

SFS 2023 Short Course – Bayesian Applications in Environmental and Ecological Studies with R and Stan

Song S. Qian

6/3/2023

Hierarchical Models

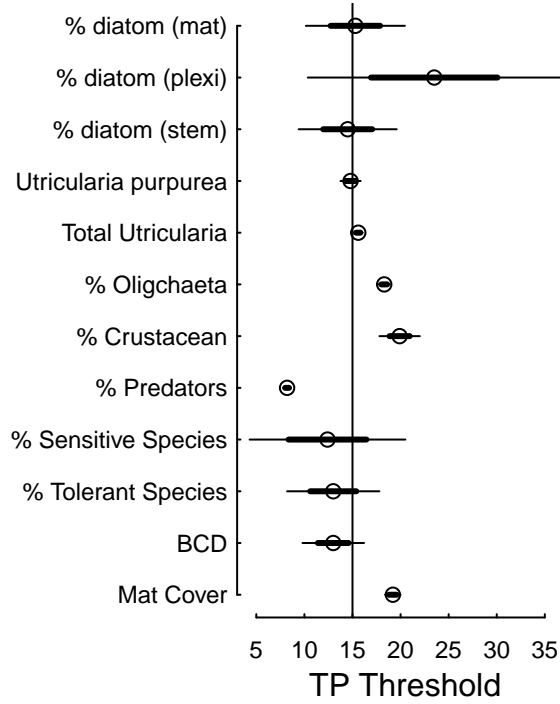
A simple start – setting environmental standard in the Everglades

Richardson et al. (2007) conducted a mesocosm study in the Everglades of South Florida to investigate the response of wetland ecosystems to elevated phosphorus (P) input. They created a phosphorus gradient using artificial flumes and observed changes in the mesocosm ecosystem. The researchers determined the thresholds of total phosphorus (TP) concentrations that would lead to significant changes in algae, macroinvertebrates, and macrophytes communities. To quantify these thresholds, they utilized 12 biological indicators that represented how quickly the indicators would respond to changes in TP concentrations. The study reported the means of the 12 thresholds along with their corresponding 95% confidence intervals. In our analysis, we employed the 95% confidence intervals (typically calculated as the mean plus/minus 2 standard errors) to estimate the standard deviation of the mean thresholds.

```
y.hat <- c(19.2, 13, 13, 12.4, 8.2, 19.9, 18.3, 15.6, 14.8, 14.5, 23.5, 15.3)
sigma.hat <- c(1.6, 6.4, 9.6, 16.2, 0.8, 4.2, 1.2, 0.9, 2.1, 10.2, 26.3, 10.3)/4

metrics <- c("Mat Cover", "BCD", "% Tolerant Species", "% Sensitive Species", "% Predators",
             "% Crustacean", "% Oligochaeta", "Total Utricularia", "Utricularia purpurea",
             "% diatom (stem)", "% diatom (plexi)", "% diatom (mat)")
metricsTEX <- c("Mat Cover", "BCD", "\\% Tol Sp", "\\% Sen Sp", "\\% Pred", "\\% Crust",
               "\\% Oligochaeta", "Tot Utr", "Utr P.", "\\% diatom (stem)", "\\% diatom (plexi)",
               "\\% diatom (mat)")

par(mar = c(3, 7, 1, 0.5), mgp = c(1.25, 0.25, 0), tck = 0.01)
plot(range(y.hat - 2 * sigma.hat, y.hat + 2 * sigma.hat), c(1, length(y.hat)), type = "n",
     xlab = "TP Threshold", ylab = " ", axes = F)
axis(1, cex.axis = 0.75)
axis(2, at = 1:length(y.hat), labels = metrics, las = 1, cex.axis = 0.75)
segments(x0 = y.hat + sigma.hat, x1 = y.hat - sigma.hat, y0 = 1:length(y.hat), y1 = 1:length(y.hat),
        lwd = 3) #,
# col=c(1,1, 2,2,2,2,2, 3,3, 4, 4,4))
segments(x0 = y.hat + 2 * sigma.hat, x1 = y.hat - 2 * sigma.hat, y0 = 1:length(y.hat),
        y1 = 1:length(y.hat), lwd = 1) #,
# col=c(1,1, 2,2,2,2,2, 3,3, 4, 4,4))
points(x = y.hat, y = 1:length(y.hat))
abline(v = 15)
```



