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 to keep below *E. coli* limits of 126 (most probable number [MPN])/100 mL based on a 30-day geometric mean and 252 MPN/100 mL based on a 7-day geometric mean. 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

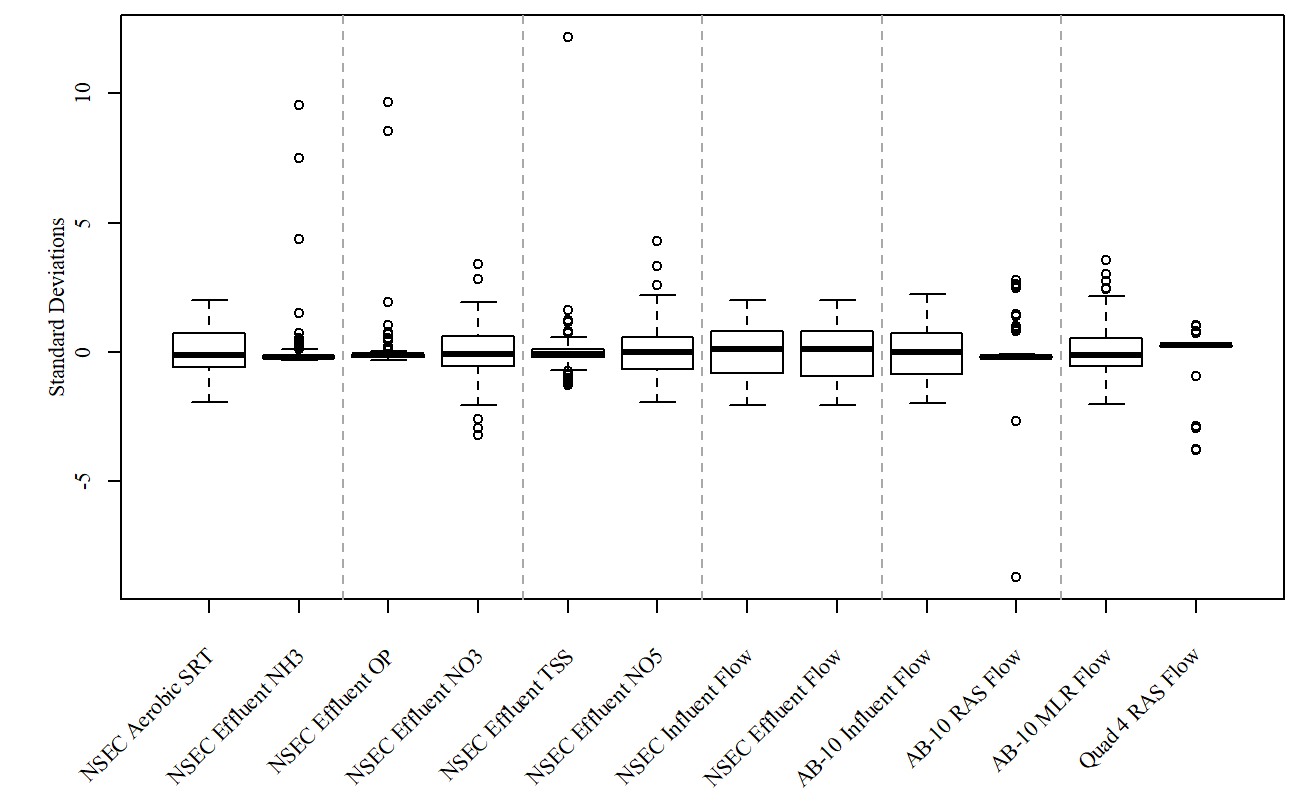
Data was provided by MWRD at a variety of intervals and locations in the treatment process:

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |

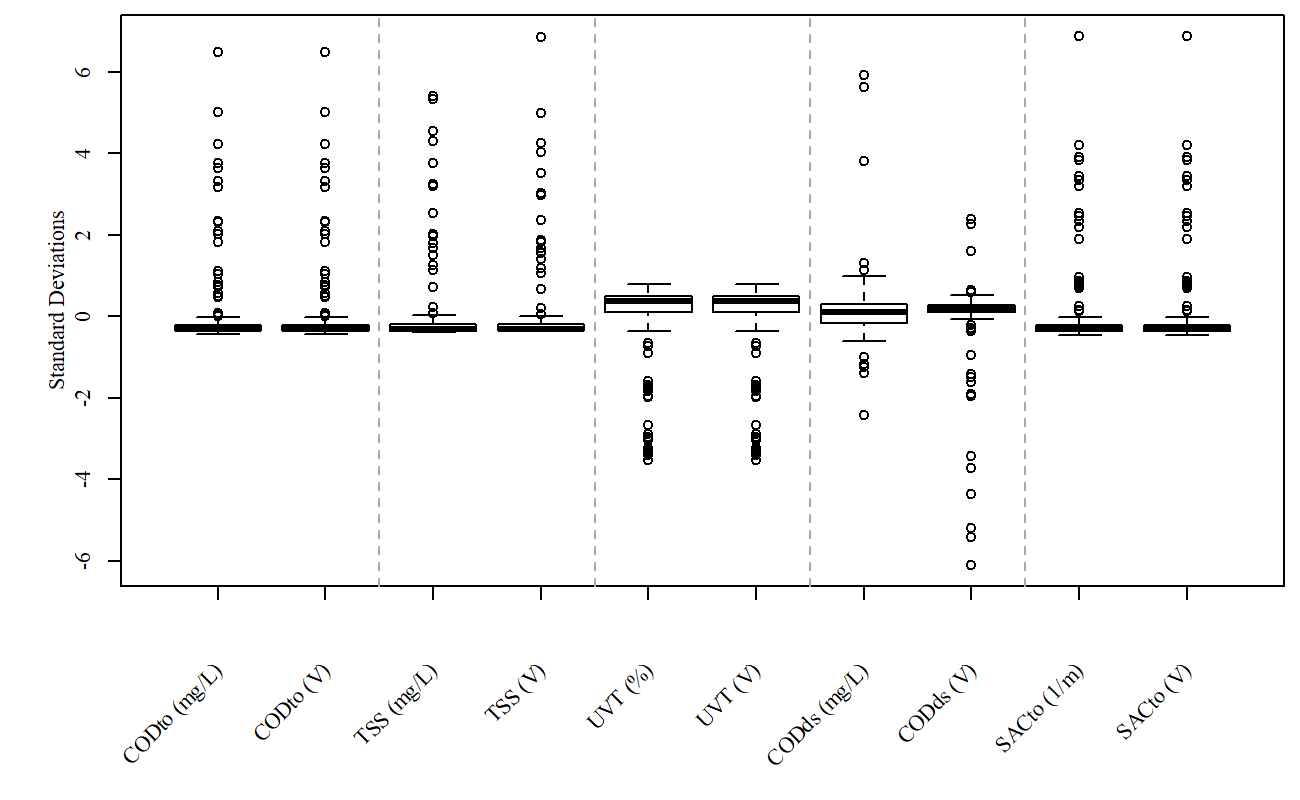
# Data cleaning

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) and data from a visual spectrum analyzer (Figure 2).



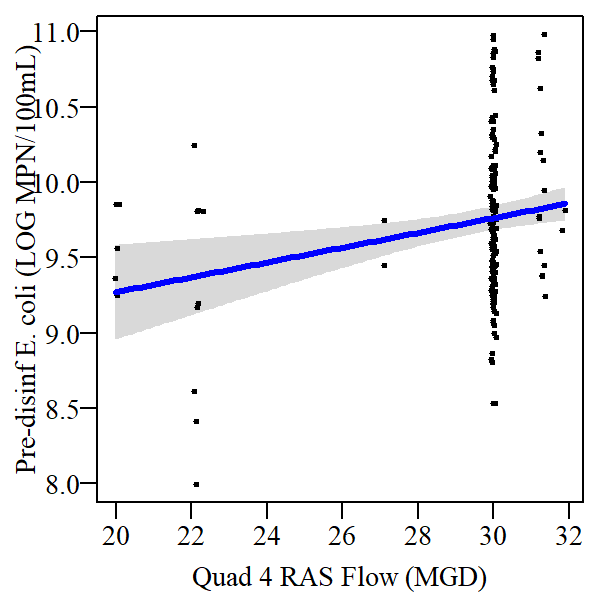
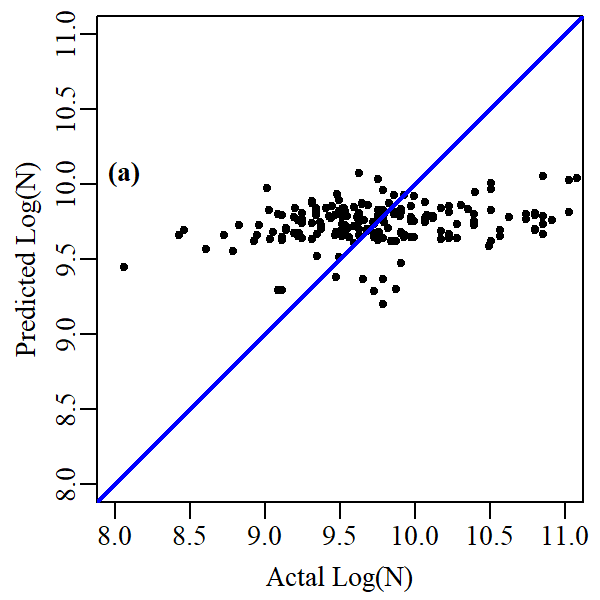
**Figure 1.** Boxplots of water quality variables from North secondary used to predict *E. coli* from 2018-06-01 / 2018-12-01.



**Figure 2.** Boxplots of water quality variables immediately upstream of PAA dosing used to predict *E. coli* from 2018-06-01 / 2018-12-01.

