```
library(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE)
```

#Recreate WTC 4 Isotope Analsyis #Firstly set a local directory to save the HQ images too

```
setwd("C:/Users/Richard/Desktop/WTC4-TreePhys-Draft")
```

#the following explains most of the variables used

#Variable definitions: ##"chamber": chamber number ("C01", "C02", etc)

##"T_treatment": temperature treatment ("ambient", or "warmed")

##"DateTime": Date and Time combination variable, in "YYYY-MM-DD HH:MM:SS", AEST

##"FluxCO2": Chamber CO2 flux. The net exchange of CO2 between the tree and ###the chamber air (mmol/s). See Barton et al. 2010 Ag For Met for details.

##"FluxH2O": Chamber H2O flux (mol/s). See Barton et al. 2010 Ag ###For Met for details.

##"DoorCnt": The number of times the door was open during the current ###15-minute measurement period. When this value is > 0, the flux calculations ###are likely incorrect, because they do not account for all sources of CO2 ###and H2O, like a big breathing human.

##"Tair_al": air temperature (degrees C) inside each WTC as measured by the Allerton system.

##"RH_al": the relative humidity (%) inside each WTC as measured by the Allerton system. This is a shielded and aspirated RH sensor mounted in the canopy of each tree.

##"DP_al": the dew point (degrees C) inside each WTC as measured by central IRGA. Note that DP_al and RH_al are measured by separate sensors, but the values should correspond very strongly.

##"CO2centralCh": chamber CO2 concentration (ppm, also umol mol-1) as measured by the central WTC Li-7000 IRGA

##"CO2L": chamber CO2 concentration (ppm, also umol mol-1) as measured by the local IRGA mounted on each WTC

##"Taref_al": the outside air temperature (degrees C)

##"RHref_al": the outside relative humidity

##"PAR": incident photosynthetically active radiation as measured by the sensor atop each WTC (umol m-2 s-1). Note that these sensors are occasionally shaded in the morning by the nearby Euc trees and in the evening by the pecans, and chambers are differentially affected. Also this PAR sensor is at the top of the canopy, so virtually all of the leaves will be experiencing PAR that is different than this value.

##"F": the fresh air input of CO2 into the chamber. (mmol s-1)

##"v": the output of CO2 from the chamber to the atmosphere. Lost to leaks to balance the pressure given the fresh air input. (mmol s-1)

##"ICO2": the rate of CO2 injection from tank gas. (mmol s-1)

##"deltaS": the rate of CO2 accumulation within the chamber relative to the last 15-minute cycle. This is the storage flux, to account for times when the concentration of CO2 has increased or decreased. (mmol s-1)

##"CONDH2O": the rate of H2O condensation out of the chamber and measured by the tipping bucket. (mol s-1)

##"Vwat": the output of H2O from the chamber to the atmosphere. Lost to leaks to balance the pressure given the fresh air input. (mol s-1)

##"Fwat": the fresh air input of H2O into the chamber. (mol s-1)

##"deltaS": the rate of H2O accumulation within the chamber relative to the last 15-minute cycle. This is the storage flux, to account for times when the concentration of H2O has increased or decreased. (mol s-1)

#Figure 2 #(a)

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(ggplot2)
library(xts)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:xts':
##
##     first, last

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(grid)
library(cowplot)
```

```
##
## ************************************************************
```

```
## Note: As of version 1.0.0, cowplot does not change the

##   default ggplot2 theme anymore. To recover the previous

##   behavior, execute:
##   theme_set(theme_cowplot())

## ******************************************************
```

```
##
## Attaching package: 'cowplot'

## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
library(plantecophys)
library(akima)
library(doBy)
```

```
##
## Attaching package: 'doBy'

## The following object is masked from 'package:dplyr':
##
##     order_by
```

```
library(reshape2)
library(gdata)
```

```
## gdata: Unable to locate valid perl interpreter
## gdata:
## gdata: read.xls() will be unable to read Excel XLS and XLSX files
## gdata: unless the 'perl=' argument is used to specify the location of a
## gdata: valid perl intrpreter.
## gdata:
## gdata: (To avoid display of this message in the future, please ensure
## gdata: perl is installed and available on the executable search path.)

## gdata: Unable to load perl libaries needed by read.xls()
## gdata: to support 'XLX' (Excel 97-2004) files.

##

## gdata: Unable to load perl libaries needed by read.xls()
## gdata: to support 'XLSX' (Excel 2007+) files.

##
```

```
## gdata: Run the function 'installXLSXsupport()'
## gdata: to automatically download and install the perl
## gdata: libaries needed to support Excel XLS and XLSX formats.

##
## Attaching package: 'gdata'

## The following objects are masked from 'package:dplyr':
##
##     combine, first, last

## The following objects are masked from 'package:xts':
##
##     first, last

## The following object is masked from 'package:stats':
##
##     nobs

## The following object is masked from 'package:utils':
##
##     object.size

## The following object is masked from 'package:base':
##
##     startsWith
```

```
library(ggpubr)
```

```
##
## Attaching package: 'ggpubr'

## The following object is masked from 'package:cowplot':
##
##     get_legend
```

```
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:gdata':
##
##     first, last

## The following objects are masked from 'package:reshape2':
##
##     dcast, melt
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last

## The following objects are masked from 'package:xts':
##
##     first, last

## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
```

```
library(rmarkdown)
library(tinytex)
library(knitr)  # Load Packages
googledriveWTC4ISOTOPEID <- "1RNzYApcCO1D1sr_mVQgQVSQFIWNWlqVm"
df <- read.csv(sprintf("https://docs.google.com/uc?id=%s&export=download",
    googledriveWTC4ISOTOPEID))  #best to load this in andthe # it out
# df = df[complete.cases(df), ]
df["Td"] <- (dmy_hm(df$stime, quiet = TRUE, tz = "UTC"))
summary(df$Td)
```

```
##                  Min.               1st Qu.                Median
## "2016-08-28 13:06:00" "2016-09-19 21:32:45" "2016-10-09 15:02:30"
##                  Mean               3rd Qu.                  Max.
## "2016-10-09 20:52:43" "2016-10-29 17:03:15" "2016-11-22 18:33:00"
##                  NA's
##                   "6"
```

```
df$date <- as.Date(df$Td, format = "%Y-%m-%d")
summary(df$date)
```

```
##         Min.      1st Qu.       Median         Mean      3rd Qu.         Max.
## "2016-08-28" "2016-09-19" "2016-10-09" "2016-10-09" "2016-10-29" "2016-11-22"
##         NA's
##          "6"
```

```
str(df$date)
```

```
##  Date[1:23634], format: "2016-08-28" "2016-08-28" "2016-08-28" "2016-08-28" "2016-08-28" ...
```

```
df["month"] <- format(df$Td, "%m")
str(df$month)
```

```
##  chr [1:23634] "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" ...
```

```
df$hm <- format(df$Td, "%H")
summary(df$hm)
```

```
##    Length     Class      Mode
##     23634 character character
```

```
df["week"] <- strftime(df$Td, format = "%W", tz = "UTC")
str(df$week)
```

```
##  chr [1:23634] "34" "34" "34" "34" "34" "34" "34" "34" "34" "34" "34" "34" ...
```

```
## Extract flux data relevant to isotope
start <- ymd_hms("2016-08-28 12:00:00 UTC")
summary(start)
```

```
##                  Min.               1st Qu.                Median
## "2016-08-28 12:00:00" "2016-08-28 12:00:00" "2016-08-28 12:00:00"
##                  Mean               3rd Qu.                  Max.
## "2016-08-28 12:00:00" "2016-08-28 12:00:00" "2016-08-28 12:00:00"
```

```
end <- ymd_hms("2016-11-26 12:00:00")
int <- interval(start, end)
summary(int)
```

```
##             Intervals     Earliest endpoint       Latest endpoint
##                   "1" "2016-08-28 12:00:00" "2016-11-26 12:00:00"
##             Time zone
##                 "UTC"
```

```
df <- subset(df, df$Td %within% int)
head(df$Td)
```

```
## [1] "2016-08-28 13:06:00 UTC" "2016-08-28 13:11:00 UTC"
## [3] "2016-08-28 13:15:00 UTC" "2016-08-28 13:20:00 UTC"
## [5] "2016-08-28 13:24:00 UTC" "2016-08-28 13:30:00 UTC"
```

```
tail(df$Td)
```

```
## [1] "2016-11-22 18:13:00 UTC" "2016-11-22 18:17:00 UTC"
## [3] "2016-11-22 18:22:00 UTC" "2016-11-22 18:26:00 UTC"
## [5] "2016-11-22 18:31:00 UTC" "2016-11-22 18:33:00 UTC"
```

```
# Remove outliers
df <- subset(df, df$d18O.corrected <= -8)
df <- subset(df, df$d18O.corrected >= -20)
dftrt <- data.frame(df$hm, df$month, df$d18O.corrected, df$T_treatment,
    df$Td)
str(dftrt)
```

```
## 'data.frame':    22950 obs. of  5 variables:
##  $ df.hm            : chr  "13" "13" "13" "13" ...
##  $ df.month         : chr  "08" "08" "08" "08" ...
##  $ df.d18O.corrected: num  -11.1 -14.5 -12.3 -14.5 -12.1 ...
##  $ df.T_treatment   : chr  "elevated" "ambient" "elevated" "ambient" ...
##  $ df.Td            : POSIXct, format: "2016-08-28 13:06:00" "2016-08-28 13:11:00" ...
```

```
dft <- dftrt %>% group_by(df.T_treatment, df.Td) %>% summarise_at(.vars = names(.)[3:3],
    na.rm = TRUE, .funs = c(mean))
dft <- subset(dft, df.T_treatment == "atm")  #Only take atmosphere data
# Sort the data into Time Class
dft$date <- as.Date(dft$df.Td, format = "%Y-%m-%d")
dft["month"] <- format(dft$df.Td, "%m")
str(dft$month)
```

```
##  chr [1:1762] "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" ...
```

```
dft["monthday"] <- format(dft$df.Td, "%m%d")
str(dft$monthday)
```

```
##  chr [1:1762] "0828" "0828" "0828" "0828" "0828" "0828" "0828" "0828" ...
```

```
dft <- transform(dft, day = as.numeric(factor(monthday)))  #day
dft <- as.data.frame(dft)
dft$hm <- format(dft$df.Td, "%H")
dfsml <- data.frame(dft$hm, dft$day, dft$df.d180.corrected)
head(dfsml)
```

```
##   dft.hm dft.day dft.df.d180.corrected
## 1     13       1             -14.46939
## 2     14       1             -14.48504
## 3     15       1             -14.18197
## 4     16       1             -13.57420
## 5     17       1             -13.24640
## 6     18       1             -12.43640
```

```
# Prepare for interpolation
colnames(dfsml) <- c("x", "y", "z")
str(dfsml)
```

```
## 'data.frame':    1762 obs. of  3 variables:
##  $ x: chr  "13" "14" "15" "16" ...
##  $ y: num  1 1 1 1 1 1 1 1 1 1 ...
##  $ z: num  -14.5 -14.5 -14.2 -13.6 -13.2 ...
```

```
dfsml$x <- as.numeric(paste(dfsml$x))
dfsml$y <- as.numeric(paste(dfsml$y))
str(dfsml)
```

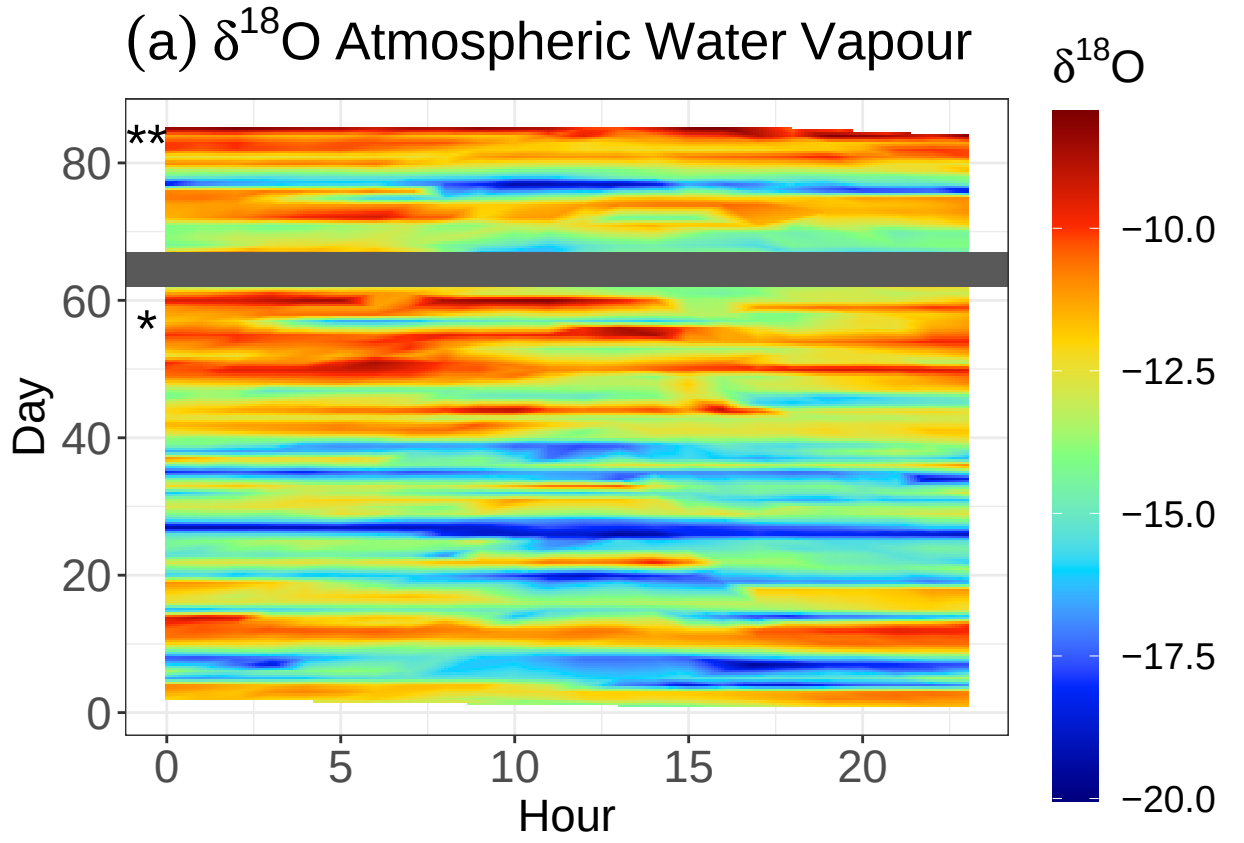
```
## 'data.frame':    1762 obs. of  3 variables:
##  $ x: num  13 14 15 16 17 18 19 20 21 22 ...
##  $ y: num  1 1 1 1 1 1 1 1 1 1 ...
##  $ z: num  -14.5 -14.5 -14.2 -13.6 -13.2 ...
```

```
x <- as.numeric(paste(dfsml$x))
y <- as.numeric(paste(dfsml$y))
z <- dfsml$z
```

```
sample = as.data.frame(cbind(x, y, z))
names(sample) <- c("d", "m", "value")
d1 <- with(sample, interp(x = d, y = m, z = value, nx = 250,
    ny = 250, duplicate = "mean", extrap = FALSE))
d2 <- melt(d1$z, na.rm = TRUE)
```

```
## Warning in melt(d1$z, na.rm = TRUE): The melt generic in data.table has
## been passed a matrix and will attempt to redirect to the relevant reshape2
## method; please note that reshape2 is deprecated, and this redirection is now
## deprecated as well. To continue using melt methods from reshape2 while both
## libraries are attached, e.g. melt.list, you can prepend the namespace like
## reshape2::melt(d1$z). In the next version, this warning will become an error.
```

```
names(d2) <- c("x", "y", "z")
d2$d <- d1$x[d2$x]
d2$m <- d1$y[d2$y]
# Plot
jet.colors <- colorRampPalette(c("#00007F", "blue", "#007FFF",
    "cyan", "#7FFF7F", "yellow", "#FF7F00", "red", "#7F0000"))

a <- ggplot(data = d2, aes(x = d, y = m, fill = z, z = z)) +
    geom_tile() + scale_fill_gradientn(colours = jet.colors(7),
    limits = c(-20, -8)) + labs(x = "Hour", y = "Day") + labs(fill = bquote(delta^18 *
    O)) + ggtitle((a) ~ delta^18 * O ~ Atmospheric ~ Water ~
    Vapour) + theme_bw() + theme(text = element_text(size = 17)) +
    theme(axis.text = element_text(size = 17)) + theme(legend.position = "right") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
    scale_x_continuous(breaks = scales::pretty_breaks(n = 5)) +
    annotate("rect", xmin = -Inf, xmax = Inf, ymin = 62, ymax = 67,
        alpha = 1) + theme(plot.title = element_text(size = 13.5,
    face = "bold")) + annotation_custom(textGrob("*", gp = gpar(col = "black",
    fontsize = 20)), xmin = -0.15, xmax = -1, ymin = 55.5, ymax = 56.5) +
    annotation_custom(textGrob("**", gp = gpar(col = "black",
        fontsize = 20)), xmin = -0.15, xmax = -1, ymin = 82.5,
        ymax = 83.5) + guides(fill = guide_colourbar(barwdith = 0.5,
    barheight = 18, nbin = 100)) + theme(plot.title = element_text(size = 20,
    face = "bold"))
a
```



#(b) The Code is almost identical just changing treatment

```
dftrt <- data.frame(df$hm, df$month, df$d18O.corrected, df$T_treatment,
    df$Td)
str(dftrt)
```

```
## 'data.frame':    22950 obs. of  5 variables:
##  $ df.hm            : chr  "13" "13" "13" "13" ...
##  $ df.month         : chr  "08" "08" "08" "08" ...
##  $ df.d18O.corrected: num  -11.1 -14.5 -12.3 -14.5 -12.1 ...
##  $ df.T_treatment   : chr  "elevated" "ambient" "elevated" "ambient" ...
##  $ df.Td            : POSIXct, format: "2016-08-28 13:06:00" "2016-08-28 13:11:00" ...
```

```
dft <- dftrt %>% group_by(df.T_treatment, df.Td) %>% summarise_at(.vars = names(.)[3:3],
    na.rm = TRUE, .funs = c(mean))
dft <- subset(dft, df.T_treatment == "ambient")  # HERE IS THE CHANGE TO AMBIENT DATA
dft$date <- as.Date(dft$df.Td, format = "%Y-%m-%d")
dft["month"] <- format(dft$df.Td, "%m")
str(dft$month)
```

```
##  chr [1:10634] "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" ...
```

```
summary(dft$month)
```

```
##    Length     Class      Mode
##     10634 character character
```

```
dft["monthday"] <- format(dft$df.Td, "%m%d")
str(dft$monthday)
```

```
##  chr [1:10634] "0828" "0828" "0828" "0828" "0828" "0828" "0828" "0828" ...
```

```
summary(dft$monthday)
```

```
##    Length     Class      Mode
##     10634 character character
```

```
dft <- transform(dft, day = as.numeric(factor(monthday)))  #day
dft <- as.data.frame(dft)
dft$hm <- format(dft$df.Td, "%H")
summary(dft$hm)
```

```
##    Length     Class      Mode
##     10634 character character
```

```
dfsml <- data.frame(dft$hm, dft$day, dft$df.d180.corrected)
colnames(dfsml) <- c("x", "y", "z")
str(dfsml)
```

```
## 'data.frame':    10634 obs. of  3 variables:
##  $ x: chr  "13" "13" "13" "13" ...
##  $ y: num  1 1 1 1 1 1 1 1 1 1 ...
##  $ z: num  -14.5 -14.5 -13.1 -13.5 -13.7 ...
```

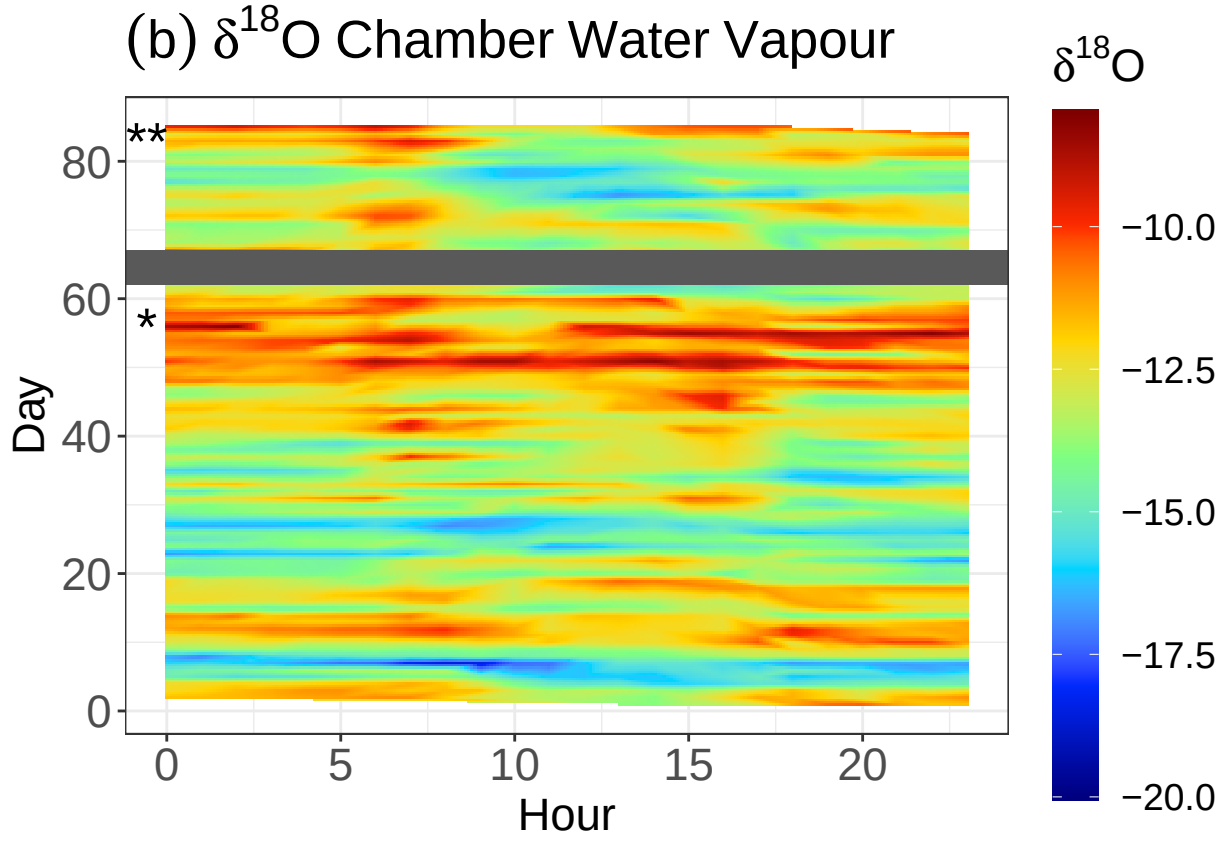
```
dfsml$x <- as.numeric(paste(dfsml$x))
dfsml$y <- as.numeric(paste(dfsml$y))
str(dfsml)
```

```
## 'data.frame':    10634 obs. of  3 variables:
##  $ x: num  13 13 13 13 13 14 14 14 14 14 ...
##  $ y: num  1 1 1 1 1 1 1 1 1 1 ...
##  $ z: num  -14.5 -14.5 -13.1 -13.5 -13.7 ...
```

```
x <- as.numeric(paste(dfsml$x))
y <- as.numeric(paste(dfsml$y))
z <- dfsml$z
sample = as.data.frame(cbind(x, y, z))
names(sample) <- c("d", "m", "value")
d1 <- with(sample, interp(x = d, y = m, z = value, nx = 250,
    ny = 250, duplicate = "mean", extrap = FALSE))
d2 <- melt(d1$z, na.rm = TRUE)
```

```
## Warning in melt(d1$z, na.rm = TRUE): The melt generic in data.table has
## been passed a matrix and will attempt to redirect to the relevant reshape2
## method; please note that reshape2 is deprecated, and this redirection is now
## deprecated as well. To continue using melt methods from reshape2 while both
## libraries are attached, e.g. melt.list, you can prepend the namespace like
## reshape2::melt(d1$z). In the next version, this warning will become an error.
```

```
names(d2) <- c("x", "y", "z")
d2$d <- d1$x[d2$x]
d2$m <- d1$y[d2$y]
jet.colors <- colorRampPalette(c("#00007F", "blue", "#007FFF",
    "cyan", "#7FFF7F", "yellow", "#FF7F00", "red", "#7F0000"))
b <- ggplot(data = d2, aes(x = d, y = m, fill = z, z = z)) +
    geom_tile() + scale_fill_gradientn(colours = jet.colors(7),
    limits = c(-20, -8)) + labs(x = "Hour", y = "Day") + labs(fill = bquote(delta^18 *
    O)) + ggtitle((b) ~ delta^18 * O ~ Chamber ~ Water ~ Vapour) +
    theme_bw() + theme(text = element_text(size = 17)) + theme(axis.text = element_text(size = 17)) +
    theme(legend.position = "right") + scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
    scale_x_continuous(breaks = scales::pretty_breaks(n = 5)) +
    annotate("rect", xmin = -Inf, xmax = Inf, ymin = 62, ymax = 67,
        alpha = 1) + theme(plot.title = element_text(size = 13.5,
    face = "bold")) + annotation_custom(textGrob("*", gp = gpar(col = "black",
    fontsize = 20)), xmin = -0.15, xmax = -1, ymin = 55.5, ymax = 56.5) +
    annotation_custom(textGrob("**", gp = gpar(col = "black",
        fontsize = 20)), xmin = -0.15, xmax = -1, ymin = 82.5,
        ymax = 83.5) + guides(fill = guide_colourbar(barwdith = 0.5,
    barheight = 18, nbin = 100)) + theme(plot.title = element_text(size = 20,
    face = "bold"))
b
```



#(c) is the same again but we use the elevated data

```
dftrt <- data.frame(df$hm, df$month, df$d18O.corrected, df$T_treatment,
    df$Td)
str(dftrt)
```

```
## 'data.frame':    22950 obs. of  5 variables:
##  $ df.hm            : chr  "13" "13" "13" "13" ...
##  $ df.month         : chr  "08" "08" "08" "08" ...
##  $ df.d18O.corrected: num  -11.1 -14.5 -12.3 -14.5 -12.1 ...
##  $ df.T_treatment   : chr  "elevated" "ambient" "elevated" "ambient" ...
##  $ df.Td            : POSIXct, format: "2016-08-28 13:06:00" "2016-08-28 13:11:00" ...
```

```
dft <- dftrt %>% group_by(df.T_treatment, df.Td) %>% summarise_at(.vars = names(.)[3:3],
    na.rm = TRUE, .funs = c(mean))
dft <- subset(dft, df.T_treatment == "elevated")  #Here we change to eleveted data
dft$date <- as.Date(dft$df.Td, format = "%Y-%m-%d")
dft["month"] <- format(dft$df.Td, "%m")
str(dft$month)
```

```
##  chr [1:10536] "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" ...
```

```
summary(dft$month)
```

```
##    Length     Class      Mode
##     10536 character character
```

```
dft["monthday"] <- format(dft$df.Td, "%m%d")
str(dft$monthday)
```

```
##  chr [1:10536] "0828" "0828" "0828" "0828" "0828" "0828" "0828" "0828" ...
```

```
summary(dft$monthday)
```

```
##    Length     Class      Mode
##     10536 character character
```

```
dft <- transform(dft, day = as.numeric(factor(monthday)))  #day
dft <- as.data.frame(dft)
dft$hm <- format(dft$df.Td, "%H")
summary(dft$hm)
```

```
##    Length     Class      Mode
##     10536 character character
```

```
dfsml <- data.frame(dft$hm, dft$day, dft$df.d18O.corrected)
colnames(dfsml) <- c("x", "y", "z")
str(dfsml)
```

```
## 'data.frame':    10536 obs. of  3 variables:
##  $ x: chr  "13" "13" "13" "13" ...
##  $ y: num  1 1 1 1 1 1 1 1 1 1 ...
##  $ z: num  -11.1 -12.3 -12.1 -11.9 -13.2 ...
```

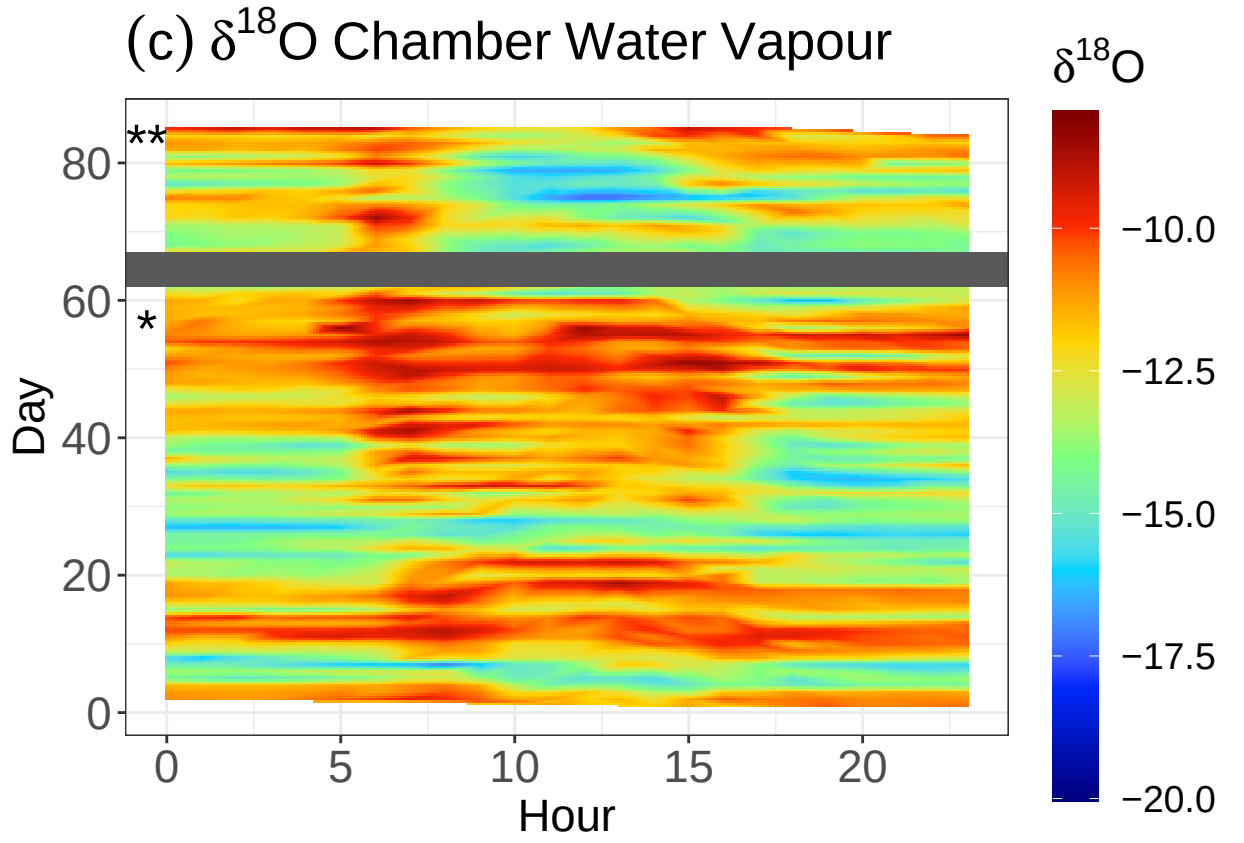
```
dfsml$x <- as.numeric(paste(dfsml$x))
dfsml$y <- as.numeric(paste(dfsml$y))
str(dfsml)
```

```
## 'data.frame':    10536 obs. of  3 variables:
##  $ x: num  13 13 13 13 13 13 14 14 14 14 ...
##  $ y: num  1 1 1 1 1 1 1 1 1 1 ...
##  $ z: num  -11.1 -12.3 -12.1 -11.9 -13.2 ...
```

```
x <- as.numeric(paste(dfsml$x))
y <- as.numeric(paste(dfsml$y))
z <- dfsml$z
sample = as.data.frame(cbind(x, y, z))
names(sample) <- c("d", "m", "value")
d1 <- with(sample, interp(x = d, y = m, z = value, nx = 250,
    ny = 250, duplicate = "mean", extrap = FALSE))
d2 <- melt(d1$z, na.rm = TRUE)
```

```
## Warning in melt(d1$z, na.rm = TRUE): The melt generic in data.table has
## been passed a matrix and will attempt to redirect to the relevant reshape2
## method; please note that reshape2 is deprecated, and this redirection is now
## deprecated as well. To continue using melt methods from reshape2 while both
## libraries are attached, e.g. melt.list, you can prepend the namespace like
## reshape2::melt(d1$z). In the next version, this warning will become an error.
```

```
names(d2) <- c("x", "y", "z")
d2$d <- d1$x[d2$x]
d2$m <- d1$y[d2$y]
jet.colors <- colorRampPalette(c("#00007F", "blue", "#007FFF",
    "cyan", "#7FFF7F", "yellow", "#FF7F00", "red", "#7F0000"))
c <- ggplot(data = d2, aes(x = d, y = m, fill = z, z = z)) +
    geom_tile() + scale_fill_gradientn(colours = jet.colors(7),
    limits = c(-20, -8)) + labs(x = "Hour", y = "Day") + labs(fill = bquote(delta^18 *
    O)) + ggtitle((c) ~ delta^18 * O ~ Chamber ~ Water ~ Vapour) +
    theme_bw() + theme(text = element_text(size = 17)) + theme(axis.text = element_text(size = 17)) +
    theme(legend.position = "right") + scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
    scale_x_continuous(breaks = scales::pretty_breaks(n = 5)) +
    annotate("rect", xmin = -Inf, xmax = Inf, ymin = 62, ymax = 67,
        alpha = 1) + theme(plot.title = element_text(size = 13.5,
    face = "bold")) + annotation_custom(textGrob("*", gp = gpar(col = "black",
    fontsize = 20)), xmin = -0.15, xmax = -1, ymin = 55.5, ymax = 56.5) +
    annotation_custom(textGrob("**", gp = gpar(col = "black",
        fontsize = 20)), xmin = -0.15, xmax = -1, ymin = 82.5,
        ymax = 83.5) + guides(fill = guide_colourbar(barwdith = 0.5,
    barheight = 18, nbin = 100)) + theme(plot.title = element_text(size = 20,
    face = "bold"))
c
```



#(d) same concept as d18O but we change to temperture #Here I use a different dataframe (I later merge parts of this data frame with wind and rain data to run multiple regression) #This data set is massive ~100mb and has all the WTC4 data (not just when the isotope laser was running)

#we later need a,b,c these are ggplot objects that form the stacked graph, but we can clear everything else

```
keep(a, b, c, sure = TRUE)
```

```
googledriveWTC4FLUXDATAFORHEATMAP <- "138BezZPxsWQ-LIqYP7szE36JeYAY0V-q"
df <- read.csv(sprintf("https://docs.google.com/uc?id=%s&export=download",
    googledriveWTC4FLUXDATAFORHEATMAP))  #best to load this in andthe # it out
df$esat <- esat(TdegC = df$Tair_al, Pa = 101)
df$vpd <- RHtoVPD(RH = df$RH_al, TdegC = df$Tair_al, Pa = 101)
summary(df$vpd)  #the vapour pr
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
## -5.4741  0.2082  0.4099  0.6388  0.8837  6.2394    1857
```

```
df["Td"] <- (ymd_hms(df$DateTime, quiet = TRUE, tz = "UTC"))
summary(df$Td)
```

```
##                  Min.               1st Qu.                Median
## "2016-02-28 00:01:00" "2016-05-06 15:32:00" "2016-07-13 08:04:00"
##                  Mean               3rd Qu.                  Max.
## "2016-07-12 09:20:22" "2016-09-18 10:07:00" "2016-11-23 23:59:00"
```

```
df$date <- as.Date(df$Td, format = "%Y-%m-%d")
summary(df$date)
```

```
##         Min.      1st Qu.       Median         Mean      3rd Qu.         Max.
## "2016-02-28" "2016-05-06" "2016-07-13" "2016-07-11" "2016-09-18" "2016-11-23"
```

```
str(df$date)
```

```
##  Date[1:302313], format: "2016-02-28" "2016-02-28" "2016-02-28" "2016-02-28" "2016-02-28" ...
```

```
df["month"] <- format(df$Td, "%m")
str(df$month)
```

```
##  chr [1:302313] "02" "02" "02" "02" "02" "02" "02" "02" "02" "02" "02" "02" ...
```

```
df$hm <- format(df$Td, "%H")
summary(df$hm)
```

```
##    Length     Class      Mode
##    302313 character character
```

```
df["week"] <- strftime(df$Td, format = "%W", tz = "UTC")
str(df$week)
```

```
##  chr [1:302313] "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" ...
```

```
df <- distinct(df)
start <- ymd_hms("2016-08-28 12:00:00 UTC")
summary(start)
```

```
##                  Min.               1st Qu.                Median
## "2016-08-28 12:00:00" "2016-08-28 12:00:00" "2016-08-28 12:00:00"
##                  Mean               3rd Qu.                  Max.
## "2016-08-28 12:00:00" "2016-08-28 12:00:00" "2016-08-28 12:00:00"
```

```
end <- ymd_hms("2016-11-26 12:00:00")
int <- interval(start, end)
summary(int)
```

```
##             Intervals     Earliest endpoint       Latest endpoint
##                   "1" "2016-08-28 12:00:00" "2016-11-26 12:00:00"
##             Time zone
##                 "UTC"
```

```
df <- subset(df, df$Td %within% int)
head(df$Td)
```

```
## [1] "2016-08-28 12:02:00 UTC" "2016-08-28 12:17:00 UTC"
## [3] "2016-08-28 12:32:00 UTC" "2016-08-28 12:47:00 UTC"
## [5] "2016-08-28 13:02:00 UTC" "2016-08-28 13:17:00 UTC"
```

```
tail(df$Td)
```

```
## [1] "2016-11-23 22:31:00 UTC" "2016-11-23 22:46:00 UTC"
## [3] "2016-11-23 23:01:00 UTC" "2016-11-23 23:16:00 UTC"
## [5] "2016-11-23 23:31:00 UTC" "2016-11-23 23:46:00 UTC"
```

```
# not that here we use Taref_al this is the reference
# (outside//ambient) air
dftrt <- data.frame(df$hm, df$month, df$Taref_al, df$T_treatment,
    df$Td)
str(dftrt)
```

```
## 'data.frame':    98502 obs. of  5 variables:
##  $ df.hm         : chr  "12" "12" "12" "12" ...
##  $ df.month      : chr  "08" "08" "08" "08" ...
##  $ df.Taref_al   : num  17.8 18.2 18.1 18.9 19.2 ...
##  $ df.T_treatment: chr  "ambient" "ambient" "ambient" "ambient" ...
##  $ df.Td         : POSIXct, format: "2016-08-28 12:02:00" "2016-08-28 12:17:00" ...
```

```
dft <- dftrt %>% group_by(df.T_treatment, df.Td) %>% summarise_at(.vars = names(.)[3:3],
    na.rm = TRUE, .funs = c(mean))
dft <- subset(dft, df.T_treatment == "ambient")
dft$date <- as.Date(dft$df.Td, format = "%Y-%m-%d")
dft["month"] <- format(dft$df.Td, "%m")
str(dft$month)
```

```
##  chr [1:49202] "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" ...
```

```
summary(dft$month)
```

```
##    Length     Class      Mode
##     49202 character character
```

```
dft["monthday"] <- format(dft$df.Td, "%m%d")
str(dft$monthday)
```

```
##  chr [1:49202] "0828" "0828" "0828" "0828" "0828" "0828" "0828" "0828" ...
```

```
summary(dft$monthday)
```

```
##    Length     Class      Mode
##     49202 character character
```

```
dft <- transform(dft, day = as.numeric(factor(monthday)))  #day
dft <- as.data.frame(dft)
dft$hm <- format(dft$df.Td, "%H")
summary(dft$hm)
```

```
##    Length     Class      Mode
##     49202 character character
```

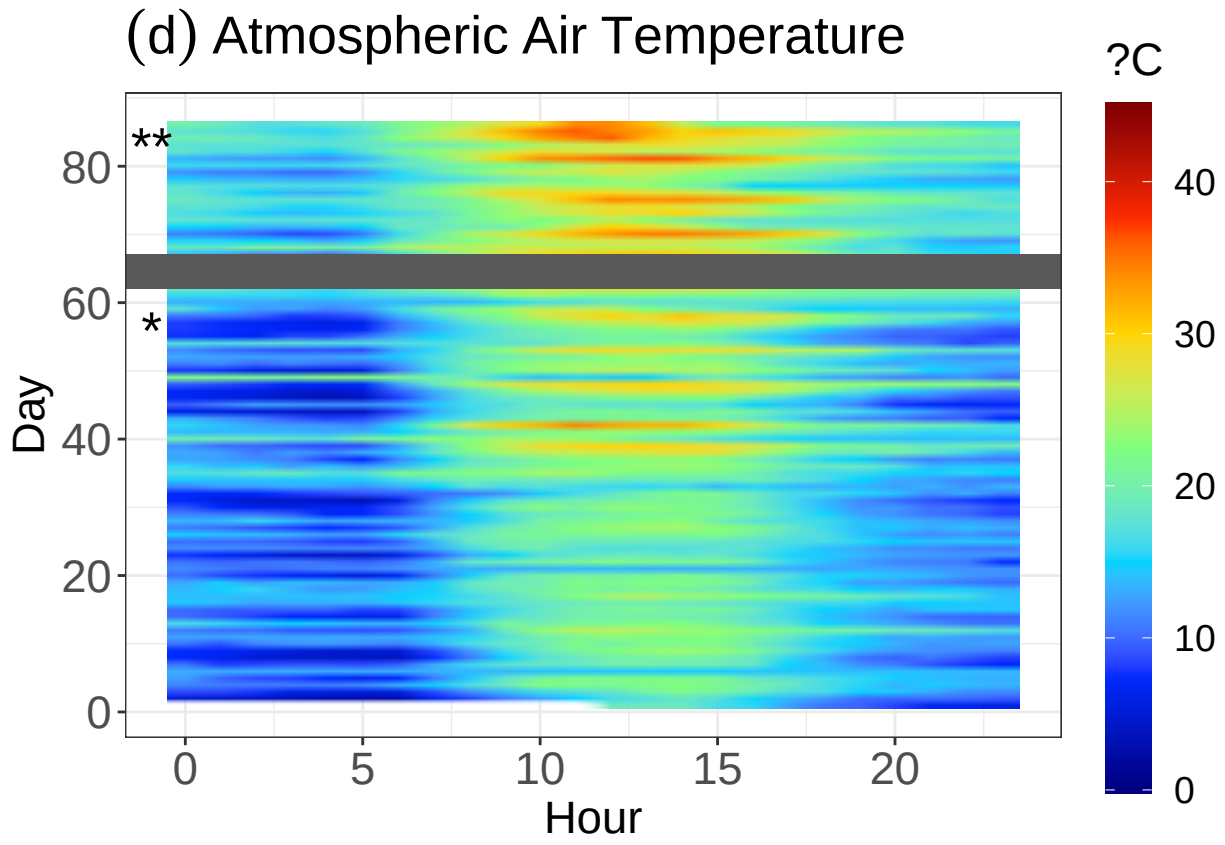
```
dfsml <- data.frame(dft$hm, dft$day, dft$df.Taref_al)
colnames(dfsml) <- c("x", "y", "z")
str(dfsml)
```

```
## 'data.frame':    49202 obs. of  3 variables:
##  $ x: chr  "12" "12" "12" "12" ...
##  $ y: num  1 1 1 1 1 1 1 1 1 1 ...
##  $ z: num  17.8 17.9 18.2 18.3 18.3 ...
```

```
dfsml$x <- as.numeric(paste(dfsml$x))
dfsml$y <- as.numeric(paste(dfsml$y))
str(dfsml)
```

```
## 'data.frame':    49202 obs. of  3 variables:
##  $ x: num  12 12 12 12 12 12 12 12 12 12 ...
##  $ y: num  1 1 1 1 1 1 1 1 1 1 ...
##  $ z: num  17.8 17.9 18.2 18.3 18.3 ...
```

```
x <- as.numeric(paste(dfsml$x))
y <- as.numeric(paste(dfsml$y))
z <- dfsml$z
gdf <- as.data.frame(cbind(x, y, z))
jet.colors <- colorRampPalette(c("#00007F", "blue", "#007FFF",
    "cyan", "#7FFF7F", "yellow", "#FF7F00", "red", "#7F0000"))
d <- ggplot(gdf, aes(x, y, fill = z)) + geom_raster(interpolate = TRUE) +
    scale_fill_gradientn(colours = jet.colors(7), limits = c(0,
        45)) + labs(x = "Hour", y = "Day") + labs(fill = ("?C")) +
    ggtitle((d) ~ Atmospheric ~ Air ~ Temperature) + theme_bw() +
    theme(text = element_text(size = 17)) + theme(axis.text = element_text(size = 17)) +
    theme(legend.position = "right") + scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
    scale_x_continuous(breaks = scales::pretty_breaks(n = 5)) +
    annotate("rect", xmin = -Inf, xmax = Inf, ymin = 62, ymax = 67,
        alpha = 1) + theme(plot.title = element_text(size = 13.5,
    face = "bold")) + annotation_custom(textGrob("*", gp = gpar(col = "black",
    fontsize = 20)), xmin = -1.1, xmax = -0.8, ymin = 55.5, ymax = 56.5) +
    annotation_custom(textGrob("**", gp = gpar(col = "black",
        fontsize = 20)), xmin = -1.1, xmax = -0.8, ymin = 82.5,
        ymax = 83.5) + guides(fill = guide_colourbar(barwdith = 0.5,
    barheight = 18, nbin = 100)) + theme(plot.title = element_text(size = 20,
    face = "bold"))
d
```



#(e) is the same but we change from reference air temp to chamber air temp

```
# Note that we use Tair_al now -- this is in chamber air temp
dftrt <- data.frame(df$hm, df$month, df$Tair_al, df$T_treatment,
    df$Td)
str(dftrt)
```

```
## 'data.frame':    98502 obs. of  5 variables:
##  $ df.hm         : chr  "12" "12" "12" "12" ...
##  $ df.month      : chr  "08" "08" "08" "08" ...
##  $ df.Tair_al    : num  17.9 18.4 17.9 18.9 19.1 ...
##  $ df.T_treatment: chr  "ambient" "ambient" "ambient" "ambient" ...
##  $ df.Td         : POSIXct, format: "2016-08-28 12:02:00" "2016-08-28 12:17:00" ...
```

```
dft <- dftrt %>% group_by(df.T_treatment, df.Td) %>% summarise_at(.vars = names(.)[3:3],
    na.rm = TRUE, .funs = c(mean))
dft <- subset(dft, df.T_treatment == "ambient")
dft$date <- as.Date(dft$df.Td, format = "%Y-%m-%d")
dft["month"] <- format(dft$df.Td, "%m")
str(dft$month)
```

```
##  chr [1:49202] "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" ...
```

```
summary(dft$month)
```

```
##    Length     Class      Mode
##     49202 character character
```

```
dft["monthday"] <- format(dft$df.Td, "%m%d")
str(dft$monthday)
```

```
##  chr [1:49202] "0828" "0828" "0828" "0828" "0828" "0828" "0828" "0828" ...
```

```
summary(dft$monthday)
```

```
##    Length     Class      Mode
##     49202 character character
```

```
dft <- transform(dft, day = as.numeric(factor(monthday)))  #day
dft <- as.data.frame(dft)
dft$hm <- format(dft$df.Td, "%H")
summary(dft$hm)
```

```
##    Length     Class      Mode
##     49202 character character
```

```
dfsml <- data.frame(dft$hm, dft$day, dft$df.Tair_al)  #Temp ref
colnames(dfsml) <- c("x", "y", "z")
str(dfsml)
```

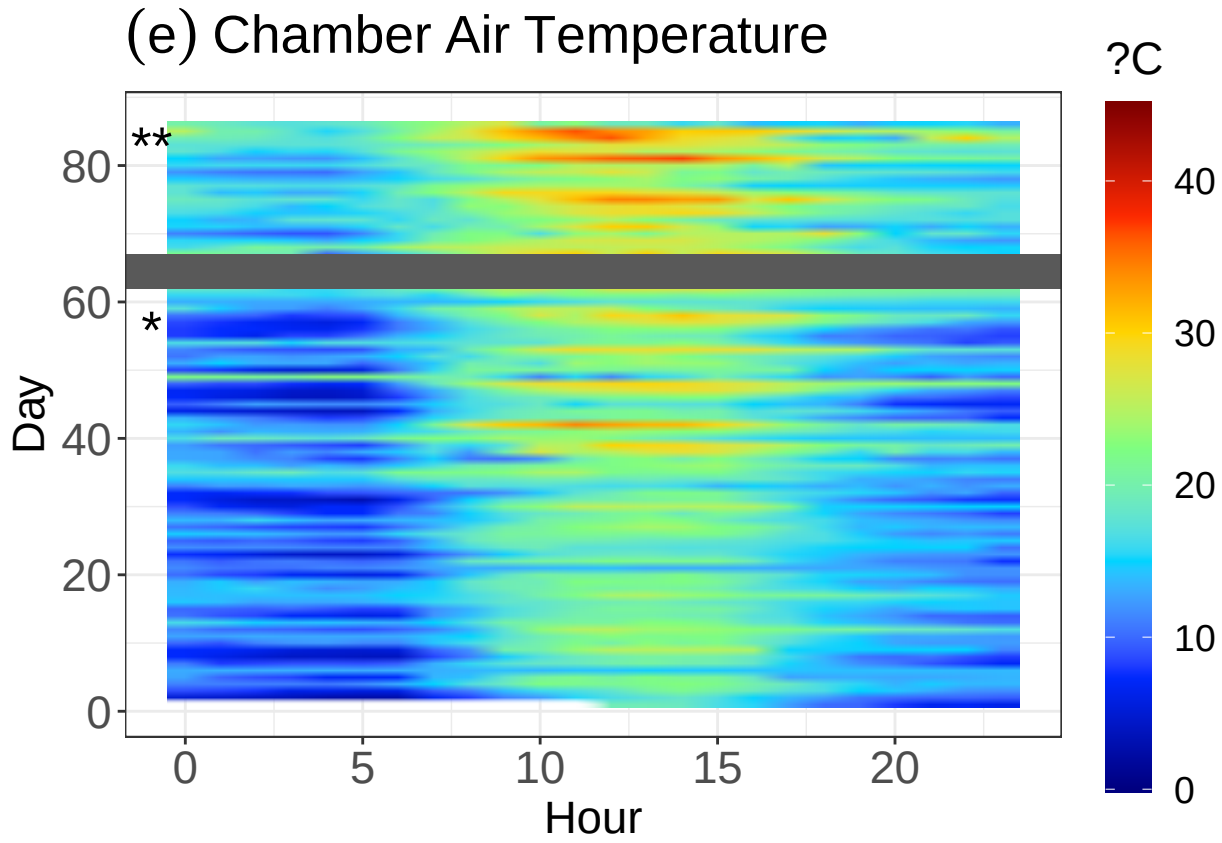
```
## 'data.frame':    49202 obs. of  3 variables:
##  $ x: chr  "12" "12" "12" "12" ...
##  $ y: num  1 1 1 1 1 1 1 1 1 1 ...
##  $ z: num  17.9 17.8 17.8 18 18.4 ...
```

```
dfsml$x <- as.numeric(paste(dfsml$x))
dfsml$y <- as.numeric(paste(dfsml$y))
str(dfsml)
```

