DSC383 HW3

2024-06-23

Setting variables to use later

```
dlm_data <- read.csv("DLM_Data.csv")

# Plot of original data to use later
dlm_data_plot <- ggplot(dlm_data,
                        aes(y = yt,
                            x = time)) +
  geom_line(linetype = "dashed",
            color = "black")

F_t <- 1.2
G_t <- 0.8
m0_tr <- 0
C0_tr <- 25
sigma2v_tr <- 9
sigma2w_tr <- 4
```

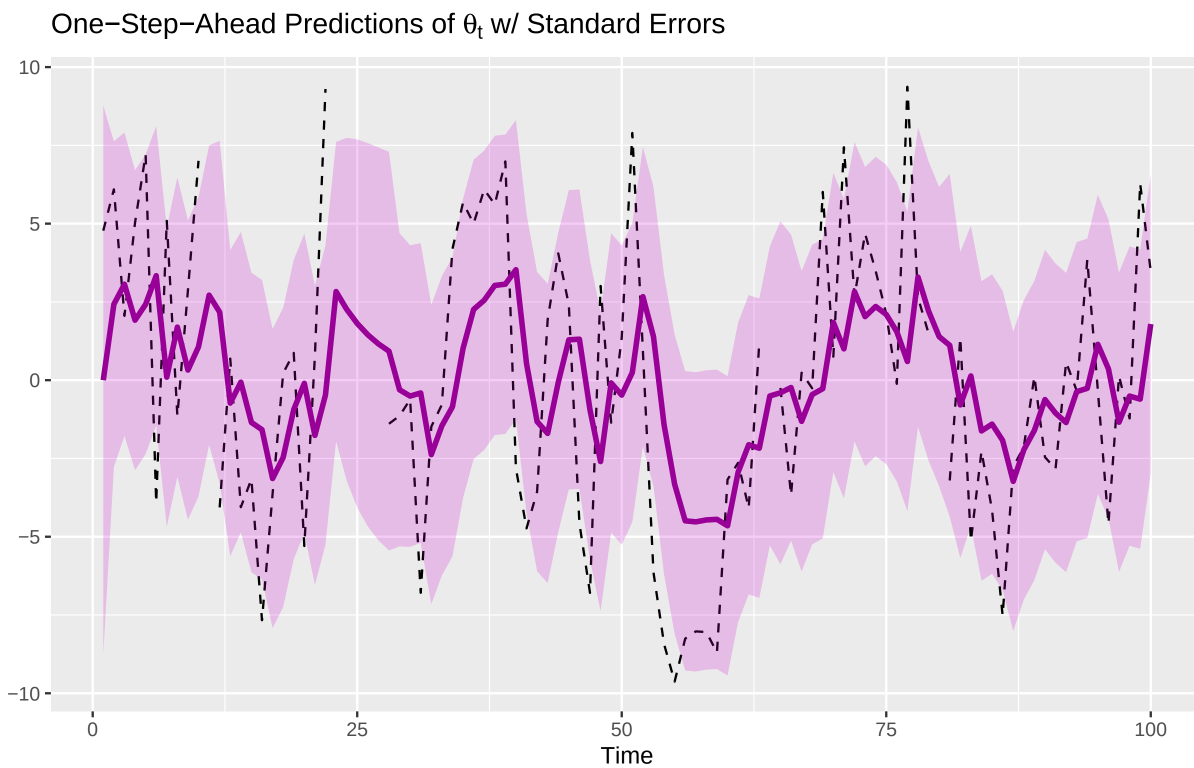
a.

Apply a Kalman filter to this data to make one-step ahead predictions of θ_t given $y_{1:(t-1)}$. Create a times-series plot containing the observations and one-step ahead predictions of y_t. Include a 95% confidence band around your θ_t predictions. Report the numerical values found for a_{40} and R_{40}.

