

Homework 3 part 2

Rafael Espinosa Castaneda

2023-02-09

Setting up the variables

```
# -----Setting the parameters to be used-----

library(ggplot2)
library(gridExtra)
library(tidyverse)
library(xts)
library(dlm)
# set true values of parameters
sigma2v_tr <- 9
sigma2w_tr <- 4
m0_tr <- 0
C0_tr <- 25

n <- 40
R <- rep(NA, n)
C <- rep(NA, n)
Q <- rep(NA, n)
m<- rep(NA, n)
a<- rep(NA, n)
f<- rep(NA, n)
e<- rep(NA, n)
G<-0.8
C_0<-25
m_0<-0
F<-1.2
W<-4
V<-9

path<-"/Users/rafa/Documents/Master Austin/MAESTRIA_AUSTIN/Advanced Predictive Models/HW3/Data_hw3.xls"
sim_data<-read.csv(path)

gg_sim <- ggplot(sim_data,
```

```

      aes(y = yt,
          x = time)) +
  geom_line(linetype = "dashed",
            color = "black")

#####
# ANALYZE SIMULATED DATA USING A DLM

# construct a dlm model object
#with parameters fixed at their true values
dlm_mod <- dlm(F = F,
               GG = G,
               V = V,
               W = W,
               m0 = m_0,
               C0 = C_0)

# filter the simulated data y using the dlm
sim_data_filtered <- dlmFilter(y = sim_data$yt,
                              mod = dlm_mod)

```

Part a

```

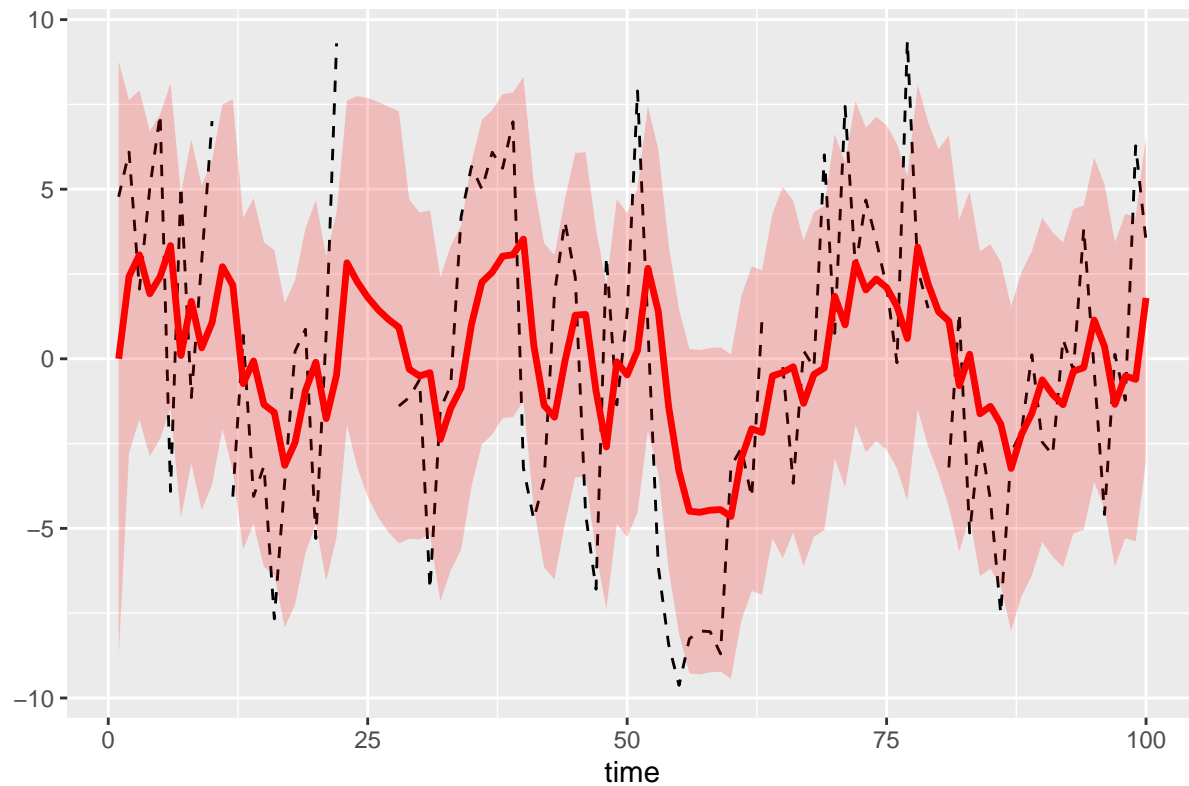
#-----Part a-----

# store and plot the one-step-ahead predictions
# of theta and standard errors
sim_data$pred <- sim_data_filtered$a
sim_data$pSE <- sqrt(unlist(
  dlmSvd2var(sim_data_filtered$U.R,
             sim_data_filtered$D.R)))

gg_sim +
  geom_line(data = sim_data,
            aes(y = pred,
                x = time),
            color = "red",
            size = 1.2) +
  geom_ribbon(data = sim_data,
            aes(x = time,
                ymin = pred - 1.96 * pSE,
                ymax = pred + 1.96 * pSE),
            fill = "red",
            alpha = 0.2) +
  labs(title = expression(
    paste("One-Step-Ahead Predictions of ",
          theta[t],
          " given observations ", y[1:t-1] ))) +
  ylab("")

