

This file consists of Chapter 11 of **MacAnova User's Guide** by Gary W. Oehlert and Christopher Bingham, issued as Technical Report Number 617, School of Statistics, University of Minnesota, revised August 1998, describing Version 4.07 of MacAnova.

This manual is Copyright © 1998 Gary W. Oehlert and Christopher Bingham, all rights reserved.

Fonts used in this manual are Palatino, Courier, and Symbol.

For information concerning MacAnova, write University of Minnesota, Department of Applied Statistics, 352 Classroom Office Building, 1994 Buford Avenue, St. Paul, MN 55108-6042.



## 11. Summary of Usage

**11.1 Commands, functions and macros** This section contains very brief summaries of usage of most MacAnova commands and pre-defined macros. It corresponds closely to the information you can retrieve by `usage()` (Sec. 2.9) and is primarily for those who are fairly familiar with the commands, but may not remember which keywords or arguments are required.

Items in [...] are optional. In commands with many keywords, they may not all be compatible. Command and macro names are in alphabetical order and are in **bold face** so that they will stand out better. The numbers in [...] are section numbers elsewhere in this manual. They mainly refer to descriptions or explanations of a command or macro, but some refer to examples of its use. There are a few commands that are mentioned nowhere else and have no section number. For details on these, use `help()` (Sec. 2.9 and 8.6) or see the *MacAnova Reference Manual*.

Transformations **abs**(*x*), **acos**(*x*), **asin**(*x*), **atan**(*x*) or **atan**(*x*,*y*), **atanh**(*x*), **ceiling**(*x*), **cos**(*x*), **cosh**(*x*), **exp**(*x*), **floor**(*x*), **lgamma**(*x*), **log**(*x*), **log10**(*x*), **round**(*x*) or **round**(*n*,*ndec*), **sin**(*x*), **sinh**(*x*), **sqrt**(*x*), **tan**(*x*), **tanh**(*x*), *x* REAL or structure with REAL components. Option **angles** (Sec. 8.1.3) determines the units, radians, degrees or cycles, of *x* for **sin**(*x*), **cos**(*x*) and **tan**(*x*) and for the value of **acos**( ), **asin**( ) and **atan**( ) [2.8.6, 2.10.1]

**addchars**([*Graph*,] *x*,*y*,[*c*] [, *impulse*:*T*, *lines*:*T*] [, *other graphics keyword phrases*]) [8.5.3]

**addlines**([*Graph*,] *x*,*y* [, *linetype*:*PosInt*, *thickness*:*PosReal*] [, *impulse*:*T*] [, *other graphics keyword phrases*]) [8.5.3]

**adddatapath**(*dirName* [,*T*]), *dirName* a quoted string or CHARACTER vector specifying one or more additional directory or folder names to search when attempting to read a file [2.11.6]

**addmacrofile**(*fileName* [,*T*]), *fileName* a quoted string or CHARACTER vector specifying one or more additional files to be searched by `getmacros` [7.5.4]

**addpoints**([*Graph*,] *x*,*y* [, *impulse*:*T*, *lines*:*T*] [, *other graphics keyword phrases*]) [8.5.3]

**addstrings**([*Graph*,] *x*,*y*,*charVec*,[, *other graphics keyword phrases*]) [8.5.3]

**alltrue**(*arg1*,*arg2*,...,*argm*), all arguments LOGICAL scalars [2.8.4]

**anova**([*Model*] [, *print*:*F* or *silent*:*T*, *fstats*:*T*, *pvals*:*T*, *coefs*:*F*, *unbalanced*:*T*, *marginal*:*T*]) [3.8, 3.23, 10.5, 10.8 - 10.15]

**anymissing**(*x*), *x* REAL, LOGICAL, or CHARACTER, returns True or False [2.7, 9.4.3]

**anytrue**(*arg1*,*arg2*,...,*argm*), all arguments LOGICAL scalars [2.8.4]

**appendnotes**(*x*,*Notes*), *Notes* a CHARACTER scalar or vector [8.9.1]

**argvalue**(*var*,*argName*, [, *properties*]), *var* any variable, *argName* CHARACTER

**scalar, properties** CHARACTER scalar or vector [9.4.6, 9.4.7]

**array**(x,n1,n2,...[,KeyPhrases]) or **array**(x,dimVec [,KeyPhrases]), x REAL, LOGICAL or CHARACTER, n1, n2, ... **positive integers** or dimVec a vector of **positive integers**, KeyPhrases can be labels:structure(lab1,lab2,...), notes:Notes and/or silent:T, where lab1, lab2, ... and Notes are CHARACTER scalars or vectors. [2.8.15, 8.4.1, 8.9.1]

**asciisave**(FileName [,all:T, v335:T, v406:T, nulls:F, options:F])  
**asciisave**() repeats previous **save**() or **asciisave**() with same options [2.17, 7.7]

**asLong**(x), x REAL with no MISSING values and with integer values between -2147483647 and 2147483647 =  $2^{32}-1$  [9.7.4]

**atan**(x) or **atan**(x,y), x and y REAL or structures with REAL components, y the same size and shape as x; value in radians (default), cycles, or degrees as set by option angles [2.8.6, 2.10.1]

**attachnotes**(x,Notes), Notes a CHARACTER scalar or vector [8.9.1]

**autoreg**(Phi,A [,reverse:T, limits:vector(i1,i2), start:startVals]),  
 REAL vector Phi, REAL vector or matrix A [5.3.3]

**batch**(fileName [,echo:T or F, prompt:string]), CHARACTER scalars fileName and string [7.6]

**bcprd**(x1 [, x2, ... ]), x1, x2, ... REAL matrices with the same number of rows [6.1.1]

**bin**(x,Bnds [,silent:T,leftendin:T]), x a REAL matrix, Bnds REAL vector, Bnds[k] < Bnds[k+1]

**bin**(x,vector(binEdge,binWidth) [,leftendin:T]), binEdge and binWidth > 0 REAL scalars

**bin**(x,nbins, [leftendin:T]), nbins positive integer

**bin**(x [,leftendin:T]) [2.12.19]

**boxcox**(x,power), x a REAL vector or matrix, power a REAL scalar [2.8.6, 10.7]

**boxplot**(x1,x2,...,xk [,vertical:T, graphics keyword phrases]), arguments REAL vectors

**boxplot**(Struc, [, vertical:T, graphics keyword phrases]), Struc a structure with REAL vector components [2.12.2, 10.2]

**cconj**(cx), cx a REAL matrix representing complex data [5.2.4]

**ceiling**(x), x REAL [2.8.6, 2.10.1]

**cellstats**(Term), Term a CHARACTER scalar of form "A.B. ...", where A, B, ... are factors in current GLM model [3.12]

**cft**(cx [,divbyT:T]), cx a REAL matrix representing complex data [5.2.7]

**changestr**(Struc,comp,x), Struc a structure, comp a CHARACTER scalar or positive integer, x a variable or expression

**changestr**(Struc, -n), n a positive integer [9.1.3]

**cholesky**(x), x a positive definite square REAL matrix with no MISSING values [6.6]

**chplot**(x,y [, c] [, lines:T, impulse:T], [graphics keyword phrases]), where x is a REAL vector or scalar, y is a real vector or matrix and c is a integer or CHARACTER scalar, vector, or matrix [2.15.3, 8.5]

**cimag**(cx), cx a REAL matrix representing complex data [5.2.4]

**cluster**(x [, nclust:n, standard:F, method:name, keep:charVec, print:T, tree:T or F, classes:T or F, reorder:T), x a REAL matrix, name a CHARACTER scalar (one of "single", "complete", "average", "ward", "mcquitty", "centroid", or "median"), charVec a CHARACTER vector with elements "all", "classes", "criterion", or "distances"

