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_conflict
s() --
## 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
```

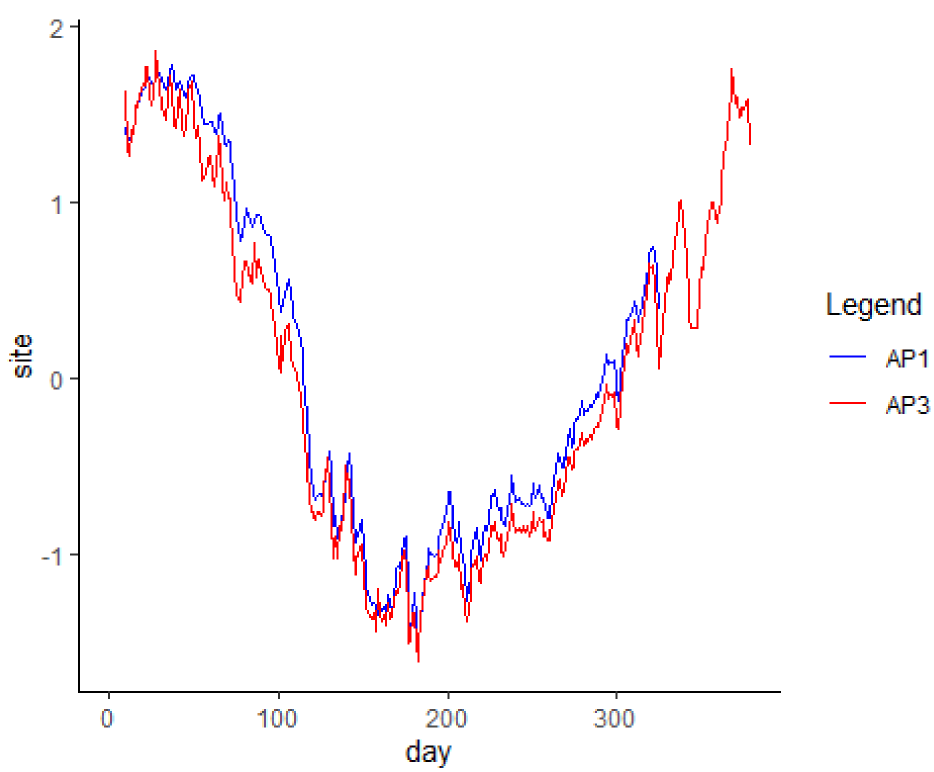
```
covariate_klamath <- readRDS('covariate.rds')
transformed_covariate_klamath <- zscore(covariate_klamath)
```

Steps:

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

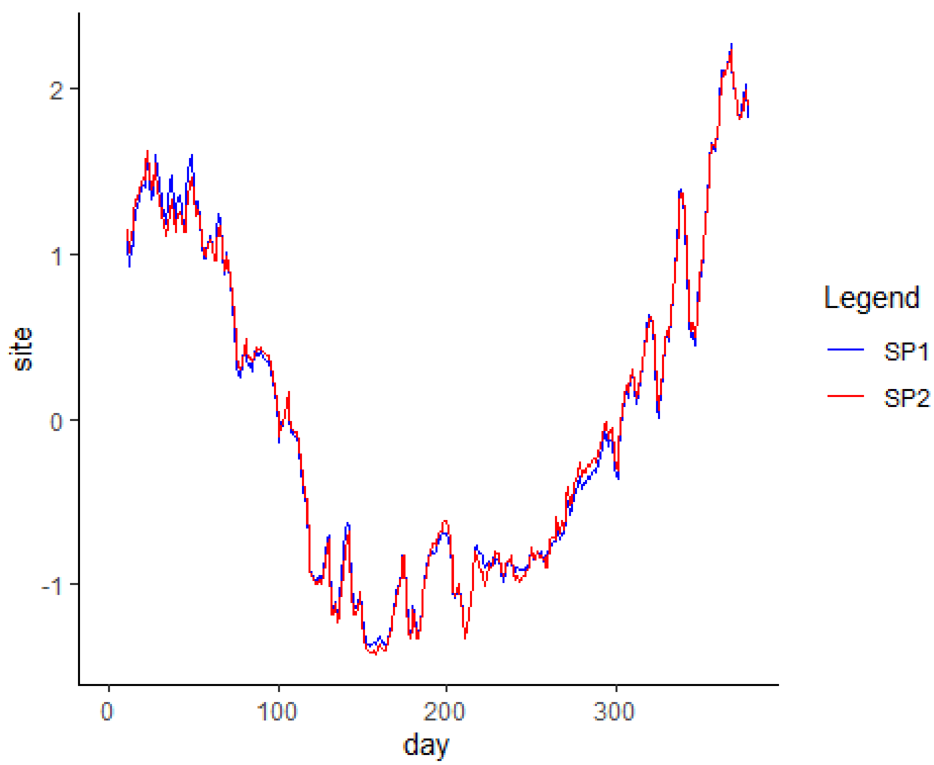
2) Then take mean across replicate sensors, so that we end up with 1 time series per habitat, 2+9+1 (12 x 378)

