2021\_MARSSModel

##Read in data

KSV\_meantemps <- readRDS('KSV\_meantemps.rds')  
daily\_means\_long\_klamath <- readRDS('daily\_means\_long.rds')  
covariate\_klamath <- readRDS('covariate.rds')  
  
daily\_means\_long\_klamath <-rbind(daily\_means\_long\_klamath, KSV = KSV\_meantemps)  
str(daily\_means\_long\_klamath)

##Data Matrix

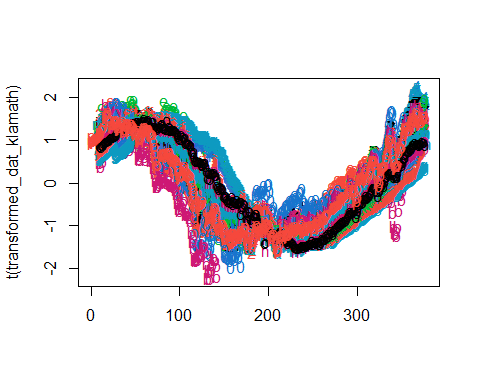
#Convert data to matrix   
daily\_means\_long\_klamath <- as.matrix(daily\_means\_long\_klamath)  
saveRDS(daily\_means\_long\_klamath,"daily\_means\_long\_klamath.rds")  
  
#z score  
transformed\_dat\_klamath <- as.matrix(daily\_means\_long\_klamath)  
transformed\_dat\_klamath <- zscore(transformed\_dat\_klamath)  
saveRDS(transformed\_dat\_klamath,"transformed\_dat\_klamath.rds")

##Covariates

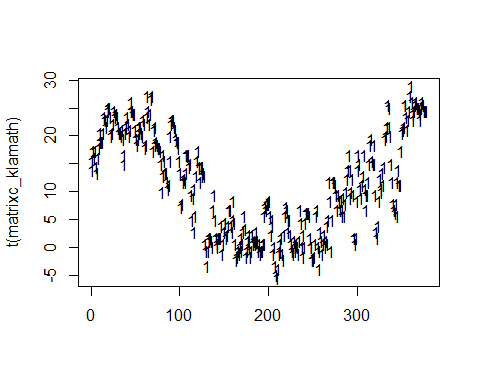
#Build the little c matrix, call it matrixc  
matrixc\_klamath <- matrix(nrow=1,ncol=378)  
matrixc\_klamath <- (as.matrix(covariate\_klamath))  
saveRDS(matrixc\_klamath,"matrixc\_klamath.rds")

###Check data and covariates

matplot(t(transformed\_dat\_klamath))



matplot(t(matrixc\_klamath))



#looks okay, we removed AP2 in the visualization .rmd file

##Z-Matrices ###matrix2\_klamath

#Hypothesis 1: All ponds and creeks and mainstem are separate  
matrix2\_klamath <- matrix(nrow=26,ncol=12)  
matrix2\_klamath[c(1:2),1] <- 1 #Alexander  
matrix2\_klamath[c(1:2),c(2:12)] <- 0  
matrix2\_klamath[c(4:5),2] <- 1 #Stender  
matrix2\_klamath[c(4:5),c(1,3:12)] <- 0   
matrix2\_klamath[c(3,6,9,11,12),3] <- 1 #Seiad Creek  
matrix2\_klamath[c(3,6,9,11,12),c(1,2,4:12)] <- 0  
matrix2\_klamath[7,4] <- 1 #Durazo  
matrix2\_klamath[7,c(1:3,5:12)] <- 0  
matrix2\_klamath[8,5] <- 1 #Lower Seiad  
matrix2\_klamath[8,c(1:4,6:12)] <- 0  
matrix2\_klamath[10,6] <- 1 #May  
matrix2\_klamath[10,c(1:5,7:12)] <- 0  
matrix2\_klamath[c(12:13),7] <- 1 #Fish Gulch  
matrix2\_klamath[c(12:13),c(1:6,8:12)] <- 0  
matrix2\_klamath[c(15:17),8] <- 1 #Goodman  
matrix2\_klamath[c(15:17),c(1:7,9:12)] <- 0  
matrix2\_klamath[c(19:21),9] <- 1 #Upper Lawrence  
matrix2\_klamath[c(19:21),c(1:8,10:12)] <- 0  
matrix2\_klamath[c(23:25),10] <- 1 #Lower Lawrence  
matrix2\_klamath[c(23:25),c(1:9,11:12)] <- 0  
matrix2\_klamath[c(14,18,22),11] <- 1 #Horse Creek  
matrix2\_klamath[c(14,18,22),c(1:10,12)] <- 0  
matrix2\_klamath[26,12] <- 1 #Klamath  
matrix2\_klamath[26,c(1:11)] <- 0  
matrix2\_klamath

## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]  
## [1,] 1 0 0 0 0 0 0 0 0 0 0 0  
## [2,] 1 0 0 0 0 0 0 0 0 0 0 0  
## [3,] 0 0 1 0 0 0 0 0 0 0 0 0  
## [4,] 0 1 0 0 0 0 0 0 0 0 0 0  
## [5,] 0 1 0 0 0 0 0 0 0 0 0 0  
## [6,] 0 0 1 0 0 0 0 0 0 0 0 0  
## [7,] 0 0 0 1 0 0 0 0 0 0 0 0  
## [8,] 0 0 0 0 1 0 0 0 0 0 0 0  
## [9,] 0 0 1 0 0 0 0 0 0 0 0 0  
## [10,] 0 0 0 0 0 1 0 0 0 0 0 0  
## [11,] 0 0 1 0 0 0 0 0 0 0 0 0  
## [12,] 0 0 0 0 0 0 1 0 0 0 0 0  
## [13,] 0 0 0 0 0 0 1 0 0 0 0 0  
## [14,] 0 0 0 0 0 0 0 0 0 0 1 0  
## [15,] 0 0 0 0 0 0 0 1 0 0 0 0  
## [16,] 0 0 0 0 0 0 0 1 0 0 0 0  
## [17,] 0 0 0 0 0 0 0 1 0 0 0 0  
## [18,] 0 0 0 0 0 0 0 0 0 0 1 0  
## [19,] 0 0 0 0 0 0 0 0 1 0 0 0  
## [20,] 0 0 0 0 0 0 0 0 1 0 0 0  
## [21,] 0 0 0 0 0 0 0 0 1 0 0 0  
## [22,] 0 0 0 0 0 0 0 0 0 0 1 0  
## [23,] 0 0 0 0 0 0 0 0 0 1 0 0  
## [24,] 0 0 0 0 0 0 0 0 0 1 0 0  
## [25,] 0 0 0 0 0 0 0 0 0 1 0 0  
## [26,] 0 0 0 0 0 0 0 0 0 0 0 1

###matrix3\_klamath

#Hypothesis 2: ponds versus creeks versus Klamath  
matrix3\_klamath <- matrix(nrow=26, ncol=3)  
matrix3\_klamath[c(1:2,4:5,7:8,10,12:13,15:17,19:21,23:25),1] <- 1 #All ponds  
matrix3\_klamath[c(1:2,4:5,7:8,10,12:13,15:17,19:21,23:25),c(2:3)] <- 0  
matrix3\_klamath[c(3,6,9,11,14,18,22),c(1,3)] <- 0 #All creeks  
matrix3\_klamath[c(3,6,9,11,14,18,22),2] <- 1  
matrix3\_klamath[26,3] <- 1 #Klamath  
matrix3\_klamath[26,c(1:2)] <- 0   
matrix3\_klamath

## [,1] [,2] [,3]  
## [1,] 1 0 0  
## [2,] 1 0 0  
## [3,] 0 1 0  
## [4,] 1 0 0  
## [5,] 1 0 0  
## [6,] 0 1 0  
## [7,] 1 0 0  
## [8,] 1 0 0  
## [9,] 0 1 0  
## [10,] 1 0 0  
## [11,] 0 1 0  
## [12,] 1 0 0  
## [13,] 1 0 0  
## [14,] 0 1 0  
## [15,] 1 0 0  
## [16,] 1 0 0  
## [17,] 1 0 0  
## [18,] 0 1 0  
## [19,] 1 0 0  
## [20,] 1 0 0  
## [21,] 1 0 0  
## [22,] 0 1 0  
## [23,] 1 0 0  
## [24,] 1 0 0  
## [25,] 1 0 0  
## [26,] 0 0 1

