“””

Company X Data Analyst Challenge:

**Challenge description:**

As part of a GPS testing experiment, we drove a car onto which we had installed 2 GPS devices (at 2 slightly different heights). We obtained 2 streams of data stored in the file to download. These data include the actual coordinates of the car and a few other parameters. These 2 timestamped streams are consistent with each other for the most part, but we want to deal only with one. Note that altitude is not an important focus of this study, and that missing data are represented by empty fields.

The goal of this challenge is:

* to create a single stream out of the 2 available here
* to provide an indicator for how each point was computed

Here are the first few lines, including the header lines, of the original downloaded GPS\_Streams.txt file:

UTC timestamp,heading\_deg:GPS Stream #1,alt:GPS Stream #1,alt:GPS Stream #2,hdop:GPS Stream #1,hdop:GPS Stream #2,lat:GPS Stream #1,lat:GPS Stream #2,lock\_type:GPS Stream #1,lock\_type:GPS Stream #2,lon:GPS Stream #1,lon:GPS Stream #2,number\_sat:GPS Stream #1,number\_sat:GPS Stream #2,time\_since\_lock:GPS Stream #1,time\_since\_lock:GPS Stream #2,true\_speed\_ms:GPS Stream #1,true\_speed\_ms:GPS Stream #2,true\_track\_deg:GPS Stream #1,true\_track\_deg:GPS Stream #2

2014-08-15 03:43:13+00:00,281.7000122,86,91.69995117,2.799999952,0.899999976,36.24509842,36.24515182,GPS\_LOCK,GPS\_LOCK,-116.8257518,-116.8257365,5,9,0,0,0.0514444,0.411555529,249.8999939,317.6700134

2014-08-15 03:43:14+00:00,281.5,86.09997559,92,1.299999952,0.899999976,36.24510223,36.24515182,GPS\_LOCK,GPS\_LOCK,-116.8257518,-116.8257441,7,8,0,0,0.1028888,0.19548887,315.7999878,317.6700134

2014-08-15 03:43:15+00:00,282.1000061,86.19995117,92.19995117,3.799999952,0.899999976,36.24510223,36.24515182,GPS\_LOCK,GPS\_LOCK,-116.8257441,-116.8257518,5,8,0,0,0.1028888,0.272655547,289.5,317.6700134

My attempted partial solutions are as follows:

“””

# Company\_X\_GPS\_Streams\_intermediate.py

import csv

with open("GPS\_Streams.txt\_orig","rb") as source:

rdr= csv.reader( source )

with open("GPS\_Streams\_intermediate\_out.txt","wb") as result:

wtr= csv.writer( result )

for r in rdr:

wtr.writerow( (r[0], r[1], r[4], r[5], r[6], r[7], r[10], r[11], r[16], r[17], r[18], r[19]) )

# Company\_X\_GPS\_Streams\_rm\_lines.py

import re

sourcefile = "GPS\_Streams\_intermediate\_out.txt"

filename2 = "GPS\_Streams\_intermediate\_output.txt"

offending = [",,",",,,",",,,,"]

def fixup( filename ):

fin = open( filename )

fout = open( filename2 , "w")

for line in fin:

line = re.sub('2014-08-15 ','',line)

# line = re.sub('+00:00','',line)

if True in [item in line for item in offending]:

continue

fout.write(line)

fin.close()

fout.close()

fixup(sourcefile)

# Company\_X\_GPS\_Streams\_rm\_00.py

import re

sourcefile = "GPS\_Streams\_intermediate\_output.txt"

filename2 = "GPS\_Streams\_intermediate\_output1.txt"

offending = [",,",",,,",",,,,"]

def fixup( filename ):

fin = open( filename )

fout = open( filename2 , "w")

for line in fin:

# line = re.sub('2014-08-15 ','',line)

line = re.sub('\+00:00','',line)

# if True in [item in line for item in offending]:

# continue

fout.write(line)

fin.close()

fout.close()

fixup(sourcefile)

# Company\_X\_GPS\_Streams\_script.py

import numpy as np

from numpy import \*

import csv

import math

from math import \*

import matplotlib

from matplotlib import \*

import matplotlib.pyplot as plt

import matplotlib.dates as mdate

from pylab import \*

import datetime as dt

from matplotlib.mlab import csv2rec

import matplotlib.mlab as mlab

from matplotlib.ticker import FormatStrFormatter, MultipleLocator

from matplotlib.ticker import \*

import sys, fileinput

import time

with open("GPS\_Streams\_intermediate\_output1.txt","rb") as source:

rdr= csv.reader( source )

with open("GPS\_Streams\_final\_output.txt","wb") as result:

wtr= csv.writer( result )

for r in rdr:

for i,x in enumerate(r[1:]):

r[i+1] = float(x)

true\_speed\_ms = (r[8]/r[2]+r[9]/r[3])/(1/r[2]+1/r[3])

hdop = sqrt((r[2]\*r[2] + r[3]\*r[3])/2)

x1 = cos(r[4]\*math.pi/180) \* cos(r[6]\*math.pi/180)

x2 = cos(r[5]\*math.pi/180) \* cos(r[7]\*math.pi/180)

y1 = cos(r[4]\*math.pi/180) \* sin(r[6]\*math.pi/180)

y2 = cos(r[5]\*math.pi/180) \* sin(r[7]\*math.pi/180)

z1 = sin(r[4]\*math.pi/180)

