PNAS Figures

PAZ

22 novembre 2016

Required R-packages:

```
# Data wrangling
library("plyr")
library("dplyr")

# Melting data sets & changin axes
library("reshape2")
library("ggrepel")

# Plotting:
library("ggplot2")
library("cowplot")
library("gridExtra")
library("Cairo")
library("GGally")
library("scales")
```

Working directory

```
# setwd("D:/Documents/these_pablo/Alteckendorf2016/R")
# setwd("/Users/DayTightChunks/Documents/PhD/Routput/Alteck/R")
# setwd("D:/Documents/these_pablo/Alteckendorf2016/00_TransparencyFolder")
getwd()
```

```
## [1] "D:/Documents/these_pablo/Alteckendorf2016/HydrologicalMonitoring"
```

```
# Show all test graphs (change to TRUE)
SHOW = FALSE
CHECK_ERR = FALSE
```

Lab & field parameters

```
# Pure and cuve isotope average
initialDelta = d13Co = -32.235

# Lab enrichment:
# Alteck
# More degr. under saturation because anaerobic is best according to Fatima
epsilon_max = -1.5 # +/- 0.3 (@ 20C, 20% vwc)
epsilon_min = -2.0 # +/- 0.2 (@ 20C, 40% vwc)

epsilon_mean = -1.75 # Lab
sd(c(epsilon_max, epsilon_min))
```

```
## [1] 0.3535534
```

```
field_epsilon = -1.692547 # Field
field_epsilon_sd = 0.363319 # Â±
```

Soils

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

```
## [1] 0
```

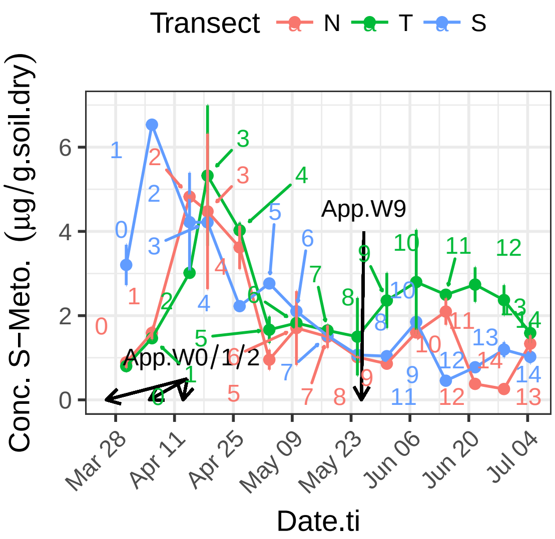
```
#weeklySoil$Conc.ComSoil.SD <-
#  ifelse(weeklySoil$Conc.ComSoil.SD == as.character("#DIV/0!"), NA, as.numeric(as.character(weeklySoil

str(weeklySoil)
```

```
## 'data.frame':    51 obs. of  27 variables:
##  $ ID                 : Factor w/ 51 levels "AW-N-0","AW-N-0x",..: 2 19 36 1 18 35 3 20 37 10 ...
##  $ Transect           : Factor w/ 3 levels "N","S","T": 1 2 3 1 2 3 1 2 3 1 ...
##  $ Wnum               : int  -1 -1 -1 0 0 0 1 1 1 2 ...
##  $ Date.Soil          : Factor w/ 17 levels "03/05/2016 13:10",..: 13 13 13 16 16 16 3 3 3 7 ...
##  $ Date.ti            : POSIXct, format: "2016-03-25 00:04:00" "2016-03-25 00:04:00" ...
##  $ Conc.mug.g.dry.soil: num  0.018 0.029 0.02 0.889 3.204 ...
##  $ Conc.ComSoil.SD    : num  NA NA NA 0.133 0.481 ...
##  $ N_compsoil         : int  NA NA NA NA NA NA 3 3 3 3 ...
##  $ comp.d13C          : num  NA NA NA NA NA ...
##  $ comp.d13C.SD       : num  NA NA NA NA NA ...
##  $ N_ngC              : int  NA NA NA NA NA NA 3 3 3 3 ...
##  $ ngC.mean           : num  NA NA NA NA NA ...
##  $ ngC.SD             : num  NA NA NA NA NA ...
##  $ prctError          : num  NA NA NA NA NA ...
##  $ comp.IMP.d13C      : num  NA NA NA NA NA ...
##  $ DD13C.comp         : num  NA NA NA NA NA ...
##  $ f.max.comp         : num  NA NA NA NA NA ...
##  $ B.max.comp         : num  NA NA NA NA NA ...
##  $ f.min.comp         : num  NA NA NA NA NA ...
##  $ B.min.comp         : num  NA NA NA NA NA ...
##  $ f.mean.comp        : num  NA NA NA NA NA ...
##  $ B.mean.comp        : num  NA NA NA NA NA ...
##  $ MassSoil.g         : num  24.82 38.23 8.66 1226.16 4224.23 ...
##  $ MassSoil.g.SD      : num  NA NA NA 184 634 ...
##  $ Area.N             : num  139266 139266 139266 139266 139266 ...
##  $ Area.T             : num  43713 43713 43713 43713 43713 ...
##  $ Area.S             : num  133175 133175 133175 133175 133175 ...
```

```
# weeklySoil = weeklySoil %>%
#  group_by(Transect) %>%
#  arrange(Transect, Wnum)

weeklySoil$Transect <- factor(weeklySoil$Transect, levels = c("N", "T", "S"))
```

Soil Concentrations

```r
################################
# Concentrations
################################
################################
################################
#weeklySoil$ti[3] <- as.POSIXct("2016-04-14 08:25:00")
#weeklySoil$ti[14] <- as.POSIXct("2016-04-14 08:25:00")
#weeklySoil$ti[24] <- as.POSIXct("2016-04-14 08:25:00")
#lb1a2 <- paste("App.")
lbW012 <- paste("App.W0/1/2")
lbW9 <- paste("App.W9")

limits_conc_soil <- aes(ymin=Conc.mug.g.dry.soil-Conc.ComSoil.SD,  ymax=Conc.mug.g.dry.soil+Conc.ComSoil
#limits_conc_soil <- aes(ymin=mean-0.5, ymax=mean+0.5)

pd <- position_dodge(0.5) # move them .05 to the left and right

co = ggplot(weeklySoil[4:48, ],
           aes(x=Date.ti, y=Conc.mug.g.dry.soil, colour=Transect, group = Transect)) +

  geom_point() +
  geom_line() +

  # Error bars
  geom_errorbar(limits_conc_soil, width=.1, position=pd) +
  # scale_y_continuous(limits=c(0,10),oob = rescale_none) +

  # Themes and axes
  theme_bw() +
  theme(legend.position = "top",
        axis.text.x=element_text(angle = 45, hjust = 1)
        #axis.text.x=element_blank(),
        #axis.title.x=element_blank()
        ) +

  scale_x_datetime(breaks = date_breaks("2 weeks"), labels = date_format("%b %d")) +
  ylab(expression(paste("Conc. S-Meto.  ", {({mu}*g / g.soil.dry)}))) +
  # facet_wrap(~Transect, nrow = 3) +
  # xlab("Date") +
  # theme() +
  scale_x_datetime(breaks = date_breaks("2 weeks"), labels = date_format("%b %d")) +

  # Smooth linear models
  # stat_smooth(method = "lm", formula = y ~ poly(x, 2)) +
  # stat_smooth(method = "lm") +

  # Text
  # W0 Application
  # annotate("text", x = as.POSIXct('2016-03-25 08:04:00'), y = 4, label = lb1a2, parse = T, size = 3.0
  geom_segment(aes(x = as.POSIXct('2016-04-14 08:04:00'), y = 0.5, xend = as.POSIXct('2016-03-26 01:04:
               arrow = arrow(length = unit(0.2, "cm"))) +
  # W1 Application
  geom_segment(aes(x = as.POSIXct('2016-04-14 08:04:00'), y = 0.5,
                   xend = as.POSIXct('2016-04-05 08:04:00'), yend = 0), color = "black",
               arrow = arrow(length = unit(0.2, "cm"))) +
  # W2 Application
  annotate("text", x = as.POSIXct('2016-04-15 08:04:00'), y = 1, label = lbW012, parse = T, size = 3.0)
  geom_segment(aes(x = as.POSIXct('2016-04-14 08:04:00'), y = 0.5,
                   xend = as.POSIXct('2016-04-13 08:04:00'), yend = 0), color = "black",
               arrow = arrow(length = unit(0.2, "cm"))) +
  # W9 Application
  annotate("text", x = as.POSIXct('2016-05-26 08:04:00'), y = 4.5, label = lbW9, parse = T, size = 3.0)
  geom_segment(aes(x = as.POSIXct('2016-05-26 08:04:00'), y = 4,
                   xend = as.POSIXct('2016-05-25 18:04:00'), yend = 0), color = "black",
               arrow = arrow(length = unit(0.2, "cm"))) +

  geom_text_repel(aes(label=as.factor(Wnum)),
                  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)
co
```



```
# Linear model
# ggsave(co, filename = "CompositeConcLM.png", width = 7, height = 5, units = "in", scale = 1)

## ggsave(co, filename = "CompositeConcLM.tiff", height = 10, width = 8.7, units = 'cm')

# No linear model
# ggsave(co, filename = "CompositeConc.png", width = 7, height = 5, units = "in", scale = 1)
```

