MWRD PAA

Kathryn B. Newhart

3/28/2019

# Introduction

The Robert W. Hite Treatment Facility, operated by the Metro Wastewater Reclamation District (MWRD) of Denver, CO, treats ~130 million gallons per day (MGD) of wastewater produced by ~2 million people from the Denver-metro area and is the largest wastewater treatment facility in the Rocky Mountain west. In an effort to reduce the cost of disinfection, a peracetic acid (PAA) system was installed to replace the existing chloramine system. However, due to variable influent *E. coli* concentrations to the disinfection system, it has been difficult to optimize the dosing of PAA. In practice, PAA is overdosed to ensure that MWRD is meeting it’s discharge limit. The goal of this work is to identify correlations between upstream operating conditions in the secondary activated sludge system, *E. coli* concentrations, and PAA dosing.

# Goals

Design a PAA disinfection dosing system that account for:

* Upstream secondary treatment performance
* Flowrate through the disinfection basin (i.e., hydraulic retention time or HRT)
* Flow conditions in receiving water body (e.g., low, mid-range, high, dry, moist)

# Questions

1. What effects pre-disinifection *E. coli*?
2. What effects PAA disinfection efficiency?

# Procedure

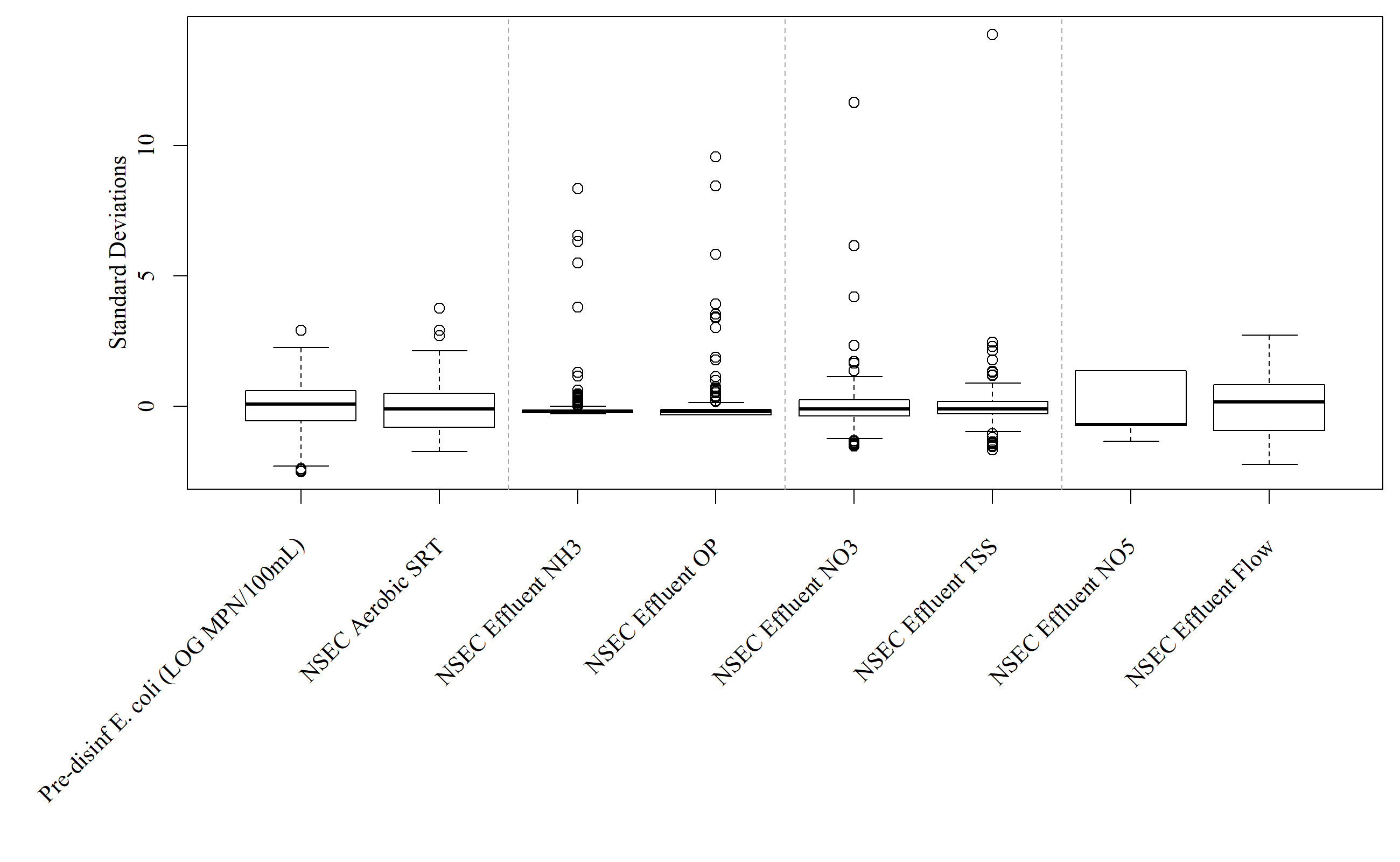
Data was provided by MWRD at a variety of intervals and locations in the treatment process:

|  |  |  |  |
| --- | --- | --- | --- |
| Location | Frequency | Source | Variables |
| North Secondary | 15 min | Sensors | Influent/recirculation/effluent flow, Temperature, Ammonia, TSS, COD, pH, DO, SRT, Nitrate, Ortho-P, Nitrite |
| North Disinfection | 15 min | Sensors | Influent flow, PAA residual, PAA pump flow, PAA setpoint, HRT |
| North Disinfection | Daily | Grab | PAA dose, Upstream residual, Pre-disinfection E. coli, Effluent flow, HRT, Effluent E. coli, CT |
| North Secondary Influent | Daily | 24 hr composite | BOD, Ammonia, TSS |
| North Secondary Effluent | Daily | 24 hr composite | cBOD, Ammonia, TSS |
| North Secondary Influent | 2-3 days | 24 hr composite | COD, Nitrate-nitrite, C:N, C:P, TP, TIN, TKN, TN |
| North Secondary Effluent | 2-3 days | 24 hr composite | Alkalinity, Nitrate-nitrite, C:N, C:P, TP, TIN, TKN, TN |
| North Secondary | 2-5 Days | Grab | SVI, TSS, VSS |
| North Secondary Influent | Weekly | 24 hr composite | Alkalinity, OP |
| North Secondary Influent | Weekly | 24 hr composite | COD, OP |

# Data cleaning

If the distribution of each variable is assumed to be univariate normal and scaled (i.e., zero mean, unit variance), boxplots can be constructed to visualize the range of observations in the dataset. The existance of numerous outliers, heavily shifted on either side of the variable’s median (Figures S1-S3) indicate that the majority of water quality variables are not normally distributed.

To achieve the goal of predicting *E. coli*, a log-transformed pre-disinfection *E. coli* grab data was merged with a reduced North secondary dataset (Figure 1). For simplicity, sensors monitoring SRT and effluent water quality of the North secondary system were used. However, the residence time is need to appropriately merge datasets.

 **Figure 1.** Boxplots of water quality variables used to predict *E. coli*

# Methods

## Regression Models

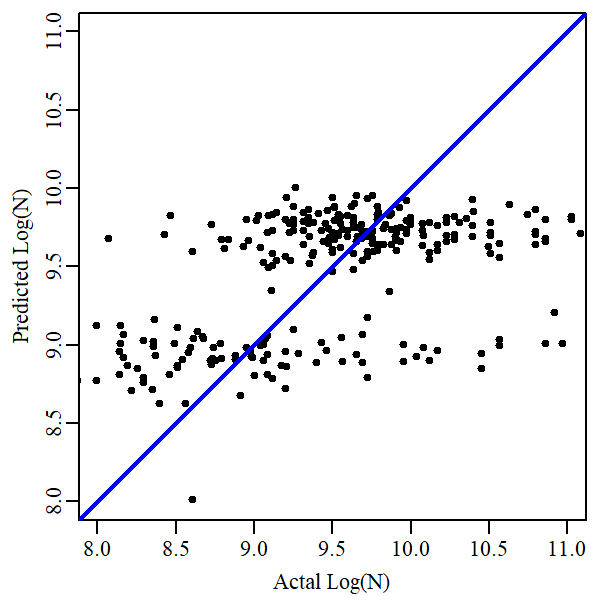
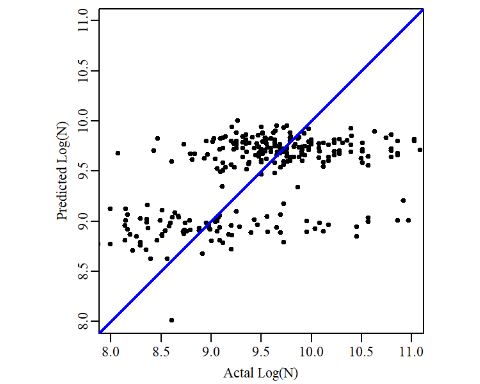
packageLoad("tiff")

## Loading required package: tiff

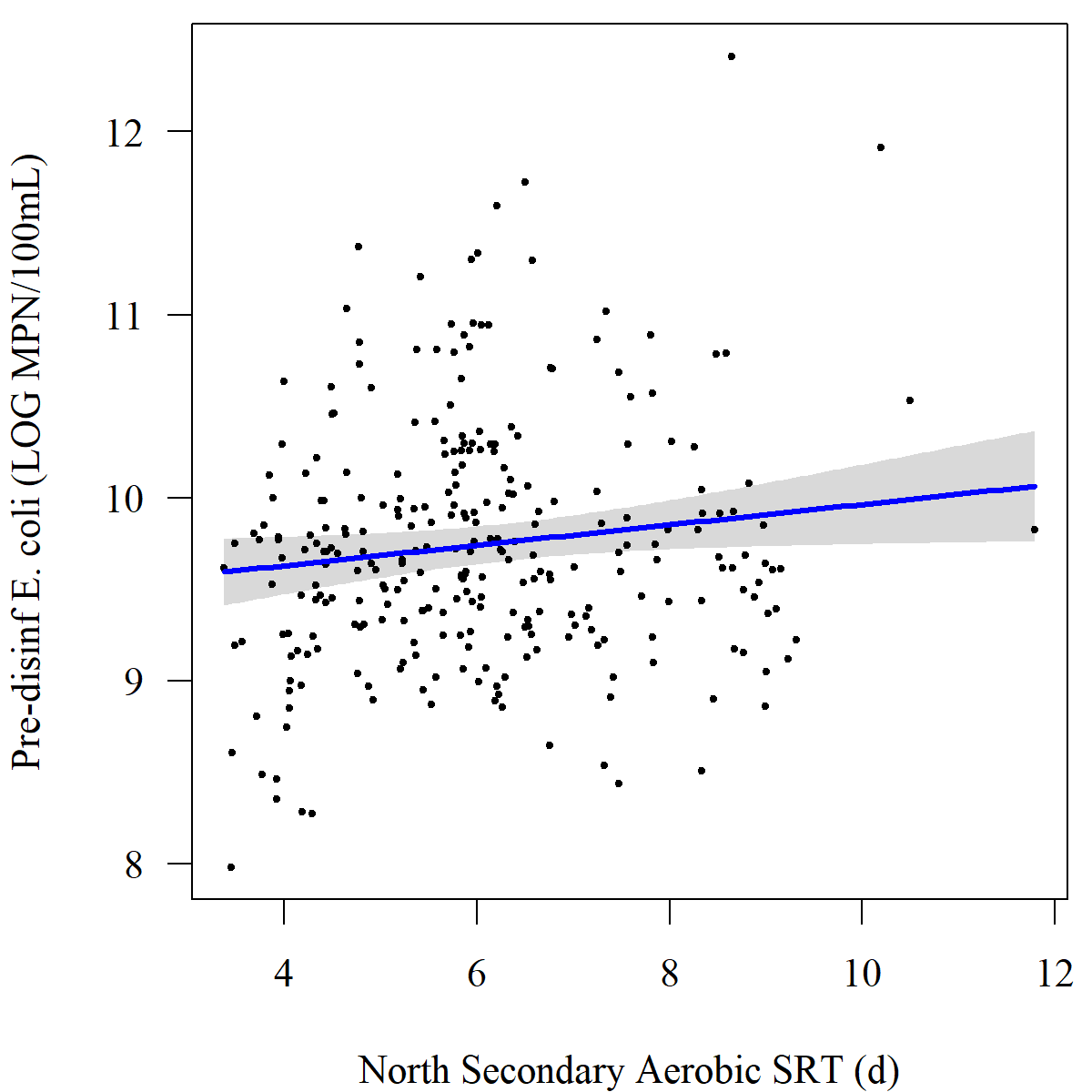
packageLoad("grid")

