Bayesian inference applied to the Lotka-Volterra model in Stan

Final Project

Bayesian Statistics and Probabilistic Programming

Authors

Alejandro Astruc

Joel Dieguez

Alba García

Clàudia Valverde

June 2024

Abstract

The Lotka-Volterra equations, also known as the predator-prey equations, are a set of first-order, non-linear differential equations used to describe the dynamics of biological systems in which two species interact, one as a predator and the other as prey. This project aims to apply Bayesian statistics to estimate the parameters and initial populations in the Lotka-Volterra model. Using the Rstan package, we encode a statistical model that accounts for measurement error and unexplained variation, employing the deterministic solutions of the equations as expected population sizes. Full Bayesian inference is then performed to solve the inverse problem of inferring parameters from noisy data. The data used contains Canadian lynx and snowshoe hare populations between 1900 and 1920, based on annual pelt collections by the Hudson's Bay Company. Furthermore, this project explores modifications to the standard model like adding new variables to the system or forecasting future (or past) populations, demonstrating the robustness and potential of Bayesian methods in ecological modeling.

Contents

1 Introduction **2**

1.1 The Lotka-Volterra model 2

2 Methodology **3**

2.1 Problem definition 3

2.2 Introduction to Rstan 4

3 Extending the base problem **6**

3.1 Forecasting 6

3.2 The competitive Lotka-Volterra model 6

3.3 Prior Analysis 8

3.3.1 σ study 8

3.3.2 θ study 8

4 Results **9**

4.1 Data Analysis 9

4.2 Simulated Posterior Distribution 9

4.3 Forecasting 10

4.4 Prior Analysis 11

4.4.1 Parametrizing sigma priors 11

4.4.2 Different prior distributions 12

5 Conclusions **14**

1 Introduction

Population dynamics is the portion of ecology that deals with the variation in time and space of population size and density for one or more species [1]. Predator-prey interactions examine the dynamic population shifts between two interdependent species. Research has shown that the sizes of these populations are linked. The primary interaction is that the prey population decreases when there are many predators, as they are consumed by them. Subsequently, when the prey population is significantly reduced, there aren't enough prey to sustain the predator population, leading to a decline in the number of predators due to starvation. This reduction in predators then allows the prey population to increase, and the cycle begins over again.

Figure 1: Lynx and snowshoe hare. © Tom and Pat Leeson

Lotka (1925) [2] and Volterra (1926) [3] formulated parametric differential equations that characterize the oscillating populations of predators and prey. The Lotka-Volterra models provide a valuable tool to help population ecologists understand the factors influencing population dynamics. These models have been particularly effective in understanding and predicting predator-prey population cycles. Although the models significantly simplify real-world conditions, they demonstrate that, under certain circumstances, predator and prey populations can oscillate over time.

One of the first studies of this type of population dynamics was done by Hewitt [4] as he observed the number of pelts collected by the Hudson's Bay Company, the largest fur trapper in Canada, between the years 1821 and 1914. The species of interest was snowshoe hares [5], an herbivorous cousin of rabbits, and Canadian lynxes [6], a feline predator whose diet consists largely of snowshoe hares (Figure 1). He observed the oscillation of population between the two species was correlated even analyzing co-founders and other external factors. In 2009, Howard [7], provided numerical data for the number of pelts collected by the Hudson's Bay Company in the years 1900-1920, which is what we will use as main dataset in our study.

In section 4.1, we have done some previous analysis of the data observing what we have described in this introduction.

1.1 The Lotka-Volterra model

The Lotka-Volterra model consists of a pair of first order differential equations:

$$\frac{\mathrm{d}}{\mathrm{d}t}u = (\alpha - \beta v)u = \alpha u - \beta uv \tag{1a}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}v = (-\gamma + \delta\, u)\, v = -\gamma v + \delta uv \tag{1b}$$

where u and v are the population densities of prey and predator respectively; α, the maximum prey per capita growth rate; β, the effect of the presence of predators on the prey death rate; γ, predator's per capita death rate; and δ, the effect of the presence of prey on the predator's growth rate. They represent the dynamics of a pair of predator-prey populations under a set of assumptions and approximations.

Some of the assumptions of the model are:

- The prey population has unlimited food.
- Predators are only sustained by prey.

- The rate of change of population is proportional (linear) to its size.
- Both environment and species do not change.
- Predators have limitless appetite.
- Both populations are summarized into a single variable, i.e., they are homogeneous in terms of hunger, reproduction, activity, etc.

In order to implement this model, we must incorporate sources of error in the measurements, variability and randomness and estimations of the different parameters involved in the model. We will use a Bayesian approach to do so.

2 Methodology

2.1 Problem definition

Our task is to estimate the different parameters in the Lotka-Volterra model 1, infer the actual populations and generate predictions for the populations. In order to achieve this, we will work within a Bayesian framework, proposing a series of priors to finally simulate the posterior. From the simulation we will obtain estimates for all the parameters.

First we propose the following priors for the model parameters:

$$\alpha, \gamma \sim \mathsf{Normal}(1, 0.5) \tag{2}$$

$$\beta, \delta \sim \mathsf{Normal}(0.05, 0.05) \tag{3}$$

We also need to account for the error in measurement, which we will be modeling it as

$$y_{n,k} \sim \mathsf{LogNormal}(\log(z_{n,k}), \sigma). \tag{4}$$

$$\sigma \sim \mathsf{LogNormal}(-1, 1) \tag{5}$$

where $y_{n,k}$ are the measures of population of species k at time-step n (i.e. the measurements that we know) and $z_{n,k}$ is the true population (a latent variable which we do not know), that is:

$$\begin{aligned} z_{n,1} &= u(t_n) \\ z_{n,2} &= v(t_n). \end{aligned}$$

In this way we are using a fully Bayesian approach for adjusting for uncertainty. This uncertainty can be seen in two forms, in one hand, the estimation uncertainty characterized by the joint posterior density $p(\alpha, \beta, \gamma, \delta, z_{\text{init}}, \sigma \mid y)$. On the other hand, as we have mention, the second form of uncertainty stems from measurement error and unexplained variation, which are both rolled into a single sampling distribution.

