CMAT Newsletter: December 2010

Wolfgang M. Hartmann

December 2010

Contents

1 General Remarks

New algorithms for an old problem of linear regression are implemented in the `lrforw()` and the `lrallv()` functions. The new `lrallv()` function contains two algorithms for solving the *all variable subsets* regression problem for subsets m of a (possibly large) number of n variables:

- full enumeration of all distinct variable combinations in sets of m, which is only possible for relatively small m and n;
- stochastic search algorithm for a good approximation to the global solution which can be used for tens of thousands of variables as it could be necessary for microarray data.

In the last chapter (*Illustration*) we report some benchmarks for large data sets. It shows that the `lrforw()` algorithm is able to solve problems up to 100,000 variables and obviously most of the time with higher precision than the new stochastic algorithm. With my old PC was full enumeration computationally not feasible for $n > 800$ variables.
I spent several weeks on enhancing the `factor()` function and found out that I had to move some of the functionality into a new function called `cfa()` (*confirmatory factor analysis*) which naturally also is able to compute EFA (exploratory factor analysis). The new `cfa()` function is able to do exact analysis of correlation matrices, where the diagonal is considered fixed and not subject to fit. (Everything connected to the old `"semt"` version is now part of the `cfa()` function and the `"semt"` version is no longer available with the `factor()` function.) It can also analyze ordinal data using polychoric correlations. In some future work we intend to implement some automatic model improvement for CFA including some factor pattern output.
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

affrma RMA (Robust Multichip Average) by Bolstad et.al (2003) for the normalization of microarray data (still not finished yet!) (affrma)

auroc this function computes the area under the ROC (receiver operator) curve and its asymptotic standard error (Mashtare & Hutson, 2010) (auroc)

cfa this function implements a new set of algorithms for exploratory and confirmatory factor analysis which were splitted off the older `factor()` function since it requires additional input arguments and has additional features similar those of the `sem` function (cfa)

delta implements the *Delta Method* (Cox, 1998) (delta)

hbttest implements the common and Satterthwaite's t test (hbttest)

lrallv implements two algorithms for all variable subsets linear regression, the traditional full enumeration and a stochastic search algorithm for very large problems where full enumeration is practically infeasible (lrallv)

noblanks removes leading and trailing white space from string data and optionally replaces inside white space (noblanks)

order obtains indices by hierarchical ranking of various data vectors (order)

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)

survcurv implements some algorithms for the computation of survival curves for censored data for Aalen's additive model and the Cox proportional hazards model for censored data (survcurv)

survfor implements a *Survival Forest* algorithm for censored data (survfor)

survreg implements some GLIM-like algorithms for survival regression, Aalen's additive model, and the Cox proportional hazards model for censored data (survreg)

survprd test set evaluation (scoring) for `survreg` (survprd)

1.2 Fixed Bugs

A number of bugs were fixed. Especially, one for the subject reduction operators when there are missing values. Subject reuction will yield missing values when applied to missing values. That means, you may have to use the `loc()` function prior to applying the subject reduction operator to find the location of nonmissings. That's why `loc()` was extended with the `"nm"` (nonmissing) option.

Another bug was fixed with the Γ random number generator. The `randlib` function has the syntax `gengam(a,r)` where a is location and r is the shape. Normally (now in CMAT and was also in Matlab), the notation $\Gamma(\alpha, \beta)$ is used where α is the shape. Therefore the following substitution was necessary when calling the `randlib function`:

$$a = 1./\beta, \quad r = alpha$$

There was a very serious bug when running `factor()` with Fuller's method and asking for confidence intervals on the parameters. This was fixed at the cost of more memory allocation. Another bug was fixed in `sem()` when the the sparse Jacobian algorithm was used together with user specified constraints.

2 Modifications of Features

2.1 Modification of the `lrforw` Function

The `lrforw` function was written as an extension of PROC DMINE of the variable selection node in the SAS *Enterprise Miner* package. Whereas PROC DMINE is only able to analyse data sets with about $n = 3000$ to $n = 10,000$ of columns of the data set, due to the size of the $\mathbf{X}^T\mathbf{X} \in \mathcal{R}^{n \times n}$ matrix, function `lrforw` can be used for much larger applications.
When developing the new `lrallv` function, we changed the syntax of the `lrforw()` function from an option vector to the (for CMAT now almost typical) two column matrix input. The following was the old set of options:

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

The `optn` argument is now 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:

Option	Second Column	Meaning
"aov"		using *AOV16* variables
"cutof"	real	cutoff for accuracy table (def=.5); (valid only for binary response)
"grp"		using *Group* variables
"maxc"	int	upper range for number of columns in $\mathbf{X}^T\mathbf{X}$ matrix (def=4000);
"maxst"	int	upper threshhold for number selected vars in step 2; another criterion for terminating the selection process.
"pobs"	int	print observationwise statistics: predicted $\hat{y}$, residuals
"print"	int	amount of printed output =0 specifies no printed output, default is 2
"r2min"	real	lower R2 threshold for selection this raw (unpartial) R^2 value from first step: only variables (effects) with larger R^2 values are candidates for selection in step 2.
"r2st"	real	lower threshold for R^2 or upper threshold for χ^2 for terminating the selection process; the R^2 or χ^2 value is now based on the partial $\mathbf{X}^T\mathbf{X}$ matrix
"sing"	real	singularity criterion for inverting the $\mathbf{X}^T\mathbf{X}$ matrix; default is run-time option, which defaults to 1.e-8
"selc"	string	specifies the selection criterion
	"r2"	use R^2 squared value
	"chi"	use probabilty of $F(n, d)$ value
	"aic"	Akaike's information criterion
	"bic"	Bayesian information criterion
	"sbc"	Schwarz SBC criterion
	"adjr2"	adjusted R^2 value
	"cp"	Mallow's Cp
"vers"		use Foster & Stine (2004) algorithm

Note, that the Foster-Stine algorithm should be specified for very large applications, e.g. where the number of varoables exceeds one or two thousand.

2.2 Modification of the `nlreg` Function

The old form of the function

gof = nlreg(func,x0<,optn<,nlpopt<,jac> . >)

was extended with a number of additional input arguments

gof = nlreg(func,x0<,optn<,nlpopt<,lbub<,lau<,nlc <,jac<,hess<,jcon> . >)

to be more consistent with the `nlp()` function. The following are the new input arguments:

lbub (optional) is the name of a $n \times 2$ matrix with lower (first column) and upper (second column) simple boundary constraints. (see **??** for `nlp()` function)

lau (optional) specifies a $n_{lc} \times n + 2$ or $n_{lc} \times n + 1$ matrix with lower bounds (first column), coefficients, and upper bounds (last column) for n_{lc} general linear constraints. (see **??** for `nlp()` function)

nlc (optional) is the name of a user specified function evaluating n_{nlc} nonlinear equality or inequality constraints $c = (c_1, \ldots, n_{nlc})^T$ for given variables (points) $x = (x_1, \ldots, x_n)^T$. (see **??** for `nlp()` function)

jac (optional) is the name of a user specified function defining the $N \times n$ Jacobian matrix of first order derivatives of the objective function. (see **??** for `nlp()` function)

hes (optional) is the name of a user specified function specifying the $n \times n$ Hessian matrix (matrix of second order derivatives) of the objective function, $\mathbf{G} = \nabla^2 f = (\frac{\partial^2 f}{\partial x_j \partial x_k})$ (see **??** for `nlp()` function)

jcon (optional) is the name of a user specified function specifying the $nc \times n$ Jacobian matrix $\mathbf{J}^{con}$ of the nc nonlinear constrains specified in `nlcon` w.r.t. the n variables, $\mathbf{J}^{con} = \begin{pmatrix} (\nabla c_1)^T \\ \vdots \\ (\nabla c_m)^T \end{pmatrix} = (\frac{\partial c_i}{\partial x_j})$ evaluated at $x = (x_1, \ldots, x_n)$. (see **??** for `nlp()` function)

2.3 Modification of the `factor` Function

For the extension of features covered with the old `factor()` function we had to split this function into an older part (which is a subset of unchanged features of the old function) and a new function which we call `cfa()`. Whereas the new function is explained below among the set of new functions we describe here only which features are still left in the `factor` function.

All methods related to the old `semt` version were moved from `factor()` into `cfa()` and the `"semt"` version is no longer avauilable with the `factor()` function. Only the old SAS-like estimation methods for ULS and ML (Tucker's method) and the ULS, GLS, ML, ALPHA, and MINRES methods by Michael Browne remain in `factor()`. This means that we will focus more on the extension of features in the new `cfa()` function and won't much change the remaining part in `factor()`. In addition to `factor()`, `cfa()` can analyze ordinal data via polychoric correlations. And `cfa()` is able to perform a correct estimation for correlation data, where the diagonal of the correlation matrix is not fitted and instead considered constant. In `factor()` even correlation matrices are analyzed essentially like covariance matrices, including the fit of the unit diagonal. But the most important feature of `cfa()` which is still in development is the

ability to specify a structural pattern for the loading and correlation matrices of the model.

Feature	factor	cfa	sem
automatic model improvement	-	+	-
multiple sample analysis	-	+	+
standard normal ASEs and CIs	+	+	+
robust CIs and fit measures	-	+	+
profile likelihood CIs	+	+	+
bootstrapping CIs and GOF	+	+	+
Bollen-Stine bootstrap for p	+	+	+
pattern specification (*1)	-	+	+
modification indices (*2)	-	+	+
different factor numbers (*3)	-	+	+
ordinal data (polychoric corrs)	-	+	-
correlation analysis (*4)	-	+	+/-
ULS	+	+	+
GLS (*5)	-	+/-	+
ML (*6)	+	+	+
WLS	-	+	+
DWLS	-	+	+
determining number of factors (*7)	+	-	-
rotation to simple structure (*8)	+	+/-	+

Ofcourse, the `sem` function is able to analyze models with more general form than that of the exploratory (EFA) or confirmatory (CFA) factor analysis.

(*1) there is a default parameter pattern for EFA

(*2) for EFA the asymptotic standard errors and the confidence intervals replace the modification indices

(*3) pattern definition may specify different number of factors for different samples

(*4) correlation data can be analyzed with all functions, `sem`, `factor` and `cfa`, however the `cfa` function is able to model correlation data with a specific model which results in a model matrix with fixed unit diagonal; the `sem` function permits specifying constraints for models similar to the `cfa` correlation structure model

(*5) GLS estimation is not available for polychoric correlations and correlation analysis (polychoric correlation matrices may not be pd)

(*6) ML estimation for correlation type analysis is still computational inefficient

(*7) for CFA, usually the pattern definition defines the number of factors; for EFA the `factor` function offers a number of test criteria for the number of factors

(*8) only EFA solutions can be rotated, for CFA the specified pattern and goodness of fit is affected

FACTOR	**CFA**
1. input data	1. input data (for multiple samples)
2. compute criteria for number of factors	2. input parameter pattern (models for all samples)
3. obtain suitable starting values for default pattern	3. obtain suitable starting values for default pattern
4. estimate optimal parameters and standard normal ASEs	4. estimate optimal parameters and standard normal ASEs
5. compute goodness-of-fit measures	5. compute goodness-of-fit measures
6. orthogonal or oblique rotation to simple structure with recomputation of standard normal ASEs	6. optional: compute robust goodness-of-fit measures and robust ASEs
7. optional: compute profile likelihood CIs	7. compute modification indices
8. optional: bootstrap for GOFs and confidence intervals	8. optional: compute profile likelihood CIs
	9. optional: bootstrap for GOFs and confidence intervals

The following is the description of the new `factor()` function which has only a subset of the features of the old function:

< gof,est,resi,cov,boci> = factor(data,optn<,xini<,prior<,targ<,wtrg> . >)

Purpose: The `factor` function implements a number of exploratory factor analysis methods which can be used for mainly for ULS and ML estimation. The factor model

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

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. The SAS-like ML method (based on work by Fuller) was extended for the computation of normal and robust asymptotic standard errors and Wald confidence intervals. `factor()` (like `cfa`) also implements a number of popular orthogonal and oblique rotation methods and computes asymptotic standard errors and Wald confidence intervals of the rotated loadings.

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

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

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

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

1. `nobs` by `nvar` matrix of raw data where the `nvar` variables are asumed to be continuous (use function `cfa()` for ordinal data)
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.

xini This argument specifies a `npar` vector of starting values for the parameter estimation.

prior This argument specifies a `nvar` vector of prior communalities. This is valid only for methods where the unique variances are specified but not estimated. Communalities h_i are defined for the the correlation analysis as 1 minus the unique variances u_i.

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 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 $\alpha = 0.1$
"ana"	string	data type for analysis
	"cor"	correlations are analyzed
	"cov"	covariances are analyzed
	"ucor"	uncorrected correlations are analyzed
	"ucov"	uncorrected covariances are analyzed
"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
"boci"		compute bootstrap confidence intervals for unique vars and rotated or unrotated loadings
"bost"		compute Bollen-Stine bootstrap value of p
"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
"bseed"	int	seed value for bootstrap random generator
		(same as in jboot and sem
"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
"cook"	real	use the scree test with Cooks distance
		for specifying the number of factors
		default threshold is .25

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
"dfred"	real	reduce degrees of freedom of χ^2 test
"demph"	real	add positive constant to initial values
"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 γ)
	"varmax"	orthogonal Varimax rotation (crafer with $\gamma = \frac{1}{n}$)
	"quamax"	orthogonal Quartimax rotation (crafer with $\gamma = 0$)
	"equmax"	orthogonal Equamax rotation (crafer with $\gamma = \frac{q}{2n}$
	"parmax"	orthogonal Parsimax rotation (crafer with $\gamma = \frac{q-1}{n+q-2}$)
	"parmax"	orthogonal Factor Parsimony rotation (crafer with $\gamma = 1$)
	"bentlr"	orthogonal Bentler rotation criterion
	"minent"	orthogonal Minimum Entropy rotation
	"oblmin"	orthogonal Direct Oblimin family (specify γ)
	"simmax"	orthogonal Simplimax (specify number nonzero loadings with γ)
	"tandm1"	orthogonal Tandem 1 rotation
	"tandm2"	orthogonal Tandem 2 rotation
	"infmax"	orthogonal Infomax rotation
	"mccamm"	orthogonal McCammon rotation
	"crafer"	oblique Crawford-Ferguson family (specify γ)
	"oblmin"	oblique Direct Oblimin family (specify γ)
	"quamin"	oblique Direct Quartimin (oblmin with $\gamma = 0$)
	"biqmin"	oblique Bi-Quartimin (oblmin with $\gamma = .5$)
	"covmin"	oblique Covarimin (oblmin with $\gamma = 1$)
	"bentlr"	oblique Bentler rotation criterion
	"simmax"	oblique Simplimax (specify number nonzero loadings with γ)
	"oblmax"	oblique Oblimax
	"geomin"	oblique Geomin
	"infmax"	oblique Infomax rotation
	"promax"	oblique Promax
	"targt1"	Target Rotation (Harman, 1976)
	"targt2"	Weighted Target (Browne,2001)
"fnorm"		Kaiser's factor pattern normalization
		use only for exploratory FACTOR model
"hey"		Heywood case: apply boundary constraints
"inic"		compute initial values

Option Name	Second Column	Meaning
"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) (only vers=fmwb and cmwb)
	"alph"	for ALPHA method (only vers=fmwb and cmwb)
	"minr"	for MINRES method (only vers=fmwb and cmwb)
	"ogas"	ML method by Ogasarawa
"maxfu"	int	maximum number function calls for optimization
"maxit"	int	maximum number iterations for optimization
"maxrot"	int	maximum number iterations for rotation
"mineig"	real	smallest eigenvalue for which factor is retained used for specifying the number of factors
"nfac"	real	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
"obl"		use oblique rotation
"ort"		use orthogonal rotation
"pall"		large amount of printed output
"pcov"		print covariance matrix of parameter estimates
"peig"		print eigenvalue distribution(s)
"perc"	real	proportion of common variance to be accounted for used for specifying the number of factors default is 1 or 100 %
"phis"		print optimization history
"pini"	int	print input information
"pres"		print raw, standardized, normalized residuals
"prin"	int	amount of printed output
"pscor"		print factor scores
"pscre"	int	print information on scree test
"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 ([?]) (same as in `jboot` and `sem`
"rob"		robust estimation of ASE and χ^2
"ridge"	real	nonnegative ridge value
"rinit"	int	initialization for rotation matrix $\mathbf{T}$
"rotp"	real	rotation parameter (γ for Crawford-Ferguson)
"rtres"	real	small threshold for setting residuals to zero
"rvers"	int	version of algorithm for rotation

Option Name	Second Column	Meaning
"samp"	int	number of bootstrap samples (default=500)
		(same as in jboot and sem
"start"	string	type of residual matrix
	"com"	input communalities
	"efa"	LS EFA by linear algebra
	"ran"	pseudo random
"sing"	real	singularity threshold (default: 1.e-8)
"seed"	real	seed value for random generator
"stud"	real	use the scree test with studendized residual
		for specifying the number of factors
		default: $\alpha = .1$
"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
	"nms"	Nelder-Mead simplex
"vers"		version of algorithm
	"comm"	common version (similar to SAS PROC FACTOR)
		only meth=ULS, ML
	"fmwb"	original algorithms by Michael Browne
		only meth=ULS, GLS, ML, MINRES, ALPHA
	"cmwb"	modified algorithms by Michael Browne
		only meth=ULS, GLS, ML, MINRES, ALPHA
	"ogas"	ML method by Ogasarawa
"wgt"	int	column number of WEIGHT variable when raw data input
"wridge"	real	ridge factor for weight matrix (GLS, WLS, DWLS)
"wpen"	real	penalty weight for WLS and DWLS estimation

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 raw data are given and `rob` is 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.

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: cfa(), noharm(), sem(), frotate(), pca()

3 Extensions to the Language

Data lists now may contain the following data objects:

- numerical (int, real, complex) and string scalars
- vectors and matrices
- tensors
- (sub-)lists

Since lists are (currently) restricted to onedimensional arrays of data objects, the extension that lists may contain sublists, is practically equivalent to mutidimensional lists. A nice example is the `parm` return argument of the new `cfa` function, which may be:

1. if no standard errors are computed and for only one sample: this is a list of three objects (F, P, U)
2. if no standard errors are computed and for $k > 1$ samples: these are s lists each with three objects (F, P, U)
3. if standard errors are computed and for only one sample: these are 5 (or 9 for additional robust estimation) lists (estimates, ASEs, t values, lower and upper CIs) each with three objects (F, P, U)
4. if standard errors are computed and for s samples: these are 5 (or 9) lists (estimates, ASEs, t values, lower and upper CIs) of s lists each with three objects (F, P, U), in total this are $5 * s * 3$ (or $9 * s * 3$) lists in three dimensions $[5, s, 3]$ (or $[9, s, 3]$).

4 Extensions to Various Functions

4.1 Extension of the `loc()` Function

The set of selection criteria was extended by `"nm"` to find the index locations of all *nonmissing* values.
The following conditions may now be specified:

Criterion	Meaning
`"ms"`	find locations of all missing values no third argument must be specified
`"nm"`	find locations of all nonmissing values no third argument must be specified
`"nz"`	find locations of all nonzeros and nonmissings no third argument must be specified
`"eq"`	find locations of all zeros no third argument is specified
`"ne"`	find locations of all nonzeros (includes locations of missing value) no third argument is specified this is the default and must not be specified
`"eq"`	find locations of all entries in `a` which are equal to zero or a specified third argument
`"ne"`	find locations of all entries in `a` which are unequal to zero or a specified third argument
`"gt"`	find locations of all entries in `a` which are larger than zero or a specified third argument
`"lt"`	find locations of all entries in `a` which are smaller than zero or a specified third argument
`"ge"`	find locations of all entries in `a` which are larger or equal than zero or a specified third argument
`"le"`	find locations of all entries in `a` which are smaller or equal than zero or a specified third argument

5 New Developments

5.1 Function `affrma`

gof = affrma(data,optn<,ref>)

< gof,dnew > = affrma(data,optn<,ref>)

Purpose: The `affrma` function implements the Bioconductor RMA algorithm by Bolstad et.al (2003) for the normalization of microarray data. This algorithm is an alternative to the `affvsn` and `affarms` methods for obtaining column normalized microarray data. However, other than fitting the parameters of a Maximum Likelihood model, the RMA method applies robust Medianpolish (Tukey, 1977a, p. 179), see also the `mpolish` function in CMAT to the data. This function is still in the works!

Input: **data** this should be a $N \times n$ matrix of microarray data where the rows correspond to features (genes) and the columns to samples which need to be normalized. The data may obtain a column for an ID variable specifying a strata. The column number of the ID variable must be specified by the `optn` argument.

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.

ref not yet implemented.

Option	Second Column	Meaning
`"bgcor"`		perform background correction default is background correction
`"idvar"`	int	column number containing ID variable; default is no ID variable
`"medp"`	int	algorithm for mean or median polish default is =0: median polish, =1: mean polish
`"mpit"`	int	maximum number of median polish iterations; default is 10 for median and 1 for mean polish
`"mpeps"`	real	for terminating median polish; default is 1.e-2
`"nobg"`		do not perform background correction default is background correction
`"nsamp"`		size $n_r \leq N$ of a reduced data set, needed only when N is too large for the computer resources. default is $n_r = 0$
`"pdat"`	int	print input and fitted data sets
`"print"`	int	amount of printed output =0 specifies no printed output, default is 1

Output: **gof** a vector of scalar results, see below for content;

dnew this is the normalized $N \times n$ data set.

Restrictions: 1. The input data must be numeric and cannot contain any string or complex data.

2. Currently the input data must not have any missing values.

Relationships: affarms(), affvsn()

Examples: 1. :

5.2 Function `auroc`

< auc,tab,xopt,cov > = auroc(data<,optn<,popt<,x0> . >)

Purpose: This function computes the area under the ROC (receiver operator curve) and its asymptotic standard error. The method by Mashtare and Hutson (2010), is based on the *Delta method*.

Input: **data** an $N \times 2$ data matrix containing the binary response variable in its first column (use the `"swap"` option if columns should be swapped)

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.

popt a two column matrix specifying the options for the optimization (see **??** for `nlp()` function)

x0 (optional) a 4-vector of initial estimates for the four ML estimates (two mean values and two variances)

Option	Second Column	Meaning
`"nint"`	int	number K of intervals, default: K=100
`"nopr"`		no printed output
`"pcov"`		print covariance matrix
`"popt"`		print optimization history
`"print"`	int	amount of printed output
		=0 specifies no printed output, default is 0
`"ptab"`		print (large) table for graph
`"swap"`		swap first with second column of input data

Output: **auc** 2-vector containing the area under the curve and its asymptotic standard error

tab table of the graph ($(K-1)(K-2)/2 \times 5$ matrix)

xopt 4-vector with optimal estimates

cov a 4 covariance matrix of the optimal ML estimates

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

Relationships: roccomp(), glim(), fider(), delta()

Examples: 1. Example used by Mashtare & Hutson(2010): N=100

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

```
nr = nrow(body);
optn = [ "swap"        ,
         "pcov"        ,
         "popt"      2 ,
         "nint"    100 ];
< auc,tab,xopt,cov > = auroc(body,optn);
print "auc=",auc;
print "tab=",tab[1:100,];
print "xopt=",xopt;
print "cov=",cov;
```

```
                    Trust Region Optimization
                    Without Parameter Scaling
               Gradient Computed by Finite Differences
           Hessian Computed by Finite Differences (dense)
                     (Using Only Function Calls)

   Iteration Start:
   N. Variables              4
   N. Bound. Constr.         2        N. Mask  Constr.          0
   Criterion       51770.74485        Max Grad Entry  49025.34826
   N. Active Constraints     0        TR Radius       1.000000000


  Iter rest nfun act   optcrit  difcrit maxgrad  lambda   radius
    1*   0    2   0  28712.18 23058.56 20512.5 1666.63  1.00000
    2*   0    3   0  16228.90 12483.28 8656.88 1030.30  1.00884
    3*   0    4   0  9705.371 6523.531 3797.05 965.500  1.07655
    4*   0    5   0  6398.869 3306.502 1826.68 990.417  1.04947
    5*   0    7   0  3902.332 2496.537 761.077 109.264  2.07607
    6*   0    8   0  2638.167 1264.165 324.474 73.0178  2.22255
    7*   0    9   0  1997.306 640.8610 142.334 49.4278  2.28763
    8*   0   14   0  1040.304 957.0016 41.5044 3.23487  50.3847
    9*   0   16   0  866.1705 174.1336 36.8935 22.3010  2.64280
   10*   0   18   0  827.6168 38.55370 39.3836 112.556  0.58205
   11*   0   21   0  744.0924 83.52444 36.3432 16.2822  5.03819
   12*   0   22   0  629.4546 114.6378 90.1460 21.1341  2.51910
   13    0   24   0  601.4987 27.95591 27.3341 32.7262  0.66390
   14    0   25   0  598.3184 3.180238 4.16412       0  0.68422
   15    0   26   0  598.2513 0.067165 0.14216       0  0.26221
   16    0   27   0  598.2512 7.0e-005  2e-004       0  0.03051
   17    0   28   0  598.2512 1.0e-010  2e-008       0   1e-003

   Successful Termination After    17 Iterations
   GCONV convergence criterion satisfied.
   Criterion       598.2511815        Max Grad Entry  1.6594e-008
```

```
N. Active Constraints     0        Ridge (lambda)            0
TR Radius       0.000971793        Act.dF/Pred.dF  1.010417372
N. Function Calls        29        N. Gradient Calls         2
N. Hessian Calls         19        Preproces. Time           0
Time for Method           0        Effective Time            0
```

The output of the tableof $(K-1)*(K-2)/2$ grid points should probably be avoided for $K = 100$:

```
     |        Xi        Xj        Hi        Hj        Hij
-----------------------------------------------------------
   1 |   0.01000   0.02000   0.29191   0.37645   0.66836
   2 |   0.01000   0.03000   0.29191   0.43369   0.72560
   3 |   0.01000   0.04000   0.29191   0.47776   0.76966
   4 |   0.01000   0.05000   0.29191   0.51383   0.80574
   5 |   0.01000   0.06000   0.29191   0.54446   0.83637
   ...........................................................
4846 |   0.96000   0.97000   0.99834   0.99885    1.9972
4847 |   0.96000   0.98000   0.99834   0.99930    1.9976
4848 |   0.96000   0.99000   0.99834   0.99970    1.9980
4849 |   0.97000   0.98000   0.99885   0.99930    1.9982
4850 |   0.97000   0.99000   0.99885   0.99970    1.9985
4851 |   0.98000   0.99000   0.99930   0.99970    1.9990
```

```
xopt=
   |         1         2         3         4
-----------------------------------------------
 1 |    17.725    20.806    1.8285    2.1393
```

```
cov=
 S |         1         2         3         4
-----------------------------------------------
 1 |   0.01286
 2 |   0.00000   0.01853
 3 |    4e-007   0.00000   0.00643
 4 |   0.00000    6e-007   0.00000   0.00926
```

The computation time (for the run without table output) is still less than 1 second which is not so when using the SAS Macro:

```
Area under curve=0.862536 Asymptotic standard error=0.0156759
         Number Intervals=100 Time for Computation=0
```

5.3 Function `cfa`

< gof,est,resi,cov,boci> = cfa(data,optn<,patt<,scal<,wgt<,xini<,targ<,wtrg> . >)

Purpose: The `cfa` function implements ULS, ML, GLS, WLS, and DWLS estimation for confirmatory and exploratory 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 factor model

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

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.