## Loading required package: grid

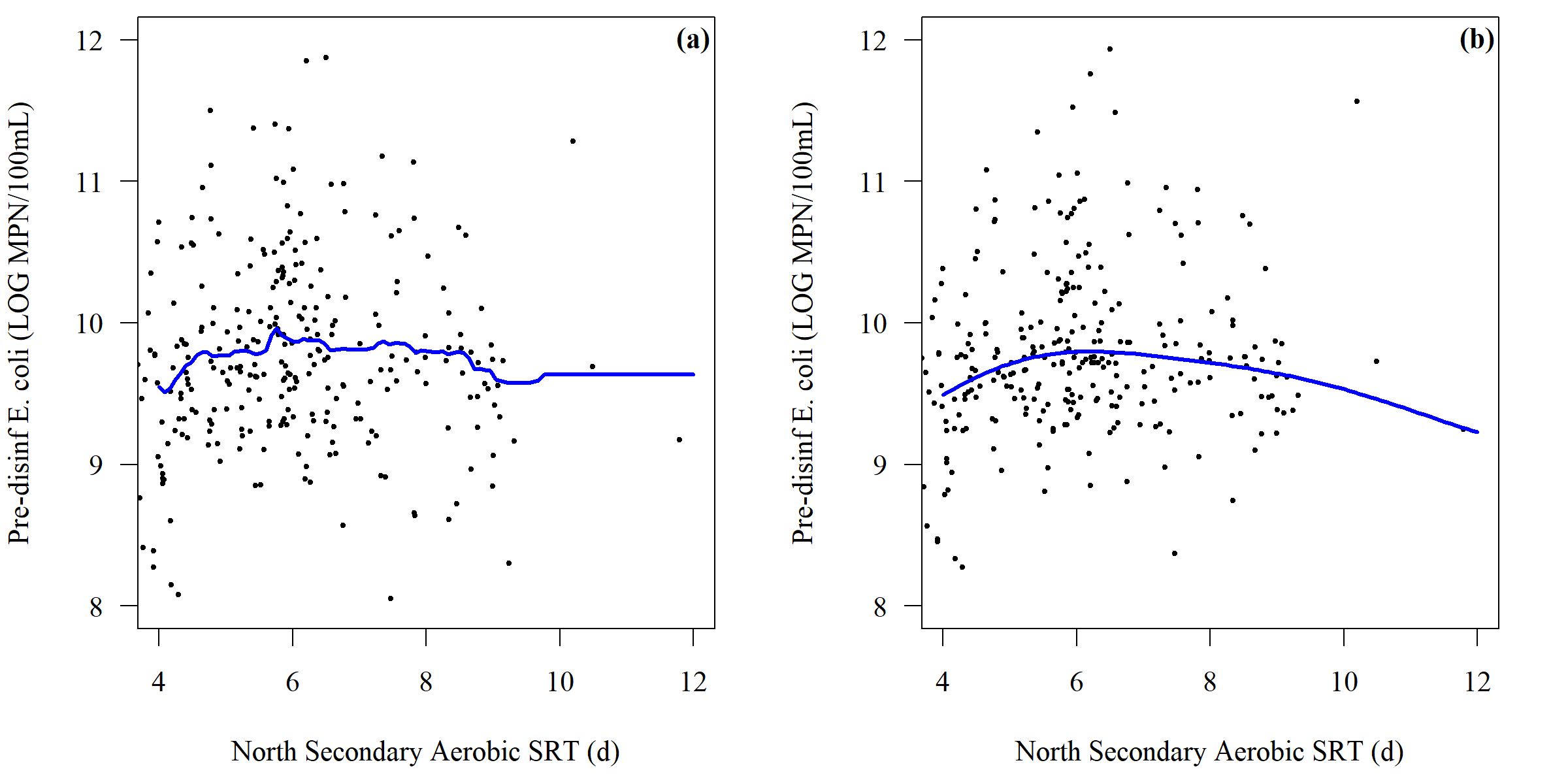
grid.raster(readTIFF("MWRD\_Ecoli\_nseconline\_lm.tiff"))

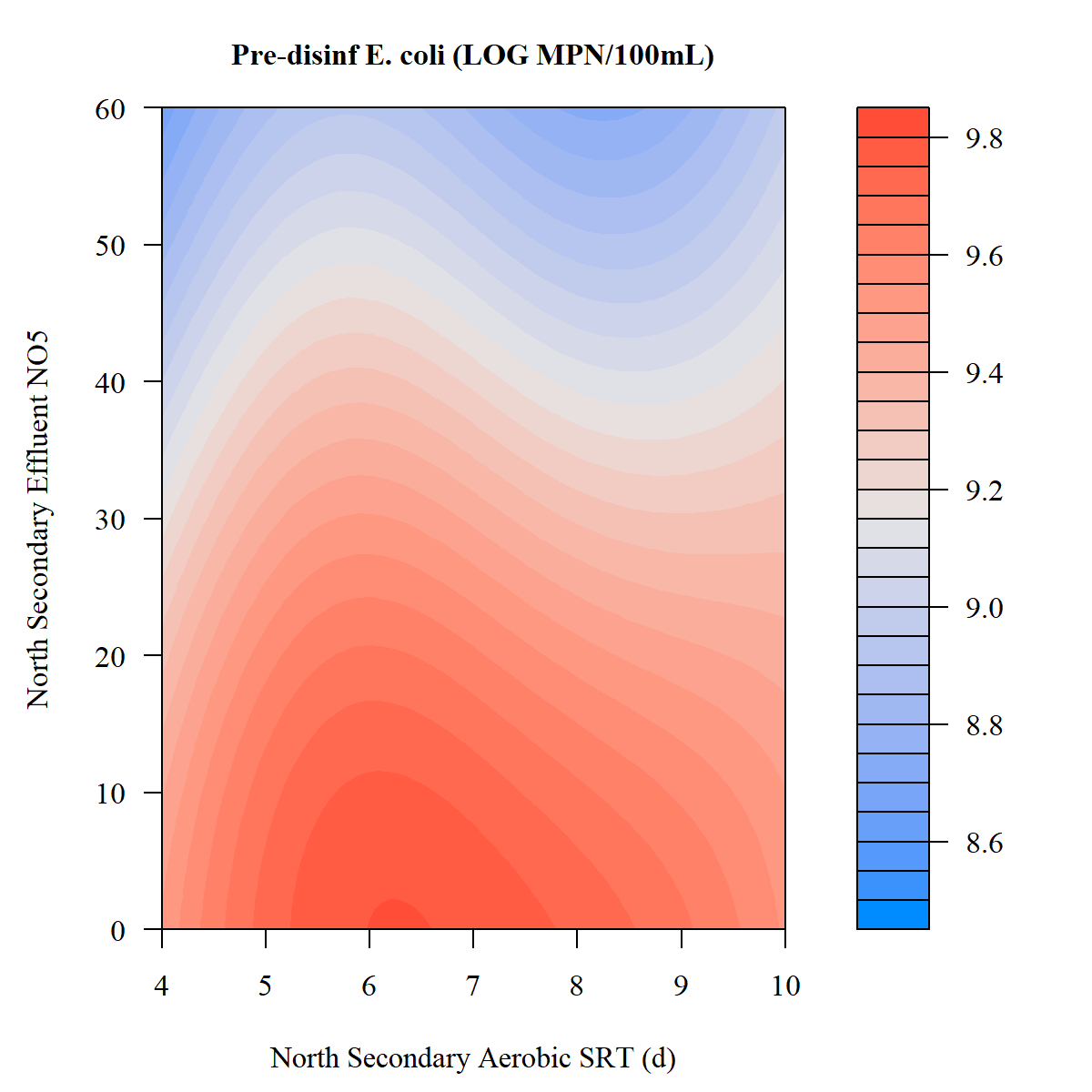


**Figure .** Multiple linear regression model fit for pre-disinfection *E. coli*. Black circles represent actual observations (x-axis) plotted against the prediction (y-axis). Blue line represents perfect model fit.

 **Figure .** Multiple linear regression model predicting pre-disinfection *E. coli*.

## Non-Regression Model

 **Figure .** Comparison of (a) random forest model and (b) support vector machine model for predicting *E. coli* into the PAA disinfection basin

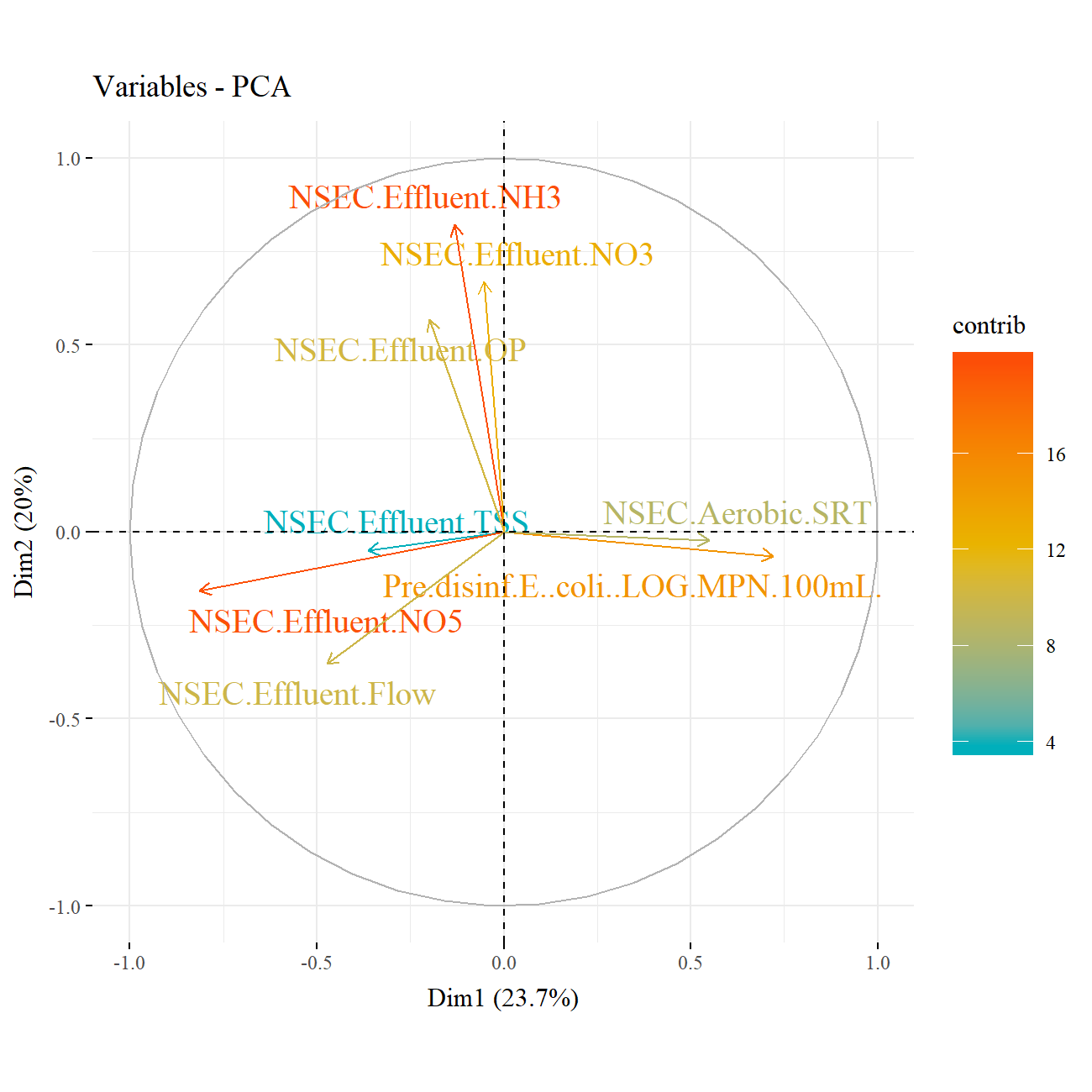
 **Figure .** Support vector machine model for predicting *E. coli* into the PAA disinfection basin

## Generalized Additive Models

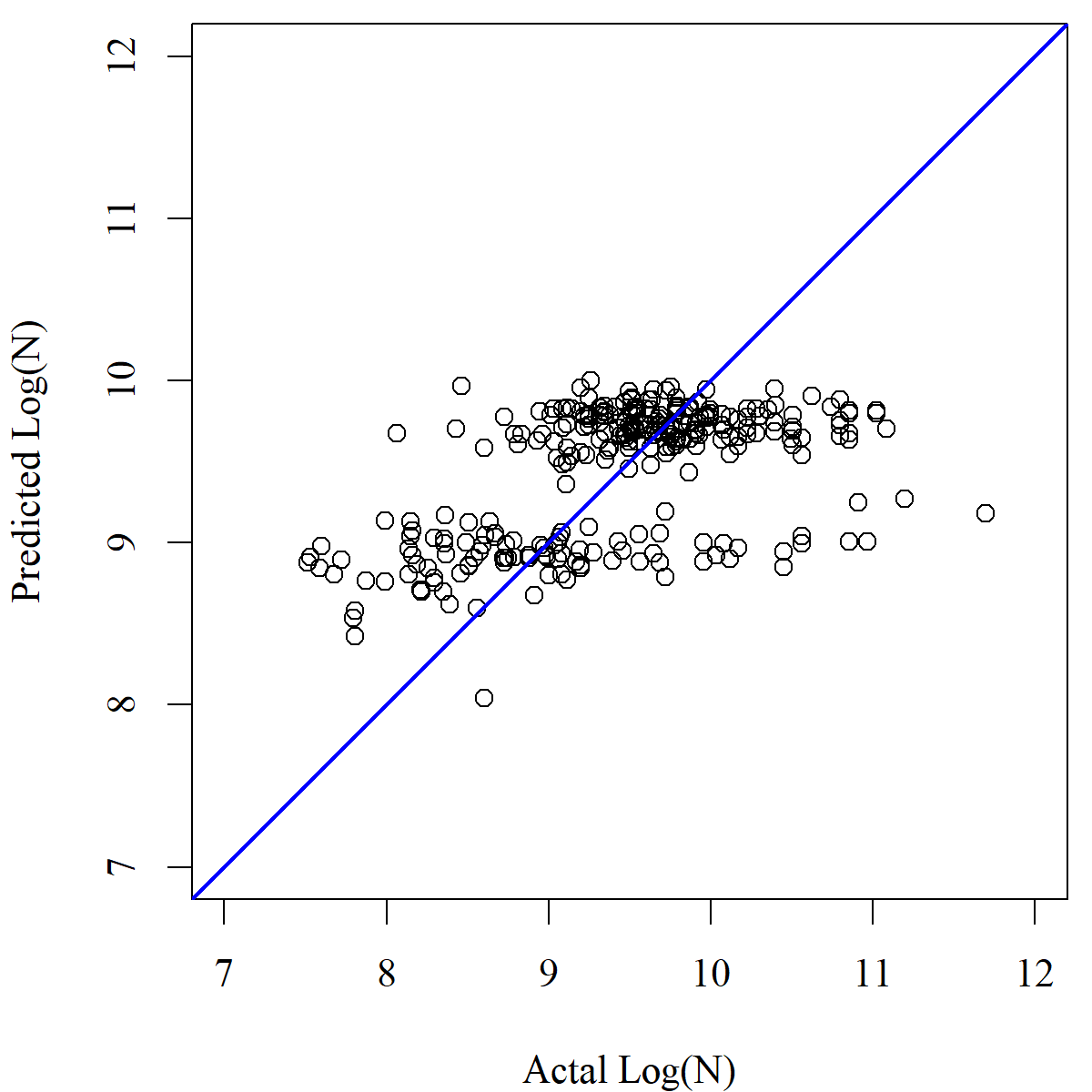
## Predictive model comparison

|  |  |
| --- | --- |
| Model | R.squared |
| Multiple Lienar Regression | 0.28 |
| Random Forest | 0.27 |
| Generalized Addative Model | 0.27 |

