Trend analysis in time series

Mark Scheuerell
Northwest Fisheries Science Center
National Marine Fisheries Service
Seattle, WA USA
mark.scheuerell@noaa.gov

Trends in time series

Trends in time series can be estimated via simple linear regression where, for a value x measured at time t,

$$x_t = \alpha + \beta t + \epsilon_t.$$

However, any estimates of the significance of α or β will be biased due to non-independence in the observation errors ϵ_t.

Random walks

In a normal random walk, the value at time t equals that at time $t-1$ plus or minus some random error, which are often assumed to be Gaussian. Specifically,

$$x_t = x_{t-1} + \epsilon_t, \tag{1}$$

and $\epsilon_t \sim \mathrm{N}(0, \sigma)$. Random walks are characterized by long deviations into positive or negative space, but there is no overall tendency to go up or down (Figure 1).

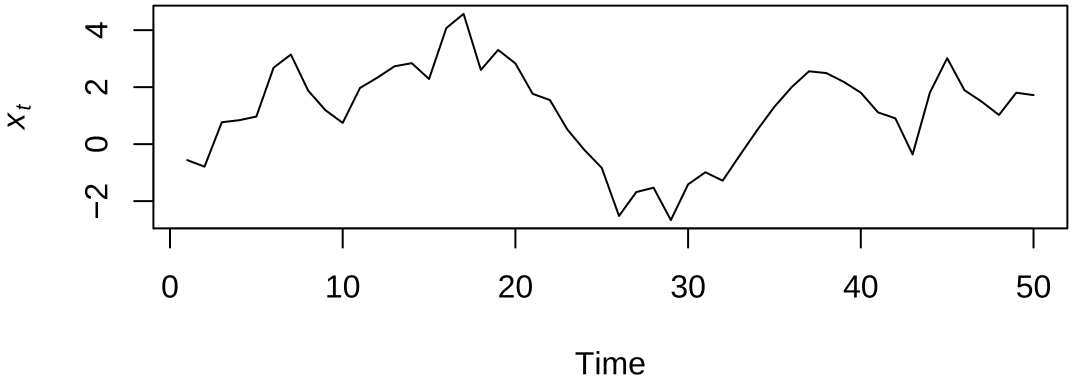

Figure 1: Example of a random walk with Guassian errors.

Biased random walk

An alternative is to assume the observations follow a so-called “biased random walk”. In a biased random walk, the value at time t is still a function of that at time $t-1$ plus or minus some random error. However, there is also an overall tendency (bias) to travel in a generally upward or downward direction (Figure 2). Specifically,

$$x_t = x_{t-1} + \mu + \epsilon_t, \tag{2}$$

μ is the bias, and $\epsilon_t \sim \mathrm{N}(0, \sigma)$.

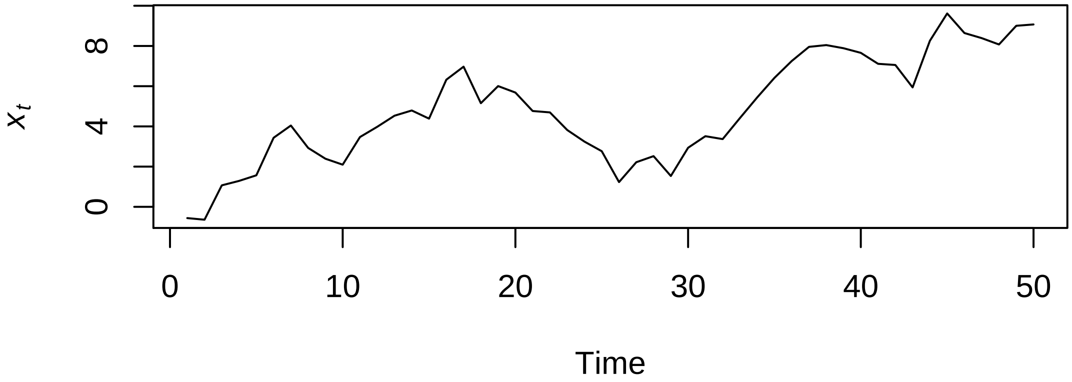

Figure 2: Example of a biased random walk with Guassian errors.

