INF 550 Section 5.7

Natasha Wesely

2022-10-19

Flux Measurements & Inter-Operability Coding Exercise

Question 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.

```
# #Install most  recent version of ffbase from GitHub, this is a package dependency of eddy4R.base
# devtools::install_github("edwindj/ffbase", subdir="pkg")
#
# #Install NEONprocIS.base from GitHub, this package is a dependency of eddy4R.base
# devtools::install_github(repo="NEONScience/NEON-IS-data-processing",
#                          ref="master",
#                          subdir="pack/NEONprocIS.base",
#                          dependencies=c(NA, TRUE)[2],
#                          repos=c(BiocManager::repositories(),   # for dependencies on Bioconductor pa
#                                  "https://cran.rstudio.com/")       # for CRAN
# )
#
# #Install eddy4R.base from GitHub
# devtools::install_github(repo="NEONScience/eddy4R",
#                          ref="master",
#                          subdir="pack/eddy4R.base",
#                          dependencies=c(NA, TRUE)[2],
#                          repos=c(BiocManager::repositories(),   # for dependencies on Bioconductor pa
#                                  "https://cran.rstudio.com/")       # for CRAN
# )

# # make a list of the required packages
# packReq <- c("rhdf5", "eddy4R.base", "jsonlite", "lubridate")
#
# # load the required packages
# lapply(packReq, function(x) {
#   print(x)
#   if(require(x, character.only = TRUE) == FALSE) {
#     install.packages(x)
#     library(x, character.only = TRUE)
#   }})
```

```
# disable file locking
h5disableFileLocking()

# choose your NEON site of interest
site <- "JORN"

#define start and end dates, optional, defaults to entire period of site operation. Use %Y-%m-%d format
dateBgn <- "2021-01-01"
dateEnd <- "2021-12-31"

# 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"))


# Specify Download directory for HDF5 files from the NEON data portal and output directory to save the
#download directory
DirDnld=tempdir()

#Output directory, change this to where you want to save the output csv
DirOutBase <-paste0("~/data_5.7/eddy/data/Ameriflux/",ver)

# Specify Data Product number, for the Bundled Eddy-Covariance files, this is DP4.00200.001
#DP number
dpID <- 'DP4.00200.001'


# Get metadata from Ameriflux Site Info BADM sheets for the site of interest

#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/AmeriF

#Grab only your neon site of interest NEON site from the AmeriFlux sites
siteNeon <- sites_web[grep(pattern = paste0("NEON.*",site), x = sites_web$SITE_NAME),]

# get the AmeriFlux site info for your site of interest
metaSite <- lapply(siteNeon$SITE_ID, function(x) {
  pathSite <- paste0("http://ameriflux-data.lbl.gov/BADM/Anc/SiteInfo/",x)
  tmp <- fromJSON(pathSite)
  return(tmp)
})


# Use Ameriflux site IDs to name metadata lists
#use NEON ID as list name
names(metaSite) <- site

# Check if dateBgn is defined, if not make it the initial operations date "IOCR" of the site
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 the UTC time offset from the Ameriflux API
timeOfstUtc <- as.integer(metaSite[[site]]$values$GRP_UTC_OFFSET[[1]]$UTC_OFFSET)

# Create the 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 ", datel
print(msg)
```

```
## [1] "Starting Ameriflux FP standard conversion processing workflow for JORN for 2021-01-01 to 2021-1
```

```
# 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)

# Download and extract data

#Initialize data List
dataList <- list()

#Read data from the API
dataList <- lapply(setDate, function(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, pacl
  files <- list.files(paste0(DirDnld, "/filesToStack00200"))
  utils::unzip(paste0(DirDnld, "/filesToStack00200/", files[grep(pattern = paste0(site,".*.", date, ".*
  files <- list.files(paste0(DirDnld, "/filesToStack00200"))
  R.utils::gunzip(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 =

  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 85.171321 MB
## Downloading 1 files
##   |                                                                      |
## 1 files successfully downloaded to /var/folders/78/lkktz4ls31510kyxfwpqst340000gn/T//RtmpgivhZQ/files
## Finding available files
##   |                                                                      |
##
## Downloading files totaling approximately 72.582647 MB

## /var/folders/78/lkktz4ls31510kyxfwpqst340000gn/T//RtmpgivhZQ/filesToStack00200 already exists. Downlo

## Downloading 1 files
##   |                                                                      |
## 1 files successfully downloaded to /var/folders/78/lkktz4ls31510kyxfwpqst340000gn/T//RtmpgivhZQ/files
## Finding available files
##   |                                                                      |
##
## Downloading