STAGE 5 TEAM Task2

April 26, 2023

[11]:
```
!pip install dash
!pip install plotly
```

```
Requirement already satisfied: dash in c:\users\ashdh\anaconda3\lib\site-
packages (2.9.3)
Requirement already satisfied: dash-html-components==2.0.0 in
c:\users\ashdh\anaconda3\lib\site-packages (from dash) (2.0.0)
Requirement already satisfied: plotly>=5.0.0 in
c:\users\ashdh\anaconda3\lib\site-packages (from dash) (5.9.0)
Requirement already satisfied: Flask>=1.0.4 in
c:\users\ashdh\anaconda3\lib\site-packages (from dash) (1.1.2)
Requirement already satisfied: dash-table==5.0.0 in
c:\users\ashdh\anaconda3\lib\site-packages (from dash) (5.0.0)
Requirement already satisfied: dash-core-components==2.0.0 in
c:\users\ashdh\anaconda3\lib\site-packages (from dash) (2.0.0)
Requirement already satisfied: click>=5.1 in c:\users\ashdh\anaconda3\lib\site-
packages (from Flask>=1.0.4->dash) (8.0.4)
Requirement already satisfied: itsdangerous>=0.24 in
c:\users\ashdh\anaconda3\lib\site-packages (from Flask>=1.0.4->dash) (2.0.1)
Requirement already satisfied: Jinja2>=2.10.1 in
c:\users\ashdh\anaconda3\lib\site-packages (from Flask>=1.0.4->dash) (2.11.3)
Requirement already satisfied: Werkzeug>=0.15 in
c:\users\ashdh\anaconda3\lib\site-packages (from Flask>=1.0.4->dash) (2.0.3)
Requirement already satisfied: tenacity>=6.2.0 in
c:\users\ashdh\anaconda3\lib\site-packages (from plotly>=5.0.0->dash) (8.0.1)
Requirement already satisfied: colorama in c:\users\ashdh\anaconda3\lib\site-
packages (from click>=5.1->Flask>=1.0.4->dash) (0.4.5)
Requirement already satisfied: MarkupSafe>=0.23 in
c:\users\ashdh\anaconda3\lib\site-packages (from
Jinja2>=2.10.1->Flask>=1.0.4->dash) (2.0.1)
Requirement already satisfied: plotly in c:\users\ashdh\anaconda3\lib\site-
packages (5.9.0)
Requirement already satisfied: tenacity>=6.2.0 in
c:\users\ashdh\anaconda3\lib\site-packages (from plotly) (8.0.1)
```

[13]:
```
# Importing required libraries
import dash
#from jupyter_dash import JupyterDash
```

```
import dash_core_components as dcc
import dash_html_components as html
from dash.dependencies import Input, Output
import pandas as pd
import numpy as np
import plotly.express as px
import plotly.graph_objs as go
from datetime import date
```

[14]:
```
#reading the data for the confirmed number of COVID cases and displaying them
confirmedCasesNew = pd.read_csv(r"..\..\DATASETS\COVID␣
 ↪DATASETS\covid_confirmed_usafacts.csv")
a = confirmedCasesNew.iloc[:,4:].diff(axis=1).loc[:, '2022-05-30':'2023-01-01']
confirmedCases = pd.concat([confirmedCasesNew.iloc[:,:4], a], axis=1)
confirmedCases
```

[14]:

	countyFIPS	County Name	State	StateFIPS	2022-05-30	\
0	0	Statewide Unallocated	AL	1	0	
1	1001	Autauga County	AL	1	9	
2	1003	Baldwin County	AL	1	55	
3	1005	Barbour County	AL	1	1	
4	1007	Bibb County	AL	1	9	
…	…	…	…	…	…	
3188	56037	Sweetwater County	WY	56	0	
3189	56039	Teton County	WY	56	0	
3190	56041	Uinta County	WY	56	0	
3191	56043	Washakie County	WY	56	0	
3192	56045	Weston County	WY	56	0	

