Time-series filtering techniques in Stata

Christopher F Baum

Department of Economics, Boston College
Chestnut Hill, MA 02467 USA

September 2006

Introduction

There are several time-series filters commonly used in macroeconomic and financial research to separate the behavior of a timeseries into trend *vs.* cyclical and irregular components. These techniques can usually be expressed in terms of linear algebra, and reliable code exists in matrix languages for their implementation.

I briefly describe the concept of time-series filtering, and then present several new implementations of time-series filters for Stata users written in Mata. These routines avoid matrix size constraints and are much faster than previous versions translated from Fortran written in the ado-file language.

Introduction

There are several time-series filters commonly used in macroeconomic and financial research to separate the behavior of a timeseries into trend *vs.* cyclical and irregular components. These techniques can usually be expressed in terms of linear algebra, and reliable code exists in matrix languages for their implementation.

I briefly describe the concept of time-series filtering, and then present several new implementations of time-series filters for Stata users written in Mata. These routines avoid matrix size constraints and are much faster than previous versions translated from Fortran written in the ado-file language.

Theoretical background

Linear filtering involves generating a linear combination of successive elements of a discrete-time signal x_t, as represented by

$$y_t = \psi(L)x_t = \sum_j \psi_j x_{t-j}$$

where L is the lag operator, equivalent to Stata's time-series operator `L.`. The sequence

$$\psi(L) = \{\cdots + \psi_{-1}L^{-1} + \psi_0 + \psi_1 L + \ldots\}$$

is the linear filter, and defines the response to an input signal in the form of a unit impulse.

Moving average filters

The sequence of filter weights $\psi(L)$ can be finite or infinite. A finite sequence is described as a *moving average filter* or a finite impulse-response (FIR) filter. When the filter produces an impulse response of an indefinite duration, it is called an infinite impulse-response (IIR) filter. If all of the filter weights are associated with non-negative powers of L, the filter is known as *causal* or *backward-looking*.

We often consider finite moving average processes in working with `arima` models. As is well known, a finite moving average or *MA(q)* process has *finite memory*: only the last q periods of the x sequence affect the "output of the filter", or as we would describe it, the response variable y.

Moving average filters

The sequence of filter weights $\psi(L)$ can be finite or infinite. A finite sequence is described as a *moving average filter* or a finite impulse-response (FIR) filter. When the filter produces an impulse response of an indefinite duration, it is called an infinite impulse-response (IIR) filter. If all of the filter weights are associated with non-negative powers of L, the filter is known as *causal* or *backward-looking*.

We often consider finite moving average processes in working with `arima` models. As is well known, a finite moving average or *MA(q)* process has *finite memory*: only the last q periods of the x sequence affect the "output of the filter", or as we would describe it, the response variable y.

If our interest is in computing moving averages, we often employ a *centered moving average*: that is, one that includes several leads and lags. A 5-term centered moving average uses two leading values, the current value, and two lagged values to generate the averaged series.

See `tssmooth ma` or the egen function `filter` from Cox's `egenmore` package for implementations of the moving average filter, centered or uncentered. In an uncentered filter, we use only lagged values in a moving average filter, allowing computation of the averaged series through the last available observation.

If our interest is in computing moving averages, we often employ a *centered moving average*: that is, one that includes several leads and lags. A 5-term centered moving average uses two leading values, the current value, and two lagged values to generate the averaged series.

See `tssmooth ma` or the `egen` function `filter` from Cox's `egenmore` package for implementations of the moving average filter, centered or uncentered. In an uncentered filter, we use only lagged values in a moving average filter, allowing computation of the averaged series through the last available observation.

Smoothers

Official Stata contains a number of other routines to smooth and forecast univariate time-series data, as described in [TS] `tssmooth`. These include several *recursive* smoothers (exponential, double exponential, and Holt–Winters with and without seasonal adjustments) as well as a nonlinear filter which applies [R] `smooth`, a robust nonlinear smoother to the time series.

As the manual entry indicates, the moving average and nonlinear filter routines are generally used to extract the trend from a time series "while omitting the high-frequency or noise components." If you are interested in those latter components, they may be recovered as the difference between the original series and the smoothed series.

Smoothers

Official Stata contains a number of other routines to smooth and forecast univariate time-series data, as described in [TS] `tssmooth`. These include several *recursive* smoothers (exponential, double exponential, and Holt–Winters with and without seasonal adjustments) as well as a nonlinear filter which applies [R] `smooth`, a robust nonlinear smoother to the time series.

As the manual entry indicates, the moving average and nonlinear filter routines are generally used to extract the trend from a time series "while omitting the high-frequency or noise components." If you are interested in those latter components, they may be recovered as the difference between the original series and the smoothed series.

The effects of linear filtering

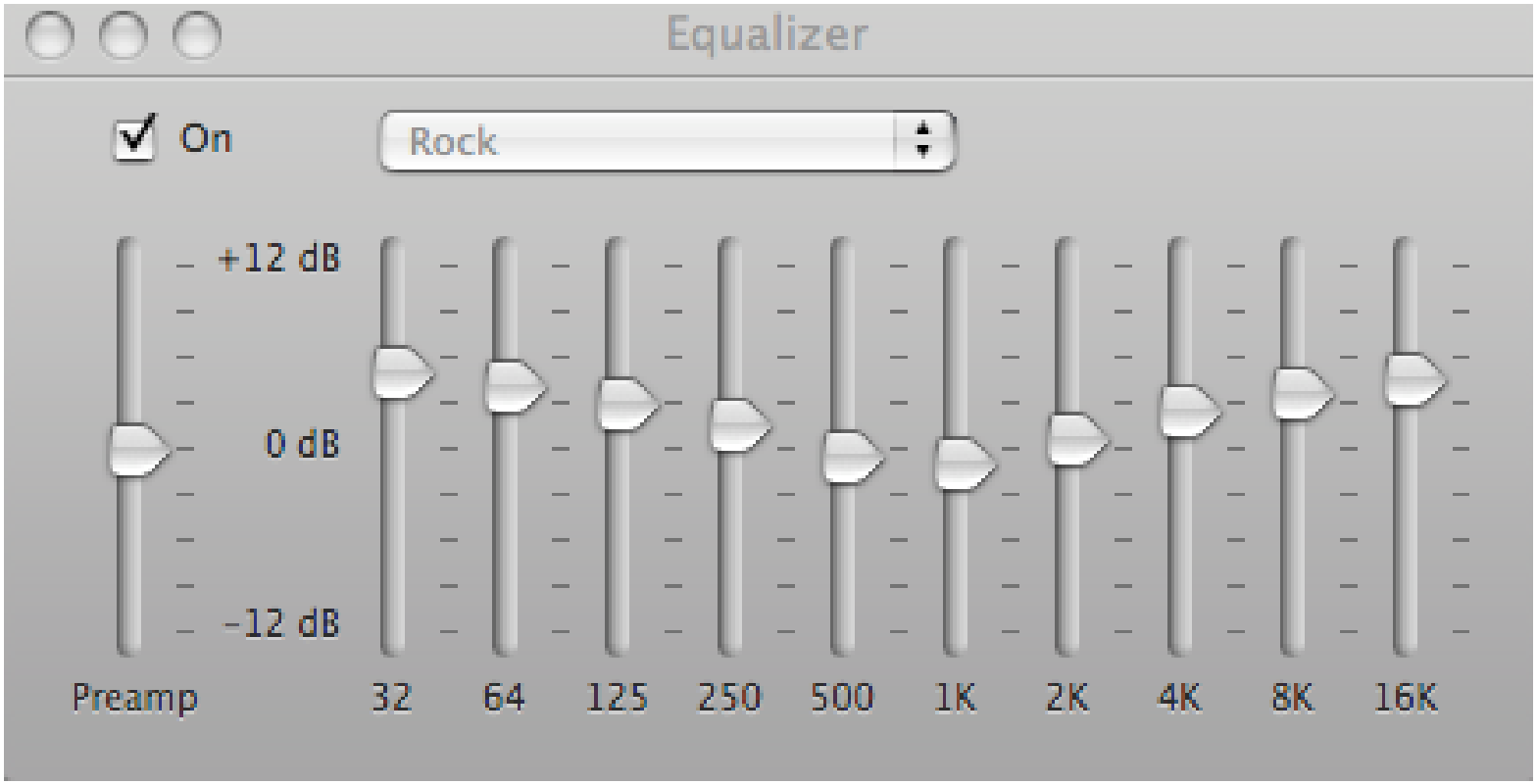
Application of a linear filter—either FIR or IIR—will affect the cyclical components of a time series, and is often studied in the frequency domain rather than the time domain. The filter is liable to alter the amplitude of any cyclical component. This effect varies according to the frequency of the component, and is described as the *gain* of the filter.