```

One-Step-Ahead Predictions of θ_t given observations $y_{1:t-1}$



NUMERICAL VALUE-ANSWER

```
## [1] "Parameter R_40"
## [1] 5.950985
## [1] "Parameter a_40"
## [1] 3.528942
```

Part b

```
n <- length(sim_data_filtered$f)
R <- rep(NA, n)
C <- rep(NA, n)
Q <- rep(NA, n)
m <- rep(NA, n)
a <- rep(NA, n)
f <- rep(NA, n)
e <- rep(NA, n)
G <- 0.8
C_0 <- 25
m_0 <- 0
F <- 1.2
W <- 4
```

```

V<-9
R[1] <- G*C_0*G+W
Q[1]<-F*R[1]*F+V
C[1] <- R[1]-R[1]*F*(1/Q[1])*F*R[1]
a[1]<-m_0*G
f[1]<-F*a[1]
e[1]<-sim_data$yt[1]-f[1]
m[1]<-a[1]+R[1]*F*(1/Q[1])*e[1]

for(t in 2:n)
{

  R[t] <- G*C[t-1]*G+W
  Q[t]<-F*R[t]*F+V
  C[t] <- R[t]-R[t]*F*(1/Q[t])*F*R[t]
  a[t]<-m[t-1]*G
  f[t]<-F*a[t]
  e[t]<-sim_data$yt[t]-f[t]
  m[t]<-a[t]+R[t]*F*(1/Q[t])*e[t]

  if ( is.na(sim_data$yt[t]) )
  {
    R[t] <- G*C[t-1]*G+W
    Q[t]<-F*R[t]*F+V
    C[t] <- R[t]
    a[t]<-m[t-1]*G
    f[t]<-F*a[t]
