Description

tsfilter separates a time series into trend and cyclical components. The trend component may contain a deterministic or a stochastic trend. The stationary cyclical component is driven by stochastic cycles at the specified periods.

Syntax

Filter one variable

tsfilter *filter* [*type*] *newvar* = *varname* [*if*] [*in*] [, *options*]

Filter multiple variables, unique names

tsfilter *filter* [*type*] *newvarlist* = *varlist* [*if*] [*in*] [, *options*]

Filter multiple variables, common name stub

tsfilter *filter* [*type*] *stub** = *varlist* [*if*] [*in*] [, *options*]

filter	Name	See
bk	Baxter–King	[TS] **tsfilter bk**
bw	Butterworth	[TS] **tsfilter bw**
cf	Christiano–Fitzgerald	[TS] **tsfilter cf**
hp	Hodrick–Prescott	[TS] **tsfilter hp**

You must tsset or xtset your data before using tsfilter; see [TS] **tsset** and [XT] **xtset**.
varname and *varlist* may contain time-series operators; see [U] **11.4.4 Time-series varlists**.
options differ across the filters and are documented in each *filter*'s manual entry.

Remarks and examples

The time-series filters implemented in tsfilter separate a time-series y_t into trend and cyclical components:

$$y_t = \tau_t + c_t$$

where τ_t is the trend component and c_t is the cyclical component. τ_t may be nonstationary; it may contain a deterministic or a stochastic trend, as discussed below.

The primary objective of the methods implemented in tsfilter is to estimate c_t, a stationary cyclical component that is driven by stochastic cycles within a specified range of periods. The trend component τ_t is calculated by the difference $\tau_t = y_t - c_t$.

Although the filters implemented in tsfilter have been widely applied by macroeconomists, they are general time-series methods and may be of interest to other researchers.

Remarks are presented under the following headings:

An example dataset

Time series are frequently filtered to remove unwanted characteristics, such as trends and seasonal components, or to estimate components driven by stochastic cycles from a specific range of periods. Although the filters implemented in `tsfilter` can be used for both purposes, their primary purpose is the latter, and we restrict our discussion to that use.

We explain the methods implemented in `tsfilter` by estimating the business-cycle component of a macroeconomic variable, because they are frequently used for this purpose. We estimate the business-cycle component of the natural log of an index of the industrial production of the United States, which is plotted below.

▷ Example 1: A trending time series

```
. use https://www.stata-press.com/data/r19/ipq
(Federal Reserve Economic Data, St. Louis Fed)
. tsline ip_ln
```


◁
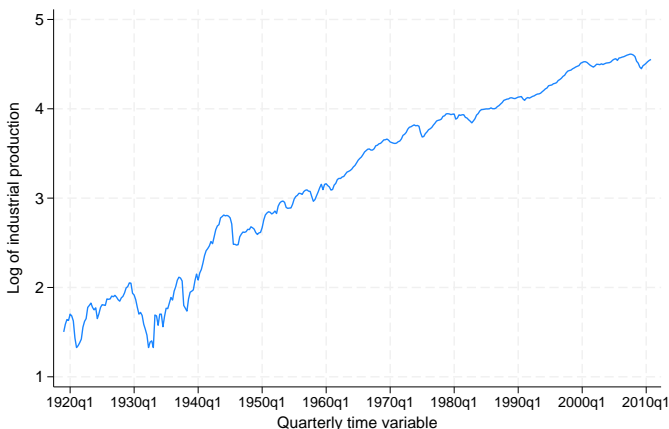
The above graph shows that `ip_ln` contains a trend component. Time series may contain deterministic trends or stochastic trends. A polynomial function of time is the most common deterministic time trend. An integrated process is the most common stochastic trend. An integrated process is a random variable that must be differenced one or more times to be stationary; see Hamilton (1994) for a discussion. The different filters implemented in `tsfilter` allow for different orders of deterministic time trends or integrated processes.

We now illustrate the four methods implemented in `tsfilter`, each of which will remove the trend and estimate the business-cycle component. Burns and Mitchell (1946) defined oscillations in business data with recurring periods between 1.5 and 8 years to be business-cycle fluctuations; we use their commonly accepted definition.

A baseline method: Symmetric moving-average (SMA) filters

Symmetric moving-average (SMA) filters form a baseline method for estimating a cyclical component because of their properties and simplicity. An SMA filter of a time series y_t, $t \in \{1, \ldots, T\}$, is the data transform defined by

$$y_t^* = \sum_{j=-q}^{q} \alpha_j y_{t-j}$$

for each $t \in \{q+1, \ldots, T-q\}$, where $\alpha_{-j} = \alpha_j$ for $j \in \{-q, \ldots, q\}$. Although the original series has T observations, the filtered series has only $T - 2q$, where q is known as the order of the SMA filter.

SMA filters with weights that sum to zero remove deterministic and stochastic trends of order 2 or less, as shown by Fuller (1996) and Baxter and King (1999).

▷ Example 2: A trend-removing SMA filter

This trend-removal property of SMA filters with coefficients that sum to zero may surprise some readers. For illustration purposes, we filter `ip_ln` by the filter

$-0.2\texttt{ip\_ln}_{t-2} - 0.2\texttt{ip\_ln}_{t-1} + 0.8\texttt{ip\_ln}_t - 0.2\texttt{ip\_ln}_{t+1} - 0.2\texttt{ip\_ln}_{t+2}$

and plot the filtered series. We do not even need `tsfilter` to implement this second-order SMA filter; we can use `generate`.

```
. generate ip_sma = -.2*L2.ip_ln-.2*L.ip_ln+.8*ip_ln-.2*F.ip_ln-.2*F2.ip_ln
(4 missing values generated)

. tsline ip_sma
```
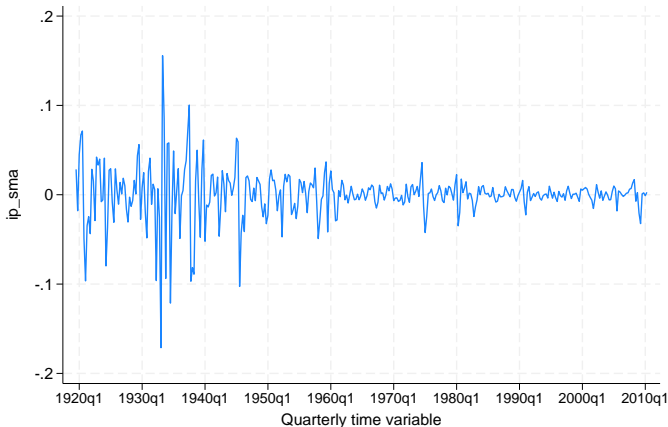

The filter has removed the trend.

