X13  0.003433426         1
       7     10 C11  0.002914307         1
       8      5  X6  0.002772838         1
       9      1  X2  0.001343905         1
      10      2  X3  0.000793343         1
      11     14 C15  0.000193129         1
      12      4  X5  2.8317e-005         1
      13     13 X14  1.3073e-005         1
      14      7  C8  8.4314e-007         1

                  *************************************
                  History of Forward Variable Selection
                  *************************************

Step  Effect  DF Rsquared    F_value    pval     TotSSQ        BIC

                    **********************************
                    Final ANOVA Table for Target: Y[1]
                    **********************************

                 Effect    DF    Rsquared  Sum-of-Squares

                  Model     0  0.00000000  0.000000000000
                  Error   689           .   150.5739130435
                  Total   689           .   150.5739130435

                  MSE    0.21853979       SBC -1056.88058
                  AIC   -1048.34389       BIC -1046.34099
                  F            -nan      Prb -1.00000000
                  CP     1.00000000       AR2  0.00000000
```

```
/* with AOV, with GROUP */
optn = cons(10,1,.);
optn[1]= 3; optn[7] = 0; optn[8]= 0;
aov = [ 1:6 ]; grp = 9;
< gof,parm,yptr > = lrforw(data,modl,optn,clas,aov,grp);
```

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

Number Valid Observations  690
Response Variable         Y[1]
N Independend Variables     14
Number Parameters           42
Design Coding:  Rank-Deficient
Use R2 Selection Criterion
With 6 AOV16 Variables
With 1 Group Variables
Max Selection Steps          0
Lower R2 Threshold   1.00e-005
Lower Threshhold for 5.00e-005
Max Variables in Model    3000


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

Intercept + X2  + X3  + X4  + X5  + X6  + X7  + C8  + C9  + C10 +
C11 + X12 + X13 + X14 + C15


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

Class  Level    Value

Y[ 1]      2        0       1
C[ 8]      2        0       1
C[ 9]      2        0       1
C[10]     23        0       1       2       3       4
                    5       6       7       8       9
                   10      11      12      13      14
                   15      16      17      19      20
                   23      40      67
C[11]      2        0       1
C[15]      2        0       1


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

Column	Nobs	Mean	Std Dev	Skewness	Kurtosis
X[2]	690	31.568203	11.853273	1.1559350	1.1920586
X[3]	690	4.7587246	4.9781632	1.4888131	2.2740217
X[4]	690	1.7666667	0.4300628	-1.1534550	-0.3382696
X[5]	690	7.3724638	3.6832648	-0.0691905	-0.8490426
X[6]	690	4.6927536	1.9923161	0.4684118	-0.1781323
X[7]	690	2.2234058	3.3465134	2.8913304	11.200192
X[12]	690	1.9289855	0.2988131	-1.9447255	6.7188876
X[13]	690	184.01449	172.15927	2.7499117	19.926698
X[14]	690	1018.3855	5210.1026	13.140655	214.66997

```
********************************
Remap Indices of Group Variables
********************************

 Effect  Nlev Levels...
C10_GRP     1 1 1 1 1 1 1 1 1 1 1
              1 1 1 1 1 1 1 1 1 1
              1 1 1
```

```
********************************
Rank Order of Rsquare of Effects
********************************
```

Rank		Effect	Rsquared	df
1	18	X5_AOV	0.121549834	13
2	19	X6_AOV	0.037462060	7
3	15	X2_AOV	0.030701225	15
4	9	C10	0.030637506	22
5	16	X3_AOV	0.020983319	14
6	20	X7_AOV	0.015931310	12
7	6	X7	0.009410823	1
8	17	X4_AOV	0.004154102	2
9	3	X4	0.004035827	1
10	8	C9	0.004001295	1
11	11	X12	0.003588089	1
12	12	X13	0.003433426	1
13	10	C11	0.002914307	1
14	5	X6	0.002772838	1
15	1	X2	0.001343905	1
16	2	X3	0.000793343	1
17	14	C15	0.000193129	1
18	4	X5	2.8317e-005	1
19	13	X14	1.3073e-005	1

20	7	C8	8.4314e-007	1
21	21	C10_GRP	0.000000000	1

```
************************************
History of Forward Variable Selection
************************************
```

Step	Effect	DF	Rsquared	F_value	pval	TotSSQ	BIC
1	X5_AOV	13	0.1215	7.1951621	0.0000	18.302234	-1102.9076
2	X2_AOV	15	0.0214	1.1024480	0.3501	3.2283708	-1085.5050
3	C10	22	0.0202	0.6998800	0.8421	3.0362688	-1049.3670
4	X3_AOV	14	0.0170	0.9256676	0.5305	2.5596753	-1024.6554
5	X7_AOV	11	0.0131	0.9094186	0.5308	1.9790248	-994.87135
6	X6_AOV	7	0.0044	0.4796478	0.8496	0.6681890	-955.51063
7	X4_AOV	2	0.0030	1.1289125	0.3241	0.4491430	-912.49488
8	X3	1	0.0006	0.4168179	0.5188	0.0829964	-910.03679
9	X7	1	0.0002	0.1484535	0.7002	0.0296017	-907.26223
10	C11	1	0.0002	0.1262981	0.7224	0.0252204	-901.48699
11	X14	1	0.0001	0.0934492	0.7599	0.0186890	-898.61859
12	X2	1	0.0001	0.0550732	0.8145	0.0110315	-895.69571

```
**********************************
Final ANOVA Table for Target: Y[1]
**********************************
```

Effect	DF	Rsquared	Sum-of-Squares
Model	89	0.20183074	30.39044493183
Error	600	.	120.1834681116
Total	689	.	150.5739130435

MSE	0.21309126	SBC	-2029.51691
AIC	-953.893764	BIC	-895.695710
F	1.60243869	Prb	8.292e-004
CP	126.000000	AR2	0.02493153

```
**********************************
Effects not chosen for Target: Y[1]
**********************************
```

Effect	DF	Rsquared	F_value	pval	Sum-of-Squares

```
     X4   0  0.00000000  0.00000000  1.0000  0.000000000000
     X5   0  4.872e-006  0.00000000  1.0000  0.000733556117
     X6   0  2.321e-008  0.00000000  1.0000  3.4951615e-006
     C8   1  8.431e-007  0.00000000  0.0000  0.000126954516
     C9   0  0.00000000  0.00000000  1.0000  0.000000000000
    X12   1  3.446e-005  0.02586165  0.8723  0.005188661445
    X13   1  2.321e-008  1.742e-005  0.9967  3.4951615e-006
    C15   1  4.872e-006  0.00365610  0.9518  0.000733556117
C10_GRP   1  0.00000000  0.00000000  0.0000  0.000000000000


        ******************************************
        History of 2 Parameter Logistic Estimation
        ******************************************

          Iter       Alpha        Beta    Fiterror

             1 -2.01562469  4.02570300  20.2690276
             2 -2.70574444  5.28028831  2.05024956
             3 -2.81577499  5.47507229  0.05004752
             4 -2.81805991  5.47906918  2.120e-005

  Logistic Regression: Intercept=-2.81806 Slope=5.47907


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

 Index                                 Value       StdErr
 Absolute Classification Error           166         .
 Classification Accuracy         75.94202899         .
 Concordant Pairs                39.86582737         .
 Discordant Pairs                4.932817433         .
 Tied Pairs                      55.20135520         .
 Goodman-Kruskal Gamma           0.779778274  0.041591273
 Kendall Tau_b                   0.407968225  0.037799909
 Stuart Tau_c                    0.304927536  0.032450233
 Somers D C|R                    0.349330099  0.035763201


                   Classification Table
                   --------------------

                        |    Predicted
               Observed |        0       1
               ---------|-----------------
```

```
            0 |        97     125
            1 |        41     427
```

Goodness of Fit
****

Dense Row Vector (ncol=25)

```
R |   ModelNpar  SSQ_model  SSQ_error  SSQ_total   DF_model
      139.00000  30.390445  120.18347  150.57391  89.000000

R |    DF_error   DF_total        MSE    Rsquare        SBC
      600.00000  689.00000  0.2130913  0.2018307 -2029.5169

R |         AIC        BIC     Fvalue      Fprob   MallowCp
     -953.89376 -895.69571  1.6024387  8.29e-004  126.00000

R |       AdjR2   LogInter   LogSlope        N00        N01
      0.0249315  0.0000000  0.0000000  97.000000  125.00000

R |         N10        N11   ClassErr   Accuracy     unused
      41.000000  427.00000  166.00000  75.942029  0.0000000
```

Parameter Estimates
****

Dense Row Vector (ncol=139)

```
R |   Intercept         X2         X3         X4         X5
      2.8529141 -0.0037119 -0.0246488  0.0000000  0.0000000

R |          X6         X7        C81        C82        C91
      0.0000000 -0.0136504  0.0000000  0.0000000  0.0000000

R |         C92       C101       C102       C103       C104
      0.0000000 -0.3397474 -0.3838756 -0.3583442 -0.3382123

R |        C105       C106       C107       C108       C109
     -0.6063149 -0.3763497 -0.3528704 -0.3673760 -0.2471146

R |       C1010      C1011      C1012      C1013      C1014
     -0.2740398 -0.4220503 -0.2909063 -0.2687961 -1.2896928

R |       C1015      C1016      C1017      C1018      C1019
     -0.5281719 -0.0607702 -0.5277829 -0.2201710 -0.7266864
```

```
R |       C1020      C1021      C1022      C1023       C111
     -0.1231846 -0.9027262 -0.6585144  4.04e-014  0.0132275

R |        C112        X12        X13        X14       C151
     -4.67e-014  0.0000000  0.0000000  1.44e-006  0.0000000

R |        C152    X2_AOV1    X2_AOV2    X2_AOV3    X2_AOV4
      0.0000000 -0.4704336 -0.4180070 -0.4170613 -0.4390728

R |     X2_AOV5    X2_AOV6    X2_AOV7    X2_AOV8    X2_AOV9
     -0.3546753 -0.3610055 -0.2743976 -0.4106716 -0.4197757

R |    X2_AOV10   X2_AOV11   X2_AOV12   X2_AOV13   X2_AOV14
     -0.3069334 -0.4882410 -0.1739709 -0.3826804 -0.5483881

R |    X2_AOV15   X2_AOV16    X3_AOV1    X3_AOV2    X3_AOV3
      0.0021945  6.82e-014 -1.2196338 -1.1275426 -1.1054884

R |     X3_AOV4    X3_AOV5    X3_AOV6    X3_AOV7    X3_AOV8
     -1.1382499 -1.1999553 -1.0366389 -1.0458114 -0.9595790

R |     X3_AOV9   X3_AOV10   X3_AOV11   X3_AOV12   X3_AOV13
     -0.9300059 -1.0165325 -0.4966911 -0.9544267 -0.5682768

R |    X3_AOV14   X3_AOV15   X3_AOV16    X4_AOV1    X4_AOV2
      0.0000000 -0.1480996  1.87e-013  0.4779966  0.0000000

R |     X4_AOV3    X4_AOV4    X4_AOV5    X4_AOV6    X4_AOV7
      0.0000000  0.0000000  0.0000000  0.0000000  0.0000000


R |     X4_AOV8    X4_AOV9   X4_AOV10   X4_AOV11   X4_AOV12
      0.0000000  0.4288187  0.0000000  0.0000000  0.0000000

R |    X4_AOV13   X4_AOV14   X4_AOV15   X4_AOV16    X5_AOV1
      0.0000000  0.0000000  0.0000000 -1.11e-013 -0.0095549

R |     X5_AOV2    X5_AOV3    X5_AOV4    X5_AOV5    X5_AOV6
      0.0950733 -0.0258700  0.0758613 -0.4088763  0.0000000

R |     X5_AOV7    X5_AOV8    X5_AOV9   X5_AOV10   X5_AOV11
     -0.1381757  0.2253057  0.0321071  0.1104544  0.0000000

R |    X5_AOV12   X5_AOV13   X5_AOV14   X5_AOV15   X5_AOV16
```

```
      0.0451814 -0.2806599  0.0783253  0.0105297  4.35e-014

R |     X6_AOV1    X6_AOV2    X6_AOV3    X6_AOV4    X6_AOV5
     -0.1765489  0.0000000 -0.0128717  0.0000000 -0.1905579

R |     X6_AOV6    X6_AOV7    X6_AOV8    X6_AOV9   X6_AOV10
      0.0000000  0.0591179  0.0000000  0.0917959  0.0000000

R |    X6_AOV11   X6_AOV12   X6_AOV13   X6_AOV14   X6_AOV15
      0.0000000  0.0000000  0.2098289  0.0000000  0.0120097

R |    X6_AOV16    X7_AOV1    X7_AOV2    X7_AOV3    X7_AOV4
      3.36e-014 -0.5552012 -0.5648778 -0.3443019 -0.3384460

R |     X7_AOV5    X7_AOV6    X7_AOV7    X7_AOV8    X7_AOV9
     -0.3459467 -0.1533728 -0.2132563 -0.2792203 -0.1285056

R |    X7_AOV10   X7_AOV11   X7_AOV12   X7_AOV13   X7_AOV14
      0.2856783 -0.1061992  1.33e-013  0.0000000  0.0000000

R |    X7_AOV15   X7_AOV16   C10_GRP1   C10_GRP2
      0.0000000 -7.31e-018  0.0000000  0.0000000


                 *********************************
                 Effect Mapping in Parameter Object
                 *********************************

                     Effect Type InMod Column Ncol
                         X2    V     Y      1    1
                         X3    V     Y      2    1
                         X4    V     N      3    1
                         X5    V     N      4    1
                         X6    V     N      5    1
                         X7    V     Y      6    1
                         C8    C     N      7    2
                         C9    C     N      9    2
                        C10    C     Y     11   23
                        C11    C     Y     34    2
                        X12    V     N     36    1
                        X13    V     N     37    1
                        X14    V     Y     38    1
                        C15    C     N     39    2
                     X2_AOV    C     Y     41   16
                     X3_AOV    C     Y     57   16
                     X4_AOV    C     Y     73   16
```

```
 X5_AOV    C     Y     89   16
 X6_AOV    C     Y    105   16
 X7_AOV    C     Y    121   16
C10_GRP    C     N    137    2

 Total Processing Time: 0
```

2. Boston Housing: nobs=506, nvar=14:

```
fid = fopen("..\\tdata\\housing.dat","r");
form = "%g %g %g %g %g %g %g %g %g %g %g %g %g %g";
hous = fscanf(fid,form,506,14);
vnam = [ "crim" "zn" "indus" "chas" "nox" "rm" "age"
         "dis" "rad" "tax" "ptrat" "b" "lstat" "medv" ];
hous = cname(hous,vnam);


 optn = [ 3 ];
 modl = "14 = 1:13";
 < gof,gof2,parm,yptr > = lrforw(hous,modl,optn);
```

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

           Number Valid Observations  506
           Response Variable         Y[14]
           N Independend Variables     13
           Number Parameters           15
           Use R2 Selection Criterion
           Without AOV16 Variables
           Without Group Variables
           Max Selection Steps          0
           Lower R2 Threshold   1.00e-005
           Lower Threshhold for 5.00e-005
           Max Variables in Model    3000


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

Intercept + X1  + X2  + X3  + X4  + X5  + X6  + X7  + X8  + X9  +
                      X10 + X11 + X12 + X13
```

****
Simple Statistics
****

Column	Nobs	Mean	Std Dev	Skewness	Kurtosis
Y[14]	506	22.532806	9.1971041	1.1080984	1.4951969
X[1]	506	3.6135236	8.6015451	5.2231487	37.130508
X[2]	506	11.363636	23.322453	2.2256663	4.0315101
X[3]	506	11.136779	6.8603530	0.2950215	-1.2335397
X[4]	506	0.0691700	0.2539940	3.4059042	9.6382638
X[5]	506	0.5546951	0.1158777	0.7293079	-0.0646673
X[6]	506	6.2846344	0.7026172	0.4036121	1.8915002
X[7]	506	68.574901	28.148862	-0.5989626	-0.9677156
X[8]	506	3.7950427	2.1057101	1.0117806	0.4879412
X[9]	506	9.5494071	8.7072594	1.0048146	-0.8672320
X[10]	506	408.23715	168.53712	0.6699559	-1.1424080
X[11]	506	18.455534	2.1649458	-0.8023248	-0.2850920
X[12]	506	356.67403	91.294863	-2.8903738	7.2268177
X[13]	506	12.653063	7.1410615	0.9064601	0.4932395

****
Rank Order of Rsquare of Effects
****

Rank	Effect		Rsquared	df
1	13	lstat	0.544146302	1
2	6	rm	0.483525442	1
3	11	ptrat	0.257847337	1
4	3	indus	0.233990041	1
5	10	tax	0.219525923	1
6	5	nox	0.182603046	1
7	1	crim	0.150780471	1
8	9	rad	0.145638581	1
9	7	age	0.142094746	1
10	2	zn	0.129920846	1
11	12	b	0.111196120	1
12	8	dis	0.062464374	1
13	4	chas	0.030716130	1

****
History of Forward Variable Selection
****

Step	Effect	DF	Rsquared	F_value	pval	TotSSQ	BIC

1	lstat	1	0.5441	601.61788	0.0000	23243.914	1853.0250
2	rm	1	0.0944	131.39417	0.0000	4033.0720	1737.6122
3	ptrat	1	0.0401	62.579071	0.0000	1711.3239	1680.1951
4	dis	1	0.0117	18.900871	0.0000	499.07760	1663.4929
5	nox	1	0.0178	30.457240	0.0000	759.56363	1635.6166
6	chas	1	0.0077	13.492008	0.0003	328.27144	1624.1693
7	b	1	0.0064	11.448446	0.0008	272.83706	1614.7291
8	zn	1	0.0044	8.0832207	0.0047	189.93616	1608.6345
9	crim	1	0.0022	4.0554868	0.0446	94.711868	1606.5912
10	rad	1	0.0054	9.9656003	0.0017	228.60449	1598.5908
11	tax	1	0.0064	12.197765	0.0005	273.61918	1588.3424
12	indus	1	0.0001	0.1120291	0.7380	2.5175522	1590.3298

```
**********************************
Final ANOVA Table for Target: medv
**********************************
```

Effect	DF	Rsquared	Sum-of-Squares
Model	12	0.74064122	31637.44918565
Error	493	.	11078.84641990
Total	505	.	42716.29560555

MSE	22.4723051	SBC	1480.70065
AIC	1587.64562	BIC	1590.32983
F	117.320145	Prb	6.079e-136
CP	13.0000000	AR2	0.73432822

```
***********************************
Effects not chosen for Target: medv
***********************************
```

Effect	DF	Rsquared	F_value	pval	Sum-of-Squares
age	1	1.448e-006	0.00274601	0.9582	0.061834213045

```
Goodness of Fit
***************
```

Dense Row Vector (ncol=25)

R |	ModelNpar	SSQ_model	SSQ_error	SSQ_total	DF_model
	14.000000	31637.449	11078.846	42716.296	12.000000

```
R |   DF_error  DF_total       MSE   Rsquare       SBC
     493.00000 505.00000 22.472305 0.7406412 1480.7006

R |        AIC       BIC    Fvalue     Fprob  MallowCp
     1587.6456 1590.3298 117.32015 6.08e-136 13.000000

R |      AdjR2  LogInter  LogSlope       N00       N01
     0.7343282 0.0000000 0.0000000     .         .

R |        N10       N11  ClassErr  Accuracy    unused
         .         .         .         .     0.0000000


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

              Dense Row Vector (ncol=14)

R |  Intercept      crim        zn     indus      chas
     36.436930 -0.1080056 0.0463337 0.0205622 2.6890263

R |        nox        rm       age       dis       rad
    -17.713543 3.8143933 0.0000000 -1.4786115 0.3057860

R |        tax     ptrat         b     lstat
    -0.0123287 -0.9522112 0.0093207 -0.5238519


              *********************************
              Effect Mapping in Parameter Object
              *********************************

                  Effect Type InMod Column Ncol
                    crim    V     Y      1    1
                      zn    V     Y      2    1
                   indus    V     Y      3    1
                    chas    V     Y      4    1
                     nox    V     Y      5    1
                      rm    V     Y      6    1
                     age    V     N      7    1
                     dis    V     Y      8    1
                     rad    V     Y      9    1
                     tax    V     Y     10    1
                   ptrat    V     Y     11    1
                       b    V     Y     12    1
                   lstat    V     Y     13    1
```

```
Total Processing Time: 0
```

3. Boston Housing with AOV16 Variables: nobs=506, nvar=14:

```
fid = fopen("..\\tdata\\housing.dat","r");
form = "%g %g %g %g %g %g %g %g %g %g %g %g %g %g";
hous = fscanf(fid,form,506,14);
vnam = [ "crim" "zn" "indus" "chas" "nox" "rm" "age"
         "dis" "rad" "tax" "ptrat" "b" "lstat" "medv" ];
hous30 = hous[1:30,];
hous30 = cname(hous30,vnam);


aov16 = [ 1:12 ];        /* aov: numbers the effects */
modl = "14 = 1:3 5:13";
optn = cons(10,1,.);   optn[1]= 3; optn[7]= 0; optn[10]= 1;
< gof,parm,yptr > = lrforw(hous30,modl,optn,.,aov16);
```

For AOV16 variables the degrees of freedom can be smaller than 16 due to empty bins:

```
                ********************************
                Rank Order of Rsquare of Effects
                ********************************

           Rank          Effect     Rsquared        df
              1      24 lstat_AOV  0.812198120        11
              2      17    rm_AOV  0.774998024        12
              3      19   dis_AOV  0.756039987        12
              4      13  crim_AOV  0.728005434         9
              5      18   age_AOV  0.701328153        13
              6      20   rad_AOV  0.677356313         4
              7      22 ptrat_AOV  0.665024987         3
              8      21   tax_AOV  0.649057050         3
              9      16   nox_AOV  0.641715445         3
             10       4       nox  0.632830510         1
             11       9       tax  0.617942270         1
             12      15 indus_AOV  0.594456046         2
             13       5        rm  0.591736634         1
             14       3     indus  0.505119633         1
             15      12     lstat  0.501338746         1
             16       1      crim  0.396981746         1
             17       6       age  0.343752149         1
             18       8       rad  0.226321823         1
             19      23     b_AOV  0.217052128         5
             20       7       dis  0.175581418         1
```

```
21     11          b  0.092501601          1
22     10      ptrat  0.084159355          1
23     14     zn_AOV  0.012358912          2
24      2         zn  0.000155577          1
```

In the follwing table are effects with names longer than eight chars refferred to by their number:

```
               *************************************
               History of Forward Variable Selection
               *************************************

Step  Effect  DF Rsquared    F_value    pval     TotSSQ        BIC

   1      24  11   0.8122  7.0768805  0.0002  857.41624  133.69250
   2 dis_AOV  12   0.1405  1.4859702  0.3259  148.34299  15.273440
   3  rm_AOV   6   0.0473          .  0.0000  49.914525          .


                 **********************************
                 Final ANOVA Table for Target: medv
                 **********************************

              Effect    DF    Rsquared  Sum-of-Squares

               Model    29  1.00000000  1055.673758921
               Error     0           .  0.000000000000
               Total    29           .  1055.673758921

               MSE    0.00000000       SBC       .
               AIC         .           BIC       .
               F           .           Prb  0.00000000
               CP          .           AR2  1.00000000


             *****************************************
             Residuals of Linear Regression (Training)
             *****************************************

               Nobs        Yobs        Yprd    Residual
                  1  24.0000000  24.0000000 -1.066e-014
                  2  21.6000004  21.6000004 -1.066e-014
                  3  34.7000008  34.7000008  7.105e-015
                  4  33.4000015  33.4000015 -7.105e-015
                  5  36.2000008  36.2000008 -7.105e-015
                  .....................................
```

```
 25  15.6000004  15.6000004  0.00000000
 26  13.8999996  13.8999996  2.842e-014
 27  16.6000004  16.6000004 -1.421e-014
 28  14.8000002  14.8000002 -2.132e-014
 29  18.3999996  18.3999996 -2.132e-014
 30  21.0000000  21.0000000 -3.553e-015

Sum-of-Squared Residual 7.63933e-027


 **********************************
 Effect Mapping in Parameter Object
 **********************************

     Effect Type InMod Column Ncol
       crim    V     N      1    1
         zn    V     N      2    1
      indus    V     N      3    1
        nox    V     N      4    1
         rm    V     N      5    1
        age    V     N      6    1
        dis    V     N      7    1
        rad    V     N      8    1
        tax    V     N      9    1
      ptrat    V     N     10    1
          b    V     N     11    1
      lstat    V     N     12    1
   crim_AOV    C     N     13   16
     zn_AOV    C     N     29   16
  indus_AOV    C     N     45   16
    nox_AOV    C     N     61   16
     rm_AOV    C     Y     77   16
    age_AOV    C     N     93   16
    dis_AOV    C     Y    109   16
    rad_AOV    C     N    125   16
    tax_AOV    C     N    141   16
  ptrat_AOV    C     N    157   16
      b_AOV    C     N    173   16
  lstat_AOV    C     Y    189   16

     Total Processing Time: 0
```

3.3 Function `pca`

<gof,eval,evec,flod,frot,prot> = pca(data,optn<,targ<,wgt>>)

Purpose: The `pca` function implements a number of algorithms for principal component analysis. Assuming a $N \times n$ data matrix $\mathbf{X}$ with $n \times n$ covariance or correlation matrix $\mathbf{C}$,

$$\mathbf{C} = \mathbf{L}\mathbf{L}^T$$

and $\mathbf{L}$ is the $ntimesm$ component loading matrix, usually in the form $\mathbf{L} = \mathbf{V}\mathbf{D}^{\mathbf{1/2}}$ where $\mathbf{V}$ is the $ntimesm$ eigenvector matrix with m orthogonal columns and $\mathbf{D}$ is the $m \times m$ diagonal matrix containing the m largest eigenvalues of $\mathbf{C}$. Using the $vers$ion option we can select an algorithm among the eight available implementations. The first four compute eigenvalue decompositions of symmetric cross product (covariance or correlation) matrices (either $\mathtt{X}^T\mathbf{X}$ or $\mathtt{XX}^T$ whichever is the smaller one) and the other four methods use the singular value decomposition of the raw matrix $\mathbf{X}$:

"cev1" (11) all values, all vectors dense eigen value decomposition (EVD): the algorithm is based on the eigenvalue decomposition of a dense symmetric X^TX or XX^T matrix and can be applied to raw input data and also to $n \times n$ input covariance or correlation matrices;

"cev2" (12) all values, few vectors dense EVD: the algorithm is based on the eigenvalue decomposition of a dense symmetric X^TX or XX^T matrix and can be applied to raw input data and also to $n \times n$ input covariance or correlation matrices;

"cev3" (13) selected triplets Lapack EVD: the algorithm is based on the eigenvalue decomposition of a dense symmetric X^TX or XX^T matrix and can be applied to raw input data and also to $n \times n$ input covariance or correlation matrices;

"cev4" (14) selected triplets Arpack EVD: the algorithm is appropriate for large and sparse data matrix $\mathbf{X}$ and is based on the eigenvalue decomposition of a dense symmetric X^TX or XX^T matrix and can be applied to raw input data and also to $n \times n$ input covariance or correlation matrices;

"rsv1" (16) standard dense singular value decomposition (SVD): the algorithm is suitable for medium sized dense data matrices $\mathbf{X}$, and cannot be applied to $n \times n$ input covariance or correlation matrices;

"rsv2" (17) selected triplets Arpack SVD: the algorithm is suitable for large sized sparse data matrices $\mathbf{X}$ and a relatively small number of factors, and cannot be applied to $n \times n$ input covariance or correlation matrices;

"rsv3" (18) selected triplets Block Lanczos SVD: the algorithm is suitable for large sized sparse data matrices $\mathbf{X}$, and cannot be applied to $n \times n$ input covariance or correlation matrices;

"rsv4" (19) selected triplets subspace iteration SVD: the algorithm is suitable for large sized sparse data matrices $\mathbf{X}$, and cannot be applied to $n \times n$ input covariance or correlation matrices.

The eigenvalue algorithms need memory in $O(n^2) + O(n * m)$ or $O(N^2) + O(N * m) + O(n * m)$ whereas the singular value algorithms need memory $O(N * m) + O(n * m)$ for dense $\mathbf{X}$ and $O(nzer) + O(n * m)$ for sparse $\mathbf{X}$. There are a number of criteria for defining the number of components (factors) which can be selected by the following options:

"nfac" This option requires an integer specifying the number components (factors).

