Data Screening

PAZ

06/04/2017

Introduction

This Data Screenining 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")

library("plotKML")

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

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

Lab parameters

```
# Initial signature measured in tank
initialDelta = d13Co = -32.253

# 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' missingess 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] -32.253
```

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


dropSoil <- c("WeekSubWeek", # "Event",
```

```r
              "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)]

soilsCheck <- soils[complete.cases(soils[ , "ID.N"]),]

timeApps <- soils[ , c("Date.ti", "timeSinceApp", "timeSinceApp.NoSo", "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
```

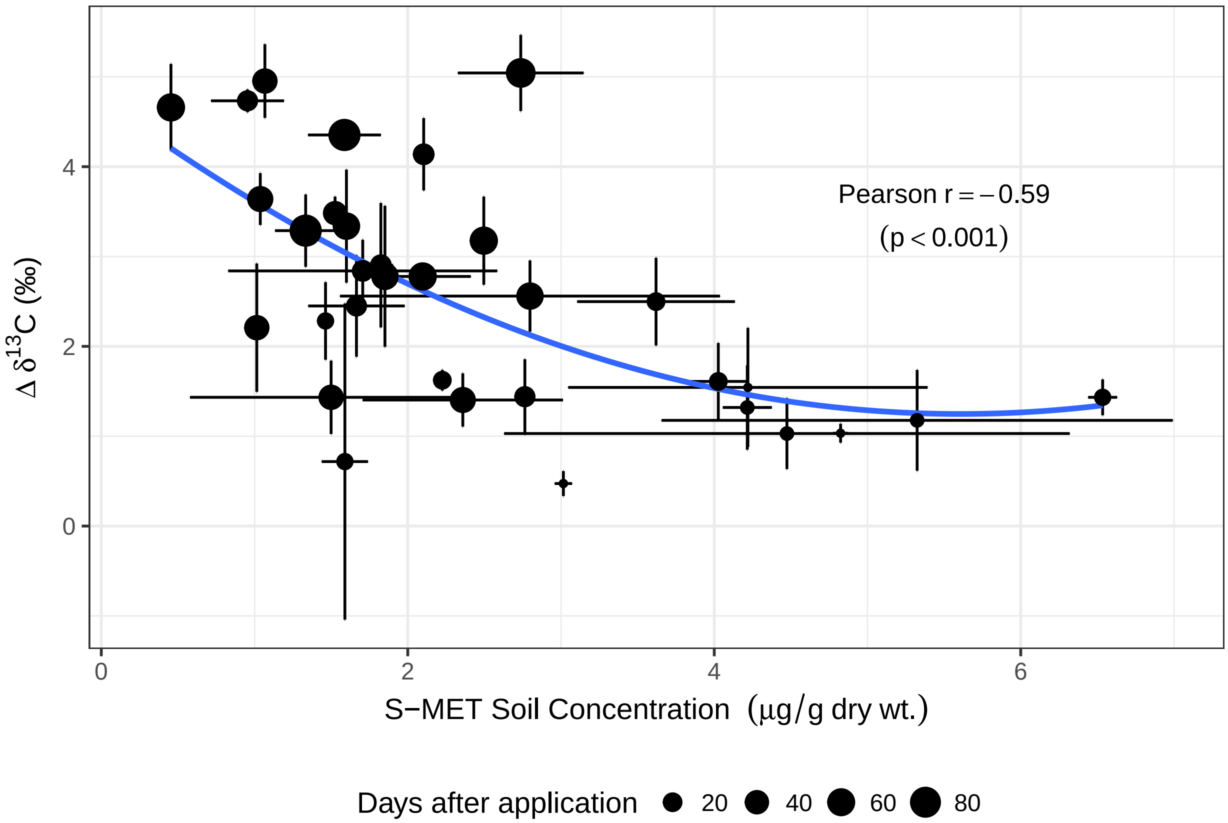
```
## [1] TRUE
```

```r
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 = -4.0218, df = 31, p-value = 0.0003438
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.7733925 -0.3031979
## sample estimates:
##        cor
## -0.5855561
```

```r
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")
```

```r
} 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))+
  geom_errorbar(aes(ymin = DD13C.comp - comp.d13C.SD, ymax = DD13C.comp + comp.d13C.SD)) +
  geom_errorbarh(aes(xmin = Conc.mug.g.dry.soil - Conc.ComSoil.SD, xmax = Conc.mug.g.dry.soil + Conc.Co
  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.NoSo)) + # , colour = Source)) +  # , shape = ngC.Labe
  theme_bw() +
  theme(legend.position = "bottom") +
  labs(size="   Days after application", colour="Source" ) + #, shape = "Mass Carbon") +
  ylab(expression(paste({Delta~delta}^"13","C", ' (\u2030)'))) +
  xlab(expression(paste("S-MET Soil Concentration  ", {({mu}*g / g~dry~wt.)}))) +
  annotate("text", x = 5.5, y = 3.7, label = as.character(r_label), parse = T, size = 3.5) +
  annotate("text", x = 5.5, y = 3.2, label = p_label, parse = T, size = 3.5) +
  scale_size_continuous(range = c(1, 5)) +
  guides(size=guide_legend(nrow=1))

  #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
```




```r
#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)
```

Rayleigh plot (no dilution accounted for)

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

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

```r
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=(N_compsoil >= 3))
cof <- as.numeric(coef(model)[2]*1000)
```

```r
se <- summary(model)$coef[[4]]*1000
lab <- sprintf(" epsilon == %0.2f ", cof)
labSE <- sprintf("\u00B1 %0.2f ", se)
labSE2 <- sprintf("Â± %0.2f ", se)

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

summary(model)
```

```
##
## Call:
## lm(formula = yRaleigh ~ xRaleigh, data = soilGrApp, subset = (N_compsoil >=
##     3))
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.612e-03 -3.042e-04  8.712e-05  4.849e-04  1.352e-03
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0010049  0.0002710   3.708 0.001611 **
## xRaleigh    -0.0012157  0.0002545  -4.776 0.000151 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0007478 on 18 degrees of freedom
##   (14 observations deleted due to missingness)
## Multiple R-squared:  0.5589, Adjusted R-squared:  0.5344
## F-statistic: 22.81 on 1 and 18 DF,  p-value: 0.0001511
```

```r
model2n<-lm(yRaleigh~xRaleigh, data= soilGrApp, subset=(N_compsoil >= 2)) # & ngC.mean >= 5))
summary(model2n)
```

```
##
## Call:
## lm(formula = yRaleigh ~ xRaleigh, data = soilGrApp, subset = (N_compsoil >=
##     2))
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -2.112e-03 -5.323e-04 -5.509e-05  4.540e-04  2.979e-03
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0012709  0.0003503   3.628  0.00102 **
## xRaleigh    -0.0014260  0.0003114  -4.579 7.16e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.001056 on 31 degrees of freedom
##   (14 observations deleted due to missingness)
## Multiple R-squared:  0.4034, Adjusted R-squared:  0.3842
## F-statistic: 20.97 on 1 and 31 DF,  p-value: 7.156e-05
```

```r
# Compre to each transect
modelTalweg<-lm(yRaleigh~xRaleigh, data=soilGrApp, subset=(Wnum < 12 & N_compsoil >= 3 & Transect == "T
eT <- coef(modelTalweg)[2]*1000

modelNorth<-lm(yRaleigh~xRaleigh, data=soilGrApp, subset=(Wnum < 12 & N_compsoil >= 3 & Transect == "N")
eN <- coef(modelNorth)[2]*1000

modelSouth<-lm(yRaleigh~xRaleigh, data=soilGrApp, subset=(Wnum < 12 & N_compsoil >= 3 & Transect == "S")
eS <- coef(modelSouth)[2]*1000

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

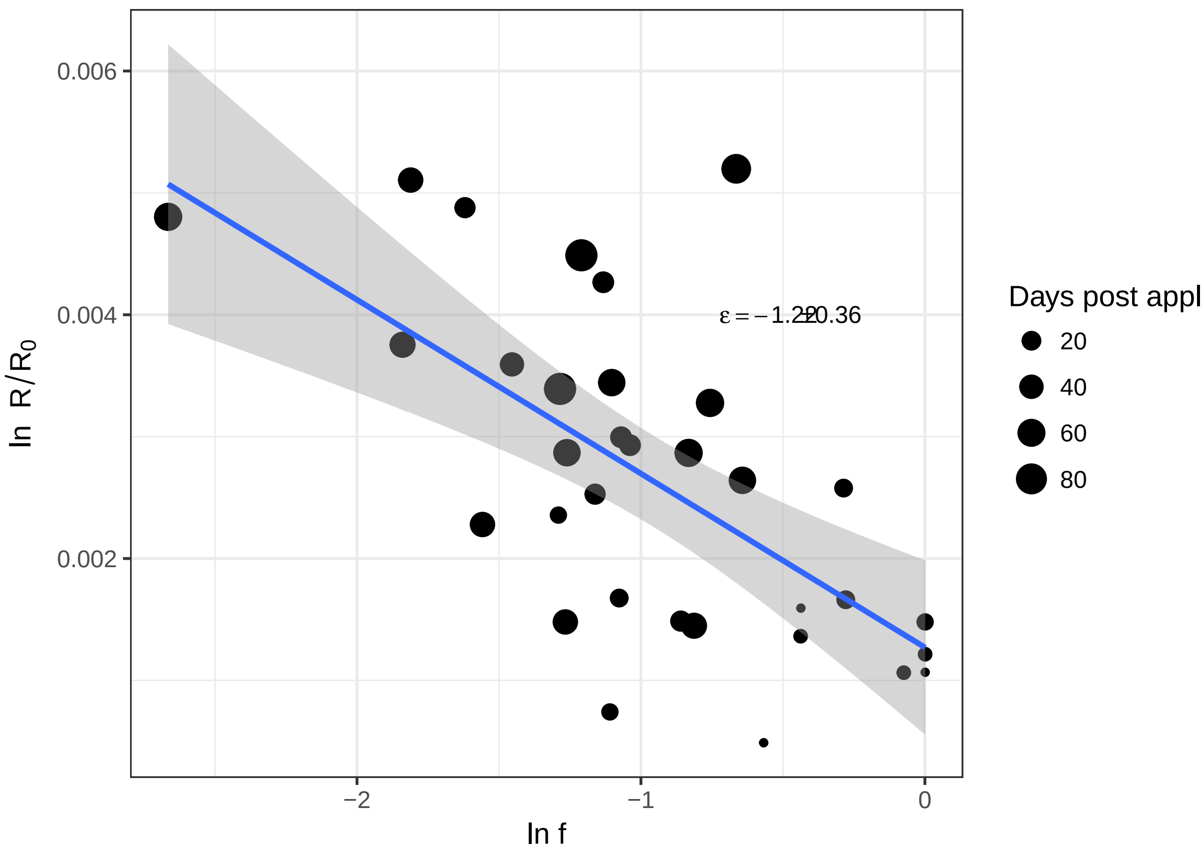
```
## [1] 0.4199896
```

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

```
## [1] -0.9300312
```

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

rayleigh <-
  ggplot(data = subset(soilGrApp, ( Wnum > 0 & N_compsoil >= 2 & !is.na(yRaleigh) )),
         aes(x=xRaleigh, y=yRaleigh)) +
  geom_point(aes(group = ID, size = timeSinceApp.NoSo ))+ #, 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", formula = y~x, se=T) +
  annotate("text", x = -0.55, y = 0.004,
           # label = as.character(expression(paste( "\u0190", " \u2030", " = ", cof))), parse = T, size
           label = lab, parse = T, size = 3.0) +
  annotate("text", x = -0.33, y = 0.004,
           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=Wnu
  #               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,
```

```r
#         nudge_x = .2)

rayleigh
```




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

Accounting for dilution

The Rayleigh equation above assumes that f reflects solely reduction in concentrations due to degradation and should thus be expressed as $f_{degradation}$. Accounting for dilution processes, the remaining fraction that is measured in the field sample becomes then f_{total}, where:

$$f_{total} = f_{degradation} \cdot f_{dilution}$$

Following Van Breukelen (2007),

$$f_{degradation} = f_{total} \cdot F$$

where the dilution factor F (i.e. the number of times the source volume has become diluted at the observation location) can be calculated if ϵ_{lab} is known:

$$F = e^{(\Delta/\epsilon_{lab}) - ln f_{total}}$$

```r
soilGrApp$Fdil =
  exp( (log((1000+d13Co+soilGrApp$DD13C.comp)/(1000+d13Co))/epsilon_lab) -
         log(soilGrApp$Conc.mug.g.dry.soil/soilGrApp$iniCo)  )

soilGrApp$Fdil <- ifelse(soilGrApp$Fdil < 1, NA, soilGrApp$Fdil)
```

We can now obtain $f_{dilution}$ and $f_{degradation}$:

```r
soilGrApp$fdil <- 1/soilGrApp$Fdil
soilGrApp$ftot <- soilGrApp$Conc.mug.g.dry.soil/soilGrApp$iniCo
soilGrApp$fdeg <- soilGrApp$ftot * soilGrApp$Fdil
```

The relationship $D * /B*$ can be obtained by:

```r
DBmodel<-lm(log(fdeg)~log(fdil), data= soilGrApp, subset=(!is.na(fdil)))
cof_DB <- as.numeric(coef(DBmodel)[2]*1000)
se_DB <- summary(DBmodel)$coef[[4]]*1000
summary(DBmodel)
```

```
##
## Call:
## lm(formula = log(fdeg) ~ log(fdil), data = soilGrApp, subset = (!is.na(fdil)))
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.701e-03 -3.208e-04  4.286e-05  4.009e-04  1.208e-03
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.0007328  0.0002572  -2.849 0.008128 **
## log(fdil)    0.0008108  0.0002183   3.714 0.000899 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0006347 on 28 degrees of freedom
## Multiple R-squared:  0.3301, Adjusted R-squared:  0.3061
## F-statistic:  13.8 on 1 and 28 DF,  p-value: 0.0008991
```

Alternatively, Van Breukelen gives the following equation to estimate the field enrichment:

$$\epsilon_{filed} = B^{*} \cdot \epsilon_{true} = \frac{\Delta}{ln f_{total}}$$

Import waters

Compare mass balance, theoretical and CSIA