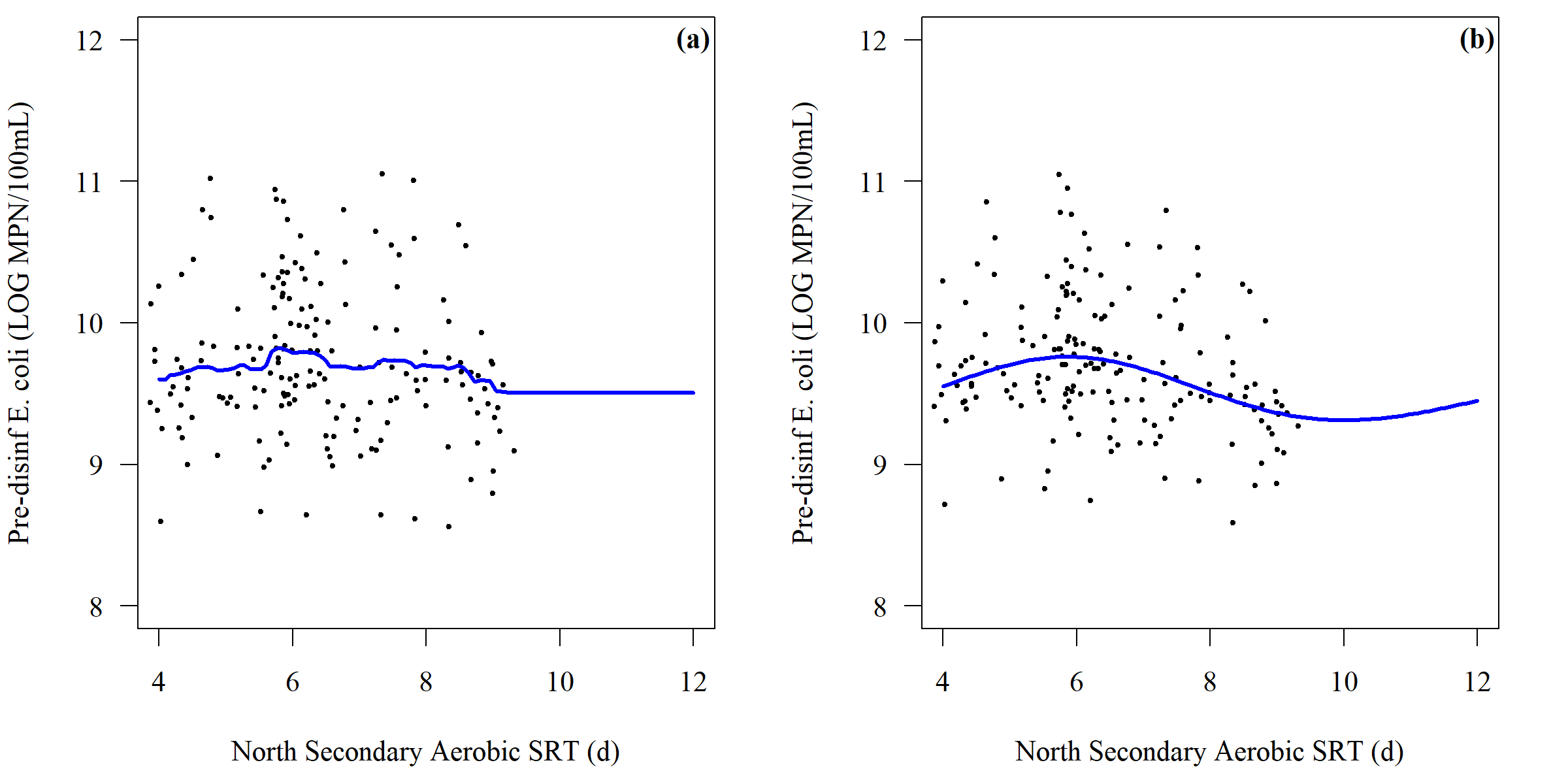
# Methods

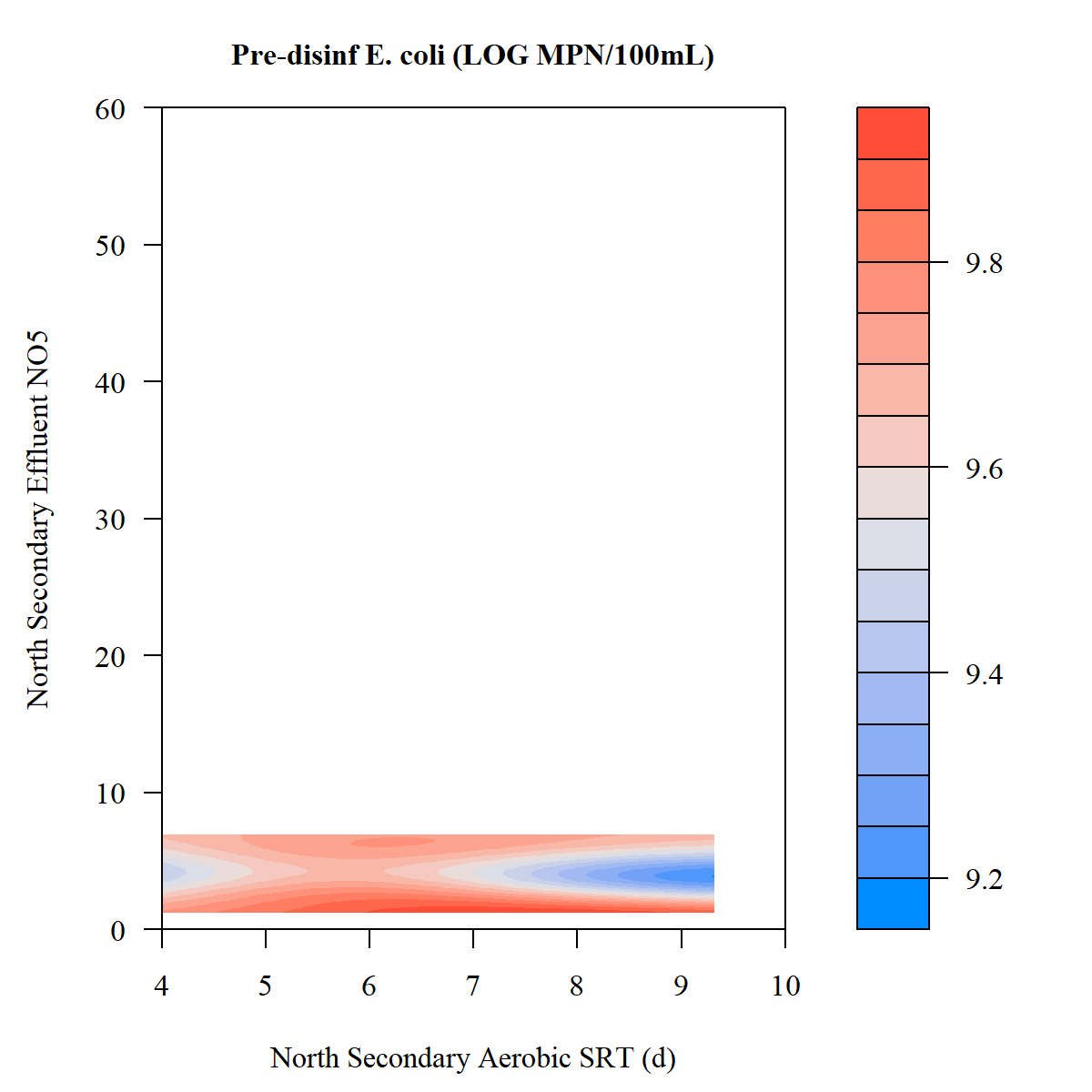
## Regression Models



**Figure 2.** Multiple linear regression model fit for pre-disinfection *E. coli*. Black circles represent actual observations. (a) Predicted pre-disinfection *E.coli* is plotted against actual observations. The blue line represents a perfect model fit. (b) Variation of predicted pre-disinfection *E. coli* given a range of aerobic solid retention times (SRTs) for north secondary.

## 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

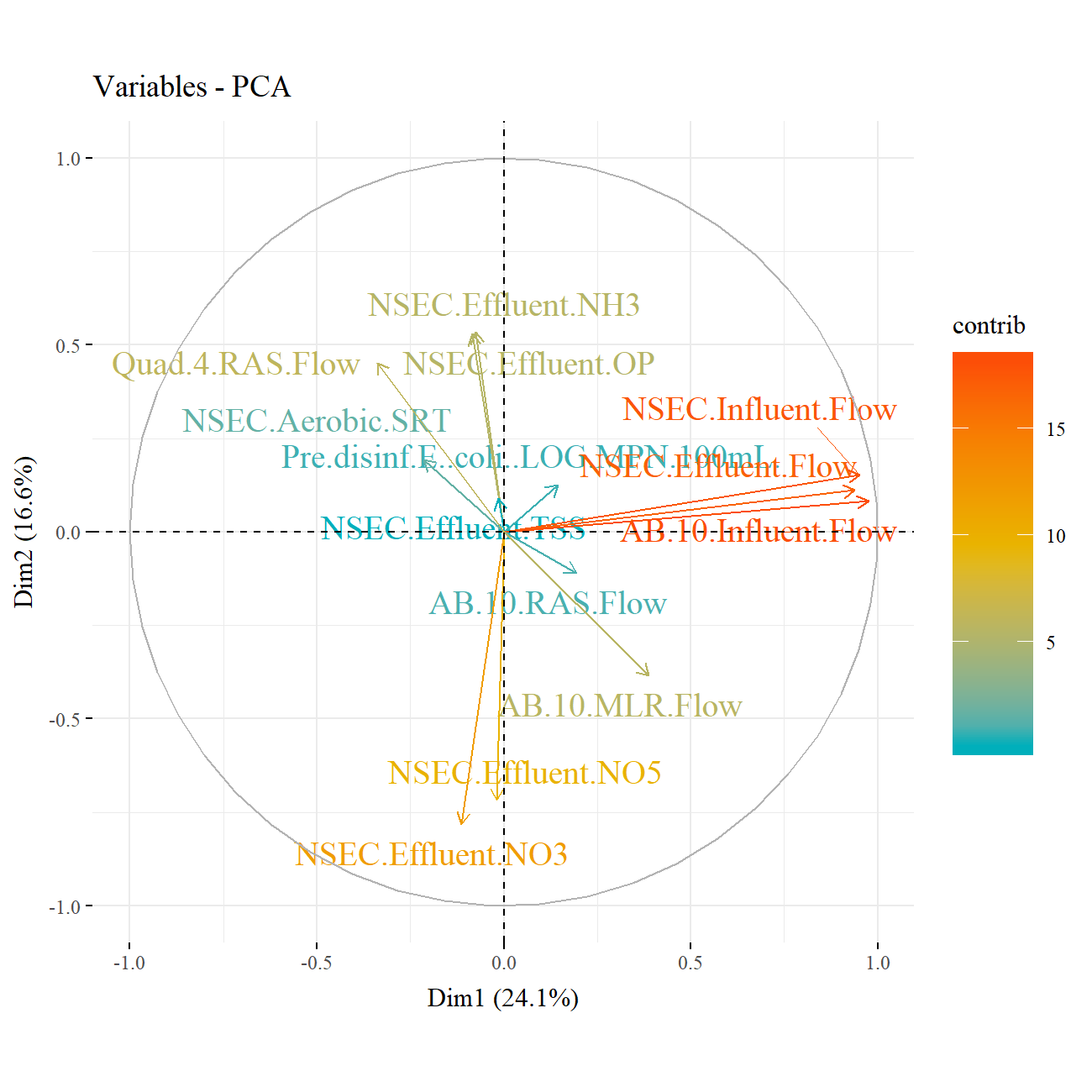
 **Figure .** Support vector machine model for predicting *E. coli* into the PAA disinfection basin

