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

Song S. Qian

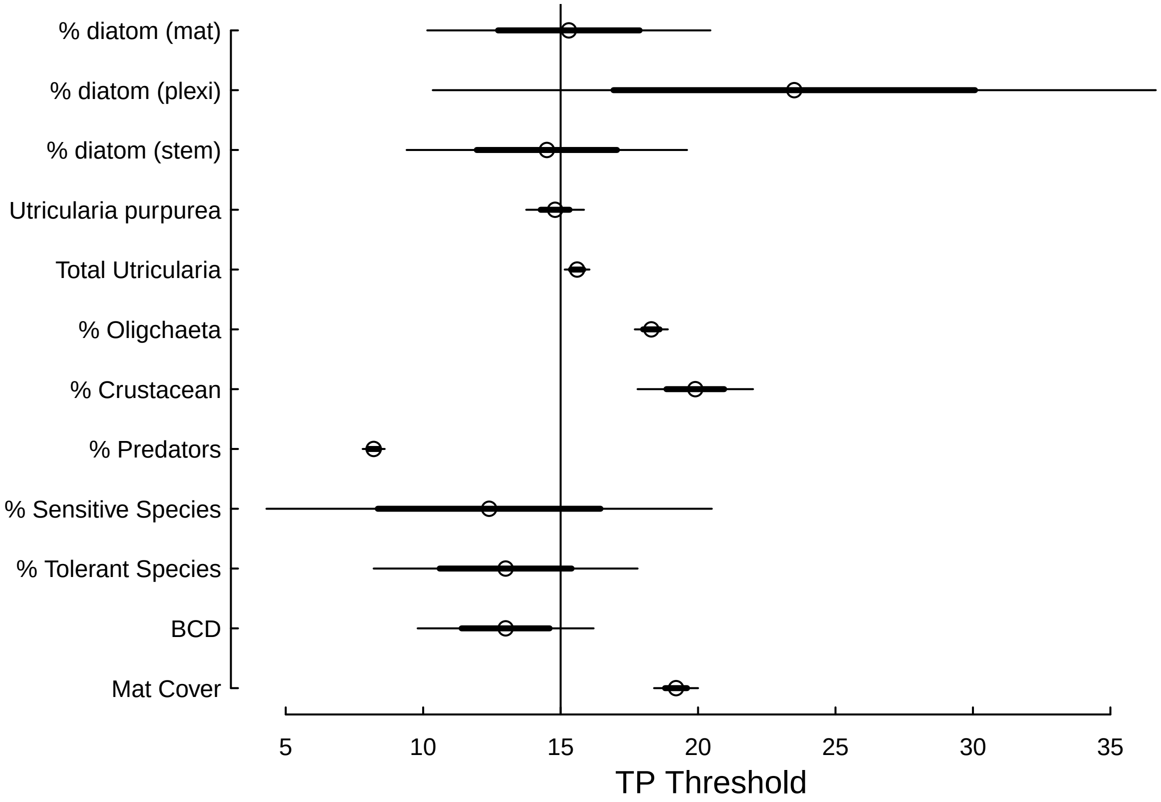
6/3/2023

Hierarchical Models

A simple start – setting environmental standard in the Everglades

Richardson et al (2007) reported a mesocosm study of wetland ecosystem responding to elevated phosphorous (P) input. The study was conducted in the Everglades of South Florida. A phosphorus gradient was created using a series of artificial flumes, from which changes in the mesocosm ecosystem were observed. Researchers calculated total P (TP) thresholds that will induce large changes in algae, macroinvertebrates, and macrophytes communities using 12 biological indicators. These indicators represent a combination of how fast the indicators would respond to changes in TP concentrations. They reported the 12 threshold means and their 95% intervals. We used the 95% intervals (typically mean plus/minus 2 standard error) to derive the standard deviation of the estimated means.

```r
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","% Oligchaeta","Total Utricularia",
             "Utricularia purpurea","% diatom (stem)","% diatom (plexi)",
             "% diatom (mat)")
metricsTEX <- c("Mat Cover","BCD","\\% Tol Sp",
                "\\% Sen Sp", "\\% Pred","\\% Crust",
                "\\% Oligchaeta","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 TP concentration standard should be 15 μg/L, close to the average of the 12 means and the mean of Utricularia purpurea, a keystone species of the Everglades wetland ecosystem. Because the legal requirement of setting a TP standard to protect "the natural balance of flora and fauna of the Everglade" is scientifically vague, a threshold based on one species is always less convincing, even though the value is close to the average of the change points of all metrics examined. 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 $(\phi_j, j = 1, \cdots, n)$. 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 data we have are the estimated change point $\hat{\phi}_j$ and its standard deviation $\hat{\sigma}_j$. These two numbers provide an indicator-level model:

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

Information in the data is summarized in the estimated mean and standard deviation $\hat{\phi}_j$ and $\hat{\sigma}_j$. When we have no additional information to determine the relative magnitude of the threshold for different metrics, we can assume that θ_j's can be modeled as follows

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

that is, assuming a common prior distribution for θ_j. The above two equations form the simplest hierarchical model. It is the natural extension of the Stein's paradox of the 1960s. Here we have 12 thresholds to estimate at the same time and Stein's paradox told us that estimating them one at a time is mathematically inadmissible. Shrinking these individually estimated means towards the overall mean can always improve the overall estimation accuracy. Statistical development since Stein's paradox has shown the value of hierarchical modeling in applied fields. In my opinion, hierarchical modeling is a key to address environmental and ecological data analysis problems where variables representing different levels of spatial, temporal, and organizational aggregations. When data analysis crossing different levels of aggregation, the hierarchical structure of the aggregation can be properly modeled under a hierarchical modeling framework. Without properly addressing data hierarchy, we can be tripped by Simpson's paradox.

This common prior distribution in the Everglades problem here reflects (1) our understanding that θ_js are likely to be different for different metrics, and (2) our lack of understanding on how θ_js are different from each other. The variance parameter τ^2 is the between metric variance. we expanded the meaning of τ^2 to be the variance among all possible metric means (not just the 12 metrics represented in the data). This model links all metrics together through the common prior distribution $N(\mu, \tau^2)$. As we have no prior knowledge of μ and τ^2, we will use Stan default weakly informative priors. Viewing from the perspective of modeling individual metrics, each time a metric mean is modeled (i.e., $\hat{\phi}_j \sim N(\theta_j, \hat{\sigma}_j^2)$) we are using a Bayesian estimation and the unknown metric mean θ_j is given a prior distribution. The prior distribution parameters in this case are estimated based on data from other metrics. If we have another metric, the hierarchical model for the 13th metric can be seen as a Bayesian estimation using informative prior. The informative prior is derived from other similar quantities. This interpretation gives me the idea of treating a prior as the distribution of similar quantities. Mathematically we call these similar quantities exchangeable units. Similar studies from exchangeable units (e.g., eutrophication studies in different lakes) are often known as parallel studies. Exchangeable units can be spatial (e.g., different lakes, different eco-regions when study climate change impacts), temporal (observations from the same location over different seasons or years), and, as in this example, organizational (different metrics representing different aspects of an ecosystem). I find that we can think any environmental and ecological data analysis problem as a hierarchical modeling problem.

Back to the Everglades example, we illustrate the famed shrinkage effect of the hierarchical modeling, which is responsible for the improved overall estimation accuracy. (Here is an intuitive explanation of why shrinking estimates towards the overall mean would improve overall accuracy. When we say that an estimate has error, we mean that the estimated value is either too high or too low. With only one parameter to estimate, we have no reason to believe the estimate is too high or too low. As a result, we prefer an unbiased estimator. On average we are right. When we have estimates of the same parameter from multiple exchangeable units, the overall mean of these means provides a reasonable reference on whether an individual estimate is likely too high or too low. As a result, shrinking these estimates towards the overall mean is more likely to improve these estimates.)

A common computation problem in hierarchical model is the potentially strong correlation among the multiple means (i.e., θ_j's), especially when the number of exchangeable units is small. The correlation is often a result of the difficulty in quantifying the hyper-parameters (μ, σ^2). The Neal's funnel is a common phenomenon. As we've seen earlier, we can reparameterize the model: instead of directly sampling θ_j as random variables, we use the relationship between a normal random variable with mean μ and standard deviation τ and the 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 improve the Stan model's computatiopnal performance by avoiding directly sampling from them:

```
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)
```

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.
```

