

Data Screening

PAZ

06/04/2017

Introduction

This Data Screening notebook follows the Guide to Statistical Analysis in Microbial Ecology (GUSTA ME). The purpose is to inspect the variables that we'll be using to test for hypotheses later on, and check whether they follow typical assumptions made in parametric tests such as normality, freedom from heteroskedasticity (difference in variability btw. two+ variables) and outliers.

Reference:

<https://sites.google.com/site/mb3gustame/home> Buttigieg PL, Ramette A (2014) A Guide to Statistical Analysis in Microbial Ecology: a community-focused, living review of multivariate data analyses. FEMS Microbiol Ecol. 90: 543-550.

Packages

```
library(sm)
library(vioplot)

library(dplyr)
library(tidyr)
library(zoo)
library(reshape)
library(ggplot2)
library("ggrepel")

library("plotly")
library("cowplot")
library("gridExtra")
library("Cairo")
library("GGally")
library("scales")

# Stats
library("vegan")
library("cluster")

# Saving a xls file
# library(xlsx)
```

Lab parameters

```
# Initial signature measured in tank
initialDelta = d13Co = -31.2144
```

```

# Define initial concentration (for Raleigh plots)
#Co <- 8 # ug/g dry soil (based on Corn applications)
Co <- 6.53 # ug/g dry soil (based on Max conc. measured in soils)
# Note: Each transect now has individual starting concentration

epsilon_lab = -1.75

```

Missing values

1. Missing chemical and isotope data due to machine failure or automatic sampling servicing program.

These have been considered to be Values Missing Completely at Random (MCAR) as they are associated to the end of the automatic sampler's capacity for a certain number of events where servicing was inadequate for the discharge amounts seen during a sampling week. Here the values' missingness is not related to any other value in the data set.

2. Isotope data for both soil and water samples due to concentration value being below the limit of detection.

These values must be considered to be Missing at Random (MAR) as the missing value has no relation to the value that 'should' be there, but does depend on other variables in the data set. Thus, other variables must be taken into account for MAR data to be considered random (i.e. missing data is "conditioned by" other data in the data set).

Import soils

Convert to single time observation for merging with water observation.

```

# Soils
soils = read.csv2("Data/MassBalance_R.csv",
                 na.strings=c('#DIV/0!', '', 'NA'), header = TRUE)
colnames(soils)[colnames(soils) == "ti"] <- "Date.ti"
soils$Date.ti <- as.POSIXct(strptime(soils$Date.ti,
                                     "%Y-%m-%d %H:%M", tz="EST")) # csv typos, option 1
sum(is.na(soils$Date.ti)) == 0

## [1] TRUE
initialDelta

## [1] -31.2144

# Get rid of imputed values to avoid unwanted bias
soils$DD13C.North <- (ifelse(!is.na(soils$comp.d13C.SD.North), soils$comp.d13C.North - (initialDelta), 1
soils$DD13C.Talweg <- (ifelse(!is.na(soils$comp.d13C.SD.Talweg), soils$comp.d13C.Talweg - (initialDelta), 1
soils$DD13C.South <- (ifelse(!is.na(soils$comp.d13C.SD.South), soils$comp.d13C.South - (initialDelta), 1

dropSoil <- c("WeekSubWeek", # "Event",
             "B.diss", "B.filt", "CumOutDiss.g", "CumOutFilt.g", "CumOutAppMass.g", "CumOutMELsm.g",
             # "CumAppMass.g",
             "ID.N", "ID.T", "Area.N", "Area.T", "Area.S",
             "comp.d13C.SE.North", "comp.d13C.SE.Talweg", "comp.d13C.SE.South",
             "f.max.comp", "f.mean.comp", "f.min.comp", "ngC.SD", "ngC.SE", "N_compsoil", "N_ngC")

```

```

soils <- soils[ , !(names(soils) %in% dropSoil)]

timeApps <- soils[ , c("Date.ti", "timeSinceApp", "Event")]

# Quasi-Molten SOILS
soilGroups = read.csv2("Data/WeeklySoils_Rng.csv",
                      na.strings=c('#DIV/0!', '', 'NA'), header = TRUE)
soilGroups$Date.ti <- as.POSIXct(strptime(soilGroups$Date.ti,
                                         "%Y-%m-%d %H:%M", tz="EST")) # csv typos, option 1
sum(is.na(soilGroups$Date.ti)) == 0

## [1] TRUE

soilGroups$comp.d13C <- ifelse(is.na(soilGroups$comp.d13C.SD), NA, soilGroups$comp.d13C)
soilGroups$ngC.Label <- ifelse(soilGroups$ngC.mean < 10, "< 10 ng", "> 10 ng")

#str(soils)

soilGrApp <- merge(soilGroups, timeApps, by = "Date.ti", all = F)
soilGrApp <- soilGrApp[complete.cases(soilGrApp[ , "timeSinceApp"]),]

soilGrApp$DD13C.comp <- ifelse(is.na(soilGrApp$comp.d13C.SD), NA, soilGrApp$DD13C.comp)

cor.test(soilGroups$comp.d13C, soilGroups$Conc.mug.g.dry.soil)

##
## Pearson's product-moment correlation
##
## data: soilGroups$comp.d13C and soilGroups$Conc.mug.g.dry.soil
## t = -3.221, df = 32, p-value = 0.002931
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.7135739 -0.1881206
## sample estimates:
## cor
## -0.4948086

pearson_r <- cor.test(soilGroups$comp.d13C, soilGroups$Conc.mug.g.dry.soil)[4]
r_label <- sprintf("Pearson-r == %0.2f", pearson_r)
p_value <- cor.test(soilGroups$comp.d13C, soilGroups$Conc.mug.g.dry.soil)[3]

if (p_value < 0.0001){
  p_label <- "(p < 0.0001)"
} else if (p_value < 0.001) {
  p_label <- "(p < 0.001)"
} else if (p_value < 0.015) {
  p_label <- "(p < 0.01)"
} else {
  p_label <- "Check significance"
}

soilGrApp$Source <- ifelse(soilGrApp$Transect == "T", "Valley", "Plateau")
soilGrApp$Source <- as.factor(soilGrApp$Source)

p <- ggplot(data = subset(soilGrApp, !is.na(ngC.Label)), aes(x=Conc.mug.g.dry.soil, y=DD13C.comp))+

```

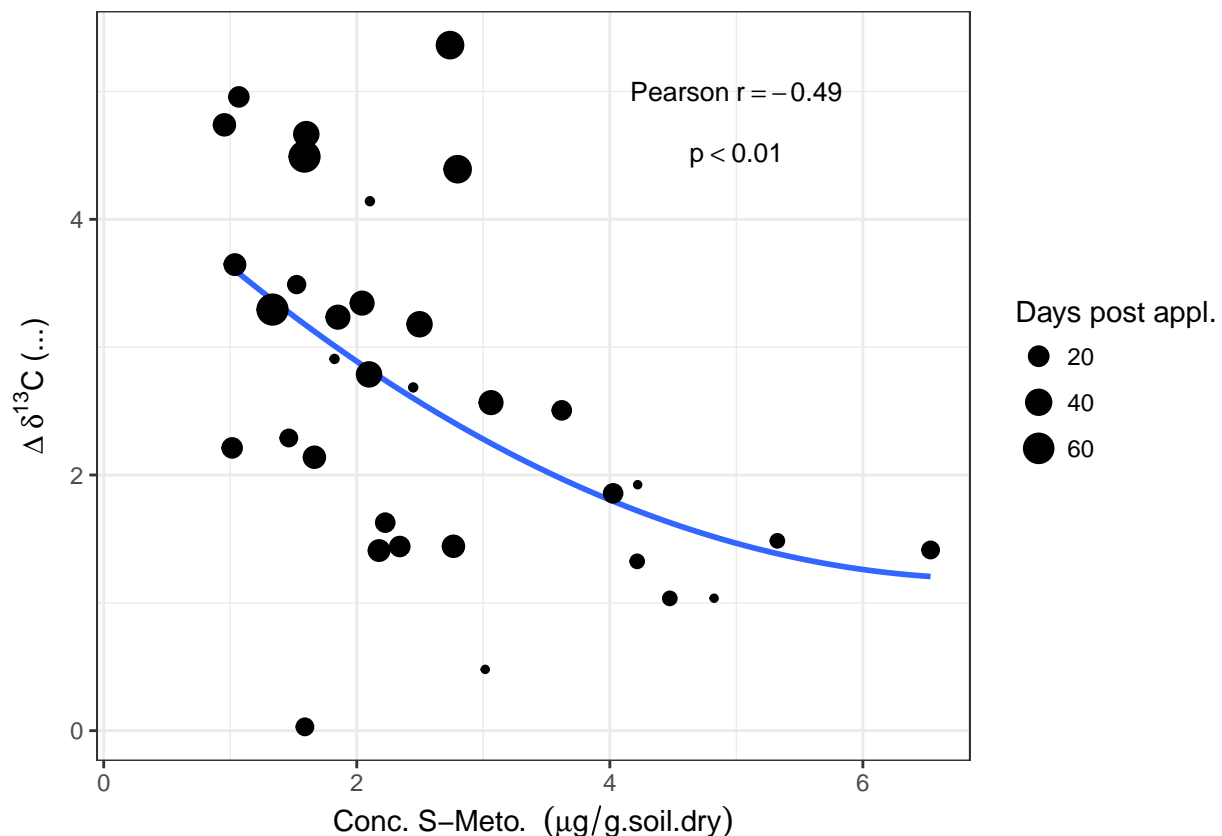
```

stat_smooth(data = subset(soilGrApp, !is.na(ngC.Label)),
            aes(x=Conc.mug.g.dry.soil, y=DD13C.comp), method = "lm", formula = y ~ poly(x, 2), se=F) +
geom_point(aes(group = ID, size = timeSinceApp)) + # , colour = Source)) + # , shape = ngC.Label)) +
theme_bw() +
labs(size="Days post appl.", colour="Source" ) + # , shape = "Mass Carbon") +
ylab(expression(paste({\Delta}^{13}\text{C}, "C", ' (\u2030)')) +
xlab(expression(paste("Conc. S-Meto. ", {\mu}\text{g} / \text{g.soil.dry}))) +
annotate("text", x = 5, y = 5, label = as.character(r_label), parse = T, size = 3.5) +
annotate("text", x = 5, y = 4.5, label = p_label, parse = T, size = 3.5) +
scale_size_continuous(range = c(1, 5))

#scale_color_hue("Group") +
#scale_fill_manual(
# "CI horizontal line", values=rep(1,4),
# guide=guide_legend(override.aes = list(colour=c("orange", "darkred"))),
# labels=c("CI of 95%", "CI of 99%")
#)
#geom_text_repel(data = subset(soilGrApp, (!is.na(ngC.Label) & Wnum > 10) ), aes(label=Wnum),
#               arrow = arrow(length = unit(0.005, 'npc'), type = "closed"),
#               force = 1,
#               point.padding = unit(1.0, 'lines'),
#               max.iter = 2e3,
#               nudge_x = .2)

```

p



```

#ggsave(p, filename = "DDvsConc_soils.png", width = 8, height = 5, units = "in", scale = 1)

```

```
#ggplotly(p)
#stat_smooth(method = "lm", formula = y ~ poly(x, 2)) +
#stat_smooth(method = "lm", formula = y~x, se=F)
```

Raleigh plot

$$\ln\left(\frac{1000 + \delta^{13}C_0 + \Delta\delta^{13}C}{1000 + \delta^{13}C_0}\right) = (\alpha - 1) \cdot \ln f = \frac{\epsilon}{1000} \cdot \ln f$$

$$f = \frac{C_t}{C_0}$$

```
soilGrApp$iniCo <- ifelse(soilGrApp$Transect == "N", 4.82,
                          ifelse(soilGrApp$Transect == "T", 5.32,
                                ifelse(soilGrApp$Transect == "S", 6.53, NA)))
#soilGrApp$iniCo <- 8

soilGrApp$yRaleigh <- log((1000+d13Co+soilGrApp$DD13C.comp)/(1000+d13Co))
soilGrApp$xRaleigh <- log(soilGrApp$Conc.mug.g.dry.soil/soilGrApp$iniCo)

# model<-lm(yRaleigh~xRaleigh, data= soilGrApp, subset=(Wnum < 12 & !is.na(ngC.Label)))
model<-lm(yRaleigh~xRaleigh, data= soilGrApp, subset=(!is.na(ngC.Label))) # & ngC.mean >= 5))
cof <- as.numeric(coef(model)[2]*1000)
se <- summary(model)$coef[[4]]*1000
lab <- sprintf(" epsilon == %0.2f ", cof)
labSE <- sprintf("\u00B1 %0.2f ", se)
labSE2 <- sprintf("± %0.2f ", se)

labSE3 <- paste(" ' ' %+-% ' 0.43' ")
lab1 <- paste(lab, labSE3)

summary(model)

##
## Call:
## lm(formula = yRaleigh ~ xRaleigh, data = soilGrApp, subset = (!is.na(ngC.Label)))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.019e-03 -8.602e-04  6.753e-05  7.700e-04  3.141e-03
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0013771  0.0004145   3.323 0.002241 **
## xRaleigh     -0.0015079  0.0004064  -3.710 0.000785 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.001196 on 32 degrees of freedom
## (3 observations deleted due to missingness)
## Multiple R-squared:  0.3008, Adjusted R-squared:  0.2789
## F-statistic: 13.76 on 1 and 32 DF, p-value: 0.0007848
```

```

# Compre to each transect
modelTalweg<-lm(yRaleigh~xRaleigh, data=soilGrApp, subset=(Wnum < 12 & !is.na(ngC.Label) & Transect == "eT" & !is.na(yRaleigh)))
eT <- coef(modelTalweg)[2]*1000

modelNorth<-lm(yRaleigh~xRaleigh, data=soilGrApp, subset=(Wnum < 12 & !is.na(ngC.Label) & Transect == "eN" & !is.na(yRaleigh)))
eN <- coef(modelNorth)[2]*1000

modelSouth<-lm(yRaleigh~xRaleigh, data=soilGrApp, subset=(Wnum < 12 & !is.na(ngC.Label) & Transect == "eS" & !is.na(yRaleigh)))
eS <- coef(modelSouth)[2]*1000

sd(c(coef(modelSouth)[2]*1000 , coef(modelNorth)[2]*1000 , coef(modelTalweg)[2]*1000))

## [1] 0.6369918

mean(c(coef(modelSouth)[2]*1000 , coef(modelNorth)[2]*1000 , coef(modelTalweg)[2]*1000))

## [1] -1.272724

#modelFull<-lm(yRaleigh~xRaleigh, data=soilGroups, subset=(Wnum < 16))
#summary(modelFull)

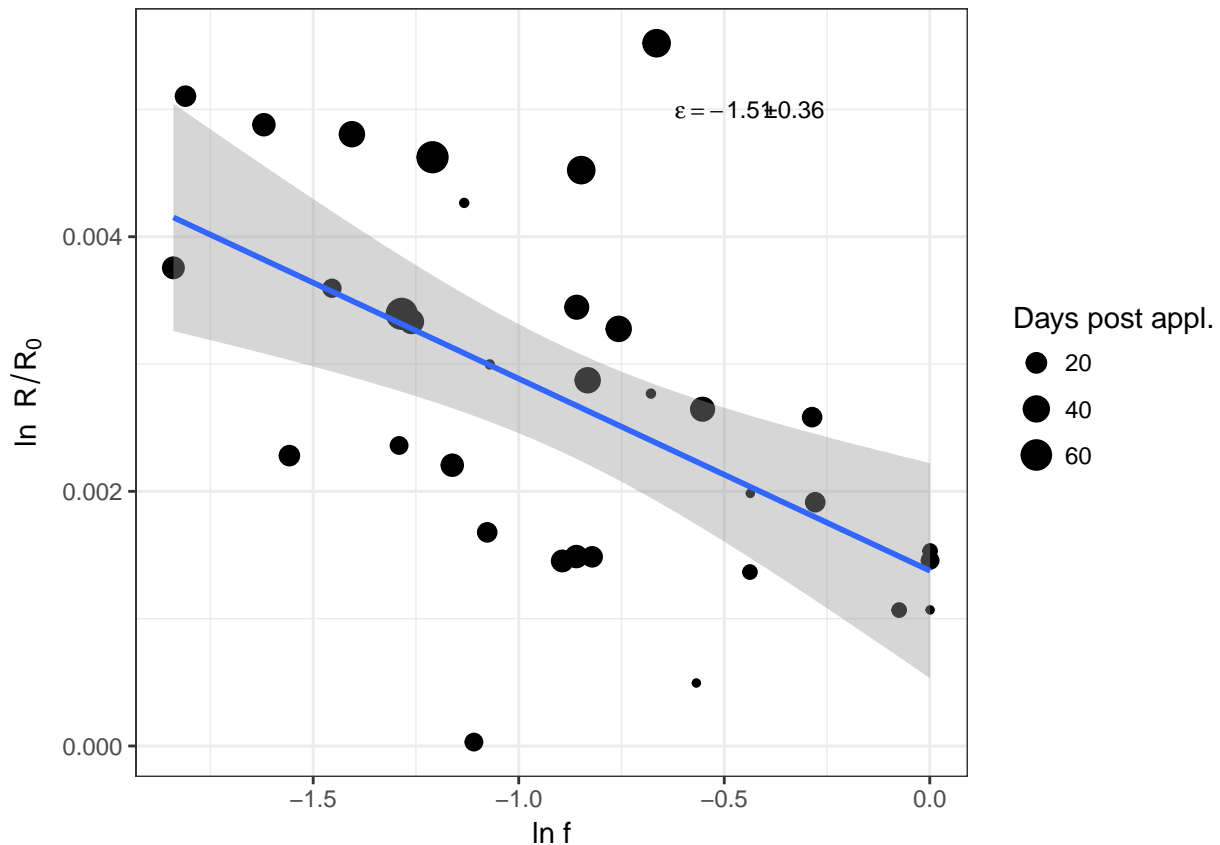
raleigh <- ggplot(data = subset(soilGrApp, ( Wnum > 0 & !is.na(ngC.Label) & !is.na(yRaleigh) )), aes(x=
  geom_point(aes(group = ID, size = timeSinceApp ))+ #, colour = Source)) + #, shape = ngC.Label)) +
  theme_bw() +
  scale_size_continuous(range = c(1, 5)) +
  labs(size="Days post appl.", colour="Source") + #, shape = "Mass Carbon") +
  xlab("ln f") +
  ylab("ln R/Ro") +
  ylab(expression(paste("ln ", R / R['0'] ))) +
  stat_smooth(data= subset(soilGrApp , ( Wnum > 0 & !is.na(ngC.Label) & !is.na(yRaleigh) )) , method = "lm") +
  annotate("text", x = -0.5, y = 0.005,
    # label = as.character(expression(paste( "\u0190", " \u2030", " = ", cof))), parse = T, size = 3.0) +
    label = lab, parse = T, size = 3.0) +
  annotate("text", x = -0.33, y = 0.005,
    label = as.character(expression(paste( "\u00B1" , 0.36))), parse = T, size = 3.0) # +

#geom_text_repel(data = subset(soilGrApp, (!is.na(ngC.Label) & Wnum > 7 & Wnum < 12) ), aes(label=Wnum),
#
#   arrow = arrow(length = unit(0.005, 'npc'), type = "closed"),
#
#   force = 1,
#
#   point.padding = unit(1.0, 'lines'),
#
#   max.iter = 2e3,
#
#   nudge_x = .2)

#geom_text_repel(aes(label=Wnum),
#
#   arrow = arrow(length = unit(0.005, 'npc'), type = "closed"),
#
#   force = 1,
#
#   point.padding = unit(1.0, 'lines'),
#
#   max.iter = 2e3,
#
#   nudge_x = .2)

raleigh

```



```
#ggplotly(raleigh)
# ggsave(raleigh, filename = "lnDDvslnConc_soils.png", width = 8, height = 5, units = "in", scale = 1)
```

Accounting for dilution

Following (Van Breukelen 2007)

