MULTIVAR
a GAUSS Library for Multivariate Analysis

written / collected by

Rainer Schlittgen

1999

Institut for Statistics and Econometry
University of Hamburg

e-mail: schlitt@hermes1.econ.uni-hamburg.de

Only programs are listed here which may be used directly but auxiliary programs aren't. See the multivar.lcg file for a complete listing.

Plots of Multivariate Data	
ANDPLOT	Andrews curves
BIPLOT	Gabriel's biplot
BOXPLOT	boxplot
CHULLPLOT	scatter diagram with of convex hulls
GROUPXY	scatter diagram with different symbols for each group
INFLUENCEPLOT	plot of influence function
KDENSPLOT	plot of kernel density estimate
LABELXY	scatter diagram with labels
NCHECKPLOT	QQ-diagram to check (multivariate) normality
SCMATPLOT	scatterplot matrix
STARPLOT	starplot

Moments and Manipulation of Data Matricies	
CENTER	centering of data
CORMV	correlation matrix allowing for missings
CORPCHOR	polychoric correlation
CORPOLY	correlation matrix with different types of variables
CORPSER	polyserial correlation
COVMV	covariance matrix allowing for missings
DELTA	sum of squared differences of two matricies
DISTE	Euclidean distance
DISTMAHAL	Mahalanobis distance
DISTMINK	Minkowski distance
MEANMV	mean allowing for missing
MEDIANMV	median allowing for missing
ORD2NORM	transforming ordered categorial data into normal variates
PROXMVE	one step minimum volume estimator
STANDARD	standardization of a data matrix
STDMV	standard deviation allowing for missings
TR	trace of a matrix
UNIMVE	univariate minimum volume estimator

	Cluster Analysis
AGGLOM	agglomerative clustering
DENDRO	dendrogram
GROUP	determing cluster membership
KMEANS	k-means clustering
	Discriminant Analysis
LINDISCR	linear discriminant analysis
LINDISCR2	linear discriminant analysis for two groups
LOGREG	logistic regression
	Factor Analysis
FACTORML	maximum likelihood factor analysis
FACTORPC	principal factor analysis
VARIMAX	varimax rotation
	Generalised Linear Models
GLIM	estimation of glm
	Multidimensonal Scaling
MDSMETR	metric mds
MDSNONM2	nonmetric two dimensional mds
PROCRUST	procrustes transformation
TWINPLOT	plot of two data matricies
	Principal Components Analysis
PRINCOMPC	principal components analysis
	Regression Trees
GROWTREE	growing a regression tree
PLOTTREE	plot of a regression tree
SURROGATE	assigning observations with missings to nodes

AGGLOM

- **Purpose**

  Agglomerative Clustering of a distance matrix

- **Format**

  $\{o, lev\} = \text{AGGLOM}(d, m);$

- **Input**

d	(n, n) distance matrix
m	scalar, gives the clustering method: 1 : single linkage, 2 : complete linkage 3 : average linkage, 4 : centroid method

- **Output**

o	$(n-1, n)$ matrix, rows give the newly built clusters
lev	$(n-1, 1)$ vector, the distances at which the clusters are glued together

- **Source**

  `cluster.src`

ANDPLOT

- **Purpose**

  Plotting Andrews curves

- **Format**

  ANDPLOT(x, tit);

- **Input**

x	(n,p)-matrix, data
tit	string, the title of the plot, " " for no title

- **Output**

  none

- **Source**

  `mvaplots.src`

BIPLOT

- **Purpose**

  Performs a Biplot

- **Format**

  BIPLOT($typ, x, l, vl, stan, flipx, flipy, varcal$);

- **Input**

typ	scalar, type of biplot: 1 = GH-Biplot , 2 = JK-Biplot 4 = RV-Plots , else = SQ-Plots
x	(n, p)-matrix, data
l	string array, labels of objects, or " "
vl	string array, labels of variables, or " "
$stan$	flag; if =1, the data matrix ist standardized
$flipx$	flag; if =1, reverse x-axis
$flipy$	flag; if =1, reverse y-axis
$varcal$	scalar for scaling the position of objects in plot

- **Output**

  none

- **Remarks**

  adapted from a program by R. Schnell

- **Source**

  `pcompana.src`

BOXPLOT

- **Purpose**

  Plotting (multiple) Box and Whisker Plot(s)

- **Format**

  BOXPLOT(x, p, tit);

- **Input**

x	(n, p)-matrix, data
p	$0 < p < 0.5$ proportion of values displayed outside the whiskers
tit	string, the title of the plot or " "

- **Output**

  none

- **Source**

  `mvaplots.src`

CENTER

- **Purpose**

  Centering of a matrix by subtraction of column means.
  Allowing for missing values.

