

LICOR Data Processing

Bryan Blue

2023-11-28

01_load_LICOR_logs

Current loads one text log from a LICOR data set. It reads each chunk of data from a line with “[Header]” skipping all lines until a line “[Data]” is found. Once found all data lines are read and assigned a group number to keep the related records together. This repeats until the end of file.

The original group names, field names, and units are stored off as text in separate dataframes from the data. There are many special characters that do not convert into column names without loss of uniqueness. The values call all be referenced by the column position.

The consolidated data is written out to an Excel CSV type text file.

TODO: Check that grouped data records are from the correct data set. Averages of the groups are needed, not the raw records for analysis.

```
load_data_chunks <- function(filename) {
  con <- file(filename, "rt")
  print("start")
  i = 0 # used to specify which group of data is being read
  dataLines <- data.frame(matrix(ncol = 1, nrow = 0))
  colnames(dataLines) <- c("original")

  while (length(oneLine <- readLines(con, n = 1, skipNul = TRUE, warn = FALSE)) > 0) {
    # print(oneLine)
    if (str_detect(oneLine, regex("\\[Data\\]"))) {
      # print("We found data start")

      i = i + 1
      groupNumber = paste("group",i, sep = "")

      # these three groups appear after [Data] in the file
```

```

dataGroups <- paste("data_group", readLines(con, n = 1, skipNul = TRUE, warn = FALSE))
dataVars    <- paste("data_group", readLines(con, n = 1, skipNul = TRUE, warn = FALSE))
dataUnits   <- paste("data_group", readLines(con, n = 1, skipNul = TRUE, warn = FALSE))

# create an empty list to store all of the data records
# read lines until it is not longer a data line

while (length(dataLine <- readLines(con, n = 1, skipNul = TRUE, warn = FALSE)) > 0) {
  # print(dataLine)
  # lines starts with and integer record number and tab
  if (str_detect(dataLine, regex("[0-9]?\\t"))) {
    finalString <- paste(groupNumber, dataLine, sep = "\\t")
    dataLines[nrow(dataLines) + 1,] = finalString
  } else {
    break
  }
}

print("stop")
close(con)

return(list(dataGroups, dataUnits, dataVars, dataLines))
}

# this will not work for multiple files without the structure, columns
# create df for each of the chunks of information
output <- load_data_chunks("data_raw/2023-11-17-btrf/2023-11-17-1144_logdata-tc-heat")

```

```

[1] "start"
[1] "stop"

```

```

groupsdf <- data.frame(do.call('rbind',strsplit(as.character(output[[1]]),'\\t',fixed=TRUE)))
unitsdf <- data.frame(do.call('rbind',strsplit(as.character(output[[2]]),'\\t',fixed=TRUE)))
varsdf <- data.frame(do.call('rbind',strsplit(as.character(output[[3]]),'\\t',fixed=TRUE)))
valuesdf <- data.frame(do.call('rbind',strsplit(as.character(output[[4]][,1]),'\\t',fixed=TRUE)))

# write out a text file that Excel can read

