Time Series and Sequence Learning

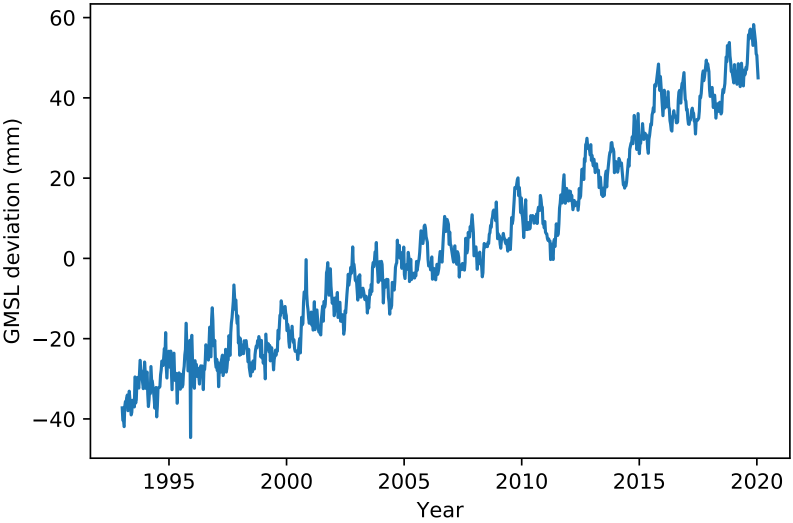
Lecture 4 – Classical Decomposition and Kalman Filter

Johan Alenlöv, Linköping University

2020-09-09

Summary

The goal for this lecture is to model a time-series by decomposing the time-series into different signals.



Summary

The goal for this lecture is to model a time-series by decomposing the time-series into different signals.

For today we will talk about

1. The classical decomposition
2. Modelling the trend
3. Kalman Filter
4. Calculating the likelihood and estimating parameters
5. Forecasting and missing data

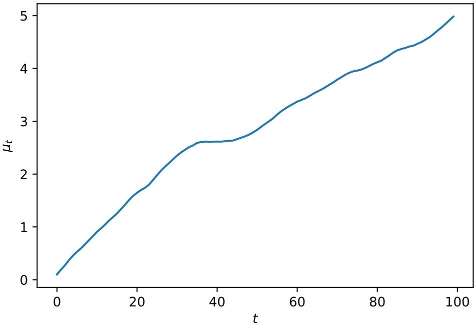
Time Series Modelling

Our goal for this lecture is to model a **time-series** $y_1, y_2, \dots y_n$ by decomposing it into unobserved stochastic processes. The three parts we will decompose the model into is the **trend**, **seasonal**, and **noise**.

Time Series Modelling

Our goal for this lecture is to model a **time-series** $y_1, y_2, \ldots y_n$ by decomposing it into unobserved stochastic processes. The three parts we will decompose the model into is the **trend**, **seasonal**, and **noise**.

μ_t The **trend** component of the time-series.

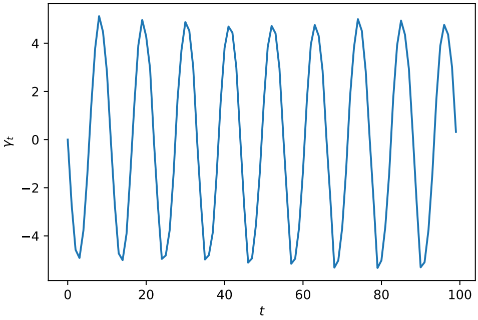


Time Series Modelling

Our goal for this lecture is to model a **time-series** $y_1, y_2, \ldots y_n$ by decomposing it into unobserved stochastic processes. The three parts we will decompose the model into is the **trend**, **seasonal**, and **noise**.

μ_t The **trend** component of the time-series.

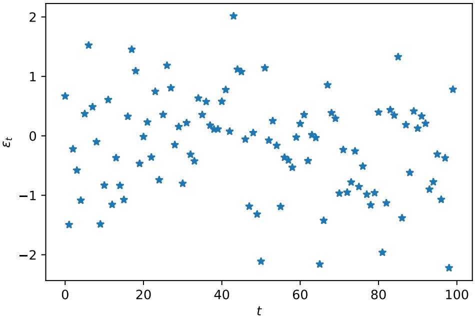
γ_t The **seasonal** component of the time-series.



Time Series Modelling

Our goal for this lecture is to model a **time-series** $y_1, y_2, \ldots y_n$ by decomposing it into unobserved stochastic processes. The three parts we will decompose the model into is the **trend**, **seasonal**, and **noise**.

- μ_t The **trend** component of the time-series.
- γ_t The **seasonal** component of the time-series.
- ε_t The **noise** component of the time-series.



