# Rayleigh Calculations

*PAZ*

*06/04/2017*

## Introduction

Degradation extent is calculated based on closed and open system Rayleigh equations

## Packages

```
library(sm)
library(vioplot)

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

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

library("plotKML")

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

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

## Lab parameters

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

# rho1 = 0.99*10^6 # soil density [g/m3] (assume for April)
```

```
# rho2 = 2.20*10^6 # soil density [g/m3] (assume for May & June)
# depth = 0.01 # [m]
```

## Soils

```
soils = read.csv2("Data/MassBalance_R.csv", # File contains both soils & waters
                  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"
```

```
soils$ti <- as.POSIXct(strptime(soils$ti, "%Y-%m-%d %H:%M", tz="EST"))
colnames(soils)[colnames(soils) == "ti"] <- "Date.ti"

keepSoils <- c(
  "Date.ti", "WeekSubWeek", "ID.N",
  "Area.N", "Area.T", "Area.S",
  "iniCo.ug.g.N", "iniCo.ug.g.T", "iniCo.ug.g.S",
  # Applied
  #"CumAppMass.g.N", "CumAppMass.g.T", "CumAppMass.g.S",
  #"CumAppMass.g" ,
```

```
  "CumAppMass.g.OT", "CumAppMass.g.N.OT", "CumAppMass.g.T.OT" , "CumAppMass.g.S.OT",
  # Concentrations
  "BulkCatch.Conc", "iniCo.Bulk",
  "Conc.mug.g.dry.soil.N", "Conc.mug.g.dry.soil.T", "Conc.mug.g.dry.soil.S",
  # Masses
  "CatchMassSoil.g", "CatchMassSoil.g.SD",
  "MassSoil.g.North", "MassSoil.g.Talweg", "MassSoil.g.South" ,
  "MassSoil.g.SD.North" , "MassSoil.g.SD.Talweg", "MassSoil.g.SD.South",
  # Isotopes
  "diss.d13C", "SD.d13C",
  "comp.d13C.North" , "comp.d13C.SD.North",
  "comp.d13C.Talweg", "comp.d13C.SD.Talweg",
  "comp.d13C.South" , "comp.d13C.SD.South",
  "BulkCatch.d13", "BulkCatch.d13.SD")

# Test
sum(is.na(soils$Date.ti)) == 0
```

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

```
soils <- soils[, colnames(soils) %in% keepSoils]
```

## Rayleigh (closed system, Elsner's notation)

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

were,

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

### Accounting for dilution

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

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

Following Van Breukelen (2007),

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

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

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

were,

$$\Delta = 1000 \cdot ln\Big(\frac{10^{-3}\delta_t^{13}C + 1}{10^{-3}\delta_0^{13}C + 1}\Big)$$

### Dilution factor ($F$)

Note that to meet the relationships D* > 1 and 0 < B* < 1 initial concentrations need to be accurate. From Van Breukelen (2007):

(i) if B* exceeds 1, the predicted concentration decrease is larger than observed,
(ii) if B* is negative, downgradient concentrations are higher than the source concentration.

Modified to top soils, in the above cases, application of the open system Rayleigh equation [would] thereby point out that the boundary conditions [e.g. initial concentrations] must be different.

```
soils$ftot.N <-soils$Conc.mug.g.dry.soil.N/soils$iniCo.ug.g.N
soils$ftot.T <-soils$Conc.mug.g.dry.soil.T/soils$iniCo.ug.g.T
soils$ftot.S <-soils$Conc.mug.g.dry.soil.S/soils$iniCo.ug.g.S

soils$ftot.Bulk <-soils$BulkCatch.Conc/soils$iniCo.Bulk

# Van Breukelen notation
soils$Delta.N <- 1000*log( (10^-3*soils$comp.d13C.North +1)/(10^-3*d13Co+1) )
soils$Delta.T <- 1000*log( (10^-3*soils$comp.d13C.Talweg +1)/(10^-3*d13Co+1) )
soils$Delta.S <- 1000*log( (10^-3*soils$comp.d13C.South +1)/(10^-3*d13Co+1) )
soils$Delta.Bulk <- 1000*log( (10^-3*soils$BulkCatch.d13 +1)/(10^-3*d13Co+1) )

soils$Fdil.N =
  exp( soils$Delta.N/epsilon_lab -log(soils$ftot.N) )
soils$Fdil.T =
  exp( soils$Delta.T/epsilon_lab -log(soils$ftot.T) )
soils$Fdil.S =
  exp( soils$Delta.S/epsilon_lab -log(soils$ftot.S) )

soils$Fdil.Bulk =
      exp( soils$Delta.Bulk/epsilon_lab -log(soils$ftot.Bulk) )
median(soils$Fdil.N, na.rm = T)
```

```
## [1] 2.161475
```

```
median(soils$Fdil.T, na.rm = T)
```

```
## [1] 1.253106
```

```
median(soils$Fdil.S, na.rm = T)
```

```
## [1] 1.107638
```

```
median(soils$Fdil.Bulk, na.rm = T)
```

```
## [1] 1.542359
```

```
  # soils$Fdil <- ifelse(soils$Fdil < 1, NA, soils$Fdil)
```

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

```
soils$fdil.N <- 1/soils$Fdil.N
soils$fdeg.N <- soils$ftot.N * soils$Fdil.N

soils$Dprct.N <- (1- soils$fdil.N)*100
soils$Bprct.N <- (1-soils$fdeg.N)*100
soils$Tprct.N <- (1- (1 - soils$Bprct.N/100) * (1 - soils$Dprct.N/100)  )*100
```

```
# Talweg
soils$fdil.T <- 1/soils$Fdil.T
soils$fdeg.T <- soils$ftot.T * soils$Fdil.T

soils$Dprct.T <- (1- soils$fdil.T)*100
soils$Bprct.T <- (1-soils$fdeg.T)*100
soils$Tprct.T <- (1-soils$ftot.T)*100

#South
soils$fdil.S <- 1/soils$Fdil.S
soils$fdeg.S <- soils$ftot.S * soils$Fdil.S

soils$Dprct.S <- (1- soils$fdil.S)*100
soils$Bprct.S <- (1-soils$fdeg.S)*100
soils$Tprct.S <- (1-soils$ftot.S)*100

# Bulk
soils$fdil.Bulk <- 1/soils$Fdil.Bulk
soils$fdeg.Bulk <- soils$ftot.Bulk * soils$Fdil.Bulk

soils$Dprct.Bulk <- (1- soils$fdil.Bulk)*100
soils$Bprct.Bulk <- (1-soils$fdeg.Bulk)*100
soils$Tprct.Bulk <- (1-soils$ftot.Bulk)*100
```

Calculating degradation and respective statistical variation errors in B, according to (**???**):

