2021\_MARSSModel\_Condense

library(MARSS)

## Warning: package 'MARSS' was built under R version 4.0.5

library(xtable)

## Warning: package 'xtable' was built under R version 4.0.5

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.0.5

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.5 v dplyr 1.0.7  
## v tidyr 1.1.4 v stringr 1.4.0  
## v readr 2.0.2 v forcats 0.5.1

## Warning: package 'ggplot2' was built under R version 4.0.5

## Warning: package 'tibble' was built under R version 4.0.5

## Warning: package 'tidyr' was built under R version 4.0.5

## Warning: package 'readr' was built under R version 4.0.5

## Warning: package 'purrr' was built under R version 4.0.5

## Warning: package 'dplyr' was built under R version 4.0.5

## Warning: package 'forcats' was built under R version 4.0.5

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

#Read in data

#Matrix Klamath + pond dataset  
daily\_means\_long\_klamath <- readRDS('daily\_means\_long\_klamath.rds')  
transformed\_dat\_klamath <- readRDS('transformed\_dat\_klamath.rds')  
transformed\_dat\_klamath\_df <- as.data.frame(t(transformed\_dat\_klamath))  
str(transformed\_dat\_klamath)  
  
#Matrix Airtemp dataset  
covariate\_klamath <- readRDS('covariate.rds')  
transformed\_covariate\_klamath <- zscore(covariate\_klamath)

# Steps:

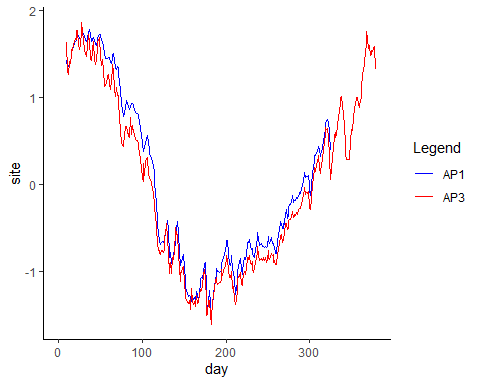
# 1) Among sensor replicates, drop time series with (many) gaps.

# 2) Then take mean across replicate sensors, so that we end up with 1 time series per habitat, 2+9+1 (12 x 378)

#AP  
color <- c("AP1" = "blue", "AP3" = "red")  
transformed\_dat\_klamath\_df %>% rowid\_to\_column(var = "day") %>% ggplot()+  
 geom\_line(aes(x = day, y = AP1, color = "AP1")) +  
 geom\_line(aes(x = day, y = AP3, color = "AP3"))+  
 labs(x = "day", y = "site", color = "Legend")+  
 theme\_classic()+  
 scale\_color\_manual(values = color, labels = c("AP1","AP3"))

## Warning: Removed 63 row(s) containing missing values (geom\_path).

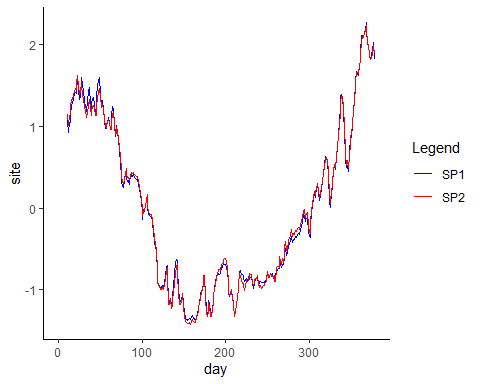
## Warning: Removed 9 row(s) containing missing values (geom\_path).



#need to remove AP1

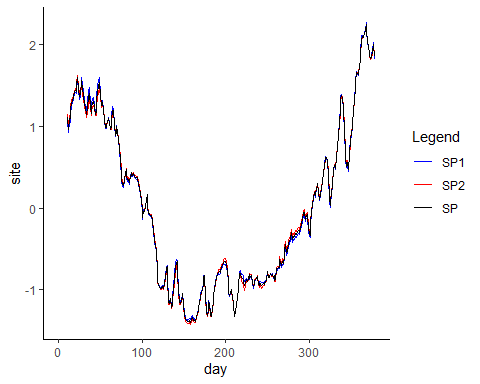
#SP  
color <- c("SP1" = "blue", "SP2" = "red")  
transformed\_dat\_klamath\_df %>% rowid\_to\_column(var = "day") %>% ggplot()+  
 geom\_line(aes(x = day, y = SP1, color = "SP1")) +  
 geom\_line(aes(x = day, y = SP2, color = "SP2"))+  
 labs(x = "day", y = "site", color = "Legend")+  
 theme\_classic()+  
 scale\_color\_manual(values = color, labels = c("SP1","SP2"))

## Warning: Removed 10 row(s) containing missing values (geom\_path).  
  
## Warning: Removed 10 row(s) containing missing values (geom\_path).



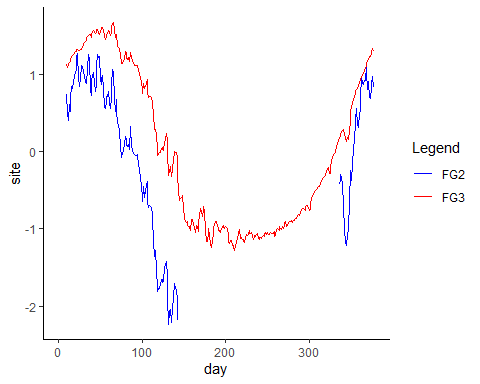
#both good, need to take average  
transformed\_dat\_klamath\_df$SP <- rowMeans(transformed\_dat\_klamath\_df[,c('SP1', 'SP2')], na.rm=TRUE)  
  
color <- c("SP1" = "blue", "SP2" = "red", SP = "black")  
transformed\_dat\_klamath\_df %>% rowid\_to\_column(var = "day") %>% ggplot()+  
 geom\_line(aes(x = day, y = SP1, color = "SP1")) +  
 geom\_line(aes(x = day, y = SP2, color = "SP2"))+  
 geom\_line(aes(x = day, y = SP, color = "SP"))+  
 labs(x = "day", y = "site", color = "Legend")+  
 theme\_classic()+  
 scale\_color\_manual(values = color, labels = c("SP1","SP2", "SP"))

## Warning: Removed 10 row(s) containing missing values (geom\_path).  
  
## Warning: Removed 10 row(s) containing missing values (geom\_path).  
  
## Warning: Removed 10 row(s) containing missing values (geom\_path).



#FG  
color <- c("FG2" = "blue", "FG3" = "red")  
transformed\_dat\_klamath\_df %>% rowid\_to\_column(var = "day") %>% ggplot()+  
 geom\_line(aes(x = day, y = FG2, color = "FG2")) +  
 geom\_line(aes(x = day, y = FG3, color = "FG3"))+  
 labs(x = "day", y = "site", color = "Legend")+  
 theme\_classic()+  
 scale\_color\_manual(values = color, labels = c("FG2","FG3"))

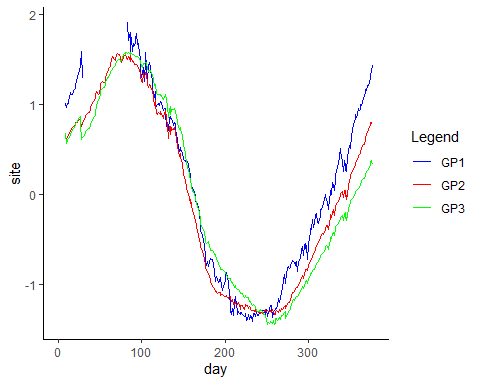
## Warning: Removed 10 row(s) containing missing values (geom\_path).  
  
