# CMAT© Newsletter: July 2011

Wolfgang M. Hartmann

July 2011

## Contents

# 1 General Remarks

At the end of last year the `sem()` like functionality was splitted off the old `factor()` function for a new function `cfa()`. The `factor()` kept only the more traditional estimation methods which are somehow related to the SAS PROC FACTOR.

The new `cfa()` function is able to do exact analysis of correlation matrices, where the diagonal is considered fixed and not subject to fit. It can also analyze ordinal data using polychoric correlations. In addition we now have added some automatic model improvement algorithm for the CFA model $\mathbf{C} = \mathbf{LPL}^T + diag(\mathbf{U})$. In some large scale applications (i.e. many variables and more than four factors) it turned out very difficult to find zero patterns in the loading matrix $\mathbf{L}$ and the factor correlation matrix $\mathbf{P}$ resulting in model fits with a $p > 0.05$ value for the either normal or robust $\chi^2$. This new feature of the `cfa()` function shows to be very helpful in automatically generating such zero patterns with large $p$ values.

The new `som()` function is based on a Matlab toolbox developed by Ballabio & Vasighi (2011), see also Ballabio et al. (2009), and Ballabio et al. (2011), adding another tool for exploratory data analysis (Kohonen maps) and predictive modeling (CP-ANN, Supervised Kohonen Nets, XY-Fused networks).

My work on survival regression and survival curves was continued. Especially, we now have added examples for the GLIM estimation. We also hope to finish up the survival forest algorithm by Breiman and Cutler. Our algorithms compare well with those by T. Therneau's work of the `survival` package in R, especially for speed, the amount of allocated memory, and the robustness of the optimization.

## 1.1 New Functions

**bidimreg** this function implements bidimensional regression between two-dimensional configurations by Tobler (1994).

**cfa** this function implements a new set of algorithms for exploratory and confirmatory factor analysis which were splitted off the older `factor()` function which has additional features like the automatic model selection algorithm which tries to find zero patterns in loadings and factor crrelations with the aim of maximizing the $p$ value of the $\chi^2$ model fit value. (cfa)

**deg2rad** converts degrees into radians (deg2rad)

**invupd** updating the inverse of a positive definite (i.e. necessarily symmetric) matrix (invupd)

**rad2deg** converts radians to degrees (rad2deg2)

**som** Self Organized Maps (Ballabio & Vasighi, 2011):

- Unsupervised Kohonen maps

- Counterpropagation ANNs
- Supervised Kohonen networks
- XY-fused networks

(som)

**smp** implements the *stochastic matching pursuit* algorithm for variable selection with the linear model (see Chen, Chu, & Weng, 2010), as a side result this function also implements the componentwise Gibbs sampler for variable selection (smp)

## 1.2 Fixed Bugs

A number of bugs were fixed.

## 2 Modifications of Features

# 3 Extensions to the Language

# 4 Extensions to Various Functions

## 4.1 Extension of the `...()` Function

# 5 New Developments

## 5.1 Function `bidimreg`

< gof,coef,covm,pred> = bidimreg(type,y,x,<,optn>)

**Purpose:** The `bidimreg()` function implements bidimensional regression between two-dimensional configurations by Tobler (1994). Additional relevant papers are those by Nakaya (1997) and Friedman and Kohler (2003). Tobler used the method to compare the distortions and projections of ancient maps. The implementation is similar to that by Carbon (2011) but was very much extended based on Nakaya (1997). Especially the estimation of multiple models of the same data and comparison of those models was added. The method was originally designed to compare two maps (configurations) of points which were differently shifted, scaled, and/or rotated. Friedman and Kohler (2003) recommended the method being used in cognitive mapping which is used in psychology.

**Input: type** specifies the estimation type; this can be

- an int or string scalar:

  **0="nul"** Null model; this model has only two estimates $\alpha_1$ and $\alpha_2$

  **1="euc"** Euclidean regression: with same scales for the two dimensions; this model has four estimates $\alpha_1, \alpha_2, \beta_1, \beta_2$

  **2="aff"** affine regression: scales both dimensions independently you may also specify: rotation angle, scales, and translations of the two dimensions; this model has four estimates $\alpha_1, \alpha_2, \beta_1, ..., \beta_4$

  **3="cur"** curvilinear model w.r.t. degree $K = 0, 1, 2, 3, 4, 5$, for $K = 0$ the model is equivalent to the Null model, for $K = 1$ the model is equivalent to the Euclidean model this model has $2(K + 1)$ estimates: $(\alpha_0, \beta_0), (\alpha_1, \beta_1), ..., (\alpha_K, \beta_K)$

  **4="emp"** Euclidean $\mu - \Phi$ regression, see Nakaya (1997)

  an int or string $t$ vector: referring to $t$ models, if the affine and curvilinear models are among them, $K$, `angle`, `scale1`, and `scale2` can be specified with the options vector

  a $t \times q$ matrix, where $q = 2, 3, 4$: the first solumn always contains the type of estimation, the second column contains `angle` for the affine, and $K$ for the curvilinear model, columns 3 and four may contain `scale1` and `scale2` for the affine model.

**y** must be $N \times 2$ matrix of the dependent configuration

**x** must be $N \times 2$ matrix of the independent configuration

**optn** must be a scalar or vector of options:

1. amount of printed output (=0: no output)
2. $\alpha$: the probability for the $t$ test, $\alpha = .05$ is default
3. $K$: for curvilinear regression only: specifying the degree of the polynomial; note, that we restrict the degree to a maximum of five since we do not think it makes much sense to use higher degrees; default is $K = 2$
4. `angle`: for affine regression only: fixing the rotation angle
5. `scale1`: for affine regression only: scaling first dimension
6. `scale2`: for affine regression only: scaling second dimension

**Output:** **gof** is either a vector or a list of $t$ vectors of goodness-of-fit measures:

1. return code (=0: no problems)
2. the $r^2$ value (this is the squared of the correlation)
3. Akaike's information criterion DAIC (Nakaya, 1997)
4. Distortion $DI_Y^2$ (Friedman & Kohler, 2003)
5. Distortion $DI_X^2$
6. Max Distortion $DMAX_Y^2$ (Friedman & Kohler, 2003)
7. Max Distortion $DMAX_X^2$
8. Scale $\Phi = \sqrt{\beta_1^2 + \beta_2^2}$
9. Angle in degrees: $\Theta = arctan(\beta_2/\beta_1)$
10. Angle in radians
11. Shearing coefficient (=0 for Euclidean)
12. goodness-of-fit $F$ value
13. probablity $p$ of $F(df1, df2)$
14. df1
15. df2

**covm** is either one $p \times p$ covariance matrix of estimates or a list of $t$ of those matrices

**coef** is either a $nrc \times 2$ matrix (or a list of $t$ matrices when specifying $t > 2$ models) of coefficients and scale parameters where the columns correspond with the two dimensions and the rows contain:

1. the estimate of the pair of parameters
2. its asymptotic standard error (sqare roots of the diagonal entries of the covariance matrix of estimates)
3. the $t$ value (Nakaya, (33))
4. the probability of the $t$ value w.r.t. $\alpha$
5. the lower confidence interval
6. the upper confidence interval

**pred** is a $N \times 4$ matrix of predicted values for the two-dimensional $y$ and $x$

**Restrictions:** 1. Strings, missing values, and complex data are not permitted as input of the first two arguments.

2. The first two input arguments must be matrices of the same size.

**Relationships:** mds(), procrust(), frotate()

**Examples:** 1. Example 1 by Friedman & Kohler(2003):

The following are the data with $N = 4$:

```
y = [12 19 21 18, 16 19 18 14]';
x = [0 10 8 6, 0 0 5 5]';
```

Due to the small degrees of freedom, only the Euclidean method with $p = 4$ parameters and not the affine method (with $p = 6$ parameters) makes sense to compute:

```
optn = 3;
< gof,coef,covm,pred > = bidimreg("aff",y,x,optn);
```

```
*********************************
Bidimensional Regression (Nobs=4)
*********************************

                Mean     Variance
     Y1  17.5000000  15.0000000
     X1  6.00000000  18.6666667
     Y2  16.7500000  4.91666667
     X2  2.50000000  8.33333333
Cov(Y1,X1)=       15.33 Cov(Y1,X2)=       6.667
Cov(Y2,X1)=       5.333 Cov(Y2,X2)=        -2.5
          Covariance(X1,X2)=3.33333
```

The results agree with those reported in table 2 by Friedman and Kohler (2003, p.480):

```
Euclidean Model: npar=4 df1=2 df2=4
-----------------------------------

Fval=0.896753 df1=2 df2=4 p=0.476691
Critical F(alpha=0.05,df1,df2)=6.94427
    Rsquared=0.309572 DAIC=1.03645
 Scaling Mu=0.477867 Rotation Phi=-0.103525
Distortion_1=0.690428 Distortion_2=0.509297
     MaxDistort_1=59.75 MaxDistort_2=81
Angle(deg)=-5.93153 Angle(rad)=-0.103525 Shear=0
```

```
  Parm    Estimate  AsymStdErr      CI_Low      CI_Upp
Alfa_1  14.5246914  2.82095220  6.69247244  22.3569103
Alfa_2  15.8580247  2.82095220  8.02580577  23.6902436
Beta_1  0.47530864  0.35682536 -0.51539739  1.46601468
Beta_2 -0.04938272  0.35682536 -1.04008875  0.94132332

  Parm    Estimate      Tvalue       Tprob
Alfa_1  14.5246914  5.14886121  0.99662512
Alfa_2  15.8580247  5.62151486  0.99753836
Beta_1  0.47530864  1.33204836  0.87316359
Beta_2 -0.04938272 -0.13839463  0.44830806

Scale_1=      0.4779 Scale_2=      0.4779

 Covariance Matrix of Parameter Estimates

            Alfa_1       Alfa_2       Beta_1       Beta_2
Alfa_1   0.771604938  6.6100e-017 -0.074074074  0.030864198
Alfa_2   6.6100e-017  0.771604938 -0.030864198 -0.074074074
Beta_1  -0.074074074 -0.030864198  0.012345679  4.3368e-019
Beta_2   0.030864198 -0.074074074  4.3368e-019  0.012345679

N       Y1dat       Y1prd       Y2dat       Y2prd

1  12.0000000  14.5246914  16.0000000  15.8580247
2  19.0000000  19.2777778  19.0000000  15.3641975
3  21.0000000  18.5740741  18.0000000  17.8395062
4  18.0000000  17.6234568  14.0000000  17.9382716
     Sum of Squared Residuals=41.2531

N       X1dat       X1prd       X2dat       X2prd

1           0 -5.22426164           0 -0.25045887
2  10.0000000  9.99449358           0  7.50759219
3  8.00000000  13.9410979  5.00000000  5.85866845
4  6.00000000  6.83180377  5.00000000 -3.11580177
     Sum of Squared Residuals=186.312
```

```
print "GOF=",gof;
print "COEF=",coef;
print "COVM=",covm;
print "Pred=",pred;
```

GOF=

| | 1 |
|---|---|
| ReturnCode | 0.00000 |
| Rsquare | 0.30957 |
| DAIC | 1.0365 |
| SquDist1 | 0.50930 |
| SquDist2 | 0.69043 |
| MaxDist1 | 59.750 |
| MaxDist2 | 81.000 |
| Phi | -0.10352 |
| Angle(Deg) | -5.9315 |
| Angle(Rad) | -0.10352 |
| Shearing | 0.00000 |
| Fvalue | 0.89675 |
| Pvalue | 0.47669 |
| DF1 | 2.0000 |
| DF2 | 4.0000 |
| unused | . |

COEF=

| | Dim_1 | Dim_2 |
|---|---|---|
| Alpha | 14.525 | 15.858 |
| Alph_ASE | 2.8210 | 2.8210 |
| Alph_T | 5.1489 | 5.6215 |
| Alph_PRB | 0.99663 | 0.99754 |
| Alph_LOW | 6.6925 | 8.0258 |
| Alph_UPP | 22.357 | 23.690 |
| Beta12 | 0.47531 | -0.0494 |
| Bet12_ASE | 0.35683 | 0.35683 |
| Bet12_T | 1.3320 | -0.13839 |
| Bet12_PRB | 0.87316 | 0.44831 |
| Bet12_LOW | -0.51540 | -1.0401 |
| Bet12_UPP | 1.4660 | 0.94132 |
| Beta34 | . | . |
| Bet34_ASE | . | . |
| Bet34_T | . | . |
| Bet34_PRB | . | . |
| Bet34_LOW | . | . |
| Bet34_UPP | . | . |
| Scale12 | 0.47787 | 0.47787 |

Covm=

| SYM | Alfa_1 | Alfa_2 | Beta_1 | Beta_2 |
|---|---|---|---|---|

```
-----------------------------------------------
 Alfa_1 |   0.77160
 Alfa_2 |    7e-017   0.77160
 Beta_1 |  -0.07407  -0.03086   0.01235
 Beta_2 |   0.03086  -0.07407    4e-019   0.01235


Pred=
   |    Y1pred    Y2pred    X1pred    X2pred
-----------------------------------------------
 1 |    14.525    15.858   -5.2243  -0.25046
 2 |    19.278    15.364    9.9945    7.5076
 3 |    18.574    17.840    13.941    5.8587
 4 |    17.623    17.938    6.8318   -3.1158
```

Note, that the data has only $N = 4$ observations and the Euclidean model estimates four parameter estimates. Since the affine model includes sechs parameters, the data will not provide an identified model, i.e. the information matrix would be singular. Our `bidimreg()` function would exit the call with an error message.

2. Example 2 by Friedman & Kohler(2003):
   In this example only the dependent (y) with the independent (x) variables are switched:

```
x = [12 19 21 18, 16 19 18 14]';
y = [0 10 8 6, 0 0 5 5]';


optn = 3;
< gof,coef,covm,pred > = bidimreg("euc",y,x,optn);
```

Even though the parameter estimates and predicted values changed, the goodness-of-fit indices stay mostly the same and the rotation angle changes the sign:

```
                 *********************************
                 Bidimensional Regression (Nobs=4)
                 *********************************

                              Mean     Variance
                      Y1  6.00000000  18.6666667
                      X1  17.5000000  15.0000000
                      Y2  2.50000000  8.33333333
                      X2  16.7500000  4.91666667
          Cov(Y1,X1)=       15.33 Cov(Y1,X2)=       5.333
```

```
Cov(Y2,X1)=       6.667 Cov(Y2,X2)=        -2.5
                Covariance(X1,X2)=3.5


          Euclidean Model: npar=4 df1=2 df2=4
          -----------------------------------


          Fval=0.896753 df1=2 df2=4 p=0.476691
         Critical F(alpha=0.05,df1,df2)=6.94427
             Rsquared=0.309572 DAIC=1.03645
        Scaling Mu=0.64782 Rotation Phi=0.103525
      Distortion_1=0.690428 Distortion_2=0.935978
           MaxDistort_1=81 MaxDistort_2=59.75
     Angle(deg)=5.93153 Angle(rad)=0.103525 Shear=0


  Parm    Estimate  AsymStdErr      CI_Low      CI_Upp
Alfa_1 -4.15481172  11.8661785 -37.1006049  28.7909815
Alfa_2 -9.46443515  11.8661785 -42.4102284  23.4813581
Beta_1  0.64435146  0.48372978 -0.69869772  1.98740065
Beta_2  0.06694561  0.48372978 -1.27610358  1.40999479


  Parm    Estimate      Tvalue       Tprob
Alfa_1 -4.15481172 -0.35013899  0.37194699
Alfa_2 -9.46443515 -0.79759757  0.23488585
Beta_1  0.64435146  1.33204836  0.87316359
Beta_2  0.06694561  0.13839463  0.55169194


Scale_1=      0.6478 Scale_2=      0.6478


 Covariance Matrix of Parameter Estimates

             Alfa_1       Alfa_2       Beta_1       Beta_2
Alfa_1   10.07112971            0 -0.292887029  0.280334728
Alfa_2             0  10.07112971 -0.280334728 -0.292887029
Beta_1  -0.292887029 -0.280334728  0.016736402            0
Beta_2   0.280334728 -0.292887029            0  0.016736402


N       Y1dat       Y1prd       Y2dat       Y2prd

1           0  2.50627615           0  1.64853556
2  10.0000000  6.81589958           0  4.05020921
3  8.00000000  8.17154812  5.00000000  3.53974895
4  6.00000000  6.50627615  5.00000000  0.76150628
```

```
          Sum of Squared Residuals=55.9247

     N       X1dat       X1prd       X2dat       X2prd

     1  12.0000000  4.86943100  16.0000000  13.8686801
     2  19.0000000  20.2231770  19.0000000  12.2734857
     3  21.0000000  16.3548306  18.0000000  20.2693976
     4  18.0000000  13.2840814  14.0000000  20.5884365
          Sum of Squared Residuals=194.505
```

3. Examples by Nakaya(1997):
   The following are the data from Table 2 by Nakaya (1997):

```
y = [0.105 0.88 1.057 1.262 2.316 3.293 -0.743 -0.964
    -1.249 -0.179 0.832 2.217 -0.352 1.434 -0.611 0.497
     2.333 3.13 2.296 ,
     0.515 -0.122 1.109 0.463 -0.037 -0.149 2.092 1.328
     2.322 -0.689 -0.926 -1.092 -1.571 -2.092 -2.908 -2.989
     1.214 1.307 2.232 ]‘;
x = [0.041 0.445 0.773 0.747 1.328 2.007 -0.316 -0.497
    -0.629 -0.124 0.309 0.847 -0.368 0.44 -0.964 -0.379
     1.83 2.184 1.763 ,
     0.536 -0.047 0.83 0.309 -0.687 -1.452 1.457 1.12
     1.632 -0.25 -0.611 -0.959 -0.503 -1.543 -1.707 -1.763
    -0.134 -0.752 0.72 ]‘;
```

   This example has enough data to compute all methods (Null, Euclidean, affine, and curvilinear with $K = 2$):

```
print "Null, Euclidean, Affine, Curvilinear Model";
optn = [   3 ,   /* print */
         .01 ];  /* alpha */
modl = [ "nul"     ,
         "euc"   . ,
         "aff"   . ,
         "cur"   2 ];
< gof,coef,covm,pred,mcmp > = bidimreg(modl,y,x,optn);
```

   The following results agree with those reported in table 2 by Nakaya (1997):

```
                **********************************
                Bidimensional Regression (Nobs=19)
                **********************************
```

```
                    Mean    Variance
            Y1  0.92389474  1.97098165
            X1  0.49668421  0.92551812
            Y2  3.684e-004  2.67795991
            X2 -0.20021053  1.11369273
Cov(Y1,X1)=       1.283 Cov(Y1,X2)=     -0.6249
Cov(Y2,X1)=      0.5287 Cov(Y2,X2)=       1.467
          Covariance(X1,X2)=-0.159756


         Null Model: npar=2 df1=0 df2=36
         -------------------------------

         Fval=  .   df1=0 df2=36 p=  .
                Rsquared=0 DAIC=0
        Distortion_1=1 Distortion_2=  .
     MaxDistort_1=83.6809 MaxDistort_2=  .
  Angle(deg)=  .   Angle(rad)=  .   Shear=  .


  Parm    Estimate  AsymStdErr      CI_Low      CI_Upp
Alfa_1  0.92389474  0.34977217 -0.02730529  1.87509476
Alfa_2  3.684e-004  0.34977217 -0.95083161  0.95156845


     Parm    Estimate      Tvalue       Tprob
   Alfa_1  0.92389474  2.64141870  0.99393133
   Alfa_2  3.684e-004  0.00105332  0.50041731


   Scale_1=      .      Scale_2=      .


    Covariance Matrix of Parameter Estimates

                  Alfa_1       Alfa_2
      Alfa_1   0.052631579            0
      Alfa_2             0  0.052631579


  N       Y1dat       Y1prd       Y2dat       Y2prd

  1  0.10500000  0.92389474  0.51500000  3.684e-004
  2  0.88000000  0.92389474 -0.12200000  3.684e-004
  3  1.05700000  0.92389474  1.10900000  3.684e-004
  4  1.26200000  0.92389474  0.46300000  3.684e-004
  5  2.31600000  0.92389474 -0.03700000  3.684e-004
```

```
 6  3.29300000  0.92389474 -0.14900000  3.684e-004
 7 -0.74300000  0.92389474  2.09200000  3.684e-004
 8 -0.96400000  0.92389474  1.32800000  3.684e-004
 9 -1.24900000  0.92389474  2.32200000  3.684e-004
10 -0.17900000  0.92389474 -0.68900000  3.684e-004
11  0.83200000  0.92389474 -0.92600000  3.684e-004
12  2.21700000  0.92389474 -1.09200000  3.684e-004
13 -0.35200000  0.92389474 -1.57100000  3.684e-004
14  1.43400000  0.92389474 -2.09200000  3.684e-004
15 -0.61100000  0.92389474 -2.90800000  3.684e-004
16  0.49700000  0.92389474 -2.98900000  3.684e-004
17  2.33300000  0.92389474  1.21400000  3.684e-004
18  3.13000000  0.92389474  1.30700000  3.684e-004
19  2.29600000  0.92389474  2.23200000  3.684e-004
      Sum of Squared Residuals=83.6809

     Euclidean Model: npar=4 df1=2 df2=34
     -------------------------------------

    Fval=258.199 df1=2 df2=34 p=2.7773e-021
     Critical F(alpha=0.01,df1,df2)=5.28928
        Rsquared=0.938227 DAIC=-101.803
    Scaling Mu=1.46251 Rotation Phi=0.397153
  Distortion_1=0.0617734 Distortion_2=0.140829
   MaxDistort_1=83.6809 MaxDistort_2=36.7058
 Angle(deg)=22.7552 Angle(rad)=0.397153 Shear=0

  Parm    Estimate  AsymStdErr      CI_Low      CI_Upp
Alfa_1  0.14076914  0.09586342 -0.12078407  0.40232236
Alfa_2 -0.01058239  0.09586342 -0.27213560  0.25097083
Beta_1  1.34867975  0.06435869  1.17308386  1.52427564
Beta_2  0.56569281  0.06435869  0.39009692  0.74128870

    Parm    Estimate      Tvalue       Tprob
  Alfa_1  0.14076914  1.46843441  0.92441100
  Alfa_2 -0.01058239 -0.11039023  0.45637447
  Beta_1  1.34867975  20.9556740  1.00000000
  Beta_2  0.56569281  8.78968799  1.00000000

  Scale_1=       1.463 Scale_2=       1.463

   Covariance Matrix of Parameter Estimates
```

| | Alfa_1 | Alfa_2 | Beta_1 | Beta_2 |
|---|---|---|---|---|
| Alfa_1 | 0.060444499 | 0 | -0.013531493 | -0.005454466 |
| Alfa_2 | 0 | 0.060444499 | 0.005454466 | -0.013531493 |
| Beta_1 | -0.013531493 | 0.005454466 | 0.027243654 | -2.1684e-019 |
| Beta_2 | -0.005454466 | -0.013531493 | -2.1684e-019 | 0.027243654 |

| N | Y1dat | Y1prd | Y2dat | Y2prd |
|---|---|---|---|---|
| 1 | 0.10500000 | -0.10714634 | 0.51500000 | 0.73550337 |
| 2 | 0.88000000 | 0.76751919 | -0.12200000 | 0.17776297 |
| 3 | 1.05700000 | 0.71377356 | 1.10900000 | 1.54610235 |
| 4 | 1.26200000 | 0.97343384 | 0.46300000 | 0.82873219 |
| 5 | 2.31600000 | 2.32044682 | -0.03700000 | -0.18588532 |
| 6 | 3.29300000 | 3.66895537 | -0.14900000 | -0.83351991 |
| 7 | -0.74300000 | -1.10962809 | 2.09200000 | 1.77568508 |
| 8 | -0.96400000 | -1.16310065 | 1.32800000 | 1.21878961 |
| 9 | -1.24900000 | -1.63076109 | 2.32200000 | 1.83464219 |
| 10 | -0.17900000 | 0.11495606 | -0.68900000 | -0.41789823 |
| 11 | 0.83200000 | 0.90314950 | -0.92600000 | -0.65982663 |
| 12 | 2.21700000 | 1.82560030 | -1.09200000 | -0.82482445 |
| 13 | -0.35200000 | -0.07100152 | -1.57100000 | -0.89714326 |
| 14 | 1.43400000 | 1.60705225 | -2.09200000 | -1.84269040 |
| 15 | -0.61100000 | -0.19372050 | -2.90800000 | -2.85810659 |
| 16 | 0.49700000 | 0.62693595 | -2.98900000 | -2.60270236 |
| 17 | 2.33300000 | 2.68465593 | 1.21400000 | 0.84391238 |
| 18 | 3.13000000 | 3.51168672 | 1.30700000 | 0.21068355 |
| 19 | 2.29600000 | 2.11119272 | 2.23200000 | 1.95778347 |

