Goodyear_WorkEx

MoWater Goodyear Team

6/23/2020

```r
library(tidyverse); theme_set(theme_minimal())
theme_update(panel.grid.minor = element_blank())
library(lubridate)
library(rcartocolor)
library(RColorBrewer)
library(viridis)
library(scales)
library(rstatix)
library(dplyr)
library(ggpubr)
library(leaps) #For Best Subset
library(plotly) #for 3D plots
library(fields)
library(here) #Optional for loading files.
#   If library not install, call it by here::here(). If installed, just here().
library(webshot) #for knitting html output into pdf
```

Authors:
Ivan Ko
Blake Loosley
Lauren Varnado

MoWater

Goodyear Artificial Wetland Project

1. Setting Important Dates

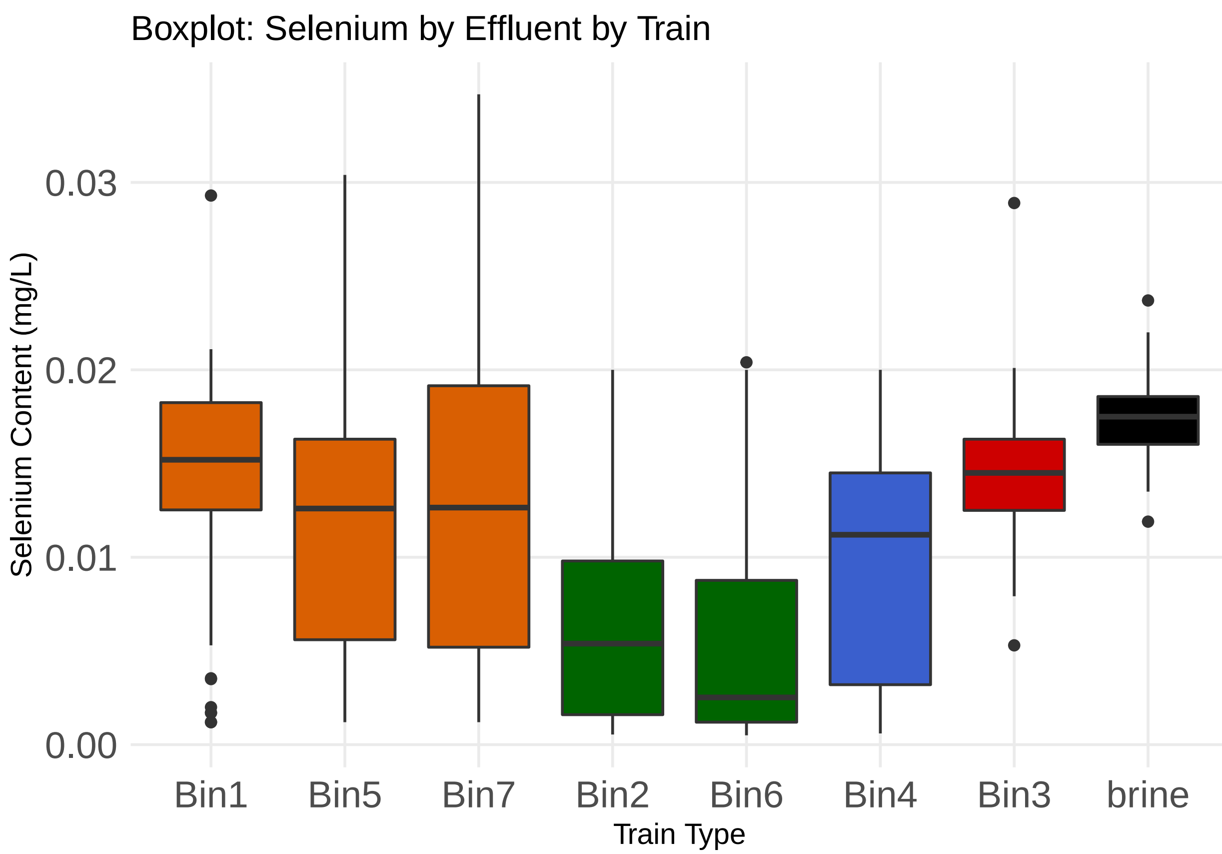
```r
#train change date: relevant for bin 2 and 4.
#Before the change, bin 2 is train 3,
#This can be our starting date since we will only be ignoring 8 months of data.
trainChangeDate <- ymd( "2011-06-15")

#unstable periods
unstablePeriodStart <- ymd( "2014-04-01")
unstablePeriodEnd <- ymd( "2016-01-01") #rough est. according to Katie (stakeholder)

#Note: 2015-04-01 may be set to 2015-01-01 because the data doesn't look right.
#There's a spike in data around Jan 2015 that should be grouped with the next
# performance period,hence this choice.

#set periods: most bins have different perfmance periods!
```

```r
#bin1, 5, 6, 7
bin1567Period1End <- ymd( "2012-03-01")
bin1567Period2End <- ymd( "2015-04-01")
bin1567Period3End <- ymd( "2017-04-01")
bin1567Periods <- c(bin1567Period1End, bin1567Period2End, bin1567Period3End)

#periods for bin2
bin2Period1End <- ymd( "2015-04-01")
bin2Period2End <- ymd( "2017-04-01")
bin2Periods <- c(bin2Period1End, bin2Period2End)

#periods for bin3, bin4
bin34Period1End <- ymd( "2015-04-01")
bin34Period2End <- ymd( "2016-12-01")
bin34Periods <- c(bin34Period1End, bin34Period2End)
```

