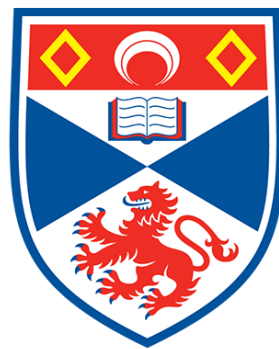


Against all odds the future of wildebeest shines bright: Bayesian SSM suggests

Misha Tseitlin || Muntasir Akash

21 November 2023

Assignment two: State-space model for wildebeest population dynamics



University of
St Andrews

School of Mathematics and Statistics

in partial fulfilment of the requirements for
MT5767: Modelling Wildlife Population Dynamics

Contents

[Introduction](#) 1

[Methodology](#) 1

[Results and Discussion](#) 2

[Authors' contribution](#) 3

[Appendix](#) 4

[Code Supplement](#) 5

[References](#) 8

Introduction

The wildebeest (*Connochaetes*) lives and moves in large herds (Estes 2014). The blue wildebeest (*Connochaetes taurinus*) migrate in a clockwise manner from Serengeti, Tanzania to Masai Mara, Kenya, forming the backbone of the Great Migration in south-eastern Africa (Estes 2014). During this annual crossing of great distances, the blue wildebeest follow the annual rain cycles and move along the trajectory. The species is also yearly breeder that depends on the rainfall (Estes 2014).

Due to expansion of agricultural practices, Rinderpest induced by the spread of cattle-farming, and poaching pressure, population of this antelope species faced a steep decline in the 60s and 70s. Since then, population monitoring has been a key focus of biologists and conservationists.

This report used 1960–1989 measures of wildebeest populations (sourced from Sutherland 2023) and undertook a Bayesian approach of state-space model to understand on population size and growth in the species including the years where no survey was carried out and project on its population into five years in future.

Methodology

State-space model A state-space model (SSM) consists of two components: an unobserved “true” population process, N_t , and the observation process, y_t , that is often entailed with observation process errors and uncertainties. Here, true wildebeest population N_t depends on the population growth rate, λ_t , that has been accounted for yearly poaching rate, c_t . To account for random variation common in natural systems, large N_{t+1} s take a normal distribution consistent with the thousands of wildebeest observed.

$$N_{t+1} \sim N(\lambda_t * (N_t - c_t), \sigma_t^2)$$

λ_t , i.e., $\frac{N_{t+1}}{N_t}$, depends on rainfall R_t as suggested by Estes (2014). To test this relation, β_0 and β_1 estimate $\log(\lambda_t)$, a logarithmic transformation assisting in to (i) require positive λ values, (ii) allow estimation of the per-capita growth rate $r_t = \log_e(\lambda_t) = \frac{\Delta N}{N \Delta t}$, and (iii) more easily interpret the relationship with β s.

$$\log(\lambda_t) \sim \beta_0 + \beta_1 * R_t$$

Finally, y_t is treated as unbiased accounted with associated known spread of measures se_t ; thus, expressed as a simple stochastic normal distribution linked to underlying true state N_t . This relationship can be expressed as

$$y_t \sim N(N_t, se_t^2)$$

Bayesian analysis of SSM The Bayesian SSM was carried out in RStudio Environment (RStudio 2023) using a JAGS user interface package (Kellner 2021). Bayesian analysis is an approach of parameter estimation where available knowledge about parameters in a statistical model, termed as prior distribution $p(\theta)$, is updated with observed data, a conditional probability, in a form of a likelihood function, $p(y|\theta)$ to determine a posterior distribution, $p(\theta|y)$, of parameters (Schout et al. 2021). In contrast to the frequentist approach where parameter estimates are described as point estimates, Bayesian analysis focuses on describing parameters as posterior distribution (Kéry and Schaub 2011). The Bayes’ rule is

$$p(\theta|y) = \frac{p(y|\theta) * p(\theta)}{p(y)}$$

$p(y)$, a normalizing factor and non-dependent of any parameter, θ , is thus can be dropped (Schout et al. 2021). So, posterior distribution of parameters becomes proportional to a joint probability

distribution, i.e., the product of prior distribution of parameters and observed data in likelihood function,

$$p(\theta|y) \sim p(y|\theta) * p(\theta)$$

Under this backdrop, for wildebeest data analysis, the state process, N_{t+1} , was considered as the prior and the observation process, y_t , as likelihood. For N_{t+1} , we dealt with the following distributions:

$$\begin{aligned} \log(\lambda_t) &= \beta_0 + \beta_1 * R_t \\ N_{t+1} &\sim N(\lambda_t * (N_t - c_t), \text{sigma}_t^2) \end{aligned}$$