Next, to estimate the initial conditions of the dynamic system we will assume:

$$z_1^{init},\ z_2^{init} \sim \mathsf{LogNormal}(\log(10), 1)$$

Actually, in order to determine the dynamics of the system it would suffice to estimate any of the $z_{1,k}$ and $z_{2,k}$ not only the initial one. So whichever time-step of the actual populations we choose to have as prior is the one we will automatically estimate from the sampling. In our case we take the initial.

In order to translate the information between times-steps we need 1. We obtain this relation through numerically solving the dynamic equations. In a way the information conveyed by the dynamical model is equivalent to:

$$p(z_{n,k} \mid z_{1,k'}, z_{2,k'}, \alpha, \beta, \gamma, \delta)$$

and in our case $k' = 0$.

The simulation aspect of the problem emerges from the high amount of priors and the complex expressions for the time dependent solutions that make an analytical solution convoluted and hard to reach.

Given that an analytical solution is not possible to obtain, we will use Stan programming to be able to sample from the posterior distributions of the parameters.

Finally, to see how well calibrated our model and the parameters are, we will use Posterior Predictive Checks. The basic idea is to take the posterior for the fitted model and use it to predict what the data should have looked like. That is, we will be replicating new y values that parallel the actual observations y. Because they are replicated values, we write them as y^{rep}. The distribution of these replicated values is given by the posterior predictive distribution.

$$p(y_{\text{rep}} \mid y) = \int p(y_{\text{rep}} \mid \theta) p(\theta \mid y)\, d\theta \tag{6}$$

where $\theta = (\alpha, \beta, \gamma, \delta, z_{\text{init}}, \sigma)$ is the vector of parameters for the model.

Observe that, in the integral, the two terms represent the two different types of uncertainty. The first term, $p(y_{\text{rep}} \mid \theta)$, is the sampling distribution for the replications, which describes the distribution of the replicated observations y_{rep} given the parameters θ. This term captures the unexplained variance and measurement error. The second term, $p(\theta \mid y)$, represents the posterior distribution, reflecting our uncertainty in the parameter estimates θ given the observed data y. The integral effectively averages the sampling distribution, weighted by the posterior distribution.

In statistical terms, this process involves computing the expectation of the function $f(\theta) = p(y_{\text{rep}} \mid \theta)$ over the posterior $p(\theta \mid y)$, resulting in the conditional expectation:

$$p(y_{\text{rep}} \mid y) = \mathbb{E}[p(y_{\text{rep}} \mid \theta) \mid y]$$

The results for this implementation are in section 4.2.

2.2 Introduction to Rstan

Before continuing we want to dedicate a section to Rstan. We will work with Stan models to estimate the parameters explained above. For this reason, we have also researched about this programming software [8].

Stan is a C++ library for Bayesian modeling and inference that uses frequentist inference (Hamiltonian Monte Carlo) via optimization to obtain posterior simulations given a user-specified model and data. In R, we use the the R package `rstan` that allows us to conveniently fit Stan models from R (R Core Team 2014) and access the output, including posterior inferences and intermediate quantities such as evaluations of the log posterior density and its gradients.

Rstan is used via an Rstan program, this can be a text file containing the following mandatory blocks: data, parameters, transformed parameters and model.

The `data` block, specifies the data that is conditioned upon in Bayes Rule. It is necessary to declare the type of each variable (real, integer, array), and we can also constrain the values (for example, making them strictly larger than zero).

In our case, the data block is the following:

```
data {
  int<lower = 0> N;           // num measurements
  real ts[N];                 // measurement times > 0
  real y_init[2];             // initial measured population
  real<lower = 0> y[N, 2];    // measured population at measurement times
}
```

The `parameters` block declares the parameters whose posterior distribution is sought, which in our case are:

```
parameters {
```

```
  real<lower = 0> theta[4];   // theta = { alpha, beta, gamma, delta }
  real<lower = 0> z_init[2];  // initial population
  real<lower = 0> sigma[2];   // error scale
}
```

The `transformed parameters` block is used to redefine some parameters so the sampler runs more efficiently because the resulting multivariate geometry is more amendable to Hamiltonian Monte Carlo. In our case we define the solutions to the Lotka-Volterra equations for a given initial state z^{init} as transformed parameters. This will allow them to be used in the model and inspected in the output. It also makes it clear that they are all functions of the initial population and parameters. Note that we use the function `integrate_ode_rk45` to compute the solutions of the differential equations. This is because Rstan has versions of many of the most useful R functions for statistical modeling, including probability distributions, matrix operations, and various special functions.

```
transformed parameters {
  real z[N, 2]
    = integrate_ode_rk45(dz_dt, z_init, 0, ts, theta,
                         rep_array(0.0, 0), rep_array(0, 0),
                         1e-6, 1e-5, 1e3);
}
```

The `model` block just reflects the conventional statistical notation for the modelled priors. In our case, this is the following:

```
model {
  theta[{1, 3}] ~ normal(1, 0.5);
  theta[{2, 4}] ~ normal(0.05, 0.05);
  sigma ~ lognormal(-1, 1);
  z_init ~ lognormal(log(10), 1);
  for (k in 1:2) {
    y_init[k] ~ lognormal(log(z_init[k]), sigma[k]);
    y[ , k] ~ lognormal(log(z[, k]), sigma[k]);
  }
}
```

Additionally, Stan also permits users to define their own functions that can be used throughout a Stan program. These functions are defined in the functions block. The functions block is optional but, if it exists, it must come before any other block. This mechanism allows users to implement statistical distributions or other functionality that is not currently available in Stan. In our case, we have used this block to define our system of differential equations.

```
functions {
  array[] real dz_dt(real t, // time
                     array[] real z,
                     // system state {prey, predator}
                     array[] real theta, // parameters
                     array[] real x_r, // unused data
                     array[] int x_i) {
    real u = z[1];
    real v = z[2];

    real alpha = theta[1];
    real beta = theta[2];
    real gamma = theta[3];
    real delta = theta[4];

    real du_dt = (alpha - beta * v) * u;
    real dv_dt = (-gamma + delta * u) * v;
    return {du_dt, dv_dt};
  }
}
```

