Preliminary Motivation

From Wong, et al. (2016, *AOAS*), "Detecting abrupt changes in the Spectra of High-Energy Astrophysical Sources":

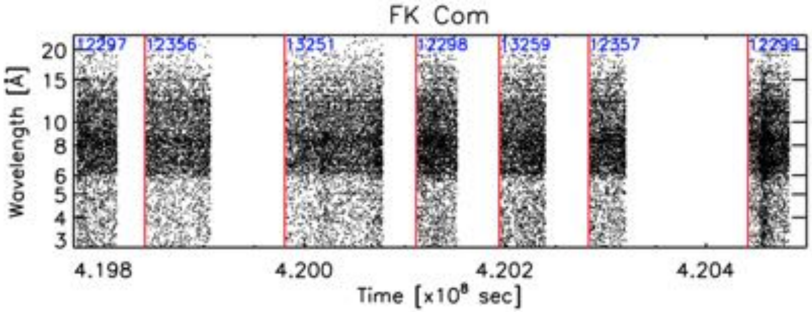


FIG. 4. *Observed data. Each point represents a photon detected with HETGS + ACIS-S configuration of Chandra (including both low- and high-energy gratings data). The wavelengths are given in Ångströms, and the time is in spacecraft clock seconds. The ObsIDs of each segment are marked to the right of the red vertical line representing the starting time of the observation segment. The collected data are sparse, but some changes in the density of points can be discerned even by eye, indicating temporal and spectral variations.*

How does the spectrum evolve with time? What properties would an ideal model posssess?

State Space Models

The state space representation of a time series model provides an extremely flexible framework within a wide range of different models can be fit.

The Kalman Filter provides a means to smoothing and making predictions (forecasts) with the resulting model.

The value is in its generality, but understanding the approach is helped by first considering some simpler models.

Outline

Overview of Time Domain Methods

Markov Models

State Space Representation

Kalman Filter

Python Examples and Exercises

Overview of Time Domain Methods

In the field of Statistics, distinction is made between time domain methods of time series, and frequency domain methods.

Frequency domain is characterized by modelling periodic structure using Fourier series, inspecting the periodogram, etc.

Time domain methods are more commonly used in fields such as finance, but can be useful in the sciences as well.

The Autocovariance Function

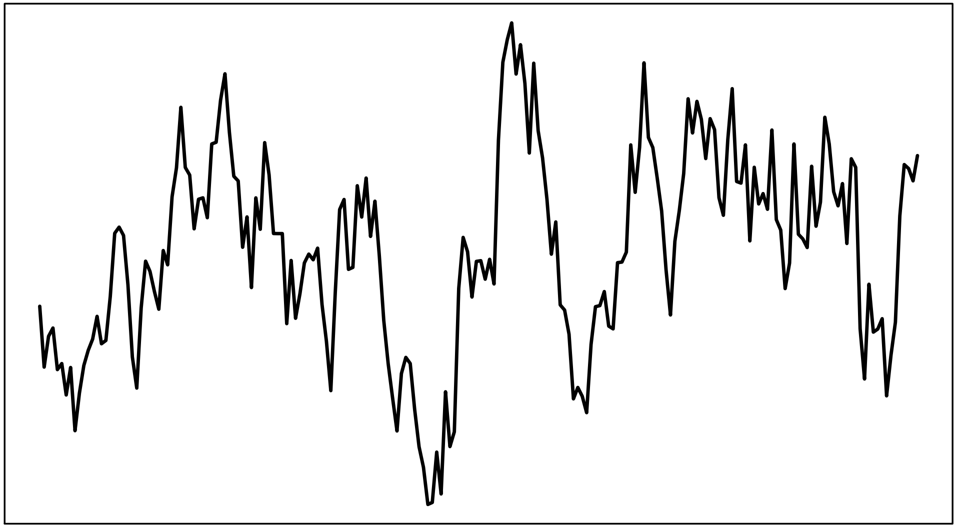
In the time domain, it is fundamental to consider the strength of the correlation at different time lags. To this end, the autocovariance function of the series $\{X_t\}$ is defined as

$$\gamma(r, s) = \mathrm{Cov}(X_r, X_s)$$

If the value of $\gamma(r, s)$ depends only on the lag $|r - s|$, and the (unconditional) mean of X_t does not depend on t, then the process is said to be stationary.