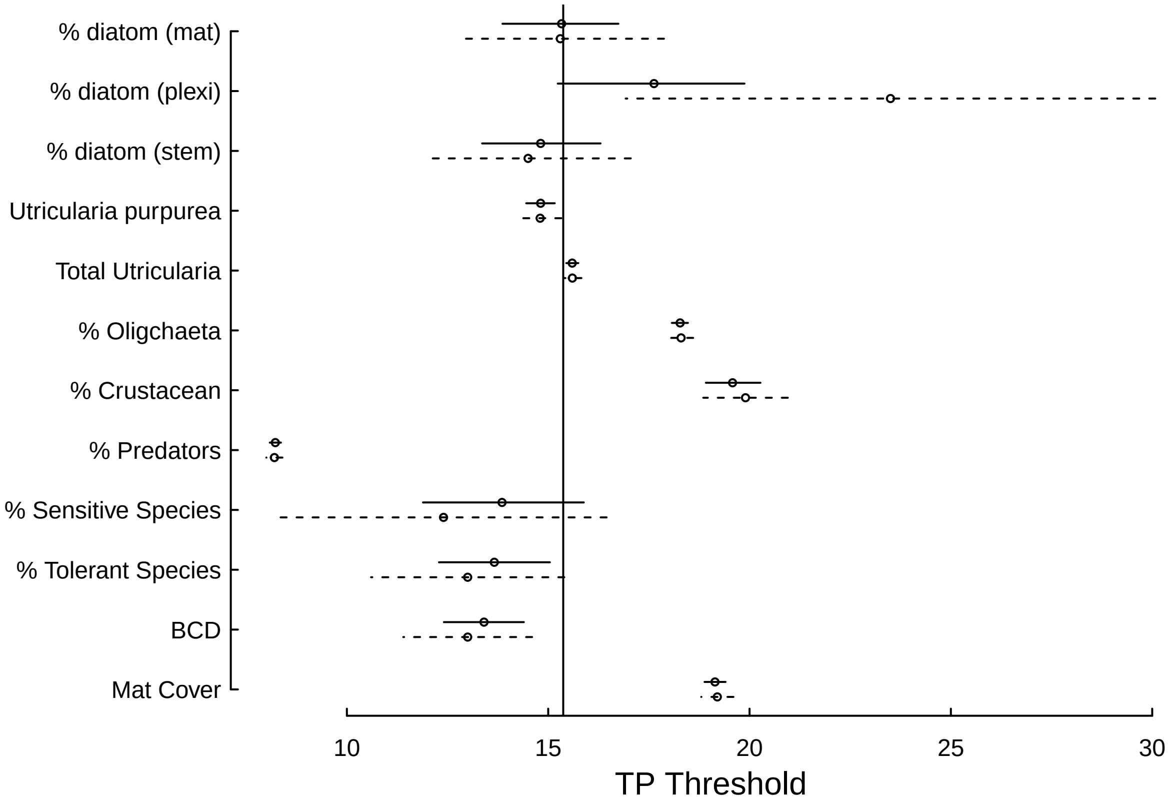
```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
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]   19.14    0.01 0.39 18.37 18.89 19.14 19.40 19.90  2349    1
## theta[2]   13.41    0.03 1.50 10.46 12.41 13.41 14.40 16.33  2474    1
## theta[3]   13.66    0.04 2.05  9.57 12.28 13.69 15.04 17.71  2419    1
## theta[4]   13.84    0.06 3.00  7.87 11.87 13.87 15.90 19.83  2268    1
## theta[5]    8.23    0.00 0.20  7.83  8.09  8.23  8.36  8.62  2252    1
## theta[6]  19.57    0.02 1.02  17.52  18.90 19.56 20.26 21.52  2253    1
## theta[7]  18.27    0.01 0.30  17.70  18.07 18.27 18.47 18.85  2383    1
## theta[8]  15.59    0.00 0.22  15.16  15.45 15.59 15.75 16.03  2381    1
## theta[9]  14.81    0.01 0.52  13.83  14.45 14.81 15.16 15.85  2597    1
## theta[10] 14.82    0.05 2.12  10.52  13.37 14.86 16.28 18.92  2222    1
## theta[11] 17.63    0.07 3.62  10.82  15.22 17.49 19.87 25.29  2479    1
## theta[12] 15.33    0.04 2.19  10.97  13.86 15.32 16.76 19.67  2444    1
## mu        15.37    0.03 1.35  12.74  14.52 15.36 16.21 18.02  2243    1
## eta[1]     1.00    0.01 0.42   0.21   0.70  0.99  1.27  1.85  2276    1
## eta[2]    -0.50    0.01 0.47  -1.47  -0.82 -0.49 -0.18  0.39  2068    1
## eta[3]    -0.43    0.01 0.57  -1.57  -0.82 -0.43 -0.04  0.68  2534    1
## eta[4]    -0.37    0.02 0.76  -1.88  -0.88 -0.38  0.12  1.16  2286    1
## eta[5]    -1.88    0.01 0.58  -3.09  -2.26 -1.86 -1.48 -0.81  2133    1
## eta[6]     1.10    0.01 0.48   0.24   0.77  1.07  1.42  2.07  2284    1
## eta[7]     0.77    0.01 0.39   0.04   0.49  0.76  1.03  1.54  2248    1
## eta[8]     0.06    0.01 0.33  -0.60  -0.15  0.06  0.28  0.71  2354    1
## eta[9]    -0.14    0.01 0.35  -0.84  -0.38 -0.15  0.08  0.56  2350    1
## eta[10]   -0.13    0.01 0.58  -1.30  -0.52 -0.12  0.25  0.99  2020    1
## eta[11]    0.55    0.02 0.85  -1.08  -0.02  0.53  1.11  2.19  2481    1
## eta[12]    0.00    0.01 0.61  -1.20  -0.40 -0.01  0.40  1.17  2421    1
## tau        4.07    0.03 1.19   2.44   3.26  3.84  4.63  7.06  2020    1
## lp__      -9.59    0.07 3.38 -17.02 -11.67 -9.31 -7.23 -3.69  2284    1
##
## Samples were drawn using NUTS(diag_e) at Thu May 18 15:42:29 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

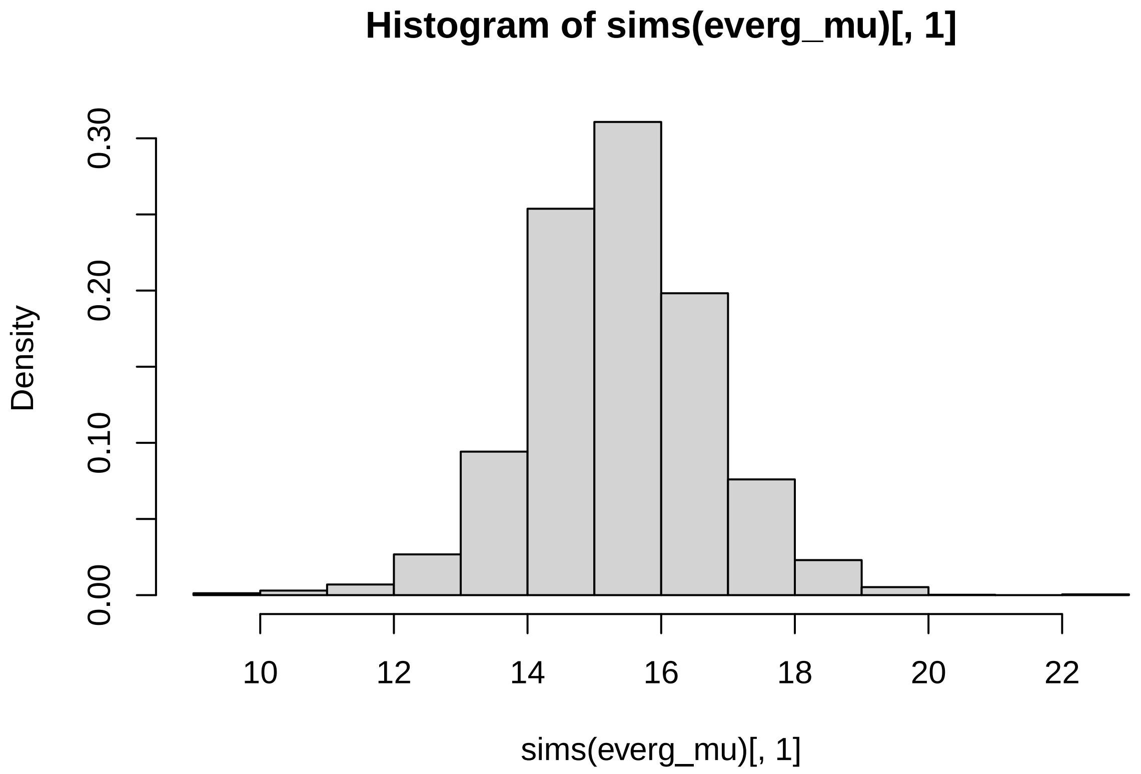
```
everg_fit1 <- rvsims(as.matrix(
    as.data.frame(rstan::extract(fit2keep, permuted=T))))

## 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)
```



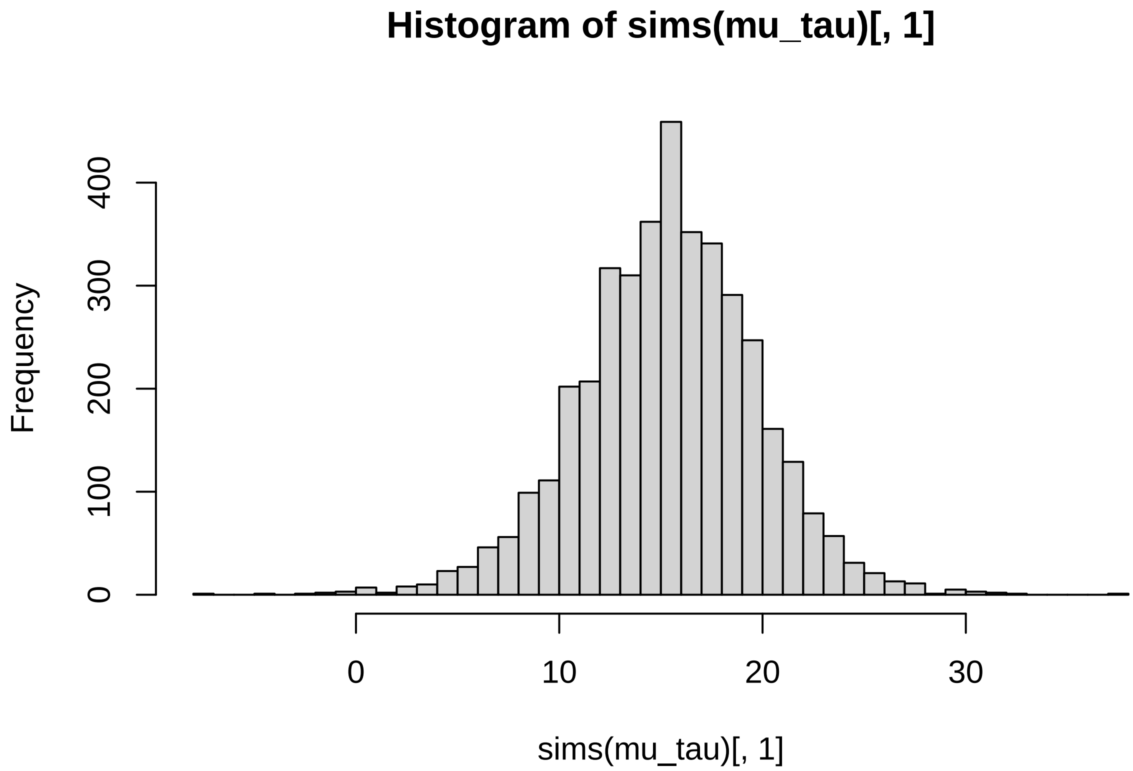
How should we determine the TP concentration standard?

It is mostly an ecological and environmental management question. Statistically, the question is whether we derive the standard based on the overall mean (μ) or the distribution of all metrics: between the posterior distribution of μ and the hyper-distribution.

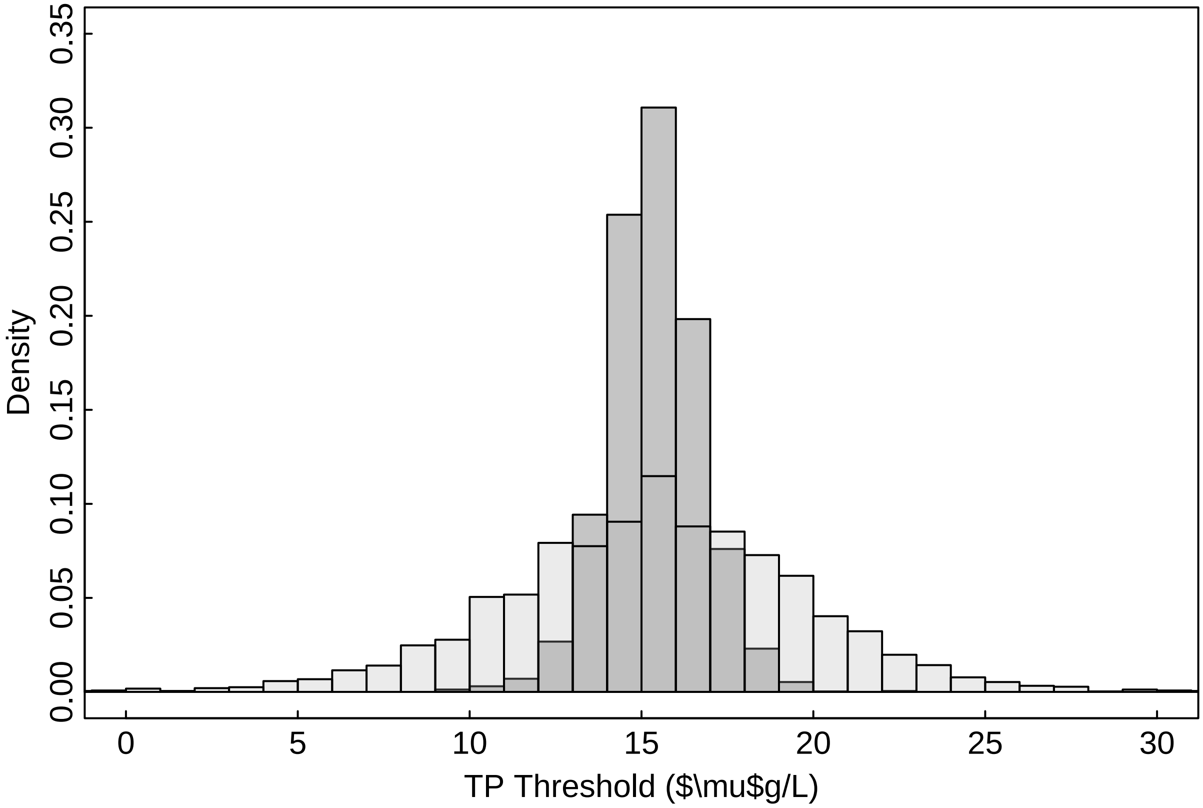
```
## mu versus N(mu, tau)
mu_tau <- rvnorm(1, everg_mu, everg_tau)
p1 <- hist(sims(everg_mu)[,1], freq=F)
```



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



