

# Kalman Filter

Author: Mattia Albertini

## Introduction

In this document, I develop and implement algorithm to apply Kalman Filtering: a statistical tool to extract core processes or “signals” from a noisy time series.

Here I consider a specific state space model, however, given the algorithm and full understanding of the literature, the algorithm can be re-adapted easily. So it can be used as a starting point.

DISCLAIMER: The theoretical background is a personal review of “Time Series Analysis” by James D. Hamilton.

For more informations about Kalman filtering visit this link.

## The Theory

The algorithm works in a two-step process. In the prediction step, the Kalman filter produces estimates of the current state variables. Once the outcome of the next measurement (with some amount of error, including random noise) is observed, these estimates are updated using a weighted average, with more weight being given to estimates with higher certainty. The algorithm is recursive. Optimality of the Kalman filter assumes that the errors are Gaussian, therefore here I assume gaussian distribute errors.

Let  $y_{1:t}$  are a collection of observations of variable  $y_t$  which follows some process. Now,

- Suppose we believe process is driven by some other latent process  $W_t$  according to a linear relationship. This is called the “space equation” e.g.

$$y_t = F(W_t) + \varepsilon_t \quad \text{with } F(\cdot) \text{ linear}$$

- Suppose further that the prediction error  $\varepsilon_t$  is gaussian white noise.
- Suppose that we know the what type of process  $W_t$  follows, this is called the “state equation”

Then, Kalman Filtering allows us to derive information on this latent process (or in jargon “extract the signal”). In particular, we are able to derive,

$$\begin{aligned}\widehat{W}_{t|t} &= E[W_t|y_{1:t}] \\ \widehat{P}_{t|t} &= V[W_t|y_{1:t}]\end{aligned}$$

Where “hat” variables are estimate of the mean and variance of the latent process.

In this particular framework suppose the space is,

$$y_t = 2 \cdot W_t + u_t$$

Further suppose the state is,

$$W_t = 0.5W_{t-1} + \varepsilon_t$$

Then initial conditions are the mean and variance of an unconditional AR1:

$$W_{0|0} = E[W_0] = \frac{\phi_0}{1 - \phi_1} = 0 \quad \text{and} \quad P_{0|0} = \frac{\sigma_\varepsilon^2}{1 - \phi_1^2} = \frac{1}{1 - 0.5^2}$$

## The algorithm

The algorithm goes as follows,

```
KalmanFilter <- function(Y, W_00, P_00, Q, R, f, H, t) {  
  # initialize Vector  
  W =c(W_00)  
  P =c(P_00)  
  # Compute W_tt and P_tt  
  for (i in 1:t) {  
    mu1 <- f**W[i]  
    mu2 <- t(H) ** mu1  
    S_11 <- f^2 ** P[i] + Q  
    S_22 <- t(H)**S_11**H + R  
    S_12 <- S_11**H  
    S_21 <- t(S_12)  
    K <- S_12 ** solve(S_22)  
    eta <- matrix(c(Y[i]), nrow=1, ncol=1) - mu2  
    W_tt <- mu1 + K ** eta  
    P_tt <- S_11 - K ** t(H) ** S_11  
    # Update Vectors  
    W = append(W, W_tt)  
    P = append(P, P_tt)  
  }  
  # Prepare Results  
  results <-as.data.frame(cbind(W[2:t], P[2:t]))  
  KF <-< results  
}
```

That is, given the initial conditions, it starts a recursion in which mean and variance are updates iteration by iteration under by doing best linear predictions. For more on the theory and understanding of all parameters refer to Time Series Analysis, James D. Hamilton (chp 13, p. 372).

Note that: - Y is the data; - W\_00 and P\_00 are initial conditions for  $W_t$ ; - H is the coefficient of the space equation, i.e. 1; - Q, R are variance covariance matrices respectively of the state and of the space - f is  $\phi_1$  - t is the number of recursions.

The parameters H, Q, R, f can be fully understood from Time Serie Analysis, James D. Hamilton (chp 13, p. 372).

## Example

To see it in practice, given the unobserved (latent) process W,

```
Wt <- arima.sim(list(order=c(1,0,0), ar=.5), n=100)
```

