Homework #4 - Fitting DFA models
Answer Key

Background

This homework builds off of the lab exercises, wherein we used DFA models to analyze the Lake Washington phytoplankton data. Here we load the data, which are contained in the `MARSS` package.

```
## load required pkgs
library("MARSS")
library("pander")
## load the data (there are 3 datasets contained here)
data(lakeWAplankton)
## we want lakeWAplanktonTrans, which has been transformed
## so the 0s are replaced with NAs and the data z-scored
all_dat <- lakeWAplanktonTrans
## use only the 10 years from 1980-1989
yr_frst <- 1980
yr_last <- 1989
plank_dat <- all_dat[all_dat[, "Year"] >= yr_frst & all_dat[, "Year"] <= yr_last,]
## create vector of phytoplankton group names
phytoplankton <- c("Cryptomonas", "Diatoms", "Greens",
                   "Unicells", "Other.algae")
## get only the phytoplankton
dat_1980 <- plank_dat[, phytoplankton]
```

Next, we transpose the data matrix and calculate the number of time series and their length.

```
## transpose data so time goes across columns
dat_1980 <- t(dat_1980)
## get number of time series
N_ts <- dim(dat_1980)[1]
## get length of time series
TT <- dim(dat_1980)[2]
```

Recall that it will be easier to estimate the real parameters of interest if we de-mean the data, so let's do that.

```
y_bar <- apply(dat_1980, 1, mean, na.rm = TRUE)
dat <- dat_1980 - y_bar
rownames(dat) <- rownames(dat_1980)
```

We will also use the following function to return the fitted values $\pm$ (1-α)% confidence intervals.

```r
get_DFA_fits <- function(MLEobj, dd = NULL, alpha = 0.05) {
  ## empty list for results
  fits <- list()
  ## extra stuff for var() calcs
  Ey <- MARSS:::MARSShatyt(MLEobj)
  ## model params
    ZZ <- coef(MLEobj, type = "matrix")$Z
    ## number of obs ts
    nn <- dim(Ey$ytT)[1]
    ## number of time steps
    TT <- dim(Ey$ytT)[2]
    ## get the inverse of the rotation matrix
    H_inv <- varimax(ZZ)$rotmat
    ## check for covars
    if(!is.null(dd)) {
      DD <- coef(MLEobj, type = "matrix")$D
      ## model expectation
      fits$ex <- ZZ %*% H_inv %*% MLEobj$states + DD %*% dd
    } else {
      ## model expectation
      fits$ex <- ZZ %*% H_inv %*% MLEobj$states
    }
    ## Var in model fits
    VtT <- MARSSkfss(MLEobj)$VtT
    VV <- NULL
    for(tt in 1:TT) {
      RZVZ <- coef(MLEobj, type = "matrix")$R - ZZ%*%VtT[,,tt]%*%t(ZZ)
      SS <- Ey$yxtT[,,tt] - Ey$ytT[,tt,drop = FALSE] %*% t(MLEobj$states[,tt,drop = FALSE])
      VV <- cbind(VV,diag(RZVZ + SS%*%t(ZZ) + ZZ%*%t(SS)))
      }
    SE <- sqrt(VV)
    ## upper & lower (1-alpha)% CI
    fits$up <- qnorm(1-alpha/2)*SE + fits$ex
    fits$lo <- qnorm(alpha/2)*SE + fits$ex
    return(fits)
}
```

Question 1

Fit other DFA models to the phytoplankton data with varying numbers of trends from 1-4 (we fit a 3-trend model above). Do not include any covariates in these models. Using `R = "diagonal and unequal"` for the observation errors, which of the DFA models has the most support from the data?

Plot the model states and loadings as in Sec 5.1. Describe the general patterns in the states and the ways the different taxa load onto those trends.

Also plot the the model fits as in Sec 5.2. Do they reasonable? Are there any particular problems or outliers?

Answer 1a

We need to fit four different DFA models and compare their support from the data via AICc. Here I use `MARSS` shorthand.

```
## num of trends to fit
trnds <- 1:4
## list with form of var-cov martix for obs errors
mod_list <- list(R = "diagonal and unequal")
## list with model control parameters
con_list <- list(maxit = 5000, allow.degen = TRUE)
## empty list for DFA results
dfa_res <- vector("list",length(trnds))
for(i in trnds) {
  ## number of trends
  mod_list$m <- i
  ## list with model inits
  init_list <- list(x0 = matrix(rep(0,i),i,1))
  dfa_res[[i]] <- MARSS(dat, model = mod_list, form = "dfa", z.score = FALSE, control = con_lis
}
```

```
## 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: -717.6663
## AIC: 1455.333   AICc: 1455.709
##
##                                Estimate
## Z.11                             0.1732
## Z.21                             0.1997
## Z.31                             0.1574
## Z.41                             0.5993
## Z.51                            -0.0344
## R.(Cryptomonas,Cryptomonas)      0.2102
## R.(Diatoms,Diatoms)              0.7335
## R.(Greens,Greens)                1.1519
```

```
## R.(Unicells,Unicells)        0.1774
## R.(Other.algae,Other.algae)  0.9357
## Initial states (x0) defined at t=0
##
## Standard errors have not been calculated.
## Use MARSSparamCIs to compute CIs and bias estimates.
##
## Success! abstol and log-log tests passed at 118 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 118 iterations.
## Log-likelihood: -705.8953
## AIC: 1439.791   AICc: 1440.514
##
##                                Estimate
## Z.11                              0.217
## Z.21                              0.144
## Z.31                              0.389
## Z.41                              0.564
## Z.51                              0.373
## Z.22                             -0.101
## Z.32                              0.172
## Z.42                             -0.174
## Z.52                              0.414
## R.(Cryptomonas,Cryptomonas)      0.198
## R.(Diatoms,Diatoms)              0.730
## R.(Greens,Greens)                0.975
## R.(Unicells,Unicells)            0.183
## R.(Other.algae,Other.algae)      0.397
## Initial states (x0) defined at t=0
##
## Standard errors have not been calculated.
## Use MARSSparamCIs to compute CIs and bias estimates.
##
## Success! abstol and log-log tests passed at 218 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 218 iterations.
## Log-likelihood: -696.3593
## AIC: 1426.719   AICc: 1427.778
```

```
##
##                                  Estimate
## Z.11                              0.2697
## Z.21                              0.4473
## Z.31                              0.3086
## Z.41                              0.4248
## Z.51                              0.2606
## Z.22                              0.3625
## Z.32                             -0.3598
## Z.42                             -0.1068
## Z.52                             -0.3927
## Z.33                              0.0309
## Z.43                             -0.1597
## Z.53                              0.1713
## R.(Cryptomonas,Cryptomonas)      0.1657
## R.(Diatoms,Diatoms)              0.2898
## R.(Greens,Greens)                0.8844
## R.(Unicells,Unicells)            0.2934
## R.(Other.algae,Other.algae)      0.4694
## Initial states (x0) defined at t=0
##
## Standard errors have not been calculated.
## Use MARSSparamCIs to compute CIs and bias estimates.
##
## Success! abstol and log-log tests passed at 4734 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 4734 iterations.
## Log-likelihood: -693.4436
## AIC: 1424.887   AICc: 1426.207
##
##                                  Estimate
## Z.11                              0.2884
## Z.21                              0.4416
## Z.31                              0.3042
## Z.41                              0.4394
## Z.51                              0.2696
## Z.22                              0.3482
## Z.32                             -0.3460
## Z.42                             -0.1134
## Z.52                             -0.4120
## Z.33                             -0.0392
## Z.43                              0.0395
## Z.53                             -0.2247
```

```
## Z.44                            -0.2107
## Z.54                             0.0526
## R.(Cryptomonas,Cryptomonas)     0.1459
## R.(Diatoms,Diatoms)             0.3086
## R.(Greens,Greens)               0.8621
## R.(Unicells,Unicells)           0.2546
## R.(Other.algae,Other.algae)     0.4102
## Initial states (x0) defined at t=0
##
## Standard errors have not been calculated.
## Use MARSSparamCIs to compute CIs and bias estimates.
```

