

# Appendix S1. Linear regression models and non-linear population dynamics

Jelena H. Pantel\*      Ruben J. Hermann†

22 November, 2024

## 1 One species, logistic growth

Population growth over time in a single species is first modelled using a Beverton-Holt (discrete-time, logistic) model (Beverton and Holt (1957)), using an intra-specific competition coefficient for density-dependent growth (Hart and Marshall (2013)) - thus  $\alpha = 1/K$ .

$$N_{i,t+1} = \frac{r_i N_{i,t}}{1 + \alpha_{ii} N_{i,t}}$$

Note that in this model, the system is at equilibrium when  $N_{i,t+1} = N_{i,t}$ , and therefore:

$$N^* = N^* \frac{r_i}{1 + \alpha_{ii} N^*}$$

$$1 = \frac{r_i}{1 + \alpha_{ii} N^*}$$

$$N^* = \frac{r_i - 1}{\alpha_{ii}}$$

### 1.1 Population dynamics simulation

We simulate population growth using  $r_i = 1.67$  and  $\alpha_{ii} = 0.00125$ , with an initial population size  $N_{1,0} = 10$ . This is implemented using the `disc_log` function from the `ecoevor` package <https://github.com/jhpantel/ecoevoR>.

```
set.seed(42)
# Simulate initial species population growth
N1.0 <- 10
r1.0 <- 1.67
alpha.11 <- 0.00125
# Simulation of model for t time steps
t <- 30
N <- rep(NA, t)
```

\*Laboratoire Chrono-environnement, UMR 6249 CNRS-UFC, 16 Route de Gray, 25030 Besançon cedex, France, [jelena.pantel@univ-fcomte.fr](mailto:jelena.pantel@univ-fcomte.fr)

†University of Duisburg-Essen, Universitätsstraße 5, 45141 Essen, Germany, [ruben.hermann@uni-due.de](mailto:ruben.hermann@uni-due.de)

```

N[1] <- N1.0
for (i in 2:t) {
  N[i] <- disc_log(r = r1.0, N0 = N[i - 1], alpha = alpha.11)
}

```

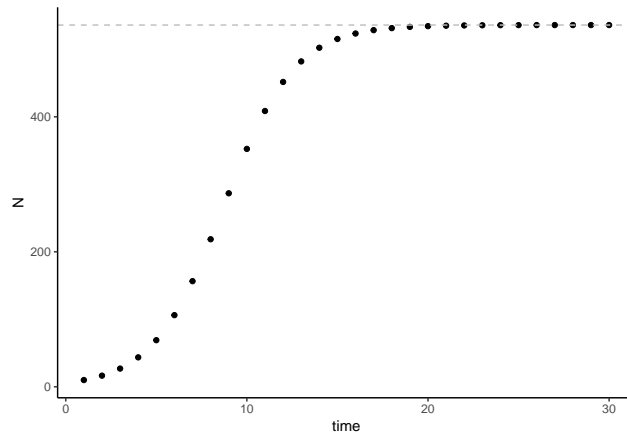


Figure 1: Population size  $N$  over time  $t$  for a discrete-time logistic growth model, with parameters  $r_i = 1.67$ ,  $N_{1,0} = 14$ , and  $\alpha_{11} = 0.00125$ .

## 1.2 Linear statistical model

We fit the population time series data to a first-order auto-regressive model to predict  $N_{t+1}$  as a function of  $N_t$ , and compare that to a linear regression. We use  $\ln$ - $N$  after Ives (1995), as also discussed in Certain et al. (2018) and Olivença et al. (2021).

$$N_{t+1} = \beta_0 + \beta_1 N_t + \epsilon_t$$

```

# Fit the model
m.1.ar <- arima(x = log(N), order = c(1, 0, 0), include.mean = T, method = "CSS")
m.1.lm <- lm(log(dat$N[2:t]) ~ log(dat$N[1:(t - 1)]))
# plotting the series along with the fitted values
m.1.ar.fit <- log(N) - residuals(m.1.ar)
m.1.lm.fit <- log(dat$N[2:t]) - m.1.lm$resid
dat$ar1.fit <- m.1.ar.fit
dat$lm.fit <- NA
dat$lm.fit[2:t] <- m.1.lm.fit