```
#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).
```



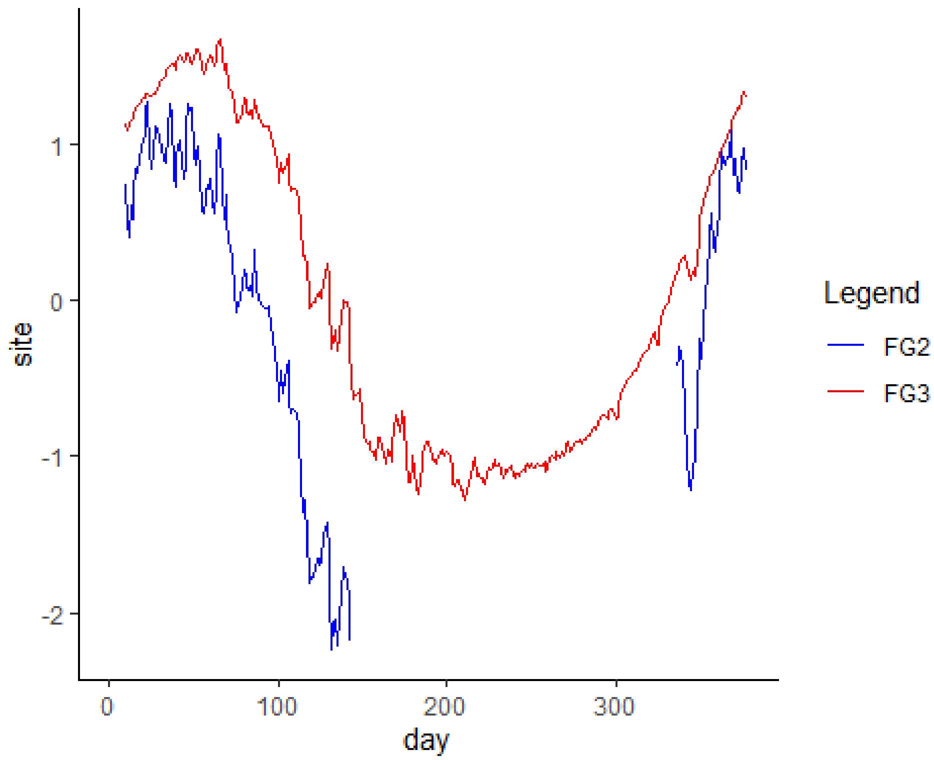