2. Exploratory Analysis

2.1 Boxplot on Bin Selenium level

```r
#different boxplot grouped by train type
dfTest <- dfDataSel
dfTest$ID <- factor(dfTest$ID , levels=c("Bin1", "Bin5", "Bin7", "Bin2",
                                          "Bin6", "Bin4", "Bin3", "brine"))

boxSelTCGroup <- dfTest %>%
    ggplot(aes(x = ID, y = Selenium)) +
    geom_boxplot(fill = c("#D95F02", "#D95F02", "#D95F02",
                          "darkgreen", "darkgreen", "royalblue3",
                          "red3", "black")) +
    xlab("Train Type") +
    ylab("Selenium Content (mg/L)") +
    labs(title= "Boxplot: Selenium by Effluent by Train") +
    theme(legend.position = "none", axis.text=element_text(size=14))

boxSelTCGroup
```

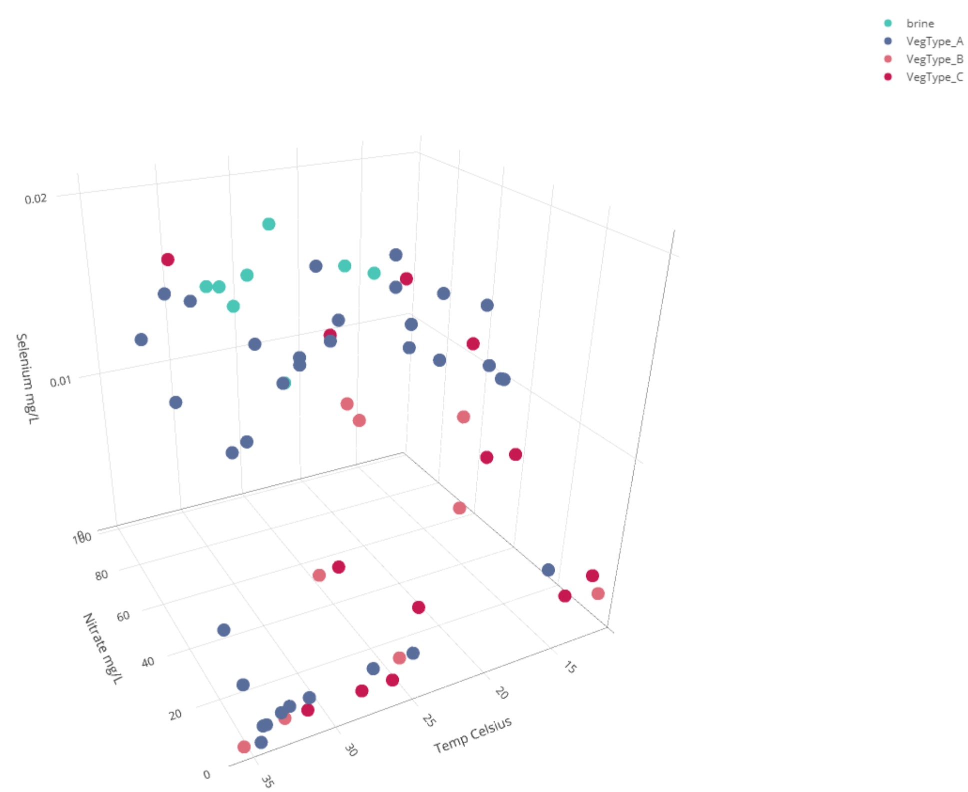


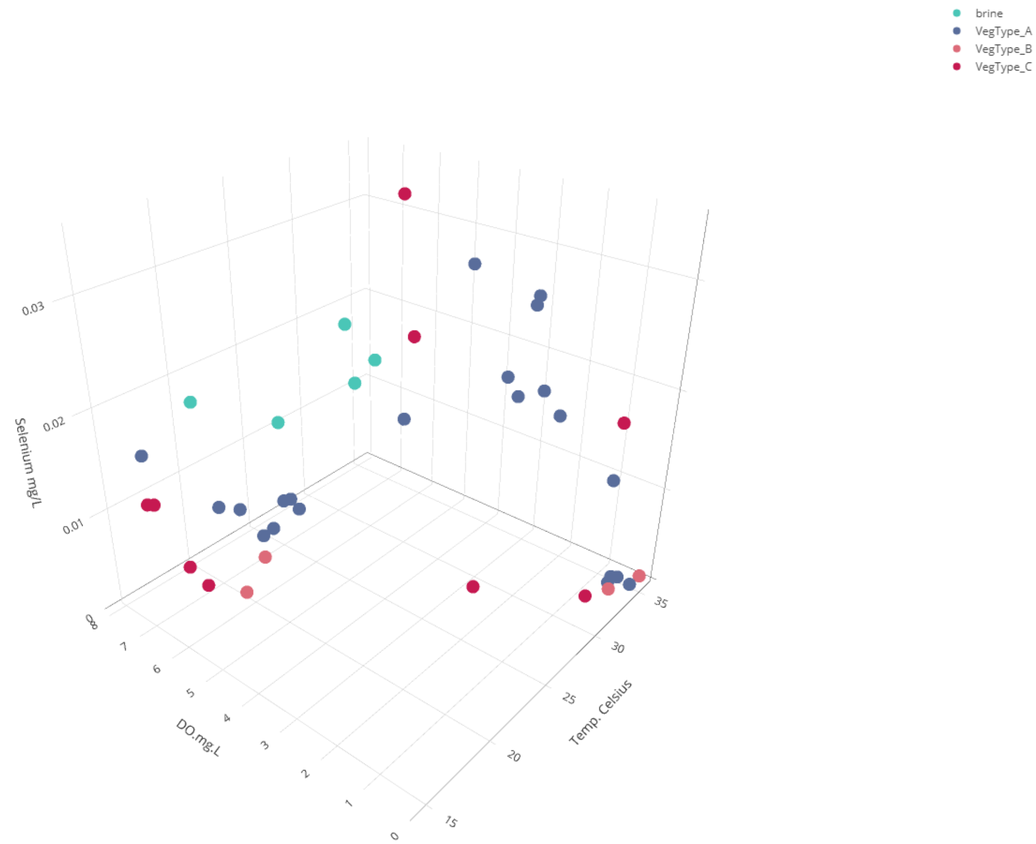
In this Boxplot, it is shown that Bin 3 and Bin 1 have a small range with most of the data occurring well above the Selenium threshold. However, there are a couple outliers that produce more successful Selenium concentrations. Additionally, Bin 2 and Bin 6 have the most consistently low Selenium concentration values compared to the other bins. In other words, Bins 2 and 6 appear to be the only bins that are skewed towards higher values whereas the other bins are skewed toward the lower values. At face value, it appears that Bins 2 and 6 seem to be the best for removing Selenium since they have the lowest medians.

