Homework #5 - Fitting DLMs

Answer Key

Background

Here are the answers for the homework problems on fitting Dynamic Linear Models (DLMs). We begin by getting the data, which have the following structure:

- `brood_yr`: the year the salmon were born
- `spawners`: number of spawners
- `recruits`: number of surviving offspring that "recruit" to the fishery
- `pdo_summer_t2`: PDO during first summer at sea
- `pdo_winter_t2`: PDO during first winter at sea

```
## get S-R data; cols are:
data(KvichakSockeye, package = "atsalibrary")
SR_data <- KvichakSockeye

## head of data file
head(SR_data)
```

```
##   brood_year spawners recruits pdo_summer_t2 pdo_winter_t2
## 1       1952       NA    20200         -2.79         -1.68
## 2       1953       NA      593         -1.20         -1.05
## 3       1954       NA      799         -1.85         -1.25
## 4       1955       NA     1500         -0.60         -0.68
## 5       1956     9440    39000         -0.50         -0.31
## 6       1957     2840     4090         -2.36         -1.78
```

The first few and last years of the data set have `NA` for the spawners, so we should remove those from the analysis.

```
SR_data <- subset(SR_data, brood_year >= 1956 & brood_year <= 1998)
```

Question 1

Begin by fitting a reduced form of Equation (9.33) that includes only a time-varying level (α_t) and observation error (v_t). That is,

$$\log(R_t/S_t) = \alpha_t + v_t$$
$$\alpha_t = \alpha_{t-1} + w_t \tag{1}$$

This model assumes no density-dependent survival in that the number of recruits is an ascending function of spawners. Plot the ts of α_t and note the AICc for this model. Also plot appropriate model diagnostics.

Answer

The stock-recruit model here is a random walk observed with error, which we have seen a lot in class. To see the equivalency, instead write the observation model as

$$y_t = x_t + v_t \tag{2}$$

where $y_t = \log(R_t/S_t)$ and $x_t = \alpha_t$. The process model is then

$$x_t = x_{t-1} + w_t \tag{3}$$

The first thing we need is to compute the response variable $y_t = \log(R_t/S_t)$.

```
## Time series of ln(R/S)
lnRS <- log(SR_data$recruits / SR_data$spawners)
dat <- matrix(lnRS, nrow = 1)
## number of years of data--we'll need this later
TT <- length(lnRS)
```