The `cfa` function is able to analyze ordinal data using 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.)

The `cfa` function is able to analyze multiple data samples (e.g. specified by an ID data column) with a simultaneous model. Different parameter patterns and even different factor numbers can be specified for the different samples.

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 covari-

ance 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.

The `cfa()` function can analyze ordinal raw data by computing polychoric correlations. If the raw input data contains both, ordinal and continuous variables, the continuous variables are binned into 16 (which is default for the `nbin` option) intervals and polychoric correlations are computed.

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

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 $\alpha = 0.1$
"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
	"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
"boci"		compute bootstrap confidence intervals for unique vars and rotated or unrotated loadings
"bost"		compute Bollen-Stine bootstrap value of p
"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
"bseed"	int	seed value for bootstrap random generator (same as in jboot and sem
"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
"dfred"	real	reduce degrees of freedom of χ^2 test
"demph"	real	add positive constant to initial values
"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 γ)
	"varmax"	orthogonal Varimax rotation (crafer with $\gamma = \frac{1}{n}$)
	"quamax"	orthogonal Quartimax rotation (crafer with $\gamma = 0$)
	"equmax"	orthogonal Equamax rotation (crafer with $\gamma = \frac{q}{2n}$
	"parmax"	orthogonal Parsimax rotation (crafer with $\gamma = \frac{q-1}{n+q-2}$)
	"parmax"	orthogonal Factor Parsimony rotation (crafer with $\gamma = 1$)
	"bentlr"	orthogonal Bentler rotation criterion
	"minent"	orthogonal Minimum Entropy rotation
	"oblmin"	orthogonal Direct Oblimin family (specify γ)
	"simmax"	orthogonal Simplimax (specify number nonzero loadings with γ)
	"tandm1"	orthogonal Tandem 1 rotation
	"tandm2"	orthogonal Tandem 2 rotation
	"infmax"	orthogonal Infomax rotation
	"mccamm"	orthogonal McCammon rotation
	"crafer"	oblique Crawford-Ferguson family (specify γ)
	"oblmin"	oblique Direct Oblimin family (specify γ)
	"quamin"	oblique Direct Quartimin (oblmin with $\gamma = 0$)
	"biqmin"	oblique Bi-Quartimin (oblmin with $\gamma = .5$)
	"covmin"	oblique Covarimin (oblmin with $\gamma = 1$)
	"bentlr"	oblique Bentler rotation criterion
	"simmax"	oblique Simplimax (specify number nonzero loadings with γ)
	"oblmax"	oblique Oblimax
	"geomin"	oblique Geomin
	"infmax"	oblique Infomax rotation
	"promax"	oblique Promax
	"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
"inic"		compute initial values
"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
"nbin"	int	number bins for continuous variables, def=16
"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
"obl"		use oblique rotation
"ort"		use orthogonal rotation
"pall"		large amount of printed output
"pcov"		print covariance matrix of parameter estimates
"pcpr"		print information on polychoric correlation
"peig"		print eigenvalue distribution(s)
"perc"	real	proportion of common variance to be accounted for
		used for specifying the number of factors
		default is 1 or 100 %
"phis"		print optimization history
"pini"	int	print input information
"pres"		print raw, standardized, normalized residuals
"prin"	int	amount of printed output
"pscor"		print factor scores
"pscre"	int	print information on scree test
"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 ([**?**])
		(same as in `jboot` and `sem`
"rob"		robust estimation of ASE and χ^2
"ridge"	real	nonnegative ridge value
"rinit"	int	initialization for rotation matrix $\mathbf{T}$
"rotp"	real	rotation parameter (γ for Crawford-Ferguson)
"rtres"	real	small threshold for setting residuals to zero
"rvers"	int	version of algorithm for rotation

Option Name	Second Column	Meaning
"samp"	int	number of bootstrap samples (default=500)
		(same as in jboot and sem
"start"	string	type of residual matrix
	"com"	input communalities
	"efa"	LS EFA by linear algebra
	"ran"	pseudo random
"sing"	real	singularity threshold (default: 1.e-8)
"seed"	real	seed value for random generator
"stud"	real	use the scree test with studendized residual
		for specifying the number of factors
		default: $\alpha = .1$
"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)

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.

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: 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");
```

1. 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
	[0.584147, 0.584147]
i4	0.3079070
	[0.307907, 0.307907]
i5	0.5443523
	[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
	[0.584147, 0.584147]
i4	0.3079070
	[0.307907, 0.307907]
i5	0.5443523
	[0.544352, 0.544352]
i11	0.8352026
	[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]
```

2. 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
               [ 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]
```

5.4 Function `delta`

< est,ecov,ocov > = delta(xstar,fest,fopt<,opt<,egrd>>)

< est,ecov,ocov > = delta(xstar,fest,ocov<,opt<,egrd>>)

Purpose: This function implements the so-called *delta method*.

Input: **xstar** an n vector which must be a nearly optimal point w.r.t. an optimization function `fopt` which is usually a Maximum Likelihood function.

fest a function defining m estimates w.r.t. n parameters x

fopt|**ocov** must be one the following two:

fopt name of a function defining the optimization criterion

ocov must be the $n \times n$ covariance matrix of the (ML) estimation problem

opt (optional) string option could be `"forw"`, `"cent"` or `"expo"` specifying the precision of numeric derivatives

egrd (optional) the name of a function defining the analytic first order derivatives of the `fest` function w.r.t. x

If `fopt` is specified and (not only `ocov`), the gradient at `xstar` is tested for near optimality (the maximum absolute value cannot be larger than 1.e-4).

Output: **est** the $m \times 2$ matrix of estimates in its first column and with asymptotic standard errors in its second column

ecov the $m \times m$ covariance matrix of estimates

ocov the $n \times n$ covariance matrix of `fopt` w.r.t. x

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

Relationships: fider(), auroc()

Examples: 1. see `cmat\test\tdelta.inp` for an example.

5.5 Function `hbttest`

< ttest,mom,ftest > = hbttest(dat1,dat2<,optn>)

Purpose: This function implements two forms of the common t test

1. univariate: for a given vector of measurement its mean is compared to a prespecified value (many times, but not always, zero)
2. bivariate: the means of two measurement vectors are tested for equality

and Satterthwaite's bivariate t test where the means of two measurement vectors are tested for inequality. Only for the bivariate form, an F test for the equality of the variances is added. (The algorithm is described in Nollau (1975, pp. 101).

Input: The function needs two or three input arguments:

dat1 this is a vector of n_1 measurements ($p = 1$) or an $n1 \times p$ matrix of measurements ($p > 1$);

dat2
- for the univariate test: this is a scalar mean value or a p vector of means (for $p = 1$);
- for the bivariate test: this is a vector of n_2 measurements or an $n2 \times p$ matrix of measurements ($p > 1$).

optn this is optional and specifies some options in vector form:

1. amount of printed output (=0: no printed output); default is 2 for all printed output;
2. the probability α, default is $\alpha = .05$.

The input data `dat1` and `dat2` must agree in the number p of samples (tests).

Output: There are two or three return arguments:

ttest The number of rows of this return argument depends on the type of data resp. test:
- for the univariate t test: this is a 1×5 row vector containing critical t value, the actual T value, the number of degrees of freedom, and the $P > |T|$ probability;
- for the bivariate t test: the 2×5 matrix contains an additional row with the results for Satterthwaite's test.

If the input data consists of $p > 1$ columns the `ttest` return argument will have either p or $2p$ rows.

mom The number of rows of this return argument depends on the type of data:

- for the univariate t test: this is a 1×5
- for the bivariate t test: the 2×5 matrix

and contains the mean, standard deviation, the asymptotic standard error, and the minimum, and maximum of the measurements (in each of the p columns).

ftest only for the bivariate t test contains the results for an equality F test of the variances in form of a 1×4 row vector. For the univariate test this returns a missing value.

Restrictions:

1. The size of `dat1` and `dat2` must agree w.r.t the number of tests p.
2. The input data must be numeric and cannot contain any string or complex data.
3. Currently the input data must not have any missing values.

Relationships: hbanova()

Examples:

1. Univariate Test (Nollau, 1975, p. 103):

```
print "Nollau, p. 103: Univariate Test with given mue";
rho = [ 1.0768 1.0772 1.0766 1.0765 1.0774
        1.0771 1.0775 1.0770 1.0769 1.0770 ];
mue = 1.0771;
< ttest,mom,ftest > = hbttest(rho,mue);
print "Result Univariate Test=",ttest;
print "Moments=",mom;
print "FTest=",ftest;
```

```
                    Some Features of Input Data
                    ***************************

                     Dense Row Vector (ncol=5)

      R |        Mean     StdDev    AStdErr    Minimum    Maximum
 Sample |   1.0770000  3.20e-004  1.01e-004  1.0765000  1.0775000


                        Univariate T Test
                        *****************

                     Dense Row Vector (ncol=4)

          R |      Crit_T    T_Value        NDF       Prob
      TTest |   2.2621572 -0.9890707  9.0000000  0.3484723
```

2. Bivariate Test (Nollau, 1975, p. 103):

```
print "Nollau, p. 107: Bivariate Test";
mat1 = [ 25.30 26.08, 26.75 28.59, 31.09 34.23, 31.27 29.88,
         30.17 31.73, 29.90 32.14, 32.81 32.10, 31.11 32.90,
         30.35 31.38, 30.76 31.74, 30.57 32.50, 38.05 39.84 ];
mat2 = [ 17.98 19.41, 21.35 24.13, 26.01 28.19, 24.28 24.54,
         23.25 25.72, 23.33 26.04, 23.59 25.15, 23.52 26.65,
         24.82 27.16, 25.58 28.45, 29.04 30.86, 34.61 35.83 ];
dat1 = mat1[,2] - mat1[,1];
dat2 = mat2[,2] - mat2[,1];
optn = [ 2, .01];
< ttest,mom,ftest > = hbttest(dat1,dat2,optn);
print "Result Bivariate Test=",ttest;
print "Moments=",mom;
print "FTest=",ftest;
```

Some Features of Input Data

Dense Matrix (2 by 5)

	Mean	StdDev	AStdErr	Minimum	Maximum
Sample1	1.2483333	1.2502860	0.3609265	-1.3900000	3.1400000
Sample2	2.0641667	0.8340859	0.2407799	0.2600000	3.1300000

Bivariate T Test

Dense Matrix (2 by 4)

	Crit_T	T_Value	NDF	Prob
TTest	2.8187561	-1.8803651	22.000000	0.0733640
Sattw	.	-1.8803651	19.172321	0.0753392

F Test for Equal Variances

Dense Row Vector (ncol=4)

R	F_Value	NDF1	NDF2	Prob
FTest	2.2469697	11.000000	11.000000	0.1951366

3. Bivariate Test (SAS/STAT Manual, p.1638):

```
print "Example of SAS/STAT Manual, p. 1638";
dat1 = [ 75 76 80 77 80 77 73 ];
dat2 = [ 82 80 85 85 78 87 82 ];
< ttest, mom,ftest > = hbttest(dat1,dat2);
print "Result Bivariate Test=",ttest;
print "Moments=",mom;
print "FTest=",ftest;
```

```
                Some Features of Input Data
                ***************************

                   Dense Matrix (2 by 5)

        |        Mean     StdDev    AStdErr    Minimum    Maximum
--------------------------------------------------------------------
Sample1 |   76.857143  2.5448360  0.9618576  73.000000  80.000000
Sample2 |   82.714286  3.1471832  1.1895234  78.000000  87.000000


                      Bivariate T Test
                      ****************

                   Dense Matrix (2 by 4)

          |      Crit_T    T_Value        NDF       Prob
-----------------------------------------------------------
   TTest  |   2.1788128 -3.8288228  12.000000  0.0024007
   Sattw  |      .      -3.8288228  11.496373  0.0025888


                F Test for Equal Variances
                **************************

                Dense Row Vector (ncol=4)

       R  |     F_Value       NDF1       NDF2       Prob
-----------------------------------------------------------
   FTest  |   1.5294118  6.0000000  6.0000000  0.6188688
```

4. Univariate Test (SAS/STAT Manual, p.1639): this is the common case for paired comparison implemented in SAS PROC MEANS:

```
print "Example of SAS/STAT Manual, p. 1639";
print "Paired Comparison using PROC MEANS";
mat = [ 80  82, 73  71, 70  95, 60  69,
```

```
        88 100, 84  71, 65  75, 37  60,
        91  95, 98  99, 52  65, 78  83,
        40  60, 79  86, 59  62 ];
dat = mat[,2] - mat[,1];
mue = 0.;
< ttest,mom > = hbttest(dat,mue);
print "Result Univariate Test=",ttest;
print "Moments=",mom;
```

```
                Some Features of Input Data
                ***************************

                 Dense Row Vector (ncol=5)

     R |        Mean     StdDev    AStdErr    Minimum    Maximum
Sample |   7.9333333  9.9316713  2.5643465 -13.000000  25.000000


                    Univariate T Test
                    *****************

                 Dense Row Vector (ncol=4)

         R |      Crit_T    T_Value        NDF       Prob
     TTest |   2.1447867  3.0937057  14.000000  0.0079309
```

5.6 Function `lrallv`

<gof,best,parm,yptr,yptt> = lrallv(msel,trn,model,optn<,class<,aov16<,grpv<,tst> . >)

Purpose: The `lrallv` function implements two algorithms for the all variables subsets linear regression of m from given n, $m \ll n$, different variables:

- the common full enumeration algorithm
- a stochastic search algorithm which generates s different samples of m variables

Each of the two algorithms tabulates the b *best* solutions from the generated variable sets. With *best* we understand w.r.t. the R^2 value if the response is interval. If the response is binary and `selc` is specified as `"acc"` the solutions are sorted w.r.t. the classification accuracy. Note, that the latter needs more computer time, since for each subset not only the least squares solution but also a logistic regression must be computed.

For large n or relatively large m the full enumeration can be computationally very expensive which was the reason for developing that stochastic search algorithm.

The univariate response should be either interval or binary. The algorithm is based on Miller (2002) and Foster & Stine (2004) and is similar to the `lrforw` function.

There are three major parts of the algorithm:

1. Ranking all predictor variables, including the derived AOV16 and Group variables by descending univariate R^2 value.
2. combinations of m variables are evaluated in two different ways and the best $b > 0$ w.r.t. a specific criterion are tabled:
   - if `"samp"=s` with $s > 0$ is specified: *promising* solutions (based on the univariate R^2 values) of the variables are randomly generated and evaluated;
   - if `"samp"=s` is not specified, or $s = 0$: all $Nc = \left(\begin{array}{c} n \\ m \end{array} \right)$ combinations of subsets of size m from a total of n variables (see `combn()` function) are evaluated
3. Only when the response is specified as a binary variable, an approximate logistic model selection (with the estimation of two additional parameters) which results in binary predicted values with a 2 by 2 classification table. Note that the two parameters of the logistic model are estimated for given (fixed) linear coefficients resulting from step 2.

A note is printed indicating the number of subsets which are dropped in the analysis due to singularities.

Input: msel must be an integer $m > 0$ specifying the size of the subsets of selected variables;

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

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

optn : The `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 the table below for content.

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

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

grpv a list of column numbers of multinomial Class variables

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

Option	Second Column	Meaning
`"aov"`		using *AOV16* variables
`"best"`	int	number b of best solutions returned
`"cutof"`	real	cutoff for accuracy table (def=.5); (valid only for binary response)
`"grp"`		using *Group* variables
`"maxc"`	int	upper range for number of columns in $\mathbf{X}^T\mathbf{X}$ matrix (def=4000);
`"pobs"`	int	print observationwise statistics: predicted $\hat{y}$, residuals
`"print"`	int	amount of printed output =0 specifies no printed output, default is 2
`"r2min"`	real	lower R2 threshold for selection this raw (unpartial) R^2 value from first step: only variables (effects) with larger R^2 values are candidates for selection in step 2.
`"r2st"`	real	lower threshold for R^2 or upper threshold for χ^2 for terminating the selection process; the R^2 or χ^2 value is now based on the partial $\mathbf{X}^T\mathbf{X}$ matrix
`"sing"`	real	singularity criterion for inverting the $\mathbf{X}^T\mathbf{X}$ matrix; default is run-time option, which defaults to 1.e-8
`"selc"`	string	specifies the selection criterion
	"r2"	use R^2 squared value
	"acc"	classification accuracy for binary target

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

best the $b \times c$ table of best solutions, where the rows of the table are sorted w.r.t. the values in the column specified with the `selc` option; the first m of the c columns contain the index values of the *best* effects found and the last columns the corresponding values of selection criteria for that subset

parm the $b \times n$ parameter matrix with the coefficients of the linear model for the b best solutions (note, for large n and relatively small m this matrix is sparse, since only $m + 1$ values in each of the rows are nonzero

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

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

There are currently seven more columns for the selection criteria of that variable subset:

- R^2 squared value
- probability of $F(n, d)$ value
- AIC value of Akaike's information criterion
- BIC value of Bayesian information criterion

- Schwarz SBC criterion
- adjusted R^2 value
- Mallow's Cp value

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

2. The response cannot be multinomial.

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

Examples: 1. Small ($N = 30, n = 12$) Housing Data: select all combinations of $m = 3$ variables

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


modl = "14 = 1:3 5:13";
optn = [ "print"         3 ,
         "best"         10 ,
         "sing"     1.e-12 ,
         "pobs"            ];
< gof,best,parm,yptr > = lrallv(3,hous30,modl,optn);
```

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

 Number Valid Observations   30
 Response Variable         Y[14]
 N Independend Variables     12

All Variables Subset Regression
  Full Enumeration Algorithm

 Number Selected Variables    3
 Number Best Solutions       10
 Selection Criterion    Rsquare
 Number Parameters           14
 Without AOV16 Variables
 Without Group Variables
```

Lower R2 Threshold 1.00e-006
Lower Threshhold for 1.00e-006
Max Variables in Model 3000

****
Model Effects
****

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

****
Simple Statistics
****

Column	Nobs	Mean	Std Dev	Skewness	Kurtosis
Y[14]	30	20.676667	6.0334520	1.2773229	1.1330363
X[1]	30	0.5274283	0.4134794	0.0910955	-1.4657563
X[2]	30	3.5166667	6.0078732	1.2192018	-0.3107288
X[3]	30	7.2153335	2.0146323	-2.1942896	3.1550181
X[5]	30	0.5221333	0.0278799	-1.7379671	1.4252539
X[6]	30	6.1293000	0.4466720	0.9520823	0.4522725
X[7]	30	75.606666	20.469000	-0.7661226	-0.4963511
X[8]	30	4.9001134	0.8887523	0.5492360	-1.1704674
X[9]	30	3.9000000	0.9595257	-1.2947809	2.1178516
X[10]	30	294.73333	29.852434	-1.9249275	2.0198509
X[11]	30	19.013333	2.5319658	-0.7049787	-1.3354936
X[12]	30	381.79467	28.624775	-2.6251234	5.8149042
X[13]	30	13.133667	6.0869739	0.4297902	0.5087852

****
Rank Order of Rsquare of Effects for Target medv
****

Rank	Effect		Rsquared	df
1	4	nox	0.632830510	1
2	9	tax	0.617942270	1
3	5	rm	0.591736634	1
4	3	indus	0.505119633	1
5	12	lstat	0.501338746	1
6	1	crim	0.396981746	1
7	6	age	0.343752149	1
8	8	rad	0.226321823	1
9	7	dis	0.175581418	1

10	11	b	0.092501601	1
11	10	ptrat	0.084159355	1
12	2	zn	0.000155577	1

There are 220 regression computations with 3 coefficients.

Parameter Estimates

****

Sparse Matrix (10 by 13)

1	Intercept	nox	rm	age
	29.704658	-73.303988	6.1644743	-0.1129193
2	Intercept	crim	rm	age
	-16.058281	-3.4041231	7.7899236	-0.1218993
3	Intercept	rm	age	tax
	5.1034706	6.6597389	-0.1139590	-0.0564248
4	Intercept	rm	age	dis
	-26.508245	8.4022177	-0.1350386	1.2030366
5	Intercept	rm	age	ptrat
	-17.172875	8.8282485	-0.1353124	-0.3171944
6	Intercept	indus	rm	age
	-14.217351	-0.5322200	7.8019614	-0.1201799
7	Intercept	rm	age	b
	-29.243188	8.7613452	-0.1383473	0.0174935
8	Intercept	zn	rm	age
	-25.875534	0.0646518	9.2378475	-0.1361877
9	Intercept	zn	nox	lstat
	91.065863	0.2036271	-124.26805	-0.4736526
10	Intercept	nox	ptrat	lstat
	91.330335	-104.53502	-0.5066534	-0.4902883

****

Final ANOVA Table for Target: medv

****

Effect	DF	Rsquared	Sum-of-Squares
Model	3	0.84512948	892.1810146644
Error	26	.	163.4927442569
Total	29	.	1055.673758921

MSE	6.28818247	SBC	37.2623474
AIC	58.8671369	BIC	62.0505689
F	47.2940588	Prb	1.142e-010
CP	4.00000000	AR2	0.82725980

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

Effect	DF	Rsquared	F_value	pval	Sum-of-Squares
crim	1	0.39698175	0	0.0000	419.0832117665
zn	1	1.556e-004	0	0.0000	0.164238393328
indus	1	0.50511963	0	0.0000	533.2415417960
dis	1	0.17558142	0	0.0000	185.3566954307
rad	1	0.22632182	0	0.0000	238.9220091168
tax	1	0.61794227	0	0.0000	652.3454393900
ptrat	1	0.08415935	0	0.0000	88.84482213020
b	1	0.09250160	0	0.0000	97.65151257526
lstat	1	0.50133875	0	0.0000	529.2501586266

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

Nobs	Yobs	Yprd	Residual
1	24.0000000	23.4361958	0.56380425
2	21.6000004	25.9978469	-4.39784650
3	34.7000008	32.7174681	1.98253266
4	33.4000015	34.0987217	-0.69872015
5	36.2000008	34.0687045	2.13129629
.....			
25	15.6000004	16.1597560	-0.55975560
26	13.8999996	15.1048250	-1.20482535
27	16.6000004	15.9045946	0.69540581
28	14.8000002	17.5164588	-2.71645860
29	18.3999996	19.6457953	-1.24579569
30	21.0000000	21.5509622	-0.55096219

Sum-of-Squared Residual 163.493

```
******************************
Table of 10 best Rsquare Values
******************************

 1   nox    rm   age  0.84512948
 2    rm  crim   age  0.83308442
 3   tax    rm   age  0.82540196
 4    rm   age   dis  0.82135857
 5    rm   age ptrat  0.81052513
 6    rm indus   age  0.80966733
 7    rm   age     b  0.79937788
 8    rm   age    zn  0.79777936
 9   nox lstat    zn  0.79767851
10   nox lstat ptrat  0.79617656
```

```
******************************
Table of all Selection Criteria
******************************
```

N	Rsquare	Prob(F)	AIC	BIC	SBC
1	0.84512948	1.142e-010	58.8671369	62.0505689	37.2623474
2	0.83308442	3.003e-010	61.1140995	64.2975315	39.5093100
3	0.82540196	5.367e-010	62.4640475	65.6474794	40.8592579
4	0.82135857	7.211e-010	63.1508724	66.3343043	41.5460828
5	0.81052513	1.541e-009	64.9171458	68.1005778	43.3123563
6	0.80966733	1.633e-009	65.0526577	68.2360896	43.4478681
7	0.79937788	3.219e-009	66.6321493	69.8155812	45.0273598
8	0.79777936	3.566e-009	66.8702360	70.0536679	45.2654465
9	0.79767851	3.589e-009	66.8851937	70.0686257	45.2804042
10	0.79617656	3.948e-009	67.1070786	70.2905106	45.5022891

Goodness of Fit

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

Dense Row Vector (ncol=25)

R |	ModelNpar	SSQ_model	SSQ_error	SSQ_total	DF_model
	13.000000	892.18101	163.49274	1055.6738	3.0000000

R |	DF_error	DF_total	MSE	Rsquare	SBC

```
      26.000000  29.000000  6.2881825  0.8451295  47.294059

R |        AIC        BIC     Fvalue      Fprob   MallowCp
      1.14e-010  37.262347  58.867137  62.050569  4.0000000

R |      AdjR2   LogInter   LogSlope        N00        N01
      0.8272598          0          0     .          .

R |        N10        N11   ClassErr   Accuracy     unused
         .          .          .          .               0


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

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


                  Total Processing Time: 0
```

2. Small ($N = 30, n = 12$) Housing Data: From the 220 combinations we now sample only $s = 100$, which is close to 50 % of all:

```
modl = "14 = 1:3 5:13";
optn = [ "print"         3 ,
         "best"         10 ,
         "samp"        100 ,
         "sing"     1.e-12 ,
         "seed"        123 ,
         "pobs"            ];
< gof,best,parm,yptr > = lrallv(3,hous30,modl,optn);
```

We show only the important part of the output:

```
There are  220 regression computations with 3 coefficients.
Using 100 sample ( 45.455 \%) out of  220 possible combinations.


