Bayesian Differential Equations

EES 5891-03

Bayesian Statistical Methods

Jonathan Gilligan

Class #25: Thursday, December 01 2022

Chimpanzees Cracking Nuts

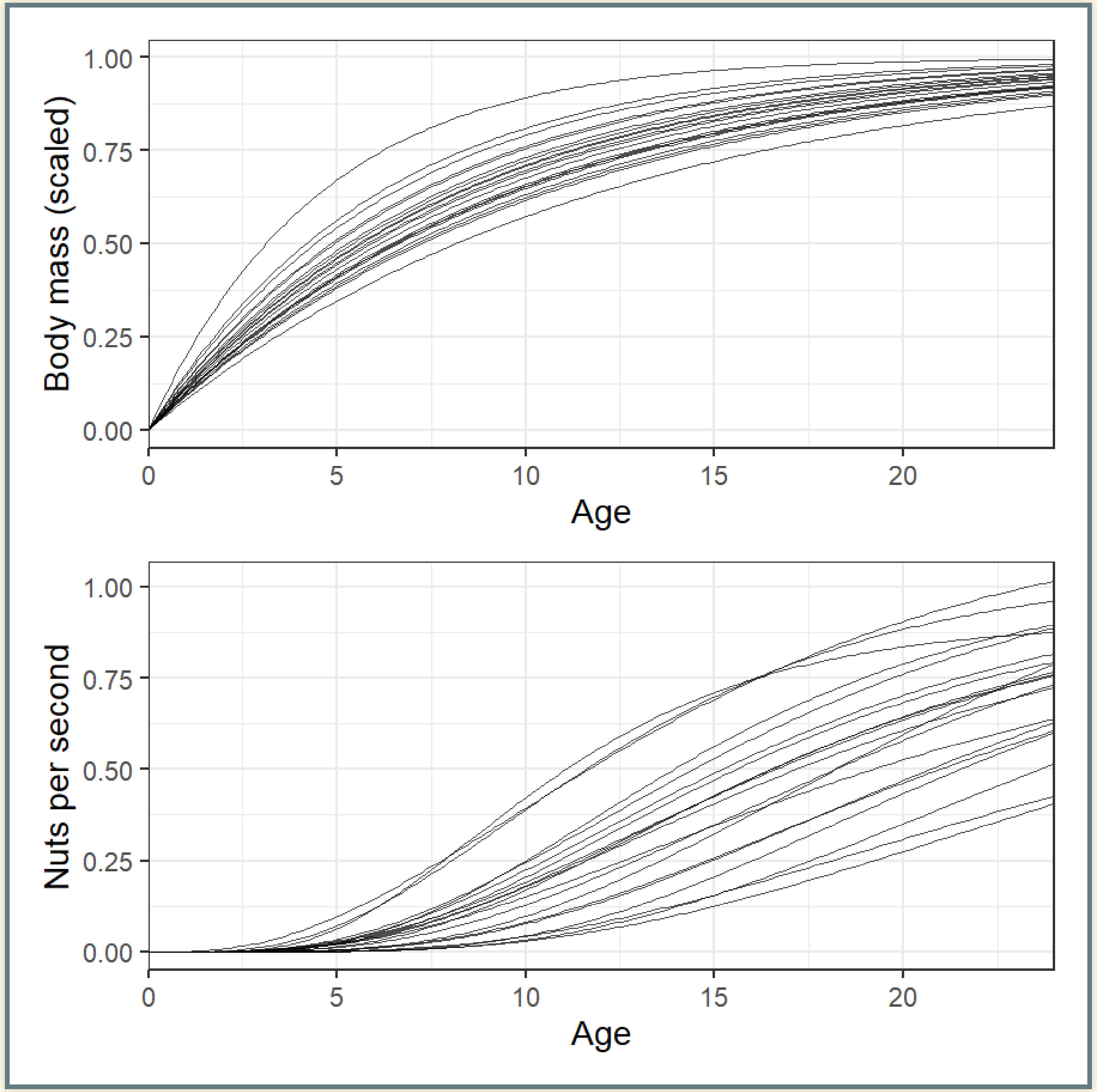
Panda Nuts (*Panda oleosa*)