"mineig" This option requires a real value for a lower bound of the smallest eigenvalue for the component

"perc" This option requires an int or real value for the proportion of common variance to be accounted for by the selected components. Values in $[0, 1]$ specify a relative proportion, values in $[1, 100]$ a percentage. Default is 1 or 100 %.

"stud" The scree test with the studendized residual is used for specifying the number of factors and requires a real input value in $(0, 1)$ for α (default: $\alpha = .1$).

"cook" The scree test with Cooks distance is used for specifying the number of factors and requires a real input value in $(0, 1)$ for the threshold (default threshold is .25).

For each of these except the `"nfac"` option, the prior computation of a set of eigenvalues is necessary. Note, that the `"perc"`, `"stud"`, and `"cook"` options require the computation of all eigenvalues. This is no problem for dense and medium sized data sets, but maybe a computational problem for very large data sets. When only the `"mineig"` option is specified only the required subset of eigenvalues is computed and sparsity of the matrix can be exploited.

The asymptotic standard errors and Wald confidence intervals of rotated factor loadings depend on:

1. the kind of rotation method appplied and especially the fact whether the rotation is orthogonal or oblique;
2. if the (unstandardized) covariance or the (standardized) correlation matrix is analysed. (Note, you may read in a correlation matrix, but if you specify the analysis option `"anal"` as `"cov"`, the results correspond to those of a covariance analysis.)

3. if Kaiser normalization of the loadings is applied for the specific rotation method.

Note, the asymptotic standard errors and confidence intervals are available only with algorithms `cev1,...,cev4` for $N > n$, i.e. when a $n \times n$ correlation or covariance matrix is computed and stored. They are also currently not available for PROMAX rotated components.

Input: data The first argument `data` must be the name of a data object specifying a

1. `nobs` by `nvar` matrix of raw data: all algorithms can be applied
2. symmetric `nvar` by `nvar` matrix of covariances or correlations.
3. `nvar+1` by `nvar` matrix that contains a symmetric covariance or correlation matrix in its first `nvar` rows and a vector of mean values in its last row.

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.

targ This argument specifies the weight matrix

wgt This argument specifies the weight matrix for target rotation.

Options Matrix Argument: The option argument is specified in form of a two column matrix:

Option Name	Second Column	Meaning
"alpha"	real	the significance level for confidence intervals; default is $\alpha = 0.05$
"ana"	string	data type for analysis
	"cor"	correlations are analyzed
	"cov"	covariances are analyzed
	"ucor"	uncorrected correlations are analyzed
	"ucov"	uncorrected covariances are analyzed
"cl"	string	type of confidence intervals
	"none"	do not compute
	"wald"	Wald linkelihood intervals (fast)
	"boot"	Bootstrap confidence intervals (slow)
"cook"	real	use the scree test with Cooks distance
		for specifying the number of factors
		default threshold is .25
"data"	string	type of input data
	"raw"	$N \times n$ raw data
	"cor"	symmetric correlation matrix
	"cov"	symmetric covariance matrix
"freq"	int	column number of FREQ variable when raw data input
"frot"	string	rotation method for exploratory FACTOR model
	"none"	no rotation
	"crafer"	orthogonal Crawford-Ferguson family (specify γ)
	"varmax"	orthogonal Varimax rotation (crafer with $\gamma = \frac{1}{n}$)
	"quamax"	orthogonal Quartimax rotation (crafer with $\gamma = 0$)
	"equmax"	orthogonal Equamax rotation (crafer with $\gamma = \frac{q}{2n}$
	"parmax"	orthogonal Parsimax rotation (crafer with $\gamma = \frac{q-1}{n+q-2}$)
	"parmax"	orthogonal Factor Parsimony rotation (crafer with $\gamma = 1$)
	"bentlr"	orthogonal Bentler rotation criterion
	"minent"	orthogonal Minimum Entropy rotation
	"oblmin"	orthogonal Direct Oblimin family (specify γ)
	"simmax"	orthogonal Simplimax (specify number nonzero loadings with γ
	"tandm1"	orthogonal Tandem 1 rotation
	"tandm2"	orthogonal Tandem 2 rotation
	"infmax"	orthogonal Infomax rotation
	"mccamm"	orthogonal McCammon rotation
	"crafer"	oblique Crawford-Ferguson family (specify γ)
	"oblmin"	oblique Direct Oblimin family (specify γ)
	"quamin"	oblique Direct Quartimin (oblmin with $\gamma = 0$)
	"biqmin"	oblique Bi-Quartimin (oblmin with $\gamma = .5$)
	"covmin"	oblique Covarimin (oblmin with $\gamma = 1$)
	"bentlr"	oblique Bentler rotation criterion
	"simmax"	oblique Simplimax (specify number nonzero loadings with γ)
	"oblmax"	oblique Oblimax
	"geomin"	oblique Geomin
	"infmax"	oblique Infomax rotation
	"promax"	oblique Promax
	"pri"	principal component rotation
"fnorm"		Kaiser's factor pattern normalization
		use only for exploratory FACTOR model

Option Name	Second Column	Meaning
"mineig"	real	smallest eigenvalue for which factor is retained
		used for specifying the number of factors
"nfac"	int	number of components (factors)
"nobs"	real	number observations when COV or CORR data input
		(is mostly an integer)
"nopr"		perform no printed output
"prin"	int	amount of printed output (=0 is noprint)
"pall"		large amount of printed output
"phis"		some technical output on the performance of algorithms
"perc"	real	proportion of common variance to be accounted for
		used for specifying the number of factors
		default is 1 or 100 %
"psho"		short amount of printed output
"psum"		summary amount of printed output
"ridge"	real	nonnegative ridge value
"rinit"	int	initialization for rotation matrix **T**
"rotp"	real	rotation parameter (γ for Crawford-Ferguson)
"rvers"	int	version of algorithm for rotation
"seed"	real	seed value for random generator
"sing"	real	singularity threshold (default: 1.e-8)
"stud"	real	use the scree test with studendized residual
		for specifying the number of factors
		default: $\alpha = .1$
"vers"		version of algorithm
	"cev1" (11)	all values, all vectors dense EVD: column oriented
	"cev2" (12)	all values, few vectors dense EVD: column oriented
	"cev3" (13)	selected triplets Lapack EVD: column oriented
	"cev4" (14)	selected triplets Arpack EVD: column oriented
	"rsv1" (16)	standard dense SVD: row oriented
	"rsv2" (17)	selected triplets Arpack SVD: row oriented
	"rsv3" (18)	selected triplets Block Lanczos SVD: row oriented
	"rsv4" (19)	selected triplets Rutishauser-Ritz SVD: row oriented
"wgt"	int	column number of WEIGHT variable when raw data input

The number of components (factors) is specified either using the `nfac`, the `mineig` or the `perc` option.

The following table shows whether orthogonal or oblique rotation is the default and if the default can be changed by specifying `par[6]`:

spec	Rotation Method	orthogonal	oblique
"quamax"	Quartimax (Carroll, 1953)	default	yes
"varmax"	Varimax (Kaiser, 1958)	default	yes
"equmax"	Equamax	default	yes
"parmax"	Parsimax	default	yes
"facpar"	Factor Parsimony	default	yes
"crafer"	Crawford-Ferguson (1970)	default	yes
"minent"	Minimum Entropy (Jennrich)	default	no
"tandm1"	Tandem 1 (Comrey,1967)	default	no
"tandm2"	Tandem 2 (Comrey, 1967)	default	no
"mccamm"	McCammon (1966)	default	no
"quamin"	Quartimin (Carroll, 1953)	yes	default
"simmax"	Simplimax (Kiers, 1994)	yes	default
"oblmin"	Oblimin (Carroll, 1960)	yes	default
"biqmin"	Bi-Quartimin	yes	default
"covmin"	Covarimin	yes	default
"bentlr"	Bentler Criterion (1977)	yes	default
"infmax"	Infomax (McKeon,1968)	yes	default
"oblmax"	Oblimax (Saunders,1961)	no	default
"geomin"	Geomin (Yates, 1984)	no	default
"promax"	Promax (Hendrickson,White)	no	default
"targt1"	Target Rotation (Harman,1976)	yes	default
"targt2"	Weighted Target (Browne,2001)	yes	default

Note, currently no asymptotic standard errors of PROMAX rotated components can be computed. However, they are available with the `factor` function.

Output: gof column vector with goodness-of-fit measures.

eval vector of m largest eigenvalues

evec the $n \times m$ matrix $\mathbf{V}$ of eigenvectors containing orthogonal eigenvectors in m columns

flod the $n \times mk$ matrix of unrotated components where $mk = m$ if no Wald ASEs are computed and $mk = 4{*}m$ if Wald ASEs are computed

frot the $n \times mk$ matrix of rotated components where $mk = m$ if no Wald ASEs are computed and $mk = 4 * m$ if Wald ASEs are computed

prot the $m \times m$ matrix of component correlations

Restrictions: 1. The input data cannot contain any missing or string data.

2.

Relationships: noharm(), sem(), frotate(), factor()

Examples: 1. Compare Different Methods :

(a) Australian data set: N=690, n=15:
This data set is tall and skinny:

```
form = fo = " %g";
for (j = 2; j <= 15; j++) form = strcat(form,fo);
fid = fopen("..\\tdata\\australian.dat","r");
data = fscanf(fid,form,.,15);
nr = nrow(data);
print "Observations of Australian.dat:",nr;
x = data[,2:15]; y = data[,1];

optn = [ "data"      "raw" ,
         "anal"      "cov" ,
         "vers"     "cev1" ,
         "nfac"         2 ,
         "frot"   "varmax" ,
         "prin"         3 ];
< gof,eval > = pca(data,optn);
```

```
                  ****************************
                  Principal Component Analysis
                  ****************************

  Input Data. . . . . . . . . . . . . . . . . . . . .  Raw Data
  Analysis of . . . . . . . . . . . . . . . . Covariance Matrix
  Number of Items . . . . . . . . . . . . . . . . . .        15
  Number of Factors (NFactor Criterion) . . . . . .           2
  Number of Subjects. . . . . . . . . . . . . . . . .       690
  C-Version . . . . . . . . . . . . . . .Standard Full EVD Code
  Orthogonal Rotation Method. . . . . .Varimax (Kaiser, 1958)
  Unnormed Rotation: Parameter. . . . . . . . . . . 1.0000000

   Name             Mean     Std Dev    Skewness    Kurtosis
   ---------------------------------------------------------
   V_1       0.678260870  0.46748239 -0.76485905 -1.41911243
   V_2       31.56820290  11.8532728  1.15593502  1.19205859
   V_3       4.758724635  4.97816324  1.48881311  2.27402175
   V_4       1.766666667  0.43006283 -1.15345501 -0.33826957
   V_5       7.372463768  3.68326479 -0.06919047 -0.84904258
   V_6       4.692753623  1.99231607  0.46841183 -0.17813233
   V_7       2.223405797  3.34651336  2.89133042  11.2001917
   V_8       0.523188406  0.49982433 -0.09305595 -1.99713783
   V_9       0.427536232  0.49508002  0.29358655 -1.91937879
   V_10      2.400000000  4.86294003  5.15251986  50.8294313
   V_11      0.457971014  0.49859186  0.16908083 -1.97715098
```

```
V_12      1.928985507  0.29881306 -1.94472547  6.71888763
V_13      184.0144928  172.159274  2.74991175  19.9266976
V_14      1018.385507  5210.10260  13.1406550  214.669972
V_15      0.444927536  0.49731827  0.22212157 -1.95634100

Mardia's Multivariate Kurtosis . . . . . . . .    382.6470
Relative Multivariate Kurtosis . . . . . . . .      2.5006
Normalized Multivariate Kurtosis . . . . . . .    222.5398
Mardia Based Kappa (Browne, 1982). . . . . . .      1.5006
Mean Scaled Univariate Kurtosis  . . . . . . .      6.5817
Adjusted Mean Scaled Univariate Kurtosis . . .      6.7595

Observation numbers with largest contribution to kurtosis


           501       183       364       234       150
      85970.27  42631.19  10507.05  10466.83  6095.615

          Eigenvalues of Model Covariance Matrix
      Total = 27175027.0492  Average = 1811668.4699

                       1           2           3           4
 Eigenvalue  27145297.4  29513.8586  143.549416  29.2552327
 Difference  27115783.5  29370.3091  114.294183  11.3534716
 Proportion  0.99890599  0.00108607  5.282e-006  1.077e-006
 Cumulative  0.99890599  0.99999206  0.99999734  0.99999842

                       5           6           7           8
 Eigenvalue  17.9017611  13.2933763  7.61042879  2.96541195
 Difference  4.60838476  5.68294754  4.64501685  2.65464859
 Proportion  6.588e-007  4.892e-007  2.801e-007  1.091e-007
 Cumulative  0.99999908  0.99999957  0.99999985  0.99999996

                       9          10          11          12
 Eigenvalue  0.31076336  0.23923626  0.21373686  0.17137565
 Difference  0.07152710  0.02549940  0.04236121  0.03691017
 Proportion  1.144e-008  8.804e-009  7.865e-009  6.306e-009
 Cumulative  0.99999997  0.99999998  0.99999998  0.99999999

                            13          14          15
       Eigenvalue  0.13446548  0.08216729  0.06420234
       Difference  0.05229819  0.01796495
       Proportion  4.948e-009  3.024e-009  2.363e-009
       Cumulative  0.99999999  1.00000000  1.00000000
```

Eigenvectors

	V_1	V_2	V_3	V_4	V_5
FAC_1	-3.25e-007	-4.22e-005	-1.18e-004	-9.91e-006	-2.17e-005
FAC_2	-1.59e-004	0.0054498	0.0066998	2.43e-005	-0.0018514

Eigenvectors

	V_6	V_7	V_8	V_9	V_10
FAC_1	-2.48e-005	-3.30e-005	-8.63e-006	-7.38e-006	-5.94e-005
FAC_2	-7.71e-004	0.0015620	2.14e-004	1.69e-004	0.0035233

Eigenvectors

	V_11	V_12	V_13	V_14	V_15
FAC_1	-1.84e-006	-8.01e-006	-0.0021703	-0.9999976	-1.68e-005
FAC_2	-4.16e-004	1.54e-004	-0.9999507	0.0021690	3.24e-004

Unrotated Factor Loadings

	FAC_1	FAC_2
V_1	0.0016922	0.0273332
V_2	0.2196791	-0.9362548
V_3	0.6128357	-1.1509950
V_4	0.0516352	-0.0041802
V_5	0.1132290	0.3180616
V_6	0.1291930	0.1325307
V_7	0.1718077	-0.2683385
V_8	0.0449876	-0.0367902
V_9	0.0384418	-0.0290412
V_10	0.3096898	-0.6052958
V_11	0.0095788	0.0714325
V_12	0.0417346	-0.0264962
V_13	11.307462	171.78751
V_14	5210.1026	-0.3726186
V_15	0.0873536	-0.0556346

Rotated Factor Loadings: Varimax (Kaiser, 1958)

	FAC_1	FAC_2
V_1	0.0016901	0.0273334
V_2	0.2197509	-0.9362379
V_3	0.6129240	-1.1509480
V_4	0.0516355	-0.0041763
V_5	0.1132046	0.3180703

```
V_6    0.1291829  0.1325406
V_7    0.1718283 -0.2683253
V_8    0.0449904 -0.0367868
V_9    0.0384441 -0.0290382
V_10   0.3097363 -0.6052720
V_11   0.0095733  0.0714332
V_12   0.0417366 -0.0264930
V_13   11.294286  171.78838
V_14   5210.1026  0.0269857
V_15   0.0873579 -0.0556279

Orthogonal Transformation Matrix T

              FAC_1      FAC_2
FAC_1    1.0000000  7.67e-005
FAC_2   -7.67e-005  1.0000000
```

The other methods cev2, cev3, and cev4 differ only by the number of available eigenvalues. Also, not much different is the output of the R-SVD methods.

(b) NIR Spectra data set: N=21, n=268
This data set is small and fat:

```
  options NOECHO;
#include "..\\tdata\\nir.dat"
  options ECHO;
  nrtrn = nrow(xtrn); nc = ncol(xtrn);
  nrtst = nrow(xtst);
  print "nrtrn,nctrn=",nrtrn,nc;

  optn = [ "data"      "raw" ,
           "anal"      "cov" ,
           "vers"     "cev1" ,
           "nfac"          2 ,
           "frot"   "varmax" ,
           "prin"          3 ];
  < gof,eval > = pca(xtrn,optn);

                  ****************************
                  Principal Component Analysis
                  ****************************

    Input Data. . . . . . . . . . . . . . . . . . . . . . Raw Data
    Analysis of . . . . . . . . . . . . . . . . Covariance Matrix
    Number of Items . . . . . . . . . . . . . . . . . . .      268
    Number of Factors (NFactor Criterion) . . . . . .            2
```

```
Number of Subjects. . . . . . . . . . . . . . . . .        21
C-Version . . . . . . . . . . . . . . .Standard Full EVD Code
Orthogonal Rotation Method. . . . . .Varimax (Kaiser, 1958)
Unnormed Rotation: Parameter. . . . . . . . . . . 1.0000000
```

Name	Mean	Std Dev	Skewness	Kurtosis
V_1	3.079785714	0.06108026	-1.81706398	6.45840428
V_2	3.051352381	0.10994384	-2.31330971	6.11300328
V_3	2.987742857	0.17333722	-1.93558666	3.68046775
V_4	2.884128571	0.23204478	-1.49044496	1.77660072
V_5	2.742685714	0.27304037	-1.18269186	0.74915327
V_6	2.588371429	0.29988459	-0.96103697	0.20859643
V_7	2.446547619	0.32125377	-0.73261085	-0.12784500
V_8	2.327895238	0.34133601	-0.44372802	-0.38651980
V_9	2.239533333	0.36645360	-0.13895768	-0.59896976
V_10	2.181271429	0.40388758	0.08763269	-0.77947816
...	...	...	...	...
V_265	0.325914762	0.01662340	0.34193210	0.19795115
V_266	0.325785714	0.01669600	0.31636266	0.18260176
V_267	0.325888571	0.01673901	0.30450867	0.16884682
V_268	0.326099048	0.01677756	0.30085481	0.16075352

This is only the first part of the 21×21 cross product matrix:

Cross Products of the Transposed Data

	1	2	3	4	5
1	1.1097232				
2	0.7505248	0.5809512			
3	0.9091725	0.5863159	0.7762054		
4	0.3701252	0.3828433	0.2458641	0.3658969	
5	0.5046231	0.3720574	0.4122105	0.2202073	0.2496557

Eigenvalues of Model Covariance Matrix

Total = 12.6025   Average = 0.6001

	1	2	3	4
Eigenvalue	6.56002901	5.88880063	0.09221895	0.02818838
Difference	0.67122838	5.79658168	0.06403057	0.00848365
Proportion	0.52053381	0.46727229	0.00731751	0.00223673
Cumulative	0.52053381	0.98780610	0.99512361	0.99736034

	5	6	7	8
Eigenvalue	0.01970473	0.01078044	0.00109392	6.298e-004
Difference	0.00892429	0.00968652	4.641e-004	3.571e-004

Proportion	0.00156356	8.554e-004	8.680e-005	4.998e-005
Cumulative	0.99892390	0.99977932	0.99986612	0.99991609

	9	10	11	12
Eigenvalue	2.727e-004	2.412e-004	1.351e-004	1.149e-004
Difference	3.148e-005	1.061e-004	2.018e-005	3.837e-005
Proportion	2.164e-005	1.914e-005	1.072e-005	9.116e-006
Cumulative	0.99993773	0.99995687	0.99996759	0.99997671

	13	14	15	16
Eigenvalue	7.652e-005	6.770e-005	5.081e-005	3.921e-005
Difference	8.823e-006	1.689e-005	1.160e-005	1.451e-005
Proportion	6.072e-006	5.372e-006	4.031e-006	3.111e-006
Cumulative	0.99998278	0.99998815	0.99999218	0.99999529

	17	18	19	20
Eigenvalue	2.470e-005	1.674e-005	1.041e-005	7.467e-006
Difference	7.955e-006	6.329e-006	2.946e-006	7.467e-006
Proportion	1.960e-006	1.328e-006	8.262e-007	5.925e-007
Cumulative	0.99999725	0.99999858	0.99999941	1.00000000

	21
Eigenvalue	-5.690e-017
Difference	
Proportion	-4.515e-018
Cumulative	1.00000000

Eigenvectors

	1	2	3	4	5
FAC_1	-0.1450285	-0.0029272	-0.1659867	0.1284992	-0.0327733
FAC_2	0.4004388	0.3123114	0.3172391	0.2075480	0.2023632

Eigenvectors

	6	7	8	9	10
FAC_1	-0.2207129	0.2737133	0.0974562	-0.0589515	-0.2630688
FAC_2	0.1826295	0.1223784	0.0827045	0.0979092	0.0437228

Eigenvectors

	11	12	13	14	15
FAC_1	0.3296719	0.2028505	0.0549809	-0.1309457	-0.3169474
FAC_2	-0.0137984	-0.0550292	-0.0811586	-0.1116958	-0.1446445

Eigenvectors

	16	17	18	19	20
FAC_1	0.3946226	0.2940454	0.1607963	-0.0150352	-0.1956892
FAC_2	-0.1351410	-0.1878072	-0.2402878	-0.2918104	-0.3365611

Eigenvectors

	21
FAC_1	-0.3885698
FAC_2	-0.3713109

Unrotated Factor Loadings

	FAC_1	FAC_2
V_1	-0.0543280	0.0480745
V_2	-0.1064314	0.1305408
V_3	-0.1797051	0.2332687
V_4	-0.2606939	0.3162974
V_5	-0.3395128	0.3479646
V_6	-0.4142059	0.3337099
V_7	-0.4877752	0.2871232
V_8	-0.5582785	0.2136800
V_9	-0.6270921	0.1253097
V_10	-0.7033861	0.0367309
....	....	....
V_265	0.0024751	-0.0150408
V_266	0.0024407	-0.0152983
V_267	0.0025101	-0.0154560
V_268	0.0026621	-0.0155474

Rotated Factor Loadings: Varimax (Kaiser, 1958)

	FAC_1	FAC_2
V_1	-0.0543280	0.0480745
V_2	-0.1064314	0.1305408
V_3	-0.1797051	0.2332687
V_4	-0.2606939	0.3162974
V_5	-0.3395128	0.3479646
V_6	-0.4142059	0.3337099
V_7	-0.4877752	0.2871232
V_8	-0.5582785	0.2136800
V_9	-0.6270921	0.1253097
V_10	-0.7033861	0.0367309
....	....	....

```
V_265   0.0024751 -0.0150408
V_266   0.0024407 -0.0152983
V_267   0.0025101 -0.0154560
V_268   0.0026621 -0.0155474

Orthogonal Transformation Matrix T

            FAC_1      FAC_2
FAC_1   1.0000000  0.0000000
FAC_2   0.0000000  1.0000000
```

2. Standard Errors for Unrotated and Rotated PCA:
   Only the correlation matrix is given. Therefore means are treated as zeros and standard deviations as ones.

```
print "Six School Subjects (Lawley and Maxwell, 1971, p.66)";
corr = [   1. ,
         .439   1. ,
         .410 .351   1. ,
         .288 .354 .164   1. ,
         .329 .320 .190 .595   1. ,
         .248 .329 .181 .470 .464  1. ];
corr = (tri2sym)corr;
nobs = 220;
```

The following output is similar for all following examples:

```
              ****************************
              Principal Component Analysis
              ****************************

Input Data. . . . . . . . . . . . . . . . .Correlation Matrix
Analysis of . . . . . . . . . . . . . . . . Covariance Matrix
Number of Items . . . . . . . . . . . . . . . . .           6
Specified Number of Factors . . . . . . . . . . .           2
Number of Subjects. . . . . . . . . . . . . . . .         220
Version . . . . . . . . . . . . . . . .Standard Full EVD Code
Rotation Method . . . . . . . . . . . . . . . . . No Rotation


             Name             Mean     Std Dev
             ---------------------------------
             V_1       0.000000000  1.00000000
             V_2       0.000000000  1.00000000
```

V_3	0.000000000	1.00000000
V_4	0.000000000	1.00000000
V_5	0.000000000	1.00000000
V_6	0.000000000	1.00000000

Covariance Matrix

	V_1	V_2	V_3	V_4	V_5
V_1	1.0000000				
V_2	0.4390000	1.0000000			
V_3	0.4100000	0.3510000	1.0000000		
V_4	0.2880000	0.3540000	0.1640000	1.0000000	
V_5	0.3290000	0.3200000	0.1900000	0.5950000	1.0000000
V_6	0.2480000	0.3290000	0.1810000	0.4700000	0.4640000

Covariance Matrix

	V_6
V_6	1.0000000

Determinant = 2.37409534e-001 (Ln = -1.4380e+000)

Eigenvalues of Model Covariance Matrix:
Total = 6.0000 Average = 1.0000

	1	2	3	4
Eigenvalue	2.73288407	1.12977037	0.61517387	0.60122188
Difference	1.60311371	0.51459650	0.01395199	0.07642497
Proportion	0.45548068	0.18829506	0.10252898	0.10020365
Cumulative	0.45548068	0.64377574	0.74630472	0.84650836

	5	6
Eigenvalue	0.52479691	0.39615290
Difference	0.12864401	
Proportion	0.08746615	0.06602548
Cumulative	0.93397452	1.00000000

Eigenvectors

	V_1	V_2	V_3	V_4	V_5
FAC_1	-0.3979211	-0.4164293	-0.3129591	-0.4466126	-0.4499766
FAC_2	0.4224739	0.2732052	0.5996225	-0.3885864	-0.3532281

```
                 Eigenvectors

                          V_6
              FAC_1  -0.4103173
              FAC_2  -0.3340032
```

(a) Unstandardized Unrotated PCA:

```
optn = [ "data"      "cor" ,
         "anal"      "cov" ,
         "nobs"        220 ,
         "vers"     "cev1" ,
         "nfac"          2 ,
         "cl"       "wald" ,
         "prin"          3 ];
< gof,eval,flod > = pca(corr,optn);
```

```
                 ****************************
                 Principal Component Analysis
                 ****************************

  Input Data. . . . . . . . . . . . . . . . .Correlation Matrix
  Analysis of . . . . . . . . . . . . . . . . Covariance Matrix
  Number of Items . . . . . . . . . . . . . . . . .          6
  Specified Number of Factors . . . . . . . . . . .          2
  Number of Subjects. . . . . . . . . . . . . . . .        220
  Version . . . . . . . . . . . . . . . .Standard Full EVD Code
  Rotation Method . . . . . . . . . . . . . . . . . No Rotation

         Unrotated Factor Loadings with Standard Errors
         ----------------------------------------------

                           FAC_1                   FAC_2
       V_1   0.6578207  0.0793558  0.4490503  0.0895841
            [ 0.502286, 0.813355] [ 0.273469, 0.624632]

       V_2   0.6884175  0.0741050  0.2903916  0.1015619
            [ 0.543174, 0.833661] [ 0.091334, 0.489449]

       V_3   0.5173663  0.0919484  0.6373427  0.0918856
            [ 0.337151, 0.697582] [ 0.457250, 0.817435]

       V_4   0.7383147  0.0733037 -0.4130310  0.0732165
            [ 0.594642, 0.881987] [-0.556533,-0.269529]
```

```
V_5   0.7438760  0.0720214 -0.3754484  0.0776596
     [ 0.602717, 0.885035] [-0.527658,-0.223238]

V_6   0.6783135  0.0759136 -0.3550142  0.0966014
     [ 0.529525, 0.827101] [-0.544350,-0.165679]
```