Soil isotope signatures

```
initialDelta
```

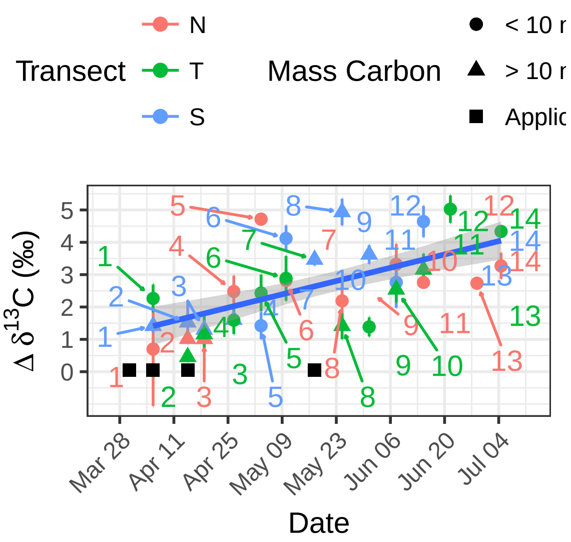
```
## [1] -32.235
```

```
weeklySoil$DD13C.comp <- (weeklySoil$comp.d13C - (initialDelta))
weeklySoil$ngC.Label <- ifelse(weeklySoil$ngC.mean<5, "< 5 ng",
                               ifelse(weeklySoil$ngC.mean<10, "< 10 ng",
                               ifelse((weeklySoil$ngC.mean >= 10 & weeklySoil$ngC.mean < 15), "< 15 ng"
                               ifelse((weeklySoil$ngC.mean >= 15 & weeklySoil$ngC.mean < 20), "< 20 ng"
                               ifelse(weeklySoil$ngC.mean >= 20 & weeklySoil$ngC.mean < 30, "< 30 ng C"

limits_dCsoil <- aes(ymin=comp.d13C-comp.d13C.SD, ymax=comp.d13C+comp.d13C.SD)
#limits_dCsoil <- aes(ymin=comp.d13C-0.5, ymax=comp.d13C+0.5)
lb1a <- paste("App.~S-meto.")
lb1ab <- paste("delta^{13}~C:-32.25")
lb1a2 <- paste("App. ")

lbW012 <- paste("App.W0/1/2")
lbW9 <- paste("App.W9")

if (SHOW) {
ggplot(weeklySoil, aes(x=Date.ti, y=comp.d13C, colour=Transect, group = Transect)) +
  geom_errorbar(limits_dCsoil, width=.05) +
  geom_point() +
  theme_bw() +
  #stat_smooth(method = "lm", formula = y ~ poly(x, 2)) +
  stat_smooth(method = "lm") +
  facet_wrap(~Transect, nrow = 3) +
  xlab("Date") +
  theme(axis.text.x=element_text(angle = 45, hjust = 1)) +
  #ylab(expression(paste({delta}^"13","C", ' \211'))) +
  ylab(expression(paste({delta}^"13","C", ' (\u2030)'))) +
  scale_y_continuous(breaks=seq(-34,-21,2)) +
  geom_hline(yintercept = -31.21, color = "dodgerblue4", linetype = "dotted") +
  geom_hline(yintercept = -30.71, color = "dodgerblue3", linetype = "dotted") +
  geom_hline(yintercept = -31.71, color = "dodgerblue3", linetype = "dotted") +
  annotate("text", x = as.POSIXct('2016-04-05 22:04:00'), y = -22.5, label = lb1a, parse = T, size = 3.
  annotate("text", x = as.POSIXct('2016-04-05 22:04:00'), y = -23.5, label = lb1ab, parse = T, size = 3

  annotate("text", x = as.POSIXct('2016-03-25 08:04:00'), y = -29, label = lb1a2, parse = T, size = 3.0
  geom_segment(aes(x = as.POSIXct('2016-03-25 08:04:00'), y = -29.8,
                   xend = as.POSIXct('2016-03-25 08:04:00'), yend = -31.0),
               arrow = arrow(length = unit(0.2, "cm"))) +
  annotate("text", x = as.POSIXct('2016-04-03 00:04:00'), y = -29, label = lb1a2, parse = T, size = 3.0
  geom_segment(aes(x = as.POSIXct('2016-04-03 00:04:00'), y = -29.8,
                   xend = as.POSIXct('2016-04-05 08:04:00'), yend = -31.0),
               arrow = arrow(length = unit(0.2, "cm"))) +
  annotate("text", x = as.POSIXct('2016-04-13 08:04:00'), y = -25, label = lb1a2, parse = T, size = 3.0
  geom_segment(aes(x = as.POSIXct('2016-04-13 08:04:00'), y = -26,
                   xend = as.POSIXct('2016-04-13 08:04:00'), yend = -31.0),
               arrow = arrow(length = unit(0.2, "cm"))) +
  annotate("text", x = as.POSIXct('2016-05-26 08:04:00'), y = -29, label = lb1a2, parse = T, size = 3.0
  geom_segment(aes(x = as.POSIXct('2016-05-26 08:04:00'), y = -29.8,
                   xend = as.POSIXct('2016-05-25 08:04:00'), yend = -31.0),
               arrow = arrow(length = unit(0.2, "cm"))) +
  #scale_x_continuous(breaks=seq(0,11,1)) +
  scale_x_datetime(breaks = date_breaks("2 weeks"), labels = date_format("%b %d")) +

  #annotate("text", x = as.POSIXct('2016-05-30 20:04:00'), y = -30.5, label = lb1a, parse = T, size = 2
  theme(legend.position = "top")

# Linear model (LM)
# ggsave(isCo, filename = "CompositeIsotopesLM.png", width = 7, height = 5, units = "in", scale = 1)
# No linear model
# ggsave(isCo, filename = "CompositeIsotopes.png", width = 7, height = 5, units = "in", scale = 1)

}

if (SHOW) {
# View(weeklySoil)
# Ommitted, graph is tautological.
### Delta vs. f (Soils)
ggplot(weeklySoil, aes(x=f.comp, y=DD13C.comp, colour=Transect, group = Transect)) +
  #geom_errorbar(limits_dCsoil, width=.05) +
  geom_point() +
  theme_bw() +
  stat_smooth(method = "lm", formula = y ~ poly(x, 2)) +
  #stat_smooth(method = "lm") +
  facet_wrap(~Transect, nrow = 3) +
  scale_x_reverse() +
  xlab("Fraction remaining (f)") +
  theme(axis.text.x=element_text(angle = 45, hjust = 1)) +
  #ylab(expression(paste({delta}^"13","C", ' \211'))) +
  ylab(expression(paste({Delta~delta}^"13","C", ' (\u2030)'))) +
  #scale_y_continuous(breaks=seq(-34,-21,2)) +
  theme(legend.position = "top") +
  #geom_text_repel(aes(label=WeekNo, color = factor(Transect)),
                  #arrow = arrow(length = unit(0.005, 'npc'), type = "closed"),
                  #force = 1,
                  #point.padding = unit(1.0, 'lines'),
                  #max.iter = 2e3,
                  #nudge_x = .2) +
  geom_point()


  ###############################
###############################
### DeltaDelta vs time
###############################
# View(weeklySoil)
# limits_DdCsoil <- aes(ymin=comp.d13C-comp.d13C.SD-initialDelta, ymax=comp.d13C+comp.d13C.SD-initialDe
limits_DdCsoil <- aes(ymin=comp.d13C-comp.d13C.SE-initialDelta, ymax=comp.d13C+comp.d13C.SE-initialDelta

# pd <- position_dodge(0.5)
# AOdf[1:27,]
```

```r
deltaTime = ggplot(na.omit(weeklySoil), aes(Date.ti, DD13C.comp)) +
  geom_errorbar(limits_DdCsoil) +
  geom_point(aes(shape = Transect, colour = ngC.Label)) +
  labs(shape="Transect", colour = "Mass Carbon") +

  # geom_point(weeklySoil[1:48, ], aes(x=Date.ti, y=DD13C.comp, colour=Transect, group = Transect)) +

  # Themes and axes
  theme_bw() +
  theme(# legend.position="none",
        # axis.title.x = element_blank(),
        axis.text.x=element_text(angle = 45, hjust = 1)
        ) +
  xlab("Date") +
  scale_x_datetime(breaks = date_breaks("2 weeks"), labels = date_format("%b %d")) +


  ylab(expression(paste({Delta~delta}^"13","C", ' (\u2030)'))) +
  scale_y_continuous(breaks=seq(0, 8, 1)) +
  # ylab(expression(paste({delta}^"13","C", ' \211'))) +
  # ylab(expression(paste({delta}^"13","C", ' (\u2030)'))) +
  # facet_wrap(~Transect, nrow = 3) +

  # Smooth linear models
  stat_smooth(data = weeklySoil, aes(group = Transect), method = "lm", se=FALSE) +
  # stat_smooth(method = "lm", formula = y ~ poly(x, 2)) +
  # stat_smooth(data=subset(weeklySoil[4:27, ]), method = "lm", formula = y~x, se=F) +
  # stat_smooth(data=subset(weeklySoil[18:36, ]), method = "lm", formula = y~x, se=F) +

  # Text
  # Application W0
  annotate("text",
           x = as.POSIXct('2016-04-04 01:04:00'), y = 6, label = lbW012, parse = T, size = 3.0) +
  geom_segment(aes(x = as.POSIXct('2016-04-03 08:04:00'), y = 5.5,
                   xend = as.POSIXct('2016-03-25 22:04:00'), yend = -0), color = "black",
               arrow = arrow(length = unit(0.2, "cm"))) +
  #annotate("text",
  #         x = as.POSIXct('2016-04-03 00:04:00'), y = 2, label = lb1a2, parse = T, size = 3.0) +

  # Application W1
  geom_segment(aes(x = as.POSIXct('2016-04-03 08:04:00'), y = 5.5,
                   xend = as.POSIXct('2016-04-05 08:04:00'), yend = 0), color = "black",
               arrow = arrow(length = unit(0.2, "cm"))) +
  # annotate("text", x = as.POSIXct('2016-04-15 08:04:00'), y = 1, label = lb1a2, parse = T, size = 3.0

  # Application W2
  geom_segment(aes(x = as.POSIXct('2016-04-03 08:04:00'), y = 5.5,
                   xend = as.POSIXct('2016-04-13 08:04:00'), yend = 0), color = "black",
               arrow = arrow(length = unit(0.2, "cm"))) +

  # Application W9
  annotate("text",
           x = as.POSIXct('2016-06-10 08:04:00'), y = 1.8, label = lbW9, parse = T, size = 3.0) +
  geom_segment(aes(x = as.POSIXct('2016-06-10 08:04:00'), y = 1.5,
                   xend = as.POSIXct('2016-05-25 18:04:00'), yend = 0), color = "black",
               arrow = arrow(length = unit(0.2, "cm"))) +

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

deltaTime

}
```

Delta time final

```
limits_DdCsoil <- aes(ymin=comp.d13C-comp.d13C.SD-initialDelta, ymax=comp.d13C+comp.d13C.SD-initialDelt

weeklySoil$ngC.Label <- ifelse(weeklySoil$ngC.mean < 10, "< 10 ng", "> 10 ng")

                               # ifelse( weeklySoil$ngC.mean >= 10 & weeklySoil$ngC.mean < 15, "< 15 ng
wk <- weeklySoil
# wk <- na.omit(weeklySoil)
wk$Application <- ifelse(wk$Date.ti == as.POSIXct('2016-05-17 09:16:00', tz = "EST") & wk$Transect == "
                    ifelse(wk$Date.ti == as.POSIXct('2016-03-30 12:18:00', tz = "EST") & wk$Transect ==
                     ifelse(wk$Date.ti == as.POSIXct('2016-04-05 15:08:00', tz = "EST") & wk$Transect =
                       ifelse(wk$Date.ti == as.POSIXct('2016-04-14 13:52:00', tz = "EST") & wk$Transect

deltaTime = ggplot(wk, aes(Date.ti, DD13C.comp)) +
  geom_errorbar(limits_DdCsoil, width = 0.5) +
  geom_point(aes(colour = Transect, shape = ngC.Label, shape = "Application"), size = 2) +
  labs(colour="Transect", shape = "Mass Carbon") +
  geom_point(aes(Date.ti, Application, shape = "Application"),
             size =2 , data = wk,  legend.title = element_blank()) +
  # Themes and axes
  theme_bw() +
  theme(legend.position="top",
        # axis.title.x = element_blank(),
        axis.text.x=element_text(angle = 45, hjust = 1)
        ) +
  xlab("Date") +
  scale_x_datetime(breaks = date_breaks("2 weeks"), labels = date_format("%b %d")) +
  guides(col = guide_legend(nrow =  3), shape = guide_legend(nrow =  3)) +
  ylab(expression(paste({Delta~delta}^"13","C", ' (\u2030)'))) +
  scale_y_continuous(breaks=seq(0, 8, 1)) +
  stat_smooth(data = weeklySoil, method = "lm", se=T) +
  geom_text_repel(data = subset(weeklySoil), aes(label=Wnum, colour = Transect),
                 arrow = arrow(length = unit(0.01, 'npc'), type = "closed"),
                 force = 1,
                 point.padding = unit(0.5, 'lines'),
                 max.iter = 2e3,
                  nudge_x = .5, show.legend = F)

deltaTime
```



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

Degradation

```
lb1a2 <- paste("App.")

lb1b <- paste("(A)~epsilon:-1.5")
lb1b2 <- paste("(B)~epsilon:-2.0")

if (SHOW) {
  ggplot(weeklySoil, aes(x=Date.ti, y=B.min.comp, colour=Transect, group = Transect)) +
    ylab("Degradation (%)") +
    scale_y_continuous(breaks=seq(0, 100, 20)) +
    geom_point() +
    # geom_point(aes(x=Date.ti, y=B.comp, colour=Transect, group = Transect)) +
    theme_bw() +
    stat_smooth(method = "lm", formula = y ~ poly(x, 2), se=FALSE) +
    annotate("text", x = as.POSIXct('2016-04-11 20:04:00'),
             y = 100, label = lb1b, parse = T, size = 3.0, color = "grey40") +
    annotate("text", x = as.POSIXct('2016-04-11 20:04:00'),
             y = 85, label = lb1b2, parse = T, size = 3.0, color = "dodgerblue4" ) +

    annotate("text", x = as.POSIXct('2016-03-25 08:04:00'), y = 30, label = lb1a2, parse = T, size = 3.
    geom_segment(aes(x = as.POSIXct('2016-03-25 08:04:00'), y = 25,
                     xend = as.POSIXct('2016-03-25 08:04:00'), yend = 20),
                 arrow = arrow(length = unit(0.2, "cm"))) +
    annotate("text", x = as.POSIXct('2016-04-03 00:04:00'), y = 30, label = lb1a2, parse = T, size = 3.
    geom_segment(aes(x = as.POSIXct('2016-04-03 00:04:00'), y = 25,
                     xend = as.POSIXct('2016-04-05 08:04:00'), yend = 20),
                 arrow = arrow(length = unit(0.2, "cm"))) +
    annotate("text", x = as.POSIXct('2016-04-13 08:04:00'), y = 30, label = lb1a2, parse = T, size = 3.
    geom_segment(aes(x = as.POSIXct('2016-04-13 08:04:00'), y = 25,
                     xend = as.POSIXct('2016-04-13 08:04:00'), yend = 20),
                 arrow = arrow(length = unit(0.2, "cm"))) +
    annotate("text", x = as.POSIXct('2016-05-26 08:04:00'), y = 30, label = lb1a2, parse = T, size = 3.
    geom_segment(aes(x = as.POSIXct('2016-05-26 08:04:00'), y = 25,
                     xend = as.POSIXct('2016-05-25 08:04:00'), yend = 20),
                 arrow = arrow(length = unit(0.2, "cm"))) +
    scale_x_datetime(breaks = date_breaks("2 weeks"), labels = date_format("%b %d")) +
    #scale_x_continuous(breaks=seq(0,11,1)) +
    theme(legend.position = "top")
}
```

Water

```
AOdf = read.csv2("Data/WeeklyHydroContam_R.csv")
str(AOdf)
```

```
## 'data.frame':    51 obs. of  111 variables:
##  $ ti                    : Factor w/ 51 levels "2016-03-25 00:04:00",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ 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 ...
##  $ 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 ...
##  $ 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 3 NA 3 ...
##  $ diss.d13C             : num  NA NA NA -31.5 -31.7 ...
##  $ SD.d13C               : num  NA NA NA 0.106 0.151 ...
##  $ se.d13C               : num  NA NA NA 0.0612 0.0874 ...
##  $ 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 ...
##  $ 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 ...
##  $ filt.se.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 ...
##  $ f.diss                : num  NA NA NA 0.707 0.759 ...
##  $ f.diss.Field          : num  NA NA NA 0.639 0.7 ...
##  $ f.filt                : num  NA NA NA NA NA ...
##  $ f.diss.min            : num  NA NA NA 0.655 0.714 ...
##  $ f.diss.min.Field      : num  NA NA NA 0.573 0.642 ...
##  $ f.filt.min            : num  NA NA NA NA NA ...
##  $ f.diss.max            : num  NA NA NA 0.746 0.792 ...
##  $ f.diss.max.Field      : num  NA NA NA 0.687 0.742 ...
##  $ f.filt.max            : num  NA NA NA NA NA ...
##  $ B.diss                : num  NA NA NA 29.3 24.1 ...
##  $ B.diss.Field          : num  NA NA NA 36.1 30 ...
##  $ B.filt                : num  NA NA NA NA NA ...
##  $ B.diss.max            : num  NA NA NA 34.5 28.6 ...
##  $ B.diss.max.Field      : num  NA NA NA 42.7 35.8 ...
##  $ B.filt.max            : num  NA NA NA NA NA ...
##  $ B.diss.min            : num  NA NA NA 25.4 20.8 ...
##  $ B.diss.min.Field      : num  NA NA NA 31.3 25.8 ...
##  $ B.filt.min            : 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  17319 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  17319 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  17319 17319 17319 17319 17319 ...
##  $ 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 ...
##   [list output truncated]
```

```
AOdf$ti <- as.POSIXct(strptime(AOdf$ti, "%Y-%m-%d %H:%M", tz="EST"))
sum(is.na(AOdf$ti)) == 0
```