## Warning: Removed 10 row(s) containing missing values (geom\_path).



#need to remove FG2

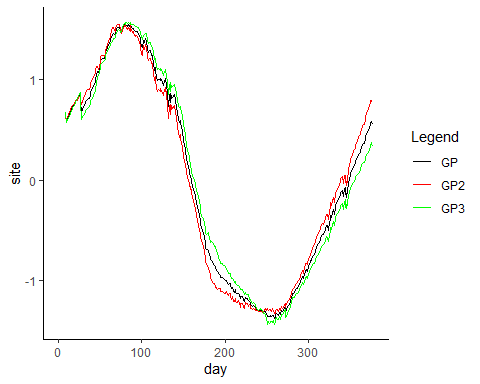
#GP  
color <- c("GP1" = "blue", "GP2" = "red", "GP3" = "green")  
transformed\_dat\_klamath\_df %>% rowid\_to\_column(var = "day") %>% ggplot()+  
 geom\_line(aes(x = day, y = GP1, color = "GP1")) +  
 geom\_line(aes(x = day, y = GP2, color = "GP2"))+  
 geom\_line(aes(x = day, y = GP3, color = "GP3"))+  
 labs(x = "day", y = "site", color = "Legend")+  
 theme\_classic()+  
 scale\_color\_manual(values = color, labels = c("GP1","GP2","GP3"))

## Warning: Removed 9 row(s) containing missing values (geom\_path).  
  
## Warning: Removed 9 row(s) containing missing values (geom\_path).  
  
## Warning: Removed 9 row(s) containing missing values (geom\_path).



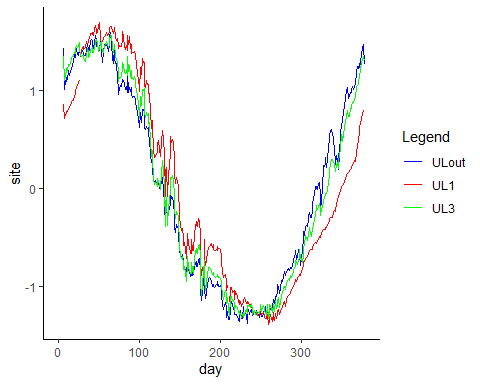
#need to remove GP1  
transformed\_dat\_klamath\_df$GP <- rowMeans(transformed\_dat\_klamath\_df[,c('GP2', 'GP3')], na.rm=TRUE)  
  
color <- c("GP" = "black", "GP2" = "red", "GP3" = "green")  
transformed\_dat\_klamath\_df %>% rowid\_to\_column(var = "day") %>% ggplot()+  
 geom\_line(aes(x = day, y = GP, color = "GP")) +  
 geom\_line(aes(x = day, y = GP2, color = "GP2"))+  
 geom\_line(aes(x = day, y = GP3, color = "GP3"))+  
 labs(x = "day", y = "site", color = "Legend")+  
 theme\_classic()+  
 scale\_color\_manual(values = color, labels = c("GP","GP2","GP3"))

## Warning: Removed 9 row(s) containing missing values (geom\_path).  
  
## Warning: Removed 9 row(s) containing missing values (geom\_path).  
  
## Warning: Removed 9 row(s) containing missing values (geom\_path).



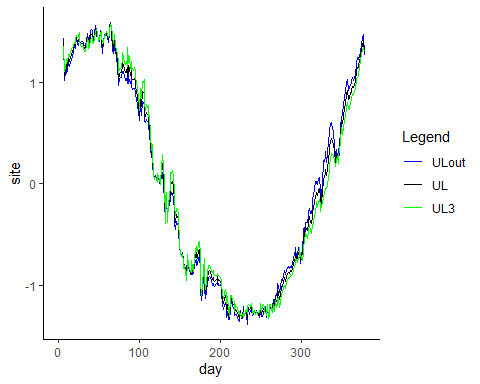
#UL  
color <- c("ULout" = "blue", "UL1" = "red", "UL3" = "green")  
transformed\_dat\_klamath\_df %>% rowid\_to\_column(var = "day") %>% ggplot()+  
 geom\_line(aes(x = day, y = ULout, color = "ULout")) +  
 geom\_line(aes(x = day, y = UL1, color = "UL1"))+  
 geom\_line(aes(x = day, y = UL3, color = "UL3"))+  
 labs(x = "day", y = "site", color = "Legend")+  
 theme\_classic()+  
 scale\_color\_manual(values = color, labels = c("ULout","UL1","UL3"))

## Warning: Removed 7 row(s) containing missing values (geom\_path).  
  
## Warning: Removed 7 row(s) containing missing values (geom\_path).  
  
## Warning: Removed 7 row(s) containing missing values (geom\_path).



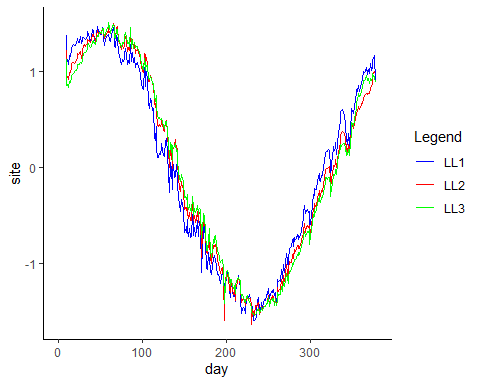
#need to remove UL1  
transformed\_dat\_klamath\_df$UL <- rowMeans(transformed\_dat\_klamath\_df[,c('ULout', 'UL3')], na.rm=TRUE)  
  
color <- c("ULout" = "blue", "UL" = "black", "UL3" = "green")  
transformed\_dat\_klamath\_df %>% rowid\_to\_column(var = "day") %>% ggplot()+  
 geom\_line(aes(x = day, y = ULout, color = "ULout")) +  
 geom\_line(aes(x = day, y = UL, color = "UL"))+  
 geom\_line(aes(x = day, y = UL3, color = "UL3"))+  
 labs(x = "day", y = "site", color = "Legend")+  
 theme\_classic()+  
 scale\_color\_manual(values = color, labels = c("ULout","UL","UL3"))

## Warning: Removed 7 row(s) containing missing values (geom\_path).  
  
## Warning: Removed 7 row(s) containing missing values (geom\_path).  
  
## Warning: Removed 7 row(s) containing missing values (geom\_path).



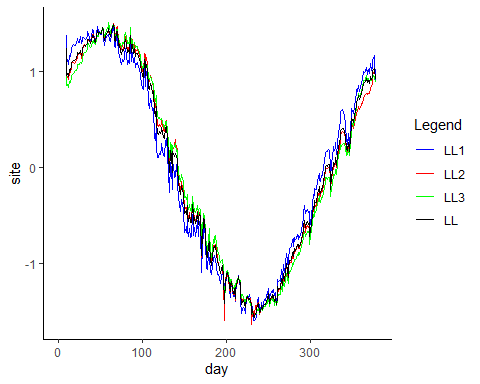
#LL  
color <- c("LL1" = "blue", "LL2" = "red", "LL3" = "green")  
transformed\_dat\_klamath\_df %>% rowid\_to\_column(var = "day") %>% ggplot()+  
 geom\_line(aes(x = day, y = LL1, color = "LL1")) +  
 geom\_line(aes(x = day, y = LL2, color = "LL2"))+  
 geom\_line(aes(x = day, y = LL3, color = "LL3"))+  
 labs(x = "day", y = "site", color = "Legend")+  
 theme\_classic()+  
 scale\_color\_manual(values = color, labels = c("LL1","LL2","LL3"))