$$|\Delta B| = \frac{1}{|\epsilon|} \cdot (100 - B[\%]) \cdot \sqrt{(ln(1 - B[\%]/100))^2 \cdot (\Delta\epsilon)^2 + (\Delta\delta_x)^2 + (\Delta\delta_0)^2}$$

```
# Deg effective
soils$DD.N <-  soils$comp.d13C.North - d13Co
soils$DD.T <-  soils$comp.d13C.Talweg - d13Co
soils$DD.S <-  soils$comp.d13C.South - d13Co
soils$DD.Bulk <- soils$BulkCatch.d13 - d13Co

# From extraction methods (Charline's article):
# For soils at 20% and 40% water content, could have reached up to: 0.765

DeltaX <- 0.765 # From extraction method, instead of individual sample SD?
Deltadelta0 <- 0.43 #


soils$Belsner.N <- (1- ((1000 + d13Co + soils$DD.N)/(1000+ d13Co))^(1000/epsilon_lab) )*100
soils$Belsner.T <- (1- ((1000 + d13Co + soils$DD.T)/(1000+ d13Co))^(1000/epsilon_lab) )*100
soils$Belsner.S <- (1- ((1000 + d13Co + soils$DD.S)/(1000+ d13Co))^(1000/epsilon_lab) )*100


# Statistical error
soils$Berr.Stat.N <- (1/-epsilon_lab)*
  (100-soils$Belsner.N)*( (log(1-soils$Belsner.N/100))^2
                          * dE^2 + (soils$comp.d13C.SD.North)^2 + (Deltadelta0)^2 )^(1/2)

soils$Berr.Stat.T <- (1/-epsilon_lab)*
  (100-soils$Belsner.T)*( (log(1-soils$Belsner.T/100))^2
```

```
                          * dE^2 + (soils$comp.d13C.SD.Talweg)^2 + (Deltadelta0)^2 )^(1/2)

soils$Berr.Stat.S <- (1/-epsilon_lab)*
  (100-soils$Belsner.S)*( (log(1-soils$Belsner.S/100))^2
                          * dE^2 + (soils$comp.d13C.SD.South)^2 + (Deltadelta0)^2 )^(1/2)

soils$Belsner.Bulk <- (1- ((1000 + d13Co + soils$DD.Bulk)/(1000+ d13Co))^(1000/epsilon_lab) )*100
```

Calculating errors from systematic variation in input parameters, in this case comparing $\epsilon_{lab}$ vs $\epsilon_{field}$:

```
soils$Belsner.BulkField <- (1- ((1000 + d13Co + soils$DD.Bulk)/(1000+ d13Co))^(1000/epsilon_field) )*10
soils$Bdiff.LabField <- soils$Belsner.BulkField - soils$Belsner.Bulk
```

```
soils$Berr.Stat.Bulk <- (1/-epsilon_lab)*
  (100-soils$Belsner.Bulk)*( (log(1-soils$Belsner.Bulk/100))^2 * dE^2 + (0.5)^2 + (.5)^2 )^(1/2)

soils$Berr.Stat.Field <- (1/-epsilon_field)*
  (100-soils$Belsner.BulkField)*( (log(1-soils$Belsner.BulkField/100))^2 * dE^2 + (0.5)^2 + (.5)^2 )^(1/
```

The above allows us to calculate the breakdown (B*) and dilution factors (D*). Note that the relationships D* $> 1$ and B* $< 1$ must be met, otherwise $\epsilon_{true}$ is incorrect or iniitial concentrations are incorrect (or both).

Calculating B* is given by:

$$B^* = \frac{log(f_{dil})}{log(f_{tot})}$$

and their relationship $D*/B*$ which relates the extent of dilution relative to degradation.

```
soils$Dstar.N = log(soils$fdil.N)/log(soils$ftot.N)
soils$Bstar.N = log(soils$fdeg.N)/log(soils$ftot.N)
soils$DB.N = soils$Dstar.N/soils$Bstar.N

soils$Dstar.T = log(soils$fdil.T)/log(soils$ftot.T)
soils$Bstar.T = log(soils$fdeg.T)/log(soils$ftot.T)
soils$DB.T = soils$Dstar.T/soils$Bstar.T

soils$Dstar.S = log(soils$fdil.S)/log(soils$ftot.S)
soils$Bstar.S = log(soils$fdeg.S)/log(soils$ftot.S)
soils$DB.S = soils$Dstar.S/soils$Bstar.S

soils$Dstar.Bulk = log(soils$fdil.Bulk)/log(soils$ftot.Bulk)
soils$Bstar.Bulk = log(soils$fdeg.Bulk)/log(soils$ftot.Bulk)
soils$DB.Bulk = soils$Dstar.Bulk/soils$Bstar.Bulk

median(soils$DB.N, na.rm = T)
```

```
## [1] 0.470307
```

```
median(soils$DB.T, na.rm = T)
```

```
## [1] 0.1250457
```

```
median(soils$DB.S, na.rm = T)
```

```
## [1] 0.05273486
```

```
mean(soils$DB.N, na.rm = T)
```

```
## [1] 0.9746164
```

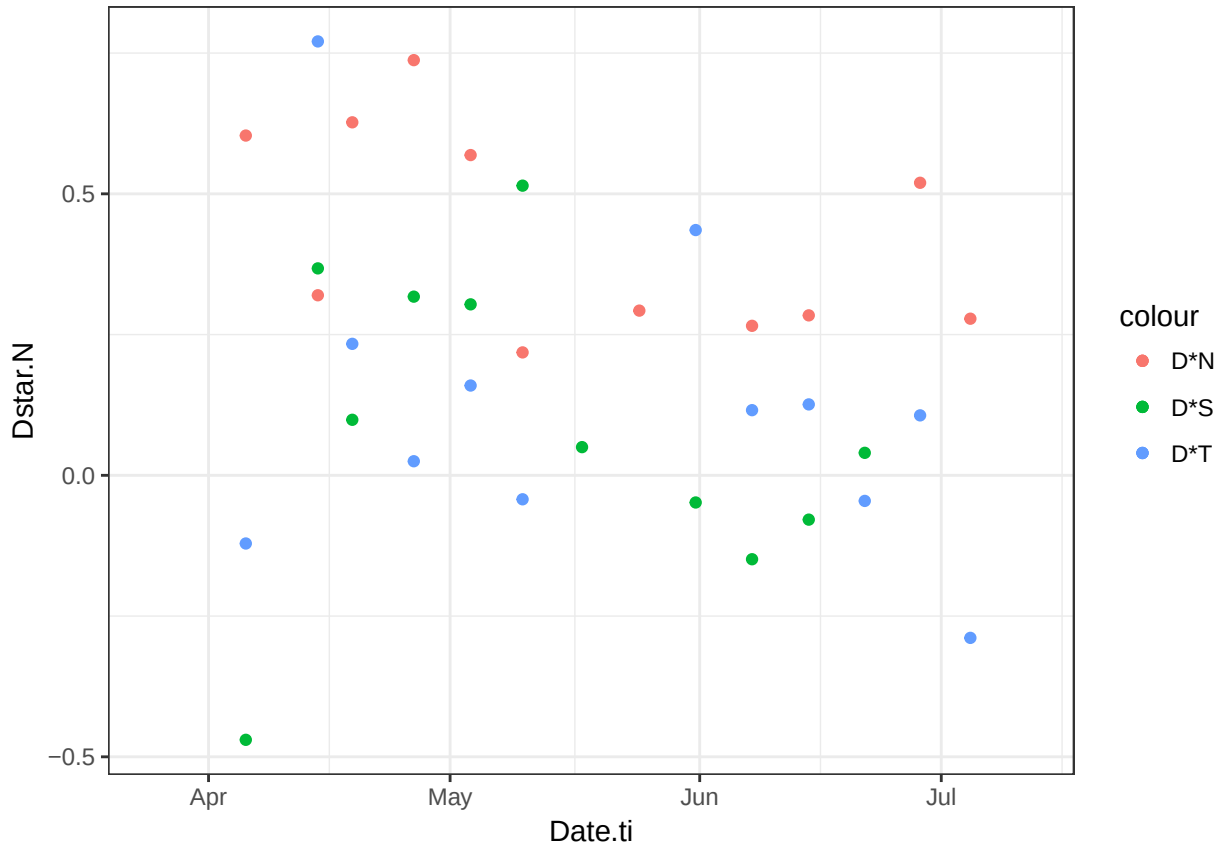
```
mean(soils$DB.T, na.rm = T)
```

```
## [1] 0.3859694
```