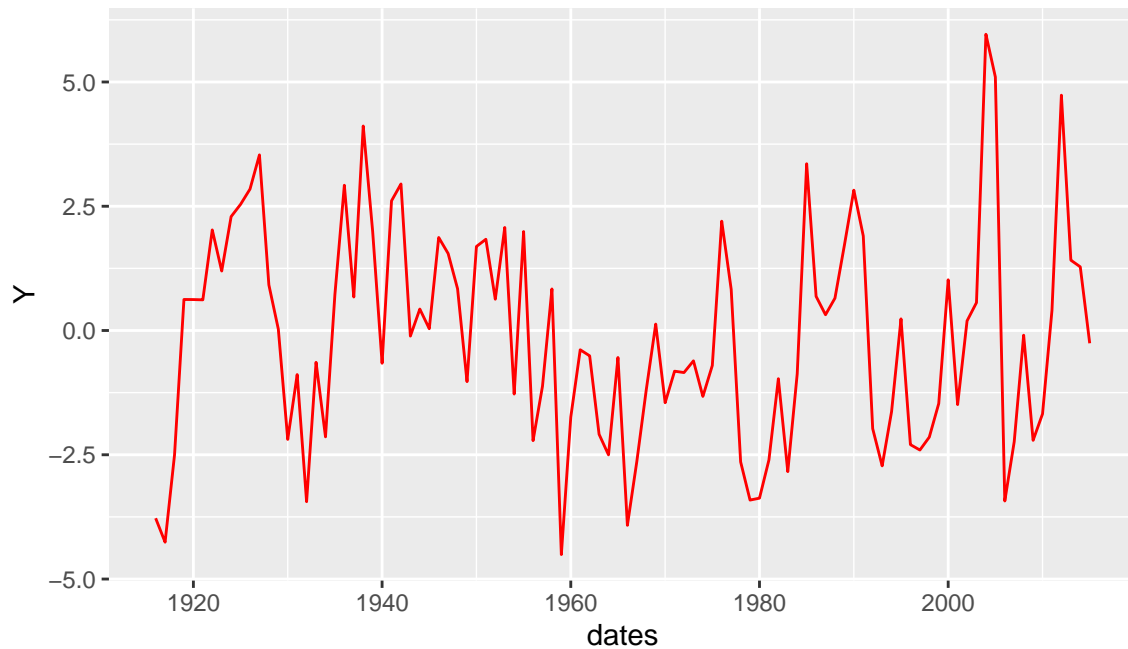
Then Y, equals the latent plus normal noise,

```
Yt <- 2*Wt+rnorm(100)
```

To get a feeling of  $Y_t$ ,

```
ggplot() +  
  geom_line(data = Yt, aes(x = dates, y = Y), color = "red")
```

```
## Don't know how to automatically pick scale for object of type <ts>. Defaulting  
## to continuous.
```

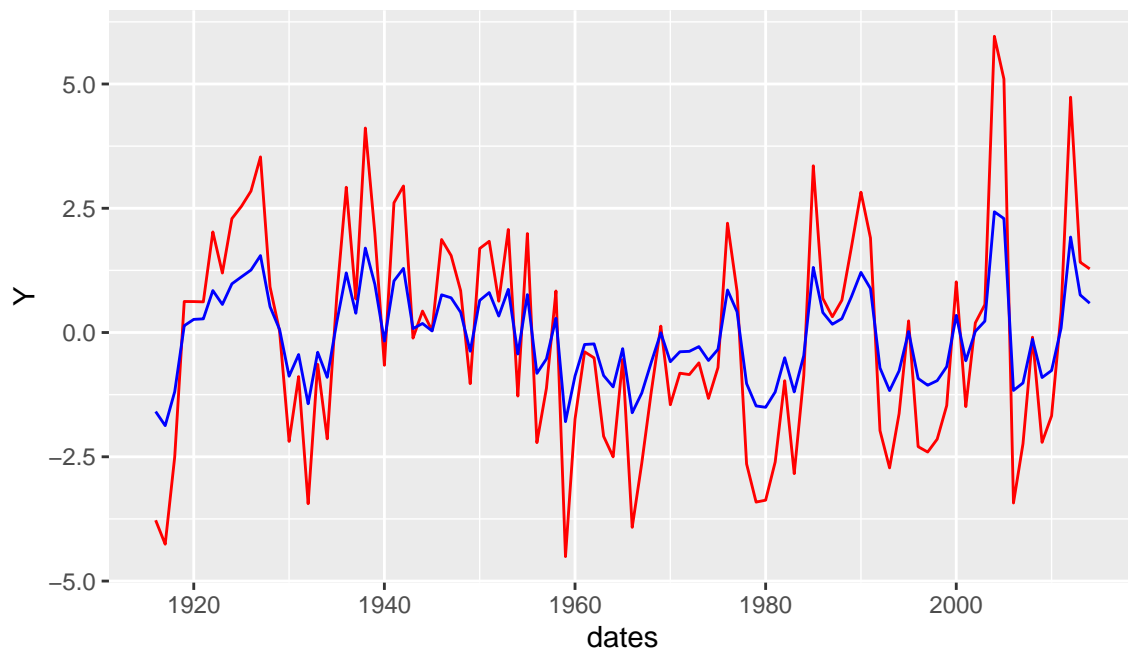


Apply the filter,

```
KalmanFilter(Y=Yt$Y, W_00=0, P_00=1/(1-0.5^2), Q=1, R=1, f=0.5, H=2, t=100)
```

Plot  $Y_t$  and  $\hat{W}_{t|t}$ ,

```
ggplot() +  
  geom_line(data = mydata, aes(x = dates, y = Y), color = "red") +  
  geom_line(data = mydata, aes(x = dates, y = V1), color = "blue")
```



Where the blue process is our estimated signal. As by definition, indeed  $Y$  was driven by the process of  $W$  plus noise.