Sum of Squared Residuals=5.16926

| N | X1dat | X1prd | X2dat | X2prd |
|---|---|---|---|---|
| 1 | 0.04100000 | -0.16155592 | 0.53600000 | 0.34085800 |
| 2 | 0.44500000 | 0.49557778 | -0.04700000 | -0.26575912 |
| 3 | 0.77300000 | 0.28161648 | 0.83000000 | 0.46361766 |
| 4 | 0.74700000 | 0.58172555 | 0.30900000 | 0.00207520 |
| 5 | 1.32800000 | 1.37854572 | -0.68700000 | -0.59194655 |
| 6 | 2.00700000 | 2.02419927 | -1.45200000 | -0.92095635 |
| 7 | -0.31600000 | -1.11332307 | 1.45700000 | 1.55948445 |
| 8 | -0.49700000 | -1.05061409 | 1.12000000 | 1.13620427 |
| 9 | -0.62900000 | -1.49320247 | 1.63200000 | 1.83833070 |
| 10 | -0.12400000 | -0.02220262 | -0.25000000 | -0.34319576 |
| 11 | 0.30900000 | 0.67794821 | -0.61100000 | -0.76001449 |
| 12 | 0.84700000 | 1.59514153 | -0.95900000 | -1.23097807 |
| 13 | -0.36800000 | 0.10197982 | -0.50300000 | -0.85357375 |
| 14 | 0.44000000 | 1.36590547 | -1.54300000 | -1.65443080 |

```
15 -0.96400000  0.29227137 -1.70700000 -1.62810047
16 -0.37900000  1.01232633 -1.76300000 -1.97220962
17  1.83000000  1.05840938 -0.13400000  0.19235658
18  2.18400000  1.53634973 -0.75200000  0.04021153
19  1.76300000  0.76584631  0.72000000  0.84402660
      Sum of Squared Residuals=11.0102


      Affine Model: npar=6 df1=4 df2=32
      ---------------------------------

    Fval=196.124 df1=4 df2=32 p=5.07278e-022
     Critical F(alpha=0.01,df1,df2)=3.96948
        Rsquared=0.960808 DAIC=-115.093
 Distortion_1=0.0391918 Distortion_2=0.0893484
   MaxDistort_1=83.6809 MaxDistort_2=36.7058
Angle(deg)=31.7752 Angle(rad)=0.554583 Shear=0.310741


  Parm    Estimate  AsymStdErr      CI_Low      CI_Upp
Alfa_1  0.19288718  0.08354908 -0.03591043  0.42168478
Alfa_2 -0.11906778  0.08354908 -0.34786538  0.10972982
Beta_1  1.32203956  0.07942395  1.10453854  1.53954059
Beta_2 -0.37146591  0.07240378 -0.56974233 -0.17318948
Beta_3  0.81890850  0.07942395  0.60140748  1.03640953
Beta_4  1.43500309  0.07240378  1.23672667  1.63327951


    Parm    Estimate      Tvalue       Tprob
  Alfa_1  0.19288718  2.30866910  0.98621289
  Alfa_2 -0.11906778 -1.42512379  0.08190138
  Beta_1  1.32203956  16.6453508  1.00000000
  Beta_2 -0.37146591 -5.13047639  6.798e-006
  Beta_3  0.81890850  10.3105986  1.00000000
  Beta_4  1.43500309  19.8194485  1.00000000


  Scale_1=       1.416 Scale_2=       1.555


   Covariance Matrix of Parameter Estimates

             Alfa_1       Alfa_2       Beta_1       Beta_2
Alfa_1   0.068110136            0 -0.028803442  0.005855547
Alfa_2             0  0.068110136            0            0
Beta_1  -0.028803442            0  0.061550477  0.008829245
Beta_2   0.005855547            0  0.008829245  0.051150627
Beta_3             0 -0.028803442            0            0
```

| | | | | |
|---|---|---|---|---|
| Beta_4 | 0 | 0.005855547 | 0 | 0 |

Covariance Matrix of Parameter Estimates

| | Beta_3 | Beta_4 |
|---|---|---|
| Alfa_1 | 0 | 0 |
| Alfa_2 | -0.028803442 | 0.005855547 |
| Beta_1 | 0 | 0 |
| Beta_2 | 0 | 0 |
| Beta_3 | 0.061550477 | 0.008829245 |
| Beta_4 | 0.008829245 | 0.051150627 |

| N | Y1dat | Y1prd | Y2dat | Y2prd |
|---|---|---|---|---|
| 1 | 0.10500000 | 0.04798507 | 0.51500000 | 0.68366913 |
| 2 | 0.88000000 | 0.79865368 | -0.12200000 | 0.17790136 |
| 3 | 1.05700000 | 0.90650705 | 1.10900000 | 1.70500106 |
| 4 | 1.26200000 | 1.06566776 | 0.46300000 | 0.93607283 |
| 5 | 2.31600000 | 2.20375279 | -0.03700000 | -0.01740441 |
| 6 | 3.29300000 | 3.38558908 | -0.14900000 | -0.55914290 |
| 7 | -0.74300000 | -0.76610316 | 2.09200000 | 1.71295663 |
| 8 | -0.96400000 | -0.88020830 | 1.32800000 | 1.08113815 |
| 9 | -1.24900000 | -1.24490807 | 2.32200000 | 1.70776381 |
| 10 | -0.17900000 | 0.12182075 | -0.68900000 | -0.57936321 |
| 11 | 0.83200000 | 0.82836307 | -0.92600000 | -0.74281194 |
| 12 | 2.21700000 | 1.66889049 | -1.09200000 | -0.80162024 |
| 13 | -0.35200000 | -0.10677603 | -1.57100000 | -1.14223266 |
| 14 | 1.43400000 | 1.34775648 | -2.09200000 | -1.97295780 |
| 15 | -0.61100000 | -0.44746666 | -2.90800000 | -3.35804585 |
| 16 | 0.49700000 | 0.34672858 | -2.98900000 | -2.95934455 |
| 17 | 2.33300000 | 2.66199601 | 1.21400000 | 1.18724437 |
| 18 | 3.13000000 | 3.35956394 | 1.30700000 | 0.59030607 |
| 19 | 2.29600000 | 2.25618747 | 2.23200000 | 2.35787014 |

Sum of Squared Residuals=3.27961

| N | X1dat | X1prd | X2dat | X2prd |
|---|---|---|---|---|
| 1 | 0.04100000 | 0.05665923 | 0.53600000 | 0.74071576 |
| 2 | 0.44500000 | 0.61495991 | -0.04700000 | -0.06495082 |
| 3 | 0.77300000 | 1.06823373 | 0.83000000 | 1.14700388 |
| 4 | 0.74700000 | 1.03230349 | 0.30900000 | 0.42701711 |
| 5 | 1.32800000 | 1.83520621 | -0.68700000 | -0.94938755 |
| 6 | 2.00700000 | 2.77353830 | -1.45200000 | -2.00656582 |
| 7 | -0.31600000 | -0.43669063 | 1.45700000 | 2.01347549 |

```
 8 -0.49700000 -0.68682040  1.12000000  1.54776427
 9 -0.62900000 -0.86923547  1.63200000  2.25531365
10 -0.12400000 -0.17135962 -0.25000000 -0.34548310
11  0.30900000  0.42701711 -0.61100000 -0.84436069
12  0.84700000  1.17049673 -0.95900000 -1.32527316
13 -0.36800000 -0.50855112 -0.50300000 -0.69511199
14  0.44000000  0.60805025 -1.54300000 -2.13232167
15 -0.96400000 -1.33218282 -1.70700000 -2.35895858
16 -0.37900000 -0.52375237 -1.76300000 -2.43634680
17  1.83000000  2.52893627 -0.13400000 -0.18517894
18  2.18400000  3.01814033 -0.75200000 -1.03921315
19  1.76300000  2.43634680  0.72000000  0.99499132
      Sum of Squared Residuals=6.14919


  Curvilinear Model K=2: npar=6 df1=4 df2=32
  ------------------------------------------


   Fval=156.082 df1=4 df2=32 p=1.65391e-020
    Critical F(alpha=0.01,df1,df2)=3.96948
       Rsquared=0.951244 DAIC=-106.795
    Scaling Mu=1.42883 Rotation Phi=0.345519
  Distortion_1=0.0487561 Distortion_2=  .
   MaxDistort_1=83.6809 MaxDistort_2=  .
Angle(deg)=19.7968 Angle(rad)=0.345519 Shear=0


  Parm    Estimate  AsymStdErr      CI_Low      CI_Upp
Alfa_0  0.19760862  0.08998562 -0.04881534  0.44403258
Beta_0 -1.761e-004  0.08998562 -0.24660006  0.24624787
Alfa_1  1.34438450  0.06525559  1.16568327  1.52308572
Beta_1  0.48392328  0.06525559  0.30522206  0.66262450
Alfa_2  0.05401921  0.03287792 -0.03601637  0.14405479
Beta_2  0.07948084  0.03287792 -0.01055474  0.16951642


    Parm    Estimate      Tvalue       Tprob
  Alfa_0  0.19760862  2.19600213  0.98227188
  Beta_0 -1.761e-004 -0.00195690  0.49922538
  Alfa_1  1.34438450  20.6018289  1.00000000
  Beta_1  0.48392328  7.41581348  0.99999999
  Alfa_2  0.05401921  1.64302382  0.94491512
  Beta_2  0.07948084  2.41745317  0.98924878


  Scale_1=       1.429 Scale_2=       1.429
```

Covariance Matrix of Parameter Estimates

| | Alfa_1 | Alfa_2 | Beta_1 | Beta_2 |
|---|---|---|---|---|
| Alfa_1 | 0.063509791 | 8.9562e-019 | -0.014536780 | -0.009680159 |
| Alfa_2 | 8.9562e-019 | 0.063509791 | 0.009680159 | -0.014536780 |
| Beta_1 | -0.014536780 | 0.009680159 | 0.033398723 | 1.9091e-020 |
| Beta_2 | -0.009680159 | -0.014536780 | 1.9091e-020 | 0.033398723 |
| Beta_3 | 0.003578014 | -0.003631245 | -0.006179333 | -0.003741615 |
| Beta_4 | 0.003631245 | 0.003578014 | 0.003741615 | -0.006179333 |

Covariance Matrix of Parameter Estimates

| | Beta_3 | Beta_4 |
|---|---|---|
| Alfa_1 | 0.003578014 | 0.003631245 |
| Alfa_2 | -0.003631245 | 0.003578014 |
| Beta_1 | -0.006179333 | 0.003741615 |
| Beta_2 | -0.003741615 | -0.006179333 |
| Beta_3 | 0.008478190 | -2.5964e-019 |
| Beta_4 | -2.5964e-019 | 0.008478190 |

| N | Y1dat | Y1prd | Y2dat | Y2prd |
|---|---|---|---|---|
| 1 | 0.10500000 | -0.01858985 | 0.51500000 | 0.71517968 |
| 2 | 0.88000000 | 0.82585726 | -0.12200000 | 0.16980694 |
| 3 | 1.05700000 | 0.93221394 | 1.10900000 | 1.41315712 |
| 4 | 1.26200000 | 1.11400895 | 0.46300000 | 0.78855377 |
| 5 | 2.31600000 | 2.24015209 | -0.03700000 | -0.07989215 |
| 6 | 3.29300000 | 3.23890763 | -0.14900000 | -0.51356375 |
| 7 | -0.74300000 | -1.11476144 | 2.09200000 | 1.69462537 |
| 8 | -0.96400000 | -1.15544744 | 1.32800000 | 1.24509490 |
| 9 | -1.24900000 | -1.72345447 | 2.32200000 | 1.82013083 |
| 10 | -0.17900000 | 0.15426797 | -0.68900000 | -0.40337335 |
| 11 | 0.83200000 | 0.86368005 | -0.92600000 | -0.67374814 |
| 12 | 2.21700000 | 1.66033787 | -1.09200000 | -0.80787786 |
| 13 | -0.35200000 | -0.03063887 | -1.57100000 | -0.88382936 |
| 14 | 1.43400000 | 1.30975570 | -2.09200000 | -1.96213006 |
| 15 | -0.61100000 | -0.11794591 | -2.90800000 | -3.09705880 |
| 16 | 0.49700000 | 0.48731672 | -2.98900000 | -2.86154483 |
| 17 | 2.33300000 | 2.86363234 | 1.21400000 | 0.99649538 |
| 18 | 3.13000000 | 3.46369687 | 1.30700000 | 0.55733946 |
| 19 | 2.29600000 | 2.56101058 | 2.23200000 | 1.88963485 |

Sum of Squared Residuals=4.07996

The following results agree with those reported in table 3 by Nakaya

(1997):

Model Comparison: F Test

Dense Lower Triangular Matrix (4 by 4)

| L | Nullmodel | Euclidean | Affine | Curvilin. |
|---|---|---|---|---|
| Nullmodel | . | 0 | 0 | 0 |
| Euclidean | 258.19944 | 258.19944 | 0 | 0 |
| Affine | 196.12445 | 9.2189145 | 196.12445 | 0 |
| Curvilin. | 156.08203 | 4.2718035 | . | 156.08203 |

Diagonal with: F Value

Critical F(alpha=0.01,df1,df2)

Dense Lower Triangular Matrix (4 by 4)

| L | Nullmodel | Euclidean | Affine | Curvilin. |
|---|---|---|---|---|
| Nullmodel | 0 | 0 | 0 | 0 |
| Euclidean | 5.2892769 | 0 | 0 | 0 |
| Affine | 3.9694772 | 5.3363429 | 0 | 0 |
| Curvilin. | 3.9694772 | 5.3363429 | . | 0 |

Model Comparison: AIC

Dense Lower Triangular Matrix (4 by 4)

| L | Nullmodel | Euclidean | Affine | Curvilin. |
|---|---|---|---|---|
| Nullmodel | 0 | 0 | 0 | 0 |
| Euclidean | -101.80274 | 0.9382266 | 0 | 0 |
| Affine | -115.09296 | -13.290211 | 0.9608082 | 0 |
| Curvilin. | -106.79515 | -4.9924040 | 8.2978075 | 0.9512439 |

Diagonal with: Rsquared Value

## 5.2 Function `cfa`

< gof,est,resi,cov,mod1,mod2,best,boci> =

= cfa(data,optn<,patt<,scal<,wgt<,xini<,targ<,wtrg> . >)

**Purpose:** The `cfa()` function implements ULS, ML, GLS, WLS, and DWLS estimation for confirmatory (CFA) and exploratory (EFA) factor analysis. From the estimation methods, WLS and DWLS estimation require either raw data input or the specification of a weight matrix with the fifth input argument. The input data may be a $N \times n$ matrix, $N$ observations, $n$ variables, of raw data or a symmetric $n \times n$ matrix of correlations or covariances. Analyzed is

- the covariance matrix,
- the correlation matrix like a covariance matrix,
- the correlation matrix without fitting the unit diagonal.

For multiple sample analysis the raw data may contain an ID variables specifying the sample number or lists of covariance or correlation matrices.

The `cfa()` function is able to analyze multiple $s \geq 1$ data samples with a simultaneous model. Different parameter patterns and even different factor numbers can be specified for the different samples.

The factor model

$$\mathbf{C} = \mathbf{LPL}^T + diag(\mathbf{U})$$

where $\mathbf{C}$ is a $n \times n$ covariance or correlation matrix, $\mathbf{L}$ is a $ntimesm$ factor loading matrix, and $\mathbf{U} = diag$ is a $n \times n$ diagonal matrix of unique variances. Currently the model is restricted to $diag(\mathbf{U})$ but an extension is intended for specifying CFA patterns for general symmetric $\mathbf{U}$.

If the pattern input argument `patt` is specified as nonmissing, the number $f$ of factors of the CFA is defined, otherwise it must be specified with the `nfact` option.

One of the larger new developments for the `cfa()` function is an algorithm which tries to find a zero pattern in loading matrix $\mathbf{F}$ and factor correlations `P` which corresponds to a large $p$ value of the $\chi^2$ fit criterion. Since the maximization of $p$ is numerically very difficult we cannot guarantee that the function will always find a global maximum, but in many applications it found CFA models with surprisingly large $p$ values.

The CFA model improvement algorithm works within a known number $f$ of factors. There are two versions:

- if the `patt` input argument is specified as missing: the algorithm start with an EFA and computes a CFA pattern which corresponds to a larger $p$ value

- if the `patt` input argument is specified as nonmissing: the algorithm start with that CFA specification ad tries to improve the CFA pattern for a larger $p$ value.

The `cfa()` function is able to analyze ordinal data by computing polychoric correlations. The `scal` input argument should be a binary vector specifying which of the variables of the input raw data are scaled ordinary (set to 1) and which is continuous (set to 0). This input argument may be a missing value indicating the default, that all variables are continuous. If there are variables of different scaling type (ordinal and continuous) the continuous variables are binned to ordinal (by default into 16 bins, but may be specified differently using the `"nbin"` option). (Nominal variables with $k$ levels may always be rescaled to $k-1$ binary dummy variables.)

Except for GLS estimation the `cfa` function is able to do genuine correlation analysis. The difference between the classical covariance analysis (even of correlation matrices) and genuine correlation analysis are:

**covariance analysis** For EFA both, the loadings and unique variances are estimated by fitting the model to the covariance or correlation matrix inluding its diagonal. (For correlation data usually only the degrees of freedom are adjusted.)

**correlation analysis** The diagonal of the correlation matrix is considered fixed to unity and the model is not fitting the diagonal of the matrix. That means, for EFA only the loadings $\mathbf{L}$ are estimated, afterward the unique variances $\mathbf{U}$ are simply set to the difference between the unit diagonal and the communalities $h_i^2 = \sum_{j=1}^{nvar} l_{ij}^2 \quad i = 1, \ldots, nvar$.

Genuine ML correlation analysis is computationally very expensive and may become very messy for Heywood cases, which occurs when the communalities $h_i^2$ become larger than 1. In those cases the asymptotic covariance matrix becomes indefinite and the ML fit criterion may become negative. Note also, that the ML fit criterion for genuine correlation analysis is changed and does not include the constant term $log(det(\mathbf{S}))$, since the polychoric correlation matrix may have negative eigenvalues and therefore a negative determinant. Therefore, for genuine correlation analysis we recommend ULS, DWLS, or WLS estimation. Note, that WLS estimation requires many data observations so that the large $(\binom{n}{2}) \times (\binom{n}{2})$ weight matrix $\mathbf{W}$ (based on fourth order moments) becomes nonsingular.

Except for ULS estimation `cfa()` is able to compute standard normal and S-B robust asymptotic standard errors and confidence intervals.

There are two definitions of the degrees of freedom (using $s$ for the number of samples and $p$ for the number of model parameters):

- analyze covariance matrix: the same for both: $df = s\binom{n}{2} - p$

- analyze correlation as covariance matrix:
  - the "old" way: $df = s(\binom{n}{2} - n) - p$
  - the "new" way: $df = s\binom{n}{2} - p$
- genuine correlation analysis: the same for both: $df = s(\binom{n}{2} - n) - p$

Default is the "new", use the `"dfold"` option for the "old" version.

For exploratory analysis it also implements a number of popular orthogonal and oblique rotation methods. In addition it incorporates robust (nonnormal, Satorra-Bentler) goodness of fit indices, robust ASEs, and nonnormal (robust) standardized residuals for ULS, ML, and GLS estimates.

1. The same rotation methods as in the `frotate` or the older `factor` function are also available in the `factor()` function.
2. Except for ULS, asymptotic standard errors and confidence intervals are 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.

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.

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

1. `nobs` by `nvar` matrix of raw data which can have continuous or ordinal scaled variables; if ordinal variables are accompanied by continuous variables, the continuous variables are binned into (by default) 16 equally sized intervals

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.

**patt** If not set to missing, this argument specifies the parameter structure for the **L**, **P**, and **U** matrices.

**scal** If not set to missing, this argument should be an `nvar` binary vector specifying which of the variables of the input raw data are scaled ordinary (set to 1) and which is continuous (set to 0).

**wgt** This argument specifies the weight matrix for GLS, WLS, or DWLS estimation.

**xini** This argument specifies a starting vector for the parameter estimation.

**targ** This argument specifies the target matrix for Browne's Procrustes rotation method. This is used for both, `"targt1"` and `"targt2"` rotation. It must be of dimension $nvar \times nfact$.

**wtrg** This argument specifies the weight matrix for the target matrix of Browne's Procrustes rotation method. This is used only for `"targt2"` rotation. It must be of dimension $nvar \times nfact$.