Richardson et al. (2007) recommended that the total phosphorus (TP) concentration standard should be 15 $\mu\text{g/L}$, which is close to the average of the 12 means and the mean of *Utricularia purpurea*, a keystone species in the Everglades wetland ecosystem. However, setting a TP standard solely based on one species is less convincing due to the scientifically vague legal requirement of protecting the natural balance of flora and fauna in the Everglades. Although the value is close to the average of all examined metrics, each metric represents a specific aspect of the ecosystem and cannot alone describe the natural balance at the ecosystem level.

Each metric represents a specific aspect of the ecosystem (individual species or species groups). These species-specific indicators by themselves cannot describe the natural balance at the ecosystem level. Suppose that there are a total of n indicators to represent the Everglades wetland ecosystem and we have estimates of thresholds of these indicators ($\phi_j, j = 1, \dots, n$). Although each individual threshold cannot adequately represent the natural imbalance, the distribution of all thresholds should provide a quantitative summary of how TP concentration levels would affect the ecosystem as a whole. Because the 12 metrics were carefully selected to represent the Everglades wetland ecosystems, ecosystem-level threshold distribution can be estimated from these individual-level thresholds. Instead of using the average of the estimated change points of the 12 metrics, we integrate these estimates using a hierarchical model to properly represent the estimation uncertainty we have about these estimates.

The available data are the estimated change point $\hat{\phi}_j$ and its standard deviation $\hat{\sigma}_j$. These two numbers form an indicator-level model:

$$\hat{\phi}_j \sim N(\theta_j, \hat{\sigma}_j^2).$$

The estimated mean and standard deviation, $\hat{\phi}_j$ and $\hat{\sigma}_j$, summarize the information in the data. In the absence of additional information to determine the relative magnitude of thresholds for different metrics, we assume that the θ_j 's can be modeled as follows:

$$\theta_j \sim N(\mu, \tau^2),$$

which represents a common prior distribution for θ_j . This hierarchical model extends the concept of Stein's paradox from the 1960s. With 12 thresholds to estimate simultaneously, Stein's paradox informs us that estimating them individually is mathematically inadmissible. By shrinking the individually estimated means towards the overall mean, we can improve the accuracy of the overall estimation. Hierarchical modeling

has shown its value in applied fields since Stein’s paradox, particularly in addressing environmental and ecological data analysis problems that involve variables representing different levels of spatial, temporal, and organizational aggregations. Failing to properly address data hierarchy can lead to Simpson’s paradox.

The common prior distribution used in the Everglades problem reflects two key considerations: (1) our understanding that the θ_j values are likely to differ across different metrics, and (2) our lack of understanding regarding the specific differences between the θ_j values. The variance parameter τ^2 represents the between-metric variance. In this context, we expanded the meaning of τ^2 to encompass the variance among all possible metric means, not just the 12 metrics represented in the available data. This hierarchical model connects all metrics together through the common prior distribution $N(\mu, \tau^2)$.

Since we have no prior knowledge about the values of μ and τ^2 , we will utilize Stan default weakly informative priors. From the perspective of modeling individual metrics, each time we model a metric mean (i.e., $\hat{\phi}_j \sim N(\theta_j, \hat{\sigma}_j^2)$), we employ Bayesian estimation and assign a prior distribution to the unknown metric mean θ_j . In this case, the prior distribution parameters are estimated based on data from other metrics. If we have an additional metric, the hierarchical model for the 13th metric can be seen as a Bayesian estimation utilizing an informative prior. This informative prior is derived from other similar quantities. This interpretation leads to the concept of treating a prior as the distribution of similar quantities, known mathematically as exchangeable units. Studies involving similar quantities from different contexts, such as eutrophication studies in various lakes, are often referred to as parallel studies. Exchangeable units can pertain to spatial aspects (e.g., different lakes, distinct eco-regions when studying climate change impacts), temporal aspects (observations from the same location over different seasons or years), and, as illustrated in this example, organizational aspects (different metrics representing various aspects of an ecosystem). I believe that any environmental and ecological data analysis problem can be approached as a hierarchical modeling problem, considering the inherent hierarchical structure of the data.

