1 Overview of dynamic linear models

Here we will use MARSS to analyze dynamic linear models (DLMs), wherein the parameters in a regression model are treated as time-varying. DLMs are used commonly in econometrics, but have received less attention in the ecological literature (c.f., Lamon et al. 1998, Scheuerell and Williams 2005). Our treatment of DLMs is rather cursory—we direct the reader to excellent textbooks by Pole et al. (1994) and Petris et al. (2009) for more in-depth treatments of DLMs. The former focuses on Bayesian estimation whereas the latter addresses both likelihood-based and Bayesian estimation methods.

We begin our description of DLMs with a *static* regression model, wherein the i^{th} observation is a linear function of an intercept, predictor variable(s), and a random error term. For example, if we had one predictor variable (F), we could write the model as

$$y_i = \alpha + \beta F_i + v_i, \tag{1}$$

where the α is the intercept, β is the regression slope, F_i is the predictor variable matched to the i^{th} observation (y_i), and $v_i \sim \mathrm{N}(0, r)$. It is important to note here that there is no implicit ordering of the index i. That is, we could shuffle any/all of the (y_i, F_i) pairs in our dataset with no effect on our ability to estimate the model parameters. We can write the model in Eqn 1 using vector notation, such that

$$\begin{aligned} y_i &= \begin{pmatrix} 1 & F_i \end{pmatrix} \times \begin{pmatrix} \alpha \\ \beta \end{pmatrix} + v_i \\ &= \mathbf{F}_i^\top \boldsymbol{\theta} + v_i, \end{aligned} \tag{2}$$

and $\mathbf{F}_i^\top = (1, F_i)$ and $\boldsymbol{\theta} = (\alpha, \beta)^\top$.

In a DLM, however, the regression parameters are *dynamic* in that they "evolve" over time. For a single observation at time t, we can write

$$y_t = \mathbf{F}_t^\top \boldsymbol{\theta}_t + v_t, \tag{3}$$

where $\mathbf{F}_t$ is a column vector of regression variables at time t, $\boldsymbol{\theta}_t$ is a column vector of regression parameters at time t and $v_t \sim \mathrm{N}(0, r)$. This formulation presents two features that distinguish it from Eqn 2. First, the observed data are explicitly time ordered (i.e., $\mathbf{y} = \{y_1, y_2, y_3, ..., y_T\}$), which means we expect them to contain implicit information. Second, the relationship between the observed datum and the predictor variables are unique at every time t (i.e., $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3, ..., \boldsymbol{\theta}_T\}$).

However, closer examination of Eqn 3 reveals an apparent complication for parameter estimation. With only one datum at each time step t, we could, at best, estimate only one regression parameter, and even then, the 1:1 correspondence between data and parameters would preclude any estimation of parameter uncertainty. To address this shortcoming, we return to the time ordering of model parameters. Rather than assume the regression parameters are independent from one time step to another, we instead model them as an autoregressive process where

$$\boldsymbol{\theta}_t = \mathbf{G}_t \boldsymbol{\theta}_{t-1} + \mathbf{w}_t, \tag{4}$$

$\mathbf{G}_t$ is the parameter "evolution" matrix, and $\mathbf{w}_t$ is a vector of process errors, such that $\mathbf{w}_t \sim \text{MVN}(\mathbf{0}, \mathbf{Q})$. The elements of $\mathbf{G}_t$ may be known and fixed *a priori*, or unknown and estimated from the data. Although we allow for $\mathbf{G}_t$ to be time-varying, we will typically assume that it is time invariant.

The idea is that the evolution matrix $\mathbf{G}_t$ deterministically maps the parameter space from one time step to the next, so the parameters at time t are temporally related to those before and after. However, the process is corrupted by stochastic error, which amounts to a degradation of information over time. If the diagonal elements of $\mathbf{Q}$ are relatively large, then the parameters can vary widely from t to $t+1$. If $\mathbf{Q} \ = \mathbf{0}$, then $\boldsymbol{\theta}_1 = \boldsymbol{\theta}_2 = \boldsymbol{\theta}_T$ and we are back to the static model in Eqn 1.