Now we can examine the AICc values to see what model is best supported.

```
## table of AICc values
aicc <- sapply(dfa_res, function(x){x$AICc})
tbl_aicc <- data.frame(trends = trnds, AICc = aicc)
pander(tbl_aicc, caption = "List of AICc values for the fitted DFA models.",
       digits = 5)
```

Table 1: List of AICc values for the fitted DFA models.

trends	AICc
1	1455.7
2	1440.5
3	1427.8
4	1426.2

```
## best supported model
dfa_top <- which(aicc == min(aicc))
```

The best of these DFA models has 4 trends. However, the difference in AICc between the model with 3 trends is only 1.6 units, which suggests we should look at both models. That is, for every additional paremeter we add to a model, we should expect to see at least a 2-unit decrease in AICc.

Answer 1b

First, let's examine the top model, which had 4 trends. We need to rotate both **Z** and **x**.

```
## get the estimated ZZ
Z_est <- coef(dfa_res[[dfa_top]], type = "matrix")$Z
## get the inverse of the rotation matrix
H_inv <- varimax(Z_est)$rotmat
## rotate factor loadings
```

```r
Z_rot <- Z_est %*% H_inv
## rotate processes
proc_rot <- solve(H_inv) %*% dfa_res[[dfa_top]]$states
```
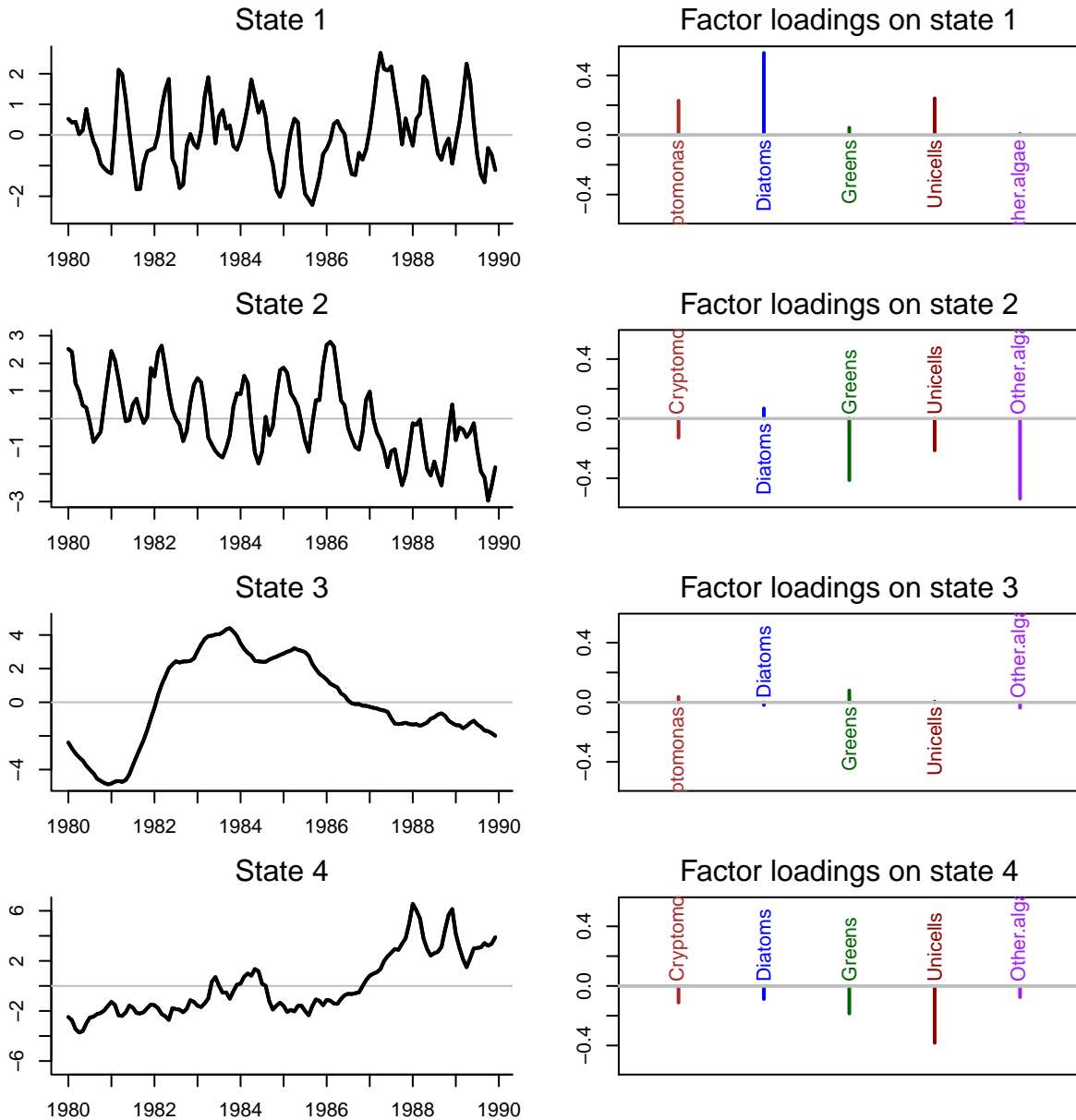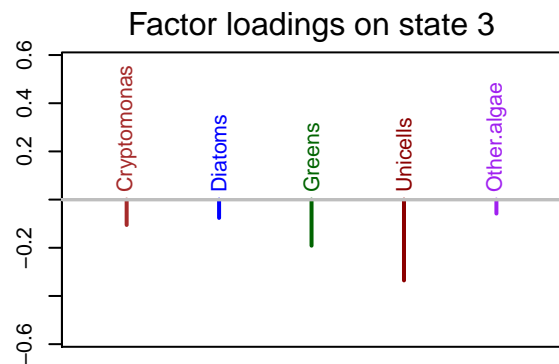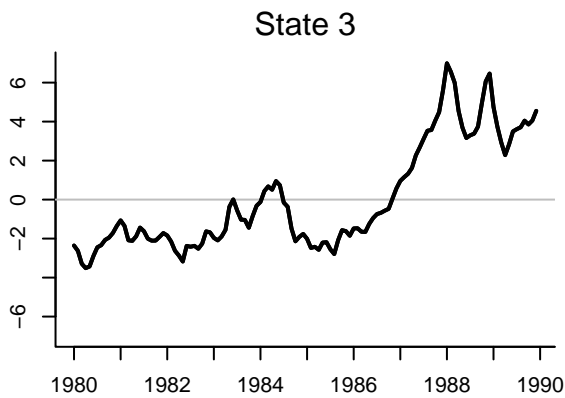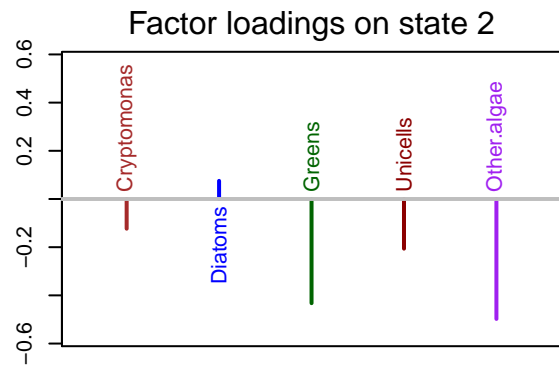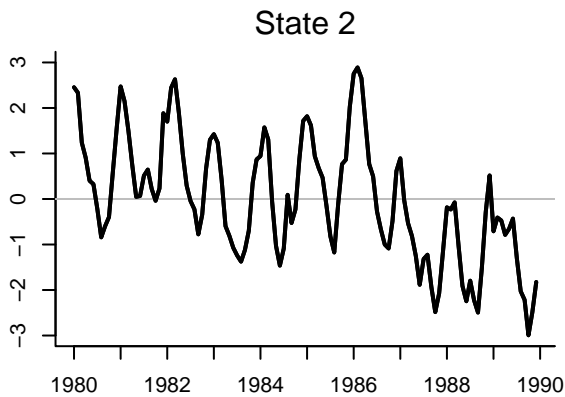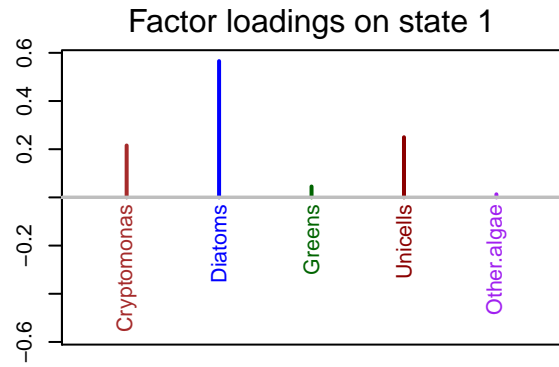
Here are plots of the trends (L) and factor loadings (R).

```r
mm <- dfa_top
ylbl <- phytoplankton
w_ts <- seq(dim(dat)[2])
clr <- c("brown","blue","darkgreen","darkred","purple")
#layout(matrix(c(1, 2, 3, 4, 5, 6), mm, 2), widths = c(2, 1))
par(mfcol = c(mm, 2), mai = c(0.3, 0.3, 0.3, 0.3), omi = rep(0.3, 4))
## plot the processes
for(i in 1:mm) {
  ylm <- c(-1, 1) * max(abs(proc_rot[i, ]))
  ## set up plot area
    plot(w_ts,proc_rot[i,], type = "n", bty = "L",
         ylim = ylm, xlab = "", ylab = "", xaxt = "n")
    ## draw zero-line
    abline(h = 0, col = "gray")
    ## plot trend line
    lines(w_ts,proc_rot[i,], lwd = 2)
    lines(w_ts,proc_rot[i,], lwd = 2)
    ## add panel labels
    mtext(paste("State", i), side = 3, line = 0.5)
    axis(1, 12 * (0:dim(dat_1980)[2])+1, yr_frst + 0:dim(dat_1980)[2])
}
## plot the loadings
minZ <- 0
ylm <- c(-1, 1)*max(abs(Z_rot))
for(i in 1:mm) {
  plot(c(1:N_ts)[abs(Z_rot[,i]) > minZ], as.vector(Z_rot[abs(Z_rot[,i]) > minZ,i]), type = "h"
       lwd = 2, xlab = "", ylab = "", xaxt = "n", ylim = ylm, xlim = c(0.5, N_ts+0.5), col = cl
    for(j in 1:N_ts) {
      if(Z_rot[j,i] > minZ) {text(j, -0.03, ylbl[j], srt = 90, adj = 1, cex = 1, col = clr[j])
      if(Z_rot[j,i] < -minZ) {text(j, 0.03, ylbl[j], srt = 90, adj = 0, cex = 1, col = clr[j])
      abline(h = 0, lwd = 1.5, col = "gray")
      }
  mtext(paste("Factor loadings on state",i), side = 3, line = 0.5)
}
```