    m[t]<-a[t]
  }
}

#store and plot the one-step-ahead predictions
# of theta and standard errors
sim_data$pred_y <- sim_data_filtered$f
sim_data$pSE2 <- sqrt(Q)

gg_sim +
  geom_line(data = sim_data,
            aes(y = pred_y,
                x = time),
            color = "purple",
            size = 1.2) +
  geom_ribbon(data = sim_data,
            aes(x = time,
                ymin = pred_y - 1.96 * pSE2,
                ymax = pred_y + 1.96 * pSE2),
            fill = "purple",

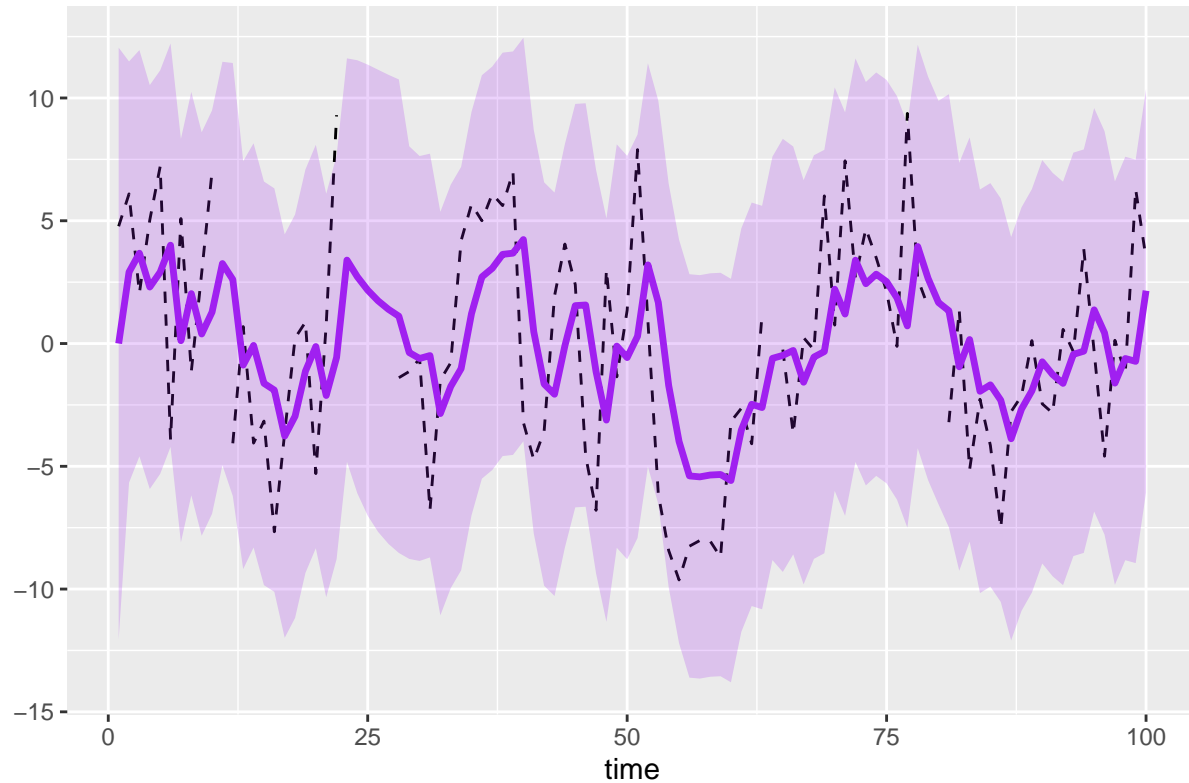
```

```

      alpha = 0.2) +
labs(title = expression(
  paste("One-Step-Ahead Predictions of ",
        y[t],
        " given observations", y[1:t-1]))) +
ylab("")

```

One-Step-Ahead Predictions of y_t given observations $y_{1:t-1}$



NUMERICAL VALUE-ANSWER

```

## [1] "Parameter Q_40"
## [1] 17.56942
## [1] "Parameter f_40"
## [1] 4.234731