(b) Standardized Unrotated PCA:

```
optn = [ "data"      "cor" ,
         "anal"      "cor" ,
         "nobs"        220 ,
         "vers"     "cev1" ,
         "nfac"          2 ,
         "cl"       "wald" ,
         "frot"     "none" ,
         "prin"          3 ];
< gof, est > = pcа(corr,optn);
```

```
                  ****************************
                  Principal Component Analysis
                  ****************************

  Input Data. . . . . . . . . . . . . . . . .Correlation Matrix
  Analysis of . . . . . . . . . . . . . . . .Correlation Matrix
  Number of Items . . . . . . . . . . . . . . . . .           6
  Specified Number of Factors . . . . . . . . . . .           2
  Number of Subjects. . . . . . . . . . . . . . . .         220
  Version . . . . . . . . . . . . . . . . .Standard Full EVD Code
  Rotation Method . . . . . . . . . . . . . . . . . No Rotation

         Unrotated Factor Loadings with Standard Errors
         ----------------------------------------------

                       FAC_1                  FAC_2
       V_1   0.6578207  0.0463014  0.4490503  0.0735708
            [ 0.567072, 0.748570] [ 0.304854, 0.593246]

       V_2   0.6884175  0.0411787  0.2903916  0.0860066
            [ 0.607709, 0.769126] [ 0.121822, 0.458961]

       V_3   0.5173663  0.0664515  0.6373427  0.0713588
            [ 0.387124, 0.647609] [ 0.497482, 0.777203]

       V_4   0.7383147  0.0341763 -0.4130310  0.0613606
            [ 0.671330, 0.805299] [-0.533296,-0.292766]
```

```
V_5   0.7438760  0.0332094 -0.3754484  0.0659694
     [ 0.678787, 0.808965] [-0.504746,-0.246151]

V_6   0.6783135  0.0427755 -0.3550142  0.0830693
     [ 0.594475, 0.762152] [-0.517827,-0.192201]
```

(c) Unstandardized Normed Orthogonally Rotated PCA:

```
optn = [ "data"      "cor" ,
         "anal"      "cov" ,
         "nobs"        220 ,
         "vers"     "cev1" ,
         "nfac"          2 ,
         "cl"       "wald" ,
         "rvers"    "comm" ,
         "frot"   "varmax" ,
         "fnorm"           ,
         "prin"          3 ];
< gof, est > = pca(corr,optn);
```

```
                  ****************************
                  Principal Component Analysis
                  ****************************

  Input Data. . . . . . . . . . . . . . . . .Correlation Matrix
  Analysis of . . . . . . . . . . . . . . . . Covariance Matrix
  Number of Items . . . . . . . . . . . . . . . . .           6
  Specified Number of Factors . . . . . . . . . . .           2
  Number of Subjects. . . . . . . . . . . . . . . .         220
  Version . . . . . . . . . . . . . . . . .Standard Full EVD Code
  Rotation Method . . . . . . . . . . .Varimax (Kaiser, 1958)
  Normed Rotation Method. . . . . . . . . . . . . . 1.0000000

                   Unrotated Factor Loadings

                             FAC_1      FAC_2
                   V_1   0.6578207  0.4490503
                   V_2   0.6884175  0.2903916
                   V_3   0.5173663  0.6373427
                   V_4   0.7383147 -0.4130310
                   V_5   0.7438760 -0.3754484
                   V_6   0.6783135 -0.3550142

          Rotated Factor Loadings with Standard Errors
          --------------------------------------------
```

```
                        FAC_1                   FAC_2
V_1   0.2253107  0.0734485  0.7639433  0.0790763
     [ 0.081354, 0.369267] [ 0.608957, 0.918930]

V_2   0.3493867  0.0923938  0.6604354  0.0928254
     [ 0.168298, 0.530475] [ 0.478501, 0.842370]

V_3  -0.0025887  0.0627659  0.8208940  0.0816661
     [-0.125608, 0.120430] [ 0.660831, 0.980957]

V_4   0.8330754  0.0642808  0.1472704  0.0625851
     [ 0.707087, 0.959063] [ 0.024606, 0.269935]

V_5   0.8136040  0.0662075  0.1798932  0.0641233
     [ 0.683840, 0.943368] [ 0.054214, 0.305573]

V_6   0.7499035  0.0786806  0.1542368  0.0791837
     [ 0.595692, 0.904115] [-9.6e-004, 0.309434]
```

(d) Standardized Unnormed Orthogonally Rotated PCA:

```
optn = [ "data"      "cor" ,
         "anal"      "cor" ,
         "nobs"        220 ,
         "vers"     "cev1" ,
         "nfac"          2 ,
         "cl"       "wald" ,
         "rvers"    "comm" ,
         "frot"   "varmax" ,
         "prin"          3 ];
< gof, est > = pca(corr,optn);
```

```
                  ****************************
                  Principal Component Analysis
                  ****************************

  Input Data. . . . . . . . . . . . . . . .Correlation Matrix
  Analysis of . . . . . . . . . . . . . . .Correlation Matrix
  Number of Items . . . . . . . . . . . . . . . .          6
  Specified Number of Factors . . . . . . . . . .          2
  Number of Subjects. . . . . . . . . . . . . . .        220
  Version . . . . . . . . . . . . . . . .Standard Full EVD Code
  Rotation Method . . . . . . . . . . .Varimax (Kaiser, 1958)
  Unnormed Rotation Method. . . . . . . . . . . . . 1.0000000
```

```
                Unrotated Factor Loadings

                          FAC_1      FAC_2
                V_1   0.6578207  0.4490503
                V_2   0.6884175  0.2903916
                V_3   0.5173663  0.6373427
                V_4   0.7383147 -0.4130310
                V_5   0.7438760 -0.3754484
                V_6   0.6783135 -0.3550142

         Rotated Factor Loadings with Standard Errors
         --------------------------------------------

                        FAC_1                  FAC_2
   V_1   0.2386811  0.0759140  0.7598721  0.0451430
        [ 0.089892, 0.387470] [ 0.671393, 0.848351]

   V_2   0.3609217  0.0907056  0.6542029  0.0718067
        [ 0.183142, 0.538701] [ 0.513464, 0.794941]

   V_3   0.0118162  0.0585867  0.8208130  0.0340498
        [-0.103012, 0.126644] [ 0.754077, 0.887549]

   V_4   0.8355314  0.0251807  0.1326295  0.0597901
        [ 0.786178, 0.884885] [ 0.015443, 0.249816]

   V_5   0.8166354  0.0292857  0.1655889  0.0616674
        [ 0.759237, 0.874034] [ 0.044723, 0.286455]

   V_6   0.7524945  0.0435023  0.1410543  0.0827566
        [ 0.667232, 0.837757] [-0.021146, 0.303254]
```

(e) Unstandardized Normed Oblique Rotated PCA:

```
optn = [ "data"      "cor" ,
         "anal"      "cov" ,
         "nobs"        220 ,
         "vers"     "ogas" ,
         "nfac"          2 ,
         "cl"       "wald" ,
         "rvers"    "comm" ,
         "frot"   "quamin" ,
         "fnorm"           ,
         "prin"          3 ];
< gof, est > = pca(corr,optn);
```

```
                ***************************
                Principal Component Analysis
                ***************************

Input Data. . . . . . . . . . . . . . . . .Correlation Matrix
Analysis of . . . . . . . . . . . . . . . . Covariance Matrix
Number of Items . . . . . . . . . . . . . . . . . .         6
Specified Number of Factors . . . . . . . . . . . .         2
Number of Subjects. . . . . . . . . . . . . . . . .       220
Version . . . . . . . . . . . . . . . . .Standard Full EVD Code
Rotation Method . . . . . . . . . Quartimin (Carroll, 1953)
Normed Rotation Method. . . . . . . . . . . . . . . 0.0000000

                 Unrotated Factor Loadings

                           FAC_1      FAC_2
                 V_1   0.6578207  0.4490503
                 V_2   0.6884175  0.2903916
                 V_3   0.5173663  0.6373427
                 V_4   0.7383147 -0.4130310
                 V_5   0.7438760 -0.3754484
                 V_6   0.6783135 -0.3550142

           Orthomax Prerotated Loadings: VARIMAX

                           FAC_1      FAC_2
                 V_1   0.2253107  0.7639433
                 V_2   0.3493867  0.6604354
                 V_3  -0.0025887  0.8208940
                 V_4   0.8330754  0.1472704
                 V_5   0.8136040  0.1798932
                 V_6   0.7499035  0.1542368

          Rotated Factor Loadings with Standard Errors
          --------------------------------------------

                          FAC_1                  FAC_2
     V_1   0.0871917  0.0848962  0.7604488  0.0894415
          [-0.079202, 0.253585] [ 0.585147, 0.935751]

     V_2   0.2386581  0.1126435  0.6261073  0.1088639
          [ 0.017881, 0.459435] [ 0.412738, 0.839477]

     V_3  -0.1647378  0.0534282  0.8666891  0.0901401
          [-0.269455,-0.060020] [ 0.690018, 1.043360]
```

```
V_4   0.8507661  0.0715458 -0.0133039  0.0644292
     [ 0.710539, 0.990993] [-0.139583, 0.112975]

V_5   0.8237643  0.0738101  0.0250612  0.0676976
     [ 0.679099, 0.968429] [-0.107624, 0.157746]

V_6   0.7615520  0.0892199  0.0108893  0.0845470
     [ 0.586684, 0.936420] [-0.154820, 0.176598]

     Factor Correlations with Standard Errors
     ----------------------------------------

                        FAC_1                FAC_2
FAC_1   1.0000000  0.0000000  0.3656592  0.0542791
       [ 1.000000, 1.000000] [ 0.259274, 0.472044]

FAC_2   0.3656592  0.0542791  1.0000000  0.0000000
       [ 0.259274, 0.472044] [ 1.000000, 1.000000]
```

(f) Standardized Unnormed Oblique Rotated PCA:

```
optn = [ "data"      "cor" ,
         "anal"      "cor" ,
         "nobs"        220 ,
         "vers"     "cev1" ,
         "comp"            ,
         "nfac"          2 ,
         "tech"   "nrridg" ,
         "cl"       "wald" ,
         "rvers"    "comm" ,
         "frot"   "quamin" ,
         "prin"          3 ];
< gof, est > = pca(corr,optn);
```

```
                ****************************
                Principal Component Analysis
                ****************************

  Input Data. . . . . . . . . . . . . . . . .Correlation Matrix
  Analysis of . . . . . . . . . . . . . . . .Correlation Matrix
  Number of Items . . . . . . . . . . . . . . . . .          6
  Specified Number of Factors . . . . . . . . . . .          2
  Number of Subjects. . . . . . . . . . . . . . . .        220
  Version . . . . . . . . . . . . . . . . .Standard Full EVD Code
  Rotation Method . . . . . . . . . Quartimin (Carroll, 1953)
```

```
Unnormed Rotation Method. . . . . . . . . . . . . . 0.0000000

               Unrotated Factor Loadings

                         FAC_1      FAC_2
               V_1   0.6578207  0.4490503
               V_2   0.6884175  0.2903916
               V_3   0.5173663  0.6373427
               V_4   0.7383147 -0.4130310
               V_5   0.7438760 -0.3754484
               V_6   0.6783135 -0.3550142

         Orthomax Prerotated Loadings: VARIMAX

                         FAC_1      FAC_2
               V_1   0.2253107  0.7639433
               V_2   0.3493867  0.6604354
               V_3  -0.0025887  0.8208940
               V_4   0.8330754  0.1472704
               V_5   0.8136040  0.1798932
               V_6   0.7499035  0.1542368

      Rotated Factor Loadings with Standard Errors
      --------------------------------------------

                     FAC_1                  FAC_2
     V_1   0.1070191  0.0961876  0.7532025  0.0606586
          [-0.081505, 0.295543] [ 0.634314, 0.872091]

     V_2   0.2551183  0.1190878  0.6197485  0.0923557
          [ 0.021710, 0.488526] [ 0.438735, 0.800762]

     V_3  -0.1423536  0.0356829  0.8590523  0.0348038
          [-0.212291,-0.072416] [ 0.790838, 0.927267]

     V_4   0.8511095  0.0324862 -0.0151831  0.0558954
          [ 0.787438, 0.914781] [-0.124736, 0.094370]

     V_5   0.8250821  0.0378689  0.0228900  0.0617596
          [ 0.750860, 0.899304] [-0.098157, 0.143937]

     V_6   0.7624512  0.0574010  0.0089956  0.0916217
          [ 0.649947, 0.874955] [-0.170580, 0.188571]

        Factor Correlations with Standard Errors
```

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

                                FAC_1                FAC_2
         FAC_1   1.0000000  0.0000000  0.3449275  0.0622212
                [ 1.000000, 1.000000] [ 0.222976, 0.466879]

         FAC_2   0.3449275  0.0622212  1.0000000  0.0000000
                [ 0.222976, 0.466879] [ 1.000000, 1.000000]
```

3.4 Function `randisc`

p = randisc(nsmp,"bin",n,p<,imet<,ipri>>)

p = randisc(nsmp,"hyp",n,m,N<,imet<,ipri>>)

p = randisc(nsmp,"poi",lambda<,imet<,ipri>>)

Purpose: The `randisc` function implements some algorithm for generationg discrete random variates. Usually the `rand` function can be used for obtaining those values. However, the methods which are implemented in `randisc` are especially designed for large amounts ($nsmp >> 1000$) of random numbers at the cost of some initialization overhead consisting especially of larger memory allocations.

Input: **nsmp** must be a positive integer defining the size of the sample of random numbers.

dist : specifies the distribution and should be one of the following:

- **"bin"** binomial (n, p)
- **"hyp"** hypergeometric (n, m, N)
- **"poi"** Poisson (λ)

(n, p) parameters $n \geq 0$, $0 \leq p \leq 1$ of the Binomial distribution

(n, m, N) parameters $n > 0$, $m > 0$, $1 \leq N \leq n+m$ of the hypergeometric distribution

(λ) parameter $\lambda \geq 0$ of the Poisson distribution

imet : specifies one of four methods:

- **"mww_a"** memory efficient method by Marsaglia, Tsang, & Wang (2004) (is default)
- **"mww_b"** fast but memory expensive method by Marsaglia, Tsang, & Wang (2004)
- **"patch"** patchwork method by Stadlober & Zechner (19999)
- **"ratun"** ratio of uniforms method by Stadlober (1989)

The distributional parameters can be specified either as scalars or as vectors of the equal dimension.

ipri : must be a nonnegative integer speifying the amount of printed output (default is 0, specifying no output.

Output: The only return object is either a *nsmp* vector or a $K \times nsmp$ matrix of the integer random variates.

Restrictions:

1. No missing values are permitted for the distributional parameters.
2.

Relationships: rand()

Examples:

1. Binomial Distribution

```
/* [1] Binomial (n,p) */
nsmp = 100000;
n = [  20  20  100  100 1000 1000 10000 10000 100000 100000 ];
p = [  .1  .4   .1   .4   .1   .4    .1    .4     .1     .4 ];
/* MTW, method=0: not so fast, but better in memory */
p = randisc(nsmp,"bin",n,p,"mtw_a",2);
```

```
                 Binomial(n=20, p=0.1)
     LowMem Method by Marsaglia, Tsang, and Wang (2004)
                 Array Size=14 Offset=0
              J table is 98.83 percent full
          Chisquare=  13.67 for df=8 : prob=0.90925


                 Binomial(n=20, p=0.4)
     LowMem Method by Marsaglia, Tsang, and Wang (2004)
                 Array Size=21 Offset=0
              J table is 97.27 percent full
         Chisquare= 317.72 for df=15 : prob=1.00000


                Binomial(n=100, p=0.1)
     LowMem Method by Marsaglia, Tsang, and Wang (2004)
                 Array Size=33 Offset=0
              J table is 96.48 percent full
         Chisquare=  12.64 for df=22 : prob=0.05719


                Binomial(n=100, p=0.4)
     LowMem Method by Marsaglia, Tsang, and Wang (2004)
                 Array Size=59 Offset=12
```

```
                J table is 93.75 percent full
          Chisquare= 900.33 for df=36 : prob=1.00000


                   Binomial(n=1000, p=0.1)
      LowMem Method by Marsaglia, Tsang, and Wang (2004)
                  Array Size=114 Offset=48
                J table is 89.84 percent full
          Chisquare=  45.52 for df=70 : prob=0.01024


                   Binomial(n=1000, p=0.4)
      LowMem Method by Marsaglia, Tsang, and Wang (2004)
                  Array Size=185 Offset=309
                J table is 83.59 percent full
          Chisquare= 122.66 for df=115 : prob=0.70478


                  Binomial(n=10000, p=0.1)
      LowMem Method by Marsaglia, Tsang, and Wang (2004)
                  Array Size=351 Offset=829
                J table is 72.66 percent full
          Chisquare= 188.86 for df=223 : prob=0.04693


                  Binomial(n=10000, p=0.4)
      LowMem Method by Marsaglia, Tsang, and Wang (2004)
                 Array Size=565 Offset=3719
                J table is 57.81 percent full
          Chisquare= 974.19 for df=365 : prob=1.00000


                  Binomial(n=100000, p=0.1)
      LowMem Method by Marsaglia, Tsang, and Wang (2004)
                 Array Size=1074 Offset=9467
                J table is 28.52 percent full
          Chisquare= 675.62 for df=706 : prob=0.21108


                  Binomial(n=100000, p=0.4)
      LowMem Method by Marsaglia, Tsang, and Wang (2004)
                Array Size=1727 Offset=39138
                J table is  0.00 percent full
         Chisquare=1270.71 for df=1153 : prob=0.99149

/* [2] Binomial (n,p) */
nsmp = 100000;
```

```
n = [  20  20  100  100 1000 1000 10000 10000 100000 100000 ];
p = [  .1  .4   .1   .4   .1   .4    .1    .4     .1     .4 ];
/* MTW, method=1: very fast, but needs lots of memory */
p = randisc(nsmp,"bin",n,p,"mtw_b",2);
```

```
                   Binomial(n=20, p=0.1)
   Fast 5Table Method by Marsaglia, Tsang, and Wang (2004)
                  Array Size=14 Offset=0
         Table Sizes: 61 187 313 441 447 Total=1449
          Chisquare=   9.73 for df=8 : prob=0.71532


                   Binomial(n=20, p=0.4)
   Fast 5Table Method by Marsaglia, Tsang, and Wang (2004)
                  Array Size=21 Offset=0
         Table Sizes: 57 439 566 630 638 Total=2330
         Chisquare=  22.90 for df=15 : prob=0.91376


                  Binomial(n=100, p=0.1)
   Fast 5Table Method by Marsaglia, Tsang, and Wang (2004)
                  Array Size=33 Offset=0
        Table Sizes: 57 440 496 1008 1024 Total=3025
         Chisquare=  23.57 for df=22 : prob=0.62992


                  Binomial(n=100, p=0.4)
   Fast 5Table Method by Marsaglia, Tsang, and Wang (2004)
                  Array Size=59 Offset=12
        Table Sizes: 51 813 1191 1574 1662 Total=5291
         Chisquare=  38.16 for df=36 : prob=0.62841


                  Binomial(n=1000, p=0.1)
   Fast 5Table Method by Marsaglia, Tsang, and Wang (2004)
                 Array Size=114 Offset=48
       Table Sizes: 41 1432 2514 2887 3644 Total=10518
         Chisquare=  62.38 for df=70 : prob=0.27018


                  Binomial(n=1000, p=0.4)
   Fast 5Table Method by Marsaglia, Tsang, and Wang (2004)
                 Array Size=185 Offset=309
       Table Sizes: 31 2058 3389 4211 4923 Total=14612
         Chisquare=  99.25 for df=115 : prob=0.14784
```

```
                   Binomial(n=10000, p=0.1)
   Fast 5Table Method by Marsaglia, Tsang, and Wang (2004)
                   Array Size=351 Offset=829
          Table Sizes: 0 3993 6452 8803 10042 Total=29290
            Chisquare= 225.94 for df=223 : prob=0.56757


                   Binomial(n=10000, p=0.4)
   Fast 5Table Method by Marsaglia, Tsang, and Wang (2004)
                  Array Size=565 Offset=3719
         Table Sizes: 0 3938 9902 13184 16370 Total=43394
            Chisquare= 379.17 for df=365 : prob=0.70629


                   Binomial(n=100000, p=0.1)
   Fast 5Table Method by Marsaglia, Tsang, and Wang (2004)
                  Array Size=1074 Offset=9467
         Table Sizes: 0 3812 17758 26282 30066 Total=77918
            Chisquare= 752.78 for df=706 : prob=0.89181


                   Binomial(n=100000, p=0.4)
   Fast 5Table Method by Marsaglia, Tsang, and Wang (2004)
                 Array Size=1727 Offset=39138
        Table Sizes: 0 3656 27515 40516 48878 Total=120565
           Chisquare=1116.84 for df=1153 : prob=0.22748
```

```
/* [3] Binomial (n,p) */
nsmp = 100000;
n = [  20  20  100  100 1000 1000 10000 10000 100000 100000 ];
p = [  .1  .4   .1   .4   .1   .4    .1    .4     .1     .4 ];
/* patchwork method by Sradlober & Zechner, 1999 */
p = randisc(nsmp,"bin",n,p,"patch",2);
```

```
                    Binomial(n=20, p=0.1)
      Patchwork Method by Stadlober and Zechner (1999)


                    Binomial(n=20, p=0.4)
      Patchwork Method by Stadlober and Zechner (1999)


                   Binomial(n=100, p=0.1)
      Patchwork Method by Stadlober and Zechner (1999)
```

Binomial(n=100, p=0.4)
Patchwork Method by Stadlober and Zechner (1999)

Binomial(n=1000, p=0.1)
Patchwork Method by Stadlober and Zechner (1999)

Binomial(n=1000, p=0.4)
Patchwork Method by Stadlober and Zechner (1999)

Binomial(n=10000, p=0.1)
Patchwork Method by Stadlober and Zechner (1999)

Binomial(n=10000, p=0.4)
Patchwork Method by Stadlober and Zechner (1999)

Binomial(n=100000, p=0.1)
Patchwork Method by Stadlober and Zechner (1999)

Binomial(n=100000, p=0.4)
Patchwork Method by Stadlober and Zechner (1999)

```
/* [4] Binomial (n,p) */
nsmp = 100000;
n = [  20  20  100  100 1000 1000 10000 10000 100000 100000 ];
p = [  .1  .4   .1   .4   .1   .4    .1    .4     .1     .4 ];
/* Ratio of Uniforms by Stadlober, 1989 */
p = randisc(nsmp,"bin",n,p,"ratun",2);
```

Binomial(n=20, p=0.1)
Ratio of Uniforms Method by Stadlober (1989)

Binomial(n=20, p=0.4)
Ratio of Uniforms Method by Stadlober (1989)

Binomial(n=100, p=0.1)
Ratio of Uniforms Method by Stadlober (1989)

Binomial(n=100, p=0.4)
Ratio of Uniforms Method by Stadlober (1989)

```
            Binomial(n=1000, p=0.1)
  Ratio of Uniforms Method by Stadlober (1989)

            Binomial(n=1000, p=0.4)
  Ratio of Uniforms Method by Stadlober (1989)

           Binomial(n=10000, p=0.1)
  Ratio of Uniforms Method by Stadlober (1989)

           Binomial(n=10000, p=0.4)
  Ratio of Uniforms Method by Stadlober (1989)

           Binomial(n=100000, p=0.1)
  Ratio of Uniforms Method by Stadlober (1989)

           Binomial(n=100000, p=0.4)
  Ratio of Uniforms Method by Stadlober (1989)
```

2. Poisson Distribution

```
/*--- [1] Poisson (lambda) ---*/
nsmp = 100000;
lambda = [ 1. 10. 25. 100. 250. 1000. ];
/* MTW, method=0: not so fast, but better in memory */
p = randisc(nsmp,"pois",lambda,"mtw_a",2);
```

```
                   Poisson(lambda=1)
     LowMem Method by Marsaglia, Tsang, and Wang (2004)
                Array Size=13 Offset=0
             J table is 98.83 percent full
         Chisquare=   6.92 for df=6 : prob=0.67164

                  Poisson(lambda=10)
     LowMem Method by Marsaglia, Tsang, and Wang (2004)
                Array Size=35 Offset=0
             J table is 96.09 percent full
        Chisquare=  32.95 for df=23 : prob=0.91814

                  Poisson(lambda=25)
     LowMem Method by Marsaglia, Tsang, and Wang (2004)
```

```
                Array Size=60 Offset=2
              J table is 93.75 percent full
         Chisquare=  38.21 for df=37 : prob=0.58570


                  Poisson(lambda=100)
     LowMem Method by Marsaglia, Tsang, and Wang (2004)
                Array Size=120 Offset=46
              J table is 89.06 percent full
         Chisquare= 753.53 for df=74 : prob=1.00000


                  Poisson(lambda=250)
     LowMem Method by Marsaglia, Tsang, and Wang (2004)
                Array Size=188 Offset=162
              J table is 83.59 percent full
         Chisquare= 868.53 for df=118 : prob=1.00000


                  Poisson(lambda=1000)
     LowMem Method by Marsaglia, Tsang, and Wang (2004)
                Array Size=370 Offset=821
              J table is 71.48 percent full
         Chisquare= 225.78 for df=235 : prob=0.34462
```

```
/*--- [2] Poisson (lambda) ---*/
nsmp = 100000;
lambda = [ 1. 10. 25. 100. 250. 1000. ];
/* MTW, method=1: very fast, but needs lots of memory */
p = randisc(nsmp,"pois",lambda,"mtw_b",2);
```

```
                    Poisson(lambda=1)
   Fast 5Table Method by Marsaglia, Tsang, and Wang (2004)
                 Array Size=13 Offset=0
          Table Sizes: 60 252 251 315 322 Total=1200
           Chisquare=   2.19 for df=6 : prob=0.09841


                   Poisson(lambda=10)
   Fast 5Table Method by Marsaglia, Tsang, and Wang (2004)
                 Array Size=35 Offset=0
          Table Sizes: 58 371 818 881 959 Total=3087
          Chisquare=   8.67 for df=23 : prob=0.00305


                   Poisson(lambda=25)
```

```
   Fast 5Table Method by Marsaglia, Tsang, and Wang (2004)
                Array Size=60 Offset=2
      Table Sizes: 51 815 1064 1507 1857 Total=5294
        Chisquare=  31.58 for df=37 : prob=0.27913


                  Poisson(lambda=100)
   Fast 5Table Method by Marsaglia, Tsang, and Wang (2004)
               Array Size=120 Offset=46
     Table Sizes: 41 1437 2190 3147 3387 Total=10202
        Chisquare=  77.56 for df=74 : prob=0.63404


                  Poisson(lambda=250)
   Fast 5Table Method by Marsaglia, Tsang, and Wang (2004)
               Array Size=188 Offset=162
     Table Sizes: 31 2062 3128 4528 5116 Total=14865
        Chisquare= 112.92 for df=118 : prob=0.38506


                 Poisson(lambda=1000)
   Fast 5Table Method by Marsaglia, Tsang, and Wang (2004)
               Array Size=370 Offset=821
     Table Sizes: 0 3993 6449 8975 11332 Total=30749
        Chisquare= 228.27 for df=235 : prob=0.38880


/*--- [3] Poisson (lambda) ---*/
nsmp = 100000;
lambda = [ 1. 10. 25. 100. 250. 1000. ];
/* patchwork method by Stadlober and Zechner, 1999 */
p = randisc(nsmp,"pois",lambda,"patch",2);


                   Poisson(lambda=1)
      Patchwork Method by Stadlober and Zechner (1999)


                  Poisson(lambda=10)
      Patchwork Method by Stadlober and Zechner (1999)


                  Poisson(lambda=25)
      Patchwork Method by Stadlober and Zechner (1999)


                  Poisson(lambda=100)
      Patchwork Method by Stadlober and Zechner (1999)
```

Poisson(lambda=250)
Patchwork Method by Stadlober and Zechner (1999)

Poisson(lambda=1000)
Patchwork Method by Stadlober and Zechner (1999)

```
/*--- [4] Poisson (lambda) ---*/
nsmp = 100000;
lambda = [ 1. 10. 25. 100. 250. 1000. ];
/* Ratio of Uniforms by Stadlober, 1989 */
p = randisc(nsmp,"pois",lambda,"ratun",2);
```

Poisson(lambda=1)
Ratio of Uniforms Method by Stadlober (1989)

Poisson(lambda=10)
Ratio of Uniforms Method by Stadlober (1989)

Poisson(lambda=25)
Ratio of Uniforms Method by Stadlober (1989)

Poisson(lambda=100)
Ratio of Uniforms Method by Stadlober (1989)

Poisson(lambda=250)
Ratio of Uniforms Method by Stadlober (1989)

Poisson(lambda=1000)
Ratio of Uniforms Method by Stadlober (1989)

3. Hypergeometric Distribution