Trends 1 & 2 show obvious seasonal patterns in the data, but appear to be somewhat out of phase with one another. Trend 3 has a quick rise and then gradual decline, but none of the time series load very hevaily on that trend. Trend 4 shows a general increase over time, but the taxa seem to load negatively on it, so that indicates general decrease over time.

Recall from lecture that I mentioned something we can look for with respect to possible overfitting of DFA models: one of the trends will have very small loadings for many/most of the groups. In this case, the loadings on trend 3 definitely appear that way. Lets look at the trends and loadings from the model with only 3 trends.

```
## get the estimated ZZ
Z_est <- coef(dfa_res[[3]], type = "matrix")$Z
## get the inverse of the rotation matrix
```
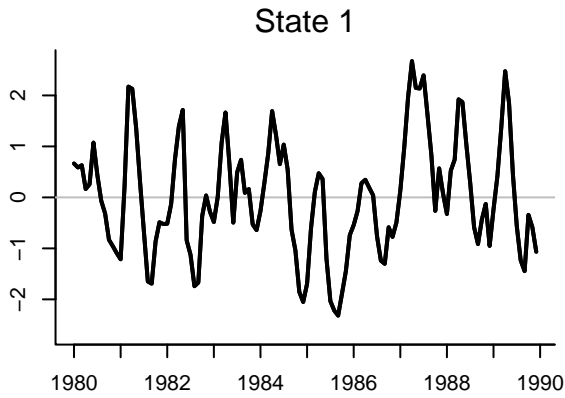
```r
H_inv <- varimax(Z_est)$rotmat
## rotate factor loadings
Z_rot <- Z_est %*% H_inv
## rotate processes
proc_rot <- solve(H_inv) %*% dfa_res[[3]]$states

## plot the trends and loadings
mm <- 3
ylbl <- phytoplankton
w_ts <- seq(dim(dat)[2])
clr <- c("brown","blue","darkgreen","darkred","purple")
par(mfcol = c(mm, 2), mai = c(0.3, 0.3, 0.3, 0.3), omi = rep(0.3, 4))
## plot the processes
for(i in 1:mm) {
  ylm <- c(-1, 1) * max(abs(proc_rot[i,]))
  ## set up plot area
    plot(w_ts,proc_rot[i,], type = "n", bty = "L",
         ylim = ylm, xlab = "", ylab = "", xaxt = "n")
    ## draw zero-line
    abline(h = 0, col = "gray")
    ## plot trend line
    lines(w_ts,proc_rot[i,], lwd = 2)
    lines(w_ts,proc_rot[i,], lwd = 2)
    ## add panel labels
    mtext(paste("State",i), side = 3, line = 0.5)
    axis(1, 12*(0:dim(dat_1980)[2]) + 1, yr_frst + 0:dim(dat_1980)[2])
}
## plot the loadings
minZ <- 0
ylm <- c(-1, 1) * max(abs(Z_rot))
for(i in 1:mm) {
  plot(c(1:N_ts)[abs(Z_rot[,i]) > minZ], as.vector(Z_rot[abs(Z_rot[,i]) > minZ, i]), type = "h"
       lwd = 2, xlab = "", ylab = "", xaxt = "n", ylim = ylm, xlim = c(0.5, N_ts + 0.5), col =
    for(j in 1:N_ts) {
       if(Z_rot[j, i] > minZ) {text(j, -0.03, ylbl[j], srt = 90, adj = 1, cex = 1, col = clr[j])
       if(Z_rot[j, i] < -minZ) {text(j, 0.03, ylbl[j], srt = 90, adj = 0, cex = 1, col = clr[j])
       abline(h = 0, lwd = 1.5, col = "gray")
       }
  mtext(paste("Factor loadings on state",i),side = 3,line = 0.5)
}
```

