FIXEDSEAS.M

fixedseas computes a simple seasonal filter with fixed seasonal factors.

Usage:
```
  s = fixedseas(data,period);
  s = fixedseas([dates,data],period);
  s = fixedseas(... ,mode);
  s = fixedseas(... ,method);
  s = fixedseas(... ,method,methodarg);
  [s,aggr] = fixedseas(...);
```

data must be a vector. fixedseas is NaN tolerant, meaning data can contain NaNs.

period is a positive number which indicates the length of the seasonal cycle (i.e. period = 12 for monthly data, period = 7 for daily data having a weekly cycle, or period = 5 if the data is weekdaily).

The optional arguments determine if the filtering should be done additively or multiplicatively, and the method of filter to use for computing the trend.

'mode' is one of the following:

'none' or 'add'	The decomposition is done additively. This is the default.
'logadd	The log is applied to the data, the decomposition is then applied additively, and the exponential of the result is returned.
'mult'	The decomposition is done multiplicatively.

'method' (and 'methodarg') determines the method of trend. For possible choices, see help trendfilter. Default is a centered moving average with length equal to period ('cma',period).

'period' can also be a positive vector. In that case, the seasonal filtering is performed several times, removing cycles at all desired frequencies. In that case, 'mode' and 'method' can be cellarrays, containing one method (plus argument) for each period. The returned s is then a structure with as many components as there are components in 'period'.

If period is a vector and mode is the same for each period, an additional aggregated structure is appended to s, providing the cumulated seasonal factors etc. In that case, aggr is returned as true.