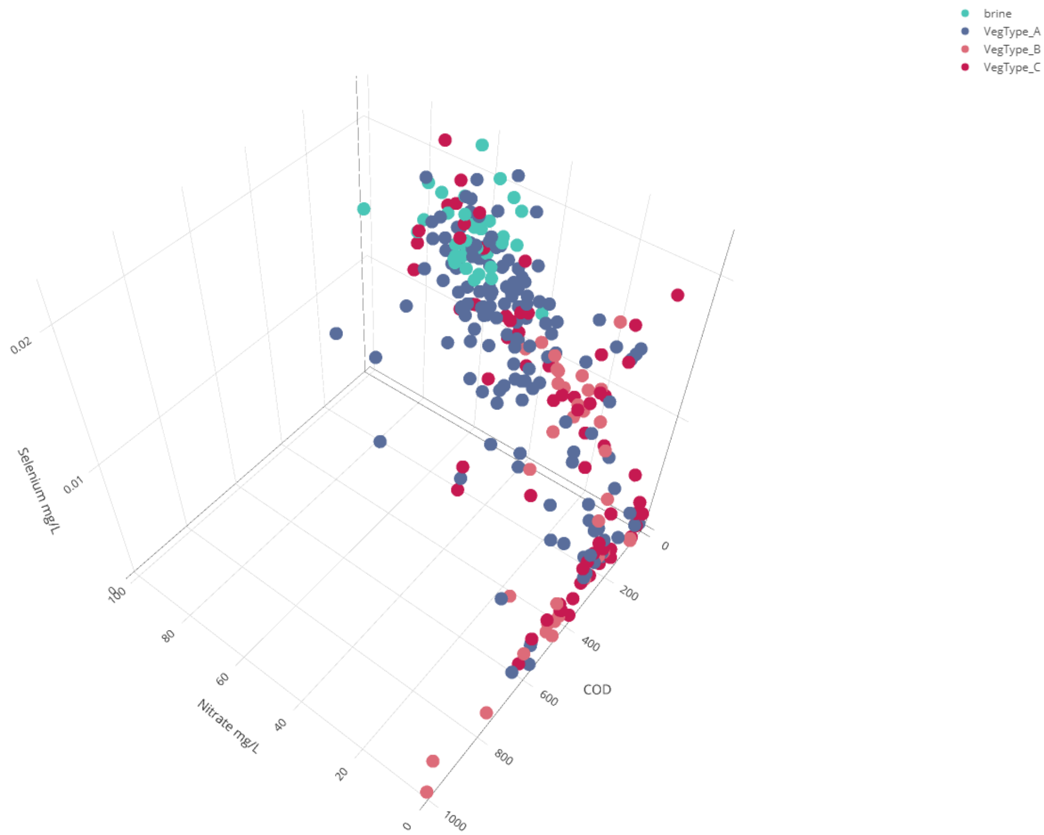
2.2 3D Plots

```r
#NOTE: This code is NOT run here when knitting due to the html output issue.

#Set color here
colorsScale <- c('#4AC6B7', '#1972A4', '#965F8A', '#FF7070', '#C61951')

#Temp vs Nit on Veg
fig <- plot_ly(dfT, x = ~Nitrate, y = ~Temp..Celsius,
               z = ~Selenium, color = ~Veg, colors = colorsScale)
fig <- fig %>% add_markers()
fig <- fig %>% layout(scene = list(xaxis = list(title = 'Nitrate mg/L'),
                                   yaxis = list(title = 'Temp Celsius'),
                                   zaxis = list(title = 'Selenium mg/L')))
fig
```

```r
#---

#Temp vs DO on Veg
figTDOV <- plot_ly(dfT, x = ~Temp..Celsius, y = ~DO.mg.L,
                   z = ~Selenium, color = ~Veg, colors = colorsScale)
figTDOV <- figTDOV %>% add_markers()
figTDOV <- figTDOV %>% layout(scene = list(xaxis = list(title = 'Temp. Celsius'),
                                           yaxis = list(title = 'DO.mg.L'),
                                           zaxis = list(title = 'Selenium mg/L')))
figTDOV

#---

#Temp vs COD
figTCV <- plot_ly(dfT, x = ~Temp..Celsius, y = ~COD,
                  z = ~Selenium, color = ~Veg, colors = colorsScale)
figTCV <- figTCV %>% add_markers()
figTCV <- figTCV %>% layout(scene = list(xaxis = list(title = 'Temp. Celsius'),
                                         yaxis = list(title = 'COD mg/L'),
                                         zaxis = list(title = 'Selenium mg/L')))
figTCV
```

