

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

Jelena H. Pantel\*      Ruben J. Hermann†

25 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)
}

```

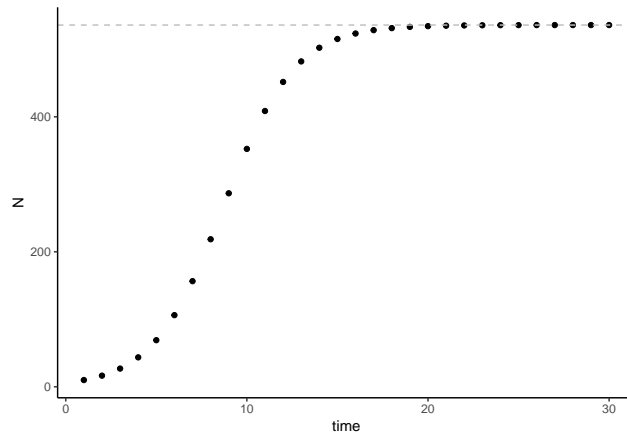


Figure 1: Population size  $N$  over time  $t$  for a discrete-time logistic growth model, with parameters  $r_i = 1.67$ ,  $N_{1,0} = 14$ , and  $\alpha_{11} = 0.00125$ .

## 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

```

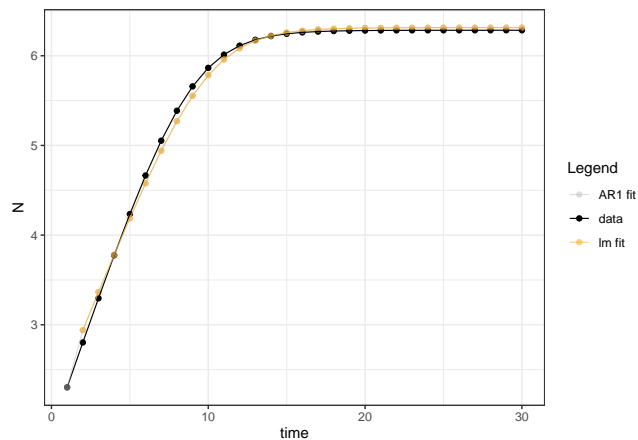


Figure 2: Population size over time (black line) with fitted values from a first-order autoregressive (grey) and linear regression (orange) model.

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

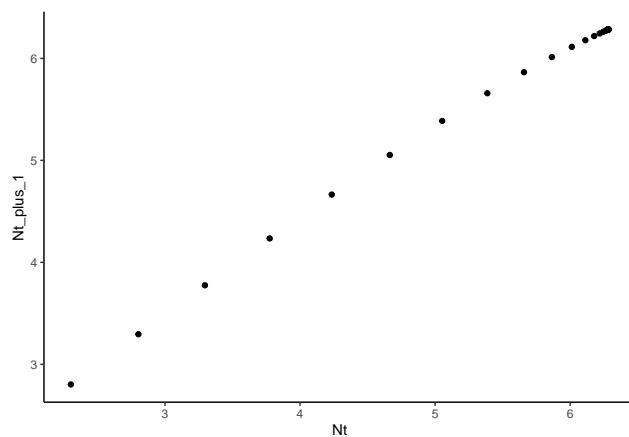


Figure 3: Population size (logarithm) at one time step  $N_{t+1}$  as a function of log-population size in the previous time step  $N_t$ .

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

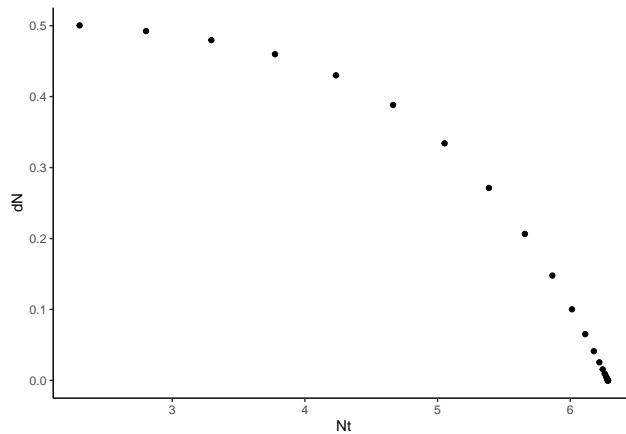


Figure 4: Change in population size from one time step to the next  $\Delta N$  as a function of  $N_{t+1}$

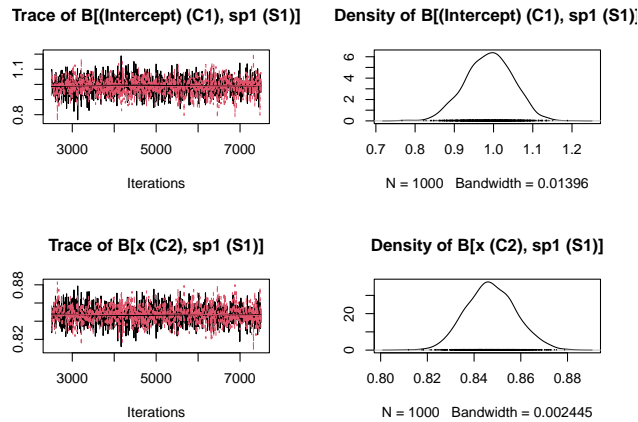
### 1.3 Bayesian linear statistical model: HMSC