```
/* [1] Hypergeo (n,m,k) */
nsmp = 100000;
bad  = [  20  100  100 1000 1000 1000 10000 10000 10000 10000 ];
good = [  20  100  100  100 1000 1000  1000  1000 10000 10000 ];
sel  = [  20   20  100  100  100 1000  1000  1000  1000 10000 ];
n = sel; m = good; N = good + bad;
/* MTW, method=0: not so fast, but better in memory */
p = randisc(nsmp,"hyp",n,m,N,"mtw_a",2);
```

```
Hypergeometric(n=20,m=20,N=40)
LowMem Method by Marsaglia, Tsang, and Wang (2004)
Array Size=19 Offset=1
J table is 98.05 percent full
Chisquare=   9.16 for df=12 : prob=0.31070


Hypergeometric(n=20,m=100,N=200)
LowMem Method by Marsaglia, Tsang, and Wang (2004)
Array Size=21 Offset=0
J table is 98.05 percent full
Chisquare= 838.94 for df=16 : prob=1.00000


Hypergeometric(n=100,m=100,N=200)
LowMem Method by Marsaglia, Tsang, and Wang (2004)
Array Size=43 Offset=29
J table is 96.09 percent full
Chisquare=  24.46 for df=26 : prob=0.45042


Hypergeometric(n=100,m=100,N=1100)
LowMem Method by Marsaglia, Tsang, and Wang (2004)
Array Size=30 Offset=0
J table is 96.88 percent full
Chisquare= 158.23 for df=19 : prob=1.00000


Hypergeometric(n=100,m=1000,N=2000)
LowMem Method by Marsaglia, Tsang, and Wang (2004)
Array Size=59 Offset=21
J table is 93.75 percent full
Chisquare=  33.51 for df=36 : prob=0.41220


Hypergeometric(n=1000,m=1000,N=2000)
LowMem Method by Marsaglia, Tsang, and Wang (2004)
Array Size=135 Offset=433
J table is 87.89 percent full
Chisquare= 900.31 for df=84 : prob=1.00000


Hypergeometric(n=1000,m=1000,N=11000)
LowMem Method by Marsaglia, Tsang, and Wang (2004)
Array Size=104 Offset=43
J table is 89.84 percent full
Chisquare=  63.85 for df=64 : prob=0.51810
```

```
         Hypergeometric(n=1000,m=1000,N=11000)
   LowMem Method by Marsaglia, Tsang, and Wang (2004)
               Array Size=104 Offset=43
              J table is 89.84 percent full
       Chisquare=  59.76 for df=64 : prob=0.37303


        Hypergeometric(n=1000,m=10000,N=20000)
   LowMem Method by Marsaglia, Tsang, and Wang (2004)
               Array Size=183 Offset=409
              J table is 82.81 percent full
       Chisquare= 814.41 for df=114 : prob=1.00000


        Hypergeometric(n=10000,m=10000,N=20000)
   LowMem Method by Marsaglia, Tsang, and Wang (2004)
              Array Size=413 Offset=4794
              J table is 64.06 percent full
       Chisquare= 276.90 for df=262 : prob=0.74786
```

```
/* [2] Hypergeo (n,m,k) */
nsmp = 100000;
bad  = [  20  100  100 1000 1000 1000 10000 10000 10000 10000 ];
good = [  20  100  100  100 1000 1000  1000  1000 10000 10000 ];
sel  = [  20   20  100  100  100 1000  1000  1000  1000 10000 ];
n = sel; m = good; N = good + bad;
/* MTW, method=1: very fast, but needs lots of memory */
p = randisc(nsmp,"hyp",n,m,N,"mtw_b",2);
```

```
              Hypergeometric(n=20,m=20,N=40)
  Fast 5Table Method by Marsaglia, Tsang, and Wang (2004)
                 Array Size=19 Offset=1
         Table Sizes: 59 314 380 250 382 Total=1385
         Chisquare=   5.31 for df=12 : prob=0.05335


             Hypergeometric(n=20,m=100,N=200)
  Fast 5Table Method by Marsaglia, Tsang, and Wang (2004)
                 Array Size=21 Offset=0
         Table Sizes: 57 438 630 632 515 Total=2272
         Chisquare=  12.96 for df=16 : prob=0.32398


             Hypergeometric(n=100,m=100,N=200)
  Fast 5Table Method by Marsaglia, Tsang, and Wang (2004)
                 Array Size=43 Offset=29
```

```
     Table Sizes: 55 562 879 1070 1151 Total=3717
      Chisquare=  33.21 for df=26 : prob=0.84380


           Hypergeometric(n=100,m=100,N=1100)
Fast 5Table Method by Marsaglia, Tsang, and Wang (2004)
                 Array Size=30 Offset=0
      Table Sizes: 56 502 627 819 830 Total=2834
      Chisquare=  13.95 for df=19 : prob=0.21333


           Hypergeometric(n=100,m=1000,N=2000)
Fast 5Table Method by Marsaglia, Tsang, and Wang (2004)
                 Array Size=59 Offset=21
     Table Sizes: 51 816 1001 1439 2110 Total=5417
      Chisquare=  42.97 for df=36 : prob=0.80261


          Hypergeometric(n=1000,m=1000,N=2000)
Fast 5Table Method by Marsaglia, Tsang, and Wang (2004)
                Array Size=135 Offset=433
    Table Sizes: 40 1500 2256 3010 3963 Total=10769
      Chisquare=  82.52 for df=84 : prob=0.47490


          Hypergeometric(n=1000,m=1000,N=11000)
Fast 5Table Method by Marsaglia, Tsang, and Wang (2004)
                Array Size=104 Offset=43
    Table Sizes: 40 1503 2072 2514 2943 Total=9072
      Chisquare=  73.83 for df=64 : prob=0.81233


          Hypergeometric(n=1000,m=1000,N=11000)
Fast 5Table Method by Marsaglia, Tsang, and Wang (2004)
                Array Size=104 Offset=43
    Table Sizes: 40 1503 2072 2514 2943 Total=9072
      Chisquare=  82.85 for df=64 : prob=0.94337


         Hypergeometric(n=1000,m=10000,N=20000)
Fast 5Table Method by Marsaglia, Tsang, and Wang (2004)
                Array Size=183 Offset=409
    Table Sizes: 31 2060 3256 4531 4936 Total=14814
      Chisquare= 126.20 for df=114 : prob=0.79523


         Hypergeometric(n=10000,m=10000,N=20000)
Fast 5Table Method by Marsaglia, Tsang, and Wang (2004)
```

```
                  Array Size=413 Offset=4794
        Table Sizes: 0 3982 7131 10383 11323 Total=32819
           Chisquare= 215.92 for df=262 : prob=0.01720
```

```
/* [3] Hypergeo (n,m,k) */
nsmp = 100000;
bad  = [  20  100  100 1000 1000 1000 10000 10000 10000 10000 ];
good = [  20  100  100  100 1000 1000  1000  1000 10000 10000 ];
sel  = [  20   20  100  100  100 1000  1000  1000  1000 10000 ];
n = sel; m = good; N = good + bad;
/* patchwork method by Stadlober and Zechner, 1999 */
p = randisc(nsmp,"hyp",n,m,N,"patch",2);
```

```
                Hypergeometric(n=20,m=20,N=40)
      Patchwork Method by Stadlober and Zechner (1999)

               Hypergeometric(n=20,m=100,N=200)
      Patchwork Method by Stadlober and Zechner (1999)

               Hypergeometric(n=100,m=100,N=200)
      Patchwork Method by Stadlober and Zechner (1999)

              Hypergeometric(n=100,m=100,N=1100)
      Patchwork Method by Stadlober and Zechner (1999)

              Hypergeometric(n=100,m=1000,N=2000)
      Patchwork Method by Stadlober and Zechner (1999)

             Hypergeometric(n=1000,m=1000,N=2000)
      Patchwork Method by Stadlober and Zechner (1999)

             Hypergeometric(n=1000,m=1000,N=11000)
      Patchwork Method by Stadlober and Zechner (1999)

             Hypergeometric(n=1000,m=1000,N=11000)
      Patchwork Method by Stadlober and Zechner (1999)

            Hypergeometric(n=1000,m=10000,N=20000)
      Patchwork Method by Stadlober and Zechner (1999)
```

Hypergeometric(n=10000,m=10000,N=20000)
Patchwork Method by Stadlober and Zechner (1999)

```
/* [4] Hypergeo (n,m,k) */
nsmp = 100000;
bad  = [  20  100  100 1000 1000 1000 10000 10000 10000 10000 ];
good = [  20  100  100  100 1000 1000  1000  1000 10000 10000 ];
sel  = [  20   20  100  100  100 1000  1000  1000  1000 10000 ];
n = sel; m = good; N = good + bad;
/* Ratio of Uniforms by Stadlober, 1989 */
p = randisc(nsmp,"hyp",n,m,N,"ratun",2);
```

Hypergeometric(n=20,m=20,N=40)
Ratio of Uniforms Method by Stadlober (1989)

Hypergeometric(n=20,m=100,N=200)
Ratio of Uniforms Method by Stadlober (1989)

Hypergeometric(n=100,m=100,N=200)
Ratio of Uniforms Method by Stadlober (1989)

Hypergeometric(n=100,m=100,N=1100)
Ratio of Uniforms Method by Stadlober (1989)

Hypergeometric(n=100,m=1000,N=2000)
Ratio of Uniforms Method by Stadlober (1989)

Hypergeometric(n=1000,m=1000,N=2000)
Ratio of Uniforms Method by Stadlober (1989)

Hypergeometric(n=1000,m=1000,N=11000)
Ratio of Uniforms Method by Stadlober (1989)

Hypergeometric(n=1000,m=1000,N=11000)
Ratio of Uniforms Method by Stadlober (1989)

Hypergeometric(n=1000,m=10000,N=20000)
Ratio of Uniforms Method by Stadlober (1989)

```
Hypergeometric(n=10000,m=10000,N=20000)
Ratio of Uniforms Method by Stadlober (1989)
```

3.5 Function `scalpha`

vest = scalpha(data<,optn>)

Purpose: The `scalpha` function computes the sample coefficient α (Cronbach, 1951) together with its normal theory approximate standard error (ASE) and corresponding confidence intervals. If the data are $N \times n$ raw data and not only a correlation or covariance matrix the nonnormal ASE (Ferguson, 1996) and corresponding CIs are computed and returned in addition to a set of multivariate kurtosis measures.

Input: data This should be either

1. a $N \times n$ data matrix or
2. a symmetric $n \times n$ correlation or covariance matrix which can be augmented with one or two additional rows containing mean vector and standard deviations.

optn This is a vector of options:

Index	Meaning
[**1**]	amount of printed output
[**2**]	form of input data =0: $N \times n$ raw data set =1: (augmented) correlation matrix =2: (augmented) covariance matrix
[**3**]	probability α for the confidence interval (default is $\alpha = .05$)
[**4**]	N the number of observations (needed for CORR or COV matrix input)
[**5**]	unbiased (=0) vs. biased (=1) kurtosis (only for raw data input)

Output: The only return argument `vest` is a vector with 5 for CORR or COV matrix input or 15 entries for raw data set input:

Index	Meaning
[**1**]	=0: correct termination
[**2**]	coefficient α
[**3**]	normal theory approximate standard error
[**4**]	lower CI for normal ASE
[**5**]	upper CI for normal ASE
[**6**]	non normal theory approximate standard error
[**7**]	lower CI for nonnormal ASE
[**8**]	upper CI for nonnormal ASE
[**9**]	unused
[**10**]	Mardias Multivariate Kurtosis
[**11**]	Relative Multivariate Kurtosis
[**12**]	Normalized Multivariate Kurtosis
[**13**]	Mardia Based Kappa (Browne, 1982)
[**14**]	Mean Scaled Univariate Kurtosis
[**15**]	Adjusted Mean Scaled Univ. Kurtosis

For erroneous input arguments the function may return a scalar missing value.

Restrictions: 1. There should be no missing values in the first input argument `data`.

2.

Relationships: mardia()

Examples: 1. 2options 429×5 data set:

```
  print "2Options Example by Albert M.-O.";
  options NOECHO;
  two = [
#include "..\\tdata\\twopts.dat"
         ];
  options ECHO;
  data = shape(two,.,5);
  print "nr=", nr = nrow(data);
  name = [ "V1" : "V5" ];
  data = cname(data,name);

  covm = bivar(data,"cov");
  print "COVM=", covm;

  /* alfa=0.614611, NTASE= 0.0432907, [0.50,0.73] */
  /*              NNTASE= 0.0283319, [0.54,0.69] */
  optn = [ 3  ,  /* ipri */
           0  ,  /* idat */
          .05    /* alfa */
             ];
  vest = scalpha(data,optn);
```

```
print "vest=",vest;

   Name            Mean     Std Dev    Skewness    Kurtosis
   ----------------------------------------------------------
   V1       0.736596737  0.44099354 -1.07427544 -0.85191868
   V2       0.522144522  0.50009258 -0.08866557 -2.00618411
   V3       0.827505828  0.37825050 -1.73371878  1.01281434
   V4       0.720279720  0.44938579 -0.98150906 -1.04396728
   V5       0.358974359  0.48025973  0.58797794 -1.66595207


                     Covariance Matrix

               V1         V2         V3         V4         V5
   V1   0.1944753
   V2   0.0630896  0.2500926
   V3   0.0759536  0.0365390  0.1430734
   V4   0.0663355  0.0435919  0.0497680  0.2019476
   V5   0.0293554  0.0551162  0.0316918  0.0352265  0.2306494
      Determinant =  1.84868243e-004 (Ln = -8.5959e+000)


   Number of Observations . . . . . . . . . . . . .       429
   Alpha. . . . . . . . . . . . . . . . . . . . . .    0.6103
   Normal ASE of Alpha. . . . . . . . . . . . . . .    0.0299
   Normal CI of Alpha . . . . . . . . . . . . [ 0.5518, 0.6688]
   NonNormal ASE of Alpha . . . . . . . . . . . . .    0.0289
   NonNormal CI of Alpha. . . . . . . . . . . [ 0.5536, 0.6670]
   Mardia's Multivariate Kurtosis . . . . . . . . .   -1.2032
   Relative Multivariate Kurtosis . . . . . . . . .    0.9656
   Normalized Multivariate Kurtosis . . . . . . . .   -1.4893
   Mardia Based Kappa (Browne, 1982). . . . . . . .   -0.0344
   Mean Scaled Univariate Kurtosis  . . . . . . . .   -0.3037
   Adjusted Mean Scaled Univariate Kurtosis . . . .   -0.1607
Mean Scaled Univariate Kurtosis smaller than lower bound -0.285714


   Observation numbers with largest contribution to kurtosis


             109         10        380        293         26
       235.4231  235.4231  130.9013  130.6617  130.6617
```

2. BHS 393×20 data set:

```
print "Rec_BHS Example by Albert M.-O.";
```

```
  options NOECHO;
  bhs = [
#include "..\\tdata\\rec_bhs.dat"
         ];
  options ECHO;
  data = shape(bhs,.,20);
  print "nr=", nr = nrow(data);
  name = [ "V1" : "V20" ];
  data = cname(data,name);

  /* alfa=0.848917, NTASE= 0.015043 , [0.81,0.89] */
  /*               NNTASE= 0.0166554, [0.81,0.89] */
  optn = [ 3  ,  /* ipri */
           0  ,  /* idat */
          .05    /* alfa */
             ];
  /* options debug="bialf*=3"; */
  vest = scalpha(data,optn);
  print "vest=",vest;
```

```
     Name             Mean     Std Dev    Skewness    Kurtosis
     ---------------------------------------------------------
     V1       0.101781170  0.30274574  2.63408518  4.97628207
     V2       0.541984733  0.49886926 -0.16853524 -1.98678201
     V3       0.139949109  0.34737646  2.07562822  2.32591215
     V4       0.195928753  0.39741989  1.53218718  0.35022102
     V5       0.086513995  0.28147989  2.94170761  6.70469271
     V6       0.213740458  0.41046825  1.39658207 -0.04998490
     V7       0.272264631  0.44569278  1.02324811 -0.96032713
     V8       0.089058524  0.28519127  2.88555803  6.37498156
     V9       0.071246819  0.25756458  3.33355235  9.18250293
     V10      0.407124682  0.49192472  0.37808351 -1.87135939
     V11      0.078880407  0.26989561  3.12460906  7.82275120
     V12      0.137404580  0.34471281  2.10644904  2.45579701
     V13      0.083969466  0.27769554  3.00014578  7.05459022
     V14      0.386768448  0.48763066  0.46501024 -1.79750922
     V15      0.066157761  0.24887450  3.49090073  10.2645657
     V16      0.501272265  0.50063573 -0.00508911 -2.01537816
     V17      0.201017812  0.40127219  1.49208072  0.22799692
     V18      0.183206107  0.38732850  1.63788501  0.68786387
     V19      0.229007634  0.42072979  1.28985194 -0.33891017
     V20      0.117048346  0.32188767  2.38246028  3.70430091

     Number of Observations . . . . . . . . . . . . . .       393
     Alpha. . . . . . . . . . . . . . . . . . . . . .      0.8486
```

```
Normal ASE of Alpha. . . . . . . . . . . . . .      0.0109
Normal CI of Alpha . . . . . . . . . . . [ 0.8272, 0.8699]
NonNormal ASE of Alpha . . . . . . . . . . .      0.0168
NonNormal CI of Alpha. . . . . . . . . . [ 0.8157, 0.8814]
Mardia's Multivariate Kurtosis . . . . . . . .   249.9324
Relative Multivariate Kurtosis . . . . . . . .     1.5680
Normalized Multivariate Kurtosis . . . . . . .    83.5117
Mardia Based Kappa (Browne, 1982). . . . . . .     0.5680
Mean Scaled Univariate Kurtosis  . . . . . . .     0.8852
Adjusted Mean Scaled Univariate Kurtosis . . .     1.0074

Observation numbers with largest contribution to kurtosis


         128       119       285       354        85
    1566.637  1503.105  1270.440  1252.466  1209.176
```

3.6 Function `screetst`

<nval,infl,lrgof,malin> = screetst(eval<,optn>)

Purpose: The `screetst()` function incorporates the scree test by Cattell (1966), Cattell & Vogelmann (1977) and Horn & Engstrom (1979). We are not aware of any computational form of this usually visual test of the plot. The examples illustrate that a goodness of fit test of the trivial stepwise regression (starting from the end) does not work properly, however, that influential measures like studendized residual and Cook's distance w.r.t. to any newly added point show very promising results. In this form this test is also applicable for testing the dimensionality of more general eigen and singular value distributions. There are three approaches implemented:

influential case: Starting with the $k = 3$ smallest (eigen-) or singular values the next larger value x_k ist tested for being outlier (misfit) with respect to a linear regression computed from the remaining $k-1$ smaller values.

linear GOF: Stepwise the linearity of the regression of k points ist tested. The values are processed starting from the $k = 3$ smallest and adding in increasing order.

Malinowski: Assuming a $m \times n$ data matrix $\mathbf{X}$, $m \geq n$, with a given set of n singular values σ_k or eigenvalues λ_k of the $\mathbf{X}^T\mathbf{X}$ matrix a number of tests for a significant number of dimensions $r \leq n$ developed by Malinowski (1991) is implemented. The input requires not only the set of n eigenvalues but also the scalar m with $m \geq n$.

The second approach seems to work much better than the first. The third table of dimension indicators is only computed when the number of rows (observations) m is specified. There are a number of differences between the *percent significance level* computed by CMAT to those listed in tables 4.4, 4.5, and 7.1 of the Malinowski book. Even though both programs, CMAT and Malinowski's sfa.m Matlab program agree in the $F(1, s-n)$ value in formula (4.83) on page 119, we disagree when computing the corresponding probability value. It seems to me that program sfa.m is not correct, at least as printed in the books second edition.)

Input: eval This should be either a n vector or a $m \times n$ matrix of eigenvalues. If the input is a matrix, each of the m rows is treated separately and the output argument `nval` is an $m \times 2$ matrix instead of a 2 vector. If the values (in each of the m rows) are not sorted in descending order, they will be sorted before the algorithm starts.

optn This is a vector of options:

Index	Meaning
[1]	amount of printed output
[2]	probability α of the test for linearity (default is $\alpha = .10$)
[3]	threshold for Cook's distance
[4]	number of rows (observations) m (only for tests by Malinowski).

Output: nval This is either a 2 vector (for $m = 1$) or a $m \times 2$ matrix and contains the number of significant leading eigen or singular values constituting a nonlinear curve.

infl This is a $n \times 6 * m$ matrix containing some influence indicators like studendized residuals DFFITS values and Cook's distance. The first two rows correspond to values n and $n-1$ and contain all missing values. The next $k = 3, .., n$ rows correspond to adding the cases $n-2, n-3, \ldots, 2, 1$ to the linear regression.

lrgof This is a $n \times 6 * m$ matrix containing some goodness of fit values for a linear regression like SSE, MSE, p value, R^2, and BIC value. The first two rows correspond to values n and $n-1$ and contain all missing values. The next $k = 3, .., n$ rows correspond to adding the cases $n-2, n-3, \ldots, 2, 1$ to the linear regression.

malin This is a $n \times 6 * m$ matrix containing the dimensional indicators by Malinowski (1991).

Restrictions:

1. There should be no missing values in the first input argument `eval`.
2. This function assumes that $n \gg 3$.
3. When m is specified, this function assumes that $m \geq n$.

Relationships: factor(), svd()

Examples: 1. TWENTY FOUR PSYCHOLOGICAL TESTS:

```
#include "..\\tdata\\sem28.dat"

  nam2 = [ "TEST1" : "TEST24" ];
  psy24 = cname(psy24,nam2);

   evl = dia2vec(eig(psy24));
   eval = evl[>];
   print "eigval=", eval;

   optn = [ 1, .10, ., 145 ];
   < nval,infl,prob,malin > = screetst(eval,optn);
```

```
               Diag of Hat Matrix and Cook's Distance
               **************************************

   Case   DiagHat    StudRes   tProb    DFFITS     CooksD   FProb

      3  0.833333       .        .         .       2.500000* 0.5918
      4  0.700000   0.186845  0.4412  0.285410   0.078711  0.0730
      5  0.600000   1.028160  0.2060  1.259233   0.778023* 0.4657
      6  0.523810   1.256308  0.1490  1.317627*  0.758420* 0.4743
      7  0.464286   1.650693* 0.0871  1.536711*  0.877902* 0.5288
      8  0.416667   0.402121  0.3521  0.339854   0.067130  0.0642
      9  0.377778   1.061623  0.1646  0.827210   0.336039* 0.2745
     10  0.345455  -1.652473* 0.0712 -1.200494*  0.592430* 0.4245
     11  0.318182  -1.089559  0.1538 -0.744310   0.271357  0.2316
     12  0.294872  -0.488872  0.3183 -0.316139   0.054088  0.0524
     13  0.274725   0.432098  0.3374  0.265938   0.038185  0.0373
     14  0.257143  -2.248165* 0.0230 -1.322704*  0.653863* 0.4624
     15  0.241667  -2.665872* 0.0103 -1.504935*  0.770476* 0.5172
     16  0.227941  -3.088375* 0.0043 -1.678094*  0.874609* 0.5614
     17  0.215686  -1.637903* 0.0619 -0.858924*  0.331668* 0.2771
     18  0.204678  -2.252596* 0.0198 -1.142741*  0.520413* 0.3960
     19  0.194737  -1.640553* 0.0602 -0.806762*  0.295984* 0.2524
     20  0.185714  -2.035353* 0.0289 -0.972017*  0.402190* 0.3253
     21  0.177489  -9.107679* 0.0000 -4.230803*  1.684472* 0.7879
     22  0.169960  -4.337604* 0.0002 -1.962793*  1.018796* 0.6210
     23  0.163043  -4.556858* 0.0001 -2.011249*  1.041919* 0.6297
     24  0.156667  -26.66338* 0.0000 -11.49222*  1.984849* 0.8387

      Recommended Dimension Based on Studendized Residual : 11
```

Recommended Dimension Based on Cooks D : 11
Threshold for Cooks Distance: 0.25

Fit Linear Regression Through N Points (Cases)
****

N	Eval	MSE	Rsquared	Fval	Prob	BIC
1	0.1724946	0.000000	1.000000	.	.	.
2	0.1896638	0.000000	1.000000	.	.	.
3	0.2681326	6.3e-004	0.879553	7.302399	0.226	-19.42300
4	0.2971980	3.2e-004	0.940481	31.60244	0.030	-26.91080
5	0.3157519	3.3e-004	0.940023	47.01882	0.006	-34.85482
6	0.3337787	3.8e-004	0.932974	55.67842	0.002	-42.21674
7	0.3403650	5.1e-004	0.908267	49.50625	0.001	-48.16537
8	0.3819639	4.4e-004	0.930810	80.71717	0.000	-57.06866
9	0.3897347	4.5e-004	0.934005	99.06916	0.000	-64.73346
10	0.4774630	5.4e-004	0.942961	132.2542	0.000	-70.56605
11	0.5094265	5.5e-004	0.954660	189.5012	0.000	-77.92145
12	0.5330397	5.1e-004	0.965013	275.8186	0.000	-86.42226
13	0.5433084	4.7e-004	0.971128	369.9926	0.000	-95.04303
14	0.6393667	6.3e-004	0.969603	382.7790	0.000	-98.66089
15	0.7068670	9.3e-004	0.965099	359.4777	0.000	-100.2812
16	0.7902186	1.5e-003	0.956727	309.5240	0.000	-99.67039
17	0.8159449	1.7e-003	0.959982	359.8326	0.000	-104.3910
18	0.9012169	2.1e-003	0.958919	373.4772	0.000	-106.7268
19	0.9429365	2.3e-003	0.961654	426.3339	0.000	-111.1146
20	1.0252044	2.7e-003	0.961993	455.5990	0.000	-113.9892
21	1.5018343	0.014337	0.875035	133.0431	0.000	-84.84585
22	1.6926049	0.027108	0.839220	104.3937	0.000	-75.09141
23	2.0960408	0.052621	0.796849	82.37156	0.000	-63.45616
24	8.1354441	1.750699	0.342379	11.45394	0.003	17.69920

Some Indices by Malinowski (1991) for Nobs=145
****

N	Eval	RE	IE	IND	SLperc
1	8.1354441	0.0689709	0.0140786	1.30e-004	2.7273878
2	2.0960408	0.0656974	0.0189652	1.36e-004	22.172428
3	1.6926049	0.0629747	0.0222649	1.43e-004	25.112637
4	1.5018343	0.0603840	0.0246517	1.51e-004	25.947848
5	1.0252044	0.0588729	0.0268717	1.63e-004	33.722174
6	0.9429365	0.0574221	0.0287110	1.77e-004	34.533755
7	0.9012169	0.0559075	0.0301935	1.93e-004	34.364786
8	0.8159449	0.0544913	0.0314606	2.13e-004	35.524205
9	0.7902186	0.0529522	0.0324265	2.35e-004	34.974838

```
 10  0.7068670  0.0515364  0.0332666  2.63e-004  36.364668
 11  0.6393667  0.0502108  0.0339929  2.97e-004  37.550633
 12  0.5433084  0.0491831  0.0347777  3.42e-004  40.421468
 13  0.5330397  0.0480071  0.0353323  3.97e-004  39.851828
 14  0.5094265  0.0467314  0.0356917  4.67e-004  39.787197
 15  0.4774630  0.0453938  0.0358870  5.60e-004  40.109329
 16  0.3897347  0.0445219  0.0363520  6.96e-004  43.986122
 17  0.3819639  0.0434632  0.0365797  8.87e-004  43.645112
 18  0.3403650  0.0425754  0.0368714  0.0011827  45.623333
 19  0.3337787  0.0414103  0.0368451  0.0016564  45.385642
 20  0.3157519  0.0399890  0.0365048  0.0024993  45.899073
 21  0.2971980  0.0380650  0.0356066  0.0042294  46.480778
 22  0.2681326  0.0353387  0.0338342  0.0088347  48.226821
 23  0.1896638  0.0344908  0.0337646  0.0344908  59.506954
 24  0.1724946      .          .          .          .

