Macroeconomic Forecasting and Stochastic Volatility Modelling Belgische Francqui-leerstoel 2019-2020 Universiteit Antwerpen

Siem Jan Koopman

Vrije Universiteit Amsterdam, Tinbergen Institute

Tuesday 8 December 2020

http://sjkoopman.net

State Space Model and Kalman Filter

State Space Model

Linear Gaussian state space model is defined in three parts:

→ State transition equation:

$$\alpha_{t+1} = T_t \alpha_t + R_t \zeta_t, \qquad \zeta_t \sim \mathcal{NID}(0, Q_t),$$

→ Observation equation:

$$y_t = Z_t \alpha_t + \varepsilon_t, \qquad \varepsilon_t \sim \mathcal{NID}(0, H_t),$$

→ Initial condition: $\alpha_1 \sim \mathcal{N}(a_1, P_1)$.

- ζ_t and ε_s independent for all t, s, and independent from α_1;
- observation y_t can be a vector;
- state vector α_t is unobserved;
- system matrices T_t, Z_t, R_t, Q_t, H_t are fixed at time t.
- system matrices are known functions of parameter vector.
- system matrices determine (dynamic) structure of model.
- initial conditions : specify a_1 and P_1.

State Space Model

- state space model is linear and Gaussian: therefore properties and results of multivariate normal distribution apply;
- state vector α_t evolves as a VAR(1) process;
- system matrices usually contain unknown parameters;
- estimation has therefore two parts:
  - measuring the unobservable state (prediction, filtering and smoothing);
  - estimation of unknown parameters (maximum likelihood estimation);

Local level model in State Space

We have state space model

$$\begin{aligned} \alpha_{t+1} &= T_t \alpha_t + R_t \zeta_t, & \zeta_t &\sim \mathcal{NID}(0, Q_t), \\ y_t &= Z_t \alpha_t + \varepsilon_t, & \varepsilon_t &\sim \mathcal{NID}(0, H_t), \end{aligned}$$

with some initialisation for the state vector, that is α_1.

The local level model $y_t = \mu_t + \varepsilon_t$ with $\mu_{t+1} = \mu_t + \eta_t$ is in state space form with

$$T_t = 1, \qquad R_t = 1, \qquad Z_t = 1, \qquad Q_t = \sigma_\eta^2, \qquad H_t = \sigma_\varepsilon^2,$$

and with $\alpha_t \equiv \mu_t$ and $\zeta_t \equiv \eta_t$, for all $t = 1, \ldots, n$.

The unknown variances are treated as parameters and are in system matrices Q_t and H_t.

We discuss initialisation issues below.

Signal plus noise model

Also other unobserved components time series models can be formulated in state space, see next week.
They can be viewed as a signal plus noise model

$$y_t = \theta_t + \varepsilon_t,$$

where θ_t is the unobserved dynamic signal with $\theta_t = Z_t \alpha_t$ and the state vector contains all dynamic structures in the time series.

The signal can consist of

- non-stationary trend (level)
- stationary autoregressive processes
- seasonal effects
- cyclical process

or any combination of these !!

ARMA models in state space form: AR(1) model

Consider AR(1) model $y_{t+1} = \phi y_t + \zeta_t$, in state space form:

$$\alpha_{t+1} = T_t \alpha_t + R_t \zeta_t, \qquad \zeta_t \sim \mathcal{NID}(0, Q_t),$$
$$y_t = Z_t \alpha_t + \varepsilon_t, \qquad \varepsilon_t \sim \mathcal{NID}(0, H_t).$$

with scalar state vector α_t and system matrices

$$Z_t = 1, \qquad H_t = 0,$$
$$T_t = \phi, \qquad R_t = 1, \qquad Q_t = \sigma^2$$

we have

- $Z_t = 1$ and $H_t = 0$ imply that $\alpha_t = y_t$;
- State equation implies $y_{t+1} = \phi y_t + \zeta_t$ with $\zeta_t \sim \mathcal{NID}(0, \sigma^2)$;
- This is the AR(1) model !
- To obtain AR(1) plus noise model, simply have $H_t > 0$.

Time Series Models in state space form

We have reviewed some standard models that can be formulated in state space form:

- unobserved components time series models
- autoregressive integrated moving average models
- dynamic regression models

We can also treat

- sums of linear dynamic components
- any linear Gaussian dynamic process

Initial conditions : stationary process

Consider AR(1) plus noise model

$$y_t = \mu_t + \varepsilon_t, \qquad \mu_{t+1} = \phi\mu_t + \eta_t,$$

In state space form, with state vector $\alpha_t = \mu_t$, we have

$$Z_t = 1, \quad H_t = \sigma^2_\varepsilon, \quad T_t = \phi, \quad R_t = 1, \quad Q_t = \sigma^2_\eta.$$

What about the initial condition $\alpha_1 \sim \mathcal{N}(a_1, p_1)$?
Here a_1 is the *unconditional* mean and p_1 is the *unconditional* variance of the stationary process of μ_t.

The state $\alpha_t \equiv \mu_t$ follows autoregressive process of order 1.

Hence $a_1 = 0$ and $p_1 = \sigma^2_\eta \,/\, (1 - \phi^2)$.

This solution leads to a general solution for all stationary elements in the state vector.

Initial conditions : non-stationary process

Consider AR(1) plus noise model

$$y_t = \mu_t + \varepsilon_t, \qquad \mu_{t+1} = \phi \mu_t + \eta_t,$$

What about the initial condition $\alpha_1 \sim \mathcal{N}(a_1, p_1)$ when state process $\alpha_t = \mu_t$ is non-stationary, that is $\phi \to 1$?
We obtain the Local Level model,

$$y_t = \mu_t + \varepsilon_t, \qquad \mu_{t+1} = \mu_t + \eta_t,$$

Here a_1 is the unconditional mean and p_1 is the unconditional variance of the non-stationary process of μ_t.

Then $a_1 = 0$ and $p_1 = \sigma_\eta^2 \,/\, (1 - \phi^2) \to \infty$ as $\phi \to 1$.

We have $a_1 = 0$ and $p_1 \to \infty$; this is generally the solution for non-stationary elements.

Kalman Filter

Given the state space model $\alpha_{t+1} = T_t\alpha_t + R_t\zeta_t, \quad y_t = Z_t\alpha_t + \varepsilon_t$ with known system matrices :

- The Kalman filter calculates the mean and variance of the unobserved state, given the observations.
- The state is Gaussian: the complete distribution is characterized by the mean and variance.
- The filter is a recursive algorithm; the current best estimate is updated whenever a new observation is obtained.
- To start the recursion, we need a_1 and P_1, which we assume given. There are various ways to initialize the Kalman filter.
- Assumption is known system matrices; later we discuss estimation of parameter vector.

