Ameriflux Data Submission Landscape Flux

Riasad Bin Mahbub and Benjamin Runkle

2024-05-13

Introduction and instruction links

This is an R Markdown document. In this document we are trying to compile the information of the landscape flux 2018-2023 data for AmeriFlux submission. The guidelines for this submission can be obtained from these links:

- YTvideo AMP webinar series: Submitting BADM in CSV format
- YTvideo AMP webinar series: Post-submission data life cycle: FP-In to BASE publishing
- AmeriFlux submission instructions
- AmeriFlux Data Submission PDF
- AmeriFlux Variable Information Instructions
- Uploading Half-Hourly/Hourly Data to AmeriFlux

Guidelines

Time format and Null values

1. The first two column are TIMESTAMP_START and TIMESTAMP_END (ISO time format: YYYYMMDDHHMM e.g., 201810220930)
2. Dont convert the scientific notations in timestamps
3. An hour column to check the daily data

Consistent Variable names

1. Do support list of common variable names (From the table)
2. Use the exact variable names and the units
3. Very first on the list, what to do with the data, remove the known values

Data quality check

1. U-star filtering of the data
2. CSV is a delimited text file that uses a comma to separate values
3. Convert NA and NaN at the end.

Location of the files

The location of the data can be obtained from these directories. Shared directory is the directory of the landscape flux group where the data are kept. The data were copied from the shared directory to local directory (rbmhabub's computer) to do the processing of the data

In shared directory:

Way3 Directory: "Y:/Rice/MasterFileSets/Way3/2021_11_20"

Way4 Directory: "Y:/Rice/MasterFileSets/Way4/2021_11_20"

In local directory:

Way3 Directory: "C:/Users/rbmahbub/Documents/RProjects/AmerifluxDataSubmission_LandscapeFlux/Data/Way3"

Way4 Directory: "C:/Users/rbmahbub/Documents/RProjects/AmerifluxDataSubmission_LandscapeFlux/Data/Way4"

Reading the files and fixing the timestamp: Time format and Null values

```
# Load necessary library
library(lubridate)
# Set the directory path and file name
directory_path <- "C:/Users/rbmahbub/Documents/RProjects/AmerifluxDataSubmission_LandscapeFlux/Data/Way3
file_name <- "Way3 2018.csv"
file_path <- file.path(directory_path, file_name)

# Read the CSV file
way3_2018_data <- read.csv(file_path)

# Create TIMESTAMP_START and TIMESTAMP_END columns
way3_2018_data <- cbind(TIMESTAMP_START = NA, TIMESTAMP_END = NA, way3_2018_data)
# Convert TIMESTAMP column to POSIXct format (1/1/2018  12:00:00 AM)
way3_2018_data$TIMESTAMP <- ymd_hms(way3_2018_data$TIMESTAMP)
# Create TIMESTAMP_START in the desired format
way3_2018_data$TIMESTAMP_START <- format(way3_2018_data$TIMESTAMP, "%Y%m%d%H%M")

# Create TIMESTAMP_END by adding 30 minutes to TIMESTAMP and formatting it
way3_2018_data$TIMESTAMP_END <- format(way3_2018_data$TIMESTAMP + minutes(30), "%Y%m%d%H%M")


# Create a new column 'HOUR' to store the hour extracted from the TIMESTAMP
way3_2018_data$HOUR <- hour(way3_2018_data$TIMESTAMP)

# Create a new column 'MONTH' to store the month extracted from the TIMESTAMP
way3_2018_data$MONTH <- month(way3_2018_data$TIMESTAMP)

# Create a new column 'DAY_OF_YEAR' to store the day of the year extracted from the TIMESTAMP
way3_2018_data$DOY <- yday(way3_2018_data$TIMESTAMP)


# Assuming way3_2018_data is your dataset
print(way3_2018_data[1:4, 1:4])
```

```
##   TIMESTAMP_START TIMESTAMP_END           TIMESTAMP filename
## 1    201801010000  201801010030 2018-01-01 00:00:00      NaN
## 2    201801010030  201801010100 2018-01-01 00:30:00      NaN
## 3    201801010100  201801010130 2018-01-01 01:00:00      NaN
## 4    201801010130  201801010200 2018-01-01 01:30:00      NaN
```