## Warning: Removed 9 row(s) containing missing values (geom\_path).  
  
## Warning: Removed 9 row(s) containing missing values (geom\_path).  
  
## Warning: Removed 9 row(s) containing missing values (geom\_path).



#all ok  
transformed\_dat\_klamath\_df$LL <- rowMeans(transformed\_dat\_klamath\_df[,c('LL1','LL2', 'LL3')], na.rm=TRUE)  
  
color <- c("LL1" = "blue", "LL2" = "red", "LL3" = "green", 'LL' = "black")  
transformed\_dat\_klamath\_df %>% rowid\_to\_column(var = "day") %>% ggplot()+  
 geom\_line(aes(x = day, y = LL1, color = "LL1")) +  
 geom\_line(aes(x = day, y = LL2, color = "LL2"))+  
 geom\_line(aes(x = day, y = LL3, color = "LL3"))+  
 geom\_line(aes(x = day, y = LL, color = "LL"))+  
 labs(x = "day", y = "site", color = "Legend")+  
 theme\_classic()+  
 scale\_color\_manual(values = color, labels = c("LL1","LL2","LL3", "LL"))

## Warning: Removed 9 row(s) containing missing values (geom\_path).  
  
## Warning: Removed 9 row(s) containing missing values (geom\_path).  
  
## Warning: Removed 9 row(s) containing missing values (geom\_path).  
  
## Warning: Removed 9 row(s) containing missing values (geom\_path).

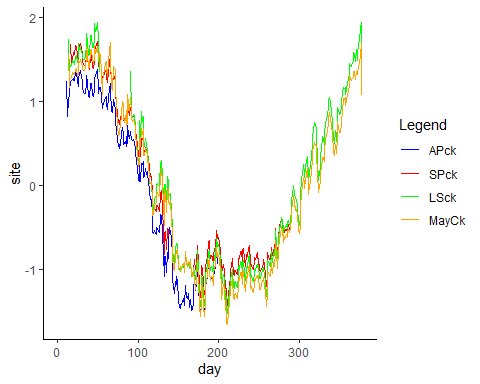


#SC  
color <- c("APck" = "blue", "SPck" = "red", "LSck" = "green", "MayCk"="orange")  
transformed\_dat\_klamath\_df %>% rowid\_to\_column(var = "day") %>% ggplot()+  
 geom\_line(aes(x = day, y = APck, color = "APck")) +  
 geom\_line(aes(x = day, y = SPck, color = "SPck"))+  
 geom\_line(aes(x = day, y = LSck, color = "LSck"))+  
 geom\_line(aes(x = day, y = MayCk, color = "MayCk"))+  
 labs(x = "day", y = "site", color = "Legend")+  
 theme\_classic()+  
 scale\_color\_manual(values = color, labels = c("APck","SPck","LSck","MayCk"))

## Warning: Removed 189 row(s) containing missing values (geom\_path).

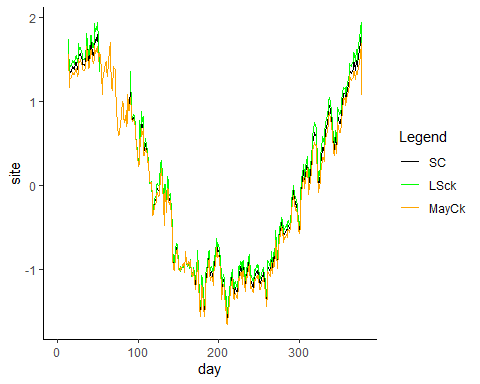
## Warning: Removed 104 row(s) containing missing values (geom\_path).

## Warning: Removed 12 row(s) containing missing values (geom\_path).  
  
## Warning: Removed 12 row(s) containing missing values (geom\_path).



#need to remove APck and SPck  
transformed\_dat\_klamath\_df$SC <- rowMeans(transformed\_dat\_klamath\_df[,c('LSck','MayCk')], na.rm=TRUE)  
  
color <- c("SC" = "black","LSck" = "green", "MayCk"="orange")  
transformed\_dat\_klamath\_df %>% rowid\_to\_column(var = "day") %>% ggplot()+  
 geom\_line(aes(x = day, y = SC, color = "SC"))+  
 geom\_line(aes(x = day, y = LSck, color = "LSck"))+  
 geom\_line(aes(x = day, y = MayCk, color = "MayCk"))+  
 labs(x = "day", y = "site", color = "Legend")+  
 theme\_classic()+  
 scale\_color\_manual(values = color, labels = c("SC","LSck","MayCk"))

## Warning: Removed 12 row(s) containing missing values (geom\_path).  
  
## Warning: Removed 12 row(s) containing missing values (geom\_path).  
  
## Warning: Removed 12 row(s) containing missing values (geom\_path).

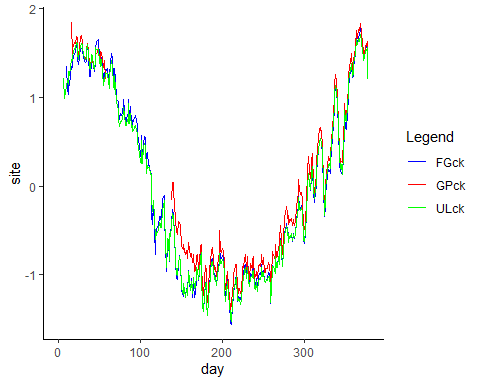


#HC  
color <- c("FGck" = "blue", "GPck" = "red", "ULck" = "green")  
transformed\_dat\_klamath\_df %>% rowid\_to\_column(var = "day") %>% ggplot()+  
 geom\_line(aes(x = day, y = FGck, color = "FGck")) +  
 geom\_line(aes(x = day, y = GPck, color = "GPck"))+  
 geom\_line(aes(x = day, y = ULck, color = "ULck"))+  
 labs(x = "day", y = "site", color = "Legend")+  
 theme\_classic()+  
 scale\_color\_manual(values = color, labels = c("FGck","GPck","ULck"))

## Warning: Removed 10 row(s) containing missing values (geom\_path).

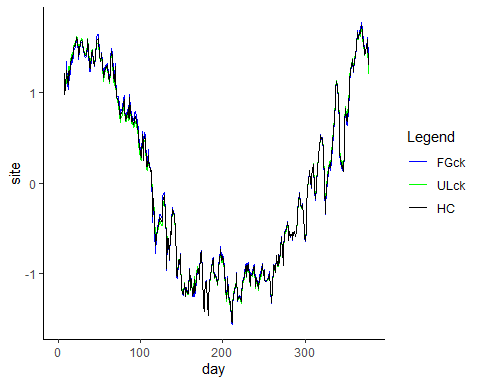
## Warning: Removed 16 row(s) containing missing values (geom\_path).

## Warning: Removed 7 row(s) containing missing values (geom\_path).



#need to remove GPck  
transformed\_dat\_klamath\_df$HC <- rowMeans(transformed\_dat\_klamath\_df[,c('ULck','FGck')], na.rm=TRUE)  
  
color <- c("FGck" = "blue", "ULck" = "green", "HC" = "black")  
transformed\_dat\_klamath\_df %>% rowid\_to\_column(var = "day") %>% ggplot()+  
 geom\_line(aes(x = day, y = FGck, color = "FGck")) +  
 geom\_line(aes(x = day, y = ULck, color = "ULck"))+  
 geom\_line(aes(x = day, y = HC, color = "HC"))+  
 labs(x = "day", y = "site", color = "Legend")+  
 theme\_classic()+  
 scale\_color\_manual(values = color, labels = c("FGck","ULck", "HC"))

## Warning: Removed 10 row(s) containing missing values (geom\_path).  
  
## Warning: Removed 7 row(s) containing missing values (geom\_path).  
  
## Warning: Removed 7 row(s) containing missing values (geom\_path).

 #Matrix of transformed data with 12 sites (condensed\_transdat)

condensed\_transdat\_df <- cbind(AP = transformed\_dat\_klamath\_df$AP3, SP = transformed\_dat\_klamath\_df$SP, Durazo = transformed\_dat\_klamath\_df$Durazo, LS = transformed\_dat\_klamath\_df$LS, May = transformed\_dat\_klamath\_df$May, FG = transformed\_dat\_klamath\_df$FG3, GP = transformed\_dat\_klamath\_df$GP, UL = transformed\_dat\_klamath\_df$UL, LL = transformed\_dat\_klamath\_df$LL, SC = transformed\_dat\_klamath\_df$SC, HC = transformed\_dat\_klamath\_df$HC, KSV = transformed\_dat\_klamath\_df$KSV)  
condensed\_transdat <- as.matrix(t(condensed\_transdat\_df))  
str(condensed\_transdat)