files totaling approximately 89.315757 MB

## /var/folders/78/lkktz4ls31510kyxfwpqst340000gn/T//RtmpgivhZQ/filesToStack00200 already exists. Downlo

## Downloading 1 files
##   |                                                                      |
## 1 files successfully downloaded to /var/folders/78/lkktz4ls31510kyxfwpqst340000gn/T//RtmpgivhZQ/files
## Finding available files
##   |                                                                      |
##
## Downloading files totaling approximately 86.457409 MB

## /var/folders/78/lkktz4ls31510kyxfwpqst340000gn/T//RtmpgivhZQ/filesToStack00200 already exists. Downlo

## Downloading 1 files
##   |                                                                      |
## 1 files successfully downloaded to /var/folders/78/lkktz4ls31510kyxfwpqst340000gn/T//RtmpgivhZQ/files
## Finding available files
##   |                                                                      |
##
## Downloading files totaling approximately 86.017641 MB
```

```
## /var/folders/78/lkktz4ls31510kyxfwpqst340000gn/T//RtmpgivhZQ/filesToStack00200 already exists. Downlo

## Downloading 1 files
##   |                                                                                  |
## 1 files successfully downloaded to /var/folders/78/lkktz4ls31510kyxfwpqst340000gn/T//RtmpgivhZQ/files
## Finding available files
##   |                                                                                  |
##
## Downloading files totaling approximately 77.874858 MB

## /var/folders/78/lkktz4ls31510kyxfwpqst340000gn/T//RtmpgivhZQ/filesToStack00200 already exists. Downlo

## Downloading 1 files
##   |                                                                                  |
## 1 files successfully downloaded to /var/folders/78/lkktz4ls31510kyxfwpqst340000gn/T//RtmpgivhZQ/files
## Finding available files
##   |                                                                                  |
##
## Downloading files totaling approximately 76.290948 MB

## /var/folders/78/lkktz4ls31510kyxfwpqst340000gn/T//RtmpgivhZQ/filesToStack00200 already exists. Downlo

## Downloading 1 files
##   |                                                                                  |
## 1 files successfully downloaded to /var/folders/78/lkktz4ls31510kyxfwpqst340000gn/T//RtmpgivhZQ/files
## Finding available files
##   |                                                                                  |
##
## Downloading files totaling approximately 90.980619 MB

## /var/folders/78/lkktz4ls31510kyxfwpqst340000gn/T//RtmpgivhZQ/filesToStack00200 already exists. Downlo

## Downloading 1 files
##   |                                                                                  |
## 1 files successfully downloaded to /var/folders/78/lkktz4ls31510kyxfwpqst340000gn/T//RtmpgivhZQ/files
## Finding available files
##   |                                                                                  |
##
## Downloading files totaling approximately 87.734152 MB

## /var/folders/78/lkktz4ls31510kyxfwpqst340000gn/T//RtmpgivhZQ/filesToStack00200 already exists. Downlo

## Downloading 1 files
##   |                                                                                  |
## 1 files successfully downloaded to /var/folders/78/lkktz4ls31510kyxfwpqst340000gn/T//RtmpgivhZQ/files
## Finding available files
##   |                                                                                  |
##
## Downloading files totaling approximately 89.210842 MB

## /var/folders/78/lkktz4ls31510kyxfwpqst340000gn/T//RtmpgivhZQ/filesToStack00200 already exists. Downlo
```

```
## Downloading 1 files
##   |                                                                              |
## 1 files successfully downloaded to /var/folders/78/lkktz4ls31510kyxfwpqst340000gn/T//RtmpgivhZQ/files
## Finding available files
##   |                                                                              |
##
## Downloading files totaling approximately 83.70935 MB

## /var/folders/78/lkktz4ls31510kyxfwpqst340000gn/T//RtmpgivhZQ/filesToStack00200 already exists. Downlo

## Downloading 1 files
##   |                                                                              |
## 1 files successfully downloaded to /var/folders/78/lkktz4ls31510kyxfwpqst340000gn/T//RtmpgivhZQ/files
## Finding available files
##   |                                                                              |
##
## Downloading files totaling approximately 92.91686 MB

## /var/folders/78/lkktz4ls31510kyxfwpqst340000gn/T//RtmpgivhZQ/filesToStack00200 already exists. Downlo

## Downloading 1 files
##   |                                                                              |
## 1 files successfully downloaded to /var/folders/78/lkktz4ls31510kyxfwpqst340000gn/T//RtmpgivhZQ/files
```