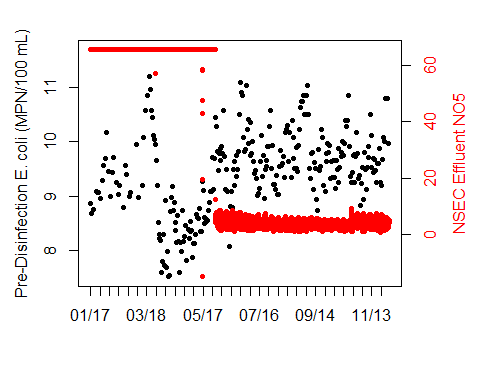
## Principal Component Analysis

 **Figure .** PCA variable loading for effluent of north secondary

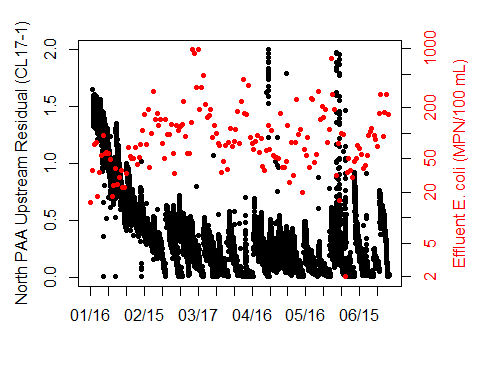
## Partial Least Squares

 **Figure .** PLS model fit for pre-disinfection *E. coli*. Black circles represent actual observations (x-axis) plotted against the prediction (y-axis). Blue line represents perfect model fit.

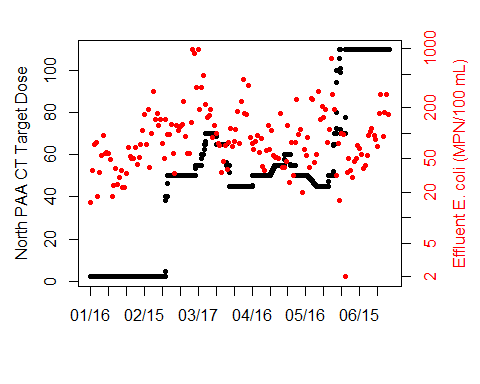
# Plot



data1 <- n.paa.online[,1][which( n.paa.online[,1] < 2.00001)]  
data2 <- n.paa.grab[,12]  
  
label1 <- colnames(data1)  
label2 <- colnames(data2)  
  
if (range(index(data1)[which(!is.na(data1))])[1] < range(index(data2)[which(!is.na(data2))])[1]) {  
 r1 <- range(index(data2)[which(!is.na(data2))])[1]  
} else {  
 r1 <- range(index(data1)[which(!is.na(data1))])[1]  
}  
if (range(index(data1)[which(!is.na(data1))])[2] > range(index(data2)[which(!is.na(data2))])[2]) {  
 r2 <- range(index(data2)[which(!is.na(data2))])[2]  
} else {  
 r2 <- range(index(data1)[which(!is.na(data1))])[2]  
}  
  
data2plot <- na.omit(data1)[paste0(r1,"/",r2)]  
data2plot <- merge(data2plot, data2[paste0(r1,"/",r2)])  
data2plot <- data.frame(data2plot)  
data2plot <- cbind(data2plot, as.numeric(difftime(as.POSIXct(rownames(data2plot)), as.POSIXct(rownames(data2plot)[1]),units = "days")))  
  
par(mar=c(5.1,4.1,2.1,4.1))  
plot(x = data2plot[,3], y = data2plot[,1], type = "p", pch = 20, col = "black", xaxt = "n", xlab = "", ylab = "", yaxt="n")  
axis(side = 2)  
mtext(side = 2, label1, line = 2.5)  
  
par(new = TRUE)  
plot(x = data2plot[,3], y = data2plot[,2], type = "p", pch = 20, col = "red", xaxt = "n", xlab = "", yaxt="n", ylab = "", log = 'y')  
axis(side = 4, col.axis = "red")  
mtext(side = 4, label2, line = 2.5, col = "red")  
# x-axis  
axis.ticks <- seq(0,round(data2plot[nrow(data2plot),3]), by = 10)  
axis.labels <- sapply(axis.ticks, function(x) which(x > data2plot[,3]))  
axis.labels <- sapply(axis.labels, function(x) x[length(x)])  
axis.labels[[1]] <- 1  
axis.labels <- as.numeric(unlist(axis.labels))  
axis(side = 1, at = axis.ticks, labels = format(as.POSIXct(rownames(data2plot)[axis.labels]), "%m/%d"))



data1 <- n.paa.online[,7]  
data2 <- n.paa.grab[,12]  
  
label1 <- colnames(data1)  
label2 <- colnames(data2)  
  
if (range(index(data1)[which(!is.na(data1))])[1] < range(index(data2)[which(!is.na(data2))])[1]) {  
 r1 <- range(index(data2)[which(!is.na(data2))])[1]  
} else {  
 r1 <- range(index(data1)[which(!is.na(data1))])[1]  
}  
if (range(index(data1)[which(!is.na(data1))])[2] > range(index(data2)[which(!is.na(data2))])[2]) {  
 r2 <- range(index(data2)[which(!is.na(data2))])[2]  
} else {  
 r2 <- range(index(data1)[which(!is.na(data1))])[2]  
}  
  
data2plot <- na.omit(data1)[paste0(r1,"/",r2)]  
data2plot <- merge(data2plot, data2[paste0(r1,"/",r2)])  
data2plot <- data.frame(data2plot)  
data2plot <- cbind(data2plot, as.numeric(difftime(as.POSIXct(rownames(data2plot)), as.POSIXct(rownames(data2plot)[1]),units = "days")))  
  
par(mar=c(5.1,4.1,2.1,4.1))  
plot(x = data2plot[,3], y = data2plot[,1], type = "p", pch = 20, col = "black", xaxt = "n", xlab = "", ylab = "", yaxt="n")  
axis(side = 2)  
mtext(side = 2, label1, line = 2.5)  
  
par(new = TRUE)  
plot(x = data2plot[,3], y = data2plot[,2], type = "p", pch = 20, col = "red", xaxt = "n", xlab = "", yaxt="n", ylab = "", log = 'y')  
axis(side = 4, col.axis = "red")  
mtext(side = 4, label2, line = 2.5, col = "red")  
# x-axis  
axis.ticks <- seq(0,round(data2plot[nrow(data2plot),3]), by = 10)  
axis.labels <- sapply(axis.ticks, function(x) which(x > data2plot[,3]))  
axis.labels <- sapply(axis.labels, function(x) x[length(x)])  
axis.labels[[1]] <- 1  
axis.labels <- as.numeric(unlist(axis.labels))  
axis(side = 1, at = axis.ticks, labels = format(as.POSIXct(rownames(data2plot)[axis.labels]), "%m/%d"))

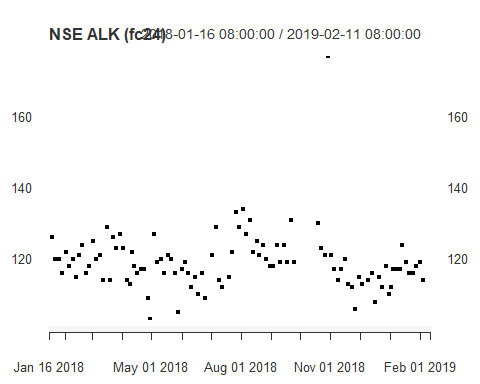


## North secondary effluent lab data  
nsec.eff.lab.fc24 <- as.data.frame(read\_excel("North Secondary and Disinfection Process Data\_20190215.xlsx",   
 sheet = "NSEC Eff Lab Data (FC24)", col\_names = FALSE,   
 col\_types = c("date", "numeric", "skip",   
 "numeric", "numeric", "numeric",   
 "numeric", "numeric", "numeric",   
 "numeric", "numeric", "numeric",   
 "numeric"), skip = 3))

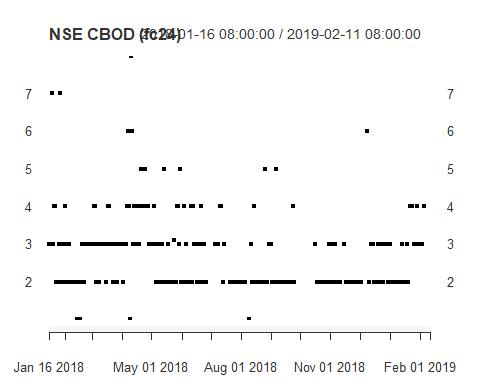
## New names:  
## \* `` -> ...1  
## \* `` -> ...2  
## \* `` -> ...3  
## \* `` -> ...4  
## \* `` -> ...5  
## \* ... and 7 more problems

nsec.eff.lab.fc24 <- xts(nsec.eff.lab.fc24[,-1], order.by = nsec.eff.lab.fc24[,1])  
colnames(nsec.eff.lab.fc24) <- as.vector(sapply(c("ALK","CBOD","COD","NH3-N","NO5-N","OP","TP","TIN","TKN","TN","TSS"), function(x) paste("NSE", x, "(fc24)")))  
sapply(nsec.eff.lab.fc24, function(x) plot.xts(x, type = "p", pch = 20, main = colnames(x), grid.col = NA))

