Estimating the effect of oil drilling on mule deer population dynamics in North Dakota using a state-space model

R Projects

University of Freiburg
Faculty of Environment and Natural Resources
Simon Schulte 3320165 | Margret Rattay 3505926
Lecturers: Prof. Carsten Dormann, Dr. Simone Ciuti

16. December, 2016

Contents

1 Introduction

Energy production out of oil and gas is always associated with human impacts on nature. In the Missouri badlands in North Dakota the count of oil wells doubled within the last twenty years. This led to substantial landscape changes in mule deer habitats and implies changes of the mortality rate due to animals killed by traffic (Ciuti et al. 2014). With animal count data from the last fifty years we want to examine the effects of this human intervention on mule deer population dynamics.

The problem which frequently arises with observed data is that we only have a count number with some unknown observation errors associated (Kéry and Schaub 2012). With the approach of a state-space model (SSM) for this error can be accounted. A SSM works from the coupling of two models (Rivot et al. 2004). The first model (process model) mimics the true but unknown development of a state, which is the population dynamics free of observation error. The second model describes the observation process with the probabilistic dependence between the latent state variable and the observed measurement (Kéry and Schaub 2012).

The purpose of this paper is to use a SSM to describe the population dynamics of mule deer in North Dakota and identify the predictors which are important to do this. Moreover we want to find out, whether the occurrence of more and more oil wells acts as a driving parameter.

2 Material and Methods

2.1 Data

Mule deer population has been observed in 26 different sites in North Dakota since 1956 (Ciuti et al. 2014). We use all aerial observations made each spring from 1959 to 2012 as for this period of time we have continuous data concerning climate, abundance of predators, hunting and oil drilling in the area. Due to long computing time and the fact that about 27% of the observations are missing we aggregate the time series over all study sites. This is done by computing the mean density of animals per square kilometre for every year (ignoring missing values) and multiply it with the sum of all areas. Next to the observation data we include the following predictors into our model:

- *min_temp*: average minimum temperature in winter (November to March)
- *coy*: abundance of coyotes
- *hunt*: hunting in the previous autumn
- *oil_wells*: the number of active oil wells within the study areas

2.2 The state-space model

In order to determine the impact of oil drilling on the mule deer population we define two models which differ in their state process: Model 1 (M1) **without** and model 2 (M2) **with** the number of active oil wells within the study areas (*oil_wells*) as predictor.

Process model In both models we use a logistic state process model. As we have a stochastic model we adopt the logistic equation on the log scale (Lande, Engen, and Saether 2003):

$$\log(N_{t+1}) = \log(N_t) + \log(1 + r \cdot N_t \cdot (1 - N_t/K) \tag{1}$$

where r represents the growth rate and K the carrying capacity. The predictors are included in the form of linear and quadratic effects. The predictor *hunt* is included as additional term into equation 1 assuming that the main effect of hunting on the population is independent of the natural population dynamics:

$$\log N_{t+1} = \log N_t + \log(1 + r \cdot N_t \cdot (1 - N_t/K) + h \cdot hunt_t + h_2 \cdot hunt_t^2) \tag{2}$$

The growth rate r differs in both models. In M1 we only include min_temp and coy. Both are supposed to influence the growth rate r of the process model. Thus, for M1 we get the following equation of r_t^1:

$$r_t^1 = \lambda_1 + c_1 \cdot min\_temp_t + c2_1 \cdot min\_temp_t^2 + co_1 \cdot coy_t + co2_1 \cdot coy_t^2 \qquad (4.1)$$

M2 additionally includes oil_wells which is considered to affect mainly the growth rate r_t because of the deer killed in road accidents. Thus, for M2 we have the following equation of r_t^2:

$$r_t^2 = \lambda_2 + ... + o_2 \cdot oil\_wells_t + o_2 \cdot oil\_wells_t^2 \qquad (4.2)$$

Observation model: The observation model is common for M1 and M2. Here we use a model which allows us to add directed noise on our observations. Therefore, we take a normal distribution in the form of

$$y_t \sim \mathcal{N}(\log N.est_t + \log(\beta), \tau) \qquad (5)$$

where β represents the proportion of animals spotted and τ the precision parameter of the distribution with $\tau = 1/\sigma^2$ and standard deviation σ.

2.3 Calibration

We calibrated the parameters using the Bayesian Markov-Chain-Monte-Carlo method (MCMC) from JAGS implemented in R by the R2jags package. For all parameters except K and β we use uninformative priors due to the lack of further information (annex, table 1). K is given a uniform distribution between 100 and 8000. As β is supposed to range in the area of $\sim$ [0.5, 0.9] we define a beta prior distribution of $\beta \sim \mathcal{B}(16, 8)$.

We iterate 200.000 times with three Markov-Chains and a thinning factor of 100. Convergence of the chains is checked both, visually and by the Gelman and Ruben diagnostics (Brooks and Gelman 1998).

Parameters are assumed to have significant impact when the 95% quantile of their posterior distribution does not overlap zero. In order to evaluate the models we simulate 20.000 new time series with each model and compare these simulated data with the observations.

3 Results

Both models describe the overall trend of the population dynamics of the mule deer population well (figure 1). However, the fluctuations which occur every three to four years are not covered properly by none of the models.

The predictors showing significant effects in M1 are co_1 and $h2_1$ (both positively affecting the growth rate) and c_1 and h_1 (negatively affecting the growth rate). Furthermore, both parameters of the observation model of M1 $tau.obs_1$ and β_1 are significant (annex, table 2). In the case of M2 the growth rate is significantly affected by co_2 (positively) and h_2 (negatively). Also o_2, the linear effect of *oil_wells* shows a significant negative effect on the growth rate. Like in M1, $tau.obs_2$ and β_2 are significant (annex, table 3).