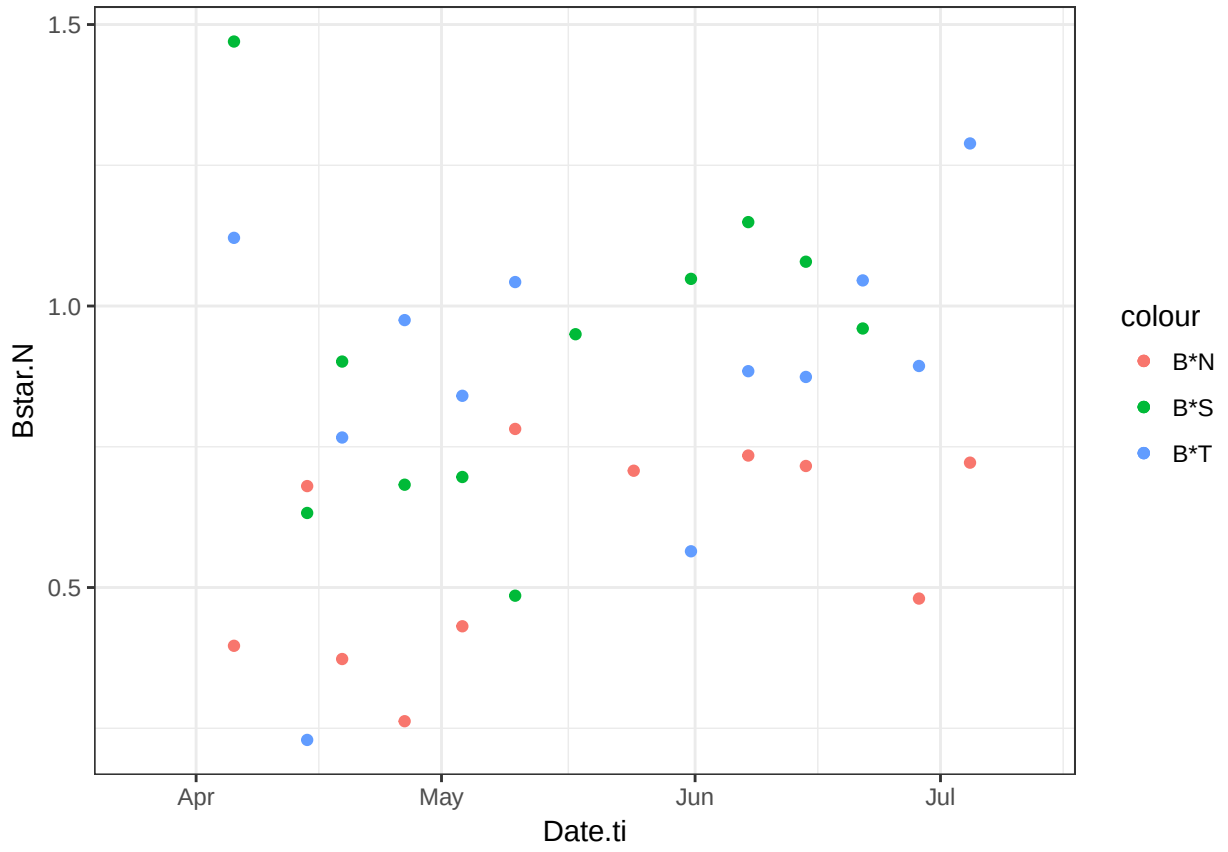
```
mean(soils$DB.S, na.rm = T)
```

```
## [1] 0.1979797
```

```
ggplot(data = soils , aes(x=Date.ti))+
  geom_point(aes(y = Dstar.N, colour = "D*N")) +
  geom_point(aes(y = Dstar.T, colour = "D*T")) +
  geom_point(aes(y = Dstar.S, colour = "D*S")) +
  theme_bw()
```




```
ggplot(data = soils , aes(x=Date.ti))+
  geom_point(aes(y = Bstar.N, colour = "B*N")) +
  geom_point(aes(y = Bstar.T, colour = "B*T")) +
  geom_point(aes(y = Bstar.S, colour = "B*S")) +
  theme_bw()
```




Calculating the cummulative mass that was degraded based on B% and converting concentrations to masses:

$$M_t\ [\mu g] = C_t\ [\mu g/g\ soil] * \rho\ [g\ soil/m^3] * A_T[m^2] * D\ [m]$$

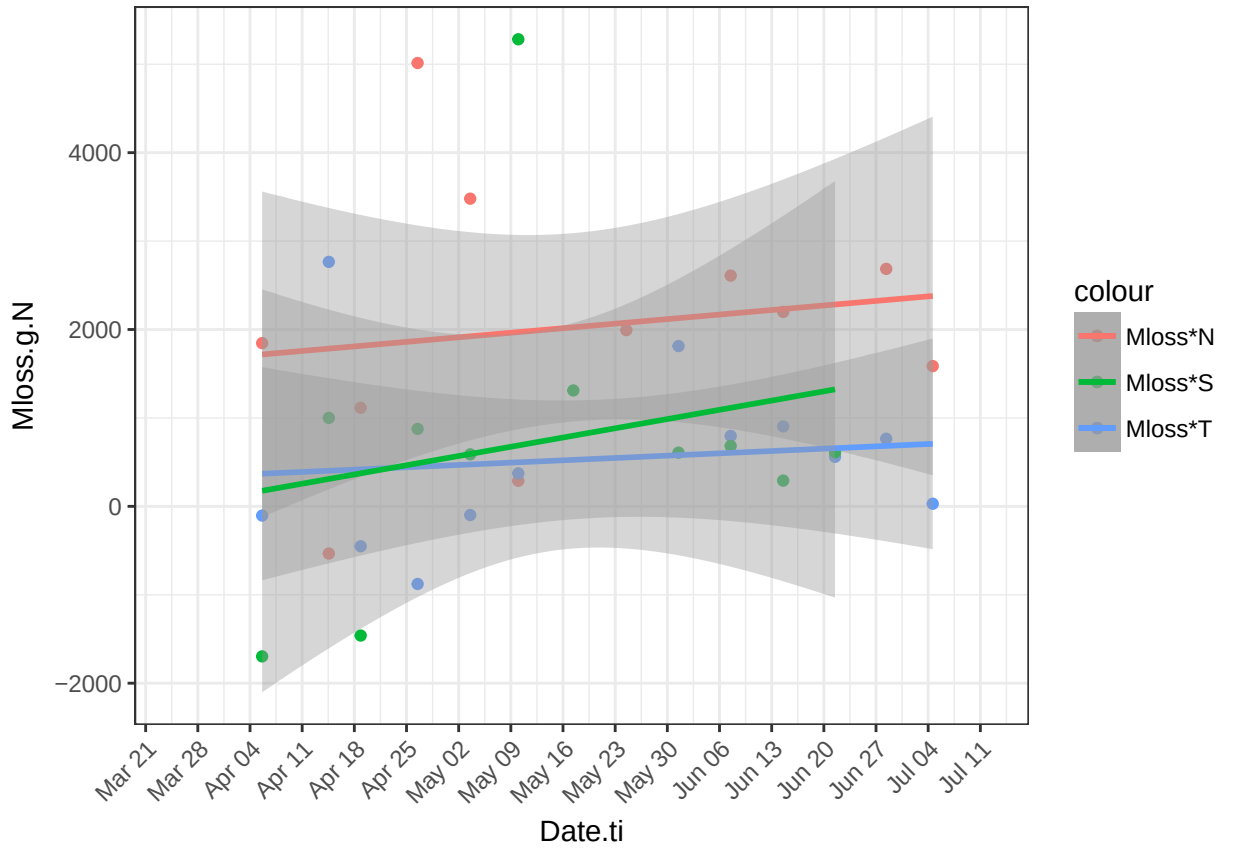
Note here that $A_T$ refers to the representative catchment area (subcatchment) of the given transect, and depth is 0.01 $m$.

```
soils$Mdeg.g.N <- soils$CumAppMass.g.N.OT*soils$Belsner.N/100
soils$Mdeg.g.T <- soils$CumAppMass.g.T.OT*soils$Belsner.T/100
soils$Mdeg.g.S <- soils$CumAppMass.g.S.OT*soils$Belsner.S/100
soils$SumDeg.g <- soils$Mdeg.g.N + soils$Mdeg.g.T + soils$Mdeg.g.S
soils$Mdeg.g <- soils$CumAppMass.g.OT*soils$Belsner.Bulk/100

#soils$Mdeg.g.N.eff <- soils$Mdeg.g.N/soils$Area.N
#soils$Mdeg.g.T.eff <- soils$Mdeg.g.T/soils$Area.T
#soils$Mdeg.g.S.eff <- soils$Mdeg.g.S/soils$Area.S

# Mass remaining needs to be calculated based on
# increasing bulk density between early and late season
# otherwise, mass balances don't add up.

colnames(soils)[colnames(soils) == "CatchMassSoil.g"] <- "Mrem.g" # Sum of N, T & S
colnames(soils)[colnames(soils) == "MassSoil.g.North"] <- "Mrem.g.N"
colnames(soils)[colnames(soils) == "MassSoil.g.Talweg"] <- "Mrem.g.T"
colnames(soils)[colnames(soils) == "MassSoil.g.South"] <- "Mrem.g.S"
```

```
soils$Mloss.g.N <- soils$CumAppMass.g.N.OT - (soils$Mdeg.g.N + soils$Mrem.g.N)
soils$Mloss.g.T <- soils$CumAppMass.g.T.OT - (soils$Mdeg.g.T + soils$Mrem.g.T)
soils$Mloss.g.S <- soils$CumAppMass.g.S.OT - (soils$Mdeg.g.S + soils$Mrem.g.S)
soils$Mloss.g <- soils$CumAppMass.g.OT - (soils$Mdeg.g + soils$Mrem.g)

#length(soils)
#names(soils)
#View(soils[, c(1, 17:20, 5:8,  12,21,26,31, 15,24,29, 96:ncol(soils)) ])
```