	2022-05-31	2022-06-01	2022-06-02	2022-06-03	2022-06-04	…	\
0	0	0	0	0	0	…	
1	24	6	9	0	0	…	
2	183	68	68	0	0	…	
3	12	3	4	0	0	…	
4	9	8	4	0	0	…	
…	…	…	…	…	…	…	
3188	29	0	0	0	0	…	
3189	62	0	0	0	0	…	
3190	23	0	0	0	0	…	
3191	3	0	0	0	0	…	
3192	4	0	0	0	0	…	

	2022-12-23	2022-12-24	2022-12-25	2022-12-26	2022-12-27	2022-12-28	\
0	0	0	0	0	0	0	
1	0	0	0	0	0	0	
2	0	0	0	0	0	0	
3	0	0	0	0	0	0	

```
4              0           0           0           0           0           0
…            …           …           …           …           …           …
3188           0           0           0           0           0           0
3189           0           0           0           0           0           0
3190           0           0           0           0           0           0
3191           0           0           0           0           0           0
3192           0           0           0           0           0           0

      2022-12-29  2022-12-30  2022-12-31  2023-01-01
0              0           0           0           0
1              0           0           0           0
2              0           0           0           0
3              0           0           0           0
4              0           0           0           0
…            …           …           …           …
3188           0           0           0           0
3189           0           0           0           0
3190           0           0           0           0
3191           0           0           0           0
3192           0           0           0           0

[3193 rows x 221 columns]
```

[15]:
```
#reading the data for the confirmed number of COVID cases and displaying them
confirmedCasesNew = pd.read_csv(r"..\..\DATASETS\COVID␣
 ↪DATASETS\covid_confirmed_usafacts.csv")
a = confirmedCasesNew.iloc[:,4:].diff(axis=1).loc[:, '2022-05-30':'2023-01-01']
confirmedCases = pd.concat([confirmedCasesNew.iloc[:,:4], a], axis=1)
confirmedCases



# reading the confirmed number of COVID Deaths and displaying them
confirmedDeathsNew = pd.read_csv(r"..\..\DATASETS\COVID␣
 ↪DATASETS\covid_deaths_usafacts.csv")
b = confirmedDeathsNew.iloc[:,4:].diff(axis=1).loc[:, '2022-05-30':'2023-01-01']
confirmedDeaths = pd.concat([confirmedDeathsNew.iloc[:,:4], b], axis=1)
```

[16]:
```
id_vars = ['countyFIPS', 'County Name', 'State', 'StateFIPS']
data1 = pd.melt(confirmedCases, id_vars= ['countyFIPS', 'County Name', 'State',␣
 ↪'StateFIPS'], var_name='date', value_name='cases')
data1 = data1[data1['County Name'] != 'Statewide Unallocated']

data2 = pd.melt(confirmedDeaths , id_vars= ['countyFIPS', 'County Name',␣
 ↪'State', 'StateFIPS'], var_name='date', value_name='deaths')
data2 = data2[data2['County Name'] != 'Statewide Unallocated']
data2
```

```
data = data1.merge(data2, on=['countyFIPS', 'County Name', 'State',␣
 ↪'StateFIPS', 'date'])
```

[17]:
```
#creating log values
data['log_cases']=np.log(data['cases'])
data['log_deaths']=np.log(data['deaths'])

#making the date column as a date type
data['date']=data['date'].astype('datetime64[ns]')

#removing all negative and non numeric values from the cases and deaths
data['cases'] = data['cases'].clip(lower=0)
data['deaths'] = data['deaths'].clip(lower=0)
data['log_cases'] = data['log_cases'].clip(lower=0)
data['log_deaths'] = data['log_deaths'].clip(lower=0)

data
```

```
C:\Users\ashdh\anaconda3\lib\site-packages\pandas\core\arraylike.py:397:
RuntimeWarning:

divide by zero encountered in log

C:\Users\ashdh\anaconda3\lib\site-packages\pandas\core\arraylike.py:397:
RuntimeWarning:

invalid value encountered in log

C:\Users\ashdh\anaconda3\lib\site-packages\pandas\core\arraylike.py:397:
RuntimeWarning:

divide by zero encountered in log

C:\Users\ashdh\anaconda3\lib\site-packages\pandas\core\arraylike.py:397:
RuntimeWarning:

invalid value encountered in log
```

