MARSS model checking

###2020 MARSS model outputs: July 7 - July 28 2020 (airtemp as a covariate, but not z scored. No seasonal correction)

```
mod1.fit.2020 <- readRDS("2020_mod1.fit.rds")
mod1.params.2020 <- readRDS("2020_mod1.params.rds")
mod2.fit.2020 <- readRDS("2020_mod2.fit.rds")
mod2.params.2020 <- readRDS("2020_mod2.params.rds")
mod3.fit.2020 <- readRDS("2020_mod3.fit.rds")
mod3.params.2020 <- readRDS("2020_mod3.params.rds")
mod4.fit.2020 <- readRDS("2020_mod4.fit.rds")
mod4.params.2020 <- readRDS("2020_mod4.params.rds")

#Model 1, hypothesis 1 (all separate)
mod1.params.2020
```

```
##
## MARSS fit is
## Estimation method: kem
## Convergence test: conv.test.slope.tol = 0.5, abstol = 0.001
## Estimation converged in 350 iterations.
## Log-likelihood: -333.3257
## AIC: 844.6514   AICc: 880.6514
##
##              ML.Est Std.Err    low.CI    up.CI
## R.diag     0.132351 0.00973  0.113287  0.15142
## Q.(1,1)    0.169511 0.06546  0.041211  0.29781
## Q.(2,1)    0.269266 0.09725  0.078654  0.45988
## Q.(3,1)    0.551157 0.21646  0.126904  0.97541
## Q.(4,1)    0.320063 0.13377  0.057874  0.58225
## Q.(5,1)    0.206263 0.08157  0.046388  0.36614
## Q.(6,1)    0.189997 0.07636  0.040334  0.33966
## Q.(7,1)    0.181741 0.06591  0.052554  0.31093
## Q.(8,1)    0.039687 0.03210 -0.023227  0.10260
## Q.(9,1)    0.129886 0.05633  0.019474  0.24030
## Q.(10,1)   0.158694 0.09329 -0.024150  0.34154
## Q.(11,1)   0.199095 0.07911  0.044036  0.35415
## Q.(2,2)    0.458918 0.16361  0.138243  0.77959
## Q.(3,2)    1.023548 0.36209  0.313857  1.73324
## Q.(4,2)    0.568621 0.21974  0.137936  0.99931
## Q.(5,2)    0.340651 0.13281  0.080351  0.60095
## Q.(6,2)    0.294280 0.12139  0.056354  0.53221
## Q.(7,2)    0.287451 0.10312  0.085341  0.48956
## Q.(8,2)    0.060401 0.05159 -0.040714  0.16152
## Q.(9,2)    0.229238 0.09072  0.051434  0.40704
## Q.(10,2)   0.238323 0.14519 -0.046246  0.52289
```

```
## Q.(11,2)   0.326268 0.12495  0.081368  0.57117
## Q.(3,3)    2.791892 0.92486  0.979201  4.60458
## Q.(4,3)    1.429326 0.53298  0.384709  2.47394
## Q.(5,3)    0.734407 0.30632  0.134021  1.33479
## Q.(6,3)    0.620703 0.27956  0.072772  1.16863
## Q.(7,3)    0.609604 0.23297  0.152988  1.06622
## Q.(8,3)    0.229998 0.13244 -0.029583  0.48958
## Q.(9,3)    0.661732 0.23183  0.207356  1.11611
## Q.(10,3)   0.734578 0.35787  0.033168  1.43599
## Q.(11,3)   0.668870 0.28340  0.113410  1.22433
## Q.(4,4)    1.004484 0.39173  0.236716  1.77225
## Q.(5,4)    0.502737 0.20510  0.100740  0.90473
## Q.(6,4)    0.416306 0.18364  0.056385  0.77623
## Q.(7,4)    0.379788 0.14891  0.087924  0.67165
## Q.(8,4)    0.109839 0.08146 -0.049814  0.26949
## Q.(9,4)    0.406402 0.14972  0.112960  0.69984
## Q.(10,4)   0.325371 0.22041 -0.106627  0.75737
## Q.(11,4)   0.442714 0.18375  0.082562  0.80287
## Q.(5,5)    0.291020 0.13224  0.031827  0.55021
## Q.(6,5)    0.256269 0.10805  0.044495  0.46804
## Q.(7,5)    0.229627 0.08874  0.055702  0.40355
## Q.(8,5)    0.053372 0.04613 -0.037035  0.14378
## Q.(9,5)    0.190825 0.08223  0.029663  0.35199
## Q.(10,5)   0.198902 0.13404 -0.063804  0.46161
## Q.(11,5)   0.247582 0.10705  0.037764  0.45740
## Q.(6,6)    0.247937 0.11479  0.022954  0.47292
## Q.(7,6)    0.210706 0.08308  0.047876  0.37354
## Q.(8,6)    0.070173 0.04447 -0.016988  0.15733
## Q.(9,6)    0.168353 0.07629  0.018836  0.31787
## Q.(10,6)   0.257721 0.13323 -0.003402  0.51884
## Q.(11,6)   0.194530 0.09693  0.004544  0.38452
## Q.(7,7)    0.201739 0.07634  0.052110  0.35137
## Q.(8,7)    0.049511 0.03553 -0.020128  0.11915
## Q.(9,7)    0.159521 0.06330  0.035454  0.28359
## Q.(10,7)   0.175293 0.10050 -0.021679  0.37227
## Q.(11,7)   0.227486 0.08704  0.056884  0.39809
## Q.(8,8)    0.060126 0.02908  0.003129  0.11712
## Q.(9,8)    0.074594 0.03801  0.000103  0.14909
## Q.(10,8)   0.182634 0.07120  0.043090  0.32218
## Q.(11,8)   0.010282 0.04342 -0.074828  0.09539
## Q.(9,9)    0.196877 0.07175  0.056258  0.33750
## Q.(10,9)   0.202934 0.09863  0.009613  0.39625
## Q.(11,9)   0.179402 0.07832  0.025904  0.33290
## Q.(10,10)  0.613988 0.21605  0.190540  1.03744
## Q.(11,10)  0.017973 0.11857 -0.214422  0.25037
## Q.(11,11)  0.344855 0.12651  0.096901  0.59281
## x0.X1     -0.242124 0.68975 -1.594005  1.10976
## x0.X2      1.192021 1.16661 -1.094495  3.47854
## x0.X3      6.321549 2.79007  0.853112 11.78999
## x0.X4      1.943099 1.93640 -1.852179  5.73838
## x0.X5     -0.301160 1.23304 -2.717864  2.11555
## x0.X6     -0.166793 1.17772 -2.475074  2.14149
## x0.X7     -0.146467 0.68792 -1.494759  1.20183
## x0.X8     -0.799341 0.45245 -1.686133  0.08745
```

```
## x0.X9     -0.153413 0.48484 -1.103677  0.79685
## x0.X10     2.739076 1.34954  0.094034  5.38412
## x0.X11    -0.822383 0.67862 -2.152451  0.50768
## C.X1       0.000722 0.00136 -0.001943  0.00339
## C.X2      -0.000399 0.00222 -0.004753  0.00395
## C.X3      -0.004292 0.00544 -0.014954  0.00637
## C.X4      -0.001681 0.00330 -0.008152  0.00479
## C.X5       0.000566 0.00185 -0.003056  0.00419
## C.X6       0.000356 0.00173 -0.003029  0.00374
## C.X7       0.000511 0.00148 -0.002396  0.00342
## C.X8       0.001427 0.00085 -0.000239  0.00309
## C.X9       0.000755 0.00144 -0.002061  0.00357
## C.X10     -0.001746 0.00261 -0.006865  0.00337
## C.X11      0.000697 0.00191 -0.003045  0.00444
## Initial states (x0) defined at t=0
##
## CIs calculated at alpha = 0.05 via method=hessian
```

```r
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.fit.2020, type = "tt1")$model.residuals[j, ],
            ylab = "Residual")
    abline(h = 0, lty = "dashed")
    acf(residuals,na.action = na.pass)
  }
```

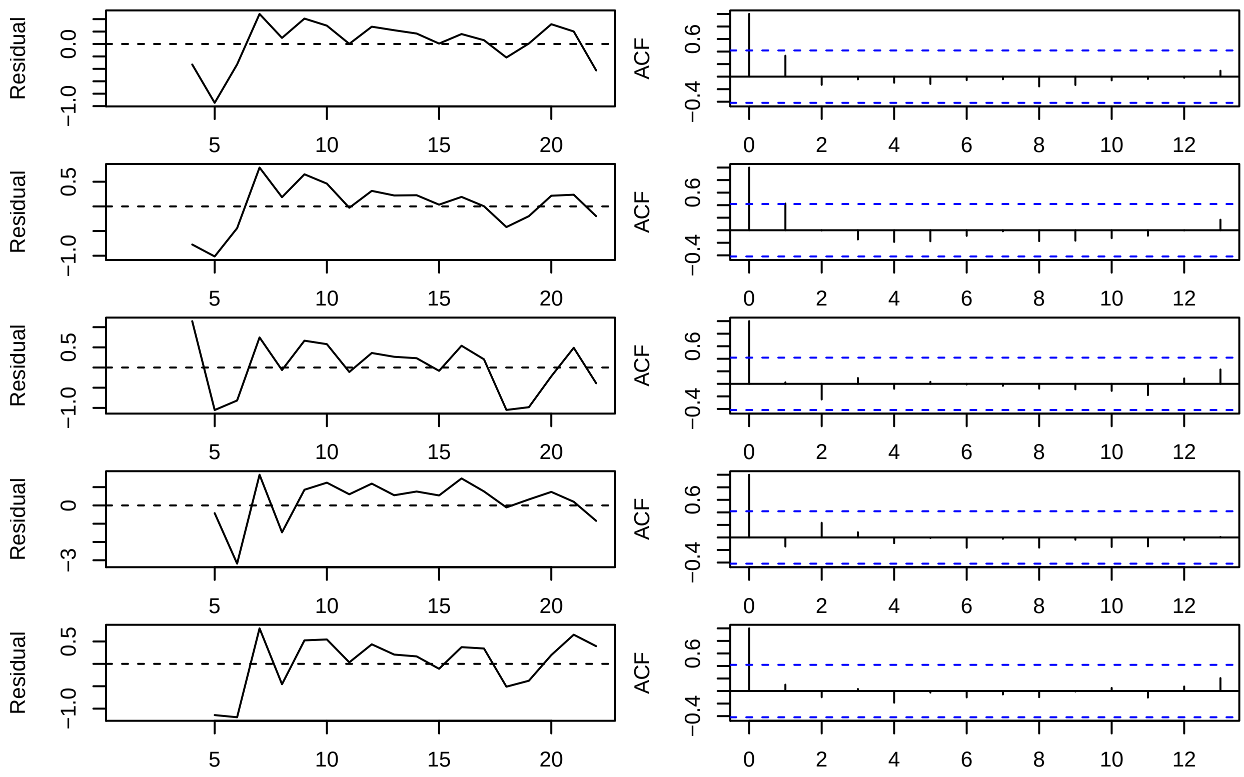
```
## MARSSresiduals.tt1 reported warnings. See msg element of returned residuals object.
```

```
## MARSSresiduals.tt1 reported warnings. See msg element of returned residuals object.
```

```
## MARSSresiduals.tt1 reported warnings. See msg element of returned residuals object.
```

```
## MARSSresiduals.tt1 reported warnings. See msg element of returned residuals object.
```

