import folium

from collections import namedtuple

def get\_arrows(locations, color, size=6, n\_arrows=3):

'''

Get a list of correctly placed and rotated

arrows/markers to be plotted

Parameters

locations : list of lists of lat lons that represent the

start and end of the line.

eg [[41.1132, -96.1993],[41.3810, -95.8021]]

arrow\_color : default is 'blue'

size : default is 6

n\_arrows : number of arrows to create. default is 3

Return

list of arrows/markers

'''

Point = namedtuple('Point', field\_names=['lat', 'lon'])

# creating point from our Point named tuple

p1 = Point(locations[0][0], locations[0][1])

p2 = Point(locations[1][0], locations[1][1])

# getting the rotation needed for our marker.

# Subtracting 90 to account for the marker's orientation

# of due East(get\_bearing returns North)

rotation = get\_bearing(p1, p2) - 90

# get an evenly space list of lats and lons for our arrows

# note that I'm discarding the first and last for aesthetics

# as I'm using markers to denote the start and end

arrow\_lats = np.linspace(p1.lat, p2.lat, n\_arrows + 2)[1:n\_arrows+1]

arrow\_lons = np.linspace(p1.lon, p2.lon, n\_arrows + 2)[1:n\_arrows+1]

arrows = []

#creating each "arrow" and appending them to our arrows list

for points in zip(arrow\_lats, arrow\_lons):

arrows.append(folium.RegularPolygonMarker(location=points,

fill\_color=color, number\_of\_sides=3,

radius=size, rotation=rotation).add\_to(m))

return arrows

def get\_bearing(p1, p2):

'''

Returns compass bearing from p1 to p2

Parameters

p1 : namedtuple with lat lon

p2 : namedtuple with lat lon

Return

compass bearing of type float

Notes

Based on https://gist.github.com/jeromer/2005586

'''

long\_diff = np.radians(p2.lon - p1.lon)

lat1 = np.radians(p1.lat)

lat2 = np.radians(p2.lat)

x = np.sin(long\_diff) \* np.cos(lat2)

y = (np.cos(lat1) \* np.sin(lat2)

- (np.sin(lat1) \* np.cos(lat2)

\* np.cos(long\_diff)))

bearing = np.degrees(np.arctan2(x, y))

# adjusting for compass bearing

if bearing < 0:

return bearing + 360

return bearing

amb\_df = pd.read\_excel('Ambiguousfile.xlsx')

amb\_df.head()

req\_df=amb\_df[amb\_df.Name == 'Big Bazaar']

m = folium.Map(location=[28.596035,77.295458],zoom\_start=10)

m.add\_child(folium.Marker([28.596035,77.295458],icon=folium.Icon(color='red', icon='info-sign')))

m.add\_child(folium.Circle(

radius=5940,#merch\_data['Radius'][i],

location=[28.596035,77.295458],

# tooltip=tooltip,

color='blue',

fill=True,

))

for i in range(len(req\_df)):

m.add\_child(folium.Marker([req\_df['Latitude'][i],req\_df['Longitude'][i]],icon=folium.Icon(color='cadetblue', icon='info-sign')))