```
#need to remove AP1
```

```
#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).
```

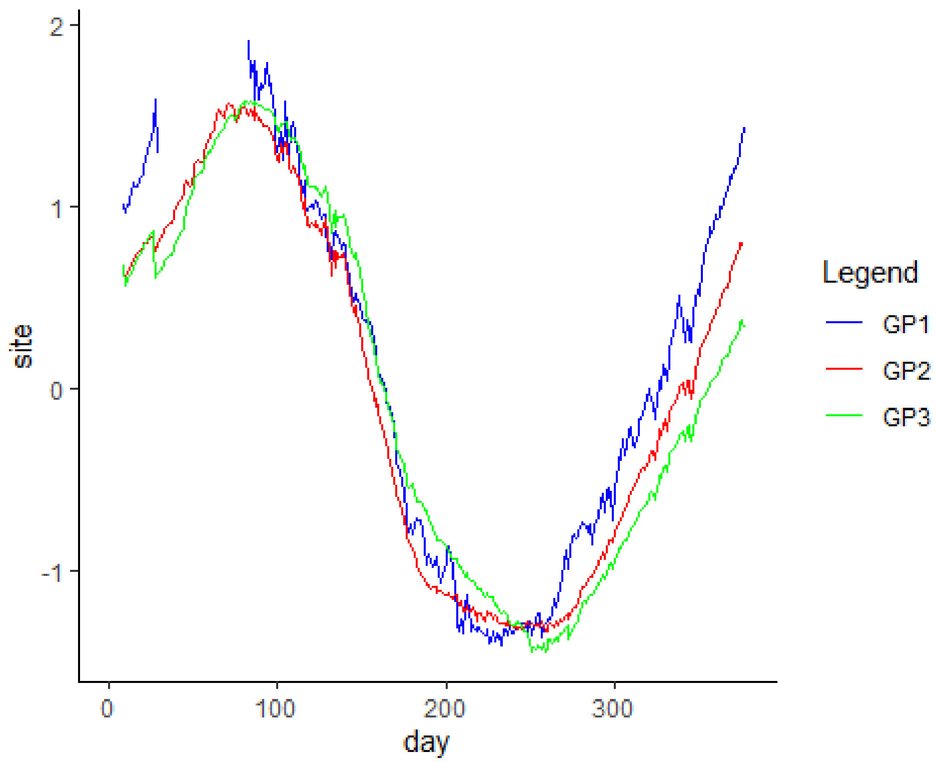



```
#both good, need to take average
transformed_dat_klamath_df$SP <- (transformed_dat_klamath_df$SP1 + transforme
d_dat_klamath_df$SP2)/2

#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"))
```

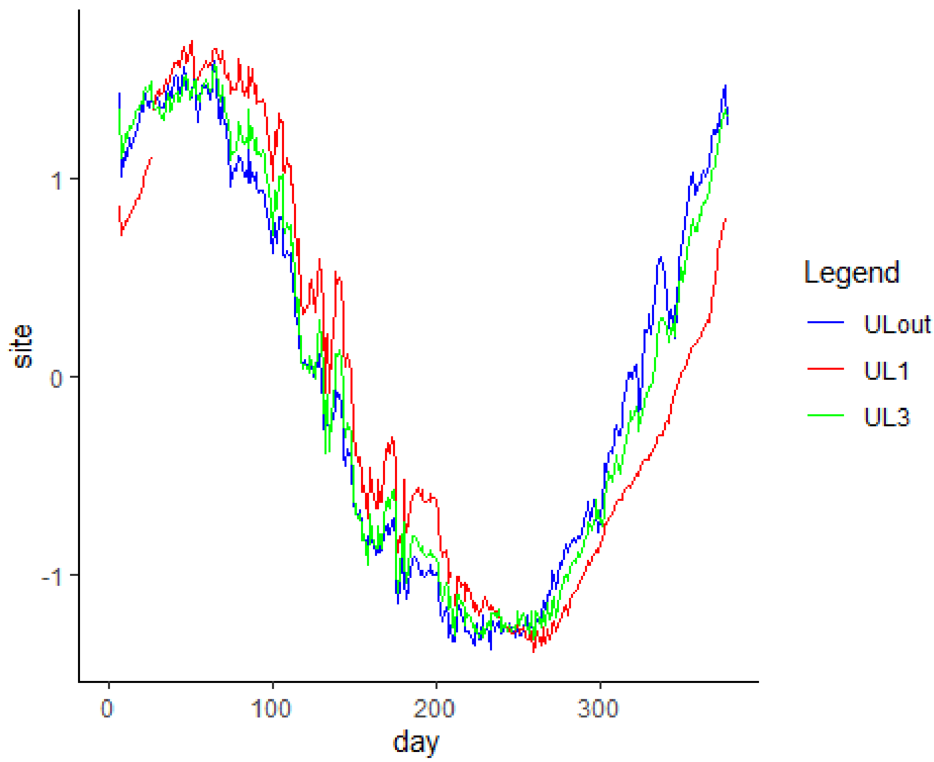
```
## Warning: Removed 10 row(s) containing missing values (geom_path).