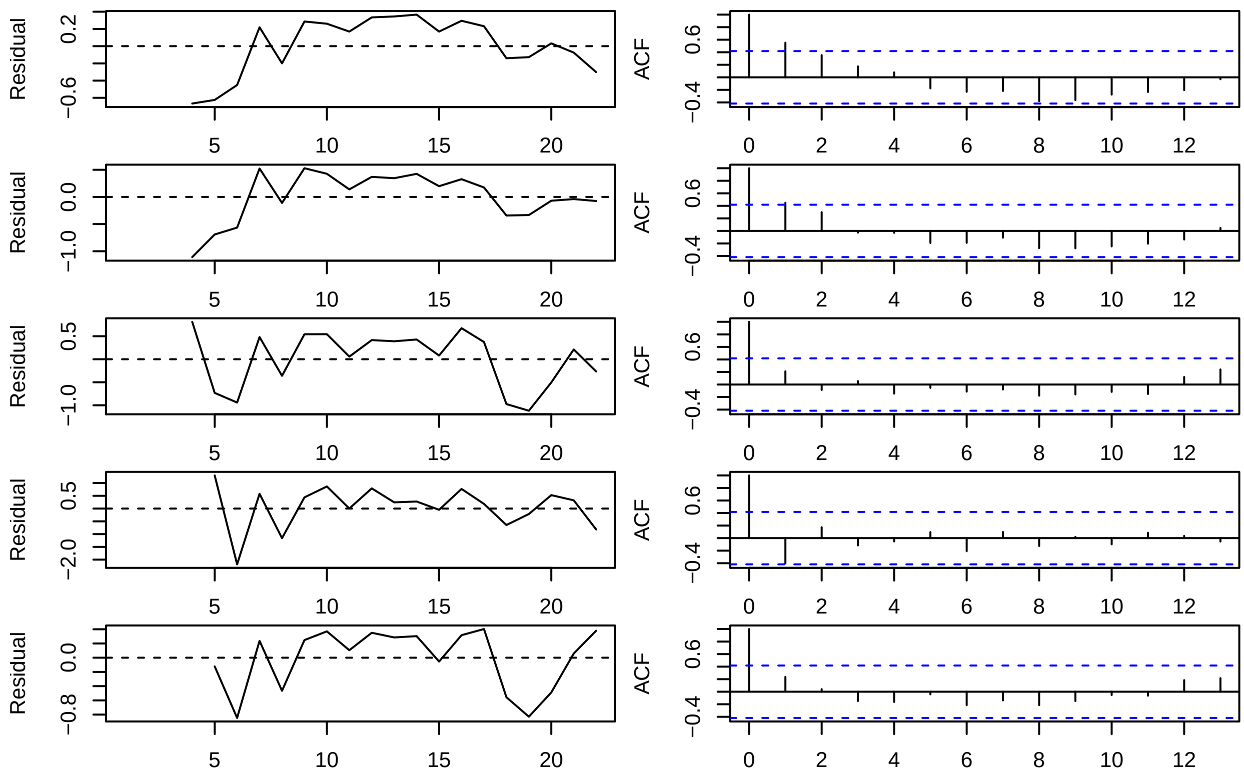
```
## MARSSresiduals.tt1 reported warnings. See msg element of returned residuals object.
```



```r
#Model 2, hypothesis 2 (creeks vs ponds)
mod2.params.2020
```

```
##
## MARSS fit is
## Estimation method: kem
## Convergence test: conv.test.slope.tol = 0.5, abstol = 0.001
## Algorithm ran 15 (=minit) iterations and convergence was reached.
## Log-likelihood: -445.091
## AIC: 906.1821   AICc: 906.4559
##
##              ML.Est Std.Err   low.CI   up.CI
## R.diag     0.256155 0.01629  0.22424 0.28807
## Q.(1,1)    0.181217 0.06157  0.06053 0.30190
## Q.(2,1)    0.294661 0.10878  0.08145 0.50787
## Q.(2,2)    0.718379 0.24935  0.22967 1.20709
## x0.X1     -0.308081 0.48608 -1.26078 0.64462
## x0.X2     -0.615078 0.96876 -2.51381 1.28365
## C.X1       0.000513 0.00137 -0.00217 0.00320
## C.X2      -0.000112 0.00273 -0.00545 0.00523
## 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.fit.2020, type = "tt1")$model.residuals[j, ],
            ylab = "Residual")
    abline(h = 0, lty = "dashed")
    acf(residuals,na.action = na.pass)
  }
```




```
#Model 3, hypothesis 3 (trib vs. trib)
mod3.params.2020
```

```
##
## MARSS fit is
## Estimation method: kem
## Convergence test: conv.test.slope.tol = 0.5, abstol = 0.001
## Estimation converged in 279 iterations.
## Log-likelihood: -419.1611
## AIC: 876.3223   AICc: 877.798
##
##             ML.Est Std.Err   low.CI    up.CI
## R.diag    0.217524 0.01412  0.18984  0.24520
## Q.(1,1)   0.319266 0.11019  0.10330  0.53523
## Q.(2,1)   0.825437 0.28866  0.25967  1.39121
## Q.(3,1)   0.222331 0.07825  0.06896  0.37570
## Q.(4,1)   0.243680 0.09351  0.06041  0.42695
```

```
## Q.(2,2)  2.664947 0.89986  0.90125  4.42865
## Q.(3,2)  0.657191 0.22573  0.21476  1.09962
## Q.(4,2)  0.510457 0.23805  0.04388  0.97703
## Q.(3,3)  0.196872 0.06786  0.06388  0.32987
## Q.(4,3)  0.131236 0.06394  0.00591  0.25656
## Q.(4,4)  0.226662 0.09534  0.03980  0.41352
## x0.X1   -0.537949 0.71215 -1.93374  0.85784
## x0.X2    5.123978 2.61182  0.00490 10.24306
## x0.X3   -0.212596 0.50179 -1.19608  0.77089
## x0.X4   -0.942995 0.61044 -2.13944  0.25345
## C.X1     0.000643 0.00182 -0.00293  0.00421
## C.X2    -0.003497 0.00532 -0.01391  0.00692
## C.X3     0.000691 0.00143 -0.00212  0.00350
## C.X4     0.000722 0.00156 -0.00234  0.00378
## 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.fit.2020, type = "tt1")$model.residuals[j, ],
            ylab = "Residual")
    abline(h = 0, lty = "dashed")
    acf(residuals,na.action = na.pass)
  }
```

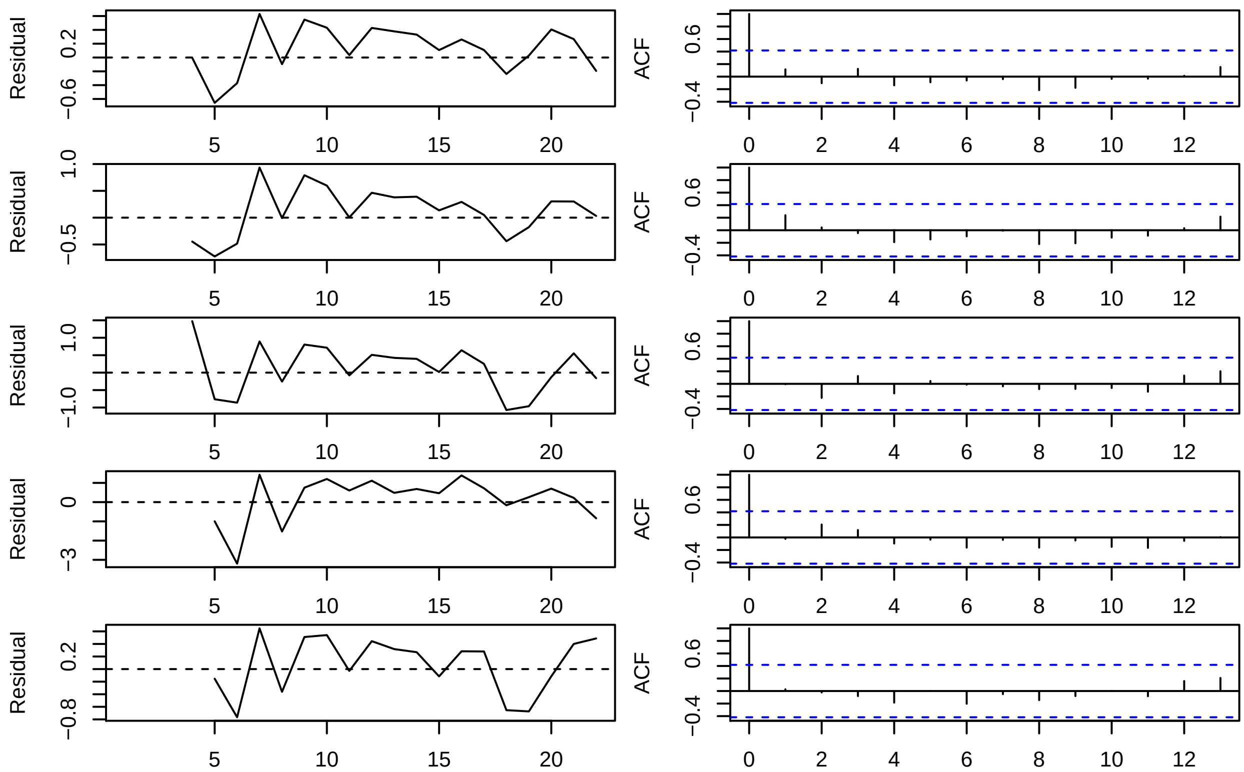
```
## MARSSresiduals.tt1 reported warnings. See msg element of returned residuals object.
```

```
## MARSSresiduals.tt1 reported warnings. See msg element of returned residuals object.
```

```
## MARSSresiduals.tt1 reported warnings. See msg element of returned residuals object.
```

```
## MARSSresiduals.tt1 reported warnings. See msg element of returned residuals object.
```

```
## MARSSresiduals.tt1 reported warnings. See msg element of returned residuals object.
```