Kalman Filter

Given the state space model $\alpha_{t+1} = T_t\alpha_t + R_t\zeta_t, \quad y_t = Z_t\alpha_t + \varepsilon_t$ with known system matrices :

The unobserved state α_t can be "estimated" from the observations with the *Kalman filter*:

$$
\begin{aligned}
v_t &= y_t - Z_t a_t,\\
F_t &= Z_t P_t Z_t' + H_t,\\
K_t &= T_t P_t Z_t' F_t^{-1},\\
a_{t+1} &= T_t a_t + K_t v_t,\\
P_{t+1} &= T_t P_t T_t' + R_t Q_t R_t' - K_t F_t K_t',
\end{aligned}
$$

for $t = 1, \ldots, n$ and starting with given values for a_1 and P_1.

- Writing $Y_t = \{y_1, \ldots, y_t\}$,

$$a_{t+1} = \mathbb{E}(\alpha_{t+1}|Y_t), \qquad P_{t+1} = \mathbb{V}\mathrm{ar}(\alpha_{t+1}|Y_t).$$

Kalman Filter : discussion

- Our best prediction of y_t is $Z_t a_t$. When the actual observation y_t arrives, calculate the prediction error $v_t = y_t - Z_t a_t$ and its variance $F_t = Z_t P_t Z_t' + H_t$.
- The best estimate a_{t+1} is based on both the old estimate a_t and the new information v_t:

  $$a_{t+1} = T_t a_t + K_t v_t,$$

  with its variance

  $$P_{t+1} = T_t P_t T_t' + R_t Q_t R_t' - K_t F_t K_t'.$$

- The *Kalman gain*

  $$K_t = T_t P_t Z_t' F_t^{-1}$$

  is the optimal weighting matrix for the new evidence.

Kalman filter for the Local Level Model

For the special case of the Local Level model,

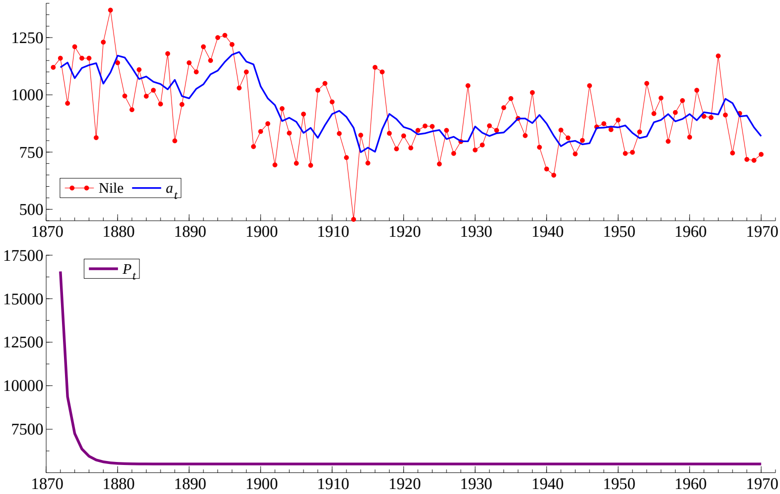
$$y_t = \mu_t + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, \sigma_\varepsilon^2), \quad \mu_{t+1} = \mu_t + \eta_t, \quad \eta_t \sim \mathcal{N}(0, \sigma_\eta^2),$$

in state space formulation we have $T_t = R_t = Z_t = 1$, $Q_t = \sigma_\eta^2$ and $H_t = \sigma_\varepsilon^2$, and the Kalman filter equations are given by

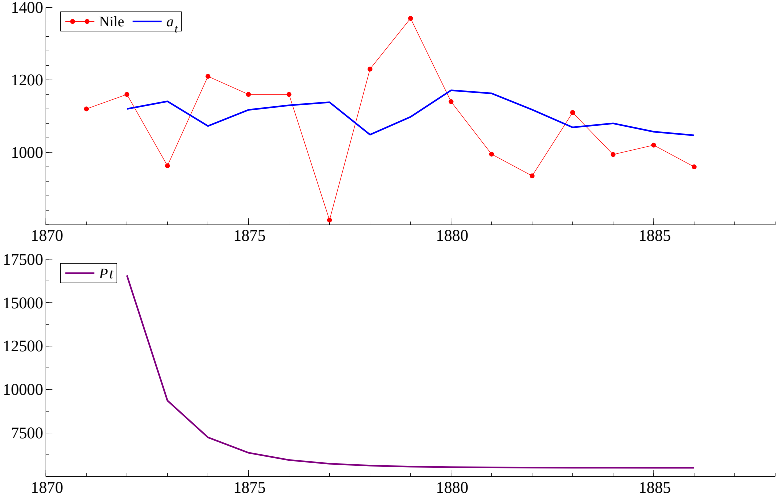
$$\begin{aligned} v_t &= y_t - a_t, & f_t &= p_t + \sigma_\varepsilon^2, \\ k_t &= p_t \,/\, f_t, & & \\ a_{t+1} &= a_t + k_t v_t, & p_{t+1} &= k_t \, \sigma_\varepsilon^2 + \sigma_\eta^2, \end{aligned}$$

for $t = 1, \ldots, n$ with initialisation $a_1 = 0$ and $p_1 = \sigma_\varepsilon^2 \times 10^7$ or $p_1 \to \infty$.

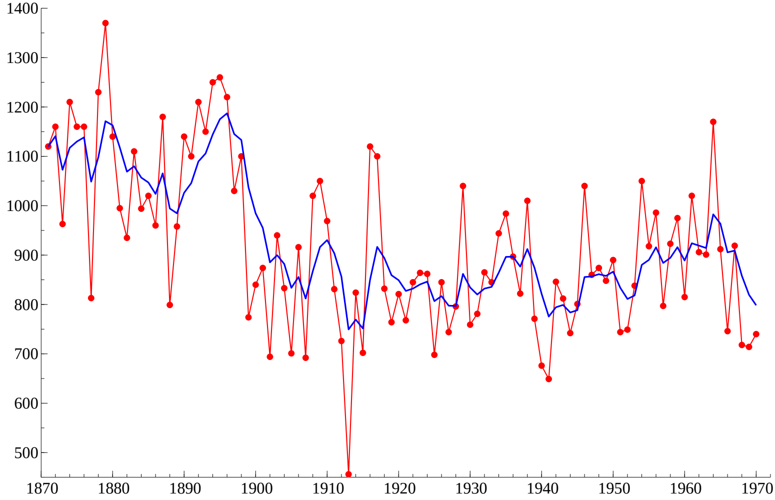
Kalman Filter prediction



Kalman Filter prediction, at the start



Signal Extraction for Nile Data: filtered estimate of level



Observation weights

The predicted estimate a_t and the filtered estimate $a_{t|t}$ of the state vector α_t, both obtained from the Kalman filter, can both be expressed as **linear, weighted** functions of the past observations (for filtering, including y_t).