2 Example of a univariate DLM

Let's consider an example from the literature. Scheuerell & Williams (2005) used a DLM to examine the relationship between marine survival of Chinook salmon and an index of ocean upwelling strength along the west coast of the USA. Upwelling brings cool, nutrient-rich waters from the deep ocean to shallower coastal areas. Scheuerell & Williams hypothesized that stronger upwelling in April should create better growing conditions for phytoplankton, which would then translate into more zooplankton. In turn, juvenile salmon ("smolts") entering the ocean in May and June should find better foraging opportunities. Thus, for smolts entering the ocean in year t,

$$\textit{survival}_t = \alpha_t + \beta_t F_t + v_t \text{ with } v_t \sim \text{N}(0, r), \tag{5}$$

and F_t is the coastal upwelling index (cubic meters of seawater per second per 100 m of coastline) for the month of April in year t.

Both the intercept and slope are time varying, so

$$\begin{align} \alpha_t &= \alpha_{t-1} + w_t^{(\alpha)} \text{ with } w_t^{(\alpha)} \sim \text{N}(0, q_\alpha); \text{ and} \tag{6} \\ \beta_t &= \beta_{t-1} + w_t^{(\beta)} \text{ with } w_t^{(\beta)} \sim \text{N}(0, q_\beta). \tag{7} \end{align}$$

If we define $\boldsymbol{\theta}_t = (\alpha_t, \beta_t)^\top$, $\mathbf{G}_t = \mathbf{I} \; \forall \; t$, $\mathbf{w}_t = (w_t^{(1)}, w_t^{(2)})^\top$, and $\mathbf{Q} = \text{diag}(q_1, q_2)$, we get Eqn 4. If we define $y_t = \textit{survival}_t$ and $\mathbf{F}_t = (1, F_t)^\top$, we can write out the full univariate DLM as a state-space model with the following form:

$$\begin{gathered} \boldsymbol{\theta}_t = \mathbf{G}_t \boldsymbol{\theta}_{t-1} + \mathbf{w}_t \text{ with } \mathbf{w}_t \sim \text{MVN}(\mathbf{0}, \mathbf{Q}); \\ y_t = \mathbf{F}_t^\top \boldsymbol{\theta}_t + v_t \text{ with } v_t \sim \text{N}(0, r); \\ \boldsymbol{\theta}_0 \sim \text{MVN}(\boldsymbol{\pi}_0, \boldsymbol{\Lambda}_0). \end{gathered} \tag{8}$$

Eqn 8 is, not surprisingly, equivalent to our standard MARSS model:

$$\begin{gathered}
\mathbf{x}_t = \mathbf{B}_t\mathbf{x}_{t-1} + \mathbf{u}_t + \mathbf{C}_t\mathbf{c}_t + \mathbf{w}_t \text{ with } \mathbf{w}_t \sim \text{MVN}(0, \mathbf{Q}_t); \\
\mathbf{y}_t = \mathbf{Z}_t\mathbf{x}_t + \mathbf{a}_t + \mathbf{D}_t\mathbf{d}_t + \mathbf{v}_t \text{ with } \mathbf{v}_t \sim \text{MVN}(0, \mathbf{R}_t); \\
\mathbf{x}_0 \sim \text{MVN}(\boldsymbol{\pi}, \boldsymbol{\Lambda});
\end{gathered} \tag{9}$$

where $\mathbf{x}_t = \boldsymbol{\theta}_t$, $\mathbf{B}_t = \mathbf{G}_t$, $\mathbf{u}_t = \mathbf{C}_t = \mathbf{c}_t = \mathbf{0}$, $\mathbf{y}_t = y_t$ (i.e., $\mathbf{y}_t$ is 1 x 1), $\mathbf{Z}_t = \mathbf{F}_t^\top$, $\mathbf{a}_t = \mathbf{D}_t = \mathbf{d}_t = \mathbf{0}$, and $\mathbf{R}_t = r$ (i.e., $\mathbf{R}_t$ is 1 x 1).

