MONASH University

Rob J Hyndman

Forecasting:

Principles and Practice

9. State space models

Outline

Exponential smoothing methods

Trend Component		Seasonal Component N (None)	Seasonal Component A (Additive)	Seasonal Component M (Multiplicative)
N	(None)	N,N	N,A	N,M
A	(Additive)	A,N	A,A	A,M
A_d	(Additive damped)	A_d,N	A_d,A	A_d,M
M	(Multiplicative)	M,N	M,A	M,M
M_d	(Multiplicative damped)	M_d,N	M_d,A	M_d,M

General notation E T S : **E**xponen**T**ial **S**moothing

A,N,N: Simple exponential smoothing with additive errors
A,A,N: Holt's linear method with additive errors
M,A,M: Multiplicative Holt-Winters' method with multiplicative errors

Exponential smoothing methods

Trend Component		Seasonal Component N (None)	A (Additive)	M (Multiplicative)
N	(None)	N,N	N,A	N,M
A	(Additive)	A,N	A,A	A,M
A_d	(Additive damped)	A_d,N	A_d,A	A_d,M
M	(Multiplicative)	M,N	M,A	M,M
M_d	(Multiplicative damped)	M_d,N	M_d,A	M_d,M

General notation E T S : **E**xponen**T**ial **S**moothing

Examples:

A,N,N: Simple exponential smoothing with additive errors
A,A,N: Holt's linear method with additive errors
M,A,M: Multiplicative Holt-Winters' method with multiplicative errors

Exponential smoothing methods

	Trend Component	Seasonal Component N (None)	Seasonal Component A (Additive)	Seasonal Component M (Multiplicative)
N	(None)	N,N	N,A	N,M
A	(Additive)	A,N	A,A	A,M
A_d	(Additive damped)	A_d,N	A_d,A	A_d,M
M	(Multiplicative)	M,N	M,A	M,M
M_d	(Multiplicative damped)	M_d,N	M_d,A	M_d,M

General notation E T S : **E**xponen**T**ial **S**moothing

↑

Trend

Examples:

A,N,N: Simple exponential smoothing with additive errors

A,A,N: Holt's linear method with additive errors

M,A,M: Multiplicative Holt-Winters' method with multiplicative errors

Exponential smoothing methods

Trend Component		Seasonal Component N (None)	A (Additive)	M (Multiplicative)
N	(None)	N,N	N,A	N,M
A	(Additive)	A,N	A,A	A,M
A_d	(Additive damped)	A_d,N	A_d,A	A_d,M
M	(Multiplicative)	M,N	M,A	M,M
M_d	(Multiplicative damped)	M_d,N	M_d,A	M_d,M

General notation E T S : **E**xponen**T**ial **S**moothing

↑ **T**rend ↖ **S**easonal

Examples:

A,N,N: Simple exponential smoothing with additive errors

A,A,N: Holt's linear method with additive errors

M,A,M: Multiplicative Holt-Winters' method with multiplicative errors

Exponential smoothing methods

Trend Component		Seasonal Component N (None)	A (Additive)	M (Multiplicative)
N	(None)	N,N	N,A	N,M
A	(Additive)	A,N	A,A	A,M
A_d	(Additive damped)	A_d,N	A_d,A	A_d,M
M	(Multiplicative)	M,N	M,A	M,M
M_d	(Multiplicative damped)	M_d,N	M_d,A	M_d,M

General notation E T S : **E**xponen**T**ial **S**moothing

↗ ↑ ↖

Error **T**rend **S**easonal

Examples:

A,N,N: Simple exponential smoothing with additive errors
A,A,N: Holt's linear method with additive errors
M,A,M: Multiplicative Holt-Winters' method with multiplicative errors

Exponential smoothing methods

Trend Component		Seasonal Component N (None)	A (Additive)	M (Multiplicative)
N	(None)	N,N	N,A	N,M
A	(Additive)	A,N	A,A	A,M
A_d	(Additive damped)	A_d,N	A_d,A	A_d,M
M	(Multiplicative)	M,N	M,A	M,M
M_d	(Multiplicative damped)	M_d,N	M_d,A	M_d,M

General notation E T S : **E**xponen**T**ial **S**moothing

↗ ↑ ↖

Error **T**rend **S**easonal

Examples:

A,N,N: Simple exponential smoothing with additive errors

A,A,N: Holt's linear method with additive errors

M,A,M: Multiplicative Holt-Winters' method with multiplicative errors

Exponential smoothing methods

Innovations state space models

- All ETS models can be written in innovations state space form.
- Additive and multiplicative versions give the same point forecasts but different prediction intervals.

General notation E T S : **E**xponen**T**ial **S**moothing

↗ ↑ ↖

Error **T**rend **S**easonal

Examples:

A,N,N:	Simple exponential smoothing with additive errors
A,A,N:	Holt's linear method with additive errors
M,A,M:	Multiplicative Holt-Winters' method with multiplicative errors

Innovations state space models

Let $\boldsymbol{x}_t = (\ell_t, b_t, s_t, s_{t-1}, \dots, s_{t-m+1})$ and $\varepsilon_t \overset{\text{iid}}{\sim} \mathrm{N}(0, \sigma^2)$.

$$\begin{aligned} y_t &= \underbrace{h(\boldsymbol{x}_{t-1})}_{\mu_t} + \underbrace{k(\boldsymbol{x}_{t-1})\varepsilon_t}_{e_t} \\ \boldsymbol{x}_t &= f(\boldsymbol{x}_{t-1}) + g(\boldsymbol{x}_{t-1})\varepsilon_t \end{aligned}$$

Additive errors:

$$k(x) = 1. \qquad y_t = \mu_t + \varepsilon_t.$$

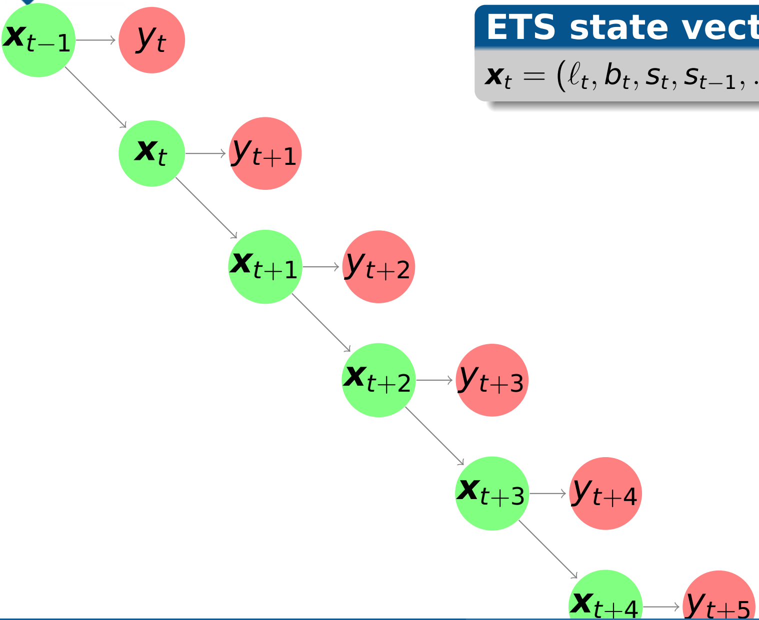
Multiplicative errors:

$$k(\boldsymbol{x}_{t-1}) = \mu_t. \qquad y_t = \mu_t(1 + \varepsilon_t).$$

$\varepsilon_t = (y_t - \mu_t)/\mu_t$ is relative error.

Outline

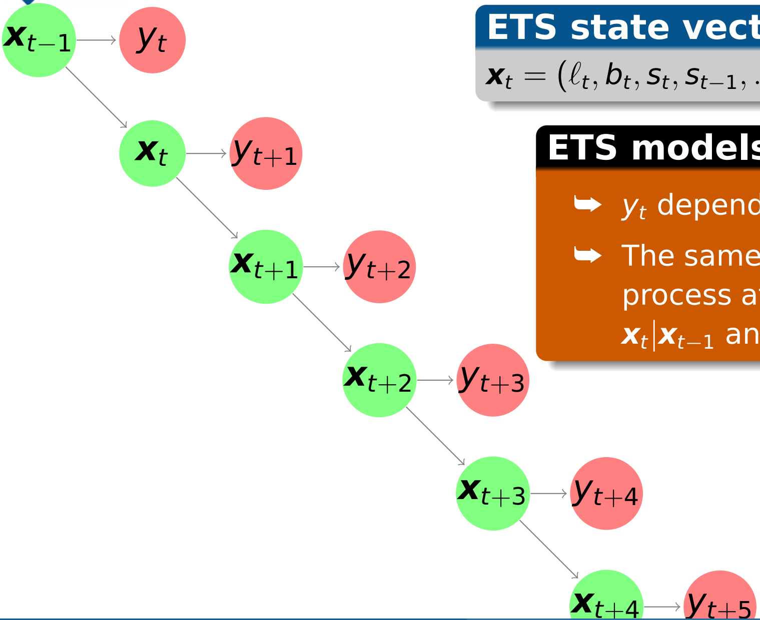
State space models



ETS state vector

$$\boldsymbol{x}_t = (\ell_t, b_t, s_t, s_{t-1}, \dots, s_{t-m+1})$$

State space models



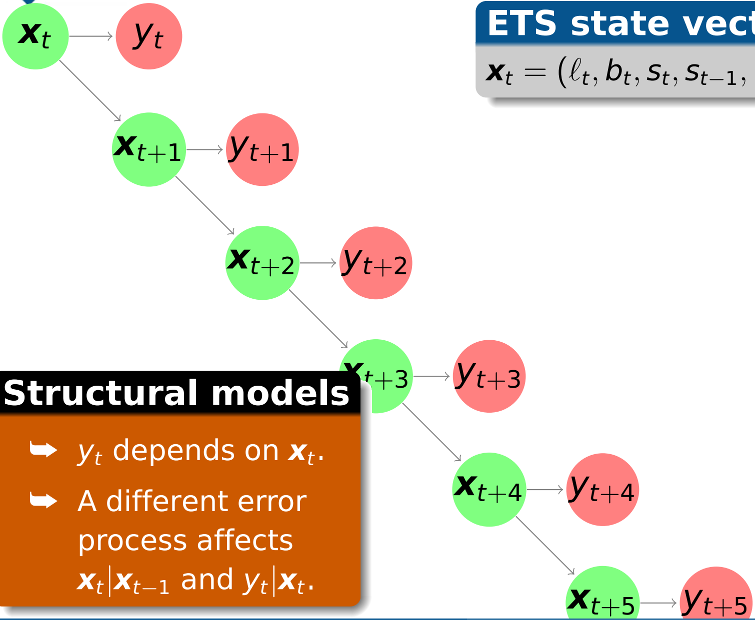
ETS state vector

$$\boldsymbol{x}_t = (\ell_t, b_t, s_t, s_{t-1}, \dots, s_{t-m+1})$$

ETS models

- y_t depends on $\boldsymbol{x}_{t-1}$.
- The same error process affects $\boldsymbol{x}_t|\boldsymbol{x}_{t-1}$ and $y_t|\boldsymbol{x}_{t-1}$.

State space models



ETS state vector

$$\boldsymbol{x}_t = (\ell_t, b_t, s_t, s_{t-1}, \dots, s_{t-m+1})$$

Structural models

- y_t depends on $\boldsymbol{x}_t$.
- A different error process affects $\boldsymbol{x}_t|\boldsymbol{x}_{t-1}$ and $y_t|\boldsymbol{x}_t$.

Local level model

Stochastically varying level (random walk) observed with noise

$$y_t = \ell_t + \varepsilon_t$$
$$\ell_t = \ell_{t-1} + \xi_t$$

- ε_t and ξ_t are independent Gaussian white noise processes.
- Compare ETS(A,N,N) where $\xi_t = \alpha\varepsilon_{t-1}$.
- Parameters to estimate: σ^2_ε and σ^2_ξ.
- If $\sigma^2_\xi = 0$, $y_t \sim \text{NID}(\ell_0, \sigma^2_\varepsilon)$.

Local level model

Stochastically varying level (random walk) observed with noise

$$y_t = \ell_t + \varepsilon_t$$
$$\ell_t = \ell_{t-1} + \xi_t$$

- ε_t and ξ_t are independent Gaussian white noise processes.
- Compare ETS(A,N,N) where $\xi_t = \alpha\varepsilon_{t-1}$.
- Parameters to estimate: σ^2_ε and σ^2_ξ.
- If $\sigma^2_\xi = 0$, $y_t \sim \text{NID}(\ell_0, \sigma^2_\varepsilon)$.

Local level model

Stochastically varying level (random walk) observed with noise

$$y_t = \ell_t + \varepsilon_t$$
$$\ell_t = \ell_{t-1} + \xi_t$$

- ε_t and ξ_t are independent Gaussian white noise processes.
- Compare ETS(A,N,N) where $\xi_t = \alpha\varepsilon_{t-1}$.
- Parameters to estimate: σ^2_ε and σ^2_ξ.
- If $\sigma^2_\xi = 0$, $y_t \sim \text{NID}(\ell_0, \sigma^2_\varepsilon)$.

Local level model

Stochastically varying level (random walk) observed with noise

$$y_t = \ell_t + \varepsilon_t$$
$$\ell_t = \ell_{t-1} + \xi_t$$

- ε_t and ξ_t are independent Gaussian white noise processes.
- Compare ETS(A,N,N) where $\xi_t = \alpha\varepsilon_{t-1}$.
- Parameters to estimate: σ_ε^2 and σ_ξ^2.
- If $\sigma_\xi^2 = 0$, $y_t \sim \text{NID}(\ell_0, \sigma_\varepsilon^2)$.

Local linear trend model

Dynamic trend observed with noise

$$y_t = \ell_t + \varepsilon_t$$
$$\ell_t = \ell_{t-1} + b_{t-1} + \xi_t$$
$$b_t = b_{t-1} + \zeta_t$$

- ε_t, ξ_t and ζ_t are independent Gaussian white noise processes.
- Compare ETS(A,A,N) where $\xi_t = (\alpha + \beta)\varepsilon_{t-1}$ and $\zeta_t = \beta\varepsilon_{t-1}$
- Parameters to estimate: σ^2_ε, σ^2_ξ, and σ^2_ζ.
- If $\sigma^2_\zeta = \sigma^2_\xi = 0$, $y_t = \ell_0 + tb_0 + \varepsilon_t$.
- Model is a time-varying linear regression.