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

Outlet - Concentrations

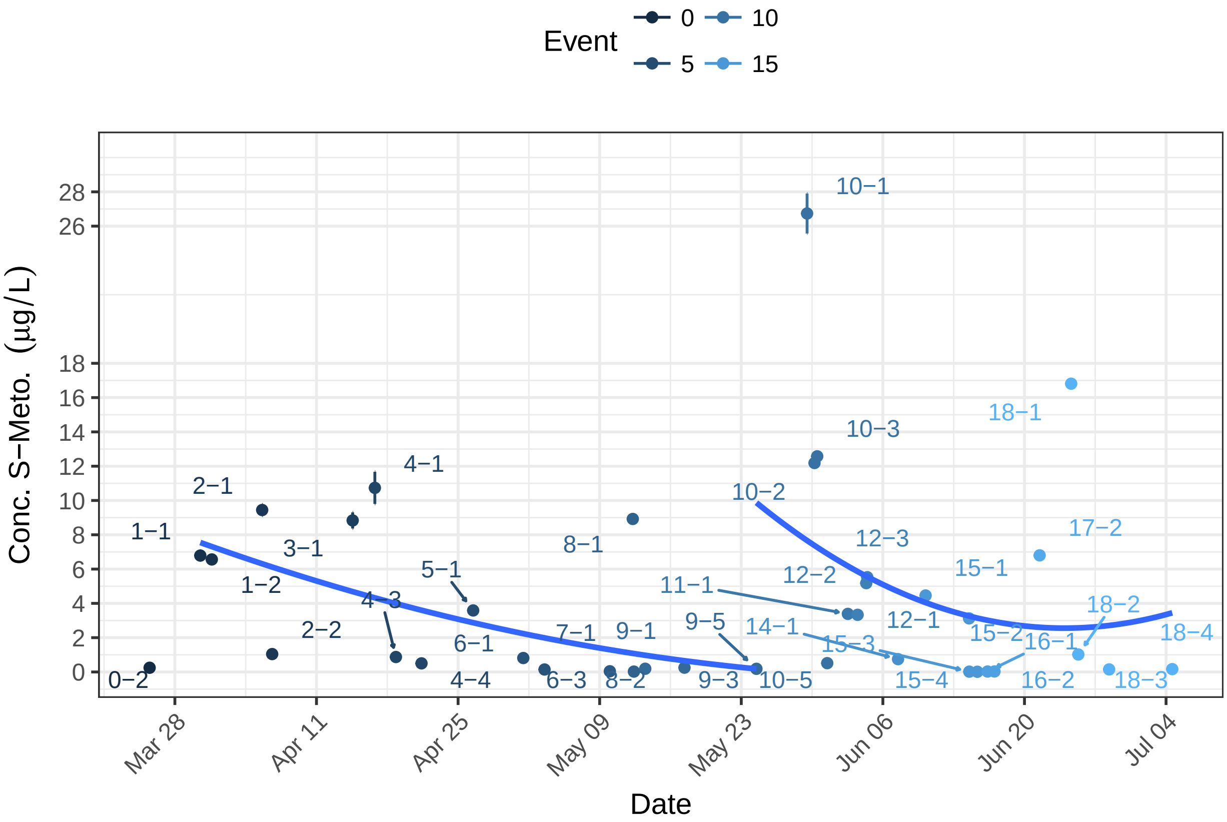
```
#, fig.height=6, fig.width=6}
# Volumes sampled vs. not sampled
vols <- AOdf %>%
  group_by(Sampled) %>%
  summarise_each(funs(sum(., na.rm=TRUE)), Volume.m3)

prctSampled <- vols[2, "Volume.m3"]/(vols[1, "Volume.m3"] + vols[2, "Volume.m3"])*100
prctSampled
```

```
##   Volume.m3
## 1  64.39556
```

```
prctNotSampled <- vols[1, "Volume.m3"]/(vols[1, "Volume.m3"] + vols[2, "Volume.m3"])*100
prctNotSampled
```

```
##   Volume.m3
## 1  35.60444
```

```
# Subset the data
# newdata <- mydata[ which(mydata$gender=='F' & mydata$age > 65), ]

limits_conc <- aes(ymin=Conc.mug.L-Conc.SD, ymax=Conc.mug.L+Conc.SD, color = Event, group = Event)
dfSampled <- AOdf[which(AOdf$Sampled == 'Sampled'), ]

dfSampled$Row <- seq.int(nrow(dfSampled))

conc1 <- ggplot(dfSampled, aes(x=ti, y=Conc.mug.L)) +
  geom_point( aes(color = Event, group = Event)) +
  # geom_point( aes(color = Event)) +

  # Error bars
  geom_errorbar(limits_conc, width=1) +
  # Themes and axes
  theme_bw() +
  theme(axis.text.x=element_text(angle = 45, hjust = 1),
        #axis.text.x=element_blank(),
        #axis.title.x=element_blank(),
        legend.position="top"
        )+
  guides(col = guide_legend(nrow =  2)) +  # Sets legend parameters
  xlab("Date") +
  scale_x_datetime(breaks = date_breaks("2 weeks"), labels = date_format("%b %d")) +
  ylab(expression(paste("Conc. S-Meto.  ", {({mu}*g / L)}))) +
  scale_y_continuous( breaks = c(28, 26, 18,16,14,12,10, 8, 6, 4, 2, 0), limits = c(0, 30) ) +

  # Smooth linear models
  geom_smooth(data=subset(AOdf[4:28, ]), method = "lm", formula = y ~ poly(x, 2), se = F) +
  geom_smooth(data=subset(AOdf[28:length(AOdf), ]), method = "lm", formula = y ~ poly(x, 2), se= F) +
  #stat_smooth(data=subset(dfSampled[19:30, ]), method = "lm", formula = y ~ x) +
  # Text
  # Application W9
  # annotate("text",
  #         x = as.POSIXct('2016-06-10 08:04:00'), y = -1, label = lbW9, parse = T, size = 3.0) +
  # geom_segment(aes(x = as.POSIXct('2016-06-05 08:04:00'), y = -1,
  #                 xend = as.POSIXct('2016-05-25 18:04:00'), yend = -0.9), color = "black",
  #             arrow = arrow(length = unit(0.2, "cm"))) +
  geom_text_repel(aes(label=Events, color = Event), # WeekSubWeek or Weeks
                  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, show.legend = F)
conc1
```



```
cor.p.SM <- cor.test(dfSampled$Conc.mug.L, dfSampled$diss.d13C, method = "pearson")
cor.p.SM
```

```
##
##  Pearson's product-moment correlation
##
## data:  dfSampled$Conc.mug.L and dfSampled$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.p.ESA <- cor.test(dfSampled$ESA_mean, dfSampled$diss.d13C)
cor.p.ESA
```

```
##
##  Pearson's product-moment correlation
##
## data:  dfSampled$ESA_mean and dfSampled$diss.d13C
## t = -1.0376, df = 25, p-value = 0.3094
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.5414053  0.1916124
## sample estimates:
```

```
##        cor
## -0.2031974
```

```
cor.p.SM.soils <- cor.test(wk$Conc.mug.g.dry.soil, wk$comp.d13C)
cor.p.SM.soils
```

```
##
##  Pearson's product-moment correlation
##
## data:  wk$Conc.mug.g.dry.soil and wk$comp.d13C
## t = -4.9215, df = 40, p-value = 1.52e-05
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.7736554 -0.3813845
## sample estimates:
##        cor
## -0.6141293
```

Outlet - Masses vs time

```
library("tidyr")

dfSampled1 <- dfSampled[ , c("ti", "Peak", "Event", "Events",
                   "TotSMout.g", "TotSMout.g.SD",
                   "DissOXA.g", "DissOXA.g.SD",
                   "DissESA.g", "DissESA.g.SD",
                   "MELsm.g", "MELsm.g.SD")]
```

Outlet Isotope Shifts (DD)

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

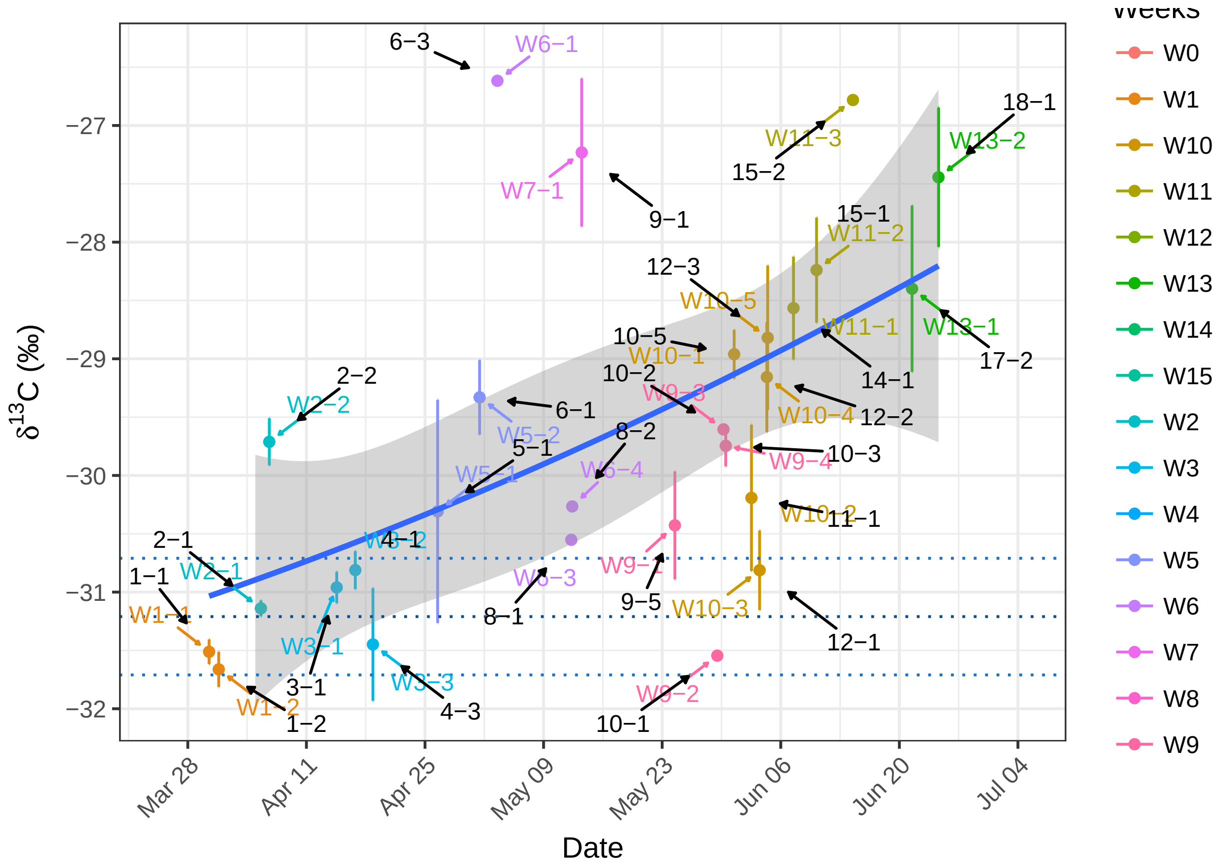
ggplot(mpg, aes(displ, hwy)) + geom_point() + scale_y_continuous(“mpg (US)”, sec.axis = sec_axis(~ . * 1.20, name = “mpg (UK)”))