Plot

```
# DLM object using given values
dlm_mod <- dlm(FF = F_t,
               GG = G_t,
               V = sigma2v_tr,
               W = sigma2w_tr,
               m0 = m0_tr,
               C0 = C0_tr)

# Apply Kalman filter
dlm_data_filtered <- dlmFilter(y = dlm_data$yt,
                               mod = dlm_mod)

# One-step-ahead predictions of theta
dlm_data$pred_theta <- dlm_data_filtered$a

# Variance
dlm_data$R_t <- unlist(
```

```
  dlmSvd2var(dlm_data_filtered$U.R,
             dlm_data_filtered$D.R))
# SE
dlm_data$p_theta_SE <- sqrt(dlm_data$R_t)

# Plot observed and one-step ahead predictions
dlm_theta_plot <- dlm_data_plot +
  geom_line(data = dlm_data,
            aes(y = pred_theta,
                x = time),
            color = "darkmagenta",
            size = 1.2) +
  geom_ribbon(data = dlm_data,
              aes(x = time,
                  ymin = pred_theta - 1.96 * p_theta_SE, # 95% CI
                  ymax = pred_theta + 1.96 * p_theta_SE), # 95% CI
              fill = "magenta3",
              alpha = 0.2) +
  labs(title = expression(paste("One-Step-Ahead Predictions of ",
                                theta[t],
                                " w/ Standard Errors")),
       x = "Time",
       y = "")
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
dlm_theta_plot
```



Predictions

```
a40 <- dlm_data$pred_theta[dlm_data$time == 40]
a40
## [1] 3.528942
R40 <- dlm_data$R_t[dlm_data$time == 40]
R40
## [1] 5.950985
```

$a_{40} = \mathbf{3.529}$
$R_{40} = \mathbf{5.951}$

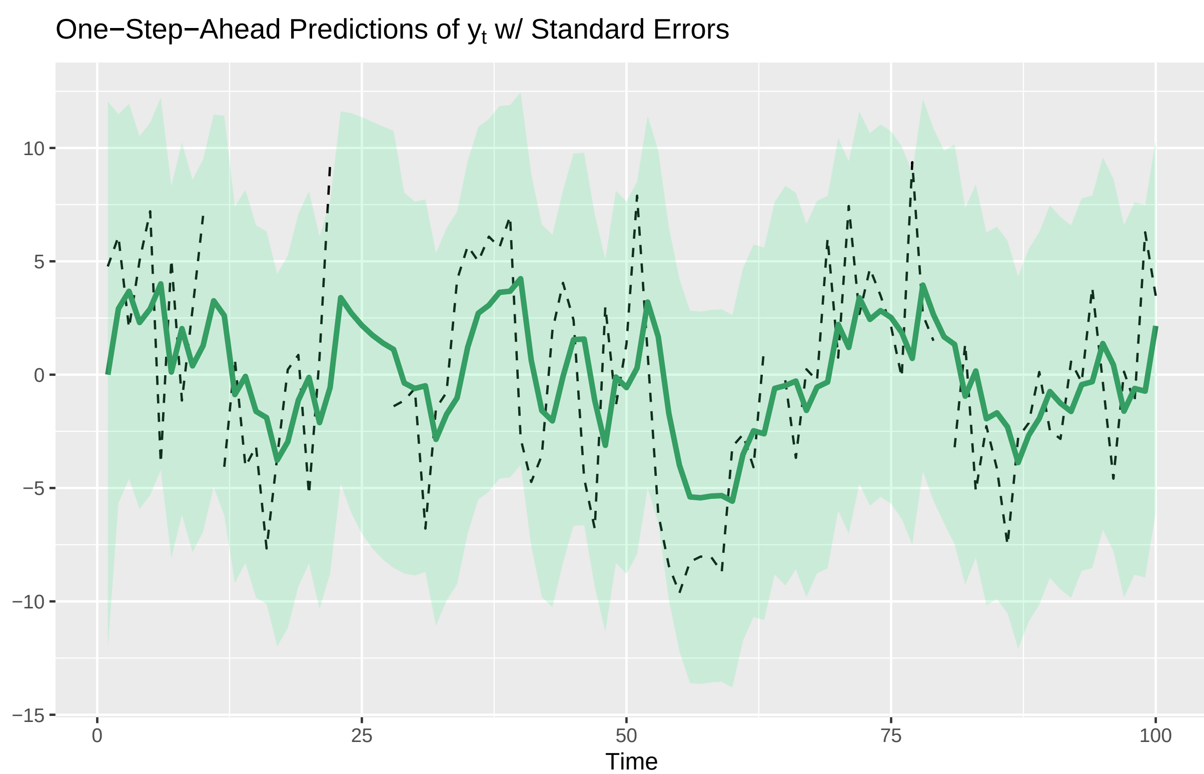
b.