###matrix4\_klamath

#Hypothesis 3: tributary versus tributary versus Klamath  
matrix4\_klamath <-matrix(nrow=26,ncol=5)  
matrix4\_klamath[c(1:2,4:5,7:8,10),1] <- 1 #Seiad Creek Ponds  
matrix4\_klamath[c(1:2,4:5,7:8,10),c(2:5)] <- 0  
matrix4\_klamath[c(3,6,9,11),2] <- 1 #Seiad Creek  
matrix4\_klamath[c(3,6,9,11),c(1,3:5)] <- 0  
matrix4\_klamath[c(12:13,15:17,19:21,23:25),3] <- 1 #Horse Creek Ponds  
matrix4\_klamath[c(12:13,15:17,19:21,23:25),c(1,2,4,5)] <- 0  
matrix4\_klamath[c(14,18,22),4] <- 1 #Horse Creek  
matrix4\_klamath[c(14,18,22),c(1:3,5)] <- 0  
matrix4\_klamath[26,5] <- 1 #Klamath  
matrix4\_klamath[c(26),c(1:4)] <- 0  
matrix4\_klamath

## [,1] [,2] [,3] [,4] [,5]  
## [1,] 1 0 0 0 0  
## [2,] 1 0 0 0 0  
## [3,] 0 1 0 0 0  
## [4,] 1 0 0 0 0  
## [5,] 1 0 0 0 0  
## [6,] 0 1 0 0 0  
## [7,] 1 0 0 0 0  
## [8,] 1 0 0 0 0  
## [9,] 0 1 0 0 0  
## [10,] 1 0 0 0 0  
## [11,] 0 1 0 0 0  
## [12,] 0 0 1 0 0  
## [13,] 0 0 1 0 0  
## [14,] 0 0 0 1 0  
## [15,] 0 0 1 0 0  
## [16,] 0 0 1 0 0  
## [17,] 0 0 1 0 0  
## [18,] 0 0 0 1 0  
## [19,] 0 0 1 0 0  
## [20,] 0 0 1 0 0  
## [21,] 0 0 1 0 0  
## [22,] 0 0 0 1 0  
## [23,] 0 0 1 0 0  
## [24,] 0 0 1 0 0  
## [25,] 0 0 1 0 0  
## [26,] 0 0 0 0 1

###matrix5\_klamath

#Hypothesis 4: All sensors are the same  
matrix5\_klamath <- matrix(nrow=26, ncol=1)  
matrix5\_klamath[,] <- 1  
matrix5\_klamath

## [,1]  
## [1,] 1  
## [2,] 1  
## [3,] 1  
## [4,] 1  
## [5,] 1  
## [6,] 1  
## [7,] 1  
## [8,] 1  
## [9,] 1  
## [10,] 1  
## [11,] 1  
## [12,] 1  
## [13,] 1  
## [14,] 1  
## [15,] 1  
## [16,] 1  
## [17,] 1  
## [18,] 1  
## [19,] 1  
## [20,] 1  
## [21,] 1  
## [22,] 1  
## [23,] 1  
## [24,] 1  
## [25,] 1  
## [26,] 1

saveRDS(matrix2\_klamath,"matrix2\_klamath.rds")  
saveRDS(matrix3\_klamath,"matrix3\_klamath.rds")  
saveRDS(matrix4\_klamath,"matrix4\_klamath.rds")  
saveRDS(matrix5\_klamath,"matrix5\_klamath.rds")

##MARSS models ###Create a FT for seasonality and combine with AirTemp

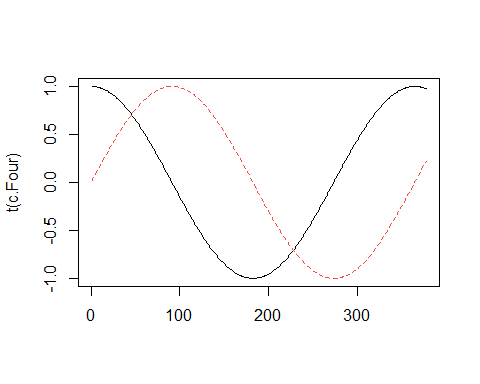
#Correct for seasonality using Fourier Series  
TT = ncol(transformed\_dat\_klamath) # number of time periods/samples  
period = 365 # number of "seasons" (e.g., 12 months per year)  
per.1st = 182 # first "season" (e.g., Jan = 1, July = 7)  
c = diag(period) # create factors for seasons  
for(i in 2:(ceiling(TT/period))) {c = cbind(c,diag(period))}  
dim(c)