Now we can set up the DLM as a level-only model (i.e., a random walk with observation error) and fit it with MARSS.

```
library(MARSS)
## MARSS model defn
## for process eqn
BB <- matrix(1)
UU <- matrix(0)
QQ <- matrix("q")
## for observation eqn
ZZ <- matrix(1)
AA <- matrix(0)
RR <- matrix("r")
## only need starting values for regr parameters
inits_list <- list(x0 = matrix(1))
## list of model matrices & vectors
mod_list <- list(B = BB, U = UU, Q = QQ, Z = ZZ, A = AA, R = RR, tinitx = 0)
## fit DLM
Q1 <- MARSS(dat, inits = inits_list, model = mod_list)
```

```
## Success! abstol and log-log tests passed at 26 iterations.
## Alert: conv.test.slope.tol is 0.5.
## Test with smaller values (<0.1) to ensure convergence.
##
## MARSS fit is
## Estimation method: kem
## Convergence test: conv.test.slope.tol = 0.5, abstol = 0.001
## Estimation converged in 26 iterations.
## Log-likelihood: -51.87187
## AIC: 109.7437   AICc: 110.3591
##
##        Estimate
## R.r     0.212
## Q.q     0.303
## x0.x0   1.001
## Initial states (x0) defined at t=0
##
## Standard errors have not been calculated.
## Use MARSSparamCIs to compute CIs and bias estimates.
```

```r
## plot the time-varying level
plot.ts(t(Q1$states), ylab = expression(alpha[italic(t)]))
```
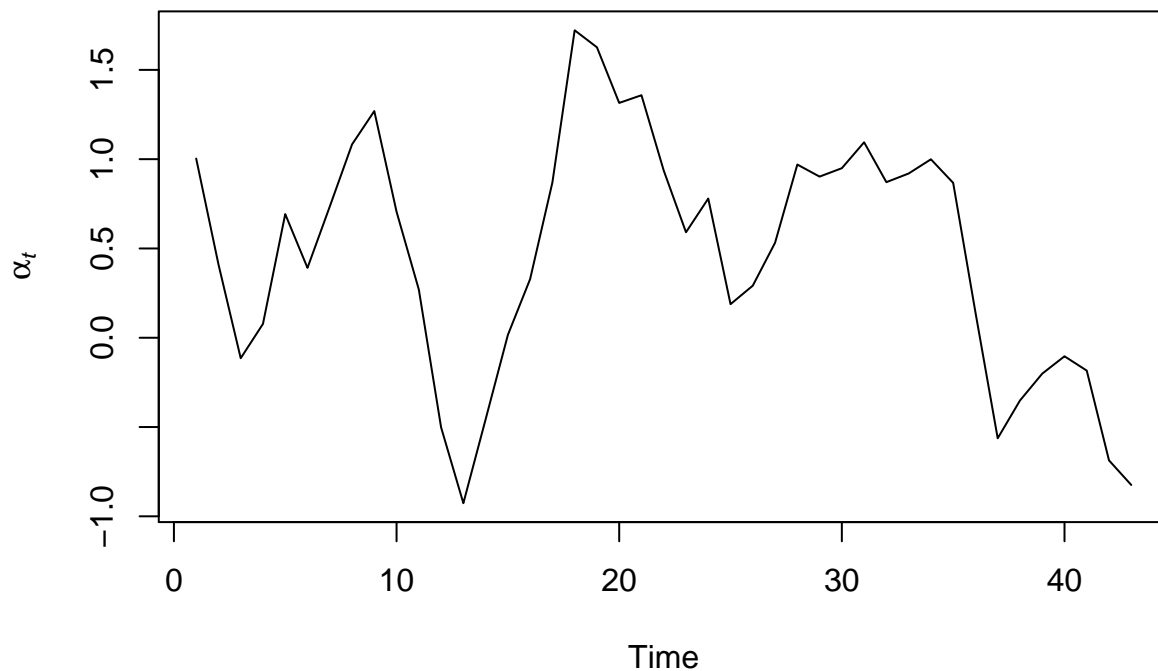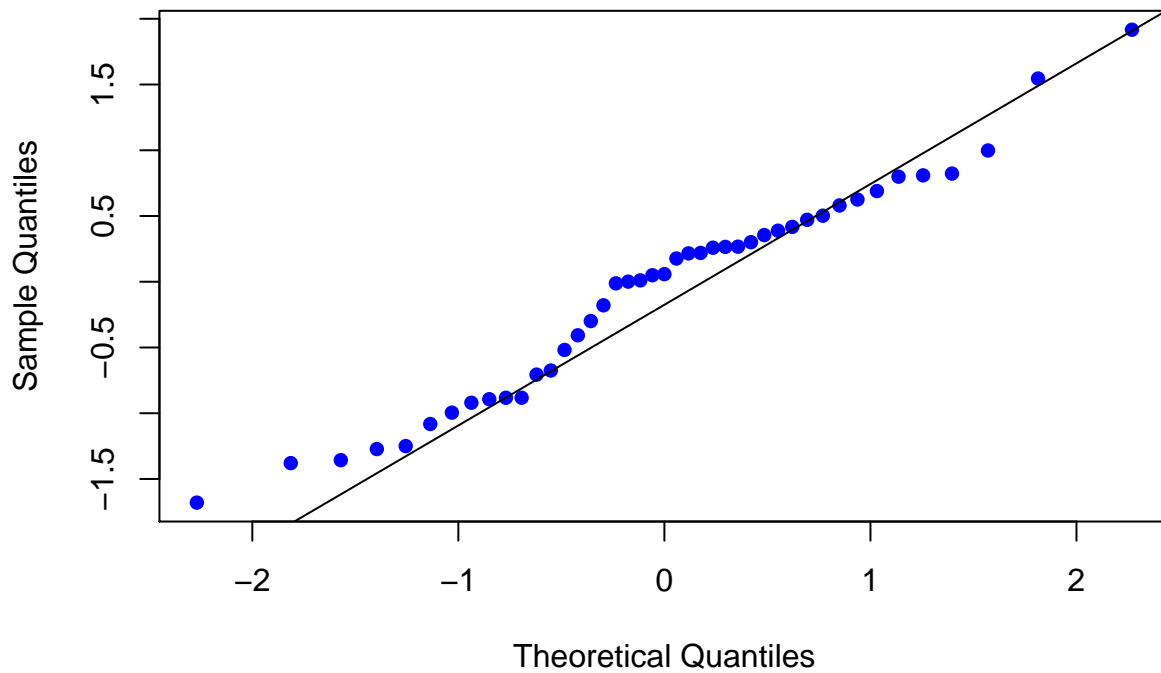


```r
## get AICc
Q1$AICc
```
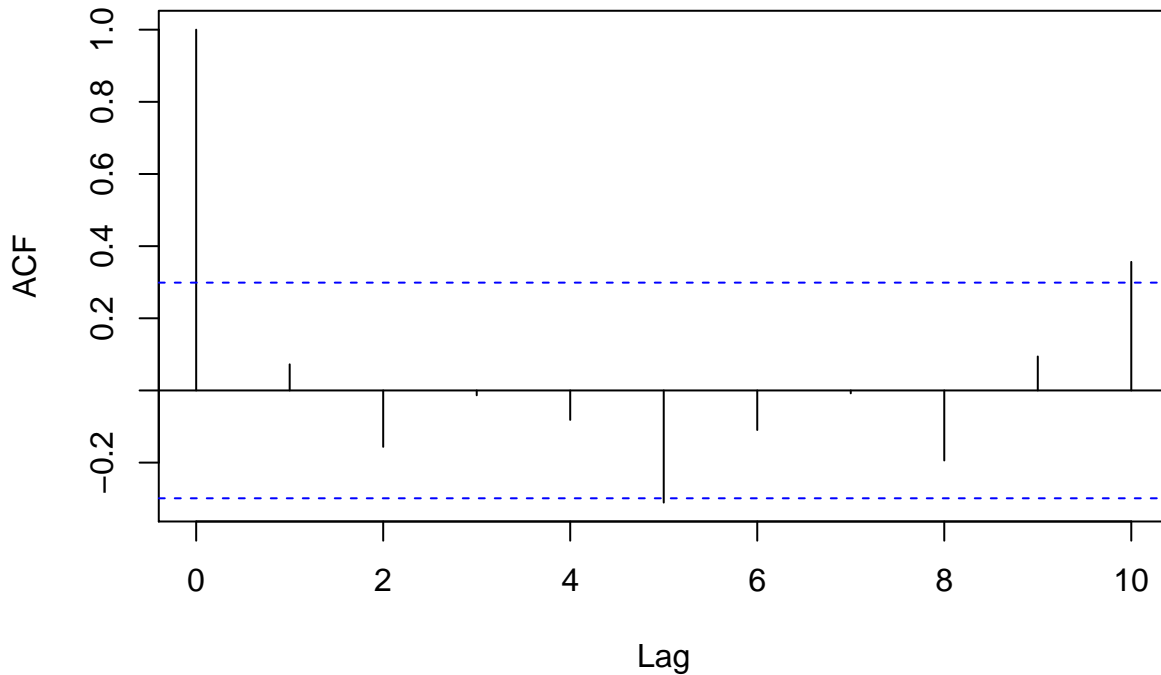
```
## [1] 110.3591
```

3

And finally examine some diagnostic plots:

```
## get list of Kalman filter output
kf_out <- MARSS::MARSSkfss(Q1)
## forecast errors
innov <- kf_out$Innov
## Q-Q plot of forecast errors
qqnorm(t(innov), main = "", pch = 16, col = "blue")
## add y = x line for easier interpretation
qqline(t(innov))
```



```
## plot ACF of innovations
acf(t(innov), lag.max = 10, main = "ACF for Q1 residuals")
```

ACF for Q1 residuals

The residuals seem to be reasonably well behaved in that there appear normal with no significant autocorrelation.