Apply a Kalman filter to this data to make one-step-ahead predictions of y_t given $y_{1:(t-1)}$. Create a time-series plot showing the observed values of y_t and one-step ahead predictions of y_t. Include a 95% confidence band around your y_t predictions. Report the numerical values of f_{40} and Q_{40}. (Hint: R's DLM package does not provide these values directly, so you will need to calculate them.)

Plot
Using $Q_t = Var[y_t|y_{1:(t-1)}] = F_t R_t F_t' + V_t$, with $F_t' = F_t$

```
# One-step-ahead predictions of y
dlm_data$pred_y <- dlm_data_filtered$f

# Variance
dlm_data$Q_t <- F_t * dlm_data$R_t * F_t + sigma2v_tr
```

```r
# SE
dlm_data$p_y_SE <- sqrt(dlm_data$Q_t)

# Plot observed and one-step ahead predictions
dlm_forecast_plot <- dlm_data_plot +
  geom_line(data = dlm_data,
            aes(y = pred_y,
                x = time),
            color = "seagreen",
            size = 1.2) +
  geom_ribbon(data = dlm_data,
              aes(x = time,
                  ymin = pred_y - 1.96 * p_y_SE, # 95% CI
                  ymax = pred_y + 1.96 * p_y_SE), # 95% CI
              fill = "seagreen2",
              alpha = 0.2) +
  labs(title = expression(paste("One-Step-Ahead Predictions of ",
                                y[t],
                                " w/ Standard Errors")),
       x = "Time",
       y = "")

dlm_forecast_plot
```



Predictions

```r
f40 <- dlm_data$pred_y[dlm_data$time == 40]
f40
## [1] 4.234731
Q40 <- dlm_data$Q_t[dlm_data$time == 40]
Q40
## [1] 17.56942
```

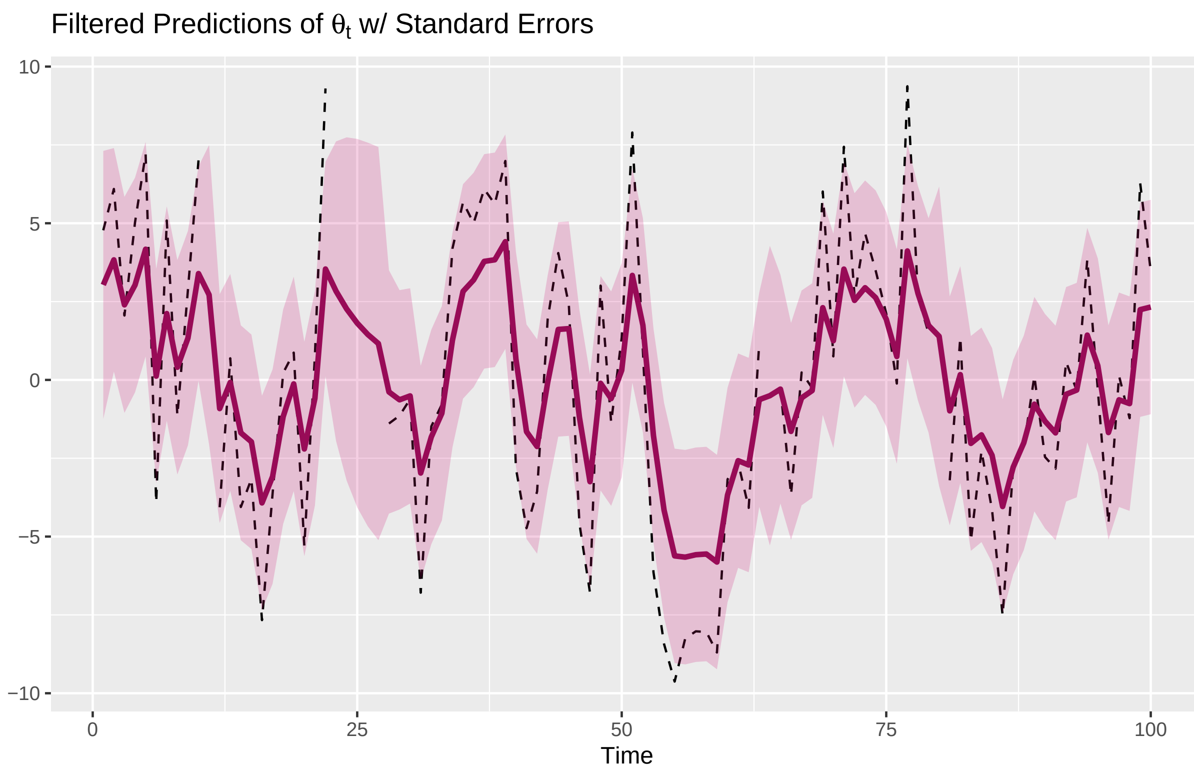
$f_{40} = \mathbf{4.235}$
$Q_{40} = \mathbf{17.569}$