## Warning: Removed 10 row(s) containing missing values (geom_path).
```



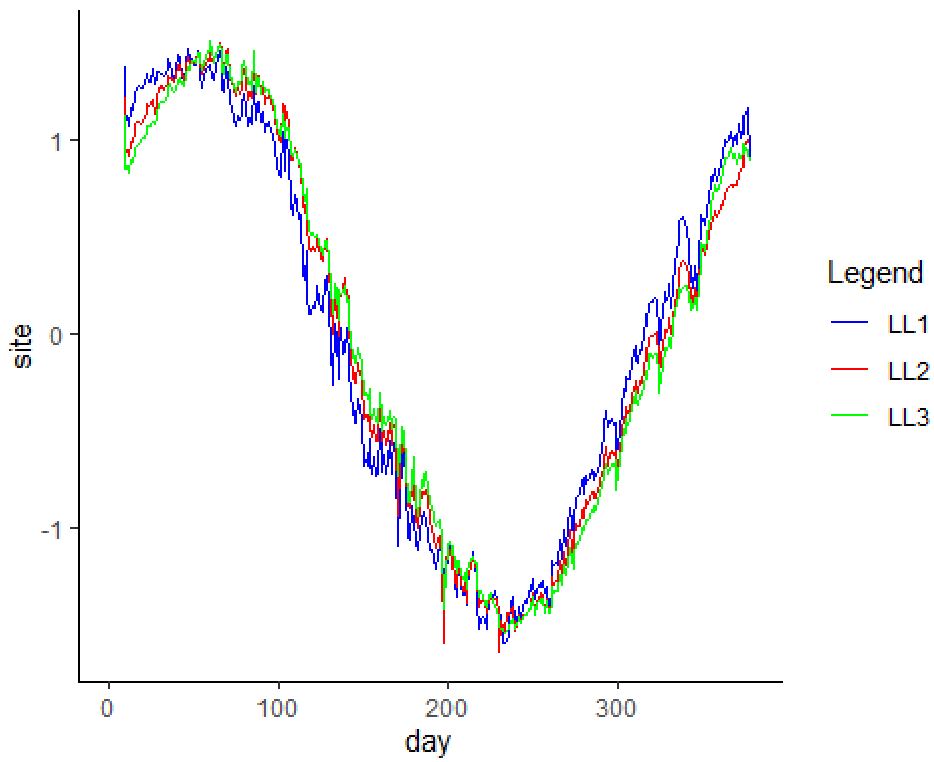

```
#need to remove FG2

#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).
```



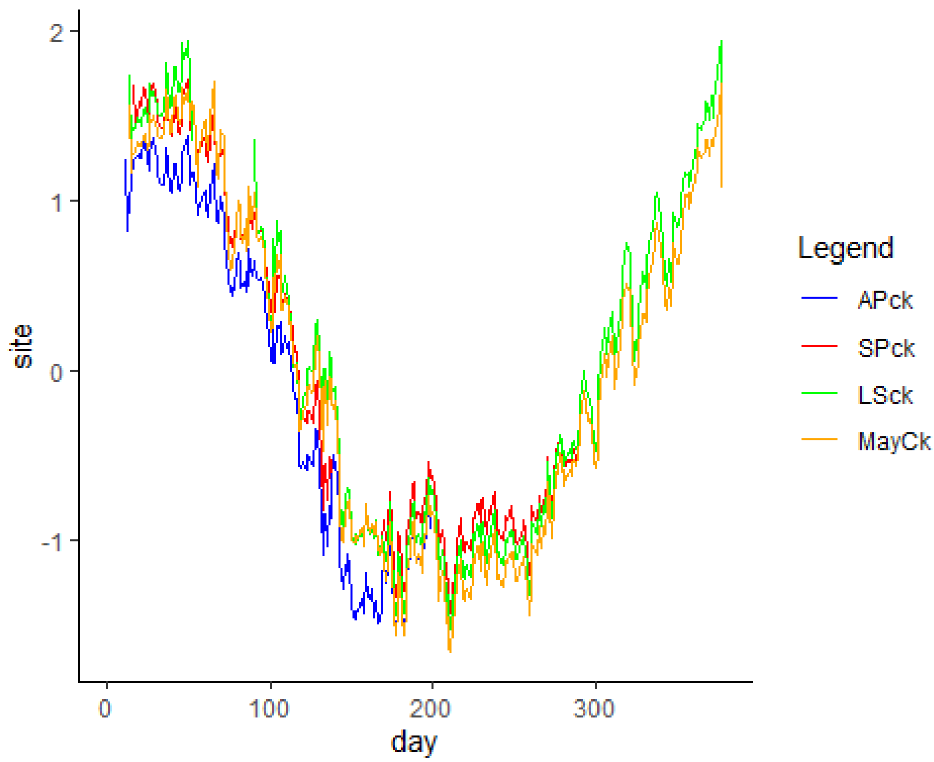

```r
#need to remove GP1
transformed_dat_klamath_df$GP <- (transformed_dat_klamath_df$GP2 + transforme
d_dat_klamath_df$GP3)/2