## [1] 365 730

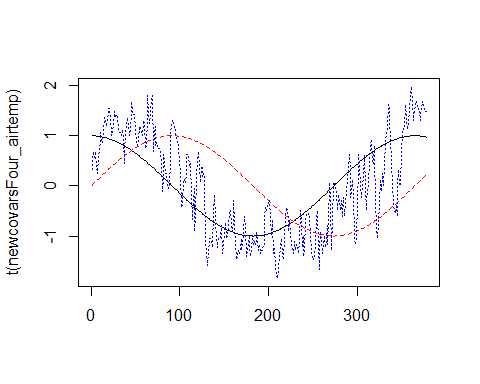
#Create Fourier Series  
cos.t = cos(2 \* pi \* seq(TT) / period)  
sin.t = sin(2 \* pi \* seq(TT) / period)  
c.Four = rbind(cos.t,sin.t)  
cor(c.Four[1,],c.Four[2,]) # not correlated!

## [1] 0.007872561

matplot(t(c.Four), type="l")



#Now fit model with seasonality AND an additional covariate (airtemp from above)  
matrixc\_klamath\_z <- zscore(matrixc\_klamath)  
newcovarsFour\_airtemp <-rbind(c.Four, "airtemp"=matrixc\_klamath\_z)  
matplot(t(newcovarsFour\_airtemp), type="l", col=c("black","red","blue"))



####Things I tried: 1) zscore data + covar; 2) no zscore; 3) zscore covar; 4) zscore data; 5) no covar at all; 6) zscore covar + FT; 6) just FT

###model 1

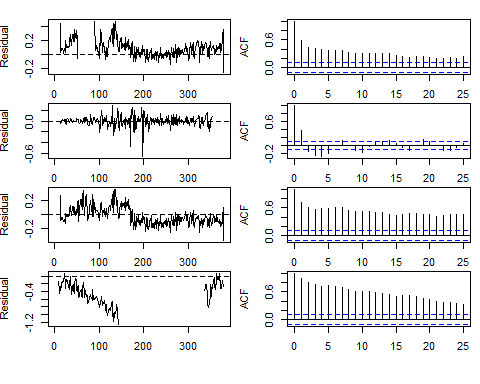
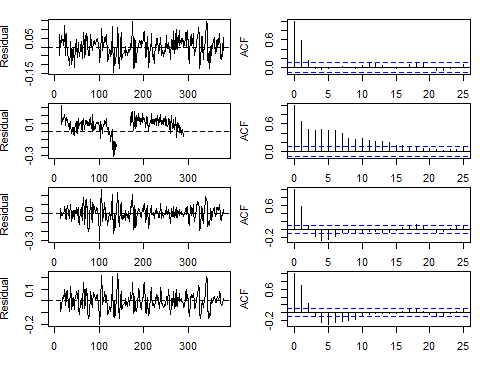
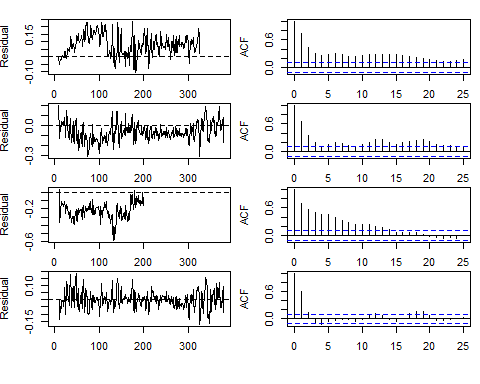
#Hypothesis 1, Model 1: all separate  
mod11\_klamath = list()  
mod11\_klamath$A = "zero" #no trend because we z scored  
mod11\_klamath$Z = matrix2\_klamath  
mod11\_klamath$R = "diagonal and equal" #all the sensors are same, so observation error should be same  
mod11\_klamath$Q = "diagonal and unequal"   
mod11\_klamath$B = "identity" #assuming no species interactions  
mod11\_klamath$U = "zero" #no trend because we z scored   
mod11\_klamath$C = "unequal" #Can set C to unequal because it is going off the Z matrix where I have already indicated how to split up the sites.  
mod11\_klamath$c = matrixc\_klamath  
mod11\_klamath.fit = MARSS(transformed\_dat\_klamath, model=mod11\_klamath, control=list(maxit=10000))