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

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

# Determine tower horizontal & vertical indices

#Find the tower top level by looking at the vertical index of the turbulent CO2 concentration measureme
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])

#Ameriflux vertical levels based off of https://ameriflux.lbl.gov/data/aboutdata/data-variables/ sectio
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
```

```r
dataListFlux <- lapply(names(dataList), function(x) {
  data.frame(
    "TIMESTAMP_START" = as.POSIXlt(dataList[[x]]$dp04$data$fluxCo2$turb$timeBgn, format="%Y-%m-%dT%H:%M
    "TIMESTAMP_END" = as.POSIXlt(dataList[[x]]$dp04$data$fluxCo2$turb$timeEnd, format="%Y-%m-%dT%H:%M:%
    # "TIMESTAMP_START" = strftime(as.POSIXlt(dataList[[x]][[idxSite]]$dp04$data$fluxCo2$turb$timeBgn, 
    # "TIMESTAMP_END" = strftime(as.POSIXlt(dataList[[x]][[idxSite]]$dp04$data$fluxCo2$turb$timeEnd, fo
    "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$qfFi
    "qfH2O_1_1_1" = dataList[[x]]$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$isoCo2[[paste0(y,"_30m")]]$i

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

      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,
```

```r
    "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")]]$tempSoni$mean)
    , 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"

## 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"
## [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
```

```
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 1232056 65.8    2627377 140.4         NA   2627377 140.4
## Vcells 3488353 26.7  115474521 881.1      16384 122752273 936.6
```

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

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::minu

# Define validation times, and remove this data from the dataset. At NEON sites, validations with a ser

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

```
## $`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

#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]
)
```

```r
# 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)

# Clean up environment
rm(list="dataDfFlux")
gc()
```

```
##           used (Mb) gc trigger  (Mb) limit (Mb)  max used  (Mb)
## Ncells 1225520 65.5    2627377 140.4         NA   2627377 140.4
## Vcells 3497997 26.7   92379617 704.9      16384 122752273 936.6
```

```
# get the ameriflux data from the true amerifulx site
```

```r
# Load dplyr package
library(dplyr)
library(amerifluxr)
floc2 <- amf_download_base(user_id = "nkw54",
                           user_email = "nkw54@nau.edu",
                           site_id = "US-xJR",
                           data_product = "BASE-BADM",
                           data_policy = "CCBY4.0",
                           agree_policy = TRUE,
                           intended_use = "remote_sensing",
                           intended_use_text = "class project",
                           verbose = TRUE,
                           out_dir = tempdir())
```

```
## Data use guidelines for AmeriFlux CC-BY-4.0 Data Policy:
##
## (1) Data user is free to Share (copy and redistribute the material in any medium or format) and/or Ad
## (2) Provide a citation to each site data product that includes the data-product DOI and/or recommende
## (3) Acknowledge funding for supporting AmeriFlux data portal: U.S. Department of Energy Office of Sci
```

```r
base <- amf_read_base(file = floc2,
                      unzip = TRUE,
                      parse_timestamp = TRUE)

data2021 <- base[base$YEAR == 2021 & base$MONTH == 5, ]

head(data2021)
```

```
##       YEAR MONTH DAY DOY HOUR MINUTE           TIMESTAMP TIMESTAMP_START
## 75889 2021     5   1 121    0     15 2021-05-01 00:15:00     2.02105e+11
## 75890 2021     5   1 121    0     45 2021-05-01 00:45:00     2.02105e+11
## 75891 2021     5   1 121    1     15 2021-05-01 01:15:00     2.02105e+11
## 75892 2021     5   1 121    1     45 2021-05-01 01:45:00     2.02105e+11
## 75893 2021     5   1 121    2     15 2021-05-01 02:15:00     2.02105e+11
## 75894 2021     5   1 121    2     45 2021-05-01 02:45:00     2.02105e+11
##       TIMESTAMP_END CO2_1_1_1 H2O_1_1_1 CO2_1_1_2 H2O_1_1_2 CO2_1_1_3 H2O_1_1_3
## 75889   2.02105e+11  420.3761  11.17238  395.9738  9.652658        NA        NA
## 75890   2.02105e+11  420.0195  10.91418  395.3210  9.213329  422.0955        NA
## 75891   2.02105e+11  419.7776  10.88880  395.1443  9.015875  421.6744        NA
## 75892   2.02105e+11  419.7486  11.06355  395.3198  9.365982  421.5739        NA
## 75893   2.02105e+11  419.2382  10.76206  394.6034  9.238141        NA        NA
## 75894   2.02105e+11  419.1735  10.22623  394.4064  8.654653  421.2731        NA
##       CO2_1_2_2 H2O_1_2_2 CO2_1_2_3 H2O_1_2_3 CO2_1_3_2 H2O_1_3_2 CO2_1_3_3
## 75889  395.7037  9.111137        NA        NA  395.3578  8.916285        NA
## 75890  395.5069  8.925351  421.6280        NA  395.6175  8.882506  421.4200
## 75891  395.0799  9.001226  420.9821        NA  394.6605  8.819896        NA
## 75892  394.8194  8.966944        NA        NA  394.5646  8.709956  420.1803
## 75893  393.8560  8.678887        NA        NA  393.5724  8.417371        NA
## 75894  394.7602  8.477029  420.9914        NA  394.7315  8.320583  420.9077
##       H2O_1_3_3 CO2_1_4_2 H2O_1_4_2 CO2_1_4_3 H2O_1_4_3 WS_1_1_1 WS_MAX_1_1_1
## 75889        NA  394.8722  8.927136        NA        NA 3.649881     4.605743
## 75890        NA  395.2014  8.723159        NA        NA 2.762708     4.048008
## 75891        NA  393.6416  8.341259  420.7709        NA 2.319437     3.187362
```