The IND function is monoton increasing showing no minimum.
     The largest 1 Eigenvalues have SLperc <= alfa=10
```

2. Political Democracy and Industrialization Indicators:

```
  options NOECHO;
  bold = [
#include "..\\tdata\\sem71.dat"
         ];
  options ECHO;

  name = [ "Y1" : "Y8" "X1" : "X3" ];
  bold = shape(bold,.,11);
  bold = cname(bold,name);


  cov = bivar(bold,"cov");
  evl = dia2vec(eig(cov));
  eval = evl[>];
  print "eigval=", eval;


  optn = [ 1, .10, ., 75 ];
  < nval,infl,prob,malin > = screetst(eval,optn);

               Diag of Hat Matrix and Cook's Distance
               *************************************

   Case   DiagHat    StudRes   tProb    DFFITS     CooksD   FProb
```

3	0.833333	.	.	.	2.500000*	0.5918
4	0.700000	-0.288134	0.4107	-0.440132	0.178866	0.1517
5	0.600000	-0.068641	0.4758	-0.084068	5.3e-003	0.0053
6	0.523810	-0.224359	0.4184	-0.235310	0.036305	0.0353
7	0.464286	2.577001*	0.0308	2.399057*	1.352202*	0.6607
8	0.416667	0.875854	0.2106	0.740232	0.285035	0.2384
9	0.377778	-3.717398*	0.0049	-2.896573*	1.481680*	0.7093
10	0.345455	-6.148440*	0.0002	-4.466738*	1.781274*	0.7708
11	0.318182	-46.45095*	0.0000	-31.73204*	2.092243*	0.8206

Recommended Dimension Based on Studendized Residual : 3
Recommended Dimension Based on Cooks D : 3
Threshold for Cooks Distance: 0.25

Fit Linear Regression Through N Points (Cases)
****

N	Eval	MSE	Rsquared	Fval	Prob	BIC
1	0.0772991	0.000000	1.000000	.	.	.
2	0.3022756	0.000000	1.000000	.	.	.
3	1.2638813	0.090437	0.886160	7.784294	0.219	-4.505140
4	1.8926012	0.048973	0.954460	41.91750	0.023	-6.838563
5	2.5099100	0.032725	0.976984	127.3450	0.001	-11.87438
6	3.2046737	0.024956	0.986837	299.8918	0.000	-17.07668
7	3.2744194	0.053111	0.973855	186.2435	0.000	-15.62295
8	3.9024696	0.051049	0.978222	269.5063	0.000	-18.99006
9	5.6937560	0.144535	0.960852	171.8064	0.000	-12.69032
10	8.5637803	0.809458	0.890977	65.37900	0.000	2.529660
11	58.086034	194.7818	0.376399	5.432307	0.045	62.57342

Some Indices by Malinowski (1991) for Nobs=75
****

N	Eval	RE	IE	IND	SLperc
1	58.086034	0.2022707	0.0609869	0.0020227	1.3496420
2	8.5637803	0.1810311	0.0771919	0.0022350	23.028669
3	5.6937560	0.1654667	0.0864121	0.0025854	28.325910
4	3.9024696	0.1544579	0.0931416	0.0031522	34.067000
5	3.2744194	0.1433771	0.0966648	0.0039827	35.105829
6	3.2046737	0.1269747	0.0937770	0.0050790	30.906328
7	2.5099100	0.1085673	0.0866067	0.0067855	30.549397
8	1.8926012	0.0854649	0.0728847	0.0094961	28.504009
9	1.2638813	0.0503041	0.0455017	0.0125760	21.258504
10	0.3022756	0.0321038	0.0306098	0.0321038	39.753188
11	0.0772991	.	.	.	.

```
    The IND function is monoton increasing showing no minimum.
         The largest 1 Eigenvalues have SLperc <= alfa=10
```

3. Continuous EFA of LOT Data (A. Maydeu):

```
  options NOECHO;
  lot8 = [
#include "..\\tdata\\lot.dat"
           ];
  options ECHO;
  lot = shape(lot8,.,8);
  cnam = [ "i1" "i4" "i5" "i11" "i3" "i8" "i9" "i12" ];
  lot = cname(lot,cnam);


  cov = bivar(lot,"cov");
  evl = dia2vec(eig(cov));
  eval = evl[>];
  print "eigval=", eval;


  optn = [ 1, .10, ., 389 ];
  < nval,infl,prob,malin > = screetst(eval,optn);


              Diag of Hat Matrix and Cook's Distance
              **************************************

   Case   DiagHat    StudRes   tProb    DFFITS     CooksD   FProb

      3  0.833333   0.000000  0.5000  0.000000   2.500000* 0.5918
      4  0.700000   2.211987  0.1351  3.378866*  1.937376* 0.6596
      5  0.600000   2.434580* 0.0677  2.981740*  1.682333* 0.6764
      6  0.523810  -1.727881* 0.0912 -1.812217*  1.097349* 0.5831
      7  0.464286  -7.131381* 0.0010 -6.638954*  2.008679* 0.7711
      8  0.416667  -10.21832* 0.0001 -8.636058*  2.044933* 0.7897

      Recommended Dimension Based on Studendized Residual : 4
             Recommended Dimension Based on Cooks D : 8
                 Threshold for Cooks Distance: 0.25


          Fit Linear Regression Through N Points (Cases)
          **********************************************

      N      Eval       MSE  Rsquared      Fval   Prob       BIC
```

```
1 0.2948953  0.000000  1.000000      .        .         .
2 0.4305853  0.000000  1.000000      .        .         .
3 0.5282622  2.4e-004  0.991233  113.0661  0.060 -22.29005
4 0.5886083  7.1e-004  0.971226  67.50837  0.014 -23.77582
5 0.6027499  1.9e-003  0.914109  31.92783  0.011 -26.17196
6 0.8295649  2.8e-003  0.930778  53.78499  0.002 -30.18918
7 1.3868988  0.030783  0.799730  19.96635  0.007 -19.44076
8 3.6059333  0.561356  0.595743  8.842046  0.025  0.190459
```

```
     Some Indices by Malinowski (1991) for Nobs=389
     **********************************************

  N       Eval         RE          IE         IND      SLperc
  1  3.6059333  0.0413754  0.0146284  8.44e-004  14.520871
  2  1.3868988  0.0374570  0.0187285  0.0010405  30.422966
  3  0.8295649  0.0354559  0.0217122  0.0014182  40.064258
  4  0.6027499  0.0344097  0.0243314  0.0021506  46.501512
  5  0.5886083  0.0327770  0.0259125  0.0036419  46.384963
  6  0.5282622  0.0305368  0.0264456  0.0076342  48.403123
  7  0.4305853  0.0275334  0.0257551  0.0275334  55.027933
  8  0.2948953      .          .          .          .

The IND function is monoton increasing showing no minimum.
       The first eigenvalue has SLperc > alfa=10 %
```

4. Malinowski (1991, p.99): Table 4.4: nobs= 200, nval= 9:

```
eval = [ 19.193396   0.368079   0.009065
           0.004414   0.000294   0.000260
           0.000141   0.000132   0.000099 ];

optn = [ 1, .10, ., 200. ];
< nval,infl,prob,malin > = screetst(eval,optn);
```

```
          Diag of Hat Matrix and Cook's Distance
          **************************************

Case   DiagHat    StudRes   tProb    DFFITS     CooksD   FProb

   3  0.833333   0.000000  0.5000  0.000000   2.500000* 0.5918
   4  0.700000  -5.254760* 0.0599 -8.026778*  2.251784* 0.6925
   5  0.600000  -0.221858  0.4225 -0.271719   0.054043  0.0517
   6  0.523810  -91.77304* 0.0000 -96.25238*  2.199217* 0.7732
   7  0.464286  -3.101698* 0.0181 -2.887523*  1.530372* 0.6970
```

```
8  0.416667  -118.8170* 0.0000 -100.4187*  2.142098* 0.8014
9  0.377778  -133.1812* 0.0000 -103.7739*  2.124281* 0.8099

Recommended Dimension Based on Studendized Residual : 4
      Recommended Dimension Based on Cooks D : 4
          Threshold for Cooks Distance: 0.25


     Fit Linear Regression Through N Points (Cases)
     **********************************************

N      Eval       MSE  Rsquared      Fval   Prob       BIC
1 9.90e-005  0.000000  1.000000        .       .         .
2 1.32e-004  0.000000  1.000000        .       .         .
3 1.41e-004  9.6e-011  0.901840  9.187500  0.203 -66.49586
4 2.60e-004  1.4e-009  0.815030  8.812582  0.097 -76.39649
5 2.94e-004  9.4e-010  0.905069  28.60190  0.013 -98.71200
6 0.0044140  2.0e-006  0.470662  3.556604  0.132 -73.73937
7 0.0090650  5.4e-006  0.627305  8.415781  0.034 -80.00133
8 0.3680790  0.012670  0.351908  3.257946  0.121 -30.13855
9 19.193396  32.11499  0.310293  3.149235  0.119  35.94167


     Some Indices by Malinowski (1991) for Nobs=200
     **********************************************

 N       Eval         RE         IE        IND     SLperc
 1  19.193396  0.0154613  0.0051538  2.42e-004  6.40e-005
 2  0.3680790  0.0032077  0.0015121  6.55e-005  0.0032451
 3  0.0090650  0.0021095  0.0012179  5.86e-005  6.6214713
 4  0.0044140  9.62e-004  6.42e-004  3.85e-005  1.8603882
 5  2.94e-004  8.89e-004  6.62e-004  5.56e-005  39.158532
 6  2.60e-004  7.87e-004  6.43e-004  8.75e-005  38.300819
 7  1.41e-004  7.60e-004  6.70e-004  1.90e-004  51.771630
 8  1.32e-004  7.04e-004  6.63e-004  7.04e-004  56.490408
 9  9.90e-005      .          .          .          .

   The IND function shows a minimum at 4 eigenvalues.
    The largest 4 Eigenvalues have SLperc <= alfa=10
```

5. Malinowski (1991, p.216): Table 7.1: nobs= 38; nval= 9:

```
eval = [ 1627.301311  2.642417  0.140080
            0.000091  0.000057  0.000035
            0.000027  0.000022  0.000015 ];
```

```
optn = [ 1, .10, ., 38. ];
< nval,infl,prob,malin > = screetst(eval,optn);
```

```
            Diag of Hat Matrix and Cook's Distance
            ************************************

 Case   DiagHat    StudRes   tProb    DFFITS     CooksD   FProb

    3  0.833333   0.000000  0.5000  0.000000   2.500000* 0.5918
    4  0.700000  -1.118034  0.2323 -1.707825*  1.296296* 0.5645
    5  0.600000  -11.68475* 0.0036 -14.31084*  2.217517* 0.7437
    6  0.523810  -3.599827* 0.0184 -3.775530*  1.786434* 0.7210
    7  0.464286  -8716.569* 0.0000 -8114.684*  2.166667* 0.7899
    8  0.416667  -42.70850* 0.0000 -36.09527*  2.136999* 0.8008
    9  0.377778  -1603.109* 0.0000 -1249.132*  2.124995* 0.8100

    Recommended Dimension Based on Studendized Residual : 5
           Recommended Dimension Based on Cooks D : 9
               Threshold for Cooks Distance: 0.25

        Fit Linear Regression Through N Points (Cases)
        ********************************************

    N      Eval       MSE  Rsquared      Fval   Prob       BIC
    1 1.50e-005  0.000000  1.000000      .        .         .
    2 2.20e-005  0.000000  1.000000      .        .         .
    3 2.70e-005  6.7e-013  0.990826  108.0000  0.061 -81.40530
    4 3.50e-005  7.5e-013  0.992949  281.6667  0.004 -106.4474
    5 5.70e-005  3.5e-011  0.900555  27.16747  0.014 -115.2074
    6 9.10e-005  1.4e-010  0.862676  25.12827  0.007 -131.1477
    7 0.1400800  2.1e-003  0.375440  3.005637  0.144 -38.23708
    8 2.6424170  0.640060  0.363550  3.427293  0.114  1.240106
    9 1627.3013  234990.1  0.300885  3.012657  0.126  116.0234

        Some Indices by Malinowski (1991) for Nobs=38
        *******************************************

     N       Eval          RE         IE         IND     SLperc
     1  1627.3013  0.0956753  0.0318918  0.0014949  5.33e-009
     2  2.6424170  0.0229684  0.0108274  4.69e-004  0.0108670
     3  0.1400800  0.0010408  6.01e-004  2.89e-005  1.71e-006
     4  9.10e-005  9.06e-004  6.04e-004  3.62e-005  29.595725
     5  5.70e-005  8.07e-004  6.02e-004  5.04e-005  35.663805
     6  3.50e-005  7.49e-004  6.12e-004  8.33e-005  44.247296
```

```
  7  2.70e-005  6.98e-004  6.15e-004  1.74e-004  49.100703
  8  2.20e-005  6.28e-004  5.92e-004  6.28e-004  55.431583
  9  1.50e-005      .          .          .          .

    The IND function shows a minimum at 3 eigenvalues.
     The largest 3 Eigenvalues have SLperc <= alfa=10
```

6. Boston Housing Data: nobs=506, nvar=14:

```
fid = fopen("..\\tdata\\housing.dat","r");
form = "%g %g %g %g %g %g %g %g %g %g %g %g %g %g";
hous = fscanf(fid,form,506,14);
vnam = [ "crim" "zn" "indus" "chas" "nox" "rm" "age"
         "dis" "rad" "tax" "ptrat" "b" "lstat" "medv" ];
hous = cname(hous,vnam);
/* print hous; */
nobs = nrow(hous); nvar = ncol(hous);


cov = bivar(hous,"cov");
evl = dia2vec(eig(cov));
eval = evl[>];
print "eigval=", eval;

optn = [ 1, .10, ., 506 ];
< nval,infl,prob > = screetst(eval,optn);
```

```
             Diag of Hat Matrix and Cook's Distance
             **************************************

 Case   DiagHat    StudRes   tProb    DFFITS     CooksD   FProb

    3  0.833333   0.000000* 0.5000  0.000000   2.500000* 0.5918
    4  0.700000  -10.37372* 0.0306 -15.84612*  2.311851* 0.6981
    5  0.600000  -3.095878* 0.0452 -3.791660*  1.861549* 0.7019
    6  0.523810  -6.987748* 0.0030 -7.328812*  2.072657* 0.7588
    7  0.464286  -2.026888* 0.0563 -1.886930*  1.097800* 0.5975
    8  0.416667  -1.255070  0.1325 -1.060727*  0.513357* 0.3774
    9  0.377778  -8.053828* 0.0001 -6.275493*  1.945078* 0.7871
   10  0.345455  -3.870210* 0.0031 -2.811642*  1.438738* 0.7074
   11  0.318182  -11.41535* 0.0000 -7.798167*  1.978534* 0.8060
   12  0.294872  -9.530954* 0.0000 -6.163380*  1.902424* 0.8005
   13  0.274725  -26.26741* 0.0000 -16.16648*  2.053570* 0.8254
   14  0.257143  -16.28878* 0.0000 -9.583480*  1.994244* 0.8213
```

Recommended Dimension Based on Studendized Residual : 6
Recommended Dimension Based on Cooks D : 14
Threshold for Cooks Distance: 0.25

Fit Linear Regression Through N Points (Cases)
****

N	Eval	MSE	Rsquared	Fval	Prob	BIC
1	0.0029253	0.000000	1.000000	.	.	.
2	0.0591054	0.000000	1.000000	.	.	.
3	0.2179345	1.8e-003	0.929389	13.16214	0.171	-16.32976
4	1.1020242	0.095370	0.757937	6.262323	0.129	-4.172537
5	2.7212114	0.368272	0.791670	11.40023	0.043	0.228976
6	8.9096155	4.771751	0.680977	8.538262	0.043	14.44349
7	13.558445	7.738133	0.773767	17.10112	0.009	19.24782
8	17.062902	8.479961	0.843463	32.32956	0.001	21.91130
9	46.683442	85.84647	0.672700	14.38709	0.007	44.79081
10	77.043124	235.8473	0.678958	16.91888	0.003	59.27541
11	267.29650	3624.454	0.481812	8.368207	0.018	94.73279
12	822.46401	36186.22	0.420906	7.268352	0.022	130.4893
13	6250.8141	2302684	0.286355	4.413826	0.060	194.9341
14	30910.013	53023988	0.284936	4.781724	0.049	253.4606

Some Indices by Malinowski (1991) for Nobs=506
****

N	Eval	RE	IE	IND	SLperc
1	30910.013	1.0683494	0.2855284	0.0063216	0.0187516
2	6250.8141	0.4550118	0.1719783	0.0031598	0.0150853
3	822.46401	0.2794485	0.1293595	0.0023095	0.8274317
4	267.29650	0.1818660	0.0972115	0.0018187	1.8328783
5	77.043124	0.1408282	0.0841610	0.0017386	8.2691511
6	46.683442	0.1038229	0.0679680	0.0016222	7.3154097
7	17.062902	0.0866127	0.0612445	0.0017676	17.864713
8	13.558445	0.0654689	0.0494898	0.0018186	12.831515
9	8.9096155	0.0402718	0.0322893	0.0016109	6.7678230
10	2.7212114	0.0261305	0.0220843	0.0016332	11.873778
11	1.1020242	0.0135805	0.0120378	0.0015089	9.3679728
12	0.2179345	0.0078291	0.0072484	0.0019573	20.211583
13	0.0591054	0.0024044	0.0023170	0.0024044	19.423598
14	0.0029253	.	.	.	.

The IND function shows a minimum at 11 eigenvalues.
The largest 6 Eigenvalues have SLperc <= alfa=10

7. Diabetes Data set from LARS Package in R:

```
 options NOECHO;
 DiabX = [
 %inc "..\\tdata\\diabX.dat";
 ];
 DiabX = shape(DiabX,.,10);
 options ECHO;
 print "dimension=", nrow(DiabX),ncol(DiabX);


cov = bivar(DiabX,"cov");
evl = dia2vec(eig(cov));
eval = evl[>];
print "eigval=", eval;

optn = [ 1, .10, ., 442 ];
< nval,infl,prob > = screetst(eval,optn);
```

```
             Diag of Hat Matrix and Cook's Distance
             **************************************

 Case   DiagHat    StudRes   tProb    DFFITS     CooksD   FProb

    3  0.833333       .        .         .       2.500000* 0.5918
    4  0.700000   0.291641  0.4097  0.445488   0.182903  0.1546
    5  0.600000   1.078151  0.1969  1.320460*  0.827034* 0.4825
    6  0.523810   1.281519  0.1450  1.344069*  0.778288* 0.4818
    7  0.464286  -0.570581  0.2994 -0.531182   0.163074  0.1461
    8  0.416667  -1.218519  0.1387 -1.029836*  0.490639* 0.3652
    9  0.377778  -1.783153* 0.0624 -1.389422*  0.736056* 0.4873
   10  0.345455  -19.12355* 0.0000 -13.89294*  2.071462* 0.8116

    Recommended Dimension Based on Studendized Residual : 2
           Recommended Dimension Based on Cooks D : 3
               Threshold for Cooks Distance: 0.25


         Fit Linear Regression Through N Points (Cases)
         **********************************************

    N      Eval       MSE  Rsquared      Fval   Prob       BIC
    1 1.94e-005  0.000000  1.000000      .       .         .
    2 1.78e-004  0.000000  1.000000      .       .         .
    3 9.83e-004  7.0e-008  0.869229  6.646935  0.236 -46.72428
    4 0.0012167  3.8e-008  0.927265  25.49704  0.037 -63.12325
```

```
 5 0.0013667  4.0e-008  0.920783  34.87054  0.010 -79.95107
 6 0.0015016  4.6e-008  0.906326  38.70137  0.003 -96.24897
 7 0.0021666  4.0e-008  0.941068  79.84315  0.000 -114.2918
 8 0.0027346  4.3e-008  0.955538  128.9471  0.000 -130.8161
 9 0.0033839  5.7e-008  0.959461  165.6735  0.000 -145.4216
10 0.0091252  2.7e-006  0.658426  15.42098  0.004 -123.7643

      Some Indices by Malinowski (1991) for Nobs=442
      **********************************************

 N       Eval         RE          IE         IND     SLperc
 1  0.0091252  0.0018456  5.84e-004  2.28e-005  11.702805
 2  0.0033839  0.0016956  7.58e-004  2.65e-005  28.355563
 3  0.0027346  0.0015499  8.49e-004  3.16e-005  29.533010
 4  0.0021666  0.0014091  8.91e-004  3.91e-005  31.041879
 5  0.0015016  0.0013050  9.23e-004  5.22e-005  36.497124
 6  0.0013667  0.0011644  9.02e-004  7.28e-005  34.673199
 7  0.0012167  9.44e-004  7.89e-004  1.05e-004  30.279063
 8  9.83e-004  4.72e-004  4.22e-004  1.18e-004  15.541715
 9  1.78e-004  2.10e-004  1.99e-004  2.10e-004  27.870959
10  1.94e-005      .           .           .          .

The IND function is monoton increasing showing no minimum.
       The first eigenvalue has SLperc > alfa=10 %
```

8. Myeloma Data: nobs=105, nvar=7009:

```
myelo = fil2obj("..\\save\\myelo");
print "nc=", nc = ncol(myelo),
      "nr=", nr = nrow(myelo);
/* print myelo[,7009]; */
z = myelo[,1:7008]; /* attrib(z); */
free myelo;
x = z`;
attrib(x);
```

```
--------------------------------------
         Table of Attributes
--------------------------------------
Object Name      name                x
Object Type      otyp       matrix_gen
Data Type        dtyp             real
Storage Type     styp        dens_full
Row Names        rnam                0
```

```
Column Names     cnam              0
Row Labels       rlab              0
Column Labels    clab              0
Number Rows      nrow           7008
Number Columns   ncol            105
Lower Bandwidth  lbw             104
Upper Bandwidth  ubw            7007
Size in Bytes    size        5886976
String Length    slen              0
Number Strings   nstr              0
Number MissVals  nmis              0
Number NonzeroV  nzer         735527
Smallest Value   vmin              0
Largest Value    vmax      1.676e+005
Frobenius Norm   nrm2              .
Recip Condition  rcond             .
Determinant      det               .
Largest SingVal  svb               .
Smallest SingV   svs               .
Num Rank Estim   rnk               .
---------------------------------------
```

```
cov = bivar(x,"scp");
evl = dia2vec(eig(cov));
eval = evl[>];
print "eigval=", eval;

optn = [ 1, .10, ., 7008 ];
< nval,infl,prob > = screetst(eval,optn);
```

From the 105 eigenvalues is one zero eigenvalue. The implemented algorithms only use the largest 104 eigenvalues:

```
          Diag of Hat Matrix and Cook's Distance
     (starting with two smallest eigenvalues adding next)
     ****************************************************

Case   DiagHat    StudRes   tProb    DFFITS     CooksD   FProb

   4  0.833333       .        .        .       2.500000* 0.5918
   5  0.700000   1.689075  0.1702  2.580104*  1.727740* 0.6334
   6  0.600000   0.308973  0.3933  0.378413   0.102505  0.0944
   7  0.523810   0.417163  0.3523  0.437525   0.120621  0.1105
   8  0.464286   1.269602  0.1365  1.181935*  0.622326* 0.4263
   9  0.416667   2.254523* 0.0369  1.905420*  1.080235* 0.6025
```

10	0.377778	1.766736*	0.0638	1.376629*	0.727181*	0.4835
11	0.345455	0.193191	0.4261	0.140350	0.011196	0.0111
12	0.318182	-0.680883	0.2576	-0.465132	0.115030	0.1074
13	0.294872	0.219974	0.4154	0.142251	0.011182	0.0111
14	0.274725	-0.680574	0.2558	-0.418864	0.092224	0.0874
15	0.257143	-0.589157	0.2838	-0.346629	0.063533	0.0612
....	....	....	....	....	....	....
100	0.039798	-5.819093*	0.0000	-1.184689*	0.524166*	0.4063
101	0.039406	-7.865195*	0.0000	-1.593017*	0.782742*	0.5400
102	0.039022	-6.380378*	0.0000	-1.285706*	0.589907*	0.4437
103	0.038645	-14.05195*	0.0000	-2.817337*	1.338707*	0.7332
104	0.038275	-16.03582*	0.0000	-3.199060*	1.447066*	0.7599
105	0.037912	-447.2841*	0.0000	-88.79019*	2.008695*	0.8606

Recommended Dimension Based on Studendized Residual : 67
Recommended Dimension Based on Cooks D : 25
Threshold for Cooks Distance: 0.25

Fit Linear Regression Through N Points (Cases)
****

N	Eval	MSE	Rsquared	Fval	Prob	BIC
1	-7.03e-007	.	.	.	.	.
2	1.76e+009	0.000000	1.000000	.	.	.
3	2.04e+009	0.000000	1.000000	.	.	.
4	2.20e+009	2.7e+015	0.973953	37.39231	0.103	109.2613
5	2.29e+009	5.1e+015	0.937463	29.98094	0.032	149.9264
6	2.48e+009	3.6e+015	0.963473	79.13130	0.003	184.3032
7	2.62e+009	2.8e+015	0.976186	163.9653	0.000	218.5749
8	2.71e+009	3.2e+015	0.976328	206.2179	0.000	254.8252
9	2.75e+009	5.4e+015	0.962077	152.2138	0.000	294.5628
10	2.82e+009	7.0e+015	0.952933	141.7253	0.000	333.0724
11	3.03e+009	6.2e+015	0.964569	217.7911	0.000	368.2000
12	3.23e+009	5.8e+015	0.972908	323.2032	0.000	403.8188
13	3.30e+009	5.2e+015	0.978674	458.9097	0.000	438.8624
14	3.50e+009	5.0e+015	0.982899	632.2260	0.000	474.3635
15	3.64e+009	4.7e+015	0.986133	853.3612	0.000	509.6880
....	....	....	....	....	....	....
100	1.19e+011	2.0e+020	0.621955	159.5831	0.000	4630.305
101	1.63e+011	3.2e+020	0.559355	124.4010	0.000	4725.341
102	1.71e+011	4.5e+020	0.527013	110.3081	0.000	4806.616
103	3.64e+011	1.3e+021	0.366388	57.82518	0.000	4965.012
104	6.69e+011	4.7e+021	0.235197	31.06020	0.000	5143.741
105	3.13e+013	9.2e+024	0.032780	3.456843	0.066	5982.159

```
Some Indices by Malinowski (1991) for Nobs=7008
***********************************************

  N       Eval          RE          IE         IND      SLperc
  1  3.13e+013   2113.8321   207.27829   0.1992489   3.12e-039
  2  6.69e+011   1891.2395   262.26773   0.1817800   0.0425938
  3  3.64e+011   1760.2821   298.96904   0.1725598   0.4769946
  4  1.71e+011   1698.5299   333.10913   0.1698530   4.2603125
  5  1.63e+011   1637.0066   358.93743   0.1670244   4.0523532
  6  1.19e+011   1592.0381   382.39518   0.1657682   7.1493665
  7  9.90e+010   1554.0537   403.17945   0.1651667   9.1187361
  8  8.77e+010   1519.8312   421.52534   0.1649122   10.386409
  9  7.78e+010   1489.0664   438.04491   0.1649935   11.772624
 10  7.30e+010   1459.4965   452.57065   0.1651762   12.216173
 11  6.71e+010   1431.7902   465.64940   0.1655440   13.062176
 12  6.53e+010   1403.9207   476.88815   0.1658696   12.840243
 13  6.25e+010   1376.4712   486.65605   0.1662204   12.935420
 14  5.79e+010   1350.5079   495.50092   0.1667294   13.656869
 15  5.31e+010   1326.3378   503.71292   0.1674457   14.649487
..................................................................
100  2.48e+009   543.72220   533.16348   33.982637   48.276053
101  2.29e+009   534.21610   526.45468   59.357344   50.454763
102  2.20e+009   520.54223   515.51272   130.13556   52.594868
103  2.04e+009   500.52648   498.11429   500.52648   58.524890
104  1.76e+009           .           .           .           .
105 -7.03e-007           .           .           .           .