## num [1:12, 1:378] NA NaN NA NA NA NA NaN NaN NaN NaN ...  
## - attr(\*, "dimnames")=List of 2  
## ..$ : chr [1:12] "AP" "SP" "Durazo" "LS" ...  
## ..$ : NULL

saveRDS(condensed\_transdat, "condensed\_transdat.rds")

# Hypothesis 1: all states have different levels of stochastic (Q) and deterministic (C) variability

## AICc -9599.906

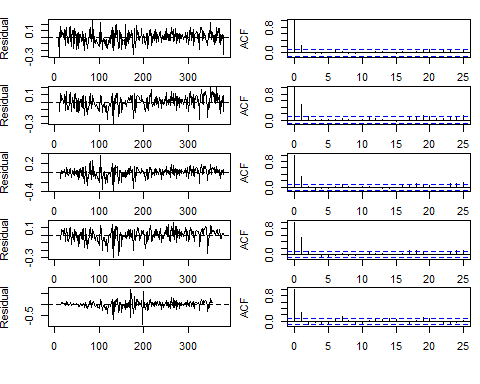
mod1\_condense = list()  
mod1\_condense$A = "zero"   
mod1\_condense$Z = "identity"  
mod1\_condense$R = "zero" #all the sensors are same, so observation error should be same  
mod1\_condense$Q = "diagonal and unequal"   
mod1\_condense$B = "diagonal and unequal" #assuming no species interactions  
mod1\_condense$U = "zero"   
mod1\_condense$C = "unequal"   
mod1\_condense$c = transformed\_covariate\_klamath  
mod1\_condense.fit = MARSS(condensed\_transdat, model=mod1\_condense)

## MARSS: NaNs in data are being replaced with NAs. There might be a problem if NaNs shouldn't be in the data.  
## NA is the normal missing value designation.  
## Success! abstol and log-log tests passed at 153 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 153 iterations.   
## Log-likelihood: 4857.285   
## AIC: -9618.57 AICc: -9617.484   
##   
## Estimate  
## B.(X.AP,X.AP) 0.942858  
## B.(X.SP,X.SP) 0.968046  
## B.(X.Durazo,X.Durazo) 0.967180  
## B.(X.LS,X.LS) 0.987704  
## B.(X.May,X.May) 0.966777  
## B.(X.FG,X.FG) 0.972812  
## B.(X.GP,X.GP) 0.987907  
## B.(X.UL,X.UL) 0.965388  
## B.(X.LL,X.LL) 0.965301  
## B.(X.SC,X.SC) 0.911952  
## B.(X.HC,X.HC) 0.890677  
## B.(X.KSV,X.KSV) 0.945554  
## Q.(X.AP,X.AP) 0.008070  
## Q.(X.SP,X.SP) 0.006837  
## Q.(X.Durazo,X.Durazo) 0.008850  
## Q.(X.LS,X.LS) 0.005552  
## Q.(X.May,X.May) 0.024014  
## Q.(X.FG,X.FG) 0.003015  
## Q.(X.GP,X.GP) 0.000809  
## Q.(X.UL,X.UL) 0.004623  
## Q.(X.LL,X.LL) 0.005431  
## Q.(X.SC,X.SC) 0.015851  
## Q.(X.HC,X.HC) 0.013369  
## Q.(X.KSV,X.KSV) 0.005327  
## x0.X.AP 2.471342  
## x0.X.SP 1.288391  
## x0.X.Durazo 1.695049  
## x0.X.LS 1.695107  
## x0.X.May 1.933986  
## x0.X.FG 1.296982  
## x0.X.GP 0.632370  
## x0.X.UL 1.631588  
## x0.X.LL 1.487285  
## x0.X.SC 3.612499  
## x0.X.HC 2.116309  
## x0.X.KSV 1.106298  
## C.X.AP 0.057603  
## C.X.SP 0.035406  
## C.X.Durazo 0.033553  
## C.X.LS 0.010417  
## C.X.May 0.028579  
## C.X.FG 0.030953  
## C.X.GP 0.018791  
## C.X.UL 0.037363  
## C.X.LL 0.037844  
## C.X.SC 0.088024  
## C.X.HC 0.113147  
## C.X.KSV 0.057921  
## Initial states (x0) defined at t=0  
##   
## Standard errors have not been calculated.   
## Use MARSSparamCIs to compute CIs and bias estimates.

mod1\_condense.params =MARSSparamCIs(mod1\_condense.fit)  
MARSSparamCIs(mod1\_condense.fit)

##   
## MARSS fit is  
## Estimation method: kem   
## Convergence test: conv.test.slope.tol = 0.5, abstol = 0.001  
## Estimation converged in 153 iterations.   
## Log-likelihood: 4857.285   
## AIC: -9618.57 AICc: -9617.484   
##   
## ML.Est Std.Err low.CI up.CI  
## B.(X.AP,X.AP) 0.942858 9.98e-03 0.923290 0.962425  
## B.(X.SP,X.SP) 0.968046 8.89e-03 0.950627 0.985464  
## B.(X.Durazo,X.Durazo) 0.967180 1.01e-02 0.947471 0.986889  
## B.(X.LS,X.LS) 0.987704 7.41e-03 0.973183 1.002224  
## B.(X.May,X.May) 0.966777 1.29e-02 0.941399 0.992155  
## B.(X.FG,X.FG) 0.972812 5.22e-03 0.962583 0.983041  
## B.(X.GP,X.GP) 0.987907 1.86e-03 0.984254 0.991560  
## B.(X.UL,X.UL) 0.965388 6.54e-03 0.952573 0.978204  
## B.(X.LL,X.LL) 0.965301 6.46e-03 0.952634 0.977969  
## B.(X.SC,X.SC) 0.911952 1.37e-02 0.885028 0.938876  
## B.(X.HC,X.HC) 0.890677 1.33e-02 0.864686 0.916667  
## B.(X.KSV,X.KSV) 0.945554 8.96e-03 0.927994 0.963113  
## Q.(X.AP,X.AP) 0.008070 5.94e-04 0.006906 0.009235  
## Q.(X.SP,X.SP) 0.006837 5.04e-04 0.005849 0.007825  
## Q.(X.Durazo,X.Durazo) 0.008850 6.54e-04 0.007567 0.010132  
## Q.(X.LS,X.LS) 0.005552 4.10e-04 0.004748 0.006357  
## Q.(X.May,X.May) 0.024014 1.84e-03 0.020409 0.027618  
## Q.(X.FG,X.FG) 0.003015 2.22e-04 0.002579 0.003450  
## Q.(X.GP,X.GP) 0.000809 5.96e-05 0.000692 0.000926  
## Q.(X.UL,X.UL) 0.004623 3.39e-04 0.003958 0.005289  
## Q.(X.LL,X.LL) 0.005431 4.00e-04 0.004648 0.006215  
## Q.(X.SC,X.SC) 0.015851 1.17e-03 0.013554 0.018148  
## Q.(X.HC,X.HC) 0.013369 9.82e-04 0.011445 0.015293  
## Q.(X.KSV,X.KSV) 0.005327 4.00e-04 0.004544 0.006111  
## x0.X.AP 2.471342 4.56e-01 1.577050 3.365633  
## x0.X.SP 1.288391 3.47e-01 0.607769 1.969013  
## x0.X.Durazo 1.695049 4.64e-01 0.785959 2.604138  
## x0.X.LS 1.695107 3.16e-01 1.075835 2.314380  
## x0.X.May 1.933986 7.65e-01 0.434243 3.433728  
## x0.X.FG 1.296982 2.09e-01 0.887063 1.706901  
## x0.X.GP 0.632370 9.13e-02 0.453480 0.811259  
## x0.X.UL 1.631588 2.17e-01 1.206646 2.056529  
## x0.X.LL 1.487285 2.75e-01 0.948641 2.025929  
## x0.X.SC 3.612499 1.14e+00 1.385920 5.839078  
## x0.X.HC 2.116309 5.45e-01 1.048644 3.183974  
## x0.X.KSV 1.106298 7.76e-02 0.954217 1.258379  
## C.X.AP 0.057603 9.89e-03 0.038220 0.076985  
## C.X.SP 0.035406 8.78e-03 0.018195 0.052618  
## C.X.Durazo 0.033553 9.94e-03 0.014065 0.053040  
## C.X.LS 0.010417 7.33e-03 -0.003956 0.024789  
## C.X.May 0.028579 1.35e-02 0.002126 0.055033  
## C.X.FG 0.030953 5.18e-03 0.020806 0.041101  
## C.X.GP 0.018791 1.84e-03 0.015179 0.022403  
## C.X.UL 0.037363 6.48e-03 0.024652 0.050073  
## C.X.LL 0.037844 6.40e-03 0.025305 0.050383  
## C.X.SC 0.088024 1.37e-02 0.061163 0.114884  
## C.X.HC 0.113147 1.32e-02 0.087314 0.138981  
## C.X.KSV 0.057921 8.88e-03 0.040523 0.075318  
## Initial states (x0) defined at t=0  
##   
## CIs calculated at alpha = 0.05 via method=hessian

