

Univariate Detrending Using SAS: Some examples and applications

Ban Chuan Cheah

April 26, 2006

1 Univariate Detrending

1.1 Introduction

Detrending or filtering is an important tool for the macroeconomist studying business cycle fluctuations and the effects of fiscal and monetary policies on output. Business cycles can be thought of as deviations of output from its trend. As such, detrending or filtering allows us to focus solely on the 'cyclical' properties of the time series, in this case, output. The effects of monetary or fiscal stimulus are the results of attempts to dampen the cycles. Analysis of policies are therefore framed in terms of what would have happened to the cyclical component of output under different scenarios.

Whether the cyclical component obtained through statistical detrending methods actually represent 'business cycles' in the sense of Burns and Mitchell (1946) remains a point of debate. In fact, it is not entirely clear how 'business cycles' are defined.¹ The Burns and Mitchell tradition of analyzing cycles using peak to peak or trough to trough time points of aggregate economic activity appears to have become less popular. Burns and Mitchell then go on to measure duration and amplitudes of the cycles, for instance. The tradition is still alive in that the approach is still used by the NBER Business Cycle Dating Committee.² Researchers use these as an informal check of the statistical approach by comparing the NBER reference dates with the results of the cyclical component of the detrended series. Large negative deviations from the trend should coincide with reference date of recessions or trough identified by the NBER committee and large positive deviations should coincide with the peak reference date.

Four methods of detrending will be presented – Beveridge-Nelson decomposition, Hodrick-Prescott filter, Baxter-King filter, and Unobserved Components. These are by no means a complete list of filters but have emerged in recent years to be the most popular choices. The main reason that they have become widely used is not necessarily because they are the optimal filters (optimal in the sense that they isolate the components of output that are truly cyclical), but mostly because they are easy to implement either in terms of code or because they have become part of popular statistical packages. None of the methods that will be discussed here are free from controversy. The main direction of economic analyses today is a call for more sensitivity analysis and robustness of results to changes in assumptions. These can easily be implemented in SAS.

Two applications of univariate detrending are then presented. The first is to produce 'stylized facts' of the cyclical components of the time series of interest. These stylized facts consist of a set of numbers: volatility, mean, and cross-correlations of the detrended series. We will see how these stylized facts can change in the presence of different methods of detrending. These empirical regularities are the yardstick against which the performance of many numerical models are held up against. Numerical business cycle models are calibrated to see how well they match these 'stylized facts'. The second is to use the detrended

¹See Zarnowitz (1993).

²<http://www.nber.org/cycles.html/>.

series to examine the consequences of monetary policy on the output gap or the cyclical component of output. Again, we will examine the consequences of using different filters as well as to examine the Hodrick-Prescott filter in detail. The Hodrick-Prescott filter has come under close scrutiny but appears as to have stood the test of time and has almost become the default filtering method. As such, it is important to understand some of the controversial issues around this method of detrending.

1.2 Detrending methods

This section will present four methods. In addition to the Beveridge-Nelson decomposition, Hodrick-Prescott filter, Baxter-King filter, and Unobserved Components Model, there are other filters that are easy to implement. The linear time trend and first difference filter are easy to implement using either PROC REG, PROC AUTOREG, PROC FORECAST or PROC ARIMA. Two sided moving averages can be implemented using PROC EXPAND.

1.2.1 Beveridge-Nelson decomposition

A series y_t can be decomposed into a trend and cyclical component. The Beveridge-Nelson decomposition separates a time series into a permanent and transitory component. The implications of this is the following: A one unit shock at time t which is 0 for all $t, t > t + 1$ results in a permanent change to the series if it affects the permanent component, while changes to the series will damp out over time if the shock affects the transitory component. The series, y_t can be decomposed into a permanent and transitory component as follows:

$$y_t = P_t + T_t$$

It is assumed that y_t is an ARIMA(p,1,q) process. Then,

$$\Delta y_t = \Delta P_t + \Delta T_t$$

The first difference of an ARIMA(p,1,q) process has a stationary infinite order moving average representation:

$$\begin{aligned} \Delta y_t &= c_0 e_t + c_1 e_{t-1} + \dots \\ \Delta y_t &= C(L) e_t \end{aligned}$$

This can be broken up as follows:

$$\Delta y_t = C(1) e_t + \Psi(L)(1 - L) e_t$$

where $\Psi(L) = \psi_0 + \psi_1 L + \dots$ is a polynomial with $\lim_{j \rightarrow \infty} \psi_j = 0$. We can identify the permanent and transitory components respectively as

$$\begin{aligned} \Delta P_t &= C(1) e_t \\ \Delta T_t &= \Psi(L)(1 - L) e_t \end{aligned}$$

Since $\Delta T_t = (1 - L)T_t$, then

$$T_t = \Psi(L)e_t$$

Thus, P_t is an I(1) process while T_t is an I(0) process. The two characteristics of this decomposition are:

1. The shocks into the permanent component $C(1)e_t$ are white noise.
2. The shocks into the permanent and transitory components are perfectly correlated through the common value e_t .

The implementation of the Beveridge-Nelson decomposition actually follows Newbold (1990) who shows that:

$$\bar{y}_t = y_t + \lim_{k \rightarrow \infty} \left[\sum_{j=1}^k \hat{w}_t(j) \right] = y_t + c_t$$

where \bar{y}_t is the permanent component of y_t and $\hat{w}_t = \Delta \hat{y}_t - \mu$, where μ is the mean of the ARMA (p,q) process of Δy_t and $\Delta \hat{y}_t$ are the forecasts from a fitted ARMA process on Δy_t . The term

$$\begin{aligned} c_t &= \lim_{k \rightarrow \infty} \left[\sum_{j=1}^k \hat{w}_t(j) \right] \\ &= \sum_{j=1}^q \hat{w}_t(j) + (1 - \phi_1 - \dots - \phi_p)^{-1} \sum_{j=1}^p \sum_{\tau=j}^p \phi_\tau \hat{w}_t(q - j + 1) \end{aligned}$$

where p and q are the orders in the fitted ARMA(p,q) model.

Consider a data set containing the time series y_t that we would like to decompose into permanent and transitory components. For concreteness, we will use the quarterly data on real gross domestic product (seasonally adjusted, 1 decimal place) available at FRED.³

```
filename gdp "..\Final\GDPC1.txt";
data GDP;
  infile gdp firstobs = 14;
  /* Read two variables: date and GDP */
  input date value;
  informat date yymmdd10.;
  format date yymmdd10.;
  lngdp = log(value);
  /* We will refer to Y instead of LNGDP for consistent notation */
  y=lngdp;
  /* Calculate the first difference of y */
  diffy = dif(y);
run;
```

³<http://research.stlouisfed.org/fred2/>. This data set will be used for all the decompositions.

Next an ARMA(p,q) model is fitted. It is assumed that we have examined the results of PROC ARIMA using the IDENTIFY statement to determine the order of p and q . The orders are usually determined after looking at autocorrelations, inverse autocorrelations and autocorrelation residual checks as well as possibly looking at various information criteria using the options SCAN, ESACF, MINIC on the IDENTIFY statement. For purposes of this decomposition, we will assume that we have determined the order of p and q to both be 2. Next we determine the number of k -step ahead forecasts to be computed as the maximum of p and q . The parameter estimates and forecasts will then be used to form the permanent and transitory components. SAS/IML is used to compute these components.

```

/* Store the order of p and q as macro variables
   since they will be used repeatedly */
%let p = 2;
%let q = 2;
/* Compute number of k-step ahead forecasts
   as the max of the p,q order */
%let nlead = %eval(%sysfunc(max(&q, &p)));
/* Store AR and MA parameter estimates into
   data set called PAR */
ods output ParameterEstimates = Par;
proc arima data=gdp;
identify var=diffy;
estimate p=&p q=&q;
forecast id = date lead = &nlead noprint out = fore;
run;
quit;

proc sql noprint;
/* Store mean of ARMA process and number of observations
   as macro variables to be used later */
   select estimate into :mu
   from Par
   where parameter = 'MU';

   select count(*) into :nobs
   from GDP;
/* Create a data set containing only AR parameters to
   be used later */
   create table ARPar as
   select *
   from Par
   where substr(parameter,1,2) = 'AR';
quit;

data fore;
   set fore;
   /* Compute the deviation of y from its mean*/
   /* This is w in the text */