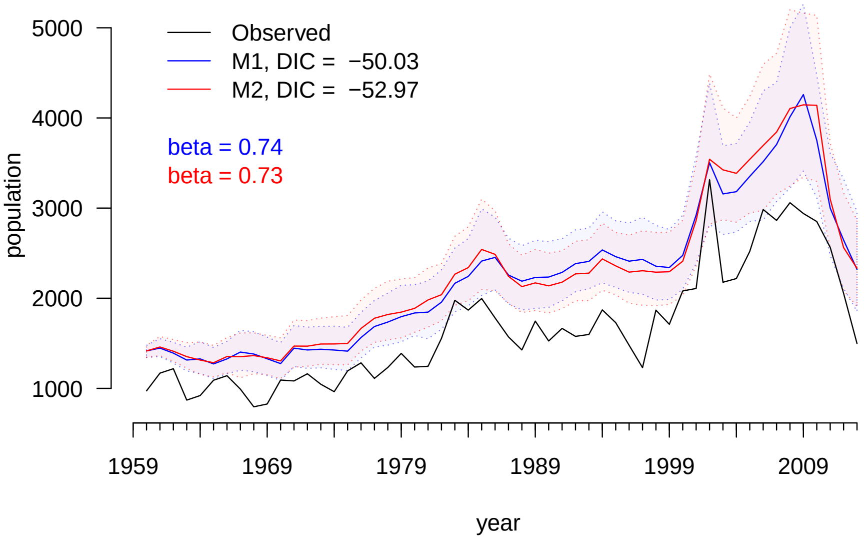


Figure 1: Observed number of mule deer (black) and estimated count of animals by model 1 (blue) and model 2 (red).

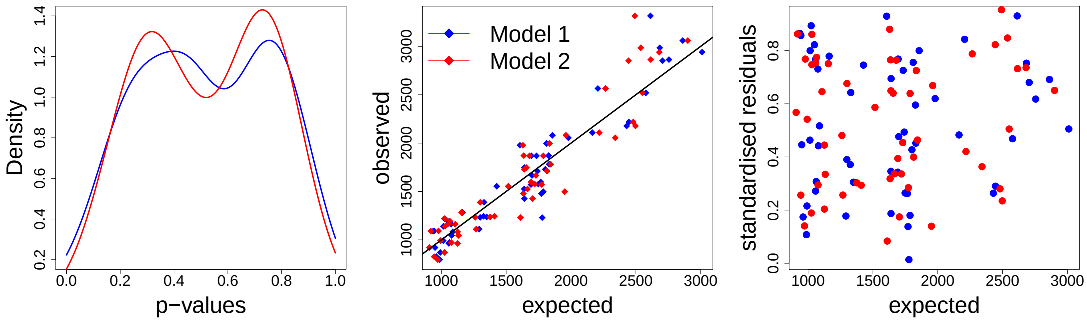


Figure 2: Left: Bayesian p-values of observation. Middle: Expected against observed counts. Right: Standardized residuals.

Comparing the model performance of M1, which describes the dynamics of mule deer population without the influence of oil wells and M2, which takes the number of oil wells into account does not show a big difference. Only the deviance information criterion (DIC) indicates that M2 (DIC = -52.97) describes the data better than M1 (DIC = -50.03).

Comparison our simulated time series with the observations reveals that around the quantiles 0.2 and 0.8 the observations are more or less evenly distributed (figure 2, left). However, both models do not cover extreme values, both high and low. The residuals show a slight underestimation of high values, which is - in the case of M2 - slightly more distinct (figure 2, middle and right).

4 Discussion and Conclusion

There are two results from which we can derive an effect of active oil drilling on mule deer population in the study area: First, including the oil wells in our model slightly improves the model performance (DIC M1 = -50.03 vs. DIC M2 = -52.97). Second, the linear effect o on the growth rate of M2 is significant. Nevertheless there are several points to discuss.

The fluctuation in population dynamics which occur every three to five years (figure 1) and which is not covered by neither model could indicate that we missed some important predictors in the model.

Furthermore we decided to include the oil wells as term that affects the growth rate (equation 4.2). By doing this we gave priority to the fact that an increasing number of oil wells influences the mortality. At the same time, however, this form of human intervention plays an important role for the habitat and thus influences the carrying capacity. Including predictors affecting the carrying capacity is an important part that was beyond the scope of this analysis.

The uncertainty of our models covers the observation of mule deer counts (observation error) and parameter (in form of the posterior distribution) but not the boundary conditions (observation error of predictors) and the structural uncertainty (model equations). With having this in mind the validity decreases.

Besides we decided to give uninformative priors to all parameters due to the fact that we do not have knowledge to specify them (annex, table 1). However, by doing this we lost the ability of ecological interpretation to some degree. The coyote for instance has a positive influence on the growth rate in both models, what would not have been expected when thinking of a predator. Now, we assume, either the causal connection is the other way round (the more prey the more predator) or the coyote acts more like an indicator for the goodness of the habitat.

Moreover we have a problem in our model structure. The MCMC sampler estimates the value for K inconsistent with our expectations. In both models the value for K is with $K_1 = 1624$ and $K_2 = 1567$ in two-thirds of our time series under the estimated number of mule deer. This fact is incompatible with our understanding of the carrying capacity as limit of the habitat. In our case a negative sign of the term $1 - N/K$ forces the growth rate also to be lower than zero in order to predict positive growth of the population. Our efforts to force K to be bigger with a corresponding prior led to a model that did not converge. We did not find a working solution for this problem.

In summary it can be said, that admittedly there is an effect of active oil drilling on the population dynamics detectable but on the other hand there are many things, which have to be taken into account that we cannot take a final and founded position.

References

Brooks, Stephen P and Andrew Gelman (1998). “General methods for monitoring convergence of iterative simulations”. In: *Journal of computational and graphical statistics* 7.4. bibtex: brooks1998general, pp. 434–455.

Ciuti, Simone, William F. Jensen, Scott E. Nielsen, and Mark S. Boyce (2014). “An evaluation of historical mule deer fawn recruitment in North Dakota - Final report”. In:

Kéry, Marc and Michael Schaub (2012). *Bayesian population analysis using WinBUGS: a hierarchical perspective.* bibtex: kery2012bayesian. Academic Press.

Lande, Russell, Steinar Engen, and Bernt-Erik Saether (2003). *Stochastic population dynamics in ecology and conservation.* bibtex: lande2003stochastic. Oxford University Press on Demand.

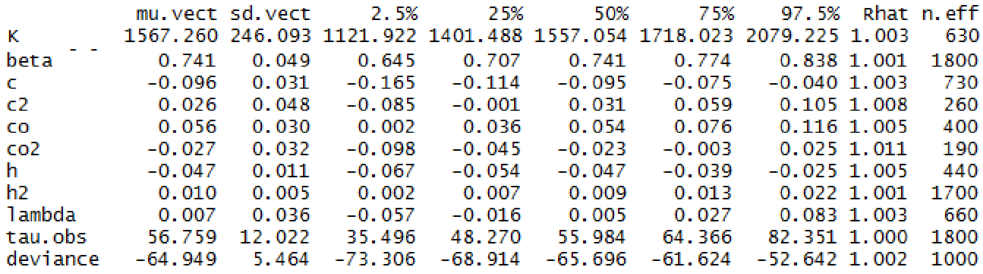
Rivot, E, E Prévost, E Parent, and JL Bagliniere (2004). “A Bayesian state-space modelling framework for fitting a salmon stage-structured population dynamic model to multiple time series of field data”. In: *Ecological Modelling* 179.4. bibtex: rivot2004bayesian, pp. 463–485.

