Chapter 5 Flux Coding Lab

Emma Reich

9/24/2020

1. NEON data are submitted to AmeriFlux quarterly after one year of non-quality flagged or otherwise missing data are available. Use the workflow above to extend the data coverage of an already submitted NEON site by downloading existing data from the AmeriFlux website and recently published HDF5 files from the NEON data portal. Process the NEON data such that it is in AmeriFlux format and plot the entire timerseries.

```
# Load NEONprocIS.base and eddy4R.base

library(NEONprocIS.base)
library(eddy4R.base)
```

First I'll read in the Ameriflux data I downloaded to see the last date it includes.

```
# Read in the Ameriflux data
xSR <- read.csv("./xSR/AMF_US-xSR_BASE_HH_1-5.csv", header = T, skip = 2)
```

```
# Format date
xSR <- transform(xSR, TIMESTAMP_START = as.POSIXct(as.character(TIMESTAMP_START), format="%Y%m%d%H%M"))
xSR <- transform(xSR, TIMESTAMP_END = as.POSIXct(as.character(TIMESTAMP_END), format="%Y%m%d%H%M"))
```

```
# Last date included
xSR$TIMESTAMP_END[nrow(xSR)]
```

```
## [1] "2019-01-31 17:00:00 PST"
```

Now I will go through the workflow to convert the NEON data.frame to the Ameriflux format.

```
# Select site
site <- "SRER"
```

```
# Define start and end dates. I will extend the Ameriflux data by a year.
dateBgn <- "2019-02-01"
dateEnd <- "2020-01-31"

# Data package from the portal
Pack <- c('basic','expanded')[1]
#The version data for the FP standard conversion processing
ver = paste0("v",format(Sys.time(), "%Y%m%dT%H%m"))
```

```
#download directory
DirDnld= tempdir()

#Output directory, change this to where you want to save the output csv
DirOutBase <-paste0("./NEONflux",ver)

#DP number
dpID <- 'DP4.00200.001'
```

```
  #Grab a list of all Ameriflux sites, containing site ID and site description
  sites_web <- jsonlite::fromJSON("http://ameriflux-data.lbl.gov/AmeriFlux/SiteSearch.svc/SiteList/Ameri

  #Grab only NEON sites
  sitesNeon <- sites_web[grep(pattern = paste0("NEON.*",site), x = sites_web$SITE_NAME),] #For all NEON
  siteNeon <- sites_web[grep(pattern = paste0("NEON.*",site), x = sites_web$SITE_NAME),]

  metaSite <- lapply(siteNeon$SITE_ID, function(x) {
    pathSite <- paste0("http://ameriflux-data.lbl.gov/BADM/Anc/SiteInfo/",x)
    tmp <- fromJSON(pathSite)
    return(tmp)
    })
```

```
#use NEON ID as list name
  names(metaSite) <- site
```

```
# Check if dateBGn is defined
  if(!exists("dateBgn") || is.na(dateBgn) || is.null(dateBgn)){
    dateBgn <- as.Date(metaSite[[site]]$values$GRP_FLUX_MEASUREMENTS[[1]]$FLUX_MEASUREMENTS_DATE_START,
  } else {
    dateBgn <- dateBgn
  }#End of checks for missing dateBgn

  #Check if dateEnd is defined, if not make it the system date
  if(!exists("dateEnd") || is.na(dateEnd) || is.null(dateEnd)){
    dateEnd <- as.Date(Sys.Date())
  } else {
    dateEnd <- dateEnd
  }#End of checks for missing dateEnd
```

```
# Grab UTC tiem offset from the Ameriflux API
timeOfstUtc <- as.integer(metaSite[[site]]$values$GRP_UTC_OFFSET[[1]]$UTC_OFFSET)

# Create date sequence
setDate <- seq(from = as.Date(dateBgn), to = as.Date(dateEnd), by = "month")
```

```
# Start processing the site time range specified, verify that the site and date range are specified as
msg <- paste0("Starting Ameriflux FP standard conversion processing workflow for ", site, " for ", dateB
  print(msg)
```

```
## [1] "Starting Ameriflux FP standard conversion processing workflow for SRER for 2019-02-01 to 2020-0
```

```
# Create output directory by checking if the download directory exists and create it if not

if(dir.exists(DirDnld) == FALSE) dir.create(DirDnld, recursive = TRUE)
#Append the site to the base output directory
DirOut <- paste0(DirOutBase, "/", siteNeon$SITE_ID)
#Check if directory exists and create if not
if(!dir.exists(DirOut)) dir.create(DirOut, recursive = TRUE)
```

```
  #Initialize data List
  dataList <- list()

  #Read data from the API
  dataList <- lapply(setDate, function(x) {
    # year <- lubridate::year(x)
```

```r
    # mnth <- lubridate::month(x)
    date <- stringr::str_extract(x, pattern = paste0("[0-9]{4}", "-", "[0-9]{2}"))
    tryCatch(neonUtilities::zipsByProduct(dpID = dpID, site = site, startdate = date, enddate = date, pa
    files <- list.files(paste0(DirDnld, "/filesToStack00200"))
    utils::unzip(paste0(DirDnld, "/filesToStack00200/", files[grep(pattern = paste0(site,".*.", date, "
    files <- list.files(paste0(DirDnld, "/filesToStack00200"))
    dataIdx <- rhdf5::h5read(file = paste0(DirDnld, "/filesToStack00200/", max(files[grep(pattern = past

    if(!is.null(dataIdx)){
       dataIdx$dp0p <- NULL
       dataIdx$dp02 <- NULL
       dataIdx$dp03 <- NULL
       dataIdx$dp01$ucrt <- NULL
       dataIdx$dp04$ucrt <- NULL
       dataIdx$dp01$data <- lapply(dataIdx$dp01$data,FUN=function(var){
         nameTmi <- names(var)
         var <- var[grepl('_30m',nameTmi)]
         return(var)})
       dataIdx$dp01$qfqm <- lapply(dataIdx$dp01$qfqm,FUN=function(var){
         nameTmi <- names(var)
         var <- var[grepl('_30m',nameTmi)]
         return(var)})
    }
    return(dataIdx)
  })
```

```