```

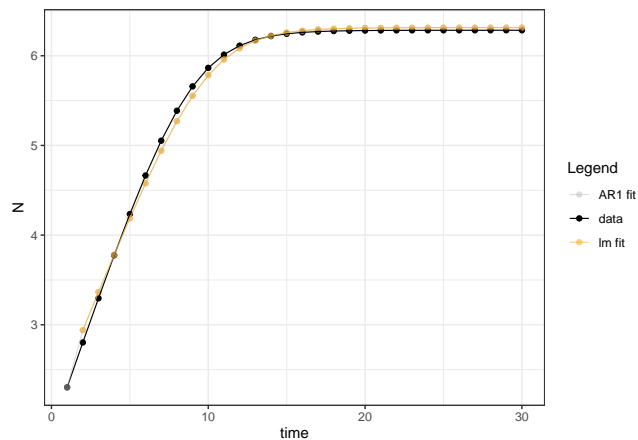


Figure 2: Population size over time (black line) with fitted values from a first-order autoregressive (grey) and linear regression (orange) model.

- 18 The linear model is a good fit, and  $N_{t+1}$  and  $N_t$  are well-represented by a linear function:

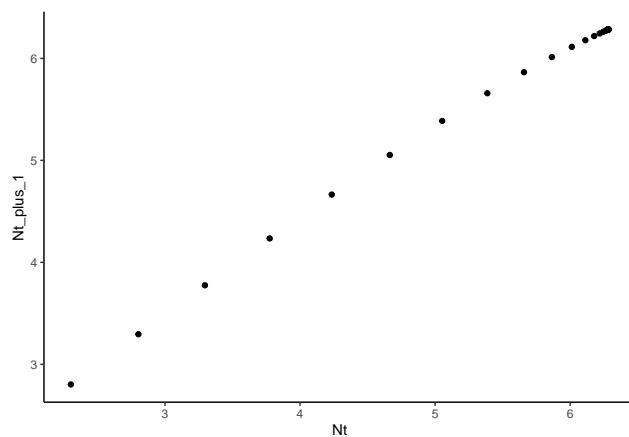


Figure 3: Population size (logarithm) at one time step  $N_{t+1}$  as a function of log-population size in the previous time step  $N_t$ .

- 19 We also examine density dependence by plotting  $\Delta N = N_{t+1} - N_t$  vs.  $N_t$ :

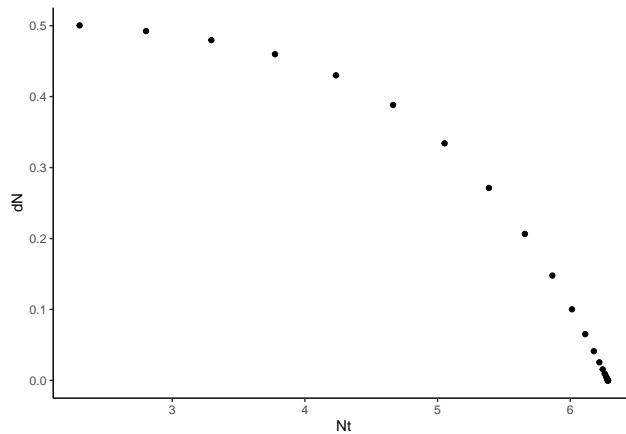


Figure 4: Change in population size from one time step to the next  $\Delta N$  as a function of  $N_{t+1}$

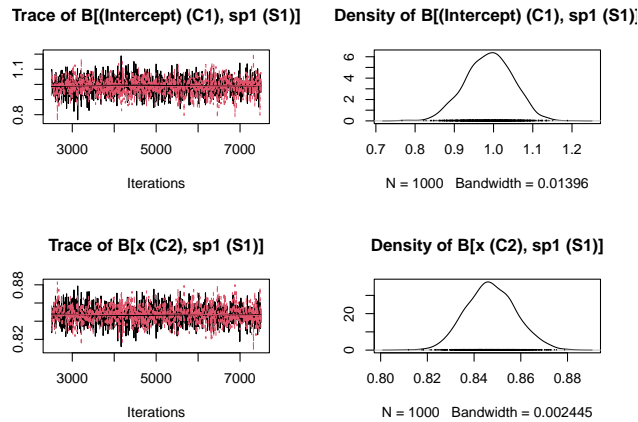
### 1.3 Bayesian linear statistical model: HMSC