## Generalized Additive Models

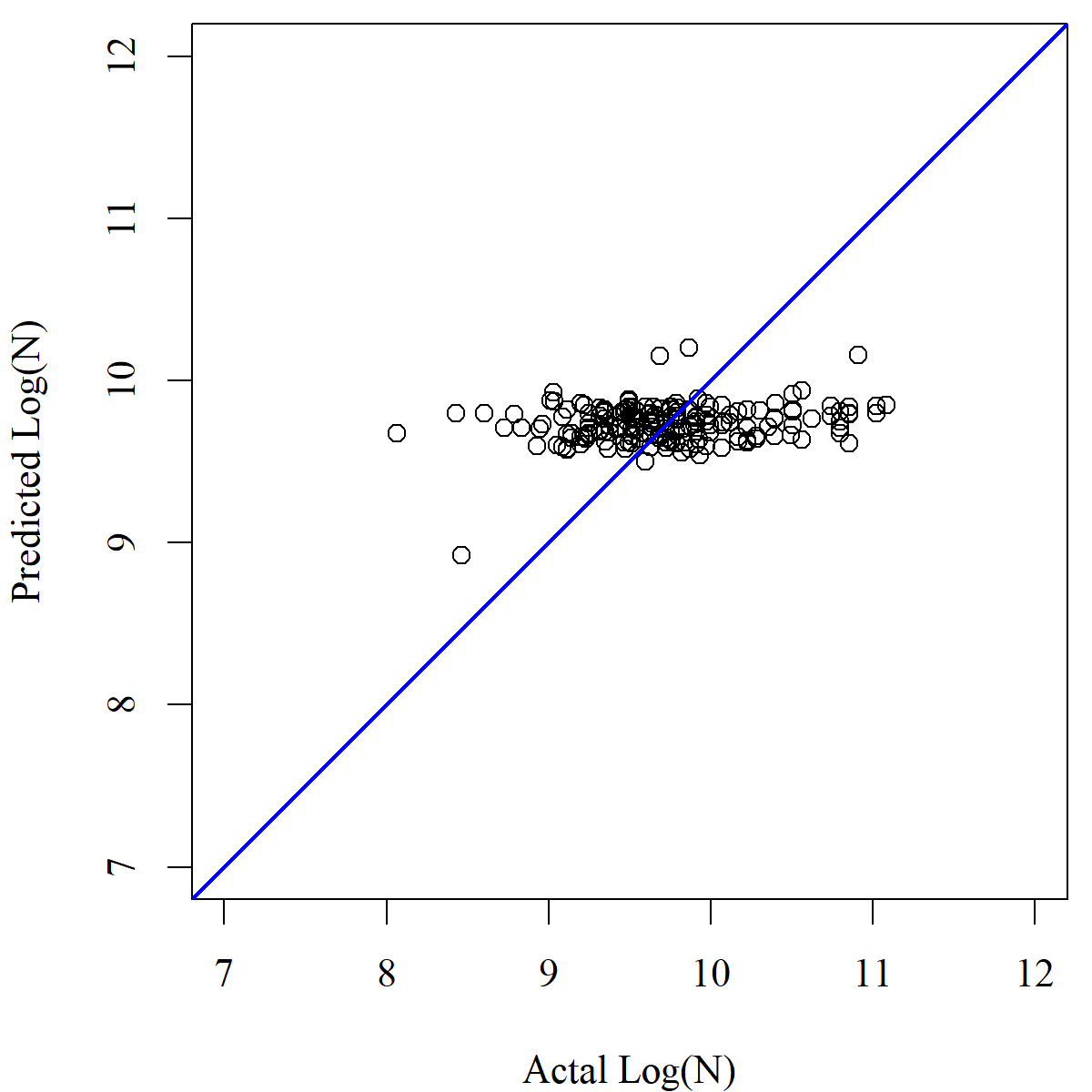
## Predictive model comparison

|  |  |
| --- | --- |
| Model | R.squared |
| Multiple Lienar Regression | 0.07 |
| Random Forest | 0.07 |
| Generalized Addative Model | 0.01 |

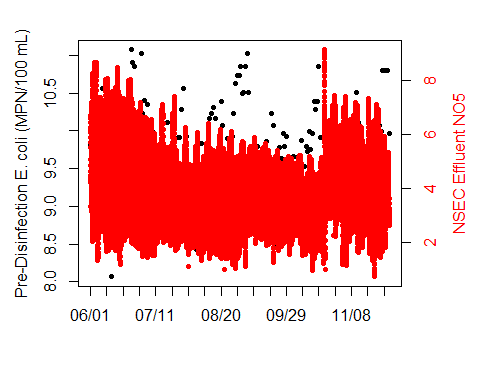
## Principal Component Analysis

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

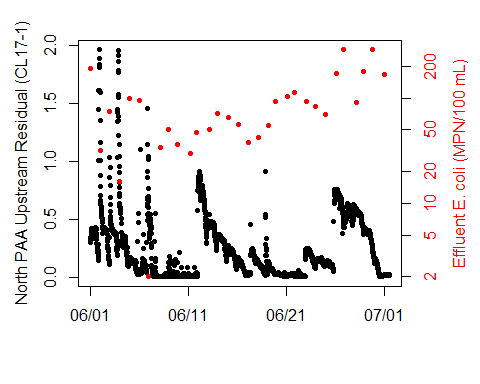
## Partial Least Squares

 **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.

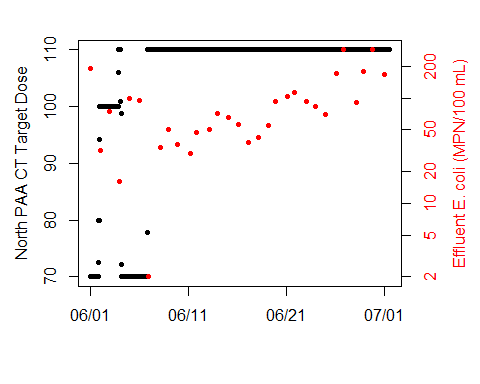
# Plot



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"))

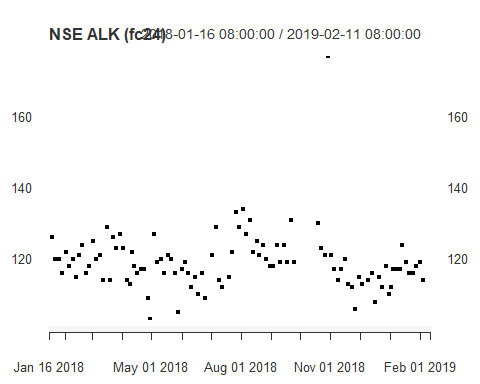


## North secondary effluent lab data  
nsec.eff.lab.fc24 <- as.data.frame(read\_excel("North Secondary and Disinfection Process Data\_20190215.xlsx",   
 sheet = "NSEC Eff Lab Data (FC24)", col\_names = FALSE,   
 col\_types = c("date", "numeric", "skip",   
 "numeric", "numeric", "numeric",   
 "numeric",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

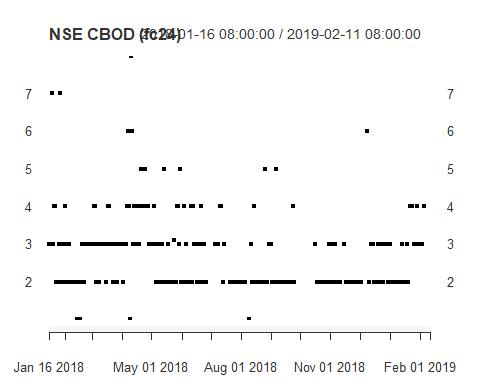
## New names:  
## \* `` -> ...1  
## \* `` -> ...2  
## \* `` -> ...3  
## \* `` -> ...4  
## \* `` -> ...5  
## \* ... and 7 more problems

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))