```

Part c

```

sim_data$filtered_theta_upt <- dropFirst(sim_data_filtered$m)
sim_data$fSE <- dropFirst(sqrt(unlist(
  dlmSvd2var(sim_data_filtered$U.C,
             sim_data_filtered$D.C))))
gg_sim +

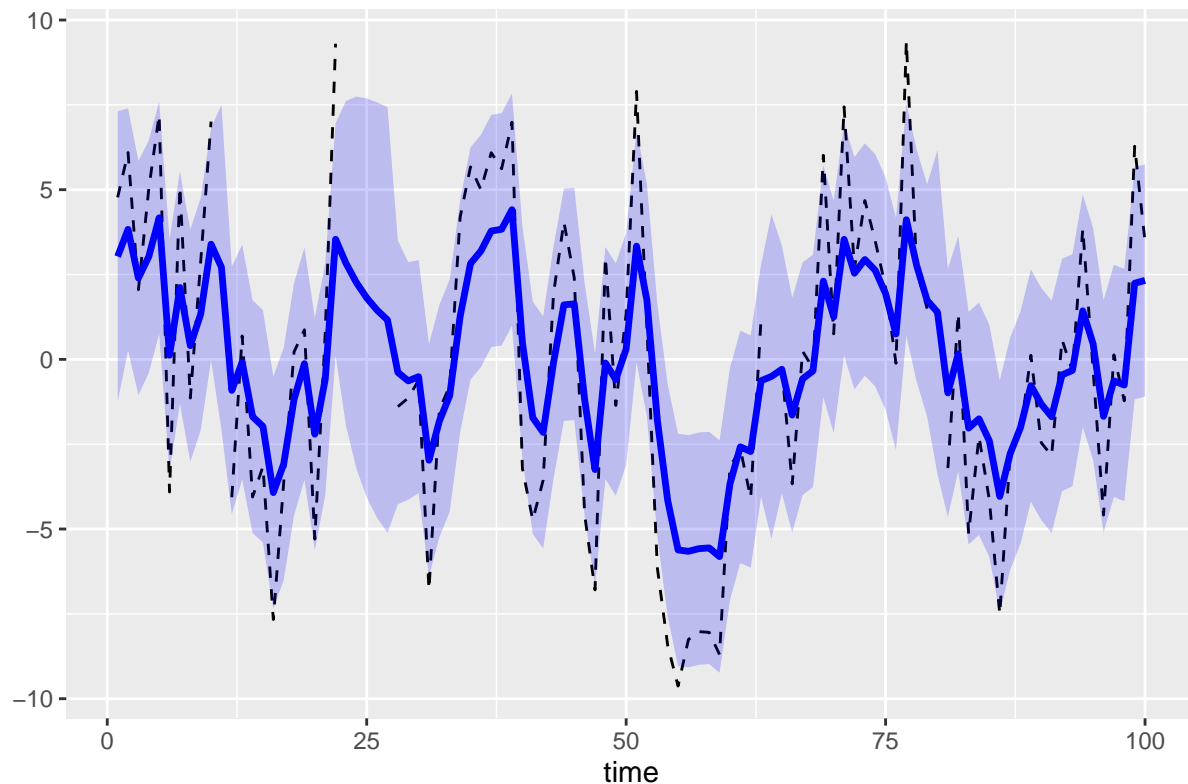
```

```

geom_line(data = sim_data,
  aes(y = filtered_theta_upt,
    x = time),
  color = "blue",
  size = 1.2) +
geom_ribbon(data = sim_data,
  aes(x = time,
    ymin = filtered_theta_upt - 1.96 * fSE,
    ymax = filtered_theta_upt + 1.96 * fSE),
  fill = "blue",
  alpha = 0.2) +
labs(title = expression(
  paste("Filtered predictions of ",
    theta[t],
    " given observations", y[1:t])))) +
ylab("")

```

Filtered predictions of θ_t given observations $y_{1:t}$



NUMERICAL VALUE-ANSWER

```

## [1] "Parameter m_40"
## [1] 0.5005116
## [1] "Parameter C_40"
## [1] 3.048414

```

Part d

```
n <- length(sim_data_filtered$f)
R <- rep(NA, n)
C <- rep(NA, n)
Q <- rep(NA, n)
m<- rep(NA, n)
a<- rep(NA, n)
f<- rep(NA, n)
e<- rep(NA, n)
G<-0.8
C_0<-25
m_0<-0
F<-1.2
W<-4
V<-9
R[1] <- G*C_0*G+W
Q[1]<-F*R[1]*F+V
C[1] <- R[1]-R[1]*F*(1/Q[1])*F*R[1]
a[1]<-m_0*G
f[1]<-F*a[1]
e[1]<-sim_data$yt[1]-f[1]
m[1]<-a[1]+R[1]*F*(1/Q[1])*e[1]

for(t in 2:n)
{

  R[t] <- G*C[t-1]*G+W
  Q[t]<-F*R[t]*F+V
  C[t] <- R[t]-R[t]*F*(1/Q[t])*F*R[t]
  a[t]<-m[t-1]*G
  f[t]<-F*a[t]
  e[t]<-sim_data$yt[t]-f[t]
  m[t]<-a[t]+R[t]*F*(1/Q[t])*e[t]

  if ( is.na(sim_data$yt[t]) )
  {
    R[t] <- G*C[t-1]*G+W
    Q[t]<-F*R[t]*F+V
    C[t] <- R[t]
    a[t]<-m[t-1]*G
    f[t]<-F*a[t]
    m[t]<-a[t]
  }
}

## [1] "Initial m_22"
## [1] 3.539185
```