```r
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))
```

```r
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
```

```r
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"                   "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"               "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"            "Appl.Mass.g.NoSo"
## [51] "timeSinceApp.NoSo"       "CumAppMass.g"
## [53] "DissSmeto.g"             "DissSmeto.g.SD"
## [55] "DissOXA.g"               "DissOXA.g.SD"
## [57] "DissESA.g"               "DissESA.g.SD"
## [59] "FiltSmeto.mg.SD"         "FiltSmeto.g"
## [61] "FiltSmeto.g.SD"          "TotSMout.g"
## [63] "TotSMout.g.SD"           "MELsm.g"
## [65] "MELsm.g.SD"              "CumOutDiss.g"
## [67] "CumOutFilt.g"            "CumOutSmeto.g"
## [69] "CumOutMELsm.g"           "BalMassDisch.g"
## [71] "prctMassOut"             "FracDeltaOut"
## [73] "Events"                  "Weeks"
## [75] "Event"                   "No_First"
## [77] "No_Second"               "No_Third"
## [79] "CumDays_First"           "CumDays_Second"
## [81] "CumDays_Third"           "remain_1st_29d"
## [83] "remain_2nd_29d"          "remain_3rd_29d"
## [85] "remain_1st_46d"          "remain_2nd_46d"
## [87] "remain_3rd_46d"          "remain_1st_12d"
## [89] "remain_2nd_12d"          "remain_3rd_12d"
## [91] "remainMedTheo_prc"       "remainMinTheo_prc"
## [93] "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)]

# Get cummualtive SD
library("TTR")
waters$CumOutSmeto.g.SD <- runSD(waters$TotSMout.g.SD, n=1, cumulative=TRUE)
waters$CumOutMELsm.g.SD <- runSD(waters$MELsm.g.SD, n=1, cumulative=TRUE)


keepWaterMB <- c("Date.ti", "CumAppMass.g",
                 # MB
                 "CumOutSmeto.g", "CumOutMELsm.g" ,
                 "CumOutSmeto.g.SD", "CumOutMELsm.g.SD",
                 "remainMedTheo_prc", "remainMinTheo_prc", "remainMaxTheo_prc",
                 # CSIA
                 "B.diss", "SD.d13C")

watersMassBal <- waters[ , (names(waters) %in% keepWaterMB)]

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

watersMassBal$SMout_prc <- (watersMassBal$CumOutSmeto.g/watersMassBal$CumAppMass.g)*100
watersMassBal$SMout.SD1 <- watersMassBal$SMout_prc + (watersMassBal$CumOutSmeto.g.SD/watersMassBal$CumAp
watersMassBal$SMout.SD2 <- watersMassBal$SMout_prc - (watersMassBal$CumOutSmeto.g.SD/watersMassBal$CumAp

watersMassBal$TPout_prc <- (watersMassBal$CumOutMELsm.g/watersMassBal$CumAppMass.g)*100
watersMassBal$TPout.SD1 <- watersMassBal$TPout_prc + (watersMassBal$CumOutMELsm.g.SD/watersMassBal$CumAp
watersMassBal$TPout.SD2 <- watersMassBal$TPout_prc - (watersMassBal$CumOutMELsm.g.SD/watersMassBal$CumAp

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)
```

```r
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$B.diss <- NULL
bal$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] "CumOutSmeto.g.SD"  "CumOutMELsm.g.SD"  "SMout_prc"
##  [7] "SMout.SD1"         "SMout.SD2"         "TPout_prc"
## [10] "TPout.SD1"         "TPout.SD2"         "B.mean"
## [13] "B.sd1"             "B.sd2"             "f.mean"
## [16] "f.sd1"             "f.sd2"             "DegMed"
## [19] "DegLow"            "DegHigh"           "Month"
```

```r
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", "Predicted (half-life)", "CSIA", "Predicted (half-life)", "Mass Balance", "Mass Ba

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"                  "Mass Balance"          "Predicted (half-life)"
```

```
balTidyType$type <- factor(balTidyType$type, levels = c("CSIA" , "Mass Balance", "Predicted (half-life)
levels(balTidyType$Sink)
```

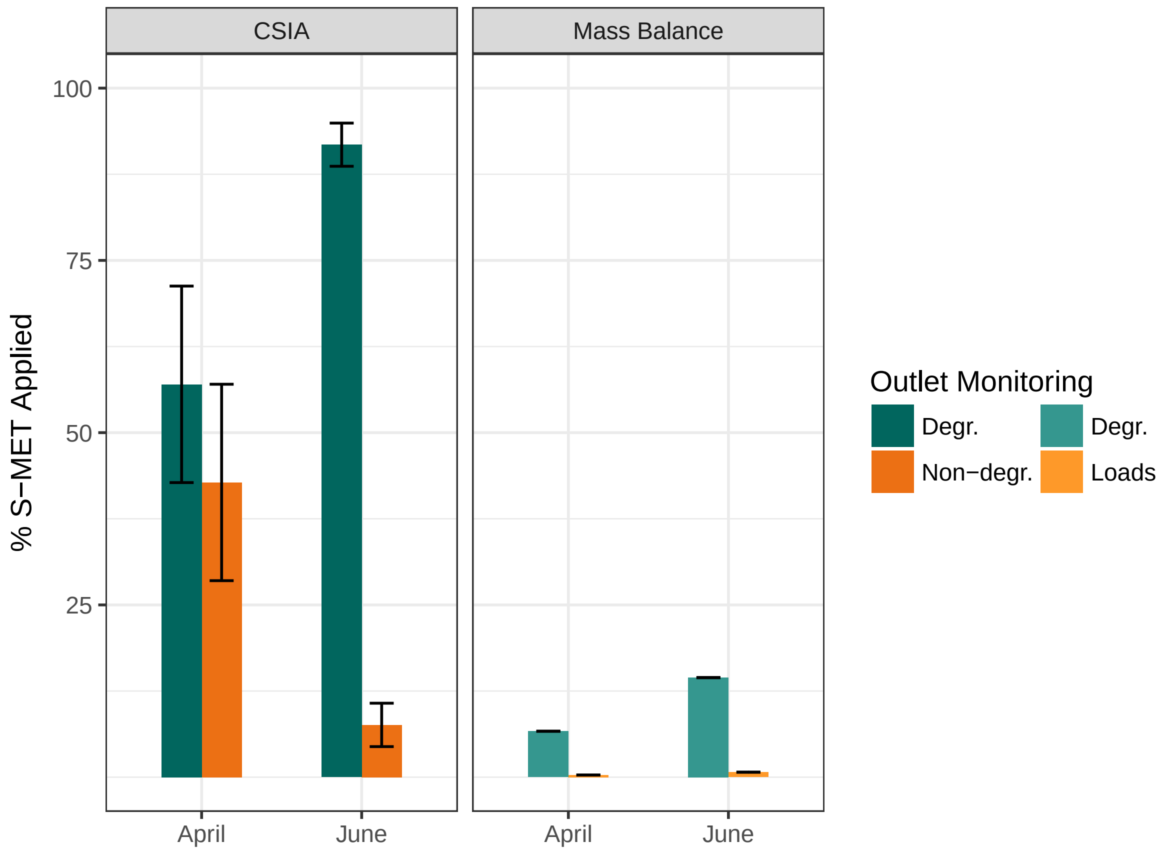
```
## [1] "B"     "TPout" "Deg"   "f"     "SMout" "Rem"
```

```
levels(balTidyType$Month)
```

```
## [1] "April" "June"
```

```
OutBars <- ggplot(data = subset(balTidyType, type != 'Predicted (half-life)') ,
                  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 Applied") +
  theme(axis.title.x = element_blank() ) +
  scale_y_continuous( breaks=c(25, 50, 75, 100), limits = c(0, 100) )+ #expand=c(0, 10, 0, 0)) +
  #xlab("Month") +
  facet_wrap(~type) +
  scale_fill_manual(#values = c("#01665e",  "#ec7014",   "#35978f",  "#fe9929", "#80cdc1", "#fec44f"), 
                    values = c("#01665e",  "#35978f", "#ec7014" ,  "#fe9929"), # blue-orange
                    #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
```

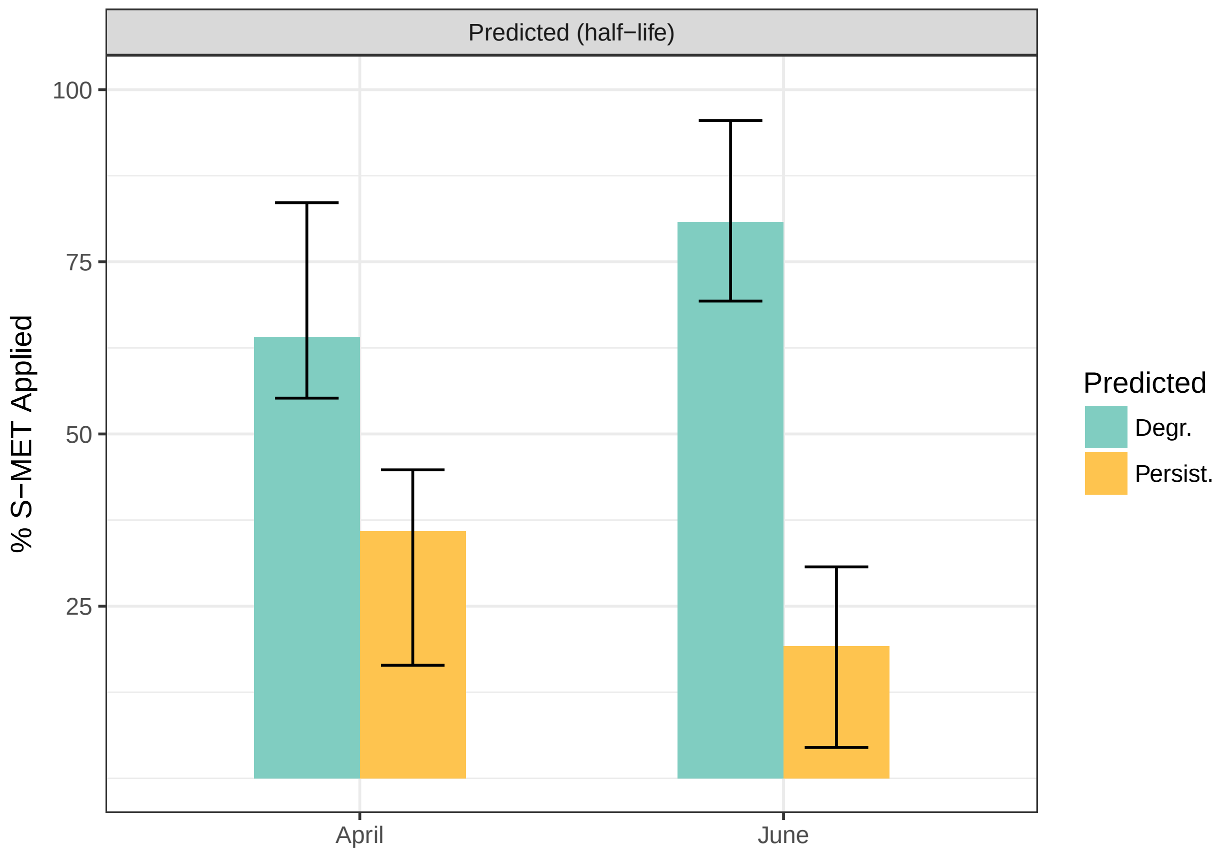



```
#pal = c("#01665e",  "#ec7014",   "#35978f",  "#fe9929", "#80cdc1", "#fec44f")
#display.pal(pal, sel=1:length(pal), names=F)
```

Predicted (half-life) only

```
theoBars <- ggplot(data = subset(balTidyType, type == 'Predicted (half-life)') ,
                   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 Applied") +
  theme(axis.title.x = element_blank() ) +
  scale_y_continuous( breaks=c(25, 50, 75, 100), limits = c(0, 100) )+ #expand=c(0, 10, 0, 0)) +
```

```r
  #xlab("Month") +
  facet_wrap(~type) +
  scale_fill_manual(values = c( "#80cdc1", "#fec44f"), # blue-orange
                    #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= "Predicted",# element_blank(), #"Mass Balance", # \n
                       breaks=c("Deg", "Rem"
                                ),
                       labels=c("Degr." , "Persist."
                                 )) +
  guides(fill=guide_legend(ncol=1))