Stan requires the system to be defined with exactly the signature defined here for the function `dz_dt()`, even if not all of them are used. The first argument is for time, which is not used here because the rates in the Lotka-Volterra equations do not depend on time. The second argument is for the system state, coded as an array $z = (u, v)$. The third argument is for the parameters of the equation, of which the Lotka-Volterra equations have four, which are coded as $\theta = (\alpha, \beta, \gamma, \delta)$. The fourth and fifth argument are for data constants, but none are needed here, so these arguments are unused.

Finally, we can also use the `generated quantities` block. This block defines predictive quantities which is executed once per iteration. In our case we use it to define the posterior predictive checks.

```
generated quantities {
  real y_init_rep[2];
  real y_rep[N, 2];
  for (k in 1:2) {
    y_init_rep[k] = lognormal_rng(log(z_init[k]), sigma[k]);
    for (n in 1:N)
      y_rep[n, k] = lognormal_rng(log(z[n, k]), sigma[k]);
  }
}
```

With this explanation, our basic Rstan model is defined. In the further implementations we will change this code accordingly to the modifications we want to perform.

3 Extending the base problem

Now that we have our base problem defined, as well as the Rstan program that we will use, we want to look at what modifications and further action we can do to extend our scope.

In particular, we have looked at three groups of modifications, first, extending the sampling from the posterior distribution to be able to forecast the values of both populations in the future. Second, modifying the base dynamic system in order to introduce a model that is better suited for real life behaviour. Finally, we have introduced changes in the prior distributions to see how the results are affected by these firsts assumptions.

3.1 Forecasting

As explained before, we use the posterior predictive checks with our same data and check how well calibrated our model is. Nevertheless, we can use the same idea and extend our model past the data that we have, producing a forecasting for both species population.

The idea is to let the model generate samples into the future for which we do not have any data. These samples are not taken into account in the fitting of parameters, but are generated each time so they can be extracted from the simulation.

What we are obtaining in practice is the simulation of the posterior predictive distribution using the same formula as in 6.

In order to obtain forecasts from our Rstan model we need to modify the initial code. More specifically, we have modified the `generated quantities` and the `transformed parameters` blocks allowing for the generation of new future samples based on the model parameters.

The results for this modification are explained in section 4.3.

3.2 The competitive Lotka-Volterra model

To introduce modifications to the Lotka-Volterra equations we can consider a lot of different factors, in particular, we will look at introduction of the **carrying capacity** variable.

The carrying capacity is roughly the maximum population that an environment can sustain. Note that in the first approach defined in equations 1, the prey population is modeled under the assumption of unlimited food supply. This assumption implies that, in case of extinction, both species have to become extinct simultaneously. For example, we cannot have that the predator population is extinct and the prey population becomes stable since if population v drops close to zero, then u is subject to unconstrained growth which eventually leads to an increase in v, stopping it from dropping to zero [9].

A better model would be to include a notion of self-competition in the individual populations, i.e. we need a term which is small for small x allowing the population to grow, but dominates the growth term when x gets larger, thus restricting its growth. The simplest example of such a model (for a single

population first) is the Logistic equation 7, originally introduced by Pierre-Francois Verhulst in 1838, the equation is non-linear and takes the form

$$\frac{\mathrm{d}}{\mathrm{d}t}x = ax\left(1 - \frac{x}{K}\right) \tag{7}$$

where a > 0 is the usual growth term and, K is the limiting population term (or carrying capacity term).

With that, we can modify our first model and impose a logistic growth on both populations 8, relying on the interaction to alter the behaviour of the system.

$$\frac{\mathrm{d}}{\mathrm{d}t}u = \alpha u\left(1 - \frac{u}{\nu_1}\right) - \beta uv \tag{8a}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}v = \gamma v\left(1 - \frac{v}{\nu_2}\right) + \delta uv \tag{8b}$$

Now, we want to apply this modification to our original model. We will do as before and try to estimate the carrying capacity parameters (ν_1 and ν_2) using the Bayesian approach.

To model the carrying capacity, we have researched what distribution we should use. It has been difficult to find how this parameter should behave and what assumptions we can make. Since we do not have a lot of knowledge on population dynamics, we have chosen to use uninformative priors (based on the data) to model ν_1 and ν_2.

The two largest values (in thousands) for the lynx population in our data are 59.4 and 51.10 before starting to decrease due to lack of hares. And for the hare population, we have 77.4 and 76.6 before decreasing due to an increase of the lynxes. We can establish an interval around these values to approximate the carrying capacity of each species. As we have said before, we will use a uniform distribution (since we do not have any other information on these parameters). With that, we will have the following:

$$\nu_1 \sim U(65, 85) \tag{9}$$

$$\nu_2 \sim U(45, 65) \tag{10}$$

With this theoretical formulation, we can change or Rstan model accordingly to obtain new estimates for the parameters (including ν_1 and ν_2). For that, we have made the following changes.

i) We have modified the equations in the `functions` block:

```
real Ku = theta[5];
real Kv = theta[6];

real du_dt = alpha * u * (1 - (u / Ku)) - beta * u * v;
real dv_dt = gamma * v * (1 - (v / Kv)) - delta * u * v;
```

ii) We have added both variables in the `model` block:

```
theta[5] ~ uniform(65, 85);
theta[6] ~ uniform(45, 65);
```

Unfortunately, we were not able to reproduce the experiments because, when fitting the model we obtain an error on the initialization of the variables. We think is because we are obtaining negative values for z with the new differential equations that yield the error when evaluating $log(z)$ for the LogNormal distribution of y.