Local linear trend model

Dynamic trend observed with noise

$$y_t = \ell_t + \varepsilon_t$$
$$\ell_t = \ell_{t-1} + b_{t-1} + \xi_t$$
$$b_t = b_{t-1} + \zeta_t$$

- ε_t, ξ_t and ζ_t are independent Gaussian white noise processes.
- Compare ETS(A,A,N) where $\xi_t = (\alpha + \beta)\varepsilon_{t-1}$ and $\zeta_t = \beta\varepsilon_{t-1}$
- Parameters to estimate: σ^2_ε, σ^2_ξ, and σ^2_ζ.
- If $\sigma^2_\zeta = \sigma^2_\xi = 0$, $y_t = \ell_0 + tb_0 + \varepsilon_t$.
- Model is a time-varying linear regression.

Local linear trend model

Dynamic trend observed with noise

$$
\begin{aligned}
y_t &= \ell_t + \varepsilon_t \\
\ell_t &= \ell_{t-1} + b_{t-1} + \xi_t \\
b_t &= b_{t-1} + \zeta_t
\end{aligned}
$$

- ε_t, ξ_t and ζ_t are independent Gaussian white noise processes.
- Compare ETS(A,A,N) where $\xi_t = (\alpha + \beta)\varepsilon_{t-1}$ and $\zeta_t = \beta\varepsilon_{t-1}$
- Parameters to estimate: σ^2_ε, σ^2_ξ, and σ^2_ζ.
- If $\sigma^2_\zeta = \sigma^2_\xi = 0$, $y_t = \ell_0 + tb_0 + \varepsilon_t$.
- Model is a time-varying linear regression.

Local linear trend model

Dynamic trend observed with noise

$$y_t = \ell_t + \varepsilon_t$$
$$\ell_t = \ell_{t-1} + b_{t-1} + \xi_t$$
$$b_t = b_{t-1} + \zeta_t$$

- ε_t, ξ_t and ζ_t are independent Gaussian white noise processes.
- Compare ETS(A,A,N) where $\xi_t = (\alpha + \beta)\varepsilon_{t-1}$ and $\zeta_t = \beta\varepsilon_{t-1}$
- Parameters to estimate: σ^2_ε, σ^2_ξ, and σ^2_ζ.
- If $\sigma^2_\zeta = \sigma^2_\xi = 0$, $y_t = \ell_0 + tb_0 + \varepsilon_t$.
- Model is a time-varying linear regression.

Local linear trend model

Dynamic trend observed with noise

$$y_t = \ell_t + \varepsilon_t$$
$$\ell_t = \ell_{t-1} + b_{t-1} + \xi_t$$
$$b_t = b_{t-1} + \zeta_t$$

- ε_t, ξ_t and ζ_t are independent Gaussian white noise processes.
- Compare ETS(A,A,N) where $\xi_t = (\alpha + \beta)\varepsilon_{t-1}$ and $\zeta_t = \beta\varepsilon_{t-1}$
- Parameters to estimate: σ^2_ε, σ^2_ξ, and σ^2_ζ.
- If $\sigma^2_\zeta = \sigma^2_\xi = 0$, $y_t = \ell_0 + tb_0 + \varepsilon_t$.
- Model is a time-varying linear regression.

Basic structural model

$$
\begin{aligned}
y_t &= \ell_t + s_{1,t} + \varepsilon_t \\
\ell_t &= \ell_{t-1} + b_{t-1} + \xi_t \\
b_t &= b_{t-1} + \zeta_t \\
s_{1,t} &= -\sum_{j=1}^{m-1} s_{j,t-1} + \eta_t \\
s_{j,t} &= s_{j-1,t-1}, \qquad j = 2,\dots,m-1
\end{aligned}
$$

- ε_t, ξ_t, ζ_t and η_t are independent Gaussian white noise processes.
- Compare ETS(A,A,A).
- Parameters to estimate: σ_ε^2, σ_ξ^2, σ_ζ^2 and σ_η^2
- Deterministic seasonality if $\sigma_\eta^2 = 0$.

Basic structural model

$$
\begin{aligned}
y_t &= \ell_t + s_{1,t} + \varepsilon_t \\
\ell_t &= \ell_{t-1} + b_{t-1} + \xi_t \\
b_t &= b_{t-1} + \zeta_t \\
s_{1,t} &= -\sum_{j=1}^{m-1} s_{j,t-1} + \eta_t \\
s_{j,t} &= s_{j-1,t-1}, \qquad j = 2, \dots, m-1
\end{aligned}
$$

- ε_t, ξ_t, ζ_t and η_t are independent Gaussian white noise processes.
- Compare ETS(A,A,A).
- Parameters to estimate: σ_ε^2, σ_ξ^2, σ_ζ^2 and σ_η^2
- Deterministic seasonality if $\sigma_\eta^2 = 0$.

Basic structural model

$$
\begin{aligned}
y_t &= \ell_t + s_{1,t} + \varepsilon_t \\
\ell_t &= \ell_{t-1} + b_{t-1} + \xi_t \\
b_t &= b_{t-1} + \zeta_t \\
s_{1,t} &= -\sum_{j=1}^{m-1} s_{j,t-1} + \eta_t \\
s_{j,t} &= s_{j-1,t-1}, \qquad j = 2,\dots,m-1
\end{aligned}
$$

- ε_t, ξ_t, ζ_t and η_t are independent Gaussian white noise processes.
- Compare ETS(A,A,A).
- Parameters to estimate: σ_ε^2, σ_ξ^2, σ_ζ^2 and σ_η^2
- Deterministic seasonality if $\sigma_\eta^2 = 0$.

Basic structural model

$$
\begin{aligned}
y_t &= \ell_t + s_{1,t} + \varepsilon_t \\
\ell_t &= \ell_{t-1} + b_{t-1} + \xi_t \\
b_t &= b_{t-1} + \zeta_t \\
s_{1,t} &= -\sum_{j=1}^{m-1} s_{j,t-1} + \eta_t \\
s_{j,t} &= s_{j-1,t-1}, \qquad j = 2, \dots, m-1
\end{aligned}
$$

- ε_t, ξ_t, ζ_t and η_t are independent Gaussian white noise processes.
- Compare ETS(A,A,A).
- Parameters to estimate: σ_ε^2, σ_ξ^2, σ_ζ^2 and σ_η^2
- Deterministic seasonality if $\sigma_\eta^2 = 0$.

Trigonometric models

$$y_t = \ell_t + \sum_{j=1}^{J} s_{j,t} + \varepsilon_t$$
$$\ell_t = \ell_{t-1} + b_{t-1} + \xi_t$$
$$b_t = b_{t-1} + \zeta_t$$
$$s_{j,t} = \cos\lambda_j s_{j,t-1} + \sin\lambda_j s^*_{j,t-1} + \omega_{j,t}$$
$$s^*_{j,t} = -\sin\lambda_j s_{j,t-1} + \cos\lambda_j s^*_{j,t-1} + \omega^*_{j,t}$$

- $\lambda_j = 2\pi j/m$
- ε_t, ξ_t, ζ_t, $\omega_{j,t}$, $\omega^*_{j,t}$ are independent Gaussian white noise processes
- $\omega_{j,t}$ and $\omega^*_{j,t}$ have same variance $\sigma^2_{\omega,j}$
- Equivalent to BSM when $\sigma^2_{\omega,j} = \sigma^2_\omega$ and $J = m/2$
- Choose $J < m/2$ for fewer degrees of freedom

Trigonometric models

$$
\begin{aligned}
y_t &= \ell_t + \sum_{j=1}^{J} s_{j,t} + \varepsilon_t \\
\ell_t &= \ell_{t-1} + b_{t-1} + \xi_t \\
b_t &= b_{t-1} + \zeta_t \\
s_{j,t} &= \quad \cos\lambda_j s_{j,t-1} + \sin\lambda_j s^*_{j,t-1} + \omega_{j,t} \\
s^*_{j,t} &= -\sin\lambda_j s_{j,t-1} + \cos\lambda_j s^*_{j,t-1} + \omega^*_{j,t}
\end{aligned}
$$

- $\lambda_j = 2\pi j/m$
- ε_t, ξ_t, ζ_t, $\omega_{j,t}$, $\omega^*_{j,t}$ are independent Gaussian white noise processes
- $\omega_{j,t}$ and $\omega^*_{j,t}$ have same variance $\sigma^2_{\omega,j}$
- Equivalent to BSM when $\sigma^2_{\omega,j} = \sigma^2_{\omega}$ and $J = m/2$
- Choose $J < m/2$ for fewer degrees of freedom

Trigonometric models

$$
\begin{aligned}
y_t &= \ell_t + \sum_{j=1}^{J} s_{j,t} + \varepsilon_t \\
\ell_t &= \ell_{t-1} + b_{t-1} + \xi_t \\
b_t &= b_{t-1} + \zeta_t \\
s_{j,t} &= \quad \cos\lambda_j s_{j,t-1} + \sin\lambda_j s^*_{j,t-1} + \omega_{j,t} \\
s^*_{j,t} &= -\sin\lambda_j s_{j,t-1} + \cos\lambda_j s^*_{j,t-1} + \omega^*_{j,t}
\end{aligned}
$$

- $\lambda_j = 2\pi j/m$
- $\varepsilon_t, \xi_t, \zeta_t, \omega_{j,t}, \omega^*_{j,t}$ are independent Gaussian white noise processes
- $\omega_{j,t}$ and $\omega^*_{j,t}$ have same variance $\sigma^2_{\omega,j}$
- Equivalent to BSM when $\sigma^2_{\omega,j} = \sigma^2_{\omega}$ and $J = m/2$
- Choose $J < m/2$ for fewer degrees of freedom

Trigonometric models

$$y_t = \ell_t + \sum_{j=1}^{J} s_{j,t} + \varepsilon_t$$

$$\ell_t = \ell_{t-1} + b_{t-1} + \xi_t$$

$$b_t = b_{t-1} + \zeta_t$$

$$s_{j,t} = \quad \cos\lambda_j s_{j,t-1} + \sin\lambda_j s^*_{j,t-1} + \omega_{j,t}$$

$$s^*_{j,t} = -\sin\lambda_j s_{j,t-1} + \cos\lambda_j s^*_{j,t-1} + \omega^*_{j,t}$$

- $\lambda_j = 2\pi j/m$
- ε_t, ξ_t, ζ_t, $\omega_{j,t}$, $\omega^*_{j,t}$ are independent Gaussian white noise processes
- $\omega_{j,t}$ and $\omega^*_{j,t}$ have same variance $\sigma^2_{\omega,j}$
- Equivalent to BSM when $\sigma^2_{\omega,j} = \sigma^2_\omega$ and $J = m/2$
- Choose $J < m/2$ for fewer degrees of freedom

Trigonometric models

$$
\begin{aligned}
y_t &= \ell_t + \sum_{j=1}^{J} s_{j,t} + \varepsilon_t \\
\ell_t &= \ell_{t-1} + b_{t-1} + \xi_t \\
b_t &= b_{t-1} + \zeta_t \\
s_{j,t} &= \quad \cos\lambda_j s_{j,t-1} + \sin\lambda_j s^*_{j,t-1} + \omega_{j,t} \\
s^*_{j,t} &= -\sin\lambda_j s_{j,t-1} + \cos\lambda_j s^*_{j,t-1} + \omega^*_{j,t}
\end{aligned}
$$

- $\lambda_j = 2\pi j/m$
- ε_t, ξ_t, ζ_t, $\omega_{j,t}$, $\omega^*_{j,t}$ are independent Gaussian white noise processes
- $\omega_{j,t}$ and $\omega^*_{j,t}$ have same variance $\sigma^2_{\omega,j}$
- Equivalent to BSM when $\sigma^2_{\omega,j} = \sigma^2_{\omega}$ and $J = m/2$
- Choose $J < m/2$ for fewer degrees of freedom

ETS vs Structural models

- ETS models are much more general as they allow non-linear (multiplicative components).
- ETS allows automatic forecasting due to its larger model space.
- Additive ETS models are almost equivalent to the corresponding structural models.
- ETS models have a larger parameter space. Structural models parameters are always non-negative (variances).
- Structural models are much easier to generalize (e.g., add covariates).
- It is easier to handle missing values with structural models.

ETS vs Structural models

- ETS models are much more general as they allow non-linear (multiplicative components).
- ETS allows automatic forecasting due to its larger model space.
- Additive ETS models are almost equivalent to the corresponding structural models.
- ETS models have a larger parameter space. Structural models parameters are always non-negative (variances).
- Structural models are much easier to generalize (e.g., add covariates).
- It is easier to handle missing values with structural models.

ETS vs Structural models

- ETS models are much more general as they allow non-linear (multiplicative components).
- ETS allows automatic forecasting due to its larger model space.
- Additive ETS models are almost equivalent to the corresponding structural models.
- ETS models have a larger parameter space. Structural models parameters are always non-negative (variances).
- Structural models are much easier to generalize (e.g., add covariates).
- It is easier to handle missing values with structural models.

ETS vs Structural models

- ETS models are much more general as they allow non-linear (multiplicative components).
- ETS allows automatic forecasting due to its larger model space.
- Additive ETS models are almost equivalent to the corresponding structural models.
- ETS models have a larger parameter space. Structural models parameters are always non-negative (variances).
- Structural models are much easier to generalize (e.g., add covariates).
- It is easier to handle missing values with structural models.

ETS vs Structural models

- ETS models are much more general as they allow non-linear (multiplicative components).
- ETS allows automatic forecasting due to its larger model space.
- Additive ETS models are almost equivalent to the corresponding structural models.
- ETS models have a larger parameter space. Structural models parameters are always non-negative (variances).
- Structural models are much easier to generalize (e.g., add covariates).
- It is easier to handle missing values with structural models.