**cluster**(dissim:d [, ...]), d a square REAL matrix

**cluster**(similar:s [, ...]), s a square REAL matrix [6.7.1]

**complx**(Re,Im), Re and Im REAL matrices with same size and shape.

**complx**(Re) [5.2.4]

**coefs**([Term] [, errorTerm:ErrorTerm, se:T, coefs:F, byterm:F]), Term a CHARACTER scalar, a positive integer, or a factor or variate in the current GLM model, ErrorTerm a CHARACTER scalar or positive integer. Use byterm:F only when Term and coefs:F omitted, and se:T included [3.13, 3.13.1, 10.3, 10.8, 10.11, 10.12]

**colplot**(x [, graphics keyword phrases]), x a REAL matrix [2.15.6, 10.12]

**compnames**(S), S a structure [9.1.2]

y <- **console**() [B.6.6, C.5.6, D.6.6]

**contrast**(Term,Coefs [,Byvar] [,errorTerm:ErrorTerm]), Term a factor in the most recent GLM or a CHARACTER scalar or positive integer specifying a term and ErrorTerm a CHARACTER scalar or positive integer, Coefs REAL, Byvar a factor in the most recent GLM or a CHARACTER scalar specifying such a factor [3.16, 3.16.1, 10.5, 10.8, 10.8.2, 10.10, 10.11, 10.12, 10.15]

**convolve**(wts, x [, reverse:T, decimate:n]), wts a REAL vector, x a REAL vector or matrix, n a positive integer [5.2.8]

**cor**(x1 [,x2,...]), x1,x2,... REAL vectors or matrices all with the same number of rows [2.12.5]

**cpolar**(hx [,unwind:F or crit:val]), hx a REAL matrix representing complex data in polar form, val a REAL scalar,  $0.5 < \text{val} \leq 1$  [5.2.4]

**cprdc**(cx1 [, cx2]), cx1 and cx2 REAL matrices representing complex data [5.2.4]

**cprdcj**(cx1 [, cx2]), cx1 and cx2 REAL matrices representing complex data [5.2.4]

**creal**(cx), cx a REAL matrix representing complex data [5.2.4]

**crect**(cx), cx a REAL matrix representing complex data [5.2.4]

**ctoh**(cx), cx a REAL matrix representing complex data [5.2.4]

**cumbeta**(x,alpha,beta[,lam]), x, alpha, beta, and lam REAL, elements of alpha,

**beta** > 0, lam = 0 [2.12.7]

**cumbin**(x,N,P), x, N and P REAL, elements of N positive integers, elements of P between 0 and 1 [2.12.7]

**cumchi**(x,df), x and df REAL, elements of df positive.

**cumchi**(x,df,lam), same x, df, lam REAL with 0 < lam[i] < 1419.5654 [2.12.7]

**cumF**(x,df1,df2 [,lam]), x, df1, df2 and lam REAL, elements of df1 and df2 positive and lam = 0 [2.12.7]

**cumdunnett**(x,ngps,df [,groupsizes][,onesided:T,epsilon:smallno]), x REAL, elements of ngps, df positive integers, groupsizes vector or array of positive integers, smallno positive number < .01 (default = .00001) [2.12.7]

**cumgamma**(x,alpha), x and alpha REAL, elements of alpha positive [2.12.7]

**cumnor**(x), x REAL [2.12.7]

**cumpoi**(x,mu), x and mu REAL, elements of mu > 0 [2.12.7]

**cumstu**(x,df), x and df REAL, elements of df > 0

**cumstu**(x,df,delta), x, df > 0, delta REAL [2.12.7]

**cumstudrng**(x, ngroup, errorDf [,epsilon:smallno]) where x is REAL, elements of ngroup integers >= 2, elements of errorDf >= 1, smallno > 0 [2.12.7]

**delete**(var1[,var2, ...] [,all:T,real:T or F,char:T or F,logical:T or F,structure:T or F,macro:T or F, graph:T or F, return:T]), F's used only with all:T [2.8.9, 9.3.6]

**describe**(data [,all:T,n:T or F,min:T or F,max:T or F,q1:T or F,q2:T or F,median:T or F,mean:T or F,var:T or F,stddev:T or F,m3:T or F,m3:T or F,m4:T or F,g1:T or F,g2:T or F]), where data is REAL or a structure with REAL components; F's should be used only with all:T [2.12.1, 10.2]

**det**(x [,mantexp:T]), where x is a REAL square matrix with no MISSING values [2.10.6]

**diag**(A), A a matrix [2.10.6]

**dim**(x) [2.8.15]

**dmat**(n,val), n positive integer, val a REAL, CHARACTER or LOGICAL scalar

**dmat**(vec), vec a REAL, CHARACTER or LOGICAL vector [2.10.6]

**edit**(obj [, T]), obj a macro or a REAL variable; Unix and extended mode DOS only

**edit**(0)

**edit**() [C.5.3]

**eigen**(x), x a REAL symmetric matrix with no MISSING values [6.2.1]

**eigenvals**(x), x a REAL symmetric matrix with no MISSING values [6.2.1]

x <- **enter**(val1 val2 val3 ...), valI a number or ?, no separating commas

needed [2.8.10]

**x** <- **enterchars**(str1 str2 str3 ...), str1 a non-quoted sequence of visible characters, separated by spaces [2.8.10]

**error**(a, b, ...[,format:Fmt or nsig:m,header:F,labels:F,missing:missStr]), Fmt and missStr CHARACTER scalars, m > 0 integer [9.4.3]

**evaluate**(cmds), cmds a quoted string or CHARACTER scalar whose value is a MacAnova expression [9.5.1]

**factor**(n1 [, n2, ...]), where n1, n2, ... are REAL scalars or vectors, all of whose elements are positive integers [3.3, 3.8, 10.5]

**fastanova**([Model] [,print:F or silent:T,fstats:T,pvals:T]) [3.19]

**floor**(x), x REAL [2.8.6, 2.10.1]

**fromclip**()

**fromclip**(ncol), ncol a positive integer [7.3]

**y** <- **getdata**(setName), where setName is the unquoted name of a data set on file specified by variable DATAFILE [2.11.4]

**getascii**(charVec1 [, charVec2 ...]), all arguments CHARACTER vectors [7.3.2]

**gethistory**(n) where n > 0 is an integer

**gethistory**() [8.8.3]

**getlabels**(x [,silent:T]) or **getlabels**(x, dims [,silent:T]), dims a vector of positive integers [8.4.2]

**getmacros**(name1 [,name2 ...]), name1, name2 ... unquoted macro names on one of files in CHARACTER vector MACROFILES [7.5.3]

**getnotes**(x [,silent:T]), x REAL, LOGICAL, CHARACTER or a macro. structure or GRAPH variable [8.9.1]

**getoptions**(option1:T [,option2:T ...]), option1, option2, ... option names. Legal option names are angles, batchecho, dumbplot, errors, format, fstats, height, inline, labelabove, labelstyle, maxwhile, missing, nsig, pvals, prompt, restoredel, seeds, update, warnings, wformat, and width plus font and fontsize on Macintosh, scrollback on Macintosh, Windows and Motif, and history and savehistory on Macintosh, Windows, Motif, the extended memory DOS version and most Unix versions.