We can estimate the same model parameters using HMSC:

```
# prepare data in HMSC format
Y <- as.matrix(log(dat$N[2:t]))
XData <- data.frame(x = log(dat$N[1:(t - 1)]))
m.1.hmsc <- Hmsc(Y = Y, XData = XData, XFormula = ~x)
# Bayesian model parameters
nChains <- 2
thin <- 5
samples <- 1000
transient <- 500 * thin
verbose <- 500 * thin
# sample MCMC
m.1.sample <- sampleMcmc(m.1.hmsc, thin = thin, sample = samples, transient = transient,
  nChains = nChains, verbose = verbose)
```

```
m.post.hmsc <- convertToCodaObject(m.1.sample)
summary(m.post.hmsc$Beta)
#>
#> Iterations = 2505:7500
#> Thinning interval = 5
#> Number of chains = 2
#> Sample size per chain = 1000
#>
#> 1. Empirical mean and standard deviation for each variable,
#>    plus standard error of the mean:
#>
#>
#>               Mean      SD Naive SE Time-series SE
#> B[(Intercept) (C1), sp1 (S1)] 0.9887 0.06022 0.0013466      0.0013461
#> B[x (C2), sp1 (S1)]          0.8473 0.01055 0.0002358      0.0002421
#>
#> 2. Quantiles for each variable:
#>
#>
#>               2.5%    25%    50%    75%    97.5%
```

```
#> B[(Intercept) (C1), sp1 (S1)] 0.8711 0.9476 0.990 1.0309 1.0983
#> B[x (C2), sp1 (S1)]          0.8274 0.8399 0.847 0.8543 0.8686
plot(m.post.hmsc$Beta)
```



22

23 These estimates match well with those from the AR1 and linear model:

```
# AR1 coefficients (recall that the intercept is the term below multiplied by 1
# - phi1)
m.1.ar$coef
#>          ar1 intercept
#> 0.8473883 6.4763916
m.1.ar$coef[2] * (1 - m.1.ar$coef[1])
#> intercept
#> 0.9883733
# linear model
summary(m.1.lm)$coefficients[1:2, 1:2]
#>              Estimate Std. Error
#> (Intercept)    0.9883734 0.053842285
#> log(dat$N[1:(t - 1)]) 0.8473883 0.009454972
# Bayesian estimates
summary(m.post.hmsc$Beta)$statistics[1:2, 1:2]
#>              Mean          SD
#> B[(Intercept) (C1), sp1 (S1)] 0.9887325 0.06022001
#> B[x (C2), sp1 (S1)]          0.8472922 0.01054677
```

```
Gradient <- constructGradient(m.1.sample, focalVariable = "x", ngrid = 29)
predY <- predict(m.1.sample, Gradient = Gradient, expected = TRUE)
plotGradient(m.1.sample, Gradient, pred = predY, showData = T, measure = "Y", main = "",
  xlab = "N_t", ylab = "predicted N_t+1")
```

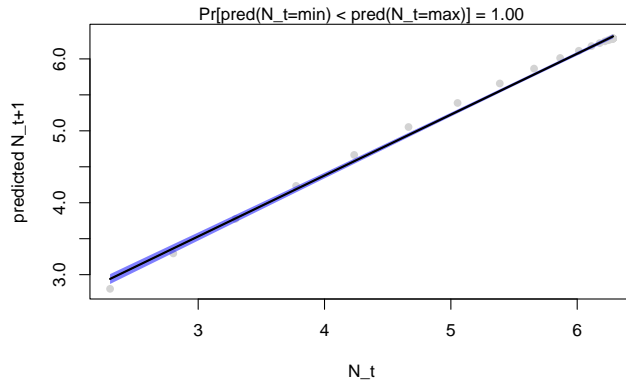


Figure 5: Observed (grey) and model-fit (blue) values for population size at time  $t$  (x-axis) and  $t+1$  (y-axis) using HMSC.

```
lm_dat <- data.frame(cbind(log(dat$N[2:t]), log(dat$N[1:(t - 1)])))
colnames(lm_dat) <- c("Nt1", "Nt")
ggplot(lm_dat, aes(Nt, Nt1)) + stat_summary(fun.data = mean_cl_normal) + geom_smooth(method = "lm") +
  theme_classic()
```

