

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

Spyder Editor

This is a temporary script file.  
"""

```
##JUSTIN LAU VAN ALLEN PROBES MIN. BZ VALUE LOCATOR & PLOTTER  
##INPUT:
```

```
## (1) BZ 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/22monthsبز.xlsx"); #input 2 source
```

```
wb2 = xlrd.open_workbook(loc2)
```

```
sheet2 = wb2.sheet_by_index(0)
```

```
sheet2.cell_value(0,0);
```

```
file = open("minBzValues.txt", "w+") #create output file
```

```
file.close();
```

```
bzvalues = []; #store Bz
```

```
multiplentimes = []; #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 min Bz dates
markrecord24 = 0;
markarray = []; #date arrays
markarray2 = [];
markreference12 = 0;
markreference24 = 0;
minBzvalue = 1e6; #a value threshold
minBzvalue2 = 1e6;
startyear = 2013; #user input start year
startingmonthaccountttwelve = []; #store previous month values
startingmonthaccounttwentyfour = [];
plotbzvalues = []; #store all min. Bz values for plot
plotbzvalues24 = [];
plottimediff = []; #store calculated time differences for plot
plottimediff24 = [];
addhourdiff = 0; #hour difference
addmindiff = 0; #time difference
addtimediff = 0; #total time difference(decimal hour)
#Bz barchart vars.
barover5 = 0;
barover0 = 0;
barfor0 = 0; #-0.5 to 0.5
barforneg1 = 0; #-1.5 to -0.5
barforneg2 = 0; #-2.5 to -1.5
barforneg3 = 0; #-3.5 to -2.5
barforneg4 = 0; #-4.5 to -3.5
barforneg5 = 0; #-5.5 to -4.5
barforneg6 = 0; #-6.5 to -5.5
barforneg7 = 0; #-7.5 to -6.5
barforneg8 = 0; #-8.5 to -9.5
barforneg9 = 0; #-9.5 to -8.5
barforneg10 = 0; #-10.5 to -9.5
barforneg11 = 0; #-11.5 to -10.5
barforneg12 = 0; #-12.5 to -11.5
barunderneg12 = 0;
barchartbz = [];
#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 BZ values(each column corresponds to month **22 MONTHS**)

```

```

for addbz in range(sheet2.nrows):

```

```

    bzvalues.append(sheet2.row_values(addbz));

```

```

testy = np.array(bzvalues); #Numpy version of bzvalue 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 bzvalues:
        if(bzvalues[e][x-1] == ""):
            break;
        e += 1;
    remainrecords = e - remainrecords;
    for p in range(remainrecords, e):
        startingmonthaccounttwelve.append(bzvalues[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 bzvalues:
        if(bzvalues[f][x-1] == ""):
            break;
        f += 1;
    remainrecords = f - remainrecords;
    for pp in range(remainrecords, f):
        startingmonthaccounttwentyfour.append(bzvalues[pp][x-1]); #stores previous month
values
    twentyfourhrrecord = 0; #set new starting 24 hr. record point

#twelvehourBZvalue
for a in range(twelvehrrecord, startingrecordindex): #find min. Bz value within 12 hours before
event
    twelvehrarray.append(bzvalues[a][x]);
for b in twelvehrarray: #main values
    if(b < minBzvalue):
        minBzvalue = b;

```

```

for c in startingmonthaccounttwelve: #account for start of month events
    if(c < minBzvalue):
        minBzvalue = c;
if(minBzvalue == 9999.99):
    plotbzvalues.append(0);
else:
    plotbzvalues.append(minBzvalue);

#retrieve date index of min. Bz value occurrence 12 hours before event
markarray = []; #resets date arrays; infinite loop occurs if not
markarray2 = [];
numappearances = 0;
sliced = testy[remainrecords:f, x-1]; #isolates previous month Bz values in case
counter = np.count_nonzero(np.where(sliced == str(minBzvalue))); #count if min. Bz value
occurs in multiple instances
if(counter != 0):
    markrecord12 = (np.where(sliced == str(minBzvalue))[0][counter-1]) + remainrecords + 1;
#sets to most recent index of min Bz. value
for g in range(twelvehrrecord, startingrecordindex): #gets index of min. Bz value for current
month if not found in a previous month
    markarray.append(bzvalues[g][x]); #all bz values of current month
xxx = np.array(markarray); #numpy version
numappearances = np.count_nonzero(xxx == minBzvalue);
if(numappearances != 0):
    markrecord12 = (np.where(xxx == minBzvalue)[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);
addmin = 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(addmin < 10):
    addmin = str(addmin);
    addmin = ("0"+addmin);
if(int(addhour) > currenthour):
    manipdate12 -= 1;
if(int(addhour) == currenthour):
    if(int(addmin) > 0):
        manipdate12 -= 1;
```