ETS vs Structural models

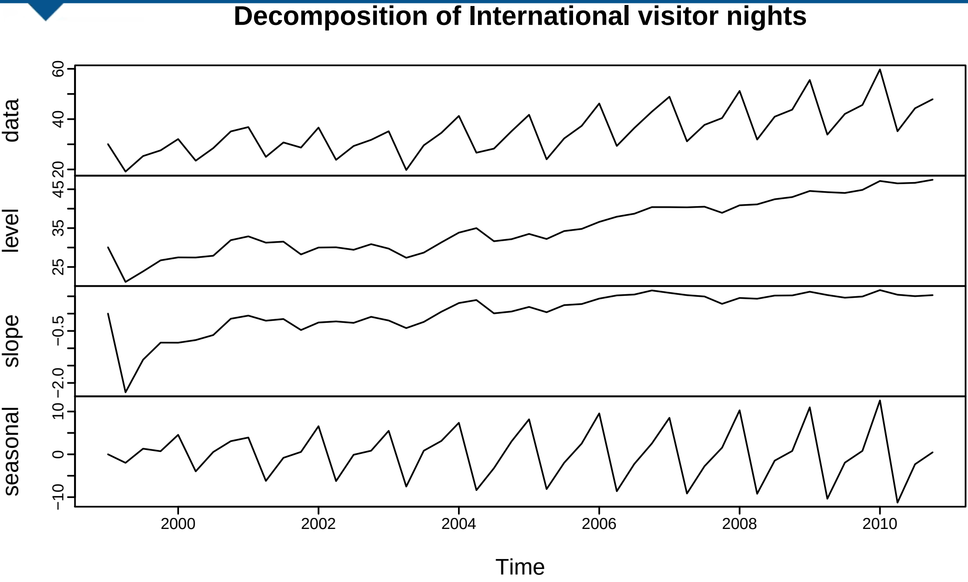
- ETS models are much more general as they allow non-linear (multiplicative components).
- ETS allows automatic forecasting due to its larger model space.
- Additive ETS models are almost equivalent to the corresponding structural models.
- ETS models have a larger parameter space. Structural models parameters are always non-negative (variances).
- Structural models are much easier to generalize (e.g., add covariates).
- It is easier to handle missing values with structural models.

Structural models in R

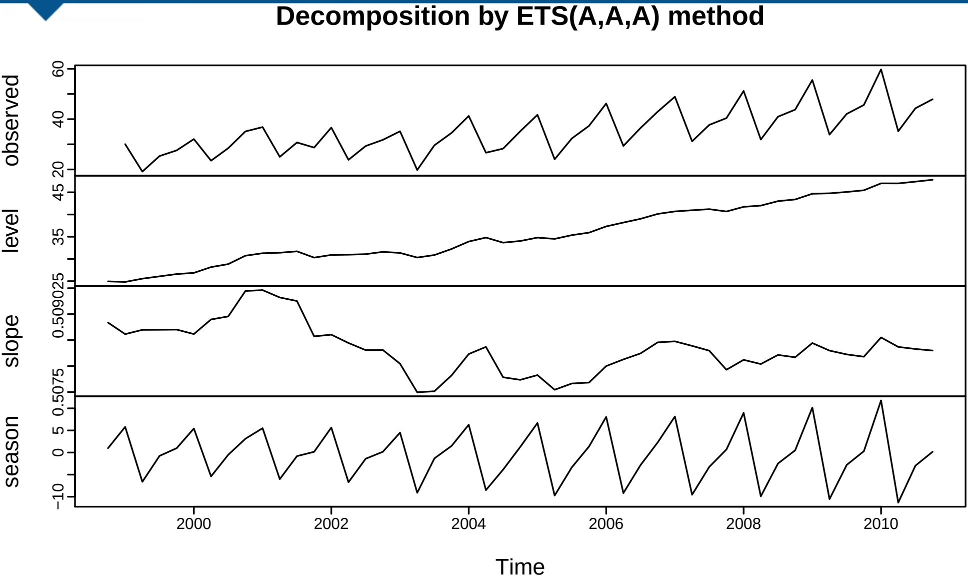
```
StructTS(oil, type="level")
StructTS(ausair, type="trend")
StructTS(austourists, type="BSM")

fit <- StructTS(austourists, type = "BSM")
decomp <- cbind(austourists, fitted(fit))
colnames(decomp) <- c("data","level","slope",
   "seasonal")
plot(decomp, main="Decomposition of
  International visitor nights")
```

Structural models in R



ETS decomposition



Outline

Linear Gaussian SS models

Observation equation	$y_t = \boldsymbol{f}'\boldsymbol{x}_t + \varepsilon_t$
State equation	$\boldsymbol{x}_t = \boldsymbol{G}\boldsymbol{x}_{t-1} + \boldsymbol{w}_t$

- State vector $\boldsymbol{x}_t$ of length p
- $\boldsymbol{G}$ a $p \times p$ matrix, $\boldsymbol{f}$ a vector of length p
- $\varepsilon_t \sim \text{NID}(0, \sigma^2)$, $\quad \boldsymbol{w}_t \sim \text{NID}(0, \boldsymbol{W})$.

Local level model:

$\boldsymbol{f} = \boldsymbol{G} = 1, \qquad \boldsymbol{x}_t = \ell_t.$

Local linear trend model:

$\boldsymbol{f}' = [1\ 0],$

$$\boldsymbol{x}_t = \begin{bmatrix} \ell_t \\ b_t \end{bmatrix} \qquad \boldsymbol{G} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \qquad \boldsymbol{W} = \begin{bmatrix} \sigma_\xi^2 & 0 \\ 0 & \sigma_\zeta^2 \end{bmatrix}$$

Linear Gaussian SS models

$$\text{Observation equation} \qquad y_t = \boldsymbol{f}'\boldsymbol{x}_t + \varepsilon_t$$
$$\text{State equation} \qquad \boldsymbol{x}_t = \boldsymbol{G}\boldsymbol{x}_{t-1} + \boldsymbol{w}_t$$

- State vector $\boldsymbol{x}_t$ of length p
- $\boldsymbol{G}$ a $p \times p$ matrix, $\boldsymbol{f}$ a vector of length p
- $\varepsilon_t \sim \text{NID}(0, \sigma^2)$, $\quad \boldsymbol{w}_t \sim \text{NID}(\boldsymbol{0}, \boldsymbol{W})$.

Local level model:

$\boldsymbol{f} = \boldsymbol{G} = 1, \qquad \boldsymbol{x}_t = \ell_t.$

Local linear trend model:

$\boldsymbol{f}' = [1\ 0],$

$$\boldsymbol{x}_t = \begin{bmatrix} \ell_t \\ b_t \end{bmatrix} \qquad \boldsymbol{G} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \qquad \boldsymbol{W} = \begin{bmatrix} \sigma_\xi^2 & 0 \\ 0 & \sigma_\zeta^2 \end{bmatrix}$$

Linear Gaussian SS models

Observation equation	$y_t = \boldsymbol{f}'\boldsymbol{x}_t + \varepsilon_t$
State equation	$\boldsymbol{x}_t = \boldsymbol{G}\boldsymbol{x}_{t-1} + \boldsymbol{w}_t$

- State vector $\boldsymbol{x}_t$ of length p
- $\boldsymbol{G}$ a $p \times p$ matrix, $\boldsymbol{f}$ a vector of length p
- $\varepsilon_t \sim \text{NID}(0, \sigma^2)$, $\quad \boldsymbol{w}_t \sim \text{NID}(\boldsymbol{0}, \boldsymbol{W})$.

Local level model:

$\boldsymbol{f} = \boldsymbol{G} = 1, \qquad \boldsymbol{x}_t = \ell_t.$

Local linear trend model:

$\boldsymbol{f}' = [1\ 0],$

$$\boldsymbol{x}_t = \begin{bmatrix} \ell_t \\ b_t \end{bmatrix} \qquad \boldsymbol{G} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \qquad \boldsymbol{W} = \begin{bmatrix} \sigma_\xi^2 & 0 \\ 0 & \sigma_\zeta^2 \end{bmatrix}$$

Linear Gaussian SS models

Observation equation $\quad y_t = \boldsymbol{f}'\boldsymbol{x}_t + \varepsilon_t$

State equation $\quad \boldsymbol{x}_t = \boldsymbol{G}\boldsymbol{x}_{t-1} + \boldsymbol{w}_t$

- State vector $\boldsymbol{x}_t$ of length p
- $\boldsymbol{G}$ a $p \times p$ matrix, $\boldsymbol{f}$ a vector of length p
- $\varepsilon_t \sim \text{NID}(0, \sigma^2)$, $\quad \boldsymbol{w}_t \sim \text{NID}(\boldsymbol{0}, \boldsymbol{W})$.

Local level model:

$\boldsymbol{f} = \boldsymbol{G} = 1, \qquad \boldsymbol{x}_t = \ell_t.$

Local linear trend model:

$\boldsymbol{f}' = [1\ 0],$

$$\boldsymbol{x}_t = \begin{bmatrix} \ell_t \\ b_t \end{bmatrix} \qquad \boldsymbol{G} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \qquad \boldsymbol{W} = \begin{bmatrix} \sigma_\xi^2 & 0 \\ 0 & \sigma_\zeta^2 \end{bmatrix}$$

Linear Gaussian SS models

Observation equation	$y_t = \boldsymbol{f}'\boldsymbol{x}_t + \varepsilon_t$
State equation	$\boldsymbol{x}_t = \boldsymbol{G}\boldsymbol{x}_{t-1} + \boldsymbol{w}_t$

- State vector $\boldsymbol{x}_t$ of length p
- $\boldsymbol{G}$ a $p \times p$ matrix, $\boldsymbol{f}$ a vector of length p
- $\varepsilon_t \sim \text{NID}(0, \sigma^2)$, $\quad \boldsymbol{w}_t \sim \text{NID}(\boldsymbol{0}, \boldsymbol{W})$.

Local level model:

$\boldsymbol{f} = \boldsymbol{G} = 1, \qquad \boldsymbol{x}_t = \ell_t.$

Local linear trend model:

$\boldsymbol{f}' = [1\ 0],$

$$\boldsymbol{x}_t = \begin{bmatrix} \ell_t \\ b_t \end{bmatrix} \qquad \boldsymbol{G} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \qquad \boldsymbol{W} = \begin{bmatrix} \sigma_\xi^2 & 0 \\ 0 & \sigma_\zeta^2 \end{bmatrix}$$

Linear Gaussian SS models

Observation equation	$y_t = \boldsymbol{f}'\boldsymbol{x}_t + \varepsilon_t$
State equation	$\boldsymbol{x}_t = \boldsymbol{G}\boldsymbol{x}_{t-1} + \boldsymbol{w}_t$

- State vector $\boldsymbol{x}_t$ of length p
- $\boldsymbol{G}$ a $p \times p$ matrix, $\boldsymbol{f}$ a vector of length p
- $\varepsilon_t \sim \text{NID}(0, \sigma^2)$, $\boldsymbol{w}_t \sim \text{NID}(\boldsymbol{0}, \boldsymbol{W})$.

Local level model:

$$\boldsymbol{f} = \boldsymbol{G} = 1, \qquad \boldsymbol{x}_t = \ell_t.$$

Local linear trend model:

$$\boldsymbol{f}' = [1\ 0],$$

$$\boldsymbol{x}_t = \begin{bmatrix} \ell_t \\ b_t \end{bmatrix} \qquad \boldsymbol{G} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \qquad \boldsymbol{W} = \begin{bmatrix} \sigma_\xi^2 & 0 \\ 0 & \sigma_\zeta^2 \end{bmatrix}$$

Linear Gaussian SS models

Observation equation	$y_t = \boldsymbol{f}'\boldsymbol{x}_t + \varepsilon_t$
State equation	$\boldsymbol{x}_t = \boldsymbol{G}\boldsymbol{x}_{t-1} + \boldsymbol{w}_t$

- State vector $\boldsymbol{x}_t$ of length p
- $\boldsymbol{G}$ a $p \times p$ matrix, $\boldsymbol{f}$ a vector of length p
- $\varepsilon_t \sim \text{NID}(0, \sigma^2)$, $\quad \boldsymbol{w}_t \sim \text{NID}(\boldsymbol{0}, \boldsymbol{W})$.

Local level model:

$$\boldsymbol{f} = \boldsymbol{G} = 1, \qquad \boldsymbol{x}_t = \ell_t.$$

Local linear trend model:

$$\boldsymbol{f}' = [1\ 0],$$

$$\boldsymbol{x}_t = \begin{bmatrix} \ell_t \\ b_t \end{bmatrix} \qquad \boldsymbol{G} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \qquad \boldsymbol{W} = \begin{bmatrix} \sigma_\xi^2 & 0 \\ 0 & \sigma_\zeta^2 \end{bmatrix}$$

Basic structural model

Linear Gaussian state space model

$$y_t = \boldsymbol{f}'\boldsymbol{x}_t + \varepsilon_t, \qquad \varepsilon_t \sim \mathrm{N}(0, \sigma^2)$$

$$\boldsymbol{x}_t = \boldsymbol{G}\boldsymbol{x}_{t-1} + \boldsymbol{w}_t \qquad \boldsymbol{w}_t \sim \mathrm{N}(\boldsymbol{0}, \boldsymbol{W})$$

$$\boldsymbol{f}' = [1\ 0\ 1\ 0\ \cdots\ 0], \quad \boldsymbol{W} = \text{diagonal}(\sigma^2_\xi, \sigma^2_\zeta, \sigma^2_\eta, 0, \ldots, 0)$$

$$\boldsymbol{x}_t = \begin{bmatrix} \ell_t \\ b_t \\ s_{1,t} \\ s_{2,t} \\ s_{3,t} \\ \vdots \\ s_{m-1,t} \end{bmatrix} \quad \boldsymbol{G} = \begin{bmatrix} 1 & 1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & -1 & -1 & \ldots & -1 & -1 \\ 0 & 0 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & 1 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & 0 & 0 \\ 0 & 0 & 0 & \ldots & 0 & 1 & 0 \end{bmatrix}$$

Basic structural model

Linear Gaussian state space model

$$y_t = \boldsymbol{f}'\boldsymbol{x}_t + \varepsilon_t, \qquad \varepsilon_t \sim \text{N}(0, \sigma^2)$$

$$\boldsymbol{x}_t = \boldsymbol{G}\boldsymbol{x}_{t-1} + \boldsymbol{w}_t \qquad \boldsymbol{w}_t \sim \text{N}(\boldsymbol{0}, \boldsymbol{W})$$

$$\boldsymbol{f}' = [1\ 0\ 1\ 0\ \ \cdots\ \ 0], \quad \boldsymbol{W} = \text{diagonal}(\sigma_\xi^2, \sigma_\zeta^2, \sigma_\eta^2, 0, \dots, 0)$$

$$\boldsymbol{x}_t = \begin{bmatrix} \ell_t \\ b_t \\ s_{1,t} \\ s_{2,t} \\ s_{3,t} \\ \vdots \\ s_{m-1,t} \end{bmatrix} \quad \boldsymbol{G} = \begin{bmatrix} 1 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & -1 & -1 & \dots & -1 & -1 \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 1 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 & 0 \end{bmatrix}$$