Import waters

Compare mass balance, theoretical and CSIA

```
waters = read.csv2("Data/WeeklyHydroContam_R.csv")
waters$ti <- as.POSIXct(strptime(waters$ti, "%Y-%m-%d %H:%M", tz="EST"))
colnames(waters)[colnames(waters) == "ti"] <- "Date.ti"
waters$Events <- factor(waters$Events, levels = unique(waters$Events))
waters$Event <- factor(waters$Event, levels = unique(waters$Event))

dropWater <- c("N.x", "N.y",
               "Markers", "TimeDiff",
               "se.d13C", "MES.mg.L", "MES.sd", "MO.mg.L", "filt.se.d13C", "f.diss", "f.filt",
               # "Appl.Mass.g",
               "DissSmeto.mg", "DissSmeto.mg.SD",
               "DissOXA.mg", "DissOXA.mg.SD",
```

```

      "DissESA.mg", "DissESA.mg.SD",
      "FiltSmeto.mg", "DissSmeto.mg.SD",
      "TotSMout.mg", "TotSMout.mg.SD",
      "FracDiss", "FracFilt")
waters <- waters[ , !(names(waters) %in% dropWater)]

# Half-life calculations (days)
median_half <- 29
max_half <- 12
min_half <- 46

waters$No_First <- ifelse(waters$Appl.Mass.g == 0, NA, waters$Appl.Mass.g)
waters$No_Second <- waters$No_First
waters$No_Second[1] <- 0
waters$No_Third <- waters$No_Second
waters$No_Second[which(!is.na(waters$No_Second))[3]] <- NA
waters$No_Third[which(!is.na(waters$No_Second))[2]] <- NA

waters$No_First <- na.locf( waters$No_First )
waters$No_Second <- na.locf(waters$No_Second)
waters$No_Third <- na.locf(waters$No_Third)

# Compute cumulative time for first, second and third applications
waters$CumDays_First <- cumsum(waters$Duration)/24

waters$dt_Second <- ifelse(waters$No_Second == 0, 0, waters$Duration)
waters$CumDays_Second <- cumsum(waters$dt_Second)/24
waters$dt_Second <- NULL

waters$dt_Third <- ifelse(waters$No_Third == 0, 0 , waters$Duration)
waters$CumDays_Third <- cumsum(waters$dt_Third)/24
waters$dt_Third <- NULL

waters$remain_1st_29d <- waters$No_First*(0.5)^(waters$CumDays_First/median_half)
waters$remain_2nd_29d <- waters$No_Second*(0.5)^(waters$CumDays_Second/median_half)
waters$remain_3rd_29d <- waters$No_Third*(0.5)^(waters$CumDays_Third/median_half)

waters$remain_1st_46d <- waters$No_First*(0.5)^(waters$CumDays_First/min_half)
waters$remain_2nd_46d <- waters$No_Second*(0.5)^(waters$CumDays_Second/min_half)
waters$remain_3rd_46d <- waters$No_Third*(0.5)^(waters$CumDays_Third/min_half)

waters$remain_1st_12d <- waters$No_First*(0.5)^(waters$CumDays_First/max_half)
waters$remain_2nd_12d <- waters$No_Second*(0.5)^(waters$CumDays_Second/max_half)
waters$remain_3rd_12d <- waters$No_Third*(0.5)^(waters$CumDays_Third/max_half)

waters$remainMedTheo_prc <- ((waters$remain_1st_29d + waters$remain_2nd_29d + waters$remain_3rd_29d)/wa
waters$remainMinTheo_prc <- ((waters$remain_1st_46d + waters$remain_2nd_46d + waters$remain_3rd_46d)/wa
waters$remainMaxTheo_prc <- ((waters$remain_1st_12d + waters$remain_2nd_12d + waters$remain_3rd_12d)/wa

colnames(waters)

## [1] "Date.ti"          "WeekSubWeek"
## [3] "tf"              "iflux"

```



```

## [5] "fflux" "changefflux"
## [7] "maxQ" "minQ"
## [9] "dryHrs" "Duration.Hrs"
## [11] "chExtreme" "Peak"
## [13] "AveDischarge.m3.h" "Volume.m3"
## [15] "Sampled.Hrs" "Sampled"
## [17] "Conc.mug.L" "Conc.SD"
## [19] "OXA_mean" "OXA_SD"
## [21] "ESA_mean" "ESA_SD"
## [23] "diss.d13C" "SD.d13C"
## [25] "N_ngC.diss" "ngC.mean.diss"
## [27] "ngC.SD.diss" "Conc.Solids.mug.gMES"
## [29] "Conc.Solids.ug.gMES.SD" "filt.d13C"
## [31] "filt.SD.d13C" "N_ngC.fl"
## [33] "ngC.mean.fl" "ngC.SD.fl"
## [35] "DD13C.diss" "DD13C.filt"
## [37] "B.diss" "B.filt"
## [39] "NH4.mM" "TIC.ppm.filt"
## [41] "Cl.mM" "NO3...mM"
## [43] "PO4..mM" "NPOC.ppm"
## [45] "TIC.ppm.unfilt" "TOC.ppm.unfilt"
## [47] "ExpMES.Kg" "Appl.Mass.g"
## [49] "timeSinceApp" "CumAppMass.g"
## [51] "DissSmeto.g" "DissSmeto.g.SD"
## [53] "DissOXA.g" "DissOXA.g.SD"
## [55] "DissESA.g" "DissESA.g.SD"
## [57] "FiltSmeto.mg.SD" "FiltSmeto.g"
## [59] "FiltSmeto.g.SD" "TotSMout.g"
## [61] "TotSMout.g.SD" "MELsm.g"
## [63] "MELsm.g.SD" "CumOutDiss.g"
## [65] "CumOutFilt.g" "CumOutSmeto.g"
## [67] "CumOutMELsm.g" "BalMassDisch.g"
## [69] "prctMassOut" "FracDeltaOut"
## [71] "Events" "Weeks"
## [73] "Event" "No_First"
## [75] "No_Second" "No_Third"
## [77] "CumDays_First" "CumDays_Second"
## [79] "CumDays_Third" "remain_1st_29d"
## [81] "remain_2nd_29d" "remain_3rd_29d"
## [83] "remain_1st_46d" "remain_2nd_46d"
## [85] "remain_3rd_46d" "remain_1st_12d"
## [87] "remain_2nd_12d" "remain_3rd_12d"
## [89] "remainMedTheo_prc" "remainMinTheo_prc"
## [91] "remainMaxTheo_prc"

dropWater2 <- c("No_First", "No_Second", "No_Third",
               "CumDays_First", "CumDays_Second", "CumDays_Third",
               "remain_1st_29d", "remain_2nd_29d", "remain_3rd_29d",
               "remain_1st_46d", "remain_2nd_46d", "remain_3rd_46d",
               "remain_1st_12d", "remain_2nd_12d", "remain_3rd_12d")

waters <- waters[ , !(names(waters) %in% dropWater2)]

keepWaterMB <- c("Date.ti", "CumAppMass.g",

```

```

# MB
"CumOutSmeto.g", "CumOutMELsm.g" ,
"remainMedTheo_prc", "remainMinTheo_prc", "remainMaxTheo_prc",
# CSIA
"B.diss", "SD.d13C")

# Get last 5 rows, omit NA's, will return rows only where B.diss was not NA
watersMassBal <- na.omit(waters[ , (names(waters) %in% keepWaterMB)])
watersMassBal <- subset(watersMassBal, SD.d13C < 1)

watersMassBal$SMout_prc <- (watersMassBal$CumOutSmeto.g/watersMassBal$CumAppMass.g)*100
watersMassBal$TPout_prc <- (watersMassBal$CumOutMELsm.g/watersMassBal$CumAppMass.g)*100
watersMassBal$f <- 100 - (watersMassBal$B.diss + watersMassBal$SMout_prc)
watersMassBal$SD.d13C <- NULL

mayBal <- subset(watersMassBal, (Date.ti > as.POSIXct("2016-04-01 00:00:00", tz = "EST")
& Date.ti < as.POSIXct("2016-05-01 00:00:00", tz = "EST")))
juneBal <- subset(watersMassBal, (Date.ti > as.POSIXct("2016-06-07 00:00:00", tz = "EST")
& Date.ti <= as.POSIXct("2016-06-24 14:52:00", tz = "EST")))

B.mean.may <- mean(mayBal$B.diss)
B.sd.may <- sd(mayBal$B.diss)
f.mean.may <- mean(mayBal$f)
f.sd.may <- sd(mayBal$f)

B.mean.june <- mean(juneBal$B.diss)
B.sd.june <- sd(juneBal$B.diss)
f.mean.june <- mean(juneBal$f)
f.sd.june <- sd(juneBal$f)

mayBal <- tail(mayBal, n=1)
mayBal$B.mean <- B.mean.may
mayBal$B.sd1 <- B.mean.may-B.sd.may
mayBal$B.sd2 <- B.mean.may+B.sd.may
mayBal$f.mean <- f.mean.may
mayBal$f.sd1 <- f.mean.may-f.sd.may
mayBal$f.sd2 <- f.mean.may+f.sd.may
mayBal$DegMed <- 100 - mayBal$remainMedTheo_prc
mayBal$DegLow <- 100 - mayBal$remainMinTheo_prc
mayBal$DegHigh <- 100 - mayBal$remainMaxTheo_prc
mayBal$Month <- "April"

juneBal <- tail(juneBal, n=1)
juneBal$B.mean <- B.mean.june
juneBal$B.sd1 <- B.mean.june-B.sd.june
juneBal$B.sd2 <- B.mean.june+B.sd.june
juneBal$f.mean <- f.mean.june
juneBal$f.sd1 <- f.mean.june-f.sd.june
juneBal$f.sd2 <- f.mean.june+f.sd.june
juneBal$DegMed <- 100 - juneBal$remainMedTheo_prc
juneBal$DegLow <- 100 - juneBal$remainMinTheo_prc
juneBal$DegHigh <- 100 - juneBal$remainMaxTheo_prc

```

```

juneBal$Month <- "June"

bal <- rbind(mayBal, juneBal)
bal$SMout.SD1 <- NA
bal$SMout.SD2 <- NA
bal$TPout.SD1 <- NA
bal$TPout.SD2 <- NA

bal$B.diss <- NULL
bal$f <- NULL
bal$Date.ti <- NULL
bal$CumAppMass.g <- NULL
bal$CumOutSmeto.g <- NULL
bal$CumOutMELsm.g <- NULL

names(bal)

## [1] "remainMedTheo_prc" "remainMinTheo_prc" "remainMaxTheo_prc"
## [4] "SMout_prc"          "TPout_prc"          "B.mean"
## [7] "B.sd1"              "B.sd2"              "f.mean"
## [10] "f.sd1"              "f.sd2"              "DegMed"
## [13] "DegLow"             "DegHigh"            "Month"
## [16] "SMout.SD1"          "SMout.SD2"          "TPout.SD1"
## [19] "TPout.SD2"

bal <- bal[c("Month",
             "B.mean" , "B.sd1" , "B.sd2",
             "f.mean" , "f.sd1" , "f.sd2",
             "DegMed", "DegLow", "DegHigh",
             "remainMedTheo_prc", "remainMinTheo_prc", "remainMaxTheo_prc",
             "SMout_prc", "SMout.SD1", "SMout.SD2",
             "TPout_prc", "TPout.SD1", "TPout.SD2")]

names(bal) <- c("Month",
               "B.measure" , "B.SD1" , "B.SD2",
               "f.measure" , "f.SD1" , "f.SD2",
               "Deg.measure", "Deg.SD1", "Deg.SD2",
               "Rem.measure", "Rem.SD1", "Rem.SD2",
               "SMout.measure", "SMout.SD1", "SMout.SD2",
               "TPout.measure", "TPout.SD1", "TPout.SD2")

balTidy <- bal %>%
  gather(measure, value, -Month) %>% # Melts data frame
  separate(measure, into = c("Sink", "temporary_var")) %>% # parses the sep = "." into...
  spread(temporary_var, value) # Moves molten temporary variable to own column

type <- rep(c("CSIA", "DT50", "CSIA", "DT50", "MB", "MB"), 2)

balTidyType <- cbind(balTidy, type)

balTidyType$Sink <- as.factor(balTidyType$Sink)
balTidyType$Month <- as.factor(balTidyType$Month)

```

```

balTidyType$Sink <- factor(balTidyType$Sink, levels = c("B" , "TPout", "Deg" ,"f" ,"SMout", "Rem"))
levels(balTidyType$type)

## [1] "CSIA" "DT50" "MB"

balTidyType$type <- factor(balTidyType$type, levels = c("CSIA" , "MB", "DT50"))
levels(balTidyType$Sink)

## [1] "B"      "TPout" "Deg"    "f"      "SMout" "Rem"

levels(balTidyType$Month)

## [1] "April" "June"

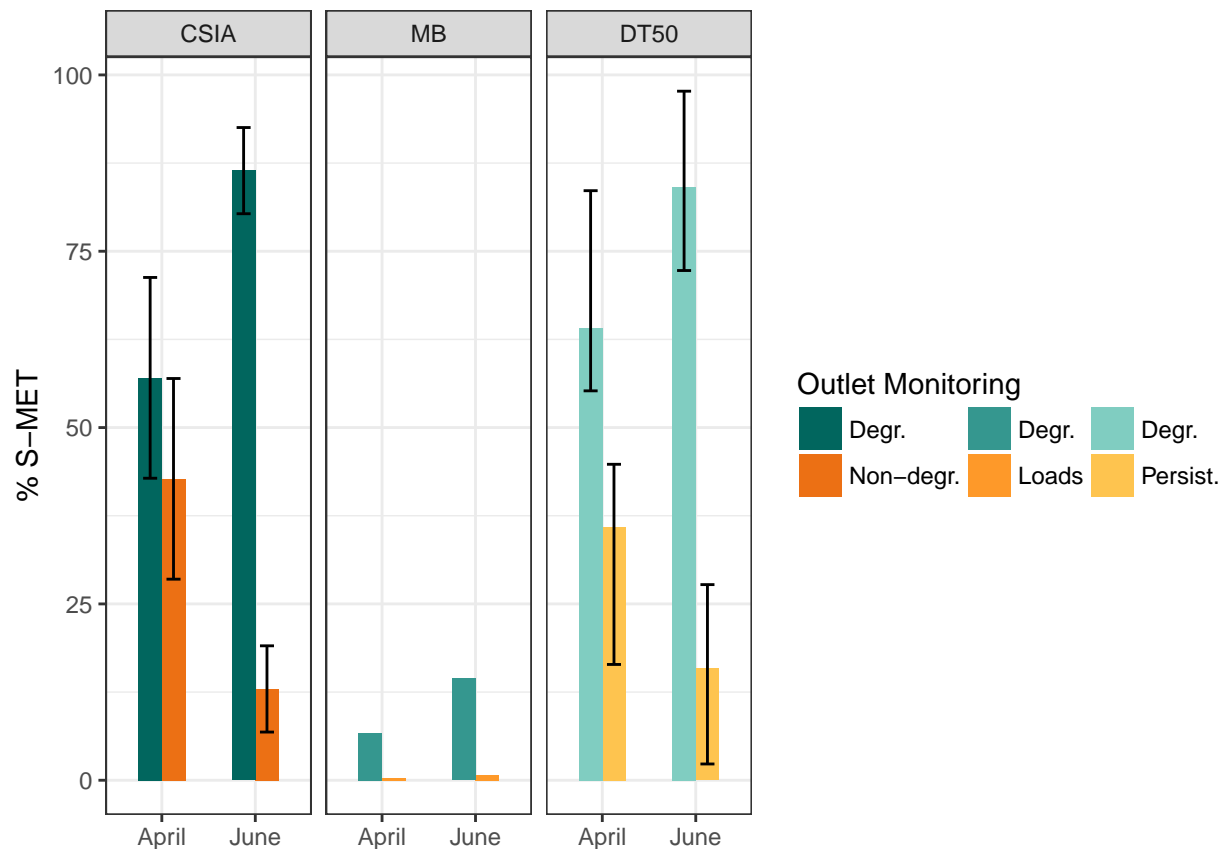
OutBars <- ggplot(data = balTidyType , aes(x=Month, y=measure, fill = Sink, ymin=SD1, ymax=SD2))+
  geom_bar(stat = "identity", position = "dodge", width = 0.5) +
  geom_errorbar(#aes(ymin=SD1, ymax=SD2),
               width=.3 , # ) + #,                                # Width of the error bars
               position=position_dodge(.5)) +

  theme_bw() +
  ylab("% S-MET") +
  theme(axis.title.x = element_blank() ) +
  #xlab("Month") +
  facet_wrap(~type) +
  scale_fill_manual(values = c("#01665e", "#35978f", "#80cdc1", "#ec7014", "#fe9929", "#fec44f" ), # bl
                    #values = c("#238b45", "#41ab5d", "#74c476", "#40004b", "#762a83", "#9970ab" ), # g
                    #values = c("#238b45", "#41ab5d", "#74c476", "#ec7014", "#fe9929", "#fec44f" ), # g
                    #values = c("#80cdc1", "#018571", "#a6611a", "#dfc27d", "#80cdc1", "#018571"),
                    name= "Outlet Monitoring" ,# element_blank(), # "Mass Balance", # \n
                    breaks=c("B", "f" ,
                              "TPout" , "SMout" ,
                              "Deg", "Rem"
                              ),
                    labels=c("Degr.", "Non-degr.",
                              "Degr.", "Loads",
                              "Degr." , "Persist."
                              )) +

  guides(fill=guide_legend(ncol=3))

OutBars

```



Compare to Catchment soils

```
names(soils)
```

```
## [1] "Date.ti"          "Event"            "timeSinceApp"
## [4] "diss.d13C"        "SD.d13C"          "CumAppMass.g"
## [7] "B.mean.comp.North" "B.max.comp.North" "B.min.comp.North"
## [10] "MassSoil.g.North" "comp.d13C.North"  "comp.d13C.SD.North"
## [13] "B.mean.comp.Talweg" "B.max.comp.Talweg" "B.min.comp.Talweg"
## [16] "MassSoil.g.Talweg" "comp.d13C.Talweg" "comp.d13C.SD.Talweg"
## [19] "B.mean.comp.South" "B.max.comp.South" "B.min.comp.South"
## [22] "MassSoil.g.South" "comp.d13C.South"  "comp.d13C.SD.South"
## [25] "ID.S"             "CatchMassSoil.g"  "BulkMass.g"
## [28] "BulkCatch.d13"    "BulkCatch.d13.SD" "f.mean.bulk"
## [31] "B.mean.bulk"      "DD13C.North"      "DD13C.Talweg"
## [34] "DD13C.South"
```

```
keepMB <- c("Date.ti", "CumAppMass.g", "CatchMassSoil.g",
            "f.mean.bulk", "B.mean.bulk")
```

```
soilsMB <- soils[, (names(soils) %in% keepMB)]
```

```
soilsMB$Rem.measure <- (soilsMB$CatchMassSoil.g/soilsMB$CumAppMass.g)*100
soilsMB$Unk.measure <- 100 - soilsMB$Rem.measure
soilsMB$CatchMassSoil.g <- NULL
```

```

soilsMB$CumAppMass.g <- NULL

soils.April <- subset(soilsMB, (Date.ti > as.POSIXct("2016-04-01 00:00:00", tz = "EST")
                    & Date.ti < as.POSIXct("2016-05-01 00:00:00", tz = "EST")))
soils.April <- na.omit(soils.April)

soils.June <- subset(soilsMB, (Date.ti > as.POSIXct("2016-06-07 00:00:00", tz = "EST")
                    & Date.ti <= as.POSIXct("2016-06-28 14:52:00", tz = "EST")))
soils.June <- na.omit(soils.June)

B.mean.maySol <- mean(soils.April$B.mean.bulk)
B.sd.maySol <- sd(soils.April$B.mean.bulk)
f.mean.maySol <- mean(soils.April$f.mean.bulk*100)
f.sd.maySol <- sd(soils.April$f.mean.bulk*100)
Rem.mean.maySol <- mean(soils.April$Rem.measure)
Rem.sd.maySol <- sd(soils.April$Rem.measure)
Unk.mean.maySol <- mean(soils.April$Unk.measure)
Unk.sd.maySol <- sd(soils.April$Unk.measure)

B.mean.juneSol <- mean(soils.June$B.mean.bulk)
B.sd.juneSol <- sd(soils.June$B.mean.bulk)
f.mean.juneSol <- mean(soils.June$f.mean.bulk*100)
f.sd.juneSol <- sd(soils.June$f.mean.bulk*100)
Rem.mean.juneSol <- mean(soils.June$Rem.measure)
Rem.sd.juneSol <- sd(soils.June$Rem.measure)
Unk.mean.juneSol <- mean(soils.June$Unk.measure)
Unk.sd.juneSol <- sd(soils.June$Unk.measure)

Month <- c("April", "June")
balSol <- data.frame(Month)
balSol$B.measure <- c(B.mean.maySol, B.mean.juneSol)
balSol$B.SD1 <- c(B.mean.maySol-B.sd.maySol, B.mean.juneSol-B.sd.juneSol)
balSol$B.SD2 <- c(B.mean.maySol+B.sd.maySol, B.mean.juneSol+B.sd.juneSol)
balSol$f.measure <- c(f.mean.maySol, f.mean.juneSol)
balSol$f.SD1 <- c(f.mean.maySol-f.sd.maySol, f.mean.juneSol-f.sd.juneSol)
balSol$f.SD2 <- c(f.mean.maySol+f.sd.maySol, f.mean.juneSol+f.sd.juneSol)

balSol$Unk.measure <- c(Unk.mean.maySol, Unk.mean.juneSol)
balSol$Unk.SD1 <- c(Unk.mean.maySol-Unk.sd.maySol, Unk.mean.juneSol-Unk.sd.juneSol)
balSol$Unk.SD2 <- c(Unk.mean.maySol+Unk.sd.maySol, Unk.mean.juneSol+Unk.sd.juneSol)

balSol$Rem.measure <- c(Rem.mean.maySol, Rem.mean.juneSol)
balSol$Rem.SD1 <- c(Rem.mean.maySol-Rem.sd.maySol, Rem.mean.juneSol-Rem.sd.juneSol)
balSol$Rem.SD2 <- c(Rem.mean.maySol+Rem.sd.maySol, Rem.mean.juneSol+Rem.sd.juneSol)

solTidy <- balSol %>%
  gather(measure, value, -Month) %>% # Melts data frame
  separate(measure, into = c("Sink", "temporary_var")) %>% # parses the sep = "." into...
  spread(temporary_var, value) # Moves molten temporary variable to own column

type <- rep(c("CSIA", "CSIA", "MB", "MB"), 2)