Annex

Table 1: Description, prior and initial value for the parameters used in the two models.

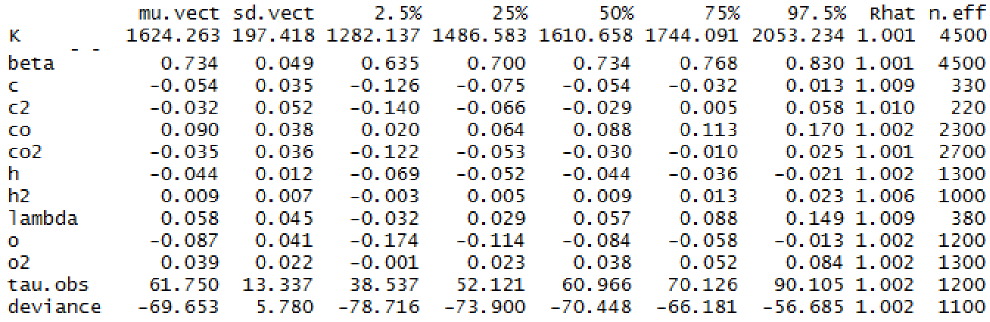
name	description	prior	initial value
K	Carrying capacity	dunif(100, 8000)	1500
β	observation rate	dbeta(16, 8)	0.7
c	linear effect of *min_temp*	dnorm(0, 0.1)	0
c2	quadratic effect of *min_temp*	dnorm(0, 0.1)	0
co	linear effect of *oil_wells*	dnorm(0, 0.1)	0
co2	quadratic effect of *coy*	dnorm(0, 0.1)	0
h	linear effect of *hunt*	dnorm(0, 0.1)	0
h2	quadratic effect of *hunt*	dnorm(0, 0.1)	0
λ	growth rate	dnorm(0, 0.1)	rnorm(1,0,0.1)
o	linear effect of *oil_wells*	dnorm(0, 0.1)	0
o2	quadratic effect of *oil_wells*	dnorm(0, 0.1)	0
tau.obs	precision parameter..	dgamma(0.01, 0.01)	rgamma(1,0.01,0.01)

Table 2: Summary of the parameter of Model 1.



```
          mu.vect sd.vect     2.5%      25%      50%      75%    97.5%  Rhat n.eff
K        1567.260 246.093 1121.922 1401.488 1557.054 1718.023 2079.225 1.003   630
beta        0.741   0.049    0.645    0.707    0.741    0.774    0.838 1.001  1800
c          -0.096   0.031   -0.165   -0.114   -0.095   -0.075   -0.040 1.003   730
c2          0.026   0.048   -0.085   -0.001    0.031    0.059    0.105 1.008   260
co          0.056   0.030    0.002    0.036    0.054    0.076    0.116 1.005   400
co2        -0.027   0.032   -0.098   -0.045   -0.023   -0.003    0.025 1.011   190
h          -0.047   0.011   -0.067   -0.054   -0.047   -0.039   -0.025 1.005   440
h2          0.010   0.005    0.002    0.007    0.009    0.013    0.022 1.001  1700
lambda      0.007   0.036   -0.057   -0.016    0.005    0.027    0.083 1.003   660
tau.obs    56.759  12.022   35.496   48.270   55.984   64.366   82.351 1.000  1800
deviance  -64.949   5.464  -73.306  -68.914  -65.696  -61.624  -52.642 1.002  1000
```

Table 3: Summary of the parameter of Model 2.



```
          mu.vect sd.vect     2.5%      25%      50%      75%    97.5%  Rhat n.eff
K        1624.263 197.418 1282.137 1486.583 1610.658 1744.091 2053.234 1.001  4500
beta        0.734   0.049    0.635    0.700    0.734    0.768    0.830 1.001  4500
c          -0.054   0.035   -0.126   -0.075   -0.054   -0.032    0.013 1.009   330
c2         -0.032   0.052   -0.140   -0.066   -0.029    0.005    0.058 1.010   220
co          0.090   0.038    0.020    0.064    0.088    0.113    0.170 1.002  2300
co2        -0.035   0.036   -0.122   -0.053   -0.030   -0.010    0.025 1.001  2700
h          -0.044   0.012   -0.069   -0.052   -0.044   -0.036   -0.021 1.002  1300
h2          0.009   0.007   -0.003    0.005    0.009    0.013    0.023 1.006  1000
lambda      0.058   0.045   -0.032    0.029    0.057    0.088    0.149 1.009   380
o          -0.087   0.041   -0.174   -0.114   -0.084   -0.058   -0.013 1.002  1200
o2          0.039   0.022   -0.001    0.023    0.038    0.052    0.084 1.002  1300
tau.obs    61.750  13.337   38.537   52.121   60.966   70.126   90.105 1.002  1200
deviance  -69.653   5.780  -78.716  -73.900  -70.448  -66.181  -56.685 1.002  1100
```

Estimating the effect of oil drilling on mule deer population dynamics in North Dakota using a State-Space model - R Code

Margret Rattay, Simon Schulte

Load packages

```r
library(R2jags)
library(reshape)
library(abind)
```

Functions

Plotfunction to visualize the results of the MCMC

```r
# function taken from 'Bayesian population analysis using WinBUGS', Marc
# K?ry and Michael Schaub 2012

graph.ssm <- function(ssm, y) {
    fitted <- lower <- upper <- numeric()
    n.years <- length(y)
    for (i in 1:n.years) {
        fitted[i] <- mean(ssm$BUGSoutput$sims.list$N.est[, i])
        lower[i] <- quantile(ssm$BUGSoutput$sims.list$N.est[, i], 0.025)
        upper[i] <- quantile(ssm$BUGSoutput$sims.list$N.est[, i], 0.975)
    }

    m1 <- min(c(y, fitted, lower))
    m2 <- max(c(y, fitted, upper))
    par(mar = c(4.5, 4, 1, 1), cex = 1.2)
    plot(0, 0, ylim = c(m1, m2), xlim = c(0.5, n.years), ylab = "population",
        xlab = "year", frame = F, axe = F)
    axis(2, las = 1)
    axis(1, at = seq(0, n.years, 5), labels = seq(0, n.years, 5))
    axis(1, at = 0:n.years, labels = rep("", n.years + 1), tcl = -0.25)
    polygon(x = c(1:n.years, n.years:1), y = c(lower, upper[n.years:1]), col = "gray90",
        border = "gray90")
    points(y, type = "l")
    points(fitted, type = "l", col = "blue")
    legend(x = 1, y = m2, legend = c("Observed", "Estimated"), lty = c(1, 1,
        1), col = c("black", "blue"), bty = "n", cex = 1)

}
```

