

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu May 16 10:16:57 2019
```

```
@author: torjus
```

```
THIS PYTHON CODE IS USED TO CONSTRUCT AN LNG TRADEFLOW NETWORK.
```

```
THE CODE IS PRELIMINARY and NOT FINISHED
```

```
"""
```

```
import sqlite3
import matplotlib.pyplot as plt
import time
import datetime
from mpl_toolkits.basemap import Basemap
import numpy as np
from scipy import stats
import pandas as pd
import sklearn.cluster as cluster
import scipy.cluster.hierarchy as hcluster
import loc_check as LC
import networkx as nx
from itertools import cycle, groupby
from pylab import boxplot
import matplotlib.colors as mcolors
import unixTimeConvert as UTC
from mpl_toolkits.axes_grid1.inset_locator import inset_axes
from datetime import timedelta
from shapely.geometry import MultiPoint
from math import radians, cos, sin, asin, sqrt
import AIS_Analysis as AIS
import areas
import AIS_Plotting as PAIS

#Read in nodes (put in mastersheet?)
readNode = 0
if readNode == 1:
    df_n = pd.read_excel('nodelist_merged.xlsx', sheet_name='test')

#Generate all terminal areas
GenTerm = 0
if GenTerm == 1:
    polygons = areas.genAllTerminalAreas(df_n)

#Plot all terminals
PlotTerm = 0
if PlotTerm == 1:
    PAIS.ocean_polygon(polygons)

#Link all AIS message type 1 to a node
assPol = 0
if assPol == 1:
    df2, time = assignPolygon(df, polygons)

#Create Portlog
def createLNGPortLog(df2, df_V, uniqueMMSI1):
    n_delete = 0
    columns = ['DateTime', 'userid', 'port', 'port_prev', 'port_next', 'latitude', 'longitude', 'LNG']
    df_portLog = pd.DataFrame(columns = columns)
    for mmsi in uniqueMMSI1:
```

```

df_temp = df2[df2['userid']==mmsi]
df_temp['diff']=df_temp['cluster'].diff()
df_temp = df_temp[df_temp['diff']!=0]
df_temp = df_temp[df_temp['cluster']!=1]
df_temp['diff2']=df_temp['cluster'].diff() #New diff
df_temp['delete'] = 0
df_temp['timediff']=df_temp['DateTime'].diff()
df_temp['timediff'][0]=timedelta(days=999)
for i in range(0,len(df_temp['cluster'])):
    if (df_temp['diff2'][i]==0) and (df_temp['timediff'][i]<timedelta(days=1)):
        df_temp['delete'][i]=1
        n_delete = n_delete +1
    else:
        date = df_temp['DateTime'][i]
        userid = df_temp['userid'][i]
        port = df_temp['cluster'][i]
        if i>0:
            port_prev = df_temp['cluster'][i-1]
        else:
            port_prev = np.nan
        if i>=(len(df_temp)-1):
            port_next =np.nan
        else:
            port_next = df_temp['cluster'][i+1]

        LNG = df_V['Gas Capacity'][df_V['MMSI']==userid]
        latitude = df_temp['latitude'][i]
        longitude = df_temp['longitude'][i]
        if len(LNG)==0:
            LNG = 0
        else:
            LNG = LNG.iloc[0]
        portvisit = pd.Series([date,userid,port,port_prev,port_next,latitude,longitude,LNG],index=
df_portLog = df_portLog.append(portvisit,ignore_index = True)

return df_portLog

def lngTrade(df_portLog,df_n,uniqueMMSI):
    #This function generate a network matrix of all port to port operations
    #If NM[1][2]=2, this means that two ships(not necessarily unique) have sailed
    #from port 1 to port 2
    #the function has the portLog as input
    NM = np.zeros((len(df_n),len(df_n)))
    lF = 0.97 #loadFactor
    ulF = 0.95 #unloadFactor
    impUnknown = np.zeros(len(df_n))
    expUnknown = np.zeros(len(df_n))
    for mmsi in uniqueMMSI:
        df_vLog = df_portLog[df_portLog['userid']==mmsi]
        for index, row in df_vLog.iterrows():
            port = row['port']
            port_prev = row['port_prev']
            port_next = row['port_next']
            mode = df_n['type'][port]
            mode_prev = df_n['type'][port_prev]
            mode_next = df_n['type'][port_next]
            if mode == 2: #This is a export operation
                if (mode_next ==1) | (mode_next ==3):
                    continue
                else:
                    trade = row['LNG']*lF
                    expUnknown[port] = expUnknown[port]+trade
            elif mode == 1: #This is a import operation
                if (mode_prev == 2) | (mode_prev == 3): # We know the export terminal
                    trade = row['LNG']*lF
                    exporter = port_prev
                    importer = port
                    NM[exporter][importer] = NM[exporter][importer]+trade
                else:
                    trade = row['LNG']*lF
                    impUnknown[port] = impUnknown[port]+trade

```