```
## 'data.frame':    49202 obs. of  3 variables:
##  $ x: num  12 12 12 12 12 12 12 12 12 12 ...
##  $ y: num  1 1 1 1 1 1 1 1 1 1 ...
##  $ z: num  17.9 17.8 17.8 18 18.4 ...
```

```
x <- as.numeric(paste(dfsml$x))
y <- as.numeric(paste(dfsml$y))
z <- dfsml$z
gdf <- as.data.frame(cbind(x, y, z))
jet.colors <- colorRampPalette(c("#00007F", "blue", "#007FFF",
    "cyan", "#7FFF7F", "yellow", "#FF7F00", "red", "#7F0000"))
e <- ggplot(gdf, aes(x, y, fill = z)) + geom_raster(interpolate = TRUE) +
    scale_fill_gradientn(colours = jet.colors(7), limits = c(0,
        45)) + labs(x = "Hour", y = "Day") + labs(fill = ("?C")) +
    ggtitle((e) ~ Chamber ~ Air ~ Temperature) + theme_bw() +
    theme(text = element_text(size = 17)) + theme(axis.text = element_text(size = 17)) +
```

```
    theme(legend.position = "right") + scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
    scale_x_continuous(breaks = scales::pretty_breaks(n = 5)) +
    annotate("rect", xmin = -Inf, xmax = Inf, ymin = 62, ymax = 67,
        alpha = 1) + theme(plot.title = element_text(size = 13.5,
    face = "bold")) + annotation_custom(textGrob("*", gp = gpar(col = "black",
    fontsize = 20)), xmin = -1.1, xmax = -0.8, ymin = 55.5, ymax = 56.5) +
    annotation_custom(textGrob("**", gp = gpar(col = "black",
        fontsize = 20)), xmin = -1.1, xmax = -0.8, ymin = 82.5,
        ymax = 83.5) + guides(fill = guide_colourbar(barwdith = 0.5,
    barheight = 18, nbin = 100)) + theme(plot.title = element_text(size = 20,
    face = "bold"))
e
```



#(f) is again the same but switch to eleveted data

```
dftrt <- data.frame(df$hm, df$month, df$Tair_al, df$T_treatment,
    df$Td)
str(dftrt)
```

```
## 'data.frame':    98502 obs. of  5 variables:
##  $ df.hm         : chr  "12" "12" "12" "12" ...
##  $ df.month      : chr  "08" "08" "08" "08" ...
##  $ df.Tair_al    : num  17.9 18.4 17.9 18.9 19.1 ...
##  $ df.T_treatment: chr  "ambient" "ambient" "ambient" "ambient" ...
##  $ df.Td         : POSIXct, format: "2016-08-28 12:02:00" "2016-08-28 12:17:00" ...
```

```
dft <- dftrt %>% group_by(df.T_treatment, df.Td) %>% summarise_at(.vars = names(.)[3:3],
    na.rm = TRUE, .funs = c(mean))
dft <- subset(dft, df.T_treatment == "elevated")
dft$date <- as.Date(dft$df.Td, format = "%Y-%m-%d")
dft["month"] <- format(dft$df.Td, "%m")
str(dft$month)
```

```
##  chr [1:49221] "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" ...
```

```
summary(dft$month)
```

```
##    Length     Class      Mode
##     49221 character character
```

```
dft["monthday"] <- format(dft$df.Td, "%m%d")
str(dft$monthday)
```

```
##  chr [1:49221] "0828" "0828" "0828" "0828" "0828" "0828" "0828" "0828" ...
```

```
summary(dft$monthday)
```

```
##    Length     Class      Mode
##     49221 character character
```

```
dft <- transform(dft, day = as.numeric(factor(monthday)))  #day
dft <- as.data.frame(dft)
dft$hm <- format(dft$df.Td, "%H")
summary(dft$hm)
```

```
##    Length     Class      Mode
##     49221 character character
```

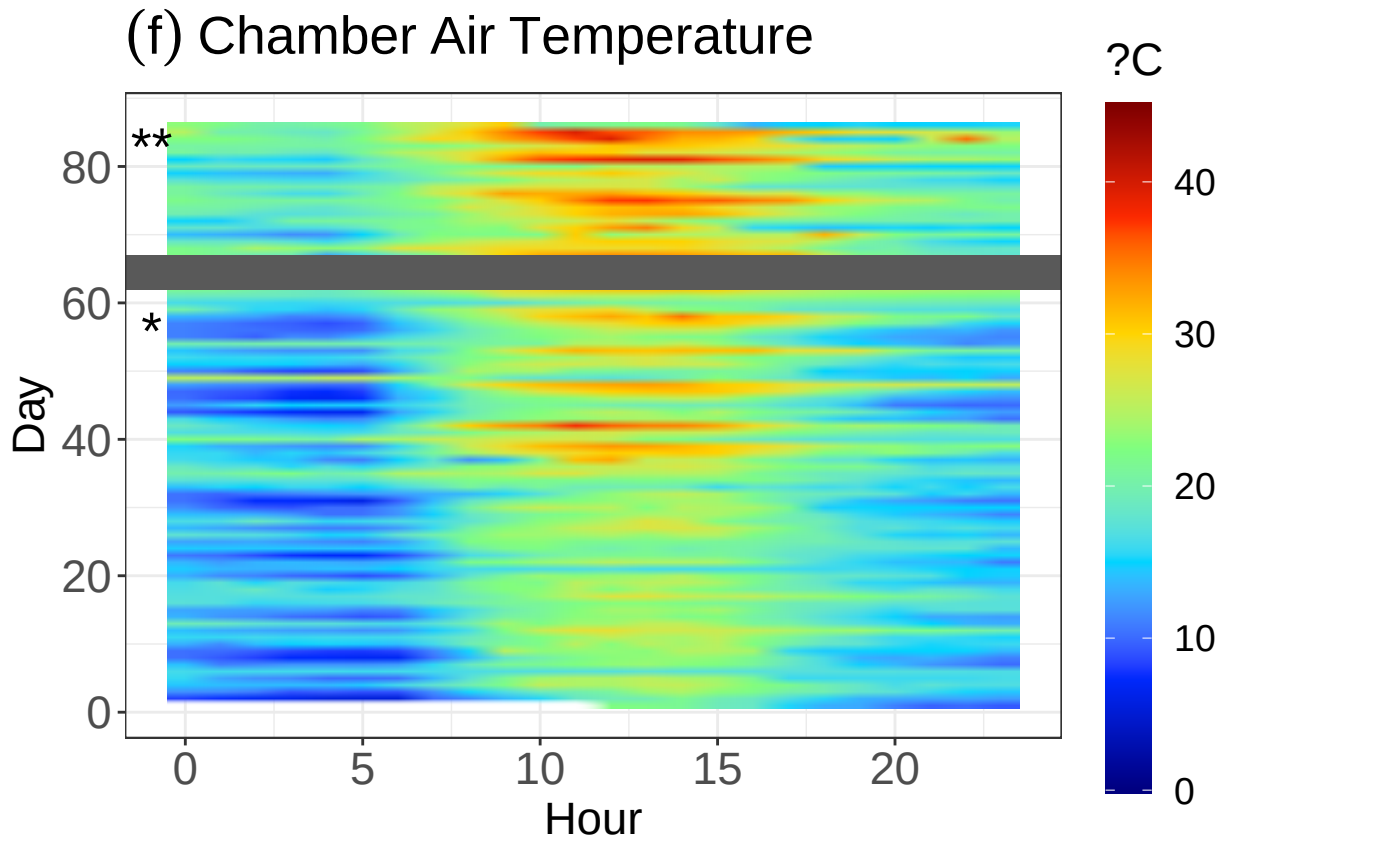
```
dfsml <- data.frame(dft$hm, dft$day, dft$df.Tair_al)
colnames(dfsml) <- c("x", "y", "z")
str(dfsml)
```

```
## 'data.frame':    49221 obs. of  3 variables:
##  $ x: chr  "12" "12" "12" "12" ...
##  $ y: num  1 1 1 1 1 1 1 1 1 1 ...
##  $ z: num  21.1 20.9 20.8 21 21.2 ...
```

```
dfsml$x <- as.numeric(paste(dfsml$x))
dfsml$y <- as.numeric(paste(dfsml$y))
str(dfsml)
```

```
## 'data.frame':    49221 obs. of  3 variables:
##  $ x: num  12 12 12 12 12 12 12 12 12 12 ...
##  $ y: num  1 1 1 1 1 1 1 1 1 1 ...
##  $ z: num  21.1 20.9 20.8 21 21.2 ...
```

```
x <- as.numeric(paste(dfsml$x))
y <- as.numeric(paste(dfsml$y))
z <- dfsml$z
gdf <- as.data.frame(cbind(x, y, z))
jet.colors <- colorRampPalette(c("#00007F", "blue", "#007FFF",
    "cyan", "#7FFF7F", "yellow", "#FF7F00", "red", "#7F0000"))
f <- ggplot(gdf, aes(x, y, fill = z)) + geom_raster(interpolate = TRUE) +
    scale_fill_gradientn(colours = jet.colors(7), limits = c(0,
        45)) + labs(x = "Hour", y = "Day") + labs(fill = ("?C")) +
    ggtitle((f) ~ Chamber ~ Air ~ Temperature) + theme_bw() +
    theme(text = element_text(size = 17)) + theme(axis.text = element_text(size = 17)) +
    theme(legend.position = "right") + scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
    scale_x_continuous(breaks = scales::pretty_breaks(n = 5)) +
    annotate("rect", xmin = -Inf, xmax = Inf, ymin = 62, ymax = 67,
        alpha = 1) + theme(plot.title = element_text(size = 13.5,
    face = "bold")) + annotation_custom(textGrob("*", gp = gpar(col = "black",
    fontsize = 20)), xmin = -1.1, xmax = -0.8, ymin = 55.5, ymax = 56.5) +
    annotation_custom(textGrob("**", gp = gpar(col = "black",
        fontsize = 20)), xmin = -1.1, xmax = -0.8, ymin = 82.5,
        ymax = 83.5) + guides(fill = guide_colourbar(barwdith = 0.5,
    barheight = 18, nbin = 100)) + theme(plot.title = element_text(size = 20,
    face = "bold"))
f
```



#(g) We use the same data frame to make the VPD plots

```
## note we calculate VPD with Reference RH and Temp
df$vpd <- RHtoVPD(RH = df$RHref_al, TdegC = df$Taref_al, Pa = 101)
summary(df$vpd)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0366  0.1890  0.5868  0.8180  1.1827  4.8727
```

```
dftrt <- data.frame(df$hm, df$month, df$vpd, df$T_treatment,
    df$Td)
str(dftrt)
```

```
## 'data.frame':    98502 obs. of  5 variables:
##  $ df.hm         : chr  "12" "12" "12" "12" ...
##  $ df.month      : chr  "08" "08" "08" "08" ...
##  $ df.vpd        : num  1.12 1.12 1.14 1.22 1.3 ...
##  $ df.T_treatment: chr  "ambient" "ambient" "ambient" "ambient" ...
##  $ df.Td         : POSIXct, format: "2016-08-28 12:02:00" "2016-08-28 12:17:00" ...
```

```
dft <- dftrt %>% group_by(df.T_treatment, df.Td) %>% summarise_at(.vars = names(.)[3:3],
    na.rm = TRUE, .funs = c(mean))
dft <- subset(dft, df.T_treatment == "ambient")
dft$date <- as.Date(dft$df.Td, format = "%Y-%m-%d")
dft["month"] <- format(dft$df.Td, "%m")
str(dft$month)
```

```
##  chr [1:49202] "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" ...
```

```
summary(dft$month)
```

```
##    Length     Class      Mode
##     49202 character character
```

```
dft["monthday"] <- format(dft$df.Td, "%m%d")
str(dft$monthday)
```

```
##  chr [1:49202] "0828" "0828" "0828" "0828" "0828" "0828" "0828" "0828" ...
```

```
summary(dft$monthday)
```

```
##    Length     Class      Mode
##     49202 character character
```

```
dft <- transform(dft, day = as.numeric(factor(monthday)))  #day
dft <- as.data.frame(dft)
dft$hm <- format(dft$df.Td, "%H")
summary(dft$hm)
```

```
##    Length     Class      Mode
##     49202 character character
```

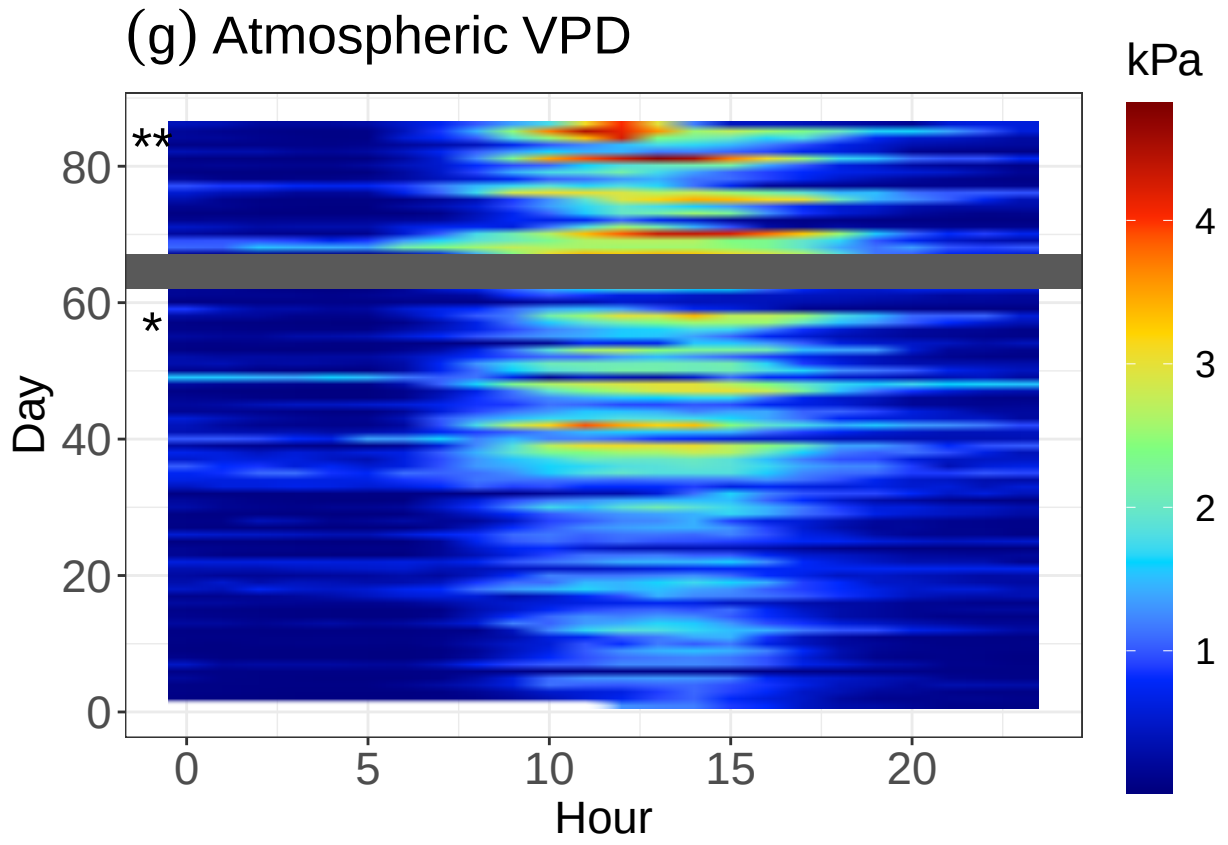
```r
dfsml <- data.frame(dft$hm, dft$day, dft$df.vpd)
colnames(dfsml) <- c("x", "y", "z")
str(dfsml)
```

```
## 'data.frame':    49202 obs. of  3 variables:
##  $ x: chr  "12" "12" "12" "12" ...
##  $ y: num  1 1 1 1 1 1 1 1 1 1 ...
##  $ z: num  1.12 1.13 1.13 1.13 1.13 ...
```

```r
dfsml$x <- as.numeric(paste(dfsml$x))
dfsml$y <- as.numeric(paste(dfsml$y))
str(dfsml)
```

```
## 'data.frame':    49202 obs. of  3 variables:
##  $ x: num  12 12 12 12 12 12 12 12 12 12 ...
##  $ y: num  1 1 1 1 1 1 1 1 1 1 ...
##  $ z: num  1.12 1.13 1.13 1.13 1.13 ...
```

```r
x <- as.numeric(paste(dfsml$x))
y <- as.numeric(paste(dfsml$y))
z <- dfsml$z
gdf <- as.data.frame(cbind(x, y, z))
jet.colors <- colorRampPalette(c("#00007F", "blue", "#007FFF",
    "cyan", "#7FFF7F", "yellow", "#FF7F00", "red", "#7F0000"))
g <- ggplot(gdf, aes(x, y, fill = z)) + geom_raster(interpolate = TRUE) +
    scale_fill_gradientn(colours = jet.colors(7), limits = c(0.03,
        4.8)) + labs(x = "Hour", y = "Day") + labs(fill = ("kPa")) +
    ggtitle((g) ~ Atmospheric ~ VPD) + theme_bw() + theme(text = element_text(size = 17)) +
    theme(axis.text = element_text(size = 17)) + theme(legend.position = "right") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
    scale_x_continuous(breaks = scales::pretty_breaks(n = 5)) +
    annotate("rect", xmin = -Inf, xmax = Inf, ymin = 62, ymax = 67,
        alpha = 1) + theme(plot.title = element_text(size = 13.5,
    face = "bold")) + annotation_custom(textGrob("*", gp = gpar(col = "black",
    fontsize = 20)), xmin = -1.1, xmax = -0.8, ymin = 55.5, ymax = 56.5) +
    annotation_custom(textGrob("**", gp = gpar(col = "black",
        fontsize = 20)), xmin = -1.1, xmax = -0.8, ymin = 82.5,
        ymax = 83.5) + guides(fill = guide_colourbar(barwdith = 0.5,
    barheight = 18, nbin = 100)) + theme(plot.title = element_text(size = 20,
    face = "bold"))
g
```



#(h) again we just switch from reference to chamber

```
## Note we switch to in chamber measurements
df$vpd <- RHtoVPD(RH = df$RH_al, TdegC = df$Tair_al, Pa = 101)
summary(df$vpd)  #the vapour pr
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  0.0459  0.2704  0.6049  0.8287  1.1360  6.2394     517
```

```
dftrt <- data.frame(df$hm, df$month, df$vpd, df$T_treatment,
    df$Td)
str(dftrt)
```

```
## 'data.frame':    98502 obs. of  5 variables:
##  $ df.hm         : chr  "12" "12" "12" "12" ...
##  $ df.month      : chr  "08" "08" "08" "08" ...
##  $ df.vpd        : num  0.994 1.027 0.985 1.083 1.113 ...
##  $ df.T_treatment: chr  "ambient" "ambient" "ambient" "ambient" ...
##  $ df.Td         : POSIXct, format: "2016-08-28 12:02:00" "2016-08-28 12:17:00" ...
```

```
dft <- dftrt %>% group_by(df.T_treatment, df.Td) %>% summarise_at(.vars = names(.)[3:3],
    na.rm = TRUE, .funs = c(mean))
dft <- subset(dft, df.T_treatment == "ambient")
dft$date <- as.Date(dft$df.Td, format = "%Y-%m-%d")
dft["month"] <- format(dft$df.Td, "%m")
str(dft$month)
```

```
##  chr [1:49202] "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" ...
```

```
summary(dft$month)
```

```
##    Length     Class      Mode
##     49202 character character
```

```
dft["monthday"] <- format(dft$df.Td, "%m%d")
str(dft$monthday)
```

```
##  chr [1:49202] "0828" "0828" "0828" "0828" "0828" "0828" "0828" "0828" ...
```

```
summary(dft$monthday)
```

```
##    Length     Class      Mode
##     49202 character character
```

```
dft <- transform(dft, day = as.numeric(factor(monthday)))  #day
dft <- as.data.frame(dft)
dft$hm <- format(dft$df.Td, "%H")
summary(dft$hm)
```

```
##    Length     Class      Mode
##     49202 character character
```

```
dfsml <- data.frame(dft$hm, dft$day, dft$df.vpd)
write.table(dft, "clipboard-16384", sep = "\t", row.names = FALSE)
colnames(dfsml) <- c("x", "y", "z")
str(dfsml)
```

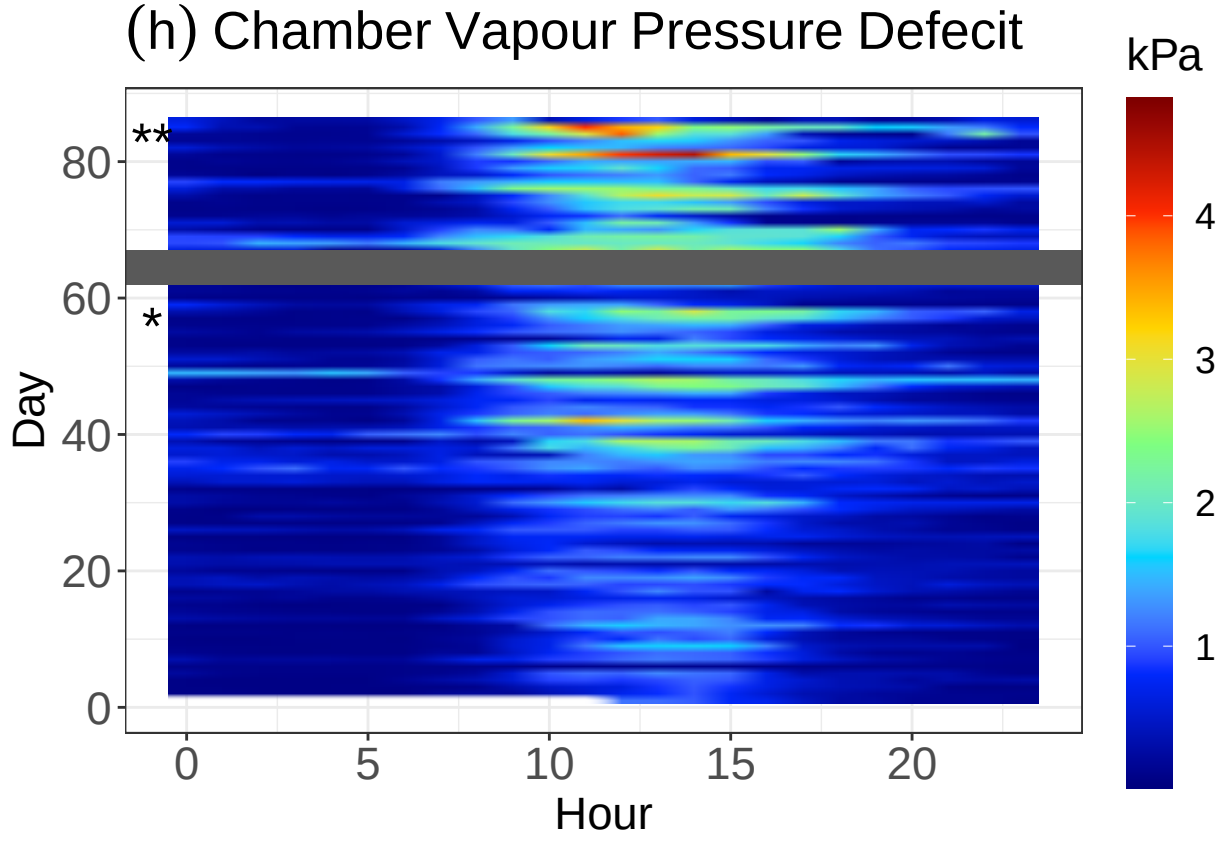
```
## 'data.frame':    49202 obs. of  3 variables:
##  $ x: chr  "12" "12" "12" "12" ...
##  $ y: num  1 1 1 1 1 1 1 1 1 1 ...
##  $ z: num  0.994 0.987 0.991 1.014 1.036 ...
```

```
dfsml$x <- as.numeric(paste(dfsml$x))
dfsml$y <- as.numeric(paste(dfsml$y))
str(dfsml)
```

```
## 'data.frame':    49202 obs. of  3 variables:
##  $ x: num  12 12 12 12 12 12 12 12 12 12 ...
##  $ y: num  1 1 1 1 1 1 1 1 1 1 ...
##  $ z: num  0.994 0.987 0.991 1.014 1.036 ...
```

```
x <- as.numeric(paste(dfsml$x))
y <- as.numeric(paste(dfsml$y))
z <- dfsml$z
gdf <- as.data.frame(cbind(x, y, z))
jet.colors <- colorRampPalette(c("#00007F", "blue", "#007FFF",
```

```
    "cyan", "#7FFF7F", "yellow", "#FF7F00", "red", "#7F0000"))
h <- ggplot(gdf, aes(x, y, fill = z)) + geom_raster(interpolate = TRUE) +
    scale_fill_gradientn(colours = jet.colors(7), limits = c(0.03,
        4.8)) + labs(x = "Hour", y = "Day") + labs(fill = ("kPa")) +
    ggtitle((h) ~ Chamber ~ Vapour ~ Pressure ~ Defecit) + theme_bw() +
    theme(text = element_text(size = 17)) + theme(axis.text = element_text(size = 17)) +
    theme(legend.position = "right") + scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
    scale_x_continuous(breaks = scales::pretty_breaks(n = 5)) +
    annotate("rect", xmin = -Inf, xmax = Inf, ymin = 62, ymax = 67,
        alpha = 1) + theme(plot.title = element_text(size = 13.5,
    face = "bold")) + annotation_custom(textGrob("*", gp = gpar(col = "black",
    fontsize = 20)), xmin = -1.1, xmax = -0.8, ymin = 55.5, ymax = 56.5) +
    annotation_custom(textGrob("**", gp = gpar(col = "black",
        fontsize = 20)), xmin = -1.1, xmax = -0.8, ymin = 82.5,
        ymax = 83.5) + guides(fill = guide_colourbar(barwidth = 0.5,
    barheight = 18, nbin = 100)) + theme(plot.title = element_text(size = 20,
    face = "bold"))
h
```



#(i) as above but for eleveted data

```
df$vpd <- RHtoVPD(RH = df$RH_al, TdegC = df$Tair_al, Pa = 101)
summary(df$vpd)  #the vapour pr
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  0.0459  0.2704  0.6049  0.8287  1.1360  6.2394     517
```

```
dftrt <- data.frame(df$hm, df$month, df$vpd, df$T_treatment,
    df$Td)
str(dftrt)
```

```
## 'data.frame':    98502 obs. of  5 variables:
##  $ df.hm         : chr  "12" "12" "12" "12" ...
##  $ df.month      : chr  "08" "08" "08" "08" ...
##  $ df.vpd        : num  0.994 1.027 0.985 1.083 1.113 ...
##  $ df.T_treatment: chr  "ambient" "ambient" "ambient" "ambient" ...
##  $ df.Td         : POSIXct, format: "2016-08-28 12:02:00" "2016-08-28 12:17:00" ...
```

```
dft <- dftrt %>% group_by(df.T_treatment, df.Td) %>% summarise_at(.vars = names(.)[3:3],
    na.rm = TRUE, .funs = c(mean))
dft <- subset(dft, df.T_treatment == "elevated")
dft$date <- as.Date(dft$df.Td, format = "%Y-%m-%d")
dft["month"] <- format(dft$df.Td, "%m")
str(dft$month)
```

```
##  chr [1:49221] "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" "08" ...
```

```
summary(dft$month)
```

```
##    Length     Class      Mode
##     49221 character character
```

```
dft["monthday"] <- format(dft$df.Td, "%m%d")
str(dft$monthday)
```

```
##  chr [1:49221] "0828" "0828" "0828" "0828" "0828" "0828" "0828" "0828" ...
```

```
summary(dft$monthday)
```

```
##    Length     Class      Mode
##     49221 character character
```

```
dft <- transform(dft, day = as.numeric(factor(monthday)))  #day
dft <- as.data.frame(dft)
dft$hm <- format(dft$df.Td, "%H")
summary(dft$hm)
```

```
##    Length     Class      Mode
##     49221 character character
```

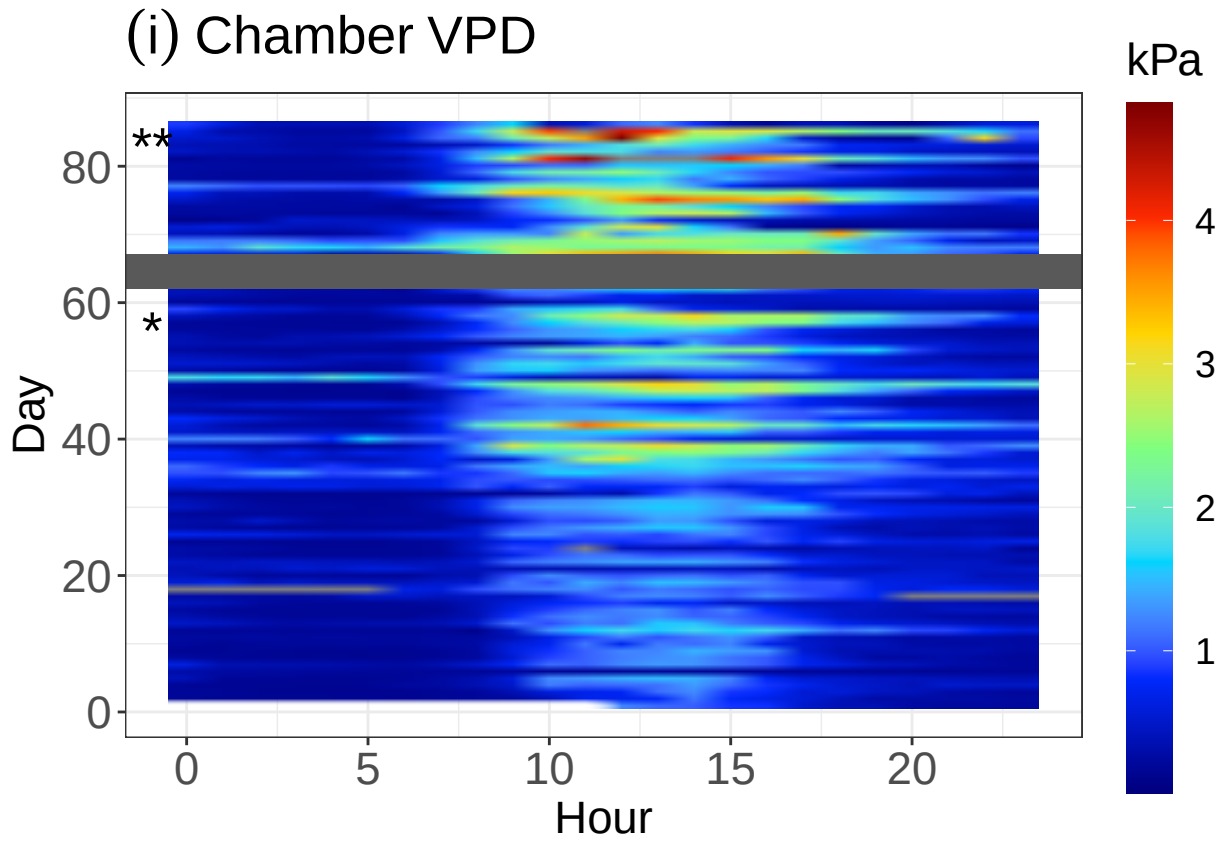
```
dfsml <- data.frame(dft$hm, dft$day, dft$df.vpd)
write.table(dft, "clipboard-16384", sep = "\t", row.names = FALSE)
colnames(dfsml) <- c("x", "y", "z")
str(dfsml)
```

```
## 'data.frame':    49221 obs. of  3 variables:
##  $ x: chr  "12" "12" "12" "12" ...
##  $ y: num  1 1 1 1 1 1 1 1 1 1 ...
##  $ z: num  1.13 1.2 1.21 1.2 1.11 ...
```

```
dfsml$x <- as.numeric(paste(dfsml$x))
dfsml$y <- as.numeric(paste(dfsml$y))
str(dfsml)
```

```
## 'data.frame':    49221 obs. of  3 variables:
##  $ x: num  12 12 12 12 12 12 12 12 12 12 ...
##  $ y: num  1 1 1 1 1 1 1 1 1 1 ...
##  $ z: num  1.13 1.2 1.21 1.2 1.11 ...
```

```
x <- as.numeric(paste(dfsml$x))
y <- as.numeric(paste(dfsml$y))
z <- dfsml$z
gdf <- as.data.frame(cbind(x, y, z))
jet.colors <- colorRampPalette(c("#00007F", "blue", "#007FFF",
    "cyan", "#7FFF7F", "yellow", "#FF7F00", "red", "#7F0000"))
i <- ggplot(gdf, aes(x, y, fill = z)) + geom_raster(interpolate = TRUE) +
    scale_fill_gradientn(colours = jet.colors(7), limits = c(0.03,
        4.8)) + labs(x = "Hour", y = "Day") + labs(fill = ("kPa")) +
    ggtitle((i) ~ Chamber ~ VPD) + theme_bw() + theme(text = element_text(size = 17)) +
    theme(axis.text = element_text(size = 17)) + theme(legend.position = "right") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
    scale_x_continuous(breaks = scales::pretty_breaks(n = 5)) +
    annotate("rect", xmin = -Inf, xmax = Inf, ymin = 62, ymax = 67,
        alpha = 1) + theme(plot.title = element_text(size = 13.5,
    face = "bold")) + annotation_custom(textGrob("*", gp = gpar(col = "black",
    fontsize = 20)), xmin = -1.1, xmax = -0.8, ymin = 55.5, ymax = 56.5) +
    annotation_custom(textGrob("**", gp = gpar(col = "black",
        fontsize = 20)), xmin = -1.1, xmax = -0.8, ymin = 82.5,
        ymax = 83.5) + guides(fill = guide_colourbar(barwdith = 0.5,
    barheight = 18, nbin = 100)) + theme(plot.title = element_text(size = 20,
    face = "bold"))
i
```

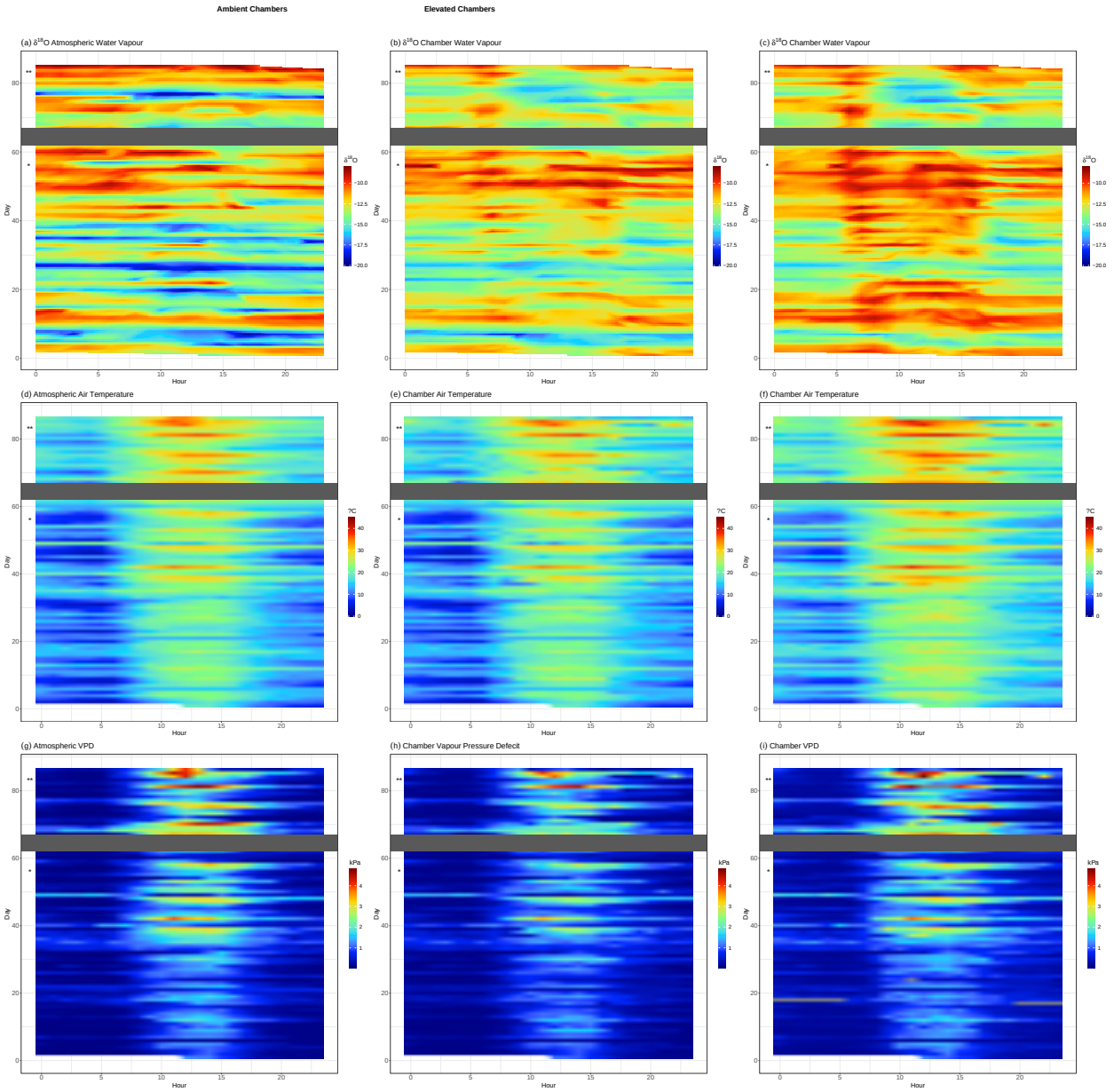


#TO finish figure 2 we need to merge all the panels

```
plotmash<-plot_grid(a,b,c,d,e,f,g,h,i, nrow=3,ncol=3, align="vh")
title <- ggdraw() +
  draw_label(
    "
    fontface = 'bold',
    x = 0,
    hjust = 0,
    size=20
  ) +
  theme(
    # add margin on the left of the drawing canvas,
    # so title is aligned with left edge of first plot
    plot.margin = margin(0, 0, 0, 7)
  )


figtemporal=plot_grid(
  title, plotmash,
  ncol = 1,
  # rel_heights values control vertical title margins
  rel_heights = c(0.05, 1)
)
#Set your working directory (UNHASH)
#setwd("D:/WTC4 Data/new temporal graph")
```

```
#UNHASH below and it saves a HQ tiff
ggsave("dayinterpHQstar.tiff", width = 50, height = 50, units = "cm", dpi=600)
```

```
plot(figtemporal)
```



#The rest of the figures use the campaign data- they all depend on the mass balance being run #Here is the Mass Balance code which uses the campaign data. Firstly lets clear the workspace

```
rm(list = ls())
```

#Mass Balance (used to calculate d18otranspiration)

```
googledriveWTC4ISOTOPEIDCAMPAIGNDATA <- "1_K_m1eq-iQGQmOC3hqoI7_8qxVve3YoB"
df <- read.csv(sprintf("https://docs.google.com/uc?id=%s&export=download",
    googledriveWTC4ISOTOPEIDCAMPAIGNDATA))
######
df["source"] = -3.28  ##from branch water
###
R <- 287.05
Rv <- 461.95
Pr <- 101.3
finr <- 0.082058
## These equations are a combo of the WTC4 MS, John Drakes GBC
## Biology and Craigs orginal WTC4 Agmet paper Pref###
df["Pref"] = ((df$AIRPRESS_mean * 1000) - df$Vwat_mean)/(R *
    (df$Tair_al_mean + 273)) + ((df$Vwat_mean)/(Rv * df$Vwat_mean +
    273))
#### Fin####
df["fin"] = (df$kfactor_mean * (sqrt(df$DIFFP_mean * (1.2/df$Pref))) *
    (273/(df$Tair_al_mean + 273)) * (df$AIRPRESS_mean/Pr))
### Finmol###
df["finmol"] = ((df$AIRPRESS_mean * 0.00986923267) * df$fin)/(finr *
    (df$Tair_al_mean + 273))
## fout###
df["fout"] = df$fin * (((df$AIRPRESS_mean) - df$Href_mean)/((df$Tair_al_mean) -
    df$Href_mean))
## fout mol
df["foutmol"] = ((df$AIRPRESS_mean * 0.00986923267) * df$fout)/(finr *
    (df$Tair_al_mean + 273))
# uoutwout
df["uoutwout"] = df$foutmol * df$Vwat_mean
# uinwin
df["uinwin"] = df$finmol * df$Fwat_mean
# condout
df["condout"] = df$CONDH2O_mean
## E
df["E"] = df$uoutwout - df$uinwin + df$condout
## E-wo
df["e-wo"] = df$E - df$Vwat_mean
## d18out
df["d18out"] = df$d18O.corrected_WV * df$uoutwout
# d18in
df["d18in"] = df$d18O.corrected_WV.AMB. * df$uinwin
## d18c
df["d18c"] = df$d18O.corrected_Cond * df$condout
## over e-wo
df["d18oute"] = df$d18out/df$`e-wo`
df["d18ine"] = df$d18in/df$`e-wo`
df["d18Ce"] = df$d18c/df$E
## d trans
df["dtrans"] = df$d18oute - df$d18ine + df$d18Ce
mean(df$dtrans, na.rm = TRUE)
```

```
## [1] -4.059054
```

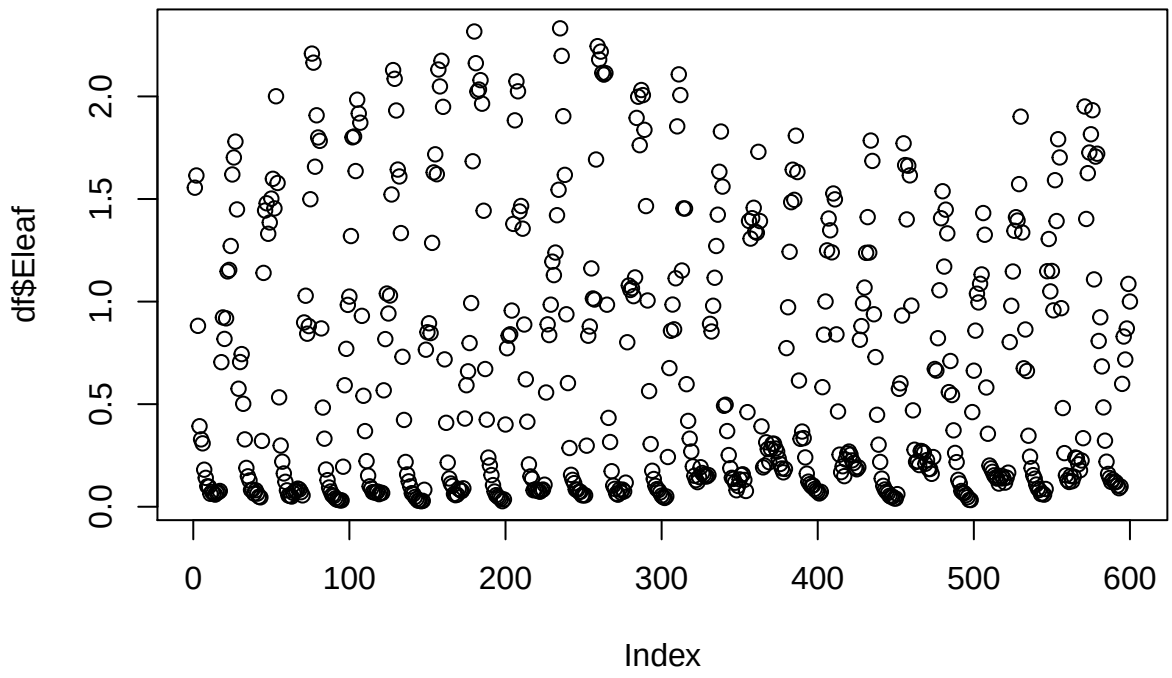
```
df$dtrans
```

```
##   [1]   -3.5817759          NA   -3.6578406  -13.3709903   -7.3800345
##   [6]   -6.1986008   -1.9460850   21.8780351   -7.3795118   -6.5298608
##  [11]          NA   -3.7050301          NA           NA           NA
##  [16]          NA   -4.5537440          NA           NA           NA
##  [21]   -2.4298365   -0.8345006   -0.5829286           NA           NA
##  [26]   -3.5738251   -3.9870011          NA   -4.5948591   -4.0612285
##  [31]   -3.6814035   -4.1516340   -2.4027941   -4.9816582   -4.1619611
##  [36]   -3.9715833          NA   -3.9737621           NA           NA
##  [41]          NA          NA   -2.2947784           NA           NA
##  [46]          NA   -0.8842089   -1.5780147   -2.9163557           NA
##  [51]          NA   -4.1291359   -4.9402839           NA   -4.1951964
##  [56]   -4.3276591   -4.4452821   -4.9153413   -6.1405438   -5.6683211
##  [61]   -5.8500171   -5.5487758          NA   -4.3619580           NA
##  [66]          NA          NA          NA   -4.7763202           NA
##  [71]          NA          NA   -3.1303461   -3.0457070   -3.8791618
##  [76]          NA          NA   -6.0251935   -3.8837173           NA
##  [81]   -4.8486924   -5.0500593   -4.5871664   -3.1248281  -21.5171516
##  [86]   -8.8153630   -2.7126613   -5.3963269           NA   -4.1062643
##  [91]          NA          NA          NA           NA   -2.5193413
##  [96]          NA          NA          NA   -0.1643553   -0.5445810
## [101]   -2.2315547          NA          NA   -5.3317258   -3.5169586
## [106]          NA   -3.8972390   -3.9256978   -3.0281474    2.4255296
## [111]   -9.3587040   -5.0877744   -6.2869296   -6.8472158           NA
## [116]   -2.9601841          NA          NA           NA           NA
## [121]   -3.5323000          NA          NA           NA   -1.4000183
## [126]   -1.9032523   -1.6426589          NA           NA   -3.8144426
## [131]   -6.2359519          NA   -5.5317872   -6.5682124   -6.6680817
## [136]   -6.6879979    6.7733551   -2.6058549   -6.0849021   -0.9843672
## [141]          NA   -5.7563768          NA           NA           NA
## [146]          NA    0.2623753          NA           NA           NA
## [151]   -2.4639520   -3.6928927   -5.6305763           NA           NA
## [156]   -6.3259672   -4.9306668          NA   -4.6104527   -4.5038540
## [161]   -4.8698100   -1.8888246   -8.0830541   -5.1031650   -3.9230117
## [166]   -5.4783092          NA   -6.0896237           NA           NA
## [171]          NA          NA   -3.9212148           NA           NA
## [176]          NA   -3.3160469   -0.9128210   -2.0312167           NA
## [181]          NA   -4.5817279   -6.1828781           NA   -5.4570836
## [186]   -4.8202438   -5.1989917   -5.2274196   -2.1136833    0.7805819
## [191]   -4.6388439   -3.7953183          NA   -2.6181093           NA
## [196]          NA          NA          NA   -2.7367348           NA
## [201]          NA          NA   -1.2509284   -0.9258594   -3.2688225
## [206]          NA          NA   -5.3867154   -4.6689617           NA
## [211]   -4.8145217   -4.8938360   -5.5612670   -3.8016982   -7.3547665
## [216]   -7.3939628   -6.0262837   -3.9760134           NA   -7.3149205
## [221]          NA          NA          NA           NA   -2.1649008
## [226]          NA          NA          NA   -0.9727131   -1.2611879
## [231]   -2.5572688          NA          NA   -4.5432074   -4.9704657
## [236]          NA   -5.1980299   -5.3716483   -5.5737748   -5.1818557
## [241]  -12.3980411   -3.5541585   -3.2896194   -2.2716788           NA
## [246]   -4.4010680          NA          NA           NA           NA
## [251]   -0.9704332          NA          NA           NA   -1.4813978
```

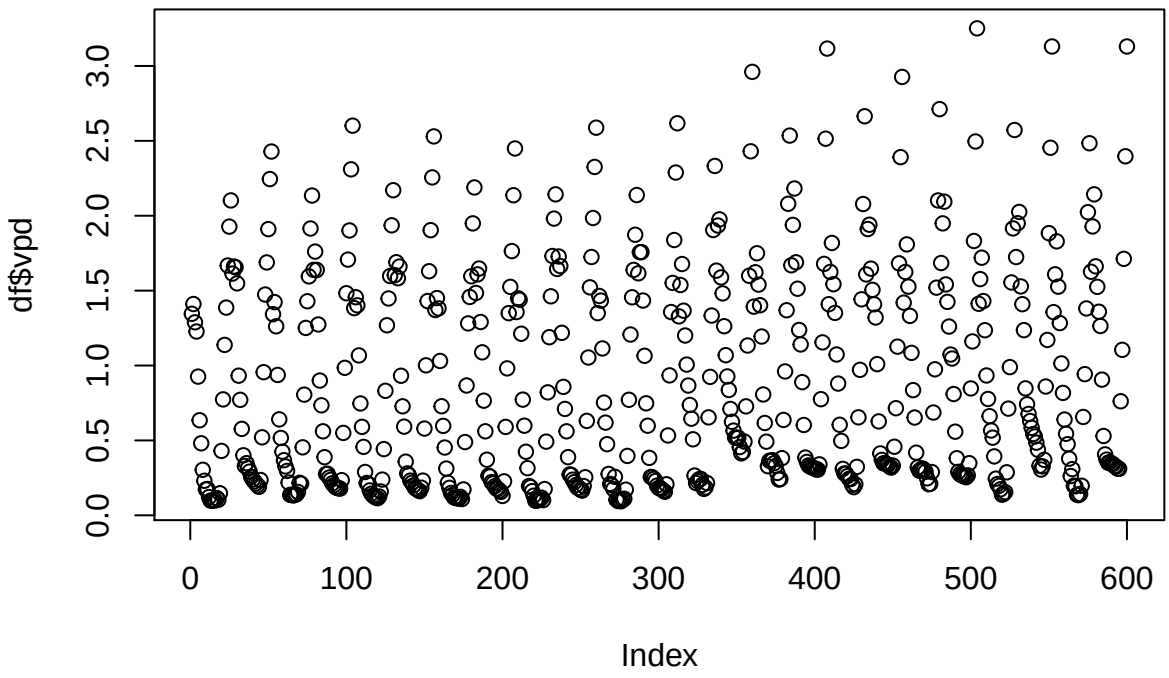
```
## [256]   -5.4091865   -1.8955771           NA           NA   -4.7736717
## [261]   -5.1820351           NA   -6.1334819   -4.7441687   -4.1363402
## [266]  -29.6377500   -5.5520104   -5.1091986   -4.0309691   -7.7952776
## [271]           NA   -6.0703927           NA           NA           NA
## [276]           NA   -2.3753544           NA           NA           NA
## [281]   -1.3246119   -2.5068918   -3.1480636           NA           NA
## [286]   -5.2429716   -6.2664421           NA   -7.6089553   -6.7242554
## [291]   -7.0553893   -7.6390885   -6.6703997  -12.5523495   -2.7176616
## [296]   -1.8264041           NA   -5.7561256           NA           NA
## [301]           NA           NA   -3.3496771           NA           NA
## [306]           NA   -2.0109732   -4.3491583   -5.9636733           NA
## [311]           NA   -5.0206232   -3.5937335   -3.5059203   -3.9680745
## [316]   -4.3512028   -4.4474200   -4.7151747   -6.7127502   -3.6910816
## [321]   -3.8026400   -3.3332518           NA           NA           NA
## [326]           NA           NA           NA           NA   -2.3834907
## [331]           NA   -0.8896192   -1.0504311   -2.3301798   -2.7989483
## [336]   -3.1328952   -3.1834573   -3.3339100   -3.6060879   11.1237978
## [341] -130.5571116   -0.1185001   -2.1644502   -2.5293219   -2.5801615
## [346]   -3.1244542           NA           NA           NA           NA
## [351]           NA           NA           NA   -1.5469813           NA
## [356]   -1.7177358   -1.2815864   -1.5075332   -2.5443232   -3.0932929
## [361]   -3.4680821   -3.3008994   -4.0373117   -2.4654335   -2.9194604
## [366]   -3.1636260   -1.7108620   -3.4273893   -4.1837679   -3.8324184
## [371]           NA           NA           NA           NA           NA
## [376]           NA           NA   -2.4875244           NA   -0.4315493
## [381]   -2.1716984   -2.0898570   -3.4586471   -3.4137969   -3.5215814
## [386]   -3.7924195   -5.7815398   -5.7016682    4.5703232  -15.1102196
## [391]  -12.7519151   -2.2407341   -2.6226298   -1.7920370           NA
## [396]           NA           NA           NA           NA           NA
## [401]           NA   -0.1172178           NA   -0.7339347   -1.1745425
## [406]   -2.2247823   -3.2595673   -3.7852533   -4.3697271   -4.4861386
## [411]   -4.5534392   -4.8980887   -3.0395887   -4.2026546   -3.9365458
## [416]   -4.5381316   -5.1998267   -4.1896091           NA           NA
## [421]           NA           NA           NA           NA           NA
## [426]   -2.0913244           NA   -0.8832264   -1.5809943   -2.7880079
## [431]   -3.8232045   -3.4835127   -3.4019611   -2.4875100   -3.0078350
## [436]   -4.4954906   -4.9826162   -5.0390145   -6.5843498   -1.1523570
## [441]   -3.2178833   -1.5615519           NA           NA           NA
## [446]           NA           NA           NA           NA   -1.6155508
## [451]           NA   -3.0152833   -2.2836510   -1.2333983   -2.3700968
## [456]   -4.2494158   -3.5254498   -4.6739472   -4.4816894   -5.3967297
## [461]    3.3620717   -3.7172227   -5.1872893   -3.2676363   -3.0222515
## [466]   -3.0533854           NA           NA           NA           NA
## [471]           NA           NA           NA   -4.3293707           NA
## [476]    0.1514782   -1.8202806   -1.7454321   -3.3591624   -3.0662236
## [481]   -3.5294173   -4.1915799   -3.6486134   -4.0698207   -4.4088784
## [486]   -4.0168058   -4.0417123   -1.6918180   -2.3708241   -3.1922415
## [491]           NA           NA           NA           NA           NA
## [496]           NA           NA   -0.2234090           NA   -0.6996132
## [501]   -1.2819709   -1.9859696   -3.0775471   -4.2234321   -3.9169042
## [506]   -4.1329508   -5.5261725   -7.4172342   -2.2848295   -4.4231698
## [511]   -3.5490625   -5.1635730   -5.3540450   -4.0868294           NA
## [516]           NA           NA           NA           NA           NA
## [521]           NA   -2.8541199           NA   -2.9167960    0.5880855
```