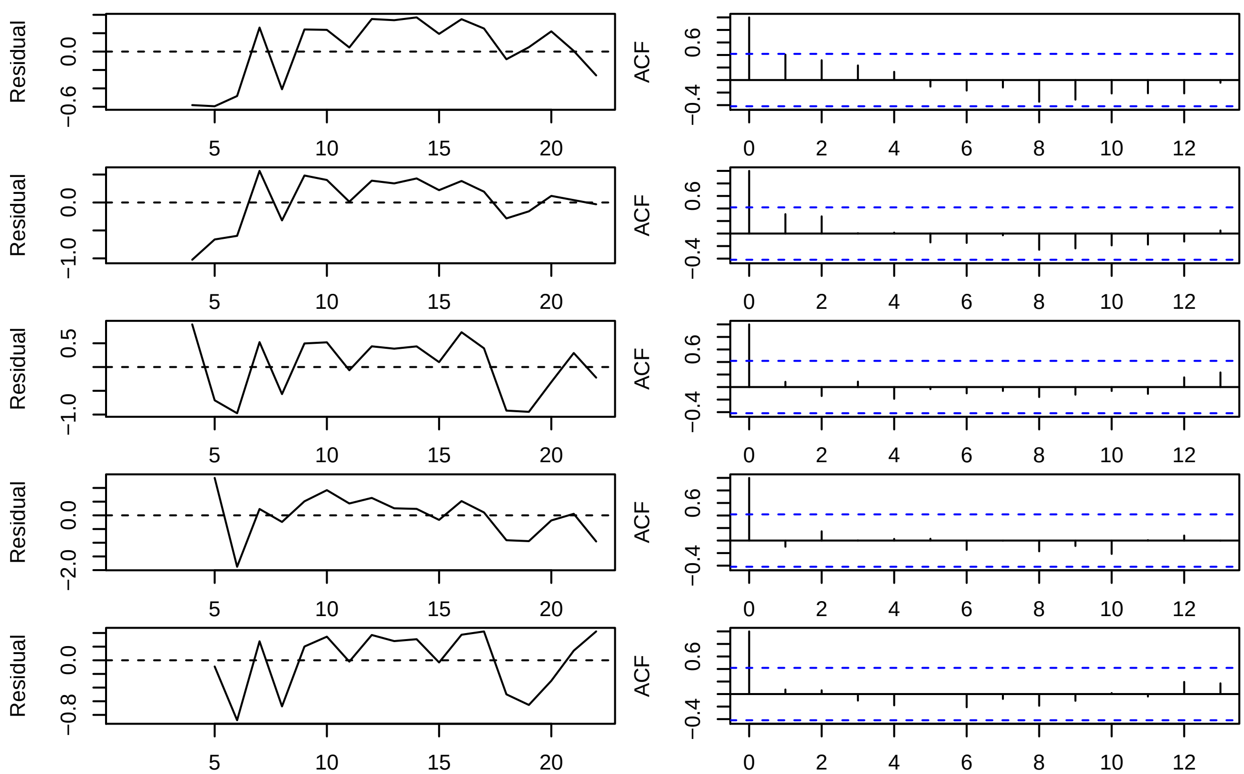



```
#Model 4, hypothesis 4 (all same)
mod4.params.2020
```

```
##
## MARSS fit is
## Estimation method: kem
## Convergence test: conv.test.slope.tol = 0.5, abstol = 0.001
## Algorithm ran 15 (=minit) iterations and convergence was reached.
## Log-likelihood: -466.3489
## AIC: 940.6978   AICc: 940.7733
##
##            ML.Est Std.Err   low.CI   up.CI
## R.diag   0.296242 0.01849  0.26001 0.33248
## Q.Q      0.233722 0.07791  0.08102 0.38643
## x0.x0   -0.346204 0.54456 -1.41351 0.72111
## C.C      0.000415 0.00155 -0.00263 0.00346
## 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(mod4.fit.2020, type = "tt1")$model.residuals[j, ],
            ylab = "Residual")
    abline(h = 0, lty = "dashed")
```

```
  acf(residuals,na.action = na.pass)
}
```



```
#...these models are actually okay, but short
```

###Comparing 2020 AICc values

```
data.frame(Model=c("Model1", "Model2", "Model3", "Model4"),
           AICc=round(c(mod1.fit.2020$AICc,
                         mod2.fit.2020$AICc,
                         mod3.fit.2020$AICc,
                         mod4.fit.2020$AICc),1))
```

```
##    Model  AICc
## 1 Model1 880.7
## 2 Model2 906.5
## 3 Model3 877.8
## 4 Model4 940.8
```

###Original MARSS model outputs (using airtemp as a covariate, airtemp not transformed, no Fourier Series correction for seasonality)

```r
mod1.fit <- readRDS("mod1.fit.rds")
mod1.params <- readRDS("mod1.params.rds")
mod2.fit <- readRDS("mod2.fit.rds")
mod2.params <- readRDS("mod2.params.rds")
mod3.fit <- readRDS("mod3.fit.rds")
mod3.params <- readRDS("mod3.params.rds")
mod4.fit <- readRDS("mod4.fit.rds")
mod4.params <- readRDS("mod4.params.rds")

#Model 1, hypothesis 1 (all separate)
mod1.params
```

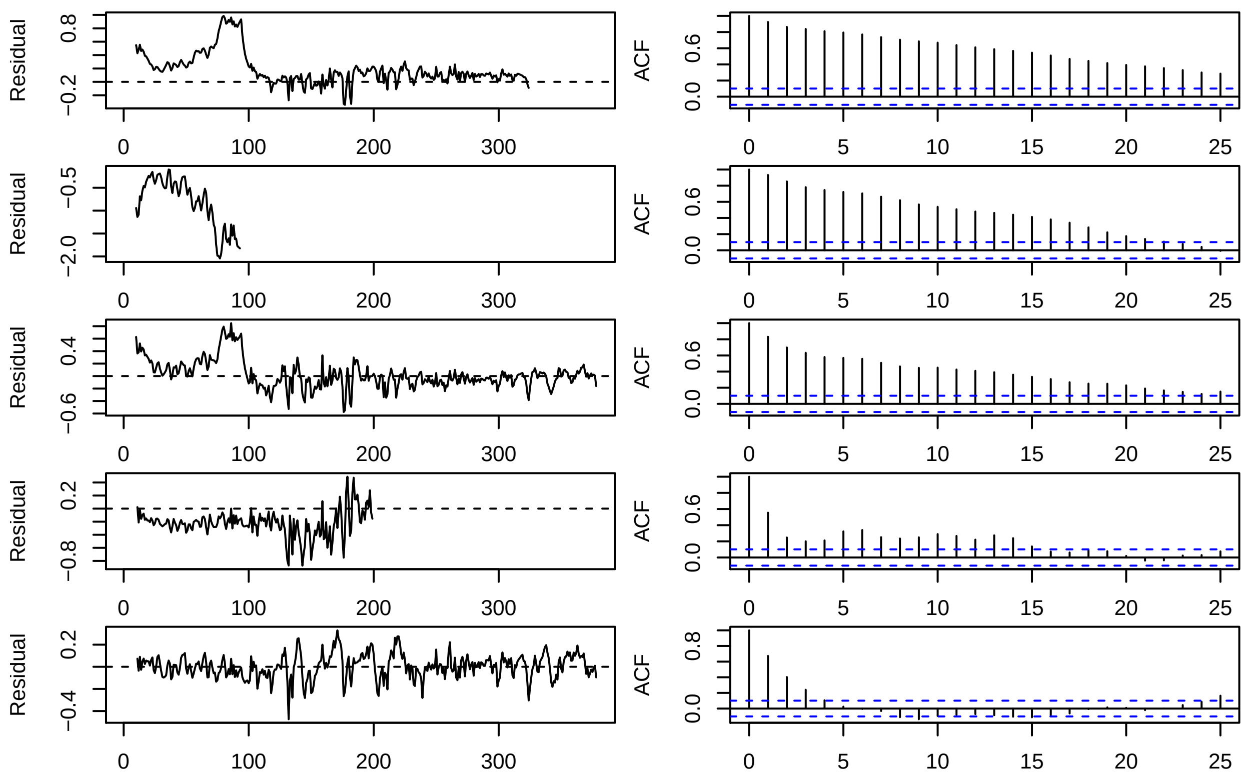
```
##
## MARSS fit is
## Estimation method: kem
## Convergence test: conv.test.slope.tol = 0.5, abstol = 0.001
## Estimation converged in 3251 iterations.
## Log-likelihood: 311.9935
## AIC: -445.987   AICc: -444.0914
##
##              ML.Est  Std.Err    low.CI    up.CI
## R.diag     4.66e-02 0.000765  4.51e-02 0.048146
## Q.(1,1)    1.08e-02 0.001547  7.79e-03 0.013853
## Q.(2,1)    9.58e-03 0.001222  7.18e-03 0.011974
## Q.(3,1)    1.01e-02 0.001347  7.46e-03 0.012739
## Q.(4,1)    9.63e-03 0.001243  7.19e-03 0.012062
## Q.(5,1)    9.21e-03 0.001261  6.74e-03 0.011681
## Q.(6,1)    1.11e-02 0.001649  7.82e-03 0.014285
## Q.(7,1)    6.48e-03 0.001058  4.41e-03 0.008558
## Q.(8,1)    1.68e-03 0.000590  5.27e-04 0.002839
## Q.(9,1)    4.73e-03 0.000690  3.38e-03 0.006086
## Q.(10,1)   3.34e-03 0.000646  2.07e-03 0.004607
## Q.(11,1)   1.05e-02 0.001379  7.75e-03 0.013159
## Q.(2,2)    1.04e-02 0.001281  7.88e-03 0.012899
## Q.(3,2)    1.02e-02 0.001268  7.70e-03 0.012676
## Q.(4,2)    1.02e-02 0.001240  7.81e-03 0.012674
## Q.(5,2)    9.73e-03 0.001239  7.30e-03 0.012155
## Q.(6,2)    1.28e-02 0.001720  9.45e-03 0.016192
## Q.(7,2)    6.86e-03 0.001020  4.86e-03 0.008859
## Q.(8,2)    1.64e-03 0.000548  5.62e-04 0.002711
## Q.(9,2)    5.15e-03 0.000678  3.82e-03 0.006484
## Q.(10,2)   3.57e-03 0.000618  2.36e-03 0.004778
## Q.(11,2)   1.08e-02 0.001313  8.19e-03 0.013339
## Q.(3,3)    1.46e-02 0.001731  1.12e-02 0.018015
## Q.(4,3)    1.04e-02 0.001335  7.82e-03 0.013057
## Q.(5,3)    9.66e-03 0.001318  7.07e-03 0.012241
## Q.(6,3)    1.02e-02 0.001652  7.00e-03 0.013474
## Q.(7,3)    7.21e-03 0.001153  4.95e-03 0.009474
## Q.(8,3)    2.68e-03 0.000683  1.35e-03 0.004022
## Q.(9,3)    5.84e-03 0.000791  4.29e-03 0.007395
## Q.(10,3)   4.80e-03 0.000766  3.30e-03 0.006303
## Q.(11,3)   1.50e-02 0.001688  1.17e-02 0.018303
## Q.(4,4)    1.02e-02 0.001342  7.54e-03 0.012802
```

```
## Q.(5,4)     9.69e-03 0.001254  7.23e-03 0.012142
## Q.(6,4)     1.23e-02 0.001680  9.05e-03 0.015640
## Q.(7,4)     6.73e-03 0.001017  4.73e-03 0.008720
## Q.(8,4)     1.78e-03 0.000551  7.05e-04 0.002863
## Q.(9,4)     5.13e-03 0.000681  3.80e-03 0.006468
## Q.(10,4)    3.61e-03 0.000620  2.40e-03 0.004828
## Q.(11,4)    1.10e-02 0.001377  8.28e-03 0.013677
## Q.(5,5)     9.63e-03 0.001415  6.85e-03 0.012399
## Q.(6,5)     1.16e-02 0.001665  8.36e-03 0.014887
## Q.(7,5)     6.19e-03 0.000996  4.24e-03 0.008145
## Q.(8,5)     1.61e-03 0.000547  5.41e-04 0.002686
## Q.(9,5)     4.65e-03 0.000648  3.38e-03 0.005923
## Q.(10,5)    3.44e-03 0.000627  2.21e-03 0.004671
## Q.(11,5)    1.03e-02 0.001396  7.55e-03 0.013027
## Q.(6,6)     1.79e-02 0.002708  1.26e-02 0.023167
## Q.(7,6)     7.88e-03 0.001378  5.18e-03 0.010581
## Q.(8,6)     1.23e-03 0.000759 -2.58e-04 0.002719
## Q.(9,6)     5.99e-03 0.000938  4.15e-03 0.007829
## Q.(10,6)    3.37e-03 0.000818  1.76e-03 0.004972
## Q.(11,6)    1.11e-02 0.001744  7.73e-03 0.014565
## Q.(7,7)     6.65e-03 0.001153  4.39e-03 0.008907
## Q.(8,7)     1.65e-03 0.000485  7.03e-04 0.002604
## Q.(9,7)     3.78e-03 0.000595  2.61e-03 0.004944
## Q.(10,7)    2.94e-03 0.000557  1.85e-03 0.004035
## Q.(11,7)    7.95e-03 0.001232  5.54e-03 0.010368
## Q.(8,8)     1.91e-03 0.000383  1.16e-03 0.002664
## Q.(9,8)     1.83e-03 0.000367  1.12e-03 0.002553
## Q.(10,8)    1.82e-03 0.000360  1.12e-03 0.002529
## Q.(11,8)    2.80e-03 0.000708  1.42e-03 0.004189
## Q.(9,9)     3.36e-03 0.000483  2.41e-03 0.004304
## Q.(10,9)    2.70e-03 0.000416  1.88e-03 0.003512
## Q.(11,9)    6.07e-03 0.000812  4.47e-03 0.007658
## Q.(10,10)   2.74e-03 0.000457  1.85e-03 0.003640
## Q.(11,10)   4.92e-03 0.000785  3.39e-03 0.006461
## Q.(11,11)   1.57e-02 0.001843  1.21e-02 0.019280
## x0.X1       8.12e-01 0.321857  1.81e-01 1.442374
## x0.X2       1.01e+00 0.311113  3.97e-01 1.617014
## x0.X3       1.03e+00 0.354114  3.32e-01 1.720270
## x0.X4       1.03e+00 0.308485  4.28e-01 1.637267
## x0.X5       1.36e+00 0.312260  7.45e-01 1.969097
## x0.X6       1.25e+00 0.438070  3.89e-01 2.106327
## x0.X7       8.37e-01 0.258043  3.31e-01 1.342717
## x0.X8       7.08e-01 0.143317  4.27e-01 0.988510
## x0.X9       1.02e+00 0.170085  6.88e-01 1.355067
## x0.X10      9.43e-01 0.162580  6.25e-01 1.261890
## x0.X11      1.08e+00 0.362198  3.68e-01 1.787971
## C.X1        1.99e-04 0.000386 -5.58e-04 0.000955
## C.X2        1.35e-04 0.000377 -6.03e-04 0.000873
## C.X3        2.92e-04 0.000446 -5.82e-04 0.001165
## C.X4        1.45e-04 0.000373 -5.86e-04 0.000876
## C.X5       -1.64e-04 0.000365 -8.78e-04 0.000551
## C.X6        6.89e-05 0.000503 -9.16e-04 0.001054
## C.X7        2.24e-04 0.000305 -3.73e-04 0.000821
## C.X8        5.23e-04 0.000166  1.98e-04 0.000847
```

```
## C.X9       3.51e-04 0.000215 -7.00e-05 0.000771
## C.X10      3.40e-04 0.000196 -4.36e-05 0.000724
## C.X11      2.73e-04 0.000461 -6.30e-04 0.001177
## Initial states (x0) defined at t=0
##
## CIs calculated at alpha = 0.05 via method=hessian
```