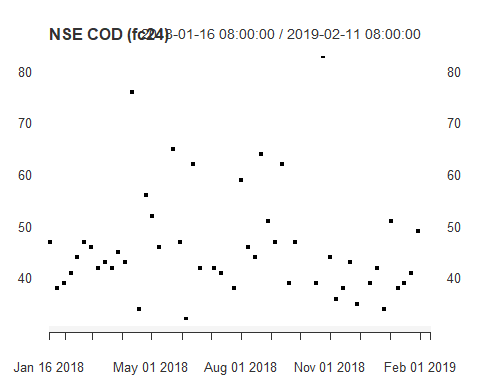
## $`NSE ALK (fc24)`



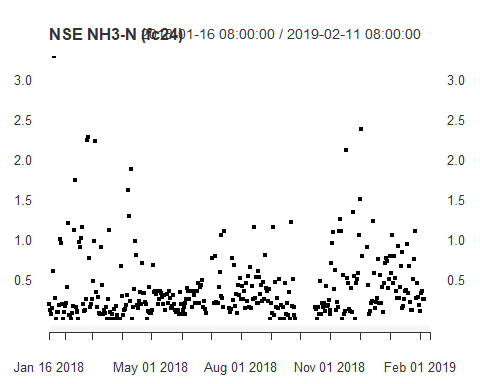
##   
## $`NSE CBOD (fc24)`



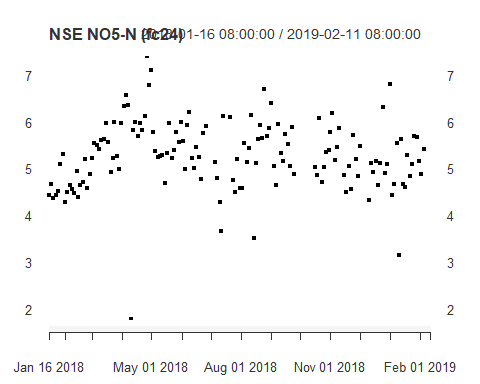
##   
## $`NSE COD (fc24)`



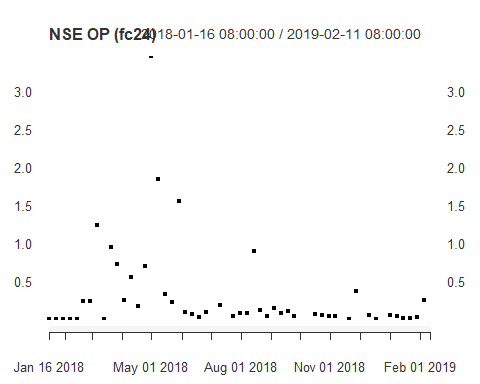
##   
## $`NSE NH3-N (fc24)`



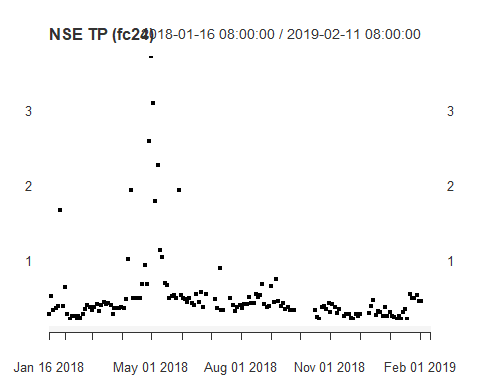
##   
## $`NSE NO5-N (fc24)`



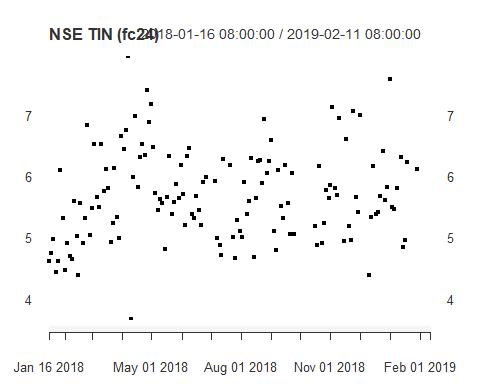
##   
## $`NSE OP (fc24)`



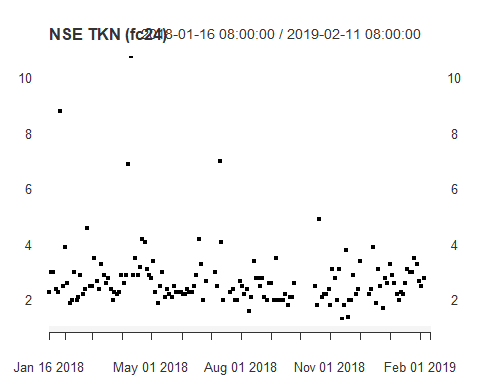
##   
## $`NSE TP (fc24)`



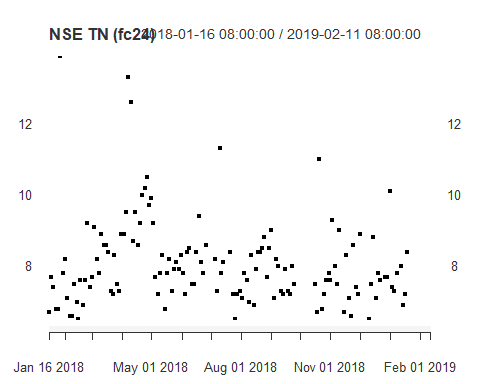
##   
## $`NSE TIN (fc24)`



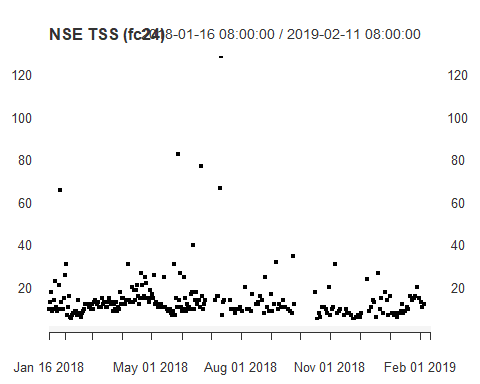
##   
## $`NSE TKN (fc24)`



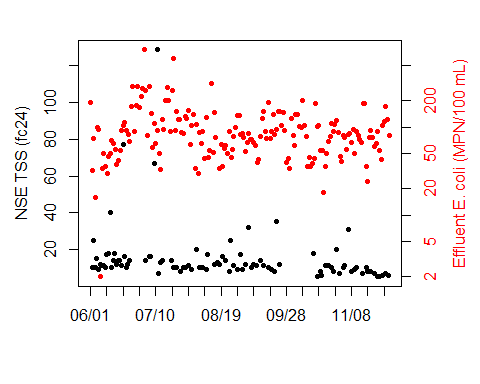
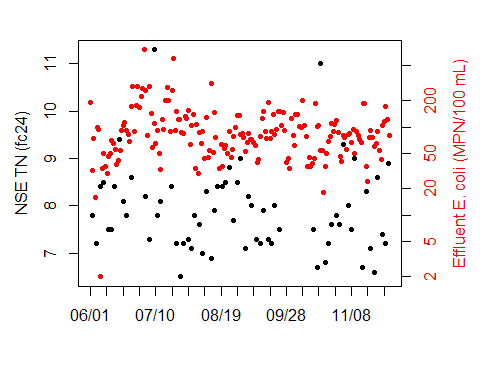
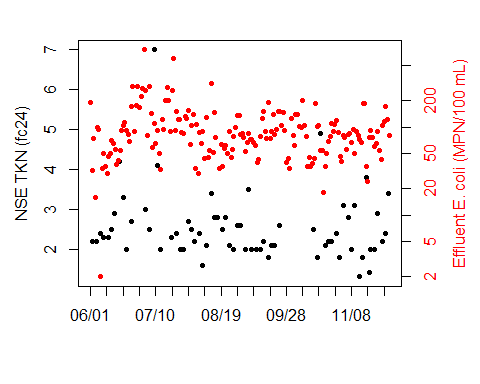
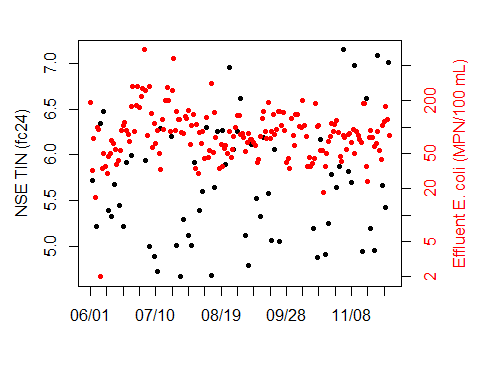
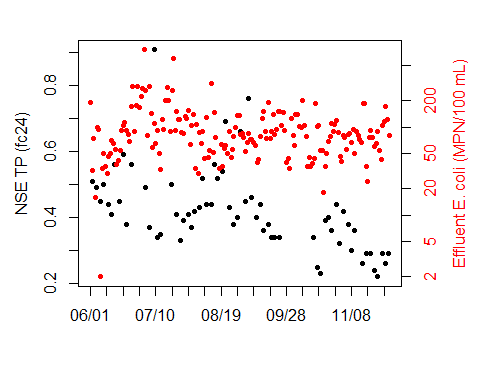
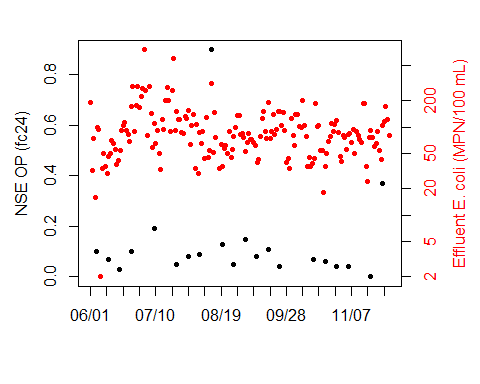
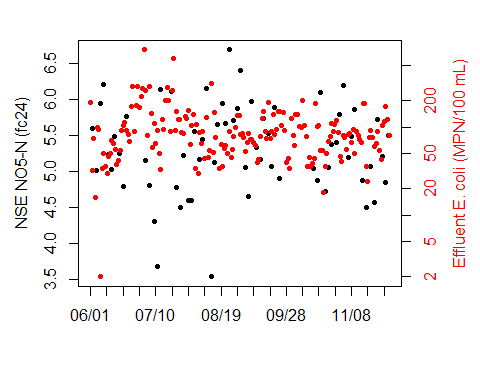
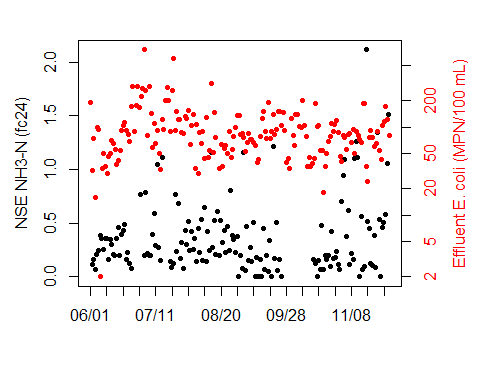
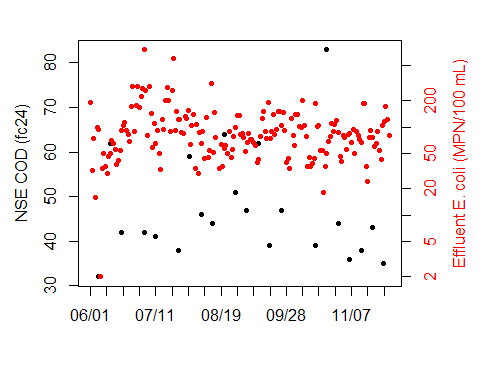
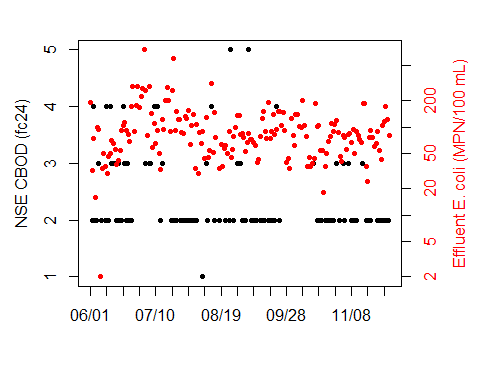
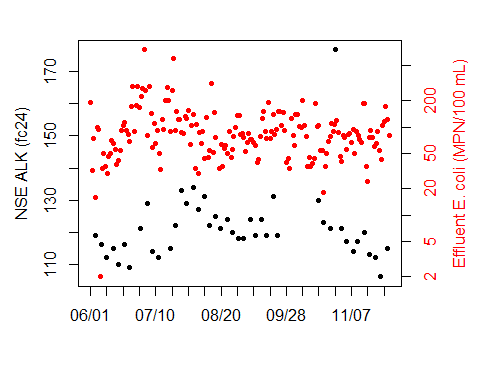
##   
## $`NSE TN (fc24)`