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(mod1\_condense.fit, type = "tt1")$model.residuals[j, ],  
 ylab = "Residual")  
 abline(h = 0, lty = "dashed")  
 acf(residuals,na.action = na.pass)  
 }



# Hypothesis 2: all states have save levels of stochastic (Q) and deterministic (C) variability

## AICc -8356.184

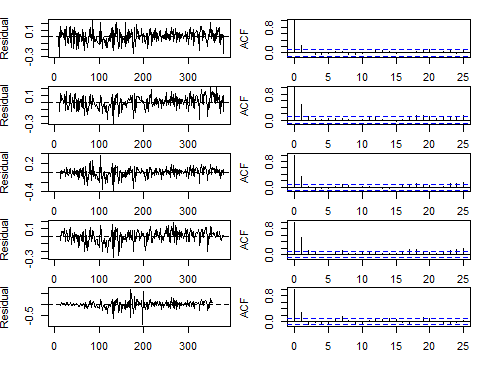
mod2\_condense = list()  
mod2\_condense$A = "zero"   
mod2\_condense$Z = "identity"  
mod2\_condense$R = "zero" #all the sensors are same, so observation error should be same  
mod2\_condense$Q = "diagonal and equal"   
mod2\_condense$B = "diagonal and equal" #assuming no species interactions  
mod2\_condense$U = "zero"   
mod2\_condense$C = "equal"   
mod2\_condense$c = transformed\_covariate\_klamath  
mod2\_condense.fit = MARSS(condensed\_transdat, model=mod2\_condense)

## MARSS: NaNs in data are being replaced with NAs. There might be a problem if NaNs shouldn't be in the data.  
## NA is the normal missing value designation.  
## Success! abstol and log-log tests passed at 91 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 91 iterations.   
## Log-likelihood: 4179.008   
## AIC: -8328.017 AICc: -8327.907   
##   
## Estimate  
## B.diag 0.96268  
## Q.diag 0.00862  
## x0.X.AP 2.10294  
## x0.X.SP 1.33729  
## x0.X.Durazo 1.72025  
## x0.X.LS 2.02143  
## x0.X.May 1.90140  
## x0.X.FG 1.38748  
## x0.X.GP 0.69061  
## x0.X.UL 1.65739  
## x0.X.LL 1.51656  
## x0.X.SC 2.21232  
## x0.X.HC 1.42914  
## x0.X.KSV 1.09279  
## C.1 0.03920  
## Initial states (x0) defined at t=0  
##   
## Standard errors have not been calculated.   
## Use MARSSparamCIs to compute CIs and bias estimates.

mod2\_condense.params =MARSSparamCIs(mod2\_condense.fit)  
MARSSparamCIs(mod2\_condense.fit)

##   
## MARSS fit is  
## Estimation method: kem   
## Convergence test: conv.test.slope.tol = 0.5, abstol = 0.001  
## Estimation converged in 91 iterations.   
## Log-likelihood: 4179.008   
## AIC: -8328.017 AICc: -8327.907   
##   
## ML.Est Std.Err low.CI up.CI  
## B.diag 0.96268 0.002541 0.95770 0.96766  
## Q.diag 0.00862 0.000184 0.00826 0.00898  
## x0.X.AP 2.10294 0.368852 1.38001 2.82588  
## x0.X.SP 1.33729 0.393307 0.56642 2.10815  
## x0.X.Durazo 1.72025 0.447742 0.84269 2.59781  
## x0.X.LS 2.02143 0.448790 1.14182 2.90104  
## x0.X.May 1.90140 0.448344 1.02266 2.78014  
## x0.X.FG 1.38748 0.367109 0.66796 2.10700  
## x0.X.GP 0.69061 0.340206 0.02382 1.35740  
## x0.X.UL 1.65739 0.288615 1.09172 2.22307  
## x0.X.LL 1.51656 0.341139 0.84794 2.18518  
## x0.X.SC 2.21232 0.449576 1.33117 3.09347  
## x0.X.HC 1.42914 0.288301 0.86408 1.99420  
## x0.X.KSV 1.09279 0.096448 0.90375 1.28182  
## C.1 0.03920 0.002528 0.03424 0.04415  
## Initial states (x0) defined at t=0  
##   
## CIs calculated at alpha = 0.05 via method=hessian

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(mod2\_condense.fit, type = "tt1")$model.residuals[j, ],  
 ylab = "Residual")  
 abline(h = 0, lty = "dashed")  
 acf(residuals,na.action = na.pass)  
 }



# Hypothesis 3: Habitat type: Creeks vs ponds vs Klamath

## AICc: -8586.263

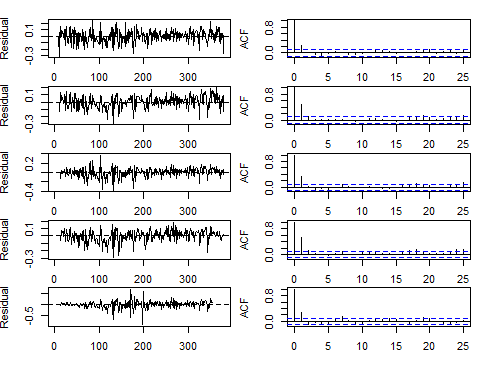
mod3\_condense = list()  
## Modify matrices  
# 1st: group time series into categories  
hypothesis = c("pond","pond","pond","pond",  
 "pond","pond","pond","pond",  
 "pond","creek","creek","Klamath")  
  
# 2nd: build C matrix (12 x 1)  
mod3\_condense$C = matrix(hypothesis)  
mod3\_condense$c = transformed\_covariate\_klamath  
  
# 3rd: build Q matrix (12 x 12, with "C vector" in its diagonal)  
Q <- matrix(list("pond",0,0,0,0,0,0,0,0,0,0,0,  
 0,"pond",0,0,0,0,0,0,0,0,0,0,  
 0,0,"pond",0,0,0,0,0,0,0,0,0,  
 0,0,0,"pond",0,0,0,0,0,0,0,0,  
 0,0,0,0,"pond",0,0,0,0,0,0,0,  
 0,0,0,0,0,"pond",0,0,0,0,0,0,  
 0,0,0,0,0,0,"pond",0,0,0,0,0,  
 0,0,0,0,0,0,0,"pond",0,0,0,0,  
 0,0,0,0,0,0,0,0,"pond",0,0,0,  
 0,0,0,0,0,0,0,0,0,"creek",0,0,  
 0,0,0,0,0,0,0,0,0,0,"creek",0,  
 0,0,0,0,0,0,0,0,0,0,0,"Klamath"),12,12)  
  