There is no good reason why we chose that particular SMA filter. Baxter and King (1999) derived a class of SMA filters with coefficients that sum to zero and get as close as possible to keeping only the specified cyclical component.

An overview of filtering in the frequency domain

We need some concepts from the frequency-domain approach to time-series analysis to motivate how Baxter and King (1999) defined "as close as possible". These concepts also motivate the other filters in `tsfilter`. The intuitive explanation presented here glosses over many technical details discussed by Priestley (1981), Hamilton (1994), Fuller (1996), and Wei (2006).

As with much time-series analysis, the basic results are for covariance-stationary processes with additional results handling some nonstationary cases. We present some useful results for covariance-stationary processes and discuss how to handle nonstationary series below.

The autocovariances γ_j, $j \in \{0, 1, \ldots, \infty\}$, of a covariance-stationary process y_t specify its variance and dependence structure. In the frequency-domain approach to time-series analysis, y_t and the autocovariances are specified in terms of independent stochastic cycles that occur at frequencies $\omega \in [-\pi, \pi]$. The spectral density function $f_y(\omega)$ specifies the contribution of stochastic cycles at each frequency ω relative to the variance of y_t, which is denoted by σ_y^2. The variance and the autocovariances can be expressed as an integral of the spectral density function. Formally,

$$\gamma_j = \int_{-\pi}^{\pi} e^{i\omega j} f_y(\omega) d\omega \tag{1}$$

where i is the imaginary number $i = \sqrt{-1}$.

Equation (1) can be manipulated to show what fraction of the variance of y_t is attributable to stochastic cycles in a specified range of frequencies. Hamilton (1994, 156) discusses this point in more detail.

Equation (1) implies that if $f_y(\omega) = 0$ for $\omega \in [\omega_1, \omega_2]$, then stochastic cycles at these frequencies contribute zero to the variance and autocovariances of y_t.

The goal of time-series filters is to transform the original series into a new series y_t^* for which the spectral density function of the filtered series $f_{y^*}(\omega)$ is zero for unwanted frequencies and equal to $f_y(\omega)$ for desired frequencies.

A linear filter of y_t can be written as

$$y_t^* = \sum_{j=-\infty}^{\infty} \alpha_j y_{t-j} = \alpha(L) y_t$$

where we let y_t be an infinitely long series as required by some of the results below. To see the impact of the filter on the components of y_t at each frequency ω, we need an expression for $f_{y^*}(\omega)$ in terms of $f_y(\omega)$ and the filter weights α_j. Wei (2006, 282) shows that for each ω,

$$f_{y^*}(\omega) = |\alpha(e^{i\omega})|^2 f_y(\omega) \tag{2}$$

where $|\alpha(e^{i\omega})|$ is known as the gain of the filter. Equation (2) makes explicit that the squared gain function $|a(e^{i\omega})|^2$ converts the spectral density of the original series, $f_y(\omega)$, into the spectral density of the filtered series, $f_{y^*}(\omega)$. In particular, (2) says that for each frequency ω, the spectral density of the filtered series is the product of the square of the gain of the filter and the spectral density of the original series.

As we will see in the examples below, the gain function provides a crucial interpretation of what a filter is doing. We want a filter for which $f_{y^*}(\omega) = 0$ for unwanted frequencies and for which $f_{y^*}(\omega) = f_y(\omega)$ for desired frequencies. So we seek a filter for which the gain is 0 for unwanted frequencies and for which the gain is 1 for desired frequencies.

In practice, we cannot find such an ideal filter exactly, because the constraints an ideal filter places on filter coefficients cannot be satisfied for time series with only a finite number of observations. The expansive literature on filters is a result of the tradeoffs involved in designing implementable filters that approximate the ideal filter.

Ideally, filters pass or block the stochastic cycles at specified frequencies by having a gain of 1 or 0. Band-pass filters, such as the Baxter–King (BK) and the Christiano–Fitzgerald (CF) filters, pass through stochastic cycles in the specified range of frequencies and block all the other stochastic cycles. High-pass filters, such as the Hodrick–Prescott (HP) and Butterworth filters, only allow the stochastic cycles at or above a specified frequency to pass through and block the lower-frequency stochastic cycles. For band-pass filters, let $[\omega_0, \omega_1]$ be the set of desired frequencies with all other frequencies being undesired. For high-pass filters, let ω_0 be the cutoff frequency with only those frequencies $\omega \geq \omega_0$ being desired.

SMA revisited: The Baxter–King filter

We now return to the class of SMA filters with coefficients that sum to zero and get as close as possible to keeping only the specified cyclical component as derived by Baxter and King (1999).

For an infinitely long series, there is an ideal band-pass filter for which the gain function is 1 for $\omega \in [\omega_0, \omega_1]$ and 0 for all other frequencies. It just so happens that this ideal band-pass filter is an SMA filter with coefficients that sum to zero. Baxter and King (1999) derive the coefficients of this ideal band-pass filter and then define the BK filter to be the SMA filter with $2q + 1$ terms that are as close as possible to those of the ideal filter. There is a tradeoff in choosing q: larger values of q cause the gain of the BK filter to be closer to the gain of the ideal filter, but larger values also increase the number of missing observations in the filtered series.

Although the mathematics of the frequency-domain approach to time-series analysis is in terms of stochastic cycles at frequencies $\omega \in [-\pi, \pi]$, applied work is generally in terms of periods p, where $p = 2\pi/\omega$. So the options for the `tsfilter` subcommands are in terms of periods.