Outline

Kalman filter

Notation:

$$\hat{\boldsymbol{x}}_{t|t} = \mathrm{E}[\boldsymbol{x}_t | y_1, \dots, y_t] \qquad \hat{\boldsymbol{P}}_{t|t} = \mathrm{Var}[\boldsymbol{x}_t | y_1, \dots, y_t]$$

$$\hat{\boldsymbol{x}}_{t|t-1} = \mathrm{E}[\boldsymbol{x}_t | y_1, \dots, y_{t-1}] \qquad \hat{\boldsymbol{P}}_{t|t-1} = \mathrm{Var}[\boldsymbol{x}_t | y_1, \dots, y_{t-1}]$$

$$\hat{y}_{t|t-1} = \mathrm{E}[y_t | y_1, \dots, y_{t-1}] \qquad \hat{v}_{t|t-1} = \mathrm{Var}[y_t | y_1, \dots, y_{t-1}]$$

Forecasting:

$$\hat{y}_{t|t-1} = \boldsymbol{f}'\hat{\boldsymbol{x}}_{t|t-1}$$

$$\hat{v}_{t|t-1} = \boldsymbol{f}'\hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f} + \sigma^2$$

Updating or State Filtering:

$$\hat{\boldsymbol{x}}_{t|t} = \hat{\boldsymbol{x}}_{t|t-1} + \hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f}\hat{v}_{t|t-1}^{-1}(y_t - \hat{y}_{t|t-1})$$

$$\hat{\boldsymbol{P}}_{t|t} = \hat{\boldsymbol{P}}_{t|t-1} - \hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f}\hat{v}_{t|t-1}^{-1}\boldsymbol{f}'\hat{\boldsymbol{P}}_{t|t-1}$$

State Prediction

$$\hat{\boldsymbol{x}}_{t+1|t} = \boldsymbol{G}\hat{\boldsymbol{x}}_{t|t}$$

$$\hat{\boldsymbol{P}}_{t+1|t} = \boldsymbol{G}\hat{\boldsymbol{P}}_{t|t}\boldsymbol{G}' + \boldsymbol{W}$$

Kalman filter

Notation:

$$\hat{\boldsymbol{x}}_{t|t} = \mathrm{E}[\boldsymbol{x}_t|y_1,\dots,y_t] \qquad \hat{\boldsymbol{P}}_{t|t} = \mathrm{Var}[\boldsymbol{x}_t|y_1,\dots,y_t]$$

$$\hat{\boldsymbol{x}}_{t|t-1} = \mathrm{E}[\boldsymbol{x}_t|y_1,\dots,y_{t-1}] \qquad \hat{\boldsymbol{P}}_{t|t-1} = \mathrm{Var}[\boldsymbol{x}_t|y_1,\dots,y_{t-1}]$$

$$\hat{y}_{t|t-1} = \mathrm{E}[y_t|y_1,\dots,y_{t-1}] \qquad \hat{v}_{t|t-1} = \mathrm{Var}[y_t|y_1,\dots,y_{t-1}]$$

Forecasting:

$$\hat{y}_{t|t-1} = \boldsymbol{f}'\hat{\boldsymbol{x}}_{t|t-1}$$

$$\hat{v}_{t|t-1} = \boldsymbol{f}'\hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f} + \sigma^2$$

Updating or State Filtering:

$$\hat{\boldsymbol{x}}_{t|t} = \hat{\boldsymbol{x}}_{t|t-1} + \hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f}\hat{v}_{t|t-1}^{-1}(y_t - \hat{y}_{t|t-1})$$

$$\hat{\boldsymbol{P}}_{t|t} = \hat{\boldsymbol{P}}_{t|t-1} - \hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f}\hat{v}_{t|t-1}^{-1}\boldsymbol{f}'\hat{\boldsymbol{P}}_{t|t-1}$$

State Prediction

$$\hat{\boldsymbol{x}}_{t+1|t} = \boldsymbol{G}\hat{\boldsymbol{x}}_{t|t}$$

$$\hat{\boldsymbol{P}}_{t+1|t} = \boldsymbol{G}\hat{\boldsymbol{P}}_{t|t}\boldsymbol{G}' + \boldsymbol{W}$$

Kalman filter

Notation:

$$\hat{\boldsymbol{x}}_{t|t} = \mathrm{E}[\boldsymbol{x}_t|y_1,\dots,y_t] \qquad \hat{\boldsymbol{P}}_{t|t} = \mathrm{Var}[\boldsymbol{x}_t|y_1,\dots,y_t]$$

$$\hat{\boldsymbol{x}}_{t|t-1} = \mathrm{E}[\boldsymbol{x}_t|y_1,\dots,y_{t-1}] \qquad \hat{\boldsymbol{P}}_{t|t-1} = \mathrm{Var}[\boldsymbol{x}_t|y_1,\dots,y_{t-1}]$$

$$\hat{y}_{t|t-1} = \mathrm{E}[y_t|y_1,\dots,y_{t-1}] \qquad \hat{v}_{t|t-1} = \mathrm{Var}[y_t|y_1,\dots,y_{t-1}]$$

Forecasting:

$$\hat{y}_{t|t-1} = \boldsymbol{f}'\hat{\boldsymbol{x}}_{t|t-1}$$

$$\hat{v}_{t|t-1} = \boldsymbol{f}'\hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f} + \sigma^2$$

Updating or State Filtering:

$$\hat{\boldsymbol{x}}_{t|t} = \hat{\boldsymbol{x}}_{t|t-1} + \hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f}\hat{v}_{t|t-1}^{-1}(y_t - \hat{y}_{t|t-1})$$

$$\hat{\boldsymbol{P}}_{t|t} = \hat{\boldsymbol{P}}_{t|t-1} - \hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f}\hat{v}_{t|t-1}^{-1}\boldsymbol{f}'\hat{\boldsymbol{P}}_{t|t-1}$$

State Prediction

$$\hat{\boldsymbol{x}}_{t+1|t} = \boldsymbol{G}\hat{\boldsymbol{x}}_{t|t}$$

$$\hat{\boldsymbol{P}}_{t+1|t} = \boldsymbol{G}\hat{\boldsymbol{P}}_{t|t}\boldsymbol{G}' + \boldsymbol{W}$$

Kalman filter

Notation:

$$\hat{\boldsymbol{x}}_{t|t} = \mathrm{E}[\boldsymbol{x}_t|y_1,\dots,y_t] \qquad \hat{\boldsymbol{P}}_{t|t} = \mathrm{Var}[\boldsymbol{x}_t|y_1,\dots,y_t]$$

$$\hat{\boldsymbol{x}}_{t|t-1} = \mathrm{E}[\boldsymbol{x}_t|y_1,\dots,y_{t-1}] \qquad \hat{\boldsymbol{P}}_{t|t-1} = \mathrm{Var}[\boldsymbol{x}_t|y_1,\dots,y_{t-1}]$$

$$\hat{y}_{t|t-1} = \mathrm{E}[y_t|y_1,\dots,y_{t-1}] \qquad \hat{v}_{t|t-1} = \mathrm{Var}[y_t|y_1,\dots,y_{t-1}]$$

Forecasting:

$$\hat{y}_{t|t-1} = \boldsymbol{f}'\hat{\boldsymbol{x}}_{t|t-1}$$

$$\hat{v}_{t|t-1} = \boldsymbol{f}'\hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f} + \sigma^2$$

Updating or State Filtering:

$$\hat{\boldsymbol{x}}_{t|t} = \hat{\boldsymbol{x}}_{t|t-1} + \hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f}\hat{v}_{t|t-1}^{-1}(y_t - \hat{y}_{t|t-1})$$

$$\hat{\boldsymbol{P}}_{t|t} = \hat{\boldsymbol{P}}_{t|t-1} - \hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f}\hat{v}_{t|t-1}^{-1}\boldsymbol{f}'\hat{\boldsymbol{P}}_{t|t-1}$$

State Prediction

$$\hat{\boldsymbol{x}}_{t+1|t} = \boldsymbol{G}\hat{\boldsymbol{x}}_{t|t}$$

$$\hat{\boldsymbol{P}}_{t+1|t} = \boldsymbol{G}\hat{\boldsymbol{P}}_{t|t}\boldsymbol{G}' + \boldsymbol{W}$$

Kalman filter

Notation:

$$\hat{\boldsymbol{x}}_{t|t} = \text{E}[\boldsymbol{x}_t|y_1,\dots,y_t] \qquad \hat{\boldsymbol{P}}_{t|t} = \text{Var}[\boldsymbol{x}_t|y_1,\dots,y_t]$$

$$\hat{\boldsymbol{x}}_{t|t-1} = \text{E}[\boldsymbol{x}_t|y_1,\dots,y_{t-1}] \qquad \hat{\boldsymbol{P}}_{t|t-1} = \text{Var}[\boldsymbol{x}_t|y_1,\dots,y_{t-1}]$$

$$\hat{y}_{t|t-1} = \text{E}[y_t|y_1,\dots,y_{t-1}] \qquad \hat{v}_{t|t-1} = \text{Var}[y_t|y_1,\dots,y_{t-1}]$$

Forecasting:

Iterate for $t = 1,\dots,T$

$$\hat{y}_{t|t-1} = \boldsymbol{f}'\hat{\boldsymbol{x}}_{t|t-1}$$

$$\hat{v}_{t|t-1} = \boldsymbol{f}'\hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f} + \sigma^2$$

Updating or State Filtering:

$$\hat{\boldsymbol{x}}_{t|t} = \hat{\boldsymbol{x}}_{t|t-1} + \hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f}\hat{v}_{t|t-1}^{-1}(y_t - \hat{y}_{t|t-1})$$

$$\hat{\boldsymbol{P}}_{t|t} = \hat{\boldsymbol{P}}_{t|t-1} - \hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f}\hat{v}_{t|t-1}^{-1}\boldsymbol{f}'\hat{\boldsymbol{P}}_{t|t-1}$$

State Prediction

$$\hat{\boldsymbol{x}}_{t+1|t} = \boldsymbol{G}\hat{\boldsymbol{x}}_{t|t}$$

$$\hat{\boldsymbol{P}}_{t+1|t} = \boldsymbol{G}\hat{\boldsymbol{P}}_{t|t}\boldsymbol{G}' + \boldsymbol{W}$$

Kalman filter

Notation:

$$\hat{\boldsymbol{x}}_{t|t} = \mathrm{E}[\boldsymbol{x}_t|y_1,\dots,y_t] \qquad \hat{\boldsymbol{P}}_{t|t} = \mathrm{Var}[\boldsymbol{x}_t|y_1,\dots,y_t]$$

$$\hat{\boldsymbol{x}}_{t|t-1} = \mathrm{E}[\boldsymbol{x}_t|y_1,\dots,y_{t-1}] \qquad \hat{\boldsymbol{P}}_{t|t-1} = \mathrm{Var}[\boldsymbol{x}_t|y_1,\dots,y_{t-1}]$$

$$\hat{y}_{t|t-1} = \mathrm{E}[y_t|y_1,\dots,y_{t-1}] \qquad \hat{v}_{t|t-1} = \mathrm{Var}[y_t|y_1,\dots,y_{t-1}]$$

Forecasting:

$$\hat{y}_{t|t-1} = \boldsymbol{f}'\hat{\boldsymbol{x}}_{t|t-1}$$

$$\hat{v}_{t|t-1} = \boldsymbol{f}'\hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f} + \sigma^2$$

Iterate for $t = 1,\dots,T$

Assume we know $\boldsymbol{x}_{1|0}$ and $\boldsymbol{P}_{1|0}$.

Updating or State Filtering:

$$\hat{\boldsymbol{x}}_{t|t} = \hat{\boldsymbol{x}}_{t|t-1} + \hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f}\hat{v}_{t|t-1}^{-1}(y_t - \hat{y}_{t|t-1})$$

$$\hat{\boldsymbol{P}}_{t|t} = \hat{\boldsymbol{P}}_{t|t-1} - \hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f}\hat{v}_{t|t-1}^{-1}\boldsymbol{f}'\hat{\boldsymbol{P}}_{t|t-1}$$

State Prediction

$$\hat{\boldsymbol{x}}_{t+1|t} = \boldsymbol{G}\hat{\boldsymbol{x}}_{t|t}$$

$$\hat{\boldsymbol{P}}_{t+1|t} = \boldsymbol{G}\hat{\boldsymbol{P}}_{t|t}\boldsymbol{G}' + \boldsymbol{W}$$

Kalman filter

Notation:

$$\hat{\boldsymbol{x}}_{t|t} = \mathrm{E}[\boldsymbol{x}_t | y_1, \dots, y_t] \qquad \hat{\boldsymbol{P}}_{t|t} = \mathrm{Var}[\boldsymbol{x}_t | y_1, \dots, y_t]$$

$$\hat{\boldsymbol{x}}_{t|t-1} = \mathrm{E}[\boldsymbol{x}_t | y_1, \dots, y_{t-1}] \qquad \hat{\boldsymbol{P}}_{t|t-1} = \mathrm{Var}[\boldsymbol{x}_t | y_1, \dots, y_{t-1}]$$

$$\hat{y}_{t|t-1} = \mathrm{E}[y_t | y_1, \dots, y_{t-1}] \qquad \hat{v}_{t|t-1} = \mathrm{Var}[y_t | y_1, \dots, y_{t-1}]$$

Forecasting:

$$\hat{y}_{t|t-1} = \boldsymbol{f}'\hat{\boldsymbol{x}}_{t|t-1}$$

$$\hat{v}_{t|t-1} = \boldsymbol{f}'\hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f} + \sigma^2$$

Iterate for $t = 1, \dots, T$

Assume we know $\boldsymbol{x}_{1|0}$ and $\boldsymbol{P}_{1|0}$.