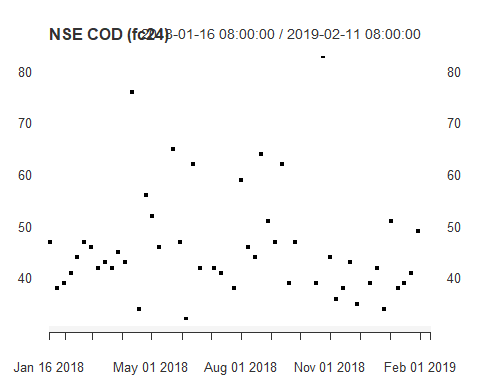
## $`NSE ALK (fc24)`



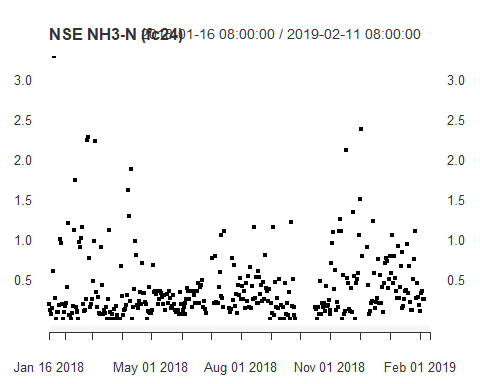
##   
## $`NSE CBOD (fc24)`



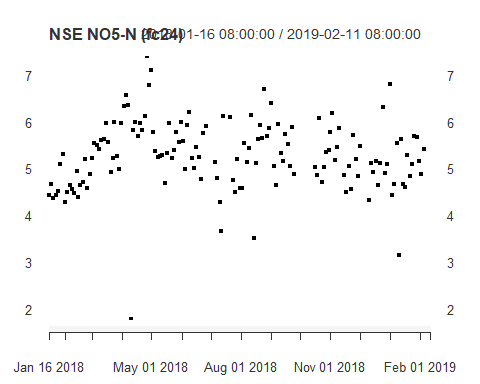
##   
## $`NSE COD (fc24)`



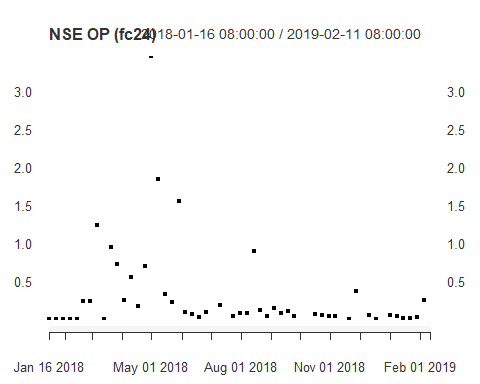
##   
## $`NSE NH3-N (fc24)`



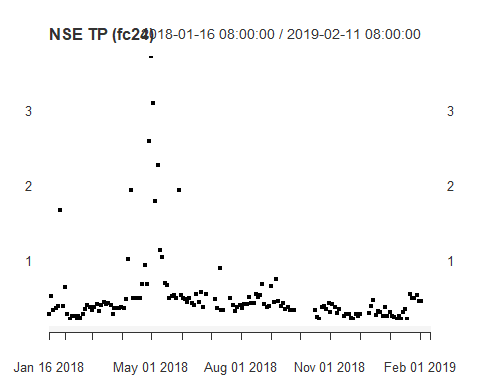
##   
## $`NSE NO5-N (fc24)`



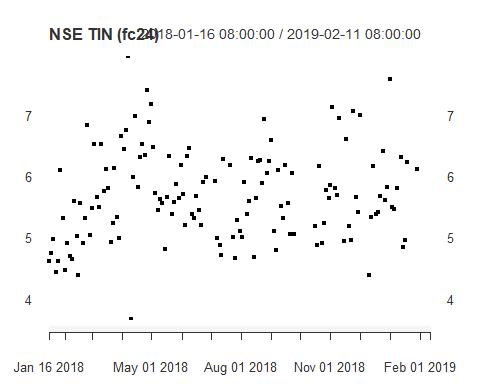
##   
## $`NSE OP (fc24)`



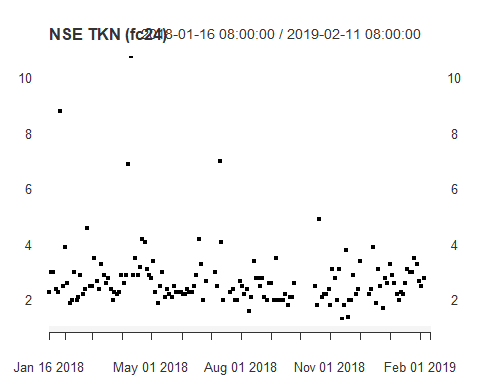
##   
## $`NSE TP (fc24)`



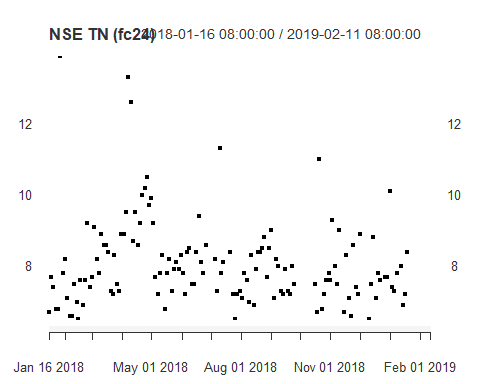
##   
## $`NSE TIN (fc24)`



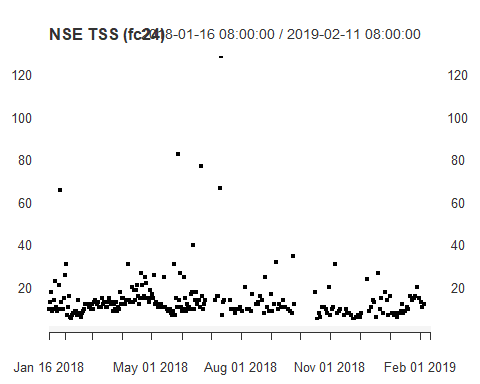
##   
## $`NSE TKN (fc24)`



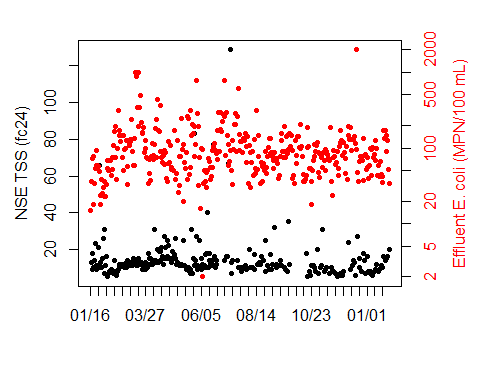
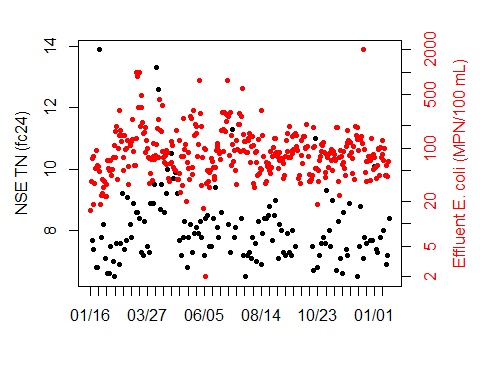
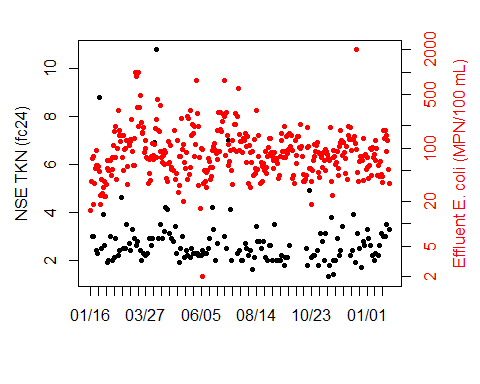
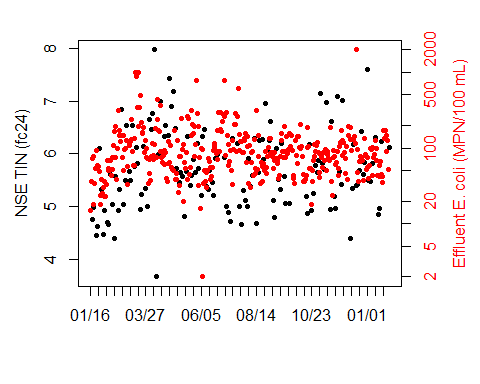
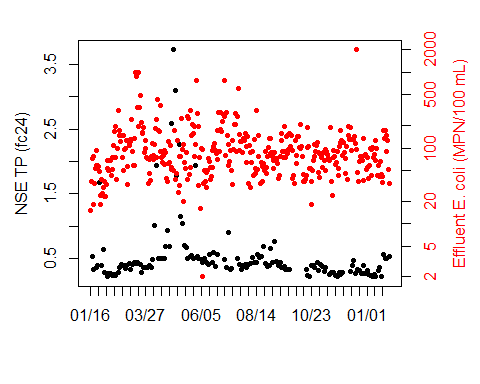
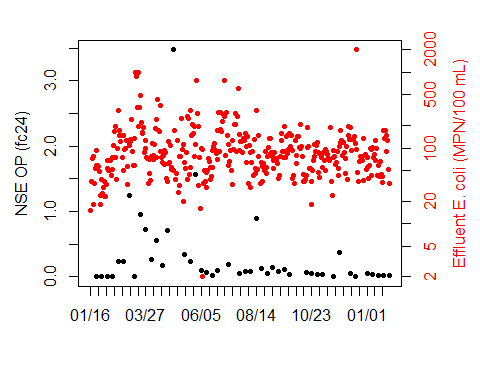
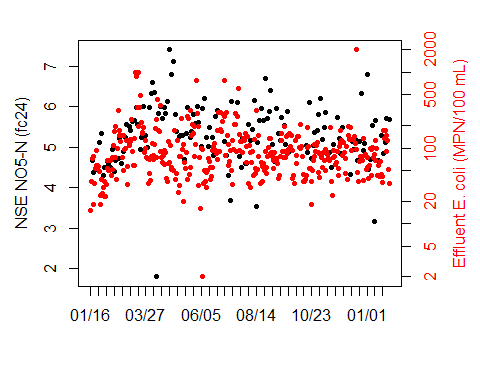
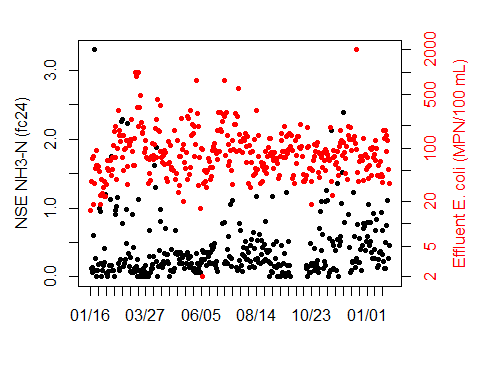
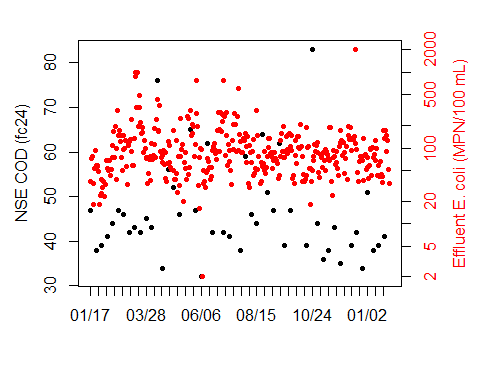
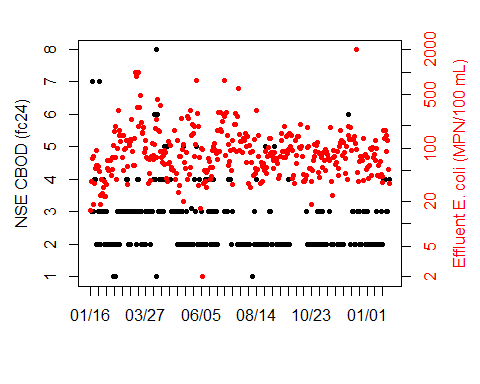
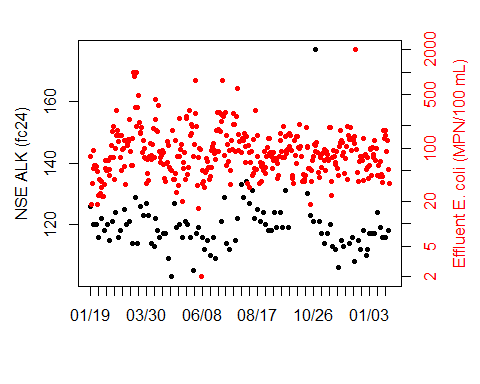
##   
## $`NSE TN (fc24)`



##   
## $`NSE TSS (fc24)`

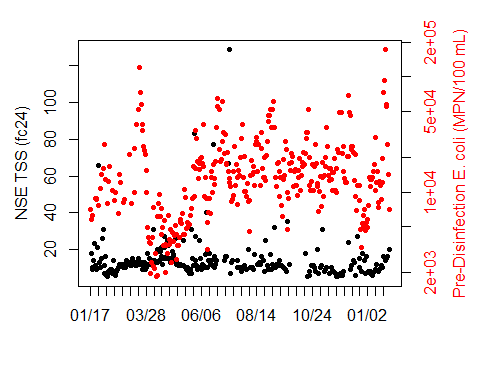
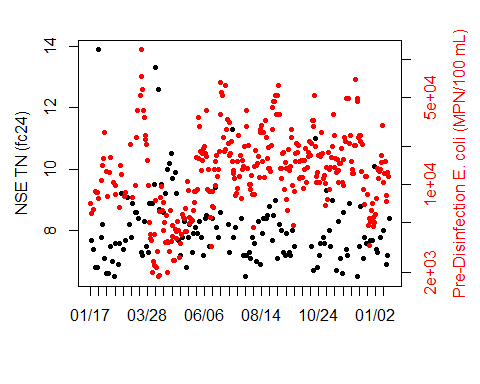
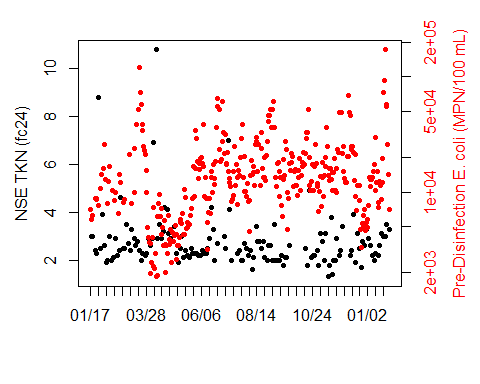
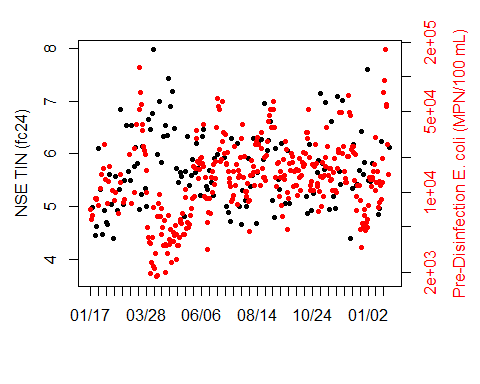
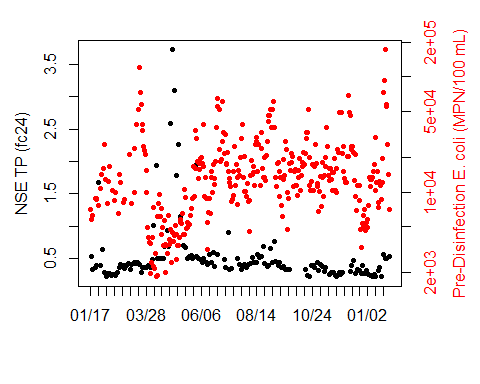
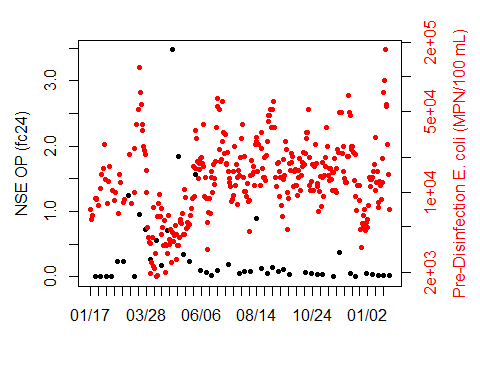
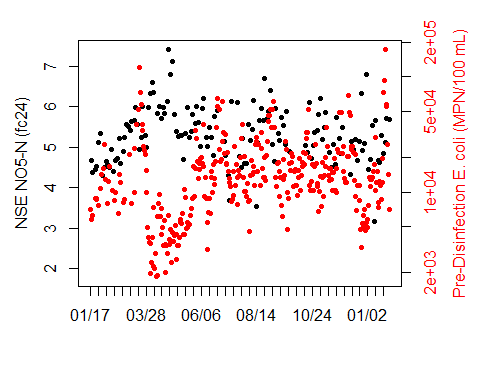
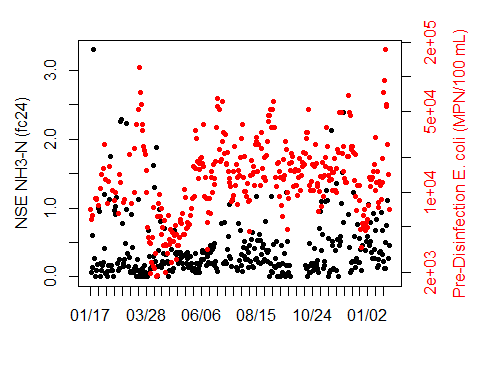
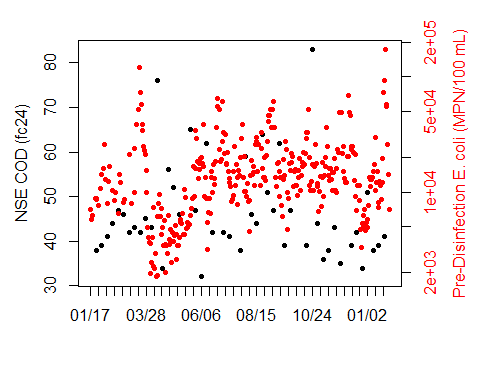
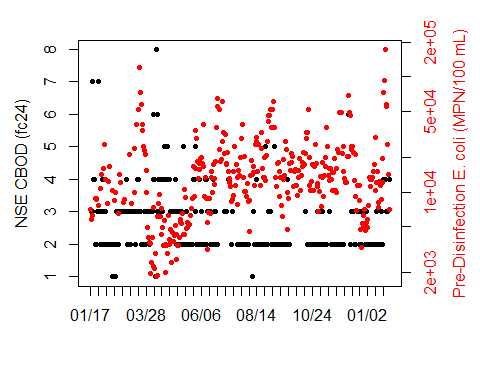
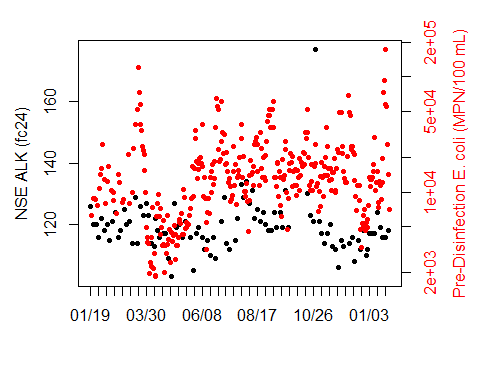


