import pandas   as pd

from dash import Dash, html, dcc, Input, Output

import plotly.express as px

import dash\_bootstrap\_components as dbc

## 1- Cargar data frame

df = pd.read\_csv('onlinefoods.csv')

## 2 - Transformación de datos (si es necesario)

## 3 - Crear la aplicación Dash

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

app.title = 'reporte de pedidos de comida Online'

## 4 - Crear el layout de la aplicación

app.layout = html.Div([

    html.Div([

        html.H1("Reporte de Pedidos de Comida Online"),

        dcc.Dropdown(

            id='selector\_Genero',

            options=[{'label': Genero, 'value': Genero} for Genero in df['Gender'].unique()],

            value=df['Gender'].unique()[0]  # Valor por defecto

        ),

        dcc.RadioItems(

            id='Selector\_ocupacion',

            options=[{'label': ocupacion, 'value': ocupacion} for ocupacion in df['Occupation'].unique()],

            value=df['Occupation'].unique()[0], # Valor por defecto

        ),

        dcc.RangeSlider(

            id='selector\_edad',

            min=df['Age'].min(),

            max=df['Age'].max(),

            value=[df['Age'].min(), df['Age'].max()],

            marks={i: str(i) for i in range(df['Age'].min(), df['Age'].max() + 1)},

            step=1

        ),

        dcc.Graph(id='grafico\_barras'),

        dcc.Graph(id='grafico\_pastel'),

        html.Div(df.columns)

    ])

])

## 5 - Crear el callback para actualizar el gráfico

@app.callback(

   [Output('grafico\_barras', 'figure'),

    Output('grafico\_pastel', 'figure')],

   [Input('Selector\_ocupacion', 'value'),

    Input('selector\_Genero', 'value'),

    Input('selector\_edad', 'value')]

)

def crear\_graficas(valor\_ocupacion,valor\_genero, valor\_edad):

    #filtrar el DataFrame por género seleccionado

    df\_filtrado = df[(df['Gender'] == valor\_genero) & (df['Occupation'] == valor\_ocupacion) & (df['Age'] >= valor\_edad[0]) & (df['Age'] <= valor\_edad[1])]

    Promedio\_feedback\_estado\_civil = df\_filtrado.groupby(["Marital Status",'Feedback'])['Age'].mean().reset\_index().sort\_values(by='Feedback', ascending=False)

    grafico\_barras = px.bar(Promedio\_feedback\_estado\_civil,

                            x='Marital Status',

                            y='Age',

                            color='Feedback',

                            title='Promedio de Tamaño de Familia por Estado Civil',

                            color\_discrete\_sequence=['#65c78c','#f74a50'])

    Conteo\_votos\_feedback = df\_filtrado.groupby(['Feedback'])['Age'].mean().reset\_index().sort\_values(by='Feedback', ascending=False)

    grafico\_pastel = px.pie(Conteo\_votos\_feedback,

                            names= 'Feedback',

                            values= 'Age',

                            title='Proporción de Tamaño de Familia por Estado Civil',

                            color\_discrete\_sequence=['#65c78c','#f74a50'])

    return grafico\_barras, grafico\_pastel

## 6 - Ejecutar la aplicación

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

    app.run(debug=True)