## Success! abstol and log-log tests passed at 65 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 65 iterations.   
## Log-likelihood: 1566.017   
## AIC: -3058.035 AICc: -3057.714   
##   
## Estimate  
## R.diag 3.09e-02  
## Q.(X1,X1) 5.35e-03  
## Q.(X2,X2) 7.28e-03  
## Q.(X3,X3) 7.68e-03  
## Q.(X4,X4) 6.63e-03  
## Q.(X5,X5) 5.44e-03  
## Q.(X6,X6) 1.62e-02  
## Q.(X7,X7) 4.93e-03  
## Q.(X8,X8) 1.77e-03  
## Q.(X9,X9) 2.59e-03  
## Q.(X10,X10) 2.21e-03  
## Q.(X11,X11) 1.00e-02  
## Q.(X12,X12) 5.10e-03  
## x0.X1 1.39e+00  
## x0.X2 9.89e-01  
## x0.X3 1.10e+00  
## x0.X4 1.22e+00  
## x0.X5 1.54e+00  
## x0.X6 1.42e+00  
## x0.X7 8.31e-01  
## x0.X8 6.65e-01  
## x0.X9 1.03e+00  
## x0.X10 1.01e+00  
## x0.X11 1.05e+00  
## x0.X12 9.96e-01  
## C.X1 2.05e-04  
## C.X2 4.26e-04  
## C.X3 4.58e-04  
## C.X4 2.04e-04  
## C.X5 -9.02e-05  
## C.X6 9.00e-05  
## C.X7 2.43e-04  
## C.X8 5.48e-04  
## C.X9 3.73e-04  
## C.X10 3.66e-04  
## C.X11 6.39e-04  
## C.X12 2.62e-04  
## Initial states (x0) defined at t=0  
##   
## Standard errors have not been calculated.   
## Use MARSSparamCIs to compute CIs and bias estimates.

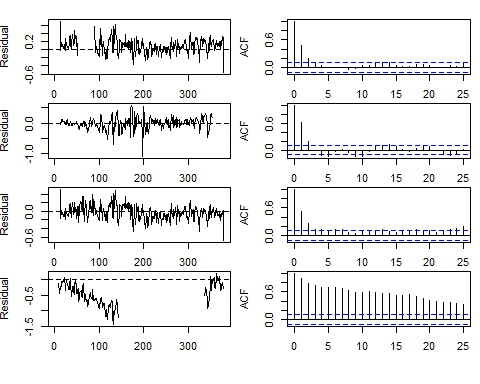
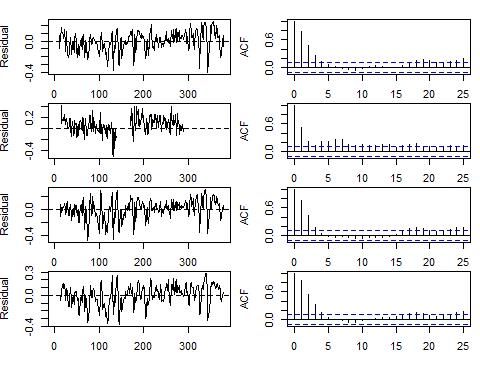
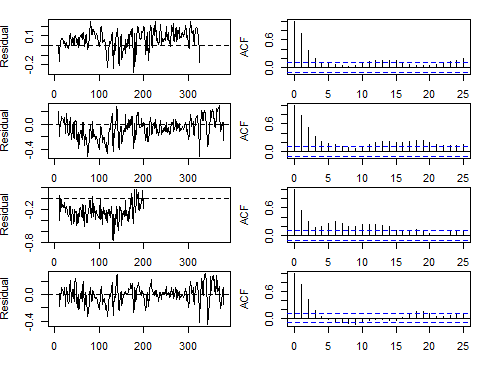
mod11\_klamath.params = MARSSparamCIs(mod11\_klamath.fit)  
saveRDS(mod11\_klamath.fit,"mod11\_klamath.fit.rds")  
saveRDS(mod11\_klamath.params,"mod11\_klamath.params.rds")