sapply(nsec.eff.lab.fc24, function(data1) {  
 # data1 <- n.paa.online[,7]  
 data2 <- n.paa.grab[,12]  
   
 label1 <- colnames(data1)  
 label2 <- colnames(data2)  
   
 if (range(index(data1)[which(!is.na(data1))])[1] < range(index(data2)[which(!is.na(data2))])[1]) {  
 r1 <- range(index(data2)[which(!is.na(data2))])[1]  
 } else {  
 r1 <- range(index(data1)[which(!is.na(data1))])[1]  
 }  
 if (range(index(data1)[which(!is.na(data1))])[2] > range(index(data2)[which(!is.na(data2))])[2]) {  
 r2 <- range(index(data2)[which(!is.na(data2))])[2]  
 } else {  
 r2 <- range(index(data1)[which(!is.na(data1))])[2]  
 }  
   
 data2plot <- na.omit(data1)[paste0(r1,"/",r2)]  
 data2plot <- merge(data2plot, data2[paste0(r1,"/",r2)])  
 data2plot <- data.frame(data2plot)  
 data2plot <- cbind(data2plot, as.numeric(difftime(as.POSIXct(rownames(data2plot)), as.POSIXct(rownames(data2plot)[1]),units = "days")))  
   
 par(mar=c(5.1,4.1,2.1,4.1))  
 plot(x = data2plot[,3], y = data2plot[,1], type = "p", pch = 20, col = "black", xaxt = "n", xlab = "", ylab = "", yaxt="n")  
 axis(side = 2)  
 mtext(side = 2, label1, line = 2.5)  
   
 par(new = TRUE)  
 plot(x = data2plot[,3], y = data2plot[,2], type = "p", pch = 20, col = "red", xaxt = "n", xlab = "", yaxt="n", ylab = "", log = 'y')  
 axis(side = 4, col.axis = "red")  
 mtext(side = 4, label2, line = 2.5, col = "red")  
 # x-axis  
 axis.ticks <- seq(0,round(data2plot[nrow(data2plot),3]), by = 10)  
 axis.labels <- sapply(axis.ticks, function(x) which(x > data2plot[,3]))  
 axis.labels <- sapply(axis.labels, function(x) x[length(x)])  
 axis.labels[[1]] <- 1  
 axis.labels <- as.numeric(unlist(axis.labels))  
 axis(side = 1, at = axis.ticks, labels = format(as.POSIXct(rownames(data2plot)[axis.labels]), "%m/%d"))  
})



## $`NSE ALK (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370  
##   
## $`NSE CBOD (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370  
##   
## $`NSE COD (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370  
##   
## $`NSE NH3-N (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370  
##   
## $`NSE NO5-N (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370  
##   
## $`NSE OP (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370  
##   
## $`NSE TP (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370  
##   
## $`NSE TIN (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370  
##   
## $`NSE TKN (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370  
##   
## $`NSE TN (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360  
##   
## $`NSE TSS (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370

sapply(nsec.eff.lab.fc24, function(data1) {  
 # data1 <- n.paa.online[,7]  
 data2 <- n.paa.grab[,6]  
   
 label1 <- colnames(data1)  
 label2 <- colnames(data2)  
   
 if (range(index(data1)[which(!is.na(data1))])[1] < range(index(data2)[which(!is.na(data2))])[1]) {  
 r1 <- range(index(data2)[which(!is.na(data2))])[1]  
 } else {  
 r1 <- range(index(data1)[which(!is.na(data1))])[1]  
 }  
 if (range(index(data1)[which(!is.na(data1))])[2] > range(index(data2)[which(!is.na(data2))])[2]) {  
 r2 <- range(index(data2)[which(!is.na(data2))])[2]  
 } else {  
 r2 <- range(index(data1)[which(!is.na(data1))])[2]  
 }  
   
 data2plot <- na.omit(data1)[paste0(r1,"/",r2)]  
 data2plot <- merge(data2plot, data2[paste0(r1,"/",r2)])  
 data2plot <- data.frame(data2plot)  
 data2plot <- cbind(data2plot, as.numeric(difftime(as.POSIXct(rownames(data2plot)), as.POSIXct(rownames(data2plot)[1]),units = "days")))  
   
 par(mar=c(5.1,4.1,2.1,4.1))  
 plot(x = data2plot[,3], y = data2plot[,1], type = "p", pch = 20, col = "black", xaxt = "n", xlab = "", ylab = "", yaxt="n")  
 axis(side = 2)  
 mtext(side = 2, label1, line = 2.5)  
   
 par(new = TRUE)  
 plot(x = data2plot[,3], y = data2plot[,2], type = "p", pch = 20, col = "red", xaxt = "n", xlab = "", yaxt="n", ylab = "", log = 'y')  
 axis(side = 4, col.axis = "red")  
 mtext(side = 4, label2, line = 2.5, col = "red")  
 # x-axis  
 axis.ticks <- seq(0,round(data2plot[nrow(data2plot),3]), by = 10)  
 axis.labels <- sapply(axis.ticks, function(x) which(x > data2plot[,3]))  
 axis.labels <- sapply(axis.labels, function(x) x[length(x)])  
 axis.labels[[1]] <- 1  
 axis.labels <- as.numeric(unlist(axis.labels))  
 axis(side = 1, at = axis.ticks, labels = format(as.POSIXct(rownames(data2plot)[axis.labels]), "%m/%d"))  
})

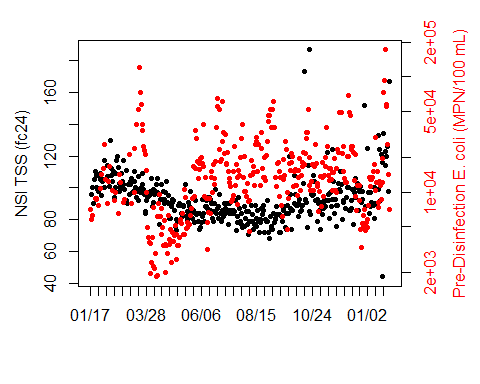
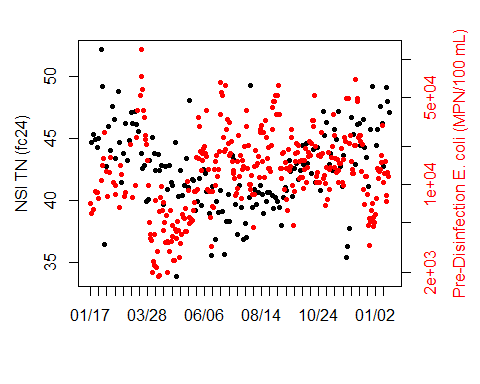
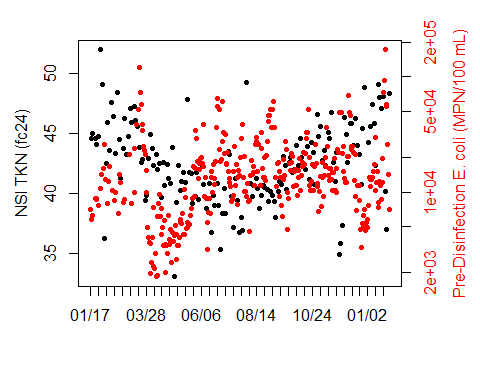
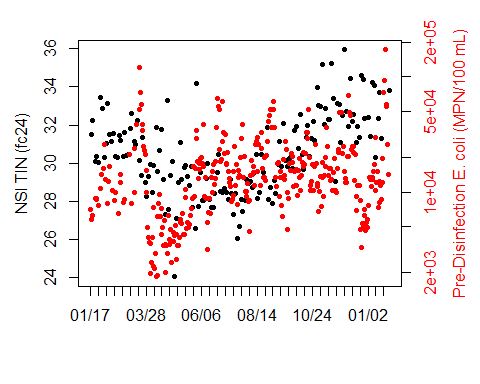
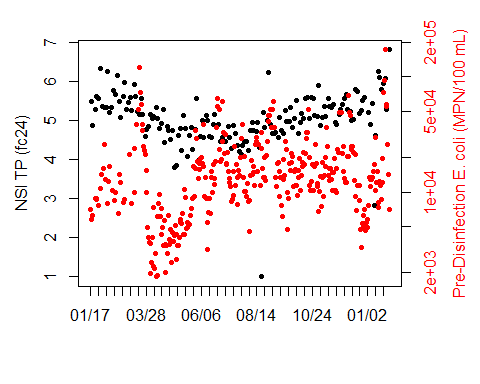
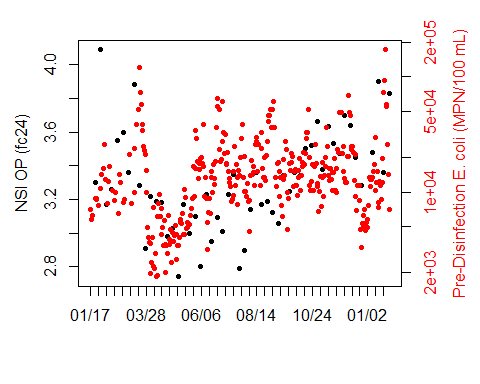
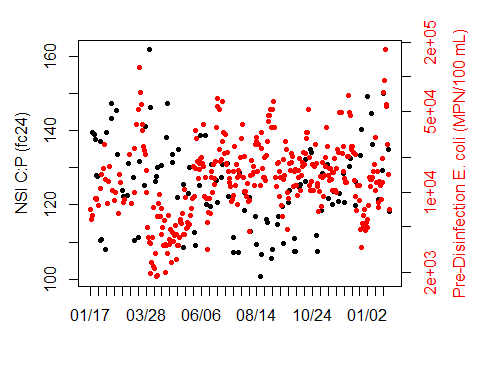
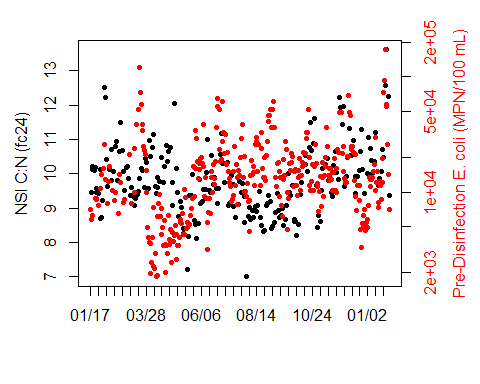
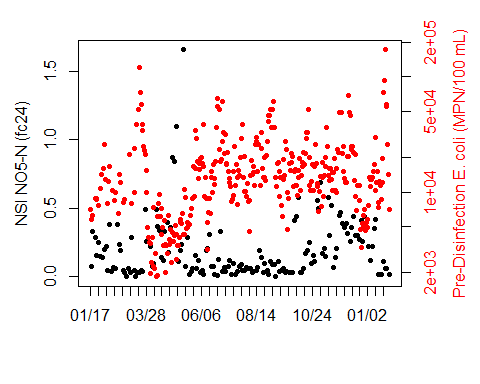
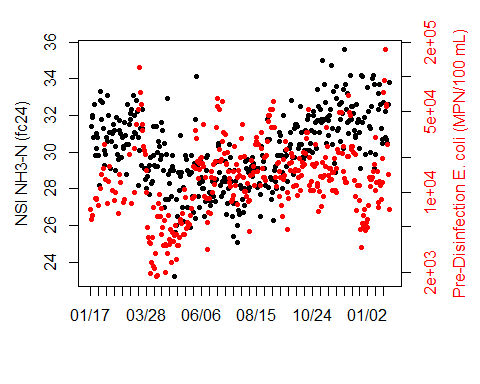
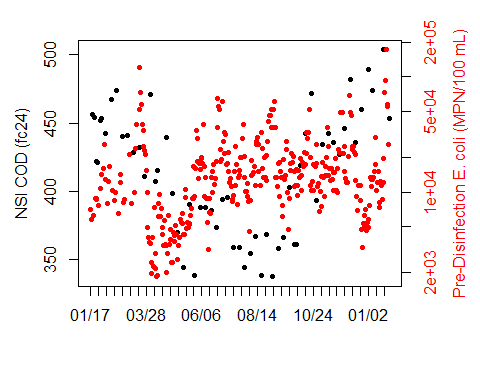
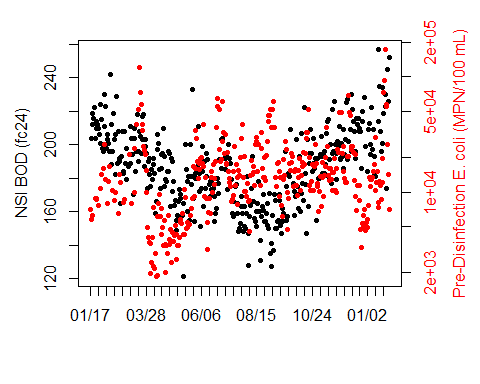
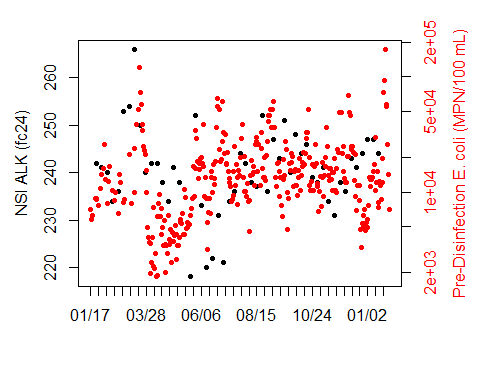