**getoptions**() or **getoptions**(all:T) gets all option values as structure [8.1.1]

**getseeds**([quiet:T]) [2.13.1]

**gettime**(), **gettime**(quiet:T), or **gettime**(keep:T [,quiet:F])

**gettime**(interval:T), **gettime**(interval:T, quiet:T), or

**gettime**(interval:T, keep:T [,quiet:F]) [9.4.3]

**glmfit**([Model] [,dist:distName,link:linkName, n:denom, incr:T, print:F or silent:T, maxiter:m, epsilon:eps, problimit:small, coefs:F, offset:OffVec, scale:sigma]), distName and linkName CHARACTER scalars,

**denom** > 0 REAL scalar or vector, integer **m** > 0, REAL **eps** > 0,  $10^{-15}$  < **small** .0001, REAL vector **OffVec** [4.2.6]

**glmprcd**(**variates**,**factors** [, **estimate**:F, **seest**:F, **sepred**:T, **n**:N]), **variates** and **factors** REAL vectors or matrices or NULL, **N** a positive scalar or REAL vector with positive elements [3.18]

**glmtable**([**wtdmeans**:T or **x**:vals, **estimate**:F, **seest**:F, **sepred**:T, **n**:N])  
**glmtable**(**Term**,[**wtdmeans**:T or **x**:vals, **estimate**:F, **seest**:F, **sepred**:T, **n**:N]) where **vals** is REAL vector and **TERM** is CHARACTER scalar of form "A.B. ...", where **A**, **B** are factors in current GLM model, **N** is a positive REAL scalar, vector or array of positive numbers [3.18]

**grade**(**x** [, **down**:T]), **x** REAL or CHARACTER or a structure with all REAL or all CHARACTER components [2.12.3]

**halfnorm**(**x** [, **ties**: "ignore" or "average" or "minimum" ]), **x** REAL or a structure with REAL components [2.12.3]

**haslabels**(**x**), **x** a variable that is not NULL [8.4.2]

**hasnotes**(**x**), **x** a variable that is not NULL [8.9.1]

**hconcat**(**a**,**b**,**c**, ...) where **a**, **b**, **c**, ... are matrices with same number of rows [2.10.6]

**hconj**(**hx**), **hx** a REAL matrix representing complex data with Hermitian symmetry [5.2.4]

**help**([**Topic1**,**Topic2**,...] [, **file**:FileName] [, **scrollback**:T])  
**help**([**Topic1**,**Topic2**,...] [, **orig**:T or **alt**:T] [, **scrollback**:T])  
**help**(**Pattern**) where **Pattern** may contain "wild card" characters "\*" and "?"  
**help**(**key**:**KeyNames**), where **KeyNames** is a CHARACTER vector or "?"  
**help**(**news**), **help**(**news**:**yymmdd1**) or **help**(**news**:**vector**(**yymmdd1**,**yymmdd2**)), where **yymmdd1** and **yymmdd2** are integers like 970903 (September 3, 1997) [2.9.1, 2.9.2, 8.6]

**hft**(**hx** [, **divby**:T:T]), **hx** a REAL matrix representing complex data with Hermitian symmetry [5.2.7]

**himag**(**hx**), **hx** a REAL matrix representing complex data with Hermitian symmetry [5.2.4]

**hist**(**x** [, **nbars**] [, **graphics** keyword phrases]), **x** a REAL vector, **nbars** 2 an integer

**hist**(**x**, **boundaries** [, **graphics** keyword phrases]), **boundaries** a REAL vector with increasing elements [2.12.2]

**hpolar**(**hx** [, **unwind**:F or **crit**:val]), **hx** a REAL matrix representing complex data with Hermitian symmetry, **val** a REAL scalar,  $0.5 < \text{val} \leq 1$ ; value affected by option **angles** [5.2.4]

**hprdh**(**hx1** [, **hx2**]), **hx1** and **hx2** REAL matrices representing complex data with Hermitian symmetry [5.2.4]

**hprdhj**(**hx1** [, **hx2**]), **hx1** and **hx2** REAL matrices representing complex data with Hermitian symmetry [5.2.4]

**hreal** (hx), hx a REAL matrix representing complex data with Hermitian symmetry [5.2.4]

**hrect** (hx), hx a REAL matrix representing complex data with Hermitian symmetry in polar form; interpretation of hx affected by option angles [5.2.4]

**htoc** (hx), hx a REAL matrix representing complex data with Hermitian symmetry [5.2.4]

**hypot** (x,y), x and y REAL of the same size and shape, or structures with matching REAL components [2.8.6, 2.10.1]

**inforead** (FileName, Name [, quiet:T, echo:T or F, silent:T, notfoundok:T, prompt:F]), FileName and Name CHARACTER scalars; FileName can also be CONSOLE or have the form string:charVal where charVal is a CHARACTER scalar or vector. [2.11.5]

**invbeta** (P, alpha, beta), P, alpha and beta REAL, elements of P between 0 and 1, those of alpha and beta > 0 [2.12.8, 2.13.2]

**invchi** (P, df [, noncen, epsilon:smallno]), P, df and noncen REAL, elements of P between 0 and 1, elements df > 0, elements of noncen 0, smallno > 0 [2.12.8]

**invdunnett** (P, ngps, df [, groupsizes[, onesided:T, epsilon:smallno]), elements of P between 0 and 1, elements of ngps, df positive integers, groupsizes vector or array of positive integers, smallno positive number < .01 (default = .00001) [2.12.8, 10.8.1]

**invF** (P, df1, df2), P, df1 and df2 REAL, elements of P between 0 and 1, those of df1 and df2 > 0 [2.12.8, 2.13.2]

**invgamma** (P, alpha), P and alpha REAL, elements of P between 0 and 1, those of alpha > 0 [2.12.8, 2.13.2]

**invnor** (P), P REAL with elements of P between 0 and 1 [2.12.8]

**invstu** (P, df), P and df REAL, elements of P between 0 and 1, elements df > 0 [2.12.8]

**invstudrng** (P, ngroup, errorDf [, epsilon:smallno]), elements of P between 0 and 1, elements of ngroup integers 2, elements of errorDf 1, smallno > 0 [2.12.8, 10.8.1]

**ipf** ([Model] [, print:F or silent:T, incr:T, pvals:T, maxit:m, epsilon:eps]), vec a REAL vector, m a positive integer, eps REAL > 0 [5.2.5, 10.19]

**isarray** (arg1 [, arg2, ...] [, real:T, logic:T, char:T]) [9.4.2]

**ischar** (arg1 [, arg2, ...]) [9.4.2]

**isdefined** (arg1 [, arg2, ...]) [9.4.2]

**isfactor** (arg1 [, arg2, ...]) [9.4.2]

**isgraph** (arg1 [, arg2, ...]) [9.4.2]

**islogic** (arg1 [, arg2, ...]) [9.4.2]



**ismacro**(arg1 [, arg2, ...]) [9.4.2]

**ismatrix**(arg1 [, arg2, ...] [,real:T, logic:T, char:T]) [9.4.2]

**ismissing**(x) where x is REAL, LOGICAL or CHARACTER or a structure all of whose components are REAL, LOGICAL, or CHARACTER [2.7]

**isnull**(arg1 [, arg2, ...]) [9.4.2]

**isreal**(arg1 [, arg2, ...]) [9.4.2]

**isscalar**(arg1 [,arg2, ...] [,real:T, logic:T, char:T]) [9.4.2]

**isstruc**(arg1 [, arg2, ... ]) [9.4.2]

**isvector**(arg1 [,arg2, ...] [,real:T, logic:T, char:T]) [9.4.2]

**keyvalue**(keyname1:value1, [keyname2:value2, ...] keyName, type), keyName, type CHARACTER scalars; keyName may contain "wild card" characters "\*" and "?"

