Met\_Master Merge with Eddy\_Pro\_Master: Data from 2019-11-18

library(tidyverse)

## Registered S3 methods overwritten by 'ggplot2':  
## method from   
## [.quosures rlang  
## c.quosures rlang  
## print.quosures rlang

## -- Attaching packages ------------------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.1.1 v purrr 0.3.2  
## v tibble 2.1.2 v dplyr 0.8.1  
## v tidyr 0.8.3 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.4.0

## -- Conflicts ---------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

source("calc\_footprint\_FFP.R")  
library(ggplot2)  
library(modelr)  
options(na.action = na.warn)

# Lists

rm(list=ls())

# Defining the data I/O directory for the Eddy Master File

\*\*

path\_eddy<-"C:\\Users\\Tommy\\flux\\Data-exploring\\02\_Concord\\"  
path.in\_eddy<-paste(path\_eddy,"01\_Proccessed\_Data",sep="")  
path.out\_eddy<-paste(path\_eddy,"03\_combined\_data",sep="")  
ver<-"Master\_Eddy"   
file.name\_eddy<-paste("master\_eddy\_pro\_concord",sep="")

# read in eddypro full output Master file, parse variable names

data\_master\_eddy<-read.csv(paste(path.in\_eddy,"\\",ver,"\\",file.name\_eddy,".csv",sep=""),  
 header=F,  
 skip=3,  
 na.strings=c(-9999),  
 stringsAsFactors = F)  
colnames(data\_master\_eddy)<-colnames(  
 read.csv(paste(path.in\_eddy,"\\",ver,"\\",file.name\_eddy,".csv",sep=""),  
 header=T,  
 skip=1))

# Defining the data I/O directory for the Master met\_data

path\_met<-"C:\\Users\\Tommy\\flux\\Data-exploring\\02\_Concord\\"  
path.in\_met<-paste(path\_met,"01\_Proccessed\_Data",sep="")  
path.out\_met<-paste(path\_met,"03\_combined\_data",sep="")  
ver<-"met\_data"   
file.name<-paste("MET\_data\_master",sep="")

#read in Met\_Data Master file, parse variable names and define N/As#

met\_data\_master<-read.csv(paste(path.in\_met,"\\",ver,"\\",file.name,".csv",sep=""),  
 header=F,  
 skip=4,  
 na.strings=c("NAN"),  
 stringsAsFactors = F)  
colnames(met\_data\_master)<-colnames(  
 read.csv(paste(path.in\_met,"\\",ver,"\\",file.name,".csv",sep=""),  
 header=T,  
 skip= 1))

# Parsing the time stamp converting it into a POSIXlt vector

**interpreting date and time into new timestamp column**

**Then taking that time stamp column and turning each time into a unique number (time.id) so I can join based on that. As it can be really tricky to join/merge based on time stamps alone**

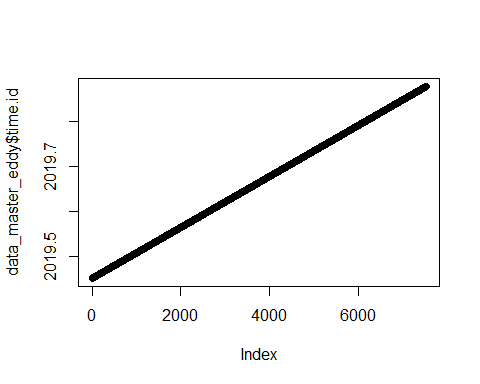
**Or I could make sure both time stamps are characters and match them that way**

**Finally ploting time.id to make sure my times translate linearily**

data\_master\_eddy$TIMESTAMP<-strptime(paste(data\_master\_eddy$date,data\_master\_eddy$time,sep=" "),format="%m/%d/%Y %H:%M", tz = "GMT")  
  
data\_master\_eddy$time.id<-data\_master\_eddy$TIMESTAMP$year+1900+(data\_master\_eddy$TIMESTAMP$yday)/366+(data\_master\_eddy$TIMESTAMP$hour)/366/24+ (data\_master\_eddy$TIMESTAMP$min)/366/24/60  
  
data\_master\_eddy$time.id[1:50]