Question 2

Fit the full model specified by Equation (9.33). For this model, obtain the time series of α_t, which is an estimate of the stock productivity in the absence of density-dependent effects. How do these estimates of productivity compare to those from the previous question? Plot the ts of α_t and note the AICc for this model. Also plot appropriate model diagnostics. (*Hint*: If you don't want a parameter to vary with time, what does that say about its process variance?)

$$\log(R_t/S_t) = \alpha_t + b\log(S_t) + v_t$$
$$\alpha_t = \alpha_{t-1} + w_t \tag{4}$$

Answer

Now we need to fit a DLM with a time-varying level (intercept), but time-invariant slope. Begin by obtaining the time series of spawners to use as the covariate.

```
Sp <- matrix(SR_data$spawners, nrow = 1)
```

Set up the MARSS model structure so α varies with time, but not β, which means $q = 0$ for β. This means that **Q** should be

$$\mathbf{Q} = \begin{bmatrix} q_\alpha & 0 \\ 0 & 0 \end{bmatrix} \tag{5}$$

```
## number of regr coefs
m <- 2
## MARSS model defn
## for process eqn
B <- diag(m)                      ## 2x2; Identity
U <- matrix(0,nrow = m, ncol = 1)   ## 2x1; both elements = 0
Q <- matrix(list(0), m, m)        ## 2x2; all 0 for now
Q[1,1] <- "q_alpha"               ## 2x2; diag = (q1,q2)
## for observation eqn
Z <- array(NA, c(1, m, TT))   ## NxMxT; empty for now
Z[1,1,] <- rep(1, TT)         ## Nx1; 1's for intercept
Z[1,2,] <- Sp                 ## Nx1; regr variable
A <- matrix(0)                ## 1x1; scalar = 0
R <- matrix("r")              ## 1x1; scalar = r
## only need starting values for regr parameters
inits_list <- list(x0 = matrix(c(0, 0), nrow = m))
## list of model matrices & vectors
mod_list <- list(B = B, U = U, Q = Q, Z = Z, A = A, R = R)
## list of control params
con_list <- list(maxit = 2000)
## fit DLM
Q2 <- MARSS(dat, inits = inits_list, model = mod_list, control = con_list)


## Success! abstol and log-log tests passed at 26 iterations.
## Alert: conv.test.slope.tol is 0.5.
## Test with smaller values (<0.1) to ensure convergence.
##
## MARSS fit is
## Estimation method: kem
## Convergence test: conv.test.slope.tol = 0.5, abstol = 0.001
## Estimation converged in 26 iterations.
## Log-likelihood: -51.80921
## AIC: 111.6184   AICc: 112.6711
##
##            Estimate
## R.r        0.217557
## Q.q_alpha 0.293165
## x0.X1      0.936822
## x0.X2      0.000007
## Initial states (x0) defined at t=0
##
## Standard errors have not been calculated.
## Use MARSSparamCIs to compute CIs and bias estimates.
```

```
## plot the time-varying level
plot.ts(Q2$states[1,], ylab = expression(alpha[italic(t)]))
```
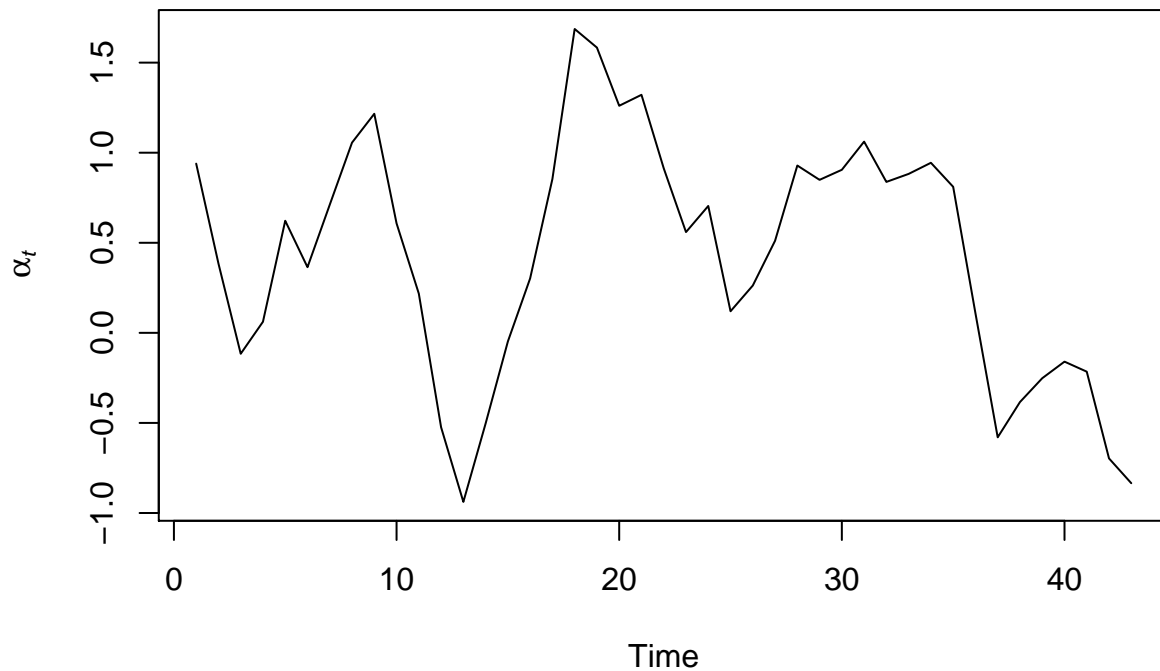

Now let's check the AICc value.

```
## get AIC
Q2$AICc
```

```
## [1] 112.6711
```

The AICc value for this model is a bit higher than that for Q1.

