```
########################################################################
                        **********
                        * anova1 *
                        **********
Apply ANOVA test to data based on a single categorical (factor) variable.

anova1(x,fac,sort=F,progress=F)

x           dependent variable (to be tested for dependence)
fac         factor on which the dependent variable may depend
            Note: if the variable is numeric, then it is
            converted to catagorical (a factor) for the test
sort        if TRUE, then the output is sorted into descending
            order of the number of occurences of each factor
progress    if TRUE, a graph is displayed showing the progress
            of the computation

Example:

anova1(mag,floor(t/10))         # test for changes in "mag" dependent on
                                # floor(t/10) as a categorical (rather
                                # than numerical) variable.  This is
                                # equivalent to using 10-day wide bins
anova1(mag,obs)                 # test for changes in "mag" dependent on
                                # the variable "obs" -- which might be
                                # observer I.D.  In this case it's OK
                                # if "obs" is a string variable -- since
                                # the anova1 routine converts the
                                # variable to a factor anyway, numeric
                                # variables are treated the same as
                                # non-numerics
########################################################################
                        **********
                        * aovper *
                        **********
Compute the AoV periodogram.

aovper(t,x,lowfreq=0,hifreq=0,bins=4,resmag=1,plot=T,outfile=NULL)

t           vector of times of observation
x           vector of data values
lowfreq     lowest frequency to test
hifreq      highest frequency to test
bins        number of bins to use in AoV computation
resmag      resolution magnification.  The number of frequencies
            tested for a given frequency range is multiplied by
            this parameter
plot        if TRUE, the periodogram is plotted
outfile     if a string variable rather than null, the results
            are written to a file of this name

Examples:

aovper(t,x)                     # compute the AoV periodogram
                                # for x as a function of t
aovper(t,x,0,.02)               # compute the AoV periodogram
                                # for frequencies from 0 to 0.02
aovper(t,x,bins=12)             # compute the AoV periodogram using
                                # 12 bins rather than the default 4
```

```
aovper(t,x,0,.01,resmag=10)     # compute the AoV periodogram for
                                # frequencies from 0 to 0.01, scanning
                                # frequency space at 10 times higher
                                # than the usual resolution (i.e.,
                                # oversampling by a factor of 40
                                # rather than the default factor of 4
########################################################################
                        *********
                        * dcdft *
                        *********
Compute the DCDFT periodogram.

dcdft(t,x,lowfreq=0,hifreq=0,resmag=1,plot=T,outfile=NULL)

t           vector of times of observation
x           vector of observed data
lowfreq     lowest frequency to test
hifreq      highest frequency to test
resmag      resolution magnification.  The number of frequencies
            tested for a given frequency range is multiplied by
            this parameter
plot        if TRUE, the periodogram is plotted
outfile     if a string variable rather than null, the results
            are written to a file of this name

Examples:

dcdft(t,x)                      # compute the DCDFT periodogram
                                # for x as a function of t
dcdft(t,x,0,.02)                # compute the DCDFT periodogram
                                # for frequencies from 0 to 0.02
dcd = dcdft(t,x,plot=F)         # compute the DCDFT and assign
                                # the output to "dcd"
                                # but don't plot the periodogram
dcdft(t,x,0,.01,resmag=10)      # compute the DCDFT periodogram for
                                # frequencies from 0 to 0.01, scanning
                                # frequency space at 10 times higher
                                # than the usual resolution (i.e.,
                                # oversampling by a factor of 40
                                # rather than the default factor of 4
########################################################################
                        *************
                        * findstart *
                        *************
Find starting index for a given value.

findstart(X,start)

X           vector of values to be scanned.  Must be sorted
            into ascending order
start       starting value to be located

Examples:

n1 = findstart(t,40000)         # find the index for the first
                                # t value which is at least 40000
########################################################
# restrict data for t,x to the time range 40000 to 45000
########################################################
```

```
n1 = findstart(t,40000)         # 1st value to include
n2 = findstart(t,45000) - 1     # last value to include
t = t[n1:n2]                    # keep only t values from n1 to n2
x = x[n1:n2]                    # keep only x values from n1 to n2
########################################################################
                        **********
                        * foldit *
                        **********
Compute and plot a folded light curve.

foldit(t,x,period,epoch=0,plot=T,bin=.05)

t           vector of times
x           vector of data values (magnitudes)
period      period with which to fold the data
epoch       epoch to use when folding the data.  Default is zero.
plot        if TRUE, produce a plot of the folded light curve
bin         size (in phase) of bins to use for averaging.
            default is bins of width 0.05 (i.e., 20 bins per cycle)

Examples:

foldit(t,x,332)                 # fold time series (t,x) using
                                # a period of 332
folded = foldit(t,x,per)        # fold the time series (t,x) using
                                # period "per" and assign the result
                                # to an object called "folded"
names(folded)                   # show the names of the object "folded"
########################################################################
                        ***********
                        * fourfit *
                        ***********
Fit a Fourier series to data.