Example: Straying in salmon

There may be a tendency for stray rates in salmon to increase or decrease over time given genetic and environmental effects. In the current case (Figure 3), there indeed appears to be a downward trend in the data.

Because we typically assume Gaussian errors in random walks, but the data lie on the unit interval $[0, 1]$, we must use some form of “link” in our model. The most common choice is the logit function, which maps $[0, 1]$ onto $(-\infty, \infty)$. Our biased random walk model would then become

$$\text{logit}(x_t) = \text{logit}(x_{t-1}) + \mu + \epsilon_t. \tag{3}$$

Note that from here on I will drop the logit notation for simplicity and assume that x_t is an appropriately transformed variate.

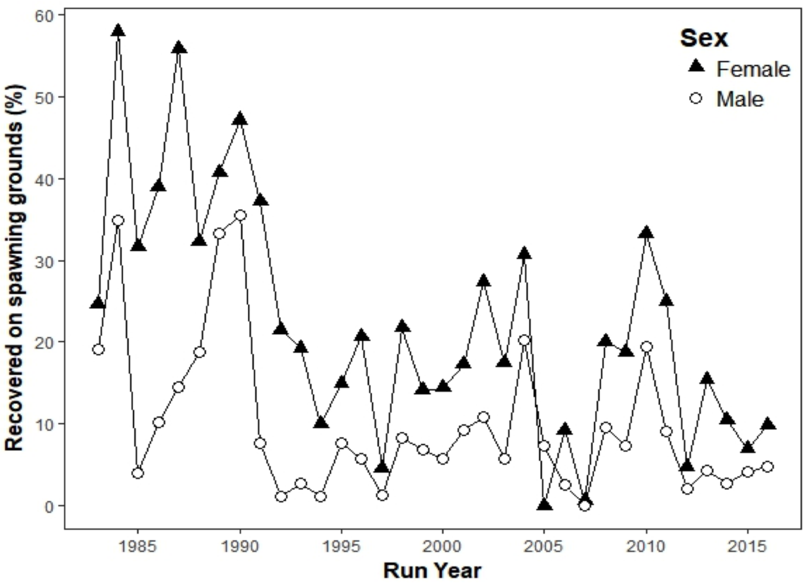

Figure 3: Observed salmon stray rates in the Elk River, Oregon.

In this case there are data for both females (F) and males (M), so if we assume different trends/biases for each of them, then we have

$$x_{F,t} = x_{F,t-1} + \mu_F + \epsilon_{F,t} \tag{4}$$
$$x_{M,t} = x_{M,t-1} + \mu_M + \epsilon_{M,t}. \tag{5}$$

We can combine these two equations into one through matrix notation. If we define $\mathbf{x}_t = [x_{F,t} \; x_{M,t}]^\top$, $\mathbf{u} = [\mu_F \; \mu_M]^\top$, and $\mathbf{e}_t = [\epsilon_{F,t} \; \epsilon_{M,t}]^\top$, then

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \mathbf{u}_t + \mathbf{e}_t \tag{6}$$

and

$$\mathbf{e}_t \sim \text{MVN}(\mathbf{0}, \mathbf{\Sigma}). \tag{7}$$

We can furthermore make assumptions about (i) whether or not the bias is the same for both sexes (*i.e.*, $\mu_F = \mu_M$), and (ii) the extent to which the errors are identical and/or independent. That is, the model as written in (7) assumes that they are IID, such that

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_F & 0 \\ 0 & \sigma_M \end{bmatrix}. \tag{8}$$

On the other hand, one could assume that the errors have the same variance and they co-vary (*e.g.*, there is no genetic difference by sex and changes in the environment affect box sexes similarly). In that case,

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma & \gamma \\ \gamma & \sigma \end{bmatrix}. \tag{9}$$

By fitting different forms of the model, we can then gauge the relative data support for each using some form of information criteria (*e.g.*, AIC). Similarly, we can compare the data support for a model with (2) and without (1) the bias term μ to see whether there is, in fact, a trend in the data.

Observation errors

The above random walk models are commonly referred to as "process" (or state) models because they are meant to represent a time-varying process (or state of nature) from which our data might have arisen. However, we rarely have perfect information from which to estimate the parameters due to sampling or observation errors. In those cases, we can combine our process model with an observation model to form a so-called "state-space" model.

