Time series Lab 3

Omkar Bhutra (omkbh878)

12 October 2019

Assignment 1:

In table 1 a script for generation of data from simulation o fthe following state space model and implementation of the Kalman filter on the data is given. Kalman filter: $z_t = A_{t-1}z_{t-1} + e_t$, $x_t = C_t z_t + \upsilon_t$, $\upsilon_t\ N(0, R_t)$, $e_t\ N(0, Q_t)$

Prediction step: $m_{t+1|t} = A_t m_{t|t}$ $P_{t+1|t} = A_t P_{t|t} A_t^T + Q_{t+1}$

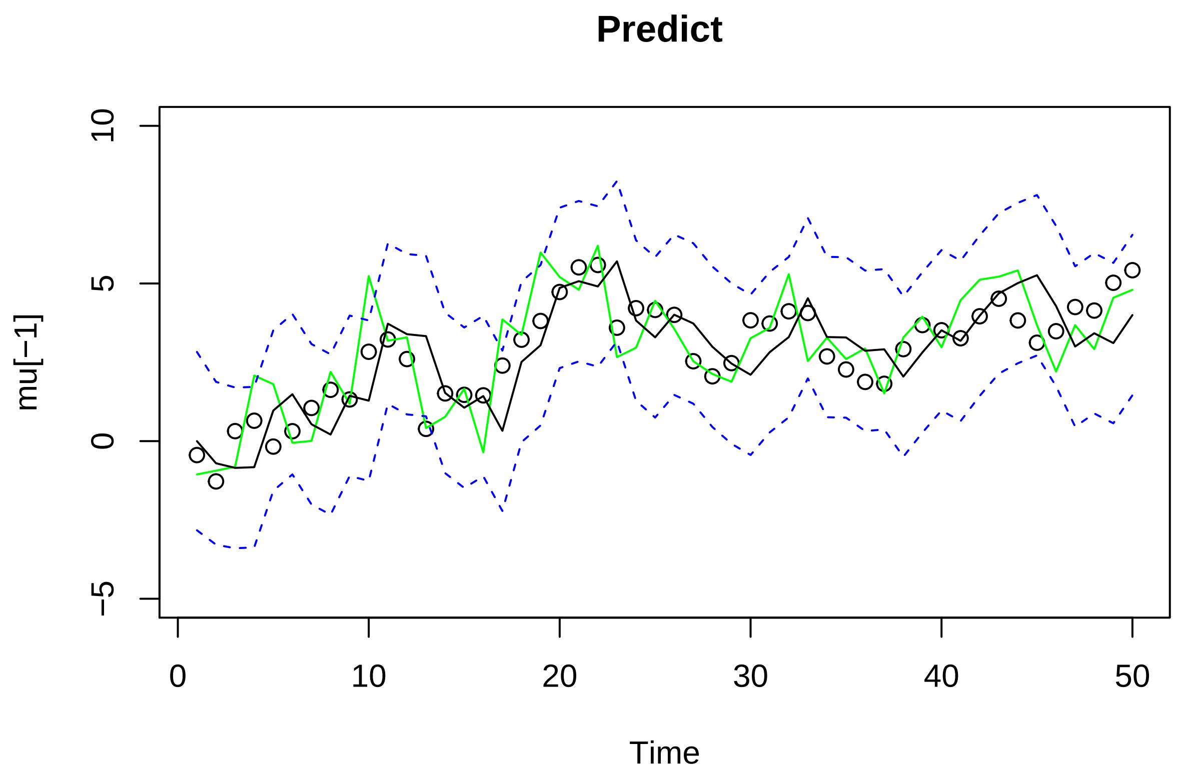
Observation Update: $G_t = \frac{P_t C^T}{AP_tA^T+R} = \frac{P_{t|t}A_t^T}{P_{t+1|t}}$ $\tilde{m}_{t|t} = m_t + G_t(z_{t+1} - Am_{t|t})$ $\tilde{P}_t = P_{t|t} - G_t(AP_{t|t}A^T + Q)G_t^T = P_{t|t} - G_t P_{t+1|t} G_t^T$

a. Write down the expression for the state space model that is being simulated.

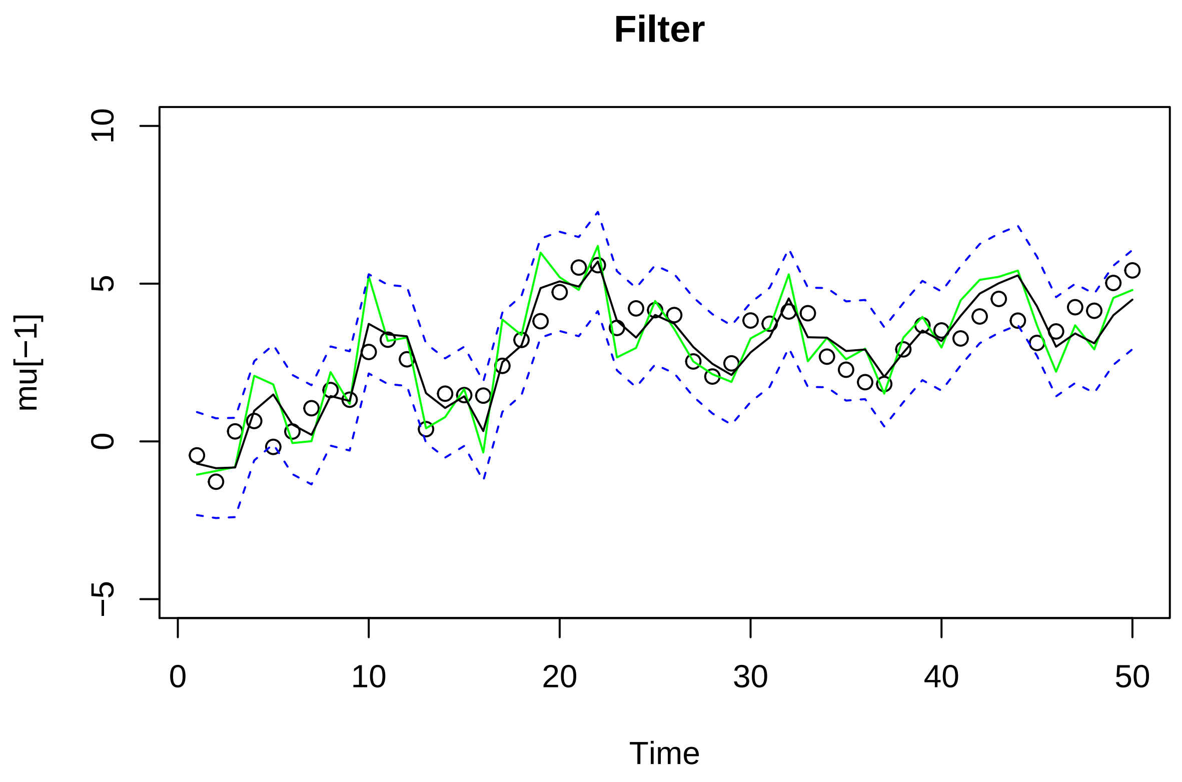
observation update using $x_t = C_t z_t + \upsilon_t - > N(z_t; m_{t|t}, P_{t|t}$ prediction using $z_{t+1} = Az_t + e_{t+1} - > N(z_t; m_{t+1|t}, P_{t+1|t})$ $\theta = \{A, C, R, Q, m_0, P_0\} = \{1, 1, 1, 1, 0, 1\}$

