```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Tue May 28 09:11:11 2019

@author: torjus
```

THIS PYTHON FILE IS USED TO CONDDUCT SPEED ANALYSIS

```python
"""
import pandas as pd 
from datetime import timedelta
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 


#Speed analysis 
def speedAnalysisMonthly(df):
    columns = ['month','mean speed', 'std speed','nMessages','nVessels']
    df_out =pd.DataFrame(columns = columns)
    lowDate = df['DateTime'].head(1).iloc[0]
    highDate = df['DateTime'].tail(1).iloc[0]
    

    monthsStart = list(pd.date_range(lowDate,highDate,freq='MS').floor('d'))
    monthsEnd = list(pd.date_range(lowDate,highDate,freq='M').floor('d')+timedelta(days=1))

    months = pd.DataFrame([monthsStart,monthsEnd])
        
    for m in range(len(months.columns)) :
        
        meanSpeed = df[str(months[m][0]):str(months[m][1])]['sog'].mean()
        stdSpeed = df[str(months[m][0]):str(months[m][1])]['sog'].std()
        nVessels = df[str(months[m][0]):str(months[m][1])]['mmsi'].nunique()
        nMessages = df[str(months[m][0]):str(months[m][1])]['sog'].count()
        daylyData = pd.Series([months[m][0],meanSpeed,stdSpeed,nMessages,nVessels],index=df_out.columns)
        df_out=df_out.append(daylyData,ignore_index=True)
    return df_out


def speedPlot(df):
```

```
    fig,ax1 = plt.subplots()

    color = 'tab:blue'
    ax1.set_xlabel('Date')
    ax1.set_ylabel(' Mean Speed', color=color)
    ax1.plot(df['month'],df['mean speed'],color=color)
    ax1.tick_params(axis='y', labelcolor=color)
    ax1.set_ylim(10,20)
    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['month'],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()

def speedPlot2(dfs_q,dfs_p,dfs_c,dfs_s,rates):
    fig,ax1 = plt.subplots()

    color = ['darkred','green','darkorange','steelblue']
    ax1.set_xlabel('Date')
    ax1.set_ylabel(' Mean Speed [knots]')
    ax1.plot(dfs_q['month'],dfs_q['mean speed'],color=color[0],label='Q-Max/Q-Flex')
    ax1.plot(dfs_p['month'],dfs_p['mean speed'],color=color[1],label='New-Panamax')
    ax1.plot(dfs_c['month'],dfs_c['mean speed'],color=color[2],label='Conventional')
    ax1.plot(dfs_s['month'],dfs_s['mean speed'],color=color[3],label='Small-Scale')
    ax1.tick_params(axis='y')
    #ax1.set_ylim(10,20)
    ax2 = ax1.twinx()

    color = 'tab:black'
    ax2.set_ylabel('US Dollar')  # we already handled the x-label with ax1
    ax2.plot(rates['Month'],rates['LNGC-160K'],color='black',linestyle='--',label='Spot Rate LNGC 160K (r
    #ax2.tick_params(axis='y', labelcolor=color)

    fig.tight_layout()  # otherwise the right y-label is slightly clipped
    plt.show()
    ax1.legend(loc = "upper left")
    ax2.legend(loc= "upper right")

def histogramSpeedPlot(df_q,df_c,df_s,df_p):
    x1 = df_q.sog
    x2 = df_p.sog
    x3 = df_c.sog
    x4 = df_s.sog

    range1 = [0,25]
    bins = 30
    alpha=1
    histtype = 'step'

    plt.hist(x1,bins=bins,color='darkred',alpha=alpha,density=True,histtype=histtype,label = 'Q-Max/Q-Flex
    plt.hist(x2,bins = bins,color='green',alpha=alpha,density=True,histtype=histtype,label='New-Panamax',
    plt.hist(x3,bins = bins,color='darkorange',alpha=alpha,density=True,histtype=histtype,label='Conventi
    plt.hist(x4,bins = bins,color='steelblue',alpha=alpha,density=True,histtype=histtype,label='Small-Sca
    plt.show()
    plt.legend()
    plt.ylabel('Fraction of AIS Messages')
    plt.xlabel('Speed [knots]')
```