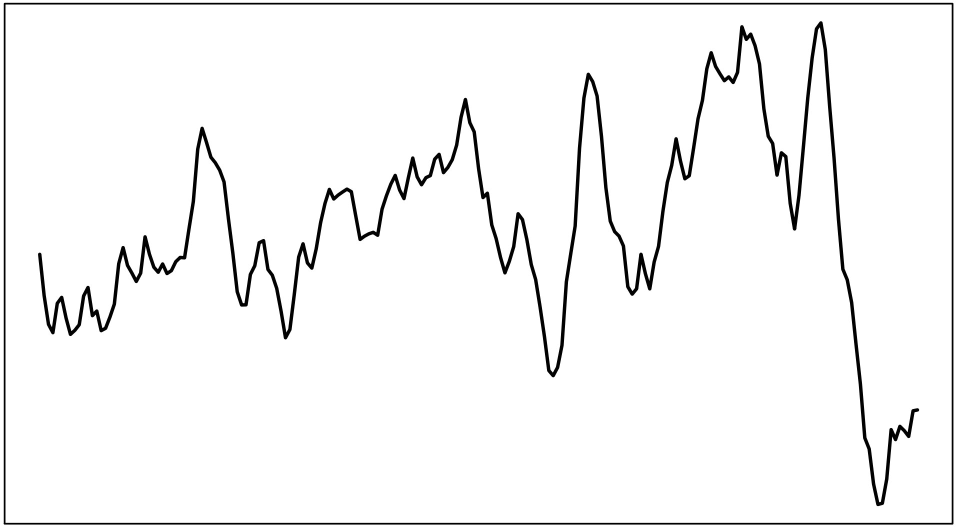
Many time series models use stationarity as an initial assumption.