# 4th: B, identical as Q  
B <- Q  
  
mod3\_condense$A = "zero"   
mod3\_condense$Z = "identity"  
mod3\_condense$R = "zero" #all the sensors are same, so observation error should be same  
mod3\_condense$Q = Q   
mod3\_condense$B = Q  
mod3\_condense$U = "zero"   
mod3\_condense$C = matrix(hypothesis)   
mod3\_condense$c = transformed\_covariate\_klamath  
mod3\_condense.fit = MARSS(condensed\_transdat, model=mod3\_condense)

## MARSS: NaNs in data are being replaced with NAs. There might be a problem if NaNs shouldn't be in the data.  
## NA is the normal missing value designation.  
## Success! abstol and log-log tests passed at 174 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 174 iterations.   
## Log-likelihood: 4322.357   
## AIC: -8602.714 AICc: -8602.502   
##   
## Estimate  
## B.pond 0.97214  
## B.creek 0.90200  
## B.Klamath 0.94555  
## Q.pond 0.00739  
## Q.creek 0.01463  
## Q.Klamath 0.00533  
## x0.X.AP 1.96679  
## x0.X.SP 1.27285  
## x0.X.Durazo 1.62143  
## x0.X.LS 1.89098  
## x0.X.May 1.78356  
## x0.X.FG 1.31606  
## x0.X.GP 0.67854  
## x0.X.UL 1.58070  
## x0.X.LL 1.43608  
## x0.X.SC 3.99633  
## x0.X.HC 1.98203  
## x0.X.KSV 1.10630  
## C.pond 0.02952  
## C.creek 0.09995  
## C.Klamath 0.05792  
## Initial states (x0) defined at t=0  
##   
## Standard errors have not been calculated.   
## Use MARSSparamCIs to compute CIs and bias estimates.

mod3\_condense.params =MARSSparamCIs(mod3\_condense.fit)  
MARSSparamCIs(mod3\_condense.fit)

##   
## MARSS fit is  
## Estimation method: kem   
## Convergence test: conv.test.slope.tol = 0.5, abstol = 0.001  
## Estimation converged in 174 iterations.   
## Log-likelihood: 4322.357   
## AIC: -8602.714 AICc: -8602.502   
##   
## ML.Est Std.Err low.CI up.CI  
## B.pond 0.97214 0.002588 0.96706 0.97721  
## B.creek 0.90200 0.009564 0.88325 0.92074  
## B.Klamath 0.94555 0.008959 0.92799 0.96311  
## Q.pond 0.00739 0.000182 0.00703 0.00774  
## Q.creek 0.01463 0.000762 0.01314 0.01613  
## Q.Klamath 0.00533 0.000400 0.00454 0.00611  
## x0.X.AP 1.96679 0.322298 1.33510 2.59848  
## x0.X.SP 1.27285 0.341415 0.60369 1.94201  
## x0.X.Durazo 1.62143 0.384107 0.86859 2.37426  
## x0.X.LS 1.89098 0.385139 1.13612 2.64583  
## x0.X.May 1.78356 0.384701 1.02956 2.53756  
## x0.X.FG 1.31606 0.320583 0.68773 1.94439  
## x0.X.GP 0.67854 0.298741 0.09302 1.26407  
## x0.X.UL 1.58070 0.256472 1.07803 2.08338  
## x0.X.LL 1.43608 0.299678 0.84872 2.02343  
## x0.X.SC 3.99633 1.134910 1.77195 6.22072  
## x0.X.HC 1.98203 0.518557 0.96568 2.99838  
## x0.X.KSV 1.10630 0.077594 0.95422 1.25838  
## C.pond 0.02952 0.002575 0.02447 0.03457  
## C.creek 0.09995 0.009524 0.08129 0.11862  
## C.Klamath 0.05792 0.008876 0.04052 0.07532  
## Initial states (x0) defined at t=0  
##   
## CIs calculated at alpha = 0.05 via method=hessian

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(mod3\_condense.fit, type = "tt1")$model.residuals[j, ],  
 ylab = "Residual")  
 abline(h = 0, lty = "dashed")  
 acf(residuals,na.action = na.pass)  
 }

 # Hypothesis 4: By watershed ## AICc: -8597.364

mod4\_condense = list()  
## Modify matrices  
# 1st: group time series into categories  
rownames(condensed\_transdat)

## [1] "AP" "SP" "Durazo" "LS" "May" "FG" "GP" "UL"   
## [9] "LL" "SC" "HC" "KSV"

hypothesis2 = c("SC","SC","SC","SC",  
 "SC","HC","HC","HC",  
 "HC","SC","HC","Klamath")  
  
# 2nd: build C matrix (12 x 1)  
mod4\_condense$C = matrix(hypothesis2)  
mod4\_condense$c = transformed\_covariate\_klamath  
  
# 3rd: build Q matrix (12 x 12, with "C vector" in its diagonal)  
Q <- matrix(list("SC",0,0,0,0,0,0,0,0,0,0,0,  
 0,"SC",0,0,0,0,0,0,0,0,0,0,  
 0,0,"SC",0,0,0,0,0,0,0,0,0,  
 0,0,0,"SC",0,0,0,0,0,0,0,0,  
 0,0,0,0,"SC",0,0,0,0,0,0,0,  
 0,0,0,0,0,"HC",0,0,0,0,0,0,  
 0,0,0,0,0,0,"HC",0,0,0,0,0,  
 0,0,0,0,0,0,0,"HC",0,0,0,0,  
 0,0,0,0,0,0,0,0,"HC",0,0,0,  
 0,0,0,0,0,0,0,0,0,"SC",0,0,  
 0,0,0,0,0,0,0,0,0,0,"HC",0,  
 0,0,0,0,0,0,0,0,0,0,0,"Klamath"),12,12)  
  
# 4th: B, identical as Q  
B <- Q  
  
mod4\_condense$A = "zero"   
mod4\_condense$Z = "identity"  
mod4\_condense$R = "zero" #all the sensors are same, so observation error should be same  
mod4\_condense$Q = Q   
mod4\_condense$B = Q  
mod4\_condense$U = "zero"   
mod4\_condense$C = matrix(hypothesis2)   
mod4\_condense$c = transformed\_covariate\_klamath  
mod4\_condense.fit = MARSS(condensed\_transdat, model=mod4\_condense)

## MARSS: NaNs in data are being replaced with NAs. There might be a problem if NaNs shouldn't be in the data.  
## NA is the normal missing value designation.  
## Success! abstol and log-log tests passed at 94 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 94 iterations.   
## Log-likelihood: 4316.005   
## AIC: -8590.01 AICc: -8589.798   
##   
## Estimate  
## B.SC 0.95981  
## B.HC 0.96701  
## B.Klamath 0.94555  
## Q.SC 0.01154  
## Q.HC 0.00577  
## Q.Klamath 0.00533  
## x0.X.AP 2.16128  
## x0.X.SP 1.37678  
## x0.X.Durazo 1.78037  
## x0.X.LS 2.09239  
## x0.X.May 1.96804  
## x0.X.FG 1.33906  
## x0.X.GP 0.67251  
## x0.X.UL 1.61312  
## x0.X.LL 1.46665  
## x0.X.SC 2.29015  
## x0.X.HC 1.39184  
## x0.X.KSV 1.10630  
## C.SC 0.04004  
## C.HC 0.03700  
## C.Klamath 0.05792  
## Initial states (x0) defined at t=0  
##   
## Standard errors have not been calculated.   
## Use MARSSparamCIs to compute CIs and bias estimates.