####Messing with Residuals

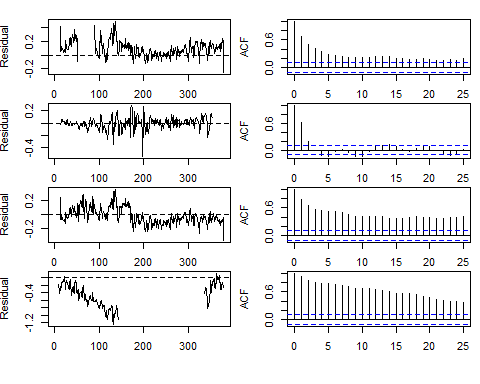
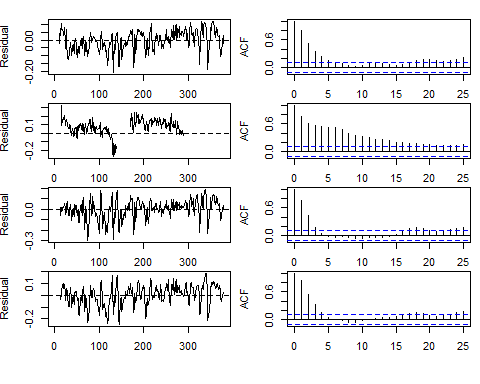
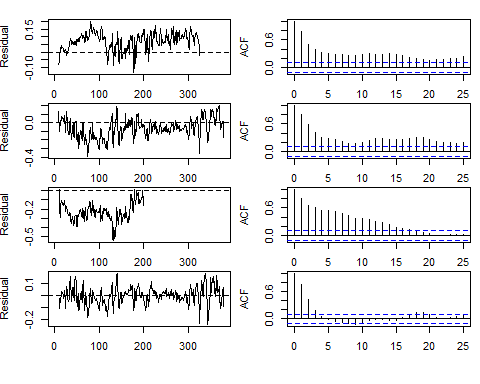
mod11\_klamath.fit <- readRDS("mod11\_klamath.fit.rds")  
mod11\_klamath.params <- readRDS("mod11\_klamath.params.rds")  
  
#model.residuals = the model residuals (data minus model predicted values) as a n x T matrix  
  
#tT: smoothed residuals conditioned on all the data t=1 to T, aka smoothation residuals \*Smoothed residuals are autocorrelated so an ACF test would not reveal model inadequacy  
par(mfrow=c(4,2), mai=c(0.1,0.5,0.2,0.1),omi=c(0.5,0,0,0))  
for (j in 1:12) {  
 plot.ts(residuals<-MARSSresiduals(mod11\_klamath.fit, type = "tT")$model.residuals[j, ],  
 ylab = "Residual")  
 abline(h = 0, lty = "dashed")  
 acf(residuals,na.action = na.pass)  
}



#tt1: one-step-ahead residuals, aka innovations residuals  
par(mfrow=c(4,2), mai=c(0.1,0.5,0.2,0.1),omi=c(0.5,0,0,0))  
for (j in 1:12) {  
 plot.ts(residuals<-MARSSresiduals(mod11\_klamath.fit, type = "tt1")$model.residuals[j, ],  
 ylab = "Residual")  
 abline(h = 0, lty = "dashed")  
 acf(residuals,na.action = na.pass)  
}



#tt: contemporaneous residuals, only for the observations  
par(mfrow=c(4,2), mai=c(0.1,0.5,0.2,0.1),omi=c(0.5,0,0,0))  
for (j in 1:12) {  
 plot.ts(residuals<-MARSSresiduals(mod11\_klamath.fit, type = "tt")$model.residuals[j, ],  
 ylab = "Residual")  
 abline(h = 0, lty = "dashed")  
 acf(residuals,na.action = na.pass)  
}