```

```

    fore_mu = forecast - &mu;
run;

proc iml;
    /* Create a t matrix from the forecasts */
    /* t is 231 + number of forecasts x 4
       containing all the observations
       and the variables diffy, fore_mu, residual and date. */
    use fore var{diffy fore_mu residual date};
    read all into t;
    /* Create a y matrix from the original
       data set containing y */
    /* The vector y contains the series-y and
       is 231 x 1. */
    use gdp var{y};
    read all into y;
    if &p > 0 then do;
        /* If there are AR estimates then create a matrix AR
           and sum the polynomials. The sum is stored in ARSUM */
        use ARPar var{estimate};
        read all into AR;
        arsum = sum(AR);
    end;
    /* Create matrix x to store our results */
    /* 8 is the number of columns we eventually want.
       See the vector colnm containing the column headings */
    x = J(&nobs,8,0);
    do n = 1 to &nobs;
        if &q > 0 then do;
            /* This is the first summation term in Newbold's eqn (2) */
            do i = 1 to &q;
                x[n,1] = x[n,1] + t[n+&q,2];
            end;
        end;
        /* This is the second term containing the double
           summation in Newbold's eqn (2) */
        if &p > 0 then do;
            do j = 1 to &p;
                do i = j to &p;
                    x[n,2] = x[n,2] + ar[i]*t[n+&q-j+1,2];
                end;
            end;
        end;
        /* This is the second term in Newbold's eqn (2) divided
           by the sum of the AR polynomials, i.e. c(t) or the
           cyclical/transitory component */
        x[n,3] = x[n,1] + 1/(1-arsum)*x[n,2];
        /* The trend is the output minus the cycle */
        x[n,4] = y[n,1]-x[n,3];
        /* Original series: DIFFY */

```

```

x[n,5] = t[n,1];
/* Forecasts adjusted by mean from ARIMA */
x[n,6] = t[n,2];
/* Original series: Y */
x[n,7] = y[n,1];
/* Original series: DATE */
x[n,8] = t[n,4];
end;
res = x;
colnm = {"SUMM1" "SUMM2" "CYCLE" "TREND" "DIFFY" "FORE_MU" "Y" "DATE"};
/* Create a data set SIM2 that contains all the results from matrix x */
create sim2 from res[colname = colnm];
append from res;
quit;

data sim2;
set sim2;
/* Compute percent cycle as deviation from trend rescaled by 100 */
pct_cycle = cycle*100;
label pct_cycle = 'Deviation from Trend: Cyclical/Transitory Component';
run;

```

The trend component is plotted against the log value of the original data series and is shown in Figure 1. The transitory or cyclical component is shown in Figure 2.

1.2.2 Baxter-King filter

The Baxter-King (1995) filter is a band-pass filter that attempts to isolate cycles with period lengths of between 6 and 32 quarters which is the typical length of U.S. business cycles. It is a symmetric centered moving average filter where

$$\tilde{y}_t = \sum_{J=-K}^{j=K} a_j L^j y_t$$

where \tilde{y}_t is the filtered series from the original time series y_t , and a_j are the weights. The weights are obtained by solving the minimization problem

$$\min_{a_j} Q = \int_{-\pi}^{\pi} |\beta(\omega) - \alpha(\omega)|^2 d\omega \quad \text{s.t.} \quad \alpha(\omega) = 0$$

where $\beta(\omega)$ is the Fourier transform of an 'ideal' filter and $\alpha(\omega)$ is the Fourier transform of the approximate filter. The solution to this minimization results

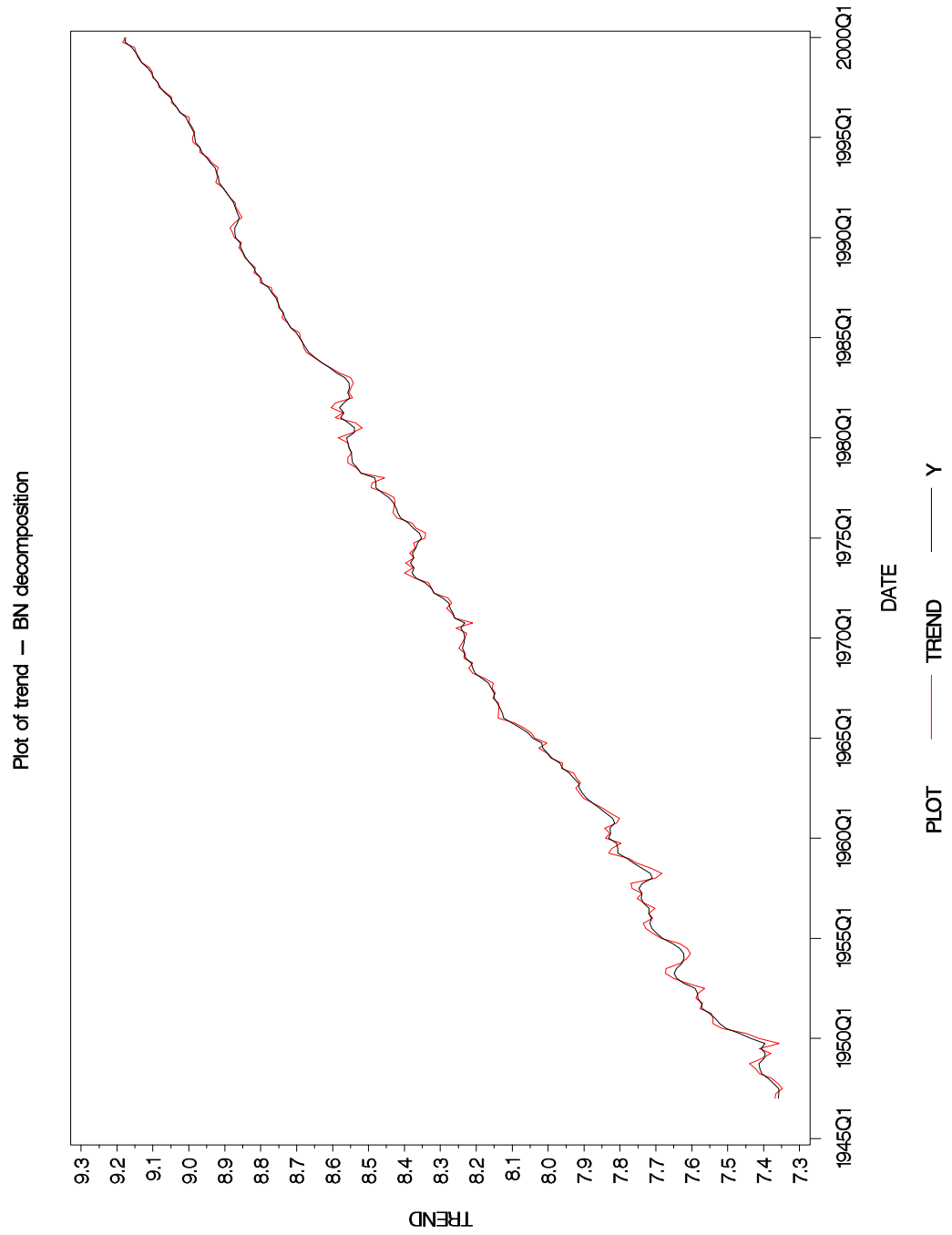


Figure 1: Beveridge-Nelson Decomposition: Trend component

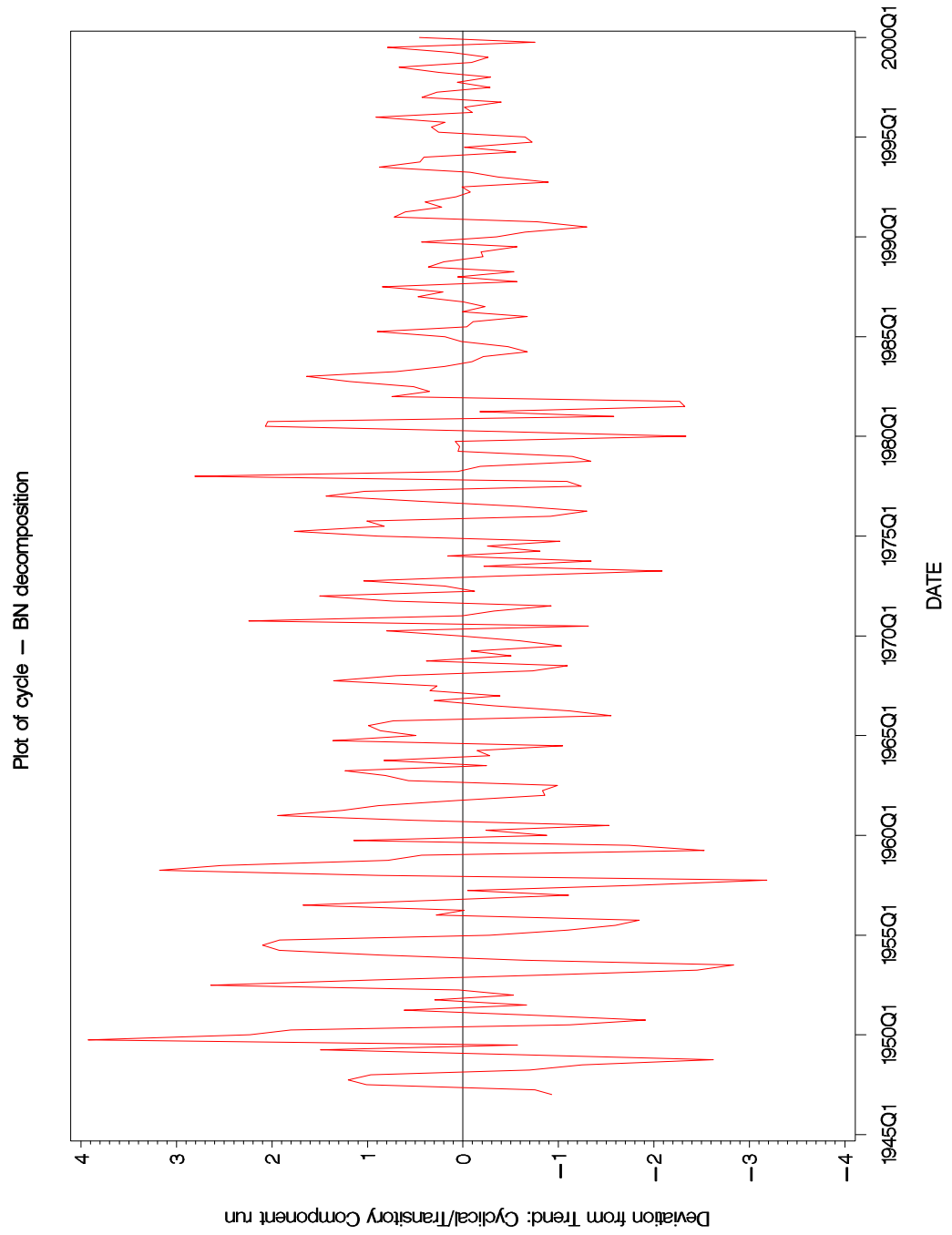


Figure 2: Beveridge-Nelson Decomposition: Transitory component

in:

$$a_j = b_j + \theta, \text{ where } j = 0, \pm 1, \dots, \pm K$$

$$b_j = \begin{cases} \frac{\omega_2 - \omega_1}{\pi} & \text{if } j = 0 \\ \frac{1}{\pi_j} (\sin \omega_2(j) - \sin \omega_1(j)) & \text{if } j = \pm 1, \pm 2, \dots \end{cases}$$

$$\theta = \frac{-\sum_{j=-K}^{j=K} b_j}{2K + 1}$$

Baxter and King propose setting $K = 12$, $\omega_1 = 2\pi\frac{1}{32}$, and $\omega_2 = 2\pi\frac{1}{6}$ or $2\pi\frac{1}{2}$ for quarterly data and $K = 3$, $\omega_1 = 2\pi\frac{1}{8}$, and $\omega_2 = \pi$. The filtered series is the cyclical component and the trend component is just the difference between the actual series and the cycle. SAS/IML is used to implement this filter.

```

/* omubar = omega2, omubar = omega1 in text */
/* Store as macro variables, values to be used */
%let up = 6; /* Denominator value for omega2 */
%let dn = 32; /* Denominator value for omega1 */
%let k = 20; /* Size of K for moving average sum */
%let phi = 0; /* See note below */
proc iml;
  pi = constant('pi');
  omubar = 2*pi/&up;
  omlbar = 2*pi/&dn;
  /* Create vector to compute b(j) */
  b = J(&k+1,1,.);
  do i = 0 to &k;
    if i = 0 then b[i+1,1] = (omubar - omlbar)/pi;
    else b[i+1,1] = (sin(i*omubar) - sin(i*omlbar))/(i*pi);
  end;
  /* Sum b where k > 0. This gives sum b from j=1 to K */
  sumb = sum(b[2:&k+1]);
  /* Now add case where k = 0 and multiply sumb by 2 since
  this is a 2-sided sum */
  thetatemp = b[1,1] + (2*sumb);
  /* Calculate theta */
  /* Phi=0 if dn<1000. If dn>1000 then assume low pass filter
  and set phi = 1*/
  /* See Baxter-King (1995) for details */
  theta = &phi - (thetatemp/(2*&k + 1));
  do i = 0 to &k;
    weight = b + theta;
  end;

  /* Create vector of centered moving average weights */
  adjvect = J(2*&k+1,1,0);
  do i = 0 to (nrow(adjvect)-1);
    if i <= round((2*&k)/2,1) then adjvect[i+1,1] = weight[&k+1-i];
    else adjvect[i+1,1] = weight[i-&k+1];
  end;

```

```

end;
/* Now filter the GDP series */
use GDP var{y date};
read all into y;
T = nrow(y);
/* FS is filtered series */
FS = J(T-2*&k,1,.);
do i = 0 to T-2*&k-1;
    *s = y[i+1:i+1+2*&k,1];
    FS[i+1,] = T(y[i+1:i+1+2*&k,1])*adjvect;
end;

fdate = y[&k+1:T-&k,2];
FS = fdate || FS;
colnm = {'date','FilteredSeries'};
create BKFiltered from FS[colname = colnm];
append from FS;
quit;

data BKFiltered;
merge BKFiltered GDP; by date;
cycle = FilteredSeries;
trend = y - cycle;
pct_cycle = cycle*100;
label pct_cycle = 'Deviation from Trend: Cyclical/Transitory Component';
run;

```

The trend component is plotted against the log value of the original data series and is shown in Figure 3. The transitory or cyclical component is shown in Figure 4.

1.2.3 Hodrick-Prescott filter

The Hodrick-Prescott (1997) filter is obtained by solving the minimization problem:

$$\min_{\{y_t\}_{t=1}^T} \sum_{t=1}^T (y_t - y_t^T)^2 + \lambda [(y_{t+1}^T - y_t^T) - (y_t^T - y_{t-1}^T)]^2$$

where λ is a penalty parameter. The filter is essentially finding a 'smooth' trend over the series. If $\lambda = 0$ then the filtered series is the original series, i.e. there is no smoothing. Hodrick and Prescott recommend using $\lambda = 1600$ for quarterly data. This filter is implemented as part of PROC EXPAND in SAS. For annual data, Ravn and Uhlig (2002) recommend using $\lambda = 6.25$ for annual data and $\lambda = 129600$ for monthly data.

```

proc expand data = GDP out=GDPPhf method = spline;
id date;
/* The output from this statement gives the trend/smooth series: yHPF=trend */
convert y = yHPF / transformout = (hp_t 1600);