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")
```

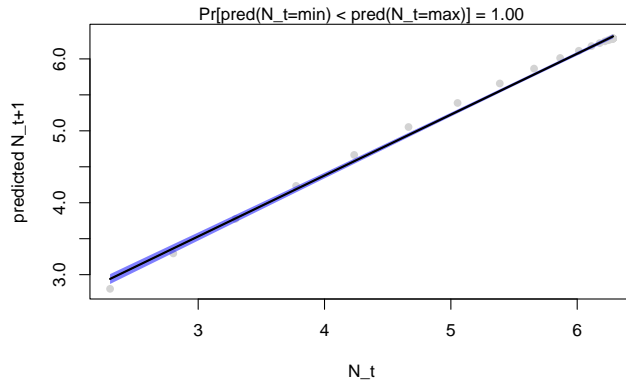


Figure 5: Observed (grey) and model-fit (blue) values for population size at time  $t$  (x-axis) and  $t+1$  (y-axis) using HMSC.

```
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()
```

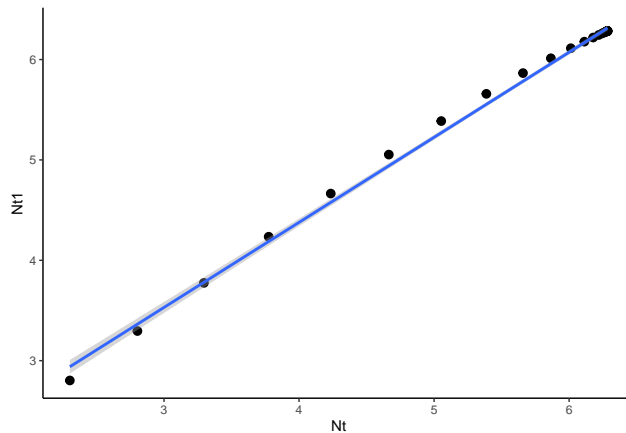


Figure 6: Observed (grey) and model-fit (blue) values for population size at time  $t$  (x-axis) and  $t+1$  (y-axis) using lm.

## 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)
}
```

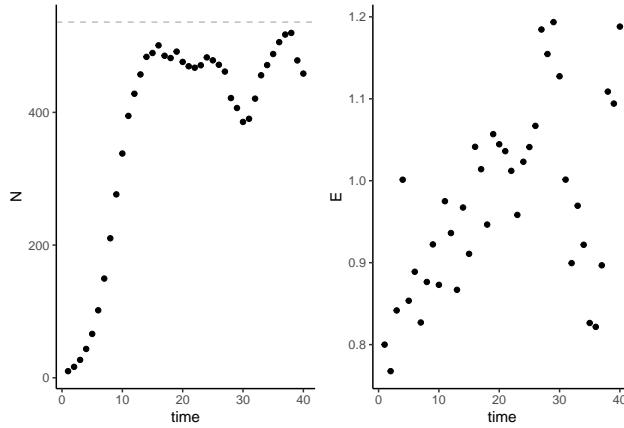


Figure 7: Population size  $N$  over time  $t$  for a discrete-time logistic growth model, with parameters  $r_i = 1.67$ ,  $N_{1,0} = 14$ , and  $\alpha_{11} = 0.00125$ . The value of  $E$  over time is also shown.

## 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

```

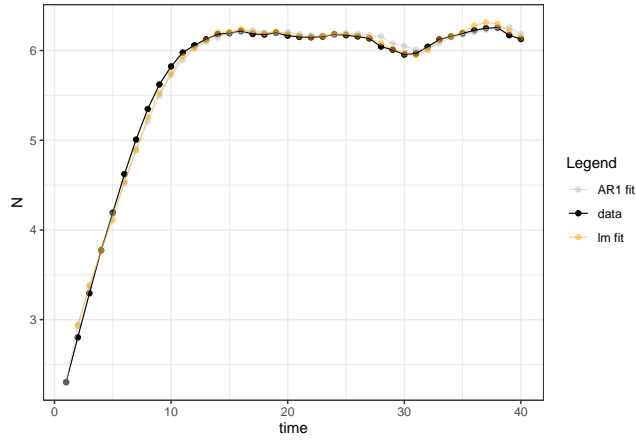


Figure 8: Population size over time (black line) with fitted values from a first-order autoregressive model (gray) and a linear model (orange).

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.

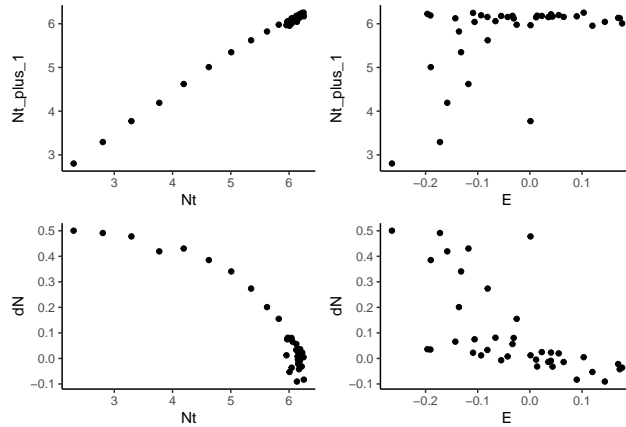


Figure 9: Population size (logarithm) at one time step  $N_{t+1}$  as a function of log-population size in the previous time step  $N_t$  and of environmental value  $E_t$ . Also shown are change in population size  $\Delta N$  with population size and  $E$ .

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



(Ingram et al. 2020), and in many applications (e.g. (Erickson and Smith 2023)) quadratic terms are used, which create bell-shaped response curves that may better match species with optimal niches (as opposed to linear, monotonically increasing relationships between population size and environmental predictors). We 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)
```

