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.

Files Used

- **MassBalance__R.csv**
- **WeeklySoils__Rng.csv**

Files Written

- **OutletData4Lutz__R.csv** (Data to compare against Lutz 2013 article)

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")
```

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

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

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).

Lab parameters

```r
source("global.R")
```

Import soils

Convert to single time observation for merging with water observation.

```r
# Soils
soils = read.csv2("Data/MassBalance_R.csv",
                        na.strings=c('#DIV/0!', '', 'NA'), header = TRUE)
names(soils)
```

```
##  [1] "ti"                   "WeekSubWeek"
##  [3] "Event"                "Duration.Hrs"
##  [5] "timeSinceApp"         "timeSinceApp.NoSo"
##  [7] "timeSinceApp.N"       "timeSinceApp.T"
##  [9] "timeSinceApp.S"       "diss.d13C"
## [11] "SD.d13C"              "CumOutDiss.g"
## [13] "CumOutFilt.g"         "TotSMout.g"
## [15] "TotSMout.g.SD"        "MELsm.g"
## [17] "MELsm.g.SD"           "Appl.Mass.g"
## [19] "Appl.Mass.g.OT"       "CumAppMass.g"
## [21] "CumAppMass.g.N"       "CumAppMass.g.T"
## [23] "CumAppMass.g.S"       "CumAppMass.g.OT"
## [25] "CumAppMass.g.N.OT"    "CumAppMass.g.T.OT"
```

```
## [27] "CumAppMass.g.S.OT"     "iniCo.ug.g.N"
## [29] "iniCo.ug.g.T"          "iniCo.ug.g.S"
## [31] "CumOutSmeto.g"         "CumOutMELsm.g"
## [33] "MassSoil.g.North"      "MassSoil.g.SD.North"
## [35] "Conc.mug.g.dry.soil.N" "comp.d13C.North"
## [37] "comp.d13C.SD.North"    "ID.N"
## [39] "Area.N"                "Area.T"
## [41] "Area.S"                "MassSoil.g.Talweg"
## [43] "MassSoil.g.SD.Talweg"  "Conc.mug.g.dry.soil.T"
## [45] "comp.d13C.Talweg"      "comp.d13C.SD.Talweg"
## [47] "ID.T"                  "MassSoil.g.South"
## [49] "MassSoil.g.SD.South"   "Conc.mug.g.dry.soil.S"
## [51] "comp.d13C.South"       "comp.d13C.SD.South"
## [53] "ID.S"                  "DD13C.North"
## [55] "DD13C.Talweg"          "DD13C.South"
## [57] "CatchMassSoil.g"       "CatchMassSoil.g.SD"
## [59] "BulkCatch.d13"         "BulkCatch.d13.SD"
## [61] "DD13.Bulk"             "Area.Catchment"
## [63] "BulkCatch.Conc"        "iniCo.Bulk"
```

```r
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
```

```r
initialDelta
```

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

```r
# Get rid of imputed values to avoid 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",
              "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",
              "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",
                        "timeSinceApp.N", "timeSinceApp.T" , "timeSinceApp.S",
                       "Event")]
```

Soils from Book: 06, to merge with "timeApps"

```r
# 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
```

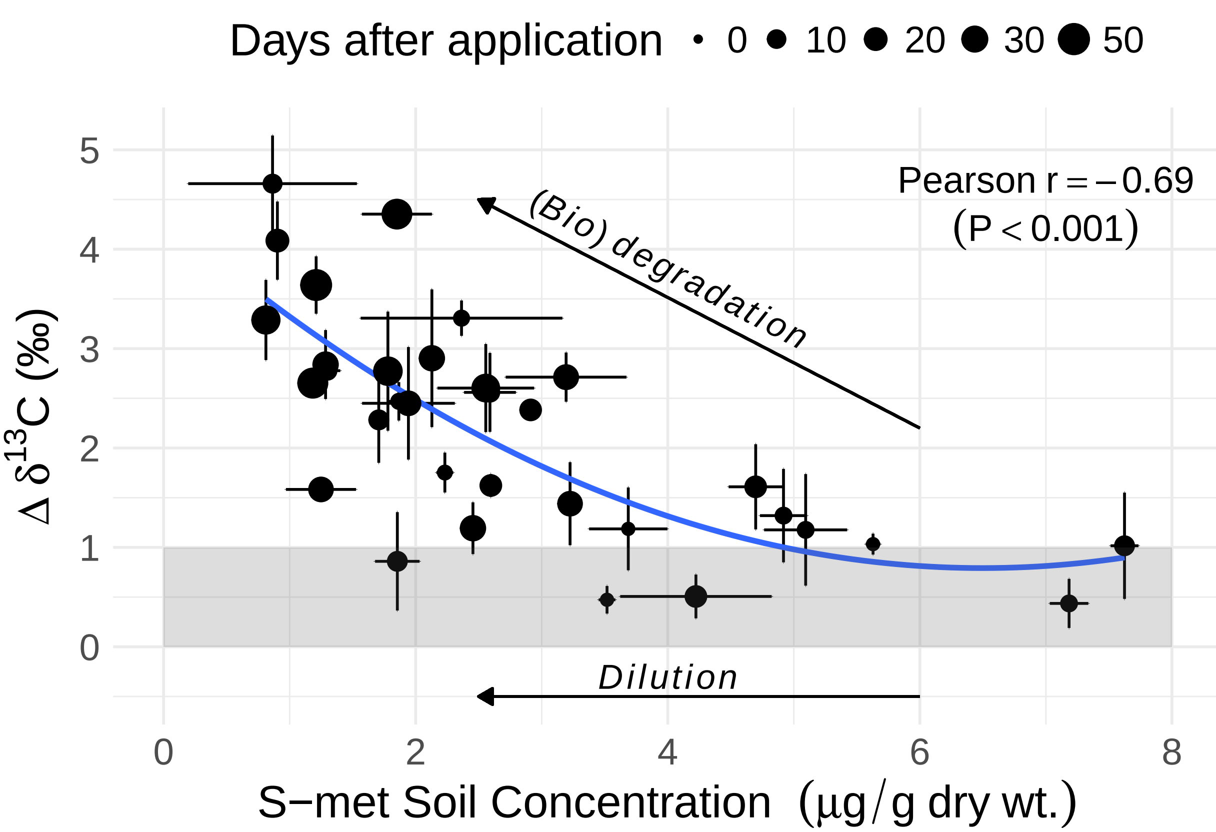
```r
soilGroups$comp.d13C <- ifelse(is.na(soilGroups$comp.d13C.SD), NA, soilGroups$comp.d13C)
# soilGroups$ngC.Label <- ifelse(soilGroups$ngC.mean < 10, "< 10 ng", "> 10 ng")

soilGroups <- subset(soilGroups, comp.d13C.SD <= 0.70)

#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)
soilGrApp <- subset(soilGrApp, comp.d13C.SD <= 0.70)

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 = -5.3227, df = 31, p-value = 8.514e-06
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.8360002 -0.4558517
## sample estimates:
##        cor
## -0.6910178
```

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

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

```
soilGrApp.N <- subset(soilGrApp, soilGrApp$Transect == "N")
soilGrApp.T <- subset(soilGrApp, soilGrApp$Transect == "T")
soilGrApp.S <- subset(soilGrApp, soilGrApp$Transect == "S")

soilGrApp.N$timeSinceApp <- soilGrApp.N$timeSinceApp.N
soilGrApp.T$timeSinceApp <- soilGrApp.T$timeSinceApp.T
soilGrApp.S$timeSinceApp <- soilGrApp.S$timeSinceApp.S

dropAppDates <- c("timeSinceApp.NoSo", "timeSinceApp.N", "timeSinceApp.T", "timeSinceApp.S")
soilGrApp.N <- soilGrApp.N[ , !(names(soilGrApp.N) %in% dropAppDates)]
soilGrApp.T <- soilGrApp.T[ , !(names(soilGrApp.T) %in% dropAppDates)]
soilGrApp.S <- soilGrApp.S[ , !(names(soilGrApp.S) %in% dropAppDates)]

soilGrApp <- rbind(soilGrApp.N, soilGrApp.T)
soilGrApp <- rbind(soilGrApp, soilGrApp.S)

p <- ggplot(data = soilGrApp, 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, Conc.mug.g.dry.soil < 8),
              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.La
  geom_point(aes(group = ID, size = timeSinceApp)) +
  # theme_bw() +
  theme_minimal() +
  theme(legend.position = "top",
        text = element_text(size=17)) +
  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 = 7.0, y = 4.7, label = as.character(r_label), parse = T, size = 5) +
  annotate("text", x = 7.0, y = 4.2, label = p_label, parse = T, size = 5) +
  scale_size_continuous(range = c(1, 5), breaks= c(0, 10, 20, 30, 50), limits = c(0, 50)) +
  scale_y_continuous(breaks=c(0, 1, 2, 3 , 4 ,5) ) +
  # scale_size_continuous(range = c(1, 5)) +
  guides(size=guide_legend(nrow=1)) +
  annotate("text", x = 4, y = -0.3, label= "italic(Dilution)", parse=T, size = 4.5) +
  geom_segment(aes(x = 6, y = -0.5, xend = 2.5, yend = -0.5),
                  arrow = arrow(length = unit(1/2, 'picas'), type = "closed")) +
  annotate("text",
           x = 4.0, y = 3.78,
           label= "paste(\"(\", italic(Bio), \") \", italic(degradation) )", parse=T, size = 4.5, angle
  geom_segment(aes(x = 6, y = 2.2, xend = 2.5, yend = 4.5),
                  arrow = arrow(length = unit(1/2, 'picas'), type = "closed")) +
  annotate("rect", xmin=0, xmax=8, ymin=0, ymax=1, alpha=0.2)

  # geom_rect(aes(xmin=0, xmax=8, ymin=-0.5, ymax=0.5), colour = "grey", alpha = 0.5) +
  #geom_hline(yintercept = 0.5, color = "dodgerblue4", linetype = "dotted") +
  #geom_hline(yintercept = 0, color = "dodgerblue3", linetype = "dotted") +
  #geom_hline(yintercept = -0.5, color = "dodgerblue3", linetype = "dotted")

  #scale_color_hue("Group") +
  #scale_fill_manual(
```

```r
  #  "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
```

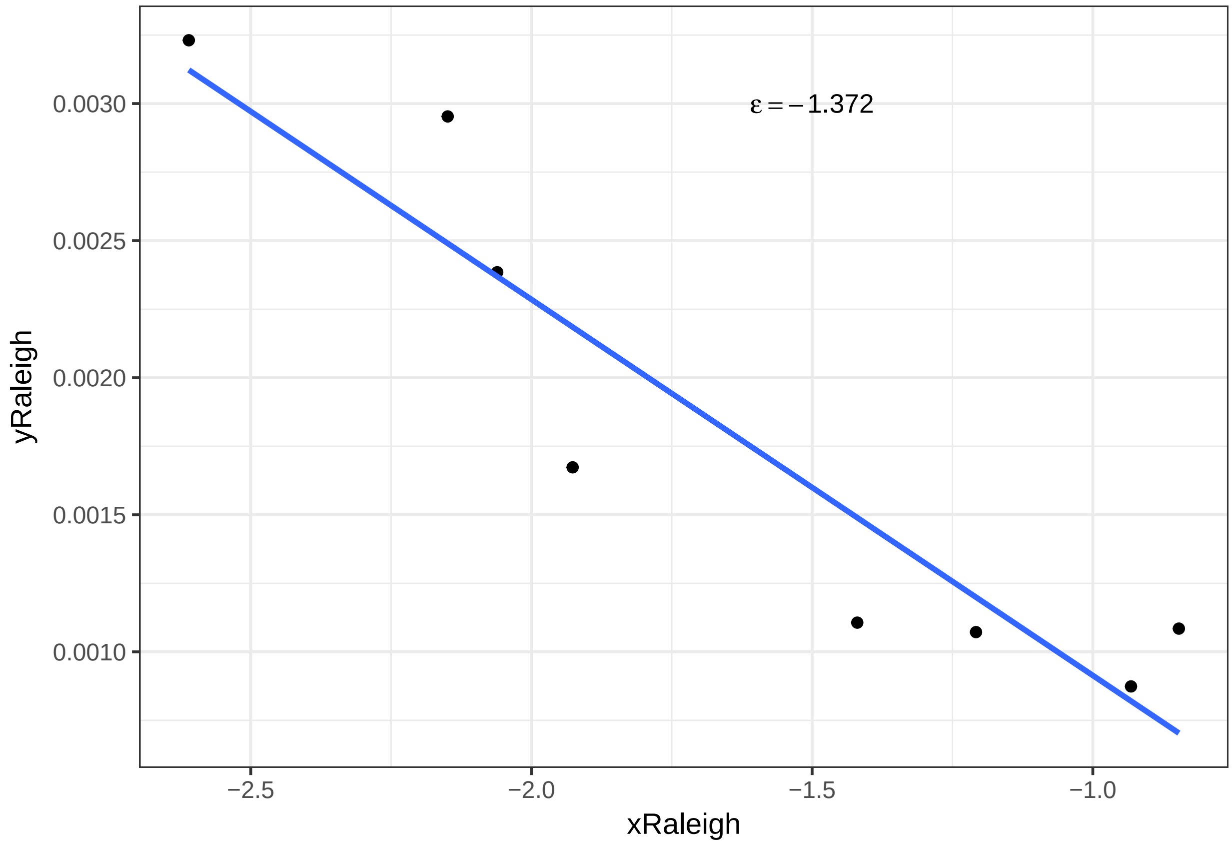



```r
#ggsave(p, filename = "images/DDvsConc_soils.png", width = 8.7, height = 6, units = "cm", scale = 1)
#
# ggsave(p, filename = "images/DDvsConc_soils_2.pdf", device = "pdf", dpi = 300,  scale = 2)

SAVE = T
PC = T
if (SAVE){
  if (PC){
    ggsave(p,
       filename = "D:/Documents/these_pablo/WriteUp/Alteck_PNAS_LaTeX/images/DDvsConc_soils.pdf",
       device = "pdf", dpi = 600,  scale = 1, # )# ,
       width = 8.7, height = 6)