Let's check out some diagnostic plots for the model in Q2. First we get the model innovations from our fitted `MARSS` object.
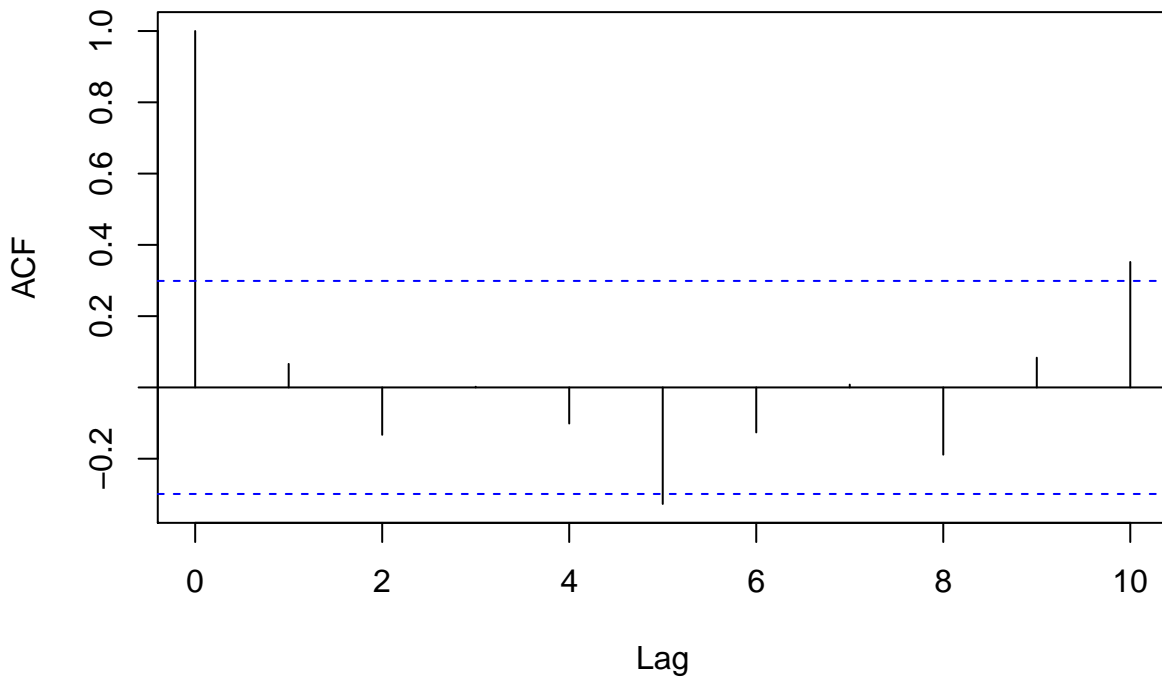
```
## get list of Kalman filter output
kf_out <- MARSS::MARSSkfss(Q2)
## forecast errors
innov <- kf_out$Innov

## Q-Q plot of forecast errors
qqnorm(t(innov), main = "", pch = 16, col = "blue")
## add y = x line for easier interpretation
qqline(t(innov))
```
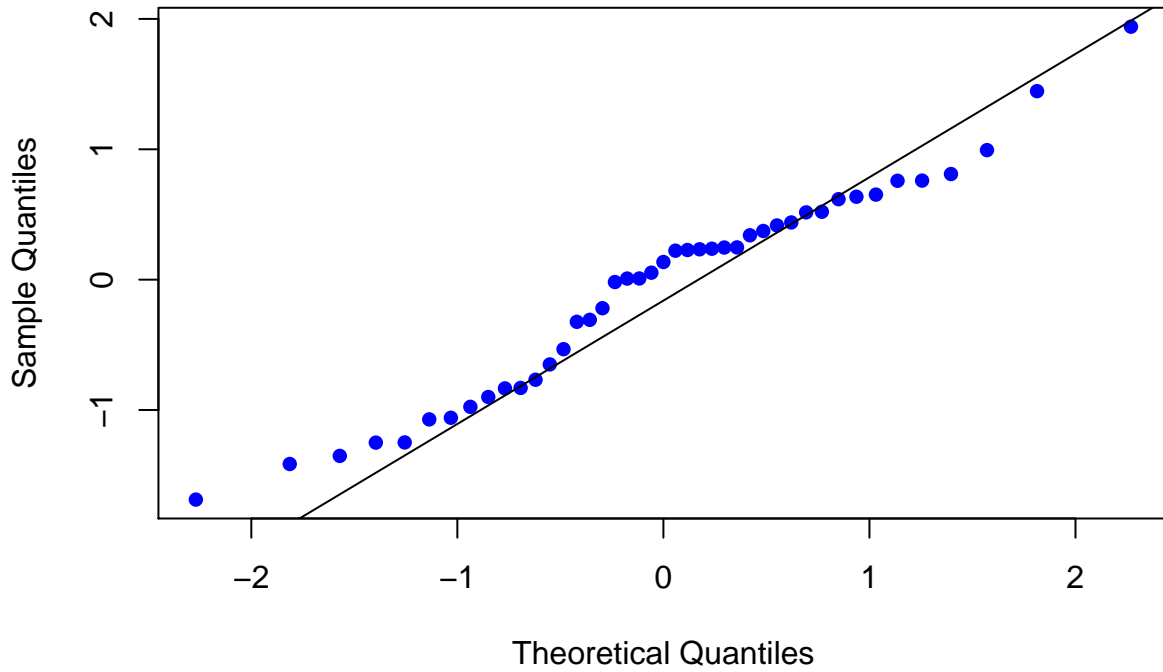
```
## plot ACF of innovations
acf(t(innov), lag.max = 10, main = "ACF for Q2 residuals")
```

ACF for Q2 residuals

The diagnostics indicate that there is some significant autocorrelation in the residuals at lags 5 & 10. The autocorrelation at lag 5 is likely a reflection of the dominant age classes of these fish.

Question 3

Fit the model specified by Equation (9.34) with the summer PDO index as the covariate (`pdo_summer_t2`). What is the mean level of productivity? Plot the ts of δ_t and note the AICc for this model. Also plot appropriate model diagnostics.

$$\log(R_t/S_t) = \alpha + \delta_t X_t - b\log(S_t) + v_t \delta_t = \delta_{t-1} + w_t \tag{6}$$

Answer

Now we need to fit a DLM so α and β are time-invariant, but δ varies by year. This means that $\mathbf{Q}$ should be

