Imputation of missing covariates for use in modeling growth rates of juvenile Chinook salmon

Mark Scheuerell
Fish Ecology Division, Northwest Fisheries Science Center, National Marine Fisheries Service, National Oceanic and Atmospheric Administration, Seattle, WA, USA

Karl Veggerby
Ocean Associates, Seattle, WA, USA

This is version 0.19.04.26.

Background

We would like to examine temperature as a possible covariate in the analyses of the growth data. However, the temperature are not complete for all sites and times, and therefore we have to impute some of the values based upon measurements from two different sources: Team Carcass and Team Tagging. To do so, we will use the **MARSS** package, which is designed to fit autoregressive models to multivariate time series data.

Specifically, we have data from $n = 7$ streams and 1-2 sources for each stream. For temperature measured at site i from source j on day d ($T_{i,j,d}$), we can write

$$\begin{gathered} \mathbf{T}_d = \mathbf{Z}\mathbf{x}_d + \mathbf{a} + \mathbf{v}_d \\ \mathbf{x}_d = \mathbf{B}\mathbf{x}_{d-1} + \mathbf{C}\mathbf{c}_{d-h} + \mathbf{w}_d, \end{gathered} \tag{1}$$

where $\mathbf{T}_d$ is an $n \times 1$ vector of the $T_{i,j,d}$ and $\mathbf{x}_d$ is an $m \times 1$ vector of the true, but unobserved temperature in each stream on day d. Here, $m < n$ because we have 2 sources of data from some streams. The vectors of observation ($\mathbf{v}_d$) and process ($\mathbf{w}_d$) errors are both distributed as multivariate normal with means $\mathbf{0}$ and covariance matrices $\mathbf{R}$ and $\mathbf{Q}$, respectively.

The specific forms for $\mathbf{Z}$, $\mathbf{a}$ and $\mathbf{B}$ will be chosen initially based on visually identified shared/different characteristics among the different sites. The values in the matrix $\mathbf{C}$ determine the effect(s) of any potential, and possibly lagged, covariates contained in $\mathbf{c}_{d-h}$. Those will also be chosen after inspection of the data from the different sources

Requirements

```r
## for analysis
library(MARSS)
## for plotting
library(viridisLite)
```

```
## for dir mgmt
library(here)
datadir <- here("data")
analdir <- here("analysis")
```

Data munging

We begin by loading the data file with the temperature summaries by time (rows) and location (cols).

```
## load obs covariates
cobs <- read.csv(file.path(datadir, "daily_mean_temp.csv"),
                 stringsAsFactors = FALSE)
## inspect data
head(cobs)
```

```
##   year doy data_source_BVA data_source_CHO data_source_ELK data_source_MAR
## 1 2003 122    team carcass    team carcass    team carcass    achord_gordy
## 2 2003 123    team carcass    team carcass    team carcass    achord_gordy
## 3 2003 124    team carcass    team carcass    team carcass    achord_gordy
## 4 2003 125    team carcass    team carcass    team carcass    achord_gordy
## 5 2003 126    team carcass    team carcass    team carcass    achord_gordy
## 6 2003 127    team carcass    team carcass    team carcass    achord_gordy
##   data_source_LAK data_source_SFS data_source_VAL daily_mean_BVA
## 1    team carcass    achord_gordy    achord_gordy             NA
## 2    team carcass    achord_gordy    achord_gordy             NA
## 3    team carcass    achord_gordy    achord_gordy             NA
## 4    team carcass    achord_gordy    achord_gordy             NA
## 5    team carcass    achord_gordy    achord_gordy             NA
## 6    team carcass    achord_gordy    achord_gordy             NA
##   daily_mean_ELK daily_mean_LAK daily_mean_MAR daily_mean_SFS
## 1             NA             NA       2.964167       3.943333
## 2             NA             NA       2.349583       3.751667
## 3             NA             NA       2.561667       3.743333
## 4             NA             NA       3.106250       3.901250
## 5             NA             NA       3.071667       3.506250
## 6             NA             NA       4.188333       4.332083
##   daily_mean_VAL daily_mean_CHO
## 1       6.193333             NA
## 2       5.201250             NA
## 3       5.589583             NA
## 4       6.169583             NA
## 5       5.454583             NA
## 6       6.882500             NA
```

