####Working code for both HOBO and TOMST data synthesis and analysis

####First: Convert HOBO csv's to the correct csv format####

####Combine all csv's and wrangle data to usable form####

####Calculate derived metrics####

####POC: Jordon Tourville

#library needed packages

library(tidyverse)

library(tidyr)

library(dplyr)

library(plyr)

library(ggplot2)

library(reshape)

library(purrr)

library(here)

library(lubridate)

library(pollen)

library(myClim)

#library(chillR)

#Ecomatik/HOBO data first

#setwd("~/hobo\_processing/missing\_converted") #set the working directory to the folder containing your raw csvs

#in the folder create a folder called "output" all of the csvs will be added to this.

#get the list of CSV files in your desired folder

file\_list <- list.files(pattern = "\*.csv",include.dirs = FALSE) #full.names = T)

# loop through each file and apply the code

for (file in file\_list) {

data <- read.csv(file, header = FALSE, skip = 1) #read in csv

file\_name <- file

name\_parts <- strsplit(file\_name, "\_")[[1]]

end\_parts <- strsplit(name\_parts[length(name\_parts)], "\\.")[[1]]

colnames(data) <- as.character(unlist(data[1,])) #column name based on the first row

data <- data[-1, -1] #remove the first row and first colun

data <- separate(data, "Date Time, GMT-04:00",

into = c("date", "time"),

sep = " ", extra = "merge") #separate the "Date Time, GMT-04:00" column into "date" and "time"

data$date <- gsub("/", "-", data$date) #convert to a format that works better in r

data$date <- as.Date(data$date, format = "%m-%d-%y") #convert the data column to data type

data$date <- gsub("-", "/", data$date) #reformat (again) to match the desired output

names(data)[1:4] <- c("Date", "Time", "Air Temp", "Soil Temp") #rename first four columns

data <- data[, 1:4] #only keep the first four columns

data$Flag <- ifelse(is.na(data$`Air Temp`) & is.na(data$`Soil Temp`), 2, #create a "flag" column based on mssing data

ifelse(is.na(data$`Air Temp`) | is.na(data$`Soil Temp`), 1, 0))

#data$Site <- file

write.csv(data, file = paste0("output/",name\_parts[1],"\_Complete\_",name\_parts[2],".csv"), row.names = FALSE) #write into a new csv file

}

#workflow - put new files in 1 folder

#run below code to merge all csv but make sure to specify path to both old and new file paths (run code twice)

#clean up two new resulting merged csv's. Rbind them into 1.

###merge csv's - repeat for both new and old files if starting fresh

#run only for new files if just adding new data

#set wd to new complete file

#change path when needed

file\_list <- list.files(path = "D:/output",#old

#"D:/HOBOware/Fall\_2024/output", #new

#D:/output\_1

#D:/output

#C:/Users/17342/Desktop/PhD Dissertation/AMC/Climate\_data/HOBO-master/Required\_Files/HOBO\_Complete\_Files

#D:/Climate\_data/HOBO-master/Required\_Files/HOBO\_Complete\_Files

recursive = TRUE,

pattern = "\*.csv",

full.names = TRUE,

include.dirs = FALSE)#read in file list

out1 = file\_list %>%

set\_names() %>%

map\_dfr(read\_csv, .id = "site") %>%

mutate(site = word(basename(site), sep = "\_"))#merge csv and make site name

out1 = out1 %>% mutate(Air = coalesce(`Air Temp`, `Air.Temp`))

out1 = out1 %>% mutate(Soil = coalesce(`Soil Temp`, `Soil.Temp`))

out1 = out1 %>% mutate(Flag = coalesce(`Flag (0/1/2)`, `Flag` , `Flag (0,1,2)`, `Flag(0/1/2)`, `Fail (0/1/2)`))

out1 = out1 %>% mutate(Time = coalesce(Time, `Time, GMT-04:00`))#collapse junk columns

#head(out1)

colnames(out1)[4] ="Air\_temp"

colnames(out1)[5] ="Soil\_temp"

###data wrangle

#remove columns, remove outliers, reshape, add time stamps

out1 = out1 %>%

select(site, Date, Time, Flag, Air\_temp, Soil\_temp)

out1 = subset(out1, Air\_temp < 40 & Soil\_temp < 29)

out1 = subset(out1, Air\_temp > -49 & Soil\_temp > -28)

out2 = out1

out3 = out1

out4 = out1

out5 = out1

#str(out2)