2.1 Fitting a univariate DLM with MARSS

Now let's go ahead and analyze the DLM specified in Eqns 5–8. We begin by getting the data set, which has 3 columns for 1) the year the salmon smolts migrated to the ocean (*year*), 2) logit-transformed survival[1] (*logit.s*), and 3) the coastal upwelling index for April (*CUI.apr*). There are 42 years of data (1964–2005).

```
## load the data
data(SalmonSurvCUI)
## get time indices
years = SalmonSurvCUI[,1]
## number of years of data
TT = length(years)
## get response data: logit(survival)
dat = matrix(SalmonSurvCUI[,2],nrow=1)
```

As we have seen in other case studies, standardizing our covariate(s) to have zero-mean and unit-variance can be helpful in model fitting and interpretation. In this case, it's a good idea because the variance of *CUI.apr* is orders of magnitude greater than *survival*.

```
## get regressor variable
CUI = SalmonSurvCUI[,3]
## z-score the CUI
CUI.z = matrix((CUI - mean(CUI))/sqrt(var(CUI)), nrow=1)
## number of regr params (slope + intercept)
m = dim(CUI.z)[1] + 1
```

Plots of logit-transformed survival and the z-scored April upwelling index are shown in Figure 1.

Next, we need to set up the appropriate matrices and vectors for MARSS. Let's begin with those for the process equation because they are straightforward.

[1] Survival in the original context was defined as the proportion of juveniles that survive to adulthood. Thus, we use the logit function, defined as $logit(p) = log_e(p/[1-p])$, to map survival from the open interval (0,1) onto the interval $(-\infty, \infty)$, which allows us to meet our assumption of normally distributed observation errors.

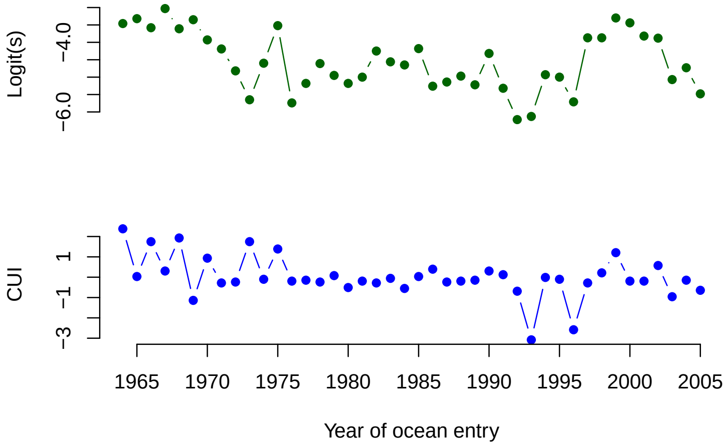

Figure 1: Time series of logit-transformed marine survival estimates for Snake River spring/summer Chinook salmon (top) and z-scores of the coastal upwelling index at 45N 125W (bottom). The x-axis indicates the year that the salmon smolts entered the ocean.

```
## for process eqn
B = diag(m)                        ## 2x2; Identity
U = matrix(0,nrow=m,ncol=1)        ## 2x1; both elements = 0
Q = matrix(list(0),m,m)            ## 2x2; all 0 for now
diag(Q) = c("q.alpha","q.beta") ## 2x2; diag = (q1,q2)
```