```
## [526]   -2.0658629   -5.0937060   -5.9773564   -3.3325231   -3.4595530
## [531]   -3.4409000   -6.1010338   -4.8263517   -5.4271628   -1.7634209
## [536]   -2.3244595   -3.1449518   -2.6179971           NA           NA
## [541]           NA           NA           NA           NA           NA
## [546]           NA           NA   -1.7645519   -2.4288349   -1.9273437
## [551]   -2.9202377   -2.7443876   -2.7541673   -2.1843729   -2.9392157
## [556]   -4.3233058   -1.7426004   -2.2819322   -1.7542212   -2.3051777
## [561]   -5.1447090   -3.1184691           NA           NA           NA
## [566]           NA           NA           NA           NA   -1.5514410
## [571]           NA   -1.8348899   -2.3290320   -2.8376645   -1.8112464
## [576]   -4.0274561   -4.7339398   -3.4095657   -4.0583611   -5.0734351
## [581]   -4.0153561   -4.4987926   -3.6560637   -9.3209161    0.4598862
## [586]   -1.3477602           NA           NA           NA           NA
## [591]           NA           NA           NA   -0.5781521           NA
## [596]   -0.6006414   -1.3920865   -2.0737908   -3.5958645   -2.7443876
```

```
# We need to express mulriple variables on a leaf area basis:
df$leafarea[df$chamber == "C01"] <- 23.62101328  #M^2
df$leafarea[df$chamber == "C02"] <- 17.17025483
df$leafarea[df$chamber == "C03"] <- 12.37672299
df$leafarea[df$chamber == "C04"] <- 24.62392584
df$leafarea[df$chamber == "C05"] <- 12.92316481
df$leafarea[df$chamber == "C06"] <- 29.71035506
df$leafarea[df$chamber == "C07"] <- 14.61233375
df$leafarea[df$chamber == "C08"] <- 24.42841037
df$leafarea[df$chamber == "C09"] <- 17.33770004
df$leafarea[df$chamber == "C10"] <- 17.13905693
df$leafarea[df$chamber == "C11"] <- 9.63132978
df$leafarea[df$chamber == "C12"] <- 27.27409398
df$LACM <- df$leafarea/10000
# leaf water turnover = W/gt*wi wi is the leaf intercellular
# vapour concentration (mol water vapour/mol moist air) using
# Licor eqn mole fraction of water vapor within the leaf,
# mmol H2O mol air-1.
df["Ei"] <- 6.13753 * exp(df$leaftemp * ((18.564 - (df$leaftemp/254.4)))/(df$leaftemp +
    255.57))
df$wi <- df$Ei/df$AIRPRESS_mean/10  #mol mol -1
## make E a flux relative to tree lef area
df$Emmol <- df$E * 1000  #go to mmol
df$Eleaf <- df$Emmol/df$leafarea
plot(df$Eleaf)  #(mmol H2O m-2 s-1)
```




```
## Calculate VPD
df$esat <- esat(TdegC = df$Tair_al_mean, Pa = 101)
df$vpd <- RHtoVPD(RH = df$RH_al_mean, TdegC = df$Tair_al_mean,
    Pa = 101)
plot(df$vpd)
```




```
summary(df$vpd)  #the vapour pressure deficit of the chamber airspace (kPa)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.09387 0.25853 0.62815 0.88106 1.43973 3.25110
```

```
## Calculate gt -gt is total conductance to water vapour
## through the stomata and leaf boundary layer
df$gs <- df$Eleaf/df$vpd/10  # mol
#################################### Calculate Leaf Water Residence Times##
df$W <- 12  #mol m-2
df$t <- df$W/(df$gs * df$wi)  # in seconds
df$tm <- df$t/60
mean(df$tm, na.rm = TRUE)
```

```
## [1] 204.0551
```

```
min(df$tm, na.rm = TRUE)
```

```
## [1] 7.468231
```

```
max(df$tm, na.rm = TRUE)  # T in mins
```

```
## [1] 1118.82
```

```r
df$th <- df$t/3600
mean(df$th, na.rm = TRUE)
```

```
## [1] 3.400919
```

```r
min(df$th, na.rm = TRUE)
```

```
## [1] 0.1244705
```

```r
max(df$th, na.rm = TRUE)  # T in hour
```

```
## [1] 18.647
```

```r
### Calculate for isotopes
df$ek = 0.027
df$alphak <- 1 + df$ek
df["Eplus"] <- 2.644 - (3.206 * ((10^3)/(df$Tair_al_mean + 273.16))) +
    (1.534 * ((10^6)/((df$Tair_al_mean + 273.16)^2)))
df$betaplus <- (df$Eplus/1000) + 1
mean(df$alphak * df$betaplus)
```

```
## [1] 1.036769
```

```r
df$p <- 1.2  ##See SI from MS
## df$alphak*df$betaplus shold be around 1.040 (Table 1
## Grahams and Lucas paper)
df$tiso <- (df$W * df$alphak * df$betaplus)/(df$p * df$gs * df$wi)
df$tmiso <- df$tiso/60
mean(df$tmiso, na.rm = TRUE)
```

```
## [1] 176.3582
```

```r
min(df$tmiso, na.rm = TRUE)
```

```
## [1] 6.452559
```

```r
max(df$tmiso, na.rm = TRUE)  # T in mins
```

```
## [1] 967.6147
```

```r
df$thiso <- df$tiso/3600
mean(df$thiso, na.rm = TRUE)
```

```
## [1] 2.939303
```

```r
min(df$thiso, na.rm = TRUE)
```

```
## [1] 0.1075426
```

```r
max(df$thiso, na.rm = TRUE)
```

```
## [1] 16.12691
```

```r
### calculate number obsertautions of dtrans within 2 mil
df$range <- ifelse(df$dtrans >= -5.28 & df$dtrans <= -1.28, "within",
    "not")
df$range <- as.factor(df$range)
summary(df$range)  # 107 not in 264 within = 367 total % witin 2 % is 258 /367
```

```
##    not within   NA's
##    107    264    229
```

```r
264/(264 + 107)  #~70%
```

```
## [1] 0.7115903
```

```r
## get chamber %
tapply(df$range, df$chamber, summary)
```

```
## $C01
##    not within   NA's
##     11     20     19
##
## $C02
##    not within   NA's
##      4     27     19
##
## $C03
##    not within   NA's
##      6     25     19
##
## $C04
##    not within   NA's
##     14     17     19
##
## $C05
##    not within   NA's
##      5     26     19
##
## $C06
##    not within   NA's
##     15     16     19
##
## $C07
##    not within   NA's
##      7     24     19
##
## $C08
##    not within   NA's
##      8     23     19
```

```
##
## $C09
##    not within   NA's
##     12     19     19
##
## $C10
##    not within   NA's
##      7     23     20
##
## $C11
##    not within   NA's
##      5     26     19
##
## $C12
##    not within   NA's
##     13     18     19
```

```
## The following lines of code define when chamber air temp
## and chamber dew point temp are within 1C or negative We
## need to figure out if Dew Point deltas could be influecing
## gs
df$dpdiffwtc <- df$Tair_al - df$DewPntC
df$dpdiffwtc
```

```
##   [1] 16.63234749 17.18488584 17.20504503 16.34970145 12.22611442  8.79860713
##   [7]  6.04169216  3.70143244  2.43402905  2.27421036  1.57991500  0.82515419
##  [13] -0.08270140 -0.63299198 -0.49419271 -0.73067868 -0.53846472  1.36141329
##  [19]  2.50379746  5.09629953  8.71769872 13.69295764 15.91456761 18.42354603
##  [25] 20.14412646 20.77773838 19.58960834 20.27975044 20.06760960 19.46918135
##  [31] 15.17716851 11.82758918  8.96927252  6.71758041  5.34439118  5.34641984
##  [37]  4.46031482  3.83844554  2.93857472  2.36893396  2.41513736  2.30349720
##  [43]  2.30171168  4.27697830  5.80215077  8.25164843 11.67172303 16.74685740
##  [49] 18.90039339 21.47947855 23.10121851 23.74359231 16.58054581 16.99965391
##  [55] 17.55247431 16.29754238 12.11630144  8.70459404  5.95782771  3.64085112
##  [61]  2.36319132  2.33596864  1.55787773  0.74147642 -0.12801130 -0.60613306
##  [67] -0.51338570 -0.75153261 -0.68698376  0.67752765  3.30986564  5.45283103
##  [73]  8.67001047 13.71833444 16.03668442 18.59840078 20.14012008 20.69266396
##  [79] 19.70110121 20.39389286 20.34128766 18.67732744 14.97539153 11.61218014
##  [85]  8.94695702  6.69113991  5.33645542  5.34151568  4.42856236  3.83115828
##  [91]  2.93524330  2.38738375  2.39924498  2.28634775  2.46039689  4.50461452
##  [97]  5.63103311  8.48076987 11.78813925 16.70159793 18.99540749 21.55966015
## [103] 23.27461700 23.97992902 16.78260102 17.21465893 17.52827358 15.90739450
## [109] 11.90244918  8.58800197  5.70314007  3.63750014  2.26346056  2.41594605
## [115]  1.35989208  0.75644157 -0.07246118 -0.60548761 -0.47361735 -0.80592476
## [121] -0.55216291  1.10205984  2.80155659  5.46620526  9.07839193 14.10423560
## [127] 16.11226006 18.46593170 20.30195417 20.93650107 19.83297887 20.31277056
## [133] 20.14264989 18.98884902 14.86903172 11.55382800  8.67849021  6.61164269
## [139]  5.26853817  5.39901190  4.33069546  3.73507080  2.95724995  2.40517828
## [145]  2.53171625  2.18618317  2.35358021  4.47598982  6.05870724  8.59143558
## [151] 12.14646401 17.11386104 19.16447563 21.57257776 23.34087839 23.91344171
## [157] 16.77025976 17.19986625 17.56877899 15.79393826 11.56858425  8.45869732
## [163]  5.70491962  3.59404669  2.24175744  2.39124923  1.31836910  0.73685095
## [169] -0.14326220 -0.64938065 -0.44152424 -0.87275647 -0.58852794  1.28759563
## [175]  2.81594992  5.80610719  9.21620789 14.17081280 16.30348368 18.44677644
```

```
## [181] 20.33089819 20.99374585 19.74166789 20.22991176 20.55815482 18.28470287
## [187] 14.80404978 11.49423957  8.61816816  6.59136126  5.23630743  5.39089768
## [193]  4.28192778  3.70436022  2.95217283  2.34229443  2.57000258  2.15411551
## [199]  2.48778377  4.51037240  5.89237297  8.80470338 12.29618206 17.16246338
## [205] 19.38013024 21.70142798 23.34565940 23.99556689 16.54713039 17.37489584
## [211] 17.51108511 15.44227193 11.64489668  8.31355828  5.47575855  3.65026144
## [217]  2.23469022  2.31358540  0.98645889  0.78024320 -0.20698355 -0.77929833
## [223] -0.40978504 -0.84366351 -0.50174913  1.21571784  3.06496633  5.80760647
## [229]  9.43449295 14.26735601 16.57612561 18.78328266 20.25651997 20.93040983
## [235] 19.57037382 20.34193344 20.51411319 18.45450034 14.19375739 11.27783027
## [241]  8.56692529  6.49539423  5.22232432  5.33781314  4.22461708  3.68121517
## [247]  2.89399989  2.29580557  2.61444852  2.14627779  2.38721769  4.82812088
## [253]  6.04676204  8.92461953 12.43383842 17.34016261 19.53325930 21.79394641
## [259] 23.19512043 23.81352234 16.56972060 17.15033488 17.56162324 16.50160546
## [265] 12.14325738  8.94127820  6.17800407  3.82775271  2.44834061  2.30419698
## [271]  1.55695467  0.39699713 -0.09186617 -0.60260607 -0.58921131 -0.70465055
## [277] -0.70324167  1.19955795  2.64591022  4.84459524  8.45030150 13.32755008
## [283] 15.91148672 18.45246196 19.89803634 20.78019691 19.48228965 20.21436553
## [289] 20.69554725 19.25756354 15.03002119 11.95563421  9.14394813  6.83517587
## [295]  5.41041489  5.32007458  4.51129489  3.86602867  2.93916311  2.38743894
## [301]  2.38696978  2.30671487  2.23756847  4.19818299  5.47469611  8.22687368
## [307] 11.54437790 16.40899887 18.86913214 21.54167604 23.00505719 23.79102707
## [313]  9.60455432 10.61055260 11.46858115 10.54325209  9.58873196  7.80611086
## [319]  5.85241280  4.28366084  3.04921370  1.94850349  1.21879210  0.93442183
## [325]  0.78345232  0.70507135  0.53850703 -0.27703729 -0.94089818 -0.16108584
## [331]  1.64101486  4.35210214  6.41483603  8.97756429 12.01562181 13.95936470
## [337] 12.60224071 13.64572244 14.37154574 13.46239934 12.62483830 10.82521343
## [343]  8.72275624  7.26819129  6.05248718  4.93192697  4.19092669  3.86311035
## [349]  3.91377978  3.71059837  3.50512152  2.71325002  2.00193787  2.44786716
## [355]  5.33502893  7.20876536  9.39802942 11.83998723 15.15660419 17.07565098
## [361]  9.71781840 10.49954705 11.46637602 10.57024608  9.58916063  7.87249470
## [367]  5.67706809  4.26432796  3.03564911  1.78963566  1.17043915  0.88216500
## [373]  0.91641359  0.70189371  0.50550990 -0.29137411 -0.96147394 -0.59969139
## [379]  2.08079844  4.53401623  6.36083422  8.93015118 12.21738472 14.12525492
## [385] 12.77377095 13.52220564 14.54769845 13.30745230 12.55889268 10.74582863
## [391]  8.62953291  7.25831409  6.00196342  4.91824460  4.14912834  3.87989788
## [397]  3.91578202  3.70244780  3.46084170  2.65387664  1.97693443  2.88384056
## [403]  5.02798872  7.31934175  9.35985575 12.00453610 15.30212917 17.24766617
## [409]  9.78748169 10.47842889 11.46915812 10.52608609  9.51106143  7.74682732
## [415]  5.55309181  4.23774276  2.99066672  1.82684803  1.13788810  0.89375014
## [421]  0.88665237  0.71744385  0.44923701 -0.32380342 -0.91842632 -0.40096278
## [427]  2.18256254  4.48140583  6.34677439  9.30159359 12.13146224 14.60465708
## [433] 12.84988608 13.48967032 14.31581254 13.42460465 12.45646358 10.76839323
## [439]  8.48446326  7.25046477  5.98418436  4.88537002  4.13469186  3.90351601
## [445]  3.87822118  3.72645345  3.43296328  2.66432691  2.04897923  2.53235941
## [451]  5.33253999  7.65974889  9.31907134 12.68988590 15.30296364 17.59300890
## [457]  9.87805429 10.46167426 11.38716645 10.48733878  9.46929932  7.75642438
## [463]  5.53186946  4.16190205  2.91954098  1.82970071  1.11355891  0.87906117
## [469]  0.87279048  0.72815170  0.42629590 -0.33311987 -0.93095045 -0.15588136
## [475]  2.05099792  4.84860477  6.32118015  9.66753869 12.19017782 14.71892633
## [481] 13.00152936 13.50245552 14.37512178 13.30702686 12.39267874 10.79534383
## [487]  8.42588048  7.19806156  5.94588552  4.88328886  4.12711968  3.84961338
## [493]  3.88173356  3.70341721  3.39901843  2.62827682  2.03012581  3.04303579
## [499]  5.10127110  7.88703260  9.28939848 12.93661432 15.11895647 17.76672115
```

```
## [505]  9.99006573 10.49140911 11.51583964 10.34982869  9.37947389  6.77362107
## [511]  5.40709031  4.16045707  2.87152309  1.81479185  1.15638606  0.91952898
## [517]  0.85552443  0.68278249  0.42911935 -0.44354224 -0.99128286 -0.28302716
## [523]  2.13180858  5.15888271  6.45015300 10.09479009 11.12938506 14.75739230
## [529] 13.06373930 13.68825059 14.47587433 13.11076384 12.31701641 10.60907903
## [535]  8.33554296  7.15533004  5.86559434  4.82146273  4.09972811  3.87592182
## [541]  3.91165190  3.68579531  3.35296130  2.54894161  2.04426217  3.15588865
## [547]  5.27720108  8.17003975  9.31137495 13.27883286 15.16551533 17.87937431
## [553]  9.58542771 10.49999843 11.68077779 10.55883451  9.62724361  7.85443482
## [559]  5.96203413  4.40932345  3.13016663  1.96984839  1.25423136  0.94120150
## [565]  0.56781011  0.70128822  0.59044414 -0.26094499 -0.95739670 -0.34883084
## [571]  1.41103768  4.41435022  6.38377090  9.23292737 11.97542300 13.78925161
## [577] 12.61900516 13.51591668 14.66356945 13.29857750 12.73705206 10.81993732
## [583]  8.87386861  7.39512014  6.18399248  4.96516228  4.21521606  3.89947348
## [589]  3.92801623  3.70767474  3.54489331  2.76250086  2.00664382  2.65425215
## [595]  4.57726264  7.34196610  9.47569079 12.29225926 15.09744587 17.87937431
```

```
# the below statement defines which observations are with or
# not within 1 or negative
df$rangewtc <- ifelse(df$dpdiffwtc >= -1e+06 & df$dpdiffwtc <=
    1, "not", "within")
df$rangewtc <- as.factor(df$rangewtc)  #t
summary(df$rangewtc)  # 80 not in 520 within
```

```
##    not within
##     80    520
```

```
# Count how many data points are comprimised from each
# treatment
sub <- df[df$trt == "ambient", ]
summary(sub$rangewtc)  # 80 outisde out of 220 ambineto observatins or 36 %
```

```
##    not within
##     80    220
```

```
subx <- df[df$trt == "elevated ", ]
summary(subx$rangewtc)  # 0
```

```
##    not within
##      0    300
```

```
## If we run teh scirpt with dew point difference we ommit amy
## values that are less then 1
df$sun <- ifelse(df$PAR >= 0.01, "day", "night")
dfsun <- filter(df, df$sun == "day")
dfnight <- filter(df, df$sun == "night")
dfsun <- subset(dfsun, dtrans > -8 & dtrans < 2)  ## rhis just removes a few absurd values
# Calculate and export daily means
dfsunmean <- dfsun[, c("trt", "Camp", "d18O.corrected_WV", "d18O.corrected_WV.AMB.",
    "dtrans")]
# We can then caluclate the daily mean to compare to source
x <- summaryBy(. ~ trt + Camp, FUN = c(mean, sd), keep.names = T,
```

```
    data = dfsunmean, na.rm = TRUE)
max(dfsun$dtrans, na.rm = TRUE)
```

```
## [1] 0.5880855
```

```
x
```

```
##         trt Camp d18O.corrected_WV.mean d18O.corrected_WV.AMB..mean dtrans.mean
## 1   ambient    1              -12.25919                   -11.98905   -3.662007
## 2   ambient    3              -13.42034                   -11.58399   -3.197766
## 3 elevated     1              -11.81352                   -12.08070   -4.189627
## 4 elevated     3              -12.68523                   -11.59565   -3.040584
##   d18O.corrected_WV.sd d18O.corrected_WV.AMB..sd dtrans.sd
## 1            1.3644977                 1.3267390  1.522695
## 2            0.9573243                 0.3004755  1.463015
## 3            1.1770663                 1.3567382  1.980665
## 4            0.9901106                 0.3061366  1.507050
```

```
# all close to source ~1 mil
write.table(x, "clipboard-16384", sep = "\t", row.names = FALSE)  ## this copies a table of mean daytim
# and below gets the atmsopheric d180
atmx <- summaryBy(. ~ Camp, FUN = c(mean, sd), keep.names = T,
    data = dfsunmean, na.rm = TRUE)
max(dfsun$dtrans, na.rm = TRUE)
```

```
## [1] 0.5880855
```

```
atmx
```

```
##   Camp d18O.corrected_WV.mean d18O.corrected_WV.AMB..mean dtrans.mean
## 1    1              -12.03064                   -12.03605   -3.932582
## 2    3              -13.06512                   -11.58962   -3.121812
##   d18O.corrected_WV.sd d18O.corrected_WV.AMB..sd dtrans.sd
## 1             1.285910                 1.3372074  1.784604
## 2             1.037673                 0.3022535  1.481521
```

```
## I make dfa to use in the code which makes Figures 2,3 and
## 4- These scripts will cal for WTC4 Ispflux model to be run
dfa <- df
```

#Now that the mass balance is done and we have dfa (our campaign data -data frame) we can graph: #To make the graphs we calcuate the mean and std error for each treatment and each campaign. #again it is somewhat repetetive making each panel, just changing variables adn when neccasry highlighting which variables #have dew point and air temp to close #Here is temperture 1= first campaign , 3 = 3rd campaign. there was a 2nd campaign but the LGR was down (blacked out area fig2)

```
wtc <- dfa
Cw <- wtc
Cw <- subset(Cw, Camp == 1)

Cw <- Cw[, c("rangewtc", "Tdh", "chamber", "trt", "Tair_al_mean")]
```

```
Trtmean <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(mean),
    keep.names = T, data = Cw)
se <- function(x) sqrt(var(x)/length(x))
Trtsd <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(se), keep.names = T,
    data = Cw)
data <- cbind(Trtmean, Trtsd)
data <- as.data.frame(data)
data["Time"] <- (dmy_hm(data$Tdh, quiet = TRUE, tz = "UTC"))
str(data)
```

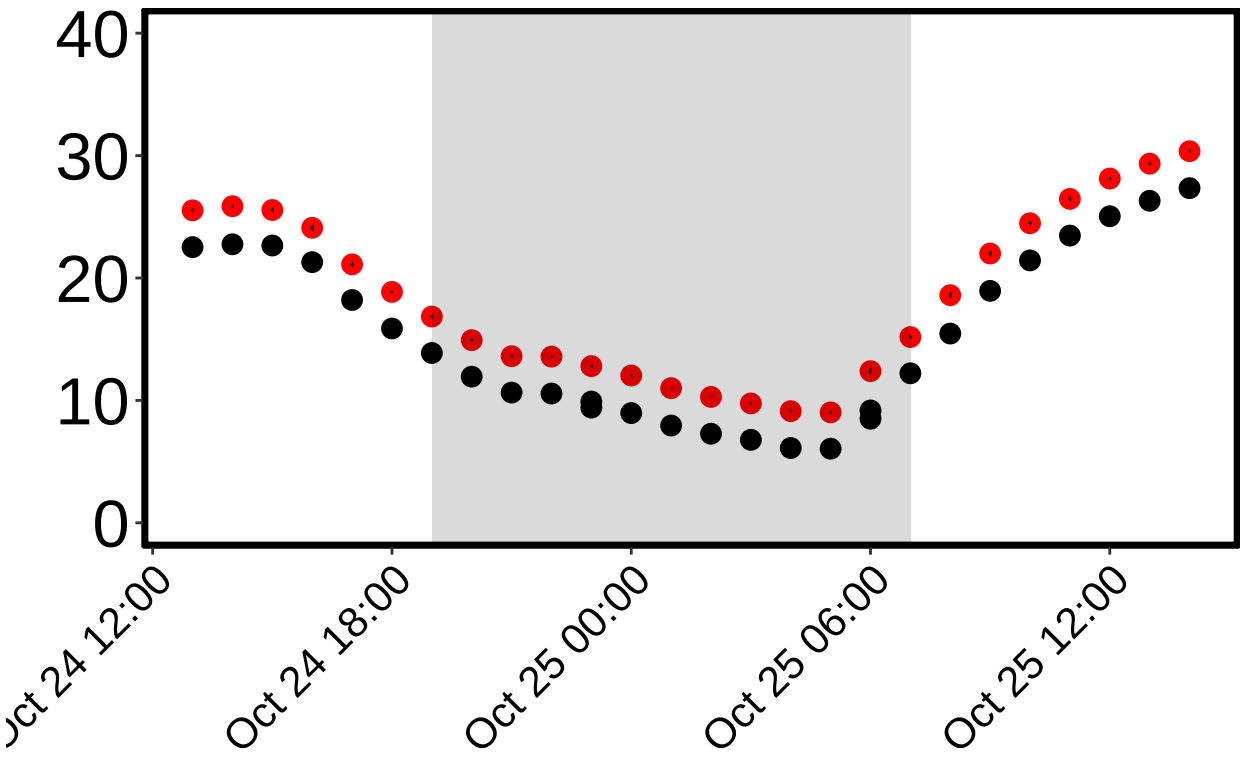
```
## 'data.frame':    54 obs. of  9 variables:
##  $ Tdh           : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 14:00"
##  $ trt           : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc      : Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Tair_al_mean  : num  22.5 25.5 22.8 25.9 22.7 ...
##  $ Tdh.1         : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 14:00"
##  $ trt.1         : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc.1    : Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Tair_al_mean.1: num  0.0342 0.0439 0.0557 0.0237 0.0551 ...
##  $ Time          : POSIXct, format: "2016-10-24 13:00:00" "2016-10-24 13:00:00" ...
```

```
########################
Cwx <- wtc
Cwx <- subset(Cwx, Camp == 3)
Cwx <- Cwx[, c("rangewtc", "Tdh", "chamber", "trt", "Tair_al_mean")]
Trtmeanx <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(mean),
    keep.names = T, data = Cwx)
# Trtmeanx<-filter(Trtmeanx, !grepl('<NA>', trt ))
se <- function(x) sqrt(var(x)/length(x))
Trtsdx <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(se), keep.names = T,
    data = Cwx)
data2 <- cbind(Trtmeanx, Trtsdx)
data2 <- as.data.frame(data2)
data2["Time"] <- (dmy_hm(data2$Tdh, quiet = TRUE, tz = "UTC"))
str(data)
```

```
## 'data.frame':    54 obs. of  9 variables:
##  $ Tdh           : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 14:00"
##  $ trt           : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc      : Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Tair_al_mean  : num  22.5 25.5 22.8 25.9 22.7 ...
##  $ Tdh.1         : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 14:00"
##  $ trt.1         : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc.1    : Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Tair_al_mean.1: num  0.0342 0.0439 0.0557 0.0237 0.0551 ...
##  $ Time          : POSIXct, format: "2016-10-24 13:00:00" "2016-10-24 13:00:00" ...
```

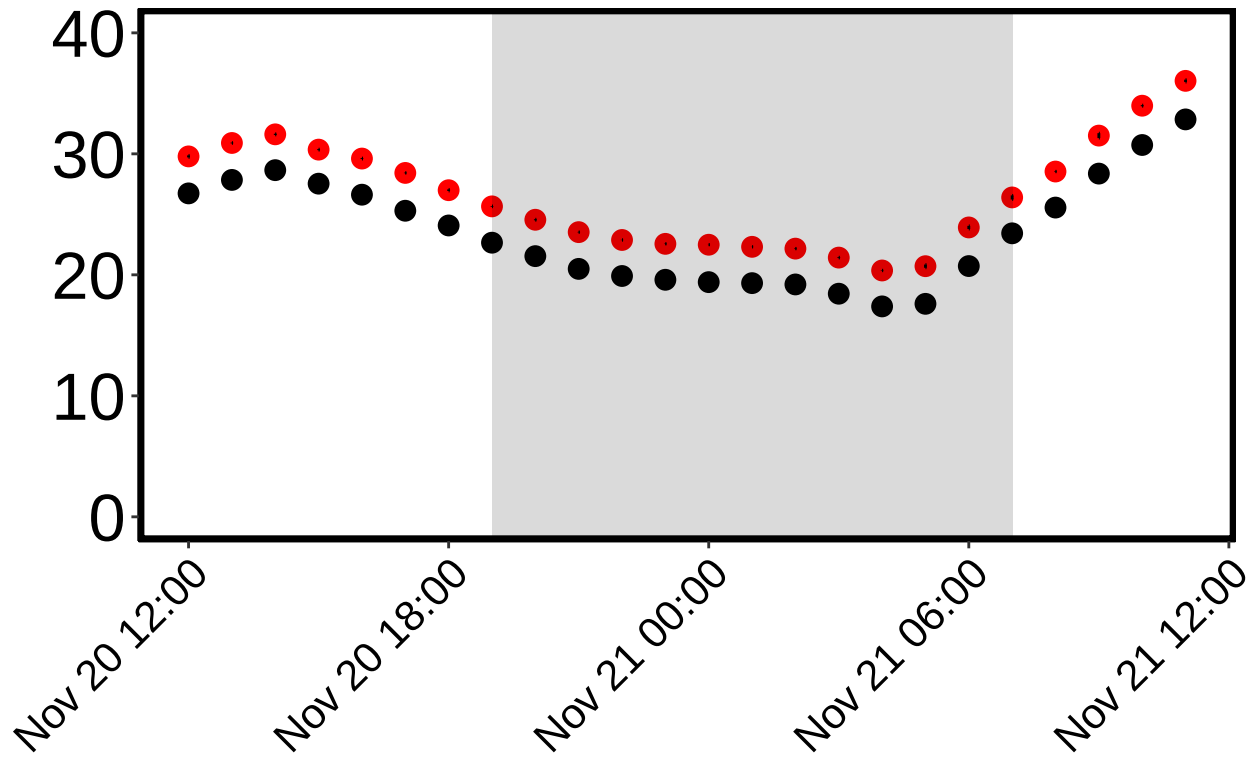
```
start = as.POSIXct("2016-10-24 19:00:00\"", tz = "UTC")
end = as.POSIXct("2016-10-25 07:00:00", tz = "UTC")
start2 = as.POSIXct("2016-11-20 19:00:00\"", tz = "UTC")
end2 = as.POSIXct("2016-11-21 07:00:00", tz = "UTC")
```

```
t1 <- ggplot(data = data, aes(x = Time, y = Tair_al_mean)) +
    geom_point(aes(colour = trt, shape = trt), size = 3.5, ) +
    annotate("rect", fill = "black", alpha = 0.15, xmin = end,
        xmax = start, ymin = -Inf, ymax = Inf) + scale_color_manual(name = "Legend:",
    labels = c("Transpired Ambient Chamber", "Transpired Elevated Chamber"),
    values = c("black", "red")) + scale_shape_manual(name = "Legend:",
    labels = c("in", "out"), values = c(16, 16)) + geom_errorbar(aes(ymin = Tair_al_mean -
    Tair_al_mean.1, ymax = Tair_al_mean + Tair_al_mean.1), width = 0.2,
    position = position_dodge(0.9)) + labs(y = "", x = "", element_text(size = 6)) +
    theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(0, 40)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 16, angle = 45, hjust = 1, vjust = 1))

t1
```




```
t3 <- ggplot(data = data2, aes(x = Time, y = Tair_al_mean)) +
    geom_point(aes(colour = trt, shape = rangewtc), size = 3.5,
        ) + annotate("rect", fill = "black", alpha = 0.15, xmin = end2,
    xmax = start2, ymin = -Inf, ymax = Inf) + scale_color_manual(name = "Legend:",
    labels = c("Transpired Ambient Chamber", "Transpired Elevated Chamber"),
    values = c("black", "red")) + scale_shape_manual(name = "Legend:",
    labels = c("in", "out"), values = c(16, 16)) + geom_errorbar(aes(ymin = Tair_al_mean -
```

```
    Tair_al_mean.1, ymax = Tair_al_mean + Tair_al_mean.1), width = 0.2,
    position = position_dodge(0.9)) + labs(y = "", x = "", element_text(size = 6)) +
    theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(0, 40)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 16, angle = 45, hjust = 1, vjust = 1))
t3
```



#now VPD

```
wtc <- dfa
Cw <- wtc
Cw <- subset(Cw, Camp == 1)

Cw <- Cw[, c("Tdh", "chamber", "trt", "vpd")]
Trtmean <- summaryBy(. ~ Tdh + trt, FUN = c(mean), keep.names = T,
    data = Cw)
# Trtmean<-filter(Trtmean, !grepl('<NA>', trt ));str(Trtmean)
se <- function(x) sqrt(var(x)/length(x))
Trtsd <- summaryBy(. ~ Tdh + trtc, FUN = c(se), keep.names = T,
    data = Cw)
data <- cbind(Trtmean, Trtsd)
data <- as.data.frame(data)
data["Time"] <- (dmy_hm(data$Tdh, quiet = TRUE, tz = "UTC"))
```

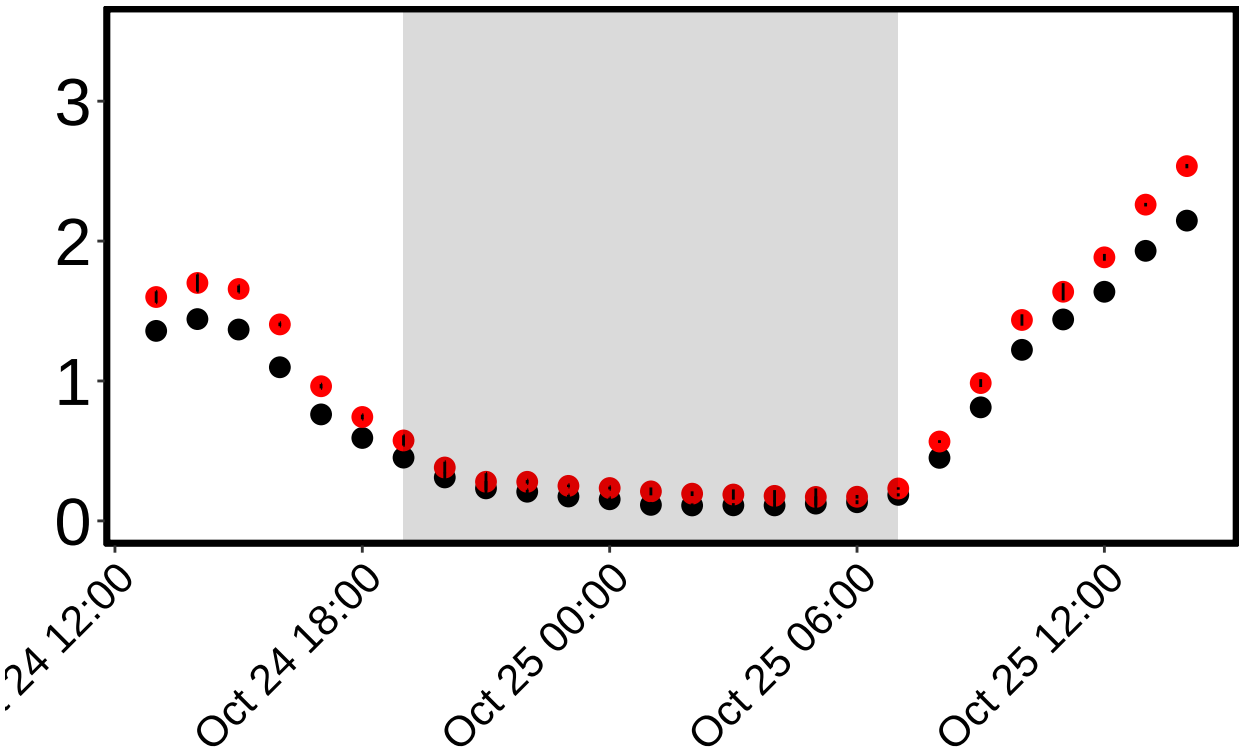
```
str(data)
```

```
## 'data.frame':    52 obs. of  6 variables:
##  $ Tdh  : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 14:00" ...
##  $ trt  : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ vpd  : num  1.36 1.6 1.44 1.7 1.37 ...
##  $ Tdh.1: chr  "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 15:00" "24/10/2016 16:00" ...
##  $ vpd.1: num  0.0384 0.0408 0.0474 0.0612 0.0397 ...
##  $ Time : POSIXct, format: "2016-10-24 13:00:00" "2016-10-24 13:00:00" ...
```

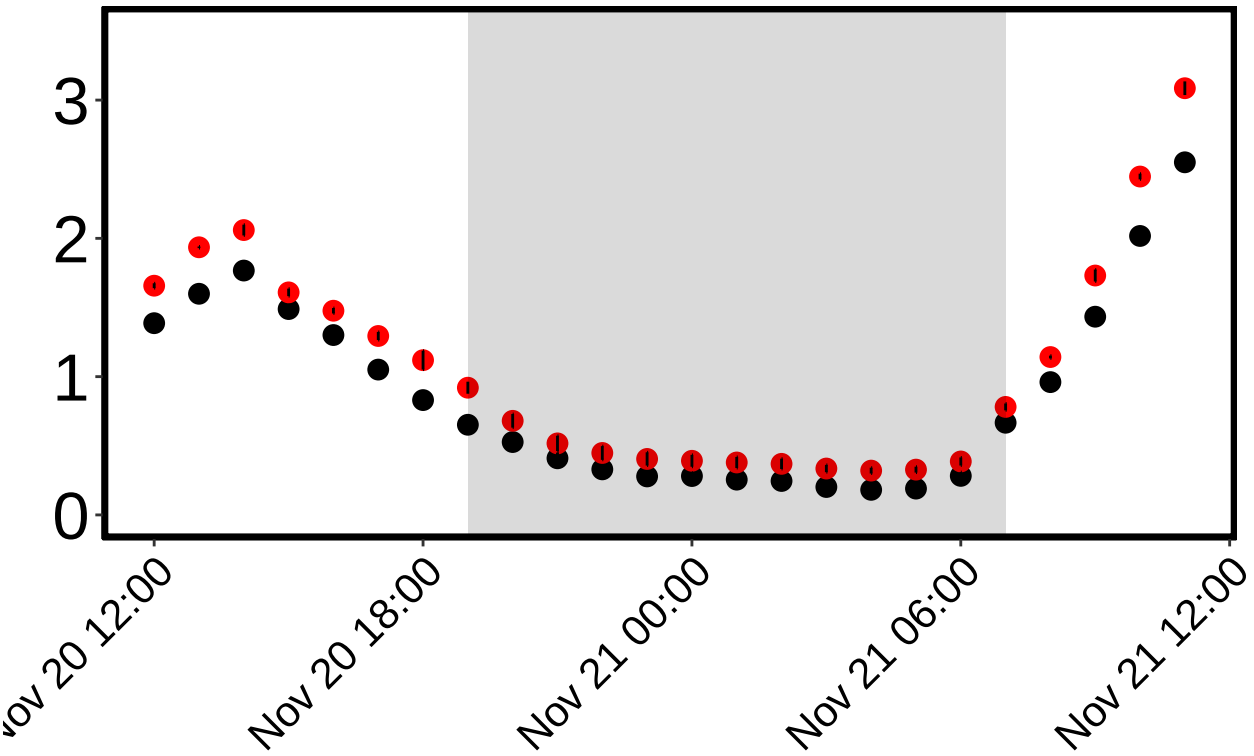
```
#########################
Cwx <- wtc
Cwx <- subset(Cwx, Camp == 3)
Cwx <- Cwx[, c("Tdh", "chamber", "trt", "vpd")]
Trtmeanx <- summaryBy(. ~ Tdh + trt, FUN = c(mean), keep.names = T,
    data = Cwx)
# Trtmeanx<-filter(Trtmeanx, !grepl('<NA>', trt ))
se <- function(x) sqrt(var(x)/length(x))
Trtsdx <- summaryBy(. ~ Tdh + trt, FUN = c(se), keep.names = T,
    data = Cwx)
data2 <- cbind(Trtmeanx, Trtsdx)
data2 <- as.data.frame(data2)
data2["Time"] <- (dmy_hm(data2$Tdh, quiet = TRUE, tz = "UTC"))
str(data)
```

```
## 'data.frame':    52 obs. of  6 variables:
##  $ Tdh  : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 14:00" ...
##  $ trt  : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ vpd  : num  1.36 1.6 1.44 1.7 1.37 ...
##  $ Tdh.1: chr  "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 15:00" "24/10/2016 16:00" ...
##  $ vpd.1: num  0.0384 0.0408 0.0474 0.0612 0.0397 ...
##  $ Time : POSIXct, format: "2016-10-24 13:00:00" "2016-10-24 13:00:00" ...
```

```
start = as.POSIXct("2016-10-24 19:00:00\"", tz = "UTC")
end = as.POSIXct("2016-10-25 07:00:00", tz = "UTC")
start2 = as.POSIXct("2016-11-20 19:00:00\"", tz = "UTC")
end2 = as.POSIXct("2016-11-21 07:00:00", tz = "UTC")
vp1 <- ggplot(data = data, aes(x = Time, y = vpd)) + geom_point(aes(colour = trt,
    shape = trt), size = 3.5, ) + annotate("rect", fill = "black",
    alpha = 0.15, xmin = end, xmax = start, ymin = -Inf, ymax = Inf) +
    scale_color_manual(name = "Legend:", labels = c("Transpired Ambient Chamber",
        "Transpired Elevated Chamber"), values = c("black", "red")) +
    scale_shape_manual(name = "Legend:", labels = c("Transpired Ambient Chamber",
        "Transpired Elevated Chamber"), values = c(16, 16)) +
    geom_errorbar(aes(ymin = vpd - vpd.1, ymax = vpd + vpd.1),
        width = 0.2, position = position_dodge(0.9)) + labs(y = "",
    x = "", element_text(size = 6)) + theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(0, 3.5)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 16, angle = 45, hjust = 1, vjust = 1))
vp1
```




```
vp3 <- ggplot(data = data2, aes(x = Time, y = vpd)) + geom_point(aes(colour = trt,
    shape = trt), size = 3.5, ) + annotate("rect", fill = "black",
    alpha = 0.15, xmin = end2, xmax = start2, ymin = -Inf, ymax = Inf) +
    scale_color_manual(name = "Legend:", labels = c("Transpired Ambient Chamber",
        "Transpired Elevated Chamber"), values = c("black", "red")) +
    scale_shape_manual(name = "Legend:", labels = c("in", "out"),
        values = c(16, 16)) + geom_errorbar(aes(ymin = vpd -
    vpd.1, ymax = vpd + vpd.1), width = 0.2, position = position_dodge(0.9)) +
    labs(y = "", x = "", element_text(size = 6)) + theme_classic() +
    theme(panel.border = element_rect(fill = "NA", colour = "black",
        size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(0, 3.5)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 16, angle = 45, hjust = 1, vjust = 1))
vp3
```



#now Transpiration - here we need to denote which values are comprised (dew point close to air temp) with a *

```
wtc <- dfa
Cw <- wtc
Cw <- subset(Cw, Camp == 1)

Cw <- Cw[, c("rangewtc", "Tdh", "chamber", "trt", "Eleaf")]
Trtmean <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(mean),
    keep.names = T, data = Cw)
# Trtmean<-filter(Trtmean, !grepl('<NA>', trt ));str(Trtmean)
se <- function(x) sqrt(var(x)/length(x))
Trtsd <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(se), keep.names = T,
    data = Cw)
data <- cbind(Trtmean, Trtsd)
data <- as.data.frame(data)
data["Time"] <- (dmy_hm(data$Tdh, quiet = TRUE, tz = "UTC"))
str(data)
```

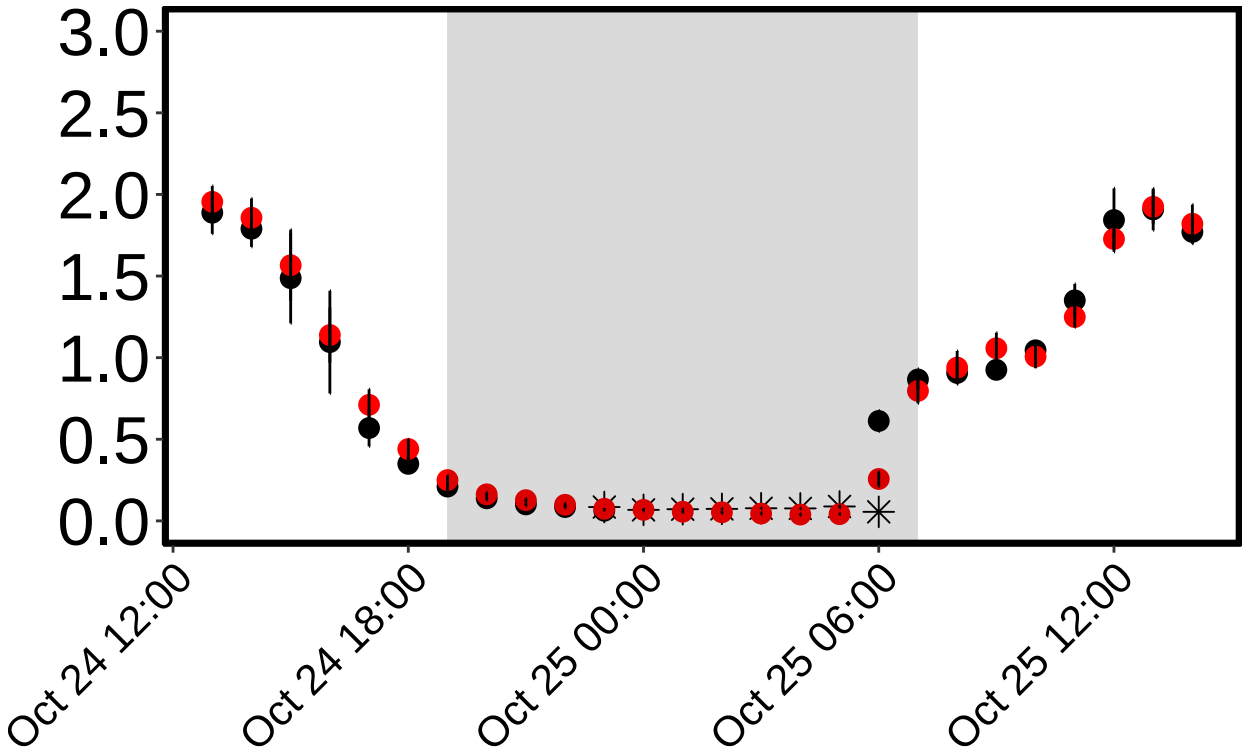
```
## 'data.frame':    54 obs. of  9 variables:
##  $ Tdh       : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 14:00" ...
##  $ trt       : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc  : Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Eleaf     : num  1.89 1.95 1.79 1.86 1.49 ...
##  $ Tdh.1     : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 14:00" ...
##  $ trt.1     : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc.1: Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
```

```
##  $ Eleaf.1   : num  0.1294 0.0973 0.1109 0.1181 0.2767 ...
##  $ Time      : POSIXct, format: "2016-10-24 13:00:00" "2016-10-24 13:00:00" ...
```

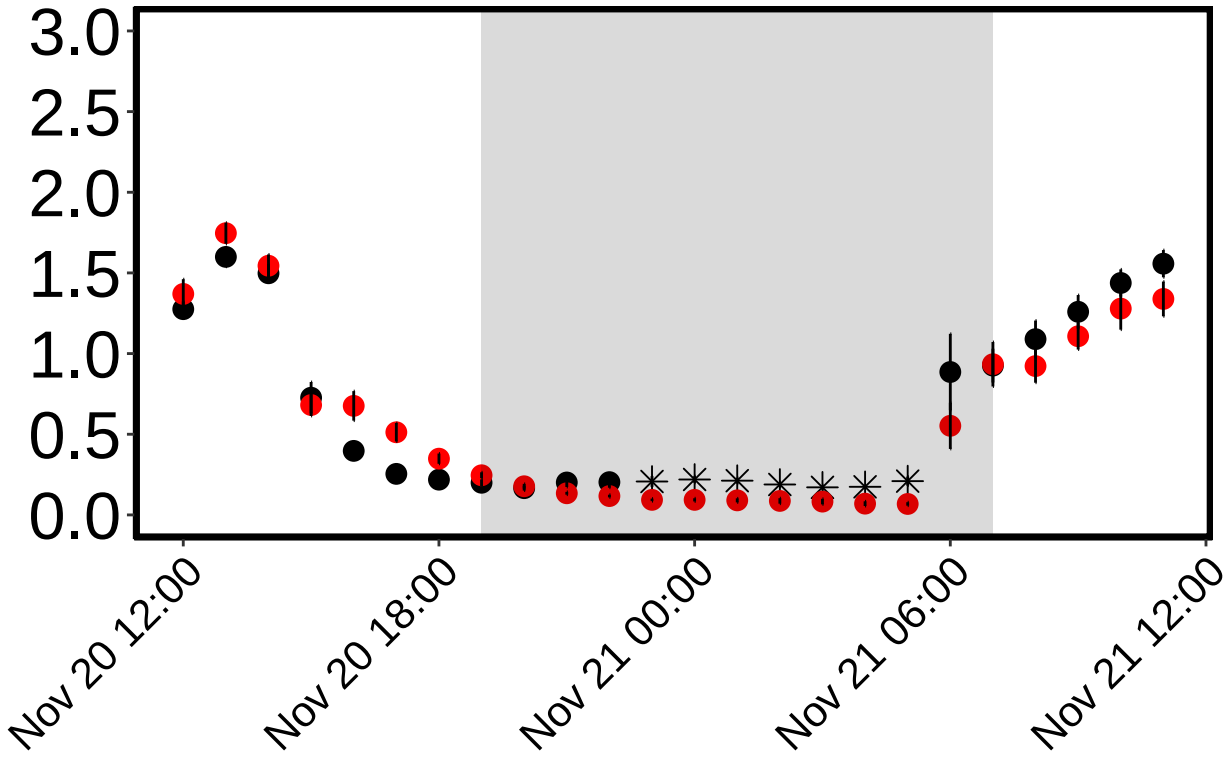
```
#######################
Cwx <- wtc
Cwx <- subset(Cwx, Camp == 3)
Cwx <- Cwx[, c("rangewtc", "Tdh", "chamber", "trt", "Eleaf")]
Trtmeanx <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(mean),
    keep.names = T, data = Cwx)
# Trtmeanx<-filter(Trtmeanx, !grepl('<NA>', trt ))
se <- function(x) sqrt(var(x)/length(x))
Trtsdx <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(se), keep.names = T,
    data = Cwx)
data2 <- cbind(Trtmeanx, Trtsdx)
data2 <- as.data.frame(data2)
data2["Time"] <- (dmy_hm(data2$Tdh, quiet = TRUE, tz = "UTC"))
str(data)
```