Indeed, these 3 trends and loadings look very similar to trends 1, 2 & 4 from the best model. Thus, the plots in Answer 1c below are based on the 3-trend model.

Answer 1c

We can plot the fits for our DFA model along with the data. Here are time series of the five phytoplankton groups (points) with the mean of the DFA fits (black line) and the 95% confidence intervals (gray lines).

```
## get model fits & CI's
mod_fit <- get_DFA_fits(dfa_res[[3]])
## plot the fits
```

```r
ylbl <- phytoplankton
par(mfrow = c(N_ts,1), mai = c(0.5,0.7,0.1,0.1), omi = rep(0.3,4))
for(i in 1:N_ts) {
  up <- mod_fit$up[i,]
  mn <- mod_fit$ex[i,]
  lo <- mod_fit$lo[i,]
  plot(w_ts, mn, xlab = "", ylab = ylbl[i], xaxt = "n", type = "n", cex.lab = 1.2,
       ylim = c(min(lo),max(up)))
  axis(1, 12 * (0:dim(dat_1980)[2]) + 1, yr_frst + 0:dim(dat_1980)[2])
  points(w_ts,dat[i,], pch = 16, col = clr[i])
  lines(w_ts, up, col = "darkgray")
  lines(w_ts, mn, col = "black", lwd = 2)
  lines(w_ts, lo, col = "darkgray")
}
```
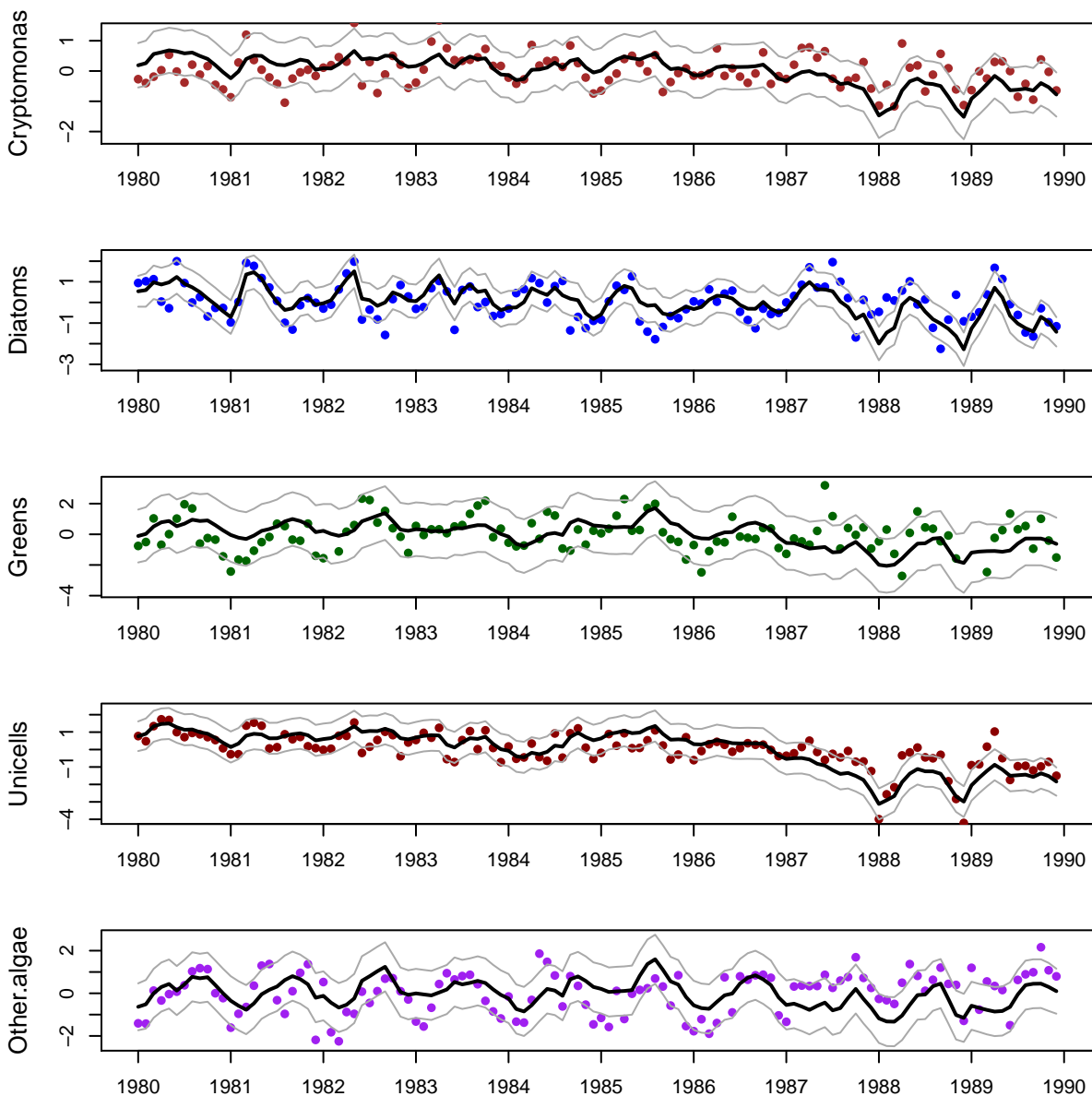
These model fits look pretty good overall as they seem to capture much of the seasonal variation.

Question 2

How does the best model from Q1 compare to a DFA model with the same number of trends, but with `R = "unconstrained"`?

Plot the model states and loadings as in Sec 5.1. Describe the general patterns in the states and the ways the different taxa load onto those trends.

Also plot the the model fits as in Sec 5.2. Do they reasonable? Are there any particular problems or outliers?