[17]:

	countyFIPS	County Name	State	StateFIPS	date	cases	\
0	1001	Autauga County	AL	1	2022-05-30	9	
1	1003	Baldwin County	AL	1	2022-05-30	55	
2	1005	Barbour County	AL	1	2022-05-30	1	
3	1007	Bibb County	AL	1	2022-05-30	9	
4	1009	Blount County	AL	1	2022-05-30	6	
…	…	…	…	…	…	…	
681809	56037	Sweetwater County	WY	56	2023-01-01	0	

```
681810       56039       Teton County     WY         56 2023-01-01      0
681811       56041       Uinta County     WY         56 2023-01-01      0
681812       56043    Washakie County     WY         56 2023-01-01      0
681813       56045      Weston County     WY         56 2023-01-01      0

        deaths  log_cases  log_deaths
0            0   2.197225         0.0
1            1   4.007333         0.0
2            0   0.000000         0.0
3            0   2.197225         0.0
4            0   1.791759         0.0
…          …         …          …
681809       0   0.000000         0.0
681810       0   0.000000         0.0
681811       0   0.000000         0.0
681812       0   0.000000         0.0
681813       0   0.000000         0.0

[681814 rows x 9 columns]
```

[18]:
```
data.info(5)
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 681814 entries, 0 to 681813
Data columns (total 9 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   countyFIPS   681814 non-null  int64
 1   County Name  681814 non-null  object
 2   State        681814 non-null  object
 3   StateFIPS    681814 non-null  int64
 4   date         681814 non-null  datetime64[ns]
 5   cases        681814 non-null  int64
 6   deaths       681814 non-null  int64
 7   log_cases    679581 non-null  float64
 8   log_deaths   680011 non-null  float64
dtypes: datetime64[ns](1), float64(2), int64(4), object(2)
memory usage: 52.0+ MB
```