b. Run this scrip and compare the filtering results with a moving average smoother of order 5.

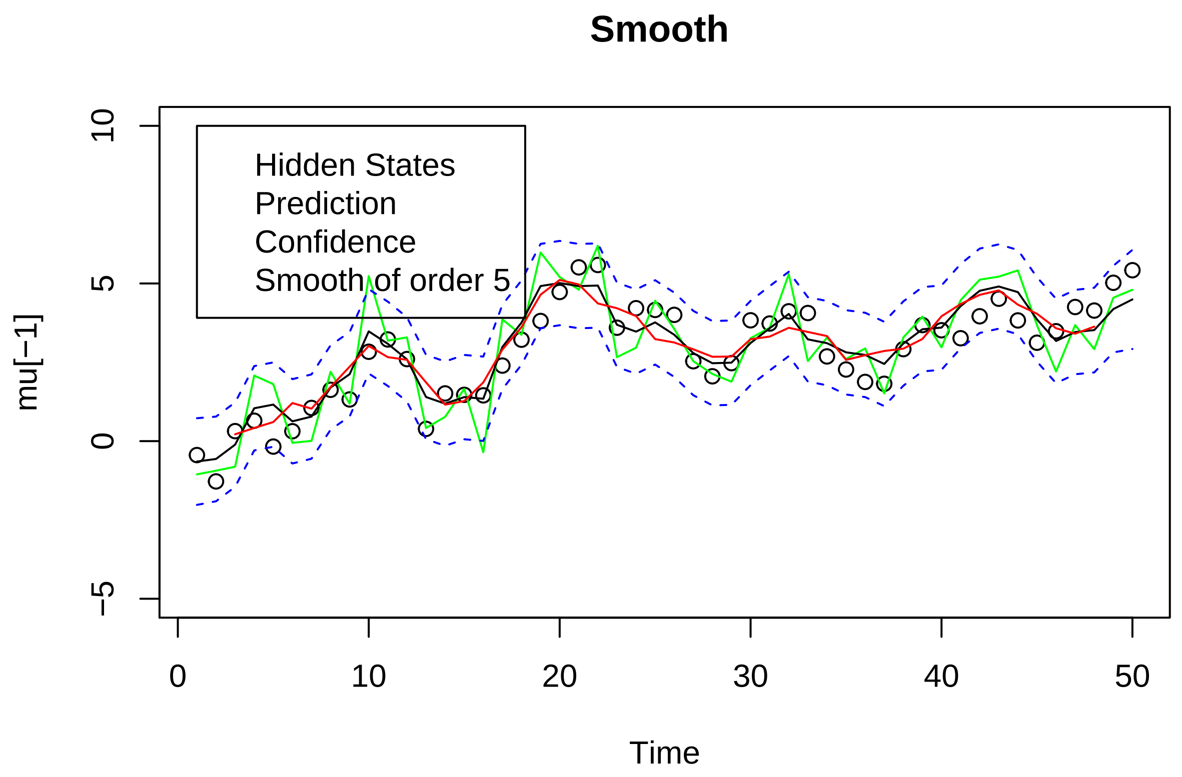
```r
# generate data
set.seed(1); num = 50

smoother<-function(x,n){
  stats::filter(as.vector(x),rep(1/n,n),sides = 2)
}

w = rnorm(num+1,0,1); v = rnorm(num ,0,1)
mu = cumsum(w) # state : mu[0], mu[1] ,... , mu[50]
y = mu[-1] + v # obs: y[1] ,... , y[50]
# filter and smooth ( Ksmooth 0 does both )
ks = Ksmooth0(num , y, A=1, mu0=0, Sigma0=1, Phi=1, cQ=1, cR=1)
# start figure
Time = 1:num
plot(Time , mu[-1], main ='Predict ', ylim =c(-5,10))
lines(Time ,y,col=" green ")
lines(ks$xp)
lines(ks$xp+2* sqrt(ks$Pp), lty =2, col=4)
lines(ks$xp -2* sqrt(ks$Pp), lty =2, col=4)
```



```
plot(Time , mu[-1], main ='Filter ', ylim =c(-5,10))
lines(Time ,y,col=" green ")
lines(ks$xf)
lines(ks$xf+2* sqrt(ks$Pf), lty =2, col=4)
lines(ks$xf -2* sqrt(ks$Pf), lty =2, col=4)
```



```
plot(Time , mu[-1], main ='Smooth ', ylim =c(-5,10))
lines(Time ,y,col=" green ")
lines(ks$xs)
lines(ks$xs+2* sqrt(ks$Ps), lty =2, col=4)
lines(ks$xs -2* sqrt(ks$Ps), lty =2, col=4)
lines(Time,smoother(y,5),col = "red") # Moving average smoother order 5
legend(1, 10, legend = c("Hidden States", "Prediction",
 "Confidence", "Smooth of order 5"), col = c("black", "green",
 "blue", "red"))
```



```
mu[1]; #initial mean smoother
```

```
## [1] -0.6264538
```

```
ks$x0n; #initial smoother covariance
```

```
##            [,1]
## [1,] -0.3241541
```

