SWEMED - Logbook

1. **Imports** - import all necessary libraries.

```
### IMPORTS
import requests
import shutil
import sys
import pandas as pd
from datetime import datetime, date, timedelta

# allows access to the website
import urllib3
urllib3.disable_warnings()

# allows crontab to access the file
import os
script_dir = os.path.dirname(os.path.realpath(__file__))
os.chdir(script_dir)

# send email
import smtplib
```

2. **Data Directory** - defines the website where the data are (from where the download will be made) and the directory where the data and log files will be saved.

```
#########################################################
# Data directory
WWWdir = 'https://www.mat.uc.pt/~obsv/SWE-MED/MINwdc/'
MOVEto = '/home/filipa/PycharmProjects/swemed/'
```

3. **Email details** - set the server for the sender email account as well as the email and the password for the login.

```
#########################################################
# Set details to send email
smtp = smtplib.SMTP('smtp.gmail.com', 587)
smtp.ehlo()
smtp.starttls()
smtp.login('sender@sever.com', 'password')   # Login
```

4. **Download SWEMED data**[1] - download the two last files (data from yesterday and today).

```
#########################################################
# Download SWEMED data
if len(sys.argv) == 1:  # is no option is given gets last
    OPTION = 'L2'       # L2 --> download the last two files
else:
    OPTION = sys.argv[1]

out = requests.get(WWWdir, verify=False)
outT = str(out.text).split('href="')

lCmin = []
for el in outT:
    file = el.split('">')[0]
    if file[0:3] == 'coi' and file[-4:] == '.min':
        lCmin.append(file)

if OPTION[0].upper() == 'L':
    l2get = [lCmin[-1]]

    if OPTION.upper() != 'L' or OPTION.upper() != 'L1':
        if OPTION[1:].isnumeric():
            npt = int(OPTION[1:])
            i_end = max([len(lCmin) - npt - 1, -1])
            for k in range(len(lCmin) - 2, i_end, -1):
                l2get.append(lCmin[k])

for li in l2get:
    if li in lCmin:
        url = WWWdir + li
        r = requests.get(url, allow_redirects=True, verify=False)
        open(li, 'wb').write(r.content)
        shutil.move(li, MOVEto + li)
    else:
        print("WARNING: ", li, " not FOUND!!!")
```

5. **Current date/time** - Set the current date and time (UTC).

```
#########################################################
# current date/time
now = datetime.utcnow()
current_time = now.strftime("%H:%M:%S")
today = date.today()
```

[1] by F.Pinheiro

6. **Data files** - using the pandas' package for Python, read the files downloaded, setting the header size and the delimiter. This allows easy separation of each column of data.

```
#########################################################
# data files
dataT = pd.read_csv(l2get[0], header=12, delimiter=r"\s+") # today's
file
dataY = pd.read_csv(l2get[1], header=12, delimiter=r"\s+") # yesterday's
file
```

7. **Create log file** - creates a log file with today's date on the name.

```
#########################################################
# create log file
log_file_name = str(today) + '_out' + '.log'  # file name with the date
it's created
log_file = open(log_file_name, 'a+')
log_file.close()
```

7.1. Creates a function to write on the file: if the error message already exists do not duplicate it.

```
def Write_file(text):
   '''function to open file and write in it'''
   with open(log_file_name, mode='r+') as file:
       for line in file:
           if text in line:
               break
       else:
           file.write(text)
```

8. **Verifying missing points** - Function to verify the file and see if lines are missing, i.e., if it's lacking some data entries, by looking if the interval between two consecutive entries is different than 1 min. If so, it will write the error message in the log file. Every 4h it also writes in the log file that the data has been verified.

```
#########################################################
# Verifying missing points
def Missing(xdata, size):
    day = xdata.get('DATE')[0]
    time_index = pd.DatetimeIndex(xdata['TIME'])
    time = xdata.get('TIME')
    mail = []
    i = 0
    while i < size:
        dif_min = pd.Timedelta('0 days 00:01:00')
        dif = time_index[i + 1] - time_index[i]
        if dif != dif_min:
            sms = f'{time[i]} and {time[i + 1]}'
            Write_file(f'{day} missing data between {sms} \n')
            mail.append(f'{day} missing data between {sms}')
        i += 1
    else:
        if (int(now.hour) % 4) == 0 and now.minute == 0:
            Write_file(f'Last verified at {today} at {current_time} \n')
    return mail
```

9. **Find data error** - Function that searches for each column of data if there are any entries with errors. If so, it will write the error message in the log file.

```
#########################################################
# Find data error
def Analyze(xdata,size):
    day = xdata.get('DATE')[0]
    time = xdata.get('TIME')
    coih, coid, coiz, coif = xdata.get('COIH'), xdata.get('COID'),
xdata.get('COIZ'), xdata.get('COIF')
    mail = []
    for i in size:
        if coih[i] == 999999999.9 or coid[i] == 999999999.9 or coiz[i] ==
999999999.9 or coif[i] == 99999.9:
            sms = f'{day} Data failed at {time[i]} \n'
            Write_file(sms)
            mail.append(sms)
    return mail
```