  }
```

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

Field enrichment derivation (for error estimation)

```
soils$yRaleigh <- log((1000+d13Co+soils$DD13.Bulk)/(1000+d13Co))
soils$xRaleigh <- log(soils$BulkCatch.Conc/soils$iniCo.Bulk)
soilModel<-lm(yRaleigh~xRaleigh, data= soils)

cofsoil <- as.numeric(coef(soilModel)[2]*1000)
minX <- confint(soilModel, "xRaleigh", level = 0.95)[1]*1000
maxX <- confint(soilModel, "xRaleigh", level = 0.95)[2]*1000
se <- summary(soilModel)$coef[[4]]*1000

e_label <- sprintf("epsilon == %0.3f", cofsoil)

CI95 = maxX - cofsoil

ggplot(data = subset(soils,  !is.na(yRaleigh) ), aes(x=xRaleigh, y=yRaleigh)) +
  geom_point() +
  stat_smooth(method = "lm", formula = y ~ x, se=F) +
  annotate("text", x = -1.5, y = 0.003, label = as.character(e_label), parse = T, size = 3.5) +
  theme_bw()
```




```r
summary(soilModel)
```

```
##
## Call:
## lm(formula = yRaleigh ~ xRaleigh, data = soils)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -5.117e-04 -1.912e-04  3.457e-05  1.766e-04  4.631e-04
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.0004578  0.0003796  -1.206 0.273211
## xRaleigh    -0.0013717  0.0002171  -6.317 0.000735 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.000365 on 6 degrees of freedom
##   (44 observations deleted due to missingness)
## Multiple R-squared:  0.8693, Adjusted R-squared:  0.8475
## F-statistic: 39.91 on 1 and 6 DF,  p-value: 0.0007347
```

Import waters

Compare mass balance, theoretical and CSIA

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

# Concentration ranges not being able to quantify CSIA
low4CSIA <- subset(waters, !is.na(diss.d13C))
min(low4CSIA$Conc.mug.L)
```

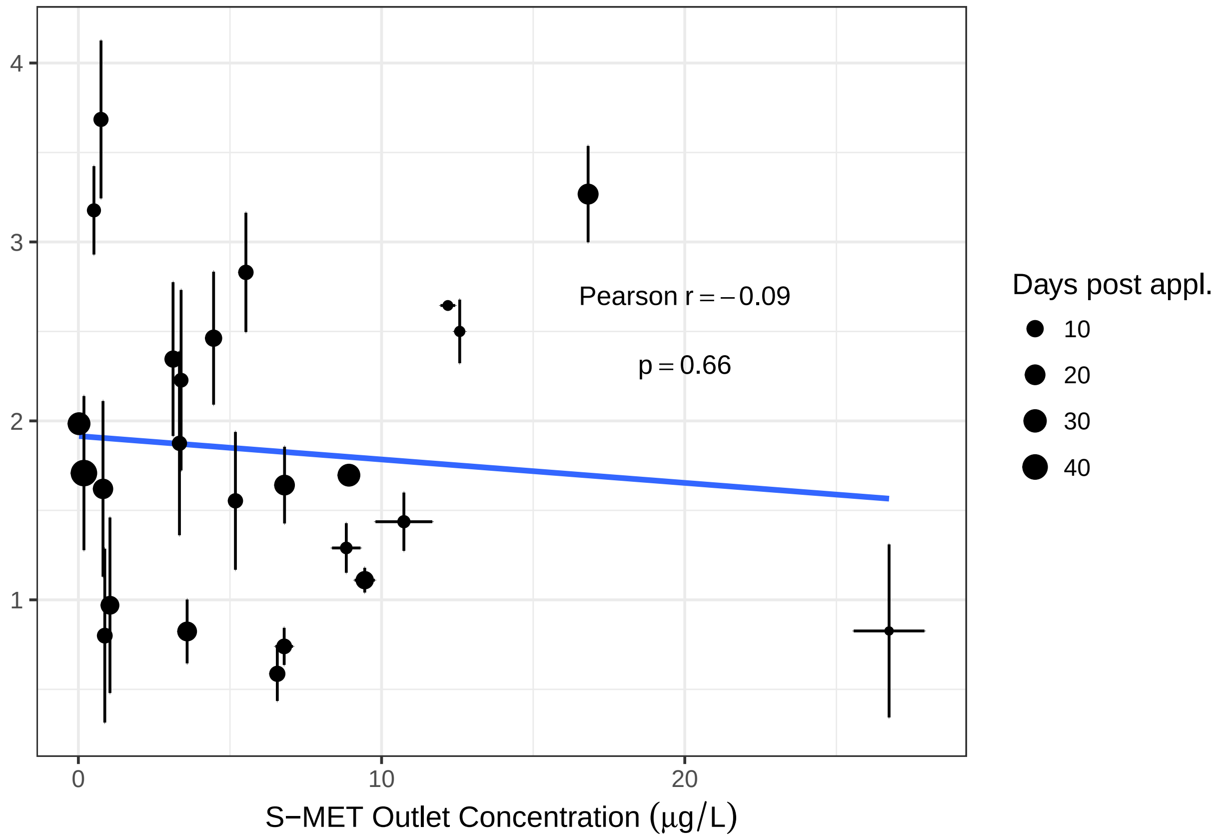
```
## [1] 0.02193412
```

```
#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

# For evidence of desorption effects, Event 7-1 (May 12th) would need to show SD < 0.54 (currently at 0
# Contingent on sample repeats
waterClean <- subset(waters, Sampled == "Sampled" & SD.d13C < 0.64) # | filt.SD.d13C <= 0.75 )

# 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), aes(x=Conc.mug.L, y=DD13C.diss))+
  stat_smooth(data = subset(waterClean),
              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()) +
  #scale_y_continuous(breaks=c(0, 1, 2, 3 , 4 ,5) ) +
  scale_y_continuous(breaks=seq(1,5,1)) +
  #ylab(expression(paste({Delta~delta}^"13","C", ' (\u2030)'))) +
  xlab(expression(paste("S-MET Outlet Concentration ", {({mu}*g / L)}))) +
  annotate("text", x = 20, y = 2.7, label = as.character(water_r_label), parse = T, size = 3.5) +
```

```r
  annotate("text", x = 20, y = 2.3, label = water_p_label, parse = T, size = 3.5)


waterIsoConc
```




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

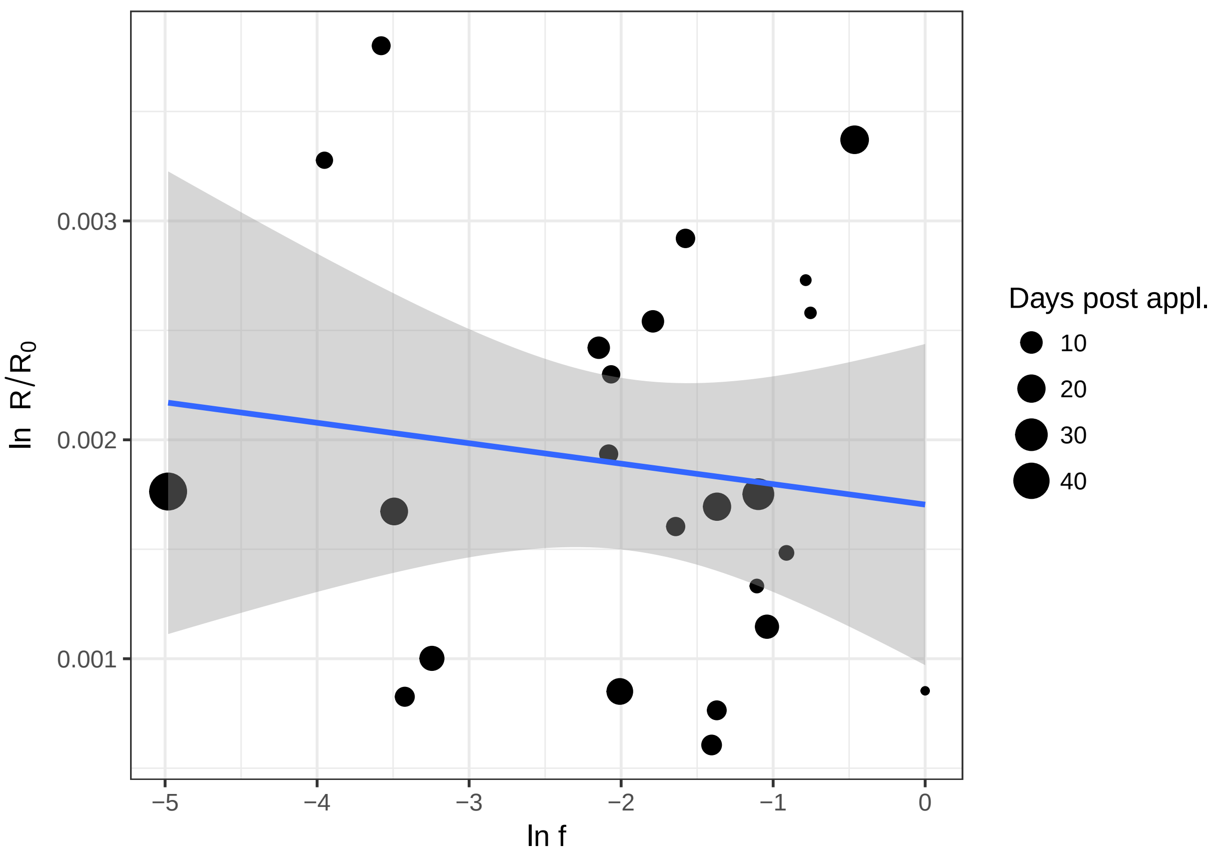
Water Rayleigh

```r
waterModel<-lm(yRaleigh~xRaleigh, data= waterClean)
summary(waterModel)
```

```
##
## Call:
## lm(formula = yRaleigh ~ xRaleigh, data = waterClean)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0012355 -0.0006707 -0.0001778  0.0006734  0.0018120
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0017468  0.0003064   5.702 8.33e-06 ***
## xRaleigh    -0.0000675  0.0001157  -0.583    0.565
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0009053 on 23 degrees of freedom
## Multiple R-squared:  0.01458,    Adjusted R-squared:  -0.02826
## F-statistic: 0.3403 on 1 and 23 DF,  p-value: 0.5653
```

```
minX <- confint(waterModel, "xRaleigh", level = 0.95)[1]*1000
maxX <- confint(waterModel, "xRaleigh", level = 0.95)[2]*1000

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

CI95 = maxX - cofwater


waterRaleigh <- ggplot(data = subset(waterClean, (!is.na(yRaleigh) & xRaleigh > -7)), aes(x=xRaleigh, y=
  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)), method = "lm", formula = y~

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
```