## Mass Losses

```
library(reshape2)
# names(soils)
s <- soils[, c(1, 5, 12,21,26, 96:ncol(soils)) ]


s$MlossCSA = (s$Mloss.g.N + s$Mloss.g.T)
s$MlossCSA.prct = ((s$Mloss.g.N + s$Mloss.g.T) / s$Mloss.g)*100

moltenSoil <- melt(s, id = "Date.ti")

# Mass Losses
ggplot(data = s , aes(x=Date.ti))+
  geom_point(aes(y = Mloss.g.N, colour = "Mloss*N")) +
  geom_point(aes(y = Mloss.g.T, colour = "Mloss*T")) +
  geom_point(aes(y = Mloss.g.S, colour = "Mloss*S")) +
  stat_smooth(method = "lm", formula = y ~ x, aes(y = Mloss.g.N, colour = "Mloss*N")) +
  stat_smooth(method = "lm", formula = y ~ x, aes(y = Mloss.g.T, colour = "Mloss*T")) +
  stat_smooth(method = "lm", formula = y ~ x, aes(y = Mloss.g.S, colour = "Mloss*S")) +
  theme_bw() +
  theme(# legend.position = "top",
        axis.text.x=element_text(angle = 45, hjust = 1)) +
  scale_x_datetime(breaks = date_breaks("1 weeks"), labels = date_format("%b %d"))
```




```
# cor.test(soils$Mloss.g.N, soils$Mloss.g.T)
```

## Critical Source Area - Losses

### April

```
subset(s[, c( "Date.ti", "CumAppMass.g.OT",
                  "MlossCSA", "MlossCSA.prct")], (Date.ti >= as.POSIXct("2016-04-18 20:30:00", tz = "ES
                & Date.ti < as.POSIXct("2016-05-01 00:00:00", tz = "EST"))
              )
```

```
##                Date.ti CumAppMass.g.OT  MlossCSA MlossCSA.prct
## 13 2016-04-18 20:30:00        36726.56  663.1892     -80.25441
## 14 2016-04-21 09:12:00        36726.56        NA            NA
## 15 2016-04-23 06:38:00        36726.56        NA            NA
## 16 2016-04-26 11:50:00        36726.56 4136.0276      94.61087
```

### June

```
subset(s[, c( "Date.ti", "CumAppMass.g.OT",
                  "MlossCSA", "MlossCSA.prct")],
              (Date.ti >= as.POSIXct("2016-06-01 00:56:00", tz = "EST")
                & Date.ti <= as.POSIXct("2016-06-28 14:52:00", tz = "EST"))
              )
```

```
##                Date.ti CumAppMass.g.OT MlossCSA MlossCSA.prct
## 34 2016-06-02 12:58:00        38529.62       NA            NA
## 35 2016-06-03 12:06:00        38529.62       NA            NA
## 36 2016-06-04 08:36:00        38529.62       NA            NA
## 37 2016-06-04 11:00:00        38529.62       NA            NA
## 38 2016-06-04 15:32:00        44837.17       NA            NA
## 39 2016-06-07 12:00:00        44837.17 3405.526      93.13291
## 40 2016-06-10 05:26:00        44837.17       NA            NA
## 41 2016-06-14 12:34:00        44837.17       NA            NA
## 42 2016-06-14 13:06:00        44837.17 3102.616      88.51522
## 43 2016-06-15 08:14:00        44837.17       NA            NA
## 44 2016-06-16 08:22:00        44837.17       NA            NA
## 45 2016-06-17 00:50:00        44837.17       NA            NA
## 46 2016-06-17 11:06:00        44837.17       NA            NA
## 47 2016-06-21 12:00:00        44837.17       NA            NA
## 48 2016-06-24 14:52:00        44837.17       NA            NA
## 49 2016-06-25 07:50:00        44837.17       NA            NA
## 50 2016-06-28 08:56:00        44837.17 3449.844            NA
```