```
## 75892        NA  393.6591  8.341316  419.9926        NA 2.705332     3.222703
## 75893        NA  393.6926  8.490646        NA        NA 2.016367     3.190156
## 75894        NA  394.5171  8.229566        NA        NA 2.390496     4.329449
##       WD_1_1_1   T_SONIC T_SONIC_SIGMA          FC          SC     NEE_PI
## 75889 252.5552 13.258978     0.5709251 -0.08884515          NA         NA
## 75890 246.6403 12.284707     0.9079627 -0.10992877          NA         NA
## 75891 213.4337 11.901863     0.9500304          NA -0.09589228         NA
## 75892 212.6684 12.066722     0.6616876          NA -0.01032604         NA
## 75893 200.3322 10.785722     1.1698547 -0.05929778 -0.04353413 -0.1028319
## 75894 125.8395  8.379748     1.0588261          NA  0.02555247         NA
##               LE        SLE      USTAR         H          SH FETCH_90 FETCH_MAX
## 75889 -0.4709535         NA 0.17294065 -3.566694  -0.5544341   733.92    283.56
## 75890 -2.2608176         NA 0.08481327        NA  -3.9375819   425.34    166.80
## 75891         NA -0.8393316         NA        NA  -2.8053657   800.64    308.58
## 75892  0.8135093  1.5991140 0.04332163  3.385802   4.0528930   800.64    308.58
## 75893         NA -3.6744678 0.07181803        NA -10.2395447   300.24    125.10
## 75894 -3.7262652 -1.5555782 0.10543657 -8.051801  -7.4650167   308.58    125.10
##         V_SIGMA WS_1_2_1 WS_1_3_1 WS_1_4_1 WS_MAX_1_2_1 WS_MAX_1_3_1
## 75889 0.3780564     2.18     1.36     0.48         3.19         2.33
## 75890 0.4256429     1.35     0.77     0.25         2.86         1.94
## 75891 0.3026909     1.38     0.85     0.16         2.30         1.82
## 75892 0.4742750     1.60     0.92     0.18         2.18         1.44
## 75893 0.5328588     1.20     0.72     0.13         2.07         1.19
## 75894 0.7912087     1.77     1.10     0.30         3.22         2.25
##       WS_MAX_1_4_1 WD_1_2_1 WD_1_3_1 WD_1_4_1 TA_1_1_1 TA_1_1_2 TA_1_2_1
## 75889         1.50   254.24   247.63   253.23  12.3874   12.302  10.3130
## 75890         0.98   247.46   236.48   210.72  11.4890   11.563   9.7122
## 75891         0.77   212.62   205.29   184.31  11.2632   11.078   9.8341
## 75892         0.60   218.86   220.91   221.53  11.3454   10.907   9.3315
## 75893         0.43   177.25   164.69   143.77  10.2826   10.472   8.8118
## 75894         1.16   124.43   122.11   117.14   7.9371    7.888   7.2675
##       TA_1_3_1 TA_1_4_1       PA    RH VPD_PI T_CANOPY_1_1_1 T_CANOPY_1_2_1
## 75889   9.6111   9.1850 87.17707 57.24 6.1759           7.46           9.20
## 75890   8.9935   8.2070 87.15285 58.54 5.6436           6.95           8.63
## 75891   8.7737   7.1126 87.12301 60.27 5.3278           6.40           8.08
## 75892   8.0191   6.6497 87.11690 61.82 5.1479           6.01           7.72
## 75893   7.8239   6.5478 87.10506 61.95 4.7801           5.73           7.48
## 75894   6.5709   6.1340 87.09572 69.49 3.2719           5.35           7.00
##       T_CANOPY_2_2_1 SW_DIF SW_IN_1_1_1 SW_IN_1_1_2 SW_OUT LW_IN LW_OUT
## 75889           9.28    0.1       -3.30         0.7   1.11 288.5  358.8
## 75890           8.45    1.0       -4.72         2.0  -0.45 288.2  355.0
## 75891           8.02    0.7       -2.76         1.4   1.03 286.7  351.5
## 75892           7.54    0.6       -2.63         1.2   1.23 285.9  350.1
## 75893           7.23    1.7       -4.42         2.6  -0.59 286.0  348.0
## 75894           7.05    3.2       -4.47         4.0  -0.47 280.1  345.0
##       LW_BC_IN LW_BC_OUT PPFD_IN_1_1_1 PPFD_IN_1_2_1 PPFD_IN_1_3_1
## 75889    286.0     360.1         -0.32          0.12         -0.36
## 75890    285.5     354.7         -0.31          0.09         -0.38
## 75891    284.1     351.1         -0.32          0.09         -0.40
## 75892    283.4     348.0         -0.32          0.10         -0.38
## 75893    283.5     346.1         -0.32          0.09         -0.39
## 75894    278.9     346.8         -0.29          0.12         -0.36
##       PPFD_IN_1_4_1 PPFD_OUT PPFD_BC_IN_1_1_1 PPFD_BC_IN_2_1_1 PPFD_BC_IN_3_1_1
## 75889          0.40    -0.50             0.03            -0.39            -0.40
```