```
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,.1,1/4),
     xlim=c(0,30), ylim=c(0,0.35), freq=F,
     xlab="TP Threshold ($\\mu$g/L)", main="")  # first histogram
plot(p2, col=rgb(.7,.7,.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% 
## 13.23633  8.16431 
```

Hierarchical Structure and Big Data

If we define "big data" as data from multiple sources and represent multiple levels of aggregation, most data we use in our work are big data. For us, the age of big data means the age of hierarchical modeling. When we don't properly address the hierarchical structure of the data, big data can almost always lead to misleading conclusions.

The US National Lake Assessment data

Qian et al (2019) discussed several studies published using data from US EPA's National Lakes Assessment program (NLA). Under NLA, over 3000 lake throughout the 48 contiguous states were surveyed in 2007 and 2012 to collect a large number of variables for assessing ecological status of the nations lakes. Each visited lake was visited at most two times.

EPA researchers published a number of papers using NLA data to derive national nutrient criteria. They typically use lake mean values of relevant variables to establish empirical relationship between variables representing ecological responses (e.g., chlorophyll a and microsystin concentrations) and variables representing nutrient enrichment (e.g., TP and TN concentrations). Qian et al (2019) suggested that such practice is prone to the trap of Simpson's paradox (correlation established at one level of aggregation can be very different from the same correlation at a different level of aggregation). Simpson's paradox is relevant because the nutrient criteria are established at a national (spatial) aggregated level, whereas the resulting criteria must be implemented at individual lakes over time.

Statistically speaking, we must properly model the hierarchical structure represented by the data. By data hierarchical structure, we mean the observation values and their attributes. In a typical dataset (think about an Excel spreadsheet format), we arrange data in a two dimensional array, rows representing observations and columns representing variables. In R, we use data frame (the most commonly used format). In both cases, variables can be classified into measured variables and identification variables. Measured variables are typically numeric and identification variables are categorical in nature. For example, `chla`, `tp`, and `tn` in our data are measured variables and `id` is identification variable. With `id`, we group measured variable into parallel units. (The concept of measured and identification variables is explicitly used in packages from the `tidyverse` family.)

Suppose that we have only measured `chla` from these lakes and the lake ids do not provide lake-specific information. If we want to know the average `chla` values for these lakes, we can assume `log_chla` values can be approximated by the normal distribution and make statistical inference about `chla` from lake j using the model:

$$\log(chla_{ij}) \sim N(\mu_j, \sigma_j^2)$$

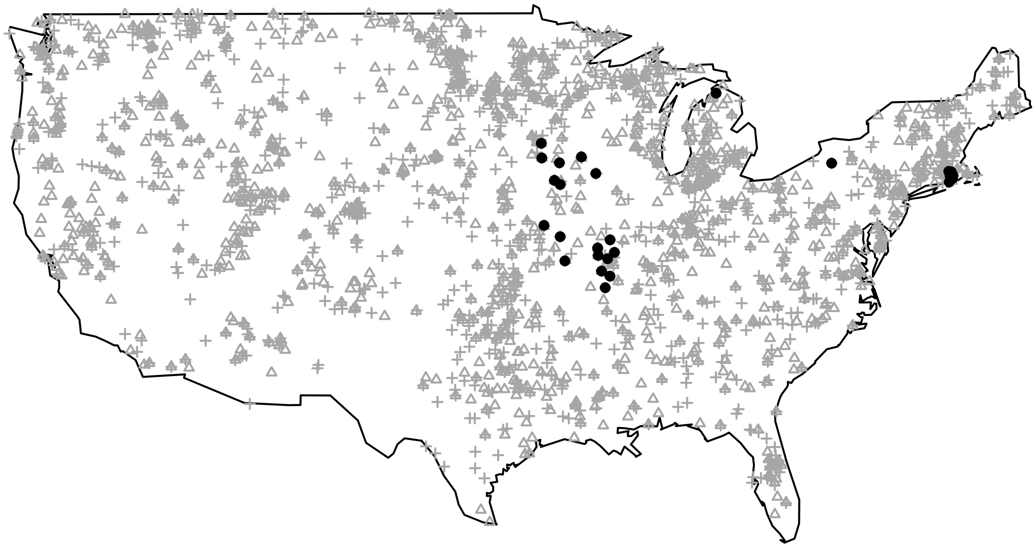
The parameters μ_j and σ_j^2 will be estimated when we have data. In Bayesian statistics, we need to use priors for these unknown parameters. In most cases, we are largely interested in the mean parameters. For a mean parameter, the central limit theorem suggests that the prior of μ_j should also be a normal distribution.

$$\mu_j \sim N(\theta, \tau^2)$$

That is, we must impose priors for all 27 lakes in this problem. Given that we have no specific information about these lakes, we do not know how to determine the likely relative magnitudes of `chla` in these lakes. In other words, we have no reason to give a high (or lower) prior mean for lake 1 than the prior mean for lake 2. As a result, to reflect our ignorance of the relative magnitudes among the lakes, we are compelled to assign a common prior to all 27 lakes. The above equation is the manifestation of our ignorance: we know that lakes mean `chla`s are likely different, but not how they are different from each other. Without further information, we use non-informative priors for θ and τ^2. The hierarchical model is a generalization of Stein's paradox (and James-Stein estimator) in classical statistics. The consequence of imposing this common prior is the shrinkage effect, illustrated in the Everglades example. The lakes are known to be exchangeable with respect to lake-specific μ_j.

To illustrate this problem, Qian et al (2019) use data from lakes shared by NLA and another large lake database (LAGOS) to compare how Simpson's paradox can be manifested in lake eutrophication studies.

The data: USA-NLA and LAGOS. We pick lakes shared in the two data bases.



We selected lakes from LAGOS with at least 10 observations for this analysis. Comparing lake-specific models fit using hierarchical model to the common practice of either combining data from all lakes or fitting a model using lake means. The goal of this example is to illustrate the importance of accounting for data hierarchical structure. The 27 lake selected here have at least 27 observations. We fit the typical log-log linear model of chlorophyll a (*chla*) predicted by total nitrogen (*tn*), total phosphorous (*tp*), and their interaction. Using the TP:TN interaction was inspired by Qian (2016) who suggested that the interaction slope is indicative of a lake's trophic status: a negative (0, positive) interaction slope indicates that the lake is likely euthophic (mesotrophic, oligotrophic).

When we have `tp` and `tn` observations from these lakes, we can no longer claim ignorance because TP and TN are nearly always positively correlated with chla. But when we model the relationship between *chla* and TP and TN using a log-log linear model:

$$\log(chla_{ij}) = \beta_{0j} + \beta_{1j}\log(TP) + \beta_{2j}\log(TP) + \beta_{3j}\log(TP)\log(TN) + \epsilon_{ij}$$

we may claim ignorance on how regression coefficients vary among lakes. As a result, we can impose a common prior for these coefficients:

$$\begin{pmatrix}\beta_{0j}\\ \beta_{1j}\\ \beta_{2j}\\ \beta_{3j}\end{pmatrix} \sim MVN\left[\begin{pmatrix}\mu_0\\ \mu_1\\ \mu_2\\ \mu_3\end{pmatrix}, \Sigma\right]$$

Now we say that these lakes are exchangeable with respect to model coefficients.

The R package `lme4` provides efficient algorithms to estimate these parameters using restricted maximum likelihood method. Although the algorithms are not efficient in estimating the variance parameters, the algorithms are fast and often provide good approximations. We can use `lme4` to explore different model forms to decide how to model the data the best and move the chosen model form to Stan for accurate quantification.

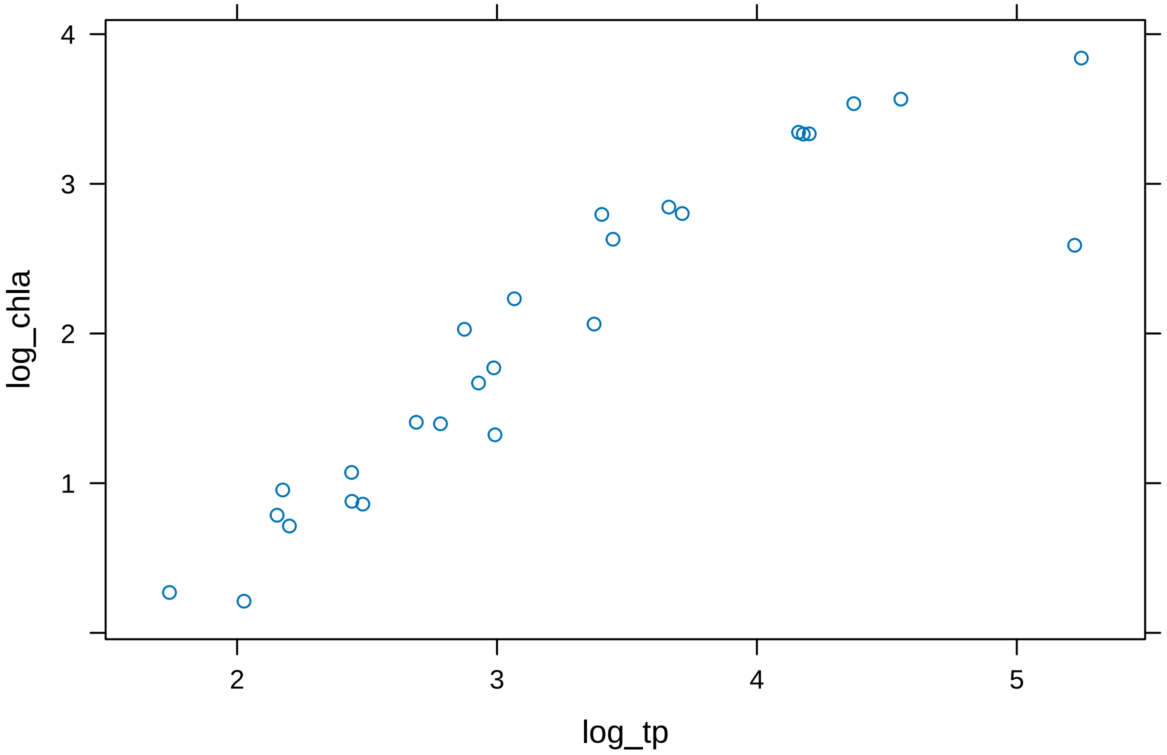
- Comparing different spatial aggregations