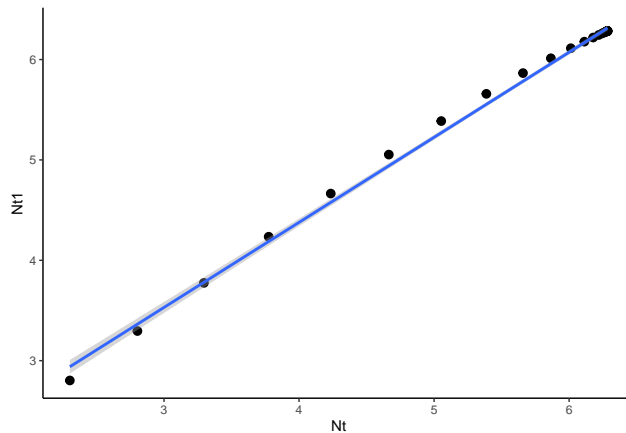


Figure 6: Observed (grey) and model-fit (blue) values for population size at time  $t$  (x-axis) and  $t+1$  (y-axis) using lm.

## 1.4 Conclusions

In this example, a first-order auto-regressive model works well, bypassing the need to estimate logistic growth parameters  $r_i$  and  $\alpha_{ii}$ . The density-dependence dynamics ( $\Delta N \sim f(N_t)$ ) show an overall declining trend over time. The Bayesian estimation implemented in HMSC gives good parameter estimates.

## 2 One species, logistic growth, environmental covariate

We now consider using a linear model to analyze population growth when the species growth rate is impacted by a single environmental covariate.

## 2.1 Growth depends on environment

First we add environment-dependent growth rate. The growth rate  $r_i$  becomes:

$$r_i = \hat{W} e^{-(E-x_{i,t})^2}$$

Here,  $\hat{W}$  is the maximal population growth rate (set to 1.67 as above),  $E$  is the local environmental trait optimum value, and  $x_{i,t}$  is species  $i$  trait value at time  $t$ . We see that if  $E = x_{i,t}$  then the growth rate is at the value  $r = 1.67$ . Here, we begin with  $E = x_{i,t} = 0.8$ , then simulate the environment  $E$  value fluctuating randomly over time, and finally use a linear model to fit  $E$  as a covariate. We implement the simulation using the `disc_log_E` function from the `ecoevor` package.

```
# Simulate initial species population growth with environment fluctuations
N1.0 <- 10
r1.0 <- 1.67
alpha.11 <- 0.00125
E.0 <- 0.8
x1.0 <- 0.8
# Simulation of model for t time steps
t <- 40
N <- rep(NA, t)
N[1] <- N1.0
E <- rep(NA, t)
E[1] <- E.0
for (i in 2:t) {
  N[i] <- disc_log_E(r = r1.0, NO = N[i - 1], alpha = alpha.11, E = E[i - 1], x = x1.0)
  E[i] <- E[i - 1] + rnorm(1, 0, 0.1)
}
```

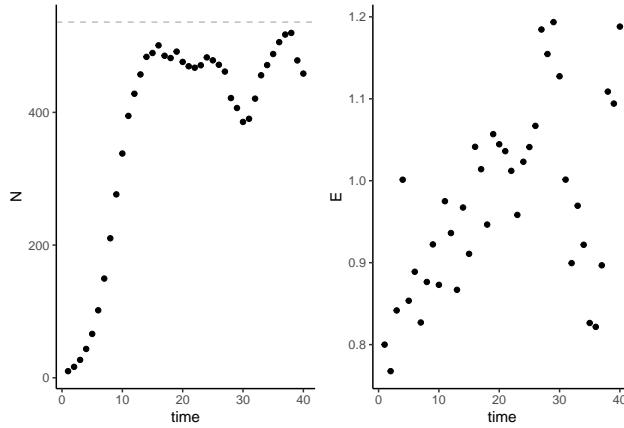


Figure 7: Population size  $N$  over time  $t$  for a discrete-time logistic growth model, with parameters  $r_i = 1.67$ ,  $N_{1,0} = 14$ , and  $\alpha_{11} = 0.00125$ . The value of  $E$  over time is also shown.