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

```
if (SHOW) {
  limits_conc <- aes(ymin=Conc.mug.L-Conc.SD, ymax=Conc.mug.L+Conc.SD, color = Event, group = Event)
  ggplot(AOdf[28:length(AOdf),], aes(x=ti, y=Conc.mug.L)) +
    geom_point( aes(color = Weeks, group = Weeks)) +
    # Error bars
    # geom_errorbar(aes(ymin=mean.d13C-SD.d13C, ymax=mean.d13C+SD.d13C), width=.1) +
    # geom_errorbar(limits_conc, width=1) +

    # Themes & axes
    # theme_gray() +
    theme_bw() +
    theme(legend.position = "none") +
    theme(axis.text.x=element_text(angle = 45, hjust = 1),
          axis.text.y = element_blank(),
          legend.title = element_blank(),
          plot.margin = unit(c(0,3.5,0,0), "lines")) +
    #scale_x_datetime(breaks = date_breaks("week"), labels = date_format("%m/%d")) +
    scale_y_continuous(breaks = c(20,15,10,5,0), limits = c(-5, 20) ) +
    xlab("Date") +
    ylab("") +

    # Smooth linear models
      stat_smooth(method = "lm", formula = y ~ poly(x, 2)) +
    #geom_hline(yintercept = -31.21, color = "dodgerblue4", linetype = "dotted") +
    #geom_hline(yintercept = -30.71, color = "dodgerblue3", linetype = "dotted") +
    #geom_hline(yintercept = -31.71, color = "dodgerblue3", linetype = "dotted") +

    # Text
    #annotate("text", x = as.POSIXct('2016-06-25 00:04:00'), y = -31.2, label = lb1, parse = T) +
    annotate("text", x = as.POSIXct('2016-05-27 08:04:00'), y = -3, label = "App.4", parse = T) +
    geom_segment(aes(x = as.POSIXct('2016-05-26 08:04:00'), y = -4,
                     xend = as.POSIXct('2016-05-26 08:04:00'), yend = -5.0),
                 arrow = arrow(length = unit(0.2, "cm"))) +
    geom_text_repel(aes(label=Weeks, color = factor(Weeks)),
                    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)

}
```

Outlet Isotopes - Continous (by Week)

```
AOdf$SD.d13C.err  <- ifelse(is.na(AOdf$SD.d13C), 0.5, AOdf$SD.d13C)
# limits_dC <- aes(ymin=diss.d13C-SD.d13C.err, ymax=diss.d13C+SD.d13C.err, color = Weeks, group = Weeks
limits_dC <- aes(ymin=diss.d13C-SD.d13C, ymax=diss.d13C+SD.d13C, color = Weeks, group = Weeks)
# View(AOdf)

iso <- ggplot(AOdf, aes(x=ti, y=diss.d13C)) +
  #geom_errorbar(aes(ymin=mean.d13C-SD.d13C, ymax=mean.d13C+SD.d13C), width=.1) +
  geom_errorbar(limits_dC, width=1) +
  #theme_gray() +
  theme_bw() +
  theme(axis.text.x=element_text(angle = 45, hjust = 1)) +
  #scale_x_datetime(breaks = date_breaks("week"), labels = date_format("%m/%d")) +
  scale_x_datetime(breaks = date_breaks("2 weeks"), labels = date_format("%b %d")) +
  geom_point( aes(color = Weeks, group = Weeks)) +
  #stat_smooth(method = "lm", formula = y ~ x) +
  geom_smooth(data=subset(AOdf[4:length(AOdf), ]), method = "lm", formula = y ~ poly(x, 2)) +
  #stat_smooth(method = "lm", formula = y ~ poly(x, 2)) +
  #theme(axis.text.x = element_blank()) +
  #theme(plot.margin = unit(c(1,1,1,1), "lines")) +
  geom_hline(yintercept = -31.21, color = "dodgerblue4", linetype = "dotted") +
  geom_hline(yintercept = -30.71, color = "dodgerblue3", linetype = "dotted") +
  geom_hline(yintercept = -31.71, color = "dodgerblue3", linetype = "dotted") +
  #annotate("text", x = as.POSIXct('2016-06-25 00:04:00'), y = -31.2, label = lb1, parse = T) +
  xlab("Date") +
  #theme(legend.position="top") +
  scale_y_continuous(breaks = c(-32,-31,-30,-29, -28, -27), limits = c(-32, -26.4) ) +
  ylab(expression(paste({delta}^"13","C", ' (\u2030)'))) +
  geom_text_repel(aes(label=WeekSubWeek, color = factor(Weeks)),
                  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, show.legend = F) +
  geom_text_repel(aes(label=Events),
                  size = 3,
                  arrow = arrow(length = unit(0.01, 'npc'), type = "closed"),
                  force = 0.5,
                  point.padding = unit(1.5, 'lines'),
                  max.iter = 2e3,
                  nudge_x = .06, show.legend = F)
  #ylab(expression(paste({delta}^"13","C", ' \211')))
  #ylab(expression(paste({delta}^"13","C")))

iso
```



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

All plots

```r
if (CHECK_ERR) {
concSoils <- co + theme(legend.position='none')
concWater <- conc1 + theme(legend.position='none')
isoSoils <- deltaTime + theme(legend.position='none')

legend_soils = get_legend(deltaTime)
legend_water = get_legend(conc1)

grid4 <- plot_grid(concSoils, isoSoils, concWater, iso2,
                    ncol =2, nrow = 2, align ="v",
                    labels = c("A", "C", "B", "D"))
fig1 <- ggdraw() +
  draw_plot(grid4, x=0, y=.2, width =  1, height = .8) +
  draw_plot(legend_water, x=0.5, y = 0.05, width = 0.5, height = 0.2) +
  draw_plot(legend_soils, x=0, y=0.05, width = 0.6, height = 0.2)

fig1
}

#ggsave(fig1, filename = "SoilsAndOutlet.tiff", height = 18, width = 17.8, units = 'cm')
```

Mass balance approach

```r
library("ggplot2")
library("scales")
library("reshape2")
library("zoo")

soilsOut = read.csv2("Data/MassBalance_R.csv", header = T)
soilsOut$ti <- as.POSIXct(soilsOut$ti, "%Y-%m-%d %H:%M", tz = "EST")
sum(is.na(soilsOut$ti))
```

```
## [1] 0
```

```r
# Remove bulk catchment values that came from
# at least one source of inputed data (all inputed data has no stdrd. devs)
soilsOut$BulkCatch.d13 <- ifelse(is.na(soilsOut$comp.d13C.SD.North), NA,
                                 ifelse(is.na(soilsOut$comp.d13C.SD.Talweg), NA,
                                              ifelse(is.na(soilsOut$comp.d13C.SD.South), NA, soilsOut$Bu
                                        )
                                 )



print("Mass Balance Soils")
```

```
## [1] "Mass Balance Soils"
```

```r
str(soilsOut)
```

```
## 'data.frame':    52 obs. of  57 variables:
##  $ 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 ...
##  $ Event               : int  0 0 0 1 1 1 2 2 2 3 ...
##  $ timeSinceApp        : num  0.5 3.9 5.5 6.6 7.6 11.6 12.6 14 20.6 2.2 ...
##  $ timeSinceApp.NoSo   : num  0.5 3.9 5.5 6.6 7.6 11.6 12.6 14 20.6 2.2 ...
##  $ diss.d13C           : num  NA NA NA -31.5 -31.7 ...
##  $ SD.d13C             : num  NA NA NA 0.106 0.151 ...
##  $ B.diss              : num  NA NA NA 29.3 24.1 ...
##  $ B.filt              : num  NA NA NA NA NA ...
##  $ 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 ...
##  $ CumAppMass.g        : num  17319 17319 17319 17319 17319 ...
##  $ CumOutMELsm.g       : num  0.302 2.38 4.76 35.001 62.009 ...
##  $ 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  24.8 NA NA 1226.2 NA ...
##  $ MassSoil.g.SD.North : num  NA NA NA 184 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 ...
##  $ Area.N              : num  139266 NA NA 139266 NA ...
##  $ Area.T              : num  43713 NA NA 43713 NA ...
##  $ Area.S              : num  133175 NA NA 133175 NA ...
##  $ B.mean.comp.Talweg  : num  NA NA NA NA NA ...
##  $ B.max.comp.Talweg   : num  NA NA NA NA NA ...
##  $ B.min.comp.Talweg   : num  NA NA NA NA NA ...
##  $ MassSoil.g.Talweg   : num  8.66 NA NA 346.54 NA ...
##  $ MassSoil.g.SD.Talweg: num  NA NA NA 52 NA ...
##  $ comp.d13C.Talweg    : num  NA NA NA NA NA ...
##  $ comp.d13C.SD.Talweg : num  NA NA NA NA NA ...
##  $ ID.T                : Factor w/ 17 levels "AW-T-0","AW-T-0x",..: 2 NA NA 1 NA NA 3 NA NA 10 ...
##  $ B.mean.comp.South   : num  NA NA NA NA NA ...
##  $ B.max.comp.South    : num  NA NA NA NA NA ...
##  $ B.min.comp.South    : num  NA NA NA NA NA ...
##  $ MassSoil.g.South    : num  38.2 NA NA 4224.2 NA ...
##  $ MassSoil.g.SD.South : num  NA NA NA 634 NA ...
##  $ comp.d13C.South     : num  NA NA NA NA NA ...
##  $ comp.d13C.SD.South  : num  NA NA NA NA NA ...
##  $ ID.S                : Factor w/ 17 levels "AW-S-0","AW-S-0x",..: 2 NA NA 1 NA NA 3 NA NA 10 ...
##  $ CatchMassSoil.g     : num  71.7 NA NA 5796.9 NA ...
##  $ CatchMassSoil.g.SD  : num  NA NA NA 382 NA ...
##  $ BulkCatch.d13       : num  NA NA NA NA NA ...
##  $ BulkCatch.d13.SD    : num  NA NA NA NA NA ...
##  $ BulkCatch.DD13      : num  NA NA NA NA NA ...
##  $ f.mean.bulk         : num  NA NA NA NA NA ...
##  $ f.mean.bulk.Field   : num  NA NA NA NA NA ...
##  $ f.min.bulk          : num  NA NA NA NA NA ...
##  $ f.min.bulk.Field    : num  NA NA NA NA NA ...
##  $ f.max.bulk          : num  NA NA NA NA NA ...
##  $ f.max.bulk.Field    : num  NA NA NA NA NA ...
##  $ B.mean.bulk         : num  NA NA NA NA NA ...
##  $ B.min.bulk          : num  NA NA NA NA NA ...
##  $ B.max.bulk          : num  NA NA NA NA NA ...
##  $ B.mean.bulk.Field   : num  NA NA NA NA NA ...
##  $ B.min.bulk.Field    : num  NA NA NA NA NA ...
##  $ B.max.bulk.Field    : num  NA NA NA NA NA ...
```

```
# Melt data set
##Subset the necessary columns
soilsRemainMass <- soilsOut[, c("ti" , "WeekSubWeek",
                                "diss.d13C", "SD.d13C",
                                "CumAppMass.g", "CatchMassSoil.g",
                                "CumOutDiss.g", "CumOutFilt.g", "CumOutMELsm.g",
                                "f.mean.bulk", "B.mean.bulk"
                                )]

# Fraction remaining with mean lab epsilon (Alteck)
soilsRemainMass$f.mean.outlet <-
  ((10^(-3)*soilsRemainMass$diss.d13C + 1)/(10^(-3)*d13Co + 1))^(1000/(epsilon_mean))

soilsRemainMass$B.mean.outlet <-
  (1 - soilsRemainMass$f.mean.outlet)*100
```

As bar chart, showing only summary when soils are measured

```
# Delete rows with only cumulative obs.
# n specifies max no. of NA's in a row allowed
delete.na <- function(DF, n=0) {
  DF[rowSums(is.na(DF)) <= n , ]
}

remainOnSampleDay <- delete.na(soilsRemainMass, n = 5)

# Omit rows whre no soil samples were made
# This implies missing data values for discharge
remainOnSampleDay <- subset(remainOnSampleDay, !is.na(CatchMassSoil.g)  ) #| !is.na(B.mean.outlet)


remainOnSampleDay$Persist.Prct <- (remainOnSampleDay$CatchMassSoil.g/remainOnSampleDay$CumAppMass.g)*100
remainOnSampleDay$TPs.PrctOut <- ((remainOnSampleDay$CumOutMELsm.g-
                                  (remainOnSampleDay$CumOutDiss.g+remainOnSampleDay$CumOutFilt.g))
                                  /remainOnSampleDay$CumAppMass.g)*100
remainOnSampleDay$SM.PrctOut <- ((remainOnSampleDay$CumOutDiss.g+remainOnSampleDay$CumOutFilt.g)
                                 /remainOnSampleDay$CumAppMass.g)*100

remainOnSampleDay$Unknown <- 100 -
  (remainOnSampleDay$Persist.Prct +
     remainOnSampleDay$TPs.PrctOut +
     remainOnSampleDay$SM.PrctOut)

remainOnSampleDay$F.Bulk <- remainOnSampleDay$f.mean.bulk*100
remainOnSampleDay$B.Bulk <- remainOnSampleDay$B.mean.bulk

remainOnSampleDay$F.Outlet <- remainOnSampleDay$f.mean.outlet*100
remainOnSampleDay$B.Outlet <- remainOnSampleDay$B.mean.outlet
```

```
keepPrct <- c("ti",
            "Persist.Prct", "TPs.PrctOut", "SM.PrctOut", "Unknown",
            "F.Bulk" #, "F.Outlet" # , "B.Bulk",  "B.Outlet"
            )

keepPrctCSIA <- c("ti",
            "Persist.Prct",
            "SM.PrctOut",
            "B.Bulk", "F.Bulk" #, "F.Outlet" #,  "B.Outlet"
            )

prctMB <- remainOnSampleDay[ , (names(remainOnSampleDay) %in% keepPrct)]

# Computing CSIA-based Mass Balance
# Not that %s have been calculated from survey applied mass.
prctCSIA <- remainOnSampleDay[ , (names(remainOnSampleDay) %in% keepPrctCSIA)]
prctCSIA$Tot <- prctCSIA$Persist.Prct + prctCSIA$B.Bulk + prctCSIA$SM.PrctOut
prctCSIA$Error <- ifelse((prctCSIA$Tot - 100) < 0 , 0, prctCSIA$Tot - 100)
prctCSIA$Persist.Prct <- ifelse(prctCSIA$Error > 0,
                                prctCSIA$Persist.Prct - prctCSIA$Error ,
                                prctCSIA$Persist.Prct)

# Chosen to substract error from the persistent fraction method
# as this method required larger assumptions:
# e.g. Extrapolation to catchment areas & GC-MS variability

prctCSIA$Unknown <- ifelse((100 - prctCSIA$Tot) < 0 , 0, 100 - prctCSIA$Tot)
prctCSIA$Tot <- NULL

if (prctMB$ti[3] == as.POSIXct("2016-03-28 22:38:00", tz = "EST")) {
  prctMB <- prctMB[4:nrow(prctMB), ]
}

prctMB <- melt(prctMB, id=c("ti"))
prctCSIA <- melt(prctCSIA, id=c("ti"))
levels(prctMB$variable)
```