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

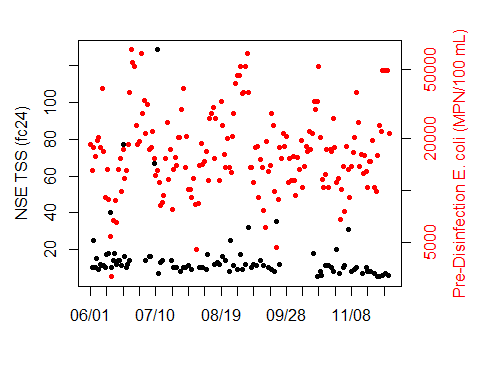
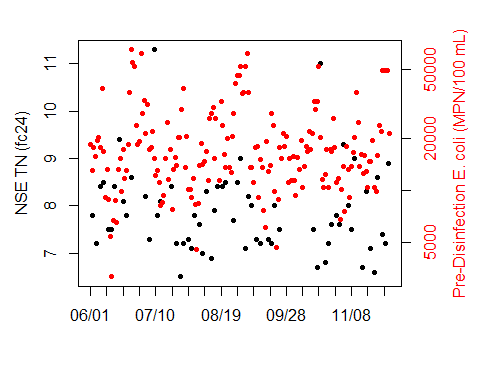
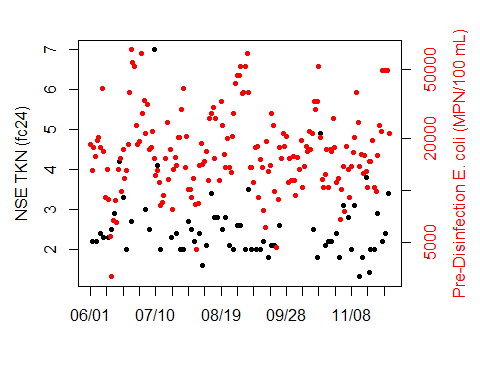
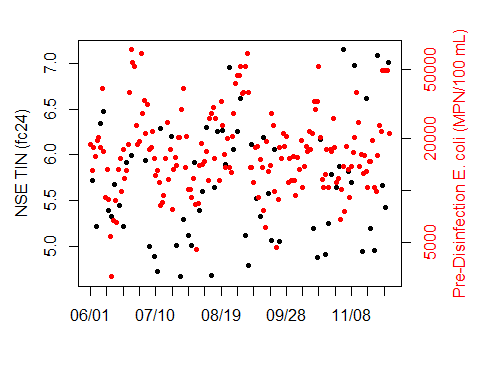
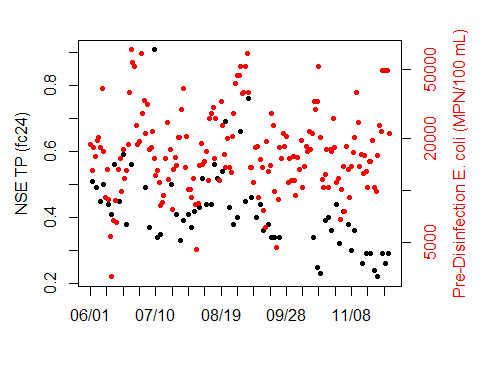
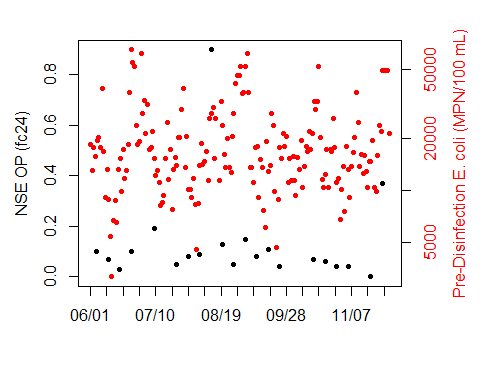
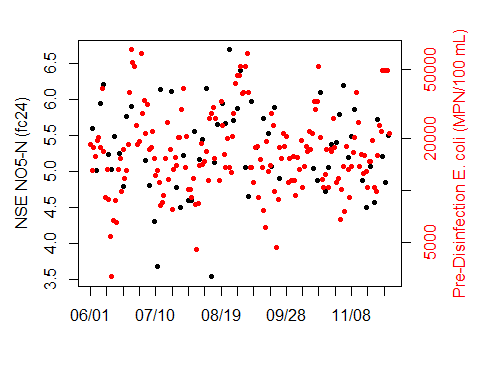
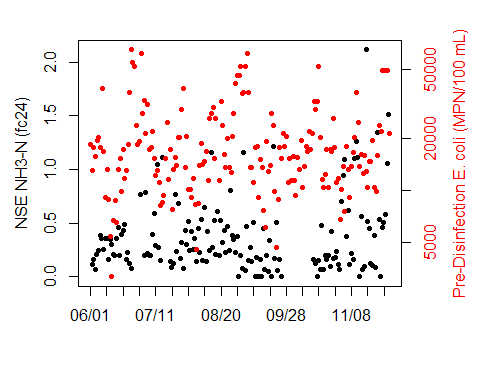
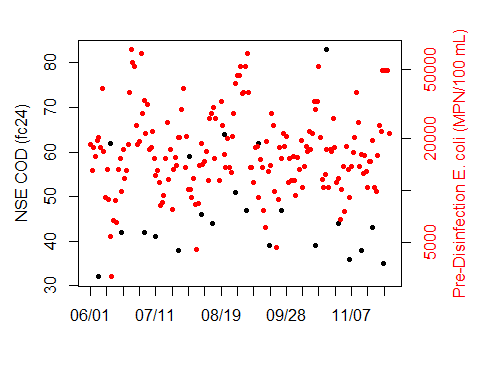
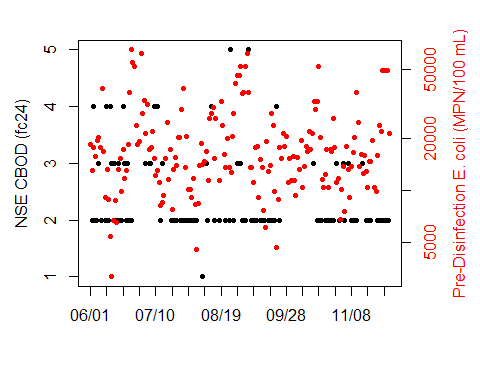
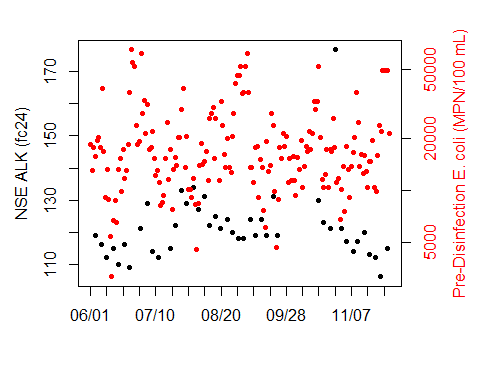


sapply(nsec.eff.lab.fc24, function(data1) {  
 # data1 <- n.paa.online[,7]  
 data2 <- n.paa.grab[,12]  
   
 label1 <- colnames(data1)  
 label2 <- colnames(data2)  
   
 if (range(index(data1)[which(!is.na(data1))])[1] < range(index(data2)[which(!is.na(data2))])[1]) {  
 r1 <- range(index(data2)[which(!is.na(data2))])[1]  
 } 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) NSE CBOD (fc24) NSE COD (fc24) NSE NH3-N (fc24)  
## [1,] 0 0 0 0  
## [2,] 10 10 10 10  
## [3,] 20 20 20 20  
## [4,] 30 30 30 30  
## [5,] 40 40 40 40  
## [6,] 50 50 50 50  
## [7,] 60 60 60 60  
## [8,] 70 70 70 70  
## [9,] 80 80 80 80  
## [10,] 90 90 90 90  
## [11,] 100 100 100 100  
## [12,] 110 110 110 110  
## [13,] 120 120 120 120  
## [14,] 130 130 130 130  
## [15,] 140 140 140 140  
## [16,] 150 150 150 150  
## [17,] 160 160 160 160  
## [18,] 170 170 170 170  
## [19,] 180 180 180 180  
## NSE NO5-N (fc24) NSE OP (fc24) NSE TP (fc24) NSE TIN (fc24)  
## [1,] 0 0 0 0  
## [2,] 10 10 10 10  
## [3,] 20 20 20 20  
## [4,] 30 30 30 30  
## [5,] 40 40 40 40  
## [6,] 50 50 50 50  
## [7,] 60 60 60 60  
## [8,] 70 70 70 70  
## [9,] 80 80 80 80  
## [10,] 90 90 90 90  
## [11,] 100 100 100 100  
## [12,] 110 110 110 110  
## [13,] 120 120 120 120  
## [14,] 130 130 130 130  
## [15,] 140 140 140 140  
## [16,] 150 150 150 150  
## [17,] 160 160 160 160  
## [18,] 170 170 170 170  
## [19,] 180 180 180 180  
## NSE TKN (fc24) NSE TN (fc24) NSE TSS (fc24)  
## [1,] 0 0 0  
## [2,] 10 10 10  
## [3,] 20 20 20  
## [4,] 30 30 30  
## [5,] 40 40 40  
## [6,] 50 50 50  
## [7,] 60 60 60  
## [8,] 70 70 70  
## [9,] 80 80 80  
## [10,] 90 90 90  
## [11,] 100 100 100  
## [12,] 110 110 110  
## [13,] 120 120 120  
## [14,] 130 130 130  
## [15,] 140 140 140  
## [16,] 150 150 150  
## [17,] 160 160 160  
## [18,] 170 170 170  
## [19,] 180 180 180

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"))  
})

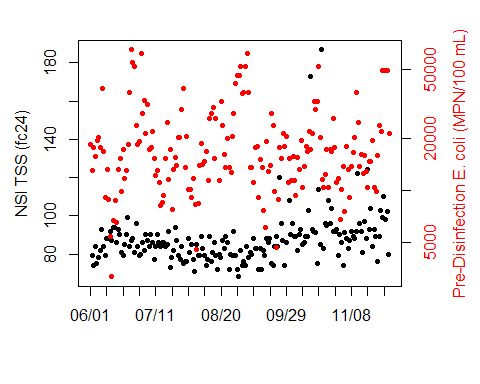
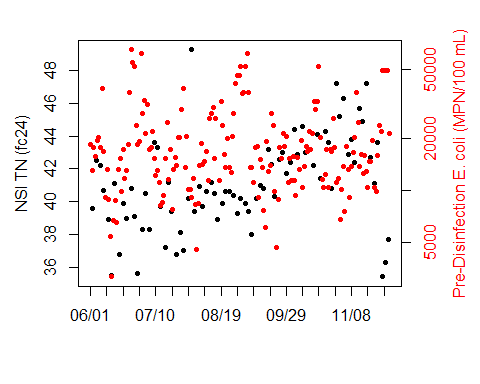
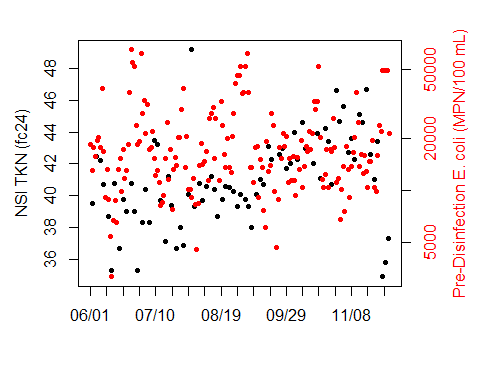
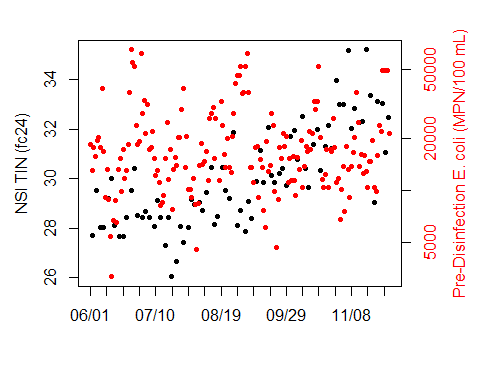
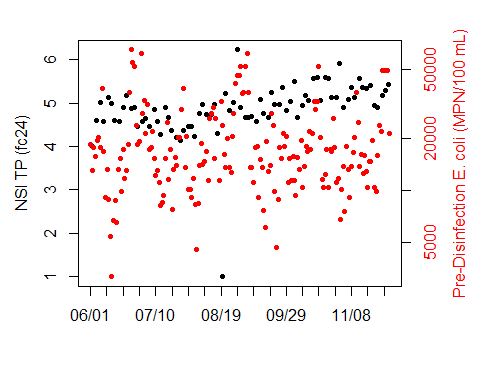
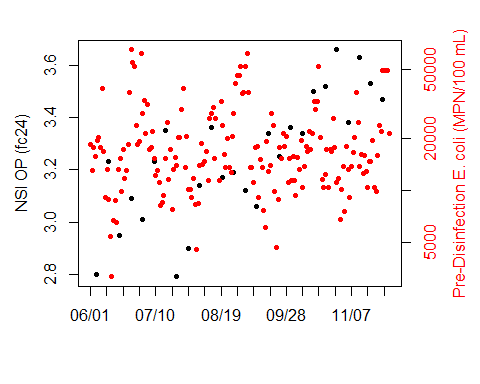
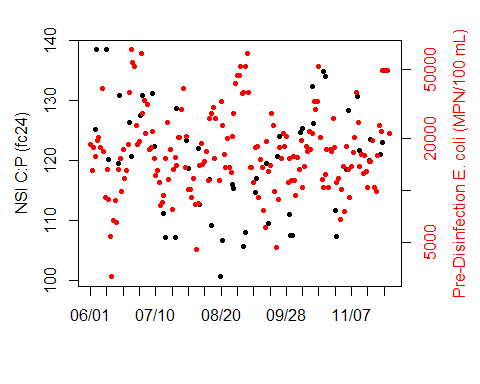
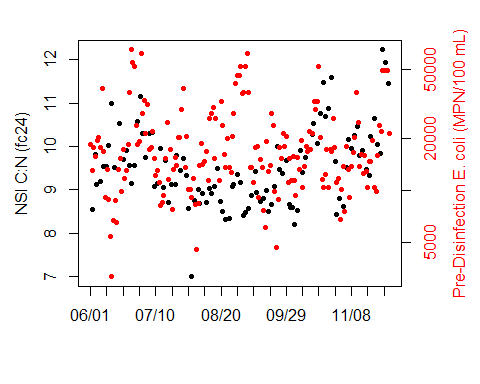
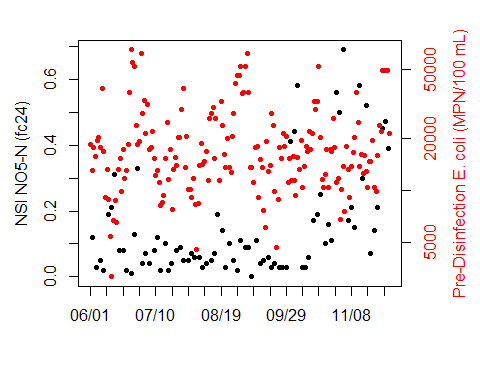
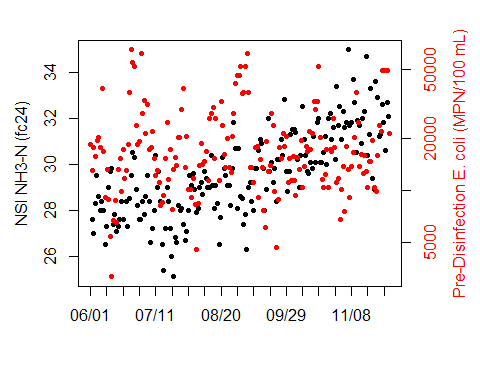
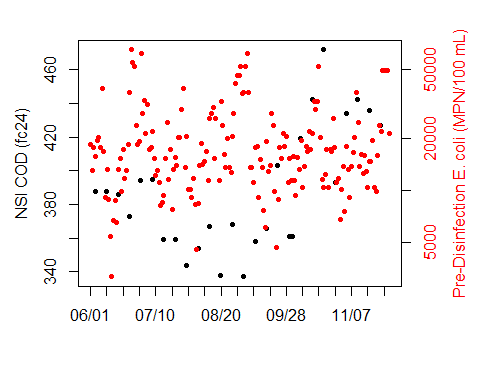
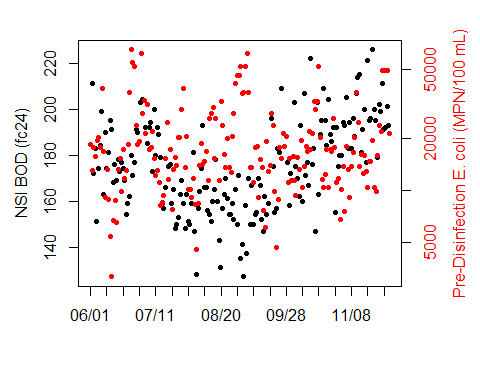
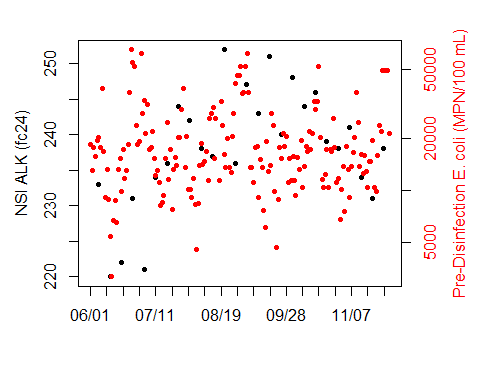