## [1] 2019.450 2019.450 2019.450 2019.450 2019.450 2019.450 2019.450  
## [8] 2019.450 2019.451 2019.451 2019.451 2019.451 2019.451 2019.451  
## [15] 2019.451 2019.451 2019.451 2019.451 2019.451 2019.451 2019.451  
## [22] 2019.451 2019.451 2019.451 2019.451 2019.452 2019.452 2019.452  
## [29] 2019.452 2019.452 2019.452 2019.452 2019.452 2019.452 2019.452  
## [36] 2019.452 2019.452 2019.452 2019.452 2019.452 2019.452 2019.452  
## [43] 2019.452 2019.453 2019.453 2019.453 2019.453 2019.453 2019.453  
## [50] 2019.453

plot(data\_master\_eddy$time.id)



which(duplicated(data\_master\_eddy$time.id))

## integer(0)

#Taking the met\_data and turning the time stamp into posixt format#

met\_data\_master$TIMESTAMP<-strptime(met\_data\_master$TIMESTAMP,  
 format ="%m/%d/%Y %H:%M", tz = "GMT")  
  
met\_data\_master$TIMESTAMP[1:20]

## [1] "2019-06-25 09:00:00 GMT" "2019-06-25 09:30:00 GMT"  
## [3] "2019-06-25 10:00:00 GMT" "2019-06-25 10:30:00 GMT"  
## [5] "2019-06-25 11:00:00 GMT" "2019-06-25 11:30:00 GMT"  
## [7] "2019-06-25 12:00:00 GMT" "2019-06-25 12:30:00 GMT"  
## [9] "2019-06-25 13:00:00 GMT" "2019-06-25 13:30:00 GMT"  
## [11] "2019-06-25 14:00:00 GMT" "2019-06-25 14:30:00 GMT"  
## [13] "2019-06-25 15:00:00 GMT" "2019-06-25 15:30:00 GMT"  
## [15] "2019-06-25 16:00:00 GMT" "2019-06-25 16:30:00 GMT"  
## [17] "2019-06-25 17:00:00 GMT" "2019-06-25 17:30:00 GMT"  
## [19] "2019-06-25 18:00:00 GMT" "2019-06-25 18:30:00 GMT"

#Making sure timestamp columns line up#

met\_data\_master$TIMESTAMP[1:10]

## [1] "2019-06-25 09:00:00 GMT" "2019-06-25 09:30:00 GMT"  
## [3] "2019-06-25 10:00:00 GMT" "2019-06-25 10:30:00 GMT"  
## [5] "2019-06-25 11:00:00 GMT" "2019-06-25 11:30:00 GMT"  
## [7] "2019-06-25 12:00:00 GMT" "2019-06-25 12:30:00 GMT"  
## [9] "2019-06-25 13:00:00 GMT" "2019-06-25 13:30:00 GMT"

data\_master\_eddy$TIMESTAMP[1:10]

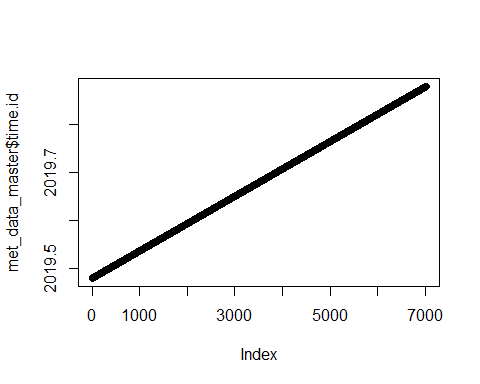
## [1] "2019-06-14 17:30:00 GMT" "2019-06-14 18:00:00 GMT"  
## [3] "2019-06-14 18:30:00 GMT" "2019-06-14 19:00:00 GMT"  
## [5] "2019-06-14 19:30:00 GMT" "2019-06-14 20:00:00 GMT"  
## [7] "2019-06-14 20:30:00 GMT" "2019-06-14 21:00:00 GMT"  
## [9] "2019-06-14 21:30:00 GMT" "2019-06-14 22:00:00 GMT"