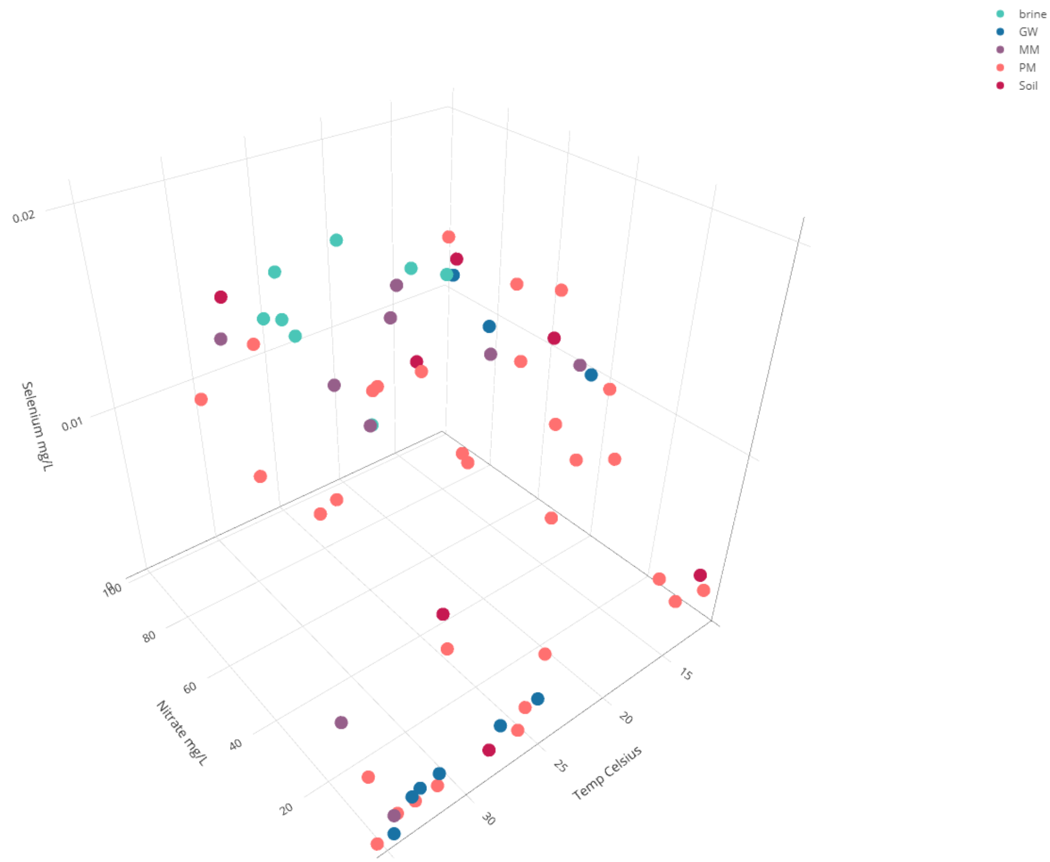


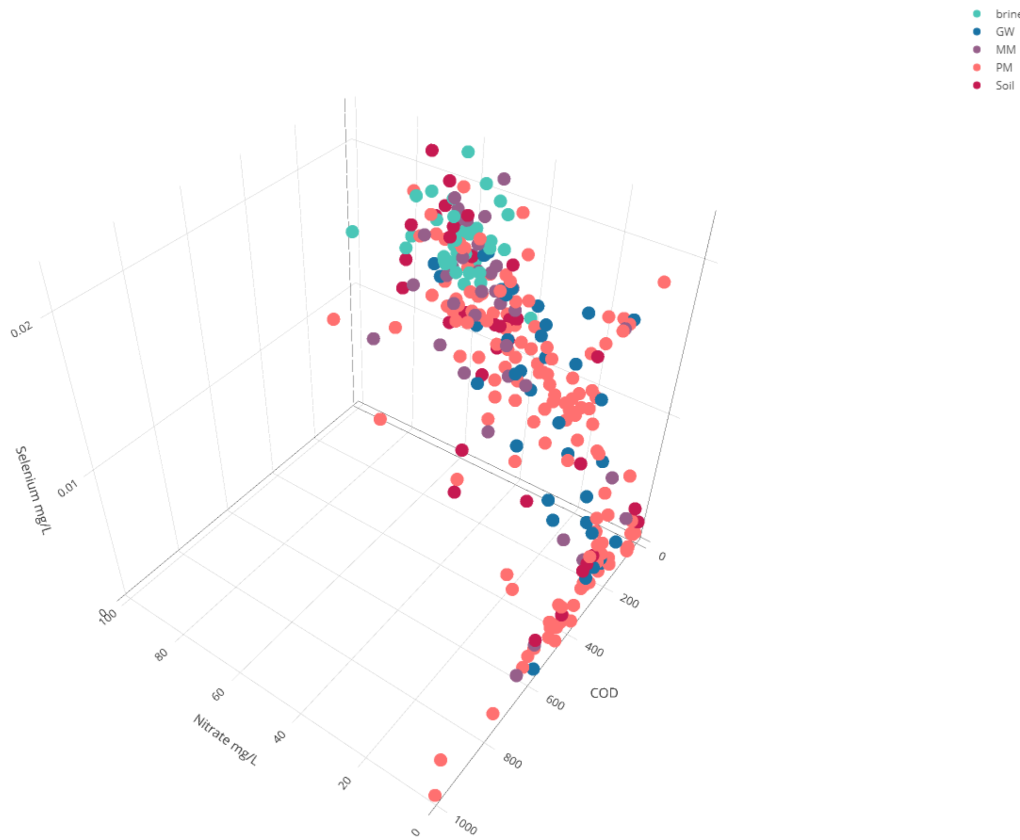

brine
VegType_A
VegType_B
VegType_C
Selenium mg/L
0.03
0.02
0.01
DO.mg.L
Temp. Celsius