#pal = c( "#35978f", "#fe9929", "#fec44f", "#01665e", "#80cdc1", "#ec7014" )
#display.pal(pal, sel=1:length(pal), names=F)

theoBars
```



Compare to Catchment soils

```r
names(soils)
```

```
##  [1] "Date.ti"             "Event"               "timeSinceApp"
##  [4] "timeSinceApp.NoSo"   "diss.d13C"           "SD.d13C"
##  [7] "CumAppMass.g"        "B.mean.comp.North"   "B.max.comp.North"
```

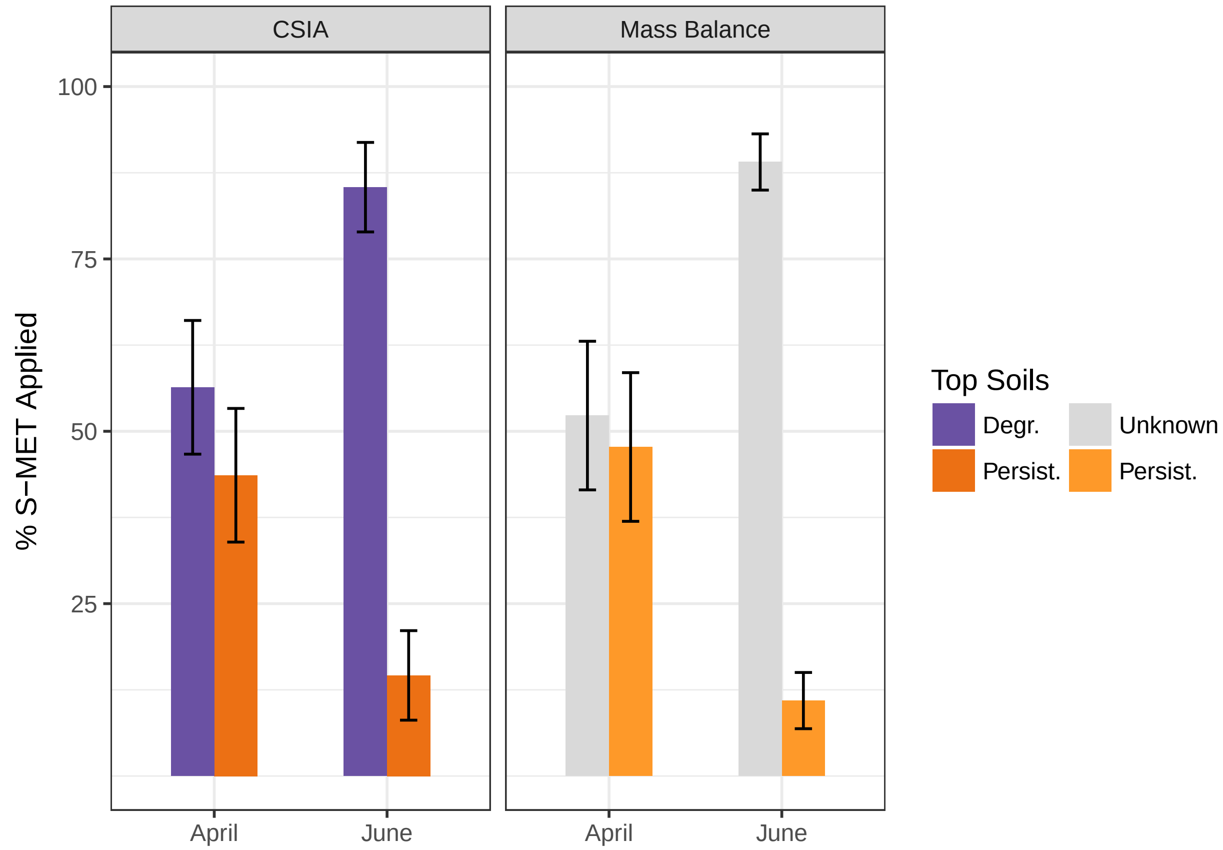
```
## [10] "B.min.comp.North"    "MassSoil.g.North"    "comp.d13C.North"
## [13] "comp.d13C.SD.North"  "ID.N"                "B.mean.comp.Talweg"
## [16] "B.max.comp.Talweg"   "B.min.comp.Talweg"   "MassSoil.g.Talweg"
## [19] "comp.d13C.Talweg"    "comp.d13C.SD.Talweg" "B.mean.comp.South"
## [22] "B.max.comp.South"    "B.min.comp.South"    "MassSoil.g.South"
## [25] "comp.d13C.South"     "comp.d13C.SD.South"  "ID.S"
## [28] "CatchMassSoil.g"     "BulkMass.g"          "BulkCatch.d13"
## [31] "BulkCatch.d13.SD"    "f.mean.bulk"         "B.mean.bulk"
## [34] "DD13C.North"         "DD13C.Talweg"        "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)
```

```r
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", "Mass Balance", "Mass Balance"), 2)

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

levels(balTidySol$Sink)
```

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

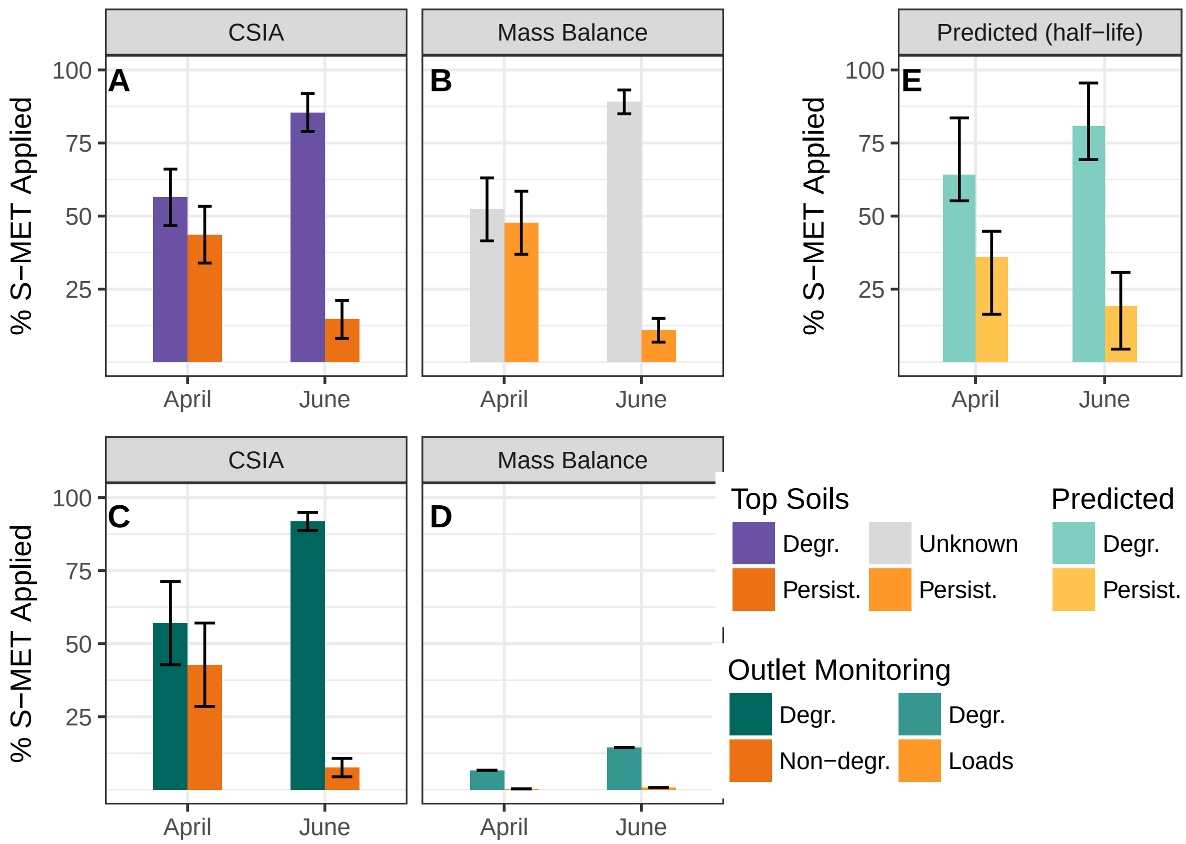
```r
balTidySol$Sink <- factor(balTidySol$Sink, levels = c("B" , "f" ,"Unk", "Rem"))

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 Applied") +
  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)

TheoBars_noLeg <- theoBars + theme(legend.position = 'none')
TheoBars_Leg <- get_legend(theoBars)

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

balAllplot <- ggdraw() +
  draw_plot(SoilBars_noLeg, x=0.01, y = 0.5, width = 0.60, height = 0.5) +
  draw_plot(TheoBars_noLeg, x=0.65, y=.5, width =  0.33, height = .5) +
  draw_plot(OutBars_noLeg, x=0.01, y=.0, width =  0.60, height = .5) +
  draw_plot(SoilBars_Leg, x=0.67, y = 0.3, width = 0.1, height = 0.1) +
  draw_plot(OutBars_Leg, x=0.69, y = 0.1, width = 0.1, height = 0.1) +
```

```
  draw_plot(TheoBars_Leg, x=0.89, y = 0.3, width = 0.05, height = 0.1) +
  draw_label("A", x= 0.11, y = .9, size = 12, fontface = "bold") +
  draw_label("C", x= 0.11, y = .39, size = 12, fontface = "bold") +
  draw_label("B", x= 0.37, y = .9, size = 12, fontface = "bold") +
  draw_label("D", x= 0.37, y = .39, size = 12, fontface = "bold") +
  draw_label("E", x= 0.75, 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
```

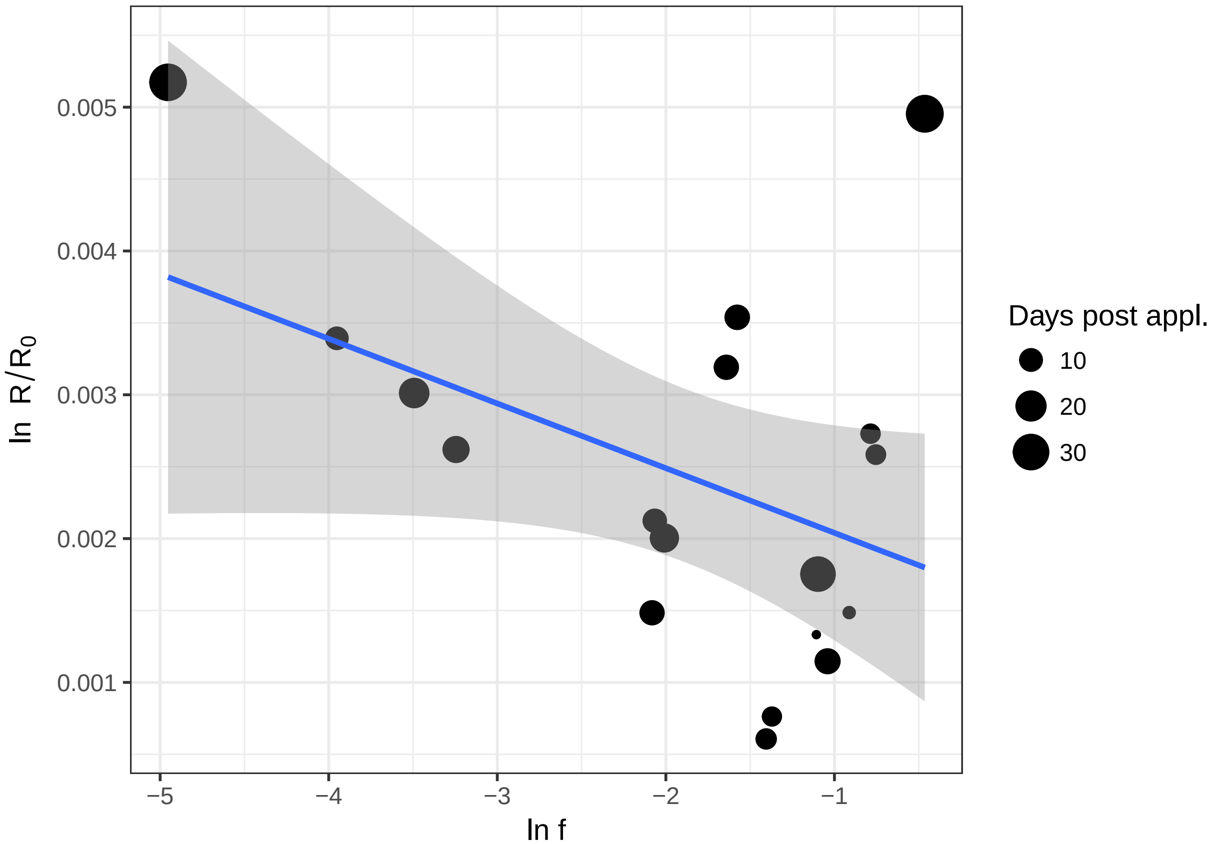
```
## [1] -32.253
```

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

```r
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
## -0.0021282 -0.0011833 -0.0002925  0.0009212  0.0029923
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0021157  0.0004968   4.259 0.000254 ***
## xRaleigh    -0.0002454  0.0001713  -1.433 0.164283
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.001507 on 25 degrees of freedom
##   (11 observations deleted due to missingness)
## Multiple R-squared:  0.07589,    Adjusted R-squared:  0.03893
## F-statistic: 2.053 on 1 and 25 DF,  p-value: 0.1643
```

```r
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
```




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

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

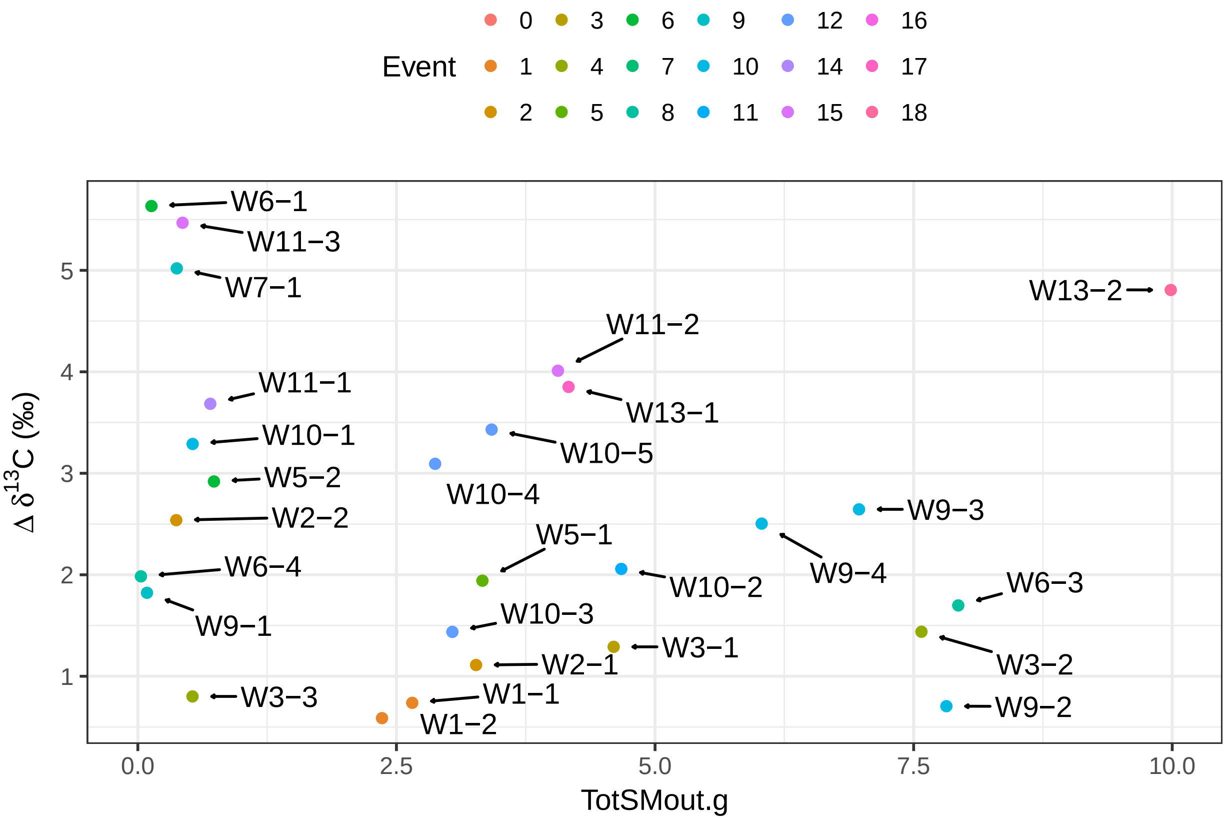
```r
str(waters)
```