## $`NSE ALK (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370  
##   
## $`NSE CBOD (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370  
##   
## $`NSE COD (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370  
##   
## $`NSE NH3-N (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370  
##   
## $`NSE NO5-N (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370  
##   
## $`NSE OP (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370  
##   
## $`NSE TP (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370  
##   
## $`NSE TIN (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370  
##   
## $`NSE TKN (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370  
##   
## $`NSE TN (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360  
##   
## $`NSE TSS (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370

## North secondary influent lab  
nsec.inf.lab.fc24 <- as.data.frame(read\_excel("North Secondary and Disinfection Process Data\_20190215.xlsx",   
 sheet = "NSEC Inf Lab Data (FC24)", col\_names = FALSE,   
 col\_types = c("date", "numeric", "numeric",   
 "numeric", "numeric", "numeric",   
 "numeric", "numeric", "numeric",   
 "numeric", "numeric", "numeric",   
 "numeric", "numeric"), skip = 3))

## New names:  
## \* `` -> ...1  
## \* `` -> ...2  
## \* `` -> ...3  
## \* `` -> ...4  
## \* `` -> ...5  
## \* ... and 9 more problems

nsec.inf.lab.fc24 <- xts(nsec.inf.lab.fc24[,-1], order.by = nsec.inf.lab.fc24[,1])  
colnames(nsec.inf.lab.fc24) <- as.vector(sapply(c("NSI ALK","NSI BOD", "NSI COD", "NSI NH3-N", "NSI NO5-N", "NSI C:N", "NSI C:P", "NSI OP", "NSI TP", "NSI TIN", "NSI TKN", "NSI TN", "NSI TSS"), function(x) paste(x, "(fc24)")))  
  
sapply(nsec.inf.lab.fc24, function(data1) {  
 # data1 <- n.paa.online[,7]  
 data2 <- n.paa.grab[,6]  
   
 label1 <- colnames(data1)  
 label2 <- colnames(data2)  
   
 if (range(index(data1)[which(!is.na(data1))])[1] < range(index(data2)[which(!is.na(data2))])[1]) {  
 r1 <- range(index(data2)[which(!is.na(data2))])[1]  
 } else {  
 r1 <- range(index(data1)[which(!is.na(data1))])[1]  
 }  
 if (range(index(data1)[which(!is.na(data1))])[2] > range(index(data2)[which(!is.na(data2))])[2]) {  
 r2 <- range(index(data2)[which(!is.na(data2))])[2]  
 } else {  
 r2 <- range(index(data1)[which(!is.na(data1))])[2]  
 }  
   
 data2plot <- na.omit(data1)[paste0(r1,"/",r2)]  
 data2plot <- merge(data2plot, data2[paste0(r1,"/",r2)])  
 data2plot <- data.frame(data2plot)  
 data2plot <- cbind(data2plot, as.numeric(difftime(as.POSIXct(rownames(data2plot)), as.POSIXct(rownames(data2plot)[1]),units = "days")))  
   
 par(mar=c(5.1,4.1,2.1,4.1))  
 plot(x = data2plot[,3], y = data2plot[,1], type = "p", pch = 20, col = "black", xaxt = "n", xlab = "", ylab = "", yaxt="n")  
 axis(side = 2)  
 mtext(side = 2, label1, line = 2.5)  
   
 par(new = TRUE)  
 plot(x = data2plot[,3], y = data2plot[,2], type = "p", pch = 20, col = "red", xaxt = "n", xlab = "", yaxt="n", ylab = "", log = 'y')  
 axis(side = 4, col.axis = "red")  
 mtext(side = 4, label2, line = 2.5, col = "red")  
 # x-axis  
 axis.ticks <- seq(0,round(data2plot[nrow(data2plot),3]), by = 10)  
 axis.labels <- sapply(axis.ticks, function(x) which(x > data2plot[,3]))  
 axis.labels <- sapply(axis.labels, function(x) x[length(x)])  
 axis.labels[[1]] <- 1  
 axis.labels <- as.numeric(unlist(axis.labels))  
 axis(side = 1, at = axis.ticks, labels = format(as.POSIXct(rownames(data2plot)[axis.labels]), "%m/%d"))  
})

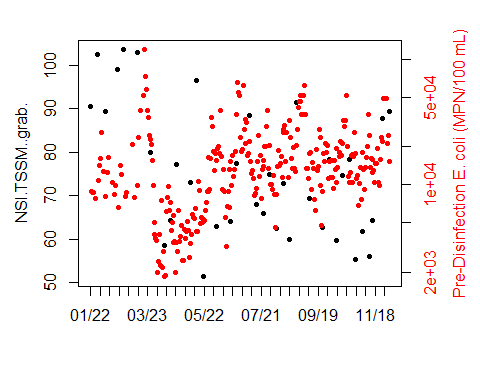
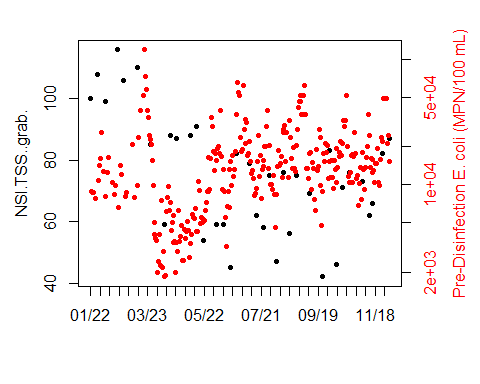
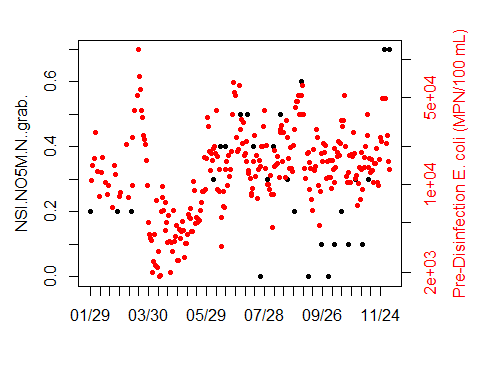
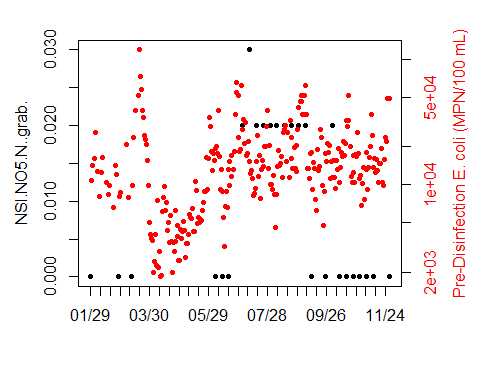
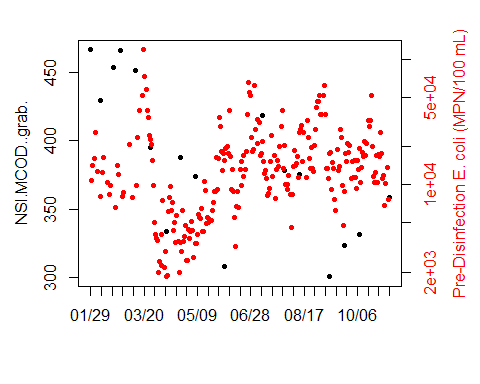
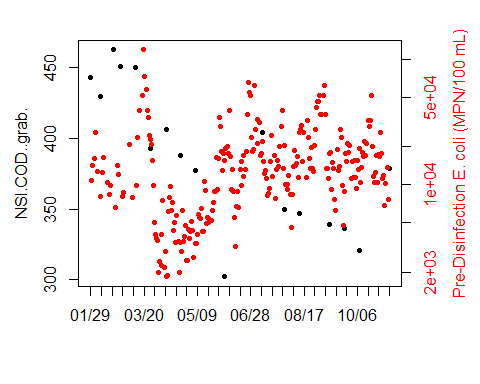


## $`NSI ALK (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370  
##   
## $`NSI BOD (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370  
##   
## $`NSI COD (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370  
##   
## $`NSI NH3-N (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370  
##   
## $`NSI NO5-N (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370  
##   
## $`NSI C:N (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370  
##   
## $`NSI C:P (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370  
##   
## $`NSI OP (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370  
##   
## $`NSI TP (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370  
##   
## $`NSI TIN (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370  
##   
## $`NSI TKN (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370  
##   
## $`NSI TN (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360  
##   
## $`NSI TSS (fc24)`  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330  
## [35] 340 350 360 370

nsec.inf.lab.grab <- as.data.frame(read\_excel("North Secondary and Disinfection Process Data\_20190215.xlsx",  
 sheet = "NSEC Inf Lab Data (Grab)", col\_names = FALSE,  
 col\_types = c("date", "numeric", "skip",  
 "skip", "skip", "skip", "skip", "date",  
 "numeric", "skip", "skip", "skip",  
 "skip", "skip", "date", "numeric",  
 "skip", "skip", "skip",  
 "skip", "skip", "date", "numeric",  
 "skip", "skip", "skip",  
 "skip", "skip", "date",  
 "numeric", "skip", "skip",  
 "skip", "skip", "skip",  
 "date", "numeric"), skip = 4))

## New names:  
## \* `` -> ...1  
## \* `` -> ...2  
## \* `` -> ...3  
## \* `` -> ...4  
## \* `` -> ...5  
## \* ... and 7 more problems

colnames(nsec.inf.lab.grab) <- c("Time", "NSI COD (grab)",  
 "Time", "NSI MCOD (grab)",  
 "Time", "NSI NO5-N (grab)",  
 "Time", "NSI NO5M-N (grab)",  
 "Time", "NSI TSS (grab)",  
 "Time", "NSI TSSM (grab)")  
for (i in c(1,3,5,7,9,11)) {  
 blah <- nsec.inf.lab.grab[,c(i,(i+1))]  
 blah <- na.omit(blah)  
 blah <- xts(blah[,2], order.by = blah[,1])  
 colnames(blah) <- colnames(nsec.inf.lab.grab)[i+1]  
 if (i == 1) {  
 nsec.inf.lab.grab.merged <- blah  
 } else {  
 nsec.inf.lab.grab.merged <- merge(nsec.inf.lab.grab.merged, blah)  
 }  
}  
  
  
sapply(nsec.inf.lab.grab.merged, function(data1) {  
 # data1 <- n.paa.online[,7]  
 data2 <- n.paa.grab[,6]  
   
 label1 <- colnames(data1)  
 label2 <- colnames(data2)  
   
 if (range(index(data1)[which(!is.na(data1))])[1] < range(index(data2)[which(!is.na(data2))])[1]) {  
 r1 <- range(index(data2)[which(!is.na(data2))])[1]  
 } else {  
 r1 <- range(index(data1)[which(!is.na(data1))])[1]  
 }  
 if (range(index(data1)[which(!is.na(data1))])[2] > range(index(data2)[which(!is.na(data2))])[2]) {  
 r2 <- range(index(data2)[which(!is.na(data2))])[2]  
 } else {  
 r2 <- range(index(data1)[which(!is.na(data1))])[2]  
 }  
   
 data2plot <- na.omit(data1)[paste0(r1,"/",r2)]  
 data2plot <- merge(data2plot, data2[paste0(r1,"/",r2)])  
 data2plot <- data.frame(data2plot)  
 data2plot <- cbind(data2plot, as.numeric(difftime(as.POSIXct(rownames(data2plot)), as.POSIXct(rownames(data2plot)[1]),units = "days")))  
   
 par(mar=c(5.1,4.1,2.1,4.1))  
 plot(x = data2plot[,3], y = data2plot[,1], type = "p", pch = 20, col = "black", xaxt = "n", xlab = "", ylab = "", yaxt="n")  
 axis(side = 2)  
 mtext(side = 2, label1, line = 2.5)  
   
 par(new = TRUE)  
 plot(x = data2plot[,3], y = data2plot[,2], type = "p", pch = 20, col = "red", xaxt = "n", xlab = "", yaxt="n", ylab = "", log = 'y')  
 axis(side = 4, col.axis = "red")  
 mtext(side = 4, label2, line = 2.5, col = "red")  
 # x-axis  
 axis.ticks <- seq(0,round(data2plot[nrow(data2plot),3]), by = 10)  
 axis.labels <- sapply(axis.ticks, function(x) which(x > data2plot[,3]))  
 axis.labels <- sapply(axis.labels, function(x) x[length(x)])  
 axis.labels[[1]] <- 1  
 axis.labels <- as.numeric(unlist(axis.labels))  
 axis(side = 1, at = axis.ticks, labels = format(as.POSIXct(rownames(data2plot)[axis.labels]), "%m/%d"))  
})