As future work, with more time and knowledge about Rstan we would like to continue exploring this modification.

3.3 Prior Analysis

Finally we are also going to analyse how the different parameters affect our model. To do so we can change the prior parameters or we can also change the distribution they follow.

The results from these modifications are explained in section 4.4.

3.3.1 σ study

Firstly, we aimed to investigate how different values of σ affected the fitted model. Recognizing that σ is related to the variability (measurement error) of our model, we conducted an analysis by altering the parameters of the original distribution for σ, which is $\sigma \sim \mathsf{LogNormal}(-1, 1)$. Adjusting these parameters effectively changes our prior assumptions about the model's variability. For example, shifting from $\mathsf{LogNormal}(-1, 1)$ to $\mathsf{LogNormal}(0.5, 1)$ implies adopting a higher mean and variance for the measurement error. To examine the impact of these changes, we selected an arbitrary range of values and made the following parameter modifications to reflect this adjustment in the prior assumptions about the model's variability:

- $\sigma \sim \mathsf{LogNormal}(-0.5, 0.5)$
- $\sigma \sim \mathsf{LogNormal}(0, 0.5)$
- $\sigma \sim \mathsf{LogNormal}(0.5, 1)$

3.3.2 θ study

We also studied different distributions of the parameters in θ, that is the different distributions for α, β, γ and δ.

We did not have much information to base our decisions so we decided to try some very well priors, some not very informative and some more specialized on the range of values we had seen these parameters take in different implementations of the Lotka-Volterra equations. The distributions we considered are:

- Normal. Our original choice. It is very typical in many natural phenomena, so it seemed appropriate for this problem. Also, it has some good properties like being symmetrical, and is centered around an expected point but can spread to consider enough new values.

  We used $\alpha, \gamma \sim \mathsf{Normal}(1, 0.5)$; $\beta, \delta \sim \mathsf{Normal}(0.05, 0.05)$.

- Beta. This distribution is constrained between 0 and 1, thus it is suitable for modeling proportions or probabilities. For example, it is useful for parameters representing proportions of resources or individuals within a population.

  We used $\alpha, \gamma \sim \mathsf{Gamma}(4, 4)$; $\beta, \delta \sim \mathsf{Gamma}(2, 40)$.

- Gamma. This distribution is used to model positive-valued variables (as are all our parameters) and is very versatile. It allows for skewness, making it suitable for parameters such as growth rates or interaction strengths.

  We used $\alpha, \gamma \sim \mathsf{Beta}(2, 2)$; $\beta, \delta \sim \mathsf{Beta}(2, 40)$.

- Exponential. It is commonly used for modeling waiting times or decay rates. It is appropriate for parameters related to rates of change or decay within ecological systems.

  We used $\alpha, \gamma \sim \mathsf{Exp}(1)$; $\beta, \delta \sim \mathsf{Exp}(20)$.

- Uniform. It assumes that all values within a given range are equally likely. We chose this distribution to try a a wide range of possibilities as we had little prior knowledge about the parameter values.

  We used $\alpha, \gamma \sim \mathsf{Unif}(0.5, 1.5)$; $\beta, \delta \sim \mathsf{Unif}(0, 0.1)$.

4 Results

In this section we present the results obtained using the base model, as well as the results from the different modifications explained in the previous section.

4.1 Data Analysis

As a first look into the data, we would like to perform some visualizations in order to understand with what type of data frame we are working on. Take into account that in order to fit the data into the Stan model we have had to perform small modifications on its structure. From the plot 2 we can clearly see that the spikes in the lynx population lag those in the hare population. When the populations are plotted against one another over time, the population dynamics orbit in an apparently stable pattern. Volterra (1926) recognized that these population oscillations could be modeled with a pair differential equations similar to that used to describe springs.

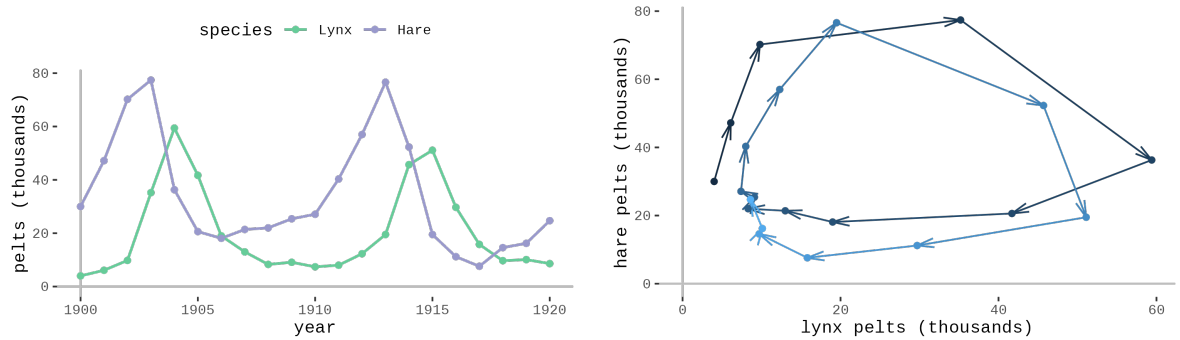

Figure 2: (a) Number of lynx and hare pelts collected by the Hudson's Bay Company between 1900 and 1920, and (b) Number of pelts collected for lynx versus hares from 1900 to 1920. This plot is similar to that of the dynamics of a spring in phase space (i.e., position vs. momentum).

4.2 Simulated Posterior Distribution

First of all we are going to fit the Hudson's Bay Company data into the our Lokta-Volterra model in Stan code. In table 1 we can see the results of the fitted model with the different parameters, we can observe that all $\hat{R}$ values are close to 1 indicating consistency in convergence and that our Stan model is a good approach for further analysis as it has produced an adequate approximation of the posterior.