Defining the correct form for the observation model is a little more tricky, however, because of how we model the effect(s) of explanatory variables. In a DLM, we need to use $\mathbf{Z}_t$ (instead of $\mathbf{d}_t$) as the matrix of known regressors/drivers that affect $\mathbf{y}_t$, and $\mathbf{x}_t$ (instead of $\mathbf{D}_t$) as the regression parameters. Therefore, we need to set $\mathbf{Z}_t$ equal to an $n \times m \times T$ array, where n is the number of response variables ($= 1$; y_t is univariate), m is the number of regression parameters ($=$ intercept $+$ slope $= 2$), and T is the length of the time series ($= 42$).

```
## for observation eqn
Z = array(NA, c(1,m,TT))    ## NxMxT; empty for now
Z[1,1,] = rep(1,TT)         ## Nx1; 1's for intercept
Z[1,2,] = CUI.z             ## Nx1; regr variable
A = matrix(0)               ## 1x1; scalar = 0
R = matrix("r")             ## 1x1; scalar = r
```

Lastly, we need to define our lists of initial starting values and model matrices/vectors.

```
## only need starting values for regr parameters
inits.list = list(x0=matrix(c(0, 0), nrow=m))
```

```
## list of model matrices & vectors
mod.list = list(B=B, U=U, Q=Q, Z=Z, A=A, R=R)
```

And now we can fit our DLM with MARSS.

```
## fit univariate DLM
dlm1 = MARSS(dat, inits=inits.list, model=mod.list)
```

```
Success! abstol and log-log tests passed at 115 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 115 iterations.
Log-likelihood: -40.03813
AIC: 90.07627   AICc: 91.74293

          Estimate
R.r        0.15708
Q.q.alpha  0.11264
Q.q.beta   0.00564
x0.X1     -3.34023
x0.X2     -0.05388

Standard errors have not been calculated.
Use MARSSparamCIs to compute CIs and bias estimates.
```

Notice that the MARSS output does not list any estimates of the regression parameters themselves. Why not? Remember that in a DLM the matrix of states ($\mathbf{x}$) contains the estimates of the regression parameters ($\boldsymbol{\theta}$). Therefore, we need to look in `dlm1$states` for the MLEs of the regression parameters, and in `dlm1$states.se` for their standard errors.

Time series of the estimated intercept and slope are shown in Figure 2. It appears as though the intercept is much more dynamic than the slope, as indicated by a much larger estimate of process variance for the former (`Q.q1`). In fact, although the effect of April upwelling appears to be increasing over time, it doesn't really become important as an explanatory variable until about 1990 when the approximate 95% confidence interval for the slope no longer overlaps zero.

3 Forecasting with a univariate DLM

Scheuerell & Williams (2005) were interested in how well upwelling could be used to actually *forecast* expected survival of salmon, so let's look at how well

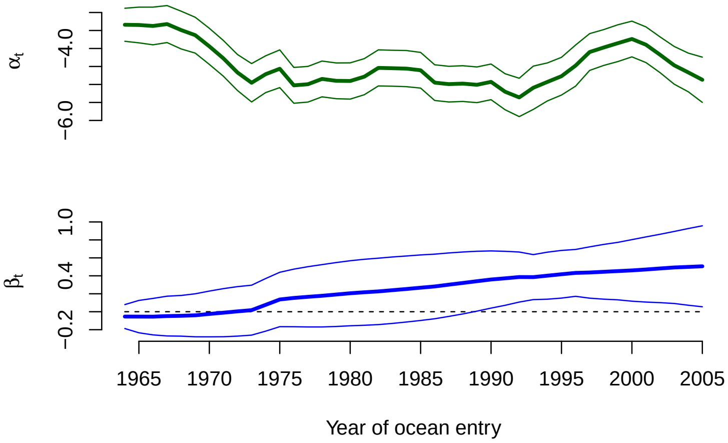

Figure 2: Time series of estimated mean states (thick lines) for the intercept (top) and slope (bottom) parameters from the univariate DLM specified by Eqns 5–8. Thin lines denote the mean ± 2 standard deviations.

our model does in that context. To do so, we need the predictive distributions for the regression parameters and observation.