**keyvalue**(str, keyName, type), str a structure

**keyvalue**(, keyName, type) [9.4.4]

**kmeans**(y [,means or classes] [,kmax:k1,kmin:k2, start:method, standard:F, weights:wts, quiet:T]), y a REAL matrix, means a REAL matrix with ncols(y) columns, classes a REAL vector with nrows(y) rows, k1 and k2 positive integers, k1 k2, method one of "random", "optimal", "means", or "classes", wts a REAL vector with nrows(wts) = nrows(y) [6.7.2]

**length**(x), x a vector, matrix, array or structure [2.8.13]

**lineplot**(x,y [,linetype:m,impulse:T] [,other graphics keyword phrases]), where x is a REAL vector or scalar, y is a real vector or matrix, and m 0 is an integer [2.15.2, 8.5]

**list**([invis:T]) or **list**(var1 [, var2, ...])

**list**(pattern) where pattern is a quoted string (not a variable) which contains "wild card" characters "\*" and/or "?".

**list**([all:T, real:T or F, char:T or F, logic:T or F, macro:T or F, struct:T or F, null:T or F, keep:T, nrows:n1, ncols:n2, ndims:n3]), use F's only with all:T, n1, n2, n3 positive integers [2.8.9]

**listbrief**([invis:T]) or **listbrief**(var1 [, var2, ...])

**listbrief**(pattern) where pattern is a quoted string (not a variable) which contains "wild card" characters "\*" and/or "?".

**listbrief**([all:T, real:T or F, char:T or F, logic:T or F, macro:T or F, struct:T or F, null:T or F, keep:T, nrows:n1, ncols:n2, ndims:n3]), n1, n2, n3 positive integers; use F's only with all:T [2.8.9]

**loadUser**(fileName [,reload:T or clear:T]), fileName a CHARACTER scalar specifying a file containing one or more user functions to be dynamically loaded [9.7.1]

**logistic**([Model],n:Denom [, incr:T, offset:vec, print:F or silent:T, pvals:T, maxit:m, epsilon:eps, problimit:small, coefs:F]), Denom > 0 a REAL scalar or vector, vec a REAL vector, m a positive integer, eps > 0,  $10^{-15} < \text{small}$

.0001 [4.2.4, 10.18]

**macro**(text [, dollars:T, inline:T or F ,notes:Notes), text a CHARACTER scalar, Notes a CHARACTER scalar or vector [9.3.1]

a <- **macroread**(FileName,macroName [,quiet:T, echo:T or F, silent:T, notfoundok:T, prompt:F]), FileName and macroName CHARACTER scalars; FileName can also be CONSOLE or have the form string:charVal where charVal is a CHARACTER scalar or vector [7.5.1]

**macrou sage**(Macro1 [,Macro2, ...]), Macro1, Macro2, ... , currently defined macros [2.9.3]

**macrowrite**(fileName,a,b,... [,name:Name,header:F,comments:charVec]), a, b,... macros, fileName and Name CHARACTER scalars, charVec a CHARACTER vector or scalar [7.5.2]

**makecols**(x,var1,var2, ...), where x is a REAL matrix, var1, var2, ... unquoted variable names

**makecols**(x,vector("var1","var2", ... )) [2.11.3]

**makefactor**(vec), vec a REAL or CHARACTER vector [3.3]

**manova**([Model] [,print:F or silent:T, coefs:F, pvals:T, fstats:T, byvar:T, sssp:F or T]), Model a CHARACTER scalar [3.22, 3.23, 10.16]

**match**(x,vec [,nomatch]), x REAL or CHARACTER, vec a vector of the same type as x and nomatch a REAL scalar

**match**(pattern,vec [,nomatch],exact:F), pattern REAL or CHARACTER containing "wild card" characters "\*" and/or "?" [9.4.1]

**matprint**(fileName, a, b, ... [, format:Fmt, nsig:n, sep:sepChar, quoted:T or bylines:T,missing:mVal, name:Name, comments:charVec, width:w, header:F]), a,b,... arbitrary variables, Fmt, sepChar and Name CHARACTER scalars with sepChar only a single character, charVec a CHARACTER vector or scalar, mVal a REAL scalar, w 30 integer [7.4.2]

y <- **matread**(FileName,setName [,quiet:T, echo:T or F, labels:Labels, silent:T, notfoundok:T, prompt:F]), FileName and setName CHARACTER scalars; FileName can also be CONSOLE or have the form string:charVal where charVal is a CHARACTER scalar or vector [2.11.3, 7.1, 7.2, 8.4.1]

**matrix**(x,Rowdim [,KeyPhrases]), x a vector, Rowdim > 0 an integer dividing length(x)

**matrix**(x [,KeyPhrases]), x a generalized matrix.

KeyPhrases can be labels:structure(rowLabs,colLabs), notes:Notes and/or silent:T, where rowLabs, colLabs and Notes are CHARACTER scalars or vectors [2.8.13, 8.4.1 , 8.9.1]

**matwrite**(fileName, a, b, ... [, format:Fmt, nsig:n, sep:sepChar, quoted:T or bylines:T,missing:mVal, name:Name, comments:charVec, width:w, header:F]), a,b,... arbitrary variables, Fmt, sepChar and Name CHARACTER scalars with sepChar only a single character, charVec a CHARACTER vector or scalar, mVal a REAL scalar, w 30 integer [7.4.2]

**max**(x), x REAL or LOGICAL or a structure with REAL or LOGICAL components.

**max**(x1, x2, ...), x1, x2, ... REAL or LOGICAL vectors, all the same type [2.12.4]

**min**(x), x REAL or LOGICAL or a structure with REAL or LOGICAL components.

**min**(x1, x2, ...), x1, x2, ... REAL or LOGICAL vectors, all the same type [2.12.4]

**modelinfo**([all:T] keyword1:T or F, keyword2:T or F ... [,nomodelok:T])  
where keywords are one or more of strmodel, termnames, xvars, y, weights, coefs, xtxinv, parameters, scale, colcount, aliased, bitmodel, link, distrib, sigmahat [3.24.4, 3.24.5]

**modelvars**(varList [,Model]), varList a vector of integers 0, Model a CHARACTER scalar

**modelvars**(y:T or x:T or variates:T or factors:T or all:T [, Model])

**modelvars**(nx:T or nvariates:T or nfactors:T or hasconst:T [, Model])  
[3.24.1]

**more**(x [, nsig:n, format:Fmt, missing:M]), where x is a macro or is a REAL, CHARACTER, or LOGICAL variable, n > 0 is an integer, Fmt and M are CHARACTER scalars; macro defined only on Unix versions [D.6.6, F.6.7]

**movavg**(phi,a [,reverse:T, limits:vector(i1,i2), start:startVals]),  
phi REAL vector, a REAL vector or matrix [5.3.2]

**nameof**(var1, var2, ... ) [9.4.3]

**nbits**(x), x consists of 1 or more integers between 0 and 4294967295 [2.8.5]

**ncols**(x), x a matrix or generalized matrix [2.8.13]

**ncomps**(Str), Str a structure [9.1.2]

**ndims**(x) [2.8.15]

**nrows**(x), x a matrix or generalized matrix [2.8.10, 2.8.13]

**outer**(x1, x2, ...), x1, x2, ... REAL [3.16, 10.12]

**padto**(x,n), x a REAL vector or matrix, n > 0 an integer [5.2.5]

**partacf**(vec [, inverse:T]), vec a REAL vector [5.3.4]

**paste**(arg1, arg2, ... [,format:Fmt,sep:C,intwidth:Iw,charwidth:Cw,missing:S]), Fmt, C, and S CHARACTER scalars, Iw and Cw positive integers

**paste**(arg,multiline:T [,format:Fmt,sep:Cs,linesep:C1,missing:S]),  
where Cs and C1 are CHARACTER scalars consisting of a single character [8.3.1-8.3.3]