#out3$Date <- mdy(out3$Date)

out3$year = year(out3$Date)

out3$month = month(out3$Date)

out3$DOY = yday(out3$Date)

#out5 (2007-2021), out4 (2024), out3 (2022), out2 (2023)

hobo\_2024 = rbind(out5, out4, out3, out2, zeal)#put new data in with old

write\_csv(hobo\_2024, "hobo\_2024.csv")

hobo\_2024\_1 = ddply(hobo\_2024, ~site + Date, summarise, tmax = max(Air\_temp), tmin = min(Air\_temp))

hobo\_2024\_2 = merge(hobo\_2024, hobo\_2024\_1, by = c("site", "Date"), all.x = TRUE)

dat = ddply(hobo\_2024\_2, ~site + year + DOY, summarise, gdd\_0\_air = max(gdd(tmax = tmax, tmin = tmin, tbase\_max = 30, tbase = 0,

type = "C")), mean\_air = mean(Air\_temp, na.rm = TRUE), mean\_soil = mean(Soil\_temp, na.rm = TRUE),

month = max(month))

write.csv(dat, "hobo\_dat.csv")

###Reading TOMST Data

#install.packages("myClim")

library(myClim)

###Dendrometer Working below

#Data wrangling - TOMST

#load data frames with metadata (coordinates) for DENDRO

path <- read.csv("local\_dend.csv", header = TRUE)

loc <- read.csv("dend\_tomst\_meta.csv", header = TRUE)

# upload metadata from data.frame

tms.d <- mc\_read\_data(files\_table = path,

localities\_table = loc,

silent = T)

#Cleaning - following is similar workflow for both TMS4 and dendrometers

tmsd <- mc\_prep\_clean(tms.d, silent = T)

tmsd <- mc\_prep\_solar\_tz(tmsd)#need metadata

mc\_info(tmsd)

mc\_info\_count(tmsd)

mc\_info\_meta(tmsd)

#prepare time series plots for all tomst sensors

p <- mc\_plot\_line(tmsd)

p <- p+ggplot2::scale\_x\_datetime(date\_breaks = "1 week", date\_labels = "%W")

p <- p+ggplot2::xlab("week")

p <- p+ggplot2::aes(size = sensor\_name)

p <- p+ggplot2::scale\_size\_manual(values = c(1, 1 ,1, 1))

p <- p+ggplot2::guides(size = "none")

p <- p+ggplot2::scale\_color\_manual(values = c("hotpink", "pink", "darkblue", "orange"), name = NULL)

p

tiff(file = "tms\_p.tiff", width = 8, height = 11, units = 'in', res = 600, pointsize = 11)

p

dev.off()

p1 = mc\_plot\_raster(tmsd, sensors = c("TMS\_T3"))

p1

tiff(file = "tms\_ras.tiff", width = 7, height = 10, units = 'in', res = 600, pointsize = 11)

p1

dev.off()

#dendrometer aggregation

agg\_data <- mc\_calc\_tomst\_dendro(tmsd, localities=NULL)

tmsd.long\_mm <- mc\_reshape\_long(agg\_data, sensors = "Dendro\_raw")

tmsd.long\_t <- mc\_reshape\_long(agg\_data, sensors = "Dendro\_T")

tmsd.wide <- mc\_reshape\_wide(agg\_data, sensors = c("Dendro\_raw", "Dendro\_T"))

#tms.all.wide <- mc\_agg(tmsd.wide, fun = c("mean", "range", "coverage", "percentile"),

# percentiles = 95, period = "hour", min\_coverage = 0.95)

dend = ggplot(tmsd.long\_mm, aes(x = datetime, y = value, fill = locality\_id)) +

geom\_line(aes(fill = locality\_id)) +

geom\_point(size = 2) +

theme\_bw()

dend

#Data wrangling - HOBO/Ecomatik

#Matching up different datasets and with temperature data

d\_temp1 = read.csv("d\_temp1.csv", header = TRUE)#read-in data

d\_temp1$date = mdy(d\_temp1$date)

d\_temp2 = subset(d\_temp1, sensor\_type == "TOMST" | sensor\_type == "ECO")

