```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Fri Feb 22 13:50:19 2019

@author: torjus


THIS PYTHON CODE IS USED TO CREATE VARIOUS PLOTS


"""

#####AIS PLOTTING ##########

#Import
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
import matplotlib.animation as animation


def CreateMap(Pos):

    maxlon,maxlat,minlon,minlat = Pos

    lat0 = (maxlat+minlat)/2
    lon0 = (maxlon+minlon)/2
    lat1 = (maxlat+minlat)/2-20

    fig,ax=plt.subplots(figsize=(15,15))
    m = Basemap(llcrnrlon=minlon,llcrnrlat=minlat,urcrnrlon=maxlon,
                urcrnrlat=maxlat,rsphere=(7578137.00,6356752.3142),
                resolution='l',projection='cyl',lat_0=lat0,lon_0=lon0,
                lat_ts = lat1)

    m.drawmapboundary(fill_color='white',zorder=0)
    m.fillcontinents(color='lightgrey',lake_color='white',zorder=1)
    return m

def LocalMap(df,Pos):
    m = CreateMap(Pos)

    x, y = m(df['longitude'],df['latitude'])

    m.scatter(x,y,0.5,marker='o',c='black',zorder=4)
    return

def densityMap(VesselData,uniqueMMSI,Pos):
    #This function makes a density plot of all ais messages worldwide
    #Input: VesselData is a DataFrame of AIS message type 1

    m = CreateMap(Pos)

    for vessels in uniqueMMSI:
        df_temp = VesselData[VesselData['mmsi']==vessels].copy()
        df_difflat = df_temp['latitude'].diff()
        df_difflon = df_temp['longitude'].diff()
        df_temp['latitude'][abs(df_difflat)>5]=np.nan
```

```
        df_temp['longitude'][abs(df_difflon)>10]=np.nan

        m.plot(df_temp['longitude'],df_temp['latitude'],linewidth = 0.1,color = 'blue',alpha=0.2)
    return

def densityMultipleVessels(df,MMSI,Pos):
    #This function is ment to visualise different vessels patterns
    #df = DataFrame with Message type 1
    #MMSI = array/list of MMSI
    #Pos = Position of Map
    m = CreateMap(Pos)

    for vessels in MMSI:
        df_temp = df[df['userid']==vessels].copy()
        df_difflat = df_temp['latitude'].diff()
        df_difflon = df_temp['longitude'].diff()
        df_temp['latitude'][abs(df_difflat)>50]=np.nan
        df_temp['longitude'][abs(df_difflon)>70]=np.nan
        m.plot(df_temp['longitude'],df_temp['latitude'],linewidth = 0.5,alpha=0.8)
    return

def plotMessages(df) :
    #input DataFrame: uniqueMMSI,nMessages
    ##########################################################
    #Make Histogram of number of messages per Vessel #
    x = list(df['nMessages'])

    fig = plt.figure()
    ax1 = fig.add_axes([0.1,0.1,0.9,0.9])
    ax1.set_xlabel('Messages per Vessel')
    ax1.set_ylabel('Vessels')
    ax1.hist(x,bins=20)


    ax2 = fig.add_axes([0.5,0.5,.4,.4])
    ax2.set_xlabel('Messages per Vessel')
    ax2.set_ylabel('Vessels')
    ax2.hist(x,range=(0,np.percentile(x,75)),bins=20)
    ##########################################################


def plotMessagesYearly(df):

    fig, ax1 = plt.subplots() #Create matplotlib figure
    width = 0.4

    color = 'tab:blue'
    ax1.set_xlabel('Year')
    ax1.set_ylabel('Unique Vessels', color=color)
    ax1.bar(df['year']-width/2,height=df['nVessels'],width=width,color=color)
    ax1.tick_params(axis='y', labelcolor=color)


    ax2 = ax1.twinx()

    color = 'tab:red'
    ax2.set_ylabel('Messages', color=color)  # we already handled the x-label with ax1
    ax2.bar(df['year']+width/2, height=df['nMessages'],width=width, color=color)
    ax2.tick_params(axis='y', labelcolor=color)

    fig.tight_layout()  # otherwise the right y-label is slightly clipped
    plt.show()

    fig, ax = plt.subplots()

    color = 'tab:blue'
    ax.set_xlabel('Year')
    ax.set_ylabel('Messages per Unique Vessel')
    ax.bar(df['year'],height=df['messages/vessel'],color=color)
    ax.tick_params(axis='y', labelcolor=color)
    plt.show()
```

```python
def speedPlot(df):
    fig,ax1 = plt.subplots()

    color = 'tab:blue'
    ax1.set_xlabel('Date')
    ax1.set_ylabel(' Mean Speed', color=color)
    ax1.plot(df['date'],df['mean speed'],color=color)
    ax1.tick_params(axis='y', labelcolor=color)

    ax2 = ax1.twinx()

    color = 'tab:red'
    ax2.set_ylabel('Standard Deviation Speed', color=color)  # we already handled the x-label with ax1
    ax2.plot(df['date'],df['std speed'],color=color)
    ax2.tick_params(axis='y', labelcolor=color)

    fig.tight_layout()  # otherwise the right y-label is slightly clipped
    plt.show()
```

```python
def plotLNGTerminals(df, Pos):
    Onshore_x = df['Longitude'][df['Concept']=='Onshore']
    Onshore_y = df['Latitude'][df['Concept']=='Onshore']
    FSRU_x = df['Longitude'][df['Concept']=='Floating']
    FSRU_y = df['Latitude'][df['Concept']=='Floating']
    m = CreateMap(Pos)
    m.scatter(Onshore_x,Onshore_y,9,marker='o',c='red',label ='Onshore',zorder=2)
    m.scatter(FSRU_x,FSRU_y,9,marker='o',c='blue',zorder=3)

def plotClusteredPorts(df,Pos):
    m = CreateMap(Pos)
    x,y = m(df['longitude'],df['latitude'])
    m.scatter(x,y,marker = 'o',c='red',zorder=2)
```

```python
def ocean_polygon(polygons):
    fig, ax = plt.subplots(figsize=(20,20))
    m = Basemap(projection='cyl',lon_0=0,resolution='l')
    #m.drawparallels(np.arange(-90,90,20),labels=[1,1,0,1],color='k')
    #m.drawmeridians(np.arange(-180,180,20),labels=[1,1,0,1],color='k')
    m.drawmapboundary(fill_color='white')
    m.fillcontinents(color='lightgrey',lake_color='white')
    #m.drawcountries()
    #m.drawcoastlines()

    for i in range(0,len(polygons)):
        x,y = zip(*polygons[i])
        m.plot(x,y,marker='.')
        ax.fill(x, y,alpha=0.2)
    plt.show()
    return m
```