Parameter	Mean	SE_Mean	SD	10%	50%	90%	N_Eff	$\hat{R}$
theta[1]	0.546	0.002	0.065	0.465	0.544	0.630	1011	1.002
theta[2]	0.028	0.000	0.004	0.023	0.027	0.033	1077	1.003
theta[3]	0.802	0.003	0.093	0.688	0.796	0.920	977	1.003
theta[4]	0.024	0.000	0.004	0.020	0.024	0.029	1026	1.003
sigma[1]	0.249	0.001	0.044	0.199	0.243	0.307	2685	1.000
sigma[2]	0.250	0.001	0.044	0.202	0.244	0.306	2662	1.000
z_init[1]	33.984	0.054	2.918	30.353	33.907	37.897	2952	1.001
z_init[2]	5.946	0.011	0.524	5.305	5.917	6.621	2177	0.999

Table 1: Summary statistics of the parameters after fitting the data

To check our model in a non-statististical approach we can compare our mean parameters with the ones provided by Howard, 2009 [7] based on the same data (denoted with *).

$$\alpha^* = 0.55,\ \beta^* = 0.028,\ \gamma^* = 0.84,\ \delta^* = 0.026$$

$$\hat{\alpha} = 0.55,\ \hat{\beta} = 0.028,\ \hat{\gamma} = 0.80,\ \hat{\delta} = 0.024$$

As it can be observed, our parameters are very close to Howard's estimates. Furthermore, the posterior intervals, which are quite wide, can also be interpreted probabilisticly:

$$\Pr[0.47 \leq \alpha \leq 0.63] = 0.8, \quad \Pr[0.23 \leq \beta \leq 0.33] = 0.8,$$

$$\Pr[0.69 \leq \delta \leq 0.91] = 0.8, \quad \Pr[0.020 \leq \gamma \leq 0.029] = 0.8$$

Finally, the error scales for both populations have the same posterior mean estimate, $\hat{\sigma}_1 = \hat{\sigma}_2 = 0.25$, and both have the same posterior 80% interval, $(0.20, 0.31)$. This suggests they may be completely pooled and modeled using a single parameter.

Having confirmed that we have a model that fits our data correctly we can start thinking about possible inferences and analysis using a Bayesian approach.

We will infer the population sizes of lynx and hares over time, it should be taken into account that we can only estimate the expected sizes of future pelt collections and thus, for now we will assume the population size is somehow directly related to the numbers of pelts collected. Howard ([7]) uses optimization-derived point estimates to derive population predictions, but we are going to follow a fully Bayesian approach for adjusting for uncertainty described in methods.

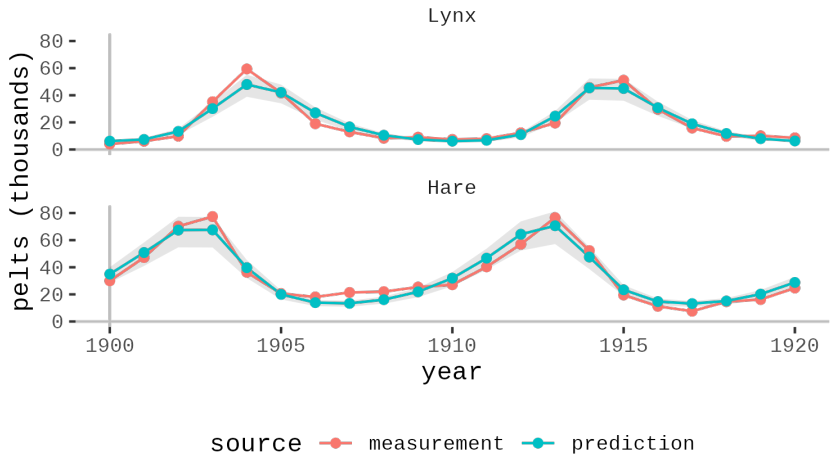

Figure 3: Posterior predictive checks, including posterior means and 50% intervals along with the measured data. If the model is well calibrated, as this one appears to be, 50% of the points are expected to fall in their 50% intervals.

We use posterior predictive checks to evaluate how well our model fits the data from which it was estimated. As mentioned above, we can have two forms of uncertainty, first we have plotted (in Figure 3) the original measured data against the simulated data and in gray the unexplained variance and measurement error, we can visually see how well our estimations fit the data, the gray area is not very good so we can confirm again that we have a good model for the estimation. For the analysis of the second uncertainty we show the expected population orbit for one hundred draws from the posterior computed with the two types of uncertainties, it can be observed how when adding y^{ref} the expected population orbit is way more noisy whereas the one with measurement error is more smooth (as shown in Figure 4).

4.3 Forecasting

After modifying the code accordingly as described in Sec. 3.1, we obtained the results in Figure 5.

We can appreciate how the oscillating behaviour extends over time. As it was to be expected, the forecast becomes more imprecise over time, broadening the uncertainty of the predictions.

Another behaviour we can observe is how at the peaks of populations the uncertainty increases. This is due to the fact that at those values (because of the linearity of the dynamic equations) the uncertainty of the dynamic parameters becomes most relevant thus leading to more variability.

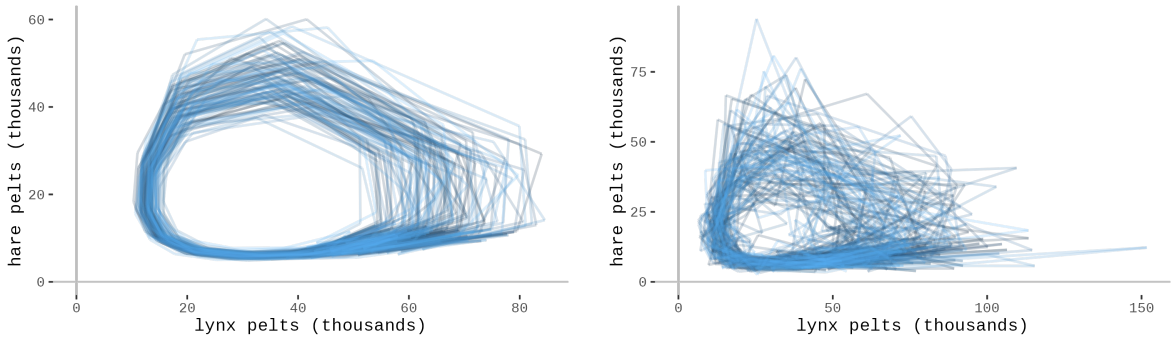

Figure 4: Comparison of expected population orbit for one hundred draws from the posterior for the different types of uncertainties. **a)** Each draw represents a different orbit determined by the differential equation system parameters. Together they provide a sketch of posterior uncertainty for the *expected* population dynamics. If the ODE solutions were extracted per month rather than per year, the resulting plots would appear fairly smooth. **b)** Even if plotted at more fine-grained time intervals, error would remove any apparent smoothness. Extreme draws as seen here are typical when large values have high error on the multiplicative scale.