The program will select default values for 'lambda','roughness', or 'degree', respectively, if you do not specify them. If you use a vector for the 'period' argument (filtering out multiple periods), then you can also specify vectors of lambda/roughness/degree-arguments, one for each

```
component of your period-vector.

s is a struct with the following fields:
  .period     Period(s) that has/have been filtered.
  .mode       Either 'none' or 'log' or 'mult'.
  .method     The method used for computing the trend.
  .methodarg  possibly a parameter for the smoothing algorithm.
  .tbl        A short explanation of the algorith.
  .dates      The original dates. If none were provided, this is just a
              vecor counting from 1 to the number of data points.
  .dat        The original data.
  .tr         Long term trend (by default the moving average, but other
              choices are possible, see above).
  .sa         Seasonally adjusted series (= dat-sf, or exp(dat-sf),
              respectively).
  .sf         Seasonal factors.
  .ir         Irregular (= sa-tr or exp(sa-tr), respectively).

Data is decomposed into the three components, trend (tr), seasonal factor
(sf), and irregular (ir). For the additive decomposition, it is always
the case that data = tr + sf + ir. Furthermore, sa = data - sf (or
equivalently, sa = tr + ir). For the multiplicative decomposition, data =
tr * sf * ir, and sa = data ./ sf (or equivalently, sa = tr * ir).

Example 1:
  truetrend = 0.02*(1:200)' + 5;
  % truecycle = sin((1:200)'*(2*pi)/20);
  truecycle = repmat([zeros(7,1);-0.6;zeros(11,1);0.9],ceil(200/20),1);
  truecycle = truecycle(1:200);
  truecycle = truecycle - mean(truecycle);
  trueresid = 0.2*randn(200,1);
  data = truetrend + truecycle + trueresid;
  s = fixedseas(data,20);
  figure('Position',[78 183 505 679]);
  subplot(3,1,1); plot([s.dat,s.sa,s.tr,truetrend]); grid on;
  title('unadjusted and seasonally adjusted data, estimated and true trend')
  subplot(3,1,2); plot([s.sf,truecycle]); grid on;
  title('estimated and true seasonal factor')
  subplot(3,1,3); plot([s.ir,trueresid]); grid on;
  title('estimated and true irregular')
  legend('estimated','true values');

Example 2 (multiple cycles):
  truecycle2 = 0.7 * sin((1:200)'*(2*pi)/14);
  data = truetrend + truecycle + truecycle2 + trueresid;
  s = fixedseas(data,[14,20],'hp');
  figure('Position',[78 183 505 679]);
  subplot(3,1,1); plot([s.dat,s.sa,s.tr,truetrend]); grid on;
  title('unadjusted and seasonally adjusted data, estimated and true trend')
  subplot(3,1,2); plot([s.sf,truecycle+truecycle2]); grid on;
  title('estimated and true seasonal factor')
```

```
  subplot(3,1,3); plot([s.ir,trueresid]); grid on;
  title('estimated and true irregular')
  legend('estimated','true values');

Note that fixedseas(data,[14,20]) is not the same as
fixedseas(data,[20,14]). The filters are applied iteratively, from left
to right. The ordering matters, so the results differ.

Detailed description of the model: Let x be some timeseries. As an
example, we compute fixedseas(x,6).
*** STEP 1 ***
We compute a 6-period centered moving average,
  trend(t) = sum(0.5x(t-3)+x(t-2)+x(t-1)+x(t)+x(t+1)+x(t+2)+0.5x(t+3))/6
The weights on the extreme values of the window are adapted so that the
sum of the weights is equal to period. So, for instance, if period = 7,
the weight on x(t-3) and x(t+3) would be 1.0; if period = 6.5, the weight
would be 0.75.
[Note: By default the trend is computed as the centered moving average,
and this is what is explained here. Other specifications are possible,
namely detrend, hodrick-prescott, spline, polynomial, or others (see help
trendfilter).]
*** STEP 2 ***
Compute the individual deviations of x from the trend,
  d = x - trend.
*** STEP 3 ***
Compute the average deviation over all observations on a cycle of 6
periods,
  m(1) = mean(d(1) + d(7) + d(13) + d(19) + ...)
  m(2) = mean(d(2) + d(8) + d(14) + d(20) + ...)
  ...
  m(6) = mean(d(6) + d(12) + d(18) + d(24) + ...)
*** STEP 4 ***
Normalize m so that its average is zero,
  n = (m(1)+m(2)+...+m(6))/6
  sf(1) = m(1) - n, sf(2) = m(2) - n, ..., sf(6) = m(6) - n
These are the seasonal factors.
*** STEP 5 ***
Compute the seasonally adjusted time series as sa = x - sf.
*** STEP 6 ***
Compute the irregular as ir = sa - trend. This is the part of the
fluctuations of x that is not explained by the seasonal factors or the
trend (= moving average).

STEP 1 as described here is for the 'moving average' trend type, which
is the default. This step is different for the different trend types
that are available. STEP 2 to 6 are, however, independent of the type of
trend that is computed.

If the multiplicative option is used, the logarithm of the data is
processed and the exponential of the processed time series is returned.
So, s = fixedseas(data,period,'log') is materially the same as
```

```
  s2 = fixedseas(log(data),period). Then, exp(s2.sa) = s.sa,
  exp(s2.sf) = s.sf, and exp(s2.tr) = s.tr.

  NOTE: This file is part of the X-13 toolbox, but it is completely
  independent of the Census X-13 program. It is part of the 'seas' addition
  to the toolbox which allows you to implement seasonal filters without
  using the Census Bureau programs.

  The toolbox consists of the following programs, guix, x13, makespec, x13spec,
  x13series, x13composite, x13series.plot,x13composite.plot, x13series.seasbreaks,
  x13composite.seasbreaks, fixedseas, camplet, spr, InstallMissingCensusProgram
  makedates, yqmd, TakeDayOff, EasterDate.

Author  : Yvan Lengwiler
Version : 1.50

If you use this software for your publications, please reference it as:
Yvan Lengwiler, 'X-13 Toolbox for Matlab, Version 1.50', Mathworks File
Exchange, 2014-2021.
url: https://ch.mathworks.com/matlabcentral/fileexchange/49120-x-13-toolbox-for-
seasonal-filtering
```