- **Format**

  $\{xc\} = \text{CENTER}(x);$

- **Input**

x	(n, k) matrix

- **Output**

xc	(n, k) matrix, centered matrix

- **Source**

  `rmoment.src`

CHULLPLOT

- **Purpose**

  Plot of a scatter diagram together with the outer convex hull and the convex hull containing at least 50% of the data

- **Format**

  CHULLPLOT(x);

- **Input**

x	$(n, 2)$-matrix, data

- **Output**

  none

- **Remarks**

  uses CHULLPOINT

- **Source**

  `convhull.src`

CORMV

- **Purpose**

  Computes the correlation matrix from a data matrix.
  Allowing for missing values.

- **Format**

  $\{c\} = \mathrm{CORMV}(x);$

- **Input**

  x (n, k) matrix

- **Output**

  c (k, k) correlation matrix

- **Source**

  `rmoment.src`

CORPCHOR

- **Purpose**

  Computes the polychoric correlation coefficient of two variables with ordered categories

- **Format**

  $\{r\} = \text{CORPCHOR}(dat);$

- **Input**

x	$(n, 2)$ matrix

- **Output**

r	scalar, polychoric correlation coefficient

- **Remarks**

  The two step estimator is computed

- **Source**

  `rmoment.src`

CORPOLY

- **Purpose**

  Computes matrix of product-moment, polyserial or polychoric correlation coefficients for mixed data

- **Format**

  $\{r\} = \text{CORPOLY}(x, ind);$

- **Input**

x	(n,p) matrix
ind	$(p,1)$ vector with 1 and 0; the columns with 1 are dealt with as ordered categorial

- **Output**

r	(p,p) matrix of correlation coefficients

- **Remarks**

  uses cormv, corpchor,corpser

- **Source**

  `rmoment.src`

CORPSER

- **Purpose**

  Computes the polyserial correlation coefficient of a variable with ordered cartegories and a continuous one

- **Format**

  $\{c\} = \text{CORPSER}(x, ind);$

- **Input**

x	$(n, 2)$ matrix
ind	$(2, 1)$ vector with one 1 and one 0; the column with the 1 is dealt with as categorial

- **Output**

r	scalar, polyserial correlation coefficient

- **Remarks**

  The two step estimator is computed.

- **Source**

  `rmoment.src`

COVMV

- **Purpose**

  Computes the covariance matrix of the columns of a matrix.
  Allowing for missing values.

- **Format**

  $\{c\} = \mathrm{COVMV}(x);$

- **Input**

  x (n, k) matrix

- **Output**

  c (k, k) covariance matrix

- **Source**

  `rmoment.src`

DELTA

- **Purpose**

  Computes the sum of squared differences of two matricies.

- **Format**

  $\{s\} = \text{DELTA}(x, y);$

- **Input**

x	(n, k) matrix,
y	(n, k) matrix

- **Output**

s	scalar, sum of squared differences of x and y.

- **Source**

  `rmoment.src`

DENDRO

- **Purpose**

  Plotting a dendrogram

- **Format**

  DENDRO($o, lev, lab, space$);

- **Input**

o	$(n-1, n)$ matrix of cluster history, output of AGGLOM
lev	$(n-1, 1)$ vector, distances at which the clusters are built, output of AGGLOM
lab	$(n-1, 1)$ string array, object labels
$space$	scalar, determines space for the labels

- **Output**

  none

- **Remarks**

  uses DENDROCO

- **Source**

  `cluster.src`

DISTE

- **Purpose**

  Matrix of euclidean distances from a data matrix

- **Format**

  $\{d\} = \mathrm{DISTE}(x);$

- **Input**

x	(n, p) matrix, the data

- **Output**

d	(n, n) matrix of distances

- **Remarks**

  uses DISTMINK

- **Source**

  `distance.src`

DISTMAHAL

- **Purpose**

  Matrix of Mahalanobis distances from data matrix

- **Format**

  $\{d\} = \mathrm{DISTMAHAL}(x);$

- **Input**

x	(n, p) matrix, the data

- **Output**

d	(n, n) matrix of distances

- **Source**

  `distance.src`

DISTMINK

- **Purpose**

  Matrix of Minkowski distances from data matrix

- **Format**

  $\{d\} = \text{DISTMINK}(x, q);$

- **Input**

x	(n, p) matrix, the data
q	scalar, the exponent of the metric

- **Output**

d	(n, n) matrix of distances

- **Source**

  `distance.src`

FACTORML

- **Purpose**

  Maximum likelihood factor analysis.