#d\_temp2$diff = d\_temp2$mean\_value - d\_temp2$baseline

d\_temp\_eco = subset(d\_temp2, sensor\_type == "ECO")

d\_temp\_tomst = subset(d\_temp2, sensor\_type == "TOMST")

d\_temp2$elevation = as.numeric(d\_temp2$elevation)

hist(d\_temp2$mean\_value)#data histogram visualization

hist(d\_temp2$temperatue)

hist(d\_temp\_eco$mean\_value)

hist(d\_temp\_tomst$mean\_value)

hist(d\_temp2$diff)

plot(d\_temp\_tomst$date, d\_temp\_tomst$mean\_value)

plot(d\_temp\_eco$date, d\_temp\_eco$mean\_value)

plot(d\_temp2$temperatue, d\_temp2$diff)

d\_temp\_wide = cast(d\_temp2, date + elevation + tree + species + init\_diam +

size\_class + temperature ~ sensor\_type)

#d\_temp\_wide$elevation = factor(d\_temp\_wide$elevation, levels = c("550", "600", "800", "1000", "1200", "1400", "1450", "1500", "1550"))

d\_temp\_wide$sensor\_diff = d\_temp\_wide$ECO - d\_temp\_wide$TOMST

d\_temp\_wide$elevation = as.numeric(d\_temp\_wide$elevation)

hist(d\_temp\_wide$sensor\_diff)

mean(d\_temp\_wide$sensor\_diff, na.rm = TRUE)

favstats(d\_temp\_wide$sensor\_diff)#get summary stats

plot(d\_temp\_wide$ECO, d\_temp\_wide$TOMST)

plot(d\_temp\_wide$temperature, d\_temp\_wide$TOMST)

plot(d\_temp\_wide$temperature, d\_temp\_wide$ECO)

plot(d\_temp\_wide$date, d\_temp\_wide$ECO)

plot(d\_temp\_wide$date, d\_temp\_wide$TOMST)

###T-tests, partition by elevation, tree size class, species

##Use d\_temp\_wide

##calculate onset of spring growth

##All data - ECO vs. TOMST

#t-tests + mixed models + icc

gm1 <- glmer(diff ~ sensor\_type + temperature + size\_class +

(1 | tree) + (1 | species),

family = gaussian, data = d\_temp2)

summary(gm1)

gm12 <- glmer(diff ~ sensor\_type + size\_class + temperature + elevation + size\_class +

(1 | tree) + (1 | species),

family = gaussian, data = d\_temp2)

summary(gm12)

cor.test(d\_temp\_wide$TOMST, d\_temp\_wide$ECO)

sub\_dat = subset(d\_temp\_wide, select = c("ECO", "TOMST"))

icc(sub\_dat, model = "twoway", type = "consistency", unit = "single")

gm2 <- glmer(diff ~ sensor\_type +

(1 | tree) + (1 | species),

family = gaussian, data = d\_temp2)

summary(gm2)

t.test(d\_temp\_wide$TOMST, d\_temp\_wide$ECO)

Cand.models <- list()

Cand.models[[1]] <- glmer(diff ~ sensor\_type + temperature + size\_class + elevation +

(1 | tree) + (1 | species),

family = gaussian, data = d\_temp2)

Cand.models[[2]] <- glmer(diff ~ sensor\_type + temperature + size\_class +

(1 | tree) + (1 | species),

family = gaussian, data = d\_temp2)

Cand.models[[3]] <- glmer(diff ~ sensor\_type + temperature +

(1 | tree) + (1 | species),

family = gaussian, data = d\_temp2)

Cand.models[[4]] <- glmer(diff ~ sensor\_type +

(1 | tree) + (1 | species),

family = gaussian, data = d\_temp2)

Cand.models[[5]] <- glmer(diff ~ (1 | tree) + (1 | species),

family = gaussian, data = d\_temp2)

Modnames <- paste("mod", 1:length(Cand.models), sep = " ")

aictab(cand.set = Cand.models, modnames = Modnames, sort = TRUE)

print(aictab(cand.set = Cand.models, modnames = Modnames, sort = TRUE), digits = 4, LL = TRUE)

#Direct comparisons with 1:1 line

time\_plot\_t = ggplot(d\_temp\_wide, aes(x = date, y = TOMST, color = size\_class)) +

geom\_point() +

xlab("Date") +

ylab("TOMST (\u0394\u03BCm from baseline)") +

ylim(-1000,5000) +

xlim(as.Date(c('1/9/2023', '1/10/2024'), format="%d/%m/%Y")) +

theme\_bw() +

scale\_color\_manual(name = "Tree Class", values = c("orange", "blue", "forestgreen"),

labels = c("Krummholz", "Abies", "Acer")) +

annotate('rect', xmin=as.Date('24/5/2024', format="%d/%m/%Y"), xmax=as.Date('1/7/2024', format="%d/%m/%Y"), ymin=-Inf, ymax=Inf, alpha=.4, fill='grey50') +