Let’s simplify some names in the file and round the observations to the nearest 0.01.

```
## simplify colnames
```

```
colnames(cobs) <- gsub("data_source", replacement = "s", x = colnames(cobs))
colnames(cobs) <- gsub("daily_mean", replacement = "T", x = colnames(cobs))
## site abbrevs
sites <- sort(gsub("T_", replacement = "", x = colnames(cobs)[10:16]))
## simplify sources
cobs[cobs=="team carcass"] <- "car"
cobs[cobs=="achord_gordy"] <- "tag"
## round temps
cobs[,grep("T_", colnames(cobs))] <- round(cobs[,grep("T_", colnames(cobs))], 2)
```

The temperature data come from two different sources:

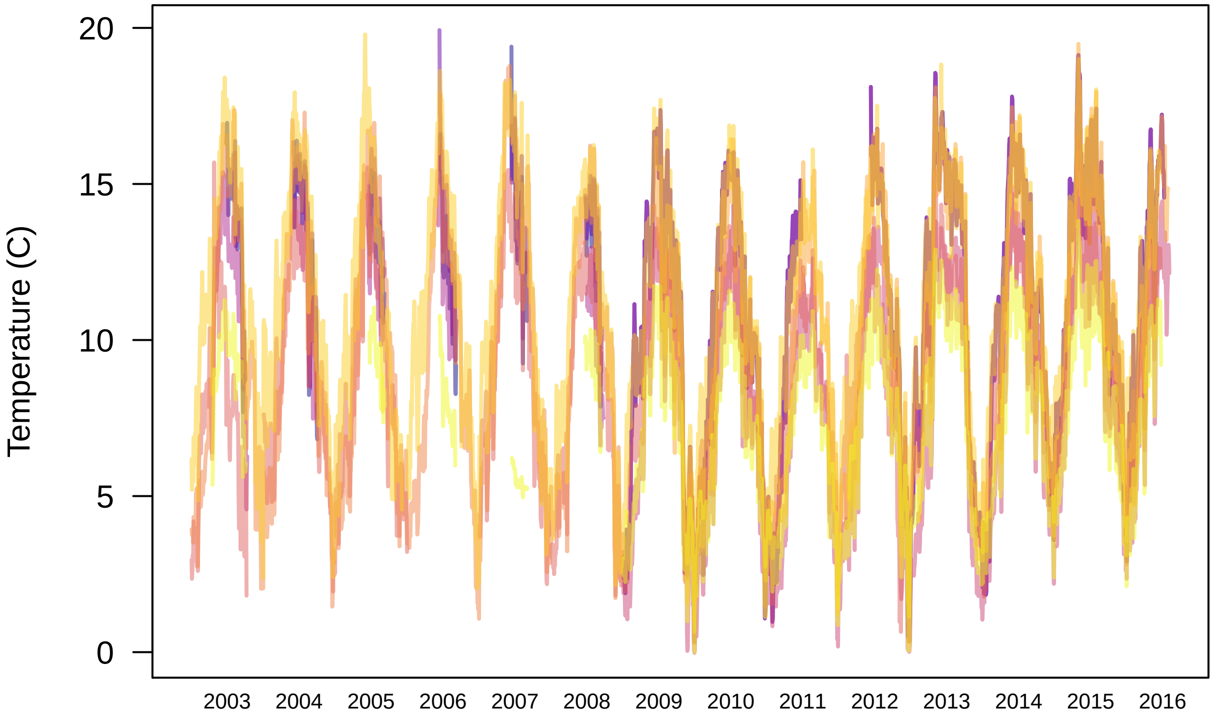
1. the food web group (Sanderson et al.), and
2. the juvenile tagging group (Achord, Axel et al.).

We need to split the data out by those groups.