Plotfunction to visualize two different models in one plot

```r
# function modified from 'Bayesian population analysis using WinBUGS', Marc
# K?ry and Michael Schaub 2012
graph.ssm_2 <- function(ssm, ssmNr2, y) {
    fitted <- lower <- upper <- numeric()
    n.years <- length(y)
    for (i in 1:n.years) {
        fitted[i] <- mean(ssm$BUGSoutput$sims.list$N.est[, i])
        lower[i] <- quantile(ssm$BUGSoutput$sims.list$N.est[, i], 0.025)
        upper[i] <- quantile(ssm$BUGSoutput$sims.list$N.est[, i], 0.975)
    }

    fitted2 <- lower2 <- upper2 <- numeric()
    for (i in 1:n.years) {
        fitted2[i] <- mean(ssmNr2$BUGSoutput$sims.list$N.est[, i])
        lower2[i] <- quantile(ssmNr2$BUGSoutput$sims.list$N.est[, i], 0.025)
        upper2[i] <- quantile(ssmNr2$BUGSoutput$sims.list$N.est[, i], 0.975)
    }

    m1 <- min(c(y, fitted, lower))
    m2 <- max(c(y, fitted, upper))
    par(mar = c(4.5, 5, 1, 1), cex = 1.2)
    plot(0, 0, ylim = c(m1, m2), xlim = c(0.5, n.years), ylab = "population",
        xlab = "year", frame = F, axe = F)
    axis(2, las = 2)
    axis(1, at = seq(0, n.years, 5), labels = seq(1959, 2012, 5))
    axis(1, at = 0:n.years, labels = rep("", n.years + 1), tcl = -0.25)
    # abline(a=ssm$BUGSoutput$mean$K, b=0, col='blue', lty=5)
    # abline(a=ssmNr2$BUGSoutput$mean$K, b=0, col='red', lty=5) polygon(x =
    # c(1:n.years, n.years:1), y= c(lower, upper[n.years:1]), col = 'gray90',
    # border = 'gray90') polygon(x = c(1:n.years, n.years:1), y= c(lower2,
    # upper2[n.years:1]), col = rgb(1, 0, 0,0.1), border = 'gray90')
    polygon(x = c(1:n.years, n.years:1), y = c(lower, upper[n.years:1]), col = rgb(0,
        0, 1, 0.04), border = rgb(0, 0, 1, 0.5), lty = 3)
    polygon(x = c(1:n.years, n.years:1), y = c(lower2, upper2[n.years:1]), col = rgb(1,
        0, 0, 0.04), border = rgb(1, 0, 0, 0.5), lty = 3)
    points(y, type = "l")
    points(fitted, type = "l", col = "blue")
    points(fitted2, type = "l", col = "red")

    # legend(x=1, y=m2, legend = c( 'Observed', paste('M1, DIC = ',
    # round(ssm$BUGSoutput$DIC,2)), paste('M2, DIC = ',
    # round(ssmNr2$BUGSoutput$DIC,2)), 'Carrying capacity M1', 'Carrying
    # capacity M2'), lty = c(1,1,1,5,5), col = c( 'black', 'blue', 'red',
    # 'blue', 'red' ), bty = 'n', cex = 1)

    legend(x = 1, y = m2, legend = c("Observed", paste("M1, DIC = ", round(ssm$BUGSoutput$DIC,
        2)), paste("M2, DIC = ", round(ssmNr2$BUGSoutput$DIC, 2))), lty = c(1,
        1, 1), col = c("black", "blue", "red"), bty = "n", cex = 1)

    legend(x = -0.6, y = 4000, legend = c(paste(expression(beta), "=", round(ssm$BUGSoutput$mean$beta,
        2)), paste(expression(beta), "=", round(ssmNr2$BUGSoutput$mean$beta,
        2))), text.col = c("blue", "red"), bty = "n", cex = 1)

}
```

```r
# same as graph.ssm function, only with output on the real scale
graph.ssm2 <- function(ssm, y) {
    fitted <- lower <- upper <- numeric()
    n.years <- length(y)
    for (i in 1:n.years) {
        fitted[i] <- mean(exp(ssm$BUGSoutput$sims.list$s.y[, i]))
        lower[i] <- quantile(exp(ssm$BUGSoutput$sims.list$s.y[, i]), 0.025)
        upper[i] <- quantile(exp(ssm$BUGSoutput$sims.list$s.y[, i]), 0.975)
    }

    m1 <- min(c(y, fitted, lower))
    m2 <- max(c(y, fitted, upper))
    par(mar = c(4.5, 4, 1, 1), cex = 1.2)
    plot(0, 0, ylim = c(m1, m2), xlim = c(0.5, n.years), ylab = "population",
        xlab = "year", frame = F, axe = F)
    axis(2, las = 1)
    axis(1, at = seq(0, n.years, 5), labels = seq(0, n.years, 5))
    axis(1, at = 0:n.years, labels = rep("", n.years + 1), tcl = -0.25)
    polygon(x = c(1:n.years, n.years:1), y = c(lower, upper[n.years:1]), col = "gray90",
        border = "gray90")
    points(y, type = "l")
    points(fitted, type = "l", col = "blue")
    legend(x = 1, y = m2, legend = c("Observed", "Estimated Observations"),
        lty = c(1, 1, 1), col = c("black", "blue"), bty = "n", cex = 1)

}
```

```r
# same as graph.ssm2 function - only for two models
graph.ssm3 <- function(ssm, ssmNr2, y) {
    fitted <- lower <- upper <- numeric()
    n.years <- length(y)
    for (i in 1:n.years) {
        fitted[i] <- mean(exp(ssm$BUGSoutput$sims.list$s.ylog[, i]))
        lower[i] <- quantile(exp(ssm$BUGSoutput$sims.list$s.ylog[, i]), 0.025)
        upper[i] <- quantile(exp(ssm$BUGSoutput$sims.list$s.ylog[, i]), 0.975)
    }

    fitted2 <- lower2 <- upper2 <- numeric()
    for (i in 1:n.years) {
        fitted2[i] <- mean(exp(ssmNr2$BUGSoutput$sims.list$s.ylog[, i]))
        lower2[i] <- quantile(exp(ssmNr2$BUGSoutput$sims.list$s.ylog[, i]),
            0.025)
        upper2[i] <- quantile(exp(ssmNr2$BUGSoutput$sims.list$s.ylog[, i]),
            0.975)
    }

    m1 <- min(c(y, fitted, lower))
    m2 <- max(c(y, fitted, upper))
    par(mar = c(4.5, 5, 1, 1), cex = 1.2)
    plot(0, 0, ylim = c(m1, m2), xlim = c(0.5, n.years), ylab = "population",
        xlab = "year", frame = F, axe = F)
    axis(2, las = 2)
    axis(1, at = seq(0, n.years, 5), labels = seq(1959, 2012, 5))
    axis(1, at = 0:n.years, labels = rep("", n.years + 1), tcl = -0.25)
    # polygon(x = c(1:n.years, n.years:1), y= c(lower, upper[n.years:1]), col =
    # 'gray90', border = 'gray90') polygon(x = c(1:n.years, n.years:1), y=
    # c(lower2, upper2[n.years:1]), col = rgb(1, 0, 0,0.1), border = 'gray90')
    polygon(x = c(1:n.years, n.years:1), y = c(lower, upper[n.years:1]), col = rgb(0,
        0, 1, 0.04), border = rgb(0, 0, 1, 0.5), lty = 3)
    polygon(x = c(1:n.years, n.years:1), y = c(lower2, upper2[n.years:1]), col = rgb(1,
        0, 0, 0.04), border = rgb(1, 0, 0, 0.5), lty = 3)
    points(y, type = "l")
    points(fitted, type = "l", col = "blue")
    points(fitted2, type = "l", col = "red")
    legend(x = 1, y = m2, legend = c("Observed", "Model 1", "Model 2"), lty = c(1,
        1, 1), col = c("black", "blue", "red"), bty = "n", cex = 1)

}
```