```
## 'data.frame':    54 obs. of  9 variables:
##  $ Tdh       : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 14:00" ...
##  $ trt       : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc  : Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Eleaf     : num  1.89 1.95 1.79 1.86 1.49 ...
##  $ Tdh.1     : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 14:00" ...
##  $ trt.1     : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc.1: Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Eleaf.1   : num  0.1294 0.0973 0.1109 0.1181 0.2767 ...
##  $ Time      : POSIXct, format: "2016-10-24 13:00:00" "2016-10-24 13:00:00" ...
```

```
start = as.POSIXct("2016-10-24 19:00:00\"", tz = "UTC")
end = as.POSIXct("2016-10-25 07:00:00", tz = "UTC")
start2 = as.POSIXct("2016-11-20 19:00:00\"", tz = "UTC")
end2 = as.POSIXct("2016-11-21 07:00:00", tz = "UTC")
E1 <- ggplot(data = data, aes(x = Time, y = Eleaf)) + geom_point(aes(colour = trt,
    shape = rangewtc), size = 3.5, ) + annotate("rect", fill = "black",
    alpha = 0.15, xmin = end, xmax = start, ymin = -Inf, ymax = Inf) +

scale_color_manual(name = "Legend:", labels = c("Transpired Ambient Chamber",
    "Transpired Elevated Chamber"), values = c("black", "red")) +
    scale_shape_manual(name = "Legend:", labels = c("in", "out"),
        values = c(8, 16)) + geom_errorbar(aes(ymin = Eleaf -
    Eleaf.1, ymax = Eleaf + Eleaf.1), width = 0.2, position = position_dodge(0.9)) +
    labs(y = "", x = "", element_text(size = 6)) + # scale_x_datetime(breaks = scales::pretty_breaks(n =
theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(0, 3)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 16, angle = 45, hjust = 1, vjust = 1))
E1
```




```
E3 <- ggplot(data = data2, aes(x = Time, y = Eleaf)) + geom_point(aes(colour = trt,
    shape = rangewtc), size = 3.5, ) + annotate("rect", fill = "black",
    alpha = 0.15, xmin = end2, xmax = start2, ymin = -Inf, ymax = Inf) +

scale_color_manual(name = "Legend:", labels = c("Transpired Ambient Chamber",
    "Transpired Elevated Chamber"), values = c("black", "red")) +
    scale_shape_manual(name = "Legend:", labels = c("in", "out"),
        values = c(8, 16)) + geom_errorbar(aes(ymin = Eleaf -
    Eleaf.1, ymax = Eleaf + Eleaf.1), width = 0.2, position = position_dodge(0.9)) +
    labs(y = "", x = "", element_text(size = 6)) + # scale_x_datetime(breaks = scales::pretty_breaks(n
theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(0, 3)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 16, angle = 45, hjust = 1, vjust = 1))

E3
```



#and lastly gs (again with *'s) night time gs should not be that high

```
wtc <- dfa
Cw <- wtc
Cw <- subset(Cw, Camp == 1)

Cw <- Cw[, c("rangewtc", "Tdh", "chamber", "trt", "gs")]
Trtmean <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(mean),
    keep.names = T, na.rm = TRUE, data = Cw)
# Trtmean<-filter(Trtmean, !grepl('<NA>', trt ));str(Trtmean)
se <- function(x) sqrt(var(x)/length(x))
Trtsd <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(se), keep.names = T,
    data = Cw)
data <- cbind(Trtmean, Trtsd)
data <- as.data.frame(data)
data["Time"] <- (dmy_hm(data$Tdh, quiet = TRUE, tz = "UTC"))
str(data)
```

```
## 'data.frame':    54 obs. of  9 variables:
##  $ Tdh       : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 14:00" ...
##  $ trt       : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc  : Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ gs        : num  0.139 0.122 0.124 0.109 0.107 ...
##  $ Tdh.1     : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 14:00" ...
##  $ trt.1     : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc.1: Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
```

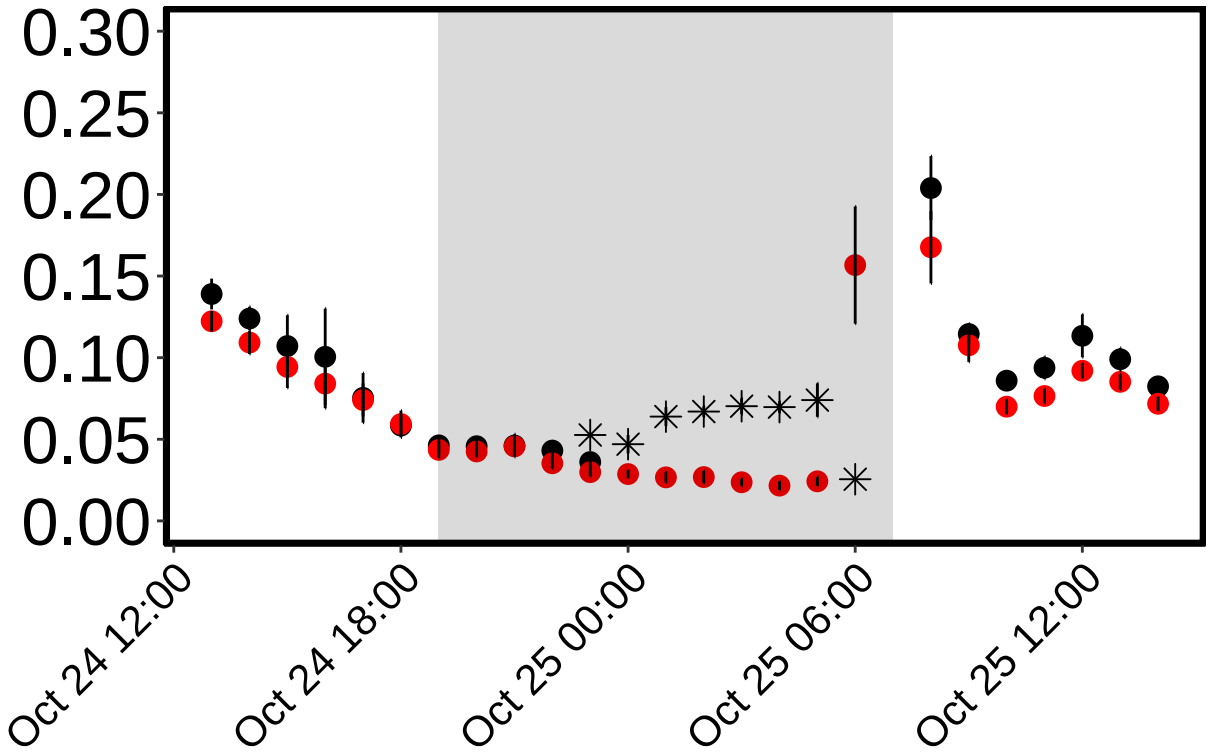
```
##  $ gs.1      : num  0.00943 0.00627 0.0072 0.00702 0.01886 ...
##  $ Time      : POSIXct, format: "2016-10-24 13:00:00" "2016-10-24 13:00:00" ...
```

```
#######################
Cwx <- wtc
Cwx <- subset(Cwx, Camp == 3)
Cwx <- Cwx[, c("rangewtc", "Tdh", "chamber", "trt", "gs")]
Trtmeanx <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(mean),
    keep.names = T, na.rm = TRUE, data = Cwx)
# Trtmeanx<-filter(Trtmeanx, !grepl('<NA>', trt ))
se <- function(x) sqrt(var(x)/length(x))
Trtsdx <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(se), keep.names = T,
    data = Cwx)
data2 <- cbind(Trtmeanx, Trtsdx)
data2 <- as.data.frame(data2)
data2["Time"] <- (dmy_hm(data2$Tdh, quiet = TRUE, tz = "UTC"))
str(data)
```

```
## 'data.frame':    54 obs. of  9 variables:
##  $ Tdh       : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 14:00" ...
##  $ trt       : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc  : Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ gs        : num  0.139 0.122 0.124 0.109 0.107 ...
##  $ Tdh.1     : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 14:00" ...
##  $ trt.1     : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc.1: Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ gs.1      : num  0.00943 0.00627 0.0072 0.00702 0.01886 ...
##  $ Time      : POSIXct, format: "2016-10-24 13:00:00" "2016-10-24 13:00:00" ...
```

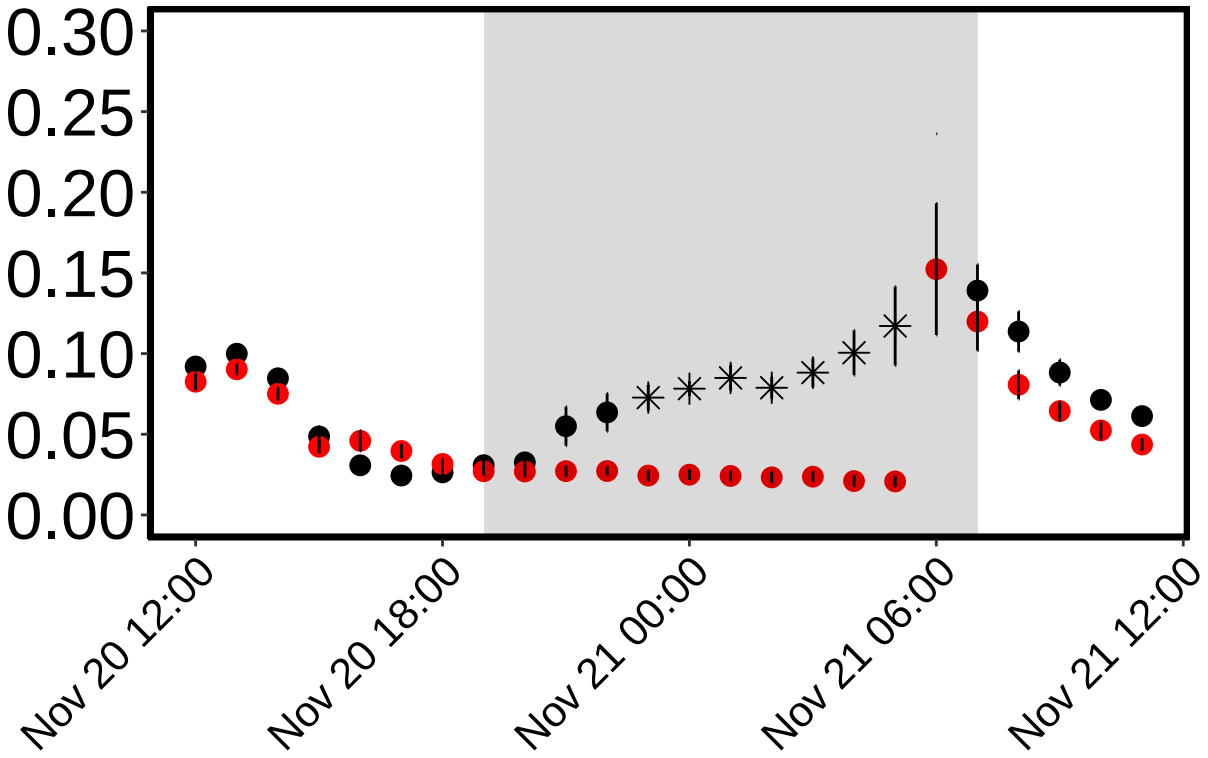
```
start = as.POSIXct("2016-10-24 19:00:00\"", tz = "UTC")
end = as.POSIXct("2016-10-25 07:00:00", tz = "UTC")
start2 = as.POSIXct("2016-11-20 19:00:00\"", tz = "UTC")
end2 = as.POSIXct("2016-11-21 07:00:00", tz = "UTC")

gs1 <- ggplot(data = data, aes(x = Time, y = gs)) + geom_point(aes(colour = trt,
    shape = rangewtc), size = 3.5, ) + annotate("rect", fill = "black",
    alpha = 0.15, xmin = end, xmax = start, ymin = -Inf, ymax = Inf) +

scale_color_manual(name = "Legend:", labels = c("Transpired Ambient Chamber",
    "Transpired Elevated Chamber"), values = c("black", "red")) +
    scale_shape_manual(name = "Legend:", labels = c("in", "out"),
        values = c(8, 16)) + geom_errorbar(aes(ymin = gs - gs.1,
    ymax = gs + gs.1), width = 0.2, position = position_dodge(0.9)) +
    labs(y = "", x = "", element_text(size = 6)) + # scale_x_datetime(breaks = scales::pretty_breaks(n
theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(0, 0.3)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 16, angle = 45, hjust = 1, vjust = 1))
gs1
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```




```
gs3 <- ggplot(data = data2, aes(x = Time, y = gs)) + geom_point(aes(colour = trt,
    shape = rangewtc), size = 3.5, ) + annotate("rect", fill = "black",
    alpha = 0.15, xmin = end2, xmax = start2, ymin = -Inf, ymax = Inf) +

scale_color_manual(name = "Legend:", labels = c("Transpired Ambient Chamber",
    "Transpired Elevated Chamber"), values = c("black", "red")) +
    scale_shape_manual(name = "Legend:", labels = c("in", "out"),
        values = c(8, 16)) + geom_errorbar(aes(ymin = gs - gs.1,
    ymax = gs + gs.1), width = 0.2, position = position_dodge(0.9)) +
    labs(y = "", x = "", element_text(size = 6)) + # scale_x_datetime(breaks = scales::pretty_breaks(n
theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(0, 0.3)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 16, angle = 45, hjust = 1, vjust = 1))
gs3
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



#We now just merge these diel obsevrations into one big figure

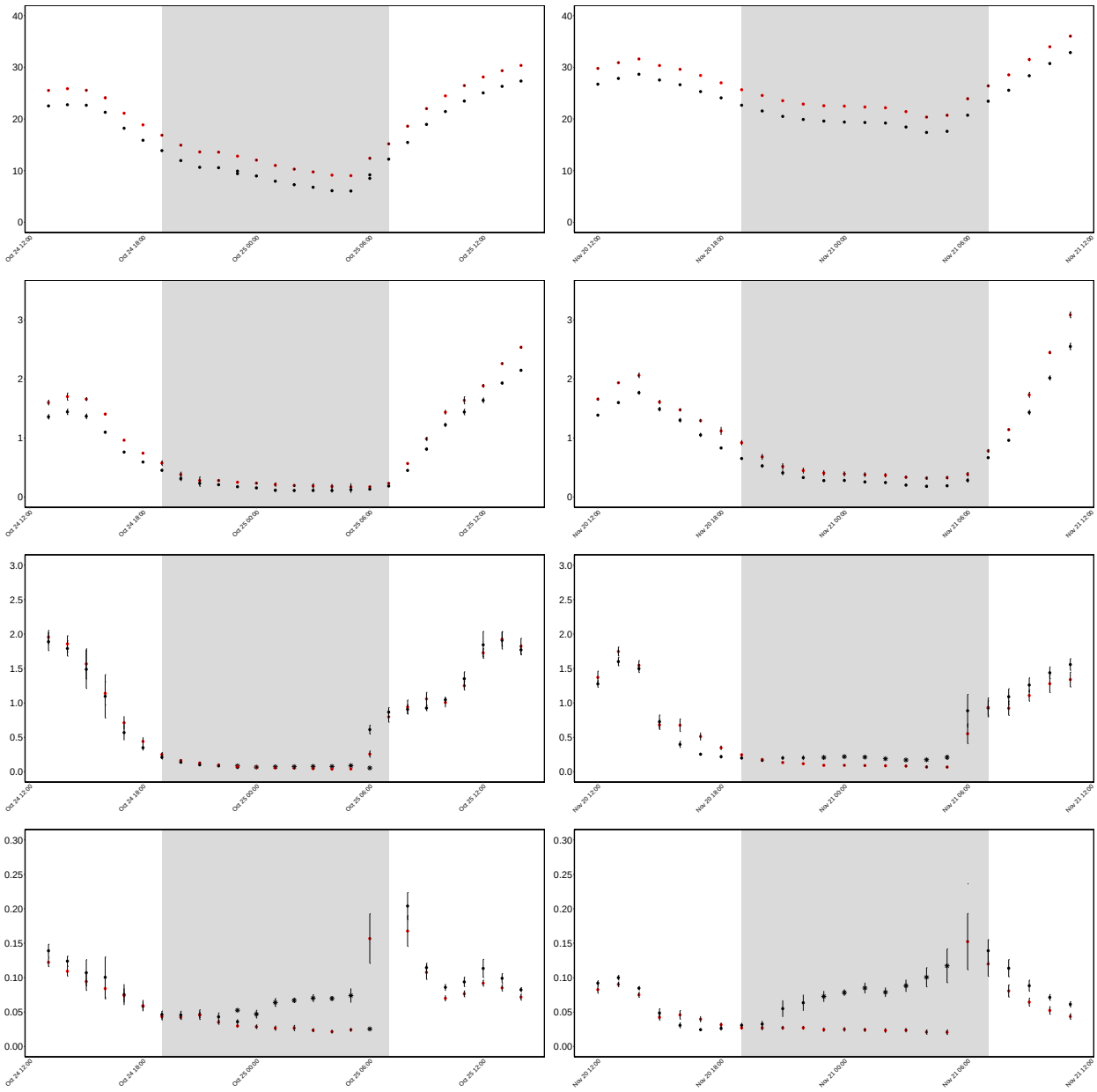
```
## FIgure 2 enviro vars
Figure2 = plot_grid(t1, t3, vp1, vp3, E1, E3, gs1, gs3, ncol = 2,
    align = "hv")
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```

```
# and then ggsave it out -- I added axis labels etc later in
# adobe illustator
```

```
plot(Figure2)
```



#We continue with dfa to make the next figure (but we can clean the environment again)

```
keep(dfa, sure = TRUE)
```

#Figure 4 starts with leaf water turnover time and isotipic leaf water turnover time #the * filter is applied to all plots in Figure 4

```
wtc <- dfa
Cw <- wtc
Cw <- subset(Cw, Camp == 1)

Cw <- Cw[, c("rangewtc", "Tdh", "chamber", "trt", "th")]
Trtmean <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(mean),
    keep.names = T, data = Cw)
```

```
# Trtmean<-filter(Trtmean, !grepl('<NA>', trt ));str(Trtmean)
se <- function(x) sqrt(var(x)/length(x))
Trtsd <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(se), keep.names = T,
    data = Cw)
data <- cbind(Trtmean, Trtsd)
data <- as.data.frame(data)
data["Time"] <- (dmy_hm(data$Tdh, quiet = TRUE, tz = "UTC"))
str(data)
```

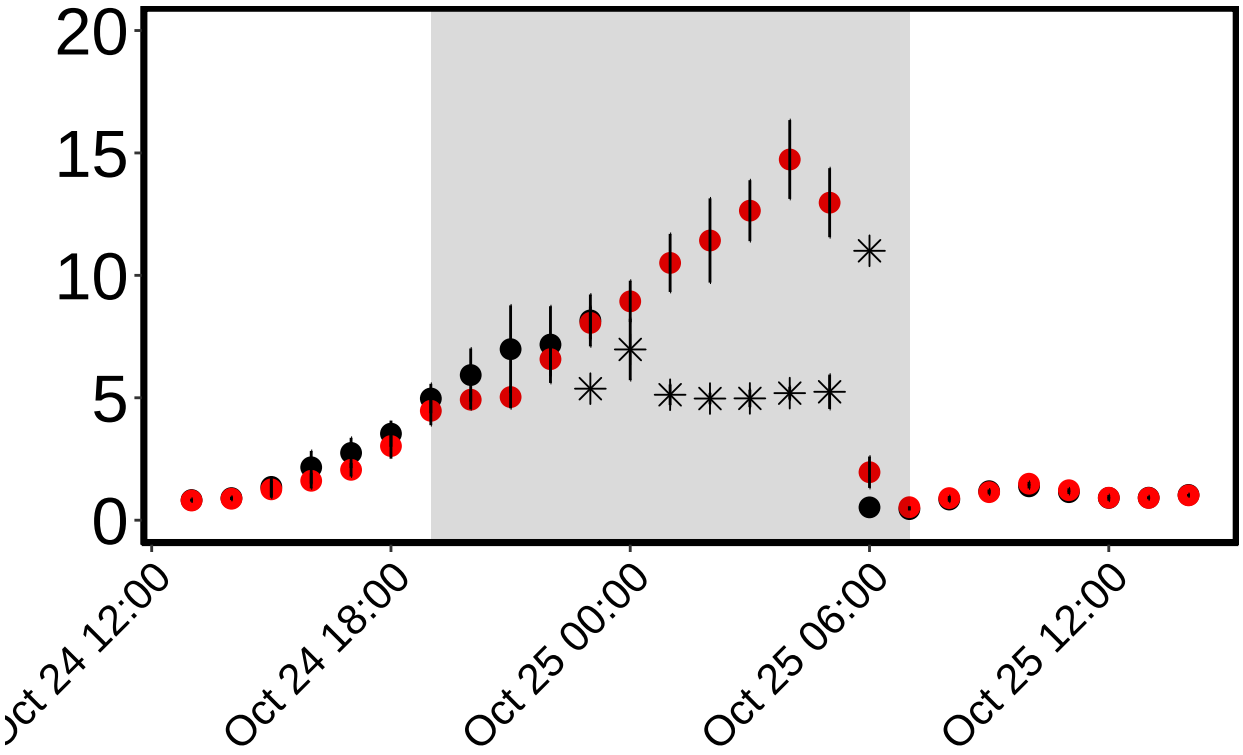
```
## 'data.frame':    54 obs. of  9 variables:
##  $ Tdh       : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 14:00" ...
##  $ trt       : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc  : Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ th        : num  0.822 0.803 0.903 0.876 1.36 ...
##  $ Tdh.1     : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 14:00" ...
##  $ trt.1     : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc.1: Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ th.1      : num  0.058 0.0299 0.0573 0.0498 0.3587 ...
##  $ Time      : POSIXct, format: "2016-10-24 13:00:00" "2016-10-24 13:00:00" ...
```

```
########################
Cwx <- wtc
Cwx <- subset(Cwx, Camp == 3)
Cwx <- Cwx[, c("rangewtc", "Tdh", "chamber", "trt", "th")]
Trtmeanx <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(mean),
    keep.names = T, data = Cwx)
# Trtmeanx<-filter(Trtmeanx, !grepl('<NA>', trt ))
se <- function(x) sqrt(var(x)/length(x))
Trtsdx <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(se), keep.names = T,
    data = Cwx)
data2 <- cbind(Trtmeanx, Trtsdx)
data2 <- as.data.frame(data2)
data2["Time"] <- (dmy_hm(data2$Tdh, quiet = TRUE, tz = "UTC"))
str(data)
```

```
## 'data.frame':    54 obs. of  9 variables:
##  $ Tdh       : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 14:00" ...
##  $ trt       : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc  : Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ th        : num  0.822 0.803 0.903 0.876 1.36 ...
##  $ Tdh.1     : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 14:00" ...
##  $ trt.1     : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc.1: Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ th.1      : num  0.058 0.0299 0.0573 0.0498 0.3587 ...
##  $ Time      : POSIXct, format: "2016-10-24 13:00:00" "2016-10-24 13:00:00" ...
```

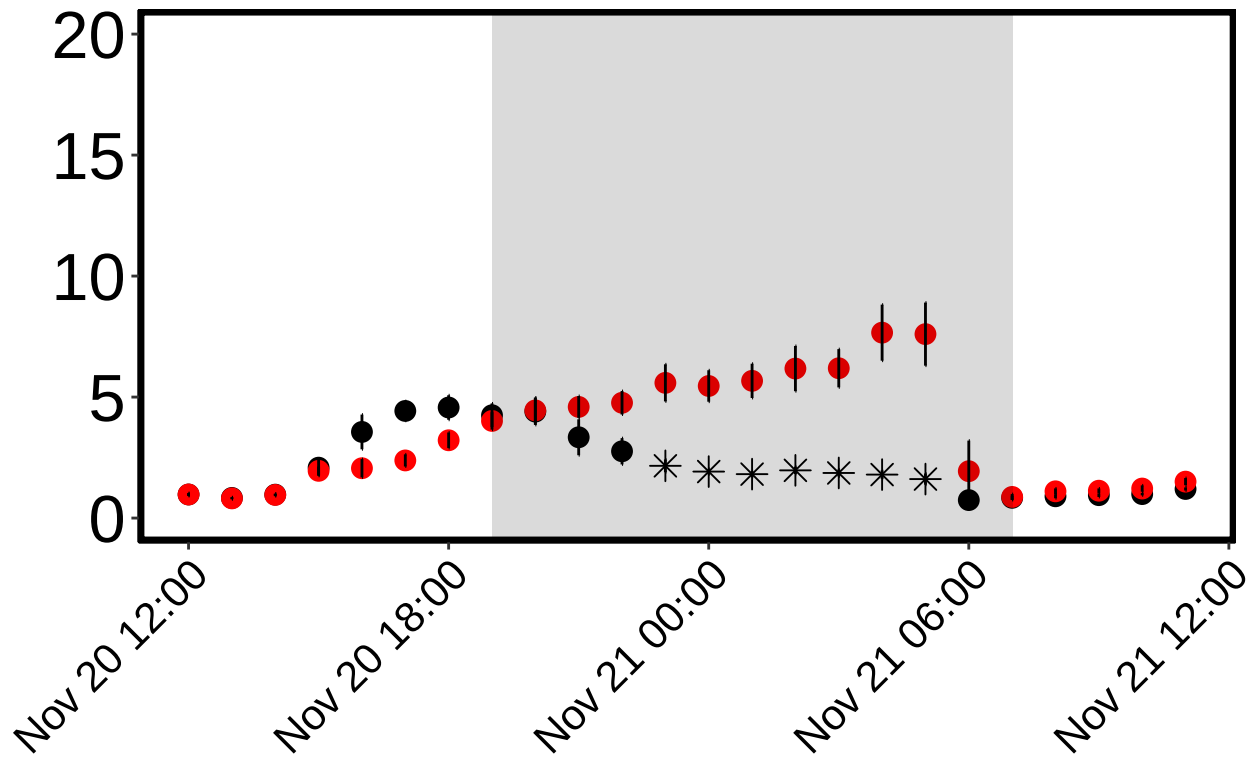
```
start = as.POSIXct("2016-10-24 19:00:00\"", tz = "UTC")
end = as.POSIXct("2016-10-25 07:00:00", tz = "UTC")
start2 = as.POSIXct("2016-11-20 19:00:00\"", tz = "UTC")
end2 = as.POSIXct("2016-11-21 07:00:00", tz = "UTC")
th1 <- ggplot(data = data, aes(x = Time, y = th)) + geom_point(aes(colour = trt,
    shape = rangewtc), size = 3.5, ) + annotate("rect", fill = "black",
```

```
    alpha = 0.15, xmin = end, xmax = start, ymin = -Inf, ymax = Inf) +
    scale_color_manual(name = "Legend:", labels = c("Transpired Ambient Chamber",
        "Transpired Elevated Chamber"), values = c("black", "red")) +
    scale_shape_manual(name = "Legend:", labels = c("in", "out"),
        values = c(8, 16)) + geom_errorbar(aes(ymin = th - th.1,
    ymax = th + th.1), width = 0.2, position = position_dodge(0.9)) +
    labs(y = "", x = "", element_text(size = 6)) + # scale_x_datetime(breaks = scales::pretty_breaks(n =
theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(0, 20)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 16, angle = 45, hjust = 1, vjust = 1))
th1
```




```
th3 <- ggplot(data = data2, aes(x = Time, y = th)) + geom_point(aes(colour = trt,
    shape = rangewtc), size = 3.5, ) + annotate("rect", fill = "black",
    alpha = 0.15, xmin = end2, xmax = start2, ymin = -Inf, ymax = Inf) +

scale_color_manual(name = "Legend:", labels = c("Transpired Ambient Chamber",
    "Transpired Elevated Chamber"), values = c("black", "red")) +
    scale_shape_manual(name = "Legend:", labels = c("in", "out"),
        values = c(8, 16)) + geom_errorbar(aes(ymin = th - th.1,
    ymax = th + th.1), width = 0.2, position = position_dodge(0.9)) +
    labs(y = "", x = "", element_text(size = 6)) + # scale_x_datetime(breaks = scales::pretty_breaks(n =
```

```
theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(0, 20)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 16, angle = 45, hjust = 1, vjust = 1))
th3
```



#as above but with isotopic leaf water turnover

```
wtc <- dfa
Cw <- wtc
Cw <- subset(Cw, Camp == 1)

Cw <- Cw[, c("rangewtc", "Tdh", "chamber", "trt", "thiso")]
Trtmean <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(mean),
    keep.names = T, data = Cw)
# Trtmean<-filter(Trtmean, !grepl('<NA>', trt ));str(Trtmean)
se <- function(x) sqrt(var(x)/length(x))
Trtsd <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(se), keep.names = T,
    data = Cw)
data <- cbind(Trtmean, Trtsd)
data <- as.data.frame(data)
data["Time"] <- (dmy_hm(data$Tdh, quiet = TRUE, tz = "UTC"))
str(data)
```

```
## 'data.frame':    54 obs. of  9 variables:
```

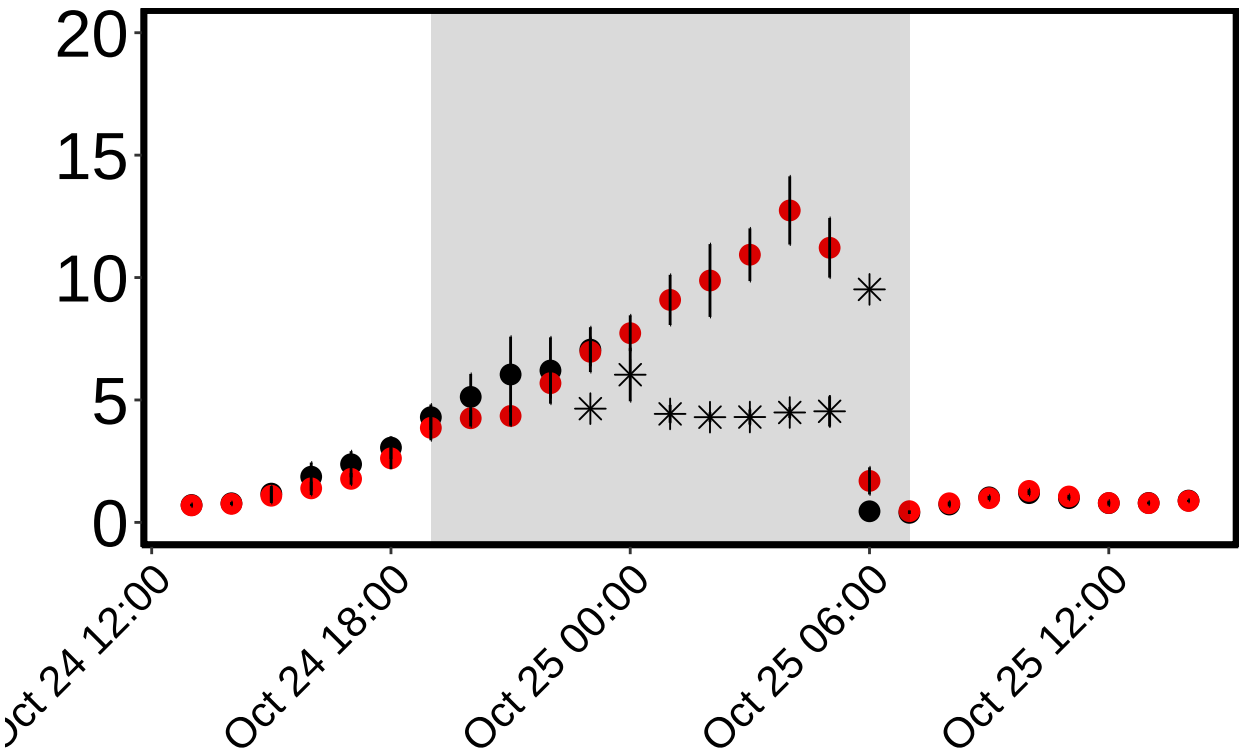
```
##  $ Tdh       : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 14:00" ...
##  $ trt       : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc  : Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ thiso     : num  0.71 0.694 0.78 0.757 1.175 ...
##  $ Tdh.1     : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 14:00" ...
##  $ trt.1     : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc.1: Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ thiso.1   : num  0.0501 0.0258 0.0495 0.043 0.3098 ...
##  $ Time      : POSIXct, format: "2016-10-24 13:00:00" "2016-10-24 13:00:00" ...
```

```
#########################
Cwx <- wtc
Cwx <- subset(Cwx, Camp == 3)
Cwx <- Cwx[, c("rangewtc", "Tdh", "chamber", "trt", "thiso")]
Trtmeanx <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(mean),
    keep.names = T, data = Cwx)
# Trtmeanx<-filter(Trtmeanx, !grepl('<NA>', trt ))
se <- function(x) sqrt(var(x)/length(x))
Trtsdx <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(se), keep.names = T,
    data = Cwx)
data2 <- cbind(Trtmeanx, Trtsdx)
data2 <- as.data.frame(data2)
data2["Time"] <- (dmy_hm(data2$Tdh, quiet = TRUE, tz = "UTC"))
str(data)
```

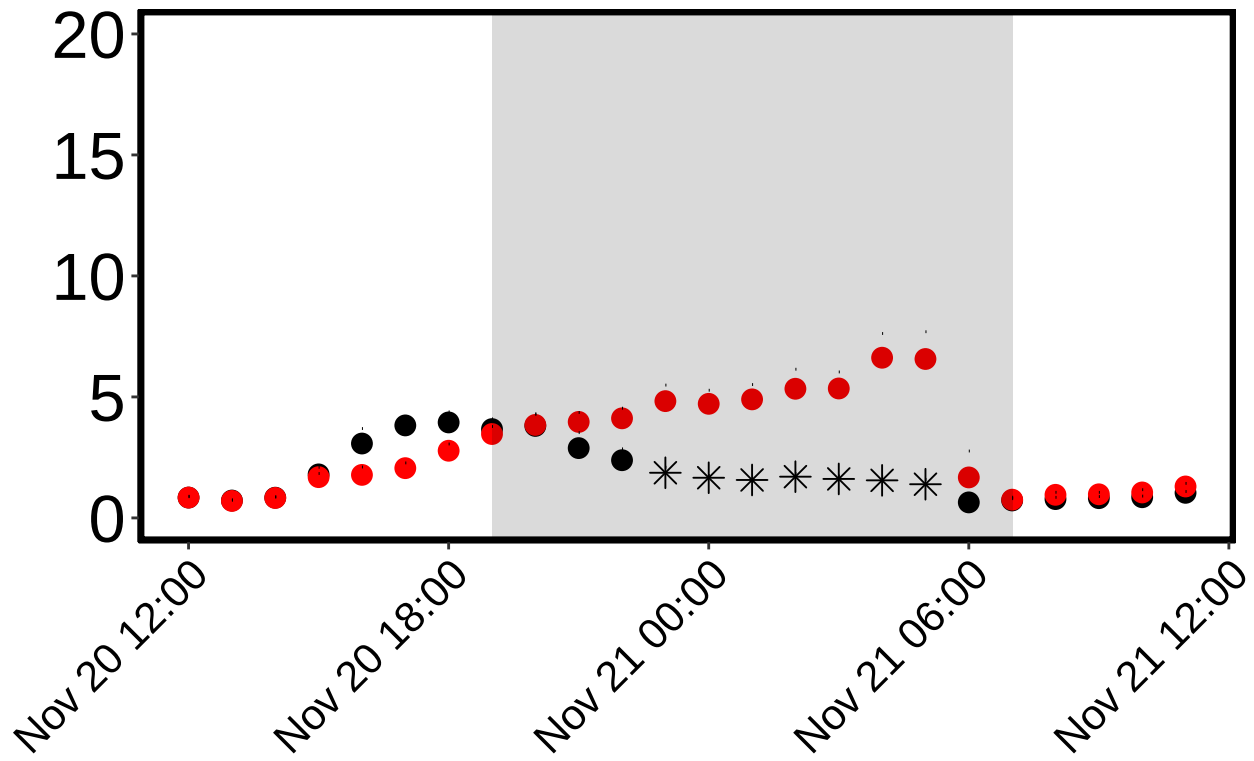
```
## 'data.frame':    54 obs. of  9 variables:
##  $ Tdh       : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 14:00" ...
##  $ trt       : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc  : Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ thiso     : num  0.71 0.694 0.78 0.757 1.175 ...
##  $ Tdh.1     : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 14:00" ...
##  $ trt.1     : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc.1: Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ thiso.1   : num  0.0501 0.0258 0.0495 0.043 0.3098 ...
##  $ Time      : POSIXct, format: "2016-10-24 13:00:00" "2016-10-24 13:00:00" ...
```

```
start = as.POSIXct("2016-10-24 19:00:00\"", tz = "UTC")
end = as.POSIXct("2016-10-25 07:00:00", tz = "UTC")
start2 = as.POSIXct("2016-11-20 19:00:00\"", tz = "UTC")
end2 = as.POSIXct("2016-11-21 07:00:00", tz = "UTC")

thi1 <- ggplot(data = data, aes(x = Time, y = thiso)) + geom_point(aes(colour = trt,
    shape = rangewtc), size = 3.5, ) + annotate("rect", fill = "black",
    alpha = 0.15, xmin = end, xmax = start, ymin = -Inf, ymax = Inf) +
    scale_color_manual(name = "Legend:", labels = c("Transpired Ambient Chamber",
        "Transpired Elevated Chamber"), values = c("black", "red")) +
    scale_shape_manual(name = "Legend:", labels = c("in", "out"),
        values = c(8, 16)) + geom_errorbar(aes(ymin = thiso -
    thiso.1, ymax = thiso + thiso.1), width = 0.2, position = position_dodge(0.9)) +
    labs(y = "", x = "", element_text(size = 6)) +
theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
```

```
        limits = c(0, 20)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 16, angle = 45, hjust = 1, vjust = 1))
thi1
```




```
thi3 <- ggplot(data = data2, aes(x = Time, y = thiso)) + geom_point(aes(colour = trt,
    shape = rangewtc), size = 3.5, ) + annotate("rect", fill = "black",
    alpha = 0.15, xmin = end2, xmax = start2, ymin = -Inf, ymax = Inf) +
    scale_color_manual(name = "Legend:", labels = c("Transpired Ambient Chamber",
        "Transpired Elevated Chamber"), values = c("black", "red")) +
    scale_shape_manual(name = "Legend:", labels = c("in", "out"),
        values = c(8, 16)) + geom_errorbar(aes(ymin = -thiso -
    thiso.1, ymax = thiso + thiso.1), width = 0.2, position = position_dodge(0.9)) +
    labs(y = "", x = "", element_text(size = 6)) + theme_classic() +
    theme(panel.border = element_rect(fill = "NA", colour = "black",
        size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(0, 20)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 16, angle = 45, hjust = 1, vjust = 1))
thi3
```



#the next panels differ slighlty in that d18oatm needs to be overlayed in black #Here is campaign 1

```
wtc <- dfa
Cw <- wtc
Cw <- subset(Cw, Camp == 1)

Cw <- Cw[, c("rangewtc", "Tdh", "chamber", "trt", "d18O.corrected_WV")]
Trtmean <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(mean),
    keep.names = T, data = Cw)
se <- function(x) sqrt(var(x)/length(x))
Trtsd <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(se), keep.names = T,
    data = Cw)
data <- cbind(Trtmean, Trtsd)
data <- as.data.frame(data)
data["Time"] <- (dmy_hm(data$Tdh, quiet = TRUE, tz = "UTC"))
str(data)
```

```
## 'data.frame':    54 obs. of  9 variables:
##  $ Tdh                : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 14
##  $ trt                : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc           : Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ d18O.corrected_WV  : num  -12.5 -11.9 -13 -12.1 -12.4 ...
##  $ Tdh.1              : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 14
##  $ trt.1              : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc.1         : Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ d18O.corrected_WV.1: num  0.157 0.251 0.404 0.445 0.658 ...
##  $ Time               : POSIXct, format: "2016-10-24 13:00:00" "2016-10-24 13:00:00" ...
```

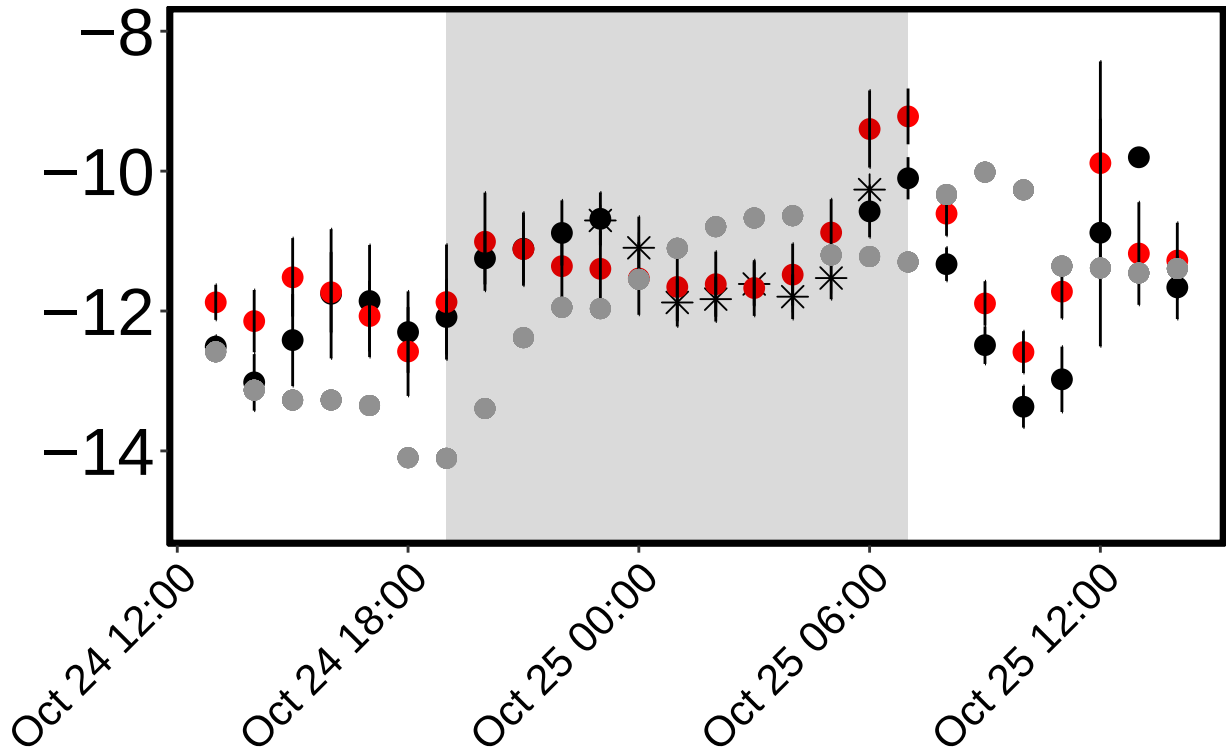
```
######################
Cwx <- wtc
Cwx <- subset(Cwx, Camp == 1)
Cwx <- Cwx[, c("Tdh", "chamber", "trt", "d18O.corrected_WV.AMB.")]
Trtmeanx <- summaryBy(. ~ Tdh, FUN = c(mean), keep.names = T,
    data = Cwx)
se <- function(x) sqrt(var(x)/length(x))
Trtsdx <- summaryBy(. ~ Tdh + chamber, FUN = c(se), keep.names = T,
    data = Cwx)
data2 <- cbind(Trtmeanx, Trtsdx)
```

```
## Warning in data.frame(..., check.names = FALSE): row names were found from a
## short variable and have been discarded
```

```
data2 <- as.data.frame(data2)
data2["Time"] <- (dmy_hm(data2$Tdh, quiet = TRUE, tz = "UTC"))
str(data)
```

```
## 'data.frame':    54 obs. of  9 variables:
##  $ Tdh                : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 14
##  $ trt                : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc           : Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ d18O.corrected_WV  : num  -12.5 -11.9 -13 -12.1 -12.4 ...
##  $ Tdh.1              : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 14
##  $ trt.1              : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc.1         : Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ d18O.corrected_WV.1: num  0.157 0.251 0.404 0.445 0.658 ...
##  $ Time               : POSIXct, format: "2016-10-24 13:00:00" "2016-10-24 13:00:00" ...
```

```
start = as.POSIXct("2016-10-24 19:00:00\"", tz = "UTC")
end = as.POSIXct("2016-10-25 07:00:00", tz = "UTC")

a <- ggplot(data = data, aes(x = Time, y = d18O.corrected_WV)) +
    geom_point(aes(colour = trt, shape = rangewtc), size = 3.5,
        ) +
annotate("rect", fill = "black", alpha = 0.15, xmin = end, xmax = start,
    ymin = -Inf, ymax = Inf) +
scale_color_manual(name = "Legend:", labels = c("Transpired Ambient Chamber",
    "Transpired Elevated Chamber"), values = c("black", "red")) +
    scale_shape_manual(name = "Legend:", labels = c("in", "out"),
        values = c(8, 16)) + geom_errorbar(aes(ymin = d18O.corrected_WV -
    d18O.corrected_WV.1, ymax = d18O.corrected_WV + d18O.corrected_WV.1),
    width = 0.2, position = position_dodge(0.9)) + labs(y = "",
    x = "", element_text(size = 6)) + theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(-15, -8)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 16, angle = 45, hjust = 1, vjust = 1))
iso1 <- a + geom_point(data = data2, mapping = aes(x = Time,
    y = d18O.corrected_WV.AMB.), size = 3, color = "gray57")

iso1
```



#and campaign 2

```
wtc <- dfa
Cw <- wtc
Cw <- subset(Cw, Camp == 3)
Cw <- Cw[, c("rangewtc", "Tdh", "chamber", "trt", "d18O.corrected_WV")]
Trtmean <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(mean),
    keep.names = T, data = Cw)
se <- function(x) sqrt(var(x)/length(x))
Trtsd <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(se), keep.names = T,
    data = Cw)
data <- cbind(Trtmean, Trtsd)
data <- as.data.frame(data)
data["Time"] <- (dmy_hm(data$Tdh, quiet = TRUE, tz = "UTC"))
str(data)
```

```
## 'data.frame':    48 obs. of  9 variables:
##  $ Tdh                : chr  "20/11/2016 12:00" "20/11/2016 12:00" "20/11/2016 13:00" "20/11/2016 13
##  $ trt                : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc           : Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ d18O.corrected_WV  : num  -14.4 -13.3 -14.6 -13.8 -14.1 ...
##  $ Tdh.1              : chr  "20/11/2016 12:00" "20/11/2016 12:00" "20/11/2016 13:00" "20/11/2016 13
##  $ trt.1              : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc.1         : Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ d18O.corrected_WV.1: num  0.112 0.2326 0.0966 0.168 0.2119 ...
##  $ Time               : POSIXct, format: "2016-11-20 12:00:00" "2016-11-20 12:00:00" ...
```

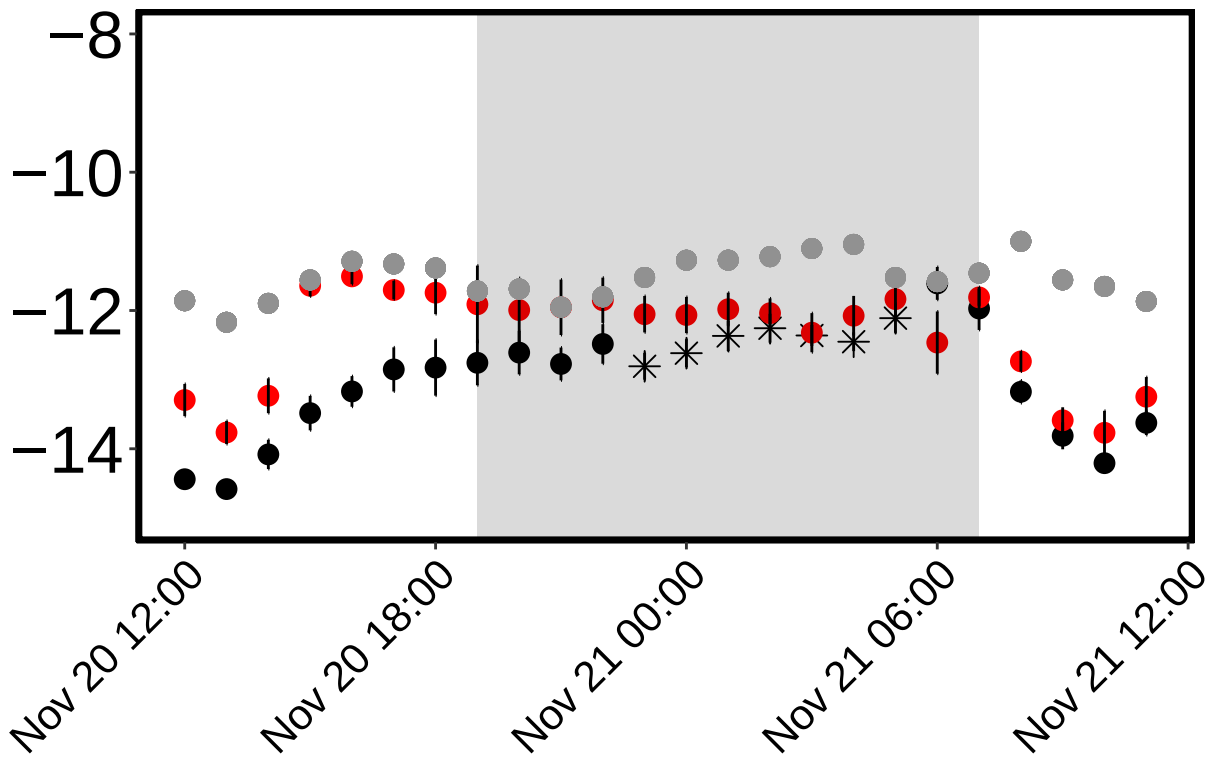
```
#######################
Cwx <- wtc
Cwx <- subset(Cwx, Camp == 3)
Cwx <- Cwx[, c("Tdh", "chamber", "trt", "d18O.corrected_WV.AMB.")]
Trtmeanx <- summaryBy(. ~ Tdh, FUN = c(mean), keep.names = T,
    data = Cwx)
se <- function(x) sqrt(var(x)/length(x))
Trtsdx <- summaryBy(. ~ Tdh + chamber, FUN = c(se), keep.names = T,
    data = Cwx)
data2 <- cbind(Trtmeanx, Trtsdx)
```

```
## Warning in data.frame(..., check.names = FALSE): row names were found from a
## short variable and have been discarded
```

```
data2 <- as.data.frame(data2)
data2["Time"] <- (dmy_hm(data2$Tdh, quiet = TRUE, tz = "UTC"))
str(data)
```

```
## 'data.frame':    48 obs. of  9 variables:
##  $ Tdh                : chr  "20/11/2016 12:00" "20/11/2016 12:00" "20/11/2016 13:00" "20/11/2016 13
##  $ trt                : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc           : Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ d18O.corrected_WV  : num  -14.4 -13.3 -14.6 -13.8 -14.1 ...
##  $ Tdh.1              : chr  "20/11/2016 12:00" "20/11/2016 12:00" "20/11/2016 13:00" "20/11/2016 13
##  $ trt.1              : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc.1         : Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ d18O.corrected_WV.1: num  0.112 0.2326 0.0966 0.168 0.2119 ...
##  $ Time               : POSIXct, format: "2016-11-20 12:00:00" "2016-11-20 12:00:00" ...
```

```
start = as.POSIXct("2016-11-20 19:00:00\"", tz = "UTC")
end = as.POSIXct("2016-11-21 07:00:00", tz = "UTC")
aa <- ggplot(data = data, aes(x = Time, y = d18O.corrected_WV)) +
    geom_point(aes(colour = trt, shape = rangewtc), size = 3.5,
        ) +
annotate("rect", fill = "black", alpha = 0.15, xmin = end, xmax = start,
    ymin = -Inf, ymax = Inf) + scale_color_manual(name = "Legend:",
    labels = c("Transpired Ambient Chamber", "Transpired Elevated Chamber"),
    values = c("black", "red")) + scale_shape_manual(name = "Legend:",
    labels = c("in", "out"), values = c(8, 16)) + geom_errorbar(aes(ymin = d18O.corrected_WV -
    d18O.corrected_WV.1, ymax = d18O.corrected_WV + d18O.corrected_WV.1),
    width = 0.2, position = position_dodge(0.9)) + labs(y = "",
    x = "", element_text(size = 6)) + # scale_x_datetime(breaks = scales::pretty_breaks(n = 12))
theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(-15, -8)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 16, angle = 45, hjust = 1, vjust = 1))

iso3 <- aa + geom_point(data = data2, mapping = aes(x = Time,
    y = d18O.corrected_WV.AMB.), size = 3, color = "gray57")
iso3
```