Updating or State Filtering:

$$\hat{\boldsymbol{x}}_{t|t} = \hat{\boldsymbol{x}}_{t|t-1} + \hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f}\hat{v}_{t|t-1}^{-1}(y_t - \hat{y}_{t|t-1})$$

$$\hat{\boldsymbol{P}}_{t|t} = \hat{\boldsymbol{P}}_{t|t-1} - \hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f}\hat{v}_{t|t-1}^{-1}\boldsymbol{f}'\hat{\boldsymbol{P}}_{t|t-1}$$

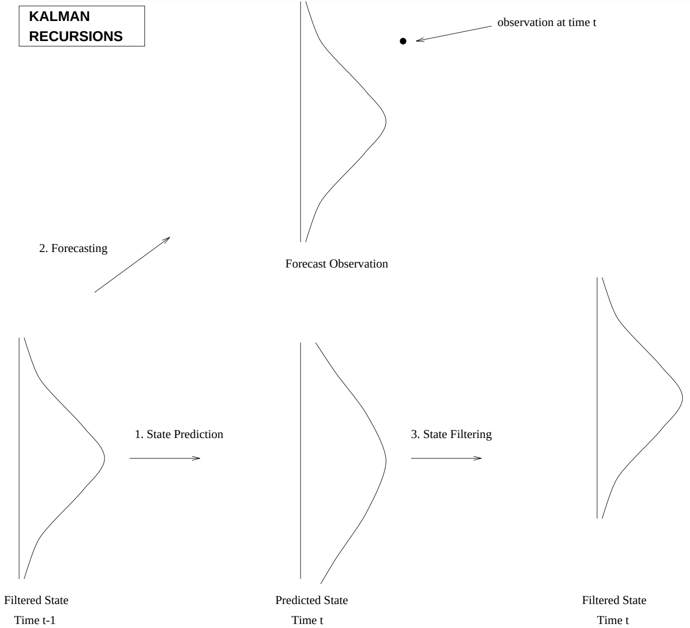
State Prediction

$$\hat{\boldsymbol{x}}_{t+1|t} = \boldsymbol{G}\hat{\boldsymbol{x}}_{t|t}$$

$$\hat{\boldsymbol{P}}_{t+1|t} = \boldsymbol{G}\hat{\boldsymbol{P}}_{t|t}\boldsymbol{G}' + \boldsymbol{W}$$

Just conditional expectations. So this gives minimum MSE estimates.

Kalman recursions



Initializing Kalman filter

- Need $\boldsymbol{x}_{1|0}$ and $\boldsymbol{P}_{1|0}$ to get started.
- Common approach for structural models: set $\boldsymbol{x}_{1|0} = 0$ and $\boldsymbol{P}_{1|0} = k\boldsymbol{I}$ for a very large k.
- Lots of research papers on optimal initialization choices for Kalman recursions.
- ETS approach was to estimate $\boldsymbol{x}_{1|0}$ and avoid $\boldsymbol{P}_{1|0}$ by assuming error processes identical.
- A random $\boldsymbol{x}_{1|0}$ could be used with ETS models, and then a form of Kalman filter would be required for estimation and forecasting.
- This gives more realistic prediction intervals.

Initializing Kalman filter

- Need $\boldsymbol{x}_{1|0}$ and $\boldsymbol{P}_{1|0}$ to get started.
- Common approach for structural models: set $\boldsymbol{x}_{1|0} = 0$ and $\boldsymbol{P}_{1|0} = k\boldsymbol{I}$ for a very large k.
- Lots of research papers on optimal initialization choices for Kalman recursions.
- ETS approach was to estimate $\boldsymbol{x}_{1|0}$ and avoid $\boldsymbol{P}_{1|0}$ by assuming error processes identical.
- A random $\boldsymbol{x}_{1|0}$ could be used with ETS models, and then a form of Kalman filter would be required for estimation and forecasting.
- This gives more realistic prediction intervals.

Initializing Kalman filter

- Need $\boldsymbol{x}_{1|0}$ and $\boldsymbol{P}_{1|0}$ to get started.
- Common approach for structural models: set $\boldsymbol{x}_{1|0} = 0$ and $\boldsymbol{P}_{1|0} = k\boldsymbol{I}$ for a very large k.
- Lots of research papers on optimal initialization choices for Kalman recursions.
- ETS approach was to estimate $\boldsymbol{x}_{1|0}$ and avoid $\boldsymbol{P}_{1|0}$ by assuming error processes identical.
- A random $\boldsymbol{x}_{1|0}$ could be used with ETS models, and then a form of Kalman filter would be required for estimation and forecasting.
- This gives more realistic prediction intervals.

Initializing Kalman filter

- Need $\boldsymbol{x}_{1|0}$ and $\boldsymbol{P}_{1|0}$ to get started.
- Common approach for structural models: set $\boldsymbol{x}_{1|0} = 0$ and $\boldsymbol{P}_{1|0} = k\boldsymbol{I}$ for a very large k.
- Lots of research papers on optimal initialization choices for Kalman recursions.
- ETS approach was to estimate $\boldsymbol{x}_{1|0}$ and avoid $\boldsymbol{P}_{1|0}$ by assuming error processes identical.
- A random $\boldsymbol{x}_{1|0}$ could be used with ETS models, and then a form of Kalman filter would be required for estimation and forecasting.
- This gives more realistic prediction intervals.

Initializing Kalman filter

- Need $\boldsymbol{x}_{1|0}$ and $\boldsymbol{P}_{1|0}$ to get started.
- Common approach for structural models: set $\boldsymbol{x}_{1|0} = 0$ and $\boldsymbol{P}_{1|0} = k\boldsymbol{I}$ for a very large k.
- Lots of research papers on optimal initialization choices for Kalman recursions.
- ETS approach was to estimate $\boldsymbol{x}_{1|0}$ and avoid $\boldsymbol{P}_{1|0}$ by assuming error processes identical.
- A random $\boldsymbol{x}_{1|0}$ could be used with ETS models, and then a form of Kalman filter would be required for estimation and forecasting.
- This gives more realistic prediction intervals.

Initializing Kalman filter

- Need $\boldsymbol{x}_{1|0}$ and $\boldsymbol{P}_{1|0}$ to get started.
- Common approach for structural models: set $\boldsymbol{x}_{1|0} = 0$ and $\boldsymbol{P}_{1|0} = k\boldsymbol{I}$ for a very large k.
- Lots of research papers on optimal initialization choices for Kalman recursions.
- ETS approach was to estimate $\boldsymbol{x}_{1|0}$ and avoid $\boldsymbol{P}_{1|0}$ by assuming error processes identical.
- A random $\boldsymbol{x}_{1|0}$ could be used with ETS models, and then a form of Kalman filter would be required for estimation and forecasting.
- This gives more realistic prediction intervals.

Local level model

$$y_t = \ell_t + \varepsilon_t \qquad \varepsilon_t \sim \text{NID}(0, \sigma^2)$$
$$\ell_t = \ell_{t-1} + u_t \qquad u_t \sim \text{NID}(0, q^2)$$

Kalman recursions:

$$\hat{y}_{t|t-1} = \hat{\ell}_{t-1|t-1}$$
$$\hat{v}_{t|t-1} = \hat{p}_{t|t-1} + \sigma^2$$
$$\hat{\ell}_{t|t} = \hat{\ell}_{t-1|t-1} + \hat{p}_{t|t-1}\hat{v}_{t|t-1}^{-1}(y_t - \hat{y}_{t|t-1})$$
$$\hat{p}_{t+1|t} = \hat{p}_{t|t-1}(1 - \hat{v}_{t|t-1}^{-1}\hat{p}_{t|t-1}) + q^2$$

Local level model

$$y_t = \ell_t + \varepsilon_t \qquad \varepsilon_t \sim \text{NID}(0, \sigma^2)$$
$$\ell_t = \ell_{t-1} + u_t \qquad u_t \sim \text{NID}(0, q^2)$$

Kalman recursions:

$$\hat{y}_{t|t-1} = \hat{\ell}_{t-1|t-1}$$
$$\hat{v}_{t|t-1} = \hat{p}_{t|t-1} + \sigma^2$$
$$\hat{\ell}_{t|t} = \hat{\ell}_{t-1|t-1} + \hat{p}_{t|t-1}\hat{v}_{t|t-1}^{-1}(y_t - \hat{y}_{t|t-1})$$
$$\hat{p}_{t+1|t} = \hat{p}_{t|t-1}(1 - \hat{v}_{t|t-1}^{-1}\hat{p}_{t|t-1}) + q^2$$

Handling missing values

Forecasting:

$$\hat{y}_{t|t-1} = \boldsymbol{f}'\hat{\boldsymbol{x}}_{t|t-1}$$
$$\hat{v}_{t|t-1} = \boldsymbol{f}'\hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f} + \sigma^2$$

Iterate for $t = 1, \dots, T$ starting with $\boldsymbol{x}_{1|0}$ and $\boldsymbol{P}_{1|0}$.

Updating or State Filtering:

$$\hat{\boldsymbol{x}}_{t|t} = \hat{\boldsymbol{x}}_{t|t-1} + \hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f}\hat{v}^{-1}_{t|t-1}(y_t - \hat{y}_{t|t-1})$$
$$\hat{\boldsymbol{P}}_{t|t} = \hat{\boldsymbol{P}}_{t|t-1} - \hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f}\hat{v}^{-1}_{t|t-1}\boldsymbol{f}'\hat{\boldsymbol{P}}_{t|t-1}$$

State Prediction

$$\hat{\boldsymbol{x}}_{t|t-1} = \boldsymbol{G}\hat{\boldsymbol{x}}_{t-1|t-1}$$
$$\hat{\boldsymbol{P}}_{t|t-1} = \boldsymbol{G}\hat{\boldsymbol{P}}_{t-1|t-1}\boldsymbol{G}' + \boldsymbol{W}$$

Handling missing values

Forecasting:

$$\hat{y}_{t|t-1} = \boldsymbol{f}'\hat{\boldsymbol{x}}_{t|t-1}$$
$$\hat{v}_{t|t-1} = \boldsymbol{f}'\hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f} + \sigma^2$$

Iterate for $t = 1, \dots, T$ starting with $\boldsymbol{x}_{1|0}$ and $\boldsymbol{P}_{1|0}$.

Updating or State Filtering:

$$\hat{\boldsymbol{x}}_{t|t} = \hat{\boldsymbol{x}}_{t|t-1} + \hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f}\hat{v}^{-1}_{t|t-1}(y_t - \hat{y}_{t|t-1})$$
$$\hat{\boldsymbol{P}}_{t|t} = \hat{\boldsymbol{P}}_{t|t-1} - \hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f}\hat{v}^{-1}_{t|t-1}\boldsymbol{f}'\hat{\boldsymbol{P}}_{t|t-1}$$

State Prediction

$$\hat{\boldsymbol{x}}_{t|t-1} = \boldsymbol{G}\hat{\boldsymbol{x}}_{t-1|t-1}$$
$$\hat{\boldsymbol{P}}_{t|t-1} = \boldsymbol{G}\hat{\boldsymbol{P}}_{t-1|t-1}\boldsymbol{G}' + \boldsymbol{W}$$

Ignored greyed out section if y_t missing.

Handling missing values

Forecasting:

$$\hat{y}_{t|t-1} = \boldsymbol{f}'\hat{\boldsymbol{x}}_{t|t-1}$$

$$\hat{v}_{t|t-1} = \boldsymbol{f}'\hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f} + \sigma^2$$

Iterate for $t = 1, \dots, T$ starting with $\boldsymbol{x}_{1|0}$ and $\boldsymbol{P}_{1|0}$.

Updating or State Filtering:

$$\hat{\boldsymbol{x}}_{t|t} = \hat{\boldsymbol{x}}_{t|t-1} + \hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f}\hat{v}_{t|t-1}^{-1}(y_t - \hat{y}_{t|t-1})$$

$$\hat{\boldsymbol{P}}_{t|t} = \hat{\boldsymbol{P}}_{t|t-1} - \hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f}\hat{v}_{t|t-1}^{-1}\boldsymbol{f}'\hat{\boldsymbol{P}}_{t|t-1}$$

State Prediction

$$\hat{\boldsymbol{x}}_{t|t-1} = \boldsymbol{G}\hat{\boldsymbol{x}}_{t-1|t-1}$$

$$\hat{\boldsymbol{P}}_{t|t-1} = \boldsymbol{G}\hat{\boldsymbol{P}}_{t-1|t-1}\boldsymbol{G}' + \boldsymbol{W}$$

Ignored greyed out section if y_t missing.

Multi-step forecasting

Forecasting:

$$\hat{y}_{t|t-1} = \boldsymbol{f}'\hat{\boldsymbol{x}}_{t|t-1}$$
$$\hat{v}_{t|t-1} = \boldsymbol{f}'\hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f} + \sigma^2$$

Iterate for $t = T+1, \dots, T+h$ starting with $\boldsymbol{x}_{T|T}$ and $\boldsymbol{P}_{T|T}$.

Updating or State Filtering:

$$\hat{\boldsymbol{x}}_{t|t} = \hat{\boldsymbol{x}}_{t|t-1} \sout{+\hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f}\hat{v}_{t|t-1}^{-1}(y_t - \hat{y}_{t|t-1})}$$
$$\hat{\boldsymbol{P}}_{t|t} = \hat{\boldsymbol{P}}_{t|t-1} \sout{-\hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f}\hat{v}_{t|t-1}^{-1}\boldsymbol{f}'\hat{\boldsymbol{P}}_{t|t-1}}$$

State Prediction

$$\hat{\boldsymbol{x}}_{t|t-1} = \boldsymbol{G}\hat{\boldsymbol{x}}_{t-1|t-1}$$
$$\hat{\boldsymbol{P}}_{t|t-1} = \boldsymbol{G}\hat{\boldsymbol{P}}_{t-1|t-1}\boldsymbol{G}' + \boldsymbol{W}$$

Multi-step forecasting

Forecasting:

$$\hat{y}_{t|t-1} = \boldsymbol{f}'\hat{\boldsymbol{x}}_{t|t-1}$$

$$\hat{v}_{t|t-1} = \boldsymbol{f}'\hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f} + \sigma^2$$

Iterate for $t = T+1, \dots, T+h$ starting with $\boldsymbol{x}_{T|T}$ and $\boldsymbol{P}_{T|T}$.

Updating or State Filtering:

$$\hat{\boldsymbol{x}}_{t|t} = \hat{\boldsymbol{x}}_{t|t-1} \cancel{+ \hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f}\hat{v}_{t|t-1}^{-1}(y_t - \hat{y}_{t|t-1})}$$

$$\hat{\boldsymbol{P}}_{t|t} = \hat{\boldsymbol{P}}_{t|t-1} \cancel{- \hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f}\hat{v}_{t|t-1}^{-1}\boldsymbol{f}'\hat{\boldsymbol{P}}_{t|t-1}}$$

State Prediction

$$\hat{\boldsymbol{x}}_{t|t-1} = \boldsymbol{G}\hat{\boldsymbol{x}}_{t-1|t-1}$$

$$\hat{\boldsymbol{P}}_{t|t-1} = \boldsymbol{G}\hat{\boldsymbol{P}}_{t-1|t-1}\boldsymbol{G}' + \boldsymbol{W}$$

Treat future values as missing.