Read and prepare data

Count data spring

```
muledat <- read.delim("final_dataset.txt")
spring <- as.data.frame(cast(muledat, year ~ StudyArea, value = "NumbMDSpring"))
area <- unique(muledat$AreaSqKm)
year <- unique(muledat$year)

densitys <- data.frame(matrix(nrow = length(year), ncol = 2))
names(densitys) <- c("year", "pop_density")

densitys[, 1] <- year

for (year_ in 1:length(year)) {
    densitys[year_, 2] <- mean(unlist(spring[year_, 2:27]/area), na.rm = T)
}
y <- log(densitys[, 2] * sum(area))[-c(1, 2, 3)]  # obeserved mules
N.obs <- y
```

Predictors

```
# Hunting -1 = previous year
hunt_all <- as.data.frame(cast(muledat, year ~ StudyArea, value = "d3.1"))
hunt_ <- data.frame(matrix(nrow = length(year), ncol = 2))
names(hunt_) <- c("year", "hunt.obs.1")
hunt_[, 1] <- year

for (year_ in 1:length(year)) {
    hunt_[year_, 2] <- sum(unlist(hunt_all[year_, 2:27] * area))
}

Huntprev.obs_all <- hunt_[, 2]  # hunting/km2 * area of unit


# fill in NAs
for (i in 1:3) {
    Huntprev.obs_all[i] <- mean(Huntprev.obs_all, na.rm = T)
}
plot(Huntprev.obs_all, type = "l")


Huntprev.obs <- as.vector(scale(Huntprev.obs_all))[-c(1, 2, 3)]
Huntprev.obs_quadrat <- as.vector(scale(Huntprev.obs_all^2))[-c(1, 2, 3)]

#---------------------------------------------------------------
# Coyotes
Coy.obs_all <- numeric(length(year))
for (i in 1:length(year)) {
    Coy.obs_all[i] <- mean(muledat$BadlandsCoyote.1000_mi[which(muledat$year ==
        year[i])], na.rm = T)
}
# fill in NAs
for (i in 1:4) {
    Coy.obs_all[i] <- mean(Coy.obs_all, na.rm = T)
}
plot(Coy.obs_all, type = "l")
hist(log(Coy.obs_all))

Coy.obs <- as.vector(scale((Coy.obs_all)))[-c(1, 2, 3)]
Coy.obs_quadrat <- as.vector(scale(Coy.obs_all^2))[-c(1, 2, 3)]

# Coy.obs <- Coy.obs[-c(1,2,3)]

#---------------------------------------------------------------
# climate average min_temp winter

min_temp_w.obs <- as.vector(scale(muledat$Average_mintemp_winter[muledat$StudyArea ==
    "Danielson_Olson"]))[-c(1, 2, 3)]
min_temp_w.obs_quadrat <- as.vector(scale((muledat$Average_mintemp_winter[muledat$StudyArea ==
    "Danielson_Olson"])^2))[-c(1, 2, 3)]
hist(muledat$Average_mintemp_winter[muledat$StudyArea == "Danielson_Olson"])


#---------------------------------------------------------------

# oil wells
oil_well_all <- as.data.frame(cast(muledat, year ~ StudyArea, value = "WELLS_ACTIVEinside"))
oil_well_ <- data.frame(matrix(nrow = length(year), ncol = 2))
names(oil_well_) <- c("year", "number_oil_well")
oil_well_[, 1] <- year

# fill in NAs
```

```
for (year_ in 1:length(year)) {
    oil_well_[year_, 2] <- unlist(sum(oil_well_all[year_, 2:27]))
}

oil_well.obs_all <- oil_well_[, 2]  # total numbers of oil_wells in the area per year
oil_well.obs <- as.numeric(scale(oil_well.obs_all))[-c(1, 2, 3)]
oil_well.obs_quadrat <- as.numeric(scale(oil_well.obs_all^2))[-c(1, 2, 3)]


n.years <- length(N.obs)
N <- numeric(n.years)
```