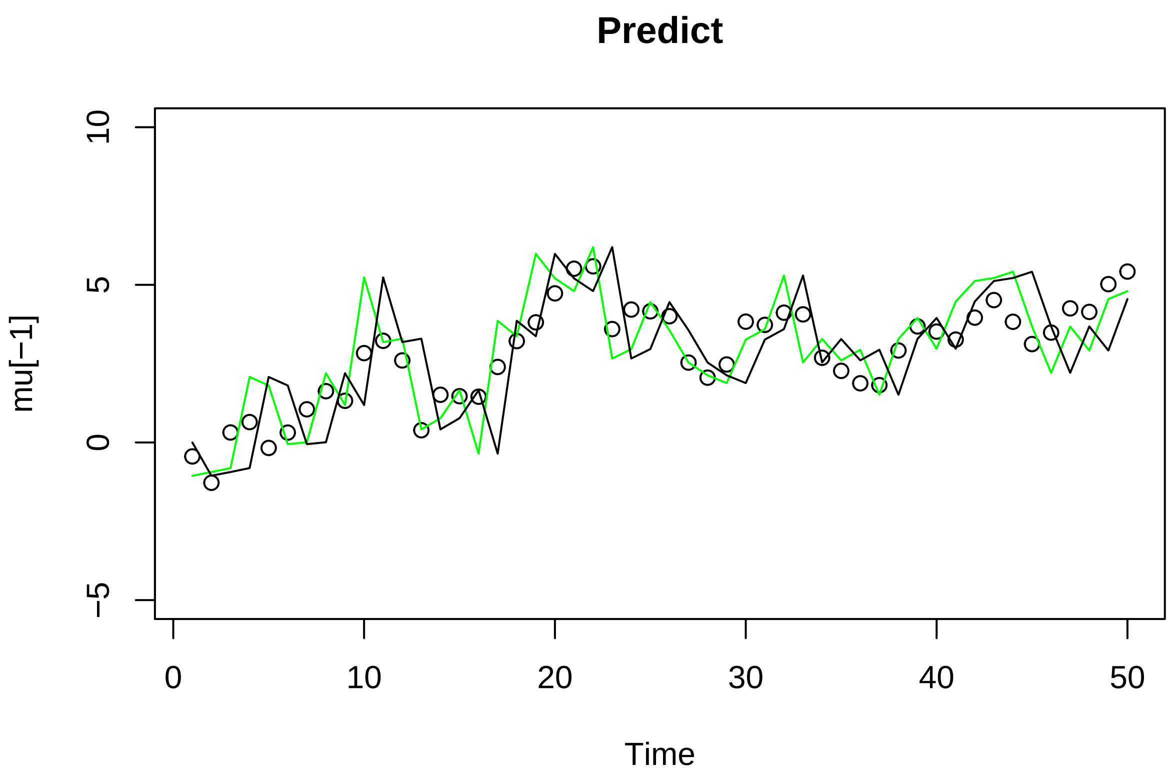
```
sqrt(ks$P0n) # initial value info
```

```
##           [,1]
## [1,] 0.7861514
```

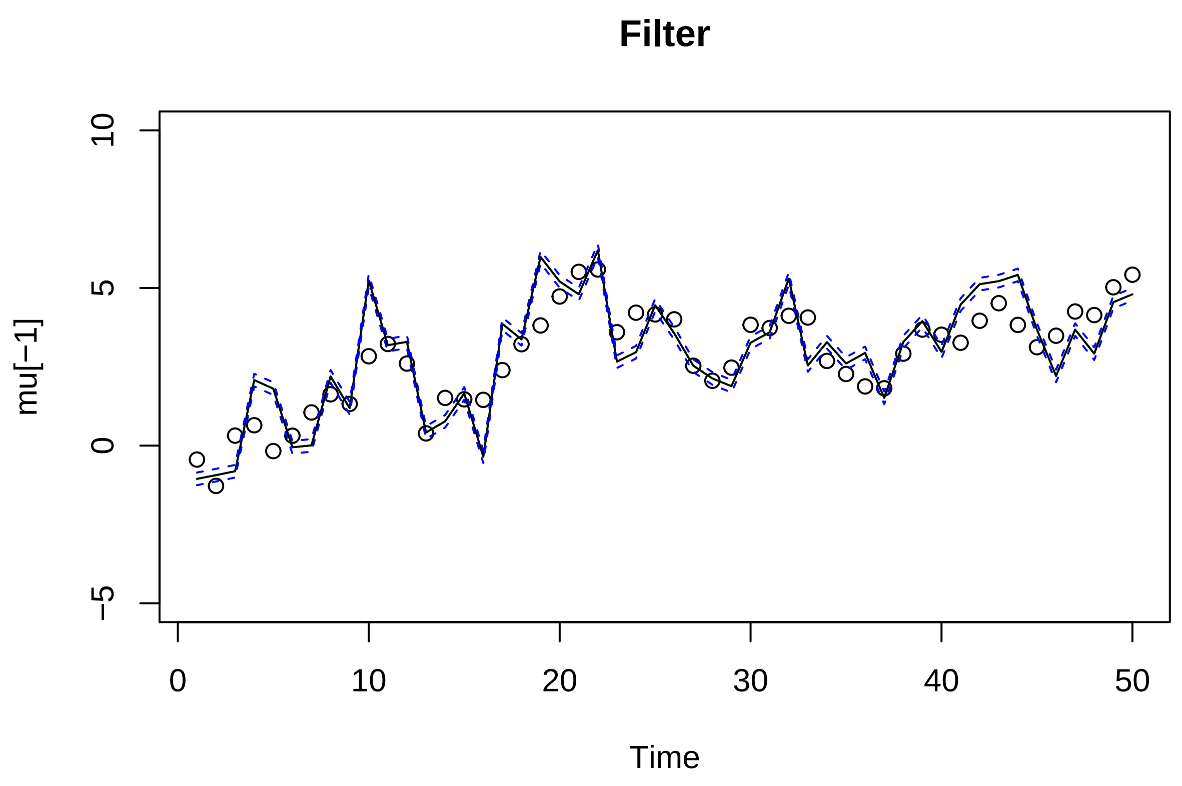
It can be visually observed that the prediction is most volatile but stays within the confidence intervals. Filtering reduces the volatility and the moving average smoothing of order 5 which is seen as the red line is the smoothest of all the functions.

c)Also, compare the filtering outcome when R in the filter is 10 times smaller than its actual value while Q in the filter is 10 times larger than its actual value. How does the filtering outcome vary?