**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 chisquare and confidence intervals; default is $\alpha = 0.05$ |
| "alphaecv" | real | the significance level for ECV; default is $\alpha = 0.1$ |
| "alpharms" | real | the significance level for RMSEA; default is $\alpha = 0.1$ |
| "amimp" | | perform automatic CFA model improvement for large $p$ |
| "ana" | string | data type for analysis |
| | "ccor" | correlations are analyzed as correlations (not available for GLS estimation) |
| | "cor" | correlations are analyzed as covariances |
| | "cov" | covariances are analyzed |
| | "ucor" | uncorrected correlations are analyzed |
| | "ucov" | uncorrected covariances are analyzed |
| "asc" | string | type of asymptotic covariances for **W** |
| | "bia" | biased: Browne's (1984) formula (3.4) |
| | "unb" | unbiased: Browne's (1984) formula (3.8) |
| | "cor" | Browne & Shapiro (1986) formula (3.2) |
| "bala" | int | =1: balanced (default) or =0:uniform resampling |
| | string | "y": yes, "n": no (same as in `jboot` and `sem` |
| "bcorr" | | bias correction for skewness |
| "bgof" | | compute bootstrap confidence intervals for goodness of fit values |
| "bin" | | number of bins for making metric to ordinal (only for mixed scaling of variables, def=16) |
| "boci" | | compute bootstrap confidence intervals for unique vars and rotated or unrotated loadings |
| "bost" | | compute Bollen-Stine bootstrap value of $p$ |
| "bseed" | int | seed value for bootstrap random generator (same as in `jboot` and `sem` |
| "btask" | string | bootstrap method |
| | "jack" | Jackknife is performed (fast) |
| | "norm" | normal theory bootstrap (fast) |
| | "hybr" | hybrid boostrap (fast) |
| | "perc" | percentile bootstrap (fast) |
| | "bc" | bias corrected bootstrap (fast) |
| | "bca" | bias accelerated corrected bootstrap |
| | "stud" | studentized bootstrap (only for "boci") |
| | "all" | perform all bootstrap methods (same as in `jboot` and `sem` default is `perc` for `"gbof"` and `"bc"` otherwise |
| "cl" | string | type of confidence intervals |
| | "none" | do not compute |
| | "wald" | Wald linkelihood intervals (fast) |
| | "plik" | profile likelihood intervals (slow) |
| | "boot" | Bootstrap confidence intervals (slow) |
| "comp" | | perform principal component analysis use only for exploratory FACTOR model |

| Option Name | Second Column | Meaning |
|---|---|---|
| "data" | string | type of input data |
| | "raw" | $N \times n$ raw data |
| | "cor" | symmetric correlation matrix |
| | "cov" | symmetric covariance matrix |
| "dfold" | real | use old definition of DFs |
| "evsing" | real | threshhold for identifying zero eigenvalues |
| "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 $\gamma$) |
| | "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 $\gamma$) |
| | "simmax" | orthogonal Simplimax (specify number nonzero loadings with $\gamma$) |
| | "tandm1" | orthogonal Tandem 1 rotation |
| | "tandm2" | orthogonal Tandem 2 rotation |
| | "infmax" | orthogonal Infomax rotation |
| | "mccamm" | orthogonal McCammon rotation |
| | "crafer" | oblique Crawford-Ferguson family (specify $\gamma$) |
| | "oblmin" | oblique Direct Oblimin family (specify $\gamma$) |
| | "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 $\gamma$) |
| | "oblmax" | oblique Oblimax |
| | "geomin" | oblique Geomin |
| | "infmax" | oblique Infomax rotation |
| | "promax" | oblique Promax |
| | "targt1" | Target Rotation (Harman, 1976) |
| | "targt2" | Weighted Target (Browne,2001) |
| "fnorm" | | Kaiser's factor pattern normalization |
| | | use only for exploratory FACTOR model |

| Option Name | Second Column | Meaning |
|---|---|---|
| "hey" | | Heywood case: apply boundary constraints |
| "idv" | int | specifies the column number of an ID variable for multiple sample analysis |
| "met" | string | estimation method |
| | "no" | no estimation, just evaluate model |
| | "ml","max" | maximum likelihood (ML) |
| | "ls","uls" | unweighted least squares (LS) |
| | "gls" | generalized least squares (GLS) |
| | "wls" | weighted least squares (WLS,ADF) |
| | "dwls" | diagonally weighted LS (DWLS, app. ADF) |
| "maxfu" | int | maximum nuber function calls for optimization |
| "maxit" | int | maximum number iterations for optimization |
| "maxrot" | int | maximum number iterations for rotation |
| "nfac" | int | number factors for exploratory FACTOR model |
| "nobs" | real | number observations when COV or CORR data input only for single sample analysis |
| "nopr" | | perform no printed output |
| "nomod" | | do not compute (print) modification indices (does not affect modification indices for model improvement) |
| "nomis" | | skip observations with missing values in raw data |
| "nost2" | | do not run stage=2 for model improvement (for $n > 30$ specifying "nost2" saves computer time) |
| "nost5" | | do not run stage=5 for model improvement (stage=5 is the stochastic search and may be necessary) |
| "nost6" | | do not run stage=6 for model improvement |
| "nst5" | int | number of stochastic runs (stage=5) |
| "obl" | | use oblique rotation |
| "ort" | | use orthogonal rotation |
| "pall" | | large amount of printed output |
| "parboot" | | perform parametric bootstrap |
| "pbest" | int | print ($\leq 20$) best solutions of model improvement |
| "pcov" | | print covariance matrix of parameter estimates |
| "pcpr" | | print information on polychoric correlation |
| "peig" | | print eigenvalue distribution(s) |
| "phis" | | print optimization history |
| "pini" | int | print input information |
| "pmod" | int | print modification indices |
| "pres" | | print raw, standardized, normalized residuals |
| "prin" | int | amount of printed output |
| "pscor" | | print factor scores |
| "psho" | | short amount of printed output |
| "psum" | | summary amount of printed output |
| "pwgt" | | print weight matrices |
| "rand" | string | specifying the random generator ("ranu": RAND Corporation, "kiss": keep-it-simple-stupid, "lecu": by L'Ecuyer ([?]) |
| | 29 | (same as in `jboot` and `sem` |
| "rob" | | robust estimation of ASE and $\chi^2$ |
| "ridge" | real | nonnegative ridge value |
| "rinit" | int | initialization for rotation matrix $\mathbf{T}$ |
| "rotp" | real | rotation parameter ($\gamma$ for Crawford-Ferguson) |
| "rtres" | real | small threshold for setting residuals to zero |
| "rvers" | int | version of algorithm for rotation |
| | string | "comm", "jennr", "prom", "ogas" |

| Option Name | Second Column | Meaning |
|---|---|---|
| "samp" | int | number of bootstrap samples (default=500) |
| | | (same as in jboot and sem |
| "seed" | real | seed value for random generator |
| "sing" | real | singularity threshold (default: 1.e-8) |
| "slmw" | real | probability for multiple Wald modification |
| "tech" | string | optimization technique |
| | "tr" | trust region |
| | "lm" | Levenberg-Marquardt |
| | "cg" | conjugate gradient |
| | "dd" | double dogleg |
| | "nrl" | Newton-Raphson line search |
| | "nrr" | Newton-Raphson ridged |
| | "dqn" | dual Quasi Newton |
| "wgt" | int | column number of WEIGHT variable when raw data input |
| "wridge" | real | ridge factor for weight matrix (GLS, WLS, DWLS) |

Notes:

- You should always specify the "data" and the "ana" string options to make sure the function knows what data are given and what data should be analyzed.
- For the robust WLS estimation the "rob" option is obsolete.

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:** **gof** column vector with goodness-of-fit measures. For specified bootstrap, the vector becomes a matrix.

**parm** vector or matrix with parameter estimates and if the `cl` option is specified specified also with asymptotic standard errors (ASEs) and confidence intervals.

- `cl` is not specified: `parm` contains the $n \times m$ matrix $\mathbf{L}$ of factor loadings.
- `cl` is specified: The first column contains all the parameter estimates, i.e. the entries of $\mathbf{L}$ and $\mathbf{U}$. The following four columns contains normal theory ASE, $t$ value, lower and upper Wald confidence limits. If `rob` is specified, raw data are given, and other than WLS estimates are specified an additional set of four columns contain the robust (nonnormal theory, Satorra-Bentler) ASE, $t$ value, lower and upper confidence limits.

**resi** matrix of predicted values and residuals. There are $n(n+1)/2$ rows corresponding to entries $(i, j), i \geq j$ of the lower triangle of a symmetric matrix and 6 or 7 columns specifying:

1. analyzed CORR or COV data ($\mathbf{S}$)
2. model matrix $\mathbf{\Sigma}$
3. raw residuals $r_{ij} = s_{ij} - \sigma_{ij}$

4. normalized residuals
5. standardized residuals
6. normal theory asymptotically standardized residuals
7. only if raw data are given: robust asymptotically standardized residuals

**cov** the result depends whether robust estimates are computed (raw data are available) or not:

- without robust estimation: `cov` contains the $p \times p$ covariance matrix of parameter estimates.
- with robust estimation: The first column of `cov` contains all $p(p+1)/2$ lower triangular entries of the normal theory covariances; the second columns contains the corresponding entries of the robust covariance matrix.

**mod1** contains the $M \times 5 or 6$ matrix of $M \leq 20$ largest Lagrange multipliers, the columns correspond to:

1. only for $s > 1$: the sample number
2. the matrix number (=1 for **L**, =2 for **P**, =3 for **U**)
3. the row number
4. the column number
5. the expected $\chi^2$ value after the zero in that location is replaced by a new parameter
6. the expected $x$ estimate for that parameter

**mod2** contains $M \times 3$ matrix of $M \leq 20$ smallest Wald indices, the columns correspond to:

1. the parameter
2. the expected $\chi^2$ value after that parameter is zeroed
3. the value of the estimate

**best** is a $21 \times 10$ matrix of the goodness-of-fit table for the 20 best solutions found with automatic improvement of the $p$ value; the last row contains the corresponding indices of the starting solution

**boci** contains bootstrap confidence intervals of (rotated or unrotated) factor loadings, unique variances, and in the case of oblique rotation also factor correlations.

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

2.

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

**Examples:** 1. This example illustrates ML and WLS estimation of polychoric correlation matrix of the lot data with `nvar=8` ordinal variables, `nobs=389`. Note, that the `"anal" "ccor"` option requires genuine *correlation* analysis where the unit diagonal of the correlation

matrix is considered constant and is not fitted. For ML and WLS estimation of applications with many variables this may require too much computer memory.

```
  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);
  print cov = bivar(lot,"cov");
```

(a) ML Estimation:

```
     print "FACTOR: ML Method with Varimax";
     optn = [ "data"      "raw" ,
              "anal"     "ccor" ,
              "meth"       "ml" ,
              "cl"       "wald" ,
              "nfac"          2 ,
              "tech"   "nrridg" ,
              "frot"   "varmax" ,
              "phis"            ,
              "pini"          2 ,
              "pcpr"          3 ,
              "prin"          3 ];
     scal = [ 8#1 ];
     gof = cfa(lot,optn,.,scal);
```

| Variable | Value | Nobs | Proportion | Freq | Weight |
|---|---|---|---|---|---|
| | 0 | 13 | 0.03341902 | 13 | 13.0000000 |
| | 1 | 84 | 0.21593830 | 84 | 84.0000000 |
| | 2 | 125 | 0.32133676 | 125 | 125.000000 |
| | 3 | 129 | 0.33161954 | 129 | 129.000000 |
| | 4 | 38 | 0.09768638 | 38 | 38.0000000 |
| | | | | | |
| | 0 | 12 | 0.03084833 | 12 | 12.0000000 |
| | 1 | 56 | 0.14395887 | 56 | 56.0000000 |
| | 2 | 121 | 0.31105398 | 121 | 121.000000 |
| | 3 | 149 | 0.38303342 | 149 | 149.000000 |
| | 4 | 51 | 0.13110540 | 51 | 51.0000000 |
| | ..... | | | | |
| | 0 | 69 | 0.17737789 | 69 | 69.0000000 |

```
                1     186  0.47814910       186 186.000000
                2      60  0.15424165        60 60.0000000
                3      58  0.14910026        58 58.0000000
                4      16  0.04113111        16 16.0000000

                      **********
                      Thresholds
                      **********

      1  -1.83276143 -0.67651357  0.17814153  1.29484771
      2  -1.86846944 -0.93533747 -0.03544818  1.12118112
      3  -1.90673221 -1.05143443 -0.27068619  0.97603151
      4  -1.60738861 -0.88654975 -0.00322189  1.42470796
      5  -1.44271677 -0.17159847  0.55195841  1.54068570
      6  -0.91556101  0.31102177  0.99699679  1.79924808
      7  -0.86761683  0.40028580  1.07410343  1.90673221
      8  -0.92540400  0.40028580  0.87704399  1.73770831

              Polychoric Correlation(s)

            i1         i4         i5        i11         i3
i1    1.0000000
i4    0.5577219  1.0000000
i5    0.4859851  0.5877967  1.0000000
i11   0.2803616  0.3723243  0.2652058  1.0000000
i3   -0.1636465 -0.2297360 -0.2780625 -0.1216389  1.0000000
i8   -0.3080950 -0.3925849 -0.3718812 -0.1979306  0.5536118
i9   -0.2654568 -0.3005806 -0.3364019 -0.2294538  0.5587520
i12  -0.2585184 -0.3656555 -0.3312995 -0.3016573  0.4891194

              Polychoric Correlation(s)

                    i8         i9        i12
          i8    1.0000000
          i9    0.7644719  1.0000000
          i12   0.5980033  0.5932275  1.0000000
   Matrix is Positive Definite: Determinant=0.0414436

             ****************************
             Correlation ML Factor Analysis
             (Exploratory Factor Analysis)
             ****************************

Input Data. . . . . . . . . . . . . . . . . . . . . . Raw Data
Analysis of . . . . . . . . . . . . . . . .Correlation Matrix
```

```
Number of Items . . . . . .          8
Number of Factors (Unknown Criterion) . . . . . .          2
Number of Subjects. . . . . .        389
Version . . . . . . SEM-Type Matrix Model
Orthogonal Rotation Method. . . . . .Varimax (Kaiser, 1958)
Unnormed Rotation: Parameter. . . . . 1.0000000
ASE for Rotated Solution. . . . . .Wald (Analyt.)
```

**********
Optimization Start
**********

Parameter Estimates
---

| Parameter | Estimate | Gradient | Lower BC | Upper BC |
|---|---|---|---|---|
| 1 L_11 | 0.50000000 | 2.4784761 | -1.0000000 | 1.0000000 |
| 2 L_21 | 0.50000000 | 11.242274 | -1.0000000 | 1.0000000 |
| 3 L_31 | 0.50000000 | 9.8605057 | -1.0000000 | 1.0000000 |
| 4 L_41 | 0.50000000 | 5.2490795 | -1.0000000 | 1.0000000 |
| 5 L_51 | 0.50000000 | 5.1232797 | -1.0000000 | 1.0000000 |
| 6 L_61 | 0.50000000 | 9.4904560 | -1.0000000 | 1.0000000 |
| 7 L_71 | 0.50000000 | 8.8006663 | -1.0000000 | 1.0000000 |
| 8 L_81 | 0.50000000 | 7.3673746 | -1.0000000 | 1.0000000 |
| 9 L_22 | 0.50000000 | 17.102696 | -1.0000000 | 1.0000000 |
| 10 L_32 | 0.50000000 | 15.551800 | -1.0000000 | 1.0000000 |
| 11 L_42 | 0.50000000 | 9.6822390 | -1.0000000 | 1.0000000 |
| 12 L_52 | 0.50000000 | 1.2985569 | -1.0000000 | 1.0000000 |
| 13 L_62 | 0.50000000 | 4.3560267 | -1.0000000 | 1.0000000 |
| 14 L_72 | 0.50000000 | 3.8771678 | -1.0000000 | 1.0000000 |
| 15 L_82 | 0.50000000 | 2.7866733 | -1.0000000 | 1.0000000 |

Value of Objective Function = 15.7358

Newton-Raphson Ridge Optimization
Without Parameter Scaling
Gradient Computed by Finite Differences
Hessian Computed by Finite Differences (dense)
(Using Only Function Calls)

```
Iteration Start:
N. Variables            15
N. Bound. Constr.       30       N. Mask  Constr.         0
```

```
 Criterion       15.73583618        Max Grad Entry  17.10269650
 N. Active Constraints     0


Iter rest nfun act   optcrit  difcrit maxgrad   ridge      rho
   1*   0    2   0  10.77424 4.961599 12.6680 128.000  1.84887
   2*   0    3   0  7.792705 2.981532 9.68714 128.000  1.82836
   3    0    4   0  1.773905 6.018800 2.96104 32.0000  1.60037
   4    0    5   0  0.889259 0.884647 1.29784 8.00000  1.42521
   5    0    6   0  0.444685 0.444573 0.51291 2.00000  1.37957
   6    0    7   0  0.159572 0.285113 0.20548 0.50000  1.46529
   7    0    8   0  0.071906 0.087666 0.13769 0.12500  0.84443
   8    0    9   0  0.056523 0.015383 0.01308 0.06250  1.05262
   9    0   10   0  0.056349 1.7e-004  2e-004       0  1.00790
  10    0   11   0  0.056349 4.8e-008  6e-008       0  1.00076

 Successful Termination After    10 Iterations
 ABSGCONV convergence criterion satisfied.
 Criterion       0.056348924        Max Grad Entry  6.4817e-008
 N. Active Constraints     0        Ridge (lambda)            0
 Act.dF/Pred.dF  1.000756616
 N. Function Calls        12        N. Gradient Calls         6
 N. Hessian Calls         24        Preproces. Time           0
 Time for Method           1        Effective Time            1

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

                        FAC_1                 FAC_2
      i1    0.6448668  0.0506316          0          0
           [ 0.545631, 0.744103] [        0,        0]

      i4    0.8318109  0.0507931  0.0135541  0.0699333
           [ 0.732258, 0.931363] [-0.123513, 0.150621]

      i5    0.6696670  0.0486295 -0.0848160  0.0610709
           [ 0.574355, 0.764979] [-0.204513, 0.034881]

      i11   0.4013681  0.0514606 -0.0608360  0.0567315
           [ 0.300507, 0.502229] [-0.172028, 0.050356]

      i3   -0.2588664  0.0625600  0.5641520  0.0497431
           [-0.381482,-0.136251] [ 0.466657, 0.661647]

      i8   -0.4272568  0.0660078  0.7121729  0.0493700
```

```
     [-0.556630,-0.297884] [ 0.615409, 0.808936]

i9   -0.3494904  0.0687378  0.7706217  0.0469364
     [-0.484214,-0.214767] [ 0.678628, 0.862615]

i12  -0.4005376  0.0592320  0.5276083  0.0511764
     [-0.516630,-0.284445] [ 0.427304, 0.627912]

                          U_Var
          i1    0.5841469          0
               [ 0.584147, 0.584147]

          i4    0.3079070          0
               [ 0.307907, 0.307907]

          i5    0.5443523          0
               [ 0.544352, 0.544352]

          i11   0.8352026          0
               [ 0.835203, 0.835203]

          i3    0.6147208          0
               [ 0.614721, 0.614721]

          i8    0.3102614          0
               [ 0.310261, 0.310261]

          i9    0.2839986          0
               [ 0.283999, 0.283999]

          i12   0.5611991          0
               [ 0.561199, 0.561199]

(1) Standalone Fit Measures: ------------------------------
Fit criterion . . . . . .    0.0000
Normal Th. Chi-square (df = 13)  0.0000 Prob>chi**2 = 1.0000
Bartlett Corrected Chi-square (df = 13)     0.0000 p= 1.0000
Normal Theory Reweighted LS Chi-square  . . . . .   189.6706
Probability of Close Fit  . . . . . .    1.0000
Z-Test of Wilson & Hilferty (1931). . . . . .     .
(2) Incremental Fit Measures: ------------------------------
Null Model Chi-square (df = 28) . . . . . .   -388.0000
RMSEA Estimate  . . . . .  0.0000  90%C.I.[ 0.0000,  0.0000]
ECVI Estimate . . . . . .   0.0792
McDonald's (1989) Centrality. . . . . .    1.0168
```

```
Tucker-Lewis Coefficient TLI. . . . . .     0.9327
Bentler & Bonett's (1980) NFI . . . . . .     1.0000
Bentler's Comparative Fit Index CFI . . . . . .      .
Parsimonious NFI (James, Mulaik, & Brett,1982). .     0.4643
Bollen (1986) Normed Index Rho1 . . . . . .     1.0000
Bollen (1988) Non-normed Index Delta2 . . . . . .     0.9676
(3) Information Criteria: ----------------------------------
Akaike's Information Criterion. . . . . .   -26.0000
Bozdogan's (1987) CAIC. . . . . .   -90.5265
Schwarz's Bayesian Criterion. . . . . .   -77.5265
(4) Other Fit Measures: ------------------------------------
Goodness of Fit Index (GFI) . . . . . .     0.9777
Parsimonious GFI (Mulaik, 1989) . . . . . .     0.4539
GFI Adjusted for Degrees of Freedom (AGFI). . . .     0.9382
Root Mean Square Residual (RMR) . . . . . .     0.0385
Hoelter's (1983) Critical N . . . . . .      .
```

Rotation computed by common estimation algorithm.
ASE and CI computed by common estimation algorithm analytic version.

Rotated Factor Loadings with Standard Errors

| | FAC_1 | | FAC_2 | |
|---|---|---|---|---|
| i1 | 0.6235542 | 0.0422227 | -0.1644182 | 0.0428581 |
| | [ 0.540799, | 0.706309] | [-0.248419, | -0.080418] |
| i4 | 0.8077756 | 0.0399853 | -0.1989762 | 0.0347872 |
| | [ 0.729406, | 0.886145] | [-0.267158, | -0.130795] |
| i5 | 0.6259097 | 0.0413365 | -0.2527542 | 0.0420524 |
| | [ 0.544892, | 0.706928] | [-0.335176, | -0.170333] |
| i11 | 0.3725920 | 0.0511884 | -0.1611601 | 0.0509114 |
| | [ 0.272265, | 0.472919] | [-0.260945, | -0.061376] |
| i3 | -0.1064721 | 0.0457239 | 0.6115087 | 0.0374094 |
| | [-0.196089, | -0.016855] | [ 0.538188, | 0.684830] |
| i8 | -0.2315572 | 0.0348744 | 0.7975712 | 0.0274638 |
| | [-0.299910, | -0.163205] | [ 0.743743, | 0.851399] |
| i9 | -0.1414585 | 0.0325585 | 0.8342607 | 0.0266517 |
| | [-0.205272, | -0.077645] | [ 0.782024, | 0.886497] |

```
       i12  -0.2527785  0.0443307  0.6122940  0.0363948
            [-0.339665,-0.165892] [ 0.540961, 0.683627]

                                        U_Var
                     i1    0.5841469          0
                          [ 0.584147, 0.584147]

                     i4    0.3079070          0
                          [ 0.307907, 0.307907]

                     i5    0.5443523          0
                          [ 0.544352, 0.544352]

                     i11   0.8352026          0
                          [ 0.835203, 0.835203]

                     i3    0.6147208          0
                          [ 0.614721, 0.614721]

                     i8    0.3102614          0
                          [ 0.310261, 0.310261]

                     i9    0.2839986          0
                          [ 0.283999, 0.283999]

                     i12   0.5611991          0
                          [ 0.561199, 0.561199]
```

(b) WLS Estimation:

```
print "FACTOR: WLS Method Varimax";
optn = [ "data"      "raw" ,
         "anal"     "ccor" ,
         "hey"             ,
         "meth"      "wls" ,
         "cl"       "wald" ,
         "nfac"          2 ,
         "tech"   "nrridg" ,
         "frot"   "varmax" ,
         "phis"            ,
         "pini"          2 ,
         "prin"          3 ];
scal = [ 8#1 ];
gof = cfa(lot,optn,.,scal);
```

```
                          ******************
                          Optimization Start
                          ******************

                         Parameter Estimates
                         -------------------

Parameter      Estimate   Gradient   Lower BC   Upper BC

  1 L_11     0.50000000 -0.1963312 -1.0000000  1.0000000
  2 L_21     0.50000000  0.6445549 -1.0000000  1.0000000
  3 L_31     0.50000000  0.9538444 -1.0000000  1.0000000
  4 L_41     0.50000000  0.8698692 -1.0000000  1.0000000
  5 L_51     0.50000000  0.4073354 -1.0000000  1.0000000
  6 L_61     0.50000000  0.5696709 -1.0000000  1.0000000
  7 L_71     0.50000000 -0.4367535 -1.0000000  1.0000000
  8 L_81     0.50000000  0.6827442 -1.0000000  1.0000000
  9 L_22     0.50000000  0.7072668 -1.0000000  1.0000000
 10 L_32     0.50000000  1.0236212 -1.0000000  1.0000000
 11 L_42     0.50000000  0.8121345 -1.0000000  1.0000000
 12 L_52     0.50000000  0.3960169 -1.0000000  1.0000000
 13 L_62     0.50000000  0.4909156 -1.0000000  1.0000000
 14 L_72     0.50000000 -0.4211811 -1.0000000  1.0000000
 15 L_82     0.50000000  0.8788230 -1.0000000  1.0000000

          Value of Objective Function =      2.39406

              Newton-Raphson Ridge Optimization
                  Without Parameter Scaling
                Internally Specified Gradient
                Internally Specified Hessian


  Iteration Start:
  N. Variables             15
  N. Bound. Constr.        30        N. Mask  Constr.          0
  Criterion       2.394056402        Max Grad Entry  1.023621193
  N. Active Constraints     0


 Iter rest nfun act   optcrit  difcrit maxgrad   ridge      rho
    1    0    2   0  0.828510 1.565547 0.28788 0.03125  2.09051
    2    0    5   1' 0.778749 0.049760 0.92634 0.01250  0.21148
    3    0    6   0' 0.161962 0.616788 0.49996       0  1.94707
```

```
 4    0    7   0  0.064244 0.097718 0.21769       0  1.63216
 5    0    8   0  0.041705 0.022539 0.01068       0  2.00594
 6    0    9   0  0.041399 3.1e-004  2e-003       0  2.48229
 7    0   10   0  0.041363 3.6e-005  9e-004       0  2.77120
 8    0   11   0  0.041357 5.8e-006  3e-004       0  2.79862
 9    0   12   0  0.041356 9.5e-007  1e-004       0  2.80686
10    0   13   0  0.041356 1.6e-007  5e-005       0  2.80966
11    0   14   0  0.041356 2.6e-008  2e-005       0  2.81104
12    0   15   0  0.041356 4.3e-009  8e-006       0  2.81176

Successful Termination After    12 Iterations
ABSGCONV convergence criterion satisfied.
Criterion       0.041355862        Max Grad Entry  8.2026e-006
N. Active Constraints     0        Ridge (lambda)            0
Act.dF/Pred.dF  2.811763482
N. Function Calls        16        N. Gradient Calls         5
N. Hessian Calls         14        Preproces. Time           0
Time for Method           0        Effective Time            0

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

                      FAC_1                 FAC_2
  i1    0.6853502  0.0458036          0          0
       [ 0.595577, 0.775124] [        0,        0]

  i4    0.8584355  0.0386668  0.0141039  0.0808171
       [ 0.782650, 0.934221] [-0.144295, 0.172503]

  i5    0.6962619  0.0435945 -0.0906292  0.0733115
       [ 0.610818, 0.781705] [-0.234317, 0.053059]

  i11   0.4413202  0.0571657 -0.0516884  0.0691033
       [ 0.329277, 0.553363] [-0.187128, 0.083752]

  i3   -0.3098049  0.0755391  0.5996825  0.0556673
       [-0.457859,-0.161751] [ 0.490577, 0.708788]

  i8   -0.4667655  0.0732418  0.7696677  0.0531047
       [-0.610317,-0.323214] [ 0.665584, 0.873751]

  i9   -0.4200136  0.0776162  0.7989244  0.0474134
       [-0.572139,-0.267889] [ 0.705996, 0.891853]

  i12  -0.4487448  0.0764086  0.5951399  0.0615523
```

```
[-0.598503,-0.298987] [ 0.474500, 0.715780]

                        U_Var
      i1    0.5302951         0
           [ 0.530295, 0.530295]

      i4    0.2628897         0
           [ 0.262890, 0.262890]

      i5    0.5070058         0
           [ 0.507006, 0.507006]

      i11   0.8025648         0
           [ 0.802565, 0.802565]

      i3    0.5444018         0
           [ 0.544402, 0.544402]

      i8    0.1897416         0
           [ 0.189742, 0.189742]

      i9    0.1853084         0
           [ 0.185308, 0.185308]

      i12   0.4444366         0
           [ 0.444437, 0.444437]

(1) Standalone Fit Measures: --------------------------------
Fit criterion . . . . . .     0.0414
Normal Th. Chi-square (df = 13) 16.0461 Prob>chi**2 = 0.2467
Probability of Close Fit  . . . . . .     0.8725
Z-Test of Wilson & Hilferty (1931). . . . . .     0.6867
(2) Incremental Fit Measures: -------------------------------
Null Model Chi-square (df = 28) . . . . . .   11415.9663
RMSEA Estimate  . . . . .  0.0246  90%C.I.[ 0.0000,  0.0588]
ECVI Estimate . . . . . .   0.1187
McDonald's (1989) Centrality. . . . . .     0.9961
Tucker-Lewis Coefficient TLI. . . . . .     0.9994
Bentler & Bonett's (1980) NFI . . . . . .     0.9986
Bentler's Comparative Fit Index CFI . . . . . .     0.9997
Parsimonious NFI (James, Mulaik, & Brett,1982). .     0.4636
Bollen (1986) Normed Index Rho1 . . . . . .     0.9970
Bollen (1988) Non-normed Index Delta2 . . . . . .     0.9997
(3) Information Criteria: -----------------------------------
Akaike's Information Criterion. . . . . .    -9.9539
```

```
Bozdogan's (1987) CAIC. . . . . .   -74.4805
Schwarz's Bayesian Criterion. . . . . .   -61.4805
(4) Other Fit Measures: ----------------------------------
Goodness of Fit Index (GFI) . . . . . .     0.9848
Parsimonious GFI (Mulaik, 1989) . . . . . .     0.4572
GFI Adjusted for Degrees of Freedom (AGFI). . . .     0.9580
Root Mean Square Residual (RMR) . . . . . .     0.0342
Hoelter's (1983) Critical N . . . . . .        542
```

Rotation computed by common estimation algorithm.
ASE and CI computed by common estimation algorithm analytic version.