c.

Apply a Kalman filter to this data to find the filtering distribution of the values of θ_t given $y_{1:(t)}$. Create a time-series plot showing the observed values of y_t and filtered predictions of θ_t. Include a 95% confidence band around your θ_t predictions. Report the numerical values of m_{40} and C_{40}.

Plot

```r
# Predictions of theta
dlm_data$filtered <- dropFirst(dlm_data_filtered$m)

# Variance
dlm_data$C_t <- dropFirst(unlist(
  dlmSvd2var(dlm_data_filtered$U.C,
             dlm_data_filtered$D.C)))
# SE
dlm_data$filtered_SE <- sqrt(dlm_data$C_t)

# Plot observed and one-step ahead predictions
dlm_filtered_plot <- dlm_data_plot +
  geom_line(data = dlm_data,
            aes(y = filtered,
                x = time),
            color = "deeppink4",
            size = 1.2) +
  geom_ribbon(data = dlm_data,
              aes(x = time,
                  ymin = filtered - 1.96 * filtered_SE, # 95% CI
                  ymax = filtered + 1.96 * filtered_SE), # 95% CI
              fill = "deeppink3",
              alpha = 0.2) +
  labs(title = expression(paste("Filtered Predictions of ",
                                theta[t],
                                " w/ Standard Errors")),
       x = "Time",
       y = "")

dlm_filtered_plot
```



Predictions

```r
m40 <- dlm_data$filtered[dlm_data$time == 40]
m40
## [1] 0.6630937
C40 <- dlm_data$C_t[dlm_data$time == 40]
C40
## [1] 3.048414
```

$m_{40} = \mathbf{0.663}$
$C_{40} = \mathbf{3.048}$

d.

The filtering distribution of $\theta_{22}|y_{1:22}$ is $N(m_{22} = 3.539, C_{22} = 3.048)$ (your answer should match this). Analytically (i.e., not using code) show that the *predictive* distribution of $\theta_{30}|y_{1:29}$ is $N(a_{30} = 0.594, R_{30} = 10.884)$. You may assume that the observations at $t = 28$ to $t = 29$ are missing as well, just like the ones from $t = 23$ to $t = 27$. (Meanwhile, think about why we would get the same distribution if we are asked to find the *forecasting* distribution of $\theta_{30}|y_{1:22}$; this part is not to be graded.)

Answer
Keeping in mind that G and W are fixed,
$\theta_{t+k}|y_{1:t} \sim N(a_t(k), R_t(k))$, with
$a_t(k) = G_{t+k}a_t(k-1) = G_t a_t(k-1)$,
$R_t(k) = G_{t+k}R_t(k-1)G'_{t+k} + W_{t+k} = G_t^2 R_t(k-1) + W_t$

Let $t = 22$ and $k = 8$ so that $t + k = 30$.

```
t <- 22
k <- 8
```

We start with the initial $m_{22} = 3.539$ and move towards $t = 30$:
$a_{22}(1) = G * m_{22}$ for $t = 23$
$a_{22}(2) = G * a_{22}(2-1) = G * G * m_{22}$ for $t = 24$
$a_{22}(3) = G * a_{22}(3-1) = G^3 * m_{22}$ for $t = 24$
...
$a_{22}(k) = G^k * m_{22}$ for $t = 30$. $a_{22}(8) = 0.8^8 * 3.539 = \mathbf{0.594}$.