Returning to the Everglades example, we can observe the well-known shrinkage effect of hierarchical modeling, which is responsible for enhancing overall estimation accuracy. Let’s provide an intuitive explanation of why shrinking estimates towards the overall mean leads to improved accuracy. When we say that an estimate has an error, we mean that the estimated value is either too high or too low. However, when we have only one parameter to estimate, we have no basis to believe that the estimate is biased towards being too high or too low. Thus, an unbiased estimator is preferred, as, on average, it tends to be correct.

When we have estimates of the same parameter from multiple exchangeable units, the overall mean of these estimates serves as a reasonable reference to gauge whether an individual estimate is likely to be too high or too low. Consequently, shrinking these estimates towards the overall mean is more likely to improve their accuracy.

One common computational issue in hierarchical models arises from the potential strong correlation among the multiple means (i.e., θ_j values), especially when there are only a small number of exchangeable units. This correlation often stems from the difficulty in accurately quantifying the hyperparameters (μ, σ^2) . The phenomenon known as Neal’s funnel is a typical manifestation of this correlation.

As we discussed earlier, we can address this challenge through reparameterizing the model. Instead of directly sampling θ_j as random variables, we can utilize the relationship between a normally distributed random variable with mean μ and standard deviation τ , and a standard normal random variable $z \sim N(0, 1)$:

$$\theta_j = \mu + \tau \times z_j.$$

By defining θ_j as a transformed variable, we can improve the Stan model’s computational performance by avoiding directly sampling from θ_j :

```
everg_stan <- "
data {
  int<lower=0> J; // number of schools
  real y[J]; // estimated treatment effects
  real<lower=0> sigma[J]; // s.e. of effect estimates
}
```

```

parameters {
  real mu;
  real<lower=0> tau;
  real eta[J];
}
transformed parameters {
  real theta[J];
  for (j in 1:J)
    theta[j] = mu + tau * eta[j];
}
model {
  eta ~ normal(0, 1);
  y ~ normal(theta, sigma);
}
"

fit1 <- stan_model(model_code = everg_stan)

## Trying to compile a simple C file
## Running /usr/lib/R/bin/R CMD SHLIB foo.c
## using C compiler: 'gcc (Ubuntu 11.3.0-1ubuntu1~22.04.1) 11.3.0'
## gcc -I"/usr/share/R/include" -DNDEBUG -I"/usr/lib/R/site-library/Rcpp/include/" -I"/usr/lib/R/site-library/RcppEigen/include/Eigen/Core:88,
## from /usr/lib/R/site-library/RcppEigen/include/Eigen/Dense:1,
## from /usr/local/lib/R/site-library/StanHeaders/include/stan/math/prim/fun/Eigen.hpp:
## from <command-line>:
## /usr/lib/R/site-library/RcppEigen/include/Eigen/src/Core/util/Macros.h:628:1: error: unknown type name 'EIGEN_DEVICE_FUNC'
## 628 | namespace Eigen {
##      | ^~~~~~
## /usr/lib/R/site-library/RcppEigen/include/Eigen/src/Core/util/Macros.h:628:17: error: expected '=',
## 628 | namespace Eigen {
##      | ^
## In file included from /usr/lib/R/site-library/RcppEigen/include/Eigen/Dense:1,
## from /usr/local/lib/R/site-library/StanHeaders/include/stan/math/prim/fun/Eigen.hpp:
## from <command-line>:
## /usr/lib/R/site-library/RcppEigen/include/Eigen/Core:96:10: fatal error: complex: No such file or directory
## 96 | #include <complex>
##      | ^~~~~~
## compilation terminated.
## make: *** [/usr/lib/R/etc/Makeconf:191: foo.o] Error 1