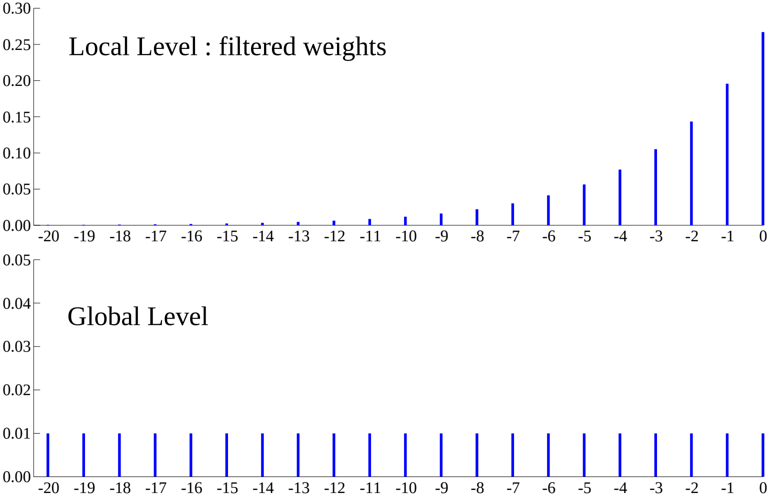
For example, we have

$$a_{t|t} = \mathbb{E}(\alpha_t | Y_t) = \sum_{j=0}^{t-1} w_{t,j}\, y_{t-j},$$

where $w_{t,j}$ are the **observation weights**.

The weight function, the weights $w_{t,j}$ against j, can be compared with the *kernel* function in nonparametric (regression) analysis.

Signal Extraction for Nile Data: observation weights



Maximum Likelihood Estimation and Time Series Analysis

Prediction error decomposition

System matrices are function of the unknown parameter vector ψ.

For a known ψ, we evaluate the likelihood function based on the joint observation density via prediction decomposition.

For two observations $n = 2$, we have

$$f(y_1, y_2; \psi) = f(y_1; \psi) f(y_2 | y_1; \psi).$$

For three observations $n = 3$, we have

$$\begin{aligned} f(y_1, y_2, y_3; \psi) &= f(y_1, y_2; \psi) f(y_3 | y_1, y_2; \psi) \\ &= f(y_1; \psi) f(y_2 | y_1; \psi) f(y_3 | y_1, y_2; \psi). \end{aligned}$$

More generally, for any n,

$$f(y_1, \ldots, y_n; \psi) = f(y_1; \psi) \prod_{t=2}^{n} f(y_t | Y_{t-1}; \psi).$$

Loglikelihood function

The loglikelihood function is given by

$$\log L(\psi) = \sum_{t=1}^{n} \log f(y_t | Y_{t-1}; \psi), \qquad Y_0 \equiv \{\},$$

where

$$f(y_t | Y_{t-1}; \psi) \equiv \mathcal{N}(Z_t a_t, F_t), \qquad t = 1, \ldots, n.$$

It follows that:

$$\log L(\psi) = -\frac{n\#(y_t)}{2} \log 2\pi - \frac{1}{2} \sum_{t=1}^{n} \log |F_t| - \frac{1}{2} \sum_{t=1}^{n} v_t' F_t^{-1} v_t,$$

where $\#(y_t)$ is the dimension of y_t and $v_t = y_t - Z_t a_t$.

Estimation proceeds by numerically maximising $\log L(\psi)$ wrt ψ.

Parameter Estimation for the Local Level model

The Kalman filter for the Local Level model is given by

$$
\begin{aligned}
v_t &= y_t - a_t, & f_t &= p_t + \sigma_\varepsilon^2, \\
k_t &= p_t \,/\, f_t, & & \\
a_{t+1} &= a_t + k_t v_t, & p_{t+1} &= k_t\, \sigma_\varepsilon^2 + \sigma_\eta^2,
\end{aligned}
$$

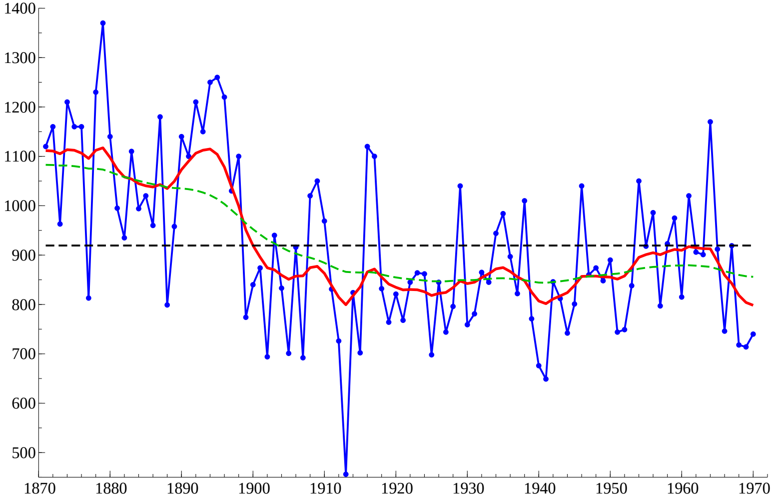
for $t = 1, \ldots, n$ with initialisation $a_1 = 0$ and $p_1 = \sigma_\varepsilon^2 \times 10^7$.

The loglikelihood function is given by

$$
\log L(\psi) = -\frac{n}{2} \log 2\pi - \frac{1}{2} \sum_{t=1}^{n} \log f_t - \frac{1}{2} \sum_{t=1}^{n} v_t^2 \,/\, f_t.
$$

Estimation is done by maximising $\log L(\psi)$ wrt $\psi = (\sigma_\varepsilon^2, \sigma_\eta^2)'$.