```

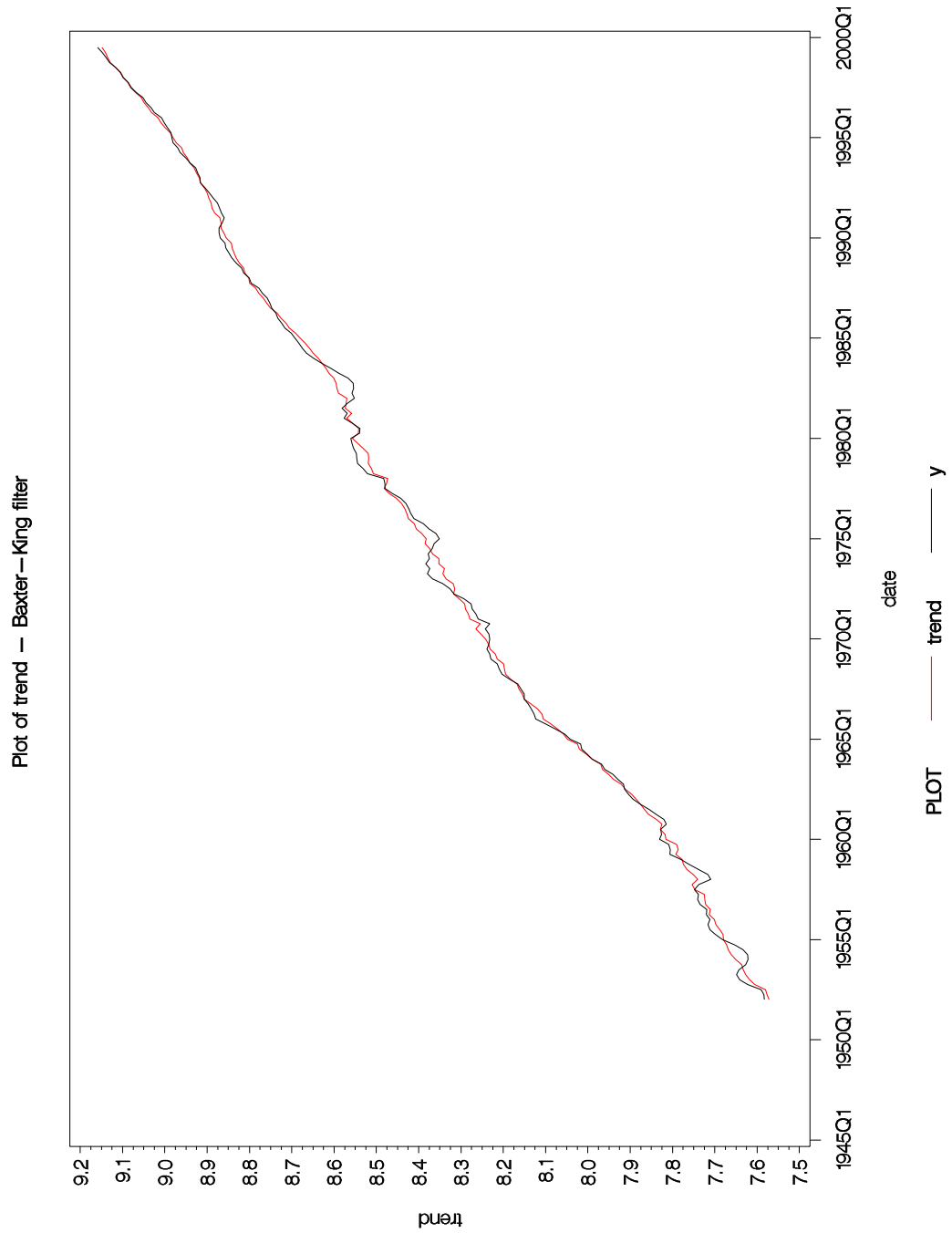


Figure 3: Baxter-King filter: Trend component

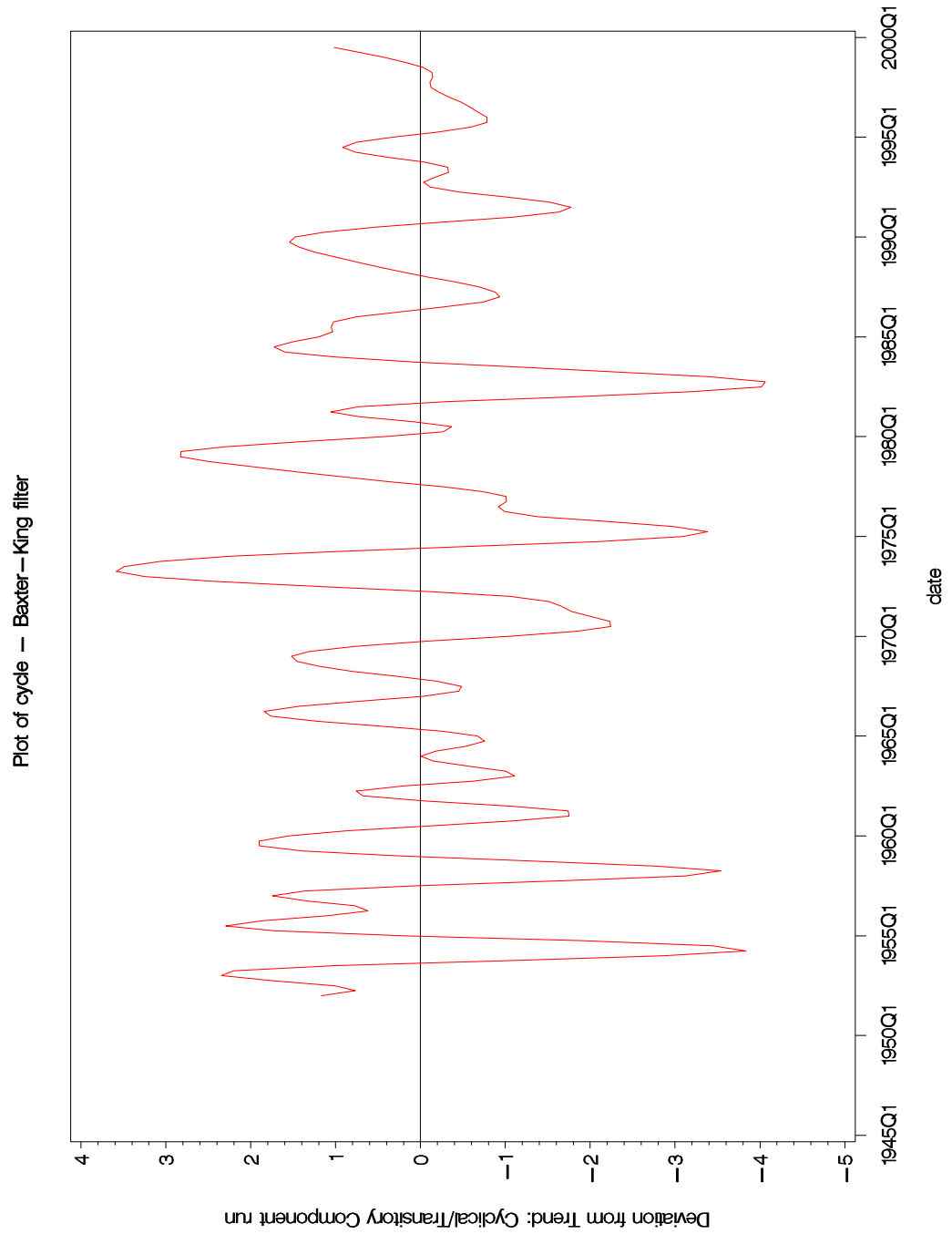


Figure 4: Baxter-King filter: Transitory component

```

/* The output from this statement gives the cyclical component: yHPC=cycle */
convert y = yHPC / transformout = (hp_c 1600);
run;

data GDPPhpf;
  set GDPPhpf;
  trend = yHPF;
  /* We can verify that the inverse is true i.e.
  compare hp_c 1600 (yHPC) with the computed cycle series. */
  cycle = y - trend;
  pct_cycle = cycle * 100;
  label pct_cycle = 'Deviation from Trend: Cyclical/Transitory Component';
run;

```

The trend component is plotted against the log value of the original data series and is shown in Figure 5. The transitory or cyclical component is shown in Figure 6.

1.2.4 Unobserved Components Model

The unobserved components model is of the following form:

$$\begin{aligned}
y_t &= \tau_t + c_t \\
t_t &= \tau_{t-1} + \mu + \eta_t, \quad \eta \sim \text{i.i.d } N(0, \sigma_\eta^2) \\
\phi_p(L)c_t &= \theta(L)\nu_t, \quad \nu \sim \text{i.i.d } N(0, \sigma_\nu^2), \\
&\text{Cov}(\eta_t, \nu_{t \pm k}) = \sigma_{\eta\nu} \text{ for } k = 0; 0 \text{ otherwise}
\end{aligned}$$

τ_t is the trend with a mean growth rate of μ and c_t is the cycle component and is stationary and ergodic. The additional condition that the cross correlations between the trend and cycle are zero,

$$\sigma_{\eta\nu} = 0$$

is commonly referred to as the UC0 model. SAS implements a version of the unobserved components model or what it calls the Basic Structural Model as:

$$\begin{aligned}
y_t &= \tau_t + c_t + \epsilon_t \\
t_t &= \tau_{t-1} + \mu_{t-1} + \eta_t, & \eta &\sim \text{i.i.d } N(0, \sigma_\eta^2) \\
\mu_t &= \mu_{t-1} + \xi_t, & \xi &\sim \text{i.i.d } N(0, \sigma_\xi^2)
\end{aligned}$$

The assumption that $\sigma_\xi^2 = 0$ means that the trend will be constant and is simply the UC0 model presented. SAS models the cycle as a deterministic cycle without an error component. This just assumes that the error term in the cyclical component will be captured by ϵ_t . The unobserved components model is implemented as a procedure called PROC UCM. The keywords `irregular` estimates the term ϵ_t , `slope` and `trend` captures the trend of the model with