$\log(\lambda_t)$ was a deterministic node in Kellner (2021) depending on β_0 , and β_1 ; whereas N_{t+1} a stochastic node. This settings left us with the following prior distributions,

$$\begin{aligned} \beta_0 &\sim \text{dnorm}(0, 1000) \\ \beta_1 &\sim \text{dnorm}(0, 1000) \\ \text{sigma}_t^2 &\sim \text{dunif}(0, 1) \end{aligned}$$

For the likelihood, we dealt with the following distribution,

$$y_t \sim N(N_t, \text{se}_t^2)$$

For all variance terms (i.e., in β_0 and β_1 , and sigma_t^2 , and se_t^2), precision, $\tau [= \frac{1}{\sigma^2}]$ was used in the analytic settings.

Model run and evaluation All six steps of a Bayesian analysis i.e., writing models in JAGS sampler, packaging JAGS data, setting initial values, defining parameters of interest, putting Markov Chain Monte Carlo (MCMC) values, and posterior predictive checking were sequentially carried out. Our priors of interests were β_0 and β_1 , and sigma_t^2 , and our latent variables were growth rate and estimated population size (in million). Markov chain is a iterative process where a value in $t + 1$ step is only related to the value of t step. Monte Carlo is a stochastic simulation procedure to determine integrals by simulating random numbers from a given distribution. In MCMC settings, we ran three different chains. To deal with the inherent autocorrelation issues, we dropped the every sixth value in the MCMC run. Our total iteration was 100000 of which 10000 was burn-ins. We checked the posterior predictions using traceplots to detect convergence visually supported with Gelman-Rubin \hat{R} statistic. We did not do prior predictive check as observations were assumed as unbiased and provided with known observation error.

R packages To deal with NA values (i.e., years without survey), we used Zeileis and Grothendieck (2005) for back-filling. However, we used years with observation only in the likelihood settings to predict growth rate and population size for these NA values. Youngflesh (2018) was followed for MCMC checks. Wickham (2016) was used for graphs.

Results and Discussion

Without looking at the model output, a clear decrease in animal density occurs over the survey period (Figure ??). The GLS model confirms these results at a high precision (Table ??)—the probability of no impact from construction is ~0% given the data—with 0.009 fewer animals per km^2 after construction. Within the survey area, a total drop of 213 animals can be linked to wind turbine construction. The remaining three variables were all highly significant (less than 3% probability of seeing our data if there was no relationship with), but their effects on animal density were smaller than windfarm construction. Having attempted several linear and GLS models, Table ??’s model ultimately shows the lowest error and best-suited information criteria.