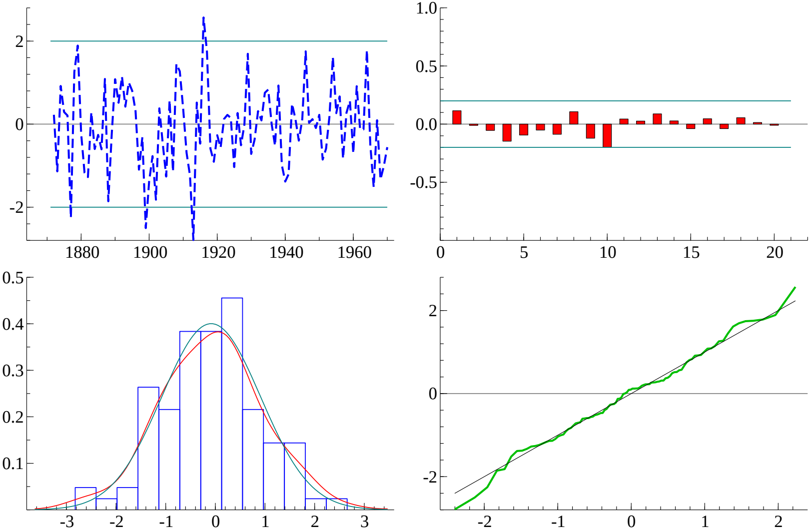
Local Level Estimates for $q \in \{0.0,\ 0.01,\ \widehat{q},\ \infty\}$



Goodness-of-Fit and Diagnostics

- Under correct model specification, the standardised residuals have property
$$v_t \,/\, \sqrt{F}_t \sim \mathcal{NID}(0, 1),$$
for $t = 1, \ldots, n$. In Local Level case, we have $v_t \,/\, \sqrt{f_t}$.
- Apply standard test for Normality, heteroskedasticity, serial correlation;
- A recursive algorithm is available to calculate smoothed disturbances (auxilliary residuals), which can be used to detect breaks and outliers;
- Model comparison and parameter restrictions: use likelihood based procedures (LR test, AIC, BIC);
- So much more to say, if only time permits ...

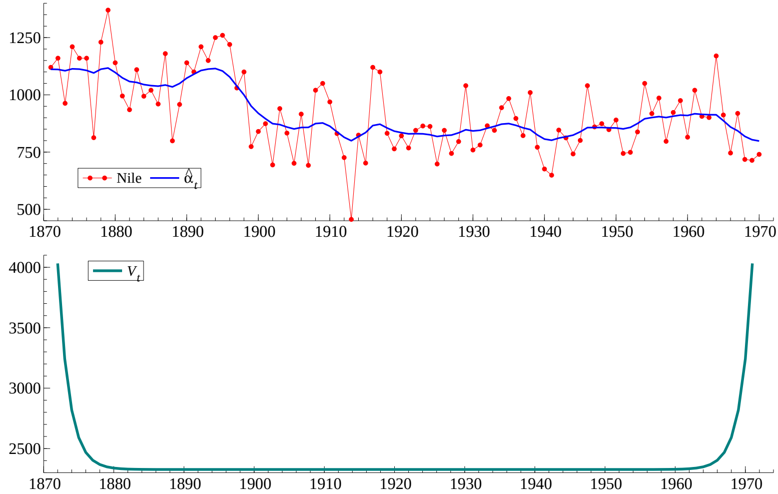
Nile Data Residuals Diagnostics



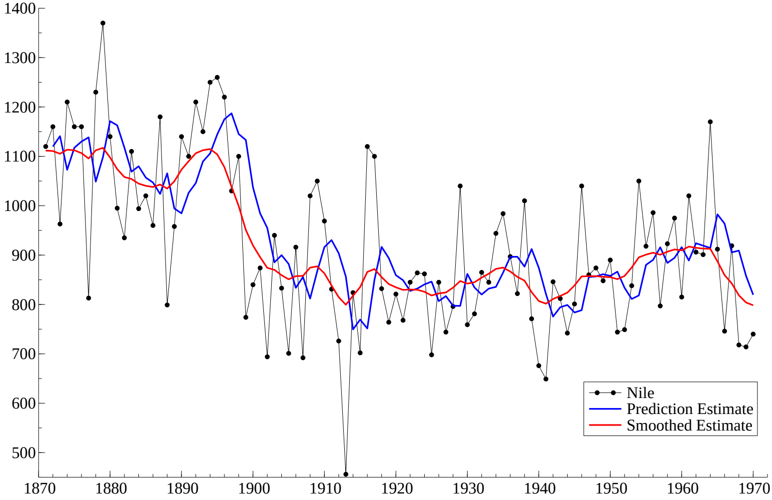
Smoothing

- The filter calculates the mean and variance of the state vector variable conditional on Y_{t-1} (prediction) or Y_t (filtering);
- Smoothing refers to the computation of the mean and variance of the state vector variable conditional on the full set of observations Y_n;
- After the predicted/filtered estimates are calculated by the Kalman filter, the smoothing recursion starts at the last observation and runs until the first.

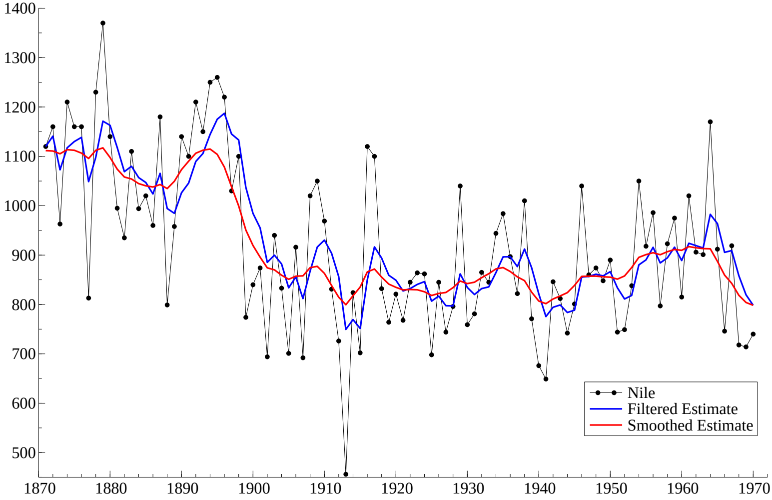
Signal Extraction for Nile Data: Smoothed Estimates