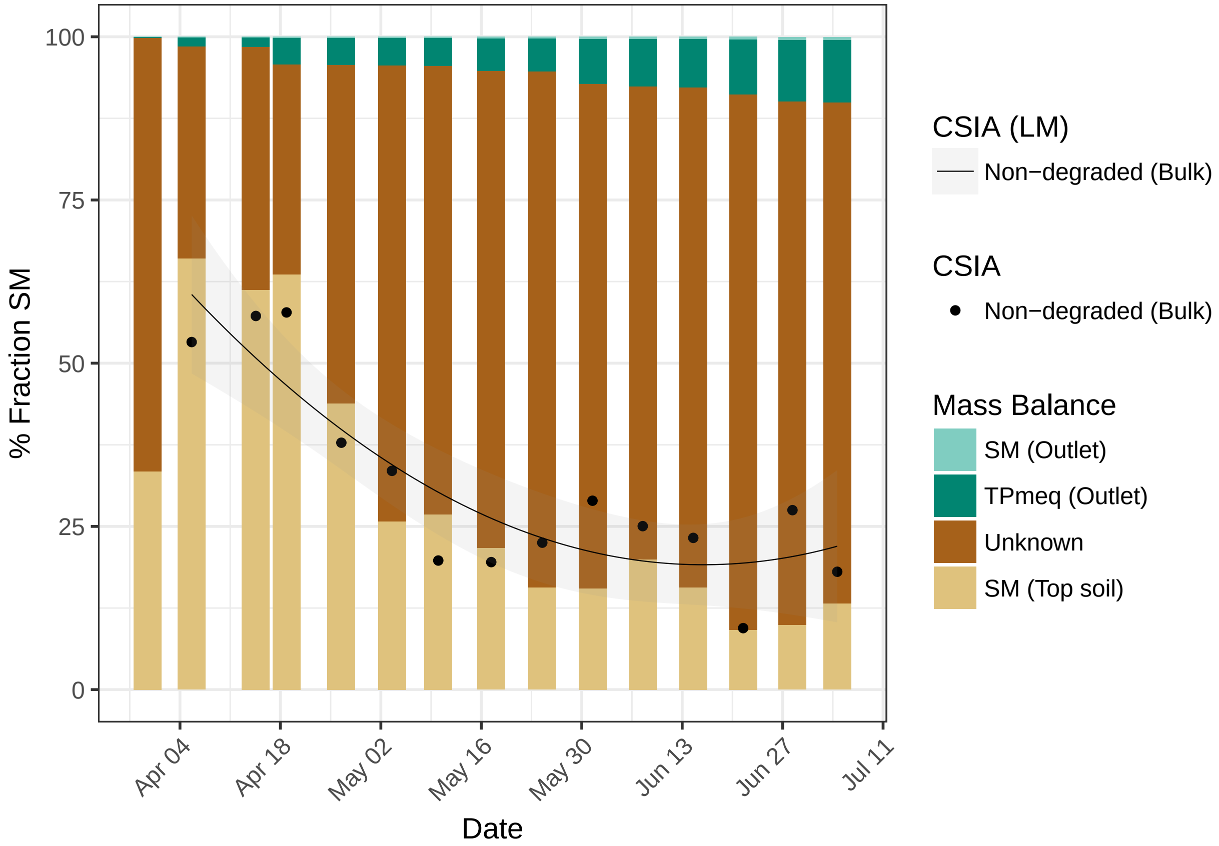
```
## [1] "Persist.Prct" "TPs.PrctOut"  "SM.PrctOut"   "Unknown"
## [5] "F.Bulk"
```

```
prctMB$variable <- factor(prctMB$variable, levels = c( "SM.PrctOut", "TPs.PrctOut", "Unknown", "Persist
                                                      "F.Bulk" #, "F.Outlet" ,
                                                      # "B.Bulk", "B.Outlet"
                                                      ))
levels(prctCSIA$variable)
```

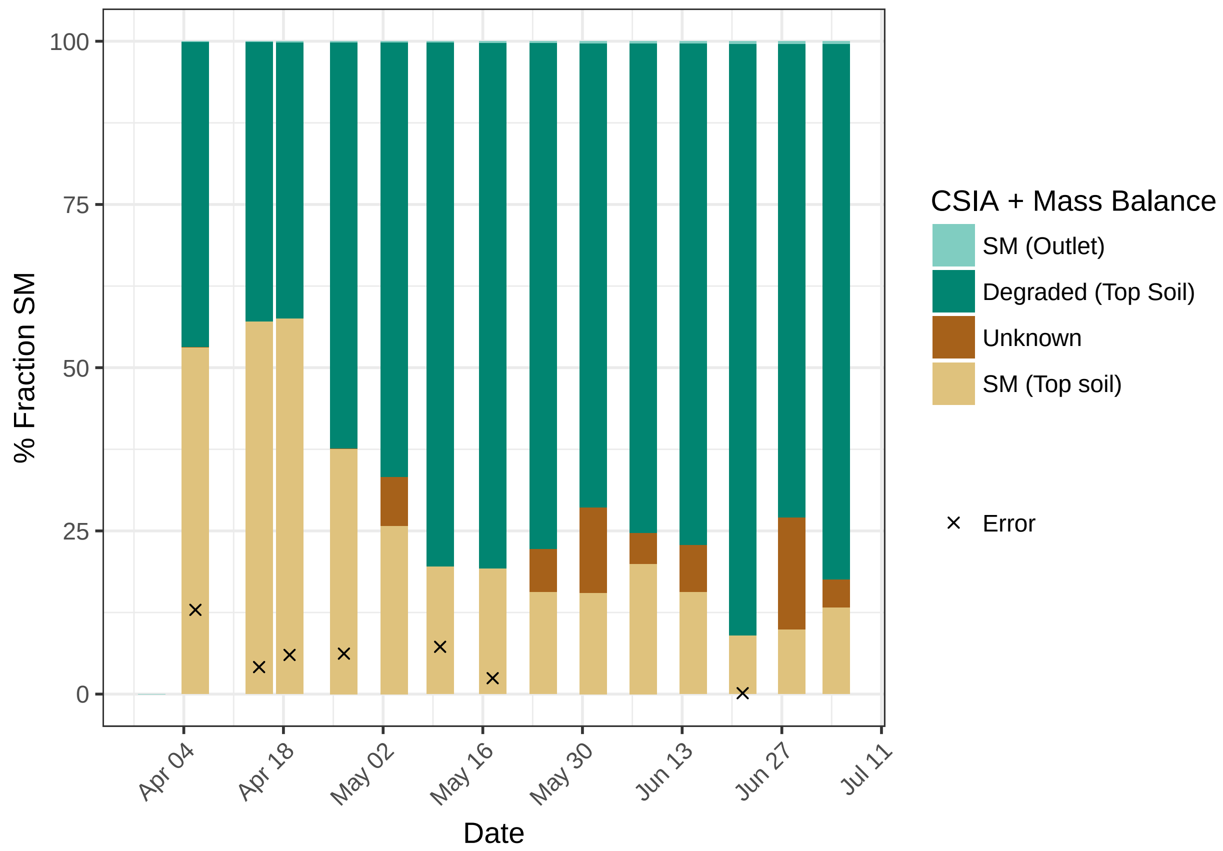
```
## [1] "Persist.Prct" "SM.PrctOut"   "F.Bulk"       "B.Bulk"
## [5] "Error"        "Unknown"
```

```
prctCSIA$variable <- factor(prctCSIA$variable, levels = c( "Error", "SM.PrctOut" , "B.Bulk", "Unknown",
                                                          "F.Bulk"
                                                      ))

prctCSIA <- within(prctCSIA, value[variable == 'Error' & value == 0] <- NA)
```

```
#massbar <- ggplot(data = prctMB , aes(x=ti, y=value))+
massbar <- ggplot(data = prctMB , aes(x=ti, y=value))+
  theme_bw() +
  # geom_bar(stat = "identity", aes(fill = variable)) +
  #geom_bar(data = subset(prctMB, !(variable %in% c("F.Bulk","B.Bulk","F.Outlet", "B.Outlet"))),
  #         aes(x=ti, y=value, fill = variable ),
  #         stat = "identity") +
  geom_bar(data = subset(prctMB,
                         variable != "F.Bulk" # & variable != "B.Bulk"
                         # & variable != "F.Outlet" # & variable != "B.Outlet"
                           ),
           aes(x=ti, y=value, fill = variable ),
           stat = "identity") +
  geom_point(data = subset(prctMB,
                           variable == "F.Bulk"  # |  variable == "B.Bulk"
                           # | variable == "F.Outlet" # | variable == "B.Outlet"
                           ),
             aes(x=ti, y=value, shape = factor(variable) )) +
  stat_smooth(data=subset(prctMB,
                           variable == "F.Bulk"),
              method = "lm", formula = y ~ poly(x, 2), se = T, aes(col = 'F.Bulk') , alpha = 0.1, size=
  # stat_smooth(data=subset(prctMB,
  #                           variable == "F.Outlet"),
  #             method = "lm", formula = y ~ poly(x, 2), se = T, aes(col = 'F.Outlet') , alpha = 0.1, si

  # Add error bars,
  # see: http://stackoverflow.com/questions/30872977/how-to-stack-error-bars-in-a-stacked-histogram-usi
  # geom_errorbar(aes(ymin=value-0.5, ymax=value+0.5), width=.5, position = "identity")+
  xlab("Date") +
  ylab("% Fraction SM") +
  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)
        ) +
  scale_fill_manual(
                    #values = c("#a6611a", "#dfc27d", "#80cdc1", "#018571"),
                    values = c("#80cdc1", "#018571", "#a6611a", "#dfc27d"),
                    name="Mass Balance", # \n
                       breaks=c("SM.PrctOut", "TPs.PrctOut", "Unknown", "Persist.Prct"),
                       labels=c("SM (Outlet)", "TPmeq (Outlet)", "Unknown", "SM (Top soil)" )) +
  scale_shape(name="CSIA", labels = c("Non-degraded (Bulk)"# , "Non-degraded (Outlet)"
                                      )) +
  scale_color_manual( name= "CSIA (LM)",
                      values = c("black"#, "dodgerblue"
                                 ),
                      labels = c("Non-degraded (Bulk)" #, "Non-degraded (Outlet)"
                                 )
                      )


  #scale_fill_brewer(# palette="OrRd",
  #                   palette = c("#a6611a", "chocolate", "green4", "dodgerblue"),
  #                  # values=c("#999999", "chocolate", "green4", "dodgerblue"),
  #                  name="% Fraction", # \n
  #                     breaks=c("Unknown", "Persist.Prct",  "SM.PrctOut", "TPs.PrctOut"),
  #                     labels=c("Unknown", "SM (Top soil)", "SM (Outlet)", "TPs (Outlet)" ))

massbar
```



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


# CSIA Based approach
csbar <- ggplot(data = prctCSIA , aes(x=ti, y=value))+
  theme_bw() +
  # geom_bar(stat = "identity", aes(fill = variable)) +
  #geom_bar(data = subset(prctMB, !(variable %in% c("F.Bulk","B.Bulk","F.Outlet", "B.Outlet"))),
  #         aes(x=ti, y=value, fill = variable ),
  #         stat = "identity") +
  geom_bar(data = subset(prctCSIA,
                         variable != "F.Bulk" & variable != "Error"
                           ),
           aes(x=ti, y=value, fill = variable ),
           stat = "identity") +
  geom_point(data = subset(prctCSIA,
                         variable == "Error"
                           ),
             aes(x=ti, y=value, shape = factor(variable) )
             ) +
  # Add error bars,
  # see: http://stackoverflow.com/questions/30872977/how-to-stack-error-bars-in-a-stacked-histogram-usi
  # geom_errorbar(aes(ymin=value-0.5, ymax=value+0.5), width=.5, position = "identity")+
  xlab("Date") +
  ylab("% Fraction SM") +
  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)
        ) +
  scale_fill_manual(
                    #values = c("#a6611a", "#dfc27d", "#80cdc1", "#018571"),
                    values = c("#80cdc1", "#018571", "#a6611a", "#dfc27d"),
                    name="CSIA + Mass Balance", # \n
                       breaks=c("SM.PrctOut", "B.Bulk", "Unknown", "Persist.Prct"),
                       labels=c("SM (Outlet)", "Degraded (Top Soil)", "Unknown", "SM (Top soil)" )) +
  scale_shape_manual(name="",
              values = (4),
              labels = c("Error"))