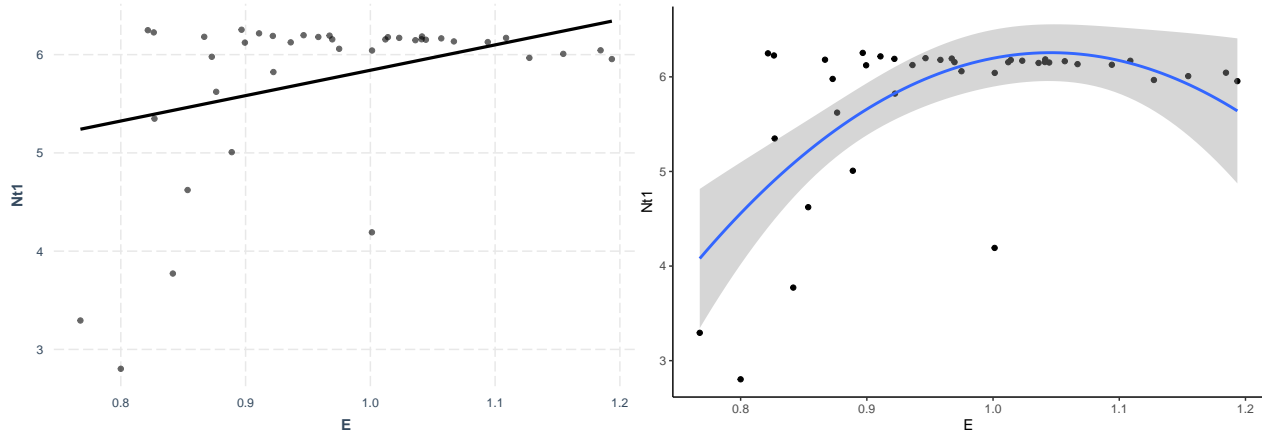


Figure 10: Population size (logarithm) at one time step  $N_{t+1}$  as a function of environment  $E$ . Linear and quadratic fits are shown.

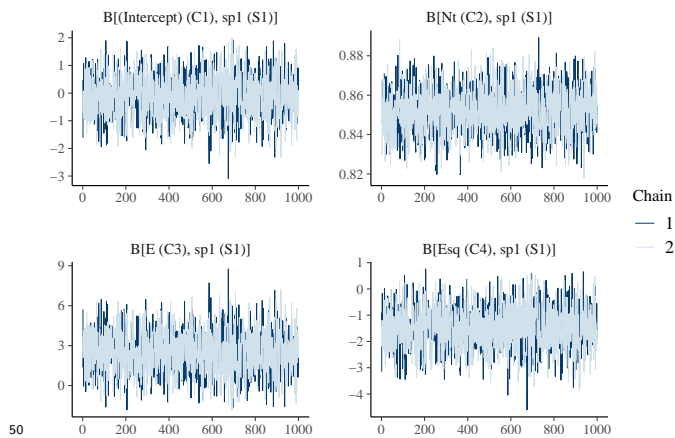
## 2.3 Bayesian linear statistical model: HMSC

We can estimate the same model parameters using HMSC:

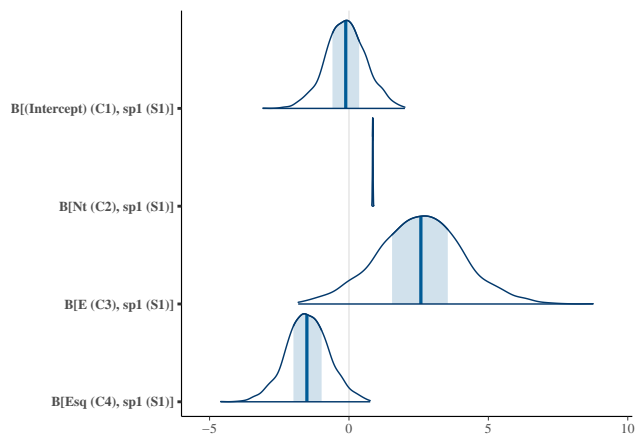
```
# prepare data in HMSC format
Y <- as.matrix(log(dat$N[2:t]))
XData <- df
m.2.hmsc <- Hmsc(Y = Y, XData = XData, XFormula = ~Nt + E + Esq)
# Bayesian model parameters
nChains <- 2
thin <- 5
samples <- 1000
transient <- 500 * thin
verbose <- 500 * thin
# sample MCMC
m.2.sample <- sampleMcmc(m.2.hmsc,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