```
## 'data.frame':    51 obs. of  83 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 ...
```

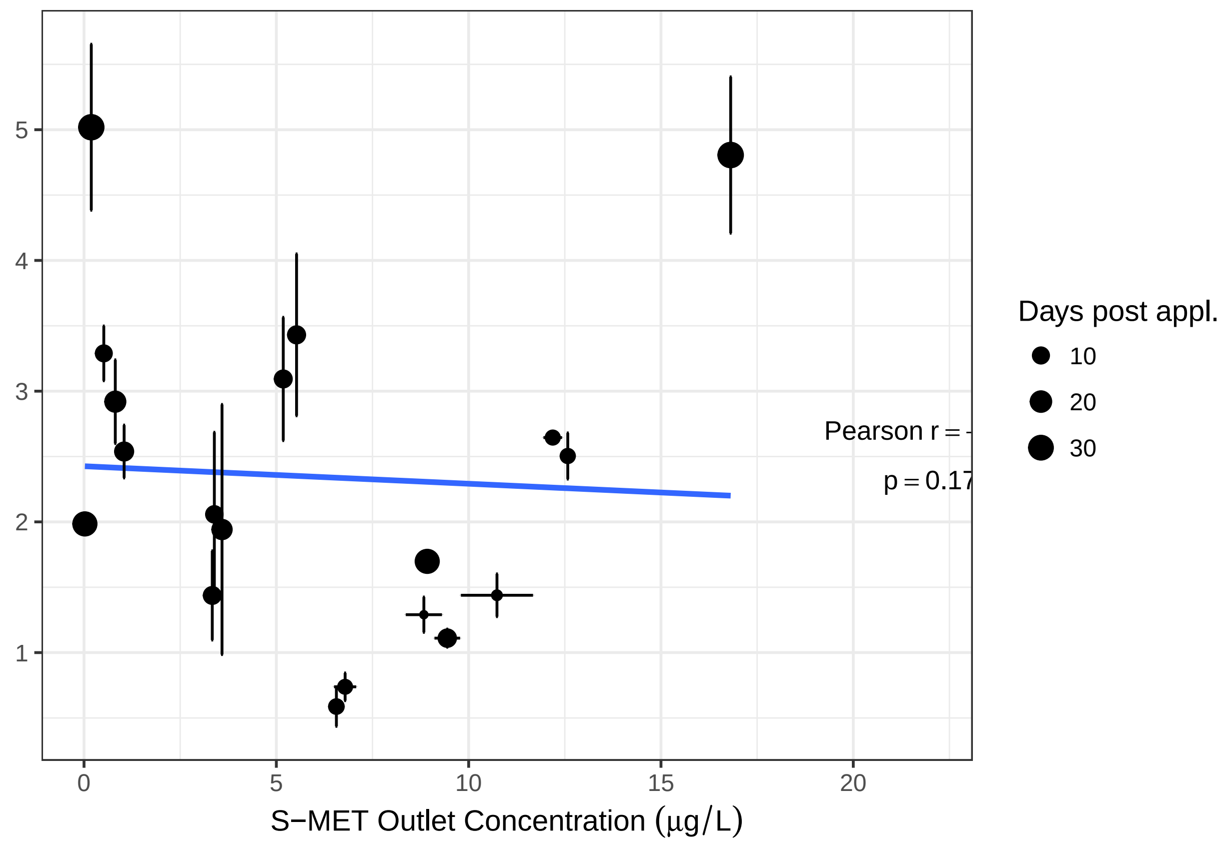
```
## $ OXA_mean              : num  4.82 4.82 17.68 30.53 32.49 ...
## $ OXA_SD                : num  1.141 1.141 5.663 10.185 0.243 ...
## $ ESA_mean              : num  18.1 18.1 32 46 41.3 ...
## $ ESA_SD                : num  3.497 3.497 3.267 3.037 0.853 ...
## $ diss.d13C             : num  NA NA NA -31.5 -31.7 ...
## $ 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 0.738 0.587 ...
## $ DD13C.filt            : num  NA NA NA NA NA ...
## $ B.diss                : num  NA NA NA 35.3 29.3 ...
## $ 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 ...
## $ Appl.Mass.g.NoSo      : num  9498 0 0 0 0 ...
## $ timeSinceApp.NoSo     : 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 -0.595 -0.532 ...
```

```
##  $ 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 ...
##  $ CumOutSmeto.g.SD      : num  NA 0.000817 0.002656 0.038861 0.035716 ...
##  $ CumOutMELsm.g.SD      : num  NA 0.1131 0.0901 1.1401 1.016 ...
##  $ yRaleigh              : num  NA NA NA 0.000763 0.000607 ...
##  $ xRaleigh              : num  -4.69 -4.69 -2.03 -1.37 -1.4 ...
##  $ DIa                   : num  1.5 1.68 2.1 549.76 289.92 ...
```

```r
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)
```



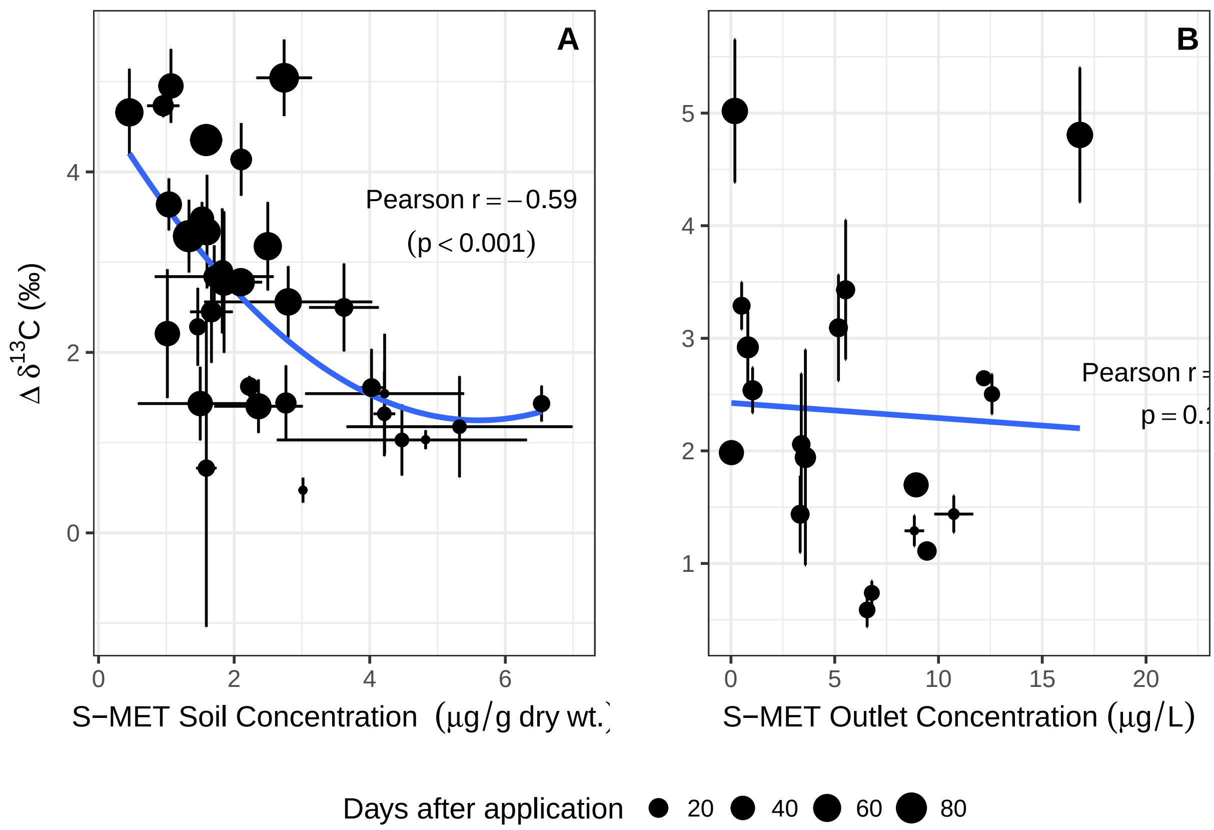

```r
# cor.test(waterClean$TotSMout.g, waterClean$DD13C.diss)

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_errorbar(aes(ymin = DD13C.diss - SD.d13C, ymax = DD13C.diss + SD.d13C)) +
  geom_errorbarh(aes(xmin = Conc.mug.L - Conc.SD, xmax = Conc.mug.L + Conc.SD)) +
  geom_point(aes(size = timeSinceApp)) +
  theme_bw() +
  scale_size_continuous(range = c(1, 4)) +
  labs(size="Days post appl.") +
  theme(axis.title.y = element_blank()) +
  #ylab(expression(paste({Delta~delta}^"13","C",  ' (\u2030)'))) +
  xlab(expression(paste("S-MET Outlet Concentration ", {({mu}*g / L)}))) +
  annotate("text", x = 22, y = 2.7, 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)
```

Join XY waters and soils

```
p_noLeg <- p + theme(legend.position = 'none')
p_Leg <- get_legend(p)

water_noLeg <- waterIsoConc + theme(legend.position = 'none')
water_Leg <- get_legend(waterIsoConc)

grid_xyConIso <- plot_grid(p_noLeg, water_noLeg,
                    ncol =2, nrow = 1, align ="v" #,
                    #labels = c("A", "B")
                    )

xyConcIso <- ggdraw() +
  draw_plot(grid_xyConIso, x=0., y=0.1, width =  1, height = .90) +
  #draw_plot(water_noLeg, x=0.5, y = 0.0, width = 0.4, height = 1) +
  draw_plot(p_Leg, x=0.48, y = 0.0, width = 0.1, height = 0.1) +
  draw_label("A", x= 0.47, y = .95, size = 12, fontface = "bold") +
  draw_label("B", x= 0.97, y = .95, size = 12, fontface = "bold")

xyConcIso
```




```
#ggsave(xyConcIso , filename = "waterSoil_DDvsConc.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 = -1.4068, df = 25, p-value = 0.1718
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.5901137  0.1217051
## sample estimates:
##        cor
## -0.2708348
```

```
#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  118 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 -31.5 -31.7 ...
##  $ 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 0.738 0.587 ...
##  $ DD13C.filt            : num  NA NA NA NA NA ...
##  $ B.diss                : num  NA NA NA 35.3 29.3 ...
```

```
##  $ 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  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 ...
##  $ Appl.Mass.g.NoSo     : num  NA NA NA 0 0 ...
##  $ timeSinceApp.NoSo.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 -0.595 -0.532 ...
##  $ 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 ...
##  $ CumOutSmeto.g.SD     : num  NA NA NA 0.0389 0.0357 ...
##  $ CumOutMELsm.g.SD     : num  NA NA NA 1.14 1.02 ...
##  $ yRaleigh             : num  NA NA NA 0.000763 0.000607 ...
##  $ 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 ...
##  $ timeSinceApp.NoSo.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 -31.5 -31.7 ...
##  $ 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 613.1 NA ...
##  $ comp.d13C.North       : num  NA NA NA NA NA ...
##  $ comp.d13C.SD.North    : num  NA NA NA NA NA ...
##  $ ID.N                  : Factor w/ 17 levels "AW-N-0","AW-N-0x",..: 2 NA NA 1 NA NA 3 NA NA 10 ...
##  $ 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 ...
##   [list output truncated]
```

```r
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"          "Appl.Mass.g.NoSo"
##  [51] "timeSinceApp.NoSo.x"     "CumAppMass.g.x"
##  [53] "DissSmeto.g"             "DissSmeto.g.SD"
##  [55] "DissOXA.g"               "DissOXA.g.SD"
##  [57] "DissESA.g"               "DissESA.g.SD"
##  [59] "FiltSmeto.mg.SD"         "FiltSmeto.g"
##  [61] "FiltSmeto.g.SD"          "TotSMout.g"
##  [63] "TotSMout.g.SD"           "MELsm.g"
##  [65] "MELsm.g.SD"              "CumOutDiss.g"
##  [67] "CumOutFilt.g"            "CumOutSmeto.g"
##  [69] "CumOutMELsm.g"           "BalMassDisch.g"
##  [71] "prctMassOut"             "FracDeltaOut"
##  [73] "Events"                  "Weeks"
##  [75] "Event.x"                 "remainMedTheo_prc"
##  [77] "remainMinTheo_prc"       "remainMaxTheo_prc"
##  [79] "CumOutSmeto.g.SD"        "CumOutMELsm.g.SD"
##  [81] "yRaleigh"                "xRaleigh"
##  [83] "DIa"                     "Event.y"
##  [85] "timeSinceApp.y"          "timeSinceApp.NoSo.y"
```

```
##  [87] "diss.d13C.y"           "SD.d13C.y"
##  [89] "CumAppMass.g.y"        "B.mean.comp.North"
##  [91] "B.max.comp.North"      "B.min.comp.North"
##  [93] "MassSoil.g.North"      "comp.d13C.North"
##  [95] "comp.d13C.SD.North"    "ID.N"
##  [97] "B.mean.comp.Talweg"    "B.max.comp.Talweg"
##  [99] "B.min.comp.Talweg"     "MassSoil.g.Talweg"
## [101] "comp.d13C.Talweg"      "comp.d13C.SD.Talweg"
## [103] "B.mean.comp.South"     "B.max.comp.South"
## [105] "B.min.comp.South"      "MassSoil.g.South"
## [107] "comp.d13C.South"       "comp.d13C.SD.South"
## [109] "ID.S"                  "CatchMassSoil.g"
## [111] "BulkMass.g"            "BulkCatch.d13"
## [113] "BulkCatch.d13.SD"      "f.mean.bulk"
## [115] "B.mean.bulk"           "DD13C.North"
## [117] "DD13C.Talweg"          "DD13C.South"
```