```

```

balTidySol <- cbind(solTidy, type)
balTidySol$Sink <- as.factor(balTidySol$Sink)

levels(balTidySol$Sink)

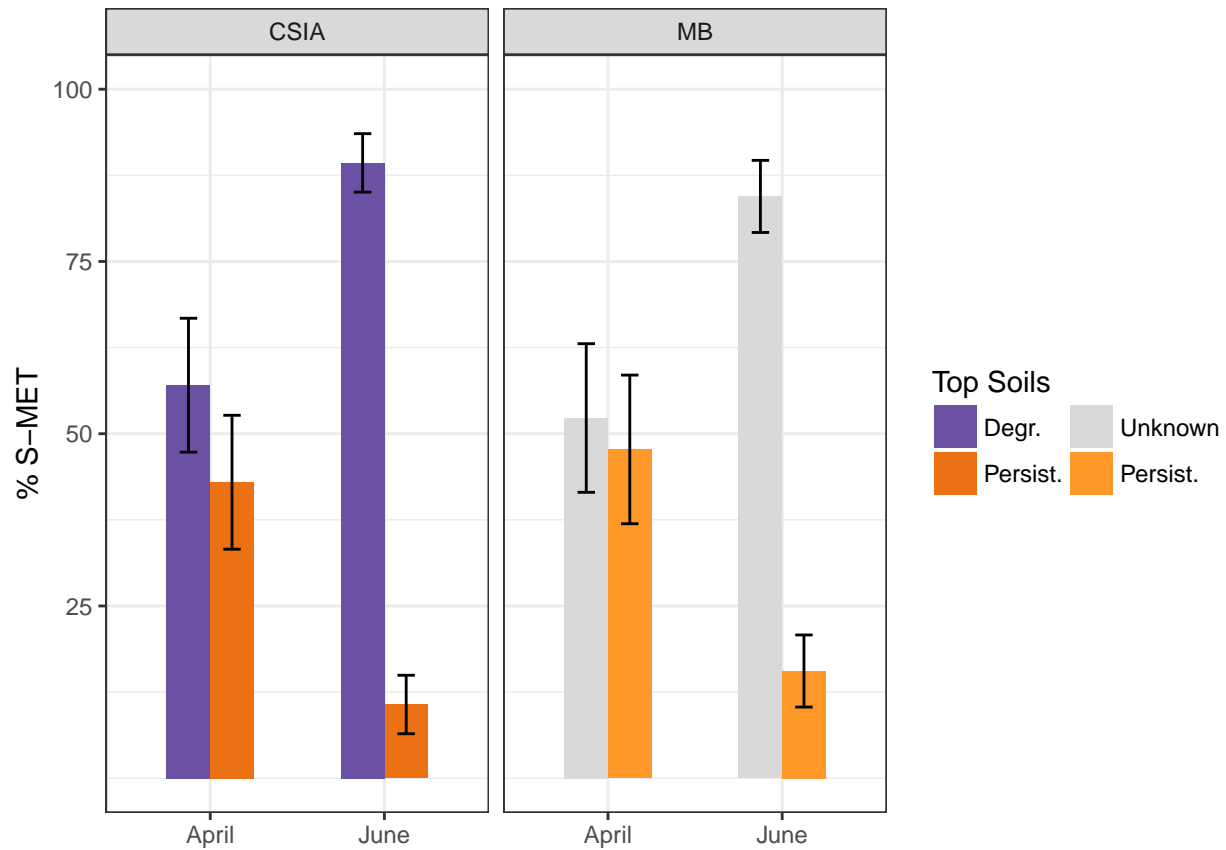
## [1] "B"    "f"    "Rem"  "Unk"

balTidySol$Sink <- factor(balTidySol$Sink, levels = c("B" , "f" , "Unk", "Rem"))

SoilBars <- ggplot(data = balTidySol , aes(x=Month, y=measure, fill = Sink, ymin=SD1, ymax=SD2))+
  geom_bar(stat = "identity", position = "dodge", width = 0.5) +
  geom_errorbar(#aes(ymin=SD1, ymax=SD2),
               width=.2 , # ) + #,                                # Width of the error bars
               position=position_dodge(.5)) +
  theme_bw() +
  ylab("% S-MET") +
  theme(axis.title.x = element_blank() ) +
  xlab("Month") +
  facet_wrap(~type) +
  scale_y_continuous( breaks=c(25, 50, 75, 100), limits = c(0, 100) )+ #expand=c(0, 10, 0, 0)) +
  scale_fill_manual(#values = c("#6a51a3" , "#ec7014", "#807dba", "#fe9929"), # purple-orange
                   values = c("#6a51a3" , "#ec7014", "#d9d9d9", "#fe9929"), # Unknown as grey
                   name= "Top Soils" ,# element_blank(), # "Mass Balance", # \n
                   breaks=c("B", "f" ,
                             "Unk" , "Rem"
                           ),
                   labels=c("Degr.", "Persist.",
                             "Unknown", "Persist." ))+
  guides(fill=guide_legend(ncol=2))

SoilBars

```



Merge both Outlet and Soils - BARS

```
#balAll <- rbind(balTidyType, balTidySol)

OutBars_noLeg <- OutBars + theme(legend.position = 'none')
OutBars_Leg <- get_legend(OutBars)

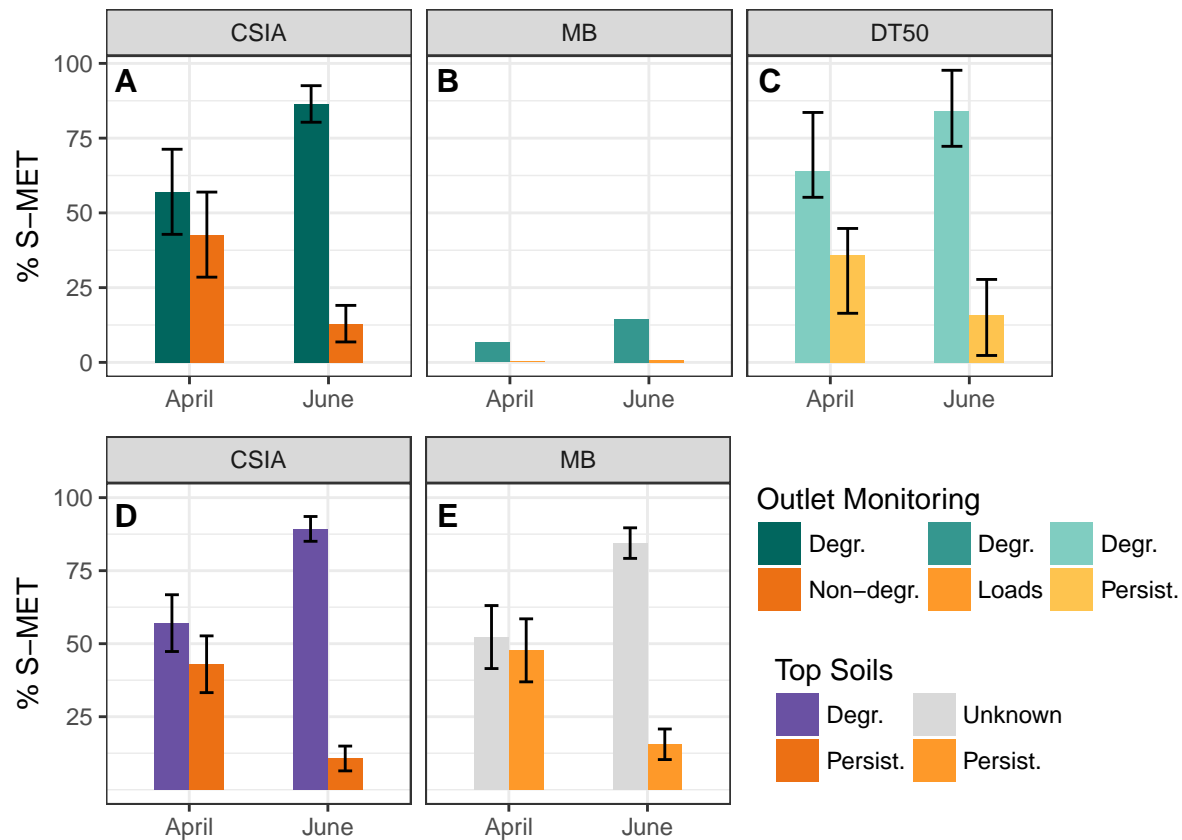
SoilBars_noLeg <- SoilBars + theme(legend.position = 'none')
SoilBars_Leg <- get_legend(SoilBars)

#plot_grid(OutBars_noLeg, SoilBars_noLeg,
#           ncol =1, nrow = 2, align ="v" )
#
#           labels = c("A", "C", "B", "D"))

balAllplot <- ggdraw() +
  draw_plot(OutBars_noLeg, x=0.01, y=.5, width = 0.86, height = .5) +
  draw_plot(SoilBars_noLeg, x=0.01, y = 0.0, width = 0.60, height = 0.5) +
  draw_plot(OutBars_Leg, x=0.74, y = 0.3, width = 0.1, height = 0.1) +
  draw_plot(SoilBars_Leg, x=0.70, y = 0.1, width = 0.1, height = 0.1) +
  draw_label("A", x= 0.11, y = .9, size = 12, fontface = "bold") +
  draw_label("D", x= 0.11, y = .39, size = 12, fontface = "bold") +
  draw_label("B", x= 0.37, y = .9, size = 12, fontface = "bold") +
  draw_label("E", x= 0.37, y = .39, size = 12, fontface = "bold") +
  draw_label("C", x= 0.63, y = .9, size = 12, fontface = "bold")
```



```
balAllplot
```



```
#ggsave(balAllplot, filename = "MB_CSIA_Bars.png", width = 8, height = 5, units = "in", scale = 1)
```

Water Rayleigh plots

```
#waters$remain_maxHalf
#waters$remain_minHalf

waterCo <- max(waters$Conc.mug.L)
d13Co

## [1] -31.2144

waters$yRaleigh <- log((1000+d13Co+waters$DD13C.diss)/(1000+d13Co))
waters$xRaleigh <- log(waters$Conc.mug.L/waterCo)
waters$Dla <- waters$maxQ*waters$Volume.m3/waters$Duration.Hrs

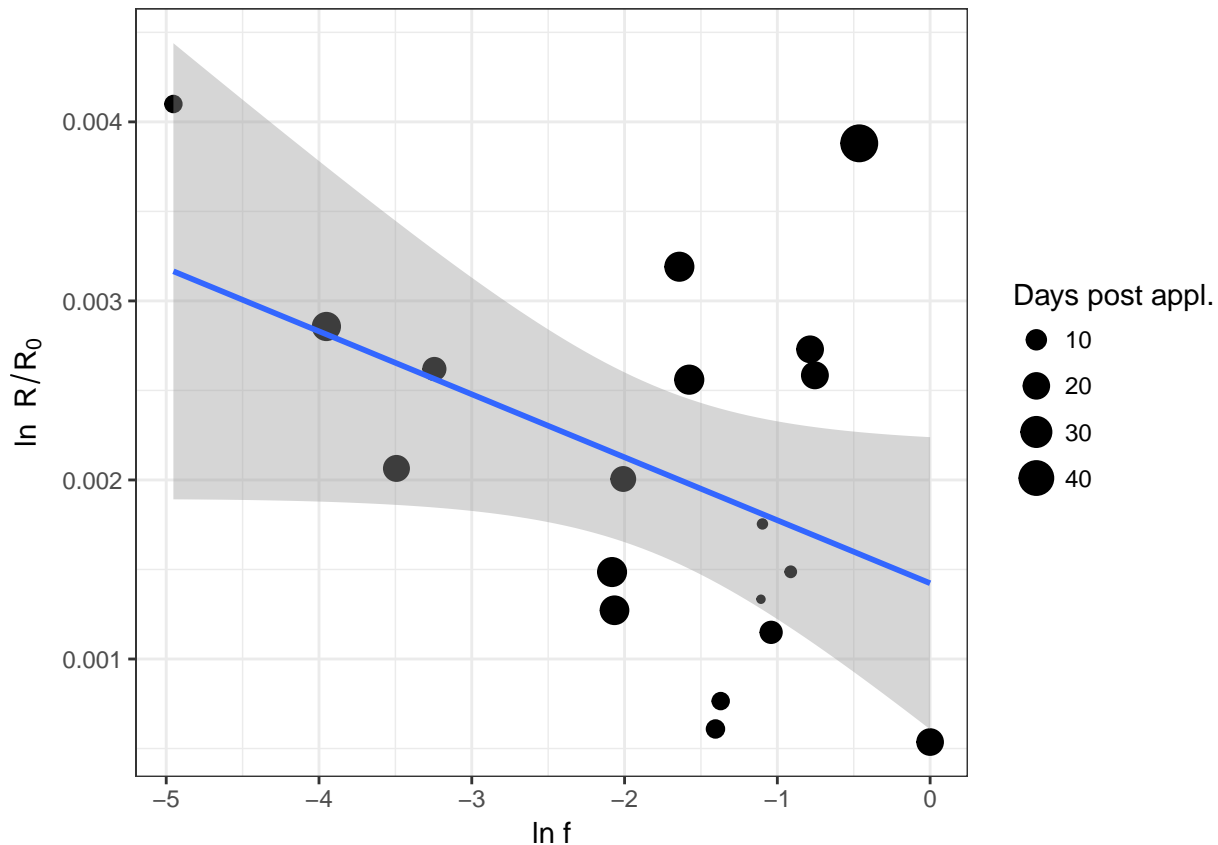
waterClean <- subset(waters, Sampled == "Sampled")
waterModel<-lm(yRaleigh~xRaleigh, data= waterClean)
summary(waterModel)

##
## Call:
## lm(formula = yRaleigh ~ xRaleigh, data = waterClean)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.0018789 -0.0008911 -0.0002357  0.0006844  0.0023033
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0019996  0.0004099   4.879  5.1e-05 ***
## xRaleigh     -0.0001201  0.0001413  -0.850   0.403
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.001243 on 25 degrees of freedom
## (11 observations deleted due to missingness)
## Multiple R-squared:  0.0281, Adjusted R-squared:  -0.01077
## F-statistic: 0.7229 on 1 and 25 DF,  p-value: 0.4033

cof <- as.numeric(coef(model)[2]*1000)
se <- summary(model)$coef[[4]]*1000

waterRaleigh <- ggplot(data = subset(waterClean, (!is.na(yRaleigh) & xRaleigh > -7 & ngC.mean.diss > 5))
  geom_point(aes(size = timeSinceApp)) +
  theme_bw() +
  scale_size_continuous(range = c(1, 6)) +
  labs(size="Days post appl.") +
  xlab("ln f") +
  ylab("ln R/Ro") +
  ylab(expression(paste("ln ", R / R['0'] ))) +
  stat_smooth(data= subset(waterClean, (!is.na(yRaleigh) & xRaleigh > -7 & ngC.mean.diss > 5)), method
waterRaleigh
```



```
# ggsave(waterRaleigh, filename = "lnDDvslnConc_water.png", width = 8, height = 5, units = "in", scale = 1)
# Date conversion correct:
sum(is.na(waters$Date.ti)) == 0
```

```
## [1] TRUE
```

```
str(waters)
```

```
## 'data.frame':   51 obs. of  79 variables:
## $ Date.ti      : POSIXct, format: "2016-03-25 00:04:00" "2016-03-25 12:04:00" ...
## $ WeekSubWeek  : Factor w/ 51 levels "W0-0x","W0-1",...: 1 2 3 4 5 6 26 27 28 29 ...
## $ tf           : Factor w/ 51 levels "2016-03-25 12:02:00",...: 1 2 3 4 5 6 7 8 9 10 ...
## $ iflux        : num  1.25 1.12 1.31 1.46 16.33 ...
## $ fflux        : num  1.13 1.31 1.46 16.45 15.18 ...
## $ changeflux   : num  -0.119 0.189 0.148 14.989 -1.15 ...
## $ maxQ         : num  1.25 1.38 1.64 38.4 18.67 ...
## $ minQ         : num  1.118 1.082 0.929 1.449 13.201 ...
## $ dryHrs       : num  0.0167 6.0167 47.3 66.1333 1.65 ...
## $ Duration.Hrs : num  12 82.5 37.6 27.3 23.1 ...
## $ chExtreme    : num  -0.13 0.256 0.33 36.944 -3.133 ...
## $ Peak         : int   NA NA NA 1 NA NA 2 NA NA 3 ...
## $ AveDischarge.m3.h : num  1.2 1.21 1.28 14.32 15.53 ...
## $ Volume.m3     : num  14.4 100.2 48.3 390.4 359.2 ...
## $ Sampled.Hrs   : num  12 82.5 37.6 27.3 23.1 ...
## $ Sampled       : Factor w/ 2 levels "Not Sampled",...: 1 2 1 2 2 1 2 2 1 2 ...
## $ Conc.mug.L    : num  0.246 0.246 3.517 6.788 6.561 ...
## $ Conc.SD       : num  0.0193 0.0193 0.1544 0.2894 0.1906 ...
## $ OXA_mean      : num  4.82 4.82 17.68 30.53 32.49 ...
```

```

## $ OXA_SD : num 1.141 1.141 5.663 10.185 0.243 ...
## $ ESA_mean : num 18.1 18.1 32 46 41.3 ...
## $ ESA_SD : num 3.497 3.497 3.267 3.037 0.853 ...
## $ diss.d13C : num NA NA NA -30.5 -30.6 ...
## $ SD.d13C : num NA NA NA 0.106 0.151 ...
## $ N_ngC.diss : int NA NA NA 3 3 NA 3 3 NA 3 ...
## $ ngC.mean.diss : num NA NA NA 42.7 54.7 ...
## $ ngC.SD.diss : num NA NA NA 1.92 2.54 ...
## $ Conc.Solids.mug.gMES : num 0.645 0.645 0.385 0.126 0.436 ...
## $ Conc.Solids.ug.gMES.SD : num 0.0232 0.0232 0.0252 0.0271 0.1232 ...
## $ filt.d13C : num NA NA NA NA NA ...
## $ filt.SD.d13C : num NA NA NA NA NA ...
## $ N_ngC.fl : int NA NA NA NA NA NA 3 3 NA NA ...
## $ ngC.mean.fl : num NA NA NA NA NA ...
## $ ngC.SD.fl : num NA NA NA NA NA ...
## $ DD13C.diss : num NA NA NA 0.741 0.59 ...
## $ DD13C.filt : num NA NA NA NA NA ...
## $ B.diss : num NA NA NA 35.4 29.4 ...
## $ B.filt : num NA NA NA NA NA ...
## $ NH4.mM : num NA NA NA 0.05 NA NA NA NA NA NA ...
## $ TIC.ppm.filt : num NA NA NA 51.8 44.8 NA 66.7 52.1 NA 69.4 ...
## $ Cl.mM : num NA NA NA 1.48 1574 ...
## $ NO3...mM : num NA NA NA 616 778 ...
## $ PO4..mM : int NA NA NA NA NA NA NA NA NA NA ...
## $ NPOC.ppm : num NA NA NA 4 4.4 NA 5.8 3.4 NA 9.1 ...
## $ TIC.ppm.unfilt : num NA NA NA 44.8 26.4 NA 39 32.3 NA 54.8 ...
## $ TOC.ppm.unfilt : num NA NA NA 4.7 5.4 NA 2.7 3.8 NA 3.9 ...
## $ ExpMES.Kg : num 5.35 5.35 14.88 24.4 8.08 ...
## $ Appl.Mass.g : num 9498 0 0 0 0 ...
## $ timeSinceApp : num 0.5 3.9 5.5 6.6 7.6 11.6 12.6 14 20.6 2.2 ...
## $ CumAppMass.g : num 9498 9498 9498 9498 9498 ...
## $ DissSmeto.g : num 0.00354 0.0246 0.17004 2.64991 2.357 ...
## $ DissSmeto.g.SD : num 0.000278 0.001934 0.007463 0.11298 0.068486 ...
## $ DissOXA.g : num 0.0695 0.4832 0.8547 11.9184 11.6727 ...
## $ DissOXA.g.SD : num 0.0165 0.1143 0.2738 3.976 0.0873 ...
## $ DissESA.g : num 0.26 1.81 1.55 17.95 14.83 ...
## $ DissESA.g.SD : num 0.0504 0.3503 0.158 1.1855 0.3066 ...
## $ FiltSmeto.mg.SD : num 0.124 0.124 0.374 0.66 0.996 ...
## $ FiltSmeto.g : num 0.00345 0.00345 0.00573 0.00307 0.00352 ...
## $ FiltSmeto.g.SD : num 0.000124 0.000124 0.000374 0.00066 0.000996 ...
## $ TotSMout.g : num 0.00699 0.02806 0.17577 2.65298 2.36052 ...
## $ TotSMout.g.SD : num 0.000216 0.00137 0.005284 0.07989 0.048432 ...
## $ MELsm.g : num 0.302 2.078 2.379 30.241 27.008 ...
## $ MELsm.g.SD : num 0.0269 0.1868 0.1789 2.4062 0.1634 ...
## $ CumOutDiss.g : num 0.00354 0.02815 0.19818 2.84809 5.2051 ...
## $ CumOutFilt.g : num 0.00345 0.0069 0.01263 0.01571 0.01923 ...
## $ CumOutSmeto.g : num 0.00699 0.03505 0.21082 2.8638 5.22432 ...
## $ CumOutMELsm.g : num 0.302 2.38 4.76 35.001 62.009 ...
## $ BalMassDisch.g : num 9498 9495 9493 9463 9436 ...
## $ prctMassOut : num 4.98e-05 2.00e-04 1.25e-03 1.89e-02 1.68e-02 ...
## $ FracDeltaOut : num 0 0 0 -0.576 -0.515 ...
## $ Events : Factor w/ 51 levels "0-1","0-2","0-3",...: 1 2 3 4 5 6 7 8 9 10 ...
## $ Weeks : Factor w/ 16 levels "W0","W1","W10",...: 1 1 1 2 2 2 9 9 9 10 ...
## $ Event : Factor w/ 19 levels "0","1","2","3",...: 1 1 1 2 2 2 3 3 3 4 ...

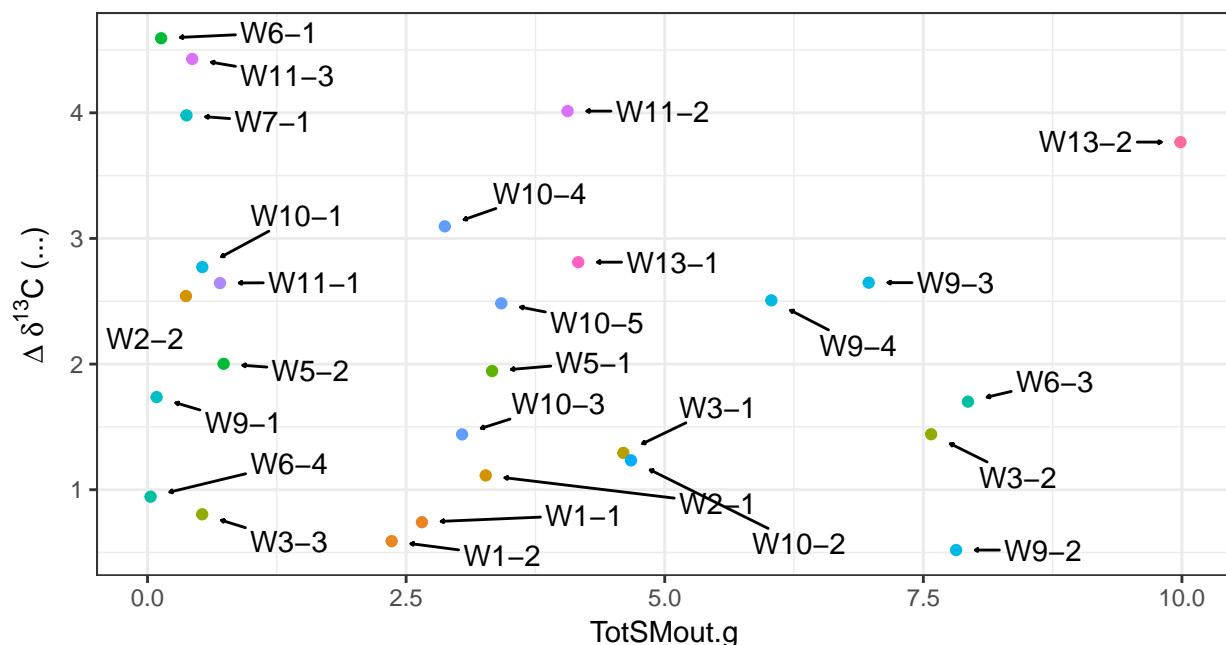
```

```
## $ remainMedTheo_prc      : num  98.8 91 87.7 85.3 83.4 ...
## $ remainMinTheo_prc     : num  99.3 94.2 92 90.5 89.2 ...
## $ remainMaxTheo_prc    : num  97.2 79.7 72.8 68.1 64.4 ...
## $ yRaleigh              : num  NA NA NA 0.000765 0.000609 ...
## $ xRaleigh              : num  -4.69 -4.69 -2.03 -1.37 -1.4 ...
## $ DIa                   : num  1.5 1.68 2.1 549.76 289.92 ...
```

```
ggplot(waterClean, aes(x=TotSMout.g, y=DD13C.diss))+
  geom_point(aes(group = Event, colour = Event))+
  theme_bw() +
  theme(legend.position="top"
        # axis.title.x = element_blank(),
        ) +
  guides(col = guide_legend(nrow = 3)) + #,
        # shape = guide_legend(nrow = 3)) +
  ylab(expression(paste({Delta~delta}^{"13"}, "C", ' (\\u2030)')) +
  # xlab(expression(paste("Conc. S-Meto. ", {(\\mu}*g / L)}))) +
  geom_text_repel(aes(label=WeekSubWeek),
    arrow = arrow(length = unit(0.005, 'npc'), type = "closed"),
    force = 1,
    point.padding = unit(1.0, 'lines'),
    max.iter = 2e3,
    nudge_x = .2)
```