```

```

write_excel_csv(groupsdf, "data_cleaned/test_licor_data_cleaned.txt", append = TRUE, col_n
write_excel_csv(unitsdf, "data_cleaned/test_licor_data_cleaned.txt", append = TRUE, col_na
write_excel_csv(varsdf, "data_cleaned/test_licor_data_cleaned.txt", append = TRUE, col_nam
write_excel_csv(valuesdf, "data_cleaned/test_licor_data_cleaned.txt", append = TRUE, col_n
#
# col_types = cols(SysObs_date = col_datetime(format = "%Y%m%d %H:%M:%S"),
# col_types$SysObs_date <- as.Date.POSIXct(col_types$SysObs_date)

# load the names of the columns of interest
select_field_list <- read_csv("data_raw/Joost_Analysis_Sample/2023-09-22-1536_logdata_fiel
      col_names = FALSE,
      # skip = 1,
      n_max = 2,
      show_col_types = FALSE)
# Append new column to front of data, group_name is required as column 1
select_field_list <- data.frame(col1 = c("data_group"), select_field_list)
# variable names are not unique, prepend the group to fix it
new_select_field_list <- paste(select_field_list[1,], select_field_list[2,], sep = "_")
colnames(select_field_list) <- new_select_field_list

# variable names are not unique, prepend the group to fix it
new_field_list <- paste(groupsdf[1,], varsdf[1,], sep = "_")
colnames(valuesdf) <- new_field_list #varsdf[1,]

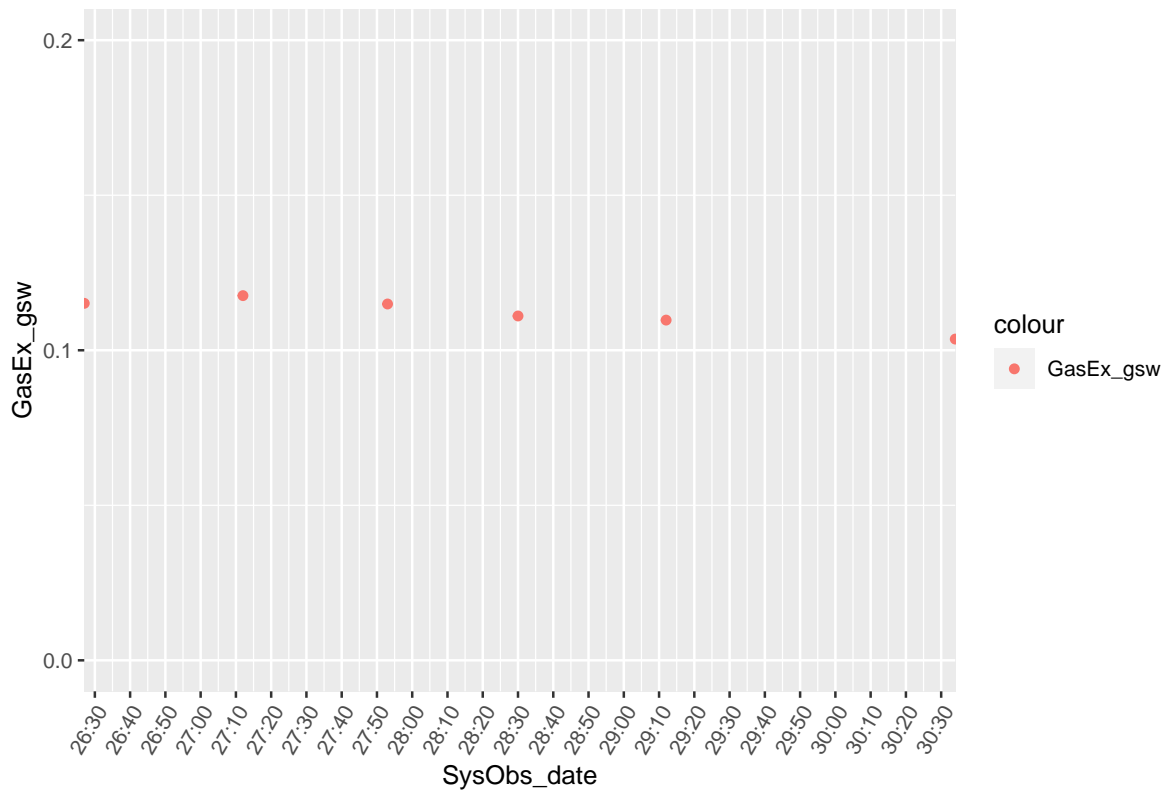
# create a set of data that the user specified in select_field_list
final_data <- valuesdf %>% select(any_of(new_select_field_list))
# final_data$SysObs_date <- strptime(final_data$SysObs_date, "%Y%m%d %H:%M:%S")
final_data$SysObs_date <- as.POSIXct(final_data$SysObs_date, format = "%Y%m%d %H:%M:%S")
final_data$GasEx_gsw <- as.numeric(final_data$GasEx_gsw)
write_excel_csv(final_data, "data_cleaned/test_licor_selected_data.txt", append = FALSE, c

ggplot() +
  # geom_point(data = final_data,
  # geom_point(data=subset(final_data, data_group_data_group=="group2"),
  #           aes(x = SysObs_date, y = GasEx_gsw, color = "GasEx_gsw")) +
  # geom_line(data = final_data, aes(x = SysObs_date, y = GasEx_gsw, color = "GasEx_gsw"
  # geom_line(data = final_data, aes(x = date, y = temperature, color = sensor_id)) +
  # geom_line(data = final_data, aes(x = date, y = temperature, color = sensor_id)) +
  # geom_line(data = final_data, aes(x = date, y = temperature, color = sensor_id)) +

```

```
# geom_line(data = final_data, aes(x = date, y = TEMP_THREE_FOOT_HEIGHT, color = "CEAC
ggtitle("LICOR Test Graph - GasEx gsw CONTROL LEAF") +
xlab("SysObs_date") +
ylab("GasEx_gsw") +
scale_y_continuous(limits=c(0.0, 0.2), breaks=c(0, 0.1, 0.2, 0.5)) +
scale_x_datetime(date_labels = ("%M:%S"),
  date_breaks = "10 sec", expand = expansion(0)) +
theme(axis.text.x=element_text(angle=60, hjust=1))
```