## 2.2 Linear statistical model with environmental covariate

We now include environment  $E$  as a covariate in the linear model:

$$N_t = \beta_0 + \beta_1 N_{t-1} + \beta_2 E_{t-1} + \epsilon_t$$

```

# Fit the model
m.2.ar <- arima(x = log(N), order = c(1, 0, 0), include.mean = T, method = "CSS",
  xreg = E)
m.2.lm <- lm(log(dat$N[2:t]) ~ log(dat$N[1:(t - 1)]) + log(E[1:(t - 1)]))
# plotting the series along with the fitted values
m.2.ar.fit <- log(N) - residuals(m.2.ar)
m.2.lm.fit <- log(dat$N[2:t]) - m.2.lm$resid
dat$ar2.fit <- m.2.ar.fit
dat$lm2.fit <- NA
dat$lm2.fit[2:t] <- m.2.lm.fit

```

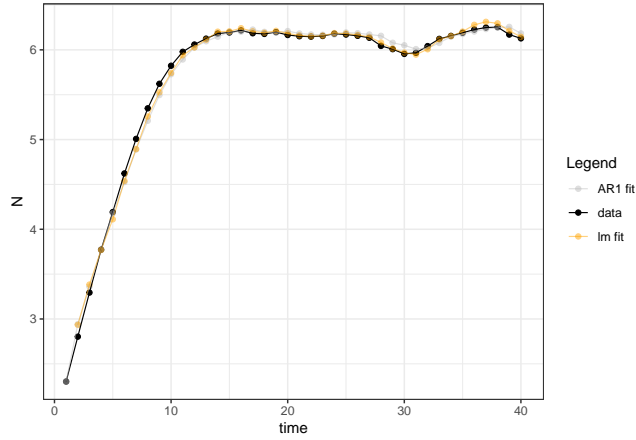


Figure 8: Population size over time (black line) with fitted values from a first-order autoregressive model (gray) and a linear model (orange).

40 The linear model is a good fit when including the environmental covariate.  $N_{t+1}$  and  $N_t$  can still be captured  
 41 by a linear relationship. However we see that the relationship between  $N_{t+1}$  and  $E_t$  is non-linear.

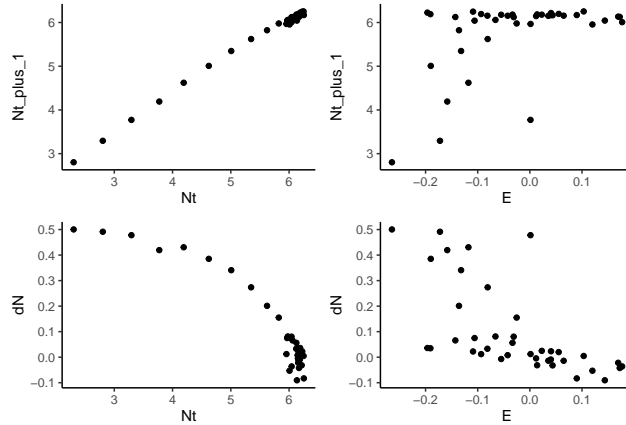


Figure 9: Population size (logarithm) at one time step  $N_{t+1}$  as a function of log-population size in the previous time step  $N_t$ . Also shown are change in population size  $\Delta N$  with population size and  $E$ .

42 This tells us that the lm is good for predictions, but not for inference (for capturing well the relationship  
 43 between the predictor and response variable). The use of linear relationships in JSDMs is discussed in  
 44 (Ingram et al. 2020), and in many applications (e.g. (Erickson and Smith 2023)) quadratic terms are used,



45 which create bell-shaped response curves that may better match species with optimal niches (as opposed to  
 46 linear, monotonically increasing relationships between population size and environmental predictors). We  
 47 thus include a quadratic term for  $E_t$  to provide a better fit to the data.

```
df <- data.frame(cbind(log(dat$N[2:t]), log(dat$N[1:(t - 1)]), E[1:(t - 1)], E[1:(t - 1)]^2))
colnames(df) <- c("Nt1", "Nt", "E", "Esq")
m.2.lm <- lm(Nt1 ~ Nt + E + Esq, data = df)
```