Event

● 0	● 3	● 6	● 9	● 12	● 16
● 1	● 4	● 7	● 10	● 14	● 17
● 2	● 5	● 8	● 11	● 15	● 18



```
# cor.test(waterClean$TotSMout.g, waterClean$DD13C.diss)

pearson_water_r <- cor.test(waterClean$Conc.mug.L, waterClean$DD13C.diss)[4]
water_r_label <- sprintf("Pearson~r == %0.2f", pearson_water_r)
water_p_value <- cor.test(waterClean$Conc.mug.L, waterClean$DD13C.diss)[3]
```

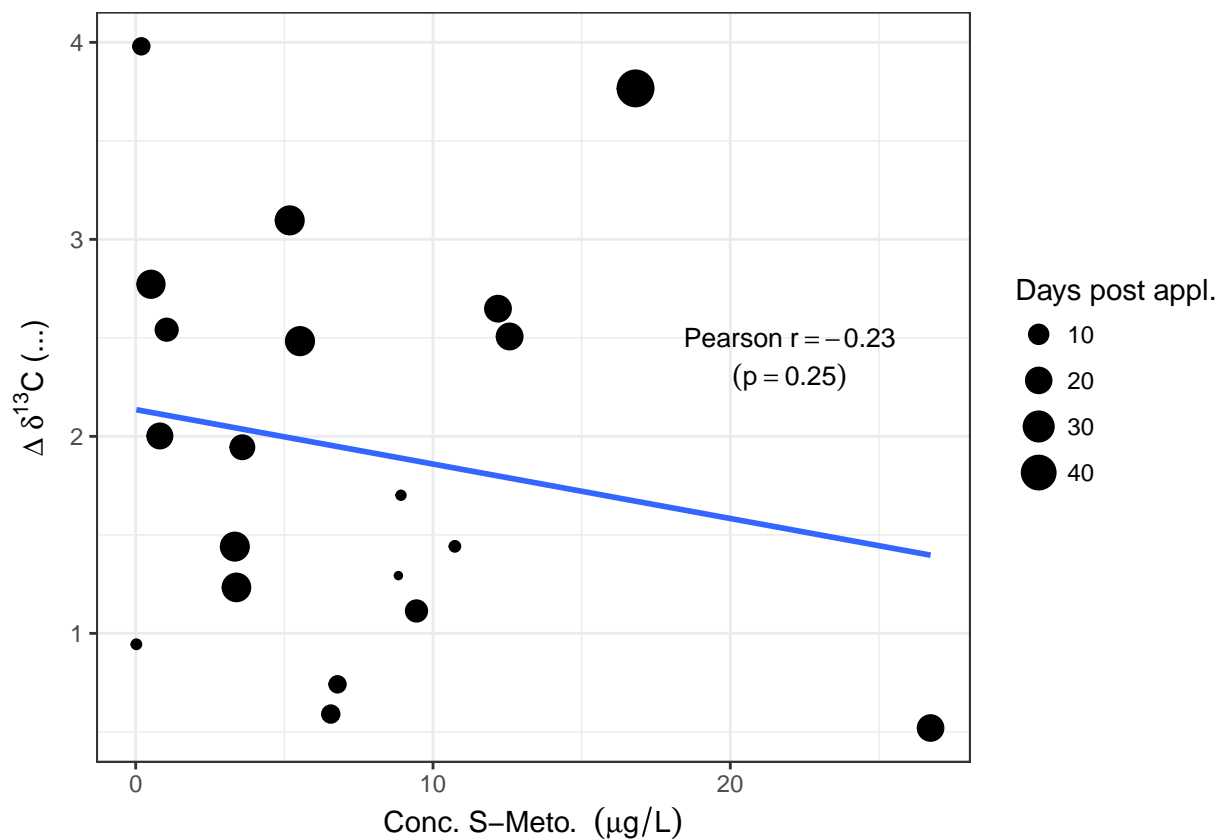
```

water_p_label <- sprintf("(p == %0.2f)", water_p_value)

waterIsoConc <- ggplot(data = subset(waterClean, ngC.mean.diss > 5), aes(x=Conc.mug.L, y=DD13C.diss))+
  stat_smooth(data = subset(waterClean, ngC.mean.diss > 5),
    aes(x=Conc.mug.L, y=DD13C.diss), method = "lm", formula = y~x, se=F) +
  geom_point(aes(size = timeSinceApp)) +
  theme_bw() +
  scale_size_continuous(range = c(1, 6)) +
  labs(size="Days post appl.") +
  ylab(expression(paste({\Delta}^{13}C, "C", " (\u2030)"))) +
  xlab(expression(paste("Conc. S-Meto. ", {\mu}g / L)))) +
  annotate("text", x = 22, y = 2.5, label = as.character(water_r_label), parse = T, size = 3.5) +
  annotate("text", x = 22, y = 2.3, label = water_p_label, parse = T, size = 3.5)

```

waterIsoConc



```

# ggsave(waterIsoConc , filename = "DDvsConc_water.png", width = 8, height = 5, units = "in", scale = 1

```

Correlations Waters

```
cor.test(waters$Conc.mug.L, waters$diss.d13C)
```

```

##
## Pearson's product-moment correlation
##
## data:  waters$Conc.mug.L and waters$diss.d13C

```

```
## t = -1.1775, df = 25, p-value = 0.2501
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.5604226 0.1651756
## sample estimates:
## cor
## -0.2292267
```

```
#cor.test(waters$TotSMout.g, waters$diss.d13C)
```

```
#esaoxa <- waters$MELsm.g-waters$TotSMout.g
# cor.test(esaoxa, waters$diss.d13C)
```

Merge Soil and Water data frames

Outlet Isotope Shifts (DD)

In the same plot consider this secondary axis, where the secondary axis is a formulat of the first:

```
ggplot(mpg, aes(displ, hwy)) + geom_point() + scale_y_continuous( "mpg (US)", sec.axis = sec_axis(~ . *
1.20, name = "mpg (UK)") )
```

Or this: <https://github.com/tidyverse/ggplot2/wiki/Align-two-plots-on-a-page>

```
waterClean_ng <- subset(waterClean, ngC.mean.diss > 0)
WaterSoils <- merge(waterClean_ng, soils, by = "Date.ti", all = T)
```

```
str(WaterSoils)
```

```
## 'data.frame': 52 obs. of 112 variables:
## $ Date.ti : POSIXct, format: "2016-03-25 00:04:00" "2016-03-25 12:04:00" ...
## $ WeekSubWeek : Factor w/ 51 levels "W0-0x","W0-1",...: NA NA NA 4 5 NA 26 27 NA 29 ...
## $ tf : Factor w/ 51 levels "2016-03-25 12:02:00",...: NA NA NA 4 5 NA 7 8 NA 10 .
## $ iflux : num NA NA NA 1.46 16.33 ...
## $ fflux : num NA NA NA 16.4 15.2 ...
## $ changeflux : num NA NA NA 14.99 -1.15 ...
## $ maxQ : num NA NA NA 38.4 18.7 ...
## $ minQ : num NA NA NA 1.45 13.2 ...
## $ dryHrs : num NA NA NA 66.13 1.65 ...
## $ Duration.Hrs : num NA NA NA 27.3 23.1 ...
## $ chExtreme : num NA NA NA 36.94 -3.13 ...
## $ Peak : int NA NA NA 1 NA NA 2 NA NA 3 ...
## $ AveDischarge.m3.h : num NA NA NA 14.3 15.5 ...
## $ Volume.m3 : num NA NA NA 390 359 ...
## $ Sampled.Hrs : num NA NA NA 27.3 23.1 ...
## $ Sampled : Factor w/ 2 levels "Not Sampled",...: NA NA NA 2 2 NA 2 2 NA 2 ...
## $ Conc.mug.L : num NA NA NA 6.79 6.56 ...
## $ Conc.SD : num NA NA NA 0.289 0.191 ...
## $ OXA_mean : num NA NA NA 30.5 32.5 ...
## $ OXA_SD : num NA NA NA 10.185 0.243 ...
## $ ESA_mean : num NA NA NA 46 41.3 ...
## $ ESA_SD : num NA NA NA 3.037 0.853 ...
```

```

## $ diss.d13C.x          : num  NA NA NA -30.5 -30.6 ...
## $ SD.d13C.x            : num  NA NA NA 0.106 0.151 ...
## $ N_ngC.diss           : int   NA NA NA 3 3 NA 3 3 NA 3 ...
## $ ngC.mean.diss        : num  NA NA NA 42.7 54.7 ...
## $ ngC.SD.diss          : num  NA NA NA 1.92 2.54 ...
## $ Conc.Solids.mug.gMES : num  NA NA NA 0.126 0.436 ...
## $ Conc.Solids.ug.gMES.SD: num  NA NA NA 0.0271 0.1232 ...
## $ filt.d13C            : num  NA NA NA NA NA ...
## $ filt.SD.d13C         : num  NA NA NA NA NA ...
## $ N_ngC.fl             : int   NA NA NA NA NA NA 3 3 NA NA ...
## $ ngC.mean.fl          : num  NA NA NA NA NA ...
## $ ngC.SD.fl            : num  NA NA NA NA NA ...
## $ DD13C.diss           : num  NA NA NA 0.741 0.59 ...
## $ DD13C.filt           : num  NA NA NA NA NA ...
## $ B.diss               : num  NA NA NA 35.4 29.4 ...
## $ B.filt               : num  NA NA NA NA NA ...
## $ NH4.mM               : num  NA NA NA 0.05 NA NA NA NA NA ...
## $ TIC.ppm.filt         : num  NA NA NA 51.8 44.8 NA 66.7 52.1 NA 69.4 ...
## $ Cl.mM                : num  NA NA NA 1.48 1574 ...
## $ NO3...mM             : num  NA NA NA 616 778 ...
## $ PO4..mM              : int   NA NA NA NA NA NA NA NA NA ...
## $ NPOC.ppm             : num  NA NA NA 4 4.4 NA 5.8 3.4 NA 9.1 ...
## $ TIC.ppm.unfilt       : num  NA NA NA 44.8 26.4 NA 39 32.3 NA 54.8 ...
## $ TOC.ppm.unfilt       : num  NA NA NA 4.7 5.4 NA 2.7 3.8 NA 3.9 ...
## $ ExpMES.Kg            : num  NA NA NA 24.4 8.08 ...
## $ Appl.Mass.g          : num  NA NA NA 0 0 ...
## $ timeSinceApp.x       : num  NA NA NA 6.6 7.6 NA 12.6 14 NA 2.2 ...
## $ CumAppMass.g.x       : num  NA NA NA 9498 9498 ...
## $ DissSmeto.g          : num  NA NA NA 2.65 2.36 ...
## $ DissSmeto.g.SD       : num  NA NA NA 0.113 0.0685 ...
## $ DissOXA.g            : num  NA NA NA 11.9 11.7 ...
## $ DissOXA.g.SD         : num  NA NA NA 3.976 0.0873 ...
## $ DissESA.g            : num  NA NA NA 18 14.8 ...
## $ DissESA.g.SD         : num  NA NA NA 1.186 0.307 ...
## $ FiltSmeto.mg.SD      : num  NA NA NA 0.66 0.996 ...
## $ FiltSmeto.g          : num  NA NA NA 0.00307 0.00352 ...
## $ FiltSmeto.g.SD       : num  NA NA NA 0.00066 0.000996 ...
## $ TotSMout.g           : num  NA NA NA 2.65 2.36 ...
## $ TotSMout.g.SD        : num  NA NA NA 0.0799 0.0484 ...
## $ MELsm.g              : num  NA NA NA 30.2 27 ...
## $ MELsm.g.SD           : num  NA NA NA 2.406 0.163 ...
## $ CumOutDiss.g         : num  NA NA NA 2.85 5.21 ...
## $ CumOutFilt.g         : num  NA NA NA 0.0157 0.0192 ...
## $ CumOutSmeto.g        : num  NA NA NA 2.86 5.22 ...
## $ CumOutMELsm.g        : num  NA NA NA 35 62 ...
## $ BalMassDisch.g       : num  NA NA NA 9463 9436 ...
## $ prctMassOut          : num  NA NA NA 0.0189 0.0168 ...
## $ FracDeltaOut         : num  NA NA NA -0.576 -0.515 ...
## $ Events               : Factor w/ 51 levels "0-1","0-2","0-3",...: NA NA NA 4 5 NA 7 8 NA 10 ...
## $ Weeks                : Factor w/ 16 levels "W0","W1","W10",...: NA NA NA 2 2 NA 9 9 NA 10 ...
## $ Event.x              : Factor w/ 19 levels "0","1","2","3",...: NA NA NA 2 2 NA 3 3 NA 4 ...
## $ remainMedTheo_prc    : num  NA NA NA 85.3 83.4 ...
## $ remainMinTheo_prc    : num  NA NA NA 90.5 89.2 ...
## $ remainMaxTheo_prc    : num  NA NA NA 68.1 64.4 ...