#calculate time difference (between event time and Bz value time) (12 hours before) in decimal hours

```
if(int(currenthour) > int(addhour) and int(addmin) > 0): #case 1
    addhourdiff = int(currenthour) - int(addhour) - 1;
    addmindiff = 60 - int(addmin);
    addmindiff = round(addmindiff / 60 , 2);
    addtimediff = addhourdiff + addmindiff;
if(int(currenthour) > int(addhour) and int(addmin) == 0): #case 2
    addhourdiff = int(currenthour) - int(addhour);
    addtimediff = addhourdiff;
if(int(currenthour) < int(addhour) and int(addmin) > 0): #case 3
    addhourdiff = 24 - int(addhour) + int(currenthour) - 1;
    addmindiff = 60 - int(addmin);
    addmindiff = round(addmindiff / 60 , 2);
    addtimediff = addhourdiff + addmindiff;
if(int(currenthour) < int(addhour) and int(addmin) == 0): #case 4
    addhourdiff = 24 - int(addhour) + int(currenthour);
    addtimediff = addhourdiff;
if(int(currenthour) == int(addhour) and int(addmin) > 0): #case 5
    addhourdiff = 23;
    addmindiff = 60 - int(addmin);
    addmindiff = round(addmindiff / 60, 2);
    addtimediff = addhourdiff + addmindiff;
if(int(currenthour) == int(addhour) and int(addmin) == 0): #case 6
```

```

    addhourdiff = 0;
    addtimediff = addhourdiff;
    plottimediff.append(addtimediff); #store all time differences for every Bz value

#write 12 hr. min. Bz value to text file
file = open("minBzValues.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(addmin) + ", MIN.
VALUE: " + str(minBzvalue));
    file.close();

#twentyfourhourBZvalue
for aa in range(twentyfourhrrecord, startingrecordindex): #find min Bz value within 24 hours
before event
    twentyfourhrarray.append(bzvalues[aa][x]);
for bb in twentyfourhrarray: #main values
    if(bb < minBzvalue2):
        minBzvalue2 = bb;
for cc in startingmonthaccounttwentyfour: #account for start of month events
    if(cc < minBzvalue2):
        minBzvalue2 = cc;
if(minBzvalue2 == 9999.99):
    plotbzvalues24.append(0);
else:
    plotbzvalues24.append(minBzvalue2);

#24 hr. date retrieval
sliced = testy[remainrecords-1:f, x-1];
counter = np.count_nonzero(np.where(sliced == str(minBzvalue2)));
if(counter != 0):
    markrecord24 = (np.where(sliced == str(minBzvalue2))[0][counter-1]) + remainrecords + 1;

for g in range(0, startingrecordindex): #gets index of min Bz value
    markarray2.append(bzvalues[g][x]);
xxx2 = np.array(markarray2);
numappearances = np.count_nonzero(np.where(xxx2 == minBzvalue2));
if(numappearances != 0):
    markrecord24 = (np.where(xxx2 == minBzvalue2)[0][numappearances-1]) + 1;

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

```

```

decimalpart1 = markrecord24 - int(markrecord24);
addmin2 = 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(addmin2 < 10):
    addmin2 = str(addmin2);
    addmin2 = ("0"+addmin2);
if(int(addhour2) > currenthour):
    manipdate24 -= 1;
if(int(addhour2) == currenthour):
    if(int(addmin2) > 0):
        manipdate24 -= 1;

#calculate time difference (24) in decimal hours
if(int(currenthour) > int(addhour2) and int(addmin2) > 0): #case 1
    addhourdiff = int(currenthour) - int(addhour2) - 1;
    addmindiff = 60 - int(addmin2);
    addmindiff = round(addmindiff / 60 , 2);
    addtimediff = addhourdiff + addmindiff;
if(int(currenthour) > int(addhour2) and int(addmin2) == 0): #case 2
    addhourdiff = int(currenthour) - int(addhour2);
    addtimediff = addhourdiff;
if(int(currenthour) < int(addhour2) and int(addmin2) > 0): #case 3
    addhourdiff = 24 - int(addhour2) + int(currenthour) - 1;
    addmindiff = 60 - int(addmin2);

