

# Multivariate/ Vector Autoregressive Model with Covariates

*Stan users mailing list*

*December 11, 2015*

Multivariate autoregressive (MAR) and vector autoregressive (VAR) are the same thing, ecologists call them MAR. Here we fit a MAR(1) model and include covariates. The model is fit using `stats::ar()`, `vars::VAR()`, and a model written in Stan. A VAR model was written in Stan by Rob Trangucci, Ryan Batt added covariates to it and put this document together. Find related discussions on the Stan users mailing list [here](#) and [here](#).

The model presented here is framed after [Ives et al. \(2003\)](#); in particular, Eq. 27 (quoting from the paper):

$$X_t = A + BX_{t-1} + CU_t + E_t$$

“where  $U_t$  is a  $q \times 1$  vector containing the values of  $q$  covariates at time  $t$ , and  $C$  is a  $p \times q$  matrix whose elements  $c_{ij}$  give the strength of effect of covariates  $j$  on species  $i$ . The covariates  $U_t$  can be any factors that affect the system ...”

The purpose of this document is to fit a MAR model in Stan. The data are simulated with an ecological process in mind (nutrients, phytoplankton [algae], and zooplankton [herbivores]), but that’s just a loose framing and shouldn’t be read-into too much.

```
# =====  
# = Load Packages =  
# =====  
library(vars)  
library(rstan)
```

```
# =====  
# = Setup =  
# =====  
set.seed(1337)  
nT <- 2E2  
nX <- 2 # don't change  
nU <- 1 # don't change  
e_sd <- 1  
  
A <- matrix(c(0.1, 10), nrow=nX, ncol=1)  
B <- matrix(c(0.75, -0.8, 0.2, 0.95), nrow=nX, ncol=nX)  
C <- matrix(c(0,1), nrow=nX, ncol=nU)  
  
X <- matrix(NA, nrow=nT, ncol=2, dimnames=list(NULL, c("zoop","phyto")))  
U <- matrix(rlnorm(nT*nU), nrow=nT, ncol=nU, dimnames=list(NULL, c("nuts")))  
E <- matrix(rnorm(nT*nX) * e_sd, nrow=nT, ncol=nX)
```

```
# =====  
# = Simulate =  
# =====  
X[1,] <- c(3,10)
```

```

for(i in 2:nT){
  X[i,] <- A + B*%matrix(X[i-1,],ncol=1) + C*%matrix(U[i,],ncol=1) + E[i,]
}

# =====
# = Plot Simulate =
# =====
par(mfrow=c(2,1), mar=c(2,2,0.1,2), cex=1, ps=9, mgp=c(1,0.1,0), tcl=-0.1)
plot(X[,1],
     type="l",
     ylim=c(min(X,0), max(X)),
     ylab="X (zoops [black] phytos [blue])",
     xlab="time step"
)

lines(X[,2], col="blue")

par(new=TRUE)
plot(U,
     type="l",
     col="red",
     xlab="", ylab="",
     xaxt="n", yaxt="n",
     lwd=0.5,
     lty="dashed"
)
axis(side=4, col="red")
mtext("U (nuts [red])", side=4, line=1)

plot(X[,2:1], type="o", lwd=0.25, pch=20)

```

Nutrients come into the system as a log-normal random variate; this representation loosely represents the distribution of nutrient pulses into a lake that would be observed in nature. This description and the simulation doesn't emphasize the distinction between nutrient loading and nutrient concentration. When there is a spike in the nutrients (red dashed line), phytoplankton jump up ( $C[2,1]$ ). Then in the next time step, the zooplankton benefit from a greater presence of phytoplankton in the previous time step ( $B[1,2]$  in the simulation, or  $B[1,1,2]$  in the Stan model output). On the other hand, phytoplankton abundance is intensely suppressed by zooplankton abundance in the previous time step ( $B[2,1]$  in simulation, or  $B[1,2,1]$  in Stan output). Thus, a nutrient pulse quickly increases phytoplankton, which are subsequently suppressed by zooplankton. Further, the diagonal elements of  $B$  are less than 1 and greater than 0, indicating a steady decline in the abundance of both phytoplankton and zooplankton, but the decline is balanced a steady (constant) input from  $A$ . The  $A$  parameter is quite small for zooplankton ( $A_1 = 0.1$ ) and much larger for phytoplankton ( $A_2 = 10$ ).