## $NSI.COD..grab.  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280  
##   
## $NSI.MCOD..grab.  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280  
##   
## $NSI.NO5.N..grab.  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300  
##   
## $NSI.NO5M.N..grab.  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300  
##   
## $NSI.TSS..grab.  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310  
##   
## $NSI.TSSM..grab.  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310

Time plots reveal few correlations between *E. coli* and process variables. TSSM in the north secondary influent and

## North Secondary Effluent Flow Composite

data1 <- nsec.eff.lab.fc24  
data2 <- n.paa.grab[,6]  
  
label1 <- colnames(data1)  
label2 <- colnames(data2)  
  
all.data <- merge(data2, data1)  
all.data.index <- which(!is.na(all.data[,1]))  
for(i in 1:(length(all.data.index)-1)) {  
 row.start <- all.data.index[i]  
 row.stop <- all.data.index[i+1]  
 data.locf <- na.locf(all.data[(row.start+1):row.stop,])  
 if (i == 1) {  
 new.data <- data.frame(data.locf[nrow(data.locf),])  
 }  
 if (i != 1) {  
 new.data <- rbind(new.data, data.frame(data.locf[nrow(data.locf),]))  
 }  
}  
  
apply(new.data, 2, function(x) length(which(!is.na(x)))/nrow(new.data))[order(-apply(new.data, 2, function(x) length(which(!is.na(x)))/nrow(new.data)))]

## Pre.Disinfection.E..coli..MPN.100.mL.   
## 1.0000000   
## NSE.NH3.N..fc24.   
## 0.7813411   
## NSE.CBOD..fc24.   
## 0.6180758   
## NSE.TSS..fc24.   
## 0.6180758   
## NSE.TKN..fc24.   
## 0.4227405   
## NSE.NO5.N..fc24.   
## 0.4169096   
## NSE.TP..fc24.   
## 0.4169096   
## NSE.TIN..fc24.   
## 0.4052478   
## NSE.TN..fc24.   
## 0.3994169   
## NSE.ALK..fc24.   
## 0.2915452   
## NSE.COD..fc24.   
## 0.1370262   
## NSE.OP..fc24.   
## 0.1370262

new.data <- new.data[,order(-apply(new.data, 2, function(x) length(which(!is.na(x)))/nrow(new.data)))[1:10]]  
new.data <- na.omit(new.data)  
  
  
# pls.fit <- plsr(new.data[,1] ~ new.data[,2] + new.data[,3] + new.data[,4], scale = TRUE, validation = "CV")  
# summary(pls.fit)  
# plot(pls.fit)  
# validationplot(pls.fit, val.type = "MSEP")  
#   
  
# pls.reg.fit <- plsreg1(predictors = new.data[,2:4], response = new.data[,1], comps = 3, crosval = TRUE)  
# plot(pls.reg.fit)  
# plot(new.data[,1], pls.reg.fit$y.pred, log = 'yx')  
  
  
mod\_gam\_nsec\_eff <- gam(new.data[,1] ~ new.data[,2] + new.data[,5] + new.data[,6] + new.data[,9])  
summary(mod\_gam\_nsec\_eff)

##   
## Family: gaussian   
## Link function: identity   
##   
## Formula:  
## new.data[, 1] ~ new.data[, 2] + new.data[, 5] + new.data[, 6] +   
## new.data[, 9]  
##   
## Parametric coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 6079 15372 0.395 0.6955   
## new.data[, 2] -11808 6908 -1.709 0.0985 .  
## new.data[, 5] -124870 69426 -1.799 0.0829 .  
## new.data[, 6] -122966 69128 -1.779 0.0861 .  
## new.data[, 9] 125164 69454 1.802 0.0823 .  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
##   
## R-sq.(adj) = 0.0833 Deviance explained = 19.8%  
## GCV = 1.5455e+08 Scale est. = 1.3114e+08 n = 33

## North Secondary Influent Grab

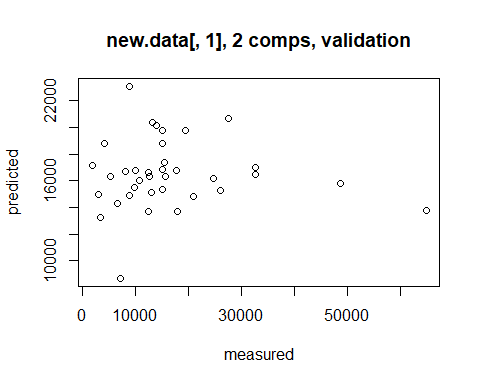
data1 <- nsec.inf.lab.grab.merged  
data2 <- n.paa.grab[,6]  
  
label1 <- colnames(data1)  
label2 <- colnames(data2)  
  
all.data <- merge(data2, data1)  
all.data.index <- which(!is.na(all.data[,1]))  
for(i in 1:(length(all.data.index)-1)) {  
 row.start <- all.data.index[i]  
 row.stop <- all.data.index[i+1]  
 data.locf <- na.locf(all.data[(row.start+1):row.stop,])  
 if (i == 1) {  
 new.data <- data.frame(data.locf[nrow(data.locf),])  
 }  
 if (i != 1) {  
 new.data <- rbind(new.data, data.frame(data.locf[nrow(data.locf),]))  
 }  
}  
  
apply(new.data, 2, function(x) length(which(!is.na(x)))/nrow(new.data))

## Pre.Disinfection.E..coli..MPN.100.mL.   
## 1.00000000   
## NSI.COD..grab.   
## 0.04956268   
## NSI.MCOD..grab.   
## 0.04956268   
## NSI.NO5.N..grab.   
## 0.07580175   
## NSI.NO5M.N..grab.   
## 0.07871720   
## NSI.TSS..grab.   
## 0.10495627   
## NSI.TSSM..grab.   
## 0.10787172

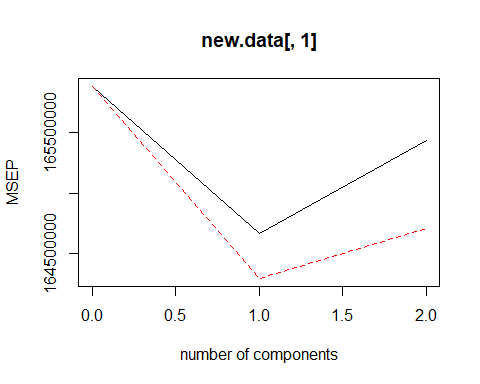
new.data <- new.data[,order(-apply(new.data, 2, function(x) length(which(!is.na(x)))/nrow(new.data)))[1:3]]  
new.data <- na.omit(new.data)  
  
library(pls)  
pls.fit <- plsr(new.data[,1] ~ new.data[,2] + new.data[,3], scale = TRUE, validation = "CV")  
summary(pls.fit)

## Data: X dimension: 36 2   
## Y dimension: 36 1  
## Fit method: kernelpls  
## Number of components considered: 2  
##   
## VALIDATION: RMSEP  
## Cross-validated using 10 random segments.  
## (Intercept) 1 comps 2 comps  
## CV 12880 12832 12862  
## adjCV 12880 12818 12834  
##   
## TRAINING: % variance explained  
## 1 comps 2 comps  
## X 91.433 100.000  
## new.data[, 1] 1.138 3.109

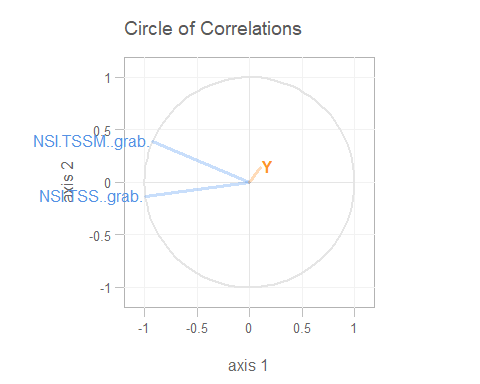
plot(pls.fit)



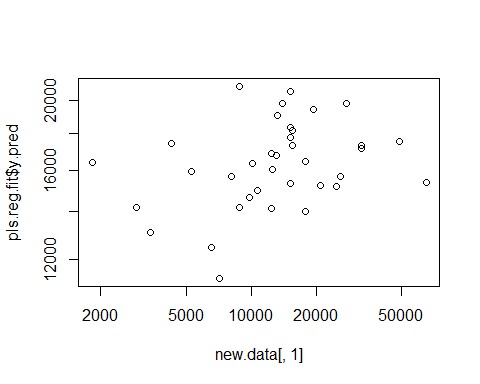
validationplot(pls.fit, val.type = "MSEP")



library(plsdepot)  
pls.reg.fit <- plsreg1(predictors = new.data[,2:3], response = new.data[,1], comps = 3, crosval = TRUE)  
plot(pls.reg.fit)



plot(new.data[,1], pls.reg.fit$y.pred, log = 'yx')



library(mgcv)  
mod\_gam\_nsec\_inf\_grab <- gam(new.data[,1] ~ new.data[,2] + new.data[,3])  
summary(mod\_gam\_nsec\_inf\_grab)

##   
## Family: gaussian   
## Link function: identity   
##   
## Formula:  
## new.data[, 1] ~ new.data[, 2] + new.data[, 3]  
##   
## Parametric coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 16904.9 11691.2 1.446 0.158  
## new.data[, 2] 210.5 289.4 0.727 0.472  
## new.data[, 3] -217.7 218.3 -0.998 0.326  
##   
##   
## R-sq.(adj) = -0.0276 Deviance explained = 3.11%  
## GCV = 1.808e+08 Scale est. = 1.6573e+08 n = 36

## North Secondary Influent Flow Composite

data1 <- nsec.inf.lab.fc24  
data2 <- n.paa.grab[,6]  
  
label1 <- colnames(data1)  
label2 <- colnames(data2)  
  
all.data <- merge(data2, data1)  
all.data.index <- which(!is.na(all.data[,1]))  
for(i in 1:(length(all.data.index)-1)) {  
 row.start <- all.data.index[i]  
 row.stop <- all.data.index[i+1]  
 data.locf <- na.locf(all.data[(row.start+1):row.stop,])  
 if (i == 1) {  
 new.data <- data.frame(data.locf[nrow(data.locf),])  
 }  
 if (i != 1) {  
 new.data <- rbind(new.data, data.frame(data.locf[nrow(data.locf),]))  
 }  
}  
  
apply(new.data, 2, function(x) length(which(!is.na(x)))/nrow(new.data))[order(-apply(new.data, 2, function(x) length(which(!is.na(x)))/nrow(new.data)))]

## Pre.Disinfection.E..coli..MPN.100.mL.   
## 1.0000000   
## NSI.NH3.N..fc24.   
## 0.8542274   
## NSI.TSS..fc24.   
## 0.8513120   
## NSI.BOD..fc24.   
## 0.8396501   
## NSI.C.N..fc24.   
## 0.5772595   
## NSI.NO5.N..fc24.   
## 0.4635569   
## NSI.TP..fc24.   
## 0.4635569   
## NSI.TKN..fc24.   
## 0.4606414   
## NSI.TIN..fc24.   
## 0.4548105   
## NSI.TN..fc24.   
## 0.4489796   
## NSI.C.P..fc24.   
## 0.2827988   
## NSI.COD..fc24.   
## 0.1690962   
## NSI.OP..fc24.   
## 0.1545190   
## NSI.ALK..fc24.   
## 0.1486880

new.data <- new.data[,order(-apply(new.data, 2, function(x) length(which(!is.na(x)))/nrow(new.data)))[1:10]]  
new.data <- na.omit(new.data)  
  
# library(pls)  
# pls.fit <- plsr(new.data[,1] ~ new.data[,2] + new.data[,3] + new.data[,4], scale = TRUE, validation = "CV")  
# summary(pls.fit)  
# plot(pls.fit)  
# validationplot(pls.fit, val.type = "MSEP")  
#   
# library(plsdepot)  
# pls.reg.fit <- plsreg1(predictors = new.data[,2:4], response = new.data[,1], comps = 3, crosval = TRUE)  
# plot(pls.reg.fit)  
# plot(new.data[,1], pls.reg.fit$y.pred, log = 'yx')  
  
library(mgcv)  
mod\_gam\_nsec\_inf\_fc24 <- gam(new.data[,1] ~ new.data[,2] + new.data[,3] + new.data[,4] + new.data[,5] + new.data[,6] + new.data[,7] + new.data[,8] + new.data[,9] + new.data[,10])  
summary(mod\_gam\_nsec\_inf\_fc24)

##   
## Family: gaussian   
## Link function: identity   
##   
## Formula:  
## new.data[, 1] ~ new.data[, 2] + new.data[, 3] + new.data[, 4] +   
## new.data[, 5] + new.data[, 6] + new.data[, 7] + new.data[,   
## 8] + new.data[, 9] + new.data[, 10]  
##   
## Parametric coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 34282.83 25246.20 1.358 0.1766   
## new.data[, 2] -1994.91 1649.42 -1.209 0.2285   
## new.data[, 3] -197.50 133.28 -1.482 0.1406   
## new.data[, 4] 77.02 78.66 0.979 0.3292   
## new.data[, 5] -447.07 1670.57 -0.268 0.7894   
## new.data[, 6] 12685.50 41597.12 0.305 0.7608   
## new.data[, 7] 2504.14 2493.36 1.004 0.3169   
## new.data[, 8] 27386.15 40106.01 0.683 0.4958   
## new.data[, 9] 3206.57 1608.69 1.993 0.0481 \*  
## new.data[, 10] -28717.58 40124.58 -0.716 0.4753   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
##   
## R-sq.(adj) = 0.0687 Deviance explained = 12.4%  
## GCV = 2.2572e+08 Scale est. = 2.1087e+08 n = 152