  #scale_fill_brewer(# palette="OrRd",
  #                   palette = c("#a6611a", "chocolate", "green4", "dodgerblue"),
  #                   # values=c("#999999", "chocolate", "green4", "dodgerblue"),
  #                   name="% Fraction", # \n
  #                      breaks=c("Unknown", "Persist.Prct",  "SM.PrctOut", "TPs.PrctOut"),
  #                      labels=c("Unknown", "SM (Top soil)", "SM (Outlet)", "TPs (Outlet)" ))

csbar
```



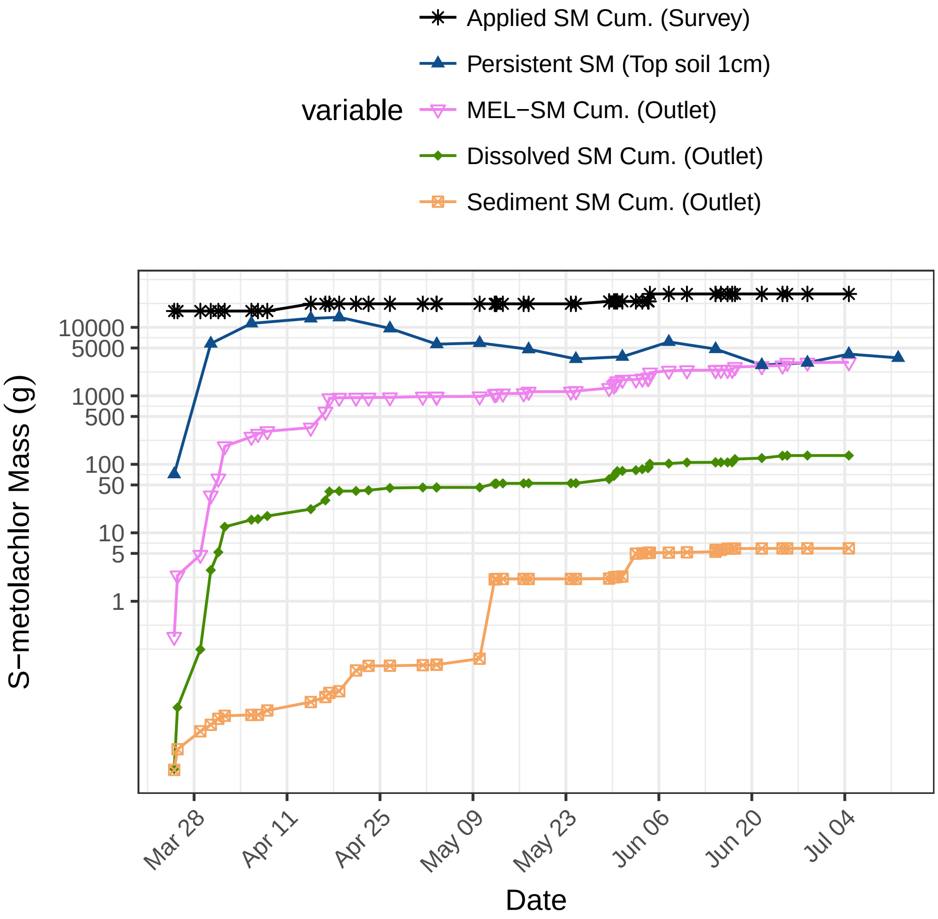
```r
# ggsave(csbar, filename = "MassBalCSIABar.png", width = 8, height = 5, units = "in", scale = 1)
```

As continous graph

```r
# Rearrange data frame
remainMassMolten <- soilsRemainMass[, c("ti" ,"CumAppMass.g", "CumOutDiss.g", "CumOutFilt.g", "CumOutME
remainMassMolten <- melt(remainMassMolten, id=c("ti"))

pg <- remainMassMolten
pg <- na.omit(pg)

# Change variable names:
levels(pg$variable)[levels(pg$variable)=="CumAppMass.g"] <- "Applied SM Cum. (Survey)"
levels(pg$variable)[levels(pg$variable)=="CumOutMELsm.g"] <- "MEL-SM Cum. (Outlet)"
levels(pg$variable)[levels(pg$variable)=="CatchMassSoil.g"] <- "Persistent SM (Top soil 1cm)"

levels(pg$variable)[levels(pg$variable)=="CumOutDiss.g"] <- "Dissolved SM Cum. (Outlet)"
levels(pg$variable)[levels(pg$variable)=="CumOutFilt.g"] <- "Sediment SM Cum. (Outlet)"


# Change the order:
levels(pg$variable)
```

```
## [1] "Applied SM Cum. (Survey)"     "Dissolved SM Cum. (Outlet)"
## [3] "Sediment SM Cum. (Outlet)"    "MEL-SM Cum. (Outlet)"
## [5] "Persistent SM (Top soil 1cm)"
```

```
pg$variable <- factor(pg$variable, levels = c("Applied SM Cum. (Survey)",  "Persistent SM (Top soil 1cm

pgSimple <- pg[which(pg$variable != ("Dissolved SM Cum. (Outlet)") & pg$variable != ("Sediment SM Cum.
# names(pg)[names(pg)=="variable"]  <- "Estimated Mass"

massBalTop <- ggplot(pg) +
   geom_line(aes(x=ti, y=value, group = variable, color=variable)) +
  geom_point(aes(x=ti, y=value, group = variable, shape=variable, color=variable)) +

  # Themes and axes
  theme_bw() +
  theme(axis.text.x=element_text(angle = 45, hjust = 1),
        # axis.text.x=element_blank(),
        # axis.title.x=element_blank(),
        legend.position="top"
        )+
  # labs(group = "Estimated Mass") +
  guides(col = guide_legend(ncol = 1)) +  # Sets legend parameters
  scale_colour_manual(values=c("black" , "dodgerblue4", "violet", "chartreuse4" , "sandybrown")) +
  scale_shape_manual(values = c(8, 17, 25, 18, 7)) +

  #scale_colour_manual(values=c("black" , "chartreuse4",  "orangered2",  "#F8766D", "dodgerblue4", "#61
  #scale_shape_manual(values = c(15, 18, 16,  23, 17, 13, 6)) +

  xlab("Date") +
  scale_x_datetime(breaks = date_breaks("2 weeks"), labels = date_format("%b %d")) +
  ylab(expression(paste("S-metolachlor Mass ", {(g)}))) +
  # scale_y_continuous(breaks = c(100, 5000, 10000, 20000), limits = c(100, 20000) )
  scale_y_continuous(trans=log_trans(), breaks=c(1,5,10,50,100,500,1000,5000, 10000))
massBalTop
```



```
massBalBottom <- ggplot(pg) +
  geom_line(aes(x=ti, y=value, color=variable)) +

  # Themes and axes
  theme_bw() +
  theme(axis.text.x=element_text(angle = 45, hjust = 1),
        #axis.text.x=element_blank(),
        #axis.title.x=element_blank(),
        legend.position="none"
        )+
  # guides(col = guide_legend(nrows = 2)) +  # Sets legend parameters
  xlab("Date") +
  scale_x_datetime(breaks = date_breaks("2 weeks"), labels = date_format("%b %d")) +
  ylab(expression(paste("Mass. S-Meto.  ", {(g)}))) +
  scale_y_continuous(breaks = c(1, 25, 50, 100), limits = c(0, 100) )


# massBal = plot_grid(massBalTop, massBalBottom, ncol = 1, nrow = 2, align = "v")



massBal_MEL <- ggplot(pgSimple) +
  geom_line(aes(x=ti, y=value, group = variable, color=variable)) +
  # Themes and axes
  theme_bw() +
  theme(# axis.text.x=element_text(angle = 45, hjust = 1),
        axis.text.x=element_blank(),
        axis.title.x=element_blank(),
        legend.position="top"

        )+
  labs(color = "Estimated Mass") +
  guides(col = guide_legend(ncol = 3)) +  # Sets legend parameters

  # xlab("Date") +
  scale_x_datetime(breaks = date_breaks("2 weeks"), labels = date_format("%b %d")) +
  ylab(expression(paste("S-metolachlor ", {(g)})))

# massBal_MEL

massBalLegend <- get_legend(massBalTop)
```

Catchment degradation based on bulk signatures

```
# Pure and cuve isotope average
d13Co
```

```
## [1] -32.235
```

```
# Lab enrichment
epsilon_mean
```

```
## [1] -1.75
```

```
if (SHOW) {
  # Remaining fraction
  soilsOut$DD13C.bulk <- (soilsOut$BulkCatch.d13 - (d13Co))

  # Max epsilon (30C, 20%)
  soilsOut$f.max.bulk <-
    ((10^(-3)*soilsOut$BulkCatch.d13 + 1)/(10^(-3)*d13Co + 1))^(1000/(epsilon_max))

  soilsOut$B.max.bulk <-
    (1 - soilsOut$f.max.bulk)*100

  # Min epsilon (20C, 40%)
  soilsOut$f.min.bulk <-
    ((10^(-3)*soilsOut$BulkCatch.d13 + 1)/(10^(-3)*d13Co + 1))^(1000/(epsilon_min))

  soilsOut$B.min.bulk <-
    (1 - soilsOut$f.min.bulk)*100

  # Mean epsilon (# Alteck)
  soilsOut$f.mean.bulk <-
    ((10^(-3)*soilsOut$BulkCatch.d13 + 1)/(10^(-3)*d13Co + 1))^(1000/(epsilon_mean))
  soilsOut$B.mean.bulk <-
    (1 - soilsOut$f.mean.bulk)*100



  ggplot(soilsOut, aes(x=ti, y=B.mean.bulk)) +
    geom_point() +
    # geom_point(aes(x=Date.ti, y=B.comp, colour=Transect, group = Transect)) +

    # Theme and axes
    theme_bw() +
    ylab("Degr. %") +
    theme(legend.position = "top",
          #axis.title = element_blank(),
          #axis.title.x = element_blank(),
          #axis.text.x = element_blank()
          axis.text.x=element_text(angle = 45, hjust = 1)
          ) +
    xlab("Date") +
    scale_x_datetime(breaks = date_breaks("2 weeks"), labels = date_format("%b %d")) +

    # scale_y_continuous(breaks = c(25, 50, 75, 100), limits = c(0, 100) ) +
    # stat_smooth(method = "lm", formula = y ~ poly(x, 2), se=FALSE)
    # geom_smooth(data=subset(weeklySoil[14:28, ]), method = "lm", formula = y ~ poly(x, 2), se = F) +
    geom_smooth(aes(group = 1), method = "lm", formula = y ~ poly(x, 2))
    # stat_smooth(data=subset(weeklySoil[4:39, ]), method = "lm", formula = y ~ poly(x, 2), se = F)
    # stat_smooth(method = "lm", formula = y ~ x, se=FALSE)
    #geom_text_repel(aes(label=Wnum, color = factor(Transect)),
     #              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)

  # bulkB
}
```

Degradation based on transect signatures

```
if (CHECK_ERR) {
# Merge Bulk Degradation into weekly soil dataframe (only one (e.g. South) transect as root)
soilsOut$PersistPrct <- (soilsOut$CatchMassSoil.g/soilsOut$CumAppMass.g)*100
soilsOut$DischPrct <- ((soilsOut$CumOutDiss.g+soilsOut$CumOutFilt.g)/soilsOut$CumAppMass.g)*100
bulkDegDF <-  soilsOut[, c("ti", "WeekSubWeek", "ID.S" ,"BulkCatch.d13", "B.mean.bulk", "B.max.bulk", "

bulkDegDF$TotBL <- bulkDegDF$B.min.bulk + bulkDegDF$PersistPrct + bulkDegDF$DischPrct
bulkDegDF$LeachPrct <- 100-bulkDegDF$TotBL
bulkDegDF$LeachPrctCorr <- ifelse(bulkDegDF$LeachPrct > 0, bulkDegDF$LeachPrct, NA)
```

```r
# Delerte rows from specified Columns
completeFun <- function(data, desiredCols) {
  completeVec <- complete.cases(data[, desiredCols])
  return(data[completeVec, ])
}

bulkDegDF <- completeFun(bulkDegDF, "TotBL")

bulkDegDF$Transect <- "Degraded (Bulk)"

# names(bulkDegDF)[names(bulkDegDF) == "B.mean.bulk"] <- "B.mean.com"
names(bulkDegDF)[names(bulkDegDF) == "ti"] <- "Date.ti"

#Splitting the identifier name into Type, Week No., tc..
bulkDegDF$ID.S <- as.character(bulkDegDF$ID.S)
split <- strsplit(bulkDegDF$ID.S, "AW-S-", fixed = T)
bulkDegDF$Wnum <- sapply(split, "[", 2) # Creates new column without "Split0"

bulkDegDF$Week = paste("W", bulkDegDF$Wnum, sep = "")

bulkDegDF <- bulkDegDF[, c("Date.ti", "Transect",
                           "B.mean.bulk",  "B.max.bulk",  "B.min.bulk",
                           "PersistPrct", "DischPrct", "TotBL", "LeachPrct", "LeachPrctCorr",
                           "Week")]
bulkDegDF$RemainLabel <- "Persistent frac. (Top Soil 1cm)"
bulkDegDF$DischLabel <- "Discharge (Outlet)"
bulkDegDF$LeachLabel <- "Leached (Inferred)"


levels(bulkDegDF$Transect)[levels(bulkDegDF$Transect)=="Bulk"] <- "Degraded (Bulk)"


wSoil <- weeklySoil[, c("Date.ti", "Transect", "B.mean.comp", "B.max.comp", "B.min.comp")]
levels(wSoil$Transect)[levels(wSoil$Transect)=="N"] <- "North"
levels(wSoil$Transect)[levels(wSoil$Transect)=="T"] <- "Talweg"
levels(wSoil$Transect)[levels(wSoil$Transect)=="S"] <- "South"