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & q_\delta \end{bmatrix} \tag{7}$$

```
## number of regr coefs
m <- 3
## MARSS model defn
## for process eqn
B <- diag(m) ## 2x2; Identity
U <- matrix(0, nrow = m, ncol = 1)
Q <- matrix(list(0), m, m)
## place delta last--it's the only one to time-vary
Q[3,3] <- ("q_delta")
## for observation eqn
Z <- array(NA, c(1, m, TT)) ## NxMxT; empty for now
Z[1,1,] <- rep(1, TT)  ## 1's for intercept
Z[1,2,] <- SR_data$spawners ## Sp regr variable
Z[1,3,] <- SR_data$pdo_summer_t2 ## summer PDO regr variable
A <- matrix(0) ## 1x1; scalar = 0
R <- matrix("r") ## 1x1; scalar = r
## only need starting values for regr parameters
inits_list <- list(x0 = matrix(c(0, 0, 0), nrow = m))
## list of model matrices & vectors
mod_list <- list(B = B, U = U, Q = Q, Z = Z, A = A, R = R)
## list of control params
con_list <- list(maxit = 2000, allow.degen = TRUE)
## fit DLM
Q3 <- MARSS(dat, inits = inits_list, model = mod_list, control = con_list)

## Success! abstol and log-log tests passed at 1172 iterations.
## Alert: conv.test.slope.tol is 0.5.
## Test with smaller values (<0.1) to ensure convergence.
##
```

```
## MARSS fit is
## Estimation method: kem
## Convergence test: conv.test.slope.tol = 0.5, abstol = 0.001
## Estimation converged in 1172 iterations.
## Log-likelihood: -54.70428
## AIC: 119.4086    AICc: 121.0302
##
##              Estimate
## R.r          7.46e-01
## Q.q_delta   0.00e+00
## x0.X1        3.56e-01
## x0.X2        1.37e-05
## x0.X3       -1.35e-01
## Initial states (x0) defined at t=0
##
## Standard errors have not been calculated.
## Use MARSSparamCIs to compute CIs and bias estimates.
```

```
## mean productivity
mean(Q3$states[1,])
```

```
## [1] 0.3563945
```

```
## plot the time-varying effect of PDO
plot.ts(Q3$states[3,], ylab = expression(delta[italic(t)]))
```
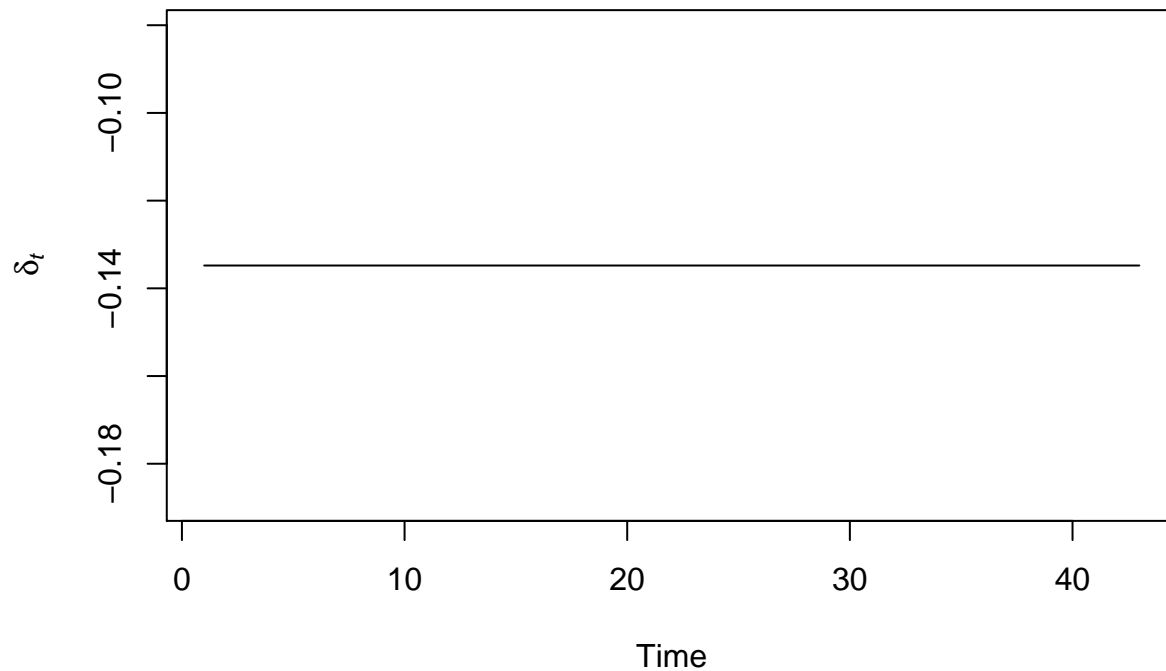

Here, as in Q2, there does not appear to be any model support for a time-varying δ.

```
## get AIC
Q3$AICc
```

```
## [1] 121.0302
```

Again, this model is no better than those in Q1 or Q2.

Here are some diagnostic plots:

```
## get list of Kalman filter output
kf_out <- MARSS::MARSSkfss(Q3)
## forecast errors
innov <- kf_out$Innov
## Q-Q plot of forecast errors
qqnorm(t(innov), main = "", pch = 16, col = "blue")
## add y = x line for easier interpretation
qqline(t(innov))
```
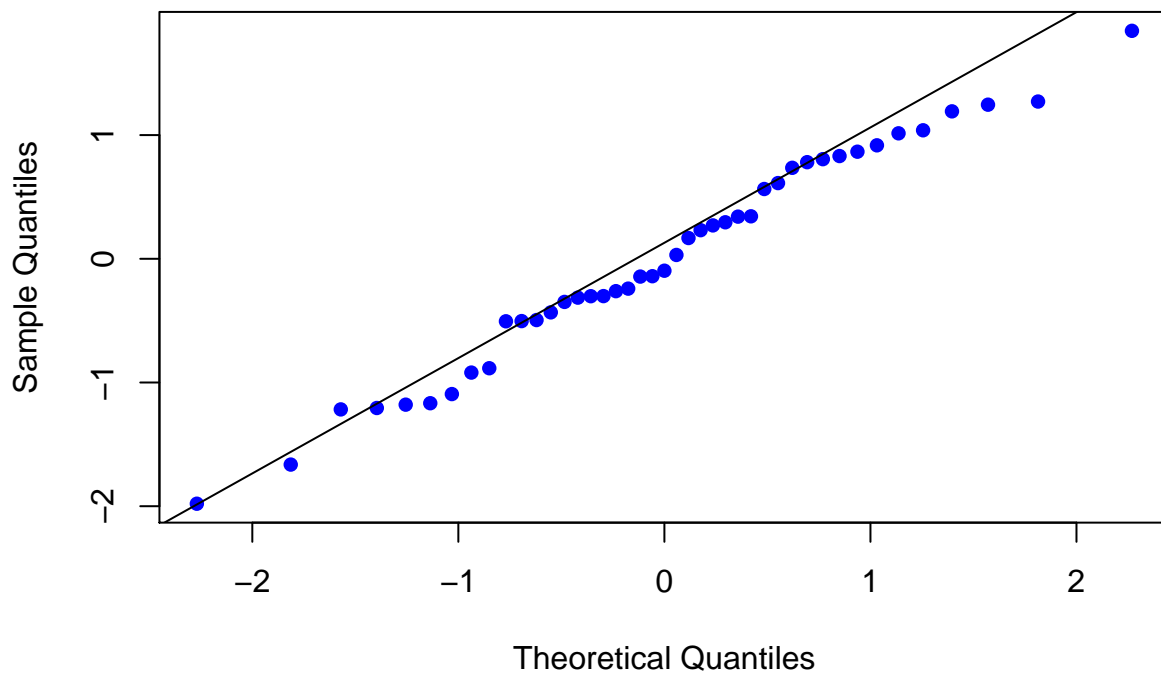


```
## plot ACF of innovations
acf(t(innov), lag.max = 10, main = "ACF for Q3 residuals")
```