**plot**(x,y [,impulse:T, lines:T] [,other graphics keyword phrases]), where x is a REAL vector or scalar, y is a REAL vector or matrix [2.15.1, 8.5]

**poisson**([Model] [, incr:T, offset:vec, print:F or silent:T, pvals:T, maxit:m, epsilon:eps, coefs:F]), vec a REAL vector, m a positive integer, eps > 0 [4.25, 10.19]

**polyroot**(coefs), coefs a REAL matrix [5.3.6]

**power**(noncen, ngrp, alpha, denomDF [, design: "rbd"]), noncen 0,  $0 < \alpha < 1$ , positive integers ngrp and nrep; some or all arguments may be vectors [3.21]

**power2**(noncent2, numDF, alpha, denomDF), noncent2 0,  $0 < \alpha < 1$ , numDF > 0, denomDF > 0; some or all arguments may be vectors [3.21]

**predtable**()

**predtable**(Term), Term a CHARACTER scalar of the form "A.B. ...", where A, B are factors in current GLM model, or term:k, where k is a positive integer [3.18, 10.11, 10.19]

**print**(a, b, ...[, format: Fmt or nsig: m, header: F, labels: F, missing: missStr, name: setName, width: w, height: h] [, file: fileName [, new: T]]), Fmt, missStr, fileName, setName CHARACTER scalars, m > 0, w 30, h 12 integers [2.8.9]

**probit**([Model], n: Denom [, incr: T, offset: vec, print: F or silent: T, pvals: T, maxit: m, epsilon: eps, problimit: small, coefs: F]), Denom > 0 a REAL scalar or vector, vec a REAL vector, m a positive integer, eps > 0,  $10^{-15} < \text{small} .0001$  [4.2.4]

**prod**(x), x REAL or LOGICAL or a structure with REAL or LOGICAL components.

**prod**(x1, x2, ...), x1, x2, ... REAL or LOGICAL vectors, all the same type [2.12.4]

**putascii**(vec [, keep: T]), vec a vector of integers > 0 and 255

**putascii**(vec, file: fileName [, new: T]) [7.4.3, 8.3.4]

**qr**(x [, pivot: T, ronly: T]), x a REAL matrix [6.4]

**quit** or **bye** or **end** or **exit** or **stop**

**quit**(F) or **bye**(F) or **end**(F) or **exit**(F) or **stop**(F) quits unconditionally [2.2]

**rank**(x [, down: T, ties: "ignore" or "average" or "minimum"]), x REAL or CHARACTER or a structure with all REAL or all CHARACTER components [2.12.3]

**rankits**(x [, ties: "ignore" or "average" or "minimum"]), x REAL or a structure with REAL components [2.12.3]

**rational**(x, a, b) or **rational**(x, a) or **rational**(x, , b), x REAL or a structure with REAL components, a and b REAL vectors [2.8.6, 2.10.1]

**rbin**(N, n, p), N positive integer, n scalar or vector of positive integers, p REAL scalar or vector of probabilities [2.13, 2.13.2]

y <- **read**(FileName, setName or macroName [, quiet: T, echo: T or F, labels: Labels, \ silent: T, notfoundok: T, prompt: F]), FileName and setName or macroName CHARACTER scalars; FileName can also be CONSOLE or have the form string: charVal where charVal is a CHARACTER scalar or vector [2.11.3, 7.1, 7.2, 8.4.1]

**readcols**(FileName, name1, name2, ..., namek [, Keyword phrases])

**readcols**(FileName, vector("name1", ..., "namek") [, keyword phrases])

FileName a quoted string or CHARACTER scalar, name1, ... unquoted variable names; fileName can also be string: charVal, charVal a CHARACTER scalar or vector.

Keyword phrases may be quiet: T, echo: T, skip: skipChar, stop: stopChar or go: goGhar [2.11.2]

**redo()** re-executes previous command; cannot be used in two successive command lines or on a line immediately following the use of REDO [8.8.1]

**REDO()** re-executes most recent command re-executed by redo. REDO is a macro created by redo [8.8.1]

**regcoefs**(Model [,pvals:T] [,byvar:F]), Model a CHARACTER scalar  
**regcoefs**([pvals:T] [,byvar:F]) [3.13.1, 10.4, 10.6, 10.18]

**regpred**(vals), vals a REAL vector or matrix [3.18, 10.4]

**regress**([Model] [,print:F or silent:T,pvals:T,coefs:F,marginal:T]) [3.8, 10.3, 10.4]

**regs**(x,y), x and y REAL matrices with the same number of rows

**releigen**(h,e), h and e symmetric REAL matrices with no MISSING values, e positive definite [6.2.3]

**releigenvals**(h,e), h and e symmetric REAL matrices with no MISSING values, e positive definite [6.2.3]

**rename**(var, newName), where newName is an undefined variable

**rep**(x, n) or **rep**(x, repFac), where x is REAL, LOGICAL, or CHARACTER, and n is a positive integer or repFac is a vector of integers 0 [2.8.12, 2.14]

**resid()** or **resid**(Model) [3.17, 10.4]

**restore**(FileName [,delete:F, list:T]) [2.17, 7.7]

**resvsindex**([varNo [,Chars]] [,graphics keyword phrases]), varNo a positive integer ncols(RESIDUALS), Chars a CHARACTER or REAL vector or scalar [3.17]

**resvsrankits**([varNo [,Chars]] [,graphics keyword phrases]), varNo a positive integer ncols(RESIDUALS), Chars a CHARACTER or REAL vector or scalar [3.17, 10.3]

**resvsyhat**([varNo [,Chars]] [,graphics keyword phrases]), varNo a positive integer ncols(RESIDUALS), Chars a CHARACTER or REAL vector or scalar [3.17, 10.3]

**reverse**(x), x a REAL or LOGICAL vector or matrix [2.10.6]

**rft**(rx [,divbyT:T]), rx a REAL matrix [5.2.7]

**rnorm**(n), n a positive integer [2.13]

**robust**([Model] [, trunc:c, print:F or silent:T, maxiter:m, epsilon:eps, fstats:T, pvals:T, marginal:T]), Model a CHARACTER scalar, c and eps positive REAL scalars, m a positive integer [4.3, 10.20]

**rotate**(x, k), x a REAL vector or matrix, k an integer [5.2.5]

**rotation**(loadings [, method:name, reorder:T, verbose:T]), where loadings is REAL matrix and name is CHARACTER scalar [6.8]

**round**(x [, ndec]), x REAL, ndec an integer [2.8.6, 2.10.1]

**rowplot**(x [, graphics keyword phrases]), x a REAL matrix [2.15.6]

**rpoi**(n,lambda), n positive integer, lambda scalar 0 or non-negative vector of length n [2.13, 2.13.2]

**rsolve**(A ,B), A a square REAL matrix, B a REAL matrix, nrow(A) = ncol(B); same as B %/% A [2.10.5]

**run**(first,last,incr) or **run**(first,last) or **run**(last) [2.8.12, 2.14]

**runi**(n), n a positive integer [2.13]

**samplesize**(noncen,ngroup,alpha,Pwr [, design:"rbd", maxn:N]),noncen, alpha, Pwr positive, ngroup, N positive integers [3.21]

**save**(FileName [,all:T, v335:T, v406:T, nulls:F, options:F, ascii:T, history:T])

save() repeats previous save() or asciisave() [2.17, 7.7]

**screen**([Model] [, method:"cp" or "rsq" or "adjrsq", mbest:m, forced:fn, s2:mse, penalty:pen, keep:list]),m positive integer, fn vector of positive integers, mse and pen positive REAL scalars, list CHARACTER scalar or vector with elements "p", "cp", "rsq", "adjrsq", "model" or "all" [3.20]