#next we want to plot the chamber condensation

```
wtc <- dfa
Cw <- wtc
Cw <- subset(Cw, Camp == 1)

Cw <- Cw[, c("rangewtc", "Tdh", "chamber", "trt", "d18O.corrected_Cond")]
Trtmean <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(mean),
    keep.names = T, na.rm = TRUE, data = Cw)
se <- function(x) sqrt(var(x)/length(x))
Trtsd <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(se), keep.names = T,
    data = Cw)
data <- cbind(Trtmean, Trtsd)
data <- as.data.frame(data)
data["Time"] <- (dmy_hm(data$Tdh, quiet = TRUE, tz = "UTC"))
str(data)
```

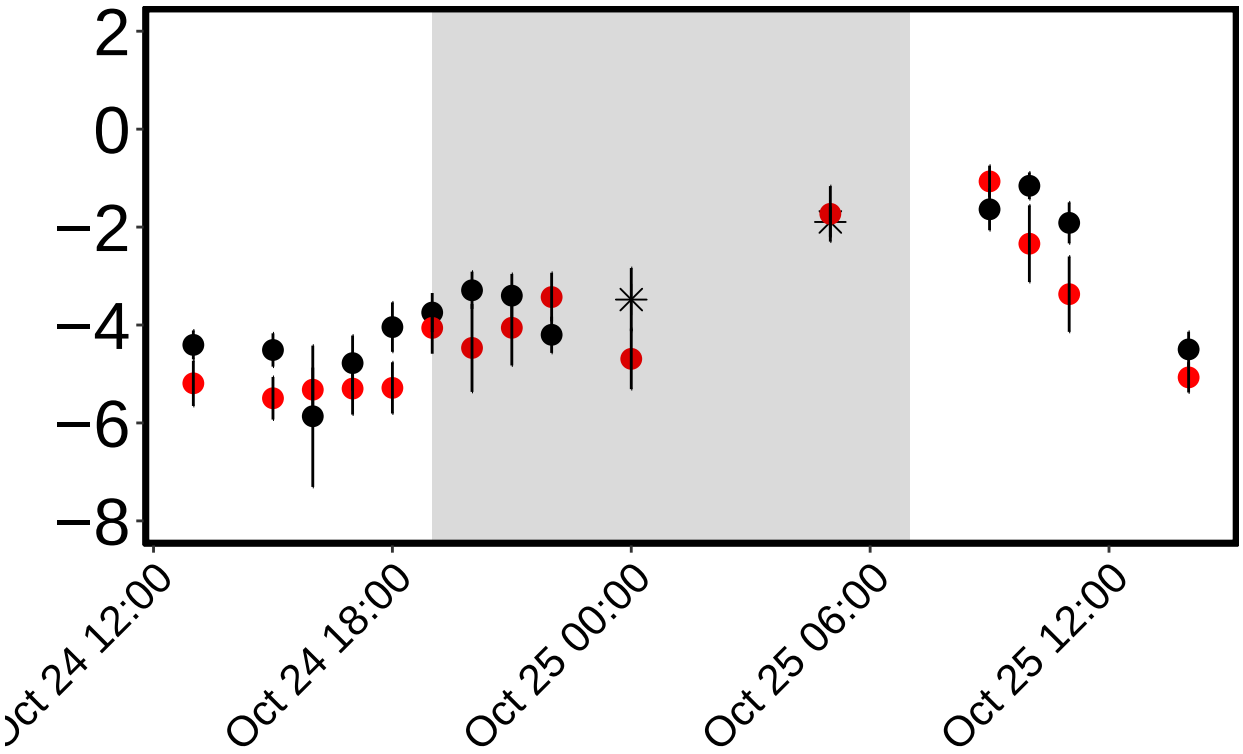
```
## 'data.frame':    54 obs. of  9 variables:
##  $ Tdh                  : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 
##  $ trt                  : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc             : Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ d18O.corrected_Cond  : num  -4.41 -5.19 NaN NaN -4.51 ...
##  $ Tdh.1                : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 
##  $ trt.1                : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc.1           : Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ d18O.corrected_Cond.1: num  0.295 0.462 NA NA 0.341 ...
##  $ Time                 : POSIXct, format: "2016-10-24 13:00:00" "2016-10-24 13:00:00" ...
```

```
######################
Cwx <- wtc
Cwx <- subset(Cwx, Camp == 3)
Cwx <- Cwx[, c("rangewtc", "Tdh", "chamber", "trt", "d18O.corrected_Cond")]
Trtmeanx <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(mean),
    keep.names = T, na.rm = TRUE, data = Cwx)
se <- function(x) sqrt(var(x)/length(x))
Trtsdx <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(se), keep.names = T,
    data = Cwx)
data2 <- cbind(Trtmeanx, Trtsdx)
data2 <- as.data.frame(data2)
data2["Time"] <- (dmy_hm(data2$Tdh, quiet = TRUE, tz = "UTC"))
str(data)
```

```
## 'data.frame':    54 obs. of  9 variables:
##  $ Tdh                  : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 
##  $ trt                  : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc             : Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ d18O.corrected_Cond  : num  -4.41 -5.19 NaN NaN -4.51 ...
##  $ Tdh.1                : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 
##  $ trt.1                : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc.1           : Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ d18O.corrected_Cond.1: num  0.295 0.462 NA NA 0.341 ...
##  $ Time                 : POSIXct, format: "2016-10-24 13:00:00" "2016-10-24 13:00:00" ...
```

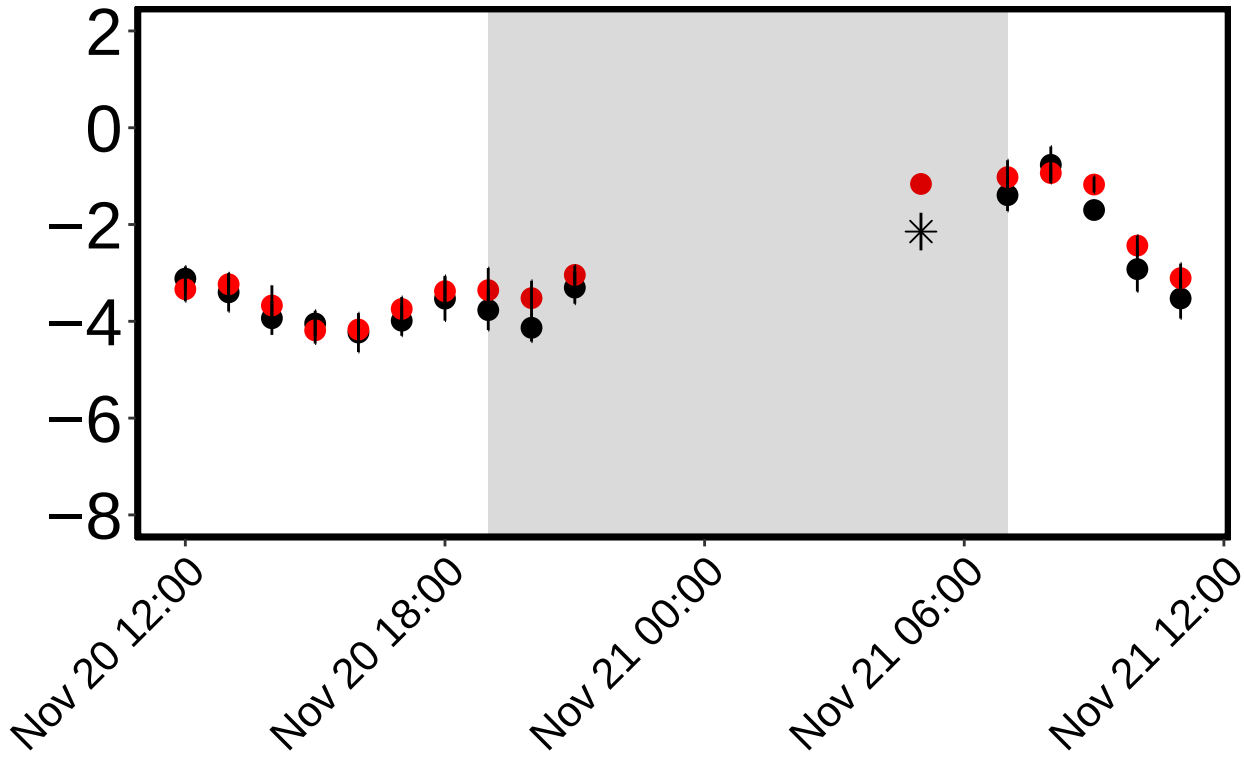
```
start = as.POSIXct("2016-10-24 19:00:00\"", tz = "UTC")
end = as.POSIXct("2016-10-25 07:00:00", tz = "UTC")
start2 = as.POSIXct("2016-11-20 19:00:00\"", tz = "UTC")
end2 = as.POSIXct("2016-11-21 07:00:00", tz = "UTC")
c1 <- ggplot(data = data, aes(x = Time, y = d18O.corrected_Cond)) +
    geom_point(aes(colour = trt, shape = rangewtc), size = 3.5,
        ) + annotate("rect", fill = "black", alpha = 0.15, xmin = end,
    xmax = start, ymin = -Inf, ymax = Inf) + scale_color_manual(name = "Legend:",
    labels = c("Transpired Ambient Chamber", "Transpired Elevated Chamber"),
    values = c("black", "red")) + scale_shape_manual(name = "Legend:",
    labels = c("in", "out"), values = c(8, 16)) + geom_errorbar(aes(ymin = d18O.corrected_Cond -
    d18O.corrected_Cond.1, ymax = d18O.corrected_Cond + d18O.corrected_Cond.1),
    width = 0.2, position = position_dodge(0.9)) + labs(y = "",
    x = "", element_text(size = 6)) + theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(-8, 2)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 16, angle = 45, hjust = 1, vjust = 1))
c1
```

```
## Warning: Removed 24 rows containing missing values (geom_point).
```




```
c3 <- ggplot(data = data2, aes(x = Time, y = d18O.corrected_Cond)) +
    geom_point(aes(colour = trt, shape = rangewtc), size = 3.5,
        ) + annotate("rect", fill = "black", alpha = 0.15, xmin = end2,
    xmax = start2, ymin = -Inf, ymax = Inf) + scale_color_manual(name = "Legend:",
    labels = c("Transpired Ambient Chamber", "Transpired Elevated Chamber"),
    values = c("black", "red")) + scale_shape_manual(name = "Legend:",
    labels = c("in", "out"), values = c(8, 16)) + geom_errorbar(aes(ymin = d18O.corrected_Cond -
    d18O.corrected_Cond.1, ymax = d18O.corrected_Cond + d18O.corrected_Cond.1),
    width = 0.2, position = position_dodge(0.9)) + labs(y = "",
    x = "", element_text(size = 6)) + theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(-8, 2)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 16, angle = 45, hjust = 1, vjust = 1))
c3
```

```
## Warning: Removed 16 rows containing missing values (geom_point).
```



#lastly we plot d18otrans - the only change here is we shade +- 2 per mil

```
wtc <- dfa
Cw <- wtc
Cw <- subset(Cw, Camp == 1)
Cw <- Cw[, c("rangewtc", "Tdh", "chamber", "trt", "dtrans")]
Trtmean <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(mean),
    keep.names = T, na.rm = TRUE, data = Cw)
se <- function(x) sqrt(var(x)/length(x))
Trtsd <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(se), keep.names = T,
    data = Cw)
data <- cbind(Trtmean, Trtsd)
data <- as.data.frame(data)
data["Time"] <- (dmy_hm(data$Tdh, quiet = TRUE, tz = "UTC"))
str(data)
```

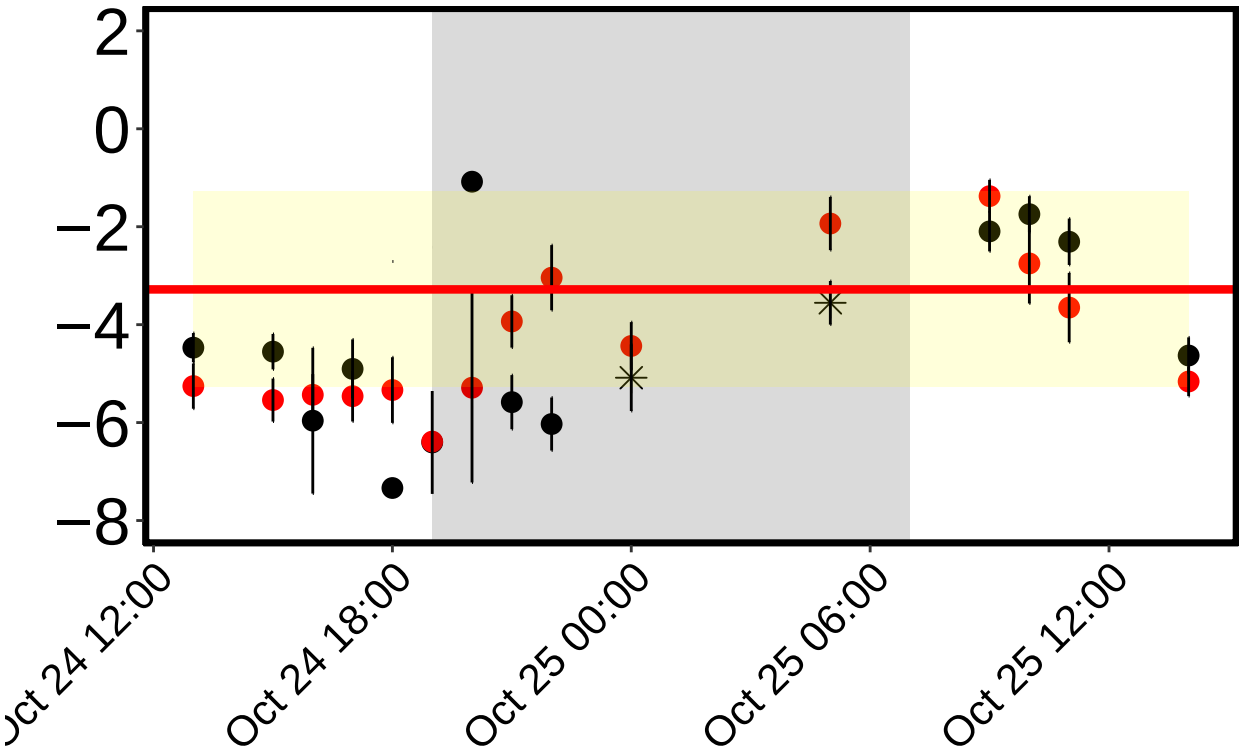
```
## 'data.frame':    54 obs. of  9 variables:
##  $ Tdh       : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 14:00" ...
##  $ trt       : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc  : Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ dtrans    : num  -4.47 -5.25 NaN NaN -4.55 ...
##  $ Tdh.1     : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 14:00" ...
##  $ trt.1     : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc.1: Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ dtrans.1  : num  0.299 0.462 NA NA 0.362 ...
##  $ Time      : POSIXct, format: "2016-10-24 13:00:00" "2016-10-24 13:00:00" ...
```

```
######################
Cwx <- wtc
Cwx <- subset(Cwx, Camp == 3)
Cwx <- Cwx[, c("rangewtc", "Tdh", "chamber", "trt", "dtrans")]
Trtmeanx <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(mean),
    keep.names = T, na.rm = TRUE, data = Cwx)
se <- function(x) sqrt(var(x)/length(x))
Trtsdx <- summaryBy(. ~ Tdh + trt + rangewtc, FUN = c(se), keep.names = T,
    data = Cwx)
data2 <- cbind(Trtmeanx, Trtsdx)
data2 <- as.data.frame(data2)
data2["Time"] <- (dmy_hm(data2$Tdh, quiet = TRUE, tz = "UTC"))
str(data)
```

```
## 'data.frame':    54 obs. of  9 variables:
##  $ Tdh       : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 14:00" ...
##  $ trt       : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc  : Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ dtrans    : num  -4.47 -5.25 NaN NaN -4.55 ...
##  $ Tdh.1     : chr  "24/10/2016 13:00" "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 14:00" ...
##  $ trt.1     : chr  "ambient" "elevated " "ambient" "elevated " ...
##  $ rangewtc.1: Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ dtrans.1  : num  0.299 0.462 NA NA 0.362 ...
##  $ Time      : POSIXct, format: "2016-10-24 13:00:00" "2016-10-24 13:00:00" ...
```

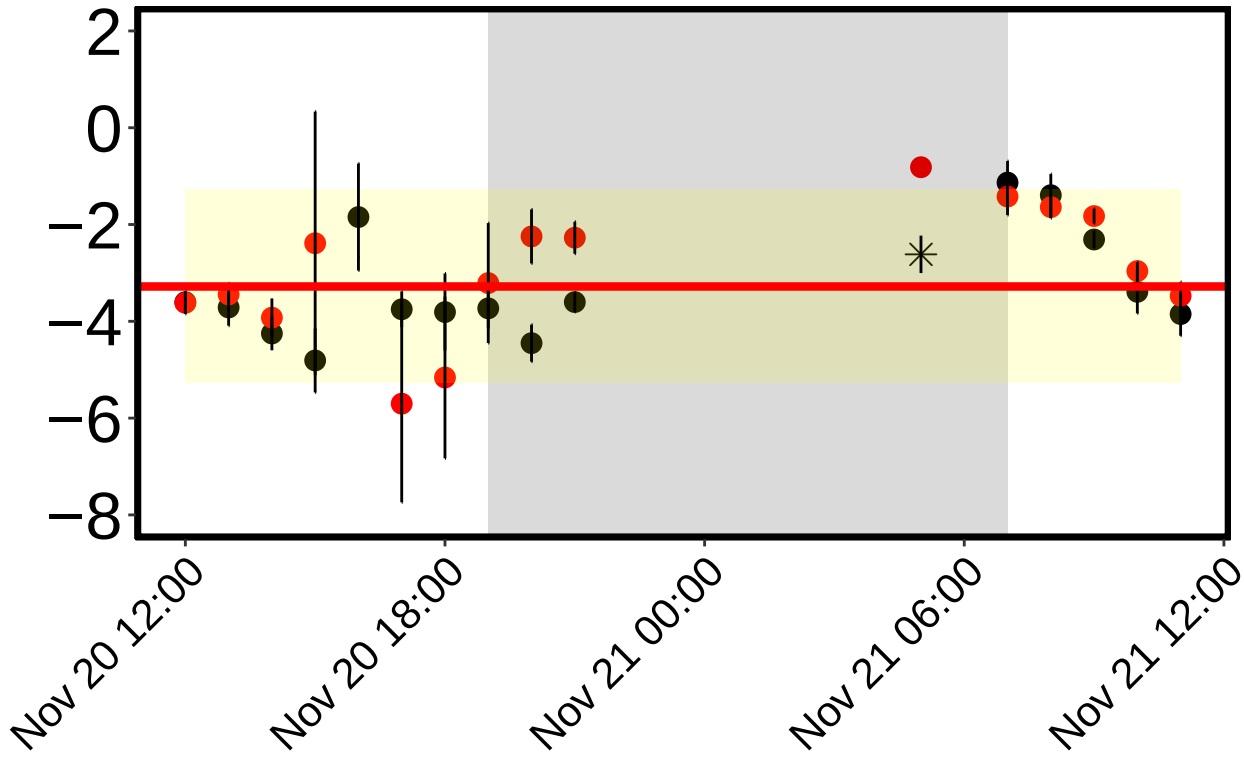
```
start = as.POSIXct("2016-10-24 19:00:00\"", tz = "UTC")
end = as.POSIXct("2016-10-25 07:00:00", tz = "UTC")
start2 = as.POSIXct("2016-11-20 19:00:00\"", tz = "UTC")
end2 = as.POSIXct("2016-11-21 07:00:00", tz = "UTC")
# shade within 2%
infa1 <- as.POSIXct("2016-10-24 13:00:00\"", tz = "UTC")
infb1 <- as.POSIXct("2016-10-25 14:00:00\"", tz = "UTC")
infa <- as.POSIXct("2016-11-20 12:00:00\"", tz = "UTC")
infb <- as.POSIXct("2016-11-21 11:00:00\"", tz = "UTC")

dt1 <- ggplot(data = data, aes(x = Time, y = dtrans)) + geom_point(aes(colour = trt,
    shape = rangewtc), size = 3.5, ) + annotate("rect", fill = "black",
    alpha = 0.15, xmin = end, xmax = start, ymin = -Inf, ymax = Inf) +
    annotate("rect", fill = "yellow", alpha = 0.15, xmin = infa1,
        xmax = infb1, ymin = -5.28, ymax = -1.28) + scale_color_manual(name = "Legend:",
    labels = c("Transpired Ambient Chamber", "Transpired Elevated Chamber"),
    values = c("black", "red")) + scale_shape_manual(name = "Legend:",
    labels = c("in", "out"), values = c(8, 16)) + geom_errorbar(aes(ymin = dtrans -
    dtrans.1, ymax = dtrans + dtrans.1), width = 0.2, position = position_dodge(0.9)) +
    labs(y = "", x = "", element_text(size = 6)) + theme_classic() +
    theme(panel.border = element_rect(fill = "NA", colour = "black",
        size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(-8, 2)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 16, angle = 45, hjust = 1, vjust = 1)) + geom_hline(yintercept = -3.28,
    colour = "red", size = 1.5)
dt1
```

```
## Warning: Removed 24 rows containing missing values (geom_point).
```




```
dt3 <- ggplot(data = data2, aes(x = Time, y = dtrans)) + geom_point(aes(colour = trt,
    shape = rangewtc), size = 3.5, ) + annotate("rect", fill = "black",
    alpha = 0.15, xmin = end2, xmax = start2, ymin = -Inf, ymax = Inf) +
    annotate("rect", fill = "yellow", alpha = 0.15, xmin = infa,
        xmax = infb, ymin = -5.28, ymax = -1.28) + scale_color_manual(name = "Legend:",
    labels = c("Transpired Ambient Chamber", "Transpired Elevated Chamber"),
    values = c("black", "red")) + scale_shape_manual(name = "Legend:",
    labels = c("in", "out"), values = c(8, 16)) + geom_errorbar(aes(ymin = dtrans -
    dtrans.1, ymax = dtrans + dtrans.1), width = 0.2, position = position_dodge(0.9)) +
    labs(y = "", x = "", element_text(size = 6)) + theme_classic() +
    theme(panel.border = element_rect(fill = "NA", colour = "black",
        size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(-8, 2)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 16, angle = 45, hjust = 1, vjust = 1)) + geom_hline(yintercept = -3.28,
    colour = "red", size = 1.5)
dt3
```

```
## Warning: Removed 17 rows containing missing values (geom_point).
```



#we finish by mergeing these panels into the figure

```
figure3 = plot_grid(th1, th3, thi1, thi3, iso1, iso3, c1, c3,
    dt1, dt3, ncol = 2, align = "hv")
```

```
## Warning: Removed 24 rows containing missing values (geom_point).
```

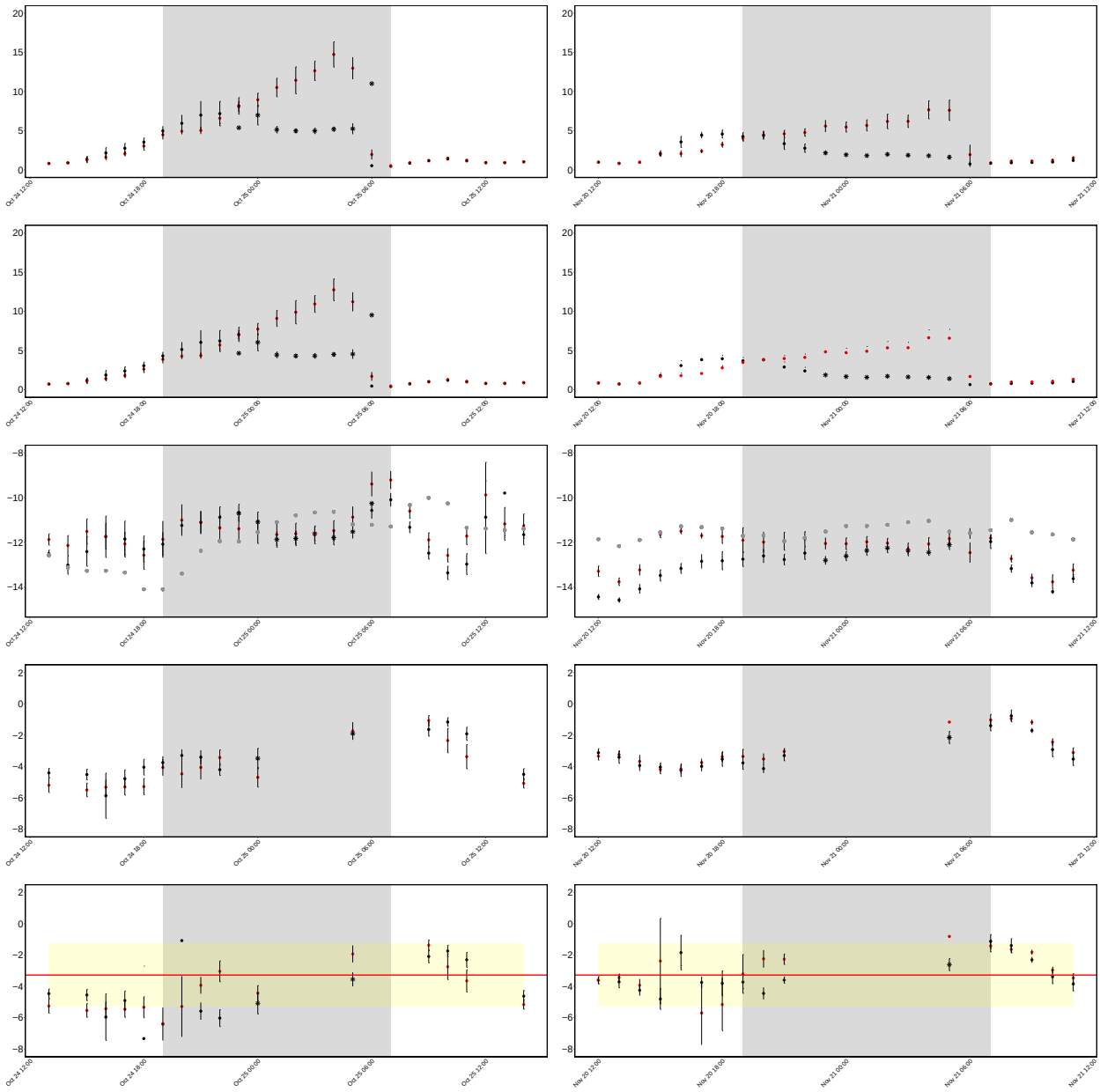
```
## Warning: Removed 16 rows containing missing values (geom_point).
```

```
## Warning: Removed 24 rows containing missing values (geom_point).
```

```
## Warning: Removed 17 rows containing missing values (geom_point).
```

```
# again ggsave the figure out and annoated in illustator
```

```
plot(figure3)
```



###The same data makes figure S4

```
wtc <- dfa
Cw <- wtc
Cw <- subset(Cw, Camp == 1)

Cw <- Cw[, c("rangewtc", "Tdh", "chamber", "trt", "dtrans")]
data = Cw
data["Time"] <- (dmy_hm(data$Tdh, quiet = TRUE, tz = "UTC"))
str(data)
```

```
## 'data.frame':    312 obs. of  6 variables:
##  $ rangewtc: Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Tdh     : chr  "24/10/2016 13:00" "24/10/2016 14:00" "24/10/2016 15:00" "24/10/2016 16:00" ...
```

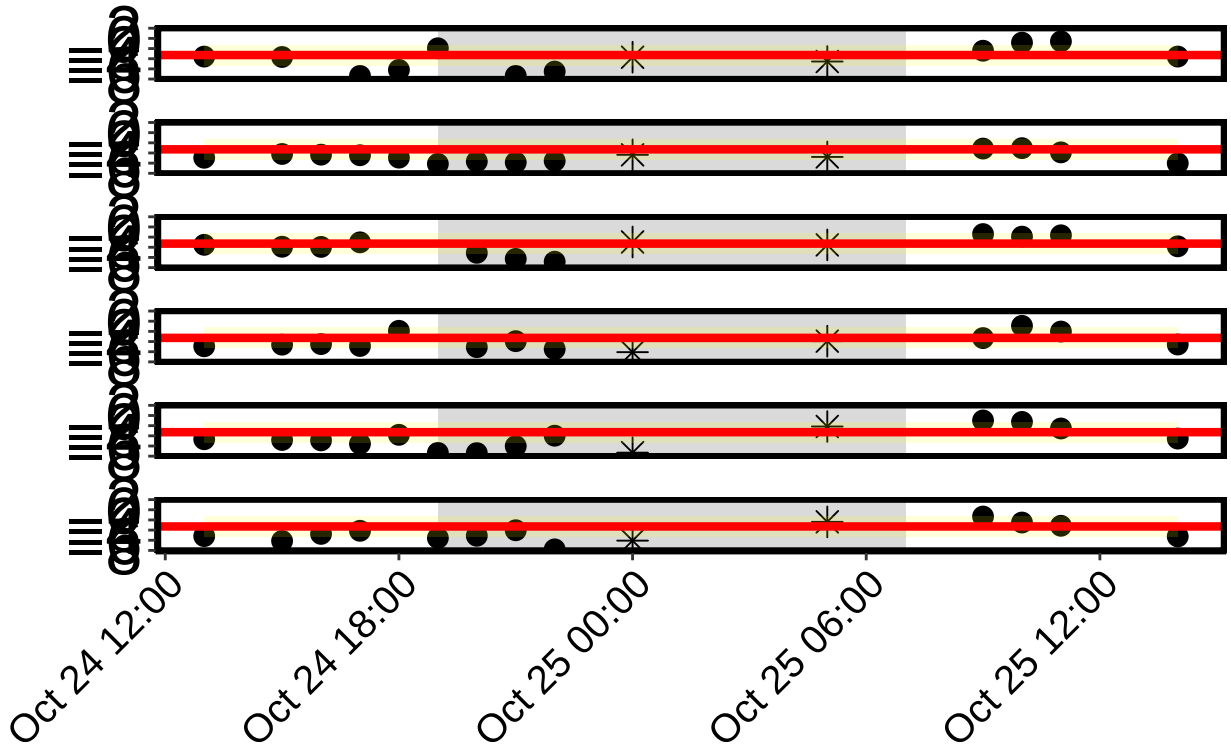
```
##  $ chamber : chr  "C01" "C01" "C01" "C01" ...
##  $ trt     : chr  "ambient" "ambient" "ambient" "ambient" ...
##  $ dtrans  : num  -3.58 NA -3.66 -13.37 -7.38 ...
##  $ Time    : POSIXct, format: "2016-10-24 13:00:00" "2016-10-24 14:00:00" ...
```

```
amb <- data[data$trt == "ambient", ]
hot <- data[data$trt == "elevated ", ]
Cwx <- wtc
Cwx <- subset(Cwx, Camp == 3)
Cwx <- Cwx[c("rangewtc", "Tdh", "chamber", "trt", "dtrans")]
datax = Cwx
datax["Time"] <- (dmy_hm(datax$Tdh, quiet = TRUE, tz = "UTC"))
str(datax)
```

```
## 'data.frame':    288 obs. of  6 variables:
##  $ rangewtc: Factor w/ 2 levels "not","within": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Tdh     : chr  "20/11/2016 12:00" "20/11/2016 13:00" "20/11/2016 14:00" "20/11/2016 15:00" ...
##  $ chamber : chr  "C01" "C01" "C01" "C01" ...
##  $ trt     : chr  "ambient" "ambient" "ambient" "ambient" ...
##  $ dtrans  : num  -3.59 -3.51 -3.97 -4.35 -4.45 ...
##  $ Time    : POSIXct, format: "2016-11-20 12:00:00" "2016-11-20 13:00:00" ...
```

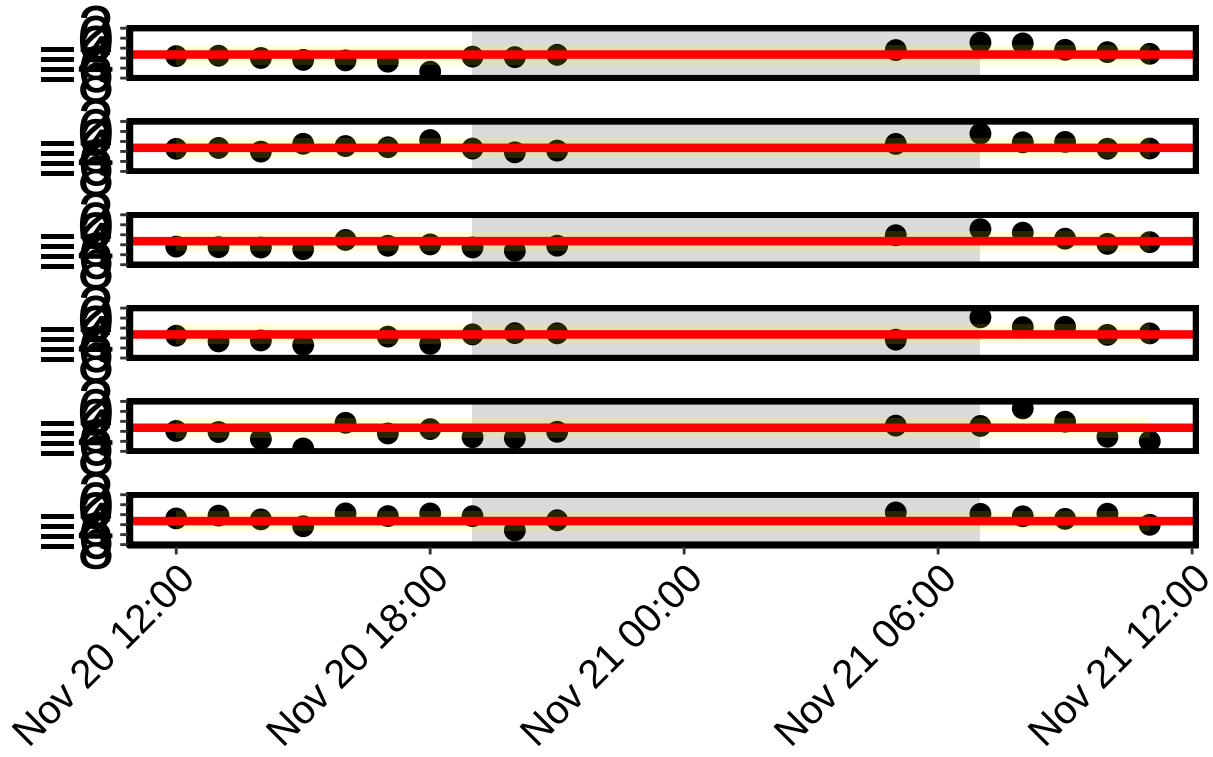
```
ambx <- datax[datax$trt == "ambient", ]
hotx <- datax[datax$trt == "elevated ", ]
start = as.POSIXct("2016-10-24 19:00:00\"", tz = "UTC")
end = as.POSIXct("2016-10-25 07:00:00", tz = "UTC")
start2 = as.POSIXct("2016-11-20 19:00:00\"", tz = "UTC")
end2 = as.POSIXct("2016-11-21 07:00:00", tz = "UTC")
# shade within 2%
infa1 <- as.POSIXct("2016-10-24 13:00:00\"", tz = "UTC")
infb1 <- as.POSIXct("2016-10-25 14:00:00\"", tz = "UTC")
infa <- as.POSIXct("2016-11-20 12:00:00\"", tz = "UTC")
infb <- as.POSIXct("2016-11-21 11:00:00\"", tz = "UTC")
a <- ggplot(data = amb, aes(x = Time, y = dtrans)) + geom_point(aes(colour = trt,
    shape = rangewtc), size = 3.5, ) + annotate("rect", fill = "black",
    alpha = 0.15, xmin = end, xmax = start, ymin = -Inf, ymax = Inf) +
    annotate("rect", fill = "yellow", alpha = 0.15, xmin = infa1,
        xmax = infb1, ymin = -5.28, ymax = -1.28) + scale_color_manual(name = "Legend:",
    labels = c("Transpired Ambient Chamber", "Transpired Elevated Chamber"),
    values = c("black", "red")) + scale_shape_manual(name = "Legend:",
    labels = c("in", "out"), values = c(8, 16)) + labs(y = "",
    x = "", element_text(size = 6)) + theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(-8, 2)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 15, angle = 45, hjust = 1, vjust = 1), plot.margin = margin(10,
    10, 10, 10)) + geom_hline(yintercept = -3.28, colour = "red",
    size = 1.5) + facet_wrap(~chamber, ncol = 1) + theme(strip.background = element_blank(),
    strip.text.x = element_blank()) + theme(panel.spacing = unit(1,
    "lines"))
a
```

```
## Warning: Removed 72 rows containing missing values (geom_point).
```



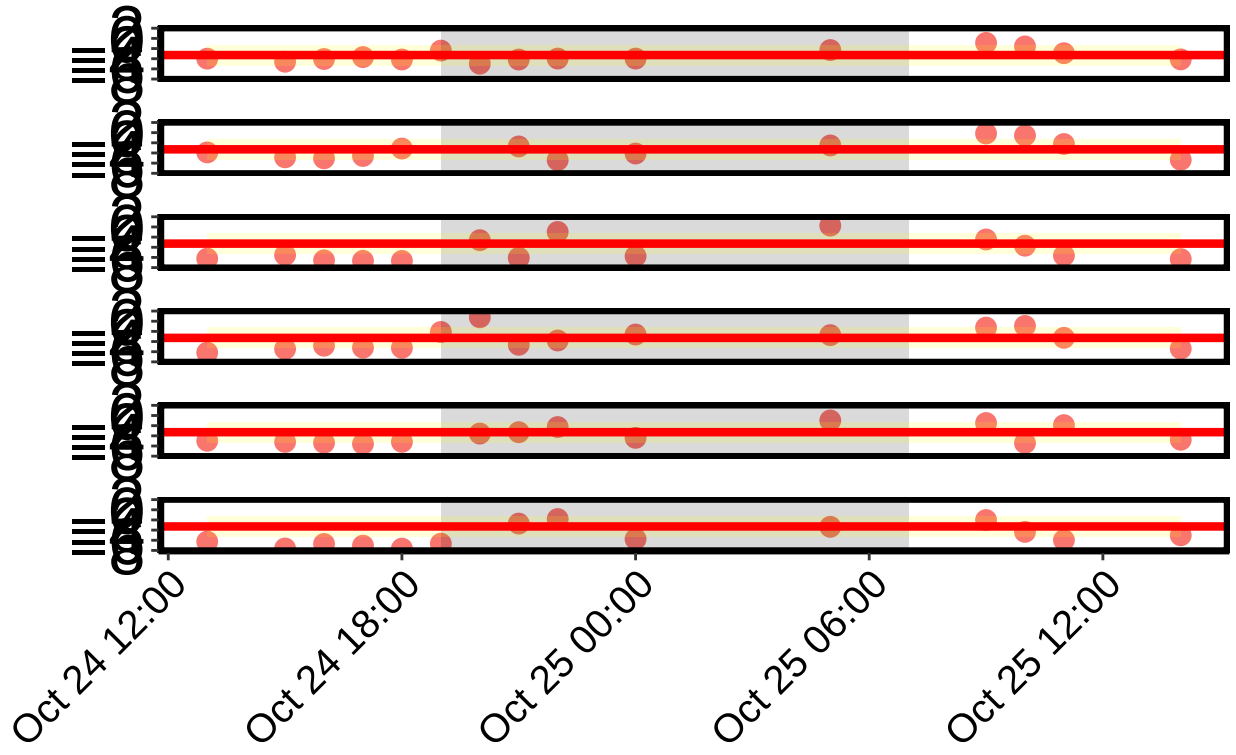

```
b <- ggplot(data = ambx, aes(x = Time, y = dtrans)) + geom_point(aes(colour = trt,
    shape = rangewtc), size = 3.5, shape = 16) + annotate("rect",
    fill = "black", alpha = 0.15, xmin = end2, xmax = start2,
    ymin = -Inf, ymax = Inf) + annotate("rect", fill = "yellow",
    alpha = 0.15, xmin = infa, xmax = infb, ymin = -5.28, ymax = -1.28) +
    scale_color_manual(name = "Legend:", labels = c("Transpired Ambient Chamber",
        "Transpired Elevated Chamber"), values = c("black", "red")) +
    scale_shape_manual(name = "Legend:", labels = c("in", "out"),
        values = c(8, 16)) + labs(y = "", x = "", element_text(size = 6)) +
    theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(-8, 2)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 15, angle = 45, hjust = 1, vjust = 1), plot.margin = margin(10,
    10, 10, 10)) + geom_hline(yintercept = -3.28, colour = "red",
    size = 1.5) + facet_wrap(~chamber, ncol = 1) + theme(strip.background = element_blank(),
    strip.text.x = element_blank()) + theme(panel.spacing = unit(1,
    "lines"))
b
```

```
## Warning: Removed 49 rows containing missing values (geom_point).
```



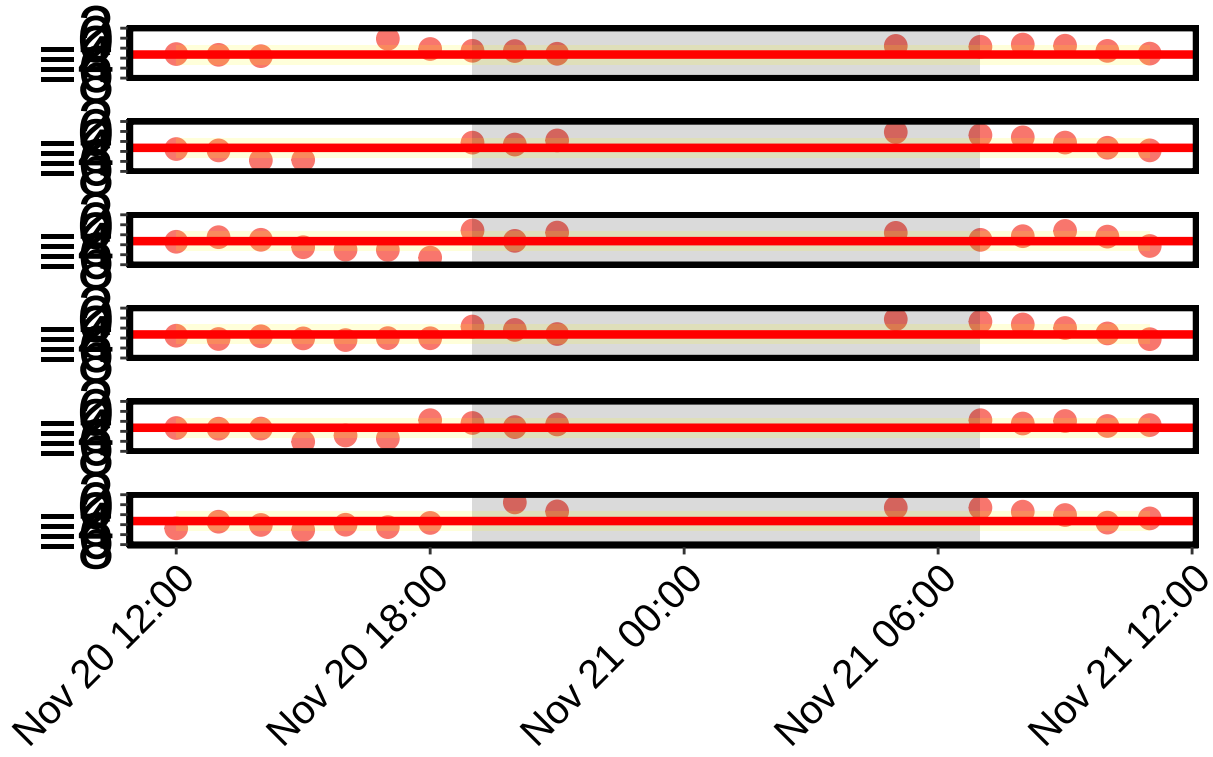

```
c <- ggplot(data = hot, aes(x = Time, y = dtrans)) + geom_point(aes(colour = "red",
    shape = rangewtc), size = 3.5, shape = 16) + annotate("rect",
    fill = "black", alpha = 0.15, xmin = end, xmax = start, ymin = -Inf,
    ymax = Inf) + annotate("rect", fill = "yellow", alpha = 0.15,
    xmin = infa1, xmax = infb1, ymin = -5.28, ymax = -1.28) +
    labs(y = "", x = "", element_text(size = 6)) + # scale_x_datetime(breaks = scales::pretty_breaks(n
theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(-8, 2)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 15, angle = 45, hjust = 1, vjust = 1), plot.margin = margin(10,
    10, 10, 10)) + geom_hline(yintercept = -3.28, colour = "red",
    size = 1.5) + facet_wrap(~chamber, ncol = 1) + theme(strip.background = element_blank(),
    strip.text.x = element_blank()) + theme(panel.spacing = unit(1,
    "lines"))
c
```

```
## Warning: Removed 71 rows containing missing values (geom_point).
```




```
d <- ggplot(data = hotx, aes(x = Time, y = dtrans)) + geom_point(aes(colour = "red"),
    size = 3.5, ) + annotate("rect", fill = "black", alpha = 0.15,
    xmin = end2, xmax = start2, ymin = -Inf, ymax = Inf) + annotate("rect",
    fill = "yellow", alpha = 0.15, xmin = infa, xmax = infb,
    ymin = -5.28, ymax = -1.28) + scale_shape_manual(name = "Legend:",
    labels = c("in", "out"), values = c(8, 16)) + labs(y = "",
    x = "", element_text(size = 6)) + theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(-8, 2)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 15, angle = 45, hjust = 1, vjust = 1), plot.margin = margin(10,
    10, 10, 10)) + geom_hline(yintercept = -3.28, colour = "red",
    size = 1.5) + facet_wrap(~chamber, ncol = 1) + theme(strip.background = element_blank(),
    strip.text.x = element_blank()) + theme(panel.spacing = unit(1,
    "lines"))
d
```

```
## Warning: Removed 55 rows containing missing values (geom_point).
```




```
figures4 = cowplot::plot_grid(a, b, c, d, ncol = 4, align = "h")
```

```
## Warning: Removed 72 rows containing missing values (geom_point).
```

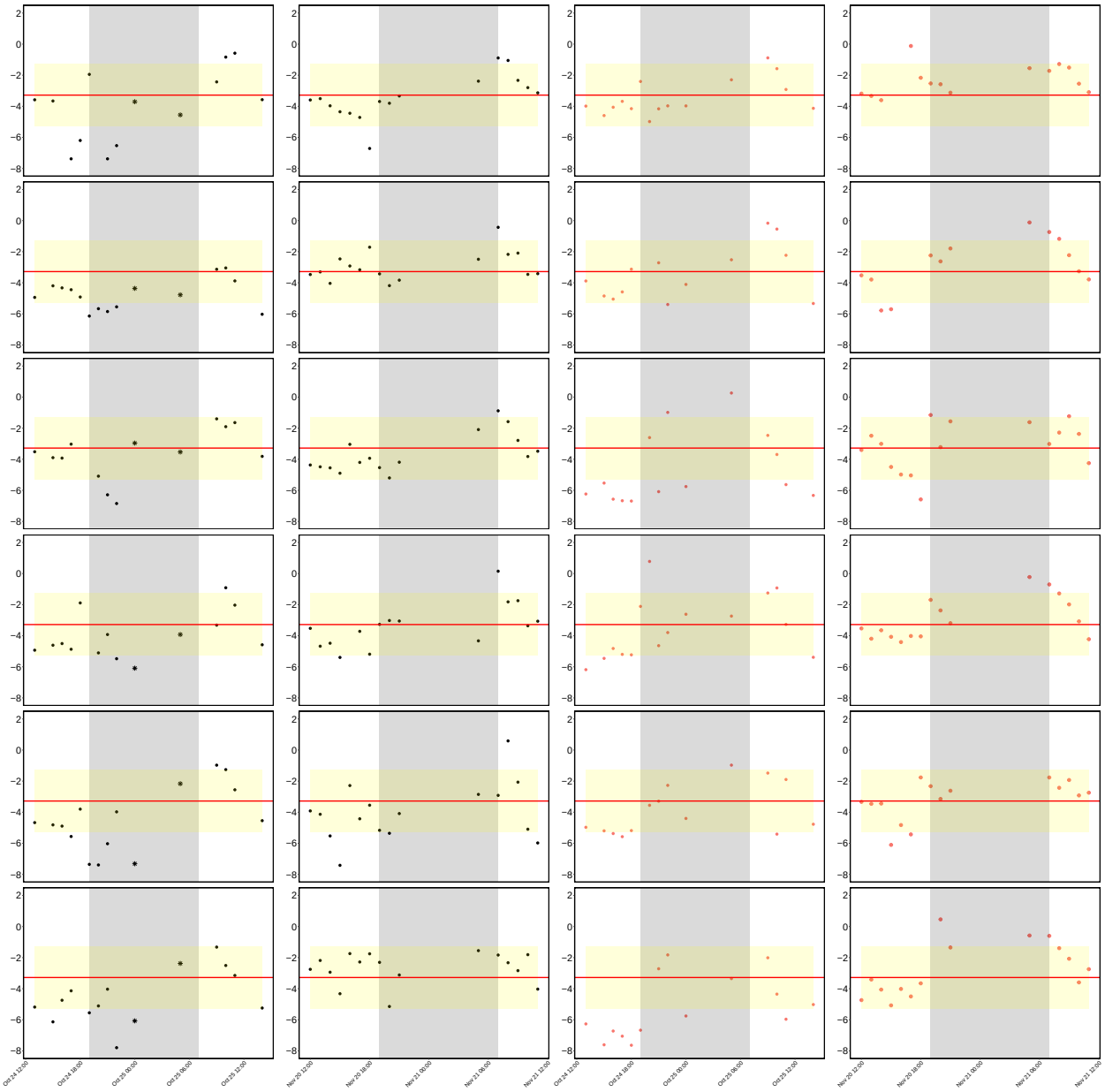
```
## Warning: Removed 49 rows containing missing values (geom_point).
```

```
## Warning: Removed 71 rows containing missing values (geom_point).
```

```
## Warning: Removed 55 rows containing missing values (geom_point).
```

```
# ggsave('Figures4.tiff', width = 40, height = 40, units =
# 'cm', dpi=300)
```

```
plot(figures4)
```