Beginning with our definition for the distribution of the parameters at time $t = 0$, $\boldsymbol{\theta}_0 \sim \text{MVN}(\boldsymbol{\pi}_0, \boldsymbol{\Lambda}_0)$ in Eqn 8, we write

$$\boldsymbol{\theta}_{t-1} | y_{1:t-1} \sim \text{MVN}(\boldsymbol{\pi}_{t-1}, \boldsymbol{\Lambda}_{t-1}) \tag{10}$$

to indicate the distribution of $\boldsymbol{\theta}$ at time $t-1$ conditioned on the observed data through time $t-1$ (i.e., $y_{1:t-1}$). Then, we can write the one-step ahead predictive distribution for $\boldsymbol{\theta}_t$ given $y_{1:t-1}$ as

$$\begin{aligned} \boldsymbol{\theta}_t | y_{1:t-1} &\sim \text{MVN}(\boldsymbol{\eta}_t, \boldsymbol{\Phi}_t), \text{ where} \\ \boldsymbol{\eta}_t &= \mathbf{G}_t \boldsymbol{\pi}_{t-1}, \text{ and} \\ \boldsymbol{\Phi}_t &= \mathbf{G}_t \boldsymbol{\Lambda}_{t-1} \mathbf{G}_t^\top + \mathbf{Q}. \end{aligned} \tag{11}$$

Consequently, the one-step ahead predictive distribution for the observation at time t given $y_{1:t-1}$ is

$$\begin{aligned} y_t | y_{1:t-1} &\sim \text{N}(\zeta_t, \Psi_t), \text{ where} \\ \zeta_t &= \mathbf{F}_t \boldsymbol{\eta}_t, \text{ and} \\ \Psi_t &= \mathbf{F}_t \Phi_t \mathbf{F}_t^\top + \mathbf{R}. \end{aligned} \tag{12}$$

3.1 Forecasting a univariate DLM with MARSS

Working from Eqn 12, we can now use MARSS to compute the expected value of the forecast at time t ($\mathrm{E}[y_t|y_{1:t-1}] = \zeta_t$), and its variance ($\mathrm{Var}[y_t|y_{1:t-1}] = \Psi_t$). For the expectation, we need $\mathbf{F}_t\boldsymbol{\eta}_t$. Recall that $\mathbf{F}_t$ is our $1 \times m$ matrix of explanatory variables at time t ($\mathbf{F}_t$ is called $\mathbf{Z}_t$ in MARSS notation). The one-step ahead forecasts of the parameters at time t ($\boldsymbol{\eta}_t$) are calculated as part of the Kalman filter algorithm—they are termed $\tilde{x}_t^{t-1}$ in MARSS notation and stored as `'xtt1'` in the list produced by the `MARSSkfss()` function.

```
## get list of Kalman filter output
kf.out = MARSSkfss(dlm1)
## forecasts of regr parameters; 2xT matrix
eta = kf.out$xtt1
## ts of E(forecasts)
fore.mean = vector()
for(t in 1:TT) {
  fore.mean[t] = Z[,,t] %*% eta[,t,drop=F]
}
```

For the variance of the forecasts, we need $\mathbf{F}_t\Phi_t\mathbf{F}_t^\top + \mathbf{R}$. As with the mean, $\mathbf{F}_t \equiv \mathbf{Z}_t$. The variances of the one-step ahead forecasts of the parameters at time t (Φ_t) are also calculated as part of the Kalman filter algorithm—they are stored as `'Vtt1'` in the list produced by the `MARSSkfss()` function. Lastly, the observation variance $\mathbf{R}$ is part of the standard MARSS output.

```
## variance of regr parameters; 1x2xT array
Phi = kf.out$Vtt1
## obs variance; 1x1 matrix
R.est = coef(dlm1, type="matrix")$R
## ts of Var(forecasts)
fore.var = vector()
for(t in 1:TT) {
  tZ = matrix(Z[,,t],m,1) ## transpose of Z
  fore.var[t] = Z[,,t] %*% Phi[,,t] %*% tZ + R.est
}
```

Plots of the model mean forecasts with their estimated uncertainty are shown in Figure 3. Nearly all of the observed values fell within the approximate prediction interval. Notice that we have a forecasted value for the first year of the time series (1964), which may seem at odds with our notion of forecasting at time t based on data available only through time $t-1$. In this case, however, MARSS is actually estimating the states at $t = 0$ ($\boldsymbol{\theta}_0$), which allows us to compute a forecast for the first time point.