When high Temperature is coupled with low Nitrate or DO, Selenium tends to be low. But COD doesn't have a clear correlation with Selenium level.

```
#Temp vs Nit on Media
figM <- plot_ly(dfT, x = ~Nitrate, y = ~Temp..Celsius,
               z = ~Selenium, color = ~MediaType, colors = colorsScale)
figM <- figM %>% add_markers()
figM <- figM %>% layout(scene = list(xaxis = list(title = 'Nitrate mg/L'),
                                     yaxis = list(title = 'Temp Celsius'),
                                     zaxis = list(title = 'Selenium mg/L')))
figM

#---

#Nit vs COD on Media
figNCM <- plot_ly(dfT, x = ~Nitrate, y = ~COD,
                 z = ~Selenium, color = ~MediaType, colors = colorsScale)
figNCM <- figNCM %>% add_markers()
figNCM <- figNCM %>% layout(scene = list(xaxis = list(title = 'Nitrate mg/L'),
                                         yaxis = list(title = 'COD'),
                                         zaxis = list(title = 'Selenium mg/L')))
figNCM
```


brine
GW
MM
PM
Soil
0.02
0.01
Selenium mg/L
Nitrate mg/L
Temp Celsius
15
20
25
30
20
40
60
80
100



In general, Media Type seems to be affected by variables in a similar way to Vegetation except for Soil type. Soil Type Media is more resistant to changes in the environment than other Media Types.

```
#Diff Nit and Temp on Veg
figNTVD <- plot_ly(dfD, x = ~Nitrate, y = ~Temp..Celsius,
                   z = ~diff_Selenium, color = ~Veg, colors = colorsScale)
figNTVD <- figNTVD %>% add_markers()
figNTVD <- figNTVD %>% layout(scene = list(xaxis = list(title = 'Nitrate mg/L'),
                                    yaxis = list(title = 'Temp Celsius'),
                                    zaxis = list(title = 'Difference Selenium mg/L')))
figNTVD

#---

#Diff Nit and Temp on Media
figNTMD <- plot_ly(dfD, x = ~Nitrate, y = ~Temp..Celsius,
                   z = ~diff_Selenium, color = ~MediaType, colors = colorsScale)
figNTMD <- figNTMD %>% add_markers()
figNTMD <- figNTMD %>% layout(scene = list(xaxis = list(title = 'Nitrate mg/L'),
                                        yaxis = list(title = 'Temp Celsius'),
                                        zaxis = list(title = 'Difference Selenium mg/L')))
figNTMD

#---
```

```
#Diff Nit vs COD on Media
figNCMD <- plot_ly(dfD, x = ~Nitrate, y = ~COD,
                   z = ~diff_Selenium, color = ~MediaType, colors = colorsScale)
figNCMD <- figNCMD %>% add_markers()
figNCMD <- figNCMD %>% layout(scene = list(xaxis = list(title = 'Nitrate mg/L'),
                                          yaxis = list(title = 'COD mg/L'),
                                          zaxis = list(title = 'Difference Selenium mg/L')))
figNCMD
```