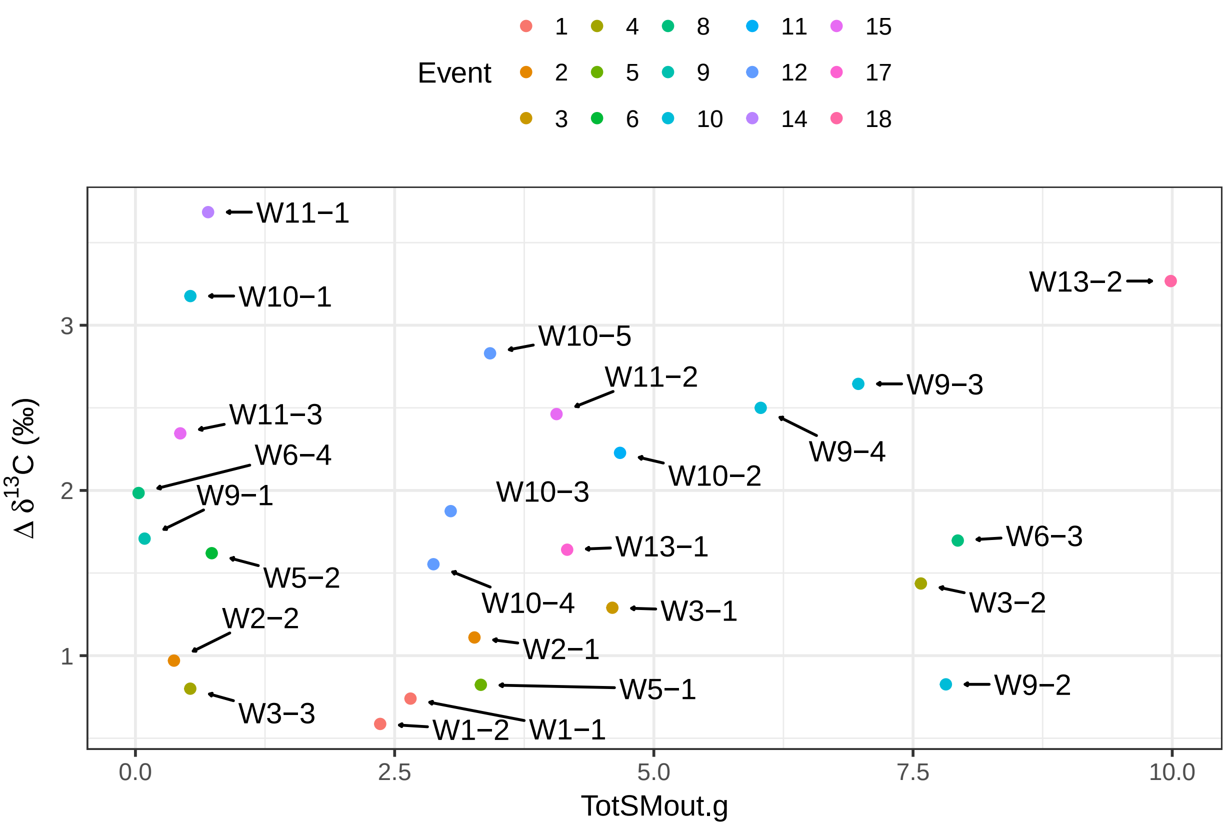
```
## [1] TRUE
```

```r
str(waters)
```

```
## 'data.frame':    51 obs. of  116 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 ...
##  $ dryHrsIni              : num  0.0167 0.0333 0.2667 0.1167 4.2167 ...
##  $ dryHrsMax              : num  2.75 24.52 13.32 4.2 5.43 ...
##  $ dryHrsAve              : num  0.745 7.827 4.859 1.289 1.314 ...
##  $ noEventHrsIni          : num  0.0167 6.0167 47.3 66.1333 1.65 ...
##  $ noEventHrsMax          : num  6 47.28 66.12 72.1 6.37 ...
##  $ noEventHrsAve          : num  3.01 26.65 56.71 30.4 3.33 ...
##  $ 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 ...
##  $ Markers                : num  NA NA NA 16.9 NA ...
##  $ TimeDiff               : Factor w/ 18 levels "106","136","150",..: NA NA NA 10 NA NA 2 NA NA 11 ..
##  $ 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 ...
##  $ CumRain.mm             : num  2.8 7.6 7.6 16.8 6 9.4 5.4 0.8 5.4 20 ...
##  $ RainInt.mmhr           : num  0.234 0.0921 0.2019 0.6161 0.2594 ...
##  $ 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 ...
##  $ N.x                    : int  NA NA NA 3 3 NA 3 5 NA 3 ...
##  $ diss.d13C              : num  NA NA NA -31.5 -31.7 ...
##  $ SD.d13C                : num  NA NA NA 0.104 0.152 ...
##  $ N_d13C.diss            : int  NA NA NA 3 3 NA 3 5 NA 3 ...
##  $ MES.mg.L               : num  NA 53.4 NA 62.5 22.5 ...
##  $ MES.sd                 : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ MO.mg.L                : num  NA 0 NA 0.001 0.0001 NA 0.0001 0.0001 NA 0.0058 ...
##  $ 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 ...
##  $ N.y                    : int  NA NA NA NA NA NA 3 3 NA NA ...
##  $ filt.d13C              : num  NA NA NA NA NA ...
##  $ filt.SD.d13C           : num  NA NA NA NA NA ...
##  $ DD13C.diss             : num  NA NA NA 0.74 0.587 ...
##  $ DD13C.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 ...
##  $ DissSmeto.mg          : num  3.54 24.6 170.04 2649.91 2357 ...
##  $ DissSmeto.mg.SD       : num  0.278 1.934 7.463 112.98 68.486 ...
##  $ 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.mg            : num  69.5 483.2 854.7 11918.4 11672.7 ...
##  $ DissOXA.mg.SD         : num  16.5 114.3 273.8 3976 87.3 ...
##  $ 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.mg            : num  260 1808 1548 17951 14830 ...
##  $ DissESA.mg.SD         : num  50.4 350.3 158 1185.5 306.6 ...
##  $ 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          : num  3.45 3.45 5.73 3.07 3.52 ...
##  $ 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.mg           : num  6.99 28.06 175.77 2652.98 2360.52 ...
##  $ TotSMout.mg.SD        : num  0.216 1.37 5.284 79.89 48.432 ...
##  $ 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 ...
##  $ FracDiss              : num  0.506 0.877 0.967 0.999 0.999 ...
##  $ FracFilt              : num  0.49352 0.12301 0.03261 0.00116 0.00149 ...
##  $ 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 ...
##  $ Appl.Mass.g           : num  31670 0 0 0 0 ...
##  $ Appl.Mass.g.OT        : num  24477 0 0 0 0 ...
##  $ Appl.Mass.g.N         : num  8429 0 0 0 0 ...
##  $ Appl.Mass.g.T         : num  6904 0 0 0 0 ...
##  $ Appl.Mass.g.S         : num  16337 0 0 0 0 ...
##  $ Appl.Mass.g.N.OT      : num  8429 0 0 0 0 ...
##  $ Appl.Mass.g.T.OT      : num  2727 0 0 0 0 ...
##  $ Appl.Mass.g.S.OT      : num  13321 0 0 0 0 ...
##  $ iniCo.ug.g.N          : num  8.46 8.46 8.46 8.46 8.46 ...
##  $ iniCo.ug.g.T          : num  7.09 7.09 7.09 7.09 7.09 ...
##  $ iniCo.ug.g.S          : num  12.4 12.4 12.4 12.4 12.4 ...
##  $ timeSinceApp          : num  0.5 3.9 5.5 6.6 7.6 11.6 12.6 14 20.6 2.2 ...
##  $ timeSinceApp.N        : num  0.5 3.9 5.5 6.6 7.6 11.6 12.6 14 20.6 2.2 ...
##  $ timeSinceApp.T        : num  0.5 3.9 5.5 6.6 7.6 11.6 12.6 14 20.6 2.2 ...
##  $ timeSinceApp.S        : 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  31670 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 ...
##   [list output truncated]
```

```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)
```



Correlations Waters

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

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

```
## t = -0.44926, df = 23, p-value = 0.6574
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.4710402  0.3134125
## sample estimates:
##         cor
## -0.09326873
```

```
#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

Objective is to plot both soils and water temporaly

Outlet Isotope Shifts (DD)

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

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

The equation for the secondary y-axis will be:

$$B = (1 - (\frac{1000 + \delta^{13}C_0 + \Delta\delta^{13}C}{1000 + \delta^{13}C_0})^{\frac{1000}{\epsilon}}) * 100$$

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

```
# SD min. selection line 914 (for dissolved)
WaterSoils <- merge(waterClean, soils, by = "Date.ti", all = T)


# Choose and rearrange variables
# names(WaterSoils)
wsSmall <- WaterSoils[c("Date.ti", "WeekSubWeek", "ID.N", "Event.y", "Events",
                     "maxQ", "AveDischarge.m3.h",
                     "dryHrsIni", "dryHrsMax", "dryHrsAve", "noEventHrsIni", "noEventHrsMax", "noEventH
                      "CumRain.mm", "RainInt.mmhr", ## Rainfall is per subsample (See Book 3)
                      "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)
```

```
##  [1] "Date.ti"            "WeekSubWeek"        "ID.N"
##  [4] "Event.y"            "Events"             "maxQ"
##  [7] "AveDischarge.m3.h"  "dryHrsIni"          "dryHrsMax"
```

```
## [10] "dryHrsAve"           "noEventHrsIni"       "noEventHrsMax"
## [13] "noEventHrsAve"       "CumRain.mm"          "RainInt.mmhr"
## [16] "DD13C.diss"          "SD.d13C.x"           "DD13C.filt"
## [19] "filt.SD.d13C"        "DD13C.Talweg"        "comp.d13C.SD.Talweg"
## [22] "DD13C.South"         "comp.d13C.SD.South"  "DD13C.North"
## [25] "comp.d13C.SD.North"  "DD13.Bulk"           "BulkCatch.d13.SD"
```

```
keepCorrTest <- c("DD13C.diss",
                  "DD13C.Talweg",
                  "DD13C.South",
                  "DD13C.North",
                  "DD13.Bulk")

wsTest <- wsSmall[ , (names(wsSmall) %in% keepCorrTest)]

names(wsSmall) <- c("Date", "Week", "IDSoil", "Event", "Events",
                    "Qmax", "Qmean",
                    "dryHrsIni", "dryHrsMax", "dryHrsAve", "noEventHrsIni", "noEventHrsMax", "noEventHrs
                    "CumRain", "RainInt", ## Rainfall is per subsample (See Book 3)
                    "diss.measure", "diss.SD",
                    "filt.measure", "filt.SD",
                    "Talweg.measure", "Talweg.SD",
                    "South.measure", "South.SD",
                    "North.measure", "North.SD",
                    "BulkDD.measure", "BulkDD.SD"
                    )

wsTest <- wsTest[7:length(wsTest$DD13C.diss) , ]
wsTest$DD13.Bulk <- na.locf(wsTest$DD13.Bulk)
wsTest$DD13C.Talweg <- na.locf(wsTest$DD13C.Talweg)
wsTest$DD13C.South <- na.locf(wsTest$DD13C.South)
wsTest$DD13C.North <- na.locf(wsTest$DD13C.North)

cor.test(wsTest$DD13.Bulk, wsTest$DD13C.diss, method = "pearson", use = "pairwise.complete.obs")
```

```
##
##  Pearson's product-moment correlation
##
## data:  wsTest$DD13.Bulk and wsTest$DD13C.diss
## t = 4.3922, df = 21, p-value = 0.0002544
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3913776 0.8591130
## sample estimates:
##       cor
## 0.6919498
```

```
cor.test(wsTest$DD13C.Talweg, wsTest$DD13C.diss, method = "pearson", use = "pairwise.complete.obs")
```

```
##
##  Pearson's product-moment correlation
##
## data:  wsTest$DD13C.Talweg and wsTest$DD13C.diss
## t = 1.5, df = 21, p-value = 0.1485
## alternative hypothesis: true correlation is not equal to 0
```

```
## 95 percent confidence interval:
##  -0.1159932  0.6410798
## sample estimates:
##       cor
## 0.3110826
```

```r
cor.test(wsTest$DD13C.North, wsTest$DD13C.diss, method = "pearson", use = "pairwise.complete.obs")
```

```
##
##  Pearson's product-moment correlation
##
## data:  wsTest$DD13C.North and wsTest$DD13C.diss
## t = 3.5821, df = 21, p-value = 0.001756
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2729387 0.8199153
## sample estimates:
##       cor
## 0.6158537
```

```r
cor.test(wsTest$DD13C.South, wsTest$DD13C.diss, method = "pearson", use = "pairwise.complete.obs")
```

```
##
##  Pearson's product-moment correlation
##
## data:  wsTest$DD13C.South and wsTest$DD13C.diss
## t = 3.7976, df = 21, p-value = 0.001053
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3064826 0.8315629
## sample estimates:
##      cor
## 0.638075
```

```r
# 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, -Events, -Week,
         -Qmax, -Qmean,
         -CumRain, -RainInt,
         -dryHrsIni, -dryHrsMax, -dryHrsAve,
         -noEventHrsMax,
         -noEventHrsIni, -noEventHrsAve) %>% # Melts data frame
  separate(measure, into = c("Location", "temporary_var")) %>% # parses the sep = "." into...
  # Location will be first string of variable name
  spread(temporary_var, value)

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

```r
                        "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 (Outlet)" "Sediment"           "Top Soil"
```

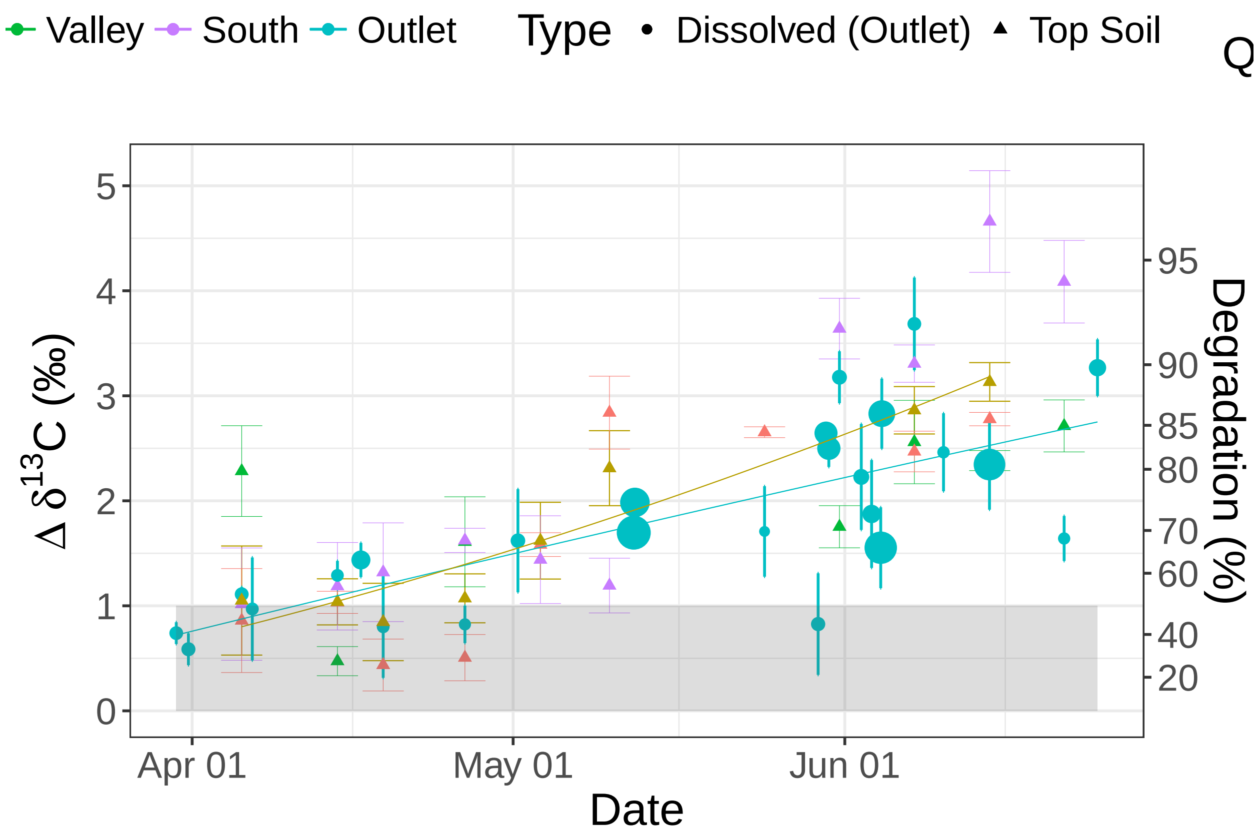
```r
#wstidier$Source <- factor(wstidier$Source, levels = c("Bulk", "Plateau","Valley", "Outlet"))
wstidier$Source <- factor(wstidier$Source, levels = c("Bulk", "North", "Valley", "South", "Outlet"))

wstidier$Type <- factor(wstidier$Type, levels = c("Top Soil", "Dissolved (Outlet)", "Sediment" ))


# epsilon
#epsilon_field
#initialDelta

#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))
```

```r
# library(scales)
## Color palette
# show_col(hue_pal()(12))

# Bulk, North, Valley, South, Outlet
# "black", "#F8766D", "#00BA38", "#DE8C00", ("#619CFF" / "#00B4F0" / "#00BFC4")
```