Although our model forecasts look reasonable in logit-space, it is worthwhile to examine how well they look when the survival data and forecasts are back-transformed onto the interval [0,1] (Figure 4). In that case, the accuracy does not seem to be affected, but the precision appears much worse, especially during

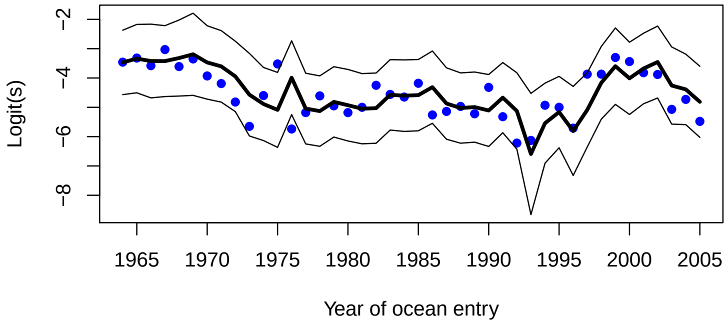

Figure 3: Time series of logit-transformed survival data (blue dots) and model mean forecasts (thick line). Thin lines denote the approximate 95% prediction intervals.

the early and late portions of the time series when survival is changing rapidly.

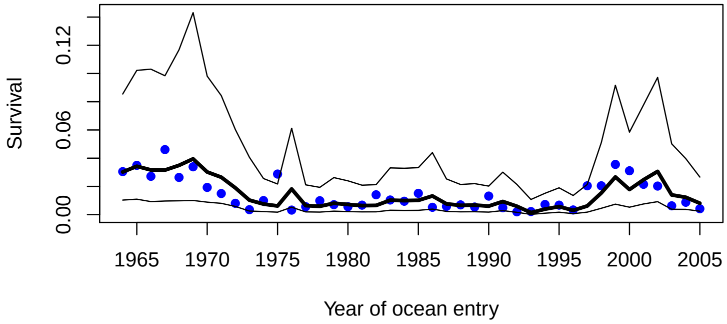

Figure 4: Time series of survival data (blue dots) and model mean forecasts (thick line). Thin lines denote the approximate 95% prediction intervals.

3.2 DLM forecast diagnostics

As with other time series models, evaluation of a DLM should include some model diagnostics. In a forecasting context, we are often interested in the forecast errors, which are simply the observed data minus the forecasts ($e_t = y_t - \zeta_t$). In particular, the following assumptions should hold true for e_t:

1) $e_t \sim \mathrm{N}(0, \sigma^2)$;

2) $\mathrm{cov}(e_t, e_{t-k}) = 0$.

In the literature on state-space models, the set of e_t are commonly referred to as "innovations". `MARSS()` calculates the innovations as part of the Kalman filter algorithm—they are stored as `'Innov'` in the list produced by the `MARSSkfss()` function.

```
## forecast errors
innov = kf.out$Innov
```

Let's see if our innovations meet the model assumptions. Beginning with (1), we can use a Q-Q plot to see whether the innovations are normally distributed with a mean of zero. We'll use the `qqnorm()` function to plot the quantiles of the innovations on the y-axis versus the theoretical quantiles from a Normal distribution on the x-axis. If the 2 distributions are similar, the points should fall on the line defined by $y = x$.

```
## Q-Q plot of innovations
qqnorm(t(innov), main="", pch=16, col="blue")
## add y=x line for easier interpretation
qqline(t(innov))
```

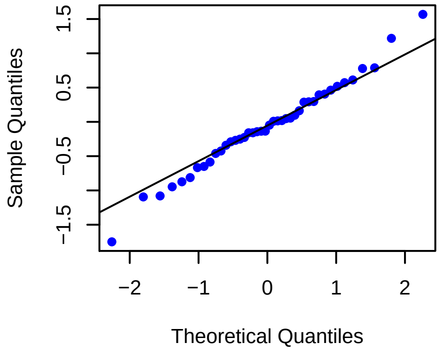

Figure 5: Q-Q plot of the forecast errors (innovations) for the DLM specified in Eqns 5–8.