- **Format**

  $\{L, com, psi\} = \text{FACTORML}(S, k);$

- **Input**

S	(p,p) covariance or correlation matrix
k	scalar, number of factors

- **Output**

L	(p,k) matrix of factor loadings
com	$(p,1)$ vector, the communalities
psi	$(p,1)$ vector, the specific variances

- **Remarks**

  Adapted from Kuesters, U. and Arminger, G. (1989):
  Programmieren in GAUSS, Stuttgart: Gustav Fischer Verlag

- **Source**

  `factor.src`

FACTORPC

- **Purpose**

  Principal factor method for exploratory factor analysis

- **Format**

  $\{L, lambda, h, com\}$ = FACTORPC($x, k, pc, flag$);

- **Input**

x	(n, p) data matrix
pc	scalar, if $= 1$, the principal factor method is used otherwise the principal components method is used
k	scalar, number of factors
$flag$	scalar, if $= 1$, the correlation matrix is used

- **Output**

L	(p, k) matrix of factor loadings
$lambda$	$(k, 1)$ vector, the eigenvalues
h	$(k, 1)$ vector, cumulative proportion of variance explained
com	$(p, 1)$ vector, the communalities

- **Remarks**

  to obtain the specific variances, compute psi = diag(covmv(x)-h

- **Source**

  `factor.src`

GLIM

- **Purpose**

  Fitting univariate Generalized Linear Models

- **Format**

  $\{b, fv, df, cor, se, chisq, dev\} = \text{GLIM}(y, x, pw, model);$

- **Input**

y	$(n, 1)$ vector, dependent variable
x	(n, p) matrix of independent variables
pw	$(n, 1)$ vector, weights (if y contains proportions, then pw is the number of observations)
$model$	string, one of the implemented models: "*logit*" logistic regression "*probit*" probit regression "*cloglog*" complementary loglog "*poisson*" poisson regression "*normal*" ordinary least squares "*gamma*" gamma regression

- **Output**

b	vector of parameter estimates
fv	fitted values
df	degrees of freedom for chisq
cor	correlation matrix of parameter estimates
se	standard errors of parameter estimates
$chisq$	Pearson chi-square (scalar)
dev	deviance of fitted model

- **Remarks**

  adapted form Lindsay's code published in: Decarli,A., Francis,B.J., Gilchrist,R., and Seeber,G.U.H. (eds.): Statistical Modelling; Lecture Notes in Statistics 57, 1990 New York : Springer

- **Source**

  `glm.src`

GROUP

- **Purpose**

  Shows clusters at a given distance level resulting from `agglom`

- **Format**

  $\{g\} = \text{GROUP}(o, lev, d);$

- **Input**

o	$(n-1, n)$ matrix of cluster history, output of AGGLOM
lev	$(n-1, 1)$ vector, distances at which the clusters are built, output of AGGLOM
d	scalar, the distance level

- **Output**

o	$(n-1, n)$ matrix, rows give the newly built clusters
lev	$(n-1, 1)$ vector, the distances at which the clusters are glued together
g	(k, n) matrix, each row represents one cluster; rows are filled up with missings

- **Source**

  `cluster.src`

GROUPXY

- **Purpose**

  Scatterplot of several datasets with labels indicating group membership

- **Format**

  GROUPXY(dat, lab);

- **Input**

dat	$(n, 3)$ matrix 1st column: group indicator (numeric) 2nd and 3rd column: x and y coordinates for n objects
lab	$(g, 1)$ character vector, labels for groups, or " "

- **Output**

  none

- **Source**

  `mvaplots.src`

GROWTREE

- **Purpose**

  growing a regression tree allowing for right censoring

- **Format**

  $\{tr, node\} = \text{GROWTREE}(dat, cens, alpha, minnod, adj, criterion, arrange);$