## Finding available files
##   |                                                                       |
##
## Downloading files totaling approximately 53.092666 MB
## Downloading 1 files
##   |                                                                       |
## 1 files successfully downloaded to /var/folders/dn/tf2lcdx92m5c3yggf3y2z2w80000gn/T//RtmpbzPeXI/files
## Finding available files
##   |                                                                       |
##
## Downloading files totaling approximately 55.694833 MB

## Warning in dir.create(filepath): '/var/folders/dn/
## tf2lcdx92m5c3yggf3y2z2w80000gn/T//RtmpbzPeXI/filesToStack00200' already exists

## Downloading 1 files
##   |                                                                       |
## 1 files successfully downloaded to /var/folders/dn/tf2lcdx92m5c3yggf3y2z2w80000gn/T//RtmpbzPeXI/files
## Finding available files
##   |                                                                       |
##
## Downloading files totaling approximately 58.306112 MB

## Warning in dir.create(filepath): '/var/folders/dn/
## tf2lcdx92m5c3yggf3y2z2w80000gn/T//RtmpbzPeXI/filesToStack00200' already exists

## Downloading 1 files
##   |                                                                       |
## 1 files successfully downloaded to /var/folders/dn/tf2lcdx92m5c3yggf3y2z2w80000gn/T//RtmpbzPeXI/files
## Finding available files
##   |                                                                              |
##
## Downloading files totaling approximately 83.889813 MB
## Warning in dir.create(filepath): '/var/folders/dn/
## tf2lcdx92m5c3yggf3y2z2w80000gn/T//RtmpbzPeXI/filesToStack00200' already exists
## Downloading 1 files
##   |                                                                              |
## 1 files successfully downloaded to /var/folders/dn/tf2lcdx92m5c3yggf3y2z2w80000gn/T//RtmpbzPeXI/files
## Finding available files
##   |                                                                              |
##
## Downloading files totaling approximately 89.241434 MB
## Warning in dir.create(filepath): '/var/folders/dn/
## tf2lcdx92m5c3yggf3y2z2w80000gn/T//RtmpbzPeXI/filesToStack00200' already exists
## Downloading 1 files
##   |                                                                              |
## 1 files successfully downloaded to /var/folders/dn/tf2lcdx92m5c3yggf3y2z2w80000gn/T//RtmpbzPeXI/files
## Finding available files
##   |                                                                              |
##
## Downloading files totaling approximately 86.876634 MB
## Warning in dir.create(filepath): '/var/folders/dn/
## tf2lcdx92m5c3yggf3y2z2w80000gn/T//RtmpbzPeXI/filesToStack00200' already exists
## Downloading 1 files
##   |                                                                              |
## 1 files successfully downloaded to /var/folders/dn/tf2lcdx92m5c3yggf3y2z2w80000gn/T//RtmpbzPeXI/files
## Finding available files
##   |                                                                              |
##
## Downloading files totaling approximately 80.985694 MB
## Warning in dir.create(filepath): '/var/folders/dn/
## tf2lcdx92m5c3yggf3y2z2w80000gn/T//RtmpbzPeXI/filesToStack00200' already exists
## Downloading 1 files
##   |                                                                              |
## 1 files successfully downloaded to /var/folders/dn/tf2lcdx92m5c3yggf3y2z2w80000gn/T//RtmpbzPeXI/files
## Finding available files
##   |                                                                              |
##
## Downloading files totaling approximately 70.176781 MB
## Warning in dir.create(filepath): '/var/folders/dn/
## tf2lcdx92m5c3yggf3y2z2w80000gn/T//RtmpbzPeXI/filesToStack00200' already exists
## Downloading 1 files
##   |                                                                              |
## 1 files successfully downloaded to /var/folders/dn/tf2lcdx92m5c3yggf3y2z2w80000gn/T//RtmpbzPeXI/files
## Finding available files
##   |                                                                              |
##
## Downloading files totaling approximately 64.181372 MB
## Warning in dir.create(filepath): '/var/folders/dn/
## tf2lcdx92m5c3yggf3y2z2w80000gn/T//RtmpbzPeXI/filesToStack00200' already exists

## Downloading 1 files
##   |                                                                              |
## 1 files successfully downloaded to /var/folders/dn/tf2lcdx92m5c3yggf3y2z2w80000gn/T//RtmpbzPeXI/files
## Finding available files
##   |                                                                              |
##
## Downloading files totaling approximately 84.388163 MB

## Warning in dir.create(filepath): '/var/folders/dn/
## tf2lcdx92m5c3yggf3y2z2w80000gn/T//RtmpbzPeXI/filesToStack00200' already exists

## Downloading 1 files
##   |                                                                              |
## 1 files successfully downloaded to /var/folders/dn/tf2lcdx92m5c3yggf3y2z2w80000gn/T//RtmpbzPeXI/files
## Finding available files
##   |                                                                              |
##
## Downloading files totaling approximately 85.197847 MB

## Warning in dir.create(filepath): '/var/folders/dn/
## tf2lcdx92m5c3yggf3y2z2w80000gn/T//RtmpbzPeXI/filesToStack00200' already exists

## Downloading 1 files
##   |                                                                              |
## 1 files successfully downloaded to /var/folders/dn/tf2lcdx92m5c3yggf3y2z2w80000gn/T//RtmpbzPeXI/files
## Finding available files
##   |                                                                              |
##
## Downloading files totaling approximately 83.615578 MB

## Warning in dir.create(filepath): '/var/folders/dn/
## tf2lcdx92m5c3yggf3y2z2w80000gn/T//RtmpbzPeXI/filesToStack00200' already exists

## Downloading 1 files
##   |                                                                              |
## 1 files successfully downloaded to /var/folders/dn/tf2lcdx92m5c3yggf3y2z2w80000gn/T//RtmpbzPeXI/files
```

```r
#Add names to list for year/month combinations
names(dataList) <- paste0(lubridate::year(setDate),sprintf("%02d",lubridate::month(setDate)))
```

```r
# Remove NULL elements from list
dataList <- dataList[vapply(dataList, Negate(is.null), NA)]
```