**secoefs**([Term] [, errorTerm:ErrorTerm, byterm:F, se:F or coefs:F]), Term a CHARACTER scalar, a positive integer, or a factor or variate in the current GLM model, ErrorTerm a CHARACTER scalar or positive integer. byterm:F only when Term, se:F and coefs:F omitted [3.13, 3.13.1, 10.3, 10.10, 10.11, 10.13, 10.15, 10.19]

**select**(k, x), k vector of positive integers or LOGICAL vector, x a matrix [2.10.6]

**sethistory**(lines), lines is a CHARACTER vector with length value of option history [8.8.3]

**setlabels**(x, labels [,silent:T]), x an existing scalar, vector, matrix, array or structure, labels a CHARACTER scalar or vector, a structure with CHARACTER scalar or vector components, or NULL [8.4.1]

**setoptions**(option1:value [,option2:value ... ]) option1,option2,... names of options. Legal option names are angles, batchecho, dumbplot, errors, format, fstats, height, inline, labelabove, labelstyle, maxwhile, missing, nsig, prompt, pvals, restoredel, seeds, update, warnings, wformat, and width plus font and fontsize on Macintosh, scrollbar on Macintosh, Windows and Motif, and history and savehistory on Macintosh, Windows, Motif, the extended memory DOS version and most Unix versions

**setoptions**(str), where str is of the form structure(option1:value, ...) [8.1.2]

**setseeds**(seed1,seed2) or **setseeds**(vector(seed1,seed2)), seed1 and seed2 non-negative integers 2147483399 [2.13.1]

**shell**(command), **shell**(command,keep:T) or **shell**(command,interact:T), command a quoted string or CHARACTER scalar; not implemented on Macintosh !command immediately after the prompt; not implemented on Macintosh [8.7.1, 8.7.2]

**showplot**([Graph] [, graphics keyword phrases]) [8.5.3]

**solve**(A), A a square REAL matrix

**solve**(A, B), A a square REAL matrix, B a REAL matrix, nrow(A) = nrow(B); same as A %\% B [2.10.5]

**sort**(x [,down:T]), x REAL or CHARACTER or a structure with all REAL or all CHARACTER components [2.12.3]

**split**(x,A [,compnames:CharVar]), x REAL, A a factor or vector of integers or LOGICAL vector, CharVar a CHARACTER scalar or vector

**split**(x,bycols:T or byrows:T [,compnames:CharVar]), x a REAL matrix [9.1.1, 10.2, 10.5]

**spool**(FileName [,new:T])

**spool**() toggles spooling on and off [2.16]

**stemleaf**(x [, nstems, outliers:F, depth:F, stats:T, title:"Your title"]), x a REAL vector [2.12.2, 10.2]

**strconcat**(var1 [,var2,...,vark] [, KeyPhrases]), where var1, var2, ... are arbitrary variables; KeyPhrases can be compnames:Charvec, labels:lab, and silent:T, where Charvec and lab are CHARACTER scalars or vectors [8.4.1, 9.1.1]

**structure**(var1 [,var2,...,vark] [, KeyPhrases]), where var1, var2, ... are arbitrary variables; KeyPhrases can be compnames:Charvec, labels:lab, notes:Notes, and silent:T, where Charvec,lab and Notes are CHARACTER scalars or vectors [2.8.16, 8.4.1, 9.1.1]

**sum**(x), x REAL or LOGICAL or a structure with REAL or LOGICAL components.

**sum**(x1, x2, ...), x1, x2, ... REAL or LOGICAL vectors, all the same type [2.12.14]

**svd**(x [,left:T or F,right:T or F, all:T]), x a REAL matrix [6.3]

**swp**(x, n1 [, n2, ...]), x a REAL matrix, n1, n2, ... positive integers or vectors of positive integers [6.1]

**t2int**(x1,x2,Coverage [, pooled:F]), x1 and x2 REAL vectors,  $0 < \text{Coverage} < 1$  [2.12.6]

**t2val**(x1,x2 [,df:T, pooled:F]) [2.12.6]

**tabs**(y,a,b,... [, mean:T, var:T, count:T]), y REAL, a, b, ... factors or vectors of positive integers

**tabs**(,a,b,...) [3.12, 10.5, 10.7]

**tint**(x,Coverage), x a REAL vector,  $0 < \text{Coverage} < 1$  a REAL scalar [2.12.6]

**toclip**(x [, missing:Code, format:Fmt, sep:C1, linesep:C2]), Code, Fmt, C1, C2 quoted strings or CHARACTER scalars

**toeplitz**(x), x a REAL vector [5.3.5]

**trace**(x), x a REAL square matrix [2.10.6]

x' or **t**(x), where x is a matrix [2.10.3]

**trideigen**(Diag, Subdiag [ [, start] , end] , values:F or vectors:F),  
Diag, Subdiag REAL vectors, start and end positive integers [6.2.2]

**trilower**(a), matrix a [6.6]

**triunpack**(vec [, lower:T or upper:T]), vec a vector of length  $p(p+1)/2$  [6.6]

**triupper**(a [,pack:T]), matrix a [6.6]

**tval**(x [,df:T]), REAL vector x [2.12.6]

**twotailt**(tval, df), REAL scalar tval,  $df > 0$  [2.12.6]

**unique**(x [,index:T]), x REAL or CHARACTER [9.4.1]

**unwind**(angleMat [, crit:Val]), angleMat a REAL matrix and .5 Val < 1; units,  
radians, degrees or cycles, of angleMat determined by option angles [5.2.4]

**usage**([Topic1,Topic2,...] [,file:FileName or orig:T or alt:T])  
**usage**(Pattern) where Pattern is a quoted string (not a variable) containing "wild  
card characters "\*" and/or "?"

**usage**(key:KeyNames), where KeyNames is a CHARACTER vector or "?" [2.9, 8.6]

**User**(funName [,resource:resName][,quiet:T],arg1 [,...])  
**User**(funName [,resource:resName][,quiet:T],callback:T,arg1 [,...])  
**User**(funName [,resource:resName][,quiet:T],symbols:T,arg1 [,...])  
**User**(funName [,resource:resName][,quiet:T],pointers:T,arg1 [,...])  
funName and resName CHARACTER scalars; arg1, ... arguments to a dynamically loaded  
user function; if an argument is a keyword phrase other than protect:arg, it is  
returned, possibly modified [9.7.2-9.7.4]

**varnames**(Model) where Model is CHARACTER scalar  
**varnames**() is equivalent to varnames(STRMODEL) [3.24.1]

**vboxplot**(x1,x2,...,xk [, graphics keyword phrases]), arguments REAL vectors  
**vboxplot**(Struc [, graphics keyword phrases]), Struc a structure with REAL  
vector components [2.12.2]

**vconcat**(a,b,c,...), a,b,c, ... matrices with same number of columns [2.10.6]

y <- **vecread**(FileName [ keyword phrases])  
y <- **vecread**(FileName, character:T [ keyword phrases])  
y <- **vecread**(FileName, bylines:T [ keyword phrases])  
Permissible keyword phrases are stop:stopChar, go:goGhar, skip:skipChar,  
silent:T, quiet:F, echo:F or T and prompt:F; FileName can also be CONSOLE or  
have the form string:charVal where charVal is a CHARACTER scalar or vector [2.11.1,  
7.2, 7.3]

**vector**(x1,x2,...,xk [,KeyPhrases]) where x1, x2, ... all have the same type,  
REAL, LOGICAL, or CHARACTER, or are structures with components all of the same type  
KeyPhrases can be labels:lab, notes:Notes and/or silent:T, where lab and  
Notes are CHARACTER scalars or vectors [2.14, 8.4.1, 10.2]



**write**(a, b, ...[,format:Fmt or nsig:m, header:F, labels:F, width:w, height:h, missing:missStr, name:setName] [, file:fileName [,new:T]]), Fmt,missStr,fileName,setName CHARACTER scalars, m > 0, w 30, h 12 integers [2.8.9, 7.4.1]

**xrows**(variates [,factors]), variates and factors REAL matrices or vectors or NULL [3.24.3]

**xvariables**(Model [, missing:val]), Model a CHARACTER scalar [3.24.2]

**yates**(x), x a REAL vector

**yhat**() or **yhat**(Model [,T]) [3.18]

**yulewalker**(vec [, inverse:T]), vec a REAL vector [5.3.4]

**11.2 Operators** This section lists the various unary and binary operators with their precedence. If two operators with different precedence appear in the same expression, the operator with the higher precedence is applied first. Thus, since “\*” has higher precedence than “+”,  $3 + 4 * 5 + 6$  is evaluated as  $3 + (4*5) + 6$  while  $3 * 4 + 5 * 6$  is evaluated as  $(3*4) + (5*6)$ .