LICOR Test Graph – GasEx gsw CONTROL LEAF



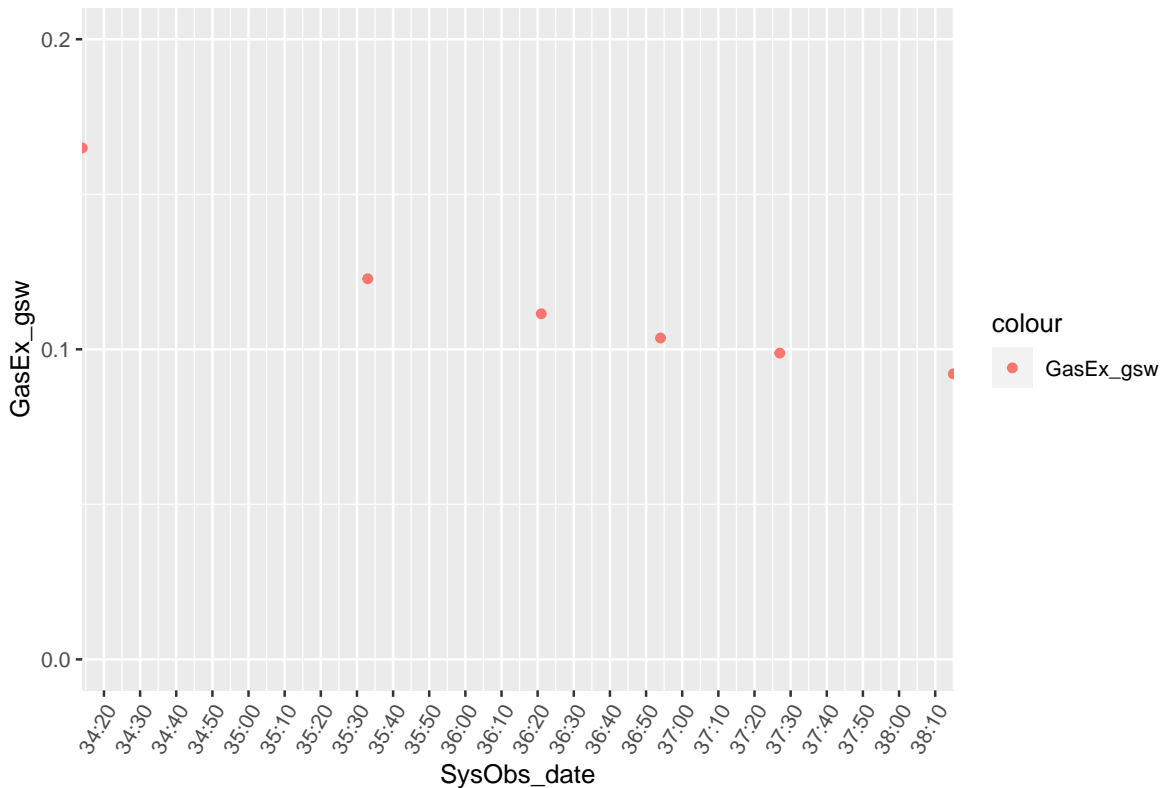
```
#
# scale_x_datetime(labels=date_format("%b %y"))
# date_breaks = "6 hours", expand = expansion(0)) +
# theme(axis.text.x=element_text(angle=60, hjust=1))
```

```

ggplot() +
  # geom_point(data = final_data,
  geom_point(data=subset(final_data, data_group_data_group=="group3"),
    aes(x = SysObs_date, y = GasEx_gsw, color = "GasEx_gsw")) +
  # geom_line(data = final_data, aes(x = SysObs_date, y = GasEx_gsw, color = "GasEx_gsw")) +
  # geom_line(data = final_data, aes(x = date, y = temperature, color = sensor_id)) +
  # geom_line(data = final_data, aes(x = date, y = temperature, color = sensor_id)) +
  # geom_line(data = final_data, aes(x = date, y = TEMP_THREE_FOOT_HEIGHT, color = "CEAC")) +
  ggtitle("LICOR Test Graph - GasEx gsw - TREATMENT LEAF") +
  xlab("SysObs_date") +
  ylab("GasEx_gsw") +
  scale_y_continuous(limits=c(0.0, 0.2), breaks=c(0, 0.1, 0.2, 0.5)) +
  scale_x_datetime(date_labels = ("%M:%S"),
    date_breaks = "10 sec", expand = expansion(0)) +
  theme(axis.text.x=element_text(angle=60, hjust=1))

```

LICOR Test Graph – GasEx gsw – TREATMENT LEAF



```
#  
  
# scale_x_datetime(labels=date_format("%b %y"))  
#   date_breaks = "6 hours",   expand = expansion(0)) +  
  
# theme(axis.text.x=element_text(angle=60, hjust=1))
```