Lab Enrichment plot

```
# Dissolved has been selected, but not soils or filters
wstidier2 = subset(wstidier, SD <= 0.54 & Date <= as.POSIXct('2016-06-24 14:52:00', tz = "EST"))

mindate = min(wstidier2$Date)
maxdate = max(wstidier2$Date)

pd <- position_dodge(width = 0.5)
limits_DdC <- aes(ymin=measure-SD, ymax=measure+SD, colour = Source)

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 == 'South'
                            | Source == 'North'
                            | Source == 'Valley'
                            ), limits_DdC, size=0.1) +
  geom_errorbar(data=subset(wstidier2, Source == 'Outlet'), limits_DdC) +
  geom_point(data=subset(wstidier2, (Source == 'South'
                                     | Source == 'North'
                                     | Source == 'Valley'
                                     )
                         # & Date > as.POSIXct('2016-05-14 08:04:00')
                         ),
             aes(shape = Type,
                 colour = Source)) +
  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)) +

  # Water
  stat_smooth(data=subset(wstidier2,
                          (Source == "Outlet"
                           # & Event > 1
                           & Type == "Dissolved (Outlet)")),
             method = "lm", formula = y ~ poly(x, 2), se = F, 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 = F, 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() +

  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(text = element_text(size=17),
        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") +
```

```r
  #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",
                     # Actual order:
                     # Bulk, North, Outlet, South, Valley
                      values = c("#B79F00", "#F8766D", "#00BFC4", "#C77CFF", "#00BA38"
                                 # working solution:
                                 #c("black", "#F8766D", "#00BFC4", "#DE8C00", "#00BA38"
                                 #"black", "#D55E00",  "#00BFC4",  "#B79F00", "#00BA38"
                                 # Bulk, North, outlet, South, Valley
                                 #"#D55E00", "darkgreen", "dodgerblue"
                                 ),
                     breaks=c("Bulk", "North" , "Valley" ,"South",  "Outlet"),
                     labels=c("Bulk", "North" , "Valley" ,"South",  "Outlet")
                     ) +

  scale_size_continuous(range = c(1, 6), breaks= c(0, 50, 100, 150, 200, 300), limits = c(0, 300))+
  annotate("rect", xmin=mindate, xmax=maxdate, ymin=0, ymax=1, alpha=0.2)
  # scale_size_continuous(range = c(1, 3))

#
# Reds
# gold = "#B79F00"
# red-pink  = "#F8766D"
# "firebrick1",
# 'yellow', "orange1","red",
# pink = "#F564E3"

# Mono
# "gray35", "ghostwhite", 'gray99'

# Greens
# 'darkgreen','darkolivegreen3','darkseagreen3','darkseagreen1'
# dark green = "chartreuse4"
# darkish freen = "#00BA38"

# Blues
# purple = "blueviolet"
# "dodgerblue", "#00BFC4" (light blue), "#619CFF" (sharp blue),
#  "deepskyblue"

wsALL_lab
```



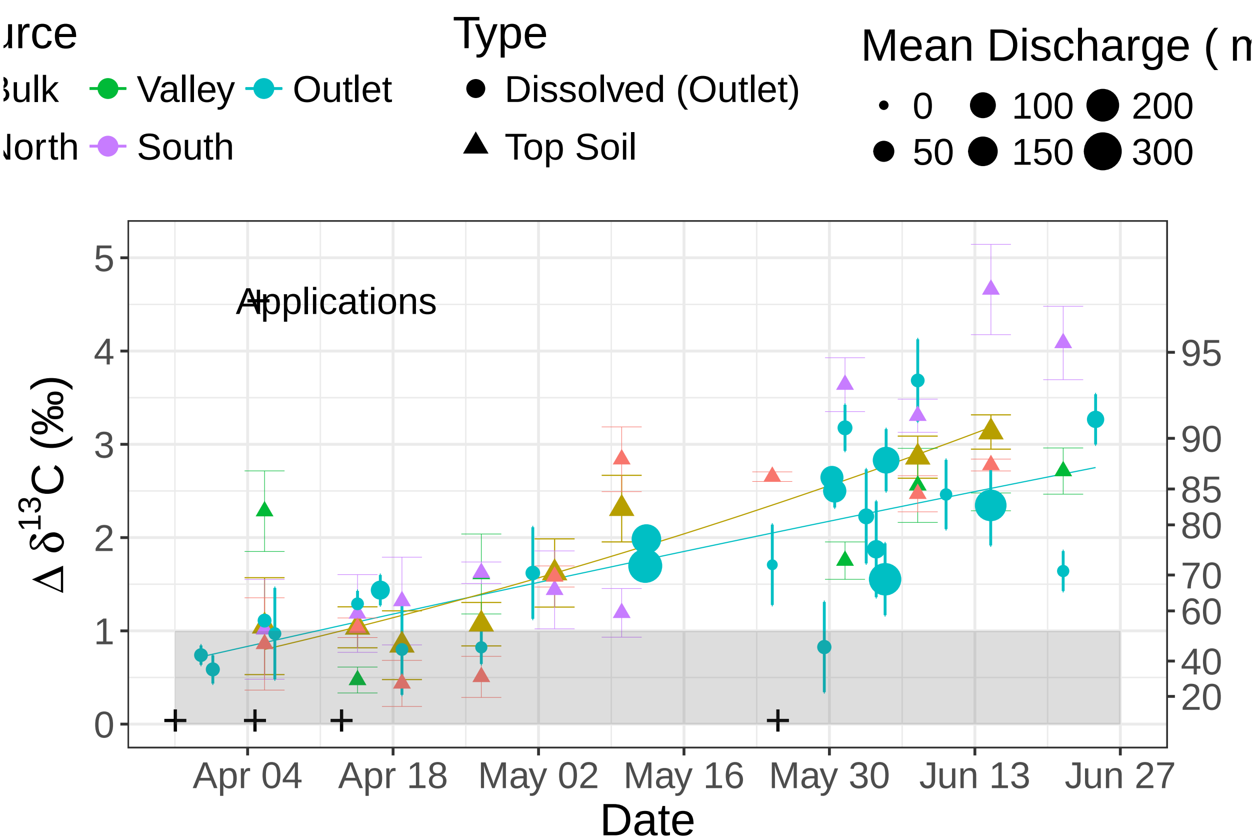

```
# ggplotly(wsALL_lab)
```

Field Enrichment Plot

```
mindate = as.POSIXct("2016-03-28 00:04:00" , tz = "EST") # min(wstidier2$Date)
maxdate = as.POSIXct("2016-06-27 00:01:00", tz = "EST")

wsALL_field <- ggplot(data = wstidier2, aes(x = Date, y = measure, group = Source) )+
  # Dissolved (Outlet) trend
  stat_smooth(data=subset(wstidier2,
                         (Source == "Outlet"
                          # & Event > 1
                          & Type == "Dissolved (Outlet)")),
              method = "lm", formula = y ~ poly(x, 2), se = F, aes(colour = 'Outlet'), alpha = 0.9, siz
  # Bulk trend
  stat_smooth(data=subset(wstidier2,
                         (Source == "Bulk" )), #| Source == "South" )),
              method = "lm", formula = y ~ poly(x, 2), se = F, aes(colour = 'Bulk'), alpha = 0.9, size=
  # Error bars
  geom_errorbar(data=subset(wstidier2, Source == 'Bulk'), limits_DdC, size=0.2) +
  geom_errorbar(data=subset(wstidier2, Source == 'South'
                          | Source == 'North'
                          | Source == 'Valley'
                          ), limits_DdC, size=0.1) +
```

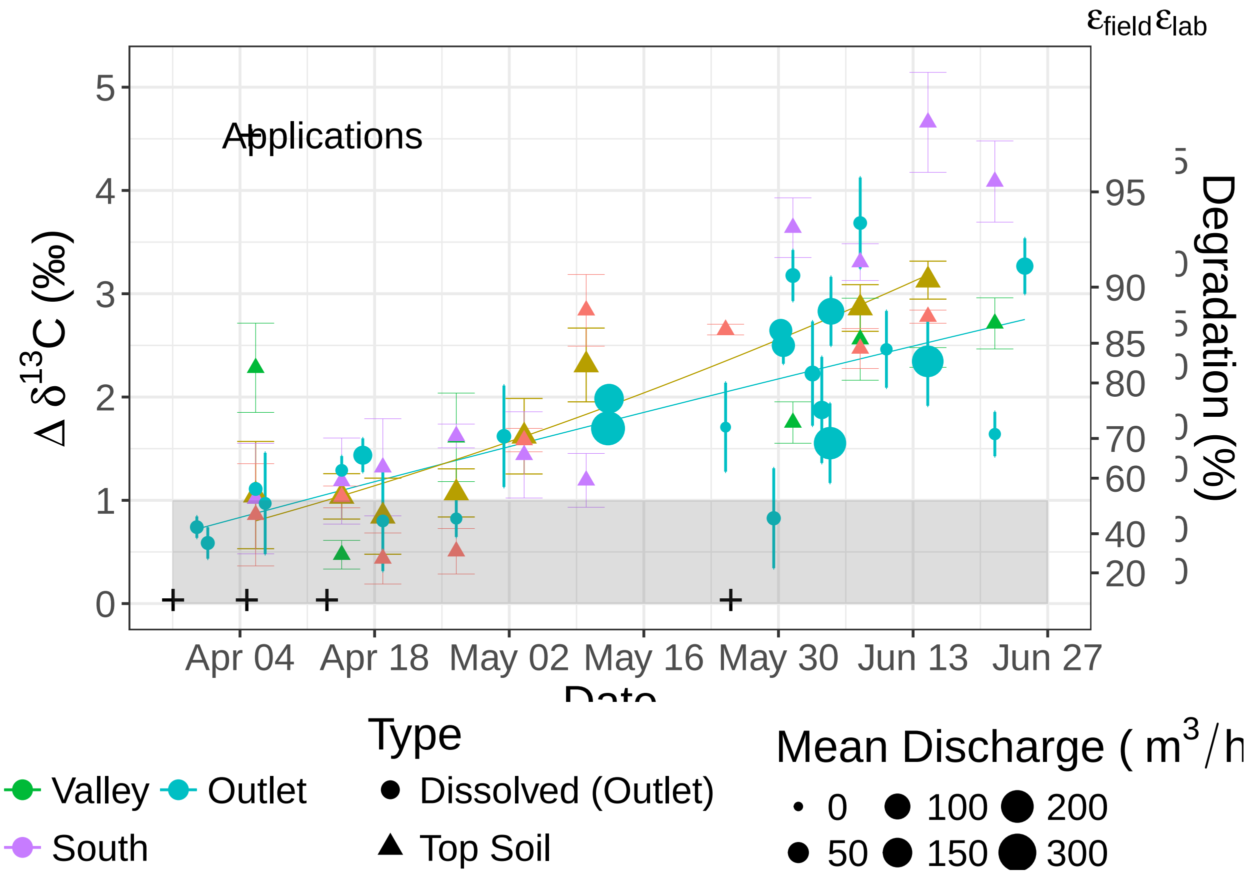
```r
geom_errorbar(data=subset(wstidier2, Source == 'Outlet'), limits_DdC) +
# Data points
geom_point(data=subset(wstidier2, Source == 'Bulk'), aes(shape = Type, colour = Source), size=3) +
geom_point(data=subset(wstidier2, (Source == 'South'
                                  | Source == 'North'
                                  | Source == 'Valley'
                                  )), aes(shape = Type, colour = Source), size=2) +
geom_point(data=subset(wstidier2, Source == 'Outlet'), aes(shape = Type, colour = Source, size = Qmean
theme_bw() +
# Applications
annotate("text", x = as.POSIXct('2016-03-28 08:04:00'), y = 0,
         label = as.character(expression(paste( "+"))), parse = T, size = 6.0) +
annotate("text", x = as.POSIXct('2016-04-05 00:04:00'), y = 0,
         label = as.character(expression(paste( "+"))), parse = T, size = 6.0) +
annotate("text", x = as.POSIXct('2016-04-13 08:04:00'), y = 0,
         label = as.character(expression(paste( "+"))), parse = T, size = 6.0) +
annotate("text", x = as.POSIXct('2016-05-25 08:04:00'), y = 0,
         label = as.character(expression(paste( "+"))), 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( "+"))), parse = T, size = 6.0) +
 annotate("text", x = as.POSIXct('2016-04-12 08:04:00'), y = 4.5,
         label = as.character(expression(paste(" Applications"))), parse = T, size = 5.0) +
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(text = element_text(size=17),
      legend.position="top"
      # axis.title.x = element_blank()
      # axis.text.x=element_text(angle = 45, hjust = 1)
      ) +
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("#B79F00", "#F8766D", "#00BFC4", "#C77CFF", "#00BA38"
                              # c("black", "#F8766D", "#00BFC4", "#DE8C00", "#00BA38"
                              #  "black", "#D55E00",  "#00BFC4",  "#B79F00", "#00BA38"
                               # Bulk, North, outlet, South, Valley
                               ),
                   breaks=c("Bulk", "North" , "Valley" ,"South",  "Outlet"),
                   labels=c("Bulk", "North" , "Valley" ,"South",  "Outlet")
                  ) +
scale_size_continuous(range = c(1, 6), breaks= c(0, 50, 100, 150, 200, 300), limits = c(0, 300)) +
# scale_size_continuous(range = c(1, 3)) +
guides(col = guide_legend(order = 1,
                          #title=expression("Source"),
                          #title.vjust = -1,
```

```
                              nrow = 2,
                              title.position = "top",
                              keyheight = 1.5
                              ),
         shape=guide_legend(title=("Type"),
                              order = 2,
                              nrow=2,
                              title.position = "top",
                              keyheight = 1.5, title.vjust = NULL, label.vjust = NULL
                              ),
         size = guide_legend(order = 3,
                               #title=expression("Mean Discharge"),
                               title=expression("Mean Discharge (" ~m^3 / h~")" ),
                               nrow=2,
                               title.position = "top"
                               # title.vjust = .26
                               #keyheight = 0,
                               #label.vjust = 0
                                )) +
  annotate("rect", xmin=mindate, xmax=maxdate, ymin=0, ymax=1, alpha=0.2)

#ggplotly(wsALL_field)
wsALL_field
```