```

53 We recall that the interpretation of the coefficients in an arimaX (arima with covariates) model is difficult.  
 54 They do not give the impact on  $N_t$  per unit increase in  $X$  as in a regression. So we do not interpret the  
 55 causation implied by the coefficient in the arimaX model. In the regression model, we can see that  $E$  has a  
 56 positive impact on  $N_t$ .

```

Gradient <- constructGradient(m.2.sample, focalVariable = "E", non.focalVariables = list(Nt = list(2),
  Esq = list(2)), ngrid = 39)
# Esq is manually constructed as gradient-produced E^2
Gradient$XDataNew$Esq <- Gradient$XDataNew$E^2
predY <- predict(m.2.sample, XData = Gradient$XDataNew, expected = TRUE)
plotGradient(m.2.sample, Gradient, pred = predY, showData = T, measure = "Y", main = "",
  xlab = "E_t", ylab = "predicted N_t+1")

```

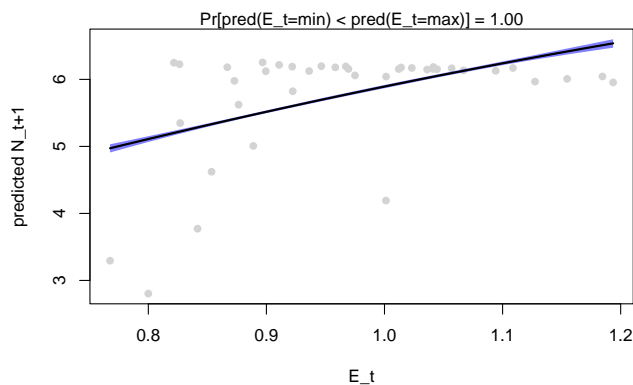


Figure 11: Observed (grey) and model-fit (blue) values for population size at time  $t+1$  (y-axis) as a function of environment  $E_t$  (x-axis).

## 2.4 Conclusions

In this example, the linear regression again works well to describe the impact of  $E_t$  for  $N_t$  when using the quadratic formulation. The arimaX model works well for fitting and subsequent prediction, but less well for inference about the impacts of  $E$ . From the quadratic regression terms for  $E$ , we correctly see that the population size is maximal at the species trait value and decreases away from that value. We will continue to use log-transformed abundance and now introduce quadratic terms for the environmental parameter.

## 3 Two species, logistic growth, competition

Here we investigate how a second species impacts the inference we can make from linear models.

### 3.1 Interspecific competition

The growth equation for each species now becomes:

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

We use a distinct growth rate for the 2nd species and introduce the interspecific interaction coefficient  $\alpha_{ij}$ .

```
# Simulate initial species population growth with a second species present
N1.0 <- 10
N2.0 <- 10
r1.0 <- 1.67
r2.0 <- 1.7
alpha.11 <- 0.00125
alpha.22 <- 0.00125
alpha.12 <- 0.008
alpha.21 <- 0.008725
# model function
disc_LV_comp <- function(r1, r2, N1.0, N2.0, alpha.11, alpha.22, alpha.12, alpha.21) {
  Nt1 <- (r1 * N1.0)/(1 + alpha.11 * N1.0 + alpha.12 * N2.0)
  Nt2 <- (r2 * N2.0)/(1 + alpha.22 * N2.0 + alpha.21 * N1.0)
  return(c(Nt1, Nt2))
}
# Simulation of model for t time steps
t <- 40
N <- array(NA, dim = c(t, 2))
N <- as.data.frame(N)
colnames(N) <- c("N1", "N2")
N$N1[1] <- N1.0
N$N2[1] <- N2.0
for (i in 2:t) {
  res <- disc_LV_comp(r1 = r1.0, r2 = r2.0, N1.0 = N[i - 1, 1], N2.0 = N[i - 1,
    2], alpha.11 = alpha.11, alpha.22 = alpha.22, alpha.12 = alpha.12, alpha.21 = alpha.21)
  N$N1[i] <- res[1]
  N$N2[i] <- res[2]
}
```

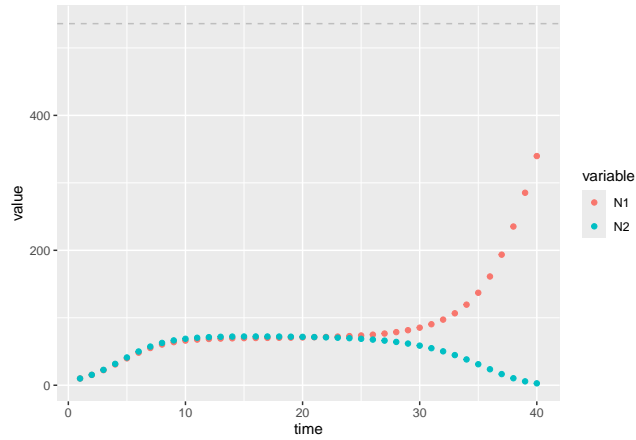


Figure 12: Population size  $N$  over time  $t$  for a discrete-time logistic growth model, with parameters  $r_1 = 1.67$ ,  $r_2 = 1.7$ ,  $N_{i,0} = 10$ ,  $\alpha_{ii} = 0.00125$ ,  $\alpha_{12} = 0.008$ , and  $\alpha_{21} = 0.008725$ .