```
## fitting hierarchical model for each lake
log_tp_mu <- mean(log(lg_lakes$tp+0.1), na.rm=T)
log_tn_mu <- mean(log(lg_lakes$tn+1), na.rm=T)

lg_lakes_cen <- data.frame(log_chla=log(lg_lakes$chla),
                           log_tp_c=log(lg_lakes$tp+0.1) - log_tp_mu,
                           log_tn_c=log(lg_lakes$tn+1) - log_tn_mu,
                           id=lg_lakes$lagoslakeid)
lg_mlm <- lmer(log_chla ~ log_tp_c + log_tn_c + log_tp_c:log_tn_c +
                 (1+log_tp_c + log_tn_c + log_tp_c:log_tn_c|id),
               data=lg_lakes_cen)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## unable to evaluate scaled gradient
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge: degenerate Hessian with 1 negative eigenvalues
```

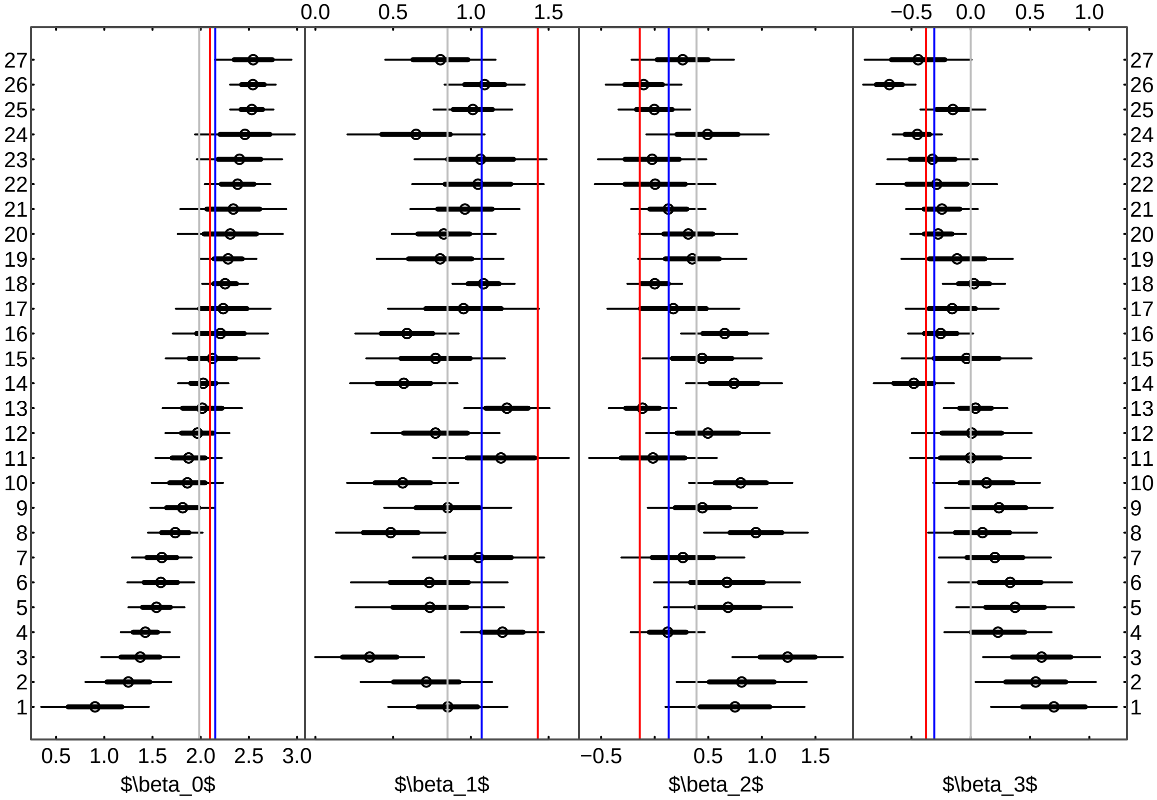
```
## Fitting a single linear regression model using lake means
## US EPA approach
lg_lakes_means <- data.frame(log_chla=tapply(log(lg_lakes$chla), lg_lakes$lagoslakeid, mean, na.rm=T),
                             log_tp=tapply(log(lg_lakes$tp+0.1), lg_lakes$lagoslakeid, mean, na.rm=T),
                             log_tn=tapply(log(lg_lakes$tn+1), lg_lakes$lagoslakeid, mean, na.rm=T))
xyplot(log_chla ~ log_tp, data=lg_lakes_means)
```



```
lg_lakes_means_lm <- lm(log_chla ~ I(log_tp-log_tp_mu)+I(log_tn-log_tn_mu)+
                          I(log_tp-log_tp_mu):I(log_tn-log_tn_mu), data=lg_lakes_means)
lg_mean_lm_coef <- coef(lg_lakes_means_lm)

## fitting using all observations (complete mixing)
lg_lakes_cen <- data.frame(log_chla=log(lg_lakes$chla),
                           log_tp_c=log(lg_lakes$tp+0.1) - log_tp_mu,
                           log_tn_c=log(lg_lakes$tn+1) - log_tn_mu,
                           id=lg_lakes$lagoslakeid)
lg_lakes_lm <- lm(log_chla ~ log_tp_c + log_tn_c + log_tp_c:log_tn_c,
               data=lg_lakes_cen)
lg_lm_coef <- coef(lg_lakes_lm)
```

Now we compare the estimated coefficients:

```
line.plots <- function(est, se, yaxis=NULL, hline=0, HL=T,
                       oo=NULL, Outer=F, xloc=1, yaxisLab=NULL, ...){
    n <- length(est)
    if (!is.null(oo)) {
        est<-est[oo]
        se <-se[oo]
    }
    if(n != length(se))stop("lengths not match")
    plot(1:n, 1:n, xlim=range(c(est+2*se, est-2*se)),
         ylim=c(0.75, n+0.25),
         type="n", axes=F, ...)
    axis(xloc)
    axis(side=c(1,3)[c(1,3)!=xloc], labels = F)
    if (!is.null(yaxis))
      axis(yaxis, at=1:n, labels=yaxisLab, las=1, outer=Outer)
    segments(y0=1:n, y1=1:n, x0=est-2*se, x1=est+2*se)
    segments(y0=1:n, y1=1:n, x0=est-1*se, x1=est+1*se, lwd=2.5)
    points(est, 1:n)
    if (HL) abline(v=hline, col="gray")
    invisible()
}

## all lakes, by lake
est <- t(fixef(lg_mlm) + t(as.matrix(ranef(lg_mlm)[["id"]])))
se <- sqrt(t(se.fixef(lg_mlm)^2+t(as.matrix(se.ranef(lg_mlm)[["id"]]))^2))
oo <- order(est[,1])

par(mfrow=c(1,4), mgp=c(1.25,0.125,0), oma=c(0, 3, 0, 3),
    tck=0.01, las=1, mar=c(3, 0, 3, 0))
line.plots(est[oo,1], se[oo,1], yaxis=2, hline=fixef(lg_mlm)[1], yaxisLab=1:27, xlab="$\\beta_0$")
abline(v=lg_mean_lm_coef[1], col="red")
abline(v=lg_lm_coef[1], col="blue")
box(col=grey(0.3))
line.plots(est[oo,2], se[oo,2], yaxisLab=1:27,
           hline=fixef(lg_mlm)[2], xloc=3, xlab="$\\beta_1$")
abline(v=lg_mean_lm_coef[2], col="red")
abline(v=lg_lm_coef[2], col="blue")
box(col=grey(0.3))
line.plots(est[oo,3], se[oo,3], yaxisLab=1:27,
           hline=fixef(lg_mlm)[3], xlab="$\\beta_2$")
abline(v=lg_mean_lm_coef[3], col="red")
abline(v=lg_lm_coef[3], col="blue")
box(col=grey(0.3))
line.plots(est[oo,4], se[oo,4], yaxisLab=1:27, xlab="$\\beta_3$",
           yaxis=4, xloc = 3)
abline(v=lg_mean_lm_coef[4], col="red")
abline(v=lg_lm_coef[4], col="blue")
box(col=grey(0.3))
```



- Comparing teporal aggregations Examining the temporal scale aggregation of the three lake with long time series

```
lg_lakes_long <- sharedLakes$lagoslakeid[sharedLakes$lg_n>100]
lg_lakes_long <- lg_nutr[is.element(lg_nutr$lagoslakeid, lg_lakes_long), ]
lg_lakes_long$date <- as.Date(lg_lakes_long$sampledate, format="%m/%d/%Y")



lake1 <- lg_lakes_long[lg_lakes_long$lagoslakeid==unique(lg_lakes_long$lagoslakeid)[1],]
lake1$log_chla <- log(lake1$chla)
lake1$log_tp_c <- log(lake1$tp+0.1) - log_tp_mu
lake1$log_tn_c <- log(lake1$tn+1) - log_tn_mu

lake2 <- lg_lakes_long[lg_lakes_long$lagoslakeid==unique(lg_lakes_long$lagoslakeid)[2],]
lake2$log_chla <- log(lake2$chla)
lake2$log_tp_c <- log(lake2$tp+0.1) - log_tp_mu
lake2$log_tn_c <- log(lake2$tn+1) - log_tn_mu
```

