Kalman_Filter_Econometrics

Sarah-Katharina Umek

2022-11-14

Model:

We have a Gaussian approximation of the distribution of ε_t involving the first two moments ($\gamma = -\Gamma'(1)$ is the Euler Mascheroni constant):

$$\mathbb{E}(\varepsilon_t) = \mu_\varepsilon = -\gamma - log(2) \approx -1.27036, \quad Var(\varepsilon_t) = \frac{\pi^2}{2}$$

yields a representation as a local level model for h_t,

$$h_t = h_{t-1} + \eta_t, \quad \eta_t \sim N(0, \theta)$$
$$y_t = h_t + \varepsilon_t, \quad N(0, \frac{\pi^2}{2})$$

involving the transformed outcome varibale $y_t = log(r_t^2) - \mu_\varepsilon$.

Using the propagation, prediction and correction equations we can obtain the following Klaman filter:

$$K_t = (P_{t-1|t-1} + \theta) * C_{t|t-1}^{-1}$$

Applying this model to financial returns of our choice to estimate the volatility h_t.

$$\mathbb{E}(\sigma_t^2|\sigma_{t-1}^2) = \sigma_{t-1}^2\mathbb{E}(e_t^\eta) = \sigma_{t-1}^2 e^{\theta/2} \approx \sigma_{t-1}^2(1 + \theta/2)$$

```
# using monthly data as that makes the charts more readable.
getSymbols("GOOG", from = "2015-10-30", to = "2022-10-30", warnings = FALSE,
    auto.assign = TRUE, periodicity = "monthly")
```

```
## [1] "GOOG"
```

```
google <- Cl(na.omit(GOOG))
google <- na.omit(diff(log(google)))
google_sqrt_r <- google^2
```

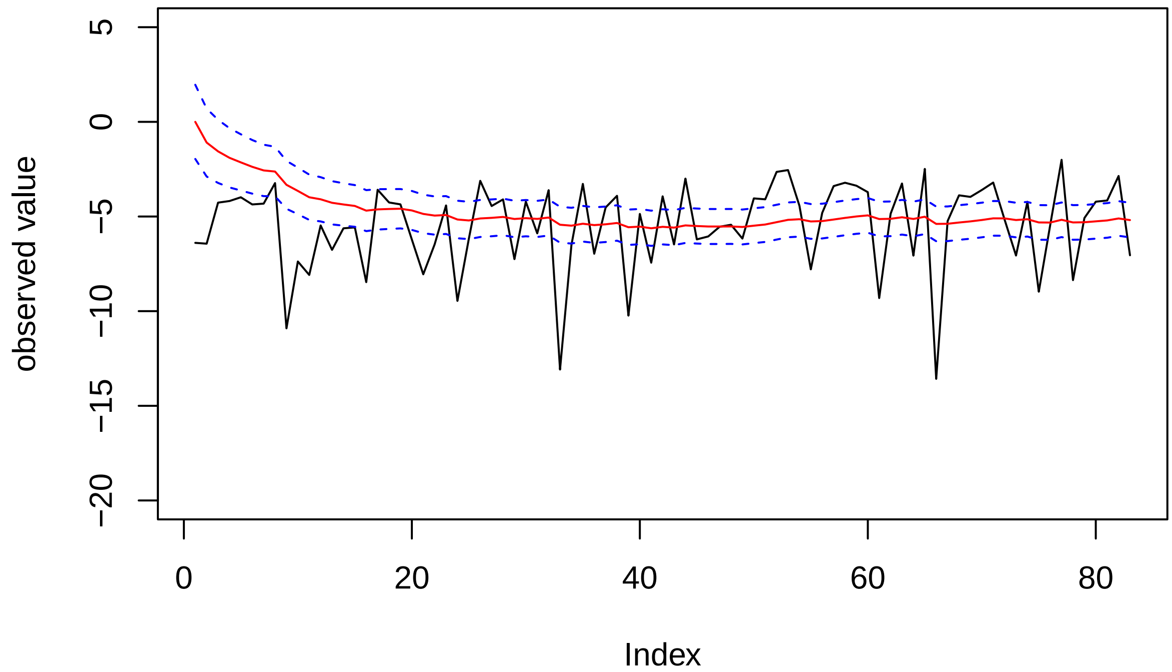
```
# We need the Euler-Mascheroni constant $\gamma$
gamma <- -digamma(1)

mu_epsilon <- (-gamma - log(2))
mu_epsilon
```

```
## [1] -1.270363
```