```



```
## $ yRaleigh          : num NA NA NA 0.000765 0.000609 ...
## $ xRaleigh          : num NA NA NA -1.37 -1.4 ...
## $ DIa               : num NA NA NA 550 290 ...
## $ Event.y           : int 0 0 0 1 1 1 2 2 2 3 ...
## $ timeSinceApp.y     : num 0.5 3.9 5.5 6.6 7.6 11.6 12.6 14 20.6 2.2 ...
## $ diss.d13C.y        : num NA NA NA -30.5 -30.6 ...
## $ SD.d13C.y          : num NA NA NA 0.106 0.151 ...
## $ CumAppMass.g.y     : num 9498 9498 9498 9498 9498 ...
## $ B.mean.comp.North  : num NA NA NA NA NA ...
## $ B.max.comp.North   : num NA NA NA NA NA ...
## $ B.min.comp.North   : num NA NA NA NA NA ...
## $ MassSoil.g.North   : num 12.4 NA NA 613.5 NA ...
## $ comp.d13C.North    : num NA NA NA NA NA ...
## $ comp.d13C.SD.North : num NA NA NA NA NA ...
## $ B.mean.comp.Talweg : num NA NA NA NA NA ...
## $ B.max.comp.Talweg  : num NA NA NA NA NA ...
## $ B.min.comp.Talweg  : num NA NA NA NA NA ...
## $ MassSoil.g.Talweg  : num 4.33 NA NA 173.11 NA ...
## $ comp.d13C.Talweg   : num NA NA NA NA NA ...
## $ comp.d13C.SD.Talweg : num NA NA NA NA NA ...
## $ B.mean.comp.South  : num NA NA NA NA NA ...
## $ B.max.comp.South   : num NA NA NA NA NA ...
## $ B.min.comp.South   : num NA NA NA NA NA ...
## [list output truncated]
```

```
names(WaterSoils)
```

```
## [1] "Date.ti"           "WeekSubWeek"
## [3] "tf"                "iflux"
## [5] "fflux"             "changeflux"
## [7] "maxQ"              "minQ"
## [9] "dryHrs"            "Duration.Hrs"
## [11] "chExtreme"         "Peak"
## [13] "AveDischarge.m3.h" "Volume.m3"
## [15] "Sampled.Hrs"       "Sampled"
## [17] "Conc.mug.L"        "Conc.SD"
## [19] "OXA_mean"          "OXA_SD"
## [21] "ESA_mean"          "ESA_SD"
## [23] "diss.d13C.x"       "SD.d13C.x"
## [25] "N_ngC.diss"        "ngC.mean.diss"
## [27] "ngC.SD.diss"       "Conc.Solids.mug.gMES"
## [29] "Conc.Solids.ug.gMES.SD" "filt.d13C"
## [31] "filt.SD.d13C"      "N_ngC.fl"
## [33] "ngC.mean.fl"       "ngC.SD.fl"
## [35] "DD13C.diss"        "DD13C.filt"
## [37] "B.diss"            "B.filt"
## [39] "NH4.mM"            "TIC.ppm.filt"
## [41] "Cl.mM"             "NO3...mM"
## [43] "PO4..mM"           "NPOC.ppm"
## [45] "TIC.ppm.unfilt"    "TOC.ppm.unfilt"
## [47] "ExpMES.Kg"         "Appl.Mass.g"
## [49] "timeSinceApp.x"    "CumAppMass.g.x"
## [51] "DissSmeto.g"       "DissSmeto.g.SD"
## [53] "DissOXA.g"         "DissOXA.g.SD"
## [55] "DissESA.g"         "DissESA.g.SD"
```

```
## [57] "FiltSmeto.mg.SD"      "FiltSmeto.g"
## [59] "FiltSmeto.g.SD"      "TotSMout.g"
## [61] "TotSMout.g.SD"      "MELsm.g"
## [63] "MELsm.g.SD"         "CumOutDiss.g"
## [65] "CumOutFilt.g"       "CumOutSmeto.g"
## [67] "CumOutMELsm.g"      "BalMassDisch.g"
## [69] "prctMassOut"        "FracDeltaOut"
## [71] "Events"             "Weeks"
## [73] "Event.x"            "remainMedTheo_prc"
## [75] "remainMinTheo_prc"  "remainMaxTheo_prc"
## [77] "yRaleigh"          "xRaleigh"
## [79] "DIa"               "Event.y"
## [81] "timeSinceApp.y"     "diss.d13C.y"
## [83] "SD.d13C.y"         "CumAppMass.g.y"
## [85] "B.mean.comp.North"  "B.max.comp.North"
## [87] "B.min.comp.North"   "MassSoil.g.North"
## [89] "comp.d13C.North"    "comp.d13C.SD.North"
## [91] "B.mean.comp.Talweg" "B.max.comp.Talweg"
## [93] "B.min.comp.Talweg"  "MassSoil.g.Talweg"
## [95] "comp.d13C.Talweg"   "comp.d13C.SD.Talweg"
## [97] "B.mean.comp.South"  "B.max.comp.South"
## [99] "B.min.comp.South"   "MassSoil.g.South"
## [101] "comp.d13C.South"    "comp.d13C.SD.South"
## [103] "ID.S"              "CatchMassSoil.g"
## [105] "BulkMass.g"         "BulkCatch.d13"
## [107] "BulkCatch.d13.SD"   "f.mean.bulk"
## [109] "B.mean.bulk"        "DD13C.North"
## [111] "DD13C.Talweg"       "DD13C.South"
```

```
keepWS <- c("Date.ti", "WeekSubWeek", "maxQ", "AveDischarge.m3.h",
            "DD13C.diss", "SD.d13C.x",
            "DD13C.filt", "filt.SD.d13C",
            "DD13C.Talweg", "comp.d13C.SD.Talweg",
            "DD13C.South", "comp.d13C.SD.South",
            "DD13C.North", "comp.d13C.SD.North",
            "Event.x", "BulkCatch.d13", "BulkCatch.d13.SD"
            # "timeSinceApp.x", "Event.x", "Events"
            )
```

```
wsSmall <- WaterSoils[ , (names(WaterSoils) %in% keepWS)]
```

```
wsSmall$DD13.Bulk <- wsSmall$BulkCatch.d13-initialDelta
```

```
names(wsSmall)
```

```
## [1] "Date.ti"      "WeekSubWeek"      "maxQ"
## [4] "AveDischarge.m3.h" "SD.d13C.x"        "filt.SD.d13C"
## [7] "DD13C.diss"    "DD13C.filt"       "Event.x"
## [10] "comp.d13C.SD.North" "comp.d13C.SD.Talweg" "comp.d13C.SD.South"
## [13] "BulkCatch.d13"   "BulkCatch.d13.SD" "DD13C.North"
## [16] "DD13C.Talweg"   "DD13C.South"      "DD13.Bulk"
```

```
wsSmall <- wsSmall[c("Date.ti", "Event.x", "WeekSubWeek", "maxQ", "AveDischarge.m3.h",
                    "DD13C.diss", "SD.d13C.x",
                    "DD13C.filt", "filt.SD.d13C",
                    "DD13C.Talweg", "comp.d13C.SD.Talweg",
```

```

      "DD13C.South", "comp.d13C.SD.South",
      "DD13C.North", "comp.d13C.SD.North",
      "DD13.Bulk", "BulkCatch.d13.SD")]
```

```

names(wsSmall) <- c("Date", "Event", "Week", "Qmax", "Qmean",
  "diss.measure", "diss.SD",
  "filt.measure", "filt.SD",
  "Talweg.measure", "Talweg.SD",
  "South.measure", "South.SD",
  "North.measure", "North.SD",
  "BulkDD.measure", "BulkDD.SD"
)
# Conventional way of melting won't work if we need SDs.
# ws <- melt(wsSmall, id=c("Date.ti", "timeSinceApp.x", "Events", "Event.x"))

# Need to rename the columns so that I can use separate and spread from the package tidyr
#names(wsSmall)[-1][seq(2, length(names(wsSmall)) - 1, 2)] <-
# paste0(names(wsSmall)[-1][seq(1, length(names(wsSmall)) - 1, 2)], "-SD")
#names(wsSmall)[-1][seq(1, length(names(wsSmall)) - 1, 2)] <-
# paste0(names(wsSmall)[-1][seq(1, length(names(wsSmall)) - 1, 2)], "-measure")

wstidier <- wsSmall %>%
  gather(measure, value, -Date, -Event, -Week, -Qmax, -Qmean) %>% # Melts data frame
  separate(measure, into = c("Location", "temporary_var")) %>% # parses the sep = "." into...
  spread(temporary_var, value)

wstidier$Type <- ifelse(wstidier$Location == "diss", "Dissolved",
  ifelse(wstidier$Location == "filt", "Sediment",
    "Top Soil"))

wstidier$Source <- ifelse(wstidier$Location == "diss", "Outlet",
  ifelse(wstidier$Location == "filt", "Outlet",
    ifelse(wstidier$Location == "South", "South",
      ifelse(wstidier$Location == "Talweg", "Valley",
        ifelse(wstidier$Location == "BulkDD", "Bulk",
          "North")))) ))

wstidier$Source <- as.factor(wstidier$Source)
wstidier$Type <- as.factor(wstidier$Type)
wstidier$Event <- as.numeric(wstidier$Event)

# Copy all data
wstidierAll <- wstidier

levels(wstidier$Source)

## [1] "Bulk" "North" "Outlet" "South" "Valley"

levels(wstidier$Type)

## [1] "Dissolved" "Sediment" "Top Soil"

#wstidier$Source <- factor(wstidier$Source, levels = c("Bulk", "Plateau", "Valley", "Outlet"))
wstidier$Source <- factor(wstidier$Source, levels = c("Bulk", "North", "Valley", "South", "Outlet"))
wstidier$Type <- factor(wstidier$Type, levels = c("Top Soil", "Dissolved", "Sediment" ))

```

```

# epsilon
epsilon_field <- cof
initialDelta

## [1] -31.2144

wstidier$DegField <- (1-((1000 + d13Co + wstidier$measure)/(1000+d13Co))^(1000/epsilon_field))*100
wstidier$DegLab <- (1-((1000 + d13Co + wstidier$measure)/(1000+d13Co))^(1000/epsilon_lab))*100

wstidier$DegDiff <- (wstidier$DegField - wstidier$DegLab)

wstidier$Location <- as.factor(wstidier$Location)
#wstidier$Week <- as.factor(wstidier$Week)

#library(dplyr)
#detach("package:plyr")
sumary <- na.omit(wstidier) %>%
  group_by(Type) %>%
  summarise(mean = mean(DegDiff))

limits_DdC <- aes(ymin=measure-SD, ymax=measure+SD, colour = Source)

wstidier2 = subset(wstidier, SD <= 1 ) ## Source != "Bulk" ) ## Date < as.POSIXct('2016-06-14 08:04:00')
NoBASE <- subset(wstidier2, Week != "W6-1" )

pd <- position_dodge(width = 0.4)
wsALL_lab <- ggplot(data = wstidier2, aes(x = Date, y = measure, group = Source) )+
  geom_errorbar(data=subset(wstidier2, Source == 'Bulk'), limits_DdC, size=0.2) +
  geom_errorbar(data=subset(wstidier2, Source == 'Valley' &
    Date > as.POSIXct('2016-05-14 08:04:00')), limits_DdC, size=0.2) +
  geom_errorbar(data=subset(wstidier2, Source == 'Outlet'), limits_DdC) +
  geom_point(data=subset(wstidier2, Source == 'Outlet'), aes(shape = Type, colour = Source, size = Qmax),
    #geom_point(data=subset(wstidier2, Source == 'Outlet'), aes(shape = Type, colour = Source, size = Qmax),
  geom_point(data=subset(wstidier2, Source == 'Bulk'), aes(shape = Type, colour = Source)) +
  geom_point(data=subset(wstidier2, Source == 'Valley' &
    Date > as.POSIXct('2016-05-14 08:04:00')), aes(shape = Type, colour = Source),
    #geom_point(data=subset(wstidier2, Source == 'Valley' & Date > as.POSIXct('2016-05-14 08:04:00')), aes(shape = Type, colour = Source),

#stat_smooth(data=subset(wstidier,
#  (Source == "Valley" & Event > 8 )),
#  method = "lm", formula = y ~ poly(x, 2), se = F, colour = 'darkgreen' , alpha = 0.1, size=0.2) +
#stat_smooth(data=subset(wstidier,
#  (Source != "Outlet" & Source != "Valley" & Event < 20 )),
#  method = "lm", formula = y ~ poly(x, 2), se = F, alpha = 0.1, size=0.2) +

# Water
stat_smooth(data=subset(NoBASE,
  (Source == "Outlet" & Event > 1 & Type == "Dissolved")),
  method = "lm", formula = y ~ poly(x, 2), se = T, aes(colour = 'Outlet'), alpha = 0.2, size=0.2) +

# North
stat_smooth(data=subset(wstidier2,
  (Source == "Bulk" )), #/ Source == "South" )),

```

```

        method = "lm", formula = y ~ poly(x, 2), se = T, aes(colour = 'Bulk'), alpha = 0.2, size=
#stat_smooth(data=subset(wstidier2,
#
#           (Source == "South")),
#           method = "lm", formula = y ~ poly(x, 2), se = T, aes(colour = 'South'), alpha = 0.2, siz
theme_bw() +
# Applies
annotate("text", x = as.POSIXct('2016-03-28 08:04:00'), y = 0,
        label = as.character(expression(paste( "\u066D"))), parse = T, size = 6.0) +
annotate("text", x = as.POSIXct('2016-04-05 00:04:00'), y = 0,
        label = as.character(expression(paste( "\u066D"))), parse = T, size = 6.0) +
annotate("text", x = as.POSIXct('2016-04-13 08:04:00'), y = 0,
        label = as.character(expression(paste( "\u066D"))), parse = T, size = 6.0) +
annotate("text", x = as.POSIXct('2016-05-25 08:04:00'), y = 0,
        label = as.character(expression(paste( "\u066D"))), parse = T, size = 6.0) +
# Title applies
annotate("text", x = as.POSIXct('2016-04-05 08:04:00'), y = 4.5,
        label = as.character(expression(paste( "\u066D", " Applications"))), parse = T, size = 4.0)

scale_x_datetime(breaks = date_breaks("2 weeks"), labels = date_format("%b %d")) +
theme(legend.position="top"
      # axis.title.x = element_blank(),
      # axis.text.x=element_text(angle = 45, hjust = 1)
      ) +
# geom_smooth(data=subset(ws, Source != "Outlet"), method = "lm", formula = y ~ poly(x, 2)) +
xlab("Date") +
#ylab(expression(paste({Delta~delta}~"13", "C", ' (\u2030)')) +
scale_y_continuous(
  expression(paste({Delta~delta}~"13", "C", ' (\u2030)')),
  sec.axis = sec_axis(trans = ~ (1-((1000 + d13Co + .)/(1000+d13Co))^(1000/epsilon_lab))*100 ,
    name = element_blank(), breaks=c(20, 40, 60, 70, 80, 85, 90, 95) )# breaks=seq(
) +
scale_color_manual(name= "Source",
  values = c("#F8766D", "#00BFC4", "#00BA38", "#B79F00", "#619CFF", "#F564E3",
    "#D55E00", "darkgreen", "dodgerblue")
  ) +
scale_size_continuous(range = c(1, 3)) # +
# scale_shape_manual(name= )

wsALL_field <- ggplot(data = wstidier2, aes(x = Date, y = measure, group = Source) )+
  geom_errorbar(data=subset(wstidier2, Source == 'Bulk'), limits_DdC, size=0.2) +
  geom_errorbar(data=subset(wstidier2, Source == 'Valley' &
    Date > as.POSIXct('2016-05-14 08:04:00')), limits_DdC, size=0.2) +
  geom_errorbar(data=subset(wstidier2, Source == 'Outlet'), limits_DdC) +
  geom_point(data=subset(wstidier2, Source == 'Outlet'), aes(shape = Type, colour = Source, size = Qmax)
# geom_point(data=subset(wstidier2, Source == 'Outlet'), aes(shape = Type, colour = Source, size = Qm
  geom_point(data=subset(wstidier2, Source == 'Bulk'), aes(shape = Type, colour = Source)) +
  geom_point(data=subset(wstidier2, Source == 'Valley' &
    Date > as.POSIXct('2016-05-14 08:04:00')), aes(shape = Type, colour = Source)

#stat_smooth(data=subset(wstidier,
#
#           (Source == "Valley" & Event > 8 )),
#           method = "lm", formula = y ~ poly(x, 2), se = F, colour = 'darkgreen' , alpha = 0.1, siz

```

```

#stat_smooth(data=subset(wstidier,
#                        (Source != "Outlet" & Source != "Valley" & Event < 20 )),
#            method = "lm", formula = y ~ poly(x, 2), se = F, alpha = 0.1, size=0.2) +

# Water
stat_smooth(data=subset(wstidier2, #NoBASE,
                        (Source == "Outlet" & Event > 1 & Type == "Dissolved")),
            method = "lm", formula = y ~ poly(x, 2), se = T, aes(colour = 'Outlet'), alpha = 0.2, size=0.2) +

# North
stat_smooth(data=subset(wstidier2,
                        (Source == "Bulk" )), #/ Source == "South" )),
            method = "lm", formula = y ~ poly(x, 2), se = T, aes(colour = 'Bulk'), alpha = 0.2, size=0.2) +

#stat_smooth(data=subset(wstidier2,
#                        (Source == "South")),
#            method = "lm", formula = y ~ poly(x, 2), se = T, aes(colour = 'South'), alpha = 0.2, size=0.2) +

theme_bw() +
# Applies
annotate("text", x = as.POSIXct('2016-03-28 08:04:00'), y = 0,
        label = as.character(expression(paste( "\u0066D"))), parse = T, size = 6.0) +
annotate("text", x = as.POSIXct('2016-04-05 00:04:00'), y = 0,
        label = as.character(expression(paste( "\u0066D"))), parse = T, size = 6.0) +
annotate("text", x = as.POSIXct('2016-04-13 08:04:00'), y = 0,
        label = as.character(expression(paste( "\u0066D"))), parse = T, size = 6.0) +
annotate("text", x = as.POSIXct('2016-05-25 08:04:00'), y = 0,
        label = as.character(expression(paste( "\u0066D"))), parse = T, size = 6.0) +

# Title applies
annotate("text", x = as.POSIXct('2016-04-05 08:04:00'), y = 4.5,
        label = as.character(expression(paste( "\u0066D", " Applications"))), parse = T, size = 4.0) +

scale_x_datetime(breaks = date_breaks("1 month"), labels = date_format("%b %d")) +
theme(legend.position="top"
      # axis.title.x = element_blank()
      # axis.text.x=element_text(angle = 45, hjust = 1)
      ) +
# geom_smooth(data=subset(ws, Source != "Outlet"), method = "lm", formula = y ~ poly(x, 2)) +
xlab("Date") +
#ylab(expression(paste({Delta~delta}~"13", "C", ' (\u2030)')) +
scale_y_continuous(
    expression(paste({Delta~delta}~"13", "C", ' (\u2030)')),
    sec.axis = sec_axis(trans = ~ (1-((1000 + d13Co + .)/(1000+d13Co))^(1000/epsilon_field))*100 ,
                        name = "Degradation (%)", breaks=c(20, 40, 60, 70, 80, 85, 90, 95) )# breaks=seq(20, 95, by=5)
) +
scale_color_manual(name= "Source",
                  values = c("#F8766D", "#00BFC4", "#00BA38", "#B79F00", "#619CFF", "#F564E3",
                            "#D55E00", "darkgreen", "dodgerblue")
                  ) +
scale_size_continuous(range = c(1, 3)) +
guides(col = guide_legend(nrow = 3), shape=guide_legend(nrow=2), size = guide_legend(nrow=3))# +
# scale_shape_manual(name= )

#ggplotly(wsALL_field)

# May 12th event

```

```
# Two samples from 2 very close (t = +3hrs) peaks
mean(c( 1.70, 0.94))
```

```
## [1] 1.32
sd(c( 1.70, 0.94))
```

```
## [1] 0.5374012
# May 29th event
# 2 samples from second peak
mean(c( 2.65, 2.56))
```

```
## [1] 2.605
sd(c( 2.65, 2.56))
```

```
## [1] 0.06363961
# 2 samples from end event (9-4 and 10-1)
mean(c(2.507, 2.772))
```

```
## [1] 2.6395
sd(c(2.507, 2.772))
```

```
## [1] 0.1873833
# June 2-3 event
# 2 samples from 1st peak
mean(c( 1.23, 1.44 ))
```

```
## [1] 1.335
sd(c( 1.23, 1.44 ))
```

```
## [1] 0.1484924
```