Specify models in bugs language

Model 1 - without count of oil wells

process model: logistic

- hunting
- min temperature
- coyote

observation model: normaldistribution

- with beta

The model

```
sm_jags5Kb <- function() {
    # priors
    logN.est[1] ~ dunif(7.2, 7.3)  #informativ
    lambda ~ dunif(-1, 1)
    tau.obs ~ dgamma(0.01, 0.01)
    K ~ dunif(100, 8000)  #informativ
    beta ~ dbeta(16, 8)  #informativ
    c ~ dnorm(0, 0.1)
    c2 ~ dnorm(0, 0.1)
    co ~ dnorm(0, 0.1)
    co2 ~ dnorm(0, 0.1)
    h ~ dnorm(0, 0.1)
    h2 ~ dnorm(0, 0.1)


    # state process
    for (t in 1:(T - 1)) {
        logN.est[t + 1] <- logN.est[t] + log(max(0.001, (1 + (lambda + c * clim[t] +
            co * coy[t] + c2 * clim2[t]^2 + co2 * coy2[t]^2)  # growth rate
 * (1 - (N.est[t]/(K)))  # carrying capacity
 +
            h * hunt[t] + h2 * hunt2[t]^2)))  # additional terms
    }
    # observation process
    for (t in 1:T) {
        y[t] ~ dnorm(logN.est[t] + log(max(1e-04, (beta))), tau.obs)
    }

    # pop size on real scale
    for (t in 1:T) {
        N.est[t] <- exp(logN.est[t])
    }
}
```

Prepare the model for mcmc

```
# bundle data
bugs.data <- list(y = N.obs, coy = Coy.obs, clim = min_temp_w.obs, hunt = Huntprev.obs,
    clim2 = min_temp_w.obs_quadrat, hunt2 = Huntprev.obs_quadrat, coy2 = Coy.obs_quadrat,
    T = length(N.obs))

# initial values
inits <- function() {
    list(lambda = runif(1, -1, 1), tau.obs = rgamma(1, 0.01, 0.01), K = 1500,
        c = 0, co = 0, h = 0, c2 = 0, co2 = 0, h2 = 0, beta = 0.7, logN.est = c(runif(1,
            7.2, 7.3), rep(NA, (length(N.obs) - 1))))
}

# pars monitored
parameters <- c("lambda", "N.est", "tau.obs", "beta", "K", "c", "co", "h", "c2",
    "co2", "h2")

# mcmc settings
ni <- 3e+05
nt <- 3
nb <- 150000
nc <- 3
```

Run dmc

```
# call winbugs
ssm5Kb <- jags(bugs.data, inits, parameters.to.save = parameters, model.file = sm_jags5Kb,
    n.chains = nc, n.iter = ni, n.burnin = nb, n.thin = 100, DIC = T)
```

Validate mcmc-output

```
plot(ssm5Kb)
ssm5Kb  # dic = -51.2

pairs.panels(data.frame(K = ssm3$BUGSoutput$sims.list$K, alpha = ssm3$BUGSoutput$sims.list$alpha,
    beta = ssm3$BUGSoutput$sims.list$beta, c = ssm3$BUGSoutput$sims.list$c,
    co = ssm3$BUGSoutput$sims.list$co, h = ssm3$BUGSoutput$sims.list$h, lambda = ssm3$BUGSoutput$sims.li
    tau.obs = ssm3$BUGSoutput$sims.list$tau.obs))
```

```
traceplot(ssm5Kb, mfrow = c(4, 4), ask = F)
autocorr.plot(ssm5b)

ssm_mcmc5b <- as.mcmc(ssm5Kb)
plot(ssm_mcmc5b)
```

Plot

```
par(mfrow = c(1, 1))
graph.ssm(ssm5Kb, exp(y))
```

Model 2 - including oil count of oil wells

process model: logistic

- hunting
- min temperature
- active oil wells
- coyote

observation model: normaldistribution

- with beta

The model

```
sm_jags5K <- function() {

    # priors
    logN.est[1] ~ dunif(7.2, 7.3)  #informative
    lambda ~ dnorm(0, 0.1)
    tau.obs ~ dgamma(0.01, 0.01)
    K ~ dunif(100, 8000)  #informative
    beta ~ dbeta(16, 8)  #informative
    c ~ dnorm(0, 0.1)
    c2 ~ dnorm(0, 0.1)
    co ~ dnorm(0, 0.1)
    co2 ~ dnorm(0, 0.1)
    h ~ dnorm(0, 0.1)
    h2 ~ dnorm(0, 0.1)
    o ~ dnorm(0, 0.1)
    o2 ~ dnorm(0, 0.1)


    # state process
    for (t in 1:(T - 1)) {
        logN.est[t + 1] <- logN.est[t] + log(max(0.001, (1 + (lambda + c * clim[t] +
            co * coy[t] + c2 * clim2[t]^2 + co2 * coy2[t]^2 + o * oil[t] + o2 *
            oil2[t]^2)  # growth rate
 * (1 - (N.est[t]/(K)))  # carrying capacity
 + h * hunt[t] + h2 * hunt2[t]^2)))  # additional terms
```

```
    }
    # observation process
    for (t in 1:T) {
        y[t] ~ dnorm(logN.est[t] + log(max(1e-04, (beta))), tau.obs)
    }

    # pop size on real scale
    for (t in 1:T) {
        N.est[t] <- exp(logN.est[t])
    }
}
```

Prepare the model for mcmc

```
# bundle data
bugs.data <- list(y = N.obs, coy = Coy.obs, clim = min_temp_w.obs, hunt = Huntprev.obs,
    clim2 = min_temp_w.obs_quadrat, hunt2 = Huntprev.obs_quadrat, coy2 = Coy.obs_quadrat,
    oil = oil_well.obs, oil2 = oil_well.obs_quadrat, T = length(N.obs))
# , initial values
inits <- function() {
    list(lambda = rnorm(1, 0, 0.1), tau.obs = rgamma(1, 0.01, 0.01), K = 1500,
        o = 0, c = 0, co = 0, h = 0, o2 = 0, c2 = 0, co2 = 0, h2 = 0, beta = 0.7,
        logN.est = c(runif(1, 7.2, 7.3), rep(NA, (length(N.obs) - 1))))
}

# pars monitored
parameters <- c("lambda", "N.est", "tau.obs", "beta", "K", "o", "c", "co", "h",
    "o2", "c2", "co2", "h2")

# mcmc settings
ni <- 3e+05
nt <- 3
nb <- 150000
nc <- 3
```

Run dmc

```
# call winbugs
ssm5K <- jags(bugs.data, inits, parameters.to.save = parameters, model.file = sm_jags5K,
    n.chains = nc, n.iter = ni, n.burnin = nb, n.thin = 100, DIC = T)
```

Validate mcmc-output

```
plot(ssm5K)
ssm5K  # dic = -53.0

pairs.panels(data.frame(K = ssm3$BUGSoutput$sims.list$K, alpha = ssm3$BUGSoutput$sims.list$alpha,
    beta = ssm3$BUGSoutput$sims.list$beta, c = ssm3$BUGSoutput$sims.list$c,
    co = ssm3$BUGSoutput$sims.list$co, h = ssm3$BUGSoutput$sims.list$h, lambda = ssm3$BUGSoutput$sims.l
    tau.obs = ssm3$BUGSoutput$sims.list$tau.obs))

traceplot(ssm5K, mfrow = c(4, 4), ask = F)
autocorr.plot(ssm5K)

ssm_mcmc5K <- as.mcmc(ssm5K)
plot(ssm_mcmc5K)
```

