573_HW_5

Matthew Stoebe

2024-12-02

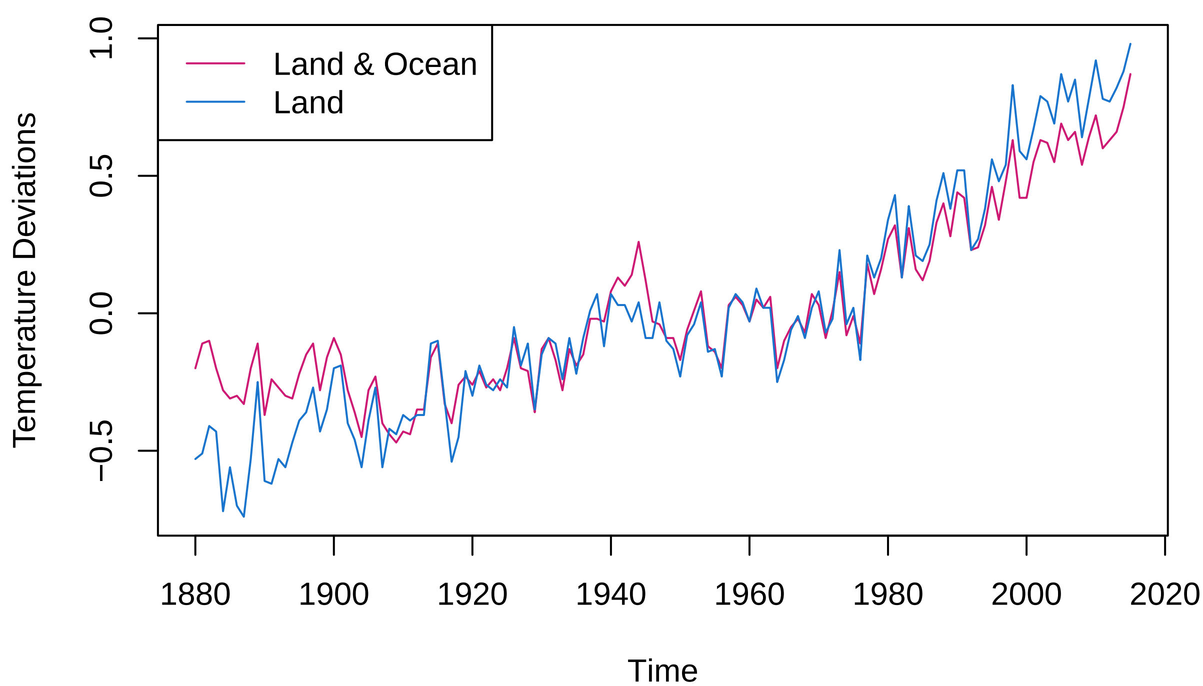
#Question 1

```r
library(MARSS)
```

```
## Warning: package 'MARSS' was built under R version 4.4.2
```

```r
library(astsa) # Contains the global temperature data

# Load the data
library(MARSS)
library(astsa) # this package contains the data
Y <- cbind(xglobtemp,xglobtempl)
ts.plot(Y, col=c(6,4), ylab='Temperature Deviations')
legend("topleft", legend = c("Land & Ocean", "Land"), col = c(6, 4), lty = 1)
```



```r
time <- time(Y)
```

##a

```r
# Define the model list
model.list <- list(
  Z = matrix(1, nrow = 2, ncol = 1),
  B = matrix(1),
  U = matrix("delta"),
  Q = matrix("q"),
  R = matrix(list("r11", "r12", "r12", "r22"), nrow = 2),
  x0 = matrix(-0.35),
  V0 = matrix(1)
)

# Fit the model
fit <- MARSS(Y, model = model.list, method = "BFGS")
```

```
## Success! Converged in 22 iterations.
## Function MARSSkfas used for likelihood calculation.
##
## MARSS fit is
## Estimation method: BFGS
## Estimation converged in 22 iterations.
## Log-likelihood: 194.7263
## AIC: -377.4527   AICc: -377.1357
##
##              Estimate
## A.xglobtempl -0.02177
## R.r11         0.00548
## R.r12         0.00947
## R.r22         0.03213
## U.delta       0.00649
## Q.q           0.00303
## Initial states (x0) defined at t=0
##
## Standard errors have not been calculated.
## Use MARSSparamCIs to compute CIs and bias estimates.
```

```r
logLik(fit)
```

```
## 'log Lik.' 194.7263 (df=6)
```

```r
fit$AICc
```

```
## [1] -377.1357
```

```r
residuals <- residuals(fit)
```