Consistent Variable names:

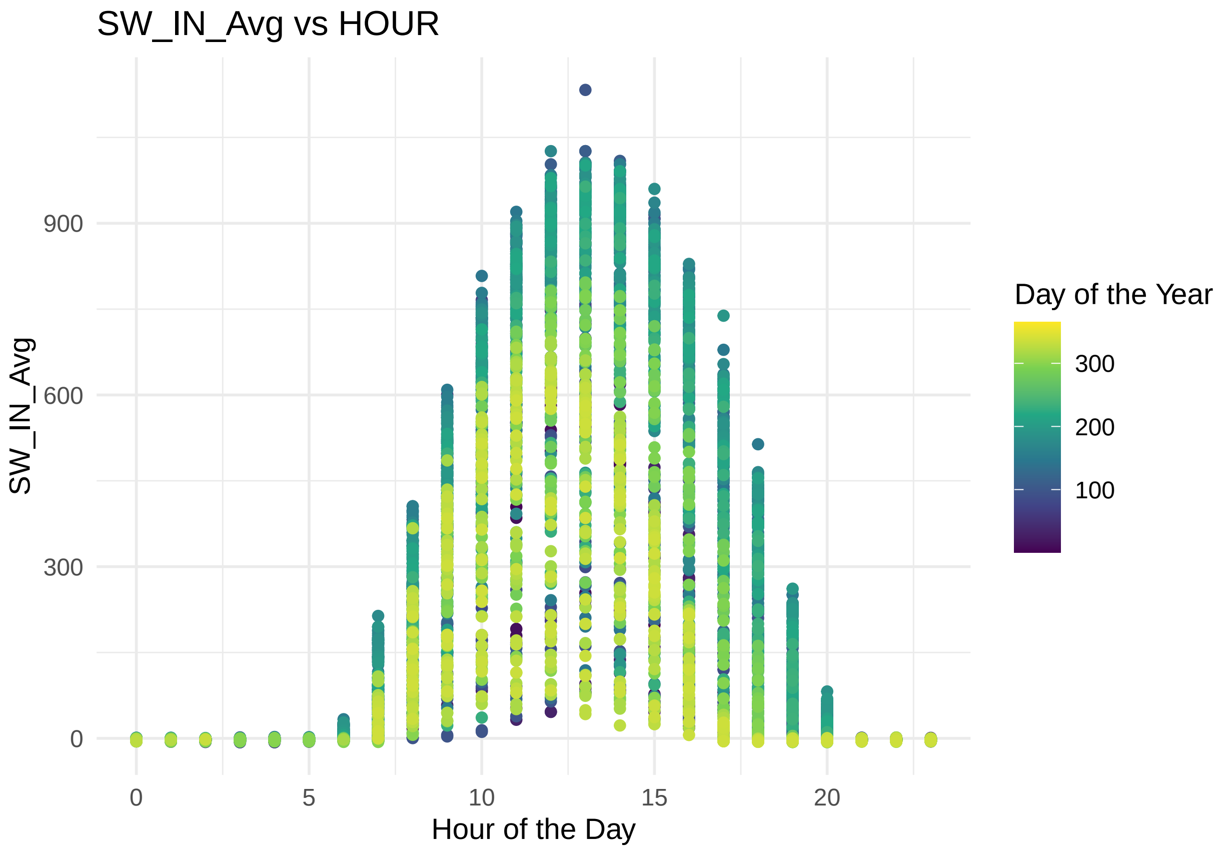
1. Filter out the variables that are relevant

2. From the description find the common variables

source of eddypro: ’https://www.licor.com/env/support/EddyPro/topics/output-files-full-output.html”

