NYU Silver
Silver School of Social Work

An Introduction to State Space Models

Robert A. Yaffee, Ph.D.
Silver School of Social Work
New York University

Short course Day 2
Norwegian University of Science and Technology
Trondheim, Norway
November 26, 2009

Outline

- Day 2: Early AM
  - Motivation
  - Definition of the classical state space form
  - Brief History
  - The Kalman Filter and how it works
    - Initial values
    - Prediction
    - Updating correction
    - Reiteration
    - Optimization: ML, QML, GLS
    - Smoothing and signal extraction
    - Forecasting
  - Classical assumptions
  - Local level model
  - Local linear trend model

Outline

- Use of dfactor in Stata
- Sargent and Sims, Geweke's dissertation
  - Andrew Harvey 1989 envisions this as part of State space models
  - Forni, Lippi, Hallin, and Richlin
  - Stock and Watson develop a coincident indicator
  - Kim and Nelson, 1999
  - Ben Bernanke (2003) looks for the driving forces of the economy from output of a structural VAR with factor augmented VAR.

Outline

- Day 2: Late PM
  - Seasonality and the Basic structural model
  - Cyclicity
  - Interventions
  - Exogenous series
  - autocorrelation
  - The general state space model

Outline

- Day 2: Early PM
  - SsfPack system file generation
  - The Kalman filter
    - Missing observations
  - The augmented Kalman filter
    - Nonstationary processes
  - The extended Kalman filter
    - Nonlinear processes

- Day 2: Late PM
  - Filtering and Forecasting
  - Smoothing
  - Diagnostics
  - Introduction to multivariate models
  - Common features
  - Cointegration
  - Adjusting the variance matrix structure

Acknowledgment

• I am very grateful to Andrew C. Harvey, Siem Jan Koopman, Neil Shephard, along with Eric Zivot, Jiahui Wang, and Ruey Tsay for their written contributions to this field. Ralph Snyder from Monash University was very helpful as well. This presentation follows their writings although I do not go into all of the detail they do owing to time and space constraints. Nevertheless, they have made great contributions to time series analysis for which many of us remain grateful. Not to be forgotten in this area are the works of Jim West and Jeff Harrison, along with Giovanni Petris.

SsfAbout()

SsfPack Extended version 3.00 (September 2008)(c) 1997-2008 Siem Jan Koopman --- www.ssfpack.com   Please quote: Koopman, S.J., N. Shephard and J.A. Doornik (1999)   Statistical algorithms for models in state space using SsfPack 2.2   Econometrics Journal, 1999, Volume 2, p.113-166.   Further details: Koopman, S.J., N. Shephard and J.A. Doornik (2008)   SsfPack 3.0: Statistical algorithms for models in state space   London: Timberlake Consultants Ltd, 2008.

Why are state space models so important?

- State space models comprise a new paradigm in time series analysis and control.
- They can be used to any type of ARIMA analysis.
- ARIMA analysis is a subset of the state space paradigm.
- State space models can model nonstationary series, which ARIMA models cannot.
- State space models can handle missing values, which ARIMA models cannot.
- State space models with proper feedback systems can be self-correcting.
- Advanced state space models can handle nonlinear systems, which ARIMA models cannot.
- Advanced state space models can accommodate nonGaussian processes, which ARIMA models cannot.
- In short, state space models comprise a new paradigm in time series analysis and control.

Why is the Kalman Filter so Important?

- The Kalman filter is one of the major contributions to modern operations research.
- It is a crucial supplement to modern econometric methods.
- The Kalman filter is a vector system of difference equations explaining state dynamics (Athens, M. 1974, 2).

What is the state space model based on the Kalman filter?

- This system is comprised of **two basic equations**: the measurement equation and the state (transition) equation.
- The starting values for the system are important.
- Filtering entails the use of **updating equations** as well. These equations sequentially update the mean and variance by a weighted average, corrected by a factor analysis. As it updates the mean and the variance, the Kalman filter proceeds according to a Markov evolutionary process (a first-order autoregressive process) plus a regression on the innovation. This Kalman filter is the predictive basis of the forecasts generated by the system. In this step, the objective is to estimate the moments of the predictive step (Hyndman et al.,2008, 197). This is the predictive step.

What is a state space model based on the Kalman filter?

- The predictive step is followed by a corrective measurement step. A factor analysis of the unobserved components corrects for inaccurate measurement error conjoined with the transition error. The objective of this step is to find the moments of the response variable.
- This completes the cycle and reiteration takes place until all of the data are filtered.
- When all of the data are filtered, ***the Kalman smoother*** can be used for signal extraction to smooth and extract the signal as well. Once the model has been specified, fit, and optimized, and diagnosed as well behaved, we can then proceed to forecast with it and evaluate those forecasts.
- **Estimation** can be accomplished by maximum likelihood (BFGS), the EM algorithm, or MCMC . We shall explain all of these things in more satisfying detail soon.

Local level model

provides an introductory example

- In other words,

- Initial values of the mean and variance of the state vector are found.
- The model filter provides for an AR(1) evolution of a random walk plus noise.   Filtering means recovering the state variable from the noise, given the previous information. It does this by an efficient one-step ahead forecast plus a regression on the error.
- The measurement model provides for the correction after the prediction step.
- Maximum likelihood estimation provides the means of minimizing the predictive error variance during the estimation of the parameters.
- The process reiterates until a steady-state solution is attained.

The Measurement (Observation) Equation

$$y_t = Z_t \alpha_t + \varepsilon_t \qquad \varepsilon_t \sim NID(0, \sigma_\varepsilon^2)$$

where

Z_t = *selection matrix of factor loadings*

α_t = *state vector (contains all elements needed to describe the current and past dynamic nature of itself).*

ε_t = *observation error (irregular component) matrix*

$\varepsilon_t \sim NID(0, H_t)$

H_t = *observation error variance matrix*

Aspects of the measurement model

- Z_t can be construed as a matrix of factor loadings from observed variables on an underlying factor.
- α_t can include lower order ($r < k$) levels, trends, seasonality, cycles, interventions, where r represents the number of these components to define the elements of the latent factor and k represents the number of observations. It contains the past and current states.
- ε_t can be thought of as measurement error.

The Transition equation

- The transition equation formulates the evolution of the state vector. The state vector α is unobserved; it is a latent variable or underlying factor.
- The transition equation formulates an AR(1) process plus a regression on the innovation.
- That is why this Markov process is sometimes called a Hidden Markov process. T is a transition matrix. ξ_t is an evolutionary innovation.

$$\alpha_{t+1} = T\alpha_t + \xi_t \qquad \xi_t : NID(\mathbf{0}, \sigma_\xi^2)$$

A note on notation

- Many authors use S_t to refer to the state vector instead of α_t. Mark Watson used this at his NBER lectures . Ruey Tsay uses it. Mike West and Jeff Harrison use θ.
- We will use the Koopman syntax formulae to avoid confusion and to be consist.
-

There are other forms of state space models

- This configuration of the state space model is a multiple source of error model. Each equation has its own error term.
- There is also a single source of error model developed by Keith Ord, Rob Hyndman, Anne Koehler, and Ralph Snyder. They call their models innovations models, but are merely single source of error state space models.
- I show both models (in the form of a local level model) on the next page.

SSOE v. MSOE state space models

$$Multiple\ source\ of\ error:$$

$$transition\ eq.: \quad \alpha_{t+1} = T\alpha_t + \xi_t \qquad \xi_t : NID(\mathbf{0}, \sigma_\xi^2)$$

$$measurement\ eq.: \quad y_t = Z\alpha_t + \varepsilon_t \qquad \varepsilon_t : NID(\mathbf{0}, \sigma_\varepsilon^2)$$

$$Cov\begin{pmatrix} Q \\ R \end{pmatrix} = \begin{pmatrix} Q & S \\ S' & R \end{pmatrix} \qquad Q = \sigma_\xi^2,\ R = \sigma_\varepsilon^2$$

$$Single\ source\ of\ error:$$

$$transition\ eq.: \quad \alpha_{t+1} = T\alpha_t + g\varepsilon_t \qquad \varepsilon_t : NID(\mathbf{0}, \sigma_\varepsilon^2)$$

$$g = scalar\ factor$$

$$measurement\ eq.: \quad y_t = Z\alpha_t + \varepsilon_t \qquad \varepsilon_t : NID(\mathbf{0}, \sigma_\varepsilon^2)$$

Other state space models use different estimation algorithms

- The extended Kalman filter: This uses nonlinear functions in lieu of the system matrices.
- The unscented Kalman filter: This checks for higher order moments as well.
- Efficient Bayesian estimation: Uses an simple exponential smoother as a transition equation and a factor analysis as a measurement equation.
- Wavelet based estimation (which I won't cover here)
- The MCMC estimation: Bayesian simulation with Gibbs sampling, Metropolis-Hastings sampling.
- The particle filter: Uses importance sampling resampling for MCMC.

Historical development of DKF:

Other types of state space models use different algorithms to obtain initial values (c=covariance matrix of the state vector, d=that of the innovation)

- Schweppe (1965) developed the Kalman filter approach to evaluating the likelihood(DeJong, 1988,2).
- Rosenberg's (1973) showed that if C=0, the ml estimator of mu can be explicitly displayed and concentrated out of the likelihood(Ibid.)
- Schweppe (1973) recommends using the precision rather than the variance as the criterion.
- Harvey and Phillips(1979) propose initiating the Kalman filter with a very large covariance matrix.
- Ansley and Kohn (1985) show that the information filter is fragile and numerically inefficient.

Historical development of DKF

- DeJong's advocates the basis for the diffuse prior (1988) for nonstationary series with a method easy to evaluate with the Kalman filter by using the innovations and the covariance matrix of the innovations obtained from the fixed point smoothing algorithm (Ibid,166). Yet one has to assume that C=nonsingular.
- DeJong(1991) advocates use of the diffuse Kalman filter mentions using the Generalized inverse when inversion of C becomes difficult. He shows that the DKF can be collapsed to the regular KF after a few iterations using an augmented state vector. Shows the DKF can be used for Dsmoothing too.
- Koopmans's methods: employ splines and random walks.
- Other approaches: Extended Kalman filter: nonlinear processes by using nonlinear functions.
- MCMC approaches: Gibbs sampler and MH method.
- Importance sampling. Importance resampling.
- Particle filter: nonlinear and nonGaussian

Other forms of state space models may use different types of variable processing

- Centering: if we center we gain a df but lose our sense of location. We reduce probability of multicollinearity.
- Standardizing: loses scale as well as location but renders variables with different metrics comparable.
- Normalizing seasonal components: Should we or should we not? Why not just use s-1 dummy variables? Hyndman et al. recommend normalization with multiplicative models (
- Partial normalization

We focus on the Harvey MSOE and the later MCMC models

- MSOE model is the model used in Stamp, in SAS
- SSOE model is used in R
- MCMC is used in R.
- In order to delve into this matter in sufficient depth, given the time provided, we have to focus on one primary method.

Initial values

- The starting values may be taken as parameters of a prior distribution. A prior mean and variance are necessary to define a Gaussian distribution.
- The Kalman filter needs a prior mean and variance to begin the analysis..
- These values must be tractable for the system to function adequately. They must not be unrealistic. If they are unrealistic, the system may fail to converge upon a solution.
- When the initial situation is essentially unknown, we say that our knowledge of it is diffuse or vague. While Harvey and Koopman tend to use this approach, others may attempt to use a random seed.

Bayesian sequential updating

- A weighted average of the previous or prior values and the current data are used to obtain a posterior predictive estimation– that is, to obtain a one-step ahead forecast.
- This will be elaborated soon.

Locally weighted averaging

- The weights used are precisions. Precisions are inverses of variances. The less a person knows, the larger the variance by which his estimates of the prior state are divided (and thereby weighted). The less he knows, the lower the weight accorded his estimate. Hence, the larger the variance divided into his estimate to weight it.
- The more knowledge a person has , the less the variance in his estimates. When his variance is inverted to obtain the precision weight of his estimates, we observe that the greater the precision of his estimates, the smaller his variance . The smaller the variance and the greater the precision, the more weight is accorded to his estimates.
- The weights reflect the amount of ignorance or assurance about a condition when the averaging is performed in order to compute the new position.
- Weights in the Kalman filter are like weights in a locally weighted average within a lowess estimation. Such weighting is used in the computation of a local level or a local linear trend, etc.

Bayesian sequential updating

- According to Bayes' theorem, when the conditional likelihood, represented by the sample, is multiplied by the prior probability distribution (which is sometimes assumed to be known by the scientist familiar with the literature) yields a joint posterior probability distribution.
- From the posterior distribution, the moments can be computed.

The *weighted average of the mean* of the DGP is the formula for a simple exponential smoother

- We call the parameter of interest, theta, Θ

$$E(\hat{\theta} \mid Y) = \delta\mu_{prior} + (1-\delta)Y \qquad a\ simple\ exponential\ smoother$$

$$where \qquad controls\ the\ evoluntionary\ process\ \ ARIMA(0,1,1)$$

$$E(\hat{\theta} \mid Y) = estimate\ of\ the\ posterior\ mean,\ given\ the\ data$$

$$Y = sample\ data$$

$$\mu_{prior}$$

$$0 \leq |\delta| \leq 1$$

$$\delta = \frac{\sigma^2}{\sigma^2 + \tau^2} \qquad an\ intraclass\ correlation\ coefficient$$

$$where$$

$$\tau^2 = variance\ of\ prior\ distribution$$

$$\sigma^2 = variance\ of\ sample\ distribution$$

This provides the basis upon which Hyndman, Ord, Koehler, and Snyder develop their approach.

- They attempt to base Kalman smoothing on exponential smoothing. They treat the Kalman filter as though it is a more sophisticated form of the exponential smoother.

Bayesian shrinkage

$$\text{Because } \delta = \frac{\sigma^2}{\sigma^2 + \tau^2}$$

where

$\tau^2 =$ *variance of prior distribution*

$\sigma^2 =$ *variance of sample distribution*

$\delta \propto$ *relative precision of the sample and the prior*

precision is the reciprocal of the variance, whereas

$\sigma^2 =$ *measure of scale, the baseline against which the precision is compared.*

Bayesian Shrinkage-ctd.

As $\tau \uparrow$ *wrt* σ, *the weight given the prior distributional assumption declines and the sample is given more weight.*

Conversely, as $\sigma \uparrow$ *wrt* τ, *the weight given the prior distributional assumption increases and the sample is given less weight.*

In general, there is some shrinkage of the posterior sample mean toward the prior.

$\delta =$ *Bayesian shrinkage factor, measuring the proportion by which the sample mean is shrunk back toward the prior mean.*

bias and variance s.t. the MSE is minimized.

If we assume Gaussianity

(Carlin and Lewis, 2009, 3rd edition, 17ff):

$We\ know\ that\ \ f(x\,|\,\theta) = N(y\,|\,\theta,\sigma^2),$

$so\ we\ can\ take\ our\ hyperparameters\ (the\ mean\ \mu\ )$

$and\ (variance\ \tau^2)\ of\ our\ prior\ and\ plug\ them$

$into\ the\ formula\ for\ the\ weighted\ average\ and\ obtain:$

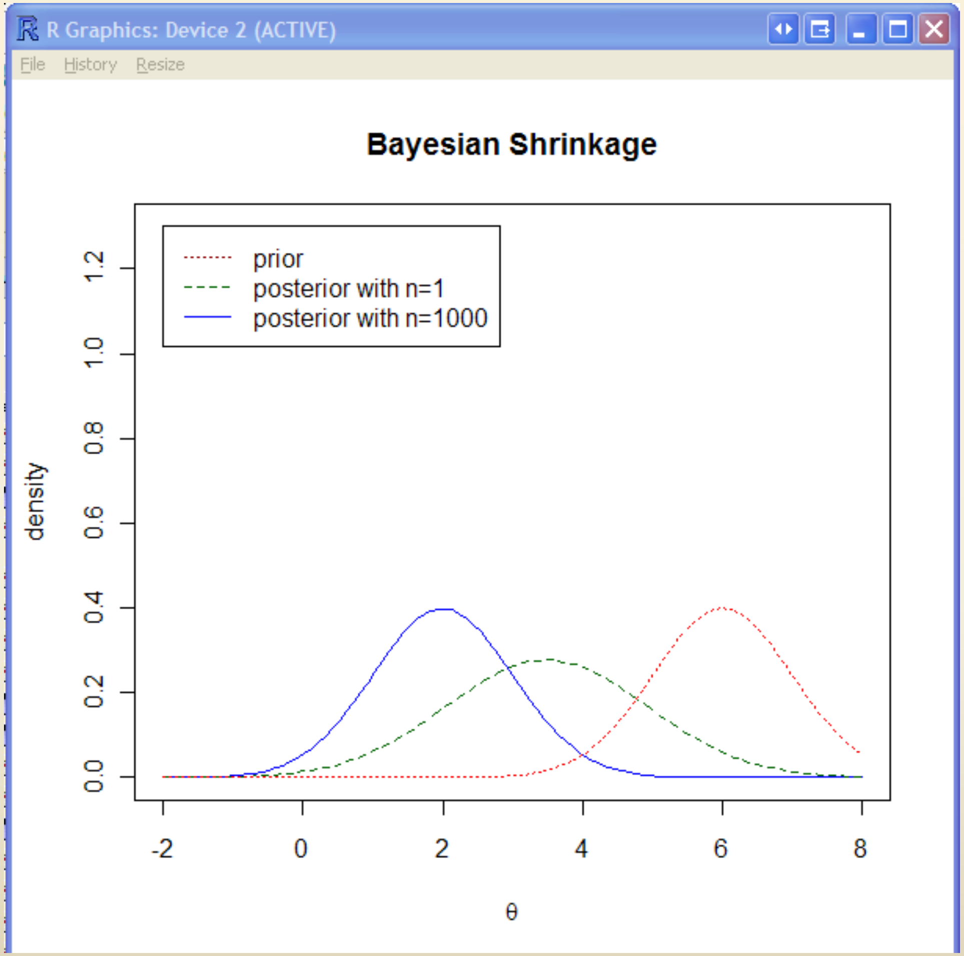
The updating would be performed with a simple weighted average of our sample with our prior distributional mean and variance assumptions:

$$p(\theta\,|\,y) = N\left(\theta \left| \frac{\left(\frac{\sigma^2}{n}\right)\mu + \tau^2 y}{\left(\frac{\sigma^2}{n}\right) + \tau^2}, \frac{\left(\frac{\sigma^2}{n}\right)\tau^2}{\left(\frac{\sigma^2}{n}\right) + \tau^2} \right.\right)$$

which simply reduces to:

$$p(\theta \mid y) = N\left(\theta \left| \frac{\sigma^2 \mu + \tau^2 y}{\sigma^2 + \tau^2}, \frac{\sigma^2 \tau^2}{\sigma^2 + \tau^2} \right. \right)$$

Shrinkage in accordance with relative ample size
(Carlin and Lewis, 2009, 17-18)



Classical state space assumptions

- Gaussianity
- Independence of observations in the residual distributions
- Homoskedasticity
- Stationarity
- Serially uncorrelated disturbances of components
- System matrices were time-invariant
- Large sample for asymptotic consistent estimation
- With minimum mean square estimator
- Information available from past observations
- Reasonable initial values
- Performs best linear prediction

What are the implications of these assumptions on the working of the Kalman filter?

- Why is Gaussianity presumed? Normal distributions of the innovations may be necessary for the proper operation of the maximum likelihood estimation. The formula for such estimation comes from the knowledge of the normal distribution parameters. They permit the construction of conventional prediction intervals.
- Independence of observations in the distributions precludes the estimation of one equation from improperly influencing that of the other equation.
- Homoskedasticity may be necessary to define the variance of the processes and confidence intervals around the estimate of the mean. However, GLS can be applied to handle possible deviations from homoskedasticity.

Implications of the assumptions for model estimation and fitting

- Stationarity: Before the development of the diffuse prior or the information filter, this used to be necessary in order to keep the eigenvalues from residing on the unit circle where variances become infinite and distributions become undefined, as some matrices fail to invert. If variances approach infinity, confidence interval construction becomes impossible. Models could then become unstable and forecasting can become impossible.

Assumption implications

- Serially uncorrelated errors of the components prevent bias in the significance testing creeping in from correlated components.
- Matrices were time-invariant: this permitted arrival at a steady-state where moment estimates could be generated.

State Space extensions

- Use of the information filter instead of the Kalman filter.
- Enhancement of basic concepts: from 2 moments to higher moments.
- Incorporation of the regression effects
- Incorporation of time-varying parameters
- Augmentation of the filter to overcome nonstationarity
- Use of QML and MCMC to overcome the requirement of Gaussianity
- Development of extended Kalman filter to handle nonlinearity

Smoothing
Harvey, A.C. and Priotti, T. (2005,10)

- Disturbance smoothing provides estimation of errors, particularly those in the measurement model.
- The main purpose of this smoothing is signal extraction.
- Standardized smoothed estimates of those errors are called auxiliary residuals

Smoothing algorithms

- Fixed interval smoothing
- Fixed point smoothing
- Smoothing splines and nonparametric regression
  - Koopman quotes Green and Silver who say that smoothing splines are equivalent to signal extraction(Harvey and Priotti, 2005, 11).

Some Historical Background

- The development of a new paradigm in time series has taken place since the early 1970s.
- This approach can handle nonstationary series and missing values, unlike the classical Box Jenkins model developed in the 1970s. The new paradigm is called a state space model.
- Rosenberg (1973) and DeJong(1988,1991) had developed a procedure for diffuse initialization by augmenting the observed vector.
- State Space Models were developed by Rudolf Kalman in 1960 as well as by Rudolf Kalman and Bucy in 1961.

Historical background continued

- Andrew C. Harvey (1983) introduced them to econometrics.
- They have since evolved into different forms. There is the multiple source of error model that was developed.
- The method by which estimation could be done at first seemed to depend on stationary series.
- Since then DeJong and others have developed the Augmented Kalman filter that can handle nonstationary series.
- Recently a single source of error model was formulated by Keith Ord, Ralph Snyder, Rob Hyndman and Ann Koehler out of the exponential smoothing literature.
- More recent developments (Kitagawa, 1996) have included estimation with importance sampling and MCMC simulation. We will explore this type of model tomorrow.

The state space models are based on a nonstationary random walk

- Because this integrated system comprises the basis of the dynamic framework, this system is theoretically capable of handling an integrated or nonstationary system.
- This represents an important shift in the time series paradigm from an ARIMA model that can only analyze stationary series to a state space model that can incorporate nonstationary in its dynamics.

Dynamic Factor Analysis

- If the state vector is considered a dynamic factor, then this approach can incorporate dynamic factors that have been interpreted by Stock and Watson (1991) has coincident economic indictors.
- Whereas early attempts to deal with dynamic factor models required stationary processes, the use of the state space form to model them permits "empirical model building" and nonstationary evolution (Reinsel quoting Aoki, p. 227).

Dynamic factor analysis

- A case of a single common factor configured as part of a state space analysis

$$f_{t+1} = T_t f_t + \eta_t$$

$$y_{t+1} = Z_t f_t + \varepsilon_t$$

$$where$$

$$f_t = \alpha_t$$

Local level model

- This model is basically a random walk plus noise model

$$measurement\ model: y_t = \mu_t + \varepsilon_t \qquad \varepsilon_t \sim NID(0,\sigma^2_\varepsilon)$$

$$transition\ model: \mu_{t+1} = \mu_t + \eta_t \qquad \eta_t \sim NID(0,\sigma^2_\eta)$$

Epsilon and eta are two white noise series that are not correlated with one another.

Tsay maintains that the initial values of eta and epsilon are known or given. They are not correlated with either of the two error terms.

Although y_t is observed, mu is not. It is a latent or hidden construct, sometimes referred to as a factor. Epsilon is observable but uncorrelated noise.

A dynamic local level model

- μ is a latent variable or factor that is not directly observable. Its condition at time t is called the state. η is the unobserved error of this random latent variable. We assume it to be normally and independently distributed. For this reason, this process is sometimes called the state or transition equation. This μ is called a trend even though it has no slope.
- The measurement equation related the observed indicator, y_t, to it. The noise gives rise to random variation in the indicator and is presumed to be normally distributed.

The local level model and the Kalman filter

Ruey Tsay (2005) Analysis of Financial Time Series 2nd ed., chapter 11.

- We need to know the conditional mean and the conditional variance of a process that is normally distributed over time.
- What we mean is that if our process is Gaussian or normally distributed, the mean and the variance are sufficient to define the normal distribution: hence, they are called sufficient statistics.

Conditional probability and its implications

- The conditional mean is the $\mu_{t|j}$, conditional on the values of y_t in information set Ψ_j from times $_{t-1}$ to $_{t0}$.
- The conditional variance is $\Sigma_{t|j}$ =Var(μ_t |Ψ_j).
- $y_{t|j}$ = the conditional mean of y_t, given Ψ_j.
- Suppose the $v_t = y_t – y_{t|t-1}$ and V_t=Var(v_t|Ψ_{t-1}).
- These are respectively, the one-step-ahead forecast error and the forecast error variance, given the information set.
- The forecast error is independent each time it occurs and the conditional variance is also the unconditional variance so Var(v_t)= Var(v_t|Ψ_j)

How is the observed variable related to the latent state?

- Tsay (2005, 494) explains the link between the latent and the observed variable in the local level model:

$$y_{t|t-1} = E(y_t \mid \psi_j) = E(\mu_t + e_t \mid \psi_j) = E(\mu_t \mid \psi_{t-1}) = \mu_t$$

Kalman filtering

- recursions for the principle steps of the Kalman filter (Lutkepohl, H. 2005, 627):
  - Initialization
  - Prediction
  - Correction or revision
  - Reiteration to a steady state
  - Forecasting

Initialization step

- In this case starting values have to be provided for both the mean of the state vector and its variance.
- If little is known about the prior distribution or its mean, the mean is customarily set to zero and a diffuse prior is assumed. In order to designate a parameter as diffuse, most programs (particularly, dlm in R, SsfPack in Ox, SsfPack in S-Plus use a -1) in the computer code to designate the parameter as having a diffuse prior distribution.
- There a multiple algorithms for the diffuse prior.

Kalman Filtering

We begin our introduction with a local level model as an example.
When the local level is forecast then the difference between the forecast and the actual can be observed and the forecast error computed:

$$v_t = y_t - y_{t|t-1} = y_t - \mu_{t|t-1}$$

from the forecast error, v_t*, the forecast error variance,* V_t*, can be computed.*

$$V_t = Var(y_t - \mu_{t|t-1} \,|\, \psi_{t-1}) = Var(\mu_t + e_t - \mu_{t|t-1} \,|\, \psi_{t-1})$$

$$= Var(\mu_t - \mu_{t|t-1} \,|\, \psi_{t-1}) + Var(e_t \,|\, \psi_{t-1}) = \Sigma_{t|t-1} + \sigma_e^2$$

The Prediction step

- Predicting (one-step-ahead) the mean and the variance of the state vector. These are standard formula for obtaining moments.

$$a_{t+1} = E(\alpha_{t+1} \mid Y_t) = T_t a_t \ \textit{conditional mean of state vector } \alpha_{t+1}$$

$$P_{t+1} = \operatorname{cov}(\alpha_{t+1} \mid Y_t) = \textit{conditional variance of state vector}$$

$$= \textit{risk or peril associated with it.}$$

$$a_{t+1} = T_t a_{t|t} + R_t \eta_t \qquad \eta_t \sim (\mathbf{0}, Q_t)$$

$$P_{t+1} = T_t P_{t|t} T_t' + R_t Q R_t'$$

where

T_t = *transition matrix*

R_t = *the selection matrix of* η_t

Q_t = *a diagonal matrix of variances of the component(s)*

Revision or Correction step

- Using the measurement equation

$$From\ y_t = Z\alpha_t + \varepsilon_t \qquad \varepsilon_t \sim NID(\mathbf{0}, H_t)$$

$$v_t = y_t - E[Z_t a_t \mid Y_t + \varepsilon_t] = y_t - Z\,\alpha_t\ \ with\ v_t = innovation$$

$$we\ can\ obtain\ its\ variance,\ F:$$

$$F_t = \mathrm{var}(v_t)$$

$$If\ \ M = \mathrm{cov}(\alpha_t v_t),\ then\ \ K = M_t / F_t$$

$$\alpha_{t+1} = T\alpha_t + Kv_t$$

$$P_{t+1} = P_t - M_t F_t^{-1} M_t$$

corollary

- Durbin and Koopman (2001,67) show:

$$
\begin{aligned}
&Because\ \ M_t = Cov(\alpha_t v_t) \\
&\qquad\qquad\qquad = E(\alpha_t (Z_t \alpha_t + \varepsilon_t)') \\
&\qquad\qquad\qquad = P_t Z_t^{'} \\
&and\ \ because\ F_t = \text{var}(v_t) = E[(Z_t \alpha_t + \varepsilon_t)(Z_t \alpha_t + \varepsilon_t)'] \\
&\qquad\qquad\qquad\qquad = Z_t P_t Z_t ' + H_t \\
&K = T_t M_t F_t^{-1} = T_t P_t Z_t^{'} F_t^{-1}. \\
&From\ the\ \ page\ before\ last,\ we\ obtained: \\
&P_{t+1} = P_t - M_t F_t^{-1} M_t = P_t - P_t Z_t^{'} F_t^{-1} Z_t P_t
\end{aligned}
$$

The correction (revision) step contd.

- It is assumed that the prediction errors are not only not serially correlated, they are not correlated with the state either.
- The error multiplied by the Kalman gain corrects the mean and variance of the state vector from the prediction variance to obtain the proper estimate of the state variance (Durbin and Koopman (2001, 66-67); Hyndman et al. (2009,189); Lutekepohl, 2005, 627).

The forecasting step

- Forecasting is merely an extension of the filtering. It is done after the optimum model has been attained and diagnosed as acceptable.
- We will delve into different methods of forecasting later.

What are the system matrices?

Koopman, Shephard, and Doornik (2008, 9)

$$\underset{m+N\,x\,1}{\begin{pmatrix}\alpha_{t+1}\\ y_t\end{pmatrix}} = \underset{m+N\,x\,1}{\begin{pmatrix}d_t\\ c_t\end{pmatrix}} + \underset{m+N\,x\,m}{\begin{pmatrix}T_t\\ Z_t\end{pmatrix}} \underset{m\,x\,1}{\alpha_t} + \underset{m+N\,x\,r}{\begin{pmatrix}H_t\\ G_t\end{pmatrix}} \underset{rx1}{\varepsilon_t}$$

$$m = dimension\ of\ the\ transition\ equation$$

$$N = dimension\ of\ the\ measurement\ model$$

$$\begin{pmatrix}\alpha_{t+1}\\ y_t\end{pmatrix} = state\ vector \qquad \delta = \underset{(m+N)x1}{\begin{pmatrix}d_t\\ c_t\end{pmatrix}} = constant\ vector$$

$$\Phi_t = \underset{(m+N)\ x\ m}{\begin{pmatrix}T_t\\ Z_t\end{pmatrix}} = transition\ matrix$$

$$u_t = \underset{(m+N)xr}{\begin{pmatrix}H_t\\ G_t\end{pmatrix}} \varepsilon_t \sim NID(\mathbf{0}, \Omega_t) \qquad \Omega_t = \underset{(m+N)x(m+N)}{\begin{pmatrix}HH' & HG'\\ GH' & HH'\end{pmatrix}}$$

$$where\ n = number\ of\ observations$$

$$r = \dim ension\ of\ the\ disturbance\ vector$$

We can define, constrain, or limit parameters in these matrices

Most matrices start with an m before their name. This is a notational convention of SsfPack.

We can decide whether these matrices will be time-varying or constant. We index these Phi, Omega, and sigma matrices by J. All elements within are = -1 except those that vary with time.

We can define whether these elements are known or unknown, to be initialized as diffuse or not.

We can insert - 1 to indicate that the element will receive diffuse initialization or not.

Input to Stsm matrix

Ibid, 24

mStsm				
Cmp	Col 1	Col 2	Col 3	Col 4
Level	σ_{η}	*0*	*0*	*0*
Slope	σ_{ζ}	*0*	*0*	*0*
Trend	σ_{ζ}	m	*0*	*0*
Seas _dummy	σ_{ω}	s	*0*	*0*
Cycle0	σ_{ψ}	λ_{c}	ρ	*0*
:	M	M	M	M
Cycle9	σ_{ψ}	λ_{c}	ρ	*0*
BWCYC	σ_{ψ}	λ_{c}	ρ	m
Irregular	σ_{ξ}	*0*	*0*	*0*

The local level model

and its components

$$\mu_{t+1} = \mu_t + \eta_t \qquad \eta_t \sim NID(0, \sigma_\eta^2)$$

$$y_t = \mu_t + \varepsilon_t \qquad \varepsilon_t \sim NID(0, \sigma_\varepsilon^2)$$

where

$\mu_t = unobserved\ local\ level$

$y_t = observed\ response$

$\eta_t = error\ of\ evolution\ or\ transition$

$\varepsilon_t = the\ irregular\ component$

$(error\ of\ measurement)$

Formulating the state space local level model with SsfPack

The most elementary models only require specification of the mPhi and mOmega .

```
#include <oxstd.h>
#include <c:\Program Files\OxMetrics51\Ox\packages\ssfpack\ssfpack_ex.h>

//   Local level Model


main()
{
   decl mPhi, mSigma, mOmega;
   GetSsfStsm
   (<CMP_IRREG, 1.0, 0, 0, 0;
    CMP_LEVEL, .5, 0, 0, 0 >,


   &mPhi, &mOmega, &mSigma);
   format ("%#6.2g");
   println("Local Level Model ");

   println("                              ");
   print("Phi = ",mPhi, " Omega = ",mOmega, " Sigma = ",mSigma);

}
```

Local Level configuration of system matrices in Ox

```
--------------- Ox at 21:20:48 on 09-Nov-2009 ---------------

Ox Professional version 6.00 (Windows/U/MT) (C) J.A. Doornik, 1994-2009
Local Level Model

Phi =
   1.0
   1.0
 Omega =
  0.25  0.00
  0.00   1.0
 Sigma =
  -1.0
  0.00
--------------- Ox at 21:25:01 on 09-Nov-2009 ---------------
```

Is the level fixed or random?

If the level is fixed, it has no error term. If a mathematical formula determines the level without measurement error, this might be possible.

If the level is fixed, there will be no variation in the error term. In that case, the variation of the error (σ_{η}^2 located in the Ω matrix) term for the level can be set to zero.

This condition is called that of a smooth trend.

In any case, this is a very flexible model.

Component Loading into State Vector

- Koopman et al. (2006, p.144) show how components load into the State vector for a model with a local level, trend, and quarterly seasonal component (3 dummy variables):

$$y_t = (10100)\alpha_t + (1\ 0\ 0)\varepsilon_t$$

where

$$\alpha_t = \begin{pmatrix} u_t \\ \beta_t \\ \gamma_{1,t} \\ \gamma_{2,t} \\ \gamma_{3,t} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & -1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \alpha_{t-1} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \varepsilon_t$$

$$\varepsilon_t = \begin{pmatrix} \eta_t \\ \zeta_t \\ \omega_t \end{pmatrix}$$

Case 3: Local Level Model Measurement Equation

$$y_t = \mu_t + \varepsilon_t$$

$$where\ \ y_t = \ln(MNAtotal)$$

$$\mu_t = level \quad for\ t = 1,...,n$$

$$\varepsilon_t = \ error\ or\ disturbance$$

$$if\ \ \mu_t = \alpha_t \ \ where\ \alpha_t = random\ walk$$

where all random variables are normally distributed

and ε_t *has constant variance.*

Case 2: Local Level Model + Interventions Measurement Equation

$$y_t = \mu_t + \sum_{i=1}^{k}\sum_{\tau=0}^{q}\Delta_{it} I_{t-\tau} + \varepsilon_t$$

$where\ y_t = \ln(MNAtotal)$

$\mu_t = trend \quad for\ t = 1,...,n$

$I_{t-\tau} = Intervention\ (outlier,\ level\ shift, slope\ shift)$

$\tau = time\ lag$

$\varepsilon_t =\ error\ or\ disturbance$

$if\ \mu_t = \alpha_t\ where\ \alpha_t = random\ walk$

where all random variables are normally distributed

and ε_t *has constant variance.*

Case 1: Local Level Model + time varying parameters + interventions

$$y_t = \mu_t + \sum_{i=1}^{k}\sum_{\tau=0}^{q}\Delta_{it}x_{i,t-\tau} + \sum_{j=1}^{h}\lambda_j I_j + \varepsilon_t$$

$$where\;\; y_t = \ln(MNAtotal)$$

$$\mu_t = \mu_{t-1} + \beta_{t-1} + \eta_t = \; trend \;\; for \;\; t = 1,\ldots,n$$

$$\beta_t = \beta_{t-1} + \varsigma_t \;\; = slope$$

$$\sum_{i=1}^{k}\sum_{\tau=0}^{q}\Delta_{it}x_{i,t-\tau} = time\;varying\;\; parameter\,estimates$$

$$\sum_{j=1}^{h}\lambda_j I_j = Interventions(level\,shifts,\; slope\;\; shifts, outliers)$$

$$\varepsilon_t = \;\; error\;\; or\;\; disturbance$$

Stacked Matrix Formulation

- Local level model (random walk plus noise) (and Koopman (2004, 287); Zivot and Wang(2005, 521).

$$
\begin{aligned}
&If\ \alpha_{t+1} = T_t\alpha_t + R_t\eta_t, \quad \eta_t = iid\ N(0,\sigma^2_{\eta_t})\ transition\ equation\\
&\quad y_t = Z_t\alpha_t + \varepsilon_t, \qquad \varepsilon_t = iid\ N(0,\sigma^2_{\varepsilon_t})\ measurement\ equation,\\
&then\\
&\begin{pmatrix}\alpha_{t+1}\\ y_t\end{pmatrix} = \begin{pmatrix}T_t\\ Z_t\end{pmatrix}(\alpha_t) + \begin{pmatrix}R_t\eta_t\\ \varepsilon_t\end{pmatrix}\\
&where\\
&y_t = \mu_t + \varepsilon_t\ , \quad \varepsilon_t = \ iid\ N(0,\sigma^2_{\varepsilon_t})\\
&\alpha_1 \sim N(\alpha_1, P_1)\\
&NB: In\ a\ local\ level\ Model: T_t = I,\ so\\
&\alpha_{t+1} = \alpha_t + R_t\eta_t
\end{aligned}
$$

The Kalman filter

The Kalman filter is the process by which the forecasting or filtering is performed.

It takes the starting values and applies its AR(1) filter to predict the next state of the latent factor (the condition of that factor at the next time period).

It corrects its prediction a measurement of that state as soon as the data become available.

The combining of the estimation with the data is performed by a Bayesian or sequential updating that is based on a weighted averaging.

It employs sequential updating of its estimates of the future state with a factor analysis upon the latent variable it encounters.

This updating process prevents the process from going too far awry.

The Kalman filtering process

$Initial\ \ state: for\ \ the\ \ mean\ \ of\ \ the\ \ state\ \alpha_0\ \ and\ its\ variance\ \ P_0$

$Updating\ \ process:$

$$for\ \ the\ \ mean: \hat{\alpha}_{t+1|t} = \hat{\alpha}_{t||t-1} + B_t X_{t|t-1} + Cov(\alpha_t \varepsilon_t)(Var(\varepsilon_t)^{-1}$$

$$\hat{\alpha}_{t+1|t} = \hat{\alpha}_{t||t-1} + B_t X_{t|t-1} + \kappa_t \varepsilon_t$$

$$where\ \ \kappa_t = Kalman\ \ gain$$

$$X_{t|t-1} = exogenous\ \ control\ \ series$$

$$for\ \ the\ variance\ estimator\ of\ \ \alpha_{t+1|Y} : \hat{P}_{t+1||t} = T_t P_{t|t-1} T_t^{'} + RQR_t^{'}$$

$Estimation\ is\ \ performed\ \ by\ \ a\ \ mean\text{-}square\ error\ \ minimization$

$process\ \ (Reinsel, 2008,\ 229)$

Kalman filtering process

$$P_{t+h|t} = E\left[(\alpha_{t+h} - \hat{\alpha}_{t+h|t})(\alpha_{t+h} - \hat{\alpha}_{t+h|t})'\right]$$

The predictive error variance is minimized in the process of filtering.

Andrew Harvey (1989,106)

presents the ***Prediction Equations***

$$Given\ P_t\ and\ \alpha_t,$$

$$\alpha_t\ \ is\ optimally\ estimated\ as$$

$$\alpha_{t|t-1} = c_t + T_t\alpha_{t-1}$$

$$where$$

$$T_t = matrix\ of\ Markovian\ transition\ coefficients$$

$$c_t = some\ constant$$

The Error Covariance Matrix $P_{t|t-1}$ *Ibid.*

$$P_{t|t-1} = T_t P_{t-1} T_t' + R_t Q_t R_t' \qquad for \ \ t = 1,\ldots,T$$

which is an asymptotic variance estimator
is amenable to an
eigenvalue decomposition

These prediction equations have updating equations

if centering renders c_t=0

$$The\ innovation$$

$$\varepsilon_t = y_t - Z\hat{\alpha}_t \qquad \varepsilon_t \sim N(\mathbf{0}, H_t)$$

$$\alpha_{t+1} = T_t\alpha_t + \varepsilon_t \qquad \eta_t \sim N(\mathbf{0}, Q_t)$$

$$\alpha_1 \sim N(\alpha_1, P_1)$$

Kalman Prediction and Updating Equations

$$\left.\begin{aligned}\hat{\alpha}_{t+1} &= T\hat{\alpha}_t + \kappa_t \varepsilon_t \\ &= \hat{\alpha}_{t|} + \kappa_t (Y - Z\alpha_t - d_t)\end{aligned}\right\} \quad prediction\ eqs.$$

$where$

$$\kappa = Kalman\ gain\ \ \&$$

$$d_t = constant\ in\ measurement\ model$$

$$Because\ \kappa_t = P_{t|t-1} Z_t^{'} \left[Z_t P_{t|t-1} Z_t^{'} + H_t \right]^{-1}$$

$$and$$

$$P_t = P_{t|t-1} - P_{t|t-1} Z_t^{'} F_t^{-1} Z_t P_{t|t-1}$$

$where$

$$F_t = Z_t P_{t|t-1} Z_t^{'} + H_t \qquad P_t + \sigma_\varepsilon^2 \quad and \quad \kappa_t = P_t / F_t$$

$$s.t. \kappa_t = P_t / (P_t + \sigma_\varepsilon^2) \qquad (an\ ICC)$$

$$= T_t P_{t|t-1} Z_t^{'} (Var(\nu_t))^{-1} \quad where\ \varepsilon_t = \upsilon_t$$

$$= (T_t P_t Z_t^{'}) F^{-1} \qquad for\ t = 1, \ldots, T$$

The Kalman filter can be expressed in terms of recursive equations

$$In\ \ the\ \ Correction\ \ step:$$

$$\alpha_{t+1|t} = (T_{t+1} - K_t Z_t)\alpha_{t|t-1} + K_t y_t + (c_t - K_t d_t)$$

$$where\ \ K_t = Kalman\ gain\ matrix$$

$$recall\ \ that\ in\ the\ measurement\ equation$$

$$y = d_t + Z\alpha_t + \varepsilon_t \quad with\ d_t = mean\ vector$$

$$and\ that\ we\ are\ subtracting\ the\ error\ K_t\varepsilon_t$$

$$in\ updating\ of\ the\ state\ vector.\ Hence, \alpha_{t+1|t}$$

$$can\ be\ tracked\ back\ to\ \alpha_1$$

By substituting

$$P_t = P_{t|t-1} - P_{t|t-1} Z_t' F_t^{-1} Z_t P_{t|t-1}$$

for the middle term in the first group

on the right-hand side of

$$P_{t|t-1} = T_t P_{t-1} T_t' + R_t Q_t R_t' \qquad for \;\; t = 1, \ldots, T$$

we obtain

the Ricatti equation:

$$P_{t+1|t} = T_{t+1}(P_{t|t-1} - P_{t|t-1} Z_t' F_t^{-1} Z_t P_{t|t-1})T_{t+1}' + R_{t+1} Q_{t+1} R_{t+1}'$$

were

$$F_t = Z_t P_{t|t-1} Z_t' + H_t$$

Stability of the system depends on the existence of a steady-state solution to the Ricatti equation

Andrew C. Harvey writes :

"The steady-state filter is said to be stable if the roots of T are less than one in absolute value. The Kalman filter has a steady-state solution if there exists a time-invariant error covariance matrix which satisfies the Ricatti equation . … If such a solution exists, we can get

$$P_{t+1|t} = P_{t|t-1} = \bar{P}$$

Harvey, A.C. (1989), 118.

An algebraic Ricatti equation formulates the steady-state solution

$$If\ such\ a\ solution\ exists,\ P_{t+1|t} = \bar{P},\ so$$

$$the\ transition\ and\ gain\ matrices\ become$$

$$\bar{T} = T - \bar{K}Z$$

$$and$$

$$\bar{K} = T\bar{P}Z'(Z\bar{P}Z' + H)^{-1}\ \ and\ the\ Ricatti\ equation$$

$$reduces\ to\ an\ algebraic\ form:$$

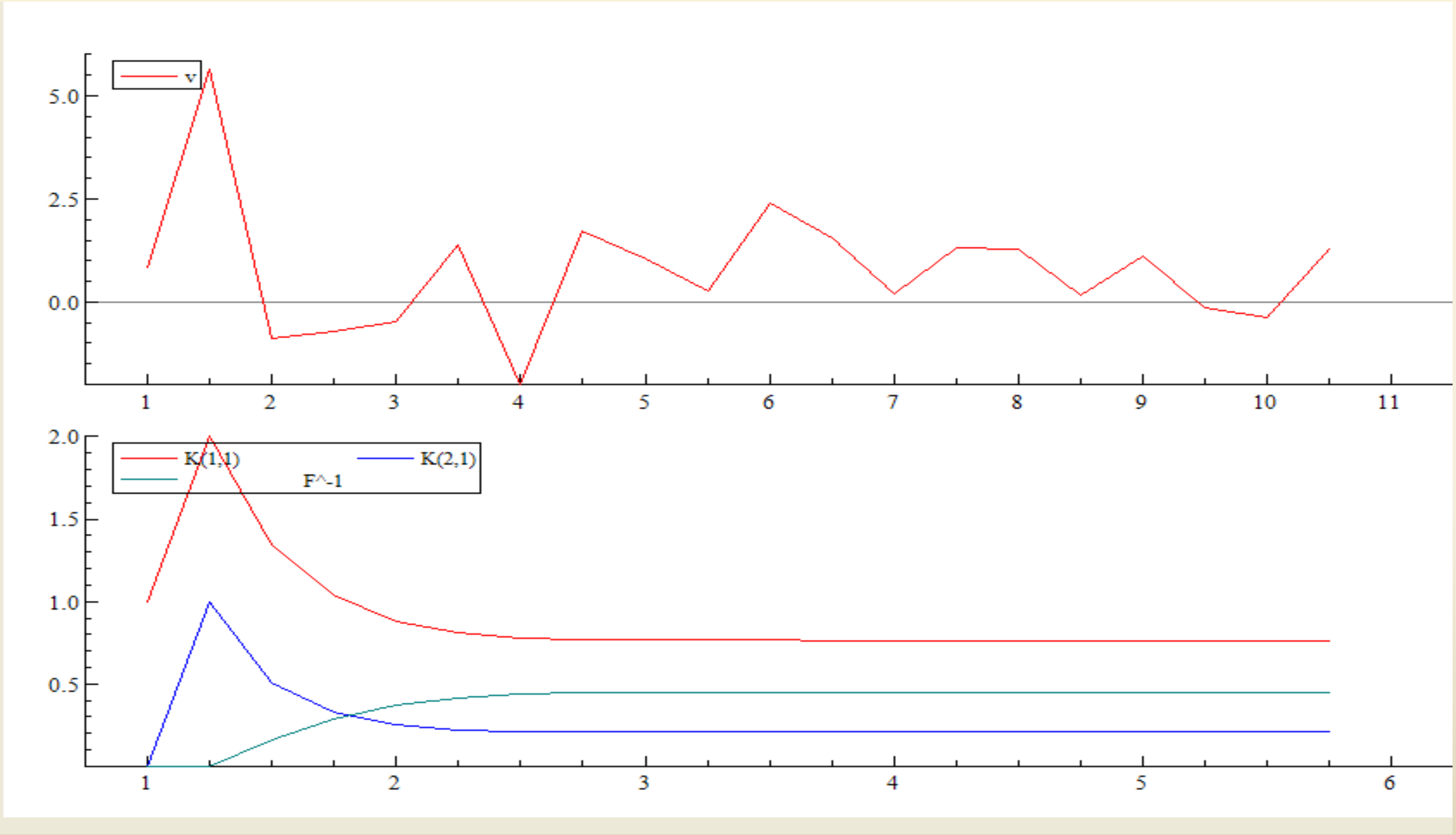
$$\bar{P} - T\bar{P}T' + T\bar{P}Z'(Z\bar{P}Z' + H)^{-1}Z\bar{P}T' - RQR' = \mathbf{0}$$

$$In\ this\ case\ \bar{P} = positive\ semi-definite.$$

If there is a steady state solution

$$\text{the matrix of}$$
$$\text{innovations} \quad \lim_{t\to\infty} F_t = \Sigma = ZPZ' + H$$

With steady-state convergence recursive filtering generates v(t), F^(-1) and kappa



The General State Space Model

$$\alpha_{t+1} = T_t \alpha_t + R\eta_t \quad \eta_t^2 \sim NID(\mathbf{0}, Q_t$$
$$y_t = \quad Z_t \alpha_t + \varepsilon_t \quad \varepsilon_t \sim NID(\mathbf{0}, H_t)$$
$$where$$

$$\alpha_t = \begin{pmatrix} \mu_t \\ \beta_t \\ \gamma_t \\ \psi_t \\ \omega_t \\ \nu_t \end{pmatrix}, \ \Phi_t = \begin{pmatrix} T_t \\ Z_t \end{pmatrix}, \Omega_t = \begin{pmatrix} R_t Q_t R_t^{'} & \mathbf{0} \\ \mathbf{0} & H_t \end{pmatrix} = \begin{pmatrix} \sigma_\eta^2 & \mathbf{0} \\ \mathbf{0} & \sigma_\varepsilon^2 \end{pmatrix}$$

$\Sigma = \begin{pmatrix} \alpha_1 \\ P_1 \end{pmatrix}$ *where* $\mu_t =$ *local level*, $\beta_t =$ *local slope*,

$\gamma_t =$ *seasonality* $\qquad \varepsilon_t =$ *vector of measurement errors*

$\psi_t =$ *cyclicity* $\qquad R_t =$ *selection matrix*

$\nu_t =$ *regression effects* $\qquad Q_t =$ *diagonal matrix of evolution variances*

$\omega_t =$ *intervention effects (outliers, level shifts)*

$P_1 =$ *initial state variance* $\qquad H_t =$ *diagonal matrix of measurement errors*

$\alpha_1 =$ *initial state mean* $\qquad \alpha_t =$ *state vector*

$Z_t =$ *matrix of factor loadings* $\qquad \beta_t =$ *slope component of trend*

$T_t =$ *transition matrix*

The General State Space Model

- Measurement model

$$y_t = \mu_t + \beta_t + \gamma_t + \psi_t + \omega_t + v_t + \varepsilon_t$$

$$where$$

$$\mu_t = local\, level + \ \eta_t \ (level\ error)$$

$$\beta_t = local\ slope + \tau_t (local\ slope\ error)$$

$$\gamma_t = seasonal\ component + \lambda_t \omega_t (seasonal\ error)$$

$$\psi_t = cyclical\ component$$

$$\omega_t = intervention\, component$$

$$v_t = regression\ effects\ component\ +\ \zeta_t (reg\, effect\ error)$$

$$\varepsilon_t = measurement\ error$$

For a multivariate state space model

$$We\ know\ that\ \ for\ a\ steady-state\ solution\ \ in$$
$$a\ state-space\ \ model$$
$$P = TPT' + RQR'$$

$$In\ a\ multivariate\ model$$
$$P - TPT' = RQR'$$
$$P(I - T \otimes T') = RQR'$$
$$vec(P) = (I - T \otimes T')^{-1} RQR'$$

Initial conditions and convergence

1. For time invariant models

    1. Starting values for the mean and variance-covariance matrix of the unconditional distribution of the state vector must be provided.

    2. The transition equation provides the mean

$$\alpha_t = c_t + T\alpha_{t-1} + R\eta_t \qquad \mathrm{var}(\eta) = Q$$

$$c_t = \alpha_t - T\alpha_{t-1} = (1-T)\alpha_t$$

$$\alpha_t = (1-T)^{-1} c_t \;\; is\ the\ mean$$

$$and \;\; P = the\ variance\ \ from\ P = TPT' + RQR'$$

Condition of nonstationarity

T must remain nonsingular for $P_0^{-1} = 0$

The state vector remains stationary in the stochastic process if $\lambda(T) < 1$ and c_t remains constant.

$P = TPT' + RQR'$.

If the variance $(K) = \sigma_k^2$ then $P = \sigma_k^2 / (1 - \rho^2)$

Condition of nonstationarity

If the transition equation is nonstationary, the unconditional distribution is undefined (Harvey, op. cit, 120).

P_0 prior variance → ∞ while the precision drops to zero. Something cannot be divided by a precision of zero and remain mathematically defined.

$$P_0 = \kappa I$$ where k=a nonnegative scalar, with a diffuse prior obtained $$\kappa \rightarrow \infty$$

This means that the initial distribution of the state vector alpha sub zero has a non-informative or diffuse prior. With kappa = infinity, we have reached the limit. We don't need it to be that large. 10^7 or 10^6 can approximate infinity and remain algorithmically tractable.

Information filter

When the variance of P_0 is infinite, the information filter may provide a more stable algorithm than the Kalman filter to apply (Harvey, 120). Hence, the Precision P_0^{-1} =zero. The inversion of F is not required by the information filter. When the dimensions of the state vector are larger than the dimension, m, of the state, the avoidance of inverting a large matrix like F could render the estimation much more efficient.

This method involves triangular structure of the stochastic equations in the covariance matrix, which have a limiting form for infinite variances. This permits the use of this filter for stationary as well as nonstationary series without modification (Hyndman et al., 2009, 189).

The information filter is deployed in Stamp during recursive filtering.

Partitioning the state vector into nonstationary and stationary portions

When the series is nonstationary, the transition matrix can be partitioned to divide the nonstationary from the stationary components. The nonstationary portion can be confined to T_1 *to which the diffuse prior can be applied.*

$$T = \begin{pmatrix} T_1 & \mathrm{M}\, T_2 \\ \mathrm{L} \;\; \mathrm{L} & \mathrm{L} \;\; \mathrm{L} \\ \mathbf{0} & \mathrm{M}\, T_4 \end{pmatrix}$$

where the dimensions of the submatrices are

$\underset{dxd}{T_1}, \quad \underset{dx(m-d)}{T_2}, \quad and \quad \underset{(m-d)x(m-d)}{T_4}$

with $\lambda(T_4) < \mathbf{1}$

(*Harvey*, ***1989***, ***123***).

Matrix partitions for nonstationary and stationary components
(Harvey, 1989, 123)

The covariance matrix P can be conformably partitioned so that

$$P_{1|0} = \begin{bmatrix} kI & \mathbf{0} \\ \mathbf{0} & P \end{bmatrix}$$

and the

Z' matrix can be partitioned as

$$\begin{bmatrix} z_1' & z_2' \end{bmatrix}$$

so

$y = Z_t'\alpha_t + \varepsilon_t$ *the measurement model still holds*

so long as the first d nonstationary elements are observable and T_1 is nonsingular

Even with a nonstationary process, it is possible to show that this system is stable and can converge to a steady state solution

Convergence is exponentially fast even with a nonstationary process.

Harvey suggests partitioning the transition matrix into segments. In one segment, the stationary elements can reside, while in another segment the nonstationary elements can reside.

This permits proper prior assignment to partitions of the initial value for the state vector.

The nonstationary partition can have a diffuse prior while the stationary segment can have a proper prior.

Stability of the process

A necessary and sufficient condition for stability of the state space evolution Is that the characteristic roots of the transition matrix, T, should have eigenvalues with a modulus less than unity (Hamilton,378; Harvey, 114). In other words, this is a condition necessary and sufficient for covariance stationarity.

$$\lambda_j(T) < \boldsymbol{1},\ for\ \ j = \boldsymbol{1}, \ldots, m$$

This means that there are no unit roots in the evolutionary process.

Partitioning the state vector and transition matrix

Harvey suggests conformably partitioning the transition matrix and the state vector :

$$\alpha_t = \begin{bmatrix} \alpha_1 \\ \mathrm{L} \\ \alpha_2 \end{bmatrix} \qquad T = \begin{bmatrix} \underset{dxd}{T_1} & \mathrm{M} & \underset{dx(m-d)}{T_2} & \\ \mathrm{L} & \mathrm{L} & \mathrm{L} & \mathrm{L} \\ \mathbf{0} & \mathrm{M} & \underset{(m-d)x(m-d)}{T_4} & \end{bmatrix}$$

α_1 ← Stationary elements

α_2 ← Nonstationary elements

Estimation

ML

Assumes that observations are Normal and iid. Hence, the likelihood is the product of the individual likelihoods. When logged, the likelihood is the sum of the individual likelihoods.

With time series, that is not the case. Observations are conditional on the previous information set.

$$L(y;\psi) = \prod_{t=1}^{T} p(y_t \mid Y_{t-1})$$

We write the joint density as a function of the conditional density times a prior distribution.

Likelihood of the state space model

Ibid, 120.

The model is estimated by maximizing the likelihood by minimizing the prediction error variance and maximizing the fit with the recursive equations, when the DGP process is NID.

$$LL = -\frac{T}{2}\log 2\pi - \frac{T}{2}\log \sigma_*^2 - \frac{1}{2}\sum_{t=1}^{T}\log |F_t| - \frac{1}{2}\sum_{t=1}^{T} v_t' |F_t|^{-1} v_t$$

where

$$v_t = y_t - \hat{y}_{t|t-1}$$

The first term on the rhs is a constant. The last term on the far rhs requires inversion of a matrix. If the rank of that matrix is of low dimension, the algorithm proceeds quickly. If it is of high order, this process can become time-consuming.

Asymptotic characteristics

The estimator is asymptotically multivariate normal if T is invertible.
The parameters within the parameter space are identifiable.
Derivatives up to order 3 exist within the parameter space of the information set and are continuous.

The term on the far rhs of the prediction error decomposition of the variance reveals that this algorithm proceeds according to a minimization of the mean square error of prediction as well as fit (as F is included in the likelihood function).

Exact Maximum Likelihood

When the model is stationary and the prior is proper, the exact likelihood can be estimated with the formula just given.

Quasi-Maximum Likelihood

Generalized least squares with a White or a Newey-West estimator can be used for multivariate models to handle situations where Kim and Nelson (1999) provide an excellent account of how this works as does Harvey. Deviations from Gaussianity can be handled so long as the distribution is symmetric. Log-normal distributions can still yield approximate likelihood.

GLS can be used to estimate state space models when unknown exogenous variables are added to the measurement model

The state space model can be rewritten in a regression form

$(Harvey, \mathbf{1989}, \mathbf{130})$ *writes that because* $y_t = z'\alpha_t + x'\beta + v_t$

we can rewrite the state space model as related regressions:

$$y_t = x'\beta + e_t$$

$$e_t = z'\alpha_t + v_t$$

If we assume $E(e_t) = \mathbf{0}$, $Var(e_t) = V$

Even if the unknown x' *in the model generates heteroskedasticity, we can solve for*

$\beta = (X'V^{-1}X)^{-1}(X'V^{-1}Y)$ *using this GLS.*

The Properties of V render it amenable to a Cholesky decomposition

Harvey, 1989, 131 writes that because V is positive definite, there is a matrix L, which is lower triangular and has ones in the principle diagonal, which can be pre and post-multiplied by F inverse to yield the inverse of V.

$$V^{-1} = L' F^{-1} L$$

By multiplying those regression equations by L, we obtain a heteroskedastic regression equation that solves for beta with GLS. (See Kim and Nelson, 1999, 20.

The F can be used in GLS for V.

The log likelihood for GLS then becomes

$$LL = LogL - \frac{1}{2}\sum_{t=\tau+1}^{T}\log|F| - \frac{1}{2}\sum_{t=\tau+1}^{T}\log|v_t{}'F^{-1}v_t|$$

$$\beta = (X'F^{-1}X)^{-1}(X'F^{-1}Y)$$

So GLS is functionally equivalent to a maximum likelihood solution for the parameters, though ML usually proceeds by arriving at the mean and the variance through the gradient and the information matrix.

The EM algorithm

The expectation maximization algorithm. It consists of expectation step followed by a maximization step. The algorithm iterates until the likelihood given the data can no longer be improved.

Commandeur and Koopman maintain that this algorithm assures nonnegativity of hyperparameter estimation.

The disadvantage of this algorithm is that it is very slow, especially when there are many parameters to be estimated.

The BFGS algorithm is much faster, but does not assure monotone convergence. A combination of these two algorithms is used to find the proper balance.

Rosenberg's algorithm (1973)

The state vector is partitioned into a stationary sub-vector and a nonstationary sub-vector. The state vector a = Ta* + Ta

B. Rosenberg's algorithm

$$\alpha_{t+1} = Ts_t + T^* s^*_{t-1} + e_t + e_t^*$$

partitions the state vector into 2 subvectors,

one s_t *is stationary and the other* s_t^* *is not.* s_t^* *uses a diffuse prior while* s_t *uses a normal prior.*

The priors are used as starting values for the mean and variance of the state vector.

DeJong's algorithm for diffuse smoothing.

Requires the inversion of large matrices but Rosenberg's does not. Rosenberg just augments the state vector to accommodate the nonstationary components using a diffuse prior for them only. To invert the singular matrix,
He employs a generalized inverse. Eventually, the nonstationary part collapses to the classical Kalman filter
As it becomes stationary and then it proceeds until convergence is attained.

DeJong in 1989 in JASA provides new algorithm for fixed lag smoothing which more efficiently performs diffuse smoothing while covering the degenerate cases and happens to be more computationally efficient. To model diffuseness of beta, he lets beta = $b + B\delta$ where b is a fixed vector, B is a fixed matrix of full column rank and delta is a random vector unrelated to the v(t) and u(t) and the nonsingular covariance matrix $\sigma^2 \Sigma$

Exact initial Kalman filter

Koopman discovered a means of finding the exact initial Kalman filter. This is more computationally efficient when dealing with nonstationary series. (Koopman, S.J. “The Exact Initial Kalman filter and the smoothing of nonstationary time series,” in Harvey and Proietti, (eds.) ,2005, 54). The smoothing of this filter leads to the exact score vector of the initial state vector.