```
lake3 <- lg_lakes_long[lg_lakes_long$lagoslakeid==unique(lg_lakes_long$lagoslakeid)[3],]
lake3$log_chla <- log(lake3$chla)
lake3$log_tp_c <- log(lake3$tp+0.1) - log_tp_mu
lake3$log_tn_c <- log(lake3$tn+1) - log_tn_mu

lake1_mlm <- lmer(log_chla ~ log_tp_c + log_tn_c + log_tp_c:log_tn_c +(1+log_tp_c+log_tn_c+log_tp_c:log
```

```
## boundary (singular) fit: see help('isSingular')
```

```
lake2_mlm <- lmer(log_chla ~ log_tp_c + log_tn_c + log_tp_c:log_tn_c +(1+log_tp_c+log_tn_c+log_tp_c:log
```

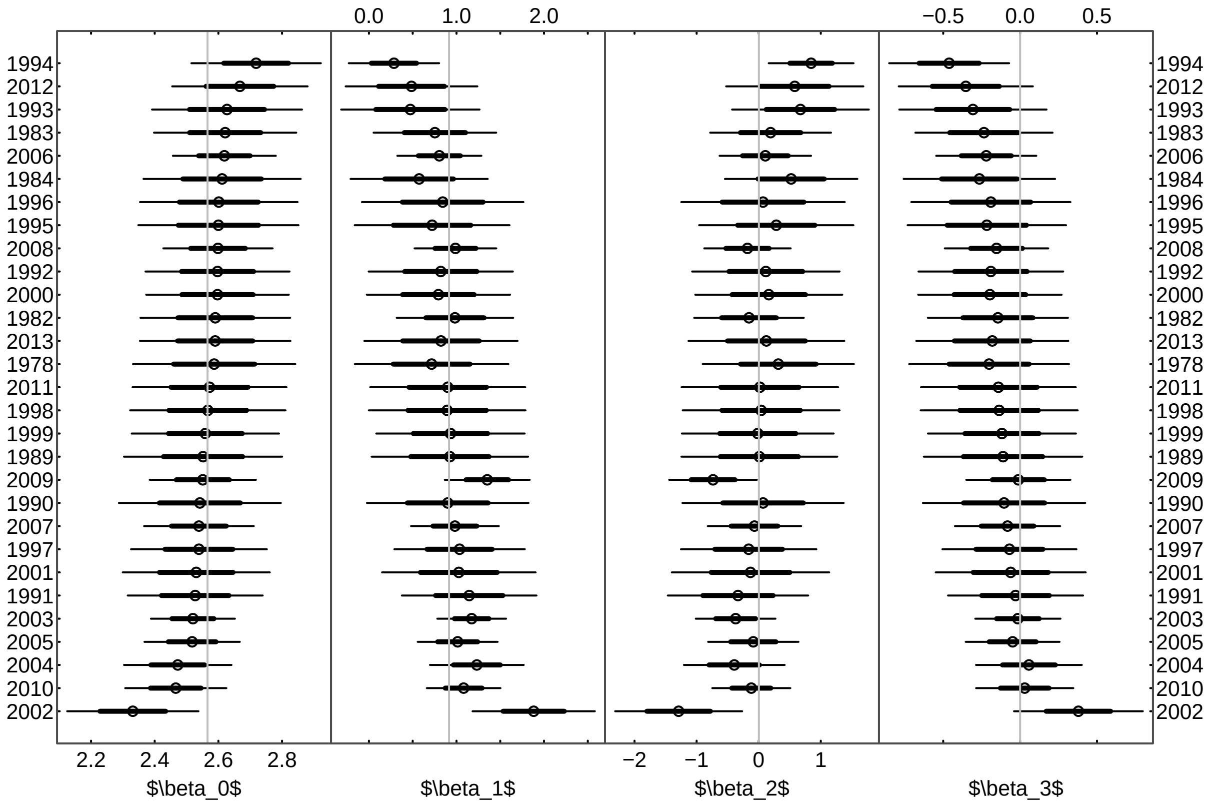
```
## boundary (singular) fit: see help('isSingular')
```

```
lake3_mlm <- lmer(log_chla ~ log_tp_c + log_tn_c + log_tp_c:log_tn_c +(1+log_tp_c+log_tn_c+log_tp_c:log
```

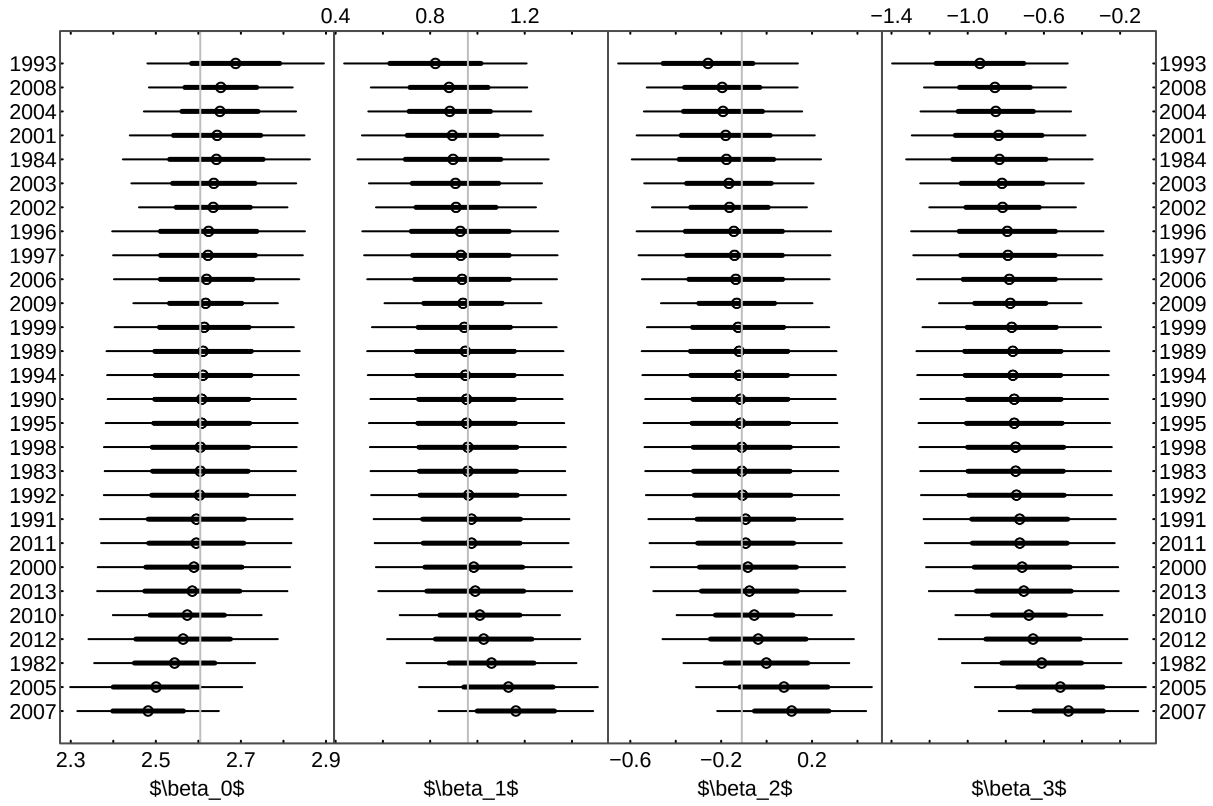
```
## boundary (singular) fit: see help('isSingular')
```

Graphical comparisons

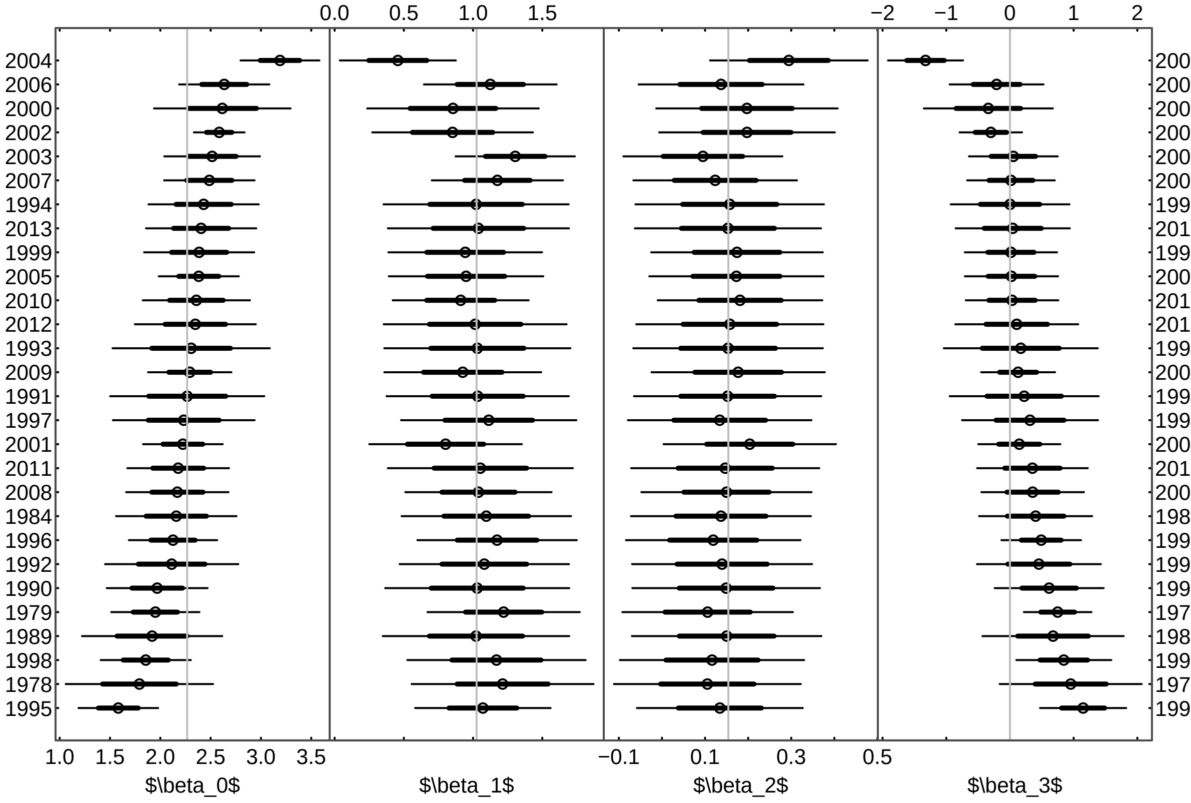
```
## lake 1 by year
est <- t(fixef(lake1_mlm) + t(as.matrix(ranef(lake1_mlm)[["sampleyear"]])))
se <- sqrt(t(se.fixef(lake1_mlm)^2+t(as.matrix(se.ranef(lake1_mlm)[["sampleyear"]]))^2))
oo <- order(est[,1])
ylb <- row.names(ranef(lake1_mlm)[["sampleyear"]])

par(mfrow=c(1,4), mgp=c(1.25,0.125,0), oma=c(0, 3, 0, 3),
    tck=0.01, las=1, mar=c(3, 0, 3, 0))
line.plots(est[oo,1], se[oo,1], yaxisLab=ylb[oo],
           yaxis=2, hline=fixef(lake1_mlm)[1], xlab="$\\beta_0$")
box(col=grey(0.3))
line.plots(est[oo,2], se[oo,2], yaxisLab=ylb[oo], xlab="$\\beta_1$",
           hline=fixef(lake1_mlm)[2], xloc=3)
box(col=grey(0.3))
line.plots(est[oo,3], se[oo,3], yaxisLab=ylb[oo], xlab="$\\beta_2$",
           hline=fixef(lake1_mlm)[3])
box(col=grey(0.3))
line.plots(est[oo,4], se[oo,4], yaxisLab=ylb[oo], xlab="$\\beta_3$",
           yaxis=4, xloc=3)
box(col=grey(0.3))
```