theme(legend.position = c(0.11,0.85)) +

annotate("text", x = as.Date('10/6/2024', format="%d/%m/%Y"), y = 4650, label = "Start\n of\n Spring")

time\_plot\_t

tiff(file = "time\_plot\_t.tiff", width = 6, height = 5, units = 'in', res = 600, pointsize = 11)

time\_plot\_t

dev.off()

time\_plot\_e = ggplot(d\_temp\_wide, aes(x = date, y = ECO, color = size\_class)) +

geom\_point() +

xlab("Date") +

ylab("ECOMATIK (\u0394\u03BCm from baseline)") +

ylim(-500,5000) +

theme\_bw()

time\_plot\_e

all\_plot = ggplot(d\_temp\_wide, aes(x = ECO, y = TOMST)) +

geom\_point(pch = 1, color = "forestgreen", size = 3) +

geom\_abline(slope = 1, intercept = 0, lty = 1, lwd = 1) +

geom\_abline(slope = 0.759535, intercept = 64.76, lty = 4, lwd = 1) +

xlab("ECOMATIK (\u0394\u03BCm from baseline)") +

ylab("TOMST (\u0394\u03BCm from baseline)") +

ylim(-500,6000) +

xlim(-500, 4000) +

annotate("text", x = 300, y = 5900, label = "ICC = 0.75 (Good Reliability)") +

theme\_bw()

all\_plot

tiff(file = "all\_plot.tiff", width = 6, height = 5, units = 'in', res = 600, pointsize = 11)

all\_plot

dev.off()

lm\_t = lm(TOMST ~ ECO, data = d\_temp\_wide)

summary(lm\_t)

quantile(d\_temp\_wide$sensor\_diff, 0.975, na.rm = TRUE)

quantile(d\_temp\_wide$sensor\_diff, 0.025, na.rm = TRUE)

all\_plot\_hist = ggplot(d\_temp\_wide, aes(x = sensor\_diff)) +

geom\_histogram(fill = "forestgreen") +

xlab("ECOMATIK - TOMST (\u0394\u03BCm from baseline)") +

ylab("Measurement Count") +

geom\_vline(xintercept = 168, color = "blue", lwd = 1) +

geom\_vline(xintercept = -40, color = "red", lwd = 1) +

geom\_vline(xintercept = 361, color = "red", lwd = 1) +

geom\_vline(xintercept = 1610, color = "red", lty = 3, lwd = 1) +

geom\_vline(xintercept = -1107, color = "red", lty = 3, lwd = 1) +

annotate("text", x = -1300, y = 1800, label = "2.5%") +

annotate("text", x = -200, y = 1800, label = "25%") +

annotate("text", x = 550, y = 1800, label = "75%") +

annotate("text", x = 1850, y = 1800, label = "97.5%") +

theme\_bw()

all\_plot\_hist

tiff(file = "all\_plot\_hist.tiff", width = 6, height = 5, units = 'in', res = 600, pointsize = 11)

all\_plot\_hist

dev.off()

median(d\_temp\_wide$ECO, na.rm = TRUE)

median(d\_temp\_wide$TOMST, na.rm = TRUE)

all\_plot\_bar = ggplot(d\_temp2, aes(x = diff, fill = sensor\_type)) +

geom\_histogram(alpha = 0.8, aes(y=..density..)) +

geom\_density(lwd = 1, alpha = 0.25) +

xlab("\u0394\u03BCm from baseline") +

ylab("Density") +

#geom\_vline(xintercept = 573, color = "orange", lwd = 0.9) +

#geom\_vline(xintercept = 384, color = "blue", lwd = 0.9) +

scale\_fill\_manual(name = "Sensor Type", values = c("orange", "blue")) +

theme\_bw() +

theme(legend.position = c(0.85, 0.85))

all\_plot\_bar

tiff(file = "all\_plot\_bar.tiff", width = 6, height = 5, units = 'in', res = 600, pointsize = 11)

all\_plot\_bar

dev.off()

all\_plot1 = ggplot(d\_temp\_wide, aes(x = ECO, y = TOMST, color = size\_class)) +

geom\_point(pch = 1, size = 2) +

geom\_abline(slope = 1, intercept = 0, lty = 1, lwd = 1) +

geom\_smooth(method = "lm", se = FALSE) +

xlab("ECOMATIK (\u0394\u03BCm from baseline)") +

ylab("TOMST (\u0394\u03BCm from baseline)") +

scale\_color\_manual(name = "Size Class", values = c("orange", "blue", "forestgreen"),