Rotated Factor Loadings with Standard Errors

| | FAC_1 | | FAC_2 | |
|---|---|---|---|---|
| i1 | 0.6562906 | 0.0382582 | -0.1974527 | 0.0400463 |
| | [ 0.581306, 0.731275] | | [-0.275942,-0.118963] | |
| i4 | 0.8261002 | 0.0351926 | -0.2338135 | 0.0333772 |
| | [ 0.757124, 0.895076] | | [-0.299232,-0.168395] | |
| i5 | 0.6406289 | 0.0378388 | -0.2873828 | 0.0394140 |
| | [ 0.566466, 0.714792] | | [-0.364633,-0.210133] | |
| i11 | 0.4077160 | 0.0484303 | -0.1766433 | 0.0480274 |
| | [ 0.312794, 0.502638] | | [-0.270775,-0.082511] | |
| i3 | -0.1238974 | 0.0421998 | 0.6635116 | 0.0312583 |
| | [-0.206607,-0.041187] | | [ 0.602246, 0.724777] | |
| i8 | -0.2252292 | 0.0288434 | 0.8715103 | 0.0177199 |
| | [-0.281761,-0.168697] | | [ 0.836780, 0.906241] | |
| i9 | -0.1720306 | 0.0284791 | 0.8860570 | 0.0173419 |
| | [-0.227849,-0.116213] | | [ 0.852067, 0.920047] | |
| i12 | -0.2582549 | 0.0387080 | 0.6991909 | 0.0278764 |
| | [-0.334121,-0.182389] | | [ 0.644554, 0.753828] | |

| | U_Var |
|---|---|
| i1 | 0.5302951 |
| | [ 0.530295, 0.530295] |

```
i4    0.2628897          0
     [ 0.262890, 0.262890]

i5    0.5070058          0
     [ 0.507006, 0.507006]

i11   0.8025648          0
     [ 0.802565, 0.802565]

i3    0.5444018          0
     [ 0.544402, 0.544402]

i8    0.1897416          0
     [ 0.189742, 0.189742]

i9    0.1853084          0
     [ 0.185308, 0.185308]

i12   0.4444366          0
     [ 0.444437, 0.444437]
```

2. The following example illustrates the automatic model improvement algorithm:

```
print "SEM33: Speed Factor Data, LORD (1956)";
print "Confirmatory Factor Analysis, JOERESKOG (1978, p. 454-5)";


lord = [ 1.00 ,
   .67   1.00 ,
   ......................................
   .28    .30    .31    .38    .36    .39    .26    .23    .25
   .27    .25    .31    .51    .53    .57    .64    .61   1.00 ];

lord = (tri2sym)lord;
mean = cons(1,18,0.);
std = [ 9.30   9.29   9.73   9.86   9.61   9.89   9.55   9.67   9.81
        9.80   9.83   9.88   9.05   9.67   9.73   9.91   9.85   9.83 ];
lord = lord |> mean |> std;
cnam = [" V1:V18 "];
lord = cname(lord,cnam);
rnam = [" V1:V18 Mean StdDev "];
lord = rname(lord,rnam); print "lord=", lord;
```

   (a) Starting from an EFA solution of six factors:

We should specify `nobs`, `data`, and `anal`, and for the model improvement `amimp` and `pbest`. In addition we suppress the toime consuming stage 2 of the model improvement algorithm. Many of the other options are the same as the default values. For the EFA starting we obtain: `p=0.0662, chi=77.25, df=60`

```
optn = [ "data"      "cor" ,
         "nobs"        649 ,
         "amimp"           ,
         "nost2"           ,
         "seed"    1234567 ,
         "anal"      "cov" ,
         "meth"       "ml" ,
         "cl"       "wald" ,
         "nfac"          6 ,
         "tech"   "nrridg" ,
         "frot"   "varmax" ,
         "pmod"            ,
         "phis"            ,
         "pbest"         1 ,
         "pini"          2 ,
         "pcpr"          3 ,
         "prin"          3 ];
< gof,parm,resi,covm,mod1,mod2 > = cfa(lord,optn);
```

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

  Input Data. . . . . .Correlation Matrix
  Analysis of . . . . . Covariance Matrix
  Number of Items . . . . . 18
  Rotation Method . . . . . 6
  Number of Subjects. . . . . 649
  Number of Parameters. . . . . 111
  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
                ---------------------------------
                V1                  0  9.30000000
                V2                  0  9.29000000
                V3                  0  9.73000000
```

```
V4                  0  9.86000000
V5                  0  9.61000000
V6                  0  9.89000000
V7                  0  9.55000000
V8                  0  9.67000000
V9                  0  9.81000000
V10                 0  9.80000000
V11                 0  9.83000000
V12                 0  9.88000000
V13                 0  9.05000000
V14                 0  9.67000000
V15                 0  9.73000000
V16                 0  9.91000000
V17                 0  9.85000000
V18                 0  9.83000000

     Variance Divisor = 648

          Covariance Matrix

             V1         V2         V3         V4         V5
V1    86.490000
V2    57.885990  86.304100
V3    64.247190  62.370273  94.672900
.....................................................
```

For EFA we obtain:

```
(1) Standalone Fit Measures: -------------------------------
Fit criterion . . . . . .     0.1192
Normal Th. Chi-square (df = 60) 77.2481 Prob>chi**2 = 0.0662
Bartlett Corrected Chi-square (df = 60)    75.9567 p= 0.0662
Normal Theory Reweighted LS Chi-square  . . . . .    77.6979
Probability of Close Fit  . . . . . .     1.0000
Z-Test of Wilson & Hilferty (1931). . . . . .     1.5048
(2) Incremental Fit Measures: -------------------------------
Null Model Chi-square (df = 153). . . . . .    9151.0889
RMSEA Estimate  . . . . . .  0.0211  90%C.I.[ . ,  0.0336]
ECVI Estimate . . . . . .  0.4722  90%C.I.[ . ,  0.5138]
McDonald's (1989) Centrality. . . . . .     0.9868
Tucker-Lewis Coefficient TLI. . . . . .     0.9951
Bentler & Bonett's (1980) NFI . . . . . .     0.9916
Bentler's Comparative Fit Index CFI . . . . . .     0.9981
Parsimonious NFI (James, Mulaik, & Brett,1982). .     0.3888
Bollen (1986) Normed Index Rho1 . . . . . .     0.9785
Bollen (1988) Non-normed Index Delta2 . . . . . .     0.9981
```

```
(3) Information Criteria: ----------------------------------
Akaike's Information Criterion. . . . . .   -42.7519
Bozdogan's (1987) CAIC. . . . . .  -371.2778
Schwarz's Bayesian Criterion. . . . . .  -311.2778
(4) Other Fit Measures: ------------------------------------
Goodness of Fit Index (GFI) . . . . . .     0.9869
Parsimonious GFI (Mulaik, 1989) . . . . . .     0.3870
GFI Adjusted for Degrees of Freedom (AGFI). . . .     0.9625
Root Mean Square Residual (RMR) . . . . . .     0.7250
Hoelter's (1983) Critical N . . . . . .        665
```

The folowing table shows some goodness-of-fit measures of the 20 best solutions found:

```
                  *********************************
                  Table of 20 Best Pattern Solutions
                  *********************************

                  N  Npar    DF  Chisquared           P

                  1    79    92  85.2135305  0.67856286
                  2    80    91  84.2673201  0.67802160
                  3    81    90  83.4467629  0.67393704
                  4    78    93  86.3848997  0.67286643
                  5    80    91  84.4531768  0.67281146
                  6    81    90  83.5408330  0.67127730
                  7    80    91  84.5127710  0.67113527
                  8    82    89  82.5913392  0.67078703
                  9    78    93  86.5004173  0.66965214
                 10    80    91  84.6237877  0.66800567
                 11    77    94  87.5951952  0.66612667
                 12    79    92  85.6671807  0.66588024
                 13    83    88  81.8506238  0.66429324
                 14    83    88  81.9426239  0.66163782
                 15    84    87  81.1757220  0.65576758
                 16    80    91  85.0787553  0.65508834
                 17    81    90  84.1702885  0.65331450
                 18    83    88  82.2329744  0.65321828
                 19    81    90  84.2072276  0.65225192
                 20    81    90  84.2279521  0.65165538

 N       RMSEA        ECVI        SRMR        PGFI         SBC

 1           0  0.38269473  0.01017654  0.59253452 -510.526279
 2           0  0.38441418  0.00955056  0.58617182 -504.997057
 3           0  0.38632754  0.00995298  0.57987654 -499.342182
```

```
 4          0  0.38132275  0.01081927  0.59884514 -515.830343
 5          0  0.38470100  0.01041157  0.58620778 -504.811200
 6          0  0.38647271  0.00967003  0.57988508 -499.248112
 7          0  0.38479296  0.01033851  0.58623987 -504.751606
 8          0  0.38818709  0.00951834  0.57350654 -493.722173
 9          0  0.38150102  0.00999945  0.59884918 -515.714825
10          0  0.38496428  0.00962449  0.58615414 -504.640589
11          0  0.38001084  0.01051875  0.60516582 -521.095480
12          0  0.38339481  0.01034245  0.59250328 -510.072629
13          0  0.39022366  0.00955591  0.56712318 -487.987455
14          0  0.39036564  0.00892742  0.56712193 -487.895455
15          0  0.39236180  0.00894192  0.56073829 -482.186924
16          0  0.38566639  0.01087797  0.58614049 -504.185622
17          0  0.38744409  0.00964478  0.57981904 -498.618656
18          0  0.39081371  0.00921256  0.56706840 -487.605105
19          0  0.38750109  0.01028350  0.57977563 -498.581717
20          0  0.38753308  0.01027658  0.57979962 -498.560992
```

The following is the best (w.r.t. $p$ value) solution found:

```
(1) Standalone Fit Measures: -------------------------------
Fit criterion . . . . . .                          0.1315
Normal Th. Chi-square (df = 92) 85.2135 Prob>chi**2 = 0.6786
Normal Theory Reweighted LS Chi-square  . . . . .    86.3447
Probability of Close Fit  . . . . . .              1.0000
Z-Test of Wilson & Hilferty (1931). . . . . .      -0.4640
(2) Incremental Fit Measures: ------------------------------
Null Model Chi-square (df = 153). . . . . .     9151.0889
RMSEA Estimate  . . . . . .  0.0000  90%C.I.[ . ,  0.0175]
ECVI Estimate . . . . . .  0.3827  90%C.I.[ . ,  0.4214]
McDonald's (1989) Centrality. . . . . .            1.0052
Tucker-Lewis Coefficient TLI. . . . . .            1.0013
Bentler & Bonett's (1980) NFI . . . . . .          0.9907
Bentler's Comparative Fit Index CFI . . . . . .    1.0000
Parsimonious NFI (James, Mulaik, & Brett,1982). .  0.5957
Bollen (1986) Normed Index Rho1 . . . . . .        0.9845
Bollen (1988) Non-normed Index Delta2 . . . . . .  1.0007
(3) Information Criteria: ----------------------------------
Akaike's Information Criterion. . . . . .    -98.7865
Bozdogan's (1987) CAIC. . . . . .           -602.5263
Schwarz's Bayesian Criterion. . . . . .     -510.5263
(4) Other Fit Measures: ------------------------------------
Goodness of Fit Index (GFI) . . . . . .            0.9854
Parsimonious GFI (Mulaik, 1989) . . . . . .        0.5925
GFI Adjusted for Degrees of Freedom (AGFI). . . .  0.9729
Root Mean Square Residual (RMR) . . . . . .        0.9522
```

Hoelter's (1983) Critical N . . . . . 879

Loadings and Uni. Vars. with ASE's and Wald CIs

| | FAC_1 | | FAC_2 | |
|---|---|---|---|---|
| V1 | 6.0747562 | 0.3192352 | 0 | 0 |
| | [ 5.449067, 6.700446] | | [ 0, 0] | |
| V2 | 6.2690451 | 0.3047188 | 3.7323110 | 0.3687300 |
| | [ 5.671807, 6.866283] | | [ 3.009613, 4.455009] | |
| V3 | 6.8315389 | 0.3164406 | 0 | 0 |
| | [ 6.211327, 7.451751] | | [ 0, 0] | |
| V4 | 6.9814744 | 0.3064515 | 0 | 0 |
| | [ 6.380840, 7.582108] | | [ 0, 0] | |
| V5 | 7.7346781 | 0.2791861 | -0.8058713 | 0.6058791 |
| | [ 7.187483, 8.281873] | | [-1.993372, 0.381630] | |
| V6 | 7.7965304 | 0.2885790 | 0 | 0 |
| | [ 7.230926, 8.362135] | | [ 0, 0] | |
| V7 | 0 | 0 | 0 | 0 |
| | [ 0, 0] | | [ 0, 0] | |
| V8 | 0 | 0 | 1.9027263 | 0.5372056 |
| | [ 0, 0] | | [ 0.849823, 2.955630] | |
| V9 | 0 | 0 | 2.5858752 | 0.5357122 |
| | [ 0, 0] | | [ 1.535898, 3.635852] | |
| V10 | 0.2418159 | 0.2230595 | 0 | 0 |
| | [-0.195373, 0.679004] | | [ 0, 0] | |
| V11 | 0 | 0 | 0 | 0 |
| | [ 0, 0] | | [ 0, 0] | |
| V12 | -0.3651054 | 0.2058398 | -1.0144925 | 0.8613447 |
| | [-0.768544, 0.038333] | | [-2.702697, 0.673712] | |
| V13 | 0 | 0 | 6.5960409 | 0.3336407 |
| | [ 0, 0] | | [ 5.942117, 7.249965] | |

```
V14           0          0  4.5460003  1.2759135
     [        0,        0] [ 2.045256, 7.046745]

V15           0          0  7.9667686  0.3361649
     [        0,        0] [ 7.307897, 8.625640]

V16           0          0  1.3741826  1.1682469
     [        0,        0] [-0.915539, 3.663904]

V17           0          0          0          0
     [        0,        0] [        0,        0]

V18           0          0          0          0
     [        0,        0] [        0,        0]

                       FAC_3                 FAC_4
V1    3.4161704  0.4041541          0          0
     [ 2.624043, 4.208298] [        0,        0]

V2    2.5573874  0.3593729          0          0
     [ 1.853029, 3.261745] [        0,        0]

V3    2.3954909  0.3553351          0          0
     [ 1.699047, 3.091935] [        0,        0]

V4            0          0  0.7430749  0.2433227
     [        0,        0] [ 0.266171, 1.219979]

V5            0          0          0          0
     [        0,        0] [        0,        0]

V6   -1.5772969  0.3935643          0          0
     [-2.348669,-0.805925] [        0,        0]

V7    1.1487373  0.3270379  7.4724639  0.3307317
     [ 0.507755, 1.789720] [ 6.824242, 8.120686]

V8            0          0  7.8868934  0.3247016
     [        0,        0] [ 7.250490, 8.523297]

V9            0          0  7.7725421  0.3331376
     [        0,        0] [ 7.119604, 8.425480]

V10           0          0  7.7908481  0.3409861
     [        0,        0] [ 7.122528, 8.459169]
```

```
V11           0          0  8.0678101  0.3398047
     [        0,        0] [ 7.401805, 8.733815]

V12  -0.3070999  0.2672710  8.2697732  0.4107713
     [-0.830941, 0.216742] [ 7.464676, 9.074870]

V13   1.2193289  0.3810979  0.4621403  0.3194442
     [ 0.472391, 1.966267] [-0.163959, 1.088239]

V14   0.5053273  0.4213431  1.2566481  0.4497717
     [-0.320490, 1.331145] [ 0.375112, 2.138184]

V15           0          0          0          0
     [        0,        0] [        0,        0]

V16  -0.8807392  0.4042794  1.2153245  0.4726850
     [-1.673112,-0.088366] [ 0.288879, 2.141770]

V17           0          0  1.8450771  0.4574200
     [        0,        0] [ 0.948550, 2.741604]

V18  -0.5605770  0.4158638  0.9477719  0.4582692
     [-1.375655, 0.254501] [ 0.049581, 1.845963]

                      FAC_5                 FAC_6
V1            0          0  3.5174959  0.3770434
     [        0,        0] [ 2.778504, 4.256487]

V2    0.5448187  0.3959450          0          0
     [-0.231219, 1.320857] [        0,        0]

V3    0.6514501  0.3773161  3.8329957  0.3868469
     [-0.088076, 1.390976] [ 3.074790, 4.591202]

V4            0          0  4.2610236  0.3890313
     [        0,        0] [ 3.498536, 5.023511]

V5            0          0  5.0550821  0.6868359
     [        0,        0] [ 3.708908, 6.401256]

V6            0          0  4.6333960  0.3848138
     [        0,        0] [ 3.879175, 5.387617]

V7            0          0  2.3526910  0.5084332
```

```
     [        0,        0] [ 1.356180, 3.349202]

V8            0         0          0         0
     [        0,        0] [        0,        0]

V9   -0.5328967  0.4558218          0         0
     [-1.426291, 0.360498] [        0,        0]

V10  -2.5985344  0.4030238  2.4476466  0.5217805
     [-3.388447,-1.808622] [ 1.424976, 3.470318]

V11  -3.3296722  0.4091657  2.2189112  0.5295357
     [-4.131622,-2.527722] [ 1.181040, 3.256782]

V12  -2.7108424  0.3956780  3.7486054  0.9394598
     [-3.486357,-1.935328] [ 1.907298, 5.589913]

V13           0         0          0         0
     [        0,        0] [        0,        0]

V14   1.3161751  0.4924278  2.2124409  1.2228953
     [ 0.351034, 2.281316] [-0.184390, 4.609272]

V15           0         0          0         0
     [        0,        0] [        0,        0]

V16           0         0  6.6596957  1.1481773
     [        0,        0] [ 4.409310, 8.910082]

V17           0         0  7.5014883  0.3536851
     [        0,        0] [ 6.808278, 8.194698]

V18           0         0  7.6458403  0.3452048
     [        0,        0] [ 6.969251, 8.322429]

                          U_Var
           V1    25.617552  2.5893319
                [ 20.54255, 30.69255]

           V2    26.290625  2.0769953
                [ 22.21979, 30.36146]

           V3    27.252253  2.0280138
                [ 23.27742, 31.22709]
```

```
V4    29.554273  1.8696118
     [ 25.88990, 33.21864]

V5    13.376145  1.2393376
     [ 10.94709, 15.80520]

V6    12.771532  2.1393653
     [ 8.578453, 16.96461]

V7    28.253298  2.0022069
     [ 24.32904, 32.17755]

V8    21.883150  1.8556336
     [ 18.24617, 25.52013]

V9    21.091634  1.7060726
     [ 17.74779, 24.43547]

V10   22.459287  1.6032229
     [ 19.31703, 25.60155]

V11   15.497833  1.8217571
     [ 11.92725, 19.06841]

V12   16.557163  1.4415652
     [ 13.73175, 19.38258]

V13   35.387935  2.5068006
     [ 30.47470, 40.30117]

V14   44.234787  2.9645971
     [ 38.42428, 50.04529]

V15   31.148385  2.6993969
     [ 25.85766, 36.43911]

V16   32.829445  2.4880042
     [ 27.95305, 37.70584]

V17   37.274933  2.6362423
     [ 32.10799, 42.44187]

V18   36.946898  2.7000343
     [ 31.65493, 42.23887]
```

Factor Correlations Phi

| | FAC_1 | FAC_2 |
|---|---|---|
| FAC_1 | 1.0000000 | 0 |
| | [ 1.000000, 1.000000] | [ 0, 0] |
| FAC_2 | 0 | 1.0000000 |
| | [ 0, 0] | [ 1.000000, 1.000000] |
| FAC_3 | 0 | 0 |
| | [ 0, 0] | [ 0, 0] |
| FAC_4 | 0 | 0.1909446 |
| | [ 0, 0] | [ 0.088154, 0.293736] |
| FAC_5 | 0 | 0 |
| | [ 0, 0] | [ 0, 0] |
| FAC_6 | 0 | 0.8874489 |
| | [ 0, 0] | [ 0.840040, 0.934857] |

| | FAC_3 | FAC_4 |
|---|---|---|
| FAC_1 | 0 | 0 |
| | [ 0, 0] | [ 0, 0] |
| FAC_2 | 0 | 0.1909446 |
| | [ 0, 0] | [ 0.088154, 0.293736] |
| FAC_3 | 1.0000000 | 0 |
| | [ 1.000000, 1.000000] | [ 0, 0] |
| FAC_4 | 0 | 1.0000000 |
| | [ 0, 0] | [ 1.000000, 1.000000] |
| FAC_5 | 0 | 0 |
| | [ 0, 0] | [ 0, 0] |
| FAC_6 | 0 | 0 |
| | [ 0, 0] | [ 0, 0] |

| | FAC_5 | FAC_6 |
|---|---|---|
| FAC_1 | 0 | 0 |
| | [ 0, 0] | [ 0, 0] |

```
FAC_2           0           0  0.8874489  0.0241884
       [        0,         0] [ 0.840040, 0.934857]

FAC_3           0           0          0          0
       [        0,         0] [        0,         0]

FAC_4           0           0          0          0
       [        0,         0] [        0,         0]

FAC_5   1.0000000           0          0          0
       [ 1.000000, 1.000000] [        0,         0]

FAC_6           0           0  1.0000000          0
       [        0,         0] [ 1.000000, 1.000000]
```

(b) Starting from a specified pattern (Joereskog, 1978, p. 454-5), see also `sem33.inp`:

```
print "*** CFA Model in SEM33: ***";

l = [" Z1  5 # 0. ,
       Z2  5 # 0. ,
       Z3  0.  0. Z19 0. 0. ,
       Z4  0.  0. Z20 0. 0. ,
       Z5  0.  0. Z21 0. 0. ,
       Z6  0.  0. Z22 0. 0. ,
       0.  Z7  4 # 0. ,
       0.  Z8  4 # 0. ,
       0.  Z9  0. 0. Z23 0. ,
       0.  Z10 0. 0. Z24 0. ,
       0.  Z11 0. 0. Z25 0. ,
       0.  Z12 0. 0. Z26 0. ,
       0.  0. Z13  3 # 0. ,
       0.  0. Z14  3 # 0. ,
       0.  0. Z15  0. 0. Z27 ,
       0.  0. Z16  0. 0. Z28 ,
       0.  0. Z17  0. 0. Z29 ,
       0.  0. Z18  0. 0. Z30 "]; print "L=",l;

p = [" 1., C1 1., C2 C3 1., 0. 0. 0. 1.,
       0. 0. 0. C4 1., 0. 0. 0. C5 C6 1. "];
p = (tri2sym)p; print "P=",p;
u = [" u1:u18 "]; print "U=",u;

list patt;
patt[1] = l; patt[2] = p; patt[3] = u;
```

```
print "Patt=",patt;
```

Note, that the starting pattern has: `Chi-square (df = 117) 174.5032 Prob>chi**2 = 0.0005";` which has a worse $p$ value than the EFA.