#creating a time id for the MET Data so I I can join the MET and Eddy Pro Data#

met\_data\_master$time.id <-met\_data\_master$TIMESTAMP$year+1900+(met\_data\_master$TIMESTAMP$yday)/366+(met\_data\_master$TIMESTAMP$hour)/366/24 + (met\_data\_master$TIMESTAMP$min)/366/24/60   
  
met\_data\_master$time.id[1:20]

## [1] 2019.479 2019.479 2019.479 2019.479 2019.479 2019.479 2019.480  
## [8] 2019.480 2019.480 2019.480 2019.480 2019.480 2019.480 2019.480  
## [15] 2019.480 2019.480 2019.480 2019.480 2019.480 2019.480

plot(met\_data\_master$time.id)



which(duplicated(met\_data\_master$time.id))

## integer(0)

#Joining the Met\_Data and Eddy Pro Data Sets#

**with merge**

combo\_master\_ed\_met<- merge(met\_data\_master[,-which(colnames(met\_data\_master)=="TIMESTAMP")], data\_master\_eddy[,-which(colnames(data\_master\_eddy)=="TIMESTAMP")], by = "time.id")  
  
  
  
colnames(combo\_master\_ed\_met)

## [1] "time.id" "RECORD"   
## [3] "BattV\_Avg" "PTemp\_C\_Avg"   
## [5] "AM25T\_ref\_Avg" "TC\_Avg.1."   
## [7] "TC\_Avg.2." "TC\_Avg.3."   
## [9] "TC\_Avg.4." "TC\_Avg.5."   
## [11] "TC\_Avg.6." "TC\_Avg.7."   
## [13] "TC\_Avg.8." "TC\_Avg.9."   
## [15] "TC\_Avg.10." "TC\_Avg.11."   
## [17] "TC\_Avg.12." "TC\_Avg.13."   
## [19] "TC\_Avg.14." "TC\_Avg.15."   
## [21] "TC\_Avg.16." "TC\_Avg.17."   
## [23] "TC\_Avg.18." "TC\_Avg.19."   
## [25] "TC\_Avg.20." "TC\_Avg.21."   
## [27] "TC\_Avg.22." "TC\_Avg.23."   
## [29] "TC\_Avg.24." "TC\_Avg.25."   
## [31] "AirT\_Avg" "RH\_Avg"   
## [33] "AtmPressure\_Avg" "NR\_mV\_Avg"   
## [35] "NR\_Wm2\_Avg" "PAR\_in\_mV\_Avg"   
## [37] "PAR\_in\_uEm2\_Avg" "PAR\_out\_mV\_Avg"   
## [39] "PAR\_out\_uEm2\_Avg" "SHF\_1\_mV\_Avg"   
## [41] "SHF\_1\_Wm2\_Avg" "SHF\_2\_mV\_Avg"   
## [43] "SHF\_2\_Wm2\_Avg" "WaterP\_Avg"   
## [45] "WaterT\_Avg" "Precip\_mm\_Tot"   
## [47] "VWC\_Avg" "EC\_Avg"   
## [49] "T\_Avg" "P\_Avg"   
## [51] "PA\_Avg" "VR\_Avg"   
## [53] "filename" "date"   
## [55] "time" "DOY"   
## [57] "daytime" "file\_records"   
## [59] "used\_records" "Tau"   
## [61] "qc\_Tau" "H"   
## [63] "qc\_H" "LE"   
## [65] "qc\_LE" "co2\_flux"   
## [67] "qc\_co2\_flux" "h2o\_flux"   
## [69] "qc\_h2o\_flux" "H\_strg"   
## [71] "LE\_strg" "co2\_strg"   
## [73] "h2o\_strg" "co2\_v.adv"   
## [75] "h2o\_v.adv" "co2\_molar\_density"   
## [77] "co2\_mole\_fraction" "co2\_mixing\_ratio"   