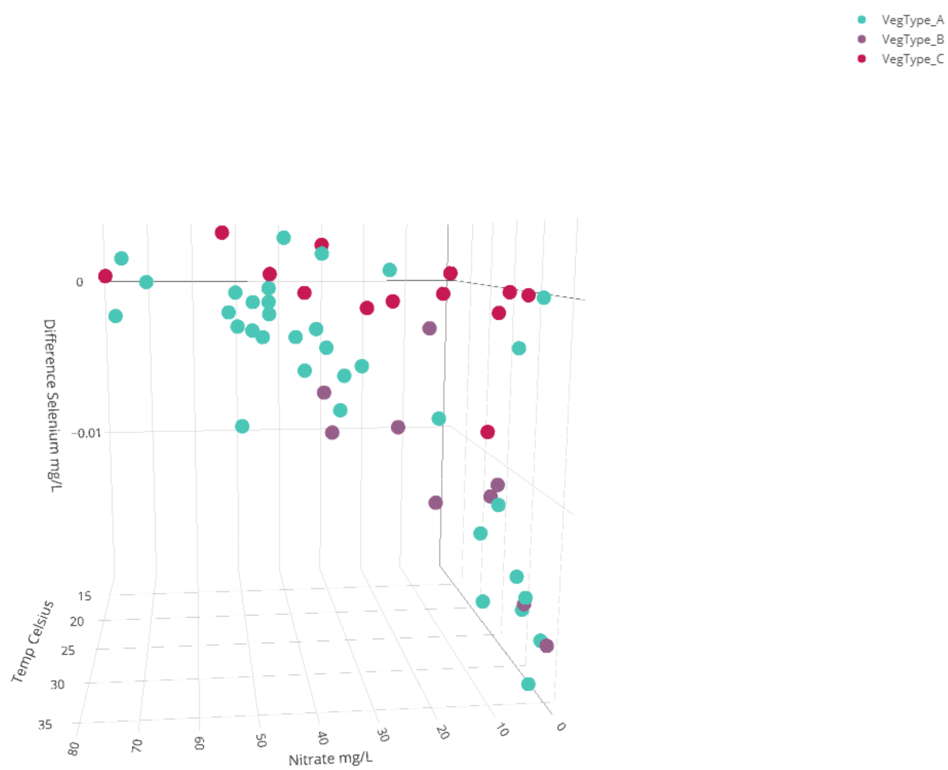


Figure 1: Diff Selenium Nit vs Temp on Veg

2.3 Best Subset Regression

```
GetLeapTable <- function(leapSummaryIn){
    result <- cbind(leapSummaryIn$adjr2, leapSummaryIn$cp, leapSummaryIn$bic)
    return(result)
}

GetMinMax <- function(leapSummaryIn){
    result <- data.frame(
        Adj.R2 = which.max(leapSummaryIn$adjr2),
        CP = which.min(leapSummaryIn$cp),
```