```
optn = [ "data"      "cor" ,
         "nobs"        649 ,
         "anal"      "cov" ,
         "hey"             ,
         "meth"       "ml" ,
         "cl"       "wald" ,
         "nfac"           6 ,
         "amimp"           ,
         "nost2"           ,
         "seed"    1234567 ,
         "tech"   "nrridg" ,
         "pmod"            ,
         "phis"            ,
         "pbest"         1 ,
         "pini"          2 ,
         "pcpr"          3 ,
         "prin"          3 ];
< gof,parm,resi,covm,mod1,mod2 > = cfa(lord,optn,patt);
```

The following table of the goodness-of-fit corresponds to the here specified CFA pattern:

```
(1) Standalone Fit Measures: -------------------------------
Fit criterion . . . . . .                      0.2693
Normal Th. Chi-square (df = 117)      174.5032  prob= 0.0005
Normal Theory Reweighted LS Chi-square  . . . . .   174.6905
Probability of Close Fit  . . . . . .          1.0000
Z-Test of Wilson & Hilferty (1931). . . . . .      3.3143
(2) Incremental Fit Measures: -------------------------------
Null Model Chi-square (df = 153). . . . . .    9151.0889
RMSEA Estimate  . . . . .  0.0275  90%C.I.[ 0.0185,  0.0358]
ECVI Estimate . . . . . .  0.4410  90%C.I.[ 0.3922,  0.5025]
McDonald's (1989) Centrality. . . . . .          0.9567
Tucker-Lewis Coefficient TLI. . . . . .          0.9916
Bentler & Bonett's (1980) NFI . . . . . .        0.9809
Bentler's Comparative Fit Index CFI . . . . . .  0.9936
Parsimonious NFI (James, Mulaik, & Brett,1982). .  0.7501
Bollen (1986) Normed Index Rho1 . . . . . .      0.9751
Bollen (1988) Non-normed Index Delta2 . . . . . .  0.9936
(3) Information Criteria: ----------------------------------
Akaike's Information Criterion. . . . . .   -59.4968
```

```
Bozdogan's (1987) CAIC. . . . . -700.1224
Schwarz's Bayesian Criterion. . . . . -583.1224
(4) Other Fit Measures: ----------------------------------
Goodness of Fit Index (GFI) . . . . . 0.9709
Parsimonious GFI (Mulaik, 1989) . . . . . 0.7425
GFI Adjusted for Degrees of Freedom (AGFI). . . . 0.9575
Root Mean Square Residual (RMR) . . . . . 2.4779
Hoelter's (1983) Critical N . . . . . 533
```

The folowing table shows some goodness-of-fit measures of the 20 best solutions found:

```
**********************************
Table of 20 Best Pattern Solutions
**********************************
```

| N | Npar | DF | Chisquared | P |
|---|---|---|---|---|
| 1 | 78 | 93 | 81.5381528 | 0.79623215 |
| 2 | 79 | 92 | 80.6208549 | 0.79574017 |
| 3 | 80 | 91 | 79.7065417 | 0.79518075 |
| 4 | 79 | 92 | 80.6894375 | 0.79417189 |
| 5 | 78 | 93 | 81.7631267 | 0.79110149 |
| 6 | 78 | 93 | 81.7631285 | 0.79110145 |
| 7 | 80 | 91 | 79.9401546 | 0.78977331 |
| 8 | 78 | 93 | 81.8288341 | 0.78959057 |
| 9 | 79 | 92 | 80.9022367 | 0.78926635 |
| 10 | 79 | 92 | 80.9402349 | 0.78838417 |
| 11 | 80 | 91 | 80.0352317 | 0.78755187 |
| 12 | 77 | 94 | 82.8914611 | 0.78679330 |
| 13 | 79 | 92 | 81.0210535 | 0.78650160 |
| 14 | 80 | 91 | 80.0900492 | 0.78626568 |
| 15 | 80 | 91 | 80.1373502 | 0.78515270 |
| 16 | 80 | 91 | 80.1479306 | 0.78490334 |
| 17 | 77 | 94 | 83.0093101 | 0.78406718 |
| 18 | 79 | 92 | 81.1947973 | 0.78242590 |
| 19 | 80 | 91 | 80.2556597 | 0.78235615 |
| 20 | 80 | 91 | 80.3037022 | 0.78121537 |

| N | RMSEA | ECVI | SRMR | PGFI | SBC |
|---|---|---|---|---|---|
| 1 | 0 | 0.37384320 | 0.01069254 | 0.59955334 | -520.677090 |
| 2 | 0 | 0.37560727 | 0.01070647 | 0.59320590 | -515.118955 |
| 3 | 0 | 0.37737594 | 0.01067296 | 0.58685509 | -509.557836 |
| 4 | 0 | 0.37571311 | 0.01066932 | 0.59319620 | -515.050372 |
| 5 | 0 | 0.37419038 | 0.01150166 | 0.59953339 | -520.452116 |

```
 6          0  0.37419039  0.01150260  0.59953332 -520.452114
 7          0  0.37773646  0.01032498  0.58681796 -509.324223
 8          0  0.37429178  0.01019904  0.59952617 -520.386409
 9          0  0.37604150  0.01008511  0.59317850 -514.837573
10          0  0.37610014  0.00997784  0.59315233 -514.799575
11          0  0.37788318  0.01008457  0.58682299 -509.229146
12          0  0.37275199  0.01068605  0.60586368 -525.799214
13          0  0.37622486  0.01049025  0.59316847 -514.718756
14          0  0.37796777  0.01056663  0.58679836 -509.174328
15          0  0.37804077  0.00989360  0.58679208 -509.127027
16          0  0.37805710  0.01038400  0.58680741 -509.116447
17          0  0.37293386  0.01090651  0.60584371 -525.681365
18          0  0.37649298  0.01131906  0.59313065 -514.545013
19          0  0.37822335  0.00986701  0.58679674 -509.008718
20          0  0.37829749  0.00992382  0.58676376 -508.960675
```

The following is the best (w.r.t. $p$ value) solution found:

```
(1) Standalone Fit Measures: ------------------------------
Fit criterion . . . . . .     0.1258
Normal Th. Chi-square (df = 93) 81.5382 Prob>chi**2 = 0.7962
Normal Theory Reweighted LS Chi-square  . . . . .    80.6368
Probability of Close Fit  . . . . . .     1.0000
Z-Test of Wilson & Hilferty (1931). . . . . .    -0.8286
(2) Incremental Fit Measures: -----------------------------
Null Model Chi-square (df = 153). . . . . .    9151.0889
RMSEA Estimate  . . . . . .  0.0000  90%C.I.[ . ,  0.0144]
ECVI Estimate . . . . . .  0.3738  90%C.I.[ . ,  0.4109]
McDonald's (1989) Centrality. . . . . .     1.0089
Tucker-Lewis Coefficient TLI. . . . . .     1.0021
Bentler & Bonett's (1980) NFI . . . . . .     0.9911
Bentler's Comparative Fit Index CFI . . . . . .     1.0000
Parsimonious NFI (James, Mulaik, & Brett,1982). .     0.6024
Bollen (1986) Normed Index Rho1 . . . . . .     0.9853
Bollen (1988) Non-normed Index Delta2 . . . . . .     1.0013
(3) Information Criteria: ---------------------------------
Akaike's Information Criterion. . . . . .  -104.4618
Bozdogan's (1987) CAIC. . . . . .  -613.6771
Schwarz's Bayesian Criterion. . . . . .  -520.6771
(4) Other Fit Measures: -----------------------------------
Goodness of Fit Index (GFI) . . . . . .     0.9864
Parsimonious GFI (Mulaik, 1989) . . . . . .     0.5996
GFI Adjusted for Degrees of Freedom (AGFI). . . .     0.9749
Root Mean Square Residual (RMR) . . . . . .     1.0015
Hoelter's (1983) Critical N . . . . . .       927
```

```
 Loadings and Uni. Vars. with ASE's and Wald CIs
 -----------------------------------------------

                    FAC_1                 FAC_2
V1    7.9048077  0.3337829          0          0
     [ 7.250605, 8.559010] [        0,        0]

V2    6.9803326  0.3695865  0.4561451  0.2878152
     [ 6.255956, 7.704709] [-0.107962, 1.020253]

V3    8.1126008  0.3468353          0          0
     [ 7.432816, 8.792386] [        0,        0]

V4    7.0617506  0.3809097  0.8680795  0.2444396
     [ 6.315181, 7.808320] [ 0.388987, 1.347172]

V5    7.9511053  0.3841680          0          0
     [ 7.198150, 8.704061] [        0,        0]

V6    7.4182764  0.3856758          0          0
     [ 6.662366, 8.174187] [        0,        0]

V7            0          0  7.1943414  0.3553490
     [        0,        0] [ 6.497870, 7.890813]

V8   -0.3636943  0.2518186  8.6001637  0.3261950
     [-0.857250, 0.129861] [ 7.960833, 9.239494]

V9            0          0  8.3824394  0.3500625
     [        0,        0] [ 7.696329, 9.068549]

V10   0.3007870  0.2294985  7.9503836  0.3444807
     [-0.149022, 0.750596] [ 7.275214, 8.625553]

V11           0          0  8.3534584  0.3748084
     [        0,        0] [ 7.618847, 9.088069]

V12           0          0  8.3826455  0.3371882
     [        0,        0] [ 7.721769, 9.043522]

V13           0          0          0          0
     [        0,        0] [        0,        0]

V14  -0.5970233  0.3772023          0          0
     [-1.336326, 0.142280] [        0,        0]
```

```
V15           0          0          0          0
     [        0,        0] [        0,        0]

V16           0          0          0          0
     [        0,        0] [        0,        0]

V17           0          0          0          0
     [        0,        0] [        0,        0]

V18           0          0 -0.6911756  0.3401656
     [        0,        0] [-1.357888,-0.024463]

                      FAC_3                 FAC_4
V1            0          0          0          0
     [        0,        0] [        0,        0]

V2    0.6764332  0.3581798  2.8017250  0.7601041
     [-0.025586, 1.378453] [ 1.311948, 4.291502]

V3            0          0  1.4098592  0.5236851
     [        0,        0] [ 0.383455, 2.436263]

V4            0          0  3.9553372  0.4428482
     [        0,        0] [ 3.087371, 4.823304]

V5   -0.2770268  0.2290819  4.2465012  0.4250808
     [-0.726019, 0.171965] [ 3.413358, 5.079644]

V6            0          0  4.7168930  0.5244604
     [        0,        0] [ 3.688970, 5.744816]

V7    0.9539578  0.3136759          0          0
     [ 0.339164, 1.568751] [        0,        0]

V8            0          0          0          0
     [        0,        0] [        0,        0]

V9    0.4825560  0.2805690          0          0
     [-0.067349, 1.032461] [        0,        0]

V10           0          0          0          0
     [        0,        0] [        0,        0]

V11  -0.2997001  0.2564510          0          0
     [-0.802335, 0.202935] [        0,        0]
```

```
V12           0          0 -1.1424638  0.5286793
     [        0,        0] [-2.178656,-0.106271]

V13   6.7410224  0.3231879          0          0
     [ 6.107586, 7.374459] [        0,        0]

V14   7.3636503  0.4048054  1.0338878  0.4236233
     [ 6.570246, 8.157054] [ 0.203601, 1.864174]

V15   7.4407064  0.3440395          0          0
     [ 6.766401, 8.115011] [        0,        0]

V16   7.0819564  0.3644657 -1.6716650  1.0041550
     [ 6.367617, 7.796296] [-3.639773, 0.296443]

V17   6.9578015  0.3581885          0          0
     [ 6.255765, 7.659838] [        0,        0]

V18   6.9989697  0.4134451          0          0
     [ 6.188632, 7.809307] [        0,        0]

                      FAC_5                 FAC_6
V1            0          0          0          0
     [        0,        0] [        0,        0]

V2            0          0 -2.3784533  0.6153506
     [        0,        0] [-3.584518,-1.172388]

V3   -0.6651066  0.3549458          0          0
     [-1.360788, 0.030574] [        0,        0]

V4            0          0          0          0
     [        0,        0] [        0,        0]

V5    0.5641462  0.2807503          0          0
     [ 0.013886, 1.114407] [        0,        0]

V6            0          0  0.8216057  0.4620565
     [        0,        0] [-0.084008, 1.727220]

V7    1.2543628  0.5103031 -0.8655516  0.3310354
     [ 0.254187, 2.254538] [-1.514369,-0.216734]

V8            0          0          0          0
```

```
     [        0,        0] [        0,        0]

V9    0.7986902  0.5535954          0          0
     [-0.286337, 1.883717] [        0,        0]

V10   3.1682100  0.4536924          0          0
     [ 2.278989, 4.057431] [        0,        0]

V11   3.6726584  0.4487705          0          0
     [ 2.793084, 4.552232] [        0,        0]

V12   2.8853864  0.4667838  1.7414352  0.5134030
     [ 1.970507, 3.800266] [ 0.735184, 2.747687]

V13   0.8441631  0.4191038          0          0
     [ 0.022735, 1.665592] [        0,        0]

V14           0          0          0          0
     [        0,        0] [        0,        0]

V15           0          0  1.6319038  0.4129220
     [        0,        0] [ 0.822592, 2.441216]

V16           0          0  5.4123258  0.9220005
     [        0,        0] [ 3.605238, 7.219414]

V17   1.6105846  0.4475966  2.4566180  0.4334586
     [ 0.733311, 2.487858] [ 1.607055, 3.306181]

V18   0.7574243  0.4705247  3.3148487  0.4473178
     [-0.164787, 1.679636] [ 2.438122, 4.191576]

                          U_Var
          V1    24.004015  2.8836145
               [ 18.35223, 29.65580]

          V2    26.413494  2.1932018
               [ 22.11490, 30.71209]

          V3    26.480339  2.1205355
               [ 22.32417, 30.63651]

          V4    29.070432  1.8914593
               [ 25.36324, 32.77762]
```

```
V5    12.542132  1.2902478
     [ 10.01329, 15.07097]

V6    14.640888  1.8346298
     [ 11.04508, 18.23670]

V7    30.268527  1.9725035
     [ 26.40249, 34.13456]

V8    20.269374  2.4183440
     [ 15.52951, 25.00924]

V9    21.341181  1.8029214
     [ 17.80752, 24.87484]

V10   22.156482  1.6144623
     [ 18.99219, 25.32077]

V11   15.763868  1.6235451
     [ 12.58178, 18.94596]

V12   15.900615  1.4611527
     [ 13.03681, 18.76442]

V13   35.740207  2.5124744
     [ 30.81585, 40.66457]

V14   41.812196  2.9777629
     [ 35.97589, 47.64850]

V15   36.436350  2.4815972
     [ 31.57251, 41.30019]

V16   27.388732  3.4595923
     [ 20.60806, 34.16941]

V17   38.328595  2.6169997
     [ 33.19937, 43.45782]

V18   38.886965  2.6762927
     [ 33.64153, 44.13240]

  Factor Correlations Phi
  -----------------------
```

```
                    FAC_1                 FAC_2
FAC_1   1.0000000         0  0.1367663  0.0455263
       [ 1.000000, 1.000000] [ 0.047536, 0.225996]

FAC_2   0.1367663  0.0455263  1.0000000         0
       [ 0.047536, 0.225996] [ 1.000000, 1.000000]

FAC_3   0.4686891  0.0384353  0.4685015  0.0377181
       [ 0.393357, 0.544021] [ 0.394575, 0.542428]

FAC_4          0          0          0          0
       [       0,         0] [       0,         0]

FAC_5          0          0          0          0
       [       0,         0] [       0,         0]

FAC_6          0          0          0          0
       [       0,         0] [       0,         0]

                    FAC_3                 FAC_4
FAC_1   0.4686891  0.0384353          0          0
       [ 0.393357, 0.544021] [       0,         0]

FAC_2   0.4685015  0.0377181          0          0
       [ 0.394575, 0.542428] [       0,         0]

FAC_3   1.0000000         0          0          0
       [ 1.000000, 1.000000] [       0,         0]

FAC_4          0          0  1.0000000         0
       [       0,         0] [ 1.000000, 1.000000]

FAC_5          0          0          0          0
       [       0,         0] [       0,         0]

FAC_6          0          0  0.6540878  0.1073309
       [       0,         0] [ 0.443723, 0.864452]

                    FAC_5                 FAC_6
FAC_1          0          0          0          0
       [       0,         0] [       0,         0]

FAC_2          0          0          0          0
       [       0,         0] [       0,         0]
```

```
FAC_3           0          0          0          0
       [        0,        0] [        0,        0]

FAC_4           0          0  0.6540878  0.1073309
       [        0,        0] [ 0.443723, 0.864452]

FAC_5   1.0000000          0  0.1603555  0.0849690
       [ 1.000000, 1.000000] [-6.2e-003, 0.326892]

FAC_6   0.1603555  0.0849690  1.0000000          0
       [-6.2e-003, 0.326892] [ 1.000000, 1.000000]
```

Standardized Factor Loadings

| | FAC_1 | FAC_2 | FAC_3 | FAC_4 | FAC_5 |
|---|---|---|---|---|---|
| V1 | 0.8499793 | 0 | 0 | 0 | 0 |
| V2 | 0.7519239 | 0.0491361 | 0.0728656 | 0.3018028 | 0 |
| V3 | 0.8335439 | 0 | 0 | 0.1448585 | -0.0683376 |
| V4 | 0.7169611 | 0.0881338 | 0 | 0.4015751 | 0 |
| V5 | 0.8283963 | 0 | -0.0288624 | 0.4424273 | 0.0587763 |
| V6 | 0.7506410 | 0 | 0 | 0.4772932 | 0 |
| V7 | 0 | 0.7527576 | 0.0998144 | 0 | 0.1312464 |
| V8 | -0.0376106 | 0.8893654 | 0 | 0 | 0 |
| V9 | 0 | 0.8543401 | 0.0491822 | 0 | 0.0814027 |
| V10 | 0.0306754 | 0.8108111 | 0 | 0 | 0.3231064 |
| V11 | 0 | 0.8491438 | -0.0304650 | 0 | 0.3733322 |
| V12 | 0 | 0.8474608 | 0 | -0.1154997 | 0.2917041 |
| V13 | 0 | 0 | 0.7449021 | 0 | 0.0932824 |
| V14 | -0.0617956 | 0 | 0.7621833 | 0.1070138 | 0 |
| V15 | 0 | 0 | 0.7655649 | 0 | 0 |
| V16 | 0 | 0 | 0.7161360 | -0.1690408 | 0 |
| V17 | 0 | 0 | 0.7077816 | 0 | 0.1638365 |
| V18 | 0 | -0.0704751 | 0.7136434 | 0 | 0.0772301 |

Standardized Factor Loadings

| | FAC_6 |
|---|---|
| V1 | 0 |
| V2 | -0.2562078 |
| V3 | 0 |
| V4 | 0 |
| V5 | 0 |
| V6 | 0.0831367 |
| V7 | -0.0905643 |
| V8 | 0 |

| | |
|---|---|
| V9 | 0 |
| V10 | 0 |
| V11 | 0 |
| V12 | 0.1760540 |
| V13 | 0 |
| V14 | 0 |
| V15 | 0.1679045 |
| V16 | 0.5473010 |
| V17 | 0.2498992 |
| V18 | 0.3379955 |

Squared Multiple Correlations

| N | Variable | ErrVariance | TotVariance | R-squared |
|---|---|---|---|---|
| 1 | V1 | 24.0040155 | 86.4900000 | 0.72246485 |
| 2 | V2 | 26.4134939 | 86.1795863 | 0.69350637 |
| 3 | V3 | 26.4803389 | 94.7247007 | 0.72044948 |
| 4 | V4 | 29.0704323 | 97.0138073 | 0.70034748 |
| 5 | V5 | 12.5421324 | 92.1252508 | 0.86385782 |
| 6 | V6 | 14.6408878 | 97.6655652 | 0.85009161 |
| 7 | V7 | 30.2685271 | 91.3422579 | 0.66862515 |
| 8 | V8 | 20.2693741 | 93.5089000 | 0.78323588 |
| 9 | V9 | 21.3411810 | 96.2674105 | 0.77831354 |
| 10 | V10 | 22.1564819 | 96.1472275 | 0.76955673 |
| 11 | V11 | 15.7638682 | 96.7765581 | 0.83711068 |
| 12 | V12 | 15.9006152 | 97.8414669 | 0.83748593 |
| 13 | V13 | 35.7402067 | 81.8942007 | 0.56358074 |
| 14 | V14 | 41.8121957 | 93.3399340 | 0.55204387 |
| 15 | V15 | 36.4363504 | 94.4635722 | 0.61428147 |
| 16 | V16 | 27.3887321 | 97.7947295 | 0.71993652 |
| 17 | V17 | 38.3285947 | 96.6374730 | 0.60337751 |
| 18 | V18 | 38.8869653 | 96.1846352 | 0.59570502 |

## 5.3 Function `deg2rad`

r = deg2rad(d)

**Purpose:** Transforms the degrees of an angle into radians.

**Input:** A scalar, vector or matrix of angles measured in degrees.

**Output:** A scalar, vector or matrix of angles measured in radians.

**Restrictions:**
1. A missing value for $r$ is returned if the input $d$ contains string or complex data.
2. For any missing value in $d$ a missing value is returned in $r$.

**Relationships:** rad2deg()

**Examples:**

```
d1 = 45; d2 = 60;
print "d1=",d1," d2=",d2;
r1 = deg2rad(d1); r2 = deg2rad(d2);
print "r1=",r1," r2=",r2;
d1 = rad2deg(r1); d2 = rad2deg(r2);
print "d1=",d1," d2=",d2;
```

```
d1= 45 d2= 60
r1= 0.7854 r2= 1.0472
d1= 45.000 d2= 60.000
```

## 5.4 Function `invupd`

b = invupd(a,v or s)

c = invupd(a,v,s)

**Purpose:** This function implements the rank-1 update of the inverse of a positive definite (symmetric) matrix. There are three forms of the update:

- when an additional ror and column is appended (the size of the matrix is increased)
- when a row and corresponding column is deleted (the size of the matrix is reduced)
- the common rank-1 update $a + s * v^T * v$

Note, for numerical stability the function `triupd()` is normally preferred.

**Input:** **a** an $n \times n$ inverse matrix w.r.t. an optimization function `fopt` which is usually a Maximum Likelihood function.

**v** is a vector

**s** is a positive real scalar

**Output:** **b**

**c**

**Restrictions:** 1. The first input argument cannot have missing values and neither string nor complex data.

**Relationships:** inv(), triupd()

**Examples:** 1. Update inverse when symmetrically appending row and column:

```
print "*** INVUPD ***";
 b = [ 2 1,
       4 3,
       6 5];
 aa = b' * b;
 print "Matrix A=",aa;
 c = chold(aa);
 print "Cholesky factor C", c;


 print "Update: append a row and column";
 v = [ .1 .2 ]; z = 1.;
 w = v -> z;
```

```
ai = inv(aa);
print "Inverse ai=", ai;
c1 = invupd(ai,w);

av = (aa -> v‘) |> (v -> z);
c2 = inv(av);
res = ssq(c1 - c2);
print "C1=",c1;
print "C2=",c2;
print "Res=", res;
```

2. Update inverse when deleting row and column:

```
print "Downdate: delete a row and column";
c3 = invupd(c2,3);
res = ssq(c3 - ai);
print "C3=",c3;
print "Ai=",ai;
print "Res=", res;
```

3. Rank-1 update of inverse:

```
c8 = invupd(ai,v,z);
w = z * v‘ * v;
c9 = inv(aa + w);
res = ssq(c8 - c9);
print "C8=",c8;
print "C9=",c9;
print "Res=", res;
```

```
ee = ai * v‘; ss = v * ee;
t = z / (1. + z * ss);
c7 = ai - t * ee * ee‘;
print "C7=", c7;
res = ssq(c7 - c9);
print "Res=", res;
```

## 5.5 Function `som`

<gof,scal,xwgt,ywgt,nass,tmtrn,cwtrn,ptrn,mtrn, tmtst,cwtst,ptst,mtst > =
= som(xtrn,ytrn<,optn<,neus<,epos<,xtst<,ytst> . >)

**Purpose:** Self Organizing Maps were originally developed for exploratory (unsupervised) data analysis by Kohonen (2002). The method is related to multidimensional scaling (MDS), however the relations between the observations are not explained by their interpoint distances but by their neighbours. This implementation is similar to that of Ballabio & Vasighi (2011) and covers four methods:

- Unsupervised Kohonen maps
- (Supervised) Counterpropagation ANNs
- Supervised Kohonen networks
- (Supervised) XY-fused networks

Note, that SKN needs slightly more memory since it extends the $X$ matrix by $k$ additional columns which is commonly the response $Y$ in $0-1$-form. Specifying the `sknsc` option to a value $t$ different than 1, makes that extension part into the $0-t$-form. The training data consist of a $N \times n$ matrix of $N$ observations of $n$ metric predictor variables. The supervised methods also need the specification of a nominal class variable with $k$ categories (levels) $Y$; which is not needed for the unsupervised Kohonen maps. The results of the algorithm depend not only on the seed of the random generator but also on a number of parameters which can be specified:

- number of neurons $s$ for each side of the map
- number of epochs $e$: number of times the objects are introduced in the net

The code contains a simple genetic algorithm (GA) for finding a promising pair $(s, e)$ from sets of $s$ and $e$ specified with input arguments 4 and 5. There are three ways to run that function:

1. As a simple model training run without cross validation for a specified pair $(s, e)$.
2. As a cross validation model training run for a specified pair $(s, e)$.
3. As a GA optimization for a specified (or default) set of $(s_i, e_i)$ pairs with a final training run for the pair $(s^*, e^*)$ that was found "best".

There is still some thought about using other definitions for the best pair $(s^*, e^*)$ after the optimization.

**Input:** **xtrn** $N \times n$ matrix of predictors for training

**ytrn** $N$ vector of response values (missing for unsupervised Kohonen networks) for training must be a (nominal scaled) class variable

**optn** the `optn` argument is 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.

**neus** vector of int numbers $s$ of neurons for GA (default: $[4, 6, 8, 10, 12, 14, 16]$)

**epos** vector of int numbers $e$ of epochs for GA (default: $[50, 100, 150, 200, 250, 300, 350]$)

**xtst** $Nt \times n$ matrix of predictors for test data

**ytst** $Nt$ vector of response values (missing for unsupervised Kohonen networks) for test data must be a (nominal scaled) class variable

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

| Option Name | Second Column | Meaning |
|---|---|---|
| "bound" | string | type of boundary condition |
| | "toro" | toroidal is default |
| | "norm" | normal |
| "block" | int | specifies fold value for blockwise CV |
| "chang" | string | |
| | "epo" | is default |
| | "obj" | |
| "chrom" | int | minimum number of chromosomes for GA, def=10 |
| "cvinc" | | CV sampling inside classes |
| "epo" | int | number $e$ epochs, def=100 |
| "eval" | int | maximum number evaluations, def=25 |
| "fold" | int | fold number of CV |
| "init" | string | method for initial values |
| | "rand" | random setting of initial values, is default |
| | "eigen" | initial values based on eigenvectors of largest eigenvalues |
| "lrini" | int | initial learning rate, def=0.5 |
| "lrend" | int | final learning rate, def=0.01 |
| "met" | string | assignment method only for CP-ANNs |
| | "maxw" | maximum output weight (def.) |
| | "maxd" | difference between highest and second highest weight |
| | "wthr" | output weight must be larger than 0.5 threshold |
| | "smoo" | assignment based on smoothing function |
| "neuro" | int | number $s$ neurons, def=10 |
| "nopr" | | no printed output |
| "opt" | | optimization with default `neus`, `epos` |
| "pcros" | real | probabilty for crossover GA, def=.5 |
| "pini" | int | print some input information |
| "pmap" | | print weight matrices |
| "pprd" | | print predicted values and residuals |
| "prbmu" | real | probability for mutation GA, def=.05 |
| "prin" | int | amount of printed output |
| "pwgt" | | print weight matrices |
| "rand" | int | specifies fold value for random CV |
| "rang" | string | specifying the random generator ("ranu": RAND Corporation, "kiss": keep-it-simple-stupid, "lecu": by L'Ecuyer ([**?**]) (same as in `jboot` and `sem` |
| "repfr" | int | maximum repetition-frequency for GA, def=25 |

| Option Name | Second Column | Meaning |
|---|---|---|
| "scal" | string | type of initial $X$ scaling |
| | "none" | no scaling, is default |
| | "cent" | centering (subtract mean) |
| | "scal" | divide by standard deviation |
| | "stan" | standardize by mean and standard deviation |
| "seed" | int | seed for random generator |
| "sknsc" | real | scale for the SKN extension, def=1 |
| "split" | int | specifies fold value for splitwise CV |
| "srang" | string | type of absolute range scaling $X$ in $[0, 1]$ |
| | "colw" | scale w.r.t. columnwise min and max |
| | "matw" | scale w.r.t. matrixwise min and max |
| "top" | string | topoidal condition |
| | "squ" | square is default |
| | "hex" | hexagonal |
| "type" | string | type of analysis method |
| | "koh" | (unsupervised) KOHonen maps |
| | "cpa" | (supervised) CounterPropagation ANNs |
| | "skn" | Supervised Kohonen Networks |
| | "xyf" | (supervised) XY-Fused networks |

**Output: gof** a vector of scalar results

**scal** a matrix with means, standard deviations, minimum, and maximum values of each variable which is useful for scaling test data;

**xwgt** the $s \times s \times n$ matrix of $X$ Kohonen weights;

**ywgt** is missing for unsupervised Kohonen maps, otherwise must contains the $s \times s \times k$ matrix of $Y$ weights; this is missing for unsuprevised Kohonen maps;

**nass** $s \times s$ matrix of class assignments of neurons

**tmtrn** $N \times 2$ matrix of the positions of observations (row and column coordinates) in the map

**cwtrn** $N \times k$ matrix of class weights this is missing for unsuprevised Kohonen maps;

**ptrn** $N \times 3$ matrix of

1. predicted values $\hat{y}$
2. input data $y$
3. residual $y - haty$

this is missing for unsuprevised Kohonen maps;

**mtrn** $k + 4 \times k$ matrix with

1. $k+1 \times k$ classification matrix where the rows correspond to levels of $\hat{y}$ and the columns to the levels of $y$
2. precision, specifity, and sensitivity of $k$ classes

**tmtst** $Nt \times 2$ matrix corresponding to `tmtrn` for test set

**cwtst** $Nt \times k$ matrix corresponding to `cwtrn` for test set

**ptst** $Nt \times 3$ matrix corresponding to `ptrn` for test set

**mtst** $k + 4 \times k$ matrix corresponding to `mtrn` for test set

**Restrictions:** 1. The current version assumes that there are no missing values in the `xtrn, ytrn, xtrn`, and `ytrn` input data arguments.

2. If specified, the length of the input vectors `neus` and `epos` must be equal to $2^K - 1$, by default for `K=3` the length is $7 = 2^3 - 1$.

3. Specifying `"hex"` topology and `"tor"` boundary requires an even number $s$ of neurons.

**Relationships:** mds(), pca(), pls(), svm(), nlreg()

**Examples:** 1. GA Optimization for the Iris Data: CP-ANN method, random

```
   options ls=68 ps=2000;
   options NOECHO;
#include "..\\tdata\\iris2.dat"
   options ECHO;

   print "*** Start Iris Data Processing ***";
   nr = nrow(X); nc = ncol(X);
   print "nr=",nr," nc=",nc;
   count = cons(3);
   for (i = 1; i <= nr; i++) count[Y[i]] += 1;
   print "Member Count=", count;
```

The following set of neuron and epochs numbers would also be the default setting when specifying the `opt` option:

```
   neus = [ 4 6 8 10 12 14 16 ];
   epos = [ 50 100 150 200 250 300 350 ];


   print "Top=square, Bound=toroid, neuro=11: Random CV, CVINC";
   optn = [ "type"      "cpa" ,
            "print"         3 ,
            "epo"          50 ,
            "neuro"        11 ,
            "top"       "squ" ,
            "bound"    "toro" ,
            "rand"          4 ,
            "cvinc"           ,
            "init"    "eigen" ,
            "scal"     "none" ,
```

```
        "seed"      12345 ,
        "lrini"        .5 ,
        "lrend"       .01 ,
        "chang"     "epo" ,
        "meth"     "maxw" ,
        "srang"     "col" ];
< gof,scal,xwgt,ywgt,nass,tmtrn,cwtrn,ptrn,mtrn >
           = som(X,Y,optn,neus,epos);


              ***************************
              SOM: Counterpropagation ANN
              ***************************

                   Level   Frequency
                       1          50
                       2          50
                       3          50


      Start of Genetic Optimizer (may take some time)
             Completed 3.125 % of Computations
             Completed 6.25 % of Computations
             Completed 9.375 % of Computations
             Completed 12.5 % of Computations
            Completed 15.625 % of Computations
             Completed 18.75 % of Computations
            Completed 21.875 % of Computations
       Completed 25 % computations after 49 seconds
       ........................................
       Completed 75 % computations after 148 seconds
            Completed 78.125 % of Computations
             Completed 81.25 % of Computations
            Completed 84.375 % of Computations
             Completed 87.5 % of Computations
            Completed 90.625 % of Computations
             Completed 93.75 % of Computations
            Completed 96.875 % of Computations
      Completed 100 % computations after 197 seconds
```

The following three tables list the triplets $(s, e, c)$ of neuron numbers $s$, epochs $e$, and fit criterion $c$ for the ten best solutions obtained with the Genetic Algorithm:

```
Best 10 Solutions in 4 Optimizations: Sizes
```

| | RANK_1 | RANK_2 | RANK_3 | RANK_4 | RANK_5 | RANK_6 |
|---|---|---|---|---|---|---|
| FOLD_1 | 12 | 8 | 8 | 10 | 4 | 12 |
| FOLD_2 | 12 | 16 | 16 | 10 | 10 | 6 |
| FOLD_3 | 14 | 6 | 4 | 10 | 6 | 8 |
| FOLD_4 | 6 | 8 | 10 | 10 | 8 | 8 |

Best 10 Solutions in 4 Optimizations: Sizes

| | RANK_7 | RANK_8 | RANK_9 | RANK_10 |
|---|---|---|---|---|
| FOLD_1 | 12 | 12 | 10 | 16 |
| FOLD_2 | 8 | 16 | 12 | 16 |
| FOLD_3 | 10 | 16 | 10 | 10 |
| FOLD_4 | 6 | 4 | 8 | 8 |

Best 10 Solutions in 4 Optimizations: Epochs

| | RANK_1 | RANK_2 | RANK_3 | RANK_4 | RANK_5 | RANK_6 |
|---|---|---|---|---|---|---|
| FOLD_1 | 350 | 50 | 150 | 200 | 100 | 150 |
| FOLD_2 | 50 | 300 | 250 | 150 | 100 | 100 |
| FOLD_3 | 100 | 250 | 150 | 200 | 200 | 50 |
| FOLD_4 | 250 | 50 | 350 | 150 | 100 | 100 |

Best 10 Solutions in 4 Optimizations: Epochs

| | RANK_7 | RANK_8 | RANK_9 | RANK_10 |
|---|---|---|---|---|
| FOLD_1 | 350 | 50 | 50 | 150 |
| FOLD_2 | 200 | 100 | 300 | 300 |
| FOLD_3 | 100 | 300 | 150 | 300 |
| FOLD_4 | 150 | 250 | 50 | 350 |

Best 10 Solutions in 4 Optimizations: Criterion

| | RANK_1 | RANK_2 | RANK_3 | RANK_4 |
|---|---|---|---|---|
| FOLD_1 | 0.990990991 | 0.973008511 | 0.966931428 | 0.966931428 |
| FOLD_2 | 1.000000000 | 1.000000000 | 1.000000000 | 0.990990991 |
| FOLD_3 | 0.953744314 | 0.916829110 | 0.916829110 | 0.908544509 |
| FOLD_4 | 0.965115335 | 0.965115335 | 0.962272580 | 0.962272580 |

Best 10 Solutions in 4 Optimizations: Criterion

| | RANK_5 | RANK_6 | RANK_7 | RANK_8 |
|---|---|---|---|---|
| FOLD_1 | 0.964230503 | 0.958153420 | 0.958153420 | 0.958153420 |
| FOLD_2 | 0.990990991 | 0.973008511 | 0.966931428 | 0.958153420 |

| | | | | |
|---|---|---|---|---|
| FOLD_3 | 0.908544509 | 0.908544509 | 0.900259909 | 0.900259909 |
| FOLD_4 | 0.953744314 | 0.916829110 | 0.916829110 | 0.916829110 |

Best 10 Solutions in 4 Optimizations: Criterion

| | RANK_9 | RANK_10 |
|---|---|---|
| FOLD_1 | 0.958153420 | 0.949375411 |
| FOLD_2 | 0.958153420 | 0.949375411 |
| FOLD_3 | 0.900259909 | 0.900259909 |
| FOLD_4 | 0.916829110 | 0.916829110 |

By averaging across the four cross validation folds and ten ranks we obtain the following two tables with the number of solutions (from the former table) and the mean criterion value:

Averaged Solutions (Sizes by Epochs) : Number

| | NEPO_50 | NEPO_100 | NEPO_150 | NEPO_200 | NEPO_250 | NEPO_300 |
|---|---|---|---|---|---|---|
| SIZE_4 | 0 | 1 | 1 | 0 | 1 | 0 |
| SIZE_6 | 0 | 1 | 1 | 1 | 2 | 0 |
| SIZE_8 | 4 | 2 | 1 | 1 | 0 | 0 |
| SIZE_10 | 1 | 2 | 3 | 2 | 0 | 1 |
| SIZE_12 | 2 | 0 | 1 | 0 | 0 | 1 |
| SIZE_14 | 0 | 1 | 0 | 0 | 0 | 0 |
| SIZE_16 | 0 | 1 | 1 | 0 | 1 | 3 |

Averaged Solutions (Sizes by Epochs) : Number

| | NEPO_350 |
|---|---|
| SIZE_4 | 0 |
| SIZE_6 | 0 |
| SIZE_8 | 1 |
| SIZE_10 | 1 |
| SIZE_12 | 2 |
| SIZE_14 | 0 |
| SIZE_16 | 0 |

Averaged Solutions (Sizes by Epochs) : Averaged Criterion

| | NEPO_50 | NEPO_100 | NEPO_150 | NEPO_200 |
|---|---|---|---|---|
| SIZE_4 | . | 0.964230503 | 0.916829110 | . |
| SIZE_6 | . | 0.973008511 | 0.916829110 | 0.908544509 |
| SIZE_8 | 0.940874366 | 0.935286712 | 0.966931428 | 0.966931428 |
| SIZE_10 | 0.958153420 | 0.945625450 | 0.951174493 | 0.937737969 |

| | | | | |
|---|---|---|---|---|
| SIZE_12 | 0.979076710 | . | 0.958153420 | . |
| SIZE_14 | . | 0.953744314 | . | . |
| SIZE_16 | . | 0.958153420 | 0.949375411 | . |

Averaged Solutions (Sizes by Epochs) : Averaged Criterion

| | NEPO_250 | NEPO_300 | NEPO_350 |
|---|---|---|---|
| SIZE_4 | 0.916829110 | . | . |
| SIZE_6 | 0.940972222 | . | . |
| SIZE_8 | . | . | 0.916829110 |
| SIZE_10 | . | 0.900259909 | 0.962272580 |
| SIZE_12 | . | 0.958153420 | 0.974572205 |
| SIZE_14 | . | . | . |
| SIZE_16 | 1.000000000 | 0.949878440 | . |

The final solution is selected by executing the SOM algorithm for the $(s, e)$ with the largest criterion in the table:

The index pair (7,5) is selected based on the largest average criterion 1. The final analysis corresponds to Size=16 and Epochs=250.

There is one misclassification for training this model:

Classification Table: Training

| | LEV_1 | LEV_2 | LEV_3 |
|---|---|---|---|
| LEV_1 | 50.00000000 | 0 | 0 |
| LEV_2 | 0 | 50.00000000 | 1.000000000 |
| LEV_3 | 0 | 0 | 49.00000000 |
| Missing | 0 | 0 | 0 |

| | |
|---|---|
| Error Rate (Misclassification). . . . . | 0.006667 |
| Correct Classification. . . . . | 0.9933 |
| Ratio of Nonassigned Samples. . . . . | 0 |

Precision, Sensitivity, and Specificity (Training)

| | LEV_1 | LEV_2 | LEV_3 |
|---|---|---|---|
| Precision | 1.000000000 | 0.980392157 | 1.000000000 |
| Sensitivity | 1.000000000 | 1.000000000 | 0.980000000 |
| Specificity | 1.000000000 | 0.990000000 | 1.000000000 |

Class Assignments of Neurons
**********

Dense Matrix (16 by 16)

| | NEUR_1 | NEUR_2 | NEUR_3 | NEUR_4 | NEUR_5 | NEUR_6 | NEUR_7 |
|---|---|---|---|---|---|---|---|
| NEUR_1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 |
| NEUR_2 | 1 | 1 | 1 | 1 | 2 | 2 | 2 |
| NEUR_3 | 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| NEUR_4 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| NEUR_5 | 3 | 3 | 3 | 2 | 3 | 2 | 2 |
| NEUR_6 | 3 | 3 | 2 | 3 | 3 | 2 | 3 |
| NEUR_7 | 3 | 3 | 3 | 2 | 2 | 2 | 3 |
| NEUR_8 | 3 | 3 | 3 | 2 | 2 | 2 | 3 |
| NEUR_9 | 3 | 3 | 3 | 2 | 2 | 2 | 3 |
| NEUR_10 | 3 | 3 | 3 | 2 | 2 | 2 | 2 |
| NEUR_11 | 3 | 3 | 3 | 2 | 2 | 2 | 2 |
| NEUR_12 | 1 | 3 | 3 | 2 | 2 | 2 | 2 |
| NEUR_13 | 1 | 1 | 1 | 1 | 2 | 2 | 2 |
| NEUR_14 | 1 | 1 | 1 | 1 | 1 | 1 | 2 |
| NEUR_15 | 1 | 1 | 1 | 1 | 1 | 1 | 2 |
| NEUR_16 | 1 | 1 | 1 | 1 | 1 | 2 | 2 |

.....

| | NEUR_15 | NEUR_16 |
|---|---|---|
| NEUR_1 | 1 | 1 |
| NEUR_2 | 1 | 1 |
| NEUR_3 | 2 | 1 |
| NEUR_4 | 2 | 2 |
| NEUR_5 | 3 | 3 |
| NEUR_6 | 3 | 3 |
| NEUR_7 | 3 | 3 |
| NEUR_8 | 3 | 3 |
| NEUR_9 | 3 | 3 |
| NEUR_10 | 3 | 3 |
| NEUR_11 | 3 | 3 |
| NEUR_12 | 1 | 1 |
| NEUR_13 | 1 | 1 |
| NEUR_14 | 1 | 1 |
| NEUR_15 | 1 | 1 |
| NEUR_16 | 1 | 1 |

```
  print "GOF=",gof;
```

```
   print "Scaling information=",scal;
   print "Kohonen X Weights=",xwgt;
   print "Kohonen Y Weights=",ywgt;
   print "Class Assignments of Neurons=",nass;
   print "Coordinates of Observations=",tmtrn;
   print "Class Weights=",cwtrn;
   print "Predicted values=",ptrn;
   print "Classification=",mtrn;
```

GOF=

| | 1 |
|---|---|
| ReturnCode | 0.00000 |
| Comp_Time | 203.00 |
| X_min | 0.100 |
| X_max | 7.9000 |
| Err_Train | 0.007 |
| Cor_Train | 0.99333 |
| Mis_Train | 0.00000 |
| Err_Test | . |
| Cor_Test | . |
| Mis_Test | . |
| unused | . |
| unused | . |

Scaling information=

| | V_1 | V_2 | V_3 | V_4 |
|---|---|---|---|---|
| Mean | 0.42870 | 0.44056 | 0.46746 | 0.45806 |
| StdDev | 0.23002 | 0.18161 | 0.29920 | 0.31760 |
| VarMin | 4.3000 | 2.0000 | 1.00000 | 0.10000 |
| VarMax | 7.9000 | 4.4000 | 6.9000 | 2.5000 |

Kohonen X Weights=

| | V_1 | V_2 | V_3 | V_4 |
|---|---|---|---|---|
| NEUR_01_01 | 0.09584 | 0.58656 | 0.07226 | 0.07519 |
| NEUR_01_02 | 0.11960 | 0.50155 | 0.10111 | 0.04168 |
| NEUR_01_03 | 0.18196 | 0.42481 | 0.09800 | 0.04057 |
| NEUR_01_04 | 0.15219 | 0.41481 | 0.06854 | 0.06100 |
| NEUR_01_05 | 0.18916 | 0.28583 | 0.18913 | 0.15951 |
| ... | ... | ... | ... | ... |
| NEUR_16_11 | 0.36287 | 0.36746 | 0.45351 | 0.49835 |
| NEUR_16_12 | 0.29908 | 0.67545 | 0.19206 | 0.22030 |

| | | | | |
|---|---|---|---|---|
| NEUR_16_13 | 0.25030 | 0.77468 | 0.09664 | 0.08831 |
| NEUR_16_14 | 0.22378 | 0.75083 | 0.09316 | 0.06399 |
| NEUR_16_15 | 0.19982 | 0.69711 | 0.06773 | 0.05360 |
| NEUR_16_16 | 0.11995 | 0.66316 | 0.02875 | 0.02851 |

Kohonen Y Weights=

| | LEV_1 | LEV_2 | LEV_3 |
|---|---|---|---|
| NEUR_01_01 | 1.00000 | 1e-021 | 8e-031 |
| NEUR_01_02 | 1.00000 | 2e-008 | 2e-031 |
| NEUR_01_03 | 1.00000 | 2e-008 | 1e-033 |
| NEUR_01_04 | 1.00000 | 2e-008 | 2e-031 |
| NEUR_01_05 | 0.68721 | 0.31279 | 3e-026 |
| ... | | | |
| NEUR_16_11 | 0.00437 | 0.99563 | 3e-015 |
| NEUR_16_12 | 0.74567 | 0.25433 | 1e-016 |
| NEUR_16_13 | 0.99999 | 0.00001 | 3e-030 |
| NEUR_16_14 | 1.00000 | 4e-025 | 5e-041 |
| NEUR_16_15 | 1.00000 | 9e-032 | 3e-048 |
| NEUR_16_16 | 1.00000 | 9e-036 | 5e-051 |

Coordinates of Observations=

| | Row | Column |
|---|---|---|
| 1 | 15 | 16 |
| 2 | 1 | 4 |
| 3 | 14 | 3 |
| 4 | 14 | 4 |
| 5 | 15 | 15 |
| ... | | |
| 145 | 8 | 13 |
| 146 | 6 | 15 |
| 147 | 7 | 8 |
| 148 | 9 | 11 |
| 149 | 6 | 13 |
| 150 | 4 | 13 |

Class Weights=

| | LEV_1 | LEV_2 | LEV_3 |
|---|---|---|---|
| 1 | 1.00000 | 1e-065 | 4e-097 |
| 2 | 1.00000 | 2e-008 | 2e-031 |
| 3 | 1.00000 | 3e-030 | 0.00000 |
| 4 | 1.00000 | 2e-029 | 4e-131 |