- **Input**

dat	$(n, p-1)$- data matrix $dat[., 1]$ dependent variable $dat[., 2]$: if cens=1: observation indicator; 1=observed, 0=censored if cens=0: first independent variable $dat[., 3 : (p-1)]$ independent variables
cens	scalar, = 1 if censoring is present, else no censoring
alpha	scalar, level of significane for the tests or limit for proportion of impurity
minnod	scalar, minimal number of observations in a node so that it can be splitted.
adj	scalar, alpha-adjustment: = 0 no adj., = 1 adj. using conditional approach, = 2 adj. using the method of Lausen/Schumacher
criterion	string with criterion used for splitting: "*lrtest*" = log rank test (also for censored data) "*wtest*" = wilcoxon rank sum test (also for censored data) "*ttest*" = t-test "*chitest*" = chi squared test (for binary data) "*ssreduc*" = reduction of sums of squares "*gini*" = reduction of Gini's impurity measure (for binary data)
arrange	scalar, arrangement of daughter nodes: = 0: greater values of covariable on the right hand side = 1: right hand side node has greater mean value of dependent variable = 2 : right hand side node has greater median value of dependent variable

- **Output**

tr	$(k, 10)$-matrix: $tr[., 1]$ node number $tr[., 2]$ number of variable used for splitting $tr[., 3]$ value at which the split is performed $tr[., 4]$ value of split criterion $tr[., 5]$ p-value of test or proportion of impurity reduction $tr[., 6]$ number of observations in node $tr[., 7]$ number of observations in left daughter node $tr[., 8]$ number of observations in right daugther node $tr[., 9]$ =1 : daughter nodes were switched to get greater value of criterion for node on the the right hand side (results in $>=$ in plottree) $= 0$: otherwise (results in $<$ in plottree) $tr[., 10]$ value of criterion for node members
$node$	$(n, 1)$-vector with highest node number the observation belongs to

- **Source**

  `tree.src`

INFLUENCEPLOT

- **Purpose**

  Determines and plots empirical influence function for an estimator

- **Format**

  $\{e\} = \text{INFLUENCEPLOT}(x, \&est, tit);$

- **Input**

x	(n,p)-matrix, data
est	procedure, the estimator
tit	string, the title of the plot or " "

- **Output**

e	$(n,1)$ vector, the influence values;
a plot is also shown.	

- **Remarks**
  Possible estimators:
  `cormv`: correlatioon matrix
  `covmv`: covariance matrix
  `meanmv`: mean of a vector
  `medianmv`: median of a vector
  `stdmv`: standard deviation of a vector

- **Source**

  `mvaplots.src`

KDENSPLOT

- **Purpose**

  Plot of Kernel Density Estimator

- **Format**

  KDENSPLOT($x, h, a, \&kern$);

- **Input**

x	$(n, 1)$-vector, data or $(n, 2)$ matrix, data with first column as group variable
h	scalar $= -1$ for automatic band selection or scalar > 1, equal bandwidth of kernel estimators or $(g, 1)$ vector of bandwidth' for each group
a	$a > 0$ integer, number of points for which the density is computed
$kern$	procedure, the kernel

- **Output**

  none

- **Remarks**

  Possible kernels:
  `unifkern`: uniform kernel
  `normkern`: normal kernel
  `epanechkern`: epanechnikov kernel
  `triangkern`: triangel kernel

- **Source**

  `mvaplots.src`

KMEANS

- **Purpose**

  K-means clustering of a data matrix

- **Format**

  {ind,c} = KMEANS(x, d);

- **Input**

x	(n,p) data matrix
d	scalar, number of clusters

- **Output**

ind	$(n,1)$ vector, indicating the cluster the row of the data matrix belongs to
c	(d,p) matrix, the centers of the clusters

- **Remarks**
  the start configuration is built from ordered euclidean distances from global mean

- **Source**

  `cluster.src`

LABELXY

- **Purpose**

  Scatterdiagram with labels

- **Format**

  LABELXY($dat, label$);

- **Input**

dat	$(n, 2)$-vector, data
label	$(n, 1)$ stringarray, the labels of the objects, or " "

- **Output**

  none

- **Source**

  `mvaplots.src`

LINDISCR

- **Purpose**

  Performs linear discriminant analysis.