The top panel shows the time series of these state variables, where the spikes in nutrients and plankton are all plainly visible, as are the shifted timings and differences in the relative sizes of the spikes.

The bottom panel of the figure shows the phase plot of the two state variables ( $X$ ). The effects of large nutrient pulses are seen as the rapid excursions to the right (more phytoplankton), which are followed by the system moving up (more zooplankton) and to the left (fewer phytoplankton), and eventually spiralling counterclockwise back in toward the equilibrium point (which the system may never perfectly reach due to shocks  $[E]$  and nutrient pulses  $[U]$  bombarding the system at each time step).

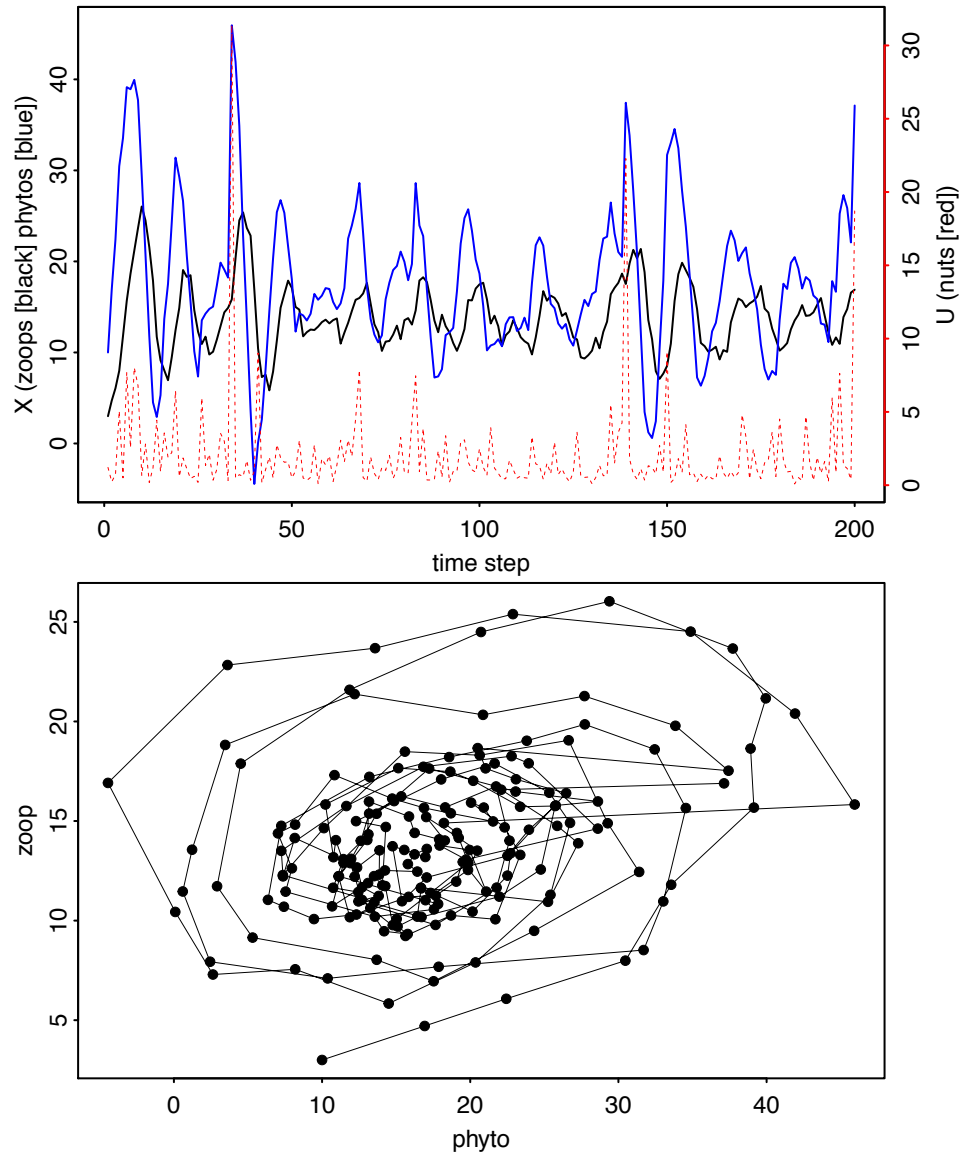


Figure 1: Time series of zooplankton ( $X[1]$ , black), phytoplankton ( $X[2]$ , blue), and nutrients ( $U$ , red)

```

# =====
# = Stan Model =
# =====
cat(readLines('VAR_p_cov.stan', warn=FALSE), sep="\n")

## data {
##   int<lower=0> T; // No. of time periods observed
##   int<lower=0> M; // No. of variables in y_t
##   int<lower=1> P; // Lag order
##   vector[M] Y[T]; // Obs of state variables
##   int nU; // No. of covariates
##   vector[nU] U[T]; // Covariates
## }
## transformed data {
##   vector[M] Y_obs[T-P];
##   for (t in 1:(T-P))
##     Y_obs[t] <- Y[t + P];
## }
## parameters {
##   matrix[M, M] B[P];
##   cholesky_factor_corr[M] L_corr_noise;
##   vector<lower=0>[M] sd_noise;
##   vector[M] A;
##   matrix[M, nU] C;
## }
## transformed parameters {
##   matrix[M,M] L_sigma;
##   L_sigma <- diag_pre_multiply(sd_noise, L_corr_noise);
## }
## model {
##   vector[M] mus[T-P];
##   for (t in 1:(T-P)) {
##     mus[t] <- A + C * U[t+P];
##     for (p in 1:P)
##       mus[t] <- mus[t] + B[p] * Y[t+p-1];
##   }
##   L_corr_noise ~ lkj_corr_cholesky(2.0);
##   sd_noise ~ normal(0,1);
##   A ~ normal(0,1);
##
##   for (p in 1:P)
##     to_vector(B[p]) ~ normal(0, 1);
##   Y_obs ~ multi_normal_cholesky(mus,L_sigma);
## }
## generated quantities {
##   matrix[M,M] Sigma;
##   Sigma <- L_sigma * L_sigma';
## }

# =====
# = Fit Models =
# =====

```

