```
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""
##JUSTIN LAU VAN ALLEN PROBES MAX. AE VALUE LOCATOR & PLOTTER
##INPUT:
## (1) AE PEAK DATA OF SATELLITE(.xlsx)
## (2) MULTIPLE NOSE EVENTS; DATE AND TIME(.xlsx)
### EACH DAY = 288 RECORDS
### EACH HOUR = 12 RECORDS
#libraries required
import numpy as np
import matplotlib.pyplot as plt
import xlrd
loc = ("/Users/jlau3/Desktop/PythonMultNoses.xlsx"); #input 1 source
wb = xlrd.open_workbook(loc);
sheet = wb.sheet_by_index(0)
sheet.cell_value(0,0);
loc2 = ("/Users/jlau3/Desktop/aevalues.xlsx"); #input 2 source
wb2 = xlrd.open_workbook(loc2)
sheet2 = wb2.sheet_by_index(0)
sheet2.cell_value(0,0);
file = open("maxBzValues.txt", "w+") #create output file
file.close();
aevalues = []; #store ae
multipletimes = []; #store event
x = 0; #index tracker for excel column(month)
e = 0; #12 hr. before start month counter
f = 0; #24 hr. before start month counter
eventcount = 0; #keeps track of event number
twelvehrb = 0; #equates to previous 12 hour event time
twentyfourhrb = 0; #equates to previous 24 hour event time
currentdate = ""; #current event details
currentday = 0;
currenthour = 0;
currentyear = 0;
startingrecordindex = 1; #records start at 1
twelvehrrecord = 1;
twentyfourhrrecord = 1;
remainhours = 0; #for diving into previous month data
remainrecords = 0;
```

```
twelvehrarray = [];
twentyfourhrarray = [];
markrecord12 = 0; #for calculating the max ae dates
markrecord24 = 0;
markarray = []; #date arrays
markarray2 = [];
markreference12 = 0;
markreference24 = 0;
maxAevalue = -1e6; #a value threshold
maxAevalue2 = -1e6;
startyear = 2013; #user input start year
startingmonthaccounttwelve = []; #store previous month values
startingmonthaccounttwentyfour = [];
plotAevalues = []; #store all max. ae values for plot (12 HR.)
plotAevalues24 = []; #(24 HR.)
plottimediff = []; #store calculated time differences for plot (12 HR.)
plottimediff24 = []; #(24 HR.)
addhourdiff = 0; #hour difference
addmaxdiff = 0; #time difference
addtimediff = 0; #total time difference(decimal hour)
#ae barchart vars.
bar0 = 0; #0-50
bar100 = 0; #50-150
bar200 = 0; #150-250
bar300 = 0; #250-350
bar400 = 0; #350-450
bar500 = 0; #450-550
bar600 = 0; #550-650
bar700 = 0; #650-750
bar800 = 0; #750-850
bar900 = 0; #850-950
bar1000 = 0; #950-1050
bar1100 = 0; #1050-1150
bar1200 = 0; #1150-1250
barover = 0; #>1250
barchartae = [];
#Time diff barchart vars.
bar0hour = 0; #0 to 0.5
bar1hour = 0; #0.5 to 1.5
bar2hour = 0; #1.5 to 2.5
bar3hour = 0; #2.5 to 3.5
bar4hour = 0; #3.5 to 4.5
bar5hour = 0; #4.5 to 5.5
```

```
bar6hour = 0; #...