Join all figures

```
#wsALL_lab
#wsALL_field
#wsPlot
# ggsave(wsALL, filename = "WaterSoilvsTime.png", width = 8, height = 5, units = "in", scale = 1)
# ggsave(wsALL, filename = "WaterBulkvsTime.png", width = 8, height = 5, units = "in", scale = 1)

wsALL_field_noLeg <- wsALL_field + theme(legend.position='none')
wsALL_lab_noLeg <- wsALL_lab + theme(legend.position='none')
wsAll_field_Leg <- get_legend(wsALL_field)

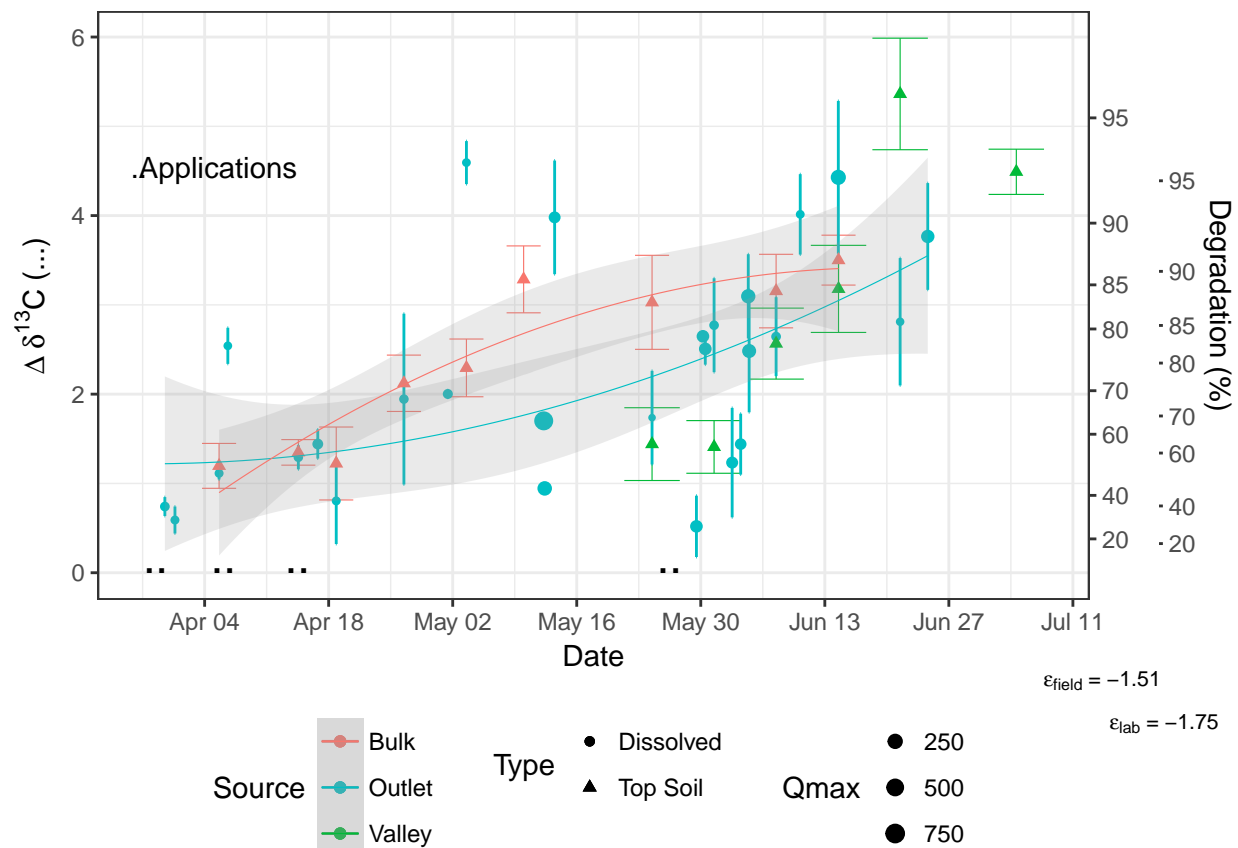
label <- substitute(paste(epsilon, " = ", epsilon_f, ", Field", epsilon, " = " , epsilon_l),
                    list(epsilon_f = signif(epsilon_field, 2), epsilon_l = signif(epsilon_lab, 2) ))

label2 <- substitute(paste(epsilon ["field"] , " = ", epsilon_f),
                    list(epsilon_f = signif(epsilon_field, 3)))
```

```
label3 <- substitute(paste(epsilon ["lab"] , " = ", epsilon_l),
                     list(epsilon_l = signif(epsilon_lab, 3)))
# adding label via ggdraw, in the ggdraw coordinates

wsALL <- ggdraw() +
  draw_plot(wsALL_field_noLeg, x=0, y=.2, width = 1, height = .8) +
  draw_plot(wsALL_lab_noLeg, x=0, y = 0.2, width = 0.935, height = 0.8) +
  draw_label(label2, x= .89, y = .2, size = 8) +
  draw_label(label3, x= .94, y = .15, size = 8) +
  draw_plot(wsAll_field_Leg, x=0.2, y=0.0, width = 0.55, height = 0.15)
```

wsALL



```
#ggsave(wsALL, filename = "WaterSoilvsTime.png", width = 8, height = 5, units = "in", scale = 1)
```

Comparison to Lutz et al. (2013)

```
Time <- c(0, 4, 6, 12, 24)
Intensity <- c(30, 0, 0, 0, 0)
Delta <- c(0, 2, 3, 3.3, 3.3)
Event <- rep("Lutz", 5)
Approach <- rep("Simulation", 5)

events <- data.frame(Time, Intensity, Delta, Event, Approach)
```



```
# Build other vectors from field data
# then rbind to events..
```

Soils and Water with labels (inspection)

```
# Data without the Plateau
wsNoPlat <- subset(wstidierAll, Source != "Plateau")
wsNoPlat <- subset(wstidier, SD < 4)
wsNoPlat$Source <- factor(wsNoPlat$Source, levels = c("Bulk", "Valley", "Outlet"))
#levels(wsNoPlat$Source)

# Subset the data to values with SD < 1
wsNoPlat2 = subset(wsNoPlat, SD < 1.50)

limits_DdC <- aes(ymin=measure-SD, ymax=measure+SD, colour = Source)

wsPlot <- ggplot(data = wsNoPlat, aes(x = Date, y = measure)) +
  geom_errorbar(limits_DdC) +
  geom_jitter(aes(shape = Type, colour = Source)) +
  stat_smooth(data=subset(wsNoPlat,
                          (Source == "Valley" & Event > 8 )),
              method = "lm", formula = y ~ poly(x, 2), se = F, colour = 'green4' , alpha = 0.1, size=0.2) +
  stat_smooth(data=subset(wsNoPlat,
                          (Source != "Outlet" & Source != "Valley" & Event < 20 )),
              method = "lm", formula = y ~ poly(x, 2), se = F, alpha = 0.1, size=0.2) +
  stat_smooth(data=subset(wsNoPlat,
                          (Source == "Outlet" & Event > 1 & Type == "Dissolved")),
              method = "lm", formula = y ~ poly(x, 2), se = T, aes(colour = 'Outlet'), alpha = 0.2, size=0.2) +
  #stat_smooth(data=subset(wsNoPlat,
  #                          (Source == "Bulk")),
  #            method = "lm", formula = y ~ poly(x, 2), se = T, aes(colour = 'Bulk'), alpha = 0.2, size=0.2) +
  theme_bw() +
  # Applies
  annotate("text", x = as.POSIXct('2016-03-28 08:04:00'), y = 0,
          label = as.character(expression(paste( "\u006D"))), parse = T, size = 6.0) +
  annotate("text", x = as.POSIXct('2016-04-05 08:04:00'), y = 0,
          label = as.character(expression(paste( "\u006D"))), parse = T, size = 6.0) +
  annotate("text", x = as.POSIXct('2016-04-13 08:04:00'), y = 0,
          label = as.character(expression(paste( "\u006D"))), parse = T, size = 6.0) +
  annotate("text", x = as.POSIXct('2016-05-17 08:04:00'), y = 0,
          label = as.character(expression(paste( "\u006D"))), parse = T, size = 6.0) +
  # Title applies
  annotate("text", x = as.POSIXct('2016-04-01 08:04:00'), y = 7.5,
          label = as.character(expression(paste( "\u006D", " Applications"))), parse = T, size = 4.0) +

  scale_x_datetime(breaks = date_breaks("2 weeks"), labels = date_format("%b %d")) +
  theme(legend.position="top"
        # axis.title.x = element_blank(),
        # axis.text.x=element_text(angle = 45, hjust = 1)
        ) +
```

```

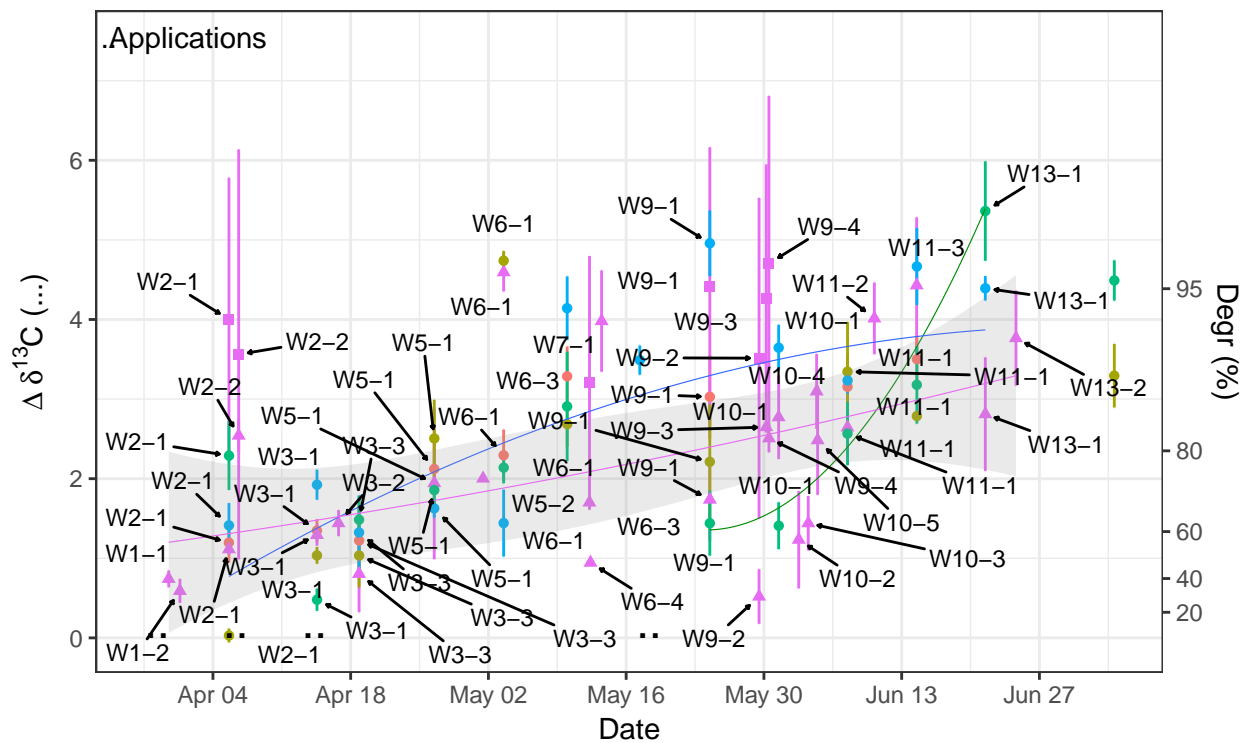
# geom_smooth(data=subset(ws, Source != "Outlet"), method = "lm", formula = y ~ poly(x, 2)) +
xlab("Date") +
#ylab(expression(paste({Delta~delta}^{"13"}, "C", ' (\u2030)')) +
scale_y_continuous(
  expression(paste({Delta~delta}^{"13"}, "C", ' (\u2030)')),
  sec.axis = sec_axis(trans = ~ (1-((1000 + d13Co + .)/(1000+d13Co))^(1000/epsilon_field))*100 ,
    name = "Degr (%)", breaks=c(20, 40, 60, 80, 95) )# breaks=seq(20, 120, 15))
) +
geom_text_repel(aes(label=as.factor(Week)),
  size = 3,
  arrow = arrow(length = unit(0.005, 'npc'), type = "closed"),
  force = 0.5,
  point.padding = unit(0.5, 'lines'),
  max.iter = 2e3,
  nudge_x = .05)

# +
#scale_color_manual(name= "Source",
#  values = c("black", "dodgerblue", "green", "red")
#  ) +
# scale_shape_manual(name= )

```

wsPlot

Source ● Bulk ● North ● Valley ● South ● Outlet Type ● Top Soil ▲ Dissolved ■ Sedime



Testing difference in $\Delta\delta$ between groups

Based on ANOVA tests, there is:

- No significant difference between soils and water

```
names(WaterSoils)
```

```
## [1] "Date.ti" "WeekSubWeek"
## [3] "tf" "iflux"
## [5] "fflux" "changeflux"
## [7] "maxQ" "minQ"
## [9] "dryHrs" "Duration.Hrs"
## [11] "chExtreme" "Peak"
## [13] "AveDischarge.m3.h" "Volume.m3"
## [15] "Sampled.Hrs" "Sampled"
## [17] "Conc.mug.L" "Conc.SD"
## [19] "OXA_mean" "OXA_SD"
## [21] "ESA_mean" "ESA_SD"
## [23] "diss.d13C.x" "SD.d13C.x"
## [25] "N_ngC.diss" "ngC.mean.diss"
## [27] "ngC.SD.diss" "Conc.Solids.mug.gMES"
## [29] "Conc.Solids.ug.gMES.SD" "filt.d13C"
## [31] "filt.SD.d13C" "N_ngC.fl"
## [33] "ngC.mean.fl" "ngC.SD.fl"
## [35] "DD13C.diss" "DD13C.filt"
## [37] "B.diss" "B.filt"
## [39] "NH4.mM" "TIC.ppm.filt"
## [41] "Cl.mM" "NO3...mM"
## [43] "PO4...mM" "NPOC.ppm"
## [45] "TIC.ppm.unfilt" "TOC.ppm.unfilt"
## [47] "ExpMES.Kg" "Appl.Mass.g"
## [49] "timeSinceApp.x" "CumAppMass.g.x"
## [51] "DissSmeto.g" "DissSmeto.g.SD"
## [53] "DissOXA.g" "DissOXA.g.SD"
## [55] "DissESA.g" "DissESA.g.SD"
## [57] "FiltSmeto.mg.SD" "FiltSmeto.g"
## [59] "FiltSmeto.g.SD" "TotSMout.g"
## [61] "TotSMout.g.SD" "MELsm.g"
## [63] "MELsm.g.SD" "CumOutDiss.g"
## [65] "CumOutFilt.g" "CumOutSmeto.g"
## [67] "CumOutMELsm.g" "BalMassDisch.g"
## [69] "prctMassOut" "FracDeltaOut"
## [71] "Events" "Weeks"
## [73] "Event.x" "remainMedTheo_prc"
## [75] "remainMinTheo_prc" "remainMaxTheo_prc"
## [77] "yRaleigh" "xRaleigh"
## [79] "DIa" "Event.y"
## [81] "timeSinceApp.y" "diss.d13C.y"
## [83] "SD.d13C.y" "CumAppMass.g.y"
## [85] "B.mean.comp.North" "B.max.comp.North"
## [87] "B.min.comp.North" "MassSoil.g.North"
## [89] "comp.d13C.North" "comp.d13C.SD.North"
## [91] "B.mean.comp.Talweg" "B.max.comp.Talweg"
## [93] "B.min.comp.Talweg" "MassSoil.g.Talweg"
```

```

## [95] "comp.d13C.Talweg"      "comp.d13C.SD.Talweg"
## [97] "B.mean.comp.South"    "B.max.comp.South"
## [99] "B.min.comp.South"     "MassSoil.g.South"
## [101] "comp.d13C.South"      "comp.d13C.SD.South"
## [103] "ID.S"                 "CatchMassSoil.g"
## [105] "BulkMass.g"           "BulkCatch.d13"
## [107] "BulkCatch.d13.SD"     "f.mean.bulk"
## [109] "B.mean.bulk"          "DD13C.North"
## [111] "DD13C.Talweg"         "DD13C.South"

keepDDtest <- c(
  "Date.ti",
  "diss.d13C.x", # "DD13C.diss",
  "comp.d13C.North", "comp.d13C.Talweg", "comp.d13C.South" #,
  # "DD13C.North", "DD13C.Talweg", "DD13C.South"
)

wsStatTest <- WaterSoils[, colnames(WaterSoils) %in% keepDDtest]

mwsStatTest <- melt(wsStatTest, id="Date.ti")
mwsStatTest$Group1 <- ifelse(mwsStatTest$variable == "diss.d13C.x", "Outlet", "Soil")
mwsStatTest$Group2 <- ifelse(mwsStatTest$variable == "diss.d13C.x", "Outlet",
  ifelse(mwsStatTest$variable == "comp.d13C.Talweg", "Valley", "Plateau"))
mwsStatTest$Group3 <- ifelse(mwsStatTest$variable == "diss.d13C.x" &
  mwsStatTest$Date.ti > as.POSIXct('2016-06-05 00:06:00', tz = 'EST'), "Outlet",
  ifelse(mwsStatTest$variable == "diss.d13C.x" &
    mwsStatTest$Date.ti <= as.POSIXct('2016-06-05 00:06:00', tz = 'EST'), "Plateau",
    ifelse(mwsStatTest$variable == "comp.d13C.Talweg" &
      mwsStatTest$Date.ti > as.POSIXct('2016-06-05 00:06:00', tz = 'EST'), "Valley",
      ifelse(mwsStatTest$variable == "comp.d13C.Talweg" &
        mwsStatTest$Date.ti <= as.POSIXct('2016-06-05 00:06:00', tz = 'EST'), "Valley",
        ifelse( (mwsStatTest$variable == "comp.d13C.North" | mwsStatTest$variable == "comp.d13C.South" &
          mwsStatTest$Date.ti <= as.POSIXct('2016-06-05 00:06:00', tz = 'EST'), "Plateau",
          ifelse( (mwsStatTest$variable == "comp.d13C.North" | mwsStatTest$variable == "comp.d13C.South" &
            mwsStatTest$Date.ti > as.POSIXct('2016-06-05 00:06:00', tz = 'EST'), "Plateau",
            ))))))))

Gr1 <- na.omit(mwsStatTest[, colnames(mwsStatTest) %in% c("value", "Group1")])
Gr2 <- na.omit(mwsStatTest[, colnames(mwsStatTest) %in% c("value", "Group2")])
Gr3 <- na.omit(mwsStatTest[, colnames(mwsStatTest) %in% c("value", "Group3")])

# Test for homogeneity of variance
# Large p-value means no confirmation of homogeneity of variance
bartlett.test(value ~ as.factor(Group3), data = Gr3)

##
## Bartlett test of homogeneity of variances
##
## data: value by as.factor(Group3)
## Bartlett's K-squared = 6.27, df = 5, p-value = 0.2808

# Non-parameteric
# Reject Ho that pop. means are the same if low p-value
res.krs.Grp3 <- kruskal.test(value ~ as.factor(Group3), data = Gr3)

```

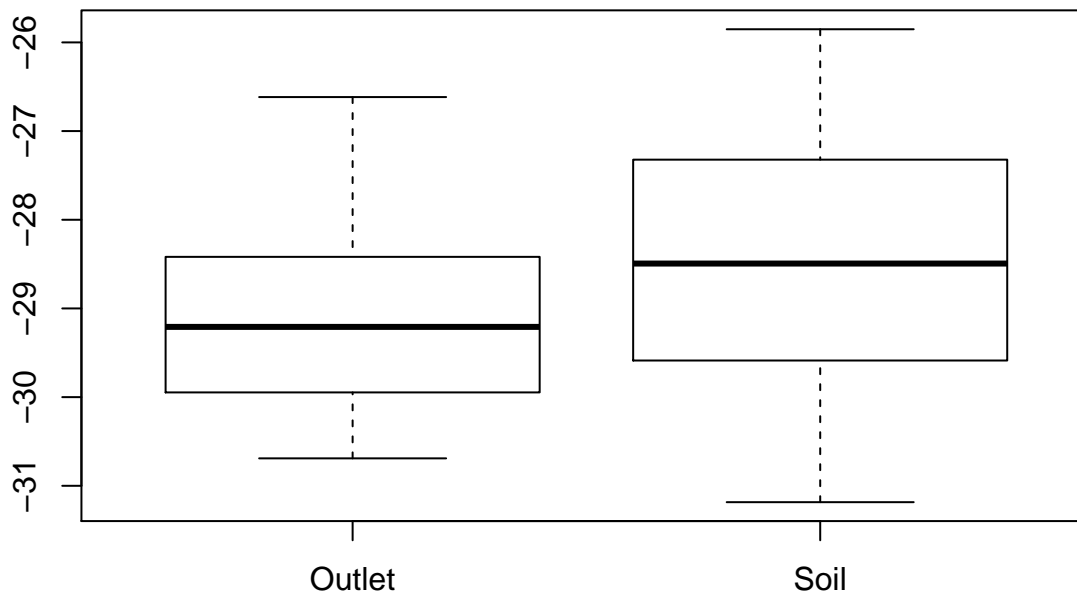
```
res.krs.Grp3

##
## Kruskal-Wallis rank sum test
##
## data: value by as.factor(Group3)
## Kruskal-Wallis chi-squared = 26.435, df = 5, p-value = 7.346e-05
# Want a TukeyHSD function, but this only works with
# parametric data. So, will pass the ranks of the data instead of the actual values
Gr3.ranks <- rank( Gr3$value )
Gr3.groups <- Gr3$Group3
group3.aov <- aov(Gr3.ranks ~ Gr3.groups)
res.grp3 <- TukeyHSD(group3.aov, ordered = T)
aov.res.grp3.df <- as.data.frame(res.grp3$Gr3.groups)
aov.res.grp3.df$P <- round(aov.res.grp3.df$p.adj, 3)
# High p-value indicates no significant difference
write.csv(aov.res.grp3.df, "aovResISOs_ranked.csv", row.names = T)
```

ANOVA and ANOSIM

Not actually used, as Grouping 3 does not have homogeneity of variance

```
# Simple ANOVA tests
# (high p-value indicates lack of difference)
# Big P-value no significant difference
boxplot(Gr1$value ~ Gr1$Group1)
```