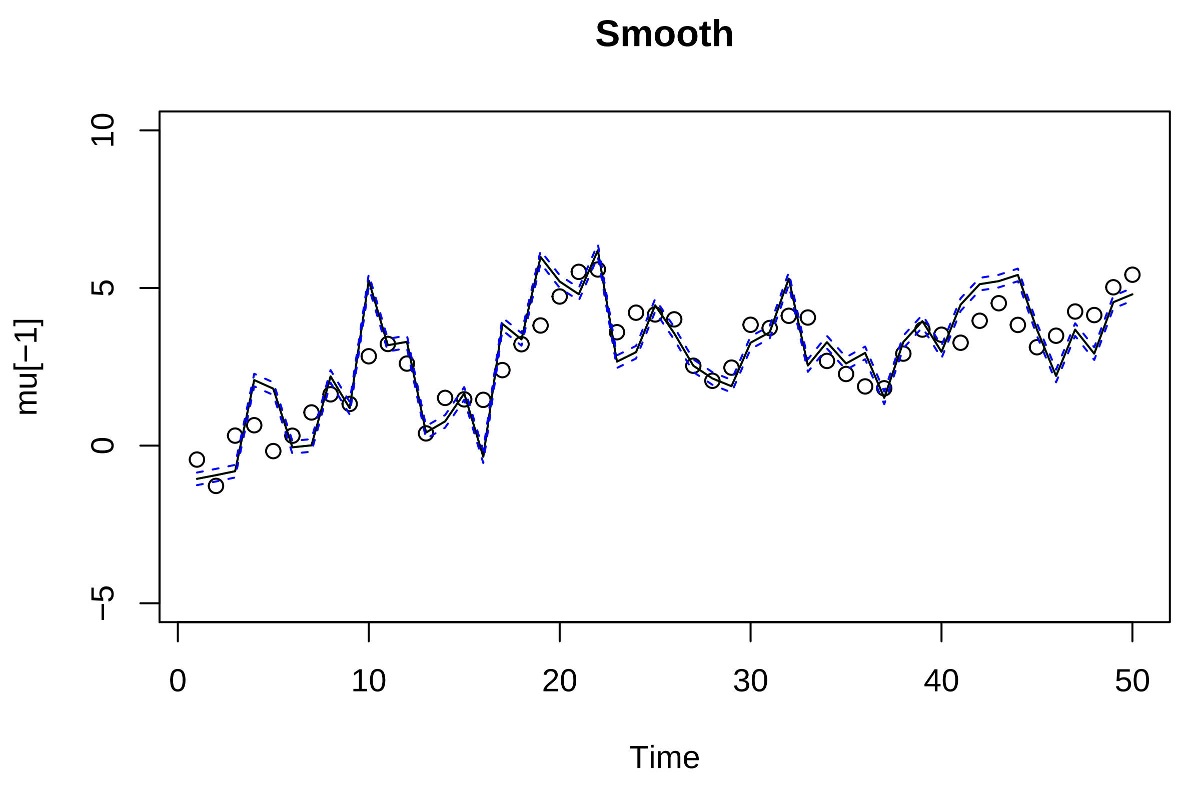
```
set.seed(1)
num =50
w = rnorm(num+1,0,1)
v = rnorm(num ,0,1)
mu = cumsum(w) # state : mu[0] , mu[1] ,... , mu[50]
y = mu[-1] + v # obs: y[1] ,... , y[50]
# filter and smooth ( Ksmooth 0 does both )
ks = Ksmooth0(num,y,A=1, mu0=0, Sigma0=1, Phi=1, cQ=10, cR =.1)
# start figure
par(mfrow=c(1,1))
Time = 1: num
# predicted
plot(Time , mu[-1] , main='Predict',ylim=c(-5,10))
lines(Time,y,col="green")
lines(ks$xp)
lines(ks$xp+2*sqrt (ks$Pp) , lty =2, col=4)
lines(ks$xp-2*sqrt(ks$Pp) ,lty=2,col=4)
```



```
# Filter
plot(Time,mu[ -1] , main ='Filter',ylim=c(-5,10))
lines(Time ,y,col="green")
lines(ks$xf )
lines(ks$xf+2*sqrt(ks$Pf) , lty =2, col=4)
lines(ks$xf-2*sqrt(ks$Pf) , lty =2, col=4)
```



```
# Smooth
plot(Time , mu[-1] , main ='Smooth', ylim =c( -5,10))
lines(Time ,y ,col="green")
lines(ks$xs )
lines(ks$xs+2*sqrt(ks$Ps) , lty =2, col=4)
lines(ks$xs-2*sqrt(ks$Ps) , lty =2, col=4)
```



```
#Initial mean smoother
mu[1]
```

```
## [1] -0.6264538
```

```
#Initial smoother covariance
ks$x0n
```

```
##             [,1]
## [1,] -0.01044278
```