- **Format**

  $\{a, lambda, m, spl, status, tb\} = \text{LINDISCR}(dat, prior);$

- **Input**

dat	$(n, p+1)$ matrix, data with first column as group indicator
prior	$(g, 1)$ vector, prior probabilities for the groups, or scalar $= 0$ for no priors (ML-method) or scalar $= 1$ for priors determined from group sizes

- **Output**

a	(p, s) matrix, disciminant funktions
lambda	$(s, 1)$ vector, eigenvalues
m	(p, g) matrix, the group centers
spl	(p, p) pooled covariance matrix
status	$(g, 4)$ matrix with information about groups $status[., 1]$: group number $status[., 2]$: # observations in group $status[., 3]$: proportion with resp. to all obs. $status[., 4]$: priors
tb	(g, g) matrix, classification results (true groups rowwise)

- **Source**

  `discrimi.src`

LINDISCR2

- **Purpose**

  Performing 2 groups linear discriminant analysis.

- **Format**

  $\{a, spl, m1, m2, ftest, pval, table\} = \text{LINDISCR2}(dat, ho);$

- **Input**

dat	$(n, p+1)$ data matrix with first column is grouping indicator
ho	scalar, flag; if =1, then classification table is computed by jackknifing

- **Output**

a	$(1, p)$ discriminant function
spl	(p, p) matrix, pooled covariance matrix
*m*1	$(1, p)$ vector, mean vector of first group
*m*2	$(1, p)$ vector, mean vector of second group
ftest	scalar, test statistic for testing difference of $m1$ and $m2$
pval	scalar, p-value of difference test
table	$(2, 2)$ matrix of classification results (true groups rowwise)

- **Source**

  `discrimi.src`

LOESS

- **Purpose**

  Robust smoothing of a scatterplot

- **Format**

  $\{yhat, ys, xs\} = \text{LOESS}(y, x);$

- **Input**

y	$(n, 1)$ vector, dependent variable
x	$(n, 1)$ vector, independent variable

- **Output**

$yhat$	$(n, 1)$ vector, predicted depvar given indvars
ys	$(numEval, 1)$ vector, ordinate values given abscissae values in xs.
xs	$(numEval, 1)$ vector, equally spaced abscissae values

- **Remarks**

  This is not part of the multivar package. But it is useful in this context so it is listed here.

- **Source**

  `intrinsic GAUSS command; avaiable in newer versions`

LOGREG

- **Purpose**

  Performs logistic regression

- **Format**

  $\{b, fv, df, cor, se, chisq, table\} = \text{LOGREG}(y, x);$

- **Input**

y	$(n, 1)$ vector, dependent variable consisting of 0 and 1
x	(n, p) matrix of independent variables

- **Output**

b	$(p+1, 1)$ vector, coefficients
fv	$(n, 1)$ vector, fitted values
df	scalar, degrees of freedom
cor	$(p+1, p+1)$ correlation matrix of parameters
se	$(p+1, 1)$ vector, stdv. of estimates
$chisq$	scalar, value of chi-square-test-statistic
$table$	$(2, 2)$ matrix of classification results first row: classification to group 1 (E(y)>0.5), second row: classification to group 2 (E(y)<=0.5)

- **Remarks**

  a constant is included if not present

- **Source**

  `logreg.src`

MDSMETR

- **Purpose**

  Performs metric multidimensional scaling

- **Format**

  $\{x\} = \text{MDSMETR}(d, dim);$

- **Input**

d	(n, n) distance matrix
dim	scalar, dimension in which the solution is determined

- **Output**

x	(n, dim) matrix, the configuration

- **Source**

  `mds.src`

MDSNONM2

- **Purpose**

  Performs nonmetric multidimensional scaling with 2 dimensional configuration

- **Format**

  $\{x\} = \text{MDSNONM2}(d);$

- **Input**

d	(n,n) distance matrix

- **Output**

x	$(n,2)$ matrix, the configuration

- **Remarks**

  algorithm by R. Mathar

- **Source**

  `mds.src`

MEANMV

- **Purpose**

  Computes the mean of every column of a matrix.
  Allowing for missing values.

- **Format**

  $\{m\} = \text{MEANMV}(x);$

- **Input**

x	(n, k) matrix, the data

- **Output**

m	$(k, 1)$ vector, containing the mean of every column of x

- **Source**

  `rmoment.src`

MEDIANMV

- **Purpose**

  Computes the median of every column of a matrix.
  Allowing for missing values.