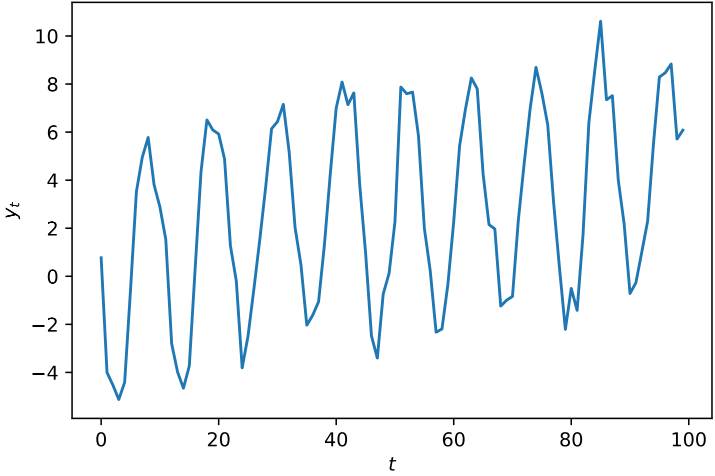
Time Series Modelling (cont.)

There are two main ways of combining these parts.

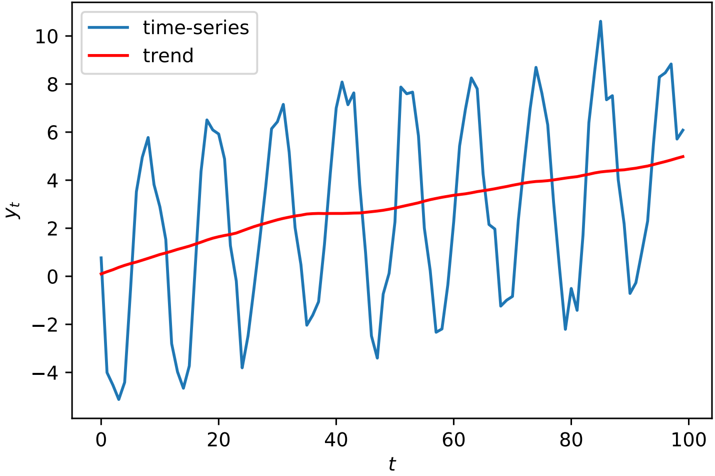
1. The most common way is an **additive model** where,

$$y_t = \underbrace{\mu_t}_{\text{trend}} + \underbrace{\gamma_t}_{\text{seasonal}} + \underbrace{\varepsilon_t}_{\text{noise}}.$$