1 Generated from a stationary process:



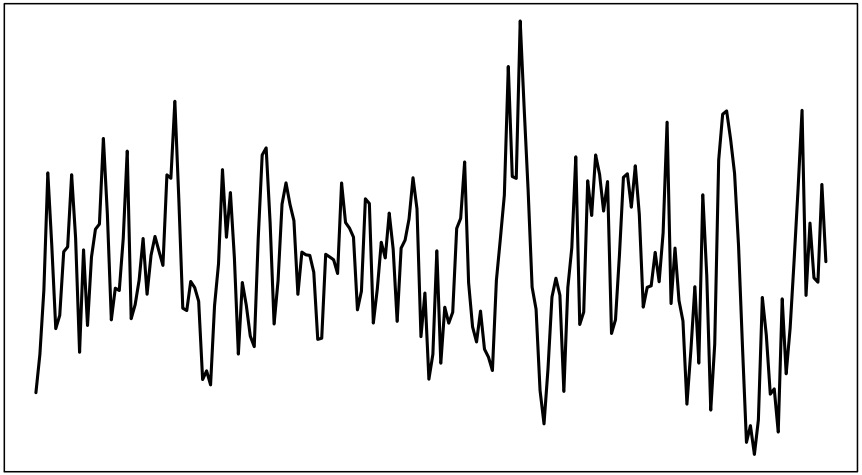
2

1 Generated from a nonstationary process:

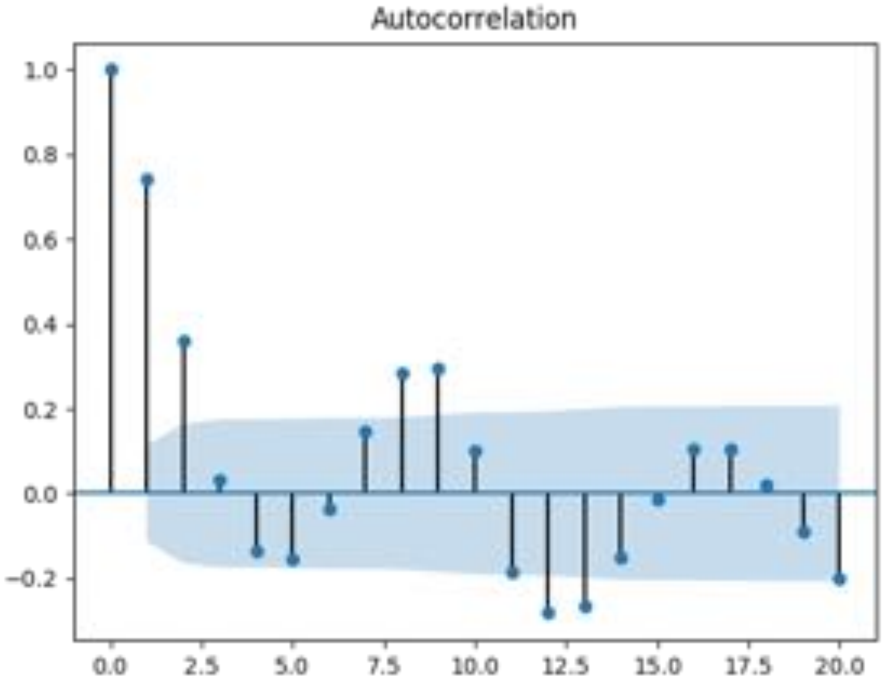


2

The same series, first-differenced, i.e., $Y_t = X_t - X_{t-1}$. Such operations are often used to transform a series which is clearly nonstationary into one where stationarity is a realistic approximation.



It is common to inspect a plot of the autocorrelation function (ACF) of a time series to get a sense of the correlation at different lags.



This notion of modelling the dependence of X_t on the preceding observations $X_{t-1}, X_{t-2}, \ldots$ in a hallmark of standard time domain models.

We will next briefly show some of the standard models, prior to considering the more general framework provided by the state space representation.

The AR(p) Models

The autoregressive (AR) models are a simple class of time series models which take the form

$$Y_t = \mu + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + \epsilon_t$$

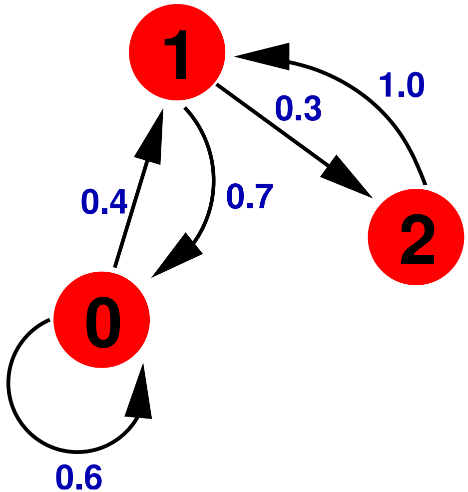
where ϵ_t is assumed to be Gaussian with mean zero and variance σ^2.

As the name implies, the value of Y_t is fit by regressing on the p previous values in the series.

The value of p is referred to as the order of the model. It is typically chosen using a criterion such as the Bayesian Information Criterion (BIC).

Markov Chain Models

The figure above depicts a stochastic model which moves between three states (the circles) labeled 0, 1, and 2.



The numbers attached to the arrows show the probabilities of moving from one state to the next.

This figure shows a (first order) Markov chain with a finite state space, $\{0, 1, 2\}$.

If we let X_t denote the state at timestep t, then the sequence $X_0, X_1, \ldots$ becomes one of simplest time series models.

In general, the Markov property is used to describe any process whose future behavior depends only on the current state. This is also referred to as the memoryless property.

The random walks that make up Markov Chain Monte Carlo (MCMC) are examples of Markov chains, typically with a continuous state space.

From Cordes (*ApJ*, 2013), "Pulsar State Switching from Markov Transitions and Stochastic Resonance":

> Along with binary states, some pulsars show multiple discrete values of subpulse drift rate or quasi-periods in pulse amplitudes... The changes in drift rate or quasi-period, often combined with nulling, imply that some objects display at least four states. These are also easily describable as Markov chains ...