##b What are the unknown parameters? What are your estimates of the unknown parameters? Write down the fitted model. (The estimates of the variance parameters do not match the book, because the book returns estimates of the cholesky decomposition chol(R) instead of the variance-covariance matrix R.)

```r
coef_estimates <- coef(fit, type = "vector")
print(coef_estimates)
```

```
## A.xglobtempl        R.r11        R.r12        R.r22      U.delta          Q.q
## -0.021771362  0.005481636  0.009469011  0.032130406  0.006494341  0.003026686
```

##c What are the standard errors of your estimates in (b). Hint: use MARSSparamCIs function.

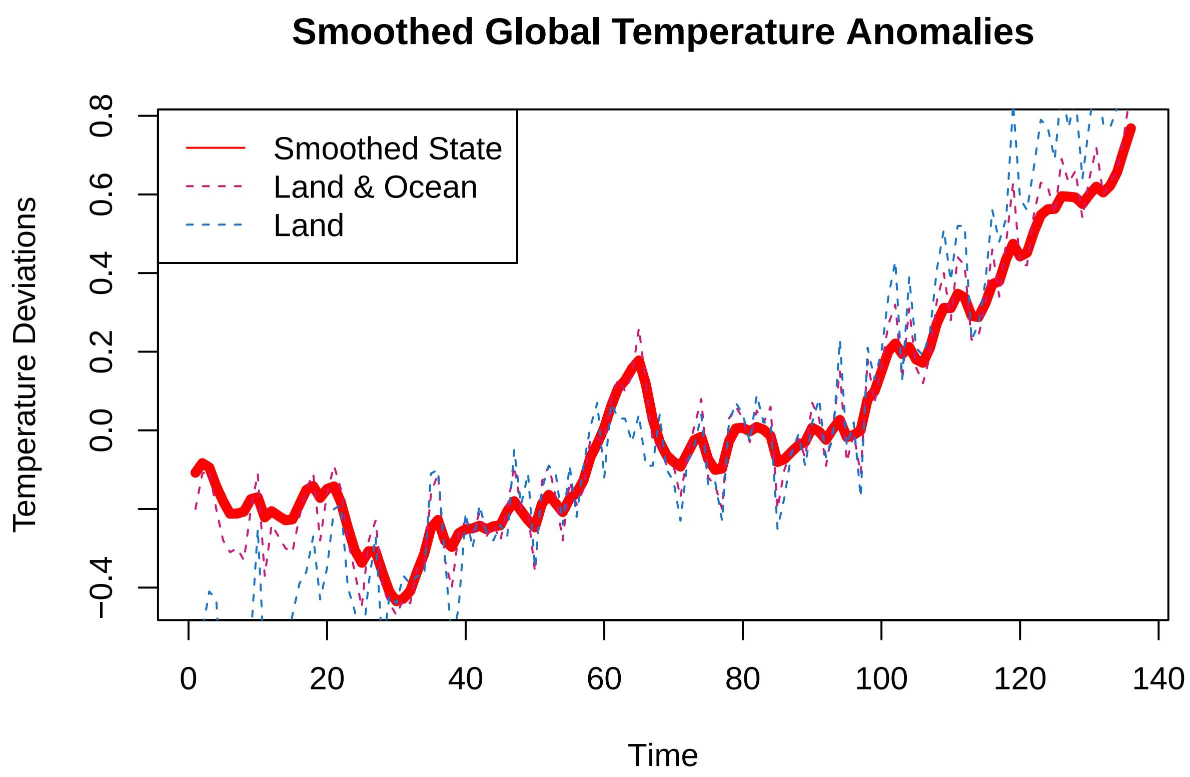
```r
param_CIs <- MARSSparamCIs(fit)
print(param_CIs$par.se)
```

```
## $Z
##      [,1]
##
## $A
##            [,1]
## [1,] 0.01171785
##
## $R
##             [,1]
## [1,] 0.001011265
## [2,] 0.001599049
## [3,] 0.004004567
##
## $B
##      [,1]
##
## $U
##            [,1]
## [1,] 0.00476605
##
## $Q
##              [,1]
## [1,] 0.0009514879
##
## $x0
##      [,1]
##
## $V0
##      [,1]
##
## $G
##      [,1]
##
## $H
##      [,1]
##
## $L
##      [,1]
```