Returning to the univariate model in (2), the observed data y at time t are assumed to be a combination of the true, but unknown state x_t, and some additional observation error v_t, such that

$$y_t = x_y + v_t. \tag{10}$$

The distributional form for v_t can vary depending on the form of the response. For example, if the data were discrete counts, we might use a Poisson or negative binomial. In many cases, Gaussian errors are used for their ease of estimation and the data are transformed, if necessary, to meet the assumption.

Combining equations (2) and (10) leads to the univariate state-space model

$$\begin{aligned} x_t &= x_{t-1} + \mu + \epsilon_t \\ y_t &= x_t + v_t, \end{aligned} \tag{11}$$

which is referred to as a "biased random walk observed with error". Returning to the multivariate model for both sexes in (6), we can write

$$\begin{aligned} \mathbf{x}_t &= \mathbf{x}_{t-1} + \mathbf{u} + \mathbf{e}_t \\ \mathbf{y}_t &= \mathbf{x}_t + \mathbf{v}_t. \end{aligned} \tag{12}$$

Here I assume $\mathbf{v}_t \sim \text{MVN}(\mathbf{0}, \mathbf{R})$. Just as with the process errors in $\mathbf{e}_t$, we can assume different forms for $\mathbf{R}$, fit the models, and evaluate the data support for each.

Fitting random walks

There are a number of ways to estimate the parameters in random walk models, but the options are more limited if you want to fit the state-space version. In particular, I am fond of the MARSS package[1] for **R**.

Here is an example of fitting the multivariate model proposed in (12) with MARSS. I use some dummy data (Figure 4), but it would be trivial to substitute real data. In particular, these data are characterized by

- negative and different biases as in (6); and
- diagonal covariance matrices as in (8) for both the process and observation errors.

Simulate data

```r
## number of processes and observed ts
NN <- 2
## length of ts
TT <- 35
## covariance matrix for process errors
QQ <- diag(c(0.3,0.3))
## process errors; dim is NN x TT
ww <- t(MASS::mvrnorm(TT, matrix(0, NN, 1), QQ))
## covariance matrix for process errors
RR <- diag(c(0.1,0.1))
## obs errors; dim is NN x TT
vv <- t(MASS::mvrnorm(TT, matrix(0, NN, 1), RR))
## neg bias; different by sex
uu <- matrix(c(-0.3,-0.2), NN, 1)
## empty matrices for x & y; dims are NN x TT
xx <- yy <- matrix(NA, NN, TT)
## set x1 to w1
xx[,1] <- ww[,1]
## calc process time series
for(t in 2:TT) {
  xx[,t] <- xx[,t-1] + uu + ww[,t]
}
## calc obs time series
yy <- xx + vv
```

[1] See https://cran.r-project.org/web/packages/MARSS/index.html.