Authors' contribution

Appendix

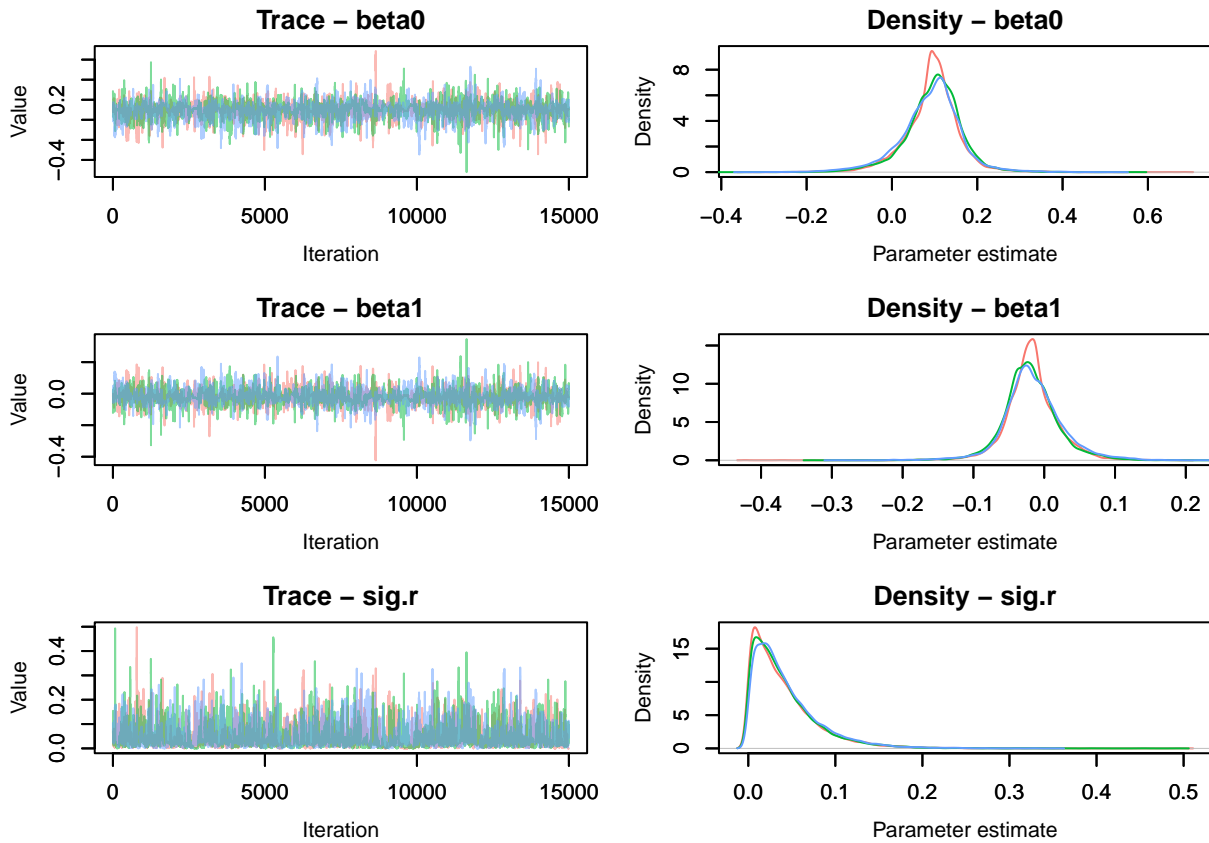


Figure 1: Latent parameter trace plots; derived hierarchical parameters not shown but also converged

	mean	sd	2.5%	50%	97.5%	Rhat	n.eff
beta0	0.0942088	0.0679976	-0.0581806	0.0993709	0.2201225	1.01	911
beta1	-0.0172110	0.0398870	-0.0948268	-0.0192961	0.0693934	1.01	1040
sig.r	0.0418267	0.0385407	0.0012515	0.0315144	0.1422859	1.00	1211

```
## To cite package 'statsecol' in publications use:
##
##   Sutherland C (2023). _statsecol: A Package Associated With The
##   Statistical Ecology MSc_. R package version 0.1.0.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {statsecol: A Package Associated With The Statistical Ecology MSc},
##     author = {Chris Sutherland},
##     year = {2023},
##     note = {R package version 0.1.0},
##   }
##
## ATTENTION: This citation information has been auto-generated from the
## package DESCRIPTION file and may need manual editing, see
## 'help("citation")'.
```