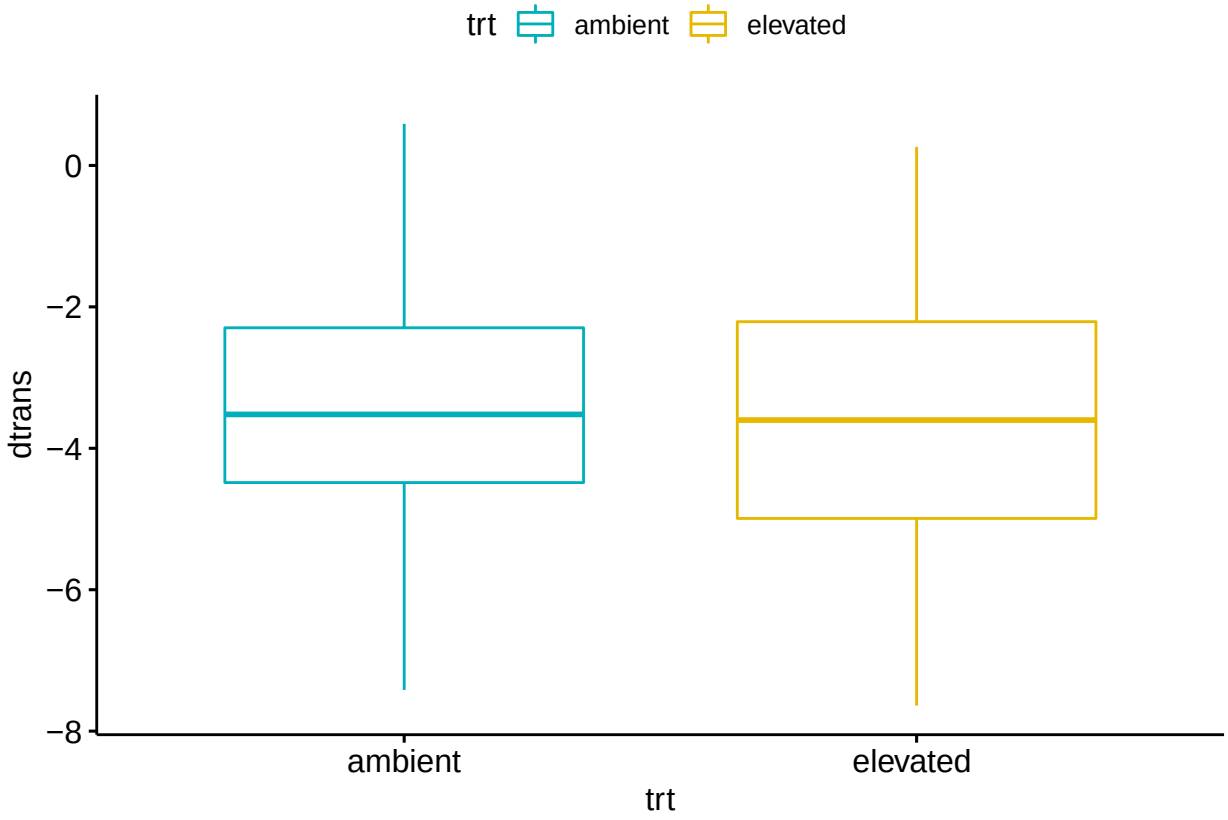


#Figure 5 is again made from dfa, so delete everything but dfa

```
keep(dfa, sure = TRUE)
```

#Here is figure 5

```
dftrans <- dfa
amb <- dfa
dftrans <- subset(dftrans, dtrans > -8 & dtrans < 2)
dftrans <- filter(dftrans, dftrans$sun == "day")
dftrans <- filter(dftrans, dftrans$rangewtc == "within")
dftrans$Camp <- as.factor(dftrans$Camp)
ggboxplot(dftrans, x = "trt", y = "dtrans", color = "trt", palette = c("#00AFBB",
    "#E7B800"), ylab = "dtrans", xlab = "trt")
```




```
res <- wilcox.test(dtrans ~ trt, data = dftrans, exact = FALSE)
res
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  dtrans by trt
## W = 8419, p-value = 0.531
## alternative hypothesis: true location shift is not equal to 0
```

```
c1 <- filter(dftrans, dftrans$Camp == "1")
c2 <- filter(dftrans, dftrans$Camp == "3")
aovc1 <- aov(c1$dtrans ~ c1$trt)
aovc2 <- aov(c2$dtrans ~ c2$trt)
summary(aovc1)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## c1$trt        1    7.3   7.309   2.242  0.137
## Residuals   109  355.3   3.259
```

```
summary(aovc2)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## c2$trt        1    1.5   1.522   0.676  0.412
## Residuals   141  317.3   2.250
```

```
## Relative HUmidty
colnames(dftrans)
```

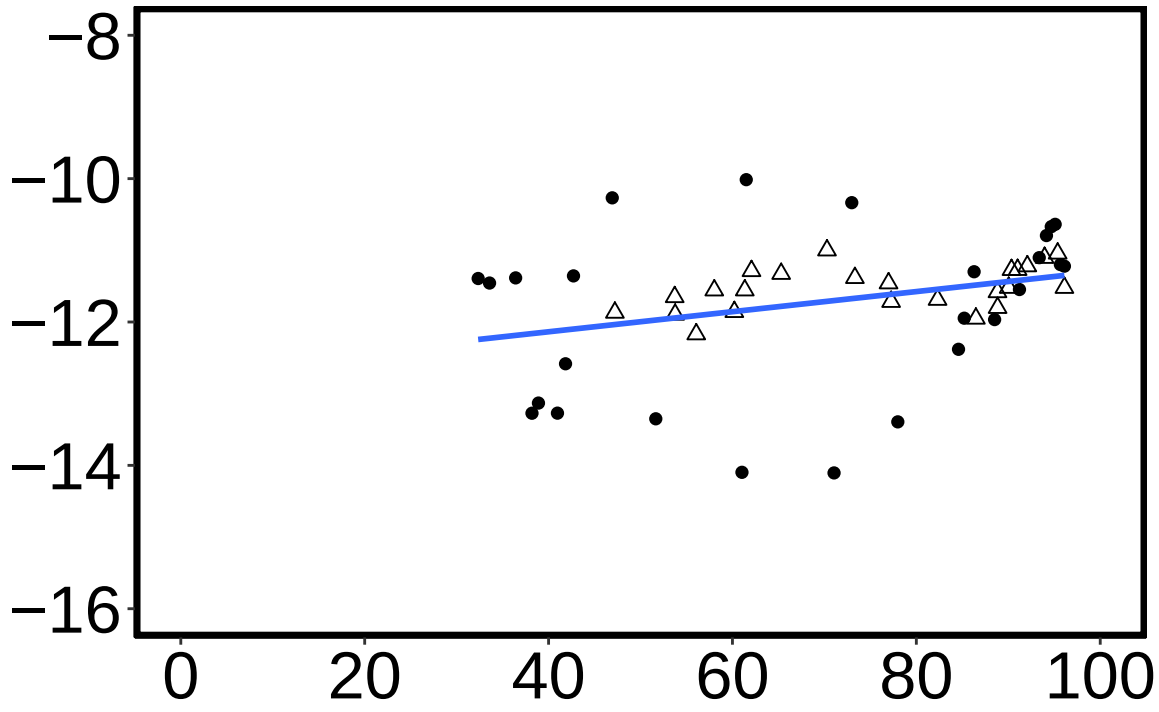
```
##   [1] "Tdh"                    "chamber"                "trt"
##   [4] "CO2L_mean"              "DIFFP_mean"             "CO2FLOW_mean"
##   [7] "CO2CChamb_mean"         "DPCChamb_mean"          "TAIR_mean"
##  [10] "AddTime_mean"           "LCO2slope_mean"         "CONDH2O_mean"
##  [13] "TBox_mean"              "DoorCnt_mean"           "CO2C_mean"
##  [16] "ROTR_RH_mean"           "AIRPRESS_mean"          "DewPntC_mean"
##  [19] "ROTR_TA_mean"           "PAR_mean"               "WIND_mean"
##  [22] "REFCO2_mean"            "RAIN_mean"              "Taref_al_mean"
##  [25] "RHref_al_mean"          "Tchilled_SP_mean"       "Tchilled_mean"
##  [28] "DPref_al_mean"          "Tair_SP_mean"           "RH_al_mean"
##  [31] "DP_al_mean"             "Tsub_al_mean"           "RH_SP_mean"
##  [34] "Tair_al_mean"           "kfactor_mean"           "ChVolume_mean"
##  [37] "HWTC_mean"              "prevDateTime_mean"      "prevHWTC_mean"
##  [40] "prevCWTC_mean"          "tcycle_mean"            "fCO2_mean"
##  [43] "Href_mean"              "ICO2_mean"              "Dref_mean"
##  [46] "Ain_mean"               "Aout_mean"              "v_mean"
##  [49] "F_mean"                 "deltaS_mean"            "Fwat_mean"
##  [52] "Vwat_mean"              "deltaH2O_mean"          "FluxH2O_mean"
##  [55] "FluxCO2_mean"           "d18O.corrected_WV"      "d18O.corrected_WV.AMB."
##  [58] "d18O.corrected_Cond"    "leaftemp"               "Camp"
##  [61] "source"                 "Pref"                   "fin"
##  [64] "finmol"                 "fout"                   "foutmol"
##  [67] "uoutwout"               "uinwin"                 "condout"
##  [70] "E"                      "e-wo"                   "d18out"
##  [73] "d18in"                  "d18c"                   "d18oute"
##  [76] "d18ine"                 "d18Ce"                  "dtrans"
##  [79] "leafarea"               "LACM"                   "Ei"
##  [82] "wi"                     "Emmol"                  "Eleaf"
##  [85] "esat"                   "vpd"                    "gs"
##  [88] "W"                      "t"                      "tm"
##  [91] "th"                     "ek"                     "alphak"
##  [94] "Eplus"                  "betaplus"               "p"
##  [97] "tiso"                   "tmiso"                  "thiso"
## [100] "range"                  "dpdiffwtc"              "rangewtc"
## [103] "sun"
```

```
amb <- filter(amb, amb$chamber == "CO1")
amb$Camp <- as.factor(amb$Camp)
astat <- lm(amb$d18O.corrected_WV.AMB. ~ amb$RHref_al_mean)
summary(astat)  # significant take these values (intercept slope r^2)
```

```
##
## Call:
## lm(formula = amb$d18O.corrected_WV.AMB. ~ amb$RHref_al_mean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4044 -0.3274  0.1573  0.4226  1.8207
##
## Coefficients:
```

```
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -12.697406   0.439462 -28.893   <2e-16 ***
## amb$RHref_al_mean   0.014021   0.005926   2.366   0.0221 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8584 on 48 degrees of freedom
## Multiple R-squared:  0.1045, Adjusted R-squared:  0.0858
## F-statistic: 5.599 on 1 and 48 DF,  p-value: 0.02206
```

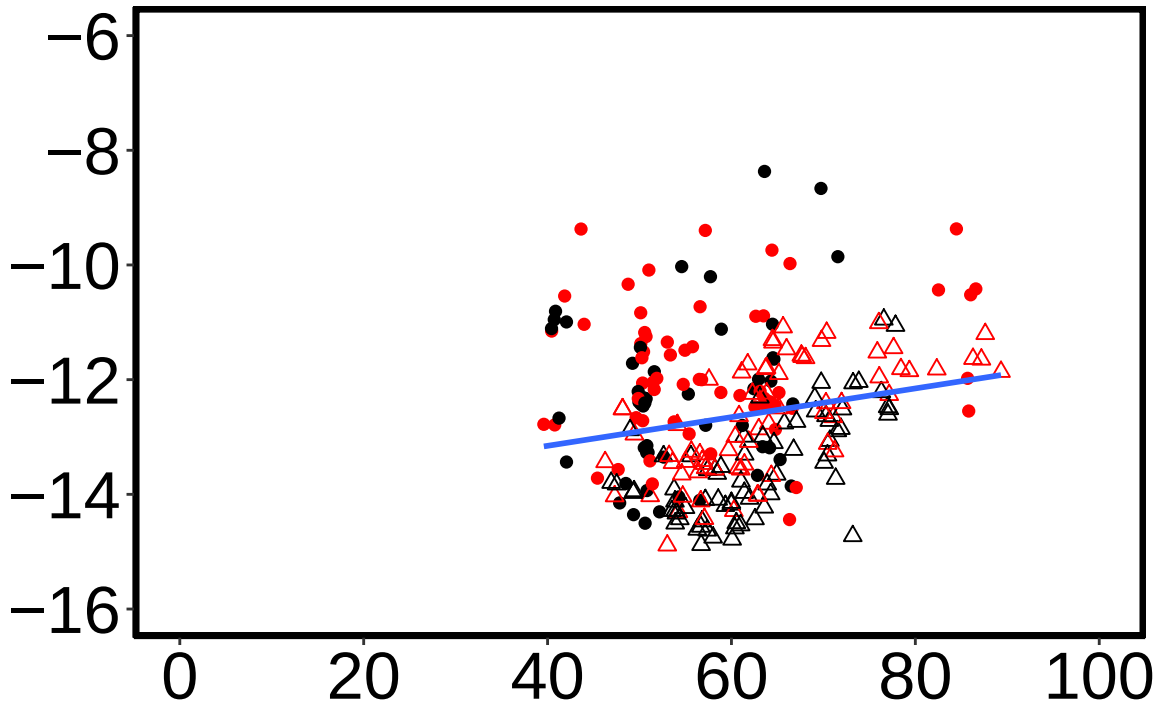
```
a <- ggplot(data = amb, aes(x = RHref_al_mean, y = d18O.corrected_WV.AMB.)) +
    geom_point(aes(colour = trt, shape = Camp), size = 2, ) +
    scale_color_manual(name = "Legend:", labels = c("Transpired Ambient Chamber",
        "Transpired Elevated Chamber"), values = c("black", "red")) +
    scale_shape_manual(name = "Legend:", values = c(16, 2)) +
    labs(y = "", x = "", element_text(size = 6)) + geom_smooth(method = "lm",
    formula = y ~ x, se = FALSE) + # scale_x_datetime(breaks = scales::pretty_breaks(n = 12))
theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(-16, -8)) + scale_x_continuous(breaks = scales::pretty_breaks(n = 5),
    limits = c(0, 100)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 25)) + theme(plot.margin = unit(c(5, 5, 5, 5), "mm"))
a
```



```r
# stats for B
bstat <- lm(dftrans$d18O.corrected_WV ~ dftrans$RH_al_mean)
summary(bstat)  #SIGNIFANT
```

```
##
## Call:
## lm(formula = dftrans$d18O.corrected_WV ~ dftrans$RH_al_mean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4031 -0.8748 -0.1004  0.7426  4.1946
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -14.154720   0.469634 -30.140  < 2e-16 ***
## dftrans$RH_al_mean   0.025020   0.007687   3.255  0.00129 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.247 on 252 degrees of freedom
## Multiple R-squared:  0.04035,    Adjusted R-squared:  0.03654
## F-statistic: 10.59 on 1 and 252 DF,  p-value: 0.001289
```

```r
b <- ggplot(data = dftrans, aes(x = RH_al_mean, y = d18O.corrected_WV)) +
    geom_point(aes(colour = trt, shape = Camp), size = 2, ) +
    scale_color_manual(name = "Legend:", labels = c("Transpired Ambient Chamber",
        "Transpired Elevated Chamber"), values = c("black", "red")) +
    scale_shape_manual(name = "Legend:", labels = c("in", "out"),
        values = c(16, 2)) + labs(y = "", x = "", element_text(size = 6)) +
    geom_smooth(method = "lm", formula = y ~ x, se = FALSE) +
    # scale_x_datetime(breaks = scales::pretty_breaks(n = 12))
theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(-16, -6)) + scale_x_continuous(breaks = scales::pretty_breaks(n = 5),
    limits = c(0, 100)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 25)) + theme(plot.margin = unit(c(5, 5, 5, 5), "mm"))
b
```

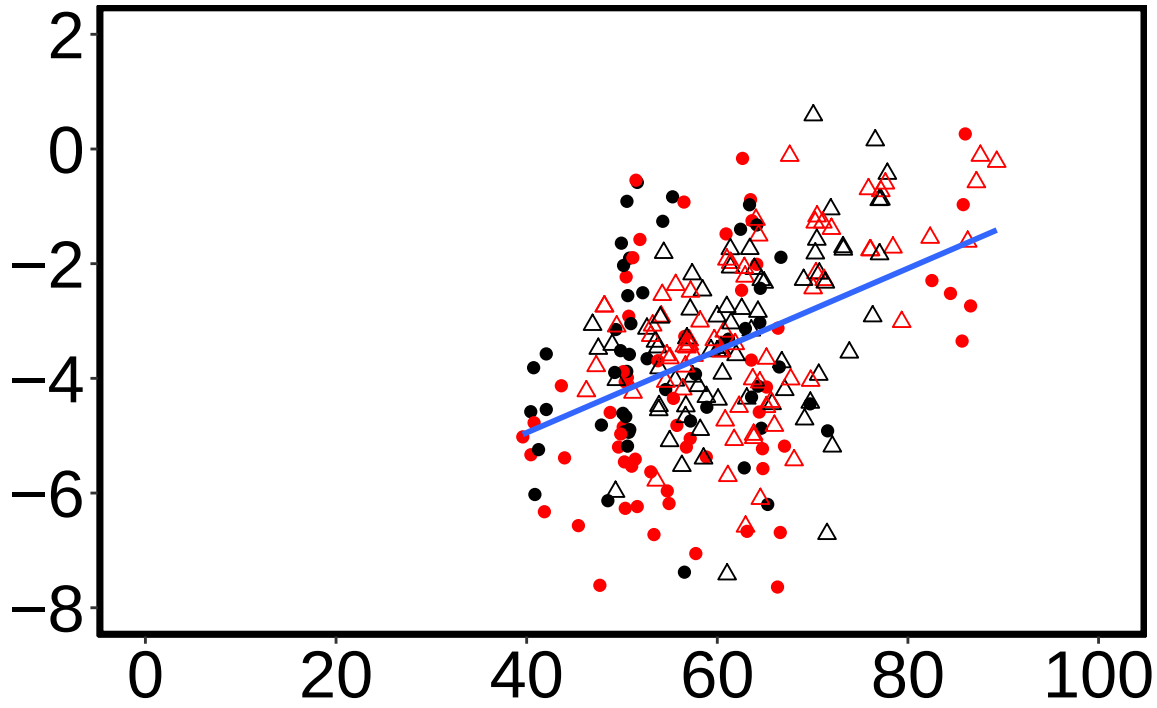



```
# stats for C
cstat <- lm(dftrans$dtrans ~ dftrans$RH_al_mean)
summary(cstat)   #siginfiant
```

```
##
## Call:
## lm(formula = dftrans$dtrans ~ dftrans$RH_al_mean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5774 -1.0282  0.1697  1.1090  3.5814
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -7.81360    0.57490 -13.591  < 2e-16 ***
## dftrans$RH_al_mean  0.07165    0.00941   7.615 5.31e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.527 on 252 degrees of freedom
## Multiple R-squared:  0.1871, Adjusted R-squared:  0.1838
## F-statistic: 57.98 on 1 and 252 DF,  p-value: 5.312e-13
```

```
c <- ggplot(data = dftrans, aes(x = RH_al_mean, y = dtrans)) +
    geom_point(aes(colour = trt, shape = Camp), size = 2, ) +
```

```
    scale_color_manual(name = "Legend:", labels = c("Transpired Ambient Chamber",
        "Transpired Elevated Chamber"), values = c("black", "red")) +
    scale_shape_manual(name = "Legend:", labels = c("in", "out"),
        values = c(16, 2)) + labs(y = "", x = "", element_text(size = 6)) +
    geom_smooth(method = "lm", formula = y ~ x, se = FALSE) +
    # scale_x_datetime(breaks = scales::pretty_breaks(n = 12))
theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(-8, 2)) + scale_x_continuous(breaks = scales::pretty_breaks(n = 5),
    limits = c(0, 100)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 25)) + theme(plot.margin = unit(c(5, 5, 5, 5), "mm"))
c
```

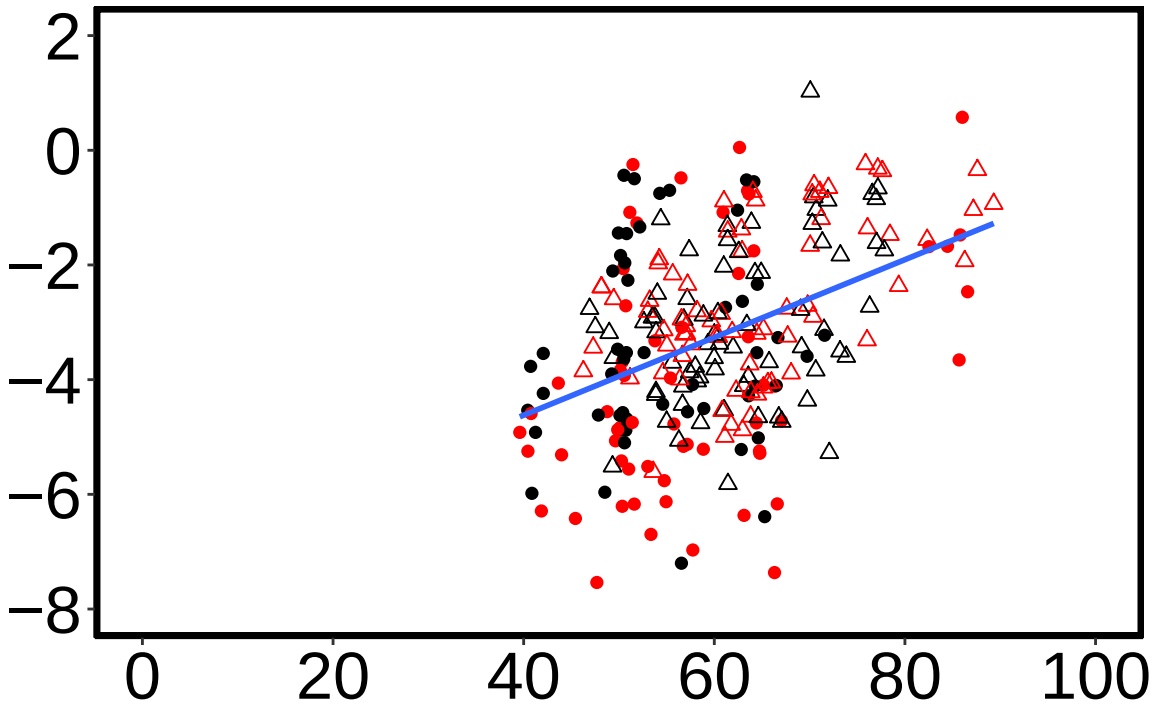



```
# stats for C
cistat <- lm(dftrans$d18O.corrected_Cond ~ dftrans$RH_al_mean)
summary(cistat)  #siginfiant
```

```
##
## Call:
## lm(formula = dftrans$d18O.corrected_Cond ~ dftrans$RH_al_mean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -4.5297 -1.1245 -0.0208  1.1065  3.6046
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -7.332338   0.576065 -12.728  < 2e-16 ***
## dftrans$RH_al_mean  0.067815   0.009429   7.193 7.24e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.53 on 252 degrees of freedom
## Multiple R-squared:  0.1703, Adjusted R-squared:  0.167
## F-statistic: 51.73 on 1 and 252 DF,  p-value: 7.239e-12
```

```
ci <- ggplot(data = dftrans, aes(x = RH_al_mean, y = d18O.corrected_Cond)) +
    geom_point(aes(colour = trt, shape = Camp), size = 2, ) +
    scale_color_manual(name = "Legend:", labels = c("Transpired Ambient Chamber",
        "Transpired Elevated Chamber"), values = c("black", "red")) +
    scale_shape_manual(name = "Legend:", labels = c("in", "out"),
        values = c(16, 2)) + labs(y = "", x = "", element_text(size = 6)) +
    geom_smooth(method = "lm", formula = y ~ x, se = FALSE) +
    # scale_x_datetime(breaks = scales::pretty_breaks(n = 12))
theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(-8, 2)) + scale_x_continuous(breaks = scales::pretty_breaks(n = 5),
    limits = c(0, 100)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 25)) + theme(plot.margin = unit(c(5, 5, 5, 5), "mm"))
ci
```




```
# VPD
amb$vpdatm <- RHtoVPD(RH = amb$RH_al, TdegC = amb$Tair_al, Pa = 101)
amb$vpdatm
```

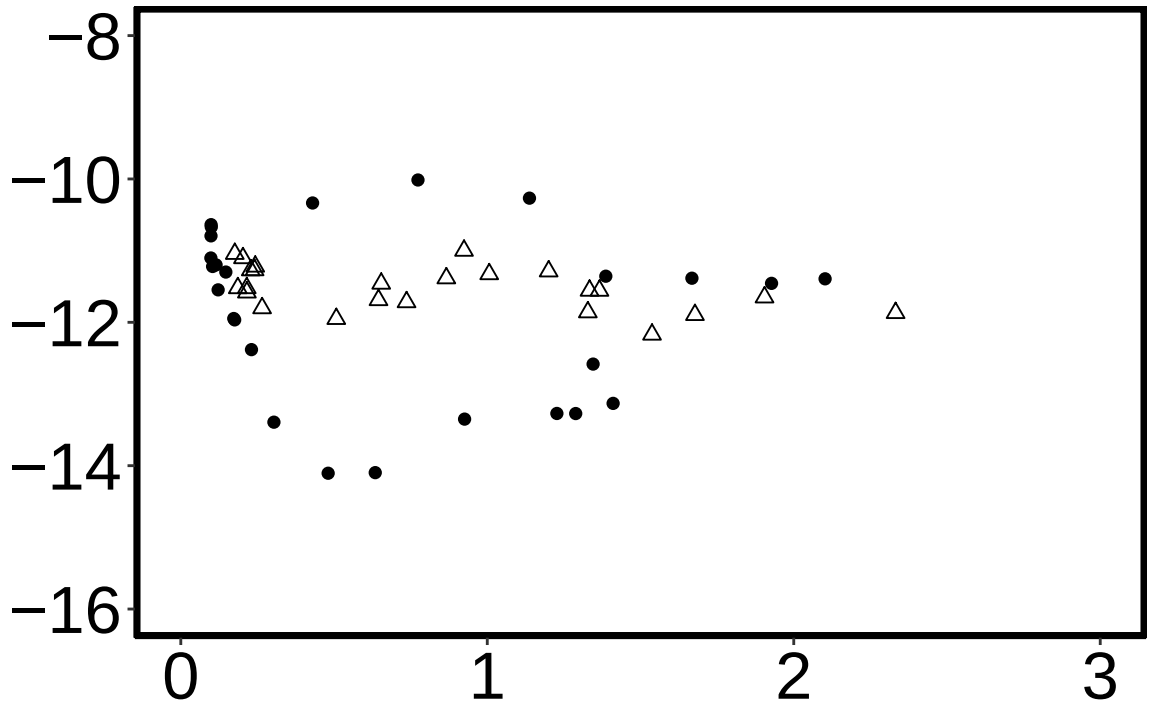
```
##  [1] 1.34531013 1.41062644 1.28854110 1.22706726 0.92594332 0.63445287
##  [7] 0.48078134 0.30383329 0.23060538 0.17277205 0.17578547 0.12180924
## [13] 0.09813477 0.09851787 0.09948217 0.09881199 0.11509009 0.10377743
## [19] 0.14696477 0.43000300 0.77386592 1.13766111 1.38678045 1.66770040
## [25] 1.92750577 2.10224011 1.32809682 1.53755754 1.67760762 1.36601660
## [31] 1.20030992 1.00637704 0.86640137 0.73659408 0.64505606 0.50730880
## [37] 0.26528472 0.21517432 0.22832205 0.24079919 0.24291401 0.20282338
## [43] 0.17613850 0.18597551 0.21563496 0.65392423 0.92414002 1.33371645
## [49] 1.90464159 2.33234543
```

```
estat <- lm(amb$d18O.corrected_WV.AMB. ~ amb$vpdatm)
summary(estat)  #NOT  significant
```

```
##
## Call:
## lm(formula = amb$d18O.corrected_WV.AMB. ~ amb$vpdatm)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4772 -0.2721  0.2627  0.4816  1.6845
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -11.5127     0.1996 -57.669   <2e-16 ***
## amb$vpdatm   -0.2406     0.2007  -1.199    0.236
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8938 on 48 degrees of freedom
## Multiple R-squared:  0.02908,    Adjusted R-squared:  0.008856
## F-statistic: 1.438 on 1 and 48 DF,  p-value: 0.2364
```

```
e <- ggplot(data = amb, aes(x = vpdatm, y = d18O.corrected_WV.AMB.)) +
    geom_point(aes(colour = trt, shape = Camp), size = 2, ) +
    scale_color_manual(name = "Legend:", labels = c("Transpired Ambient Chamber",
        "Transpired Elevated Chamber"), values = c("black", "red")) +
    scale_shape_manual(name = "Legend:", labels = c("in", "out"),
        values = c(16, 2)) + labs(y = "", x = "", element_text(size = 6)) +
    # scale_x_datetime(breaks = scales::pretty_breaks(n = 12))
theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(-16, -8)) + scale_x_continuous(breaks = scales::pretty_breaks(n = 4),
    limits = c(0, 3)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 25)) + theme(plot.margin = unit(c(5, 5, 5, 5), "mm"))
e
```



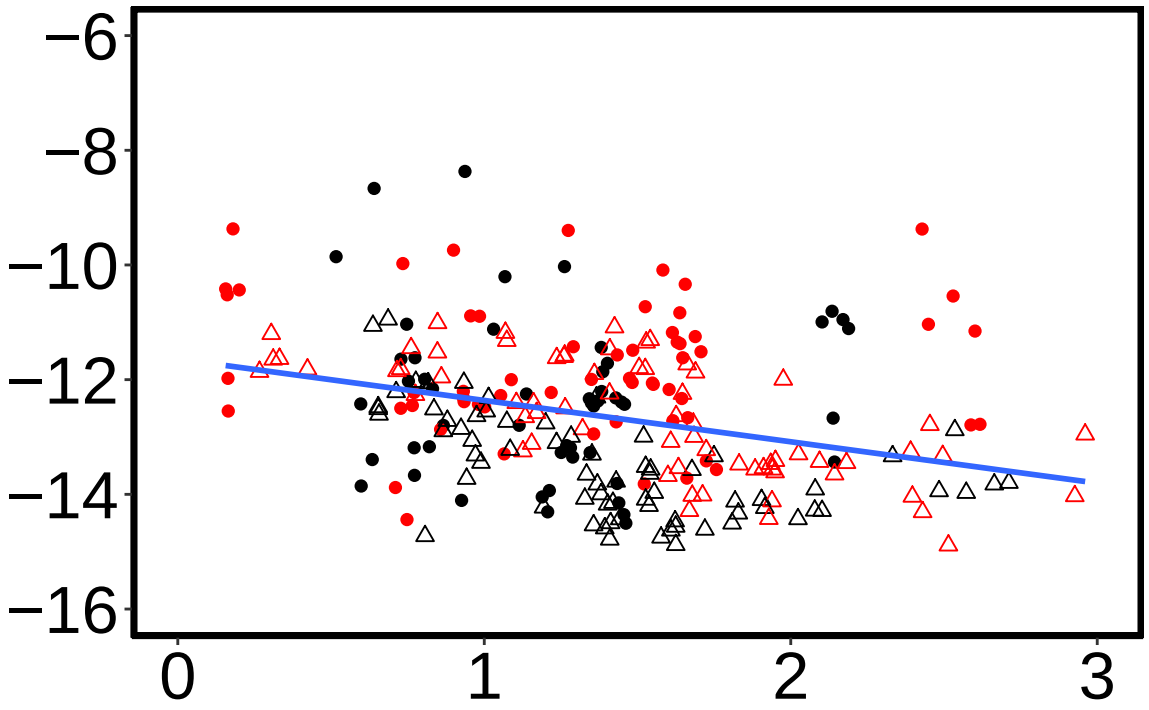
```r
fstat <- lm(dftrans$d18O.corrected_WV ~ dftrans$vpd)
summary(fstat)  #SIG
```

```
##
## Call:
## lm(formula = dftrans$d18O.corrected_WV ~ dftrans$vpd)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4797 -0.7958 -0.0961  0.7120  3.9640
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -11.7136     0.1939 -60.414  < 2e-16 ***
## dftrans$vpd  -0.6611     0.1263  -5.234  3.5e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.209 on 252 degrees of freedom
## Multiple R-squared:  0.09804,    Adjusted R-squared:  0.09446
## F-statistic: 27.39 on 1 and 252 DF,  p-value: 3.5e-07
```

```r
f <- ggplot(data = dftrans, aes(x = vpd, y = d18O.corrected_WV)) +
    geom_point(aes(colour = trt, shape = Camp), size = 2, ) +
    scale_color_manual(name = "Legend:", labels = c("Transpired Ambient Chamber",
        "Transpired Elevated Chamber"), values = c("black", "red")) +
    scale_shape_manual(name = "Legend:", labels = c("in", "out"),
        values = c(16, 2)) + labs(y = "", x = "", element_text(size = 6)) +
    geom_smooth(method = "lm", formula = y ~ x, se = FALSE) +
    # scale_x_datetime(breaks = scales::pretty_breaks(n = 12))
theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(-16, -6)) + scale_x_continuous(breaks = scales::pretty_breaks(n = 4),
    limits = c(0, 3)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 25)) + theme(plot.margin = unit(c(5, 5, 5, 5), "mm"))
f
```

```
## Warning: Removed 4 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 4 rows containing missing values (geom_point).
```




```
gstat <- lm(dftrans$dtrans ~ dftrans$vpd)
summary(gstat)  #SIG
```

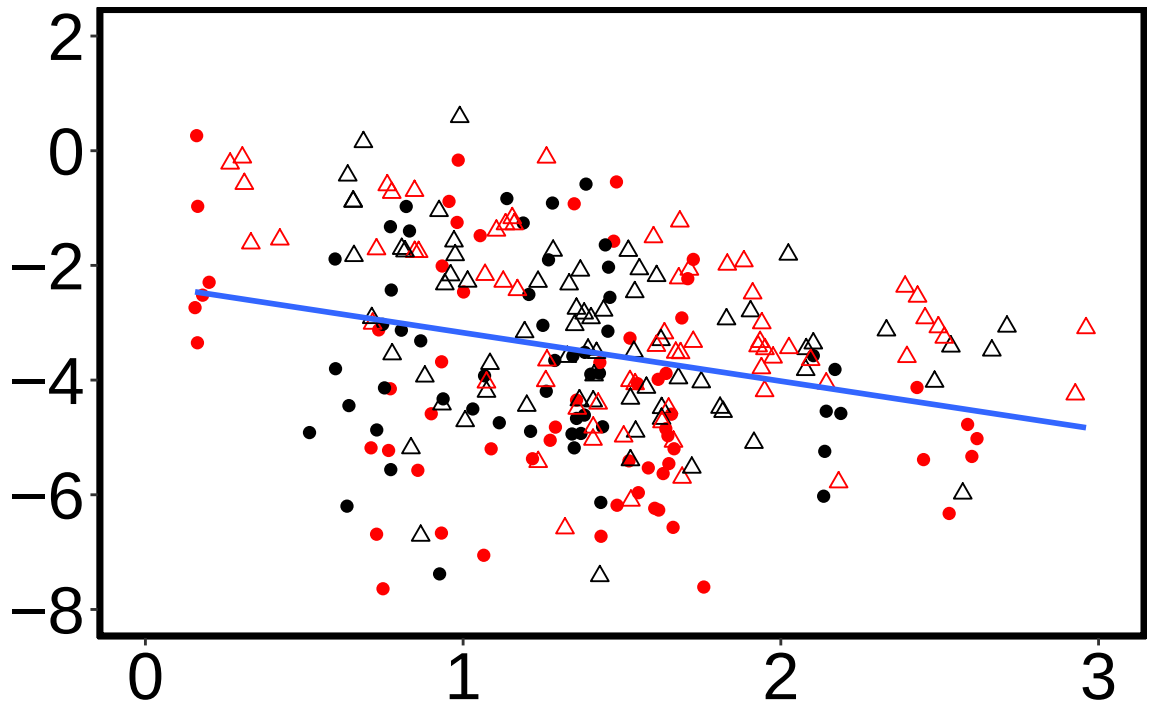
```
##
## Call:
## lm(formula = dftrans$dtrans ~ dftrans$vpd)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6222 -1.1659  0.0829  1.3057  3.7794
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.4769     0.2624  -9.439  < 2e-16 ***
## dftrans$vpd  -0.7221     0.1709  -4.224 3.35e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.636 on 252 degrees of freedom
## Multiple R-squared:  0.06612,    Adjusted R-squared:  0.06242
## F-statistic: 17.84 on 1 and 252 DF,  p-value: 3.354e-05
```

```
g <- ggplot(data = dftrans, aes(x = vpd, y = dtrans)) + geom_point(aes(colour = trt,
    shape = Camp), size = 2, ) + scale_color_manual(name = "Legend:",
    labels = c("Transpired Ambient Chamber", "Transpired Elevated Chamber"),
```

```
    values = c("black", "red")) + scale_shape_manual(name = "Legend:",
    labels = c("in", "out"), values = c(16, 2)) + labs(y = "",
    x = "", element_text(size = 6)) + geom_smooth(method = "lm",
    formula = y ~ x, se = FALSE) + # scale_x_datetime(breaks = scales::pretty_breaks(n = 12))
theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(-8, 2)) + scale_x_continuous(breaks = scales::pretty_breaks(n = 4),
    limits = c(0, 3)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 25)) + theme(plot.margin = unit(c(5, 5, 5, 5), "mm"))
g
```

```
## Warning: Removed 4 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 4 rows containing missing values (geom_point).
```




```
gistat <- lm(dftrans$d18O.corrected_Cond ~ dftrans$vpd)
summary(gistat)  #SIG
```

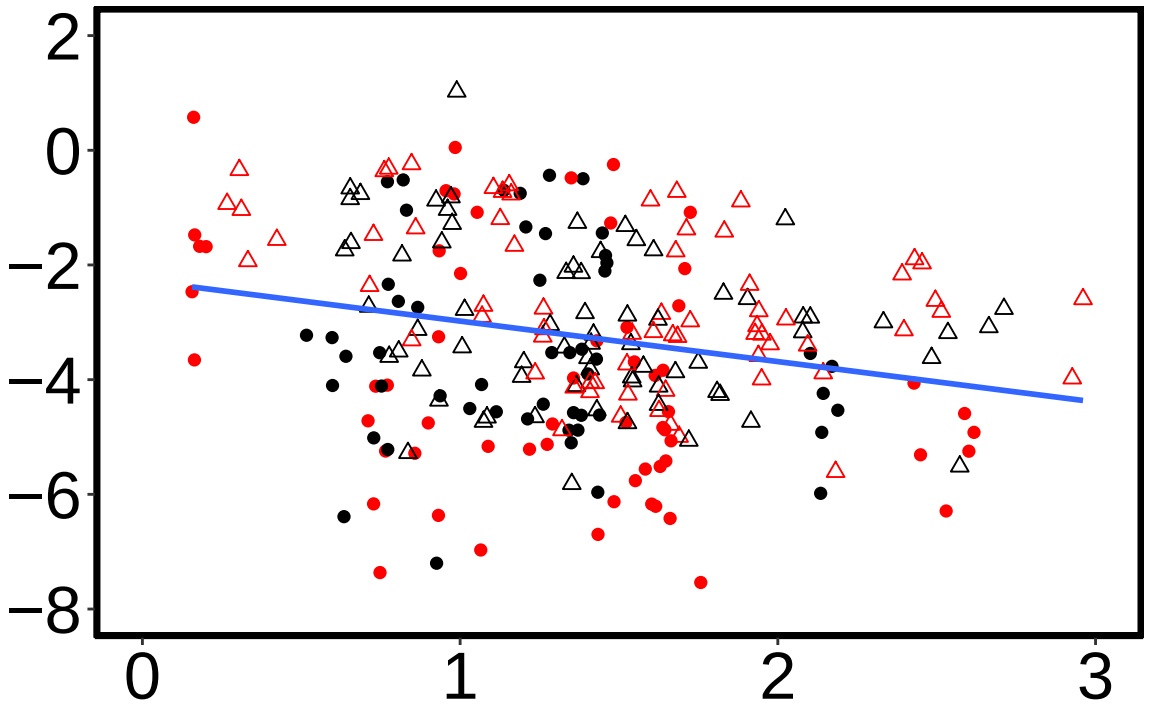
```
##
## Call:
## lm(formula = dftrans$d18O.corrected_Cond ~ dftrans$vpd)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5127 -1.2532  0.0034  1.2629  4.0197
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.4077     0.2631  -9.150  < 2e-16 ***
## dftrans$vpd  -0.5940     0.1714  -3.465 0.000622 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.641 on 252 degrees of freedom
## Multiple R-squared:  0.04549,    Adjusted R-squared:  0.0417
## F-statistic: 12.01 on 1 and 252 DF,  p-value: 0.0006223
```

```
gi <- ggplot(data = dftrans, aes(x = vpd, y = d18O.corrected_Cond)) +
    geom_point(aes(colour = trt, shape = Camp), size = 2, ) +
    scale_color_manual(name = "Legend:", labels = c("Transpired Ambient Chamber",
        "Transpired Elevated Chamber"), values = c("black", "red")) +
    scale_shape_manual(name = "Legend:", labels = c("in", "out"),
        values = c(16, 2)) + labs(y = "", x = "", element_text(size = 6)) +
    geom_smooth(method = "lm", formula = y ~ x, se = FALSE) +
    # scale_x_datetime(breaks = scales::pretty_breaks(n = 12))
theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(-8, 2)) + scale_x_continuous(breaks = scales::pretty_breaks(n = 4),
    limits = c(0, 3)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 25)) + theme(plot.margin = unit(c(5, 5, 5, 5), "mm"))
gi
```

```
## Warning: Removed 4 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 4 rows containing missing values (geom_point).
```

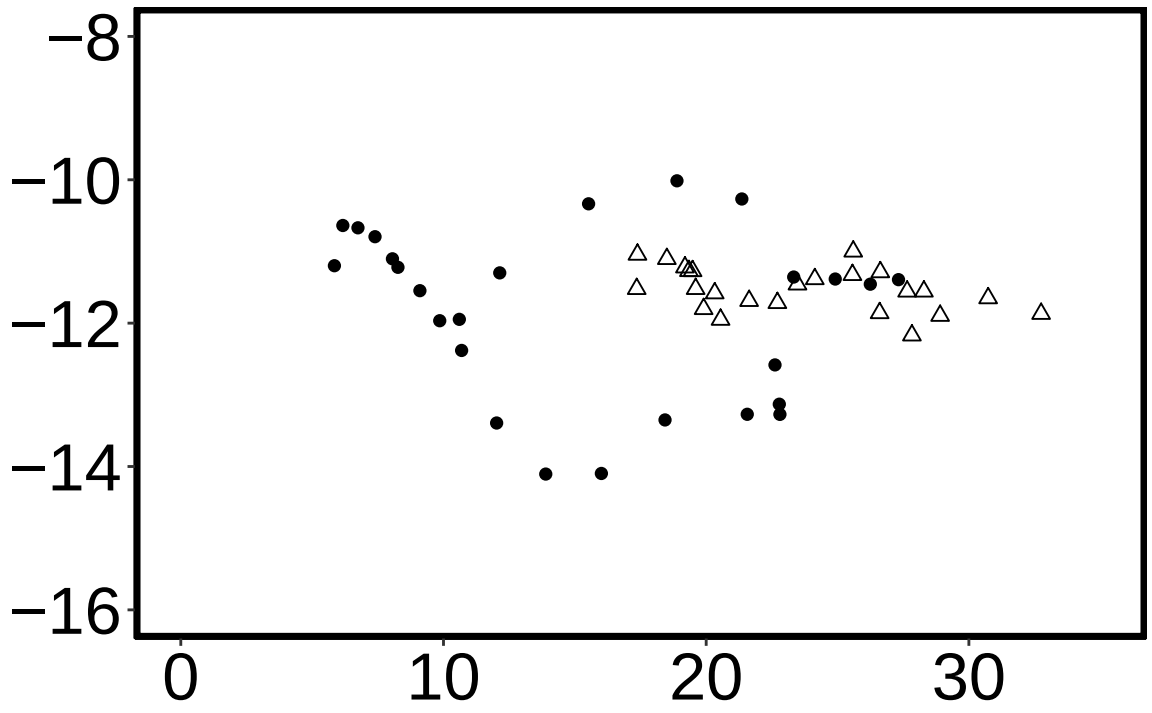



```
# Temp
istat <- lm(amb$d18O.corrected_WV.AMB. ~ amb$Taref_al_mean)
summary(istat)   # NOT
```

```
##
## Call:
## lm(formula = amb$d18O.corrected_WV.AMB. ~ amb$Taref_al_mean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4612 -0.2074  0.2253  0.4310  1.6792
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -11.50758    0.37294 -30.856   <2e-16 ***
## amb$Taref_al_mean  -0.00985    0.01812  -0.544    0.589
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9043 on 48 degrees of freedom
## Multiple R-squared:  0.006119,   Adjusted R-squared:  -0.01459
## F-statistic: 0.2955 on 1 and 48 DF,  p-value: 0.5892
```

```
i <- ggplot(data = amb, aes(x = Taref_al_mean, y = d18O.corrected_WV.AMB.)) +
    geom_point(aes(colour = trt, shape = Camp), size = 2, ) +
```

```
    scale_color_manual(name = "Legend:", labels = c("Transpired Ambient Chamber",
        "Transpired Elevated Chamber"), values = c("black", "red")) +
    scale_shape_manual(name = "Legend:", labels = c("in", "out"),
        values = c(16, 2)) + labs(y = "", x = "", element_text(size = 6)) +
    # scale_x_datetime(breaks = scales::pretty_breaks(n = 12))
theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(-16, -8)) + scale_x_continuous(breaks = scales::pretty_breaks(n = 5),
    limits = c(0, 35)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 25)) + theme(plot.margin = unit(c(5, 5, 5, 5), "mm"))
i
```

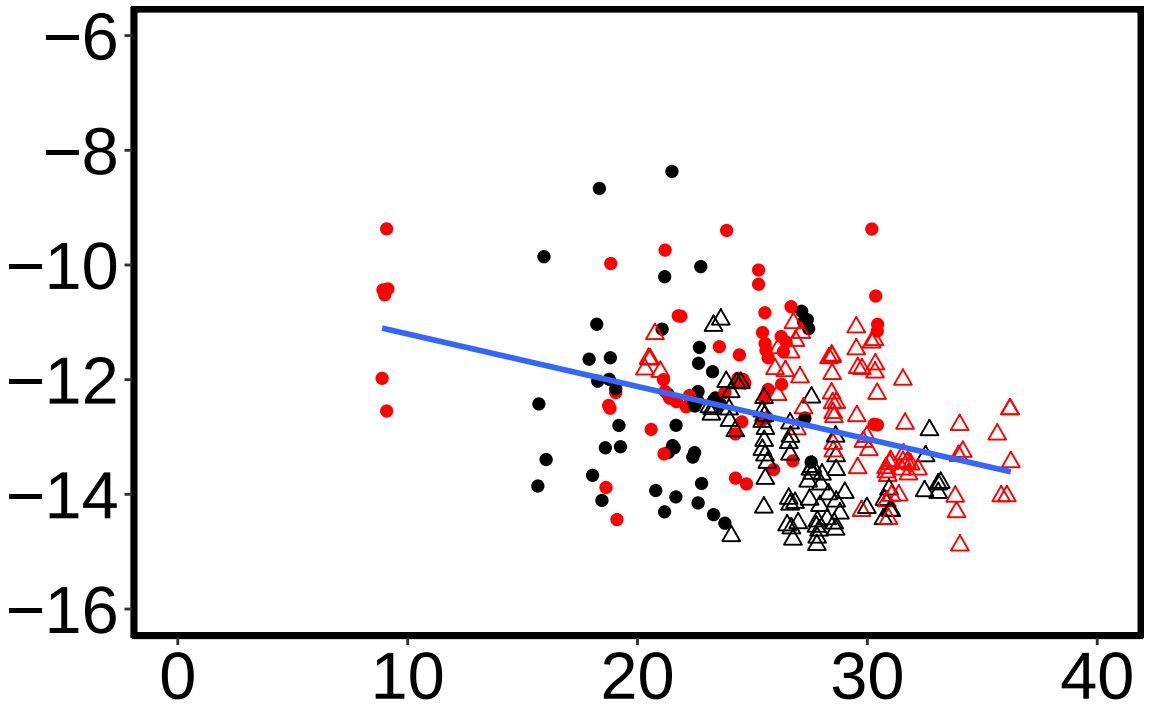



```
# Jstat
jstat <- lm(dftrans$d18O.corrected_WV ~ dftrans$Tair_al_mean)
summary(jstat)  #SIGNIFANT
```

```
##
## Call:
## lm(formula = dftrans$d18O.corrected_WV ~ dftrans$Tair_al_mean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4026 -0.8490 -0.0747  0.6654  3.8876
```

```
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -10.28714    0.38666 -26.605  < 2e-16 ***
## dftrans$Tair_al_mean  -0.09161    0.01473  -6.221 2.05e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.185 on 252 degrees of freedom
## Multiple R-squared:  0.1331, Adjusted R-squared:  0.1297
## F-statistic:  38.7 on 1 and 252 DF,  p-value: 2.049e-09
```

```
j <- ggplot(data = dftrans, aes(x = Tair_al_mean, y = d18O.corrected_WV)) +
    geom_point(aes(colour = trt, shape = Camp), size = 2, ) +
    scale_color_manual(name = "Legend:", labels = c("Transpired Ambient Chamber",
        "Transpired Elevated Chamber"), values = c("black", "red")) +
    scale_shape_manual(name = "Legend:", labels = c("in", "out"),
        values = c(16, 2)) + labs(y = "", x = "", element_text(size = 6)) +
    geom_smooth(method = "lm", formula = y ~ x, se = FALSE) +
    # scale_x_datetime(breaks = scales::pretty_breaks(n = 12))
theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(-16, -6)) + scale_x_continuous(breaks = scales::pretty_breaks(n = 5),
    limits = c(0, 40)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 25)) + theme(plot.margin = unit(c(5, 5, 5, 5), "mm"))
j
```