Chimpanzees and Tools

- Tool use by non-human primates
- How do chimpanzees learn to use tools to crack panda nuts?
- Observe how # nuts cracked/second changes as chimps age
- Model:
  - Strength is related to body size
  - Mass (*M*) vs. age (*t*):
    - Growth: $$\small \frac{\mathrm{d}M}{\mathrm{d}t} = k (M_{\text{max}} - M(t)) $$
    - Solve the differential equation $$\small M(t) = M_{\text{max}} (1 - \exp(-kT)) $$
  - Strength: $\small S(t) = \beta M(t)$
- Nut-cracking rate per second (λ): $$ \begin{align} \lambda &= \alpha S(t)^\theta \\ &= \alpha \left(\beta M_{\text{max}} (1 - \exp(-kt)) \right)^\theta \end{align} $$
- Scale mass and simplify: $$ \begin{align} \lambda &= \alpha \left(\beta (1 - \exp(-kt)) \right)^\theta \\ &= \alpha \beta^\theta (1 - \exp(-kt))^\theta \\ &= \phi (1 - \exp(-kt))^\theta \end{align} $$

Statistical Model

- # nuts cracked in *d* seconds: $$ \begin{align} n &\sim \text{Poisson}(\lambda) \\ \lambda &= d \phi (1 - \exp(-kt))^\theta \end{align} $$
- Priors: $$ \begin{align} \phi &\sim \text{Log-Normal}(\log(1), 0.1) \\ k &\sim \text{Log-Normal}(\log(2), 0.25) \\ \theta &\sim \text{Log-Normal}(\log(5), 0.25) \\ \end{align} $$