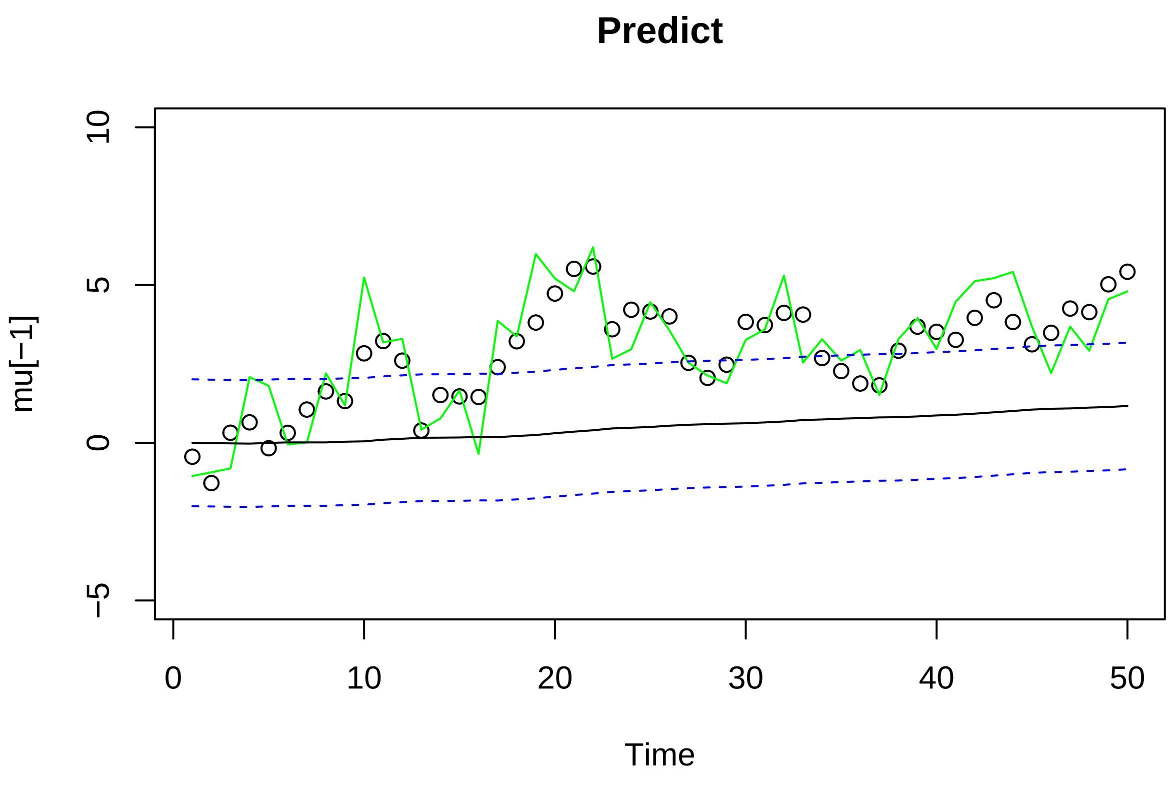
```
#Initial value info
sqrt(ks$P0n)
```

```
##           [,1]
## [1,] 0.9950377
```

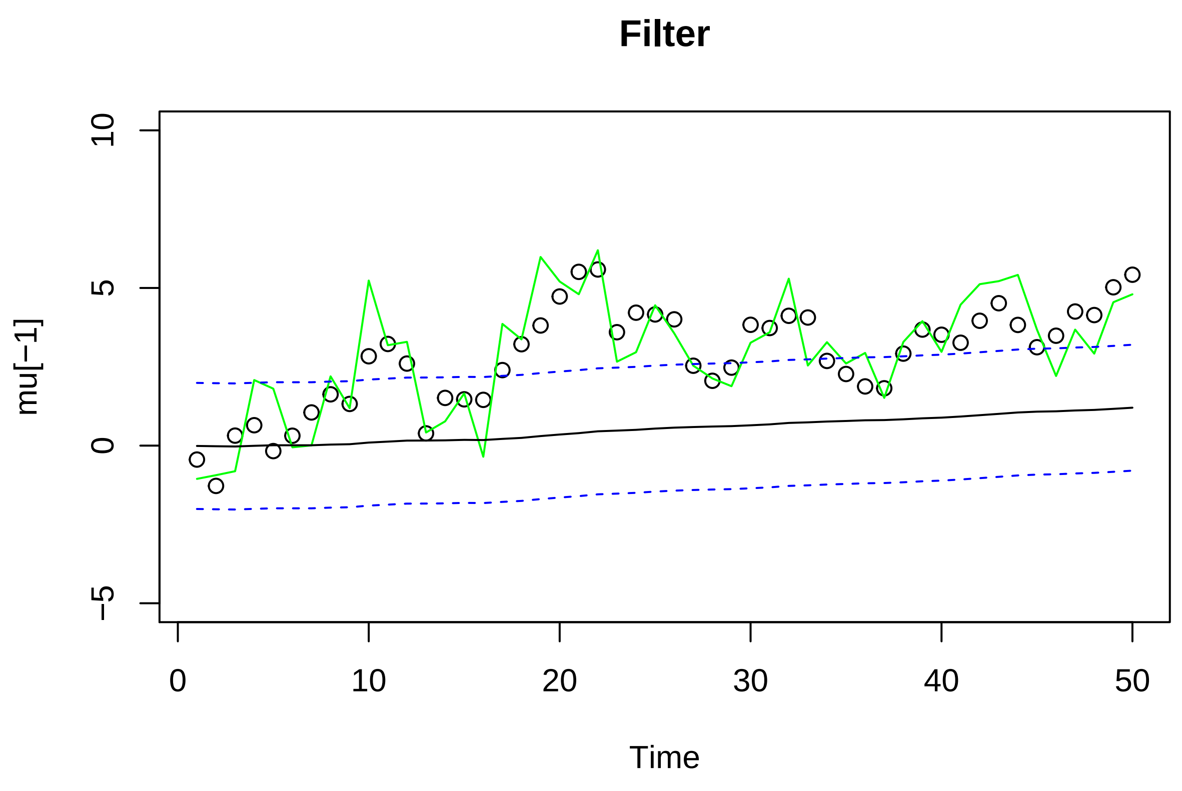
The data and filtering model as in 1b is taken but the R in the filter is 10 times smaller and Q in the filter is larger by 10 times, both compared to their actual values. The filtering and smoothed lines are closely similar.

d. Now compare the filtering outcome when R in the filter is 10 times larger than its actual value while Q in the filter is 10 times smaller than its actual value. How does the filtering outcome varies?