#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).
```



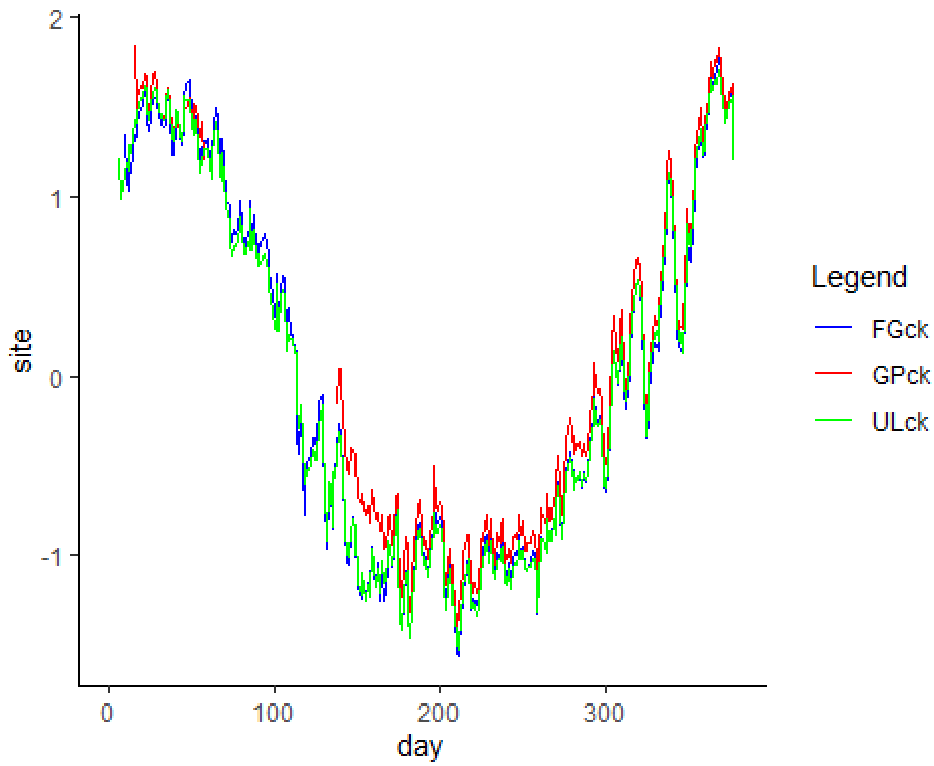

```
#need to remove UL1
transformed_dat_klamath_df$UL <- (transformed_dat_klamath_df$ULout + transfor
med_dat_klamath_df$UL3)/2

#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$LL <- (transformed_dat_klamath_df$LL1 + transforme
d_dat_klamath_df$LL2 + transformed_dat_klamath_df$LL3)/3