**Summary absolute percent mass loss**

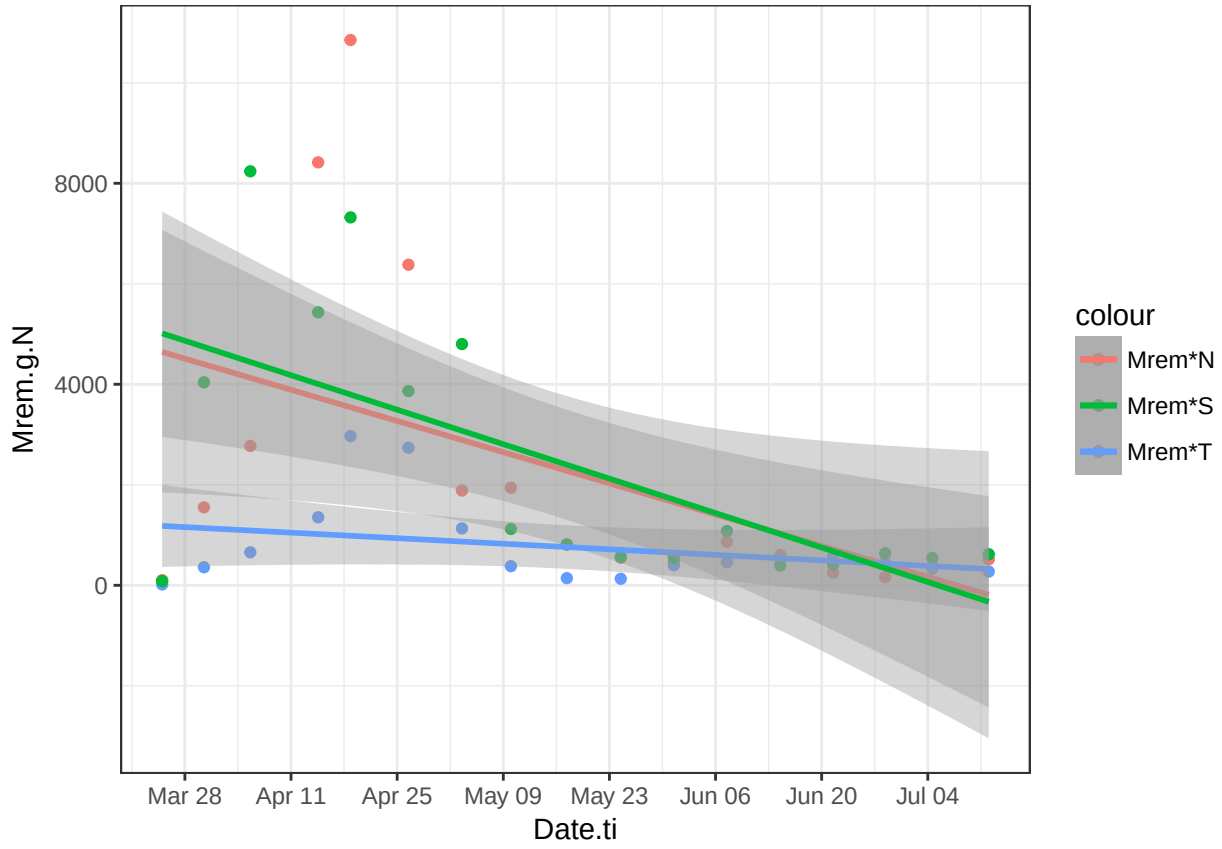
```
soilsApril <- subset(soils, (Date.ti >= as.POSIXct("2016-04-18 20:30:00", tz = "EST")
                & Date.ti < as.POSIXct("2016-05-01 00:00:00", tz = "EST")))
                    # & Date.ti <= as.POSIXct("2016-05-03 13:11:00", tz = "EST")))

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

soilsJune <- subset(soils, (Date.ti >= as.POSIXct("2016-06-01 00:56:00", tz = "EST")
                & Date.ti <= as.POSIXct("2016-06-28 14:52:00", tz = "EST")) )


library(dplyr)
sa <- soilsApril %>%
  summarise_each(funs(mean(. , na.rm=T) ))

sj <- soilsJune %>%
  summarise_each(funs(mean(. , na.rm=T) ))

saj <- rbind(sa,sj)
saj$PrctAbsLoss <- (saj$Mloss.g/soils$CumAppMass.g.OT[nrow(soils)-1])*100

saj$PrctAbsLoss[1]/saj$PrctAbsLoss[2]*100
```

```
## [1] 49.50232
```

## Mass remaining

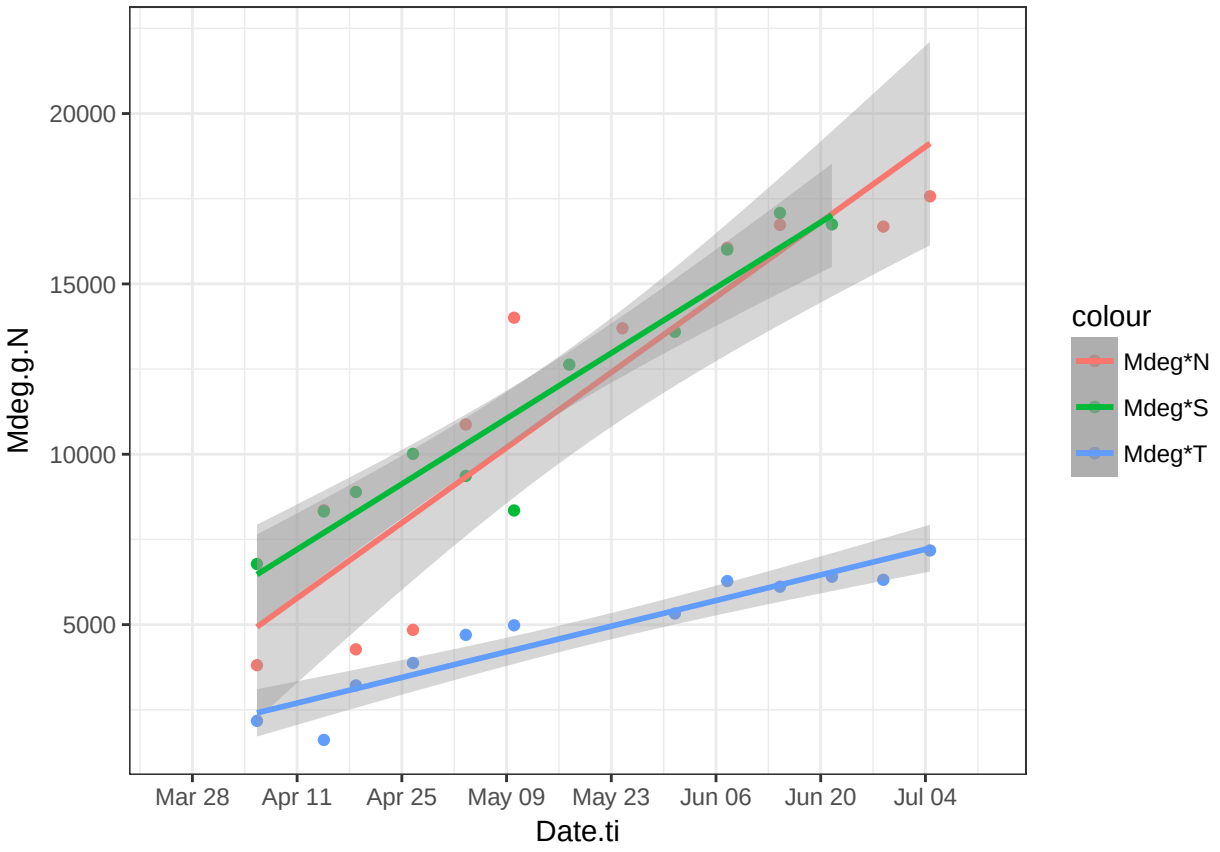
```
ggplot(data = s , aes(x=Date.ti))+
  geom_point(aes(y = Mrem.g.N, colour = "Mrem*N")) +
  geom_point(aes(y = Mrem.g.T, colour = "Mrem*T")) +
  geom_point(aes(y = Mrem.g.S, colour = "Mrem*S")) +
  stat_smooth(method = "lm", formula = y ~ x, aes(y = Mrem.g.N, colour = "Mrem*N")) +
```

```
  stat_smooth(method = "lm", formula = y ~ x, aes(y = Mrem.g.T, colour = "Mrem*T")) +
  stat_smooth(method = "lm", formula = y ~ x, aes(y = Mrem.g.S, colour = "Mrem*S")) +
  scale_x_datetime(breaks = date_breaks("2 weeks"), labels = date_format("%b %d")) +
  theme_bw()
```




## Degradation

```
ggplot(data = s , aes(x=Date.ti))+
  geom_point(aes(y = Mdeg.g.N, colour = "Mdeg*N")) +
  geom_point(aes(y = Mdeg.g.T, colour = "Mdeg*T")) +
  geom_point(aes(y = Mdeg.g.S, colour = "Mdeg*S")) +
  stat_smooth(method = "lm", formula = y ~ x, aes(y = Mdeg.g.N, colour = "Mdeg*N")) +
  stat_smooth(method = "lm", formula = y ~ x, aes(y = Mdeg.g.T, colour = "Mdeg*T")) +
  stat_smooth(method = "lm", formula = y ~ x, aes(y = Mdeg.g.S, colour = "Mdeg*S")) +
  scale_x_datetime(breaks = date_breaks("2 weeks"), labels = date_format("%b %d")) +
  theme_bw()
```