Diffuse Log Likelihood

(Schweppes, 1965), according to Koopman, 2005, 55.

$$Diffuse\ LL(y) = \log L(y) + \frac{m}{2}\log(\kappa)$$

where

$$Log(y) = \ constant + \frac{1}{2}\sum_{t=1}^{n}\log|F_{*,t}^{-} + F_{\infty,t}^{-}| - \frac{1}{2}\sum_{t=1}^{n} v_t^{'} F_{*,t}^{-} v_t$$

Identification

Hamilton (1994, 387) maintains that unless proper constraints are introduced, into the T, Q, G, H, β, and R matrices, the state space model will be unidentified.

Recall that the model is

$$\alpha_t = T\alpha_{t-1} + WX_t + R_t\eta_t \qquad \eta_t \sim NID(\mathbf{0}, Q)$$
$$y_t = Z\alpha_{t-1} + \beta X_t + G_t\varepsilon_t \qquad \varepsilon_t \sim NID(\mathbf{0}, R)$$

So the question arises, how many of what kind of constraints must be applied to identify such a model?

Smoothing

Smoothing is estimation of the signal from the measurement model.

It involves extraction and projection of this signal onto the y vector.

Interpolation is the projection of x(t), or alpha(t), onto the y(t) space

(DeJong, P. *Smoothing and Interpolation with State Space Models*, in

(1989) Harvey and Proietti (eds) (2005), 73).

Smoothing

A set of backward recursions using the output of the Kalman filter, formulated in Durbin and Koopman (2001) *Time Series Analysis by State Space Methods*.
This smoothing is functionally equivalent to the output of a *Weiner-Kolmogorov* filter (Hyndman et al.,2009,225):

$$WK\ \ filter\ (l_{s,t}) = \frac{\alpha^2 y_t}{[1-(1-\alpha)L][1-(1-\alpha)L^{-1}}$$

$$= \frac{\alpha}{2-\alpha}\sum_{j=-\infty}^{\infty}(1-\alpha)^{|j|}y_{t-j}$$

Smoothing can be used for interpolation, signal extraction, residual analysis, deleted residuals, and auxiliary residuals. Residuals can be useful model diagnostics.

Diffuse Smoothing

Smoothing algorithms depend on initial values of the mean and variance of the state vector. If it is assumed that nothing is known about the first state, a noninformative prior distribution may be used from which to obtain these values. Noninformative or diffuse prior distributions are combined by a weighted average with current data to arrive at an estimate. The noninformative prior has a variance that is approximately infinite or extremely large. The precision (the inverse of the variance) is used as the weight given to this part of the weighted average. However, nothing can be divided by zero and remain finite. Therefore, an approximation can be because the convergence properties of the Kalman filter can handle such quantities.

After enough iterations to overcome the impact of the nonstationary elements, the diffuse Kalman filter will collapse to the classical Kalman filter and the filter, if there is a steady-state solution, will then converge toward it.

A diffuse Kalman filter can generate the diffuse smoothing

$$\beta = B + \delta b$$

$B = fixed\ vector\ of\ full\ column\ rank$

$\delta = random\ vector\ unrelated\ to\ u_t\ or\ v_t$

$\quad that\ has\ a\ nonsingular\ covariance$

$as\ \Sigma -> \infty, \Sigma^{-1} -> \mathbf{0}\ as\ the\ state\ variance$

becomes diffuse.

The diffuse projections can be performed by an augmented Kalman filter.

Preliminary State Space Model Analysis

Download the data and record the source, time, date, and study description
Of the dataset.

Be sure the dates are correct for time series data.

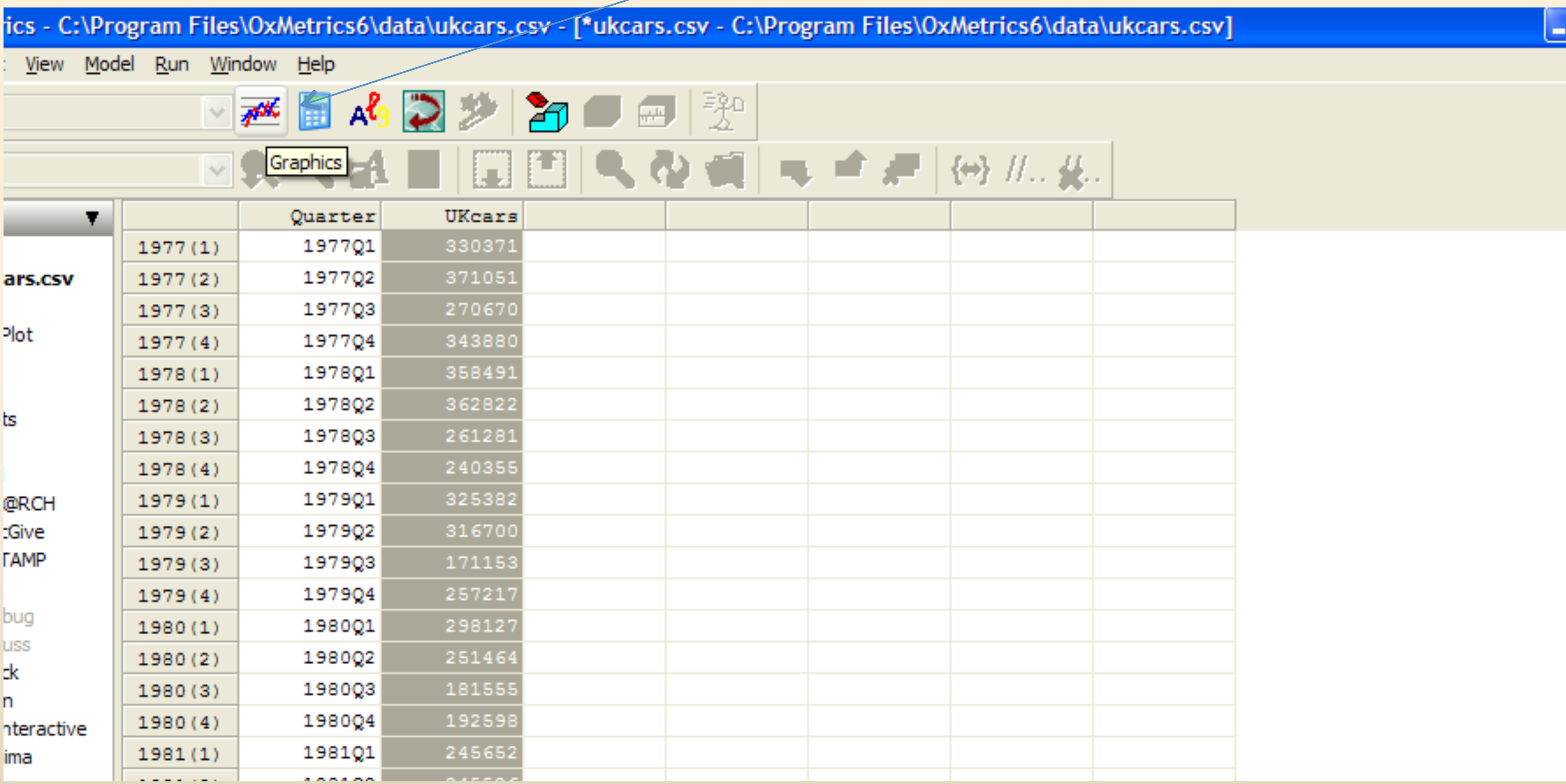
Check for missing values.

Time plot of the data

Look for abnormalities in the data

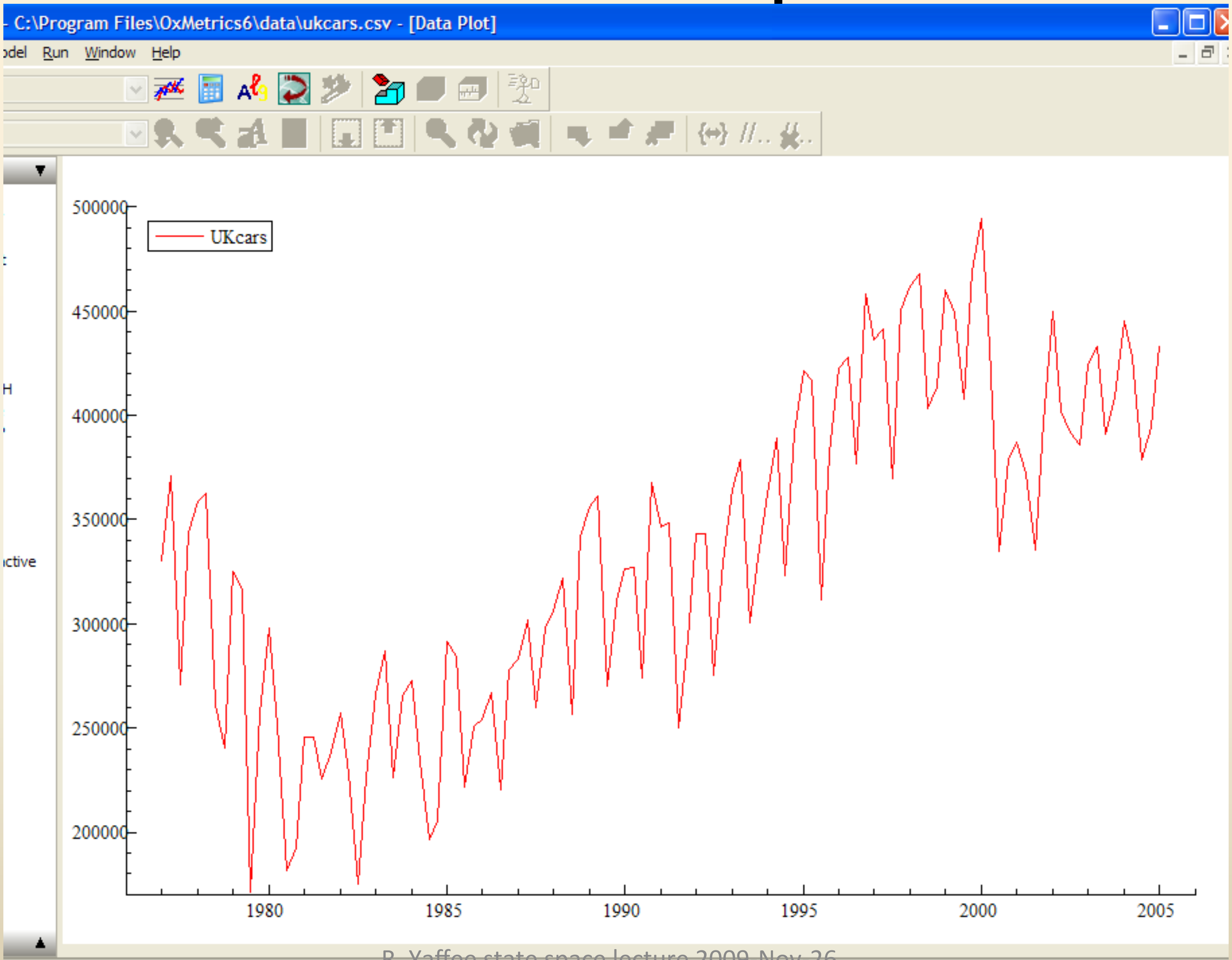
UK cars downloaded from http://www.exponential smoothing.net 11/13/2009

Click here to graph



	Quarter	UKcars
1977(1)	1977Q1	330371
1977(2)	1977Q2	371051
1977(3)	1977Q3	270670
1977(4)	1977Q4	343880
1978(1)	1978Q1	358491
1978(2)	1978Q2	362822
1978(3)	1978Q3	261281
1978(4)	1978Q4	240355
1979(1)	1979Q1	325382
1979(2)	1979Q2	316700
1979(3)	1979Q3	171153
1979(4)	1979Q4	257217
1980(1)	1980Q1	298127
1980(2)	1980Q2	251464
1980(3)	1980Q3	181555
1980(4)	1980Q4	192598
1981(1)	1981Q1	245652

Time series plot



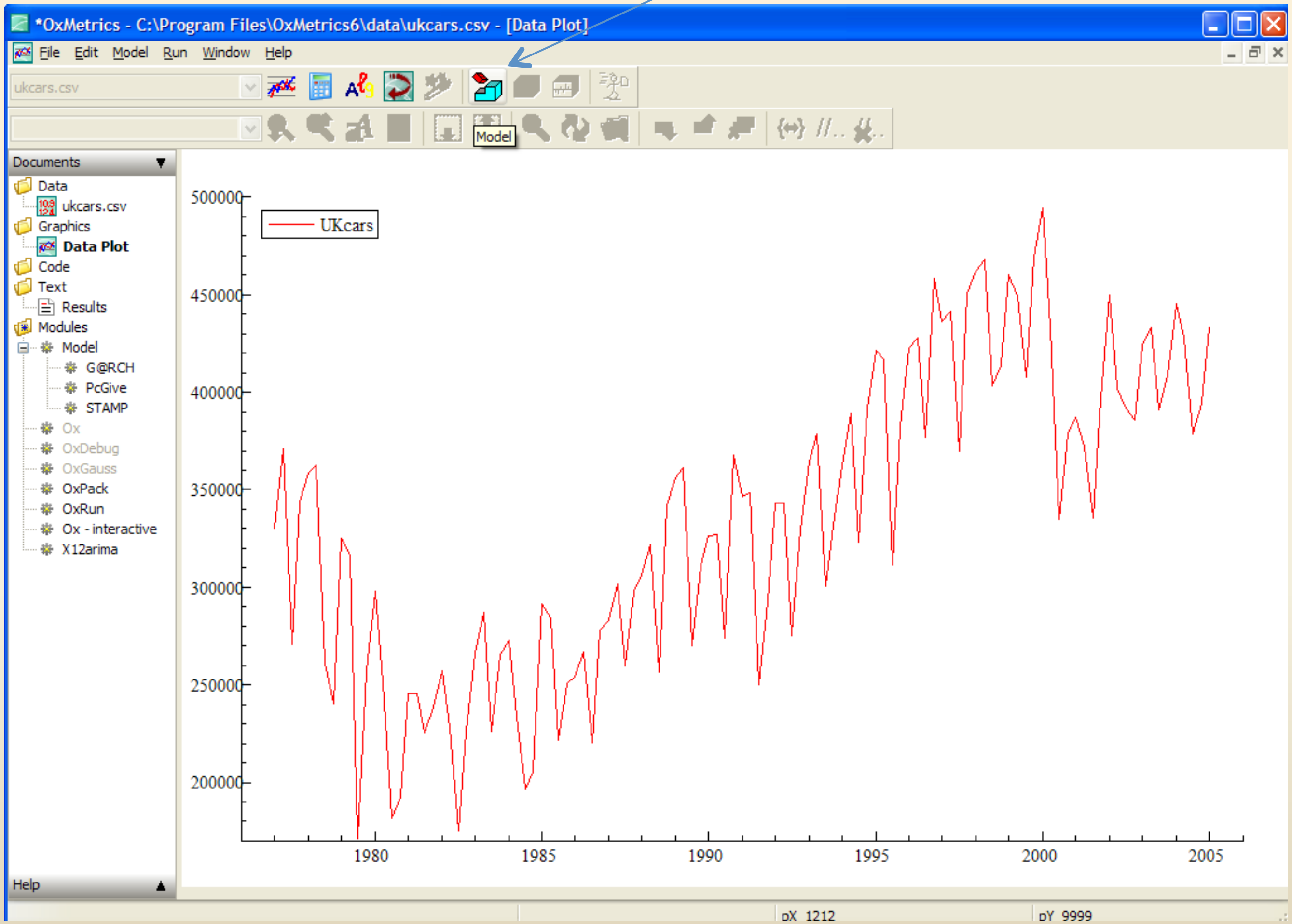
What's potentially problematic in this graph

- There is a regime shift in 1980-1982
- the number of cars stops falling and begins to rise.
- This would be a problem for an ARIMA model but not for a state space model.
- Why?
- There is another level shift in 2000-2001.
- Is this a problem? It would be for an ARIMA model? Is it for a state space model? If so,
- why? If not, why not?

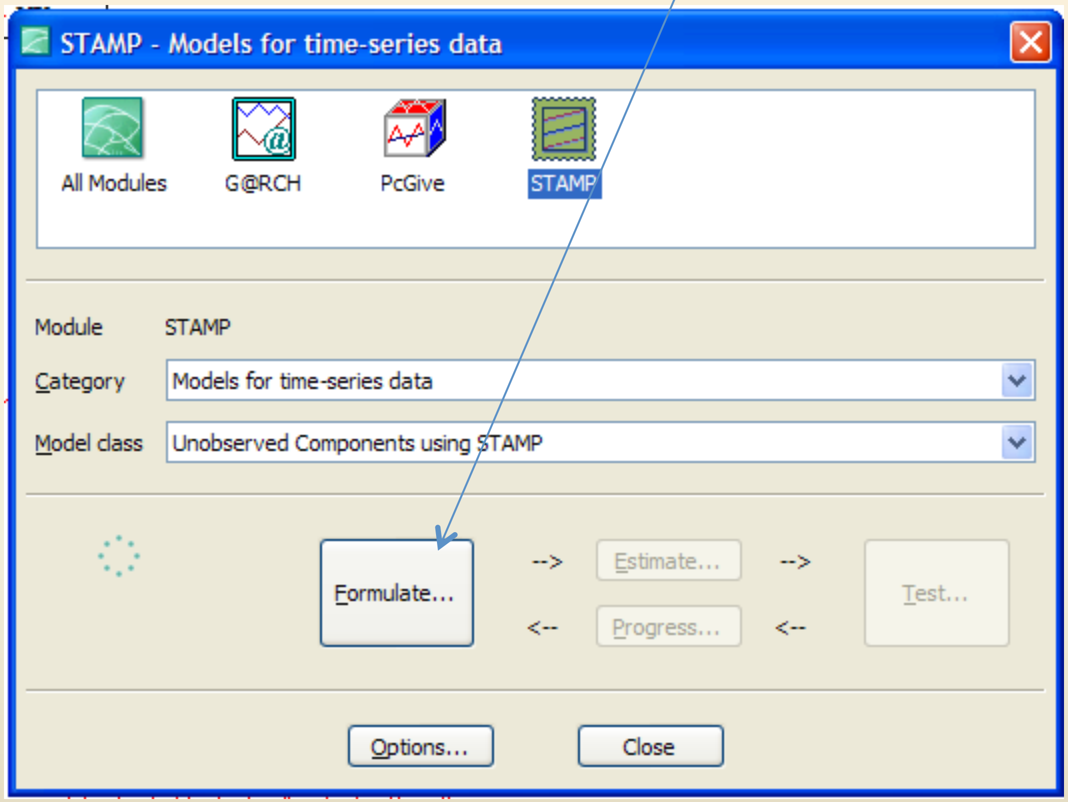
Basic Structural Model

- The basic structural model contains a local level, a local slope, and a local seasonal.
- We begin by allowing all of these components to be stochastic.

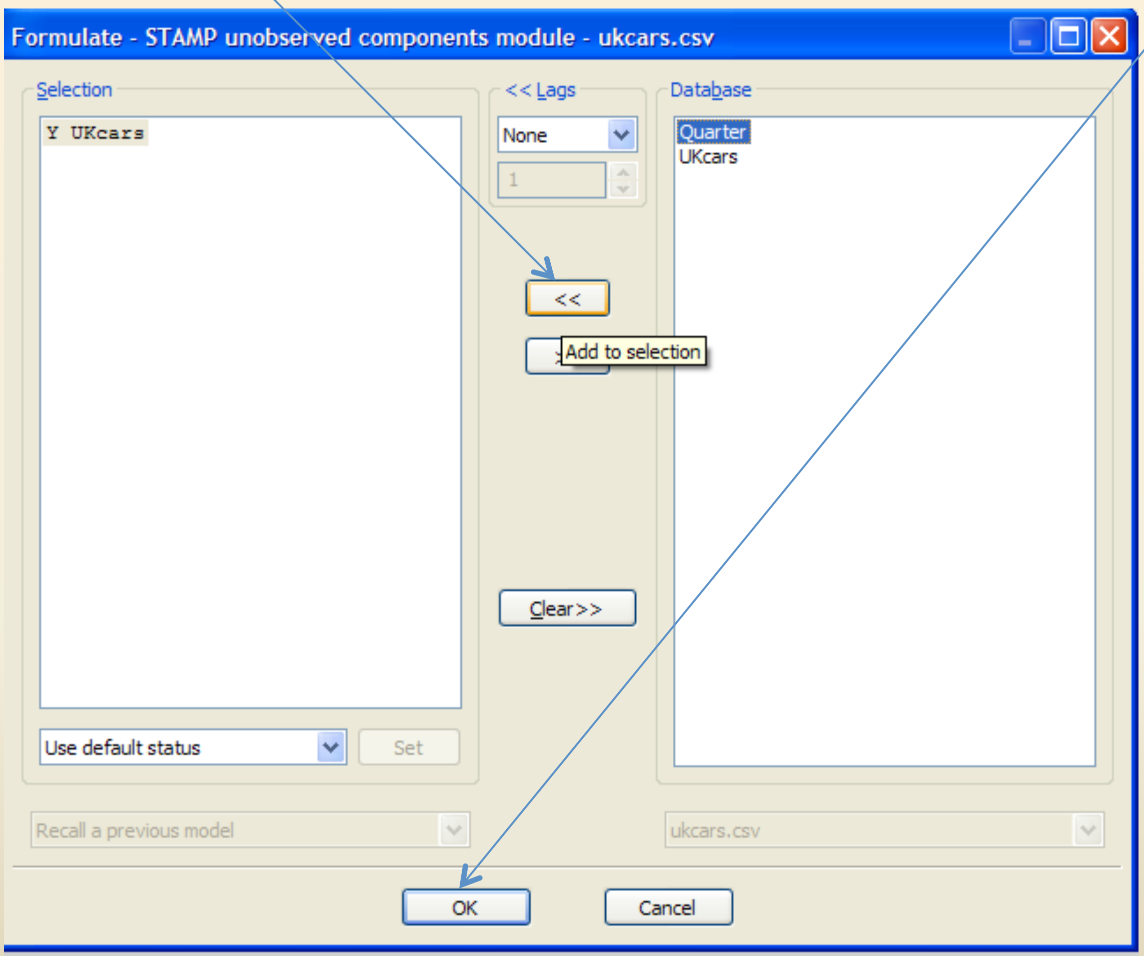
Click on the model icon



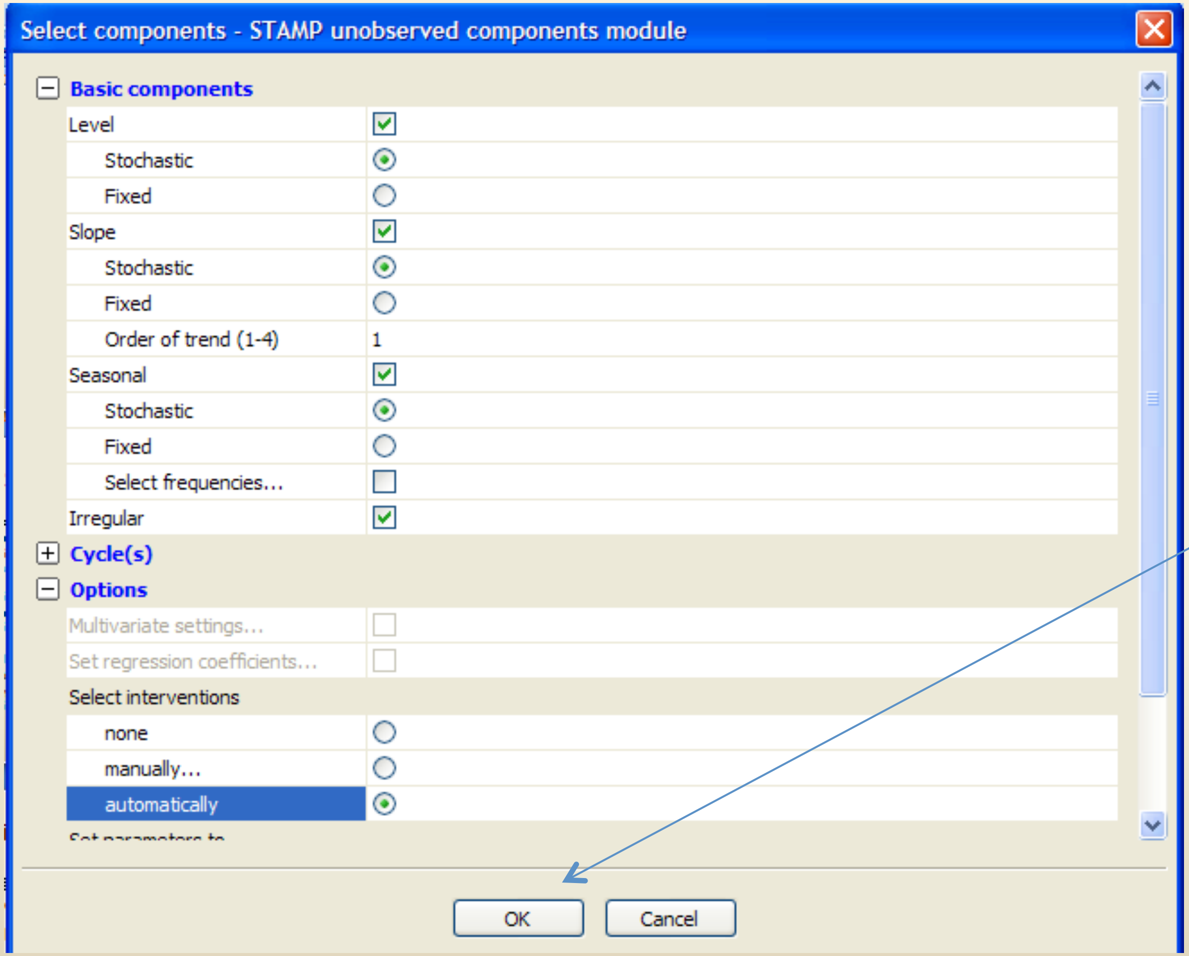
Dialog box comes down click on formulate



Add the Ukcars variable to the selection box by clicking on the move arrow in the middle, then click OK

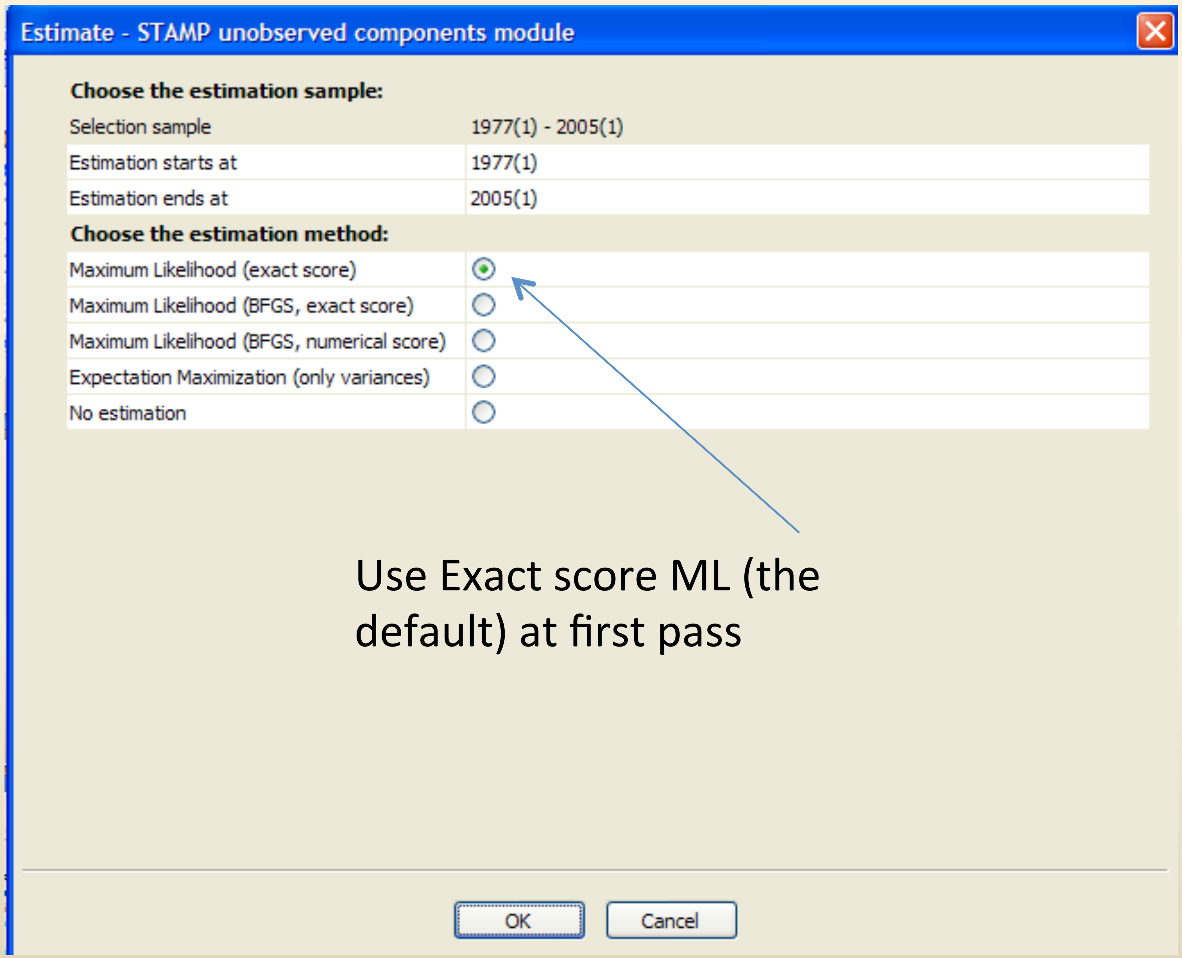


Select the level, slope, seasonal as stochastic (random), and interventions as automatic



Then click OK

An estimation dialog box appears.
We don't shorten the estimation horizon at first—leave this the full sample for the first pass



Use Exact score ML (the default) at first pass

Omnibus Estimation review

Strong convergence and steady state found are good indicators

```
Ox Professional version 6.00 (Windows/U) (C) J.A. Doornik, 1994-2009
STAMP 8.20 (C) S.J. Koopman and A.C. Harvey, 1995-2009
---- STAMP 8.20 session started at 11:50:04 on 19-11-2009 ----

 Estimating...
 Very strong convergence relative to 1e-007
 - likelihood cvg 7.80019e-015
 - gradient cvg 2.46577e-009
 - parameter cvg 5.60377e-008
 - number of bad iterations 0
....
 Very strong convergence relative to 1e-007
 - likelihood cvg 1.91037e-016
 - gradient cvg 1.78719e-010
 - parameter cvg 7.46128e-008
 - number of bad iterations 0
 Estimation process completed.

UC( 1) Estimation done by Maximum Likelihood (exact score)
    The database used is C:\Program Files\OxMetrics6\data\ukcars.csv
    The selection sample is: 1977(1) - 2005(1) (T = 113, N = 1)
    The dependent variable Y is: UKcars
    The model is:  Y = Trend + Seasonal + Irregular + Interventions
    Steady state. found

Log-Likelihood is -1063.73 (-2 LogL = 2127.46).
Prediction error variance is 5.15371e+008
```

Estimation completed

Model description

Predictive error variance

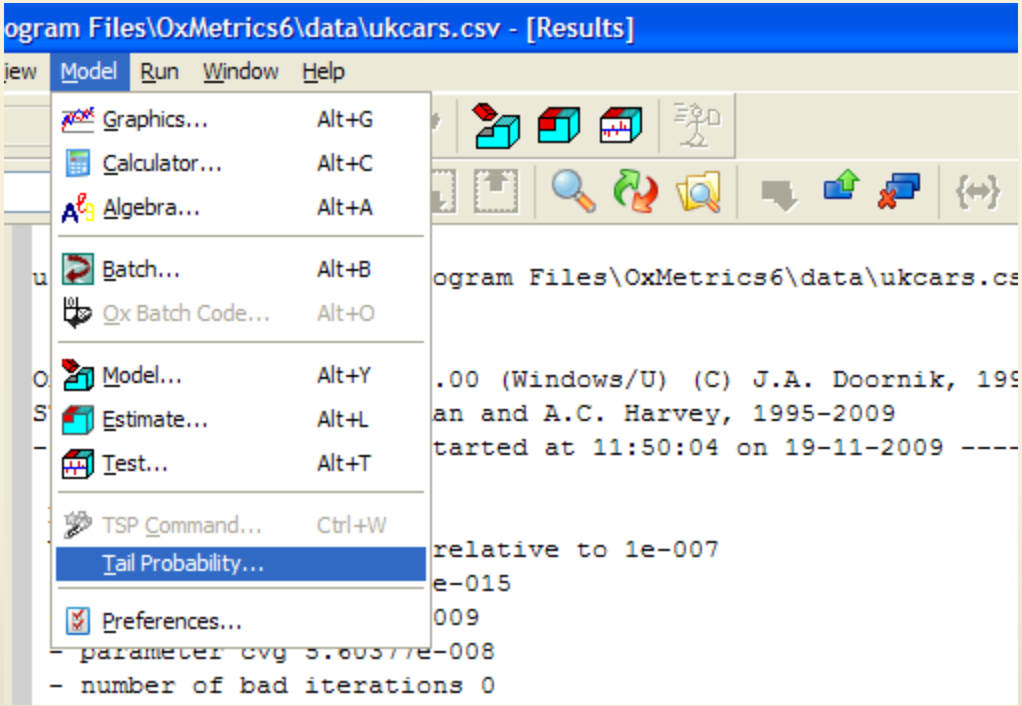
Omnibus statistical review

```
UC( 1) Estimation done by Maximum Likelihood (exact score)
    The database used is C:\Program Files\OxMetrics6\data\ukcars.csv
    The selection sample is: 1977(1) - 2005(1) (T = 113, N = 1)
    The dependent variable Y is: UKcars
    The model is:  Y = Trend + Seasonal + Irregular + Interventions
    Steady state. found

Log-Likelihood is -1063.73 (-2 LogL = 2127.46).
Prediction error variance is 5.15371e+008

Summary statistics
                  UKcars
 T                113.00
 p                3.0000
 std.error        22702.
 Normality       0.11890
 H(35)           0.79677
 DW               2.0367
 r(1)          -0.033528
 q                12.000
 r(q)           -0.10845
 Q(q,q-p)         41.693
 Rs^2            0.33489

 Variances of disturbances:
                   Value     (q-ratio)
Level        2.39535e+008  (     1.000)
Slope             29756.2  (0.0001242)
Seasonal     1.21628e+006  ( 0.005078)
Irregular    1.07839e+008  (   0.4502)
```

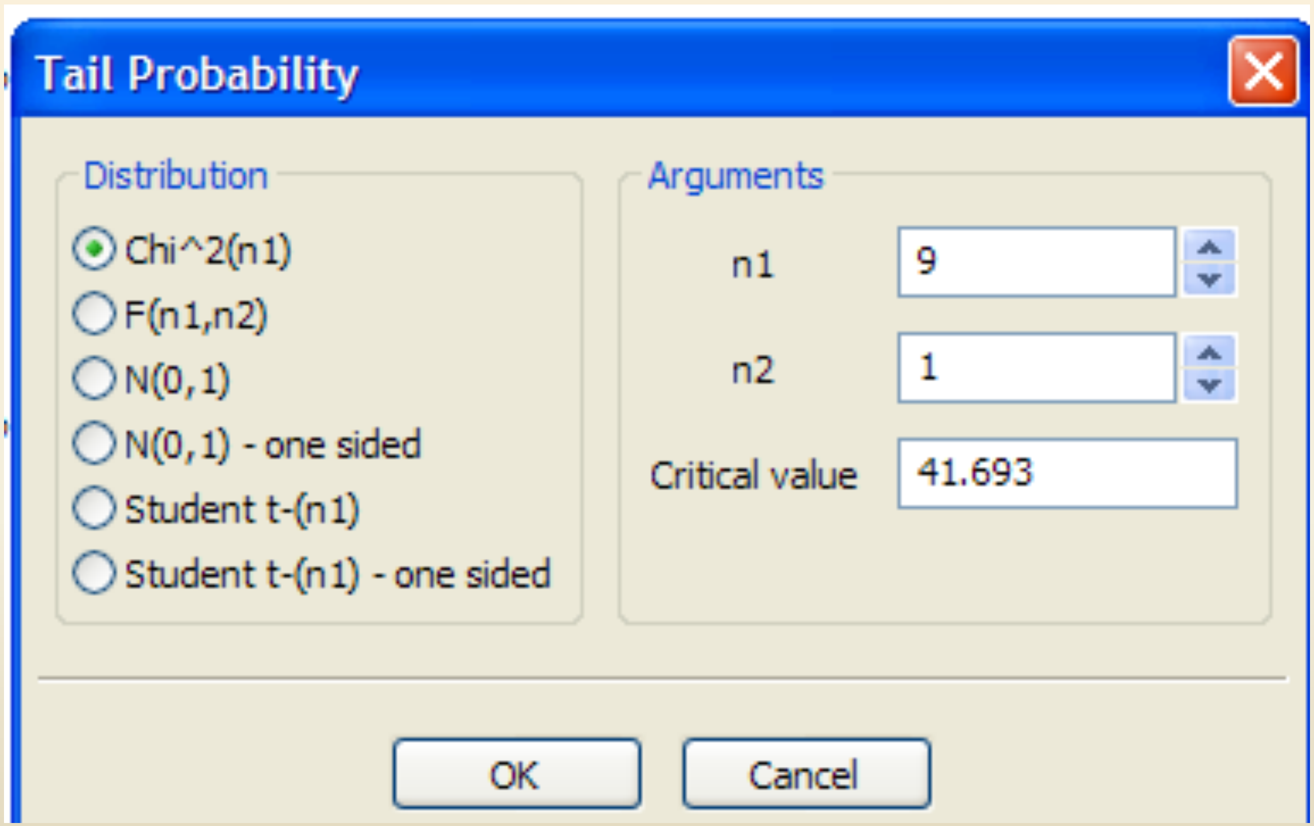
Box-Ljung Q stat (12,9)

All components are stochastic (each > 0)

To test the chi-square at 9 df, click on model in the menu bar and then on Tail Probability



Enter 9df in n1 and insert the critical value given in the output



At the bottom of your output, the significance test will be recorded

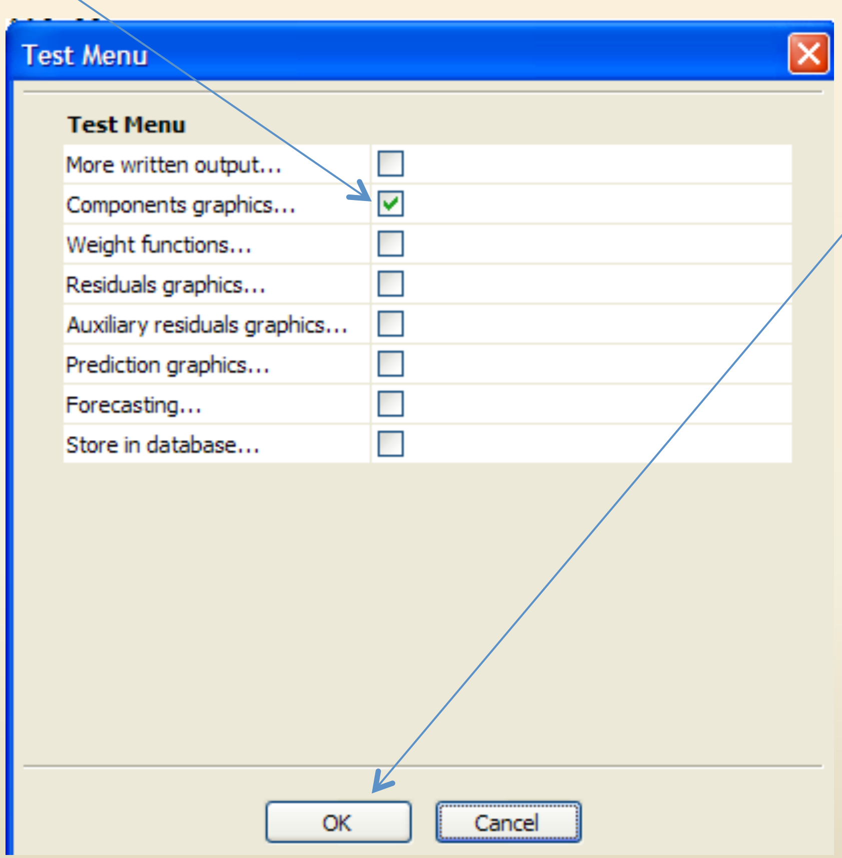
```
Chi^2(9) = 41.693 [0.0000] **
```

A significant result indicates that there remains serial correlation in the residuals.

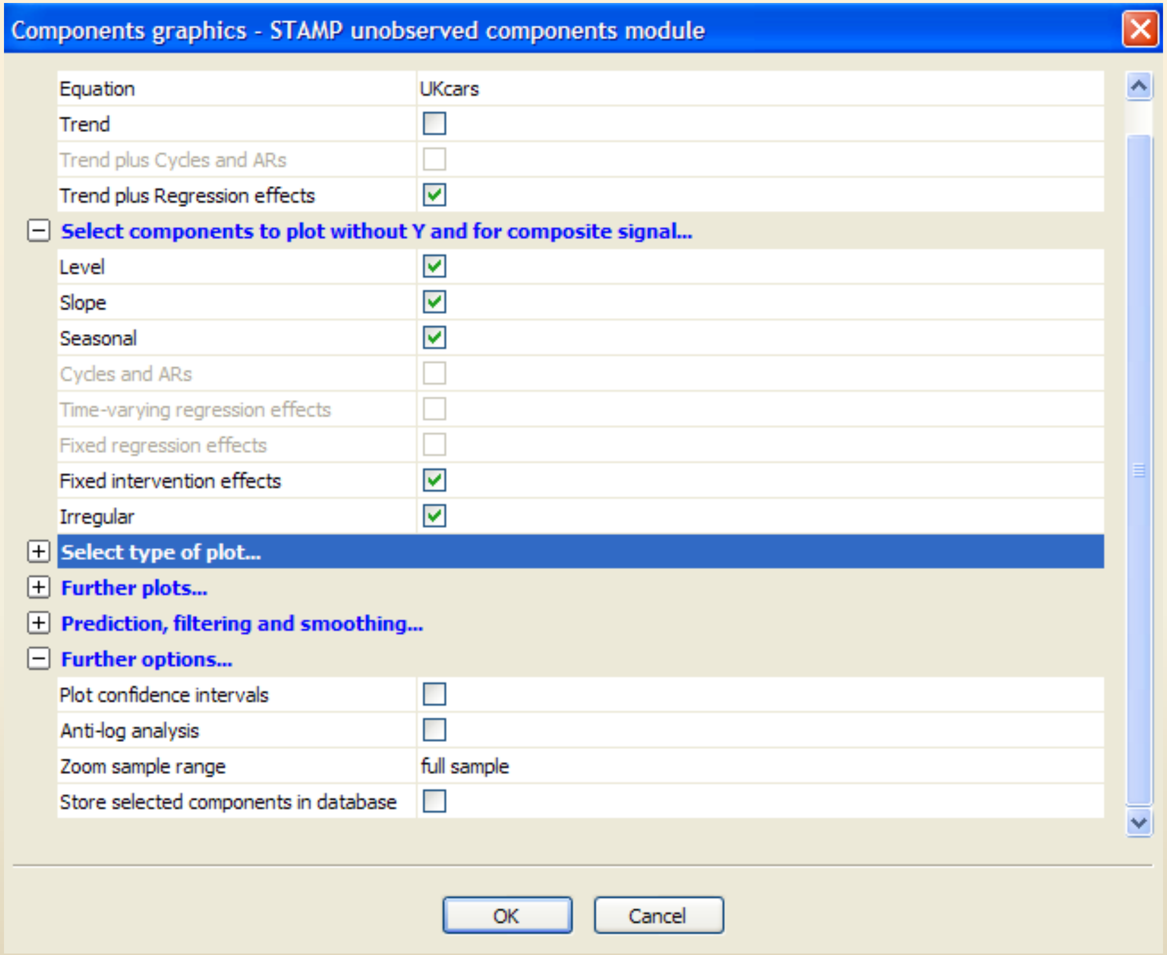
This result will bias your t-tests , F-tests, and R^2 upward.

To be able to trust those tests, you will need to neutralize the serial correlation in the residuals

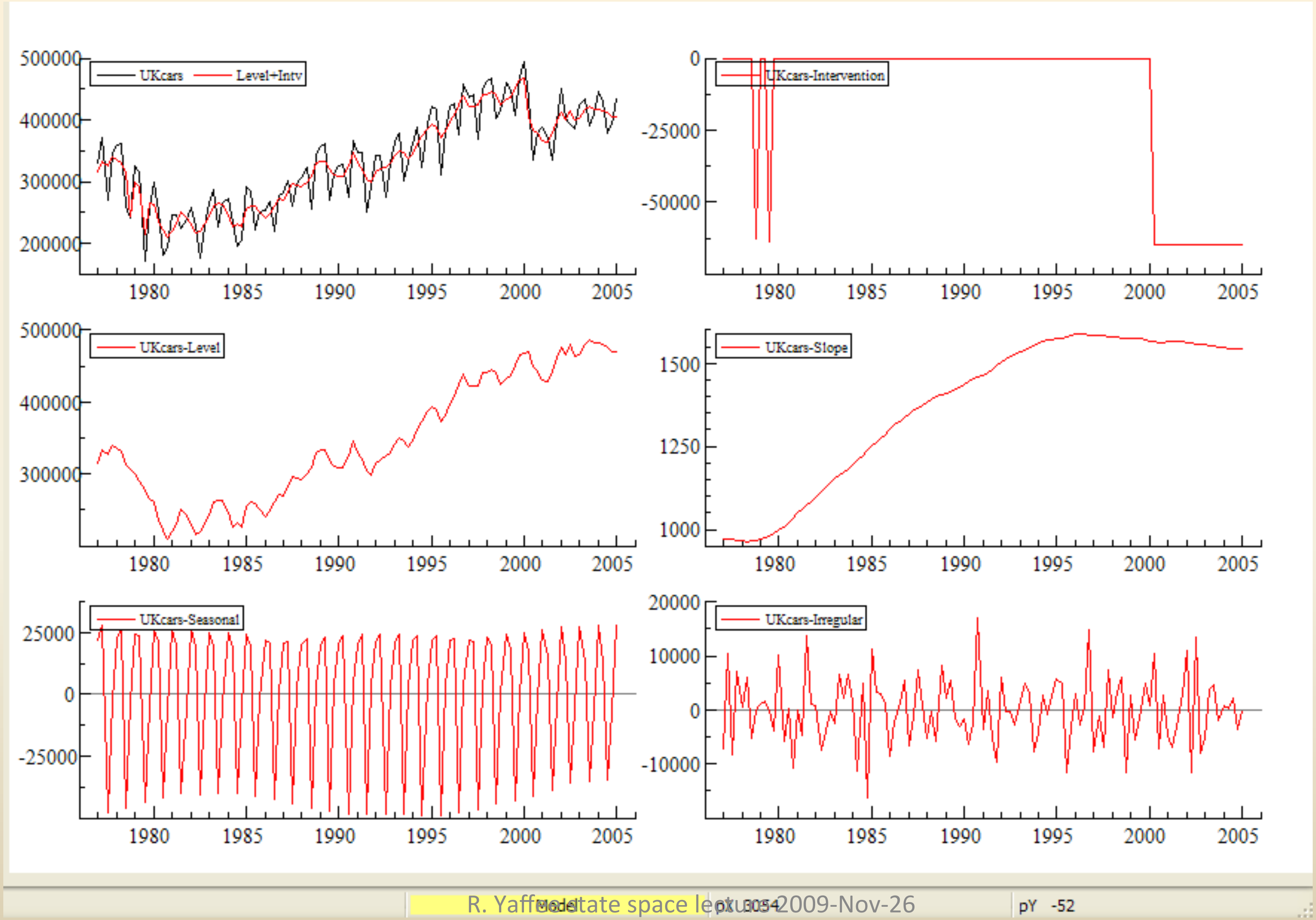
Click on the test cube on the right, and select component graphics from the test menu, and click ok



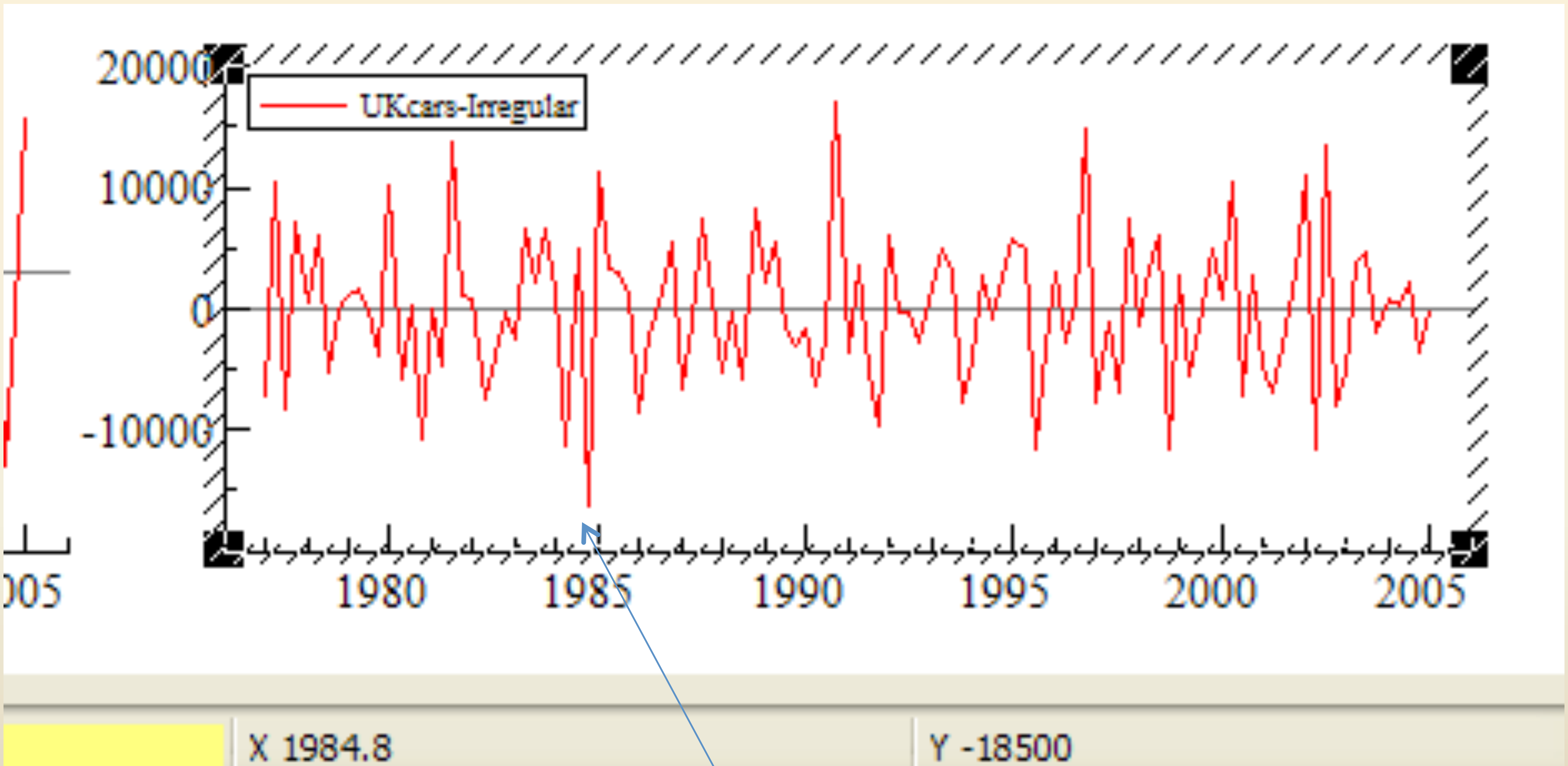
A first pass, just consider the main components



Component graphics are generated from the smoothed components of the state vector (signal extraction)



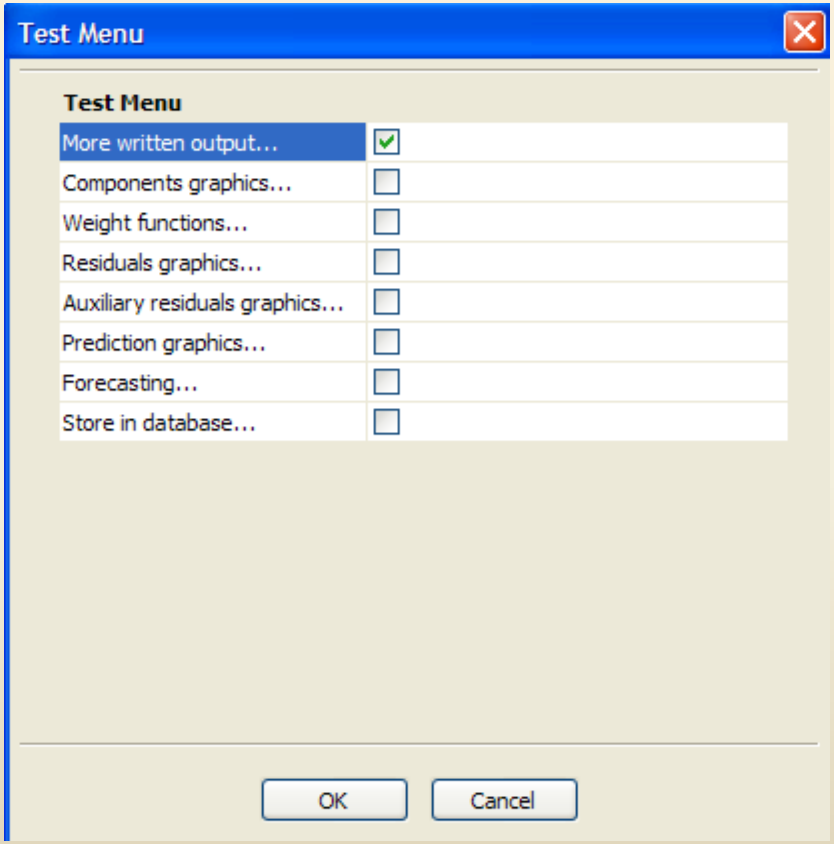
Diagnosing outliers and level shifts



Place cursor on outlier and the proper date will appear to the left

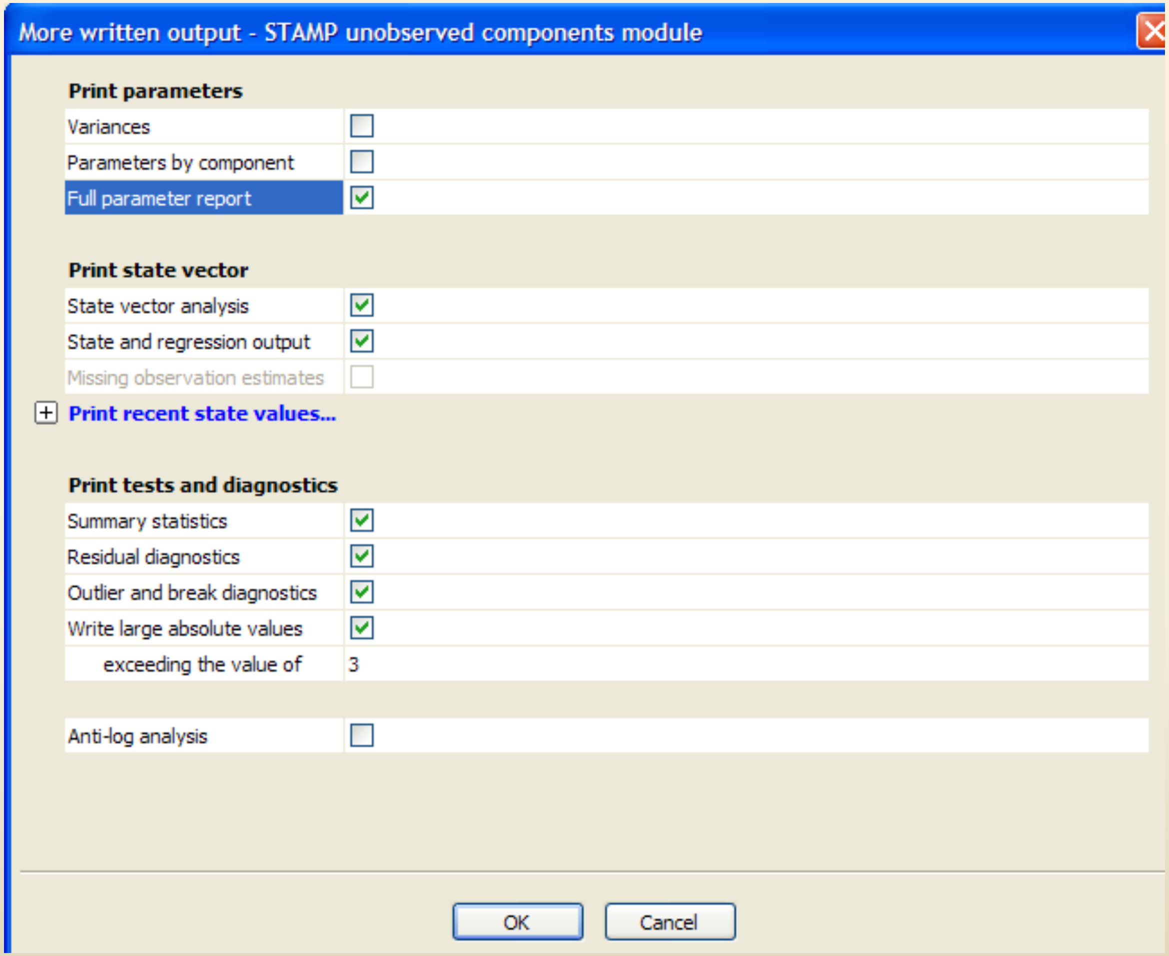
A more detailed examination requires more output

(click on the test icon and then in the test menu, on “More written output”. Then click ok.



The More Written Output menu

Select these options and click ok.