bar7hour = 0;
bar8hour = 0;
bar9hour = 0;
bar10hour = 0;
bar11hour = 0;
bar12hour = 0;
bar13hour = 0;
bar14hour = 0;
bar15hour = 0;
bar16hour = 0;
bar17hour = 0;
bar18hour = 0;
bar19hour = 0;
bar20hour = 0;
bar21hour = 0;
bar22hour = 0;
bar23hour = 0;
bar24hour = 0; #23.5 to 24
barcharttd = [];

#stores all AE values(each column corresponds to month **22 MONTHS**)
for addae in range(sheet2.nrows):
    aevalues.append(sheet2.row_values(addae));
testy = np.array(aevalues); #Numpy version of aevalue array

#stores multiple noses dates and times
for addevent in range(sheet.nrows):
    multipletimes.append(sheet.row_values(addevent));
for i in multipletimes: #loops through events
    currentdate = str(multipletimes[eventcount][0]); #gets the event date
    currentmonth = int(currentdate[4:6]); #substring and gets the event month
    currentyear = int(currentdate[0:4]); #substring and gets the event year
    currentday = int(currentdate[6:8])-1; #substring and gets the event day
    currenthour = int(multipletimes[eventcount][1]*24); #converts decimal hour to integer hour
    startingrecordindex = (currentday*288)+1; #calculates reference record point for event
    startingrecordindex += (currenthour*12);

    if(currentyear != startyear): #account for shift of year
        currentmonth += 12;
    while(currentmonth != (x+1)): #account for shift of month
        x += 1;
```

```
   twentyfourhrb = int(currenthour); #calculate hours before event
   if(currenthour < 12):
      twelvehrb = currenthour + 12;
   elif(currenthour >= 12):
      twelvehrb = currenthour - 12;
   twelvehrrecord = ((startingrecordindex - 288) + 144); #calculates reference record point for
both previous hours
   twentyfourhrrecord = ((startingrecordindex - 288));

   if(currentday == 0 and currenthour < 12): #account for 12 hrs. before start of month
      remainhours = 12 - currenthour;
      remainrecords = remainhours * 12;
      for d in aevalues:
         if(aevalues[e][x-1] == ""):
            break;
         e += 1;
      remainrecords = e - remainrecords;
      for p in range(remainrecords, e):
         startingmonthaccounttwelve.append(aevalues[p][x-1]); #stores previous month values
      twelvehrrecord = 0; #set new starting 12 hr. record point

   if(currentday == 0 and currenthour < 24): #account for 24 hrs. before start of month
      remainhours = 24 - currenthour;
      remainrecords = remainhours * 12;
      for dd in aevalues:
         if(aevalues[f][x-1] == ""):
            break;
         f += 1;
      remainrecords = f - remainrecords;
      for pp in range(remainrecords, f):
         startingmonthaccounttwentyfour.append(aevalues[pp][x-1]); #stores previous month
values
      twentyfourhrrecord = 0; #set new starting 24 hr. record point

   #twelvehourAEvalue
   for a in range(twelvehrrecord, startingrecordindex): #find max. ae value within 12 hours
before event
      twelvehrarray.append(aevalues[a][x]);
   for b in twelvehrarray: #main values
      if(b > maxAevalue):
         if(b != 9999.99): #prevent skewing of data set
            maxAevalue = b;
   for c in startingmonthaccounttwelve: #account for start of month events
```

```
        if(c > maxAevalue):
            if(c != 9999.99):
                maxAevalue = c;
    if(maxAevalue == -1e6): #account for case of no valid data
        maxAevalue = 9999.99;
    plotAevalues.append(maxAevalue);

    #retrieve date index of max. ae value occurrence 12 hours before event
    markarray = []; #resets date arrays; infinite loop occurs if not
    markarray2 = [];
    numappearances = 0;
    sliced = testy[remainrecords-1:f, x-1]; #isolates previous month ae values in case
    counter = np.count_nonzero(np.where(sliced == str(maxAevalue))); #count if max. ae value
occurs in multiple instances
    if(counter != 0):
        markrecord12 = (np.where(sliced == str(maxAevalue))[0][counter-1]) + remainrecords + 1;
#sets to most recent index of max ae. value
    for g in range(twelvehrrecord, startingrecordindex): #gets index of max. ae value for current
month if not found in a previous month
        markarray.append(aevalues[g][x]); #all ae values of current month
    xxx = np.array(markarray); #numpy version
    numappearances = np.count_nonzero(xxx == maxAevalue);