z2 = sin(r[5]\*math.pi/180)

x = (x1/r[2] + x2/r[3])/(1/r[2]+1/r[3])

y = (y1/r[2] + y2/r[3])/(1/r[2]+1/r[3])

z = (z1+z2)/2

lon = atan2(y, x)\*180/math.pi

lat = atan2(z, sqrt(x \* x + y \* y))\*180/math.pi

true\_track\_deg = atan2( (sin(r[10]\*math.pi/180) + sin(r[11]\*math.pi/180))/2, (cos(r[10]\*math.pi/180) + cos(r[11]\*math.pi/180))/2 )\*180/math.pi

if true\_track\_deg < 0:

true\_track\_deg = true\_track\_deg + 360

else:

true\_track\_deg = true\_track\_deg

wtr.writerow( (r[0], r[1], hdop, lat, lon, true\_speed\_ms, true\_track\_deg) )

# print r[0]

dates = matplotlib.dates.datestr2num(r[0])

# plt.plot(r[1],hdop)

fig, ax = plt.subplots()

ax.plot\_date(dates, r[1])

date\_fmt = '%H:%M:%S'

date\_formatter = mdate.DateFormatter(date\_fmt)

ax.xaxis.set\_major\_formatter(date\_formatter)

# Sets the tick labels diagonal so they fit easier.

fig.autofmt\_xdate()

plt.savefig('testi2.png')

plot\_GPS\_Streams\_final\_output.py

import csv

import math

from math import \*

from numpy import \*

import matplotlib

from matplotlib import \*

matplotlib.use('PDF')

import matplotlib.pyplot as plt

import matplotlib.mlab as mlab

from matplotlib.ticker import FormatStrFormatter, MultipleLocator

from matplotlib.ticker import \*

from pylab import \*

import sys, fileinput

def getColumn(filename, column):

results = csv.reader(open(filename))

return [result[column] for result in results]

heading= getColumn("GPS\_Streams\_final\_output.txt",1)

hdop = getColumn("GPS\_Streams\_final\_output.txt",2)

lat = getColumn("GPS\_Streams\_final\_output.txt",3)

lon = getColumn("GPS\_Streams\_final\_output.txt",4)

true\_speed = getColumn("GPS\_Streams\_final\_output.txt",5)

true\_track = getColumn("GPS\_Streams\_final\_output.txt",6)

ax=subplot(111)

for label in ax.get\_xticklabels() + ax.get\_yticklabels():

label.set\_fontsize(7.1)

fmt = FormatStrFormatter('%.6f')

ax.xaxis.set\_major\_formatter(fmt)

labels = ax.get\_xticklabels()

setp(labels, rotation=20, ha='right', fontsize=7.1)

ax.yaxis.set\_major\_formatter(fmt)

ax.plot(lat,lon,'bo')

ax.set\_title('Latitude vs. Longitude')

plt.xlabel("Latitude (degrees)")

plt.ylabel("Longitude (degrees)")

# plt.show()

savefig('GPS\_lat\_lon.pdf')

plt.close()

ax=subplot(111)

for label in ax.get\_xticklabels() + ax.get\_yticklabels():

label.set\_fontsize(7.1)

fmt = FormatStrFormatter('%.6f')

ax.xaxis.set\_major\_formatter(fmt)

labels = ax.get\_xticklabels()

setp(labels, rotation=20, ha='right', fontsize=7.1)

ax.yaxis.set\_major\_formatter(fmt)

ax.plot(lon,lat,'bo')

ax.set\_title('Longitude vs. Latitude')

plt.xlabel("Longitude (degrees)")

plt.ylabel("Latitude (degrees)")

# plt.show()

savefig('GPS\_lon\_lat.pdf')

plt.close()

ax=subplot(111)

for label in ax.get\_xticklabels() + ax.get\_yticklabels():

label.set\_fontsize(7.1)

fmt = FormatStrFormatter('%.6f')

ax.xaxis.set\_major\_formatter(fmt)

labels = ax.get\_xticklabels()

setp(labels, rotation=20, ha='right', fontsize=7.1)

ax.yaxis.set\_major\_formatter(fmt)

ax.plot(heading,lat,'bo')

ax.set\_title('Heading vs. Latitude')

plt.xlabel("Heading (degrees)")

plt.ylabel("Latitude (degrees)")

savefig('GPS\_heading\_lat.pdf')

plt.close()

“””

Here are the first few lines, of the final/output GPS\_Streams\_final\_output.txt file:

03:43:13,281.7000122,2.0796633968,36.2451299128,-116.825740222,0.323960930685,283.78500365

03:43:14,281.5,1.11803395118,36.2451286009,-116.82574725,0.157607022952,316.7350006

03:43:15,282.1000061,2.76134021736,36.245132373,-116.825750326,0.240147021348,303.5850067

03:43:16,281.6000061,2.76134021736,36.2451315552,-116.8257441,0.0571361514515,267.63500975

03:43:17,281.6000061,2.76134021736,36.2451292393,-116.825750326,0.309761165726,254.73500825

03:43:18,281.3999939,2.76134021736,36.2451269173,-116.82575647,0.283710559593,247.73500825

03:43:19,281.3999939,2.76134021736,36.2451260955,-116.825762615,0.544216447176,271.95000455

03:43:20,281.7000122,1.68077365639,36.2451251727,-116.825760352,0.18453611993,296.05000305

“””