Answer 2a

Here is the DFA model with 4 trends and diagonal and unequal variance-covariance matrix.

```
## list with form of var-cov martix for obs errors
mod_list <- list(m = dfa_top, R = "unconstrained")
## list with model control parameters
con_list <- list(maxit = 6000, allow.degen = TRUE)
## list with model inits
init_list <- list(x0 = matrix(rep(0,i),i,1))
## fit DFA model
dfa_2 <- MARSS(dat, model = mod_list, form = "dfa", z.score = FALSE, control = con_list)


## Warning! Abstol convergence only. Maxit (=6000) reached before log-log convergence.
##
## MARSS fit is
## Estimation method: kem
## Convergence test: conv.test.slope.tol = 0.5, abstol = 0.001
## WARNING: Abstol convergence only no log-log convergence.
##  maxit (=6000) reached before log-log convergence.
##  The likelihood and params might not be at the ML values.
##  Try setting control$maxit higher.
## Log-likelihood: -680.6768
## AIC: 1419.354   AICc: 1422.428
##
##          Estimate
## Z.11       0.1791
## Z.21       0.5648
## Z.31       0.0384
## Z.41       0.2589
## Z.51      -0.1526
## Z.22       0.1651
## Z.32      -0.3787
## Z.42      -0.0855
```

```
## Z.52     -0.1694
## Z.33      0.5601
## Z.43      0.2298
## Z.53      0.3460
## Z.44      0.1971
## Z.54     -0.0840
## R.(1,1)   0.2077
## R.(2,1)   0.0217
## R.(3,1)   0.0514
## R.(4,1)   0.1233
## R.(5,1)   0.1251
## R.(2,2)   0.2640
## R.(3,2)   0.0527
## R.(4,2)  -0.0283
## R.(5,2)   0.0783
## R.(3,3)   0.4603
## R.(4,3)  -0.1297
## R.(5,3)  -0.0891
## R.(4,4)   0.3335
## R.(5,4)   0.1079
## R.(5,5)   0.5650
## Initial states (x0) defined at t=0
##
## Standard errors have not been calculated.
## Use MARSSparamCIs to compute CIs and bias estimates.
##
## Convergence warnings
##  Warning: the  Z.31  parameter value has not converged.
##  Warning: the  Z.51  parameter value has not converged.
##  Warning: the  Z.32  parameter value has not converged.
##  Warning: the  Z.42  parameter value has not converged.
##  Warning: the  Z.52  parameter value has not converged.
##  Warning: the  R.(3,1)  parameter value has not converged.
##  Type MARSSinfo("convergence") for more info on this warning.
```

```
round(dfa_2$AICc,1)
```

```
## [1] 1422.4
```

The AICc value for this model is 1422.4, which is about 3.8 units lower than the best model is Q1.

Answer 2b

First, we need to rotate both $\mathbf{Z}$ and $\mathbf{x}$.

```
## get the estimated ZZ
Z_est <- coef(dfa_2, type = "matrix")$Z
```

13

```r
## get the inverse of the rotation matrix
H_inv <- varimax(Z_est)$rotmat
## rotate factor loadings
Z_rot <- Z_est %*% H_inv
## rotate processes
proc_rot <- solve(H_inv) %*% dfa_2$states
```
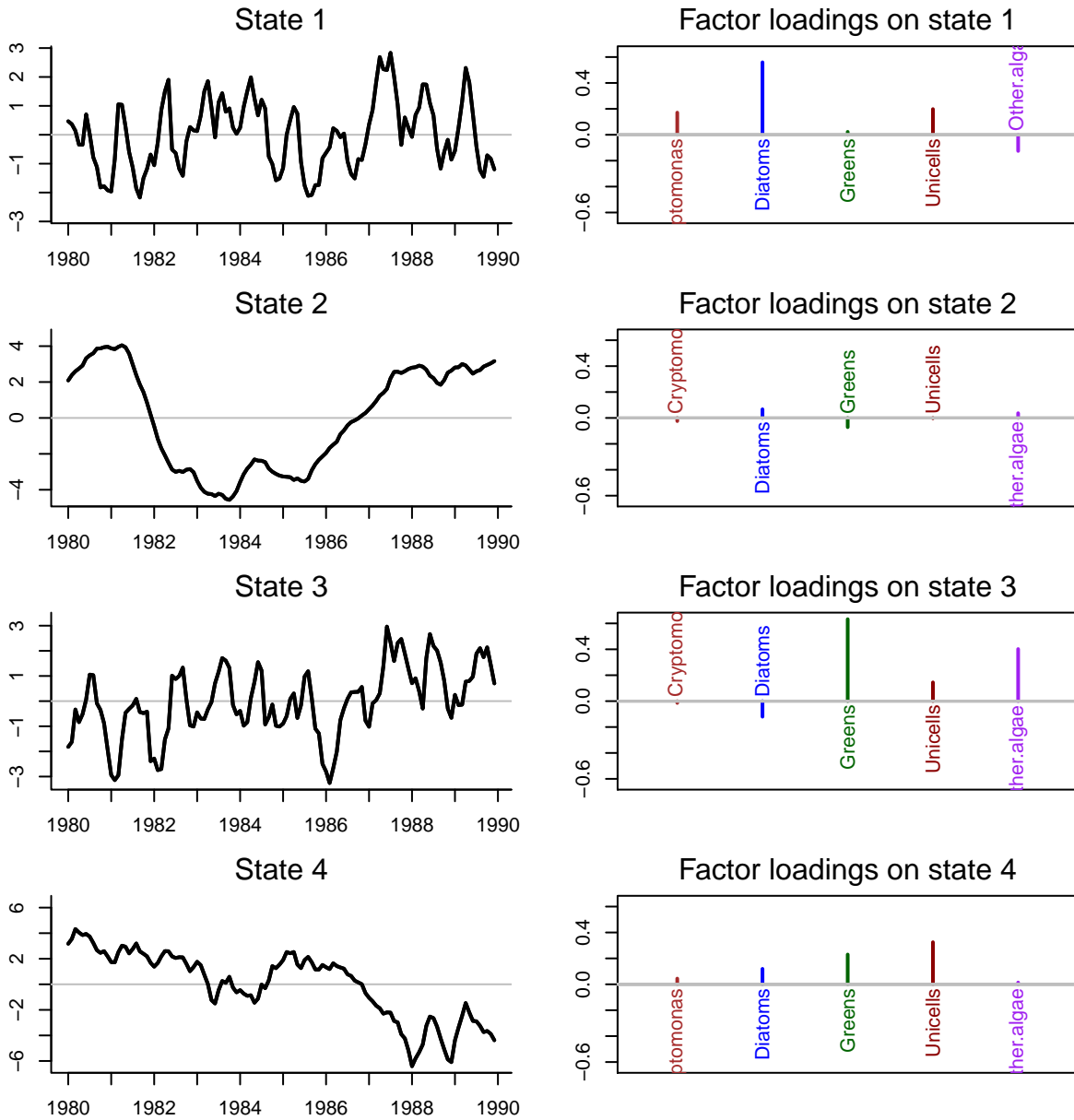
Here are plots of the trends (L) and factor loadings (R).

```r
mm <- dfa_top
par(mfcol = c(mm,2), mai = c(0.3,0.3,0.3,0.3), omi = rep(0.3,4))
## plot the processes
for(i in 1:mm) {
  ylm <- c(-1,1)*max(abs(proc_rot[i,]))
  ## set up plot area
    plot(w_ts,proc_rot[i,], type = "n", bty = "L",
        ylim = ylm, xlab = "", ylab = "", xaxt = "n")
    ## draw zero-line
    abline(h = 0, col = "gray")
    ## plot trend line
    lines(w_ts,proc_rot[i,], lwd = 2)
    lines(w_ts,proc_rot[i,], lwd = 2)
    ## add panel labels
    mtext(paste("State",i), side = 3, line = 0.5)
    axis(1, 12 * (0:dim(dat_1980)[2]) + 1, yr_frst + 0:dim(dat_1980)[2])
}
## plot the loadings
minZ <- 0
ylm <- c(-1, 1) * max(abs(Z_rot))
for(i in 1:mm) {
  plot(c(1:N_ts)[abs(Z_rot[,i]) > minZ], as.vector(Z_rot[abs(Z_rot[,i]) > minZ, i]), type = "h"
        lwd = 2, xlab = "", ylab = "", xaxt = "n", ylim = ylm, xlim = c(0.5, N_ts + 0.5), col =
    for(j in 1:N_ts) {
      if(Z_rot[j,i] > minZ) {text(j, -0.03, ylbl[j], srt = 90, adj = 1, cex = 1, col = clr[j])
      if(Z_rot[j,i] < -minZ) {text(j, 0.03, ylbl[j], srt = 90, adj = 0, cex = 1, col = clr[j])
      abline(h = 0, lwd = 1.5, col = "gray")
      }
  mtext(paste("Factor loadings on state",i),side = 3,line = 0.5)
}
```