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

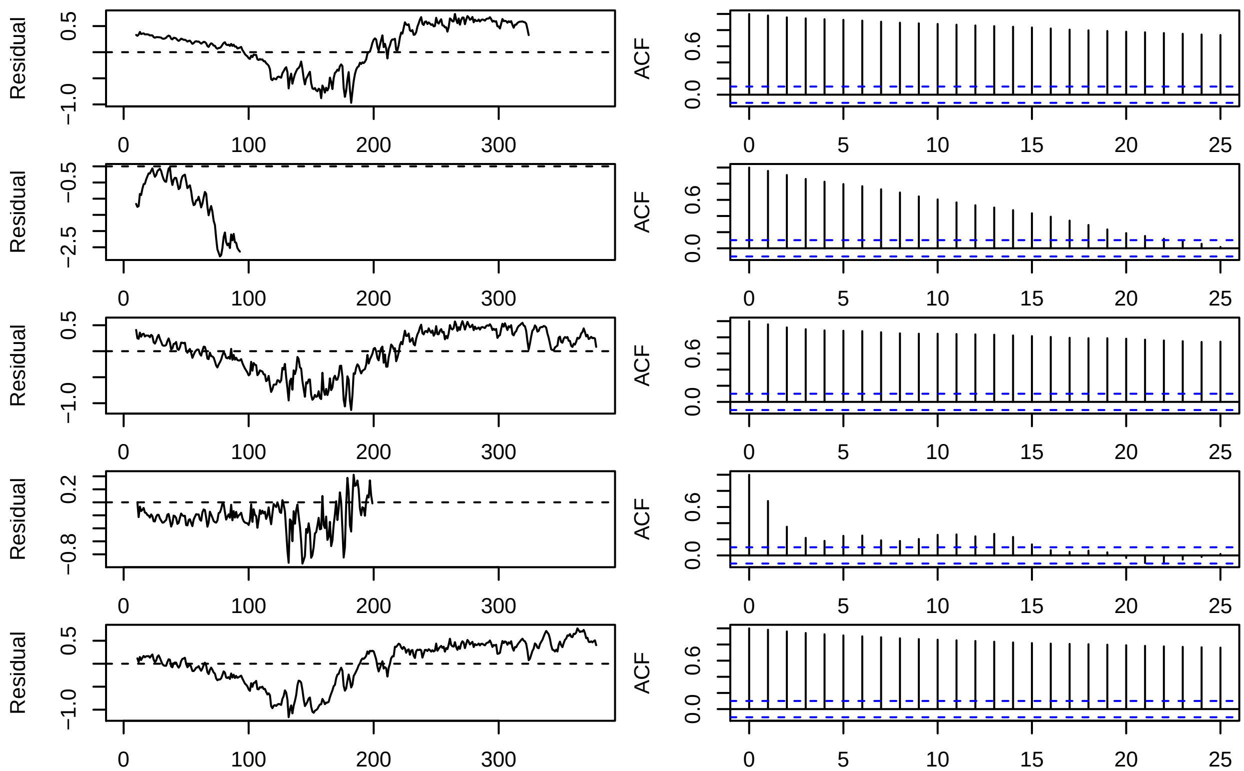
```
## MARSSresiduals.tt1 reported warnings. See msg element of returned residuals object.

## MARSSresiduals.tt1 reported warnings. See msg element of returned residuals object.

## MARSSresiduals.tt1 reported warnings. See msg element of returned residuals object.

## MARSSresiduals.tt1 reported warnings. See msg element of returned residuals object.

## MARSSresiduals.tt1 reported warnings. See msg element of returned residuals object.
```



```r
#Model 2, hypothesis 2 (creeks vs ponds)
mod2.params
```

```
##
## MARSS fit is
## Estimation method: kem
## Convergence test: conv.test.slope.tol = 0.5, abstol = 0.001
## Estimation converged in 91 iterations.
## Log-likelihood: -4891.9
## AIC: 9799.801   AICc: 9799.817
##
##           ML.Est  Std.Err    low.CI    up.CI
## R.diag  0.176657 0.002753  0.171261 0.182053
## Q.(1,1) 0.004101 0.000690  0.002748 0.005453
## Q.(2,1) 0.005458 0.000909  0.003676 0.007240
## Q.(2,2) 0.008550 0.001468  0.005672 0.011428
## x0.X1   0.933060 0.198706  0.543604 1.322515
## x0.X2   1.055530 0.293369  0.480537 1.630523
## C.X1    0.000300 0.000239 -0.000168 0.000768
## C.X2    0.000325 0.000346 -0.000353 0.001003
## Initial states (x0) defined at t=0
##
## CIs calculated at alpha = 0.05 via method=hessian
```

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




```r
#Model 3, hypothesis 3 (trib vs. trib)
mod3.params
```

```
##
## MARSS fit is
## Estimation method: kem
## Convergence test: conv.test.slope.tol = 0.5, abstol = 0.001
## Estimation converged in 2518 iterations.
## Log-likelihood: -1836.822
## AIC: 3711.645   AICc: 3711.734
##
##           ML.Est  Std.Err    low.CI    up.CI
## R.diag  0.082798 0.001311  8.02e-02 0.085368
## Q.(1,1) 0.010326 0.001421  7.54e-03 0.013112
## Q.(2,1) 0.009662 0.001319  7.08e-03 0.012248
## Q.(3,1) 0.003657 0.000625  2.43e-03 0.004882
## Q.(4,1) 0.009611 0.001286  7.09e-03 0.012132
## Q.(2,2) 0.012367 0.001709  9.02e-03 0.015717
## Q.(3,2) 0.004285 0.000690  2.93e-03 0.005637
## Q.(4,2) 0.011955 0.001590  8.84e-03 0.015070
## Q.(3,3) 0.002523 0.000420  1.70e-03 0.003347
## Q.(4,3) 0.004213 0.000667  2.91e-03 0.005520
## Q.(4,4) 0.011584 0.001559  8.53e-03 0.014640
## x0.X1   0.992099 0.320593  3.64e-01 1.620449
## x0.X2   1.076092 0.339679  4.10e-01 1.741850
```

```
## x0.X3   0.918146 0.153676  6.17e-01 1.219345
## x0.X4   1.083917 0.326099  4.45e-01 1.723059
## C.X1    0.000116 0.000377 -6.23e-04 0.000854
## C.X2    0.000300 0.000412 -5.08e-04 0.001108
## C.X3    0.000373 0.000187  5.91e-06 0.000741
## C.X4    0.000293 0.000398 -4.88e-04 0.001074
## 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.fit, type = "tt1")$model.residuals[j, ],
            ylab = "Residual")
    abline(h = 0, lty = "dashed")
    acf(residuals,na.action = na.pass)
  }
```