```
## 75890          0.37    -0.50             0.02            -0.39            -0.40
## 75891          0.38    -0.52             0.04            -0.38            -0.38
## 75892          0.39    -0.52             0.04            -0.38            -0.38
## 75893          0.37    -0.53             0.03            -0.38            -0.38
## 75894          0.39    -0.50             0.04            -0.38            -0.38
##       SWC_1_1_1 SWC_1_2_1 SWC_1_3_1 SWC_1_4_1 SWC_1_5_1 SWC_1_6_1 SWC_1_7_1
## 75889        NA        NA        NA        NA        NA        NA        NA
## 75890        NA        NA        NA        NA        NA        NA        NA
## 75891        NA        NA        NA        NA        NA        NA        NA
## 75892        NA        NA        NA        NA        NA        NA        NA
## 75893        NA        NA        NA        NA        NA        NA        NA
## 75894        NA        NA        NA        NA        NA        NA        NA
##       SWC_1_8_1 SWC_2_1_1 SWC_2_2_1 SWC_2_3_1 SWC_2_4_1 SWC_2_5_1 SWC_2_6_1
## 75889        NA        NA        NA        NA        NA        NA        NA
## 75890        NA        NA        NA        NA        NA        NA        NA
## 75891        NA        NA        NA        NA        NA        NA        NA
## 75892        NA        NA        NA        NA        NA        NA        NA
## 75893        NA        NA        NA        NA        NA        NA        NA
## 75894        NA        NA        NA        NA        NA        NA        NA
##       SWC_2_7_1 SWC_2_8_1 SWC_3_1_1 SWC_3_2_1 SWC_3_3_1 SWC_3_4_1 SWC_3_5_1
## 75889        NA        NA        NA        NA        NA        NA        NA
## 75890        NA        NA        NA        NA        NA        NA        NA
## 75891        NA        NA        NA        NA        NA        NA        NA
## 75892        NA        NA        NA        NA        NA        NA        NA
## 75893        NA        NA        NA        NA        NA        NA        NA
## 75894        NA        NA        NA        NA        NA        NA        NA
##       SWC_3_6_1 SWC_3_7_1 SWC_3_8_1 SWC_4_1_1 SWC_4_2_1 SWC_4_3_1 SWC_4_4_1
## 75889        NA        NA        NA        NA        NA        NA        NA
## 75890        NA        NA        NA        NA        NA        NA        NA
## 75891        NA        NA        NA        NA        NA        NA        NA
## 75892        NA        NA        NA        NA        NA        NA        NA
## 75893        NA        NA        NA        NA        NA        NA        NA
## 75894        NA        NA        NA        NA        NA        NA        NA
##       SWC_4_5_1 SWC_4_6_1 SWC_4_7_1 SWC_4_8_1 SWC_5_1_1 SWC_5_2_1 SWC_5_3_1
## 75889        NA        NA        NA        NA        NA        NA        NA