# colnames(wSoil) <- c("Date.ti", "Transect", "B.mean.bulk",  "B.max.bulk",  "B.min.bulk")

# wSoil$Week <- NA
# wSoilBulkDeg <- rbind(wSoil, bulkDegDF)

#names(bulkDegDF)[names(bulkDegDF) == "ID.S"] <- "ID"
#wSoilBulkDeg <- merge(weeklySoil, bulkDegDF, by="ID", all = T)
#wSoilBulkDeg$BulkLabel[!is.na(wSoilBulkDeg$WeekSubWeek)] <- "Bulk"


levels(wSoil$Transect)
#wSoil$Transect <- factor(wSoil$Transect, levels = c("Bulk Fraction", "North", "Talweg", "South" ))
wSoil$Transect <- factor(wSoil$Transect, levels = c("North", "Talweg", "South" ))

#wSoilBulkDeg$B.max.bulk[wSoilBulkDeg$Transect == "North"] <- NA
#wSoilBulkDeg$B.max.bulk[wSoilBulkDeg$Transect == "Talweg"] <- NA
```

```
#wSoilBulkDeg$B.max.bulk[wSoilBulkDeg$Transect == "South"] <- NA

#wSoilBulkDeg$B.min.bulk[wSoilBulkDeg$Transect == "North"] <- NA
#wSoilBulkDeg$B.min.bulk[wSoilBulkDeg$Transect == "Talweg"] <- NA
#wSoilBulkDeg$B.min.bulk[wSoilBulkDeg$Transect == "South"] <- NA

limits_bulkdeg <- aes(ymin=B.min.bulk, ymax=B.max.bulk, x = Date.ti, colour = Transect, group = Transect

Bsoil1 <-
  ggplot() +
  # geom_point(size = 2) +
  # ggplot(data = wSoilBulkDeg, aes(Date.ti, B.mean.bulk, colour = Transect, shape=Transect, group = Tr
  geom_point(data = wSoil,
             aes(Date.ti, B.mean.comp, colour = Transect, shape=Transect, group = Transect), size = 2) +
  geom_point(data = bulkDegDF,
             aes(Date.ti, B.mean.bulk, colour = Transect, shape=Transect, group = Transect), size = 2) +
  geom_point(data = bulkDegDF,
             aes(Date.ti, PersistPrct, colour = RemainLabel, shape=RemainLabel, group = RemainLabel), si
  geom_point(data = bulkDegDF,
             aes(Date.ti, DischPrct, colour = DischLabel, shape=DischLabel, group = DischLabel), size =
  geom_point(data = bulkDegDF,
             aes(Date.ti, LeachPrctCorr, colour = LeachLabel, shape=LeachLabel, group = LeachLabel), si

  geom_errorbar(data = bulkDegDF, limits_bulkdeg) + # With 2 data frames
  stat_smooth(data=subset(bulkDegDF, Transect == "Degraded (Bulk)"),
              mapping = aes(y = B.mean.bulk, x = Date.ti), # With 2 data frames
              colour = " black",
              method = "lm", formula = y ~ poly(x, 2), se=F) +

  stat_smooth(data=subset(bulkDegDF, RemainLabel == "Persistent frac. (Top Soil 1cm)"),
              mapping = aes(y = PersistPrct, x = Date.ti), # With 2 data frames
              colour = "darkblue",
              method = "lm", formula = y ~ poly(x, 2), se=F) +
  scale_colour_manual(values=c("black" , "chartreuse4",  "orangered2",  "#F8766D", "dodgerblue4", "#619
  scale_shape_manual(values = c(15, 18, 16,  23, 17, 13, 6)) +
  theme_bw() +
  ylab("S-metolachlor Sinks (%)") +
  theme(legend.position = "top",
        legend.title = element_blank(),
        #axis.title = element_blank(),
        axis.title.x = element_blank(),
        axis.text.x = element_blank()
        #axis.text.x=element_text(angle = 45, hjust = 1)
        ) +
  # xlab("Date") +
  scale_x_datetime(breaks = date_breaks("2 weeks"), labels = date_format("%b %d")) +
  # ylab(expression(paste("Degradation %"))) +

  # scale_y_continuous(breaks = c(25, 50, 75, 100), limits = c(0, 100) ) +

  # geom_smooth(data=subset(weeklySoil[14:28, ]), method = "lm", formula = y ~ poly(x, 2), se = F) +
  # geom_smooth(aes(group = 1), method = "lm", formula = y ~ poly(x, 2)) +
  # stat_smooth(data=subset(weeklySoil[4:39, ]), method = "lm", formula = y ~ poly(x, 2), se = F)
```

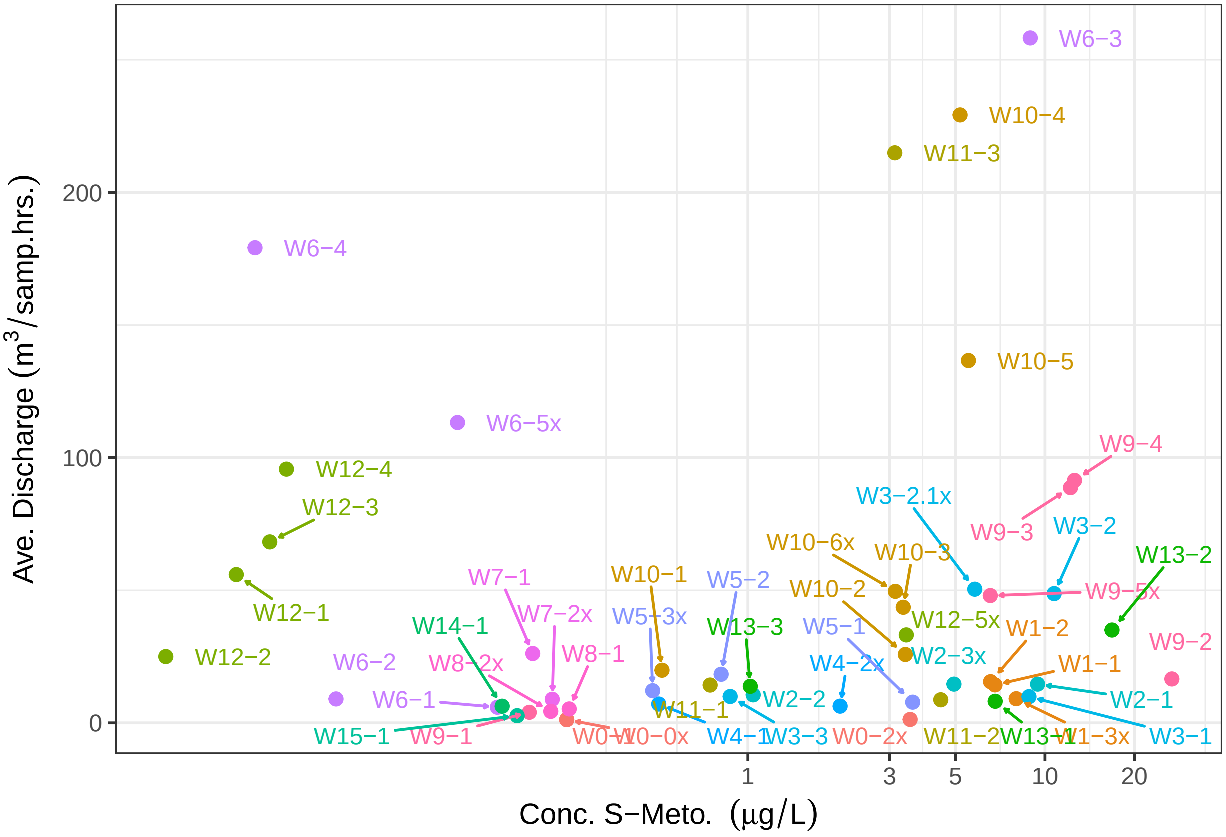
```r
  # stat_smooth(method = "lm", formula = y ~ x, se=FALSE)
  geom_text_repel(data=subset(bulkDegDF, Transect == "Degraded (Bulk)"),
                  mapping = aes(y=B.mean.bulk, x= Date.ti, label=Week),
                  size = 3,
                  arrow = arrow(length = unit(0.005, 'npc'), type = "closed"),
                  force = 0.9,
                  point.padding = unit(0.9, 'lines'),
                  max.iter = 2e3,
                  nudge_x = .05, show.legend = F)

Bsoil1

colnames(bulkDegDF)[which(names(bulkDegDF) == "Transect")] <- "Catchment"
limits_bulkdegCAT <- aes(ymin=B.min.bulk, ymax=B.max.bulk, x = Date.ti, colour = Catchment, group = Cat
solidpts <- ggplot() +
  # geom_point(size = 2) +
  # ggplot(data = wSoilBulkDeg, aes(Date.ti, B.mean.bulk, colour = Transect, shape=Transect, group = Tr
  geom_point(data = bulkDegDF,
             aes(Date.ti, B.mean.bulk, colour = Catchment, shape=Catchment, group = Catchment), size = 2
  geom_point(data = bulkDegDF,
             aes(Date.ti, PersistPrct, colour = RemainLabel, shape=RemainLabel, group = RemainLabel), si
  geom_point(data = bulkDegDF,
             aes(Date.ti, DischPrct, colour = DischLabel, shape=DischLabel, group = DischLabel), size =
  geom_point(data = bulkDegDF,
             aes(Date.ti, LeachPrctCorr, colour = LeachLabel, shape=LeachLabel, group = LeachLabel), si

  geom_errorbar(data = bulkDegDF, limits_bulkdegCAT) + # With 2 data frames
  stat_smooth(data=subset(bulkDegDF, Catchment == "Degraded (Bulk)"),
              mapping = aes(y = B.mean.bulk, x = Date.ti), # With 2 data frames
              colour = " black",
              method = "lm", formula = y ~ poly(x, 2), se=F) +

  stat_smooth(data=subset(bulkDegDF, RemainLabel == "Persistent frac. (Top Soil 1cm)"),
              mapping = aes(y = PersistPrct, x = Date.ti), # With 2 data frames
              colour = "darkblue",
              method = "lm", formula = y ~ poly(x, 2), se=F) +
  #scale_colour_manual(values=c(" black" ,"#F8766D", "#7CAE00", "#619CFF")) +
  scale_colour_manual(values=c("black" , "chartreuse4",  "orangered2",  "dodgerblue4")) +
  scale_shape_manual(values = c(15, 18,  16, 17)) +
  theme_bw() +
  theme(legend.position = "top",
        legend.title = element_blank(),
        #axis.title = element_blank(),
        axis.title.x = element_blank(),
        axis.text.x = element_blank()
        #axis.text.x=element_text(angle = 45, hjust = 1)
        ) +
  # xlab("Date") +
  scale_x_datetime(breaks = date_breaks("2 weeks"), labels = date_format("%b %d")) +
  geom_text_repel(data=subset(bulkDegDF, Catchment == "Degraded (Bulk)"),
                  mapping = aes(y=B.mean.bulk, x= Date.ti, label=Week),
                  size = 3,
                  arrow = arrow(length = unit(0.005, 'npc'), type = "closed"),
                  force = 0.9,
                  point.padding = unit(0.9, 'lines'),
                  max.iter = 2e3,
                  nudge_x = .05, show.legend = F)

solidpts

#colnames(wSoil)[which(names(wSoil) == "Transect")] <- "Transect-Deg."
emptypts <-
  ggplot() +
  # geom_point(size = 2) +
  # ggplot(data = wSoilBulkDeg, aes(Date.ti, B.mean.bulk, colour = Transect, shape=Transect, group = Tr
  geom_point(data = wSoil,
             aes(Date.ti, B.mean.comp, colour = Transect , shape=Transect , group = Transect ), size = 2

  #scale_colour_manual(values=c(" black" ,"#F8766D", "#7CAE00", "#619CFF")) +

  scale_colour_manual(values=c("#F8766D","#7CAE00",  "#619CFF")) +
  scale_shape_manual(values = c(23,  6,  13)) +
  # guides(guide_legend(title = waiver()))+
  # labs(color = "Number of gears") +
  #scale_colour_manual(values=c("black" , "chartreuse4",  "orangered2",  "#F8766D", "dodgerblue4", "#61
  #scale_shape_manual(values = c(15, 18, 16,  23, 17, 13, 6)) +
  theme_bw() +
  ylab("S-metolachlor Sinks (%)") +
  theme(#legend.position = "top",
        # legend.title = element_blank(),
        #axis.title = element_blank(),
        axis.title.x = element_blank(),
        axis.text.x = element_blank()
        #axis.text.x=element_text(angle = 45, hjust = 1)
        ) +
  # xlab("Date") +
  scale_x_datetime(breaks = date_breaks("2 weeks"), labels = date_format("%b %d"))

emptypts
#colnames(wSoil)[which(names(wSoil) == "Transect-Deg.")] <- "Transect"