Determine tower horizontal & vertical indices

```r
# Find the tower top level by looking at the vertical index of the turbulent CO2 concentration measurem
LvlTowr <- grep(pattern = "_30m", names(dataList[[1]]$dp01$data$co2Turb), value = TRUE)
LvlTowr <- gsub(x = LvlTowr, pattern = "_30m", replacement = "")

#get tower top level
LvlTop <- strsplit(LvlTowr,"")
LvlTop <- base::as.numeric(LvlTop[[1]][6])

# Fix towers levels because NEON and Ameriflux have opp conventions)
idxVerAmfx <- base::seq(from = 1, to = LvlTop, by = 1)
#get the sequence from top to first level
```

```
LvlMeas <- base::seq(from = LvlTop, to = 1, by = -1)
#Recreate NEON naming conventions
LvlMeas <- paste0("000_0",LvlMeas,"0",sep="")
#Give NEON naming conventions to Ameriflux vertical levels
names(idxVerAmfx) <- LvlMeas

#Ameriflux horizontal index
idxHorAmfx <- 1
```

Subset to the Ameriflux variables to convert

```
dataListFlux <- lapply(names(dataList), function(x) {
  data.frame(
        "TIMESTAMP_START" = as.POSIXlt(dataList[[x]]$dp04$data$fluxCo2$turb$timeBgn,
                                       format="%Y-%m-%dT%H:%M:%OSZ", tz = "GMT"),
        "TIMESTAMP_END" = as.POSIXlt(dataList[[x]]$dp04$data$fluxCo2$turb$timeEnd,
                                     format="%Y-%m-%dT%H:%M:%OSZ", tz = "GMT"),
        # "TIMESTAMP_START" = strftime(as.POSIXlt(dataList[[x]][[idxSite]]$dp04$data$fluxCo2$turb$timeB
        # "TIMESTAMP_END" = strftime(as.POSIXlt(dataList[[x]][[idxSite]]$dp04$data$fluxCo2$turb$timeEnd
        "FC"= dataList[[x]]$dp04$data$fluxCo2$turb$flux,
        "SC"= dataList[[x]]$dp04$data$fluxCo2$stor$flux,
        "NEE"= dataList[[x]]$dp04$data$fluxCo2$nsae$flux,
        "LE" = dataList[[x]]$dp04$data$fluxH2o$turb$flux,
        "SLE" = dataList[[x]]$dp04$data$fluxH2o$stor$flux,
        "USTAR" = dataList[[x]]$dp04$data$fluxMome$turb$veloFric,

        "H" = dataList[[x]]$dp04$data$fluxTemp$turb$flux,
        "SH" = dataList[[x]]$dp04$data$fluxTemp$stor$flux,
        "FETCH_90" = dataList[[x]]$dp04$data$foot$stat$distXaxs90,
        "FETCH_MAX" = dataList[[x]]$dp04$data$foot$stat$distXaxsMax,
        "V_SIGMA" = dataList[[x]]$dp04$data$foot$stat$veloYaxsHorSd,
        #"W_SIGMA" = dataList[[x]]$dp04$data$foot$stat$veloZaxsHorSd,
        "CO2_1_1_1" = dataList[[x]]$dp01$data$co2Turb[[paste0(LvlTowr,"_30m")]]$rtioMoleDryCo2$mean,
        "H2O_1_1_1" = dataList[[x]]$dp01$data$h2oTurb[[paste0(LvlTowr,"_30m")]]$rtioMoleDryH2o$mean,
        "qfFinlH2oTurbFrt00Samp" = dataList[[x]]$dp01$qfqm$h2oTurb[[paste0(LvlTowr,"_30m")]]$frt00Samp$
        "qfH2O_1_1_1" = dataList[[x]]$dp01$qfqm$h2oTurb[[paste0(LvlTowr,"_30m")]]$rtioMoleDryH2o$qfFinl
        "qfCO2_1_1_1" = dataList[[x]]$dp01$qfqm$co2Turb[[paste0(LvlTowr,"_30m")]]$rtioMoleDryCo2$qfFinl
        "qfSC" = dataList[[x]]$dp04$qfqm$fluxCo2$stor$qfFinl,
        "qfSLE" = dataList[[x]]$dp04$qfqm$fluxH2o$stor$qfFinl,
        "qfSH" = dataList[[x]]$dp04$qfqm$fluxTemp$stor$qfFinl,
        "qfT_SONIC" = dataList[[x]]$dp01$qfqm$soni[[paste0(LvlTowr,"_30m")]]$tempSoni$qfFinl,
        "qfWS_1_1_1" = dataList[[x]]$dp01$qfqm$soni[[paste0(LvlTowr,"_30m")]]$veloXaxsYaxsErth$qfFinl,
        rbind.data.frame(lapply(names(idxVerAmfx), function(y) {
          tryCatch({rlog$debug(y)}, error=function(cond){print(y)})
          rpt <- list()
          rpt[[paste0("CO2_1_",idxVerAmfx[y],"_2")]] <- dataList[[x]]$dp01$data$co2Stor[[paste0(y,"_30m


          rpt[[paste0("H2O_1_",idxVerAmfx[y],"_2")]] <- dataList[[x]]$dp01$data$h2oStor[[paste0(y,"_30m
          rpt[[paste0("CO2_1_",idxVerAmfx[y],"_3")]] <- dataList[[x]]$dp01$data$isoCo2[[paste0(y,"_30m"

          rpt[[paste0("H2O_1_",idxVerAmfx[y],"_3")]] <- dataList[[x]]$dp01$data$isoCo2[[paste0(y,"_30m"
          rpt[[paste0("qfCO2_1_",idxVerAmfx[y],"_2")]] <- dataList[[x]]$dp01$qfqm$co2Stor[[paste0(LvlTow
          rpt[[paste0("qfH2O_1_",idxVerAmfx[y],"_2")]] <- dataList[[x]]$dp01$qfqm$h2oStor[[paste0(LvlTow
          rpt[[paste0("qfCO2_1_",idxVerAmfx[y],"_3")]] <- dataList[[x]]$dp01$qfqm$isoCo2[[paste0(LvlTow
          rpt[[paste0("qfH2O_1_",idxVerAmfx[y],"_3")]] <- dataList[[x]]$dp01$qfqm$isoH2o[[paste0(LvlTow

          rpt <- rbind.data.frame(rpt)
          return(rpt)
        }
        )),


        "WS_1_1_1" = dataList[[x]]$dp01$data$soni[[paste0(LvlTowr,"_30m")]]$veloXaxsYaxsErth$mean,
        "WS_MAX_1_1_1" = dataList[[x]]$dp01$data$soni[[paste0(LvlTowr,"_30m")]]$veloXaxsYaxsErth$max,
        "WD_1_1_1" = dataList[[x]]$dp01$data$soni[[paste0(LvlTowr,"_30m")]]$angZaxsErth$mean,
        "T_SONIC" = dataList[[x]]$dp01$data$soni[[paste0(LvlTowr,"_30m")]]$tempSoni$mean,
        "T_SONIC_SIGMA" = base::sqrt(dataList[[x]]$dp01$data$soni[[paste0(LvlTowr,"_30m")]]$tempSoni$mea
        , stringsAsFactors = FALSE)
    })
```

```
## [1] "000_040"
## [1] "000_030"
## [1] "000_020"
## [1] "000_010"
```

```
## Warning in base::sqrt(dataList[[x]]$dp01$data$soni[[paste0(LvlTowr, "_30m")]]
## $tempSoni$mean): NaNs produced
```

```
## [1] "000_040"
## [1] "000_030"
## [1] "000_020"
## [1] "000_010"
## [1] "000_040"
## [1] "000_030"
## [1] "000_020"
## [1] "000_010"
## [1] "000_040"
## [1] "000_030"
## [1] "000_020"
## [1] "000_010"
## [1] "000_040"
## [1] "000_030"
## [1] "000_020"
## [1] "000_010"
## [1] "000_040"
## [1] "000_030"
## [1] "000_020"
## [1] "000_010"
## [1] "000_040"
## [1] "000_030"
## [1] "000_020"
## [1] "000_010"
## [1] "000_040"
## [1] "000_030"
## [1] "000_020"
## [1] "000_010"
## [1] "000_040"
## [1] "000_030"
## [1] "000_020"
## [1] "000_010"
```

```
## Warning in base::sqrt(dataList[[x]]$dp01$data$soni[[paste0(LvlTowr, "_30m")]]
## $tempSoni$mean): NaNs produced
```

```
## [1] "000_040"
## [1] "000_030"
## [1] "000_020"
## [1] "000_010"
```

```
## Warning in base::sqrt(dataList[[x]]$dp01$data$soni[[paste0(LvlTowr, "_30m")]]
## $tempSoni$mean): NaNs produced
```

```
## [1] "000_040"
## [1] "000_030"
## [1] "000_020"
## [1] "000_010"
```

```
## Warning in base::sqrt(dataList[[x]]$dp01$data$soni[[paste0(LvlTowr, "_30m")]]
## $tempSoni$mean): NaNs produced
```

```
## [1] "000_040"
## [1] "000_030"
## [1] "000_020"
## [1] "000_010"
```

```
## Warning in base::sqrt(dataList[[x]]$dp01$data$soni[[paste0(LvlTowr, "_30m")]]
## $tempSoni$mean): NaNs produced
```

```
    names(dataListFlux) <- names(dataList)
```

Combine the monthly data into a single dataframe, remove lists and clean memory

```
dataDfFlux <- do.call(rbind.data.frame,dataListFlux)
rm(list=c("dataListFlux","dataList"))
gc()
```

```
##           used (Mb) gc trigger  (Mb) limit (Mb)  max used  (Mb)
## Ncells 1342431 71.7    3637715 194.3         NA   3637715 194.3
## Vcells 9798141 74.8  105466192 804.7      16384 116616503 889.8
```