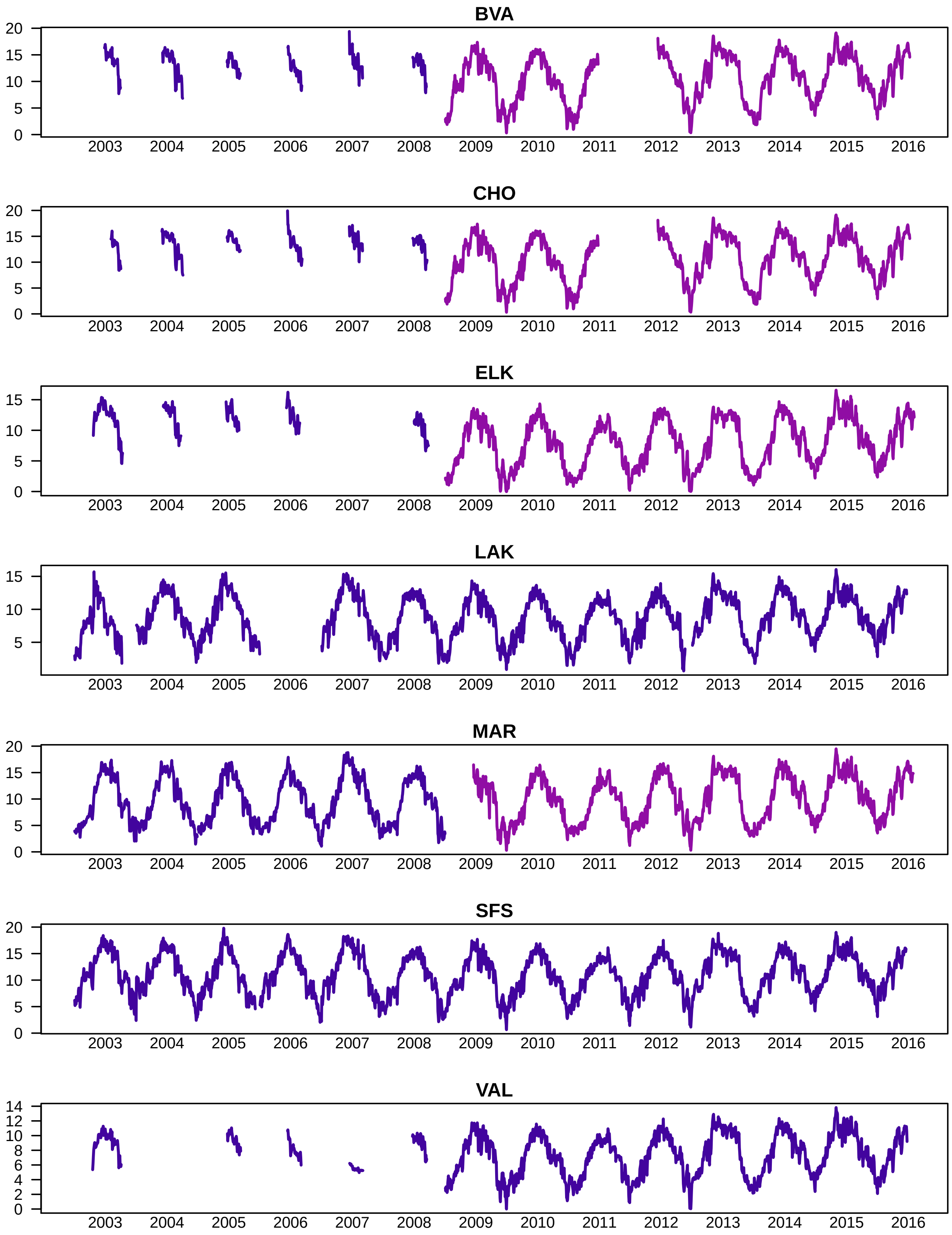
```
## empty data frames
car <- tag <- matrix(NA, nrow(cobs), length(sites),
                     dimnames = list(NULL, sites))
## measurements
vals <- cobs[,-(1:9)]
## indices of data type
i_car <- cobs[,3:9] == "car"
i_tag <- cobs[,3:9] == "tag"
## group-specific data
car[i_car] <- vals[i_car]
tag[i_tag] <- vals[i_tag]
## drop any sites with all NA's
car <- car[,apply(car, 2, function(x) !all(is.na(x)))]
colnames(car) <- paste0(colnames(car), "_car")
tag <- tag[,apply(tag, 2, function(x) !all(is.na(x)))]
colnames(tag) <- paste0(colnames(tag), "_tag")
## regroup by sites & source
cobs_m <- cbind(car, tag)
cobs_m <- cobs_m[,sort(colnames(cobs_m))]
## remove tailing rows with all NA
all_na <- apply(apply(cobs_m, 1, is.na), 2, all)
cobs_m <- cobs_m[!all_na,]
```

Plot the data

All sites together



By site


BVA
0
5
10
15
20
CHO
0
5
10
15
20
ELK
0
5
10
15
LAK
5
10
15
MAR
0
5
10
15
20
SFS
0
5
10
15
20
VAL
0
2
4
6
8
10
12
14
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016

Imputation

Based on the plots above, we will use the following forms for the vectors and matrices in Eqn (1) that relate the states to themselves and the observations to the states. Specifically, it looks as though the seasonal patterns dominate in each stream, with very little evidence for an increasing or decreasing trend. Furthermore, other than subtle changes in the overall level (mean), the streams appear to move up and down in synchrony. Thus, we will set

