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

Song S. Qian

6/3/2023

Why Simpson's paradox

What is the cause of Simpson's paradox?

The cause of Simpson's paradox has been extensively discussed, along with strategies to avoid it. We will highlight two main lines of argumentation that shed light on this phenomenon. Lindley and Novick (1981) focused on the concept of exchangeable units and emphasized that the fallacy arises when applying model results to subjects that are not exchangeable with the data used to develop the model. Pearl et al. (2016) emphasized the significance of accurately delineating the causal structure of the problem, particularly in identifying hidden causes. To illustrate the importance of these two lines of arguments, we will refer to the study conducted by Cheng and Basu (2017).

Cheng and Basu (2017) compiled a dataset consisting of 600 lentic water bodies, including lakes, reservoirs, and wetlands, from various studies, such as the North American Treatment Database (NATD) v2.0 for constructed wetlands. In NATD, wetlands were often represented by a small number of records, typically the temporal (e.g., annual) and spatial (e.g., segments) averages of key factors like flow, hydraulic residence time, and nutrient concentration and loading. Using this data, they calculated the nutrient retention for each water body as the ratio of the retained nutrient mass to 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. Additionally, they estimated two parameters commonly used in water quality models to simulate the fate and transport of contaminants for phosphorus retention in wetlands. These parameters are the effective removal rate constant k and the hydraulic residence time τ. In a simplified water quality model based on the first-order reaction mechanism, these parameters are used to estimate nutrient retention. - Assuming the water is well mixed, the continuously stirred tank reactor (CSTR) model is used:

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

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

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

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 found that the estimated slope β_1 was negative, indicating that as the hydraulic residence time (τ) decreases, the phosphorus effective removal rate constant (k) increases. Based

on the positive correlation between a wetland's τ and its surface area, Cheng and Basu (2017) concluded that small wetlands are more effective at removing phosphorus per unit area compared to large wetlands.

```
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
```

First, we examine the interpretation of model coefficients using the concept of exchangeability. The model described in the previous equation is inherently a model for individual water bodies. Therefore, when fitting the model using combined data from lakes, reservoirs, and wetlands, we are combining nonexchangeable units and exposing ourselves to Simpson's paradox. To illustrate this, we fit the same model using the combined data from lakes, reservoirs, and wetlands, and compare the resulting model coefficients with the coefficients obtained from fitting the same model to the data from lakes, reservoirs, and wetlands separately. Interestingly, the slope estimated using the combined data is significantly lower than the slopes estimated using the data from the three types of water bodies 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
```

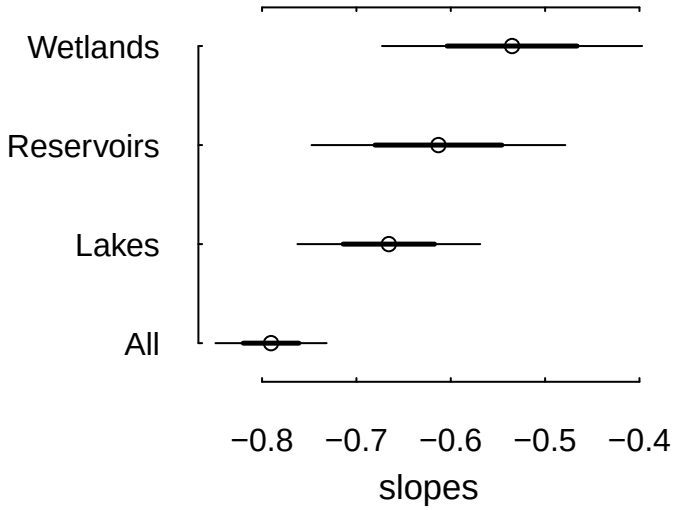
```
lm1_wet_CSTR <- lm(log(k_TP_CSTR) ~ log(HRT_tau), data = CB_data, subset = k_TP_CSTR >
    0 & Wetland == 1)
```

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

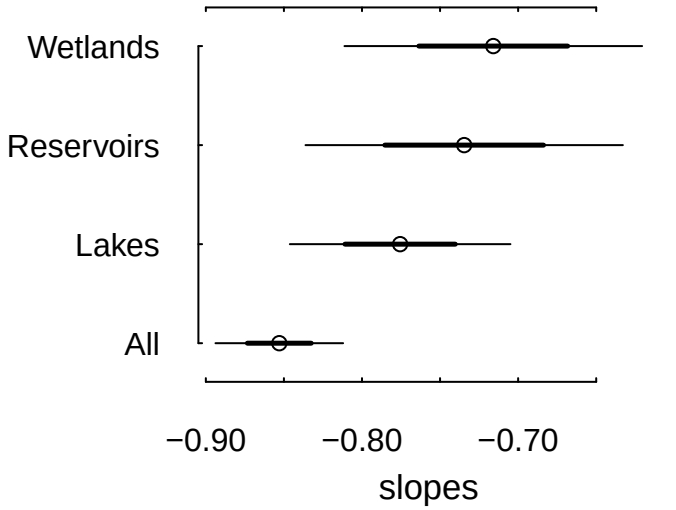
```
lm1_wet_PFR <- lm(log(k_TP_PFR) ~ log(HRT_tau), data = CB_data, subset = k_TP_PFR >
    0 & Wetland == 1)
```

