

99%20-%20Exercise%201%20Solution

April 13, 2023

Topics

```
[ ]: # 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/sampleddata/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'})
#diabetes = diabetes[diabetes['year'] == 2019]
```

```
[ ]: diabetes.info()
```

```
[ ]: diabetes = diabetes.drop('encounter_id',axis=1)
diabetes = diabetes.drop('patient_nbr',axis=1)
diabetes = diabetes.drop('admission_type_id',axis=1)
diabetes = diabetes.drop('discharge_disposition_id',axis=1)
diabetes = diabetes.drop('diag_1',axis=1)
```

0.1 Run the next cell to see what data you have to work with

```
[ ]: diabetes.head()
```

0.2 Create a dataframe to hold the data you want to work with

```
[ ]: diabetes = diabetes.drop('A1Cresult',axis=1)
diabetes = diabetes.drop('insulin',axis=1)
diabetes = diabetes.drop('diabetesMed',axis=1)
diabetes = diabetes.drop('readmitted',axis=1)

[ ]: d_month = diabetes.groupby(['year','month','race','gender','age']).sum().
    ↪reset_index()
d_month

[ ]: d_month = d_month.sort_values(['year','month'])

[ ]: d_gender = diabetes.
    ↪groupby('gender')[['time_in_hospital','num_lab_procedures','num_procedures','num_medication
    ↪sum().reset_index()
```

1 Exercise 1 - plotly charts - 30 minutes

- read in the diabetes_for_plotly dataset (already done above)
- group data as needed
- Use express or graph objects
- Create a scatter plot of any two measures. Use a third measure to adjust the size. Color by a categorical value. Add hover text to show the age group.
- Create a side-by-side bar chart showing number of lab procedures and number of non lab procedures by gender.
- Create a line chart showing number of number of medications by month.
- Create a line chart showing number of number of procedures by month.
- Create a fifth chart of your choice (NOT scatter, bar or line) using the documentation.

1.0.1 scatterplot is given a variable name: labs

```
[ ]: # express version
# Create a scatter plot of any two measures. Use a third measure to adjust the
    ↪size. Color by a categorical value.
# Add hover text to show the age group.
labs = px.scatter(diabetes, x=diabetes.num_lab_procedures,
                  y=diabetes.num_medications,
                  # size = diabetes.time_in_hospital,
                  # color = diabetes.gender,
                  # hover_data = ['age']
                  )
labs.show()

[ ]: # go version
fig = go.Figure()
```

```
fig.add_trace(go.Scatter(
    x=diabetes.num_lab_procedures,
    y=diabetes.num_medications,
    mode = 'markers',
    #marker_color='indianred'
    marker_color = diabetes.time_in_hospital
))

fig.show()
```

```
[ ]: # express version
# Create a side-by-side bar chart showing number of lab procedures and number
↳ of non lab procedures by gender.

fig = px.bar(d_gender, x='gender', y=['num_lab_procedures', 'num_procedures'],
↳ barmode = 'group')
fig.show()
```

```
[ ]: # go version
fig = 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")
    )
)
fig.show()
```

```
[ ]: d_month = diabetes.
↳ groupby('month')[['time_in_hospital', 'num_lab_procedures', 'num_procedures', 'num_medications
↳ sum().reset_index() # aggregating d_month even more!
```

```
[ ]: d_month.head()
```

```
[ ]: # Create a line chart showing number of number of medications by month.

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

# fig = go.Figure(go.Scatter(x=d_month.month, y=d_month.
↳ num_medications, mode='lines')) DEFAULT is a line
```

```
[ ]: # Create a line chart showing number of number of procedures by month.

#d_month = diabetes.groupby('month').sum().reset_index()
#d_month = d_month.sort_values('month')
fig = px.line(d_month,x='month', y='num_procedures')
fig.show()

# fig = go.Figure(go.Scatter(x=d_month.month, y=d_month.num_procedures,
↪mode='lines'))
```

```
[ ]: # Create a line chart showing number of number of procedures by gender.

#d_gender = diabetes.groupby('gender').sum().reset_index()
#d_month = d_month.sort_values('month')
fig = px.bar(d_gender,x='gender', y='num_procedures')
fig.show()
```