```
keepWS <- c("Date.ti", "WeekSubWeek", "ID.N", "Event.x",
            "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",

            "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"   "ID.N"                 "comp.d13C.SD.Talweg"
## [13] "comp.d13C.SD.South"   "BulkCatch.d13"        "BulkCatch.d13.SD"
## [16] "DD13C.North"          "DD13C.Talweg"         "DD13C.South"
## [19] "DD13.Bulk"
```

```
wsSmall <- wsSmall[c("Date.ti", "WeekSubWeek", "ID.N", "Event.x",
                     "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", "Week", "IDSoil", "Event",
                    "Qmax", "Qmean",
            "diss.measure", "diss.SD",
            "filt.measure", "filt.SD",
```

```r
            "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, -IDSoil, -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$IDSoil <- as.factor(wstidier$IDSoil)
wstidier$Event <- as.numeric(wstidier$Event)

# Copy all data
wstidierAll <- wstidier

levels(wstidier$Source)
```

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

```r
levels(wstidier$Type)
```

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

```r
#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] -32.253
```

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

```r
  # Applics
  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 applics
  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("1 month"), labels = date_format("%b %d")) +
  #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 = "Degradation (%)",
                        #name = element_blank(),
                        breaks=c(20, 40, 60, 70, 80, 85, 90, 95) )# breaks=seq(20, 120, 15))
  ) +
  scale_color_manual(name= "Source",
                     values = c("#F8766D", "#00BFC4", "#00BA38", "#B79F00", "#619CFF", "#F564E3",
                                "#D55E00", "darkgreen", "dodgerblue")
                     ) +
  scale_size_continuous(range = c(1, 3)) #



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 = Qma
  geom_point(data=subset(wstidier2, Source == 'Outlet'), aes(shape = Type, colour = Source, size = Qmean
  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, siz
# 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() +
# Applics
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 applics
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")) +
#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 = element_blank(),
                      #name = "Degradation (%)",
                      breaks=c(20, 40, 60, 70, 80, 85, 90, 95) )# breaks=seq(20, 120, 15))
)  +
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(order = 1,
                          #title=expression("Source"),
                          #title.vjust = -1,
                          nrow = 2,
                          title.position = "left"
                          ),
```

```r
         shape=guide_legend(#title=expression("Type"),
                            order = 2,
                            nrow=2,
                            title.position = "left",
                            keyheight = NULL, title.vjust = NULL, label.vjust = NULL),
         size = guide_legend(order = 3,
                             title=expression("Mean\nDischarge " ~ (m^3 / h) ),
                             nrow=2, title.position = "left", title.vjust = .26
                              ))# +
  # 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
```

```r
sd(c( 1.70, 0.94))
```

```
## [1] 0.5374012
```

```r
# May 29th event
# 2 samples from second peak
mean(c( 2.65, 2.56))
```

```
## [1] 2.605
```

```r
sd(c( 2.65, 2.56))
```

```
## [1] 0.06363961
```

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

```
## [1] 2.6395
```

```r
sd(c(2.507, 2.772))
```

```
## [1] 0.1873833
```

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

```
## [1] 1.335
```

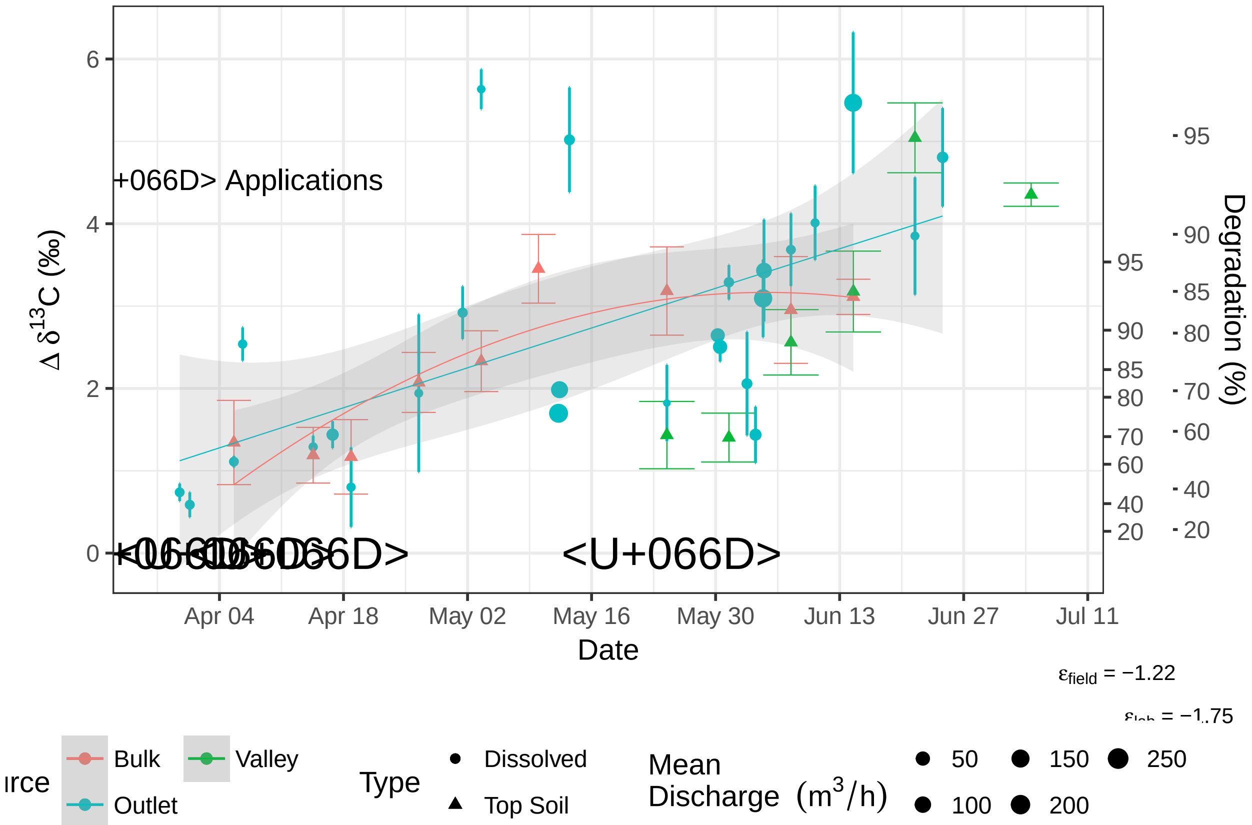
```r
sd(c( 1.23, 1.44 ))
```

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

Join all figures

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

```r
# 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_lab_noLeg, x=0, y = 0.2, width = 1, height = 0.8) + # bottom
  draw_plot(wsALL_field_noLeg, x=0, y=.2, width = 0.935, height = .8) + # top
  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 = 0.50, height = 0.15)

wsALL
```

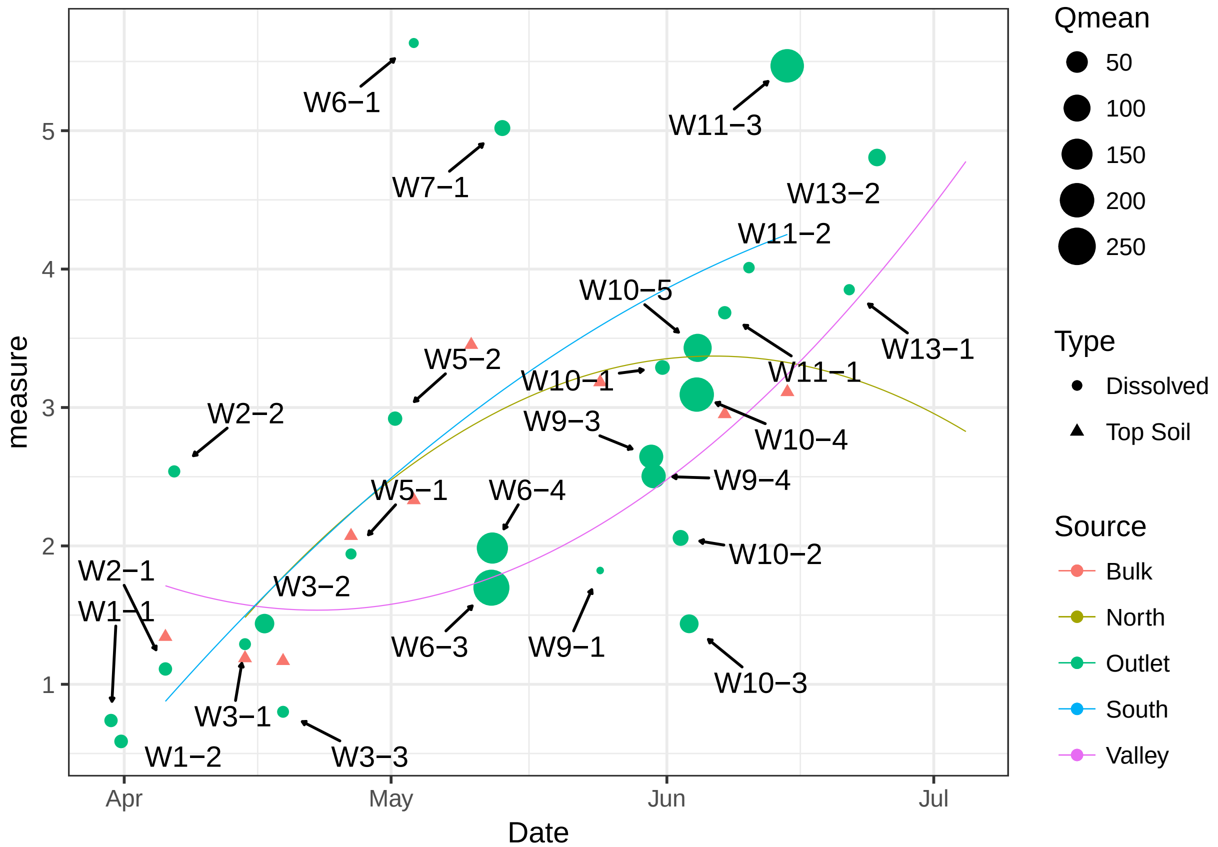



```
# ggsave(wsALL, filename = "WaterSoilvsTime.png", width = 8, height = 5, units = "in", scale = 1)
```

Check SOils

```
wstidier2$IDSoil <- as.character(wstidier2$IDSoil)
split <- strsplit(wstidier2$IDSoil, "-", fixed = TRUE)
wstidier2$Soil.ID <- sapply(split, "[", 3)
wstidier2$Soil.ID <- as.factor(wstidier2$Soil.ID)

ggplot(data = wstidier2, aes(x = Date, y = measure, group = Source) )+
  theme_bw() +
  #geom_errorbar(data=subset(wstidier2, Type == 'Top Soil'), 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 = Qma
  geom_point(data=subset(wstidier2, Type == 'Dissolved'), aes(shape = Type, colour = Source, size = Qme
  stat_smooth(data=subset(wstidier2,
                          (Source == "North" )), #| Source == "South" )),
              method = "lm", formula = y ~ poly(x, 2), se = F, aes(colour = Source), alpha = 0.2, size=
  stat_smooth(data=subset(wstidier2,
                          (Source == "Valley" )), #| Source == "South" )),
              method = "lm", formula = y ~ poly(x, 2), se = F, aes(colour = Source), alpha = 0.2, size=
```

```
  stat_smooth(data=subset(wstidier2,
                          (Source == "South" )), #| Source == "South" )),
              method = "lm", formula = y ~ poly(x, 2), se = F, aes(colour = Source), alpha = 0.2, size=(
  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 = Sourc
  #geom_text_repel(data=subset(wstidier2, Source == 'Bulk'), aes(label=Soil.ID),
  #               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(data=subset(wstidier2, Source == 'Outlet'), aes(label=Week),
                  arrow = arrow(length = unit(0.005, 'npc'), type = "closed"),
                  force = 1,
                  point.padding = unit(1.0, 'lines'),
                  max.iter = 2e3,
                  nudge_x = .2)
```




```
sum(wstidier2$Location == "North")   # 10
```

```
## [1] 9
```

```
sum(wstidier2$Location == "South")   # 12
```

```
## [1] 11
```

```r
sum(wstidier2$Location == "Talweg")  # 12
```

```
## [1] 12
```

```r
sum(wstidier2$Source == "Bulk")  # 9
```

```
## [1] 9
```

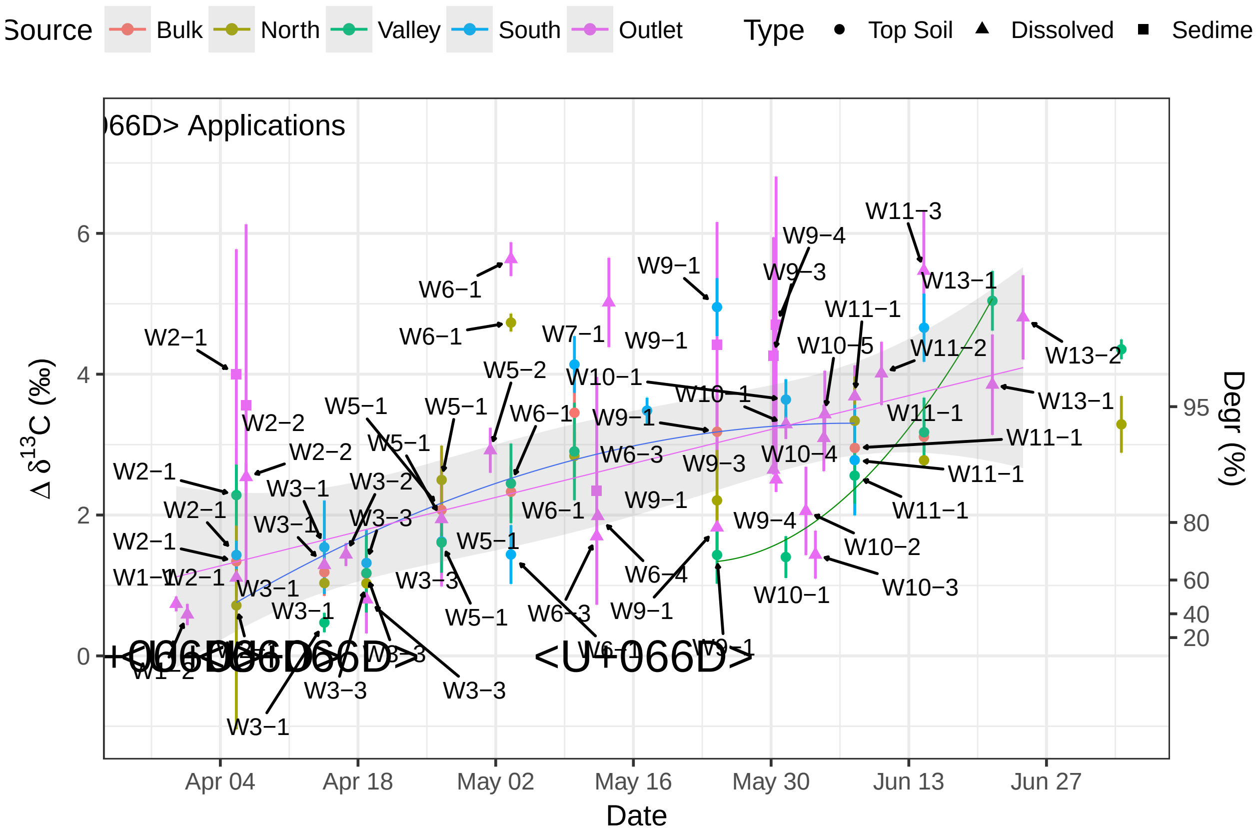
Comparison to Lutz et al. (2013)