mod4\_condense.params =MARSSparamCIs(mod4\_condense.fit)  
MARSSparamCIs(mod4\_condense.fit)

##   
## MARSS fit is  
## Estimation method: kem   
## Convergence test: conv.test.slope.tol = 0.5, abstol = 0.001  
## Estimation converged in 94 iterations.   
## Log-likelihood: 4316.005   
## AIC: -8590.01 AICc: -8589.798   
##   
## ML.Est Std.Err low.CI up.CI  
## B.SC 0.95981 0.00444 0.95110 0.96851  
## B.HC 0.96701 0.00292 0.96129 0.97273  
## B.Klamath 0.94555 0.00896 0.92799 0.96311  
## Q.SC 0.01154 0.00035 0.01086 0.01223  
## Q.HC 0.00577 0.00019 0.00540 0.00615  
## Q.Klamath 0.00533 0.00040 0.00454 0.00611  
## x0.X.AP 2.16128 0.43895 1.30095 3.02160  
## x0.X.SP 1.37678 0.46591 0.46361 2.28995  
## x0.X.Durazo 1.78037 0.53408 0.73360 2.82713  
## x0.X.LS 2.09239 0.53694 1.04000 3.14477  
## x0.X.May 1.96804 0.53572 0.91804 3.01804  
## x0.X.FG 1.33906 0.29324 0.76433 1.91380  
## x0.X.GP 0.67251 0.27192 0.13956 1.20547  
## x0.X.UL 1.61312 0.23258 1.15727 2.06897  
## x0.X.LL 1.46665 0.27338 0.93083 2.00246  
## x0.X.SC 2.29015 0.53909 1.23355 3.34674  
## x0.X.HC 1.39184 0.23209 0.93694 1.84674  
## x0.X.KSV 1.10630 0.07759 0.95422 1.25838  
## C.SC 0.04004 0.00444 0.03134 0.04873  
## C.HC 0.03700 0.00290 0.03133 0.04268  
## C.Klamath 0.05792 0.00888 0.04052 0.07532  
## Initial states (x0) defined at t=0  
##   
## CIs calculated at alpha = 0.05 via method=hessian

#Plot Covariates

mod1\_condense.fit = MARSS(condensed\_transdat, model=mod1\_condense)

## MARSS: NaNs in data are being replaced with NAs. There might be a problem if NaNs shouldn't be in the data.  
## NA is the normal missing value designation.  
## Success! abstol and log-log tests passed at 153 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 153 iterations.   
## Log-likelihood: 4857.285   
## AIC: -9618.57 AICc: -9617.484   
##   
## Estimate  
## B.(X.AP,X.AP) 0.942858  
## B.(X.SP,X.SP) 0.968046  
## B.(X.Durazo,X.Durazo) 0.967180  
## B.(X.LS,X.LS) 0.987704  
## B.(X.May,X.May) 0.966777  
## B.(X.FG,X.FG) 0.972812  
## B.(X.GP,X.GP) 0.987907  
## B.(X.UL,X.UL) 0.965388  
## B.(X.LL,X.LL) 0.965301  
## B.(X.SC,X.SC) 0.911952  
## B.(X.HC,X.HC) 0.890677  
## B.(X.KSV,X.KSV) 0.945554  
## Q.(X.AP,X.AP) 0.008070  
## Q.(X.SP,X.SP) 0.006837  
## Q.(X.Durazo,X.Durazo) 0.008850  
## Q.(X.LS,X.LS) 0.005552  
## Q.(X.May,X.May) 0.024014  
## Q.(X.FG,X.FG) 0.003015  
## Q.(X.GP,X.GP) 0.000809  
## Q.(X.UL,X.UL) 0.004623  
## Q.(X.LL,X.LL) 0.005431  
## Q.(X.SC,X.SC) 0.015851  
## Q.(X.HC,X.HC) 0.013369  
## Q.(X.KSV,X.KSV) 0.005327  
## x0.X.AP 2.471342  
## x0.X.SP 1.288391  
## x0.X.Durazo 1.695049  
## x0.X.LS 1.695107  
## x0.X.May 1.933986  
## x0.X.FG 1.296982  
## x0.X.GP 0.632370  
## x0.X.UL 1.631588  
## x0.X.LL 1.487285  
## x0.X.SC 3.612499  
## x0.X.HC 2.116309  
## x0.X.KSV 1.106298  
## C.X.AP 0.057603  
## C.X.SP 0.035406  
## C.X.Durazo 0.033553  
## C.X.LS 0.010417  
## C.X.May 0.028579  
## C.X.FG 0.030953  
## C.X.GP 0.018791  
## C.X.UL 0.037363  
## C.X.LL 0.037844  
## C.X.SC 0.088024  
## C.X.HC 0.113147  
## C.X.KSV 0.057921  
## Initial states (x0) defined at t=0  
##   
## Standard errors have not been calculated.   
## Use MARSSparamCIs to compute CIs and bias estimates.

mod1\_condense.params =MARSSparamCIs(mod1\_condense.fit)  
MARSSparamCIs(mod1\_condense.fit)

##   
## MARSS fit is  
## Estimation method: kem   
## Convergence test: conv.test.slope.tol = 0.5, abstol = 0.001  
## Estimation converged in 153 iterations.   
## Log-likelihood: 4857.285   
## AIC: -9618.57 AICc: -9617.484   
##   
## ML.Est Std.Err low.CI up.CI  
## B.(X.AP,X.AP) 0.942858 9.98e-03 0.923290 0.962425  
## B.(X.SP,X.SP) 0.968046 8.89e-03 0.950627 0.985464  
## B.(X.Durazo,X.Durazo) 0.967180 1.01e-02 0.947471 0.986889  
## B.(X.LS,X.LS) 0.987704 7.41e-03 0.973183 1.002224  
## B.(X.May,X.May) 0.966777 1.29e-02 0.941399 0.992155  
## B.(X.FG,X.FG) 0.972812 5.22e-03 0.962583 0.983041  
## B.(X.GP,X.GP) 0.987907 1.86e-03 0.984254 0.991560  
## B.(X.UL,X.UL) 0.965388 6.54e-03 0.952573 0.978204  
## B.(X.LL,X.LL) 0.965301 6.46e-03 0.952634 0.977969  
## B.(X.SC,X.SC) 0.911952 1.37e-02 0.885028 0.938876  
## B.(X.HC,X.HC) 0.890677 1.33e-02 0.864686 0.916667  
## B.(X.KSV,X.KSV) 0.945554 8.96e-03 0.927994 0.963113  
## Q.(X.AP,X.AP) 0.008070 5.94e-04 0.006906 0.009235  
## Q.(X.SP,X.SP) 0.006837 5.04e-04 0.005849 0.007825  
## Q.(X.Durazo,X.Durazo) 0.008850 6.54e-04 0.007567 0.010132  
## Q.(X.LS,X.LS) 0.005552 4.10e-04 0.004748 0.006357  
## Q.(X.May,X.May) 0.024014 1.84e-03 0.020409 0.027618  
## Q.(X.FG,X.FG) 0.003015 2.22e-04 0.002579 0.003450  
## Q.(X.GP,X.GP) 0.000809 5.96e-05 0.000692 0.000926  
## Q.(X.UL,X.UL) 0.004623 3.39e-04 0.003958 0.005289  
## Q.(X.LL,X.LL) 0.005431 4.00e-04 0.004648 0.006215  
## Q.(X.SC,X.SC) 0.015851 1.17e-03 0.013554 0.018148  
## Q.(X.HC,X.HC) 0.013369 9.82e-04 0.011445 0.015293  
## Q.(X.KSV,X.KSV) 0.005327 4.00e-04 0.004544 0.006111  
## x0.X.AP 2.471342 4.56e-01 1.577050 3.365633  
## x0.X.SP 1.288391 3.47e-01 0.607769 1.969013  
## x0.X.Durazo 1.695049 4.64e-01 0.785959 2.604138  
## x0.X.LS 1.695107 3.16e-01 1.075835 2.314380  
## x0.X.May 1.933986 7.65e-01 0.434243 3.433728  
## x0.X.FG 1.296982 2.09e-01 0.887063 1.706901  
## x0.X.GP 0.632370 9.13e-02 0.453480 0.811259  
## x0.X.UL 1.631588 2.17e-01 1.206646 2.056529  
## x0.X.LL 1.487285 2.75e-01 0.948641 2.025929  
## x0.X.SC 3.612499 1.14e+00 1.385920 5.839078  
## x0.X.HC 2.116309 5.45e-01 1.048644 3.183974  
## x0.X.KSV 1.106298 7.76e-02 0.954217 1.258379  
## C.X.AP 0.057603 9.89e-03 0.038220 0.076985  
## C.X.SP 0.035406 8.78e-03 0.018195 0.052618  
## C.X.Durazo 0.033553 9.94e-03 0.014065 0.053040  
## C.X.LS 0.010417 7.33e-03 -0.003956 0.024789  
## C.X.May 0.028579 1.35e-02 0.002126 0.055033  
## C.X.FG 0.030953 5.18e-03 0.020806 0.041101  
## C.X.GP 0.018791 1.84e-03 0.015179 0.022403  
## C.X.UL 0.037363 6.48e-03 0.024652 0.050073  
## C.X.LL 0.037844 6.40e-03 0.025305 0.050383  
## C.X.SC 0.088024 1.37e-02 0.061163 0.114884  
## C.X.HC 0.113147 1.32e-02 0.087314 0.138981  
## C.X.KSV 0.057921 8.88e-03 0.040523 0.075318  
## Initial states (x0) defined at t=0  
##   
## CIs calculated at alpha = 0.05 via method=hessian