The Q-Q plot (Figure 5) indicates that the innovations appear to be more-or-less normally distributed (i.e., most points fall on the line). Furthermore, it looks like the mean of the innovations is about 0, but we should use a more reliable test than simple visual inspection. We can formally test whether the mean of the innovations is significantly different from 0 by using a one-sample t-test. based on a null hypothesis of $\mathrm{E}(e_t) = 0$. To do so, we will use the function `t.test()` and base our inference on a significance value of $\alpha = 0.05$.

```
## p-value for t-test of H0: E(innov) = 0
t.test(t(innov), mu=0)$p.value
```

```
[1] 0.4840901
```

The p-value $>> 0.05$ so we cannot reject the null hypothesis that $\mathrm{E}(e_t) = 0$.

Moving on to assumption (2), we can use the sample autocorrelation function (ACF) to examine whether the innovations covary with a time-lagged version of themselves. Using the `acf()` function, we can compute and plot the correlations of e_t and e_{t-k} for various values of k. Assumption (2) will be met if none of the correlation coefficients exceed the 95% confidence intervals defined by $\pm\, z_{0.975}/\sqrt{n}$.

```
## plot ACF of innovations
acf(t(innov), lag.max=10)
```

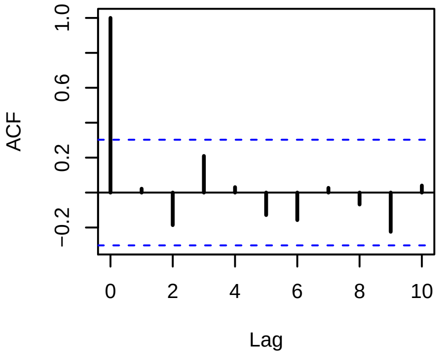

Figure 6: Autocorrelation plot of the forecast errors (innovations) for the DLM specified in Eqns 5–8. Horizontal blue lines define the upper and lower 95% confidence intervals.

The ACF plot (Figure 6) shows no significant autocorrelation in the innovations at lags 1–10, so it looks like both of our model assumptions have indeed been met.

4 Homework problems

For the homework this week we will use a DLM to examine some of the time-varying properties of the spawner-recruit relationship for Pacific salmon. Much work has been done on this topic, particularly by Randall Peterman and his students and post-docs at Simon Fraser University. To do so, researchers commonly use a Ricker model because of its relatively simple form, such that the number of recruits (offspring) born in year t (R_t) from the number of spawners (parents) (S_t) is

$$R_t = aS_t e^{-bS+v_t}. \tag{13}$$

The parameter a determines the maximum reproductive rate in the absence of any density-dependent effects (the slope of the curve at the origin), b is the strength of density dependence, and $v_t \sim N(0, \sigma)$. In practice, the model is typically log-transformed so as to make it linear with respect to the predictor variable S_t, such that

$$\begin{aligned} \log(R_t) &= \log(a) + \log(S_t) - bS_t + v_t \\ \log(R_t) - \log(S_t) &= \log(a) - bS_t + v_t \\ \log(R_t/S_t) &= \log(a) - bS_t + v_t. \end{aligned} \tag{14}$$

Substituting $y_t = \log(R_t/S_t)$, $x_t = S_t$, and $\alpha = \log(a)$ yields a simple linear regression model with intercept α and slope b.