### 3.2 Linear statistical model with covariate for both species

We fit each species' population time series to an arimaX and lm with population size of the others species as a covariate.

$$N_{1,t} = \beta_0 + \beta_1 N_{1,t-1} + \beta_2 N_{2,t-1} + \epsilon_t$$

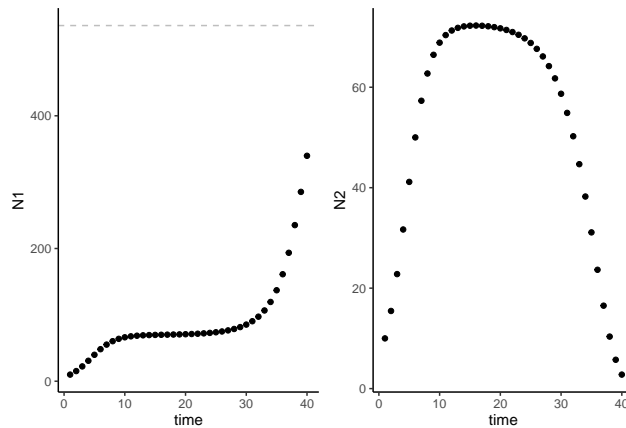


Figure 13: Population size  $N$  over time  $t$  for a discrete-time logistic growth model, with competition between 2 species ( $N_1$  and  $N_2$ ).

```
## Fit the model for Species 1
m.3.ar.n1 <- arima(x = log(N$N1), order = c(1, 0, 0), include.mean = T, method = "CSS",
  xreg = N$N2)
m.3.lm.n1 <- lm(log(dat$N1[2:t]) ~ log(dat$N1[1:(t - 1)]) + log(dat$N2[1:(t - 1)]))
# plotting the series along with the fitted values
m.3.ar.fit.n1 <- log(N$N1) - residuals(m.3.ar.n1)
m.3.lm.fit.n1 <- log(dat$N1[2:t]) - m.3.lm.n1$resid
dat$ar3.fit.n1 <- m.3.ar.fit.n1
dat$lm3.fit.n1 <- NA
```

```

dat$lm3.fit.n1[2:t] <- m.3.lm.fit.n1
## Species 2
m.3.ar.n2 <- arima(x = log(N$N2), order = c(1, 0, 0), include.mean = T, method = "CSS",
  xreg = N$N1)
m.3.lm.n2 <- lm(log(dat$N2[2:t]) ~ log(dat$N2[1:(t - 1)]) + log(dat$N1[1:(t - 1)]))
# plotting the series along with the fitted values
m.3.ar.fit.n2 <- log(N$N2) - residuals(m.3.ar.n2)
m.3.lm.fit.n2 <- log(dat$N2[2:t]) - m.3.lm.n2$resid
dat$ar3.fit.n2 <- m.3.ar.fit.n2
dat$lm3.fit.n2 <- NA
dat$lm3.fit.n2[2:t] <- m.3.lm.fit.n2

```

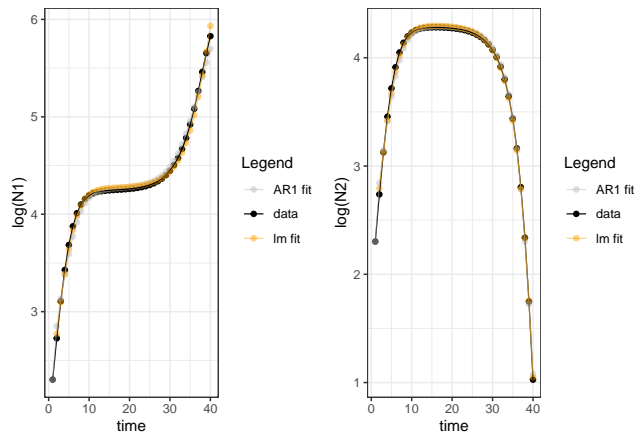


Figure 14: Population size over time (black line) with fitted values from a first-order autoregressive model (gray line) and from a linear model (orange line).

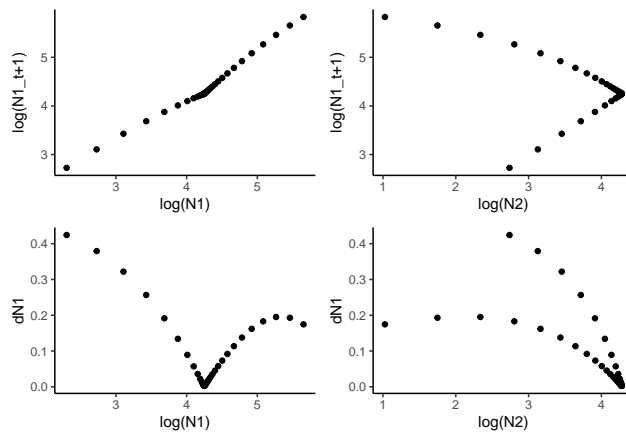


Figure 15: Population size (logarithm) in species 1 at one time step  $N_{1,t+1}$  as a function of log-population size of Species 1 and Species 2 in the previous time step  $N_t$ . Also shown is the change in population size of Species 1 as a function of  $N_t$  for both species.