```

As usual, we first organize input data and initial values

```

everg_in <- function(y = y.hat, sig = sigma.hat, n.chains = nchains) {
  J <- length(y)
  data <- list(y = y, sigma = sig, J = J)
  inits <- list()
  for (i in 1:n.chains) inits[[i]] <- list(eta = rnorm(J), mu = rnorm(1), tau = runif(1))
  pars <- c("theta", "mu", "eta", "tau")
  return(list(data = data, inits = inits, pars = pars, chains = n.chains))
}

input.to.stan <- everg_in()
fit2keep <- sampling(fit1, data = input.to.stan$data, init = input.to.stan$inits,

```

```

pars = input.to.stan$pars, iter = niters, thin = nthin, chains = input.to.stan$chains,
control = list(max_treedepth = 25))

## Warning: There were 1 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.

## Warning: Examine the pairs() plot to diagnose sampling problems

print(fit2keep)

```

```

## Inference for Stan model: anon_model.
## 8 chains, each with iter=5000; warmup=2500; thin=8;
## post-warmup draws per chain=313, total post-warmup draws=2504.
##
##      mean se_mean   sd  2.5%   25%   50%   75%  97.5% n_eff Rhat
## theta[1]  19.15    0.01 0.41  18.35  18.89 19.15 19.41 19.96 2506   1
## theta[2]  13.40    0.03 1.49  10.51  12.42 13.37 14.42 16.27 2482   1
## theta[3]  13.79    0.04 2.07   9.77  12.42 13.78 15.21 17.81 2473   1
## theta[4]  13.88    0.06 3.00   7.81  11.88 13.97 15.96 19.58 2522   1
## theta[5]   8.22    0.00 0.20   7.83   8.08  8.22  8.35  8.58 2415   1
## theta[6]  19.60    0.02 1.04  17.52  18.91 19.59 20.28 21.62 2582   1
## theta[7]  18.28    0.01 0.31  17.69  18.08 18.28 18.49 18.88 2592   1
## theta[8]  15.60    0.00 0.23  15.16  15.45 15.60 15.76 16.04 2455   1
## theta[9]  14.82    0.01 0.52  13.85  14.47 14.82 15.16 15.85 2363   1
## theta[10] 14.77    0.04 2.19  10.46  13.30 14.84 16.23 18.98 2437   1
## theta[11] 17.55    0.07 3.62  10.65  15.11 17.40 19.86 25.06 2397   1
## theta[12] 15.30    0.05 2.10  10.99  13.91 15.33 16.69 19.50 2171   1
## mu        15.40    0.03 1.33  12.76  14.57 15.39 16.22 18.11 2305   1
## eta[1]     0.99    0.01 0.42   0.20   0.70  0.98  1.26  1.85 2286   1
## eta[2]    -0.52    0.01 0.49  -1.49  -0.85 -0.51 -0.19  0.43 2305   1
## eta[3]    -0.41    0.01 0.58  -1.58  -0.80 -0.41 -0.02  0.72 2425   1
## eta[4]    -0.38    0.02 0.75  -1.88  -0.87 -0.38  0.12  1.09 2519   1
## eta[5]    -1.90    0.01 0.58  -3.11  -2.29 -1.88 -1.50 -0.83 2368   1
## eta[6]     1.10    0.01 0.47   0.21   0.78  1.09  1.41  2.05 2234   1
## eta[7]     0.76    0.01 0.38   0.03   0.50  0.75  1.01  1.55 2275   1
## eta[8]     0.05    0.01 0.33  -0.60  -0.16  0.06  0.27  0.73 2280   1
## eta[9]    -0.15    0.01 0.35  -0.84  -0.39 -0.15  0.08  0.54 2261   1
## eta[10]   -0.16    0.01 0.61  -1.39  -0.56 -0.14  0.25  1.04 2442   1
## eta[11]    0.52    0.02 0.86  -1.20  -0.04  0.52  1.09  2.22 2428   1
## eta[12]   -0.02    0.01 0.59  -1.19  -0.41 -0.03  0.35  1.18 2415   1
## tau        4.04    0.02 1.13   2.47   3.23  3.85  4.58  6.81 2372   1
## lp__       -9.71    0.07 3.39 -16.96 -11.84 -9.51 -7.31 -3.79 2298   1
##
## Samples were drawn using NUTS(diag_e) at Tue May 30 13:08:02 2023.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).