## [79] "co2\_time\_lag" "co2\_def\_timelag"   
## [81] "h2o\_molar\_density" "h2o\_mole\_fraction"   
## [83] "h2o\_mixing\_ratio" "h2o\_time\_lag"   
## [85] "h2o\_def\_timelag" "sonic\_temperature"   
## [87] "air\_temperature" "air\_pressure"   
## [89] "air\_density" "air\_heat\_capacity"   
## [91] "air\_molar\_volume" "ET"   
## [93] "water\_vapor\_density" "e"   
## [95] "es" "specific\_humidity"   
## [97] "RH" "VPD"   
## [99] "Tdew" "u\_unrot"   
## [101] "v\_unrot" "w\_unrot"   
## [103] "u\_rot" "v\_rot"   
## [105] "w\_rot" "wind\_speed"   
## [107] "max\_wind\_speed" "wind\_dir"   
## [109] "yaw" "pitch"   
## [111] "roll" "u."   
## [113] "TKE" "L"   
## [115] "X.z.d..L" "bowen\_ratio"   
## [117] "T." "model"   
## [119] "x\_peak" "x\_offset"   
## [121] "x\_10." "x\_30."   
## [123] "x\_50." "x\_70."   
## [125] "x\_90." "un\_Tau"   
## [127] "Tau\_scf" "un\_H"   
## [129] "H\_scf" "un\_LE"   
## [131] "LE\_scf" "un\_co2\_flux"   
## [133] "co2\_scf" "un\_h2o\_flux"   
## [135] "h2o\_scf" "spikes\_hf"   
## [137] "amplitude\_resolution\_hf" "drop\_out\_hf"   
## [139] "absolute\_limits\_hf" "skewness\_kurtosis\_hf"   
## [141] "skewness\_kurtosis\_sf" "discontinuities\_hf"   
## [143] "discontinuities\_sf" "timelag\_hf"   
## [145] "timelag\_sf" "attack\_angle\_hf"   
## [147] "non\_steady\_wind\_hf" "u\_spikes"   
## [149] "v\_spikes" "w\_spikes"   
## [151] "ts\_spikes" "co2\_spikes"   
## [153] "h2o\_spikes" "chopper\_LI.7500"   
## [155] "detector\_LI.7500" "pll\_LI.7500"   
## [157] "sync\_LI.7500" "mean\_value\_RSSI\_LI.7500"   
## [159] "u\_var" "v\_var"   
## [161] "w\_var" "ts\_var"   
## [163] "co2\_var" "h2o\_var"   
## [165] "w.ts\_cov" "w.co2\_cov"   
## [167] "w.h2o\_cov" "vin\_sf\_mean"   
## [169] "co2\_mean" "h2o\_mean"   
## [171] "dew\_point\_mean" "co2\_signal\_strength\_7500\_mean"

#Add back time stamp to the combo\_master\_ed\_met#

combo\_master\_ed\_met$TIMESTAMP<-strptime(paste(combo\_master\_ed\_met$date,combo\_master\_ed\_met$time,sep=" "),format="%m/%d/%Y %H:%M", tz = "GMT")  
  
combo\_master\_ed\_met$TIMESTAMP[1:10]

## [1] "2019-06-25 09:00:00 GMT" "2019-06-25 09:30:00 GMT"  
## [3] "2019-06-25 10:00:00 GMT" "2019-06-25 10:30:00 GMT"  
## [5] "2019-06-25 11:00:00 GMT" "2019-06-25 11:30:00 GMT"  
## [7] "2019-06-25 12:00:00 GMT" "2019-06-25 12:30:00 GMT"  
## [9] "2019-06-25 13:00:00 GMT" "2019-06-25 13:30:00 GMT"