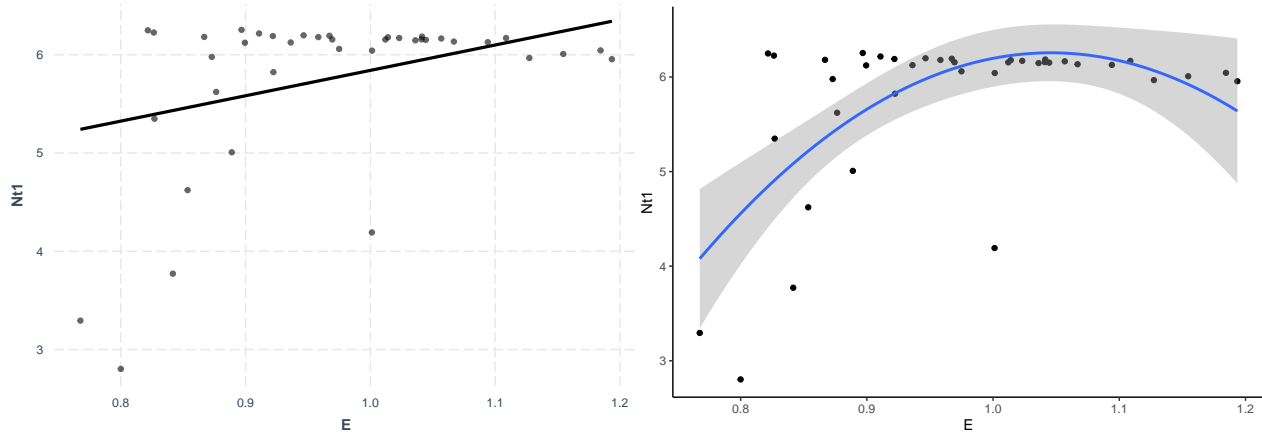


Figure 10: Population size (logarithm) at one time step  $N_{t+1}$  as a function of environment  $E$ . Linear and quadratic fits are shown.

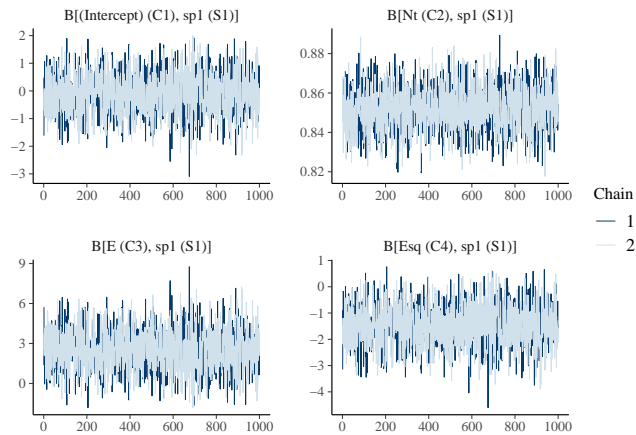
## 48 2.3 Bayesian linear statistical model: HMSC