Join all figures

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

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

labely1 = expression(epsilon ["field"])
labely2 = expression(epsilon ["lab"])

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, " \u00B1 ", "0.53" ,"\u2030"),
                     list(epsilon_f = signif(epsilon_field, 3)))

label3 <- substitute(paste(epsilon ["lab"] , " = ", epsilon_l, " \u00B1 ", "0.47" ,"\u2030"),
                     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 = 0.15, width = 1, height = 0.82) + # bottom
  draw_plot(wsALL_field_noLeg, x=0, y=.15, width = 0.945, height = .82) + # top
  draw_label(label2, x= .886, y = .10, size = 15) + # Epsilon field (bottom)
  draw_label(label3, x= .89, y = .05, size = 15) + # Epsilon lab (bottom)
  draw_label(labely1 , x= .90, y = .98, size = 14) + # Epsilon field (top)
  draw_label(labely2 , x= .95, y = .98, size = 14) + # Epsilon lab (top)
  draw_plot(wsAll_field_Leg, x=0.2, y=0.0, width = 0.50, height = 0.15)

wsALL
```



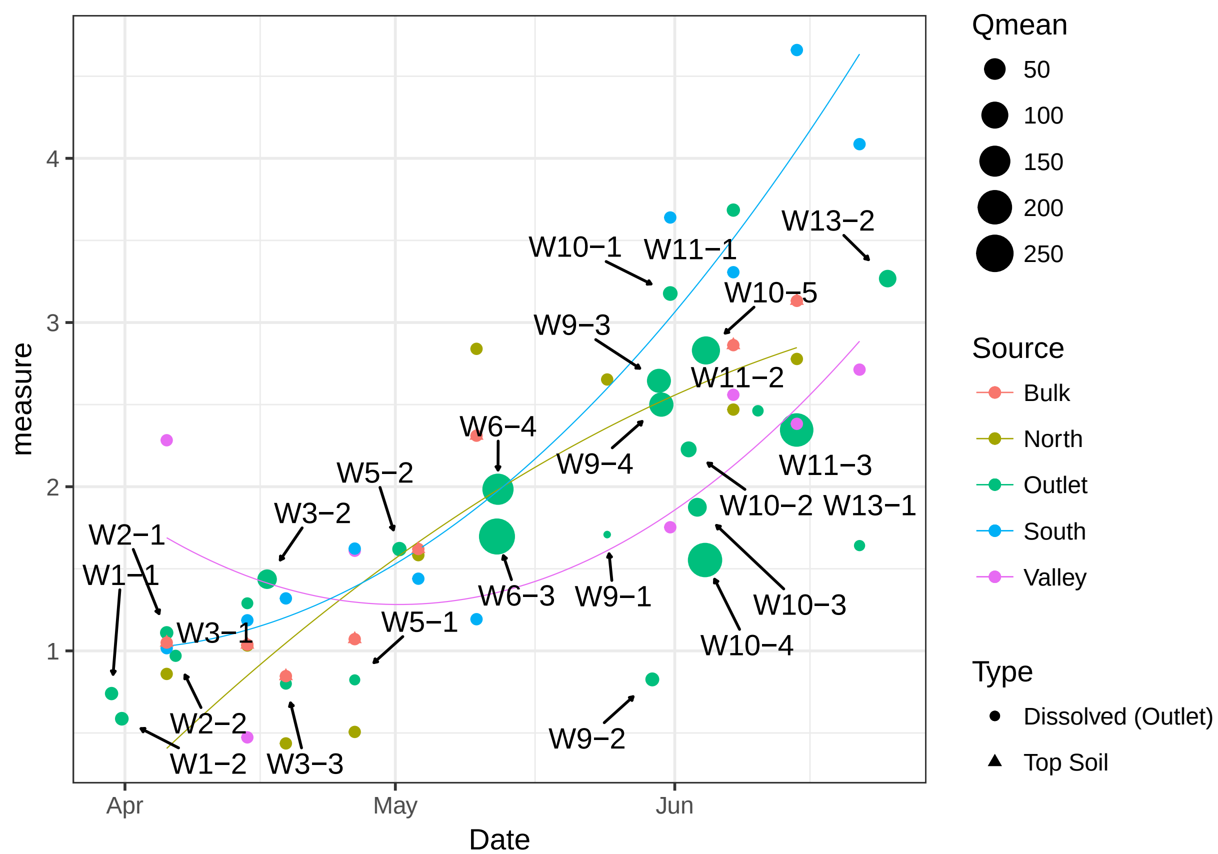

```
# ggsave(wsALL, filename = "images/WaterSoilvsTime.png", width = 8, height = 5, units = "in", scale = 1

SAVE = F
PC = T
if (SAVE){
  if (PC) {
    # cairo and ggdraw having issues.. works fine on MAC though
    ggsave(wsALL,
           filename = "D:/Documents/these_pablo/WriteUp/Alteck_PNAS_LaTeX/images/CatchOutlet.pdf",
           device= pdf, dpi = 600,  scale = 1, # )# ,
           width = 11.4, height = 6.4)
  } else {
    ggsave(wsALL,
     filename = "/Users/DayTightChunks/Documents/PhD/Writeups/PNAS/Alteck_PNAS_LaTeX/images/CatchOutlet
      device=cairo_pdf, dpi = 600,  scale = 1, # )# ,
       width = 11.4, height = 6.4)
  }
}


#install.packages("extrafont")
#library(extrafont)
```

Encodings

```
#pdf('test.pdf',encoding="MacRoman")
#plot.new()
#text(0,labels="\u2030")
#dev.off()
```

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 (Outlet)'), aes(shape = Type, colour = Source, si
  geom_point(data=subset(wstidier2, Type == 'Top Soil'), aes(colour = Source)) +

  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=Soil.ID),
  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)
```




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

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

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

```
## [1] 10
```

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

```
## [1] 7
```

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

```
## [1] 8
```

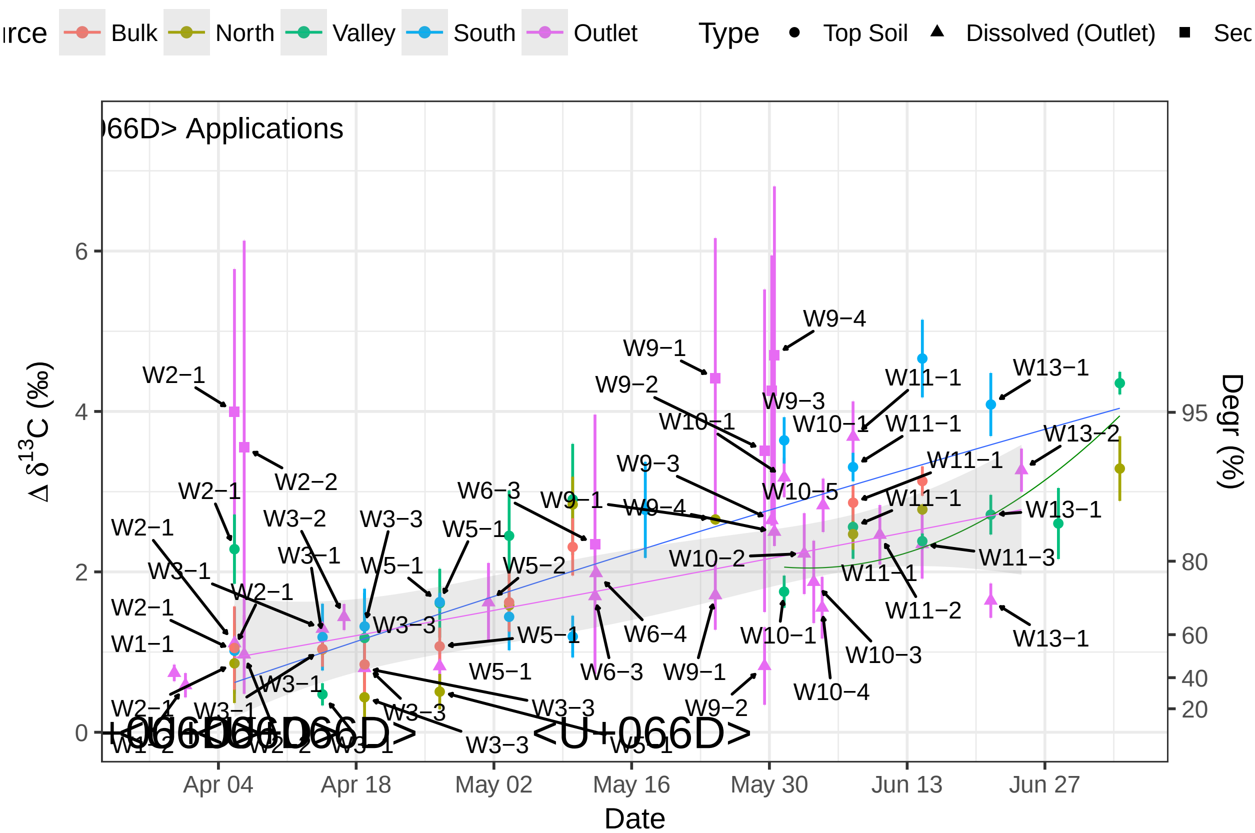
```r
write.csv2(wstidier2,
           'Data/OutletData4Lutz_R.csv', row.names = F)
```

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=0.2
  stat_smooth(data=subset(wsNoPlat,
                          (Source != "Outlet" & Source != "Valley" & Event < 20 )),
              method = "lm", formula = y ~  poly(x, 2), se = F, alpha = 0.1, size=0.2) +
  stat_smooth(data=subset(wsNoPlat,
                          (Source == "Outlet" & Event > 1 & Type == "Dissolved (Outlet)")),
              method = "lm", formula = y ~ poly(x, 2), se = T, aes(colour = 'Outlet'), alpha = 0.2, size
  #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() +
  # 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

```
names(WaterSoils)
```

```
##   [1] "Date.ti"               "WeekSubWeek"
##   [3] "tf"                    "iflux"
##   [5] "fflux"                 "changeflux"
##   [7] "maxQ"                  "minQ"
##   [9] "dryHrsIni"             "dryHrsMax"
##  [11] "dryHrsAve"             "noEventHrsIni"
##  [13] "noEventHrsMax"         "noEventHrsAve"
```

```
##  [15] "Duration.Hrs.x"          "chExtreme"
##  [17] "Peak"                    "Markers"
##  [19] "TimeDiff"                "AveDischarge.m3.h"
##  [21] "Volume.m3"               "Sampled.Hrs"
##  [23] "Sampled"                 "CumRain.mm"
##  [25] "RainInt.mmhr"            "Conc.mug.L"
##  [27] "Conc.SD"                 "OXA_mean"
##  [29] "OXA_SD"                  "ESA_mean"
##  [31] "ESA_SD"                  "N.x"
##  [33] "diss.d13C.x"             "SD.d13C.x"
##  [35] "N_d13C.diss"             "MES.mg.L"
##  [37] "MES.sd"                  "MO.mg.L"
##  [39] "Conc.Solids.mug.gMES"    "Conc.Solids.ug.gMES.SD"
##  [41] "N.y"                     "filt.d13C"
##  [43] "filt.SD.d13C"            "DD13C.diss"
##  [45] "DD13C.filt"              "NH4.mM"
##  [47] "TIC.ppm.filt"            "Cl.mM"
##  [49] "NO3...mM"                "PO4..mM"
##  [51] "NPOC.ppm"                "TIC.ppm.unfilt"
##  [53] "TOC.ppm.unfilt"          "ExpMES.Kg"
##  [55] "DissSmeto.mg"            "DissSmeto.mg.SD"
##  [57] "DissSmeto.g"             "DissSmeto.g.SD"
##  [59] "DissOXA.mg"              "DissOXA.mg.SD"
##  [61] "DissOXA.g"               "DissOXA.g.SD"
##  [63] "DissESA.mg"              "DissESA.mg.SD"
##  [65] "DissESA.g"               "DissESA.g.SD"
##  [67] "FiltSmeto.mg"            "FiltSmeto.mg.SD"
##  [69] "FiltSmeto.g"             "FiltSmeto.g.SD"
##  [71] "TotSMout.mg"             "TotSMout.mg.SD"
##  [73] "TotSMout.g.x"            "TotSMout.g.SD.x"
##  [75] "FracDiss"                "FracFilt"
##  [77] "MELsm.g.x"               "MELsm.g.SD.x"
##  [79] "CumOutDiss.g"            "CumOutFilt.g"
##  [81] "CumOutSmeto.g.x"         "CumOutMELsm.g"
##  [83] "Appl.Mass.g.x"           "Appl.Mass.g.OT.x"
##  [85] "Appl.Mass.g.N"           "Appl.Mass.g.T"
##  [87] "Appl.Mass.g.S"           "Appl.Mass.g.N.OT"
##  [89] "Appl.Mass.g.T.OT"        "Appl.Mass.g.S.OT"
##  [91] "iniCo.ug.g.N.x"          "iniCo.ug.g.T.x"
##  [93] "iniCo.ug.g.S.x"          "timeSinceApp.x"
##  [95] "timeSinceApp.N.x"        "timeSinceApp.T.x"
##  [97] "timeSinceApp.S.x"        "Appl.Mass.g.NoSo"
##  [99] "timeSinceApp.NoSo.x"     "CumAppMass.g.x"
## [101] "CumAppMass.g.OT.x"       "CumAppMass.g.N.x"
## [103] "CumAppMass.g.T.x"        "CumAppMass.g.S.x"
## [105] "CumAppMass.g.N.OT.x"     "CumAppMass.g.T.OT.x"
## [107] "CumAppMass.g.S.OT.x"     "BalMassDisch.g"
## [109] "prctMassOut"             "FracDeltaOut"
## [111] "Events"                  "Weeks"
## [113] "Event.x"                 "yRaleigh.x"
## [115] "xRaleigh.x"              "DIa"
## [117] "Event.y"                 "Duration.Hrs.y"
## [119] "timeSinceApp.y"          "timeSinceApp.NoSo.y"
## [121] "timeSinceApp.N.y"        "timeSinceApp.T.y"
```

```
## [123] "timeSinceApp.S.y"       "diss.d13C.y"
## [125] "SD.d13C.y"              "TotSMout.g.y"
## [127] "TotSMout.g.SD.y"        "MELsm.g.y"
## [129] "MELsm.g.SD.y"           "Appl.Mass.g.y"
## [131] "Appl.Mass.g.OT.y"       "CumAppMass.g.y"
## [133] "CumAppMass.g.N.y"       "CumAppMass.g.T.y"
## [135] "CumAppMass.g.S.y"       "CumAppMass.g.OT.y"
## [137] "CumAppMass.g.N.OT.y"    "CumAppMass.g.T.OT.y"
## [139] "CumAppMass.g.S.OT.y"    "iniCo.ug.g.N.y"
## [141] "iniCo.ug.g.T.y"         "iniCo.ug.g.S.y"
## [143] "CumOutSmeto.g.y"        "MassSoil.g.North"
## [145] "MassSoil.g.SD.North"    "Conc.mug.g.dry.soil.N"
## [147] "comp.d13C.North"        "comp.d13C.SD.North"
## [149] "ID.N"                   "MassSoil.g.Talweg"
## [151] "MassSoil.g.SD.Talweg"   "Conc.mug.g.dry.soil.T"
## [153] "comp.d13C.Talweg"       "comp.d13C.SD.Talweg"
## [155] "MassSoil.g.South"       "MassSoil.g.SD.South"
## [157] "Conc.mug.g.dry.soil.S"  "comp.d13C.South"
## [159] "comp.d13C.SD.South"     "ID.S"
## [161] "DD13C.North"            "DD13C.Talweg"
## [163] "DD13C.South"            "CatchMassSoil.g"
## [165] "CatchMassSoil.g.SD"     "BulkCatch.d13"
## [167] "BulkCatch.d13.SD"       "DD13.Bulk"
## [169] "Area.Catchment"         "BulkCatch.Conc"
## [171] "iniCo.Bulk"             "yRaleigh.y"
## [173] "xRaleigh.y"
```

```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",
                             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'), "Ou
                      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'), "Va
                      ifelse( (mwsStatTest$variable == "comp.d13C.North" | mwsStatTest$variable == "comp
                               mwsStatTest$Date.ti <= as.POSIXct('2016-06-05 00:06:00', tz = 'EST'), "Pl
                      ifelse( (mwsStatTest$variable == "comp.d13C.North" | mwsStatTest$variable == "comp
                               mwsStatTest$Date.ti > as.POSIXct('2016-06-05 00:06:00', tz = 'EST'), "Pla
                             ))))))
```

```r
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 = 0.77167, df = 5, p-value = 0.9788
```

```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
```