Unfortunately, however, residuals from this simple model typically show high-autocorrelation due to common environmental conditions that affect overlaping generations. Therefore, to correct for this and allow for an index of stock productivity that controls for any density-dependent effects, the model may be re-witten as

$$\begin{aligned} \log(R_t/S_t) &= \alpha_t - bS_t + v_t, \\ \alpha_t &= \alpha_{t-1} + w_t, \end{aligned} \tag{15}$$

and $w_t \sim N(0, q)$. By treating the brood-year specific productivity as a random walk, we allow it to vary, but in an autocorrelated manner so that consecutive years are not independent from one another.

More recently, interest has grown in using covariates (*e.g.*, sea-surface temperature) to explain the interannual variability in productivity. In that case, we can can write the model as

$$\log(R_t/S_t) = \alpha + \delta_t X_t - bS_t + v_t. \tag{16}$$

In this case we are estimating some base-level productivity (α) plus the time-varying effect of some covariate X_t (δ_t).

4.1 Spawner-recruit data

The data come from a large public database begun by Ransom Myers many years ago. If you are interested, you can find lots of time series of spawning-stock, recruitment, and harvest for a variety of fishes around the globe. Here is the website:

`http://ram.biology.dal.ca/~myers/about_site.html`

For this exercise, we will use spawner-recruit data for sockeye salmon (*Oncorhynchus nerka*) from the Kvichak River in SW Alaska that span the years 1952-1989. In addition, we'll examine the potential effects of the Pacific Decadal Oscillation (PDO) during the salmon's first year in the ocean, which is widely believed to be a "bottleneck" to survival.

Here are the data:

```
## get S-R data; object is 'SRdata"
## file cols are:
## 1: brood yr (brood.yr)
## 2: number of spawners (Sp)
## 3: number of recruits (Rec)
## 4: PDO during first summer at sea (PDO.t2)
## 5: PDO during first winter at sea (PDO.t3)
load("KvichakSockeye.RData")
## head of data file
head(SRdata)
```

```
  brood.yr   Sp   Rec PDO.t2 PDO.t3
1     1952 5970 17310  -0.61  -0.61
2     1953  320   520  -1.48  -2.66
3     1954  240   750  -2.05  -1.26
4     1955  250  1280   0.01   0.11
5     1956 9443 39036   0.86   0.37
6     1957 2843  4091  -0.25   0.29
```

4.2 Questions

Use the information above to answer the following questions. Note that if any model is not converging, then you will need to increase the `maxit` parameter in the `control` argument/list that gets passed to `MARSS`. For example, you might try `control=list(maxit=2000)`.

1. Begin by fitting a reduced form of Equation 15 that includes only a time-varying level (α_t) and observation error (v_t). That is,

$$\begin{aligned}\log(R_t) &= \alpha_t + \log(S_t) + v_t \\ \log(R_t/S_t) &= \alpha_t + v_t\end{aligned} \tag{17}$$

This model assumes no density-dependent survival in that the number of recruits is an ascending function of spawners. Plot the ts of α_t and note the AICc for this model. Also plot appropriate model diagnostics.

2. Fit the full model specified by Equation 15. For this model, obtain the time series of α_t, which is an estimate of the stock productivity in the absence of density-dependent effects. How do these estimates of productivity compare to those from the previous question? Plot the ts of α_t and note the AICc for this model. Also plot appropriate model diagnostics. (*Hint*: If you don't want a parameter to vary with time, what does that say about its process variance?)

3. Fit the model specified by Equation 16 with the summer PDO index as the covariate (`PDO.t2`). What is the mean level of productivity? Plot the ts of δ_t and note the AICc for this model. Also plot appropriate model diagnostics.

4. Fit the model specified by Equation 16 with the winter PDO index as the covariate (`PDO.t3`). What is the mean level of productivity? Plot the ts of δ_t and note the AICc for this model. Also plot appropriate model diagnostics.

5. Based on AICc, which of the models above is the most parsimonius? Is it well behaved (*i.e.*, are the model assumptions met)? Plot the model forecasts for the best model. Is this a good forecast model?