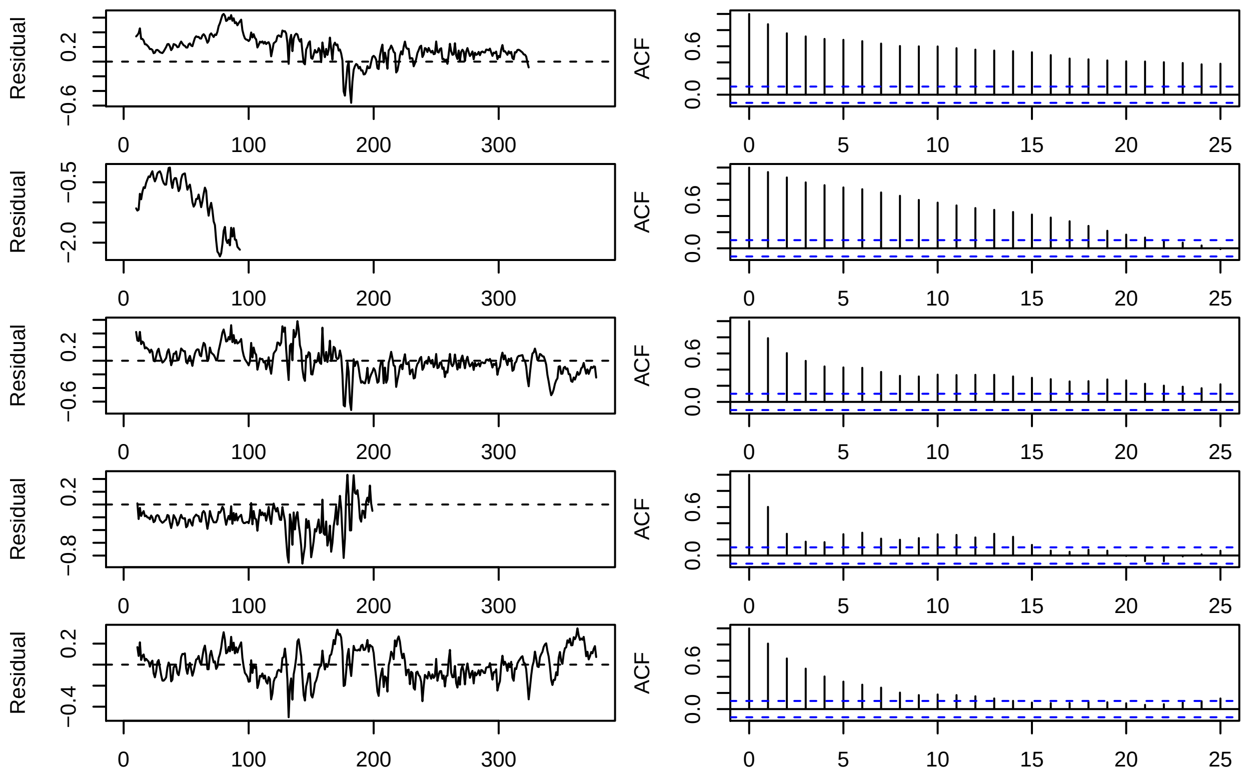
```
## MARSSresiduals.tt1 reported warnings. See msg element of returned residuals object.
```

```
## MARSSresiduals.tt1 reported warnings. See msg element of returned residuals object.
```

```
## MARSSresiduals.tt1 reported warnings. See msg element of returned residuals object.
```

```
## MARSSresiduals.tt1 reported warnings. See msg element of returned residuals object.
```

```
## MARSSresiduals.tt1 reported warnings. See msg element of returned residuals object.
```

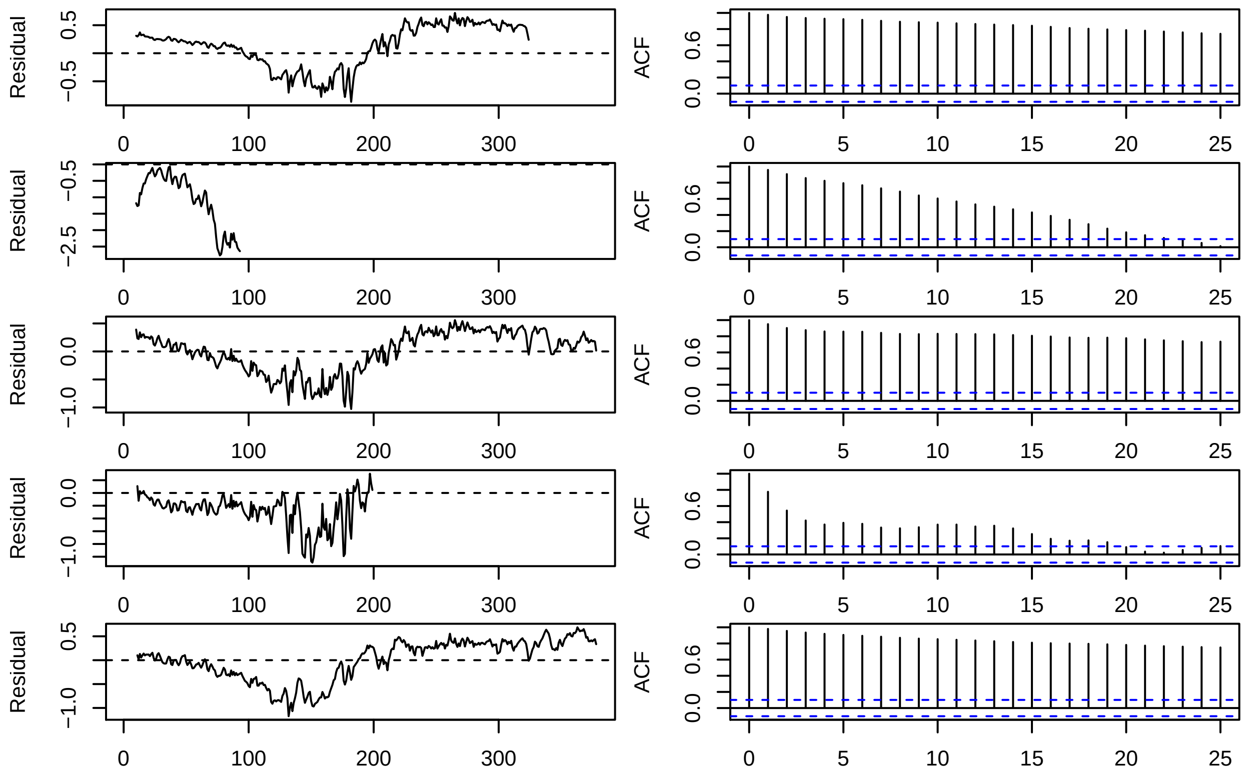


```r
#Model 4, hypothesis 4 (all same)
mod4.params
```

```
##
## MARSS fit is
## Estimation method: kem
## Convergence test: conv.test.slope.tol = 0.5, abstol = 0.001
## Estimation converged in 36 iterations.
## Log-likelihood: -5032.398
## AIC: 10072.8   AICc: 10072.8
##
##          ML.Est  Std.Err    low.CI   up.CI
## R.diag 0.184194 0.002858  0.178593 0.18980
## Q.Q    0.004608 0.000759  0.003121 0.00610
## x0.x0  0.936723 0.210525  0.524102 1.34934
## C.C    0.000314 0.000253 -0.000182 0.00081
## Initial states (x0) defined at t=0
##
## CIs calculated at alpha = 0.05 via method=hessian
```

```r
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.fit, type = "tt1")$model.residuals[j, ],
            ylab = "Residual")
    abline(h = 0, lty = "dashed")
```

```
    acf(residuals,na.action = na.pass)
  }
```



```
#...these models are not good
```

###Comparing Original AICc values

```
data.frame(Model=c("Model1", "Model2", "Model3", "Model4"),
           AICc=round(c(mod1.fit$AICc,
                        mod2.fit$AICc,
                        mod3.fit$AICc,
                        mod4.fit$AICc),1))
```

```
##    Model    AICc
## 1 Model1  -444.1
## 2 Model2  9799.8
## 3 Model3  3711.7
## 4 Model4 10072.8
```

###correct for seasonality using Fourier Series, and z-scoring airtemperature as an additional covariate ###Question: Did I do the period correctly?

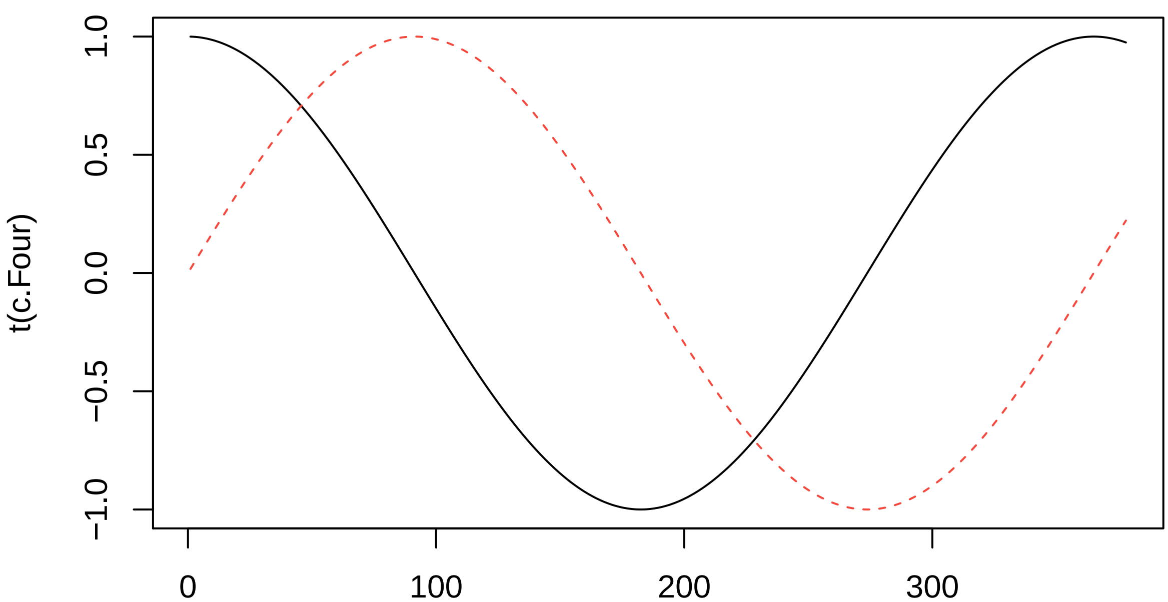
```r
#Correct for seasonality using Fourier Series
TT = ncol(transformed_dat) # number of time periods/samples
period = 365 # number of "seasons" (e.g., 12 months per year)
per.1st = 1 # first "season" (e.g., Jan = 1, July = 7)
c = diag(period) # create factors for seasons
for(i in 2:(ceiling(TT/period)))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

