Dash\_wildfire.py

import pandas as pd

import dash

from dash import html, dcc

from dash.dependencies import Input, Output, State

import plotly.graph\_objects as go

import plotly.express as px

from dash import no\_update

import datetime as dt

#Create app

app = dash.Dash(\_\_name\_\_)

# Clear the layout and do not display exception till callback gets executed

app.config.suppress\_callback\_exceptions = True

# Read the wildfire data into pandas dataframe

df = pd.read\_csv('https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DV0101EN-SkillsNetwork/Data%20Files/Historical\_Wildfires.csv')

#Extract year and month from the date column

df['Month'] = pd.to\_datetime(df['Date']).dt.month\_name() #used for the names of the months

df['Year'] = pd.to\_datetime(df['Date']).dt.year

#Layout Section of Dash

#Task 2.1 Add the Title to the Dashboard

app.layout = html.Div(children=[html.H1(Australia Wildfire Dashboard),

# TASK 2.2: Add the radio items and a dropdown right below the first inner division

#outer division starts

html.Div([

html.H2('Select Region:', style={'margin-right': '2em'}),

#Radio items to select the region

dcc.RadioItems(['NSW','QL','SA','TA','VI','WA'], 'NSW', id='region',inline=True)]),

#Dropdown to select year

html.Div([

html.H2('Select Year:', style={'margin-right': '2em.}),

dcc.Dropdown(df.Year.unique(), value = 2005,id='year')

]),

#Second Inner division for adding 2 inner divisions for 2 output graphs

#TASK 2.3: Add two empty divisions for output inside the next inner division.

html.Div([

html.Div([ ], id='plot1'),

html.Div([ ], id='plot2')

], style={'.........}),

])

#outer division ends

])

#layout ends

#TASK 2.4: Add the Ouput and input components inside the app.callback decorator.

#Place to add @app.callback Decorator

@app.callback([Output(component\_id='plot1', component\_property='children'),

Output(component\_id='plot2', component\_property='children')],

[Input(component\_id='region', component\_property='value'),

Input(component\_id='year', component\_property='value')])

#TASK 2.5: Add the callback function.

#Place to define the callback function .

ddef reg\_year\_display(input\_region,input\_year):

#data

region\_data = df[df['Region'] == input\_region]

y\_r\_data = region\_data[region\_data['Year']==input\_year]

#Plot one - Monthly Average Estimated Fire Area

est\_data = y\_r\_data.groupby('Month')['Estimated\_fire\_area'].mean().reset\_index()

fig1 = px.pie(est\_data, values='Estimated\_fire\_area', names='Month', title="{} : Monthly Average Estimated Fire Area in year {}".format(input\_region,input\_year))

#Plot two - Monthly Average Count of Pixels for Presumed Vegetation Fires

veg\_data = y\_r\_data.groupby('Month')['Count'].mean().reset\_index()

fig2 = px.bar(veg\_data, x='Month', y='Count', title='{} : Average Count of Pixels for Presumed Vegetation Fires in year {}'.format(input\_region,input\_year))

return [dcc.Graph(figure=fig1),

dcc.Graph(figure=fig2) ]

if \_\_name\_\_ == '\_\_main\_\_':

app.run\_server()

python3.8 Dash\_wildfire.py