## Merging both figures

# massBalNoL <- massBal + theme(legend.position='none')
BsoilNoL <- Bsoil1 + theme(legend.position='none')
MBalNoL <- massBalTop + theme(legend.position = 'none')

legend_deg_solid = get_legend(solidpts +
                          theme(legend.title = element_text(face = "bold"),
                                legend.text = element_text(size = 9),
                                legend.key.height = unit(x = 0.35, units = 'cm'),
                                legend.background = element_rect(colour = "black")) +
                          guides(colour = guide_legend(title.position = "top", ncol = 1, byrow = T)))

legend_deg_empty = get_legend(emptypts +
                          theme(legend.title = element_text(face = "bold"),
                                legend.text = element_text(size = 9),
                                legend.key.height = unit(x = 0.35, units = 'cm'),
                                legend.background = element_rect(colour = "black")) +
                          guides(colour = guide_legend(title.position = "top", ncol = 1, byrow = T)))

legend_mass = get_legend(massBalTop +
                           theme(legend.text = element_text(size = 9),
                                 legend.key.height = unit(x = 0.4, units = 'cm'),
                                 legend.background = element_rect(colour = "black"),
                                  legend.title = element_blank()))
                         #+   guides(colour = guide_legend(title = "Mass Distribution",  title.position

gridPartB <- plot_grid(BsoilNoL, MBalNoL,
                    ncol =1, nrow = 2, align ="v",
                    labels = c("A", "B"))
gridPartB

fig2 <- ggdraw() +
  draw_plot(gridPartB, x=0, y=0, width =  0.93, height = 1) +
  draw_plot(legend_deg_solid, x=0.48, y =0.62, width = 0.70, height = 0.20) +
  draw_plot(legend_deg_empty, x=0.52, y =0.65, width = 0.75, height = 0.35) +
  draw_plot(legend_mass, x=0.5, y=0.05, width = 0.63, height = 0.28)

fig2

# ggsave(fig2, filename = "BvsMassBal.tiff", height = 17, width = 17.8, units = 'cm')
}
```

Degradation per transect (no bulk)

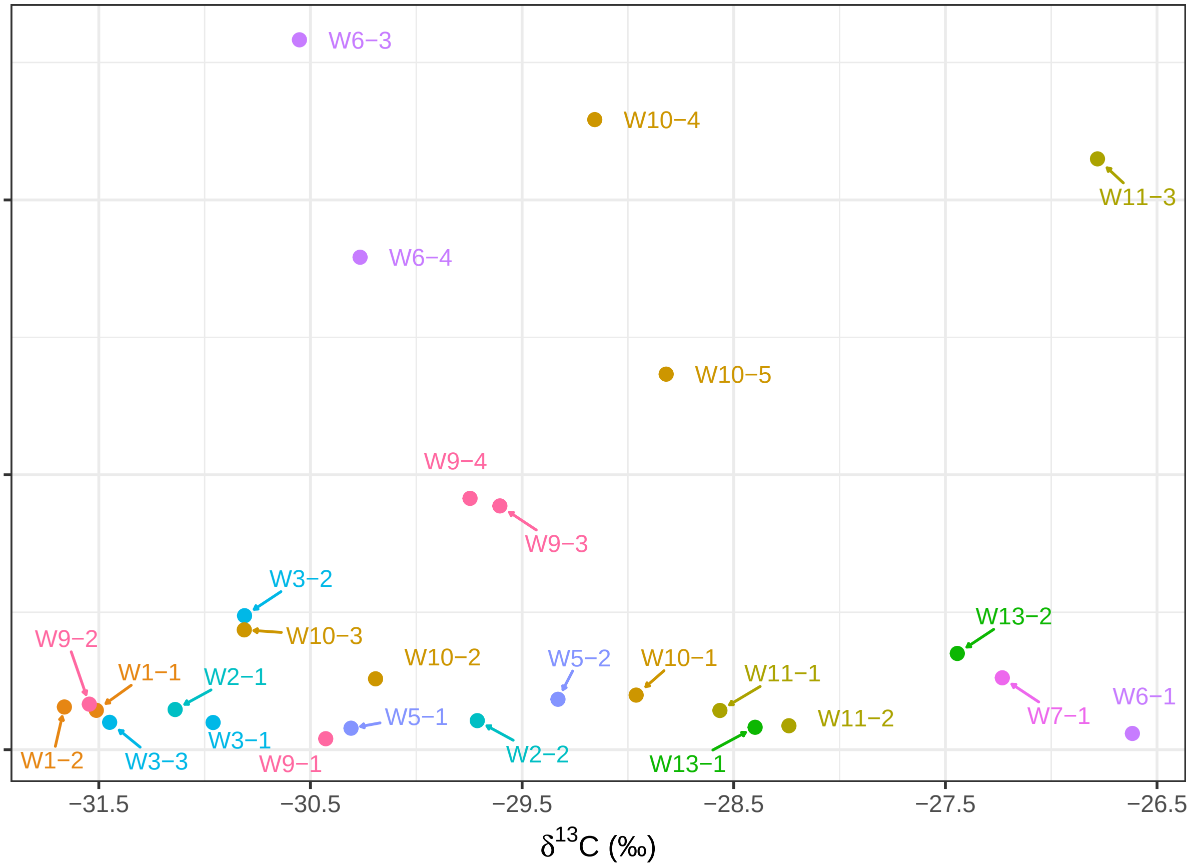
```
if (SHOW) {
  Bsoil2 = ggplot(wSoil, aes(x=Date.ti, y=B.mean.comp, colour=Transect, shape=Transect, group = Transec
    # Bsoil2 = ggplot(wSoilBulkDeg, aes(x=Date.ti)) +
    #
    # stat_smooth(aes(y=B.mean.bulk), method = "lm", formula = y ~ poly(x, 2), se=T) +
    # geom_point(aes(y=B.mean.com, colour=Transect, group = Transect)) +
    # geom_point(aes(y=B.mean.bulk, group=BulkLabel, colour="Bulk Isotopes")) +
    geom_point(size = 2) +

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

    scale_colour_manual(values=c("#F8766D", "#7CAE00", "#619CFF")) +
    scale_shape_manual(values = c(15, 16, 17)) +
    # scale_shape_manual(values = c(23, 15, 16, 17)) +
    # geom_point(aes(x=Date.ti, y=B.comp, colour=Transect, group = Transect)) +
    # Theme and axes
    theme_bw() +
    ylab("Degr. %") +
    theme(legend.position = "top",
          #axis.title = element_blank(),
          axis.title.x = element_blank(),
          axis.text.x = element_blank()
          #axis.text.x=element_text(angle = 45, hjust = 1)
          ) +
    # xlab("Date") +
    scale_x_datetime(breaks = date_breaks("2 weeks"), labels = date_format("%b %d"))
    # ylab(expression(paste("Degradation %"))) +

    # scale_y_continuous(breaks = c(25, 50, 75, 100), limits = c(0, 100) ) +

    # geom_smooth(data=subset(weeklySoil[14:28, ]), method = "lm", formula = y ~ poly(x, 2), se = F) +
    # geom_smooth(aes(group = 1), method = "lm", formula = y ~ poly(x, 2)) +
    # stat_smooth(data=subset(weeklySoil[4:39, ]), method = "lm", formula = y ~ poly(x, 2), se = F)
    # stat_smooth(method = "lm", formula = y ~ x, se=FALSE)


    #geom_text_repel(aes(y=B.mean.bulk, label=Wnum, color = factor(Transect)),
     #               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)

  Bsoil2
}
```

XY-Plots

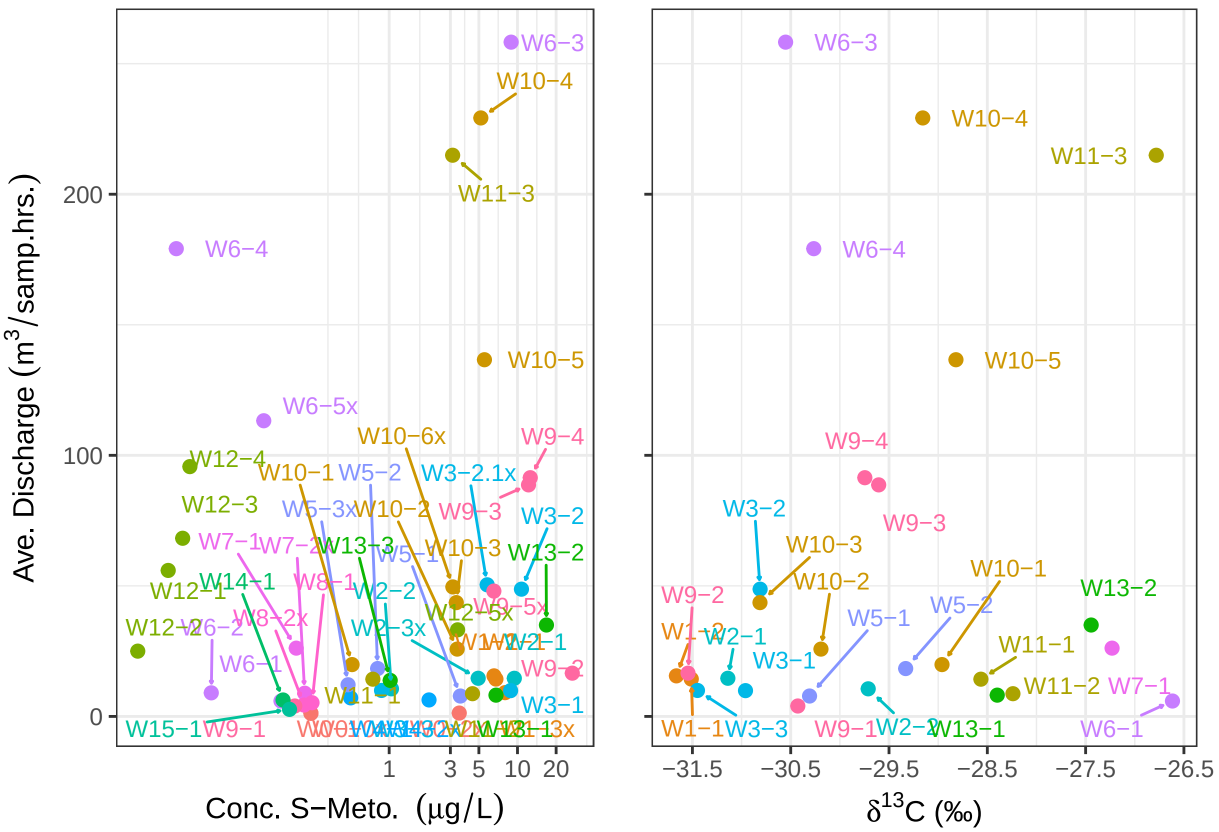
```
QC <- ggplot(AOdf, aes(y=AveDischarge.m3.h, x=Conc.mug.L, group = WeekSubWeek, color = Weeks)) +
  geom_point(size = 2) +
  theme_bw() +
  theme(axis.text.y = element_blank()) +
  theme(legend.title=element_blank()) +
  theme(plot.margin = unit(c(0,0.5,0,0), "lines")) +
  #stat_smooth(method = "lm", formula = y ~ poly(x, 2)) +
  theme_bw() +
  theme(legend.position="none") +
  #scale_y_continuous(trans=log_trans(), breaks=c(1, 5, 10, 50, 100, 200)) +
  scale_x_continuous(trans=log_trans(), breaks=c(1, 3, 5, 10, 20)) +
  ylab(expression(paste("Ave. Discharge ", {(m^{3} / samp.hrs. )}))) +
  xlab(expression(paste("Conc. S-Meto.  ", {({mu}*g / L)}))) +
  geom_text_repel(aes(label=WeekSubWeek, color = factor(Weeks)),
                  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)

QC
```



```
QD <- ggplot(AOdf, aes(y=AveDischarge.m3.h, x=diss.d13C, group = WeekSubWeek, color = Weeks)) +
  geom_point(size = 2) +
  theme_bw() +
  theme(axis.text.y = element_blank()) +
  theme(plot.margin = unit(c(0,0.8,0,0), "lines")) +
  #theme(legend.title=element_blank()) +
  #theme(legend.text = element_text(size = 10)) +
  theme(legend.position="none") +
  #stat_smooth(method = "lm", formula = y ~ poly(x, 2)) +
  #scale_y_continuous(trans=log_trans(), breaks=c(1, 3, 5, 8, 10, 30, 50, 80, 100, 300)) +
  ylab(expression(paste("Ave. Discharge ", {(m^{3} / sample)}))) +
  ylab("") +
  scale_x_continuous(breaks=seq(-31.5, -26.5, 1)) +
  xlab(expression(paste({delta}^"13","C", ' (\u2030)'))) +
  geom_text_repel(aes(label=WeekSubWeek, color = factor(Weeks)),
                  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)

QD
```



```
acd = plot_grid(QC, QD, ncol = 2, nrow = 1, align = "h")
acd
```



```
#ggsave(acd, filename = "Disch_Conc_Delta_XYlabs.png", width = 8, height = 5, units = "in", scale = 1)
#ggsave(acd, filename = "Disch_Conc_Delta_XY.png", width = 8, height = 5, units = "in", scale = 1)
#ggsave(acd, filename = "Disch_Conc_Delta_W.pdf", width = 8, height = 4.6, units = "in", scale = 1)
```