Full parameter report shows no problem computing first and 2nd derivatives (or asymptotic standard errors)

Full parameter report

Actual parameters (all)

	Value
Var Level	2.3954e+008
Var Slope	29756.
Var Seasonal	1.2163e+006
Var Irregular	1.0784e+008

Transformed parameters (not fixed)

	Transform	1stDer	2ndDer	asymp.s.e
Var Level	9.6471	0.033280	-0.69824	0.32377
Var Slope	5.1504	-2.1566e-005	-0.00038409	6.1493
Var Seasonal	7.0057	0.0049193	-0.044708	0.87008
Var Irregular	9.2481	0.020847	-0.27549	0.63997

Actual parameters (not fixed) with 68% asymmetric confidence interval

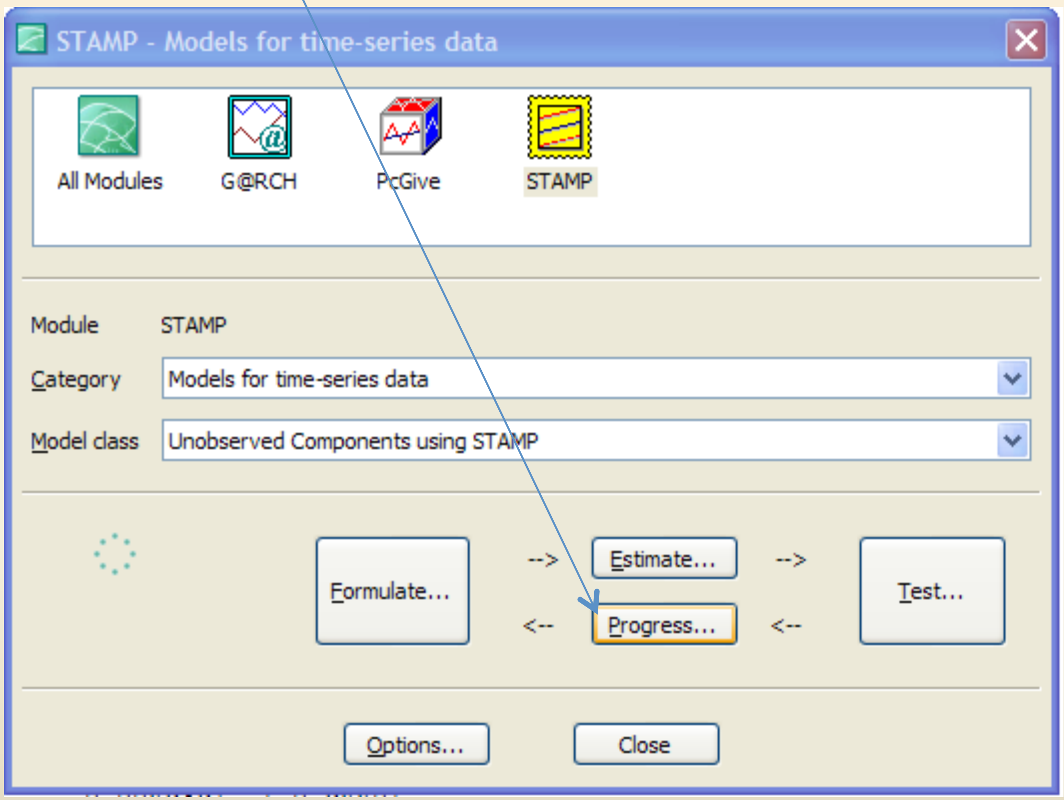
	Value	leftbound	rightbound
Var Level	2.3954e+008	1.2536e+008	4.5771e+008
Var Slope	29756.	0.13563	6.5281e+009
Var Seasonal	1.2163e+006	2.1345e+005	6.9307e+006
Var Irregular	1.0784e+008	2.9985e+007	3.8784e+008

All components are significant except the slope

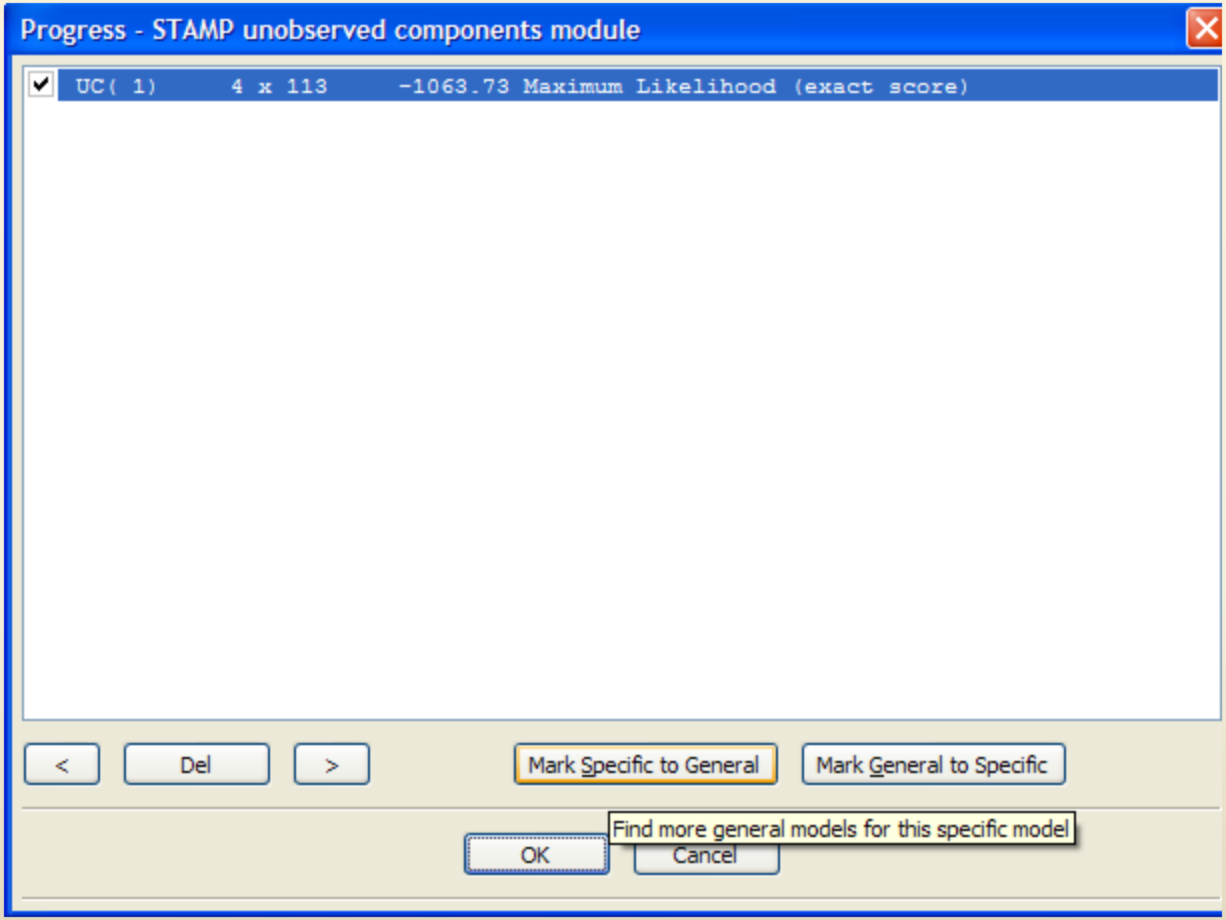
we could trim the model and gain power with more df for a test by pruning out the slope

```
State vector analysis at period 2005(1)
                              Value       Prob
Level                   469891.41528 [0.00000]
Slope                     1542.90170 [0.39354]
Seasonal chi2 test          45.19769 [0.00000]
Seasonal effects:
                Period        Value       Prob
                     1 27966.58096 [0.00002]
                     2 13519.89654 [0.04214]
                     3-34818.84346 [0.00000]
                     4 -6667.63404 [0.28321]

State vector at period 2005(1)

              Coefficient         RMSE     t-value       Prob
Level        469891.41528 23742.14541    19.79145 [0.00000]
Slope          1542.90170  1800.91354     0.85673 [0.39354]
Seasonal      31392.71221  4930.13796     6.36751 [0.00000]
Seasonal 2    10093.76529  5062.46656     1.99384 [0.04876]
Seasonal 3    -3426.13125  4076.24819    -0.84051 [0.40253]

Regression effects in final state at time 2005(1)

                    Coefficient         RMSE     t-value       Prob
Outlier 1978(4)    -62534.93392 18121.05052    -3.45096 [0.00081]
Outlier 1979(3)    -63562.56645 17875.22817    -3.55590 [0.00057]
Level break 2000(2)-64718.35882 21068.46118    -3.07181 [0.00271]
```

To test the removal of the slope against a significant change in the LL, click on formulate icon, (the left hand cube) and then on progress button in the dialog box



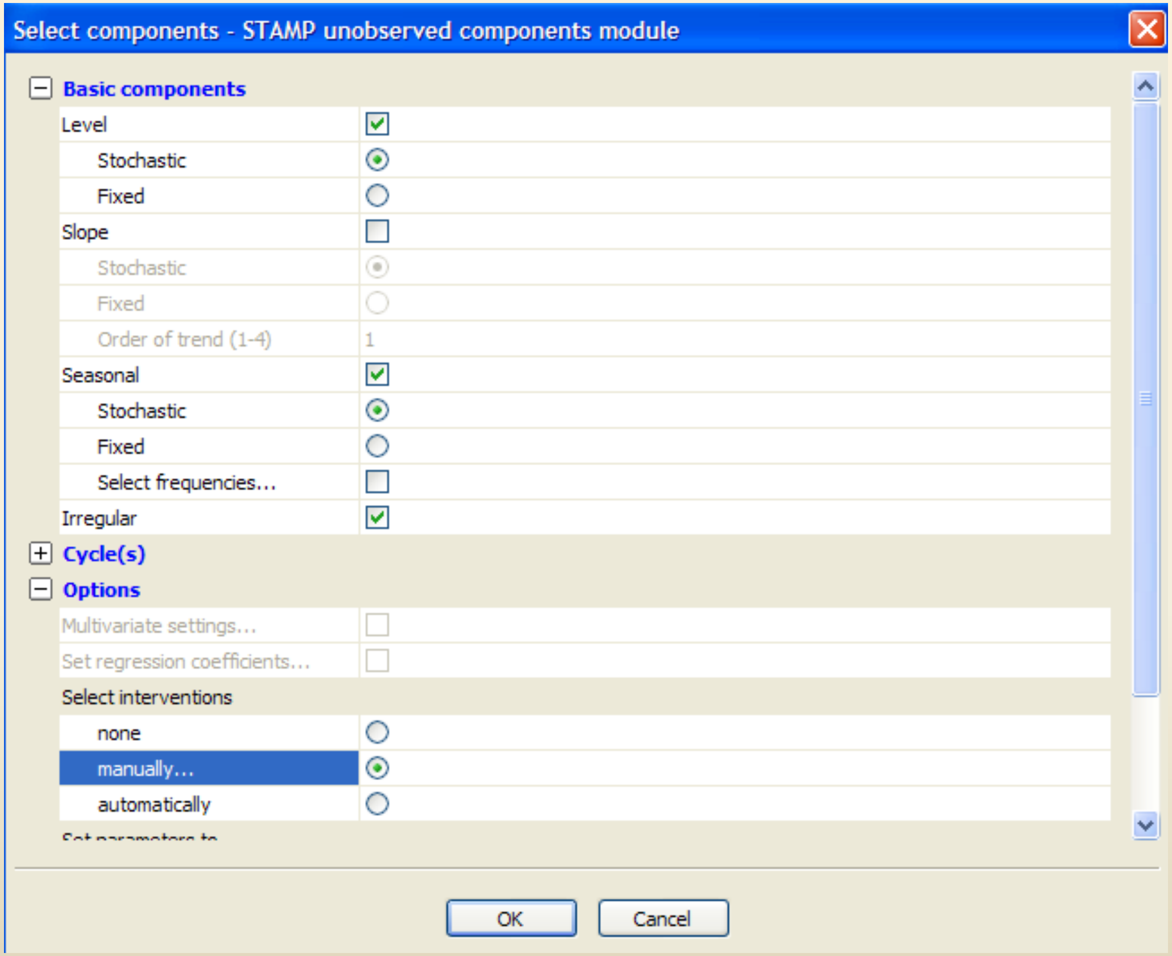
Click on "mark general to specific" and then ok.



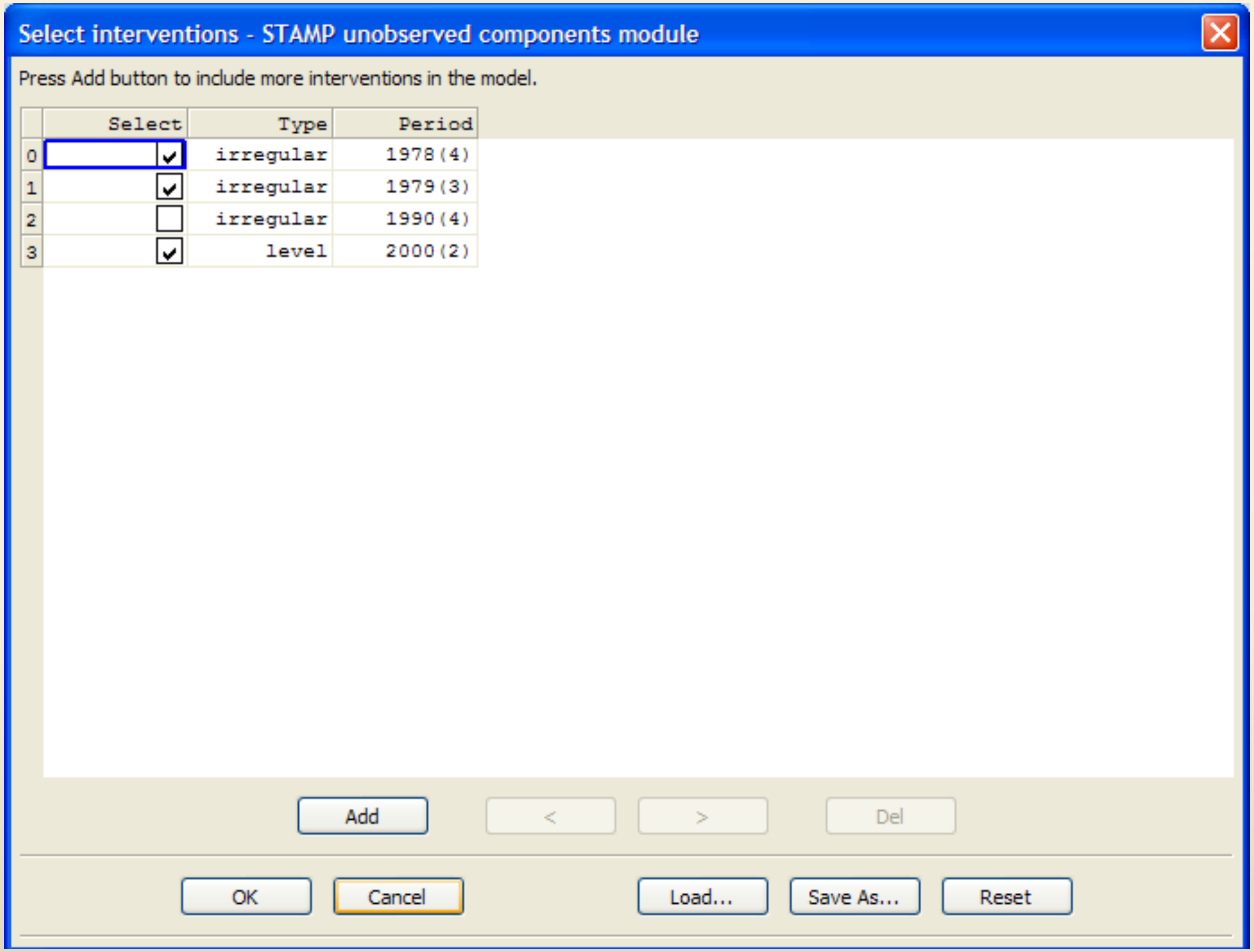
This leaves a record in memory as to the goodness of fit of that model as indicated by LL or IC

```
Progress to date
Model             T     p          log-likelihood          SC           HQ          AIC
UC( 1)          113     4  Maximum Likelihood (exact score)     -1063.7319       18.994<      18.937<      18.89
```

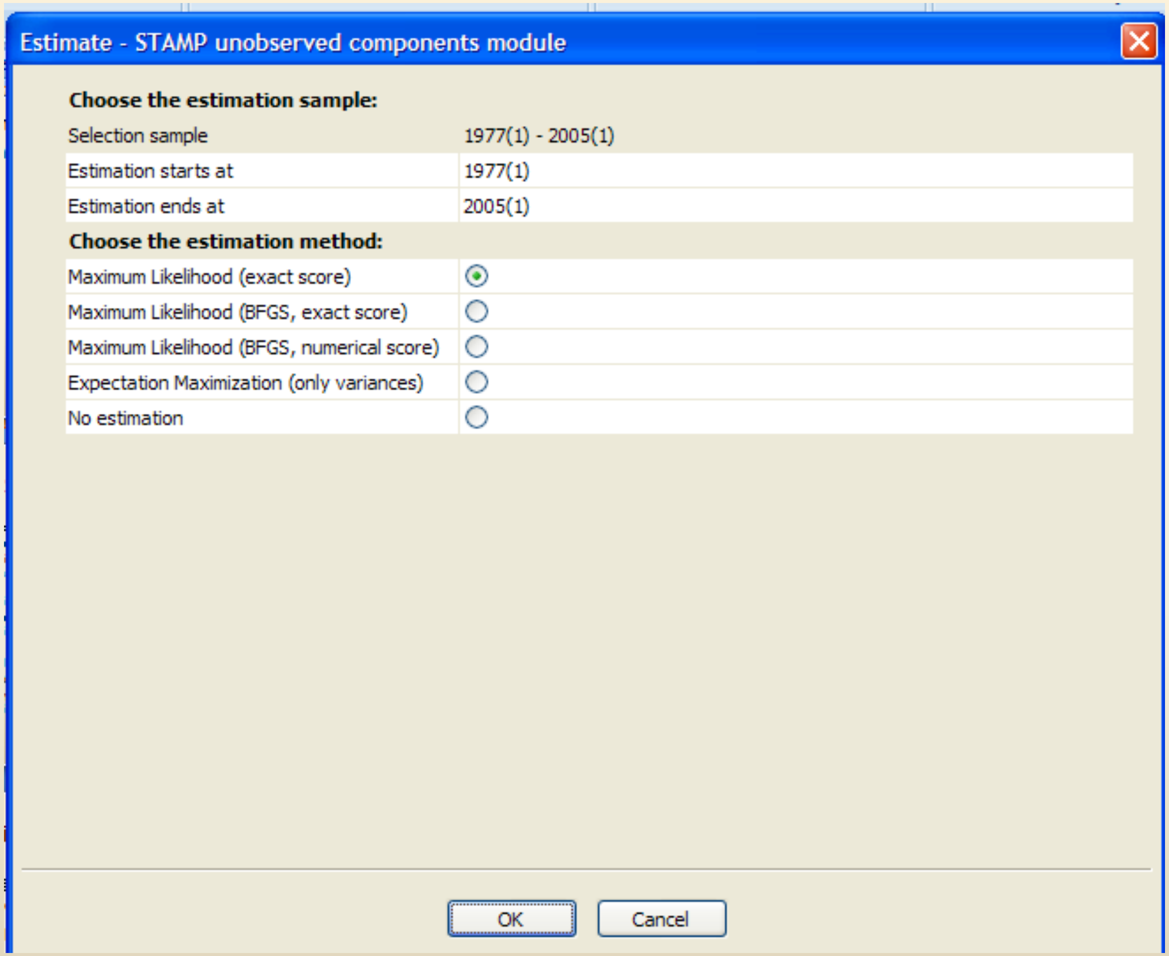
When the select components menu appears, deselect the slope and leave intervention selection on manual then click ok.



When the select interventions menu appears, we do not change it, and click ok.   We are only testing the significance of the slope.



The estimation menu, we also leave the same, and proceed to click ok.



The New Model appears with all components significant.

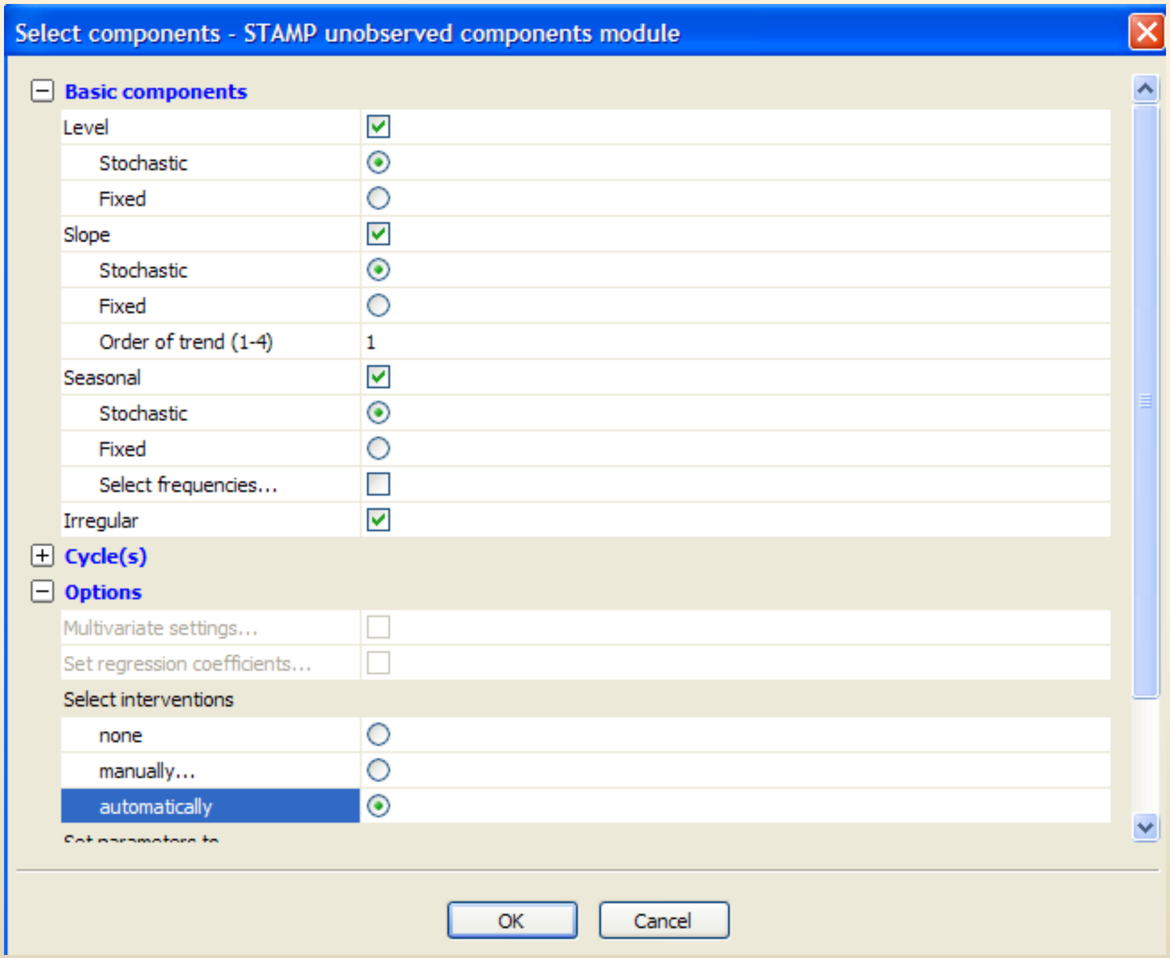
- We go back to the formulate icon (the left hand cube) and click on it.
- We click on the progress button on the drop down formulate dialog box.
- We then click on General to specific.
- What appears at the bottom of our output is:

```
Progress to date
Model          T    p         log-likelihood          SC          HQ         AIC
UC( 1)       113    4  Maximum Likelihood (exact score)    -1063.7319    18.994<    18.937<    18.89
UC( 2)       113    3  Maximum Likelihood (exact score)    -1070.9629    19.081     19.038     19.00

Tests of model reduction (please ensure models are nested for test validity)
UC( 1) --> UC( 2): Chi^2( 1) =     14.462 [0.0001] **
```

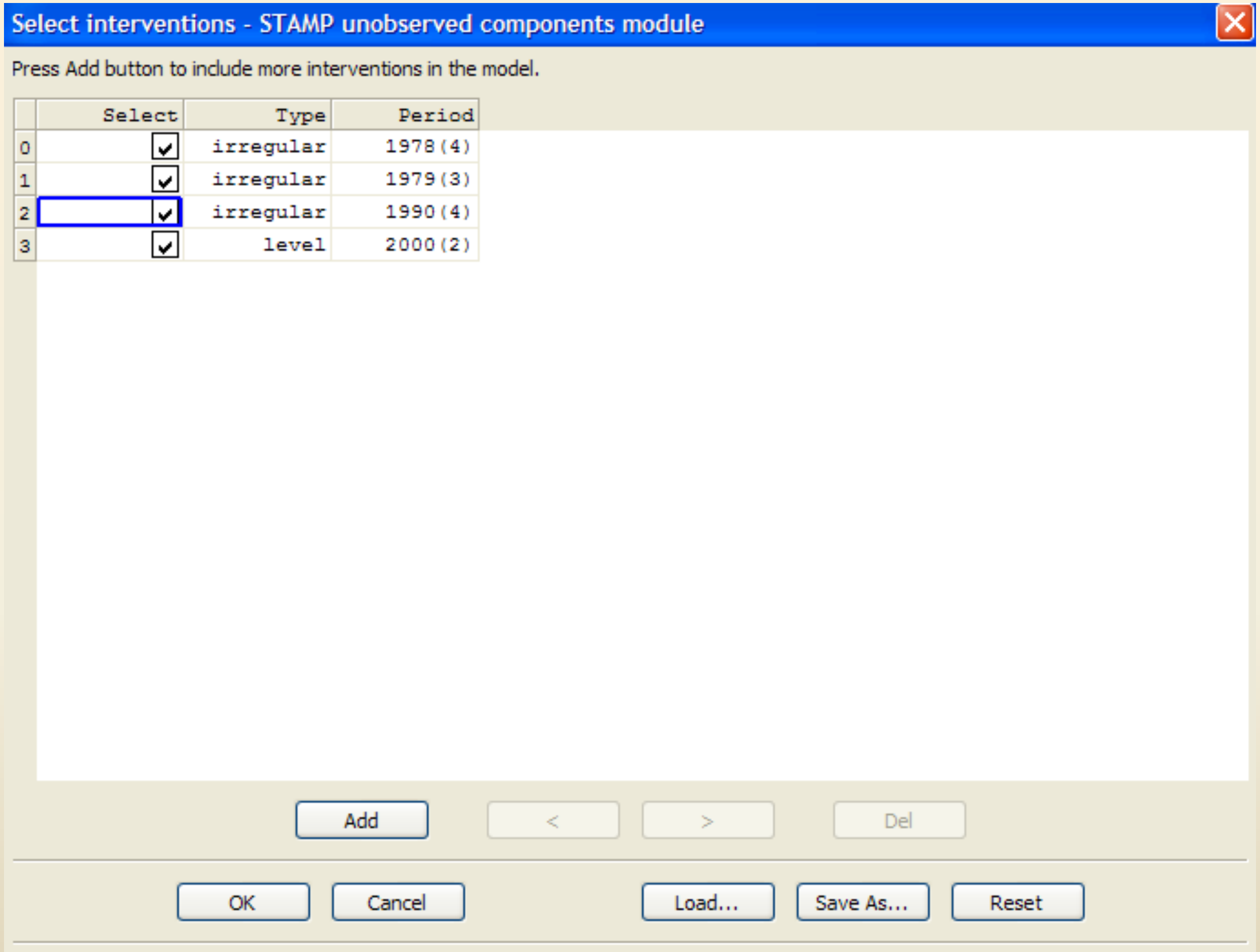
Removal of the slope component significantly reduced the log-likelihood

- Therefore, we will restore the stochastic slope component to our state vector, even though it did not appear to be important by comparison to the other components.
- The signal to noise ratio was not very high for that component. However, there was a perceptible bend in the curve that matched the trend.

2nd pass—select all components as stochastic and reiterate



If this menu appears regardless of your having opted for automatic selection, select all suggested and then ok.



State Space Model Diagnosis

- Omnibus model diagnostics
- Component diagnostics
- Residual analysis
  - Auxiliary residuals
  - Residuals
  - Graphical diagnostics
- Intervention diagnostics
- Explanatory variable diagnostics
- Forecasting
- Forecasting evaluation
- Model fitting strategies
- Model adequacy
- Model optimality and the progress option

This model did not fully converge to a steady state.

```
 Estimating....
 Weak convergence relative to 1e-007
 - likelihood cvg 1.91461e-010
 - gradient cvg 4.44545e-006
 - parameter cvg 2.85583e-005
 - number of bad iterations 0
...
 Very strong convergence relative to 1e-007
 - likelihood cvg 4.30692e-015
 - gradient cvg 6.41512e-009
 - parameter cvg 7.26624e-008
 - number of bad iterations 0
 Estimation process completed.

UC( 3) Estimation done by Maximum Likelihood (exact score)
    The database used is C:\Program Files\OxMetrics6\data\ukcars.csv
    The selection sample is: 1977(1) - 2005(1) (T = 113, N = 1)
    The dependent variable Y is: UKcars
    The model is:  Y = Trend + Seasonal + Irregular + Interventions
    Steady state.......... found without full convergence

Log-Likelihood is -1063.91 (-2 LogL = 2127.81).
Prediction error variance is 5.14948e+008
```

What are our options?

- We may fine-tune the model, by
- Trying different starting values for parameters
- We may fit other interventions to improve convergence to a steady-state.
- We begin by looking at the component graphics and then asking for more written output.

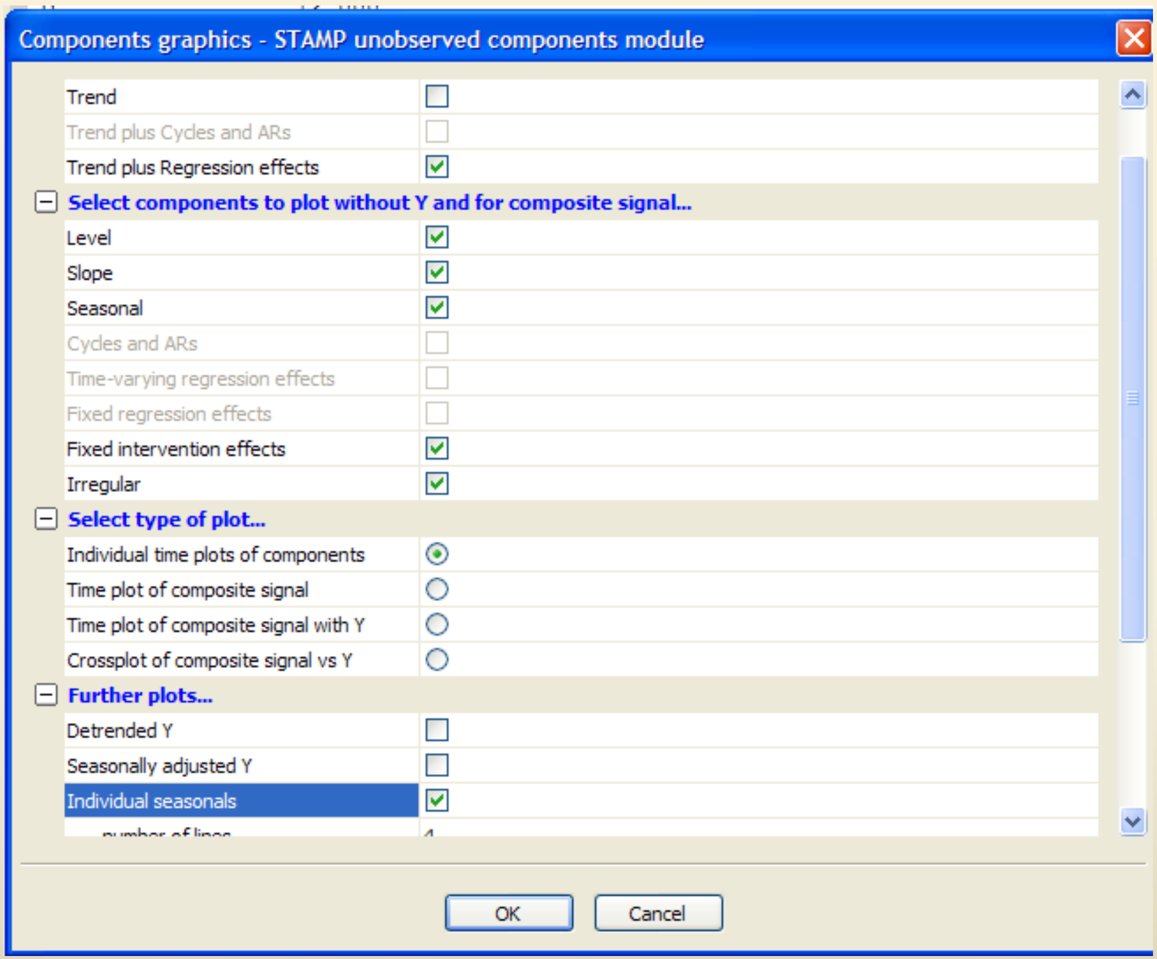
Omnibus Diagnosis

- We test our Box-Ljung Q for residual serial autocorrelation with 9 df.   It is still significant so we have autocorrelation in the residuals.
- We will have to deal with that to avoid biasing our significance tests .
- We opt for component graphics first, and also select individual seasonals from further plots.

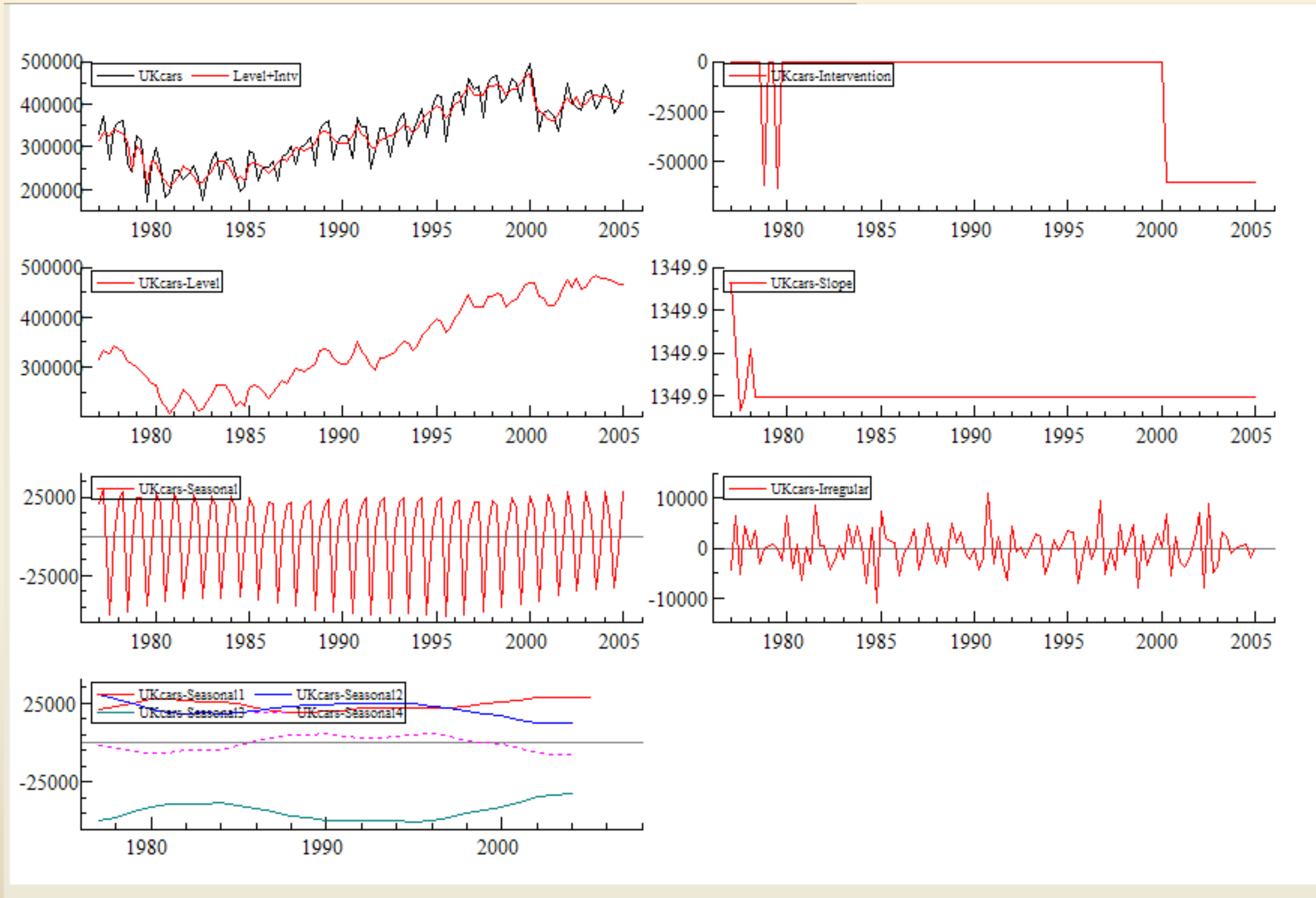
Omnibus model diagnostics

- These diagnostics assess overall model goodness of fit
- They may provide hints wrt problems
- They are helpful in initial comparison of models

Selecting component graphics for the 2nd pass



The slope is very different this time. We need to ascertain why.



The Full parameter report reveals that inversion of the vcv matrix failed and a generalized inverse was used to proceed.

```
Outlier 1979(3)    -63242.04916 16246.70236    -3.89261 [0.00017]
Level break 2000(2)-60747.83025 20022.10404    -3.03404 [0.00304]
 Chi^2(9) = 37.431 [0.0000] **

 Full parameter report
 Actual parameters (all)
                      Value
Var Level       2.5464e+008
Var Slope           0.00000
Var Seasonal    1.6112e+006
Var Irregular   5.8689e+007
Warning: invertgen: invertsym failed, proceeding with generalized p.s.d. inverse
SEst.ox (2593): PrintPar
 Transformed parameters (not fixed)
                  Transform        1stDer        2ndDer     asymp.s.e
Var Level            9.6777       0.11332      -0.97661       0.16515
Var Seasonal         7.1462      0.018175     -0.067805       0.36380
Var Irregular        8.9439      0.053840     -0.097339       0.38523
 Actual parameters (not fixed) with 68% asymmetric confidence interval
                      Value     leftbound    rightbound
Var Level       2.5464e+008   1.8301e+008   3.5430e+008
Var Seasonal    1.6112e+006   7.7832e+005   3.3353e+006
Var Irregular   5.8689e+007   2.7161e+007   1.2681e+008

State vector at period 2005(1)

               Coefficient         RMSE     t-value       Prob
Level         465183.25500 22203.14185    20.95124 [0.00000]
Slope           1349.90063  1522.88053     0.88641 [0.37742]
Seasonal       30761.14809  5033.38760     6.11142 [0.00000]
Seasonal 2     10197.28991  5239.73598     1.94615 [0.05431]
Seasonal 3     -2221.12361  4151.19778    -0.53506 [0.59374]
```

Lack of variance in the slope may have caused it to be modeled as fixed

Variances of disturbances:

	Value	(q-ratio)
Level	2.54638e+008	(1.000)
Slope	0.000000	(0.0000)
Seasonal	1.61120e+006	(0.006327)
Irregular	5.86886e+007	(0.2305)

Omnibus statistics

```
Goodness-of-fit based on Residuals UKcars
                                                 Value
Prediction error variance (p.e.v)          5.1495e+008
Prediction error mean deviation (m.d)      4.6036e+008
Ratio p.e.v. / m.d in squares                  0.79655
Coefficient of determination R^2               0.92178
... based on differences Rd^2                  0.80538
... based on diff around seas mean Rs^2        0.33544
Information criterion Akaike (AIC)              20.219
... Bayesian Schwartz (BIC)                     20.436

Serial correlation statistics for Residuals UKcars
 Durbin-Watson test is 2.12445
 Asymptotic deviation for correlation is 0.09759
   Lag        df      Ser.Corr    BoxLjung        prob
     4         1      -0.18074      14.966   [ 0.0001]
     8         5        0.1077      27.708   [ 0.0000]
    12         9      -0.10418      37.431   [ 0.0000]
```

R^2 is probably inflated owing to the residual serial autocorrelation

- The R square = .92 but this is questionable
- Serial correlation in the residuals could inflate this, F and t values.
- Yet all residuals appear to be normal:
  - Irregular
  - Level
  - Slope

Residuals appear normally distributed for all components

```
Normality test for Irregular residual
                          Value
 Sample size            113.00
Mean                -0.0030608
St.Dev                  1.1131
Skewness               0.17326
Excess kurtosis       0.017761
Minimum                -2.9569
Maximum                 2.9587

                        Chi^2        prob
Skewness              0.56536   [ 0.4521]
Kurtosis            0.0014852   [ 0.9693]
Bowman-Shenton        0.56685   [ 0.7532]

Normality test for Level residual
                          Value
 Sample size            111.00
Mean                 -0.010333
St.Dev                  1.0891
Skewness             -0.027343
Excess kurtosis       -0.64500
Minimum                -2.4666
Maximum                 2.8404

                        Chi^2        prob
Skewness             0.013831   [ 0.9064]
Kurtosis               1.9241   [ 0.1654]
Bowman-Shenton          1.938   [ 0.3795]

Normality test for Slope residual
                          Value
 Sample size            111.00
Mean                   0.58561
```

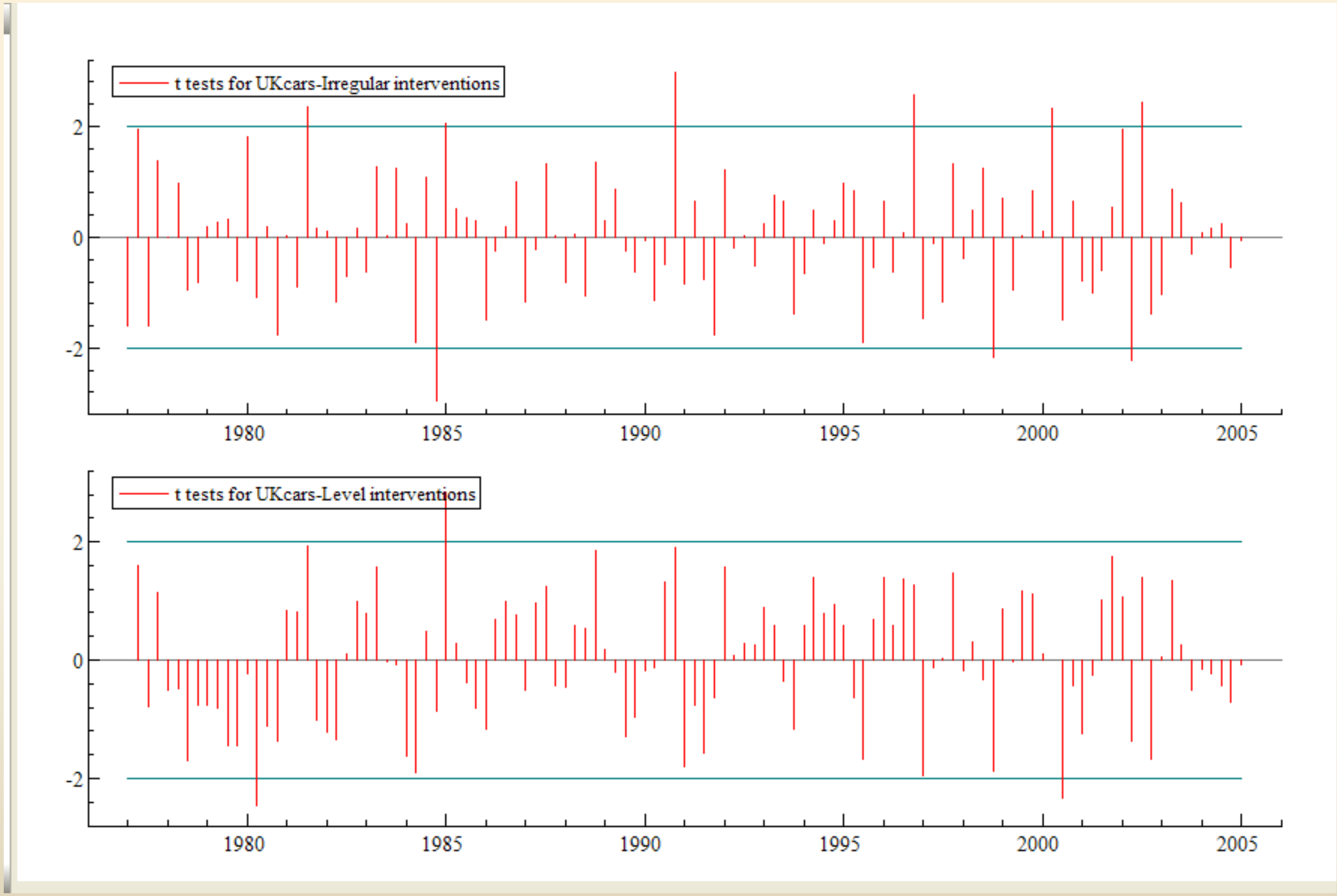
We need to examine the graphics of the irregular to look for other interventions

Normality test for Slope residual

	Value
Sample size	111.00
Mean	0.58561
St.Dev	0.78850
Skewness	-0.088005
Excess kurtosis	-0.66432
Minimum	-1.5964
Maximum	2.2688

	Chi^2	prob
Skewness	0.14328	[0.7050]
Kurtosis	2.0411	[0.1531]
Bowman-Shenton	2.1844	[0.3355]

2 candidate outliers in the irregular and one in the level



Candidate outliers

- For the irregular, Aug 1984 and Sept 1990.
- For the level, Jan 1985.
- If these do not eliminate residual serial autocorrelation, when implemented, then we introduce an ar1 lag into the model.
- I found that the slope had been changed by the system to fixed so I reset that to stochastic, selected the those 2 outliers and 1 level shift, and then re-estimated.

Pass 4 (changed the slope to random)

```
Estimating....
Weak convergence relative to 1e-007
- likelihood cvg 4.16272e-012
- gradient cvg 3.6852e-007
- parameter cvg 3.20674e-006
- number of bad iterations 0
Estimation process completed.

UC( 4) Estimation done by Maximum Likelihood (exact score)
   The database used is C:\Program Files\OxMetrics6\data\ukcars.csv
   The selection sample is: 1977(1) - 2005(1) (T = 113, N = 1)
   The dependent variable Y is: UKcars
   The model is:  Y = Trend + Seasonal + Irregular + Interventions
   Steady state. found

Log-Likelihood is -1027.58 (-2 LogL = 2055.15).
Prediction error variance is 4.4347e+008

Summary statistics
                   UKcars
 T                 113.00
 p                 3.0000
 std.error         21059.
 Normality        0.64445
 H(34)             1.0693
 DW                2.0533
 r(1)           -0.049024
 q                 12.000
 r(q)            -0.12002
 Q(q,q-p)          35.466
 Rs^2             0.44403

 Variances of disturbances:
```

Omnibus review

- Steady state was found, model converged.

Variances of disturbances:

	Value	(q-ratio)
Level	2.82462e+008	(1.000)
Slope	66766.8	(0.0002364)
Seasonal	2.30098e+006	(0.008146)
Irregular	4.10468e+007	(0.1453)

State vector analysis at period 2005(1)

	Value	Prob
Level	401104.17363	[0.00000]
Slope	1174.68520	[0.59189]
Seasonal chi2 test	33.40502	[0.00000]

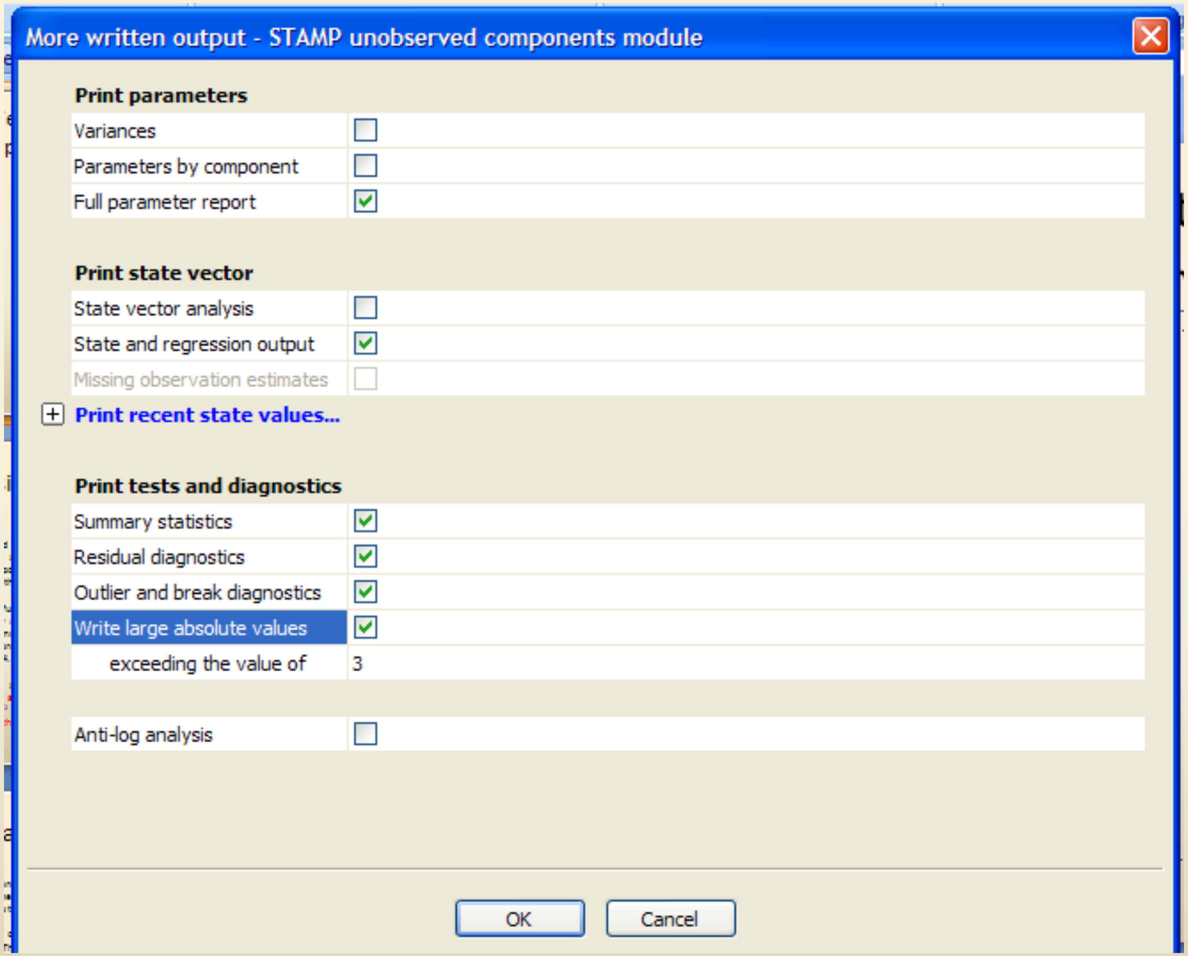
Seasonal effects:

Period	Value	Prob
1	28618.62288	[0.00008]
2	12072.42238	[0.10717]
3	-31771.90420	[0.00002]
4	-8919.14106	[0.19131]

Regression effects in final state at time 2005(1)

	Coefficient	RMSE	t-value	Prob
Outlier 1978(4)	-62928.26379	16569.84786	-3.79776	[0.00025]
Outlier 1979(3)	-62495.35296	16240.32708	-3.84816	[0.00021]
Outlier 1990(4)	45799.27066	15877.98409	2.88445	[0.00478]
Level break 2000(2)	-58782.04032	20441.50495	-2.87562	[0.00491]
Outlier 1984(3)	23482.87575	16007.04136	1.46703	[0.14544]
Level break 1985(1)	[illegible]	[illegible]	[illegible]	[0.00320]

We ask for more written output from the test menu



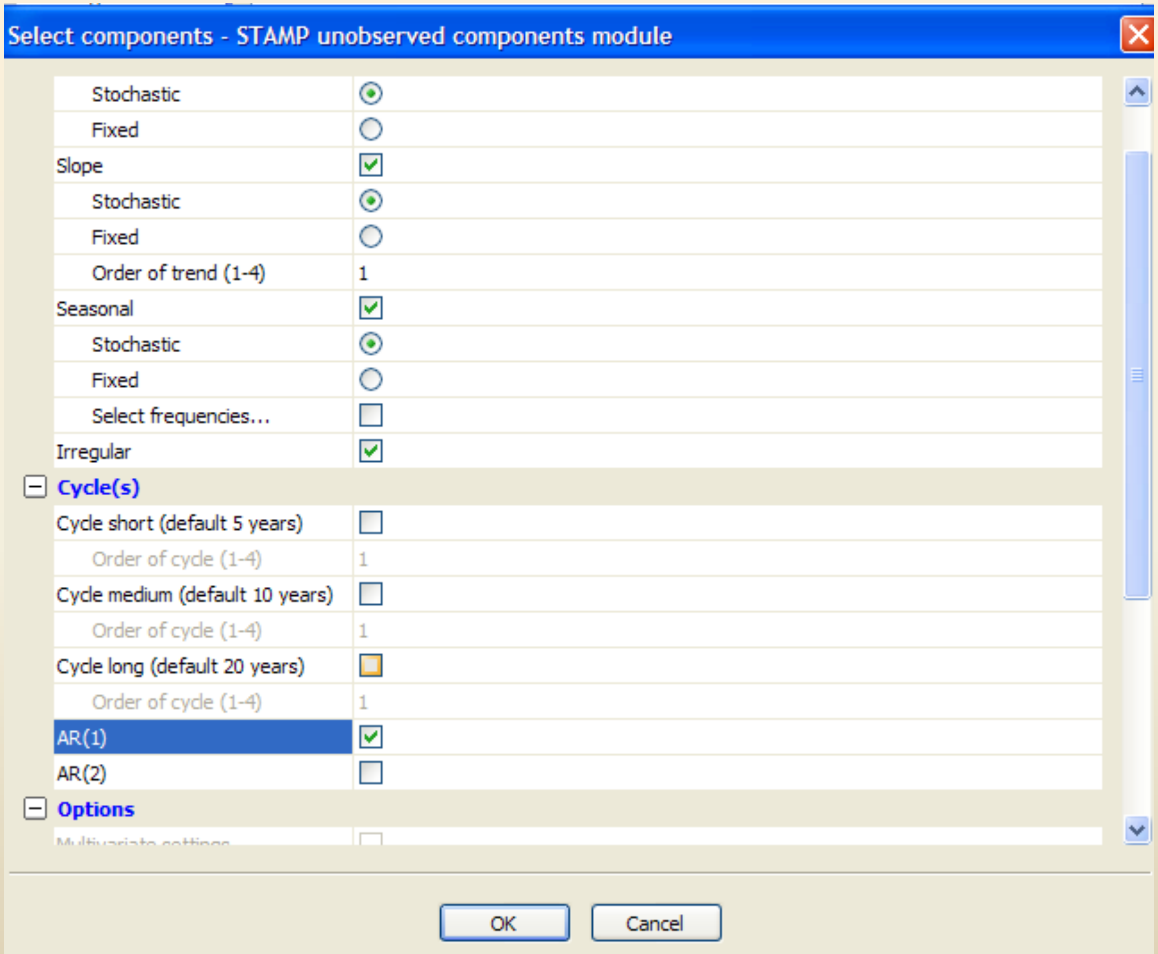
The dependent variable is normally distributed, residual serial correlation persists, and this could inflate the R^2 and t-tests.

```
Normality test for Residuals UKcars
                        Value
 Sample size           102.00
Mean                 0.091095
St.Dev                0.99584
Skewness             -0.18420
Excess kurtosis      -0.12521
Minimum               -2.4413
Maximum                2.7477

                      Chi^2       prob
Skewness             0.57682  [ 0.4476]
Kurtosis             0.06663  [ 0.7963]
Bowman-Shenton       0.64345  [ 0.7249]

Goodness-of-fit based on Residuals UKcars
                                                Value
Prediction error variance (p.e.v)          4.4347e+008
Prediction error mean deviation (m.d)      3.5376e+008
Ratio p.e.v. / m.d in squares                   1.0005
Coefficient of determination R^2               0.93456
... based on differences Rd^2                  0.83718
... based on diff around seas mean Rs^2        0.44403
Information criterion Akaike (AIC)              20.123
... Bayesian Schwartz (BIC)                     20.412

Serial correlation statistics for Residuals UKcars
 Durbin-Watson test is 2.05325
 Asymptotic deviation for correlation is 0.0990148
   Lag        df      Ser.Corr     BoxLjung        prob
     4         1      -0.15051       9.8817  [ 0.0017]
     8         5      0.073764       23.503  [ 0.0003]
    12         9      -0.12002       35.466  [ 0.0000]
```

Residuals are otherwise well-behaved. The slope (not shown here) residuals are also normally distributed.

```
Normality test for Irregular residual
                          Value
 Sample size            113.00
Mean                -0.0032692
St.Dev                 0.98884
Skewness               0.23553
Excess kurtosis       0.021461
Minimum                -2.2647
Maximum                 2.5556

                         Chi^2       prob
Skewness                1.0448  [ 0.3067]
Kurtosis             0.0021685  [ 0.9629]
Bowman-Shenton           1.047  [ 0.5925]

Normality test for Level residual
                          Value
 Sample size            111.00
Mean                 -0.011262
St.Dev                 0.98813
Skewness              -0.14663
Excess kurtosis       -0.52923
Minimum                -2.3309
Maximum                 1.9978

                         Chi^2       prob
Skewness               0.39776  [ 0.5282]
Kurtosis                1.2954  [ 0.2551]
Bowman-Shenton          1.6932  [ 0.4289]
```

As a last resort, we click on select components and add the ar(1) component and re-estimate



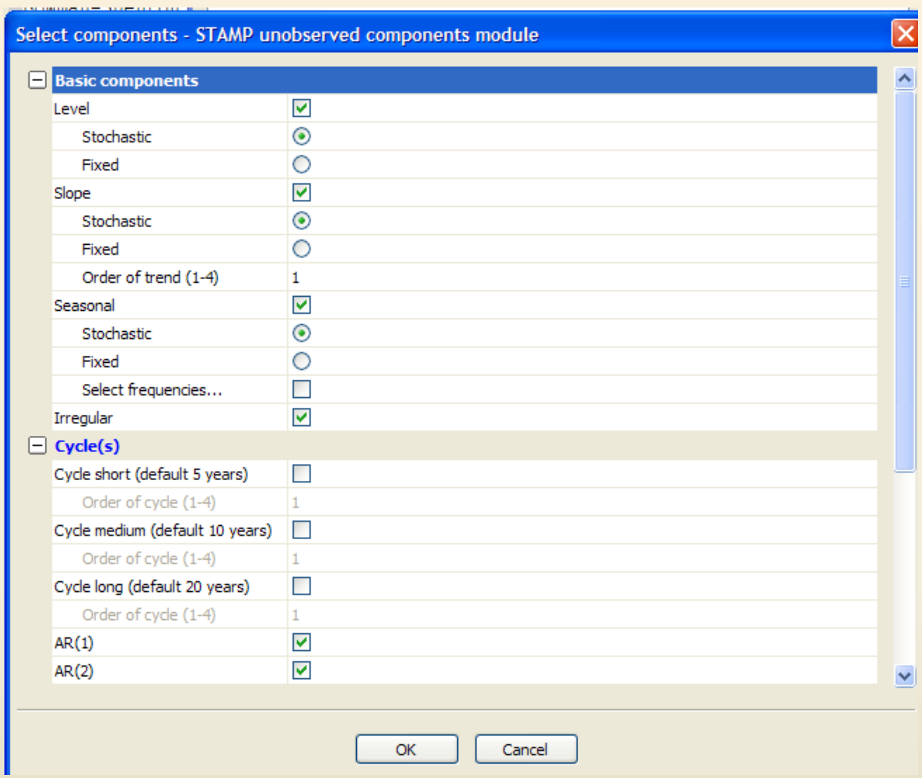
Steady state is found again on this pass

```
UC( 5) Estimation done by Maximum Likelihood (exact score)
    The database used is C:\Program Files\OxMetrics6\data\ukcars.csv
    The selection sample is: 1977(1) - 2005(1) (T = 113, N = 1)
    The dependent variable Y is: UKcars
    The model is:  Y = Trend + Seasonal + Irregular + AR(1) + Interventions
    Steady state. found

Log-Likelihood is -1026.08 (-2 LogL = 2052.17).
Prediction error variance is 4.32196e+008

Summary statistics
                   UKcars
 T                 113.00
 p                 5.0000
 std.error         20789.
 Normality        0.93851
 H(34)             1.0154
 DW                2.0385
 r(1)           -0.033170
 q                 14.000
 r(q)             0.13231
 Q(q,q-p)          29.775
 Rs^2             0.45817

 Variances of disturbances:
                    Value    (q-ratio)
Level             32508.0  (5.302e-005)
Slope             526970.  (0.0008595)
Seasonal     2.39504e+006  ( 0.003907)
AR(1)        6.13083e+008  (    1.000)
Irregular         113757.  (0.0001855)

AR(1) other parameters:
AR coefficient     0.71183
```

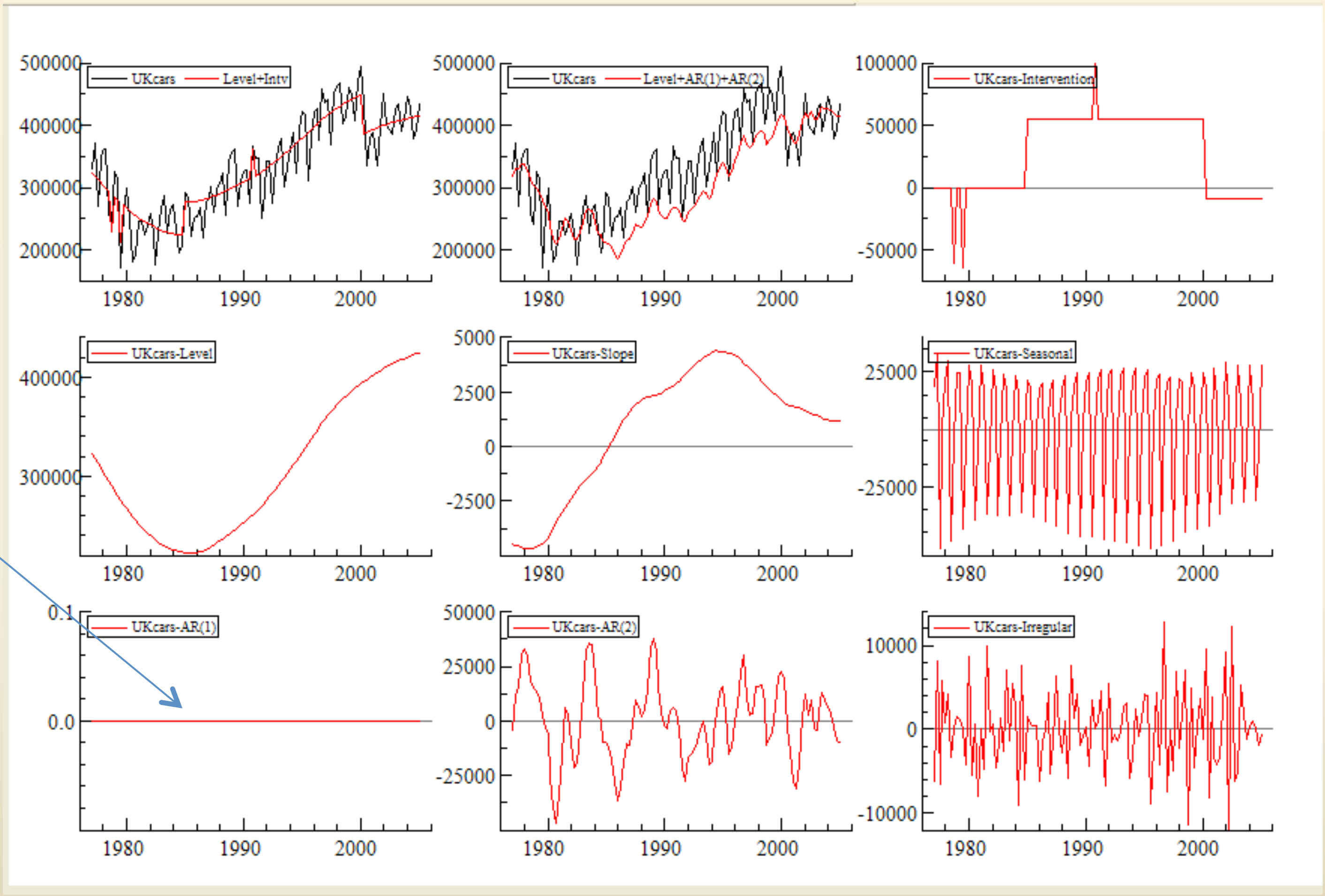
Some residual ar is attenuated, but not all. We enter the ar2 and re-estimate

```
Bowman-Shenton     0.57886  [ 0.7487]

Goodness-of-fit based on Residuals UKcars
                                               Value
Prediction error variance (p.e.v)          4.322e+008
Prediction error mean deviation (m.d)     3.4017e+008
Ratio p.e.v. / m.d in squares                  1.0277
Coefficient of determination R^2              0.93622
... based on differences Rd^2                 0.84132
... based on diff around seas mean Rs^2       0.45817
Information criterion Akaike (AIC)             20.097
... Bayesian Schwartz (BIC)                    20.386

Serial correlation statistics for Residuals UKcars
 Durbin-Watson test is 2.03846
 Asymptotic deviation for correlation is 0.0990148
   Lag       df     Ser.Corr    BoxLjung       prob
     4        0    -0.084352      5.7345   [ 1.0000]
     8        4     0.048756      16.389   [ 0.0025]
    12        8     -0.11991      27.159   [ 0.0007]

Normality test for Irregular residual
                      Value
 Sample size         113.00
Mean              0.0062564
St.Dev              0.98946
Skewness            0.20587
Excess kurtosis    0.052500
Minimum             -2.4043
Maximum              2.5012
```

We try an ar(1) and ar(2) and re-estimate.