10. **Analyze yesterday's file and send email** - applies the 'Missing' and 'Analyze' functions to yesterday's file, but only until 2 am today to guarantee the file is all checked out. If it finds some error or missing data, it will automatically send an email with the file's name to check and a list of warning messages.

```
#######################################################
# Analyze yesterday's file and send email
y = []
yy = []
if int(now.hour) <= 2 and now.minute == 0:
   y.append(Missing(dataY, (len(dataY)-1)))      # fetch missing points
on yesterday's file
   yy.append(Analyze(dataY, (len(dataY)-1)))

if len(y) != 0:
   msg = f"""\
Subject: SWEMED DATA ALERT

WARNING --> Missing Data. Please check "{log_file_name}" file.

List of missing data:
{y}
"""

if len(yy) != 0:
   msg = f"""\
Subject: SWEMED DATA ALERT

WARNING --> Data with error. Please check "{log_file_name}" file.

List of wrong values:
{yy}
"""
```

11. **Analyze today's file and send an email** - Applies the 'Missing' function to today's file. If it finds some missing data, it will automatically send an email with the logfile's name to check and the list of missing data.

```
#########################################################
# Analyze today's file and send email

# fetch missing points on today's file
t = Missing(dataT, len(dataT)-1)

if len(t) != 0:
   msg = f"""\
Subject: SWEMED DATA ALERT

WARNING --> Missing Data. Please check "{log_file_name}" file.

List of missing data:
{t}
"""
```

11.1. Define de instant (t-15) min so that the file is analyzed only up to 15 min before the current moment. The 'Analyze' function is then applied until the defined time. Again, if some errors are found, it will automatically send an email with the logfile's name to check and the list of errors.

```
# set the instant (t-15)min
time = dataT.get('TIME')
time_index = pd.DatetimeIndex(dataT['TIME'])
time15min = now - timedelta(minutes=15)
for el, i in enumerate(time_index.strftime("%H:%M")):
   if i == time15min.strftime("%H:%M"):
       now_index = el
a = list(range(0,(now_index + 1)))

# fetch wrong data values on today's file
tt = Analyze(dataT,a)

if len(tt) != 0:
   msg = f"""\
Subject: SWEMED DATA ALERT

WARNING --> Data with error. Please check "{log_file_name}" file.

List of wrong values:
{tt}
"""
```

12. **Send email** - Set the receiver's and sender's emails and closes the connection to the email server.

```
#########################################################
# sending email
to = ["receiver@email.com"]
smtp.sendmail(from_addr="filipafrancisco24@gmail.com",
              to_addrs=to, msg=msg)
smtp.quit() # Close the connection to gmail
```

➔ Run automatically:
  ➔ Linux terminal: `crontab -e`
  ➔ Inside crontab: (runs at every 15 minutes)

```
*/15 * * * * python3 /home/filipa/PycharmProjects/swemed/swemed_data.py
```

➔ Side notes:
  ➔ To test the script, the easiest way is substituting, on line 69, `now = datetime.utcnow()` for `now = datetime.now()`

Email example:

```
WARNING --> Data with error. Please check "2023-04-06_out.log" file.

List of wrong values:
['2023-04-06 Data failed at 09:10:00.000 \n', '2023-04-06 Data failed at
09:11:00.000 \n', '2023-04-06 Data failed at 09:12:00.000 \n',
'2023-04-06 Data failed at 09:13:00.000 \n', '2023-04-06 Data failed at
09:14:00.000 \n', '2023-04-06 Data failed at 09:15:00.000 \n',
'2023-04-06 Data failed at 09:16:00.000 \n', '2023-04-06 Data failed at
09:17:00.000 \n', '2023-04-06 Data failed at 09:18:00.000 \n',
'2023-04-06 Data failed at 09:19:00.000 \n', '2023-04-06 Data failed at
09:20:00.000 \n', '2023-04-06 Data failed at 09:21:00.000 \n',
'2023-04-06 Data failed at 09:22:00.000 \n', '2023-04-06 Data failed at
09:23:00.000 \n', '2023-04-06 Data failed at 09:24:00.000 \n',
'2023-04-06 Data failed at 09:25:00.000 \n', '2023-04-06 Data failed at
09:26:00.000 \n', '2023-04-06 Data failed at 09:27:00.000 \n',
'2023-04-06 Data failed at 09:28:00.000 \n', '2023-04-06 Data failed at
09:29:00.000 \n', '2023-04-06 Data failed at 09:30:00.000 \n',
'2023-04-06 Data failed at 09:31:00.000 \n', '2023-04-06 Data failed at
09:32:00.000 \n', '2023-04-06 Data failed at 09:33:00.000 \n',
'2023-04-06 Data failed at 09:34:00.000 \n', '2023-04-06 Data failed at
09:35:00.000 \n', '2023-04-06 Data failed at 09:36:00.000 \n']
```

logfile example:

```
Last verified at 2023-04-06 at 00:00:02
Last verified at 2023-04-06 at 04:00:03
Last verified at 2023-04-06 at 08:00:02
2023-04-06 Data failed at 09:10:00.000
2023-04-06 Data failed at 09:11:00.000
2023-04-06 Data failed at 09:12:00.000
2023-04-06 Data failed at 09:13:00.000
2023-04-06 Data failed at 09:14:00.000
2023-04-06 Data failed at 09:15:00.000
2023-04-06 Data failed at 09:16:00.000
2023-04-06 Data failed at 09:17:00.000
2023-04-06 Data failed at 09:18:00.000
2023-04-06 Data failed at 09:19:00.000
2023-04-06 Data failed at 09:20:00.000
```