Calculation:

```
m_22 <- 3.539

G_t^k * m_22
## [1] 0.5937457
```

Following the same propagation for R_t, we begin with the initial $C_{22} = 3.048$ to obtain
$R_{22}(1) = G^2 * C_{22} + W$ for $t = 23$
$R_{22}(2) = G^2 * R_{22}(2-1) + W = G^2 * (G^2 * C_{22} + W) + W$ for $t = 24$
$R_{22}(3) = G^6 C_{22} + G^4 W + G^2 W + W$ for $t = 25$
...
$R_{22}(k) = G^{2k} C_{22} + \sum_{i=1}^{k-1} G^{2i} W + W$
$R_{22}(8) = 0.8^{(2*8)}(3.048) + \sum_{i=1}^{k-1} 4G^{2i} + 4 = \mathbf{10.884}$

Calculation:

```
C_22 <- 3.048
i <- seq(1, k - 1)

summation <- sum(G_t^(2*i) * sigma2w_tr)
G_t^(2*k)*C_22 + summation +sigma2w_tr
## [1] 10.88415
```

Therefore, we have obtained $N(a_{30} = 0.594, R_{30} = 10.884)$.

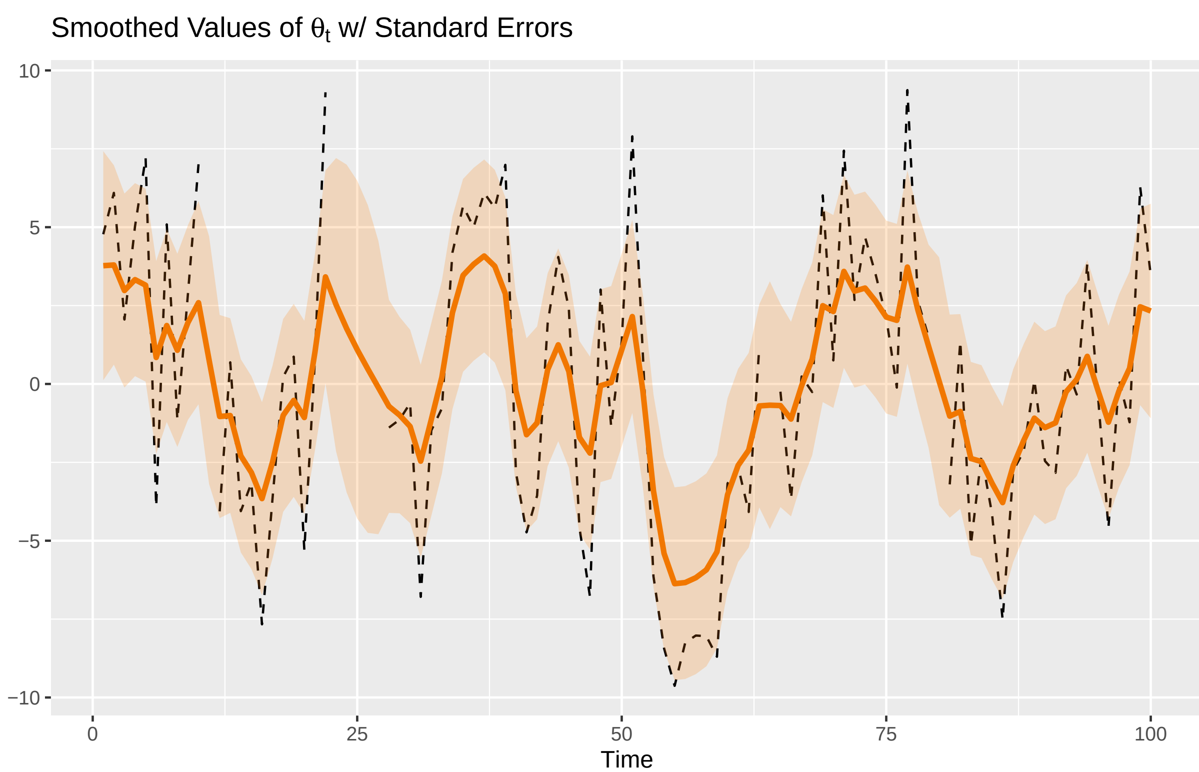
e.