TIMESTAMP == TIMESTAMP
TIMESTAMP_START == created from TIMESTAMP
TIMESTAMP_END == created from TIMESTAMP
FETCH_70 == x_70%
FETCH_80 == NF
FETCH_90 == x_90%
FETCH_FILTER == NF
FETCH_MAX == x_peak
CH4 == ch4_mole_fraction
CH4_MIXING_RATIO == ch4_mixing_ratio
CO2 == co2_mole_fraction
CO2_MIXING_RATIO == co2_mixing_ratio
FC == co2_flux
FCH4 == ch4_flux
H2O == h2o_mole_fraction
H2O_MIXING_RATIO == h2o_mixing_ratio
FH2O == h2o_flux
G == shf_Avg.1. /shf_Avg.2./shf_Avg.3.
H == H
LE == LE
SG == NF
SH == H_strg
SLE == LE_strg
PA == air_pressure
RH == RH
T_SONIC == sonic_temperature
T_SONIC_SIGMA == NF
GPP == Needs to be derived
NEE == Needs to be derived
RECO == Needs to be derived
FC_SSITC_TEST == qc_co2_flux
FCH4_SSITC_TEST == qc_ch4_flux
H_SSITC_TEST == qc_H
LE_SSITC_TEST == qc_LE
TAU_SSITC_TEST == qc_Tau
CO2_SIGMA == co2_var
SC == co2_strg
SCH4 == ch4_strg
U_SIGMA == u_var

V_SIGMA == v_var
W_SIGMA == w_var
WD == wind_dir
WD_SIGMA == NF
WS == wind_speed
WS_MAX == max_wind_speed
ZL == X_z_d___L
TA == air_temperature
VPD == VPD
P == NF
LW_IN == LW_IN_Avg
LW_OUT == LW_OUT_Avg
PPFD_IN == PAR_IN_Avg
PPFD_OUT == PAR_OUT_Avg
SW_IN == SW_IN_Avg
SW_OUT == SW_OUT_Avg
SWC= SWC_1_1_1
TS == TS_2_1_2/ TS_2_2_2
WTD == WTD_Avg/ Lvl_m_Avg
MO_LENGTH == L
TAU == Tau
USTAR == u*/u_



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

Change the name of the columns

```r
# Assuming way3_2018_data is your DataFrame
library(tidyverse)
way3_2018_data_filtered <- way3_2018_data %>%
  select(
    TIMESTAMP_START,
    TIMESTAMP_END,
    `x_70_`,
    `x_90_`,
    x_peak,
    ch4_mole_fraction,
    ch4_mixing_ratio,
    co2_mole_fraction,
    co2_mixing_ratio,
    co2_flux,
    ch4_flux,
    h2o_mole_fraction,
    h2o_mixing_ratio,
    h2o_flux,
    #`shf_Avg.1./shf_Avg.2./shf_Avg.3.`,
    H,
    LE,
    #NF,  # Duplicate NF keys are ignored
    H_strg,
    LE_strg,
    air_pressure,
    RH,
    sonic_temperature,
    #NF,  # Duplicate NF keys are ignored
    #`Needs to be derived`,  # Duplicate "Needs to be derived" are ignored
    qc_co2_flux,
    qc_ch4_flux,
    qc_H,
    qc_LE,
    qc_Tau,
    co2_var,
    co2_strg,
    ch4_strg,
    u_var,
    v_var,
    w_var,
    wind_dir,
    #NF,  # Duplicate NF keys are ignored
    wind_speed,
    max_wind_speed,
    X_z_d__L,
    air_temperature,
    VPD,
    #NF,  # Duplicate NF keys are ignored
    LW_IN_Avg,
    LW_OUT_Avg,
    PAR_IN_Avg,
    PAR_OUT_Avg,
    SW_IN_Avg,
    SW_OUT_Avg,
    SWC_1_1_1,
    #`TS_2_1_2/TS_2_2_2`,
    #`WTD_Avg/Lvl_m_Avg`,
    L,
    Tau
    #`u*/u_`
  )

# Rename the filtered columns
way3_2018_data_f1iltered <- way3_2018_data_filtered %>%
  rename(
    TIMESTAMP_START = TIMESTAMP_START,
    TIMESTAMP_END = TIMESTAMP_END,
    FETCH_70 = `x_70_`,
    FETCH_90 = `x_90_`,
    FETCH_MAX = x_peak,
    CH4 = ch4_mole_fraction,
    CH4_MIXING_RATIO = ch4_mixing_ratio,
    CO2 = co2_mole_fraction,
    CO2_MIXING_RATIO = co2_mixing_ratio,
    FC = co2_flux,
    FCH4 = ch4_flux,
    H2O = h2o_mole_fraction,
    H2O_MIXING_RATIO = h2o_mixing_ratio,
    FH2O = h2o_flux,
    #G = `shf_Avg.1./shf_Avg.2./shf_Avg.3.`,
    H = H,
    LE = LE,
    #SG = NF,  # Duplicate NF keys are ignored
    SH = H_strg,
    SLE = LE_strg,
    PA = air_pressure,
    RH = RH,
    T_SONIC = sonic_temperature,
    #T_SONIC_SIGMA = NF,  # Duplicate NF keys are ignored
    #GPP = `Needs to be derived`,  # Duplicate "Needs to be derived" are ignored
    #NEE = `Needs to be derived`,  # Duplicate "Needs to be derived" are ignored
    #RECO = `Needs to be derived`,  # Duplicate "Needs to be derived" are ignored
    FC_SSITC_TEST = qc_co2_flux,
    FCH4_SSITC_TEST = qc_ch4_flux,
    H_SSITC_TEST = qc_H,
    LE_SSITC_TEST = qc_LE,
    TAU_SSITC_TEST = qc_Tau,
    CO2_SIGMA = co2_var,
    SC = co2_strg,
    SCH4 = ch4_strg,
    U_SIGMA = u_var,
    V_SIGMA = v_var,
    W_SIGMA = w_var,
    WD = wind_dir,
    #WD_SIGMA = NF,  # Duplicate NF keys are ignored
    WS = wind_speed,
    WS_MAX = max_wind_speed,
    ZL = X_z_d__L,
    TA = air_temperature,
    VPD = VPD,
    #P = NF,  # Duplicate NF keys are ignored
    LW_IN = LW_IN_Avg,
    LW_OUT = LW_OUT_Avg,
    PPFD_IN = PAR_IN_Avg,
    PPFD_OUT = PAR_OUT_Avg,
    SW_IN = SW_IN_Avg,
    SW_OUT = SW_OUT_Avg,
    SWC = SWC_1_1_1,
    #TS = `TS_2_1_2/TS_2_2_2`,
    #WTD = `WTD_Avg/Lvl_m_Avg`,
    MO_LENGTH = L,
    TAU = Tau
    #USTAR = `u*/u_`
  )
```