▷ Example 3: A BK estimate of the business-cycle component

Below we use `tsfilter bk`, which implements the BK filter, to estimate the business-cycle component composed of stochastic cycles between 6 and 32 periods, and then we graph the estimated component.

```
. tsfilter bk ip_bk = ip_ln, minperiod(6) maxperiod(32)
. tsline ip_bk
```
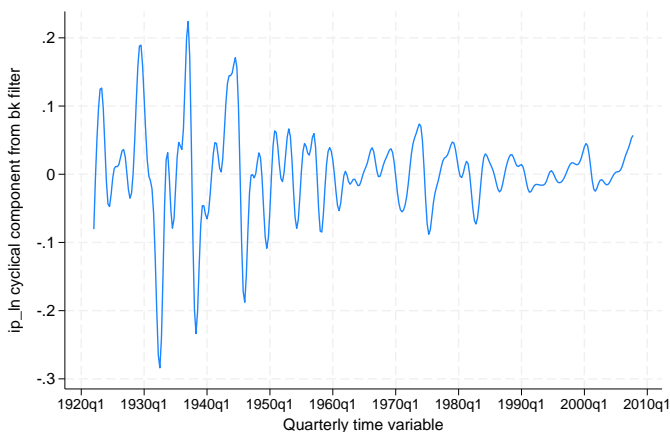

The above graph tells us what the estimated business-cycle component looks like, but it presents no evidence as to how well we have estimated the component. A periodogram is better for this purpose. A periodogram is an estimator of a transform of the spectral density function; see [TS] **pergram** for details. Below we plot the periodogram for the BK estimate of the business-cycle component. `pergram` displays the results in natural frequencies, which are the standard frequencies divided by 2π. We use the `xline()` option to draw vertical lines at the lower natural-frequency cutoff $(1/32 = 0.03125)$ and the upper natural-frequency cutoff $(1/6 \approx 0.16667)$.

```
. pergram ip_bk, xline(0.03125 0.16667)
```
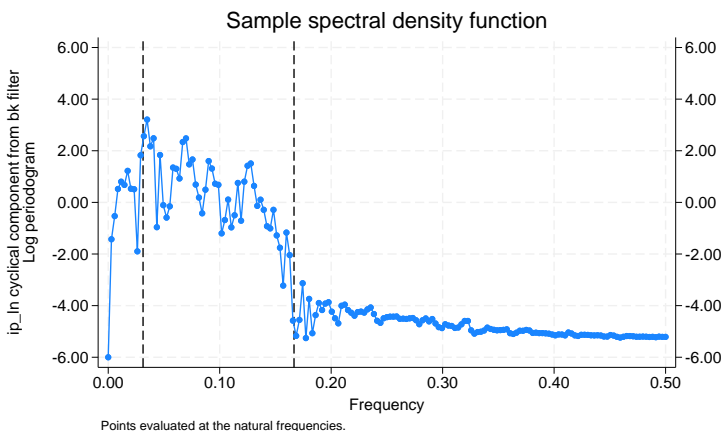

If the filter completely removed the stochastic cycles corresponding to the unwanted frequencies, the periodogram would be a flat line at the minimum value of -6 outside the range identified by the vertical lines. That the periodogram takes on values greater than -6 outside the specified range indicates the inability of the BK filter to pass through only stochastic cycles at frequencies inside the specified band.

We can also evaluate the BK filter by plotting its gain function against the gain function of an ideal filter. In the output below, we reestimate the business-cycle component to store the gain of the BK filter for the specified parameters. (The coefficients and the gain of the BK filter are completely determined by the specified minimum period, the maximum period, and the order of the SMA filter.) We label the variable bkgain for the graph below.

```
. drop ip_bk
. tsfilter bk ip_bk = ip_ln, minperiod(6) maxperiod(32) gain(bkgain abk)
. label variable bkgain "BK filter"
```

Below we generate ideal, the gain function of the ideal band-pass filter at the frequencies f. Then we plot the gain of the ideal filter and the gain of the BK filter.

```
. generate f = _pi*(_n-1)/_N
. generate ideal = cond(f<_pi/16, 0, cond(f<_pi/3, 1,0))
. label variable ideal "Ideal filter"
. twoway line ideal f || line bkgain abk
```
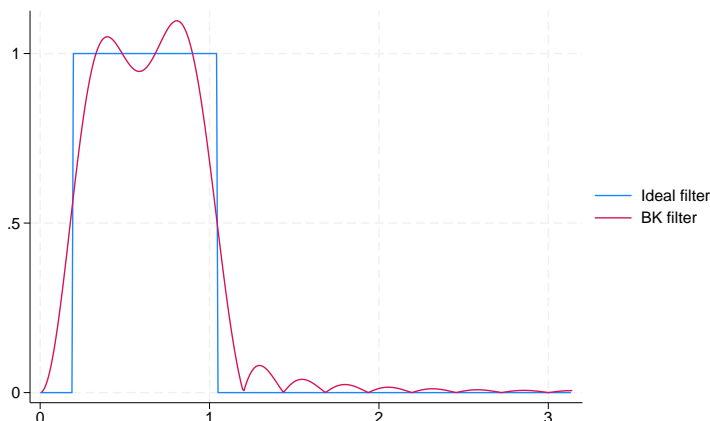

The graph reveals that the gain of the BK filter deviates markedly from the square-wave gain of the ideal filter. Increasing the symmetric moving average via the smaorder() option will cause the gain of the BK filter to more closely approximate the gain of the ideal filter at the cost of lost observations in the filtered series.

◁

Filtering a random walk: The Christiano–Fitzgerald filter

Although Baxter and King (1999) minimized the error between the coefficients in their filter and the ideal band-pass filter, Christiano and Fitzgerald (2003) minimized the mean squared error between the estimated component and the true component, assuming that the raw series is a random-walk process. Christiano and Fitzgerald (2003) give three important reasons for using their filter:

1. The true dependence structure of the data affects which filter is optimal.

2. Many economic time series are well approximated by random-walk processes.

3. Their filter does a good job passing through stochastic cycles of desired frequencies and blocking stochastic cycles from unwanted frequencies on a range of processes that are close to being a random-walk process.