#Creating a CSV File of my combine Master File!#

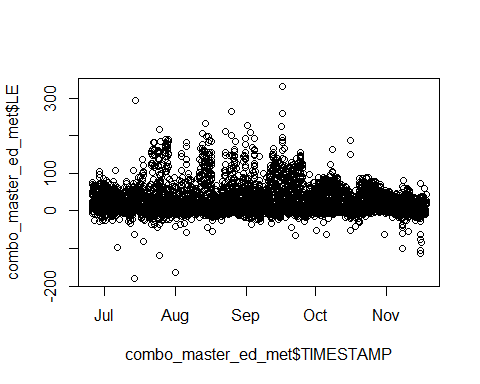
write.csv(combo\_master\_ed\_met,  
 paste(path.out\_eddy,ver,"combo\_master\_ed\_met",sep=""),  
 quote = T,  
 row.names = F)

#filtering data for quality control filters# **filtering latent heat, sensible heat, co2 flux, qc\_tau and h20flux by quality controls**

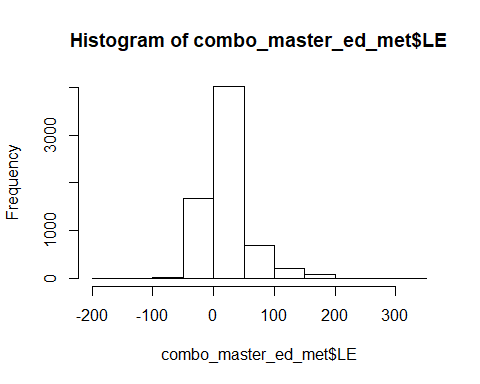
combo\_master\_ed\_met$LE.[!is.na(combo\_master\_ed\_met$qc\_LE)&combo\_master\_ed\_met$qc\_LE==2]<-NA  
  
combo\_master\_ed\_met$qqc\_h2o\_flux[!is.na(combo\_master\_ed\_met$qc\_h2o\_flux)&combo\_master\_ed\_met$qc\_h2o\_flux==2]<-NA  
  
combo\_master\_ed\_met$H[!is.na(combo\_master\_ed\_met$qc\_H)&combo\_master\_ed\_met$qc\_H==2]<-NA  
  
combo\_master\_ed\_met$u.[!is.na(combo\_master\_ed\_met$qc\_Tau)&combo\_master\_ed\_met$qc\_Tau==2]<-NA  
  
combo\_master\_ed\_met$co2\_flux[!is.na(combo\_master\_ed\_met$co2\_flux)&combo\_master\_ed\_met$co2\_flux==2]<-NA

# latent heat, sensible heat, h20\_flux, and relative humidity by day for whole period

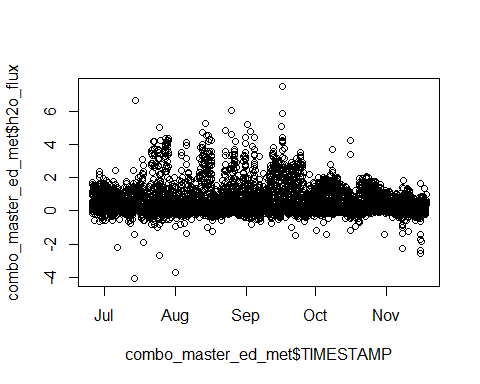
plot(combo\_master\_ed\_met$TIMESTAMP ,combo\_master\_ed\_met$LE)



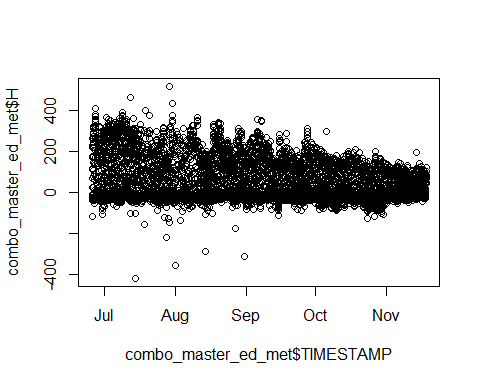
hist(combo\_master\_ed\_met$LE)