Component graphics



AR1 does not seem functional

The variance-covariance matrix did not invert and a generalized inverse was used. This may be associate with the failure of AR(1) component.

```
 Full parameter report
 Actual parameters (all)
                           Value
Var Level                0.00000
Var Slope            5.2207e+005
Var Seasonal         2.6252e+006
Var AR(1)                0.00000
AR(1) coefficient 1      0.11531
Var AR(2)            5.0905e+008
AR(2) coefficient 1      0.51795
AR(2) coefficient 2      0.51836
Var Irregular        8.1471e+007
Warning: invertgen: invertsym failed, proceeding with generalized p.s.d. inverse
SEst.ox (2593): PrintPar
 Transformed parameters (not fixed)
                       Transform        1stDer        2ndDer     asymp.s.e
Var Slope                 6.5828   7.4301e-005     -0.057022       0.39707
Var Seasonal              7.3903    0.00030442      -0.10323       0.39615
AR(1) coefficient 1      -2.0377       0.00000       0.00000   6.4975e-016
Var AR(2)                 10.024    -0.0010687      -0.62373       0.18127
AR(2) coefficient 1     0.071841   -0.00025521     -0.045275        1.5582
AR(2) coefficient 2     0.073454   -0.00025835     -0.045332        1.5574
Var Irregular             9.1079    0.00068996      -0.26568       0.30715
 Actual parameters (not fixed) with 68% asymmetric confidence interval
                           Value     leftbound    rightbound
Var Slope            5.2207e+005   2.3596e+005   1.1551e+006
Var Seasonal         2.6252e+006   1.1887e+006   5.7976e+006
AR(1) coefficient 1      0.11531       0.11531       0.11531
Var AR(2)            5.0905e+008   3.5425e+008   7.3150e+008
AR(2) coefficient 1      0.51795       0.18447       0.83617
AR(2) coefficient 2      0.51836       0.18484       0.83628
Var Irregular        8.1471e+007   4.4077e+007   1.5059e+008

State vector at period 2005(1)
```

We can see that the AR(1) component did not work

State vector at period 2005(1)

	Coefficient	RMSE	t-value	Prob
Level	424546.93797	34489.62307	12.30941	[0.00000]
Slope	1149.54803	2428.03794	0.47345	[0.63690]
Seasonal	29629.21775	5654.74466	5.23971	[0.00000]
Seasonal 2	10745.27324	5985.14446	1.79532	[0.07553]
Seasonal 3	-1568.80378	4726.82869	-0.33189	[0.74064]
AR(1)	0.00000	0.00000	.NaN	[.NaN]
AR(2)	-9905.76290	18360.67871	-0.53951	[0.59070]
AR(2) 2	2424.62188	5055.14654	0.47963	[0.63250]

Regression effects in final state at time 2005(1)

	Coefficient	RMSE	t-value	Prob
Outlier 1978(4)	-61014.93554	16775.76961	-3.63709	[0.00043]
Outlier 1979(3)	-64257.53147	16400.80328	-3.91795	[0.00016]
Outlier 1990(4)	44941.31741	15990.80695	2.81045	[0.00592]
Level break 2000(2)	-64117.42562	19364.57045	-3.31107	[0.00128]
Level break 1985(1)	54913.02641	18998.22880	2.89043	[0.00469]

Irregular and slope residuals are good but slope residuals are not.

```
Normality test for Irregular residual
                         Value
 Sample size           113.00
Mean                -0.015329
St.Dev                 1.0136
Skewness              0.14430
Excess kurtosis      -0.15552
Minimum               -2.3730
Maximum                2.5258

                       Chi^2       prob
Skewness             0.39215  [ 0.5312]
Kurtosis             0.11388  [ 0.7358]
Bowman-Shenton       0.50604  [ 0.7765]

Normality test for Level residual
                         Value
 Sample size           112.00
Mean                -0.017971
St.Dev                0.97626
Skewness             -0.15121
Excess kurtosis      -0.55133
Minimum               -2.4257
Maximum                1.9597

                       Chi^2       prob
Skewness             0.42681  [ 0.5136]
Kurtosis              1.4185  [ 0.2337]
Bowman-Shenton        1.8453  [ 0.3975]
```

Residual serial correlation and misbehaved slope residuals plague this model (failure of AR(1)) component

```
Goodness-of-fit based on Residuals UKcars
                                                Value
Prediction error variance (p.e.v)         4.3824e+008
Prediction error mean deviation (m.d)     3.4892e+008
Ratio p.e.v. / m.d in squares                  1.0043
Coefficient of determination R^2               0.9347
... based on differences Rd^2                 0.83752
... based on diff around seas mean Rs^2        0.4452
Information criterion Akaike (AIC)             20.093
... Bayesian Schwartz (BIC)                    20.358

Serial correlation statistics for Residuals UKcars
 Durbin-Watson test is 2.08782
 Asymptotic deviation for correlation is 0.0985329
   Lag       df      Ser.Corr     BoxLjung        prob
     4       -1     -0.084399       4.5647   [ 1.0000]
     8        3      0.033324       14.458   [ 0.0023]
    12        7     -0.076398       24.558   [ 0.0009]
```

```
Normality test for Slope residual
                          Value
 Sample size             111.00
Mean                    0.30453
St.Dev                  0.93478
Skewness              0.0084911
Excess kurtosis         -1.2212
Minimum                 -1.6725
Maximum                  2.0243

                         Chi^2         prob
Skewness             0.0013338   [ 0.9709]
Kurtosis                6.8975   [ 0.0086]
Bowman-Shenton          6.8988   [ 0.0318]
```

We try again with and hope that a different starting value will lead to a more propitious result.

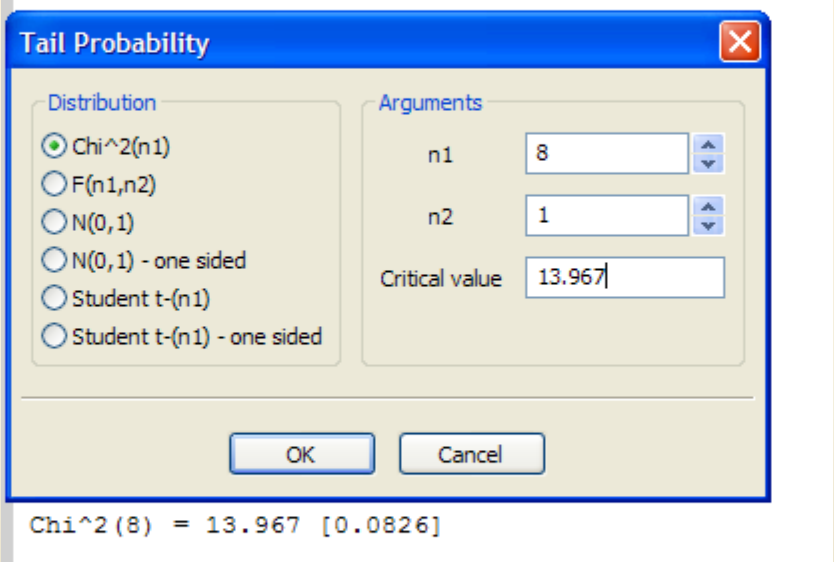
```
Estimating.......
Strong convergence relative to 1e-007
- likelihood cvg 0
- gradient cvg 8.18304e-005
- parameter cvg 0
- number of bad iterations 5
........
Strong convergence relative to 1e-007
- likelihood cvg 0
- gradient cvg 0.000234708
- parameter cvg 0
- number of bad iterations 5
Estimation process completed.

UC( 7) Estimation done by Maximum Likelihood (exact score)
    The database used is C:\Program Files\OxMetrics6\data\ukcars.csv
    The selection sample is: 1977(1) - 2003(1) (T = 105, N = 1)
    The dependent variable Y is: UKcars
    The model is:  Y = Trend + Seasonal + Irregular + AR(1) + AR(2) + Interventions
    Steady state. found

Log-Likelihood is -958.678 (-2 LogL = 1917.36).
Prediction error variance is 4.57688e+008
```

This time a stead state is found, and although there were 5 bad iterations the model converged.

This time the Box-Ljung Q is smaller and the ar(1) and ar(2) worked.

```
Log-Likelihood is -958.678 (-2 LogL = 1917.36).
Prediction error variance is 4.57688e+008

Summary statistics
                    UKcars
 T                  105.00
 p                  7.0000
 std.error          21394.
 Normality          1.5991
 H(31)              1.2357
 DW                 1.9409
 r(1)             0.026838
 q                  15.000
 r(q)             0.068166
 Q(q,q-p)           13.967
 Rs^2              0.44419

 Variances of disturbances:
                     Value     (q-ratio)
Level             0.000000  (    0.0000)
Slope              503794.  (   0.04722)
Seasonal      4.84139e+006  (    0.4537)
AR(1)         1.21538e+006  (    0.1139)
AR(2)         5.37549e+008  (     50.38)
Irregular     1.06702e+007  (     1.000)

AR(1) other parameters:
AR coefficient      0.60756
AR(2) other parameters:
AR coefficient         0.60756
AR(1) coefficient      1.05169
AR(2) coefficient     -0.27621
```

A Box-Ljung Q test with 8 df shows that the autocorrelation of the residuals is not significant.



The residuals are no longer signficantly autocorrelated.

```
Goodness-of-fit based on Residuals UKcars
                                                    Value
Prediction error variance (p.e.v)             4.5769e+008
Prediction error mean deviation (m.d)         4.0859e+008
Ratio p.e.v. / m.d in squares                      0.7988
Coefficient of determination R^2                  0.93111
... based on differences Rd^2                     0.83889
... based on diff around seas mean Rs^2           0.44419
Information criterion Akaike (AIC)                 20.151
... Bayesian Schwartz (BIC)                        20.429

Serial correlation statistics for Residuals UKcars
 Durbin-Watson test is 1.9409
 Asymptotic deviation for correlation is 0.102598
   Lag        df      Ser.Corr    BoxLjung        prob
     4        -1      -0.10978      3.4393   [ 1.0000]
     8         3       0.10541       7.628   [ 0.0544]
    12         7     -0.067548      13.134   [ 0.0689]
```

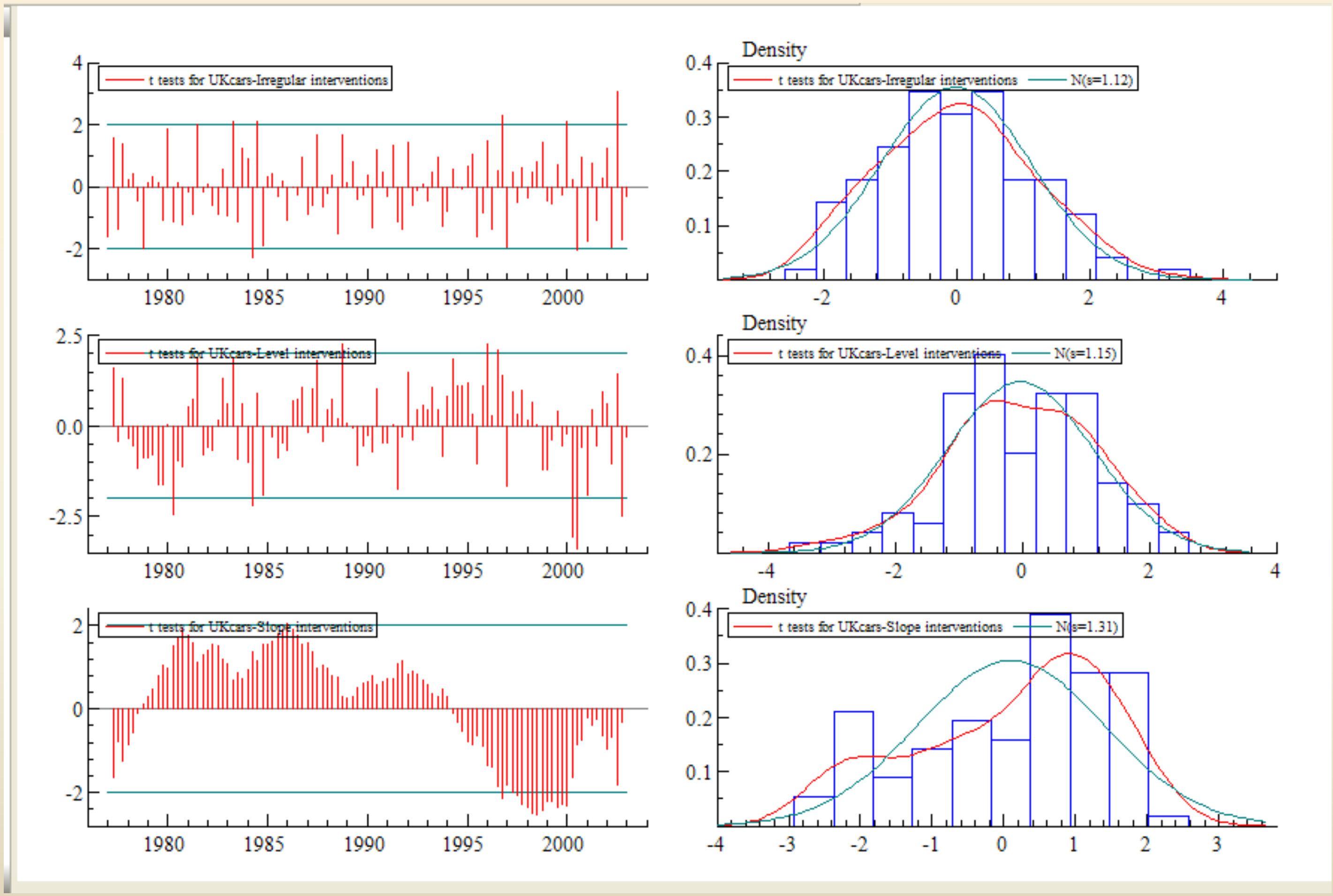
Irregular and level residuals remain well-behaved.

```
Normality test for UKcars-Irregular residual
                      Value
 Sample size         105.00
Mean              -0.0073490
St.Dev               1.1223
Skewness            0.17703
Excess kurtosis    -0.42145
Minimum             -2.2919
Maximum              3.0698

                     Chi^2       prob
Skewness           0.54846  [ 0.4589]
Kurtosis            0.7771  [ 0.3780]
Bowman-Shenton      1.3256  [ 0.5154]

Values larger than 3 for UKcars-Level residual:
                    Value      prob
2000(2)          -3.04466 [0.00147]
2000(3)          -3.36943 [0.00053]

Normality test for UKcars-Level residual
                      Value
 Sample size         103.00
Mean               -0.027816
St.Dev               1.1513
Skewness           -0.32333
Excess kurtosis   -0.015384
Minimum             -3.3694
Maximum              2.2561
```

Only the slope residual is potentially problematic

```
Normality test for UKcars-Level residual
                     Value
 Sample size        103.00
Mean             -0.027816
St.Dev              1.1513
Skewness          -0.32333
Excess kurtosis  -0.015384
Minimum            -3.3694
Maximum             2.2561

                    Chi^2        prob
Skewness           1.7947  [ 0.1804]
Kurtosis        0.0010158  [ 0.9746]
Bowman-Shenton     1.7957  [ 0.4075]

Normality test for UKcars-Slope residual
                     Value
 Sample size        103.00
Mean               0.10017
St.Dev              1.3109
Skewness          -0.56602
Excess kurtosis   -0.86870
Minimum            -2.5361
Maximum             2.0508

                    Chi^2        prob
Skewness           5.4998  [ 0.0190]
Kurtosis           3.2386  [ 0.0719]
Bowman-Shenton     8.7384  [ 0.0127]
```

However, the auxiliary residuals indicate that the problem area for the slope is before 2000.



We decide to use this model for forecasting

- Modeling residuals before 2000 would not help solve the problem with the slope residuals.
- We therefore suspect that this is about the best model that we can get with these data.

- This is confirmed by a likelihood ratio test of the LL for the last model and this model.
- Hence, our decision to use it as a basis for out-of-sample (1 year) forecast.
- We set the date of forecast origin to

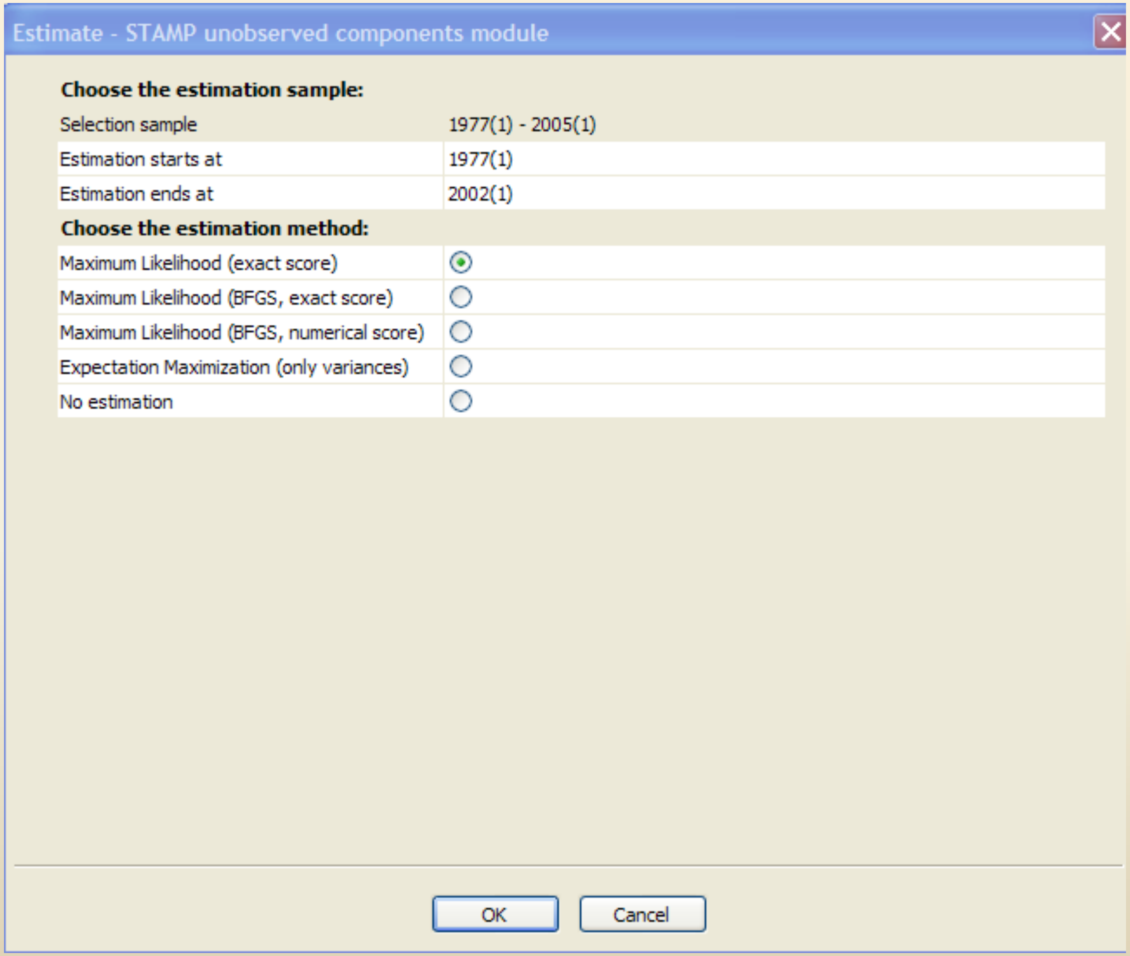
We find a significant improvement between the last and the current model.

```
Progress to date
Model                T     p          log-likelihood           SC
UC( 6)             113     7  Maximum Likelihood (exact score)
UC( 7)             105     8  Maximum Likelihood (exact score)
 Chi^2(1) = 77.313 [0.0000] **
```

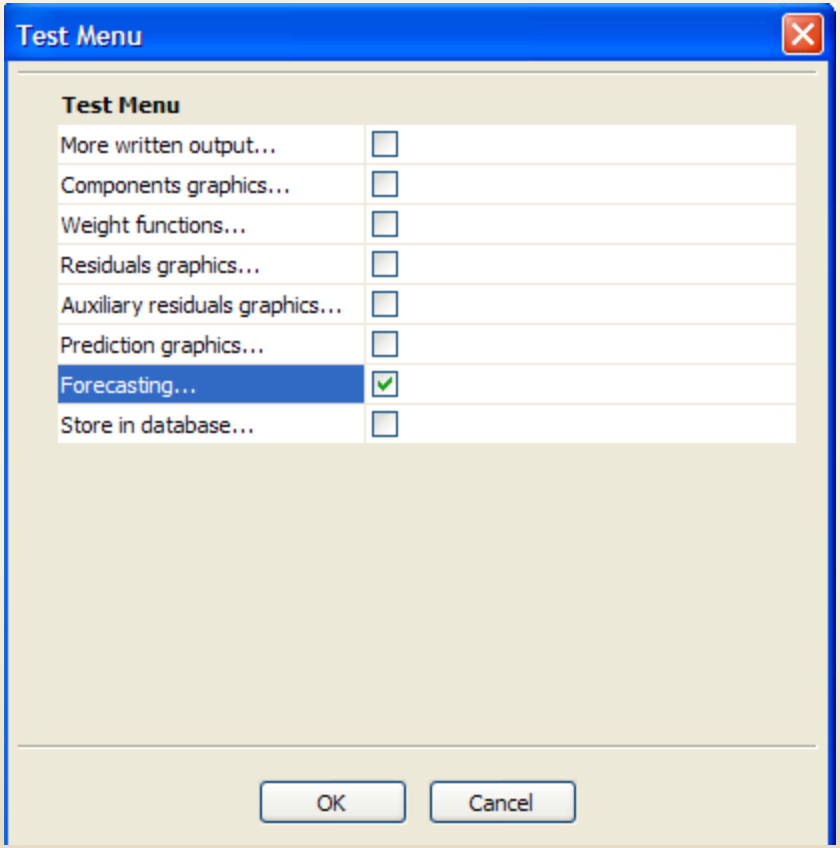
Out-of-sample forecasting

- We decide to forecast over a horizon of one year, with the forecast origin set at 2002(1).
- To do so, we have to reset the estimation period.
- The remainder of time the data span will be called the validation segment of the data and will be used to test the accuracy of the forecast.

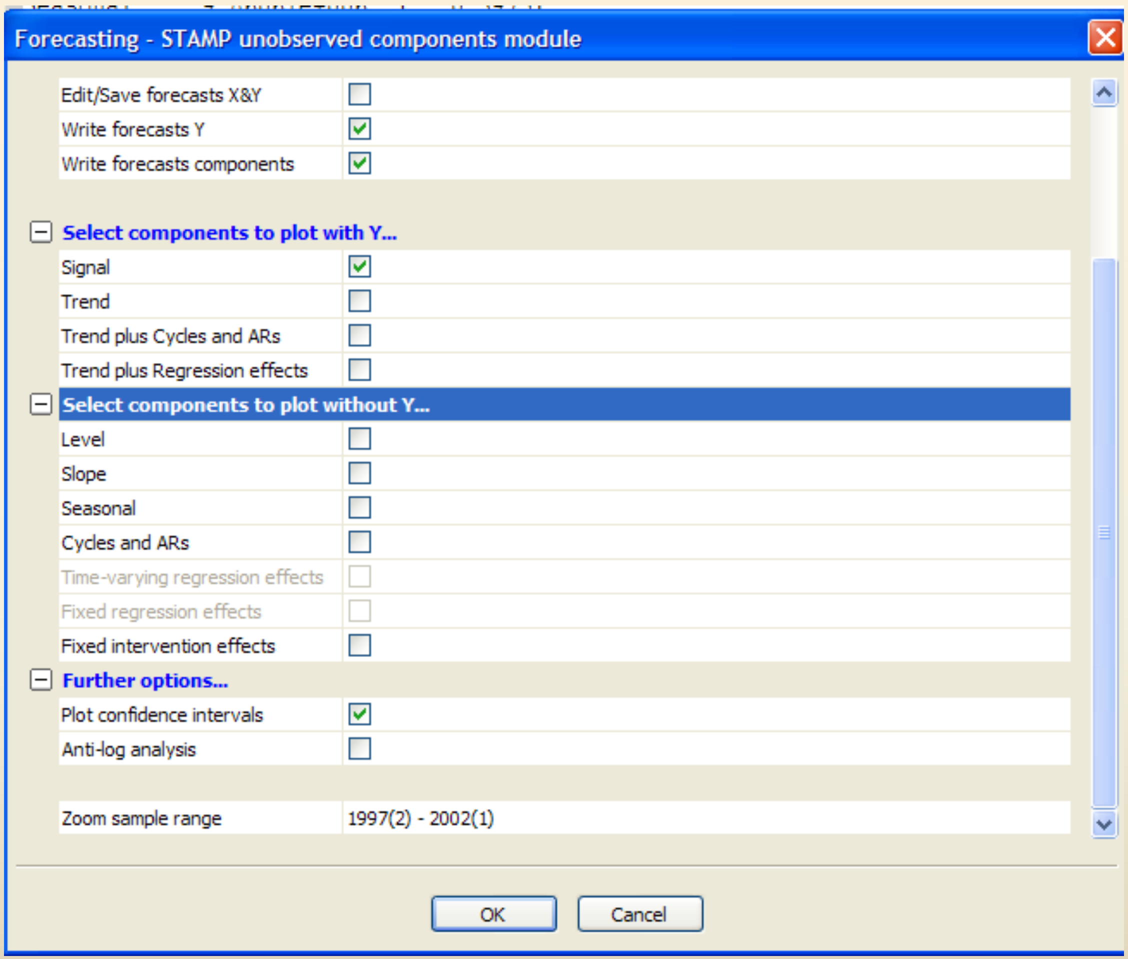
We reformulate and when we come to the estimation period, we set the forecast origin to 2002(1)



Then we go to the test menu and select forecasting



We are then presented with the forecast menu and select:



Stamp provides forecasts and evaluations over their forecast horizon

Forecasts with 68% confidence interval from period 2002(1) forwards:

	Forecast	stand.err	leftbound	rightbound
1	429678.53364	21681.73062	407996.80302	451360.26426
2	379333.78944	29574.04509	349759.74434	408907.83453
3	430063.36012	35881.17977	394182.18035	465944.53988
4	461409.32726	38900.70874	422508.61852	500310.03600
5	433303.11375	45150.85676	388152.25699	478453.97051
6	379487.26281	48534.39887	330952.86394	428021.66167
7	429173.50164	51589.51195	377583.98969	480763.01358
8	460493.47752	53212.57159	407280.90593	513706.04911
9	432718.90920	57626.73875	375092.17044	490345.64795
10	379303.51494	60401.47215	318902.04279	439704.98710
11	429347.22775	63220.45218	366126.77557	492567.67993
12	460950.04895	64951.33082	395998.71813	525901.37977

Forecast accuracy measures from period 2002(1) forwards:

	Error	RMSE	RMSPE	MAE	MAPE
1	27426.53364	27426.53364	0.68182	27426.53364	6.81825
2	-12513.21056	21316.60370	0.53238	19969.87210	5.00582
3	44173.36012	30876.53332	0.79104	28037.70144	7.15293
4	37084.32726	32539.69982	0.81256	30299.35790	7.54960
5	23.11375	29104.39415	0.72678	24244.10907	6.04074
6	-11725.73719	26996.36304	0.67465	22157.71375	5.53350
7	20433.50164	26159.81022	0.65255	21911.39774	5.45716
8	15035.47752	25041.00457	0.62196	21051.90771	5.19693
9	4516.90920	23656.84714	0.58745	19214.68565	4.73670
10	255.51494	22443.00122	0.55730	17318.76858	4.26977
11	35305.22775	23900.06113	0.59610	18953.90123	4.69613
12	28154.04895	24283.04040	0.60082	19720.58021	4.84688

Criteria of forecast evaluation

- Criteria:

$$Error=(y-f)$$

$$RMSE=\sqrt{\frac{\sum_{i=1}^{T}\left(y_i-\hat{y}_i\right)^2}{T}}$$

$$RMSPE=\sqrt{\frac{\sum_{i=1}^{H}\left(y_i-f_i\right)^2}{H}}*100$$

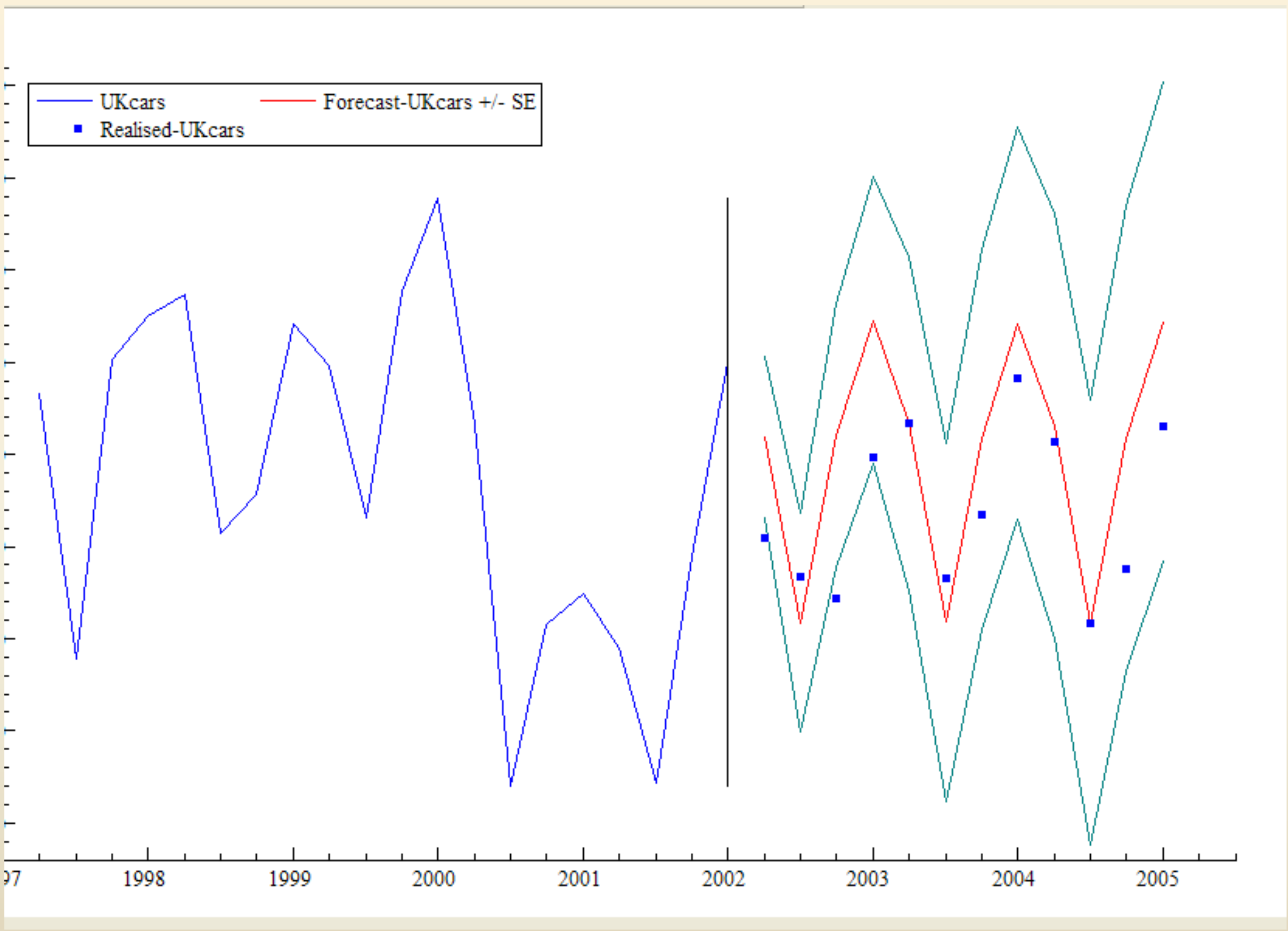
$$MAE=\frac{\sum_{i=1}^{H}|y-\hat{y}|}{H}$$

$$MAPE=100*\frac{\sum_{i=1}^{H}|y-\hat{y}|}{H}$$

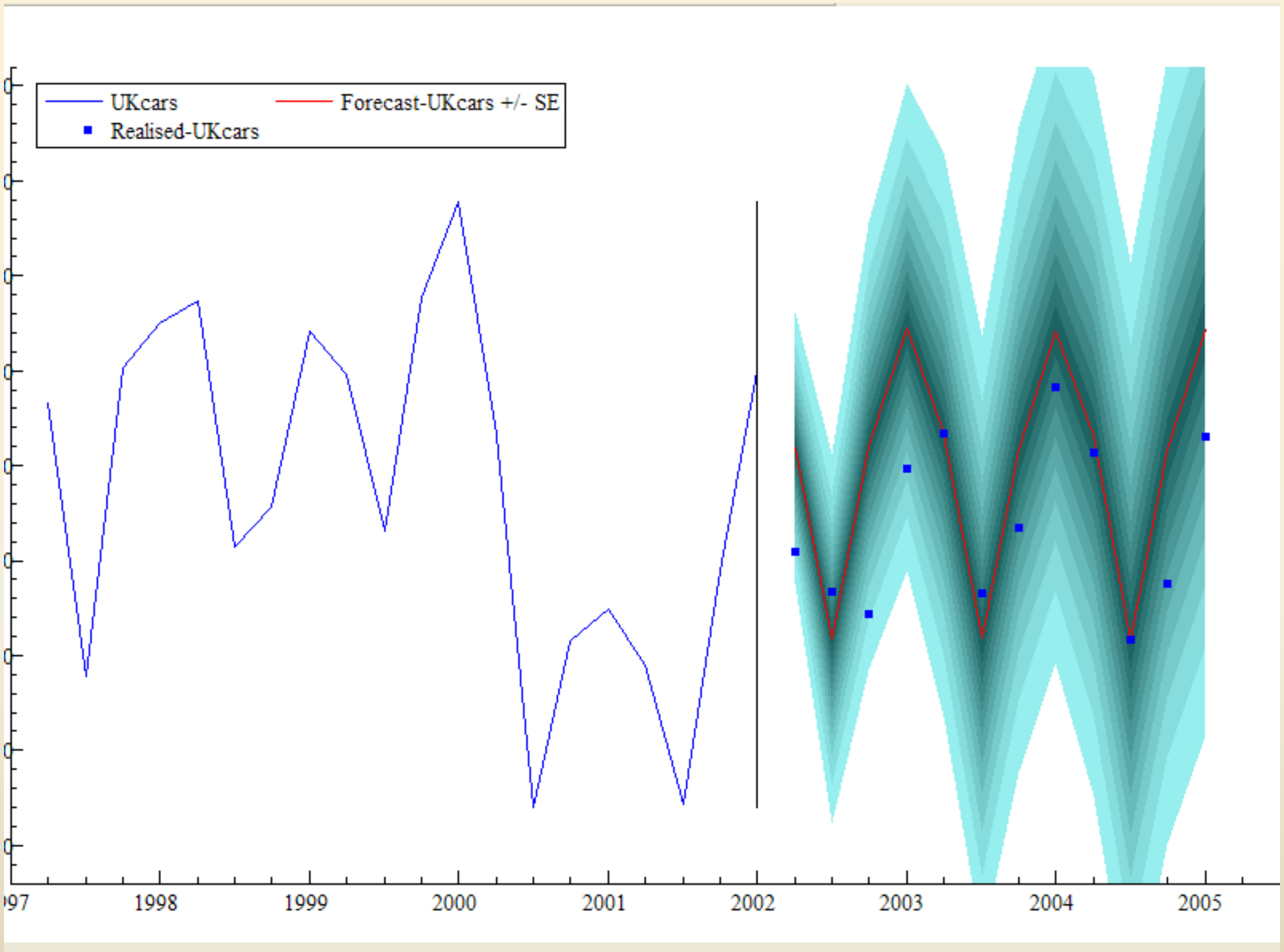
Stamp will also provide forecasts and confidence intervals of those forecasts for each component as well(including AR(1) and AR(2)).

```
Forecast values for Level
Forecasts with 68% confidence interval from period 2002(1) forwards:
           Forecast   stand.err   leftbound  rightbound
1      354856.39999 33151.93043321704.46956388008.33042
2      355137.34377 34816.42015320320.92361389953.76392
3      355418.28754 36656.63551318761.65203392074.92305
4      355699.23132 38665.30696317033.92436394364.53828
5      355980.17510 40834.44925315145.72584396814.62435
6      356261.11887 43155.83164313105.28724399416.95051
7      356542.06265 45621.31810310920.74456402163.38075
8      356823.00643 48223.09462308599.91181405046.10105
9      357103.95021 50953.80654306150.14367408057.75675
10     357384.89398 53806.62906303578.26492411191.52304
11     357665.83776 56775.29103300890.54673414441.12879
12     357946.78154 59854.06794298092.71359417800.84948
Ssf() warning: SLOPE can not be part of signal

Forecast values for Seasonal
Forecasts with 68% confidence interval from period 2002(1) forwards:
           Forecast   stand.err   leftbound  rightbound
1        7334.96779  8522.78876 -1187.82097 15857.75655
2      -46174.62175  8499.90280-54674.52454-37674.71895
3        3715.91763  8520.94847 -4805.03083 12236.86610
4       35123.73632  9569.83400 25553.90233 44693.57032
5        7334.96779 10560.43744 -3225.46965 17895.40523
6      -46174.62175 10541.97601-56716.59775-35632.64574
7        3715.91763 10558.95228 -6843.03465 14274.86991
8       35123.73632 11422.19915 23701.53718 46545.93547
9        7334.96779 12264.08372 -4929.11593 19599.05151
10     -46174.62175 12248.19043-58422.81217-33926.43132
11       3715.91763 12262.80490 -8546.88727 15978.72253
12      35123.73632 13013.51390 22110.22243 48137.25022
```

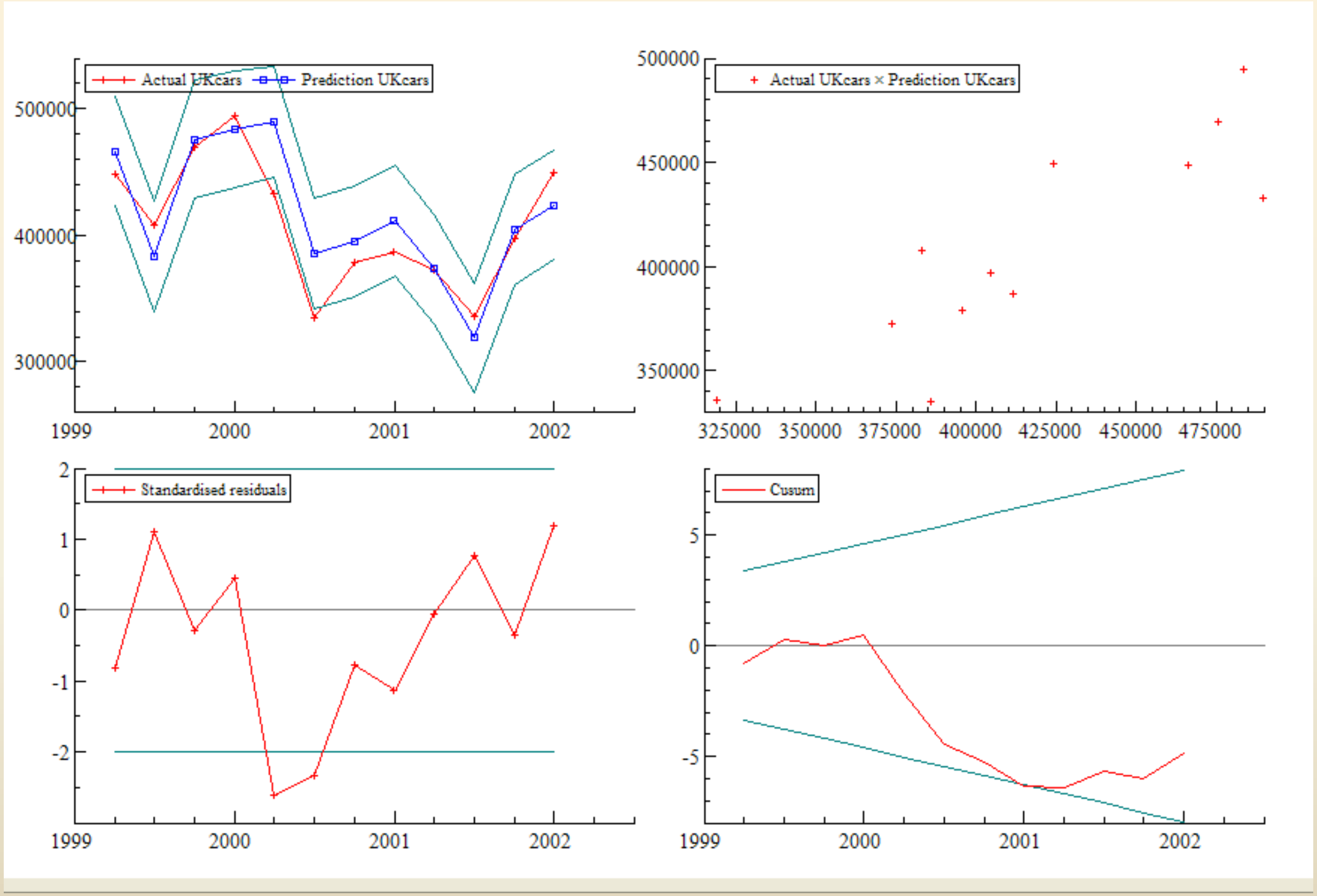
Forecasts can be graphed as well



Forecast Profiles may be edited and changed into forecast fan charts or error bar charts



Other prediction graphics are available, including coverage, cumsum, and cumsum squared plots.



Diagnosing the State Space model

Residuals are used for diagnosis. These are the innovations. But the auxiliary residuals are estimators of the disturbances associated with the unobserved components. Although they are related to the residuals, they may display the information somewhat differently.

Residuals are useful for diagnostics in large samples. However, in finite samples, auxiliary residuals may be more helpful. They may be regarded as minimum mean square estimators under conditions of Gaussianity, according to Harvey and Koopman 1992, in Harvey and Proietti (2005), 84.

Auxiliary residuals are serially correlated. However, they are useful in detecting outliers and level shifts. The Bowman-Shenton test , which is distributed as a χ^2 test with 2 df, is modified to account for this autocorrelation, they can be used to distinguish between them as well.

Computation of Auxiliary Residuals

Just run the Kalman filter and then the smoother. When computing the variances at the beginning or end of the series, they will seem very large compared to the others.

For test statistics, only the observations in the middle of the series should be used. The variances at either end are much larger.

Auxiliary residuals are standardized for presentation.
By dividing the residual by the square root of the variance t-tests for significance are obtained. Using all of the data as a basis for the significance test, these auxiliary residuals are usually preferred for the first pass of the diagnostics of model adequacy.

Applications of the Auxiliary Residuals

Testing for a level shift is best done with the use of the auxiliary residuals.

Testing for seasonal change would be better done with auxiliary residuals.

Testing for an individual outlier is perhaps better done with residuals that are not autocorrelated. Harvey and Koopman argue that auxiliary residuals combine in the best way to use them for testing in this kind of case (*Ibid, 86)*

Tests based on skewness and kurtosis:

Forecasting with State Space Models

Three methods are provided with Stata's space.

One-step-ahead forecasting is performed by the Kalman filter in its filtering process.

Iterative projection results from repeated application of this process.

Forecast evaluation

Forecast evaluation is performed by out-of-sample comparison of the forecasts to the actual data.

Aside from the predictive error variance computed from the predictive error decomposition, error, Root mean square error, root mean square percentage error, mean absolute error, and mean absolute percentage error are criteria employed to evaluate the forecast accuracy.

Other predictive graphics tests can also be applied to the forecasts for evaluation: Predictive error variance, cumsum, and cumsum squares, and Chow's predictive failure test are among them.

Relationship of State Space to ARIMA models

(Ruey Tsay, class notes)

Cayley-Hamilton Theorem:
for any m x m matrix F, with characteristic equation, such a matrix is reduce able to an ARIMA model. (Details are not presented here).

What do ARIMA models look like when presented as State Space System form?
We consider just a few cases.

- The AR(1) case
- The AR(2) case
- The MA(1) case
- The MA(2) case
- The ARMA(1,1) case
- The ARMA(2,2) case
- The ARIMA(0,1,1) case
- The ARIMA(0,2,2) case

The equations are stacked within System matrices (α, Φ, Ω, and Σ, T, Z, R and H(=1 and in front of epsilon)

$$
\begin{aligned}
&If\ \ \alpha_{t+1} = T_t\alpha_t + R_t\eta_t, \quad \eta_t = iid\ \ N(0,\sigma^2_{\eta_t})\ \ transition\ equation \\
&\quad y_t = Z_t\alpha_t + \varepsilon_t, \qquad \varepsilon_t = iid\ \ N(0,\sigma^2_{\varepsilon_t})\ \ measurement\ equation, \\
&then \qquad \alpha_{t+1} \qquad\qquad\qquad\qquad \Phi_t \\
&\begin{pmatrix} \alpha_{t+1} \\ y_t \end{pmatrix} = \begin{pmatrix} T_t \\ Z_t \end{pmatrix}(\alpha_t) + \begin{pmatrix} R_t\eta_t \\ \varepsilon_t \end{pmatrix} \qquad\qquad \Omega_t = \begin{pmatrix} \sigma^2_\eta & \\ 0 & \sigma^2_\varepsilon \end{pmatrix} \\
&where \\
&y_t = \mu_t + \varepsilon_t\ , \quad \varepsilon_t =\ \ iid\ \ N(0,\sigma^2_{\varepsilon_t}) \qquad \Sigma = \begin{pmatrix} -1 \\ 0 \end{pmatrix} \\
&\alpha_1 \sim N(\alpha_1, P_1) \\
&NB: In\ a\ local\ level\ Model: T_t = I,\ so \\
&\alpha_{t+1} = \alpha_t + R_t\eta_t
\end{aligned}
$$

SsfPack program to generate AR(1) and AR(2) system matrices

GetSsfArma_eg1.ox - C:\Program Files\OxMetrics6\Ox\packages\ssfpack\GetSsfArma_eg1.ox

```
#include <oxstd.h>
#include <oxdraw.h>
#import  <maximize>
#include <c:\Program Files\OxMetrics51\Ox\packages\ssfpack\ssfpack_ex.h>


main()
{
decl mPhi, mOmega, mSigma;
 println("System matrices for an AR1 model");
 println("AR(1) = 0.6   sigma_eps^2=.4 ");
 println("________________________________");

 GetSsfArma
 (<.6>,<>,sqrt(.4),  &mPhi, &mOmega, &mSigma);
 print("Phi = ", mPhi, "Omega = ", mOmega, "Sigma = ",mSigma);

println ("_________________________________________________________");

 println(" Specification of the AR2 Model");
 println( "AR1 = .5, AR2= -.4,  sigma_eps^2 = .9");

GetSsfArma
 (<0.5,-0.4>,<>, sqrt(0.9), &mPhi, &mOmega, &mSigma);
 print("Phi = ", mPhi, "Omega = ", mOmega, "Sigma = ",mSigma);
println ("_________________________________________________________");
```

ARIMA(1,0,0) and ARIMA(2,0,0) state space system matrices

AR(1) model: ϕ_1=0.6, σ^2=.4; AR(2) model: ϕ_1=0.5, ϕ_2=-.4, σ^2=.9

$\phi(1)$ coef

$\sigma(2)$ coef

GetSsfArma_eg1.out

```
System matrices for an AR1 model
AR(1) = 0.6    sigma_eps^2=.4
________________________________
Phi =
      0.60000
       1.0000
Omega =
      0.40000       0.00000
      0.00000       0.00000
Sigma =
      0.62500
      0.00000
__________________________________________________
 Specification of the AR2 Model
AR1 = .5, AR2= -.4,  sigma_eps^2 = .9
Phi =
      0.50000        1.0000
     -0.40000       0.00000
       1.0000       0.00000
Omega =
      0.90000       0.00000       0.00000
      0.00000       0.00000       0.00000
      0.00000       0.00000       0.00000
Sigma =
       1.2281      -0.17544
     -0.17544       0.19649
      0.00000       0.00000
```

unconditional variance of initial state is in Sigma

ARIMA(0,0,1) and ARIMA(0,0,2) state space system matrices SsfPack code

GetSsfArma_eg1.ox - C:\Program Files\OxMetrics6\Ox\packages\ssfpack\GetSsfArma_eg1.ox

```
GetSsfArma
 (<0.5,-0.4>,<>, sqrt(0.9), &mPhi, &mOmega, &mSigma);
 print("Phi = ", mPhi, "Omega = ", mOmega, "Sigma = ",mSigma);
println ("____________________________________________________________");

println(" MA(1) Model");
println( "MA1 = .4  sigma_eps = .6 ");
GetSsfArma(<>,<.4>,sqrt(.6), &mPhi, &mOmega, &mSigma);
print("Phi = ", mPhi, "Omega = ", mOmega, "Sigma = ",mSigma);

println("______________________________________________________________");

 println(" The Following specification is that of an MA(2) model");
 println( "The parms are ar=0, ma1 = -.3, ma2 = - .4, sigma=sqrt(5)");


GetSsfArma
(<>,<-0.3,-.4>,sqrt(.5), &mPhi, &mOmega, &mSigma);
print("Phi = ",mPhi," Omega = ",mOmega," Sigma = ",mSigma);



}
```

State Space System matrices for MA(1) model $\theta_1 = .4$ $\sigma^2 = .6$

GetSsfArma_eg1.out

```
 MA(1) Model
MA1 = .4  sigma_eps = .6
Phi =
      0.00000       1.0000
      0.00000      0.00000
       1.0000      0.00000
Omega =
      0.60000      0.24000      0.00000
      0.24000     0.096000      0.00000
      0.00000      0.00000      0.00000
Sigma =
      0.69600      0.24000
      0.24000     0.096000
      0.00000      0.00000
```

.4*.6

State Space System Matrices for MA(2) model θ_1=-0.3 θ_2 =-0.4 σ^2=5

GetSsfArma_eg1.out

```
 The Following specification is that of an MA(2) model
The parms are ar=0, ma1 = -.3, ma2 = - .4, sigma_eps^2=5
Phi =
      0.00000        1.0000      0.00000
      0.00000       0.00000       1.0000
      0.00000       0.00000      0.00000
       1.0000       0.00000      0.00000
 Omega =
      0.50000      -0.15000     -0.20000      0.00000
     -0.15000      0.045000     0.060000      0.00000
     -0.20000      0.060000     0.080000      0.00000
      0.00000       0.00000      0.00000      0.00000
 Sigma =
      0.62500     -0.090000     -0.20000
    -0.090000       0.12500     0.060000
     -0.20000      0.060000     0.080000
      0.00000       0.00000      0.00000
```

-0.3 x .5

-0.4 x .5

SsfPack code snippet for system matrices for ARMA(1,1) and ARMA(2,2) models

GetSsfARMA.ox - C:\Program Files\OxMetrics6\Ox\packages\ssfpack\CKbook\Chapter_3\GetSsf...

```

println("                    ARMA(1,1)    Models      ");
println(" ar1 = .6    ma = -0.3        sigma_esp^2=.4    ");
 GetSsfArma
(<.6>,<-0.3>,  sqrt(.4), &mPhi, &mOmega, &mSigma);
print("Phi = ",mPhi," Omega = ",mOmega," Sigma = ",mSigma);

println("____________________________________________________________");

println("                    ARMA(2,2)    Models      ");
println(" ar1 = 0.6  ar2= -0.3  ma1 = -0.3  ma2=.5     sigma_esp^2=.4    ");
 GetSsfArma
(<.6,-.3>,<-0.3,.5>,  sqrt(.4), &mPhi, &mOmega, &mSigma);
print("Phi = ",mPhi," Omega = ",mOmega," Sigma = ",mSigma);

}
```

System matrices for ARMA(1,1) models
Can you tell from output what the parameters are?

GetSsfARMA.out

```
                    ARMA(1,1)     Models
 ar1 = .6     ma = -0.3        sigma_esp^2=.4
Phi =
      0.60000       1.0000
      0.00000      0.00000
       1.0000      0.00000
 Omega =
      0.40000     -0.12000      0.00000
     -0.12000     0.036000      0.00000
      0.00000      0.00000      0.00000
 Sigma =
      0.45625     -0.12000
     -0.12000     0.036000
      0.00000      0.00000
```

ARMA(2,2) system matrices

Can you tell what the parameters are from this output (ignoring the listing of
Them on the top?

GetSsfARMA.out

```
                   ARMA(2,2)     Models
 ar1 = 0.6   ar2= -0.3   ma1 = -0.3   ma2=.5      sigma_esp^2=.4
Phi =
      0.60000       1.0000       0.00000
     -0.30000      0.00000        1.0000
      0.00000      0.00000       0.00000
       1.0000      0.00000       0.00000
 Omega =
      0.40000     -0.12000       0.20000      0.00000
     -0.12000     0.036000     -0.060000      0.00000
      0.20000    -0.060000       0.10000      0.00000
      0.00000      0.00000       0.00000      0.00000
 Sigma =
      0.50354     -0.11588       0.20000
     -0.11588     0.061319     -0.060000
      0.20000    -0.060000       0.10000
      0.00000      0.00000       0.00000
```

SsfPack code snippet: System matrices for an ARIMA(1,1,1) model

ARIMA.ox - C:\Program Files\OxMetrics6\Ox\packages\ssfpack\CKbook\Chapter_3\ARIMA.ox

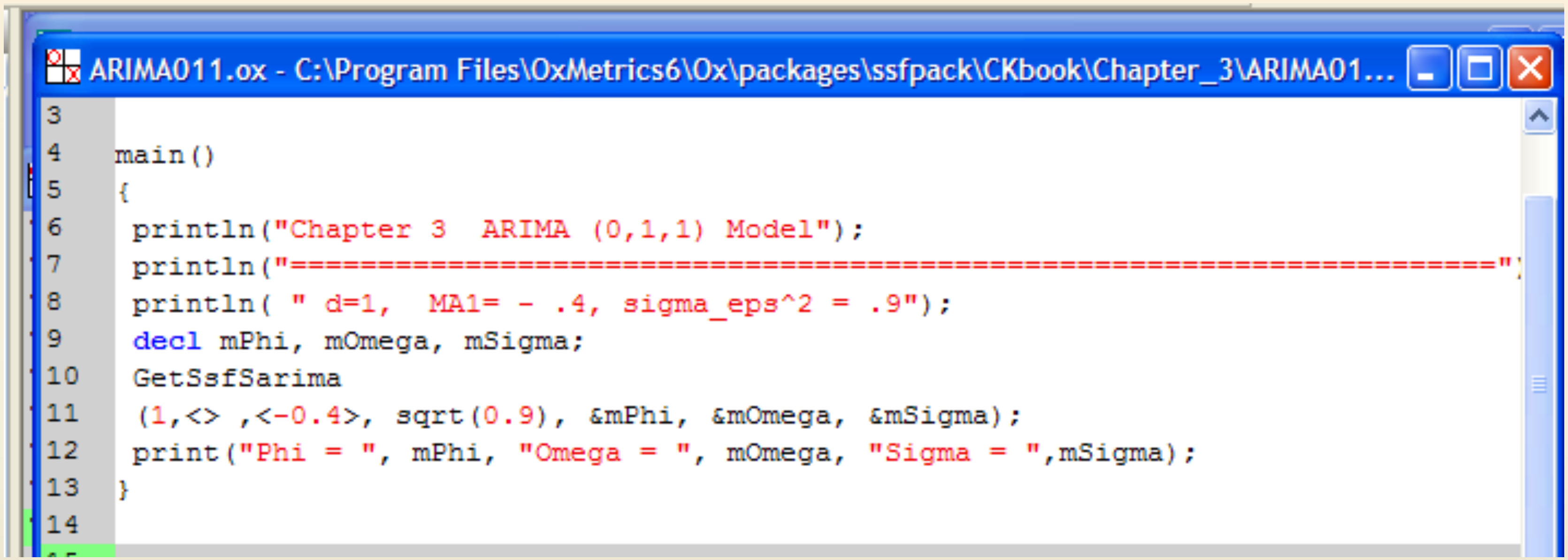
```
#include <oxstd.h>
#include <c:\Program Files\OxMetrics51\Ox\packages\ssfpack\ssfpack_ex.h>

main()
{
 println("Chapter 3  ARIMA Model(with 1st differencing)");
 println("=====================================================================";
 println( " d=1, ar1=.6,  MA1= - .4, sigma_eps^2 = .9");
 decl mPhi, mOmega, mSigma;
 GetSsfSarima
 (1,<0.6>,<-0.4>, sqrt(0.9), &mPhi, &mOmega, &mSigma);
 print("Phi = ", mPhi, "Omega = ", mOmega, "Sigma = ",mSigma);
}

```

State space system matrices for ARIMA(1,1,1) model

ARIMA.out

```
Chapter 3  ARIMA Model(with 1st differencing)
=================================================================
 d=1, ar1=.6,  MA1= - .4, sigma_eps^2 = .9
Phi =
        1.0000        1.0000        0.00000
       0.00000       0.60000         1.0000
       0.00000       0.00000        0.00000
        1.0000        1.0000        0.00000
Omega =
       0.00000       0.00000        0.00000       0.00000
       0.00000       0.90000       -0.36000       0.00000
       0.00000      -0.36000        0.14400       0.00000
       0.00000       0.00000        0.00000       0.00000
Sigma =
       -1.0000       0.00000        0.00000
       0.00000       0.95625       -0.36000
       0.00000      -0.36000        0.14400
       0.00000       0.00000        0.00000
```

ARIMA(0,1,1) aka simple exponential smoothing


ARIMA011.ox - C:\Program Files\OxMetrics6\Ox\packages\ssfpack\CKbook\Chapter_3\ARIMA01...

```

main()
{
 println("Chapter 3  ARIMA (0,1,1) Model");
 println("=================================================================="
 println( " d=1,  MA1= - .4, sigma_eps^2 = .9");
 decl mPhi, mOmega, mSigma;
 GetSsfSarima
 (1,<> ,<-0.4>, sqrt(0.9), &mPhi, &mOmega, &mSigma);
 print("Phi = ", mPhi, "Omega = ", mOmega, "Sigma = ",mSigma);
}

```

System matrices for ARIMA(0,1,1) model

ARIMA011.out

```
Chapter 3  ARIMA (0,1,1) Model
==================================================================
 d=1,  MA1= - .4, sigma_eps^2 = .9
Phi =
       1.0000       1.0000      0.00000
      0.00000      0.00000       1.0000
      0.00000      0.00000      0.00000
       1.0000       1.0000      0.00000
Omega =
      0.00000      0.00000      0.00000      0.00000
      0.00000      0.90000     -0.36000      0.00000
      0.00000     -0.36000      0.14400      0.00000
      0.00000      0.00000      0.00000      0.00000
Sigma =
      -1.0000      0.00000      0.00000
      0.00000       1.0440     -0.36000
      0.00000     -0.36000      0.14400
      0.00000      0.00000      0.00000
```

ARIMA(0,2,2) system matrices

*ARIMA011.ox - C:\Program Files\OxMetrics6\Ox\packages\ssfpack\CKbook\Chapter_3\ARIMA0...

```

main()
{
 println("Chapter 3  ARIMA (0,2,2) Model");
 println("=============================================================="
 println( " d=2,  MA1= - .4, MA2 = 0.5,  sigma_eps^2 = .9");
 decl mPhi, mOmega, mSigma;
 GetSsfSarima
 (1,<> ,<-0.4>, sqrt(0.9), &mPhi, &mOmega, &mSigma);
 print("Phi = ", mPhi, "Omega = ", mOmega, "Sigma = ",mSigma);
}
```

ARMA(2,2) system matrices

GetSsfARMA.out

```
                    ARMA(2,2)     Models
 ar1 = 0.6  ar2= -0.3  ma1 = -0.3  ma2=.5      sigma_esp^2=.4
Phi =
      0.60000       1.0000      0.00000
     -0.30000      0.00000       1.0000
      0.00000      0.00000      0.00000
       1.0000      0.00000      0.00000
 Omega =
      0.40000     -0.12000      0.20000      0.00000
     -0.12000     0.036000    -0.060000      0.00000
      0.20000    -0.060000      0.10000      0.00000
      0.00000      0.00000      0.00000      0.00000
 Sigma =
      0.50354     -0.11588      0.20000
     -0.11588     0.061319    -0.060000
      0.20000    -0.060000      0.10000
      0.00000      0.00000      0.00000
```

System matrices for ARIMA(0,2,2) model

ARIMA022.out

```
Ox Professional version 6.00 (Windows/U/MT) (C) J.A. Doornik, 1994-2009
Chapter 3  ARIMA (0,2,2) Model
======================================================================
 d=2,  MA1= - .4, MA2 = 0.5,  sigma_eps^2 = .9
Phi =
       1.0000       1.0000       1.0000      0.00000      0.00000
      0.00000       1.0000       1.0000      0.00000      0.00000
      0.00000      0.00000      0.00000       1.0000      0.00000
      0.00000      0.00000      0.00000      0.00000       1.0000
      0.00000      0.00000      0.00000      0.00000      0.00000
       1.0000       1.0000       1.0000      0.00000      0.00000
Omega =
      0.00000      0.00000      0.00000      0.00000      0.00000      0.00000
      0.00000      0.00000      0.00000      0.00000      0.00000      0.00000
      0.00000      0.00000      0.90000     -0.36000      0.45000      0.00000
      0.00000      0.00000     -0.36000      0.14400     -0.18000      0.00000
      0.00000      0.00000      0.45000     -0.18000      0.22500      0.00000
      0.00000      0.00000      0.00000      0.00000      0.00000      0.00000
Sigma =
      -1.0000      0.00000      0.00000      0.00000      0.00000
      0.00000      -1.0000      0.00000      0.00000      0.00000
      0.00000      0.00000       1.2690     -0.54000      0.45000
      0.00000      0.00000     -0.54000      0.36900     -0.18000
      0.00000      0.00000      0.45000     -0.18000      0.22500
      0.00000      0.00000      0.00000      0.00000      0.00000
```

Structural time series model (Stsm)

Koopman et al.(2008)

$$\textit{The state space form of the structural time series model is}$$

$$y_t = \mu_t + \beta_t + \gamma_t + \psi_t$$

where

$$\mu_t = \textit{unobserved trend (level) component}$$

$$\beta_t = \textit{unobserved slope component}$$

$$\gamma_t = \textit{unobserved seasonal component}$$

$$\psi_t = \textit{unobserved cyclical component}$$

$$\xi_t = \textit{unobserved irregular component}$$

Structural time series models

Koopman, S.J., Shephard, N. and Doornik, J. (2008)

$$y_t = \mu_t + \gamma_t + \psi_t + \xi_t \qquad \xi_t \sim NID(\mathbf{0}, \sigma_\xi^2)$$

where

$$y_t = \text{response variable}$$

$$\mu_t = \text{local level}$$

$$\beta_t = \text{unobserved slope component}$$

$$\gamma_t = \text{seasonality}$$

$$\psi_t = \text{cycle}$$

$$\xi_t = \text{measurement error}$$

Nonstationary trend component

When the trend contains drift or deterministic slope, it is not stationary.

Hence, the slope component is added to the trend in order to handle

Such nonstationarity. All of these components are random effects.

They are characterized by their own measurement error, sampling

error, or other error in variables. Consequently, each unobserved

component has its own error term. (Zivot and Wang, 530.)

Is the level fixed or random

Usually, the level will be time-varying and possess an evolutionary error ,eta , (η_t). Moreover, unless measured without error, the measurement error , epsilon(ε_t) will be nonzero as well. Hence, their standard deviations, apparent in the sigma matrix, will also be nonzero.

If there is no error of measurement, then the epsilon would be fixed at zero. In the measurement equation. The error term for the transition (evolutionary) process then can be set to zero by equating eta to 0. and letting its standard deviation in the sigma matrix = 0 as well. This can be done in the transition equation while any representation of that variation in the sigma matrix can be set to zero.