The CF filter obtains its optimality properties at the cost of an additional parameter that must be estimated and a loss of robustness. The CF filter is optimal for a random-walk process. If the true process is a random walk with drift, then the drift term must be estimated and removed; see [TS] **tsfilter cf** for details. The CF filter is not symmetric, so it will not remove second-order deterministic or second-order integrated processes. `tsfilter cf` also implements another filter that Christiano and Fitzgerald (2003) derived that is an SMA filter with coefficients that sum to zero. This filter is designed to be as close as possible to the random-walk optimal filter under the constraint that it be an SMA filter with constraints that sum to zero; see [TS] **tsfilter cf** for details.

❑ Technical note

A random-walk process is a first-order integrated process; it must be differenced once to produce a stationary process. Formally, a random-walk process is given by $y_t = y_{t-1} + \epsilon_t$, where ϵ_t is a zero-mean stationary random variable. A random-walk-plus-drift process is given by $\tilde{y}_t = \mu + \tilde{y}_{t-1} + \epsilon_t$, where ϵ_t is a zero-mean stationary random variable.

❑

▷ Example 4: A CF estimate of the business-cycle component

In this example, we use the CF filter to estimate the business-cycle component, and we plot the periodogram of the CF estimates. We specify the `drift` option because `ip_ln` is well approximated by a random-walk-plus-drift process.

```
. tsfilter cf ip_cf = ip_ln, minperiod(6) maxperiod(32) drift
. pergram ip_cf, xline(0.03125 0.16667)
```


Sample spectral density function

Points evaluated at the natural frequencies.
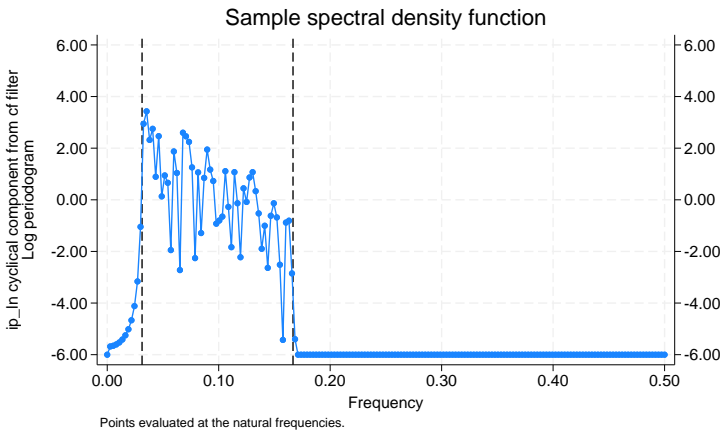
The periodogram of the CF estimates of the business-cycle component indicates that the CF filter did a better job than the BK filter of passing through only the desired stochastic cycles. Given that `ip_ln` is well approximated by a random-walk-plus-drift process, the relative performance of the CF filter is not surprising.

As with the BK filter, plotting the gain of the CF filter and the gain of the ideal filter gives an impression of how well the filter isolates the specified components. In the output below, we reestimate the business-cycle component, using the `gain()` option to store the gain of the CF filter, and we plot the gain functions.

```
. drop ip_cf
. tsfilter cf ip_cf = ip_ln, minperiod(6) maxperiod(32) drift gain(cfgain acf)
. label variable cfgain "CF filter"
. twoway line ideal f || line cfgain acf
```


Comparing this graph with the graph of the BK gain function reveals that the CF filter is closer to the gain of the ideal filter than is the BK filter. The graph also reveals that the gain of the CF filter oscillates above and below 1 for desired frequencies.

◁

The choice between the BK or the CF filter is one between robustness or efficiency. The BK filter handles a broader class of stochastic processes, but the CF filter produces a better estimate of c_t if y_t is close to a random-walk process or a random-walk-plus-drift process.

## A one-parameter high-pass filter: The Hodrick–Prescott filter

Hodrick and Prescott (1997) motivated the Hodrick–Prescott (HP) filter as a trend-removal technique that could be applied to data that came from a wide class of data-generating processes. In their view, the technique specified a trend in the data, and the data were filtered by removing the trend. The smoothness of the trend depends on a parameter λ. The trend becomes smoother as $\lambda \to \infty$. Hodrick and Prescott (1997) recommended setting λ to 1,600 for quarterly data.

King and Rebelo (1993) showed that removing a trend estimated by the HP filter is equivalent to a high-pass filter. They derived the gain function of this high-pass filter and showed that the filter would make integrated processes of order 4 or less stationary, making the HP filter comparable with the band-pass filters discussed above.

▷ Example 5: An HP estimate of the business-cycle component

We begin by applying the HP high-pass filter to ip_ln and plotting the periodogram of the estimated business-cycle component. We specify the gain() option because will use the gain of the filter in the next example.

```
. tsfilter hp ip_hp = ip_ln, gain(hpg1600 ahp1600)
. label variable hpg1600 "HP(1600) filter"
. pergram ip_hp, xline(0.03125)
```


Points evaluated at the natural frequencies.
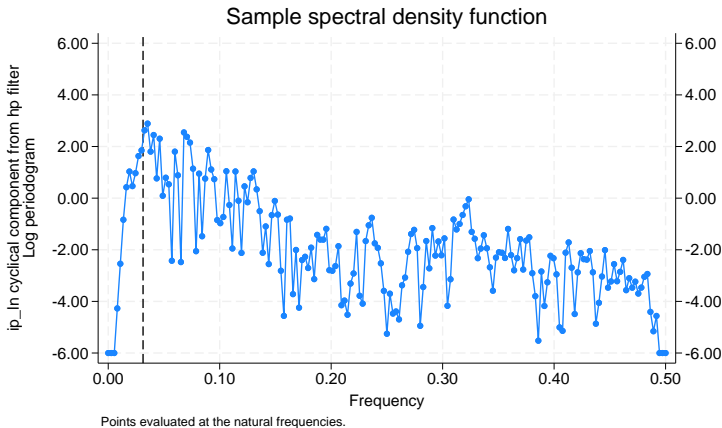
Because the HP filter is a high-pass filter, the high-frequency stochastic cycles corresponding to those periods below 6 remain in the estimated component. Of more concern is the presence of the low-frequency stochastic cycles that the filter should remove. We address this issue in the example below.