```
## 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
```




```
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_p
```



To gain further insights, we proceed to fit the model using data from individual wetlands. We select six wetlands from the database that have more than 10 observations and estimate wetland-specific slopes using a hierarchical model, assuming that the regression coefficients for each wetland are exchangeable. These six wetlands vary greatly in size, with mean volumes ranging from 5.24 to 47,585.27 m^3.

By considering the concept of exchangeable units, we acknowledge that observations from the wetland with an average volume of 5.24 m^3 cannot be treated as exchangeable with observations from the wetland with an

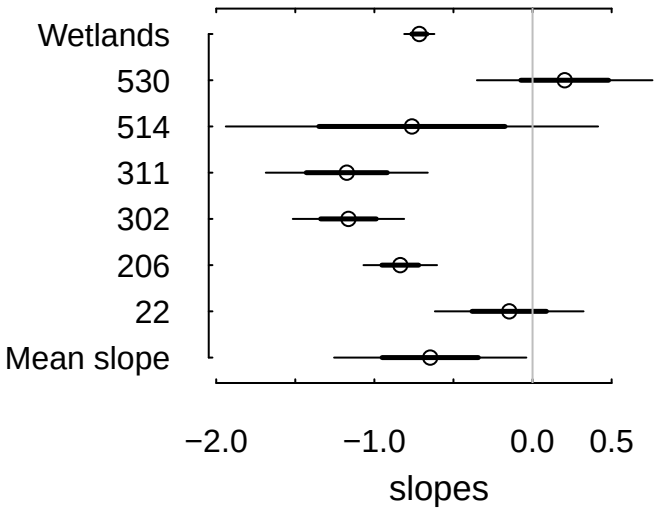
average volume of 47,585.27 m^3. Therefore, direct combination of data from these wetlands is not appropriate. However, by assuming exchangeability of individual wetlands with respect to the regression model coefficients, we can partially pool the data from multiple wetlands using a hierarchical model.

The results reveal interesting patterns. The slopes estimated for the smallest and largest wetlands are not significantly different from 0 (although larger than the slope estimated using the combined wetland data). In contrast, the slopes of the four intermediate-sized wetlands either exhibit high uncertainty (e.g., wetland 514) or are notably smaller than the slope estimated using the combined wetland data.

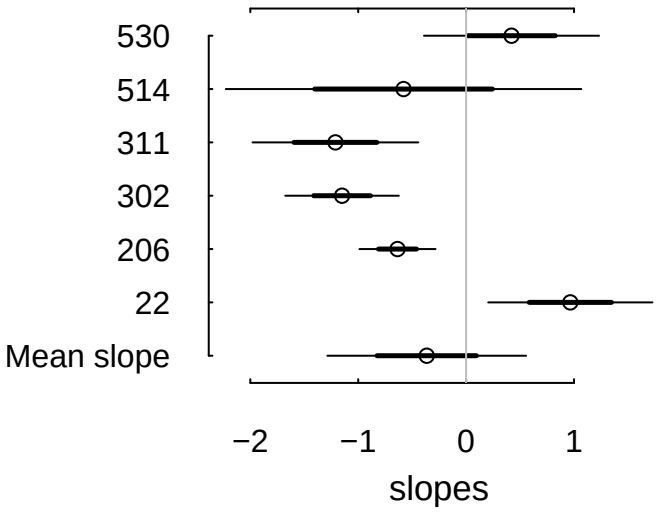
```
CB_wetland <- CB_data[CB_data$Wetland == 1, ]
CB_wetland$sites <- as.numeric(CB_wetland$Year)
## we determined that Year was mislabeled