```
## lake 2 by year
est <- t(fixef(lake2_mlm) + t(as.matrix(ranef(lake2_mlm)[["sampleyear"]])))
se <- sqrt(t(se.fixef(lake2_mlm)^2+t(as.matrix(se.ranef(lake2_mlm)[["sampleyear"]]))^2))
oo <- order(est[,1])
ylb <- row.names(ranef(lake2_mlm)[["sampleyear"]])

par(mfrow=c(1,4), mgp=c(1.25,0.125,0), oma=c(0, 3, 0, 3),
    tck=0.01, las=1, mar=c(3, 0, 3, 0))
line.plots(est[oo,1], se[oo,1], yaxisLab =ylb[oo], xlab="$\\beta_0$",
           yaxis=2, hline=fixef(lake2_mlm)[1])
box(col=grey(0.3))
line.plots(est[oo,2], se[oo,2], yaxisLab=ylb[oo], xlab="$\\beta_1$",
           hline=fixef(lake2_mlm)[2], xloc=3)
box(col=grey(0.3))
line.plots(est[oo,3], se[oo,3], yaxisLab =ylb[oo], xlab="$\\beta_2$",
           hline=fixef(lake2_mlm)[3])
box(col=grey(0.3))
line.plots(est[oo,4], se[oo,4], yaxisLab =ylb[oo], xlab="$\\beta_3$",
           yaxis=4, xloc = 3)
box(col=grey(0.3))
```



```
## lake 3 by year
est <- t(fixef(lake3_mlm) + t(as.matrix(ranef(lake3_mlm)[["sampleyear"]])))
se <- sqrt(t(se.fixef(lake3_mlm)^2+t(as.matrix(se.ranef(lake3_mlm)[["sampleyear"]]))^2))
oo <- order(est[,1])
ylb <- row.names(ranef(lake3_mlm)[["sampleyear"]])

par(mfrow=c(1,4), mgp=c(1.25,0.125,0), oma=c(0, 3, 0, 3),
    tck=0.01, las=1, mar=c(3, 0, 3, 0))
line.plots(est[oo,1], se[oo,1], yaxisLab=ylb[oo], xlab="$\\beta_0$",
           yaxis=2, hline=fixef(lake3_mlm)[1])
box(col=grey(0.3))
line.plots(est[oo,2], se[oo,2], yaxisLab =  ylb[oo], xlab="$\\beta_1$",
           hline=fixef(lake3_mlm)[2], xloc = 3)
box(col=grey(0.3))
line.plots(est[oo,3], se[oo,3], yaxisLab =  ylb[oo], xlab="$\\beta_2$",
           hline=fixef(lake3_mlm)[3])
box(col=grey(0.3))
line.plots(est[oo,4], se[oo,4], yaxisLab =  ylb[oo], xlab="$\\beta_3$",
           yaxis=4, xloc = 3)
box(col=grey(0.3))
```



See Section 6.4.3 of Qian et al (2022) for details on programming multilevel models in Stan.

Why Simpson's paradox

What is the cause of Simpson's paradox?

There are numerous discussions on the causes of and the means to avoiding Simpson's paradox. We discussed two main lines of arguments. Lindley and Novick (1981) emphasized the concept of exchangeable units, suggesting that the fallacy lies in applying results of a model to subjects that are not exchangeable with the data used for model development. Pearl et al (2016) stressed the importance of properly outlining the causal structure of the problem, particularly, identifying hidden causes. We use the paper by Cheng and Basu (2017) to illustrate the importance of these two lines of arguments.

Cheng and Basu (2017) compiled a data set of 600 lentic water bodies (lakes, reservoirs, and wetlands) around the globe from several dozen studies, including the North American Treatment Database (NATD) v2.0 for constructed wetlands. In NATD, most wetlands were represented by a small number of records which were often the temporal (e.g., annual) and spatial (e.g., segments) averages of each of the key relevant factors examined in their study (i.e., flow, hydraulic residence time, and nutrient loading). Using the data, they calculated, for each water, the nutrient retention as a ratio of the amount nutrient retained in the water over the input loading:

$$R = \frac{M_{in} - M_{out}}{M_{in}}$$

where M_{in} is the input mass loading and M_{out} is the output loading. In addition, they estimated two parameters that are part of water quality models commonly used to simulate the fate and transport of contaminants. Specifically for model phosphorus retention in wetlands, they are the effective removal rate constant k and the hydraulic residence time τ. In a typical simplified water quality model based on the first-order reaction mechanism, these two parameters are used to estimate nutrient retention: - Assuming the water is well mixed, use the continuously stirred tank reactor (CSTR) model

$$k = \frac{R}{1-R}\left(\frac{1}{\tau}\right).$$

- Assuming the water flows from inlet to outlet without longitudinal diffusion and dispersion, use the plug-flow reactor (PFR) model

$$k = \log(1-R)\left(\frac{1}{\tau}\right).$$

Once k and τ were estimated separately for each wetland, lake, and reservoir, Cheng and Basu (2017) fit a regression model using τ as the predictor variable and k as the response variable:

$$\log(k_j) = \beta_0 + \beta_1 \log(\tau_j) + \epsilon_j$$

where j represents individual waters. They showed that the estimated slope β_1 is negative, suggesting that the shorter the hydraulic residence time (τ) is, the larger the phosphorus effective removal rate constant (k) is. Because a wetland's τ is positively correlated with its surface area, Cheng and Basu (2017) concluded that small wetlands are more effective in removing phosphorus over the landscape than large wetlands on a per unit area basis (k).

```
CB_data <- read.csv(paste(dataDIR, "ChengBasu.csv", sep="/"))
CB_data$k_TP_CSTR <- as.numeric(as.character(CB_data$k_TP_CSTR))
```

```
## Warning: NAs introduced by coercion
```

```
CB_data$k_TP_PFR <- as.numeric(as.character(CB_data$k_TP_PFR))

CB_data$Wetland <- 1
CB_data$Wetland[substring(CB_data$Type, 2, 2)!="W"] <- 0
```

We illustrate the issues with this analysis in two steps.

First, we examine the meaning of model coefficients using the exchangeable concept. The model in the previous equation is inevitably a model for individual waters. As a result, fitting the model using combined data from lakes, reservoirs, and wetlands combines nonexchangeable units together and is susceptible to Simpson's paradox. We fit the same model using combined data from lakes, reservoirs, and wetlands, and compare the resulting model coefficients to the coefficients from the same model fit to data from lakes, reservoirs, and wetlands separately. The slope estimated using the combined data is much lower than the slopes estimated using data from the three types of water separately:

```
## all data
lm1_all_CSTR <- lm(log(k_TP_CSTR) ~ log(HRT_tau), data=CB_data,
               subset=k_TP_CSTR>0)
```

```
## Warning in log(k_TP_CSTR): NaNs produced
```

```
lm1_all_PFR <- lm(log(k_TP_PFR) ~ log(HRT_tau), data=CB_data,
               subset=k_TP_PFR>0)
```

```
## Warning in log(k_TP_PFR): NaNs produced
```

```
lm1_lake_CSTR <- lm(log(k_TP_CSTR) ~ log(HRT_tau), data=CB_data,
                      subset=k_TP_CSTR>0 & Type=="Lake")
```

```
## Warning in log(k_TP_CSTR): NaNs produced
```

```
lm1_lake_PFR <- lm(log(k_TP_PFR) ~ log(HRT_tau), data=CB_data,
                      subset=k_TP_PFR>0 & Type=="Lake")
```

```
## Warning in log(k_TP_PFR): NaNs produced
```

```
lm1_res_CSTR <- lm(log(k_TP_CSTR) ~ log(HRT_tau), data=CB_data,
                     subset=k_TP_CSTR>0 & Type=="Reservoir")
```

```
## Warning in log(k_TP_CSTR): NaNs produced
```

```
lm1_res_PFR <- lm(log(k_TP_PFR) ~ log(HRT_tau), data=CB_data,
                     subset=k_TP_PFR>0 & Type=="Reservoir")
```

```
## Warning in log(k_TP_PFR): NaNs produced
```

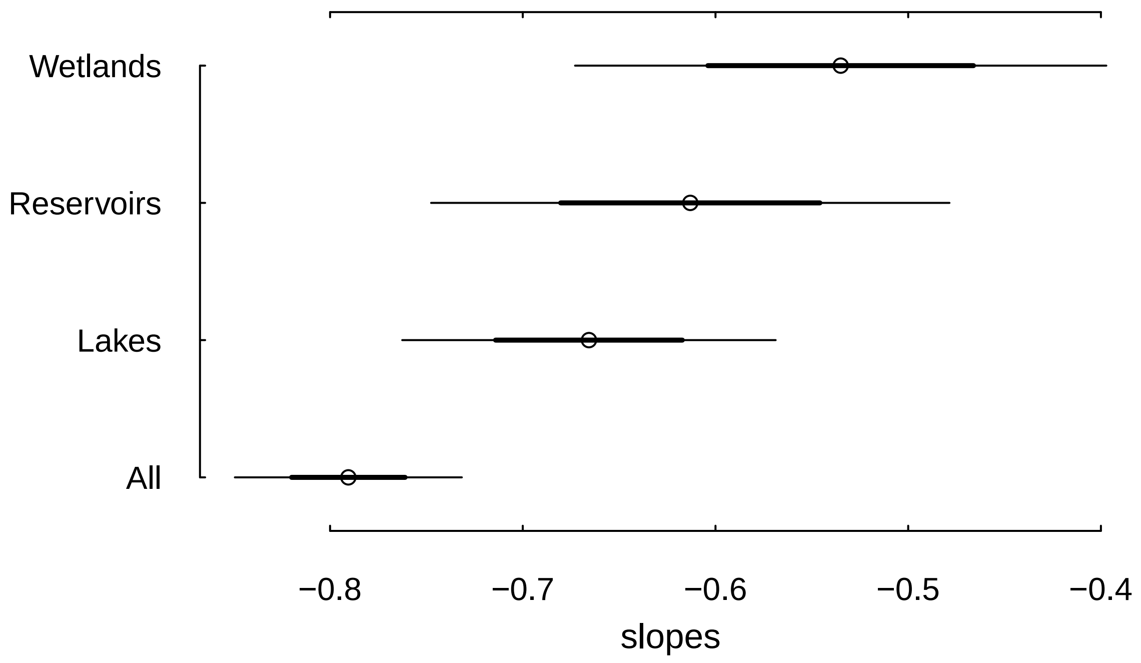
```
lm1_wet_CSTR <- lm(log(k_TP_CSTR) ~ log(HRT_tau), data=CB_data,
                     subset=k_TP_CSTR>0 & Wetland==1)
```