The concept is familiar to anyone who uses a stereo: the treble control strengthens or weakens the high frequencies of your favorite tune, while the bass control affects the low frequencies. In a fancier setup, a multi-band graphic equalizer may allow you to vary a number of the frequency bands.

The effects of linear filtering

Application of a linear filter—either FIR or IIR—will affect the cyclical components of a time series, and is often studied in the frequency domain rather than the time domain. The filter is liable to alter the amplitude of any cyclical component. This effect varies according to the frequency of the component, and is described as the *gain* of the filter.

The concept is familiar to anyone who uses a stereo: the treble control strengthens or weakens the high frequencies of your favorite tune, while the bass control affects the low frequencies. In a fancier setup, a multi-band graphic equalizer may allow you to vary a number of the frequency bands.


Equalizer
On
Rock
+12 dB
0 dB
-12 dB
Preamp
32
64
125
250
500
1K
2K
4K
8K
16K

In this example from iTunes' equalizer, the low frequencies (32, 64, 125 Hz) are toward the left, while the high frequencies (4 KHz, 8 Khz, 16 Khz) are to the right. By moving the sliders up or down, we can cause the resulting filtered signal to contain only a subset of the frequencies in the original music.

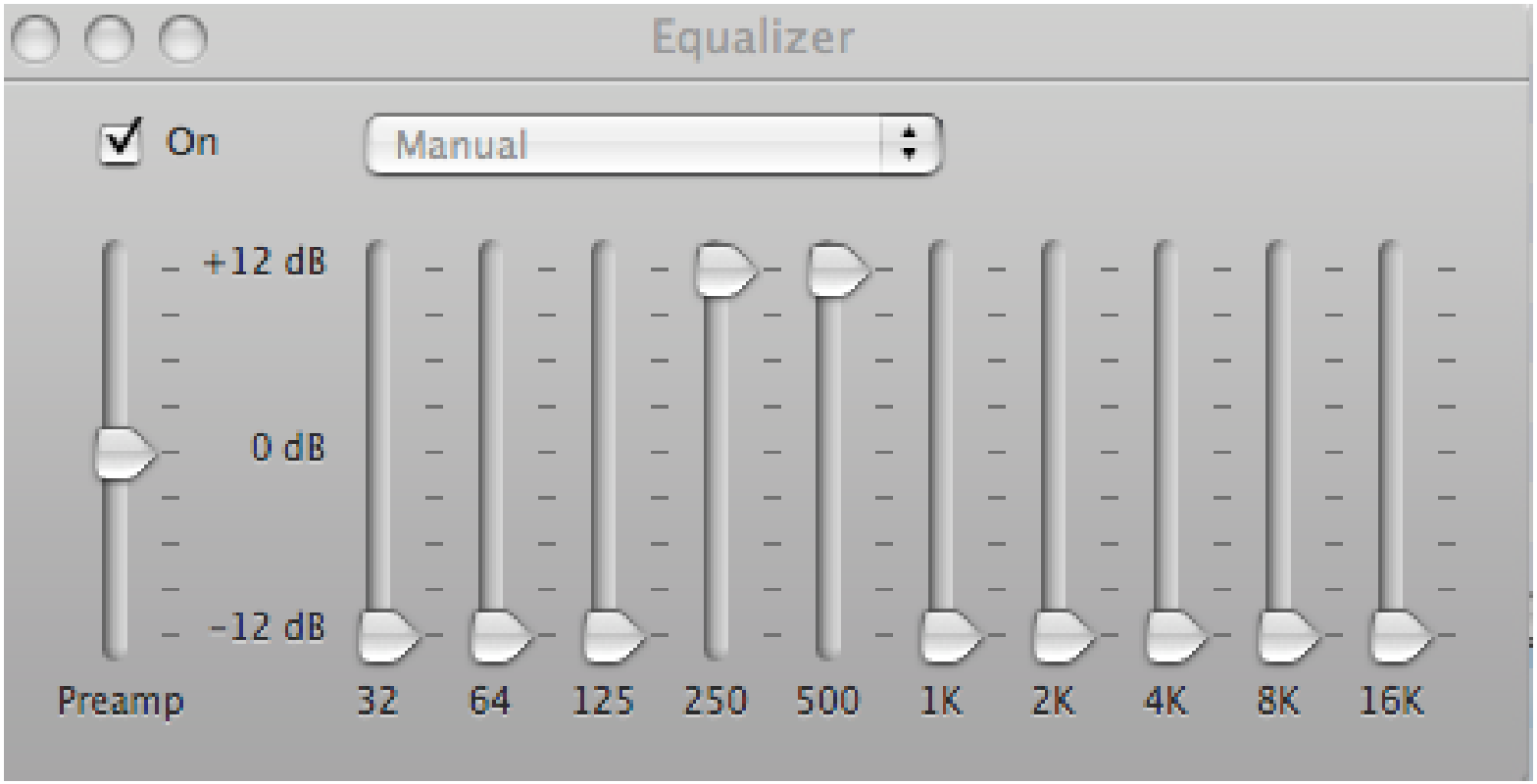
This allows us to illustrate three kinds of time-series filters: the *lowpass* filter, which "passes" only the low (bass) frequencies; the *highpass* filter, which passes only the high (treble) frequencies; and the *bandpass* filter, which passes only the frequencies within a certain frequency band.

For example:

In this example from iTunes' equalizer, the low frequencies (32, 64, 125 Hz) are toward the left, while the high frequencies (4 KHz, 8 Khz, 16 Khz) are to the right. By moving the sliders up or down, we can cause the resulting filtered signal to contain only a subset of the frequencies in the original music.

This allows us to illustrate three kinds of time-series filters: the *lowpass* filter, which "passes" only the low (bass) frequencies; the *highpass* filter, which passes only the high (treble) frequencies; and the *bandpass* filter, which passes only the frequencies within a certain frequency band.

For example:


Equalizer
On
Manual
+12 dB
0 dB
-12 dB
Preamp
32
64
125
250
500
1K
2K
4K
8K
16K

This *bandpass* filter accentuates the components of the music in the vicinity of 250 Hz - 500 Hz, and downplays all other frequencies. (Note that "concert pitch", sounded by the oboe in tuning a symphony orchestra, is 'A above middle C', 440 Hz). Since iTunes' equalizer does not allow me to completely suppress the unwanted frequencies, it is not a pure bandpass filter, but it does a pretty good job—in the sense that the resulting music sounds pretty bad! Those suppressed low and high frequencies are recognized as important components of the "signal", or musical tones.