Apply a Kalman smoother to this data to create the smoothing distribution for θ_t given $y_{1:T}$. Create a time-series plot showing the observed values of y_t and smoothed estimates of θ_t. Include a 95% confidence band around your θ_t predictions. Additionally, report your values of θ_t for the values of t such that y_t is missing.

Plot

```
dlm_data_smoothed <- dlmSmooth(dlm_data_filtered)
dlm_data$smoothed <- dropFirst(dlm_data_smoothed$s)
dlm_data$smoothed_SE <- dropFirst(sqrt(unlist(
  dlmSvd2var(dlm_data_smoothed$U.S,
             dlm_data_smoothed$D.S))))

dlm_smoothed_plot <- dlm_data_plot +
  geom_line(data = dlm_data,
```

```
              aes(y = smoothed,
                  x = time),
              color = "darkorange2",
              size = 1.2) +
  geom_ribbon(data = dlm_data,
                aes(x = time,
                    ymin = smoothed - 1.96 * smoothed_SE,
                    ymax = smoothed + 1.96 * smoothed_SE),
                fill = "darkorange1",
                alpha = 0.2) +
  labs(title = expression(paste("Smoothed Values of ",
                                  theta[t],
                                  " w/ Standard Errors")),
       x = "Time",
       y = "")
dlm_smoothed_plot
```



Predicted theta for missing y values

```
# Filter to rows where yt is NA
dlm_data[is.na(dlm_data$yt), c("time", "smoothed")]
##     time    smoothed
## 11    11   0.7579377
## 23    23   2.5279545
## 24    24   1.7674532
## 25    25   1.0953245
## 26    26   0.4779620
```

```
## 27   27 -0.1155023
## 64   64 -0.6779439
## 80   80  0.0865769
```

$\theta_{11} = \mathbf{0.758}$
$\theta_{23} = \mathbf{2.528}$
$\theta_{24} = \mathbf{1.767}$
$\theta_{25} = \mathbf{1.095}$
$\theta_{26} = \mathbf{0.478}$
$\theta_{27} = \mathbf{-0.116}$
$\theta_{64} = \mathbf{-0.678}$
$\theta_{80} = \mathbf{0.087}$