Transition matrix for four state model from Cordes (2013). There are two "null" states and two "burst" states:

$$\boldsymbol{Q}_{(n_1,n_2,b_1,b_2)} \approx \begin{pmatrix} 0.01 & 0 & 0.886 & 0.104 \\ 0 & 0.940 & 0.006 & 0.054 \\ 0.469 & 0.521 & 0.01 & 0 \\ 0.012 & 0.100 & 0 & 0.887 \end{pmatrix}. \quad (9)$$

Hidden Markov Models

A useful extension of the Markov chain model is the hidden Markov model. Here, there is an underlying (hidden or “latent”) Markov chain that determines the distribution for the observable data.

Useful when there is some underlying (unobservable) process, and the observations are (noisy) functions of the current state of that process.

The State Space Representation

The state space representation of a time series model for $\{\mathbf{Y}_t\}$ is given by a pair of equations:

The state equation is

$$\boldsymbol{\alpha}_t = \mathbf{T}_t \boldsymbol{\alpha}_{t-1} + \mathbf{c}_t + \mathbf{R}_t \boldsymbol{\eta}_t$$

where

- $\boldsymbol{\alpha}_t$ is the ℓ-dimensional state vector
- $\mathbf{T}_t$ is a ℓ by ℓ transition matrix
- $\mathbf{c}_t$ is the state intercept
- $\mathbf{R}_t$ is the selection matrix
- $\boldsymbol{\eta}_t$ is Gaussian with mean zero and covariance $\mathbf{Q}_t$

The states are unobserved. The observation equation is

$$\mathbf{Y}_t = \mathbf{Z}_t \boldsymbol{\alpha}_t + \mathbf{d}_t + \boldsymbol{\epsilon}_t$$

where

- $\mathbf{Y}_t$ is the k-dimensional observable
- $\mathbf{d}_t$ is the observation intercept
- $\mathbf{Z}_t$ is the design matrix
- $\boldsymbol{\epsilon}_t$ is Gaussian with mean zero and covariance $\mathbf{H}_t$

It is further assumed that $\boldsymbol{\eta}_t$ and $\boldsymbol{\epsilon}_t$ are uncorrelated.

It is often the case that $\mathbf{T}_t$, $\mathbf{R}_t$, $\mathbf{c}_t$, $\mathbf{Q}_t$, $\mathbf{Z}_t$, $\mathbf{d}_t$, and/or $\mathbf{H}_t$ do not vary with t, and the t subscript is dropped.

The state vector at time t is often thought to be "hidden." The state equation is

$$\boldsymbol{\alpha}_t = \mathbf{T}_t \boldsymbol{\alpha}_{t-1} + \mathbf{c}_t + \mathbf{R}_t \boldsymbol{\eta}_t$$

so the state is updated at each time step, modelled to be a linear combination of the previous states, plus error.

$\boldsymbol{\alpha}_t$ can be taken to be the "ideal" observable, while the actual observable, $\mathbf{Y}_t$ is related to the current state via the observation equation:

$$\mathbf{Y}_t = \mathbf{Z}_t \boldsymbol{\alpha}_t + \mathbf{d}_t + \boldsymbol{\epsilon}_t$$

The power of this model is its generality: A wide range of models fit into this framework.

Technical Comment: Specifying the Initial Distribution

To fit these models (typically done via maximum likelihood), one needs to specify an initial distribution for the state vector. It is typically assumed to be Gaussian with some specified mean and covariance.

In some cases this distribution will follow naturally from model assumptions (e.g., stationarity), while in others it is just assumed that the mean is a vector filled with zeros or ones, and the covariance matrix is diagonal with very large entries on the diagonal.

Example: The AR(p) Models

The autoregressive (AR) models are a simple class of time series models which take the form

$$Y_t = \mu + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + \epsilon_t$$

where ϵ_t is assumed to be Gaussian with mean zero and variance σ^2.

As the name implies, the value of Y_t is fit by regressing on the p previous values in the series.