Plot

```
par(mfrow = c(1, 1))
graph.ssm(ssm5K, exp(y))
```

Create a plot for the report

```
pdf("model_plt_ohne_K.pdf", width = 8, height = 5)
par(mfrow = c(1, 1))
graph.ssm_2(ssm5Kb, ssm5K, exp(y))
dev.off()
```

Model diagnostics

correlation of parameters

```
pars.cor <- function(model, exclude) {
    library(psych)
    pairs.panels(data.frame(model$BUGSoutput$sims.list[!names(model$BUGSoutput$sims.list) %in%
        exclude]))
}
pars.cor(ssm5K, "N.est")
```

Model 1

```
model_M1 <- function() {
    # priors
    logN.est[1] ~ dunif(7.2, 7.3)
    # lambda ~ dnorm(0, 0.01)
    lambda ~ dunif(-1, 1)
    tau.obs ~ dgamma(0.01, 0.01)
    K ~ dunif(100, 8000)
    # alpha ~ dnorm(0, 0.01) alpha ~ dnorm(-1,1)
    beta ~ dbeta(16, 8)  #ä beta verteilung
    # beta ~ dnorm(0, 0.01) w ~ dnorm(0, 0.01)
    c ~ dnorm(0, 0.1)
    c2 ~ dnorm(0, 0.1)
    co ~ dnorm(0, 0.1)
    co2 ~ dnorm(0, 0.1)
    h ~ dnorm(0, 0.1)
    h2 ~ dnorm(0, 0.1)


    # state process
    for (t in 1:(T - 1)) {
        logN.est[t + 1] <- logN.est[t] + log(max(0.001, (1 + (lambda + c * clim[t] +
            co * coy[t] + c2 * clim2[t]^2 + co2 * coy2[t]^2)  # growth rate
 * (1 - (N.est[t]/(K)))  # carrying capacity
 +
            h * hunt[t] + h2 * hunt2[t]^2)))  # additional terms
    }
    # observation process
    for (t in 1:T) {
        y[t] ~ dnorm(logN.est[t] + log(max(1e-04, (beta))), tau.obs)
        # y[t] <- max(y[t], logN.est[t]) y[t] ~ dnorm(logN.est[t] + log(pow((alpha *
        # wood[t] + beta), gamma)), tau.obs) y[t] ~ dnorm(logN.est[t], tau.obs)
    }

    # pop size on real scale
    for (t in 1:T) {
        N.est[t] <- exp(logN.est[t])
    }


    # # # # # # # # # # # # # # # simulate data here: replace all latent
    # variables L with an S.L (mu, lambda, psi, z), as well as the response:

    # state process
    s.logN.est[1] ~ dunif(7.2, 7.3)
    for (t in 1:(T - 1)) {
        s.logN.est[t + 1] <- s.logN.est[t] + log(max(0.001, (1 + (lambda + c *
            clim[t] + co * coy[t] + c2 * clim2[t]^2 + co2 * coy2[t]^2)  # growth rate
 * (1 - (s.N.est[t]/(K)))  # carrying capacity
 +
            h * hunt[t] + h2 * hunt2[t]^2)))  # additional terms
    }
    # observation process
    for (t in 1:T) {
        s.ylog[t] ~ dnorm(s.logN.est[t] + log(max(1e-04, (beta))), tau.obs)
        # y[t] <- max(y[t], logN.est[t]) y[t] ~ dnorm(logN.est[t] + log(pow((alpha *
        # wood[t] + beta), gamma)), tau.obs) y[t] ~ dnorm(logN.est[t], tau.obs)
    }

    # pop size on real scale
    for (t in 1:T) {
        s.N.est[t] <- exp(s.logN.est[t])
    }
}
```

run dmc

```
parameters1 <- c("lambda", "tau.obs", "beta", "K", "c", "co", "h", "c2", "co2",
    "h2", "s.ylog", "s.logN.est")  # which parameters are we interested in getting reported?
# mcmc settings
ni <- 20000
nt <- 10
nb <- 8000
nc <- 3
bugs.data1 <- list(y = N.obs, coy = Coy.obs, clim = min_temp_w.obs, hunt = Huntprev.obs,
    clim2 = min_temp_w.obs_quadrat, hunt2 = Huntprev.obs_quadrat, coy2 = Coy.obs_quadrat,
    T = length(N.obs))

preds_M1 <- jags(bugs.data1, inits = NULL, parameters1, model.file = model_M1,
    n.chains = nc, n.thin = nt, n.iter = ni, n.burnin = nb, working.directory = getwd())
```

Bayesian p_values

```
qq1 <- numeric(54)
for (i in 1:54) {
    qq1[i] <- ecdf(preds_M1$BUGSoutput$sims.list$s.ylog[, i])(y[i])
}
plot(density(qq1, from = 0, to = 1), main = "Bayesian p-values of observations")
```

```
p_value1 <- numeric(54)
for (i in seq(p_value2)) {
    p_value1[i] <- mean(preds_M1$BUGSoutput$sims.list$s.ylog[, i] < y[i])
}
plot(density(p_value1), from = 0, to = 1)
# plot(c(1959:2012), p_value)
```