mod1\_condense.params

##   
## MARSS fit is  
## Estimation method: kem   
## Convergence test: conv.test.slope.tol = 0.5, abstol = 0.001  
## Estimation converged in 153 iterations.   
## Log-likelihood: 4857.285   
## AIC: -9618.57 AICc: -9617.484   
##   
## ML.Est Std.Err low.CI up.CI  
## B.(X.AP,X.AP) 0.942858 9.98e-03 0.923290 0.962425  
## B.(X.SP,X.SP) 0.968046 8.89e-03 0.950627 0.985464  
## B.(X.Durazo,X.Durazo) 0.967180 1.01e-02 0.947471 0.986889  
## B.(X.LS,X.LS) 0.987704 7.41e-03 0.973183 1.002224  
## B.(X.May,X.May) 0.966777 1.29e-02 0.941399 0.992155  
## B.(X.FG,X.FG) 0.972812 5.22e-03 0.962583 0.983041  
## B.(X.GP,X.GP) 0.987907 1.86e-03 0.984254 0.991560  
## B.(X.UL,X.UL) 0.965388 6.54e-03 0.952573 0.978204  
## B.(X.LL,X.LL) 0.965301 6.46e-03 0.952634 0.977969  
## B.(X.SC,X.SC) 0.911952 1.37e-02 0.885028 0.938876  
## B.(X.HC,X.HC) 0.890677 1.33e-02 0.864686 0.916667  
## B.(X.KSV,X.KSV) 0.945554 8.96e-03 0.927994 0.963113  
## Q.(X.AP,X.AP) 0.008070 5.94e-04 0.006906 0.009235  
## Q.(X.SP,X.SP) 0.006837 5.04e-04 0.005849 0.007825  
## Q.(X.Durazo,X.Durazo) 0.008850 6.54e-04 0.007567 0.010132  
## Q.(X.LS,X.LS) 0.005552 4.10e-04 0.004748 0.006357  
## Q.(X.May,X.May) 0.024014 1.84e-03 0.020409 0.027618  
## Q.(X.FG,X.FG) 0.003015 2.22e-04 0.002579 0.003450  
## Q.(X.GP,X.GP) 0.000809 5.96e-05 0.000692 0.000926  
## Q.(X.UL,X.UL) 0.004623 3.39e-04 0.003958 0.005289  
## Q.(X.LL,X.LL) 0.005431 4.00e-04 0.004648 0.006215  
## Q.(X.SC,X.SC) 0.015851 1.17e-03 0.013554 0.018148  
## Q.(X.HC,X.HC) 0.013369 9.82e-04 0.011445 0.015293  
## Q.(X.KSV,X.KSV) 0.005327 4.00e-04 0.004544 0.006111  
## x0.X.AP 2.471342 4.56e-01 1.577050 3.365633  
## x0.X.SP 1.288391 3.47e-01 0.607769 1.969013  
## x0.X.Durazo 1.695049 4.64e-01 0.785959 2.604138  
## x0.X.LS 1.695107 3.16e-01 1.075835 2.314380  
## x0.X.May 1.933986 7.65e-01 0.434243 3.433728  
## x0.X.FG 1.296982 2.09e-01 0.887063 1.706901  
## x0.X.GP 0.632370 9.13e-02 0.453480 0.811259  
## x0.X.UL 1.631588 2.17e-01 1.206646 2.056529  
## x0.X.LL 1.487285 2.75e-01 0.948641 2.025929  
## x0.X.SC 3.612499 1.14e+00 1.385920 5.839078  
## x0.X.HC 2.116309 5.45e-01 1.048644 3.183974  
## x0.X.KSV 1.106298 7.76e-02 0.954217 1.258379  
## C.X.AP 0.057603 9.89e-03 0.038220 0.076985  
## C.X.SP 0.035406 8.78e-03 0.018195 0.052618  
## C.X.Durazo 0.033553 9.94e-03 0.014065 0.053040  
## C.X.LS 0.010417 7.33e-03 -0.003956 0.024789  
## C.X.May 0.028579 1.35e-02 0.002126 0.055033  
## C.X.FG 0.030953 5.18e-03 0.020806 0.041101  
## C.X.GP 0.018791 1.84e-03 0.015179 0.022403  
## C.X.UL 0.037363 6.48e-03 0.024652 0.050073  
## C.X.LL 0.037844 6.40e-03 0.025305 0.050383  
## C.X.SC 0.088024 1.37e-02 0.061163 0.114884  
## C.X.HC 0.113147 1.32e-02 0.087314 0.138981  
## C.X.KSV 0.057921 8.88e-03 0.040523 0.075318  
## Initial states (x0) defined at t=0  
##   
## CIs calculated at alpha = 0.05 via method=hessian

mod1\_condense\_df <- broom::tidy(mod1\_condense.fit)  
  
png("Fig\_AirTempEffects.png", width = 700, height = 400)  
labels <- c("Alexander", "Durazo", "FishGulch", "Goodman","HorseCreek","Klamath","LLawrence","LowerSeiad","May","SeiadCreek","Stender","ULawrence")  
ggplot(data = mod1\_condense\_df) +   
 geom\_pointrange(data = mod1\_condense\_df[c(37:48),], aes(x = term, y = estimate, ymin = conf.low, ymax = conf.up), color = "black") +  
 geom\_hline(yintercept = 0) +  
 labs(x = "Sites",  
 y = "Air Temperature Effects") +  
 ggtitle("Air Temperature Effects") +  
 theme\_classic()+  
 theme(text=element\_text(size=20),axis.text.x=element\_text(angle = 90, hjust = 1))+  
 scale\_x\_discrete(labels= labels)  
dev.off()

## png   
## 2

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(mod4\_condense.fit, type = "tt1")$model.residuals[j, ],  
 ylab = "Residual")  
 abline(h = 0, lty = "dashed")  
 acf(residuals,na.action = na.pass)  
 }