The local linear trend model

Contains drift or stochastic trend (random walk) (error allowed to vary)
or
Contains deterministic trend (error=0) sometimes called smooth trend

$$\mu_{t+1} = \mu_t + \beta_t + \eta_t \qquad \eta_t : GWN(\mathbf{0}, \sigma_\eta^2)$$

$$\beta_{t+1} = \beta_t + \zeta_t \qquad \zeta_t \sim GWN(\mathbf{0}, \sigma_\zeta^2)$$

when the error = 0, these trends are smooth and fixed(deterministic), but when the errors and error variances are nonzero, these components are random (having either measurement, sampling, or some other kind of error.

Initial values of trend

$$\mu_1 = N(\mathbf{0}, \sigma_\eta^2)$$

$$\beta_1 = N(\mathbf{0}, \sigma_\varsigma^2)$$

when the error variance- $> \infty$*,*

this indicates reduction in the precision of

the prior parameter to 0, so the data receieve almost

give all of the weight in the sequential weighted averaging

process that predicts the posterior predictive

mean or variance. On the computer, a very large

number, such as 10^6 *or* 10^7 *replaces the infinite*

variance. After a few more iterations, the process usually

converges to a solution.

This is a local trend

The trend is a local rather than global trend.   The trend is allowed to varying over time.

It can be time varying or fixed,   depending upon whether the errors positive or equal to zero.

Trends are evident in changes in the level and/or slope, sometimes apparent in a graph of the series.

Identifying the nature of the trend

When we test the signal to noise ratio of the trend and find that it is zero,

We infer that the trend is not stochastic but fixed (deterministic).

There are also higher order trends such as u m-1, u m-2, … that can be interpreted as first, second, or higher order (m) derivatives.

$$\mu_{i,t+1} = \mu_{i,t} + \mu_{i-1,t}$$

$$where$$

$$i = 2, 3, \ldots, m$$

Local Linear Trend model with Ox (Koopman, Shephard, and Doornik, 2008, 8)

```
#include <oxstd.h>
#include <c:\Program Files\OxMetrics51\Ox\packages\ssfpack\ssfpack_ex.h>

//   Local Linear Trend Model


main()
{
   decl mPhi, mSigma, mOmega;
   GetSsfStsm
   (<CMP_IRREG, 1.0, 0,0,0;
     CMP_LEVEL, .5,0,0,0;
     CMP_SLOPE, .1,0,0,0>,


   &mPhi, &mOmega, &mSigma);
   format ("%#6.2g");
   println("Local Linear Trend Model ");

   println("                              ");
   print("Phi = ",mPhi, " Omega = ",mOmega, " Sigma = ",mSigma);
```

Local Linear Trend
Model system matrices

```
--------------- Ox at 21:32:37 on 09-Nov-2009 ---------------

Ox Professional version 6.00 (Windows/U/MT) (C) J.A. Doornik, 1994-2009
Local Linear Trend Model

Phi =
   1.0    1.0
  0.00    1.0
   1.0   0.00
 Omega =
  0.25   0.00   0.00
  0.00  0.010   0.00
  0.00   0.00    1.0
 Sigma =
  -1.0   0.00
  0.00   -1.0
  0.00   0.00
```

Seasonal component

Seasonality, like all unobserved components, can be stochastic (random) or fixed or nonexistent.

Seasonality, an annual variation, may render a series nonstationary and difficult to use for forecasting.

Seasonality may be defined by dummy variables or trigonometric functions.

Defining seasonality

$$\gamma_{1,t+1} = -\gamma_{1,t} - \gamma_{2,t} - \gamma_{3,t} + \omega_t \qquad \omega_t \sim NID(0, \sigma_\omega^2)$$

Another formulation is

$$\gamma(L) = 1 + L + L^2 + \mathrm{L} \quad + L^{s-1} + \omega_t$$

where

$s = seasonal\ \ periodicity$

$\omega_t = random\ \ error\ of\ \ seasonal\ \ component$

If omega is non-zero,   the series is random (stochastic).

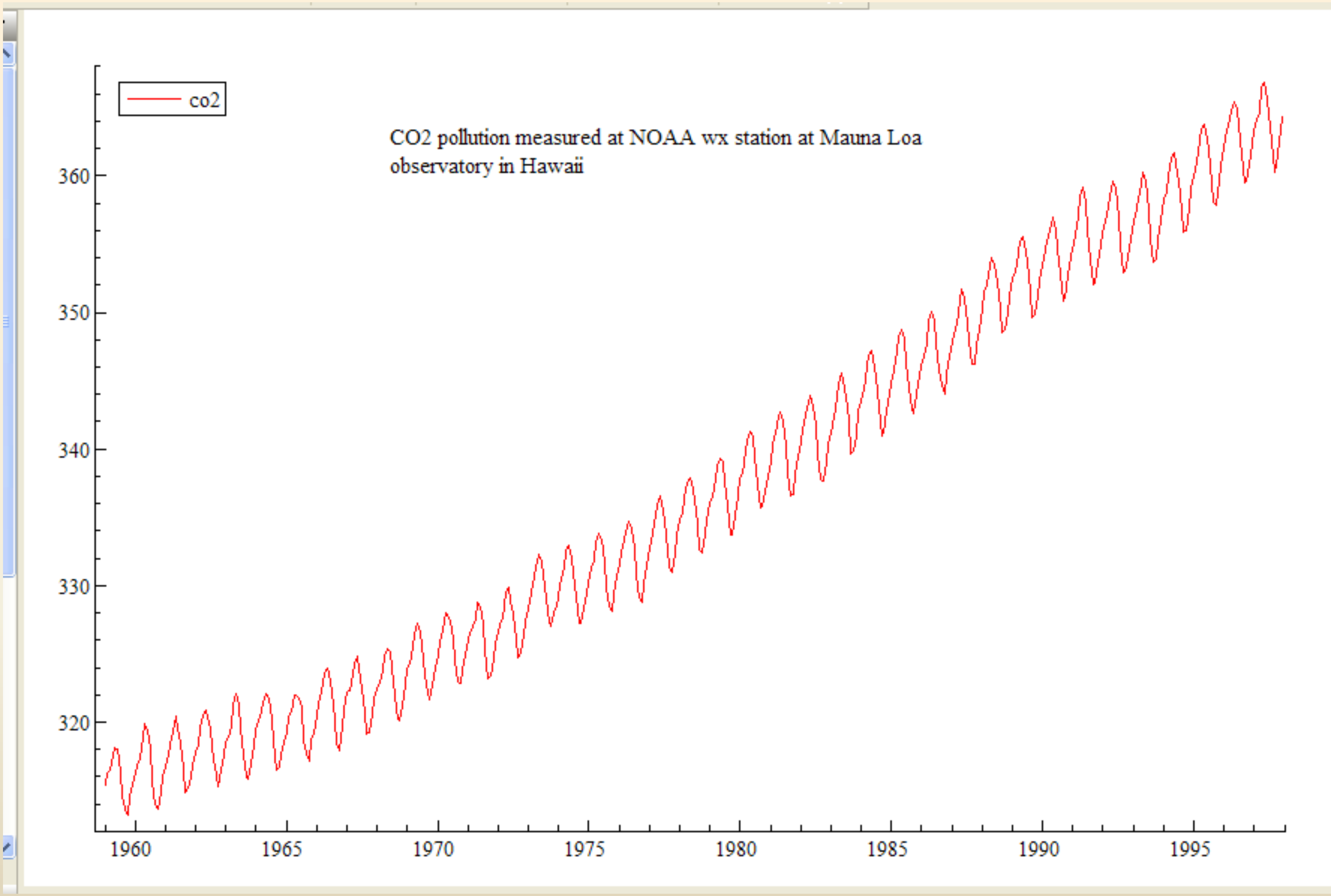
If omega is zero, the series can be seasonal yet have a fixed seasonality.

Identifying and assessing seasonality

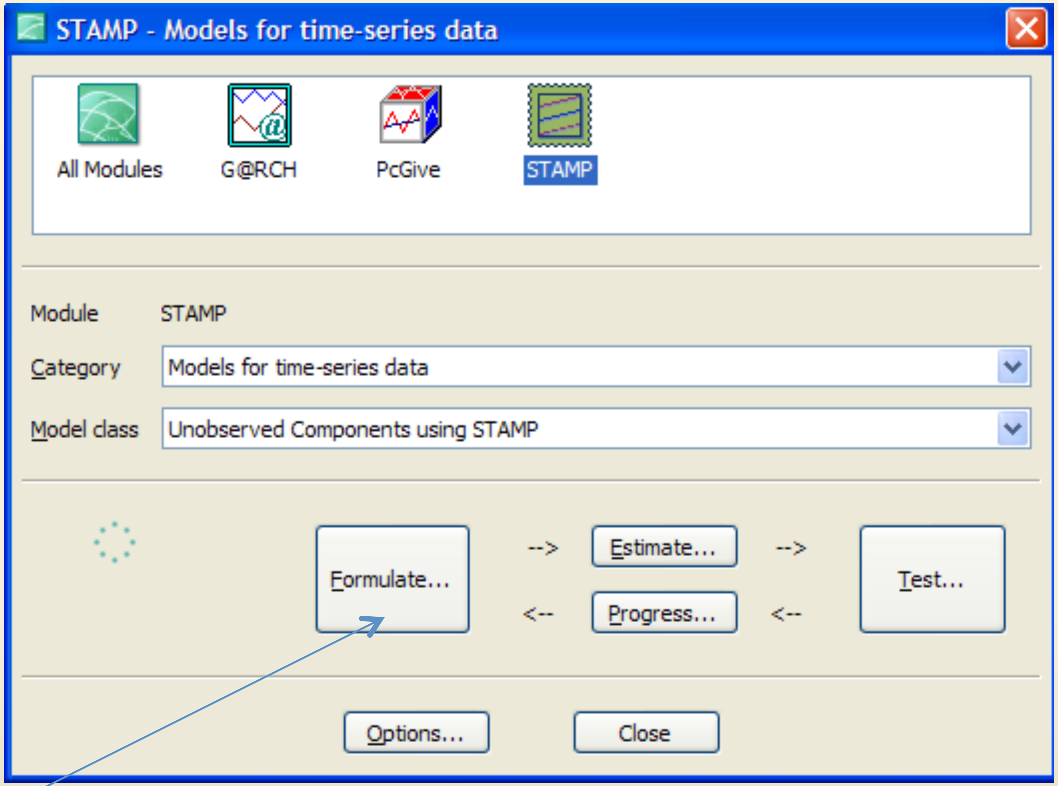
Is it fixed or random? Is it continuous or discrete? Should we select
Dummy variables or trigonometric variables to represent the seasonality?

All these questions need to be answered for us to decide how to define the variable.

Koopman et al. generally suggest beginning with a stochastic model
And looking at the signal to noise ratio (q) for evidence of a random effect. If the coefficient = 0 , it may be fixed or non-existent. We try it as fixed and test for the model fit. We select the better fit.

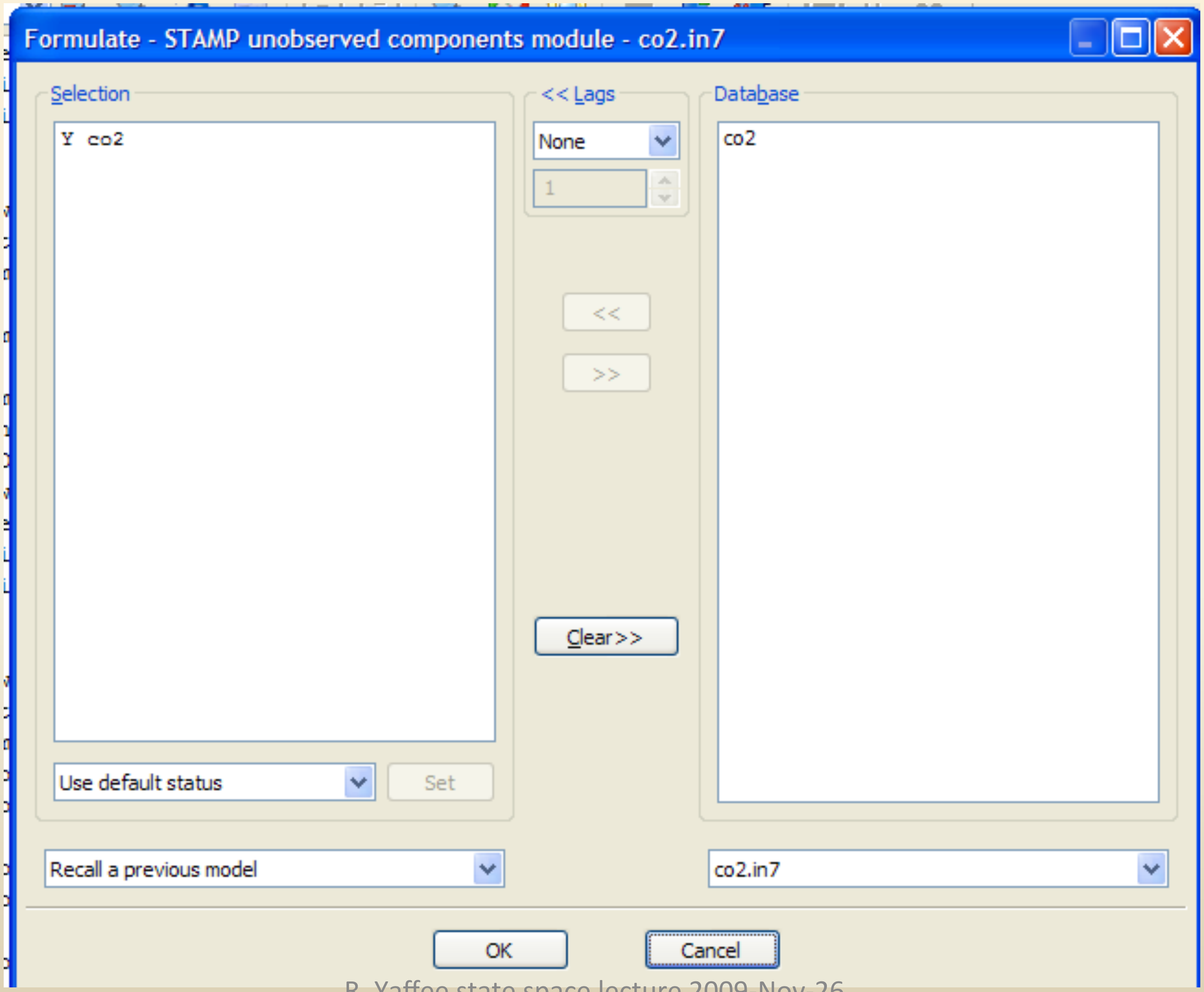

co2
CO2 pollution measured at NOAA wx station at Mauna Loa observatory in Hawaii
360
350
340
330
320
1960
1965
1970
1975
1980
1985
1990
1995

Co2 measurement at NOAA observatory at Mauna Loa



Click on formulate

Move the dependent variable over into the selection box

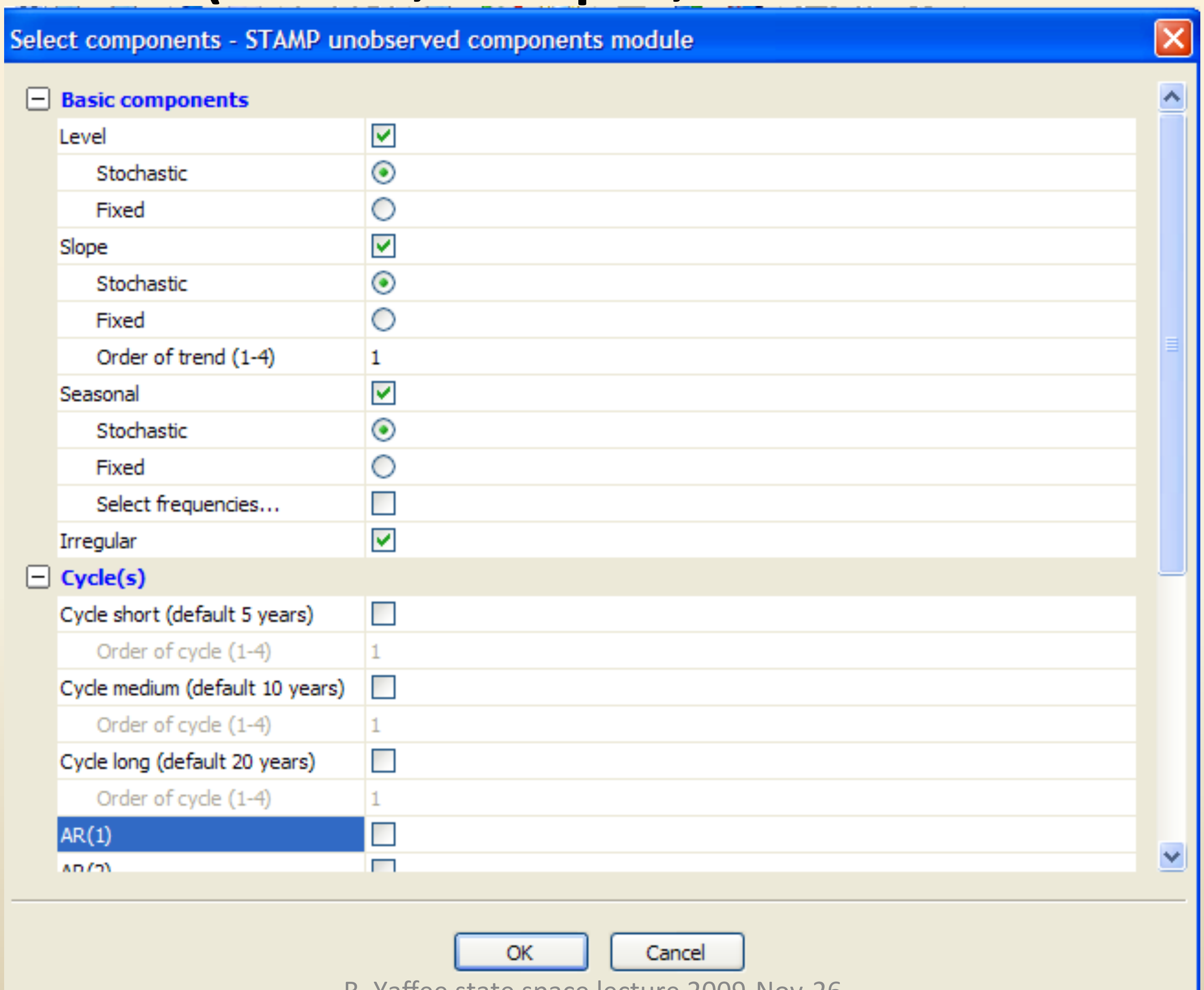


Then click
On OK.

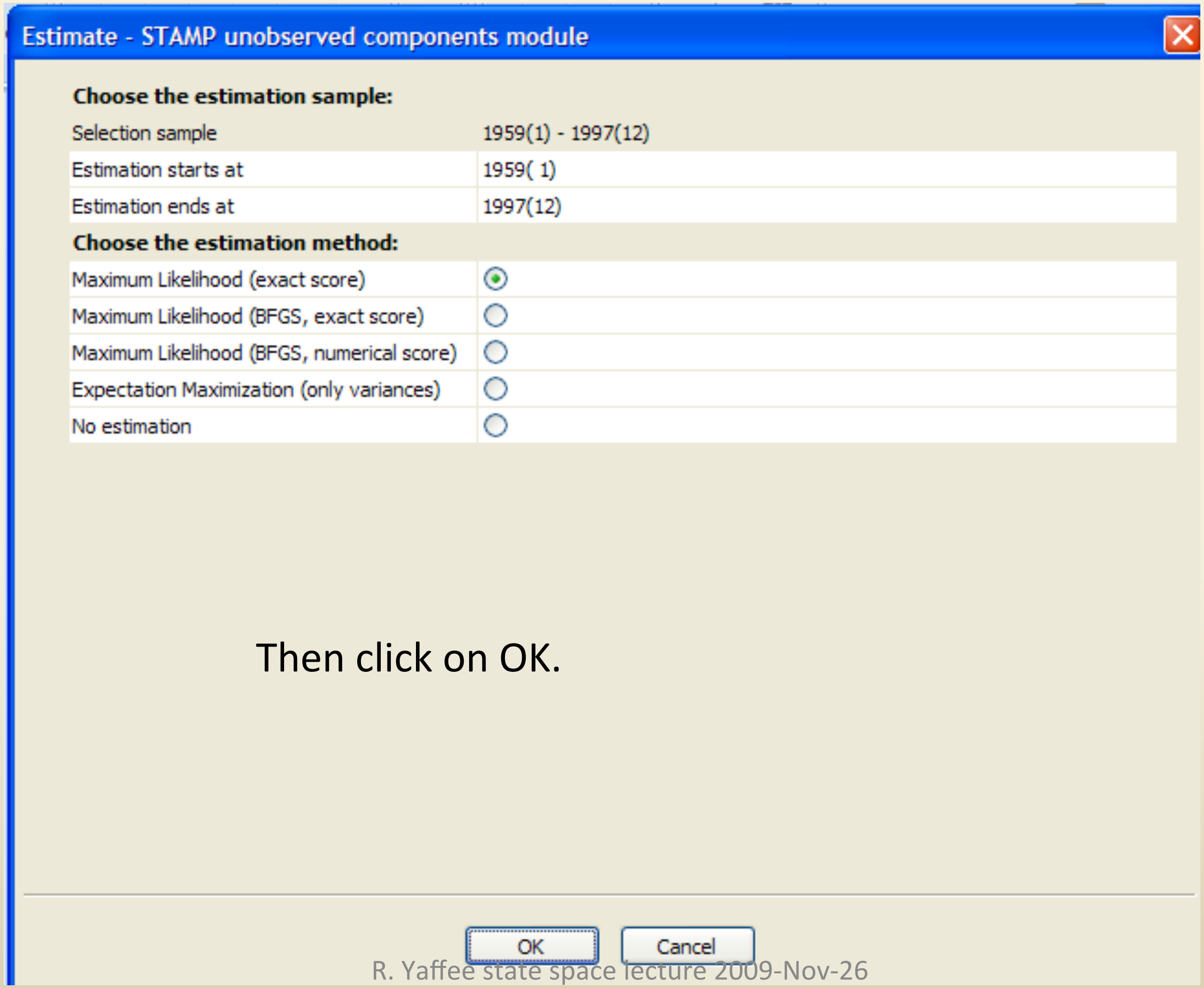
Begin by testing a basic structural model (level, slope, and seasonal)

Allow the Stochastic Options to be checked at the first pass.

Then click On OK.



Leave the default estimation checked and first test on the full sample



Then click on OK.

Examine the errors. Each component reveals a nonzero error variance.

The model converged as is indicated by the steady state having been found.

We do observe some normality of the residuals.

```
UC( 6) Estimation done by Maximum Likelihood (exact score)
    The database used is C:\Program Files\OxMetrics6\data\co2.in7
    The selection sample is: 1959(1) - 1997(12) (T = 468, N = 1)
    The dependent variable Y is: co2
    The model is:  Y = Trend + Seasonal + Irregular
    Steady state. found

Log-Likelihood is 537.692 (-2 LogL = -1075.38).
Prediction error variance is 0.0834561

Summary statistics
                        co2
 T                   468.00
 p                   3.0000
 std.error          0.28889
 Normality           1.3316
 H(151)             0.96985
 DW                  1.8683
 r(1)              0.055094
 q                   24.000
 r(q)             -0.056483
 Q(q,q-p)            30.012
 Rs^2              0.085267

 Variances of disturbances:
                    Value    (q-ratio)
Level           0.0285623  (    1.000)
Slope        4.44185e-006  (0.0001555)
Seasonal     2.48387e-005  (0.0008696)
Irregular       0.0254314  (   0.8904)
```

This means that all components are random and have an error term.

All components of the state vector are significant. We retain all of them.

State vector analysis at period 1997(12)

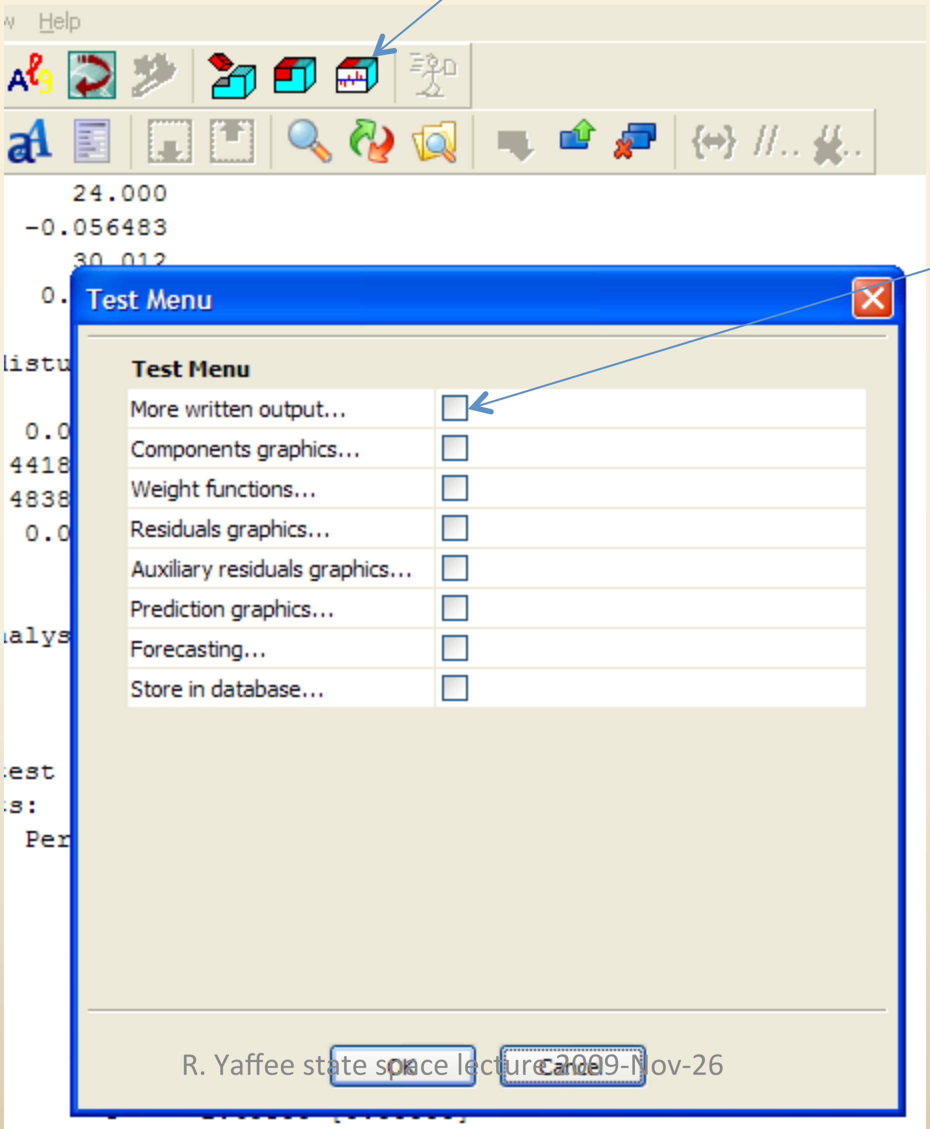
	Value	Prob
Level	364.97931	[0.00000]
Slope	0.12858	[0.00000]
Seasonal chi2 test	3837.23560	[0.00000]

Seasonal effects:

Period	Value	Prob
1	0.02164	[0.82882]
2	0.75558	[0.00000]
3	1.45031	[0.00000]
4	2.71683	[0.00000]
5	3.15913	[0.00000]
6	2.35544	[0.00000]
7	0.76412	[0.00000]
8	-1.40800	[0.00000]
9	-3.43157	[0.00000]
10	-3.40383	[0.00000]
11	-2.13673	[0.00000]
12	-0.84294	[0.00000]

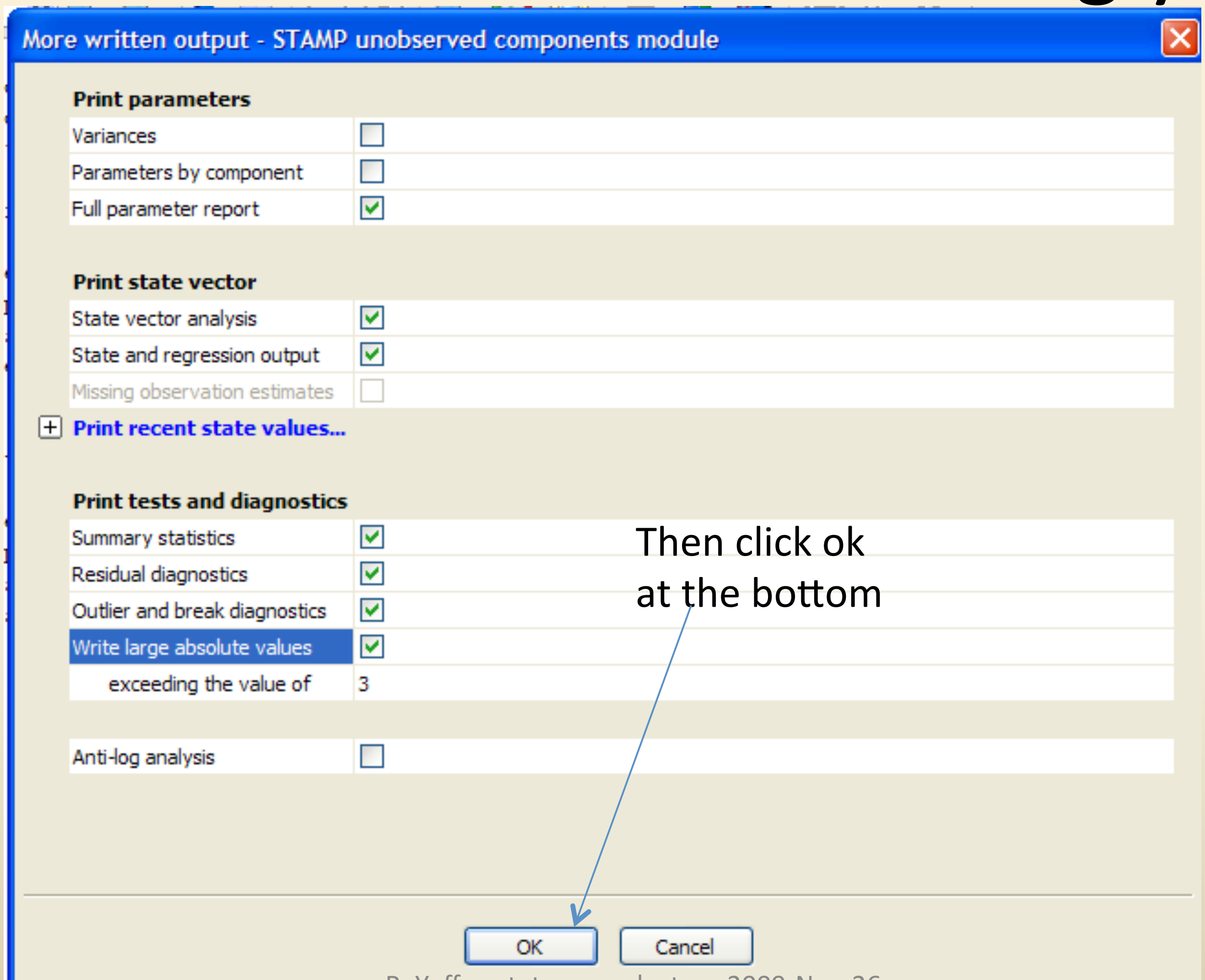
We then proceed.

We click on the test icon to obtain a test dialog box



We click on more test output and then ok at the bottom

Another dialog box appear and we check the boxes accordingly



The full parameter report shows actual and transformed stochastic parameter

```
 Full parameter report
 Actual parameters (all)
                    Value
Var Level        0.028562
Var Slope      4.4419e-006
Var Seasonal   2.4839e-005
Var Irregular    0.025431
 Transformed parameters (not fixed)
                Transform       1stDer         2ndDer     asymp.s.e
Var Level         -1.7778   7.9936e-010      -0.45820      0.091672
Var Slope         -6.1622   3.5527e-010    -0.0077765       0.52466
Var Seasonal      -5.3016  -9.7700e-010     -0.066040       0.18925
Var Irregular     -1.8359  -1.0436e-009      -0.55919      0.081027
 Actual parameters (not fixed) with 68% asymmetric confidence interval
                    Value     leftbound    rightbound
Var Level        0.028562      0.023778      0.034310
Var Slope      4.4419e-006   1.5554e-006   1.2685e-005
Var Seasonal   2.4839e-005   1.7012e-005   3.6267e-005
Var Irregular    0.025431      0.021627      0.029905
```

We observe that all derivatives were successfully computed. Then we look below

We note that all components are significant and look below

State vector analysis at period 1997(12)

		Value	Prob
Level		364.97931	[0.00000]
Slope		0.12858	[0.00000]
Seasonal chi2 test		3837.23560	[0.00000]
Seasonal effects:			
	Period	Value	Prob
	1	0.02164	[0.82882]
	2	0.75558	[0.00000]
	3	1.45031	[0.00000]
	4	2.71683	[0.00000]
	5	3.15913	[0.00000]
	6	2.35544	[0.00000]
	7	0.76412	[0.00000]
	8	-1.40800	[0.00000]
	9	-3.43157	[0.00000]
	10	-3.40383	[0.00000]
	11	-2.13673	[0.00000]
	12	-0.84294	[0.00000]

We observe the coefficients for the components and note their sign, magnitude and significance.

State vector at period 1997(12)

	Coefficient	RMSE	t-value	Prob
Level	364.97931	0.14260	2559.42414	[0.00000]
Slope	0.12858	0.01902	6.76195	[0.00000]
Seasonal	-1.73438	0.05053	-34.32131	[0.00000]
Seasonal 2	2.38882	0.05105	46.79554	[0.00000]
Seasonal 3	0.84459	0.04038	20.91501	[0.00000]
Seasonal 4	-0.02917	0.04106	-0.71032	[0.47787]
Seasonal 5	0.12645	0.03774	3.35040	[0.00087]
Seasonal 6	-0.05475	0.03800	-1.44095	[0.15029]
Seasonal 7	-0.11719	0.03706	-3.16209	[0.00167]
Seasonal 8	-0.03915	0.03664	-1.06832	[0.28594]
Seasonal 9	0.00874	0.03697	0.23637	[0.81325]
Seasonal10	-0.00263	0.03601	-0.07309	[0.94176]
Seasonal11	0.02885	0.03082	0.93594	[0.34980]

We begin to diagnose the model

We look for violation of the assumptions of normality, independence of observations, and white noise residuals.

The residuals appear to be normally distributed but there is evidence of spurious correlation and consequent bias in our estimates upward.

```
Normality test for Residuals co2
                            Value
 Sample size               455.00
Mean                     0.065963
St.Dev                    0.99782
Skewness                -0.077487
Excess kurtosis          -0.23697
Minimum                   -2.7429
Maximum                    2.5714

                          Chi^2        prob
Skewness                 0.45532   [ 0.4998]
Kurtosis                  1.0646   [ 0.3022]
Bowman-Shenton            1.5199   [ 0.4677]

Goodness-of-fit based on Residuals co2
                                                       Value
Prediction error variance (p.e.v)                   0.083456
Prediction error mean deviation (m.d)               0.067494
Ratio p.e.v. / m.d in squares                        0.97333
Coefficient of determination R^2                     0.99964
... based on differences Rd^2                        0.94402
... based on diff around seas mean Rs^2             0.085267
Information criterion Akaike (AIC)                   -2.4236
... Bayesian Schwartz (BIC)                          -2.2995

Serial correlation statistics for Residuals co2
 Durbin-Watson test is 1.86825
 Asymptotic deviation for correlation is 0.0468807
   Lag         df        Ser.Corr       BoxLjung         prob
     4          1       -0.054437          6.298    [ 0.0121]
     5          2       -0.039249         7.0098    [ 0.0300]
     6          3       -0.081798         10.108    [ 0.0177]
     7          4        -0.01865          10.27    [ 0.0361]
     8          5       -0.068798         12.472    [ 0.0289]
```

We examine the goodness of fit test and find the R^2 to be too high

Autocorrelation
In the residuals
And unmodeled
outliers seem to
be evident.

The model fit
could be
improved by
adding an ar(1)
Component and
modeling the
outliers.

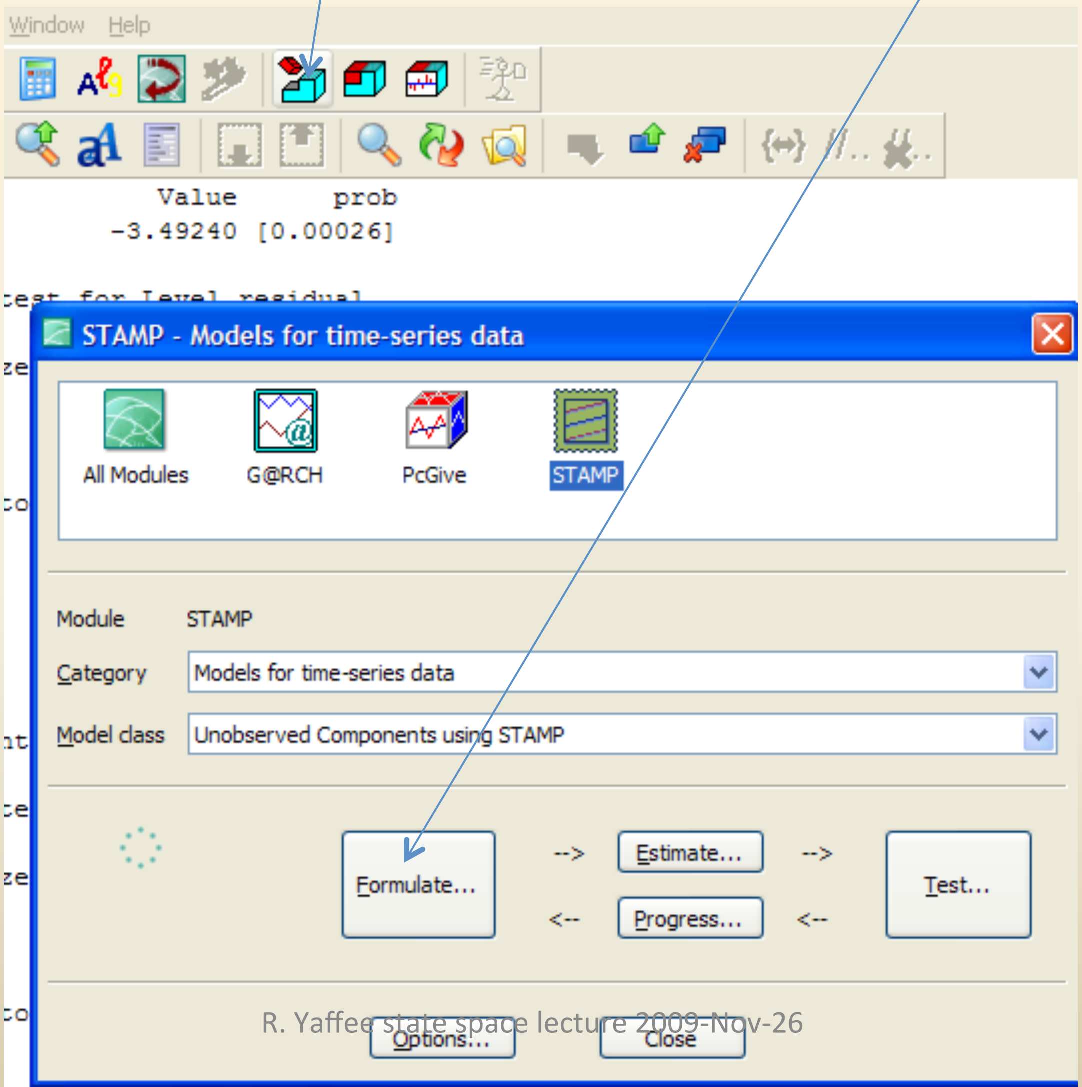
```
Goodness-of-fit based on Residuals co2
                                                     Value
Prediction error variance (p.e.v)                 0.083456
Prediction error mean deviation (m.d)             0.067494
Ratio p.e.v. / m.d in squares                      0.97333
Coefficient of determination R^2                   0.99964
... based on differences Rd^2                      0.94402
... based on diff around seas mean Rs^2           0.085267
Information criterion Akaike (AIC)                 -2.4236
... Bayesian Schwartz (BIC)                        -2.2995

Serial correlation statistics for Residuals co2
 Durbin-Watson test is 1.86825
 Asymptotic deviation for correlation is 0.0468807
   Lag        df       Ser.Corr     BoxLjung         prob
     4         1     -0.054437         6.298  [ 0.0121]
     5         2     -0.039249        7.0098  [ 0.0300]
     6         3     -0.081798        10.108  [ 0.0177]
     7         4      -0.01865         10.27  [ 0.0361]
     8         5     -0.068798        12.472  [ 0.0289]
    12         9      0.029252        18.643  [ 0.0284]
    24        21     -0.056483        30.012  [ 0.0918]
    36        33   -0.00081262        49.121  [ 0.0352]

Values larger than 3 for Irregular residual:
                   Value      prob
1971(4)          -3.30415 [0.00051]
1972(3)          -3.11111 [0.00099]
1986(9)           3.10229 [0.00102]
```

There remain problems in the level and slope residuals as well

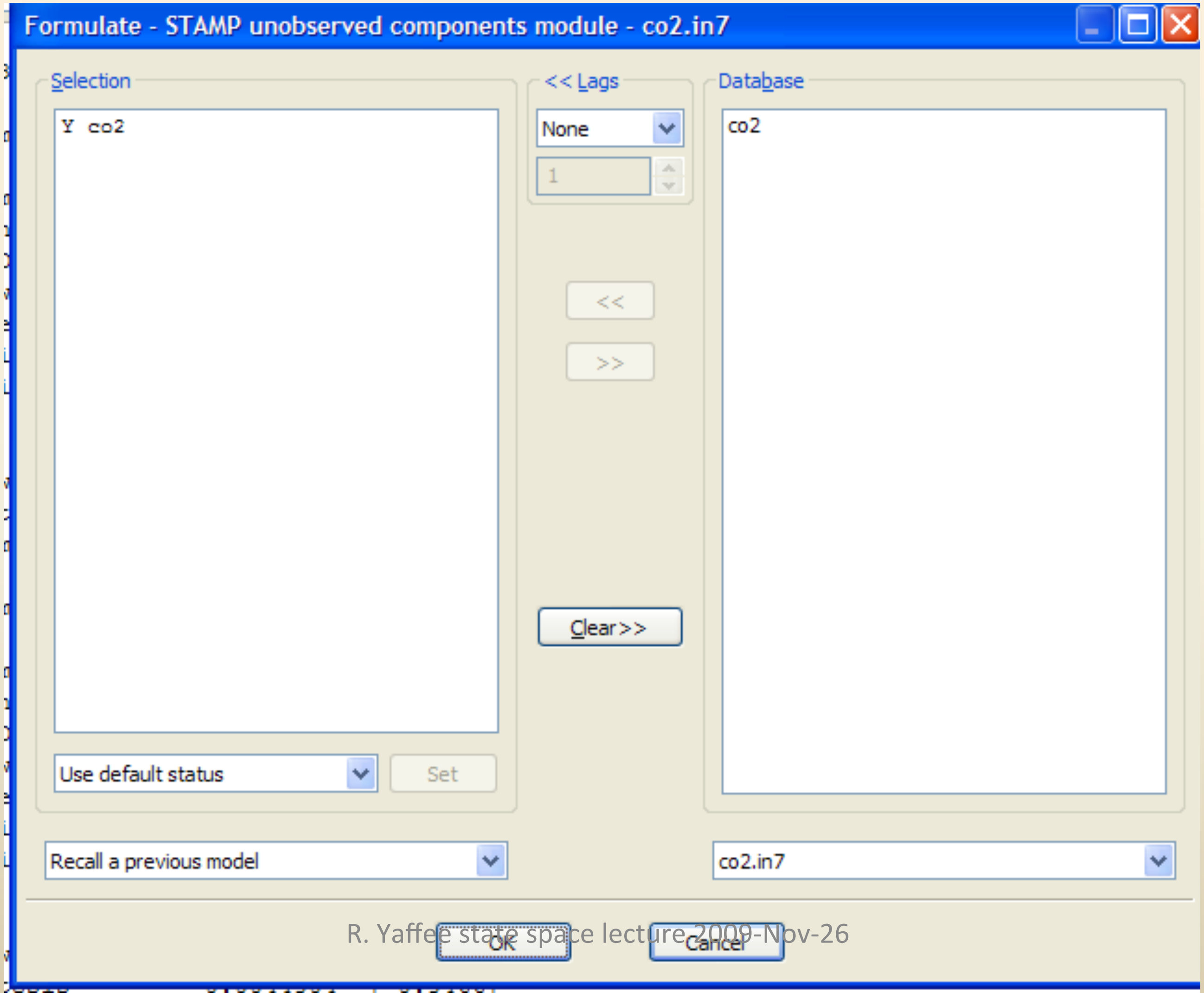
```
Values larger than 3 for Level residual:
                     Value      prob
1973(12)          -3.49240 [0.00026]

Normality test for Level residual
                       Value
 Sample size          468.00
Mean              0.00032373
St.Dev                1.0007
Skewness           -0.097486
Excess kurtosis     0.040163
Minimum              -3.4924
Maximum               2.4945

                     Chi^2        prob
Skewness           0.74127   [ 0.3893]
Kurtosis          0.031455   [ 0.8592]
Bowman-Shenton     0.77272   [ 0.6795]

Normality test for Slope residual
                       Value
 Sample size          467.00
Mean                 0.76903
St.Dev               0.62943
Skewness            -0.28796
Excess kurtosis    -0.015191
Minimum             -0.78047
Maximum               2.9626

                     Chi^2        prob
Skewness            6.4541   [ 0.0111]
Kurtosis         0.0044904   [ 0.9466]
Bowman-Shenton      6.4586   [ 0.0396]
```

There is an unmodeled level residual and the slope residuals are not quite normal.

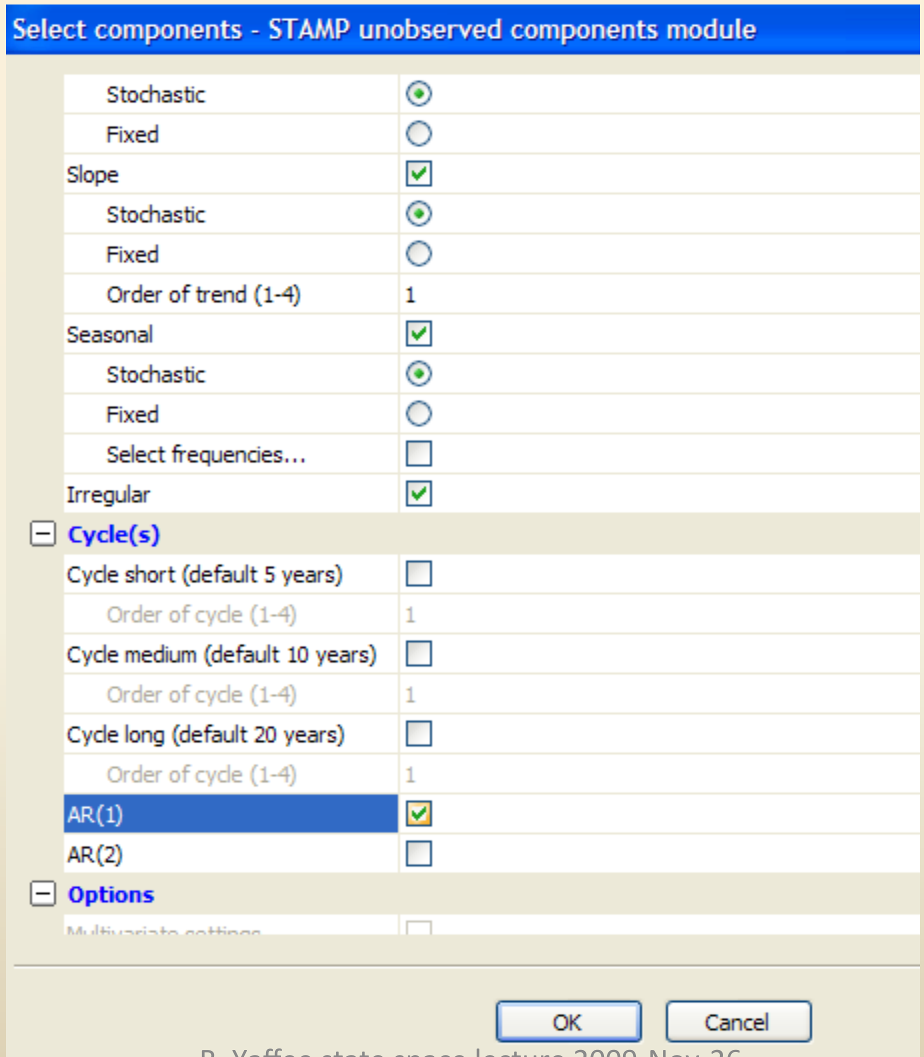
Click on the model icon and then on formulate.



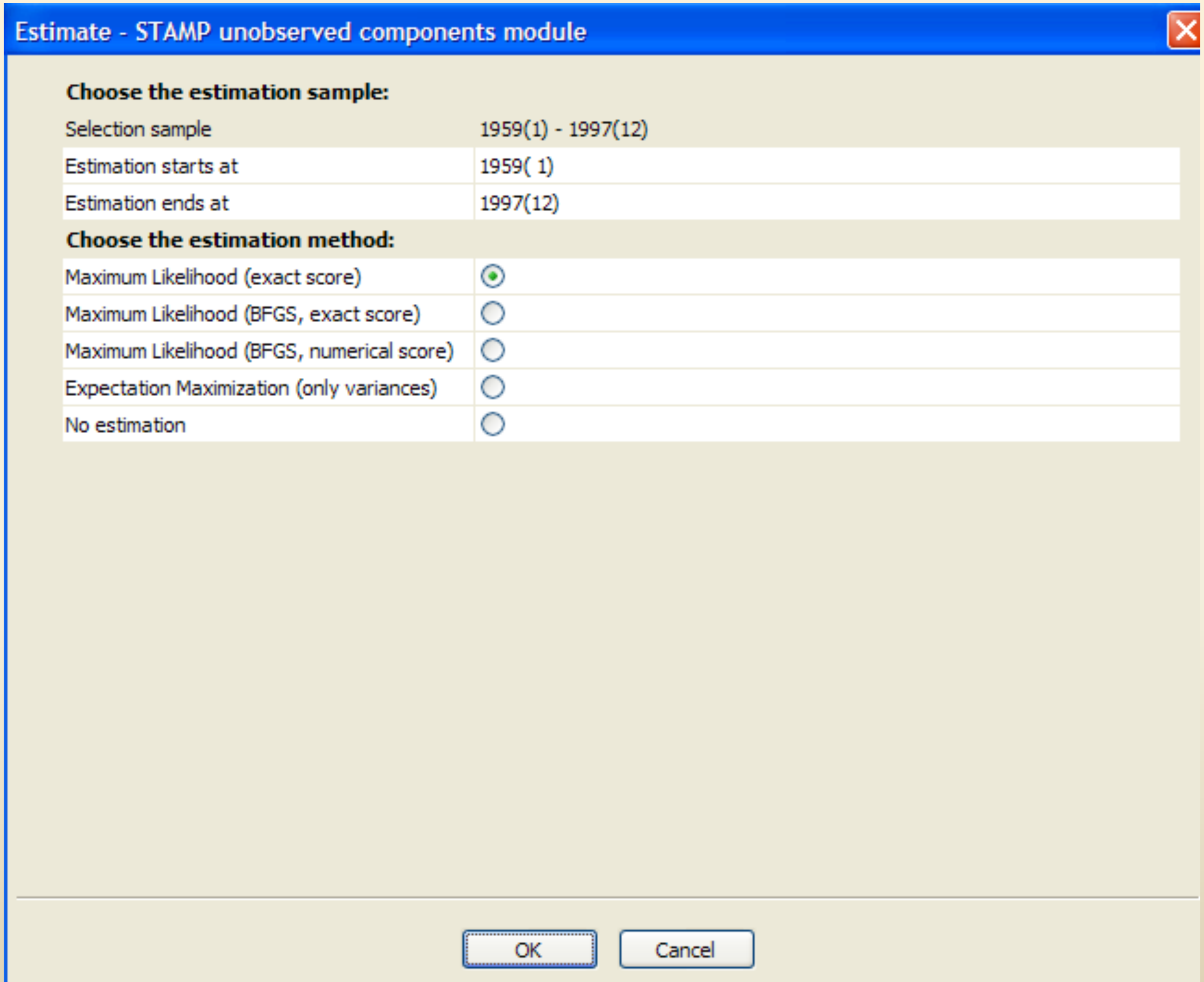
Click on ok again



In the Select components box, click on Cycles(s), on ar(1), and then Ok.



When the estimate box appears, leave the defaults checked and click ok.



All components remain significant—including the AR(1) component

```
UC( 7) Estimation done by Maximum Likelihood (exact score)
    The database used is C:\Program Files\OxMetrics6\data\co2.in7
    The selection sample is: 1959(1) - 1997(12) (T = 468, N = 1)
    The dependent variable Y is: co2
    The model is:  Y = Trend + Seasonal + Irregular + AR(1)
    Steady state. found

Log-Likelihood is 539.403 (-2 LogL = -1078.81).
Prediction error variance is 0.0827491

Summary statistics
                      co2
 T                 468.00
 p                 5.0000
 std.error        0.28766
 Normality         1.1240
 H(151)            1.0028
 DW                1.9210
 r(1)            0.029069
 q                 25.000
 r(q)            0.076226
 Q(q,q-p)          29.432
 Rs^2            0.093016

 Variances of disturbances:
                    Value    (q-ratio)
Level           0.0190756  (   0.6216)
Slope         4.81875e-006 (0.0001570)
Seasonal      2.24881e-005 (0.0007328)
AR(1)           0.0306863  (    1.000)
Irregular       0.0165861  (   0.5405)

AR(1) other parameters:
AR coefficient     0.57117
```

All state vector components appear significant as we scroll down

Variances of disturbances:

	Value	(q-ratio)
Level	0.0190756	(0.6216)
Slope	4.81875e-006	(0.0001570)
Seasonal	2.24881e-005	(0.0007328)
AR(1)	0.0306863	(1.000)
Irregular	0.0165861	(0.5405)

AR(1) other parameters:

AR coefficient 0.57117

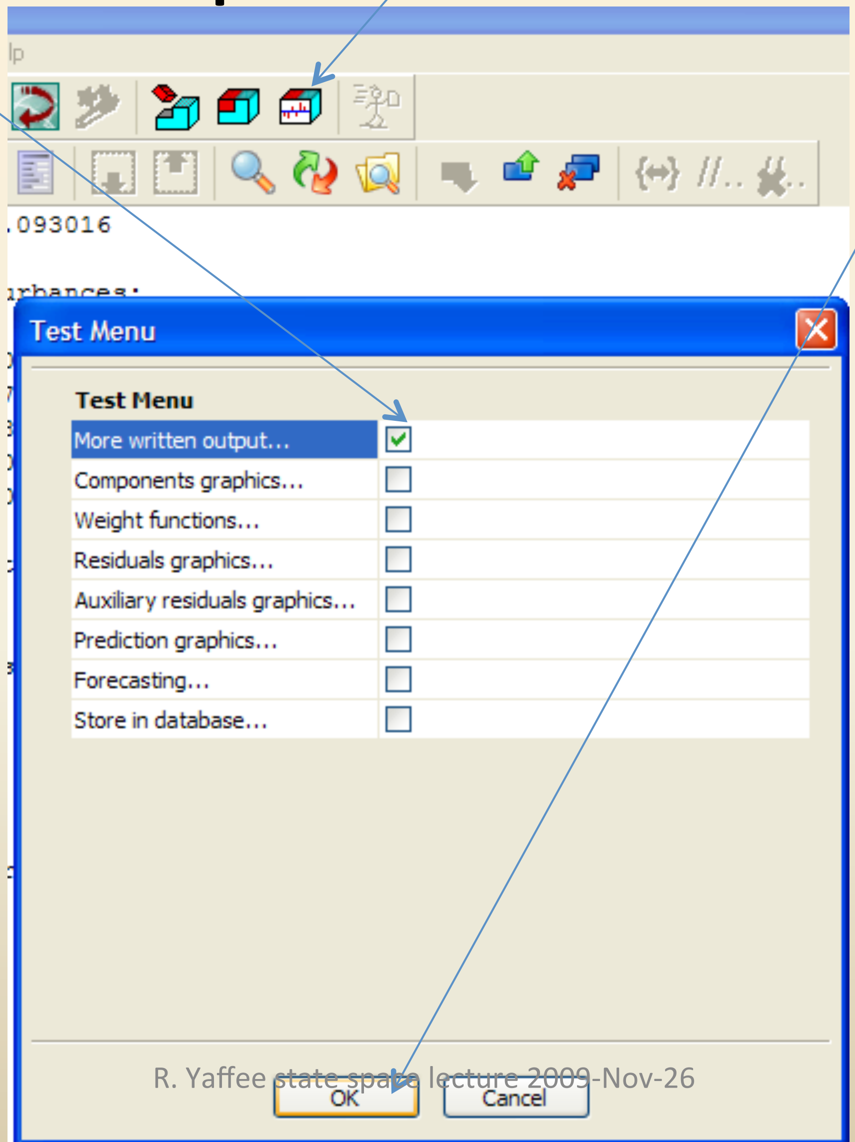
State vector analysis at period 1997(12)

	Value	Prob
Level	364.86769	[0.00000]
Slope	0.12907	[0.00000]
Seasonal chi2 test	3759.50663	[0.00000]

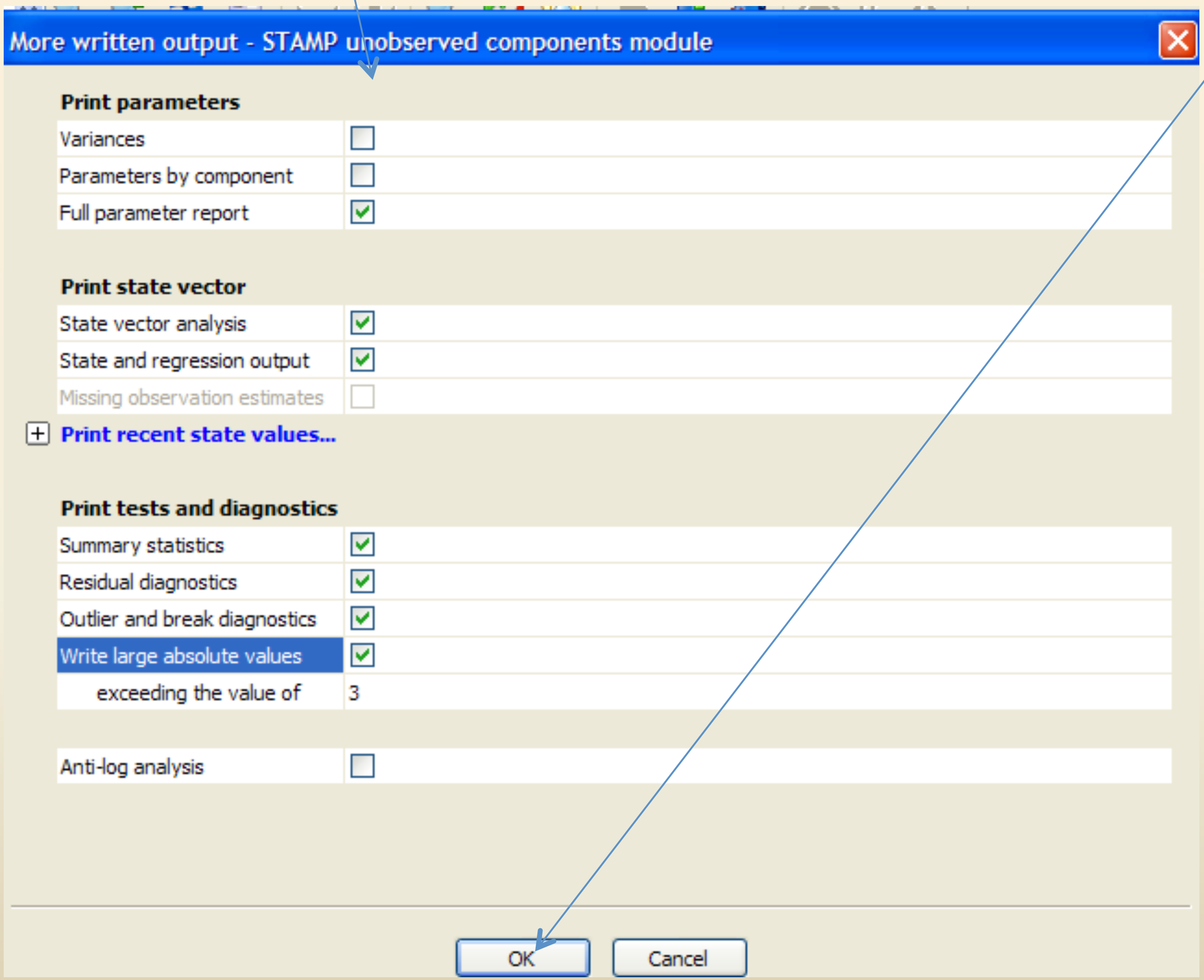
Seasonal effects:

Period	Value	Prob
1	0.01539	[0.87580]
2	0.74286	[0.00000]
3	1.43756	[0.00000]
4	2.70289	[0.00000]
5	3.14948	[0.00000]
6	2.35294	[0.00000]
7	0.76663	[0.00000]
8	-1.39481	[0.00000]
9	-3.40881	[0.00000]
10	-3.38622	[0.00000]
11	-2.13111	[0.00000]
12	-0.84677	[0.00000]

We click on the test icon and the more written output box and then OK.



In the More Written output, we check the boxes below and then click on ok.



Observe a decline in the BIC, a high R^2 but no more residual autocorrelation.

Our model, however, is not Yet optimized because we still have unmodeled outliers.