plots

```
par(mfrow = c(1, 1))
graph.ssm2(preds_M1, exp(y))
```

Model 2

```
model_M2 <- function() {
    # priors
    logN.est[1] ~ dunif(7.2, 7.3)
    # lambda ~ dnorm(0, 0.01)
    lambda ~ dnorm(0, 0.1)
    tau.obs ~ dgamma(0.01, 0.01)
    K ~ dunif(100, 8000)
    beta ~ dbeta(16, 8)  #ä beta verteilung
    c ~ dnorm(0, 0.1)
    c2 ~ dnorm(0, 0.1)
```

```
   co ~ dnorm(0, 0.1)
   co2 ~ dnorm(0, 0.1)
   h ~ dnorm(0, 0.1)
   h2 ~ dnorm(0, 0.1)
   o ~ dnorm(0, 0.1)
   o2 ~ dnorm(0, 0.1)


   # state process
   for (t in 1:(T - 1)) {
       logN.est[t + 1] <- logN.est[t] + log(max(0.001, (1 + (lambda + c * clim[t] +
           co * coy[t] + c2 * clim2[t]^2 + co2 * coy2[t]^2 + o * oil[t] + o2 *
           oil2[t]^2)  # growth rate
* (1 - (N.est[t]/(K)))  # carrying capacity
+ h * hunt[t] + h2 * hunt2[t]^2)))  # additional terms
   }
   # observation process
   for (t in 1:T) {
       y[t] ~ dnorm(logN.est[t] + log(max(1e-04, (beta))), tau.obs)
       # y[t] <- max(y[t], logN.est[t]) y[t] ~ dnorm(logN.est[t] + log(pow((alpha *
       # wood[t] + beta), gamma)), tau.obs) y[t] ~ dnorm(logN.est[t], tau.obs)
   }

   # pop size on real scale
   for (t in 1:T) {
       N.est[t] <- exp(logN.est[t])
   }

   # # # # # # # # # # # # # # # simulate data here: replace all latent
   # variables L with an S.L (mu, lambda, psi, z), as well as the response:

   # state process
   s.logN.est[1] ~ dunif(7.2, 7.3)
   for (t in 1:(T - 1)) {
       s.logN.est[t + 1] <- s.logN.est[t] + log(max(0.001, (1 + (lambda + c *
           clim[t] + co * coy[t] + c2 * clim2[t]^2 + co2 * coy2[t]^2 + o *
           oil[t] + o2 * oil2[t]^2)  # growth rate
* (1 -
           (s.N.est[t]/(K)))  # carrying capacity
+ h *
           hunt[t] + h2 * hunt2[t]^2)))  # additional terms
   }
   # observation process
   for (t in 1:T) {
       s.ylog[t] ~ dnorm(s.logN.est[t] + log(max(1e-04, (beta))), tau.obs)
       # y[t] <- max(y[t], logN.est[t]) y[t] ~ dnorm(logN.est[t] + log(pow((alpha *
       # wood[t] + beta), gamma)), tau.obs) y[t] ~ dnorm(logN.est[t], tau.obs)
   }

   # pop size on real scale
   for (t in 1:T) {
       s.N.est[t] <- exp(s.logN.est[t])
   }
```

```
}
```

run dmc

```
parameters2 <- c("lambda", "tau.obs", "beta", "K", "o", "c", "co", "h", "o2",
    "c2", "co2", "h2", "s.ylog", "s.logN.est")  # which parameters are we interested in getting reporte
# mcmc settings
ni <- 20000
nt <- 10
nb <- 8000
nc <- 3
bugs.data2 <- list(y = N.obs, coy = Coy.obs, clim = min_temp_w.obs, hunt = Huntprev.obs,
    clim2 = min_temp_w.obs_quadrat, hunt2 = Huntprev.obs_quadrat, coy2 = Coy.obs_quadrat,
    oil = oil_well.obs, oil2 = oil_well.obs_quadrat, T = length(N.obs))

preds_M2 <- jags(bugs.data2, inits = NULL, parameters2, model.file = model_M2,
    n.chains = nc, n.thin = nt, n.iter = ni, n.burnin = nb, working.directory = getwd())


plot(preds_M1)
```

Bayesian p_values

```
qq2 <- numeric(54)
for (i in 1:54) {
    qq2[i] <- ecdf(preds_M2$BUGSoutput$sims.list$s.ylog[, i])(y[i])
}
plot(density(qq2, from = 0, to = 1), main = "Bayesian p-values of observations")
```

```
p_value2 <- numeric(54)
for (i in seq(p_value2)) {
    p_value2[i] <- mean(preds_M2$BUGSoutput$sims.list$s.ylog[, i] < y[i])
}
plot(density(p_value2), from = 0, to = 1)
# plot(c(1959:2012), p_value)
```

plots M2

```
par(mfrow = c(1, 1))
graph.ssm2(preds_M2, exp(y))
```

plots for report

```
par(mfrow = c(1, 1))
graph.ssm3(preds_M1, preds_M2, exp(y))
```

```
pdf("bay_p_values_M1andM2.pdf", width = 8, height = 8)
par(mar = c(7, 6, 4, 2), mfrow = c(1, 1))
plot(density(qq1, from = 0, to = 1), main = "", xlab = "p-values", col = "blue",
    ylim = c(0.2, 1.4), lwd = 3, cex = 2, yaxt = "n", xaxt = "n", cex.lab = 3,
    ann = F)
axis(1, cex.axis = 2, line = 0, outer = F)
axis(2, cex.axis = 2)
mtext(side = 1, text = "p-values", line = 4, cex = 3)
mtext(side = 2, text = "Density", line = 3.5, cex = 3)
lines(density(qq2, from = 0, to = 1), main = "", col = "red", lwd = 3)
dev.off()

pdf("expected_observed_M1andM2.pdf", width = 8, height = 8)
par(mar = c(7, 6, 4, 2), mfrow = c(1, 1))
plot(apply(exp(preds_M1$BUGSoutput$sims.list$s.ylog), 2, mean), exp(y), las = 1,
    xlab = "", ylab = "", col = "blue", pch = 18, cex = 2, cex.lab = 3, yaxt = "n",
    xaxt = "n")
points(apply(exp(preds_M2$BUGSoutput$sims.list$s.ylog), 2, mean), exp(y), las = 1,
    col = "red", pch = 18, cex = 2)
axis(1, cex.axis = 2, line = 0, outer = F)
axis(2, cex.axis = 2)
mtext(side = 1, text = "expected", line = 4, cex = 3)
mtext(side = 2, text = "observed", line = 3.5, cex = 3)
abline(0, 1, lwd = 3)
legend("topleft", legend = c("Model 1", "Model 2"), pch = c(18, 18), lty = c(1,
    1), col = c("blue", "red"), bty = "n", cex = 3)
dev.off()

pdf("residuals_M1andM2.pdf", width = 8, height = 8)
par(mar = c(7, 6, 4, 2), mfrow = c(1, 1))
plot(qq1 ~ apply(exp(preds_M1$BUGSoutput$sims.list$s.ylog), 2, mean), las = 1,
    xlab = "", ylab = "", col = "blue", pch = 20, cex = 3, yaxt = "n", xaxt = "n",
    cex.lab = 3)
axis(1, cex.axis = 2, line = 0, outer = F)
axis(2, cex.axis = 2)
mtext(side = 1, text = "expected", line = 4, cex = 3)
mtext(side = 2, text = "standardised residuals", line = 3.5, cex = 3)
points(qq2 ~ apply(exp(preds_M2$BUGSoutput$sims.list$s.ylog), 2, mean), las = 1,
    col = "red", pch = 20, cex = 3)
dev.off()
```