71 We can see that the population size of Species 2 does not have a linear relationship with that of Species 1.

72 #> [1] 0.9845708

```

73 #>
74 #> Call:
75 #> lm(formula = Nt1 ~ I(1/N1) + I(1/N2), data = df)
76 #>
77 #> Residuals:
78 #>      Min       1Q   Median       3Q      Max
79 #> -0.30952 -0.04467 -0.03442  0.03430  0.22447
80 #>
81 #> Coefficients:
82 #>              Estimate Std. Error t value Pr(>|t|)
83 #> (Intercept)   6.4056     0.1060   60.45 < 2e-16 ***
84 #> I(1/N1)    -12.2863     0.3774  -32.55 < 2e-16 ***
85 #> I(1/N2)     3.3309     0.2448   13.61 9.24e-16 ***
86 #> ---
87 #> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
88 #>
89 #> Residual standard error: 0.1063 on 36 degrees of freedom
90 #> Multiple R-squared:  0.9694, Adjusted R-squared:  0.9677
91 #> F-statistic: 569.8 on 2 and 36 DF,  p-value: < 2.2e-16

```

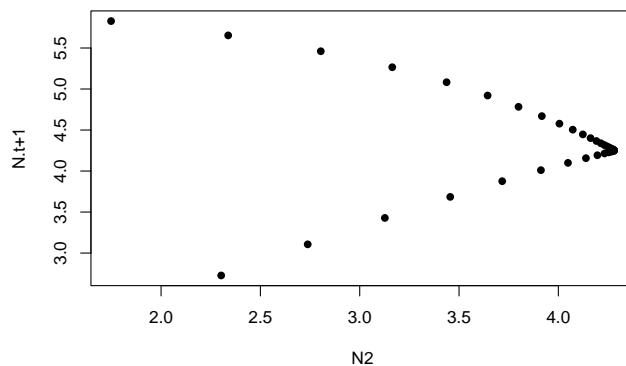


Figure 16: Population size (logarithm) in species 1 at one time step  $N_{1,t+1}$  as a function of log-population size of Species 2 in the previous time step  $N_t$ .

```

92 #> Using data df from global environment. This could cause incorrect results
93 #> if df has been altered since the model was fit. You can manually provide
94 #> the data to the "data =" argument.

```

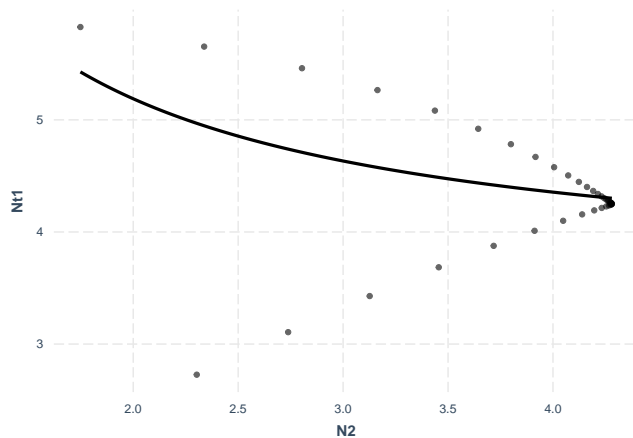


Figure 17: Linear model-predicted population size (logarithm) at one time step  $N_{t+1}$  as a function of log-population size of the second species in the previous time step  $N_t$ . Fitted (line) and observed (points) values are shown.

We can already see that the linear model does not provide a useful fit to the data points (neither does taking the inverse function help). The regression parameters can predict well the population size of the other species, but they are not informative for inference. This has been observed previously (Certain et al. (2018)), and non-linear least squares models (among others) are more often used to estimate parameters for dynamics resulting from these kinds of species interaction models (Kloppers and Greeff (2013); Mühlbauer et al. (2020); Olivença et al. (2021)). The inability of the linear models to reproduce the interaction coefficients is more clearly demonstrated in Certain et al. (2018). However, they do indicate that the slope of the linear model can reflect the direction of effect for the interacting species.

Instead of further exploring techniques to fit to time-series derived from non-linear competition processes, we look more closely at how HMSC manages to fit data emerging from competition dynamics and make inference about species interactions. Instead of estimating full interaction coefficients, or assuming sparse interactions, they instead assume that a reduced number of linear combinations of species abundances that are most relevant to determining future growth rates for species in the community can bypass the ‘curse of dimensionality’ problem. This is well presented in Ovaskainen et al. (2017). We apply that approach here.

### 3.3 Bayesian linear statistical model: HMSC

We can fit the data to a linear model using HMSC, with latent variables to capture the species associations. These are implemented by including a random effect at the time level. For this analysis, we do not expect the impact of Species 1 and 2 that results from competition to vary over time - competition will always have negative impacts for species abundances. Our goal is to estimate the overall impact of species covariance across all of the time points. We do not estimate these impacts as a function of the lag in time points. In other words, we assume that each time point has random variance and that species may covary in their response at each time point. We do not consider that species covariance is a function of distance between sampled time points (see Ovaskainen and Abrego (2020), equation 5.9). We instead assume that site loadings  $\eta$  are independent among the sampled time points.

```
# prepare data in HMSC format
Y <- as.matrix(cbind(log(dat$N1[2:t]), log(dat$N2[2:t])))
XData <- df[, 2:3]
studyDesign = data.frame(sample = as.factor(1:(t - 1)))
studyDesign$sample <- as.factor(studyDesign$sample)
rL.sample = HmscRandomLevel(units = studyDesign$sample)
m.3.hmsc = Hmsc(Y = Y, XData = XData, XFormula = ~1, studyDesign = studyDesign, ranLevels = list(sample
```