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

               Sparse Matrix (10 by 13)

  1 |   Intercept        nox         rm        age
        29.704658 -73.303988  6.1644743 -0.1129193

  2 |   Intercept       crim         rm        age
       -16.058281 -3.4041231  7.7899236 -0.1218993

  3 |   Intercept         rm        age        tax
        5.1034706  6.6597389 -0.1139590 -0.0564248

  4 |   Intercept      indus         rm        age
       -14.217351 -0.5322200  7.8019614 -0.1201799

  5 |   Intercept        nox      ptrat      lstat
        91.330335 -104.53502 -0.5066534 -0.4902883

  6 |   Intercept         rm        age        rad
       -29.349028  9.6343422 -0.1378157  0.3573614

  7 |   Intercept         rm        age      lstat
       -27.756272  9.5711725 -0.1446264  0.0535320

  8 |   Intercept        tax      ptrat      lstat
        69.367729 -0.0968565 -0.7498627 -0.4482242

  9 |   Intercept        nox         rm      lstat
        54.293027 -96.628642  3.4543642 -0.3301561

 10 |   Intercept        nox        tax      lstat
        119.11707 -250.19496  0.1329982 -0.5333076
```

The best solution found by sampling is the same as with full enumeration:

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

```
Effect    DF    Rsquared  Sum-of-Squares

 Model     3  0.84512948  892.1810146644
 Error    26           .   163.4927442569
 Total    29           .   1055.673758921

 MSE    6.28818247       SBC  37.2623474
 AIC    58.8671369       BIC  62.0505689
 F      47.2940588       Prb  1.142e-010
 CP     4.00000000       AR2  0.82725980
```

But only a few of the the first solutions are the same as with full enumeration, which is not surprising:

```
*******************************
Table of 10 best Rsquare Values
*******************************

 1   nox    rm   age  0.84512948
 2    rm  crim   age  0.83308442
 3   tax    rm   age  0.82540196
 4    rm indus   age  0.80966733
 5   nox lstat ptrat  0.79617656
 6    rm   age   rad  0.79575753
 7    rm lstat   age  0.79450456
 8   tax lstat ptrat  0.78819035
 9   nox    rm lstat  0.78776692
10   nox   tax lstat  0.78337476
```

3. Small ($N = 30, n = 12$) Housing Data: The forward selection algorithm performs much worse than sampling:

```
modl = "14 = 1:3 5:13";
optn = [ "print"         3 ,
         "maxst"         3 ,
         "selc"       "r2" ,
         "sing"     1.e-12 ,
         "pobs"            ];
< gof,parm,yptr > = lrforw(hous30,modl,optn);
```

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

(first row incremental, second row current status)
```

```
Step  Effect  DF Rsquared    F_value    pval        SSE        BIC

   1     nox   1   0.6328  48.259060  0.0000  668.06256
               2   0.6328  24.129530  0.0000  387.61120  83.039672
   2   lstat   1   0.1273  14.324327  0.0008  134.35838
               3   0.7601  28.516122  0.0000  253.25282  72.637702
   3      zn   1   0.0376  4.8287485  0.0371  39.667331
               4   0.7977  25.627086  0.0000  213.58549  70.068626


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

          Sparse Row Vector (ncol=13)

R |   Intercept         zn        nox      lstat
      91.065863  0.2036271 -124.26805 -0.4736526


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

            Effect    DF    Rsquared  Sum-of-Squares

             Model     3  0.79767851  842.0882694075
             Error    26           .   213.5854895138
             Total    30           .   1055.673758921

             MSE    8.21482652       SBC  45.2804042
             AIC    66.8851937       BIC  70.0686257
             F      34.1694483       Prb  3.589e-009
             CP     4.00000000       AR2  0.77433372
```

4. Small ($N = 30, n = 12$) Housing Data: select all combinations of $m = 6$ variables:

```
modl = "14 = 1:3 5:13";
optn = [ "print"         3 ,
         "best"         10 ,
         "sing"     1.e-12 ,
         "pobs"            ];
< gof,best,parm,yptr,yptt > = lrallv(6,hous30,modl,optn);
print "GOF=",gof;
```

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

Number Valid Observations   30
Response Variable         Y[14]
N Independend Variables     12

All Variables Subset Regression
 Full Enumeration Algorithm

Number Selected Variables    6
Number Best Solutions       10
Selection Criterion    Rsquare
Number Parameters           14
Without AOV16 Variables
Without Group Variables
Lower R2 Threshold   1.00e-006
Lower Threshhold for 1.00e-006
Max Variables in Model    3000

There are 924 regression computations with 6 coefficients.

Update failures are due to the following variables:

                    ptrat    zn

[warning] The algorithm encountered 25 unsuccessful updates with 2
          different effects.

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

Effect    DF    Rsquared  Sum-of-Squares

Model      6  0.86765021  915.9555554236
Error     23           .   139.7182034977
Total     29           .   1055.673758921

MSE    6.07470450       SBC  22.3445237
AIC    60.1529054       BIC  66.2285198
F      25.1303186       Prb  5.168e-009
```

CP 7.00000000 AR2 0.83312417

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

Effect	DF	Rsquared	F_value	pval	Sum-of-Squares
zn	1	1.556e-004	0	0.0000	0.164238393328
indus	1	0.50511963	0	0.0000	533.2415417960
nox	1	0.63283051	0	0.0000	668.0625631257
ptrat	1	0.08415935	0	0.0000	88.84482213020
b	1	0.09250160	0	0.0000	97.65151257526
lstat	1	0.50133875	0	0.0000	529.2501586266

****
Table of 10 best Rsquare Values
****

1	tax	rm	crim	age	rad	dis	0.86765021
2	nox	rm	crim	age	rad	dis	0.86353629
3	tax	rm	crim	age	rad	b	0.86157970
4	tax	rm	lstat	crim	age	rad	0.86095652
5	nox	tax	rm	crim	age	rad	0.86049884
6	tax	rm	indus	crim	age	rad	0.86027895
7	tax	rm	crim	age	rad	ptrat	0.86009062
8	tax	rm	crim	age	rad	zn	0.85996625
9	tax	rm	lstat	age	rad	ptrat	0.85955132
10	tax	rm	lstat	age	rad	zn	0.85946231

****
Table of all Selection Criteria
****

N	Rsquare	Prob(F)	AIC	BIC	SBC
1	0.86765021	5.168e-009	60.1529054	66.2285198	22.3445237
2	0.86353629	7.285e-009	61.0712149	67.1468293	23.2628333
3	0.86157970	8.546e-009	61.4982956	67.5739100	23.6899140
4	0.86095652	8.987e-009	61.6330562	67.7086705	23.8246745
5	0.86049884	9.324e-009	61.7316423	67.8072567	23.9232606
6	0.86027895	9.490e-009	61.7788930	67.8545073	23.9705113
7	0.86009062	9.635e-009	61.8193019	67.8949163	24.0109203
8	0.85996625	9.731e-009	61.8459592	67.9215736	24.0375775
9	0.85955132	1.006e-008	61.9347191	68.0103335	24.1263374

```
  10  0.85946231  1.013e-008  61.9537263  68.0293407  24.1453447
```

Goodness of Fit
****

Dense Row Vector (ncol=25)

```
R |   ModelNpar  SSQ_model  SSQ_error  SSQ_total   DF_model
      13.000000  915.95556  139.71820  1055.6738  6.0000000

R |    DF_error   DF_total        MSE    Rsquare        SBC
      23.000000  29.000000  6.0747045  0.8676502  25.130319

R |         AIC        BIC     Fvalue      Fprob   MallowCp
      5.17e-009  22.344524  60.152905  66.228520  7.0000000

R |       AdjR2   LogInter   LogSlope        N00        N01
      0.8331242          0          0     .          .

R |         N10        N11   ClassErr   Accuracy     unused
          .          .          .          .               0
```

Parameter Estimates
****

Dense Matrix (10 by 13)

```
    |   Intercept       crim         zn      indus        nox
----------------------------------------------------------------
  1 |   13.624479 -5.1564085          0          0          0
  2 |   31.928477 -3.9234993          0          0 -77.705415
  3 |   4.2341542 -2.3464070          0          0          0
  4 |   7.3587678 -2.6119002          0          0          0
  5 |  -5.1762171 -3.0800574          0          0  40.229913
  6 |   5.4784392 -2.4277299          0 -0.0960748          0
  7 |   3.6472324 -2.9594778          0          0          0
  8 |   5.6144451 -2.5738456 -0.0027504          0          0
  9 |   21.334569          0          0          0          0
 10 |   17.085189          0  0.1649547          0          0

    |          rm        age        dis        rad        tax
----------------------------------------------------------------
  1 |   6.8863835 -0.1014344 -1.8128687  1.9286203 -0.0794146
  2 |   6.9286790 -0.1070948 -1.6388562  1.2948330          0
```

```
 3 |   6.0714938 -0.1115701          0  0.6979563 -0.0594829
 4 |   6.0145673 -0.0978472          0  0.8759447 -0.0586389
 5 |   6.3651197 -0.1077230          0  0.9642621 -0.0955364
 6 |   6.3118281 -0.1088296          0  0.7694408 -0.0552627
 7 |   6.3763779 -0.1089599          0  0.7434140 -0.0557667
 8 |   6.3445513 -0.1095983          0  0.7222421 -0.0576407
 9 |   5.4662402 -0.0872921          0  0.9854717 -0.0743588
10 |   5.4975578 -0.0868838          0  1.1555858 -0.0904234
```

```
   |       ptrat          b      lstat
-----------------------------------------
 1 |           0          0          0
 2 |           0          0          0
 3 |           0  0.0097201          0
 4 |           0          0 -0.0689113
 5 |           0          0          0
 6 |           0          0          0
 7 |   0.0674629          0          0
 8 |           0          0          0
 9 |  -0.3903704          0 -0.1574075
10 |           0          0 -0.1501334
```

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

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


                     Total Processing Time: 0
```

5. Small ($N = 30, n = 12$) Housing Data: randomly select only 100 of all 924 combinations of $m = 6$ variables:

```
There are  924 regression computations with 6 coefficients.
Using 100 sample ( 10.823 \%) out of  924 possible combinations.
```

The best solution found is the 4th best of those found by full enumeration, but it still is close in goodness of fit to that found by full enumeration:

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

Effect    DF    Rsquared  Sum-of-Squares

 Model     6  0.86095652  908.8892008160
 Error    23           .   146.7845581053
 Total    29           .   1055.673758921

 MSE    6.38193731       SBC  23.8246745
 AIC    61.6330562       BIC  67.7086705
 F      23.7359796       Prb  8.987e-009
 CP     7.00000000       AR2  0.82468430


*******************************
Table of 10 best Rsquare Values
*******************************

 1   tax    rm lstat  crim   age   rad  0.86095652
 2   nox   tax    rm  crim   age   rad  0.86049884
 3   tax    rm indus  crim   age   rad  0.86027895
 4   nox    rm lstat  crim   age   rad  0.85927702
 5   nox    rm indus  crim   age   rad  0.85904801
 6   nox    rm  crim   age   rad     b  0.85872166
 7   nox   tax    rm indus lstat   age  0.85830915
 8   nox    rm indus  crim   age     b  0.85789284
 9   nox   tax    rm indus  crim   age  0.85643709
10   nox   tax    rm indus   age   dis  0.85643677


*******************************
Table of all Selection Criteria
*******************************

N     Rsquare     Prob(F)         AIC         BIC         SBC
1  0.86095652  8.987e-009  61.6330562  67.7086705  23.8246745
2  0.86049884  9.324e-009  61.7316423  67.8072567  23.9232606
```

3	0.86027895	9.490e-009	61.7788930	67.8545073	23.9705113
4	0.85927702	1.028e-008	61.9932522	68.0688666	24.1848706
5	0.85904801	1.047e-008	62.0420345	68.1176489	24.2336528
6	0.85872166	1.075e-008	62.1114142	68.1870286	24.3030325
7	0.85830915	1.110e-008	62.1988811	68.2744955	24.3904995
8	0.85789284	1.147e-008	62.2868976	68.3625120	24.4785159
9	0.85643709	1.286e-008	62.5926549	68.6682693	24.7842732
10	0.85643677	1.286e-008	62.5927209	68.6683352	24.7843392

6. Small ($N = 30, n = 12$) Housing Data: forward regression including $m = 6$ variables:

```
modl = "14 = 1:3 5:13";
optn = [ "print"         3 ,
         "maxst"         6 ,
         "selc"       "r2" ,
         "sing"     1.e-12 ,
         "pobs"            ];
< gof,parm,yptr,yptt > = lrforw(hous30,modl,optn);
print "GOF=",gof;
print "Parm=",parm;
```

The first three selection steps are the same as above which were not so great:

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

          (first row incremental, second row current status)

 Step  Effect  DF Rsquared    F_value    pval        SSE        BIC

    1     nox   1   0.6328  48.259060  0.0000  668.06256
                2   0.6328  24.129530  0.0000  387.61120  83.039672
    2   lstat   1   0.1273  14.324327  0.0008  134.35838
                3   0.7601  28.516122  0.0000  253.25282  72.637702
    3      zn   1   0.0376  4.8287485  0.0371  39.667331
                4   0.7977  25.627086  0.0000  213.58549  70.068626
    4      rm   1   0.0178  2.4043806  0.1336  18.739369
                5   0.8154  22.089935  0.0000  194.84612  70.050382
    5     age   1   0.0386  6.3401585  0.0189  40.716837
                6   0.8540  23.397098  0.0000  154.12928  65.972831
    6     rad   1   0.0046  0.7451282  0.3969  4.8366166
                7   0.8586  19.948129  0.0000  149.29267  68.216950
```

Parameter Estimates
****

Dense Row Vector (ncol=13)

R	Intercept	crim	zn	indus	nox
	35.583018	0	0.1012489	0	-81.674060

R	rm	age	dis	rad	tax
	5.5009643	-0.0879234	0	0.6298342	0

R	ptrat	b	lstat
	0	0	-0.1632049

Again, stepwise forward solution is not performing as well as the sampling algorithm:

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

Effect	DF	Rsquared	Sum-of-Squares
Model	6	0.85858068	906.3810923188
Error	23	.	149.2926666025
Total	30	.	1055.673758921

MSE	6.49098550	SBC	24.3329544
AIC	62.1413361	BIC	68.2169505
F	23.2728166	Prb	1.087e-008
CP	7.00000000	AR2	0.82168868

There are, however, examples where sampling is doing worse than stepwise forward regression.

5.7 Function noblanks

b = noblanks(a<,str>)

Purpose: The noblanks function removes leading and trailing white space from string data and optionally replaces inside white space.

Input: **a** must be either a string scalar or a vector or mnatrix containing strings.

str is an optional string argument specifying which string should replace the removed inside white space, it defaults to (underscore) _.

Output: The only return is an object b which is the same as the input a except that the string data have been modified.

Restrictions:

Relationships: byte()

Examples: 1. Scalar string data:

```
st1 = "   blank padded   string   ";
print "st2=", st2 = noblanks(st1);
print "st3=", st2 = noblanks(st1," ");
```

```
st2=blank_padded_string
st3=blank padded string
```

2. Mixed type matrix:

```
fid = fopen("..\\tdata\\cgd1.dat","r");
form = "%d%d%22c%11c%8c%7c%d%g%g";
cgd1 = fscanf(fid,form,.,9);
nr = nrow(cgd1); nc = ncol(cgd1);
print "CGD1: nr=",nr," nc=",nc;
cnam = [" N id center random treat sex age height weight "];
cgd1 = cname(cgd1,cnam);
print "CGD1=", cgd1[1:100,];
```

```
CGD1: nr= 203 nc= 9

CGD1=
                    Mixed Type Matrix (100 by 9)

          N  id                center      random    treat     sex
```

```
  1   1   1      Scripps Institute  1989-06-07   rIFN-g  female
  2   2   1      Scripps Institute  1989-06-07   rIFN-g  female
  3   3   1      Scripps Institute  1989-06-07   rIFN-g  female
  4   4   2      Scripps Institute  1989-06-07  placebo    male
 ..............................................................
 97  97  53   Mott Children's Hosp  1989-09-14  placebo    male
 98  98  53   Mott Children's Hosp  1989-09-14  placebo    male
 99  99  53   Mott Children's Hosp  1989-09-14  placebo    male
100 100  53   Mott Children's Hosp  1989-09-14  placebo    male
```

```
      age    height   weight
 1    12  147.0000  62.0000
 2    12  147.0000  62.0000
 3    12  147.0000  62.0000
 4    15  159.0000  47.5000
 ........................
97     5  114.5000  23.0000
98     5  114.5000  23.0000
99     5  114.5000  23.0000
100    5  114.5000  23.0000
```

```
cgd2 = noblanks(cgd1);
print "CGD2=",cgd2[1:100,];
```

```
      N  id               center     random   treat    sex age
  1   1   1     Scripps_Institute 1989-06-07  rIFN-g female  12
  2   2   1     Scripps_Institute 1989-06-07  rIFN-g female  12
  3   3   1     Scripps_Institute 1989-06-07  rIFN-g female  12
  4   4   2     Scripps_Institute 1989-06-07 placebo   male  15
 ..............................................................
 97  97  53  Mott_Children's_Hosp 1989-09-14 placebo   male   5
 98  98  53  Mott_Children's_Hosp 1989-09-14 placebo   male   5
 99  99  53  Mott_Children's_Hosp 1989-09-14 placebo   male   5
100 100  53  Mott_Children's_Hosp 1989-09-14 placebo   male   5
```

```
       height   weight
 1   147.0000  62.0000
 2   147.0000  62.0000
 3   147.0000  62.0000
 4   159.0000  47.5000
 ....................
97   114.5000  23.0000
98   114.5000  23.0000
```

```
                    99  114.5000  23.0000
                   100  114.5000  23.0000
```

```
cgd3 = noblanks(cgd1," ");
print "CGD3=",cgd3[1:100,];
```

```
                Mixed Type Matrix (100 by 9)

       N  id               center     random   treat    sex age
   1   1   1     Scripps Institute 1989-06-07  rIFN-g female  12
   2   2   1     Scripps Institute 1989-06-07  rIFN-g female  12
   3   3   1     Scripps Institute 1989-06-07  rIFN-g female  12
   4   4   2     Scripps Institute 1989-06-07 placebo   male  15
  ..........................................................
  97  97  53  Mott Children's Hosp 1989-09-14 placebo   male   5
  98  98  53  Mott Children's Hosp 1989-09-14 placebo   male   5
  99  99  53  Mott Children's Hosp 1989-09-14 placebo   male   5
 100 100  53  Mott Children's Hosp 1989-09-14 placebo   male   5
```

```
                      height   weight
                   1  147.0000  62.0000
                   2  147.0000  62.0000
                   3  147.0000  62.0000
                   4  159.0000  47.5000
                  .....................
                  97  114.5000  23.0000
                  98  114.5000  23.0000
                  99  114.5000  23.0000
                 100  114.5000  23.0000
```

5.8 Function `order`

indx = order(vec_1,vec_2,...)

Purpose: The `order` function finds the indices for the hierarchical search of a number of data vectors. The order of the input arguments specifies the hierarchy of the sortings:

1. First are the ranks determined w.r.t. ascending data values of the first input argument `vec_1`.
2. Inside the tie blocks of the first input vector the indices are ranked w.r.t. to ascending values of the second input argument `vec_2`.
3. The algorithm is terminated either when there are no more blocks of ties left or when all input arguments are processed.

Note, that sorting is done ascending. The descending sorting is simply achieved by specifying the negative data vector. (See below for an example.) This function is similar to the R function with the same name.

Input: There can be $m \geq 1$ vectors, all of size n. The order of the input arguments is important.

Output: The result is an n vector of indices. By default the indices are 1-based, but could also be zero based when specifying the `INDBASE` runtime option.

Restrictions: 1. The input arguments cannot have any string data or missing values.

2. Complex data are sorted according to absolute value (cmod).
3. All input vectors must have the same size.

Relationships: index operators: <, >, <:, >:, < |, > |, <!, >!, sortrow(), branks(), ranktie()

Examples: 1. The Myeloma data are also used for survival regression and survival curves:

```
  options NOECHO;
#include "..\\tdata\\myeloma1.dat"
  options ECHO;

  cnam = [" Time Vstatus LogBUN HGB Platelet Age LogWBC Frac
            LogPBM Protein SCalc "];
  myeloma1 = cname(myeloma1,cnam);
```

We want the hierarchical ranking of:

(a) column 5= `group`: ascending
(b) column 1= `time`: ascending (inside of tied groups)
(c) column 2= `status`: descending

```
print "Testing the order function";
time  = myeloma1[,1];
group = myeloma1[,5];
stat  = myeloma1[,2];
indx  = order(group,time,-stat);
print "Index=", indx;
```

```
Index=                    .........
   |    1                56 |   63
---------                 57 |   42
 1 |    5                 58 |   64
 2 |    4                 59 |   43
 3 |   10                 60 |   44
 4 |   14                 61 |   45
 5 |   22                 62 |   65
 6 |   57                 63 |   46
 7 |   31                 64 |   47
 8 |   36                 65 |   48
 9 |   40
10 |    1
.........
```

Applying the index vector to the data gives the ordered data matrix:

```
newdat = myeloma1[indx,];
print "Newdata=",newdat;
```

```
   |       Time    Vstatus     LogBUN        HGB
-------------------------------------------------
 1 |     2.0000    1.00000     1.3010     5.1000
 2 |     2.0000    1.00000     1.7482     11.300
 3 |     6.0000    1.00000     2.1139     10.200
 4 |     7.0000    1.00000     1.0414     5.1000
 5 |     13.000    1.00000    0.77820     5.5000
 6 |     13.000    0.00000     1.6628     4.9000
 7 |     19.000    1.00000     1.2553     7.5000
 8 |     35.000    1.00000     1.1139     7.0000
 9 |     51.000    1.00000     1.5682     7.7000
10 |     1.2500    1.00000     2.2175     9.4000
.................................................
60 |     66.000    1.00000     1.4472     6.6000
```

61	67.000	1.00000	1.3222	12.800
62	77.000	0.00000	1.0792	14.000
63	88.000	1.00000	1.1761	10.600
64	89.000	1.00000	1.3222	14.000
65	92.000	1.00000	1.4314	11.000

	Platelet	Age	LogWBC	Frac
1	0.00000	57.000	3.7243	1.00000
2	0.00000	75.000	3.8062	1.00000
3	0.00000	70.000	3.5441	1.00000
4	0.00000	61.000	3.7324	1.00000
5	0.00000	60.000	3.5798	1.00000
6	0.00000	71.000	3.6435	0.00000
7	0.00000	60.000	3.7924	1.00000
8	0.00000	48.000	3.6532	1.00000
9	0.00000	74.000	3.4150	1.00000
10	1.00000	67.000	3.6628	1.00000
...				
60	1.00000	59.000	3.7853	1.00000
61	1.00000	52.000	3.6435	1.00000
62	1.00000	60.000	3.6812	0.00000
63	1.00000	47.000	3.5563	0.00000
64	1.00000	63.000	3.6532	1.00000
65	1.00000	58.000	4.0755	1.00000

	LogPBM	Protein	SCalc
1	2.0000	3.0000	9.0000
2	1.2553	0.00000	12.000
3	1.3617	1.00000	8.0000
4	2.0000	1.00000	10.0000
5	1.3979	2.0000	10.0000
6	1.7924	0.00000	9.0000
7	1.9294	5.0000	9.0000
8	1.1761	4.0000	10.0000
9	1.0414	4.0000	13.000
10	1.9542	12.000	10.0000
...			
60	1.8195	0.00000	9.0000
61	1.0414	1.00000	10.0000
62	0.95420	0.00000	12.000
63	1.7559	21.000	9.0000
64	1.6232	1.00000	9.0000
65	1.4150	4.0000	11.000

5.9 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 τ, the vector of the coefficients β becomes more sparse (but at the cost of numeric stability). If the input argument `tau` is a vector of $t > 1$ positive values τ_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 τ 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 "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 =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 β

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 τ values, also the specification of a larger probability `acc` makes the coefficient vector β 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 τ_k
2. sum of validation errors in CV
3. number of selected variables for τ_k

Restrictions:

1. Observations with missing values are ignored.
2. For large values of τ 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
```

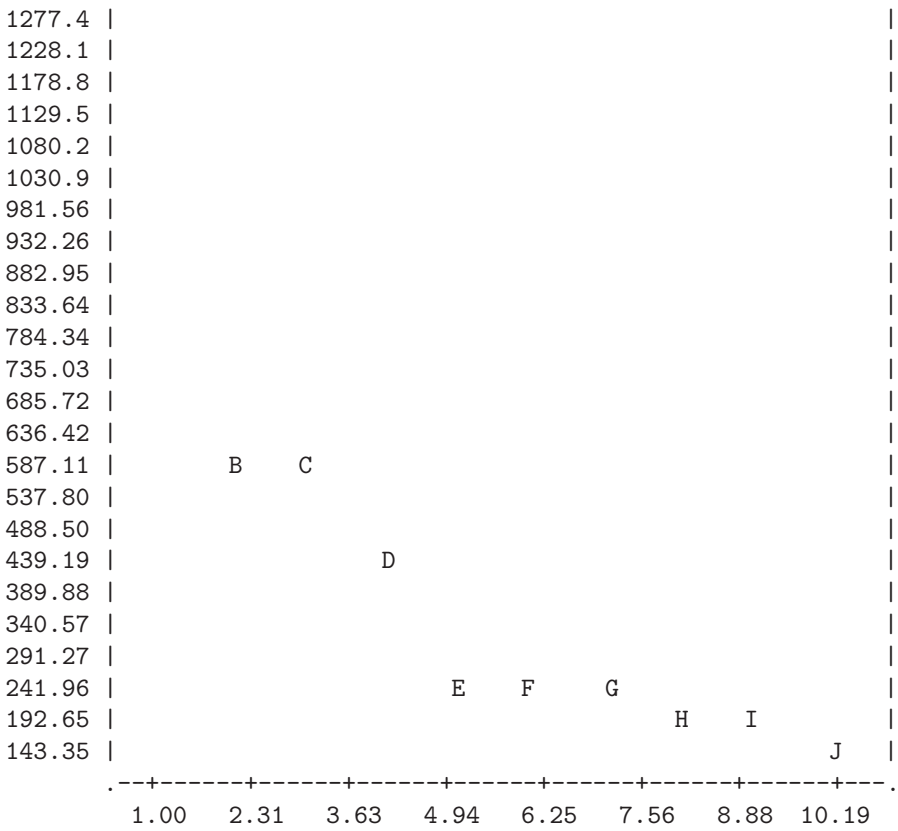
```
Initial Sigma           1.5000
Number Total Iteration   10000
Number Discarded Itera    7000
Rho                     0.5000
Cross Validation         Block
Cross Validation Fold        5
Linear Model Without Intercept
```