```
## Warning in log(k_TP_CSTR): NaNs produced
```

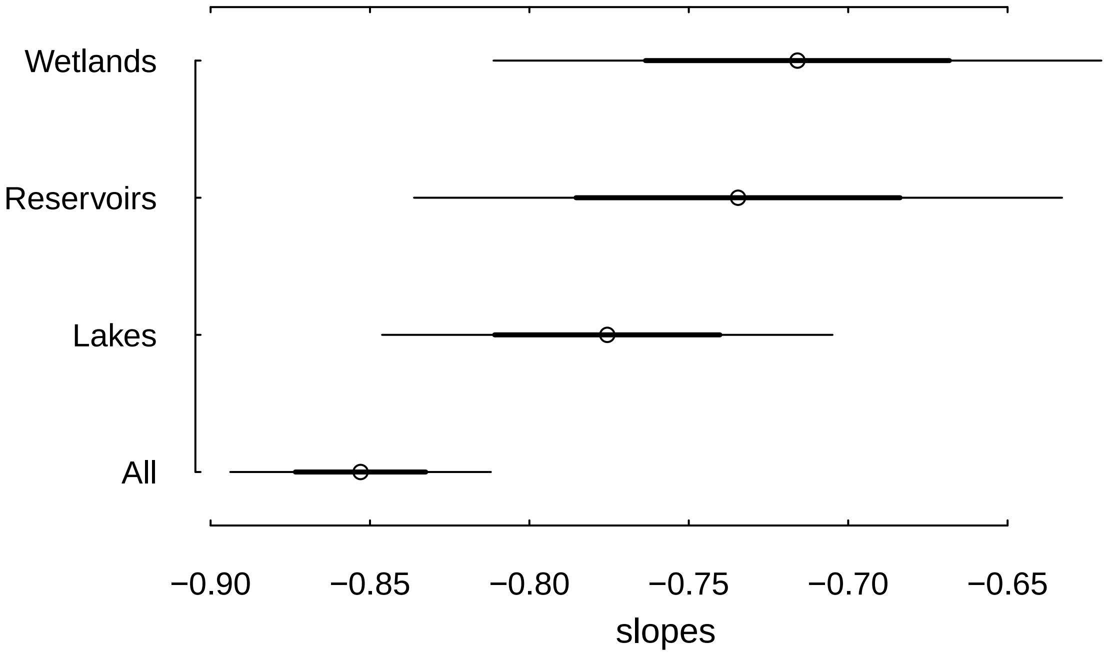
```
lm1_wet_PFR <- lm(log(k_TP_PFR) ~ log(HRT_tau), data=CB_data,
                     subset=k_TP_PFR>0 & Wetland==1)
```

```
## Warning in log(k_TP_PFR): NaNs produced
```

```
## Figure 1 -- aggregated slopes
slp_cstr <- rbind(
    summary(lm1_all_CSTR)$coef[2,1:2],
    summary(lm1_lake_CSTR)$coef[2,1:2],
    summary(lm1_res_CSTR)$coef[2,1:2],
    summary(lm1_wet_CSTR)$coef[2,1:2])
row.names(slp_cstr)  <- c("All", "Lakes", "Reservoirs","Wetlands")

slp_pfr <- rbind(
    summary(lm1_all_PFR)$coef[2,1:2],
    summary(lm1_lake_PFR)$coef[2,1:2],
    summary(lm1_res_PFR)$coef[2,1:2],
    summary(lm1_wet_PFR)$coef[2,1:2])
row.names(slp_pfr)  <- c("All", "Lakes", "Reservoirs","Wetlands")

par(mar=c(5,6,4, 2), mgp=c(2.25,1,0), tck=0.01, cex.main=1.3, cex.lab=1.1)
line.plots(slp_cstr[,1], slp_cstr[,2], yaxis=2, ylab="", xlab="slopes",
           yaxisLab =  rownames(slp_cstr))
```



```
par(mar=c(5,6,4, 2), mgp=c(2.25,1,0), tck=0.01, cex.main=1.3, cex.lab=1.1)
line.plots(slp_pfr[,1],slp_pfr[,2], yaxis=2, xlab="slopes", ylab="",
           yaxisLab =  rownames(slp_pfr))
```

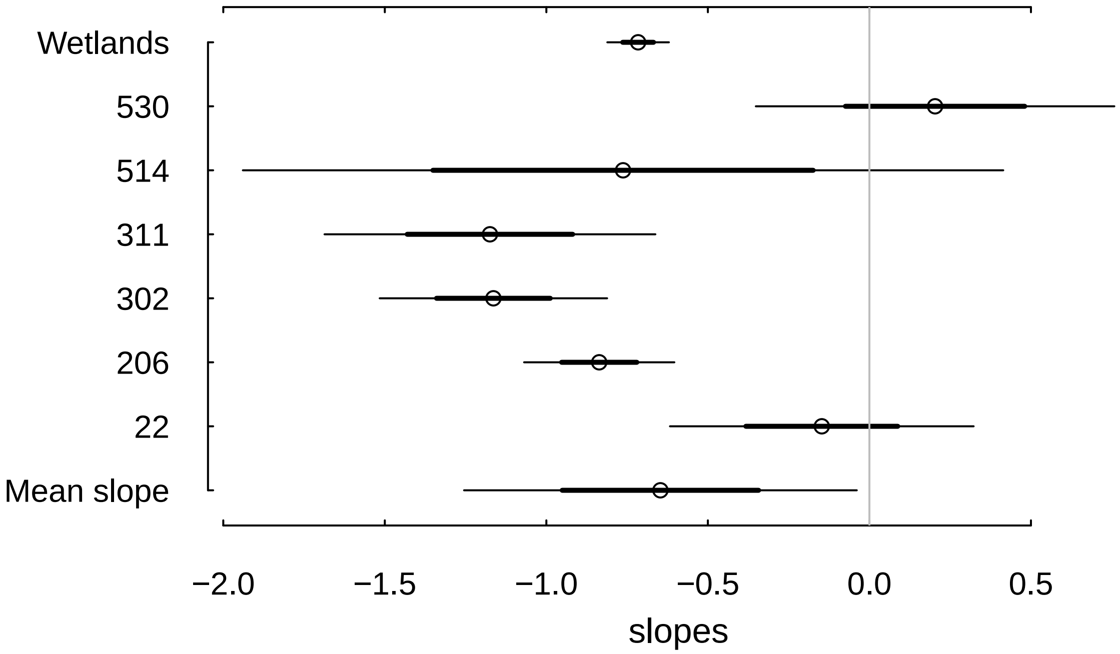


We can further fit the model to data from individual wetlands. Using six wetlands in the database (with more than 10 observations), we estimate the wetland-specific slopes using a hierarchical model assuming the wetland-specific regression coefficients are exchangeable. The six wetlands range in mean size (by volume) from 5.24 to 47,585.27 m^3. Using the concept of exchangeable units, we recognize that observations from the wetland with an average volume of 5.24 m^3 cannot be exchangeable with observations from the wetland with an average volume of 47,585.27 m^3. Consequently, we cannot directly combine data from these wetlands. However, by assuming individual wetlands are exchangeable with respect to the regression model coefficients, we can partially pool data from multiple wetlands using a hierarchical model. The slopes for the smallest and the largest wetlands are not different from 0 (larger than the slope estimated using the combined wetland data), while the slopes of the four intermediate sized wetlands are either highly uncertain (wetland 514) or well below the slope estimated using the combined wetland data.

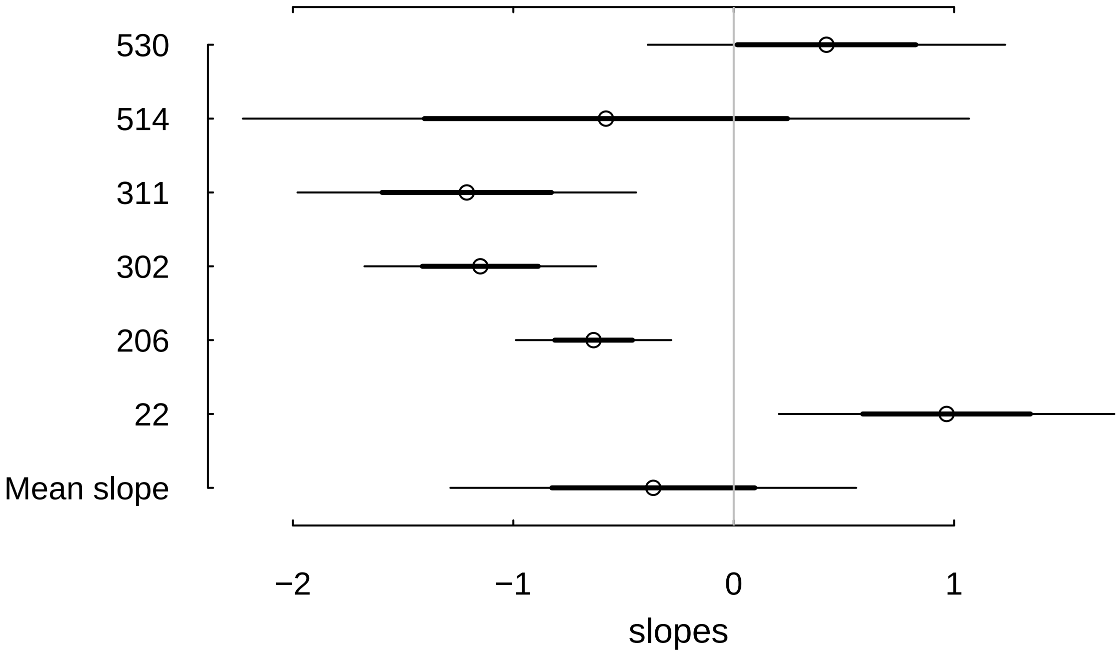
```
CB_wetland <- CB_data[CB_data$Wetland==1,]
CB_wetland$sites <- as.numeric(CB_wetland$Year)
## we determined that Year was mislabeled