Export the data

```
# Create the directory if it doesn't exist
# Convert TIMESTAMP_START and TIMESTAMP_END to character type
way3_2018_data_f1iltered$TIMESTAMP_START <- as.character(way3_2018_data_f1iltered$TIMESTAMP_START)
way3_2018_data_f1iltered$TIMESTAMP_END <- as.character(way3_2018_data_f1iltered$TIMESTAMP_END)

# Replace NaN and NA values with -9999
way3_2018_data_f1iltered[is.na(way3_2018_data_f1iltered)] <- -9999
# Define custom function to handle NaN values in data frames
is.nan.data.frame <- function(x) {
  do.call(cbind, lapply(x, is.nan))
}
# Replace NaN with -9999
way3_2018_data_f1iltered[is.nan.data.frame(way3_2018_data_f1iltered)] <- -9999


dir.create("C:/Users/rbmahbub/Documents/RProjects/AmerifluxDataSubmission_LandscapeFlux/Data/AFguidedSu

# Specify the file path for saving
file_path <- "C:/Users/rbmahbub/Documents/RProjects/AmerifluxDataSubmission_LandscapeFlux/Data/AFguided

# Save the dataframe
write.csv(way3_2018_data_f1iltered, file = file_path, row.names = FALSE)

# Confirmation message
cat("way3_2018_data saved successfully.\n")
```

```
## way3_2018_data saved successfully.
```

Before saving the files these information needs to be checked

Precipitation comes from way 4 so get the precipitation data from way 4 for each year and put them in way 4 The G data that we obtain Create a fetch filter column where “You can make this by using ~270+/-85 degrees as 1 (keep it) and other wind directions as 0 (discard it)” We have a how-to, for G, I’ll look for it if you don’t remember seeing it