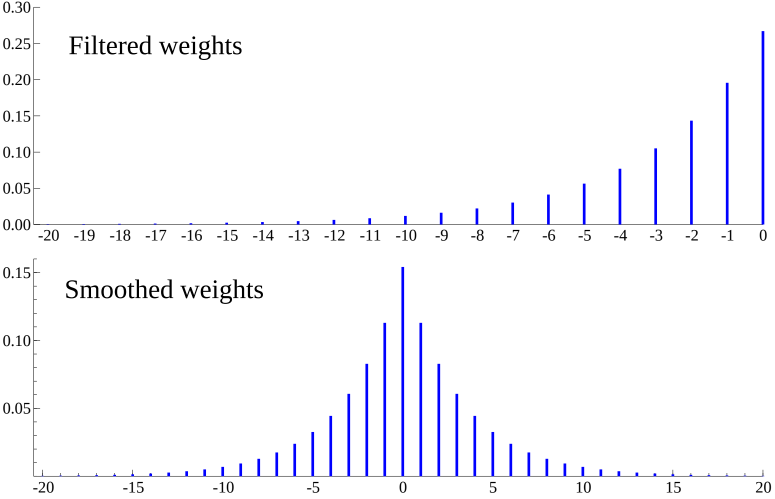
Prediction and Smoothed Estimates



Filtered and Smoothed Estimates



Filtered and Smoothed Weight Functions



Kalman Filter and Missing Observations

Given the state space model $\alpha_{t+1} = T_t\alpha_t + R_t\zeta_t, \quad y_t = Z_t\alpha_t + \varepsilon_t$ with known system matrices :

The unobserved state α_t can be "estimated" from the observations with the *Kalman filter*:

$$
\begin{aligned}
v_t &= y_t - Z_t a_t, \\
F_t &= Z_t P_t Z_t' + H_t, \\
K_t &= T_t P_t Z_t' F_t^{-1}, \\
a_{t+1} &= T_t a_t + K_t v_t, \\
P_{t+1} &= T_t P_t T_t' + R_t Q_t R_t' - K_t F_t K_t',
\end{aligned}
$$

for $t = 1, \ldots, n$ and starting with given values for a_1 and P_1.

- Writing $Y_t = \{y_1, \ldots, y_t\}$,

$$a_{t+1} = \mathbb{E}(\alpha_{t+1}|Y_t), \qquad P_{t+1} = \mathbb{V}\text{ar}(\alpha_{t+1}|Y_t).$$

Missing Observations

Missing observations are very easy to handle in the Kalman filter:

$$v_t = y_t - Z_t a_t, \quad F_t = Z_t P_t Z_t' + H_t, \quad K_t = T_t P_t Z_t' F_t^{-1},$$

with update

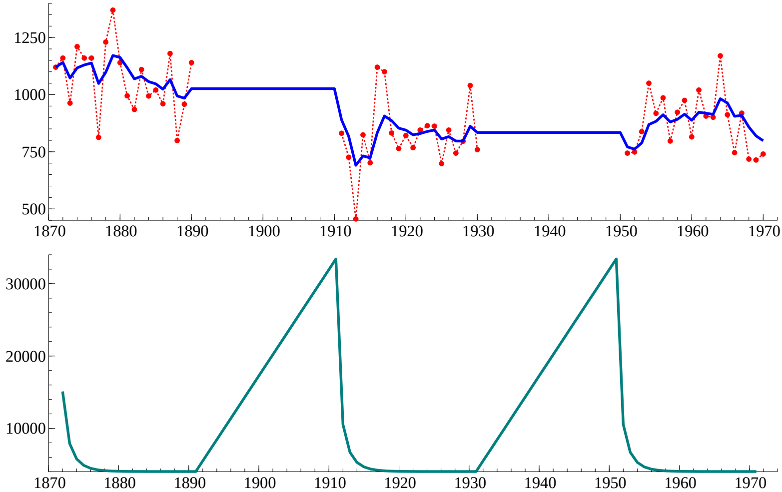
$$a_{t+1} = T_t a_t + K_t v_t, \quad P_{t+1} = T_t P_t T_t' + R_t Q_t R_t' - K_t F_t K_t'.$$

- suppose y_j is missing, its value is not known
- put $v_j =??$ and let $F_j \to \infty$ in the Kalman filter
- it follows that $K_j \to 0$
- proceed further calculations as normal

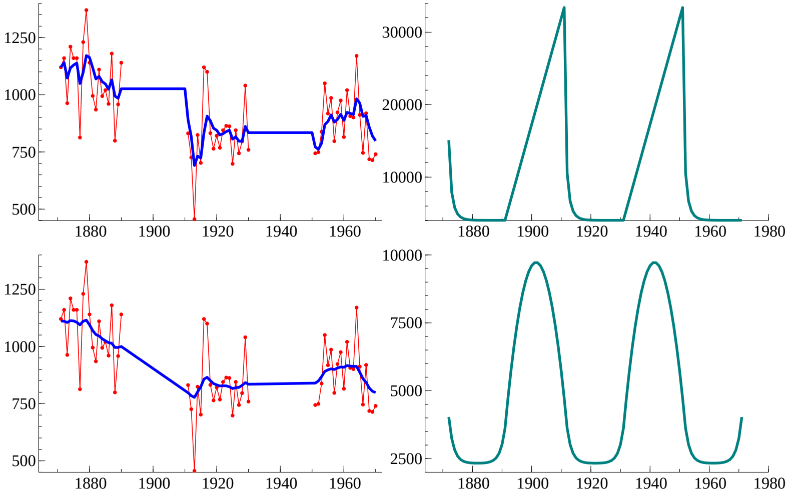
Update : $a_{j+1} = T_j a_j$ and $P_{j+1} = T_j P_j T_j' + R_j Q_j R_j'$.

The Kalman filter extrapolates the state vector until a new observation arrives. The backwards smoother interpolates between observations.