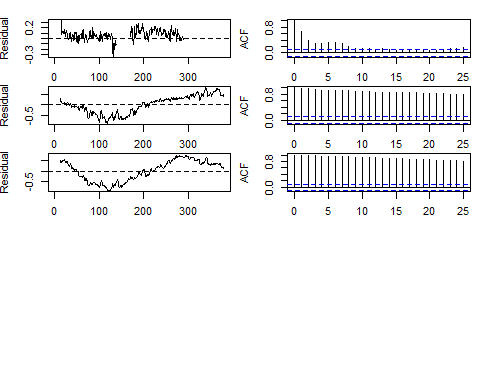
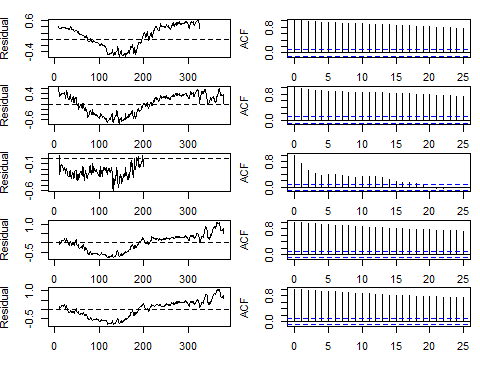


###model 2

#Hypothesis 2, Model 3: ponds vs. creeks   
mod12\_klamath <- mod11\_klamath   
mod12\_klamath$Z <- matrix3\_klamath  
mod12\_klamath$c <- matrixc\_klamath  
mod12\_klamath.fit = MARSS(transformed\_dat\_klamath, model=mod12\_klamath, control=list(maxit=10000))

## Success! abstol and log-log tests passed at 48 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 48 iterations.   
## Log-likelihood: -4126.332   
## AIC: 8272.665 AICc: 8272.69   
##   
## Estimate  
## R.diag 0.141571  
## Q.(X1,X1) 0.002861  
## Q.(X2,X2) 0.006578  
## Q.(X3,X3) 0.004944  
## x0.X1 0.972943  
## x0.X2 1.090202  
## x0.X3 1.034314  
## C.X1 0.000368  
## C.X2 0.000429  
## C.X3 0.000158  
## Initial states (x0) defined at t=0  
##   
## Standard errors have not been calculated.   
## Use MARSSparamCIs to compute CIs and bias estimates.

mod12\_klamath.params = MARSSparamCIs(mod12\_klamath.fit)  
saveRDS(mod12\_klamath.fit,"mod12\_klamath.fit.rds")  
saveRDS(mod12\_klamath.params,"mod12\_klamath.params.rds")  
  
par(mfrow=c(5,2), mai=c(0.1,0.5,0.2,0.1), omi=c(0.5,0,0,0))  
 for (j in 1:8) {  
 plot.ts(residuals<-MARSSresiduals(mod12\_klamath.fit, type = "tt")$model.residuals[j, ],  
 ylab = "Residual")  
 abline(h = 0, lty = "dashed")  
 acf(residuals,na.action = na.pass)  
 }

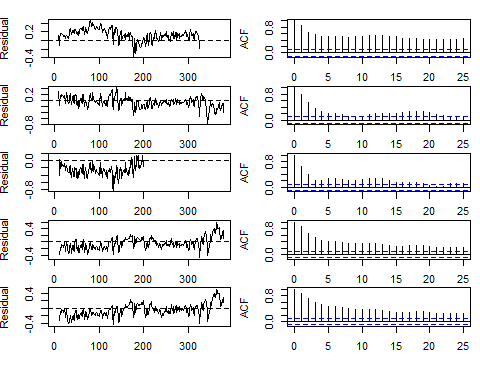


###model 3

#Hypothesis 3, Model 3: Tributary versus tributary  
mod13\_klamath <- mod11\_klamath  
mod13\_klamath$Z = matrix4\_klamath  
mod13\_klamath.fit = MARSS(transformed\_dat\_klamath, model=mod13\_klamath, control=list(maxit=10000))

