```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Jun  6 19:33:08 2019

@author: torjus

THIS PYTHON FILE IS USED TO PLOT DENSITY PLOTS

"""

#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
import matplotlib.patches as mpatches
import matplotlib.pyplot as plt
from matplotlib.lines import Line2D
from matplotlib.patches import Patch

#Density plots

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 densityqmax(df_q,unique_q):
    Pos = [180, 90, -180, -90]
    m = CreateMap(Pos)

    for vessels in unique_q:
        df_temp = df_q[df_q['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 = 'darkred',alpha=0.15)
    legend_element = [Line2D([0], [0], color='darkred', lw=4, label='Q-Max/Q-Flex')]
    plt.legend(handles=legend_element,loc='lower left')
    return
```

```
def densityconv(df_c,unique_c):
    Pos = [180, 90, -180, -90]
    m = CreateMap(Pos)

    for vessels in unique_c:
        df_temp = df_c[df_c['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 = 'darkorange',alpha=0.15)
    legend_element = [Line2D([0], [0], color='darkorange', lw=4, label='Conventional')]
    plt.legend(handles=legend_element,loc='lower left')
    return
```

```
def densitypanam(df_p,unique_p):
    Pos = [180, 90, -180, -90]
    m = CreateMap(Pos)

    for vessels in unique_p:
        df_temp = df_p[df_p['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 = 'green',alpha=0.15)
    legend_element = [Line2D([0], [0], color='green', lw=4, label='New-Panamax')]
    plt.legend(handles=legend_element,loc='lower left')
    return
```

```
def densitysmall(df_s,unique_s):
    Pos = [180, 90, -180, -90]
    m = CreateMap(Pos)

    for vessels in unique_s:
        df_temp = df_s[df_s['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 = 0.2,color = 'steelblue',alpha=1)
    legend_element = [Line2D([0], [0], color='steelblue', lw=4, label='Small-Scale')]
    plt.legend(handles=legend_element,loc='lower left')
    return
```