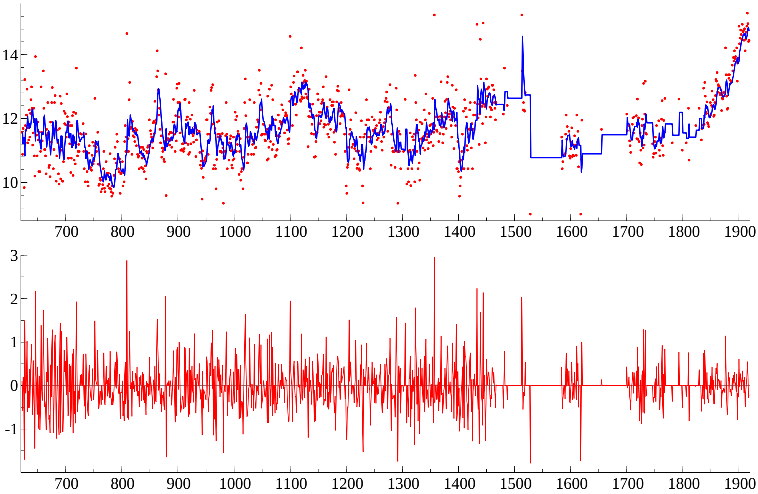
Local Level Model for Nile Data with Missing Observations



Missing Observations, Filter and Smoother



Climate Change: Nile Volume Predictions over Centuries



Filtering and Smoothing in Political Polling at Politico

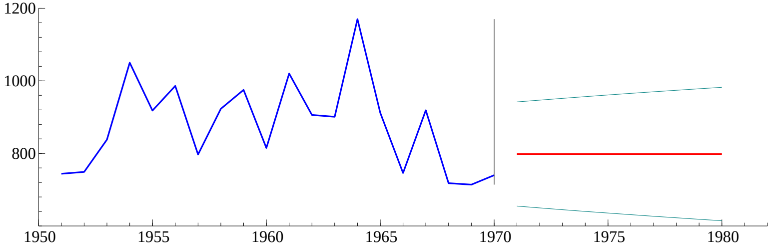
https://www.politico.eu/europe-poll-of-polls/united-kingdom/

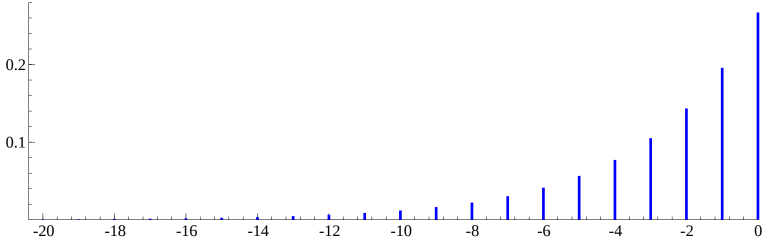
Forecasting

Forecasting requires no extra theory: just treat future observations as missing:

- put $v_j =??$ and $F_j \to \infty$ for $j = n + 1, \ldots, n + k$
- have $K_j \to 0$
- proceed further calculations as normal
- forecast for y_j is $Z_j a_j$ and is provided by the Kalman filter
- the forecast itself as well as its standard error
- also forecasts for any function of state vector

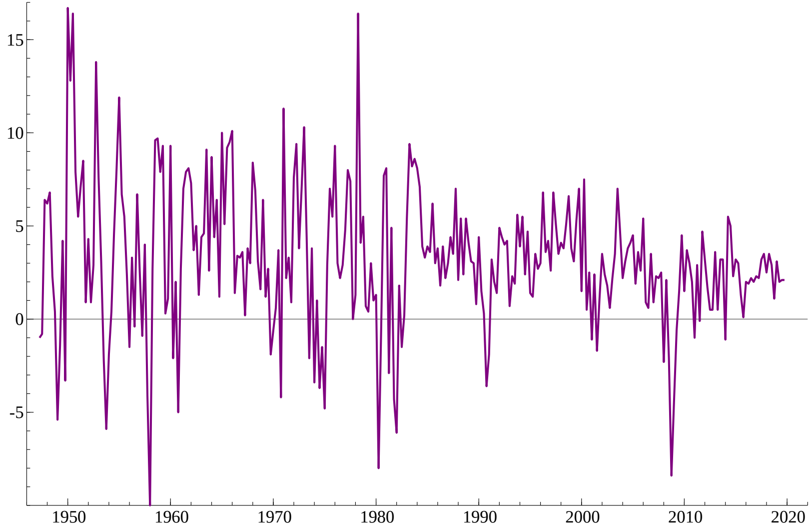
Forecasting





Macroeconomic time series

Gross Domestic Product Growth 1947Q1 – 2019Q4



Trend-Cycle Decomposition

Macroeconomic time series are often decomposed into trend and cycle components:

$$y_t = \mu_t + \psi_t + \varepsilon_t,$$

where the trend component is μ_t with growth component ν_t:

$$\mu_{t+1} = \mu_t + \nu_t, \qquad \nu_{t+1} = \nu_t + \eta_t,$$

where the cycle component is ψ_t and can be modelled as the stationary autoregressive process:

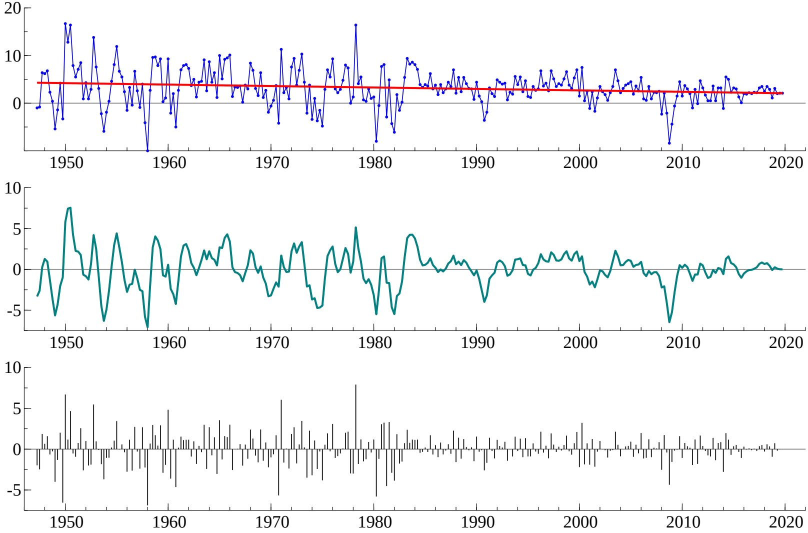
$$\psi_{t+1} = \phi_1 \psi_t + \phi_2 \psi_{t-1} + \kappa_t,$$

with disturbances

$$\varepsilon_t \sim \mathcal{NID}(0, \sigma_\varepsilon^2), \qquad \eta_t \sim \mathcal{NID}(0, \sigma_\eta^2), \qquad \kappa_t \sim \mathcal{NID}(0, \sigma_\kappa^2),$$

which are serially and mutually independent noise series.