```
  5 |   1.00000    2e-035    2e-088
.....................................
145 |    2e-076    2e-023   1.00000
146 |    2e-140    8e-008   1.00000
147 |    3e-104   0.00016   0.99984
148 |    6e-053   0.00001   0.99999
149 |    2e-141    3e-007   1.00000
150 |    1e-009   0.03028   0.96972
```

```
    |  Predicted     Y_Data   Residual
-------------------------------------
  1 |          1          1          0
  2 |          1          1          0
  3 |          1          1          0
  4 |          1          1          0
  5 |          1          1          0
.....................................
120 |          2          3          1
.....................................
145 |          3          3          0
146 |          3          3          0
147 |          3          3          0
148 |          3          3          0
149 |          3          3          0
150 |          3          3          0
```

```
Classification=
             |     LEV_1     LEV_2     LEV_3
---------------------------------------------
       LEV_1 |    50.000   0.00000   0.00000
       LEV_2 |   0.00000    50.000   1.00000
       LEV_3 |   0.00000   0.00000    49.000
     Missing |   0.00000   0.00000   0.00000
   Precision |   1.00000   0.98039   1.00000
 Sensitivity |   1.00000   1.00000   0.98000
 Specificity |   1.00000   0.99000   1.00000
```

2. GA Optimization for the Iris Data: SKN method

```
print "SKN Top=square, Bound=toroid, neuro=11: Block CV, CVINC";
optn = [ "type"      "skn" ,
         "print"         3 ,
         "epo"          50 ,
         "neuro"        11 ,
```

```
        "top"       "squ" ,
        "bound"    "toro" ,
        "block"         4 ,
        "cvinc"           ,
        "init"    "eigen" ,
        "scal"     "none" ,
        "seed"      12345 ,
        "lrini"        .5 ,
        "lrend"       .01 ,
        "chang"     "epo" ,
        "meth"     "maxw" ,
        "srang"     "col" ];
< gof,scal,xwgt,ywgt,nass,tmtrn,cwtrn,ptrn,mtrn >
           = som(X,Y,optn,neus,epos);


             ******************************
             SOM: Supervised Kohonen Network
             ******************************

                   Level   Frequency
                       1          50
                       2          50
                       3          50


      Start of Genetic Optimizer (may take some time)
             Completed 3.125 % of Computations
             Completed 6.25 % of Computations
             Completed 9.375 % of Computations
             Completed 12.5 % of Computations
            Completed 15.625 % of Computations
             Completed 18.75 % of Computations
            Completed 21.875 % of Computations
       Completed 25 % computations after 51 seconds
       ........................................
       Completed 75 % computations after 148 seconds
            Completed 78.125 % of Computations
             Completed 81.25 % of Computations
            Completed 84.375 % of Computations
             Completed 87.5 % of Computations
            Completed 90.625 % of Computations
             Completed 93.75 % of Computations
            Completed 96.875 % of Computations
      Completed 100 % computations after 211 seconds
```

Best 10 Solutions in 4 Optimizations: Sizes

| | RANK_1 | RANK_2 | RANK_3 | RANK_4 | RANK_5 | RANK_6 |
|---|---|---|---|---|---|---|
| FOLD_1 | 10 | 6 | 14 | 16 | 4 | 4 |
| FOLD_2 | 14 | 10 | 8 | 10 | 6 | 10 |
| FOLD_3 | 6 | 6 | 6 | 4 | 10 | 10 |
| FOLD_4 | 8 | 6 | 12 | 4 | 10 | 16 |

Best 10 Solutions in 4 Optimizations: Sizes

| | RANK_7 | RANK_8 | RANK_9 | RANK_10 |
|---|---|---|---|---|
| FOLD_1 | 8 | 10 | 12 | 10 |
| FOLD_2 | 14 | 12 | 14 | 12 |
| FOLD_3 | 4 | 6 | 4 | 6 |
| FOLD_4 | 4 | 8 | 8 | 8 |

Best 10 Solutions in 4 Optimizations: Epochs

| | RANK_1 | RANK_2 | RANK_3 | RANK_4 | RANK_5 | RANK_6 |
|---|---|---|---|---|---|---|
| FOLD_1 | 350 | 250 | 250 | 150 | 50 | 150 |
| FOLD_2 | 50 | 250 | 250 | 150 | 100 | 100 |
| FOLD_3 | 200 | 350 | 150 | 50 | 50 | 100 |
| FOLD_4 | 100 | 250 | 100 | 50 | 200 | 300 |

Best 10 Solutions in 4 Optimizations: Epochs

| | RANK_7 | RANK_8 | RANK_9 | RANK_10 |
|---|---|---|---|---|
| FOLD_1 | 250 | 200 | 250 | 300 |
| FOLD_2 | 250 | 50 | 50 | 300 |
| FOLD_3 | 150 | 150 | 150 | 350 |
| FOLD_4 | 150 | 50 | 350 | 350 |

Best 10 Solutions in 4 Optimizations: Criterion

| | RANK_1 | RANK_2 | RANK_3 | RANK_4 |
|---|---|---|---|---|
| FOLD_1 | 0.981981982 | 0.964230503 | 0.958153420 | 0.958153420 |
| FOLD_2 | 0.990990991 | 0.973008511 | 0.973008511 | 0.966931428 |
| FOLD_3 | 0.970800845 | 0.916829110 | 0.916829110 | 0.908544509 |
| FOLD_4 | 0.970800845 | 0.965115335 | 0.953744314 | 0.948058804 |

Best 10 Solutions in 4 Optimizations: Criterion

| | RANK_5 | RANK_6 | RANK_7 | RANK_8 |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| FOLD_1 | 0.942800789 | 0.920447074 | 0.917159763 | 0.908612755 |
| FOLD_2 | 0.954954955 | 0.949375411 | 0.949375411 | 0.934253780 |
| FOLD_3 | 0.908544509 | 0.908544509 | 0.904605263 | 0.900259909 |
| FOLD_4 | 0.945216049 | 0.945216049 | 0.933398311 | 0.916829110 |

Best 10 Solutions in 4 Optimizations: Criterion

| | RANK_9 | RANK_10 |
|---|---|---|
| FOLD_1 | 0.908612755 | 0.908612755 |
| FOLD_2 | 0.908612755 | 0.908612755 |
| FOLD_3 | 0.880482456 | 0.872441520 |
| FOLD_4 | 0.916829110 | 0.916829110 |

Averaged Solutions (Sizes by Epochs) : Number

| | NEPO_50 | NEPO_100 | NEPO_150 | NEPO_200 | NEPO_250 | NEPO_300 |
|---|---|---|---|---|---|---|
| SIZE_4 | 3 | 0 | 4 | 0 | 0 | 0 |
| SIZE_6 | 0 | 1 | 2 | 1 | 2 | 0 |
| SIZE_8 | 1 | 1 | 0 | 0 | 2 | 0 |
| SIZE_10 | 1 | 2 | 1 | 2 | 1 | 1 |
| SIZE_12 | 1 | 1 | 0 | 0 | 1 | 1 |
| SIZE_14 | 2 | 0 | 0 | 0 | 2 | 0 |
| SIZE_16 | 0 | 0 | 1 | 0 | 0 | 1 |

Averaged Solutions (Sizes by Epochs) : Number

| | NEPO_350 |
|---|---|
| SIZE_4 | 0 |
| SIZE_6 | 2 |
| SIZE_8 | 2 |
| SIZE_10 | 1 |
| SIZE_12 | 0 |
| SIZE_14 | 0 |
| SIZE_16 | 0 |

Averaged Solutions (Sizes by Epochs) : Averaged Criterion

| | NEPO_50 | NEPO_100 | NEPO_150 | NEPO_200 |
|---|---|---|---|---|
| SIZE_4 | 0.933134701 | . | 0.909733276 | . |
| SIZE_6 | . | 0.954954955 | 0.908544509 | 0.970800845 |
| SIZE_8 | 0.916829110 | 0.970800845 | . | . |
| SIZE_10 | 0.908544509 | 0.928959960 | 0.966931428 | 0.926914402 |
| SIZE_12 | 0.934253780 | 0.953744314 | . | . |

```
SIZE_14   0.949801873      .            .            .
SIZE_16       .            .        0.958153420      .
```

Averaged Solutions (Sizes by Epochs) : Averaged Criterion

| | NEPO_250 | NEPO_300 | NEPO_350 |
|---|---|---|---|
| SIZE_4 | . | . | . |
| SIZE_6 | 0.964672919 | . | 0.894635315 |
| SIZE_8 | 0.945084137 | . | 0.916829110 |
| SIZE_10 | 0.973008511 | 0.908612755 | 0.981981982 |
| SIZE_12 | 0.908612755 | 0.908612755 | . |
| SIZE_14 | 0.953764415 | . | . |
| SIZE_16 | . | 0.945216049 | . |

The index pair (4,7) is selected based on the largest average criterion 0.981982. The final analysis corresponds to Size=10 and Epochs=350.

Classification Table: Training

| | LEV_1 | LEV_2 | LEV_3 |
|---|---|---|---|
| LEV_1 | 50.00000000 | 0 | 0 |
| LEV_2 | 0 | 49.00000000 | 0 |
| LEV_3 | 0 | 1.000000000 | 50.00000000 |
| Missing | 0 | 0 | 0 |

```
Error Rate (Misclassification). . . . . .   0.006667
Correct Classification. . . . . .     0.9933
Ratio of Nonassigned Samples. . . . . .          0
```

Precision, Sensitivity, and Specificity (Training)

| | LEV_1 | LEV_2 | LEV_3 |
|---|---|---|---|
| Precision | 1.000000000 | 1.000000000 | 0.980392157 |
| Sensitivity | 1.000000000 | 0.980000000 | 1.000000000 |
| Specificity | 1.000000000 | 1.000000000 | 0.990000000 |

3. GA Optimization for the Iris Data: XYF method

```
print "SKN: Top=square, Bound=toroid, neuro=11: Split CV: CVINC";
print "Misclassifications: lev=1: 0, lev=2: 3, lev=3: 3";
optn = [ "type"      "skn" ,
```

```
        "print"         3 ,
        "epo"          50 ,
        "neuro"        11 ,
        "top"       "squ" ,
        "bound"    "toro" ,
        "split"         4 ,
        "cvinc"           ,
        "init"    "eigen" ,
        "scal"     "none" ,
        "seed"      12345 ,
        "lrini"        .5 ,
        "lrend"       .01 ,
        "chang"     "epo" ,
        "meth"     "maxw" ,
        "srang"     "col" ];
< gof,scal,xwgt,ywgt,nass,tmtrn,cwtrn,ptrn,mtrn >
           = som(X,Y,optn,neus,epos);


                  *********************
                  SOM: XY-Fused Network
                  *********************

                    Level   Frequency
                        1          50
                        2          50
                        3          50


      Start of Genetic Optimizer (may take some time)
             Completed 3.125 % of Computations
             Completed 6.25 % of Computations
             Completed 9.375 % of Computations
             Completed 12.5 % of Computations
            Completed 15.625 % of Computations
             Completed 18.75 % of Computations
            Completed 21.875 % of Computations
       Completed 25 % computations after 103 seconds
       ........................................
       Completed 75 % computations after 247 seconds
            Completed 78.125 % of Computations
             Completed 81.25 % of Computations
            Completed 84.375 % of Computations
             Completed 87.5 % of Computations
            Completed 90.625 % of Computations
             Completed 93.75 % of Computations
            Completed 96.875 % of Computations
```

Completed 100 % computations after 319 seconds

Best 10 Solutions in 4 Optimizations: Sizes

| | RANK_1 | RANK_2 | RANK_3 | RANK_4 | RANK_5 | RANK_6 |
|---|---|---|---|---|---|---|
| FOLD_1 | 14 | 4 | 12 | 12 | 16 | 12 |
| FOLD_2 | 14 | 8 | 12 | 12 | 16 | 10 |
| FOLD_3 | 6 | 14 | 8 | 8 | 14 | 6 |
| FOLD_4 | 6 | 12 | 4 | 4 | 6 | 10 |

Best 10 Solutions in 4 Optimizations: Sizes

| | RANK_7 | RANK_8 | RANK_9 | RANK_10 |
|---|---|---|---|---|
| FOLD_1 | 16 | 4 | 14 | 16 |
| FOLD_2 | 14 | 14 | 4 | 6 |
| FOLD_3 | 16 | 12 | 8 | 14 |
| FOLD_4 | 12 | 6 | 16 | 10 |

Best 10 Solutions in 4 Optimizations: Epochs

| | RANK_1 | RANK_2 | RANK_3 | RANK_4 | RANK_5 | RANK_6 |
|---|---|---|---|---|---|---|
| FOLD_1 | 150 | 200 | 50 | 150 | 350 | 350 |
| FOLD_2 | 300 | 100 | 100 | 50 | 250 | 250 |
| FOLD_3 | 50 | 50 | 250 | 200 | 100 | 300 |
| FOLD_4 | 350 | 300 | 50 | 350 | 200 | 200 |

Best 10 Solutions in 4 Optimizations: Epochs

| | RANK_7 | RANK_8 | RANK_9 | RANK_10 |
|---|---|---|---|---|
| FOLD_1 | 200 | 150 | 250 | 150 |
| FOLD_2 | 150 | 50 | 250 | 150 |
| FOLD_3 | 50 | 50 | 50 | 150 |
| FOLD_4 | 200 | 50 | 150 | 50 |

Best 10 Solutions in 4 Optimizations: Criterion

| | RANK_1 | RANK_2 | RANK_3 | RANK_4 |
|---|---|---|---|---|
| FOLD_1 | 1.000000000 | 0.981981982 | 0.981981982 | 0.958153420 |
| FOLD_2 | 1.000000000 | 0.981981982 | 0.981981982 | 0.958153420 |
| FOLD_3 | 0.962272580 | 0.953744314 | 0.916829110 | 0.916829110 |
| FOLD_4 | 0.970800845 | 0.945216049 | 0.941682911 | 0.938921378 |

Best 10 Solutions in 4 Optimizations: Criterion

| | RANK_5 | RANK_6 | RANK_7 | RANK_8 |
|---|---|---|---|---|
| FOLD_1 | 0.958153420 | 0.958153420 | 0.958153420 | 0.955452494 |
| FOLD_2 | 0.958153420 | 0.958153420 | 0.958153420 | 0.925706772 |
| FOLD_3 | 0.916829110 | 0.908544509 | 0.900259909 | 0.900259909 |
| FOLD_4 | 0.930636777 | 0.908544509 | 0.908544509 | 0.904605263 |

Best 10 Solutions in 4 Optimizations: Criterion

| | RANK_9 | RANK_10 |
|---|---|---|
| FOLD_1 | 0.949375411 | 0.949375411 |
| FOLD_2 | 0.925706772 | 0.925706772 |
| FOLD_3 | 0.900259909 | 0.891975309 |
| FOLD_4 | 0.900259909 | 0.900259909 |

Averaged Solutions (Sizes by Epochs) : Number

| | NEPO_50 | NEPO_100 | NEPO_150 | NEPO_200 | NEPO_250 | NEPO_300 |
|---|---|---|---|---|---|---|
| SIZE_4 | 1 | 0 | 1 | 1 | 1 | 0 |
| SIZE_6 | 2 | 0 | 1 | 1 | 0 | 1 |
| SIZE_8 | 1 | 1 | 0 | 1 | 1 | 0 |
| SIZE_10 | 1 | 0 | 0 | 1 | 1 | 0 |
| SIZE_12 | 3 | 1 | 1 | 1 | 0 | 1 |
| SIZE_14 | 2 | 1 | 3 | 0 | 1 | 1 |
| SIZE_16 | 1 | 0 | 2 | 1 | 1 | 0 |

Averaged Solutions (Sizes by Epochs) : Number

| | NEPO_350 |
|---|---|
| SIZE_4 | 1 |
| SIZE_6 | 1 |
| SIZE_8 | 0 |
| SIZE_10 | 0 |
| SIZE_12 | 1 |
| SIZE_14 | 0 |
| SIZE_16 | 1 |

Averaged Solutions (Sizes by Epochs) : Averaged Criterion

| | NEPO_50 | NEPO_100 | NEPO_150 | NEPO_200 |
|---|---|---|---|---|
| SIZE_4 | 0.941682911 | . | 0.955452494 | 0.981981982 |
| SIZE_6 | 0.933438921 | . | 0.925706772 | 0.930636777 |

| | | | | |
|---|---|---|---|---|
| SIZE_8 | 0.900259909 | 0.981981982 | . | 0.916829110 |
| SIZE_10 | 0.900259909 | . | . | 0.908544509 |
| SIZE_12 | 0.946798437 | 0.981981982 | 0.958153420 | 0.908544509 |
| SIZE_14 | 0.939725543 | 0.916829110 | 0.950042909 | . |
| SIZE_16 | 0.900259909 | . | 0.924817660 | 0.958153420 |

Averaged Solutions (Sizes by Epochs) : Averaged Criterion

| | NEPO_250 | NEPO_300 | NEPO_350 |
|---|---|---|---|
| SIZE_4 | 0.925706772 | . | 0.938921378 |
| SIZE_6 | . | 0.908544509 | 0.970800845 |
| SIZE_8 | 0.916829110 | . | . |
| SIZE_10 | 0.958153420 | . | . |
| SIZE_12 | . | 0.945216049 | 0.958153420 |
| SIZE_14 | 0.949375411 | 1.000000000 | . |
| SIZE_16 | 0.958153420 | . | 0.958153420 |

The index pair (6,6) is selected based on the largest average criterion 1. The final analysis corresponds to Size=14 and Epochs=300.

Classification Table: Training

| | LEV_1 | LEV_2 | LEV_3 |
|---|---|---|---|
| LEV_1 | 50.00000000 | 0 | 0 |
| LEV_2 | 0 | 48.00000000 | 0 |
| LEV_3 | 0 | 2.000000000 | 50.00000000 |
| Missing | 0 | 0 | 0 |

| | |
|---|---|
| Error Rate (Misclassification). . . . . | 0.01333 |
| Correct Classification. . . . . | 0.9867 |
| Ratio of Nonassigned Samples. . . . . | 0 |

Precision, Sensitivity, and Specificity (Training)

| | LEV_1 | LEV_2 | LEV_3 |
|---|---|---|---|
| Precision | 1.000000000 | 1.000000000 | 0.961538462 |
| Sensitivity | 1.000000000 | 0.960000000 | 1.000000000 |
| Specificity | 1.000000000 | 1.000000000 | 0.980000000 |

## 5.6 Function `rad2deg`

d = rad2deg(r)

**Purpose:** Transforms the radians of an angle into degrees.

**Input:** A scalar, vector or matrix of angles measured in radians.

**Output:** A scalar, vector or matrix of angles measured in degrees.

**Restrictions:**
1. A missing value for $d$ is returned if the input $r$ contains string or complex data.
2. For any missing value in $r$ a missing value is returned in $d$.

**Relationships:** deg2rad()

**Examples:**

```
d1 = 45; d2 = 60;
print "d1=",d1," d2=",d2;
r1 = deg2rad(d1); r2 = deg2rad(d2);
print "r1=",r1," r2=",r2;
d1 = rad2deg(r1); d2 = rad2deg(r2);
print "d1=",d1," d2=",d2;
```

```
d1= 45 d2= 60
r1= 0.7854 r2= 1.0472
d1= 45.000 d2= 60.000
```

## 5.7 Function `smp`

< gof,beta,yprd,errt > = smp(data,tau,modl<,optn<,clas>>)

**Purpose:** The `smp` function implements the *stochastic matching pursuit* algorithm by Chen, Chu, and Weng (2010), for variable selection. As a side product, a *componentwise Gibbs sampler* is implemented.

Note, that with increasing value of $\tau$, the vector of the coefficients $\beta$ becomes more sparse (but at the cost of numeric stability). If the input argument `tau` is a vector of $t > 1$ positive values $\tau_j$, $k$ fold cross validation is used to select the result corresponding to the smallest validation error.

**Input: data** This is an $N \times nc$ data set which contains the response and $p$ predictor variables to which the model string refers to.

**tau** : That must either one positive scalar $\tau$ or a $t$ vector of positive $\tau_1, \ldots, \tau_t$ values.

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

**optn** This argument must be specified in form of a two column matrix where the first column defines the option as string value (in quotes) and the second column can be used for a numeric or string specification of the option. See table below for content. The option statement is not optional, since at least the columns of the censor variable and the group or strata variable must be specified.

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

| Option | Second Column | Meaning |
|---|---|---|
| "acc" | real | threshold for marginal probabilities for variable being active, must be in $(0, 1)$ (default=.5) |
| "bin" | int | number of initial "burn-in" iterations which are discarded, must be smaller than maxit, def=7000 |
| "birt" | real | probability for deciding birth vs. dead stage, must be in $(0, 1)$ (default=.5) |
| "block" | int | fold number $k$ for blockwise CV |
| "cgs" | | execute componentwise Gibbs sampler |
| "cloo" | | Leave-one-out CV |
| "maxit" | int | upper range for number of iterations (sweeps throgh the $p$ variables, def=10000); |
| "meth" | string | "cgs": componentwise Gibbs sampler<br>"smp": stochastic matching pursuit |
| "nost" | | no prior normalization of data |
| "peff" | | print effects table |
| "pini" | int | print some input information |
| "pobs" | int | print observationwise statistics: predicted $\hat{y}$, residuals |
| "print" | int | amount of printed output<br>=0 specifies no printed output, default is 2 |
| "rand" | int | fold number $k$ for random CV |
| "rho" | real | prior probability for excluding a variable, must be in $(0, 1)$ (default=.5) |
| "split" | int | fold number $k$ for splitwise CV |
| "vers" | | use method 2 for scaling $\beta$ |

A variable is accepted for the model if the number of selections $msel$ is sufficiently large compared to the maximum possible number,

$$msel >= acc * (maxit - bin)$$

That means, not only larger $\tau$ values, also the specification of a larger probability `acc` makes the coefficient vector $\beta$ more sparse and vice versa.

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

**beta** the $n \times 4$ matrix containing the

1. estimated coefficients of the linear model
2. the number the variable is selected
3. the `bsum` vector
4. 0 if the variable is neglected, 1 if the variable is accepted for the model

**yprd** the $N \times 3$ matrix with

1. response data $y_i$,

2. predicted values $\hat{y}_i$ ,
3. residuals $y_i - \hat{y}_i$

**errt** the $t \times 3$ matrix with

1. input value $\tau_k$
2. sum of validation errors in CV
3. number of selected variables for $\tau_k$

**Restrictions:**

1. Observations with missing values are ignored.
2. For large values of $\tau$ the algorithm is sensitive toward overgflows of the `exp()` function.

**Relationships:** lrallv(), lrforw(), reg(), varsel()

**Examples:** 1. Saturated design SMP:

```
  options NOECHO;
#include "..\\tdata\\SMP.dat"
  options ECHO;

  nr = nrow(SMP1); nc = ncol(SMP1);
  print "nr=",nr," nc=",nc;
  print "SMP1=",SMP1;
```