# GAM

# South process GAM

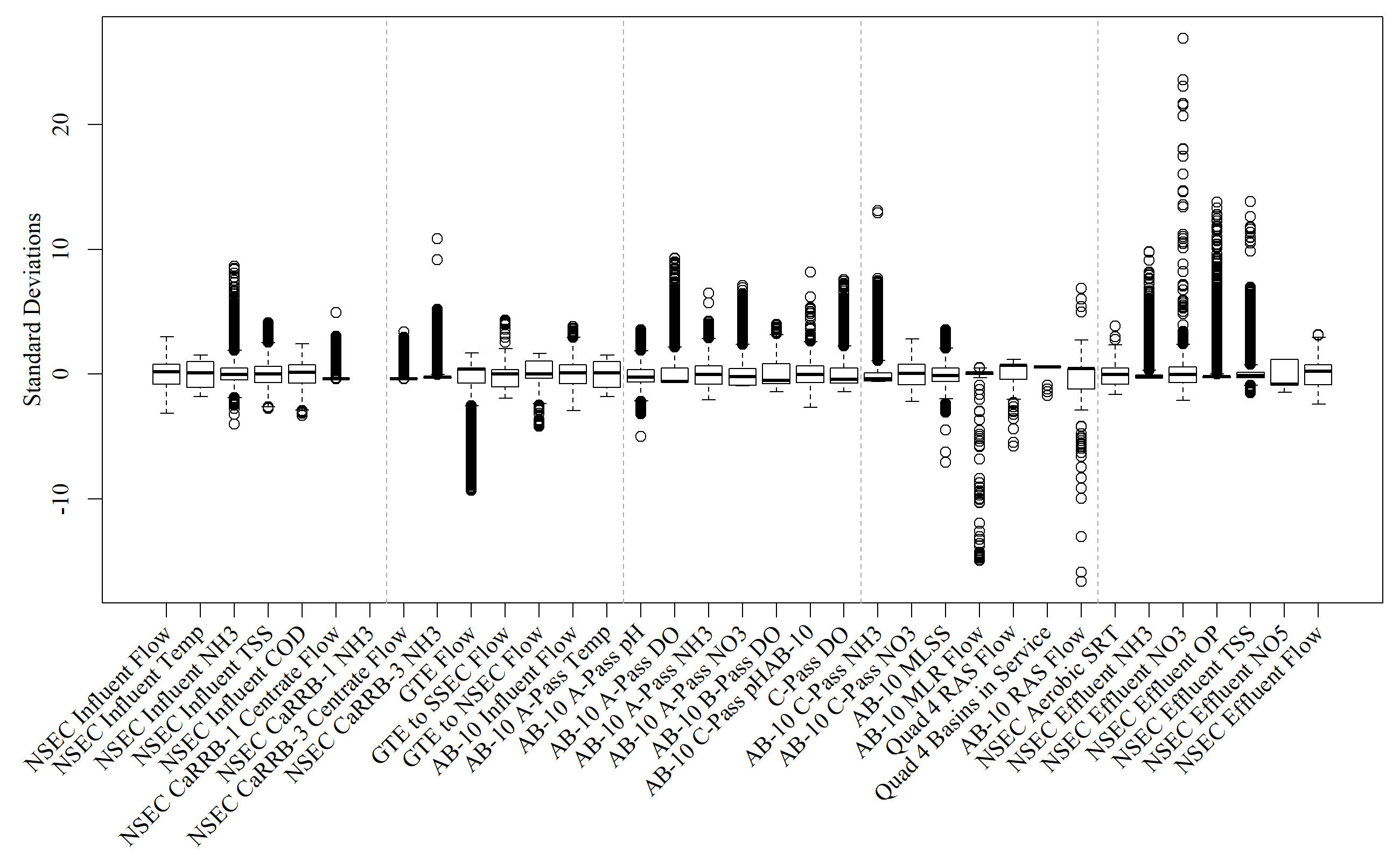
# North process GAM

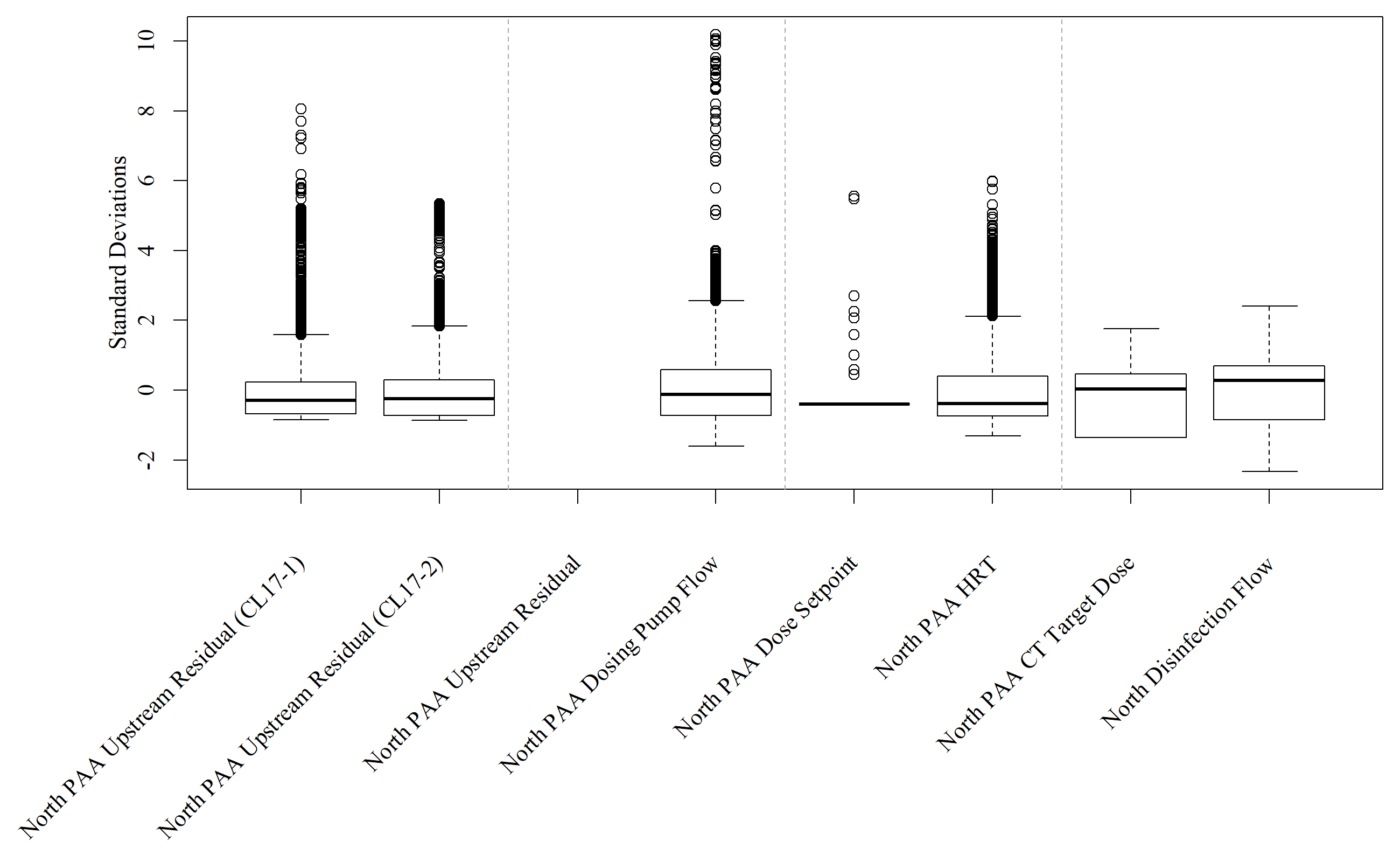
# Merge sensor and daily data

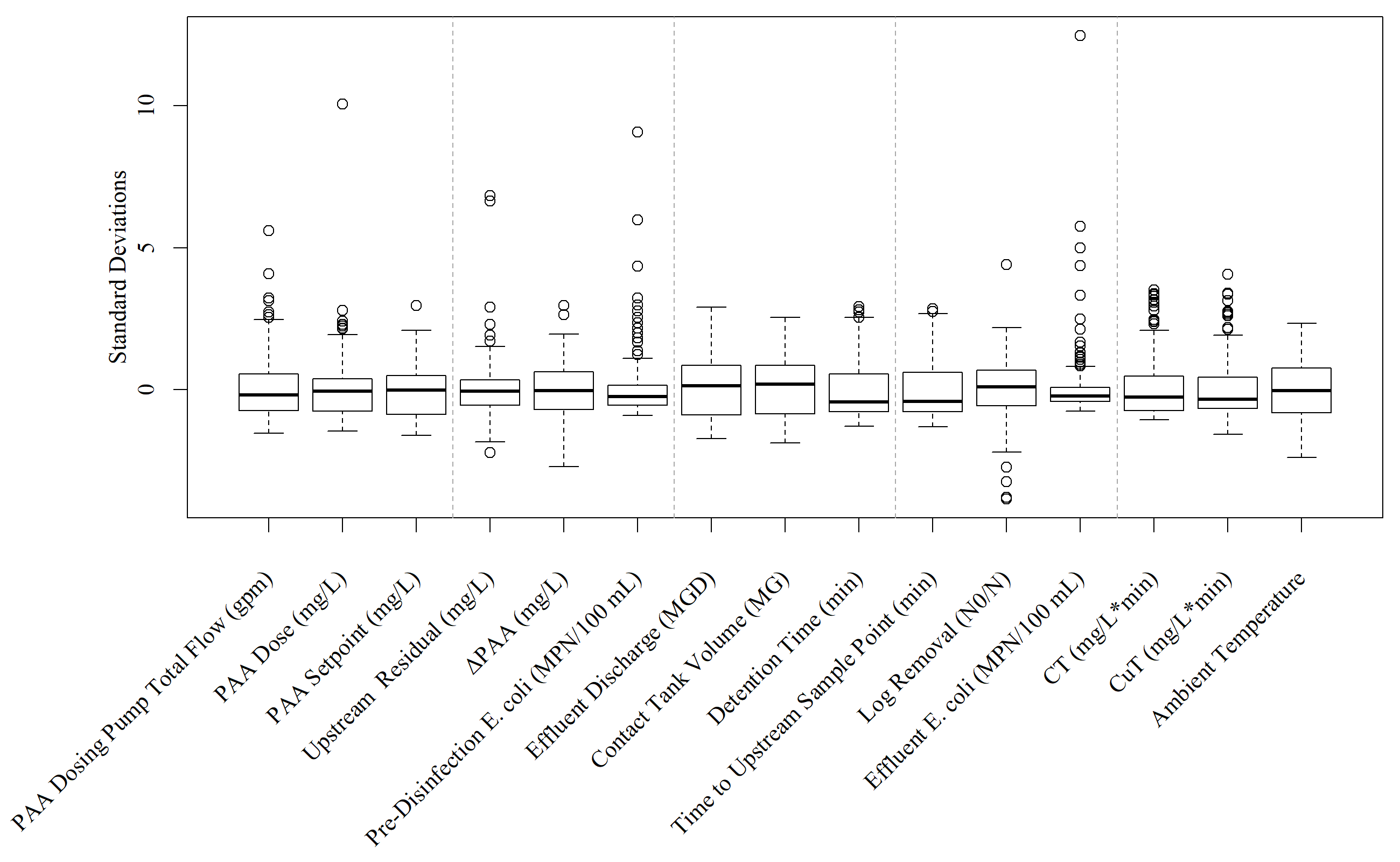
# Pairwise plots

# Supplementary Information

## Figures

 **Figure S1.** Centered and scaled boxplots of north secondary online data from MWRD

 **Figure S2.** Centered and scaled boxplots of north disinfection online data from MWRD

 **Figure S3.** Centered and scaled boxplots of north disinfection grab sample data from MWRD

## Tables

**Table S1.** PCA variable contributions

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Dim.1 | Dim.2 | Dim.3 | Dim.4 | Dim.5 | Dim.6 | Dim.7 | Dim.8 |
| Pre.disinf.E..coli..LOG.MPN.100mL. | 27.36 | 0.26 | 5.75 | 5.23 | 13.98 | 13.24 | 0.01 | 34.17 |
| NSEC.Aerobic.SRT | 15.87 | 0.03 | 13.62 | 1.30 | 12.37 | 56.80 | 0.01 | 0.01 |
| NSEC.Effluent.NH3 | 0.90 | 42.21 | 0.30 | 1.64 | 6.14 | 1.12 | 46.15 | 1.55 |
| NSEC.Effluent.OP | 2.07 | 20.11 | 0.11 | 49.01 | 5.57 | 1.27 | 20.65 | 1.21 |
| NSEC.Effluent.NO3 | 0.15 | 27.91 | 1.09 | 33.31 | 9.50 | 0.47 | 26.14 | 1.43 |
| NSEC.Effluent.TSS | 6.98 | 0.15 | 65.14 | 0.00 | 13.37 | 13.22 | 0.09 | 1.05 |
| NSEC.Effluent.NO5 | 34.85 | 1.56 | 4.42 | 0.23 | 0.15 | 4.02 | 3.16 | 51.61 |
| NSEC.Effluent.Flow | 11.82 | 7.78 | 9.58 | 9.28 | 38.92 | 9.87 | 3.79 | 8.96 |