ACF for Q3 residuals
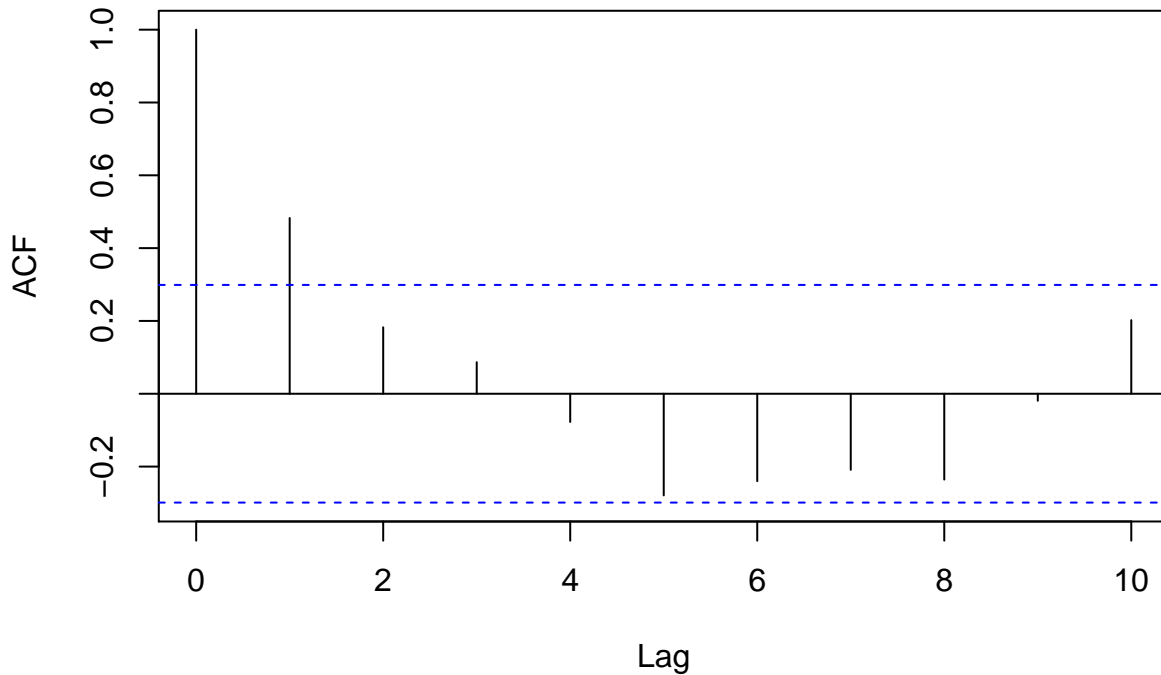
There is some indication that our model is not adequately accounting for autocorrelation in the residuals (i.e., significant correlation at lag $= 1$).

Question 4

Fit the model specified by Equation (9.34) with the winter PDO index as the covariate (`pdo_winter_t2`). What is the mean level of productivity? Plot the ts of δ_t and note the AICc for this model. Also plot appropriate model diagnostics.

Again we need to fit a DLM so that α and β are time-invariant, but δ varies by year. As for Q3, $\mathbf{Q}$ should be

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & q_\delta \end{bmatrix} \tag{8}$$

```
## number of regr coefs
m <- 3
## MARSS model defn
## for process eqn
B <- diag(m) ## 2x2; Identity
U <- matrix(0,nrow = m, ncol = 1)
Q <- matrix(list(0), m, m)
## place delta last--it's the only one to time-vary
Q[3,3] <- ("q_delta")
```