What does this have to do with economic time series? The *frequencies* in that case refer to the periodic behavior of the series. For instance, 2-5 year cycles in economic activity are often referred to as *business cycle frequencies*. Lower frequencies refer to longer cycles in the data, while higher frequencies refer to, for instance, annual cycles in quarterly or monthly data: what we consider seasonality.

The shortest period which we can study is a cycle lasting two periods, corresponding to the *Nyquist frequency*. If we wish to study shorter cycles—e.g. quarterly movements in companies' cash positions—we must sample the data at a higher (monthly, weekly, daily) frequency.

What does this have to do with economic time series? The *frequencies* in that case refer to the periodic behavior of the series. For instance, 2-5 year cycles in economic activity are often referred to as *business cycle frequencies*. Lower frequencies refer to longer cycles in the data, while higher frequencies refer to, for instance, annual cycles in quarterly or monthly data: what we consider seasonality.

The shortest period which we can study is a cycle lasting two periods, corresponding to the *Nyquist frequency*. If we wish to study shorter cycles—e.g. quarterly movements in companies' cash positions—we must sample the data at a higher (monthly, weekly, daily) frequency.

Filtering economic time series

The filtering techniques commonly applied to economic time series, usually sampled at quarterly, annual or monthly frequencies, may be characterized as various sorts of lowpass, highpass or bandpass filters. Unlike the application of moving averages—where the averaged series itself may be the object of interest—the filtering techniques employed in economics often focus on producing a smoothed series x_t^* and recovering the residual components of the original series $z_t = x_t - x_t^*$ as detrended x_t.

Stata filtering routines

Taking advantage of the similarity between Stata's Mata language and that of MATLAB, I have produced translations of several filtering routines originally coded by Pawel Kowal of the Warsaw School of Economics.

These include:

- `hprescott`: Hodrick–Prescott filter
- `bking`: Baxter–King filter
- `cfitzrw`: Christiano-Fitzgerald random walk band pass filter
- `butterworth`: Butterworth square wave high pass filter

Stata filtering routines

Taking advantage of the similarity between Stata's Mata language and that of MATLAB, I have produced translations of several filtering routines originally coded by Pawel Kowal of the Warsaw School of Economics.

These include:

- `hprescott`: Hodrick–Prescott filter
- `bking`: Baxter–King filter
- `cfitzrw`: Christiano-Fitzgerald random walk band pass filter
- `butterworth`: Butterworth square wave high pass filter

Stata filtering routines

Taking advantage of the similarity between Stata's Mata language and that of MATLAB, I have produced translations of several filtering routines originally coded by Pawel Kowal of the Warsaw School of Economics.

These include:

- `hprescott`: Hodrick–Prescott filter
- `bking`: Baxter–King filter
- `cfitzrw`: Christiano-Fitzgerald random walk band pass filter
- `butterworth`: Butterworth square wave high pass filter

Stata filtering routines

Taking advantage of the similarity between Stata's Mata language and that of MATLAB, I have produced translations of several filtering routines originally coded by Pawel Kowal of the Warsaw School of Economics.

These include:

- `hprescott`: Hodrick–Prescott filter
- `bking`: Baxter–King filter
- `cfitzrw`: Christiano-Fitzgerald random walk band pass filter
- `butterworth`: Butterworth square wave high pass filter

Hodrick–Prescott filter

For better or worse, the filter best known to most economics is the "HP" filter (J.Money,Credit,Banking 1997). As Ravn and Uhlig (Rev.Econ.Stat., 2002) state, "...the HP filter has become a standard method for removing trend movements in the business cycle literature." As those authors state, use of the filter has been subject to heavy criticism, but it "has withstood the test of time and the fire of discussion relatively well."

One attraction of the HP filter is that it may be applied to nonstationary time series (series containing one or more unit roots in their autoregressive representation), a relevant concern for many macroeconomic and financial time series.

Hodrick–Prescott filter

For better or worse, the filter best known to most economics is the "HP" filter (J.Money,Credit,Banking 1997). As Ravn and Uhlig (Rev.Econ.Stat., 2002) state, "...the HP filter has become a standard method for removing trend movements in the business cycle literature." As those authors state, use of the filter has been subject to heavy criticism, but it "has withstood the test of time and the fire of discussion relatively well."

One attraction of the HP filter is that it may be applied to nonstationary time series (series containing one or more unit roots in their autoregressive representation), a relevant concern for many macroeconomic and financial time series.

The HP filter removes a smooth trend τ_t from a time series x_t by solving the minimization problem

$$min \sum_{t=1}^{T} \left[(x_t - \tau_t)^2 + \lambda \left((\tau_{t+1} - \tau_t) - (\tau_t - \tau_{t-1}) \right)^2 \right]$$

with respect to τ_t. The residual, or deviation from trend $z_t = x_t - \tau_t$ is commonly referred to as the business cycle component, and is the object of economic interest. In this sense the HP filter is a highpass filter, removing the trend and returning high-frequency components in z_t.

The parameter λ penalizes fluctuations in the second differences of x_t, and must be specified by the user of the HP filter.

Consider a time series $x_t = \tau_t + c_t$ composed of trend and cyclical components (where the latter will also incorporate any irregular components). If c_t as well as the second difference of τ_t is normally and independently distributed, the HP filter is an "optimal filter" and λ is given as the ratio of the two variances, $\frac{\sigma_c^2}{\sigma_{\Delta^2\tau}^2}$, where Δ^2 is the second difference operator (Stata's `D2`). With an unknown τ_t, these variances are not known.

Common wisdom has been to use $\lambda = 1600$ when applying the HP filter to quarterly economic data. For other frequencies, Ravn and Uhlig have shown that quite different values should be used: 6.25 for annual data, and 129,600 for monthly data.

Consider a time series $x_t = \tau_t + c_t$ composed of trend and cyclical components (where the latter will also incorporate any irregular components). If c_t as well as the second difference of τ_t is normally and independently distributed, the HP filter is an "optimal filter" and λ is given as the ratio of the two variances, $\frac{\sigma_c^2}{\sigma_{\Delta^2\tau}^2}$, where Δ^2 is the second difference operator (Stata's `D2`). With an unknown τ_t, these variances are not known.

Common wisdom has been to use $\lambda = 1600$ when applying the HP filter to quarterly economic data. For other frequencies, Ravn and Uhlig have shown that quite different values should be used: 6.25 for annual data, and 129,600 for monthly data.

The hprescott command

The new `hprescott` command will apply the HP filter to one or more variables in a time series context. It may be applied in a panel context with the `by:` prefix. The routine returns two variables for each input variable: the smoothed series and the "business cycle component" described as z_t above. The syntax:

`hprescott` *varlist* `[if` *exp*`]` `[in` *range*`],` `stub(`*abbrev*`)` `[smooth(#)]`

where the `stub` provides prefixes for the new variables (and must be given). The `smooth` parameter gives λ, and defaults to 1600. If Stata can determine that the frequency of the data is annual or monthly, the appropriate Ravn–Uhlig value is used instead. In any case the desired λ may be specified.

The hprescott command

This new `hprescott` command, written using Mata, is a considerable improvement of my previous version of the HP filter. That version, written in the ado-file language and translated from Prescott's original Fortran code, ran quite slowly. It was not limited by Stata's matrix language limits as was Schmidt's original routine (Stata Tech. Bull. 17), but even for series of moderate length the Mata-based version runs about 4 times faster. The original routine is available for use by Stata 8.2 users as `hprescott8`, and is included in the SSC archive package `hprescott`.

I will discuss the several filtering routines and then display examples of their application.