Kalman filter

What's so special about the Kalman filter

- Very general equations for any model in state space format.
- Any model in state space format can easily be generalized.
- Optimal MSE forecasts
- Easy to handle missing values.
- Easy to compute likelihood.

Kalman filter

What's so special about the Kalman filter

- Very general equations for any model in state space format.
- Any model in state space format can easily be generalized.
- Optimal MSE forecasts
- Easy to handle missing values.
- Easy to compute likelihood.

Kalman filter

What's so special about the Kalman filter

- Very general equations for any model in state space format.
- Any model in state space format can easily be generalized.
- Optimal MSE forecasts
- Easy to handle missing values.
- Easy to compute likelihood.

Kalman filter

What's so special about the Kalman filter

- Very general equations for any model in state space format.
- Any model in state space format can easily be generalized.
- Optimal MSE forecasts
- Easy to handle missing values.
- Easy to compute likelihood.

Kalman filter

What's so special about the Kalman filter

- Very general equations for any model in state space format.
- Any model in state space format can easily be generalized.
- Optimal MSE forecasts
- Easy to handle missing values.
- Easy to compute likelihood.

Likelihood calculation

θ = all unknown parameters

$f_\theta(y_t|y_1, y_2, \dots, y_{t-1})$ = one-step forecast density.

Likelihood

$$L(y_1, \dots, y_T; \theta) = \prod_{t=1}^{T} f_\theta(y_t|y_1, \dots, y_{t-1})$$

Gaussian log likelihood

$$\log L = -\frac{T}{2}\log(2\pi) - \frac{1}{2}\sum_{t=1}^{T}\log \hat{v}_{t|t-1} - \frac{1}{2}\sum_{t=1}^{T} e_t^2/\hat{v}_{t|t-1}$$

where $e_t = y_t - \hat{y}_{t|t-1}$.

All terms obtained from Kalman filter equations.

Likelihood calculation

θ = all unknown parameters

$f_\theta(y_t|y_1, y_2, \dots, y_{t-1})$ = one-step forecast density.

Likelihood

$$L(y_1, \dots, y_T; \theta) = \prod_{t=1}^{T} f_\theta(y_t|y_1, \dots, y_{t-1})$$

Gaussian log likelihood

$$\log L = -\frac{T}{2}\log(2\pi) - \frac{1}{2}\sum_{t=1}^{T}\log \hat{v}_{t|t-1} - \frac{1}{2}\sum_{t=1}^{T} e_t^2/\hat{v}_{t|t-1}$$

where $e_t = y_t - \hat{y}_{t|t-1}$.

All terms obtained from Kalman filter equations.

Likelihood calculation

θ = all unknown parameters

$f_\theta(y_t|y_1, y_2, \ldots, y_{t-1})$ = one-step forecast density.

Likelihood

$$L(y_1, \ldots, y_T; \theta) = \prod_{t=1}^{T} f_\theta(y_t|y_1, \ldots, y_{t-1})$$

Gaussian log likelihood

$$\log L = -\frac{T}{2}\log(2\pi) - \frac{1}{2}\sum_{t=1}^{T}\log \hat{v}_{t|t-1} - \frac{1}{2}\sum_{t=1}^{T} e_t^2/\hat{v}_{t|t-1}$$

where $e_t = y_t - \hat{y}_{t|t-1}$.

All terms obtained from Kalman filter equations.

Outline

ARMA models in state space form

AR(2) model

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + e_t, \qquad e_t \sim \text{NID}(0, \sigma^2)$$

Let $\boldsymbol{x}_t = \begin{bmatrix} y_t \\ y_{t-1} \end{bmatrix}$ and $\boldsymbol{w}_t = \begin{bmatrix} e_t \\ 0 \end{bmatrix}$.

Then

$$y_t = [1 \;\; 0]\boldsymbol{x}_t$$

$$\boldsymbol{x}_t = \begin{bmatrix} \phi_1 & \phi_2 \\ 1 & 0 \end{bmatrix} \boldsymbol{x}_{t-1} + \boldsymbol{w}_t$$

- Now in state space form

ARMA models in state space form

AR(2) model

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + e_t, \qquad e_t \sim \text{NID}(0, \sigma^2)$$

Let $\boldsymbol{x}_t = \begin{bmatrix} y_t \\ y_{t-1} \end{bmatrix}$ and $\boldsymbol{w}_t = \begin{bmatrix} e_t \\ 0 \end{bmatrix}$.

Then

$$y_t = [1 \;\; 0]\boldsymbol{x}_t$$

$$\boldsymbol{x}_t = \begin{bmatrix} \phi_1 & \phi_2 \\ 1 & 0 \end{bmatrix} \boldsymbol{x}_{t-1} + \boldsymbol{w}_t$$

- Now in state space form
- We can use Kalman filter to compute likelihood and forecasts.

ARMA models in state space form

AR(2) model

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + e_t, \qquad e_t \sim \text{NID}(0, \sigma^2)$$

Let $\boldsymbol{x}_t = \begin{bmatrix} y_t \\ y_{t-1} \end{bmatrix}$ and $\boldsymbol{w}_t = \begin{bmatrix} e_t \\ 0 \end{bmatrix}$.

Then

$$y_t = [1 \;\; 0]\boldsymbol{x}_t$$

$$\boldsymbol{x}_t = \begin{bmatrix} \phi_1 & \phi_2 \\ 1 & 0 \end{bmatrix} \boldsymbol{x}_{t-1} + \boldsymbol{w}_t$$

- Now in state space form
- We can use Kalman filter to compute likelihood and forecasts.

ARMA models in state space form

AR(2) model

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + e_t, \qquad e_t \sim \text{NID}(0, \sigma^2)$$

Let $\boldsymbol{x}_t = \begin{bmatrix} y_t \\ y_{t-1} \end{bmatrix}$ and $\boldsymbol{w}_t = \begin{bmatrix} e_t \\ 0 \end{bmatrix}$.

Then

$$y_t = [1 \;\; 0]\boldsymbol{x}_t$$
$$\boldsymbol{x}_t = \begin{bmatrix} \phi_1 & \phi_2 \\ 1 & 0 \end{bmatrix} \boldsymbol{x}_{t-1} + \boldsymbol{w}_t$$

- Now in state space form
- We can use Kalman filter to compute likelihood and forecasts.

ARMA models in state space form

AR(2) model

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + e_t, \qquad e_t \sim \text{NID}(0, \sigma^2)$$

Let $\boldsymbol{x}_t = \begin{bmatrix} y_t \\ y_{t-1} \end{bmatrix}$ and $\boldsymbol{w}_t = \begin{bmatrix} e_t \\ 0 \end{bmatrix}$.

Then

$$y_t = [1 \ \ 0]\boldsymbol{x}_t$$
$$\boldsymbol{x}_t = \begin{bmatrix} \phi_1 & \phi_2 \\ 1 & 0 \end{bmatrix} \boldsymbol{x}_{t-1} + \boldsymbol{w}_t$$

- Now in state space form
- We can use Kalman filter to compute likelihood and forecasts.

ARMA models in state space form

AR(2) model

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + e_t, \qquad e_t \sim \text{NID}(0, \sigma^2)$$

Alternative formulation

Let $\boldsymbol{x}_t = \begin{bmatrix} y_t \\ \phi_2 y_{t-1} \end{bmatrix}$ and $\boldsymbol{w}_t = \begin{bmatrix} e_t \\ 0 \end{bmatrix}$.

$$y_t = \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{x}_t$$

$$\boldsymbol{x}_t = \begin{bmatrix} \phi_1 & 1 \\ \phi_2 & 0 \end{bmatrix} \boldsymbol{x}_{t-1} + \boldsymbol{w}_t$$

- Alternative state space form

ARMA models in state space form

AR(2) model

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + e_t, \qquad e_t \sim \text{NID}(0, \sigma^2)$$

Alternative formulation

Let $\boldsymbol{x}_t = \begin{bmatrix} y_t \\ \phi_2 y_{t-1} \end{bmatrix}$ and $\boldsymbol{w}_t = \begin{bmatrix} e_t \\ 0 \end{bmatrix}$.

$$y_t = \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{x}_t$$

$$\boldsymbol{x}_t = \begin{bmatrix} \phi_1 & 1 \\ \phi_2 & 0 \end{bmatrix} \boldsymbol{x}_{t-1} + \boldsymbol{w}_t$$

- Alternative state space form
- We can use Kalman filter to compute likelihood and forecasts.

ARMA models in state space form

AR(2) model

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + e_t, \qquad e_t \sim \text{NID}(0, \sigma^2)$$

Alternative formulation

Let $\boldsymbol{x}_t = \begin{bmatrix} y_t \\ \phi_2 y_{t-1} \end{bmatrix}$ and $\boldsymbol{w}_t = \begin{bmatrix} e_t \\ 0 \end{bmatrix}$.

$$y_t = \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{x}_t$$

$$\boldsymbol{x}_t = \begin{bmatrix} \phi_1 & 1 \\ \phi_2 & 0 \end{bmatrix} \boldsymbol{x}_{t-1} + \boldsymbol{w}_t$$

- Alternative state space form
- We can use Kalman filter to compute likelihood and forecasts.

ARMA models in state space form

AR(2) model

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + e_t, \qquad e_t \sim \text{NID}(0, \sigma^2)$$

Alternative formulation

Let $\boldsymbol{x}_t = \begin{bmatrix} y_t \\ \phi_2 y_{t-1} \end{bmatrix}$ and $\boldsymbol{w}_t = \begin{bmatrix} e_t \\ 0 \end{bmatrix}$.

$$y_t = \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{x}_t$$
$$\boldsymbol{x}_t = \begin{bmatrix} \phi_1 & 1 \\ \phi_2 & 0 \end{bmatrix} \boldsymbol{x}_{t-1} + \boldsymbol{w}_t$$

- Alternative state space form
- We can use Kalman filter to compute likelihood and forecasts.

ARMA models in state space form

AR(2) model

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + e_t, \qquad e_t \sim \text{NID}(0, \sigma^2)$$

Alternative formulation

Let $\boldsymbol{x}_t = \begin{bmatrix} y_t \\ \phi_2 y_{t-1} \end{bmatrix}$ and $\boldsymbol{w}_t = \begin{bmatrix} e_t \\ 0 \end{bmatrix}$.

$$y_t = \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{x}_t$$

$$\boldsymbol{x}_t = \begin{bmatrix} \phi_1 & 1 \\ \phi_2 & 0 \end{bmatrix} \boldsymbol{x}_{t-1} + \boldsymbol{w}_t$$

- Alternative state space form
- We can use Kalman filter to compute likelihood and forecasts.

ARMA models in state space form

AR(p) model

$$y_t = \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + e_t, \qquad e_t \sim \text{NID}(0, \sigma^2)$$

Let $\boldsymbol{x}_t = \begin{bmatrix} y_t \\ y_{t-1} \\ \vdots \\ y_{t-p+1} \end{bmatrix}$ and $\boldsymbol{w}_t = \begin{bmatrix} e_t \\ 0 \\ \vdots \\ 0 \end{bmatrix}$.

$$y_t = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \end{bmatrix} \boldsymbol{x}_t$$

$$\boldsymbol{x}_t = \begin{bmatrix} \phi_1 & \phi_2 & \dots & \phi_{p-1} & \phi_p \\ 1 & 0 & \dots & 0 & 0 \\ \vdots & & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & 1 & 0 \end{bmatrix} \boldsymbol{x}_{t-1} + \boldsymbol{w}_t$$

ARMA models in state space form

AR(p) model

$$y_t = \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + e_t, \qquad e_t \sim \text{NID}(0, \sigma^2)$$

Let $\boldsymbol{x}_t = \begin{bmatrix} y_t \\ y_{t-1} \\ \vdots \\ y_{t-p+1} \end{bmatrix}$ and $\boldsymbol{w}_t = \begin{bmatrix} e_t \\ 0 \\ \vdots \\ 0 \end{bmatrix}$.

$$y_t = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \end{bmatrix} \boldsymbol{x}_t$$

$$\boldsymbol{x}_t = \begin{bmatrix} \phi_1 & \phi_2 & \dots & \phi_{p-1} & \phi_p \\ 1 & 0 & \dots & 0 & 0 \\ \vdots & & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & 1 & 0 \end{bmatrix} \boldsymbol{x}_{t-1} + \boldsymbol{w}_t$$

ARMA models in state space form

ARMA$(1,1)$ model

$$y_t = \phi y_{t-1} + \theta e_{t-1} + e_t, \qquad e_t \sim \text{NID}(0, \sigma^2)$$

Let $\boldsymbol{x}_t = \begin{bmatrix} y_t \\ \theta e_t \end{bmatrix}$ and $\boldsymbol{w}_t = \begin{bmatrix} e_t \\ \theta e_t \end{bmatrix}$.

$$y_t = \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{x}_t$$

$$\boldsymbol{x}_t = \begin{bmatrix} \phi & 1 \\ 0 & 0 \end{bmatrix} \boldsymbol{x}_{t-1} + \boldsymbol{w}_t$$

ARMA models in state space form

ARMA$(1,1)$ model

$$y_t = \phi y_{t-1} + \theta e_{t-1} + e_t, \qquad e_t \sim \text{NID}(0, \sigma^2)$$

Let $\boldsymbol{x}_t = \begin{bmatrix} y_t \\ \theta e_t \end{bmatrix}$ and $\boldsymbol{w}_t = \begin{bmatrix} e_t \\ \theta e_t \end{bmatrix}$.

$$y_t = \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{x}_t$$

$$\boldsymbol{x}_t = \begin{bmatrix} \phi & 1 \\ 0 & 0 \end{bmatrix} \boldsymbol{x}_{t-1} + \boldsymbol{w}_t$$

ARMA models in state space form

ARMA(p, q) model

$$y_t = \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \theta_1 e_{t-1} + \cdots + \theta_q e_{t-q} + e_t$$

Let $r = \max(p, q+1)$, $\quad \theta_i = 0, q < i \leq r, \quad \phi_j = 0, p < j \leq r$.

$$y_t = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix} \boldsymbol{x}_t$$

$$\boldsymbol{x}_t = \begin{bmatrix} \phi_1 & 1 & 0 & \dots & 0 \\ \phi_2 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \phi_{r-1} & 0 & \dots & 0 & 1 \\ \phi_r & 0 & 0 & \dots & 0 \end{bmatrix} \boldsymbol{x}_{t-1} + \begin{bmatrix} 1 \\ \theta_1 \\ \vdots \\ \theta_{r-1} \end{bmatrix} e_t$$

The `arima` function in R is implemented using this formulation.