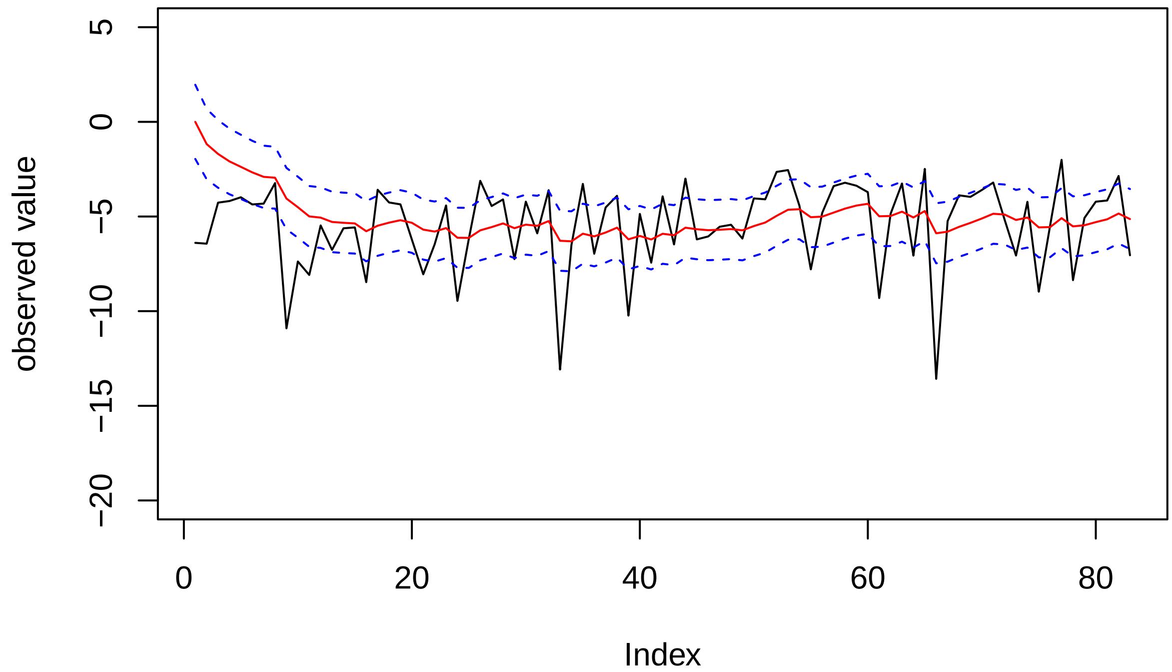
Recall, transformed variable outcome $y_t = log(r_t^2) - \mu_\varepsilon$

```
y_t <- log(google_sqrt_r) - mu_epsilon

# setting up the initial distribution
h_0 <- 0  #mean
P_0 <- 1  #variance

# Kalman filter function

kalman_filter <- function(h0, P0, theta, yt) {
    # prediction vectors
    h_t <- rep(h0, length(yt))
    P_t <- rep(P0, length(yt))
    # Kalman filter equations:
    for (i in 2:length(yt)) {
        # Propagation
        ht_tminus1 <- h_t[i - 1]
        Pt_tminus1 <- P_t[i - 1] + theta
        # Prediction
        yt_tminus1 <- ht_tminus1
        Ct_tminus1 <- Pt_tminus1 + (pi^2/2)
        # Correction
        Kt <- Pt_tminus1 * (Ct_tminus1)^(-1)
        ht_t <- ht_tminus1 + Kt * (yt[i] - yt_tminus1)
        Pt_t <- (1 - Kt) * Pt_tminus1

        # save the calculated values
        h_t[i] <- ht_t
        P_t[i] <- Pt_t
    }
    return(list(`ht = ` = h_t, `Pt =` = P_t))
}
```