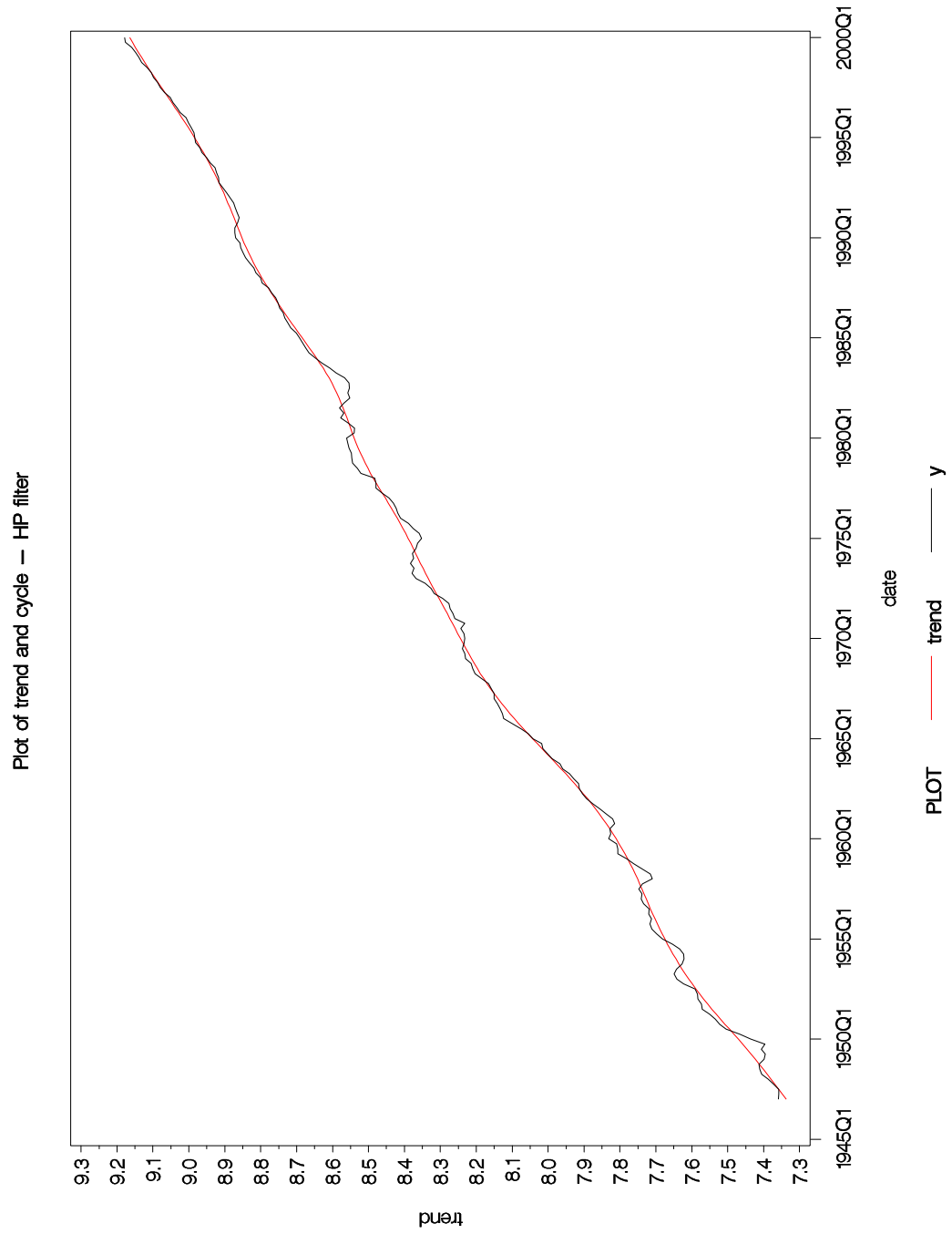


Figure 5: HP filter: Trend component

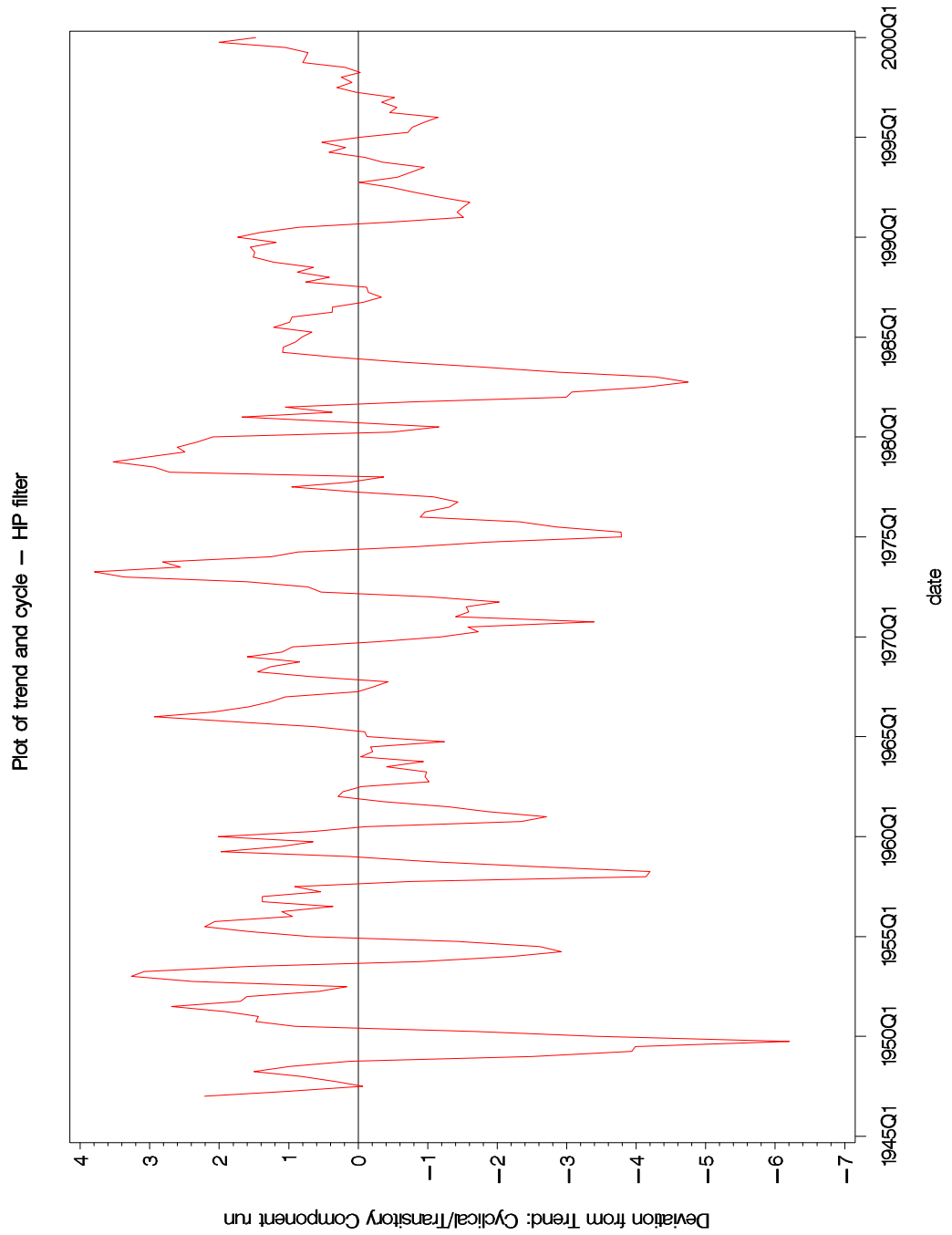


Figure 6: HP filter: Transitory component

a growth rate, and `cycle` estimates the cyclical component.⁴ PROC UCM outputs both smoothed (prefixed by S_) and filtered series (prefixed by F_). We use the smoothed series for the decomposition.

```
proc ucm data = GDP;
id date interval=quarterly;
model y;
/* Level and slope captures a trend with a growth rate */
level;
/* Noest variance = 0 -- do not estimate the variance,
   i.e. no error in slope */
slope noest variance = 0;
cycle;
/* Irregular captures the error term of the BSM */
irregular;
estimate;
forecast lead = 0 outfor = ucmfor;
run;

data ucmfor2;
  set ucmfor;

  /* S_LEVEL, S_SLOPE and S_CYCLE are smoothed series from
     the estimation process. By default, we use these series.
     The unsmoothed series are prefixed with F_ */
  strend = s_level + s_slope;
  pct_cycle = s_cycle * 100;
  label pct_cycle = 'Deviation from Trend: Cyclical/Transitory Component'
        strend = 'Trend';
run;
```

The trend component is plotted against the log value of the original data series and is shown in Figure 7. The transitory or cyclical component is shown in Figure 8.

1.3 Applications

1.3.1 Generating Stylized Facts

The cyclical components from the detrended series are used to generate statistics on the various time series of interest. These statistics consist of a volatility measure of the cyclical component (usually the standard deviation) as well as autocorrelations at various leads and lags. The reference time series is usually

⁴Unfortunately, PROC UCM does not allow a more flexible parameterization of the covariance structure between the error components such as that investigated by Morley, et. al. (2003). Morley et. al. finds that the restriction that the zero covariance between the error terms in the trend and cycle components is rejected by the data.