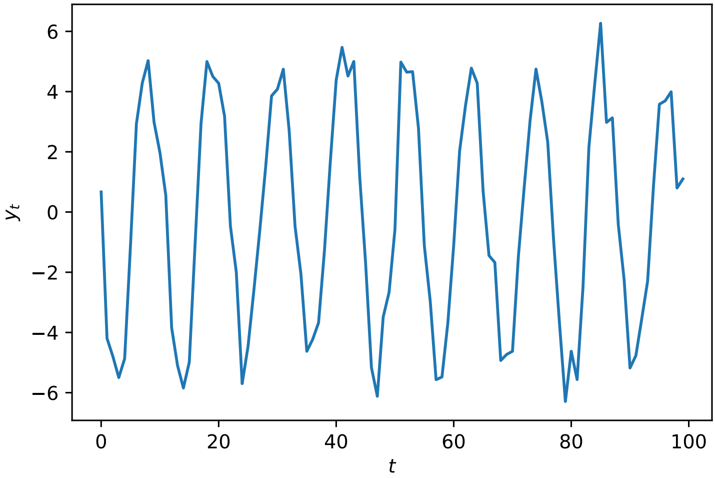
ex) Additive Model



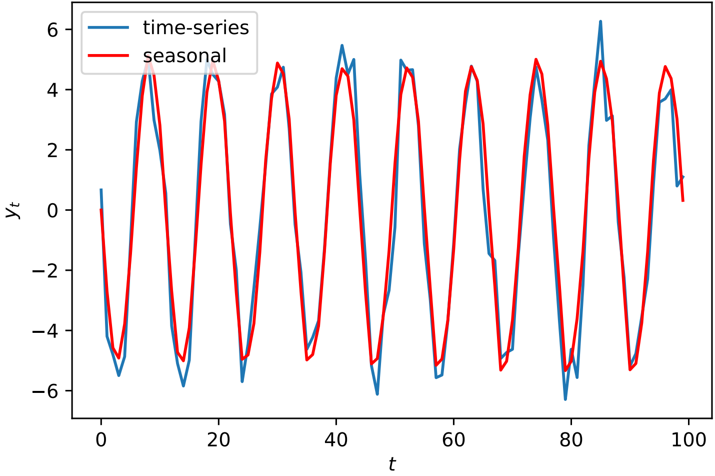
ex) Additive Model



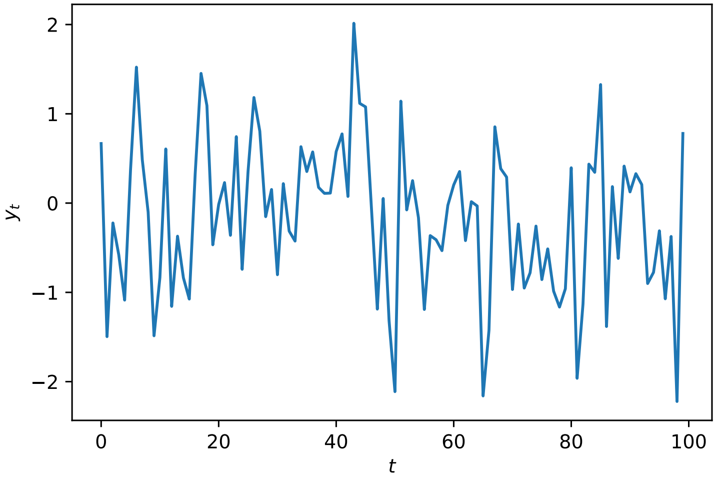
ex) Additive Model



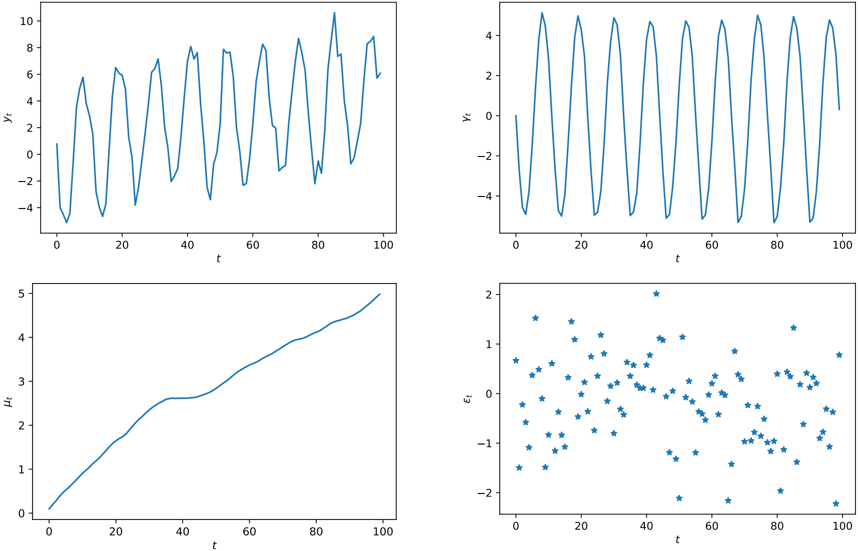
ex) Additive Model



ex) Additive Model



ex) Additive Model



$$y_t = \underbrace{\mu_t}_{\text{trend}} + \underbrace{\gamma_t}_{\text{seasonal}} + \underbrace{\varepsilon_t}_{\text{noise}}$$

Time Series Modelling (cont.)

There are two main ways of combining these parts.

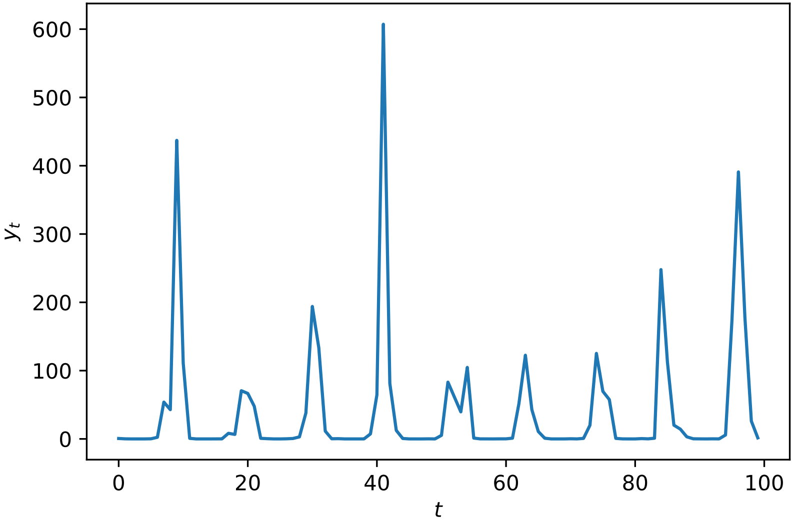
1. The most common way is an **additive model** where,

$$y_t = \underbrace{\mu_t}_{\text{trend}} + \underbrace{\gamma_t}_{\text{seasonal}} + \underbrace{\varepsilon_t}_{\text{noise}}.$$

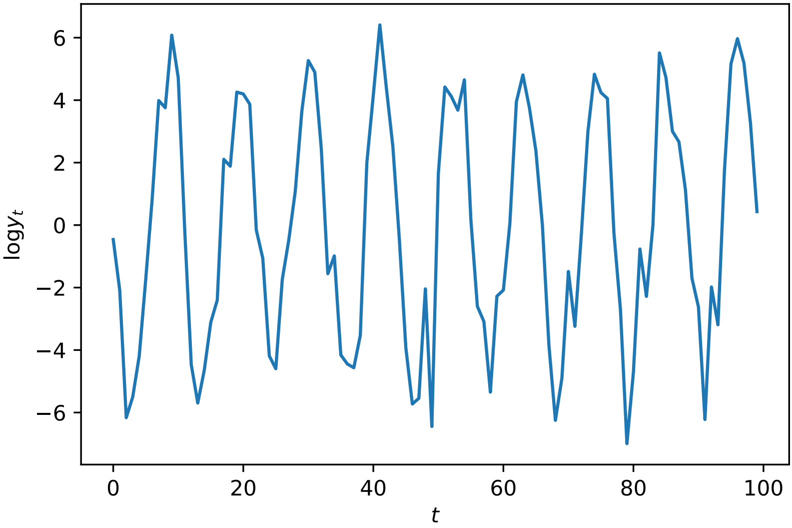
2. The other way is a **multiplicative model** where,