```

Now processing Stan results

```

everg_fit1 <- rvsims(as.matrix(as.data.frame(rstan::extract(fit2keep, permuted = T))))

## shrinkage effect
everg_theta <- rvsims(as.matrix(as.data.frame(rstan::extract(fit2keep, permuted = T,
  pars = "theta"))))

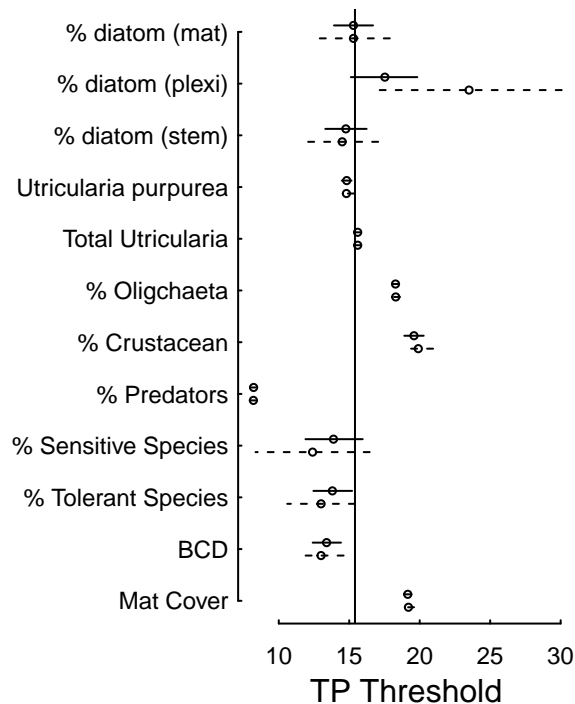
```

```

everg_mu <- rvsims(as.matrix(as.data.frame(rstan::extract(fit2keep, permuted = T,
  pars = "mu"))))
everg_tau <- rvsims(as.matrix(as.data.frame(rstan::extract(fit2keep, permuted = T,
  pars = "tau"))))
theta <- summary(everg_theta)
mu <- summary(everg_mu)
tau <- summary(everg_tau)

par(mar = c(3, 7, 1, 0.5), mgp = c(1.25, 0.25, 0), tck = 0.01)
plot(range(y.hat - 1 * sigma.hat, y.hat + 1 * sigma.hat), c(1, length(y.hat)), type = "n",
  xlab = "TP Threshold", ylab = " ", axes = F)
axis(1, cex.axis = 0.75)
axis(2, at = seq(1, length(y.hat)), labels = metrics, las = 1, cex.axis = 0.75)
segments(x0 = y.hat + sigma.hat, x1 = y.hat - sigma.hat, y0 = seq(1, length(y.hat)) -
  0.125, y1 = seq(1, length(y.hat)) + 0.125, lwd = 1, lty = 2)
## col=c(1,1, 2,2,2,2,2, 3,3, 4, 4,4),
segments(x0 = theta$"25%", x1 = theta$"75%", y0 = (seq(1, length(y.hat))) + 0.125,
  y1 = (seq(1, length(y.hat))) + 0.125)
## col=c(1,1, 2,2,2,2,2, 3,3, 4, 4,4))
points(x = y.hat, y = seq(1, length(y.hat)) - 0.125, cex = 0.5)
## col=c(1,1, 2,2,2,2,2, 3,3, 4, 4,4),
points(x = theta$mean, y = 0.125 + (seq(1, length(y.hat))), cex = 0.5)
## pch=16,col=c(1,1, 2,2,2,2,2, 3,3, 4, 4,4))
abline(v = mu$mean)