## 75890        NA        NA        NA        NA        NA        NA        NA
## 75891        NA        NA        NA        NA        NA        NA        NA
## 75892        NA        NA        NA        NA        NA        NA        NA
## 75893        NA        NA        NA        NA        NA        NA        NA
## 75894        NA        NA        NA        NA        NA        NA        NA
##       SWC_5_4_1 SWC_5_5_1 SWC_5_6_1 SWC_5_7_1 SWC_5_8_1 TS_1_1_1 TS_1_2_1
## 75889        NA        NA        NA        NA        NA   14.198   16.633
## 75890        NA        NA        NA        NA        NA   13.890   16.278
## 75891        NA        NA        NA        NA        NA   13.493   15.933
## 75892        NA        NA        NA        NA        NA   13.028   15.522
## 75893        NA        NA        NA        NA        NA   12.591   15.126
## 75894        NA        NA        NA        NA        NA   12.191   14.757
##       TS_1_3_1 TS_1_4_1 TS_1_5_1 TS_1_6_1 TS_1_7_1 TS_1_8_1 TS_1_9_1 TS_2_1_1
## 75889   19.643   20.134   19.941   19.541       NA   18.237   17.725   13.940
## 75890   19.430   20.072   19.946   19.536       NA   18.238   17.727   13.681
## 75891   19.223   20.005   19.951   19.531       NA   18.239   17.728   13.318
## 75892   18.995   19.931   19.956   19.527       NA   18.241   17.730   12.876
## 75893   18.767   19.847   19.961   19.523       NA   18.242   17.732   12.497
## 75894   18.546   19.771   19.966   19.518       NA   18.244   17.733   12.165
```

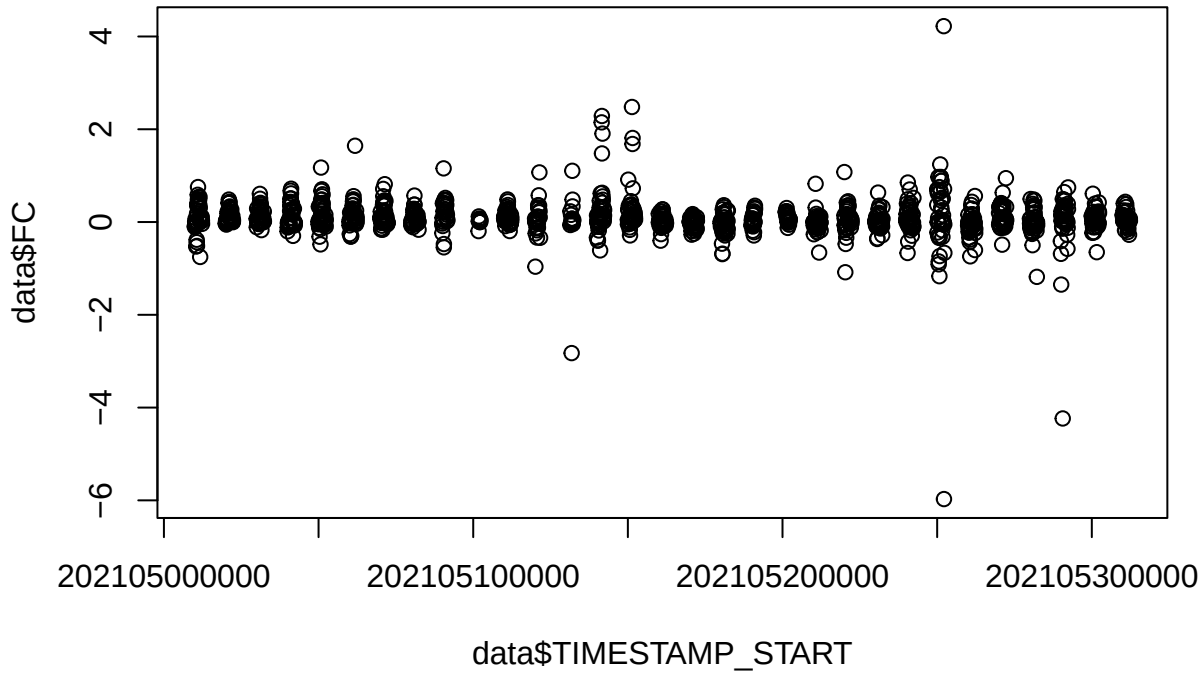
```
##       TS_2_2_1 TS_2_3_1 TS_2_4_1 TS_2_5_1 TS_2_6_1 TS_2_7_1 TS_2_8_1 TS_2_9_1
## 75889   16.193   18.832   19.316   18.928   19.130   18.500       NA   17.489
## 75890   15.884   18.642   19.254   18.934   19.126   18.499       NA   17.491
## 75891   15.577   18.457   19.187   18.940   19.122   18.499       NA   17.493
## 75892   15.218   18.258   19.112   18.947   19.119   18.499       NA   17.494
## 75893   14.872   18.057   19.031   18.956   19.116   18.499       NA   17.496
## 75894   14.568   17.863   18.946   18.961   19.112   18.499       NA   17.498
##       TS_3_1_1 TS_3_2_1 TS_3_3_1 TS_3_4_1 TS_3_5_1 TS_3_6_1 TS_3_7_1 TS_3_8_1
## 75889   14.134   16.311   19.304       NA   19.274   19.540   19.084   18.467
## 75890   13.811   15.951   19.097       NA   19.280   19.535   19.083   18.468
## 75891   13.393   15.594   18.893       NA   19.288   19.531   19.082   18.469
## 75892   12.933   15.201   18.681       NA   19.294   19.526   19.082   18.470
## 75893   12.515   14.818   18.468       NA   19.301   19.522   19.081   18.472
## 75894   12.145   14.469   18.259       NA   19.307   19.518   19.081   18.473
##       TS_3_9_1 TS_4_1_1 TS_4_2_1 TS_4_3_1 TS_4_4_1 TS_4_5_1 TS_4_6_1 TS_4_7_1
## 75889   17.764   