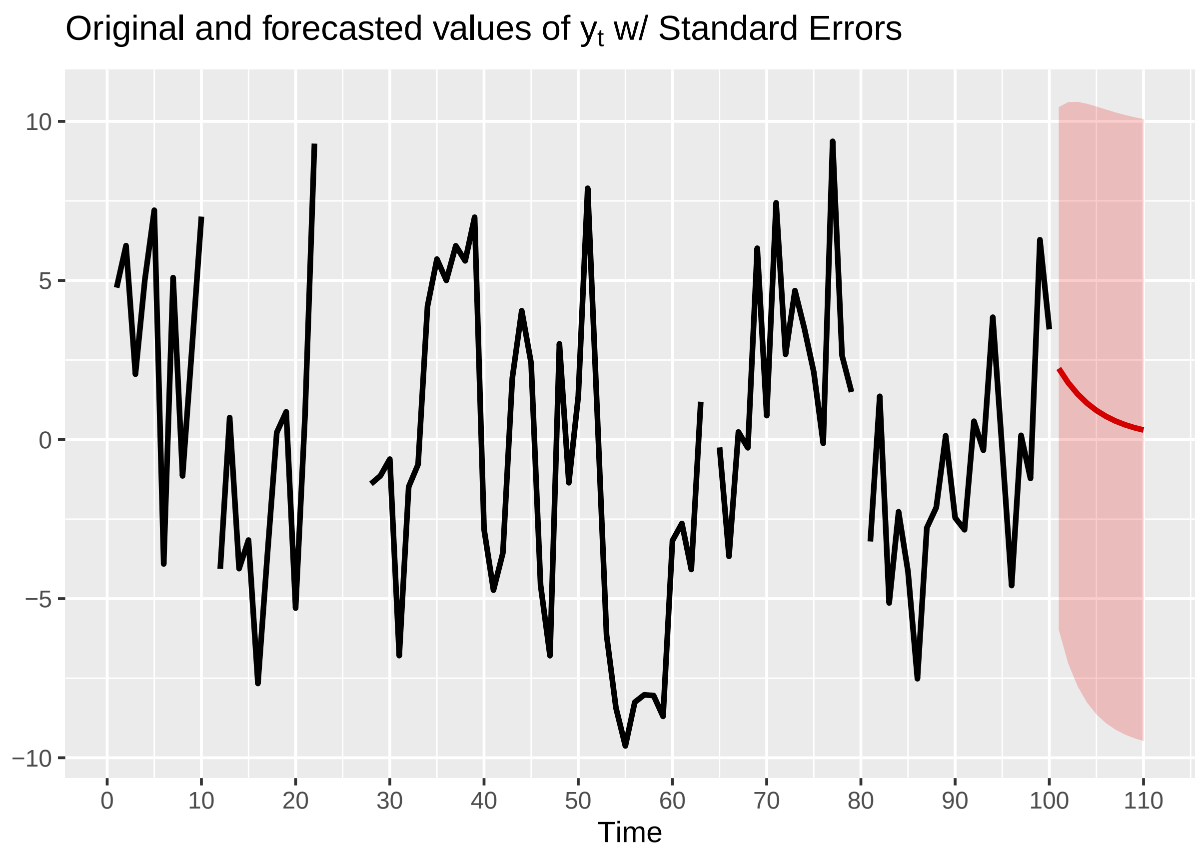
f.

Create a plot showing forecasted values (using the DLM forecasting methods discussed in lecture) of $y_{101:110}$ (including confidence bands), along with the original plot of $y_{1:100}$. Report the numerical values of Q_{101} and Q_{110} and provide a non-technical explanation for why the predictive variance of y_{101} is less than that y_{110}?

Plot

```r
# Forecast the next 10 values
forecast_future <- dlmForecast(dlm_data_filtered,
                               nAhead = 10)

forecast_data <- data.frame(
  time = 101:110,
  forecast = forecast_future$f
)

# Variance
forecast_data$Q_t <- unlist(forecast_future$Q)

# SE
forecast_data$forecast_SE <- sqrt(forecast_data$Q_t)

# Plot observed and future predictions
dlm_future_forecast_plot <- ggplot() +
  geom_line(data = dlm_data,
            aes(y = yt,
                x = time),
            color = "black",
            size = 1) +
  geom_line(data = forecast_data,
            aes(y = forecast,
                x = time),
            color = "red3",
            size = 1) +
  geom_ribbon(data = forecast_data,
              aes(x = time,
                  ymin = forecast - 1.96 * forecast_SE, # 95% CI
                  ymax = forecast + 1.96 * forecast_SE), # 95% CI
              fill = "red2",
```

```
               alpha = 0.2) +
  labs(title = expression(paste("Original and forecasted values of ",
                                y[t],
                                " w/ Standard Errors")),
       x = "Time",
       y = "") +
  scale_x_continuous(breaks=seq(0, 110, 10))

dlm_future_forecast_plot
```



Future variances

```
Q_101 <- forecast_data$Q_t[forecast_data$time == 101]
Q_101
## [1] 17.56942
Q_110 <- forecast_data$Q_t[forecast_data$time == 110]
Q_110
## [1] 24.86614
```

$Q_{101} = \mathbf{17.569}$
$Q_{110} = \mathbf{24.866}$

The predictive variance at t=101 is less than that at t=110 because there is more uncertainty as time moves further from the original data, thus we expect more variance at later time points.