Baxter–King filter

In contrast to the HP filter—a highpass filter—the Baxter–King filter (Rev.Econ.Stat., 1999) is a *bandpass* filter which allows suppression of both the low frequency trend components and the high frequency components in an economic series. They argue that the National Bureau of Economic Research (NBER) definition of a business cycle requires a bandpass approach: for instance, retaining "components of the time series with periodic fluctuations between six and 32 quarters, while removing components at higher and lower frequencies."

In the theory of linear filtering, the optimal bandpass filter is well defined, but it is a moving average of infinite order. Baxter and King's study focuses on constructing a good approximation to the optimal filter. The approximation improves with a longer moving average, but a longer MA will drop additional observations at each end of the filtered series. They conclude that three years of data (at either annual or quarterly frequencies) should be used to construct the filter.

BK set out six objectives for an appropriate filter: (1) it should not modify the properties of the extracted component; (2) it should not create "phase shift", or alter the timing relationships in the series; it should be an optimal approximation to the ideal bandpass filter; (4) it should be applicable to $I(1)$ or $I(2)$ data, as well as to series exhibiting a quadratic trend; (5) it should provide constant (non-time-varying) coefficients in the MA representation of the filter; (6) it should be operational.

The first two criteria imply that a moving average of the data with symmetric weights on leads and lags must be defined. This in turn implies that the filter will drop observations on both ends of the original series.

BK set out six objectives for an appropriate filter: (1) it should not modify the properties of the extracted component; (2) it should not create "phase shift", or alter the timing relationships in the series; it should be an optimal approximation to the ideal bandpass filter; (4) it should be applicable to $I(1)$ or $I(2)$ data, as well as to series exhibiting a quadratic trend; (5) it should provide constant (non-time-varying) coefficients in the MA representation of the filter; (6) it should be operational.

The first two criteria imply that a moving average of the data with symmetric weights on leads and lags must be defined. This in turn implies that the filter will drop observations on both ends of the original series.

The bking command

The new `bking` command will apply the BK filter to one or more variables in a time series context. It may be applied in a panel context with the `by:` prefix. The routine returns a variable containing the filtered series. The syntax:

`bking` *varlist* `[if` *exp*`]` `[in` *range*`], plo(#) phi(#) stub(`*abbrev*`) [k(#)]`

where the `stub` provides prefixes for the new variables (and must be given). The `plo` and `phi` parameters give the minimum periodicity and maximum periodicity to be included in the filtered series. The optional parameter `k` gives the lead-lag length of the filter. `k` observations will be lost at each end of the filtered series.

The bking command

For quarterly data, Baxter and King recommend the "Burns–Mitchell" settings of 6 and 32 quarters (1.5–8 years) for `plo, phi`, and `k=12`. For annual data, they recommend 2 and 8 years, with `k=3`, which implies a lowpass filter. For monthly data, they recommend 18 and 96 months (1.5–8 years), with `k=12`.

This new `bking` command, written using Mata, is a considerable improvement of my previous version of the BK filter. That version, written in the ado-file language using Stata's matrix language (which restricted the length of series it could filter), ran quite slowly relative to the Mata version. The original routine is available for use by Stata 8.2 users as `bking8`, and is included in the SSC archive package `bking`.

The bking command

For quarterly data, Baxter and King recommend the "Burns–Mitchell" settings of 6 and 32 quarters (1.5–8 years) for `plo, phi`, and `k=12`. For annual data, they recommend 2 and 8 years, with `k=3`, which implies a lowpass filter. For monthly data, they recommend 18 and 96 months (1.5–8 years), with `k=12`.

This new `bking` command, written using Mata, is a considerable improvement of my previous version of the BK filter. That version, written in the ado-file language using Stata's matrix language (which restricted the length of series it could filter), ran quite slowly relative to the Mata version. The original routine is available for use by Stata 8.2 users as `bking8`, and is included in the SSC archive package `bking`.

Christiano–Fitzgerald random walk filter

Christiano and Fitzgerald, in their Intl.Econ.Rev. (2003) article, discuss optimal finite-sample approximations to the ideal bandpass filter, including one-sided filters that can be used in real time. Most of their work leads to quite complex structures that require numerical integration. Thankfully, they find that "a simple approach, based on the generally false assumption that the data are generated by a random walk, is nearly optimal."

CF conclude that optimal filter approximations violate some of BK's desired properties: e.g., "the weights in the OFA are not symmetric in future and past values of the data, and they vary over time." They find that imposing stationary and symmetric weights on the approximation problem is usually inappropriate.

They compare their random walk (RW) filter to the BK filter and an alternative based on regression on sine and cosine functions (the Trigonometric Regression filter) and find that the RW filter dominates in terms of an optimality criterion (match to the optimal bandpass filter). Unlike the BK filter, no observations are lost by the RW filter.

CF also find that their RW filter is an improvement over the HP filter when applied to quarterly data, particularly toward the end of the sample. The difference is small in most applications, but may be more marked when the filters are applied to other data frequencies.

CF conclude that their RW filter compares very favorably to the optimal approximation to the bandpass filter (i.e., one that does not involve the assumption of random walk behavior) for the economic series they study. Given that the RW filter is very easily computed, it provides an attractive alternative to HP, BK, and the computational burden of the optimal approximation.

CF also find that their RW filter is an improvement over the HP filter when applied to quarterly data, particularly toward the end of the sample. The difference is small in most applications, but may be more marked when the filters are applied to other data frequencies.

CF conclude that their RW filter compares very favorably to the optimal approximation to the bandpass filter (i.e., one that does not involve the assumption of random walk behavior) for the economic series they study. Given that the RW filter is very easily computed, it provides an attractive alternative to HP, BK, and the computational burden of the optimal approximation.

The cfitzrw command

The `cfitzrw` command will apply the RW filter to one or more variables in a time series context. It may be applied in a panel context with the `by:` prefix. The routine returns a variable containing the filtered series. The syntax:

`cfitzrw` *varlist* `[if` *exp*`] [in` *range*`], plo(#) phi(#) stub(`*abbrev*`)`

where the `stub` provides prefixes for the new variables (and must be given). The `plo` and `phi` parameters give the minimum periodicity and maximum periodicity to be included in the filtered series.

The routine is available as the SSC package `cfitzrw`.

Butterworth square-wave high pass filter

In a recent article in J. Econometrics (2000), Pollock argues that commonly used linear filters such as HP are not sufficiently flexible in the context of common features of economic data: e.g. a sharp break in the underlying trend of a series. He advocates a *rational square-wave filter*, known to engineers as the digital Butterworth filter, as a better solution.

Recall that the HP filter uses a single parameter, λ, to control the smoothness of the trend series. Pollock points out that this parameter affects both the location of the cut-off point on the frequency axis and the rate of transition between the passband and stopband. In contrast, the Butterworth filter has greater flexibility in approximating a phase-neutral square wave filter, which precisely demarcates frequencies to be passed and frequencies to be discarded.

Butterworth square-wave high pass filter

In a recent article in J. Econometrics (2000), Pollock argues that commonly used linear filters such as HP are not sufficiently flexible in the context of common features of economic data: e.g. a sharp break in the underlying trend of a series. He advocates a *rational square-wave filter*, known to engineers as the digital Butterworth filter, as a better solution.

Recall that the HP filter uses a single parameter, λ, to control the smoothness of the trend series. Pollock points out that this parameter affects both the location of the cut-off point on the frequency axis and the rate of transition between the passband and stopband. In contrast, the Butterworth filter has greater flexibility in approximating a phase-neutral square wave filter, which precisely demarcates frequencies to be passed and frequencies to be discarded.

The Butterworth filter involves a smoothing operation which is applied both forwards and backwards via a recursive filtering process to the time series to be filtered. The combination of the resulting filtered series is guaranteed to preserve phase, as the phase shifts caused by smoothing cancel each other.