$$y_t = \underbrace{\mu_t}_{\text{trend}} \times \underbrace{\gamma_t}_{\text{seasonal}} \times \underbrace{\varepsilon_t}_{\text{noise}}.$$

ex) Multiplicative Model



ex) Multiplicative Model



Time Series Modelling (cont.)

There are two main ways of combining these parts.

1. The most common way is an **additive model** where,

$$y_t = \underbrace{\mu_t}_{\text{trend}} + \underbrace{\gamma_t}_{\text{seasonal}} + \underbrace{\varepsilon_t}_{\text{noise}}.$$

2. The other way is a **multiplicative model** where,

$$y_t = \underbrace{\mu_t}_{\text{trend}} \times \underbrace{\gamma_t}_{\text{seasonal}} \times \underbrace{\varepsilon_t}_{\text{noise}}.$$

As we could see, it is often possible to look at the **logarithm** of the multiplicative model and get an additive model

$$\log y_t = \underbrace{\log \mu_t}_{\text{trend}} + \underbrace{\log \gamma_t}_{\text{seasonal}} + \underbrace{\log \varepsilon_t}_{\text{noise}}.$$

Modelling the Trend

- We skip the **seasonal** part γ_t.

Modelling the Trend

- We skip the **seasonal** part γ_t.
- Work with **time-series** y_t, an additive model with Gaussian noise.

$$y_t = \mu_t + \varepsilon_t, \qquad \varepsilon_t \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_\varepsilon^2).$$

Modelling the Trend

- We skip the **seasonal** part γ_t.
- Work with **time-series** y_t, an additive model with Gaussian noise.

$$y_t = \mu_t + \varepsilon_t, \qquad \varepsilon_t \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_\varepsilon^2).$$

- Focus on modelling of the **trend** μ_t today.

Modelling the Trend

- We skip the **seasonal** part γ_t.
- Work with **time-series** y_t, an additive model with Gaussian noise.

$$y_t = \mu_t + \varepsilon_t, \qquad \varepsilon_t \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_\varepsilon^2).$$

- Focus on modelling of the **trend** μ_t today.
- Recall the AR model of order p from previous lecture,

$$\mu_t = a_1\mu_{t-1} + a_2\mu_{t-2} + \ldots + a_p\mu_{t-p} + w_t, \qquad w_t \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2).$$

Modelling the Trend

- We skip the **seasonal** part γ_t.
- Work with **time-series** y_t, an additive model with Gaussian noise.

$$y_t = \mu_t + \varepsilon_t, \qquad \varepsilon_t \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_\varepsilon^2).$$

- Focus on modelling of the **trend** μ_t today.
- Recall the AR model of order p from previous lecture,

$$\mu_t = a_1 \mu_{t-1} + a_2 \mu_{t-2} + \ldots + a_p \mu_{t-p} + w_t, \qquad w_t \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2).$$

- Will start with a **simple random walk**.

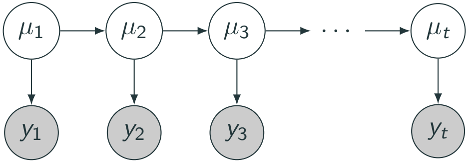
Simple Random Walk: The stochastic process $\mu_1, \mu_2, \ldots$ is a **simple random walk** if $\mu_{t+1} = \mu_t + \eta_t$ where $\eta_t \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_\eta^2)$

Basic State-Space Model

The Local Level Model Two stochastic processes $y_1, y_2, y_3, \ldots$ and $\mu_1, \mu_2, \mu_3, \ldots$

$$
\begin{aligned}
y_t &= \mu_t + \varepsilon_t, & \varepsilon_t &\overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_\varepsilon^2) \\
\mu_{t+1} &= \mu_t + \eta_t, & \eta_t &\overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_\eta^2) \\
& & \mu_1 &\sim \mathcal{N}(a_1, P_1)
\end{aligned}
$$

This is a **state-space model (SSM)**, where y_t is the **observed** time-series and μ_t is the **unobserved** process.

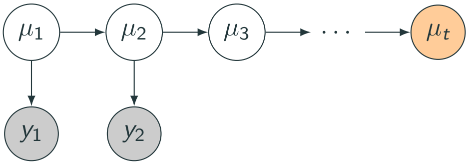


Filtering, Smoothing, and Predicting

Given a time-series $y_{1:n} = (y_1, y_2, \ldots, y_n)$ we wish to calculate the distribution of μ_t conditioned on the observed time-series $y_{1:n}$.

This problem changes depending on the relationship of n and t.

$n < t$: This is known as the **forecasting** problem.