◁

▷ Example 6: Choosing the parameters for the HP filter

Hodrick and Prescott (1997) argued that the smoothing parameter λ should be 1,600 on the basis of a heuristic argument that specified values for the variance of the cyclical component and the variance of the second difference of the trend component, both recorded at quarterly frequencies. In this example, we choose the smoothing parameter to be 677.13, which sets the gain of the filter to 0.5 at the frequency corresponding to 32 periods, as explained in the technical note below. We then plot the periodogram of the filtered series.

```
. tsfilter hp ip_hp2 = ip_ln, smooth(677.13)  gain(hpg677 ahp677)
. label variable hpg677 "HP(677) filter"
. pergram ip_hp, xline(0.03125)
```


Sample spectral density function

Points evaluated at the natural frequencies.
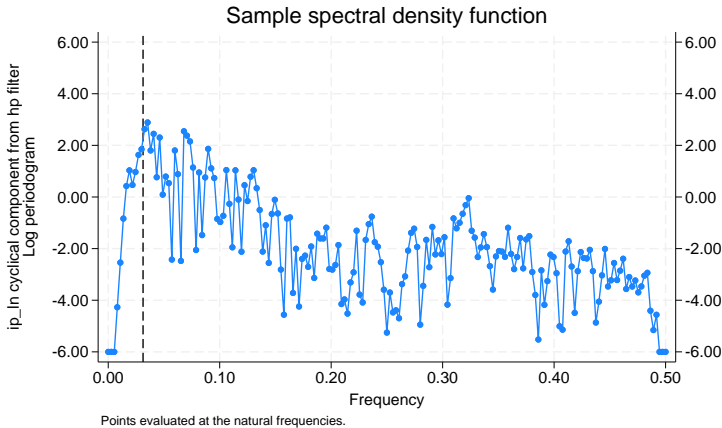
Although the periodogram looks better than the periodogram with the default smoothing, the HP filter still did not zero out the low-frequency stochastic cycles as well as the CF filter did. We take another look at this issue by plotting the gain functions for these filters along with the gain function from the ideal band-pass filter.

```
. twoway line ideal f || line hpg677 ahp677
```
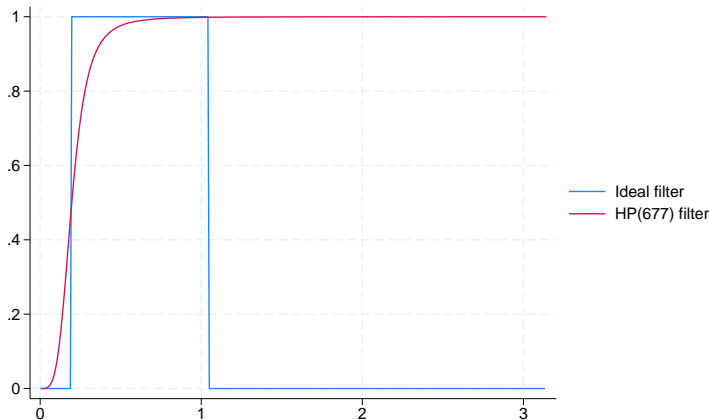

Comparing the gain graphs reveals that the gain of the CF filter is closest to the gain of the ideal filter. Both the BK and the HP filters allow some low-frequency stochastic cycles to pass through. The plot also illustrates that the HP filter is a high-pass filter because its gain is 1 for those stochastic cycles at frequencies above 6 periods, whereas the other gain functions go to zero.

❑ Technical note

Conventionally, economists have used $\lambda = 1600$, which Hodrick and Prescott (1997) recommended for quarterly data. Ravn and Uhlig (2002) derived values for λ at monthly and annual frequencies that are rescalings of the conventional $\lambda = 1600$ for quarterly data. These heuristic values are the default values; see [TS] **tsfilter hp** for details. In the filter literature, filter parameters are set as functions of the cutoff frequency; see Pollock (2000, 324), for instance. This method finds the filter parameter that sets the gain of the filter equal to $1/2$ at the cutoff frequency. Applying this method to selecting λ at the cutoff frequency of 32 periods requires solving

$$1/2 = \frac{4\lambda \left\{1 - \cos(2\pi/32)\right\}^2}{1 + 4\lambda \left\{1 - \cos(2\pi/32)\right\}^2}$$

for λ, which yields $\lambda \approx 677.13$, which was used in the previous example.

The gain function of the HP filter is a function of the parameter λ, and λ sets both the location of the cutoff frequency and the slope of the gain function. The graph below illustrates this dependence by plotting the gain function of the HP filter for λ set to 10, 677.13, and 1,600 along with the gain function for the ideal band-pass filter with cutoff periods of 32 periods and 6 periods.

❑
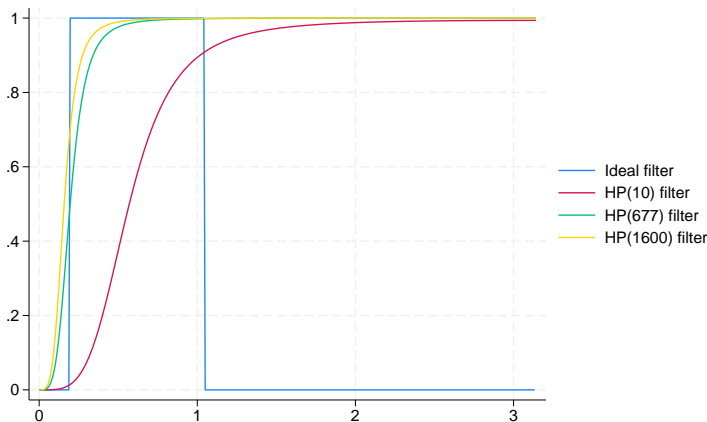
A two-parameter high-pass filter: The Butterworth filter

Engineers have used Butterworth filters for a long time because they are "maximally flat". The gain functions of these filters are as close as possible to being a flat line at 0 for the unwanted periods and a flat line at 1 for the desired periods; see Butterworth (1930) and Bianchi and Sorrentino (2007, 17–20).