```

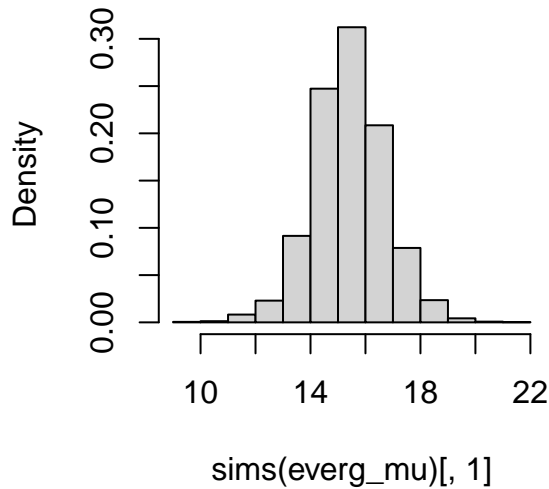


How should we determine the TP concentration standard?

Determining the TP (total phosphorus) concentration standard is primarily an ecological and environmental management decision. From a statistical perspective, the question is whether we should derive the standard based on the overall mean (μ) or consider the distribution of all metrics: between the posterior distribution of μ and the hyper-distribution.

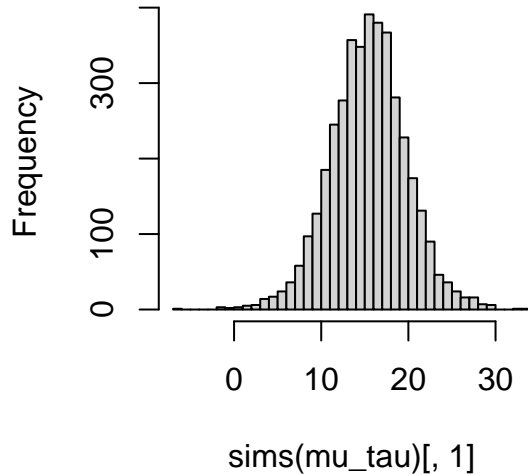
```
## mu versus N(mu, tau)
mu_tau <- rvmnorm(1, everg_mu, everg_tau)
p1 <- hist(sims(everg_mu)[, 1], freq = F)
```

Histogram of sims(everg_mu)[, 1]

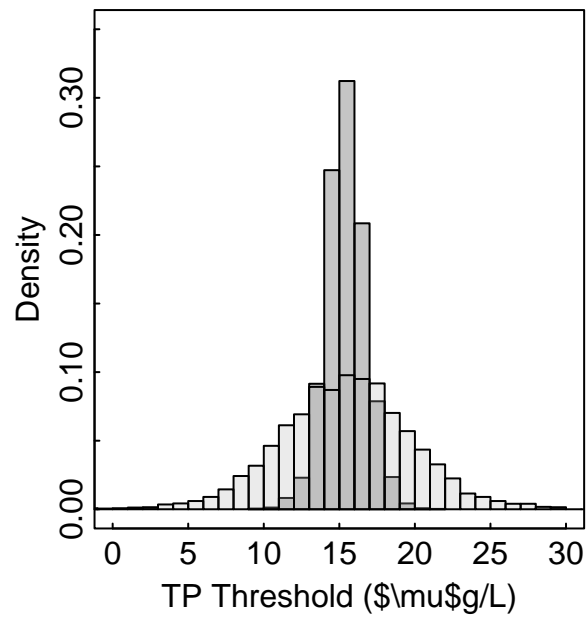


```
p2 <- hist(sims(mu_tau)[, 1], nclass = 35)
```

Histogram of sims(mu_tau)[, 1]



```
par(mar = c(3, 3, 1, 0.5), mgp = c(1.25, 0.125, 0), tck = 0.01)
plot(p1, col = rgb(0.1, 0.1, 0.1, 1/4), xlim = c(0, 30), ylim = c(0, 0.35), freq = F,
     xlab = "TP Threshold ( $\mu$ /L)", main = "") # first histogram
plot(p2, col = rgb(0.7, 0.7, 0.7, 1/4), xlim = c(0, 30), ylim = c(0, 0.35), freq = F,
     add = T) # second
box()
```



```
c(quantile(sims(everg_mu), prob = 0.05), quantile(sims(mu_tau), prob = 0.05))
```

```
##          5%          5%
## 13.307739  8.280074
```