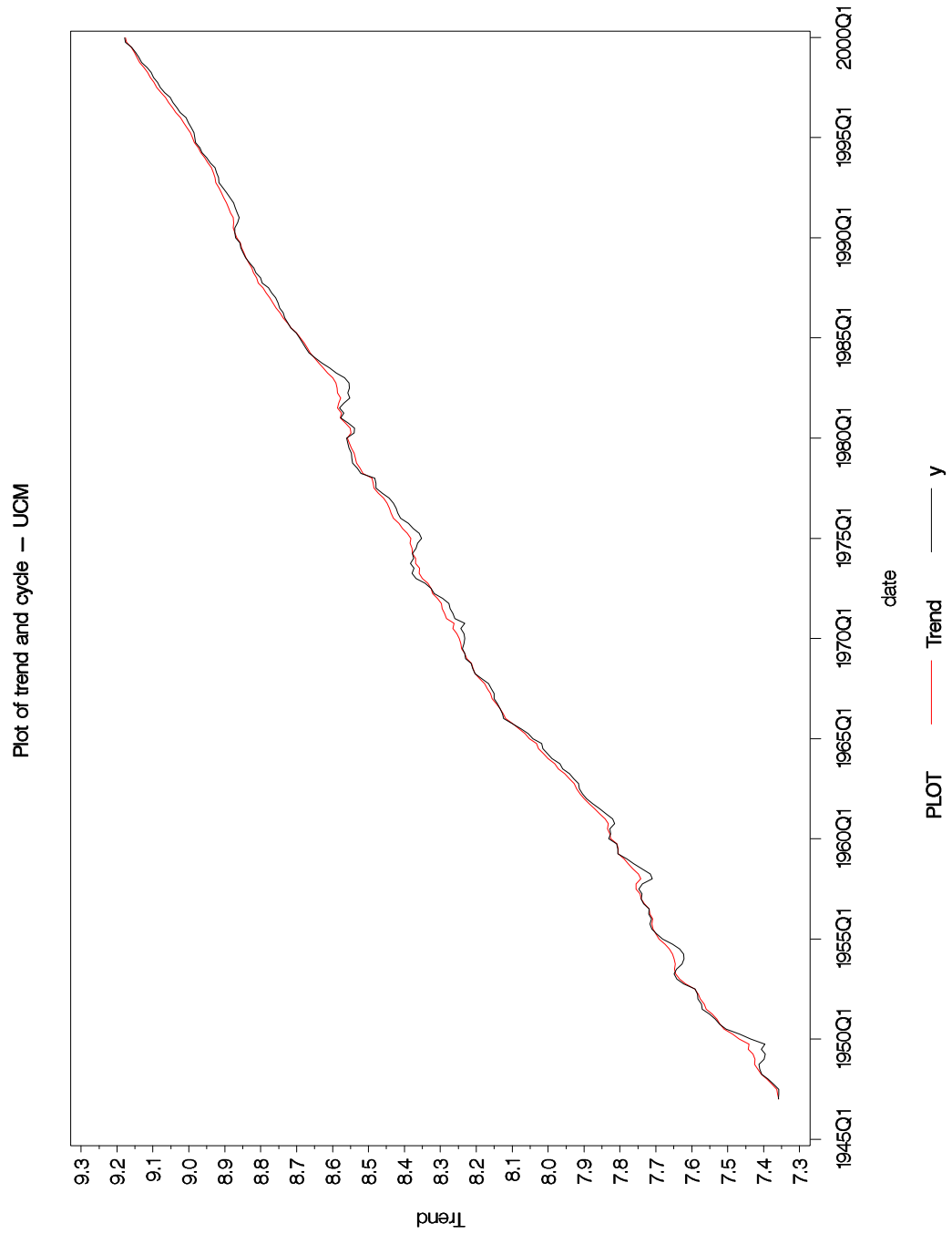


Figure 7: Unobserved Components Model: Trend component

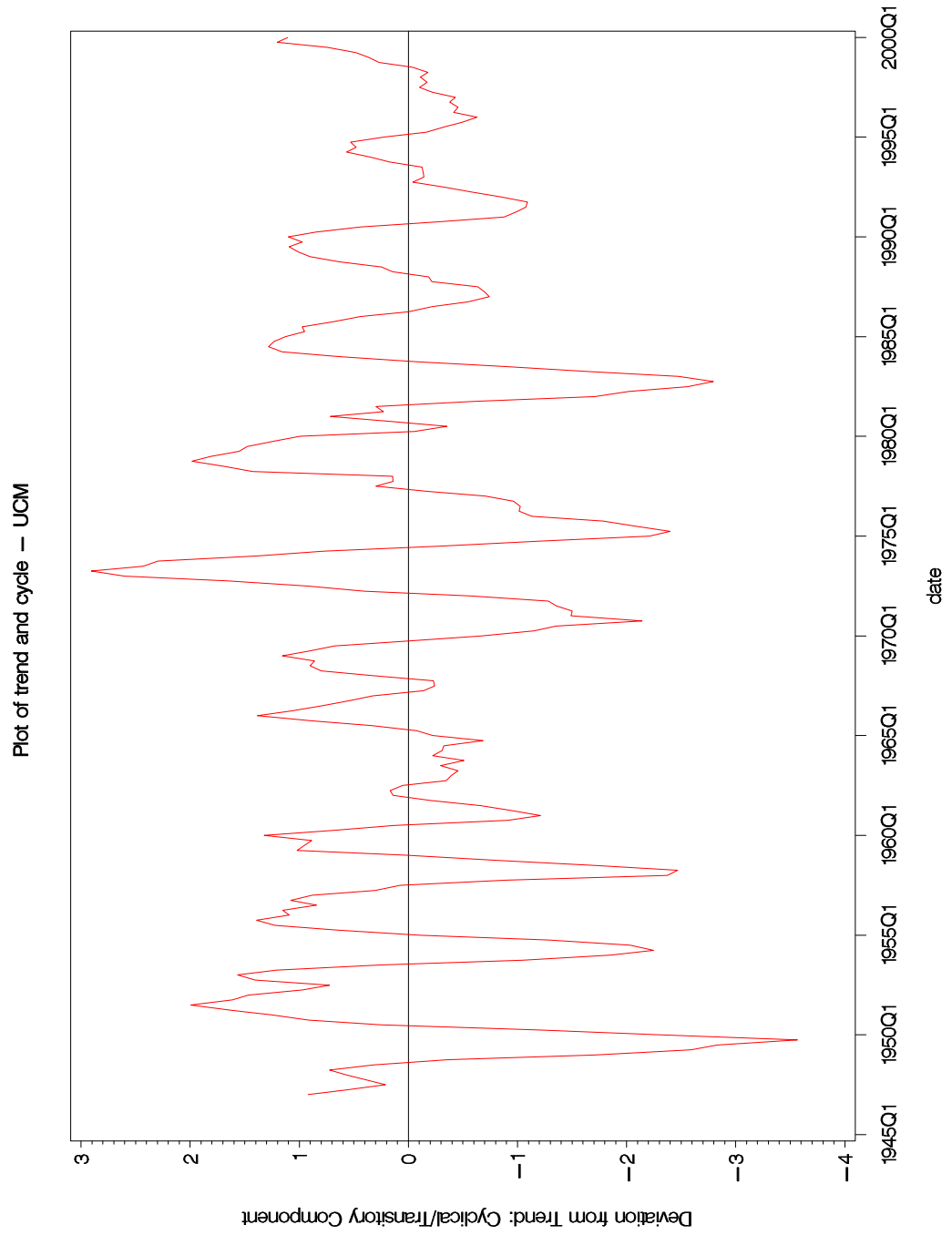


Figure 8: Unobserved Components Model: Transitory component

a measure of overall aggregate economic activity such as GNP (although some have used GDP or indices of manufacturing activity). The autocorrelation of the cyclical component of GNP at different leads and lags are computed and then the cyclical components of other time series are cross correlated with the reference time series at the different leads and lags. The number of leads and lags varies from 5 to 8. The statistics are then presented in a tabular format. Numerical models of business cycles then attempt to replicate these statistics.

The following code demonstrates how SAS can be used to generate the stylized facts. The HP filter is used as an example. The time series are log values of GNP, private investment (fixed investment and nonresidential), government consumption and investment, and components of personal consumption (overall, durable and services). The data is available from FRED or the BEA. The main procedure used is PROC ARIMA with the CROSSCORR option. We are not interested in anything else that is a result of the procedure except for the cross correlations. Unfortunately, the variables in the output data set with the cross correlations are unlabeled. Most of the code is used to label the variables and reshape the data for presentation.

```

libname here '.';
/* HP filter the log values of the time series */
/* Save the cyclical components suffixed with C
   to a data set called HPF */
proc expand data = here.dataCrossCorr out=hpf method = spline;
id date;
convert lngnp = lngnpC / transformout = (hp_c 1600);
convert lnfpic = lnfpicC / transformout = (hp_c 1600);
convert lngce = lngceC / transformout = (hp_c 1600);
convert lnpecC = lnpecC / transformout = (hp_c 1600);
convert lnpcsvC = lnpcsvC / transformout = (hp_c 1600);
convert lnpcndgC = lnpcndgC / transformout = (hp_c 1600);
convert lnpnfiC = lnpnfiC / transformout = (hp_c 1600);
run;

/* Output the values of cross correlations to
   a SAS data set called C1 */
ods output CorrGraph = c1;
proc arima data = hpf;
/* Compute correlations at lead and lag of 5 */
identify var = lngnpC
         crosscorr = (lnfpicC lngceC lnpecC lnpcsvC lnpcndgC lnpnfiC) nlag=5;
run;
quit;

/* Number of lead and lag */
%let nlag = 5;
/* Number of series in cross corr statement in PROC ARIMA */

```

```

%let ncrosscorr = 6;
/* List of variables for cross correlations */
%let crosscorrlist = lnfpic lngceC lnpcecC lnpcesvC lnpcondgC lnpnfiC;
/* This macro is used to label the variables since the output
   from PROC ARIMA does not label the cross correlations */
%macro nameit;
    %do i = 1 %to &ncrosscorr; /* Number of series in Crosscorr */
        %let cvar = %scan (&crosscorrlist, &i);
        begin = (&i-1)*(2*(&nlag + 1)) + &nlag + 1 - &i+1;
        end = &i*(2*(&nlag + 1)) + &nlag + 1 - &i;
        if begin < _N_ <= end then do;
            variable = %upcase("&cvar");
            varnum = &i;
        end;
    %end;
%mend;

data c2;
    set c1 end=eof;
    length w with variable $8;
    if _N_ <= (&nlag + 1) then do;
        /* Assume that the first &nlag + 1 rows are is the reference series */
        /* This is the variable specified in the VAR of the IDENTIFY
           statement in PROC ARIMA */
        variable = 'LNGNP';
        varNum = 0;
    end;
    else do ;
        /* Now name the rest of the series for all leads and lags */
        %nameit;
    end;
    /* Create variable names and labels to identify leads and lags:
       x_t__ are lags and x_t_ are leads */
    if lag < 0 then do;
        w = 'x_t_' || compress(put(abs(lag),2.));
        with = 'x(t' || compress(put(lag,2.)) || ')';
    end;
    else if lag = 0 then do;
        w = 'x_t';
        with = 'x(t)';
    end;
    else if lag > 0 then do;
        w = 'x_t_' || compress(put(abs(lag),2.));
        with = 'x(t+' || compress(put(lag,2.)) || ')';
    end;
run;