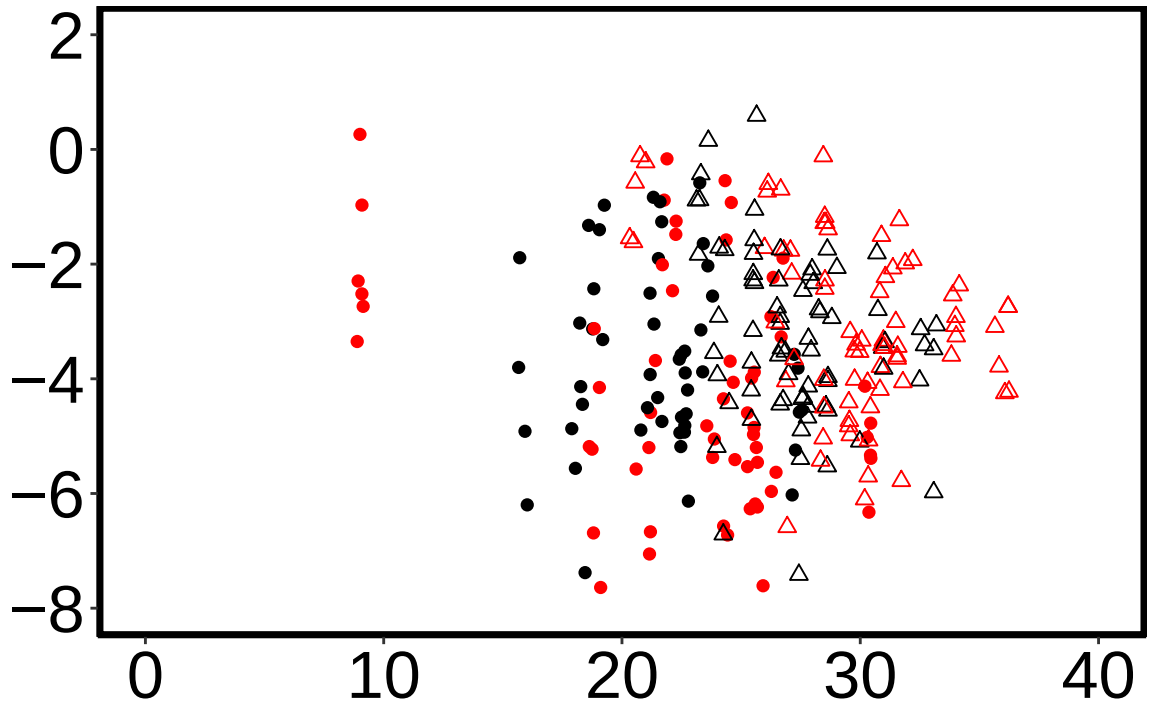



```
kstat <- lm(dftrans$dtrans ~ dftrans$Tair_al_mean)
summary(kstat)   #NOT
```

```
##
## Call:
## lm(formula = dftrans$dtrans ~ dftrans$Tair_al_mean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2966 -1.1806 -0.0102  1.3093  4.0824
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -2.89903    0.55101  -5.261 3.06e-07 ***
## dftrans$Tair_al_mean -0.02321    0.02099  -1.106     0.27
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.689 on 252 degrees of freedom
## Multiple R-squared:  0.00483,    Adjusted R-squared:  0.0008807
## F-statistic: 1.223 on 1 and 252 DF,  p-value: 0.2698
```

```
k <- ggplot(data = dftrans, aes(x = Tair_al_mean, y = dtrans)) +
    geom_point(aes(colour = trt, shape = Camp), size = 2, ) +
    scale_color_manual(name = "Legend:", labels = c("Transpired Ambient Chamber",
```

```
        "Transpired Elevated Chamber"), values = c("black", "red")) +
    scale_shape_manual(name = "Legend:", labels = c("in", "out"),
        values = c(16, 2)) + labs(y = "", x = "", element_text(size = 6)) +
    # scale_x_datetime(breaks = scales::pretty_breaks(n = 12))
theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(-8, 2)) + scale_x_continuous(breaks = scales::pretty_breaks(n = 5),
    limits = c(0, 40)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 25)) + theme(plot.margin = unit(c(5, 5, 5, 5), "mm"))
k
```

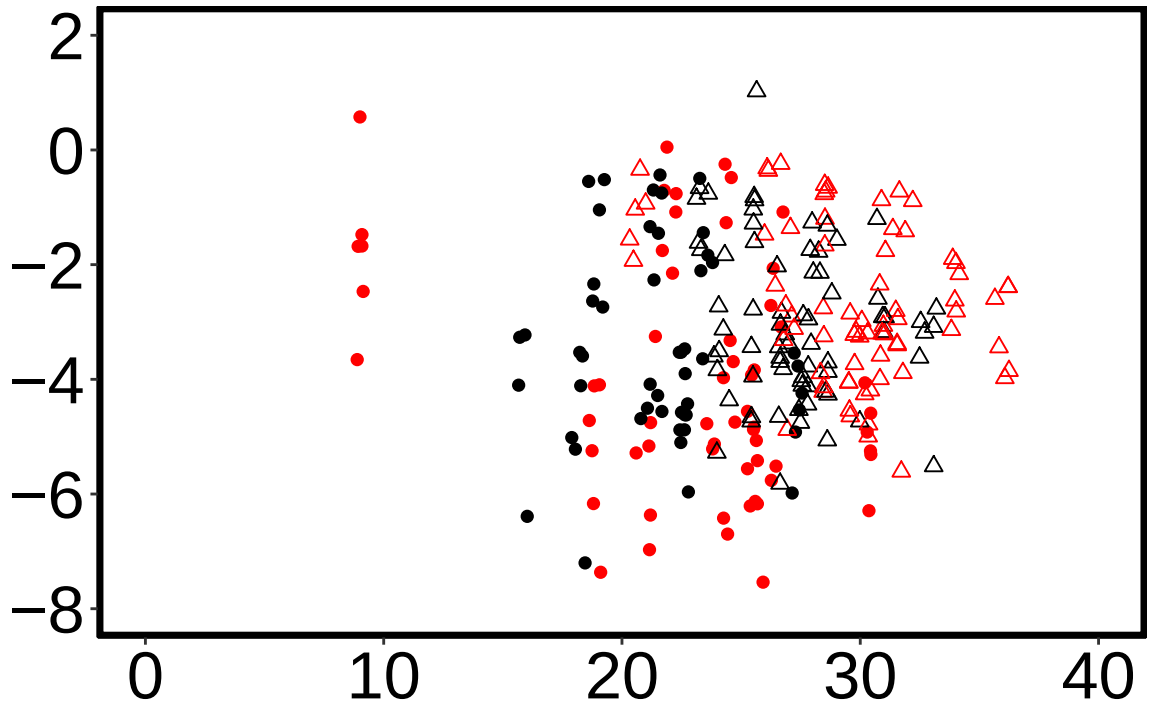



```
kistat <- lm(dftrans$d18O.corrected_Cond ~ dftrans$Tair_al_mean)
summary(kistat)  #NOT
```

```
##
## Call:
## lm(formula = dftrans$d18O.corrected_Cond ~ dftrans$Tair_al_mean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2895 -1.2780 -0.0934  1.3722  4.2706
##
## Coefficients:
```

```
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -3.114977   0.547781  -5.687 3.58e-08 ***
## dftrans$Tair_al_mean -0.005119   0.020864  -0.245    0.806
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.679 on 252 degrees of freedom
## Multiple R-squared:  0.0002388,  Adjusted R-squared:  -0.003728
## F-statistic: 0.0602 on 1 and 252 DF,  p-value: 0.8064
```

```
ki <- ggplot(data = dftrans, aes(x = Tair_al_mean, y = d18O.corrected_Cond)) +
    geom_point(aes(colour = trt, shape = Camp), size = 2, ) +
    scale_color_manual(name = "Legend:", labels = c("Transpired Ambient Chamber",
        "Transpired Elevated Chamber"), values = c("black", "red")) +
    scale_shape_manual(name = "Legend:", labels = c("in", "out"),
        values = c(16, 2)) + labs(y = "", x = "", element_text(size = 6)) +
    # scale_x_datetime(breaks = scales::pretty_breaks(n = 12))
theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(-8, 2)) + scale_x_continuous(breaks = scales::pretty_breaks(n = 5),
    limits = c(0, 40)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 25)) + theme(plot.margin = unit(c(5, 5, 5, 5), "mm"))
ki
```



```
rh <- plot_grid(a, b, c, ci, align = "vh", ncol = 1)
vpd <- plot_grid(e, f, g, gi, align = "vh", ncol = 1)
```

```
## Warning: Removed 4 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 4 rows containing missing values (geom_point).
```

```
## Warning: Removed 4 rows containing non-finite values (stat_smooth).
```

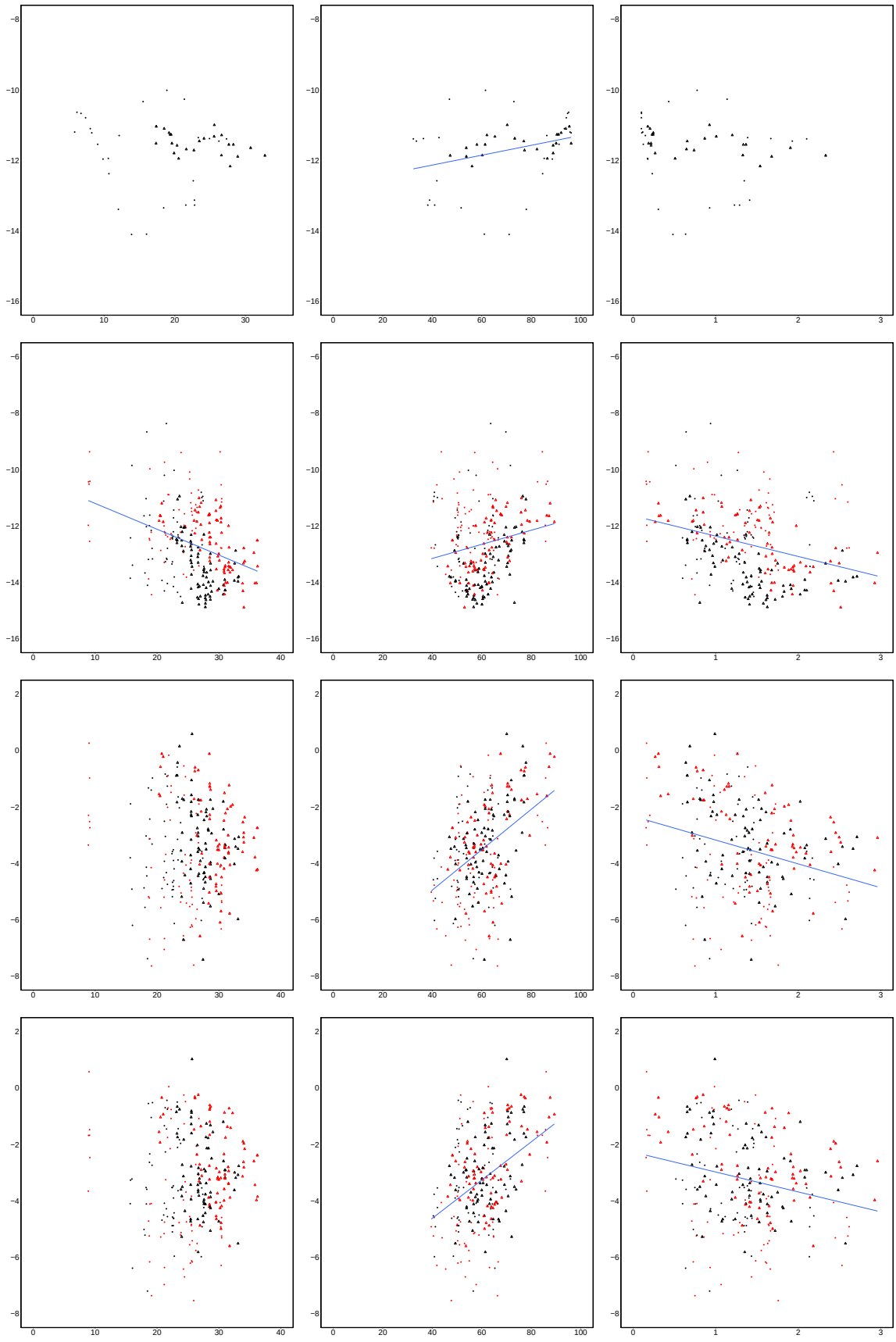
```
## Warning: Removed 4 rows containing missing values (geom_point).
```

```
## Warning: Removed 4 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 4 rows containing missing values (geom_point).
```

```
temp <- plot_grid(i, j, k, ki, align = "vh", ncol = 1)
figure4 = plot_grid(temp, rh, vpd, align = "vh", nrow = 1)
# ggsave('newgraphcelluloseyaxisaddcond.tiff', width = 40,
# height = 60, units = 'cm', dpi=300) again i used illustator
# to finish off
```

```
plot(figure4)
```



#That finishes the campaign data

```
rm(list = ls())
```

#The next data we investigate is d18O cellulose, assimilation weighted temperture, VPD and RH, trunk growth and our PEX proxy. # I used John Drakes code to calculate assimilation weighted values #https://github.com/jedrake/wtc4_flux/blob/master/R/combine_data_plot_assimiationWeightedLeafT.R

```
require(lubridate)
require(ggplot2)
require(xts)
require(dplyr)
require(grid)
require(cowplot)
require(ggpubr)
require(plantecophys)
require(doBy)

# read in flux data to calculate CO2 assimilation
googledriveWTC4dfID <- "10Y5-KjrbqeSaG6b_c6N7dJFGMsXSYqyq"
df <- read.csv(sprintf("https://docs.google.com/uc?id=%s&export=download",
    googledriveWTC4dfID))  #best to load this in andthe # it out
## as it as an all vars file for whole period so quite large.
df["Td"] <- (dmy_hm(df$DateTime, quiet = TRUE, tz = "UTC"))
str(df$Td)
```

```
##  POSIXct[1:149780], format: "2016-02-28 00:00:00" "2016-02-28 00:00:00" "2016-02-28 00:00:00" ...
```

```
df$halfhour = cut(df$Td, breaks = "60 min")
df$Date <- as.Date(df$Td)

# Dates to get data
period1start <- as.Date("2016-06-07")
period1end <- as.Date("2016-07-19")
period2start <- as.Date("2016-08-25")
period2end <- as.Date("2016-9-26")
period3start <- as.Date("2016-10-23")
period3end <- as.Date("2016-11-23")

# Calculate Flux if co2 flux is less then zero mulriiply it
# by 3 - this is respiration
df$netresp <- ifelse(df$FluxCO2 < 0, df$FluxCO2 * 3, NA)
# is co2 flux is greater then zero this is assimliaton
df$netasim <- ifelse(df$FluxCO2 > 0, df$FluxCO2, NA)

p1 <- subset(df, Date >= period1start & Date <= period1end)
p2 <- subset(df, Date >= period2start & Date <= period2end)
p3 <- subset(df, Date >= period3start & Date <= period3end)

p1sum <- summaryBy(. ~ chamber + T_treatment, FUN = c(sum), keep.names = T,
    data = p1, na.rm = TRUE)
p1sum$flux <- ((p1sum$netasim + p1sum$netresp) * 1440)/1000
p1sum$Period = 1
```

```
p2sum <- summaryBy(. ~ chamber + T_treatment, FUN = c(sum), keep.names = T,
    data = p2, na.rm = TRUE)
p2sum$flux <- ((p2sum$netasim + p2sum$netresp) * 1440)/1000
p2sum$Period = 2

p3sum <- summaryBy(. ~ chamber + T_treatment, FUN = c(sum), keep.names = T,
    data = p3, na.rm = TRUE)
p3sum$flux <- ((p3sum$netasim + p3sum$netresp) * 1440)/1000
p3sum$Period = 3

dfFLUX <- rbind(p1sum, p2sum, p3sum)
# get growth data
googledrivestemdfID <- "1g1QXzgqcEgUH3n94BagWP0b6rBwY8EYc"
dfstem <- read.csv(sprintf("https://docs.google.com/uc?id=%s&export=download",
    googledrivestemdfID))
dfPEX <- merge(dfFLUX, dfstem, by = c("chamber", "Period"))
# get tree volume
dfPEX$sdiamM = dfPEX$Start.diam.mm./1000  #start diameter into meters
dfPEX$sheightM = dfPEX$start.Height.cm.../100  #start height in meters
# assume cone
dfPEX$ediamM = dfPEX$end.diam/1000  #end diameter into meters
dfPEX$eheightM = dfPEX$end.height/100  #end height in meters
dfPEX$vstart = (dfPEX$sheightM/3) * 3.146 * ((dfPEX$sdiamM/2)^2)  #volume start
dfPEX$vend = (dfPEX$eheightM/3) * 3.146 * ((dfPEX$ediamM/2)^2)  #volume end
dfPEX$volincrease = dfPEX$vend - dfPEX$vstart
dfPEX$PEX = dfPEX$netasim/dfPEX$volincrease
dfPEX$PEX
```

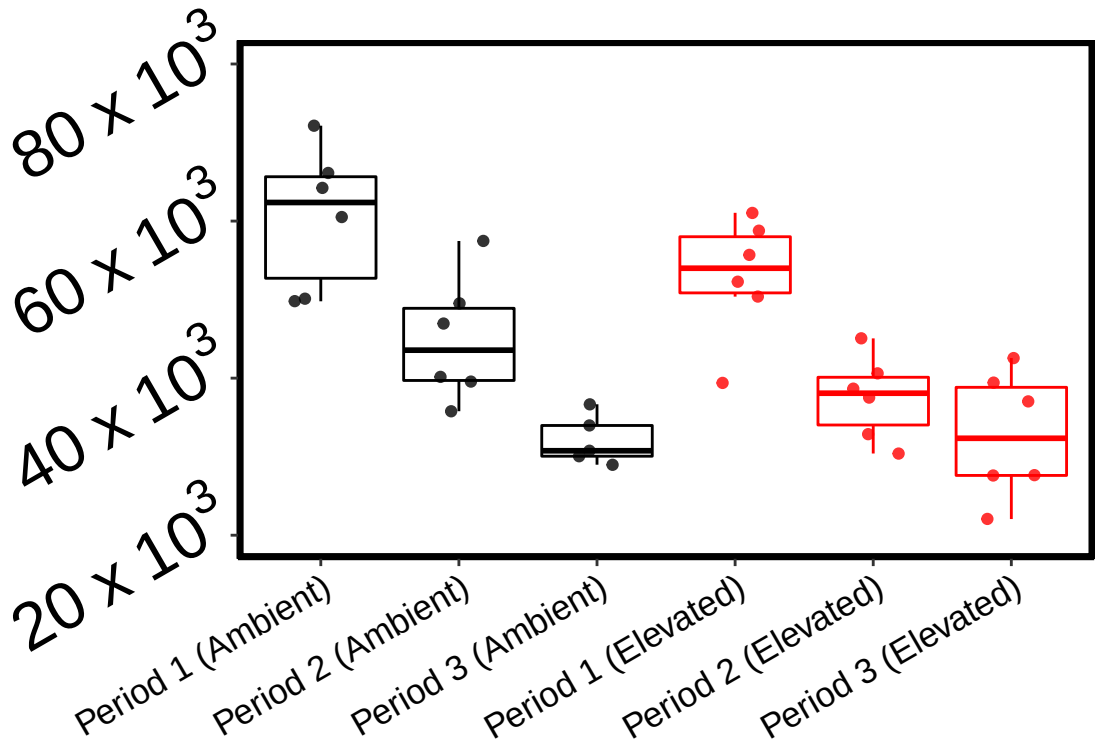
```
##  [1] 72142.78 39550.66 28976.90 58769.56 40599.87 27602.16 64218.86 46959.88
##  [9] 30061.42 39375.95 30405.42 27648.37 66118.37 49537.91 30755.29 52281.86
## [17] 37515.67 22058.06 60512.29 35789.56 33974.40 50380.75 32873.11 39426.77
## [25] 50121.17 40162.80 36672.11 61050.81 38650.33 42561.46 49784.37 57462.81
## [33] 55707.26 45062.21 37041.48
```

```
dfPEX$id <- paste(dfPEX$T_treatment.x, dfPEX$END.Date)
dfPEXid <- as.factor(dfPEX$id)
str(dfPEX$id)
```

```
##  chr [1:35] "ambient 20/07/2016" "ambient 28/09/2016" "ambient 23/11/2016" ...
```

```
labelsy = c("Period 1 (Ambient)", "Period 2 (Ambient)", "Period 3 (Ambient)",
    "Period 1 (Elevated)", "Period 2 (Elevated)", "Period 3 (Elevated)")
labels = c(expression(20 ~ x ~ 10^3), expression(40 ~ x ~ 10^3),
    expression(60 ~ x ~ 10^3), expression(80 ~ x ~ 10^3))
#############
ggboxplot(dfPEX, x = "id", y = "PEX", color = "T_treatment.x",
    palette = c("black", "red"), width = 0.8, outlier.shape = NA) +
    geom_jitter(aes(colour = T_treatment.x), width = 0.2, alpha = 0.8) +
    theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + labs(y = "",
    x = "", fill = "Technique") + theme(axis.title.y = element_text(angle = 90,
    vjust = 0, hjust = 0.5, size = 18)) + scale_y_continuous(labels = labels,
```

```
    limits = c(20000, 80000)) + scale_x_discrete(labels = labelsy) +
    theme(axis.text.y = element_text(angle = 30, vjust = 1, hjust = 0.5,
        size = 25, color = "Black")) + theme(axis.text.x = element_text(angle = 30,
    vjust = 1, hjust = 1, size = 13, color = "Black")) + theme(legend.position = "none") +
    labs(x = "", y = "", element_text(size = 30)) + theme(plot.margin = unit(c(5,
    5, 5, 5), "mm"))
```




```
dfPEX$Period <- as.factor(dfPEX$Period)
str(dfPEX)
```

```
## 'data.frame':    35 obs. of  46 variables:
##  $ chamber           : chr  "C01" "C01" "C01" "C02" ...
##  $ Period            : Factor w/ 3 levels "1","2","3": 1 2 3 1 2 3 1 2 3 1 ...
##  $ T_treatment.x     : chr  "ambient" "ambient" "ambient" "elevated" ...
##  $ DIFFP             : num  199602 148939 117911 196577 155189 ...
##  $ AIRPRESS          : num  197895 155934 154846 199517 156344 ...
##  $ Vwat              : num  6.5 5.69 6.3 7.87 7.11 ...
##  $ Fwat              : num  6.31 5.54 5.96 6.52 5.91 ...
##  $ Href              : num  8.23e+15 1.81e+06 2.24e+06 8.23e+15 1.81e+06 ...
##  $ kfactor           : num  1477 1164 1161 1547 1213 ...
##  $ CONDH2O           : num  7.55 11.93 18.22 5 10.37 ...
##  $ RH_al             : num  149587 114850 104339 143659 110402 ...
##  $ DPref_al          : num  33762833 13280 18254 33762864 13334 ...
##  $ DewPntC           : num  12360 13711 17036 12376 13754 ...
##  $ Taref_al          : num  33772819 21526 30186 33773085 21607 ...
```

```
##  $ RHref_al          : num  33899460 113901 101372 33900046 114196 ...
##  $ FluxCO2           : num  42 51.3 71.1 30.9 50.2 ...
##  $ FluxH2O           : num  7.76 12.14 18.19 6.39 11.62 ...
##  $ Tair_al           : num  21940 21590 30048 27885 26201 ...
##  $ PAR               : num  295122 416104 723869 307143 418626 ...
##  $ VPD               : num  692 687 1387 936 904 ...
##  $ TargTempC_Avg     : num  23237 22553 31188 28540 26414 ...
##  $ PPFD_Avg          : num  345492 455386 554913 331416 401176 ...
##  $ LeafT_Avg.1.      : num  9209 21361 29056 0 0 ...
##  $ LeafT_Avg.2.      : num  9779 22403 29438 0 0 ...
##  $ week              : num  51064 56540 69046 51476 56688 ...
##  $ netresp           : num  -26 -26.1 -25.9 -22.1 -25.8 ...
##  $ netasim           : num  50.6 60 79.8 38.3 58.7 ...
##  $ flux              : num  35.5 48.7 77.6 23.3 47.5 ...
##  $ T_treatment.y     : chr  "ambient" "ambient" "ambient" "elevated" ...
##  $ HW_treatment      : chr  "control" "control" "control" "control" ...
##  $ Start.diam.mm.    : num  36.8 46.6 56.2 27.9 38 47.7 32.8 41 48.1 40 ...
##  $ start.Height.cm...: num  508 552 642 523 632 ...
##  $ end.diam          : num  42.5 53.8 63.2 33.5 45.1 54.4 37.8 46 54.2 47.8 ...
##  $ end.height        : int  529 614 771 584 720 915 460 559 702 638 ...
##  $ d18Ocellulose     : num  32.7 34.3 34.8 33.4 33.7 ...
##  $ START.Date        : chr  "8/06/2016" "17/08/2016" "12/10/2016" "8/06/2016" ...
##  $ END.Date          : chr  "20/07/2016" "28/09/2016" "23/11/2016" "20/07/2016" ...
##  $ sdiamM            : num  0.0368 0.0466 0.0562 0.0279 0.038 0.0477 0.0328 0.041 0.0481 0.04 ...
##  $ sheightM          : num  5.08 5.52 6.42 5.23 6.32 ...
##  $ ediamM            : num  0.0425 0.0538 0.0632 0.0335 0.0451 0.0544 0.0378 0.046 0.0542 0.0478 ...
##  $ eheightM          : num  5.29 6.14 7.71 5.84 7.2 9.15 4.6 5.59 7.02 6.38 ...
##  $ vstart            : num  0.0018 0.00314 0.00532 0.00107 0.00239 ...
##  $ vend              : num  0.00251 0.00466 0.00807 0.00172 0.00384 ...
##  $ volincrease       : num  0.000701 0.001517 0.002753 0.000651 0.001447 ...
##  $ PEX               : num  72143 39551 28977 58770 40600 ...
##  $ id                : chr  "ambient 20/07/2016" "ambient 28/09/2016" "ambient 23/11/2016" "elevated
```

```
pexamb <- filter(dfPEX, dfPEX$T_treatment.x == "ambient")
pexhot <- filter(dfPEX, dfPEX$T_treatment.x == "elevated")
pexstat <- lm(pexamb$d18Ocellulose ~ pexamb$PEX)
summary(pexstat)  #ambinet d18 vs pex
```

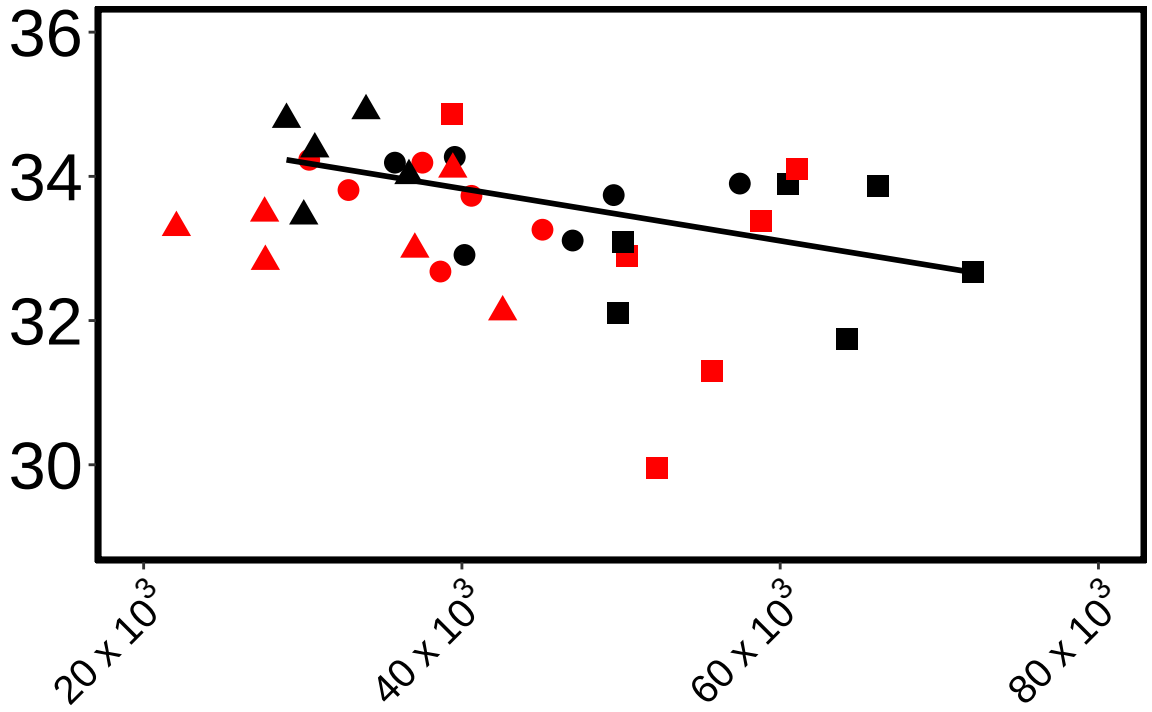
```
##
## Call:
## lm(formula = pexamb$d18Ocellulose ~ pexamb$PEX)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.3760 -0.4682  0.2072  0.5604  0.9757
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.528e+01  6.719e-01  52.509   <2e-16 ***
## pexamb$PEX  -3.622e-05  1.386e-05  -2.614   0.0196 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7575 on 15 degrees of freedom
```

```
## Multiple R-squared:  0.3129, Adjusted R-squared:  0.2671
## F-statistic: 6.831 on 1 and 15 DF,  p-value: 0.01956
```

```
pexstathot <- lm(pexhot$d18Ocellulose ~ pexhot$PEX)
summary(pexstathot)  #ambinet d18 vs pex
```

```
##
## Call:
## lm(formula = pexhot$d18Ocellulose ~ pexhot$PEX)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8544 -0.5624  0.1191  0.7681  1.6270
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.454e+01  1.055e+00  32.741 4.32e-16 ***
## pexhot$PEX  -3.321e-05  2.485e-05  -1.337      0.2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.139 on 16 degrees of freedom
## Multiple R-squared:  0.1004, Adjusted R-squared:  0.0442
## F-statistic: 1.786 on 1 and 16 DF,  p-value: 0.2001
```

```
pex <- ggplot(data = dfPEX, aes(x = PEX, y = d18Ocellulose)) +
    geom_point(aes(colour = T_treatment.x, shape = Period), size = 3.5,
        ) + scale_color_manual(name = "Legend:", labels = c("Transpired Ambient Chamber",
    "Transpired Elevated Chamber"), values = c("black", "red")) +
    scale_shape_manual(name = "Legend:", values = c(15, 16, 17)) +
    geom_smooth(data = pexamb, method = "lm", formula = y ~ x,
        se = FALSE, colour = "black") + labs(y = "", x = "",
    element_text(size = 6)) + # scale_x_datetime(breaks = scales::pretty_breaks(n = 12))
theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(29, 36)) + scale_x_continuous(labels = labels,
    limits = c(20000, 80000)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 15, angle = 45, hjust = 1)) + theme(plot.margin = unit(c(5,
    5, 5, 5), "mm"))
pex
```



#We now plot the other cellulose data

```
# this is the data frame with d18o cellulose and the
# assimliation weighted values
googledriveWTC4ASWID <- "1Aq-2eFUC8xZHyacd88ePAhAwzfK4anCE"
dfcel <- read.csv(sprintf("https://docs.google.com/uc?id=%s&export=download",
    googledriveWTC4ASWID))  #best to load this in andthe # it out
# dfcel<-summaryBy(.~T_treatment+Area,FUN=c(mean),keep.names=T,data=dfcel,na.rm=TRUE)
dfamb <- filter(dfcel, dfcel$T_treatment == "ambient")
dfhot <- filter(dfcel, dfcel$T_treatment == "elevated")

##### Relative Humidty
rhastat <- lm(dfamb$d18O ~ dfamb$RHco2fluxweighted)
summary(rhastat)  #ambinet d18 vs rhassimlatino wighted SIG
```

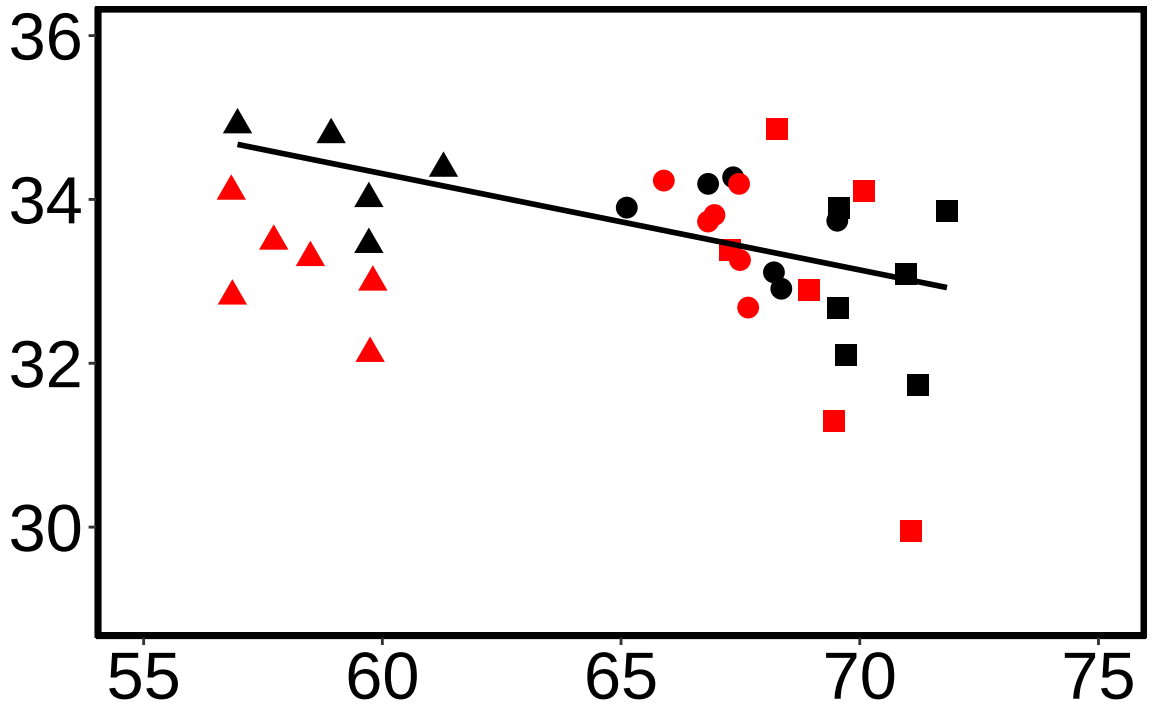
```
##
## Call:
## lm(formula = dfamb$d18O ~ dfamb$RHco2fluxweighted)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2567 -0.4222  0.1871  0.5457  0.9359
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                41.37156    2.35160  17.593 2.01e-11 ***
## dfamb$RHco2fluxweighted -0.11761    0.03545  -3.318  0.00469 **
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.694 on 15 degrees of freedom
## Multiple R-squared:  0.4232, Adjusted R-squared:  0.3848
## F-statistic: 11.01 on 1 and 15 DF,  p-value: 0.004687
```

```
rhastathot <- lm(dfhot$d18O ~ dfhot$RHco2fluxweighted)
summary(rhastathot)  #ambinet d18 vs rhassimlatino wighted SIG
```

```
##
## Call:
## lm(formula = dfhot$d18O ~ dfhot$RHco2fluxweighted)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0062 -0.3889  0.1193  0.6911  1.8044
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)               35.46995    3.74870   9.462  5.9e-08 ***
## dfhot$RHco2fluxweighted -0.03537    0.05767  -0.613    0.548
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.187 on 16 degrees of freedom
## Multiple R-squared:  0.02297,    Adjusted R-squared:  -0.03809
## F-statistic: 0.3762 on 1 and 16 DF,  p-value: 0.5483
```

```
rh <- ggplot(data = dfcel, aes(x = RHco2fluxweighted, y = d18O)) +
    geom_point(aes(colour = T_treatment, shape = Area), size = 3.5,
        ) + scale_color_manual(name = "Legend:", labels = c("Transpired Ambient Chamber",
    "Transpired Elevated Chamber"), values = c("black", "red")) +
    scale_shape_manual(name = "Legend:", values = c(15, 16, 17)) +
    geom_smooth(data = dfamb, method = "lm", formula = y ~ x,
        se = FALSE, colour = "black") + labs(y = "", x = "",
    element_text(size = 6)) + # scale_x_datetime(breaks = scales::pretty_breaks(n = 12))
theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(29, 36)) + scale_x_continuous(breaks = scales::pretty_breaks(n = 5),
    limits = c(55, 75)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 25)) + theme(plot.margin = unit(c(5, 5, 5, 5), "mm"))
rh
```



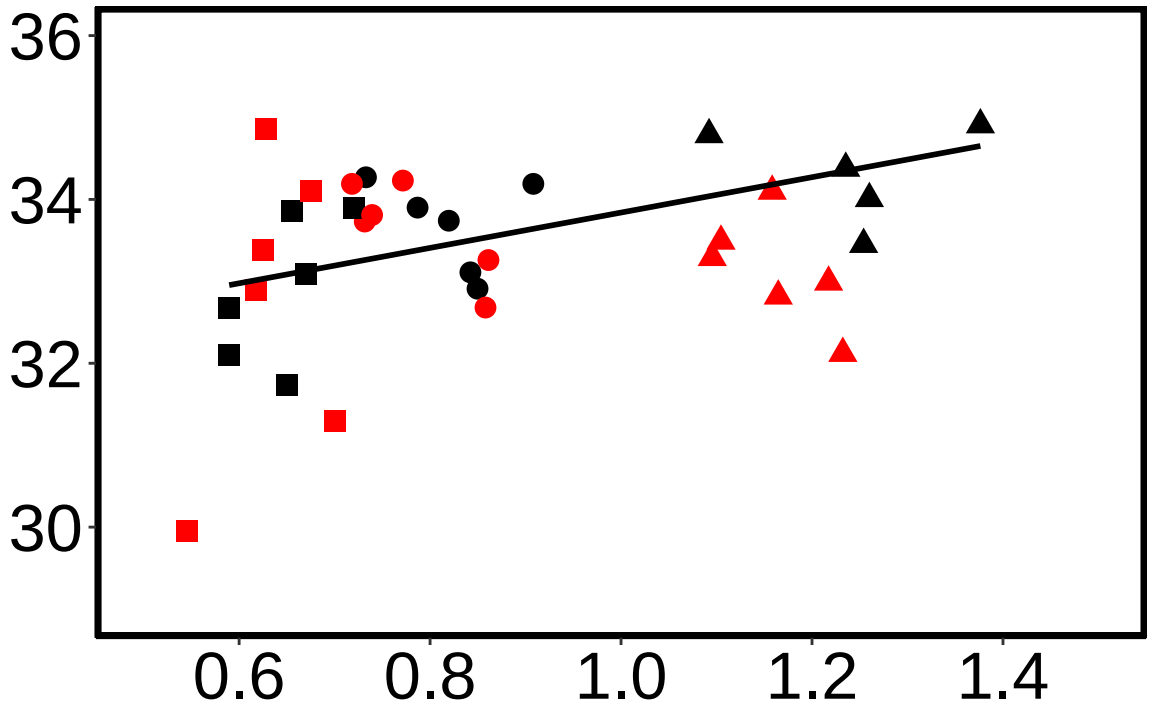

```
########################## VPD#####
vpdambstat <- lm(dfamb$d18O ~ dfamb$VPDco2fluxweighted)
summary(vpdambstat)  #amb d18 vs vpd wighted SIG
```

```
##
## Call:
## lm(formula = dfamb$d18O ~ dfamb$VPDco2fluxweighted)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.34501 -0.39117  0.03223  0.54852  1.00691
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 31.6812     0.6253  50.665  < 2e-16 ***
## dfamb$VPDco2fluxweighted    2.1586     0.6801   3.174  0.00629 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7068 on 15 degrees of freedom
## Multiple R-squared:  0.4018, Adjusted R-squared:  0.3619
## F-statistic: 10.08 on 1 and 15 DF,  p-value: 0.006288
```

```
vpdhotstat <- lm(dfhot$d18O ~ dfhot$VPDco2fluxweighted)
summary(vpdhotstat)  #elevated d18 vs vpd wighted SIG
```

```
##
## Call:
## lm(formula = dfhot$d18O ~ dfhot$VPDco2fluxweighted)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1612 -0.3823  0.1671  0.8090  1.7314
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 32.9960     1.0945  30.148 1.59e-15 ***
## dfhot$VPDco2fluxweighted    0.2112     1.2322   0.171    0.866
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.2 on 16 degrees of freedom
## Multiple R-squared:  0.001833,   Adjusted R-squared:  -0.06055
## F-statistic: 0.02939 on 1 and 16 DF,  p-value: 0.866
```

```
vpd <- ggplot(data = dfcel, aes(x = VPDco2fluxweighted, y = d18O)) +
    geom_point(aes(colour = T_treatment, shape = Area), size = 3.5,
        ) + scale_color_manual(name = "Legend:", labels = c("Transpired Ambient Chamber",
    "Transpired Elevated Chamber"), values = c("black", "red")) +
    scale_shape_manual(name = "Legend:", values = c(15, 16, 17)) +
    geom_smooth(data = dfamb, method = "lm", formula = y ~ x,
        se = FALSE, colour = "black") + labs(y = "", x = "",
    element_text(size = 6)) + theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(29, 36)) + scale_x_continuous(breaks = scales::pretty_breaks(n = 5),
    limits = c(0.5, 1.5)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 25)) + theme(plot.margin = unit(c(5, 5, 5, 5), "mm"))
vpd
```



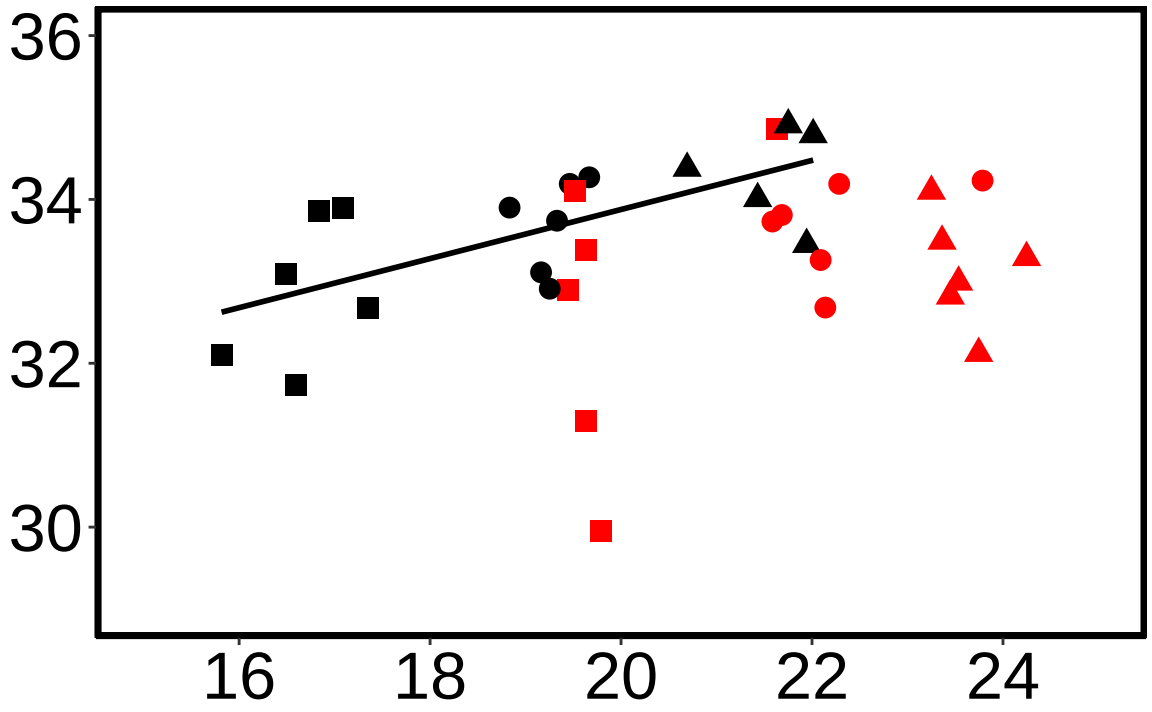

```
########################## VPD#####
tempambstat <- lm(dfamb$d18O ~ dfamb$weightedMeanLeafT)
summary(tempambstat)  #amb d18 vs vpd wighted SIG
```

```
##
## Call:
## lm(formula = dfamb$d18O ~ dfamb$weightedMeanLeafT)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1180 -0.5161  0.2639  0.4739  0.9316
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               27.8844     1.5145  18.411 1.04e-11 ***
## dfamb$weightedMeanLeafT   0.2996     0.0791   3.788  0.00179 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6533 on 15 degrees of freedom
## Multiple R-squared:  0.4889, Adjusted R-squared:  0.4548
## F-statistic: 14.35 on 1 and 15 DF,  p-value: 0.001786
```

```
temphotstat <- lm(dfhot$d18O ~ dfhot$weightedMeanLeafT)
summary(temphotstat)  #elevated d18 vs vpd wighted SIG
```

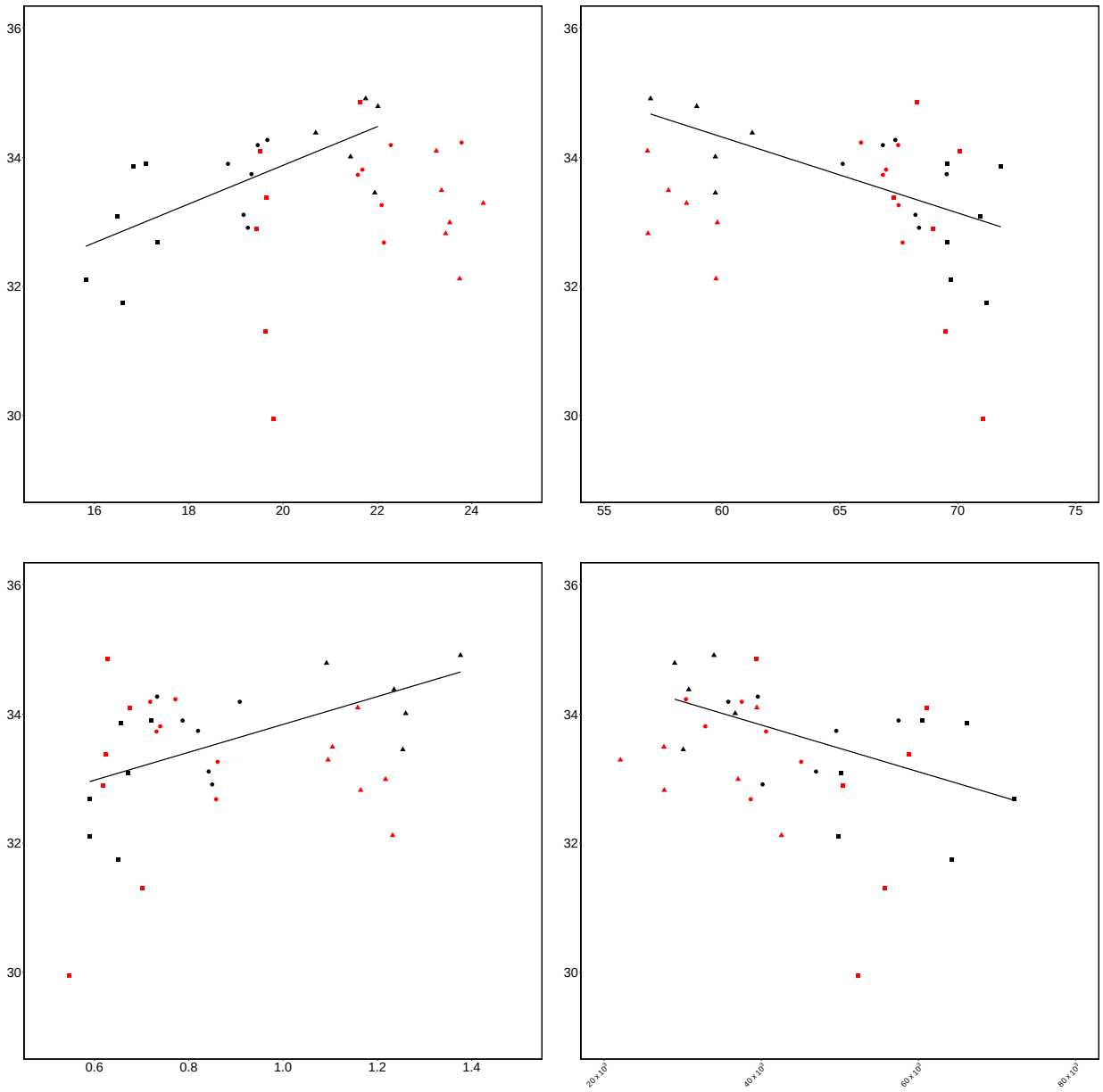
```
##
## Call:
## lm(formula = dfhot$d18O ~ dfhot$weightedMeanLeafT)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7935 -0.5318  0.1340  0.6755  1.7428
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               28.7398     3.6325   7.912  6.4e-07 ***
## dfhot$weightedMeanLeafT   0.2023     0.1651   1.225    0.238
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.148 on 16 degrees of freedom
## Multiple R-squared:  0.08574,    Adjusted R-squared:  0.0286
## F-statistic: 1.501 on 1 and 16 DF,  p-value: 0.2383
```

```
temp <- ggplot(data = dfcel, aes(x = weightedMeanLeafT, y = d18O)) +
    geom_point(aes(colour = T_treatment, shape = Area), size = 3.5,
        ) + scale_color_manual(name = "Legend:", labels = c("Transpired Ambient Chamber",
    "Transpired Elevated Chamber"), values = c("black", "red")) +
    scale_shape_manual(name = "Legend:", values = c(15, 16, 17)) +
    geom_smooth(data = dfamb, method = "lm", formula = y ~ x,
        se = FALSE, colour = "black") + labs(y = "", x = "",
    element_text(size = 6)) + theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + ggtitle("") + theme(legend.position = "none") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 5),
        limits = c(29, 36)) + scale_x_continuous(breaks = scales::pretty_breaks(n = 5),
    limits = c(15, 25)) + theme(axis.text.y = element_text(color = "black",
    size = 25)) + theme(axis.text.x = element_text(color = "black",
    size = 25)) + theme(plot.margin = unit(c(5, 5, 5, 5), "mm"))
temp
```




```
figure5 <- plot_grid(temp, rh, vpd, pex, align = "vh", ncol = 2)
# ggsave('cellulosegraphwithPEX.tiff', width = 30, height =
# 30, units = 'cm', dpi=300)
```

```
plot(figure5)
```