We will begin to model those next.

```
Goodness-of-fit based on Residuals co2
                                                    Value
Prediction error variance (p.e.v)                0.082749
Prediction error mean deviation (m.d)            0.066934
Ratio p.e.v. / m.d in squares                     0.97299
Coefficient of determination R^2                  0.99964
... based on differences Rd^2                     0.94449
... based on diff around seas mean Rs^2          0.093016
Information criterion Akaike (AIC)                -2.4321
... Bayesian Schwartz (BIC)                        -2.308

Serial correlation statistics for Residuals co2
 Durbin-Watson test is 1.92102
 Asymptotic deviation for correlation is 0.0468807
   Lag        df       Ser.Corr     BoxLjung        prob
     5         1      -0.016078       2.6012   [ 0.1068]
     6         2      -0.069725       4.8526   [ 0.0884]
     7         3     -0.0061134       4.8699   [ 0.1816]
     8         4      -0.070771       7.1998   [ 0.1257]
     9         5        0.10077       11.934   [ 0.0357]
    12         8       0.035624       13.613   [ 0.0924]
    24        20      -0.060105       26.622   [ 0.1462]
    36        32 -4.4464e-005        45.76   [ 0.0546]

Values larger than 3 for Irregular residual:
                    Value       prob
1971(4)          -3.19234 [0.00075]
1972(3)          -3.24716 [0.00062]
1986(9)           3.14235 [0.00089]
```

We examine the residuals of the other components too.

Normality is no longer a problem for irregular or level residuals, although both components have unmodeled outliers.

```
Normality test for Irregular residual
                              Value
 Sample size                 468.00
Mean                     0.00094326
St.Dev                       1.0026
Skewness                   0.031373
Excess kurtosis             0.12461
Minimum                     -3.2472
Maximum                      3.1423

                             Chi^2         prob
Skewness                  0.076775   [ 0.7817]
Kurtosis                    0.3028   [ 0.5821]
Bowman-Shenton             0.37958   [ 0.8271]

Values larger than 3 for Level residual:
                            Value        prob
1973(12)                 -3.38167  [0.00039]

Normality test for Level residual
                              Value
 Sample size                 467.00
Mean                     0.00043138
St.Dev                       1.0016
Skewness                  -0.034242
Excess kurtosis           -0.018064
Minimum                     -3.3817
Maximum                      2.6242

                             Chi^2         prob
Skewness                   0.09126   [ 0.7626]
Kurtosis                 0.0063491   [ 0.9365]
Bowman-Shenton            0.097609   [ 0.9524]
```

Yet the slope residual is still not normal.

Normality test for Slope residual

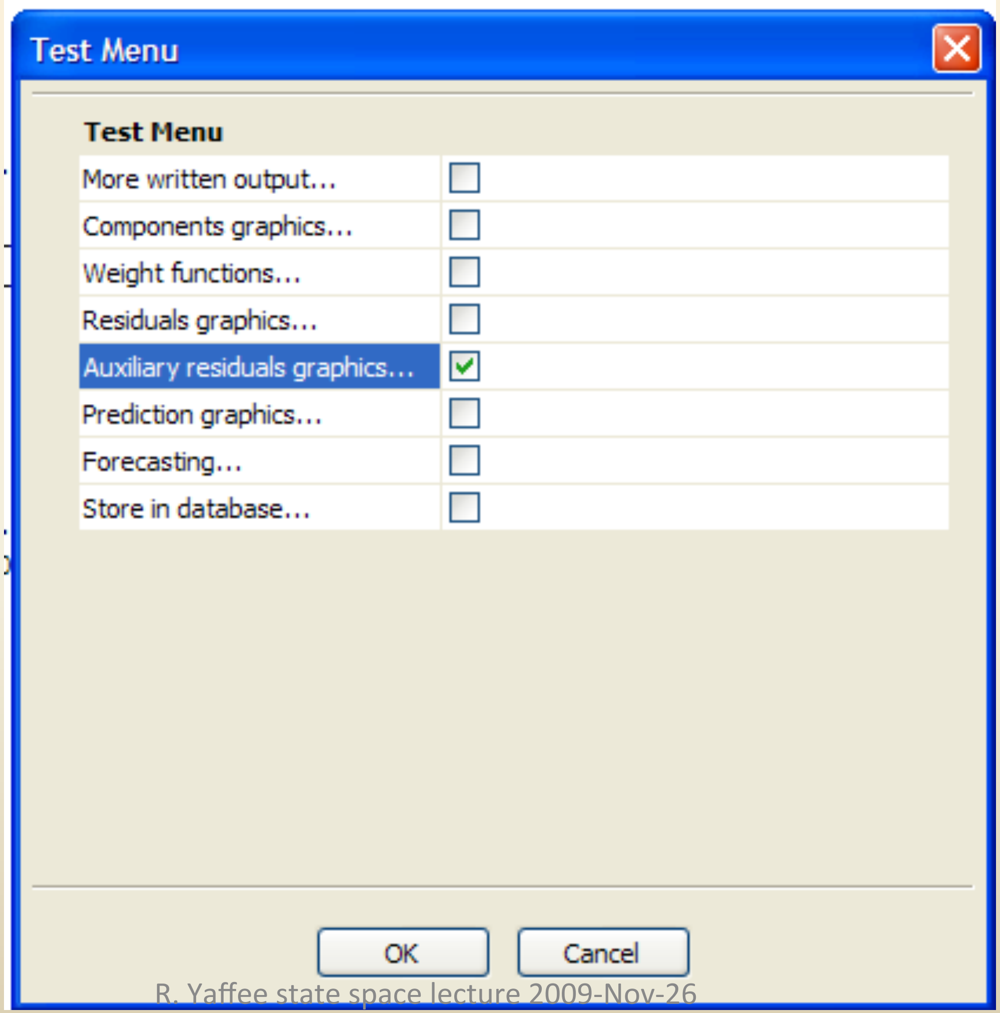
	Value
Sample size	466.00
Mean	0.69058
St.Dev	0.71606
Skewness	-0.38642
Excess kurtosis	-0.16212
Minimum	-1.1089
Maximum	2.9056

	Chi^2	prob
Skewness	11.598	[0.0007]
Kurtosis	0.51034	[0.4750]
Bowman-Shenton	12.108	[0.0023]

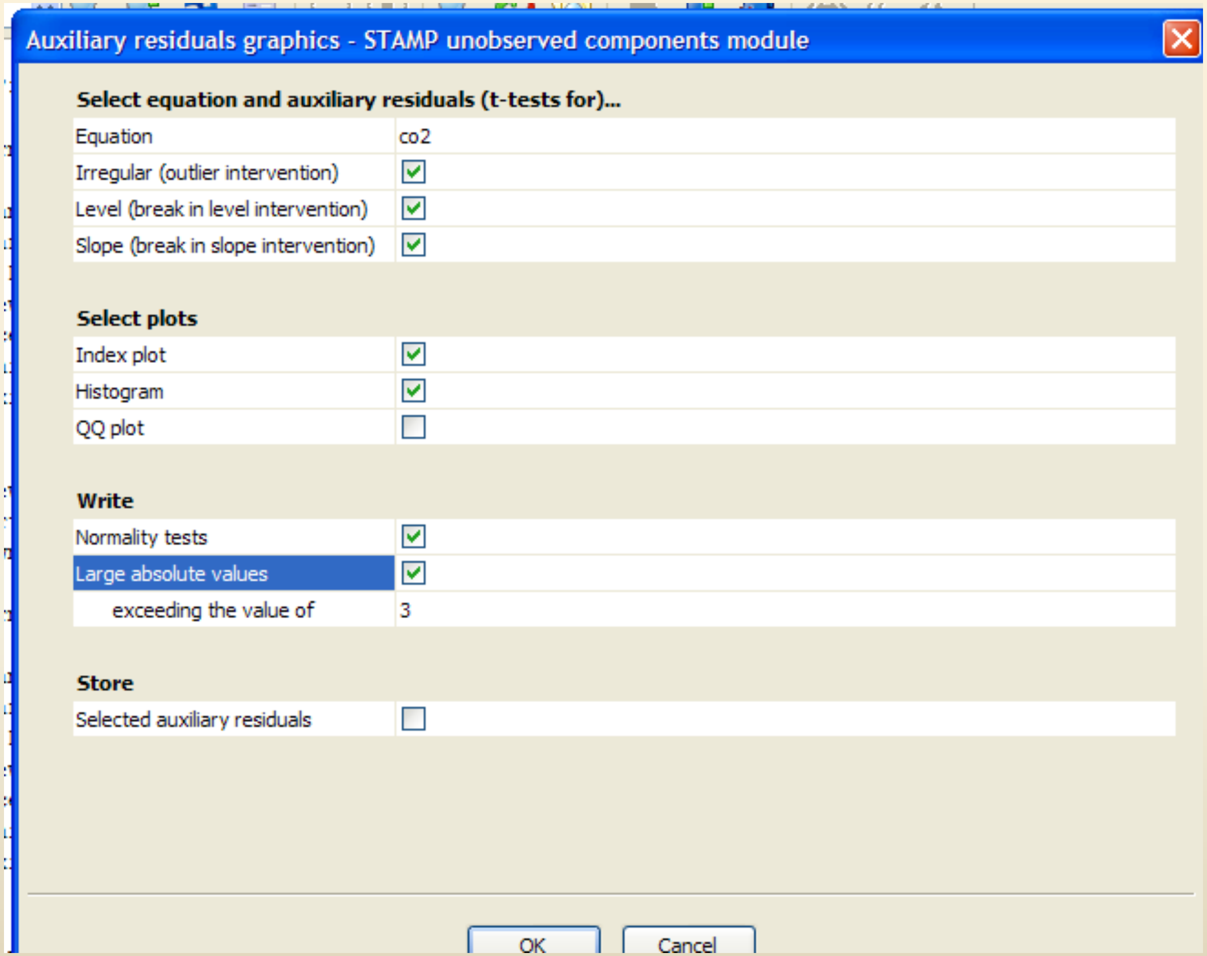
Diagnosis of residual problems begins with the Auxiliary residuals

- The auxiliary residuals are smoothed residuals divided by the square root of the F, the measurement variance. So they in effect are t-tests.
- We can look at graphical analysis of them for quick

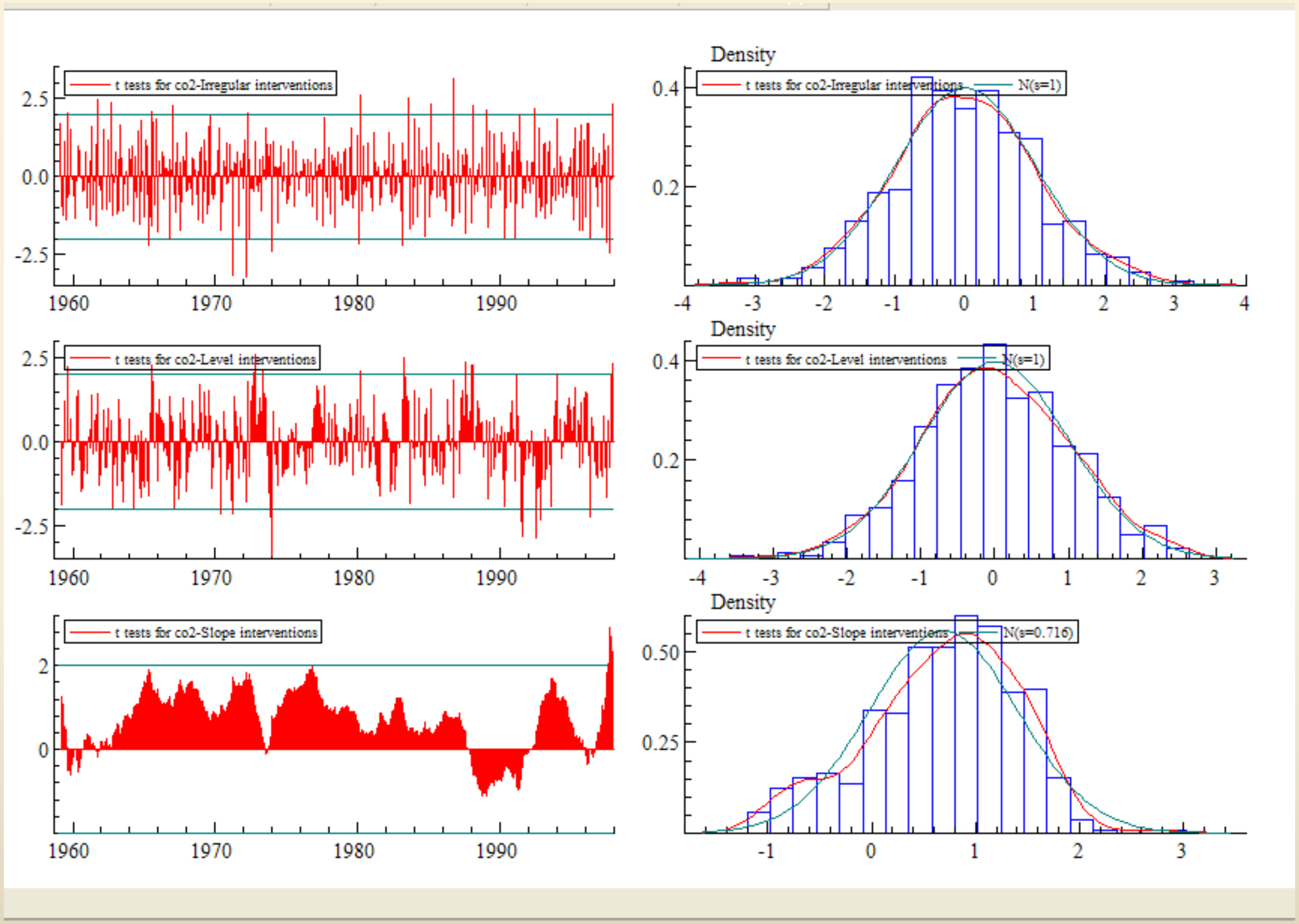
In the test menu, we select Auxiliary residuals graphics and then click OK.



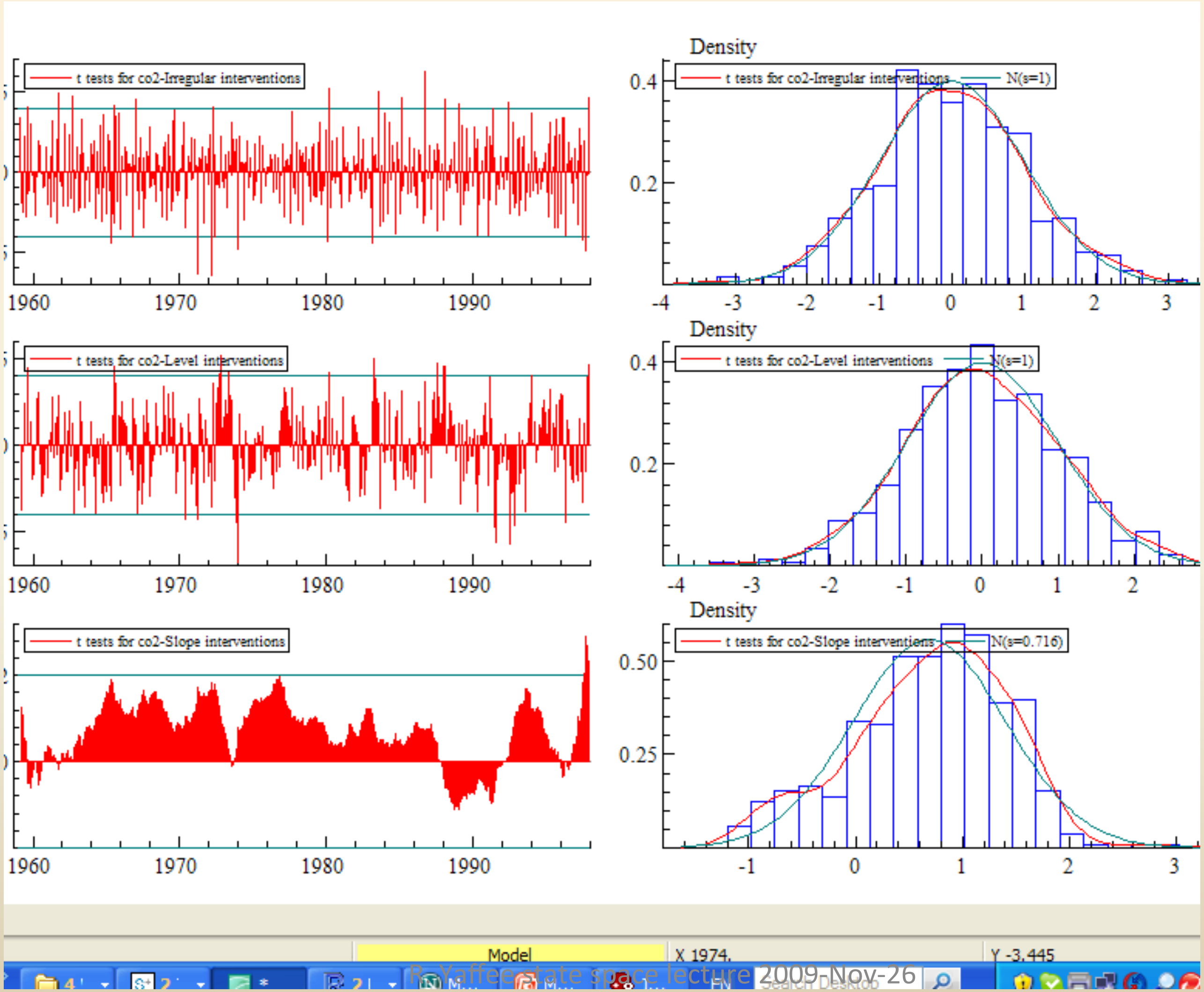
In the drop-down menu, we make the selections shown below:



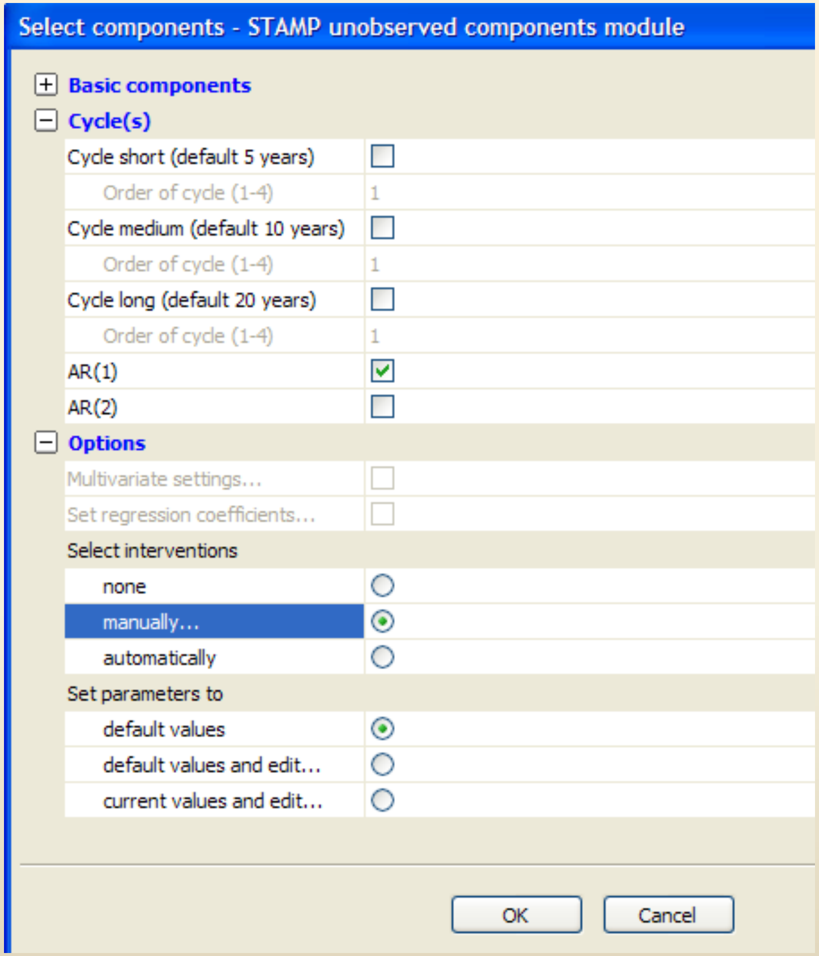
Time index plots and histograms of our auxiliary residuals



We examine the outliers and find that during the oil embargo of 1973 there was a huge drop in co_2 level and irregular



We go to our select components menu again and in the select interventions choose manual insertion

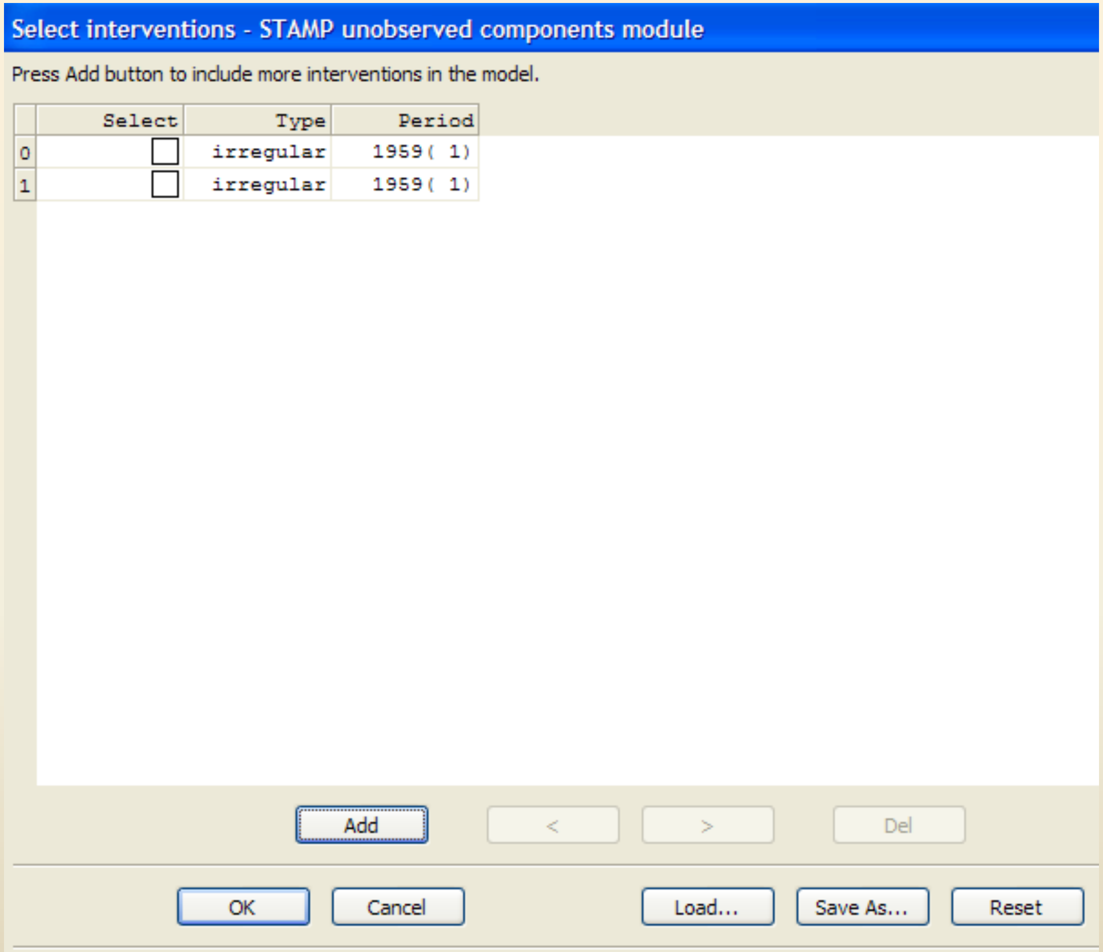


We then click on OK.

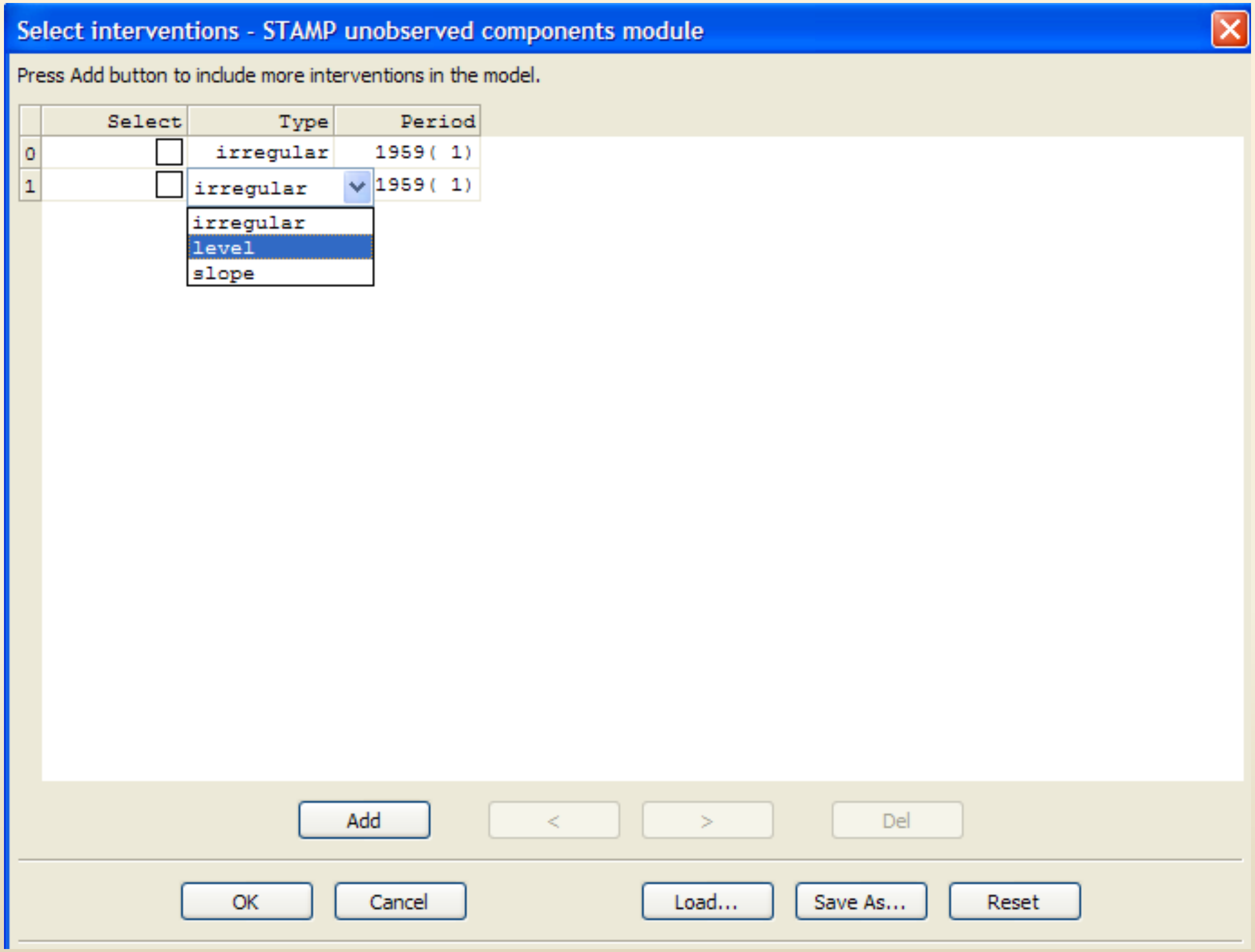
In the select menu, click on add to open up two intervention boxes

Now we will proceed to define the interventions

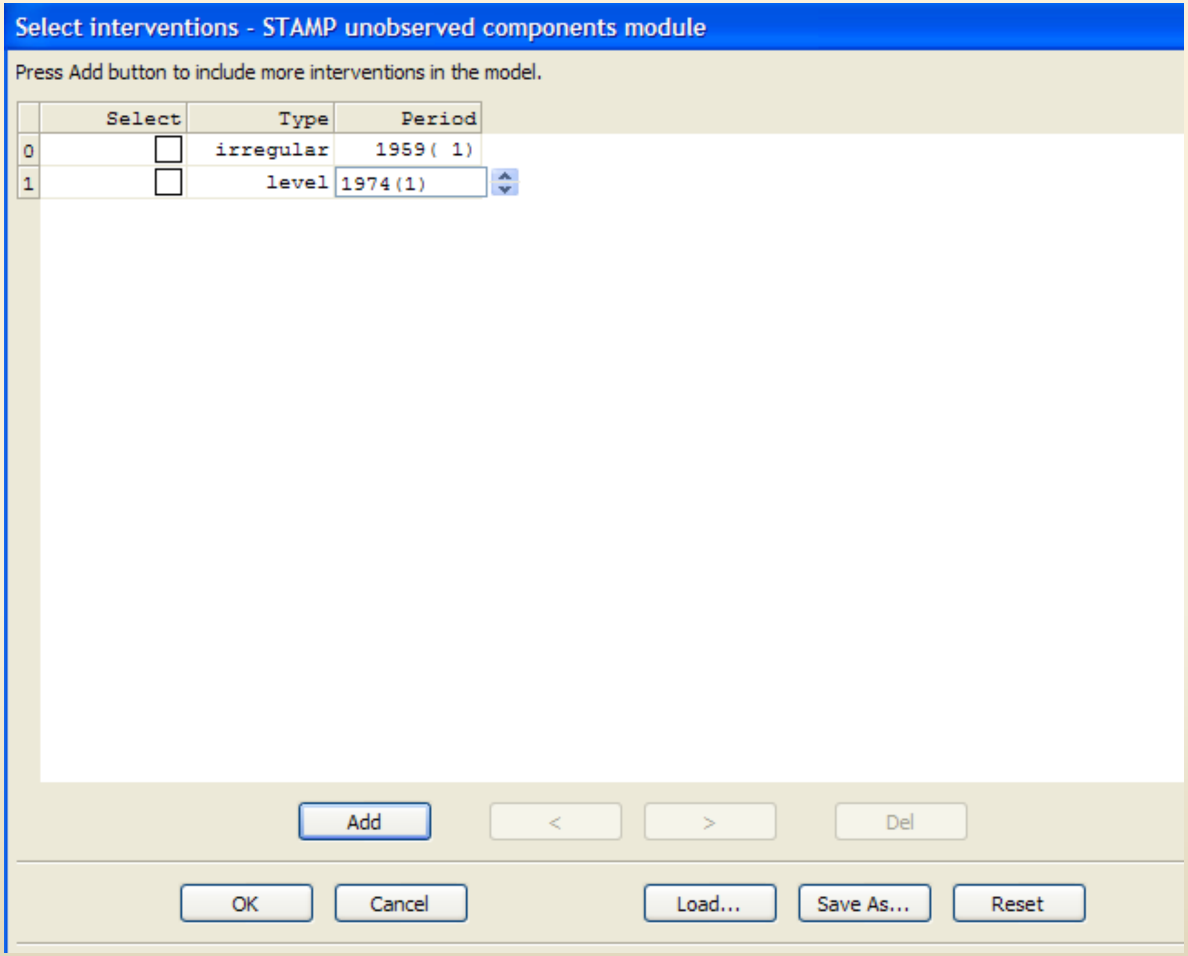
.



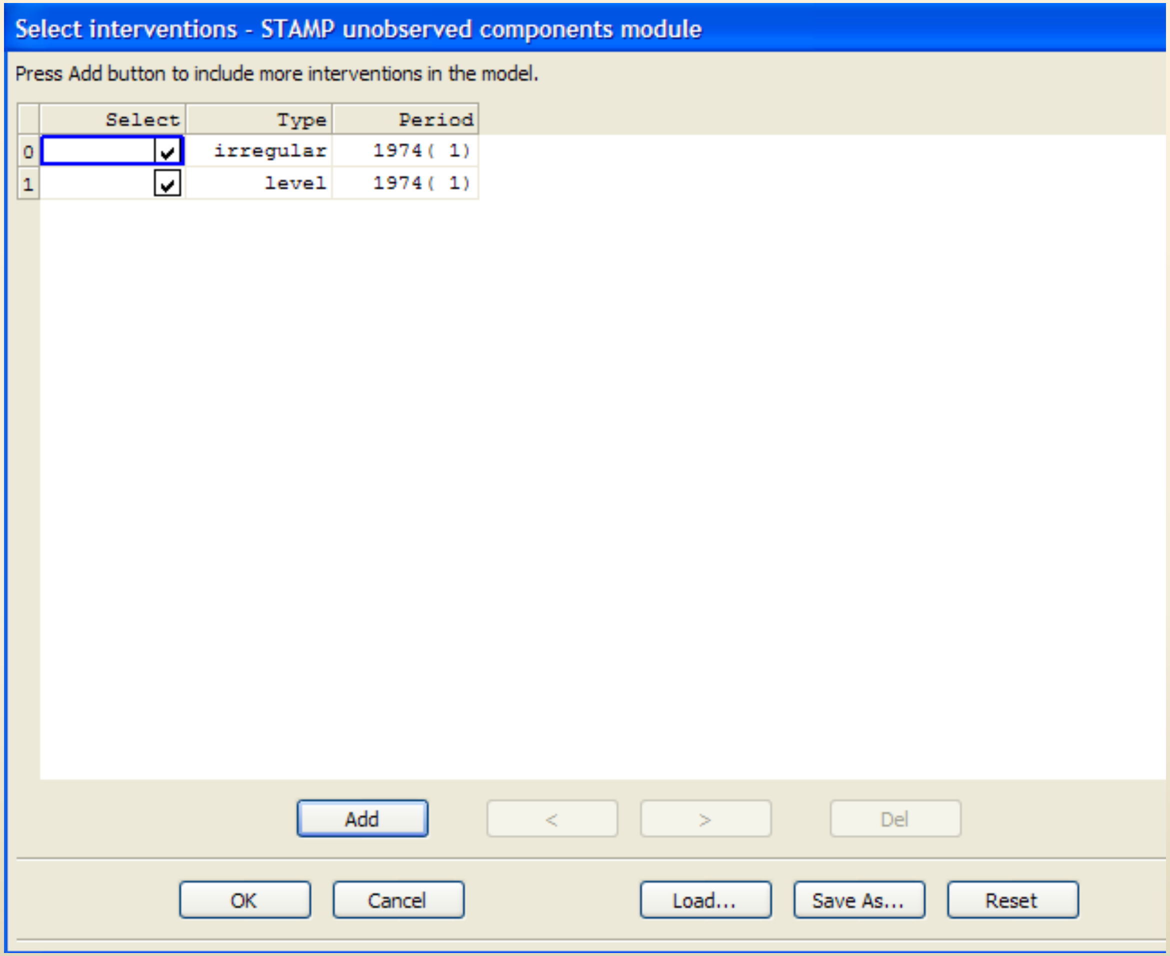
We click on type in the lower box and choose level



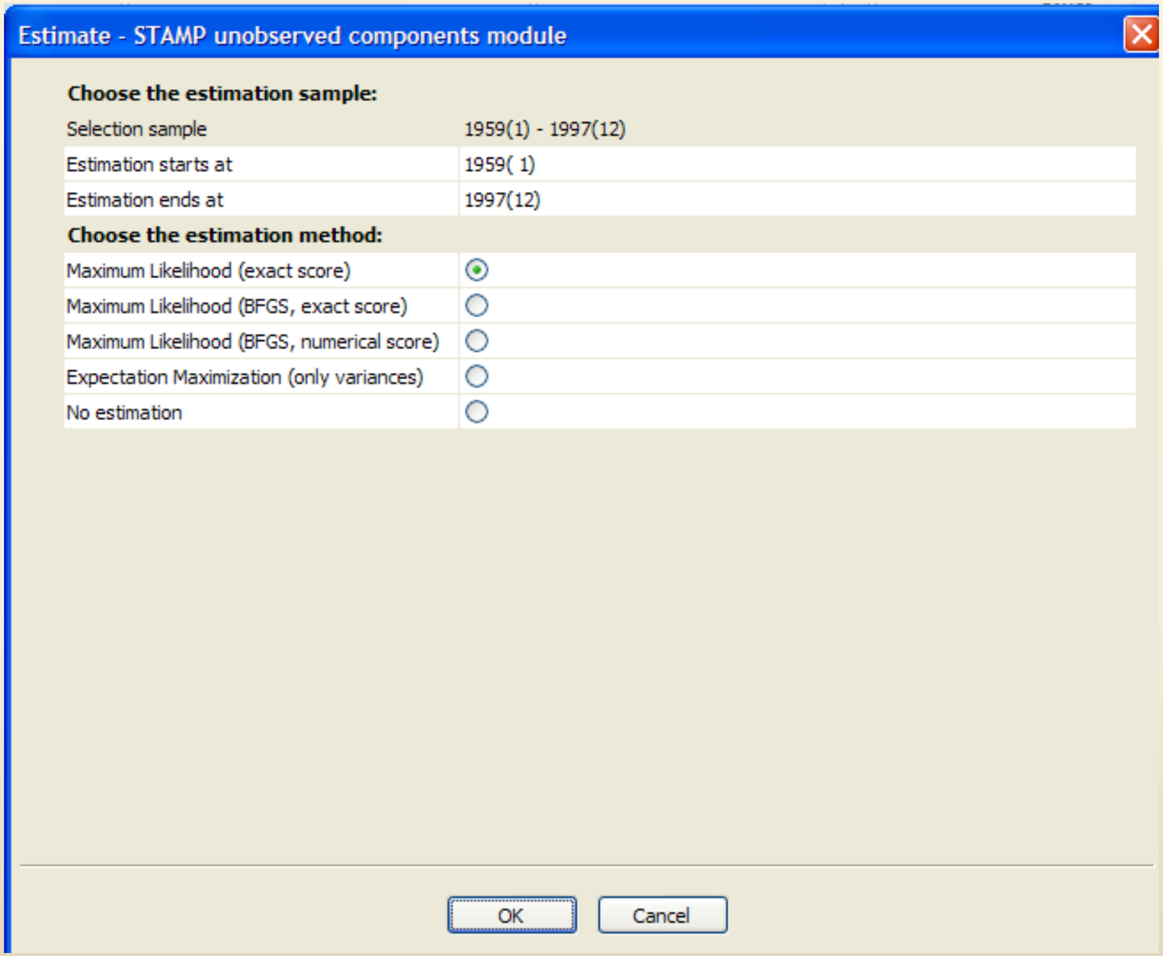
We change the date to the proper date and then click the box on the left



We configure the other outlier and then click ok at the bottom



Leave the defaults in the Estimate menu and click ok.



Our new model appears. Steady state strong convergence is found.

```
Estimating........
Strong convergence relative to 1e-007
- likelihood cvg 8.24892e-011
- gradient cvg 2.03461e-008
- parameter cvg 5.52095e-006
- number of bad iterations 0
Estimation process completed.

UC( 8) Estimation done by Maximum Likelihood (exact score)
   The database used is C:\Program Files\OxMetrics6\data\co2.in7
   The selection sample is: 1959(1) - 1997(12) (T = 468, N = 1)
   The dependent variable Y is: co2
   The model is:  Y = Trend + Seasonal + Irregular + AR(1) + Interventions
   Steady state. found

Log-Likelihood is 539.123 (-2 LogL = -1078.25).
Prediction error variance is 0.0821776

Summary statistics
                      co2
 T                 468.00
 p                 5.0000
 std.error        0.28667
 Normality         1.0323
 H(151)            1.0042
 DW                1.9184
 r(1)            0.030328
 q                 25.000
 r(q)            0.077997
 Q(q,q-p)          29.647
 Rs^2             0.10126
```

We see that all components remain stochastic (with a random error)

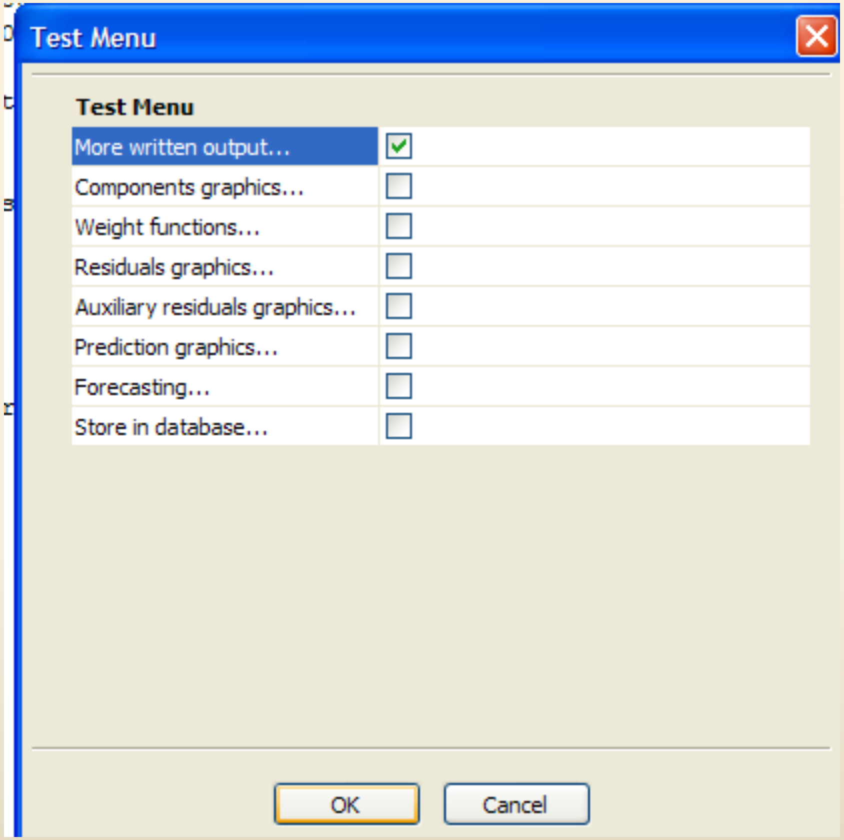
```
Summary statistics
                        co2
 T                   468.00
 p                   5.0000
 std.error          0.28667
 Normality           1.0323
 H(151)              1.0042
 DW                  1.9184
 r(1)              0.030328
 q                   25.000
 r(q)              0.077997
 Q(q,q-p)            29.647
 Rs^2               0.10126

 Variances of disturbances:
                      Value     (q-ratio)
Level             0.0182309  (   0.5882)
Slope          4.98100e-006  (0.0001607)
Seasonal       2.24629e-005  (0.0007248)
AR(1)             0.0309919  (    1.000)
Irregular         0.0155950  (   0.5032)

AR(1) other parameters:
AR coefficient      0.52894
```

Observe that the level shift at 1974 is not quite significant (n=468)

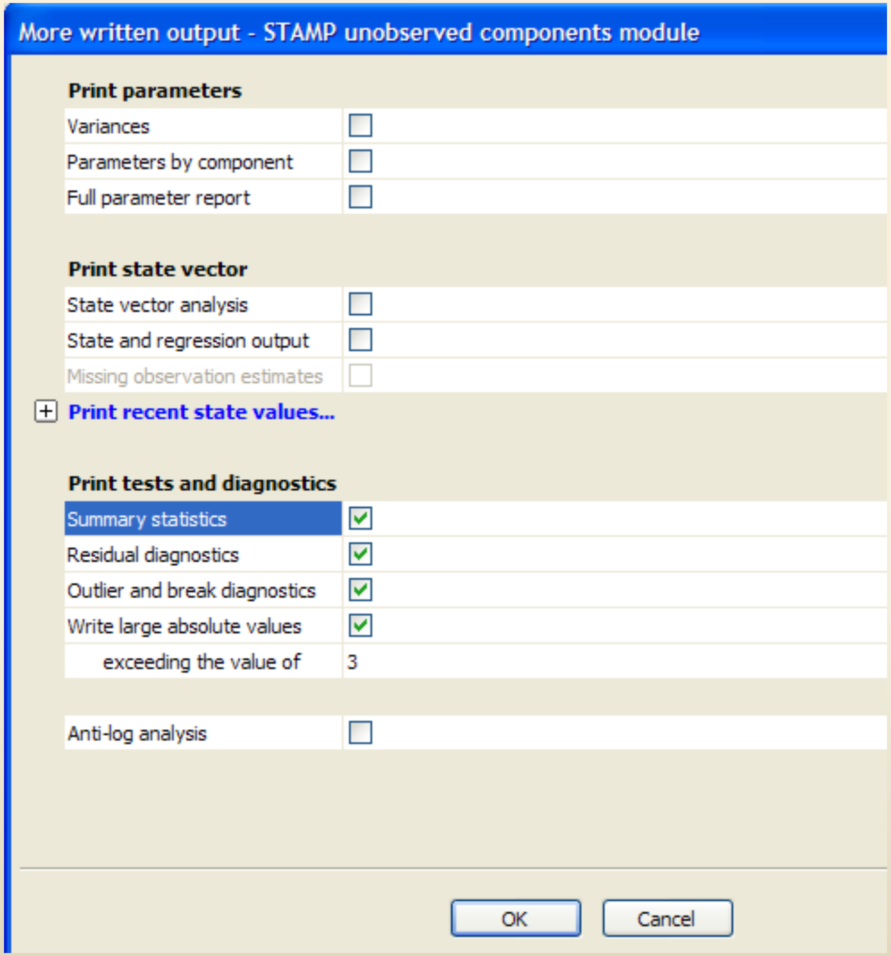
There is plenty of reason to believe that other outliers have not yet been modeled and that our model is afflicted by specification error.

```
State vector analysis at period 1997(12)
                              Value       Prob
Level                     365.31871 [0.00000]
Slope                       0.12929 [0.00000]
Seasonal chi2 test       3809.58740 [0.00000]
Seasonal effects:
                Period        Value       Prob
                     1      0.01656 [0.86626]
                     2      0.74517 [0.00000]
                     3      1.44037 [0.00000]
                     4      2.70524 [0.00000]
                     5      3.15084 [0.00000]
                     6      2.35317 [0.00000]
                     7      0.76517 [0.00000]
                     8     -1.39704 [0.00000]
                     9     -3.41117 [0.00000]
                    10     -3.38855 [0.00000]
                    11     -2.13292 [0.00000]
                    12     -0.84685 [0.00000]

Regression effects in final state at time 1997(12)

                    Coefficient        RMSE     t-value        Prob
Level break 1974(1)    -0.45970     0.25408    -1.80928 [0.07107]
```

We therefore request more written output



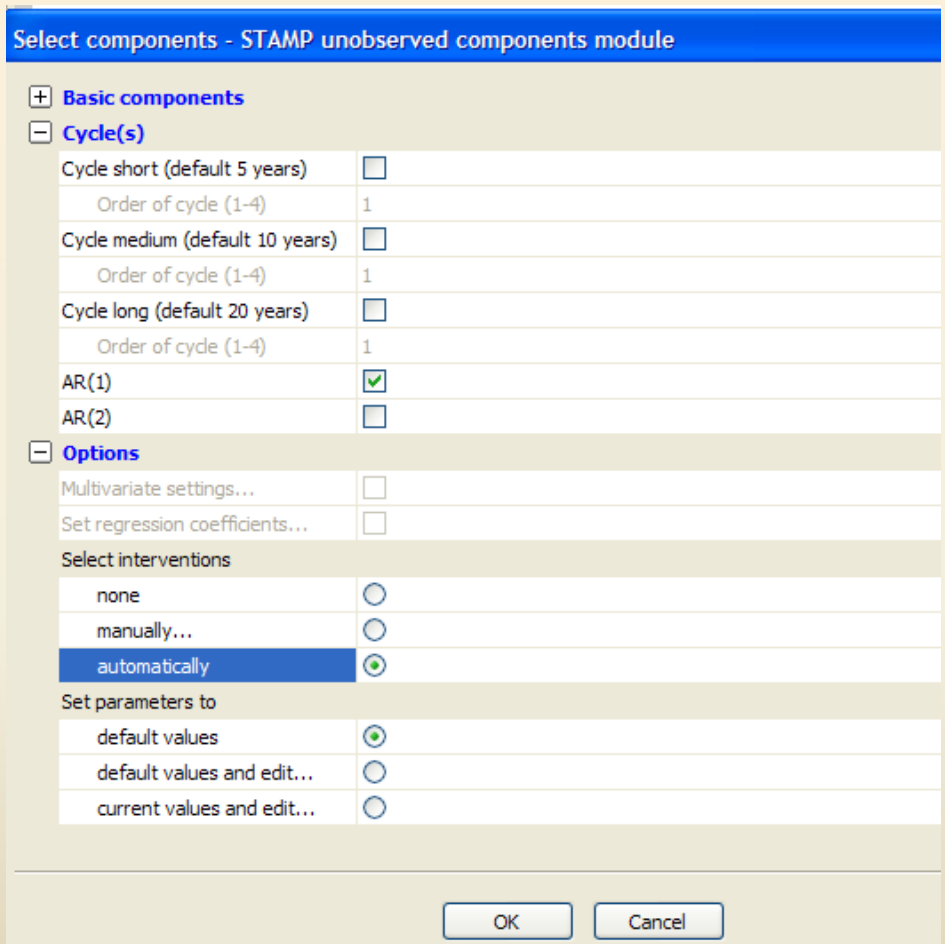
We ask for a rerun of the residual diagnostics and click on OK



We observe more unmodeled outliers.

```
Goodness-of-fit based on Residuals co2
                                              Value
Prediction error variance (p.e.v)          0.082178
Prediction error mean deviation (m.d)      0.066542
Ratio p.e.v. / m.d in squares               0.97093
Coefficient of determination R^2            0.99964
... based on differences Rd^2                 0.945
... based on diff around seas mean Rs^2     0.10126
Information criterion Akaike (AIC)          -2.4348
... Bayesian Schwartz (BIC)                 -2.3018

Serial correlation statistics for Residuals co2
 Durbin-Watson test is 1.91839
 Asymptotic deviation for correlation is 0.0469323
   Lag        df      Ser.Corr    BoxLjung       prob
     5         1      -0.01416      2.8394  [ 0.0920]
     6         2     -0.066555      4.8863  [ 0.0869]
     7         3    -0.0050167       4.898  [ 0.1794]
     8         4      -0.06751      7.0135  [ 0.1352]
     9         5       0.09933      11.604  [ 0.0406]
    12         8      0.035746      13.419  [ 0.0982]
    24        20     -0.066052      26.711  [ 0.1436]
    36        32     0.0015844      45.994  [ 0.0521]

Values larger than 3 for Irregular residual:
                    Value      prob
1971(4)          -3.19262 [0.00075]
1972(3)          -3.25050 [0.00062]
1986(9)           3.14345 [0.00089]
```

We select automatic intervention modeling this time around.



This time we have a good model with strong convergence.

```
AR(1) other parameters:
AR coefficient     0.41730

State vector analysis at period 1997(12)
                            Value       Prob
Level                   367.30883 [0.00000]
Slope                     0.13797 [0.00000]
Seasonal chi2 test     4971.96405 [0.00000]
Seasonal effects:
                Period      Value       Prob
                     1    0.01425 [0.86699]
                     2    0.73328 [0.00000]
                     3    1.42725 [0.00000]
                     4    2.67870 [0.00000]
                     5    3.11441 [0.00000]
                     6    2.31165 [0.00000]
                     7    0.80697 [0.00000]
                     8   -1.36072 [0.00000]
                     9   -3.39110 [0.00000]
                    10   -3.36828 [0.00000]
                    11   -2.12097 [0.00000]
                    12   -0.84544 [0.00000]

Regression effects in final state at time 1997(12)

                     Coefficient        RMSE     t-value       Prob
Outlier 1971(4)         -0.70182     0.20702    -3.39007 [0.00076]
Outlier 1972(3)         -0.72682     0.20699    -3.51135 [0.00049]
Outlier 1986(9)          0.72766     0.20687     3.51745 [0.00048]
Level break 1973(12)    -0.86752     0.22330    -3.88503 [0.00012]
Level break 1991(7)     -0.78972     0.22515    -3.50757 [0.00050]
Level break 1992(7)     -0.81321     0.22558    -3.60503 [0.00035]
```

We proceed to residual diagnosis looking for white noise residuals

Goodness-of-fit based on Residuals co2

	Value
Prediction error variance (p.e.v)	0.072109
Prediction error mean deviation (m.d)	0.061917
Ratio p.e.v. / m.d in squares	0.86347
Coefficient of determination R^2	0.99969
... based on differences Rd^2	0.95227
... based on diff around seas mean Rs^2	0.22006
Information criterion Akaike (AIC)	-2.5441
... Bayesian Schwartz (BIC)	-2.3668

Serial correlation statistics for Residuals co2

Durbin-Watson test is 1.93891

Asymptotic deviation for correlation is 0.0471929

Lag	df	Ser.Corr	BoxLjung	prob
5	1	-0.019196	3.4346	[0.0638]
6	2	-0.072638	5.8465	[0.0538]
7	3	0.023065	6.0902	[0.1073]
8	4	-0.045068	7.0229	[0.1347]
9	5	0.085687	10.402	[0.0646]
12	8	-0.00031172	10.676	[0.2207]
24	20	-0.074838	26.077	[0.1633]
36	32	0.033135	52.499	[0.0126]

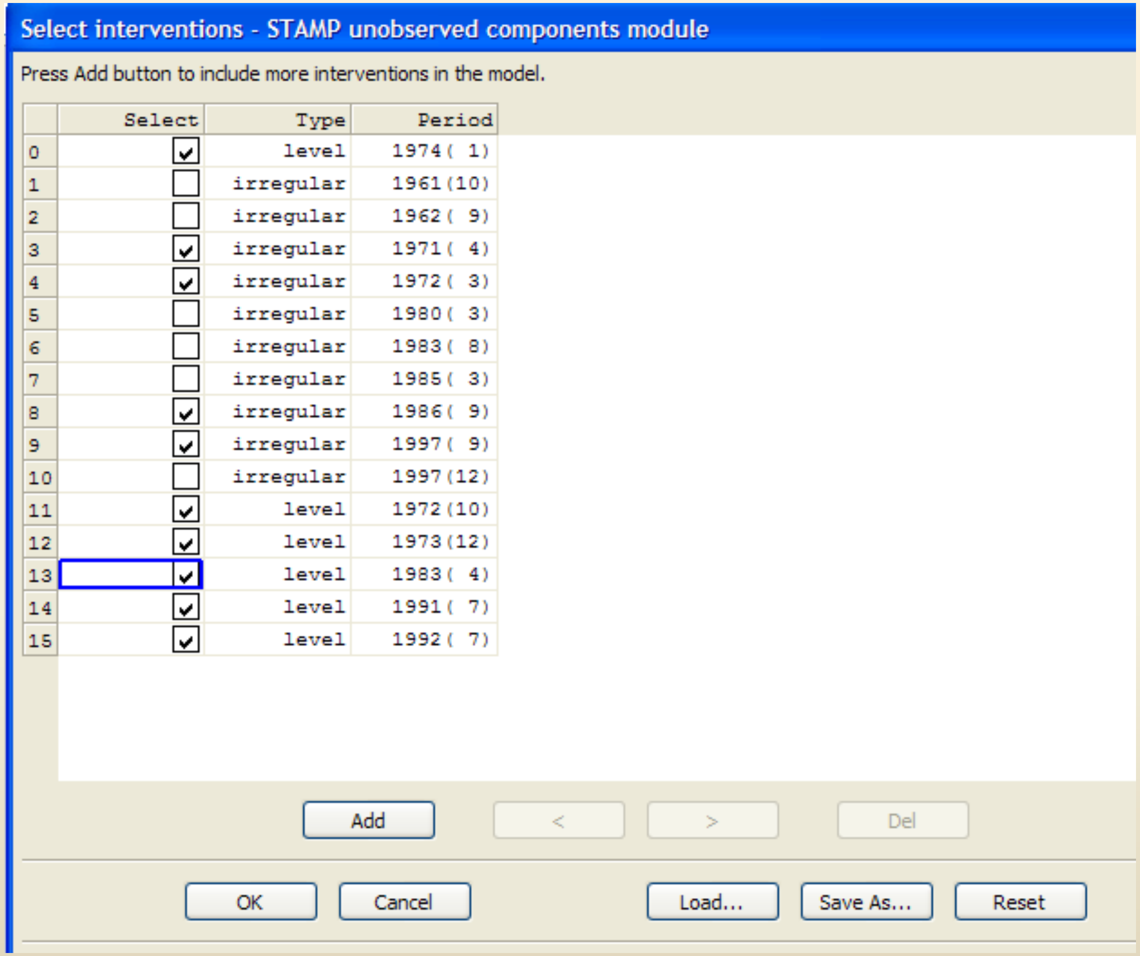
Normality test for Irregular residual

	Value
Sample size	468.00
Mean	0.00078048
St.Dev	1.0601
Skewness	0.11722
Excess kurtosis	-0.13869
Minimum	-2.7478
Maximum	2.9396

The level residuals are good, but there remains a slope shift at 1997(9)

```
Normality test for Irregular residual
                          Value
 Sample size             468.00
Mean                 0.00078048
St.Dev                   1.0601
Skewness                0.11722
Excess kurtosis        -0.13869
Minimum                 -2.7478
Maximum                  2.9396

                         Chi^2        prob
Skewness                1.0718  [ 0.3005]
Kurtosis               0.37508  [ 0.5402]
Bowman-Shenton          1.4468  [ 0.4851]

Normality test for Level residual
                          Value
 Sample size             467.00
Mean                -0.00096108
St.Dev                   1.0666
Skewness                0.26111
Excess kurtosis        -0.15426
Minimum                 -2.7689
Maximum                  2.9577

                         Chi^2        prob
Skewness                5.3066  [ 0.0212]
Kurtosis               0.46305  [ 0.4962]
Bowman-Shenton          5.7696  [ 0.0559]

Values larger than 3 for Slope residual:
                        Value      prob
1997(9)               3.18025 [0.00078]
```

We add that intervention and then run the model



	Select	Type	Period
0	☑	level	1974(1)
1	☐	irregular	1961(10)
2	☐	irregular	1962(9)
3	☑	irregular	1971(4)
4	☑	irregular	1972(3)
5	☐	irregular	1980(3)
6	☐	irregular	1983(8)
7	☐	irregular	1985(3)
8	☑	irregular	1986(9)
9	☑	irregular	1997(9)
10	☐	irregular	1997(12)
11	☑	level	1972(10)
12	☑	level	1973(12)
13	☑	level	1983(4)
14	☑	level	1991(7)
15	☑	level	1992(7)

The new model is interesting for the interventions found

State vector at period 1997(12)

	Coefficient	RMSE	t-value	Prob
Level	365.49444	0.60679	602.34127	[0.00000]
Slope	0.12829	0.01678	7.64424	[0.00000]
Seasonal	-1.74460	0.04788	-36.43570	[0.00000]
Seasonal 2	2.35447	0.04737	49.70084	[0.00000]
Seasonal 3	0.82588	0.03782	21.83749	[0.00000]
Seasonal 4	-0.00646	0.03872	-0.16676	[0.86764]
Seasonal 5	0.13084	0.03477	3.76308	[0.00019]
Seasonal 6	-0.06255	0.03530	-1.77220	[0.07704]
Seasonal 7	-0.13079	0.03388	-3.86044	[0.00013]
Seasonal 8	-0.03528	0.03352	-1.05251	[0.29313]
Seasonal 9	0.01058	0.03364	0.31452	[0.75327]
Seasonal10	0.00014	0.03291	0.00411	[0.99673]
Seasonal11	0.02368	0.02819	0.84005	[0.40133]
AR(1)	0.01123	0.17655	0.06359	[0.94932]

Regression effects in final state at time 1997(12)

	Coefficient	RMSE	t-value	Prob
Outlier 1971(4)	-0.72283	0.21741	-3.32474	[0.00096]
Outlier 1972(3)	-0.70844	0.21740	-3.25863	[0.00121]
Outlier 1986(9)	0.71065	0.21728	3.27074	[0.00116]
Slope break 1997(9)	0.19328	0.09086	2.12719	[0.03395]
Level break 1972(10)	0.68416	0.23031	2.97064	[0.00313]
Level break 1973(12)	-0.85455	0.23026	-3.71127	[0.00023]
Level break 1983(4)	0.61582	0.23026	2.67450	[0.00776]
Level break 1991(7)	-0.79240	0.23193	-3.41655	[0.00069]
Level break 1992(7)	-0.79662	0.23242	-3.42757	[0.00067]

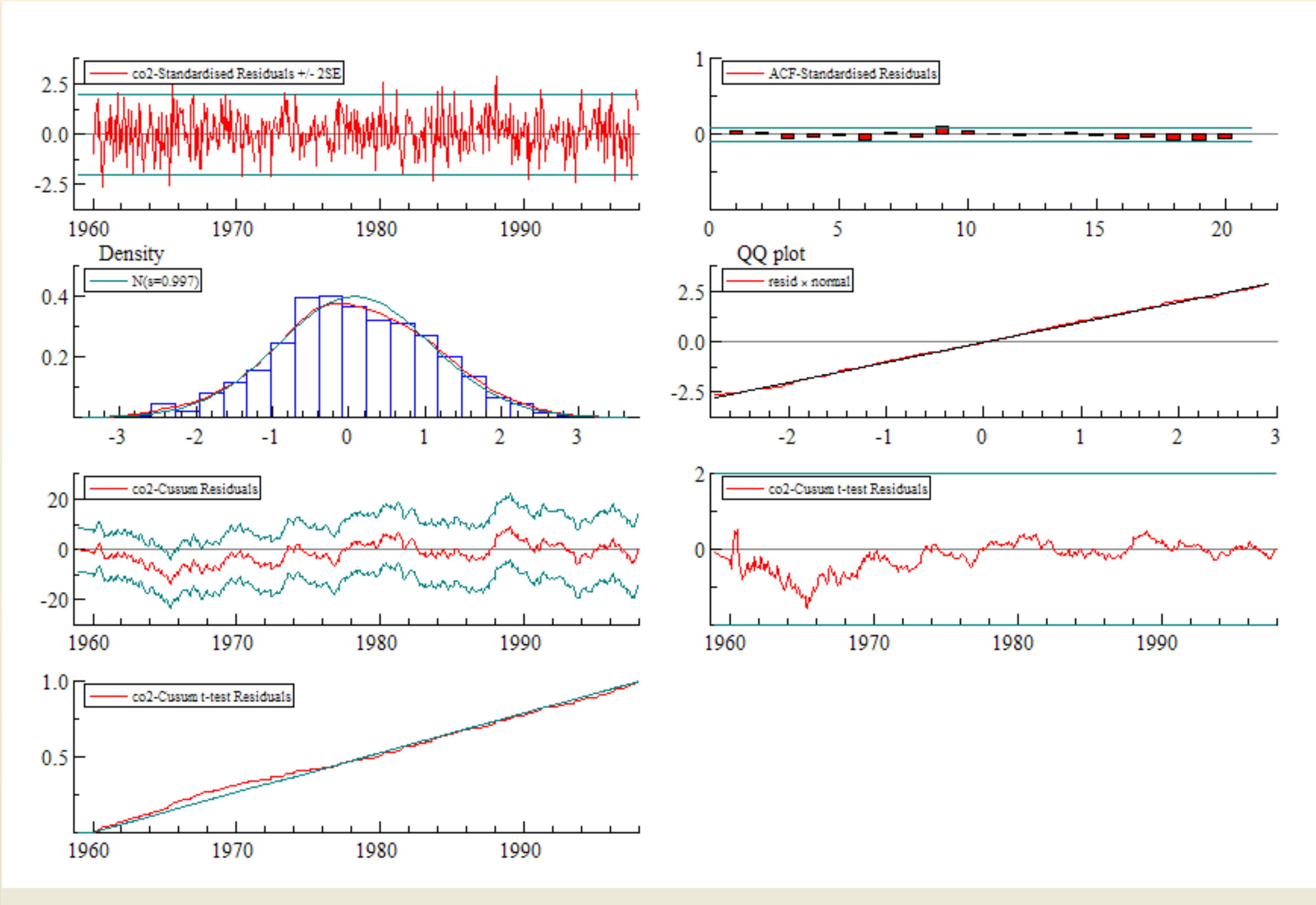
Although the AR(1) term is no longer significant, we leave it in to avoid biased estimation

- This is a judgment call. It we should try it both ways and see what happens.   As long as we have well behaved residuals with the slight exception of the slope residual which appears to be significantly skewed and hence nonnormal, we know from quasi-Maximum likelihood that this may not be a real problem.