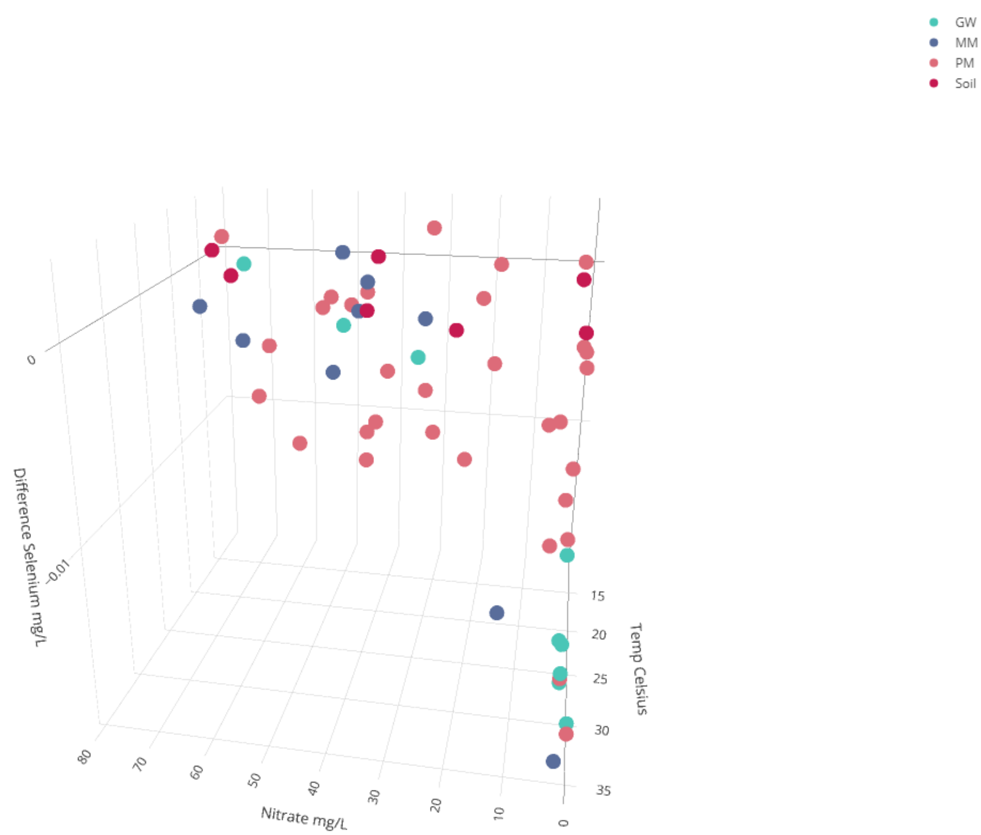


Figure 2: Diff Selenium Nit vs Temp on Media

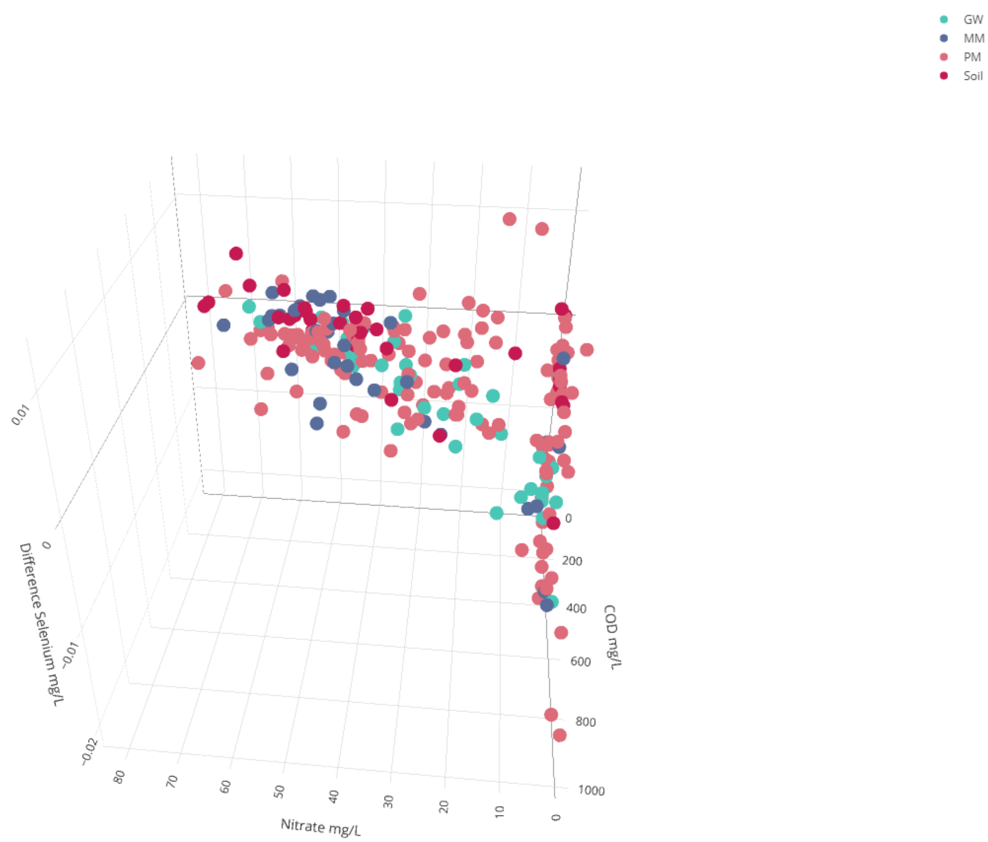


Figure 3: Diff Selenium Nit vs COD on Media

```r
        BIC = which.min(leapSummaryIn$bic)
    )
    return(result)
}

#Just using veg, no media
leapsResultVegL <- regsubsets(Selenium ~ Nitrate + COD + Phosphorus + Arsenic +
                                Veg, data = dfCLong, nvmax = 5)
# view results
leapSummaryVegL <- summary(leapsResultVegL)
leapTableVegL <- GetLeapTable(leapSummaryVegL)
minMaxLeapVegL <- GetMinMax(leapSummaryVegL)

minMaxLeapVegL
```

```
##   Adj.R2 CP BIC
## 1      5  5   3
```

```r
#Adj.R2 CP BIC
#5      5  3
#---

leapTableVegL
```

```
##           [,1]      [,2]      [,3]
## [1,] 0.5706988 23.785741 -203.8751
## [2,] 0.5790363 19.422252 -204.3136
## [3,] 0.5886022 14.326682 -205.6097
## [4,] 0.5958105 10.759677 -205.5666
## [5,] 0.6022738  7.694982 -205.1338
```

```r
# [3,] 0.589 14.327 -205.610
# [5,] 0.602  7.695 -205.134

leapSummaryVegL
```

```