14.100   16.381   19.424   20.142   19.497   19.368   18.860
## 75890   17.765   13.791   16.017   19.188   20.062   19.505   19.364   18.859
## 75891   17.767   13.366   15.652   18.959   19.977   19.513   19.359   18.858
## 75892   17.768   12.837   15.200   18.705   19.884   19.521   19.356   18.858
## 75893   17.769   12.359   14.763   18.452   19.783   19.528   19.352   18.859
## 75894   17.771   11.956   14.372   18.208   19.679   19.535   19.348   18.859
##       TS_4_8_1 TS_4_9_1 TS_5_1_1 TS_5_2_1 TS_5_3_1 TS_5_4_1 TS_5_5_1 TS_5_6_1
## 75889   18.292   17.689   14.557   16.719   19.653   20.484   20.415   19.681
## 75890   18.293   17.691   14.153   16.317   19.444   20.428   20.416   19.677
## 75891   18.293   17.692   13.743   15.950   19.242   20.366   20.417   19.673
## 75892   18.295   17.694   13.314   15.577   19.038   20.301   20.418   19.670
## 75893   18.297   17.695   12.907   15.211   18.831   20.230   20.419   19.667
## 75894   18.298   17.697   12.581   14.884   18.631   20.154   20.420   19.663
##       TS_5_7_1 TS_5_8_1 TS_5_9_1 G_1_1_1 G_2_1_1 G_3_1_1 NETRAD P
## 75889   19.001   18.178   17.921 -37.315 -36.529 -33.087 -74.71 0
## 75890   19.001   18.180   17.922 -36.724 -35.442 -32.408 -71.07 0
## 75891   19.002   18.181   17.924 -37.093 -36.012 -32.936 -68.59 0
## 75892   19.003   18.183   17.925 -37.914 -36.594 -33.849 -68.06 0
## 75893   19.004   18.185   17.927 -38.425 -36.891 -34.338 -65.83 0
## 75894   19.004   18.187   17.928 -38.699 -36.666 -34.449 -68.90 0
##       THROUGHFALL_1_1_1 THROUGHFALL_2_1_1 THROUGHFALL_4_1_1 THROUGHFALL_5_1_1
## 75889                 0                 0                 0                 0
## 75890                 0                 0                 0                 0
## 75891                 0                 0                 0                 0
## 75892                 0                 0                 0                 0
## 75893                 0                 0                 0                 0
## 75894                 0                 0                 0                 0
##       CH4_MIXING_RATIO_1_1_1 CH4_MIXING_RATIO_1_2_1 CH4_MIXING_RATIO_1_3_1
## 75889                     NA                     NA                     NA
## 75890                     NA                     NA                     NA
## 75891                     NA                     NA                     NA
## 75892                     NA                     NA                     NA
## 75893                     NA                     NA                     NA
## 75894                     NA                     NA                     NA
##       CH4_MIXING_RATIO_1_4_1
## 75889                     NA
## 75890                     NA
## 75891                     NA
## 75892                     NA
```