## Success! abstol and log-log tests passed at 53 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 53 iterations.   
## Log-likelihood: -1045.504   
## AIC: 2123.008 AICc: 2123.069   
##   
## Estimate  
## R.diag 0.066270  
## Q.(X1,X1) 0.006797  
## Q.(X2,X2) 0.005902  
## Q.(X3,X3) 0.001860  
## Q.(X4,X4) 0.008371  
## Q.(X5,X5) 0.004792  
## x0.X1 1.259193  
## x0.X2 1.138327  
## x0.X3 0.939603  
## x0.X4 1.067854  
## x0.X5 1.007397  
## C.X1 0.000275  
## C.X2 0.000357  
## C.X3 0.000390  
## C.X4 0.000519  
## C.X5 0.000202  
## Initial states (x0) defined at t=0  
##   
## Standard errors have not been calculated.   
## Use MARSSparamCIs to compute CIs and bias estimates.

mod13\_klamath.params = MARSSparamCIs(mod13\_klamath.fit)  
saveRDS(mod13\_klamath.fit,"mod13\_klamath.fit.rds")  
saveRDS(mod13\_klamath.params,"mod13\_klamath.params.rds")  
  
par(mfrow=c(5,2), mai=c(0.1,0.5,0.2,0.1), omi=c(0.5,0,0,0))  
 for (j in 1:5) {  
 plot.ts(residuals<-MARSSresiduals(mod13\_klamath.fit, type = "tt1")$model.residuals[j, ],  
 ylab = "Residual")  
 abline(h = 0, lty = "dashed")  
 acf(residuals,na.action = na.pass)  
 }

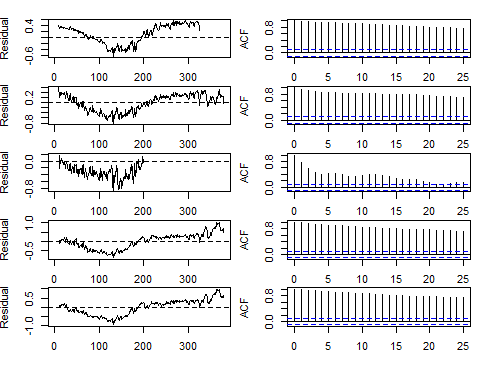


###model 4

#Hypothesis 4, Model 4: All same  
mod14\_klamath <- mod11\_klamath  
mod14\_klamath$Z <- matrix5\_klamath  
mod14\_klamath.fit = MARSS(transformed\_dat\_klamath, model=mod14\_klamath, control=list(maxit=10000))

## Success! abstol and log-log tests passed at 21 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 21 iterations.   
## Log-likelihood: -4232.919   
## AIC: 8473.839 AICc: 8473.843   
##   
## Estimate  
## R.diag 0.148267  
## Q.Q 0.003658  
## x0.x0 0.975406  
## C.C 0.000435  
## Initial states (x0) defined at t=0  
##   
## Standard errors have not been calculated.   
## Use MARSSparamCIs to compute CIs and bias estimates.

mod14\_klamath.params =MARSSparamCIs(mod14\_klamath.fit)  
saveRDS(mod14\_klamath.fit,"mod14\_klamath.fit.rds")  
saveRDS(mod14\_klamath.params,"mod14\_klamath.params.rds")  
  
par(mfrow=c(5,2), mai=c(0.1,0.5,0.2,0.1), omi=c(0.5,0,0,0))  
 for (j in 1:5) {  
 plot.ts(residuals<-MARSSresiduals(mod14\_klamath.fit, type = "tt1")$model.residuals[j, ],  
 ylab = "Residual")  
 abline(h = 0, lty = "dashed")  
 acf(residuals,na.action = na.pass)  
 }



##AICc

data.frame(Model=c("Model11\_klamath", "Model12\_klamath", "Model13\_klamath", "Model14\_klamath"),  
 AICc=round(c(mod11\_klamath.fit$AICc,  
 mod12\_klamath.fit$AICc,  
 mod13\_klamath.fit$AICc,  
 mod14\_klamath.fit$AICc),1))

## Model AICc  
## 1 Model11\_klamath -3057.7  
## 2 Model12\_klamath 8272.7  
## 3 Model13\_klamath 2123.1  
## 4 Model14\_klamath 8473.8