```r
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)

```r
# 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.
  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, siz
  #stat_smooth(data=subset(wsNoPlat,
  #                         (Source == "Bulk")),
  #            method = "lm", formula = y ~ poly(x, 2), se = T, aes(colour = 'Bulk'), alpha = 0.2, size
  theme_bw() +
```

```r
  # Applics
  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 08:04:00'), 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('2016-05-17 08:04:00'), y = 0,
           label = as.character(expression(paste( "\u066D"))), parse = T, size = 6.0) +
  # Title applics
  annotate("text", x = as.POSIXct('2016-04-01 08:04:00'), y = 7.5,
           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_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
```



Testing difference in $\Delta\delta$ between groups

Based on ANOVA tests, there is:

- No significant difference between soils and water

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

```r
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",
```

```r
                                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'), "Ou
                       ifelse(mwsStatTest$variable == "diss.d13C.x" &
                                  mwsStatTest$Date.ti <= as.POSIXct('2016-06-05 00:06:00', tz = 'EST'), "O
                       ifelse(mwsStatTest$variable == "comp.d13C.Talweg" &
                                  mwsStatTest$Date.ti > as.POSIXct('2016-06-05 00:06:00', tz = 'EST'), "Va
                       ifelse(mwsStatTest$variable == "comp.d13C.Talweg" &
                                  mwsStatTest$Date.ti <= as.POSIXct('2016-06-05 00:06:00', tz = 'EST'), "V
                       ifelse( (mwsStatTest$variable == "comp.d13C.North" | mwsStatTest$variable == "com
                                  mwsStatTest$Date.ti <= as.POSIXct('2016-06-05 00:06:00', tz = 'EST'), "P
                       ifelse( (mwsStatTest$variable == "comp.d13C.North" | mwsStatTest$variable == "com
                                  mwsStatTest$Date.ti > as.POSIXct('2016-06-05 00:06:00', tz = 'EST'), "Pl
                                 ))))))

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 = 3.4782, df = 5, p-value = 0.6267
```

```r
# 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
```

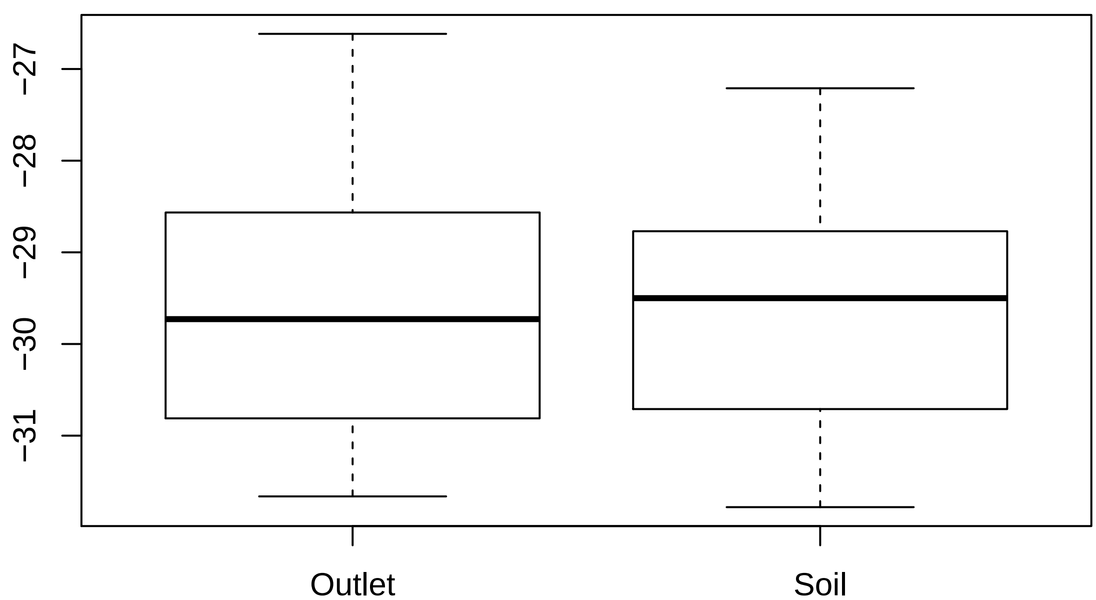
```
##
##  Kruskal-Wallis rank sum test
##
## data:  value by as.factor(Group3)
## Kruskal-Wallis chi-squared = 20.897, df = 5, p-value = 0.0008473
```

```r
# 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 homegenity of variance

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



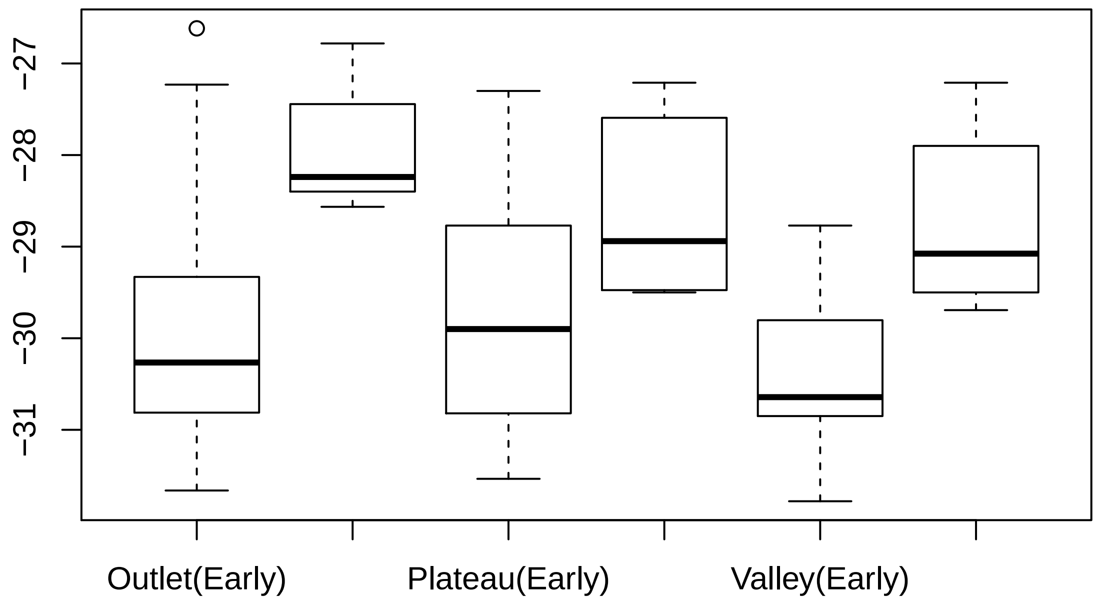

```
summary(aov(Gr1$value ~ Gr1$Group1))
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## Gr1$Group1    1   0.09  0.0879   0.047  0.829
## Residuals    66 122.76  1.8599
```

```
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 0.07399219 -0.6054846 0.753469 0.8285537
```

```
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  36.26   7.251   5.192 0.000482 ***
## Residuals   62  86.59   1.397
## ---
## 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: 0.001733
##       Significance: 0.353
##
## Permutation: free
## Number of permutations: 999
##
```

```
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0356 0.0536 0.0708 0.0954
##
## Dissimilarity ranks between and within classes:
##           0%    25%    50%     75% 100%    N
## Between 11.0 563.25 1137.0 1725.75 2278 1092
## Outlet  12.0 605.00 1208.0 1813.00 2277  325
## Soil     5.5 557.00 1122.5 1665.00 2264  861
```

```r
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)
```

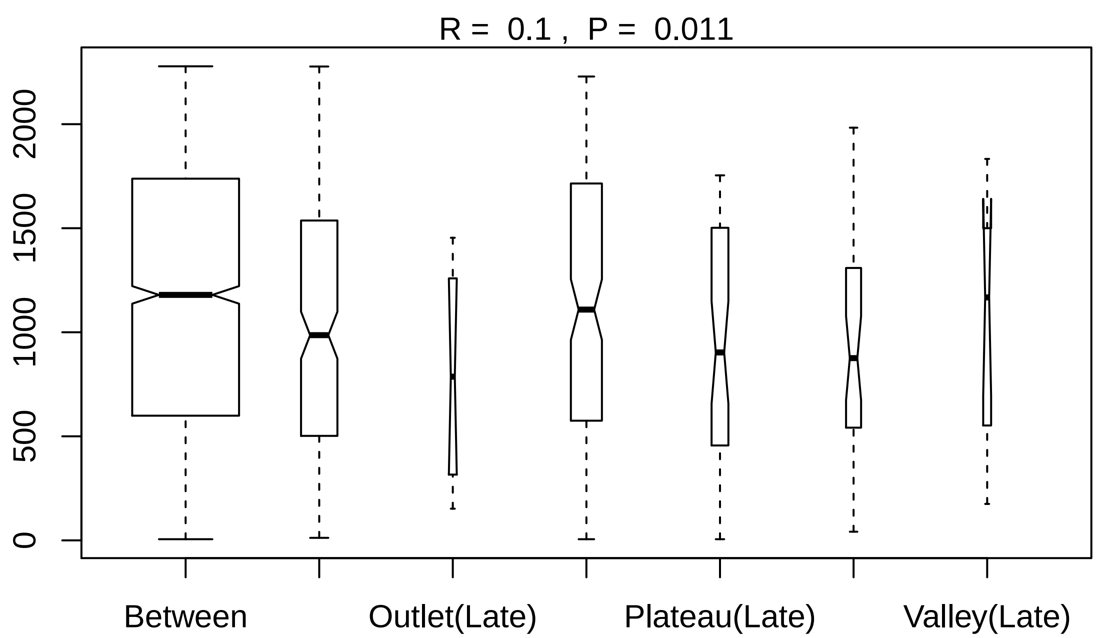
```
##
## Call:
## anosim(dat = Gr2.hell.daisy, grouping = Group2)
## Dissimilarity:
##
## ANOSIM statistic R: -0.03174
##       Significance: 0.894
##
## Permutation: free
## Number of permutations: 999
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0364 0.0542 0.0707 0.0830
##
## Dissimilarity ranks between and within classes:
##           0%   25%  50%     75% 100%    N
## Between  5.5 554.5 1124 1698.50 2278 1484
## Outlet  12.0 605.0 1208 1813.00 2277  325
## Plateau  5.5 583.0 1161 1687.75 2247  378
## Valley  41.5 620.5 1059 1589.75 2264   91
```

```r
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:   0.1
##       Significance: 0.011
##
## Permutation: free
## Number of permutations: 999
##
```