```
## [1] 365 730
```

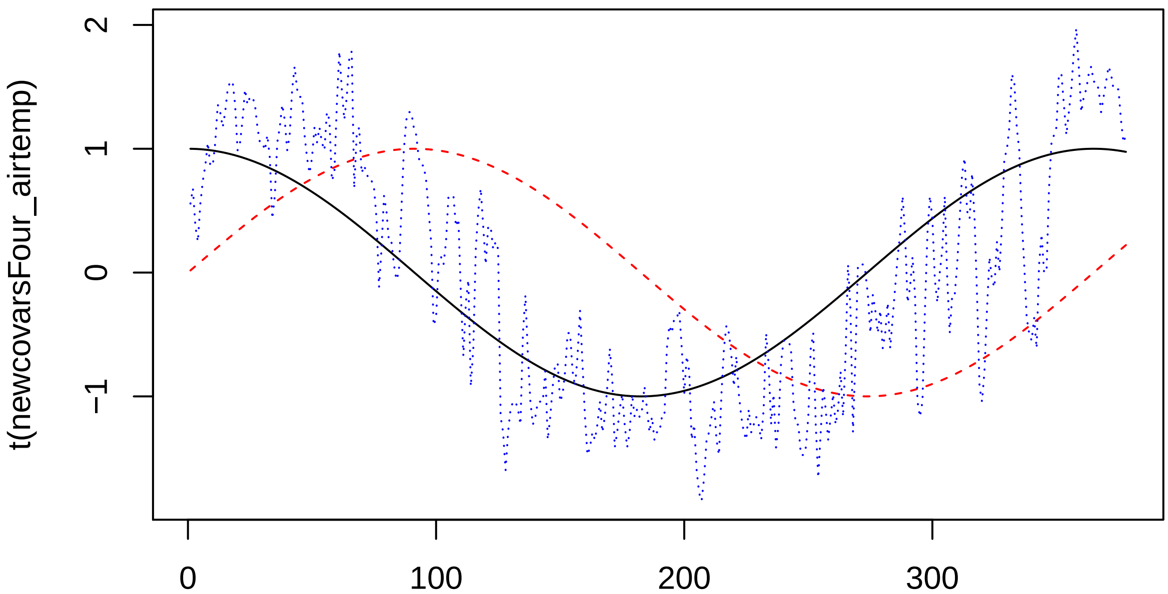
```r
#Create Fourier Series
cos.t = cos(2 * pi * seq(TT) / period)
sin.t = sin(2 * pi * seq(TT) / period)
c.Four = rbind(cos.t,sin.t)
cor(c.Four[1,],c.Four[2,]) # not correlated!
```

```
## [1] 0.007872561
```

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




```r
#Now fit model with seasonality AND an additional covariate (airtemp from above)
airtemp_z <- zscore(airtemp$TAVG)
newcovarsFour_airtemp <-rbind(c.Four, "airtemp"=airtemp_z)
matplot(t(newcovarsFour_airtemp), type="l", col=c("black","red","blue"))
```



###Checking model results and residuals when log transformed

```r
mod5.fit <- readRDS("mod5.fit.rds")
mod5.params <- readRDS("mod5.params.rds")
mod6.fit <- readRDS("mod6.fit.rds")
mod6.params <- readRDS("mod6.params.rds")
mod7.fit <- readRDS("mod7.fit.rds")
mod7.params <- readRDS("mod7.params.rds")
mod8.fit <- readRDS("mod8.fit.rds")
mod8.params <- readRDS("mod8.params.rds")

#Model 5, hypothesis 1 (all separate)
mod5.params
```

```
##
## MARSS fit is
## Estimation method: kem
## Convergence test: conv.test.slope.tol = 0.5, abstol = 0.001
## Estimation converged in 2833 iterations.
## Log-likelihood: 386.5636
## AIC: -551.1273   AICc: -548.1774
##
##                    ML.Est  Std.Err    low.CI     up.CI
## R.diag           0.046473 0.000759  4.50e-02  0.047961
## Q.(1,1)          0.010158 0.001506  7.21e-03  0.013110
## Q.(2,1)          0.009165 0.001216  6.78e-03  0.011549
```

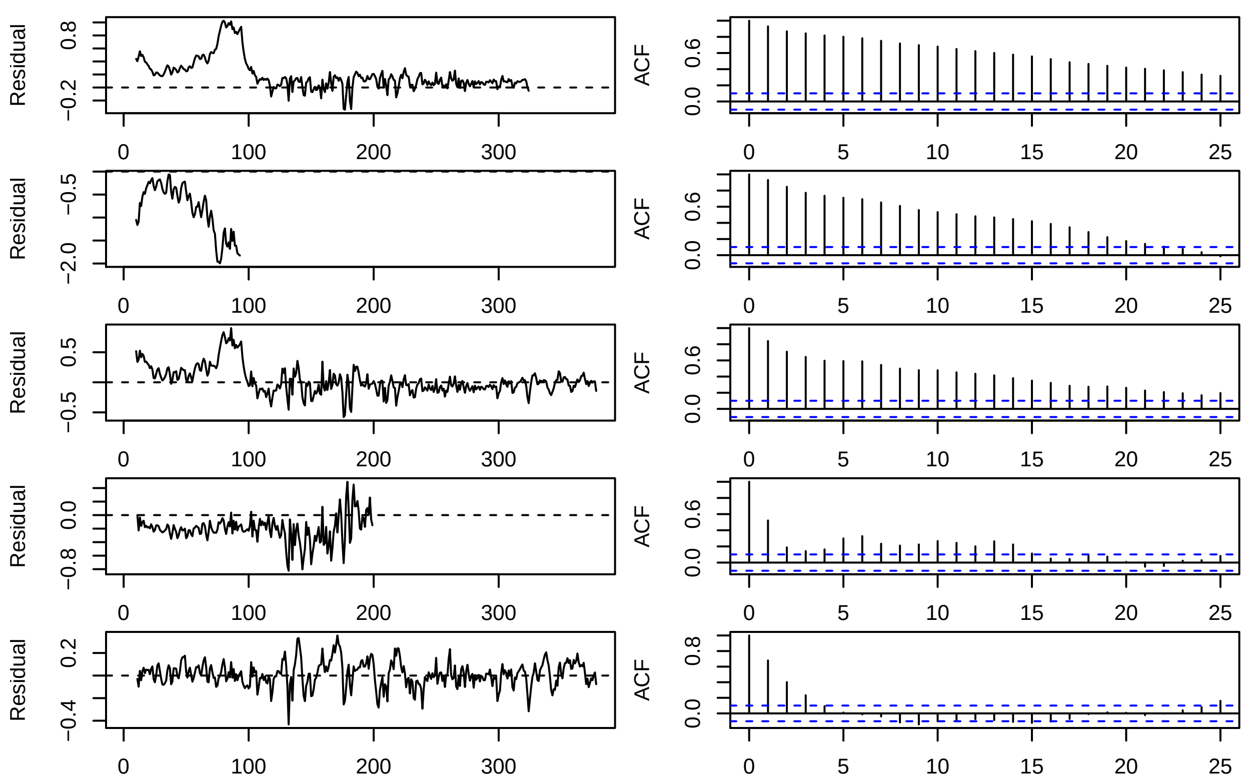
```
## Q.(3,1)          0.009139 0.001270  6.65e-03  0.011629
## Q.(4,1)          0.008580 0.001184  6.26e-03  0.010900
## Q.(5,1)          0.008557 0.001234  6.14e-03  0.010974
## Q.(6,1)          0.011278 0.001651  8.04e-03  0.014514
## Q.(7,1)          0.005811 0.001014  3.82e-03  0.007798
## Q.(8,1)          0.000830 0.000477 -1.05e-04  0.001766
## Q.(9,1)          0.004203 0.000609  3.01e-03  0.005397
## Q.(10,1)         0.002646 0.000550  1.57e-03  0.003725
## Q.(11,1)         0.010720 0.001434  7.91e-03  0.013530
## Q.(2,2)          0.010017 0.001353  7.37e-03  0.012669
## Q.(3,2)          0.009388 0.001231  6.97e-03  0.011801
## Q.(4,2)          0.009248 0.001201  6.89e-03  0.011602
## Q.(5,2)          0.009075 0.001228  6.67e-03  0.011482
## Q.(6,2)          0.012821 0.001729  9.43e-03  0.016210
## Q.(7,2)          0.006211 0.000999  4.25e-03  0.008170
## Q.(8,2)          0.000715 0.000454 -1.75e-04  0.001606
## Q.(9,2)          0.004508 0.000607  3.32e-03  0.005697
## Q.(10,2)         0.002712 0.000533  1.67e-03  0.003757
## Q.(11,2)         0.011067 0.001394  8.34e-03  0.013799
## Q.(3,3)          0.013393 0.001641  1.02e-02  0.016610
## Q.(4,3)          0.009152 0.001256  6.69e-03  0.011614
## Q.(5,3)          0.008776 0.001280  6.27e-03  0.011285
## Q.(6,3)          0.010262 0.001630  7.07e-03  0.013456
## Q.(7,3)          0.006238 0.001091  4.10e-03  0.008375
## Q.(8,3)          0.001496 0.000552  4.14e-04  0.002578
## Q.(9,3)          0.004957 0.000687  3.61e-03  0.006304
## Q.(10,3)         0.003860 0.000651  2.58e-03  0.005135
## Q.(11,3)         0.015018 0.001701  1.17e-02  0.018353
## Q.(4,4)          0.008663 0.001352  6.01e-03  0.011312
## Q.(5,4)          0.008441 0.001184  6.12e-03  0.010761
## Q.(6,4)          0.011778 0.001645  8.55e-03  0.015002
## Q.(7,4)          0.005720 0.000954  3.85e-03  0.007589
## Q.(8,4)          0.000853 0.000435  2.70e-07  0.001706
## Q.(9,4)          0.004265 0.000599  3.09e-03  0.005439
## Q.(10,4)         0.002588 0.000515  1.58e-03  0.003598
## Q.(11,4)         0.010750 0.001427  7.95e-03  0.013546
## Q.(5,5)          0.008782 0.001465  5.91e-03  0.011652
## Q.(6,5)          0.011434 0.001663  8.17e-03  0.014694
## Q.(7,5)          0.005485 0.000970  3.58e-03  0.007386
## Q.(8,5)          0.000821 0.000443 -4.68e-05  0.001689
## Q.(9,5)          0.004152 0.000608  2.96e-03  0.005344
## Q.(10,5)         0.002672 0.000543  1.61e-03  0.003736
## Q.(11,5)         0.010456 0.001475  7.56e-03  0.013348
## Q.(6,6)          0.017758 0.002693  1.25e-02  0.023036
## Q.(7,6)          0.007440 0.001336  4.82e-03  0.010058
## Q.(8,6)          0.000452 0.000606 -7.34e-04  0.001639
## Q.(9,6)          0.005568 0.000825  3.95e-03  0.007185
## Q.(10,6)         0.002661 0.000688  1.31e-03  0.004009
## Q.(11,6)         0.012286 0.001848  8.66e-03  0.015908
## Q.(7,7)          0.005763 0.001094  3.62e-03  0.007908
## Q.(8,7)          0.000596 0.000373 -1.35e-04  0.001328
## Q.(9,7)          0.002840 0.000494  1.87e-03  0.003807
## Q.(10,7)         0.001981 0.000454  1.09e-03  0.002870
## Q.(11,7)         0.007796 0.001271  5.31e-03  0.010287
```