## Subset selection object
## Call: regsubsets.formula(Selenium ~ Nitrate + COD + Phosphorus + Arsenic +
##     Veg, data = dfCLong, nvmax = 5)
## 7 Variables  (and intercept)
##              Forced in Forced out
## Nitrate          FALSE      FALSE
## COD              FALSE      FALSE
## Phosphorus       FALSE      FALSE
## Arsenic          FALSE      FALSE
## VegVegType_A     FALSE      FALSE
## VegVegType_B     FALSE      FALSE
## VegVegType_C     FALSE      FALSE
## 1 subsets of each size up to 5
## Selection Algorithm: exhaustive
##          Nitrate COD Phosphorus Arsenic VegVegType_A VegVegType_B VegVegType_C
## 1  ( 1 ) "*"     " " " "        " "     " "          " "          " "
## 2  ( 1 ) "*"     " " " "        "*"     " "          " "          " "
## 3  ( 1 ) "*"     "*" "*"        " "     " "          " "          " "
## 4  ( 1 ) "*"     "*" "*"        "*"     " "          " "          " "
```

```
## 5  ( 1 ) "*"     "*" "*"         "*"     " "            " "            "*"
```

```r
#Nit, COD, Phosphorus
#Nit, COD, Phosphorus, Arsenic, Veg Type C

# plot a table of models showing variables in each model.
# models are ordered by the selection statistic.
plot(leapsResultVegL, scale = "Cp",
     main = "5 Best Subsets Regression on Selenium")
```