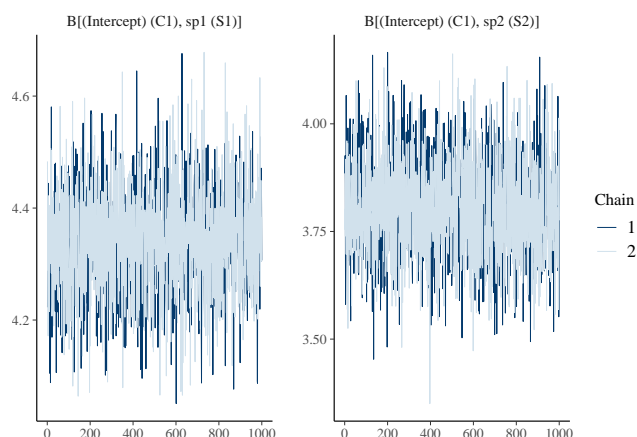


```

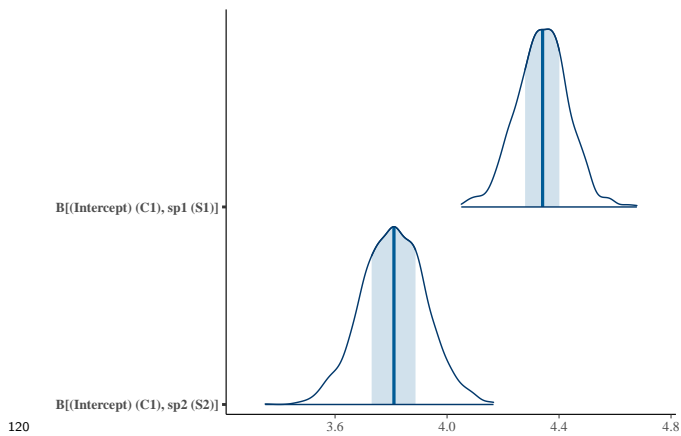
# Bayesian model parameters
nChains <- 2
thin <- 5
samples <- 1000
transient <- 500 * thin
verbose <- 500 * thin
# sample MCMC
m.3.sample <- sampleMcmc(m.3.hmsc, thin = thin, sample = samples, transient = transient,
  nChains = nChains, verbose = verbose)
#> Computing chain 1
#> Chain 1, iteration 2500 of 7500 (transient)
#> Chain 1, iteration 5000 of 7500 (sampling)
#> Chain 1, iteration 7500 of 7500 (sampling)
#> Computing chain 2
#> Chain 2, iteration 2500 of 7500 (transient)
#> Chain 2, iteration 5000 of 7500 (sampling)
#> Chain 2, iteration 7500 of 7500 (sampling)

m3.post.hmsc <- convertToCodaObject(m.3.sample)
summary(m3.post.hmsc$Beta)
#>
#> Iterations = 2505:7500
#> Thinning interval = 5
#> Number of chains = 2
#> Sample size per chain = 1000
#>
#> 1. Empirical mean and standard deviation for each variable,
#>    plus standard error of the mean:
#>
#>               Mean      SD Naive SE Time-series SE
#> B[(Intercept) (C1), sp1 (S1)] 4.34 0.09341 0.002089      0.002039
#> B[(Intercept) (C1), sp2 (S2)] 3.81 0.11431 0.002556      0.002555
#>
#> 2. Quantiles for each variable:
#>
#>               2.5%  25%   50%   75% 97.5%
#> B[(Intercept) (C1), sp1 (S1)] 4.154 4.279 4.341 4.401 4.511
#> B[(Intercept) (C1), sp2 (S2)] 3.581 3.731 3.810 3.887 4.035
bayesplot::mcmc_trace(m3.post.hmsc$Beta)

```

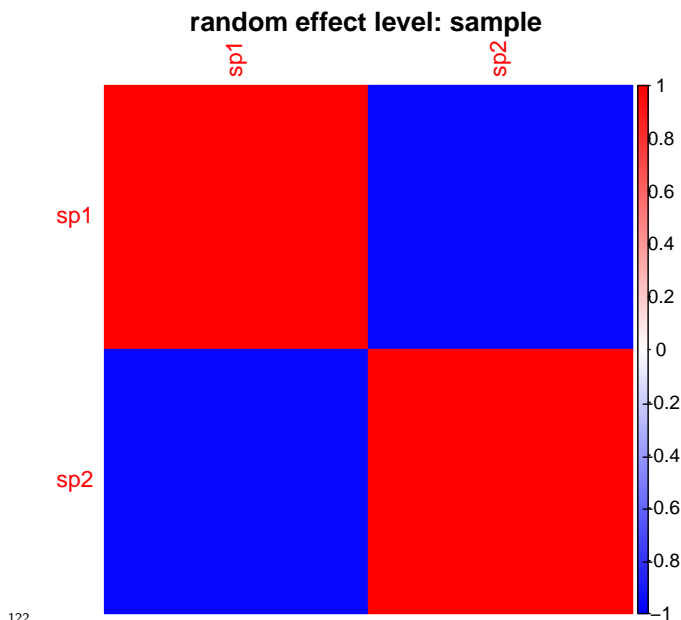


```
bayesplot::mcmc_areas(m3.post.hmsc$Beta, area_method = c("equal height"))
```