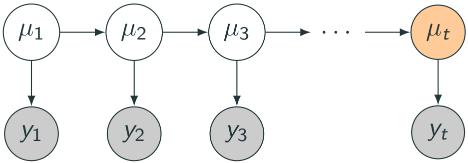
Filtering, Smoothing, and Predicting

Given a time-series $y_{1:n} = (y_1, y_2, \ldots, y_n)$ we wish to calculate the distribution of μ_t conditioned on the observed time-series $y_{1:n}$.

This problem changes depending on the relationship of n and t.

$n < t$: This is known as the **forecasting** problem.

$n = t$: This is known as the **filtering** problem.



Filtering, Smoothing, and Predicting

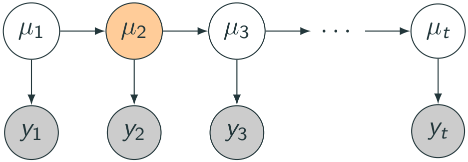
Given a time-series $y_{1:n} = (y_1, y_2, \ldots, y_n)$ we wish to calculate the distribution of μ_t conditioned on the observed time-series $y_{1:n}$.

This problem changes depending on the relationship of n and t.

$n < t$: This is known as the **forecasting** problem.

$n = t$: This is known as the **filtering** problem.

$n > t$: This is known as the **smoothing** problem.



Calculating the Filter Distribution

We will now focus on finding the **filter distribution** and **one step ahead predictor**.

That is the distribution of $\mu_t \mid y_{1:t}$ and $\mu_t \mid y_{1:t-1}$ in the local-level model with Gaussian noise.

Let $\hat{\mu}_{i \mid j} = \mathbb{E}[\mu_i \mid y_{1:j}]$ and $P_{i \mid j} = \mathrm{Var}[\mu_i \mid y_{1:j}]$.

Calculating the Filter Distribution

We will now focus on finding the **filter distribution** and **one step ahead predictor**.

That is the distribution of $\mu_t \,|\, y_{1:t}$ and $\mu_t \,|\, y_{1:t-1}$ in the local-level model with Gaussian noise.

Let $\hat{\mu}_{i\,|\,j} = \mathbb{E}[\mu_i \,|\, y_{1:j}]$ and $P_{i\,|\,j} = \mathrm{Var}[\mu_i \,|\, y_{1:j}]$.

Will proceed as follows:

1. Assume that $\mu_t \,|\, y_{1:t-1} \sim \mathcal{N}(\hat{\mu}_{t\,|\,t-1}, P_{t\,|\,t-1})$.
2. Show that $\mu_t \,|\, y_{1:t} \sim \mathcal{N}(\hat{\mu}_{t\,|\,t}, P_{t\,|\,t})$
3. Show that $\mu_{t+1} \,|\, y_{1:t} \sim \mathcal{N}(\hat{\mu}_{t+1\,|\,t}, P_{t+1\,|\,t})$

Calculating the Filter Distribution

We will now focus on finding the **filter distribution** and **one step ahead predictor**.

That is the distribution of $\mu_t \mid y_{1:t}$ and $\mu_t \mid y_{1:t-1}$ in the local-level model with Gaussian noise.

Let $\hat{\mu}_{i \mid j} = \mathbb{E}[\mu_i \mid y_{1:j}]$ and $P_{i \mid j} = \mathrm{Var}[\mu_i \mid y_{1:j}]$.

Will proceed as follows:

1. Assume that $\mu_t \mid y_{1:t-1} \sim \mathcal{N}(\hat{\mu}_{t \mid t-1}, P_{t \mid t-1})$.
2. Show that $\mu_t \mid y_{1:t} \sim \mathcal{N}(\hat{\mu}_{t \mid t}, P_{t \mid t})$
3. Show that $\mu_{t+1} \mid y_{1:t} \sim \mathcal{N}(\hat{\mu}_{t+1 \mid t}, P_{t+1 \mid t})$

And find the expressions for the mean and variance of the Gaussian distributions.

The Kalman Filter

Kalman Filter for local-level model

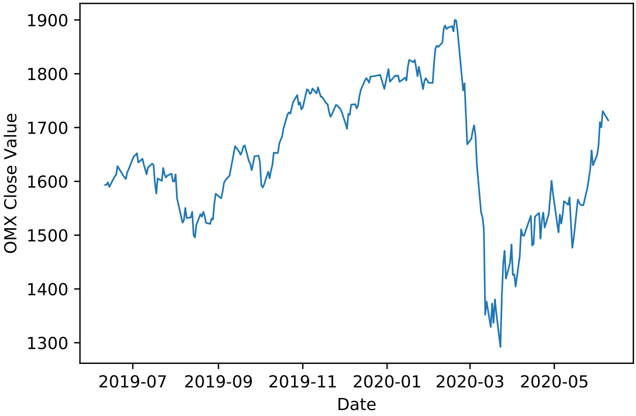
For each iteration $t = 1, 2, 3, \ldots$ repeat the following steps:

- Measurement updates
  1. Forecasting error: $v_t = y_t - \hat{\mu}_{t\,|\,t-1}$
  2. Forecasting variance: $F_t = P_{t\,|\,t-1} + \sigma_\varepsilon^2$
  3. Kalman gain: $K_t = P_{t\,|\,t-1}/F_t$
  4. Filter mean: $\hat{\mu}_{t\,|\,t} = \hat{\mu}_{t\,|\,t-1} + K_t v_t$
  5. Filter variance: $P_{t\,|\,t} = P_{t\,|\,t-1}(1 - K_t)$
- Time updates
  6. Predictor mean: $\hat{\mu}_{t+1\,|\,t} = \hat{\mu}_{t\,|\,t}$
  7. Predictor variance: $P_{t+1\,|\,t} = P_{t\,|\,t} + \sigma_\eta^2$

Initialized using $\hat{\mu}_{1\,|\,0} = a_1$ and $P_{1\,|\,0} = P_1$.

ex) The Kalman Filter

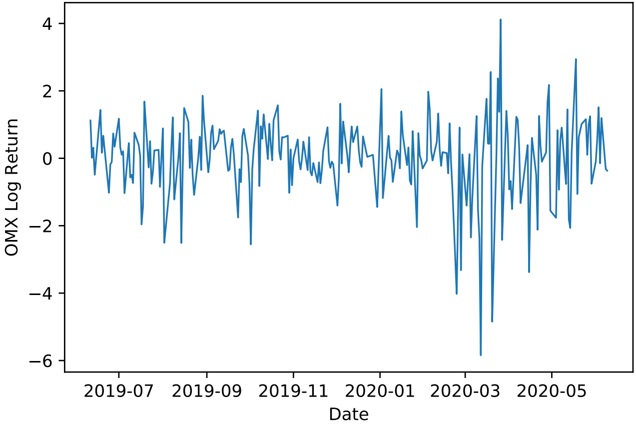
We look at the time-series $S_1, \ldots, S_{248}$ which are the **OMXS30** closing values over a year.



ex) The Kalman Filter

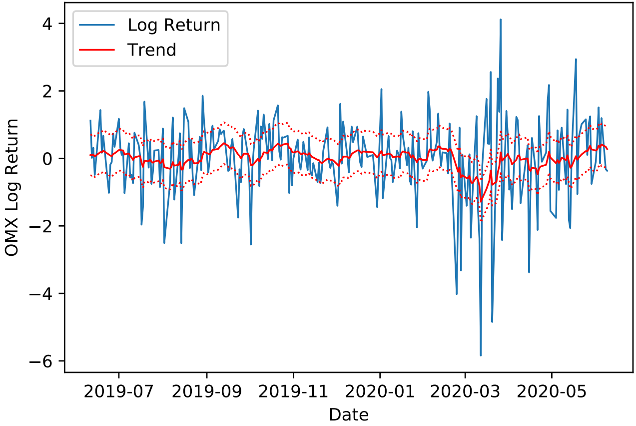
Typically we look at the **log-returns**

$$y_t = 100 \cdot \log\left(\frac{S_t}{S_{t-1}}\right).$$



ex) The Kalman Filter

We run the **Kalman filter** using $\hat{\mu}_1 = 0, P_1 = 0.1, \sigma^2_\varepsilon = 1$, and $\sigma^2_\eta = 0.01$



Calculating the Likelihood

- So far we assumed that the parameter $\theta = (a_1, P_1, \sigma_\varepsilon^2, \sigma_\eta^2)$ are all **known**.

Calculating the Likelihood

- So far we assumed that the parameter $\theta = (a_1, P_1, \sigma_\varepsilon^2, \sigma_\eta^2)$ are all **known**.
- In practice some or all of them are **unknown** and needs to be estimated.

Calculating the Likelihood

- So far we assumed that the parameter $\theta = (a_1, P_1, \sigma_\varepsilon^2, \sigma_\eta^2)$ are all **known**.
- In practice some or all of them are **unknown** and needs to be estimated.
- The **maximum-likelihood** estimator gives the parameters which maximizes the **likelihood**.

Calculating the Likelihood

- So far we assumed that the parameter $\theta = (a_1, P_1, \sigma_\varepsilon^2, \sigma_\eta^2)$ are all **known**.
- In practice some or all of them are **unknown** and needs to be estimated.
- The **maximum-likelihood** estimator gives the parameters which maximizes the **likelihood**.
- The **likelihood** is given by,

$$L(\theta) = p_\theta(y_{1:n}) = p_\theta(y_1) \prod_{t=2}^{n} p_\theta(y_t \mid y_{1:t-1}).$$

Calculating the Likelihood

- So far we assumed that the parameter $\theta = (a_1, P_1, \sigma_\varepsilon^2, \sigma_\eta^2)$ are all **known**.
- In practice some or all of them are **unknown** and needs to be estimated.
- The **maximum-likelihood** estimator gives the parameters which maximizes the **likelihood**.
- The **likelihood** is given by,

$$L(\theta) = p_\theta(y_{1:n}) = p_\theta(y_1) \prod_{t=2}^{n} p_\theta(y_t \mid y_{1:t-1}).$$