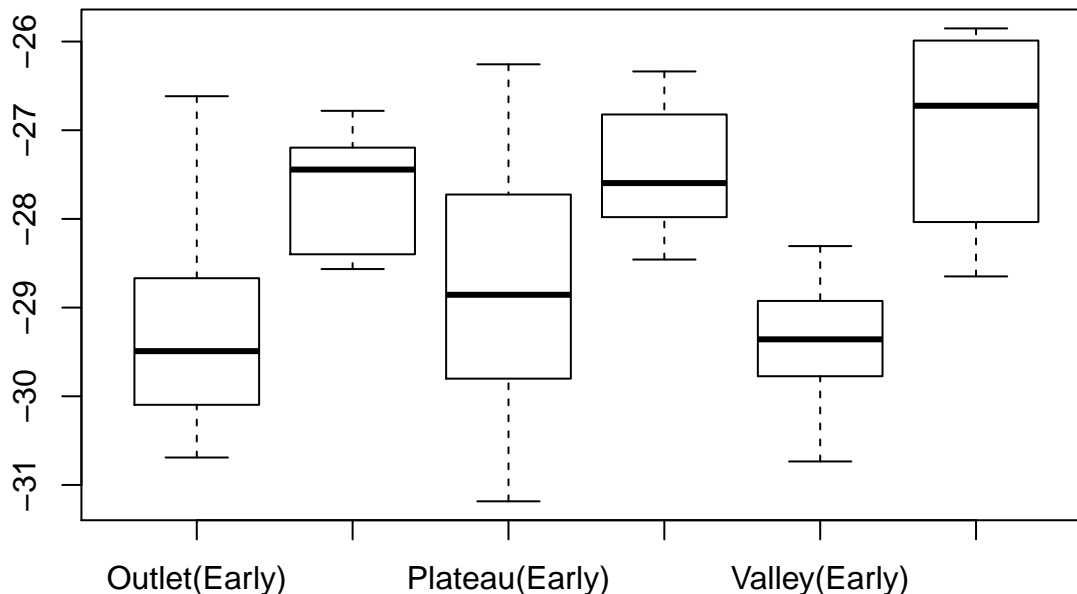
```
summary(aov(Gr1$value ~ Gr1$Group1))

##           Df Sum Sq Mean Sq F value Pr(>F)
## Gr1$Group1  1   5.99   5.995    3.549  0.0639 .
## Residuals  67 113.18   1.689
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
TukeyHSD(aov(Gr1$value ~ Gr1$Group1))
```

```
## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = Gr1$value ~ Gr1$Group1)
##
## $`Gr1$Group1`
##          diff          lwr          upr          p adj
## Soil-Outlet 0.603957 -0.03596571 1.24388 0.0639287
```

```
boxplot(Gr3$value ~ Gr3$Group3)
```



```
group3.aov <- aov(Gr3$value ~ Gr3$Group3)
summary(group3.aov)
```

```
##          Df Sum Sq Mean Sq F value    Pr(>F)
## Gr3$Group3   5  44.94   8.988   7.627 1.18e-05 ***
## Residuals  63  74.24   1.178
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Transform data and compute dissimilarity
Gr1.hell <- decostand(Gr1[, 1], "hellinger", na.rm=T, MARGIN = 2) # Transform/Standardize
Gr1.hell.daisy = daisy(Gr1.hell, "euclidean") # Dissimilarity
attach(Gr1)
anosim.group1 <- anosim(Gr1.hell.daisy, grouping = Group1)
summary(anosim.group1)
```

```
##
## Call:
## anosim(dat = Gr1.hell.daisy, grouping = Group1)
## Dissimilarity:
##
## ANOSIM statistic R: -0.002349
## Significance: 0.462
```

```

##
## Permutation: free
## Number of permutations: 999
##
## Upper quantiles of permutations (null model):
##   90%   95%  97.5%   99%
## 0.0406 0.0566 0.0793 0.1083
##
## Dissimilarity ranks between and within classes:
##      0%   25%   50%   75% 100%   N
## Between 4 581.25 1163.5 1780.75 2342 1134
## Outlet  2 540.00 1076.0 1623.50 2297 351
## Soil    1 619.00 1217.0 1787.00 2346 861

Gr2.hell <- decostand(Gr2[ , 1], "hellinger", na.rm=T, MARGIN = 2) # Transform/Standardize
Gr2.hell.daisy = daisy(Gr2.hell, "euclidean") # Dissimilarity
attach(Gr2)
anosim.group2 <- anosim(Gr2.hell.daisy, grouping = Group2)
summary(anosim.group2)

##
## Call:
## anosim(dat = Gr2.hell.daisy, grouping = Group2)
## Dissimilarity:
##
## ANOSIM statistic R: 0.01931
##      Significance: 0.208
##
## Permutation: free
## Number of permutations: 999
##
## Upper quantiles of permutations (null model):
##   90%   95%  97.5%   99%
## 0.0391 0.0543 0.0694 0.0861
##
## Dissimilarity ranks between and within classes:
##      0%   25%   50%   75% 100%   N
## Between 3 592.25 1179.5 1782.75 2346 1526
## Outlet  2 540.00 1076.0 1623.50 2297 351
## Plateau 1 629.50 1221.0 1766.75 2344 378
## Valley  32 653.00 1183.0 1943.00 2343 91

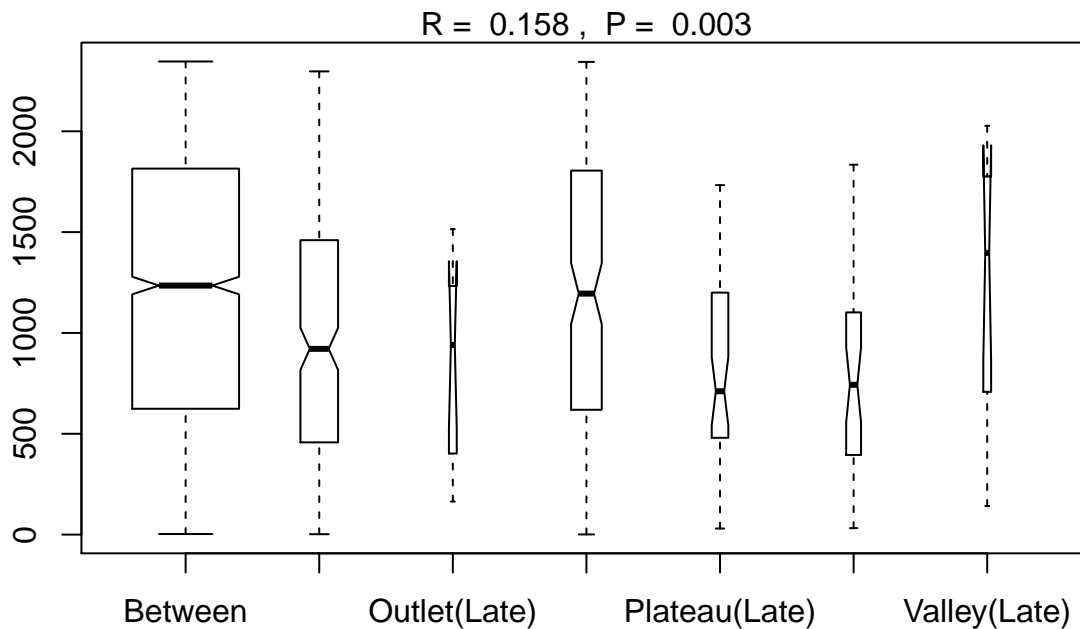
Gr3.hell <- decostand(Gr3[ , 1], "hellinger", na.rm=T, MARGIN = 2) # Transform/Standardize
Gr3.hell.daisy = daisy(Gr3.hell, "euclidean") # Dissimilarity
attach(Gr3)
anosim.group3 <- anosim(Gr3.hell.daisy, grouping = Group3)
summary(anosim.group3)

##
## Call:
## anosim(dat = Gr3.hell.daisy, grouping = Group3)
## Dissimilarity:
##
## ANOSIM statistic R: 0.1577
##      Significance: 0.003

```

```
##
## Permutation: free
## Number of permutations: 999
##
## Upper quantiles of permutations (null model):
##   90%   95%  97.5%   99%
## 0.0484 0.0707 0.0898 0.1143
##
## Dissimilarity ranks between and within classes:
##           0%   25%   50%   75% 100%   N
## Between      3 624.0 1235.0 1815.00 2346 1861
## Outlet(Early) 2 457.5  921.0 1460.00 2297  231
## Outlet(Late) 164 460.0  940.5 1192.00 1515   10
## Plateau(Early) 1 619.0 1195.0 1805.00 2344  153
## Plateau(Late) 30 480.0  711.0 1200.00 1733   45
## Valley(Early) 32 400.0  743.0 1093.00 1834   36
## Valley(Late) 142 740.0 1397.0 1755.75 2027   10
```

```
plot(anosim.group3)
```



Loadings

```
keepLoads <- c("Date.ti",
               "DissOXA.g", "DissESA.g", "DissSmeto.g", "FiltSmeto.g",
               "Event.x", "Events")
wsLoads <- WaterSoils[ , (names(WaterSoils) %in% keepLoads)]

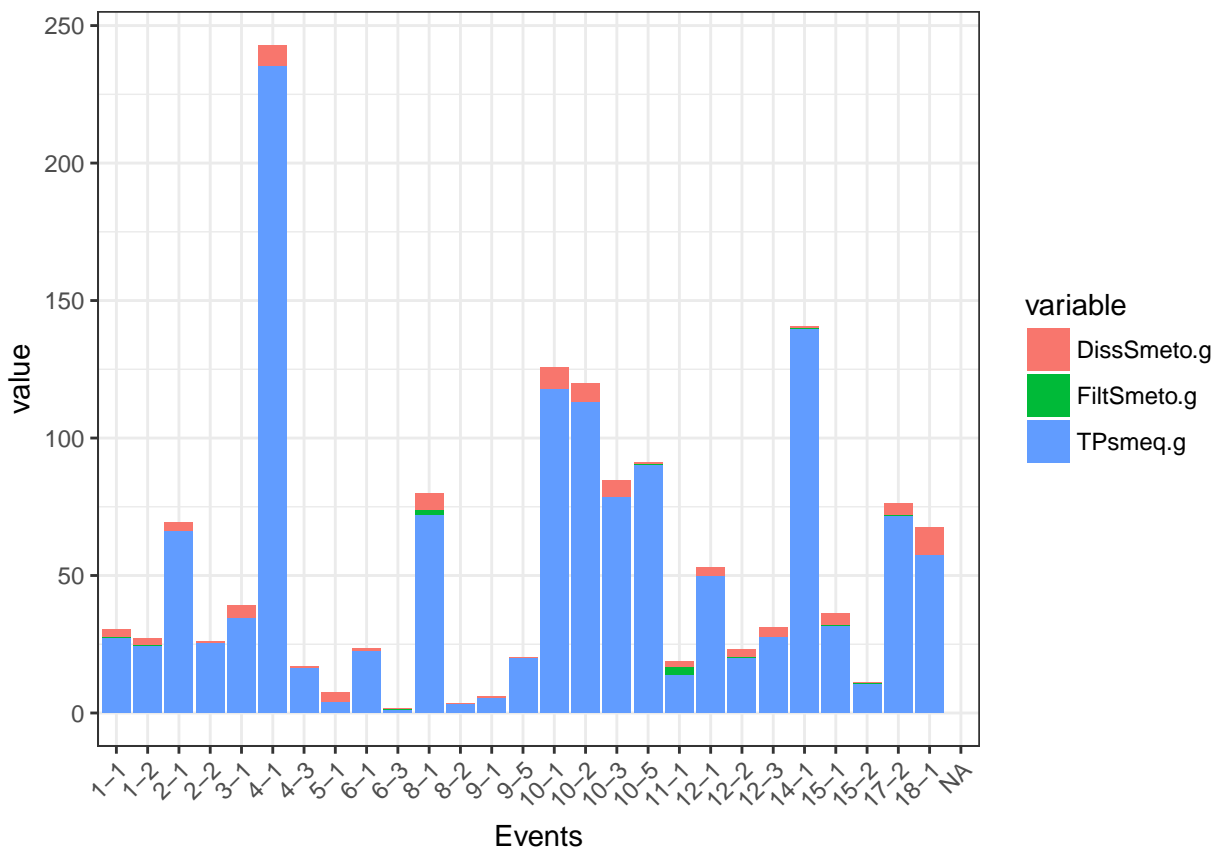
mw.SM <- 283.796 # g/mol
mw.MOXA <- 279.33 # g/ml
mw.MESA <- 329.1 # g/mol
wsLoads$TPsmeq.g <-
  wsLoads$DissOXA.g * (mw.SM/mw.MOXA) +
  wsLoads$DissESA.g * (mw.SM/mw.MESA)
```



```
wsLoads <- wsLoads[ , !(names(wsLoads) %in% c("DissOXA.g", "DissESA.g"))]

loads <- melt(wsLoads, id=c("Date.ti", "Events", "Event.x"))

ggplot(data = loads , aes(x=Events, y=value, fill = variable))+
  theme_bw() +
  geom_bar(stat = "identity") +
  theme(# legend.position="top"
        # axis.title.x = element_blank(),
        axis.text.x=element_text(angle = 45, hjust = 1)
  )
```



```
# geom_bar(stat = "identity", position = position_dodge())

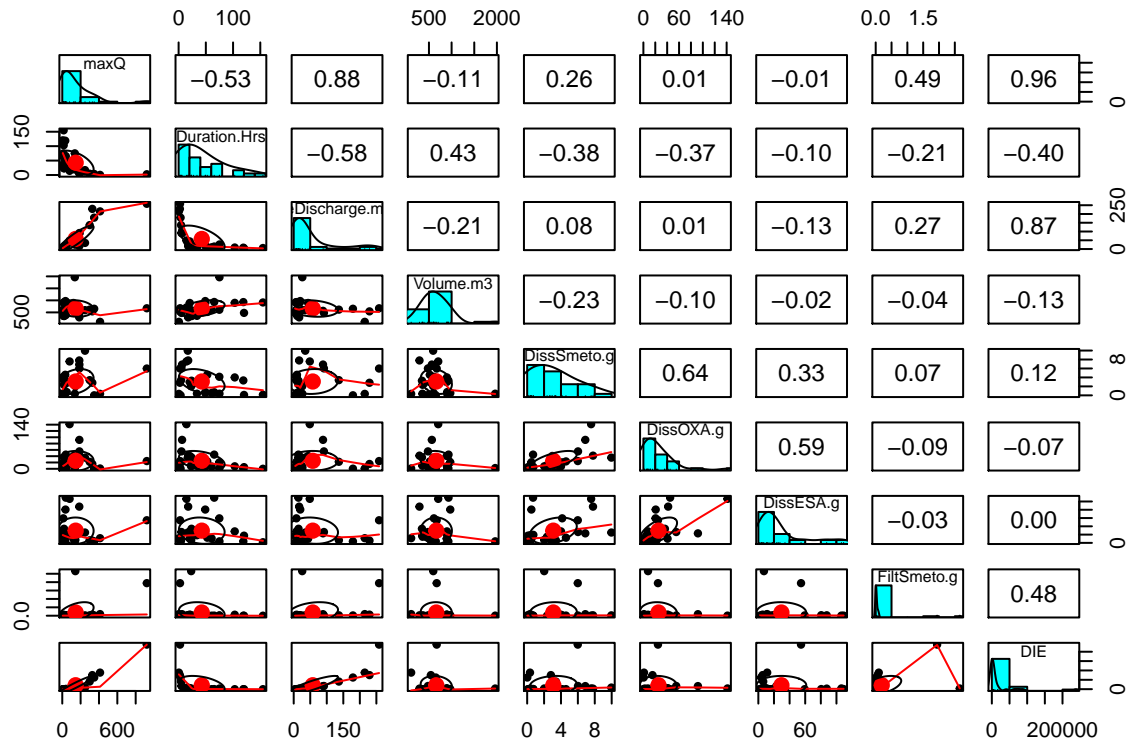
WaterSoils$DIE <- WaterSoils$maxQ*WaterSoils$Volume.m3/WaterSoils$Duration.Hrs

keepCor <- c("maxQ", "Duration.Hrs", "AveDischarge.m3.h", "Volume.m3", "DIE",
             "DissOXA.g", "DissESA.g", "DissSmeto.g", "FiltSmeto.g" #,
             #"NH4.mM", "TIC.ppm.filt", "Cl.mM", "NO3..mM", "PO4..mM", "NPOC.ppm",
             #"TIC.ppm.unfilt", "TOC.ppm.unfilt"
            )

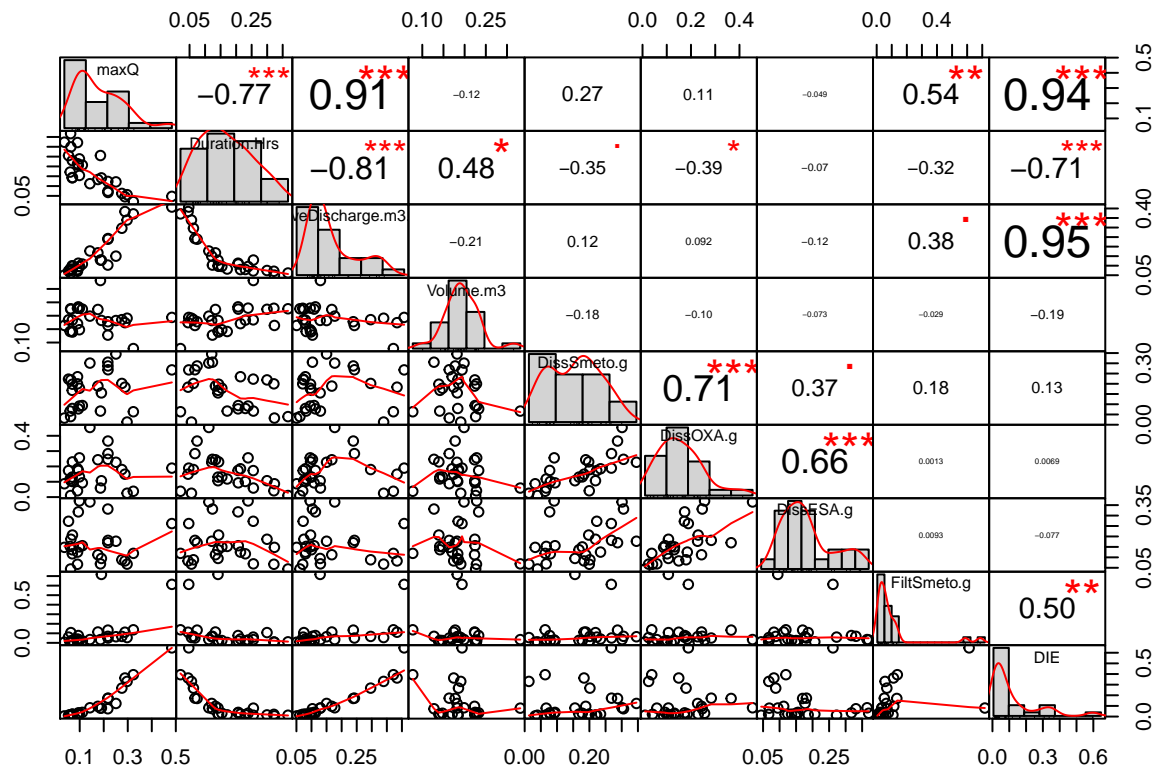
corData <- WaterSoils[ , (names(WaterSoils) %in% keepCor)]

# Transform / normalize
corData.hell <- decostand(corData, "hellinger", na.rm=T, MARGIN = 2)
```

```
library(psych)
pairs.panels(corData)
```



```
library(PerformanceAnalytics)
chart.Correlation(corData.hell)
```



```
keepLoads <- c("Date.ti",
               "DissOXA.g", "DissESA.g", "DissSmeto.g", "FiltSmeto.g",
               "Event.x", "Events")
wsLoads <- WaterSoils[ , (names(WaterSoils) %in% keepLoads)]
```

Outliers

```
# Test function
g_param = 1.5
# g_param = 2.2 # (Hoaglin et al., 1986; Hoaglin & Iglewicz, 1987)
is_outlier <- function(x) {
  return(x < quantile(x, 0.25) - g_param * IQR(x) | x > quantile(x, 0.75) + g_param * IQR(x))
}
```

Soil concentrations

Correlation will be made after variable transformation. Options tested:

- a) Z-scoring transformation by translation and expansion is done to create unit-free variables with means of zero and standard deviations of one. Standardised values differ from one another in units of standard deviation. The mean of each variable is subtracted from the original values and the difference divided by the variable's standard deviation and is given by:

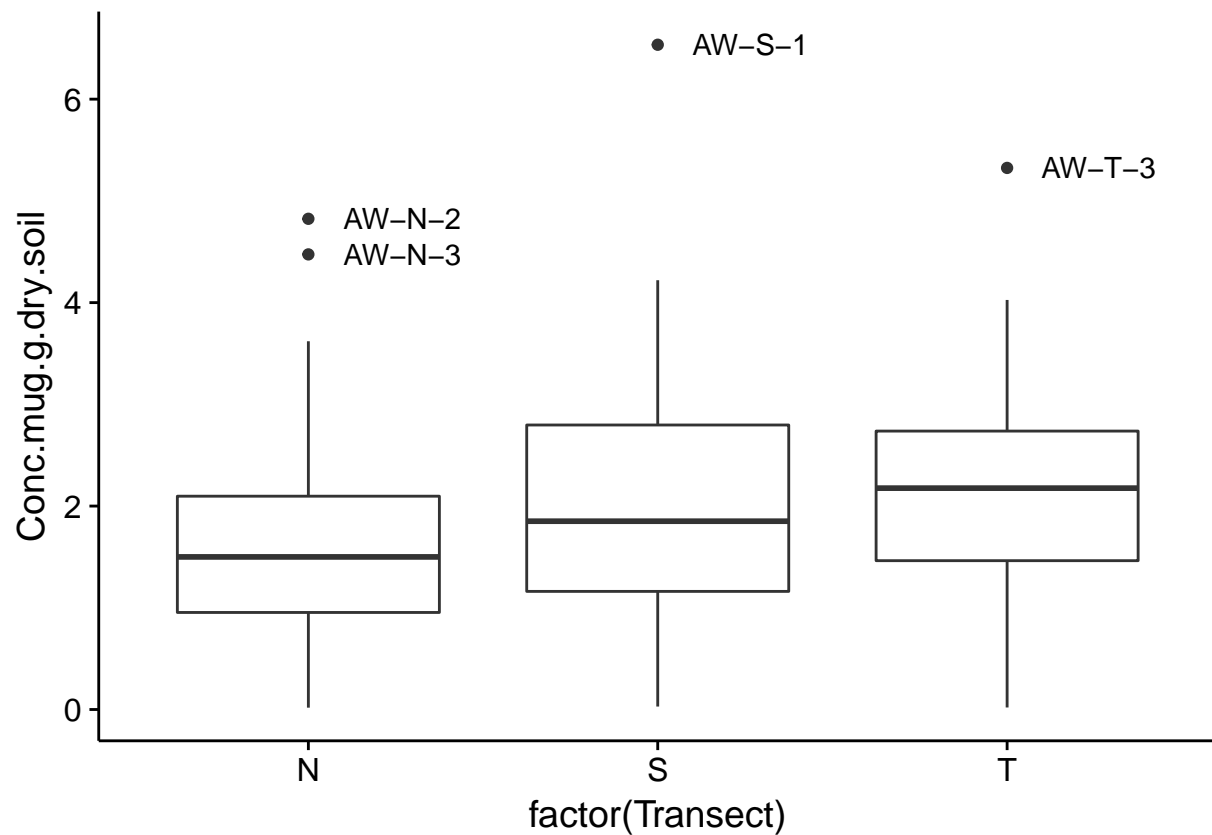
$$z_i = \frac{y_i - \bar{y}}{s_y}$$

Z-scoring did not change correlation results, nor outlier reduction.