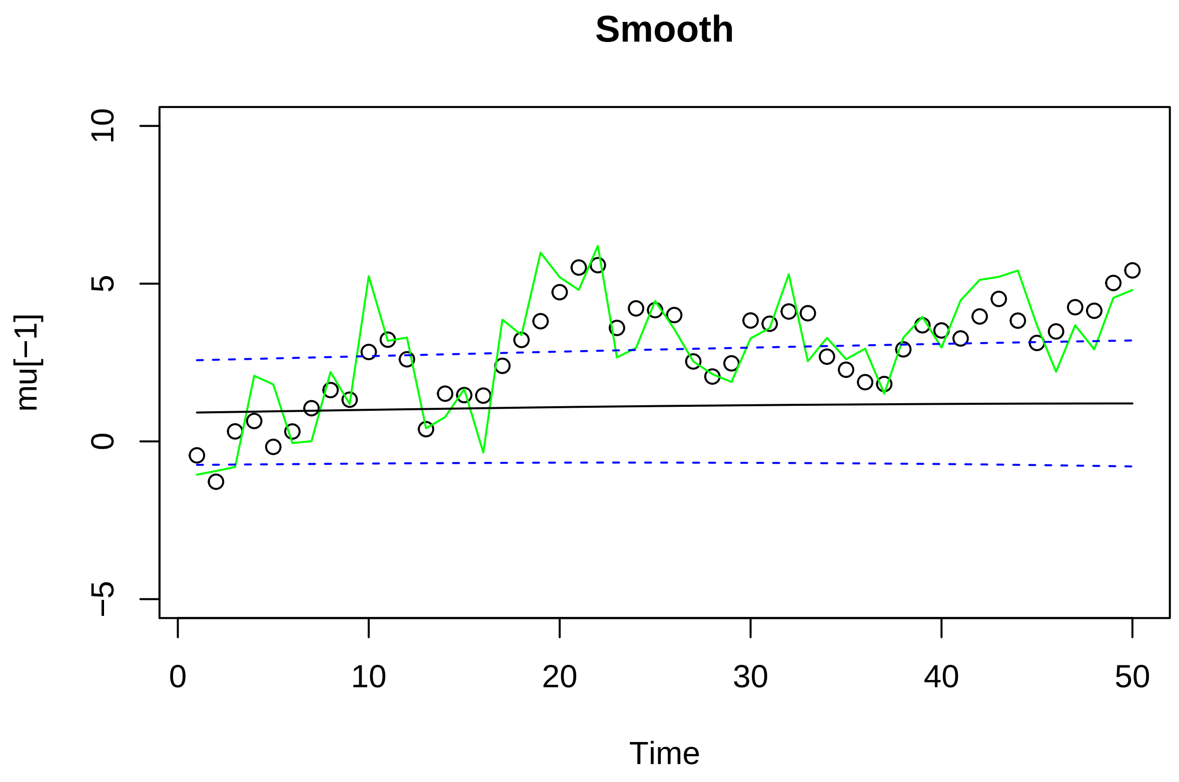
```r
set.seed(1)
num =50
w = rnorm(num+1,0,1)
v = rnorm(num ,0,1)
mu = cumsum(w) # state : mu[0] , mu[1] ,... , mu[50]
y = mu[-1] + v # obs: y[1] ,... , y[50]
# filter and smooth ( Ksmooth 0 does both )
ks = Ksmooth0(num,y,A=1, mu0=0, Sigma0=1, Phi=1, cQ=0.10, cR =10)
# start figure
par(mfrow=c(1,1))
Time = 1: num
# predicted
plot(Time , mu[-1] , main='Predict',ylim=c(-5,10))
lines(Time,y,col="green")
lines(ks$xp)
lines(ks$xp+2*sqrt (ks$Pp) , lty =2, col=4)
lines(ks$xp-2*sqrt(ks$Pp) ,lty=2,col=4)
```



```r
# Filter
plot(Time,mu[ -1] , main ='Filter',ylim=c(-5,10))
lines(Time ,y,col="green")
lines(ks$xf )
lines(ks$xf+2*sqrt(ks$Pf) , lty =2, col=4)
lines(ks$xf-2*sqrt(ks$Pf) , lty =2, col=4)
```



```
# Smooth
plot(Time , mu[-1] , main ='Smooth', ylim =c( -5,10))
lines(Time ,y ,col="green")
lines(ks$xs )
lines(ks$xs+2*sqrt(ks$Ps) , lty =2, col=4)
lines(ks$xs-2*sqrt(ks$Ps) , lty =2, col=4)
```



```r
#Initial mean smoother
mu[1]
```

```
## [1] -0.6264538
```

```r
#Initial smoother covariance
ks$x0n
```

```
##          [,1]
## [1,] 0.905755
```