Regularize timeseries to 30 minutes in case timestamps are missing from NEON files due to processing errors

```
timeRglr <- eddy4R.base::def.rglr(timeMeas = as.POSIXlt(dataDfFlux$TIMESTAMP_START), dataMeas = dataDfF

#Reassign data to data.frame
dataDfFlux <- timeRglr$dataRglr
#Format timestamps
dataDfFlux$TIMESTAMP_START <- strftime(timeRglr$timeRglr + lubridate::hours(timeOfstUtc), format = "%Y%
dataDfFlux$TIMESTAMP_END <- strftime(timeRglr$timeRglr + lubridate::hours(timeOfstUtc) + lubridate::min
```

Define validation times, and remove this data from the dataset. At NEON sites, validations with a series of gasses of known concentration are run every 23.5 hours. These values are used to correct for measurment drift and are run every 23.5 hours to achive daily resolution while also spreading the impact of lost measurements throughout the day.

```
    #Remove co2Turb and h2oTurb data based off of qfFlow (qfFinl frt00)
    dataDfFlux$FC[(which(dataDfFlux$qfCO2_1_1_1 == 1))] <- NaN
    dataDfFlux$LE[(which(dataDfFlux$qfH2O_1_1_1 == 1))] <- NaN
    dataDfFlux$USTAR[(which(dataDfFlux$qfWS_1_1_1 == 1))] <- NaN
    dataDfFlux$H[(which(dataDfFlux$qfT_SONIC_1_1_1 == 1))] <- NaN
    dataDfFlux$SC[(which(dataDfFlux$qfSC == 1))] <- NaN
    dataDfFlux$SLE[(which(dataDfFlux$qfSLE == 1))] <- NaN
    dataDfFlux$SH[(which(dataDfFlux$qfSH == 1))] <- NaN
    dataDfFlux$T_SONIC[(which(dataDfFlux$qfT_SONIC_1_1_1 == 1))] <- NaN
    dataDfFlux$T_SONIC_SIGMA[(which(dataDfFlux$qfT_SONIC_1_1_1 == 1))] <- NaN
    dataDfFlux$WS_1_1_1[(which(dataDfFlux$qfWS_1_1_1 == 1))] <- NaN
    dataDfFlux$WS_MAX_1_1_1[(which(dataDfFlux$qfWS_1_1_1 == 1))] <- NaN
    dataDfFlux$WD_1_1_1[(which(dataDfFlux$qfWS_1_1_1 == 1))] <- NaN

    dataDfFlux$H2O_1_1_1[(which(dataDfFlux$qfH2O_1_1_1 == 1))] <- NaN
    dataDfFlux$CO2_1_1_1[(which(dataDfFlux$qfCO2_1_1_1 == 1))] <- NaN

    lapply(idxVerAmfx, function(x){
      #x <- 1
      dataDfFlux[[paste0("H2O_1_",x,"_2")]][(which(dataDfFlux[[paste0("qfH2O_1_",x,"_2")]] == 1))] <<-
      dataDfFlux[[paste0("H2O_1_",x,"_3")]][(which(dataDfFlux[[paste0("qfH2O_1_",x,"_3")]] == 1))] <<-
      dataDfFlux[[paste0("CO2_1_",x,"_2")]][(which(dataDfFlux[[paste0("qfCO2_1_",x,"_2")]] == 1))] <<-
      dataDfFlux[[paste0("CO2_1_",x,"_3")]][(which(dataDfFlux[[paste0("qfCO2_1_",x,"_3")]] == 1))] <<-
    })
```

```
## $`000_040`
## [1] NaN
##
## $`000_030`
## [1] NaN
##
## $`000_020`
## [1] NaN
##
## $`000_010`
## [1] NaN
```

Remove quality flagging variables from output

```
    setIdxQf <- grep("qf", names(dataDfFlux))
    dataDfFlux[,setIdxQf] <- NULL
```

Set range thresholds. Will get rid of outliers and have things conform to a reasonable range.

```
    #assign list
    Rng <- list()

    Rng$Min <- data.frame(
      "FC" = -100,            #[umol m-2 s-1]
      "SC" = -100,            #[umol m-2 s-1]
      "NEE" = -100,            #[umol m-2 s-1]
      "LE" = -500,            #[W m-2]
      "H" = -500,             #[W m-2]
      "USTAR" = 0,            #[m s-1]
      "CO2" = 200,            #[umol mol-1]
      "H2O" = 0,              #[mmol mol-1]
      "WS_1_1_1" = 0,         #[m s-1]
      "WS_MAX_1_1_1" = 0,     #[m s-1]
      "WD_1_1_1" = -0.1,      #[deg]
      "T_SONIC" = -55.0       #[C]
    )
```

Set Max thresholds

```
Rng$Max <- data.frame(
      "FC" = 100,           #[umol m-2 s-1]
      "SC" = 100,           #[umol m-2 s-1]
      "NEE" = 100,           #[umol m-2 s-1]
      "LE" = 1000,           #[W m-2]
      "H" = 1000,             #[W m-2]
      "USTAR" = 5,            #[m s-1]
      "CO2" = 800,           #[umol mol-1]
      "H2O" = 100,             #[mmol mol-1]
      "WS_1_1_1" = 50,         #[m s-1]
      "WS_MAX_1_1_1" = 50,     #[m s-1]
      "WD_1_1_1" = 360,     #[deg]
      "T_SONIC" = 45.0       #[C]
    )
```

Grab all CO2/H2O columns to apply same thresholds, replace missing values with -9999

```
    nameCO2 <- grep("CO2",names(dataDfFlux),value = TRUE)
    nameH2O <- grep("H2O",names(dataDfFlux),value = TRUE)
    #Apply the CO2/H2O threshold to all variables in HOR_VER_REP
    Rng$Min[nameCO2] <- Rng$Min$CO2
    Rng$Min[nameH2O] <- Rng$Min$H2O
    Rng$Max[nameCO2] <- Rng$Max$CO2
    Rng$Max[nameH2O] <- Rng$Max$H2O

    #Apply the range test to the output, and replace values with NaN
    lapply(names(dataDfFlux), function(x) {
      dataDfFlux[which(dataDfFlux[,x]<Rng$Min[[x]] | dataDfFlux[,x]>Rng$Max[[x]]),x] <<- NaN})
```

```
## [[1]]
## [1] NaN
##
## [[2]]
## [1] NaN
##
## [[3]]
## [1] NaN
##
## [[4]]
## [1] NaN
##
## [[5]]
## [1] NaN
##
## [[6]]
## [1] NaN
##
## [[7]]
## [1] NaN
##
## [[8]]
## [1] NaN
##
## [[9]]
## [1] NaN
##
## [[10]]
## [1] NaN
##
## [[11]]
## [1] NaN
##
## [[12]]
## [1] NaN
##
## [[13]]
## [1] NaN
##
## [[14]]
## [1] NaN
##
## [[15]]
## [1] NaN
##
## [[16]]
## [1] NaN
##
## [[17]]
## [1] NaN
##
## [[18]]
## [1] NaN
##
## [[19]]
## [1] NaN
##
## [[20]]
## [1] NaN
##
## [[21]]
## [1] NaN
##
## [[22]]
## [1] NaN
##
## [[23]]
## [1] NaN
##
## [[24]]
## [1] NaN
##
## [[25]]
## [1] NaN
##
## [[26]]
## [1] NaN
##
## [[27]]
## [1] NaN
##
## [[28]]
## [1] NaN
##
## [[29]]
## [1] NaN
##
## [[30]]
## [1] NaN
##
## [[31]]
## [1] NaN
##
## [[32]]
## [1] NaN
##
## [[33]]
## [1] NaN
##
## [[34]]
## [1] NaN
##
## [[35]]
## [1] NaN
##
## [[36]]
## [1] NaN
```

```
    # Delete any NEE that have either FC or SC removed
    dataDfFlux[is.na(dataDfFlux$FC) | is.na(dataDfFlux$SC),"NEE"] <- NaN

    #Change NA to -9999
    dataDfFlux[is.na(dataDfFlux)] <- -9999
```