- b) Scaling by expansion where all values are divided by the maximum observation.

Outliers before transformation

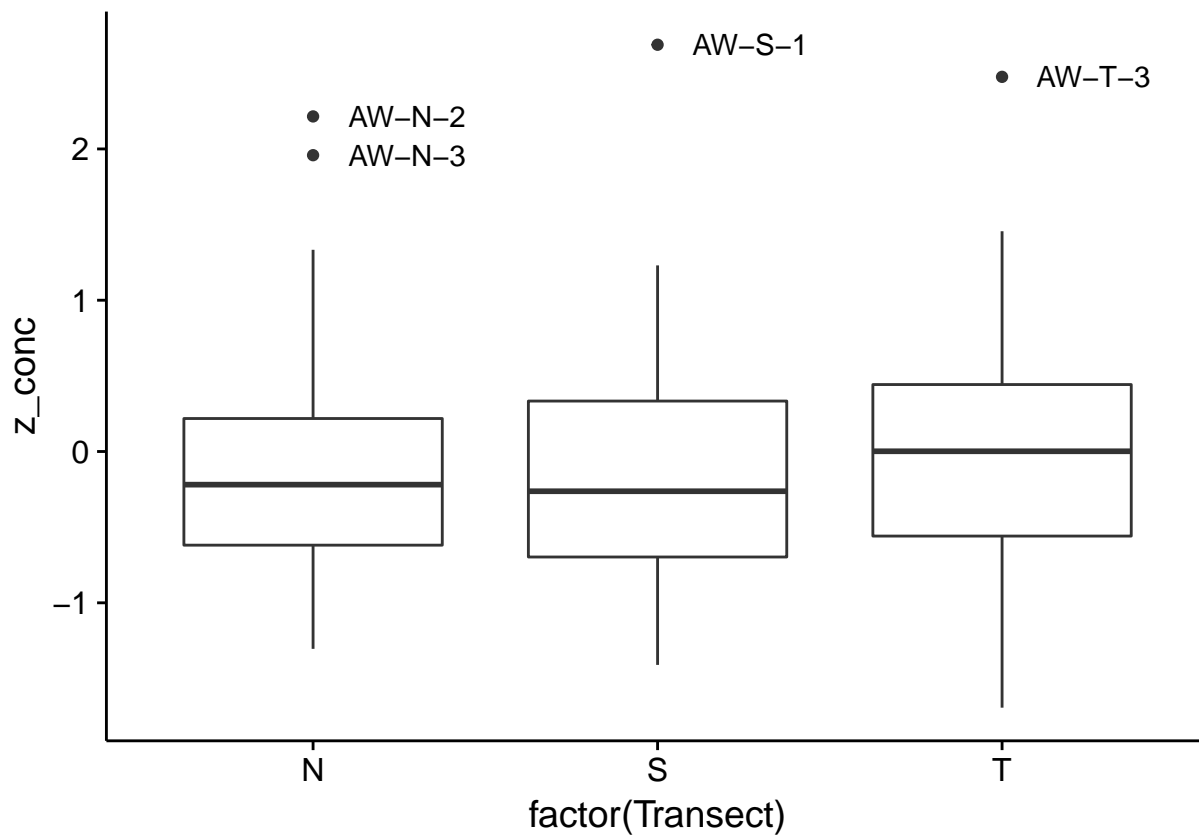
```
# Concentrations
soilGroups %>%
  group_by(Transect) %>%
  mutate(outlier = ifelse(is_outlier(Conc.mug.g.dry.soil), as.character(ID), NA)) %>%
  ggplot(., aes(x = factor(Transect), y = Conc.mug.g.dry.soil)) +
  geom_boxplot() +
  geom_text(aes(label = outlier), na.rm = TRUE, hjust = -0.3)
```



Outliers after transformation

```
soilGroups <- soilGroups %>%
  group_by(Transect) %>%
  mutate(z_conc = (Conc.mug.g.dry.soil - mean(Conc.mug.g.dry.soil)) / sd(Conc.mug.g.dry.soil))

soilGroups %>%
  group_by(Transect) %>%
  mutate(outlier = ifelse(is_outlier(z_conc), as.character(ID), NA)) %>%
  ggplot(., aes(x = factor(Transect), y = z_conc)) +
    geom_boxplot() +
    geom_text(aes(label = outlier), na.rm = TRUE, hjust = -0.3)
```

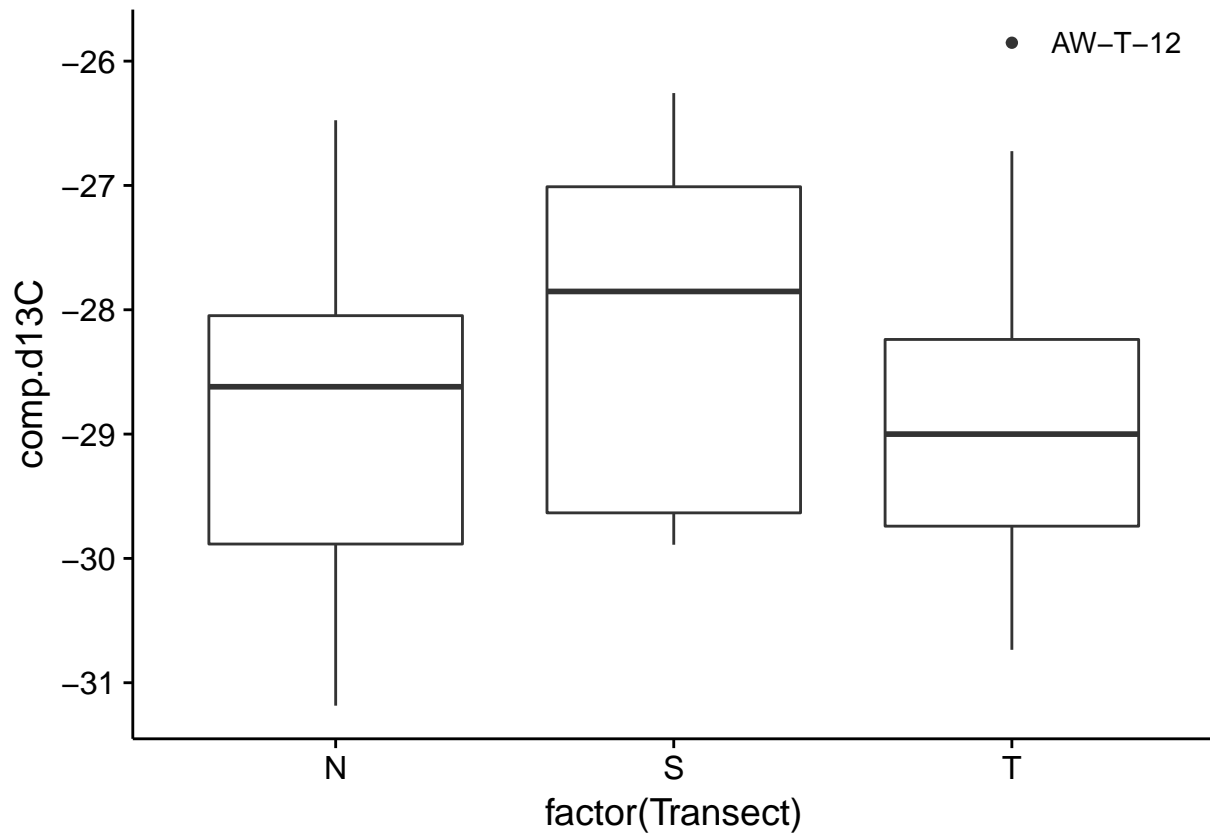


Soil Isotopes

```
# Isotopes

temp <- na.omit(soilGroups)

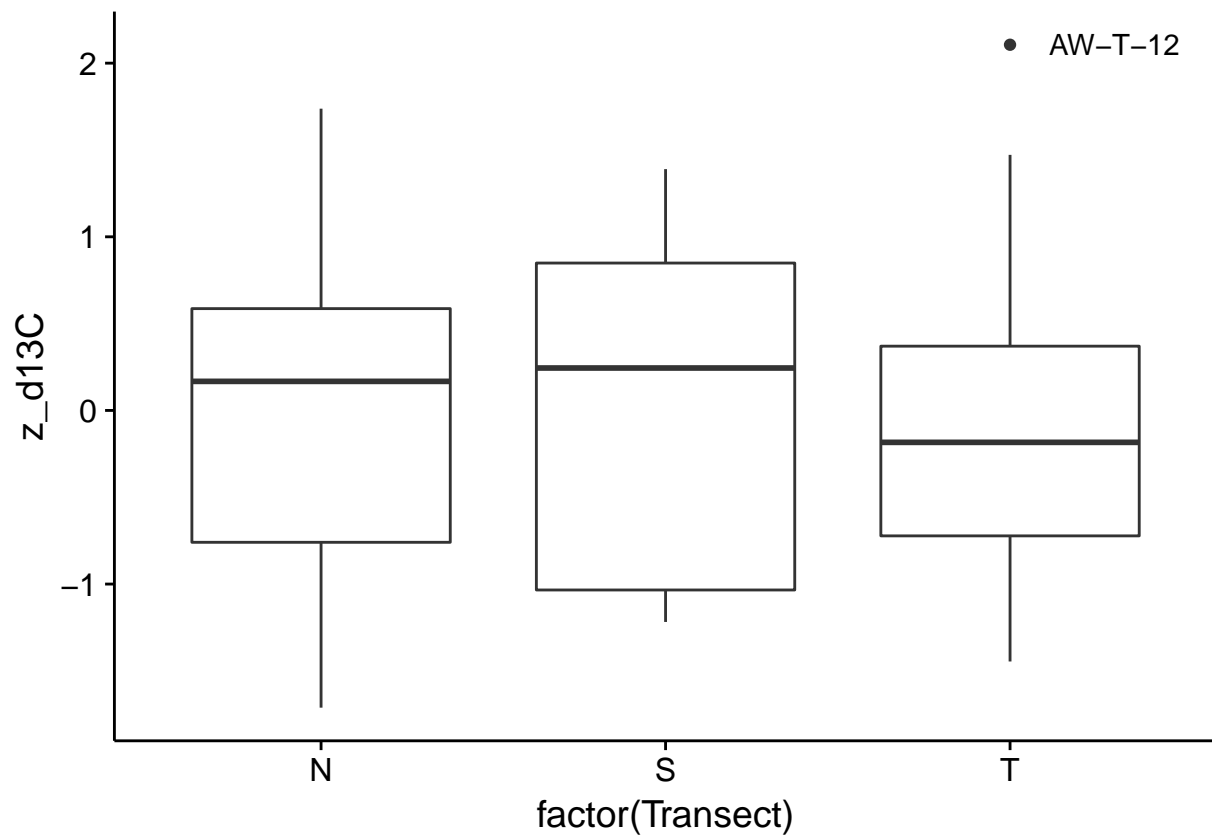
temp %>%
  group_by(Transect) %>%
  mutate(outlier = ifelse(is_outlier(comp.d13C), as.character(ID), NA)) %>%
  ggplot(., aes(x = factor(Transect), y = comp.d13C)) +
    geom_boxplot() +
    geom_text(aes(label = outlier), na.rm = TRUE, hjust = -0.3)
```



Looks like 7 potential outliers in concentrations and 1 for isotopes. Removing NA's for isotopes and re-computing outliers, reduces the number of outliers to 2 in concentrations and 1 for isotopes.

```
temp <- temp %>%
  group_by(Transect) %>%
  mutate(z_d13C = (comp.d13C - mean(comp.d13C)) / sd(comp.d13C))

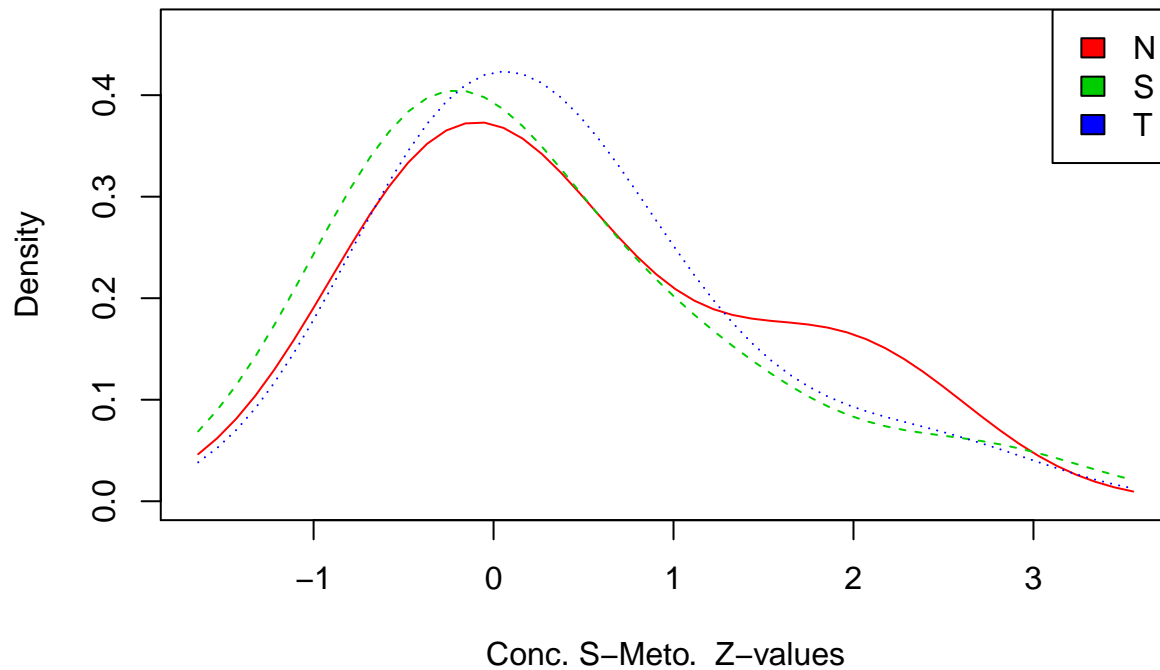
temp %>%
  group_by(Transect) %>%
  mutate(outlier = ifelse(is_outlier(z_d13C), as.character(ID), NA)) %>%
  ggplot(., aes(x = factor(Transect), y = z_d13C)) +
    geom_boxplot() +
    geom_text(aes(label = outlier), na.rm = TRUE, hjust = -0.3)
```



Distribution of z values (same as non-transformed)

```
# plot densities
#sm.density.compare(temp$z_conc, temp$Transect, xlab=expression(paste("Conc. S-Meto. ", {(\mu)*g / g.s
sm.density.compare(temp$z_conc, temp$Transect, xlab=expression(paste("Conc. S-Meto. Z-values")))
title(main="Catchment Soil - Concentrations")
legend("topright", levels( soilGroups$Transect), fill=2+(0:nlevels(soilGroups$Transect)))
```

Catchment Soil – Concentrations



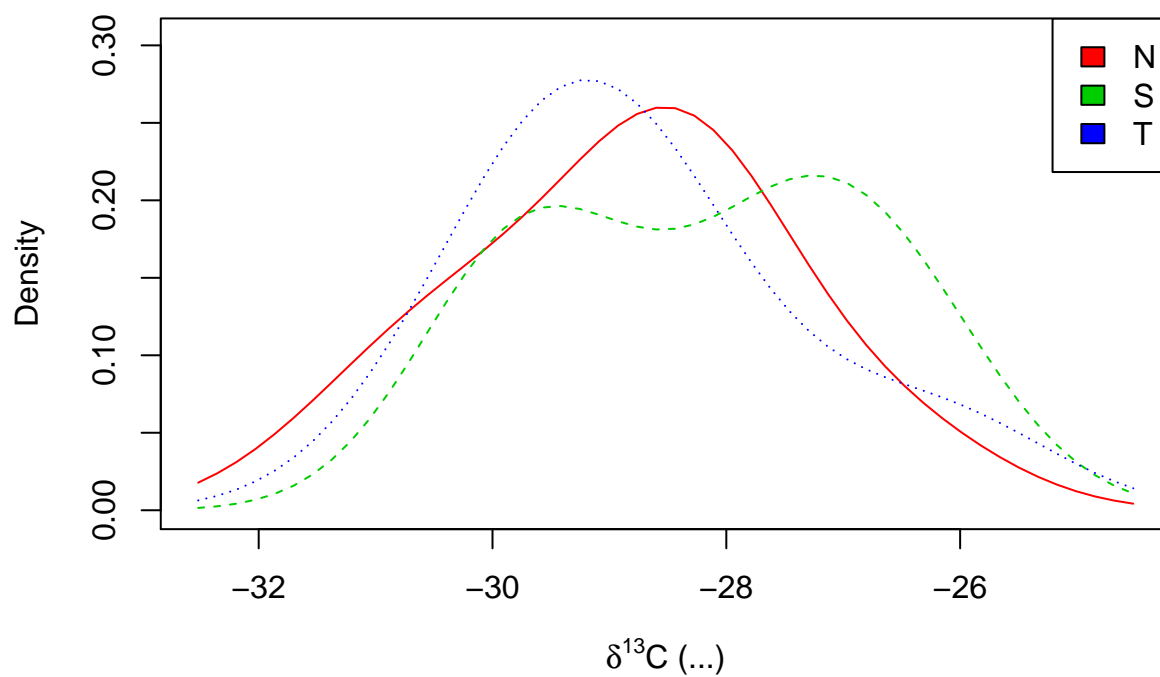
```
#vioplot(soilGroups$Conc.mug.g.dry.soil, names = "Catchment")
#title(expression(paste("Conc. S-Meto. ", {(\mu}*g / g.soil.dry)})))
```

Soil Isotopes

```
#vioplot(na.omit(soilGroups$comp.d13C), names = "Catchment")
#title(expression(paste({\delta}^13, "C", ' (\u2030)')))
```

```
temp <- na.omit(soilGroups)
sm.density.compare(temp$comp.d13C, temp$Transect,
  xlab=expression(paste({\delta}^13, "C", ' (\u2030)'))
title(main="Catchment Soil - Isotope Distribution")
legend("topright", levels( soilGroups$Transect), fill=2+(0:nlevels(soilGroups$Transect)))
```


Catchment Soil – Isotope Distribution



Van Breukelen, Boris M. 2007. “Quantifying the degradation and dilution contribution to natural attenuation of contaminants by means of an open system Rayleigh equation.” *Environ. Sci. Technol.* 41 (14): 4980–5.