The value of p is referred to as the order of the model.

The AR(p) models can be placed into state space form.

For example, for the AR(2) model we can write the transitions as

$$\begin{bmatrix} Y_t \\ Y_{t-1} \end{bmatrix} = \begin{bmatrix} \phi_1 & \phi_2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} Y_{t-1} \\ Y_{t-2} \end{bmatrix} + \begin{bmatrix} \mu \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} \sigma^2 \end{bmatrix}$$

and we then observe at time t

$$\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} Y_t \\ Y_{t-1} \end{bmatrix}$$

The AR(p) models can be placed into state space form.

For example, for the AR(2) model we can write the transitions as

$$\begin{bmatrix} Y_t \\ Y_{t-1} \end{bmatrix} = \begin{bmatrix} \phi_1 & \phi_2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} Y_{t-1} \\ Y_{t-2} \end{bmatrix} + \begin{bmatrix} \mu \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} \sigma^2 \end{bmatrix}$$

and we then observe at time t

$$\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} Y_t \\ Y_{t-1} \end{bmatrix}$$

So, we are in state space form with

$$\boldsymbol{\alpha}_t = \begin{bmatrix} Y_t \\ Y_{t-1} \end{bmatrix}, \ \mathbf{T} = \begin{bmatrix} \phi_1 & \phi_2 \\ 1 & 0 \end{bmatrix}, \ \mathbf{c} = \begin{bmatrix} \mu \\ 0 \end{bmatrix}, \ \mathbf{R} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \ \mathbf{Q} = \begin{bmatrix} \sigma^2 \end{bmatrix}$$

$$\mathbf{Z} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \ \mathbf{d} = \mathbf{0}, \ \mathbf{H} = \mathbf{0}$$

Suppose I wanted to incorporate an additional predictor X into the model, i.e., consider a model of the form

$$Y_t = \mu + \beta X_t + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \epsilon_t$$

I could simply update my prior state space form to use

$$\mathbf{c}_t = \begin{bmatrix} \mu + \beta X_t \\ 0 \end{bmatrix}$$

Again, the value here is in the flexibility in the form of the models.

The Kalman Filter

A key appeal of the state space formulation of a time series model is the ease with which standard inference goals can be achieved via a set of recursive algorithms referred to as the Kalman Filter.

Keep in mind that within this framework it is assumed that the state vector $\boldsymbol{\alpha}_t$ is of primary interest.

We want to consider three fundamental challenges:

- **Filtering:** Estimate $\boldsymbol{\alpha}_t$ using $\mathbf{Y}_0, \mathbf{Y}_1, \ldots, \mathbf{Y}_t$.
- **Prediction:** Forecast $\boldsymbol{\alpha}_t$ using $\mathbf{Y}_0, \mathbf{Y}_1, \ldots, \mathbf{Y}_{t-1}$.
- **Smoothing:** Estimate $\boldsymbol{\alpha}_t$ using $\mathbf{Y}_0, \mathbf{Y}_1, \ldots, \mathbf{Y}_n$ for $t < n$.

Tsay (2010) makes the following analogy to reading a handwritten note:

> Filtering is figuring out the word you are reading based on knowledge accumulated from the beginning of the note, predicting is to guess the next word, and smoothing is deciphering a particular word once you have read through the note. (page 561)

Technical Aside: See a standard text such as Brockwell and Davis (2016) for derivations. I am going to simplify the situation by assuming $\mathbf{c}_t = \mathbf{d}_t = \mathbf{0}$, $\mathbf{R}_t = \mathbf{I}$, and $\mathbf{Q}$, $\mathbf{H}$, $\mathbf{Z}$, and $\mathbf{T}$ do not vary with t.

The following notation is needed:

- Let $P_t(\alpha_i)$ denote the best linear predictor of state α_i given $\mathbf{Y}_0$, $\mathbf{Y}_1, \ldots, \mathbf{Y}_t$, for $1 \leq i \leq \ell$. The entries are pulled together into a ℓ by one vector $P_t(\boldsymbol{\alpha})$.