[1] "Initial C_22"

[1] 3.048414

ANSWER

We will use the fact that for missing data

$$R_t = C_t$$

and

$$a_t = m_t$$

along with the recursive equations

$$a_t = G_t m_{t-1}$$

and

$$R_t = G_t C_{t-1} G'_t + W_t$$

. Also, from the observations data, we have that we miss the data from t=23 up to t=27.

Starting with

$$a_{28}$$

. We have that

$$a_{28} = G m_{27} = G a_{27} = G(G m_{26}) = G^2 a_{26} = G^2(G m_{25}) = G^3 a_{25} = G^4 m_{24} = G^5 m_{23} = G^6 m_{22}$$

.

Hence, plugging in the values

$$a_{28} = G^6 m_{22} = (0.8^6)(3.539185) = 0.92776 \approx 0.928$$

Now, for

$$R_{28}$$

, we have that

$$R_{28} = G R_{27} G + W = R_{28} = G(R_{26} + W)G + W = G^4 R_{26} + W G^2 + W = G^4(G R_{25} G + W) + W G^2 + W$$

Continuing with the calculation,

$$R_{28} = G^6 R_{25} + G^4 W + W G^2 + W = G^6(G^2 R_{24} + W) + G^4 W + W G^2 + W = G^8 R_{24} + G^6 W + G^4 W + W G^2 + W$$

Continuing with the calculation,

$$R_{28} = G^8(G^2 R_{23} + W) + G^6 W + G^4 W + W G^2 + W = G^{10} R_{23} + G^8 W + G^6 W + G^4 W + W G^2 + W$$

Finally,

$$R_{28} = G^{10}(G^2 C_{22} + W) + G^8 W + G^6 W + G^4 W + W G^2 + W$$

Hence,

$$R_{28} = G^{12} C_{22} + G^{10} W + G^8 W + G^6 W + G^4 W + W G^2 + W$$

Plugging in the values

$$R_{28} = (0.8)^{12}(3.048414) + (0.8)^{10}(4) + (0.8)^8(4) + (0.8)^6(4) + (0.8)^4(4) + (4)(0.8)^2 + 4 = 10.557046$$

So, $\theta_{28}|y_{1:27}$ follows a distributioun $N(a = 0,928, R_{28} = 10.557)$