Write output data to csv

```
# Create output filename based off of Ameriflux file naming convention
nameFileOut <- base::paste0(DirOut,"/",siteNeon$SITE_ID,'_HH_',dataDfFlux$TIMESTAMP_START[1],'_',utils:

# Write output to .csv
write.csv(x = dataDfFlux, file = nameFileOut, row.names = FALSE)
```

Combine Ameriflux and NEON data

```
# Format date
dataDfFlux <- transform(dataDfFlux, TIMESTAMP_START = as.POSIXct(as.character(TIMESTAMP_START), format=
dataDfFlux <- transform(dataDfFlux, TIMESTAMP_END = as.POSIXct(as.character(TIMESTAMP_END), format="%Y%
```

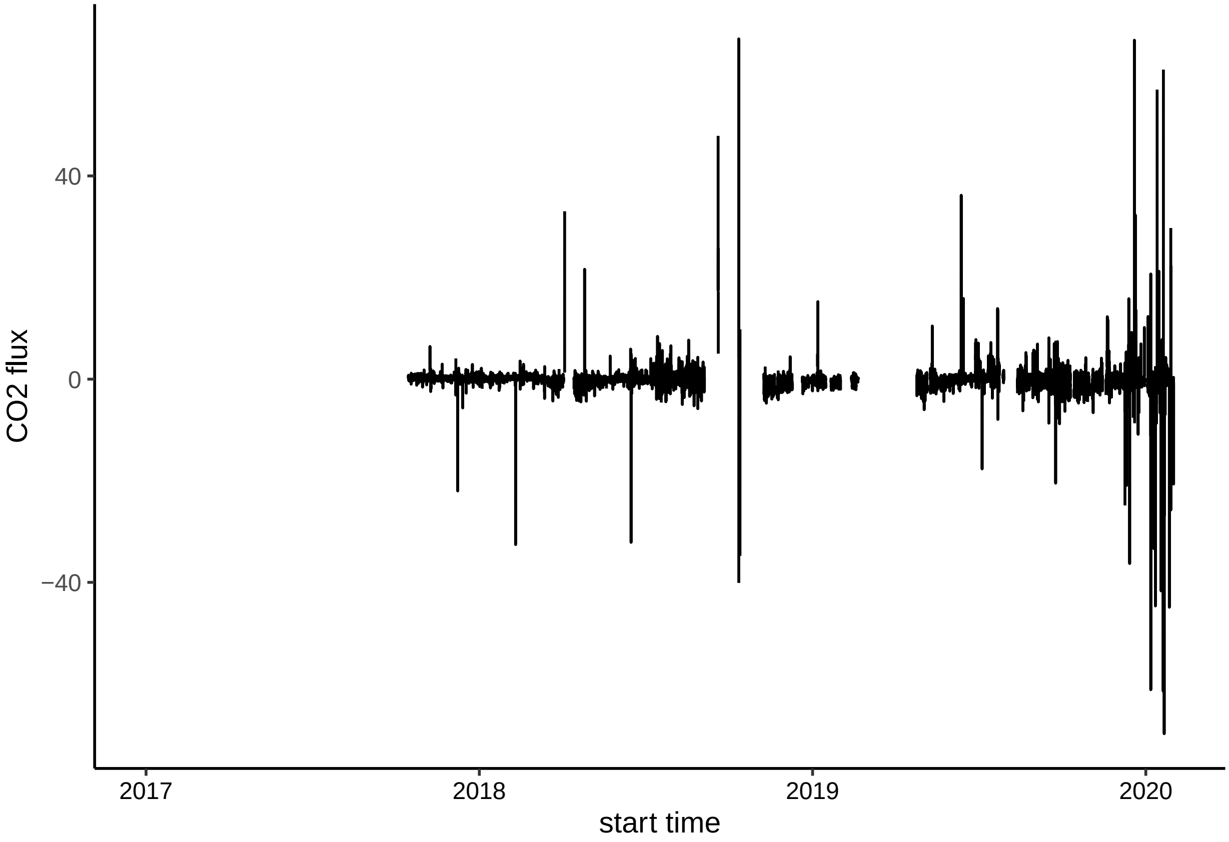
```
# Join
xSR_plus <- full_join(xSR, dataDfFlux)
```

```
## Joining, by = c("TIMESTAMP_START", "TIMESTAMP_END", "FC", "SC", "LE", "SLE", "USTAR", "H", "SH", "FE
# Turn -9999 into NAs for graph
xSR_plus[xSR_plus == -9999] = NA
```

Plot the timeseries

```
ggplot(xSR_plus) +
  geom_line(aes(x = TIMESTAMP_START, y = FC)) +
  labs(y = "CO2 flux", x = "start time") +
    theme(panel.background = element_rect(fill="white"),
          axis.line = element_line(color = "black"),
          axis.text.x = element_text(colour="black"))
```

```
## Warning: Removed 13811 row(s) containing missing values (geom_path).
```



There are a lot of gaps in the data, especially in 2017. The variability also increase a lot in 2020, which makes me concerned about the data quality for that year.