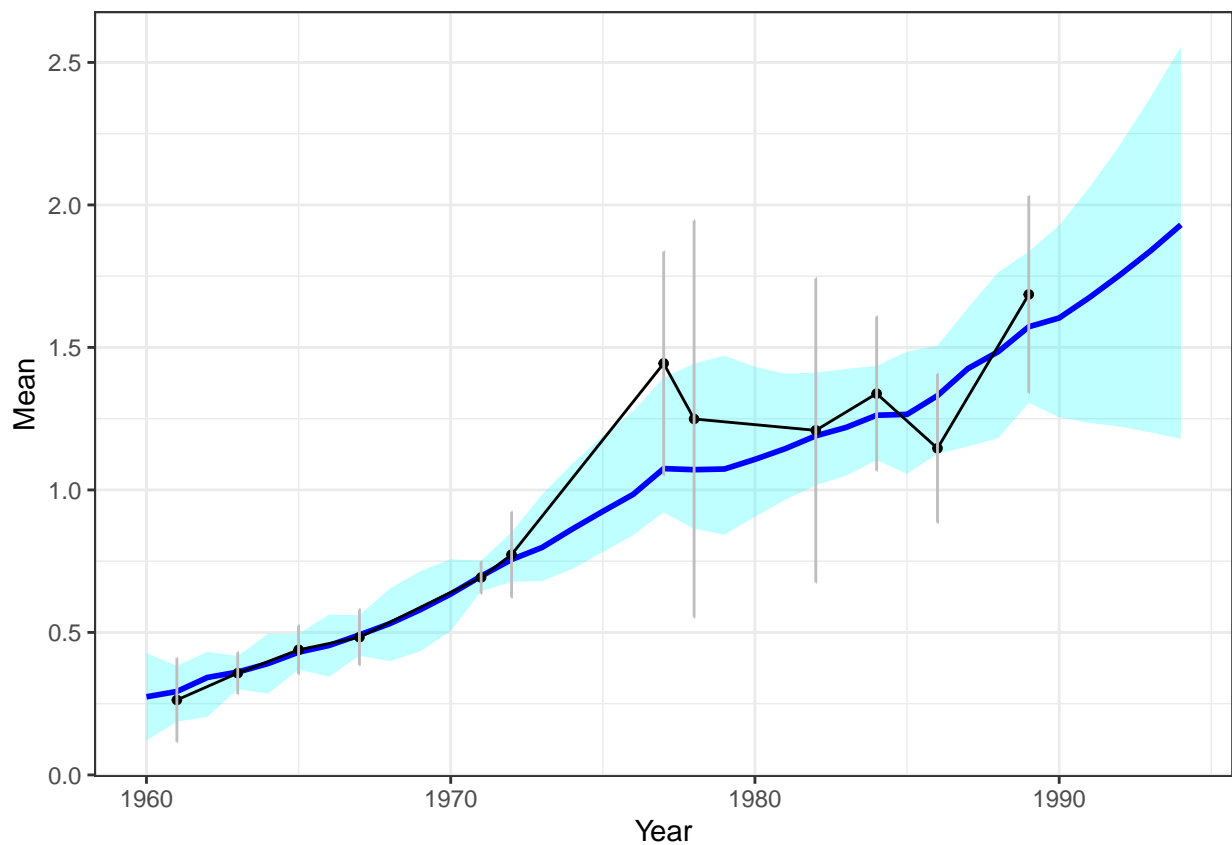


Figure 2: Projected values for true and estimated wildebeest population sizes

Code Supplement

```
# NO ~ uniform(0, U)

# Nt | Nt-1 ~ normal [lambda*(Nt-1 - ct-1), sigmaN]

# yt | Nt ~ normal (Nt, sigmaY)

# writing the model in BUGS

# model specification

cat("
model{
  # priors
  # this prior is quite irrelevant to the final data spread: tested between 0.5 and 3
  n1 ~ dunif(0,0.7)  # 0.7 is the most suitable option theoretorically
  N.est[1] <- n1
  beta0 ~ dnorm(0,0.001)
  beta1 ~ dnorm(0,0.001)
  sig.r ~ dunif(0, 1)
  sig2.r <- pow(sig.r, 2)
  tau.r <- pow(sig.r, -2)
```

```

# likelihood - state process
for(t in 1:(nyrs-1)){
  log.lambda[t] <- beta0 + beta1*R[t]
  log(lambda[t]) <- log.lambda[t]
  N.est[t+1] ~ dnorm(lambda[t]*(N.est[t] - c[t]), tau.r)
}

# likelihood - observation process
for (t in validYrs) {
  y[t] ~ dnorm(N.est[t], obs.tau[t])
}

}
",fill = TRUE, file='wildessmBasic1.txt')

# JAGS package data

wilddata <- list(y = wildebeest$Nhat,

                nyrs = nrow(wildebeest),

                validYrs = validObs,

                R = wildebeest$rain,

                obs.tau = wildebeest$sehat^-2,

                c = wildebeest$Catch)

# set initial values for the unknown parameters

wildeinits <- function(){

  list(beta0 = rnorm(1),

        beta1 = rnorm(1),

        sig.r = runif(1),

        N = wildebeestImpute$Nhat)

    #N.est = wildebeestImpute$Nhat*1000)
}

# parameters monitoring '

wildeparms <- c("beta0", "beta1", "sig.r", "lambda", "N.est")

# MCMC settings

nt <- 6 # thinning rate to reduce autocorrelation

nc <- 3 # number of chains