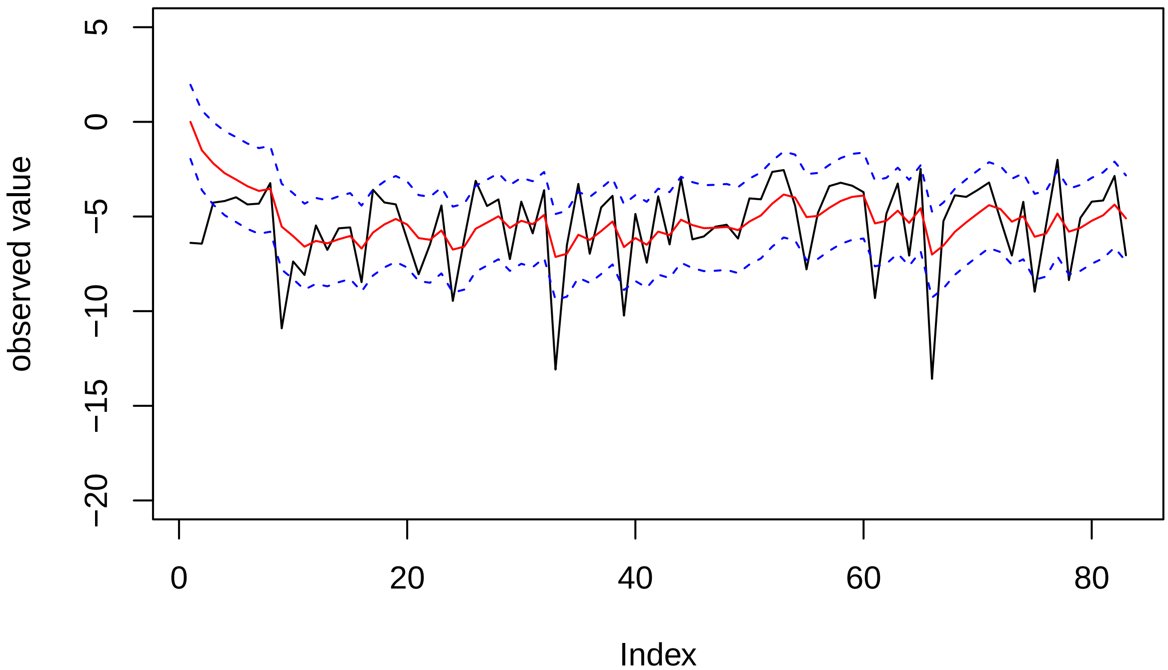
```
# testing using a theta of 0.01
test_theta_01 <- kalman_filter(h_0, P_0, 0.01, y_t)


sd1_theta_01 <- test_theta_01$`ht = ` + 1.96 * sqrt(test_theta_01$`Pt =`)
sd2_theta_01 <- test_theta_01$`ht = ` - 1.96 * sqrt(test_theta_01$`Pt =`)

plot(as.numeric(y_t), type = "l", ylab = "observed value", ylim = c(-20,
    5))
lines(test_theta_01$`ht = `, type = "l", col = "red")
lines(as.numeric(sd1_theta_01), type = "l", lty = 2, col = "blue")
lines(as.numeric(sd2_theta_01), type = "l", lty = 2, col = "blue")
```




```r
# testing using a theta of 0.1

test_theta_01 <- kalman_filter(h_0, P_0, 0.1, y_t)


sd1_theta_01 <- test_theta_01$`ht = ` + 1.96 * sqrt(test_theta_01$`Pt =`)
sd2_theta_01 <- test_theta_01$`ht = ` - 1.96 * sqrt(test_theta_01$`Pt =`)

plot(as.numeric(y_t), type = "l", ylab = "observed value", ylim = c(-20,
    5))
lines(test_theta_01$`ht = `, type = "l", col = "red")
lines(as.numeric(sd1_theta_01), type = "l", lty = 2, col = "blue")
lines(as.numeric(sd2_theta_01), type = "l", lty = 2, col = "blue")
```




```r
# testing using a theta of 0.5

test_theta_01 <- kalman_filter(h_0, P_0, 0.5, y_t)


sd1_theta_01 <- test_theta_01$`ht = ` + 1.96 * sqrt(test_theta_01$`Pt =`)
sd2_theta_01 <- test_theta_01$`ht = ` - 1.96 * sqrt(test_theta_01$`Pt =`)

plot(as.numeric(y_t), type = "l", ylab = "observed value", ylim = c(-20,
    5))
lines(test_theta_01$`ht = `, type = "l", col = "red")
lines(as.numeric(sd1_theta_01), type = "l", lty = 2, col = "blue")
lines(as.numeric(sd2_theta_01), type = "l", lty = 2, col = "blue")
```



5
0
−5
−10
−15
−20
observed value
0
20
40
60
80
Index