Coding the Model

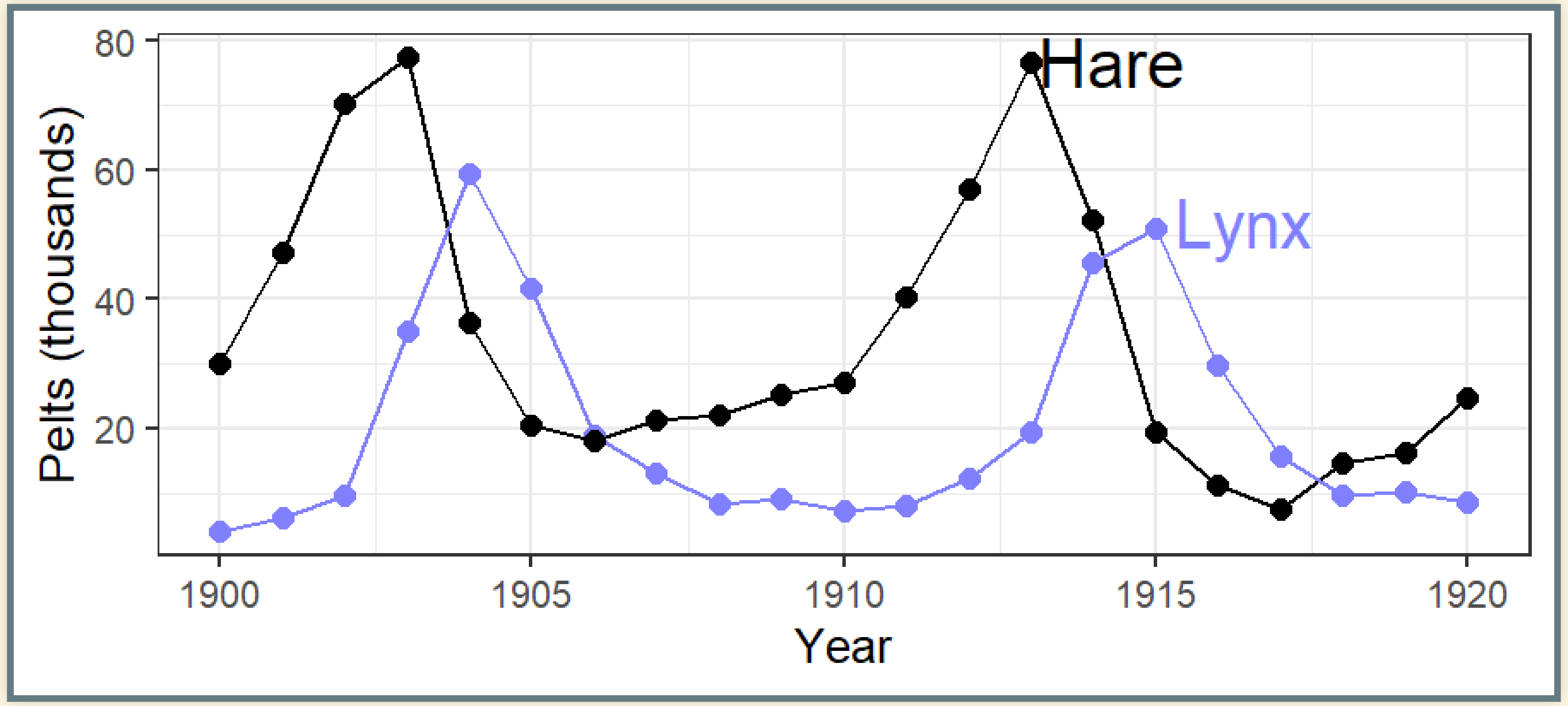
Population Dynamics

Predator-Prey Model

- Hares (prey) and Lynx (predators)

```
data(Lynx_Hare)
df <- Lynx_Hare
```

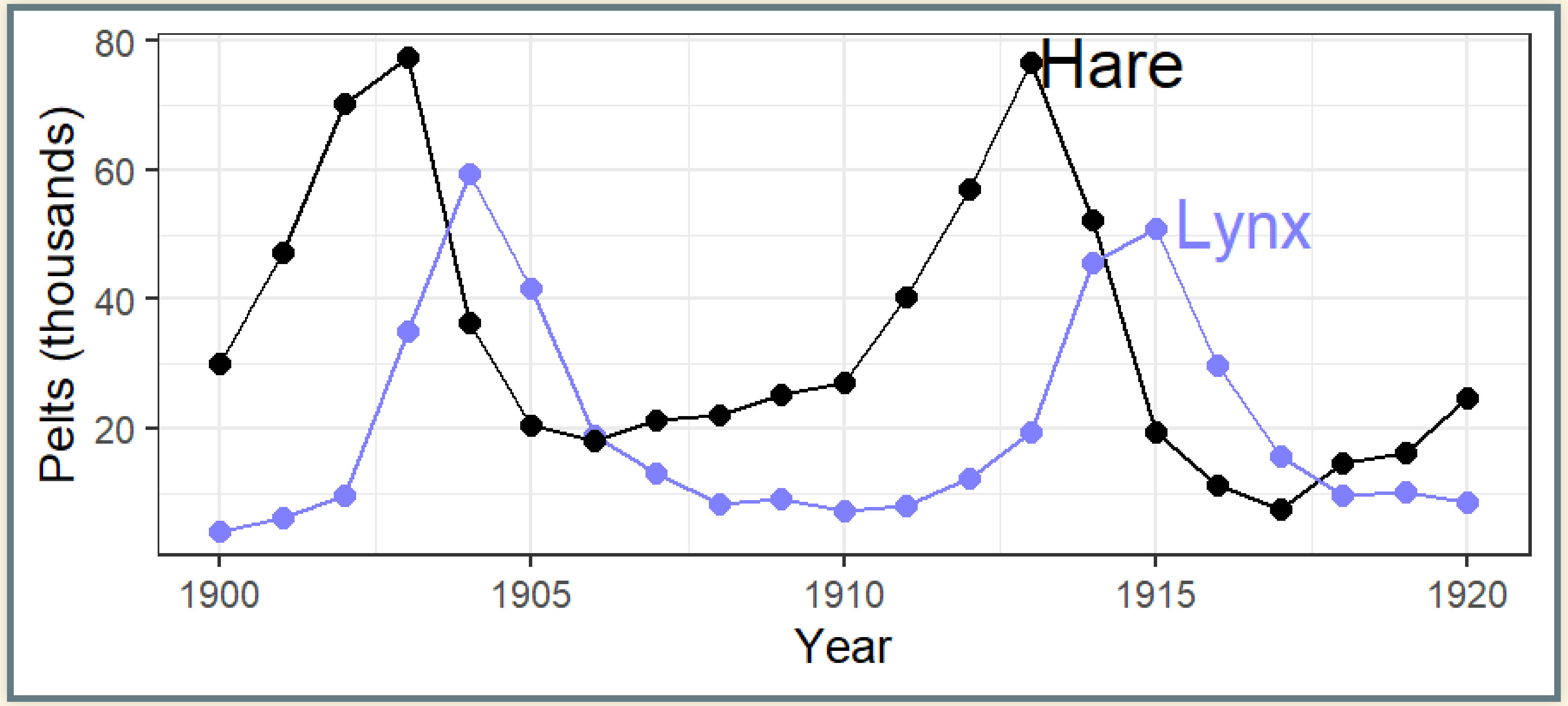
- Lynx eat hare
  - Next year: more lynx, fewer hares.
- As hare population drops, Lynx don't have enough food
  - Lynx population drops
- As lynx population drops, hare population rebounds