- **Format**

  $\{m\} = \text{MEDIANMV}(x);$

- **Input**

x	(n, k) matrix, the data

- **Output**

m	$(k, 1)$ vector, containing the median of every column of x

- **Source**

  `rmoment.src`

MONOTONREG

- **Purpose**

  Performs a monotone regression

- **Format**

  $\{a1\} = \text{MONOTONREG}(a);$

- **Input**

a	$(n, 1)$ vector of distances

- **Output**

$a1$	$(n, 1)$ vector, the increments of the convex minorant

- **Remarks**

  algorithm by R. Mathar

- **Source**

  `mds.src`

NCHECKPLOT

- **Purpose**

  QQ-Plot to assess Normality

- **Format**

  NCHECKPLOT(x);

- **Input**

x	(n,p) matrix, the data

- **Output**

  none

- **Remarks**

  Univariate data are standardized and quantiles from normal distribution are used;
  for multivariate data Mahalanobis distances are computed and chi square approximation is used;
  envelopes and pointwise 5%, 95% bands are computed by simulation

- **Source**

  `mvaplots.src`

ORD2NORM

- **Purpose**

  transformes the values of ordered categorial variables into the observations of standard normal variables

- **Format**

  $\{d1, d2\} = \text{ORD2NORM}(x);$

- **Input**

x	(n,p) data matrix

- **Output**

$d1$	(n,p) matrix of boundaries of the transformation given by $\Phi^{-1}(\#(X \le x_i)/n)$
$d2$	(n,p) matrix of conditional expectations for each bin resulting from $d1$

- **Source**

  `rmoment.src`

PLOTTREE

- **Purpose**

  Plot of a regression tree

- **Format**

  PLOTTREE($tree, vars, tit, mod, f$);

- **Input**

$tree$	$(k, 10)$-matrix, output of growtree
$vars$	$(p, 1)$ string array, variable labels or " "
tit	string, title of plot
mod	scalar, the mode of the plot $= 0$: right nodes correspond to $>$ of covariables $= 1$: position of nodes are given by (mean,sqrt(ni)) $= 2$: position of nodes are given by (median,sqrt(ni))
f	scalar, factor to scale messages in the plot

- **Output**

  none

- **Remarks**

  mod: option 1 or 2 only when there are no missings in $tree[., 10]$, i.e. when option mean or median are choosen in growtree. No check here!)

- **Source**

  `tree.src`

PRINCOMPC

- **Purpose**

  Computes principal components of a data matrix.

- **Format**

  $\{c, v, a\} = \text{PRINCOMPC}(x, j);$

- **Input**

x	(n, p) matrix, the data
j	scalar, number of principal components to be computed

- **Output**

c	(n, j) matrix of the first j principal components of x in descending order of amount of variance 'explained'.
v	$(j, 1)$ vector of variance explained
a	(j, p) matrix of factor loadings, such that $x = c{*}a{+}error$

- **Remarks**

  adapted from an intern GAUSS procedure

- **Source**

  `pcompana.src`

PROCRUST

- **Purpose**

  Optimal transformation of the $(n, 2)$ matrix Y of x-y coordinates into an other $(n, 2)$ matrix YS so that the sum of squared differences to the $(n, 2)$ matrix X is minimized

- **Format**

  $\{xc, ys, m2a, m2b\} = \text{PROCRUST}(x, y, dflag, sflag);$

- **Input**

x	$(n, 2)$-data matrix
y	$(n, 2)$-data matrix
$dflag$	scalar, if =1, scale effekt (dilation) is removed
$sflag$	scalar, if =1, symmetrical analysis

- **Output**

xc	$(n, 2)$ matrix, centered x- coordinates
ys	$(n, 2)$ matrix optimaly transformed y-coordinates
$m2a$	scalar, sum of squared differences before transformation
$m2b$	scalar, sum of squared differences after transformation

- **Remarks**

  procedure by R.Schnell

- **Source**

  `mds.src`

PROXMVE

- **Purpose**

  Computation of "one-step-reweighted" estimators for location and scatter with MVE as start values

- **Format**

  $\{t, v\} = \text{PROXMVE}(x);$

- **Input**

x	(n, p) matrix, the data (n observations, p variables)

- **Output**

t	$(p, 1)$ vector, location estimates
v	(p, p) matrix, estimation of covariance matrix