## NSE ALK (fc24) NSE CBOD (fc24) NSE COD (fc24) NSE NH3-N (fc24)  
## [1,] 0 0 0 0  
## [2,] 10 10 10 10  
## [3,] 20 20 20 20  
## [4,] 30 30 30 30  
## [5,] 40 40 40 40  
## [6,] 50 50 50 50  
## [7,] 60 60 60 60  
## [8,] 70 70 70 70  
## [9,] 80 80 80 80  
## [10,] 90 90 90 90  
## [11,] 100 100 100 100  
## [12,] 110 110 110 110  
## [13,] 120 120 120 120  
## [14,] 130 130 130 130  
## [15,] 140 140 140 140  
## [16,] 150 150 150 150  
## [17,] 160 160 160 160  
## [18,] 170 170 170 170  
## [19,] 180 180 180 180  
## NSE NO5-N (fc24) NSE OP (fc24) NSE TP (fc24) NSE TIN (fc24)  
## [1,] 0 0 0 0  
## [2,] 10 10 10 10  
## [3,] 20 20 20 20  
## [4,] 30 30 30 30  
## [5,] 40 40 40 40  
## [6,] 50 50 50 50  
## [7,] 60 60 60 60  
## [8,] 70 70 70 70  
## [9,] 80 80 80 80  
## [10,] 90 90 90 90  
## [11,] 100 100 100 100  
## [12,] 110 110 110 110  
## [13,] 120 120 120 120  
## [14,] 130 130 130 130  
## [15,] 140 140 140 140  
## [16,] 150 150 150 150  
## [17,] 160 160 160 160  
## [18,] 170 170 170 170  
## [19,] 180 180 180 180  
## NSE TKN (fc24) NSE TN (fc24) NSE TSS (fc24)  
## [1,] 0 0 0  
## [2,] 10 10 10  
## [3,] 20 20 20  
## [4,] 30 30 30  
## [5,] 40 40 40  
## [6,] 50 50 50  
## [7,] 60 60 60  
## [8,] 70 70 70  
## [9,] 80 80 80  
## [10,] 90 90 90  
## [11,] 100 100 100  
## [12,] 110 110 110  
## [13,] 120 120 120  
## [14,] 130 130 130  
## [15,] 140 140 140  
## [16,] 150 150 150  
## [17,] 160 160 160  
## [18,] 170 170 170  
## [19,] 180 180 180

## 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"))  
})

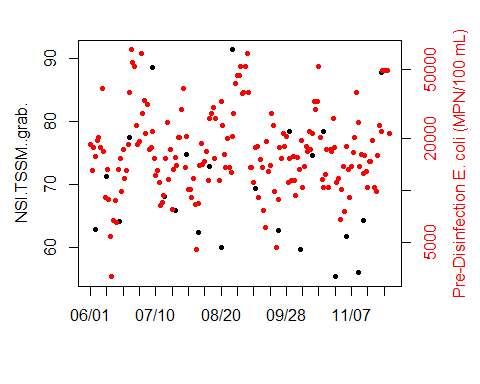
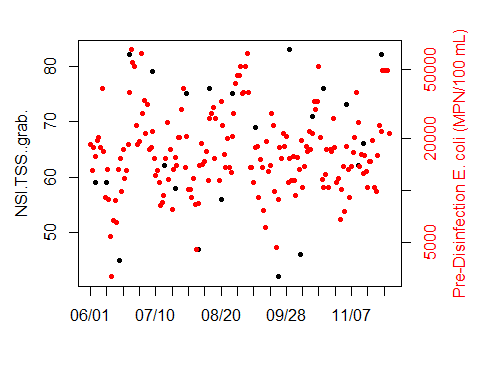
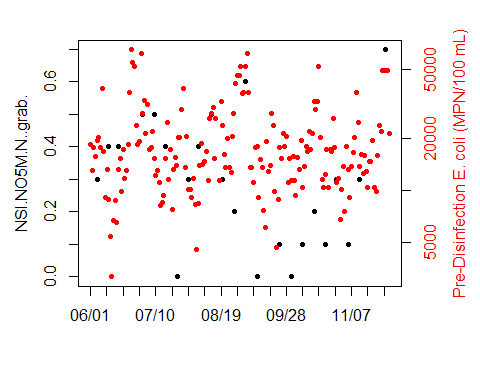
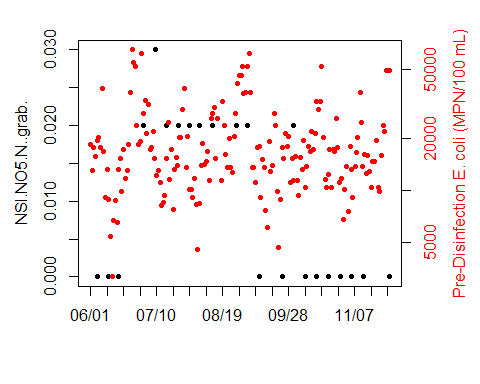
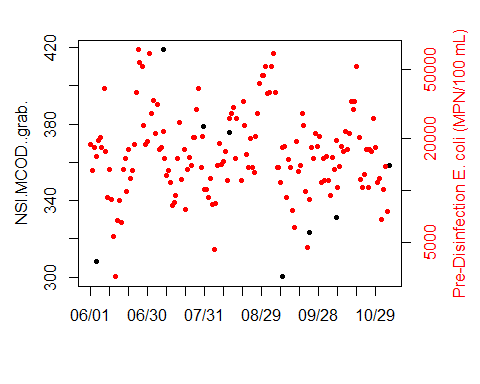
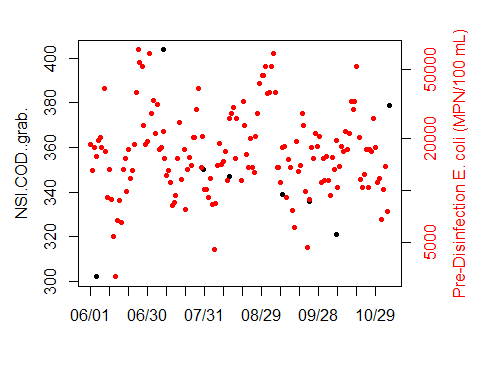