- Modeling the data:
  - Geocentric: **autoregressive model** $E(H_t) = \alpha + \beta_1 H_{t - 1}$ *E = expectation value*
    - Add an epicycle for the Lynx $E(H_t) = \alpha + \beta_1 H_{t - 1} + \beta_2 L_{t - 1}$

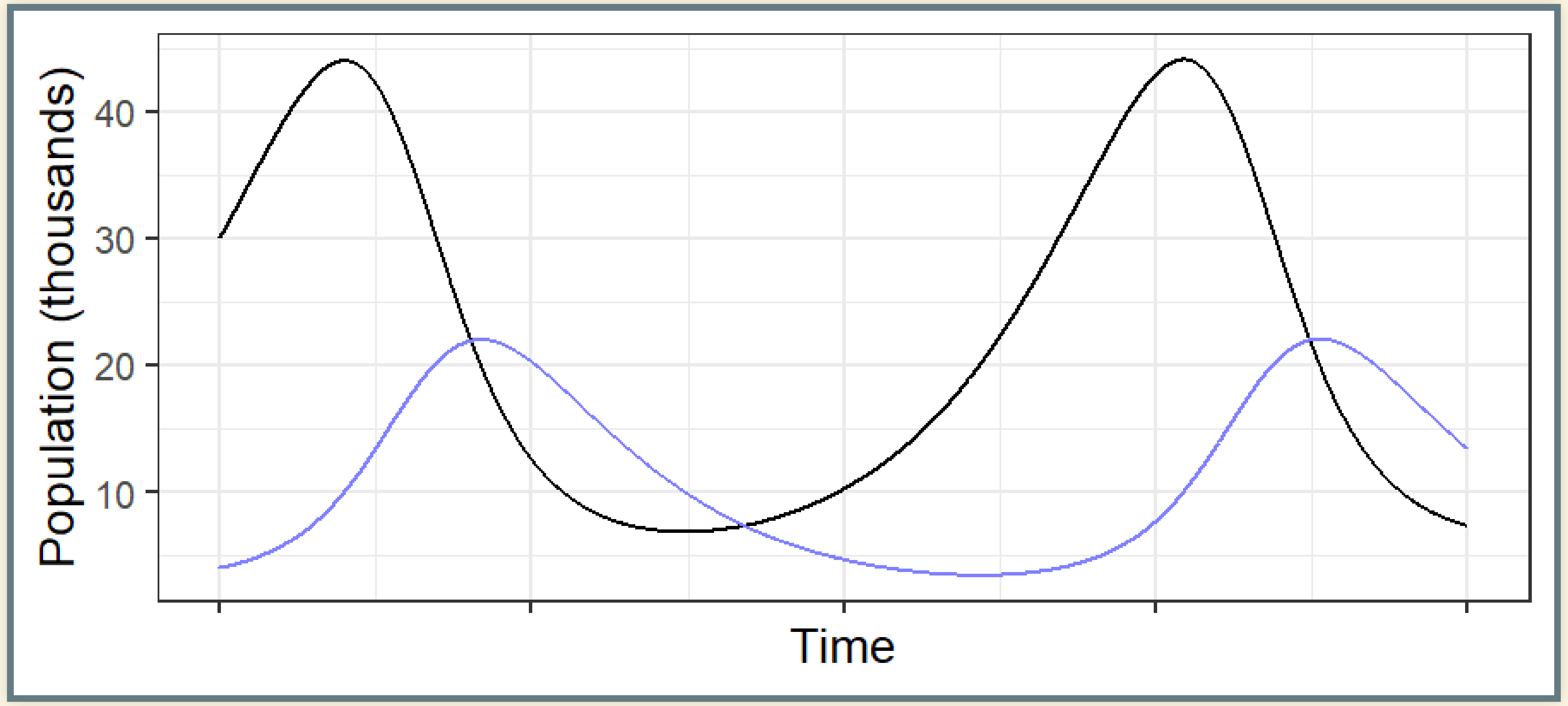
Lotka-Volterra Model

- Ordinary Differential Equation:
  - Birth and death: \[\begin{align} \frac{\mathrm{d}H}{\mathrm{d}t} =& H_t \times (\text{birth rate}) \\ & - H_t \times (\text{mortality rate}) \\ =& b_H H_t - m_H H_t \\ =& (b_H - m_H) H_t \end{align} \]
  - Hare mortality depends on # lynx
  - Lynx birth rate depends on # hare \[\begin{align} \frac{\mathrm{d}H}{\mathrm{d}t} &= (b_H - m_H L_t) H_t \\ \frac{\mathrm{d}L}{\mathrm{d}t} &= (b_L H_t - m_L) L_t \end{align} \]



Simulate Population Dynamics

```
sim_lh <- function(n_steps, init, theta, dt = 0.002) {
  L <- rep(NA, n_steps)
  H <- rep(NA, n_steps)
  L[1] <- init[1]
  H[1] <- init[2]
  for (i in 2:n_steps) {
    H[i] <- H[i-1] + dt * H[i-1] * (theta[1] - theta[2] *
         L[i-1])
    L[i] <- L[i-1] + dt * L[i-1] * (theta[3] * H[i-1] -
         theta[4])
  }
  return(tibble(H = H, L = L, time = seq_along(H)))
}
```