Read all way 3 and way 4 files

Check if they have same number of columns

Check if they have same columns

Check if they have same serialized columns

```r
# Load necessary libraries
# Load necessary libraries
library(dplyr)

# Function to read all files and return a list of dataframes
read_files <- function(file_paths) {
  lapply(file_paths, read.csv, stringsAsFactors = FALSE)
}

# Function to print the number of rows and columns for each dataframe
print_dimensions <- function(data_list, file_names) {
  for (i in seq_along(data_list)) {
    rows <- nrow(data_list[[i]])
    cols <- ncol(data_list[[i]])
    cat("File:", file_names[i], "- Rows:", rows, "- Columns:", cols, "\n")
  }
}

# Specify the file paths for way 3 and way 4 files
way3_dir <- "C:/Users/rbmahbub/Documents/RProjects/AmerifluxDataSubmission_LandscapeFlux/Data/Way3"
way4_dir <- "C:/Users/rbmahbub/Documents/RProjects/AmerifluxDataSubmission_LandscapeFlux/Data/Way4"

way3_files <- list.files(path = way3_dir, pattern = "*.csv", full.names = TRUE)
way4_files <- list.files(path = way4_dir, pattern = "*.csv", full.names = TRUE)

# Read all files
way3_data <- read_files(way3_files)
way4_data <- read_files(way4_files)

# Print the number of rows and columns for each file
print_dimensions(way3_data, basename(way3_files))
```

```
## File: Way3 2018.csv - Rows: 17520 - Columns: 528
## File: Way3 2019.csv - Rows: 17520 - Columns: 528
## File: Way3 2020.csv - Rows: 17568 - Columns: 528
## File: Way3 2021.csv - Rows: 12602 - Columns: 528
## File: Way3 2022.csv - Rows: 17473 - Columns: 121
## File: Way3 2023.csv - Rows: 17473 - Columns: 121
```

```r
print_dimensions(way4_data, basename(way4_files))
```

```
## File: Way4 2018.csv - Rows: 17520 - Columns: 481
## File: Way4 2019.csv - Rows: 17520 - Columns: 481
## File: Way4 2020.csv - Rows: 17568 - Columns: 481
## File: Way4 2021.csv - Rows: 13811 - Columns: 481
## File: Way4 2022 WTD_Corr.csv - Rows: 17473 - Columns: 140
## File: Way4 2022.csv - Rows: 17473 - Columns: 140
## File: Way4 2023 WTD_Corr.csv - Rows: 11185 - Columns: 135
## File: Way4 2023.csv - Rows: 17474 - Columns: 135
```

```r
# Function to check if all dataframes in a list have the same number of columns
check_same_num_columns <- function(data_list) {
  num_columns <- sapply(data_list, ncol)
  return(length(unique(num_columns)) == 1)
}

# Function to check if all dataframes in a list have the same column names
check_same_columns <- function(data_list) {
  column_names <- lapply(data_list, colnames)
  return(length(unique(column_names)) == 1)
}

# Function to check if all dataframes in a list have the same serialized columns
check_same_serialized_columns <- function(data_list) {
  serialized_columns <- sapply(data_list, function(df) paste(colnames(df), collapse = ""))
  return(length(unique(serialized_columns)) == 1)
}

# Check way 3 files
way3_same_num_columns <- check_same_num_columns(way3_data)
way3_same_columns <- check_same_columns(way3_data)
way3_same_serialized_columns <- check_same_serialized_columns(way3_data)

# Check way 4 files
way4_same_num_columns <- check_same_num_columns(way4_data)
way4_same_columns <- check_same_columns(way4_data)
way4_same_serialized_columns <- check_same_serialized_columns(way4_data)

# Compare way 3 and way 4 files
if (way3_same_num_columns && way4_same_num_columns) {
  way3_num_columns <- ncol(way3_data[[1]])
  way4_num_columns <- ncol(way4_data[[1]])
  same_num_columns <- (way3_num_columns == way4_num_columns)
} else {
  same_num_columns <- FALSE
}

if (way3_same_columns && way4_same_columns) {
  way3_columns <- colnames(way3_data[[1]])
  way4_columns <- colnames(way4_data[[1]])
  same_columns <- all(way3_columns %in% way4_columns) && all(way4_columns %in% way3_columns)
} else {
  same_columns <- FALSE
}

if (way3_same_serialized_columns && way4_same_serialized_columns) {
  way3_serialized_columns <- paste(colnames(way3_data[[1]]), collapse = "")
  way4_serialized_columns <- paste(colnames(way4_data[[1]]), collapse = "")
  same_serialized_columns <- (way3_serialized_columns == way4_serialized_columns)
} else {
  same_serialized_columns <- FALSE
}

# Output the results
results <- list(
  way3_same_num_columns = way3_same_num_columns,
  way3_same_columns = way3_same_columns,
  way3_same_serialized_columns = way3_same_serialized_columns,
  way4_same_num_columns = way4_same_num_columns,
  way4_same_columns = way4_same_columns,
  way4_same_serialized_columns = way4_same_serialized_columns,
  same_num_columns = same_num_columns,
  same_columns = same_columns,
  same_serialized_columns = same_serialized_columns
)

print(results)
```

```
## $way3_same_num_columns
## [1] FALSE
##
## $way3_same_columns
## [1] FALSE
##
## $way3_same_serialized_columns
## [1] FALSE
##
## $way4_same_num_columns
## [1] FALSE
##
## $way4_same_columns
## [1] FALSE
##
## $way4_same_serialized_columns
## [1] FALSE
##
## $same_num_columns
## [1] FALSE
##
## $same_columns
## [1] FALSE
##
## $same_serialized_columns
## [1] FALSE
```

```r
# Define the file path
file_path <- "C:/Users/rbmahbub/Documents/RProjects/AmerifluxDataSubmission_LandscapeFlux/Data/Variable/

# Read the CSV file using read.csv
data <- read.csv(file_path)

# Print the first few rows of the data to verify
#head(data)
nrow(data)
```

```
## [1] 144
```