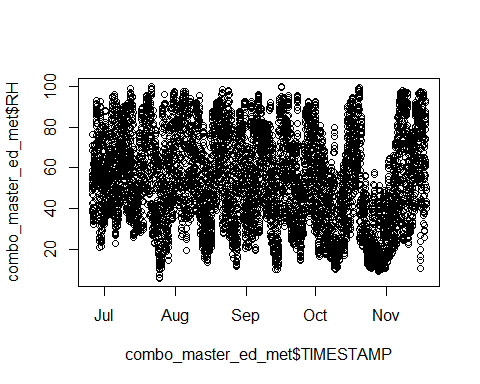
plot(combo\_master\_ed\_met$TIMESTAMP ,combo\_master\_ed\_met$h2o\_flux)



plot(combo\_master\_ed\_met$TIMESTAMP ,combo\_master\_ed\_met$H)

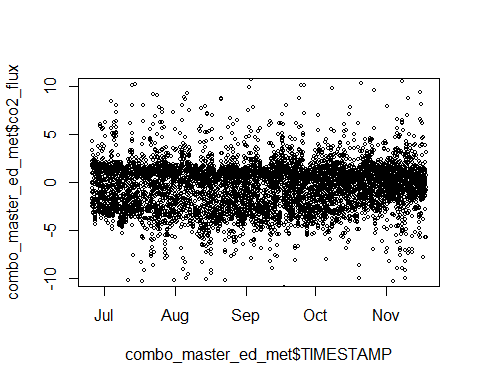


plot(combo\_master\_ed\_met$TIMESTAMP ,combo\_master\_ed\_met$RH)

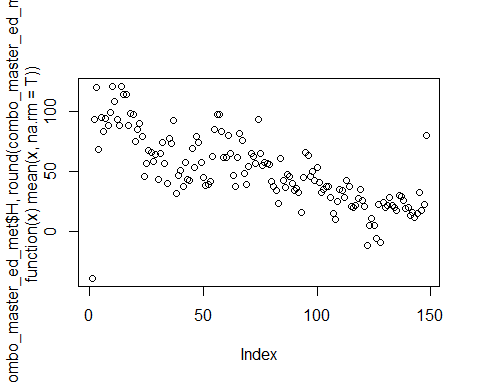


#Plotting Co2 flux by timstamp on x-axis.

plot(combo\_master\_ed\_met$TIMESTAMP ,combo\_master\_ed\_met$co2\_flux, ylim = c(-10,10),cex=0.6)

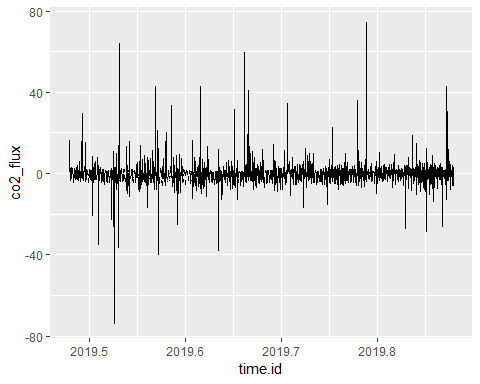


plot(tapply(combo\_master\_ed\_met$H,round(combo\_master\_ed\_met$DOY),function(x) mean(x,na.rm=T)))



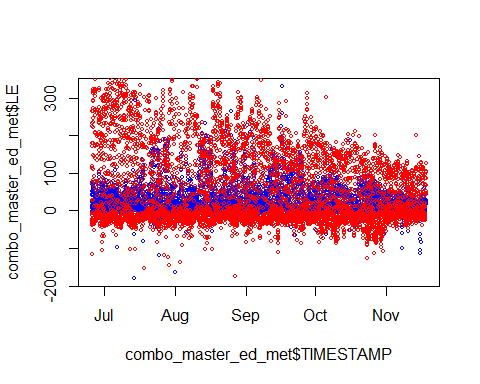
#line plot of c02\_fluxes#

ggplot(data = combo\_master\_ed\_met) +   
 geom\_line(mapping = aes(x = time.id , y = co2\_flux))