## NSI ALK (fc24) NSI BOD (fc24) NSI COD (fc24) NSI NH3-N (fc24)  
## [1,] 0 0 0 0  
## [2,] 10 10 10 10  
## [3,] 20 20 20 20  
## [4,] 30 30 30 30  
## [5,] 40 40 40 40  
## [6,] 50 50 50 50  
## [7,] 60 60 60 60  
## [8,] 70 70 70 70  
## [9,] 80 80 80 80  
## [10,] 90 90 90 90  
## [11,] 100 100 100 100  
## [12,] 110 110 110 110  
## [13,] 120 120 120 120  
## [14,] 130 130 130 130  
## [15,] 140 140 140 140  
## [16,] 150 150 150 150  
## [17,] 160 160 160 160  
## [18,] 170 170 170 170  
## [19,] 180 180 180 180  
## NSI NO5-N (fc24) NSI C:N (fc24) NSI C:P (fc24) NSI OP (fc24)  
## [1,] 0 0 0 0  
## [2,] 10 10 10 10  
## [3,] 20 20 20 20  
## [4,] 30 30 30 30  
## [5,] 40 40 40 40  
## [6,] 50 50 50 50  
## [7,] 60 60 60 60  
## [8,] 70 70 70 70  
## [9,] 80 80 80 80  
## [10,] 90 90 90 90  
## [11,] 100 100 100 100  
## [12,] 110 110 110 110  
## [13,] 120 120 120 120  
## [14,] 130 130 130 130  
## [15,] 140 140 140 140  
## [16,] 150 150 150 150  
## [17,] 160 160 160 160  
## [18,] 170 170 170 170  
## [19,] 180 180 180 180  
## NSI TP (fc24) NSI TIN (fc24) NSI TKN (fc24) NSI TN (fc24)  
## [1,] 0 0 0 0  
## [2,] 10 10 10 10  
## [3,] 20 20 20 20  
## [4,] 30 30 30 30  
## [5,] 40 40 40 40  
## [6,] 50 50 50 50  
## [7,] 60 60 60 60  
## [8,] 70 70 70 70  
## [9,] 80 80 80 80  
## [10,] 90 90 90 90  
## [11,] 100 100 100 100  
## [12,] 110 110 110 110  
## [13,] 120 120 120 120  
## [14,] 130 130 130 130  
## [15,] 140 140 140 140  
## [16,] 150 150 150 150  
## [17,] 160 160 160 160  
## [18,] 170 170 170 170  
## [19,] 180 180 180 180  
## NSI TSS (fc24)  
## [1,] 0  
## [2,] 10  
## [3,] 20  
## [4,] 30  
## [5,] 40  
## [6,] 50  
## [7,] 60  
## [8,] 70  
## [9,] 80  
## [10,] 90  
## [11,] 100  
## [12,] 110  
## [13,] 120  
## [14,] 130  
## [15,] 140  
## [16,] 150  
## [17,] 160  
## [18,] 170  
## [19,] 180

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  
##   
## $NSI.MCOD..grab.  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150  
##   
## $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  
##   
## $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  
##   
## $NSI.TSS..grab.  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180  
##   
## $NSI.TSSM..grab.  
## [1] 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160  
## [18] 170 180

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.   
## 0.6868132   
## NSE.TSS..fc24.   
## 0.4615385   
## NSE.CBOD..fc24.   
## 0.4560440   
## NSE.TKN..fc24.   
## 0.3681319   
## NSE.NO5.N..fc24.   
## 0.3571429   
## NSE.TP..fc24.   
## 0.3571429   
## NSE.TIN..fc24.   
## 0.3571429   
## NSE.TN..fc24.   
## 0.3571429   
## NSE.ALK..fc24.   
## 0.2417582   
## NSE.COD..fc24.   
## 0.1208791   
## NSE.OP..fc24.   
## 0.1153846

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) 12679 32692 0.388 0.712  
## new.data[, 2] -12959 16254 -0.797 0.456  
## new.data[, 5] -186950 133650 -1.399 0.211  
## new.data[, 6] -180621 132407 -1.364 0.221  
## new.data[, 9] 183944 133424 1.379 0.217  
##   
##   
## R-sq.(adj) = -0.013 Deviance explained = 39.2%  
## GCV = 2.9892e+08 Scale est. = 1.6305e+08 n = 11

## North Secondary Influent Grab

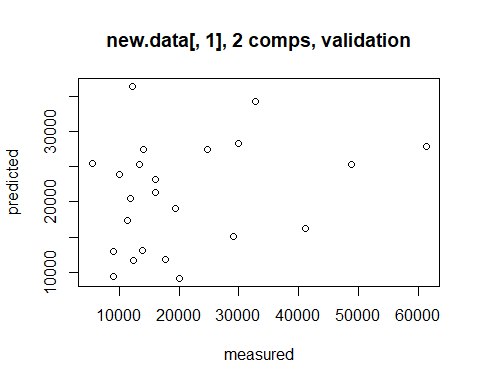
data1 <- nsec.inf.lab.grab.merged  
data2 <- n.paa.grab[,6]  
  
label1 <- colnames(data1)  
label2 <- colnames(data2)  
  
all.data <- merge(data2, data1)  
all.data.index <- which(!is.na(all.data[,1]))  
for(i in 1:(length(all.data.index)-1)) {  
 row.start <- all.data.index[i]  
 row.stop <- all.data.index[i+1]  
 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.   
## 0.04395604   
## NSI.MCOD..grab.   
## 0.04395604   
## NSI.NO5.N..grab.   
## 0.12637363   
## NSI.NO5M.N..grab.   
## 0.12637363   
## NSI.TSS..grab.   
## 0.12087912   
## NSI.TSSM..grab.   
## 0.12637363

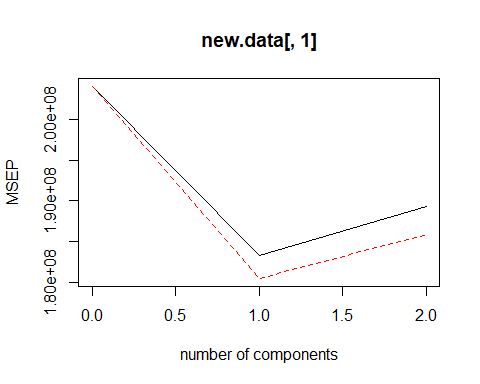
new.data <- new.data[,order(-apply(new.data, 2, function(x) length(which(!is.na(x)))/nrow(new.data)))[1:3]]  
new.data <- na.omit(new.data)  
  
library(pls)  
pls.fit <- plsr(new.data[,1] ~ new.data[,2] + new.data[,3], scale = TRUE, validation = "CV")  
summary(pls.fit)

## Data: X dimension: 23 2   
## Y dimension: 23 1  
## Fit method: kernelpls  
## Number of components considered: 2  
##   
## VALIDATION: RMSEP  
## Cross-validated using 10 random segments.  
## (Intercept) 1 comps 2 comps  
## CV 14283 13538 13761  
## adjCV 14283 13436 13633  
##   
## TRAINING: % variance explained  
## 1 comps 2 comps  
## X 60.60 100.00  
## new.data[, 1] 29.71 30.65

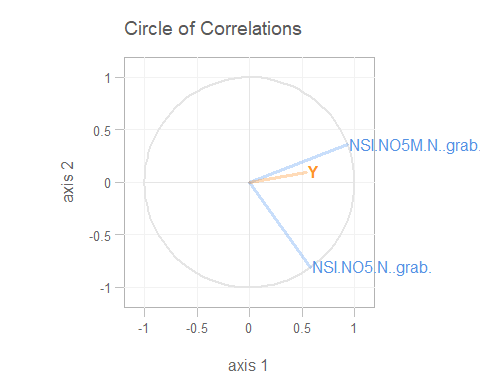
plot(pls.fit)



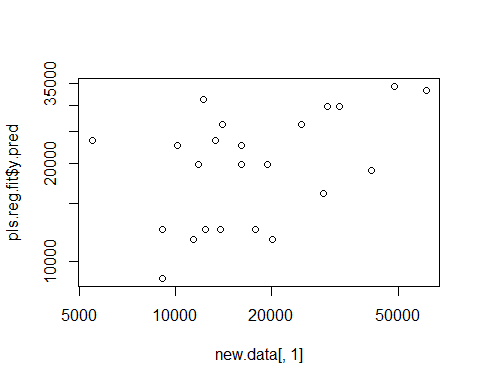
validationplot(pls.fit, val.type = "MSEP")



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) 8844 4831 1.831 0.0821 .  
## new.data[, 2] 138620 247391 0.560 0.5815   
## new.data[, 3] 36516 13616 2.682 0.0143 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
##   
## R-sq.(adj) = 0.237 Deviance explained = 30.7%  
## GCV = 1.7119e+08 Scale est. = 1.4886e+08 n = 23

## 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.   
## 0.8186813   
## NSI.TSS..fc24.   
## 0.8131868   
## NSI.BOD..fc24.   
## 0.7912088   
## NSI.C.N..fc24.   
## 0.5549451   
## NSI.NO5.N..fc24.   
## 0.4340659   
## NSI.TP..fc24.   
## 0.4340659   
## NSI.TIN..fc24.   
## 0.4340659   
## NSI.TKN..fc24.   
## 0.4285714   
## NSI.TN..fc24.   
## 0.4285714   
## NSI.C.P..fc24.   
## 0.2692308   
## NSI.COD..fc24.   
## 0.1483516   
## NSI.ALK..fc24.   
## 0.1428571   
## NSI.OP..fc24.   
## 0.1428571

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) 41369.471 38298.679 1.080 0.284  
## new.data[, 2] -238.308 2110.639 -0.113 0.910  
## new.data[, 3] -150.363 187.793 -0.801 0.426  
## new.data[, 4] 6.639 112.788 0.059 0.953  
## new.data[, 5] 402.942 2656.479 0.152 0.880  
## new.data[, 6] 7066.524 59640.143 0.118 0.906  
## new.data[, 7] 2571.653 2773.016 0.927 0.357  
## new.data[, 8] 1670.564 2095.859 0.797 0.428  
## new.data[, 9] 14444.766 55285.235 0.261 0.795  
## new.data[, 10] -16107.098 55248.770 -0.292 0.772  
##   
##   
## R-sq.(adj) = 0.00406 Deviance explained = 12.4%  
## GCV = 1.8526e+08 Scale est. = 1.6088e+08 n = 76