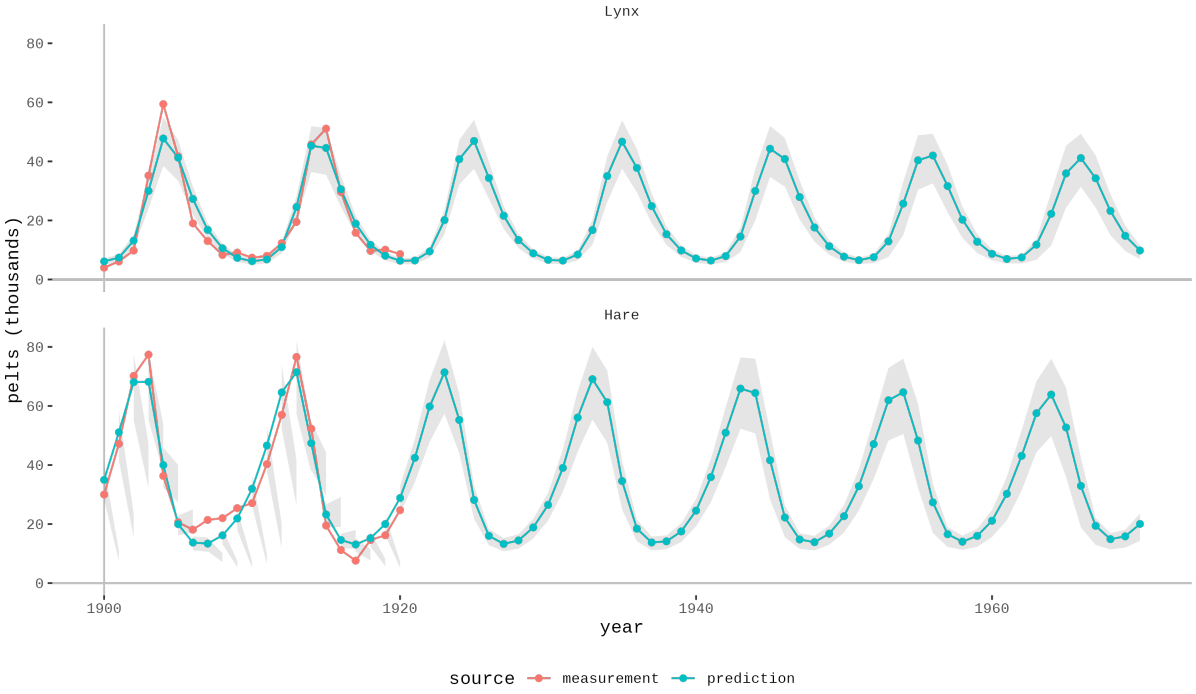

Figure 5: Forecasted data for next 20 years after last recorded value. Gray area represents maximal a minimal values for the generated estimations.

4.4 Prior Analysis

4.4.1 Parametrizing sigma priors

In order to interpret the results of the different parameters of σ we decided to plot the density distribution for each parameter from the MCMC chains seen in Figure 6. As we have seen in class, in Bayesian statistics, MCMC (Markov Chain Monte Carlo) is commonly used to obtain samples from the posterior distribution of model parameters, allowing for Bayesian inference and model fitting. The plots are organized in a grid with each row representing different σ parameters. For every tested parameters we plot two σ corresponding to the different errors 2.1, one related to measurement error and the other to estimation of the posterior distribution. Each density plot shows the distribution of sampled values