Such a recursive filter is sensitive to the initial conditions at either end of the time series. Pollock circumvents difficulties with initial conditions by generating them from a "version of the data sequence which has been reduced to stationarity by repeated differencing." For the filter to work properly, the `order` of the filter must exceed the number of unit roots in the series. The equivalent of the HP smoothing parameter λ is determined by the user-specified minimum periodicity to be retained in the filtered series.

The Butterworth filter involves a smoothing operation which is applied both forwards and backwards via a recursive filtering process to the time series to be filtered. The combination of the resulting filtered series is guaranteed to preserve phase, as the phase shifts caused by smoothing cancel each other.

Such a recursive filter is sensitive to the initial conditions at either end of the time series. Pollock circumvents difficulties with initial conditions by generating them from a "version of the data sequence which has been reduced to stationarity by repeated differencing." For the filter to work properly, the `order` of the filter must exceed the number of unit roots in the series. The equivalent of the HP smoothing parameter λ is determined by the user-specified minimum periodicity to be retained in the filtered series.

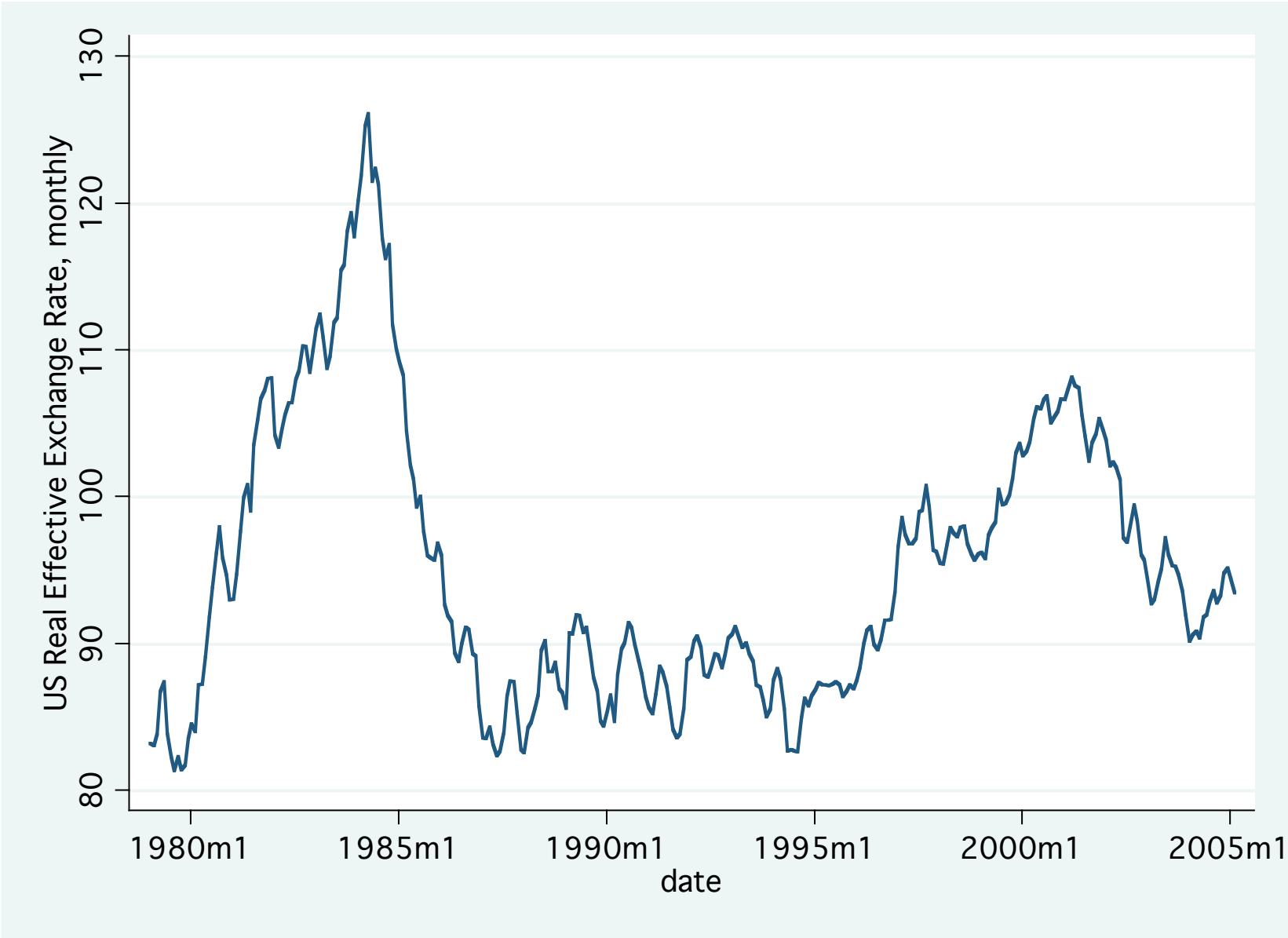
butterworth

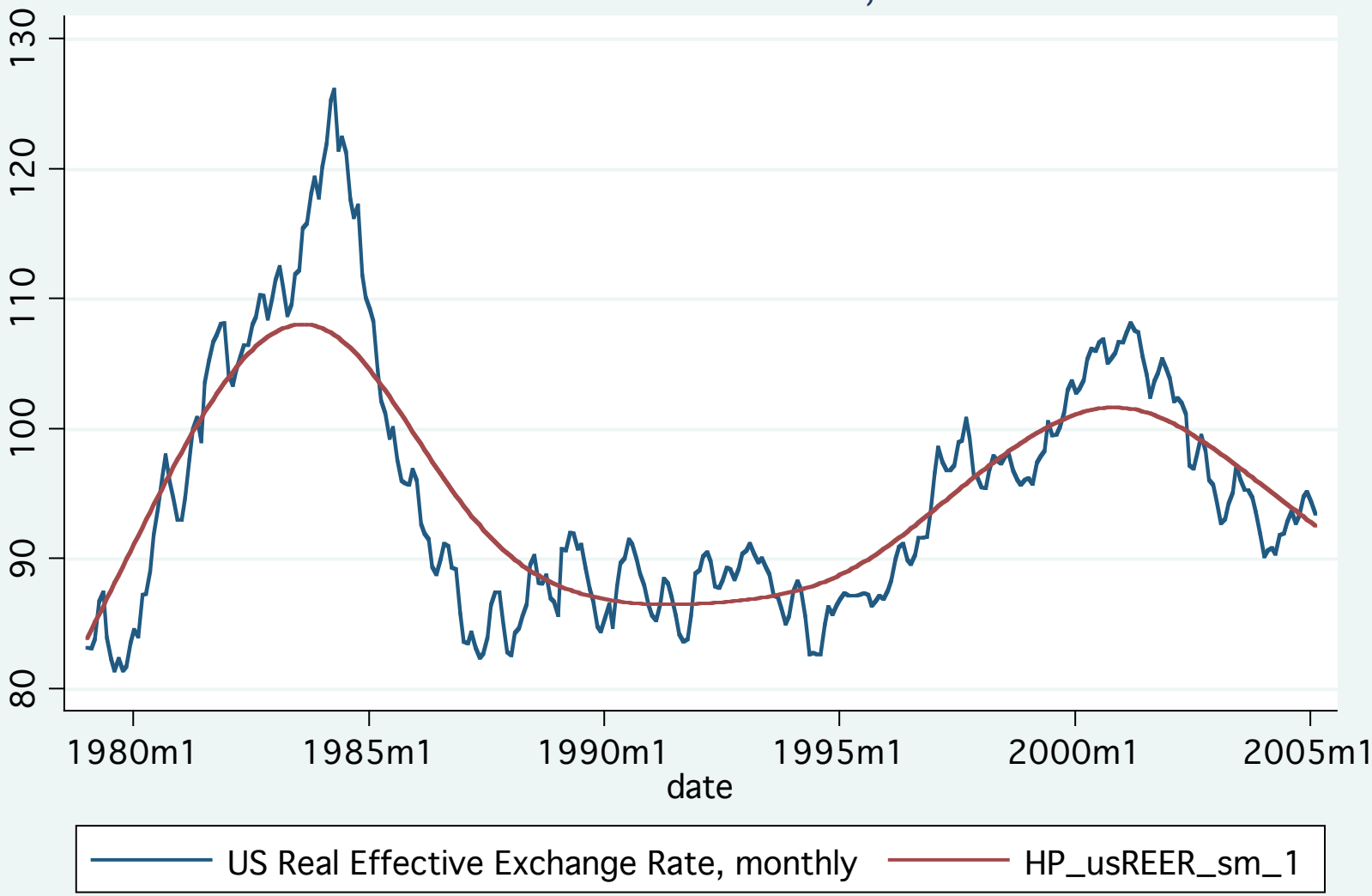
The `butterworth` command will apply the Butterworth square-wave high pass filter to one or more variables in a time series context. It may be applied in a panel context with the `by:` prefix. The routine returns a variable containing the filtered series. The syntax:

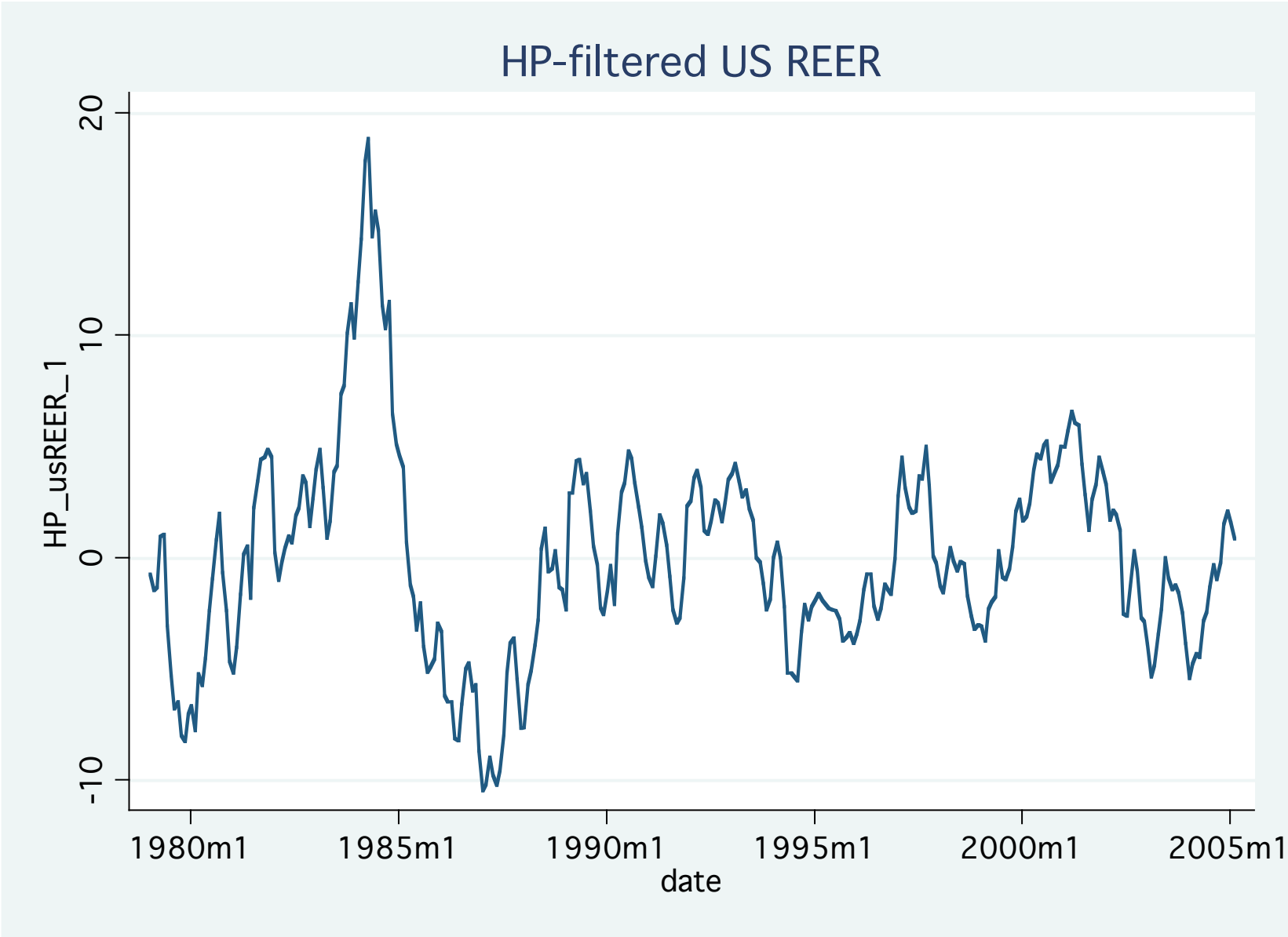
`butterworth` *varlist* `[if` *exp*`] [in` *range*`], freq(#) stub(`*abbrev*`) [order(#)]`

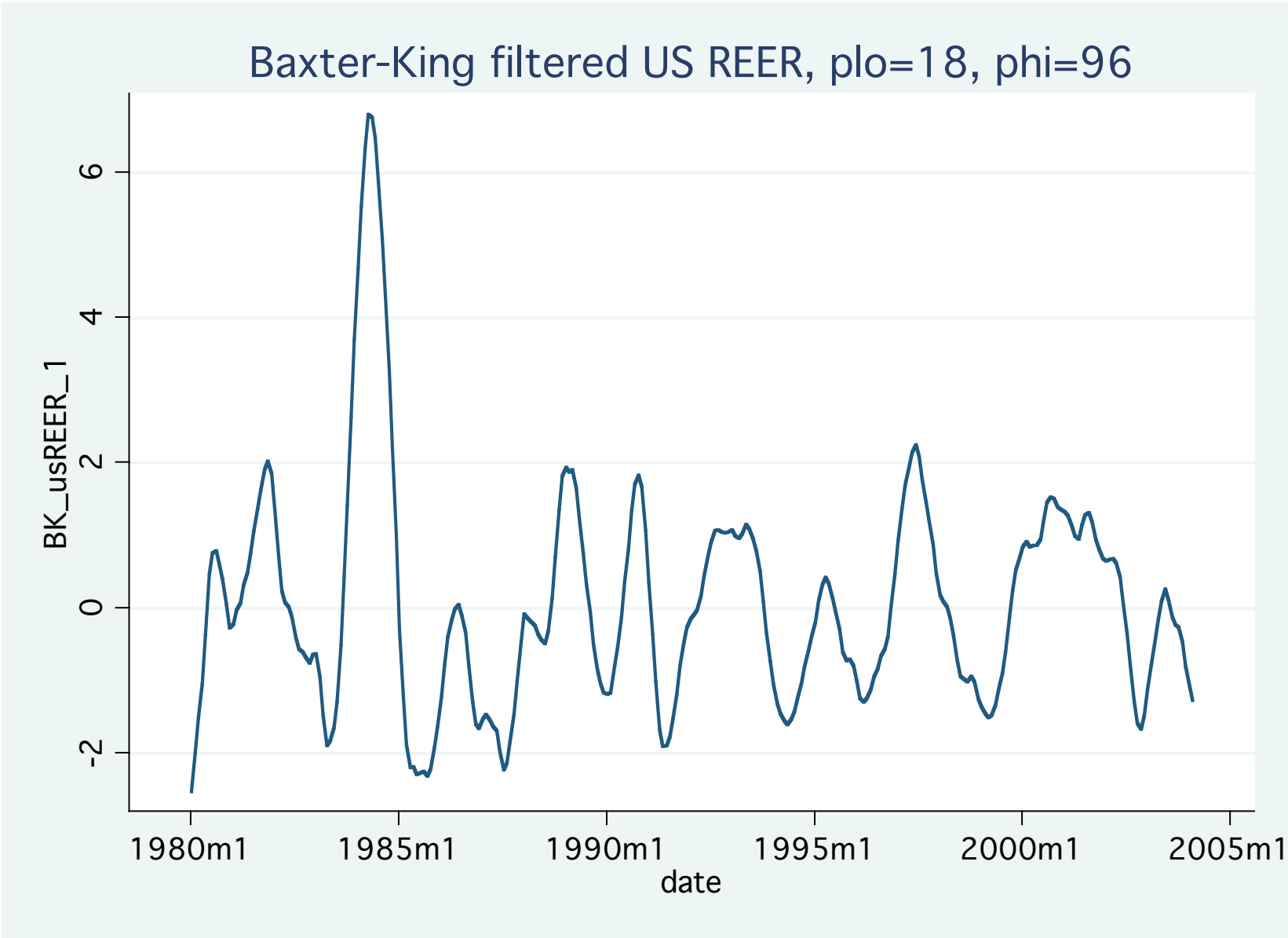
where the `stub` provides prefixes for the new variables (and must be given). The `freq` parameter gives the minimum period of oscillation to be included in the filtered series. The optional parameter `order` specifies the order of the filter (default 2).