fourfit(t,x,p)

t           vector of times
x           vector of data values
p           period to fit to the data

Examples:

xfit = fourfit(t,x,332)         # fit a Fourier series to (t,x)
                                # using the single period 332
pers = c(332,166,110.667,83)    # create a vector "pers" consisting
                                # of the numbers 332, 166, 110.667, and 83
xfit = fourfit(t,x,pers)        # fit a Fourier series to (t,x)
                                # using periods 332, 166, 110.667, and 83
names(xfit)                     # show the names of the object "xfit"
########################################################################
                        **********
                        * movave *
                        **********
Compute moving averages.

movave(X,L=12)

X           quantity for which to compute a moving average
L           number of points to include in each moving average
```

```
Examples:

tave = movave(t,30)             # compute a 30-point moving average of t
                                # and assign it to a variable "tave"
xave = movave(x,30)             # compute a 30-point moving average of x
                                # and assign it to a variable "xave"
plot(tave,xave,type="l")        # make a line plot of xave vs tave
########################################################################
                        *********
                        * peak1 *
                        *********
Find the strongest peak in a periodogram.

peak1(dcd,t=NULL,x=NULL)

dcd         output from the "dcdft" or "aovper" scripts
t, x        times and data values used to create the "dcd" object.
            If these are given, and the periodogram is a DCDFT
            rather than AoV periodogram, then standard errors are
            computed for the peak frequency and (semi-)amplitude

Examples:

dcd = dcdft(t,x,0,.01)          # compute the DCDFT of (t,x) for
                                # frequencies from 0 to 0.01, and
                                # assign it to an object called "dcd"
peak1(dcd)                      # find the strongest peak in the
                                # object "dcd" and the upper and lower
                                # FWHM limits for frequency and period
pk1 = peak1(dcd,t,x)            # find the strongest peak in the
                                # object "dcd" and the upper and lower
                                # FWHM limits for frequency and period
                                # as well as the standard errors for
                                # period and amplitude, and assign the
                                # result to an object called "pk1"
########################################################################
                        *********
                        * peaks *
                        *********
Find all peaks in a periodogram.

peaks(dcd,lofre=0,hifre=0,maxpeak=0,plot=T)

dcd         output from the "dcdft" or "aovper" scripts
lofre       lowest frequency to include
hifre       highest frequency to include
maxpeak     maximum number of peaks to report.  If specified,
            only this many strongest peaks will be returned
plot        if TRUE, a plot is made of the observed chi-square
            values of the peaks vs the theoretical chi-square
            peak values (based on treating peaks as chi-square
            with 3 degrees of freedom)

Examples:

dcd = dcdft(t,x,0,.01)          # compute the DCDFT of (t,x) for
                                # frequencies from 0 to 0.01, and
                                # assign it to an object called "dcd"
```

```
peaks(dcd)                      # find all the peaks in the periodogram
                                # saved as "dcd"
pp = peaks(dcd)                 # find all the peaks in the periodogram
                                # saved as "dcd" and store them in an
                                # object called "pp"
########################################################################
                        **********
                        * timave *
                        **********
Average data over bins of a fixed time width.

timave(t,x,t2ave=-1,t.off=0,n.min=2,wide=2,plot=T,lines=T,tit=NULL,big=F)

t           vector of times of observations
x           vector of data values
t2ave       width of time spans over which to average.  If not
            specified, the width defaults to 1 time unit
t.off       time offset.  Normally, time bins start at zero, but
            if t.off is specified they start at t.off
n.min       minimum number of data points required for a given bin
wide        width of error bars (in units of the standard error)
plot        if TRUE, a plot of the averages, with error bars, is created
lines       if TRUE, the plot is a points-and-lines plot; otherwise
            only points are plotted
tit         title to be used for the plot
big         if TRUE, then on the plot those values that are
            significantly different from the overall mean are
            circled in red, those which are more than 2.5 standard
            deviations from the overall mean (whether significant
            or not) are circled in blue

Examples:

timave(t,x,10)                  # average the time series (t,x)
                                # using 10-day wide time bins
                                # and plot the result.  Note: this
                                # plot uses a "normal" (not inverted)
                                # y-axis, so it's "upside-down" when
                                # the "x" variable is magnitude
q = timave(t,x,10)              # average the time series (t,10)
                                # using 10-day wide time bins
                                # and store the result in an object
                                # called "q"
########################################################################
```