Just in Answer 1b, Trends 1 & 2 show obvious seasonal patterns in the data, but appear to be somewhat out of phase with one another. Trend 3 has a quick rise and then gradual decline, but none of the time series load very hevaily on that trend. Trend 4 shows a general increase over time, but the taxa seem to load negatively on it, so that indicates general decrease over time.

Answer 2c

We can plot the fits for our DFA model along with the data. Here are time series of the five phytoplankton groups (points) with the mean of the DFA fits (black line) and the 95% confidence intervals (gray lines).

```
## get model fits & CI's
mod_fit <- get_DFA_fits(dfa_res[[dfa_top]])
```

```r
## plot the fits
ylbl <- phytoplankton
par(mfrow = c(N_ts, 1), mai = c(0.5, 0.7, 0.1, 0.1), omi = rep(0.3, 4))
for(i in 1:N_ts) {
  up <- mod_fit$up[i,]
  mn <- mod_fit$ex[i,]
  lo <- mod_fit$lo[i,]
  plot(w_ts, mn, xlab = "", ylab = ylbl[i], xaxt = "n", type = "n", cex.lab = 1.2,
       ylim = c(min(lo), max(up)))
  axis(1, 12 * (0:dim(dat_1980)[2]) + 1, yr_frst + 0:dim(dat_1980)[2])
  points(w_ts, dat[i,], pch = 16, col = clr[i])
  lines(w_ts, up, col = "darkgray")
  lines(w_ts, mn, col = "black", lwd = 2)
  lines(w_ts, lo, col = "darkgray")
}
```
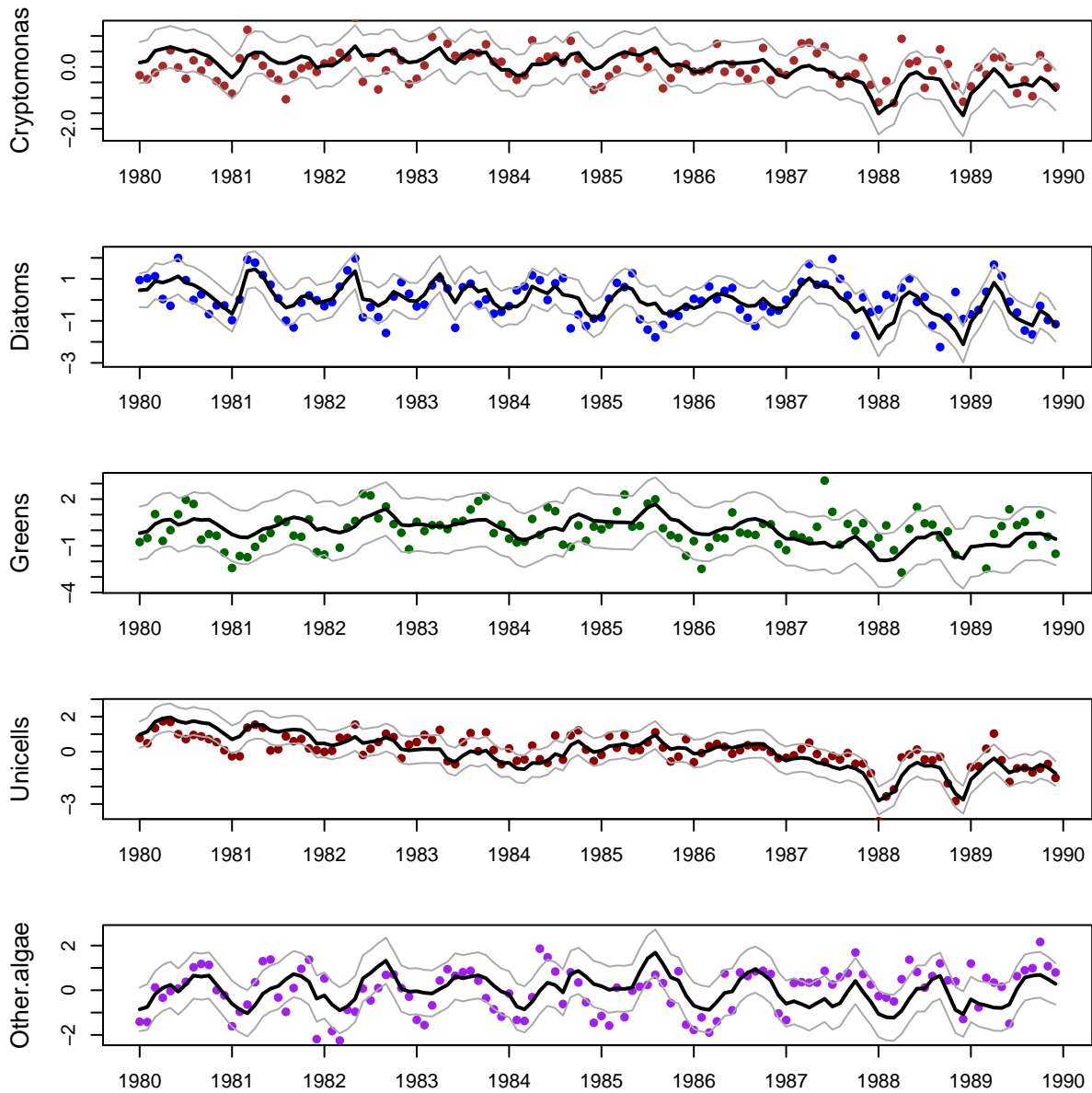
These model fits look pretty good overall as they seem to capture much of the seasonal variation.

Question 3

Fit a DFA model that includes temperature as a covariate and 3 trends (as in Sec 6.1), but with `R = "unconstrained"`? How does this model compare to the model with `R = "diagonal and unequal"`? How does it compare to the model in Q2?

Plot the model states and loadings as in Sec 5.1. Describe the general patterns in the states and the ways the different taxa load onto those trends.

Also plot the the model fits as in Sec 5.2. Do they reasonable? Are there any particular problems or outliers?