```
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0511 0.0662 0.0797 0.0993
##
## Dissimilarity ranks between and within classes:
##                   0%    25%     50%     75% 100%    N
## Between          5.5 599.50 1179.50 1737.50 2278 1814
## Outlet(Early)   12.0 506.50  986.00 1536.75 2277  210
## Outlet(Late)   152.0 390.75  786.50 1190.50 1454   10
## Plateau(Early)   5.5 575.00 1109.00 1715.00 2229  153
## Plateau(Late)    5.5 456.00  903.00 1502.00 1754   45
## Valley(Early)   41.5 547.75  875.75 1275.00 1983   36
## Valley(Late)   175.0 573.50 1167.50 1485.00 1833   10
```

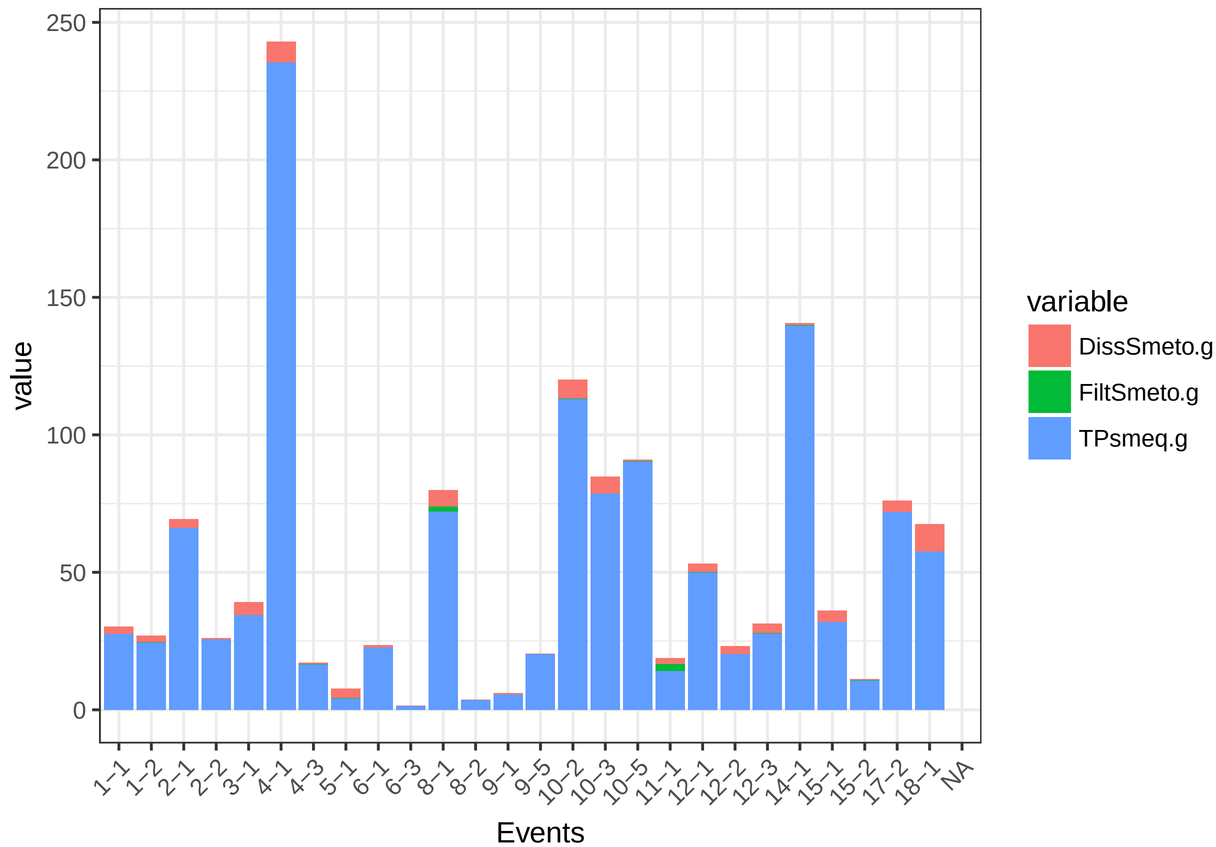
```r
plot(anosim.group3)
```



Loadings

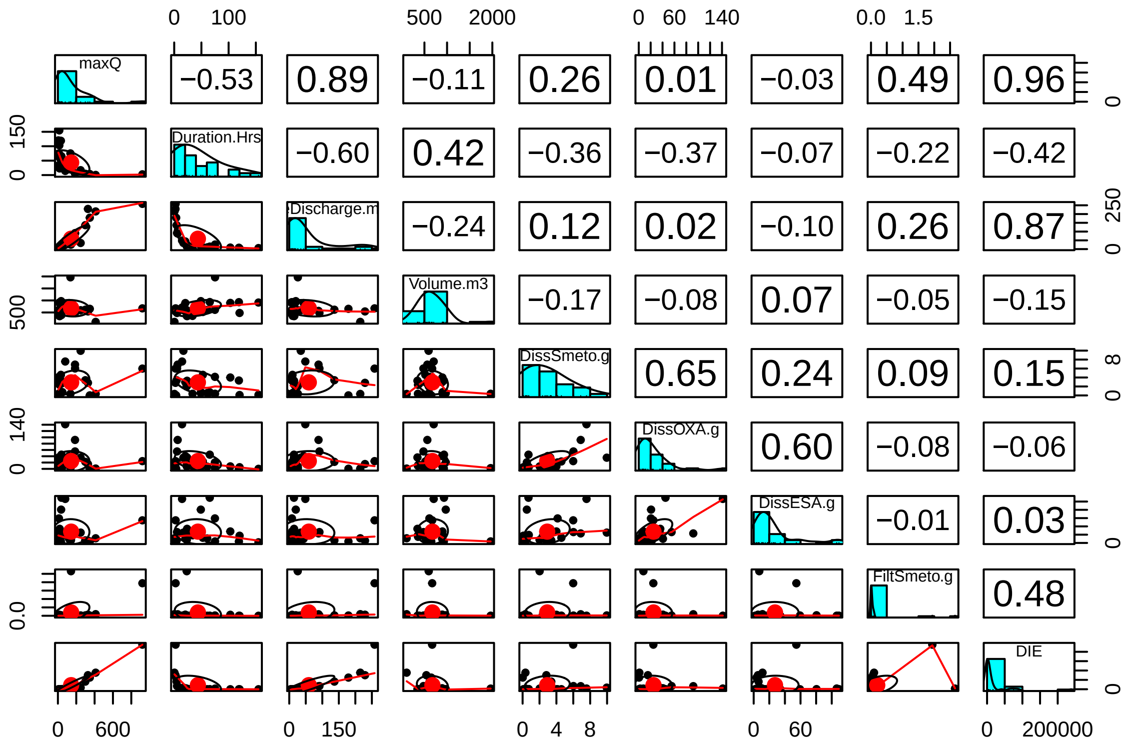
```r
keepLoads <- c("Date.ti",
            "DissOXA.g", "DissESA.g", "DissSmeto.g", "FiltSmeto.g",
            "Event.x", "Events")
wsLoads <- WaterSoils[ , (names(WaterSoils) %in% keepLoads)]

mw.SM <- 283.796 # g/mol
mw.MOXA <- 279.33 # g/ml
```

```r
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)
        )
```

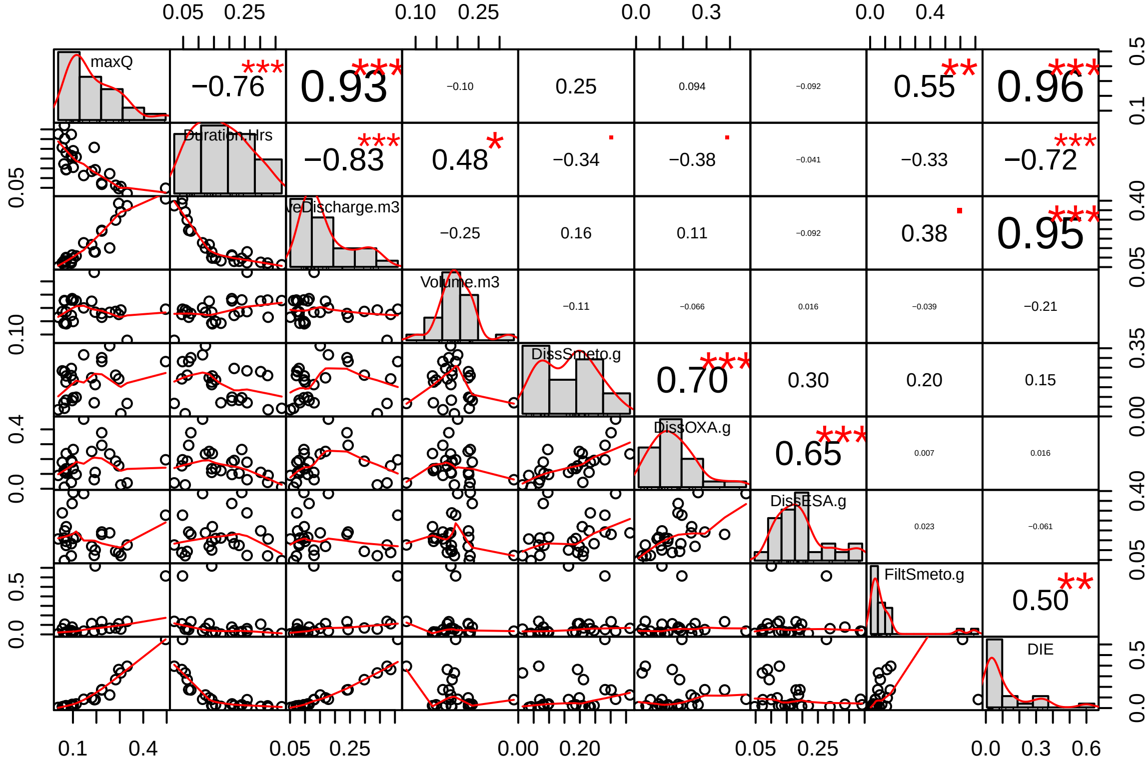



```r
  # geom_bar(stat = "identity", position = position_dodge())

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"
            )
```

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

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

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




```r
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:

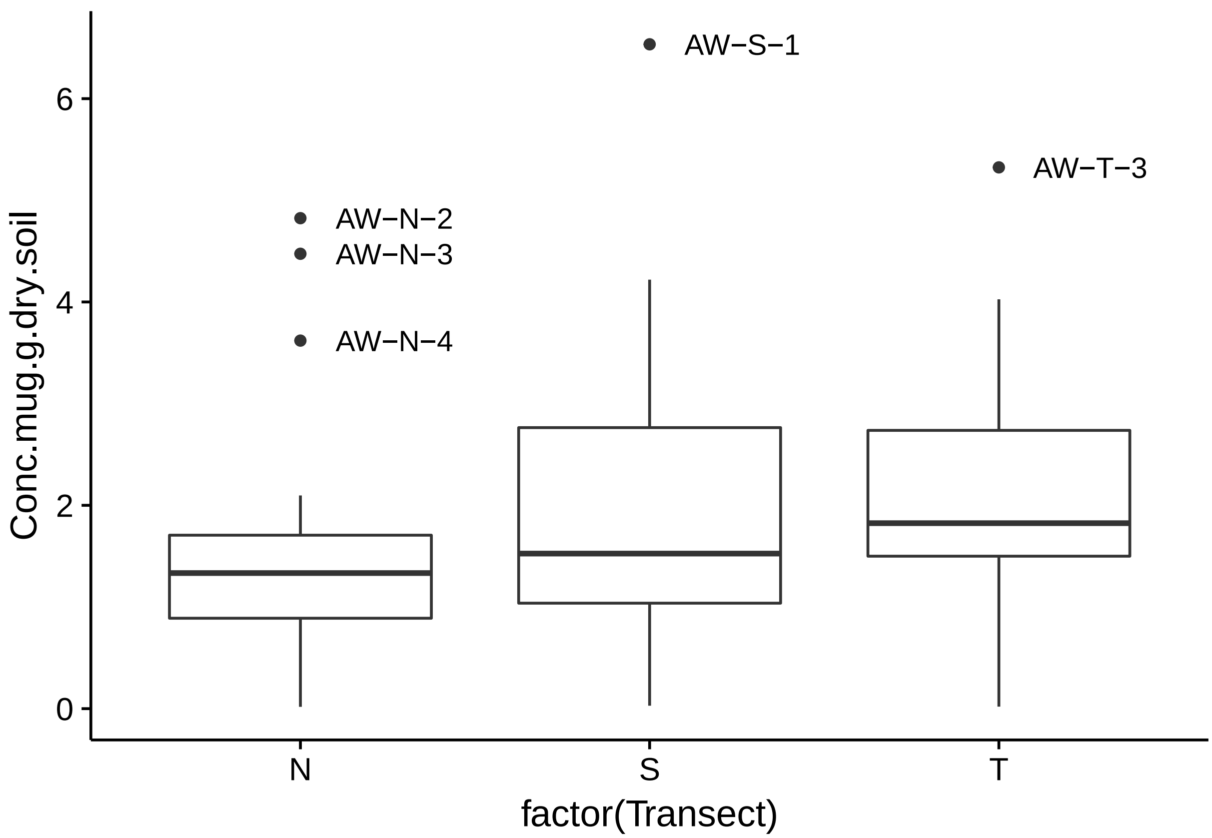
$$z_i = \frac{y_i - \bar{y}}{s_y}$$

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

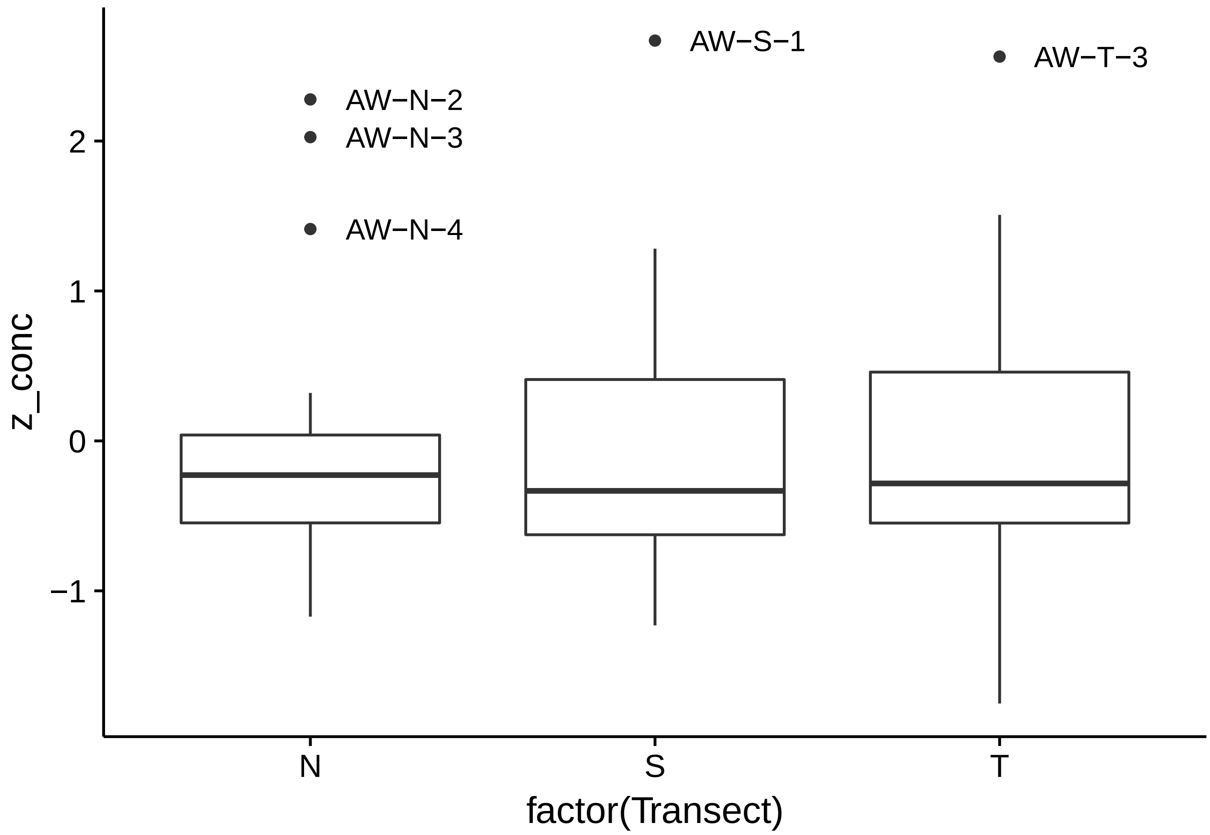
```
# Concentrations
soilGroups %>%
  group_by(Transect) %>%
  mutate(outlier = ifelse(is_outlier(Conc.mug.g.dry.soil), as.character(ID), NA)) %>%
  ggplot(., aes(x = factor(Transect), y = Conc.mug.g.dry.soil)) +
    geom_boxplot() +
    geom_text(aes(label = outlier), na.rm = TRUE, hjust = -0.3)
```