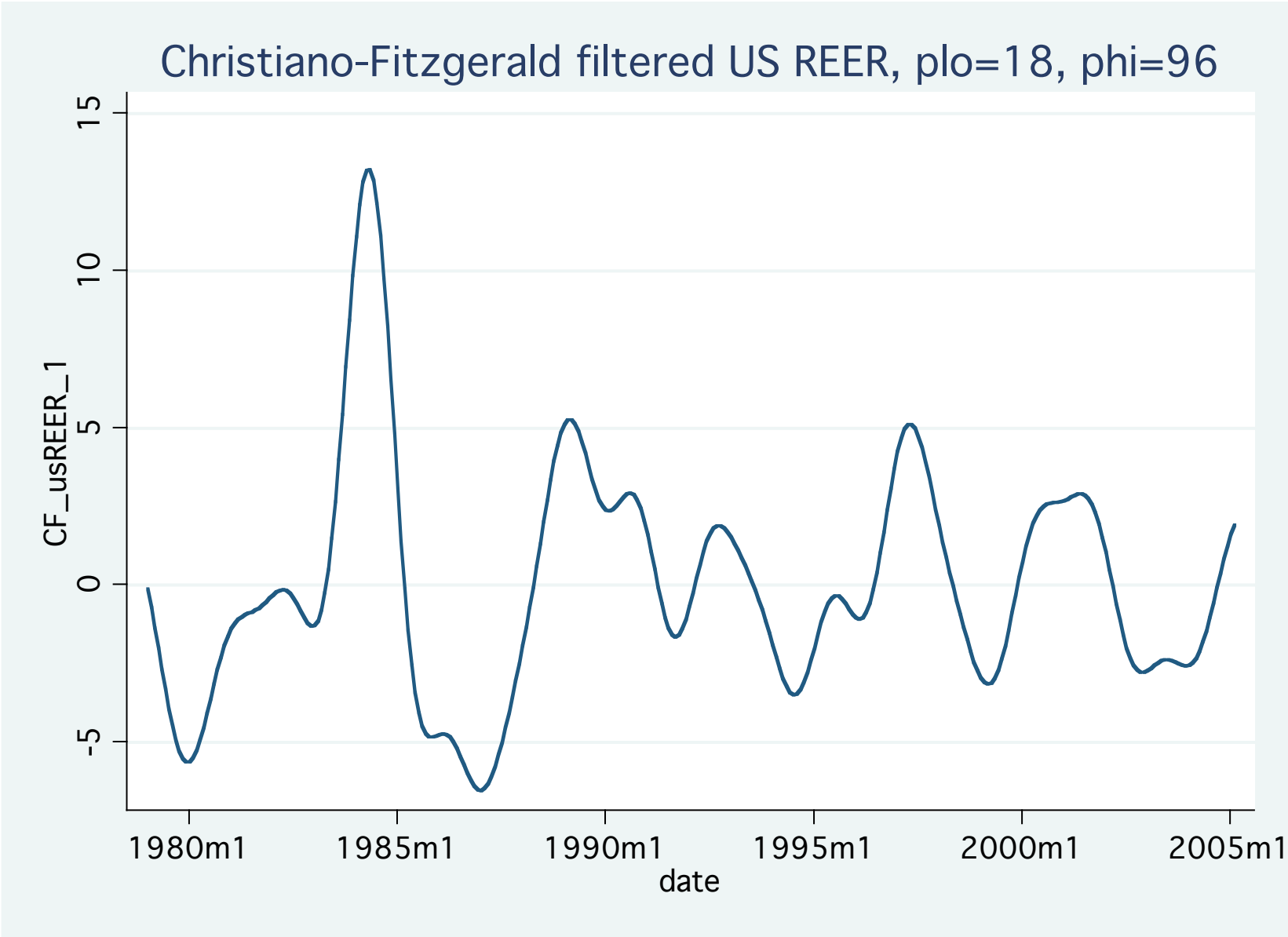
The routine is available as the SSC package `butterworth`.