Answer 3a

We need to modify the `MARSS` call slightly here to incorporate lake temperature as a covariate in the DFA model.

```
## get temp covariate
Temp <- t(plank_dat[, "Temp", drop = FALSE])
## list with form of var-cov martix for obs errors
mod_list <- list(m = 3, R = "unconstrained")
## list with model control parameters
con_list <- list(maxit = 6000, allow.degen = TRUE)
## list with model inits
init_list <- list(x0 = matrix(rep(0, i), i, 1))
## fit DFA model with Temp covariate
dfa_3 <- MARSS(dat, model = mod_list, form = "dfa", z.score = FALSE, control = con_list,
               covariates = Temp)
```

```
## Success! abstol and log-log tests passed at 1580 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 1580 iterations.
## Log-likelihood: -644.0382
## AIC: 1352.076   AICc: 1355.828
##
##                  Estimate
## Z.11              0.22375
## Z.21              0.58549
## Z.31              0.27197
## Z.41              0.26861
## Z.51              0.10028
## Z.22              0.12986
## Z.32             -0.06680
## Z.42             -0.06968
## Z.52              0.09998
## Z.33              0.05516
## Z.43              0.31078
## Z.53             -0.10989
## R.(1,1)           0.18909
## R.(2,1)          -0.00181
## R.(3,1)          -0.02036
## R.(4,1)           0.11383
## R.(5,1)           0.05301
## R.(2,2)           0.21889
```

```
## R.(3,2)        -0.04949
## R.(4,2)        -0.01637
## R.(5,2)        -0.03875
## R.(3,3)         0.70400
## R.(4,3)        -0.11142
## R.(5,3)         0.01424
## R.(4,4)         0.31260
## R.(5,4)         0.12167
## R.(5,5)         0.55267
## D.Cryptomonas  0.00936
## D.Diatoms      -0.29593
## D.Greens        0.56481
## D.Unicells      0.14072
## D.Other.algae  0.49367
## Initial states (x0) defined at t=0
##
## Standard errors have not been calculated.
## Use MARSSparamCIs to compute CIs and bias estimates.


round(dfa_3$AICc,1)


## [1] 1355.8
```

Here is a table of the model selection results for all of the models.

```
## update AICc values for all models
tbl_aicc_all <- data.frame(trends = c(trnds,4,4),
                           R = c(rep("diag & unequal", 4), rep("unconstrained", 2)),
                           AICc = c(aicc, dfa_2$AICc, dfa_3$AICc))
pander(tbl_aicc_all, caption = "List of AICc values for all of the DFA models.",
       digits = 5)
```

Table 2: List of AICc values for all of the DFA models.

trends	R	AICc
1	diag & unequal	1455.7
2	diag & unequal	1440.5
3	diag & unequal	1427.8
4	diag & unequal	1426.2
4	unconstrained	1422.4
4	unconstrained	1355.8

The DFA model from Q3 that includes the temperature covariate is by far the best of all.
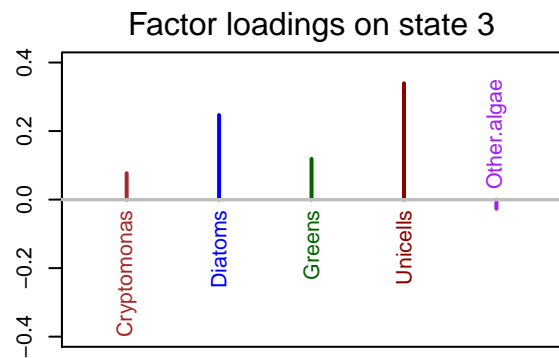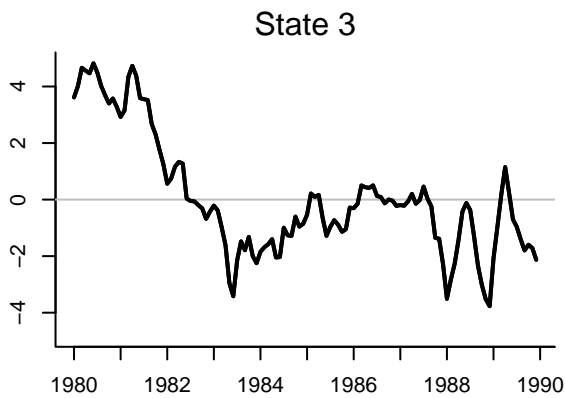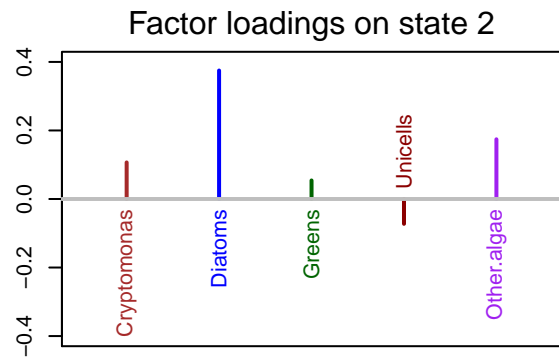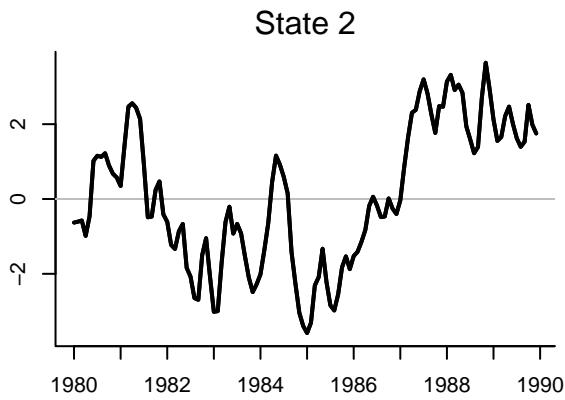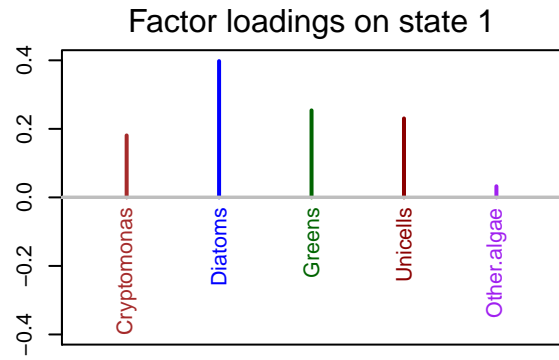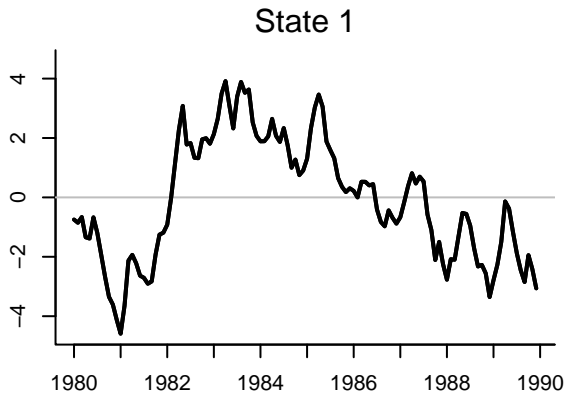
Answer 3b

First, we need to rotate both **Z** and **x**.

```
## get the estimated ZZ
Z_est <- coef(dfa_3, type = "matrix")$Z
## get the inverse of the rotation matrix
H_inv <- varimax(Z_est)$rotmat
## rotate factor loadings
Z_rot <- Z_est %*% H_inv
## rotate processes
proc_rot <- solve(H_inv) %*% dfa_3$states
```