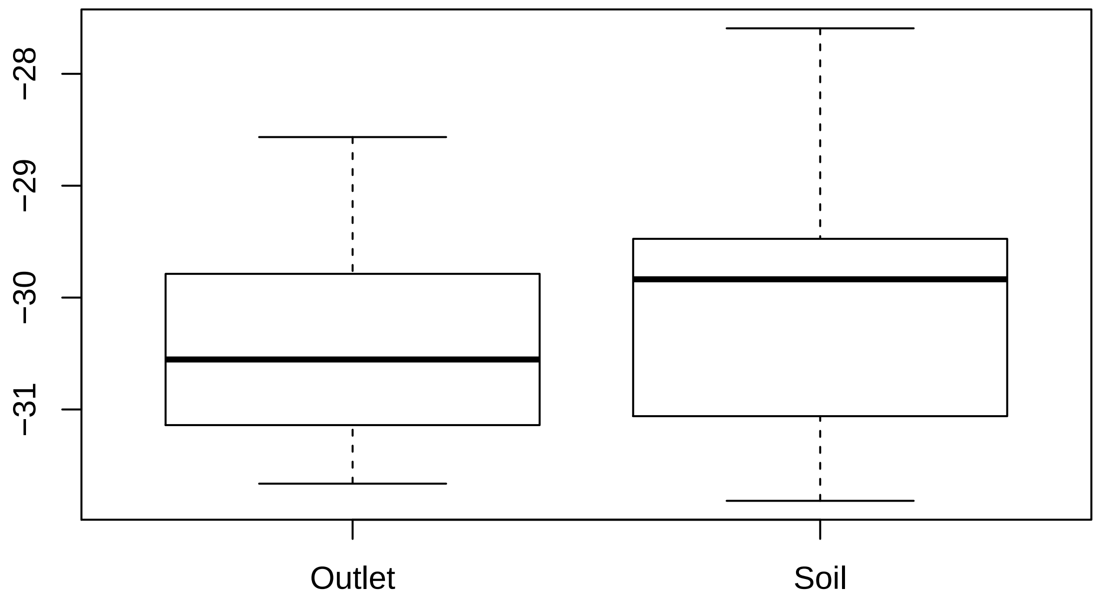
```
##
##  Kruskal-Wallis rank sum test
##
## data:  value by as.factor(Group3)
## Kruskal-Wallis chi-squared = 21.066, df = 5, p-value = 0.000787
```

```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

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



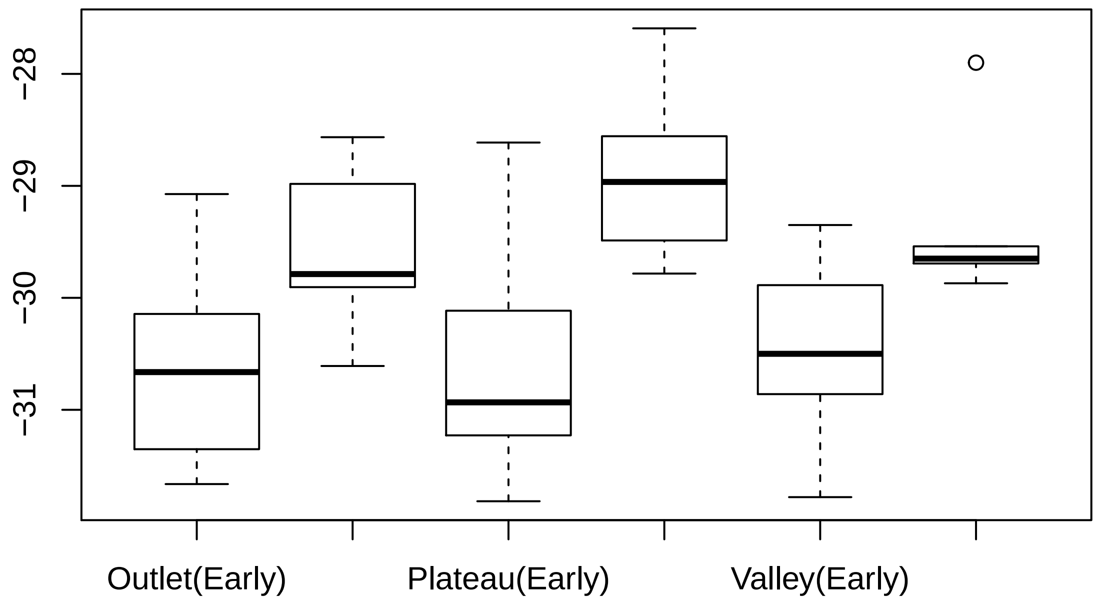

```r
summary(aov(Gr1$value ~ Gr1$Group1))
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## Gr1$Group1   1   1.89   1.889   1.833  0.181
## Residuals   57  58.75   1.031
```

```r
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 0.3620841 -0.1735138 0.897682 0.1811627
```

```r
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  24.36   4.873    7.12 3.7e-05 ***
## Residuals   53  36.27   0.684
## ---
## 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: -0.01734
##       Significance: 0.667
##
## Permutation: free
## Number of permutations: 999
##
```

```
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0384 0.0595 0.0807 0.1063
##
## Dissimilarity ranks between and within classes:
##         0%     25%   50%     75% 100%   N
## Between  1 423.250 836.5 1279.50 1708 850
## Outlet   3 413.125 748.0 1158.25 1656 300
## Soil     6 457.000 965.5 1355.50 1711 561
```

```
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)
```

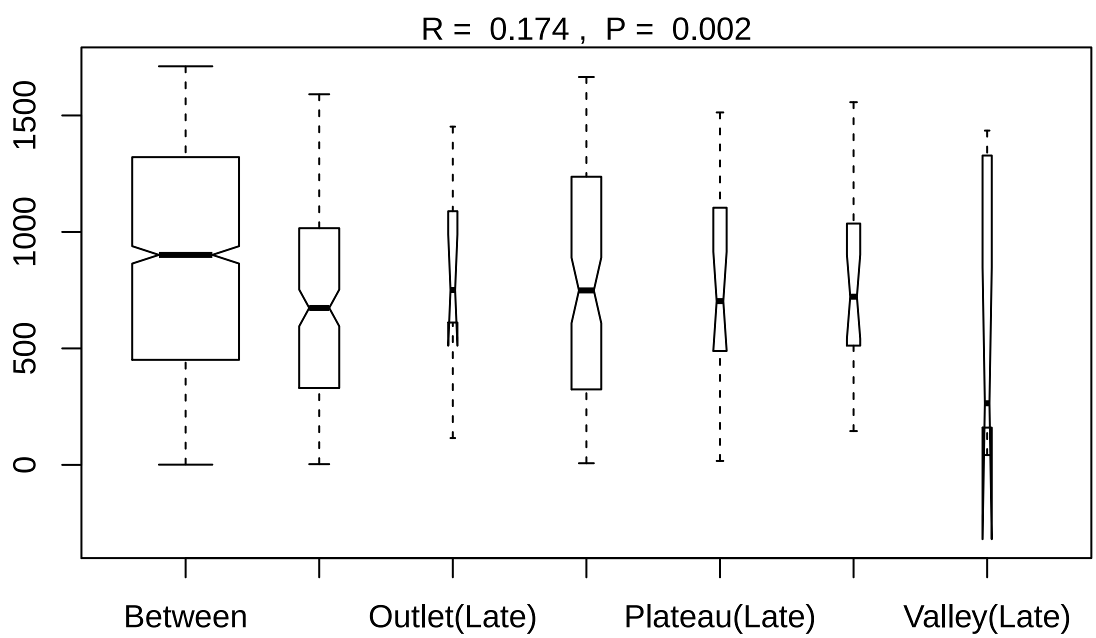
```
##
## Call:
## anosim(dat = Gr2.hell.daisy, grouping = Group2)
## Dissimilarity:
##
## ANOSIM statistic R: 0.007198
##       Significance: 0.362
##
## Permutation: free
## Number of permutations: 999
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0380 0.0566 0.0751 0.0883
##
## Dissimilarity ranks between and within classes:
##         0%     25%     50%     75% 100%    N
## Between  1 428.875  863.25 1279.50 1710 1114
## Outlet   3 413.125  748.00 1158.25 1656  300
## Plateau  6 490.500 1072.00 1447.50 1711  231
## Valley  42 316.750  774.50 1289.25 1705   66
```

```
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: 0.1736
##       Significance: 0.002
##
## Permutation: free
## Number of permutations: 999
##
```