Statistical Model

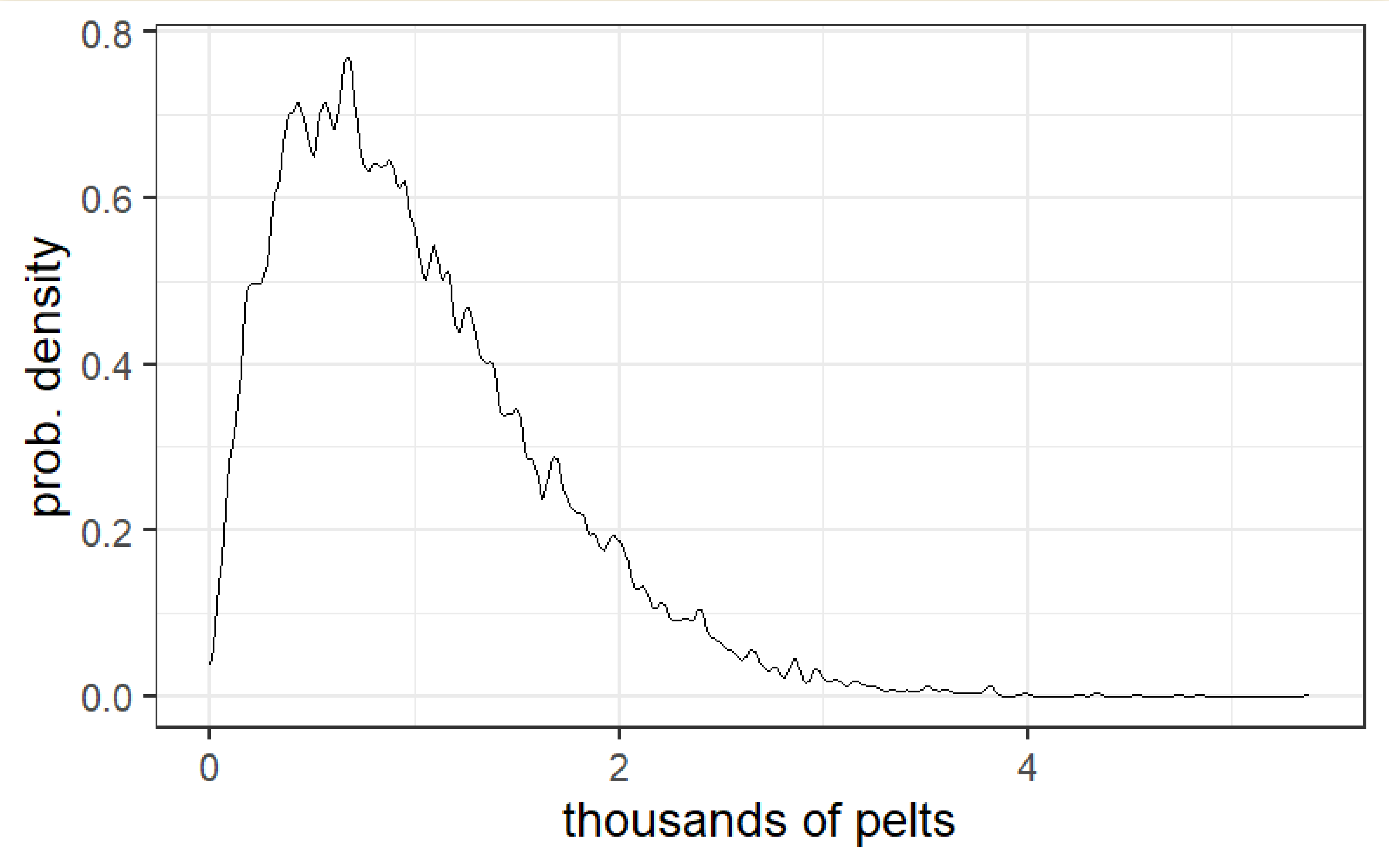
- Use Bayesian regression analysis to figure out b_H, m_H, b_L, and m_L.
- Complication: The data show the number of pelts collected by hunters, not the actual populations.
  - Estimate the population from the observations (pelts)
  - H, L are populations, h, ℓ are observations
  - Hares and lynx are trapped with some probability p_t, which varies from year to year
    - $p_t \sim \text{Beta}(2, 18)$ (mean = 0.1)
    - The reported value of h is rounded to the nearest 100, and divided by 1000

- What does this look like?

```
N <- 1e4
Ht <- 1e4

sim_hares <- tibble(
  p = rbeta(N, 2, 18),
  h = rbinom(N, size=Ht, prob=p)
) %>% mutate(h = round(h / 1000, 2))
```

```
ggplot(sim_hares, aes(x = h)) +
  geom_density(bw = 0.02) +
  labs(x = "thousands of pelts", y = "prob. density")
```



Full Statistical Model

$$
\begin{align} h_t &\sim \text{Log-Normal}(\log(p_H H_t), \sigma_H) \\ \ell_t &\sim \text{Log-Normal}(\log(p_L L_t), \sigma_L) \\ H_t &\sim \text{Log-Normal}(\log(10), 1) \\ L_t &\sim \text{Log-Normal}(\log(10), 1) \\ H_{t > 1} &= H_1 + \int_1^T H_t (b_H - m_L L_t) dt \\ L_{t > 1} &= L_1 + \int_1^T L_t (b_L H_t - m_L) dt \end{align}
$$

- Priors:

$$
\begin{align} \sigma_H &\sim \text{Exponential}(1) \\ \sigma_L &\sim \text{Exponential}(1) \\ p_H &\sim \text{Beta}(\alpha_H, \beta_H) \\ p_L &\sim \text{Beta}(\alpha_L, \beta_L) \\ b_H &\sim \text{Half-Normal}(1, 0.5) \\ b_L &\sim \text{Half-Normal}(0.05, 0.05) \\ m_H &\sim \text{Half-Normal}(0.05, 0.05) \\ m_L &\sim \text{Half-Normal}(1, 0.5) \\ \end{align}
$$