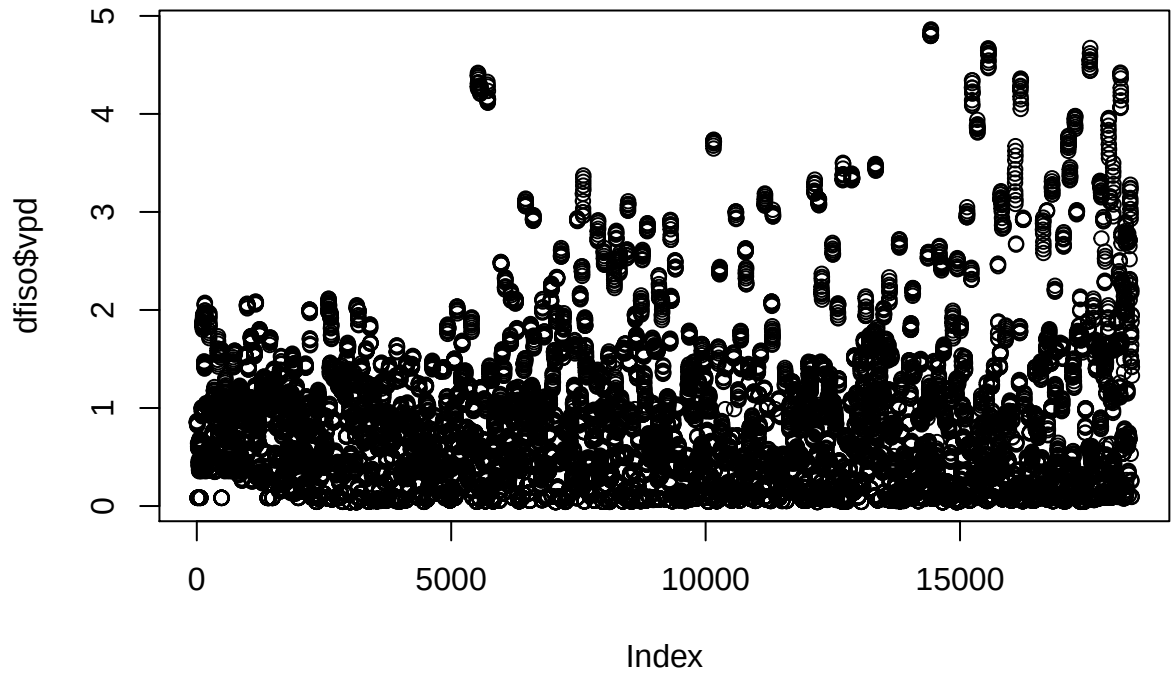
#the remaning plot is SI3 (the multiple regression is in a .xls) #to do this we needed to match a few data frames: #wind + rain + d18oatm&FLUX

```
library(data.table)
# Flux and Isotope Data
googledriveWTC4ISOTOPEID <- "1-lD4W_gdrlGssKnSpJw-3yIExXoCtdYw"
dfiso <- read.csv(sprintf("https://docs.google.com/uc?id=%s&export=download",
    googledriveWTC4ISOTOPEID))
dfiso["Td"] <- (dmy_hm(dfiso$Td, quiet = TRUE, tz = "UTC"))
summary(dfiso$Td)
```

```
##                  Min.               1st Qu.                Median
## "2016-08-28 13:30:00" "2016-09-18 07:12:00" "2016-10-06 01:14:00"
##                  Mean               3rd Qu.                  Max.
```

```
## "2016-10-08 05:06:36" "2016-10-27 13:30:00" "2016-11-22 18:33:00"
```

```
#### AddVPD
dfiso$esat <- esat(TdegC = dfiso$Tair_al, Pa = 101)
dfiso$vpd <- RHtoVPD(RH = dfiso$RHref_al, TdegC = dfiso$Taref_al,
    Pa = 101)
plot(dfiso$vpd)
```




```
summary(dfiso$vpd)  #VPD for ATM beacuse We used REFvalues
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.03778 0.17684 0.51940 0.74210 1.06062 4.86214
```

```
# Wind data
googledriveWind <- "1gQfK5zCCIEb6m4nWRHUV3mxWLu7KlwA2"
dfwind <- read.csv(sprintf("https://docs.google.com/uc?id=%s&export=download",
    googledriveWind))
dfwind["Td"] <- (ymd_hms(dfwind$DateTime, quiet = TRUE, tz = "UTC"))
summary(dfwind$Td)
```

```
##                  Min.               1st Qu.                Median
## "2016-08-01 00:00:00" "2016-09-07 23:58:45" "2016-10-15 23:57:30"
##                  Mean               3rd Qu.                  Max.
## "2016-10-15 23:57:30" "2016-11-22 23:56:15" "2016-12-30 23:55:00"
```

```
# Rain Data
googledriveRain <- "1YEzDoVrDS-_jqv0jTl_CwlLhdTIs_caJ"
dfrain <- read.csv(sprintf("https://docs.google.com/uc?id=%s&export=download",
    googledriveRain))
dfrain$Td <- (ymd_hms(dfrain$date, quiet = TRUE, tz = "UTC"))
summary(dfrain$Td)
```

```
##                  Min.               1st Qu.                Median
## "2016-01-01 00:00:00" "2016-04-01 06:00:00" "2016-07-01 12:00:00"
##                  Mean               3rd Qu.                  Max.
## "2016-07-01 12:00:00" "2016-09-30 18:00:00" "2016-12-31 00:00:00"
```

```
start <- ymd_hms("2016-08-28 12:00:00 UTC")
summary(start)
```

```
##                  Min.               1st Qu.                Median
## "2016-08-28 12:00:00" "2016-08-28 12:00:00" "2016-08-28 12:00:00"
##                  Mean               3rd Qu.                  Max.
## "2016-08-28 12:00:00" "2016-08-28 12:00:00" "2016-08-28 12:00:00"
```

```
end <- ymd_hms("2016-11-26 12:00:00")
int <- interval(start, end)
summary(int)
```

```
##             Intervals     Earliest endpoint       Latest endpoint
##                   "1" "2016-08-28 12:00:00" "2016-11-26 12:00:00"
##             Time zone
##                 "UTC"
```

```
dfwind <- subset(dfwind, dfwind$Td %within% int)
dfiso <- subset(dfiso, dfiso$Td %within% int)
dfiso <- subset(dfiso, T_treatment == "ambient")  ##
dfiso <- subset(dfiso, chamber == "C03")  # Since we are using ref values and atm we can just take an a
dfrain <- subset(dfrain, dfrain$Td %within% int)
max(dfrain$rain)
```

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

```
iso <- data.table(dfiso, key = "Td")
wind <- data.table(dfwind, key = "Td")
isowind <- wind[iso, roll = "nearest", allow.cartesian = TRUE]
isowind <- na.omit(isowind)
str(isowind)
```

```
## Classes 'data.table' and 'data.frame':   1586 obs. of  83 variables:
##  $ X                  : int  43062 43074 43086 43098 43110 43122 43134 43146 43158 43170 ...
##  $ DateTime           : chr  "2016-08-28 13:30:00" "2016-08-28 14:30:00" "2016-08-28 15:30:00" "201
##  $ RECORD             : int  112377 112389 112401 112413 112425 112437 112449 112461 112473 112485
##  $ PPFD_Avg           : num  1095 972 473 367 25 ...
##  $ AirTC_Avg          : num  19.7 20.6 19.2 17.8 16.3 ...
```

```
##  $ RH                : num  26.2 26.7 29.2 33.2 41 ...
##  $ WS_ms_Avg         : num  1.19 1.25 2.78 2.45 0 ...
##  $ WS_ms_Max         : num  1.64 1.88 3.8 2.89 0 ...
##  $ WindDir           : num  63.9 194.1 162.1 153.6 179.6 ...
##  $ NetSW_Avg         : num  334.4 292.4 124.2 96.1 0.129 ...
##  $ NetLW_Avg         : num  -115.9 -110 -89.7 -88.6 -61 ...
##  $ NetRad_Avg        : num  218.4 182.4 34.49 7.46 -60.91 ...
##  $ LWmV_Avg          : num  264 264 264 264 265 ...
##  $ LWMDry_Tot        : num  5 5 5 5 5 5 5 0 0 ...
##  $ LWMCon_Tot        : num  0 0 0 0 0 0 0 0 5 0 ...
##  $ LWMWet_Tot        : num  0 0 0 0 0 0 0 0 0 5 ...
##  $ Date              : chr  "2016-08-28" "2016-08-28" "2016-08-28" "2016-08-28" ...
##  $ Source            : chr  "ROS_WS_Table05min_20160831.dat" "ROS_WS_Table05min_20160831.dat" "ROS
##  $ Td                : POSIXct, format: "2016-08-28 13:30:00" "2016-08-28 14:31:00" ...
##  $ chamber           : chr  "C03" "C03" "C03" "C03" ...
##  $ CO2L              : num  410 411 404 402 408 ...
##  $ DIFFP             : num  117 118 117 117 117 ...
##  $ CO2FLOW           : num  260 260 262 263 252 ...
##  $ CO2CChamb         : num  405 416 416 417 410 ...
##  $ DPCChamb          : num  6.4 6.8 7.1 6.4 5.4 6 5.7 5.7 4.7 2.2 ...
##  $ TAIR              : num  19.9 19.9 17.3 16.5 14.5 11.7 10.8 7.4 7 7.3 ...
##  $ AddTime           : num  401 424 349 155 16 ...
##  $ LCO2slope         : num  0.526 0.537 0.531 0.537 0.529 ...
##  $ CONDH2O           : num  0.01463 0.0163 0.01074 0.00481 0.0037 ...
##  $ TBox              : num  30.9 29 26.4 22.1 19.8 17.3 15.1 13.8 12.3 11 ...
##  $ DoorCnt           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ CO2C              : num  399 399 398 397 398 ...
##  $ ROTR_RH           : num  32.7 27.9 32.8 38.7 48.9 ...
##  $ AIRPRESS          : num  102 102 102 102 102 ...
##  $ DewPntC           : num  3.1 3.2 3.2 4 4.2 4.8 5.2 5.1 5.2 5.1 ...
##  $ ROTR_TA           : num  21.2 21.3 19.6 18.4 14.2 ...
##  $ PAR               : num  935.8 552.2 316 240.8 8.2 ...
##  $ WIND              : num  0.34 0.32 1.28 0.9 0 0 0 0 0 0 ...
##  $ REFCO2            : num  399 399 397 396 398 ...
##  $ RAIN              : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Taref_al          : num  19.6 19.6 17.7 16.5 14.1 ...
##  $ RHref_al          : num  43.8 40.5 48.1 52.2 58.6 ...
##  $ Tchilled_SP       : num  2.26 1.68 2.48 2.51 1.86 ...
##  $ Tchilled          : num  2.1 1.85 2.05 2.25 1.9 ...
##  $ DPref_al          : num  6.82 5.45 6.45 6.52 5.86 ...
##  $ Tair_SP           : num  19.6 19.7 17.8 16.5 14.1 ...
##  $ RH_al             : num  50.4 50.3 56.4 55.6 59.5 ...
##  $ DP_al             : num  8.67 8.78 8.59 7.81 6.53 ...
##  $ Tsub_al           : num  20.2 19.4 17.8 16.6 14.1 ...
##  $ RH_SP             : num  43.6 40.4 48 52.1 58.6 ...
##  $ Tair_al           : num  19.2 19.1 17.2 16.7 14.4 ...
##  $ kfactor           : num  0.85 0.85 0.85 0.85 0.85 0.85 0.85 0.85 0.85 0.85 ...
##  $ ChVolume          : int  52800 52800 52800 52800 52800 52800 52800 52800 52800 52800 ...
##  $ HWTC              : num  1118 1112 1105 1060 975 ...
##  $ prevDateTime      : chr  "28/08/2016 13:19" "28/08/2016 14:19" "28/08/2016 15:19" "28/08/2016 1
##  $ prevHWTC          : num  1109 1139 1198 1041 999 ...
##  $ prevCWTC          : num  403 414 398 406 410 ...
##  $ tcycle            : int  900 900 900 900 900 900 900 900 900 900 ...
##  $ fCO2              : num  4.34 4.33 4.36 4.38 4.2 ...
```

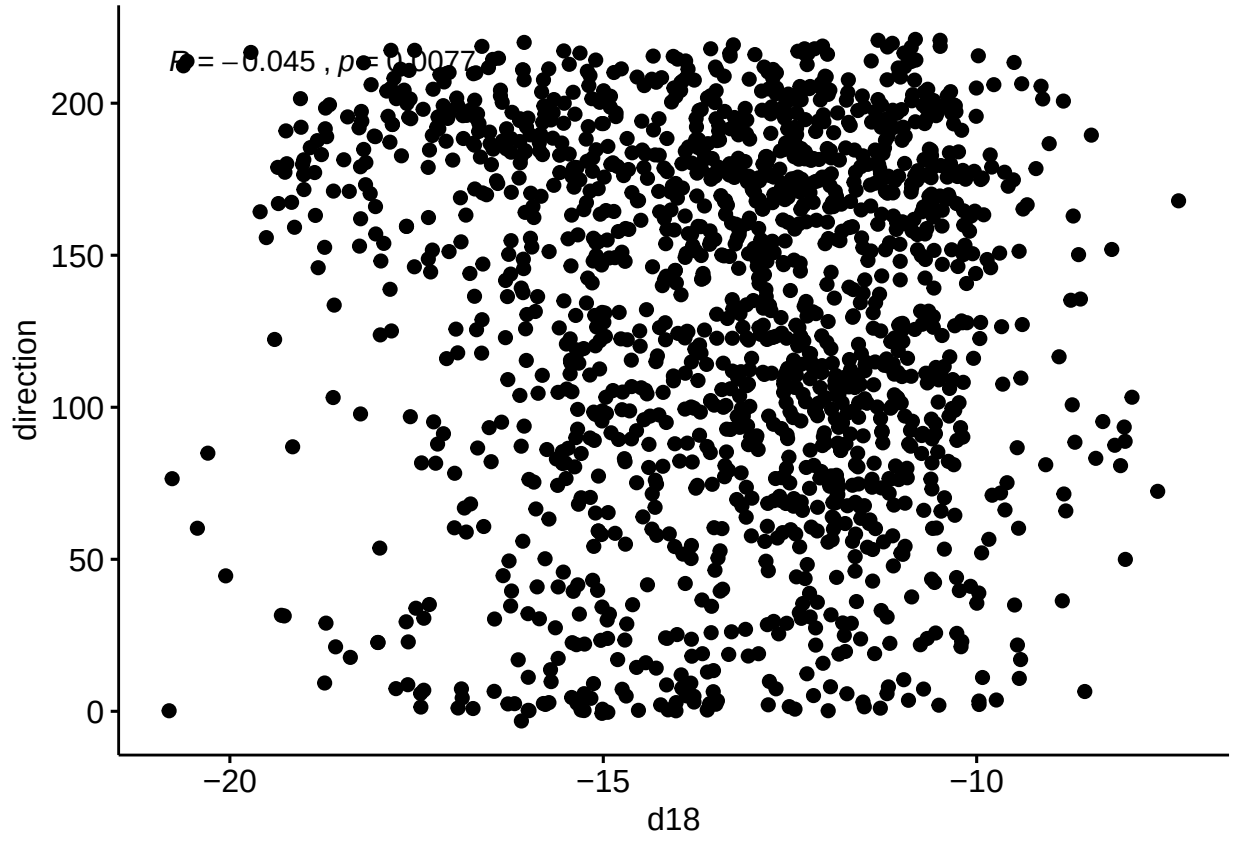
```
##  $ Href                : num  996 922 976 979 940 ...
##  $ ICO2                : num  0.08634 0.09114 0.07558 0.03365 0.00333 ...
##  $ Dref                : num  1.21 1.21 1.22 1.23 1.24 ...
##  $ Ain                 : num  8.61 8.63 8.64 8.65 8.68 ...
##  $ Aout                : num  8.62 8.65 8.65 8.66 8.68 ...
##  $ v                   : num  0.156 0.16 0.16 0.161 0.159 ...
##  $ F                   : num  0.153 0.154 0.153 0.153 0.154 ...
##  $ deltaS              : num  0.00602 0.00314 0.0461 0.02776 0 ...
##  $ Fwat                : num  0.00375 0.00348 0.00369 0.0037 0.00356 ...
##  $ Vwat                : num  0.00422 0.00421 0.00418 0.00401 0.0037 ...
##  $ deltaH2O            : num  0.000235 -0.000684 -0.002376 0.000473 -0.000624 ...
##  $ FluxH2O             : num  0.01533 0.01634 0.00886 0.00559 0.00321 ...
##  $ FluxCO2             : num  0.07785 0.08134 0.02203 -0.00243 -0.00126 ...
##  $ T_treatment         : chr  "ambient" "ambient" "ambient" "ambient" ...
##  $ MIU                 : chr  "   V:13 ambient" "   V:13 ambient" "   V:13 ambient" "   V:13 ambient
##  $ d18atm              : num  -14.5 -14.5 -14.2 -13.6 -13.2 ...
##  $ d18O.corrected.STD  : num  0.266 0.309 0.288 0.293 0.288 ...
##  $ Time                : chr  "28/08/2016 13:30" "28/08/2016 14:31" "28/08/2016 15:31" "28/08/2016 1
##  $ i.MIU               : chr  "  V:3 chamber 3" "  V:3 chamber 3" "  V:3 chamber 3" "  V:3 chamber 3
##  $ d18wtc              : num  -13.5 -13.5 -12.7 -14 -13.1 ...
##  $ i.d18O.corrected.STD: num  0.26 0.226 0.232 0.259 0.25 ...
##  $ i.Time              : chr  "28/08/2016 13:44" "28/08/2016 14:44" "28/08/2016 15:44" "28/08/2016 1
##  $ esat                : num  2231 2220 1968 1914 1647 ...
##  $ vpd                 : num  1.285 1.362 1.057 0.899 0.666 ...
##  - attr(*, "sorted")= chr "Td"
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
isowindspeed <- data.table(isowind, key = "Td")
raintable <- data.table(dfrain, key = "Td")
allvars <- raintable[isowindspeed, roll = "nearest", allow.cartesian = TRUE]
allvars$rain[is.na(allvars$rain)] <- 0
colnames(isowind)
```

```
##  [1] "X"                 "DateTime"          "RECORD"
##  [4] "PPFD_Avg"          "AirTC_Avg"         "RH"
##  [7] "WS_ms_Avg"         "WS_ms_Max"         "WindDir"
## [10] "NetSW_Avg"         "NetLW_Avg"         "NetRad_Avg"
## [13] "LWmV_Avg"          "LWMDry_Tot"        "LWMCon_Tot"
## [16] "LWMWet_Tot"        "Date"              "Source"
## [19] "Td"                "chamber"           "CO2L"
## [22] "DIFFP"             "CO2FLOW"           "CO2CChamb"
## [25] "DPCChamb"          "TAIR"              "AddTime"
## [28] "LCO2slope"         "CONDH2O"           "TBox"
## [31] "DoorCnt"           "CO2C"              "ROTR_RH"
## [34] "AIRPRESS"          "DewPntC"           "ROTR_TA"
## [37] "PAR"               "WIND"              "REFCO2"
## [40] "RAIN"              "Taref_al"          "RHref_al"
## [43] "Tchilled_SP"       "Tchilled"          "DPref_al"
## [46] "Tair_SP"           "RH_al"             "DP_al"
## [49] "Tsub_al"           "RH_SP"             "Tair_al"
## [52] "kfactor"           "ChVolume"          "HWTC"
## [55] "prevDateTime"      "prevHWTC"          "prevCWTC"
## [58] "tcycle"            "fCO2"              "Href"
## [61] "ICO2"              "Dref"              "Ain"
```

```
## [64] "Aout"                 "v"                     "F"
## [67] "deltaS"               "Fwat"                  "Vwat"
## [70] "deltaH2O"             "FluxH2O"               "FluxCO2"
## [73] "T_treatment"          "MIU"                   "d18atm"
## [76] "d18O.corrected.STD"   "Time"                  "i.MIU"
## [79] "d18wtc"               "i.d18O.corrected.STD" "i.Time"
## [82] "esat"                 "vpd"
```

```
# We want to test if VPD and Temp are predictors of d18O d18
# vs dir
ggscatter(isowind, x = "d18atm", y = "WindDir", conf.int = TRUE,
    cor.coef = TRUE, cor.method = "kendall", xlab = "d18", ylab = "direction")
```



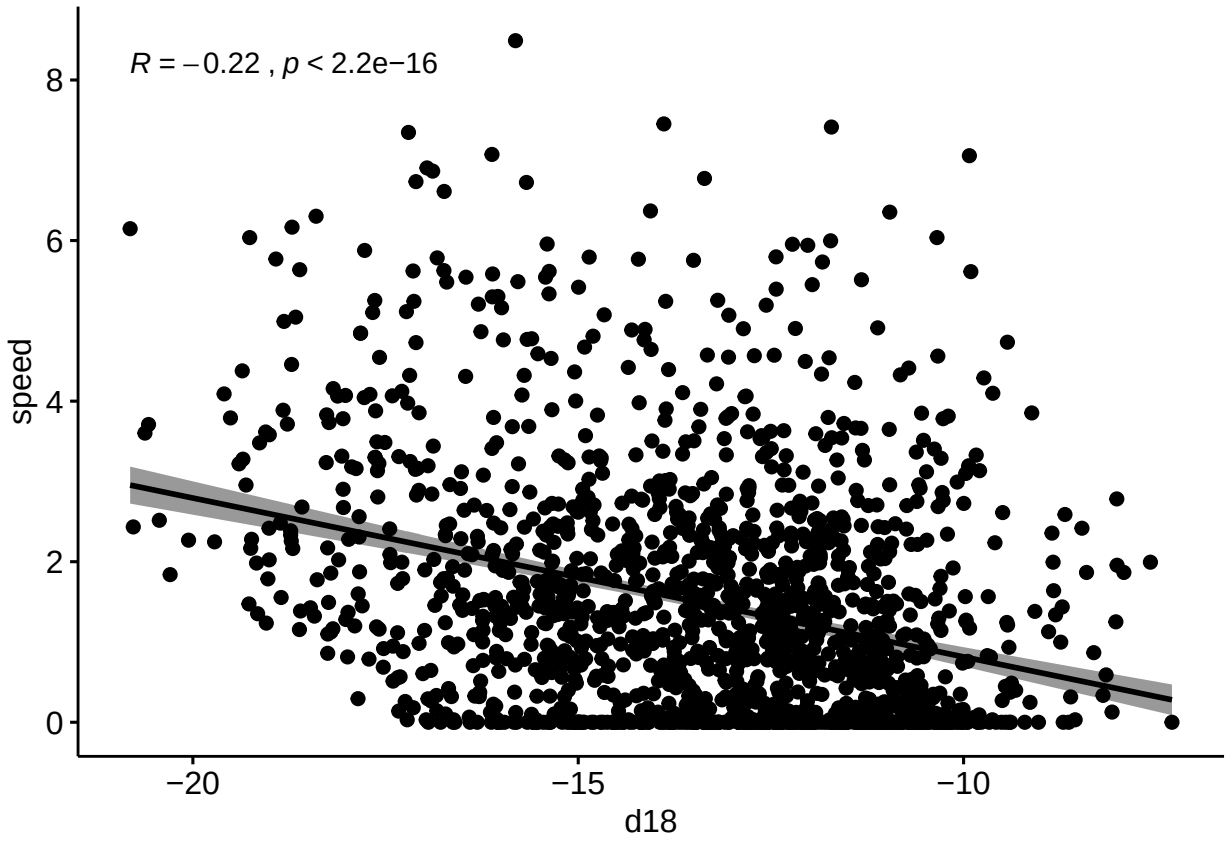

```
dir <- cor.test(isowind$WindDir, isowind$d18atm, method = "pearson")
dir
```

```
##
##  Pearson's product-moment correlation
##
## data:  isowind$WindDir and isowind$d18atm
## t = -2.1981, df = 1584, p-value = 0.02809
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.104084695 -0.005939958
## sample estimates:
```

```
##         cor
## -0.05514553
```

```
# d18 vs speed

ggscatter(isowind, x = "d18atm", y = "WS_ms_Avg", add = "reg.line",
    conf.int = TRUE, cor.coef = TRUE, cor.method = "kendall",
    xlab = "d18", ylab = "speed")
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
speed <- cor.test(isowind$WS_ms_Avg, isowind$d18atm, method = "pearson")
speed
```

```
##
##  Pearson's product-moment correlation
##
## data:  isowind$WS_ms_Avg and isowind$d18atm
## t = -13.21, df = 1584, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.3586825 -0.2699872
## sample estimates:
##        cor
## -0.3150225
```

```r
colnames(isowindspeed)
```

```
##  [1] "X"                    "DateTime"             "RECORD"
##  [4] "PPFD_Avg"             "AirTC_Avg"            "RH"
##  [7] "WS_ms_Avg"            "WS_ms_Max"            "WindDir"
## [10] "NetSW_Avg"            "NetLW_Avg"            "NetRad_Avg"
## [13] "LWmV_Avg"             "LWMDry_Tot"           "LWMCon_Tot"
## [16] "LWMWet_Tot"           "Date"                 "Source"
## [19] "Td"                   "chamber"              "CO2L"
## [22] "DIFFP"                "CO2FLOW"              "CO2CChamb"
## [25] "DPCChamb"             "TAIR"                 "AddTime"
## [28] "LCO2slope"            "CONDH2O"              "TBox"
## [31] "DoorCnt"              "CO2C"                 "ROTR_RH"
## [34] "AIRPRESS"             "DewPntC"              "ROTR_TA"
## [37] "PAR"                  "WIND"                 "REFCO2"
## [40] "RAIN"                 "Taref_al"             "RHref_al"
## [43] "Tchilled_SP"          "Tchilled"             "DPref_al"
## [46] "Tair_SP"              "RH_al"                "DP_al"
## [49] "Tsub_al"              "RH_SP"                "Tair_al"
## [52] "kfactor"              "ChVolume"             "HWTC"
## [55] "prevDateTime"         "prevHWTC"             "prevCWTC"
## [58] "tcycle"               "fCO2"                 "Href"
## [61] "ICO2"                 "Dref"                 "Ain"
## [64] "Aout"                 "v"                    "F"
## [67] "deltaS"               "Fwat"                 "Vwat"
## [70] "deltaH2O"             "FluxH2O"              "FluxCO2"
## [73] "T_treatment"          "MIU"                  "d18atm"
## [76] "d18O.corrected.STD"   "Time"                 "i.MIU"
## [79] "d18wtc"               "i.d18O.corrected.STD" "i.Time"
## [82] "esat"                 "vpd"
```

```r
multiple <- lm(d18atm ~ WS_ms_Avg + WindDir + Taref_al + vpd +
    RAIN, data = isowindspeed)
summary(multiple)
```

```
##
## Call:
## lm(formula = d18atm ~ WS_ms_Avg + WindDir + Taref_al + vpd +
##     RAIN, data = isowindspeed)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.2757 -1.4818  0.2244  1.5701  6.3267
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.461e+01  2.496e-01 -58.532   <2e-16 ***
## WS_ms_Avg   -8.616e-01  4.659e-02 -18.492   <2e-16 ***
## WindDir     -3.822e-04  9.069e-04  -0.421    0.673
## Taref_al     1.604e-01  1.721e-02   9.322   <2e-16 ***
## vpd         -2.522e-02  1.498e-01  -0.168    0.866
## RAIN         1.281e+00  1.972e+00   0.649    0.516
```

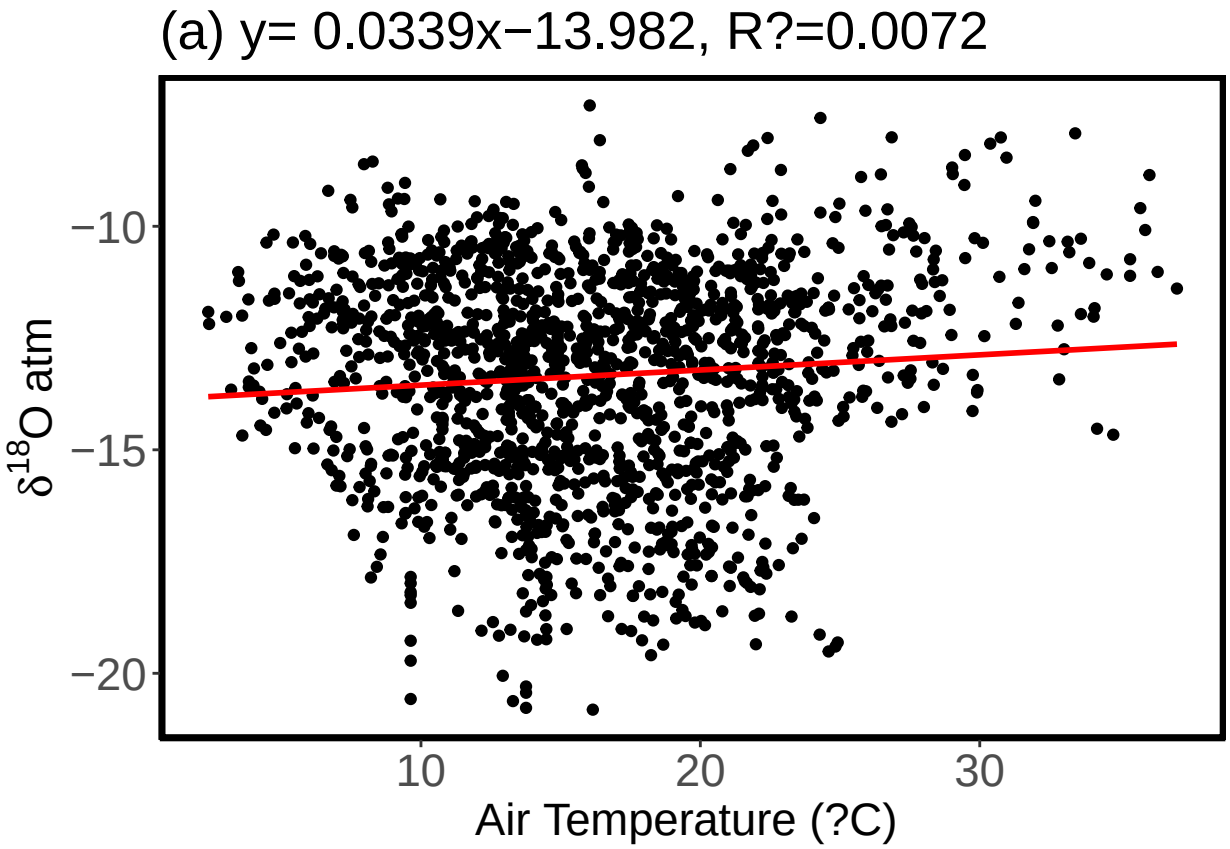
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.145 on 1580 degrees of freedom
## Multiple R-squared:  0.2057, Adjusted R-squared:  0.2032
## F-statistic: 81.85 on 5 and 1580 DF,  p-value: < 2.2e-16
```

```
si <- allvars

tlm <- lm(d18atm ~ Taref_al, data = si)
summary(tlm)
```

```
##
## Call:
## lm(formula = d18atm ~ Taref_al, data = si)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.4685 -1.6606  0.3725  1.7834  6.0523
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -13.892422   0.172699 -80.443  < 2e-16 ***
## Taref_al      0.033883   0.009984   3.394 0.000707 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.395 on 1584 degrees of freedom
## Multiple R-squared:  0.007219,   Adjusted R-squared:  0.006592
## F-statistic: 11.52 on 1 and 1584 DF,  p-value: 0.0007065
```

```
tair <- ggplot(data = si, aes(x = Taref_al, y = d18atm)) + geom_point(aes()) +
    geom_smooth(method = "lm", se = FALSE, colour = "red") +
    labs(y = (bquote(delta^18 * O ~ atm)), x = "Air Temperature (?C) ") +
    ggtitle("(a) y= 0.0339x-13.982, R?=0.0072") + # theme(axis.line=element_blank())+
theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + theme(text = element_text(size = 17)) +
    theme(axis.text = element_text(size = 17)) + theme(legend.position = "none")
tair
```

```
## `geom_smooth()` using formula 'y ~ x'
```

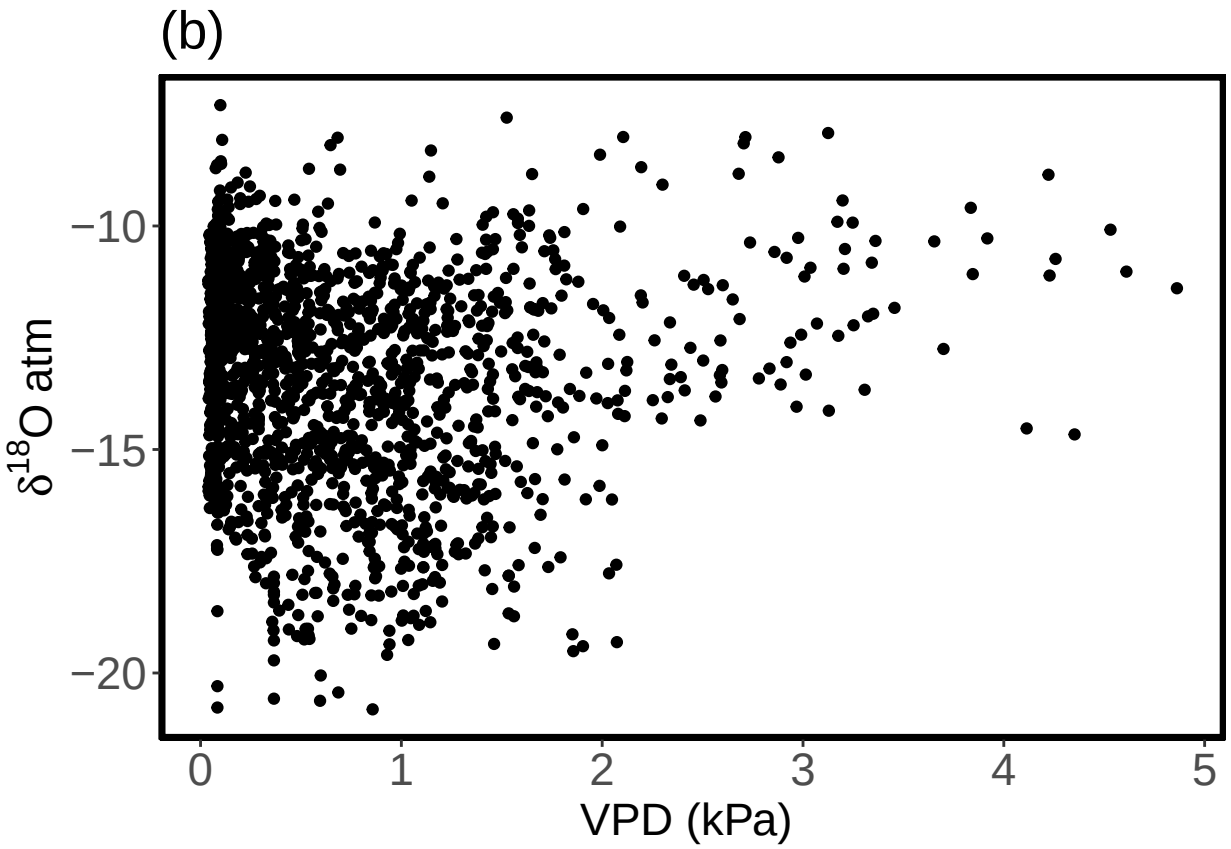



```
vlm <- lm(d18atm ~ vpd, data = si)
summary(vlm)
```

```
##
## Call:
## lm(formula = d18atm ~ vpd, data = si)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.4660 -1.7196  0.3749  1.8015  6.0244
##
## Coefficients:
##             Estimate Std. Error  t value Pr(>|t|)
## (Intercept) -13.31751    0.08508 -156.524   <2e-16 ***
## vpd          -0.03517    0.08273   -0.425    0.671
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.403 on 1584 degrees of freedom
## Multiple R-squared:  0.0001141,  Adjusted R-squared:  -0.0005172
## F-statistic: 0.1807 on 1 and 1584 DF,  p-value: 0.6708
```

```
vpd <- ggplot(data = si, aes(x = vpd, y = d18atm)) + geom_point(aes()) +
    # scale_colour_gradientn(colours = jet.colors(7))+
# geom_smooth(method = 'lm', se=FALSE, colour = 'red')+
```

```
labs(y = (bquote(delta^18 * O ~ atm)), x = "VPD (kPa) ") + # labs(fill=bquote(delta^18*O))+ labs(fill=(
ggtitle("(b)") + # theme(axis.line=element_blank())+
theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + theme(text = element_text(size = 17)) +
    theme(axis.text = element_text(size = 17)) + theme(legend.position = "none")
# facet_wrap(vars(Cw.chamber), nrow = 4)
vpd
```



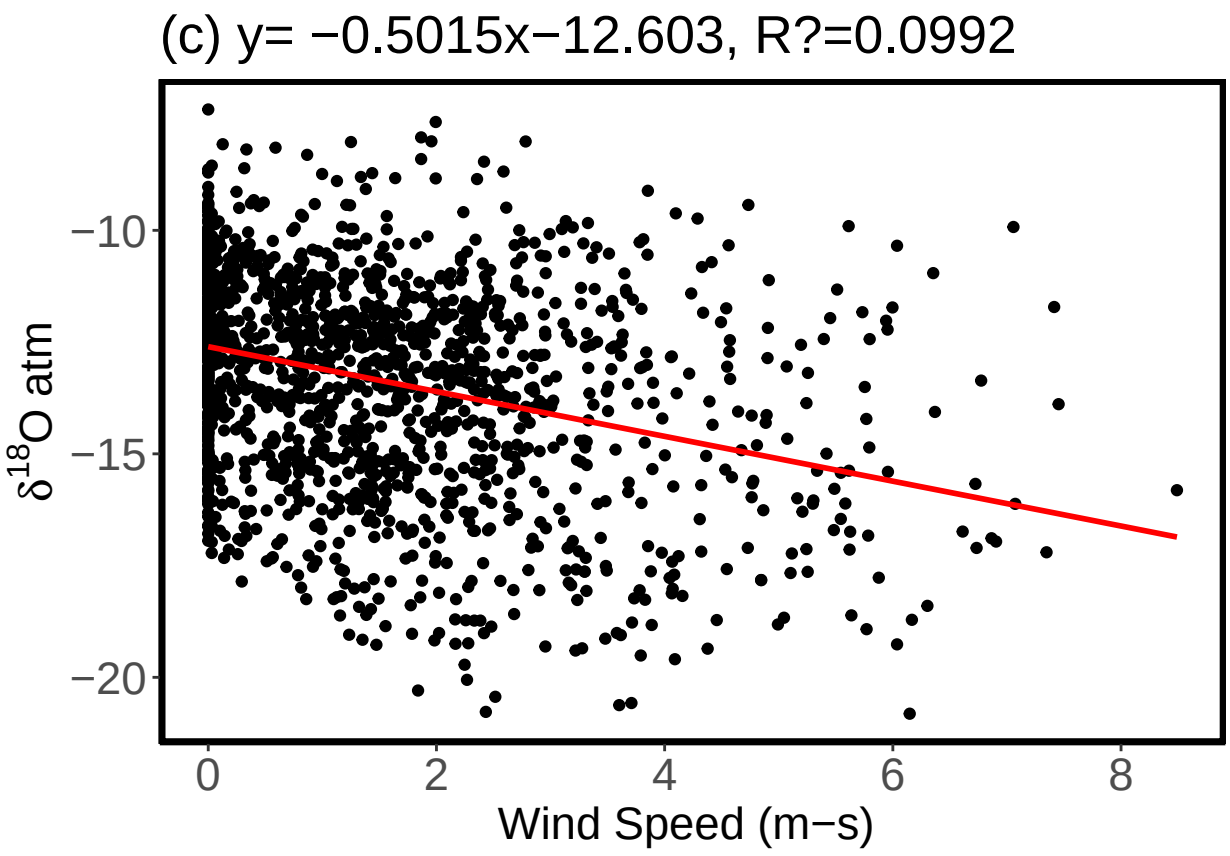

```
wslm <- lm(d18atm ~ WS_ms_Avg, data = si)
summary(wslm)
```

```
##
## Call:
## lm(formula = d18atm ~ WS_ms_Avg, data = si)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9500 -1.6149  0.2579  1.6776  6.2171
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -12.60268    0.08013 -157.27   <2e-16 ***
## WS_ms_Avg    -0.50155    0.03797  -13.21   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 2.281 on 1584 degrees of freedom
## Multiple R-squared:  0.09924,    Adjusted R-squared:  0.09867
## F-statistic: 174.5 on 1 and 1584 DF,  p-value: < 2.2e-16
```

```
windspeed <- ggplot(data = si, aes(x = WS_ms_Avg, y = d18atm)) +
    geom_point(aes()) + geom_smooth(method = "lm", se = FALSE,
    colour = "red") + labs(y = (bquote(delta^18 * O ~ atm)),
    x = "Wind Speed (m-s) ") + ggtitle("(c) y= -0.5015x-12.603, R?=0.0992") +
    # theme(axis.line=element_blank())+
theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + theme(text = element_text(size = 17)) +
    theme(axis.text = element_text(size = 17)) + theme(legend.position = "none")
windspeed
```

```
## `geom_smooth()` using formula 'y ~ x'
```



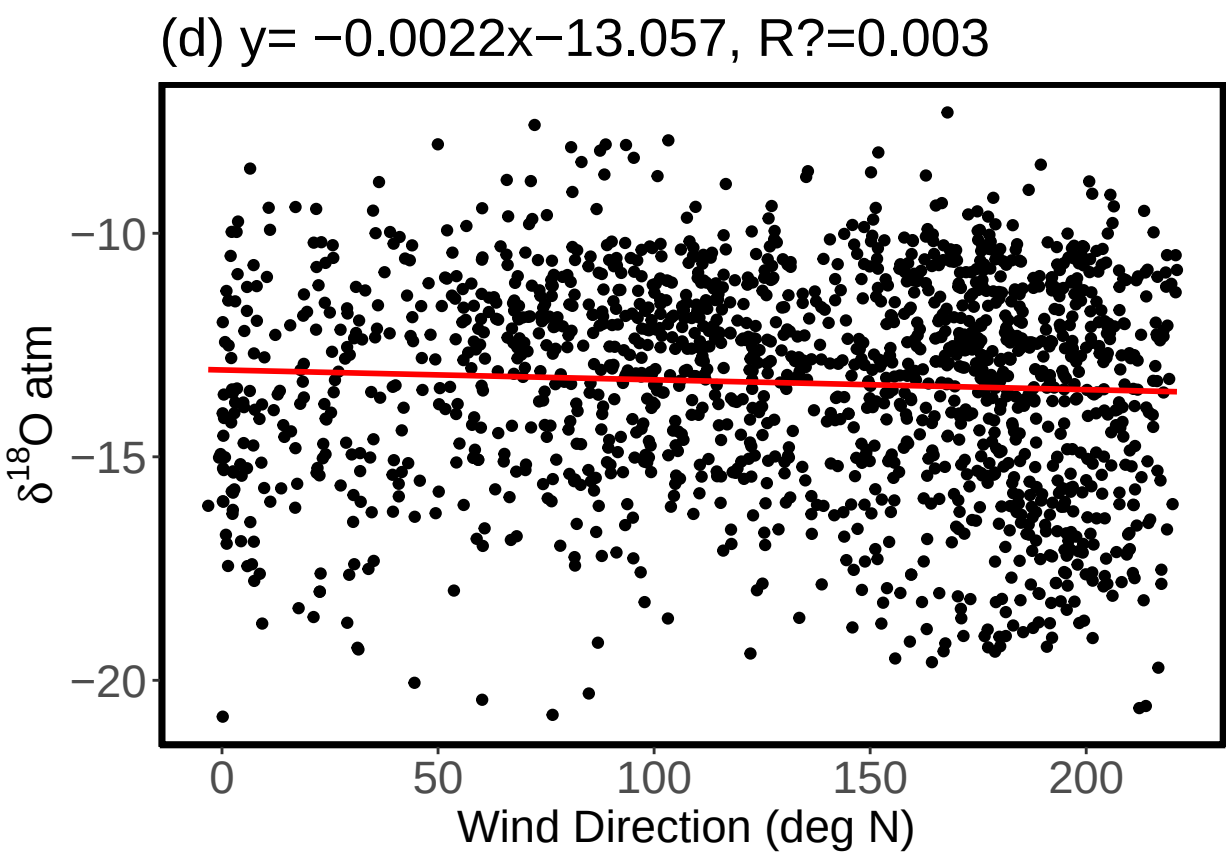

```
wdlm <- lm(d18atm ~ WindDir, data = si)
summary(wdlm)
```

```
##
## Call:
## lm(formula = d18atm ~ WindDir, data = si)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.7560 -1.7198  0.3906  1.7809  6.1306
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -13.057259   0.143285 -91.128   <2e-16 ***
## WindDir      -0.002203   0.001002  -2.198   0.0281 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.4 on 1584 degrees of freedom
## Multiple R-squared:  0.003041,   Adjusted R-squared:  0.002412
## F-statistic: 4.832 on 1 and 1584 DF,  p-value: 0.02809
```

```
winddirection <- ggplot(data = si, aes(x = WindDir, y = d18atm)) +
    geom_point(aes()) + geom_smooth(method = "lm", se = FALSE,
    colour = "red") + labs(y = (bquote(delta^18 * O ~ atm)),
    x = "Wind Direction (deg N) ") + ggtitle("(d) y= -0.0022x-13.057, R?=0.003") +
    # theme(axis.line=element_blank())+
theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + theme(text = element_text(size = 17)) +
    theme(axis.text = element_text(size = 17)) + theme(legend.position = "none")
winddirection
```

```
## `geom_smooth()` using formula 'y ~ x'
```

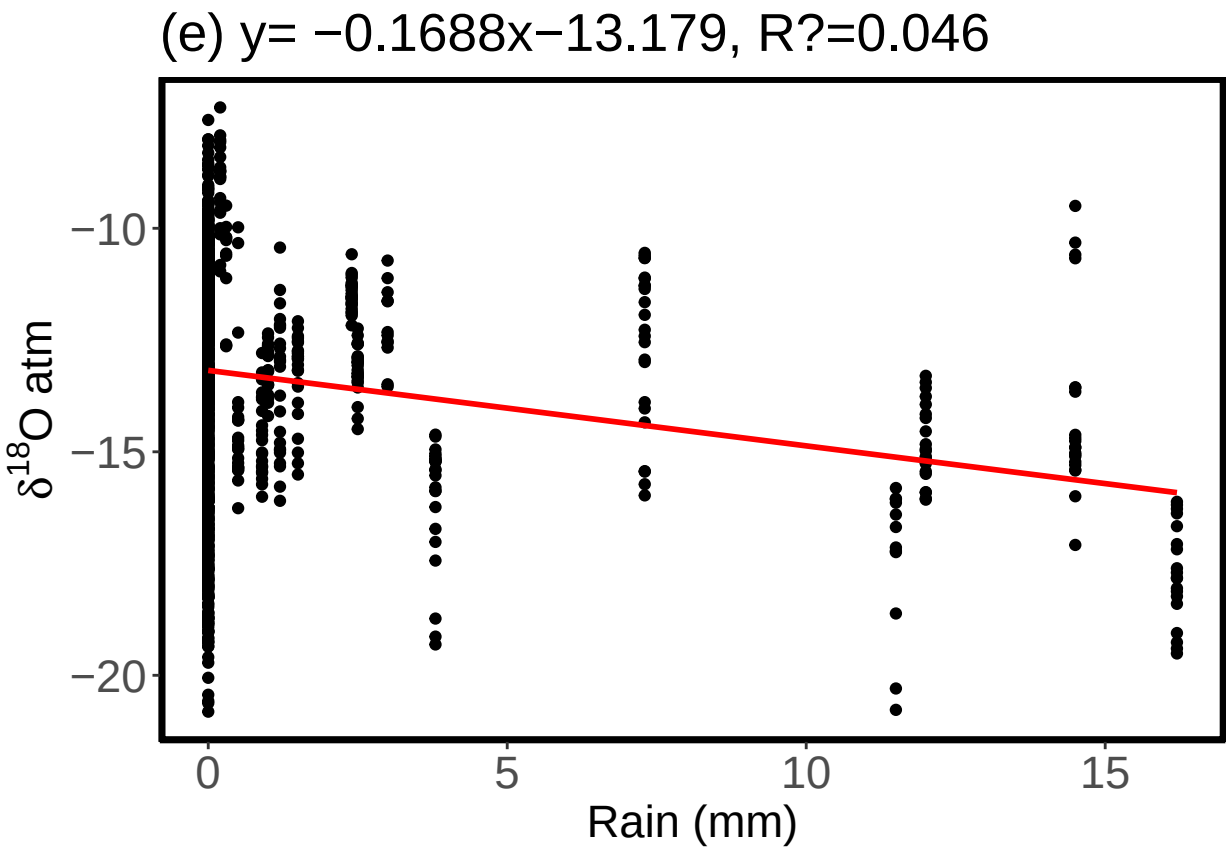


```r
rainlm <- lm(d18atm ~ rain, data = si)
summary(rainlm)
```

```
##
## Call:
## lm(formula = d18atm ~ rain, data = si)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.6346 -1.5637  0.3794  1.7586  6.1278
##
## Coefficients:
##              Estimate Std. Error  t value Pr(>|t|)
## (Intercept) -13.17910    0.06190 -212.895   <2e-16 ***
## rain         -0.16880    0.01942   -8.692   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.348 on 1584 degrees of freedom
## Multiple R-squared:  0.04553,    Adjusted R-squared:  0.04492
## F-statistic: 75.55 on 1 and 1584 DF,  p-value: < 2.2e-16
```

```r
rain <- ggplot(data = si, aes(x = rain, y = d18atm)) + geom_point(aes()) +
    geom_smooth(method = "lm", se = FALSE, colour = "red") +
    labs(y = (bquote(delta^18 * O ~ atm)), x = "Rain (mm)") +
    ggtitle("(e) y= -0.1688x-13.179, R?=0.046 ") + # theme(axis.line=element_blank())+
theme_classic() + theme(panel.border = element_rect(fill = "NA",
    colour = "black", size = 2)) + theme(text = element_text(size = 17)) +
    theme(axis.text = element_text(size = 17)) + theme(legend.position = "none")
rain
```

```
## `geom_smooth()` using formula 'y ~ x'
```



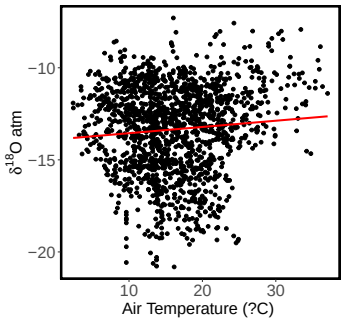

```
figsi3 = plot_grid(tair, vpd, windspeed, winddirection, rain,
    ncol = 1, nrow = 5, align = "vh")
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```

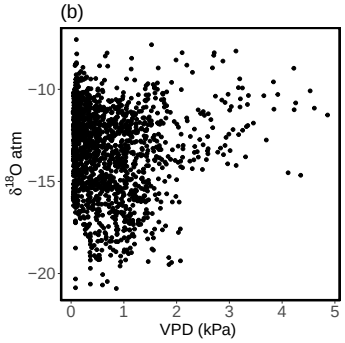
```
ggsave("SI3LQ.tiff", width = 17, height = 35, units = "cm", dpi = 90)
```

```
plot(figsi3)
```

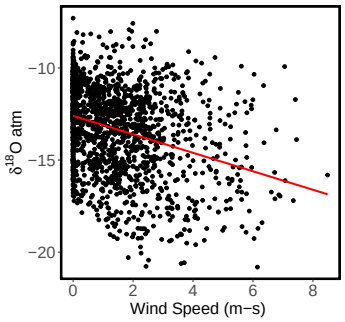
(a) y= 0.0339x−13.982, R?=0.007



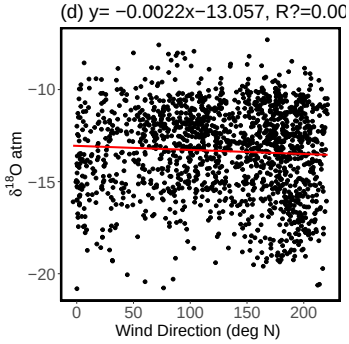
(b)



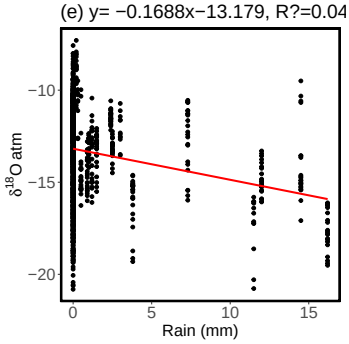
(c) y= −0.5015x−12.603, R?=0.09



(d) y= −0.0022x−13.057, R?=0.00



(e) y= −0.1688x−13.179, R?=0.04



#Done!