```
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0639 0.0856 0.1039 0.1347
##
## Dissimilarity ranks between and within classes:
##                 0%    25%    50%     75% 100%    N
## Between          1 451.25 901.50 1320.75 1711 1354
## Outlet(Early)    3 332.00 673.75 1013.50 1591  190
## Outlet(Late)   115 633.75 750.50 1072.25 1452   10
## Plateau(Early)   7 324.00 749.00 1237.00 1665  105
## Plateau(Late)   17 489.00 703.00 1104.00 1513   21
## Valley(Early)  145 512.00 722.00 1036.00 1557   21
## Valley(Late)    42 164.25 264.50 1314.00 1435   10
```

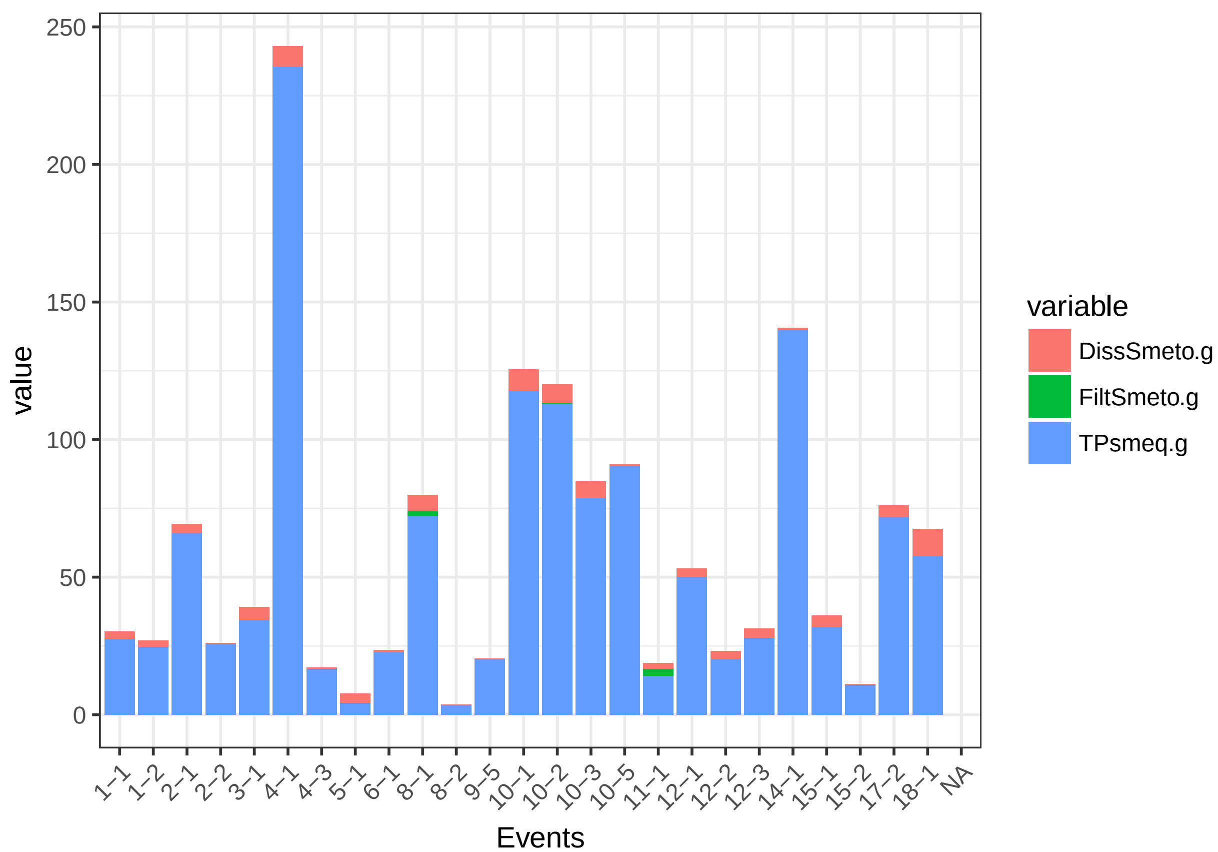
```
plot(anosim.group3)
```



Loadings

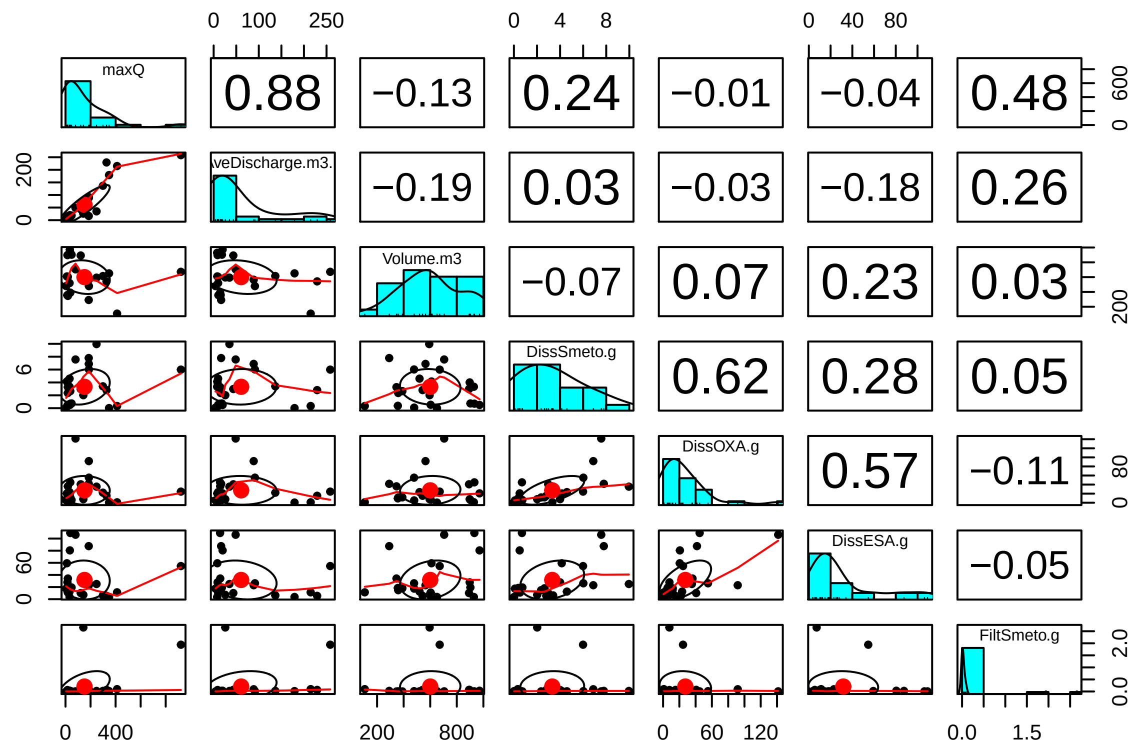
```
keepLoads <- c("Date.ti",
             "DissOXA.g", "DissESA.g", "DissSmeto.g", "FiltSmeto.g",
             "Event.x", "Events")
wsLoads <- WaterSoils[ , (names(WaterSoils) %in% keepLoads)]

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

```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)
        )
```

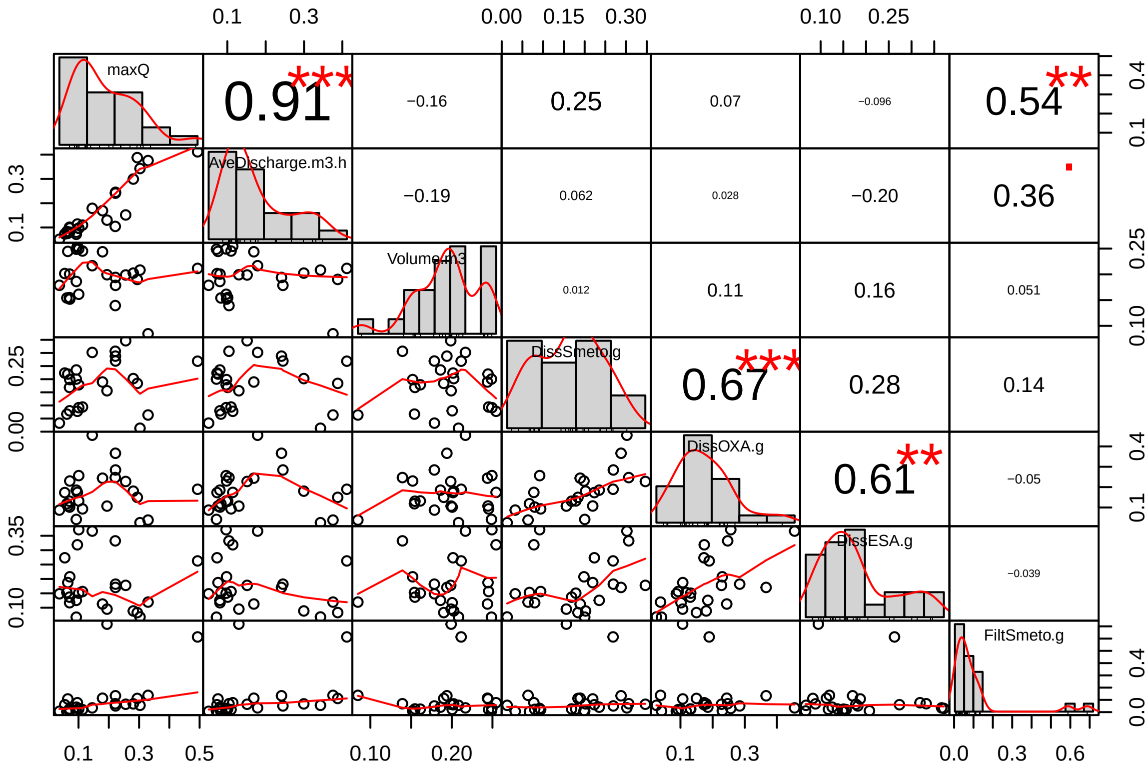



```r
  # geom_bar(stat = "identity", position = position_dodge())

# 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:

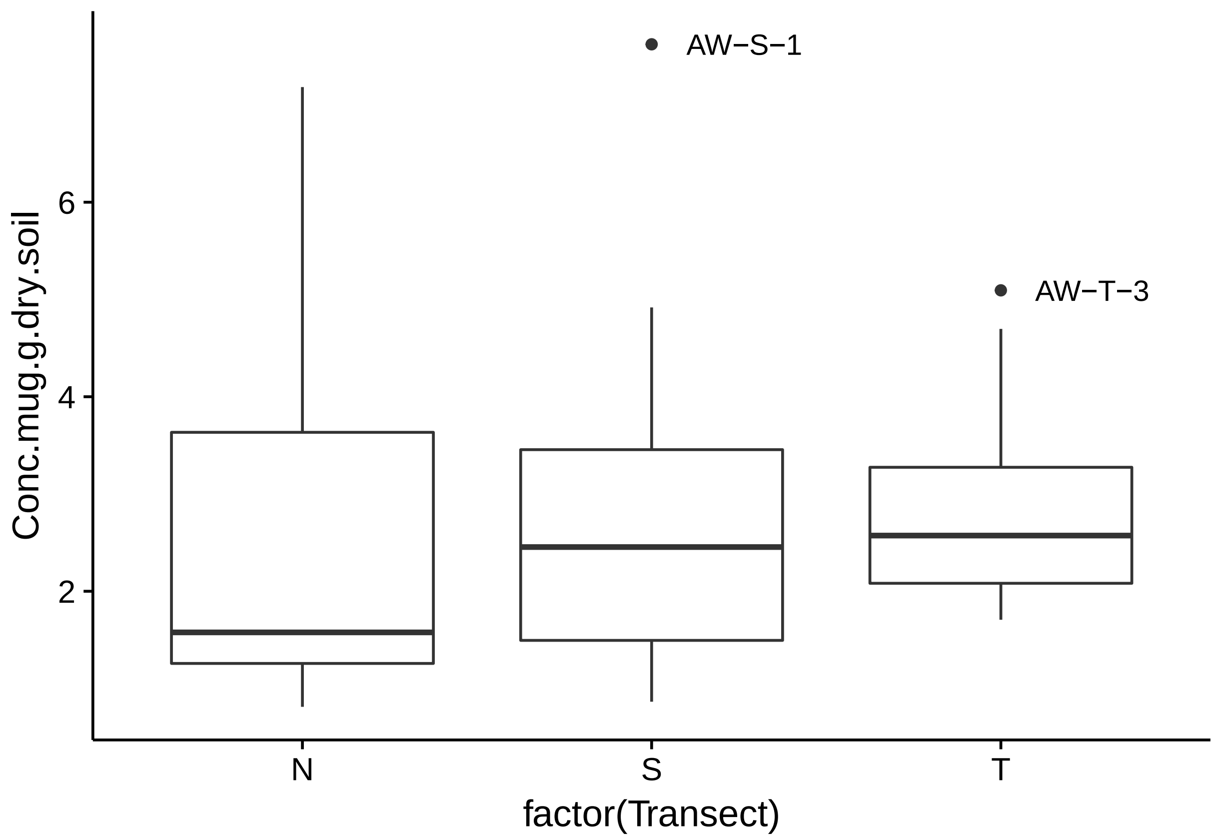
$$z_i = \frac{y_i - \bar{y}}{s_y}$$

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

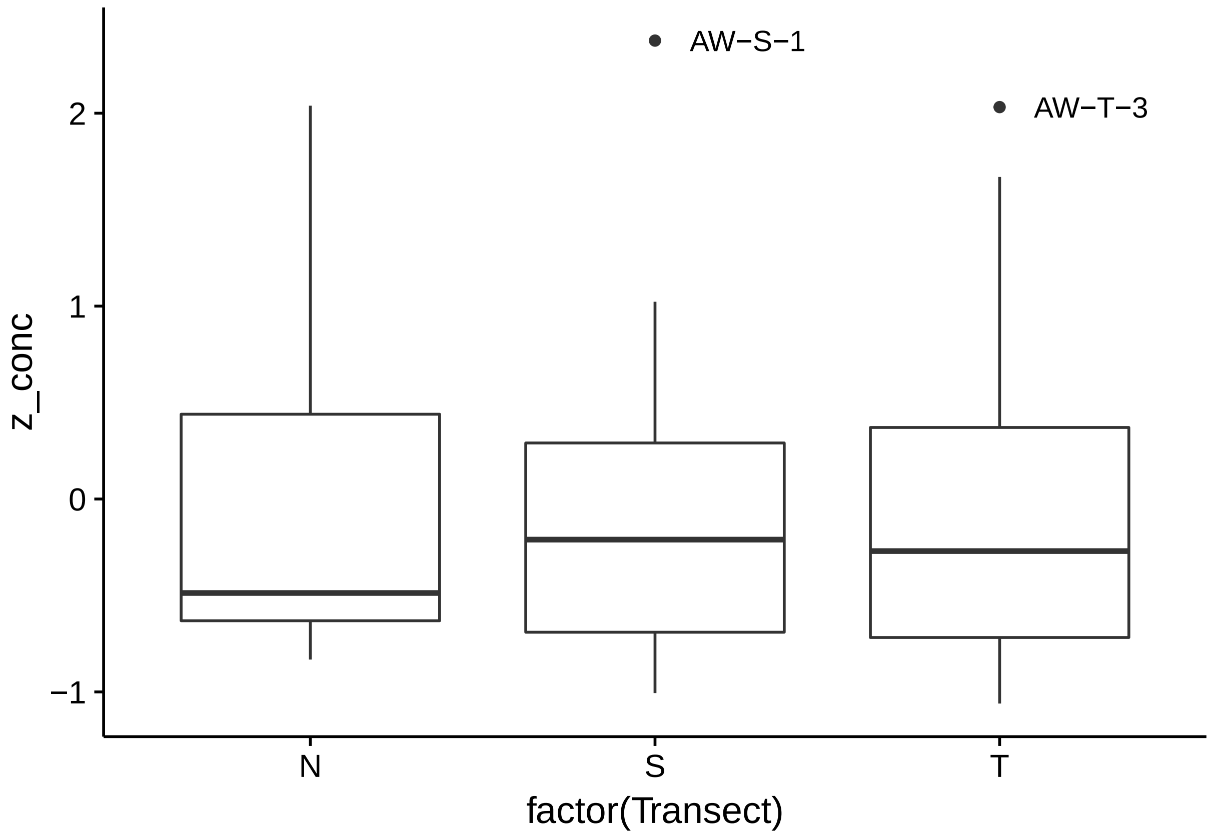
```
# Concentrations
soilGroups %>%
  group_by(Transect) %>%
  mutate(outlier = ifelse(is_outlier(Conc.mug.g.dry.soil), as.character(ID), NA)) %>%
  ggplot(., aes(x = factor(Transect), y = Conc.mug.g.dry.soil)) +
    geom_boxplot() +
    geom_text(aes(label = outlier), na.rm = TRUE, hjust = -0.3)
```



Outliers after transformation

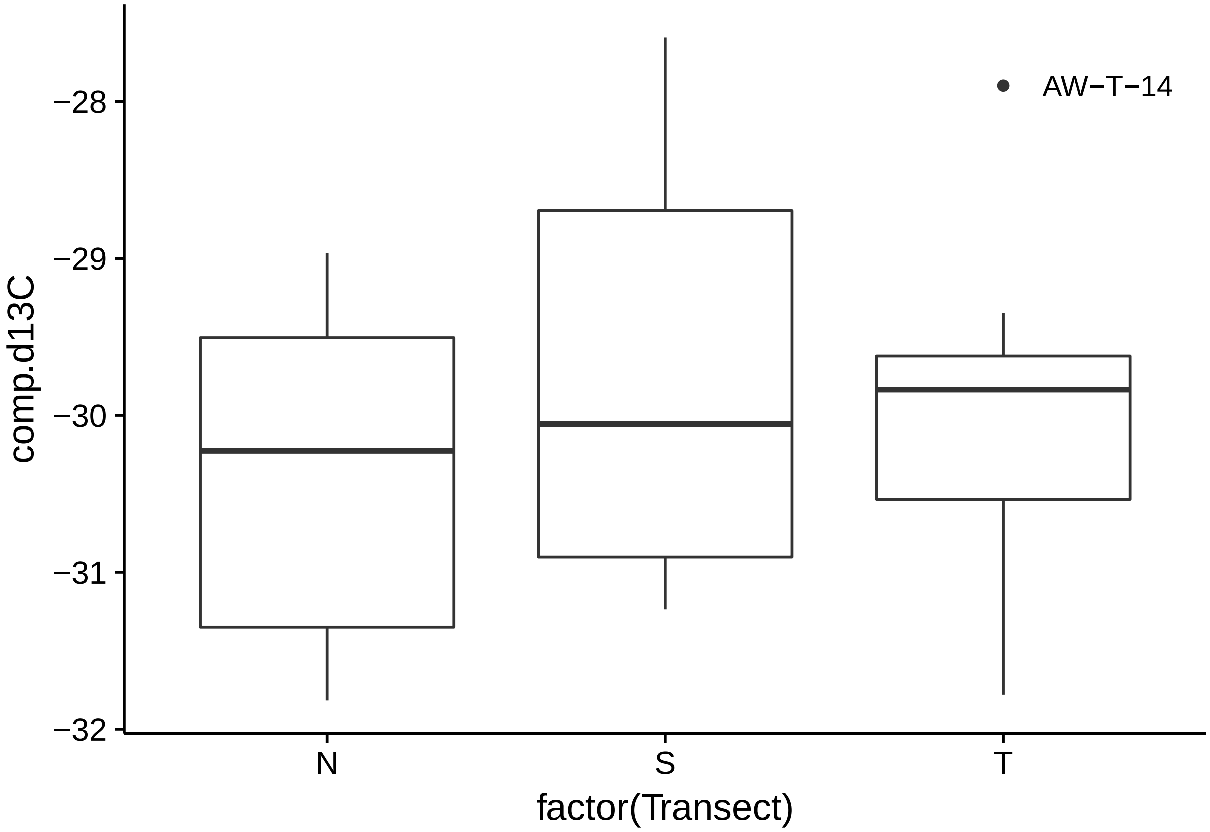
```
soilGroups <- soilGroups %>%
  group_by(Transect) %>%
  mutate(z_conc = (Conc.mug.g.dry.soil-mean(Conc.mug.g.dry.soil))/sd(Conc.mug.g.dry.soil))
```