GDP Growth: Trend, Cycle and Noise



Trend-Cycle Decomposition

For macroeconomic time series, the trend-cycle decomposition is effective and useful:

$$y_t = \mu_t + \psi_t + \varepsilon_t,$$

with trend μ_t and cycle ψ_t.

- The unobserved cycle process ψ_t can be modelled as an AR(2) process, as it can represent a cyclical process.
- Conditions for stationarity are well-known and can be derived.
- Conditions for cyclicality are less known, it requires the roots of the AR polynomial to be within the complex range.
- We can enforce cyclicality in autoregressive processes within state space models, see next slide.

Business Cycle component

An ARMA(2,1) model for which the autoregressive polynomial is enforced to have complex roots is given by

$$\begin{pmatrix} \psi_{t+1} \\ \psi^{\dagger}_{t+1} \end{pmatrix} = \rho \begin{bmatrix} \cos\lambda & \sin\lambda \\ -\sin\lambda & \cos\lambda \end{bmatrix} \begin{pmatrix} \psi_{t} \\ \psi^{\dagger}_{t} \end{pmatrix} + \begin{pmatrix} \kappa_{t} \\ \kappa^{\dagger}_{t} \end{pmatrix},$$

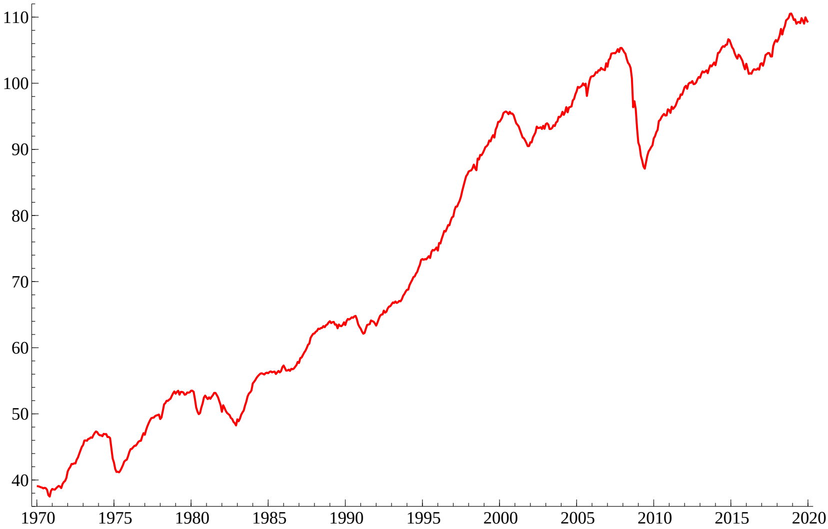
with damping factor ρ and frequency λ, and disturbances

$$\kappa_t, \kappa^{\dagger}_t \sim \mathcal{NID}(0, \sigma^2_{\kappa}),$$

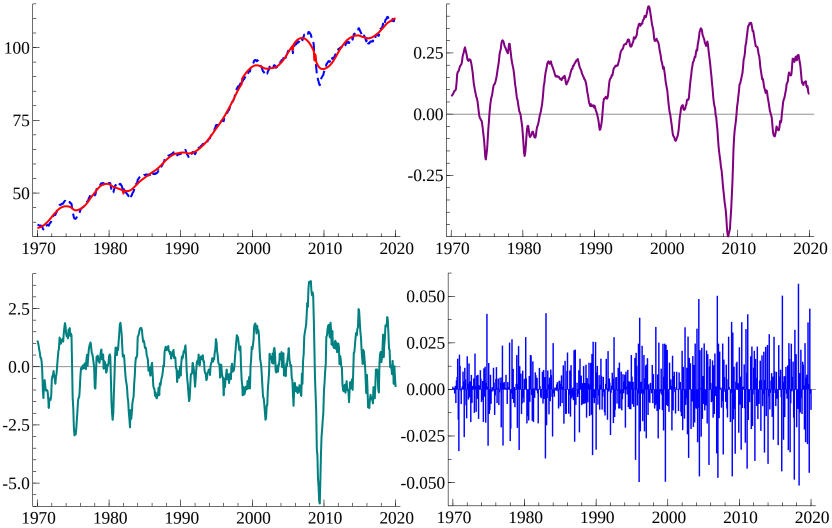
which are serially and mutually independent series.

- The stationary cycle has $0 < \rho < 1$ and period $2\pi \,/\, \lambda$.
- Business cycle length is typically between 1.5 and 12 years. For example, when we have a quarterly time series, we restrict λ such that $6 < 2\pi \,/\, \lambda < 48$.

US Industrial Production, SA, monthly 1970-2019



Decomposition of Industrial Production



The take-aways on time series decompositions

The trend-cycle decomposition model for macro time series:

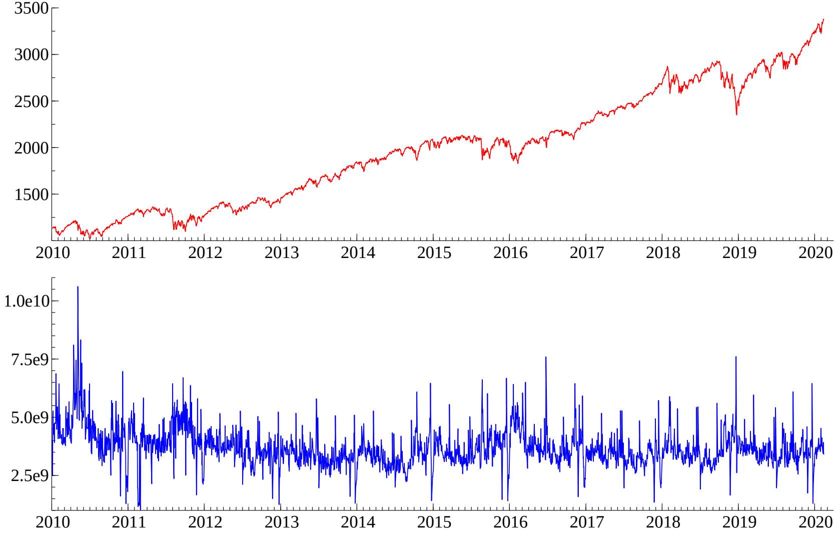
- introduces a novelty in time series modelling;
- exploits the flexibility of the state space framework;
- combines stationary and non-stationary features into one model.

What is relevant here ?