```
## 75893                     NA
## 75894                     NA
```

```
n2f <- read.csv(nameFileOut)
tstart <- data2021$TIMESTAMP_START
n2f_filtered <- data.frame()
for (x in tstart) {
  n2f_filtered <- rbind(n2f_filtered, n2f[n2f$TIMESTAMP_START == x, ])
}
```

```
# plot(n2f$TIMESTAMP_START, n2f$FC)


data = n2f_filtered[which(n2f_filtered$FC != -9999),]
plot(data$TIMESTAMP_START, data$FC)
```

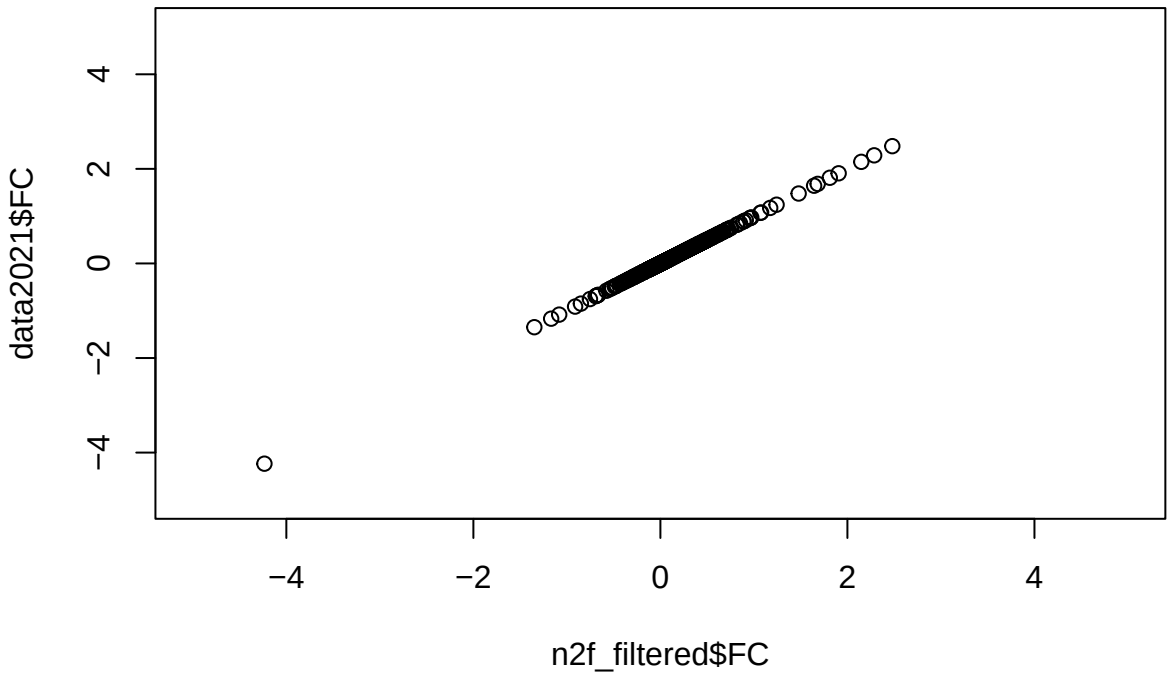



```
# data = n2f[which(n2f$FC != -9999),]
# plot(data$TIMESTAMP_START, data$FC)
#
#
# n2f_filtered
```

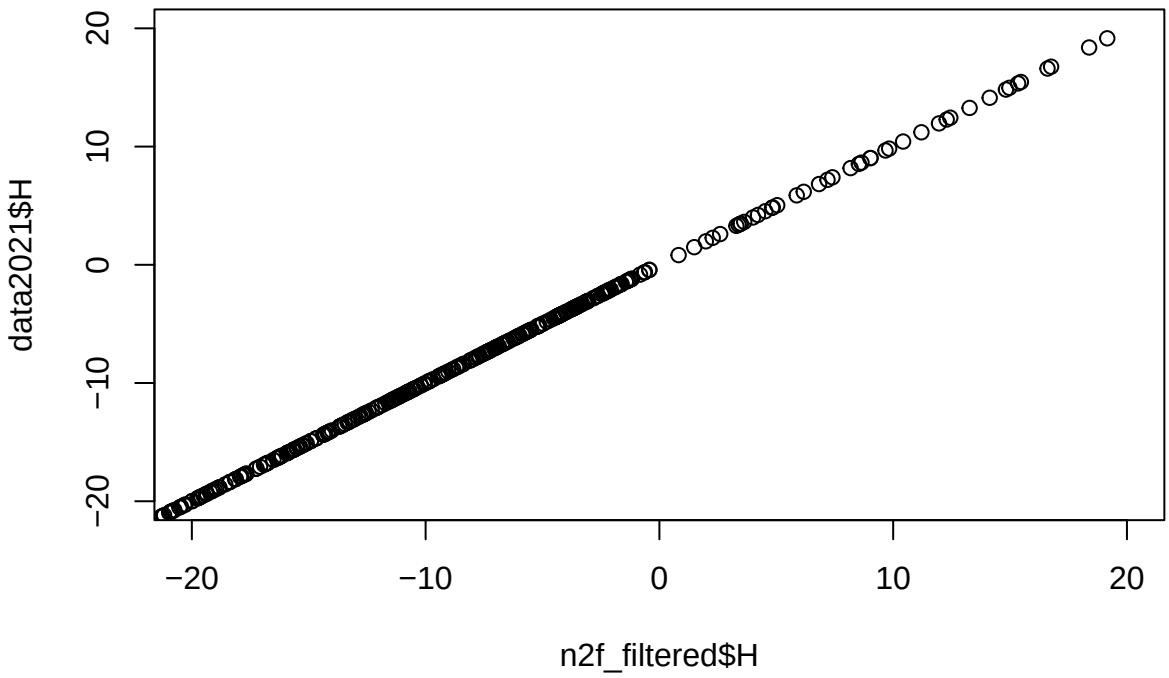
Question 2

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.

```
plot(data2021$FC ~ n2f_filtered$FC, xlim = c(-5,5), ylim = c(-5,5))
```




```
plot(data2021$H ~ n2f_filtered$H, xlim = c(-20,20), ylim = c(-20,20))
```



They look pretty close to the 1:1 line!