```
# MAR
ar(X) # can't handle covariates :(
```

```
##
## Call:
## ar(x = X)
##
## $ar
## , , 1
##
##      zoop  phyto
## zoop  0.7197 0.1918
## phyto -0.7783 0.9448
##
##
## $var.pred
##      zoop  phyto
## zoop  1.7656 0.8752
## phyto 0.8752 14.3658
```

```
# VAR
VAR(X, exogen=U)
```

```
##
## VAR Estimation Results:
## =====
##
## Estimated coefficients for equation zoop:
## =====
## Call:
## zoop = zoop.l1 + phyto.l1 + const + nuts
##
##      zoop.l1    phyto.l1      const      nuts
## 0.720085330 0.200444289 0.417703606 -0.007097964
##
##
## Estimated coefficients for equation phyto:
## =====
## Call:
## phyto = zoop.l1 + phyto.l1 + const + nuts
##
##      zoop.l1  phyto.l1      const      nuts
## -0.822652 0.948470 10.362954 1.009031
```

```
# Stan VAR/ MAR
library(rstan)
stan_data <- list(T = nT, M = nX, P = 1, Y = X, U = U, nU = nU)

model_w_data <- stan(file = 'VAR_p_cov.stan', data=stan_data, iter=200, chains=4, cores=4)

print(model_w_data, pars=c("A", "B", "C"))
```

```
## Inference for Stan model: VAR_p_cov.
## 4 chains, each with iter=200; warmup=100; thin=1;
## post-warmup draws per chain=100, total post-warmup draws=400.
##
##      mean se_mean   sd  2.5%  25%   50%   75%  97.5% n_eff Rhat
## A[1]    0.40    0.03  0.32 -0.25  0.22  0.41  0.63  1.06   92 1.02
## A[2]    9.43    0.04  0.34  8.67  9.21  9.47  9.66 10.05   67 1.07
## B[1,1,1] 0.72    0.00  0.02  0.68  0.71  0.72  0.73  0.77  134 1.01
## B[1,1,2] 0.20    0.00  0.01  0.18  0.19  0.20  0.21  0.22  400 1.01
## B[1,2,1] -0.77   0.00  0.02 -0.81 -0.79 -0.77 -0.75 -0.72   97 1.04
## B[1,2,2] 0.96    0.00  0.01  0.94  0.95  0.96  0.96  0.98  298 1.00
## C[1,1]   -0.01   0.00  0.02 -0.05 -0.02 -0.01  0.01  0.04  400 0.99
## C[2,1]    1.01   0.00  0.03  0.97  1.00  1.02  1.03  1.06  400 1.00
##
## Samples were drawn using NUTS(diag_e) at Fri Dec 11 17:27:55 2015.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