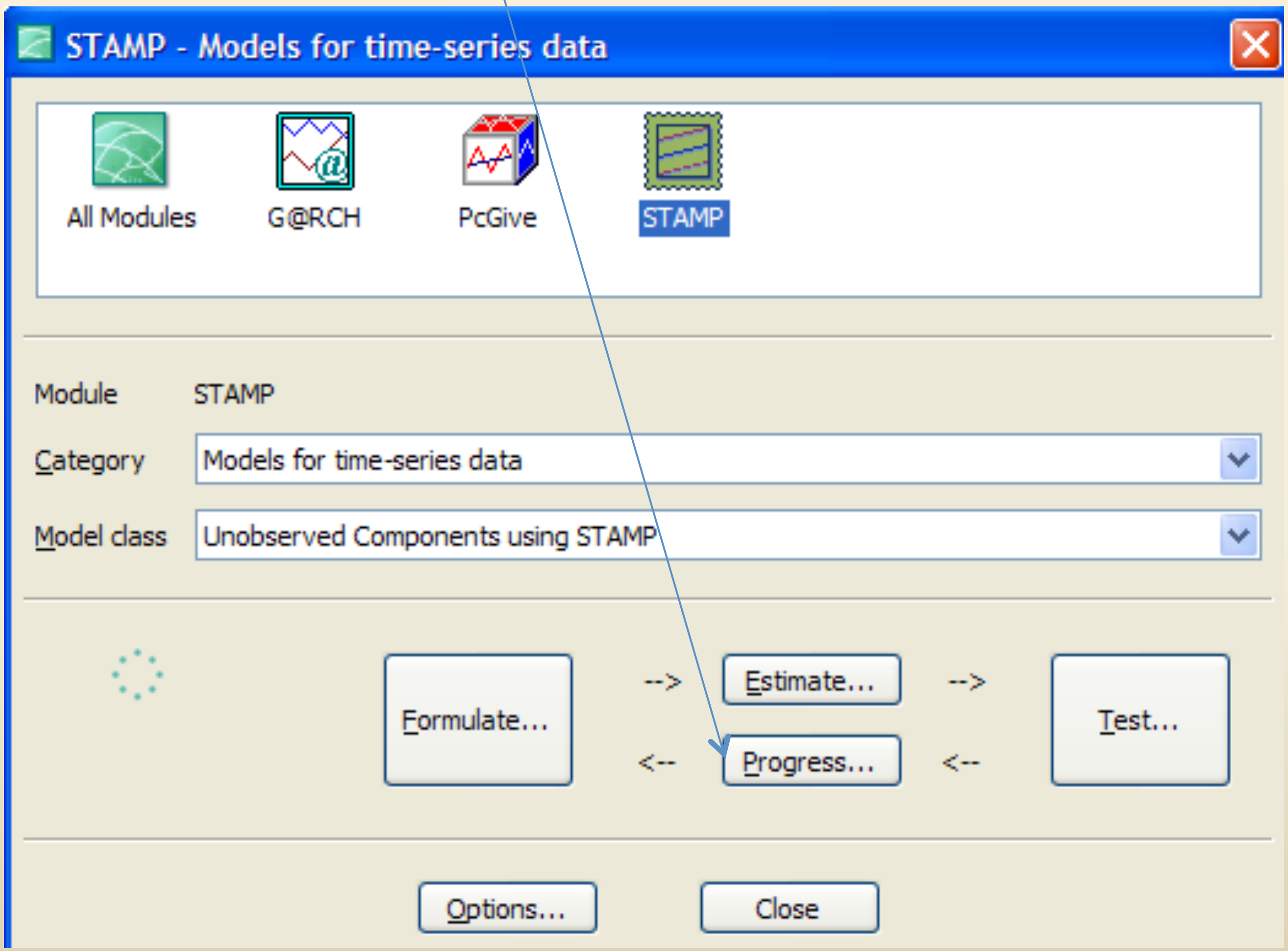
Further diagnosis

- There is no evidence of specification error since we modeled all of the outliers and level shifts we could.
- Now we review our residual graphics to see how well our model fits.

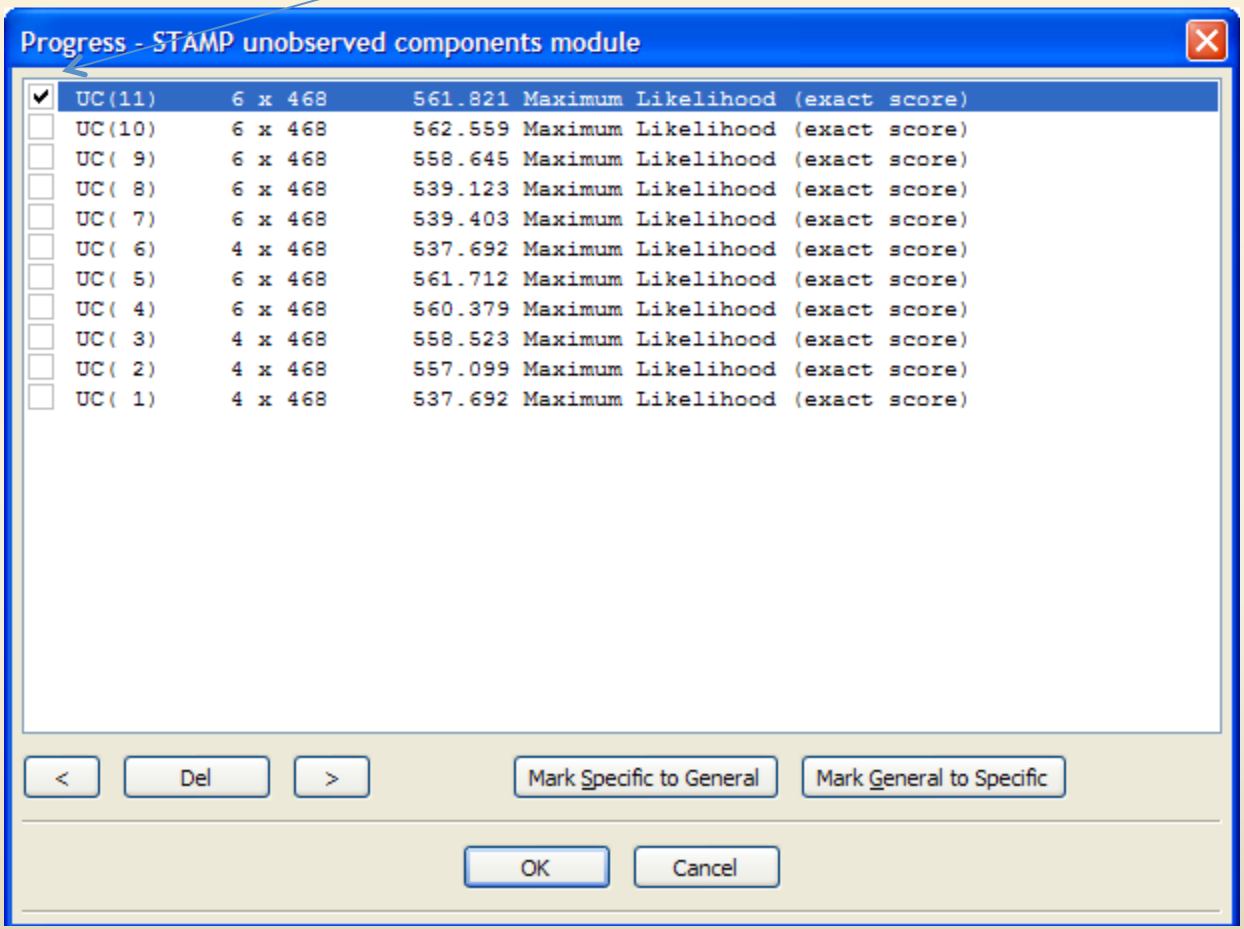
Residual graphics show residuals to be well behaved —not too noisy and they stay in line.



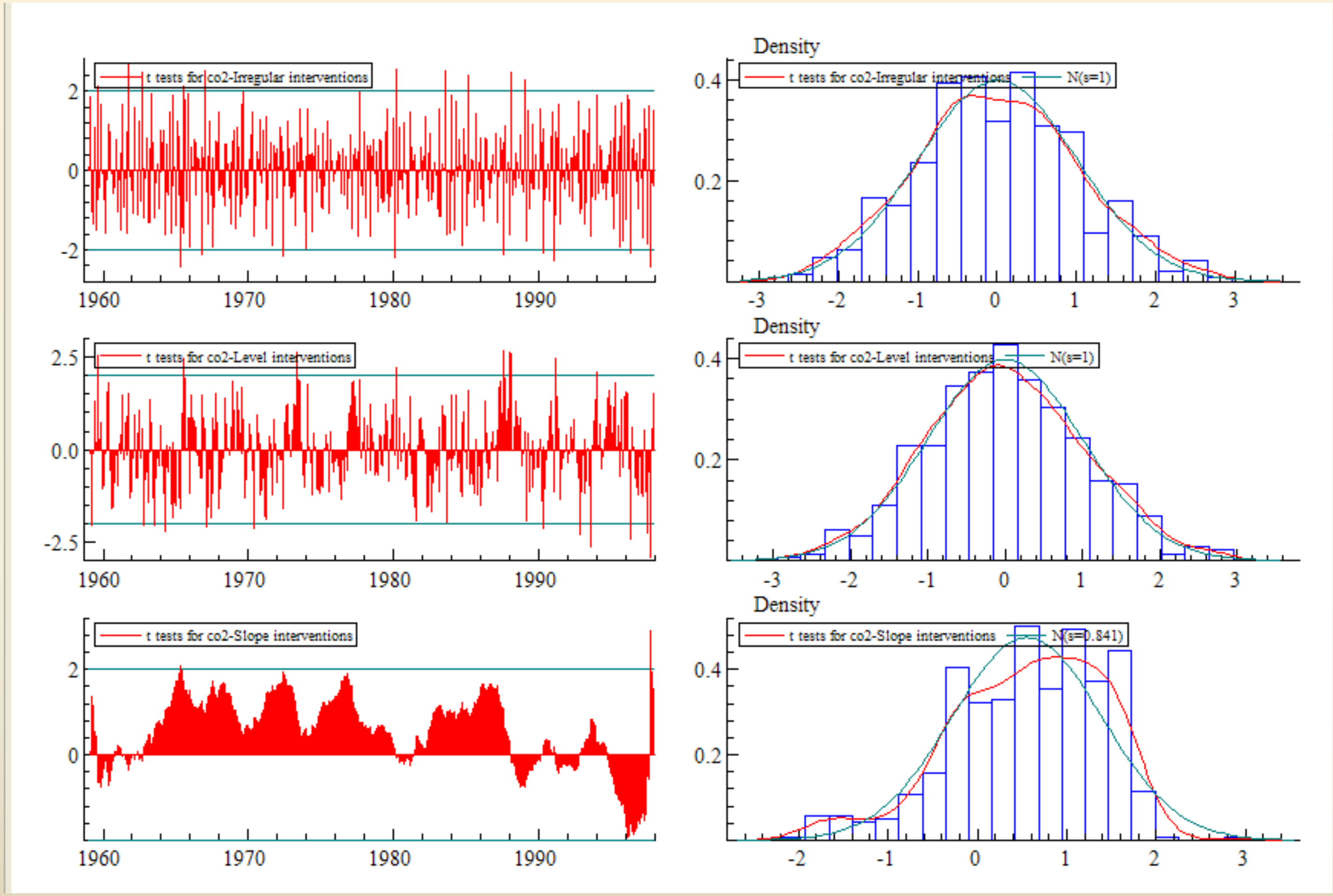
Click on formulate icon and then the progress button



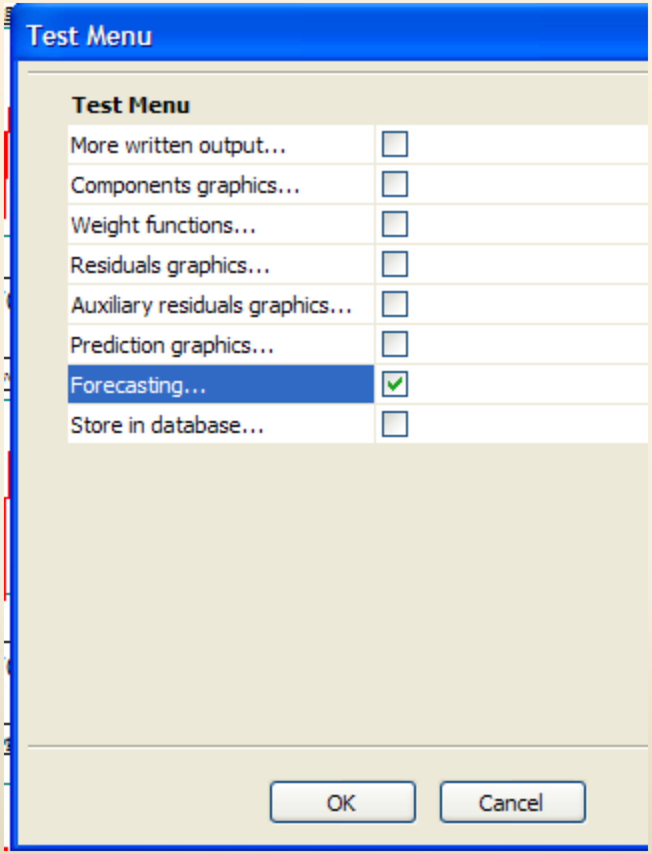
Of all the models run, the most recent has the maximum likelihood.



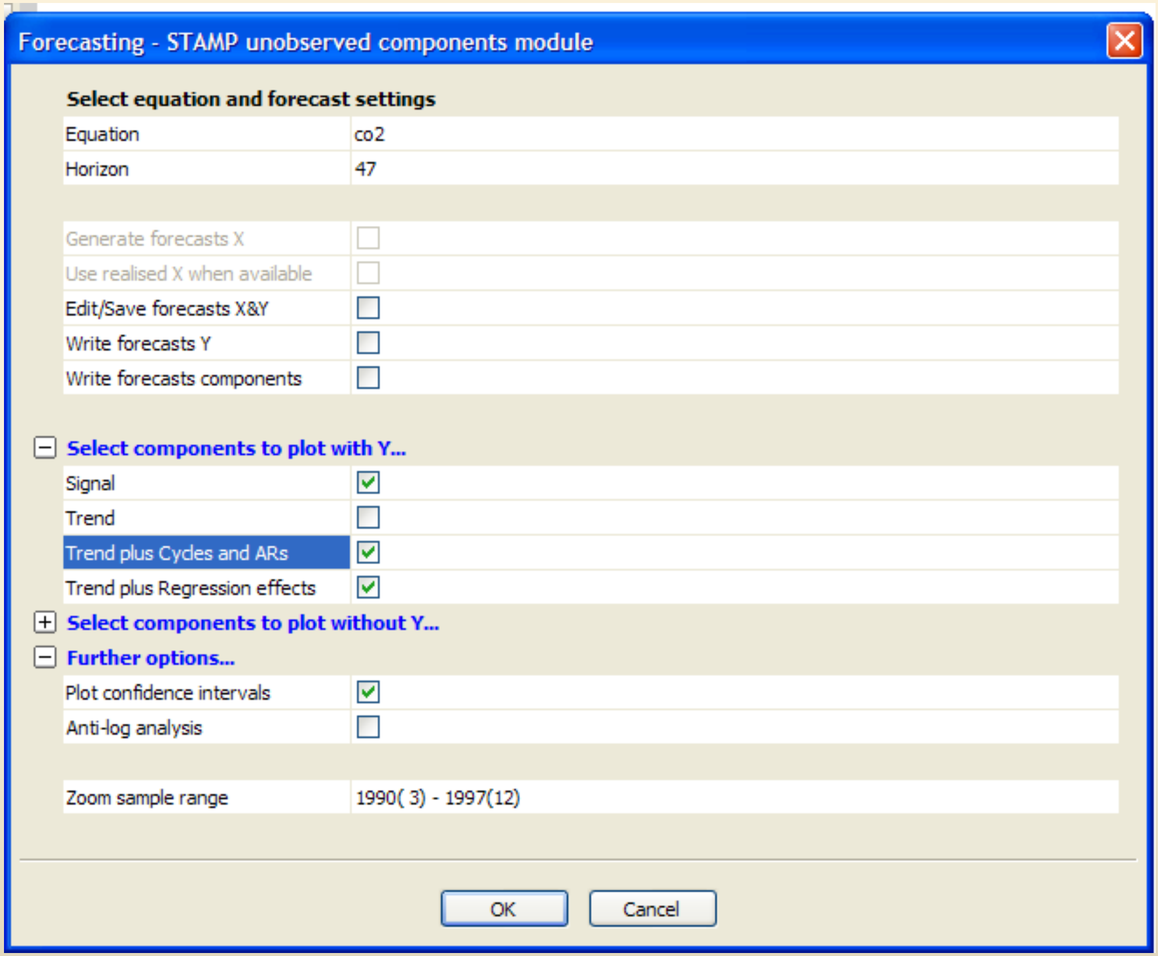
The end effect on the slope may bias a forecast but not the fit. Be wary of using this model for forecasting.



Suppose we had to forecast, we would then select forecast in the test menu



We could select these options and click on OK.



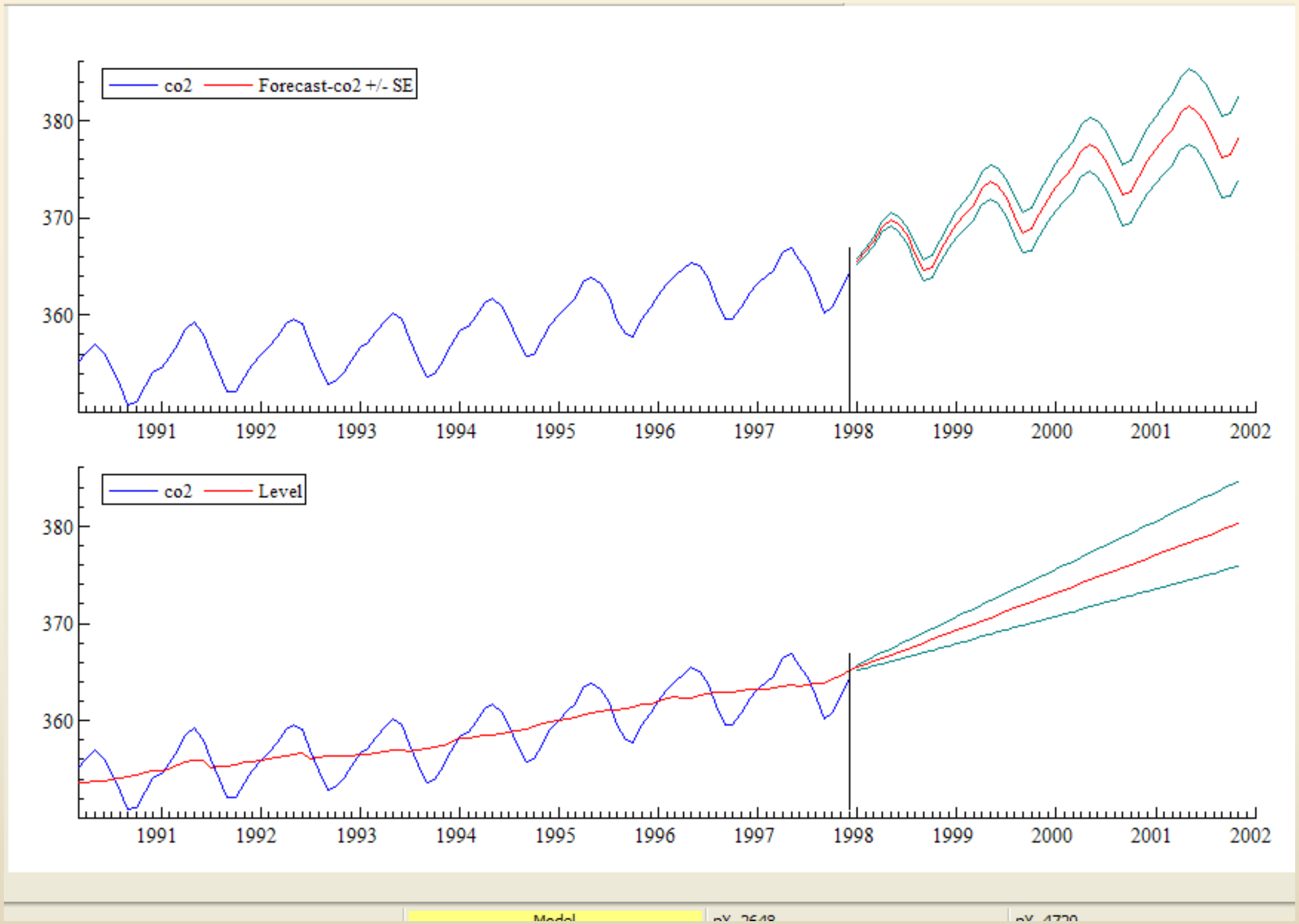
We can obtain ex ante forecasts for the whole series as well as for the separate components

```
Forecasts with 68% confidence interval from period 1997(12) forwards:
           Forecast   stand.err   leftbound  rightbound
1         365.46677     0.31697   365.14980   365.78373
2         366.51499     0.41157   366.10342   366.92656
3         367.52956     0.50445   367.02511   368.03401
4         369.10108     0.59376   368.50733   369.69484
5         369.85947     0.68349   369.17598   370.54296
6         369.37681     0.77149   368.60533   370.14830
7         368.19980     0.86125   367.33854   369.06105
8         366.35732     0.95181   365.40551   367.30912
9         364.62113     1.03029   363.59084   365.65142
10        364.95447     1.10898   363.84549   366.06346
11        366.51755     1.18799   365.32956   367.70554
12        368.09861     1.26200   366.83661   369.36060

Forecast values for Level
Forecasts with 68% confidence interval from period 1997(12) forwards:
           Forecast   stand.err   leftbound  rightbound
1         365.62273     0.62018   365.00255   366.24290
2         365.75101     0.63372   365.11729   366.38473
3         365.87930     0.64743   365.23188   366.52673
4         366.00759     0.66129   365.34630   366.66888
5         366.13588     0.67532   365.46056   366.81120
6         366.26417     0.68950   365.57467   366.95367
7         366.39246     0.70384   365.68862   367.09630
8         366.52075     0.71833   365.80242   367.23908
9         366.64904     0.73297   365.91607   367.38200
10        366.77732     0.74776   366.02957   367.52508
11        366.90561     0.76269   366.14292   367.66830
12        367.03390     0.77777   366.25613   367.81167
Ssf() warning: SLOPE can not be part of signal
```

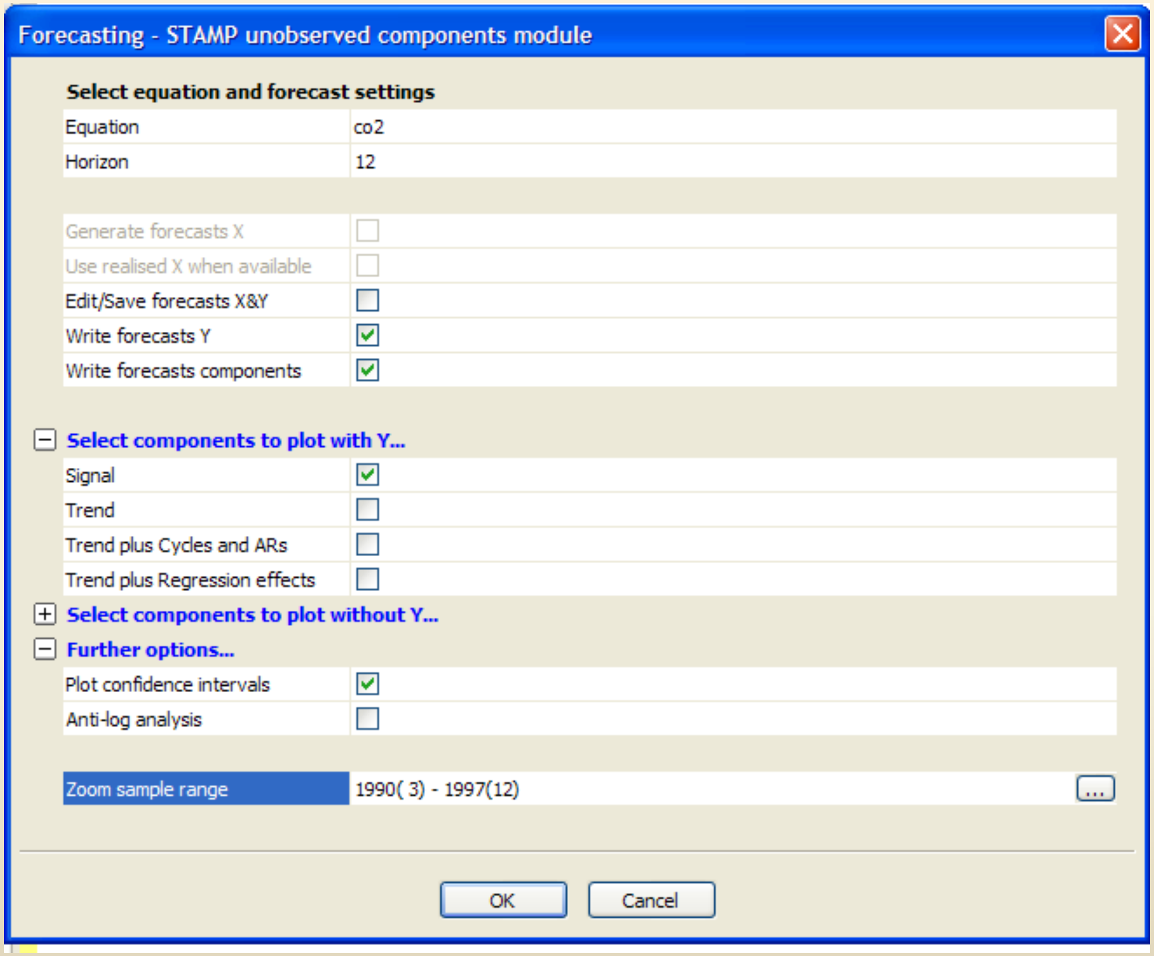
Stamp warns us about the slope being an unreliable part of the signal (the end effect)

```
Forecast values for Level
Forecasts with 68% confidence interval from period 1997(12) forwards:
           Forecast   stand.err   leftbound  rightbound
1         365.62273     0.62018   365.00255   366.24290
2         365.75101     0.63372   365.11729   366.38473
3         365.87930     0.64743   365.23188   366.52673
4         366.00759     0.66129   365.34630   366.66888
5         366.13588     0.67532   365.46056   366.81120
6         366.26417     0.68950   365.57467   366.95367
7         366.39246     0.70384   365.68862   367.09630
8         366.52075     0.71833   365.80242   367.23908
9         366.64904     0.73297   365.91607   367.38200
10        366.77732     0.74776   366.02957   367.52508
11        366.90561     0.76269   366.14292   367.66830
12        367.03390     0.77777   366.25613   367.81167
Ssf() warning: SLOPE can not be part of signal

Forecast values for Seasonal
Forecasts with 68% confidence interval from period 1997(12) forwards:
           Forecast   stand.err   leftbound  rightbound
1           0.01323     0.09192    -0.07870     0.10515
2           0.74218     0.09192     0.65026     0.83410
3           1.43682     0.09203     1.34478     1.52885
4           2.68793     0.09212     2.59581     2.78005
5           3.12558     0.09237     3.03321     3.21796
6           2.32195     0.09261     2.22934     2.41456
7           0.82378     0.09287     0.73091     0.91665
8          -1.33997     0.09344    -1.43341    -1.24653
9          -3.39751     0.09248    -3.48999    -3.30503
10         -3.38559     0.09243    -3.47802    -3.29316
11         -2.14398     0.09317    -2.23715    -2.05080
12         -0.88441     0.09669    -0.98111    -0.78772
```

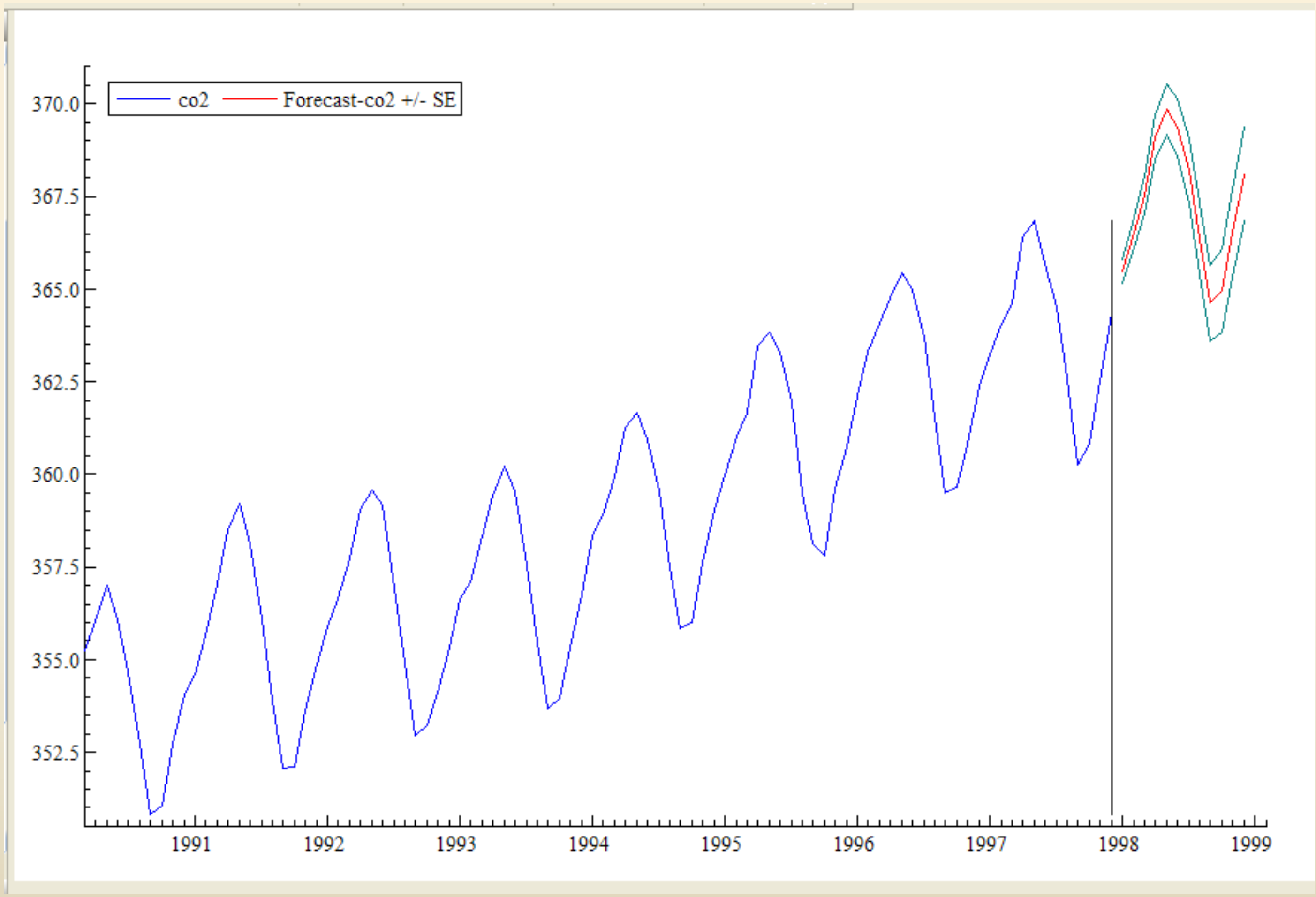
Ex ante forecast



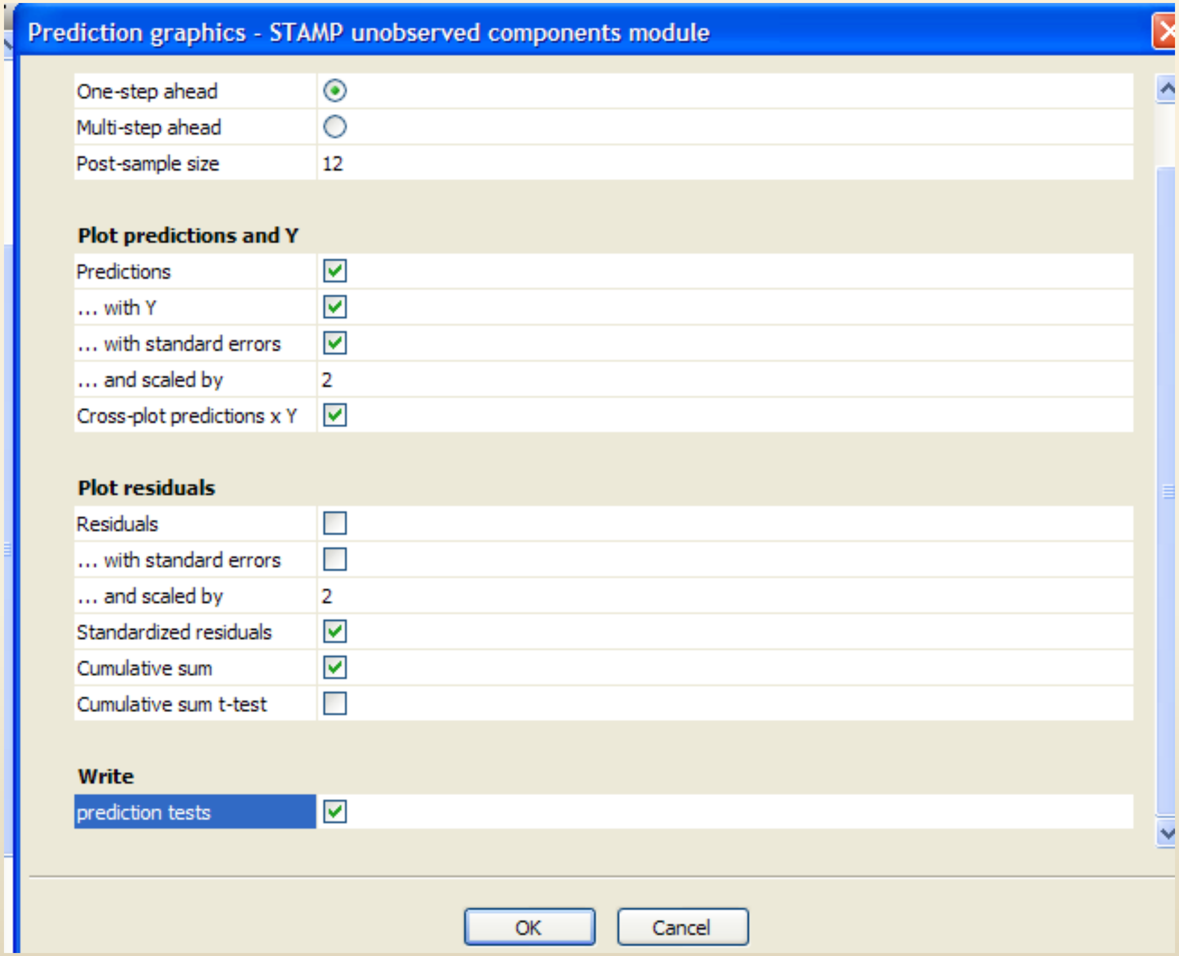
Forecast evaluation



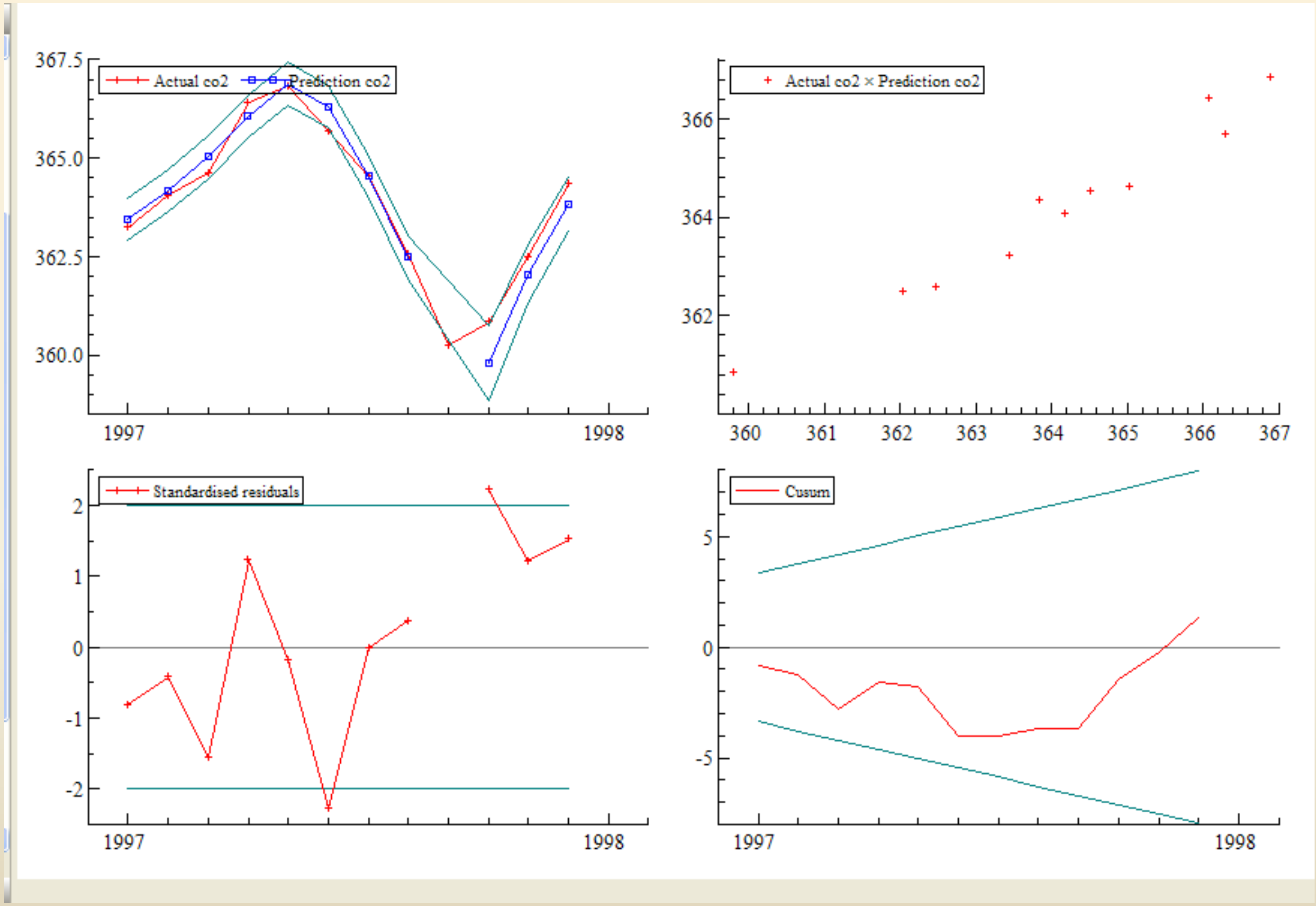
We can shorten the forecast horizon to improve our forecast accuracy



After requesting Prediction Graphics select the boxes below for forecast evaluation



Some of the Prediction graphics



Out-of-sample forecast evaluation

Prediction analysis for 12 post-sample predictions (with 1 missing values).

	error	stand.err	residual	cusum	sqrsum
1997(1)	-0.2208	0.2692	-0.8204	-0.8204	0.6730
1997(2)	-0.1134	0.2692	-0.4212	-1.242	0.8504
1997(3)	-0.4193	0.2692	-1.557	-2.799	3.276
1997(4)	0.3308	0.2692	1.229	-1.570	4.786
1997(5)	-0.04834	0.2692	-0.1796	-1.750	4.818
1997(6)	-0.6106	0.2692	-2.268	-4.018	9.964
1997(7)	-0.001921	0.2703	-0.007105	-4.025	9.964
1997(8)	0.09913	0.2694	0.3680	-3.657	10.10
1997(9)	.NaN	3162.	0.0000	-3.657	10.10
1997(10)	1.041	0.4683	2.222	-1.435	15.04
1997(11)	0.4474	0.3688	1.213	-0.2217	16.51
1997(12)	0.5104	0.3348	1.525	1.303	18.83

Post-sample predictive tests.
Failure Chi2(11) test is 18.8346 [0.0641]
Cusum t(11) test is 0.3929 [0.7019]

Post-sample prediction statistics.
Sum of 11 absolute prediction errors is 3.84283
Sum of 11 squared prediction errors is 2.27563
Sum of 11 absolute prediction resids is 11.8111
Sum of 11 squared prediction resids is 18.8346

The cyclical component

Stochastic cyclicity:

$$\begin{bmatrix}\psi_{t+1}\\ \psi^{*}_{t+1}\end{bmatrix} = \rho\begin{bmatrix}\cos\lambda_c & \sin\lambda_c\\ -\sin\lambda_c & \cos\lambda_c\end{bmatrix}\begin{bmatrix}\psi_t\\ \psi^{*}_t\end{bmatrix} + \begin{bmatrix}\chi_t\\ \chi^{*}_t\end{bmatrix} \qquad \chi_t, \chi^{*}_t \sim N\left(0, \sigma^2_{\chi_t}\right)$$

$$\mathrm{cov}(\chi_t, \chi^{*}_t) = 0$$

where

$$\rho = damping\ \ parameter\ \ s.t.\ \ 0 <= \rho <= 1$$

$$\lambda_c = frequency\ of\ the\ cycle = \frac{2\pi}{p_c}$$

$$where\ \ p_c = period\ \ of\ the\ cycle$$

The cyclic distribution Koopman et al. (2008, 22-23).

$$\begin{bmatrix} \psi_{t+1} \\ \psi^*_{t+1} \end{bmatrix} = \rho \begin{bmatrix} \cos\lambda_c & \sin\lambda_c \\ -\sin\lambda_c & \cos\lambda_c \end{bmatrix} \begin{bmatrix} \psi_t \\ \psi^*_t \end{bmatrix} + \begin{bmatrix} \chi_t \\ \chi^*_t \end{bmatrix} \qquad \chi_t, \chi^*_t \sim N\left(0, \sigma^2_{\chi_t}\right)$$

$$\text{cov}(\chi_t, \chi^*_t) = 0$$

$$\begin{pmatrix} \chi_t \\ \chi^*_t \end{pmatrix} \sim NID\left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \sigma^2_\psi(1-\rho^2)I_2\right]$$

when $\rho \to 1$, *the cycle component reduces to a deterministic (but stationary) sine - cosine wave.*

What are the system matrices?

Koopman, Shephard, and Doornik (2008, 9)

$$\underset{m+N\,x\,1}{\begin{pmatrix}\alpha_{t+1}\\ y_t\end{pmatrix}}=\underset{m+N\,x\,1}{\begin{pmatrix}d_t\\ c_t\end{pmatrix}}+\underset{m+N\,x\,m}{\begin{pmatrix}T_t\\ Z_t\end{pmatrix}}\underset{m\,x\,1}{\alpha_t}+\underset{m+N\,x\,r}{\begin{pmatrix}H_t\\ G_t\end{pmatrix}}\underset{rx1}{\varepsilon_t}$$

$$m = dimension\ of\ the\ transition\ equation$$

$$N = dimension\ of\ the\ measurement\ model$$

$$\begin{pmatrix}\alpha_{t+1}\\ y_t\end{pmatrix} = state\ vector \qquad \delta = \underset{(m+N)x1}{\begin{pmatrix}d_t\\ c_t\end{pmatrix}} = constant\ vector$$

$$\boldsymbol{\Phi}_t = \underset{(m+N)\ x\ m}{\begin{pmatrix}T_t\\ Z_t\end{pmatrix}} = transition\ matrix$$

$$u_t = \underset{(m+N)xr}{\begin{pmatrix}H_t\\ G_t\end{pmatrix}}\varepsilon_t \sim NID(\boldsymbol{0},\Omega_t) \qquad \Omega_t = \underset{(m+N)x(m+N)}{\begin{pmatrix}HH' & HG'\\ GH' & HH'\end{pmatrix}}$$

$$where\ n = number\ of\ observations$$

$$r = \dim ension\ of\ the\ disturbance\ vector$$

We can define, constrain, or limit parameters in these matrices

Most matrices start with an m before their name. This is a notational convention of SsfPack.

We can decide whether these matrices will be time-varying or constant. We index these Phi, Omega, and sigma matrices by J. All elements within are = -1 except those that vary with time.

We can define whether these elements are known or unknown, to be initialized as diffuse or not.

We can insert - 1 to indicate that the element will receive diffuse initialization or not.

Input to Stsm matrix

Ibid, 24

mStsm				
Cmp	Col 1	Col 2	Col 3	Col 4
Level	σ_η	*0*	*0*	*0*
Slope	σ_ζ	*0*	*0*	*0*
Trend	σ_ζ	m	*0*	*0*
Seas _dummy	σ_ω	s	*0*	*0*
Cycle0	σ_ψ	λ_c	ρ	*0*
:	M	M	M	M
Cycle9	σ_ψ	λ_c	ρ	*0*
BWCYC	σ_ψ	λ_c	ρ	m
Irregular	σ_ξ	*0*	*0*	*0*

Missing Values

Missing data can be estimated by data augmentation or filtering if they exist in the measurement model or the data matrix.

Periods signify missing values in Ox. In SPlus, the missing value is NA. Vectors with missing values are automatically reduced to Vectors without missing values for analysis.

The system matrices are presumed known and given and cannot have missing values within them. When some matrices are not Relevant for the formulation of a state space, they can be left blank.

Data matrix mXt

mXt is a k by n matrix of exogenous variables.

The number of columns = sample size
The number of rows= number of time-varying elements in the matrix.

If this is a time series, it is usually called mYt

A side note.

When regressors are added to a local level model, the time-varying level serves as a constant. Therefore, we do not add a column of ones to them to avoid unnecessary multicollinearity Ibid, 28.

If you are adding a deterministic time trend and do not already have a local level, a constant would be acceptable so long as you did not previously center your data.

Adding Regressors to the State Space Model

GetSsfReg is the function that is used for this purpose.

When it does so, it estimates the model by recursive least squares. This is an OLS algorithm applied to a widening window expanding one step ahead each cycle of window extension. Koopman et al(2008,27) suggest that the multiple linear regression analysis can be specified in state space form as

$$\alpha_{t+1} = \alpha_t$$

$$y_t = X_t \alpha_t + G_t \varepsilon_t \quad \varepsilon_t \sim NID(\mathbf{0}, \sigma_\varepsilon^2) \quad t = \mathbf{1}, \ldots, n$$

To add regressors to a model in SsfPack

```
#include <oxstd.h>
#include <c:\Program Files\OxMetrics51\Ox\packages\ssfpack\ssfpack_ex.h>


// Adding randomly generated Regression terms as time-varying parameters
//  to the model      CK software program.


main()
{
   decl mPhi, mOmega, mSigma, mJPhi=<>;
   GetSsfStsm
   (<CMP_IRREG, 1, 0, 0, 0;
     CMP_TREND,.4, 2, 0, 0>,
   &mPhi, &mOmega, &mSigma);
   AddSsfReg(rann(3,20), &mPhi, &mOmega, &mSigma, &mJPhi);
   format("%#6.2g");
   println("System Matrices for adding regression terms ");
   println("                              ");
   print("Phi = ",mPhi, " Omega = ",mOmega, " Sigma = ",mSigma);
   println("JPhi = ",mJPhi);

}
```

Adding the random Regression data to System matrices that have just been specified.

System matrices for adding regressors as time-varying parameters.

```
--------------- Ox at 22:44:03 on 09-Nov-2009 ---------------

Ox Professional version 6.00 (Windows/U/MT) (C) J.A. Doornik, 1994-2009
System Matrices for adding regression terms

Phi =
     1.0   0.00   0.00   0.00   0.00
    0.00    1.0   0.00   0.00   0.00
    0.00   0.00    1.0   0.00   0.00
    0.00   0.00   0.00    1.0    1.0
    0.00   0.00   0.00   0.00    1.0
    0.00   0.00   0.00    1.0   0.00
 Omega =
    0.00   0.00   0.00   0.00   0.00   0.00
    0.00   0.00   0.00   0.00   0.00   0.00
    0.00   0.00   0.00   0.00   0.00   0.00
    0.00   0.00   0.00   0.00   0.00   0.00
    0.00   0.00   0.00   0.00   0.16   0.00
    0.00   0.00   0.00   0.00   0.00    1.0
 Sigma =
    -1.0   0.00   0.00   0.00   0.00
    0.00   -1.0   0.00   0.00   0.00
    0.00   0.00   -1.0   0.00   0.00
    0.00   0.00   0.00   -1.0   0.00
    0.00   0.00   0.00   0.00   -1.0
    0.00   0.00   0.00   0.00   0.00
JPhi =
    -1.0   -1.0   -1.0   -1.0   -1.0
    -1.0   -1.0   -1.0   -1.0   -1.0
    -1.0   -1.0   -1.0   -1.0   -1.0
    -1.0   -1.0   -1.0   -1.0   -1.0
    -1.0   -1.0   -1.0   -1.0   -1.0
    0.00    1.0    2.0   -1.0   -1.0
```

The Jphi matrix contains -1 except where time varying parameters are specified.

Trend-cycle models

Order of trend

```
#include <oxstd.h>
#include <c:\Program Files\OxMetrics51\Ox\packages\ssfpack\ssfpack_ex.h>

//   Trend Cycle Model   CK page 26


main()
{
   decl mPhi, mOmega, mSigma;
   GetSsfStsm
   (<CMP_IRREG, 1.0, 0, 0,0;      // sd=1  this is the irregular component denom in q
     CMP_TREND, 0.4, 3, 0, 0;     // 3rd order trend    sd=.4
     CMP_BWCYC, .6, .9, .3, 2>,   // Butterworth filter for order 2, sd=.6, damping=.9, freq=.3
   &mPhi, &mOmega, &mSigma);
   format ("%#5.2g");
   println("Trend-cycle  CK p.26 ");
   print("=============================================");
   println("                              ");


   print("Phi = ",mPhi, " Omega = ",mOmega, " Sigma = ",mSigma);

}
```

Trend-cycle system matrices

```
--------------- Ox at 21:43:55 on 09-Nov-2009 ---------------

Ox Professional version 6.00 (Windows/U/MT) (C) J.A. Doornik, 1994-2009
Trend-cycle  CK p.26
=============================================
Phi =
  1.0  1.0 0.00 0.00 0.00 0.00 0.00
 0.00  1.0  1.0 0.00 0.00 0.00 0.00
 0.00 0.00  1.0 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.19 0.23  1.0 0.00
 0.00 0.00 0.00-0.23 0.19 0.00  1.0
 0.00 0.00 0.00 0.00 0.00 0.19 0.23
 0.00 0.00 0.00 0.00 0.00-0.23 0.19
  1.0 0.00 0.00  1.0 0.00 0.00 0.00
 Omega =
 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.16 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.33 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00 0.33 0.00
 0.00 0.00 0.00 0.00 0.00 0.00 0.00  1.0
 Sigma =
 -1.0 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 -1.0 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 -1.0 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.47 0.000.074-0.093
 0.00 0.00 0.00 0.00 0.470.0930.074
 0.00 0.00 0.000.0740.093 0.36 0.00
 0.00 0.00 0.00-0.0930.074 0.00 0.36
 0.00 0.00 0.00 0.00 0.00 0.00 0.00
```

Interventions

Just as regressors can be added to the model, so can dummy variable identifying additive outliers or level shifts. Several adjacent outliers can define outlier patches.

Adding Regressors

code from Zivot and Wang,(2005), pp. 526ff

When regressors are added to a local level model, the time-varying level serves as a constant. Therefore, we do not add a column of ones to them to avoid unnecessary multicollinearity Ibid, 28.

```
     X.mat = cbind(1, as.matrix(seriesData(excessReturns.ts[,"SP500"])))
     Phi.t = rbind(diag(2), rep(0,2))
     Phi.t
     Omega = diag(c((.01^2),(.05^2),(.1)^2))
     Omega
     J.Phi= matrix(-1,3,2)      # time-varying parameter J matrix
     J.Phi[3,1]=1
     J.Phi[3,2]=2
     J.Phi
     Sigma= - Phi.t
     mX=X.mat
     mX        # construction of design matrix

     #mapping matrices into SsfPack
     ssf.tvp.capm = list(mPhi=Phi.t,mOmega=Omega, mJPhi=J.Phi,mSigma=Sigma,
              mX=X.mat)
     ssf.tvp.capm

#  let's simulate the model
#  Simulation


#   first check it
     ssf.tvp.capm       // This checks to see whether your code is syntactically correct.
```

Non-parametric cubic splines for smoothing

Nonparametric cubic splines are smoothers used to extract signal from noise. They are designed to capture the nonlinearity of a function. These may be added as regressors to define a function or process that serves as an explanatory variable in a model.

If we have a stationary error process, such that $y_t = \mu_t + \varepsilon_t$. *We are trying to find a nonlinear or piecewise function* μ_t *for which*

$$argmin = \sum_{t=1}^{T} \left(y_t - \mu_t \right)^2 + \lambda \sum_{t=1}^{n} \left(\Delta^2 \mu_t \right)^2$$

where

the term on the far right = penalty function (Durbin and Koopman (2001, 61).

Basic structural model

It has a level, a slope, and a seasonal component

$$y_t = \mu_t + \beta_t + \gamma_t + \psi_t$$

$$\begin{aligned}
&where \\
&\mu_t = unobserved\ trend\ (level)\ component \\
&\beta_t = unobserved\ slope\ component \\
&\gamma_t = unobserved\ seasonal\ component \\
&\xi_t = unobserved\ irregular\ component
\end{aligned}$$

The GetSsfStsm function

If we provide the input of what components we wish to have in our model this function in SsfPack (in Ox or in S-Plus) will construct our system matrices for us.

The system matrices are the model matrices which stack the state equation atop the measurement equation. They are the Phi, the Omega, and the Sigma matrices.

GetSsfStsm in S-Plus

```
##################  Unobserved component formulation ###################################################
args(GetSsfStsm)

################  The local level model ##############################################################

ssf.stsm = GetSsfStsm(irregular=1, level=.5)     # irregular=1 and level = .5 specify sigma_epsilon = 1
                                                 #  and sigma_eta=1
class(ssf.stsm)
names(ssf.stsm)
ssf.stsm      # displays the system matrices of the state space local level model

args(GetSsfStsm)
names(GetSsfStsm)
```

```
> names(GetSsfStsm)
 [1] "irregular"     "level"         "slope"         "trend"         "seasonalDummy" "seasonalTrig"  "seasonalHS"
 [8] "BWcycle"       "cycle0"        "cycle1"        "cycle2"        "cycle3"        "cycle4"        "cycle5"
[15] "cycle6"        "cycle7"        "cycle8"        "cycle9"        "AR1"           "AR2"           ""
```

System matrices for local level model

```
ssf.stsm = GetSsfStsm(irregular=1, level=.5)      # irregular=1 and level = .5 specify sigma_epsilon = 1
                                                  #  and sigma_eta=1
class(ssf.stsm)
names(ssf.stsm)
ssf.stsm      # displays the system matrices of the state space local level model

args(GetSsfStsm)
names(GetSsfStsm)

ssf.stsm
```

```
> ssf.stsm
$mPhi:
     [,1]
[1,]    1
[2,]    1

$mOmega:
     [,1] [,2]
[1,] 0.25    0
[2,] 0.00    1

$mSigma:
     [,1]
[1,]   -1
[2,]    0
```

Local level model with stochastic regressors (time-varying parameters)

we could treat the parameters as random walks (Zivot and Wang, 2005, 533)

$$\alpha_{t+1} = T_t \alpha_t + H_t \eta_t$$

$$y_t \quad = x_t' \beta_t + \quad \xi_t$$

with

$$H_t = \begin{bmatrix} \sigma_{\beta_1} \\ \sigma_{\beta_2} \\ \vdots \\ \sigma_{\beta_k} \end{bmatrix}$$

in this case the exogenous series are treated as random walks so

$$\beta_{i,t+1} = \beta_{i,t} + \sigma_{i,\beta_t}$$

SsfPack code for reading the Norwegian traffic fatalities data

```
}

main()
{
    decl data;
    data = loadmat("NorwayFinland.txt")'; // load data, transpose

    print("\nNorway 1970-2003");
    print("\n----------------\n");
    s_mY = log(data[1][]);        // log 1970-2003
    s_cT = columns(s_mY);        // no of observations

    MaxLik();
    DrawComponents(s_mY);
}
```

Ox code for setting up a stochastic local level model

Commandeur and Koopman code snippet

chapter2-1&2-2.ox - C:\Program Files\OxMetrics6\Ox\packages\ssfpack\CKbook\Chapter_2\c...

```
   ***************************************************************/

#include <oxstd.h>
#include <oxdraw.h>
#import  <maximize>
#include <c:\Program Files\OxMetrics51\Ox\packages\ssfpack\ssfpack_ex.h>

static decl s_mY, s_cT;          // data (1 x n) and n
static decl s_mStsm, s_vVarCmp;  // matrices for state space model
static decl s_dVar;              // scale factor
static decl s_vPar;              // parameter vector of model
                               //  Determination of the level variance as stochas
static decl s_iLvlVar = 0;   // 0 = stochastic level; -1 = deterministic level
static decl s_asCmps;        // string array of component names

SetStsmModel(const vP)
{
    // map to local level model
    s_mStsm = < CMP_LEVEL,  0.5,     0,   0;
                CMP_IRREG,  1,       0,   0>;
    // change BFGS parameters into error variances
    decl vr = exp(2.0 * vP);
    // s_vVarCmp is used to update diagonal(Omega)
    if (s_iLvlVar != -1)
        //           level irregular
        s_vVarCmp = vr[0]| vr[1];
    else
```

Ox code for Local level model

```
}
InitialPar()
{
    decl dlik, dvar, vp;

    if (s_iLvlVar != -1)                // diffuse prior
    {
        s_asCmps = {"level      ", "irregular "};    //local level model
        vp = log(<0.5; 1>);
    }
    else
    {
        s_asCmps = {"irregular "};
        vp = log(<1>);
    }

    SetStsmModel(vp);        // map vP to local level model
    LogLikStsm(s_mY, &dlik, &dvar);
    // scale initial estimates by scale factor
    return vp + 0.5 * log(dvar);
}
Likelihood(const vP, const pdLik, const pvSco, const pmHes)
{                               // arguments dictated by MaxBFGS()
    decl ret_val;

    SetStsmModel(vP);        // map vP to local level model
    ret_val = pvSco ? LogLikScoStsm(s_mY, pdLik, pvSco)
                : LogLikStsm(s_mY, pdLik, &s_dVar);
    return ret_val;      // 1 indicates success, 0 failure
}
```

Code snippet

```
MaxLik()
{
    decl vp, dlik, ir;

    vp = InitialPar();          // initialise unconstrained BFGS parameters
    print("\ninitial values BFGS parameters",vp);
    print("\n");
    MaxControl(50, 1, 1);       // start iterations BFGS algorithm
    ir = MaxBFGS(Likelihood, &vp, &dlik, 0, FALSE);

    println("\n", MaxConvergenceMsg(ir),
          " using analytical derivatives",
          "\n(1/n) Log-likelihood = ", "%.8g", dlik,
          "; n = ", s_cT, ";");

    // set up system matrices and compute AIC
    decl mphi, momega, msigma, daic, i;
    GetSsfStsm(s_mStsm, &mphi, &momega, &msigma);
    daic = (-2*dlik*s_cT) + (2*(rows(vp)+columns(mphi)));
    println("\nAkaike Information Criterion = ", daic/s_cT);

    s_vPar = vp;
    print("\nparameter estimates (unconstrained)");
    for(i=0;i<=rows(vp)-1; i++)
        print("\n    ", s_asCmps[i], vp[i]);
    print("\n\nerror variance estimates");
    for(i=0;i<=rows(vp)-1; i++)
        print("\n    ", s_asCmps[i], exp(2.0 * vp[i]));
    print("\n");
    println("Printing system matrices for local level model");
```

Output of local level model for Norwegian traffic fatalities

(data from Commandeur and Koopman (2007)

```
Ox Professional version 6.00 (Windows/U/MT) (C) J.A. Doornik, 1994-2009

Norway 1970-2003
----------------

initial values BFGS parameters
      -3.1656
      -2.4725

it0    f=     0.7755300 df=    0.1692 e1=    0.5780 e2=  0.006215 step=1
it1    f=     0.8205218 df=    0.1248 e1=    0.4053 e2=  0.009751 step=1
it2    f=     0.8464840 df=   0.02166 e1=   0.06664 e2=   0.01080 step=1
it3    f=     0.8468296 df=  0.005806 e1=   0.01800 e2= 0.0007435 step=1
it4    f=     0.8468620 df= 0.0003184 e1= 0.0009332 e2= 0.0003626 step=1
it5    f=     0.8468622 df=1.947e-005 e1=5.706e-005 e2=2.895e-005 step=1
BFGS: Strong convergence

Strong convergence using analytical derivatives
(1/n) Log-likelihood = 0.84686222; n = 34;

Akaike Information Criterion = -1.51725

parameter estimates (unconstrained)
    level     -2.67982
    irregular -2.86173

error variance estimates
    level     0.0047026
    irregular 0.00326838
```

System Matrices

```
Printing system matrices for local level model
Phi =
       1.0000
       1.0000
 Omega =
      0.25000      0.00000
      0.00000       1.0000
 Sigma =
      -1.0000
      0.00000

mKF[][0:4]
      0.00000    -0.049415    -0.098486     0.010679  -0.00050003
       1.0000      0.70920      0.68234      0.67961      0.67933
      0.00000       88.973       97.192       98.028       98.114
      0.00000       1.0000       1.0000       1.0000       1.0000

          lag   autocorrelation
       1.0000     -0.12733
       2.0000    -0.012441
       3.0000      0.10949
       4.0000     -0.10540
       5.0000     -0.13824
       6.0000     -0.22535
       7.0000     -0.15301
       8.0000    -0.047786
       9.0000     0.042202
       10.000     -0.10378

95%-confidence limit = 0.342997
```

Initial conditions depend on prior distribution

To indicate a diffuse distribution and or a noninformative prior, the variance of the prior is flat and almost infinite. This means that the precision of such knowledge is the inverse or reciprocal of the variance. The precision -> 0 as the variance-> infinity.

