Lynx Hare Population Dynamics

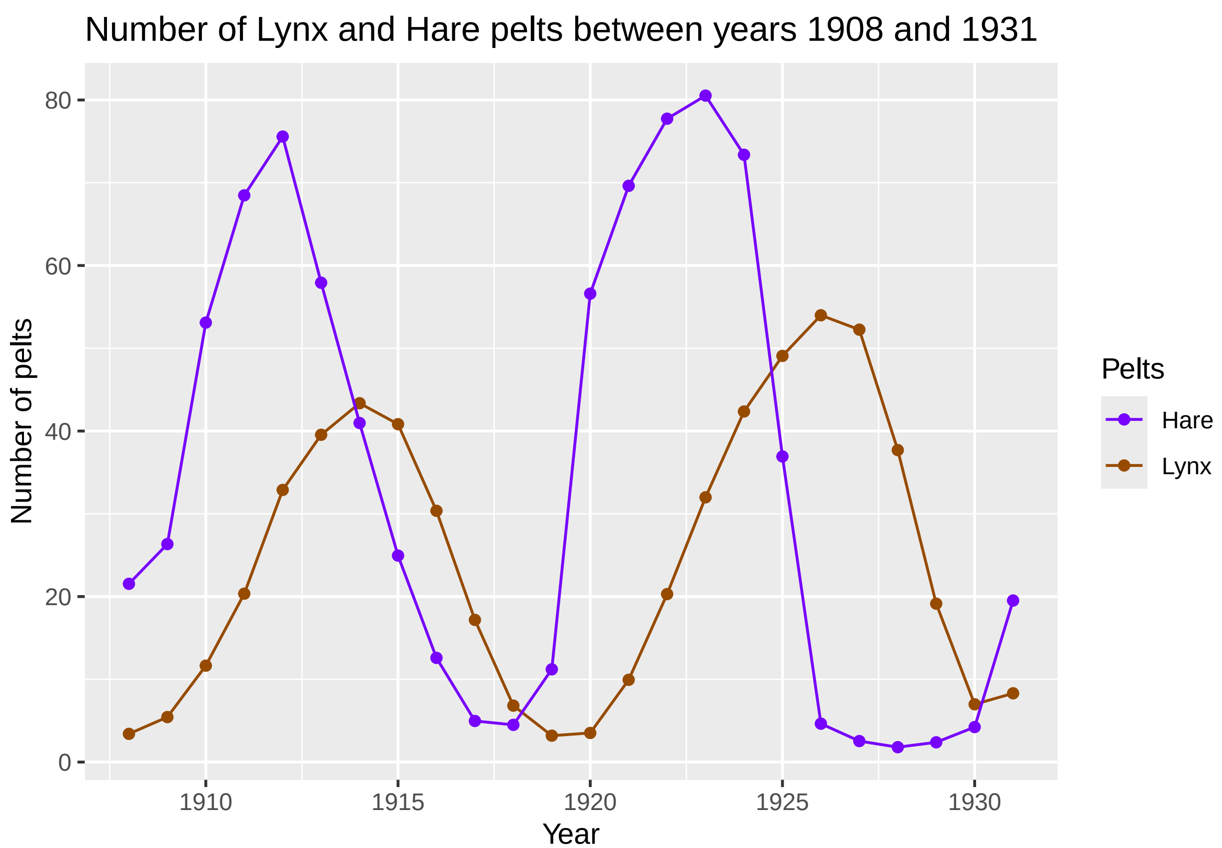
Padma Tanikella

2024-05-02

```
library(ggplot2)
g <- ggplot(df, aes(year))
g <- g + geom_line(aes(y=lynx, color="Lynx")) + geom_point(aes(y=lynx, color="Lynx"))
g <- g + geom_line(aes(y=hare, color="Hare")) + geom_point(aes(y=hare, color="Hare"))
g = g + xlab("Year") + ylab("Number of pelts")

# add legend manually using scale_color_manual
g = g +   scale_color_manual(values=c("Lynx"="#964B00", "Hare"="#7703fc"))  +
guides(color=guide_legend(title="Pelts"))

g + ggtitle("Number of Lynx and Hare pelts between years 1908 and 1931")
```



We observe a lag between the peaks of Hare pelts and Lynx pelts.

```r
## Create a function to evaluate the derivative
## lotka volterra system is defined such that it can be input for ode()
lotka <- function(t, y, parms) {
    ## split out "y"
    H=y[1] # prey popln
    L=y[2] # predator popln

    ## split out parameters in
    alpha=parms[1]
    beta=parms[2]
    gamma = parms[3]
    delta = parms[4]

    ## evaluate derivatives
    dH_dt <- alpha * H  - beta * H * L
    dL_dt <-  - gamma * L + delta * H * L
    ## return as a list
    list(c(dH_dt, dL_dt))
}
```

```r
## Initial conditions: H(0)=10, L(0)=10
init <- c(H=10,L=10)

parameters <- c(alpha = 2/3, beta = 4/3, gamma = 1, delta = 1)
## Time frame
## at what time points do we want to simulate the ODE from?
times <- seq(1900, 1940, by = 0.5)
```

Let $H(t)$ is the true hare population at time t according to the Lotka-Volterra dynamics, while h_t denotes the observed value i.e. the number of pelts. Similarly, define $L(t)$ is the "true" unobserved lynx count while l_t denotes the observed number of pelts. Then the data can be modeled as:

$$h_t \sim \mathcal{N}(H(t), 1) \quad \text{and} \quad l_t \sim \mathcal{N}(L(t), 1)$$

```r
nll.lotka=function(params,init,y,t.obs,h){
  # input y is a matrix [H.obs, L.obs]
    H.seen = y[,1]
    L.seen = y[,2]
    ## solve ODE model
    out <- ode(y = init, times = seq(1900,1940,by=h), func = lotka, parms = params)
    ## get H(t), L(t) at observation times
    out <- as.data.frame(out)
    tvals=out$time
    idx.obs.times=integer()
    for(i in 1:length(t.obs)){
        idx.obs.times[i]=max(which(tvals<=t.obs[i]))
    }
    L.obs=out$L[idx.obs.times]
    H.obs=out$H[idx.obs.times]
    ## get normal negative log-likelihood
    -sum(dnorm(x = H.seen ,mean = H.obs,log=TRUE)) -sum(dnorm(x = L.seen, mean = L.obs ,log=TRUE))
}
```

```r
nll.lotka(parameters, init,df[,2:3],1908:1931, 0.5)
```

```
## [1] 34772.5
```

```r
params.init = parameters
fit=optim(params.init,nll.lotka,init = init, y = df[,2:3], t.obs = 1908:1931 ,h=.05,hessian=TRUE)
fit
```

```
## $par
##     alpha      beta     gamma     delta
## 0.8668181 4.1017039 2.4919760 0.4889241
##
## $value
## [1] 27371.35
##
## $counts
## function gradient
##      501       NA
##
## $convergence
## [1] 1
##
## $message
## NULL
##
## $hessian
##           alpha      beta     gamma     delta
## alpha 456243289 -17596205 293161526 -74146617
## beta  -17596205  20736680 -15617287  13120228
## gamma 293161526 -15617287 104687912 -12923134
## delta -74146617  13120228 -12923134  24905335
```