Pollock (2000) showed that Butterworth filters can be derived from some axioms that specify properties we would like a filter to have. Although the Butterworth and BK filters share the properties of symmetry and phase neutrality, the coefficients of Butterworth filters do not need to sum to zero. (Phase-neutral filters do not shift the signal forward or backward in time; see Pollock [1999].) Although the BK filter relies on the detrending properties of SMA filters with coefficients that sum to zero, Pollock (2000) shows that Butterworth filters have detrending properties that depend on the filters' parameters.

tsfilter bw implements the high-pass Butterworth filter using the computational method that Pollock (2000) derived. This filter has two parameters: the cutoff period and the order of the filter denoted by m. The cutoff period sets the location where the gain function starts to filter out the high-period (low-frequency) stochastic cycles, and m sets the slope of the gain function for a given cutoff period. For a given cutoff period, the slope of the gain function at the cutoff period increases with m. For a given m, the slope of the gain function at the cutoff period increases with the cutoff period.

We cannot obtain a vertical slope at the cutoff frequency, which is the ideal, because the computation becomes unstable; see Pollock (2000). The m for which the computation becomes unstable depends on the cutoff period.

Pollock (2000) and Gómez (1999) argue that the additional flexibility produced by the additional parameter makes the high-pass Butterworth filter a better filter than the HP filter for estimating the cyclical components.

Pollock (2000) shows that the high-pass Butterworth filter can estimate the desired components of the dth difference of a dth-order integrated process as long as $m \geq d$.

▷ Example 7: A Butterworth filter that removes low-frequency components

Below we use tsfilter bw to estimate the components driven by stochastic cycles greater than 32 periods using Butterworth filters of order 2 and order 6. We also compute, label, and plot the gain functions for each filter.

```
. tsfilter bw ip_bw1 = ip_ln, gain(bwgain1 abw1) maxperiod(32) order(2)
. label variable bwgain1 "BW 2"
. tsfilter bw ip_bw6 = ip_ln, gain(bwgain6 abw6) maxperiod(32) order(6)
. label variable bwgain6 "BW 6"
. twoway line ideal f || line bwgain1 abw1 || line bwgain6 abw6
```
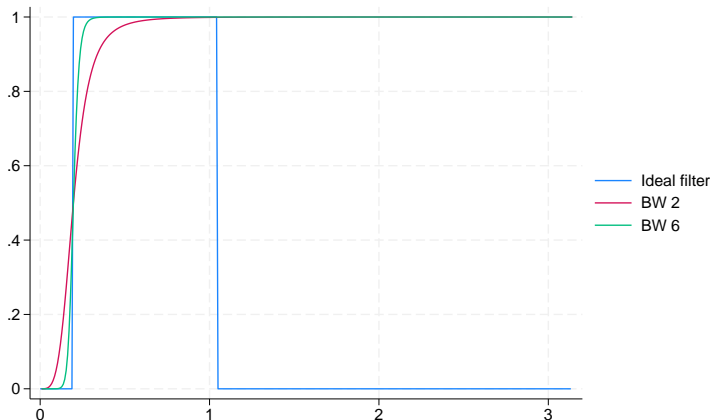

The graph illustrates that the slope of the gain function increases with the order of the filter.

The graph below provides another perspective by plotting the gain function from the ideal band-pass filter on a graph with plots of the gain functions from the Butterworth filter of order 6, the CF filter, and the HP(677) filter.

```
. twoway line ideal f || line bwgain6 abw6 || line cfgain acf
> || line hpg677 ahp677
```
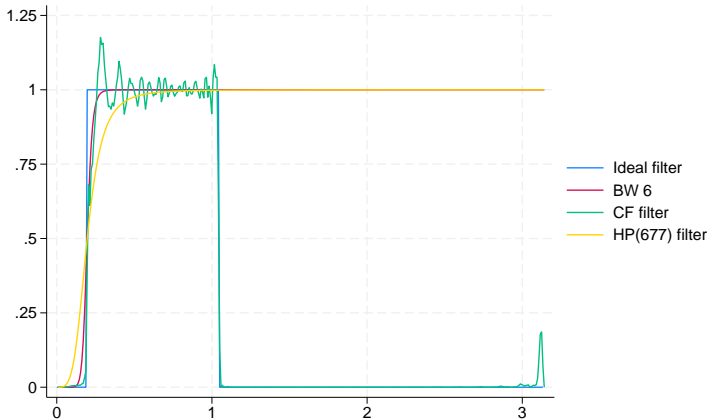

Although the slope of the gain function from the CF filter is closer to being vertical at the cutoff frequency, the gain function of the Butterworth filter does not oscillate above and below 1 after it first reaches the value of 1. The flatness of the Butterworth filter below and above the cutoff frequency is not an accident; it is one of the filter's properties.

◁

▷ Example 8: A Butterworth filter that removes high-frequency components

In the previous example, we used the Butterworth filter of order 6 to remove low-frequency stochastic cycles, and we saved the results in ip_bw6. The Butterworth filter did not address the high-frequency stochastic cycles below 6 periods because it is a high-pass filter. We remove those high-frequency stochastic cycles in this example by keeping the trend produced by refiltering the previously filtered series.

This example uses a common trick: keeping the trend produced by a high-pass filter turns that high-pass filter into a low-pass filter. Because we want to remove the high-frequency stochastic cycles still in the previously filtered series ip_bw6, we need a low-pass filter. So we keep the trend produced by refiltering the previously filtered series.

In the output below, we apply a Butterworth filter of order 20 to the previously filtered series ip_bw6. We explain why we used order 20 in the next example. We specify the trend() option to keep the low-frequency components from these filters. Then we compute and graph the periodogram for the trend variable.

```
. tsfilter bw ip_bwu20 = ip_bw6, gain(bwg20 fbw20) maxperiod(6) order(20)
> trend(ip_bwb)

. label variable bwg20 "BW upper filter 20"

. pergram ip_bwb, xline(0.03125 0.16667)
```

Sample spectral density function

Points evaluated at the natural frequencies.
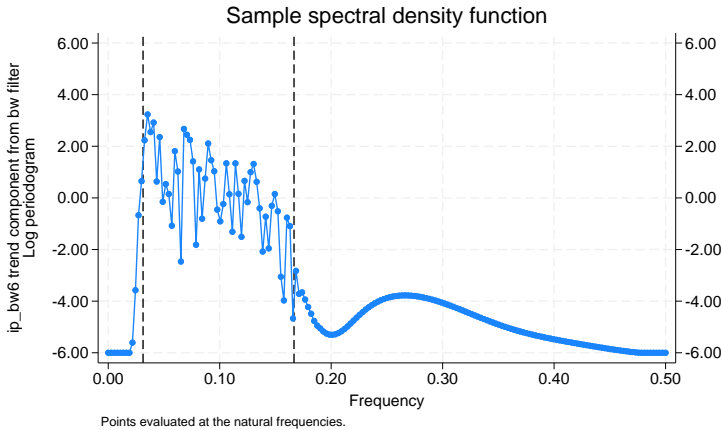
The periodogram reveals that the two-pass process has passed the original series ip_ln through a band-pass filter. It also reveals that the two-pass process did a reasonable job of filtering out the stochastic cycles corresponding to the unwanted frequencies.