- $\mathbf{Z}$ equal to an $n \times 1$ column vector of 1's ($[1\ 1 \dots 1]^\top$), such that all of the data are assumed to be observations of a single overall temperature regime;
- $\mathbf{a}$ equal to an $n \times 1$ column vector where the elements are shared for data from the same stream (*e.g.*, both sets of data for Elk would get the same level, but Elk is different from Valley);
- $\mathbf{v}_d \sim \text{MVN}(\mathbf{0}, \mathbf{R})$ with $\mathbf{R}$ equal to an $n \times n$ matrix with the same variance (r) in each element of the diagonal and 0's elsewhere,

$$\mathbf{R} = \begin{bmatrix} r & 0 & \dots & 0 \\ 0 & r & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & r \end{bmatrix};$$

- $\mathbf{B}$ equal to an $m \times m$ matrix with different parameters down the diagonal and 0's elsewhere, which will allow the true temperature in each stream to follow a first-order autoregressive process,

$$\mathbf{B} = \begin{bmatrix} b_1 & 0 & \dots & 0 \\ 0 & b_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & b_m \end{bmatrix};$$

- $\mathbf{C}$ equal to an $m \times 1$ vector with unique parameters to allow for varying effects of the seasonal signal on each stream ($[C_1\ C_2 \dots C_m]^\top$);
- $\mathbf{c}$ equal to a discrete sine wave with a period of 365 days, such that $\mathbf{c}_t$ is a scalar; and
- $\mathbf{w}_d \sim \text{MVN}(\mathbf{0}, \mathbf{Q})$ with $\mathbf{Q}$ equal to an $m \times m$ matrix with the same variance (q) in each element of the diagonal and the same covariance (p) elsewhere,

$$\mathbf{Q} = \begin{bmatrix} q & p & \dots & p \\ p & q & \dots & p \\ \vdots & \vdots & \ddots & \vdots \\ p & p & \dots & q \end{bmatrix}.$$

MARSS modeling

Now we can translate our model from Eqn (1) into a form suitable for `MARSS()`.

Observation equation

We begin with the observation equation.

```
## number of data sets
nn <- dim(cobs_m)[2]
## number of streams
mm <- length(sites)
## empty list for model defn
mod_list <- list()
## Z
# ZZ <- matrix(1, nrow = nn, ncol = 1)
ZZ <- matrix(0, nrow = nn, ncol = mm)
for(i in seq(mm)) {
  ZZ[which(gsub("_.*", "", colnames(cobs_m)) %in% sites[i]),i] <- 1
}
mod_list$Z <- ZZ
## a
aa <- matrix(list(0), nrow = nn, ncol = 1)
# aa[,] <- colnames(cobs_m)
# aa[grep("tag", colnames(cobs_m)),] <- 0
mod_list$A <- aa
## R
RR <- matrix(list(0), nrow = nn, ncol = nn)
diag(RR) <- rep("r", nn)
mod_list$R <- RR
```

Process equation

And now the process (state) equation.

```
## B
BB <- matrix(list(0), nrow = mm, ncol = mm)
diag(BB) <- sites
mod_list$B <- BB
## u
mod_list$U <- matrix(0, nrow = mm, ncol = 1)
## C
# mod_list$C <- matrix(sites, nrow = mm, ncol = 1)
## c (discrete sine wave)
# mod_list$c <- matrix(sin(2 * pi * seq(nrow(cobs_m)) / n_days), nrow = 1)
## Q
QQ <- matrix(list("p"), nrow = mm, ncol = mm)
diag(QQ) <- rep("q", mm)
mod_list$Q <- QQ
```

Fit the model

```
## fit base model (if not already saved)
if(!file.exists(file.path(analdir, "temp_imputed.rds"))) {
  tmp <- MARSS(y = t(cobs_m), model = mod_list, control = list(maxit = 2000))
  ## save results to file
  saveRDS(tmp, file.path(analdir, "temp_imputed.rds"))
}
```

```
## Success! abstol and log-log tests passed at 359 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 359 iterations.
## Log-likelihood: -11104.33
## AIC: 22242.67   AICc: 22242.72
##
##       Estimate
## R.r     0.0418
## B.BVA   0.9883
## B.CHO   0.9886
## B.ELK   0.9861
## B.LAK   0.9851
## B.MAR   0.9886
## B.SFS   0.9886
## B.VAL   0.9823
## Q.q     0.6432
## Q.p     0.5282
## x0.X1   8.3238
## x0.X2   6.1139
## x0.X3   4.6533
## x0.X4   2.9893
## x0.X5   4.0477
## x0.X6   6.1769
## x0.X7   2.8008
## Initial states (x0) defined at t=0
##
## Standard errors have not been calculated.
## Use MARSSparamCIs to compute CIs and bias estimates.
```