[Code] Decide topical phases

Code Snippet

```
import numpy as np
import pandas as pd
import scipy.signal as signal
import matplotlib.pyplot as plt

## [India] Extract the id_list and plot

total_df = pd.read_pickle('./data/Hindi/hindi_all_tokenized.pkl', compression='gzip')
daily = total_df.groupby('date')
daily_count = daily.count()['id']

N  = 1   # Filter order
Wn = 0.2 # Cutoff frequency
B, A = signal.butter(N, Wn, output='ba')

smooth_data = signal.filtfilt(B,A, daily_count.tolist())

daily = daily_count.reset_index()
daily['smoothed'] = smooth_data

# Calculate diff and double_diff
diff = np.append([0], np.diff(smooth_data)).tolist()
daily['diff'] = diff
double_diff = np.append([0], np.diff(diff)).tolist()
daily['double_diff'] = double_diff

print("Total # of the target days:", daily['date'].count(), '\n')
daily[['diff', 'double_diff']].plot(figsize = (15, 8))

# The first case of the 2019–20 coronavirus pandemic in India was reported on 30 January 2020.
first_date = '2020-01-30'
first_date_idx = daily.loc[daily.date==first_date].index.tolist()[0]

first_smoothed = daily['smoothed'][first_date_idx]
first_diff = math.floor(daily['diff'][first_date_idx]) + 1
first_double_diff = math.floor(daily['double_diff'][first_date_idx])

idx_list = daily[(daily['diff'] > 0) & (daily['diff'] < first_diff) & (daily['double_diff'] > first_double_diff)].index
daily[['smoothed']].plot(figsize = (15, 8))
plt.title("India")

list_phase_date = []  # list of the detected 'start dates of new phases'
for idx in idx_list:
    plt.axvline(x=idx, color='k', linestyle='--')
    list_phase_date.append(str(daily['date'][idx]).split(" ")[0])
    print("index:", idx)
    print("date:", str(daily['date'][idx]).split(" ")[0])
    print("smoothed:", round(daily['smoothed'][idx], 2))
    print("diff:", round(daily['diff'][idx], 2))
    print("double_diff:", round(daily['double_diff'][idx], 2))
    print('\n')
```

```
## Check the decided phases: their composing dates and no. of tweets by day

new_phase_date = []; phase_no = {}; i = 0
total_df['date'] = pd.to_datetime(total_df['date'])

for phase_date in list_phase_date:
    datetime_str = phase_date
    datetime_object = np.datetime64(datetime.date(datetime.strptime(datetime_str, '%Y-%m-%d')))

    print("End of Phase_" + str(i) + ": " + str(datetime_object))
    new_phase_date.append(datetime_object)

    if i == 0:
        phase_no[i] = total_df.loc[(total_df.date < new_phase_date[i])]
    elif i+1 <= len(list_phase_date):
        phase_no[i] = total_df.loc[(total_df.date >= new_phase_date[i-1]) & (total_df.date < new_phase_date[i])]

    print(phase_no[i]['date'].value_counts(sort=True, ascending=False), '\n')
    i += 1

phase_no[i] = total_df.loc[(total_df.date >= new_phase_date[i-1])]
print("End of Phase_" + str(i) + ": till the end")
print(phase_no[i]['date'].value_counts(sort=True, ascending=False))
```