- More commonly we use the **log-likelihood**,

$$\ell(\theta) = \log L(\theta) = \log p_\theta(y_1) + \sum_{t=2}^{n} \log p_\theta(y_t \mid y_{1:t-1}).$$

Parameter Estimation Using Kalman Filter

From the derivation of the **Kalman filter** we have that:

$$y_t \mid y_{1:t-1} \sim \mathcal{N}(\hat{\mu}_{t \mid t-1}(\theta), F_t(\theta)),$$

given that we get that the components of the log-likelihood is calculated by

$$\log p_\theta(y_t \mid y_{1:t-1}) = -\frac{1}{2}\left(\log 2\pi + \log F_t(\theta) + \frac{(y_t - \hat{\mu}_{t \mid t-1}(\theta))^2}{F_t(\theta)}\right).$$

Parameter Estimation Using Kalman Filter

From the derivation of the **Kalman filter** we have that:

$$y_t \mid y_{1:t-1} \sim \mathcal{N}(\hat{\mu}_{t \mid t-1}(\theta), F_t(\theta)),$$

given that we get that the components of the log-likelihood is calculated by

$$\log p_{\theta}(y_t \mid y_{1:t-1}) = -\frac{1}{2}\left(\log 2\pi + \log F_t(\theta) + \frac{(y_t - \hat{\mu}_{t \mid t-1}(\theta))^2}{F_t(\theta)}\right).$$

The **log-likelihood** for the local-level model is given by

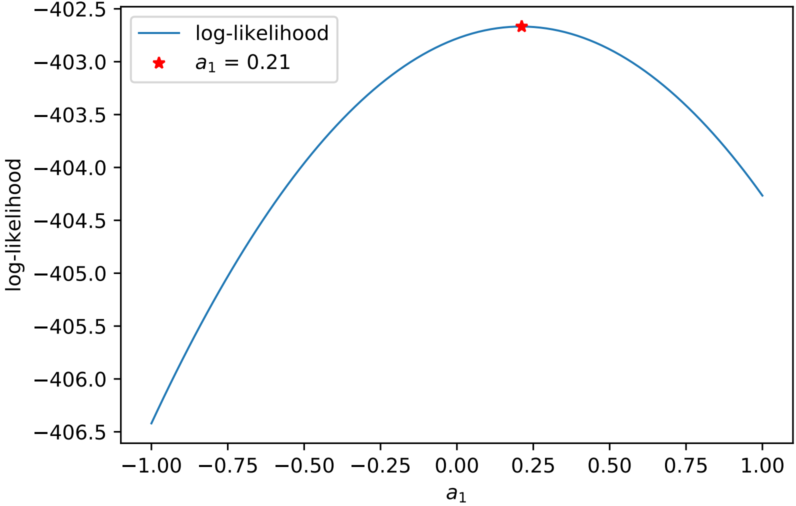
$$\ell(\theta) = -\frac{n}{2}\log 2\pi - \frac{1}{2}\sum_{t=1}^{n}\left(\log F_t(\theta) + \frac{(y_t - \hat{\mu}_{t \mid t-1}(\theta))^2}{F_t(\theta)}\right)$$

ex) Estimating Initial Parameters

We now look at estimating a_1 while keeping the other parameters fixed in the **OMXS30** log returns.

ex) Estimating Initial Parameters

We now look at estimating a_1 while keeping the other parameters fixed in the **OMXS30** log returns.



Forecasting

- We switch over the problem of **forecasting**.

Forecasting

- We switch over the problem of **forecasting**.
- Focus on finding the distribution of $\mu_{t+j} \mid y_{1:t}$.

Forecasting

- We switch over the problem of **forecasting**.
- Focus on finding the distribution of $\mu_{t+j} \mid y_{1:t}$.
- As previously this distribution will be **Gaussian**, need the **mean** $\hat{\mu}_{t+j\mid t}$ and **variance** $P_{t+j\mid t}$.

Forecasting

- We switch over the problem of **forecasting**.
- Focus on finding the distribution of $\mu_{t+j} \mid y_{1:t}$.
- As previously this distribution will be **Gaussian**, need the **mean** $\hat{\mu}_{t+j\mid t}$ and **variance** $P_{t+j\mid t}$.
- Direct calculations give

$$\hat{\mu}_{t+j\mid t} = \mathbb{E}[\mu_{t+j} \mid y_{1:t}] = \mathbb{E}[\mu_t + \sum_{i=0}^{j-1} \eta_{t+i} \mid y_{1:t}] = \hat{\mu}_{t\mid t},$$