Now we look at the estimates for the species associations:

```
OmegaCor = computeAssociations(m.3.sample)
supportLevel = 0.95
toPlot = ((OmegaCor[[1]]$support > supportLevel) + (OmegaCor[[1]]$support < (1 -
  supportLevel)) > 0) * OmegaCor[[1]]$mean
corrplot::corrplot(toPlot, method = "color", col = colorRampPalette(c("blue", "white",
  "red"))(200), title = paste("random effect level:", m.3.sample$rLNames[1]), mar = c(0,
  0, 1, 0))
```



The model fixed effect 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.3.ar.n1$coef
#>      ar1      intercept      N$N2
#> 0.79696422  5.69840619 -0.01881979
```

```

m.3.ar.n1$coef[2] * (1 - m.3.ar.n1$coef[1])
#> intercept
#> 1.15698
# linear model
summary(m.3.lm.n1)$coefficients[1:3, 1:2]
#> Estimate Std. Error
#> (Intercept) 0.9839341 0.053084123
#> log(dat$N1[1:(t - 1)]) 0.9201864 0.009157214
#> log(dat$N2[1:(t - 1)]) -0.1443693 0.008777527
# Bayesian estimates
summary(m3.post.hmsc$Beta)$statistics[1:2, 1:2]
#> Mean SD
#> B[(Intercept) (C1), sp1 (S1)] 4.340340 0.09340565
#> B[(Intercept) (C1), sp2 (S2)] 3.809673 0.11430500

```

124 The HMSC model correctly shows the negative association between the species (shown here as correlations  
 125 - JSDM Chapter 7 tells us the intra-specific correlation is always 1).

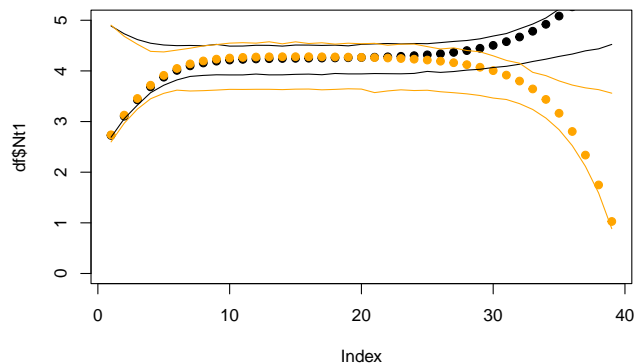


Figure 18: Observed (points) and HMSC model-predicted population size (logarithm) for Species 1 (black) and Species 2 (orange), using environment  $E_t$  as a fixed effect predictor and specifying temporal random effects to capture the species interactions.

```
knitr::knit_exit()
```

- 126 Beverton, R. J., and S. J. Holt. 1957. On the dynamics of exploited fish populations (Vol. 11). Springer  
 127 Science & Business Media.
- 128 Certain, G., F. Barraquand, and A. Gårdmark. 2018. [How do MAR\(1\) models cope with hidden nonlinear-](#)  
 129 [ities in ecological dynamics?](#) Methods in Ecology and Evolution 9:1975–1995.
- 130 Erickson, K. D., and A. B. Smith. 2023. [Modeling the rarest of the rare: A comparison between multi-](#)  
 131 [species distribution models, ensembles of small models, and single-species models at extremely low sample](#)  
 132 [sizes.](#) Ecography 2023:e06500.
- 133 Hart, S. P., and D. J. Marshall. 2013. [Environmental stress, facilitation, competition, and coexistence.](#)  
 134 Ecology 94:2719–2731.
- 135 Ingram, M., D. Vukcevic, and N. Golding. 2020. [Multi-output gaussian processes for species distribution](#)  
 136 [modelling.](#) Methods in Ecology and Evolution 11:1587–1598.
- 137 Ives, A. R. 1995. [Predicting the response of populations to environmental change.](#) Ecology 76:926–941.
- 138 Kloppers, P. H., and J. C. Greeff. 2013. [Lotka-volterra model parameter estimation using experiential data.](#)  
 139 Applied Mathematics and Computation 224:817–825.
- 140 Mühlbauer, L. K., M. Schulze, W. S. Harpole, and A. T. Clark. 2020. [gauseR: Simple methods for fitting](#)  
 141 [lotka-volterra models describing gauss's "struggle for existence".](#) Ecology and Evolution 10:13275–13283.

- 142 Olivença, D. V., J. D. Davis, and E. O. Voit. 2021. [Comparison between lotka-volterra and multivariate](#)  
143 [autoregressive models of ecological interaction systems](#). bioRxiv.
- 144 Ovaskainen, O., and N. Abrego. 2020. [Joint species distribution modelling: With applications in r](#). Ecology  
145 biodiversity and conservation. Cambridge University Press, United Kingdom.
- 146 Ovaskainen, O., G. Tikhonov, D. Dunson, V. Grøtan, S. Engen, B.-E. Sæther, and N. Abrego. 2017. [How](#)  
147 [are species interactions structured in species-rich communities? A new method for analysing time-series](#)  
148 [data](#). Proceedings of the Royal Society B: Biological Sciences 284:20170768.