```
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
```

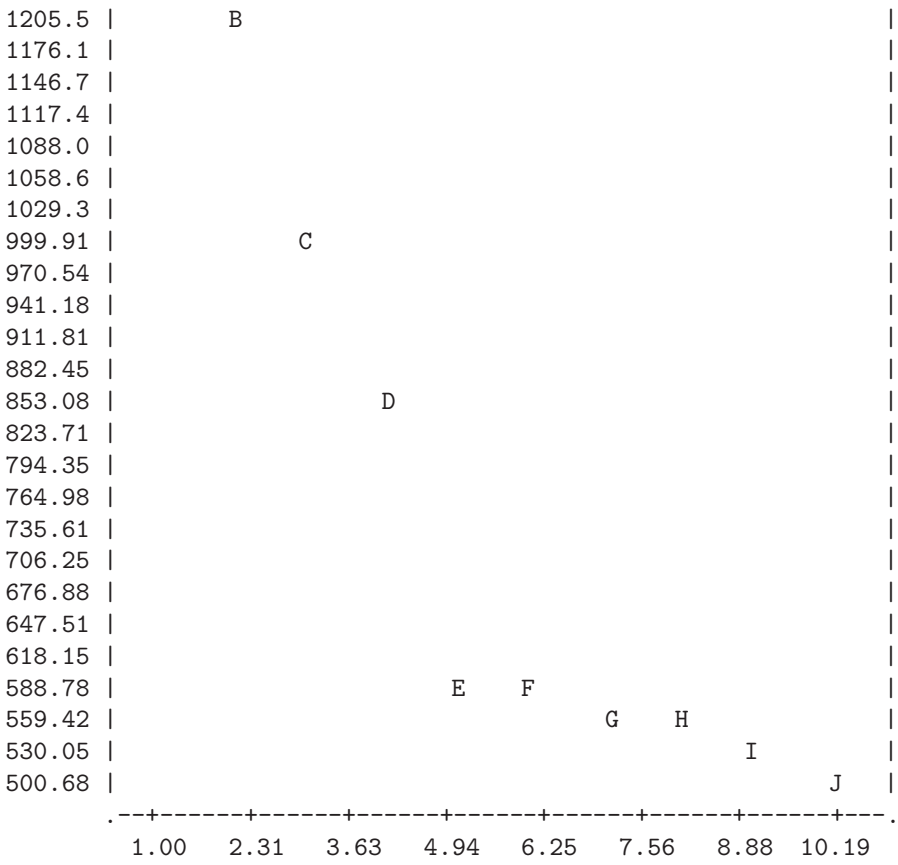
```
     Stochastic Matching Pursuit (Method 1)
          Componentwise Gibbs Sampler

          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
       ********************************************

      .--+------+------+------+------+------+------+------+---.
1969.0 |   A                                                   |
1939.6 |                                                       |
1910.3 |                                                       |
1880.9 |                                                       |
1851.5 |                                                       |
1822.2 |                                                       |
1792.8 |                                                       |
1763.4 |                                                       |
1734.1 |                                                       |
1704.7 |                                                       |
1675.3 |                                                       |
1646.0 |                                                       |
1616.6 |                                                       |
1587.2 |                                                       |
1557.9 |                                                       |
1528.5 |                                                       |
1499.1 |                                                       |
1469.8 |                                                       |
1440.4 |                                                       |
1411.0 |                                                       |
1381.7 |                                                       |
1352.3 |                                                       |
1322.9 |                                                       |
1293.6 |                                                       |
1264.2 |                                                       |
1234.8 |                                                       |
```



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

5.10 Function `survcurv`

< gof,curv,dase > = survcurv(sopt,data,modl,optn<,clas>)

< gof,curv,clu > = survcurv(sopt,data,modl,optn<,clas<data2>>)

Purpose: The `survcurv` function estimates the function values of survival curves for ng groups of data. Following different curve types are implemented:

"adjcox" Cox's proportional hazards model (Zhang et al. 2007): there are two forms,

- one for the stratified PH model
- and one for the simply grouped model,

that means the user may specify either the `"strata"` or the `"group"` column with the options argument. The survival curves are based on the optimal estimates of two different parametrized PH models. They do not use additional data `"data2"`.

"adjaal" Aalen's additive model (Akcin & Zhang, 2010): the groups of observations are defined by a data column which is specified with the `"group"` options argument. There can be no `"strata"` option for Aalen's model. They do not use additional data `"data2"`.

"kapme","kalpr","green","exact" survival curves based on the optimal estimates of Cox's PH model ("kapme": Kaplan-Meier,"kalpr": Kalbfleisch-Prentice, "green": Greenwood); They do not use the `"indiv"` option with additional data `"data2"`.

"aalen","tsiatis","breslow" survival curves based on the optimal estimates of Cox's PH model. Able to use the `"indiv"` option with additional data `"data2"`.

"fleha","efron" survival curves based on the optimal estimates of Cox's PH model ("fleha": Fleming-Harrington). Able to use the `"indiv"`idual option with additional data `"data2"`.

"kapme2",fleha2","fh2" survival curves which are neither using the estimates of Cox' nor Aaalen's model. They do not use additional data `"data2"`.

Preferred are `adjcox` and `adjaal` if the data fit to the Cox PH or Aalen's model respectively. Many of the options of the `survreg()` function, especially those for the optimization technique, are valid for Cox' PH model since the algorithm executes internally a specific form of this function. (For Aalen's model such specifications are ignored.)

Methods `"kapme", "kalpr", "green", "exact", "aalen", "tsiatis", "breslow", "fleha","efron"` are computed by default w.r.t. the mean vector of the data. However, the $n2$ observations of an aditional data set may replace the mean vector for computing $n2$ curves.

- If the `"indiv"` option is specified together with an additional data set `data2` with $n2 > 1$ observations, only a single curve is computed assuming that the rows of `data2` corresponds to different time epochs for only one individual. Note, that methods `"kapme", "kalpr", "green", "exact"` cannot be used together with the `"indiv"` option.
- If the `"indiv"` option is **not** specified together with an additional data set `data2` with $n2 > 1$ observations, $n2$ curves are computed, one for each individual.

"adjcox"	PHC	(*0)
"adjaal"	AAL	-
"kapme"	PHC	(*1)
"kalpr"	PHC	(*1)
"green"	PHC	(*1)
"exact"	PHC	(*1)
"aalen"	PHC	(*2)
"tsiatis"	PHC	(*2)
"breslow"	PHC	(*3)
"fleha"	PHC	(*3)
"efron"	PHC	(*3)
"kapme2"	-	(*4)
"fleha2"	-	(*4)
"fh2"	-	(*4)

Input: sopt This is a string option specifying the type of curve and basic model:

"adjaal" adjusted Aalen's additive model (Akcin & Zhang): based on estimates of Aalen's additive model

"adjcox" adjusted Cox's PH model (Zhang et al): based on estimates of Cox PH model

"aalen" Aaalen: based on Cox PH estimates

"tsiatis" Tsiatis: based on Cox PH estimates

"breslow" Breslow: based on Cox PH estimates

"kapme" Kaplan-Meier: based on Cox PH estimates

"kalpr" Kalbfleisch-Prentice: based on Cox PH estimates

"green" Greenwood: based on Cox PH estimates

"exact" Exact: based on Cox PH estimates

"fleha" Fleming-Harrington: based on Cox PH estimates

"efron" Efron: based on Cox PH estimates

"kapme2" Kaplan-Meier: not based on any model

"fleha2" Fleming-Harrington: not based on any model

"fh2" FH2: not based on any model

data This is a $N \times nc$ data set which contains in its nc columns:

- the dependent time (stop) variable
- the binary censor variable
- a set of n independent covariate variables
- an optional start variable
- an optional strata variable (for groups of observations)
- an optional offset variable
- an optional weight variable

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.)

data2 This is a $n2 \times nc$ data set which is in its columns compatible with `data`. It is valid only for the curve types: `"aalen", "tsiatis", "breslow", "kapme", "kalpr", "green", "exact", "fleha", "efron"`

Option	Second Column	Meaning
"alpha"	real	probability for confidence intervals, default=.05
"absftol"	real	absolute function criterion for terminating see function nlp()
"absgtol"	real	absolute gradient criterion for terminating see function nlp()
"absxtol"	real	absolute x criterion for terminating see function nlp()
"cens"	int	column number of census variable
"cityp"	string	type of CI (for (*1,*2,*3,*4))
	"plain"	plain CI
	"log"	log CI
	"loglog"	loglob CI
"cltyp"	string	type of lower CI (only for (*4))
	"usual"	usual lower CI
	"peto"	Peto lower CI
	"modif"	modified lower CI
"fda"	'f' or 'c'	use finite differences for gradient and Hessian
"fdg"	'f' or 'c'	use finite differences for gradient
"fdh"	'f' or 'c'	use finite differences for Hessian
"ftol"	real	relative function criterion for terminating see function nlp()
"group"	int	column number of group variable
"gtol"	real	relative gradient criterion for terminating see function nlp()
"indiv"		rows in data2 belong to one indiv.
"lsmet"	string	solving LS problem for "adjaal"
	"crt"	using the $\mathbf{X}^T\mathbf{X}$*matrix* (def)
	"svd"	using the svd() method
"maxf"	int	maximum number function calls see function nlp()
"maxi"	int	maximum number iterations see function nlp()
"offset"	int	column number of offset variable
"orig"		use TTs optimization algorithm
"phist"	int	print iteration history (def=0)
"phmet"	string	estimation method
	"bresl"	Breslow's treatment of ties
	"efron"	Efron's treatment of ties
	"exact"	exact treatment of ties

Option	Second Column	Meaning
"pinit"	int	printed output of input (def=0)
"print"	int	amount of printed output (def=0)
"psurv"		print survival curves and ASE's
"robcov"		use robust covariance matrix
"seed"	int	seed for random generator
"sing"	real	singularity criterion, default=10^{-8}
"stat"	int	column number of census variable
"start"	int	column number of start time variable
"strat"	int	column number of strata variable
		(not valid for Aalen's model)
"techop"	string	optimization technique
		see function `nlp()`
"weight"	int	column number of weight variable
"wgt"	int	column number of weight variable
"xtol"	real	relative x criterion for terminating
		see function `nlp()`
"ytyp"	string	type of censoring
	"right"	only stop time is given
	"left"	only stop time is given
	"interval"	start and stop time are given
	"counting"	start and stop time are given (Anderson-Gill)

Output: Assuming there are Nev observations with events (for which the binary census variable is 1) and ng groups of observations (indicated either by the group or strata ID variable).

gof this is vector of some scalar results, see below for content.

curv contains the $Nev \times nc1$ matrix with survival curves and its asymptotic standard errors. The $nc1$ columns are:

1. the time value of the event (x coordinate in the (x, y) plot)
2. the corresponding row number in the input data
3. the rank of the linear system (valid only for Aalen's model)
4. ng columns for the curve function, i.e. y values of the (x, y) plot
5. ng columns for the asymptotic standard errors of the curve values

clu | **dase** **clu** for `"aalen", "tsiatis", "breslow", "kapme", "kalpr", "green", "exact", "fleha", "efron"` and for `"kapme2","fleha2","fh2"`: contains in its first ng columns the lower and its last ng columns the upper confidence limits for the curve.

dase for `"adjcox","adjaal"`: contains the $Nev \times nc2$ matrix with asymptotic standard errors of the $nc2$ pairwise differences of the survival curves. The $nng = n * (ng - 1)/2$ columns correspond to the pairs:

$$(1, 2), ...(1, ng), (2, 3), ..., (2, ng), ..., (ng - 1, ng)$$

The content of `gof` is:

1.

Restrictions: 1. For the adjusted Aalen curves `"adjaal"`, only groups but no strata may be specified.

2. Observations with missing values in significant columns are not processed.

3. For `"adjaal"` at least the data must contain observations for at least two groups.

4. For `"adjcox"` at least the data must contain observations for at least two groups or strata.

Relationships: survreg()

Examples: 1. The following data are being used: The data have unique event times.

```
sarcom = [ 1 0 1  153, 1 0 1  945, 1 0 1 1400, 1 0 1 1887, 1 0 0 2557,
           1 0 0 3134, 1 0 0 3226, 1 0 0 3348, 1 0 0 3501, 1 0 0 3743,
           1 1 1  151, 1 1 1  152, 1 1 1  212, 1 1 1  214, 1 1 1  242,
           1 1 1  243, 1 1 1  244, 1 1 1  245, 1 1 1  249, 1 1 1  273,
           1 1 1  336, 1 1 1  337, 1 1 1  396, 1 1 1  427, 1 1 1  457,
           1 1 1  761, 1 1 1 1249, 1 1 1 1310, 1 1 0 2708, 2 0 1   31,
           2 0 1  335, 2 0 1  366, 2 0 1  426, 2 0 1  456, 2 0 1  578,
           2 0 1  589, 2 0 0  762, 2 0 1  792, 2 0 0  913, 2 0 0  914,
           2 0 1  974, 2 0 1 1005, 2 0 1 1035, 2 0 0 1065, 2 0 1 1096,
           2 0 0 1107, 2 0 0 1219, 2 0 0 1250, 2 0 0 1312, 2 0 0 1403,
           2 0 0 1461, 2 0 0 1553, 2 0 0 1645, 2 0 0 1706, 2 0 0 1734,
           2 0 0 1826, 2 0 0 1948, 2 0 0 1949, 2 0 0 1979, 2 0 0 2222,
           2 0 0 2374, 2 0 0 2435, 2 0 0 2465, 2 0 0 2526, 2 1 1    0,
           2 1 1   91, 2 1 1  183, 2 1 1  334, 2 1 1  338, 2 1 1  365,
           2 1 1  391, 2 1 1  518, 2 1 1  547, 2 1 1  608, 2 1 1  609,
           2 1 1  651 ];
  cnam = [" group ldh event time "];
  sarcom = cname(sarcom,cnam);
```

(a) PHCOX Stratified Model:

```
    modl = "4 = 2";
    optn = [ "status"         3 ,
             "strata"         1 ,
             "pinit"          2 ,
             "psurv"            ,
             "phist"          1 ,
             "print"          3 ];
  < gof,curv,dase > = survcurv("phcox",sarcom,modl,optn);
```

```
print "Gof=",gof;
print "Curv=",curv;
print "DASE=",dase;
```

Survival Curves by Cox PH Model (Stratified: Efron)
Time Data: Right Censored

Number Rows of Training Data.	76
Number Columns of Matrix	4
Number Training Observations.	76
Number Independent Covariates	1
Number (Train) Strata	2
Number (Train) Events	46
Number Unique Time Events	46
Linear Solver Method.	CRT

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

Iteration Start:			
N. Variables	1		
Criterion	147.3410167	Max Grad Entry	16.90946472

Iter	rest	nfun	act	optcrit	difcrit	maxgrad	ridge	rho
1	0	2	0	129.0962	18.24486	1.02554	0	0.92053
2	0	3	0	129.0155	0.080670	0.02970	0	0.98057
3	0	4	0	129.0154	6.5e-005	2e-005	0	1.00047
4	0	5	0	129.0154	3.3e-011	1e-011	0	0.99927

Successful Termination After	4 Iterations		
GCONV convergence criterion satisfied.			
Criterion	129.0154183	Max Grad Entry	1.0710e-011
Ridge (lambda)	0	Act.dF/Pred.dF	0.999269124
N. Function Calls	6	N. Gradient Calls	5
N. Hessian Calls	6	Preproces. Time	0
Time for Method	0	Effective Time	0

Survival Curves and Asymptotic Standard Errors
Stratified PH Model

N	EventTime	Curv1	ASE_1	Curv2	ASE_2

```
 1           0  1.00000000           0  0.97101858  0.02821614
 2 31.00000000  1.00000000           0  0.94179151  0.03896484
 3 91.00000000  1.00000000           0  0.91400357  0.04565551
 4 151.0000000  0.97693348  0.02258343  0.91400357  0.04565551
 5 152.0000000  0.95371485  0.03135041  0.91400357  0.04565551
 6 153.0000000  0.93032833  0.03766660  0.91400357  0.04565551
 7 183.0000000  0.93032833  0.03766660  0.88593458  0.05087544
 8 212.0000000  0.90789026  0.04214453  0.88593458  0.05087544
 9 214.0000000  0.88526463  0.04585838  0.88593458  0.05087544
10 242.0000000  0.86242942  0.04898681  0.88593458  0.05087544
 ..........................................................
40 1005.000000  0.54927861  0.06447870  0.42315698  0.05154254
41 1035.000000  0.54927861  0.06447870  0.40006982  0.05100989
42 1096.000000  0.54927861  0.06447870  0.37748346  0.05089957
43 1249.000000  0.52057597  0.06461041  0.37748346  0.05089957
44 1310.000000  0.48778793  0.06569888  0.37748346  0.05089957
45 1400.000000  0.44699676  0.06849861  0.37748346  0.05089957
46 1887.000000  0.41189082  0.07038976  0.37748346  0.05089957
```

```
   Asymptotic Standard Errors of Pairwise Differences

                          N      DASE21
                          1  0.02821614
                          2  0.03896484
                          3  0.04565551
                          4  0.05097895
                          5  0.05546492
                          6  0.05930617
                          7  0.06343837
                          8  0.06624596
                          9  0.06872003
                         10  0.07089890
                         ..............
                         40  0.07949721
                         41  0.07863768
                         42  0.07810133
                         43  0.07769785
                         44  0.07806150
                         45  0.07995940
                         46  0.08098638
```

(b) PHCOX UNstratified Model:

```
modl = "4 = 2";
optn = [ "status"         3 ,
         "group"          1 ,
```

```
         "pinit"          2 ,
         "psurv"            ,
         "phist"          1 ,
         "print"          3 ];
< gof,curv,dase > = survcurv("phcox",sarcom,modl,optn);
print "Gof=",gof;
print "Curv=",curv;
print "DASE=",dase;
```

```
      Survival Curves by Cox PH Model (Grouped: Efron)
                  Time Data: Right Censored

   Number Rows of Training Data. . . . . .          76
   Number Columns of Matrix  . . . . . .            4
   Number Training Observations. . . . . .          76
   Number Independent Covariates . . . . . .         1
   Number (Train) Strata . . . . . .                 2
   Number (Train) Events . . . . . .                46
   Number Unique Time Events . . . . . .            46
   Linear Solver Method. . . . . .                 CRT

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


 Iteration Start:
 N. Variables             2
 Criterion       179.7007234       Max Grad Entry  19.31170860


Iter rest nfun act   optcrit  difcrit maxgrad   ridge      rho
   1    0    2   0  160.4187 19.28207 4.21314       0  0.78528
   2    0    3   0  159.3512 1.067464 0.40962       0  0.93437
   3    0    4   0  159.3420 9.2e-003  2e-003       0  1.00373
   4    0    5   0  159.3420 2.9e-007  8e-008       0  1.00002

 Successful Termination After     4 Iterations
 GCONV convergence criterion satisfied.
 Criterion       159.3419643       Max Grad Entry  7.6534e-008
 Ridge (lambda)            0       Act.dF/Pred.dF  1.000022327
 N. Function Calls         6       N. Gradient Calls         5
 N. Hessian Calls          6       Preproces. Time           0
 Time for Method           0       Effective Time            0
```

Survival Curves and Asymptotic Standard Errors
Unstratified PH Model

N	EventTime	Curv1	ASE_1	Curv2	ASE_2
1	0	0.98776219	0.01224809	0.98604848	0.01397378
2	31.00000000	0.97543547	0.01737710	0.97204043	0.01983589
3	91.00000000	0.96334316	0.02115831	0.95834305	0.02415444
4	151.0000000	0.95115816	0.02436833	0.94458561	0.02785930
5	152.0000000	0.93892440	0.02720867	0.93081908	0.03110024
6	153.0000000	0.92663923	0.02978878	0.91704163	0.03398658
7	183.0000000	0.91461168	0.03204762	0.90359926	0.03647234
8	212.0000000	0.90247858	0.03413763	0.89008578	0.03881049
9	214.0000000	0.89028732	0.03611129	0.87655552	0.04094344
10	242.0000000	0.87803440	0.03799023	0.86300592	0.04288505
...	...	...	...	...	...
40	1005.000000	0.48429578	0.06137452	0.45546612	0.04974913
41	1035.000000	0.47141875	0.06117477	0.44284229	0.04959597
42	1096.000000	0.45888137	0.06104894	0.43054344	0.04957210
43	1249.000000	0.44637889	0.06102344	0.41826154	0.04976186
44	1310.000000	0.43219509	0.06127904	0.40429555	0.05011418
45	1400.000000	0.41548110	0.06192765	0.38777806	0.05063531
46	1887.000000	0.39472011	0.06328377	0.36714809	0.05270673

Asymptotic Standard Errors of Pairwise Differences

N	DASE21
1	0.00455514
2	0.00870382
3	0.01265317
4	0.01652685
5	0.02029192
6	0.02394657
7	0.02740397
8	0.03078131
9	0.03404661
10	0.03719788
...	...
40	0.06914751
41	0.06850547
42	0.06791152
43	0.06737410
44	0.06686503
45	0.06644190
46	0.06623028

(c) Aalen's Grouped Model:

```
modl = "4 = 2";
optn = [ "status"          3 ,
         "strata"          1 ,
         "lsmet"       "crt" ,
         "pinit"           2 ,
         "psurv"             ,
         "phist"           1 ,
         "print"           3 ];
< gof,curv,dase > = survcurv("aalen",sarcom,modl,optn);
print "Gof=",gof;
print "Curv=",curv;
print "DASE=",dase;
```

```
            Survival Curves by Aalen's Additive Model
                    Time Data: Right Censored

   Number Rows of Training Data. . . . . .          76
   Number Columns of Matrix  . . . . . .            4
   Number Training Observations. . . . . .          76
   Number Independent Covariates . . . . . .         1
   Number (Train) Strata . . . . . .                2
   Number (Train) Events . . . . . .               46
   Number Unique Time Events . . . . . .           46
   Linear Solver Method. . . . . .                CRT

       Survival Curves and Asymptotic Standard Errors

   N   EventTime       Curv1       ASE_1       Curv2       ASE_2
   1           0  1.01224213  0.01259761  0.97206332  0.02727306
   2 31.00000000  1.00808290  0.01302818  0.95358824  0.03254267
   3 91.00000000  1.02155121  0.01958911  0.92556296  0.04150639
   4 151.0000000  0.99338172  0.03385780  0.92204027  0.04119764
   5 152.0000000  0.96492360  0.04327196  0.91850944  0.04087186
   6 153.0000000  0.91954493  0.06035813  0.92489868  0.04226488
   7 183.0000000  0.93393737  0.06347897  0.89629665  0.04885998
   8 212.0000000  0.90550458  0.06782197  0.89245830  0.04832242
   9 214.0000000  0.87673497  0.07164607  0.88860302  0.04776249
  10 242.0000000  0.84760121  0.07502662  0.88473485  0.04717792
  ..........................................................
  40 1005.000000  0.57075469  0.12139222  0.44544705  0.06118247
  41 1035.000000  0.57075469  0.12139222  0.42565273  0.06158201
  42 1096.000000  0.57075469  0.12139222  0.40489341  0.06197823
  43 1249.000000  0.55664085  0.12149518  0.39488104  0.05888633
```

```
44 1310.000000  0.54260346  0.12151844  0.38492292  0.05576851
45 1400.000000  0.48138847  0.12181123  0.38696680  0.05665218
46 1887.000000  0.42018560  0.12014636  0.38963795  0.05789604
```

Asymptotic Standard Errors of Pairwise Differences

N	DASE21
1	0.03987067
2	0.04176061
3	0.05813542
4	0.06242868
5	0.06624499
6	0.08232308
7	0.09267043
8	0.09399858
9	0.09516840
10	0.09618682
...	...
40	0.15544516
41	0.15478913
42	0.15409006
43	0.15125024
44	0.14841946
45	0.14830321
46	0.14692977

2. The following data are being used: The event time data are not unique and have ties.

```
  options NOECHO;
#include "..\\tdata\\myeloma1.dat"
  options ECHO;

  cnam = [" Time Vstatus LogBUN HGB Platelet Age LogWBC Frac
            LogPBM Protein SCalc "];
  myeloma1 = cname(myeloma1,cnam);
```

3. PHCOX Stratified Model:

```
 modl = "1 = 3 4 11";
 optn = [ "status"          2 ,
          "strata"          5 ,
          "phmet"   "breslow" ,
          "pinit"           2 ,
          "psurv"             ,
```

```
          "phist"           1 ,
          "print"           3 ];
< gof,curv,dase > = survcurv("phcox",myeloma1,modl,optn);
print "Gof=",gof;
print "Curv=",curv;
print "DASE=",dase;
```

Survival Curves by Cox PH Model (Stratified: Breslow)
Time Data: Right Censored

Number Rows of Training Data.	65
Number Columns of Matrix	11
Number Training Observations.	65
Number Independent Covariates	3
Number (Train) Strata	2
Number (Train) Events	48
Number Unique Time Events	31
Linear Solver Method.	CRT

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

Iteration Start:			
N. Variables	3		
Criterion	134.3237580	Max Grad Entry	24.47419208

Iter	rest	nfun	act	optcrit	difcrit	maxgrad	ridge	rho
1	0	2	0	129.0019	5.321887	1.72979	0	0.92307
2	0	3	0	128.9474	0.054484	0.01812	0	1.00046
3	0	4	0	128.9474	2.0e-006	3e-006	0	1.00010

Successful Termination After	3 Iterations		
ABSGCONV convergence criterion satisfied.			
Criterion	128.9473858	Max Grad Entry	3.0068e-006
Ridge (lambda)	0	Act.dF/Pred.dF	1.000096850
N. Function Calls	5	N. Gradient Calls	5
N. Hessian Calls	5	Preproces. Time	0
Time for Method	0	Effective Time	0

Survival Curves and Asymptotic Standard Errors

Stratified PH Model

N	EventTime	Curv1	ASE_1	Curv2	ASE_2
1	1.250000000	1.00000000	0	0.96379281	0.02485122
2	2.000000000	0.84786712	0.09543890	0.94355413	0.03115596
3	3.000000000	0.84786712	0.09543890	0.92312631	0.03615444
4	5.000000000	0.84786712	0.09543890	0.87815597	0.04537173
5	6.000000000	0.76819110	0.11200580	0.80968433	0.05473468
6	7.000000000	0.68335806	0.12581690	0.76658034	0.05815070
7	9.000000000	0.68335806	0.12581690	0.74292054	0.06003354
8	11.00000000	0.68335806	0.12581690	0.63190110	0.06433984
9	13.00000000	0.60369364	0.13319847	0.63190110	0.06433984
10	14.00000000	0.60369364	0.13319847	0.60782353	0.06504415
...					
25	54.00000000	0.30290149	0.15500969	0.23767411	0.05680451
26	58.00000000	0.30290149	0.15500969	0.20912368	0.05612389
27	66.00000000	0.30290149	0.15500969	0.18186667	0.05470034
28	67.00000000	0.30290149	0.15500969	0.14937586	0.05293671
29	88.00000000	0.30290149	0.15500969	0.11189865	0.05057417
30	89.00000000	0.30290149	0.15500969	0.07677660	0.04502436
31	92.00000000	0.30290149	0.15500969	0.04603549	0.03648082

Asymptotic Standard Errors of Pairwise Differences

N	DASE21
1	0.02482635
2	0.10054299
3	0.10226045
4	0.10587399
5	0.12497045
6	0.13933318
7	0.14035859
8	0.14182193
9	0.14865680
10	0.14933421
...	
25	0.16067266
26	0.15957475
27	0.15813387
28	0.15781800
29	0.15702726
30	0.15512079
31	0.15190661

4. PHCOX UNStratified Model:

```
 modl = "1 = 3 4 11";
 optn = [ "status"          2 ,
          "group"           5 ,
          "phmet"   "breslow" ,
          "pinit"           2 ,
          "psurv"             ,
          "phist"           1 ,
          "print"           3 ];
< gof,curv,dase > = survcurv("phcox",myeloma1,modl,optn);
print "Gof=",gof;
print "Curv=",curv;
print "DASE=",dase;
```

Survival Curves by Cox PH Model (Grouped: Breslow)
Time Data: Right Censored

Number Rows of Training Data.	65
Number Columns of Matrix	11
Number Training Observations.	65
Number Independent Covariates	3
Number (Train) Strata	2
Number (Train) Events	48
Number Unique Time Events	31
Linear Solver Method.	CRT

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

Iteration Start:
N. Variables 4
Criterion 154.8579914 Max Grad Entry 38.41628820

Iter	rest	nfun	act	optcrit	difcrit	maxgrad	ridge	rho
1	0	2	0	148.4967	6.361260	6.29624	0	0.82235
2	0	3	0	148.0385	0.458232	0.40675	0	1.03639
3	0	4	0	148.0366	1.9e-003	3e-003	0	1.00424
4	0	5	0	148.0366	8.0e-008	1e-007	0	1.00003

Successful Termination After 4 Iterations
GCONV convergence criterion satisfied.
Criterion 148.0365801 Max Grad Entry 1.3163e-007

Ridge (lambda)	0	Act.dF/Pred.dF	1.000029469
N. Function Calls	6	N. Gradient Calls	5
N. Hessian Calls	6	Preproces. Time	0
Time for Method	0	Effective Time	0

Survival Curves and Asymptotic Standard Errors
Unstratified PH Model

N	EventTime	Curv1	ASE_1	Curv2	ASE_2
1	1.250000000	0.96915705	0.02430031	0.97016679	0.02072382
2	2.000000000	0.92028521	0.04372573	0.92279040	0.03316854
3	3.000000000	0.90354985	0.04963598	0.90653746	0.03631797
4	5.000000000	0.86740524	0.06135344	0.87138274	0.04264279
5	6.000000000	0.79535893	0.08073682	0.80109592	0.05176289
6	7.000000000	0.74294447	0.09339887	0.74978151	0.05566873
7	9.000000000	0.72399824	0.09766183	0.73119545	0.05702671
8	11.00000000	0.63420154	0.11359732	0.64283433	0.06108579
9	13.00000000	0.61516172	0.11625366	0.62404115	0.06184589
10	14.00000000	0.59495727	0.11939584	0.60407624	0.06245098
...					
25	54.00000000	0.22366568	0.12907322	0.23275284	0.05666072
26	58.00000000	0.19609288	0.12565262	0.20474637	0.05584722
27	66.00000000	0.16983952	0.12088897	0.17799068	0.05431679
28	67.00000000	0.13863635	0.11199773	0.14605470	0.05243488
29	88.00000000	0.10295821	0.09916756	0.10931000	0.04989635
30	89.00000000	0.06969335	0.08198021	0.07475037	0.04426390
31	92.00000000	0.04075495	0.06156080	0.04433028	0.03557657

Asymptotic Standard Errors of Pairwise Differences

N	DASE21
1	0.01422770
2	0.03522336
3	0.04199786
4	0.05587488
5	0.08050455
6	0.09590843
7	0.10094487
8	0.12096176
9	0.12438772
10	0.12771714
...	...
25	0.12616287
26	0.11993757
27	0.11275064

```
                              28  0.10234372
                              29  0.08726519
                              30  0.06907524
                              31  0.04836435
```

5. Aalen's Model:

```
 modl = "1 = 3 4";
 optn = [ "status"          2 ,
          "group"           5 ,
          "lsmet"       "crt" ,
          "pinit"           2 ,
          "psurv"             ,
          "phist"           1 ,
          "print"           3 ];
< gof,curv,dase > = survcurv("aalen",myeloma1,modl,optn);
print "Gof=",gof;
print "Curv=",curv;
print "DASE=",dase;
```

```
            Survival Curves by Aalen's Additive Model
                    Time Data: Right Censored

    Number Rows of Training Data. . . . . .          65
    Number Columns of Matrix  . . . . . .            11
    Number Training Observations. . . . . .          65
    Number Independent Covariates . . . . . .         2
    Number (Train) Strata . . . . . .                 2
    Number (Train) Events . . . . . .                48
    Number Unique Time Events . . . . . .            31
    Linear Solver Method. . . . . .                 CRT
```

The log output shows that 8 of the 31 unique time events have singular coeffient matrices and the curves are shortened correspondingly:

```
[warning] file tsurvcur2.inp, line 104: From 31 unique time events
          8 have singular linear systems.
```

```
          Survival Curves and Asymptotic Standard Errors

    N   EventTime       Curv1       ASE_1       Curv2       ASE_2
    1 1.250000000  0.99726953  0.01739743  0.96790524  0.02089373
    2 2.000000000  0.78960953  0.13977781  0.95213445  0.02818393
    3 3.000000000  0.82567652  0.15139862  0.92823445  0.03532805
```

4	5.000000000	0.87435348	0.16772624	0.88167394	0.04244341
5	6.000000000	0.76545660	0.20348277	0.82452901	0.04992492
6	7.000000000	0.66336920	0.19783122	0.78451707	0.05251282
7	9.000000000	0.68689128	0.20628907	0.76044865	0.05328398
8	11.00000000	0.68087858	0.21368671	0.67066128	0.06202558
9	13.00000000	0.58015689	0.20441215	0.67420354	0.06524085
10	14.00000000	0.54376016	0.19369126	0.65842089	0.06593166
...					
20	35.00000000	0.21947675	0.15701178	0.41716302	0.07848692
21	37.00000000	0.25310581	0.18859996	0.38253879	0.07375257
22	41.00000000	0.26520032	0.20267762	0.33412381	0.07415039
23	51.00000000	0.09756175	0.12279099	0.33412381	0.07415039
24	52.00000000	0	.	0	.
25	54.00000000	0	.	0	.
26	58.00000000	0	.	0	.
27	66.00000000	0	.	0	.
28	67.00000000	0	.	0	.
29	88.00000000	0	.	0	.
30	89.00000000	0	.	0	.
31	92.00000000	0	.	0	.

Asymptotic Standard Errors of Pairwise Differences

N	DASE21
1	0.02822760
2	0.14771687
3	0.16510944
4	0.18860357
5	0.22630188
6	0.21780285
7	0.22880719
8	0.23781090
9	0.22895554
10	0.21700051
...	
20	0.18662062
21	0.21438860
22	0.23075850
23	0.15175890
24	.
25	.
26	.
27	.
28	.
29	.

30 .
31 .

5.11 Function `survreg`

< gof,parm,cov,res,tres > = survreg(sopt,data,modl,optn<,clas<,test>>)

Purpose: The `survreg` function estimates different methods for survival data which contain one or two time and a binary censor variable.

- Aalen's additive model
- Cox's proportional hazards model
- GLIM-like: extreme distribution
- GLIM-like: logistic distribution
- GLIM-like: Gaussian distribution
- GLIM-like: Weibull distribution (log(extreme))
- GLIM-like: loglogistic distribution (log(logistic))
- GLIM-like: lognormal distribution (log(Gaussian))
- GLIM-like: exponential distribution (this is Weibull distribution for $scale = 1$)
- GLIM-like: Rayleigh distribution (this is Weibull distribution for $scale = .5$)

The implementation is based on research by Terry Therneau (2000) and his wonderful `survival` package in R. Similar models are estimated by the SAS PROCs PHREG and LIFEREG. Note that Aalen's method has an analytical solution and does not need a general optimization technique for the estimation of the parameters.

Input: **sopt** This is a string option specifying the regression method (model) which is applied to the data:

"aalen" Aalen's additive model
"phcox" Cox's proportional hazards model
"extre" extreme distribution
"logis" logistic distribution
"gauss" Gaussian distribution
"weibu" Weibull distribution
"loglo" loglogistic distribution
"logno" lognormal distribution
"expon" exponential distribution (Weibull with fixed scale of 1)
"rayle" exponential distribution (Weibull with fixed scale of .5)

data This is a $N \times nc$ data set which contains in its nc columns:

- the dependent time (stop) variable
- the binary censor variable

- a set of n independent covariate variables
- an optional start variable
- an optional strata variable (for groups of observations)
- an optional offset variable
- an optional weight variable

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 column of the censor 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.)

test An optional test data set with the same column structure can be "scored". Input data, residuals and predicted values are returned in the `tres` output argument. Currently, test set scoring is not implemented with Aalen's method.

Option	Second Column	Meaning
`"alpha"`	real	probability for confidence intervals, default=.05
`"absftol"`	real	absolute function criterion for terminating see function `nlp()`
`"absgtol"`	real	absolute gradient criterion for terminating see function `nlp()`
`"absxtol"`	real	absolute x criterion for terminating see function `nlp()`
`"cens"`	int	column number of census variable
`"estsc"`		estimate scale parameter (valid only for GLIM-like models)
`"fda"`	'f' or 'c'	use finite differences for gradient and Hessian
`"fdg"`	'f' or 'c'	use finite differences for gradient
`"fdh"`	'f' or 'c'	use finite differences for Hessian
`"ftol"`	real	relative function criterion for terminating see function `nlp()`
`"gtol"`	real	relative gradient criterion for terminating see function `nlp()`

Option	Second Column	Meaning
"maxf"	int	maximum number function calls
		see function nlp()
"maxi"	int	maximum number iterations
		see function nlp()
"nowgtr"		do not weight residuals
"nopht"		do not perform proportional hazards test
		(valid only for "phcox")
"offset"	int	column number of offset variable
"orig"		use TTs optimization algorithm
"phist"	int	print iteration history (def=0)
"phmet"	string	estimation method (only for "phcox")
	"bresl"	Breslow's treatment of ties
	"efron"	Efron's treatment of ties
	"exact"	exact treatment of ties
"phtest"	string	transformation for PH test
		(valid only for "phcox")
	"km"	Kaplan-Meier transform
	"lo"	log transform
	"ra"	rank transform
	"id"	identity transform
"pinit"	int	printed output of input (def=0)
"print"	int	amount of printed output (def=0)
"ptyp"	string	type of predicted values (see below)
"robcov"		use robust covariance matrix
"rtyp"	string	type of residuals (see below)
"scal"	real	set scale (fixed or initial value)
		(valid only for GLIM-like models)
"seed"	int	seed for random generator
"sing"	real	singularity criterion, default=10^{-8}
"stat"	int	column number of census variable
"start"	int	column number of start time variable
"strat"	int	column number of strata variable
		(not valid for Aalen's model)
"techop"	string	optimization technique
		see function nlp()
"weight"	int	column number of weight variable
"wgt"	int	column number of weight variable
"xtol"	real	relative x criterion for terminating
		see function nlp()
"ytype"	string	form of census variable
	"right"	right censored
	"left"	right censored
	"right"	right censored
	"count"	count censored

Note, for Weibull and Rayleigh distribution you cannot use the `"estsc"` option or `"scale"` values different from its definititions (1.0 for Weibull and 0.5 for Rayleigh).

Types of Predicted Values: `"ptyp"`

PH Cox Estimation	
"te"	terms of linear predictors (def.)
"lp"	linear predictors
"ri"	risk score
"ex"	expectated number of events
GLIM Estimation	
"re"	response (def.) : [N,1]
"lp"	linear predictors : [N,1]
"te"	terms of linear predictors : [N,n-1]
"qu"	quantile predictors: [N,nq]
"uq"	Uquantile predictors: [N,nq]

Currently, only the (.1, .9) quantiles are implemented. It would need some more input to permit other quantiles. I would do so if requested.

Types of Residuals: `"rtyp"`

PH Cox Estimation	
"mart"	Martingale residuals (def) : [N,1]
"devi"	deviance residuals : [N,1]
"part"	partial residuals : [N,1]
"dfbe"	DFBE residuals : [N,n]
"dfbs"	DFBS residuals : [N,n]
"scor"	score residuals : [N,n]
"scho"	Schoenfeld residuals : [N,n]
"scas"	scaled Schoenfeld residuals : [N,n]
GLIM Estimation	
"resp"	response residuals (def.) : [N,1]
"devi"	deviance residuals : [N,1]
"work"	working residuals : [N,1]
"dfbe"	DFBE residuals : [N,p]
"dfbs"	DFBS residuals : [N,p]
"ldca"	LD case residuals : [N,1]
"ldre"	LD response residuals : [N,1]
"ldsh"	LD shape residuals : [N,1]

1. for `"aalen"` and `"phcox"`:
   - setting `scal` has no impact.
   - `"ytyp"` can only be `"right"` (with only stop time) and `"count"` (with input start and stop time)
2. for GLIM-like models:
   - For `"expon"` and `"rayle"`, the scale must be fixed (1.0 and .5).
   - For stratified GLIM-like models, the scale is not being used.

- For each strata an additional parameter is estimated.
- `"ytyp"` cannot be `"count"`; for `"right"` censoring only stop time, for `"left"` censored data only start time, for interval censored data both start and stop time can be specified.

Output: **gof** this is vector of some scalar results, see below for content.

parm contains the n optimal parameters β_j for each of the independent covariates, together with their risk ratios ($exp(\beta_j)$), z and p values, and asymptotic standard error.

cov contains the $n \times n$ covariance matrix of parameter estimates.

res contains the input data, predicted values, and residuals. Currently for Aalen's method predicted values and residuals are not computed and returned.

tres contains predicted values and residuals of a specified test data set. Currently, test set scoring is not implemented with Aalen's method which means that this return is missing.

The content of `gof` is:

1.

Restrictions: 1. Currently there are no residuals or predicted values for Aalen's method.

2. For GLIM estimation you can have stratified models only together with scale estimation. For k strata there are k additional weight parameters to estimate. A specified `"scale"` value is used as starting estimate for the weights.

Relationships: survprd() survfor(), survcurv() reg(), glim()

Examples: 1. PH Cox Estimation: Ovarian Cancer Data: UNStratified Example: nvar=4:

```
print "Ovarian Data: UNStratified Example: nvar=4";
ovar = [ 1      59      1 72.3315        2  1       1,
         2     115      1 74.4932        2  1       1,
         3     156      1 66.4658        2  1       2,
         ..........................................
         24    353      1 63.2192        1  2       2,
         25    365      1 64.4247        2  2       1,
         26    377      0 58.3096        1  2       1 ];

   cnam = [" N futime fustat age resid_ds rx ecog_ps "];
   ovar = cname(ovar,cnam);
   nr = nrow(ovar); nc = ncol(ovar);
   print "Interval Data: nr=",nr," nc=",nc;
   print ovar[1:10,];
```

```
 modl = "2 = 4:7";
 optn = [ "status"       3 ,
          "phmet"  "efron" ,
          "rtyp"    "devi" ,
          "pinit"        2 ,
          "phist"        1 ,
          "print"        3 ];
< gof,parm,covm,resi > = survreg("phcox",ovar,modl,optn);
print "Gof=",gof;
print "Parm=",parm;
print "Covm=",covm;
print "Resi=",resi;
```

Cox Proportional Hazards Regression (Method by Efron)
Time Data: Right Censored

Number Rows of Training Data.	26
Number Columns of Matrix	7
Number Training Observations.	26
Number Independent Covariates	4
Number (Train) Events	12
Seed for Random Generator	234

****
Simple Statistics
****

Column	Nobs	Mean	Std Dev	Skewness	Kurtosis
Y[2]	26	599.53846	339.68847	0.4394487	-0.7630457
X[4]	26	56.165442	10.100361	0.0461065	-0.5306959
X[5]	26	1.5769231	0.5038315	-0.3307984	-2.0553360
X[6]	26	1.5000000	0.5099020	0	-2.1739130
X[7]	26	1.4615385	0.5083911	0.1639158	-2.1447981

Specific Training Data

Nrow	Nobs	Status	Time
1	1	1	59.000000
2	2	1	115.00000
3	3	1	156.00000
.....			
24	24	1	353.00000
25	25	1	365.00000
26	26	0	377.00000

Training Covariates

	age	resid_ds	rx	ecog_ps
OBS_01	72.33150000	2.000000000	1.000000000	1.000000000
OBS_02	74.49320000	2.000000000	1.000000000	1.000000000
OBS_03	66.46580000	2.000000000	1.000000000	2.000000000
.....				
OBS_24	63.21920000	1.000000000	2.000000000	2.000000000
OBS_25	64.42470000	2.000000000	2.000000000	1.000000000
OBS_26	58.30960000	1.000000000	2.000000000	1.000000000

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

Iteration Start:
N. Variables 4
Criterion 34.98494037 Max Grad Entry 100.2389003

Iter	rest	nfun	act	optcrit	difcrit	maxgrad	ridge	rho
1	0	2	0	27.84642	7.138518	16.5039	0	0.68597
2	0	3	0	26.49962	1.346804	2.36532	0	0.87613
3	0	4	0	26.46330	0.036322	0.01929	0	1.00377
4	0	5	0	26.46329	2.6e-006	7e-006	0	1.00006

Successful Termination After 4 Iterations
ABSGCONV convergence criterion satisfied.
Criterion 26.46329352 Max Grad Entry 7.2365e-006
Ridge (lambda) 0 Act.dF/Pred.dF 1.000064104
N. Function Calls 6 N. Gradient Calls 5
N. Hessian Calls 6 Preproces. Time 0
Time for Method 0 Effective Time 0

Test Statistics for Training Data

-2 Log L: Without: 69.9699 With Covariates: 52.9266
R_squared: 0.818106 Max(R_squared): 0.999085
Likelihood Ratio Test: 17.0433 df=4 Prob=0.00189587
Score (logrank) Test: 20.8128 df=4 Prob=0.000344912
Wald Test: 14.2501 df=4 Prob=0.00653826

AIC=60.9266 SBC=62.1369

Global PH Test (Kaplan-Meier): ChiSqu=3.36086 Prob=0.499348

Coefficients, ASEs, Z, and P Values

Variable	Coef	AsyStdErr	Z_Value	P_Value
age	0.12481307	0.04689036	2.66180652	0.00777225
resid_ds	0.82618635	0.78961103	1.04632069	0.29541300
rx	-0.91449982	0.65331561	-1.39978259	0.16157843
ecog_ps	0.33621161	0.64392305	0.52213011	0.60157974

Risk (exp(coef)) and Confidence Intervals

Var	Coef	Exp(Coef)	CI_Lower	CI_Upper
1	0.1248131	1.1329367	1.0334572	1.2419919
2	0.8261864	2.2845895	0.4860602	10.738072
3	-0.9144998	0.4007170	0.1113614	1.4419196
4	0.3362116	1.3996352	0.3961928	4.9445087

Proportional Hazards Test (Kaplan-Meier Transform)

Variable	Rho	ChiSqu	P_Value
age	-0.03988567	0.02617998	0.87146154
resid_ds	-0.14168273	0.24625825	0.61972217
rx	0.13246201	0.20011980	0.65462416
ecog_ps	0.48447739	1.88192366	0.17011567

Covariance Matrix of Parameter Estimates

****

Dense Symmetric Matrix (4 by 4)

S	age	resid_ds	rx	ecog_ps
age	0.0021987			
resid_ds	-0.0132674	0.6234856		
rx	0.0051495	-0.0939057	0.4268213	
ecog_ps	-0.0038605	0.1554163	-0.0862510	0.4146369

Training Data, Predicted Values, and Residuals

****

```
                    Dense Matrix (26 by 8)

       |       Event      Start       Time       LinP  P_LPterms
     ------------------------------------------------------------
OBS_01 |   1.0000000          0  59.000000  2.6693509  2.6693509
OBS_02 |   1.0000000          0  115.00000  2.9391594  2.9391594
OBS_03 |   1.0000000          0  156.00000  2.2734466  2.2734466
.............................................................
OBS_24 |   1.0000000          0  353.00000  0.1275423  0.1275423
OBS_25 |   1.0000000          0  365.00000  0.7679792  0.7679792
OBS_26 |          0           0  377.00000 -0.8214516 -0.8214516


       |       P_ASE   R_Martin R_Deviance
     -----------------------------------
OBS_01 |   0.7831646  0.8410328  1.4128160
OBS_02 |   0.8657932  0.5442439  0.6950591
OBS_03 |   0.7192268  0.5967083  0.7891600
.....................................
OBS_24 |   0.7549817  0.8211167  1.3415700
OBS_25 |   0.6125701  0.5513655  0.7073631
OBS_26 |   0.8005724 -0.0915402 -0.4278789


                Total Computation Time: 0
```

The score residuals:

```
       |       P_ASE   R_Martin  R_Score_1  R_Score_2  R_Score_3
     ------------------------------------------------------------
OBS_01 |   0.7831646  0.8410328  2.2650331  0.0568636 -0.1056538
OBS_02 |   0.8657932  0.5442439  3.0252549  0.0464131 -0.0862366
OBS_03 |   0.7192268  0.5967083 -0.0685129  0.0713143 -0.1325036
.............................................................
OBS_24 |   0.7322996  0.8211167  3.9608430 -0.5656692  0.3464895
OBS_25 |   0.7684894  0.5513655  4.3002574  0.1524126  0.2241752
OBS_26 |   0.6536023 -0.0915402  0.3149064  0.0709177 -0.0521220


       |   R_Score_4
     ---------------
OBS_01 |  -0.4266169
OBS_02 |  -0.3482128
OBS_03 |   0.0616753
...................
OBS_24 |   0.4490741
OBS_25 |  -0.2039044
```

```
OBS_26 |   0.0484562
```

Schoenfeld residuals are nonmissing only for events:

```
       |       P_ASE   R_Martin  R_Schoe_1  R_Schoe_2  R_Schoe_3
     ------------------------------------------------------------
OBS_01 |   0.7831646  0.8410328  2.6931568  0.0676116 -0.1256239
OBS_02 |   0.8657932  0.5442439  5.3639019  0.0803912 -0.1493686
OBS_03 |   0.7192268  0.5967083 -0.8987740  0.1068399 -0.1985109
.....................................................................
OBS_24 |   0.7322996  0.8211167  4.0610442 -0.7058765  0.4535120
OBS_25 |   0.7684894  0.5513655  5.5003587  0.2534827  0.4796229
OBS_26 |   0.6536023 -0.0915402     .          .          .
```

```
       |   R_Schoe_4
     ---------------
OBS_01 |  -0.5072536
OBS_02 |  -0.6031317
OBS_03 |   0.1984379
...................
OBS_24 |   0.5282024
OBS_25 |  -0.4413864
OBS_26 |      .
```

2. PH Cox Estimation: Ovarian Data: Stratified Example: nvar=2:

```
 modl = "2 = 4 7";
 optn = [ "strat"        6 ,
          "status"       3 ,
          "phmet"  "efron" ,
          "pinit"        2 ,
          "phist"        1 ,
          "print"        3 ];
< gof,parm,covm,resi > = survreg("phcox",ovar,modl,optn);
print "Gof=",gof;
print "Parm=",parm;
print "Covm=",covm;
print "Resi=",resi;
```

```
     Cox Proportional Hazards Regression (Method by Efron)
                  Time Data: Right Censored

    Number Rows of Training Data. . . . . .          26
    Number Columns of Matrix  . . . . . .             7
```

```
Number Training Observations. . . . . .          26
Number Independent Covariates . . . . . .         2
Number (Train) Strata . . . . . .                 2
Number (Train) Events . . . . . .                12
Seed for Random Generator . . . . . .           234
```

Specific Training Data

Nrow	Nobs	Strata	Status	Time
1	1	0	1	59.000000
2	2	0	1	115.00000
3	3	0	1	156.00000
...	...	...	...	...
24	24	1	1	353.00000
25	25	1	1	365.00000
26	26	1	0	377.00000

Strata Frequencies for Training Data

Strata	Freqency	CumFreq	Nevents
0	13	13	7
1	13	26	5

Training Covariates

	age	ecog_ps
OBS_01	72.33150000	1.000000000
OBS_02	74.49320000	1.000000000
OBS_03	66.46580000	2.000000000
...	...	...
OBS_24	63.21920000	2.000000000
OBS_25	64.42470000	1.000000000
OBS_26	58.30960000	1.000000000

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

```
Iteration Start:
N. Variables             2
Criterion       26.90887268        Max Grad Entry  104.6791318
```

```
Iter rest nfun act   optcrit  difcrit maxgrad   ridge      rho
   1    0    2   0  20.65629 6.252583 10.2205       0  1.02140
   2    0    3   0  20.55354 0.102745 0.87744       0  1.05724
   3    0    4   0  20.55269 8.5e-004  8e-003       0  1.00641
   4    0    5   0  20.55269 8.0e-008  8e-007       0  1.00006

 Successful Termination After     4 Iterations
 GCONV convergence criterion satisfied.
 Criterion       20.55268978        Max Grad Entry  7.9862e-007
 Ridge (lambda)            0        Act.dF/Pred.dF  1.000063240
 N. Function Calls         6        N. Gradient Calls         5
 N. Hessian Calls          6        Preproces. Time           0
 Time for Method           0        Effective Time            0

            Test Statistics for Training Data

    -2 Log L: Without: 53.8177 With Covariates: 41.1054
        R_squared: 0.30456 Max(R_squared): 0.785114
    Likelihood Ratio Test: 12.7124 df=2 Prob=0.00173598
    Score (logrank) Test: 12.2431 df=2 Prob=0.00219503
          Wald Test: 8.43157 df=2 Prob=0.0147608
                  AIC=45.1054 SBC=48.2161

 Global PH Test (Kaplan-Meier): ChiSqu=1.88404 Prob=0.38984


            Coefficients, ASEs, Z, and P Values

    Variable        Coef   AsyStdErr     Z_Value     P_Value
         age  0.13852785  0.04801415  2.88514644  0.00391232
     ecog_ps -0.09669743  0.62994270 -0.15350194  0.87800246


         Risk (exp(coef)) and Confidence Intervals

        Var       Coef  Exp(Coef)   CI_Lower   CI_Upper
          1  0.1385278  1.1485817  1.0454233  1.2619193
          2 -0.0966974  0.9078306  0.2641172  3.1204199


    Proportional Hazards Test (Kaplan-Meier Transform)

          Variable         Rho      ChiSqu     P_Value
               age -0.02034256  0.00466864  0.94552497
```

```
ecog_ps  0.51701863  1.85869534  0.17277545


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

Dense Symmetric Matrix (2 by 2)

          S |         age    ecog_ps
          --------------------------
        age |   0.0023054
    ecog_ps |  -0.0049991  0.3968278


Training Data, Predicted Values, and Residuals
**********************************************

Dense Matrix (26 by 8)

       |       Event     Strata      Start       Time       LinP
     ------------------------------------------------------------
OBS_01 |   1.0000000          0          0  59.000000  2.2840788
OBS_02 |   1.0000000          0          0  115.00000  2.5835344
OBS_03 |   1.0000000          0          0  156.00000  1.3748185
.............................................................
OBS_24 |   1.0000000  1.0000000          0  353.00000  0.9250740
OBS_25 |   1.0000000  1.0000000          0  365.00000  1.1887668
OBS_26 |          0  1.0000000          0  377.00000  0.3416551

       |   P_LPterms      P_ASE   R_Martin
     ------------------------------------
OBS_01 |   2.2840788  0.8727063  0.7839421
OBS_02 |   2.5835344  0.9713348  0.3366863
OBS_03 |   1.3748185  0.5515392  0.6252141
.....................................
OBS_24 |  -0.2682553  0.2930349  0.8520456
OBS_25 |  -1.6121555  0.5992499  0.5813637
OBS_26 |   0.4221902  0.3516331 -0.1794490