#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$SC <- (transformed_dat_klamath_df$LSck + transform
ed_dat_klamath_df$MayCk)/2

#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$HC <- (transformed_dat_klamath_df$FGck + transform
ed_dat_klamath_df$ULck)/2
```

#Matrix of transformed data with 12 sites (condensed_transdat)

```
condensed_transdat_df <- cbind(AP = transformed_dat_klamath_df$AP3, SP = tran
sformed_dat_klamath_df$SP, Durazo = transformed_dat_klamath_df$Durazo, LS = t
ransformed_dat_klamath_df$LS, May = transformed_dat_klamath_df$May, FG = tran
sformed_dat_klamath_df$FG3, GP = transformed_dat_klamath_df$GP, UL = transfor
med_dat_klamath_df$UL, LL = transformed_dat_klamath_df$LL, SC = transformed_d
at_klamath_df$SC, HC = transformed_dat_klamath_df$HC, KSV = transformed_dat_k
lamath_df$KSV)
condensed_transdat <- as.matrix(t(condensed_transdat_df))
str(condensed_transdat)

##  num [1:12, 1:378] NA NA NA NA NA NA NA NA NA NA ...
##  - 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$R = "zero" #all the sensors are same, so observation error shou
ld be same
mod1_condense$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)
```

```
## Success! abstol and log-log tests passed at 141 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 141 iterations.
## Log-likelihood: 4848.501
## AIC: -9601.003   AICc: -9599.906
##
##                          Estimate
## B.(X.AP,X.AP)            0.942875
## B.(X.SP,X.SP)            0.968046
## B.(X.Durazo,X.Durazo)    0.967188
## B.(X.LS,X.LS)            0.987706
## B.(X.May,X.May)          0.966784
## 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)            0.965302
## B.(X.SC,X.SC)            0.920376
## B.(X.HC,X.HC)            0.891281
## 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)         0.014274
## Q.(X.HC,X.HC)         0.013268
## Q.(X.KSV,X.KSV)       0.005327
## x0.X.AP               2.469808
## x0.X.SP               1.288292
## x0.X.Durazo           1.693972
## x0.X.LS               1.694906
## x0.X.May              1.932740
## x0.X.FG               1.296963
## x0.X.GP               0.632369
## x0.X.UL               1.631587
## x0.X.LL               1.487274
## x0.X.SC               3.274469
## x0.X.HC               2.572893
## x0.X.KSV              1.106298
## C.X.AP                0.057589
## C.X.SP                0.035406
## C.X.Durazo            0.033547
## C.X.LS                0.010415
## C.X.May               0.028575
## C.X.FG                0.030953
## C.X.GP                0.018791
## C.X.UL                0.037363
## C.X.LL                0.037844
## C.X.SC                0.082593
## C.X.HC                0.112211
## 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 141 iterations.
## Log-likelihood: 4848.501
## AIC: -9601.003   AICc: -9599.906
##
##                           ML.Est  Std.Err    low.CI    up.CI
## B.(X.AP,X.AP)         0.942875 9.98e-03  0.923307 0.962442
## B.(X.SP,X.SP)         0.968046 8.89e-03  0.950628 0.985464
## B.(X.Durazo,X.Durazo) 0.967188 1.01e-02  0.947479 0.986897
## B.(X.LS,X.LS)         0.987706 7.41e-03  0.973185 1.002226
## B.(X.May,X.May)       0.966784 1.29e-02  0.941406 0.992162
## 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)         0.965302 6.46e-03  0.952634 0.977969
## B.(X.SC,X.SC)         0.920376 1.34e-02  0.894165 0.946586
## B.(X.HC,X.HC)         0.891281 1.33e-02  0.865288 0.917274
## 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)         0.014274 1.11e-03  0.012093 0.016455
## Q.(X.HC,X.HC)         0.013268 9.78e-04  0.011351 0.015185
## Q.