```
SMP1=
   |         1         2         3         4         5         6
---------------------------------------------------------------------
 1 |   1.00000   1.00000   1.00000  -1.00000  -1.00000  -1.00000
 2 |   1.00000  -1.00000  -1.00000  -1.00000  -1.00000  -1.00000
 3 |   1.00000   1.00000  -1.00000   1.00000   1.00000  -1.00000
 4 |   1.00000   1.00000  -1.00000   1.00000  -1.00000   1.00000
 5 |  -1.00000  -1.00000   1.00000   1.00000   1.00000   1.00000
 6 |  -1.00000  -1.00000   1.00000   1.00000   1.00000   1.00000
 7 |  -1.00000  -1.00000  -1.00000  -1.00000   1.00000  -1.00000
 8 |  -1.00000   1.00000   1.00000  -1.00000  -1.00000   1.00000
 9 |  -1.00000  -1.00000  -1.00000  -1.00000  -1.00000   1.00000
10 |   1.00000   1.00000   1.00000   1.00000  -1.00000   1.00000
11 |  -1.00000   1.00000  -1.00000   1.00000   1.00000  -1.00000
12 |   1.00000  -1.00000  -1.00000  -1.00000   1.00000   1.00000
13 |   1.00000   1.00000   1.00000   1.00000   1.00000  -1.00000
14 |  -1.00000  -1.00000   1.00000  -1.00000  -1.00000  -1.00000
.....................................................................

   |        19        20        21        22        23        24
---------------------------------------------------------------------
```

```
 1 |   1.00000  -1.00000  -1.00000  -1.00000   1.00000    133.00
 2 |  -1.00000   1.00000   1.00000  -1.00000  -1.00000    62.000
 3 |  -1.00000  -1.00000   1.00000   1.00000  -1.00000    45.000
 4 |   1.00000  -1.00000  -1.00000  -1.00000  -1.00000    52.000
 5 |   1.00000  -1.00000   1.00000   1.00000   1.00000    56.000
 6 |   1.00000   1.00000   1.00000  -1.00000  -1.00000    47.000
 7 |   1.00000   1.00000  -1.00000  -1.00000   1.00000    88.000
 8 |  -1.00000   1.00000   1.00000   1.00000  -1.00000    193.00
 9 |  -1.00000  -1.00000  -1.00000   1.00000   1.00000    32.000
10 |  -1.00000   1.00000  -1.00000  -1.00000   1.00000    53.000
11 |  -1.00000  -1.00000  -1.00000   1.00000   1.00000    276.00
12 |  -1.00000   1.00000   1.00000   1.00000   1.00000    145.00
13 |   1.00000   1.00000  -1.00000   1.00000  -1.00000    130.00
14 |   1.00000  -1.00000   1.00000  -1.00000  -1.00000    127.00
```

```
print "Stochastic Matching Pursuit";
modl = "24 = 1:23";
tau  = [ 5. 15. 25. 35. 45. ];
optn = [ "block"      5 ,
         "maxit"  10000 ,
         "bin"     7000 ,
         "time"       1 ,
         "slice"      1 ,
         "stand"        ,
         "peff"         ,
         "print"      3 ];
< gof,beta,yprd,tauerr > = smp(SMP1,tau,modl,optn);
print "GOF=",gof;
print "Beta=",beta;
print "Yprd=",yprd;
print "TAUerr=",tauerr;
```

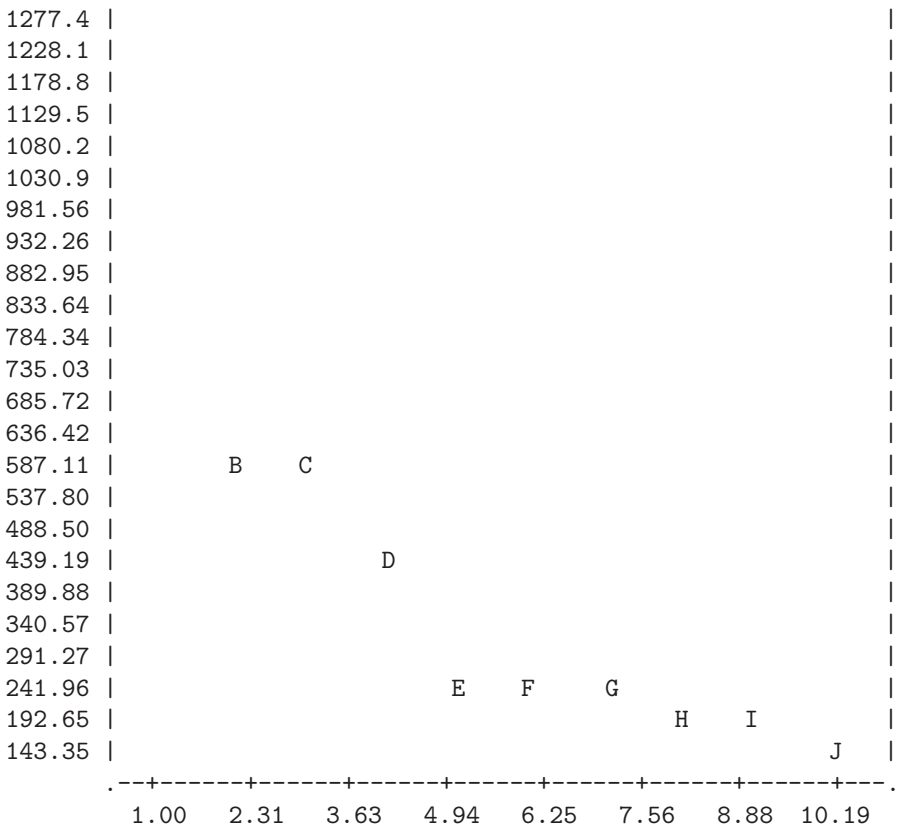
```
                    *****************
                    Model Information
                    *****************

               Number Valid Observations   14
               Response Variable         Y[24]
               N Independent Variables     23

          Stochastic Matching Pursuit (Method 1)

               Record Posterior             1
               Slice                        1
               Number Tau Values            5
```

```
Initial Sigma           1.5000
Number Total Iteration   10000
Number Discarded Itera    7000
Rho                     0.5000
Cross Validation         Block
Cross Validation Fold        5
Linear Model Without Intercept
```

```
CV selects tau=35 with smallest validation error.
```

As the authors show, only variable 14 is selected since it was more than 1500 = acc * (maxit-bin) times selected:

```
   Scree Plot of 10 Most Often Selected Effects
   ********************************************

        .--+------+------+------+------+------+------+------+---.
2658.0 |   A                                                     |
2608.7 |                                                         |
2559.4 |                                                         |
2510.1 |                                                         |
2460.8 |                                                         |
2411.5 |                                                         |
2362.2 |                                                         |
2312.9 |                                                         |
2263.5 |                                                         |
2214.2 |                                                         |
2164.9 |                                                         |
2115.6 |                                                         |
2066.3 |                                                         |
2017.0 |                                                         |
1967.7 |                                                         |
1918.4 |                                                         |
1869.1 |                                                         |
1819.8 |                                                         |
1770.5 |                                                         |
1721.2 |                                                         |
1671.9 |                                                         |
1622.6 |                                                         |
1573.2 |                                                         |
1523.9 |                                                         |
1474.6 |                                                         |
1425.3 |                                                         |
1376.0 |                                                         |
1326.7 |                                                         |
```




Rank Order of 10 Most Often Selected Effects (pacc=0.5)

| | | Variable | Percent | Nsel |
|---|---|---|---|---|
| A | 14* | X14 | 88.600000 | 2658 |
| B | 12 | X12 | 20.200000 | 606 |
| C | 19 | X19 | 19.300000 | 579 |
| D | 16 | X16 | 14.700000 | 441 |
| E | 8 | X8 | 8.433333 | 253 |
| F | 2 | X2 | 7.566667 | 227 |
| G | 4 | X4 | 7.266667 | 218 |
| H | 3 | X3 | 6.533333 | 196 |
| I | 22 | X22 | 6.400000 | 192 |
| J | 15 | X15 | 5.600000 | 168 |

Parameter Estimates

Dense Matrix (23 by 4)

| | Beta | Nsel | Bsum | Select |
|---|---|---|---|---|
| X1 | -8.96e-005 | 68.000000 | -0.2688093 | 0 |
| X2 | 0.0519519 | 227.00000 | 155.85582 | 0 |
| X3 | -0.0126010 | 196.00000 | -37.803119 | 0 |
| X4 | 0.0464352 | 218.00000 | 139.30552 | 0 |
| X5 | 0.0298856 | 136.00000 | 89.656835 | 0 |
| X6 | 0.0019868 | 66.000000 | 5.9603024 | 0 |
| X7 | -0.0349974 | 161.00000 | -104.99228 | 0 |
| X8 | 0.0724643 | 253.00000 | 217.39291 | 0 |
| X9 | 0.0283334 | 108.00000 | 85.000224 | 0 |
| X10 | 0.0041490 | 155.00000 | 12.446962 | 0 |
| X11 | 0.0355193 | 156.00000 | 106.55796 | 0 |
| X12 | -0.2932239 | 606.00000 | -879.67180 | 0 |
| X13 | 0.0016206 | 102.00000 | 4.8618416 | 0 |
| X14 | -2.6382934 | 2658.0000 | -7914.8802 | 1.0000000 |
| X15 | -0.0634560 | 168.00000 | -190.36808 | 0 |
| X16 | -0.2288411 | 441.00000 | -686.52326 | 0 |
| X17 | 0.0036511 | 55.000000 | 10.953364 | 0 |
| X18 | 0.0356133 | 157.00000 | 106.83989 | 0 |
| X19 | -0.2368603 | 579.00000 | -710.58085 | 0 |
| X20 | -0.0275371 | 144.00000 | -82.611162 | 0 |
| X21 | 0.0018563 | 87.000000 | 5.5690067 | 0 |
| X22 | 0.0892863 | 192.00000 | 267.85895 | 0 |
| X23 | 0.0047137 | 113.00000 | 14.141066 | 0 |

Except for observation 7, the predicted values are reasonably good:

Predicted Values of Training Data
**********

Dense Matrix (14 by 3)

| | Response | Predicted | Residual |
|---|---|---|---|
| 1 | 0.4512886 | 0.6331433 | -0.1818547 |
| 2 | -0.6091863 | -0.5838222 | -0.0253642 |
| 3 | -0.8631029 | -0.7667739 | -0.0963290 |
| 4 | -0.7585490 | -0.6388486 | -0.1197004 |
| 5 | -0.6988039 | -0.7167272 | 0.0179232 |
| 6 | -0.8332303 | -0.7621559 | -0.0710744 |
| 7 | -0.2208434 | 0.4631080 | -0.6839514 |
| 8 | 1.3474647 | 0.9521911 | 0.3952736 |
| 9 | -1.0572743 | -0.7996361 | -0.2576382 |

```
 10 |  -0.7436127 -0.7785754  0.0349627
 11 |   2.5871749  1.0496371  1.5375377
 12 |   0.6305239  0.7356399 -0.1051160
 13 |   0.4064798  0.6459441 -0.2394642
 14 |   0.3616710  0.5668758 -0.2052048


       ************************
       Cross Validation Errors
       ************************

N  NVSel  Percent         Tau        Error
1      1   4.3478  5.00000000  8.477419775
2      1   4.3478  15.0000000  7.530527707
3      1   4.3478  25.0000000  7.692043281
4      1   4.3478  35.0000000  6.361230431
5      1   4.3478  45.0000000  7.184214931


      Total Processing Time: 21
```

2. Saturated design CGS:

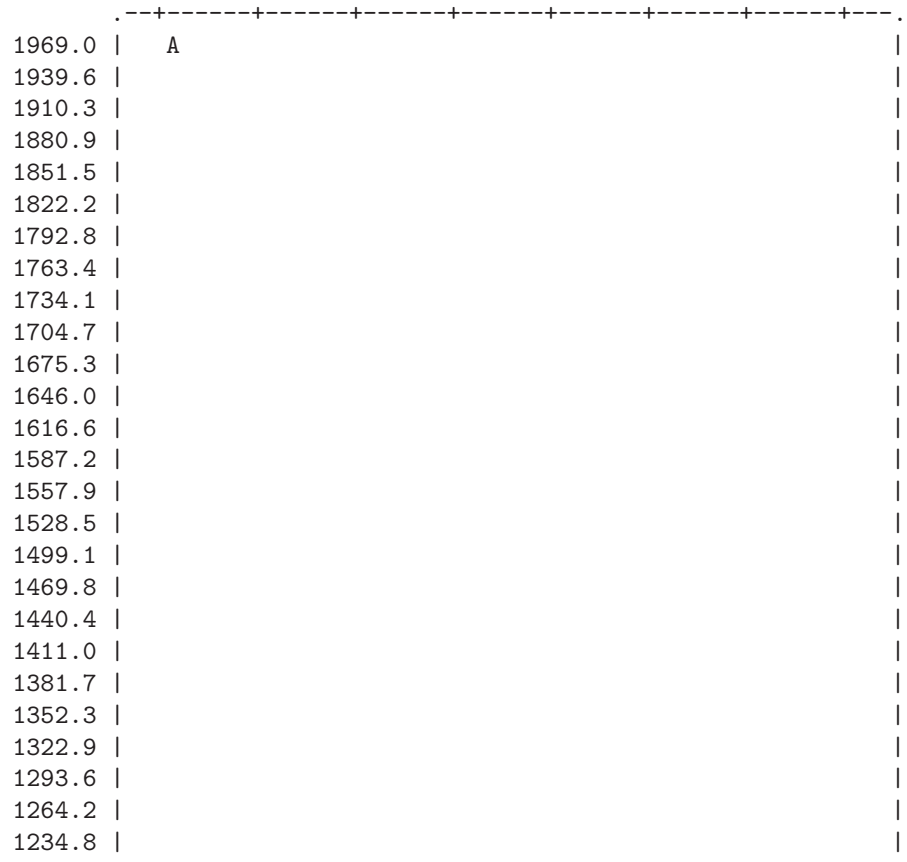
```
print "Componentwise Gibbs Sampler";
modl = "24 = 1:23";
tau  = [ 5. 15. 25. 35. 45. ];
optn = [ "block"      5 ,
         "maxit"  10000 ,
         "bin"     7000 ,
         "cgs"          ,
         "stand"        ,
         "peff"         ,
         "print"      3 ];
< gof,beta,yprd,tauerr > = smp(SMP1,tau,modl,optn);
print "GOF=",gof;
print "Beta=",beta;
print "Yprd=",yprd;
print "TAUerr=",tauerr;


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

                Number Valid Observations   14
                Response Variable         Y[24]
                N Independent Variables     23
```

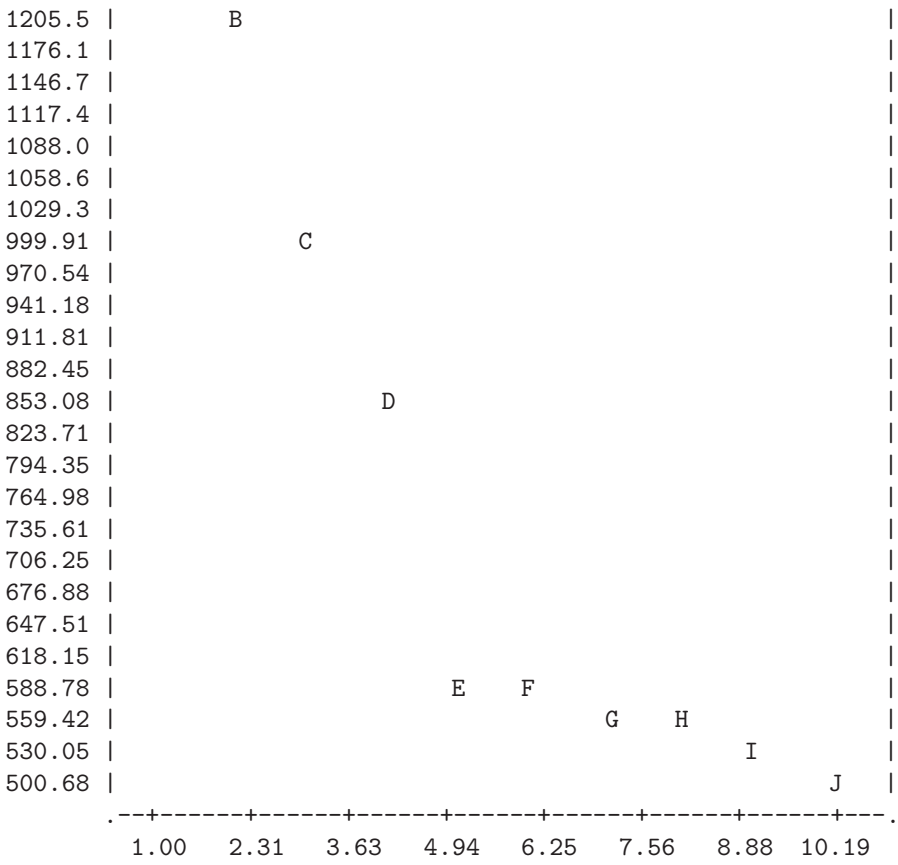
Stochastic Matching Pursuit (Method 1)
Componentwise Gibbs Sampler

| | |
|---|---|
| Number Tau Values | 5 |
| Initial Sigma | 1.5000 |
| Number Total Iteration | 10000 |
| Number Discarded Itera | 7000 |
| Rho | 0.5000 |
| Cross Validation | Block |
| Cross Validation Fold | 5 |
| Linear Model Without Intercept | |

CV selects tau=25 with smallest validation error.

Scree Plot of 10 Most Often Selected Effects
**********






Rank Order of 10 Most Often Selected Effects (pacc=0.5)

| | | Variable | Percent | Nsel |
|---|---|---|---|---|
| A | 14* | X14 | 65.633333 | 1969 |
| B | 16 | X16 | 40.366667 | 1211 |
| C | 12 | X12 | 33.433333 | 1003 |
| D | 19 | X19 | 28.133333 | 844 |
| E | 4 | X4 | 19.900000 | 597 |
| F | 2 | X2 | 19.766667 | 593 |
| G | 22 | X22 | 19.066667 | 572 |
| H | 23 | X23 | 18.933333 | 568 |
| I | 15 | X15 | 17.433333 | 523 |
| J | 8 | X8 | 16.200000 | 486 |

Parameter Estimates

Dense Matrix (23 by 4)

| | Beta | Nsel | Bsum | Select |
|---|---|---|---|---|
| X1 | -0.1571058 | 411.00000 | -471.31745 | 0 |
| X2 | 0.2625344 | 593.00000 | 787.60334 | 0 |
| X3 | 0.0020967 | 198.00000 | 6.2900876 | 0 |
| X4 | 0.1280028 | 597.00000 | 384.00839 | 0 |
| X5 | 0.0525859 | 291.00000 | 157.75784 | 0 |
| X6 | -0.1888193 | 415.00000 | -566.45805 | 0 |
| X7 | -0.1019207 | 274.00000 | -305.76214 | 0 |
| X8 | 0.2045907 | 486.00000 | 613.77200 | 0 |
| X9 | 0.1491082 | 429.00000 | 447.32449 | 0 |
| X10 | -0.0132063 | 211.00000 | -39.619001 | 0 |
| X11 | 0.0958576 | 268.00000 | 287.57281 | 0 |
| X12 | -0.4143676 | 1003.0000 | -1243.1028 | 0 |
| X13 | 0.0196076 | 185.00000 | 58.822707 | 0 |
| X14 | -1.7723392 | 1969.0000 | -5317.0177 | 1.0000000 |
| X15 | -0.2016056 | 523.00000 | -604.81671 | 0 |
| X16 | -0.7365879 | 1211.0000 | -2209.7638 | 0 |
| X17 | -0.0312824 | 190.00000 | -93.847087 | 0 |
| X18 | 0.1418779 | 340.00000 | 425.63384 | 0 |
| X19 | -0.3506513 | 844.00000 | -1051.9539 | 0 |
| X20 | -0.0348370 | 200.00000 | -104.51109 | 0 |
| X21 | 0.0029902 | 213.00000 | 8.9706724 | 0 |
| X22 | 0.2337191 | 572.00000 | 701.15721 | 0 |
| X23 | 0.2482897 | 568.00000 | 744.86900 | 0 |

Again, observation 7 seems to be an outlier:

Predicted Values of Training Data
**********

Dense Matrix (14 by 3)

| | Response | Predicted | Residual |
|---|---|---|---|
| 1 | 0.4512886 | 0.3378881 | 0.1134005 |
| 2 | -0.6091863 | -0.3699232 | -0.2392632 |
| 3 | -0.8631029 | -0.7296452 | -0.1334577 |
| 4 | -0.7585490 | -0.4575544 | -0.3009946 |
| 5 | -0.6988039 | -0.5333941 | -0.1654099 |
| 6 | -0.8332303 | -0.7519588 | -0.0812715 |
| 7 | -0.2208434 | 0.0615475 | -0.2823909 |
| 8 | 1.3474647 | 0.9844181 | 0.3630466 |

```
  9 |  -1.0572743 -0.7679589 -0.2893154
 10 |  -0.7436127 -0.6244541 -0.1191586
 11 |   2.5871749  1.4622519  1.1249230
 12 |   0.6305239  0.5576659  0.0728579
 13 |   0.4064798  0.5177457 -0.1112659
 14 |   0.3616710  0.3133715  0.0482995
```

```
**********************
Cross Validation Errors
**********************
```

| N | NVSel | Percent | Tau | Error |
|---|---|---|---|---|
| 1 | 1 | 4.3478 | 5.00000000 | 9.404049114 |
| 2 | 1 | 4.3478 | 15.0000000 | 8.712110536 |
| 3 | 1 | 4.3478 | 25.0000000 | 8.682938516 |
| 4 | 1 | 4.3478 | 35.0000000 | 8.864343280 |
| 5 | 1 | 4.3478 | 45.0000000 | 8.964943882 |

Total Processing Time: 27

# 6 Illustration:

## 6.1 CMAT Code for Estimating the Curvilinear Model $K = 2$

The following is a listing for the $K = 2$ curvilinear bidimensional regression model in complex algebra:

```
print "Original example by Nakaya (1997): N=19";
y = [0.105 0.88 1.057 1.262 2.316 3.293 -0.743 -0.964
    -1.249 -0.179 0.832 2.217 -0.352 1.434 -0.611 0.497
     2.333 3.13 2.296 ,
     0.515 -0.122 1.109 0.463 -0.037 -0.149 2.092 1.328
     2.322 -0.689 -0.926 -1.092 -1.571 -2.092 -2.908 -2.989
     1.214 1.307 2.232 ]‘;
x = [0.041 0.445 0.773 0.747 1.328 2.007 -0.316 -0.497
    -0.629 -0.124 0.309 0.847 -0.368 0.44 -0.964 -0.379
     1.83 2.184 1.763 ,
     0.536 -0.047 0.83 0.309 -0.687 -1.452 1.457 1.12
     1.632 -0.25 -0.611 -0.959 -0.503 -1.543 -1.707 -1.763
    -0.134 -0.752 0.72 ]‘;


nr = nrow(x);
complex z[nr];
z = x[,1] + x[,2] * 1.i;
print "Z=", z;
d1 = d2 = d3 = cons(nr,2);
d1 = [ 19 # 1. 19 # 0. ,
       19 # 0. 19 # 1. ]‘;
d2[,1] =  x[,1] |> x[,2];
d2[,2] = -x[,2] |> x[,1];
print "D2=",d2;


zz = z .* z;  zr = (real)zz; zi = (imag)zz;
d3[,1] = zr |> -zi;
d3[,2] = zi |>  zr;
print "D3=",d3;


d = d1 -> d2 -> d3;
print "D=",d;


cov = d‘ * d;
print "COV=",cov;
```

```
yy = y[,1] |> y[,2];
rhs = d‘ * yy;
print "RHS=",rhs;
est = cov \ rhs;
print "Est=",est;


yprd = d * est;
print "Yprd=",yprd;
sse = ssq(yprd - yy);
print "SSE=",sse;


ymu = yy[+] / nrow(yy);
ssy = ssq(yy - ymu);
print "SSY=",ssy;


rsqr = 1. - sse / ssy;
print "rsqr=",rsqr;
```