2. Using metScanR package, find co-located NEON and AmeriFlux sites. Download data for an overlapping time period, and compare FC and H values by making a scatter plot and seeing how far off the data are from a 1:1 line.

```
library("metScanR")
```

```
## Warning: package 'metScanR' was built under R version 4.0.2

## Welcome to metScanR!
## This package takes a few extra seconds to load because it doesn't use lazyload for the database.  Tha
## To access the full and most up-to-date database, please run the updateDatabase() function.
```

```r
(sf <- siteFinder(country = "United States", siteID = "NEON:SRER", radius = 50))
```

```
## $`US-Aud`
## $`US-Aud`$namez
## [1] "Audubon Research Ranch"
##
## $`US-Aud`$identifiers
##      idType     id
## 1 AMERIFLUX US-Aud
##
## $`US-Aud`$platform
## [1] "AmeriFlux"
##
## $`US-Aud`$elements
##   element date.begin date.end
## 1      NA         NA       NA
##
## $`US-Aud`$location
##   latitude_dec longitude_dec elev       country state county utcoffset
## 1      31.5907     -110.5104 1469 UNITED STATES    NA     NA        NA
##   date.begin date.end
## 1       2002  present
##
##
## $`US-SRC`
## $`US-SRC`$namez
## [1] "Santa Rita Creosote"
##
## $`US-SRC`$identifiers
##      idType     id
## 1 AMERIFLUX US-SRC
##
## $`US-SRC`$platform
## [1] "AmeriFlux"
##
## $`US-SRC`$elements
##   element date.begin date.end
## 1      NA         NA       NA
##
## $`US-SRC`$location
##   latitude_dec longitude_dec elev       country state county utcoffset
## 1      31.9083     -110.8395  991 UNITED STATES    NA     NA        NA
##   date.begin date.end
## 1       2008  present
##
##
## $`US-SRG`
## $`US-SRG`$namez
## [1] "Santa Rita Grassland"
##
## $`US-SRG`$identifiers
##      idType     id
## 1 AMERIFLUX US-SRG
##
```

```
## $`US-SRG`$platform
## [1] "AmeriFlux"
## 
## $`US-SRG`$elements
##   element date.begin date.end
## 1      NA         NA       NA
## 
## $`US-SRG`$location
##   latitude_dec longitude_dec elev       country state county utcoffset
## 1      31.7894     -110.8277 1291 UNITED STATES    NA     NA        NA
##   date.begin date.end
## 1       2008  present
## 
## 
## $`US-SRM`
## $`US-SRM`$namez
## [1] "Santa Rita Mesquite"
## 
## $`US-SRM`$identifiers
##      idType     id
## 1 AMERIFLUX US-SRM
## 
## $`US-SRM`$platform
## [1] "AmeriFlux"
## 
## $`US-SRM`$elements
##   element date.begin date.end
## 1      NA         NA       NA
## 
## $`US-SRM`$location
##   latitude_dec longitude_dec elev       country state county utcoffset
## 1      31.8214     -110.8661 1120 UNITED STATES    NA     NA        NA
##   date.begin date.end
## 1       2004  present
## 
## 
## $SRER
## $SRER$namez
## [1] "Santa Rita Experimental Range Site"
## 
## $SRER$identifiers
##   idType   id
## 1   NEON SRER
## 
## $SRER$platform
## [1] "NEON"
## 
## $SRER$elements
##          element date.begin date.end
## 1  DP1.10066.001    2013-11  2013-11
## 2  DP1.10023.001    2016-04  2018-09
## 3  DP1.30006.001    2017-08  2017-08
## 4  DP2.30020.001    2017-08  2017-08
## 5  DP1.10058.001    2016-01  2018-04
## 6  DP1.00096.001    2013-11  2013-11
## 7  DP1.00002.001    2017-02  2018-01
## 8  DP1.00001.001    2017-02  2018-11
## 9  DP3.30019.001    2017-08  2017-08
## 10 DP2.30011.001    2017-08  2017-08
## 11 DP1.00097.001    2013-11  2013-11
## 12 DP1.00003.001    2017-02  2017-05
## 13 DP3.30026.001    2017-08  2017-08
## 14 DP3.30010.001    2017-08  2017-08
## 15 DP1.10033.001    2016-05  2018-10
## 16 DP1.00098.001    2017-02  2018-11
## 17 DP2.30012.001    2017-08  2017-08
## 18 DP1.10076.001    2016-01  2018-05
## 19 DP3.30018.001    2017-08  2017-08
## 20 DP1.10093.001    2016-04  2018-09
## 21 DP1.10017.001    2016-04  2018-10
## 22 DP1.00042.001    2017-02  present
## 23 DP1.30003.001    2017-08  2017-08
## 24 DP3.30024.001    2017-08  2017-08
## 25 DP1.10020.001    2016-02  2016-11
## 26 DP3.30011.001    2017-08  2017-08
## 27 DP1.00043.001    2017-02  present
## 28 DP1.10003.001    2017-05  2018-05
## 29 DP2.30026.001    2017-08  2017-08
## 30 DP1.00038.001    2018-10  2018-11
## 31 DP1.10078.001    2016-07  2016-07
## 32 DP3.30012.001    2017-08  2017-08
## 33 DP1.10022.001    2016-04  2018-07
## 34 DP1.00014.001    2017-02  2018-01
## 35 DP3.30025.001    2017-08  2017-08
## 36 DP1.00013.001    2018-10  2018-10
## 37 DP2.30014.001    2017-08  2017-08
## 38 DP3.30022.001    2017-08  2017-08
## 39 DP1.00066.001    2017-02  2018-01
## 40 DP1.00040.001    2017-02  2018-11
## 41 DP1.30001.001    2017-08  2017-08
## 42 DP1.00006.001    2017-02  2018-01
## 43 DP1.00023.001    2017-02  2018-01
## 44 DP3.30006.001    2017-08  2017-08
## 45 DP1.10100.001    2016-07  2016-07
## 46 DP1.00041.001    2017-02  2018-01
## 47 DP1.10045.001    2016-01  2017-01
## 48 DP3.30014.001    2017-08  2017-08
## 49 DP2.30016.001    2017-08  2017-08
## 50 DP1.00033.001    2017-02  present
## 51 DP1.00024.001    2017-02  2017-03
## 52 DP1.30010.001    2017-08  2017-08
## 53 DP4.00200.001    2017-09  2018-02
## 54 DP1.00004.001    2017-02  2018-01
## 55 DP2.30022.001    2017-08  2017-08
## 56 DP3.30016.001    2017-08  2017-08
## 57 DP2.30019.001    2017-08  2017-08
## 58 DP1.10098.001    2016-01  2017-01
## 59 DP1.10072.001    2016-05  2018-10
## 60 DP1.00094.001    2017-07  2018-11
## 61 DP1.10038.001    2016-01  2017-09
## 62 DP1.10055.001    2016-06  2018-11
## 63 DP1.00022.001    2017-02  2018-01
## 64 DP1.10043.001    2016-04  2018-11
## 65 DP3.30015.001    2017-08  2017-08
## 66 DP1.00005.001    2017-02  2018-01
## 67 DP1.10086.001    2016-06  2018-08
## 68 DP3.30020.001    2017-08  2017-08
## 69 DP2.30018.001    2017-08  2017-08
## 70 DP1.30008.001    2017-08  2017-08
##
## $SRER$location
##   latitude_dec longitude_dec elev       country state county utcoffset
## 1     31.91068     -110.8355  983 UNITED STATES    AZ     NA        NA
##   date.begin date.end
## 1    2013-11  present
```

```
(sf_map <- mapResults(sf))
```

Seems like there are 3 other towers in the Santa Rita area, US-SRC, US-SRG, and US-SRM. US-Aud is also nearby the NEON SRER site, but no longer collecting data. I think I will download the data for US-SRM to compare to the SRER site.

```
# Read in US-SRM
SRM <- read.csv("./SRM/AMF_US-SRM_BASE-BADM_20-5/AMF_US-SRM_BASE_HH_20-5.csv", header = T, skip = 2)
```