12

```r
## for observation eqn
Z <- array(NA, c(1, m, TT)) ## NxMxT; empty for now
Z[1,1,] <- rep(1, TT)  ## 1's for intercept
Z[1,2,] <- SR_data$spawners ## Sp regr variable
Z[1,3,] <- SR_data$pdo_winter_t2 ## winter PDO regr variable
A <- matrix(0) ## 1x1; scalar = 0
R <- matrix("r") ## 1x1; scalar = r
## only need starting values for regr parameters
inits_list <- list(x0 = matrix(c(0,0,0), nrow = m))
## list of model matrices & vectors
mod_list <- list(B = B, U = U, Q = Q, Z = Z, A = A, R = R)
## list of control params
con_list <- list(maxit = 2000, allow.degen = TRUE)
## fit DLM
Q4 <- MARSS(dat, inits = inits_list, model = mod_list, control = con_list)
```

```
## Success! abstol and log-log tests passed at 18 iterations.
## Alert: conv.test.slope.tol is 0.5.
## Test with smaller values (<0.1) to ensure convergence.
##
## MARSS fit is
## Estimation method: kem
## Convergence test: conv.test.slope.tol = 0.5, abstol = 0.001
## Estimation converged in 18 iterations.
## Log-likelihood: -54.09732
## AIC: 118.1946    AICc: 119.8163
##
##              Estimate
## R.r          0.508701
## Q.q_delta  0.143651
## x0.X1        0.239353
## x0.X2        0.000016
## x0.X3       -0.042999
## Initial states (x0) defined at t=0
##
## Standard errors have not been calculated.
## Use MARSSparamCIs to compute CIs and bias estimates.
```

```r
## mean productivity
mean(Q4$states[1,])
```

```
## [1] 0.2393535
```

```r
## plot the time-varying effect of PDO
plot.ts(Q4$states[3,], ylab = expression(delta[italic(t)]))
```
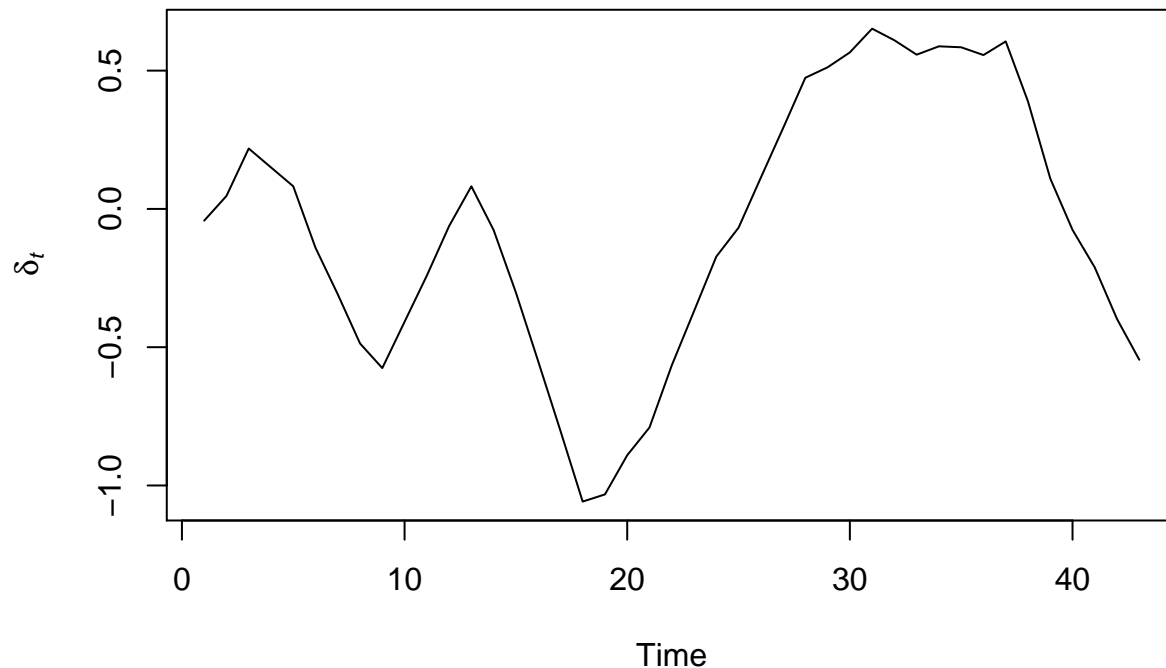
Here it appears as though there is, in fact, some data support for a time-varying effect of PDO.
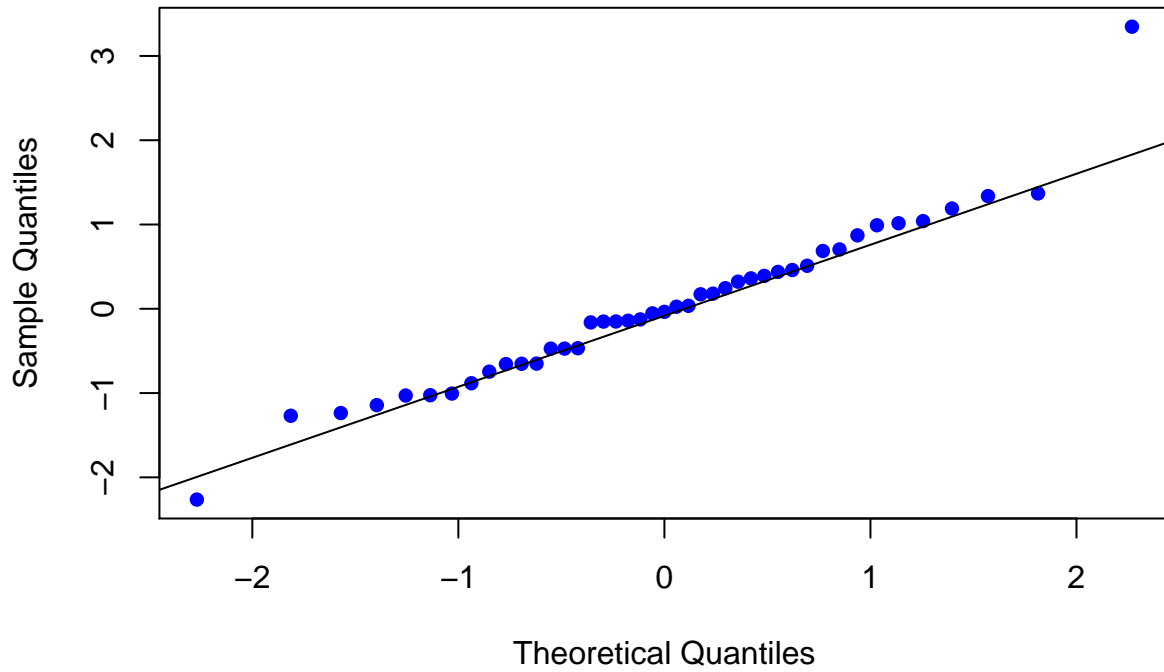
```
## get AIC
Q4$AICc
```

```
## [1] 119.8163
```

However, this model is not any better than that in Q1 or Q2.

Here are some diagnostic plots:

```
## get list of Kalman filter output
kf_out <- MARSS::MARSSkfss(Q4)
## forecast errors
innov <- kf_out$Innov
## Q-Q plot of forecast errors
qqnorm(t(innov), main = "", pch = 16, col = "blue")
## add y = x line for easier interpretation
qqline(t(innov))
```

```
## plot ACF of innovations
acf(t(innov), lag.max = 10, main = "ACF for Q4 residuals")
```

ACF for Q4 residuals
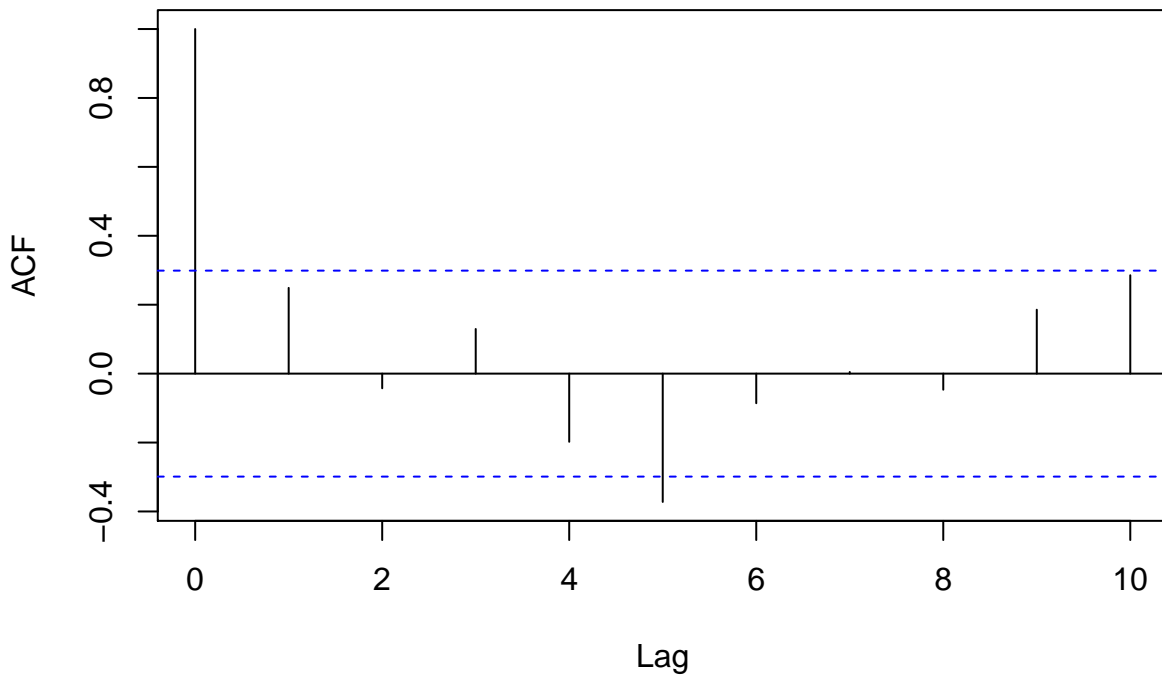
As in Q3 there is some indication that our model is not adequately accounting for autocorrelation in the residuals (i.e., signficant correlation at lags 1 & 4-5).