labels = c("Krummholz", "Large", "Small")) +

ylim(-500,6000) +

xlim(-500, 4000) +

theme\_bw() +

theme(legend.position = c(0.895, 0.85))

all\_plot1

tiff(file = "all\_plot1.tiff", width = 6, height = 5, units = 'in', res = 600, pointsize = 11)

all\_plot1

dev.off()

all\_plot2 = ggplot(d\_temp\_wide, aes(x = ECO, y = TOMST, fill = elevation)) +

geom\_point(pch = 21, size = 2) +

geom\_abline(slope = 1, intercept = 0, lty = 1, lwd = 1) +

xlab("ECOMATIK (\u0394\u03BCm from baseline)") +

ylab("TOMST (\u0394\u03BCm from baseline)") +

scale\_fill\_viridis\_b(name = "Elevation (m)", breaks = c(500,800,1100,1400,1700)) +

ylim(-500,6000) +

xlim(-500, 4000) +

theme(legend.position = c(0.85, 0.85)) +

theme\_bw()

all\_plot2

tiff(file = "all\_plot2.tiff", width = 6, height = 5, units = 'in', res = 600, pointsize = 11)

all\_plot2

dev.off()

time\_plot\_diff = ggplot(d\_temp\_wide, aes(x = date, y = sensor\_diff, color = size\_class)) +

geom\_point() +

xlab("Date") +

ylab("ECOMATIK - TOMST (\u0394\u03BCm from baseline)") +

ylim(-3000,2000) +

theme\_bw()

time\_plot\_diff

#Regressions with temperature - use temps value - extra stuff

DE1 = read.csv("Book12.csv", header = TRUE)

DE2 = ddply(DE1, ~elevation + date, summarise, temperature = mean(Air\_temp))

DE2$date = mdy(DE2$date)

write.csv(DE2, "DE2.csv")

d\_temp\_wide1 = merge(d\_temp\_wide, DE2, by = c("elevation", "date"), all = TRUE)

d\_temp\_wide2 = d\_temp\_wide1 %>%

mutate(temperature = coalesce(temperature.x, temperature.y))

t\_plot\_t = ggplot(d\_temp\_wide2, aes(x = temperature, y = TOMST, color = size\_class)) +

geom\_point(pch = 1, alpha = 0.3) +

geom\_smooth(method = "loess", se = FALSE, lwd = 1.5) +

#geom\_smooth(method = "lm", formula = y ~ splines::ns(x, 5), se=FALSE, lwd = 2) +

xlab("Temperature \u00B0C") +

ylab("TOMST (\u0394\u03BCm from baseline)") +

scale\_color\_manual(name = "Tree Class", values = c("orange", "blue", "forestgreen"),

labels = c("Krummholz", "Abies", "Acer")) +

ylim(-1000,5000) +

theme\_bw() +

theme(legend.position = c(0.895, 0.85))

t\_plot\_t

tiff(file = "t\_plot\_t.tiff", width = 6, height = 5, units = 'in', res = 600, pointsize = 11)

t\_plot\_t

dev.off()

t\_plot\_e = ggplot(d\_temp\_wide2, aes(x = temperature, y = ECO, color = size\_class)) +

geom\_point(pch = 1) +

geom\_smooth(method = "loess", se = FALSE, lwd = 1.5) +

#geom\_smooth(method = "lm", formula = y ~ splines::ns(x, 5), se=FALSE, lwd = 2) +

xlab("Temperature \u00B0C") +

ylab("ECOMATIK (\u0394\u03BCm from baseline)") +

scale\_color\_manual(name = "Tree Class", values = c("orange", "blue", "forestgreen"),

labels = c("Krummholz", "Abies", "Acer")) +

ylim(-1000,4000) +

theme\_bw() +

theme(legend.position = c(0.895, 0.85))

t\_plot\_e

tiff(file = "t\_plot\_e.tiff", width = 6, height = 5, units = 'in', res = 600, pointsize = 11)

t\_plot\_e

dev.off()

#Extra regressions including below groups (model structure)

gm1 <- glmer(diff ~ sensor\_type + size\_class + temperature +

(1 | tree) + (1 | species),

family = gaussian, data = d\_temp2)

summary(gm1)

temp\_count = ddply(d\_temp\_wide2, ~elevation, summarise, dia1 = min(init\_diam), dia2 = max(init\_diam))