    if(numappearances != 0):
        markrecord12 = (np.where(xxx == maxAevalue)[0][numappearances-1]) +1
+twelvehrrecord;

    #calculates the 12 hr. date parameters needed to be added
    markreference12 = markrecord12;
    markrecord12 = (markrecord12 - 1) / 12;
    decimalpart = markrecord12 - int(markrecord12);
    addmax = int(round(decimalpart * 60));
    markrecord12 = int(markrecord12);
    markrecord12 = markrecord12 / 24;
    addday = int(markrecord12);
    decimalpart2 = markrecord12 - int(markrecord12);
    addhour = int(round(decimalpart2 * 24));

    #perform operations to get the date to print(12 hour)
    manipdate12 = currentdate;
    manipdate12 = float(manipdate12);
    manipdate12 = int(manipdate12);
    addhour = int(addhour);
    if(currentday == 0 and markreference12 > startingrecordindex):
        manipdate12 -= 100;
        manipdate12 += addday;
        if(addhour > currenthour):
            manipdate12 += 1;

    #formatting fixes
    if(addhour < 10):
        addhour = str(addhour);
        addhour = ("0" + addhour);
    if(addmax < 10):
        addmax = str(addmax);
        addmax = ("0"+addmax);
    if(int(addhour) > currenthour):
        manipdate12 -= 1;
    if(int(addhour) == currenthour):
        if(int(addmax) > 0):
            manipdate12 -= 1;

    #calculate time difference (between event time and ae value time) (12 hours before) in
decimal hours
    if(int(currenthour) > int(addhour) and int(addmax) > 0): #case 1
        addhourdiff = int(currenthour) - int(addhour) - 1;
        addmaxdiff = 60 - int(addmax);
        addmaxdiff = round(addmaxdiff / 60 , 2);
        addtimediff = addhourdiff + addmaxdiff;
    if(int(currenthour) > int(addhour) and int(addmax) == 0): #case 2
        addhourdiff = int(currenthour) - int(addhour);
        addtimediff = addhourdiff;
    if(int(currenthour) < int(addhour) and int(addmax) > 0): #case 3
        addhourdiff = 24 - int(addhour) + int(currenthour) - 1;
        addmaxdiff = 60 - int(addmax);
        addmaxdiff = round(addmaxdiff / 60 , 2);
        addtimediff = addhourdiff + addmaxdiff;
    if(int(currenthour) < int(addhour) and int(addmax) == 0): #case 4
        addhourdiff = 24 - int(addhour) + int(currenthour);
        addtimediff = addhourdiff;
    if(int(currenthour) == int(addhour) and int(addmax) > 0): #case 5
        addhourdiff = 23;
        addmaxdiff = 60 - int(addmax);
        addmaxdiff = round(addmaxdiff / 60, 2);
        addtimediff = addhourdiff + addmaxdiff;
    if(int(currenthour) == int(addhour) and int(addmax) == 0): #case 6
        addhourdiff = 0;
        addtimediff = addhourdiff;
    plottimediff.append(addtimediff); #store all time differences for every ae value

    #write 12 hr. max. ae value to text file
    file = open("maxBzValues.txt", "a+");
    for w in range(1):
        file.write("EVENT DATE: " + str(currentdate) + " EVENT TIME: " + str(currenthour) + ":00 |||
(12) DATE: " + str(manipdate12) + ", TIME: " + str(addhour) + ":" + str(addmax) + ", MAX.
VALUE: " +  str(maxAevalue));
    file.close();

    #twentyfourhourBZvalue
    for aa in range(twentyfourhrrecord, startingrecordindex): #find max ae value within 24 hours
before event
        twentyfourhrarray.append(aevalues[aa][x]);
    for bb in twentyfourhrarray: #main values
        if(bb > maxAevalue2):
            if(bb != 9999.99):
                maxAevalue2 = bb;
    for cc in startingmonthaccounttwentyfour: #account for start of month events
        if(cc > maxAevalue2):
            if(cc != 9999.99):
                maxAevalue2 = cc;
    if(maxAevalue2 == -1e6):
        maxAevalue2 = 9999.99;
    plotAevalues24.append(maxAevalue2);

    #24 hr. date retrieval
    sliced = testy[remainrecords-1 :f, x-1];
    counter = np.count_nonzero(np.where(sliced == str(maxAevalue2)));
    if(counter != 0):
        markrecord24 = (np.where(sliced == str(maxAevalue2))[0][counter-1]) + remainrecords + 1;

    for g in range(0, startingrecordindex): #gets index of max ae value
        markarray2.append(aevalues[g][x]);
    xxx2 = np.array(markarray2);
    numappearances = np.count_nonzero(np.where(xxx2 == maxAevalue2));
    if(numappearances != 0):
        markrecord24 = (np.where(xxx2 == maxAevalue2)[0][numappearances-1]) +1;