The IND function shows a minimum at 8 eigenvalues.
Largest 7 Eigenvalue(s) with SLperc <= alfa= 10 %
```

3.7 Function split

<gof,nodes,levmap,yptr,yptt> = split(trn,model,optn<,class<,tst>>)

Purpose: The `split()` function incorporates a simple algorithm for the tree regression (CART) model. At this time only binary classification is implemented, that means that the response variable y must be binary with values 0 and 1.

Input: **trn** the $N \times M$ input data set containing the $N \times m$ predictor matrix $\mathbf{X}$ and the N response vector $\mathbf{Y}$. Which data columns are selected for $\mathbf{X}$ and $\mathbf{Y}$ must be specified with the model string. Before computing the *garotte* estimates, the data $(\mathbf{X}, Y)$ will be mean centered.

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 : The option argument is specified in form of a simple vector.

Index	Meaning
[**1**]	amount of printed output
[**2**]	lower chi2 threshold for accepting splits; default=0.
[**3**]	number bins for interval variable; default=100
[**4**]	number of passes through data = number splits; default=12;
[**6**]	prob cutoff for accuracy table; default=.5
[**10**]	print observationwise stat: predicted y, residuals

class : This optional argument should be an integer scalar or vector of integer scalars naming the number of columns which are considered categorical (nominal scaled) variables.

test specifies a matrix of test data which is not used for modeling but for predicted values are computed (scored) that can be returned with the last output argument `yptt`.

Output: gof matrix of goodness-of-fit measures

nodes a matrix with information for each node of the tree

levmap mapping of the split levels

yptrn $N_{step} \times N$ matrix with predicted values for training data

yptst $N_{step} \times N(test)$ matrix with predicted values for $N(test)$ test data

Restrictions: 1. Missing values in the input data set are replaced by the variable mean.

2. The response must be $(0,1)$ binary.

Relationships: reg()

Examples: 1. Boston Housing Data: nobs=506, nvar=14

```
/*--- Housing: nobs=506, nvar=14, last var --------------------*/
/*--- Small (nobs=30) Boston Housing Data ---*/
fid = fopen("..\\tdata\\housing.dat","r");
form = "%g %g %g %g %g %g %g %g %g %g %g %g %g %g";
hous = fscanf(fid,form,506,14);
vnam = [ "crim" "zn" "indus" "chas" "nox" "rm" "age"
         "dis" "rad" "tax" "ptrat" "b" "lstat" "medv" ];

/*--- Binary target: make var 14 binary ---*/
hous31 = hous[1:30,];
```

```
sopt = [ "ari" "std" "med" ];
mom = univar(hous31,sopt);
cutof = mom[3,14];
y = hous31[,14]; y = y .< cutof; hous31[,14] = y;
hous31 = cname(hous31,vnam);
modl = "14 = 1:3 5:13";
clas = [ 9 14 ]; /* rad[9] is CLASS with 6 levels */

/* use CHI2: bins=1000 */
optn = cons(10,1,.);
optn[1]= 2; optn[3]= 1000; optn[10]= 1;
< gof,nodes,remap,yptr > = split(hous31,modl,optn,clas);
```

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

                 Number Valid Observations  506
                 Response Variable         Y[14]
                 N Independend Variables     12
                 Number Parameters           21
                 Number Passes               12
                 Number Bins               1000
                 Chisquare Threshold  0.0000000
                 Design Coding:  Rank-Deficient

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

 Class  Level    Value

 Y[ 1]      2       0       1
 C[ 9]      9       1       2       3       4       5
                    6       7       8      24

                         *************
                         Split History
                         *************

         Node Parnt         Split   ChiSquare   (Val,Lev)
            1     0         lstat  228.570825  11.6960003
        --- Pass=2 SSE=83 Accuracy=0.835968 Misc=83 ---
```

```
   2     1            rm  106.649779  6.06090093
   3     1           age  31.1180759  72.7149000
--- Pass=3 SSE=83 Accuracy=0.835968 Misc=83 ---

   4     2           nox  7.98573673  0.59738199
   5     2         lstat  30.3440461  8.61650625
   6     3           rad  12.5742577     3     6
   7     3         indus  29.0141598  4.06095998
--- Pass=4 SSE=78 Accuracy=0.84585 Misc=78 ---

   8     4         ptrat  6.18279075  16.8018002
   9     4           nox  6.00000000  0.71804753
  10     5         ptrat  13.6306461  19.7064001
  11     5            rm  9.30058407  6.25667604
  12     6         indus  4.28571429  4.93391997
  13     6           tax  7.47708174  277.128000
  15     7         lstat  16.8161895  14.3759484
--- Pass=5 SSE=65 Accuracy=0.871542 Misc=65 ---

  16     8         indus  6.00000000  3.65175998
  17     8           age  8.63007055  30.2822001
  21    10            rm  11.0000000  6.31105802
  22    11           rad  9.16363636     4     5
  23    11         indus  10.9803922  1.90584000
  25    12            rm  5.83333333  5.79473195
  26    13           nox  8.57142857  0.46421799
  27    13            rm  17.0000000  6.11309092
  28    15          crim  15.0000000  1.51880790
  29    15          crim  15.5689058  0.18425975
--- Pass=6 SSE=45 Accuracy=0.911067 Misc=45 ---

  32    17         indus  6.66666667  3.24255998
  33    17         ptrat  12.9807692  17.4047914
  37    22          crim  4.95000000  0.18425975
  39    23             b  16.4848485  70.9112389
  40    25            rm  4.00000000  5.70761641
  42    26           tax  7.00000000  243.059616
  47    28             b  4.74725275  393.727354
  48    29             b  7.87500000  396.106834
  49    29           rad  46.9953052     1     6
--- Pass=7 SSE=34 Accuracy=0.932806 Misc=34 ---

  51    32         ptrat  6.00000000  16.9005660
  53    33          crim  7.97333333  0.80704889
  55    37         lstat  5.00000000  9.81134996
  57    39          crim  33.0000000  4.72172344
```

```
  62    47         lstat  5.31818182  12.0309939
  64    48           tax  5.02857143  277.128000
  67    49            rm  22.4942789  6.83331287
 --- Pass=8 SSE=8 Accuracy=0.98419 Misc=8 ---

  71    53           dis  3.00000000  3.15302966
  77    62         lstat  4.27777778  14.0007557
  78    64          crim  5.00000000  0.15045120
  80    67           age  12.9934783  82.5102509
  81    67           age  3.00000000  99.3178725
 --- Pass=9 SSE=9 Accuracy=0.982213 Misc=9 ---

                                                  Page 31

 Node Parnt         Split   ChiSquare   (Val,Lev)
  85    77             b  4.00000000  350.845952
  88    80           age  10.0000000  82.0008927
--- Pass=10 SSE=1 Accuracy=0.998024 Misc=1 ---
    --- Pass=11 SSE=0 Accuracy=1 Misc=0 ---

                 **************
                 Effect Summary
                 **************

            Effect Type 1st Split  N Splits
             lstat    V         1         6
                rm    V         2         7
               age    V         3         5
               nox    V         4         3
               rad    C         6         3
             indus    V         7         5
             ptrat    V         8         4
               tax    V        13         3
              crim    V        28         6
                 b    V        39         4
               dis    V        71         1

  *****************************************
  Residuals of Linear Regression (Training)
  *****************************************

    Nobs        Yobs        Yprd    Residual
       1  0.00000000  0.00000000  0.00000000
       2  0.00000000  0.00000000  0.00000000
       3  0.00000000  0.00000000  0.00000000
```

```
  4  0.00000000  0.00000000  0.00000000
  5  0.00000000  0.00000000  0.00000000
  6  0.00000000  0.00000000  0.00000000
  7  0.00000000  0.00000000  0.00000000
  8  0.00000000  0.00000000  0.00000000
  9  1.00000000  1.00000000  0.00000000
 10  1.00000000  1.00000000  0.00000000
 11  1.00000000  1.00000000  0.00000000
 12  1.00000000  1.00000000  0.00000000
 13  0.00000000  0.00000000  0.00000000
 14  1.00000000  1.00000000  0.00000000
 15  1.00000000  1.00000000  0.00000000
 16  1.00000000  1.00000000  0.00000000
 17  0.00000000  0.00000000  0.00000000
 18  1.00000000  1.00000000  0.00000000
 19  1.00000000  1.00000000  0.00000000
 20  1.00000000  1.00000000  0.00000000


500  1.00000000  1.00000000  0.00000000
501  1.00000000  1.00000000  0.00000000
502  0.00000000  0.00000000  0.00000000
503  1.00000000  1.00000000  0.00000000
504  0.00000000  0.00000000  0.00000000
505  0.00000000  0.00000000  0.00000000
506  1.00000000  1.00000000  0.00000000

      Sum-of-Squared Residual 0


Training Data SSE=0 Accuracy=1 Misc=0
      Total Processing Time: 0
```

```
print "GOF=", gof;
print "Nodes=", nodes;
print "Remap=", remap;
print "Ypredict=", yptr;
```

```
GOF=
   |     Npasses      Nnodes   SSQ_error         N00         N01
----------------------------------------------------------------
 1 |      11.000      95.000       0.000     255.000       0.000

   |         N10         N11    ClassErr    Accuracy      unused
----------------------------------------------------------------
```

```
  1 |       0.000     251.000       0.000       1.000           .

    |      unused      unused      unused      unused      unused
-----------------------------------------------------------------
  1 |           .           .           .           .           .

    |      unused      unused      unused      unused      unused
-----------------------------------------------------------------
  1 |           .           .           .           .           .

    |      unused      unused      unused      unused      unused
-----------------------------------------------------------------
  1 |           .           .           .           .           .
```

Nodes=

	Pass	Node	Type	Index	Parent
1	1.000	0.000	0.000	11.000	-1.000
2	2.000	1.000	0.000	5.000	0.000
3	2.000	2.000	0.000	6.000	0.000
4	3.000	3.000	0.000	4.000	1.000
5	3.000	4.000	0.000	11.000	1.000
6	3.000	5.000	1.000	0.000	2.000
7	3.000	6.000	0.000	3.000	2.000
8	4.000	7.000	0.000	9.000	3.000
9	4.000	8.000	0.000	4.000	3.000
10	4.000	9.000	0.000	9.000	4.000
11	4.000	10.000	0.000	5.000	4.000
12	4.000	11.000	0.000	3.000	5.000
13	4.000	12.000	0.000	8.000	5.000
14	4.000	13.000	0.000	-1.000	6.000
15	4.000	14.000	0.000	11.000	6.000
16	5.000	15.000	0.000	3.000	7.000
17	5.000	16.000	0.000	6.000	7.000
18	5.000	17.000	0.000	-1.000	8.000
19	5.000	18.000	0.000	-1.000	8.000
20	5.000	19.000	0.000	-1.000	9.000
21	5.000	20.000	0.000	5.000	9.000
22	5.000	21.000	1.000	0.000	10.000
23	5.000	22.000	0.000	3.000	10.000
24	5.000	23.000	0.000	-1.000	11.000
25	5.000	24.000	0.000	5.000	11.000
26	5.000	25.000	0.000	4.000	12.000
27	5.000	26.000	0.000	5.000	12.000
28	5.000	27.000	0.000	1.000	14.000

	Pass	Node	Type	Index	Parent
29	5.000	28.000	0.000	1.000	14.000
30	6.000	29.000	0.000	-1.000	15.000
31	6.000	30.000	0.000	-1.000	15.000
32	6.000	31.000	0.000	3.000	16.000
33	6.000	32.000	0.000	9.000	16.000
34	6.000	33.000	0.000	-1.000	20.000
35	6.000	34.000	0.000	-1.000	20.000
36	6.000	35.000	0.000	-1.000	21.000
37	6.000	36.000	0.000	1.000	21.000
38	6.000	37.000	0.000	-1.000	22.000
39	6.000	38.000	0.000	10.000	22.000
40	6.000	39.000	0.000	5.000	24.000
41	6.000	40.000	0.000	-1.000	24.000
42	6.000	41.000	0.000	8.000	25.000
43	6.000	42.000	0.000	-1.000	25.000
44	6.000	43.000	0.000	-1.000	26.000
45	6.000	44.000	0.000	-1.000	26.000
46	6.000	45.000	0.000	-1.000	27.000
47	6.000	46.000	0.000	10.000	27.000
48	6.000	47.000	0.000	10.000	28.000
49	6.000	48.000	1.000	0.000	28.000
50	7.000	49.000	0.000	-1.000	31.000
51	7.000	50.000	0.000	9.000	31.000
52	7.000	51.000	0.000	-1.000	32.000
53	7.000	52.000	0.000	1.000	32.000
54	7.000	53.000	0.000	-1.000	36.000
55	7.000	54.000	0.000	11.000	36.000
56	7.000	55.000	0.000	-1.000	38.000
57	7.000	56.000	0.000	1.000	38.000
58	7.000	57.000	0.000	-1.000	39.000
59	7.000	58.000	0.000	-1.000	39.000
60	7.000	59.000	0.000	-1.000	41.000
61	7.000	60.000	0.000	-1.000	41.000
62	7.000	61.000	0.000	11.000	46.000
63	7.000	62.000	0.000	-1.000	46.000
64	7.000	63.000	0.000	8.000	47.000
65	7.000	64.000	0.000	-1.000	47.000
66	7.000	65.000	0.000	-1.000	48.000
67	7.000	66.000	0.000	5.000	48.000
68	8.000	67.000	0.000	-1.000	50.000
69	8.000	68.000	0.000	-1.000	50.000
70	8.000	69.000	0.000	-1.000	52.000

71	8.000	70.000	0.000	7.000	52.000
72	8.000	71.000	0.000	-1.000	54.000
73	8.000	72.000	0.000	-1.000	54.000
74	8.000	73.000	0.000	-1.000	56.000
75	8.000	74.000	0.000	-1.000	56.000
76	8.000	75.000	0.000	-1.000	61.000
77	8.000	76.000	0.000	11.000	61.000
78	8.000	77.000	0.000	1.000	63.000
79	8.000	78.000	0.000	-1.000	63.000
80	8.000	79.000	0.000	6.000	66.000
81	8.000	80.000	0.000	6.000	66.000
82	9.000	81.000	0.000	-1.000	70.000
83	9.000	82.000	0.000	-1.000	70.000
84	9.000	83.000	0.000	-1.000	76.000
85	9.000	84.000	0.000	10.000	76.000
86	9.000	85.000	0.000	-1.000	77.000
87	9.000	86.000	0.000	-1.000	77.000
88	9.000	87.000	0.000	6.000	79.000
89	9.000	88.000	0.000	-1.000	79.000
90	9.000	89.000	0.000	-1.000	80.000
91	9.000	90.000	0.000	-1.000	80.000
92	10.000	91.000	0.000	-1.000	84.000
93	10.000	92.000	0.000	-1.000	84.000
94	10.000	93.000	0.000	-1.000	87.000
95	10.000	94.000	0.000	-1.000	87.000

	Effect	Value	Ymean	Chisqu	Level1
1	11.000	11.696	0.496	228.571	.
2	5.000	6.061	0.172	106.650	.
3	6.000	72.715	0.844	31.118	.
4	4.000	0.597	0.673	7.986	.
5	11.000	8.617	0.056	30.344	.
6	0.000	.	0.568	12.574	3.000
7	3.000	4.061	0.905	29.014	.
8	9.000	16.802	0.744	6.183	.
9	4.000	0.718	0.167	6.000	.
10	9.000	19.706	0.006	13.631	.
11	5.000	6.257	0.208	9.301	.
12	3.000	4.934	0.200	4.286	.
13	8.000	277.128	0.759	7.477	.
14	-1.000	.	0.000	-1.000	0.000
15	11.000	14.376	0.919	16.816	.
16	3.000	3.652	0.333	6.000	.
17	6.000	30.282	0.811	8.630	.

18	-1.000	.	0.000	-1.000	0.000
19	-1.000	.	1.000	-1.000	0.000
20	-1.000	.	0.000	-1.000	0.000
21	5.000	6.311	0.091	11.000	.
22	0.000	.	0.444	9.164	4.000
23	3.000	1.906	0.086	10.980	.
24	-1.000	.	1.000	-1.000	0.000
25	5.000	5.795	0.143	5.833	.
26	4.000	0.464	0.500	8.571	.
27	5.000	6.113	0.941	17.000	.
28	1.000	1.519	0.750	15.000	.
29	1.000	0.184	0.957	15.569	.
30	-1.000	.	1.000	-1.000	0.000
31	-1.000	.	0.000	-1.000	0.000
32	3.000	3.243	0.500	6.667	.
33	9.000	17.405	0.926	12.981	.
34	-1.000	.	1.000	-1.000	0.000
35	-1.000	.	0.000	-1.000	0.000
36	-1.000	.	0.000	-1.000	0.000
37	1.000	0.184	0.727	4.950	.
38	-1.000	.	1.000	-1.000	0.000
39	10.000	70.911	0.059	16.485	.
40	5.000	5.708	0.500	4.000	.
41	-1.000	.	0.000	-1.000	0.000
42	8.000	243.060	0.857	7.000	.
43	-1.000	.	0.000	-1.000	0.000
44	-1.000	.	1.000	-1.000	0.000
45	-1.000	.	0.000	-1.000	0.000
46	-1.000	.	1.000	-1.000	0.000
47	10.000	393.727	0.438	4.747	.
48	10.000	396.107	0.778	7.875	.
49	0.000	.	0.979	46.995	1.000
50	-1.000	.	1.000	-1.000	0.000
51	9.000	16.901	0.167	6.000	.
52	-1.000	.	0.000	-1.000	0.000
53	1.000	0.807	0.962	7.973	.
54	-1.000	.	1.000	-1.000	0.000
55	11.000	9.811	0.400	5.000	.
56	-1.000	.	1.000	-1.000	0.000

	Effect	Value	Ymean	Chisqu	Level1
57	1.000	4.722	0.030	33.000	.
58	-1.000	.	0.000	-1.000	0.000
59	-1.000	.	1.000	-1.000	0.000

60	-1.000	.	1.000	-1.000	0.000
61	-1.000	.	0.000	-1.000	0.000
62	11.000	12.031	0.308	5.318	.
63	-1.000	.	1.000	-1.000	0.000
64	8.000	277.128	0.875	5.029	.
65	-1.000	.	0.000	-1.000	0.000
66	-1.000	.	0.000	-1.000	0.000
67	5.000	6.833	0.986	22.494	.
68	-1.000	.	1.000	-1.000	0.000
69	-1.000	.	0.000	-1.000	0.000
70	-1.000	.	1.000	-1.000	0.000
71	7.000	3.153	0.667	3.000	.
72	-1.000	.	0.000	-1.000	0.000
73	-1.000	.	1.000	-1.000	0.000
74	-1.000	.	0.000	-1.000	0.000
75	-1.000	.	1.000	-1.000	0.000
76	-1.000	.	1.000	-1.000	0.000
77	11.000	14.001	0.182	4.278	.
78	1.000	0.150	0.600	5.000	.
79	-1.000	.	1.000	-1.000	0.000
80	6.000	82.510	0.993	12.993	.
81	6.000	99.318	0.667	3.000	.
82	-1.000	.	1.000	-1.000	0.000
83	-1.000	.	0.000	-1.000	0.000
84	-1.000	.	0.000	-1.000	0.000
85	10.000	350.846	0.500	4.000	.
86	-1.000	.	1.000	-1.000	0.000
87	-1.000	.	0.000	-1.000	0.000
88	6.000	82.001	0.900	10.000	.
89	-1.000	.	1.000	-1.000	0.000
90	-1.000	.	1.000	-1.000	0.000
91	-1.000	.	0.000	-1.000	0.000
92	-1.000	.	1.000	-1.000	0.000
93	-1.000	.	0.000	-1.000	0.000
94	-1.000	.	1.000	-1.000	0.000
95	-1.000	.	0.000	-1.000	0.000

	Level2	NodLft	NodRgt	Remap	unused
1	.	1.000	2.000	0.000	0.000
2	.	3.000	4.000	0.000	0.000
3	.	5.000	6.000	0.000	0.000
4	.	7.000	8.000	0.000	0.000
5	.	9.000	10.000	0.000	0.000

6	6.000	11.000	12.000	0.000	0.000
7	.	13.000	14.000	0.000	0.000
8	.	15.000	16.000	0.000	0.000
9	.	17.000	18.000	0.000	0.000
10	.	19.000	20.000	0.000	0.000
11	.	21.000	22.000	0.000	0.000
12	.	23.000	24.000	0.000	0.000
13	.	25.000	26.000	0.000	0.000
14	0.000	0.000	0.000	0.000	0.000
15	.	27.000	28.000	0.000	0.000
16	.	29.000	30.000	0.000	0.000
17	.	31.000	32.000	0.000	0.000
18	0.000	0.000	0.000	0.000	0.000
19	0.000	0.000	0.000	0.000	0.000
20	0.000	0.000	0.000	0.000	0.000
21	.	33.000	34.000	0.000	0.000
22	5.000	35.000	36.000	9.000	0.000
23	.	37.000	38.000	0.000	0.000
24	0.000	0.000	0.000	0.000	0.000
25	.	39.000	40.000	0.000	0.000
26	.	41.000	42.000	0.000	0.000
27	.	43.000	44.000	0.000	0.000
28	.	45.000	46.000	0.000	0.000
29	.	47.000	48.000	0.000	0.000
30	0.000	0.000	0.000	0.000	0.000
31	0.000	0.000	0.000	0.000	0.000
32	.	49.000	50.000	0.000	0.000
33	.	51.000	52.000	0.000	0.000
34	0.000	0.000	0.000	0.000	0.000
35	0.000	0.000	0.000	0.000	0.000
36	0.000	0.000	0.000	0.000	0.000
37	.	53.000	54.000	0.000	0.000
38	0.000	0.000	0.000	0.000	0.000
39	.	55.000	56.000	0.000	0.000
40	.	57.000	58.000	0.000	0.000
41	0.000	0.000	0.000	0.000	0.000
42	.	59.000	60.000	0.000	0.000
43	0.000	0.000	0.000	0.000	0.000
44	0.000	0.000	0.000	0.000	0.000
45	0.000	0.000	0.000	0.000	0.000
46	0.000	0.000	0.000	0.000	0.000
47	.	61.000	62.000	0.000	0.000
48	.	63.000	64.000	0.000	0.000
49	6.000	65.000	66.000	18.000	0.000
50	0.000	0.000	0.000	0.000	0.000
51	.	67.000	68.000	0.000	0.000

	Level2	NodLft	NodRgt	Remap	unused
52	0.000	0.000	0.000	0.000	0.000
53	.	69.000	70.000	0.000	0.000
54	0.000	0.000	0.000	0.000	0.000
55	.	71.000	72.000	0.000	0.000
56	0.000	0.000	0.000	0.000	0.000

	Level2	NodLft	NodRgt	Remap	unused
57	.	73.000	74.000	0.000	0.000
58	0.000	0.000	0.000	0.000	0.000
59	0.000	0.000	0.000	0.000	0.000
60	0.000	0.000	0.000	0.000	0.000
61	0.000	0.000	0.000	0.000	0.000
62	.	75.000	76.000	0.000	0.000
63	0.000	0.000	0.000	0.000	0.000
64	.	77.000	78.000	0.000	0.000
65	0.000	0.000	0.000	0.000	0.000
66	0.000	0.000	0.000	0.000	0.000
67	.	79.000	80.000	0.000	0.000
68	0.000	0.000	0.000	0.000	0.000
69	0.000	0.000	0.000	0.000	0.000
70	0.000	0.000	0.000	0.000	0.000
71	.	81.000	82.000	0.000	0.000
72	0.000	0.000	0.000	0.000	0.000
73	0.000	0.000	0.000	0.000	0.000
74	0.000	0.000	0.000	0.000	0.000
75	0.000	0.000	0.000	0.000	0.000
76	0.000	0.000	0.000	0.000	0.000
77	.	83.000	84.000	0.000	0.000
78	.	85.000	86.000	0.000	0.000
79	0.000	0.000	0.000	0.000	0.000
80	.	87.000	88.000	0.000	0.000
81	.	89.000	90.000	0.000	0.000
82	0.000	0.000	0.000	0.000	0.000
83	0.000	0.000	0.000	0.000	0.000
84	0.000	0.000	0.000	0.000	0.000
85	.	91.000	92.000	0.000	0.000
86	0.000	0.000	0.000	0.000	0.000
87	0.000	0.000	0.000	0.000	0.000
88	.	93.000	94.000	0.000	0.000
89	0.000	0.000	0.000	0.000	0.000
90	0.000	0.000	0.000	0.000	0.000
91	0.000	0.000	0.000	0.000	0.000
92	0.000	0.000	0.000	0.000	0.000
93	0.000	0.000	0.000	0.000	0.000

```
94 |      0.000      0.000      0.000      0.000      0.000
95 |      0.000      0.000      0.000      0.000      0.000


Remap=
   |   1   2   3   4   5   6   7   8   9  10  11  12  13  14
-------------------------------------------------------------
 1 |   1   0   1   1   0   0   1   1   1   1   0   0   1   1

   |  15  16  17  18  19  20  21  22  23  24  25  26  27
---------------------------------------------------------
 1 |   1   1   0   0   0   1   1   1   1   1   0   0   1


Ypredict=

  L |   Ypredict   Residual
---------------------------
  1 |    0.00000          0
  2 |    0.00000    0.00000
  3 |    0.00000    0.00000
  4 |    0.00000    0.00000
  5 |    0.00000    0.00000
  6 |    0.00000    0.00000
  7 |    0.00000    0.00000
  8 |    0.00000    0.00000
  9 |    1.00000    0.00000
 10 |    1.00000    0.00000

  L |   Ypredict   Residual
---------------------------
500 |    1.00000    0.00000
501 |    1.00000    0.00000
502 |    0.00000    0.00000
503 |    1.00000    0.00000
504 |    0.00000    0.00000
505 |    0.00000    0.00000
506 |    1.00000    0.00000
```

2. Australian Data Set: nobs=690, nvar=15

```
/* Australian: nobs=690, nvar=15 */
form = fo = " %g";
for (j = 2; j <= 15; j++) form = strcat(form,fo);
fid = fopen("..\\tdata\\australian.dat","r");
data = fscanf(fid,form,.,15);
```

```
nr = nrow(data);
print "Observations of Australian.dat:",nr;
x = data[,2:15]; y = data[,1];
modl = "1 = 2 : 15";
clas = [ 1 8 9 10 11 15 ];


/* use CHI2: bins=1000 */
optn = cons(10,1,.);
optn[1]= 2; optn[3]= 1000; optn[10]= 1;
/* options debug="bispl*=4"; */
< gof,nodes,remap,yptr > = split(data,modl,optn,clas);
```

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

                 Number Valid Observations  690
                 Response Variable          Y[1]
                 N Independend Variables     14
                 Number Parameters           41
                 Number Passes               12
                 Number Bins               1000
                 Chisquare Threshold  0.0000000
                 Design Coding:  Rank-Deficient


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

 Class  Level    Value

 Y[ 1]      2       0       1
 C[ 8]      2       0       1
 C[ 9]      2       0       1
 C[10]     23       0       1       2       3       4
                    5       6       7       8       9
                   10      11      12      13      14
                   15      16      17      19      20
                   23      40      67
 C[11]      2       0       1
 C[15]      2       0       1