- Let $\widehat{\boldsymbol{\alpha}}_t$ denote $P_{t-1}(\boldsymbol{\alpha}_t)$, the forecast of $\boldsymbol{\alpha}_t$ based on the information available at time $t-1$.

- Let Ω_t be the error covariance matrix for the prediction at time t,

$$\Omega_t = E[(\boldsymbol{\alpha}_t - \widehat{\boldsymbol{\alpha}}_t)(\boldsymbol{\alpha}_t - \widehat{\boldsymbol{\alpha}}_t)'].$$

The Prediction Problem

The forecasts for the next values of the state vector are given by

$$\widehat{\boldsymbol{\alpha}}_{t+1} = \mathbf{T}\widehat{\boldsymbol{\alpha}}_t + \Theta_t \Delta_t^{-1}(\mathbf{Y}_t - \mathbf{Z}\widehat{\boldsymbol{\alpha}}_t)$$

with covariance matrices

$$\Omega_{t+1} = \mathbf{T}\Omega_t\mathbf{T}' + \mathbf{Q} - \Theta_t\Delta_t^{-1}\Theta_t'$$

where

$$\Delta_t = \mathbf{Z}\Omega_t\mathbf{Z}' + \mathbf{H} \quad \text{and} \quad \Theta_t = \mathbf{T}\Omega_t\mathbf{T}'$$

The Filtering and Smoothing Problems

The filtered estimates are

$$P_t(\boldsymbol{\alpha}_t) = \widehat{\boldsymbol{\alpha}}_t + \Omega_t \mathbf{Z}' \Delta_t^{-1}(\mathbf{Y}_t - \mathbf{Z}\widehat{\boldsymbol{\alpha}}_t)$$

The smoothed estimates are

$$P_n(\boldsymbol{\alpha}_t) = P_{n-1}(\boldsymbol{\alpha}_t) + \Omega_{t,n} \mathbf{Z}' \Delta_n^{-1}(\mathbf{Y}_n - \mathbf{Z}\widehat{\boldsymbol{\alpha}}_n)$$

Each of these has appropriately messy expressions for deriving the covariance matrix of the estimates.

Example: Local Trends Model

We consider a simple model for a univariate time series $\{Y_t\}$ called a local trends model:

$$Y_t = \mu_t + \epsilon_t$$

where

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

ϵ_t is Gaussian with mean zero and variance σ_ϵ^2 and η_t is normal with mean zero and variance σ_η^2.

This model can be easily placed into state space form with

- the "state" at time t (α_t) is μ_t;
- the "transition matrix" $(\mathbf{T})$ is simply the scalar ϕ;
- the "design matrix" ($\mathbf{Z}$) is identity;
- The covariance matrix for the state equation error term ($\mathbf{Q}$) consists solely of σ^2_η.
- The covariance matrix for the observation equation error term ($\mathbf{H}$) consists solely of σ^2_ϵ.

This model is very simple, but it displays an appealing property of the state space approach: It is possible to separate observational error (the ϵ_i) from the fluctuations that model natural changes in the process (the η_i).

Since measurement error is abundant in astronomical data analysis problems, one could take advantage of this to uncover true underlying structure.

Let's consider an example from finance with a similar structure.