◁

▷ Example 9: Choosing the order of a Butterworth filter

In the previous example, when the cutoff period was 6, we set the order of the Butterworth filter to 20. In contrast, in example 7, when the cutoff period was 32, we set the order of the Butterworth filter to 6. We had to increase filter order because the slope of the gain function of the Butterworth filter is increasing with the cutoff period. We needed a larger filter order to get an acceptable slope at the lower cutoff period.

We illustrate this point in the output below. We apply Butterworth filters of orders 1 and 6 to the previously filtered series ip_bw6, we compute the gain functions, we label the gain variables, and then we plot the gain functions from the ideal filter and the Butterworth filters.

```
. tsfilter bw ip_bwu1  = ip_bw6, gain(bwg1 fbw1)    maxperiod(6) order(2)
. label variable bwg1 "BW upper filter 2"
. tsfilter bw ip_bwu6  = ip_bw6, gain(bwg6 fbw6)    maxperiod(6) order(6)
. label variable bwg6 "BW upper filter 6"
. twoway line ideal f || line bwg1 fbw1 || line bwg6 fbw6 || line bwg20 fbw20
```
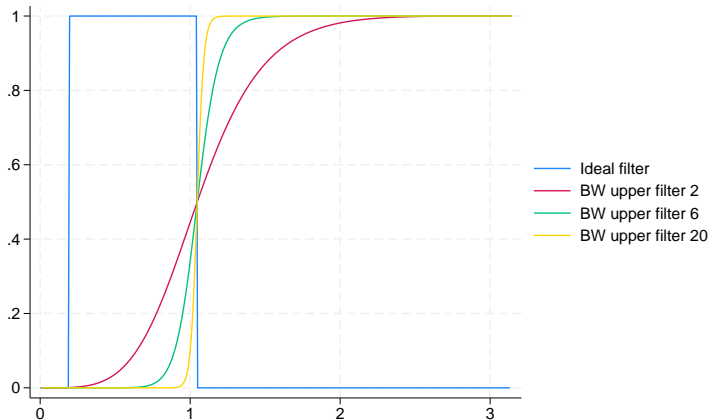

Because the cutoff period is 6, the gain functions for $m = 2$ and $m = 6$ are much flatter than the gain functions for $m = 2$ and $m = 6$ in example 7 when the cutoff period was 32. The gain function for $m = 20$ is reasonably close to vertical, so we used it in example 8. We mentioned above that for any given cutoff period, the computation eventually becomes unstable for larger values of m. For instance, when the cutoff period is 32, $m = 20$ is not numerically feasible.

◁

▷ Example 10: Comparing the Butterworth and CF estimates

As a conclusion, we plot the business-cycle components estimated by the CF filter and by the two passes of Butterworth filters. The shaded areas identify recessions. The two estimates are close but the differences could be important. Which estimate is better depends on whether the oscillations around 1 in the graph of the CF gain function (the second graph of example 7) cause more problems than the nonvertical slopes at the cutoff periods that occur in the BW6 gain function of that same graph and the BW upper filter 20 gain function graphed above.
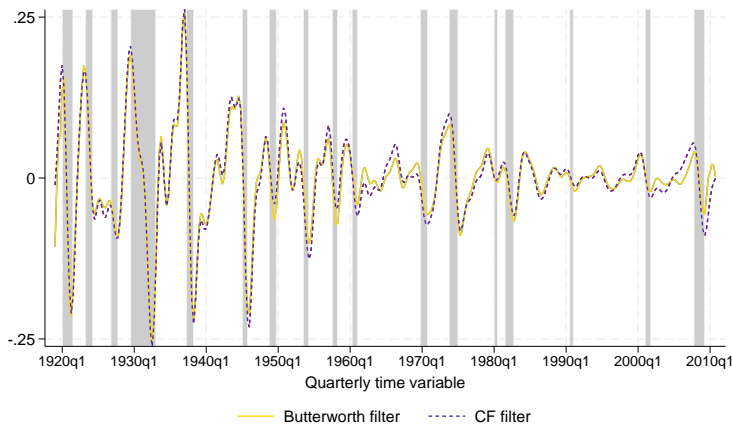
There is a long tradition in economics of using models to estimate components. Instead of comparing filters by their gain functions, some authors compare filters by finding underlying models for which the filter parameters are the model parameters. For instance, Harvey and Jaeger (1993), Gómez (1999, 2001), Pollock (2000, 2006), and Harvey and Trimbur (2003) derive models that correspond to the HP or the Butterworth filter. Some of these references also compare components estimated by filters with components estimated by making predictions from estimated models. In effect, these references point out that arima, dfactor, sspace, and ucm (see [TS] **arima**, [TS] **dfactor**, [TS] **sspace**, and [TS] **ucm**) implement alternative methods to component estimation.

Methods and formulas

All filters work with both time-series data and panel data when there are many observations on each panel. When used with panel data, the calculations are performed separately within each panel.

For these filters, the default minimum and maximum periods of oscillation correspond to the boundaries used by economists (Burns and Mitchell 1946) for business cycles. Burns and Mitchell defined business cycles as oscillations in business data with recurring periods between 1.5 and 8 years. Their definition continues to be cited by economists investigating correlations between business cycles.