##d

```
states <- fit$states

time_sequence <- seq_along(Y[, 1])
plot(time_sequence, states[1, ], type = "l", col = "red", ylab = "Temperature Deviations",
     xlab = "Time", main = "Smoothed Global Temperature Anomalies", lwd = 5)
lines(time_sequence, Y[, 1], col = 6, lty = 2)
lines(time_sequence, Y[, 2], col = 4, lty = 2)
legend("topleft", legend = c("Smoothed State", "Land & Ocean", "Land"),
       col = c("red", 6, 4), lty = c(1, 2, 2))
```



##e

Unconstrained R Assumptions - Errors vt1 and vt2 can have different variances. - There may be nonzero covariance between the errors

Diagonal and Unequal R Assumptions - Errors Vt1 and vt2 can have different variances - There is no covariance between the errors => Observation errors are uncorrelated

Diagonal and Equal R: - Errors are uncorrelated and have the same variance

##f Fit the two models in (e). Compare the AIC values of the three different models for R, which one is better?

```
fit_unconstrained <- fit

model_diag_unequal <- model.list
model_diag_unequal$R <- matrix(list("r11", 0, 0, "r22"), 2, 2)
fit_diag_unequal <- MARSS(Y, model = model_diag_unequal, method = "BFGS")
```

```
## Success! Converged in 33 iterations.
## Function MARSSkfas used for likelihood calculation.
##
## MARSS fit is
## Estimation method: BFGS
## Estimation converged in 33 iterations.
## Log-likelihood: 187.8447
## AIC: -365.6894   AICc: -365.4639
##
##              Estimate
## A.xglobtempl -0.02176
## R.r11         0.00364
## R.r22         0.01678
## U.delta       0.00809
## Q.q           0.00521
## Initial states (x0) defined at t=0
##
## Standard errors have not been calculated.
## Use MARSSparamCIs to compute CIs and bias estimates.
```

```r
model_diag_equal <- model.list
model_diag_equal$R <- matrix(list("r11", 0, 0, "r11"), 2, 2)
fit_diag_equal <- MARSS(Y, model = model_diag_equal, method = "BFGS")
```

```
## Success! Converged in 19 iterations.
## Function MARSSkfas used for likelihood calculation.
##
## MARSS fit is
## Estimation method: BFGS
## Estimation converged in 19 iterations.
## Log-likelihood: 181.4277
## AIC: -354.8554   AICc: -354.7055
##
##              Estimate
## A.xglobtempl -0.02176
## R.r11         0.01049
## U.delta       0.00897
## Q.q           0.00303
## Initial states (x0) defined at t=0
##
## Standard errors have not been calculated.
## Use MARSSparamCIs to compute CIs and bias estimates.
```

```r
AIC_values <- data.frame(
  Model = c("Unconstrained R", "Diagonal and Unequal R", "Diagonal and Equal R"),
  AICc = c(fit_unconstrained$AICc, fit_diag_unequal$AICc, fit_diag_equal$AICc)
)

print(AIC_values)
```

```
##                    Model      AICc
## 1        Unconstrained R -377.1357
## 2 Diagonal and Unequal R -365.4639
## 3   Diagonal and Equal R -354.7055
```

#Question 2 Modify the model above to make it slightly more flexible. We keep the state equation the same as problem 1, since the random walk model with drift makes sense for this data set as we have seen in the previous lecture. However, we will modify the observation equation to allow for different coefficients (a1, a2) for the two series, and we will estimate the initial state parameters instead of fixing it.

##a

```
model.list <- list(
  Z = matrix(1, nrow = 2, ncol = 1),
  A = matrix(list("a1", "a2"),nrow = 2, ncol=1),
  B = matrix(1),
  U = matrix("delta"),
  Q = matrix("q"),
  R = matrix(list("r11", "r12", "r12", "r22"), nrow = 2),
  x0 = "unequal",
  V0 = matrix(1)
)

fit <- MARSS(Y, model = model.list, method = "BFGS")
```

```
## Success! Converged in 17 iterations.
## Function MARSSkfas used for likelihood calculation.
##
## MARSS fit is
## Estimation method: BFGS
## Estimation converged in 17 iterations.
## Log-likelihood: 194.7542
## AIC: -373.5083   AICc: -372.9608
##
##          Estimate
## A.a1      0.09126
## A.a2      0.06949
## R.r11     0.00549
## R.r12     0.00951
## R.r22     0.03220
## U.delta   0.00648
## Q.q       0.00303
## x0.x0    -0.20425
## Initial states (x0) defined at t=0
##
## Standard errors have not been calculated.
## Use MARSSparamCIs to compute CIs and bias estimates.
```

```
logLik(fit)
```

```
## 'log Lik.' 194.7542 (df=8)
```

```
fit$AICc
```

```
## [1] -372.9608
```

```r
residuals <- residuals(fit)
```