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

```
                Split History
                *************

 Node Parnt         Split   ChiSquare   (Val,Lev)
    1     0            X6  16.5271718  3.00800000
--- Pass=2 SSE=222 Accuracy=0.678261 Misc=222 ---

    2     1            X3  5.43123288  2.01600000
    3     1            X5  19.1423837  10.0090000
--- Pass=3 SSE=210 Accuracy=0.695652 Misc=210 ---

    4     2            X3  5.65564961  1.21161600
    5     2            X6  4.67187500  2.00199200
    6     3           C10  13.6371849     2    18
    7     3            X5  20.5159173  11.0027590
--- Pass=4 SSE=260 Accuracy=0.623188 Misc=260 ---

    8     4            X3  6.34271605  0.50039741
    9     4            X5  13.0000000  5.00400000
   10     5            X3  3.25083612  3.00339200
   12     6           X13  5.14285714  82.0000000
   13     6            X7  6.58959772  8.52150000
   14     7           C10  8.00373134     6     9
   15     7            X2  8.11370301  20.7990000
--- Pass=5 SSE=177 Accuracy=0.743478 Misc=177 ---

   16     8           X13  5.83333333  46.0000000
   17     8            X7  9.00000000  0.02850000
   20    10           C11  3.93750000     1     1
   21    10            X3  6.07500000  6.00298496
   23    12           X13  3.00000000  254.620000
   24    13           X13  6.00936718  930.000000
   27    14            X7  7.76518283  3.04950000
   28    15            X3  5.00000000  1.51200000
   29    15           C10  5.83954802     7     8
--- Pass=6 SSE=174 Accuracy=0.747826 Misc=174 ---

   30    16            X2  3.00000000  22.7275000
   34    20            X2  2.00000000  23.1265000
   36    21            X2  3.93750000  18.8705000
   37    21           X13  2.75294118  186.000000
   40    24           X13  6.76305987  375.720000
   42    27           X14  3.02834008  501.000000
   43    27           X13  11.7333333  22.0000000
   46    29            X3  3.86416673  11.1440000
--- Pass=7 SSE=165 Accuracy=0.76087 Misc=165 ---
```

```
   53    36           C10  3.42857143     3     1
   54    37           X14  3.76739927  1201.00000
   56    40           X13  10.0705776  360.315480
   57    40           C10  10.4829545     1     8
   58    42           C10  3.25133690     3     4
   59    42           C10  5.95833333     6     1
   60    43            X3  5.00000000  4.64800000
   62    46           X13  5.61522647  142.000000
   63    46            X2  5.76000000  25.0200210
--- Pass=8 SSE=147 Accuracy=0.786957 Misc=147 ---


 Node Parnt         Split   ChiSquare   (Val,Lev)
   65    53            X7  2.00000000  0.02850000
   66    54            X2  6.46153846  17.9395000
   67    54           C10  3.00000000     2     1
   68    56            X4  5.74741711  1.00200000
   69    56           C10  6.00000000     3     1
   71    57            X6  5.08762421  7.00466400
   73    58            X2  2.92473118  16.5430000
   74    59            X3  12.0000000  10.0240000
   78    62           C10  3.95833333     1     4
   79    62           C10  4.69333333     4     3
   81    63           X13  1.87500000  2.00000000
--- Pass=9 SSE=131 Accuracy=0.810145 Misc=131 ---

   85    66           C10  5.95833333     1     4
   88    68            X7  15.6722074  0.00852150
   89    68            X3  5.00627096  4.25600000
   92    71            X7  3.08571429  2.50532100
   93    71           X13  3.42857143  432.256560
   95    73           X14  4.76317460  2.50000000
   98    78            X3  4.00000000  3.04231200
  100    79            X7  2.76679842  10.0035000
  103    81            X7  2.00000000  1.65300000
--- Pass=10 SSE=123 Accuracy=0.821739 Misc=123 ---

  104    85            X3  2.00000000  14.5158298
  107    88           X14  4.97674423  101.000000
  108    89            X3  7.08703565  1.46406400
  109    89           C10  6.98412698    10     6
  111    92            X7  9.00000000  3.00466386
  112    93            X2  2.91666667  24.5895000
  114    95           X13  2.52403846  262.000000
  115    95            X4  4.44444444  1.00200000
```

```
    118   100            X7  5.21848739  3.50122500
--- Pass=11 SSE=128 Accuracy=0.814493 Misc=128 ---

    124   107            X3  2.83610944  1.17600000
    125   107            X7  33.0000000  3.96705650
    126   108            X2  7.35161108  41.2145000
    127   108            X7  4.46198313  0.87771450
    128   109            X5  3.56321037  6.00900400
    132   112            X2  2.00000000  23.3321180
    134   114            X2  3.81818182  20.1105920
    135   114           C15  7.00000000     1     1
    137   115            X7  2.25000000  0.37508850
    138   118           C15  4.73533951     1     1
--- Pass=12 SSE=121 Accuracy=0.824638 Misc=121 ---

                  **************
                  Effect Summary
                  **************

             Effect Type 1st Split  N Splits
                 X6    V         1         3
                 X3    V         2        14
                 X5    V         3         4
                C10    C         6        13
                X13    V        12        12
                 X7    V        13        13
                 X2    V        15        11
                C11    C        20         1
                X14    V        42         4
                 X4    V        68         2
                C15    C       135         2

     *****************************************
     Residuals of Linear Regression (Training)
     *****************************************

       Nobs        Yobs        Yprd    Residual
          1  1.00000000  1.00000000  0.00000000
          2  0.00000000  1.00000000 -1.00000000
          3  0.00000000  1.00000000 -1.00000000
          4  0.00000000  0.00000000  0.00000000
          5  1.00000000  1.00000000  0.00000000
          6  0.00000000  1.00000000 -1.00000000
          7  1.00000000  1.00000000  0.00000000
          8  0.00000000  0.00000000  0.00000000
```

```
  9  1.00000000  1.00000000  0.00000000
 10  0.00000000  1.00000000 -1.00000000
 11  1.00000000  1.00000000  0.00000000
 12  1.00000000  1.00000000  0.00000000
 13  1.00000000  1.00000000  0.00000000
 14  1.00000000  1.00000000  0.00000000
 15  1.00000000  1.00000000  0.00000000
 16  1.00000000  1.00000000  0.00000000
 17  1.00000000  1.00000000  0.00000000
 18  0.00000000  1.00000000 -1.00000000
 19  1.00000000  1.00000000  0.00000000
 20  0.00000000  0.00000000  0.00000000


Nobs        Yobs        Yprd    Residual
 680  1.00000000  1.00000000  0.00000000
 681  1.00000000  1.00000000  0.00000000
 682  1.00000000  1.00000000  0.00000000
 683  0.00000000  0.00000000  0.00000000
 684  0.00000000  0.00000000  0.00000000
 685  1.00000000  1.00000000  0.00000000
 686  1.00000000  1.00000000  0.00000000
 687  1.00000000  1.00000000  0.00000000
 688  0.00000000  1.00000000 -1.00000000
 689  0.00000000  0.00000000  0.00000000
 690  1.00000000  1.00000000  0.00000000

      Sum-of-Squared Residual 98



Training Data SSE=98 Accuracy=0.857971 Misc=98
           Total Processing Time: 1
```

3. NIR Spectra Data Set: train: nr=21, test: nr=7

```
/*  (1) NIR Spectra data set: train: nr=21, test: nr=7                  */
/*      nvar= 268   (use median of y for binary target)                 */

 print "NIR Spectra data set: train: nr=21, test: nr=7";
 options NOECHO;
#include "..\\tdata\\nir.dat"
 options ECHO;
 nr = nrow(xtrn); nc = ncol(xtrn);
 print "nrtrn,nctrn=",nr,nc;
```

```
sopt = [ "ari" "std" "med" ];
mom = univar(ytrn`,sopt);
/* print "Moments of y=", mom; */
cutof = mom[3];
y = ytrn .< cutof; tnir = xtrn -> y`; /* attrib(tnir); */
y = ytst .< cutof; test = xtst -> y`; /* attrib(test); */
/* print tnir[ , 200:269 ]; */
/* model: */
xind = [ 1:268 ]; yind = 269;


/* use chi2: bins=100 */
modl = "269 = 1:268";
clas = 269;
optn = cons(10,1,.);
optn[1]= 3; optn[10]= 1;
< gof,nodes,remap,yptr > = split(tnir,modl,optn,clas);


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

                 Number Valid Observations   21
                 Response Variable        Y[269]
                 N Independend Variables    268
                 Number Parameters          269
                 Number Passes               12
                 Number Bins                100
                 Chisquare Threshold  0.0000000
                 Design Coding:  Rank-Deficient


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

 Class  Level    Value

Y[  1]      2        0       1


                         *************
                         Split History
                         *************

         Node Parnt         Split   ChiSquare   (Val,Lev)
```

```
       1     0           X51  21.0000000  2.60646400


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

Index                                  Value       StdErr
Absolute Classification Error              0       .
Classification Accuracy          100.0000000       .
Concordant Pairs                 100.0000000       .
Discordant Pairs                 0.000000000       .
Tied Pairs                       0.000000000       .
Goodman-Kruskal Gamma            1.000000000  0.000000000
Kendall Tau_b                    1.000000000  0.000000000
Stuart Tau_c                     0.997732426  0.020759080
Somers D C|R                     1.000000000  0.000000000


                   Classification Table
                   --------------------

              |    Predicted
     Observed |        0       1
     ---------|-----------------
            0 |       11       0
            1 |        0      10



                  **************
                  Effect Summary
                  **************

                Effect Type 1st Split  N Splits
                   X51    V         1         1


        ****************************************
        Residuals of Linear Regression (Training)
        ****************************************

          Nobs        Yobs        Yprd    Residual
             1  0.00000000  0.00000000  0.00000000
             2  0.00000000  0.00000000  0.00000000
             3  0.00000000  0.00000000  0.00000000
```

4	0.00000000	0.00000000	0.00000000
5	0.00000000	0.00000000	0.00000000
6	0.00000000	0.00000000	0.00000000
7	0.00000000	0.00000000	0.00000000
8	0.00000000	0.00000000	0.00000000
9	0.00000000	0.00000000	0.00000000
10	0.00000000	0.00000000	0.00000000
11	1.00000000	1.00000000	0.00000000
12	1.00000000	1.00000000	0.00000000
13	1.00000000	1.00000000	0.00000000
14	1.00000000	1.00000000	0.00000000
15	0.00000000	0.00000000	0.00000000
16	1.00000000	1.00000000	0.00000000
17	1.00000000	1.00000000	0.00000000
18	1.00000000	1.00000000	0.00000000
19	1.00000000	1.00000000	0.00000000
20	1.00000000	1.00000000	0.00000000
21	1.00000000	1.00000000	0.00000000

Sum-of-Squared Residual 0

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

Index	Value	StdErr
Absolute Classification Error	0	.
Classification Accuracy	100.0000000	.
Concordant Pairs	100.0000000	.
Discordant Pairs	0.000000000	.
Tied Pairs	0.000000000	.
Goodman-Kruskal Gamma	1.000000000	0.000000000
Kendall Tau_b	1.000000000	0.000000000
Stuart Tau_c	0.997732426	0.020759080
Somers D C|R	1.000000000	0.000000000

```
     Classification Table
     --------------------

         |    Predicted
Observed |        0       1
---------|-----------------
       0 |       11       0
       1 |        0      10
```

```
               Training Data SSE=0 Accuracy=1 Misc=0
                     Total Processing Time: 0


 print "GOF=", gof;
 print "Nodes=", nodes;
 print "Ypredict=", yptr;


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

                    Dense Row Vector (ncol=25)

R |      Npasses     Nnodes  SSQ_error        N00        N01
       3.0000000  3.0000000  0.0000000  11.000000  0.0000000

R |          N10        N11   ClassErr   Accuracy     unused
       0.0000000  10.000000  0.0000000  1.0000000     .

R |       unused     unused     unused     unused     unused
          .          .          .          .          .

R |       unused     unused     unused     unused     unused
          .          .          .          .          .

R |       unused     unused     unused     unused     unused
          .          .          .          .          .


                           Tree of Nodes
                           *************

                      Dense Matrix (3 by 15)

  |         Pass       Node       Type      Index     Parent
-----------------------------------------------------------
1 |    1.0000000  0.0000000  0.0000000  50.000000 -1.0000000
2 |    2.0000000  1.0000000  0.0000000 -1.0000000  0.0000000
3 |    2.0000000  2.0000000  0.0000000 -1.0000000  0.0000000

  |       Effect      Value      Ymean     Chisqu     Level1
-----------------------------------------------------------
1 |    50.000000  2.6064640  0.4761905  21.000000      .
2 |   -1.0000000      .      0.0000000 -1.0000000  0.0000000
3 |   -1.0000000      .      1.0000000 -1.0000000  0.0000000
```

	Level2	NodLft	NodRgt	Remap	unused
1	.	1.0000000	2.0000000	0.0000000	0.0000000
2	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
3	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000

GOF=

	Npasses	Nnodes	SSQ_error	N00	N01
1	3.0000	3.0000	0.0000	11.0000	0.0000

	N10	N11	ClassErr	Accuracy	unused
1	0.0000	10.0000	0.0000	1.0000	.

	unused	unused	unused	unused	unused
1	.	.	.	.	.

Page 7

	unused	unused	unused	unused	unused
1	.	.	.	.	.

	unused	unused	unused	unused	unused
1	.	.	.	.	.

Nodes=

	Pass	Node	Type	Index	Parent
1	1.0000	0.0000	0.0000	50.0000	-1.0000
2	2.0000	1.0000	0.0000	-1.0000	0.0000
3	2.0000	2.0000	0.0000	-1.0000	0.0000

	Effect	Value	Ymean	Chisqu	Level1
1	50.0000	2.6065	0.4762	21.0000	.
2	-1.0000	.	0.0000	-1.0000	0.0000
3	-1.0000	.	1.0000	-1.0000	0.0000

	Level2	NodLft	NodRgt	Remap	unused
1	.	1.0000	2.0000	0.0000	0.0000
2	0.0000	0.0000	0.0000	0.0000	0.0000

```
  3 |    0.0000    0.0000    0.0000    0.0000    0.0000
```

```
Yprediict=
```

```
Ypredict=
 L |   Ypredict   Residual
--------------------------
 1 |    0.00000          0
 2 |    0.00000    0.00000
 3 |    0.00000    0.00000
 4 |    0.00000    0.00000
 5 |    0.00000    0.00000
 6 |    0.00000    0.00000
 7 |    0.00000    0.00000
 8 |    0.00000    0.00000
 9 |    0.00000    0.00000
10 |    0.00000    0.00000
11 |    1.00000    0.00000
12 |    1.00000    0.00000
13 |    1.00000    0.00000
14 |    1.00000    0.00000
15 |    0.00000    0.00000
16 |    1.00000    0.00000
17 |    1.00000    0.00000
18 |    1.00000    0.00000
19 |    1.00000    0.00000
20 |    1.00000    0.00000
21 |    1.00000    0.00000
```

3.8 Function `varclus`

<gof,est,tree,scor,struct,trace> = varclus(data,optn<,ingrp>)

Purpose: The `varclus` function implements an algorithm for variable clustering similar to that of PROC VARCLUS in SAS STAT. However, the algorithm was extended for the important case where there are more variables than observations $nvar \gg Nobs$. In that case an $Nobs \times nvar$ raw data matrix must be provided which is kept incore. Memory of order $nvar^2$ is not needed but instead memory of order $Nobs^2$ is used. This modification is many times computationally slower but makes it possible to solve applications in Chemometrics and the analysis of Microarray data where the $nvar \times nvar$ covariance or correlation matrix cannot be stored incore.

Input: data The input data can be

1. an $N \times n$ raw data set
2. a symmetric $n \times n$ corrected or uncorrected correlation or covariance matrix.

Note, as with the `factor` function, the correlation or covariance input matrix can be extended with additional rows for mean and standard deviation of the variables.

optn : is specified in form of a two column matrix:

Option Name	Second Column	Meaning
"anal"	string	which data are analyzed
	"corr"	the corrected correlation matrix
	"cov"	the corrected covariance matrix
	"ucorr"	the uncorrected correlation matrix
	"ucov"	the uncorrected covariance matrix
"cent"		perform centroid analysis
"data"	string	what kind of data is given
	"corr"	the corrected correlation matrix
	"cov"	the corrected covariance matrix
	"ucorr"	the uncorrected correlation matrix
	"ucov"	the uncorrected covariance matrix
"evtech"	string	algorithm for eigenvalue computation
	"tql"	TQL (Householder) is default
	"bis"	bisection method
	"arp"	Arnoldi iteration (ARPACK)
"hier"		obtain hierarchical structure
		this is easier, faster search

Option Name	Second Column	Meaning
"init"	string	kind of initial structure
		specified in third input argument
	"group"	ingrp contains n entries with
		initial cluster membership in [1,maxcl]
	"input"	ingrp contains n entries with
		cluster scores
	"rand"	variables are randomly assigned to clusters
	"seed"	ingrp contains K entries specifying
		variable numbers for cluster representatives
"mincl"	int	lower bound for number of clusters
"maxcl"	int	upper bound for number of clusters
"maxit"	int	maximum number of iterations in ALS
		default: centroid: 1, otherwise 10
"maxse"	int	maximum number of searches
		default: centroid: 10, otherwise 0
"maxeig"	real	lower bound for second eigenvalue
"mult"		perform Harman's multigroup analysis
		mincl=1, maxiter=maxsearch=prop=0,
		init="group"
"nobs"	int	number of observations
"noint"		specifies uncorrected input data
"nopr"		suppresses printed output
"nomis"		skip observations with missing values
"perc"	real	value in [1,100] specifying percentage of
		variance explaind by cluster structure
"prop"	real	same as perc but value is in $[0, 1]$
"prin"	int	amount of printed output
"sing"	real	singularity criterion (def=1.e-8)
"seed"	int	seed parameter for random generator
		default is time of day
"trans"		solve the transposed problem
		useful only for $nvar \gg Nobs$
"vardef"	string	definition of variance divisor; string may be "n",
		"df", "wgt", or "wdf".
"vardiv"	real	variance divisor s (default=nobs-1)

The `"trans"` option is not computationally efficient for $Nobs \gg nvar$ but becomes efficient when $nvar \gg Nobs$. See the example using the NIR data.

There are two cases where the `"arp"` (Arpack) option can save computer time compared to the `"tql"` or `"bis"` option:

1. for large $nvar$ when $nvar < Nobs$, i.e. `"trans"` is not specified;
2. for large $Nobs$ when $nvar > Nobs$, i.e. `"trans"` is specified.

ingrp (optional) vector used for specifying initial cluster structure (see the `init` option.

Output: Assuming there are K steps of the cluster tree, starting with $k1$ clusters and ending with $k2$ clusters. The number $k2$ is determined by the input specification of `maxclus` (as upper bound), the `maxeig`, and the `proport` option vector entries.

gof a vector of goodness-of-fit indicators

est $5K \times n$ matrix of cluster properties

tree

scor

struct

trace

Restrictions: 1. Missing values in raw data are replaced by var means. Missing values in correlation or covariance matrices are not permitted.

2.

Relationships: cluster()

Examples: 1. First Example in SAS/STAT Manual:

```
options ls=68 ps=2000;
print "VARCL01: EIGHT PHYSICAL VARIABLES: HARMAN(1967), p.80";
print "Data from MULLEN (1939): Test Akaike IC and Schwarz IC";

corr = [  1.0     .846    .805    .859    .473    .398    .301    .382,
          .846   1.0      .881    .826    .376    .326    .277    .415,
          .805    .881   1.0      .801    .380    .319    .237    .345,
          .859    .826    .801   1.0      .436    .329    .327    .365,
          .473    .376    .380    .436   1.0      .762    .730    .629,
          .398    .326    .319    .329    .762   1.0      .583    .577,
          .301    .277    .237    .327    .730    .583   1.0      .539,
          .382    .415    .345    .365    .629    .577    .539   1.0 ];


labl = [ "HEIGHT"  "ARM SPAN"  "LENGTH OF FOREARM"
         "LENGTH OF LOWER LEG"  "WEIGHT"  "BITROCHANTERIC DIAMETER"
         "CHEST GIRTH"   "CHEST WIDTH" ];
name = [ "Height" "Arm_Span" "Forearm" "Low_Leg"
         "Weight" "Bit_Diam" "Girth"   "Width"  ];
corr = corr;  corr = cname(corr,name);


/*--- Principal Comp: 2 Clusters by maxeig=1 ---*/
optn = [ "print"         3 ,
         "data"     "corr" ,
         "anal"     "corr" ,
```

```
        "nobs"       305 ];
< gof,est,tree,scor,struc,trac > = varclus(corr,optn);


          *******************************************
          Oblique Principal Component Cluster Analysis
          *******************************************

  Number of Subjects. . . . . . . . . . . . . . . .      305
  Number of Variables . . . . . . . . . . . . . . .        8
  Specified Minimum Number of Clusters. . . . . . .        1
  Specified Maximum Number of Clusters. . . . . . .        8
  Proportion. . . . . . . . . . . . . . . . . . . .        .
  Maximum Second Eigenvalue . . . . . . . . . . . .        .
  Maximum Number of Iterations. . . . . . . . . . .       10

                       Correlation Matrix

             Height   Arm_Span    Forearm    Low_Leg     Weight
Height     1.0000000
Arm_Span   0.8460000  1.0000000
Forearm    0.8050000  0.8810000  1.0000000
Low_Leg    0.8590000  0.8260000  0.8010000  1.0000000
Weight     0.4730000  0.3760000  0.3800000  0.4360000  1.0000000
Bit_Diam   0.3980000  0.3260000  0.3190000  0.3290000  0.7620000
Girth      0.3010000  0.2770000  0.2370000  0.3270000  0.7300000
Width      0.3820000  0.4150000  0.3450000  0.3650000  0.6290000

                       Correlation Matrix

                       Bit_Diam      Girth      Width
           Bit_Diam   1.0000000
           Girth      0.5830000  1.0000000
           Width      0.5770000  0.5390000  1.0000000
      Determinant =  1.00000000e+000 (Ln =  0.0000e+000)

                   **************************
                   One Cluster Initialization
                   **************************


                ********************************
                Cluster Summary for 1 Cluster(s)
```

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

                       Cluster  Variation Proportion     Second
    Cluster Members  Variation  Explained  Explained Eigenvalue
          1       8  8.0000000  4.6728796  0.5841099  1.7709828

      Total Variation Explained = 4.67288 Proportion = 0.5841
Cluster 1 will be split.


                  *******************************
                  Cluster Summary for 2 Cluster(s)
                  *******************************

                       Cluster  Variation Proportion     Second
    Cluster Members  Variation  Explained  Explained Eigenvalue
          1       4  4.0000000  3.5092182  0.8773046  0.2361353
          2       4  4.0000000  2.9172841  0.7293210  0.4764177

      Total Variation Explained = 6.4265 Proportion = 0.8033


                           **************
                           R-squared with
                           **************

                                    Own       Next     1-R**2
                    Variable    Cluster    Closest      Ratio
      Cluster  4-------------------------------------------------
                      Height  0.8777469  0.2087650  0.1545092
                    Arm_Span  0.9002460  0.1658202  0.1195833
                     Forearm  0.8660604  0.1412742  0.1559748
                     Low_Leg  0.8651649  0.1829399  0.1650247
      Cluster  4-------------------------------------------------
                      Weight  0.8476636  0.1973853  0.1898002
                    Bit_Diam  0.7385756  0.1341028  0.3019116
                       Girth  0.6981094  0.0928904  0.3328049
                       Width  0.6329355  0.1619496  0.4379981


                 Standardized Scoring Coefficients

                                  CLUS1      CLUS2
                  Height     0.2669773  0.0000000
                  Arm_Span   0.2703773  0.0000000
                  Forearm    0.2651940  0.0000000
```

Low_Leg	0.2650569	0.0000000
Weight	0.0000000	0.3155971
Bit_Diam	0.0000000	0.2945905
Girth	0.0000000	0.2864066
Width	0.0000000	0.2727100

Cluster Structure

	CLUS1	CLUS2
Height	0.9368815	0.4569081
Arm_Span	0.9488130	0.4072103
Forearm	0.9306237	0.3758646
Low_Leg	0.9301424	0.4277147
Weight	0.4442806	0.9206865
Bit_Diam	0.3662006	0.8594042
Girth	0.3047792	0.8355294
Width	0.4024296	0.7955724

Inter-Cluster Correlations

	CLUS1	CLUS2
CLUS1	1.0000000	0.4451302
CLUS2	0.4451302	1.0000000

No cluster meets the criterion for splitting.

****
Oblique Principal Component Cluster Analysis
****

Number of Clusters	Total Variation Explained by Clusters	Proportion of Variation Explained by a Cluster	Minimum Proportion Explained by a Cluster	Maximum Second Eigenvalue in a Cluster
1	4.672879598	0.584109950	0.584109950	1.770982845
2	6.426502368	0.803312796	0.729321030	0.476417740

Number of Clusters	Minimum R**2 for a Variable	Maximum 1-R**2 for a Variable
1	0.380959614	.
2	0.632935494	0.437998113

The following shows ther content of the return arguments:

```
print "GOF=", gof;
print "Est=", est;
print "Tree=", tree;
print "Score=", scor;
print "Struc=", struc;
print "Trace=", trac;
```

```
GOF=
             |        1
-------------------------
  Return_Code |   0.00000
     Totalvar |   8.00000
     VAFtotal |   6.42650
      CorrDet |   1.00000
         Time |   1.00000
       unused |    .
       unused |    .
       unused |    .
       unused |    .
       unused |    .
```

```
Est=
          |        1         2         3         4
---------------------------------------------------
   Member |  8.00000    .         .         .
     VAF1 |  4.67288    .         .         .
  Portion |  0.58411    .         .         .
   Groups |  1.00000   1.00000   1.00000   1.00000
  Rsquare |  4.67288   0.00000   0.00000   0.00000
   Member |  4.00000   4.00000    .         .
     VAF2 |  3.50922   2.91728    .         .
  Portion |  0.87730   0.72932    .         .
   Groups |  1.00000   1.00000   1.00000   1.00000
  Rsquare |  3.50922   2.91728   0.00000   0.00000

          |        5         6         7         8
---------------------------------------------------
   Member |   .         .         .         .
     VAF1 |   .         .         .         .
  Portion |   .         .         .         .
   Groups |  1.00000   1.00000   1.00000   1.00000
```

Rsquare	0.00000	0.00000	0.00000	0.00000
Member	.	.	.	.
VAF2	.	.	.	.
Portion	.	.	.	.
Groups	2.00000	2.00000	2.00000	2.00000
Rsquare	0.00000	0.00000	0.00000	0.00000

Tree=

	Name	Parent	NCl	Varexp
1	CLUS1		1.000000	4.672880
2	CLUS2	CLUS1	2.000000	6.426502
3	CLUS3	CLUS1	2.000000	6.426502
4	Height	CLUS2	8.000000	8.000000
5	Arm_Span	CLUS2	8.000000	8.000000
6	Forearm	CLUS2	8.000000	8.000000
7	Low_Leg	CLUS2	8.000000	8.000000
8	Weight	CLUS3	8.000000	8.000000
9	Bit_Diam	CLUS3	8.000000	8.000000
10	Girth	CLUS3	8.000000	8.000000
11	Width	CLUS3	8.000000	8.000000

	Propor	MinPro	MaxEig
1	0.584110	0.584110	1.770983
2	0.803313	0.729321	0.476418
3	0.803313	0.729321	0.476418
4	1.000000	1.000000	0.000000
5	1.000000	1.000000	0.000000
6	1.000000	1.000000	0.000000
7	1.000000	1.000000	0.000000
8	1.000000	1.000000	0.000000
9	1.000000	1.000000	0.000000
10	1.000000	1.000000	0.000000
11	1.000000	1.000000	0.000000

Score=

UPP	Height	Arm_Span	Forearm	Low_Leg
CLUS1	0.26698	0.27038	0.26519	0.26506
CLUS2	0	0.00000	0.00000	0.00000

UPP	Weight	Bit_Diam	Girth	Width

```
  CLUS1 |     0.00000     0.00000     0.00000     0.00000
  CLUS2 |     0.31560     0.29459     0.28641     0.27271


Struc=
        |      Height    Arm_Span     Forearm     Low_Leg
---------------------------------------------------------
  CLUS1 |     0.93688     0.94881     0.93062     0.93014
  CLUS2 |     0.45691     0.40721     0.37586     0.42771

        |      Weight    Bit_Diam       Girth       Width
---------------------------------------------------------
  CLUS1 |     0.44428     0.36620     0.30478     0.40243
  CLUS2 |     0.92069     0.85940     0.83553     0.79557
```

2. Second Example in SAS/STAT Manual:

```
/*--- Centroid Analysis: 3 Clusters by Proportion ---*/
optn = [ "print"        2 ,
         "data"     "corr" ,
         "anal"     "corr" ,
         "cent"            ,
         "nobs"       305 ];
< gof,est,tree,scor,struc,trac > = varclus(corr,optn);


          *******************************************
          Oblique Centroid Component Cluster Analysis
          *******************************************

  Number of Subjects. . . . . . . . . . . . . . . .      305
  Number of Variables . . . . . . . . . . . . . . .        8
  Specified Minimum Number of Clusters. . . . . . .        1
  Specified Maximum Number of Clusters. . . . . . .        8
  Proportion. . . . . . . . . . . . . . . . . . . . 0.7500000
  Maximum Second Eigenvalue . . . . . . . . . . . .    .
  Maximum Number of Iterations. . . . . . . . . . .        1
  Maximum Number of Searches. . . . . . . . . . . .       10


                   **************************
                   One Cluster Initialization
                   **************************


                ********************************
                Cluster Summary for 1 Cluster(s)
```

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

                         Cluster  Variation Proportion     Second
       Cluster Members  Variation  Explained  Explained Eigenvalue
             1       8  8.0000000  4.6310000  0.5788750          .

          Total Variation Explained = 4.631 Proportion = 0.5789
Cluster 1 will be split.


                    *******************************
                    Cluster Summary for 2 Cluster(s)
                    *******************************

                         Cluster  Variation Proportion     Second
       Cluster Members  Variation  Explained  Explained Eigenvalue
             1       4  4.0000000  3.5090000  0.8772500          .
             2       4  4.0000000  2.9100000  0.7275000          .

          Total Variation Explained = 6.419 Proportion = 0.8024


                             **************
                             R-squared with
                             **************

                                     Own       Next     1-R**2
                      Variable   Cluster    Closest      Ratio
         Cluster  4-------------------------------------------
                        Height  0.8777501  0.2074670  0.1542522
                      Arm_Span  0.8993879  0.1669447  0.1207748
                       Forearm  0.8662845  0.1409760  0.1556598
                       Low_Leg  0.8657877  0.1823753  0.1641491
         Cluster  4-------------------------------------------
                        Weight  0.8368248  0.1975082  0.2033356
                      Bit_Diam  0.7335124  0.1341111  0.3077619
                         Girth  0.6987890  0.0929156  0.3320650
                         Width  0.6473389  0.1618017  0.4207371

Cluster 2 will be split.


                    *******************************
                    Cluster Summary for 3 Cluster(s)
                    *******************************

                         Cluster  Variation Proportion     Second
```

Cluster	Members	Variation	Explained	Explained	Eigenvalue
1	4	4.0000000	3.5090000	0.8772500	.
2	3	3.0000000	2.3833333	0.7944444	.
3	1	1.0000000	1.0000000	1.0000000	.