```

```

proc sort data = c2; by descending varnum; run;

/* Reshape the data using the variable name and labels for reporting */
proc transpose data = c2 out = c2t let; by variable varnum notsorted;
id w;
idlabel with;
var correlation;
run;

/* ARIMA only computes autocorrelation of leads for reference series */
/* Since the autocorrelations are symmetric, fill in the lags with leads
at the corresponding lag value */
data c3(drop = i);
set c2t(drop = _NAME_);

array f(*) x_t_1 - x_t_&nlag;
array e(*) x_t_1 - x_t_&nlag;

if varnum = 0 then do;
do i = 1 to dim (f);
e(i) = f(i);
end;
end;
proc sort; by varnum variable;
run;

/* Compute the volatility of the series */
proc means data = hpf mean std;
var lngnpC &crosscorrlist;
output out = vol std = ;
run;

data volatility(keep = varnum vol);
set vol(drop = _type_ _freq_);

array v(*) lngnpC &crosscorrlist;
/* Rescale by 100 */
do varnum = 0 to (dim(v)-1);
vol = v(varnum + 1)*100;
output;
end;
run;

/* This macro is used to label the leads and lags
for the tabular report */

```

```

%macro labelit;
    %do i=1 %to &nlag;
        x_t_&i = "X(&i)"
        x_t_--&i = "X(-&i)"
    %end;
%mend labelit;

proc format;
    value $vn
        LNGNP      = 'Gross National Product'
        LNFPIC     = 'Private Fixed Investment'
        LNGCEC     = 'Government Consumption Expenditures & Gross Investment'
        LNPCECC    = 'Personal Consumption Expenditures'
        LNPCE SVC  = 'Personal Consumption Expenditures: Services'
        LNPCNDGC   = 'Personal Consumption Expenditures: Nondurable Goods'
        LNPNFIC    = 'Private Nonresidential Fixed Investment'
    ;
run;

/* Merge the data with the volatility and the cross correlations
   for tabular report */
data report;
    merge volatility c3; by varnum;
    varname = put(variable, $vn.);
    label x_t = "X(0)"
    %labelit
    vol = "Volatility"
    ;
run;

ods tagsets.tablesonlylatex file="crossCorr1.tex" (notop nobot);
/* Produce the table */
proc print data = report noobs label split='*';
var varname vol x;
format vol x: 8.2;
label varname = 'Variable/Series';
run;
ods tagsets.tablesonlylatex close;

```

The results are presented in Table 1.

Variable/Series	Volatility	X(-5)	X(-4)	X(-3)	X(-2)	X(-1)	X(0)	X(1)	X(2)	X(3)	X(4)	X(5)
Gross National Product	1.70	-0.10	0.08	0.32	0.60	0.84	1.00	0.84	0.60	0.32	0.08	-0.10
Private Fixed Investment	5.10	-0.19	-0.03	0.20	0.44	0.67	0.80	0.74	0.58	0.36	0.17	0.01
Government Consumption Expenditures & Gross Investment	3.62	0.34	0.39	0.38	0.34	0.26	0.17	0.04	-0.05	-0.10	-0.11	-0.09
Personal Consumption Expenditures	1.27	-0.25	-0.12	0.06	0.32	0.56	0.75	0.75	0.64	0.47	0.30	0.12
Personal Consumption Expenditures: Services	0.71	-0.03	0.10	0.26	0.43	0.59	0.70	0.69	0.57	0.40	0.20	0.02
Personal Consumption Expenditures: Nondurable Goods	1.11	-0.10	0.05	0.22	0.43	0.62	0.72	0.68	0.55	0.37	0.20	0.03
Private Nonresidential Fixed Investment	4.96	0.11	0.33	0.55	0.71	0.81	0.77	0.55	0.29	0.04	-0.13	-0.23

Table 1: Application: Generating Stylized Facts

The results presented are comparable to those obtained by Kydland and Prescott's (1990) Table 2.

1.3.2 The effects of Taylor rules under different detrending assumptions

One of the uses of the cyclical component is its interpretation as a measure of output gap, i.e. how far output is from its trend. This component is used sometimes featured in monetary policy, that is what should the interest rate be based on the current or future cycle in GDP? One of the more well known monetary policy rules is the Taylor rule. The following discussion is taken from McCallum (2000).

$$R_t = \bar{r} + \Delta p_t^a + 0.5(\Delta p_t^a - \pi^*) + 0.5\tilde{y}_t$$

R_t is the short term nominal interest rate to be set by the policy maker, \bar{r} is the long-run average real interest rate, Δp_t^a is the average recent inflation rate or a forecasted inflation rate for the current period, π^* is the targeted inflation rate, and \tilde{y}_t is the output gap. The following assumptions are made: $\pi^* = \bar{r} = 2$ and $\Delta p_t^a = 0.25(\Delta p_{t-1} + \Delta p_{t-2} + \Delta p_{t-3} + \Delta p_{t-4})$. This variable needs to be rescaled to 1000 and since the policy maker does not know current prices, $\Delta p_{t-1} = p_{t-1} - p_{t-2}$ and $\Delta p_{t-2} = p_{t-2} - p_{t-3}$, and so forth.

The code is very straightforward. We follow McCallum's use of data from FRED. The output series is filtered using three different filters: the Baxter-King filter, HP filter and UCM. The cyclical components from each filtered series is merged onto the data set that contains the federal funds rate and the interest rate implied by the Taylor rule under each of the filters is then plotted. The results are shown in Figure 9. There does not appear to be large differences in the interest rate rules due to different detrending assumptions.

```

/* real GDP series */
filename gdp "..\Final\GDPC1.txt";
data rGDP;
    infile gdp firstobs = 14;
    input date rgdp;
    informat date yymmdd10.;
    format date yymmdd10.;
    lngdp = log(rgdp);
    y=lngdp;
    label y = 'Real GDP (chain-linked)';
run;
/* Nominal GDP series */
filename ngdp "..\Final\GDP.txt";
data nGDP;
    infile ngdp firstobs = 14;
    input date ngdp;
    informat date yymmdd10.;