## Percentages

```
soils$PrctRem.N <- (soils$Mrem.g.N/soils$CumAppMass.g.N.OT)*100
soils$PrctRem.T <- (soils$Mrem.g.T/soils$CumAppMass.g.T.OT)*100
soils$PrctRem.S <- (soils$Mrem.g.S/(soils$CumAppMass.g.S.OT))*100

soils$PrctDeg <- (soils$SumDeg.g/soils$CumAppMass.g.OT)*100 # Sum of transect degradations relative to
soils$PrctDeg.N <- (soils$Mdeg.g.N/soils$CumAppMass.g.N.OT)*100
soils$PrctDeg.T <- (soils$Mdeg.g.T/soils$CumAppMass.g.T.OT)*100
soils$PrctDeg.S <- (soils$Mdeg.g.S/(soils$CumAppMass.g.S.OT))*100

soils$PrctRem <- (soils$Mrem.g/soils$CumAppMass.g.OT)*100
soils$PrctLoss <- 100 - (soils$PrctRem + soils$PrctDeg)
soils$PrctLoss.N <- 100 - (soils$PrctRem.N + soils$PrctDeg.N)
soils$PrctLoss.T <- 100 - (soils$PrctRem.T + soils$PrctDeg.T)
soils$PrctLoss.S <- 100 - (soils$PrctRem.S + soils$PrctDeg.S)

names(soils)
```

```
##   [1] "Date.ti"              "WeekSubWeek"
##   [3] "diss.d13C"            "SD.d13C"
##   [5] "CumAppMass.g.OT"      "CumAppMass.g.N.OT"
##   [7] "CumAppMass.g.T.OT"    "CumAppMass.g.S.OT"
##   [9] "iniCo.ug.g.N"         "iniCo.ug.g.T"
##  [11] "iniCo.ug.g.S"         "Mrem.g.N"
```

```
##  [13] "MassSoil.g.SD.North"   "Conc.mug.g.dry.soil.N"
##  [15] "comp.d13C.North"       "comp.d13C.SD.North"
##  [17] "ID.N"                  "Area.N"
##  [19] "Area.T"                "Area.S"
##  [21] "Mrem.g.T"              "MassSoil.g.SD.Talweg"
##  [23] "Conc.mug.g.dry.soil.T" "comp.d13C.Talweg"
##  [25] "comp.d13C.SD.Talweg"   "Mrem.g.S"
##  [27] "MassSoil.g.SD.South"   "Conc.mug.g.dry.soil.S"
##  [29] "comp.d13C.South"       "comp.d13C.SD.South"
##  [31] "Mrem.g"                "CatchMassSoil.g.SD"
##  [33] "BulkCatch.d13"         "BulkCatch.d13.SD"
##  [35] "BulkCatch.Conc"        "iniCo.Bulk"
##  [37] "ftot.N"                "ftot.T"
##  [39] "ftot.S"                "ftot.Bulk"
##  [41] "Delta.N"               "Delta.T"
##  [43] "Delta.S"               "Delta.Bulk"
##  [45] "Fdil.N"                "Fdil.T"
##  [47] "Fdil.S"                "Fdil.Bulk"
##  [49] "fdil.N"                "fdeg.N"
##  [51] "Dprct.N"               "Bprct.N"
##  [53] "Tprct.N"               "fdil.T"
##  [55] "fdeg.T"                "Dprct.T"
##  [57] "Bprct.T"               "Tprct.T"
##  [59] "fdil.S"                "fdeg.S"
##  [61] "Dprct.S"               "Bprct.S"
##  [63] "Tprct.S"               "fdil.Bulk"
##  [65] "fdeg.Bulk"             "Dprct.Bulk"
##  [67] "Bprct.Bulk"            "Tprct.Bulk"
##  [69] "DD.N"                  "DD.T"
##  [71] "DD.S"                  "DD.Bulk"
##  [73] "Belsner.N"             "Belsner.T"
##  [75] "Belsner.S"             "Berr.Stat.N"
##  [77] "Berr.Stat.T"           "Berr.Stat.S"
##  [79] "Belsner.Bulk"          "Belsner.BulkField"
##  [81] "Bdiff.LabField"        "Berr.Stat.Bulk"
##  [83] "Berr.Stat.Field"       "Dstar.N"
##  [85] "Bstar.N"               "DB.N"
##  [87] "Dstar.T"               "Bstar.T"
##  [89] "DB.T"                  "Dstar.S"
##  [91] "Bstar.S"               "DB.S"
##  [93] "Dstar.Bulk"            "Bstar.Bulk"
##  [95] "DB.Bulk"               "Mdeg.g.N"
##  [97] "Mdeg.g.T"              "Mdeg.g.S"
##  [99] "SumDeg.g"              "Mdeg.g"
## [101] "Mloss.g.N"             "Mloss.g.T"
## [103] "Mloss.g.S"             "Mloss.g"
## [105] "PrctRem.N"             "PrctRem.T"
## [107] "PrctRem.S"             "PrctDeg"
## [109] "PrctDeg.N"             "PrctDeg.T"
## [111] "PrctDeg.S"             "PrctRem"
## [113] "PrctLoss"              "PrctLoss.N"
## [115] "PrctLoss.T"            "PrctLoss.S"
```

**April % Remaining**

```
subset(soils[, c( "Date.ti",
                  "PrctRem.N", "PrctDeg.N", "PrctLoss.N",
                  "PrctRem.T", "PrctDeg.T", "PrctLoss.T",
                  "PrctRem.S", "PrctDeg.S", "PrctLoss.S")], (Date.ti >= as.POSIXct("2016-04-18 20:30:00
                 & Date.ti < as.POSIXct("2016-05-01 00:00:00", tz = "EST"))
              )
```

```
##                Date.ti PrctRem.N PrctDeg.N PrctLoss.N PrctRem.T PrctDeg.T
## 13 2016-04-18 20:30:00  66.82091  26.31731   6.861779  51.77759  56.09031
## 14 2016-04-21 09:12:00        NA        NA         NA        NA        NA
## 15 2016-04-23 06:38:00        NA        NA         NA        NA        NA
## 16 2016-04-26 11:50:00  39.28467  29.83949  30.875848  47.74756  67.56623
##    PrctLoss.T PrctRem.S PrctDeg.S PrctLoss.S
## 13  -7.867898  49.62667  60.27747  -9.904144
## 14         NA        NA        NA         NA
## 15         NA        NA        NA         NA
## 16 -15.313791  26.19633  67.86707   5.936597
```

**June % Remaining**

```
subset(soils[, c( "Date.ti",
                  "PrctRem.N", "PrctDeg.N", "PrctLoss.N",
                  "PrctRem.T", "PrctDeg.T", "PrctLoss.T",
                  "PrctRem.S", "PrctDeg.S", "PrctLoss.S")],
              (Date.ti >= as.POSIXct("2016-06-01 00:56:00", tz = "EST")
                 & Date.ti <= as.POSIXct("2016-06-28 14:52:00", tz = "EST"))
              )
```

```
##                Date.ti PrctRem.N PrctDeg.N PrctLoss.N PrctRem.T PrctDeg.T
## 34 2016-06-02 12:58:00        NA        NA         NA        NA        NA
## 35 2016-06-03 12:06:00        NA        NA         NA        NA        NA
## 36 2016-06-04 08:36:00        NA        NA         NA        NA        NA
## 37 2016-06-04 11:00:00        NA        NA         NA        NA        NA
## 38 2016-06-04 15:32:00        NA        NA         NA        NA        NA
## 39 2016-06-07 12:00:00 4.4262911  82.21435   13.35936  6.140617  83.29768
## 40 2016-06-10 05:26:00        NA        NA         NA        NA        NA
## 41 2016-06-14 12:34:00        NA        NA         NA        NA        NA
## 42 2016-06-14 13:06:00 3.0831557  85.65910   11.25774  6.905966  81.10468
## 43 2016-06-15 08:14:00        NA        NA         NA        NA        NA
## 44 2016-06-16 08:22:00        NA        NA         NA        NA        NA
## 45 2016-06-17 00:50:00        NA        NA         NA        NA        NA
## 46 2016-06-17 11:06:00        NA        NA         NA        NA        NA
## 47 2016-06-21 12:00:00 1.2789058        NA         NA  7.573062  84.99337
## 48 2016-06-24 14:52:00        NA        NA         NA        NA        NA
## 49 2016-06-25 07:50:00        NA        NA         NA        NA        NA
## 50 2016-06-28 08:56:00 0.8455039  85.40662   13.74787  6.061721  83.79547
##    PrctLoss.T PrctRem.S PrctDeg.S PrctLoss.S
## 34         NA        NA        NA         NA
## 35         NA        NA        NA         NA
## 36         NA        NA        NA         NA
## 37         NA        NA        NA         NA
```

```
## 38         NA        NA        NA         NA
## 39  10.561699  6.073121  90.08156   3.845318
## 40         NA        NA        NA         NA
## 41         NA        NA        NA         NA
## 42  11.989355  2.221163  96.13938   1.639459
## 43         NA        NA        NA         NA
## 44         NA        NA        NA         NA
## 45         NA        NA        NA         NA
## 46         NA        NA        NA         NA
## 47   7.433572  2.319682  94.24306   3.437255
## 48         NA        NA        NA         NA
## 49         NA        NA        NA         NA
## 50  10.142812  3.577898        NA         NA
```

