Code for IButton data

#El Gradient####

library(dplyr)

library(ggplot2)

library(data.table)

library(lubridate)

`%notin%` <- function(x,y) !(x %in% y)

tea<-read.csv( - read in your tea data here!)

#Read Data

filenames <- list.files(path="scripts/users/hthomas/Data/Tea/Env.Vars/Kluane\_Gradient/") # create list of temperature data files

tempdata <- lapply(filenames, function(x) read.csv(paste("scripts/users/hthomas/Data/Tea/Env.Vars/Kluane\_Gradient/",x,sep=""), sep=",", header=F, skip=15)) # read in all data files into large list

for(i in 1:length(filenames)) {

  #colnames(tempdata[[i]]) <- tempdata[[i]][-1,]; # put first line as header

  tempdata[[i]] <- tempdata[[i]][,-2]; # delete dispensable second column

  tempdata[[i]][,2] <- as.numeric(as.character(tempdata[[i]][,2]))

  names(tempdata[[i]])[c(1:2)]<-c("Date/Time","Value")

  tempdata[[i]][,1] <- as.POSIXct(tempdata[[i]][,1],format="%d/%m/%Y %H:%M") # convert date column into date format

  #days <- tempdata[[i]][,1]

  #tempdata[[i]] <- aggregate(tempdata[[i]][,2], by = list(days), mean); # compute mean temperatures per day

  fileinfo <- strsplit(file\_path\_sans\_ext(filenames[i]),"\_")[[1]]; # extract plot infos from file name

  for (j in 1:3) {

    tempdata[[i]][,2+j] <- fileinfo[j]

  } # add columns for plot infos

  colnames(tempdata[[i]])[c(1:5)] <- c("date","value","name","site","Plot")

}

#Raw Data

temp <- do.call("rbind", tempdata)

temp$date<-temp$date + years(2000) #correct years

temp$Day<-cut(temp$date, "day") #Extract days

temp$gradient\_site<-substr(temp$Plot, 1, nchar(temp$Plot)-1)

#Plot data

# ggplot(temp,aes(date,value,colour=factor(gradient\_site)))+

#   geom\_point(size=0.1)+

#   geom\_smooth()+

#   theme\_bw()

#Daily means

daily\_temp<- temp %>%

  group\_by(Day,Plot) %>%

  summarise(Daily\_mean = mean(value))

daily\_temp$Day<- as.Date(daily\_temp$Day)

# ggplot(daily\_temp,aes(Day,Daily\_mean,colour=factor(Plot)))+

#   geom\_point(size=0.4)+

#   geom\_smooth()+

#   theme\_bw()

#Plot means

#Trim data to dates of tea burial

plot\_temp\_means=NULL

for(i in unique(daily\_temp$Plot)){

  DOB<-min(tea[tea$Plot==i,]$Burial) #Find date of burial

  DOR<-max(tea[tea$Plot==i,]$Recovery) #Find date of recovery

  daily\_temp\_cut<-subset(daily\_temp,Day>=DOB&Day<=DOR) #Trim data

  Out<- daily\_temp\_cut %>%

    filter(Plot==i) %>%

    group\_by(Plot) %>%

    summarise(Mean\_Temperature = mean(Daily\_mean),

              sd\_Temperature = sd(Daily\_mean),

              GSL\_0 = length(which(Daily\_mean > 0)),

              GSL\_5 = length(which(Daily\_mean > 5)),

              GDD\_0 = sum(Daily\_mean[Daily\_mean>0]),

              GDD\_5 = sum(Daily\_mean[Daily\_mean>5]))

  plot\_temp\_means<-rbind(plot\_temp\_means,Out)

}

plot\_temp\_means$source<-"ibutton data"

plot\_temp\_means$scale<-"plot"

Plot\_Variables\_Soil<-rbind(Plot\_Variables\_Soil,plot\_temp\_means)

#El Gradient####

library(dplyr)

library(ggplot2)

library(data.table)

library(lubridate)

`%notin%` <- function(x,y) !(x %in% y)

tea<-read.csv( - read in your tea data here!)

#Read Data

filenames <- list.files(path="scripts/users/hthomas/Data/Tea/Env.Vars/Kluane\_Gradient/") # create list of temperature data files

tempdata <- lapply(filenames, function(x) read.csv(paste("scripts/users/hthomas/Data/Tea/Env.Vars/Kluane\_Gradient/",x,sep=""), sep=",", header=F, skip=15)) # read in all data files into large list

for(i in 1:length(filenames)) {

  #colnames(tempdata[[i]]) <- tempdata[[i]][-1,]; # put first line as header

  tempdata[[i]] <- tempdata[[i]][,-2]; # delete dispensable second column

  tempdata[[i]][,2] <- as.numeric(as.character(tempdata[[i]][,2]))

  names(tempdata[[i]])[c(1:2)]<-c("Date/Time","Value")

  tempdata[[i]][,1] <- as.POSIXct(tempdata[[i]][,1],format="%d/%m/%Y %H:%M") # convert date column into date format

  #days <- tempdata[[i]][,1]

  #tempdata[[i]] <- aggregate(tempdata[[i]][,2], by = list(days), mean); # compute mean temperatures per day

  fileinfo <- strsplit(file\_path\_sans\_ext(filenames[i]),"\_")[[1]]; # extract plot infos from file name

  for (j in 1:3) {

    tempdata[[i]][,2+j] <- fileinfo[j]

  } # add columns for plot infos

  colnames(tempdata[[i]])[c(1:5)] <- c("date","value","name","site","Plot")

}

#Raw Data

temp <- do.call("rbind", tempdata)

temp$date<-temp$date + years(2000) #correct years

temp$Day<-cut(temp$date, "day") #Extract days

temp$gradient\_site<-substr(temp$Plot, 1, nchar(temp$Plot)-1)

#Plot data

# ggplot(temp,aes(date,value,colour=factor(gradient\_site)))+

#   geom\_point(size=0.1)+

#   geom\_smooth()+

#   theme\_bw()

#Daily means

daily\_temp<- temp %>%

  group\_by(Day,Plot) %>%

  summarise(Daily\_mean = mean(value))

daily\_temp$Day<- as.Date(daily\_temp$Day)

# ggplot(daily\_temp,aes(Day,Daily\_mean,colour=factor(Plot)))+

#   geom\_point(size=0.4)+

#   geom\_smooth()+

#   theme\_bw()

#Plot means

#Trim data to dates of tea burial

plot\_temp\_means=NULL

for(i in unique(daily\_temp$Plot)){

  DOB<-min(tea[tea$Plot==i,]$Burial) #Find date of burial

  DOR<-max(tea[tea$Plot==i,]$Recovery) #Find date of recovery

  daily\_temp\_cut<-subset(daily\_temp,Day>=DOB&Day<=DOR) #Trim data

  Out<- daily\_temp\_cut %>%

    filter(Plot==i) %>%

    group\_by(Plot) %>%

    summarise(Mean\_Temperature = mean(Daily\_mean),

              sd\_Temperature = sd(Daily\_mean),

              GSL\_0 = length(which(Daily\_mean > 0)),

              GSL\_5 = length(which(Daily\_mean > 5)),

              GDD\_0 = sum(Daily\_mean[Daily\_mean>0]),

              GDD\_5 = sum(Daily\_mean[Daily\_mean>5]))

  plot\_temp\_means<-rbind(plot\_temp\_means,Out)

}

plot\_temp\_means$source<-"ibutton data"

plot\_temp\_means$scale<-"plot"

Plot\_Variables\_Soil<-rbind(Plot\_Variables\_Soil,plot\_temp\_means)