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.fixe
    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))
```



At this point, it becomes evident that the variation in estimated slopes is a clear manifestation of Simpson's paradox. The average model coefficients for all wetlands, denoted as "Mean slope," are parameterized by the hyper-distribution model, in which the wetland-specific coefficients (β_j) are assumed to follow a normal distribution ($N(\mu_\beta, \sigma_\beta^2)$). It is highly likely that the hyper-distribution mean (μ_β) differs from the slope estimated using the combined wetland data.

The hyper-distribution mean holds a significant physical interpretation as it represents the average of the wetland-specific coefficients. In contrast, the slope estimated using the combined data lacks a clear and meaningful interpretation. Therefore, the hyper-distribution mean provides a more reliable and meaningful estimate of the true underlying relationship between the variables.

By acknowledging Simpson's paradox and employing a hierarchical modeling approach, we can better account for the hierarchical structure of the data and obtain more accurate and interpretable estimates of the model coefficients.

– spurious correlation

Secondly, we can explore the differences in estimated slopes at various levels of aggregation using causal inference, as demonstrated in Tang et al. (2019). Although we lack additional variables for such an analysis,

we can approach the linear regression model from a causal perspective. The two parameters of interest, k and τ, represent distinct aspects of a wetland and are likely independent of each other (Carleton and Montas, 2007; Hejzlar et al., 2007; Vollenweider, 1975).

In this context, the connection between these parameters is established through the percent removal (R) in the CSTR or PFR models. The parameter k reflects the inherent characteristics of a wetland, while τ represents a parameter determined by the external input of water relative to the wetland's size. The percent removal is influenced by both k and τ, as it approximates the duration the nutrient mass remains within the system. In other words, the causal diagram for wetland phosphorus removal can be represented as $k \rightarrow R \leftarrow \tau$, indicating that k and τ jointly determine R but they are independent of each other.

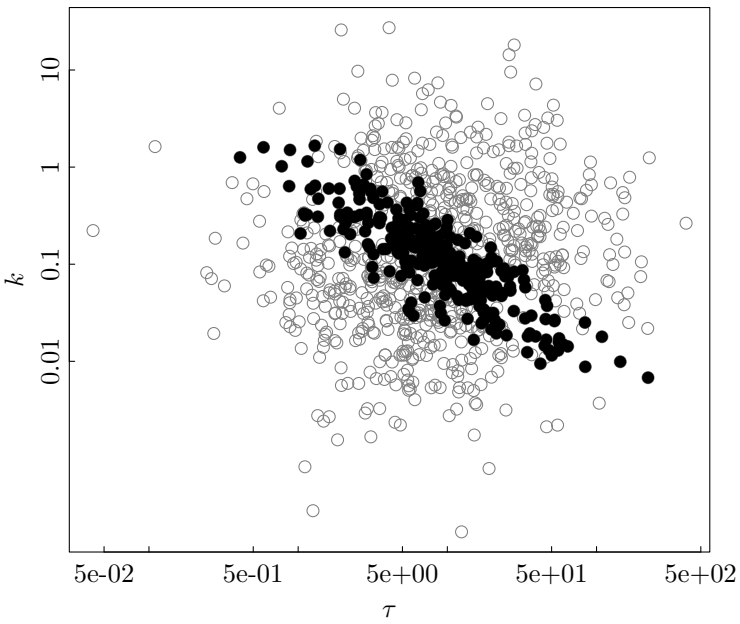
However, a spurious correlation between k and τ can emerge when R exhibits narrow variation. In computer science literature, k and τ are known to be "d-separated" by R. When two variables are d-separated, any apparent correlation between them is likely to be spurious.

To illustrate this effect, we can employ a simulation. We randomly generate values of k and τ and calculate R using the CSTR model and introduce random noise to the resulting R ($R_i = k_i\tau_i/(1 + k_i\tau_i) + \epsilon_i$).

In this simulation, the parameters k_i and τ_i are 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 $\sigma_\tau = 1.269$), derived from the log values of k and τ for TP in the data analyzed by Cheng and Basu (2017). We then create a scatter plot of the randomly generated k and τ values, highlighting the data points where the corresponding R falls between 32% and 65%. Cheng and Basu (2017) indicated that wetlands with a percent removal (R) in this range exhibit "no significant differences between systems and across constituents."

The highlighted data points in the scatter plot demonstrate the (spurious) negative correlation between k and τ, which strikingly resembles the pattern observed in Cheng and Basu (2017) during their data analysis. This similarity suggests that the conclusion stating small wetlands are more effective in phosphorus retention on a per unit area basis may result from 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))
R <- (ln_K_TP * ln_T_TP)/(1 + (ln_K_TP * ln_T_TP))  ##+ (rnorm(1000, 0, 0.1))
## R1 <- R+rnorm(1000, 0, 0.1)
tmp <- logit(R) + rnorm(1000, 0, 0.5)
R1 <- invlogit(tmp)

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
5e-02
5e-01
5e+00
5e+01
5e+02
τ