TIMESTAMP_START TIMESTAMP_END TIMESTAMP way3_data[[1]], 2018 way3_data[[2]], 2019 way3_data[[3]], 2020 way3_data[[4]], 2021 way3_data[[5]], 2022 way3_data[[6]], 2023

way4_data[[1]], 2018 way4_data[[2]], 2019 way4_data[[3]], 2020 way4_data[[4]], 2021 way4_data[[5]], 2022 way4_data[[6]], 2023

```r
#colnames(way3_data[[1]])
```

```r
# Load necessary libraries
library(readxl)
library(dplyr)
library(openxlsx)

# Define the path to the Excel file
file_path <- "C:/Users/rbmahbub/Documents/RProjects/AmerifluxDataSubmission_LandscapeFlux/Data/Variable/

# Read the third and fourth sheets
sheet3 <- read_excel(file_path, sheet = 3)
sheet4 <- read_excel(file_path, sheet = 4)

# Display the first few rows of both sheets to understand their structure
print("Sheet 3:")
```

```
## [1] "Sheet 3:"
```

```r
print(head(sheet3))
```

```
## # A tibble: 6 x 1
##   Label
##   <chr>
## 1 TIMESTAMP
## 2 filename
## 3 date
## 4 time
## 5 DOY
## 6 daytime
```

```r
print("\nSheet 4:")
```

```
## [1] "\nSheet 4:"
```

```
print(head(sheet4))
```

```
## # A tibble: 6 x 4
##   Label        Units        ...3  ...4
##   <chr>        <chr>        <chr> <chr>
## 1 filename     <NA>         NaN   <NA>
## 2 date         [yyyy-mm-dd] NaN   <NA>
## 3 time         [HH:MM]      NaN   <NA>
## 4 DOY          [ddd.ddd]    NaN   <NA>
## 5 daytime      [1=daytime]  NaN   <NA>
## 6 file_records [#]          NaN   <NA>
```

```
# Merge the sheets based on the first column
merged_data <- merge(sheet3, sheet4, by = names(sheet3)[1])

# Display the merged data
print("Merged Data:")
```

```
## [1] "Merged Data:"
```

```
print(head(merged_data))
```

```
##                 Label                      Units ...3 ...4
## 1 absolute_limits_hf 8u/v/w/ts/co2/h2o/ch4/none  NaN <NA>
## 2        air_density                 [kg+1m-3]  NaN <NA>
## 3  air_heat_capacity              [J+1kg-1K-1]  NaN <NA>
## 4   air_molar_volume                [m+3mol-1]  NaN <NA>
## 5         air_p_mean                        --  NaN <NA>
## 6       air_pressure                      [Pa]  NaN <NA>
```

```
View(merged_data)

# Save the merged data to a new Excel file
output_file_path <- "C:/Users/rbmahbub/Documents/RProjects/AmerifluxDataSubmission_LandscapeFlux/Data/Va
write.xlsx(merged_data, output_file_path)

print(paste("Merged data saved to:", output_file_path))
```

```
## [1] "Merged data saved to: C:/Users/rbmahbub/Documents/RProjects/AmerifluxDataSubmission_LandscapeFlu
```