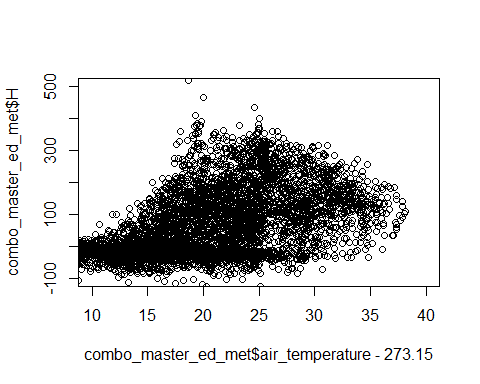
#latent heat and Sensible heat plotted over eachother

plot(combo\_master\_ed\_met$TIMESTAMP ,combo\_master\_ed\_met$LE,pch=1,col="blue",cex=0.6)  
points(combo\_master\_ed\_met$TIMESTAMP ,combo\_master\_ed\_met$H,col="red",pch=1,cex=0.6)

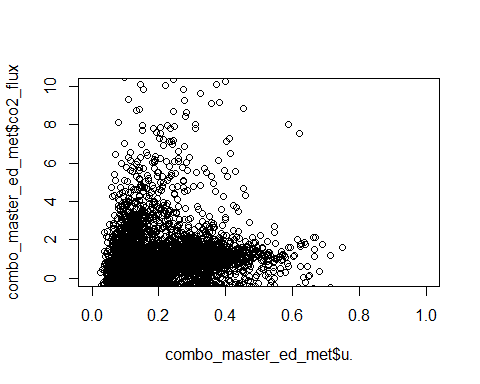


#air temperatue by sensible heat an U\* by co2 flux

plot(combo\_master\_ed\_met$air\_temperature-273.15,combo\_master\_ed\_met$H,xlim=c(10,40),ylim=c(-100,500))



plot(combo\_master\_ed\_met$u. ,combo\_master\_ed\_met$co2\_flux, ylim = c(0, 10), xlim = c(0,1))



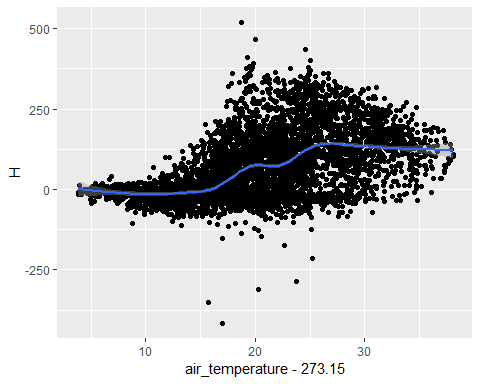
#Adding best fit line to air temperature by sensible heat

ggplot(data = combo\_master\_ed\_met) +   
 geom\_point(mapping = aes(x = air\_temperature-273.15, y = H)) +  
 geom\_smooth(mapping = aes(x = air\_temperature-273.15, y = H))

## `geom\_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'

## Warning: Removed 875 rows containing non-finite values (stat\_smooth).

## Warning: Removed 875 rows containing missing values (geom\_point).



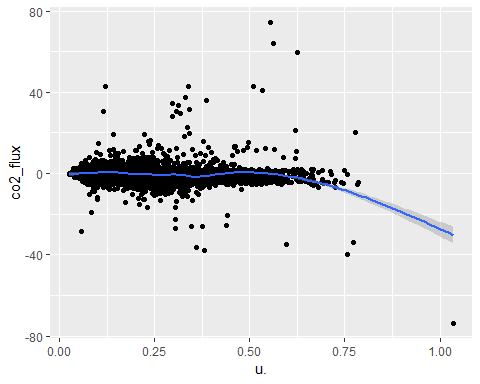
#best fit line to U\* bt co2 flux

ggplot(data = combo\_master\_ed\_met) +   
 geom\_point(mapping = aes(x = u., y = co2\_flux)) +  
 geom\_smooth(mapping = aes(x = u., y = co2\_flux))