    #calculate 24 hr. date parameters
    markreference24 = markrecord24;
    markrecord24 = (markrecord24 - 1) / 12;
```

```
decimalpart1 = markrecord24 - int(markrecord24);
addmax2 = int(round(decimalpart1 * 60));
markrecord24 = int(markrecord24);
markrecord24 = markrecord24 / 24;
addday2 = int(markrecord24);
decimalpart12 = markrecord24 - int(markrecord24);
addhour2 = int(round(decimalpart12 * 24));

#performs operations to get the date to print(24 hour)
manipdate24 = currentdate;
manipdate24 = float(manipdate24);
manipdate24 = int(manipdate24);
if(currentday == 0 and markreference24 > startingrecordindex):
    manipdate24 -= 100;
    manipdate24 += addday2
    if(addhour2 > currenthour):
            manipdate24 += 1;

#formatting corrections
if(addhour2 < 10):
    addhour2 = str(addhour2);
    addhour2 = ("0" + addhour2);
if(addmax2 < 10):
    addmax2 = str(addmax2);
    addmax2 = ("0"+addmax2);
if(int(addhour2) > currenthour):
    manipdate24 -= 1;
if(int(addhour2) == currenthour):
    if(int(addmax2) > 0):
        manipdate24 -= 1;

#calculate time difference (24) in decimal hours
if(int(currenthour) > int(addhour2) and int(addmax2) > 0): #case 1
    addhourdiff = int(currenthour) - int(addhour2) - 1;
    addmaxdiff = 60 - int(addmax2);
    addmaxdiff = round(addmaxdiff / 60 , 2);
    addtimediff = addhourdiff + addmaxdiff;
if(int(currenthour) > int(addhour2) and int(addmax2) == 0): #case 2
    addhourdiff = int(currenthour) - int(addhour2);
    addtimediff = addhourdiff;
if(int(currenthour) < int(addhour2) and int(addmax2) > 0): #case 3
    addhourdiff = 24 - int(addhour2) + int(currenthour) - 1;
    addmaxdiff = 60 - int(addmax2);
```

```
        addmaxdiff = round(addmaxdiff / 60 , 2);
        addtimediff = addhourdiff + addmaxdiff;
    if(int(currenthour) < int(addhour2) and int(addmax2) == 0): #case 4
        addhourdiff = 24 - int(addhour2) + int(currenthour);
        addtimediff = addhourdiff;
    if(int(currenthour) == int(addhour2) and int(addmax2) > 0): #case 5
        addhourdiff = 23;
        addmaxdiff = 60 - int(addmax2);
        addmaxdiff = round(addmaxdiff / 60, 2);
        addtimediff = addhourdiff + addmaxdiff;
    if(int(currenthour) == int(addhour2) and int(addmax2) == 0): #case 6
        addhourdiff = 24;
        addtimediff = addhourdiff;
    plottimediff24.append(addtimediff); #store all the time differences for every ae value

    #write 24 hr. max. ae value to text file
    file = open("maxBzValues.txt", "a+");
    for w in range(1):
        file.write(" ||| (24) DATE: " + str(manipdate24) + ", TIME: " + str(addhour2) + ":" +
str(addmax2) + ", MAX. VALUE: " +  str(maxAevalue2) + " \n \n");
    file.close();

    #reset values for next event run
    maxAevalue = -1e6;
    maxAevalue2 = -1e6;
    twelvehrarray = [];
    twentyfourhrarray = [];
    startingmonthaccounttwelve = [];
    startingmonthaccounttwentyfour = [];
    e = 0;
    f = 0;
    eventcount += 1;
    markrecord12 = 0;
    markrecord24 = 0;
    ###end of run

#creates scatter plot for data
#
=====================================================================
=====
# plt.plot(plottimediff, plotAevalues, ".b" , label = "data");
# plt.grid();
# plt.title("AE Value v.s. Time Difference (12 hr.)", fontsize = 18);
# plt.xlabel("Difference between Event Time and AE Time(Decimal Hours)" , fontsize = 12);
# plt.ylabel("AE Value" , fontsize = 12);
# plt.legend();
# plt.show();
# plt.savefig('scatter.png');
#
=============================================================================
=====

#
=============================================================================
=====
# plt.plot(plottimediff24, plotAevalues24, ".b" , label = "data");
# plt.grid();
# plt.title("AE Value v.s. Time Difference (24 hr.)", fontsize = 18);
# plt.xlabel("Difference between Event Time and AE Time(Decimal Hours)" , fontsize = 12);
# plt.ylabel("AE Value" , fontsize = 12);
# plt.legend();
# plt.show();
# plt.savefig('scatter.png');
#
=============================================================================
=====
# plt.margins(0); #utilize to set margins, zoom in, or zoom out

#separates values for ae barchart

#barchartae = plotAevalues; #12 HR.
#barchartae = plotAevalues24; #24 HR.