Units file with the

```
# Define the file paths
file_path_met <- "C:/Users/rbmahbub/Documents/RProjects/AmerifluxDataSubmission_LandscapeFlux/Data/Units
file_path_ec <- "C:/Users/rbmahbub/Documents/RProjects/AmerifluxDataSubmission_LandscapeFlux/Data/Units_
file_path_soil <- "C:/Users/rbmahbub/Documents/RProjects/AmerifluxDataSubmission_LandscapeFlux/Data/Unit

# Read the files into different variables
way3_met_units <- read.csv(file_path_met, header = FALSE, sep = ",")
```

```r
way3_ec_ec_units <- read.csv(file_path_ec, header = FALSE, sep = ",")
way3_soil_soil_units <- read.csv(file_path_soil, header = FALSE, sep = ",")

# Print the number of columns and column names for each file
cat("Way3_Met_units:\n")
```

```
## Way3_Met_units:
```

```r
cat("Number of columns:", ncol(way3_met_units), "\n")
```

```
## Number of columns: 222
```

```r
print(colnames(way3_met_units))
```

```
##   [1] "V1"   "V2"   "V3"   "V4"   "V5"   "V6"   "V7"   "V8"   "V9"   "V10" 
##  [11] "V11"  "V12"  "V13"  "V14"  "V15"  "V16"  "V17"  "V18"  "V19"  "V20" 
##  [21] "V21"  "V22"  "V23"  "V24"  "V25"  "V26"  "V27"  "V28"  "V29"  "V30" 
##  [31] "V31"  "V32"  "V33"  "V34"  "V35"  "V36"  "V37"  "V38"  "V39"  "V40" 
##  [41] "V41"  "V42"  "V43"  "V44"  "V45"  "V46"  "V47"  "V48"  "V49"  "V50" 
##  [51] "V51"  "V52"  "V53"  "V54"  "V55"  "V56"  "V57"  "V58"  "V59"  "V60" 
##  [61] "V61"  "V62"  "V63"  "V64"  "V65"  "V66"  "V67"  "V68"  "V69"  "V70" 
##  [71] "V71"  "V72"  "V73"  "V74"  "V75"  "V76"  "V77"  "V78"  "V79"  "V80" 
##  [81] "V81"  "V82"  "V83"  "V84"  "V85"  "V86"  "V87"  "V88"  "V89"  "V90" 
##  [91] "V91"  "V92"  "V93"  "V94"  "V95"  "V96"  "V97"  "V98"  "V99"  "V100"
## [101] "V101" "V102" "V103" "V104" "V105" "V106" "V107" "V108" "V109" "V110"
## [111] "V111" "V112" "V113" "V114" "V115" "V116" "V117" "V118" "V119" "V120"
## [121] "V121" "V122" "V123" "V124" "V125" "V126" "V127" "V128" "V129" "V130"
## [131] "V131" "V132" "V133" "V134" "V135" "V136" "V137" "V138" "V139" "V140"
## [141] "V141" "V142" "V143" "V144" "V145" "V146" "V147" "V148" "V149" "V150"
## [151] "V151" "V152" "V153" "V154" "V155" "V156" "V157" "V158" "V159" "V160"
## [161] "V161" "V162" "V163" "V164" "V165" "V166" "V167" "V168" "V169" "V170"
## [171] "V171" "V172" "V173" "V174" "V175" "V176" "V177" "V178" "V179" "V180"
## [181] "V181" "V182" "V183" "V184" "V185" "V186" "V187" "V188" "V189" "V190"
## [191] "V191" "V192" "V193" "V194" "V195" "V196" "V197" "V198" "V199" "V200"
## [201] "V201" "V202" "V203" "V204" "V205" "V206" "V207" "V208" "V209" "V210"
## [211] "V211" "V212" "V213" "V214" "V215" "V216" "V217" "V218" "V219" "V220"
## [221] "V221" "V222"
```

