Overview

'

Goal

My current goal is to simply write a script to automate the process of pollutant load modelling by fetching the data (in our case simulated data from Googlesheets), converting into data frame, classifying each day load as OK, Alert based on Minnesota standards and saving output as a CSV in the Google Sheet tab for maintaining records and sending an email notification to the supposed Manager everyday with very simple details of that day.

Near Future goal : Later, I do want train and build a Random Forest model which has shown to outperform standard Flux32 pollutant load estimation to improve load concentrate predictions and its especially useful as it can be imputed through KNN to handle missing values when we do not have enough data and when some data is missing.

Units of Observation

Daily Water Quality conditions at one location

Column Description

Flow_Rate_cms : River discharge (m^3/s); higher = more water volume.
Water_Temp_C : Water temperature (°C).
pH : Acidity/basicity; 7 = neutral.
Specific_Conductance_uScm : Dissolved ions; higher = more salts (µS/cm).
Turbidity_NTU : Cloudiness; higher = more suspended sediment (NTU).
TSS_mg_L : Total Suspended Solids (mg/L).
TP_mg_L : Total Phosphorus (mg/L).
DOP_mg_L : Dissolved Organic Phosphorus (mg/L).
Nitrate_N_mg_L : Nitrate as N (mg/L).
DO_mg_L : Dissolved Oxygen (mg/L).
Flag_Notes : Status text (“OK” / “ALERT”) after comparing to standards.

'

```
# Installing packages
library(tidyverse)
library(googlesheets4)
library(dplyr)
library(cronR)
library(mailR)
library(blastula)
library(glue)


#Authenticating
gs4_deauth()

# reading the sheet and inserting  data
df <-
read_sheet("https://docs.google.com/spreadsheets/d/1PH5RPoDfx5ONpsVGl0ixRvbsvo9PQnoyNvMhRIZkzXw/edit?gid=1688413609#gid=1688413609")
dim(df)
df %>%
  head()



gs4_auth()
sheet_id = "1PH5RPoDfx5ONpsVGl0ixRvbsvo9PQnoyNvMhRIZkzXw"

# Writing to sheets file using the sheet_write() function that takes the sheet id as argument and able to write sheets and manipulate etc



 # Calculating the load_concentrate and adding it to the df which'll be our resonse variable
```

```
MN_TSS_std  <- 65    # mg/L
MN_TP_std   <- 0.50  # mg/L  (or lower if you want to be stricter)
MN_NO3_std  <- 10    # mg/L
MN_DO_min   <- 5     # mg/L  # minimum, so reverse logic
MN_Turb_std <- 5     # NTU   # example

df <- df %>%
  mutate(
    Volume_L_day = Flow_Rate_cms * 86400 * 1000,
    load_concentrate_kg_day = TSS_mg_L * Volume_L_day * 1e-6,

    Flag_Notes = case_when(                                  # using case when for multiple conditionals
      TSS_mg_L   > MN_TSS_std  ~ "ALERT: high TSS",
      TP_mg_L    > MN_TP_std   ~ "ALERT: high TP",
      Nitrate_N_mg_L > MN_NO3_std ~ "ALERT: high nitrate",
      DO_mg_L    < MN_DO_min   ~ "ALERT: low oxygen",
      Turbidity_NTU > MN_Turb_std ~ "ALERT: high turbidity",
      TRUE ~ "OK"
    )
  ) %>%
  relocate(Volume_L_day, load_concentrate_kg_day, .before = Flag_Notes) # I want to see the
Flag_Notes variable at the end



df %>%
  sheet_write(ss = sheet_id, sheet = "Load Modelling") # Writing the cleaned df to Load Modelling tab
in the Google sheet

view(df)

# Setting up the email and authenticating it


# Reading just today's data
gs4_auth()
df
```

```
today_row <- df %>%
  filter(Date == Sys.Date())



# The email body with water date for that day and Flag_Note ("Ok", "Alert") for turbidity, nitrate,
TSS etc
flag_text <- paste0(
  "Daily Water Quality Alert for ", Sys.Date(), "\n\n",
  "Flag_Notes: ", today_row$Flag_Notes[1]
)

smtp_password <- Sys.getenv("SMTP_PASS") # Stored credentials safely in the R_environment

send.mail(
  from = "smohamm2@macalester.edu",
  to   = "smohamm2@macalester.edu",
  subject = "Water Quality Alert",
  body = flag_text,
  smtp = list(
    host.name = "smtp.gmail.com",
    port = 587,
    user.name = "smohamm2@macalester.edu",
    passwd = smtp_password,
    tls = TRUE
  ),
  authenticate = TRUE,
  send = TRUE
)



# Automating the script with cro
script_path <- "../Automating_with_RScript/send_daily_flag_Notes.R"
cmd <- cron_rscript("send_daily_flag_Notes.R")
cron_add(command = cmd, frequency = 'daily', at = "07:00", id = 'daily_flag_email')
```