49 We can estimate the same model parameters using HMSC:

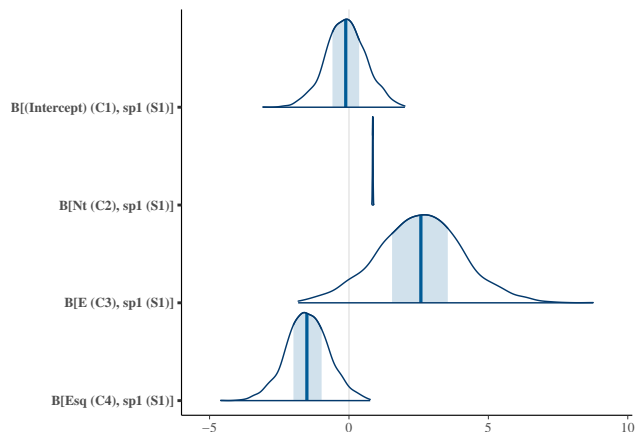
```
# prepare data in HMSC format
Y <- as.matrix(log(dat$N[2:t]))
XData <- df
m.2.hmsc <- Hmsc(Y = Y, XData = XData, XFormula = ~Nt + E + Esq)
# Bayesian model parameters
nChains <- 2
thin <- 5
samples <- 1000
transient <- 500 * thin
verbose <- 500 * thin
# sample MCMC
m.2.sample <- sampleMcmc(m.2.hmsc, thin = thin, sample = samples, transient = transient,
  nChains = nChains, verbose = verbose)
```

```
m2.post.hmsc <- convertToCodaObject(m.2.sample)
summary(m2.post.hmsc$Beta)
#>
#> Iterations = 2505:7500
#> Thinning interval = 5
#> Number of chains = 2
#> Sample size per chain = 1000
```

```
#>
#> 1. Empirical mean and standard deviation for each variable,
#>    plus standard error of the mean:
#>
#>               Mean      SD Naive SE Time-series SE
#> B[(Intercept) (C1), sp1 (S1)] -0.1172 0.72629 0.0162404      0.0162433
#> B[Nt (C2), sp1 (S1)]          0.8521 0.01108 0.0002477      0.0002143
#> B[E (C3), sp1 (S1)]           2.5496 1.53081 0.0342299      0.0342367
#> B[Esq (C4), sp1 (S1)]         -1.4924 0.76626 0.0171341      0.0171372
#>
#> 2. Quantiles for each variable:
#>
#>               2.5%      25%      50%      75%      97.5%
#> B[(Intercept) (C1), sp1 (S1)] -1.5785 -0.5918 -0.1164  0.3659 1.33237
#> B[Nt (C2), sp1 (S1)]           0.8302  0.8448  0.8520  0.8594 0.87430
#> B[E (C3), sp1 (S1)]           -0.4942  1.5454  2.5787  3.5439 5.62748
#> B[Esq (C4), sp1 (S1)]         -3.0327 -1.9918 -1.5099 -0.9818 0.06678
bayesplot::mcmc_trace(m2.post.hmsc$Beta)
```



```
bayesplot::mcmc_areas(m2.post.hmsc$Beta, area_method = c("equal height"))
```



These estimates match well with those from the AR1 and linear model:

```

# AR1 coefficients (recall that the intercept is the term below multiplied by 1
# - phi1)
m.2.ar$coef
#>      ar1      intercept      E
#> 0.83925564 6.29776167 -0.02674129
m.2.ar$coef[2] * (1 - m.2.ar$coef[1])
#> intercept
#> 1.01233
# linear model
summary(m.2.lm)$coefficients[1:4, 1:2]
#>      Estimate Std. Error
#> (Intercept) -0.1317811 0.63418671
#> Nt          0.8523506 0.01002202
#> E           2.5819242 1.33687286
#> Esq         -1.5113693 0.66907748
# Bayesian estimates
summary(m2.post.hmsc$Beta)$statistics[1:4, 1:2]
#>      Mean      SD
#> B[(Intercept) (C1), sp1 (S1)] -0.1172053 0.72629078
#> B[Nt (C2), sp1 (S1)]          0.8521043 0.01107945
#> B[E (C3), sp1 (S1)]           2.5496453 1.53080925
#> B[Esq (C4), sp1 (S1)]        -1.4923591 0.76625968

knitr::knit_exit()

```

- 53 Beverton, R. J., and S. J. Holt. 1957. On the dynamics of exploited fish populations (Vol. 11). Springer  
54 Science & Business Media.
- 55 Certain, G., F. Barraquand, and A. Gårdmark. 2018. [How do MAR\(1\) models cope with hidden nonlinear-](#)  
56 [ities in ecological dynamics?](#) *Methods in Ecology and Evolution* 9:1975–1995.
- 57 Erickson, K. D., and A. B. Smith. 2023. [Modeling the rarest of the rare: A comparison between multi-](#)  
58 [species distribution models, ensembles of small models, and single-species models at extremely low sample](#)  
59 [sizes.](#) *Ecography* 2023:e06500.
- 60 Hart, S. P., and D. J. Marshall. 2013. [Environmental stress, facilitation, competition, and coexistence.](#)  
61 *Ecology* 94:2719–2731.
- 62 Ingram, M., D. Vukcevic, and N. Golding. 2020. [Multi-output gaussian processes for species distribution](#)  
63 [modelling.](#) *Methods in Ecology and Evolution* 11:1587–1598.
- 64 Ives, A. R. 1995. [Predicting the response of populations to environmental change.](#) *Ecology* 76:926–941.
- 65 Olivença, D. V., J. D. Davis, and E. O. Voit. 2021. [Comparison between lotka-volterra and multivariate](#)  
66 [autoregressive models of ecological interaction systems.](#) *bioRxiv*.