##b

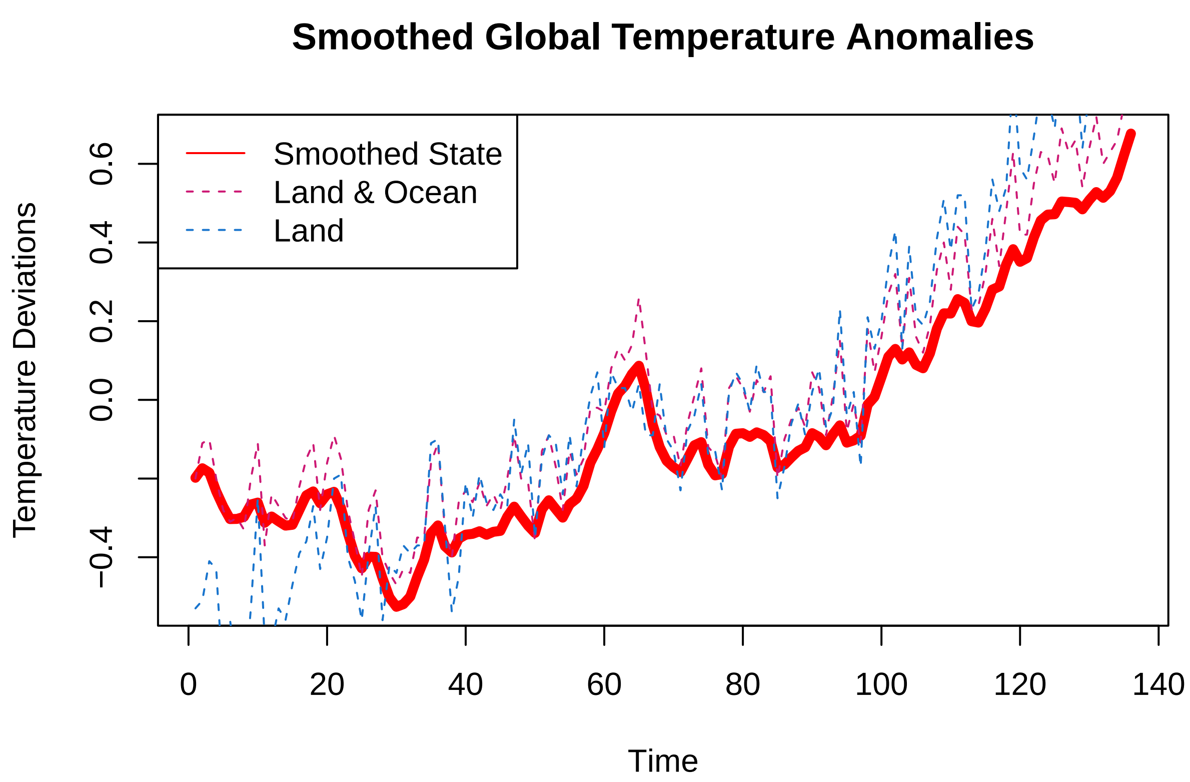
```r
coef_estimates <- coef(fit, type = "vector")
print(coef_estimates)
```

```
##         A.a1         A.a2        R.r11        R.r12        R.r22      U.delta
##  0.091255845  0.069490381  0.005490853  0.009506674  0.032195612  0.006479642
##          Q.q        x0.x0
##  0.003030947 -0.204253775
```

##c

```r
states <- fit$states

time_sequence <- seq_along(Y[, 1])
plot(time_sequence, states[1, ], type = "l", col = "red", ylab = "Temperature Deviations",
     xlab = "Time", main = "Smoothed Global Temperature Anomalies", lwd = 5)
lines(time_sequence, Y[, 1], col = 6, lty = 2)
lines(time_sequence, Y[, 2], col = 4, lty = 2)
legend("topleft", legend = c("Smoothed State", "Land & Ocean", "Land"),
       col = c("red", 6, 4), lty = c(1, 2, 2))
```



Comment: this line is smoother than the one in the previous fit. You see that when you look at the amplitude of fluctuation on each peak and valley.