Diabetes Research Dashboard

November 9, 2021

Table of Contents

[1]:
```
# Uses a slider to control the year of the chart
from jupyter_dash import JupyterDash
from dash.dependencies import Output, Input
from dash import no_update
from dash import dcc
from dash import html

import pandas as pd
import plotly.graph_objects as go
import plotly.express as px


diabetes = pd.read_csv('https://bitbucket.org/jimcody/sampledata/raw/
 ↪b2aa6df015816ec35afc482b53df1b7ca7a31f80/diabetes_for_plotly.csv')
diabetes['gender'] = diabetes['gender'].replace({'M':'Male', 'Mle':'Male', 'F':
 ↪'Female',
                                                 'female':'Female', 'male':
 ↪'Male',
                                                 '?':'Female', 'Unknown/
 ↪Invalid':'Female'})

# The following group by statements will run before I reduce the number of rows␣
 ↪in diabetes
d_gender = diabetes.groupby('gender').sum().reset_index()
d_month = diabetes.groupby(['year','month']).sum().reset_index()
d_month = d_month.sort_values(['year','month'])

# Reduce the number of rows
#diabetes = diabetes[(diabetes['diabetesMed'] == 'No') & (diabetes['year'] ==␣
 ↪2021) &
#                     (diabetes['admission_type_id'] == 1)]
# This leaves 4292 rows to process



#fig1 = px.scatter(diabetes, x=diabetes.num_lab_procedures,
#                  y=diabetes.num_medications,
#                  size = diabetes.time_in_hospital,
#                  color = diabetes.gender,
#                  hover_data = ['age'])
#fig1.show()


#fig3 = px.bar(d_gender, x='gender', y=['num_lab_procedures',␣
 ↪'num_procedures'], barmode = 'group')
#fig3.show()

#fig4 = go.Figure(
#     data=[go.Bar(name = 'labs', x=d_gender.gender, y = d_gender.
 ↪num_lab_procedures),
#         go.Bar(name = 'non labs', x=d_gender.gender, y = d_gender.
 ↪num_procedures)],
#     layout=go.Layout(
#         title=go.layout.Title(text="A Figure Specified By A Graph Object")
#     )
#)
#fig4.show()

fig5 = px.line(d_month,x='month', y='num_medications')
#fig5.show()

####### Build the App. #################
app = JupyterDash(__name__)

app.layout = html.Div([
    dcc.Graph(id='x', figure = fig5),
    dcc.Slider(
        id='year-slider',
        min=d_month['year'].min(),
        max=d_month['year'].max(),
        value=d_month['year'].min(),
        marks={str(year): str(year) for year in d_month['year'].unique()},
        step=None
    )
])
@app.callback(
    Output('x', 'figure'),
    Input('year-slider', 'value'))

def update_figure(selected_year):
    d_year = d_month[d_month.year == selected_year]

    fig5 = px.line(d_year,x='month', y='num_medications')
    fig5.update_layout(transition_duration=500)

    return fig5

app.run_server(mode='inline')
```

```
<IPython.lib.display.IFrame at 0x7f97f31cb490>
```

[4]:
```
# Uses a dropdown list to control the year of the chart
from jupyter_dash import JupyterDash
from dash.dependencies import Output, Input
from dash import no_update
from dash import dcc
from dash import html

import pandas as pd
import plotly.graph_objects as go
import plotly.express as px

diabetes = pd.read_csv('https://bitbucket.org/jimcody/sampledata/raw/
 ↪b2aa6df015816ec35afc482b53df1b7ca7a31f80/diabetes_for_plotly.csv')
diabetes['gender'] = diabetes['gender'].replace({'M':'Male', 'Mle':'Male', 'F':
 ↪'Female',
                                                  'female':'Female', 'male':
 ↪'Male',
                                                  '?':'Female', 'Unknown/
 ↪Invalid':'Female'})

d_month = diabetes.groupby(['year','month']).sum().reset_index()
d_month = d_month.sort_values(['year','month'])

fig5 = px.line(d_month,x='month', y='num_medications')

####### Build the App. ##################
app = JupyterDash(__name__)

app.layout = html.Div([
    dcc.Dropdown(id='dropdown',
                 options=[
                {'label': i, 'value': i} for i in d_month.year.unique()
                ],  value=2019,
                    clearable=False,placeholder='Filter by year...'),
    dcc.Graph(id='x', figure = fig5)
])
```

```
@app.callback(
    Output('x', 'figure'),
    Input('dropdown', 'value'))

def update_figure(selected_year):
    d_month2 = d_month[d_month.year == selected_year]

    fig5 = px.line(d_month2,x='month', y='num_medications')


    fig5.update_layout(transition_duration=100)

    return fig5

app.run_server(mode='inline')
```

```
<IPython.lib.display.IFrame at 0x7fe480d770a0>
```

[ ]:

```

```