### Critical Source Area contribution

$$M_{appl}$$

### Differences in degradation extent, $\epsilon_{lab}$ *vs.* $\epsilon_{field}$

```
# soils$RemPlusDeg <- soils$PrctRem + soils$PrctDeg

# names(soils)
dropS <- c("Area.Catch", "Area.N", "Area.T", "Area.S",
            "WeekSubWeek",
           # Isotopes Soils
           "comp.d13C.North", "comp.d13C.SD.North",
           "comp.d13C.Talweg", "comp.d13C.SD.Talweg"  ,
           "comp.d13C.South", "comp.d13C.SD.South",
           "BulkCatch.d13", "BulkCatch.d13.SD",
           "DD.N", "DD.T", "DD.S","DD.Bulk",
           # ISotopes Water
           # "diss.d13C", "SD.d13C",
           # Concentrations
           "BulkCatch.Conc" ,
           "Conc.mug.g.dry.soil.N", "Conc.mug.g.dry.soil.T", "Conc.mug.g.dry.soil.S",

           "iniCo.Bulk", "iniCo.ug.g.N" , "iniCo.ug.g.T",  "iniCo.ug.g.S",

           # Mass
           "CatchMassSoil.g", "CatchMassSoil.g.SD",

           # VanBreukelen
           "Fdil.N", "Fdil.T", "Fdil.S", "Fdil.Bulk",
            "ftot.N","ftot.T", "ftot.S", "ftot.Bulk",
           "fdil.N", "fdil.T", "fdil.S", "fdil.Bulk",
           "fdeg.N", "fdeg.T", "fdeg.S", "fdeg.Bulk",
           "Bstar.N", "Bstar.T", "Bstar.S", "Bstar.Bulk",
           "Dstar.N", "Dstar.T", "Dstar.S", "Dstar.Bulk",
           "dM.g.N", "dM.g.T", "dM.g.S"# ,
           #"Mdeg.g.N", "Mdeg.g.T", "Mdeg.g.S"
           )
```

```
soilsRay <- soils[ , !colnames(soils) %in% dropS]
soilsRay <- soilsRay[complete.cases(soilsRay[ , "ID.N"]), ]
soilsRay$ID.N.x <- soilsRay$ID.N
names(soilsRay)
```

```
##  [1] "Date.ti"              "diss.d13C"            "SD.d13C"
##  [4] "CumAppMass.g.OT"      "CumAppMass.g.N.OT"    "CumAppMass.g.T.OT"
##  [7] "CumAppMass.g.S.OT"    "Mrem.g.N"             "MassSoil.g.SD.North"
## [10] "ID.N"                 "Mrem.g.T"             "MassSoil.g.SD.Talweg"
## [13] "Mrem.g.S"             "MassSoil.g.SD.South"  "Mrem.g"
## [16] "Delta.N"              "Delta.T"              "Delta.S"
## [19] "Delta.Bulk"           "Dprct.N"              "Bprct.N"
## [22] "Tprct.N"              "Dprct.T"              "Bprct.T"
## [25] "Tprct.T"              "Dprct.S"              "Bprct.S"
## [28] "Tprct.S"              "Dprct.Bulk"           "Bprct.Bulk"
## [31] "Tprct.Bulk"           "Belsner.N"            "Belsner.T"
## [34] "Belsner.S"            "Berr.Stat.N"          "Berr.Stat.T"
## [37] "Berr.Stat.S"          "Belsner.Bulk"         "Belsner.BulkField"
## [40] "Bdiff.LabField"       "Berr.Stat.Bulk"       "Berr.Stat.Field"
## [43] "DB.N"                 "DB.T"                 "DB.S"
## [46] "DB.Bulk"              "Mdeg.g.N"             "Mdeg.g.T"
## [49] "Mdeg.g.S"             "SumDeg.g"             "Mdeg.g"
## [52] "Mloss.g.N"            "Mloss.g.T"            "Mloss.g.S"
## [55] "Mloss.g"              "PrctRem.N"            "PrctRem.T"
## [58] "PrctRem.S"            "PrctDeg"              "PrctDeg.N"
## [61] "PrctDeg.T"            "PrctDeg.S"            "PrctRem"
## [64] "PrctLoss"             "PrctLoss.N"           "PrctLoss.T"
## [67] "PrctLoss.S"           "ID.N.x"
```

```
mean(soilsRay$Bdiff.LabField[3:8])
```

```
## [1] 2.528147
```

```
mean(soilsRay$Bdiff.LabField[12:13])
```

```
## [1] 1.807863
```

```
mean(soilsRay$Berr.Stat.Bulk[3:8])
```

```
## [1] 23.05811
```

```
mean(soilsRay$Berr.Stat.Bulk[12:13])
```

```
## [1] 10.12707
```

```
mean(soils$Berr.Stat.Field, na.rm = T)
```

```
## [1] 20.20983
```

```
# names(soilsRay)
```

Calculating a field enrichment after correcting for via breakdown factor ($B^*$),

$$\epsilon_{field} = B^* \cdot \epsilon_{lab} = \frac{\Delta}{lnf_{total}}$$

```
soils$Efield.N <- soils$Bstar.N * epsilon_lab
soils$Efield.T <- soils$Bstar.T * epsilon_lab
soils$Efield.S <- soils$Bstar.S * epsilon_lab
soils$Efield.Bulk <- soils$Bstar.Bulk * epsilon_lab
# soils$EfieldAdj <- soils$BstarAdj * epsilon_lab

Efield.Bulk <- mean(soils$Efield.Bulk, na.rm = T)
Efield.Bulk
```

```
## [1] -1.065417
```

```
sd(soils$Efield, na.rm = T)
```

```
## [1] NA
```

```
# median(soils$EfieldAdj)
```

### Save soils and water for Bar Plots

```
# Check correct:
soils$Etrue =  soils$Efield.Bulk/soils$Bstar.Bulk
median(soils$Etrue, na.rm = T) # Should be close to -2.2, YES.
```

```
## [1] -1.476
```

```
soils$Etrue <- NULL
#write.csv2(soils,
#           'Data/Rayleigh_Soils.csv', row.names = F)

write.csv2(soilsRay,
           'Data/Rayleigh_Soils.csv', row.names = F)
```

In Van Breuklen, the degraded and diluted fractions are plotted against each other. They find that in the fringe of the plume, where more dilution occurs, also more degradation occurs, likely associated to oxidant availability and lower toxicity levels.

```
if (FALSE){
  # Van Breuklen plot this
  soils$degY = -log(soils$fdeg)
  soils$dilX = -log(soils$fdil)


  DBmodel<-lm( degY ~ dilX , data= soils, subset=(!is.na(Etrue) ))
  cof_DB <- as.numeric(coef(DBmodel)[2])
  # se_DB <- summary(DBmodel)$coef[[4]]*1000
  summary(DBmodel)

  ggplot(data = subset(soils, ( !is.na(fdil) & dilX>0 )), aes(x=dilX, y=degY)) +
    geom_point()

}
```

In contrast, in top soils, a slightly negative but not significant relationship between extent of dilution and degradation was found. This is to be expected as concentration in tops soils are lower than in aquifer systems for legacy contaminants. At lower concentrations, lack of sufficient substrate may be associated to lower bacterial communities capable of degradation.