(X.KSV,X.KSV)       0.005327 4.00e-04  0.004544 0.006111
## x0.X.AP               2.469808 4.56e-01  1.575747 3.363869
## x0.X.SP               1.288292 3.47e-01  0.607678 1.968907
## x0.X.Durazo           1.693972 4.64e-01  0.785031 2.602912
## x0.X.LS               1.694906 3.16e-01  1.075654 2.314157
## x0.X.May              1.932740 7.65e-01  0.433209 3.432270
## x0.X.FG               1.296963 2.09e-01  0.887045 1.706882
## x0.X.GP               0.632369 9.13e-02  0.453480 0.811259
## x0.X.UL               1.631587 2.17e-01  1.206646 2.056529
## x0.X.LL               1.487274 2.75e-01  0.948631 2.025918
## x0.X.SC               3.274469 9.89e-01  1.336580 5.212358
## x0.X.HC               2.572893 8.24e-01  0.957794 4.187993
## x0.X.KSV              1.106298 7.76e-02  0.954217 1.258379
## C.X.AP                0.057589 9.89e-03  0.038206 0.076971
## C.X.SP                0.035406 8.78e-03  0.018195 0.052617
## C.X.Durazo            0.033547 9.94e-03  0.014059 0.053034
## C.X.LS                0.010415 7.33e-03 -0.003957 0.024788
## C.X.May               0.028575 1.35e-02  0.002122 0.055029
## 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                0.082593 1.33e-02  0.056465 0.108721
## C.X.HC                0.112211 1.31e-02  0.086439 0.137983
## 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
```

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$R = "zero" #all the sensors are same, so observation error shou
ld be same
mod2_condense$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)
```

```
## 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: 4193.148
## AIC: -8356.295   AICc: -8356.184
##
##             Estimate
## B.diag        0.9632
## Q.diag        0.0084
## x0.X.AP       2.0931
## x0.X.SP       1.3314
## x0.X.Durazo   1.7112
## x0.X.LS       2.0103
## x0.X.May      1.8911
## x0.X.FG       1.3816
## x0.X.GP       0.6887
## x0.X.UL       1.6519
## x0.X.LL       1.5104
## x0.X.SC       2.1999
## x0.X.HC       1.5357
## x0.X.KSV      1.0923
## C.1           0.0388
## 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: 4193.148
## AIC: -8356.295   AICc: -8356.184
##
##             ML.Est Std.Err  low.CI   up.CI
## B.diag      0.9632 0.00251 0.95832 0.96814
## Q.diag      0.0084 0.00018 0.00805 0.00876
## x0.X.AP     2.0931 0.36301 1.38157 2.80454
## x0.X.SP     1.3314 0.38696 0.57302 2.08986
## x0.X.Durazo 1.7112 0.44018 0.84846 2.57393
## x0.X.LS     2.0103 0.44120 1.14556 2.87503
## x0.X.May    1.8911 0.44077 1.02721 2.75499
## x0.X.FG     1.3816 0.36131 0.67345 2.08976
## x0.X.GP     0.6887 0.33496 0.03219 1.34519
## x0.X.UL     1.6519 0.28435 1.09464 2.20926
## x0.X.LL     1.5104 0.33587 0.85215 2.16873
## x0.X.SC     2.1999 0.44197 1.33363 3.06611
## x0.X.HC     1.5357 0.36160 0.82702 2.24446
## x0.X.KSV    1.0923 0.09520 0.90569 1.27888
## C.1         0.0388 0.00250 0.03394 0.04374
## Initial states (x0) defined at t=0
##
## CIs calculated at alpha = 0.05 via method=hessian
```

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$R = "zero" #all the sensors are same, so observation error shou
ld be same
mod3_condense$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)
mod3_condense.params =MARSSparamCIs(mod3_condense.fit)
MARSSparamCIs(mod3_condense.fit)
```

Hypothesis 4: By watershed

AICc: -8597.364

```
mod4_condense = list()
## Modify matrices
# 1st: group time series into categories
rownames(condensed_transdat)
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$R = "zero" #all the sensors are same, so observation error shou
ld be same
mod4_condense$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)
mod4_condense.params =MARSSparamCIs(mod4_condense.fit)
MARSSparamCIs(mod4_condense.fit)
```