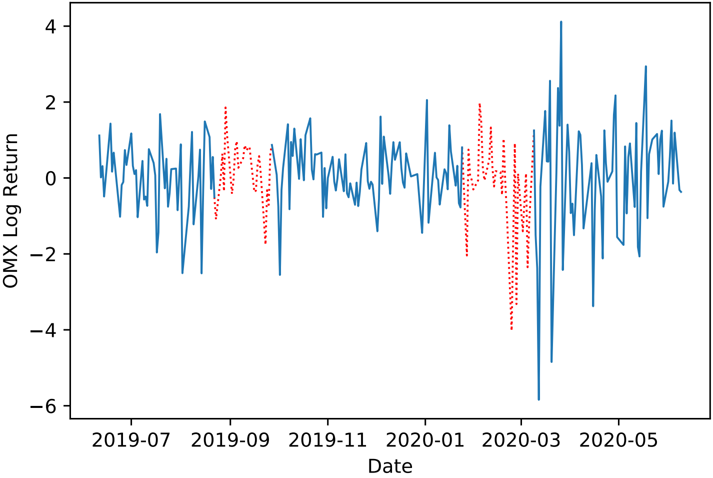
$$\begin{aligned} P_{t+j\mid t} &= \mathrm{Var}[\mu_{t+j} \mid y_{1:t}] = \mathrm{Var}[\mu_t + \sum_{i=0}^{j-1} \eta_{t+i} \mid y_{1:t}] \\ &= P_{t\mid t} + \sum_{i=0}^{j-1} \sigma_\eta^2 = P_{t\mid t} + j \cdot \sigma_\eta^2. \end{aligned}$$

Missing Data

- A problem related to forecasting is when we have **missing observations**.

Missing Data

- A problem related to forecasting is when we have **missing observations**.
- Assume that we are **missing** a set of observations $y_{\tau:\tau^*}$.



Missing Data

- A problem related to forecasting is when we have **missing observations**.
- Assume that we are **missing** a set of observations $y_{\tau:\tau^*}$.
- Running the **Kalman filter** for these **missing observations** will be like **forecasting**.
  For $t = \tau, \dots, \tau^*$:

$$
\begin{aligned}
\hat{\mu}_{t \mid t} &= \hat{\mu}_{t \mid \tau-1} = \hat{\mu}_{\tau \mid \tau-1}, \\
\hat{\mu}_{t+1 \mid t} &= \hat{\mu}_{t+1 \mid \tau-1} = \hat{\mu}_{\tau \mid \tau-1}, \\
P_{t \mid t} &= P_{t \mid \tau-1} = P_{\tau \mid \tau-1} + (t-\tau)\sigma_\eta^2, \\
P_{t+1 \mid t} &= P_{t+1 \mid \tau-1} = P_{\tau \mid \tau-1} + (t-\tau+1)\sigma_\eta^2.
\end{aligned}
$$

The Kalman Filter Again

Kalman Filter for local-level model

For each iteration $t = 1, 2, 3, \ldots$ repeat the following steps:

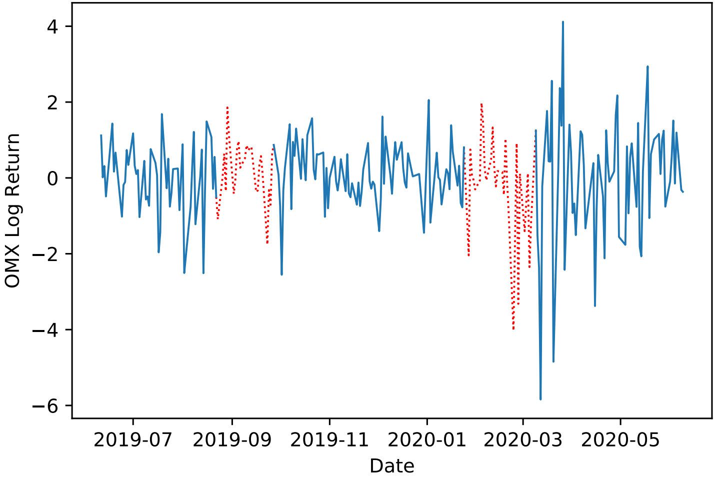
If y_t is missing, set $\hat{\mu}_{t\,|\,t} = \hat{\mu}_{t\,|\,t-1}$, $P_{t\,|\,t} = P_{t\,|\,t-1}$ and skip to the prediction updates

- Measurement updates
  1. Forecasting error: $v_t = y_t - \hat{\mu}_{t\,|\,t-1}$
  2. Forecasting variance: $F_t = P_{t\,|\,t-1} + \sigma_\varepsilon^2$
  3. Kalman gain: $K_t = P_{t\,|\,t-1}/F_t$
  4. Filter mean: $\hat{\mu}_{t\,|\,t} = \hat{\mu}_{t\,|\,t-1} + K_t v_t$
  5. Filter variance: $P_{t\,|\,t} = P_{t\,|\,t-1}(1 - K_t)$
- Time updates
  6. Predictor mean: $\hat{\mu}_{t+1\,|\,t} = \hat{\mu}_{t\,|\,t}$
  7. Predictor variance: $P_{t+1\,|\,t} = P_{t\,|\,t} + \sigma_\eta^2$

Initialized using $\hat{\mu}_{1\,|\,0} = a_1$ and $P_{1\,|\,0} = P_1$.

ex) Missing Observations and Forecasting

We again look at the **OMXS30** log-returns. We remove some of the observations from the time-series and again run the **Kalman filter** on this data.



ex) Missing Observations and Forecasting

We again look at the **OMXS30** log-returns. We remove some of the observations from the time-series and again run the **Kalman filter** on this data.