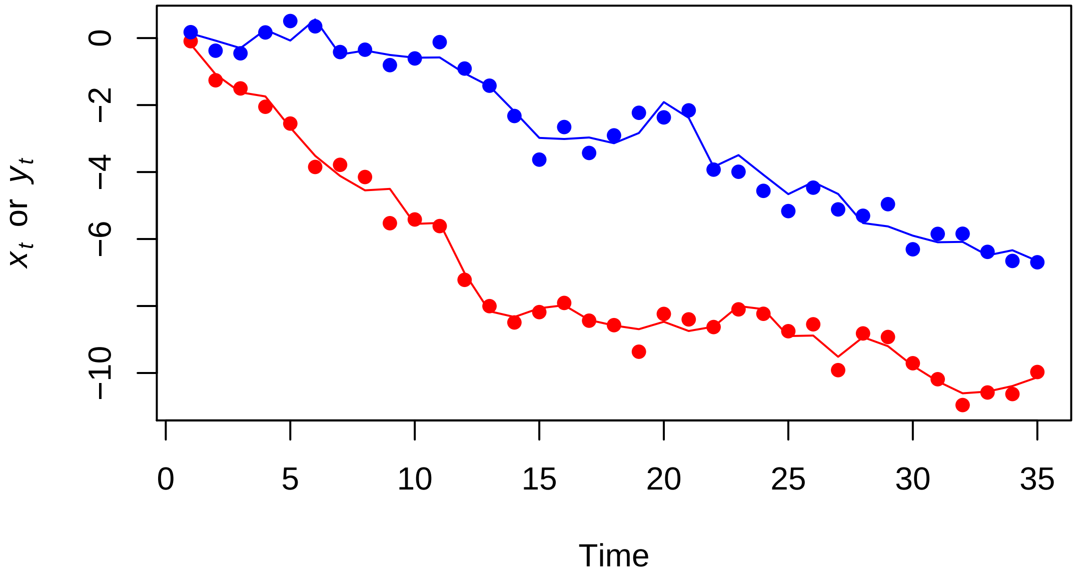

Figure 4: Observations (points) of two negatively biased random walks (lines).

Model fitting

`MARSS` defines state-space models in the same manner that you would on a white board. Thus, the model definition below will look similar to the code above. All we have to do is identify which vectors and matrices `MARSS` should fit, and what form to use.

```
library(MARSS)
## define process model
## bias
UU <- matrix(list("F","M"), NN, 1)
## cov of errs
QQ <- matrix(list(0), NN, NN)
diag(QQ) <- c("F","M")
## define obs model
## cov of errs
RR <- matrix(list(0), NN, NN)
diag(RR) <- c("F","M")
## needed for MARSS, but not in our model
BB <- diag(NN)
ZZ <- diag(NN)
AA <- matrix(0, NN, 1)
## combine for MARSS
mod_list <- list(U = UU, Q = QQ,
                 R = RR,
                 B = BB, Z = ZZ, A = AA,
                 tinitx=0)
```

```
biased_RW <- MARSS(yy, mod_list,
                   control = list(maxit = 2000))
```

```
## Success! abstol and log-log tests passed at 34 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 34 iterations.
## Log-likelihood: -65.61274
## AIC: 147.2255   AICc: 149.5861
##
##         Estimate
## R.F       0.0699
## R.M       0.1029
## U.F      -0.2912
## U.M      -0.2011
## Q.F       0.2778
## Q.M       0.1874
## x0.X.Y1   0.0462
## x0.X.Y2   0.3094
## Initial states (x0) defined at t=0
##
## Standard errors have not been calculated.
## Use MARSSparamCIs to compute CIs and bias estimates.
```

Model comparison

We can also fit a model without the bias term and compare its AICc to that for the model above. We only need to make one change to do so

```
## redefine process model
## bias = 0
mod_list$U <- matrix(0, NN, 1)
## fit unbiased model
unbiased_RW <- MARSS(yy, mod_list,
                     control = list(maxit = 2000, allow.degen = TRUE))
```

```
## Warning! Abstol convergence only. Maxit (=2000) reached before log-log convergence.
##
## MARSS fit is
## Estimation method: kem
## Convergence test: conv.test.slope.tol = 0.5, abstol = 0.001
## WARNING: Abstol convergence only no log-log convergence.
##  maxit (=2000) reached before log-log convergence.
##  The likelihood and params might not be at the ML values.
##  Try setting control$maxit higher.
## Log-likelihood: -71.84669
## AIC: 155.6934   AICc: 157.0267
##
##         Estimate
## R.F      0.00249
## R.M      0.04965
## Q.F      0.49064
## Q.M      0.32780
## x0.X.Y1 -0.09894
## x0.X.Y2  0.11082
## Initial states (x0) defined at t=0
##
## Standard errors have not been calculated.
## Use MARSSparamCIs to compute CIs and bias estimates.
##
## Convergence warnings
##  Warning: the  R.F  parameter value has not converged.
##  Type MARSSinfo("convergence") for more info on this warning.
```

Here's a comparison of the AICc for each model (*i.e.*, corrected for small sample size).

```
biased_RW$AICc
```

```
## [1] 149.5861
```

```
unbiased_RW$AICc
```

```
## [1] 157.0267
```

The data clearly favor the first model with bias terms included, as the difference in AIC is -7.4 units lower. Also note that the biased model converged in 34 iterations of the Kalman filter, but the unbiased model failed to converge after 2000 iterations, suggesting it is not a very good model for the data.