# Waters (under draft, test only)

Conversion of initial concentration in soils to pore water, assuming all S-met mass is available, may lead to an Fdil_w.

For waters, no dilution factor can be applied. The degraded fraction of off-site transport must be equivalent to catchment soils, as degradation is negligible within an event. However, it may be of interest to obtain the degradation extent that would be computed if only outlet observations were conducted, without knowledge of dilution extent in catchment soils. As such, the fraction degraded in outlet waters will be obtained via $\epsilon_{lab}$ and with the closed system Rayleigh equation, without making use of concentration data.

```
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))

names(waters)
```

```
##   [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"           "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"              "SD.d13C"
##  [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"             "TotSMout.g.SD"
##  [75] "FracDiss"               "FracFilt"
##  [77] "MELsm.g"                "MELsm.g.SD"
##  [79] "CumOutDiss.g"           "CumOutFilt.g"
##  [81] "CumOutSmeto.g"          "CumOutMELsm.g"
##  [83] "Appl.Mass.g"            "Appl.Mass.g.OT"
##  [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"           "iniCo.ug.g.T"
##  [93] "iniCo.ug.g.S"           "timeSinceApp"
##  [95] "timeSinceApp.N"         "timeSinceApp.T"
##  [97] "timeSinceApp.S"         "Appl.Mass.g.NoSo"
##  [99] "timeSinceApp.NoSo"      "CumAppMass.g"
## [101] "CumAppMass.g.OT"        "CumAppMass.g.N"
## [103] "CumAppMass.g.T"         "CumAppMass.g.S"
## [105] "CumAppMass.g.N.OT"      "CumAppMass.g.T.OT"
## [107] "CumAppMass.g.S.OT"      "BalMassDisch.g"
## [109] "prctMassOut"            "FracDeltaOut"
## [111] "Events"                 "Weeks"
## [113] "Event"
```

```
keepWater <- c(
  "Date.ti", "WeekSubWeek", "Sampled" , "Volume.m3", "AveDischarge.m3.h",
  "Markers" , "TimeDiff", "Duration.Hrs",
  "Conc.mug.L", "Conc.SD" ,
  "OXA_mean", "OXA_SD",
  "ESA_mean", "ESA_SD",
  "diss.d13C", "SD.d13C",
  "filt.d13C", "filt.SD.d13C",
  "DD13C.diss", "DD13C.filt",
  "Appl.Mass.g",  "CumAppMass.g",
  "DissSmeto.g", "DissSmeto.g.SD",
  "DissOXA.g", "DissOXA.g.SD",
  "DissESA.g", "DissESA.g.SD",
  "FiltSmeto.g", "FiltSmeto.g.SD",
  "TotSMout.g",  "TotSMout.g.SD",
  "MELsm.g", "MELsm.g.SD",
  "CumOutDiss.g", "CumOutFilt.g",
  "CumOutSmeto.g", "CumOutMELsm.g",
  "Events"
)
waters <- waters[ , colnames(waters) %in% keepWater]
```

## Field Assumptions

Converting soil to volumetric concentrations in soils,

$$C_{soil}\ [\mu g/L_{soil}] = C_{soil}[\frac{\mu g\ _{S-met}}{g_{soil}}] \cdot \rho_{soil}[\frac{g_{soil}}{m^3}] \cdot \frac{1m^3}{10^3 L_{soil}}$$

$$C_{soil}\ [\mu g/L_{H_2O}] = \frac{C_{soil}}{\theta_{sat} + \rho_{soil} \cdot K_d}$$

```
# S-metolachlor Mass [g]
# Conc. [ug/g dry soil] * [g/10^6 ug] * density [g/m3] * depth [m]* A [m2]
# Soil bulk density: 2200 or 0.99? -> Leaching experiments: 0.98 [g/cm3]
#rho = 0.98*10^6 # soil density [g/m3]
#depth = 0.01 # [m]
theta = 0.4
# Kd = 2.397/10^6 # m3/g (aged)
Kd = 3.99/10^6 #m3/g (fresh)
```

```
# Cosed vs. Open system Rayleigh
OPEN = T

if (OPEN) {
  waters$f.diss <-
  ((10^(-3)*waters$diss.d13C + 1)/(10^(-3)*d13Co + 1))^(1000/(epsilon_lab))

  waters$f.diss.min <-
    ((10^(-3)*waters$diss.d13C + 1)/(10^(-3)*d13Co + 1))^(1000/(epsilon_max))

  waters$f.diss.max <-
    ((10^(-3)*waters$diss.d13C + 1)/(10^(-3)*d13Co + 1))^(1000/(epsilon_min))

  waters$B.diss <- (1-waters$f.diss)*100
  waters$B.diss.max <- (1-waters$f.diss.min)*100
  waters$B.diss.min <- (1-waters$f.diss.max)*100
} else if (!VERTICAL){
  ws <- merge(waters, soils, by = "Date.ti", all = T )

  # Assume Bulk soils conc. doesn't evolve close to event
  ws$BulkCatch.Conc <- na.locf(ws$BulkCatch.Conc)

  # Convert concentrations from mass to vol H20, assuming linear sorption
  ws$poolCo_w <- (ws$BulkCatch.Conc*rho/10^3)/(theta + rho*Kd)
  ws$iniCo_w <- (iniCo*rho/10^3)/(theta + rho*Kd)

  # f_tot
  # Problem here is initial Co is not applied but, available at time of discharge
  ws$ftot_w <- ws$Conc.mug.L/ws$poolCo_w

  # Van Breukelen notation
  ############################
  ############################
  # Do we need to change d13Co to initial at event or initial product ??
  ws$Delta_w <- 1000*log( (10^-3*ws$diss.d13C.x + 1)/(10^-3*d13Co+1) )

  ws$Fdil_w =
    exp( ws$Delta_w/epsilon_lab -log(ws$ftot_w) )
  median(ws$Fdil_w, na.rm = T)

  # Fdil < 1, otherwise this
  ws$Fdil_w <- ifelse(ws$Fdil_w < 1, NA, ws$Fdil_w)

  ws$fdil_w <- 1/ws$Fdil_w
  ws$fdeg_w <- ws$ftot_w * ws$Fdil_w
```

```
  ws$Dprct_w <- (1- ws$fdil_w)*100
  #ws$DprctAdj <- (1- ws$fdilAdj)*100

  ws$Bprct_w <- (1-ws$fdeg_w)*100
  #ws$BprctAdj <- (1-ws$fdegAdj)*100

  ws$Tprct_w <- (1-ws$ftot_w)*100

  ws$Dstar_w = log(ws$fdil_w)/log(ws$ftot_w)
  ws$Bstar_w = log(ws$fdeg_w)/log(ws$ftot_w)

  ws$Dstar_w  <- ifelse(ws$Dstar_w< 0, NA, ws$Dstar_w)
  ws$Bstar_W  <- ifelse(ws$Bstar_w> 1, NA, ws$Bstar_w)

  ws$DB_w = ws$Dstar_w/ws$Bstar_w
}
```

## Save Waters for Bar Plots

```
write.csv2(waters,
           'Data/Rayleigh_Waters.csv', row.names = F)
```

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