**11.2.1 Arithmetic operators** These all operate on REAL or LOGICAL data and have a REAL value. Logical True and False are treated as 1 and 0, respectively. They can operate on structures with REAL or LOGICAL components if the structures are compatible.

Operator	Precedence	Description
a + b	9	Addition: sum of a and b
a - b	9	Subtraction: difference of a and b
a * b	10	Multiplication: product of a and b
a / b	10	Division: a divided by b
a %% b	10	Modular division of a by b
-a	12	Unary minus: negative of a
+a	12	Unary plus: a
a ^ b or a ** b	13	Exponentiation: a to the b-th power

**11.2.2 Comparison operators** These all operate on REAL, LOGICAL or CHARACTER data and have a LOGICAL value. LOGICAL True and False are treated as 1 and 0, respectively. For CHARACTER data, order is alphabetical using the ASCII collating sequence for which all lower case letter come before all upper case letter, for example, "MacAnova" < "Macintosh" and "Residuals" < "rESIDUALS". See Sec. 2.12.3 for a table of CHARACTER ordering.

Operator	Precedence	Description
a == b	8	True if a and b are equal or the same
a != b	8	True if a and b are not equal or the same
a < b	8	True if a < b
a <= b	8	True if a ≤ b
a > b	8	True if a > b
a >= b	8	True if a ≥ b

**11.2.3 Logical operators** These all operate on LOGICAL data and have a LOGICAL value. They can operate on structures with LOGICAL components if the structures are compatible. Both operands of || and && will be evaluated, even if the second is not needed to determine the value. Pre-defined macros anytrue and alltrue can be used instead if it is important not to make unnecessary evaluations of operands.

Operator	Precedence	Description
a    b	5	Logical Or: T    T, T    F, F    T are True, F    F is False
a && b	6	Logical And: T&&T is True, T&&F, F&&T, and F&&F are False )
!a	7	Logical Not: !T is False, !F is True

**11.2.4 Matrix operators** These all operate on REAL matrices and have a REAL matrix value. The transpose operator (a') operates on multidimensional arrays and LOGICAL and CHARACTER data. They can operate on structures with matrix components if the structures are compatible.

Operator	Precedence	Description
a' or t(a)	12	Transpose of a
a %*% b	11	Matrix product of a and b
a %c% b	11	Matrix product of a' and b
a %C% b	11	Matrix product of a and b'
a %/% b	11	Matrix product of a and b <sup>-1</sup>
a %\% b	11	Matrix product of a <sup>-1</sup> and b
solve(a)		Matrix inverse a <sup>-1</sup>

**11.2.5 Bitwise operators** These operate on non-negative integers  $4294967295 = 2^{32}-1$ . Any operand < 0 or > 4294967295 yields a MISSING value. Each bit of the operands has value 0 or 1 and is operated on independently of any other bits. The description column refers to the operation on a single bit. They can operate on structures with REAL components if the structures are compatible

Operator	Precedence	Description
<code>a %  b</code>	1	Bitwise Or: <code>1% 0</code> , <code>0% 1</code> and <code>1% 1</code> are 1, <code>0% 0</code> is 0
<code>a %^ b</code>	2	Bitwise Exclusive Or: <code>1%^0</code> and <code>0%^1</code> are 1, <code>1%^1</code> and <code>0%^0</code> are 0
<code>a %&amp; b</code>	3	Bitwise And: <code>1%&amp;1</code> is 1, <code>1%&amp;0</code> , <code>0%&amp;1</code> and <code>0%&amp;0</code> are 0.
<code>%!a</code>	4	Bitwise Not or Complement: <code>!0</code> is 1 and <code>!1</code> is 0

**11.2.6 Assignment operators** There is one pure assignment operator (`<-`) and five arithmetic assignment operators. The latter operate only on `REAL` and `LOGICAL` variables. The left operand of an arithmetic assignment operator is modified by combining it with the right operand using the indicated operation. Thus `a <-+ b` is equivalent to `a <- a + b`. The left operand cannot be a value defined by a subscript so that, for example, `a[3] <-+ 1` is illegal.

Operator	Precedence	Description
<code>a &lt;- b</code>	0	Assign b to a
<code>a &lt;-+ b</code>	0	<code>a &lt;- a + b</code>
<code>a &lt;-- b</code>	0	<code>a &lt;- a - b</code>
<code>a &lt;-* b</code>	0	<code>a &lt;- a * b</code>
<code>a &lt;- / b</code>	0	<code>a &lt;- a / b</code>
<code>a &lt;-^ b</code> or <code>a &lt;-** b</code>	0	<code>a &lt;- a ^ b</code>

**11.3 Control syntax elements** There are several syntax elements for conditional execution, looping and leaving a loop.

**11.3.1 Syntax elements for conditional execution** These are `if`, `elseif`, `else`. Their usage is as follows:

```

if (Logical1){command1;command2; ...}
if (Logical1){command1;command2; ...}else{...}
if (Logical1){command1;command2; ...}elseif(Logical2){
...}else{...}]

```

See Sec. 9.2.2.

**11.3.2 Syntax elements for looping** These are `for` and `while`. Their usage is as follows:

```

for(i,vec){command1;command2; ...}, vec a REAL vector
for(i,i1,i2){command1;command2;...} like for(i,run(i1,i2)){...}
for(i,i1,i2,d){command1;command2;...} like for(i,run(i1,i2,d)){...}

while(Logical){statement1;statement2;...;}

```

See Sec. 9.2.3.

**11.3.3 Syntax elements for leaving a loop** These are `break` and `breakall`, plus macro `breakif`. Here are example of their usage:

```
for(i,run(n)){if(x[i] < 0){break} .... }
for(i,run(n)){for(j,run(m)){if(x[i,j] < 0){break 2}}}  
for(i,run(n)){for(j,run(m)){if(x[i,j] < 0){breakall}}}  
for(i,run(n)){breakif(x[i] < 0) .... }
for(i,run(n)){for(j,run(m)){breakif(x[i,j] < 0, 2) ...}}
```

```
s <- n <- 1;while(n<100){
    tt <- x^n; s <-+ tt;
    if(abs(tt) <= 1e-10*abs(s)){break}
    n <-+ 1}
```

See Sec. 9.2.4.

**11.3.4 Syntax element for skipping to the end of a loop** Syntax element `next` skips to the end of a loop. Here are examples of its usage:

```
for(i,run(n)){if(x[i] < 0){next} .... }
for(i,run(n)){for(j,run(m)){if(x[i,j] < 0){next 2} ... }
... }#exit j loop and skip to end of i loop
```

See Sec. 9.2.5.