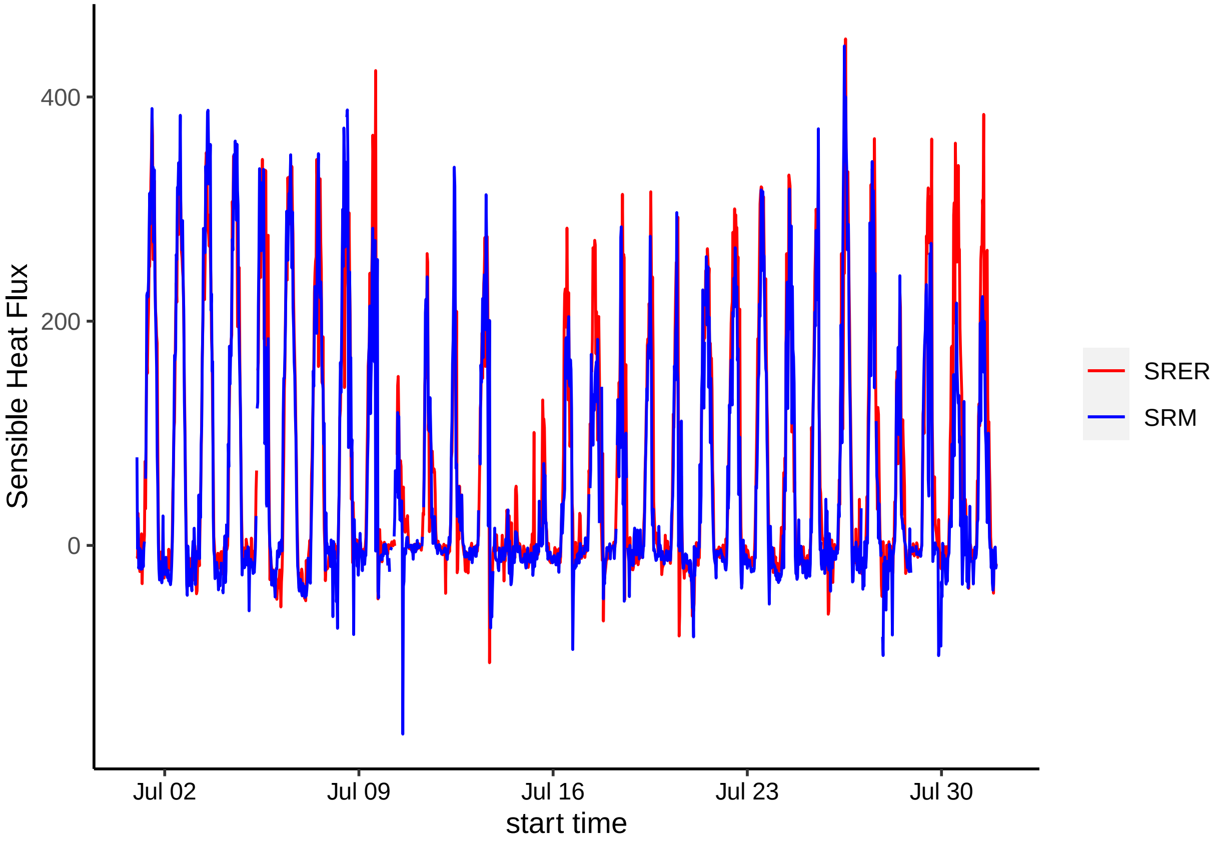
```
# Format date
SRM <- transform(SRM, TIMESTAMP_START = as.POSIXct(as.character(TIMESTAMP_START), format="%Y%m%d%H%M"))
SRM <- transform(SRM, TIMESTAMP_END = as.POSIXct(as.character(TIMESTAMP_END), format="%Y%m%d%H%M"))
```

```
# Turn -9999 into NAs for graph
SRM[SRM == -9999] = NA
```

I'll plot SRM and SRER over July 2018, since that's one of the months they overlap. Plus, that's monsoon season.

```
SRM_2018 <- SRM %>%
  mutate(year = as.numeric(format(TIMESTAMP_START, format = "%Y"))) %>%
  mutate(month = as.numeric(format(TIMESTAMP_START, format = "%m"))) %>%
  filter(year == 2018) %>%
  filter(month == 7)

xSR_plus_2018 <- xSR_plus %>%
  mutate(year = as.numeric(format(TIMESTAMP_START, format = "%Y"))) %>%
  mutate(month = as.numeric(format(TIMESTAMP_START, format = "%m"))) %>%
  filter(year == 2018) %>%
  filter(month == 7)
```

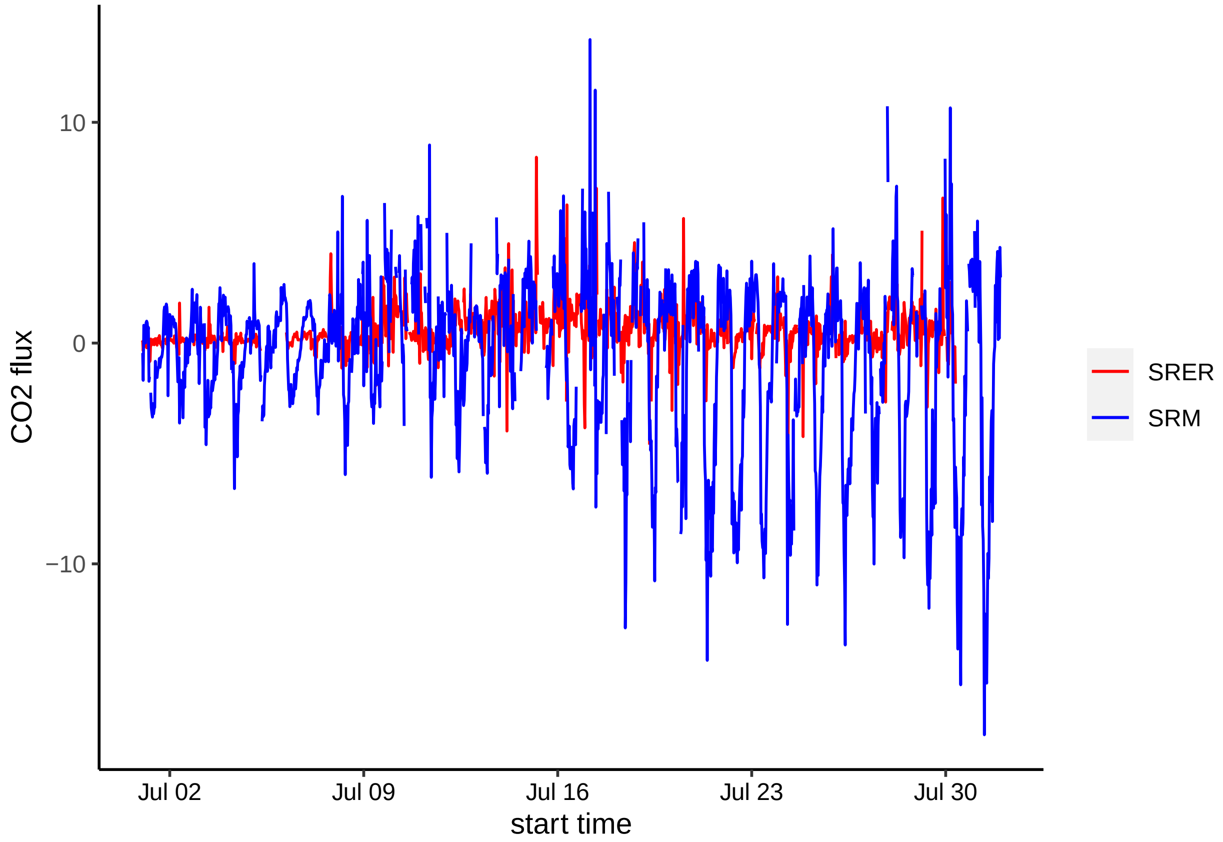
```
# Plot H
ggplot() +
  geom_line(data = xSR_plus_2018, aes(x = TIMESTAMP_START, y = H, color = "SRER")) +
  geom_line(data = SRM_2018, aes(x = TIMESTAMP_START, y = H, color = "SRM")) +
  labs(y = "Sensible Heat Flux", x = "start time") +
  scale_color_manual(name = NULL, values=c("red","blue")) +
    theme(panel.background = element_rect(fill="white"),
          axis.line = element_line(color = "black"),
          axis.text.x = element_text(colour="black"))
```