CB_NADB <- CB_wetland[CB_wetland$sites<1000,] ## small wetlands
CB_NADB2 <- CB_NADB[CB_NADB$sites %in%
              names(table(CB_NADB$sites)[table(CB_NADB$sites)>10]),
                c("sites", "HRT_tau", "k_TP_PFR","k_TP_CSTR")]
## more than 10 observations
CB_NADB2 <- CB_NADB2[!is.na(CB_NADB2$k_TP_CSTR)&CB_NADB2$k_TP_CSTR>0,]
table(CB_NADB2$sites)
```

```
##
##  22 206 302 311 514 530
##  16  34  31  26  12  27
```

```
lmer_nadb_PFR <- lmer(log(k_TP_PFR) ~ log(HRT_tau) + (1+log(HRT_tau)|sites), data=CB_NADB2)

lmer_nadb_CSTR <- lmer(log(k_TP_CSTR) ~ log(HRT_tau) + (1+log(HRT_tau)|sites), data=CB_NADB2)

lmer_cstr_slp <- fixef(lmer_nadb_CSTR)[2] + ranef(lmer_nadb_CSTR)$sites[,2]
lmer_cstr_slp_se <- se.ranef(lmer_nadb_CSTR)$sites[,2]
lmer_slp_cstr <- data.frame(Estimate=c(fixef(lmer_nadb_CSTR)[2],
                                       lmer_cstr_slp),
                            se.estimate=c(se.fixef(lmer_nadb_CSTR)[2],
                                          lmer_cstr_slp_se))
rownames(lmer_slp_cstr)[1] <- "Mean slope"

lmer_pfr_slp <- fixef(lmer_nadb_PFR)[2] + ranef(lmer_nadb_PFR)$sites[,2]
lmer_pfr_slp_se <- se.ranef(lmer_nadb_PFR)$sites[,2]
lmer_slp_pfr <- data.frame(Estimate=c(fixef(lmer_nadb_PFR)[2],
                                      lmer_pfr_slp),
                            se.estimate=c(se.fixef(lmer_nadb_PFR)[2],
                                          lmer_pfr_slp_se))
rownames(lmer_slp_pfr)[1] <- "Mean slope"

lmer_slp_pfr_plot <- rbind(lmer_slp_pfr, slp_pfr[4,])
rownames(lmer_slp_pfr_plot)[8] <- "Wetlands"

par(mar=c(5,6,4, 2), mgp=c(2.25,1,0), tck=0.01, cex.main=1.3, cex.lab=1.1)
line.plots(lmer_slp_pfr_plot[,1],lmer_slp_pfr_plot[,2], yaxis=2, ylab="", xlab="slopes",
           yaxisLab=rownames(lmer_slp_pfr_plot))
```



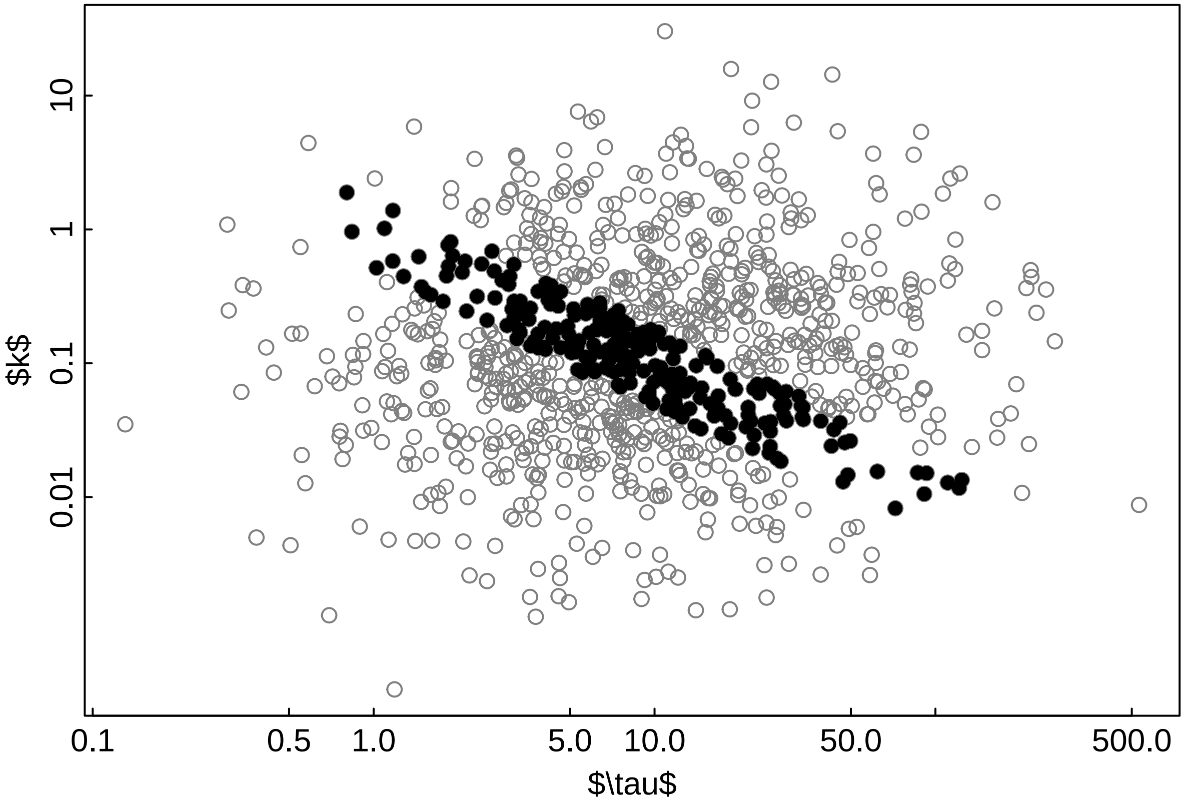
```
par(mar=c(5,6,4, 2), mgp=c(2.25,1,0), tck=0.01, cex.main=1.3, cex.lab=1.1)
line.plots(lmer_slp_cstr[,1],lmer_slp_cstr[,2], yaxis=2, ylab="", xlab="slopes",
           yaxisLab=rownames(lmer_slp_cstr))
```



By now, we recognize that the variation in estimated slopes is a manifestation of Simpson's paradox. The average model coefficients for all wetlands (labeled "Mean slope' ') are parameterized by the hyper-distribution model (i.e., $\beta_j \sim N(\mu_\beta, \sigma^2_\beta)$). The hyper-distribution mean (μ_β) is most likely different from the estimated slope using combined wetland data. The hyper-distribution mean has a clear physical meaning (the mean of the wetland-specific coefficients), whereas the slope estimated using combined data does not.

– spurious correlation Second, we can explain the differences in estimated slopes at different levels of aggregation using causal inference, as in Tang et al (2019). In this case, we don't have additional variables for such analysis. However, we can examine the linear regression model from a causal analysis angle. The two parameters in question (k and τ) represent two different aspects of a wetland and they are most likely independent of each other (Carleton and Montas, 2007; Hejzlar et al, 2007; Vollenweider 1975). In this case, the link between the two parameters is established by the percent removal (R in the CSTR or PFR models). The parameter k reflects the intrinsic characteristics of a wetland, while τ is a parameter determined by external input of water relative to the size of the wetland. The percent removal is a function of both k and τ (approximating the amount of time the nutrient mass stays in the system). In other words, the causal diagram for wetland phosphorus removal should be represented as $k \rightarrow\ R \leftarrow\ \tau$, which means that k and τ together determine R, but k and τ are independent of each other. A spurious correlation between k and τ arises when R is set to vary within a narrow range. In computer science literature, k and τ are known to be direction-separated (d-separated) by R. If two variables are d-separated, the apparent correlation between them is most likely spurious. We can use a simulation to demonstrate the effect of this d-separation. We randomly generate values of k and τ to calculate R using the CSTR model plus random noise ($R_i = k_i\tau_i/(1 + k_i\tau_i) + \epsilon_i$). In this case, the parameters k_i and τ_i were independently drawn from log-normal distributions with log means ($\mu_k = -2.726$ and $\mu_\tau = 1.914$) and log standard deviations ($\sigma_k = 1.371$ and $\sigma_\tau = 1.269$) calculated from the log values for k and τ for TP from the data used by Cheng and Basu (2017). We then use a scatter plot of the randomly drawn k and τ to show the spurious correlation by highlighting the data points with k and τ values resulted in R values between 32% and 65%. Cheng and Basu (2017) indicated that wetlands with a percent removal (R) between 32% and 65% are of "no significant differences between systems and across constituents.' ' The (spurious) negative correlation between k and τ shown by the highlighted data points in the following Figure is remarkably similar to the pattern reported in Cheng and Basu (2017) in their data analysis, which suggests that the conclusion that small wetlands are more effective in phosphorus retention on a per unit area basis is a result of the spurious correlation induced by the d-separated relationship between k and τ.

```
mu_k <- mean(log(CB_data$k_TP_CSTR[CB_data$Wetland==1 & CB_data$k_TP_CSTR>0]),
             na.rm=TRUE)
s_k <- sd(log(CB_data$k_TP_CSTR[CB_data$Wetland==1 & CB_data$k_TP_CSTR>0]),
          na.rm=TRUE)

mu_tau <- mean(log(CB_data$HRT_tau[CB_data$Wetland==1 & CB_data$k_TP_CSTR>0]),
               na.rm=TRUE)
s_tau <- sd(log(CB_data$HRT_tau[CB_data$Wetland==1 & CB_data$k_TP_CSTR>0]),
            na.rm=TRUE)

ln_K_TP <- exp(rnorm(1000,mu_k,s_k))
ln_T_TP <- exp(rnorm(1000,mu_tau,s_tau))
R1 <- (ln_K_TP * ln_T_TP)/(1+(ln_K_TP * ln_T_TP))

par(mar=c(3,3,1,1),mgp=c(1.25,0.125,0),tck=0.01)
plot(ln_T_TP,ln_K_TP, log="xy",
     ylab="$k$",
     xlab="$\\tau$",
     col=gray(0.5), axes=F)
axis(1)
axis(2, at=c(0.01,0.1,1,10), labels=c("0.01","0.1","1","10"))
box()
points(ln_T_TP[R1>0.32 & R1<0.65],
       ln_K_TP[R1>0.32 & R1<0.65], pch=16)
```


10
1
0.1
0.01
k
0.1
0.5
1.0
5.0
10.0
50.0
500.0
τ