```
traceplot(model_w_data, inc_warmup=FALSE, pars=c("A", "B", "C"))
```

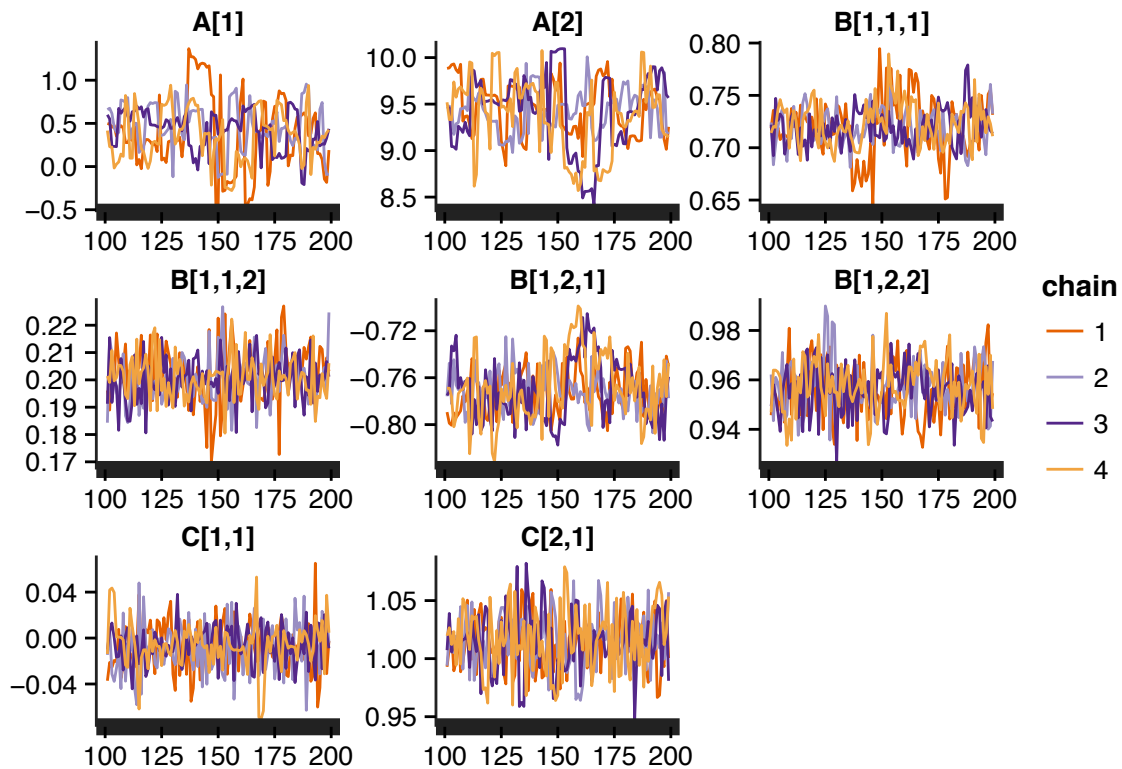


Figure 2:

```
# =====
# = Summary =
# =====
```

## Summary

All three models estimated the coefficients and intercepts ~correctly (`ar` didn't print out the intercept terms, oddly, but it at least got the coefficients pretty close). Both VAR and the Stan model estimated all parameters quite accurately.