**11.4 Some differences from S-Plus** On the surface, MacAnova appears to be quite similar to statistics program S and its successor S-Plus™, but it differs in many important ways. Here are some of the differences, listed here to make it easier for a S-Plus user to use MacAnova.

- MacAnova variable names may contain “\_” but not “.”.
- MacAnova has temporary variables (names starting with “@”) and allows for the generation of unique variable names in macros. S-Plus organizes things in “frames” so that different functions can use the same names for variables without danger of “collision.”
- MacAnova has fewer types of variables than does S-Plus. There are REAL, CHARACTER and LOGICAL scalars, vectors, matrices and arrays, factors (a special type of REAL vector, see Sec. 3.3), macros (Sec. 2.3, 9.3), GRAPH variables (Sec. 8.5.3) and structures whose components may be any type of variable (Sec. 2.8.16, 9.1). In particular, there is no special time series object and complex data is encoded in REAL variables (Sec. 5.2.3).
- MacAnova variables have dimensions and optional coordinate labels (Sec. 2.8.13, 2.8.15, 8.4) but do not have attributes in the S-Plus sense.
- Missing values are entered as ? and printed as MISSING in MacAnova but entered and printed as NA in S-Plus.
- In MacAnova 'This is a "quoted" string' is illegal; use "This is a \"quoted\" string" instead. “'” is the MacAnova transpose operator.
- Line continuation at the prompt is different between MacAnova and S-Plus. In

MacAnova a line is continued automatically if a quoted string (Sec. 2.5) lacks a terminating " or a compound command starting with "{" lacks a terminating "}". You can force continuation by ending a line with "\". There is a continuation prompt only on non-windowed versions. S-Plus continues a line if any bracketted expression starting with "(" or "[" or "{" is incomplete.

- MacAnova keyword phrases have the form `keyword:value` instead of `keyword=value` and you can't abbreviate most keywords.
- In MacAnova, `a %/% b` is equivalent to `rsolve(b,a)`, computing a solution  $x$  to the equation  $x \%* \% b = a$ ; in S-Plus, `a %/% b` evaluates to `floor(a/b)`.
- In MacAnova, `vector(1,3,5,2)` does what `c(1,3,5,2)` does in S-Plus.
- In MacAnova, `sum(x)`, `prod(x)`, `max(x)`, and `min(x)` operate over the first dimension of  $x$ , returning a variable whose first dimension is 1 and the remaining dimensions are the same as those of  $x$ . In particular, if  $x$  is a matrix, the result is a row vector each of whose elements is derived from one column of  $x$  (Sec. 2.12.4). In S-Plus, the value of these functions is a scalar summarizing all the values in  $x$ .
- MacAnova maintains the equivalence of a vector and a column matrix (matrix with 1 column). It never treats a vector as a row vector. Thus, for example, if  $a$  is a matrix with 3 rows, `vector(1,3,4) \%* \% a` is illegal in MacAnova but `c(1,3,4) \%* \% a` is equivalent to `t(c(1,3,4)) \%* \% a` in S-Plus.
- If  $a$  is a  $m$  by  $n$  matrix, in MacAnova `a[1,]` is a 1 by  $n$  matrix (a row vector) and `a[,1]` is a  $m$  by 1 matrix (a column vector). In S-Plus both are just plain vectors; in particular, contrary to what you might expect, `t(a[1,])` is a 1 by  $n$  matrix.
- MacAnova has quite different rules for binary operations that combine two variables of different sizes. Suppose  $m > 1$  and  $n > 1$ ,  $a$  is a column vector of length  $m$ ,  $b$  is a 1 by  $n$  row vector,  $c$  is a  $m$  by  $n$  matrix, and  $op$  is a binary element-wise operator such as `*`. Then `a op c` combines  $a$  with every column of  $c$ , `b op c` combines  $b$  with every row of  $c$ , and `a op b` is a  $m$  by  $n$  matrix combining every element of  $a$  with every element of  $b$ . See Sec. 2.10.2. S-Plus, when it allows the operation at all, extends a vector cyclically to get a dimension match.
- MacAnova does not have the construct S-Plus `n1:n2`. Use `run(n1,n2)` instead (Sec. 2.8.12 and 2.14).
- MacAnova for loops have the form `for(index,values){...}` (Sec. 9.2.3) instead of `for(index in values){...}`.
- In MacAnova, the statement or statements controlled by `if`, `elseif` and `else` must be a compound command, that is, enclosed in `{...}` (see Sec. 9.2.1), with "{" on the same line as `if`, `elseif` or `else` (Sec. 9.2.2). S-Plus has neither limitation.
- In MacAnova, the statement or statements forming the body of `while` and `for` loops must be a compound command, with "{" on the same line as `while` or `for` (Sec. 9.2.3). S-Plus has neither limitation.
- Although they have some features in common, a MacAnova structure differs from an S-Plus list. MacAnova allows structures to be operands of most operators and arguments of transformations and some functions like `sum()` and `max()`. In

MacAnova, if `str` is a structure, then the value of both `str[2]` and `str[[2]]` is component 2 of `str` and is a structure only if the component itself is a structure. In S-Plus, if `lis` is a list, `lis[2]` is a list whose only component is component 2 of `lis`, while `lis[[2]]` is the component itself. See Sec. 2.8.16 and 9.1.

- Linear and generalized linear model analysis is done quite differently in MacAnova and S-Plus. For example, models are specified in different ways. A MacAnova model is a CHARACTER scalar of the form "depvar = term1 + term2 + ...", with typical terms being `x1`, `age` or `x1.age`. A term of the form `a.b` may express an interaction (when both `a` and `b` have previously appeared in the model), a nesting of `b` within `a` (when `a` but not `b` has previously appeared), or a multidimensional factor (when neither `a` or `b` has previously appeared). See Sec. 3.4. In S-Plus, a model is specified not by a CHARACTER variable but by a "formula" in which "~" separates the dependent variable from the right hand side of the model, `a:b` specifies an interaction and `b%in%a` expresses nesting of `b` in `a`. Also MacAnova codes categorical data as "factors" (Sec. 3.3) and S-Plus uses "category" variables.
- MacAnova has very different functions for reading and writing files. When reading unstructured numerical data separated by white space from file `data.txt`,

```
x <- vecread("data.txt")
```

in MacAnova has the same result as

```
x <- scan("data.txt")
```

in S-Plus; `vecread()` and `scan()` differ in more complex situations. MacAnova function `matread()` reads matrices and arrays retrieving dimension information, but requires that the file be in a special format. It has some similarity with S-Plus function `read.table()`. If `x` is a matrix, to obtain roughly the same effect as

```
write.table(x, file="data.txt")
```

in S-Plus, you can use

```
matprint("data.txt",x,sep=" ",nsig:17,new:T)
```

in MacAnova.

- Graphing is done differently since MacAnova has no concept of graphics device. In its simplest usage, MacAnova function `plot()` roughly corresponds to S-Plus function `plot()`. Other basic MacAnova plotting commands are `lineplot()` and `chplot()`. All recognize keyword phrase `add:T`, indicating new information is to be added to the previous plot. You can specify that a plot will be low resolution by keyword phrase `dumb:T` and that it should be written as PostScript to file `myplot.ps`, say, by keyword phrase `file:"myplot.ps"`. By default, a GRAPH variable LASTPLOT encapsulating the entire graph is produced by all MacAnova plotting commands. See Sec. 2.15 and 8.6.
- In addition to the above differences, many MacAnova functions, such as `paste()`, `cat()`, `diag()` and `structure()`, have the same name as an S-Plus function but differ in minor and sometimes major ways.
- All MacAnova variables are kept in memory rather than disk. Most S-Plus variables correspond to disk files.