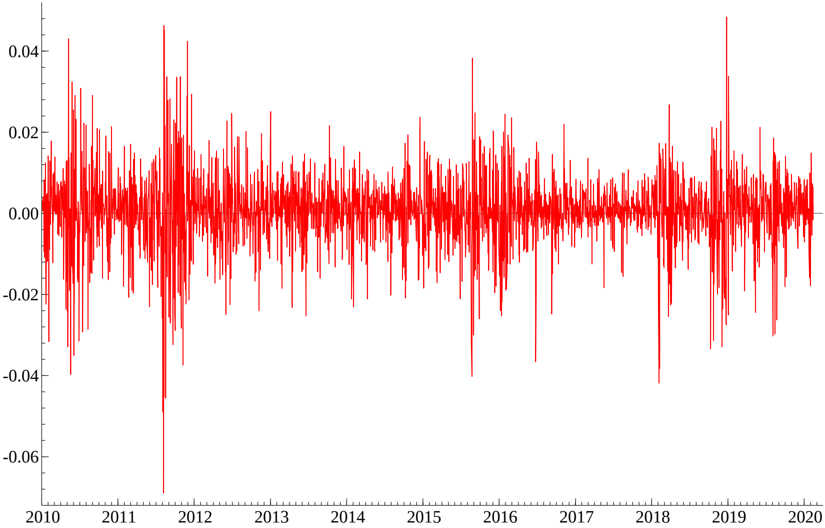
- statistical formulation of trend-cycle model;
- the state space representation of a trend-cycle model;
- initial conditions for the state vector;

Financial Time Series: Volatility

Standard and Poor's 500 index/volume, daily, from 2010



Standard and Poor's 500 returns, daily, from 2010



Stochastic Volatility (SV) model

The stochastic volatility (SV) model for a time series of financial (daily) returns $y_t = \log(P_t) - \log(P_{t-1})$ is given by

$$y_t = \mu + \sigma_t \varepsilon_t, \qquad \log \sigma_t^2 = h_t, \qquad h_{t+1} = \omega + \phi h_t + \sigma_\eta \eta_t,$$

where μ is the overall mean of returns (positive for S&P500),
σ_t^2 is the stochastically time-varying (daily) volatility,
$\varepsilon_t \sim \mathcal{NID}(0,1)$,
h_t is the log-volatility, ϕ is the persistence in log-volatility,
$\omega/(1-\phi)$ is the unconditional mean in log-volatility, and
σ_η determines the overall scale of volatility, known as vol-of-vol.

First example of NOT having a dynamic process in the mean equation, but in the variance equation.

Full treatment of the SV model in the next weeks.

The Stochastic Volatility (SV) model

Given the SV model

$$y_t = \mu + \sigma_t \varepsilon_t, \qquad \log \sigma_t^2 = h_t, \qquad h_{t+1} = \omega + \phi h_t + \sigma_\eta \eta_t,$$

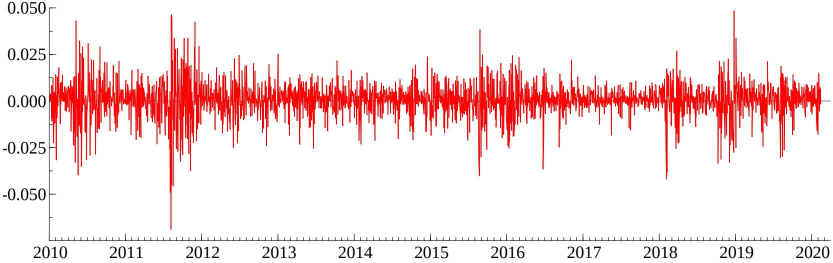
and after transformation $x_t = \log(y_t - \mu)^2$, we obtain a stationary linear decomposition model for x_t

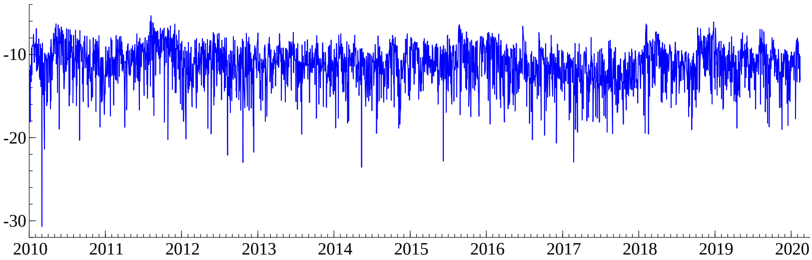
$$x_t = h_t + u_t, \qquad h_{t+1} = \omega + \phi h_t + \sigma_\eta \eta_t,$$

where $u_t = \log \varepsilon_t^2$.

In effect, we have a linear AR(1) plus noise model for x_t, with h_t now in the mean equation but disturbance u_t is not necessarily Gaussian !!

S&P 500 returns transformation x_t





Quasi-ML method for Stochastic Volatility model

For the extraction of volatility, we can base the analysis on the linear decomposition model for $x_t = \log(y_t - \mu)^2$, that is

$$x_t = h_t + u_t, \qquad h_{t+1} = \omega + \phi h_t + \sigma_\eta \eta_t,$$

where $u_t = \log \varepsilon_t^2$. Notice: we replace μ by the sample mean of y_t.

Sure, it is empirically challenging to assume normally distributed u_t's, but let's do it anyways, but we also account for possible assumption $\varepsilon_t \sim \mathcal{NID}(0,1)$. We have

$$u_t \sim \mathcal{NID}(c, d), \qquad t = 1, \ldots, n,$$

where c and d are mean and variance of $\log \chi^2(1)$ distribution, respectively.

In this case, we can apply all state space methods to the model for x_t, including parameter estimation: **quasi-ML method**.

Volatility extraction

Based on the model

$$x_t = h_t + u_t, \qquad h_{t+1} = \omega + \phi h_t + \sigma_\eta \eta_t,$$

we can extract h_t by using Kalman filter and smoothing methods.

However, we are interested in the signal extraction of

$$\sigma_t^2 = \exp(\, h_t \,), \qquad t = 1, \ldots, n.$$

We are also aware of the notion that

$$\widehat{\sigma}_t^2 \neq \exp\left(\, \widehat{h}_t \,\right), \qquad t = 1, \ldots, n,$$

where $\widehat{\sigma}_t^2 = \mathbb{E}(\sigma_t^2 | Y_n)$ and $\widehat{h}_t = \mathbb{E}(h_t | Y_n)$, for $t = 1, \ldots, n$.

This problem can be tackled, but it is another meeting !!

S&P 500 volatility signal extraction