These are really close values, which makes sense since these towers are so close to each other.

```
# Plot FC
ggplot() +
  geom_line(data = xSR_plus_2018, aes(x = TIMESTAMP_START, y = FC, color = "SRER")) +
  geom_line(data = SRM_2018, aes(x = TIMESTAMP_START, y = FC, color = "SRM")) +
  labs(y = "CO2 flux", x = "start time") +
  scale_color_manual(name = NULL, values=c("red","blue")) +
    theme(panel.background = element_rect(fill="white"),
          axis.line = element_line(color = "black"),
          axis.text.x = element_text(colour="black"))
```

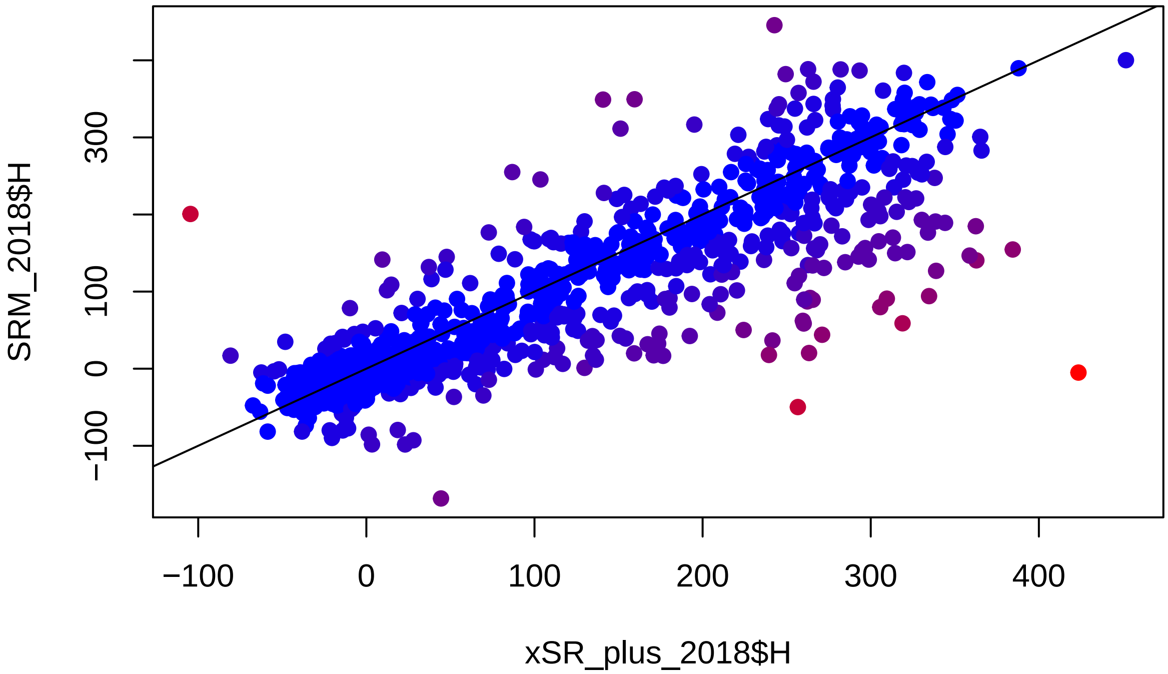
```
## Warning: Removed 78 row(s) containing missing values (geom_path).

## Warning: Removed 1 row(s) containing missing values (geom_path).
```



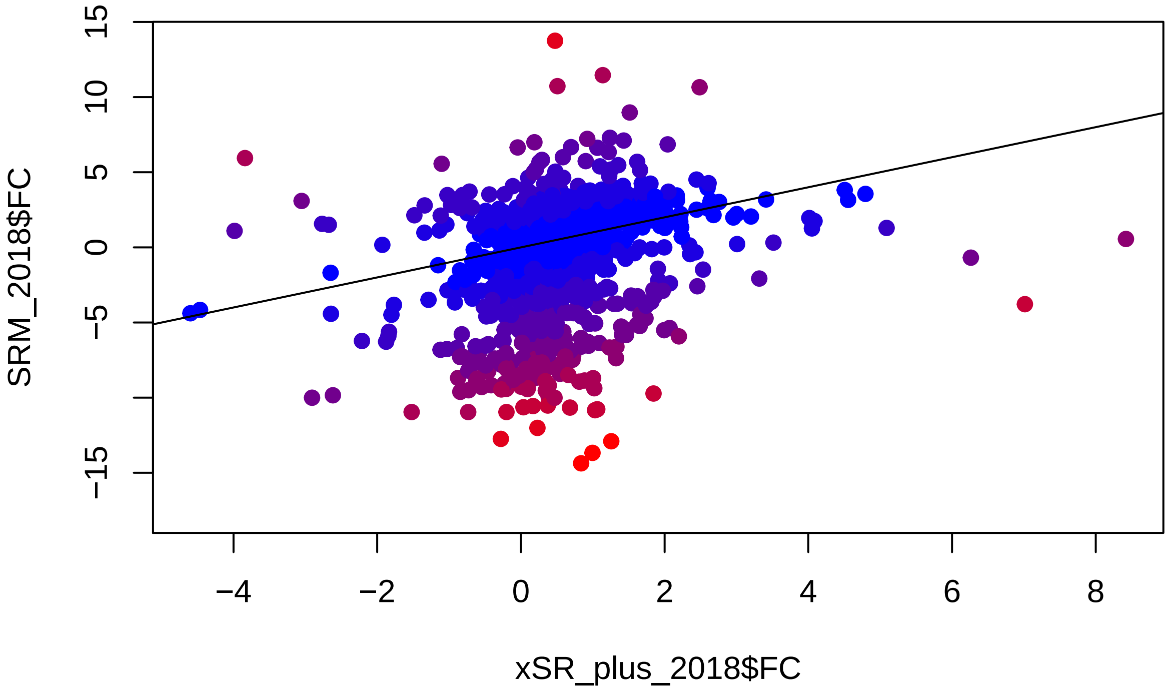
Wow! The CO2 flux is really different. The plants at SRM must be photosynthesizing more, especially later in the month. The SRM site observes encroaching mesquite shrubs, and I wonder if this means there is more vegetation that causes such negative CO2 flux numbers.

Now I'll make scatterplots to see how far off the data are from 1:1 lines.

```
# H scatter
deviationH <- abs(xSR_plus_2018$H-SRM_2018$H)
rbPalH <- colorRampPalette(c('blue','red'))
data_colH=rbPalH(10)[as.numeric(cut(deviationH,breaks = 10))]
dfH <- cbind(xSR_plus_2018$H,SRM_2018$H,deviationH)
plot(xSR_plus_2018$H, SRM_2018$H, pch = 21, col=data_colH, bg=data_colH)
abline(0, 1)
```



```
# FC scatter
deviationFC <- abs(xSR_plus_2018$FC-SRM_2018$FC)
rbPalFC <- colorRampPalette(c('blue','red'))
data_colFC=rbPalFC(10)[as.numeric(cut(deviationFC,breaks = 10))]
dfFC <- cbind(xSR_plus_2018$FC,SRM_2018$FC,deviationFC)
plot(xSR_plus_2018$FC, SRM_2018$FC, pch = 21, col=data_colFC, bg=data_colFC)
abline(0, 1)
```



The scatterplots agree with what the timeseries suggests, in that the heat flux is more similar between the two nearby sites than the CO2 flux.