```r
cat("\nWay3_EC_EC_units:\n")
```

```
## 
## Way3_EC_EC_units:
```

```r
cat("Number of columns:", ncol(way3_ec_ec_units), "\n")
```

```
## Number of columns: 194
```

```r
print(colnames(way3_ec_ec_units))
```

```
##   [1] "V1"   "V2"   "V3"   "V4"   "V5"   "V6"   "V7"   "V8"   "V9"   "V10"
##  [11] "V11"  "V12"  "V13"  "V14"  "V15"  "V16"  "V17"  "V18"  "V19"  "V20"
##  [21] "V21"  "V22"  "V23"  "V24"  "V25"  "V26"  "V27"  "V28"  "V29"  "V30"
##  [31] "V31"  "V32"  "V33"  "V34"  "V35"  "V36"  "V37"  "V38"  "V39"  "V40"
##  [41] "V41"  "V42"  "V43"  "V44"  "V45"  "V46"  "V47"  "V48"  "V49"  "V50"
##  [51] "V51"  "V52"  "V53"  "V54"  "V55"  "V56"  "V57"  "V58"  "V59"  "V60"
##  [61] "V61"  "V62"  "V63"  "V64"  "V65"  "V66"  "V67"  "V68"  "V69"  "V70"
##  [71] "V71"  "V72"  "V73"  "V74"  "V75"  "V76"  "V77"  "V78"  "V79"  "V80"
##  [81] "V81"  "V82"  "V83"  "V84"  "V85"  "V86"  "V87"  "V88"  "V89"  "V90"
##  [91] "V91"  "V92"  "V93"  "V94"  "V95"  "V96"  "V97"  "V98"  "V99"  "V100"
## [101] "V101" "V102" "V103" "V104" "V105" "V106" "V107" "V108" "V109" "V110"
## [111] "V111" "V112" "V113" "V114" "V115" "V116" "V117" "V118" "V119" "V120"
## [121] "V121" "V122" "V123" "V124" "V125" "V126" "V127" "V128" "V129" "V130"
## [131] "V131" "V132" "V133" "V134" "V135" "V136" "V137" "V138" "V139" "V140"
## [141] "V141" "V142" "V143" "V144" "V145" "V146" "V147" "V148" "V149" "V150"
## [151] "V151" "V152" "V153" "V154" "V155" "V156" "V157" "V158" "V159" "V160"
## [161] "V161" "V162" "V163" "V164" "V165" "V166" "V167" "V168" "V169" "V170"
## [171] "V171" "V172" "V173" "V174" "V175" "V176" "V177" "V178" "V179" "V180"
## [181] "V181" "V182" "V183" "V184" "V185" "V186" "V187" "V188" "V189" "V190"
## [191] "V191" "V192" "V193" "V194"
```

```
cat("\nWay3_Soil_SOIL_units:\n")
```

```
##
## Way3_Soil_SOIL_units:
```

```
cat("Number of columns:", ncol(way3_soil_soil_units), "\n")
```

```
## Number of columns: 90
```

```
print(colnames(way3_soil_soil_units))
```

```
##  [1] "V1"  "V2"  "V3"  "V4"  "V5"  "V6"  "V7"  "V8"  "V9"  "V10" "V11" "V12"
## [13] "V13" "V14" "V15" "V16" "V17" "V18" "V19" "V20" "V21" "V22" "V23" "V24"
## [25] "V25" "V26" "V27" "V28" "V29" "V30" "V31" "V32" "V33" "V34" "V35" "V36"
## [37] "V37" "V38" "V39" "V40" "V41" "V42" "V43" "V44" "V45" "V46" "V47" "V48"
## [49] "V49" "V50" "V51" "V52" "V53" "V54" "V55" "V56" "V57" "V58" "V59" "V60"
## [61] "V61" "V62" "V63" "V64" "V65" "V66" "V67" "V68" "V69" "V70" "V71" "V72"
## [73] "V73" "V74" "V75" "V76" "V77" "V78" "V79" "V80" "V81" "V82" "V83" "V84"
## [85] "V85" "V86" "V87" "V88" "V89" "V90"
```

Saving the files