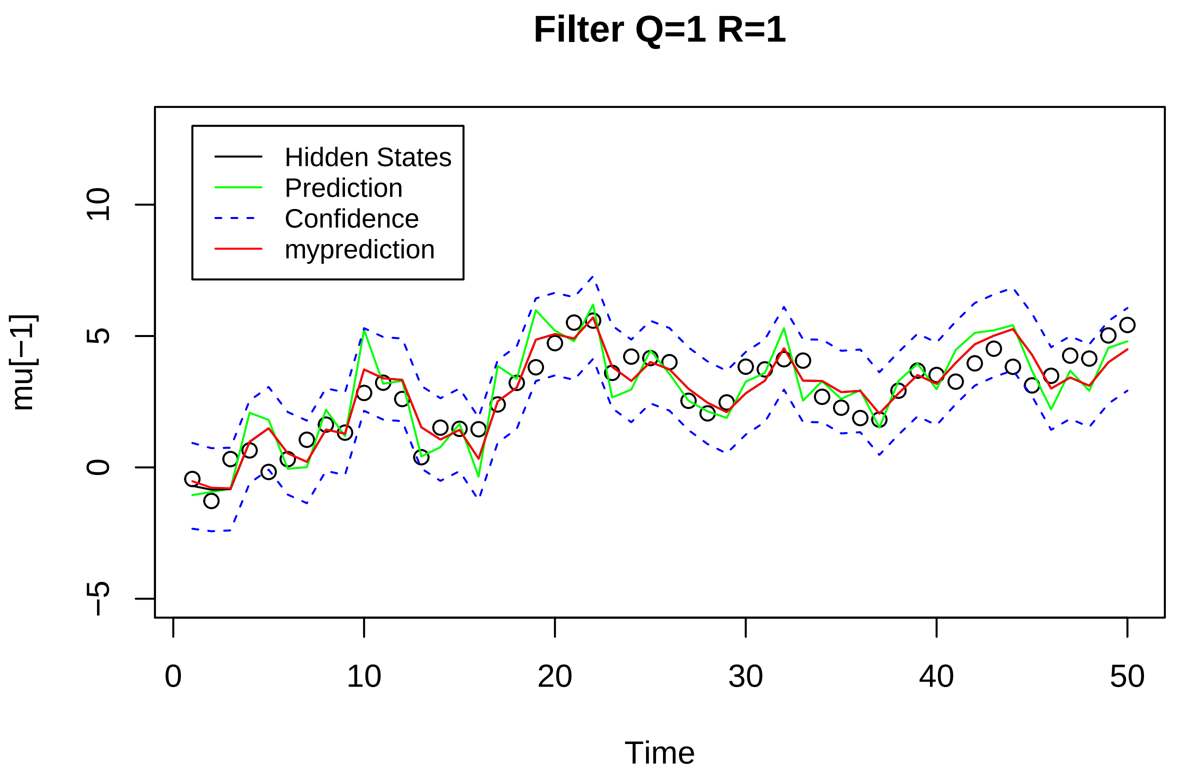
```r
#Initial value info
sqrt(ks$P0n)
```

```
##         [,1]
## [1,] 0.82731
```

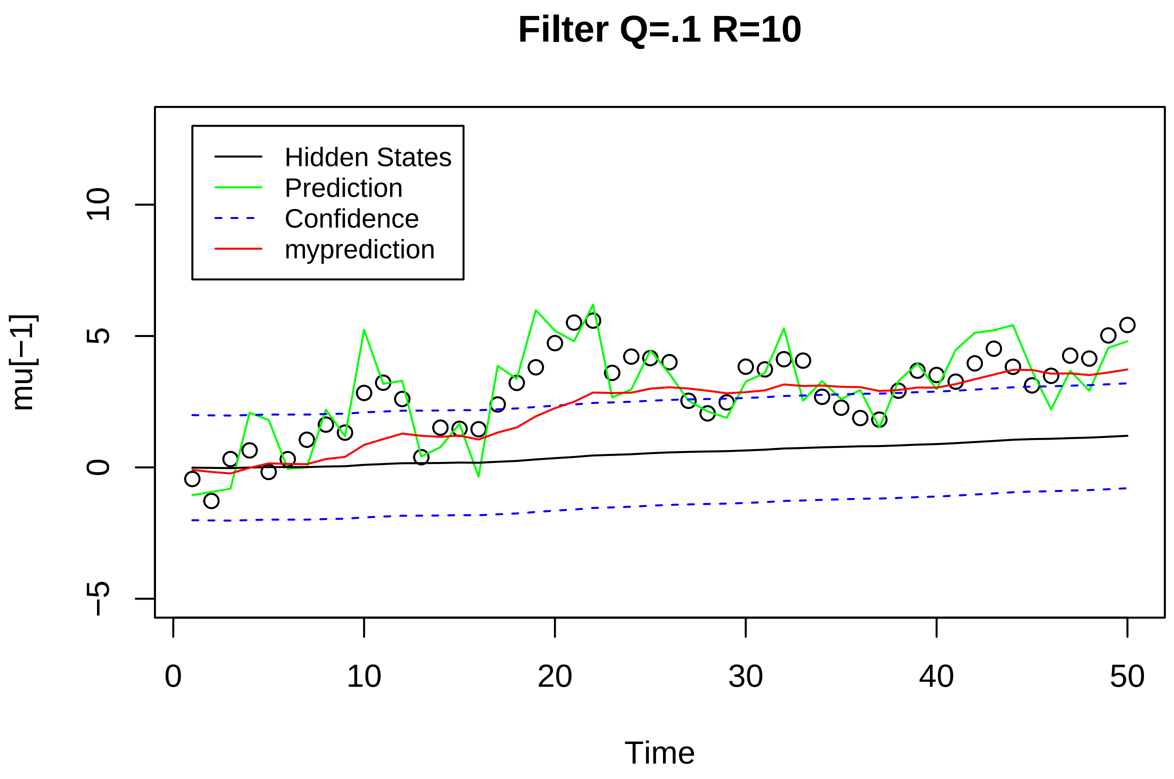
The data and filtering model as in 1b is taken but the Q in the filter is 10 times smaller and R in the filter is larger by 10 times, both compared to their actual values. The model considers data as error when the distance from the observations to the line are smaller than 10.

1e. Implement your own Kalman filter and replace ksmooth0 function with your script.