for bar in barchartae:
    if(bar < 50):
        bar0 += 1;
    elif(bar >= 50 and bar < 150):
        bar100 += 1;
    elif(bar >= 150 and bar < 250):
        bar200 += 1;
    elif(bar >= 250 and bar < 350):
        bar300 += 1;
    elif(bar >= 350 and bar < 450):
        bar400 += 1;
    elif(bar >= 450 and bar < 550):
        bar500 += 1;
    elif(bar >= 550 and bar < 650):
        bar600 += 1;
    elif(bar >= 650 and bar < 750):
        bar700 += 1;
    elif(bar >= 750 and bar < 850):
        bar800 += 1;
    elif(bar >= 850 and bar < 950):
        bar900 += 1;
    elif(bar >= 950 and bar < 1050):
        bar1000 += 1;
    elif(bar >= 1050 and bar < 1150):
        bar1100 += 1;
    elif(bar >= 1150 and bar < 1250):
        bar1200 += 1;
    elif(bar > 1250):
        barover += 1;

#creates bar chart for # of events v.s. ae value
#
============================================================================
=====
# bzvalue = ('0', '100', '200', '300', '400', '500', '600', '700', '800', '900', '1000', '1100', '1200',
'1200+');
# numevents = np.arange(len(bzvalue))
# stats = [bar0, bar100, bar200, bar300, bar400, bar500, bar600, bar700, bar800, bar900,
bar1000, bar1100, bar1200, barover];
# plt.bar(numevents, stats, align='center', alpha=0.5)
# plt.xticks(numevents, bzvalue);
# plt.ylabel('# of events')
# plt.xlabel('AE Value');
# plt.title('# of Events v.s. AE Value');
# plt.grid();
# plt.show();
#
============================================================================
=====

#separates values for time diff. barchart

#barcharttd = plottimediff; #12 HOUR BAR CHART
#barcharttd = plottimediff24; #24 HOUR BAR CHART

for bar in barcharttd:
```

```
if(bar >= 0 and bar < 0.5):
    bar0hour += 1;
elif(bar >= 0.5 and bar < 1.5):
    bar1hour += 1;
elif(bar >= 1.5 and bar < 2.5):
    bar2hour += 1;
elif(bar >= 2.5 and bar < 3.5):
    bar3hour += 1;
elif(bar >= 3.5 and bar < 4.5):
    bar4hour += 1;
elif(bar >= 4.5 and bar < 5.5):
    bar5hour += 1;
elif(bar >= 5.5 and bar < 6.5):
    bar6hour += 1;
elif(bar >= 6.5 and bar < 7.5):
    bar7hour += 1;
elif(bar >= 7.5 and bar < 8.5):
    bar8hour += 1;
elif(bar >= 8.5 and bar < 9.5):
    bar9hour += 1;
elif(bar >= 9.5 and bar < 10.5):
    bar10hour += 1;
elif(bar >= 10.5 and bar < 11.5):
    bar11hour += 1;
elif(bar >= 11.5 and bar < 12.5):
    bar12hour += 1;
elif(bar >= 12.5 and bar < 13.5):
    bar13hour += 1;
elif(bar >= 13.5 and bar < 14.5):
    bar14hour += 1;
elif(bar >= 14.5 and bar < 15.5):
    bar15hour += 1;
elif(bar >= 15.5 and bar < 16.5):
    bar16hour += 1;
elif(bar >= 16.5 and bar < 17.5):
    bar17hour += 1;
elif(bar >= 17.5 and bar < 18.5):
    bar18hour += 1;
elif(bar >= 18.5 and bar < 19.5):
    bar19hour += 1;
elif(bar >= 19.5 and bar < 20.5):
    bar20hour += 1;
elif(bar >= 20.5 and bar < 21.5):
        bar21hour += 1;
    elif(bar >= 21.5 and bar < 22.5):
        bar22hour += 1;
    elif(bar >= 22.5 and bar < 23.5):
        bar23hour += 1;
    elif(bar >= 23.5 and bar < 24.0):
        bar24hour += 1;
#creates bar chart for # of events v.s. Time Difference
#
=============================================================================
=====
# hours = ('0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20',
'21', '22', '23', '24');
# plothours = np.arange(len(hours))
# stats = [bar0hour, bar1hour, bar2hour, bar3hour, bar4hour, bar5hour, bar6hour, bar7hour,
bar8hour, bar9hour, bar10hour, bar11hour, bar12hour, bar13hour, bar14hour, bar15hour,
bar16hour, bar17hour, bar18hour, bar19hour, bar20hour, bar21hour, bar22hour, bar23hour,
bar24hour];
# plt.bar(plothours, stats, align='center', alpha=0.5)
# plt.xticks(plothours, hours);
# plt.ylabel('# of events')
# plt.xlabel('Time Difference(Decimal Hrs.)');
# plt.title('# of Events v.s. Time Difference');
# plt.grid();
# plt.show();
#
=============================================================================
=====
```