```



```

ni <- 100000 # number of iteration

nb <- 10000 # number of burn-ins / warm-ups

wildeout1 <- jags(data = wilddata,
  inits = wildeinits,
  parameters.to.save = wildeparms,
  model.file = "wildessmBasic1.txt",
  n.chains = nc,
  n.iter = ni,
  n.burnin = nb,
  n.thin = nt)

wilde_traj1 <- data.frame(Year = wildebeest$year,
  Mean = wildeout1$mean$N.est,
  Lower = wildeout1$q2.5$N.est,
  Upper = wildeout1$q97.5$N.est,
  Obs = wildebeest$Nhat,
  LowerObs = wildebeest$lci,
  UpperObs = wildebeest$uci)

ggplot(data = wilde_traj1) +
  geom_ribbon(aes(x=Year, y=Mean, ymin=Lower, ymax=Upper),
    fill="cyan", alpha = 0.25) +
  geom_line(aes(x=Year, y=Mean), linewidth=1, color="blue") +
  geom_point(aes(x=Year, y=Obs), size=1.2) +
  geom_line(data = na.omit(wilde_traj1), aes(x=Year, y=Obs)) +
  geom_errorbar(aes(x=Year,
    y=Obs,
    ymin=LowerObs,
    ymax=UpperObs), width=0, color="grey") +
  theme_bw()

#project values forward 5 years from 1990-1994
nproj <- 5

#assume that illegal harvesting continues at current levels
#use average observed rainfall for future projections
#impute last year values for Nhat and sehat; they aren't referenced though
wilddata_proj1 <- list(y = c(wildebeest$Nhat, rep(wildebeest$Nhat[nrow(wildebeest)], nproj)),
  nyrs = nrow(wildebeest) + nproj,
  validYrs = validObs,
  R = c(wildebeest$rain, rep(mean(wildebeest$rain), nproj)),
  obs.tau = c(wildebeest$sehat^-2, rep(wildebeest$sehat[nrow(wildebeest)], nproj)),
  c = c(wildebeest$Catch, rep(wildebeest$Catch[nrow(wildebeest)], nproj))

wildeproj1 <- jags(data = wilddata_proj1,
  inits = wildeinits,
  parameters.to.save = wildeparms,

```

```

        model.file = "wildessmBasic1.txt",
        n.chains = nc,
        n.iter = ni,
        n.burnin = nb,
        n.thin = nt)

wilde_proj1 <- data.frame(Year = c(wildebeest$year, 1990:1994),
                          Mean = wildeproj1$mean$N.est,
                          Lower = wildeproj1$q2.5$N.est,
                          Upper = wildeproj1$q97.5$N.est,
                          Obs = c(wildebeest$Nhat, rep(NA, nproj)),
                          LowerObs = c(wildebeest$lci, rep(NA, nproj)),
                          UpperObs = c(wildebeest$uci, rep(NA, nproj)))

ggplot(data = wilde_proj1) +
  geom_ribbon(aes(x=Year, y=Mean, ymin=Lower, ymax=Upper),
            fill="cyan", alpha = 0.25) +
  geom_line(aes(x=Year, y=Mean), linewidth=1, color="blue") +
  geom_point(aes(x=Year, y=Obs), size=1.2) +
  geom_line(data = na.omit(wilde_proj1), aes(x=Year, y=Obs)) +
  geom_errorbar(aes(x=Year,
                    y=Obs,
                    ymin=LowerObs,
                    ymax=UpperObs), width=0, color="grey") +
  theme_bw()

```

References

- Estes, Richard D. 2014. *The Gnu's World: Serengeti Wildebeest Ecology and Life History*. University of California Press.
- Kellner, Ken. 2021. *jagsUI: A Wrapper Around 'Rjags' to Streamline 'JAGS' Analyses*. <https://CRAN.R-project.org/package=jagsUI>.
- Kéry, Marc, and Michael Schaub. 2011. *Bayesian Population Analysis Using WinBUGS: A Hierarchical Perspective*. Academic Press.
- RStudio. 2023. *RStudio: Integrated Development Environment for r*. Posit Software, PBC, Boston, MA: R Foundation for Statistical Computing. <http://www.posit.co/>.
- Schoot, Rens van de, Sarah Depaoli, Ruth King, Bianca Kramer, Kaspar Mörtens, Mahlet G Tadesse, Marina Vannucci, et al. 2021. "Bayesian Statistics and Modelling." *Nature Reviews Methods Primers* 1 (1): 1.
- Sutherland, Chris. 2023. *Statsecol: A Package Associated with the Statistical Ecology MSc*.
- Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. <https://ggplot2.tidyverse.org>.
- Youngflesh, Casey. 2018. "MCMCvis: Tools to Visualize, Manipulate, and Summarize MCMC Output." *Journal of Open Source Software* 3 (24): 640. <https://doi.org/10.21105/joss.00640>.
- Zeileis, Achim, and Gabor Grothendieck. 2005. "Zoo: S3 Infrastructure for Regular and Irregular Time Series." *Journal of Statistical Software* 14 (6): 1–27. <https://doi.org/10.18637/jss.v014.i06>.