```

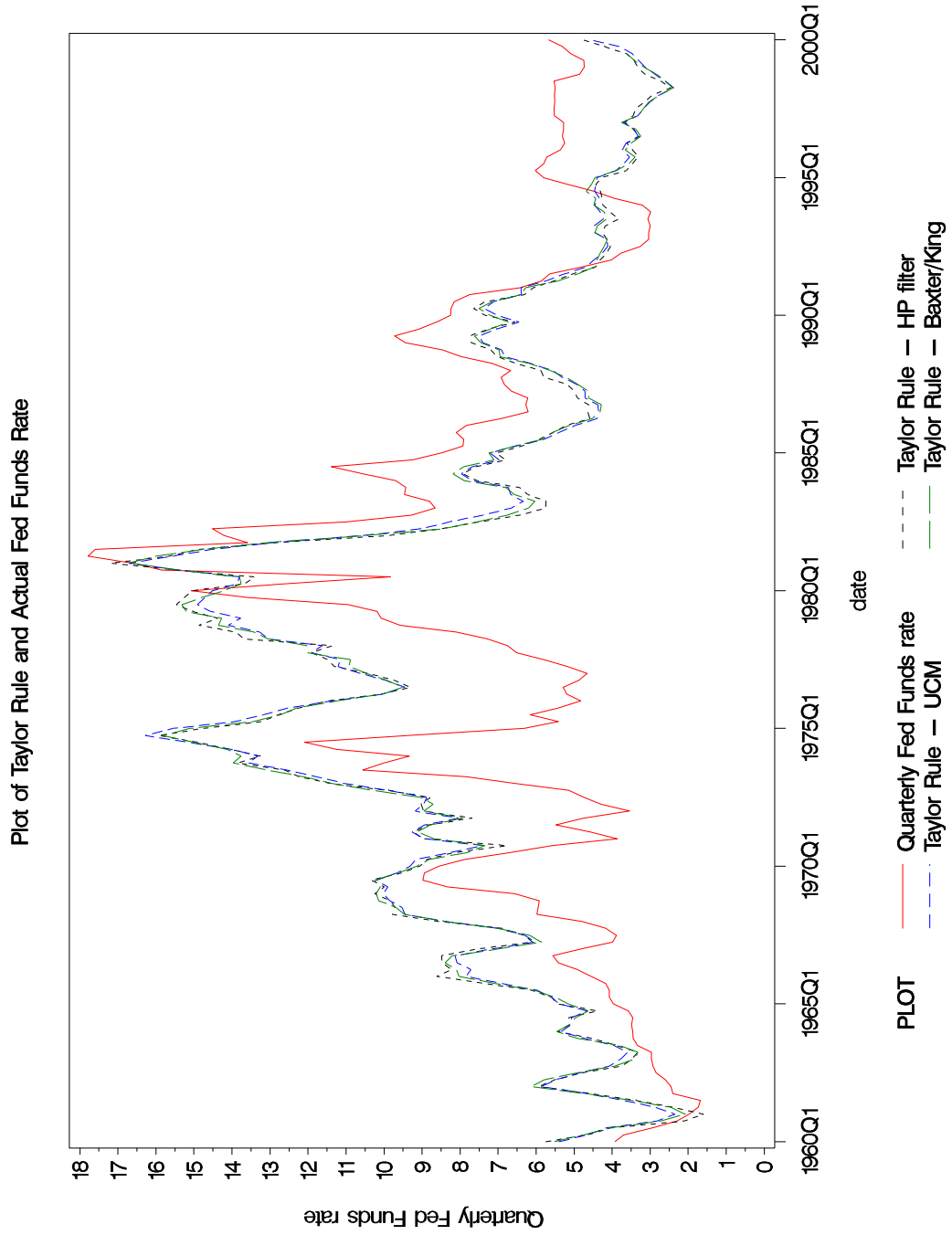


Figure 9: Effects of Detrending on Taylor Rule

```

        format date yymmdd10.;
        lnngdp = log(ngdp);
        x=lnngdp;
        label x = 'Nominal GDP';
run;
/* Federal Funds Rate series */
filename ff "..\Final\FEDFUNDS.txt";
data mFF;
    infile ff firstobs = 13;
    input date ffmonthly;
    informat date yymmdd10.;
    format date yymmdd10.;
    label ffmonthly = 'Fed Funds Rate (Monthly)';
run;

/* Reduce monthly to quarterly using PROC EXPAND */
proc expand data = mFF out = qff from=month to=quarter;
/* Assume that monthly federal funds rate is the average for the month */
convert ffmonthly = ffquarterly/method = aggregate observed=average;
id date;
run;

/* Using HP filter */
proc expand data = rGDP out=rGDPHPf method = spline;
id date;
convert y = yHPC / transformout = (hp_c 1600);
run;

/* Using UCM */
proc ucm data = rGDP;
id date interval=quarterly;
model y;
level;
slope noest variance = 0;
cycle;
irregular;
estimate;
forecast lead = 0 outfor = ucmfor;
run;

data ucmfor2;
    set ucmfor;

    yUCMC = s_cycle;
    label yUCMC = 'UCM Cyclical Component'
        ;

```

```

run;

/* Using Baxter-King filter */
/* omubar = omega2, omlbar = omega1 in text */
/* Store as macro variables, values to be used */
%let up = 6; /* Denominator value for omega2 */
%let dn = 32; /* Denominator value for omega1 */
%let k = 20; /* Size of K for moving average sum */
%let phi = 0; /* See note below */
proc iml;
    pi = constant('pi');
    omubar = 2*pi/&up;
    omlbar = 2*pi/&dn;
    /* Create vector to compute b(j) */
    b = J(&k+1,1,.);
    do i = 0 to &k;
        if i = 0 then b[i+1,1] = (omubar - omlbar)/pi;
        else b[i+1,1] = (sin(i*omubar) - sin(i*omlbar))/(i*pi);
    end;
    /* Sum b where k > 0. This gives sum b from j=1 to K */
    sumb = sum(b[2:&k+1]);
    /* Now add case where k = 0 and multiply sumb by 2 since
    this is a 2-sided sum */
    thetatemp = b[1,1] + (2*sumb);
    /* Calculate theta */
    /* Phi=0 if dn<1000. If dn>1000 then assume low pass filter
    and set phi = 1*/
    /* See Baxter-King (1995) for details */
    theta = &phi - (thetatemp/(2*&k + 1));
    do i = 0 to &k;
        weight = b + theta;
    end;

    *print adjweight;
    /* Create vector of centered moving average weights */
    adjvect = J(2*&k+1,1,0);
    do i = 0 to (nrow(adjvect)-1);
        if i <= round((2*&k)/2,1) then adjvect[i+1,1] = weight[&k+1-i];
        else adjvect[i+1,1] = weight[i-&k+1];
    end;
    *print adjvect;
    /* Now filter the GDP series */
    use rGDP var{y date};
    read all into y;
    T = nrow(y);
    /* FS is filtered series */

```

```

FS = J(T-2*&k,1,.);
do i = 0 to T-2*&k-1;
    *s = y[i+1:i+1+2*&k,1];
    FS[i+1,] = T(y[i+1:i+1+2*&k,1])*adjvect;
end;

fdate = y[&k+1:T-&k,2];
FS = fdate || FS;
colnm = {'date','FilteredSeries'};
create BKFiltered from FS[colname = colnm];
append from FS;
quit;

data BKFiltered;
merge BKFiltered rGDP; by date;
yBKC = FilteredSeries;
BKtrend = y - yBKC;
label yBKC = 'Baxter-King Cyclical Component';
run;

/* Merge all data and filtered cyclical components */
data taylor;
merge rgdp(in=r) ngdp(in=n) qff(in=f)
    rGDPPhf(in=h keep=date yHPC) ucmfor2(in=u keep = date yUCMC)
    BKFiltered(in=bk keep = date yBKC);
by date;

label ffquarterly = 'Quarterly Fed Funds rate'; /* using PROC EXPAND */
if r and n and f and h and u and bk then output;
run;

data taylorSim;
set taylor;
rbar = 2;
pi_star = 2;
p = x/y;
/* delta_p_a = 0.25*(dif(p) + dif2(p) + dif3(p) + dif4(p))*1000; */
delta_p_a = 0.25*(dif2(p) + dif3(p) + dif4(p) + dif5(p))*1000;
Taylor_HP = rbar + delta_p_a + (0.5*(delta_p_a - pi_star)) + (0.5*yHPC*100);
Taylor_UCM = rbar + delta_p_a + (0.5*(delta_p_a - pi_star)) + (0.5*yUCMC*100);
Taylor_BKC = rbar + delta_p_a + (0.5*(delta_p_a - pi_star)) + (0.5*yBKC*100);
label Taylor_HP = 'Taylor Rule - HP filter'
    Taylor_UCM = 'Taylor Rule - UCM'
    Taylor_BKC = 'Taylor Rule - Baxter/King';
run;

```

```

/* Plot filtered components */
options orientation = landscape;
goptions device = pdfc ftext = swiss fontres = presentation htext = 1.0;
axis1 label = (angle=90);
axis2 order = ('1Jan1960'd to '01Jul2004'd by year5) minor = none;
symbol1 color = red interpol = j line = 1;
symbol2 color = black interpol = j line = 2;
symbol3 color = blue interpol = j line = 3;
symbol4 color = green interpol = j line = 4;

ods pdf file = 'taylorRules.pdf';
title1 "Plot of Taylor Rule and Actual Fed Funds Rate";
proc gplot data = TaylorSim;
plot (ffquarterly Taylor_HP Taylor_UCM Taylor_BKC) * date /overlay legend
    vaxis = axis1 haxis = axis2;
where date >='01Jan1960'd;
format date yyq6.;
run;
quit;
ods pdf close;

```

1.4 Bibliography

1. Baxter, Marianne and Robert G. King (1995). *Measuring Business Cycles: Approximate Band-Pass Filters for Economic Time Series*, NBER Working Paper 5022.
2. Beveridge, Stephen and Charles R. Nelson (1981). *A new approach to the decomposition of economic time series into permanent and transitory components with particular attention to measurement of the 'business cycle'*, Journal of Monetary Economics, Vol. 19, pp. 125-127.
3. Burns, Arthur F., and Wesley C. Mitchell (1946). *Measuring Business Cycles*, National Bureau of Economic Research.
4. Hodrick, Robert K. and Edward C. Prescott (1997). *Postwar U.S. Business Cycles: An Empirical Investigation*, Journal of Money, Credit and Banking. vol. 29(1), pp. 1-16.
5. Kydland, Finn E. and Edward G. Prescott (1990). *Business Cycles: Real Facts and a Monetary Myth*, Federal Reserve Bank of Minneapolis Quarterly Review, Vol. 14 No. 2, pp. 3-18.
6. Marcet, Albert and Morten O. Ravn (2003). *The HP-Filter in Cross-Country Comparisons*, working paper.
7. McCallum, Bennett T. (2000). *Alternative Monetary Policy Rules: A Comparison with Historical Settings for the United States, the United*

- Kingdom, and Japan*, Federal Reserve Bank of Richmond Quarterly Review, Volume 86(1), Winter, pp. 49-79.
8. Morley, James, Charles R. Nelson and Eric Zivot (2003). *Why Are Unobserved Component and Beveridge-Nelson Trend-Cycle Decompositions of GDP So Different?*, Review of Economics and Statistics, vol. 85, no. 2 (May), pp. 235-243.
 9. Newbold, Paul (1990). *Precise and efficient computation of the Beveridge-Nelson decomposition of economic time series*, Journal of Monetary Economics, Vol. 26, pp. 453-457.
 10. Ravn, Morten O. and Harald F. Uhlig (2002). *On Adjusting the HP-filter for the Frequency of Observations*, Review of Economics and Statistics. vol. 84(1), 371-80.
 11. Serletis, Apostolos and David Krause (1996). *Nominal stylized facts of U.S. business cycles*, Review, Federal Reserve Bank of St. Louis, issue Jul, pages 49-54.
 12. Zarnowitz, Victor (1993). *What is a business cycle?*, NBER Working Paper 3863.