```
## Q.(8,8)          0.000707 0.000217  2.82e-04  0.001133
## Q.(9,8)          0.000548 0.000227  1.03e-04  0.000992
## Q.(10,8)         0.000664 0.000214  2.45e-04  0.001083
## Q.(11,8)         0.001822 0.000631  5.85e-04  0.003059
## Q.(9,9)          0.002243 0.000386  1.49e-03  0.003000
## Q.(10,9)         0.001554 0.000288  9.90e-04  0.002119
## Q.(11,9)         0.005728 0.000773  4.21e-03  0.007243
## Q.(10,10)        0.001559 0.000342  8.89e-04  0.002229
## Q.(11,10)        0.004413 0.000730  2.98e-03  0.005845
## Q.(11,11)        0.017198 0.002019  1.32e-02  0.021156
## x0.X1            1.004073 0.318263  3.80e-01  1.627856
## x0.X2            1.127373 0.312950  5.14e-01  1.740745
## x0.X3            1.171648 0.345779  4.94e-01  1.849363
## x0.X4            1.239960 0.289868  6.72e-01  1.808091
## x0.X5            1.380072 0.301517  7.89e-01  1.971035
## x0.X6            1.342542 0.438404  4.83e-01  2.201798
## x0.X7            0.833946 0.249183  3.46e-01  1.322335
## x0.X8            0.589145 0.098786  3.96e-01  0.782761
## x0.X9            0.870386 0.143302  5.90e-01  1.151253
## x0.X10           0.840911 0.128746  5.89e-01  1.093248
## x0.X11           1.175008 0.386189  4.18e-01  1.931925
## C.(X1,cos.t)    -0.024999 0.016304 -5.70e-02  0.006956
## C.(X2,cos.t)    -0.004857 0.015634 -3.55e-02  0.025785
## C.(X3,cos.t)    -0.017093 0.017866 -5.21e-02  0.017923
## C.(X4,cos.t)    -0.007122 0.014761 -3.61e-02  0.021809
## C.(X5,cos.t)    -0.005733 0.015306 -3.57e-02  0.024266
## C.(X6,cos.t)     0.016513 0.021392 -2.54e-02  0.058440
## C.(X7,cos.t)     0.000410 0.013149 -2.54e-02  0.026181
## C.(X8,cos.t)     0.014757 0.005707  3.57e-03  0.025944
## C.(X9,cos.t)     0.014909 0.007670 -1.25e-04  0.029942
## C.(X10,cos.t)    0.017251 0.007092  3.35e-03  0.031152
## C.(X11,cos.t)   -0.008135 0.020074 -4.75e-02  0.031208
## C.(X1,sin.t)    -0.033801 0.009749 -5.29e-02 -0.014694
## C.(X2,sin.t)    -0.027244 0.009508 -4.59e-02 -0.008608
## C.(X3,sin.t)    -0.032358 0.010931 -5.38e-02 -0.010933
## C.(X4,sin.t)    -0.028507 0.008907 -4.60e-02 -0.011050
## C.(X5,sin.t)    -0.026807 0.009112 -4.47e-02 -0.008948
## C.(X6,sin.t)    -0.016866 0.012844 -4.20e-02  0.008307
## C.(X7,sin.t)    -0.021720 0.007615 -3.66e-02 -0.006794
## C.(X8,sin.t)    -0.012635 0.003042 -1.86e-02 -0.006673
## C.(X9,sin.t)    -0.016143 0.004585 -2.51e-02 -0.007155
## C.(X10,sin.t)   -0.015757 0.004046 -2.37e-02 -0.007826
## C.(X11,sin.t)   -0.028513 0.012338 -5.27e-02 -0.004331
## C.(X1,airtemp)   0.027508 0.012689  2.64e-03  0.052378
## C.(X2,airtemp)   0.010778 0.012049 -1.28e-02  0.034393
## C.(X3,airtemp)   0.024818 0.013737 -2.11e-03  0.051741
## C.(X4,airtemp)   0.012634 0.011413 -9.74e-03  0.035004
## C.(X5,airtemp)   0.007170 0.011941 -1.62e-02  0.030575
## C.(X6,airtemp)  -0.010031 0.016617 -4.26e-02  0.022538
## C.(X7,airtemp)   0.010245 0.010376 -1.01e-02  0.030582
## C.(X8,airtemp)   0.006441 0.004683 -2.74e-03  0.015620
## C.(X9,airtemp)   0.003239 0.005980 -8.48e-03  0.014960
## C.(X10,airtemp)  0.000764 0.005654 -1.03e-02  0.011845
## C.(X11,airtemp)  0.016581 0.015404 -1.36e-02  0.046773
```

```
## 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(mod5.fit, type = "tt1")$model.residuals[j, ],
            ylab = "Residual")
    abline(h = 0, lty = "dashed")
    acf(residuals,na.action = na.pass)
  }
```

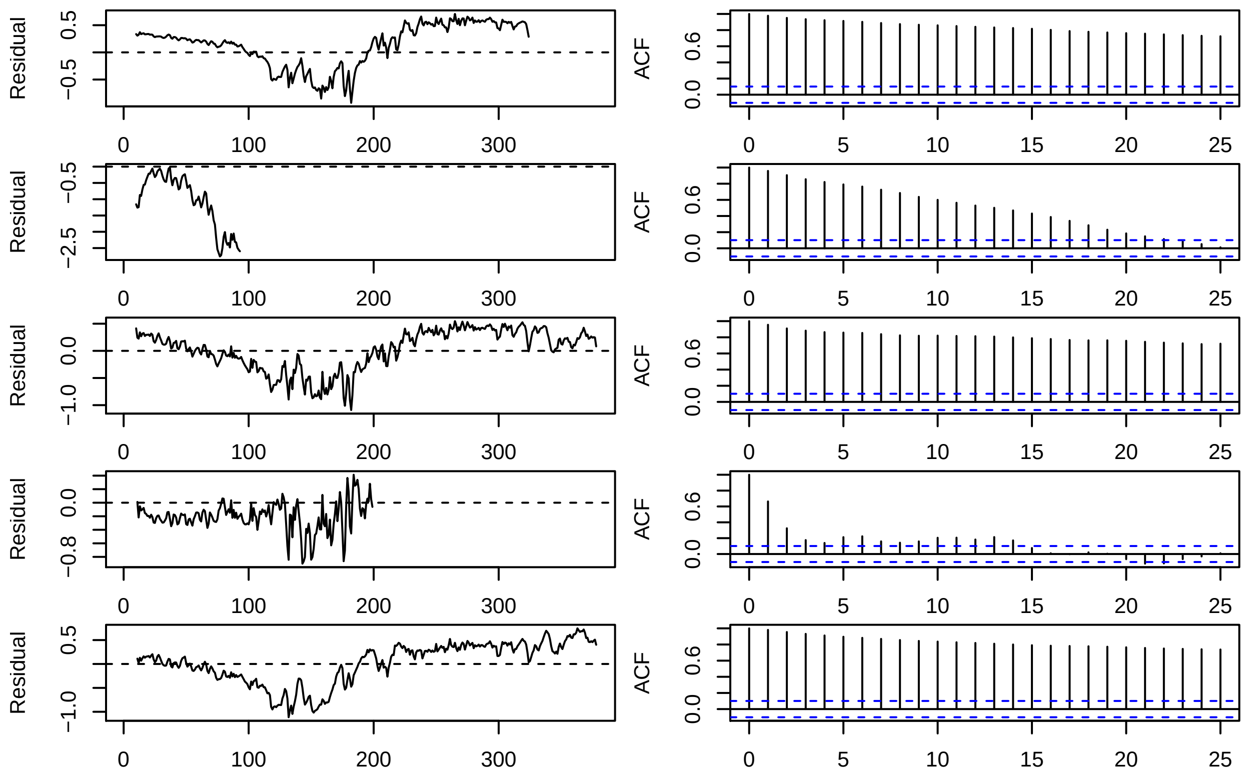
```
## MARSSresiduals.tt1 reported warnings. See msg element of returned residuals object.

## MARSSresiduals.tt1 reported warnings. See msg element of returned residuals object.

## MARSSresiduals.tt1 reported warnings. See msg element of returned residuals object.

## MARSSresiduals.tt1 reported warnings. See msg element of returned residuals object.

## MARSSresiduals.tt1 reported warnings. See msg element of returned residuals object.
```



```r
#Model 6, hypothesis 2 (creeks vs ponds)
mod6.params
```

```
##
## MARSS fit is
## Estimation method: kem
## Convergence test: conv.test.slope.tol = 0.5, abstol = 0.001
## Estimation converged in 196 iterations.
## Log-likelihood: -4871.922
## AIC: 9767.843   AICc: 9767.88
##
##                     ML.Est  Std.Err   low.CI    up.CI
## R.diag            1.77e-01 0.002751  0.17150  0.18228
## Q.(1,1)           2.97e-03 0.000550  0.00189  0.00405
## Q.(2,1)           4.67e-03 0.000798  0.00310  0.00623
## Q.(2,2)           7.84e-03 0.001417  0.00506  0.01061
## x0.X1             9.19e-01 0.175447  0.57535  1.26309
## x0.X2             1.14e+00 0.288131  0.57517  1.70463
## C.(X1,cos.t)     -3.69e-05 0.009149 -0.01797  0.01789
## C.(X2,cos.t)     -2.06e-02 0.014972 -0.04996  0.00873
## C.(X1,sin.t)     -2.25e-02 0.005374 -0.03301 -0.01194
## C.(X2,sin.t)     -3.41e-02 0.008765 -0.05130 -0.01695
## C.(X1,airtemp)    1.18e-02 0.007176 -0.00223  0.02590
## C.(X2,airtemp)    2.87e-02 0.011763  0.00567  0.05178
## Initial states (x0) defined at t=0
##
## CIs calculated at alpha = 0.05 via method=hessian
```