# GAM

# South process GAM

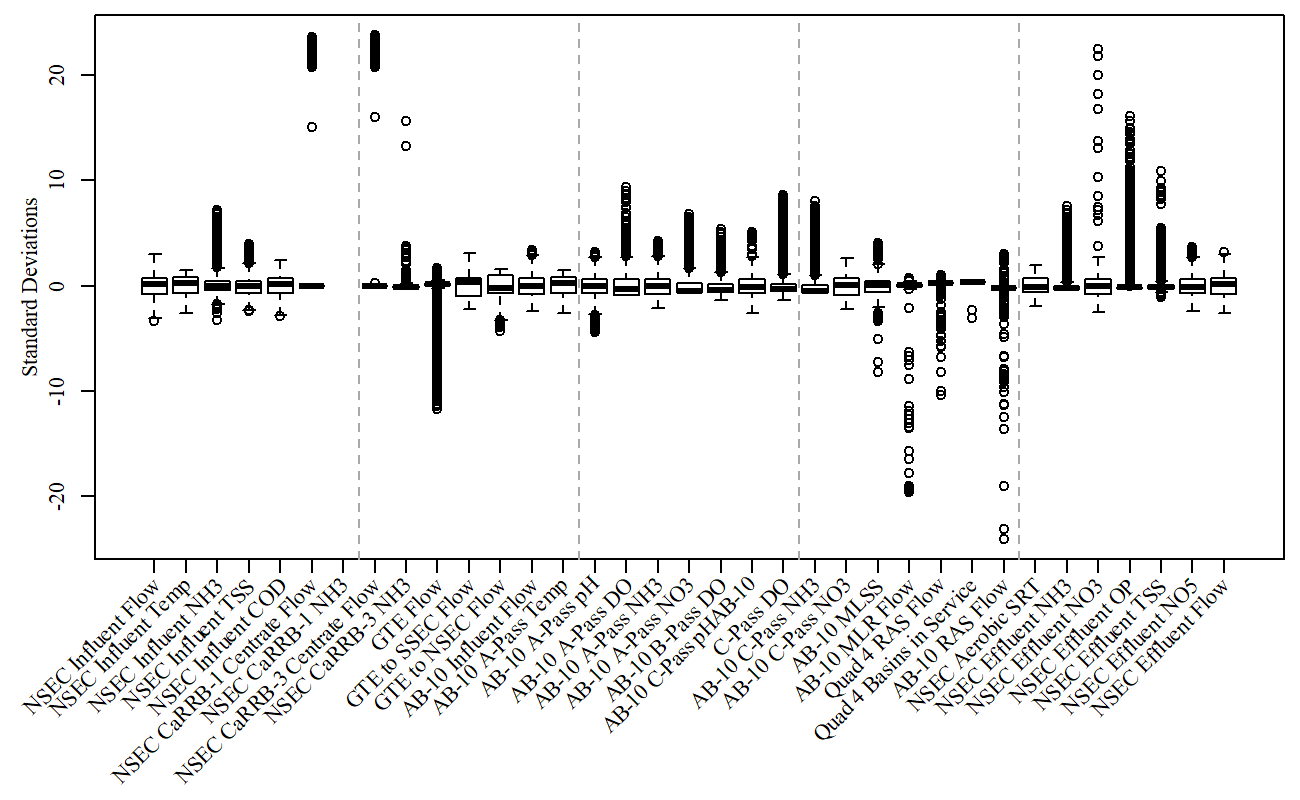
# North process GAM

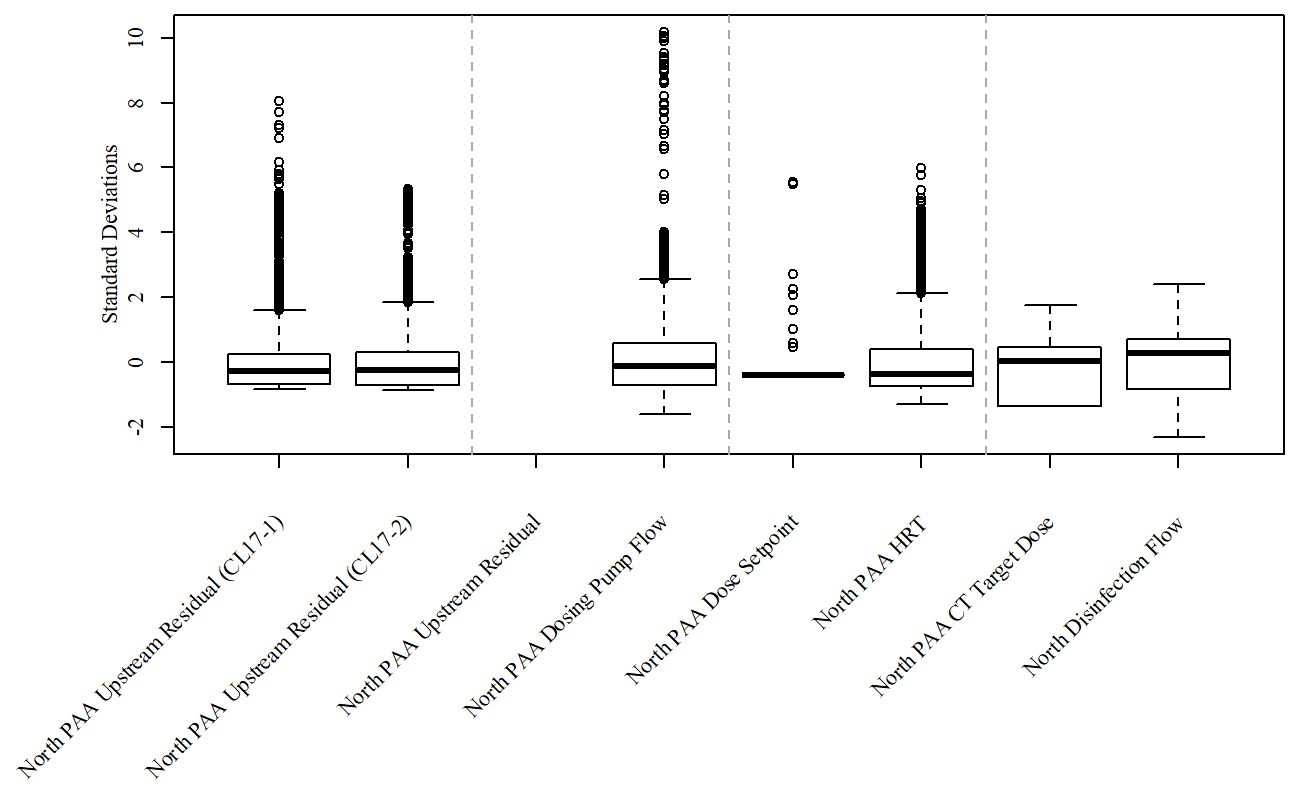
# Merge sensor and daily data

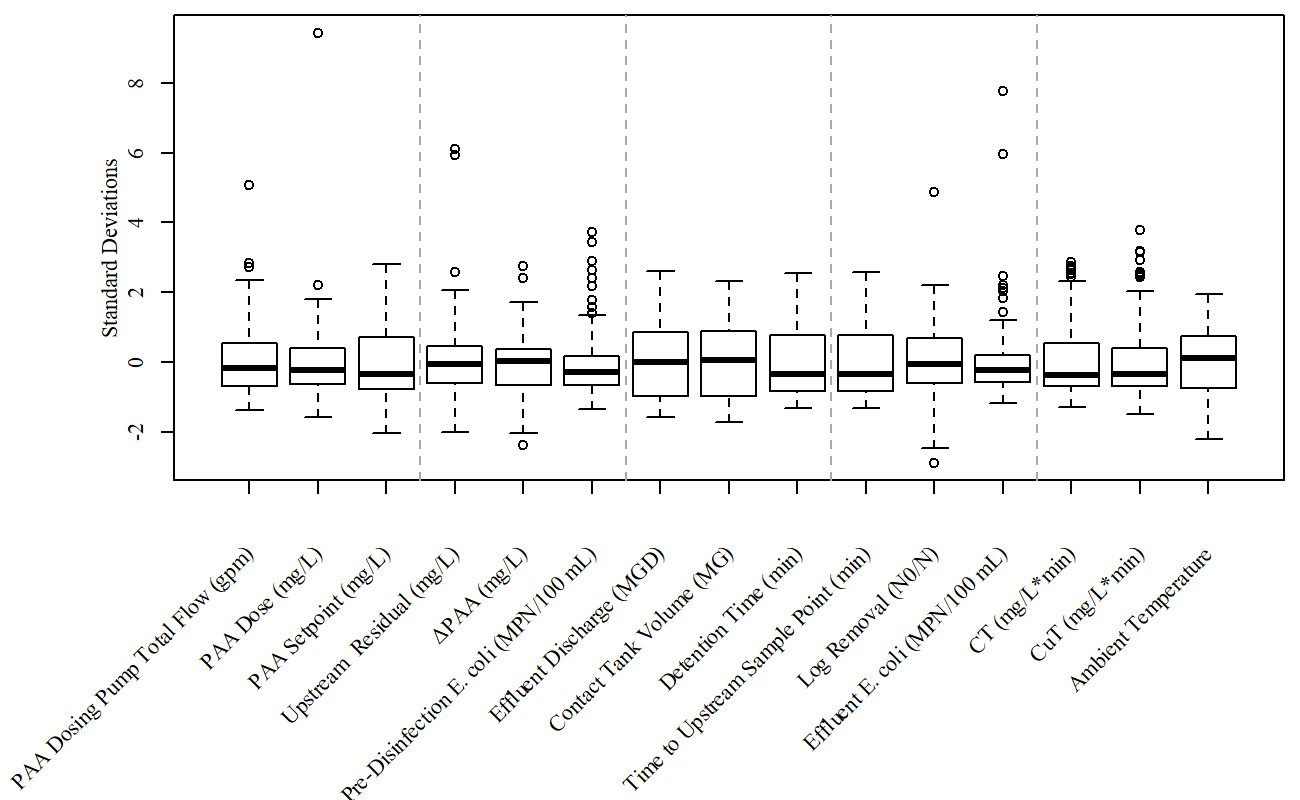
# Pairwise plots

# 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 | Dim.9 | Dim.10 | Dim.11 | Dim.12 | Dim.13 |
| Pre.disinf.E..coli..LOG.MPN.100mL. | 0.66 | 0.74 | 3.79 | 10.36 | 3.38 | 54.21 | 20.24 | 3.45 | 1.00 | 2.00 | 0.17 | 0.01 | 0.00 |
| NSEC.Aerobic.SRT | 1.48 | 1.73 | 14.02 | 0.07 | 28.40 | 0.00 | 4.30 | 35.78 | 2.69 | 2.72 | 8.31 | 0.43 | 0.06 |
| NSEC.Effluent.NH3 | 0.18 | 13.22 | 10.65 | 18.38 | 1.98 | 6.01 | 0.11 | 0.88 | 41.89 | 6.52 | 0.07 | 0.12 | 0.00 |
| NSEC.Effluent.OP | 0.22 | 12.99 | 9.95 | 24.28 | 0.75 | 0.47 | 0.51 | 0.05 | 48.03 | 0.55 | 2.15 | 0.02 | 0.00 |
| NSEC.Effluent.NO3 | 0.40 | 28.25 | 4.08 | 13.17 | 0.71 | 0.57 | 2.17 | 0.95 | 0.00 | 8.20 | 41.41 | 0.00 | 0.09 |
| NSEC.Effluent.TSS | 0.01 | 0.37 | 0.02 | 0.91 | 56.25 | 1.94 | 11.82 | 27.36 | 0.15 | 0.92 | 0.19 | 0.03 | 0.03 |
| NSEC.Effluent.NO5 | 0.01 | 23.84 | 0.29 | 27.46 | 0.00 | 0.91 | 0.42 | 3.50 | 0.01 | 3.51 | 39.46 | 0.19 | 0.39 |
| NSEC.Influent.Flow | 28.94 | 1.09 | 2.30 | 0.15 | 0.00 | 0.97 | 0.28 | 0.05 | 0.01 | 0.28 | 0.40 | 11.21 | 54.32 |
| NSEC.Effluent.Flow | 28.08 | 0.58 | 3.61 | 0.46 | 0.15 | 0.61 | 0.45 | 0.28 | 0.35 | 0.35 | 0.18 | 64.02 | 0.90 |
| AB.10.Influent.Flow | 30.34 | 0.32 | 0.71 | 0.41 | 0.09 | 0.33 | 0.41 | 0.18 | 0.08 | 0.45 | 0.05 | 23.11 | 43.52 |
| AB.10.RAS.Flow | 1.20 | 0.56 | 16.83 | 0.39 | 0.44 | 27.89 | 42.56 | 1.43 | 0.35 | 7.57 | 0.74 | 0.00 | 0.05 |
| AB.10.MLR.Flow | 4.81 | 6.85 | 17.77 | 0.08 | 7.16 | 1.60 | 1.04 | 20.22 | 1.74 | 35.02 | 2.39 | 0.80 | 0.53 |
| Quad.4.RAS.Flow | 3.67 | 9.47 | 15.99 | 3.88 | 0.69 | 4.48 | 15.68 | 5.88 | 3.71 | 31.92 | 4.48 | 0.06 | 0.10 |