```



```

        addmindiff = round(addmindiff / 60 , 2);
        addtimediff = addhourdiff + addmindiff;
    if(int(currenthour) < int(addhour2) and int(addmin2) == 0): #case 4
        addhourdiff = 24 - int(addhour2) + int(currenthour);
        addtimediff = addhourdiff;
    if(int(currenthour) == int(addhour2) and int(addmin2) > 0): #case 5
        addhourdiff = 23;
        addmindiff = 60 - int(addmin2);
        addmindiff = round(addmindiff / 60, 2);
        addtimediff = addhourdiff + addmindiff;
    if(int(currenthour) == int(addhour2) and int(addmin2) == 0): #case 6
        addhourdiff = 24;
        addtimediff = addhourdiff;
    plottimediff24.append(addtimediff); #store all the time differences for every Bz value

#write 24 hr. min. Bz value to text file
file = open("minBzValues.txt", "a+");
for w in range(1):
    file.write(" ||| (24) DATE: " + str(manipdate24) + ", TIME: " + str(addhour2) + ":" +
str(addmin2) + ", MIN. VALUE: " + str(minBzvalue2) + " \n \n");
    file.close();

#reset values for next event run
minBzvalue = 1e6;
minBzvalue2 = 1e6;
twelvehrarray = [];
twentyfourhrarray = [];
startingmonthaccountttwelve = [];
startingmonthaccounttwentyfour = [];
e = 0;
f = 0;
eventcount += 1;
markrecord12 = 0;
markrecord24 = 0;
####end of run

#creates scatter plot for data
#
=====
=====
# plt.plot(plottimediff, plotbzvalues, ".b" , label = "data");
# plt.grid();
# plt.title("BZ Value v.s. Time Difference (12 hr.)", fontsize = 18);

```

```

# plt.xlabel("Difference between Event Time and BZ Time(Decimal Hours)" , fontsize = 12);
# plt.ylabel("BZ Value" , fontsize = 12);
# plt.legend();
# plt.show();
# plt.savefig('scatter.png');
#
=====
=====

#
=====
=====

# plt.plot(plottimediff24, plotbzvalues24, ".b" , label = "data");
# plt.grid();
# plt.title("BZ Value v.s. Time Difference (24 hr.)", fontsize = 18);
# plt.xlabel("Difference between Event Time and BZ Time(Decimal Hours)" , fontsize = 12);
# plt.ylabel("BZ 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 Bz barchart
#bchartbz = plotbzvalues; #12 HOUR
#bchartbz = plotbzvalues24; #24 HOUR
for bar in bchartbz:
    if(bar >= 5):
        barover5 += 1;
    elif(bar >= 0.5 and bar < 5):
        barover0 += 1;
    elif(bar > -0.5 and bar < 0.5):
        barfor0 += 1;
    elif(bar > -1.5 and bar <= -0.5):
        barforneg1 += 1;
    elif(bar > -2.5 and bar <= -1.5):
        barforneg2 += 1;
    elif(bar > -3.5 and bar <= -2.5):
        barforneg3 += 1;
    elif(bar > -4.5 and bar <= -3.5):
        barforneg4 += 1;

```

```

elif(bar > -5.5 and bar <= -4.5):
    barforneg5 += 1;
elif(bar > -6.5 and bar <= -5.5):
    barforneg6 += 1;
elif(bar > -7.5 and bar <= -6.5):
    barforneg7 += 1;
elif(bar > -8.5 and bar <= -7.5):
    barforneg8 += 1;
elif(bar > -9.5 and bar <= -8.5):
    barforneg9 += 1;
elif(bar > -10.5 and bar <= -9.5):
    barforneg10 += 1;
elif(bar > -11.5 and bar <= -10.5):
    barforneg11 += 1;
elif(bar > -12.5 and bar <= -11.5):
    barforneg12 += 1;
elif(bar < -12.5):
    barunderneg12 += 1;
#creates bar chart for # of events v.s. Bz value
#
=====
=====
# bzvalue = ('<-12', '-12', '-11', '-10', '-9', '-8', '-7', '-6', '-5', '-4', '-3', '-2', '-1', '0', '0-5', '>5');
# numevents = np.arange(len(bzvalue))
# stats = [barunderneg12, barforneg12, barforneg11, barforneg10, barforneg9, barforneg8,
barforneg7, barforneg6, barforneg5, barforneg4, barforneg3, barforneg2, barforneg1, barfor0,
barover0, barover5];
# plt.bar(numevents, stats, align='center', alpha=0.5)
# plt.xticks(numevents, bzvalue);
# plt.ylabel('# of events')
# plt.xlabel('BZ Value');
# plt.title('# of Events v.s. BZ 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');
# plohours = 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(plohours, stats, align='center', alpha=0.5)
# plt.xticks(plohours, hours);
# plt.ylabel('# of events')
# plt.xlabel('Time Difference(Decimal Hrs.)');
# plt.title('# of Events v.s. Time Difference');
# plt.grid();
# plt.show();
#
=====
=====

```