                Total Computation Time: 0
```

3. Aalen's Model: Lung Cancer Data: N=228 with one observation missing:

```
print "Lung Cancer Data Example";
```

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


  nr = nrow(lung);  nc = ncol(lung);
  print "LUNG: nr=",nrow(lung)," nc=",ncol(lung);
  lung[,4] = (lung[,4] .== 2.);


  cnam = [" N inst time status age sex ph_ecog ph_karno pat_karno
            meal_cal wt_loss "];
  lung = cname(lung,cnam);
  print "LUNG=",lung[1:10,];


  print "fit = aareg(Surv(time, status) ~ age + sex + ph.ecog,";
  print " data=lung, nmin=1, taper=1:10 )";
   modl = "3 = 5 6 7";
   optn = [ "status"       4 ,
            "nmin"         1 ,
            "pinit"        2 ,
            "phist"        1 ,
            "print"        3 ];
   < gof,parm,covm,resi > = survreg("aalen",lung,modl,optn);
   print "Gof=",gof;
   print "Parm=",parm;
   print "Covm=",covm;
```

Survival Regression by Aalen's Additive Model
Test Type: Aalen
Time Data: Right Censored

Number Rows of Training Data.	228
Number Columns of Matrix	11
Rows of Train Data with Missing Values. . .	1
Number Training Observations.	227
Number Independent Covariates	3
Number (Train) Events	164

ChiSquared=26.7609 DF=3 Prob= 0.0000

Coefficients, ASEs, Z, and P Values

Variable	Slope	Coef	AsyStdErr	Z_Value	P_Value

```
Intercept  0.0052477  0.0059854  4.8e-003  1.259483  0.2079
      age  4.29e-005  7.03e-005  7.2e-005  0.969643  0.3322
      sex -0.0033033 -0.0040289  1.2e-003 -3.299540  0.0010
  ph_ecog  0.0031461  0.0038107  1.0e-003  3.704676  0.0002


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

              Dense Symmetric Matrix (4 by 4)

          S |   Intercept        age        sex    ph_ecog
          ---------------------------------------------------
  Intercept |   2.3720996
        age |  -152.70046  12023.778
        sex |  -3.7440302  17.854357  35.049321
    ph_ecog |   2.2576720 -283.50098 -5.4212658  86.187939


                  Total Computation Time: 0
```

4. Unstratified Weibull Estimation:
   Like exponential distribution with Fixed Scale=1:

```
print "Ovarian Data: UNStratified Example: nvar=4";
ovar = [ 1      59      1 72.3315        2  1       1,
         2     115      1 74.4932        2  1       1,
         3     156      1 66.4658        2  1       2,
         4     421      0 53.3644        2  2       1,
         5     431      1 50.3397        2  1       1,
         .........................................
        21    1227      0 59.5890        1  2       2,
        22     268      1 74.5041        2  1       2,
        23     329      1 43.1370        2  1       1,
        24     353      1 63.2192        1  2       2,
        25     365      1 64.4247        2  2       1,
        26     377      0 58.3096        1  2       1 ];
  cnam = [" N futime fustat age resid_ds rx ecog_ps "];
  ovar = cname(ovar,cnam);
  nr = nrow(ovar); nc = ncol(ovar);
  print "Interval Data: nr=",nr," nc=",nc;
  print ovar[1:10,];


  print "Exponential Model: Weibull with fixed scale = 1: NRR";
  print "rtyp=working, ptyp=quant";
  modl = "2 = 6 7";
```

```
optn = [ "status"         3 ,
         "scale"          1.,
         "rtyp"       "work" ,
         "ptyp"      "quant" ,
         "pinit"          2 ,
         "phist"          1 ,
         "print"          3 ];
< gof,parm,covm,resi > = survreg("weibu",ovar,modl,optn);
print "Gof=",gof;
print "Parm=",parm;
print "Covm=",covm;
print "Resi=",resi;
```

Survival Regression: Weibull Distribution
Time Data: Right Censored

Number Rows of Training Data.	26
Number Columns of Matrix	7
Number Training Observations.	26
Number Independent Covariates	3
Number (Train) Events	12
Column for Censor Variable.	3
Type of Predictions	Quant
Type of Residuals	Working
Fixed Scale at.	1.00000000

Estimating the Intercept

Iter	LogLik	LL_Change	TestCrit	Gmax
0	-31.1965753	.	.	6.62841110
1	-29.8890423	1.30753300	0.04374623	1.05100598
2	-29.8456097	0.04343258	0.00145524	0.04120561
3	-29.8455391	7.058e-005	2.365e-006	7.042e-005
4	-29.8455391	2.066e-010	6.924e-012	2.066e-010

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

Iteration Start:
N. Variables 3

```
 Criterion       80.21900607        Max Grad Entry  108.6753985

Iter rest nfun act   optcrit  difcrit maxgrad   ridge      rho
   1    0    2   0  40.57078 39.64822 35.9737       0  1.23528
   2    0    3   0  30.57028 10.00050 9.95337       0  1.19523
   3    0    4   0  29.08604 1.484245 1.73897       0  1.12554
   4    0    5   0  29.01177 0.074270 0.10488       0  1.04550
   5    0    6   0  29.01138 3.9e-004  6e-004       0  1.00408
   6    0    7   0  29.01138 1.5e-008  2e-008       0  1.00003

 Successful Termination After     6 Iterations
 GCONV convergence criterion satisfied.
 Criterion       29.01137820        Max Grad Entry  2.2917e-008
 Ridge (lambda)            0        Act.dF/Pred.dF  1.000025733
 N. Function Calls         8        N. Gradient Calls         5
 N. Hessian Calls          8        Preproces. Time           0
 Time for Method           0        Effective Time            0


      Loglik: Intercept only = -98.0322 Model = -97.198
          Chisquare=1.66832 with df=2 Prob=0.434239


            Coefficients, ASEs, Z, and P Values

     Variable        Coef   AsyStdErr     Z_Value     P_Value
    Intercept  6.96183761  1.32187737  5.26662894  1.390e-007
           rx  0.58150273  0.58699362  0.99064573  0.32185860
      ecog_ps -0.43313465  0.58699362 -0.73788647  0.46058344
                  Fixed Scale Value = 1


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

               Dense Symmetric Matrix (3 by 3)

                 S |   Intercept         rx    ecog_ps
                   -------------------------------------
         Intercept |   1.7473598
                rx |  -0.5264985  0.3445615
           ecog_ps |  -0.5798865  0.0242335  0.3445615
```

The columns PQU_1 and PQU_2 show the quantile predicted values with its asymptotic statndard errors in clolumns PQASE_1 and PQASE_2:

```
            Training Data, Predicted Values, and Residuals
```

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

                Dense Matrix (26 by 8)

       |       Event      Start       Time      PQU_1      PQU_2
-------------------------------------------------------------------
OBS_01 |   1.0000000          0  59.000000  129.00335  2819.2837
OBS_02 |   1.0000000          0  115.00000  129.00335  2819.2837
OBS_03 |   1.0000000          0  156.00000  83.655210  1828.2298
OBS_04 |          0           0  421.00000  230.75143  5042.9215
OBS_05 |   1.0000000          0  431.00000  129.00335  2819.2837
...................................................................
OBS_21 |          0           0  1227.0000  149.63611  3270.1993
OBS_22 |   1.0000000          0  268.00000  83.655210  1828.2298
OBS_23 |   1.0000000          0  329.00000  129.00335  2819.2837
OBS_24 |   1.0000000          0  353.00000  149.63611  3270.1993
OBS_25 |   1.0000000          0  365.00000  230.75143  5042.9215
OBS_26 |          0           0  377.00000  230.75143  5042.9215


       |     PQASE_1    PQASE_2   Residual
-------------------------------------------
OBS_01 |   67.302152  1470.8445 -19.752532
OBS_02 |   67.302152  1470.8445 -9.6469509
OBS_03 |   36.899283  806.40967 -4.0896803
OBS_04 |   126.66853  2768.2578  1.0000000
OBS_05 |   67.302152  1470.8445 -1.8408338
.......................................
OBS_21 |   78.066442  1706.0910  1.0000000
OBS_22 |   36.899283  806.40967 -1.9626497
OBS_23 |   67.302152  1470.8445 -2.7215786
OBS_24 |   78.066442  1706.0910 -3.0233130
OBS_25 |   126.66853  2768.2578 -5.0003095
OBS_26 |   126.66853  2768.2578  1.0000000


              Total Computation Time: 0
```

5. Unstratified Weibull Estimation: With Scale Estimation

```
print "Exponential Model: Weibull with scale Estimation: NRR";
modl = "2 = 6 7";
optn = [ "status"         3 ,
         "estsc"            ,
         "rtyp"      "resp" ,
```

```
        "ptyp"      "resp" ,
        "pinit"          2 ,
        "phist"          1 ,
        "print"          3 ];
< gof,parm,covm,resi > = survreg("weibu",ovar,modl,optn);
print "Gof=",gof;
print "Parm=",parm;
print "Covm=",covm;
print "Resi=",resi;
```

Estimating the Intercept

Iter	LogLik	LL_Change	TestCrit	Gmax
0	-31.2185467	.	.	5.24078495
1	-30.3525589	0.86598784	0.02853097	5.06898497
2	-29.7782941	0.57426472	0.01928467	0.63828278
3	-29.7672466	0.01104750	3.711e-004	0.01496534
4	-29.7672400	6.680e-006	2.244e-007	9.713e-006
5	-29.7672400	2.689e-012	9.037e-014	3.839e-012

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

```
 Iteration Start:
 N. Variables              4
 Criterion       92.31459082        Max Grad Entry  142.0760369
```

Iter	rest	nfun	act	optcrit	difcrit	maxgrad	ridge	rho
1	0	2	0	38.61533	53.69926	30.9045	128.000	1.42308
2	0	3	0	33.41613	5.199201	10.8726	32.0000	1.31935
3	0	4	0	31.05608	2.360044	4.28609	8.00000	1.33893
4	0	5	0	29.87365	1.182432	1.75689	2.00000	1.42601
5	0	6	0	29.09014	0.783508	0.77031	0.50000	1.40867
6	0	7	0	28.90946	0.180677	0.22210	0.12500	1.18986
7	0	8	0	28.89793	0.011530	0.02135	0.03125	1.07228
8	0	9	0	28.89783	1.0e-004	2e-004	0	1.00397
9	0	10	0	28.89783	4.0e-009	1e-008	0	1.00003

```
 Successful Termination After     9 Iterations
 GCONV convergence criterion satisfied.
 Criterion       28.89783209        Max Grad Entry  1.0358e-008
 Ridge (lambda)            0        Act.dF/Pred.dF  1.000026854
```

```
N. Function Calls        11        N. Gradient Calls         7
N. Hessian Calls         11        Preproces. Time           0
Time for Method           0        Effective Time            0


     Loglik: Intercept only = -97.9539 Model = -97.0845
          Chisquare=1.73882 with df=2 Prob=0.4192

            Coefficients, ASEs, Z, and P Values

    Variable        Coef   AsyStdErr     Z_Value     P_Value
   Intercept  6.89669313  1.17754721  5.85682938  4.718e-009
          rx  0.52864550  0.52920190  0.99894860  0.31781959
      ecog_ps -0.38504255  0.52700178 -0.73062855  0.46500607
     STRAT_1 -0.12344181  0.25221310 -0.48943457  0.62453407

          Estimated Scale (exp(coef)) = 0.883873


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

              Dense Symmetric Matrix (4 by 4)

          S |   Intercept         rx    ecog_ps    STRAT_1
          ------------------------------------------------
  Intercept |   1.3866174
         rx |  -0.4020102  0.2800547
     ecog_ps |  -0.4662070  0.0110496  0.2777309
     STRAT_1 |   0.0294539  0.0260271 -0.0230128  0.0636114
```

6. Unstratified Rayleigh Estimation: Weibull With Fixed Scale=.5

```
print "Rayleigh Model: Weibull with fixed scale = .5";
modl = "2 = 6 7";
optn = [ "status"         3 ,
         "rtyp"      "resp" ,
         "ptyp"      "resp" ,
         "pinit"          2 ,
         "phist"          1 ,
         "print"          3 ];
< gof,parm,covm,resi > = survreg("rayleigh",ovar,modl,optn);


     Loglik: Intercept only = -101.51 Model = -100.392
          Chisquare=2.23644 with df=2 Prob=0.32686
```

```
                Coefficients, ASEs, Z, and P Values

      Variable        Coef   AsyStdErr     Z_Value     P_Value
     Intercept  6.75844737  0.67182649  10.0598108  8.316e-024
            rx  0.34144404  0.29429937  1.16019291  0.24597027
       ecog_ps -0.23462632  0.29429937 -0.79723693  0.42531344

                    Fixed Scale Value = 0.5
```

7. Stratified Weibull Estimation: Lung Cancer Data

```
  print "Lung Data Example";
  options NOECHO;
#include "..\\tdata\\lung.dat"
  options ECHO;

  lung[,4] = (lung[,4] .== 2);
  cnam = [" N inst time status age sex ph_ecog ph_karno pat_karno meal_cal wt_loss
  lung = cname(lung,cnam);
  print "Lung=",lung[1:10,];
```

The data have a missing value for observation 14. The current form of the `survival` package in R (T. Therneau) does have some problems with the computation of residuals and predicted values when there are missing values in the data.

```
 modl = "3 = 5 7";
 optn = [ "status"         4 ,
          "strata"         6 ,
          "rtyp"      "resp" ,
          "ptyp"      "resp" ,
          "pinit"          2 ,
          "phist"          1 ,
          "print"          3 ];
 < gof,parm,covm,resi > = survreg("weibu",lung,modl,optn);
 print "Gof=",gof;
 print "Parm=",parm;
 print "Covm=",covm;
 print "Resi=",resi;
```

```
          Survival Regression: Weibull Distribution
                  Time Data: Right Censored
```

```
Number Rows of Training Data. . . . . .          228
Number Columns of Matrix  . . . . . .           11
Rows of Train Data with Missing Values. . .      1
Number Training Observations. . . . . .         227
Number Independent Covariates . . . . . .         3
Number (Train) Strata . . . . . .                 2
Number (Train) Events . . . . . .               164
Column for Censor Variable. . . . . .             4
Column for Strata Variable. . . . . .             6
Type of Predictions . . . . . .          Response
Type of Residuals . . . . . .            Response
Initial value of Scale. . . . . .               .
```

Estimating the Intercept

Iter	LogLik	LL_Change	TestCrit	Gmax
0	-318.263675	.	.	65.6815565
1*	-279.491038	38.7726369	0.25888500	31.3907284
2	-276.268512	3.22252663	0.01166447	3.26959527
3	-276.185260	0.08325106	3.014e-004	0.16840113
4	-276.185159	1.010e-004	3.657e-007	3.147e-004
5	-276.185159	2.836e-010	1.027e-012	9.012e-010

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

```
Iteration Start:
N. Variables              5
Criterion       1657.198755        Max Grad Entry  175887.0408
```

Iter	rest	nfun	act	optcrit	difcrit	maxgrad	ridge	rho
1	0	2	0	509.0380	1148.161	48384.7	8192.00	1.25904
2	0	3	0	314.4829	194.5551	12186.5	2048.00	1.18647
3	0	4	0	285.7410	28.74195	2829.38	512.000	1.24055
4	0	5	0	274.2397	11.50133	767.965	128.000	1.37686
5	0	6	0	270.0939	4.145733	192.791	32.0000	1.31853
6	0	7	0	268.1693	1.924584	53.2828	8.00000	1.48234
7	0	8	0	267.3609	0.808460	38.1587	2.00000	1.26159
8	0	9	0	267.2873	0.073568	4.96731	0.50000	1.09538
9	0	10	0	267.2865	8.6e-004	0.06786	0.12500	1.02364
10	0	11	0	267.2865	5.4e-007	4e-005	0.03125	1.00559

```
Successful Termination After    10 Iterations
GCONV convergence criterion satisfied.
Criterion       267.2864563        Max Grad Entry  4.2858e-005
Ridge (lambda)            0        Act.dF/Pred.dF  1.005591131
N. Function Calls        12        N. Gradient Calls         5
N. Hessian Calls         12        Preproces. Time           0
Time for Method           0        Effective Time            0


     Loglik: Intercept only = -1146.15 Model = -1137.25
        Chisquare=17.7974 with df=2 Prob=0.000136566

          Coefficients, ASEs, Z, and P Values

     Variable        Coef   AsyStdErr     Z_Value     P_Value
    Intercept  6.73234256  0.42395812  15.8797348  8.755e-057
          age -0.00580885  0.00693040 -0.83816924  0.40193566
      ph_ecog -0.32443045  0.08649460 -3.75087519  1.762e-004
      STRAT_1 -0.24408490  0.07919758 -3.08197408  0.00205633
      STRAT_2 -0.42345147  0.10669258 -3.96889323  7.221e-005


          Scale Estimate(s) (Scale=exp(Coef))

                  1     STRAT_1  0.78342112
                  2     STRAT_2  0.65478295


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

           Dense Symmetric Matrix (5 by 5)

       S |   Intercept        age    ph_ecog    STRAT_1    STRAT_2
       -------------------------------------------------------------
Intercept |   0.1797405
      age |  -0.0028453  4.80e-005
  ph_ecog |   0.0016839 -1.52e-004  0.0074813
  STRAT_1 |  -0.0011477  4.21e-005 -6.29e-004  0.0062723
  STRAT_2 |   0.0090870 -1.60e-004 -8.31e-004 -4.56e-004  0.0113833


      Training Data, Predicted Values, and Residuals
      **********************************************

              Dense Matrix (228 by 7)
```

	Event	Strata	Start	Time	LPrd
OBS_001	1.0000000	0	0	306.00000	394.67285
OBS_002	1.0000000	0	0	455.00000	565.29105
OBS_003	0	0	0	1010.0000	606.10118
OBS_004	1.0000000	0	0	210.00000	435.63627
OBS_005	1.0000000	0	0	883.00000	592.18053
OBS_006	0	0	0	1022.0000	394.67285
OBS_007	1.0000000	1.0000000	0	310.00000	295.44416
OBS_008	1.0000000	1.0000000	0	361.00000	290.34018
OBS_009	1.0000000	0	0	218.00000	445.87696
OBS_010	1.0000000	0	0	166.00000	307.70508
OBS_011	1.0000000	0	0	170.00000	435.63627
OBS_012	1.0000000	1.0000000	0	654.00000	295.44416
OBS_013	1.0000000	1.0000000	0	728.00000	408.67094
OBS_014	.	.	.	.	.
OBS_015	1.0000000	0	0	567.00000	435.63627
.....					
OBS_226	0	1.0000000	0	105.00000	283.67178
OBS_227	0	0	0	174.00000	413.44644
OBS_228	0	1.0000000	0	177.00000	433.11305

	PASE	Residual
OBS_001	39.623536	-88.672847
OBS_002	68.950481	-110.29105
OBS_003	65.544448	403.89882
OBS_004	30.823808	-225.63627
OBS_005	62.722311	290.81947
OBS_006	39.623536	627.32715
OBS_007	30.017567	14.555845
OBS_008	31.083562	70.659816
OBS_009	39.498039	-227.87696
OBS_010	32.313166	-141.70508
OBS_011	30.823808	-265.63627
OBS_012	30.017567	358.55584
OBS_013	29.238893	319.32906
OBS_014	.	.
OBS_015	30.823808	131.36373
.....		
OBS_226	33.951724	-178.67178
OBS_227	26.902221	-239.44644
OBS_228	29.171086	-256.11305

8. Left Censored Data for Tobit Analysis:

```
print "Tobit Model as in SAS/Stat Manual: nr=20";
tobit = [ 1   0.0 57.7   236,  2   0.7 50.9   283,
          3   0.0 48.5   207,  4   0.0 41.7   220,
          5   0.0 47.7   238,  6   0.0 59.8   216,
          7   0.0 44.3   284,  8   3.7 45.1   221,
          9   0.0 51.7   275, 10   3.0 50.0   269,
         11  10.4 46.8   207, 12   0.0 58.0   249,
         13   0.0 58.9   246, 14   0.0 40.0   277,
         15   1.5 34.1   231, 16   0.0 39.9   219,
         17   0.0 33.4   240, 18   3.5 48.1   266,
         19   6.1 46.6   214, 20   0.0 53.1   251 ];


  stat = (tobit[,2] .> 0);
  tobit = tobit -> stat;
  cnam = [" id durable  age quant status "];
  tobit = cname(tobit,cnam);
  print "TOBIT=",tobit;


  print "Surv(durable,durable > 0, type='left') ~ age + quant,\n";
  print "data=tobin, dist='gaussian')";
   modl = "2 = 3 4";
   optn = [ "status"         5 ,
            "strata"         6 ,
            "ytyp"       "left" ,
            "pinit"          2 ,
            "phist"          1 ,
            "print"          3 ];
   < gof,parm,covm,resi > = survreg("gauss",tobit,modl,optn);
```

```
          Survival Regression: Gaussian Distribution
                   Time Data: Left Censored

     Number Rows of Training Data. . . . . .          20
     Number Columns of Matrix  . . . . . .             5
     Number Training Observations. . . . . .          20
     Number Independent Covariates . . . . .           3
     Number (Train) Events . . . . . .                 7
     Column for Censor Variable. . . . . .             5
     Column for Strata Variable. . . . . .             6
     Type of Predictions . . . . . .            Response
     Type of Residuals . . . . . .              Response
     Initial value of Scale. . . . . .       0.50000000
```

Estimating the Intercept

Iter	LogLik	LL_Change	TestCrit	Gmax
0	-29.5631734	.	.	1.18398784
1	-29.4930697	0.07010371	0.00237696	0.05998636
2	-29.4922000	8.697e-004	2.949e-005	4.121e-004
3	-29.4921995	4.343e-007	1.472e-008	2.413e-007
4	-29.4921995	1.208e-013	3.997e-015	6.739e-014

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

Iteration Start:

N. Variables	4		
Criterion	31.14787840	Max Grad Entry	321.5831986

Iter	rest	nfun	act	optcrit	difcrit	maxgrad	ridge	rho
1	0	7	0	30.58009	0.567786	317.875	8.00000	0.19653
2	0	8	0	29.16079	1.419298	36.8104	4.00000	1.12203
3	0	9	0	28.96201	0.198781	3.55003	1.00000	1.12373
4	0	10	0	28.95595	6.1e-003	0.01434	0.25000	1.07736
5	0	11	0	28.95414	1.8e-003	0.01042	0.06250	1.93099
6	0	12	0	28.94017	0.013969	0.12001	0	0.99364
7	0	13	0	28.94013	4.2e-005	2e-003	0	1.00182
8	0	14	0	28.94013	4.1e-010	5e-008	0	1.00003

Successful Termination After 8 Iterations
GCONV convergence criterion satisfied.

Criterion	28.94013320	Max Grad Entry	5.0437e-008
Ridge (lambda)	0	Act.dF/Pred.dF	1.000025101
N. Function Calls	15	N. Gradient Calls	5
N. Hessian Calls	10	Preproces. Time	0
Time for Method	0	Effective Time	0

Loglik: Intercept only = -29.4922 Model = -28.9401
Chisquare=1.10413 with df=2 Prob=0.575759

Coefficients, ASEs, Z, and P Values

Variable	Coef	AsyStdErr	Z_Value	P_Value
Intercept	15.1448663	16.0794532	0.94187695	0.34625564
age	-0.12905928	0.21858360	-0.59043444	0.55489943

```
      quant -0.04554166  0.05825412 -0.78177589  0.43434629
    STRAT_1  1.71785092  0.31032272  5.53569175  3.100e-008

         Estimated Scale (exp(coef)) = 5.57254


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

             Dense Symmetric Matrix (4 by 4)

         S |   Intercept        age      quant    STRAT_1
         -----------------------------------------------
 Intercept |   258.54882
       age |  -1.7354582  0.0477788
     quant |  -0.7280315 -0.0019161  0.0033935
   STRAT_1 |   0.2385677 -0.0125409 -1.32e-004  0.0963002


        Training Data, Predicted Values, and Residuals
        **********************************************

                    Dense Matrix (20 by 7)

        |       Event     Strata      Start       Time       LPrd
     ---------------------------------------------------------------
OBS_01 |           0          0          0          0 -3.0496868
OBS_02 |   1.0000000          0          0  0.7000000 -4.3125418
OBS_03 |           0          0          0          0 -0.5416332
OBS_04 |           0          0          0          0 -0.2560716
OBS_05 |           0          0          0          0 -1.8501773
OBS_06 |           0          0          0          0 -2.4098780
OBS_07 |           0          0          0          0 -3.5062922
OBS_08 |   1.0000000          0          0  3.7000000 -0.7404149
OBS_09 |           0          0          0          0 -4.0514559
OBS_10 |   1.0000000          0          0  3.0000000 -3.5588052
OBS_11 |   1.0000000          0          0  10.400000 -0.3222324
OBS_12 |           0          0          0          0 -3.6804462
OBS_13 |           0          0          0          0 -3.6599746
OBS_14 |           0          0          0          0 -2.6325456
OBS_15 |   1.0000000          0          0  1.5000000  0.2238206
OBS_16 |           0          0          0          0  0.0217767
OBS_17 |           0          0          0          0 -0.0957128
OBS_18 |   1.0000000          0          0  3.5000000 -3.1769675
OBS_19 |   1.0000000          0          0  6.1000000 -0.6152122
OBS_20 |           0          0          0          0 -3.1391390
```

```
       |        PASE   Residual
     ------------------------
OBS_01 |   3.2462445  3.0496868
OBS_02 |   3.1910011  5.0125418
OBS_03 |   2.8510134  0.5416332
OBS_04 |   2.3770635  0.2560716
OBS_05 |   1.9676950  1.8501773
OBS_06 |   4.0207367  2.4098780
OBS_07 |   3.2444257  3.5062922
OBS_08 |   2.2250721  4.4404149
OBS_09 |   2.9231083  4.0514559
OBS_10 |   2.6049438  6.5588052
OBS_11 |   2.7634675  10.722232
OBS_12 |   3.2344825  3.6804462
OBS_13 |   3.3929324  3.6599746
OBS_14 |   3.2788470  2.6325456
OBS_15 |   3.2083745  1.2761794
OBS_16 |   2.5528602 -0.0217767
OBS_17 |   3.3487494  0.0957128
OBS_18 |   2.4306742  6.6769675
OBS_19 |   2.4809955  6.7152122
OBS_20 |   2.4918265  3.1391390