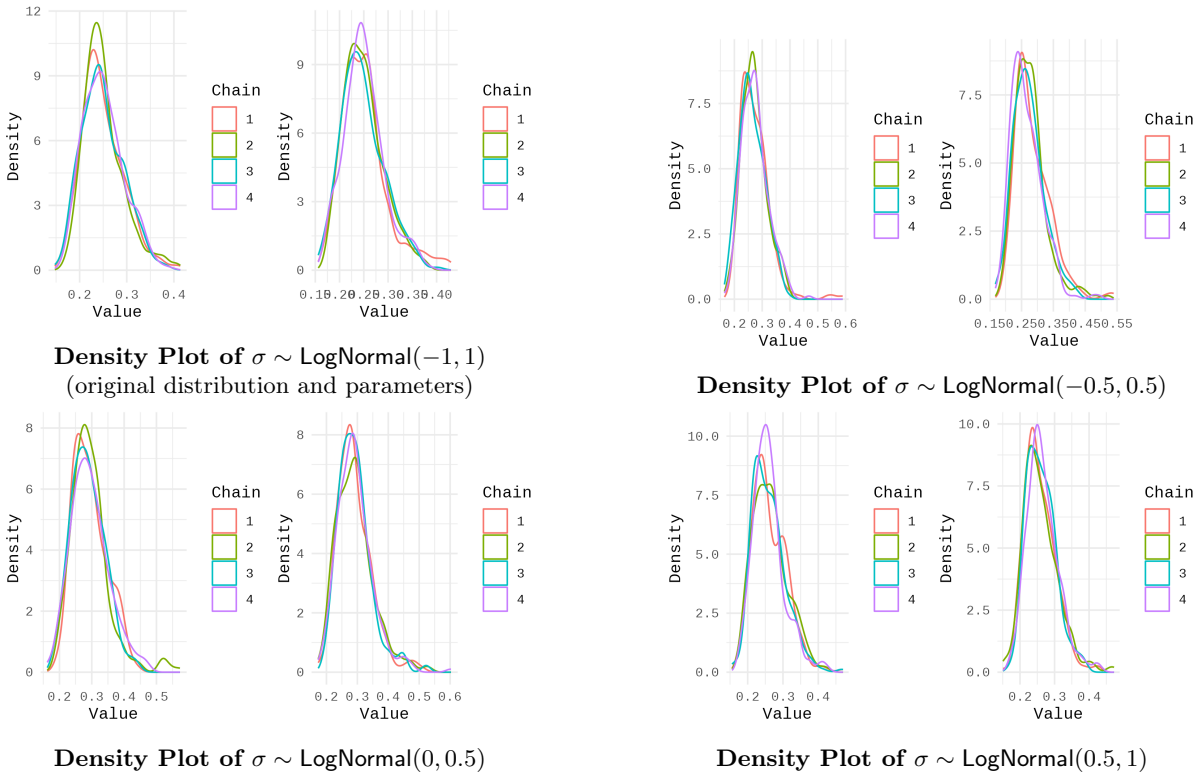

Figure 6: Density plots of MCMC samples for the parameters σ across four different chains. For each tested parameter we plot the two types of σ; the one related to measurement error and the other related to the parameter estimates.

from the MCMC chains, with different colors representing the four different chains. The overlapping of different chains in the density plots indicates the convergence and robustness of the MCMC sampling under the new prior settings. Due to the consistent similarity in density distributions observed across all tested parameters, we can conclude that the model behaves consistently in terms of variance across the two types of σ.

4.4.2 Different prior distributions

As we explained in 3.3, we considered five different distributions for the parameters of the model: Normal, Beta, Gamma, Exponential and Uniform.

In Figure 7, you can see the histogram of the sampling from the different distributions for the parameters.

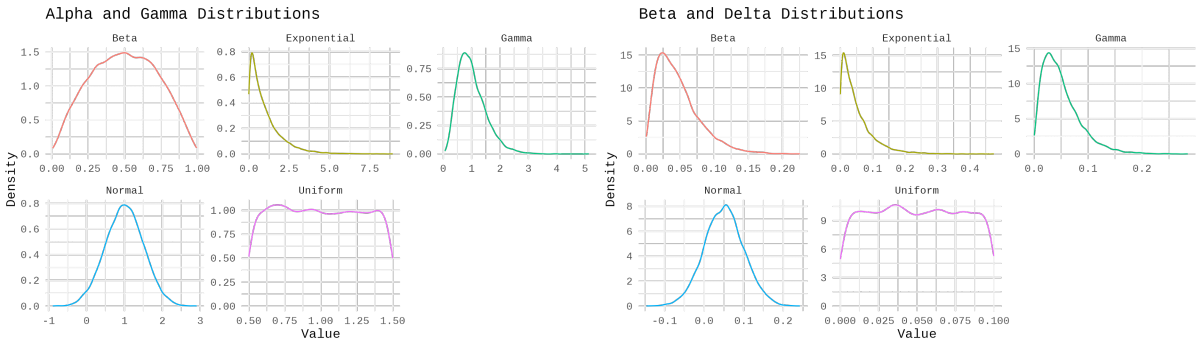

Figure 7: Line histogram of the sampling from each prior distribution.

Then, we fit a model with each one of them. In Figure 8, we show the before and after fitting the data for each distribution. We can clearly see how the parameters have narrowed down to similar values.

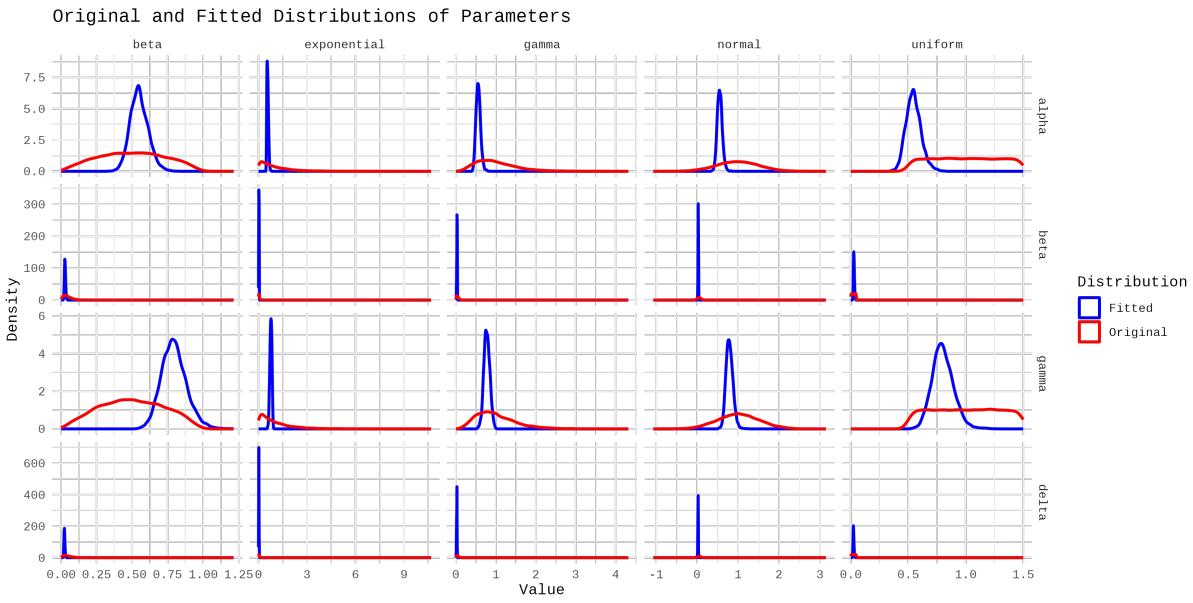

Figure 8: Comparison of the parameter distributions before and after fitting the data.