Problems of estimation arise when you approach the perilous precipice (boundary) of the parameter space. Estimates tend to break down at such extremes.

Therefore, we use in our computers approximations. Infinity is therefore represented by a very large number, such as 10^7 . We suggest such a condition by assigning a value of -1 to a parameter for an initial condition. If parameters are mean-centered, the initial value of their means can easily be zero.

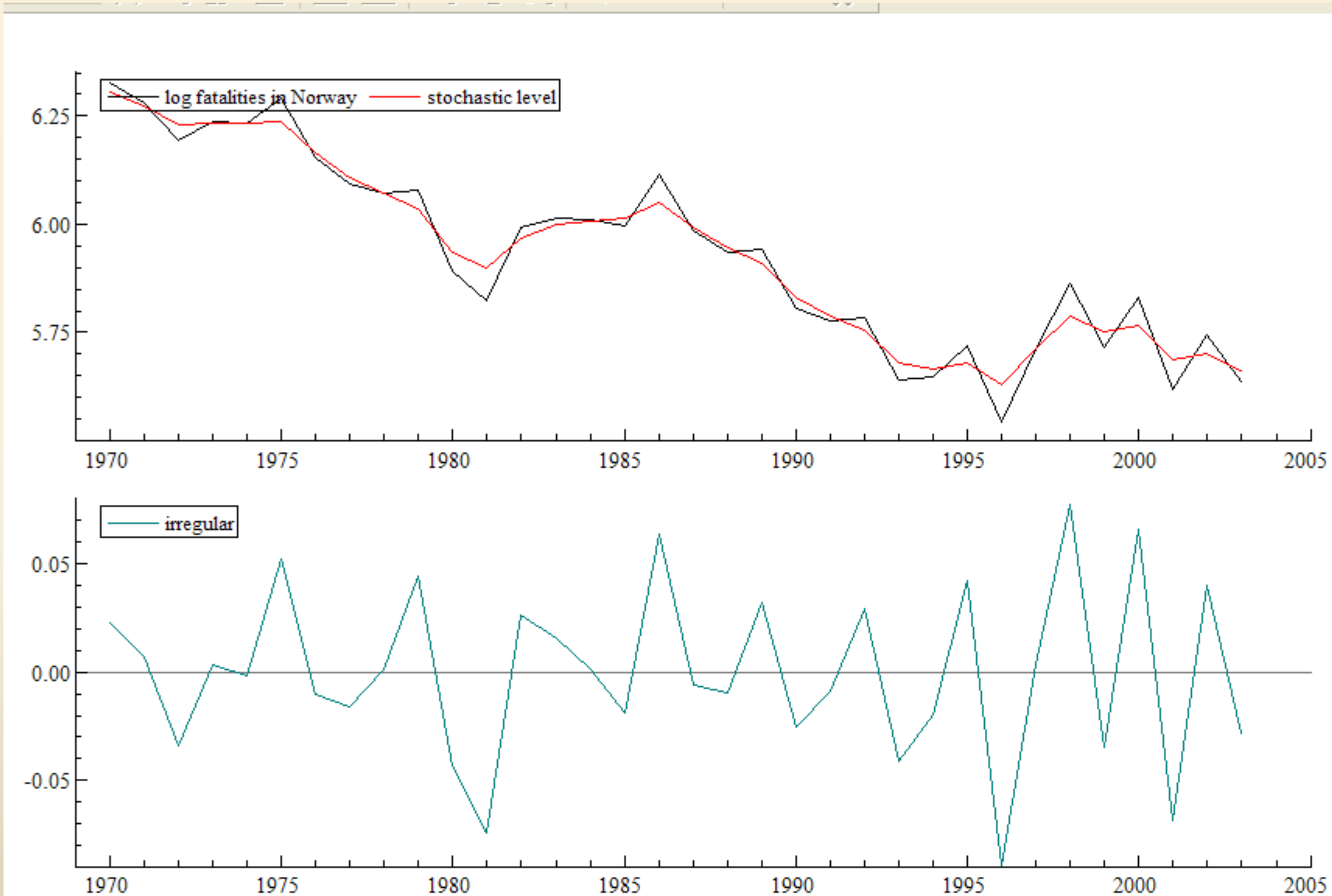
Time-varying parameters

Many simple models can be defined by specifying the mPhi, mOmega, and mSigma matrices.

However, sometimes parameters vary over time. They may be random coefficients.

To indicate such parameters, we use J matrices. Instead of mPhi, the matrix would be called, mJ_Phi. This would indicate the presence of a non-constant system matrix for mPhi.

Graphical output of Model of Norwegian traffic fatalities



Local linear Trend Model

$$\mu_{t+1} = d_t + T\mu_t + \beta_t + \underset{mx1}{\eta_t} \qquad \eta_t \sim NID(0, \sigma^2_{\eta_t}) = (0, \underset{mxr}{H_t}\ \underset{rx1}{\varepsilon_t})$$

$$\beta_{t+1} = \quad \beta_t + \zeta_t \qquad \zeta_t \sim NID(0, \sigma^2_{\zeta_t}) = (0, \underset{mxr}{H}\ \varepsilon_t)$$

$$y_t \quad = c_t + \underset{Nxm}{Z_t}\ \mu_t + \xi_t \qquad \xi_t \sim NID(0, \sigma^2_{\xi_t}) = (0, \underset{Nxr}{G}\ \varepsilon_t)$$

Local linear Trend model System matrices

$$\alpha_{t+1} = \begin{pmatrix} \mu_{t+1} \\ \beta_{t+1} \\ y_t \end{pmatrix}_{m\,x\,1} = state\ vector\ (unobserved\ \ factor)$$

$$\delta_t = \begin{pmatrix} d_t \\ c_t \end{pmatrix} = mean\ matrix(if\ \ model\ is\ not\ mean\ centered)$$

$$\Phi_t = \begin{pmatrix} T_t \\ Z_t \end{pmatrix}_{(m+N)\,x\,m} = mPhi\ matrix$$

$$\Sigma_t = \begin{pmatrix} HH' & HG' \\ GH' & GG' \end{pmatrix}_{(m+1)\,x\,m} = mSigma\ matrix$$

$assumed\ \ that\ \mathrm{cov}\,ariance = \mathbf{0}\ \ so\ mSigma\ matrix\ is$

$defined\ \ by\ \ it's\ \ principle\ diagonal$

$where$

$m = dimension\,of\ state\ vector$

$N = number\ \ of\ \ variables$

$n = number\ \ of\ \ observations$

$r = dimension\ of\ error\ \ vector$

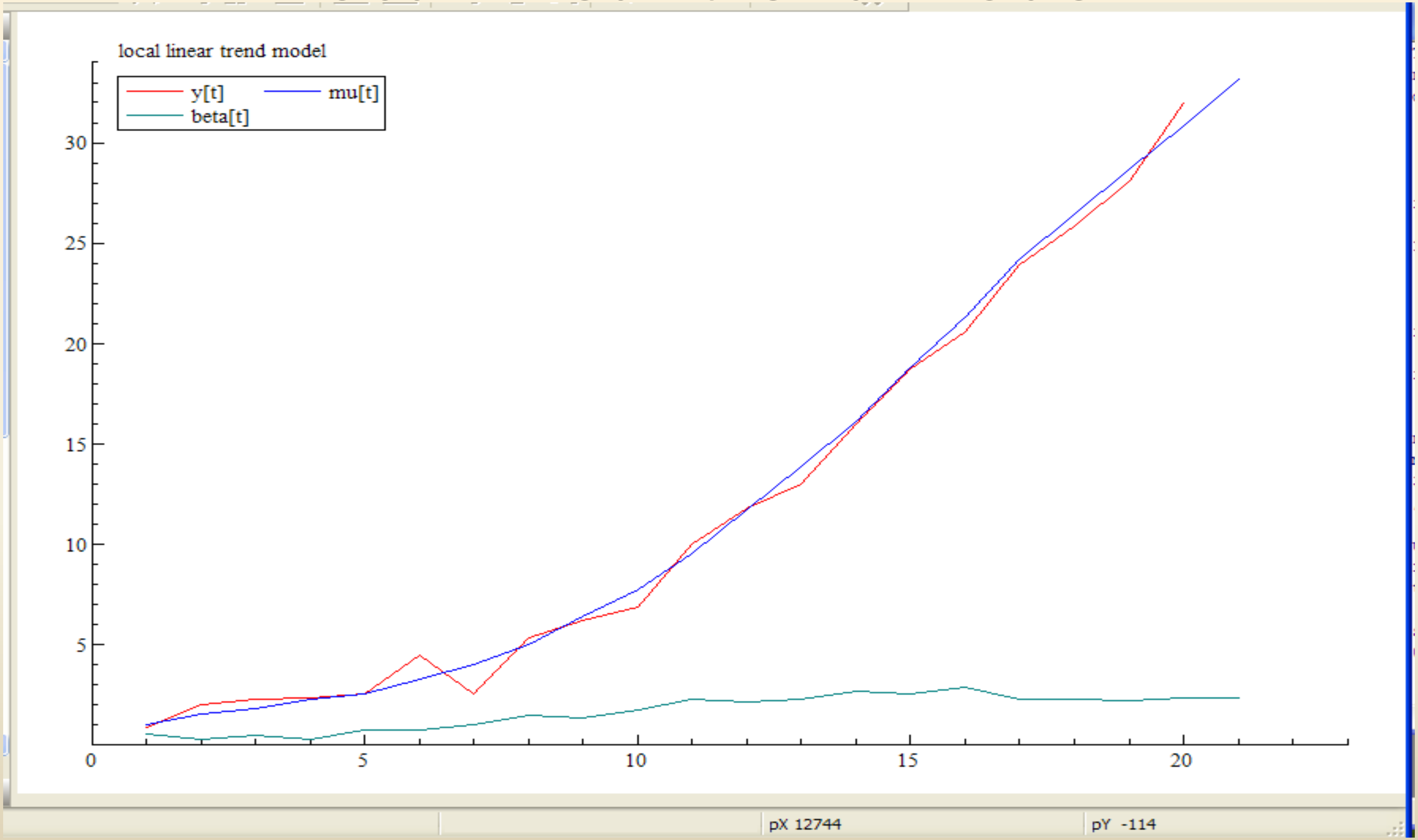
Ox Code for local linear trend model

```
#include <oxstd.h>
#include <oxdraw.h>
#include <oxfloat.h>
#include <packages/ssfpack/ssfpack_ex.h>

main()
{
    decl mPhi = <1,1;0,1;1,0>;
    decl mOmega = diag(<0,0.1,1>);
    decl mSigma = <0,0;0,0;1,.5>;       // Note that Q is zero

    decl mr = sqrt(mOmega) * rann(3, 21);
    decl md = SsfRecursion(mr, mPhi, mOmega, mSigma);
    decl mYt = md[2][1:] ~ M_NAN;       // 20 observations

    print("Generated data (t=10) for local linear trend model",
        "%c", {"mu[t+1]","beta[t+1]","y[t]"}, md[][10]');
    DrawTitle(0, "local linear trend model");
    DrawTMatrix(0, mYt | md[:1][],
        {"y[t]", "mu[t]","beta[t]"}, 1, 1, 1);        // local linear trend model
    ShowDrawWindow();
}
```

Ox Output

```
--------------- Ox at 23:30:40 on 18-Nov-2009 ---------------

Ox Professional version 6.00 (Windows/U/MT) (C) J.A. Doornik, 1994-2009
Generated data (t=10) for local linear trend model
      mu[t+1]    beta[t+1]        y[t]
       9.4880       2.2385       6.8327
```

Ox graphical output for local linear trend model



Defining the system matrices and specifying the model

It can be done without reference to ARIMA models, as we have already shown.

We will now provide examples of how these models may be formulated in an ARIMA framework as well.

We shall give examples of both,   with Ox and S-Plus.

Ox code specifying an AR(1) model

```
#include <oxstd.h>
#include <packages/ssfpack/ssfpack_ex.h>

main()
{
    decl mPhi, mOmega, mSigma;
    format("%5.1f");
    GetSsfArma
    (<0.6>, <>, sqrt(0.9), &mPhi, &mOmega, &mSigma);
    print(" An AR(1) model  ");
    print("Phi =",mPhi, "Omega =",mOmega, "Sigma =",mSigma);
}
```

AR(1) system matrix output

```
-------------- Ox at 23:55:19 on 18-Nov-2009 --------------

Ox Professional version 6.00 (Windows/U/MT) (C) J.A. Doornik, 1994-2009
 An AR(1) model  Phi =
  0.6
  1.0
Omega =
  0.9  0.0
  0.0  0.0
Sigma =
  1.4
  0.0
```

Local Level model with stochastic regressors with AR(2) errors

$$\begin{pmatrix} \alpha_{t+1} \\ y_t \end{pmatrix} = \begin{pmatrix} T^* \\ Z_t^* \end{pmatrix} \alpha^* + \begin{pmatrix} H\eta_t \\ \mathbf{0} \end{pmatrix}$$

$$T^* = \begin{pmatrix} T & \mathbf{0} \\ \mathbf{0} & I_k \end{pmatrix}$$

$$Z_t^* = \begin{pmatrix} \mathbf{1} & \mathbf{0} & \mathrm{L} & x_t' \end{pmatrix}$$

AR(2) Ox code

Koopman, Shephard, and Doornik (2008, 16)

```
#include <oxstd.h>
#include <packages/ssfpack/ssfpack_ex.h>

main()
{
    decl mPhi, mOmega, mSigma;
    format("%5.1f");
    GetSsfArma
    (<0.6,0.3>, <>, sqrt(0.9), &mPhi, &mOmega, &mSigma);
    print(" An AR(2) model  ");
    print("Phi =",mPhi, "Omega =",mOmega, "Sigma =",mSigma);
}
```

AR(2) system matrix output

```
--------------- Ox at 23:58:23 on 18-Nov-2009 ---------------

Ox Professional version 6.00 (Windows/U/MT) (C) J.A. Doornik, 1994-2009
 An AR(2) model  Phi =
  0.6  1.0
  0.3  0.0
  1.0  0.0
Omega =
  0.9  0.0  0.0
  0.0  0.0  0.0
  0.0  0.0  0.0
Sigma =
  3.7  1.0
  1.0  0.3
  0.0  0.0
```

Ox Code specifying an MA1 model

```
#include <oxstd.h>
#include <packages/ssfpack/ssfpack_ex.h>

main()
{
    decl mPhi, mOmega, mSigma;
    format("%5.1f");
    GetSsfArma
    (<>, <0.8>, sqrt(0.9), &mPhi, &mOmega, &mSigma);
    print(" An MA(1) model\n");
    print("Phi =",mPhi, "Omega =",mOmega, "Sigma =",mSigma);
```

MA 1 system matrix output

```
--------------- Ox at 23:58:23 on 18-Nov-2009 ---------------

Ox Professional version 6.00 (Windows/U/MT) (C) J.A. Doornik, 1994-2009
 An AR(2) model  Phi =
  0.6  1.0
  0.3  0.0
  1.0  0.0
Omega =
  0.9  0.0  0.0
  0.0  0.0  0.0
  0.0  0.0  0.0
Sigma =
  3.7  1.0
  1.0  0.3
  0.0  0.0
```

MA(2) system matrix output

```
#include <oxstd.h>
#include <packages/ssfpack/ssfpack_ex.h>

main()
{
    decl mPhi, mOmega, mSigma;
    format("%5.1f");
    GetSsfArma
    (<>, <0.6,0.3>, sqrt(0.9), &mPhi, &mOmega, &mSigma);
    print(" An MA(2) model\n");
    print("Phi =",mPhi, "Omega =",mOmega, "Sigma =",mSigma);
}
```

MA2 system matrix output

```
-------------- Ox at 00:05:11 on 19-Nov-2009 --------------

Ox Professional version 6.00 (Windows/U/MT) (C) J.A. Doornik, 1994-20O
 An MA(2) model
Phi =
  0.0  1.0  0.0
  0.0  0.0  1.0
  0.0  0.0  0.0
  1.0  0.0  0.0
Omega =
  0.9  0.5  0.3  0.0
  0.5  0.3  0.2  0.0
  0.3  0.2  0.1  0.0
  0.0  0.0  0.0  0.0
Sigma =
  1.3  0.7  0.3
  0.7  0.4  0.2
  0.3  0.2  0.1
  0.0  0.0  0.0
```

An ARMA(2,1) model

Ox code from Koopman, Shephard, and Doornik (2008, 16)

```
#include <oxstd.h>
#include <packages/ssfpack/ssfpack_ex.h>

main()
{
    decl mPhi, mOmega, mSigma;
    format("%5.1f");
    GetSsfArma
    (<0.6,0.2>, <-0.2>, sqrt(0.9), &mPhi, &mOmega, &mSigma);
    println(" An ARMA(2,1) model");
    print("Phi =",mPhi, "Omega =",mOmega, "Sigma =",mSigma);
}
```

ARMA(2,1) model output

```
--------------- Ox at 00:11:43 on 19-Nov-2009 ---------------

Ox Professional version 6.00 (Windows/U/MT) (C) J.A. Doornik, 1994-2009
 An ARMA(2,1) model
Phi =
  0.6  1.0
  0.2  0.0
  1.0  0.0
Omega =
  0.9 -0.2  0.0
 -0.2  0.0  0.0
  0.0  0.0  0.0
Sigma =
  1.6  0.0
  0.0  0.1
  0.0  0.0
```

ARIMA(2,1,1) model specification

Koopman, Shephard, and Doornik(2008, 18) .

```
#include <oxstd.h>
#include <packages/ssfpack/ssfpack_ex.h>

main()
{
    decl mPhi, mOmega, mSigma;
    GetSsfSarima
    (1,<0.6,0.3>,<-0.2>,sqrt(0.9), &mPhi, &mOmega, &mSigma);
    println("ARIMA (2,1,1)  model specification");
    print("Phi =",mPhi, "Omega =",mOmega, "Sigma =",mSigma);
}

```

System matrices for an ARIMA(2,1,1) model

```
Ox Professional version 6.00 (Windows/U/MT) (C) J.A. Doornik, 1994-2009
ARIMA (2,1,1)  model specification
Phi =
       1.0000       1.0000      0.00000
      0.00000      0.60000       1.0000
      0.00000      0.30000      0.00000
       1.0000       1.0000      0.00000
Omega =
      0.00000      0.00000      0.00000      0.00000
      0.00000      0.90000     -0.18000      0.00000
      0.00000     -0.18000     0.036000      0.00000
      0.00000      0.00000      0.00000      0.00000
Sigma =
      -1.0000      0.00000      0.00000
      0.00000       2.5988      0.41112
      0.00000      0.41112      0.26989
      0.00000      0.00000      0.00000
```

Adding data containing exogenous series to the model using SsfPack

```
#############  State Space Form = local level + stochastic regression effects
#   Initial state:   alpha = 0
#                    P = k*I
#       assumption is that prior=diffuse
#    sigma = <-1, 0; 0, -1; 0, 0>
#

#    code from Zivot and Wang,(2005), pp. 526ff

       # Construction of the system matrices for time-varying parameters

     X.mat = cbind(1, as.matrix(seriesData(excessReturns.ts[,"SP500"])))
     Phi.t = rbind(diag(2), rep(0,2))
     Phi.t
     Omega = diag(c((.01^2),(.05^2),(.1)^2))
     Omega
     J.Phi= matrix(-1,3,2)      # time-varying parameter J matrix
     J.Phi[3,1]=1
     J.Phi[3,2]=2
     J.Phi
     Sigma= - Phi.t
     mX=X.mat
     mX        # construction of design matrix

     #mapping matrices into SsfPack
     ssf.tvp.capm = list(mPhi=Phi.t,mOmega=Omega, mJPhi=J.Phi,mSigma=Sigma,
              mX=X.mat)
     ssf.tvp.capm
```

How the Kalman filter functions

- The Kalman filter evaluates moments of the state vector over time.
- Filtering is a one-step-ahead forecast of the mean and variance plus a regression on the innovation to provide a correction at one-lag of this process. Hence, there is iterative correction over time.

To estimate the mean and variance of the state vector

The Kalman filter adds the data to the structure specified by the system matrices and uses the data to recursively compute the innovations that will be used to correct the one-step-ahead expectation of the state first and second moments: the state mean and state variance:

$$\alpha_{t+1} = E(\alpha_t \mid Y_t)$$

$$P_{t+1} = \text{cov}(\alpha_t \mid Y_t)$$

Kalman filter ALgorithm

- A recursive algorithm proceeding 1 step at a time.:

$$v_t = y_t - c_t - Z_t\alpha_t \quad \textit{innovations are computed}$$

$$F_t = \text{var}(v_t) = Z_t P_t Z_t + G_t G_t' \quad \textit{innovation variance is computed;}$$

subject to eigenvalue decomposition for further analysis, so Z, P, G *and* G' *become defined.*

$\kappa =$ *Kalman gain can be computed from*

$$\kappa = (T_t P_t Z_t' + H_t G_t')F^{-1} \quad \textit{so } T \textit{ is now known.}$$

$$\alpha_{t+1} = d_t + T\alpha_t + \kappa v_t \quad (\textit{if not mean centered}; d_t = \mathbf{0} \textit{ if mean-centered})$$

$$P_t = T_t P_t T_t' + H_t H_t' \quad \textit{can finally be computed.}$$

The state moments are estimated.

Convergence problems

- If $|F|=0$, or when there is not enough computer memory, this procedure may not converge.
- It has to be able to invert F. If F -> large, the speed will degrade.

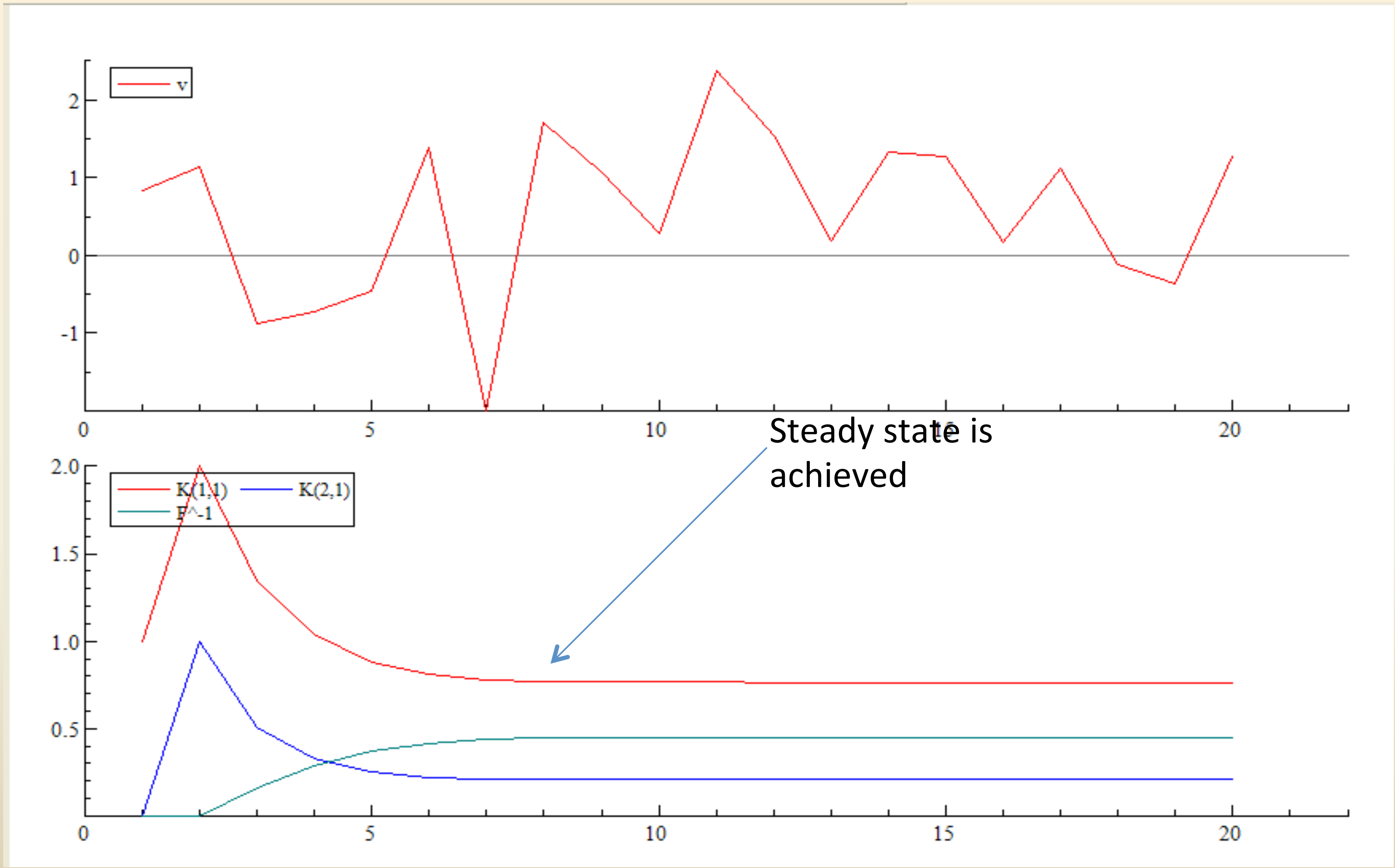
Kalman filter without diffuse initialization Md->md-> Recursion

```
#include <oxstd.h>
#include <oxdraw.h>
#include <oxfloat.h>
#include <packages/ssfpack/ssfpack_ex.h>                    // Kalman filter
                                                            // withhout diffuse initialization
main()                                                      // as our dgp is stationary
{
    decl mPhi = <1,1;0,1;1,0>;
    decl mOmega = diag(<0,0.1,1>);
    decl mSigma = <0,0;0,0;1,.5>;          // Note that Q is zero

    decl mr = sqrt(mOmega) * rann(3, 21);      // mr is designed to input into md for recursion
    decl md = SsfRecursion(mr, mPhi, mOmega, mSigma);
    println("mr = ",mr);
    println("md = ",md);
    decl mYt = md[2][1:];        // 20 observations of data
    println("mYt = ",mYt);
    decl mKF = KalmanFil(mYt, mPhi, mOmega);
    print("mKF\' (t=10)","%c",{"v","K(1,1)","K(2,1)","F^-1"},
           mKF[][9]');

    DrawTMatrix(0, mKF[0][], {"v"},1,1,1);
    DrawTMatrix(1, mKF[1:][], {"K(1,1)","K(2,1)","F^-1"},1,1,1);
    ShowDrawWindow();
}
```

Data matrix and Kalman Filter output at t=10

```
mYt =
     0.83317      1.9783      2.2372      2.3553      2.5553      4.4531
      2.5337      5.3342      6.2209      6.8327      9.9575      11.804
      13.006      15.987      18.685      20.571      23.932      25.869
      28.095      31.990
mKF' (t=10)
           v      K(1,1)      K(2,1)       F^-1
     0.27618     0.76491     0.21161    0.44669
```

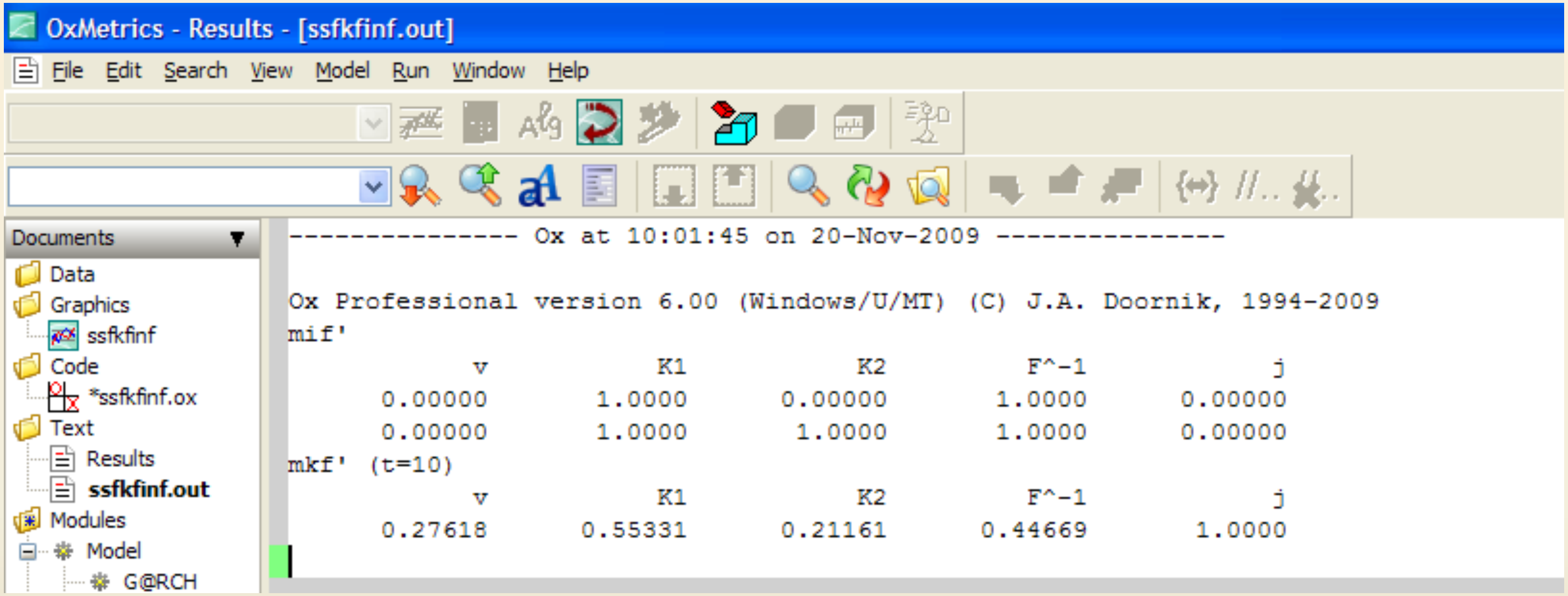
Conventional Kalman filter output



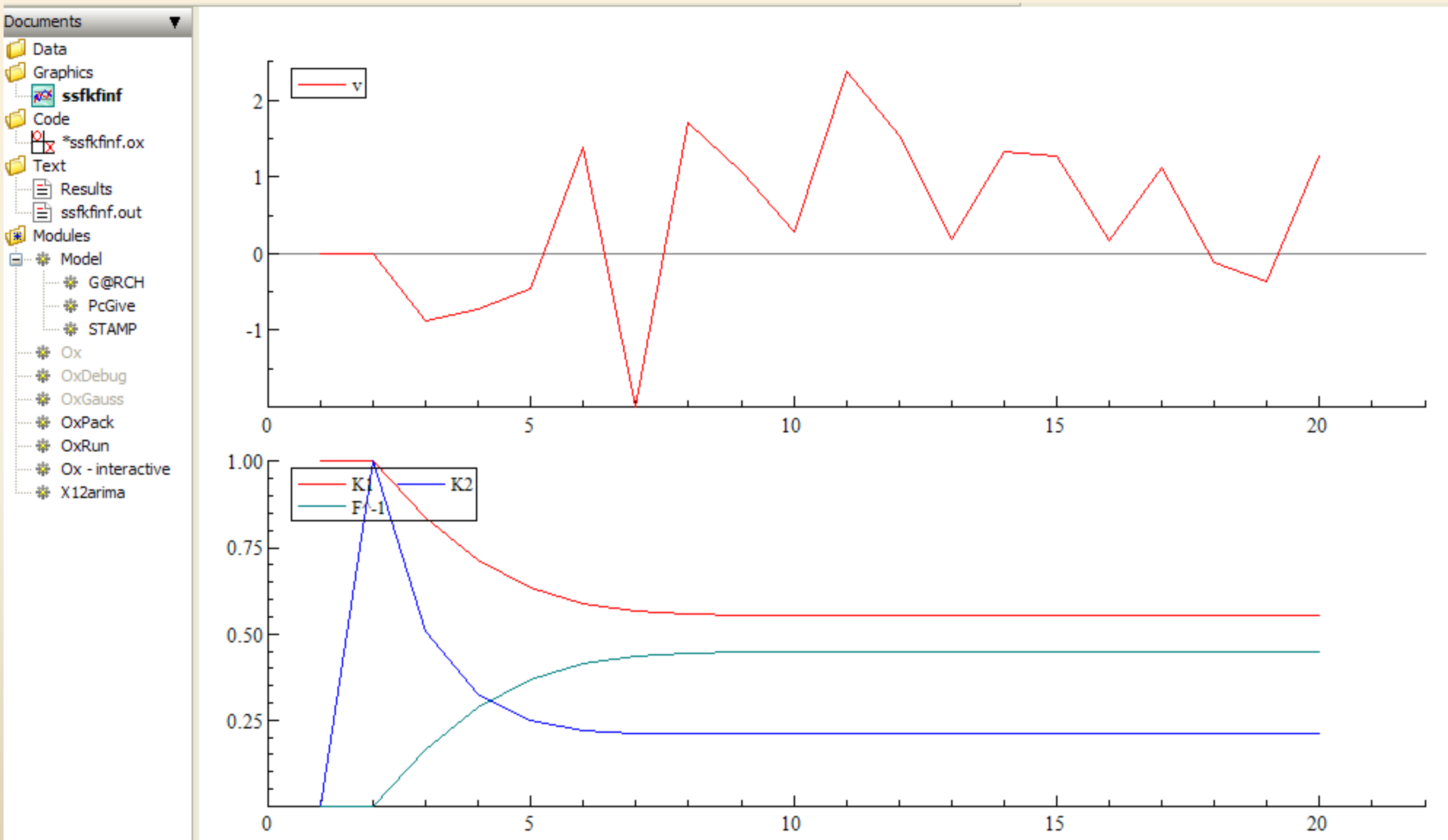
Kalman filter with diffuse initialization

```
#include <oxstd.h>
#include <oxdraw.h>
#include <oxfloat.h>
#include <packages/ssfpack/ssfpack_ex.h>

main()
{
    decl mphi = <1,1;0,1;1,0>;
    decl momega = diag(<0,0.1,1>);
    decl msigma = <0,0;0,0;1,.5>;        // Note that Q is zero

    decl mr = sqrt(momega) * rann(3, 21);
    decl md = SsfRecursion(mr, mphi, momega, msigma);
    decl myt = md[2][1:];        // 20 observations

    decl mif = KalmanInit(myt, mphi, momega);          // applies univariate KF to nonstationary local l
    decl mkf = KalmanFilEx(mif, myt, mphi, momega);    //  nonstationary time series model

    print("mif\'", "%c", {"v", "K1", "K2", "F^-1", "j"}, mif');
    print("mkf\' (t=10)", "%c", {"v", "K1", "K2", "F^-1", "j"}, mkf[][9]');

    DrawTMatrix(0, mkf[0][], {"v"}, 1, 1, 1);
    DrawTMatrix(1, mkf[1:3][], {"K1", "K2", "F^-1"}, 1, 1, 1);
    ShowDrawWindow();
}
```

Output from diffuse initialization



Graphical output from diffuse initialization applied to nonstationary data



Displaying the state vector

A function called mstate will generate the state vector after

After the data and mpred and the system matrices are combined in

Kalman Smoother

For signal extraction, for residual analysis, and auxiliary residual analysis, we need to

1. smooth the moments,
2. smooth the disturbances,
3. and smooth the states.

Moment Smoothing

ibid, 40; Durbin and Koopman, 2001,15-23

All smoothing equations depend on backward recursions based on:

$$e_t = F_t^{-1} v_t - \kappa_t^{'} r_t \qquad (Nx\mathbf{1}) \quad \textit{smoothing error} \quad e_t = \sigma_{\varepsilon_t}^2 u_t$$

$$D_t = F_t^{-1} - \kappa_t^{'} N_t \kappa_t \quad (NxN) \quad \textit{smoothing variance}$$

$$r_{t-1} = Z_t^{'} F_t^{-1} v_t + L_t^{'} r_t \quad (m\,x\mathbf{1}) \quad Var(r_t) = N_t$$

$$N_{t-1} = Z_t^{'} F_t^{-1} Z_t + L_t^{'} N_t L_t \quad (mxm)$$

$$L_t = T_t - K_t Z_t = \mathbf{1} - K_t = \frac{\sigma_\varepsilon^2}{F}$$

with initialization that $r_n = \mathbf{0}$ *and* $N_n = \mathbf{0}$ *for* $t = n, \ldots, \mathbf{1}$

Disturbance smoothing

Koopman, Shephard, and Doornik, 2008, 43

$$E(\eta_{t|} \mid Y_t) = \sigma_\eta^2 r_t \;\; = smoothed\ \ mean\ \ of\ \ state\ disturbance$$

$$Var(\eta_{t|} \mid Y_t) = \sigma_\eta^2 - \sigma_\eta^4 N_t \;\; smoothed\ \ variance\ \ of\ \ state$$

$$disturbance$$

$$E(u_t \mid Y_n) = E(H_t \varepsilon_t \mid Y_n) = H_t H_t^{'} r_t$$

$$E(G_t \varepsilon_t \mid Y_n) = G_t G_t{}' e_t$$

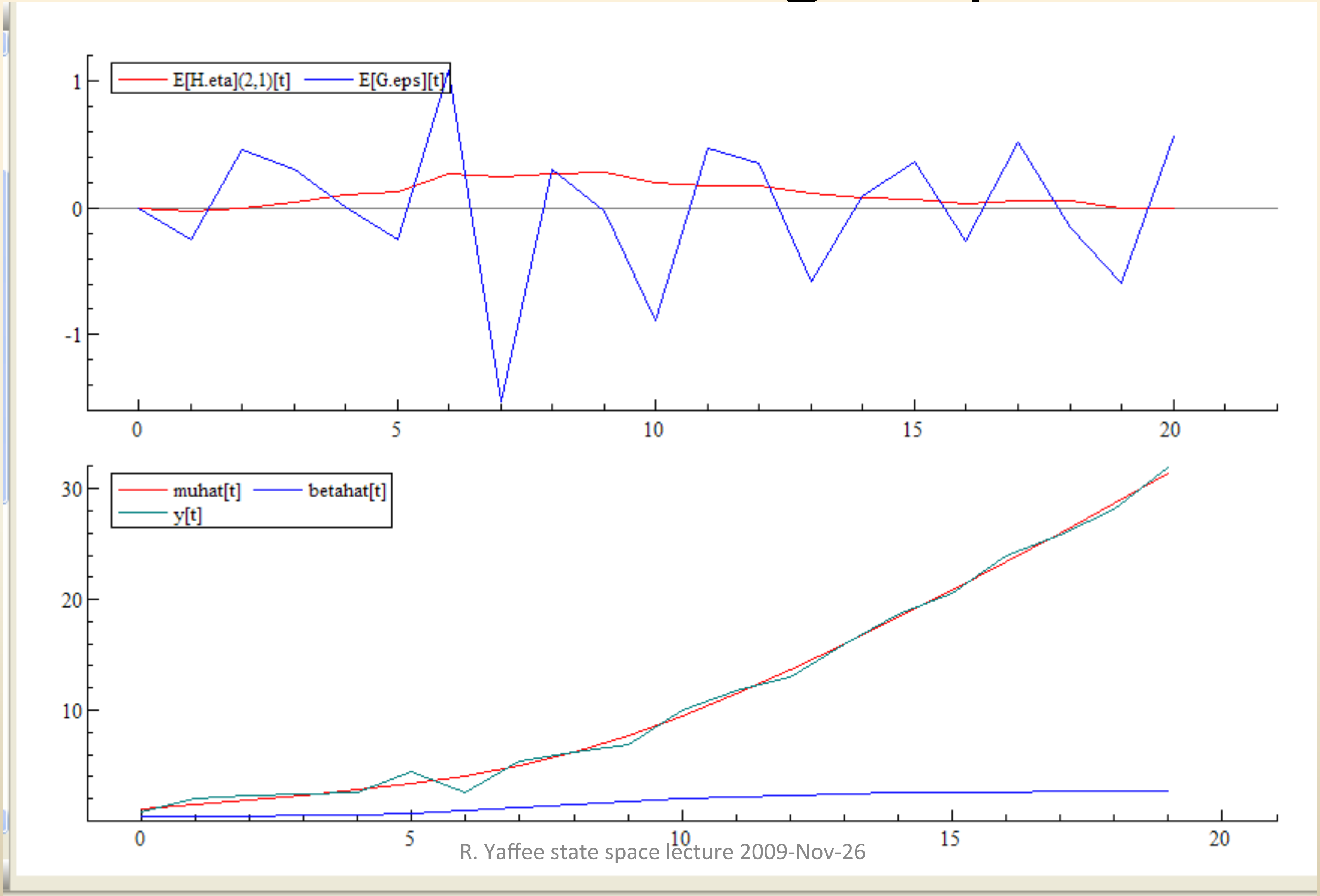
$$Var(u_t \mid Y_n) = Var(G_t \varepsilon_t \mid Y_n) = G_t G_t^{'} D_t G_t G_t^{'}$$

$$Var(H_t \varepsilon_t \mid Y_n) = H_t H_t^{'} N_t H_t H_t^{'}$$

State Smoothing

```
#include <oxdraw.h>
#include <oxfloat.h>
#include <packages/ssfpack/ssfpack_ex.h>

main()
{
    decl mPhi = <1,1;0,1;1,0>;
    decl mOmega = diag(<0,0.1,1>);
    decl mSigma = <0,0;0,0;1,.5>;       // Note that Q is zero

    decl mr = sqrt(mOmega) * rann(3, 21);
    decl md = SsfRecursion(mr, mPhi, mOmega, mSigma);

    decl mYt = md[2][1:];       // 20 observations
    decl mKF = KalmanFil(mYt, mPhi, mOmega);
    decl mKS = KalmanSmo(mKF, mPhi, mOmega);
    print("Basic smoother output: mKS\' (t=10)",
        "%c", {"r(1,1)","r(2,1)","e","N(1,1)","N(2,2)","D"},
        mKS[][10]');
    decl msmodist = mKS[0:2][0] ~ mOmega * mKS[0:2][1:];
    print("Smoothed disturbances (t=10)",                  // Smoothed disturbances
        "%c", {"E[H.eta](1,1)","E[H.eta](2,1)","E[G.eps]"},
        msmodist[][10]');
    decl msmostat = SsfRecursion(msmodist, mPhi, mOmega);     // smoothed  states
    print("Smoothed states (t=10)", "%c",
        {"muhat[t+1]","betahat[t+1]","y[t]"}, msmostat[][10]');

    DrawTMatrix(0, msmodist[1:2][],
        {"E[H.eta](2,1)[t]","E[G.eps][t]"}, 0, 1, 1);       // smoothed disturbances
    DrawTMatrix(1, msmostat[0:1][:columns(mYt)-1] | mYt ,
        {"muhat[t]","betahat[t]","y[t]"}, 0, 1, 1);       // smoothed states
    ShowDrawWindow();
}
```

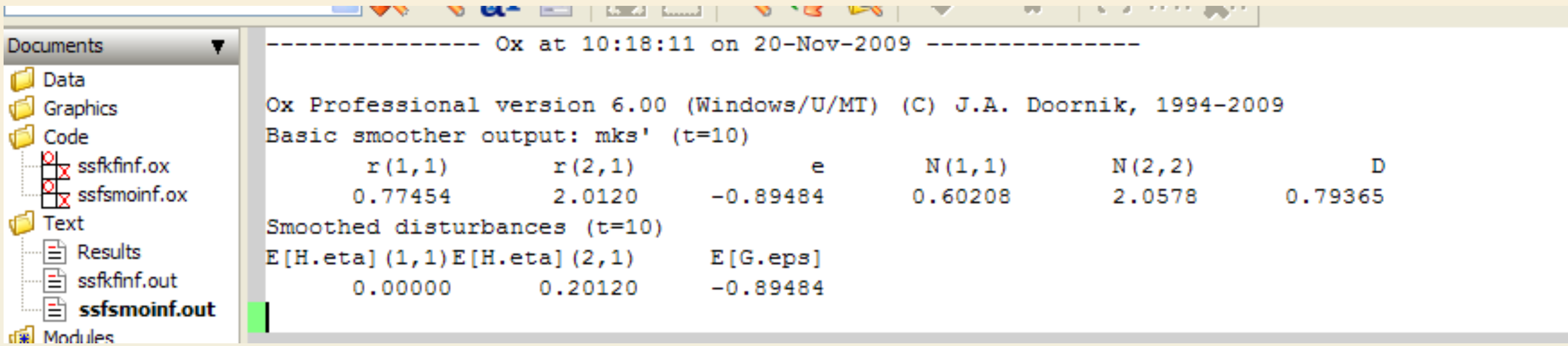
State smoothing output



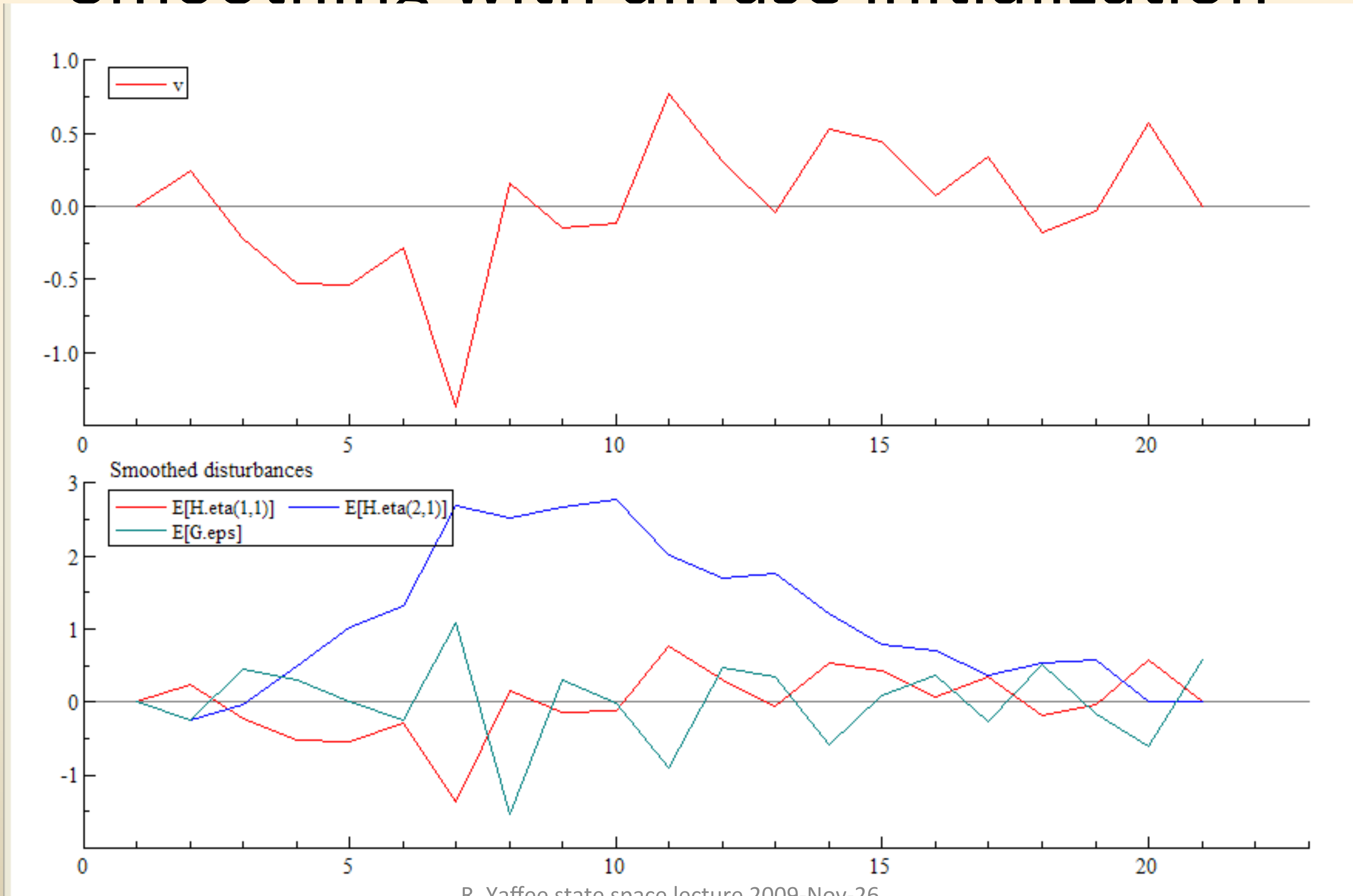
Kalman smoothing with diffuse initialization

```
#include <oxstd.h>
#include <oxdraw.h>
#include <packages/ssfpack/ssfpack_ex.h>
                            //  Kalman smoother with diffuse initialization
main()
{
    decl mphi = <1,1;0,1;1,0>;
    decl momega = diag(<0,0.1,1>);
    decl msigma = <0,0;0,0;1,.5>;       // Note that Q is zero

    decl mr = sqrt(momega) * rann(3, 21);
    decl md = SsfRecursion(mr, mphi, momega, msigma);
    decl myt = md[2][1:];        // 20 observations

    decl mif = KalmanInit(myt, mphi, momega);
    decl mkf = KalmanFilEx(mif, myt, mphi, momega);
    decl mks = KalmanSmoEx(mkf, mphi, momega);
    print("Basic smoother output: mks\' (t=10)",
        "%c", {"r(1,1)","r(2,1)","e","N(1,1)","N(2,2)","D"}, mks[][10]');

    decl msmodist = mks[0:2][0] ~ momega * mks[0:2][1:];
    print("Smoothed disturbances (t=10)",
        "%c", {"E[H.eta](1,1)","E[H.eta](2,1)","E[G.eps]"}, msmodist[][10]');
    DrawTitle(1, "Smoothed disturbances");
    DrawTMatrix(0, mks[0][], {"v"}, 1, 1, 1);
    DrawTMatrix(1, mks[0:2][], {"E[H.eta(1,1)]", "E[H.eta(2,1)]","E[G.eps]"}, 1, 1, 1);
    ShowDrawWindow();
}
```

Output of Kalman smoothing with diffuse intialization



Smoothing with diffuse initialization

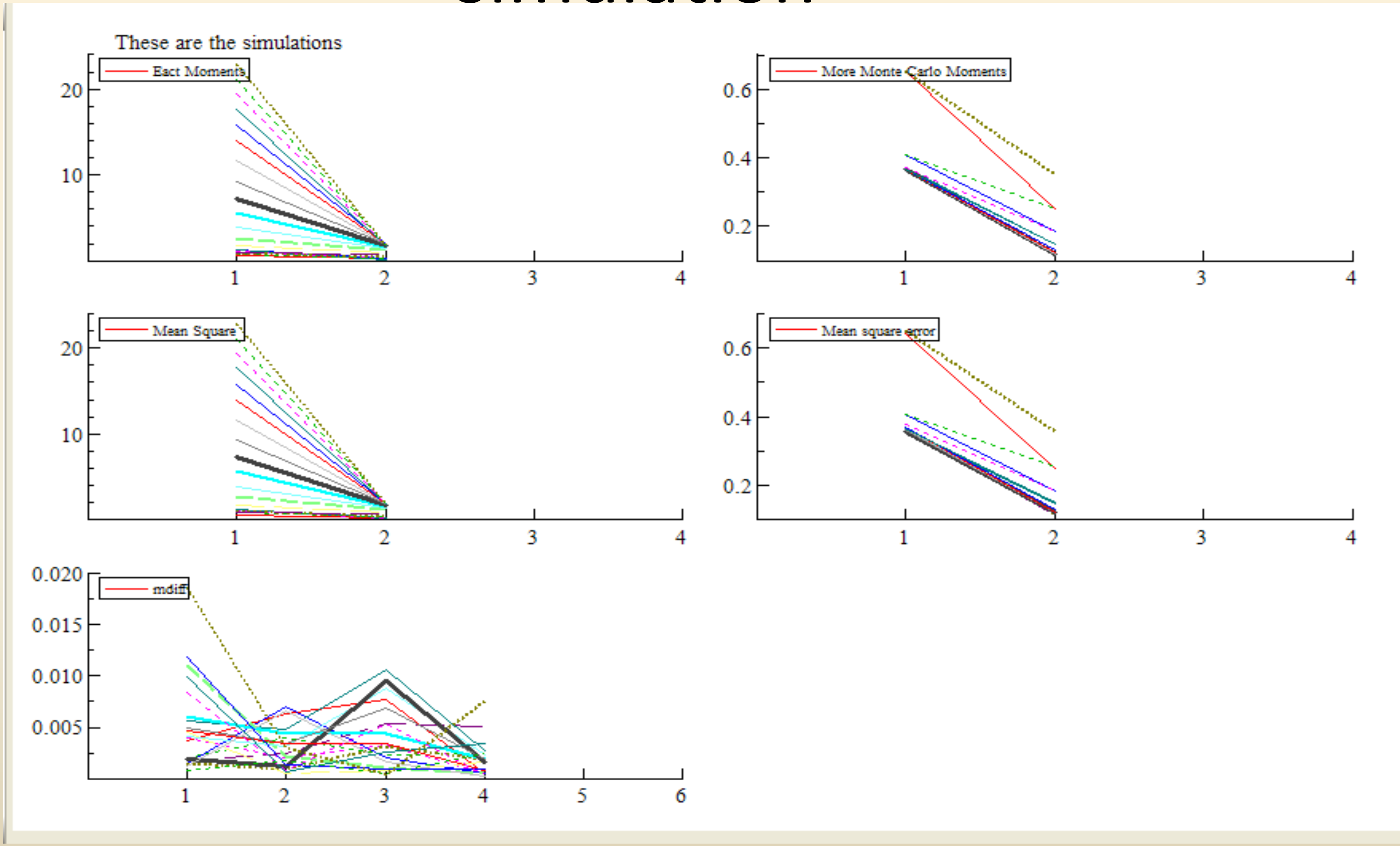


Simulation smoothing with MCMC

Koopman's code snippet : Ssfsimmc4.ox

```
main()
{
    decl mphi = <1,1;0,1;1,0>;
    decl momega = diag(<.5,.1,1>);
    decl msigma = diag(<-1,-1>) | 0;
    decl cst = columns(mphi), csy = rows(mphi);

    decl myt = SsfSimObs(sqrt(momega) * rann(3, 21), mphi,
        momega, <0,0;0,0;1,.5>);
    decl ct = columns(myt);
    decl mif = KalmanInit(myt, mphi, momega, msigma);
    decl mkf = KalmanFilEx(mif, myt, mphi, momega, msigma);

    // monte carlo study
    decl i, imc = 10000, md, mdcum, mdcum2;
    mdcum = mdcum2 = zeros(columns(mphi), ct);
    for (i=0; i<imc; i++)
    {
//      md = SsfCondDens(DS_SIM, myt, mphi, momega, msigma);
        md = SimStSmoDraw(rann(3, 21), mkf, myt, mphi, momega, msigma);
        mdcum  += md;
        mdcum2 += sqr(md);
    }
    mdcum  ./= imc; mdcum2 ./= imc; // Mean, Mean squared
    mdcum2 -= sqr(mdcum); // Variance
//  mdcum2 = diagonal(momega[:cst-1][:cst-1])' - mdcum2; // Cond Variance

    decl mmom;
    SsfMomentEstEx(ST_SMO, &mmom, myt, mphi, momega, msigma);
    println("Exact moments:");
    println("      --------mean--------     --mean square error--",
        mmom[:1][]' ~ mmom[3:4][]');
    println("Monte Carlo moments:");
    println("      --------mean--------     --mean square error--",
```

Convergence of distribution under simulation



Simulation smoothing output

Exact moments:

--------mean--------		--mean square error--	
0.62080	0.25966	0.65215	0.24967
1.1948	0.19680	0.40750	0.18349
1.2778	0.15669	0.37147	0.14786
1.0354	0.19640	0.36913	0.13103
0.85005	0.31246	0.36903	0.12381
1.0021	0.46060	0.36824	0.12094
0.93074	0.71513	0.36737	0.11987
1.7458	0.94967	0.36679	0.11950
2.6431	1.1947	0.36647	0.11938
3.8615	1.4350	0.36634	0.11935
5.6175	1.6111	0.36634	0.11938
7.2890	1.7751	0.36647	0.11950
9.2294	1.9060	0.36679	0.11987
11.646	1.9348	0.36737	0.12094
13.917	1.8963	0.36824	0.12381
15.809	1.8587	0.36903	0.13103
17.662	1.8222	0.36913	0.14786
19.456	1.7914	0.37147	0.18349
21.065	1.7970	0.40750	0.24967
22.890	1.7970	0.65215	0.34967

Monte Carlo moments:

--------mean--------		--mean square error--	
0.62446	0.25338	0.64441	0.25021
1.1963	0.18978	0.40544	0.18404
1.2722	0.15188	0.36087	0.14509
1.0315	0.19428	0.36585	0.13160
0.84797	0.31643	0.36666	0.12130
1.0036	0.46153	0.36503	0.11900
0.92894	0.71756	0.36198	0.11485
1.7504	0.94910	0.36603	0.11790
2.6541	1.1925	0.36532	0.12095
3.8656	1.4318	0.35747	0.12171

Interventions

The data are smoothed with backward recursions condition not just on the previous observation, but on the whole dataset. The result is a smoother signal. Then a residual diagnosis can reveal outliers and level shifts which can seriously bias estimation of a model. Unless these structural breaks are modeled, their effects will be in the error term. Intervention dummy variables can be constructed to model these outliers or level shifts to remove them from the aggregate error.

Identifiability

- According to Andrew Harvey, the order condition is necessary and sufficient for identification of a structural time series model (Harvey, 1989,209).
- Under the condition of normality assumption, identifiability depends upon the nature of the covariance matrix(ibid,206). If this is stationary, so is the autocovariance. To attain stationarity, It may be necessary to place restrictions on the structural model.
- Harvey notes that Hotta (1983) has shown that an order condition is both necessary and sufficient for identifiability (Ibid).

Identifiability-contd.

- For each of the variances of the innovations, we need a separate independent equation to solve for them. These elements constitute the main diagonal of the Ω_t system matrix.
- Also, each of the polynomials must be stationary and of order p_m.
- Each of the parameters of Φ_t and of θ_t must be invertible.
- Any nonstationary polynomial must have no common factor.
- Each error must sum to zero.
- The errors should be normally distributed and independent of the others.
- If the model had an ARIMA configuration of ARIMA(p,d,q), then $p+d > q + 1$ would be sufficient for identification. For example, an ARMA(2,1) is identified if both autocorrelations > 0.
- For more detail, consult Harvey (1989,208).

Diagnostic tests

- Diagnostic tests are applied to identify the components and parameters of the model.
- Diagnostic tests are performed to test the independence, normality, heteroskedasticity, and serial correlation of the residuals.
- These tests are applied to the models to demonstrate that the assumptions are not violated. They are tests of the validity of the model.
- These tests may be applied to filtered or smoothed moments of the model.

Kalman Smoothing

Smoothing for state space models is used for signal extraction and maximum likelihood estimation.

It is used for missing value interpolation (Ansley and Kohn, 1986), cross-validation (Ansley and Kohn, 1987).

Kitagawa (1987) dealt with smoothing for nonlinear processes.

Moment smoothing

Simulation smoothing

Disturbance smoothing

Spline smoothing

Multivariate State Space Models

Multiple time series analysis

Common trends
levels
slopes

Common trends and cycles

cointegration

Dynamic factor analysis

Practical Modeling issues

Assessment problems
- Non-constant innovations problem
- Non-constant variance problems

Prior problems
- Infinite variance problems
- Convergence to zero problems

Gaussianity problems: Conditional Gaussianity

Data irregularities
- Different sampling frequencies
- delayed observations
- Outlier problems
- Level shift problems

Convergence problems
- Multi-modal problems
- Convergence to zero problems

Nonlinearity

Time-varying parameters

Diagnostic Checking of the model

The auxiliary residuals should be used for diagnosing the model. They examine the state residuals as by dividing them by the square root of their variance to provide an effective t-test of the significance of the signal.

These tests are performed on the smoothed residuals and dividing them by their std error.

The auxiliary residuals are functions of the innovations and therefore might be serially correlated. Check to be sure that they are not correlated with the measurement error, which is not supposed to be serially correlated. If there is a cross-correlation here, it may bias estimation in the model (Harvey and Koopman , 2005, 77).

Diagnostic checking of the model

Examine the residuals for nonnormality , serial correlation and lack of independence, homoskedasticity, and excess kurtosis

Look for outliers and level shifts that could render increase the aggregate error and bias the significance test results downward.

References

Athens, M. (1974) The Importance of Kalman Filtering Methods for Economic Systems, Annals of Economic and Social Measurement, V3, no1.

Carlin and Lewis, T. (2009). Bayesian Methods for Data Analysis, 3rd ed. New York: CRC/Chapman and Hall, 17ff.

Commandeur, J.J.F. and Koopman, S.J. (2008). An Introduction to State Space Time Series Analysis. Oxford: Oxford University Press.

DeJong, P. (1988). The Likelihood for a State Space Model. Biometrika, v75. No.1, 165-169.

DeJong, P. (1989). Smoothing and Interpolation with the state Space model. JASA. V84. no.408, 1085-1088.

DeJong, P. (1991). The Diffuse Kalman filter. The Annals of Statistics, v19, No.2., 1073-1083..

Durbin, J. and Koopman, S. J. (2001). Time Series Analysis by State Space Methods. Oxford, U.K.: Oxford University Press, 16-22.

References-ctd.

Harvey, A. C. (1989). Forecasting, Structural Time Series and the Kalman Filter. New York: Cambridge University Press.

Harvey, A. C. and Proietti, T. (2005). Readings in Unobserved Components Models, Diagnostic checking in Unobserved Components State Space Models.

Hyndman, R. J., Koehler, A.B., Ord, J. K., and Snyder, R.D. (2008). Forecasting with Exponential Smoothing: The State Space Approach. New York: Springer.

Kim, C.-J. and Nelson, C. R. (1999). State-Space Models with Regime Switching. Boston: MIT Press.

Koopman, S.J. (2005). Exact Initial Kalman Filtering and Smoothing in Nonstationary Time Series Models, in Harvey, A.C. and Proietti, T. (2005). Readings in Unobserved Component Models. Oxford, U.K. : Oxford University Press, 55.

Koopman, S.J., Shephard, N., and Doornik, J. (2008). SsfPack 3.0. London: Timberlake Consultants, Ltd.,22

References—contd.

Lutkepohl, H. (2008). Multiple Time Series Analysis. 2nd ed. Berlin: Springer.

Reinsel, G. (1997) Elements of multivariate time series analysis. 2nd ed. Springer: New York, Chapter 7.

Tsay, R. (2005). Analysis of Financial Time Series 2nd ed. New York : Wiley., Chapter 11.

Watson, M. (2008). NBER lecture 5 on the Kalman filter, Nonlinear Filtering, and MCMC at NBER summer institute. July 15, 2008.