Good code

#!/usr/bin/env python

# coding: utf-8

# In[ ]

import dash

from dash import dcc

from dash import html

from dash.dependencies import Input, Output

import pandas as pd

import plotly.graph\_objs as go

import plotly.express as px

# Load the data using pandas

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

# Initialize the Dash app

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

# Set the title of the dashboard

#app.title = "Automobile Statistics Dashboard"

#---------------------------------------------------------------------------------

# Create the dropdown menu options

dropdown\_options = [

    {'label': 'Yearly Statistics', 'value': 'Yearly Statistics'},

    {'label': 'Recession Period Statistics', 'value': 'Recession Period Statistics'}

]

# List of years

year\_list = [i for i in range(1980, 2024, 1)]

#---------------------------------------------------------------------------------------

# Create the layout of the app

app.layout = html.Div([

    #TASK 2.1 Add title to the dashboard

    html.H1("Automobile Sales Statistics Dashboard", style={'textAlign': 'center', 'color': '#503D36', 'font-size': 24} ),#May include style for title

    html.Div([#TASK 2.2: Add two dropdown menus

        html.Label("Select Statistics:"),

        dcc.Dropdown(

            id='dropdown-statistics',

            options=dropdown\_options,

            value='Select Statistics',

            placeholder='Select a report type'

        )

    ]),

    html.Div(dcc.Dropdown(

            id='select-year',

            options=[{'label': i, 'value': i} for i in year\_list],

            value='Select Year'

        )),

    html.Div([#TASK 2.3: Add a division for output display

    html.Div(id='output-container', className='chart-grid', style={'display': 'flex'}),])

])

 #TASK 2.4: Creating Callbacks

# Define the callback function to update the input container based on the selected statistics

@app.callback(

    Output(component\_id='select-year', component\_property='disabled'),

    Input(component\_id='dropdown-statistics',component\_property='value'))

def update\_input\_container(selected\_statistics):

    if selected\_statistics =='Yearly Statistics':

        return False

    else:

        return True

    #Callback for plotting

# Define the callback function to update the input container based on the selected statistics

@app.callback(

    Output(component\_id='output-container', component\_property='children'),

    [Input(component\_id='select-year', component\_property='value'), Input(component\_id='dropdown-statistics', component\_property='value')])

def update\_output\_container(selected\_statistics, input\_year):

    if selected\_statistics == 'Recession Period Statistics':

        # Filter the data for recession periods

        recession\_data = data[data['Recession'] == 1]

        #Plot 1 Automobile sales fluctuate over Recession Period (year wise)

        # use groupby to create relevant data for plotting

        yearly\_rec=recession\_data.groupby('Year')['Automobile\_Sales'].mean().reset\_index()

        R\_chart1 = dcc.Graph(

            figure=px.line(yearly\_rec,

                x='Year',

                y='Automobile\_Sales',

                title="Average Automobile Sales fluctuation over Recession Period"))

        #Plot 2 Calculate the average number of vehicles sold by vehicle type

        # use groupby to create relevant data for plotting

        average\_sales =recession\_data.groupby('Vehicle\_Type')['Automobile\_Sales'].mean().reset\_index()

        R\_chart2  = dcc.Graph(figure=px.bar(average\_sales, x='Vehicle\_Type', y='Automobile\_Sales'))

        exp\_rec =recession\_data.groupby('Vehicle\_Type')['Automobile\_Sales'].sum().reset\_index()

        R\_chart3 = dcc.Graph(

                figure=px.pie(exp\_rec,

                values='Automobile\_Sales',

                names='Vehicle\_Type',

                title="Total Advertising Expenditure per Vehicle Type during Recession"))

         # Plot 4 bar chart for the effect of unemployment rate on vehicle type and sales

        R\_chart4 = dcc.Graph(figure=px.bar(recession\_data, x='Unemployment\_Rate', y='Automobile\_Sales', color= 'Vehicle\_Type'))

        return [

            html.Div(className='chart-item', children=[html.Div(children=R\_chart1), html.Div(children=R\_chart2)]),

            html.Div(className='chart-item', children=[html.Div(children=R\_chart3), html.Div(children=R\_chart4)])

            ]

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

    app.run\_server(debug=True, port=8053)