Outliers after transformation

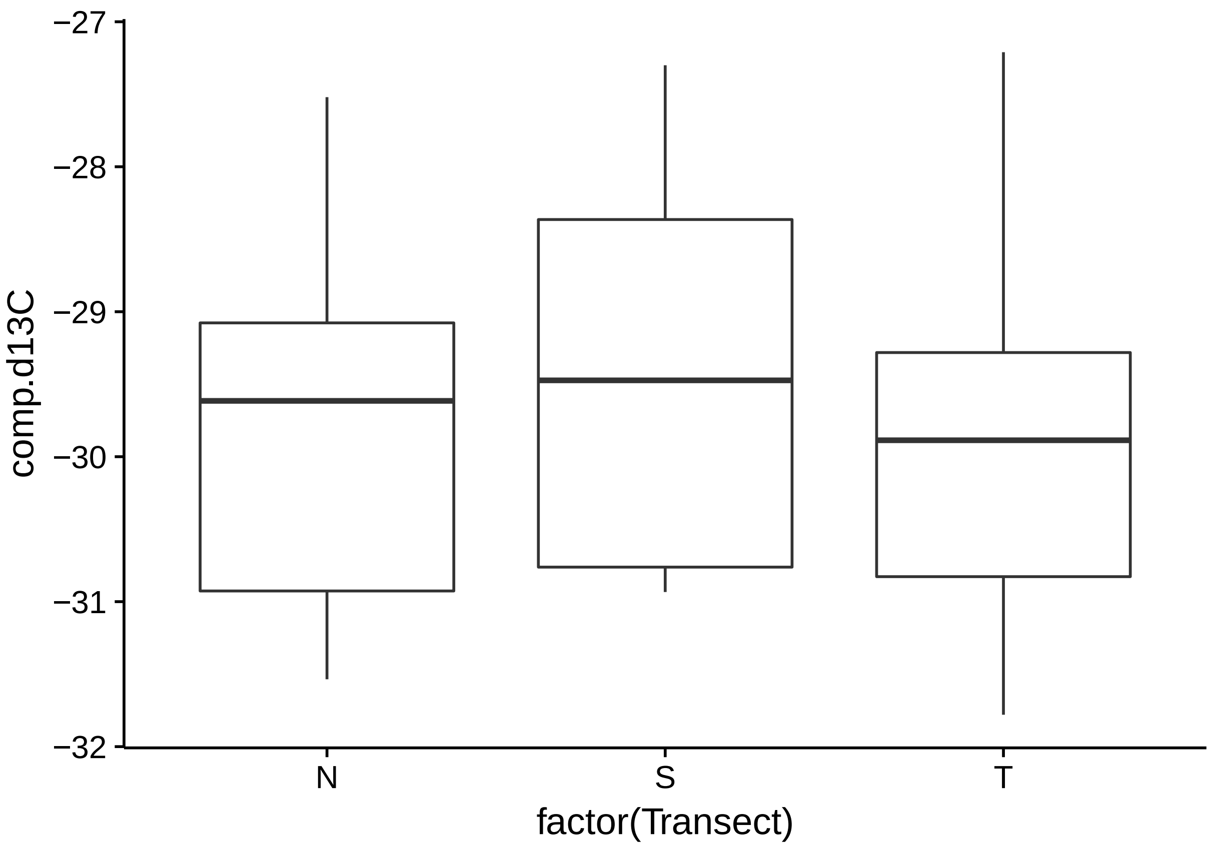
```
soilGroups <- soilGroups %>%
  group_by(Transect) %>%
  mutate(z_conc = (Conc.mug.g.dry.soil-mean(Conc.mug.g.dry.soil))/sd(Conc.mug.g.dry.soil))
```

```r
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)
```



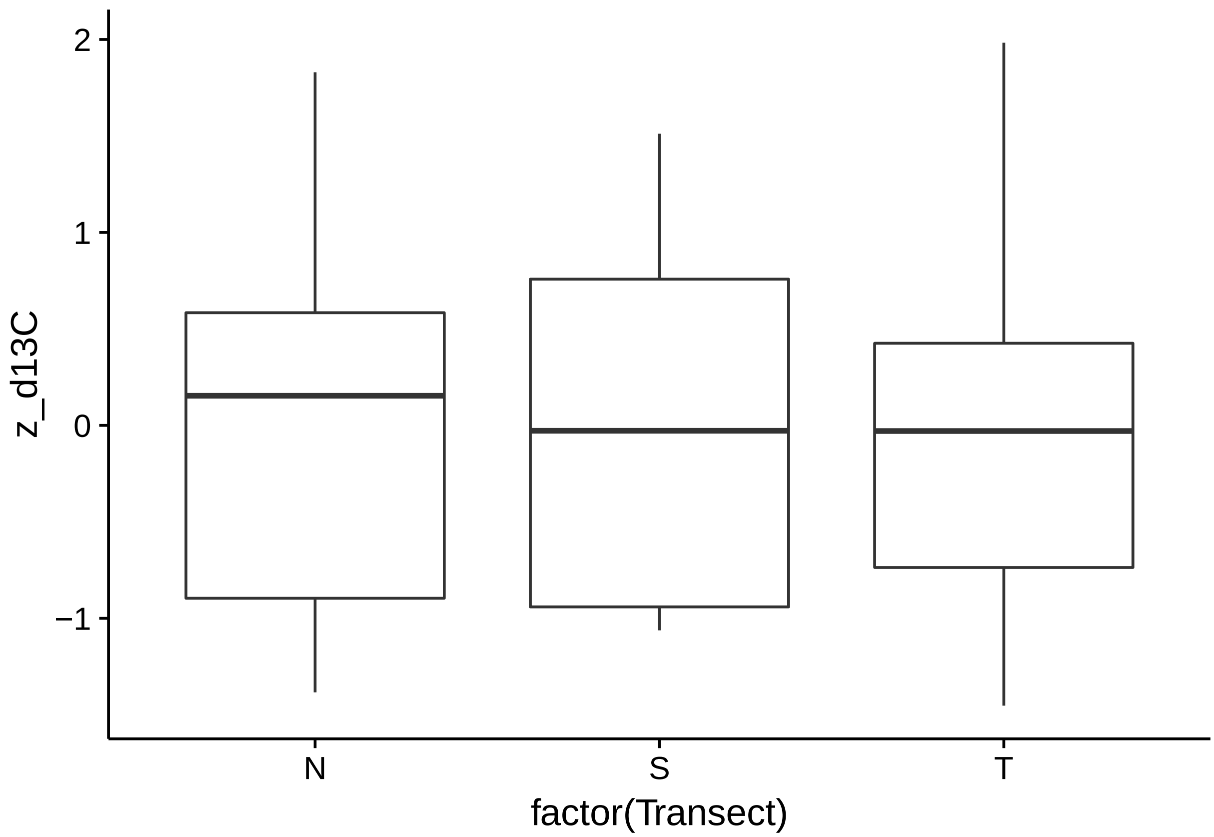
Soil Isotopes

```r
# 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)
```



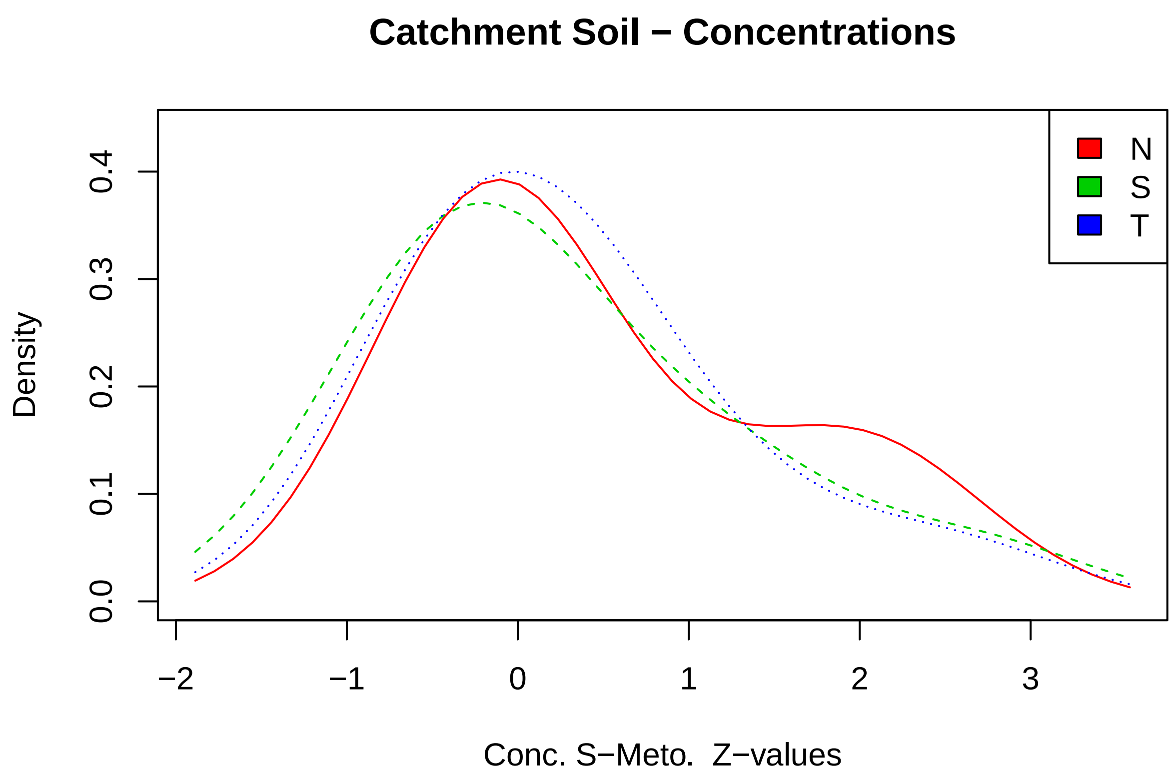
Looks like 7 potential outliers in concentrations and 1 for isotopes. Removing NA's for isotopes and re-computing outliers, reduces the number of outliers to 2 in concentrations and 1 for isotopes.

```
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)))
```

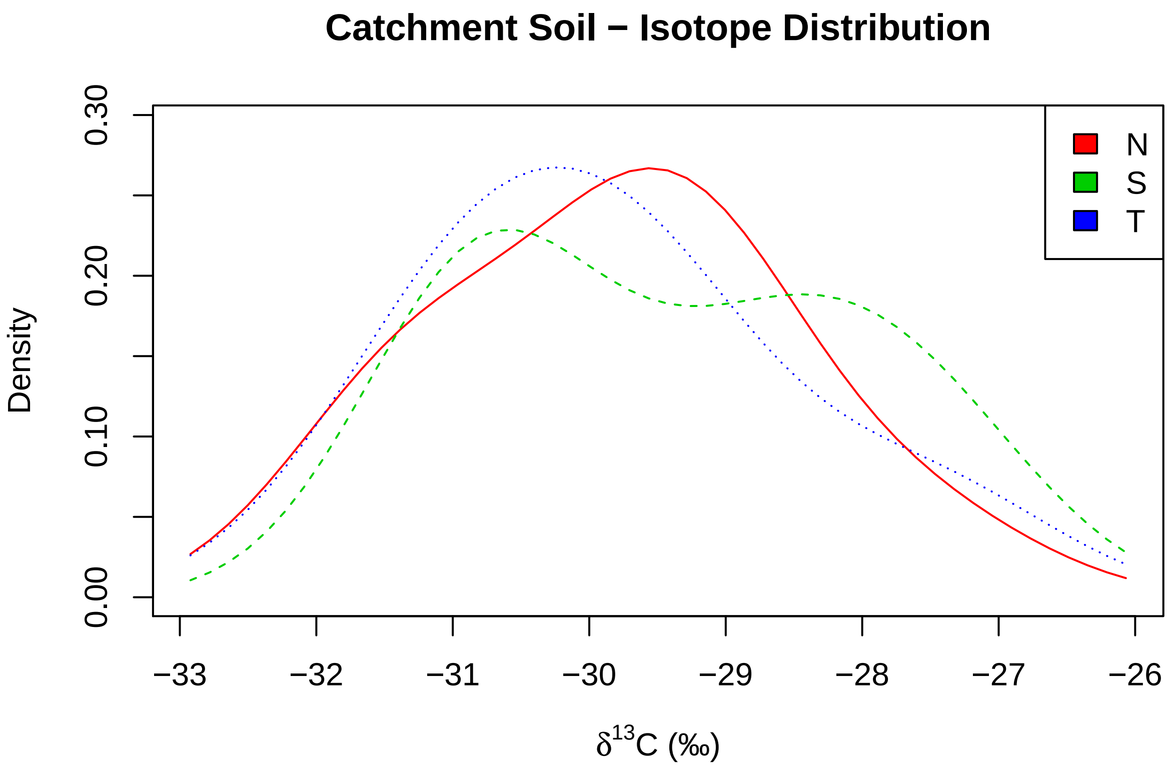



```
#vioplot(soilGroups$Conc.mug.g.dry.soil,  names = "Catchment")
#title(expression(paste("Conc. S-Meto.  ", {({mu}*g / g.soil.dry)})))
```

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)))
```



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.