The log daily return of an asset can be decomposed

$$\text{log return for day } j = \sum_{i=1}^{m} r_{ij}$$

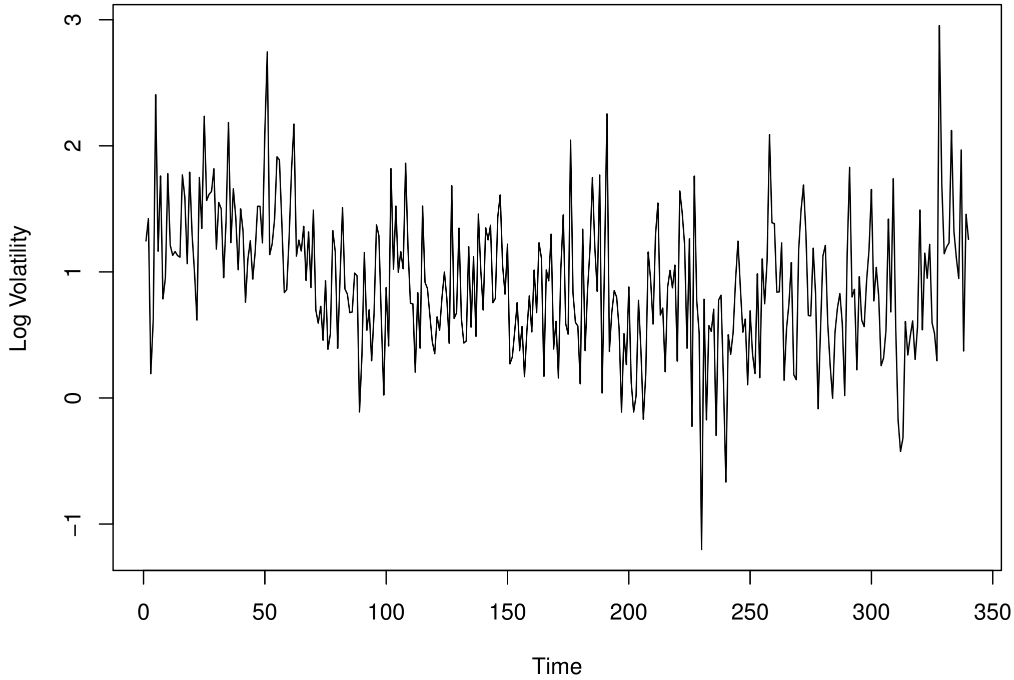
where r_{ij} is the log return over a small time interval, e.g., ten minutes.

Then, the realized volatility is

$$\text{realized volatility for day } j = \sum_{i=1}^{m} r_{ij}^2$$

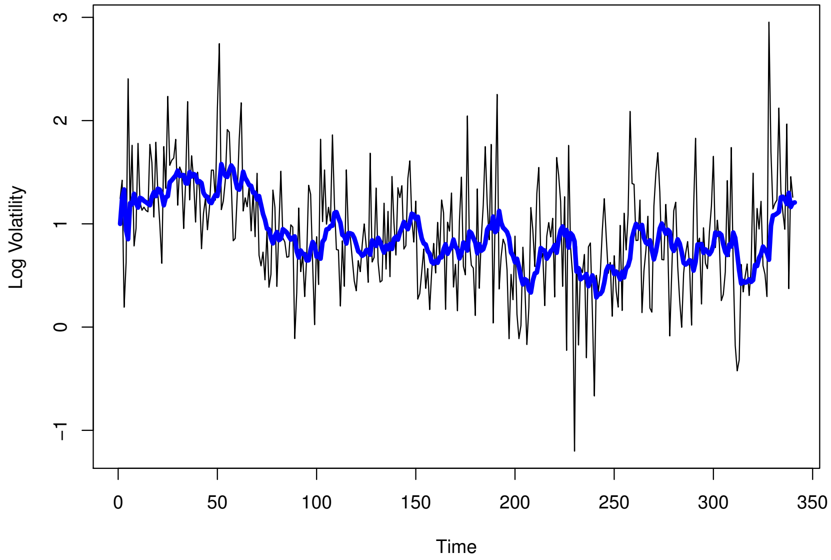
The distribution of the log of the realized volatility is reasonably approximated by a Gaussian, with mean equal to the log volatility of the asset. We will hence employ the local trends model above, with μ_t equal to the (unobservable) log volatility on day t and Y_t is the realized volatility on day t.

The data we have (from Tsay) are for Alcoa, from January 2, 2003 to May 7, 2004. Ten minute time intervals were used in this data set.