Stan Model

```
data("Lynx_Hare_model")
cat(Lynx_Hare_model)
```

```
## functions {
##   real[] dpop_dt( real t,                 // time
##                   real[] pop_init,          // initial state
{lynx, hares}
##                   real[] theta,             // parameters
##                   real[] x_r, int[] x_i) {  // unused
##     real L = pop_init[1];
##     real H = pop_init[2];
##     real bh = theta[1];
##     real mh = theta[2];
##     real ml = theta[3];
##     real bl = theta[4];
##     // differential equations
##     real dH_dt = (bh - mh * L) * H;
##     real dL_dt = (bl * H - ml) * L;
##     return { dL_dt , dH_dt };
##   }
## }
## data {
##   int<lower=0> N;              // number of measurement times
##   real<lower=0> pelts[N,2];    // measured populations
## }
## transformed data{
##   real times_measured[N-1];    // N-1 because first time is
initial state
##   for ( i in 2:N ) times_measured[i-1] = i;
## }
```

```
## parameters {
##   real<lower=0> theta[4];      // { bh, mh, ml, bl }
##   real<lower=0> pop_init[2];   // initial population state
##   real<lower=0> sigma[2];      // measurement errors
##   real<lower=0,upper=1> p[2];  // trap rate
## }
## transformed parameters {
##   real pop[N, 2];
##   pop[1,1] = pop_init[1];
##   pop[1,2] = pop_init[2];
##   pop[2:N,1:2] = integrate_ode_rk45(
##     dpop_dt, pop_init, 0, times_measured, theta,
##     rep_array(0.0, 0), rep_array(0, 0),
##     1e-5, 1e-3, 5e2);
## }
## model {
##   // priors
##   theta[{1,3}] ~ normal( 1 , 0.5 );    // bh,ml
##   theta[{2,4}] ~ normal( 0.05, 0.05 ); // mh,bl
##   sigma ~ exponential( 1 );
##   pop_init ~ lognormal( log(10) , 1 );
##   p ~ beta(40,200);
##   // observation model
##   // connect latent population state to observed pelts
##   for ( t in 1:N )
##     for ( k in 1:2 )
##       pelts[t,k] ~ lognormal( log(pop[t,k]*p[k]) , sigma[k] );
## }
## generated quantities {
##   real pelts_pred[N,2];
##   for ( t in 1:N )
##     for ( k in 1:2 )
##       pelts_pred[t,k] = lognormal_rng( log(pop[t,k]*p[k]) ,
sigma[k] );
## }
```

Apply the Model

```r
dat_list <- list(
    N = nrow(Lynx_Hare),
    pelts = Lynx_Hare[,2:3])

mdl_lh <- stan(model_code = Lynx_Hare_model, data = dat_list,
  chains = 4, cores = 4, control = list(adapt_delta = 0.95))

post <- extract.samples(mdl_lh)
pelts <- dat_list$pelts

plot(1:21, pelts[,2], pch=16, ylim=c(0, 120), xlab="year",
     ylab="thousands of pelts", xaxt="n")
at <- c(1,11,21)
axis(1, at = at, labels = Lynx_Hare$Year[at])
points(1:21, pelts[,1], col = rangi2, pch = 16)
# 21 time series from posterior
for (s in 1:21) {
  lines(1:21, post$pelts_pred[s,,2],
        col = col.alpha("black", 0.1), lwd=2)
  lines(1:21, post$pelts_pred[s,,1],
        col = col.alpha(rangi2, 0.2), lwd=2)
}
# text labels
text(17, 90, "Lepus", pos = 2)
text(19, 50, "Lynx",  pos = 2, col = rangi2)

plot(NULL, pch = 16, xlim = c(1, 21), ylim = c(0, 500),
     xlab="year", ylab = "thousands of animals", xaxt = "n")
at <- c(1,11,21)
axis(1, at = at, labels = Lynx_Hare$Year[at])
for (s in 1:21) {
    lines(1:21, post$pop[s,,2], col = col.alpha("black", 0.2),
          lwd = 2)
    lines(1:21, post$pop[s,,1], col = col.alpha(rangi2, 0.4),
          lwd = 2)
}
```