[19]:
```
#ORIGINAL
app = dash.Dash(__name__)
df=data
states = data['State'].unique().tolist()
states.insert(0,"All States")
# Create dropdown options
dropdown_options = [{'label': state, 'value': state} for state in states]
```

```
app.layout = html.Div([
            html.H1("Covid-19 7 Day moving average",style={'text-align':
 ↪'center'}),
            dcc.DatePickerRange(
                id='my-date-picker-range',
                min_date_allowed=date(2022, 5, 30),
                max_date_allowed=date(2023, 1, 1),
                initial_visible_month=date(2022, 6, 1),
                start_date_placeholder_text='Start Date',
                end_date_placeholder_text='End Date',
                display_format='MM/DD/YYYY',
                month_format='MMMM, YYYY',
                start_date=date(2022, 6, 1),
                end_date=date(2023, 1, 1)
            ),
    html.Br(),

    #SETTING A DROP DOWN FOR STATE SELECTION
    html.Div([
        html.Label('Select State:'),
        dcc.Dropdown(
            id='state-dropdown',
            options=dropdown_options,
            multi=True,
            #value="NJ" # Set default value
            value='All States'
            )]),
    html.Br(),
    #SETTING TOGGLE BUTTON FOR CASES/DEATHS
     html.Div([
            html.Label("Cases Or Deaths:"),
            dcc.RadioItems(id="CorD",
                options=[
                    {"label": "Cases", "value": "cases"},
                    {"label": "Deaths", "value": "deaths"}
                ],
                value="cases", labelStyle={"display":"block"})]),
    html.Br(),

    dcc.Graph(id='Trend_Plot')
    ])

@app.callback(
    Output(component_id='Trend_Plot', component_property='figure'),
    Input('my-date-picker-range', 'start_date'),
    Input('my-date-picker-range', 'end_date'),
    Input('CorD', 'value'),
```

```
    Input('state-dropdown', 'value')
)

def updated_graph(start_date, end_date, CorD_value, selected_state):
    if 'All States' in selected_state:
        df_state = data.groupby("date").sum().reset_index()
    else:
        df_state = data[data['State'].isin(selected_state)].groupby("date").
 ↪sum().reset_index()

    if start_date is not None:
            df_1 = df_state[(df_state['date'] >= start_date) &␣
 ↪(df_state['date'] <= end_date)]
    else:
        df_1 = df_state

    if CorD_value == 'cases':
        fig_c = px.scatter(df_1, x='date', y='cases',␣
 ↪color='cases',trendline='rolling',trendline_options=dict(window=7),title='7␣
 ↪day moving average')
        return fig_c
    elif CorD_value == 'deaths':
        fig_c = px.scatter(df_1, x='date', y='deaths',␣
 ↪color='deaths',trendline='rolling',trendline_options=dict(window=7),title='7␣
 ↪day moving average')
        return fig_c

# Run the app
if __name__ == '__main__':
    app.run_server(debug=False)
```

```
Dash is running on http://127.0.0.1:8050/

 * Serving Flask app "__main__" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production
deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:8050/ (Press CTRL+C to quit)
127.0.0.1 - - [26/Apr/2023 18:17:44] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [26/Apr/2023 18:17:44] "GET /_dash-layout HTTP/1.1" 200 -
127.0.0.1 - - [26/Apr/2023 18:17:44] "GET /_dash-dependencies HTTP/1.1" 200 -
127.0.0.1 - - [26/Apr/2023 18:17:44] "GET /_dash-component-
suites/dash/dcc/async-datepicker.js HTTP/1.1" 200 -
127.0.0.1 - - [26/Apr/2023 18:17:44] "GET /_dash-component-
```

```
suites/dash/dcc/async-graph.js HTTP/1.1" 200 -
127.0.0.1 - - [26/Apr/2023 18:17:44] "GET /_dash-component-
suites/dash/dcc/async-dropdown.js HTTP/1.1" 200 -
127.0.0.1 - - [26/Apr/2023 18:17:44] "GET /_dash-component-
suites/dash/dcc/async-plotlyjs.js HTTP/1.1" 200 -
127.0.0.1 - - [26/Apr/2023 18:17:44] "POST /_dash-update-component HTTP/1.1" 200
-
127.0.0.1 - - [26/Apr/2023 18:18:14] "POST /_dash-update-component HTTP/1.1" 200
-
127.0.0.1 - - [26/Apr/2023 18:18:53] "POST /_dash-update-component HTTP/1.1" 200
-
127.0.0.1 - - [26/Apr/2023 18:18:54] "POST /_dash-update-component HTTP/1.1" 200
-
127.0.0.1 - - [26/Apr/2023 18:18:57] "POST /_dash-update-component HTTP/1.1" 200
-
127.0.0.1 - - [26/Apr/2023 18:18:59] "POST /_dash-update-component HTTP/1.1" 200
-
127.0.0.1 - - [26/Apr/2023 18:19:02] "POST /_dash-update-component HTTP/1.1" 200
-
127.0.0.1 - - [26/Apr/2023 18:19:06] "POST /_dash-update-component HTTP/1.1" 200
-
127.0.0.1 - - [26/Apr/2023 18:19:09] "POST /_dash-update-component HTTP/1.1" 200
-
127.0.0.1 - - [26/Apr/2023 18:19:37] "POST /_dash-update-component HTTP/1.1" 200
-
```

[20]:
```
%tb
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_35320\3670559490.py in <module>
      1 #ORIGINAL
----> 2 app = dash.Dash(__name__)
      3 df=data
      4 states = data['State'].unique().tolist()
      5 states.insert(0,"All States")

NameError: name 'dash' is not defined
```