- **Remarks**

  see Rousseeuw and van Zomeren

- **Source**

  `rmoment.src`

SCMATPLOT

- **Purpose**

  Plotting a Matrix of Scatterdiagrams

- **Format**

  SCMATPLOT(x, var);

- **Input**

x	(n, p)-matrix, data
var	string, vector with names of variables or " "

- **Output**

  none

- **Source**

  `mvaplots.src`

STARPLOT

- **Purpose**

  Starplot of Objects

- **Format**

  STARPLOT($x, r, c, objects, tit$);

- **Input**

x	(n,p)-matrix, data
r	number of rows for the stars in the plot
c	number of columns for the stars in the plot
	$r * c \leq rows(x)$
$objects$	string-vector with names of objects or " "
tit	string, the title of the plot or " "

- **Output**

  none

- **Remarks**

  The first variable is at 12 o'clock; the others follow clockwise.

- **Source**

  `mvaplots.src`

STANDARD

- **Purpose**

  Computes the standardization of a matrix.
  Allowing for missing values.

- **Format**

  $\{x0\} = \text{STANDARD}(x);$

- **Input**

x	(n, k) matrix

- **Output**

$x0$	(n, k) matrix, standardized data with mean 0 and variance 1

- **Remarks**

  ```
  x0 = (x-meanmv(x)')./stdmv(x)'
  ```

- **Source**

  `rmoment.src`

STDMV

- **Purpose**

  Computes the standard deviation of every column of a matrix. Allowing for missing values.

- **Format**

  $\{s\} = \mathrm{STDMV}(x);$

- **Input**

x	(n, k) matrix

- **Output**

s	$(k, 1)$ vector, containing the standard deviation of every column of x

- **Source**

  `rmoment.src`

SURROGATE

- **Purpose**

  Assignes observations with missings on split variables to higher nodes

- **Format**

  $\{trn, noden\} = \text{SURROGATE}(dat, cens, tr, node);$

- **Input**

dat	$(n, p+1)$ or $(n, p+2)$ data matrix as in growtree
$cens$	scalar, censoring indicator: $= 1$: data are censored (i.e. the second column of dat contains censoring information) $\neq 1$: data are not censored
tr	$(k, 10)$-matrix, the tree, output of growtree
$node$	$(n, 1)$ vector, indices of hightest nodes the observations belong to, output of growtree

- **Output**

trn	$(k, 10)$ matrix, the new tree
$noden$	$(n, 1)$ vector, the new indices of node membership

- **Source**

  `tree.src`

TR

- **Purpose**

  Computes the trace of a matrix.

- **Format**

  $\{tr\} = \mathrm{TR}(x);$

- **Input**

x	(n, k) matrix

- **Output**

tr	scalar, trace of x

- **Remarks**

  The matrix need not to be square.

- **Source**

  `rmoment.src`

TWINPLOT

- **Purpose**

  Scatterplot of two data matricies in one plot

- **Format**

  TWINPLOT $(x, y, lflag, xlab, ylab)$;

- **Input**

x	$(n, 2)$-matrix, data
y	$(m, 2)$-matrix, data
$lflag$	scalar, if =1, objects are labled
$xlab$	string (length 1), label of x objects
$ylab$	string (length 1), label of y objects

- **Output**

  none

- **Remarks**

  written by R.Schnell

- **Source**

  `mds.src`

UNIMVE

- **Purpose**

  Computes univariate minimum volume estimator (length of the shortest half) and the corresponding location estimator.

- **Format**

  $\{l, s\} = \text{UNIMVE}(x);$

- **Input**

x	(n,1) vector, the data

- **Output**

l	scalar, MVE location estimate
s	scalar, MVE scale estimate

- **Source**

  `rmoment.src`

VARIMAX

- **Purpose**

  Varimax rotation of factor loadings.

- **Format**

  $\{lt, trans\} =$ VARIMAX($lding$);

- **Input**

$lding$	(p, k) matrix of factor loadings

- **Output**

lt	(p, k) matrix of rotated factor loadings
$trans$	(k, k) matrix, the transformation matrix

- **Remarks**

  adapted from a program written in MATLAB by L.F.Marcus, in: Reyment,R. and K.G.Joereskog: Applied Factor Analysis in the Natural Sciences Cambridge Universty Press 1996; see also: Morrison: Multivariate Statistical Methods

- **Source**

  `factor.src`