Furthermore, to compare the results properly, Figure 9 shows the final fitted distribution for each parameter. The results for all of them converged to pretty much the same results. However, to truly evaluate the performance of each one of them we needed a metric. We decided to compare them to the results found by Peter Howard using a non-statistically motivated error term and optimization[7]. In table 2 we show the estimates found from each distribution and their euclidean distance to our reference.

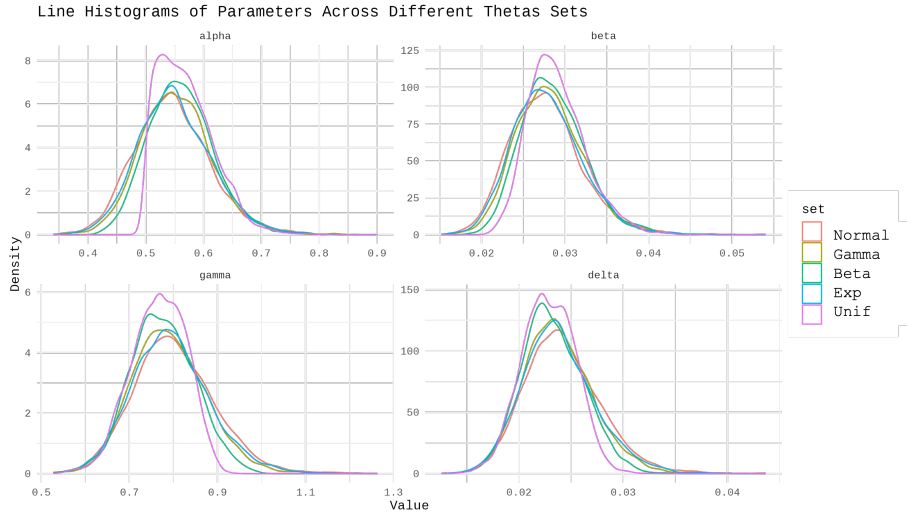

Figure 9: Comparison of the fitted distributions for each parameter.

Distribution	**alpha**	**beta**	**gamma**	**delta**	**dist to ref**
Reference	0.5500000	0.02800000	0.8400000	0.02600000	-
Normal	0.5463612	0.02767999	0.8016689	0.02416330	**0.03854851**
Gamma	0.5528420	0.02812879	0.7908872	0.02375394	0.04924638
Beta	0.5615432	0.02862592	0.7751996	0.02318254	0.06588379
Exp	0.5498255	0.02786482	0.7945136	0.02381079	0.04553961
Unif	0.5689974	0.02902932	0.7664628	0.02282393	0.07602479

Table 2: Comparison of the estimates found for each parameter using the different distributions and their distance to the reference chosen.

The metric we used it not super accurate as the reference values might not be 100% correct but it is the only way we could find to evaluate them. After seeing this results, we can conclude that the Normal Distribution is the one that approximates the aimed estimations the best.

5 Conclusions

After completing our analysis, we successfully applied a fully Bayesian approach to a system of dynamic differential equations, specifically the Lotka-Volterra model. We were able to fit the data, simulate the posterior distributions, and perform forecasting. Additionally, we analyzed the impact of different parameters and prior distributions. Our key findings are as follows:

i) Our Bayesian approach is accurate, fits our data correctly, and can reproduce the same results as other non-statistical approaches.

ii) The Rstan model con generate forecasts for the populations, but they become more variable over time as the uncertainty from the dynamic parameters propagate into the estimations.

iii) In exploring different parameters of the prior distribution for σ, we observed minimal differences in their density distributions because the MCMC chains consistently converged. When a model behaves correctly it should converge independently of the prior (if we are always in the same distribution), we can conclude that it is the case.

iv) From the study of parameters of the prior distribution of σ we can also conclude that the different sigmas for each parameter always follow the same distribution indicating that the two types of errors are more or less the same in all cases.

v) We also tried different modifications of the model by testing different distributions for the parameters of the equations. The results were very similar for all of them but we reached to the conclusion that the Normal distribution performed the best with our metric.

Although we achieved positive results, we recognize that other contemporary methods might be more suitable and precise for certain tasks we performed. Nonetheless, by following the Bayesian approach, we identified additional tasks we would have liked to address. These include fitting our model to new datasets with different species and corresponding priors, and incorporating more confounders such as environmental factors and interactions with external species. Unfortunately, we were limited by the availability of new and sufficiently robust datasets to apply the Bayesian approach to these tasks.

References

[1] S. A. Juliano, "Population dynamics," *Journal of the American Mosquito Control Association*, vol. 23, no. 2 Suppl, p. 265, 2007.

[2] A. J. Lokta, "Elements of physical biology," *Williams and Wilkins, Baltimore*, 1925.

[3] V. Volterra, "Fluctuations in the abundance of a species considered mathematically," *Nature*, vol. 118, no. 2972, pp. 558–560, 1926.

[4] C. G. Hewitt, *The conservation of the wild life of Canada*. New York: C. Scribner, 1921.

[5] C. J. Krebs, S. Boutin, R. Boonstra, A. Sinclair, J. Smith, M. R. Dale, K. Martin, and R. Turkington, "Impact of food and predation on the snowshoe hare cycle," *Science*, vol. 269, no. 5227, pp. 1112–1115, 1995.

[6] K. G. Poole, "A review of the canada lynx, lynx canadensis, in canada," *The Canadian Field-Naturalist*, vol. 117, no. 3, pp. 360–376, 2003.

[7] P. Howard, "Modeling basics," *Lecture Notes for Math*, vol. 442, 2009.

[8] S. D. Team, *RStan: the R interface to Stan*, 2024.

[9] C. Prior, "The competitive lotka-volterra equations." Department of Mathematical Sciences, Durham University.