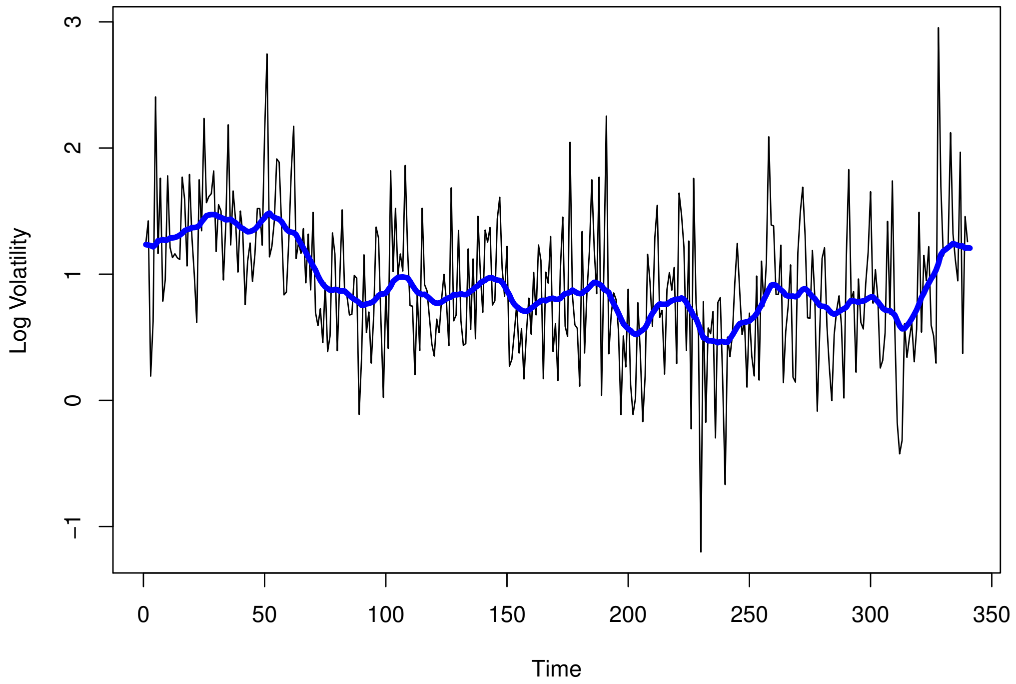


The parameter estimates are $\widehat{\sigma}_{\epsilon}^2 = 0.23$, $\widehat{\sigma}_{\eta}^2 = 0.0057$, and $\widehat{\phi} = 0.99$.

The Kalman filter was applied, and the figure below compares the observed series and the filtered states.



1 Comparison of the observed series and the smoothed states.



2

Diagnostics

It is also possible to obtain forecasts for the next observation of the series.

These allow for the calculations of residuals as the differences between these forecasts and the actual observations.

One useful step is to look at the ACF of the residuals to determine if there is remaining serial correlation.

Example: Regression with Time-Varying Coefficients

State space representations create a natural way of having regression models with time varying coefficients.

Consider the following model:

$$\begin{aligned} Y_t &= a_t + b_t X_t + \epsilon_t \\ a_t &= a_{t-1} + \eta_t \\ b_t &= b_{t-1} + \tau_t \end{aligned}$$

where ϵ_t, η_t, and τ_t are assumed to be Gaussian with mean zero and respective variances σ^2_ϵ, σ^2_η, and σ^2_τ.

This model can be placed into state space form using

$$\boldsymbol{\alpha}_t = \begin{bmatrix} a_t \\ b_t \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} \sigma_\eta^2 & 0 \\ 0 & \sigma_\tau^2 \end{bmatrix}, \quad \mathbf{T} = \mathbf{R} = \mathbf{I}$$

$$\mathbf{Z}_t = \begin{bmatrix} 1 & X_t \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \sigma_\epsilon^2 \end{bmatrix}$$

Note that in this case the matrix $\mathbf{Z}_t$ does depend on t.

Implementation

Next, we will learn how to implement these techniques in Python...

References

Brockwell and Davis (2016) *Introduction to Time Series and Forecasting*, Third Edition, Springer.

Cordes (2013) *Astrophysical Journal*, Volume 775, Number 1, Page 47.

Tsay (2010) *Analysis of Financial Time Series*, Third Edition, Wiley.

Wong, et al. (2016) *Annals of Applied Statistics*, Volume 10, Number 2, Page 1107.