                  Total Computation Time: 0
```

5.12 Function `survprd`

< gof,res > = survprd(parm,covm,sopt,data,modl,optn<,clas>)

Purpose: The `survprd` function can be used for scoring a test data set using the parameter estimates and covariance matrix of estimates of a former training run.

Input: The first two input arguments correspond to output arguments of a former `survreg` call. The following input arguments should be the ssame as those used with that `survreg` call except for the name of data object and the `rtyp` and `ptyp` options.

parm This is either a p vector or a $p \times nc$ matrix containing the parameter estimates from a `survreg` training call. If this argument is a $p \times nc$ matrix it should correspond to the output object of a former `survreg` call, i.e. must contain the p estimates in its first column.

covm This is the symmetric $p \times p$ covariance matrix of parameter estimates of a former `survreg` call.

sopt This is a string option specifying the regression method (model) which is applied to the data:

"aalen" Aalen's additive model
"phcox" Cox's proportional hazards model
"extre" extreme distribution
"logis" logistic distribution
"gauss" Gaussian distribution
"weibu" Weibull distribution
"loglo" loglogistic distribution
"logno" lognormal distribution
"expon" exponential distribution (Weibull with fixed scale of 1)
"rayle" exponential distribution (Weibull with fixed scale of .5)

data This is a $N \times nc$ data set which contains in its nc columns:

- the dependent time (stop) variable
- the binary censor variable
- a set of n independent covariate variables
- an optional start variable
- an optional strata variable (for groups of observations)
- an optional offset variable
- an optional weight variable

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 column of the censor 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.)

Output: **gof** this is vector of some scalar results, see below for content.

res contains the input data, predicted values, and residuals. Currently for Aalen's method predicted values and residuals are not computed and returned.

The content of `gof` is:

1.

Restrictions: 1. Currently there are no residuals or predicted values for Aalen's method.

2. For GLIM estimation you can have stratified models only together with scale estimation. For k strata there are k additional weight parameters to estimate. A specified `"scale"` value is used as starting estimate for the weights.

Relationships: survreg() survcurv()

Examples: 1. PHCOX Regression: Ovarian cancer, N=26, see `survreg` function:

```
modl = "2 = 4:7";
optn = [ "status"       3 ,
         "phmet"  "efron" ,
         "rtyp"    "devi" ,
         "phist"        1 ,
         "print"        3 ];
< gof,parm,covm,resi,tres > = survreg("phcox",ovar,modl,optn,.,ovar);
```

Since we use the identical input data set as in the training the following call must obtain the same residual output `resi`:

```
modl = "2 = 4:7";
optn = [ "status"       3 ,
         "phmet"  "efron" ,
         "rtyp"    "devi" ,
         "pinit"        2 ,
         "print"        3 ];
< gof,resi > = survprd(parm,covm,"phcox",ovar,modl,optn);
```

2. Weibull Regression: Ovarian cancer, N=26, see `survreg` function:

```
print "Exponential Model: Weibull with fixed scale = 1: NRR";
print "rtyp=responce, ptyp=response: this is the default";
modl = "2 = 6 7";
optn = [ "status"         3 ,
         "scale"          1.,
         "rtyp"      "resp" ,
         "ptyp"      "resp" ,
         "phist"          1 ,
         "print"          3 ];
< gof,parm,covm,resi > = survreg("weibu",ovar,modl,optn);
```

Since we use the identical input data set as in the training the following call must obtain the same residual output `resi`:

```
modl = "2 = 6 7";
optn = [ "status"         3 ,
         "scale"          1.,
         "rtyp"      "resp" ,
         "ptyp"      "resp" ,
         "pinit"          2 ,
         "print"          3 ];
< gof,resi > = survprd(parm,covm,"weibu",ovar,modl,optn);
```

6 Illustration

6.1 Comparing `lrforw()` and `lrallv()` Results

6.1.1 Summary

We illustrate the differences between three approaches of linear regression for data sets with very many variables but not so many observations:

- stepwise forward selection `lrforw()`,
- stochastic search algorithm (approximating the result of the full enumeration) all variables subsets method,
- full enumeration with `lrallv()` for the all variables subsets problem, which is not always possible due to the size of the computational problem.

We are not aware of any software except CMAT which can solve linear regression problems of the size which `lrforw()` and `lrallv()` are able to.
The reference manual illustrates some similar differences for the `lrallv()` function in its *Examples* subsection for a small data set. Here we show results with two much larger data sets, especially in the number n of variables.
For all applications we select only $m = 3$ variables out of the n variables of the data set.
We show the results of full enumeration, sampling, forward stepwise, and the new `smp` for the following data sets:

- Boston Housing: nobs=506, nvar=14: all three methods obtain the optimal result
- Australian Data (binary response): nobs=690, nvar=15: all three methods obtain the optimal result
- NIR Spectra data set: nobs= 21, nvar= 268 full enumeration is best, sampling becomes close to full enumeration, and forward stepwise is slightly worse
- Caco2 Data (Bi & Bennett, 2002): nobs= 27, nvar= 714 full enumeration not possible, forward stepwise is better than sampling, only very large samples approach the good forward stepwise result
- Myeloma Data (binary response): nobs=105, nvar=7009: forward stepwise is better than sampling, obviously sample size is much too small; this application takes too long to run `smp` on my PC
- Affymetrix micro array data it_vsn.csv (binary response): nobs=37, nvar=22283 forward stepwise is better than sampling, obviously sample size of 1000 is much too small; this application takes too long to run `smp` on my PC

- Breast Cancer Data (van Veer, 2002) (binary response): nobs=98, nvar=24189 forward stepwise is better than sampling, obviously sample size of 1000 is much too small; this application takes too long to run `smp` on my PC

For the NIR data we find that stochastic sampling yields better results than the forward stepwise regression. For the CACO2 data however, stepwise forward selection finds a great result which cannot be matched with the stochastic algorithm. Therefore, it is recommended to compute both, forward stepwise and a few sampling solutions when full enumeration is computationally not feasible. For the remaining three very big problems stepwise forward selection yields better results than the stochastic algorithm maybe because the sample size is much too small for so many variables. Note, that `smp` and linear regression do not select the same variable set if some of the variables are highly correlated.

6.1.2 Boston Housing Data: nobs=506, nvar=14

The Boston housing data has 506 observations and 13 predictors with an interval response.

Algorithm	R^2	Time
Full Enumeration	0.67862416	1
Sample 100	0.67862416	0
Forward Stepwise	0.67862416	0

Forward selection and about 40 % sampling finds the best R^2 solution with three predictors which can be confirmed by full enumeration.

1. Order of Selection by Full Enumeration:

```
*****************************
Table of 10 best Rsquare Values
*****************************

 1 lstat    rm ptrat  0.67862416
 2 lstat    rm  chas  0.65140281
 3 lstat    rm     b  0.65055484
 4 lstat    rm   tax  0.64851477
 5 lstat    rm   dis  0.64678214
 6 lstat    rm  crim  0.64585205
 7 lstat    rm   rad  0.64278606
 8 lstat    rm indus  0.63999171
 9 lstat    rm    zn  0.63988560
10 lstat    rm   age  0.63903412
```

2. Order of Selection by Sampling:

```
*******************************
Table of 10 best Rsquare Values
*******************************

 1 lstat    rm ptrat  0.67862416
 2 lstat    rm ptrat  0.67862416
 3 lstat    rm  chas  0.65140281
 4 lstat    rm     b  0.65055484
 5 lstat    rm   tax  0.64851477
 6 lstat    rm   dis  0.64678214
 7 lstat    rm  crim  0.64585205
 8 lstat    rm   rad  0.64278606
 9 lstat    rm indus  0.63999171
10 lstat    rm    zn  0.63988560
```

3. Order of Selection by Forward Stepwise:

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

Step	Effect	DF	Rsquared	F_value	pval	SSE	BIC
1	lstat	1	0.5441	601.61788	0.0000	23243.914	1853.0250
2	rm	1	0.0944	131.39417	0.0000	4033.0720	1737.6122
3	ptrat	1	0.0401	62.579071	0.0000	1711.3239	1680.1951

4. Order of Selection by SMP: Stochastic Matching Pursuit

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

		Variable	Percent	Nsel
A	6*	rm	97.566667	2927
B	13*	lstat	95.100000	2853
C	11*	ptrat	94.100000	2823
D	8*	dis	92.600000	2778
E	12*	b	90.600000	2718
F	5*	nox	85.400000	2562
G	4*	chas	81.933333	2458
H	9	rad	48.466667	1454
I	10	tax	41.066667	1232
J	2	zn	35.400000	1062

5. Order of Selection by SMP: Gibbs Sampler

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

		Variable	Percent	Nsel
A	6*	rm	99.866667	2996
B	11*	ptrat	98.333333	2950
C	13*	lstat	85.300000	2559
D	8*	dis	81.966667	2459
E	12*	b	81.766667	2453
F	4*	chas	75.400000	2262
G	5*	nox	75.033333	2251
H	10*	tax	61.100000	1833
I	2*	zn	60.166667	1805
J	1*	crim	57.666667	1730

This is surprising since for a subset of 30 observations of the Boston Housing data the forward stepwise algorithm was not able to find the optimal solution (see the *CMAT Reference Manual* for the `lrallv()` function).

6.1.3 Australian (binary response) Data: nobs=690, nvar=15

The Australian data set has 690 observations and 14 predictors with a binary response. The following table shows two results, one ([1]) for the best R^2 and another ([2]) for the best classification accuracy. (Note, that the fit is very poor for $m = 3$ selected variables.)

Algorithm	R^2	Accuracy	Time
Full Enumeration [1]	0.02096169	67.82608696	1
Full Enumeration [2]	0.00883089	67.97101449	2
Sample 100 [1]	0.02096169	67.82608696	0
Sample 100 [2]	0.00883089	67.97101449	1
Forward Stepwise	0.02096169	67.82608696	0

Forward selection and about 40 % sampling finds the best R^2 solution with three predictors which can be confirmed by full enumeration. However, the best R^2 subset is not the best with regard to the classification accuracy. Note, that SMP is not designed for categorical response.

1. Order of Selection by Full enumeration:

```
*****************************
Table of 10 best Rsquare Values
*****************************
```

```
 1  X7  X9 X13   67.83  0.02096169
 2  X7  X4  X9   67.83  0.02050930
 3  X7  X9  X3   67.25  0.01950499
 4  X7  X4 X13   67.83  0.01895938
 5  X7  X9 X12   67.83  0.01884334
 6  X7  X9 X11   67.83  0.01857289
 7  X7  X9  X6   67.83  0.01806308
 8  X7  X4 X12   67.68  0.01787285
 9  X7  X4  X3   67.54  0.01755602
10  X7  X9 X15   67.83  0.01714758
```

2. Order of Selection by Sampling:

```
*******************************
Table of 10 best Rsquare Values
*******************************

 1  X7  X9 X13   67.83  0.02096169
 2  X7  X4  X9   67.83  0.02050930
 3  X7  X4  X9   67.83  0.02050930
 4  X7  X9  X3   67.25  0.01950499
 5  X7  X4 X13   67.83  0.01895938
 6  X7  X9 X12   67.83  0.01884334
 7  X7  X9 X11   67.83  0.01857289
 8  X7  X9  X6   67.83  0.01806308
 9  X7  X4 X12   67.68  0.01787285
10  X7  X4 X12   67.68  0.01787285
```

3. Order of Selection by Forward Stepwise:

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

Step  Effect  DF Rsquared    F_value    pval        SSE        BIC

   1      X7   1   0.0094  6.5361572  0.0108  1.4170245 -1050.8565
   2      X9   1   0.0076  5.2994742  0.0216  1.1417791 -1054.1441
   3     X13   1   0.0040  2.7803387  0.0959  0.5974795 -1054.9146
```

4. Order of Selection by SMP: Stochastic Matching Pursuit

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

                 Variable    Percent        Nsel
      A      6*         X7  62.866667        1886
      B      3          X4  37.366667        1121
      C     11         X12  36.166667        1085
      D      8          X9  25.500000         765
      E      2          X3  16.266667         488
      F      1          X2  12.633333         379
      G      5          X6  10.266667         308
      H      4          X5   7.733333         232
      I     12         X13   7.633333         229
      J     14         X15   6.233333         187
```

5. Order of Selection by SMP: Gibbs Sampler

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

                 Variable    Percent        Nsel
      A      6*         X7  76.033333        2281
      B      3*         X4  66.900000        2007
      C      8*         X9  64.066667        1922
      D     12*        X13  57.100000        1713
      E     11         X12  49.166667        1475
      F      5          X6  32.500000         975
      G      1          X2  26.766667         803
      H      2          X3  25.233333         757
      I     10         X11  20.300000         609
      J      4          X5  19.266667         578
```

6.1.4 NIR Spectra Data: nobs= 21, nvar= 268

The following shows the data input:

```
  print "NIR Spectra data set: train: nr=21, test: nr=7: nvar=268";
  options NOECHO;
#include "..\\tdata\\nir.dat"
  options ECHO;
  nr = nrow(xtrn); nc = ncol(xtrn);
  print "nrtrn,nctrn=",nr,nc;
  cdat = xtrn -> ytrn‘; /* attrib(cdat); */
  test = xtst -> ytst‘; /* attrib(test); */
```

The following output shows that the stochastic search with even a very small sample of 1000 different subsets, which is only 0.032 % out of the 3172316 possible combinations of 3 variables from 268, performs better than stepwise forward selection. Naturally, the R^2 value is larger for full enumeration than for the stochastic search and forward stepwise, but takes also much more computer time. But note, that a sample of 100,000 subsets (which is only 3.152 % of all possible subsets) yields a solution with an almost optimal R^2 value.

Algorithm	R^2	Time
Full Enumeration	0.99997500	5115
Sample 1000	0.99988789	2
Sample 100000	0.99996351	186
Forward Stepwise	0.99942670	0

1. Forward Stepwise:

```
modl = "269 = 1:268";
optn = [ "print"         2 ,
         "vers"            ,
         "maxst"         3 ,
         "selc"       "r2" ,
         "sing"     1.e-12 ,
         "pobs"            ];
< gof,parm,yptr > = lrforw(cdat,modl,optn);
```

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

Step  Effect  DF Rsquared    F_value    pval        SSE        BIC

   1     X40   1   0.9607  465.01940  0.0000  17941.241  81.005481
   2    X241   1   0.0356  173.41643  0.0000  664.11920  33.905363
   3    X230   1   0.0031  92.458010  0.0000  58.227118 -2.3763361
```

```
                 ***********************************
                 Final ANOVA Table for Target: Y[269]
                 ***********************************

               Effect    DF    Rsquared  Sum-of-Squares

                Model     3  0.99942670  18663.58733991
                Error    17           .   10.70606009120
                Total    20           .   18674.29340000
```

```
MSE    0.62976824       SBC -26.3260521
AIC   -6.14796235       BIC -2.37633605
F      9878.54798       Prb  9.589e-028
CP     4.00000000       AR2  0.99932552


      Total Processing Time: 0
```

2. Full Enumeration:

```
modl = "269 = 1:268";
optn = [ "print"         2 ,
         "vers"            ,
         "best"         10 ,
         "sing"     1.e-12 ,
         "pobs"            ];
< gof,best,parm,yptr > = lrallv(3,cdat,modl,optn);


There are  3172316 regression computations with 3 coefficients.


          ***********************************
          Final ANOVA Table for Target: Y[269]
          ***********************************

        Effect    DF    Rsquared  Sum-of-Squares

         Model     3  0.99997500  18673.82658855
         Error    17           .   0.466811445477
         Total    20           .   18674.29340000

         MSE    0.02745950       SBC -92.1114880
         AIC   -71.9333982       BIC -68.1617720
         F      226683.282       Prb  2.617e-039
         CP     4.00000000       AR2  0.99997059


            ******************************
            Table of 10 best Rsquare Values
            ******************************

            1 X215 X164 X225  0.99997500
```

```
 2 X213 X165 X225  0.99997495
 3 X212 X165 X226  0.99997467
 4 X214 X164 X225  0.99997422
 5 X214 X160  X25  0.99997335
 6 X212 X166 X226  0.99997310
 7 X215 X160  X25  0.99997302
 8 X214 X161  X25  0.99997279
 9 X213 X166 X225  0.99997245
10 X215 X165 X224  0.99997209


Total Processing Time: 5925
```

3. Select only 1000 Samples:

```
modl = "269 = 1:268";
optn = [ "print"         2 ,
         "vers"           ,
         "samp"       1000 ,
         "best"         10 ,
         "sing"     1.e-12 ,
         "seed"        123 ,
         "pobs"           ];
< gof,best,parm,yptr > = lrallv(3,cdat,modl,optn);


There are  3172316 regression computations with 3 coefficients.
    Using 1000 sample (  0.032 \%) out of  3172316 possible
                         combinations.


                  ************************************
                  Final ANOVA Table for Target: Y[269]
                  ************************************

                Effect    DF    Rsquared  Sum-of-Squares

                 Model     3  0.99988789  18672.19977099
                 Error    17           .   2.093629006065
                 Total    20           .   18674.29340000

                 MSE    0.12315465       SBC -60.5961835
                 AIC   -40.4180937       BIC -36.6464674
                 F      50538.6254       Prb  9.071e-034
                 CP     4.00000000       AR2  0.99986810
```

```
*******************************
Table of 10 best Rsquare Values
*******************************

   1 X229 X158 X228  0.99988789
   2 X216 X159  X29  0.99977131
   3  X37  X71 X228  0.99974983
   4 X217 X218 X161  0.99974718
   5 X212 X227 X238  0.99971753
   6 X206 X166 X133  0.99967362
   7  X44  X78  X49  0.99963923
   8  X72 X164  X30  0.99962780
   9 X211  X89 X246  0.99961722
  10  X51  X80  X52  0.99961375


  Total Processing Time: 2
```

4. Select only 3.152 % of all subsets, i.e. 100,000 Samples:

```
Using 100000 sample (  3.152 \%) out of  3172316 possible
                    combinations.


 ***********************************
 Final ANOVA Table for Target: Y[269]
 ***********************************

Effect    DF    Rsquared  Sum-of-Squares

 Model     3  0.99996351  18673.61199234
 Error    17           .   0.681407656486
 Total    20           .   18674.29340000

 MSE    0.04008280       SBC -84.1685462
 AIC   -63.9904565       BIC -60.2188302
 F      155291.966       Prb  6.516e-038
 CP     4.00000000       AR2  0.99995707


*******************************
Table of 10 best Rsquare Values
*******************************
```

```
 1 X157 X232  X28  0.99996351
 2 X231 X157 X132  0.99995001
 3 X157 X233  X27  0.99994663
 4 X165 X203  X26  0.99994558
 5 X209 X165  X24  0.99994364
 6 X165 X203  X25  0.99994355
 7 X164 X167  X26  0.99994220
 8 X215 X165 X219  0.99994097
 9 X165 X204  X26  0.99993714
10 X231 X157 X217  0.99993624


Total Processing Time: 186
```

5. SMP: Stocastic Matching Pursuit

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

              Variable    Percent        Nsel
A    129*         X129  100.00000        3000
B     22*          X22  100.00000        3000
C    111*         X111  100.00000        3000
D     83*          X83  100.00000        3000
E     37*          X37  98.266667        2948
F     96*          X96  96.200000        2886
G     74*          X74  96.066667        2882
H    112*         X112  94.333333        2830
I    182*         X182  90.633333        2719
J      8*           X8  87.600000        2628
```

6. SMP: Componentwise Gibbs sampler

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

              Variable    Percent        Nsel
A     57*          X57  56.000000        1680
B    153*         X153  55.800000        1674
C    240*         X240  55.666667        1670
D    235*         X235  54.900000        1647
E    234*         X234  54.800000        1644
F     32*          X32  54.633333        1639
```

```
G    237*        X237  54.533333        1636
H    248*        X248  54.033333        1621
I    151*        X151  53.866667        1616
J    238*        X238  53.866667        1616
```

6.1.5 CACO2 Data: nobs= 27, nvar= 714

The following shows the data input:

```
/* Caco: nobs= 27, nvar= 714 (Bi & Bennett) */
fo10 = " %16s %16s %16s %16s %16s %16s %16s %16s %16s %16s";
form = fo10;
for (j = 2; j <= 71; j++) form = strcat(form,fo10);
fo4 = " %16s %16s %16s %16s";
form = strcat(form,fo4);


fid = fopen("..\\tdata\\caco_nam.txt","r");
/* sscanf() is faster when nr and nc are specified: */
c_nam = fscanf(fid,form,.,1);
/* print c_nam; */


/*--- read data: 715 cols, first column is obs number ---*/
fid = fopen("..\\tdata\\caco.dat","r");
form = fo1 = " %g %g %g %g %g %g %g %g %g %g";
for (j = 2; j <= 71; j++) form = strcat(form,fo1);
fo5 = " %g %g %g %g %g";
form = strcat(form,fo5);
c_dat = fscanf(fid,form,27,.);
nr = nrow(c_dat); nc = ncol(c_dat);
print "CACO2: Observations of c_dat.dat:",nr;
print "CACO2: Columns of c_dat.dat:",nc;
caco = c_dat[,2:715]; /* cut out col 1 */
```

Full enumeration would have to evaluate 60157236 different subsets of three variables. This is a computationally not feasible task for most of todays PCs. The following table shows that for this example stepwise forward selection works very well compared with the stochastic algorithm. Even large sample numbers do not beat the good R^2 found with stepwise forward regression. Note, that a sample of 1000,000 subsets is only 1.662 % of all 60157236 full enumeration subsets.

Algorithm	R^2	Time
Forward Stepwise	0.96430730	0
Sample 1000	0.90855945	5
Sample 10000	0.91401189	63
Sample 50000	0.92303618	290
Sample 100000	0.92611891	603
Sample 200000	0.93851506	1227
Sample 300000	0.95214617	1937
Sample 500000	0.95489169	3288
Sample 1000000	0.95489169	7015

The obvious reason why for this specific application the stepwise solution with $(X679, X689, X32)$ is not easy to match with the stochastic search algorithm is, that the univariate R^2 values for two of the variables, X689=0.0750 and X32=0.0484, belong to the smallest of all. The stochastic algorithm works best when the best solution consists of variables which almost all have some reasonably large univariate R^2 values.

1. Forward Stepwise:

```
modl = "714 = 1:713";
optn = [ "print"         2 ,
         "vers"            ,
         "maxst"         3 ,
         "selc"       "r2" ,
         "sing"     1.e-12 ,
         "pobs"            ];
< gof,parm,yptr,yptt > = lrforw(caco,modl,optn);
print "GOF=",gof;
```

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

Step  Effect  DF Rsquared    F_value    pval        SSE        BIC

   1    X679   1   0.8409  132.11881  0.0000  21.863003 -44.341202
   2    X689   1   0.0750  21.412501  0.0001  1.9506401 -59.148459
   3     X32   1   0.0484  31.187126  0.0000  1.2583465 -79.674023


                ***********************************
                 Final ANOVA Table for Target: Y[714]
                 ***********************************
```

```
Effect    DF    Rsquared  Sum-of-Squares

 Model     3  0.96430730  25.07198947890
 Error    23           .   0.928010133711
 Total    26           .   25.99999961262

 MSE    0.04034827       SBC -104.188184
 AIC   -83.0048363       BIC -79.6740234
 F      207.129835       Prb  8.790e-017
 CP     4.00000000       AR2  0.95965173


      Total Processing Time: 0
```

2. R^2 Select samp=1000:

```
modl = "714 = 1:713";
optn = [ "print"         2 ,
         "vers"           ,
         "samp"       1000 ,
         "best"         10 ,
         "sing"     1.e-12 ,
         "seed"        123 ,
         "pobs"           ];
< gof,best,parm,yptr,yptt > = lrallv(3,caco,modl,optn);
print "GOF=",gof;
print "Best=",best;
print "Parm=",parm;
```

```
*****************************
Table of 10 best Rsquare Values
*****************************

   1 X679 X672 X696  0.90855945
   2 X679 X253 X242  0.88009579
   3 X679 X176 X354  0.87526611
   4 X679  X75  X62  0.87397422
   5 X679  X52 X223  0.87224688
   6 X679 X252  X59  0.87176576
   7 X679 X146 X463  0.87170882
   8 X679  X70 X316  0.86914254
   9 X679  X75 X258  0.86800443
  10 X679 X149 X383  0.86624873
```

```
Total Processing Time: 5
```

3. R^2 Select nsamp=10,000:

```
*****************************
 Table of 10 best Rsquare Values
 *****************************

   1 X679 X672  X64  0.91401189
   2 X679 X327 X245  0.91341483
   3 X679 X672 X136  0.91030254
   4 X679 X672 X696  0.90855945
   5 X679 X672 X298  0.90523423
   6 X679 X672 X165  0.90512865
   7 X684 X146 X183  0.90367865
   8 X679 X682  X57  0.90238402
   9 X679 X644 X411  0.90073814
  10 X679 X489 X332  0.89634095


   Total Processing Time: 63
```

4. Select nsamp=50,000:

```
*****************************
Table of 10 best Rsquare Values
*****************************

   1 X679 X185 X253  0.92303618
   2 X679 X327  X64  0.91975302
   3 X679 X613 X253  0.91779284
   4 X679 X672 X183  0.91614979
   5 X679 X672  X64  0.91401189
   6 X679 X327 X245  0.91341483
   7 X679 X672 X633  0.91338442
   8 X679 X672 X136  0.91030254
   9 X679 X672 X378  0.91030112
  10 X679 X672 X396  0.90955451


  Total Processing Time: 290
```

5. Select nsamp= 100,000:

```
*******************************
Table of 10 best Rsquare Values
*******************************

   1 X679 X672  X21  0.92611891
   2 X679 X185 X253  0.92303618
   3 X679 X684 X626  0.92117801
   4 X679 X686 X672  0.92086329
   5 X679 X672 X509  0.92005567
   6 X679 X327  X64  0.91975302
   7 X679 X672 X311  0.91896993
   8 X679 X613 X253  0.91779284
   9 X679 X672 X183  0.91614979
  10 X679 X687 X332  0.91571923


  Total Processing Time: 603
```

6. Select nsamp= 200,000:

```
*******************************
Table of 10 best Rsquare Values
*******************************

   1 X679 X672 X266  0.93851506
   2 X679 X672  X98  0.93237781
   3 X679 X253 X644  0.92876647
   4 X679 X672  X21  0.92611891
   5 X679 X185 X253  0.92303618
   6 X679 X684 X626  0.92117801
   7 X679 X686 X672  0.92086329
   8 X679 X672 X509  0.92005567
   9 X679 X327  X64  0.91975302
  10 X679 X672 X311  0.91896993


  Total Processing Time: 1227
```