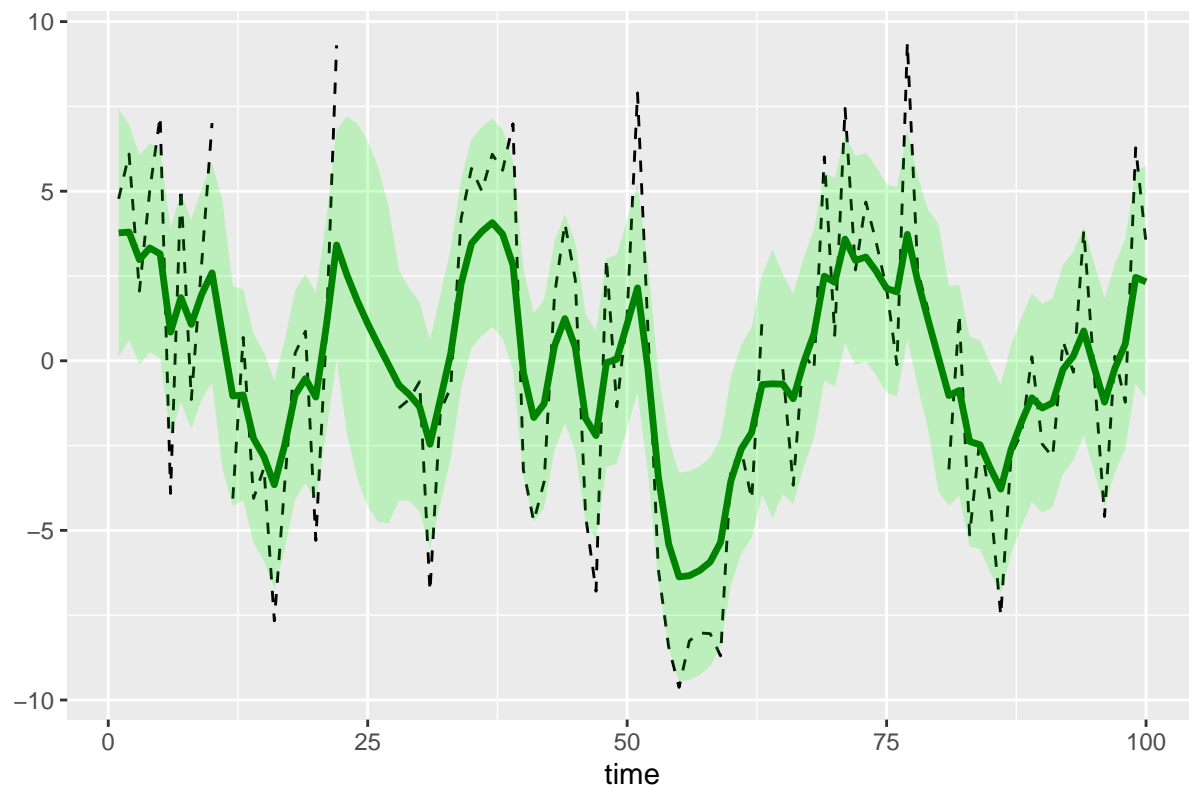
Part e

#-----Part e-----

```
sim_data_smoothed <- dlmSmooth(sim_data_filtered)
sim_data$smoothed <- dropFirst(sim_data_smoothed$s)
sim_data$sSE <- dropFirst(sqrt(unlist(
  dlmSvd2var(sim_data_smoothed$U.S,
    sim_data_smoothed$D.S))))

gg_sim + geom_line(data = sim_data,
  aes(y = smoothed,
    x = time),
  color = "darkgreen",
  size = 1.2) +
  geom_ribbon(data = sim_data,
    aes(x = time,
      ymin = smoothed - 1.96 * sSE,
      ymax = smoothed + 1.96 * sSE),
    fill = "green",
    alpha = 0.2) +
  labs(title = expression(
    paste("Smoothed estimates of ",
      theta[t],
      " given", y[1:T]))) +
  ylab("")
```

Smoothed estimates of θ_t given $y_{1:T}$



NUMERICAL VALUE-ANSWER

```
## [1] "For missing y in time t= 11 theta is "
```

```
## [1] 0.7579377
```

```
## [1] "For missing y in time t= 23 theta is "
```

```
## [1] 2.527954
```

```
## [1] "For missing y in time t= 24 theta is "
```

```
## [1] 1.767452
```

```
## [1] "For missing y in time t= 25 theta is "
```

```
## [1] 1.095323
```

```
## [1] "For missing y in time t= 26 theta is "
```

```
## [1] 0.4779599
```

```
## [1] "For missing y in time t= 27 theta is "
```

```
## [1] -0.1155051
```

```
## [1] "For missing y in time t= 64 theta is "
```

```
## [1] -0.6779439
```

```
## [1] "For missing y in time t= 80 theta is "
```

```
## [1] 0.0865769
```