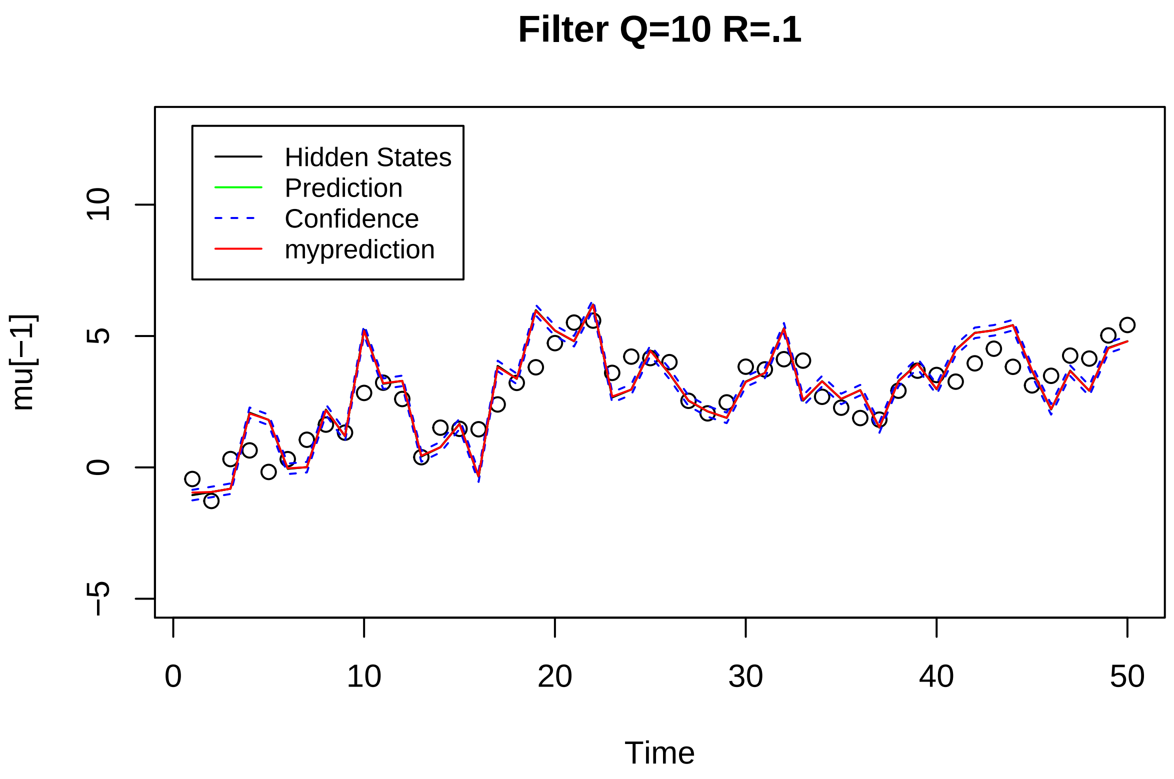
```r
kalmanfilter <- function(num, y, A, mu0, Sigma0, Phi, cQ, cR){
# init
T <- num
xs <- rep(0,T)
Ps <- rep(0,T)
I <- diag(1,dim(matrix(mu0))[1])
for(t in 1:T){
  #obs update step
  K <- Sigma0%*% t(Phi) %*% solve(Phi%*%Sigma0%*%t(Phi)+cR)
  xs[t] <- mu0 + K%*%(y[t] - Phi%*%mu0)
  Ps[t] <- (I - K%*%Phi)%*%Sigma0
  #prediction step
  mu0 <- A%*%xs[t]
  Sigma0 <- A%*%Ps[t]%*%t(A)+cQ
}
filteroutput <- list()
filteroutput$xs <- xs
filteroutput$Ps <- Ps
filteroutput$x0n <- mu0
filteroutput$P0n <- Sigma0
return(filteroutput)
}
# Q=1, R=1
ourkf <- kalmanfilter(num, y, A = 1, mu0 = 0, Sigma0 = 1, Phi = 1,
cQ = 1, cR = 1)
ks = Ksmooth0(num, y, A = 1, mu0 = 0, Sigma0 = 1, Phi = 1,
cQ = 1, cR = 1)
plot(Time, mu[-1], main = "Filter Q=1 R=1", ylim = c(-5,13))
lines(Time, y, col = "green")
lines(ks$xf)
lines(ks$xf + 2 * sqrt(ks$Pf), lty = 2, col = 4)
lines(ks$xf - 2 * sqrt(ks$Pf), lty = 2, col = 4)
lines(ourkf$xs, col = "red")
legend(1, 13, legend = c("Hidden States", "Prediction","Confidence", "myprediction"), col = c("black",
```



```r
# Q=0.10,R=10
ourkf1 <- kalmanfilter(num, y, A = 1, mu0 = 0, Sigma0 = 1, Phi = 1,cQ = .1, cR = 10)
ks1 = Ksmooth0(num, y, A = 1, mu0 = 0, Sigma0 = 1, Phi = 1,cQ = .1, cR = 10)
plot(Time, mu[-1], main = "Filter Q=.1 R=10", ylim = c(-5,13))
lines(Time, y, col = "green")
lines(ks1$xf)
lines(ks1$xf + 2 * sqrt(ks1$Pf), lty = 2, col = 4)
lines(ks1$xf - 2 * sqrt(ks1$Pf), lty = 2, col = 4)
lines(ourkf1$xs, col = "red")
legend(1, 13, legend = c("Hidden States", "Prediction",
"Confidence", "myprediction"), col = c("black", "green",
"blue", "red"), lty = c(1, 1, 2), cex = 0.8)
```



```
# Q=10,R=0.10
ourkf2 <- kalmanfilter(num, y, A = 1, mu0 = 0, Sigma0 = 1, Phi = 1,cQ = 10, cR = 0.1)
ks2 = Ksmooth0(num, y, A = 1, mu0 = 0, Sigma0 = 1, Phi = 1,cQ = 10, cR = 0.1)
plot(Time, mu[-1], main = "Filter Q=10 R=.1", ylim = c(-5,13))
lines(Time, y, col = "green")
lines(ks2$xf)
lines(ks2$xf + 2 * sqrt(ks2$Pf), lty = 2, col = 4)
lines(ks2$xf - 2 * sqrt(ks2$Pf), lty = 2, col = 4)
lines(ourkf2$xs, col = "red")
legend(1, 13, legend = c("Hidden States", "Prediction",
"Confidence", "myprediction"), col = c("black", "green",
"blue", "red"), lty = c(1, 1, 2), cex = 0.8)
```



The kalman function was constructed according to the Lab questions provided. The function varies from the function ksmooth0 from the package astsa as it appears to be smoother than our implementation.

f. How do you interpret the Kalman gain?

The Kalman gain, G_t is expressing a belief parameter for the predicted prior $P_{t+1|t}$ accounting for the next state $(t+1)$. It is a value $\in (0,1)$ and is computed as $G_t = \frac{P_t C^T}{A P_t A^T + R} = \frac{P_{t|t} A_t^T}{P_{t+1|t}}$. Magnitude of belief is decided by the autocovariance matrix. Higher uncertainity in the prior suggests that it should be believed lesser.