7. Select nsamp= 300,000:

```
*******************************
Table of 10 best Rsquare Values
*******************************
```

```
 1 X679 X672 X121  0.95214617
 2 X679 X672 X266  0.93851506
 3 X679 X672  X98  0.93237781
 4 X679 X253 X644  0.92876647
 5 X679 X672  X21  0.92611891
 6 X679 X185 X253  0.92303618
 7 X679 X684 X626  0.92117801
 8 X679 X686 X672  0.92086329
 9 X679 X672 X517  0.92083303
10 X679 X672 X509  0.92005567


Total Processing Time: 1937
```

8. R^2 Select nsamp= 500,000:

```
Using 500000 sample (  0.831 \%) out of  60157236 possible combinations.

*******************************
 Table of 10 best Rsquare Values
 *******************************

   1 X679 X672 X253  0.95489169
   2 X679 X672 X121  0.95214617
   3 X679 X672 X266  0.93851506
   4 X679 X672  X98  0.93237781
   5 X679 X253 X644  0.92876647
   6 X679 X672 X626  0.92676630
   7 X679 X672  X21  0.92611891
   8 X679 X185 X253  0.92303618
   9  X75 X684 X712  0.92243680
  10 X679 X684 X626  0.92117801


  Total Processing Time: 3288
```

9. Select nsamp= 1000,000:

```
Using 1000000 sample (  1.662 \%) out of  60157236 all combinations.


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

```
 Final ANOVA Table for Target: Y[714]
 ************************************

Effect    DF    Rsquared  Sum-of-Squares

 Model     3  0.95489169  24.82718348911
 Error    23           .   1.172816123509
 Total    26           .   25.99999961262

 MSE    0.05099201       SBC -97.8669322
 AIC   -76.6835848       BIC -73.3527719
 F      162.294614       Prb  1.292e-015
 CP     4.00000000       AR2  0.94900799


   ******************************
   Table of 10 best Rsquare Values
   ******************************

      1 X679 X672 X253  0.95489169
      2 X684  X46 X517  0.95236907
      3 X679 X672 X121  0.95214617
      4 X679 X672 X266  0.93851506
      5 X679 X327  X46  0.93466779
      6 X679 X672  X98  0.93237781
      7 X327 X684 X517  0.93215911
      8 X679 X672  X72  0.93068327
      9  X75 X684 X183  0.93009065
     10 X679 X253 X644  0.92876647


     Total Processing Time: 7015
```

10. SMP: Stochastic Matching Pursuit

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

                Variable    Percent        Nsel
A     89*          X89  100.00000        3000
B    140*         X140  100.00000        3000
C    270*         X270  100.00000        3000
D    526*         X526  100.00000        3000
E    405*         X405  100.00000        3000
```

```
F    562*        X562  100.00000        3000
G    679*        X679  100.00000        3000
H    386*        X386  100.00000        3000
I    587*        X587  100.00000        3000
J    432*        X432  100.00000        3000
```

11. SMP: Componentwise Gibbs sampler

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

                Variable    Percent        Nsel
A      1*             X1  75.833333        2275
B    556*          X556  64.833333        1945
C    441*          X441  64.600000        1938
D    479*          X479  64.600000        1938
E    574*          X574  64.433333        1933
F    526*          X526  64.133333        1924
G    622*          X622  64.066667        1922
H    417*          X417  64.033333        1921
I    532*          X532  63.933333        1918
J    401*          X401  63.866667        1916
```

6.1.6 Myeloma Data (binary response): nobs=105, nvar=7009

The following code shows the data input:

```
print "Myeloma Data: nobs=105, nvar=7009";
myelo = fil2obj("..\\save\\myelo");
cnam = cname(myelo);
print "nc=", nc = ncol(myelo),
      " nr=",nr = nrow(myelo);
/* print myelo[,7009]; */
print "Compute univariate stat";
sopt = [ "ari" "std" "med" ];
mom = univar(myelo,sopt);
print mom[,1] -> mom[,7009];
```

The following table shows two results, one ([1]) for the best R^2 value and another ([2]) for the best classification accuracy.

Algorithm	R^2	Accuracy	Time
Forward Stepwise	0.91759303	100	13
Sample 1000 [1]	0.90376994	100	369
Sample 1000 [2]	0.90376994	100	570
Sample 10000 [1]	0.90376994	100	3595
Sample 10000 [2]	0.90376994	100	5519

1. Forward Stepwise:

```
clas = 7009;
modl = "7009 = 1 : 7008";
optn = [ "print"         2 ,
         "vers"           ,
         "maxst"         3 ,
         "selc"       "r2" ,
         "sing"     1.e-12 ,
         "pobs"            ];
< gof,parm,yptr > = lrforw(myelo,modl,optn,clas);
```

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

Step  Effect  DF Rsquared    F_value    pval        SSE        BIC

  1   X5454   1   0.8144  451.95804  0.0000  17.792709 -335.59644
  2   X6073   1   0.0716  64.024920  0.0000  1.5637127 -384.65095
  3   X5922   1   0.0316  38.753079  0.0000  0.6908009 -416.61144
```

```
                ************************************
                Final ANOVA Table for Target: Y[7009]
                ************************************

               Effect    DF    Rsquared  Sum-of-Squares

                Model     3  0.91759303  20.04722291764
                Error   101           .   1.800396129980
                Total   104           .   21.84761904762

                MSE    0.01782570       SBC -445.540973
                AIC   -418.925132       BIC -416.611437
                F      374.874818       Prb  1.397e-054
                CP     4.00000000       AR2  0.91514530
```

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

                          |    Predicted
                 Observed |        0       1
                 ---------|-----------------
                        0 |       31       0
                        1 |        0      74


                  Total Processing Time: 13
```

2. R^2 Select nsamp= 1000:

```
print "Use Rsqu: w/o AOV w/o Group: FOSSTI sample select 3";
clas = 7009;
modl = "7009 = 1 : 7008";
optn = [ "print"         2 ,
         "vers"           ,
         "samp"       1000 ,
         "best"         10 ,
         "sing"     1.e-12 ,
         "seed"        123 ,
         "pobs"           ];
< gof,best,parm,yptr > = lrallv(3,myelo,modl,optn,clas);
```

```
There are 56970889416 regression computations with 3 coefficients.
     Using 1000 sample (  0.000 \%) out of 56970889416 possible
                          combinations.


                  ************************************
                  Final ANOVA Table for Target: Y[7009]
                  ************************************

                 Effect    DF    Rsquared  Sum-of-Squares

                  Model     3  0.90376994  19.74522131144
                  Error   101           .   2.102397736181
                  Total   105           .   21.84761904762

                  MSE    0.02081582       SBC -429.258439
                  AIC   -402.642597       BIC -400.328902
                  F      316.189355       Prb  3.491e-051
```

```
*******************************
Table of 10 best Rsquare Values
*******************************

 1 X5454 X6073 X2348  100.00  0.90376994
 2 X6073 X6780  X684  100.00  0.85453741
 3 X5454 X5128 X1980   99.05  0.85391386
 4 X5454 X1081 X6075  100.00  0.83022280
 5 X5454 X2447 X6892   99.05  0.82623598
 6 X2306 X5922 X6379   97.14  0.81632868
 7 X1601 X2547 X2812   95.24  0.81140816
 8 X3919 X6212 X5293   98.10  0.79889250
 9 X2348 X3650 X5132   98.10  0.79519196
10 X5082 X2008  X923   97.14  0.79109821


*******************************
Table of all Selection Criteria
*******************************

 N    Accuracy     Rsquare     Prob(F)         AIC         BIC
 1  100.000000  0.90376994  3.491e-051 -402.642597 -400.328902
   -429.258439
 2  100.000000  0.85453741  3.916e-042 -359.258995 -356.945301
   -385.874837
 3  99.0476190  0.85391386  4.858e-042 -358.809861 -356.496166
   -385.425703
 4  100.000000  0.83022280  9.477e-039 -343.029351 -340.715656
   -369.645192
 5  99.0476190  0.82623598  3.053e-038 -340.592180 -338.278485
   -367.208022
 6  97.1428571  0.81632868  4.992e-037 -334.769964 -332.456269
   -361.385805
 7  95.2380952  0.81140816  1.892e-036 -331.994049 -329.680354
   -358.609890
 8  98.0952381  0.79889250  4.817e-035 -325.247328 -322.933634
   -351.863170
 9  98.0952381  0.79519196  1.207e-034 -323.332804 -321.019109
   -349.948645
10  97.1428571  0.79109821  3.270e-034 -321.254742 -318.941048
   -347.870584


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

```
          |    Predicted
Observed  |        0       1
----------|-----------------
       0  |       31       0
       1  |        0      74


Total Processing Time: 369
```

3. Acc Select nsamp= 1000:

```
 ************************************
 Final ANOVA Table for Target: Y[7009]
 ************************************

Effect    DF    Rsquared  Sum-of-Squares

 Model     3  0.90376994  19.74522131144
 Error   101           .   2.102397736181
 Total   105           .   21.84761904762

 MSE    0.02081582       SBC -429.258439
 AIC   -402.642597       BIC -400.328902
 F      316.189355       Prb  3.491e-051


    ******************************
    Table of 10 best Rsquare Values
    ******************************

 1 X5454 X6073 X2348  100.00  0.90376994
 2 X6073 X6780  X684  100.00  0.85453741
 3 X5454 X1081 X6075  100.00  0.83022280
 4 X5454 X5128 X1980   99.05  0.85391386
 5 X5454 X2447 X6892   99.05  0.82623598
 6 X3919 X1153 X2840   99.05  0.77519131
 7 X6457 X6843 X6169   99.05  0.73812058
 8 X3919 X6212 X5293   98.10  0.79889250
 9 X2348 X3650 X5132   98.10  0.79519196
10 X3919 X2583 X6488   98.10  0.77845093


    ******************************
    Table of all Selection Criteria
```

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

 N    Accuracy     Rsquare     Prob(F)         AIC         BIC
 1  100.000000  0.90376994  3.491e-051 -402.642597 -400.328902
   -429.258439
 2  100.000000  0.85453741  3.916e-042 -359.258995 -356.945301
   -385.874837
 3  100.000000  0.83022280  9.477e-039 -343.029351 -340.715656
   -369.645192
 4  99.0476190  0.85391386  4.858e-042 -358.809861 -356.496166
   -385.425703
 5  99.0476190  0.82623598  3.053e-038 -340.592180 -338.278485
   -367.208022
 6  99.0476190  0.77519131  1.318e-032 -313.549258 -311.235564
   -340.165100
 7  99.0476190  0.73812058  2.864e-029 -297.522650 -295.208955
   -324.138491
 8  98.0952381  0.79889250  4.817e-035 -325.247328 -322.933634
   -351.863170
 9  98.0952381  0.79519196  1.207e-034 -323.332804 -321.019109
   -349.948645
10  98.0952381  0.77845093  6.315e-033 -315.082854 -312.769159
   -341.698695


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

                       |    Predicted
             Observed  |        0       1
             ----------|-----------------
                    0  |       31       0
                    1  |        0      74


             Total Processing Time: 570
```

4. R^2 Select nsamp= 10000:

```
          ************************************
          Final ANOVA Table for Target: Y[7009]
          ************************************

         Effect    DF    Rsquared  Sum-of-Squares
```

```
Model     3  0.90376994  19.74522131144
Error   101           .   2.102397736181
Total   105           .   21.84761904762

MSE    0.02081582       SBC -429.258439
AIC   -402.642597       BIC -400.328902
F      316.189355       Prb  3.491e-051
```

```
*******************************
Table of 10 best Rsquare Values
*******************************

 1 X5454 X6073 X2348  100.00  0.90376994
 2 X5454 X3275 X6693   99.05  0.86973895
 3 X5454 X2697 X2918  100.00  0.86640589
 4 X6073 X2697 X1354   99.05  0.86504938
 5 X5454 X2697 X5778  100.00  0.86358141
 6 X5454 X3833 X3305  100.00  0.85710267
 7 X6073 X2685 X6623   99.05  0.85612819
 8 X6073 X6780  X684  100.00  0.85453741
 9 X5454 X5128 X1980   99.05  0.85391386
10 X5454 X5182 X5719   99.05  0.84987658
```

```
*******************************
Table of all Selection Criteria
*******************************

N    Accuracy     Rsquare     Prob(F)         AIC         BIC
1  100.000000  0.90376994  3.491e-051 -402.642597 -400.328902
  -429.258439
2  99.0476190  0.86973895  1.499e-044 -370.848729 -368.535034
  -397.464570
3  100.000000  0.86640589  5.359e-044 -368.195839 -365.882144
  -394.811680
4  99.0476190  0.86504938  8.919e-044 -367.135048 -364.821353
  -393.750889
5  100.000000  0.86358141  1.539e-043 -365.999042 -363.685347
  -392.614883
6  100.000000  0.85710267  1.597e-042 -361.127214 -358.813519
  -387.743055
7  99.0476190  0.85612819  2.249e-042 -360.413602 -358.099908
  -387.029444
8  100.000000  0.85453741  3.916e-042 -359.258995 -356.945301
```

```
   -385.874837
 9  99.0476190  0.85391386  4.858e-042 -358.809861 -356.496166
   -385.425703
10  99.0476190  0.84987658  1.920e-041 -355.947424 -353.633729
   -382.563265
```

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

                             |    Predicted
                   Observed  |        0       1
                   ----------|-----------------
                          0  |       31       0
                          1  |        0      74


                   Total Processing Time: 3959
```

5. Acc Select nsamp= 10000:

```
 ************************************
 Final ANOVA Table for Target: Y[7009]
 ************************************

Effect    DF    Rsquared  Sum-of-Squares

 Model     3  0.90376994  19.74522131144
 Error   101           .   2.102397736181
 Total   105           .   21.84761904762

 MSE    0.02081582       SBC -429.258439
 AIC   -402.642597       BIC -400.328902
 F      316.189355       Prb  3.491e-051


    ******************************
    Table of 10 best Rsquare Values
    ******************************

 1 X5454 X6073 X2348  100.00  0.90376994
 2 X5454 X2697 X2918  100.00  0.86640589
 3 X5454 X2697 X5778  100.00  0.86358141
 4 X5454 X3833 X3305  100.00  0.85710267
```

```
 5 X6073 X6780  X684  100.00  0.85453741
 6 X5454 X3105 X6343  100.00  0.84479624
 7 X5454  X602 X4377  100.00  0.83856462
 8 X5454 X5291 X5279  100.00  0.83747073
 9 X6073 X5505 X2917  100.00  0.83477209
10 X5454 X4619 X6887  100.00  0.83470553
```

Table of all Selection Criteria

N	Accuracy	Rsquare	Prob(F)	AIC	BIC
1	100.000000	0.90376994	3.491e-051	-402.642597	-400.328902
	-429.258439				
2	100.000000	0.86640589	5.359e-044	-368.195839	-365.882144
	-394.811680				
3	100.000000	0.86358141	1.539e-043	-365.999042	-363.685347
	-392.614883				
4	100.000000	0.85710267	1.597e-042	-361.127214	-358.813519
	-387.743055				
5	100.000000	0.85453741	3.916e-042	-359.258995	-356.945301
	-385.874837				
6	100.000000	0.84479624	1.028e-040	-352.452909	-350.139215
	-379.068751				
7	100.000000	0.83856462	7.479e-040	-348.319471	-346.005776
	-374.935312				
8	100.000000	0.83747073	1.051e-039	-347.610385	-345.296690
	-374.226226				
9	100.000000	0.83477209	2.411e-039	-345.881276	-343.567581
	-372.497117				
10	100.000000	0.83470553	2.460e-039	-345.838991	-343.525296
	-372.454832				

Classification Table

Observed	Predicted 0	Predicted 1
0	31	0
1	0	74

Total Processing Time: 5519

6.1.7 Genetic data it_vsn.csv: nobs=37, nvar=22283

```
myelo = fil2obj("..\\save\\affym");
cnam = cname(myelo);
print "nc=", nc = ncol(myelo),
      "nr=", nr = nrow(myelo);
/* print myelo[,7009]; */
print "Compute univariate stat";
sopt = [ "ari" "std" "med" ];
mom = univar(myelo,sopt);
print mom[,1] -> mom[,7009];
```

The following table shows that forward selection is doing quite well in terms of classification accuracy:

Algorithm	R^2	Accuracy	Time
Forward Stepwise	0.70668739	94.44444444	108
Sample 1000 [1]	0.60523805	91.66666667	543
Sample 1000 [2]	0.60523805	91.66666667	763
Sample 10000 [2]	0.60523805	91.66666667	6651

1. Forward Stepwise:

```
clas = 22284;
modl = "22284 = 1 : 22283";
optn = [ "print"         2 ,
         "vers"            ,
         "maxst"         3 ,
         "selc"       "r2" ,
         "sing"     1.e-12 ,
         "pobs"            ];
< gof,parm,yptr > = lrforw(affym,modl,optn,clas);
```

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

Step  Effect  DF Rsquared    F_value    pval        SSE        BIC

   1     192   1   0.3509  18.380445  0.0001  3.1581252 -59.236445
   2    2980   1   0.2236  17.338695  0.0002  2.0121794 -72.137486
   3    2301   1   0.1322  14.423831  0.0006  1.1898819 -83.092435


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

```
Final ANOVA Table for Target: 1007_s_at
***************************************

Effect    DF    Rsquared  Sum-of-Squares

 Model     3  0.70668739  6.360186540592
 Error    32           .   2.639813459408
 Total    35           .   9.000000000000

 MSE    0.08249417       SBC -108.395260
 AIC   -86.0611846       BIC -83.0924346
 F      25.6995393       Prb  1.179e-008
 CP     4.00000000       AR2  0.67918934


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

                |    Predicted
     Observed   |        0       1
     ---------  |-----------------
            0   |       17       1
            1   |        1      17


     Total Processing Time: 108
```

2. R^2 Select nsamp= 1000:

```
clas = 22284;
modl = "22284 = 1 : 22283";
optn = [ "print"         2 ,
         "samp"       1000 ,
         "best"         10 ,
         "sing"     1.e-12 ,
         "seed"        123 ,
         "pobs"            ];
< gof,best,parm,yptr > = lrallv(3,affym,modl,optn,clas);


   There are 1821537346665 regression computations with 3
                        coefficients.
  Using 1000 sample (  0.000 \%) out of 1821537346665 possible
                        combinations.
```

```
*************************************
Final ANOVA Table for Target: 1007_s_at
*************************************

Effect    DF    Rsquared  Sum-of-Squares

 Model     3  0.60523805  5.447142453983
 Error    32           .   3.552857546017
 Total    36           .   9.000000000000

 MSE    0.11102680       SBC -97.7016775
 AIC   -75.3676018       BIC -72.3988518
 F      16.3538369       Prb  1.273e-006


   ******************************
   Table of 10 best Rsquare Values
   ******************************

 1     1     2     3   91.67  0.60523805
 2     2   117   643   77.78  0.54853650
 3    32    47  3437   80.56  0.46779610
 4     2   252  4700   80.56  0.44884645
 5    50   951  4687   77.78  0.41455618
 6   389   528   734   83.33  0.41317629
 7    66   356  4742   86.11  0.40900257
 8    69   105  8815   75.00  0.40260926
 9    22   765  1079   83.33  0.40238730
10    32  5176  8506   83.33  0.39988914


     Total Processing Time: 543
```

3. Acc Select nsamp= 1000:

```
  There are 1821537346665 regression computations with 3
                    coefficients.
Using 1000 sample (  0.000 \%) out of 1821537346665 possible
                    combinations.


*************************************
Final ANOVA Table for Target: 1007_s_at
*************************************
```

```
Effect    DF    Rsquared  Sum-of-Squares

 Model     3  0.60523805  5.447142453983
 Error    32           .   3.552857546017
 Total    36           .   9.000000000000

 MSE    0.11102680       SBC -97.7016775
 AIC   -75.3676018       BIC -72.3988518
 F      16.3538369       Prb  1.273e-006


   ******************************
   Table of 10 best Rsquare Values
   ******************************

 1     1     2     3   91.67  0.60523805
 2    66   356  4742   86.11  0.40900257
 3    32   133  4236   86.11  0.38655299
 4   389   528   734   83.33  0.41317629
 5    22   765  1079   83.33  0.40238730
 6    32  5176  8506   83.33  0.39988914
 7    75   233  3173   83.33  0.36161392
 8   153  1257  2058   83.33  0.33785925
 9     8   851  1765   83.33  0.32145251
10   274  2365  3112   83.33  0.24614974


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

                |    Predicted
     Observed   |        0       1
     ---------|-----------------
            0 |       17       1
            1 |        2      16


     Total Processing Time: 763
```

4. Acc Select nsamp= 10000:

```
There are 1821537346665 regression computations with 3
                    coefficients.
```

```
Using 10000 sample (  0.000 \%) out of 1821537346665 possible
                      combinations.


          ************************************
          Final ANOVA Table for Target: 1007_s_at
          ************************************

          Effect    DF    Rsquared  Sum-of-Squares

           Model     3  0.60523805  5.447142453983
           Error    32           .   3.552857546017
           Total    36           .   9.000000000000

           MSE    0.11102680       SBC -97.7016775
           AIC   -75.3676018       BIC -72.3988518
           F      16.3538369       Prb  1.273e-006



             ******************************
             Table of 10 best Rsquare Values
             ******************************

           1     1     2     3   91.67  0.60523805
           2    44    61  7301   88.89  0.53361785
           3    13  1067 14952   88.89  0.41014527
           4    13    99  1578   88.89  0.39073644
           5   160   174  1670   88.89  0.33017588
           6     6   191  6441   86.11  0.54400088
           7    29    81  1893   86.11  0.51879497
           8    44  4148  5534   86.11  0.51817447
           9    24  1963  6654   86.11  0.49065664
          10     2   293 11872   86.11  0.49051318


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

                        |    Predicted
                Observed |        0       1
                ---------|-----------------
                       0 |       17       1
                       1 |        2      16
```

```
Total Processing Time: 6651
```

6.1.8 Breast Cancer (van Veer, 2002) : nobs=98, nvar=24189

```
vanVeer = fil2obj("..\\save\\Veer98_1");
/* make binary target */
resp = vanVeer[,24189];
resp = (resp > 60) ;
vanVeer[,24189] = resp;
vanVeer = cname(vanVeer,cnam);
print "Response", vanVeer[,24189];
```

Algorithm	R^2	Accuracy	Time
Forward Stepwise	0.52074918	84.37500000	133
Sample 1000 [1]	0.44453570	76.04166667	1378
Sample 1000 [2]	0.28134208	79.16666667	1967

1. Forward Stepwise:

```
clas = 24189;
modl = "24189 = 1 : 24188";
optn = [ "print"         2 ,
         "vers"            ,
         "maxst"         3 ,
         "selc"       "r2" ,
         "sing"     1.e-12 ,
         "pobs"            ];
< gof,parm,yptr > = lrforw(vanVeer,modl,optn,clas);


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

 Step  Effect  DF Rsquared    F_value    pval        SSE        BIC

    1   12106   1   0.2557  32.298529  0.0000  6.1269037 -155.52081
    2   10755   1   0.1518  23.818584  0.0000  3.6357178 -175.30373
    3    1438   1   0.1133  21.743225  0.0000  2.7136611 -193.51802


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

```
Effect    DF    Rsquared  Sum-of-Squares

 Model     3  0.52074918  12.47628255187
 Error    92           .   11.48205078146
 Total    95           .   23.95833333333

 MSE    0.12480490       SBC -222.119458
 AIC   -195.862065       BIC -193.518020
 F      33.3220960       Prb  1.137e-014
 CP     4.00000000       AR2  0.50512144
```

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

                |    Predicted
     Observed   |        0       1
     ---------  |-----------------
            0   |       43       7
            1   |        8      38


     Total Processing Time: 133
```

2. R^2 Select nsamp= 1000:

```
clas = 24189;
modl = "24189 = 1 : 24188";
optn = [ "print"         2 ,
         "vers"           ,
         "samp"      1000 ,
         "best"        10 ,
         "sing"     1.e-12 ,
         "seed"        123 ,
         "pobs"           ];
< gof,best,parm,yptr > = lrallv(3,vanVeer,modl,optn,clas);
```

```
   There are 2320165657202 regression computations with 3
                        coefficients.
  Using 1000 sample (  0.000 \%) out of 2320165657202 possible
                        combinations.
```

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

Effect    DF    Rsquared  Sum-of-Squares

 Model     3  0.44453570  10.65033442266
 Error    92           .   13.30799891067
 Total    96           .   23.95833333333

 MSE    0.14465216       SBC -207.951753
 AIC   -181.694360       BIC -179.350314
 F      24.5424017       Prb  9.357e-012


   ******************************
   Table of 10 best Rsquare Values
   ******************************

 1      NM_006544       AL080059 Contig32185_RC   76.04  0.44453570
 2 Contig47405_RC Contig31839_RC       AL137571   73.96  0.32165774
 3 Contig32185_RC         D86957 Contig39285_RC   75.00  0.29948376
 4       AB033007      NM_005176      NM_002093   76.04  0.29067562
 5      NM_016598 Contig49219_RC Contig40286_RC   78.13  0.28339818
 6      NM_000207      NM_020659  Contig8221_RC   70.83  0.28170604
 7      NM_012310      NM_000419 Contig58235_RC   79.17  0.28134208
 8      NM_016017    Contig39387 Contig53635_RC   63.54  0.27571674
 9      NM_016017 Contig26837_RC      NM_004890   72.92  0.27267956
10      NM_004798      NM_006829 Contig47178_RC   73.96  0.26915357


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

                          |    Predicted
                 Observed |        0       1
                 ---------|-----------------
                        0 |       37      13
                        1 |       10      36


                 Total Processing Time: 1378
```

3. Acc Select nsamp= 1000:

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

Effect    DF    Rsquared  Sum-of-Squares

 Model     3  0.28134208  6.740487318952
 Error    92           .   17.21784601438
 Total    96           .   23.95833333333

 MSE    0.18715050       SBC -183.223964
 AIC   -156.966572       BIC -154.622526
 F      12.0054667       Prb  1.055e-006


   ******************************
   Table of 10 best Rsquare Values
   ******************************

 1      NM_012310      NM_000419 Contig58235_RC   79.17  0.28134208
 2      NM_016598 Contig49219_RC Contig40286_RC   78.13  0.28339818
 3      NM_014489      NM_020651 Contig55080_RC   78.13  0.22107000
 4 Contig42933_RC      NM_001084      NM_004708   77.08  0.23762697
 5         U03886   Contig263_RC       AL137557   77.08  0.23338833
 6      NM_006544       AL080059 Contig32185_RC   76.04  0.44453570
 7       AB033007      NM_005176      NM_002093   76.04  0.29067562
 8 Contig42933_RC         D42047      NM_018097   76.04  0.21617426
 9 Contig14780_RC Contig52940_RC      NM_015376   76.04  0.19981946
10 Contig39477_RC      NM_014400       AB033066   76.04  0.19218895


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

             |    Predicted
    Observed |        0       1
    ---------|-----------------
           0 |       42       8
           1 |       12      34


    Total Processing Time: 1967
```