ARMA models in state space form

ARMA(p, q) model

$$y_t = \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \theta_1 e_{t-1} + \cdots + \theta_q e_{t-q} + e_t$$

Let $r = \max(p, q+1)$, $\theta_i = 0, q < i \leq r$, $\phi_j = 0, p < j \leq r$.

$$y_t = \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix} \boldsymbol{x}_t$$

$$\boldsymbol{x}_t = \begin{bmatrix} \phi_1 & 1 & 0 & \ldots & 0 \\ \phi_2 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \phi_{r-1} & 0 & \ldots & 0 & 1 \\ \phi_r & 0 & 0 & \ldots & 0 \end{bmatrix} \boldsymbol{x}_{t-1} + \begin{bmatrix} 1 \\ \theta_1 \\ \vdots \\ \theta_{r-1} \end{bmatrix} e_t$$

The `arima` function in R is implemented using this formulation.

ARMA models in state space form

ARMA(p, q) model

$$y_t = \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \theta_1 e_{t-1} + \cdots + \theta_q e_{t-q} + e_t$$

Let $r = \max(p, q+1)$, $\quad \theta_i = 0, q < i \leq r, \quad \phi_j = 0, p < j \leq r$.

$$y_t = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix} \boldsymbol{x}_t$$

$$\boldsymbol{x}_t = \begin{bmatrix} \phi_1 & 1 & 0 & \dots & 0 \\ \phi_2 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \phi_{r-1} & 0 & \dots & 0 & 1 \\ \phi_r & 0 & 0 & \dots & 0 \end{bmatrix} \boldsymbol{x}_{t-1} + \begin{bmatrix} 1 \\ \theta_1 \\ \vdots \\ \theta_{r-1} \end{bmatrix} e_t$$

The `arima` function in R is implemented using this formulation.

ARMA models in state space form

ARMA(p,q) model

$$y_t = \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \theta_1 e_{t-1} + \cdots + \theta_q e_{t-q} + e_t$$

Let $r = \max(p, q+1)$, $\quad \theta_i = 0, q < i \le r, \quad \phi_j = 0, p < j \le r.$

$$y_t = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix} \boldsymbol{x}_t$$

$$\boldsymbol{x}_t = \begin{bmatrix} \phi_1 & 1 & 0 & \dots & 0 \\ \phi_2 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \phi_{r-1} & 0 & \dots & 0 & 1 \\ \phi_r & 0 & 0 & \dots & 0 \end{bmatrix} \boldsymbol{x}_{t-1} + \begin{bmatrix} 1 \\ \theta_1 \\ \vdots \\ \theta_{r-1} \end{bmatrix} e_t$$

The `arima` function in R is implemented using this formulation.

Outline

Kalman smoothing

Want estimate of $\boldsymbol{x}_t | y_1, \dots, y_T$ where $t < T$. That is, $\hat{\boldsymbol{x}}_{t|T}$.

$$\begin{aligned}
\hat{\boldsymbol{x}}_{t|T} &= \hat{\boldsymbol{x}}_{t|t} + \boldsymbol{A}_t \left(\hat{\boldsymbol{x}}_{t+1|T} - \hat{\boldsymbol{x}}_{t+1|t} \right) \\
\hat{P}_{t|T} &= \hat{P}_{t|t} + \boldsymbol{A}_t \left(\hat{P}_{t+1|T} - \hat{P}_{t+1|t} \right) \boldsymbol{A}_t' \\
\text{where} \quad A_t &= \hat{P}_{t|t} \boldsymbol{G}' \left(\hat{P}_{t+1|t} \right)^{-1}.
\end{aligned}$$

- Uses all data, not just previous data.
- Useful for estimating missing values: $\hat{y}_{t|T} = \boldsymbol{f}' \hat{\boldsymbol{x}}_{t|T}$.
- Useful for seasonal adjustment when one of the states is a seasonal component.

Kalman smoothing

Want estimate of $\boldsymbol{x}_t | y_1, \dots, y_T$ where $t < T$. That is, $\hat{\boldsymbol{x}}_{t|T}$.

$$\begin{aligned}
\hat{\boldsymbol{x}}_{t|T} &= \hat{\boldsymbol{x}}_{t|t} + \boldsymbol{A}_t \left(\hat{\boldsymbol{x}}_{t+1|T} - \hat{\boldsymbol{x}}_{t+1|t}\right) \\
\hat{P}_{t|T} &= \hat{P}_{t|t} + \boldsymbol{A}_t \left(\hat{P}_{t+1|T} - \hat{P}_{t+1|t}\right) \boldsymbol{A}_t' \\
\text{where} \quad A_t &= \hat{P}_{t|t} \boldsymbol{G}' \left(\hat{P}_{t+1|t}\right)^{-1}.
\end{aligned}$$

- Uses all data, not just previous data.
- Useful for estimating missing values: $\hat{y}_{t|T} = \boldsymbol{f}' \hat{\boldsymbol{x}}_{t|T}$.
- Useful for seasonal adjustment when one of the states is a seasonal component.

Kalman smoothing

Want estimate of $\boldsymbol{x}_t | y_1, \dots, y_T$ where $t < T$. That is, $\hat{\boldsymbol{x}}_{t|T}$.

$$\begin{aligned}
\hat{\boldsymbol{x}}_{t|T} &= \hat{\boldsymbol{x}}_{t|t} + \boldsymbol{A}_t \left(\hat{\boldsymbol{x}}_{t+1|T} - \hat{\boldsymbol{x}}_{t+1|t} \right) \\
\hat{P}_{t|T} &= \hat{P}_{t|t} + \boldsymbol{A}_t \left(\hat{P}_{t+1|T} - \hat{P}_{t+1|t} \right) \boldsymbol{A}'_t \\
\text{where} \quad A_t &= \hat{P}_{t|t} \boldsymbol{G}' \left(\hat{P}_{t+1|t} \right)^{-1}.
\end{aligned}$$

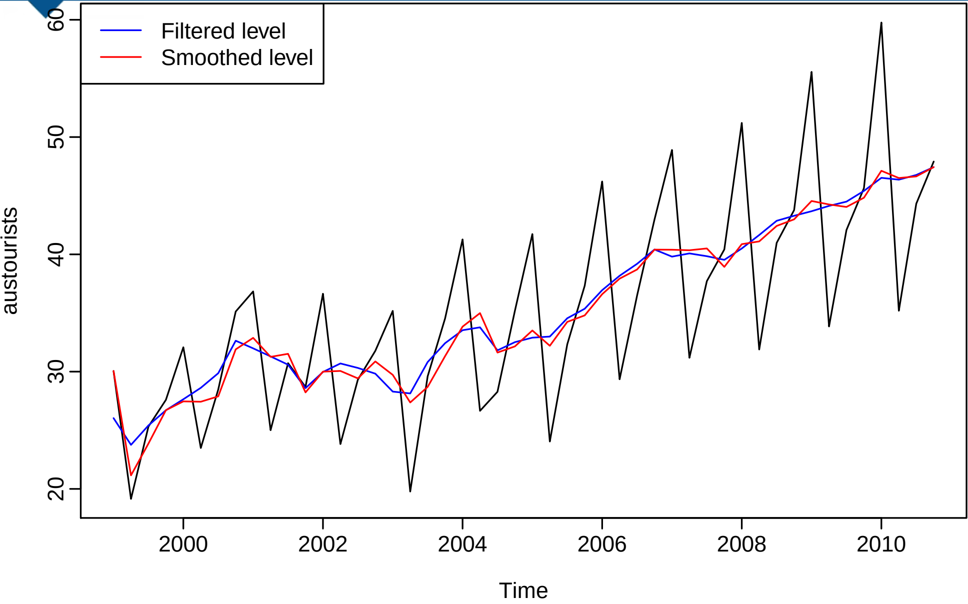
- Uses all data, not just previous data.
- Useful for estimating missing values: $\hat{y}_{t|T} = \boldsymbol{f}' \hat{\boldsymbol{x}}_{t|T}$.
- Useful for seasonal adjustment when one of the states is a seasonal component.

Kalman smoothing in R

```
fit <- StructTS(austourists, type = "BSM")
sm <- tsSmooth(fit)

plot(austourists)
lines(sm[,1],col='blue')
lines(fitted(fit)[,1],col='red')
legend("topleft",col=c('blue','red'),lty=1,
  legend=c("Filtered level","Smoothed level"))
```

Kalman smoothing in R



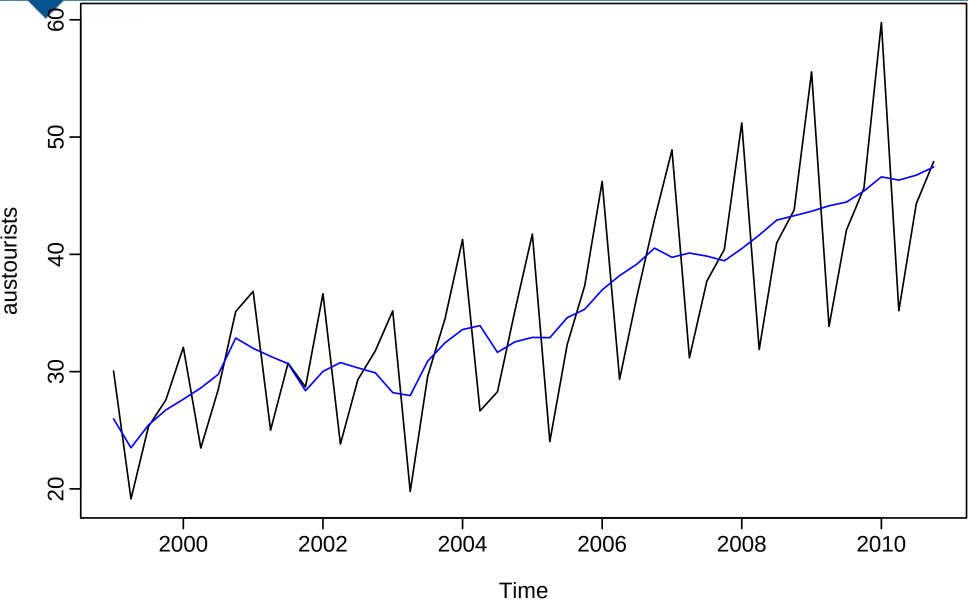
Kalman smoothing in R

```
fit <- StructTS(austourists, type = "BSM")
sm <- tsSmooth(fit)

plot(austourists)

# Seasonally adjusted data
aus.sa <- austourists - sm[,3]
lines(aus.sa,col='blue')
```

Kalman smoothing in R

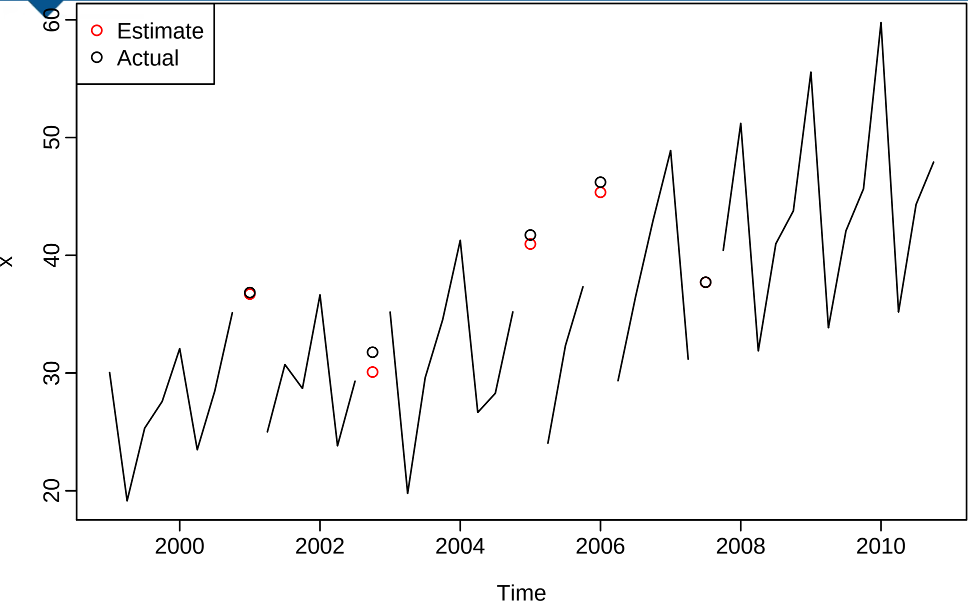


Kalman smoothing in R

```
x <- austourists
miss <- sample(1:length(x), 5)
x[miss] <- NA
fit <- StructTS(x, type = "BSM")
sm <- tsSmooth(fit)
estim <- sm[,1]+sm[,3]

plot(x, ylim=range(austourists))
points(time(x)[miss], estim[miss],
  col='red', pch=1)
points(time(x)[miss], austourists[miss],
  col='black', pch=1)
legend("topleft", pch=1, col=c(2,1),
  legend=c("Estimate","Actual"))
```

Kalman smoothing in R



Outline

Time varying parameter models

Linear Gaussian state space model

$$y_t = \boldsymbol{f}_t'\boldsymbol{x}_t + \varepsilon_t, \qquad \varepsilon_t \sim \text{N}(0, \sigma_t^2)$$

$$\boldsymbol{x}_t = \boldsymbol{G}_t\boldsymbol{x}_{t-1} + \boldsymbol{w}_t \qquad \boldsymbol{w}_t \sim \text{N}(\boldsymbol{0}, \boldsymbol{W}_t)$$

Kalman recursions:

$$\hat{y}_{t|t-1} = \boldsymbol{f}_t'\hat{\boldsymbol{x}}_{t|t-1}$$

$$\hat{v}_{t|t-1} = \boldsymbol{f}_t'\hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f}_t + \sigma_t^2$$

$$\hat{\boldsymbol{x}}_{t|t} = \hat{\boldsymbol{x}}_{t|t-1} + \hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f}_t\hat{v}_{t|t-1}^{-1}(y_t - \hat{y}_{t|t-1})$$

$$\hat{\boldsymbol{P}}_{t|t} = \hat{\boldsymbol{P}}_{t|t-1} - \hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f}_t\hat{v}_{t|t-1}^{-1}\boldsymbol{f}_t'\hat{\boldsymbol{P}}_{t|t-1}$$

$$\hat{\boldsymbol{x}}_{t|t-1} = \boldsymbol{G}_t\hat{\boldsymbol{x}}_{t-1|t-1}$$

$$\hat{\boldsymbol{P}}_{t|t-1} = \boldsymbol{G}_t\hat{\boldsymbol{P}}_{t-1|t-1}\boldsymbol{G}_t' + \boldsymbol{W}_t$$