Here are plots of the trends (L) and factor loadings (R).

```
mm <- 3
par(mfcol = c(mm,2), mai = c(0.3,0.3,0.3,0.3), omi = rep(0.3,4))
## plot the processes
for(i in 1:mm) {
  ylm <- c(-1,1)*max(abs(proc_rot[i,]))
  ## set up plot area
    plot(w_ts,proc_rot[i,], type = "n", bty = "L",
         ylim = ylm, xlab = "", ylab = "", xaxt = "n")
    ## draw zero-line
    abline(h = 0, col = "gray")
    ## plot trend line
    lines(w_ts,proc_rot[i,], lwd = 2)
    lines(w_ts,proc_rot[i,], lwd = 2)
    ## add panel labels
    mtext(paste("State",i), side = 3, line = 0.5)
    axis(1, 12 * (0:dim(dat_1980)[2]) + 1, yr_frst + 0:dim(dat_1980)[2])
}
## plot the loadings
minZ <- 0
ylm <- c(-1, 1) * max(abs(Z_rot))
for(i in 1:mm) {
  plot(c(1:N_ts)[abs(Z_rot[,i]) > minZ], as.vector(Z_rot[abs(Z_rot[,i]) > minZ,i]), type = "h"
       lwd = 2, xlab = "", ylab = "", xaxt = "n", ylim = ylm, xlim = c(0.5, N_ts + 0.5), col =
    for(j in 1:N_ts) {
      if(Z_rot[j,i] > minZ) {text(j, -0.03, ylbl[j], srt = 90, adj = 1, cex = 1, col = clr[j])
      if(Z_rot[j,i] < -minZ) {text(j, 0.03, ylbl[j], srt = 90, adj = 0, cex = 1, col = clr[j])
      abline(h = 0, lwd = 1.5, col = "gray")
      }
  mtext(paste("Factor loadings on state",i),side = 3,line = 0.5)
}
```

Note now that trends 1 & 2 do not reflect the dominant seasonal signal that we saw in Q1 and Q2. Trend 3 shows the general decline we saw previously, but now the trend is downward and the loadings are mostly positive.
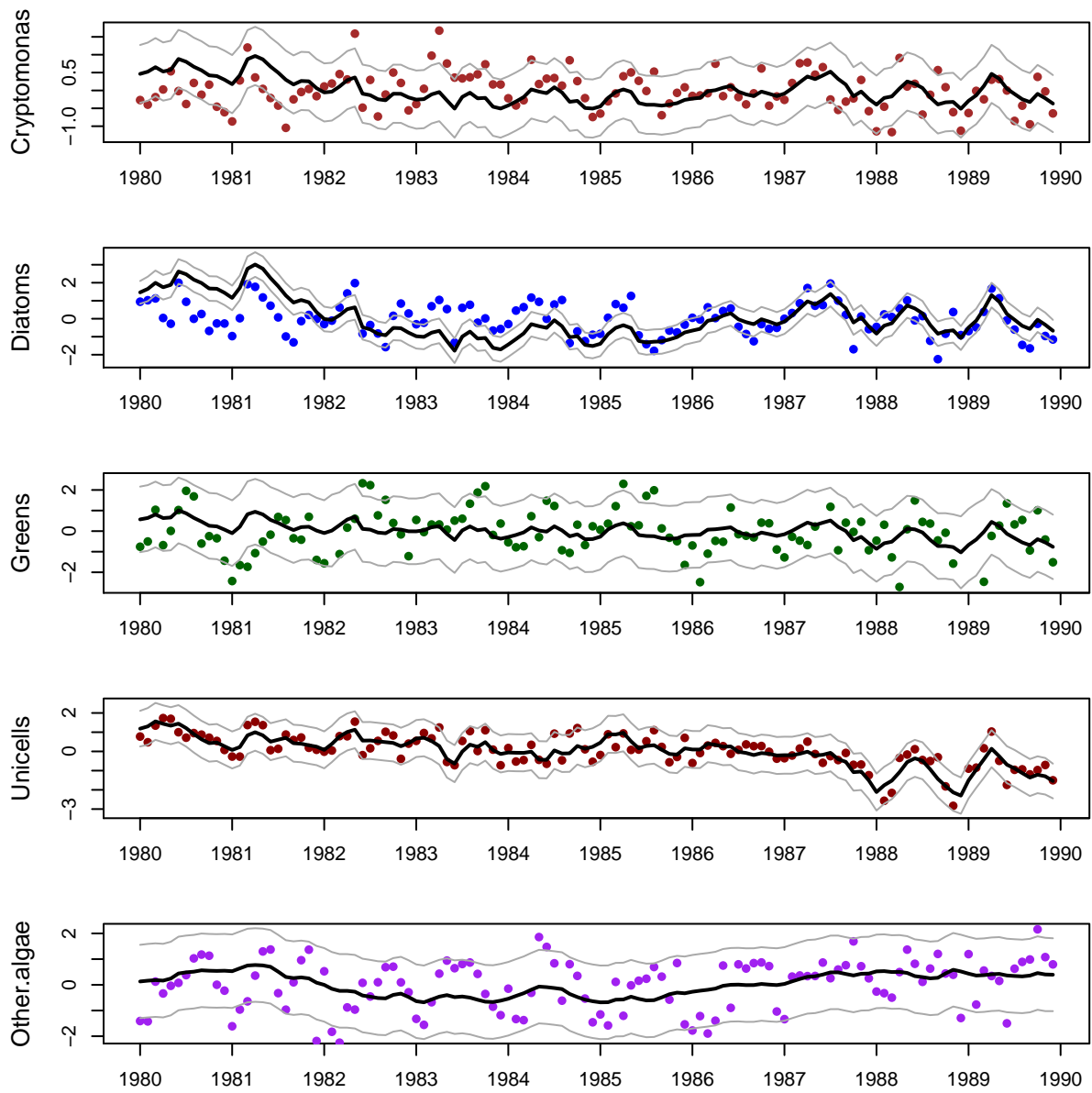
Answer 3c

We can plot the fits for our DFA model along with the data. Here are time series of the five phytoplankton groups (points) with the mean of the DFA fits (black line) and the 95% confidence intervals (gray lines).

```
## get model fits & CI's
mod_fit <- get_DFA_fits(dfa_3)
```

```r
## plot the fits
ylbl <- phytoplankton
par(mfrow = c(N_ts, 1), mai = c(0.5, 0.7, 0.1, 0.1), omi = rep(0.3, 4))
for(i in 1:N_ts) {
  up <- mod_fit$up[i,]
  mn <- mod_fit$ex[i,]
  lo <- mod_fit$lo[i,]
  plot(w_ts, mn, xlab = "", ylab = ylbl[i], xaxt = "n", type = "n", cex.lab = 1.2,
       ylim = c(min(lo), max(up)))
  axis(1, 12 * (0:dim(dat_1980)[2]) + 1, yr_frst + 0:dim(dat_1980)[2])
  points(w_ts,dat[i,], pch = 16, col = clr[i])
  lines(w_ts, up, col = "darkgray")
  lines(w_ts, mn, col = "black", lwd = 2)
  lines(w_ts, lo, col = "darkgray")
}
```

These model fits look pretty good overall as they seem to capture much of the seasonal variation.