```r
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(mod6.fit, type = "tt1")$model.residuals[j, ],
            ylab = "Residual")
    abline(h = 0, lty = "dashed")
    acf(residuals,na.action = na.pass)
  }
```



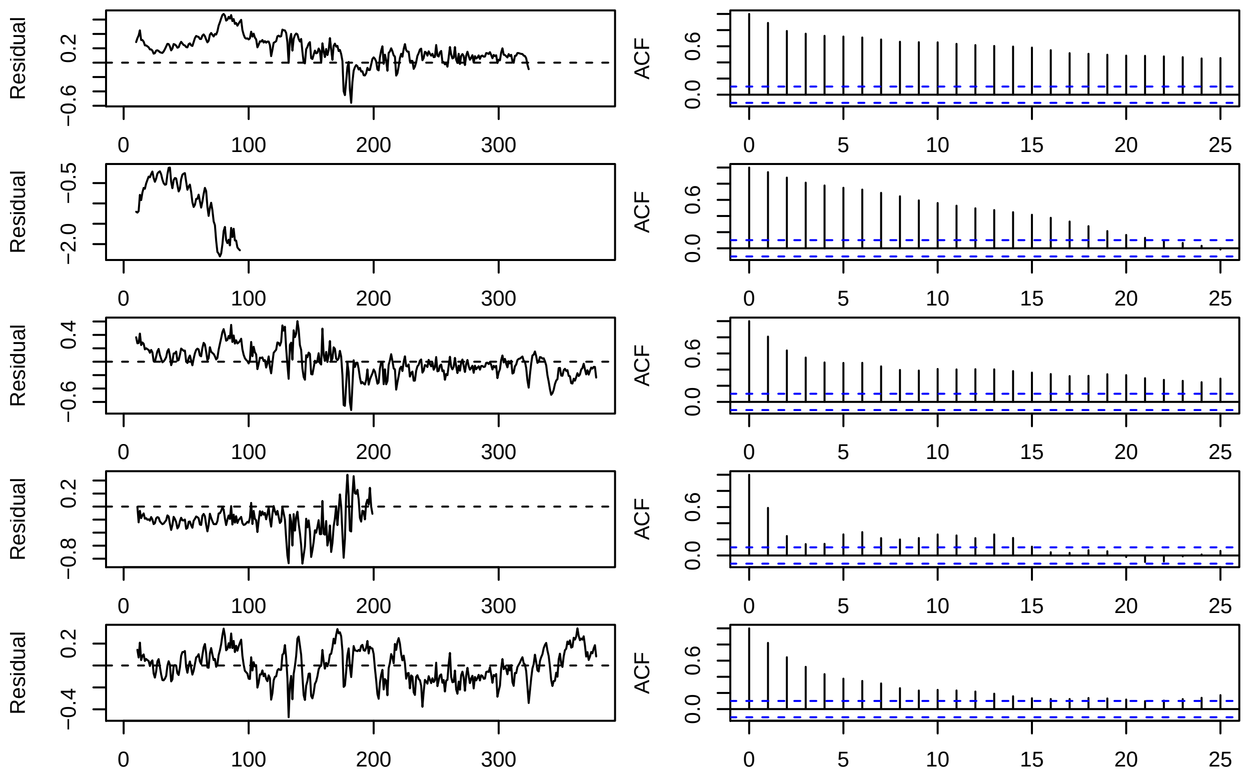

```r
#Model 7, hypothesis 3 (trib vs. trib)
mod7.params
```

```
##
## MARSS fit is
## Estimation method: kem
## Convergence test: conv.test.slope.tol = 0.5, abstol = 0.001
## Estimation converged in 3245 iterations.
## Log-likelihood: -1790.67
## AIC: 3635.341   AICc: 3635.519
##
##                  ML.Est  Std.Err   low.CI    up.CI
## R.diag          0.08268 0.001304  0.080121  0.08523
## Q.(1,1)         0.00989 0.001390  0.007160  0.01261
## Q.(2,1)         0.00883 0.001253  0.006374  0.01128
## Q.(3,1)         0.00281 0.000467  0.001896  0.00372
## Q.(4,1)         0.00986 0.001349  0.007212  0.01250
## Q.(2,2)         0.01087 0.001611  0.007714  0.01403
## Q.(3,2)         0.00306 0.000502  0.002082  0.00405
## Q.(4,2)         0.01160 0.001577  0.008509  0.01469
## Q.(3,3)         0.00117 0.000232  0.000712  0.00162
## Q.(4,3)         0.00345 0.000549  0.002378  0.00453
## Q.(4,4)         0.01251 0.001777  0.009031  0.01600
## x0.X1           1.05597 0.320861  0.427090  1.68484
## x0.X2           1.18240 0.327436  0.540634  1.82416
```

```
## x0.X3            0.78272 0.109468  0.568165  0.99727
## x0.X4            1.17979 0.345826  0.501981  1.85759
## C.(X1,cos.t)   -0.00858 0.015715 -0.039381  0.02222
## C.(X2,cos.t)   -0.01866 0.016721 -0.051431  0.01412
## C.(X3,cos.t)    0.01416 0.005922  0.002556  0.02577
## C.(X4,cos.t)   -0.01592 0.017692 -0.050597  0.01876
## C.(X1,sin.t)   -0.02797 0.009499 -0.046582 -0.00935
## C.(X2,sin.t)   -0.03301 0.010034 -0.052673 -0.01334
## C.(X3,sin.t)   -0.01549 0.003431 -0.022210 -0.00876
## C.(X4,sin.t)   -0.03196 0.010695 -0.052923 -0.01100
## C.(X1,airtemp)  0.01313 0.012137 -0.010659  0.03692
## C.(X2,airtemp)  0.02633 0.012964  0.000922  0.05174
## C.(X3,airtemp)  0.00376 0.004686 -0.005427  0.01294
## C.(X4,airtemp)  0.02383 0.013676 -0.002969  0.05064
## 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(mod7.fit, type = "tt1")$model.residuals[j, ],
            ylab = "Residual")
    abline(h = 0, lty = "dashed")
    acf(residuals,na.action = na.pass)
  }
```

```
## MARSSresiduals.tt1 reported warnings. See msg element of returned residuals object.

## MARSSresiduals.tt1 reported warnings. See msg element of returned residuals object.

## MARSSresiduals.tt1 reported warnings. See msg element of returned residuals object.

## MARSSresiduals.tt1 reported warnings. See msg element of returned residuals object.

## MARSSresiduals.tt1 reported warnings. See msg element of returned residuals object.
```

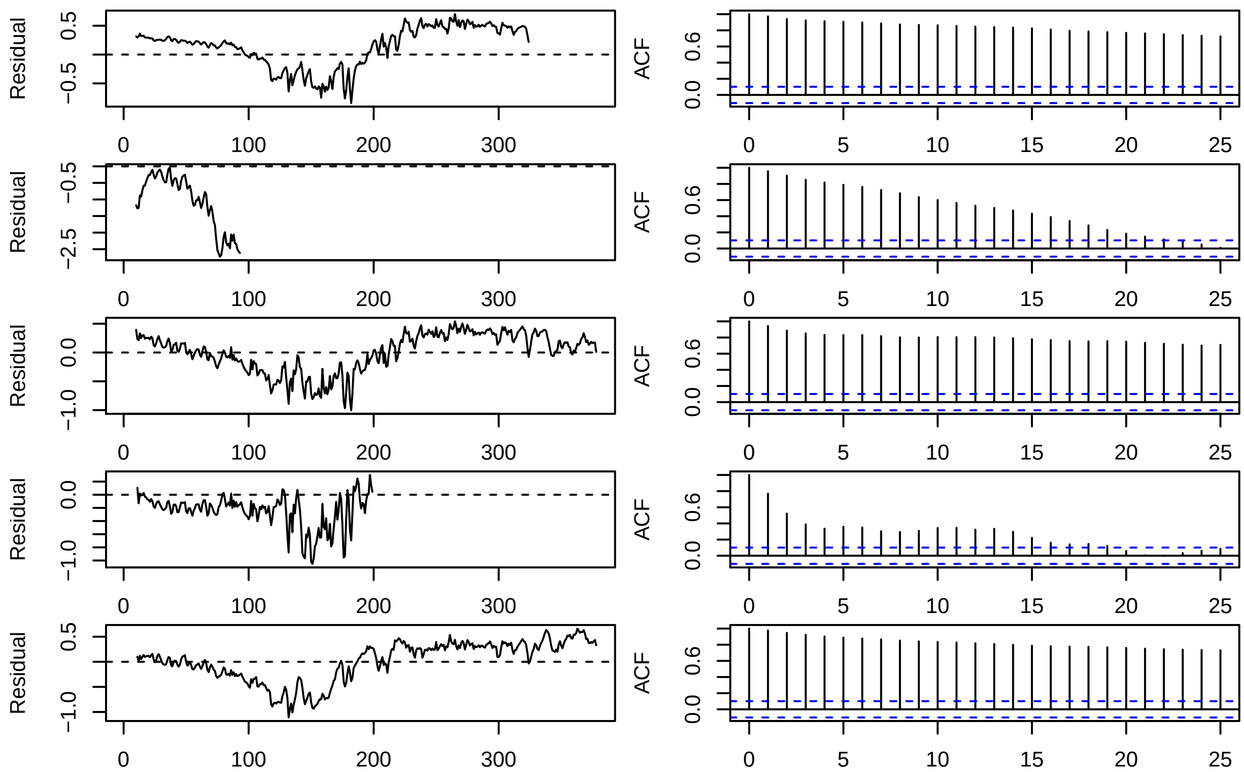



```
#Model 8, hypothesis 4 (all same)
mod8.params
```

```
##
## MARSS fit is
## Estimation method: kem
## Convergence test: conv.test.slope.tol = 0.5, abstol = 0.001
## Estimation converged in 39 iterations.
## Log-likelihood: -5020.639
## AIC: 10053.28   AICc: 10053.29
##
##                   ML.Est  Std.Err    low.CI    up.CI
## R.diag           0.18437 0.002857  0.178767  0.18997
## Q.Q              0.00358 0.000631  0.002345  0.00482
## x0.x0            0.94813 0.193187  0.569495  1.32677
## C.(X1,cos.t)    -0.00513 0.010009 -0.024749  0.01449
## C.(X1,sin.t)    -0.02543 0.005889 -0.036968 -0.01389
## C.(X1,airtemp)   0.01621 0.007838  0.000849  0.03157
## 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(mod8.fit, type = "tt1")$model.residuals[j, ],
```

```
            ylab = "Residual")
    abline(h = 0, lty = "dashed")
    acf(residuals,na.action = na.pass)
  }
```




```
#...these models are not good
```

###Comparing corrected AICc values

```
data.frame(Model=c("Model5", "Model6", "Model7", "Model8"),
           AICc=round(c(mod5.fit$AICc,
                        mod6.fit$AICc,
                        mod7.fit$AICc,
                        mod8.fit$AICc),1))
```

```
##    Model    AICc
## 1 Model5  -548.2
## 2 Model6  9767.9
## 3 Model7  3635.5
## 4 Model8 10053.3
```