```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)
```



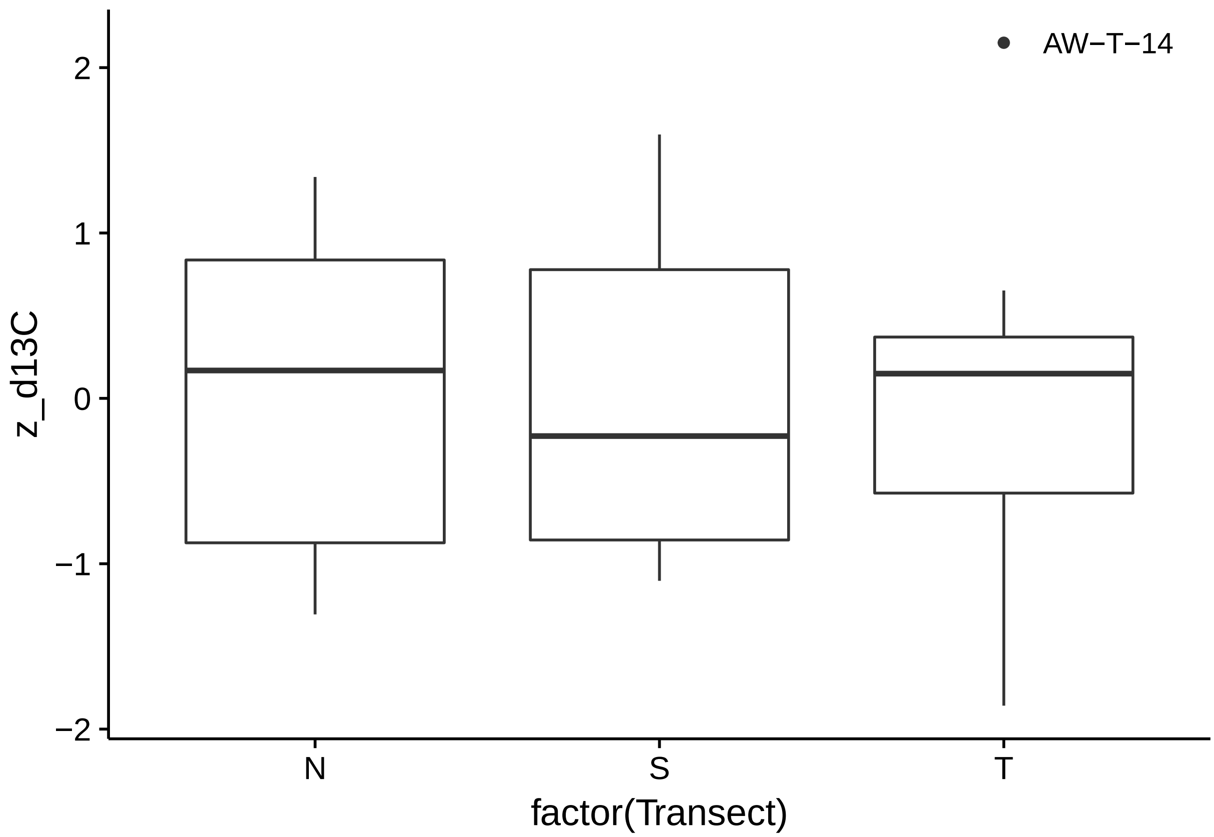
Soil Isotopes

```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)
```



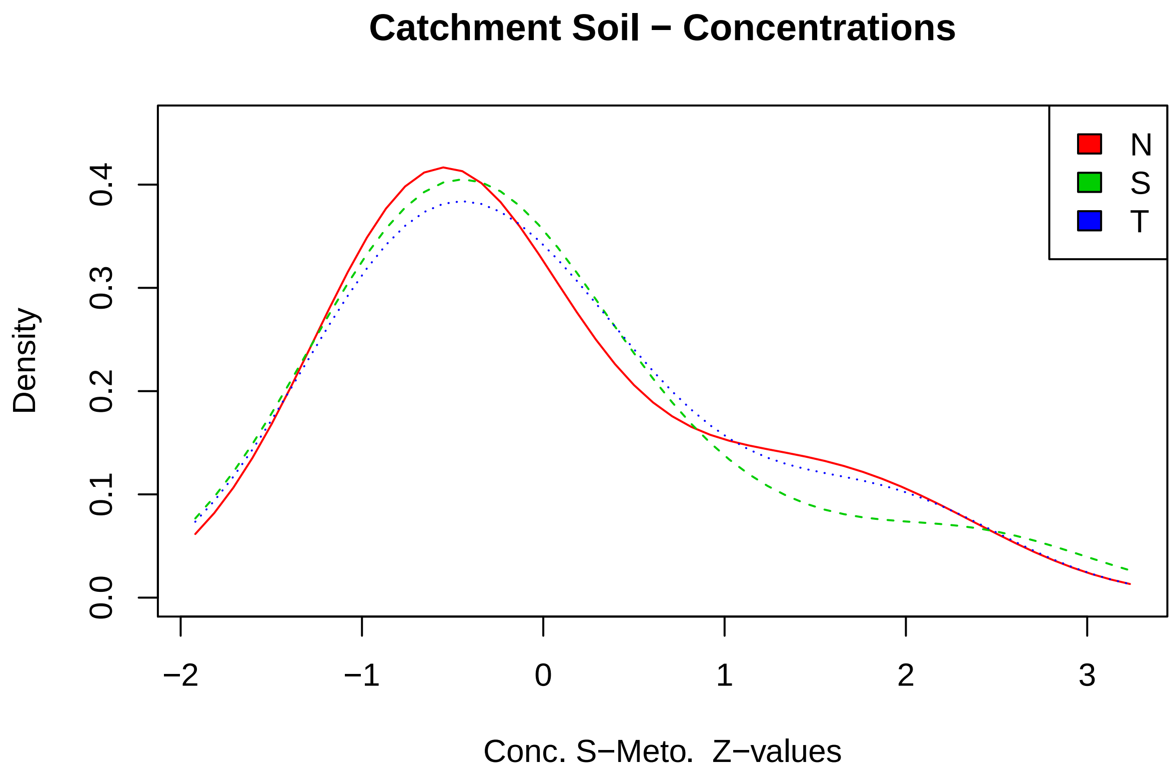
Looks like 7 potential outliers in concentrations and 1 for isotopes. Removing NA's for isotopes and re-computing outliers, reduces the number of outliers to 2 in concentrations and 1 for isotopes.

```
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)))
```

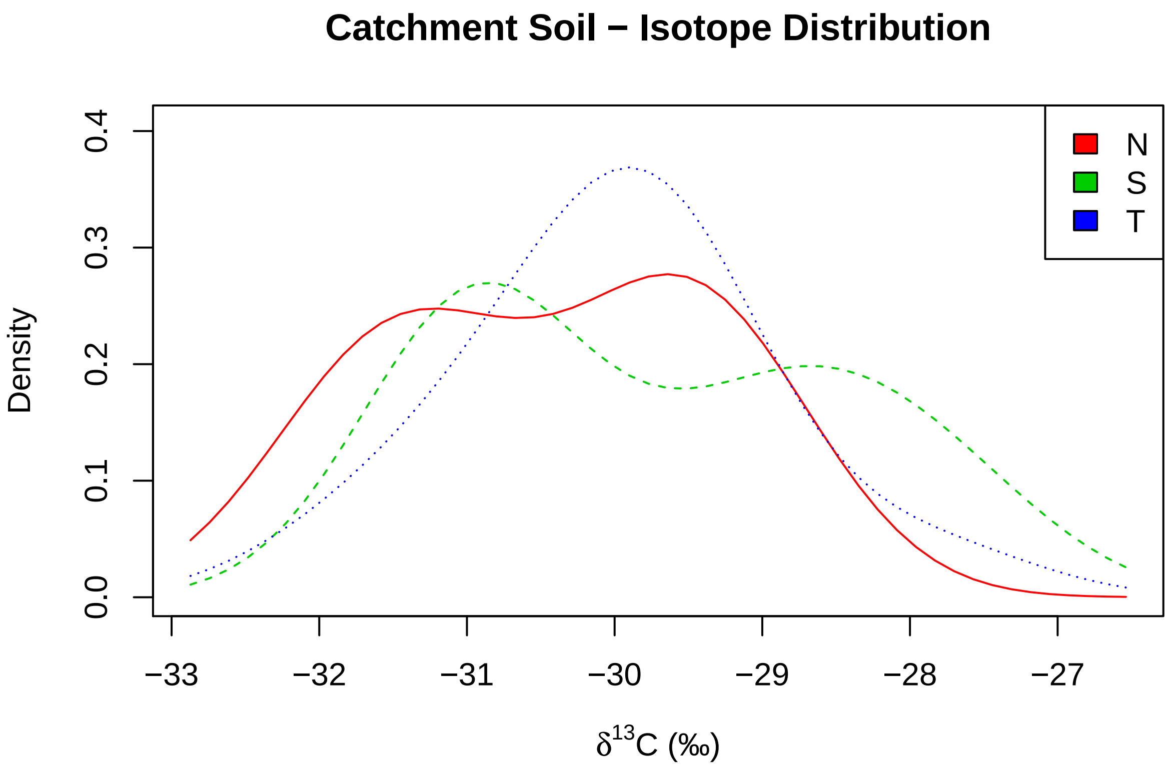



```
#vioplot(soilGroups$Conc.mug.g.dry.soil,  names = "Catchment")
#title(expression(paste("Conc. S-Meto.  ", {({mu}*g / g.soil.dry)})))
```

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)))
```



Catchment Soil − Isotope Distribution
Density
0.0
0.1
0.2
0.3
0.4
−33
−32
−31
−30
−29
−28
−27
δ13C (‰)
N
S
T