Part f

```
d1m_mod2 <- d1m(FF = F,  
                GG = G,  
                V = V,
```

```

W = W,
m0 = m_0,
C0 = C_0)

# filter the simulated data y using the dlm
sim_data_filtered2<- dlmFilter(y = sim_data$yt,
                             mod = dlm_mod2)

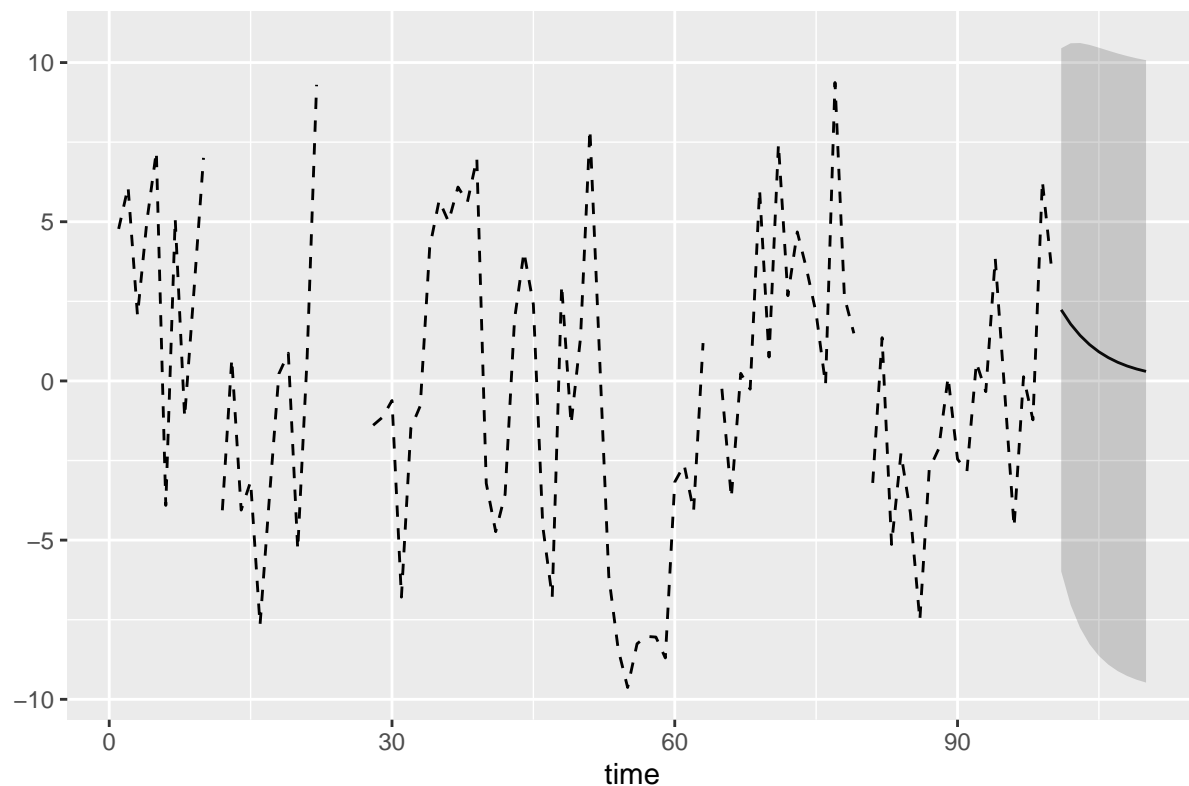
sim_data_forecasted<-dlmForecast(sim_data_filtered2, nAhead = 10, method = c("svd"))

pred_forecast <- data.frame(yt=101:110,Time = 101:110,
                           forecast = c(sim_data_forecasted$f),
                           sim_data_fsE2 = c(sqrt(unlist(sim_data_forecasted$Q))))

gg_sim+geom_line(data=pred_forecast,aes(y = forecast, x =Time)) + geom_ribbon(data = pred_forecast,
                                   aes(x = Time,
                                       ymin = forecast - 1.96*sim_data_fsE2,
                                       ymax = forecast + 1.96*sim_data_fsE2),
                                   alpha = .2) +labs(title = expression(
  paste("Original observations and forecasted values ",
        y[101:110]))) +
  ylab("")

```

Original observations and forecasted values $y_{101:110}$



ANSWER

```
## [1] "Q_101 value is"
```

```
## [1] 17.56942
```

```
## [1] "Q_110 value is"
```

```
## [1] 24.86614
```

Answering the question: The predictive variance of y_{101} is less than y_{110} because we are less uncertain about our guess (forecast) to nearer times to our observations. When we forecast just one time after a 100 observations, we have more certainty about our guess than what could happen 10 times further. As we forecast more and more into the future, we have less certainty about our guess, so increasing the variance.