Time varying parameter models

Linear Gaussian state space model

$$y_t = \boldsymbol{f}_t'\boldsymbol{x}_t + \varepsilon_t, \qquad \varepsilon_t \sim \mathrm{N}(0, \sigma_t^2)$$

$$\boldsymbol{x}_t = \boldsymbol{G}_t\boldsymbol{x}_{t-1} + \boldsymbol{w}_t \qquad \boldsymbol{w}_t \sim \mathrm{N}(\boldsymbol{0}, \boldsymbol{W}_t)$$

Kalman recursions:

$$\hat{y}_{t|t-1} = \boldsymbol{f}_t'\hat{\boldsymbol{x}}_{t|t-1}$$

$$\hat{v}_{t|t-1} = \boldsymbol{f}_t'\hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f}_t + \sigma_t^2$$

$$\hat{\boldsymbol{x}}_{t|t} = \hat{\boldsymbol{x}}_{t|t-1} + \hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f}_t\hat{v}_{t|t-1}^{-1}(y_t - \hat{y}_{t|t-1})$$

$$\hat{\boldsymbol{P}}_{t|t} = \hat{\boldsymbol{P}}_{t|t-1} - \hat{\boldsymbol{P}}_{t|t-1}\boldsymbol{f}_t\hat{v}_{t|t-1}^{-1}\boldsymbol{f}_t'\hat{\boldsymbol{P}}_{t|t-1}$$

$$\hat{\boldsymbol{x}}_{t|t-1} = \boldsymbol{G}_t\hat{\boldsymbol{x}}_{t-1|t-1}$$

$$\hat{\boldsymbol{P}}_{t|t-1} = \boldsymbol{G}_t\hat{\boldsymbol{P}}_{t-1|t-1}\boldsymbol{G}_t' + \boldsymbol{W}_t$$

Structural models with covariates

Local level with covariate

$$y_t = \ell_t + \beta z_t + \varepsilon_t$$
$$\ell_t = \ell_{t-1} + \xi_t$$

$$\boldsymbol{f}_t' = [1 \; z_t] \quad \boldsymbol{x}_t = \begin{bmatrix}\ell_t \\ \beta\end{bmatrix} \quad \boldsymbol{G} = \begin{bmatrix}1 & 0 \\ 0 & 1\end{bmatrix} \quad \boldsymbol{W}_t = \begin{bmatrix}\sigma_\xi^2 & 0 \\ 0 & 0\end{bmatrix}$$

- Assumes z_t is fixed and known (as in regression)
- Estimate of β is given by $x_{T|T}$.
- Equivalent to simple linear regression with time varying intercept.
- Easy to extend to multiple regression with additional terms.

Structural models with covariates

Local level with covariate

$$y_t = \ell_t + \beta z_t + \varepsilon_t$$
$$\ell_t = \ell_{t-1} + \xi_t$$

$$\boldsymbol{f}_t' = [1 \; z_t] \quad \boldsymbol{x}_t = \begin{bmatrix} \ell_t \\ \beta \end{bmatrix} \quad \boldsymbol{G} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \boldsymbol{W}_t = \begin{bmatrix} \sigma_\xi^2 & 0 \\ 0 & 0 \end{bmatrix}$$

- Assumes z_t is fixed and known (as in regression)
- Estimate of β is given by $\hat{x}_{T|T}$.
- Equivalent to simple linear regression with time varying intercept.
- Easy to extend to multiple regression with additional terms.

Structural models with covariates

Local level with covariate

$$y_t = \ell_t + \beta z_t + \varepsilon_t$$
$$\ell_t = \ell_{t-1} + \xi_t$$

$$\boldsymbol{f}_t' = [1\ z_t] \quad \boldsymbol{x}_t = \begin{bmatrix}\ell_t \\ \beta\end{bmatrix} \quad \boldsymbol{G} = \begin{bmatrix}1 & 0 \\ 0 & 1\end{bmatrix} \quad \boldsymbol{W}_t = \begin{bmatrix}\sigma_\xi^2 & 0 \\ 0 & 0\end{bmatrix}$$

- Assumes z_t is fixed and known (as in regression)
- Estimate of β is given by $\hat{x}_{T|T}$.
- Equivalent to simple linear regression with time varying intercept.
- Easy to extend to multiple regression with additional terms.

Structural models with covariates

Local level with covariate

$$y_t = \ell_t + \beta z_t + \varepsilon_t$$
$$\ell_t = \ell_{t-1} + \xi_t$$

$$\boldsymbol{f}_t' = [1 \; z_t] \quad \boldsymbol{x}_t = \begin{bmatrix} \ell_t \\ \beta \end{bmatrix} \quad \boldsymbol{G} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \boldsymbol{W}_t = \begin{bmatrix} \sigma_\xi^2 & 0 \\ 0 & 0 \end{bmatrix}$$

- Assumes z_t is fixed and known (as in regression)
- Estimate of β is given by $\hat{x}_{T|T}$.
- Equivalent to simple linear regression with time varying intercept.
- Easy to extend to multiple regression with additional terms.

Structural models with covariates

Local level with covariate

$$y_t = \ell_t + \beta z_t + \varepsilon_t$$
$$\ell_t = \ell_{t-1} + \xi_t$$

$$\boldsymbol{f}_t' = [1 \; z_t] \quad \boldsymbol{x}_t = \begin{bmatrix} \ell_t \\ \beta \end{bmatrix} \quad \boldsymbol{G} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \boldsymbol{W}_t = \begin{bmatrix} \sigma_\xi^2 & 0 \\ 0 & 0 \end{bmatrix}$$

- Assumes z_t is fixed and known (as in regression)
- Estimate of β is given by $\hat{x}_{T|T}$.
- Equivalent to simple linear regression with time varying intercept.
- Easy to extend to multiple regression with additional terms.

Structural models with covariates

Local level with covariate

$$y_t = \ell_t + \beta z_t + \varepsilon_t$$
$$\ell_t = \ell_{t-1} + \xi_t$$

$$\boldsymbol{f}_t' = [1\ z_t] \quad \boldsymbol{x}_t = \begin{bmatrix} \ell_t \\ \beta \end{bmatrix} \quad \boldsymbol{G} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \boldsymbol{W}_t = \begin{bmatrix} \sigma_\xi^2 & 0 \\ 0 & 0 \end{bmatrix}$$

- Assumes z_t is fixed and known (as in regression)
- Estimate of β is given by $\hat{x}_{T|T}$.
- Equivalent to simple linear regression with time varying intercept.
- Easy to extend to multiple regression with additional terms.

Time varying regression

Simple linear regression with time varying parameters

$$y_t = \ell_t + \beta_t z_t + \varepsilon_t$$
$$\ell_t = \ell_{t-1} + \xi_t$$
$$\beta_t = \beta_{t-1} + \zeta_t$$

$$\boldsymbol{f}_t' = [1\ z_t] \quad \boldsymbol{x}_t = \begin{bmatrix} \ell_t \\ \beta_t \end{bmatrix} \quad \boldsymbol{G} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \boldsymbol{W}_t = \begin{bmatrix} \sigma_\xi^2 & 0 \\ 0 & \sigma_\zeta^2 \end{bmatrix}$$

- Allows for a linear regression with parameters that change slowly over time.
- Parameters follow independent random walks.
- Estimates of parameters given by Kalman filter.

Time varying regression

Simple linear regression with time varying parameters

$$
\begin{aligned}
y_t &= \ell_t + \beta_t z_t + \varepsilon_t \\
\ell_t &= \ell_{t-1} + \xi_t \\
\beta_t &= \beta_{t-1} + \zeta_t
\end{aligned}
$$

$$
\boldsymbol{f}_t' = [1\ z_t] \quad \boldsymbol{x}_t = \begin{bmatrix} \ell_t \\ \beta_t \end{bmatrix} \quad \boldsymbol{G} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \boldsymbol{W}_t = \begin{bmatrix} \sigma_\xi^2 & 0 \\ 0 & \sigma_\zeta^2 \end{bmatrix}
$$

- Allows for a linear regression with parameters that change slowly over time.
- Parameters follow independent random walks.
- Estimates of parameters given by $\hat{x}_{t|t}$ or $\hat{x}_{t|T}$.

Time varying regression

Simple linear regression with time varying parameters

$$
\begin{aligned}
y_t &= \ell_t + \beta_t z_t + \varepsilon_t \\
\ell_t &= \ell_{t-1} + \xi_t \\
\beta_t &= \beta_{t-1} + \zeta_t
\end{aligned}
$$

$$
\boldsymbol{f}_t' = [1\ z_t] \quad \boldsymbol{x}_t = \begin{bmatrix} \ell_t \\ \beta_t \end{bmatrix} \quad \boldsymbol{G} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \boldsymbol{W}_t = \begin{bmatrix} \sigma_\xi^2 & 0 \\ 0 & \sigma_\zeta^2 \end{bmatrix}
$$

- Allows for a linear regression with parameters that change slowly over time.
- Parameters follow independent random walks.
- Estimates of parameters given by $\hat{x}_{t|t}$ or $\hat{x}_{t|T}$.

Time varying regression

Simple linear regression with time varying parameters

$$y_t = \ell_t + \beta_t z_t + \varepsilon_t$$
$$\ell_t = \ell_{t-1} + \xi_t$$
$$\beta_t = \beta_{t-1} + \zeta_t$$

$$\boldsymbol{f}_t' = [1\ z_t] \quad \boldsymbol{x}_t = \begin{bmatrix} \ell_t \\ \beta_t \end{bmatrix} \quad \boldsymbol{G} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \boldsymbol{W}_t = \begin{bmatrix} \sigma_\xi^2 & 0 \\ 0 & \sigma_\zeta^2 \end{bmatrix}$$

- Allows for a linear regression with parameters that change slowly over time.
- Parameters follow independent random walks.
- Estimates of parameters given by $\hat{x}_{t|t}$ or $\hat{x}_{t|T}$.

Time varying regression

Simple linear regression with time varying parameters

$$y_t = \ell_t + \beta_t z_t + \varepsilon_t$$
$$\ell_t = \ell_{t-1} + \xi_t$$
$$\beta_t = \beta_{t-1} + \zeta_t$$

$$\boldsymbol{f}_t' = [1\ z_t] \quad \boldsymbol{x}_t = \begin{bmatrix} \ell_t \\ \beta_t \end{bmatrix} \quad \boldsymbol{G} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \boldsymbol{W}_t = \begin{bmatrix} \sigma_\xi^2 & 0 \\ 0 & \sigma_\zeta^2 \end{bmatrix}$$

- Allows for a linear regression with parameters that change slowly over time.
- Parameters follow independent random walks.
- Estimates of parameters given by $\hat{x}_{t|t}$ or $\hat{x}_{t|T}$.

Updating ("online") regression

- Same idea can be used to estimate a regression iteratively as new data arrives.

Simple linear regression with updating parameters

$$
\begin{aligned}
y_t &= \ell_t + \beta_t z_t + \varepsilon_t \\
\ell_t &= \ell_{t-1} + \xi_t \\
\beta_t &= \beta_{t-1} + \zeta_t
\end{aligned}
$$

$$\boldsymbol{f}_t' = [1\ z_t] \quad \boldsymbol{x}_t = \begin{bmatrix} \ell_t \\ \beta_t \end{bmatrix} \quad \boldsymbol{G} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \boldsymbol{W}_t = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Updating ("online") regression

- Same idea can be used to estimate a regression iteratively as new data arrives.

Simple linear regression with updating parameters

$$
\begin{aligned}
y_t &= \ell_t + \beta_t z_t + \varepsilon_t \\
\ell_t &= \ell_{t-1} + \xi_t \\
\beta_t &= \beta_{t-1} + \zeta_t
\end{aligned}
$$

$$\boldsymbol{f}_t' = [1\ z_t] \quad \boldsymbol{x}_t = \begin{bmatrix} \ell_t \\ \beta_t \end{bmatrix} \quad \boldsymbol{G} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \boldsymbol{W}_t = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

- Updated parameter estimates given by $\hat{x}_{t|t}$.
- Recursive residuals given by $y_t - \hat{y}_{t|t-1}$.

Updating ("online") regression

- Same idea can be used to estimate a regression iteratively as new data arrives.

Simple linear regression with updating parameters

$$y_t = \ell_t + \beta_t z_t + \varepsilon_t$$
$$\ell_t = \ell_{t-1} + \xi_t$$
$$\beta_t = \beta_{t-1} + \zeta_t$$

$$\boldsymbol{f}_t' = [1 \; z_t] \quad \boldsymbol{x}_t = \begin{bmatrix} \ell_t \\ \beta_t \end{bmatrix} \quad \boldsymbol{G} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \boldsymbol{W}_t = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

- Updated parameter estimates given by $\hat{x}_{t|t}$.
- Recursive residuals given by $y_t - \hat{y}_{t|t-1}$.

Updating ("online") regression

- Same idea can be used to estimate a regression iteratively as new data arrives.

Simple linear regression with updating parameters

$$y_t = \ell_t + \beta_t z_t + \varepsilon_t$$
$$\ell_t = \ell_{t-1} + \xi_t$$
$$\beta_t = \beta_{t-1} + \zeta_t$$

$\boldsymbol{f}_t' = [1 \ z_t] \quad \boldsymbol{x}_t = \begin{bmatrix} \ell_t \\ \beta_t \end{bmatrix} \quad \boldsymbol{G} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \boldsymbol{W}_t = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$

- Updated parameter estimates given by $\hat{x}_{t|t}$.
- Recursive residuals given by $y_t - \hat{y}_{t|t-1}$.

Updating ("online") regression

- Same idea can be used to estimate a regression iteratively as new data arrives.

Simple linear regression with updating parameters

$$y_t = \ell_t + \beta_t z_t + \varepsilon_t$$
$$\ell_t = \ell_{t-1} + \xi_t$$
$$\beta_t = \beta_{t-1} + \zeta_t$$

$$\boldsymbol{f}_t' = [1 \; z_t] \quad \boldsymbol{x}_t = \begin{bmatrix} \ell_t \\ \beta_t \end{bmatrix} \quad \boldsymbol{G} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \boldsymbol{W}_t = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

- Updated parameter estimates given by $\hat{x}_{t|t}$.
- Recursive residuals given by $y_t - \hat{y}_{t|t-1}$.