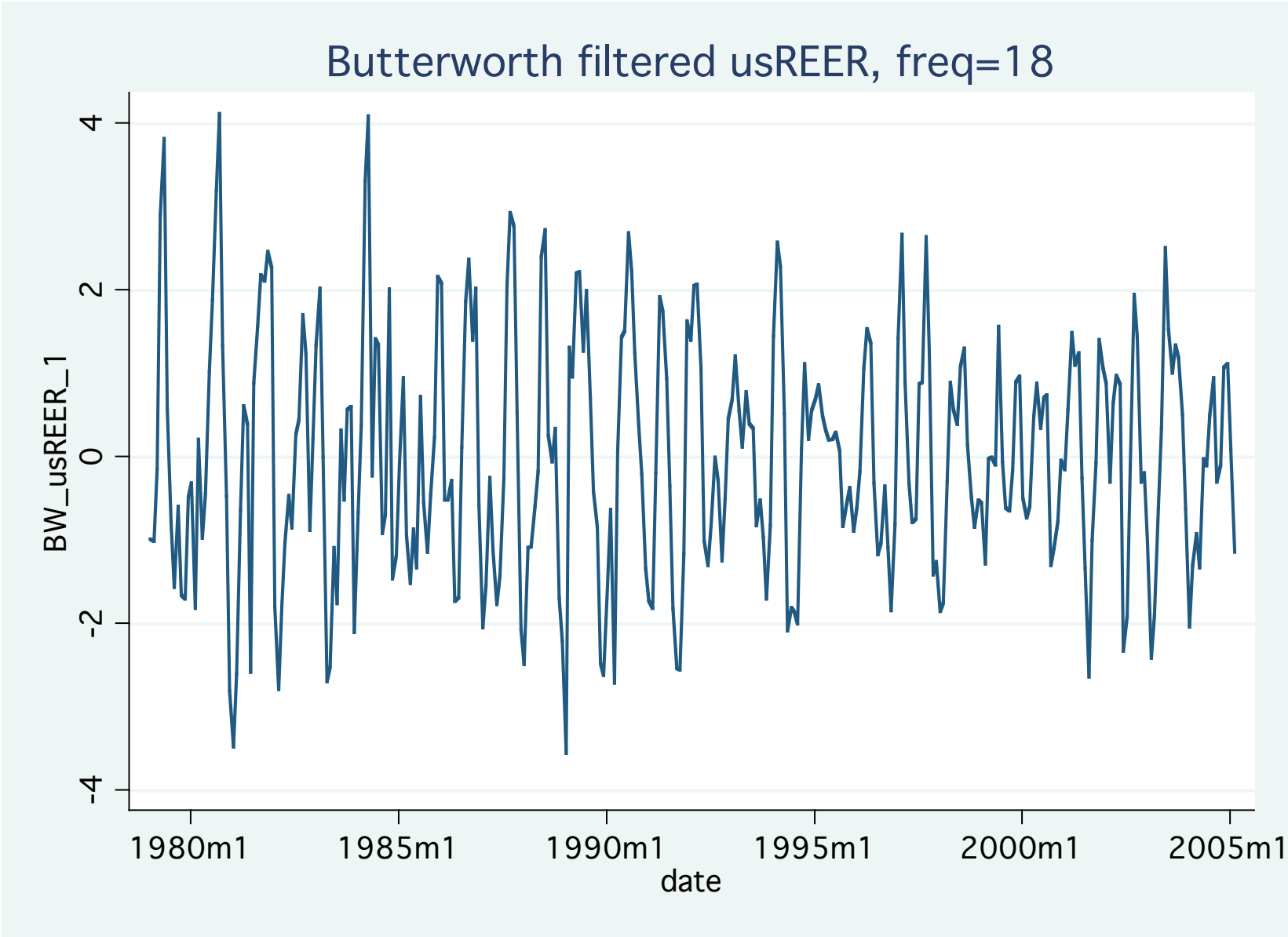

US Real Effective Exchange Rate, monthly
80
90
100
110
120
130
1980m1
1985m1
1990m1
1995m1
2000m1
2005m1
date

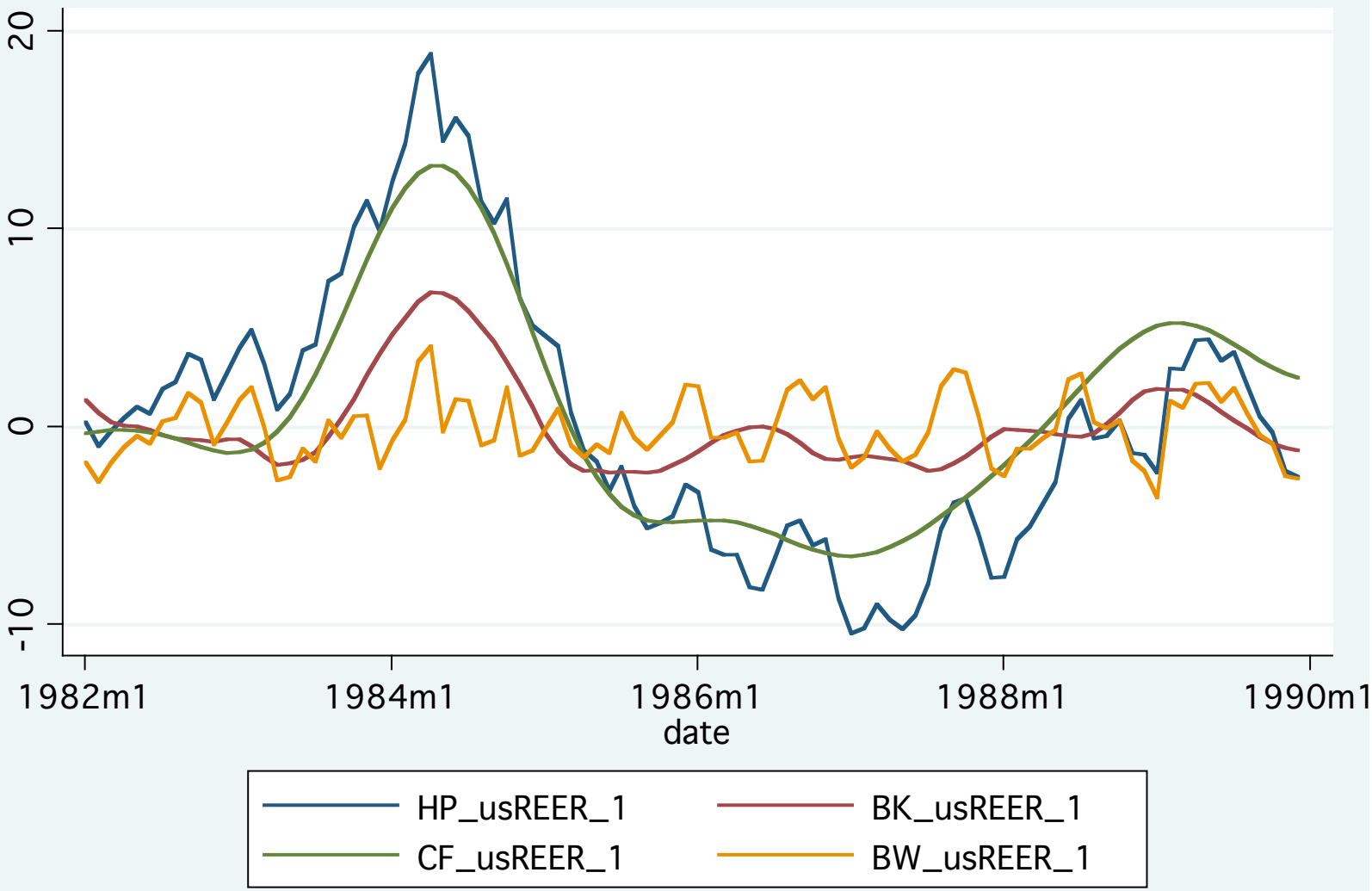

Hodrick-Prescott estimated trend, lambda=129600
80
90
100
110
120
130
1980m1
1985m1
1990m1
1995m1
2000m1
2005m1
date
US Real Effective Exchange Rate, monthly
HP_usREER_sm_1


HP-filtered US REER
HP_usREER_1
20
10
0
-10
1980m1
1985m1
1990m1
1995m1
2000m1
2005m1
date


Baxter-King filtered US REER, plo=18, phi=96
BK_usREER_1
6
4
2
0
-2
1980m1
1985m1
1990m1
1995m1
2000m1
2005m1
date


Christiano-Fitzgerald filtered US REER, plo=18, phi=96
CF_usREER_1
15
10
5
0
-5
1980m1
1985m1
1990m1
1995m1
2000m1
2005m1
date


Butterworth filtered usREER, freq=18
BW_usREER_1
4
2
0
-2
-4
1980m1
1985m1
1990m1
1995m1
2000m1
2005m1
date


Alternate time series filters for US REER, 1982-1989
20
10
0
-10
1982m1
1984m1
1986m1
1988m1
1990m1
date
HP_usREER_1
BK_usREER_1
CF_usREER_1
BW_usREER_1

Coming attractions and acknowledgements

I am testing a `bandpass` routine, based on Fitzgerald's MATLAB routine, that implements a number of alternative bandpass filters (including the Trigonometric Regression filter mentioned above). When that routine is available, it will be available from SSC as package `bandpass`.

I am grateful to Martha Lopez for the translation and testing of several of these routines from MATLAB, and to the Boston College Undergraduate Research Assistant Program for funding her efforts on this project.