If y_t is a time series, then the cyclical component is

$$c_t = B(L)y_t = \sum_{j=-\infty}^{\infty} b_j y_{t-j}$$

where b_j are the coefficients of the impulse–response sequence of some ideal filter. The impulse–response sequence is the inverse Fourier transform of either a square wave or step function depending upon whether the filter is a band-pass or high-pass filter, respectively.

In finite sequences, it is necessary to approximate this calculation with a finite impulse–response sequence $\widehat{b}_j$:

$$\widehat{c}_t = \widehat{B}_t(L)y_t = \sum_{j=-n_1}^{n_2} \widehat{b}_j y_{t-j}$$

The infinite-order impulse–response sequence for the filters implemented in `tsfilter` are symmetric and time-invariant.

In the frequency domain, the relationships between the true cyclical component and its finite estimates respectively are

$$c(\omega) = B(\omega)y(\omega)$$

and

$$\widehat{c}(\omega) = \widehat{B}(\omega)y(\omega)$$

where $B(\omega)$ and $\widehat{B}(\omega)$ are the frequency transfer functions of the filters B and $\widehat{B}$.

The frequency transfer function for $B(\omega)$ can be expressed in polar form as

$$B(\omega) = |B(\omega)|\exp\{i\theta(\omega)\}$$

where $|B(\omega)|$ is the filter's gain function and $\theta(\omega)$ is the filter's phase function. The gain function determines whether the amplitude of the stochastic cycle is increased or decreased at a particular frequency. The phase function determines how a cycle at a particular frequency is shifted forward or backward in time.

In this form, it can be shown that the spectrum of the cyclical component, $f_c(\omega)$, is related to the spectrum of y_t series by the squared gain:

$$f_c(\omega) = |B(\omega)|^2 f_y(\omega)$$

Each of the four filters in `tsfilter` has an option for returning an estimate of the gain function together with its associated scaled frequency $a = \omega/\pi$, where $0 \leq \omega \leq \pi$. These are consistent estimates of $|B(\omega)|$, the gain from the ideal linear filter.

The band-pass filters implemented in `tsfilter`, the BK and CF filters, use a square wave as the ideal transfer function:

$$B(\omega) = \begin{cases} 1 & \text{if } |\omega| \in [\omega_l, \omega_h] \\ 0 & \text{if } |\omega| \notin [\omega_l, \omega_h] \end{cases}$$

The high-pass filters, the Hodrick–Prescott and Butterworth filters, use a step function as the ideal transfer function:

$$B(\omega) = \begin{cases} 1 & \text{if } |\omega| \geq \omega_h \\ 0 & \text{if } |\omega| < \omega_h \end{cases}$$

Acknowledgments

References

Baxter, M., and R. G. King. 1999. Measuring business cycles: Approximate band-pass filters for economic time series. *Review of Economics and Statistics* 81: 575–593. https://doi.org/10.1162/003465399558454.

Bianchi, G., and R. Sorrentino. 2007. *Electronic Filter Simulation and Design*. New York: McGraw–Hill.

Burns, A. F., and W. C. Mitchell. 1946. *Measuring Business Cycles*. New York: National Bureau of Economic Research.

Butterworth, S. 1930. On the theory of filter amplifiers. *Experimental Wireless and the Wireless Engineer* 7: 536–541.

Christiano, L. J., and T. J. Fitzgerald. 2003. The band pass filter. *International Economic Review* 44: 435–465. https://doi.org/10.1111/1468-2354.t01-1-00076.

Fuller, W. A. 1996. *Introduction to Statistical Time Series*. 2nd ed. New York: Wiley.

Gómez, V. 1999. Three equivalent methods for filtering finite nonstationary time series. *Journal of Business and Economic Statistics* 17: 109–116. https://doi.org/10.1080/07350015.1999.10524800.

———. 2001. The use of Butterworth filters for trend and cycle estimation in economic time series. *Journal of Business and Economic Statistics* 19: 365–373. https://doi.org/10.1198/073500101681019909.

Hamilton, J. D. 1994. *Time Series Analysis*. Princeton, NJ: Princeton University Press. https://doi.org/10.2307/j.ctv14jx6sm.

Harvey, A. C., and A. Jaeger. 1993. Detrending, stylized facts and the business cycle. *Journal of Applied Econometrics* 8: 231–247. https://doi.org/10.1002/jae.3950080302.

Harvey, A. C., and T. M. Trimbur. 2003. General model-based filters for extracting cycles and trends in economic time series. *Review of Economics and Statistics* 85: 244–255. https://doi.org/10.1162/003465303765299774.

Hodrick, R. J., and E. C. Prescott. 1997. Postwar U.S. business cycles: An empirical investigation. *Journal of Money, Credit, and Banking* 29: 1–16. https://doi.org/10.2307/2953682.

King, R. G., and S. T. Rebelo. 1993. Low frequency filtering and real business cycles. *Journal of Economic Dynamics and Control* 17: 207–231. https://doi.org/10.1016/S0165-1889(06)80010-2.

Leser, C. E. V. 1961. A simple method of trend construction. *Journal of the Royal Statistical Society*, B ser., 23: 91–107. https://doi.org/10.1111/j.2517-6161.1961.tb00393.x.

Pollock, D. S. G. 1999. *A Handbook of Time-Series Analysis, Signal Processing and Dynamics*. London: Academic Press.

———. 2000. Trend estimation and de-trending via rational square-wave filters. *Journal of Econometrics* 99: 317–334. https://doi.org/10.1016/S0304-4076(00)00028-2.

———. 2006. Econometric methods of signal extraction. *Computational Statistics and Data Analysis* 50: 2268–2292. https://doi.org/10.1016/j.csda.2005.07.010.

Priestley, M. B. 1981. *Spectral Analysis and Time Series*. London: Academic Press.

Ravn, M. O., and H. Uhlig. 2002. On adjusting the Hodrick–Prescott filter for the frequency of observations. *Review of Economics and Statistics* 84: 371–376. https://doi.org/10.1162/003465302317411604.

Wei, W. W. S. 2006. *Time Series Analysis: Univariate and Multivariate Methods*. 2nd ed. Boston: Pearson.

Also see

[TS] **tsset** — Declare data to be time-series data

[TS] **tssmooth** — Smooth and forecast univariate time-series data

[XT] **xtset** — Declare data to be panel data