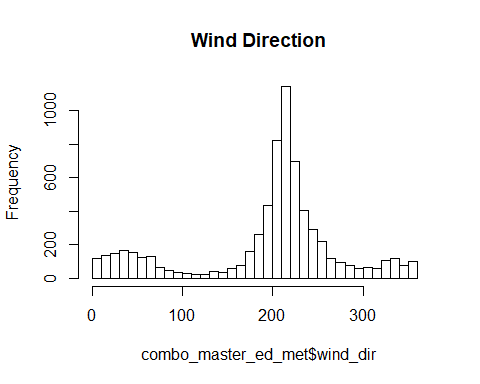
## `geom\_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'

## Warning: Removed 725 rows containing non-finite values (stat\_smooth).

## Warning: Removed 725 rows containing missing values (geom\_point).

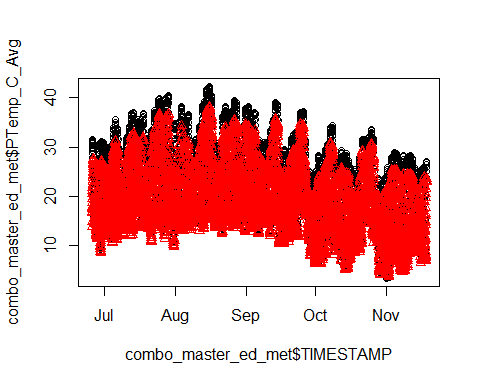


library(tidyverse)  
hist(combo\_master\_ed\_met$wind\_dir, xlim = c(0,370), breaks = 36, main = "Wind Direction")



#Comparing MET Temperature data to temperature reading from Licor intruments#

plot(combo\_master\_ed\_met$TIMESTAMP ,combo\_master\_ed\_met$PTemp\_C\_Avg)  
points(combo\_master\_ed\_met$TIMESTAMP,combo\_master\_ed\_met$air\_temperature-273.15,col="red",pch=2)



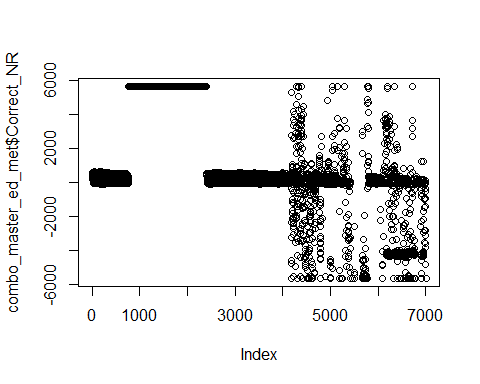
#Adding coefficient to Net radiation#

combo\_master\_ed\_met$Correct\_NR = (combo\_master\_ed\_met$NR\_Wm2\_Avg\*10)/14.2

summary(combo\_master\_ed\_met$Correct\_NR[c(1:600,2500:4000)])

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -106.27 -69.79 -17.23 121.53 339.30 644.37

plot(combo\_master\_ed\_met$Correct\_NR)



filtering out bad NR numbers

combo\_master\_ed\_met$Correct\_NR[!is.na(combo\_master\_ed\_met$Correct\_NR)&(combo\_master\_ed\_met$Correct\_NR<(-150)|combo\_master\_ed\_met$Correct\_NR>800)]<-NA  
summary(combo\_master\_ed\_met$Correct\_NR)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## -146.97 -67.96 -49.57 88.05 250.14 775.35 3106

plot(combo\_master\_ed\_met$Correct\_NR)