Total Variation Explained = 6.89233 Proportion = 0.8615

****
R-squared with
****

	Variable	Own Cluster	Next Closest	1-R**2 Ratio
Cluster 4				
	Height	0.8777501	0.1921097	0.1513200
	Arm_Span	0.8993879	0.1722250	0.1215452
	Forearm	0.8662845	0.1225309	0.1523877
	Low_Leg	0.8657877	0.1667782	0.1610763
Cluster 3				
	Weight	0.8685404	0.3956410	0.2175190
	Bit_Diam	0.7690944	0.3329290	0.3461485
	Girth	0.7482474	0.2905210	0.3548415
Cluster 1				
	Width	1.0000000	0.4258776	0.0000000

No cluster meets the criterion for splitting.

****
Oblique Centroid Component Cluster Analysis
****

Number of Clusters	Total Variation Explained by Clusters	Proportion of Variation Explained by Clusters	Minimum Proportion Explained by a Cluster	Maximum Second Eigenvalue in a Cluster
1	4.631000000	0.578875000	0.578875000	0.000000000
2	6.419000000	0.802375000	0.727500000	0.000000000
3	6.892333333	0.861541667	0.794444444	0.000000000

Number of Clusters	Minimum R**2 for a Variable	Maximum 1-R**2 for a Variable

1	0.430577521	.
2	0.647338918	0.420737065
3	0.748247413	0.354841493

Goodness-of-Fit Vector
****

Dense Column Vector (nrow=10)

C |	Return_Code	Totalvar	VAFtotal	CorrDet	Time
	0.0000000	8.0000000	6.8923333	1.0000000	0.0000000

C |	unused	unused	unused	unused	unused
	.	.	.	.	.

Estimates
****

Dense Matrix (15 by 8)

	1	2	3	4	5
Member	8.0000000	.	.	.	.
VAF1	4.6310000	.	.	.	.
Portion	0.5788750	.	.	.	.
Groups	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
Rsquare	4.6310000	0.0000000	0.0000000	0.0000000	0.0000000
Member	4.0000000	4.0000000	.	.	.
VAF2	3.5090000	2.9100000	.	.	.
Portion	0.8772500	0.7275000	.	.	.
Groups	1.0000000	1.0000000	1.0000000	1.0000000	2.0000000
Rsquare	3.5090000	2.9100000	0.0000000	0.0000000	0.0000000
Member	4.0000000	3.0000000	1.0000000	.	.
VAF3	3.5090000	2.3833333	1.0000000	.	.
Portion	0.8772500	0.7944444	1.0000000	.	.
Groups	1.0000000	1.0000000	1.0000000	1.0000000	2.0000000
Rsquare	3.5090000	2.3833333	1.0000000	0.0000000	0.0000000

	6	7	8
Member	.	.	.
VAF1	.	.	.
Portion	.	.	.
Groups	1.0000000	1.0000000	1.0000000
Rsquare	0.0000000	0.0000000	0.0000000

```
  Member |      .         .         .
    VAF2 |      .         .         .
 Portion |      .         .         .
  Groups |   2.0000000  2.0000000  2.0000000
 Rsquare |   0.0000000  0.0000000  0.0000000
  Member |      .         .         .
    VAF3 |      .         .         .
 Portion |      .         .         .
  Groups |   2.0000000  2.0000000  3.0000000
 Rsquare |   0.0000000  0.0000000  0.0000000
```

```
                          Tree Structure
                          **************

                     Sparse Matrix (12 by 7)

     1 |        Name     Parent        NCl     Varexp     Propor
               CLUS1              1.0000000  4.6310000  0.5788750
       |      MinPro     MaxEig
           0.5788750  0.0000000

     2 |        Name     Parent        NCl     Varexp     Propor
               CLUS3      CLUS1  2.0000000  6.4190000  0.8023750
       |      MinPro     MaxEig
           0.7275000  0.0000000

     3 |        Name     Parent        NCl     Varexp     Propor
               CLUS2      CLUS1  3.0000000  6.8923333  0.8615417
       |      MinPro     MaxEig
           0.7944444  0.0000000

     4 |        Name     Parent        NCl     Varexp     Propor
               CLUS4      CLUS3  3.0000000  6.8923333  0.8615417
       |      MinPro     MaxEig
           0.7944444  0.0000000

     5 |        Name     Parent        NCl     Varexp     Propor
              Height      CLUS2  8.0000000  8.0000000  1.0000000
       |      MinPro     MaxEig
           1.0000000  0.0000000

     6 |        Name     Parent        NCl     Varexp     Propor
            Arm_Span      CLUS2  8.0000000  8.0000000  1.0000000
       |      MinPro     MaxEig
           1.0000000  0.0000000
```

```
 7 |        Name     Parent        NCl     Varexp     Propor
         Forearm      CLUS2  8.0000000  8.0000000  1.0000000
   |      MinPro     MaxEig
       1.0000000  0.0000000

 8 |        Name     Parent        NCl     Varexp     Propor
         Low_Leg      CLUS2  8.0000000  8.0000000  1.0000000
   |      MinPro     MaxEig
       1.0000000  0.0000000

 9 |        Name     Parent        NCl     Varexp     Propor
          Weight      CLUS4  8.0000000  8.0000000  1.0000000
   |      MinPro     MaxEig
       1.0000000  0.0000000

10 |        Name     Parent        NCl     Varexp     Propor
        Bit_Diam      CLUS4  8.0000000  8.0000000  1.0000000
   |      MinPro     MaxEig
       1.0000000  0.0000000

11 |        Name     Parent        NCl     Varexp     Propor
           Girth      CLUS4  8.0000000  8.0000000  1.0000000
   |      MinPro     MaxEig
       1.0000000  0.0000000

12 |        Name     Parent        NCl     Varexp     Propor
           Width      CLUS3  8.0000000  8.0000000  1.0000000
   |      MinPro     MaxEig
       1.0000000  0.0000000
```

3. Third Example in SAS/STAT Manual:

```
/*--- Principal Comp: 8 Clusters Specified ---*/
optn = [ "print"         2 ,
         "data"     "corr" ,
         "anal"     "corr" ,
         "maxcl"         8 ,
         "nobs"        305 ];
< gof,est,tree,scor,struc,trac > = varclus(corr,optn);
```

```
          ********************************************
          Oblique Principal Component Cluster Analysis
```

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

    Number of Subjects. . . . . . . . . . . . . . . .      305
    Number of Variables . . . . . . . . . . . . . . .        8
    Specified Minimum Number of Clusters. . . . . . .        1
    Specified Maximum Number of Clusters. . . . . . .        8
    Proportion. . . . . . . . . . . . . . . . . . . . 1.0000000
    Maximum Second Eigenvalue . . . . . . . . . . . . 0.0000000
    Maximum Number of Iterations. . . . . . . . . . .       10

                     **************************
                     One Cluster Initialization
                     **************************


                  ********************************
                  Cluster Summary for 1 Cluster(s)
                  ********************************

                     Cluster  Variation Proportion     Second
    Cluster Members  Variation  Explained  Explained Eigenvalue
          1       8  8.0000000  4.6728796  0.5841099  1.7709828

      Total Variation Explained = 4.67288 Proportion = 0.5841
Cluster 1 will be split.

                  ********************************
                  Cluster Summary for 2 Cluster(s)
                  ********************************

                     Cluster  Variation Proportion     Second
    Cluster Members  Variation  Explained  Explained Eigenvalue
          1       4  4.0000000  3.5092182  0.8773046  0.2361353
          2       4  4.0000000  2.9172841  0.7293210  0.4764177

      Total Variation Explained = 6.4265 Proportion = 0.8033

                        **************
                        R-squared with
                        **************

                               Own       Next     1-R**2
                    Variable    Cluster    Closest      Ratio
      Cluster  4----------------------------------------------
```

```
                  Height  0.8777469  0.2087650  0.1545092
                Arm_Span  0.9002460  0.1658202  0.1195833
                 Forearm  0.8660604  0.1412742  0.1559748
                 Low_Leg  0.8651649  0.1829399  0.1650247

      Cluster  4-------------------------------------------
                  Weight  0.8476636  0.1973853  0.1898002
                Bit_Diam  0.7385756  0.1341028  0.3019116
                   Girth  0.6981094  0.0928904  0.3328049
                   Width  0.6329355  0.1619496  0.4379981

Cluster 2 will be split.


                  ********************************
                  Cluster Summary for 3 Cluster(s)
                  ********************************

                       Cluster  Variation Proportion     Second
    Cluster Members  Variation  Explained  Explained Eigenvalue
          1       4  4.0000000  3.5092182  0.8773046  0.2361353
          2       3  3.0000000  2.3861292  0.7953764  0.4183688
          3       1  1.0000000  1.0000000  1.0000000      .

      Total Variation Explained = 6.89535 Proportion = 0.8619


                           **************
                           R-squared with
                           **************

                                Own       Next     1-R**2
                Variable    Cluster    Closest      Ratio
      Cluster  4-------------------------------------------
                  Height  0.8777469  0.1939258  0.1516648
                Arm_Span  0.9002460  0.1722250  0.1205085
                 Forearm  0.8660604  0.1237435  0.1528543
                 Low_Leg  0.8651649  0.1680161  0.1620646
      Cluster  3-------------------------------------------
                  Weight  0.8754128  0.3956410  0.2061477
                Bit_Diam  0.7685724  0.3329290  0.3469309
                   Girth  0.7421440  0.2905210  0.3634442
      Cluster  1-------------------------------------------
                   Width  1.0000000  0.4267125  0.0000000

Cluster 2 will be split.
```

```
                    *******************************
                    Cluster Summary for 4 Cluster(s)
                    *******************************

                         Cluster  Variation Proportion     Second
      Cluster Members  Variation  Explained  Explained Eigenvalue
            1       4  4.0000000  3.5092182  0.8773046  0.2361353
            2       2  2.0000000  1.7620000  0.8810000  0.2380000
            3       1  1.0000000  1.0000000  1.0000000      .
            4       1  1.0000000  1.0000000  1.0000000      .

        Total Variation Explained = 7.27122 Proportion = 0.9089

                             **************
                             R-squared with
                             **************

                                        Own       Next     1-R**2
                      Variable    Cluster    Closest      Ratio
        Cluster  4-------------------------------------------------
                        Height  0.8777469  0.2152784  0.1557916
                      Arm_Span  0.9002460  0.1722250  0.1205085
                       Forearm  0.8660604  0.1386495  0.1554995
                       Low_Leg  0.8651649  0.1660684  0.1616861
        Cluster  2-------------------------------------------------
                        Weight  0.8810000  0.5329000  0.2547634
                      Bit_Diam  0.8810000  0.3398890  0.1802727
        Cluster  1-------------------------------------------------
                         Width  1.0000000  0.4127230  0.0000000
        Cluster  1-------------------------------------------------
                         Girth  1.0000000  0.4892080  0.0000000

Cluster 2 will be split.

                    *******************************
                    Cluster Summary for 5 Cluster(s)
                    *******************************

                         Cluster  Variation Proportion     Second
      Cluster Members  Variation  Explained  Explained Eigenvalue
            1       4  4.0000000  3.5092182  0.8773046  0.2361353
            2       1  1.0000000  1.0000000  1.0000000      .
            3       1  1.0000000  1.0000000  1.0000000      .
            4       1  1.0000000  1.0000000  1.0000000      .
            5       1  1.0000000  1.0000000  1.0000000      .
```

```
           Total Variation Explained = 7.50922 Proportion = 0.9387

                      **************
                      R-squared with
                      **************

                              Own       Next     1-R**2
                  Variable  Cluster   Closest     Ratio
      Cluster  4-------------------------------------------
                    Height  0.8777469  0.2237290  0.1574876
                  Arm_Span  0.9002460  0.1722250  0.1205085
                   Forearm  0.8660604  0.1444000  0.1565446
                   Low_Leg  0.8651649  0.1900960  0.1664829
      Cluster  1-------------------------------------------
                  Bit_Diam  1.0000000  0.5806440  0.0000000
      Cluster  1-------------------------------------------
                     Width  1.0000000  0.3956410  0.0000000
      Cluster  1-------------------------------------------
                     Girth  1.0000000  0.5329000  0.0000000
      Cluster  1-------------------------------------------
                    Weight  1.0000000  0.5806440  0.0000000

Cluster 1 will be split.

                 *******************************
                 Cluster Summary for 6 Cluster(s)
                 *******************************

                    Cluster  Variation Proportion     Second
    Cluster Members  Variation  Explained  Explained Eigenvalue
          1       2  2.0000000  1.8590000  0.9295000  0.1410000
          2       1  1.0000000  1.0000000  1.0000000      .
          3       1  1.0000000  1.0000000  1.0000000      .
          4       1  1.0000000  1.0000000  1.0000000      .
          5       1  1.0000000  1.0000000  1.0000000      .
          6       2  2.0000000  1.8810000  0.9405000  0.1190000

        Total Variation Explained = 7.74 Proportion = 0.9675

                      **************
                      R-squared with
                      **************
```

```
                              Own       Next     1-R**2
              Variable    Cluster    Closest      Ratio
Cluster  2-----------------------------------------------
                Height  0.9295000  0.7245617  0.2559557
               Low_Leg  0.9295000  0.7036494  0.2378939
Cluster  1-----------------------------------------------
              Bit_Diam  1.0000000  0.5806440  0.0000000
Cluster  1-----------------------------------------------
                 Width  1.0000000  0.3956410  0.0000000
Cluster  1-----------------------------------------------
                 Girth  1.0000000  0.5329000  0.0000000
Cluster  1-----------------------------------------------
                Weight  1.0000000  0.5806440  0.0000000
Cluster  2-----------------------------------------------
              Arm_Span  0.9405000  0.7519053  0.2398278
               Forearm  0.9405000  0.6937160  0.1942641
```

Cluster 1 will be split.

```
                ********************************
                Cluster Summary for 7 Cluster(s)
                ********************************

                     Cluster  Variation Proportion     Second
   Cluster Members  Variation  Explained  Explained Eigenvalue
         1       1  1.0000000  1.0000000  1.0000000          .
         2       1  1.0000000  1.0000000  1.0000000          .
         3       1  1.0000000  1.0000000  1.0000000          .
         4       1  1.0000000  1.0000000  1.0000000          .
         5       1  1.0000000  1.0000000  1.0000000          .
         6       2  2.0000000  1.8810000  0.9405000  0.1190000
         7       1  1.0000000  1.0000000  1.0000000          .

    Total Variation Explained = 7.881 Proportion = 0.9851

                         **************
                         R-squared with
                         **************

                              Own       Next     1-R**2
              Variable    Cluster    Closest      Ratio
Cluster  1-----------------------------------------------
               Low_Leg  1.0000000  0.7378810  0.0000000
Cluster  1-----------------------------------------------
              Bit_Diam  1.0000000  0.5806440  0.0000000
```

```
Cluster  1--------------------------------------------
               Width  1.0000000  0.3956410  0.0000000
Cluster  1--------------------------------------------
               Girth  1.0000000  0.5329000  0.0000000
Cluster  1--------------------------------------------
              Weight  1.0000000  0.5806440  0.0000000
Cluster  2--------------------------------------------
            Arm_Span  0.9405000  0.7157160  0.2092977
             Forearm  0.9405000  0.6480250  0.1690461
Cluster  1--------------------------------------------
              Height  1.0000000  0.7378810  0.0000000
```

Cluster 6 will be split.

```
                 ********************************
                 Cluster Summary for 8 Cluster(s)
                 ********************************

                      Cluster  Variation Proportion     Second
    Cluster Members  Variation  Explained  Explained Eigenvalue
          1       1  1.0000000  1.0000000  1.0000000          .
          2       1  1.0000000  1.0000000  1.0000000          .
          3       1  1.0000000  1.0000000  1.0000000          .
          4       1  1.0000000  1.0000000  1.0000000          .
          5       1  1.0000000  1.0000000  1.0000000          .
          6       1  1.0000000  1.0000000  1.0000000          .
          7       1  1.0000000  1.0000000  1.0000000          .
          8       1  1.0000000  1.0000000  1.0000000          .

        Total Variation Explained = 8 Proportion = 1.0000

                          **************
                          R-squared with
                          **************

                                Own       Next     1-R**2
               Variable    Cluster    Closest      Ratio
Cluster  1--------------------------------------------
             Low_Leg  1.0000000  0.7378810  0.0000000
Cluster  1--------------------------------------------
            Bit_Diam  1.0000000  0.5806440  0.0000000
Cluster  1--------------------------------------------
               Width  1.0000000  0.3956410  0.0000000
Cluster  1--------------------------------------------
               Girth  1.0000000  0.5329000  0.0000000
```

```
Cluster  1--------------------------------------------
               Weight  1.0000000  0.5806440  0.0000000
Cluster  1--------------------------------------------
              Forearm  1.0000000  0.7761610  0.0000000
Cluster  1--------------------------------------------
               Height  1.0000000  0.7378810  0.0000000
Cluster  1--------------------------------------------
             Arm_Span  1.0000000  0.7761610  0.0000000


        ******************************************
        Oblique Principal Component Cluster Analysis
        ******************************************

                 Total   Proportion      Minimum      Maximum
  Number     Variation of Variation   Proportion       Second
    of        Explained    Explained    Explained   Eigenvalue
 Clusters   by Clusters by a Cluster by a Cluster in a Cluster

        1  4.672879598  0.584109950  0.584109950  1.770982845
        2  6.426502368  0.803312796  0.729321030  0.476417740
        3  6.895347443  0.861918430  0.795376398  0.418368751
        4  7.271218248  0.908902281  0.877304562  0.238000000
        5  7.509218248  0.938652281  0.877304562  0.236135269
        6  7.740000000  0.967500000  0.929500000  0.141000000
        7  7.881000000  0.985125000  0.940500000  0.119000000
        8  8.000000000  1.000000000  1.000000000  0.000000000


                         Minimum      Maximum
             Number        R**2        1-R**2
               of         for a        for a
            Clusters    Variable     Variable

                   1  0.380959614        .
                   2  0.632935494  0.437998113
                   3  0.742143965  0.363444211
                   4  0.865164871  0.254763434
                   5  0.865164871  0.166482853
                   6  0.929500000  0.255955661
                   7  0.940500000  0.209297745
                   8  1.000000000  0.000000000
```

4. NIR Spectra Data: Nobs=21, nvar=268:
   For this example specifying the `"trans"` option does not penalize the computer time. It takes the same 1 second for both runs.

```
  options NOECHO;
#include "..\\tdata\\nir.dat"
  options ECHO;
  nr = nrow(xtrn); nc = ncol(xtrn);
  print "nrtrn,nctrn=",nr,nc;


  /* evtech=tql: this is transposed */
  optn = [ "print"        1 ,
           "anal"     "corr" ,
           "trans"           ,
           "evtech"    "arp" ,
           "data"      "raw" ];
  < gof,est,tree,scor,struc,trac > = varclus(xtrn,optn);


          *******************************************
          Oblique Principal Component Cluster Analysis
          *******************************************

          Corrected Correlation Matrix is Analysed
    Number of Subjects. . . . . . . . . . . . . . . .        21
    Number of Variables . . . . . . . . . . . . . . .       268
    Specified Minimum Number of Clusters. . . . . . .         1
    Specified Maximum Number of Clusters. . . . . . .       268
    Proportion. . . . . . . . . . . . . . . . . . . .         .
    Maximum Second Eigenvalue . . . . . . . . . . . . 1.0000000
    Maximum Number of Iterations. . . . . . . . . . .        10
    Eigenvalue Technique. . . . . . . . . . . . . . .    ARPACK


          *******************************************
          Oblique Principal Component Cluster Analysis
          *******************************************

                  Total   Proportion      Minimum      Maximum
      Number    Variation of Variation   Proportion       Second
        of      Explained    Explained    Explained   Eigenvalue
    Clusters  by Clusters by a Cluster by a Cluster in a Cluster

           1  140.6798269  0.524924727  0.524924727  77.97269091
           2  197.1539608  0.735649108  0.655252980  27.30910193
           3  224.5109621  0.837727471  0.790859518  14.83773032
           4  236.0718689  0.880865182  0.837906881  5.981293330
           5  241.9468982  0.902786933  0.865131001  5.558335287
           6  246.3771444  0.919317703  0.867927299  2.946958870
           7  248.8313813  0.928475303  0.867927299  2.795983286
```

```
 8  250.6428055  0.935234349  0.867927299  2.438684932
 9  252.3300480  0.941530030  0.868010514  1.651270561
10  253.8438472  0.947178534  0.868010514  1.865551691
11  255.2790275  0.952533685  0.875944498  1.131011847
12  256.4001153  0.956716848  0.885537474  1.117669924
13  257.4980587  0.960813652  0.885537474  0.978549381

                 Minimum      Maximum
       Number       R**2       1-R**2
         of        for a        for a
     Clusters    Variable     Variable

            1  0.000123285       .
            2  0.007861354  0.994890817
            3  0.272809473  0.991415199
            4  0.369370290  0.992490773
            5  0.374522166  0.982948447
            6  0.374522166  0.990830055
            7  0.421451339  0.959709367
            8  0.421451339  0.958019162
            9  0.421451339  0.997297711
           10  0.407269181  0.997297711
           11  0.590797219  0.997297711
           12  0.590797219  0.995073858
           13  0.590797219  0.995073858

       Total Computation Time: 1
```

```
/* evtech=tql: this is transposed */
optn = [ "print"         1 ,
         "anal"     "corr" ,
         "trans"           ,
         "data"      "raw" ];
< gof,est,tree,scor,struc,trac > = varclus(xtrn,optn);
```

```
          ********************************************
          Oblique Principal Component Cluster Analysis
          ********************************************

          Corrected Correlation Matrix is Analysed
  Number of Subjects. . . . . . . . . . . . . . . . .       21
  Number of Variables . . . . . . . . . . . . . . . .      268
  Specified Minimum Number of Clusters. . . . . . .          1
```

```
Specified Maximum Number of Clusters. . . . . . .       268
Proportion. . . . . . . . . . . . . . . . . . . .         .
Maximum Second Eigenvalue . . . . . . . . . . . . 1.0000000
Maximum Number of Iterations. . . . . . . . . . .        10
Eigenvalue Technique. . . . . . . . . . . . . . .       TQL
```

```
********************************************
Oblique Principal Component Cluster Analysis
********************************************
```

Number of Clusters	Total Variation Explained by Clusters	Proportion of Variation Explained by a Cluster	Minimum Proportion Explained by a Cluster	Maximum Second Eigenvalue in a Cluster
1	140.6798269	0.524924727	0.524924727	77.97269091
2	197.1539608	0.735649108	0.655252980	27.30910193
3	224.5109621	0.837727471	0.790859518	14.83773032
4	236.0718689	0.880865182	0.837906881	5.981293330
5	241.9468982	0.902786933	0.865131001	5.558335287
6	246.3771444	0.919317703	0.867927299	2.946958870
7	248.8313813	0.928475303	0.867927299	2.795983286
8	250.6428055	0.935234349	0.867927299	2.438684932
9	252.3300480	0.941530030	0.868010514	1.651270561
10	253.8438472	0.947178534	0.868010514	1.865551691
11	255.2790275	0.952533685	0.875944498	1.131011847
12	256.4001153	0.956716848	0.885537474	1.117669924
13	257.4980587	0.960813652	0.885537474	0.978549381

Number of Clusters	Minimum R**2 for a Variable	Maximum 1-R**2 for a Variable
1	0.000123290	.
2	0.007861384	0.994890785
3	0.272809437	0.991415278
4	0.369370290	0.992490868
5	0.374522165	0.982948439
6	0.374522165	0.990830047
7	0.421451335	0.959709370
8	0.421451335	0.958019176
9	0.421451335	0.997297686
10	0.407269179	0.997297686
11	0.590797238	0.997297686

```
12  0.590797238  0.995073766
13  0.590797238  0.995073766


Total Computation Time: 1
```

For typical microarray data, where there are about 25000 variables and 200 observations, we cannot store the correlation or covariance matrices and must use the "trans" option. This is not so for $Nobs \gg nvar$: For example, specifying the "trans" option with the Australian data (Nobs=690, nvar=14) requires 69 seconds compared to 0 seconds without the "trans" option.

4 Illustration

4.1 Fast Computation of Singular Vectors for Large Matrices

1. First we illustrate that we only need to compute one set of singular values. Using matrix multiplication we are able to compute the other set.

   Matrix **A** is random 6×3: We compute the original singular values q and the left and right singular vectors u and v:

```
srand(3);
a = rand(6,3);
print a;
< q,v,u > = svd(a,"eco");
print "Sigma=",q;
print "U=",u;
print "V=",v;
```

```
  |         1         2         3
---------------------------------
1 |   0.55489   0.91027   0.19947
2 |   0.77820   0.76481   0.90783
3 |   0.62894   0.59508   0.14938
4 |   0.19970   0.45796   0.57161
5 |   0.56018   0.20155   0.87107
6 |   0.89158   0.81784   0.06979
```

Using one set of vectors u resp. v we compute the other set $v2$ resp. $u2$:

```
at = a‘;
qi = inv(q);
u2 = a * v * qi;
print "U2=",u2;
print "SSQ(U2)=",ssq(u-u2);
v2 = at * u * qi;
print "V2=",v2;
print "U2=",u2;
print "SSQ(V2)=",ssq(v-v2);
```

```
SSQ(U2)= 1.739e-030
SSQ(V2)= 2.360e-030
```

2. Now we illustrate that we can compute the singular values and vectors of the "small" matrix $\mathbf{A}^T\mathbf{A}$. The new singular values `q3` are the squared values of the values `q` of the original **A** matrix.

```
ata = a‘ * a;
< q3,v3,un > = svd(ata,"eco");
print "Q3=",q3;
print "V3=",v3;
print "SSQ(V3)=",ssq(v-v3);
/* Now large U can be computed from v3: */
u3 =  a * v3 * inv(sqrt(q3));
print "U3=",u3;
print "SSQ(U3)=",ssq(u-u3);
```

```
SSQ(V3)= 5.577e-031
SSQ(U3)= 2.773e-030
```

The large singular vectors **U3** are now computed using from the small vectors **V3** using the matrix multiplication method illustrated above.

3. Actually the eigenvalues and eigenvectors of the symmetric p.d. matrix $\mathbf{A}^T\mathbf{A}$ agree with its singular values and vectors :

```
< eval,evec > = eig(ata);
print "EVAL=", eval;
print "EVEC=", evec;
```

That means, we could actually compute the svd of a large matrix $\mathbf{A}$ from the eigenvalue decomposition of whatever is the smaller matrix $\mathbf{A}^T\mathbf{A}$ or $\mathbf{A}\mathbf{A}^T$ and obtain the larger set opf singular vectors simply using matrix multiplication. This fact goes back to papers of Peter Schoenemann and is also used in software like svdpack (Berry, M.B. & Liang, M., 1992) see subroutine `svdtrip`.