Question 5

Based on AICc, which of the models above is the most parsimonius? Is it well behaved (*i.e.*, are the model assumptions met)? Plot the model forecasts for the best model. Is this a good forecast model?

Here is a table of AICc values for all 4 models.

```
tbl_aicc <- data.frame(model = paste0("Q",seq(4)),
                       AICc = round(c(Q1$AICc,Q2$AICc,Q3$AICc,Q4$AICc),1))
tbl_aicc
```

```
##   model  AICc
## 1    Q1 110.4
## 2    Q2 112.7
## 3    Q3 121.0
## 4    Q4 119.8
```

The model we fit in Q1 appears to have the lowest AIC, so let's use that for forecasting.

Here's how to obtain the time series of forecasts (and their SE) for the best model.

```
## get list of Kalman filter output
kf_out <- MARSS::MARSSkfss(Q1)
## forecasts of regr parameters; 2xT matrix
eta <- kf_out$xtt1
## predictor variable (1's only for the intercept)
Z <- array(1, c(1,1,TT))   ## NxMxT; empty for now
## ts of E(forecasts)
fore_mean <- vector()
for(t in 1:TT) {
  fore_mean[t] <- Z[,,t] %*% eta[,t,drop = F]
}
## variance of regr parameters; 1x2xT array
Phi <- kf_out$Vtt1
## obs variance; 1x1 matrix
R_est <- coef(Q1, type = "matrix")$R
## ts of Var(forecasts)
fore_var <- vector()
for(t in 1:TT) {
  tZ <- matrix(Z[,,t], 1, 1) ## transpose of Z
  fore_var[t] <- Z[,,t] %*% Phi[,,t] %*% tZ + R_est
}
```
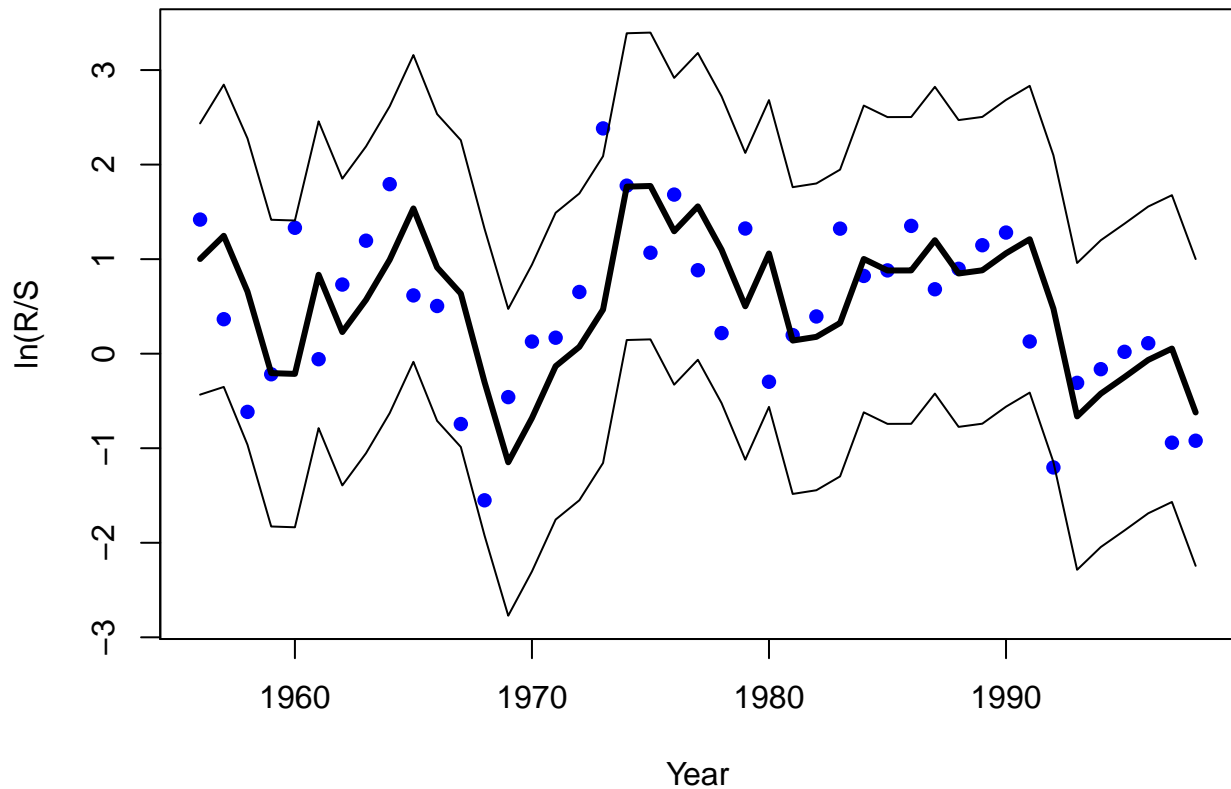
And now we can plot them.

```
fup <- fore_mean + 2 * sqrt(fore_var)
flo <- fore_mean - 2 * sqrt(fore_var)
```

```
par(mar = c(4, 4, 0.1, 0), oma = c(0, 0, 2, 0.5))
ylims <- c(min(flo), max(fup))
plot(SR_data$brood_year, t(dat), type = "p", pch = 16, ylim = ylims,
     col = "blue", xlab = "Year", ylab = "ln(R/S") ##, xaxt = "n")
lines(SR_data$brood_year, fore_mean, type = "l", xaxt = "n", ylab = "", lwd = 3)
lines(SR_data$brood_year, fup)
lines(SR_data$brood_year, flo)
```


Overall, the accuracy of the forecasts is a bit suspect as many observations are at 0.5+ log-units from the forecast. Although most of the observed ln(R/S) fell within the 95% forecast intervals, the intervals themselves are relatively large and span a large range of R/S.