Course: DATS 6401 – Visualization of Complex Data

Instructor: Dr. Reza Jafari

Homework Number: 4

Homework Name: Callbacks using Dash

Name: Rehapriadarsini Manikandasamy

RM
Initials

04.06.22
Date

Abstract

The main emphasis of the assignment is to practice the callback functions in Dash. Three different dash apps were created to understand the usage of different core components of dash. An app with option to display the text typed on input was generated and deployed on GCP. App to generate sine wave and FFT of the sine wave was created using graph and input component. Finally, a dropdown component was used to display the value selected from the options.

CHAPTER 1

INTRODUCTION

The lab helps to understand the working of callback functions in dash. The first app can be generated by having the html components to accommodate the heading and return value of the callback. An input component was created to take values from users. The callback function is set to display the value typed on an input in the output string.

The second app involves selecting the mean, standard deviation, number of sample and number of cycles of a sinusoidal wave in an input component. The output component is two graphs which displays the sine wave and FFT of the sine wave.

The third app helps in the creation of a dropdown menu using dash core component. A html component was used to accommodate the value from callback.

A detailed explanation of methods and packages are explained in the next chapter.

CHAPTER 2

METHOD, THEORY AND PROCEDURES

Dash Callbacks:
functions that are automatically called by Dash whenever an input component's property changes, to update some property in another component (the output).

Dash Core Components:
The Dash Core Components module (dash.dcc) can be imported and used with from dash import dcc and gives you access to many interactive components, including, dropdowns, checklists, and sliders.

Dropdown:
To create a basic dropdown, provide options and a value to dcc.Dropdown in that order.

Graph:
The dcc.Graph component can be used to render any plotly-powered data visualization, passed as the figure argument.

Input:
Number type is now close to native HTML5 input behavior across browsers. We also apply a strict number casting in callbacks: valid number converts into corresponding number types, and invalid number converts into None.

Dash HTML Components:
Dash is a web application framework that provides pure Python abstraction around HTML, CSS, and JavaScript. Instead of writing HTML or using an HTML templating engine, you compose your layout using Python with the Dash HTML Components module (dash.html).

Procedure:

1. Load the necessary libraries
2. Create the First App: Initialize app and Layout. Create Input component to take user input and setup callback with output component
3. Deploy the app in GCP and generate the link
4. Create the second app: Create input components to accommodate the values and generate two graphs using callbacks
5. Deploy the app in GCP and generate the link
6. Create the third app: Create a dropdown component and display the selected option using callback
7. Deploy the app in GCP and generate the link

CHAPTER 3

ANSWERS TO ASKED QUESTIONS

Required libraries:

The following python libraries are required to run the lab code file seamlessly.

```
import dash
from dash import html
from dash import dcc
from dash.dependencies import Input, Output
import plotly.express as px
import pandas as pd
```

1. Using Dash in write a program that creates an input field and displays the entered data as a text on the line bellow. You need to create a callback function for the exercise. Then deploy the created App through GCP and provide the working world web address into your report. Answer:
   GCP Link: https://dashapp-okhupijl3a-ue.a.run.app/

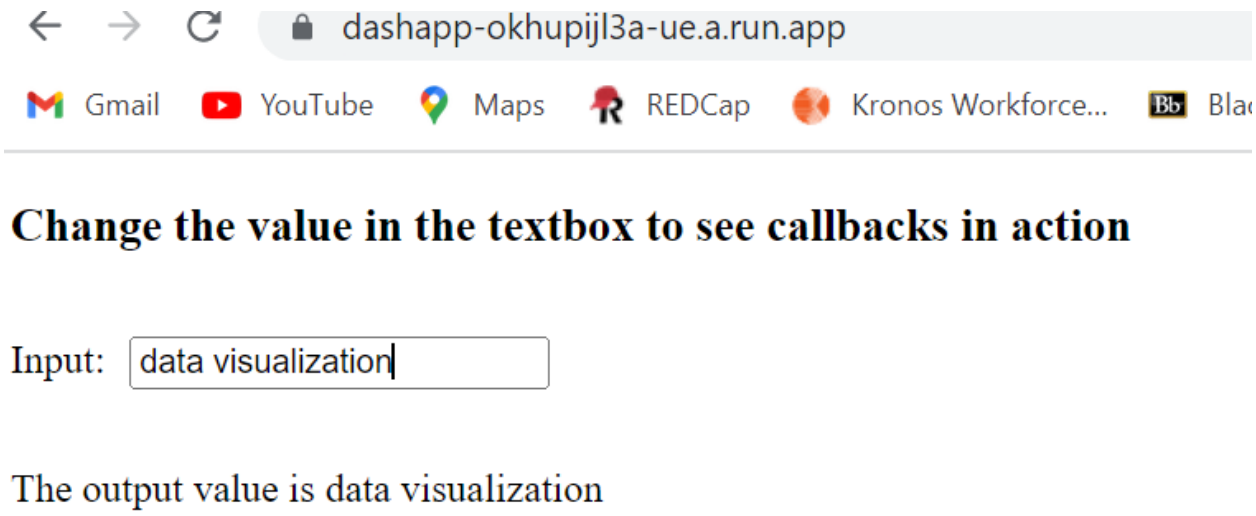

Fig 1: Output for Question 1

2. Using Dash create an app that user can input the followings:
   a. Number of cycles of sinusoidal.
   b. Mean of the white noise.
   c. Standard deviation of the white noise.
   d. Number of samples.
   Then generates the data accordingly (f(x) = sin(x) + noise). Plot the function f(x) and the Fast Fourier Transform (FFT) of the generated data. The range of the x axis is -pi to pi. For tr FFT,

you can use :from scipy.fft import fftThen deploy the created App through GCP and provide the working web address into your report.

Answer:
GCP link was generated due to service unavailability.

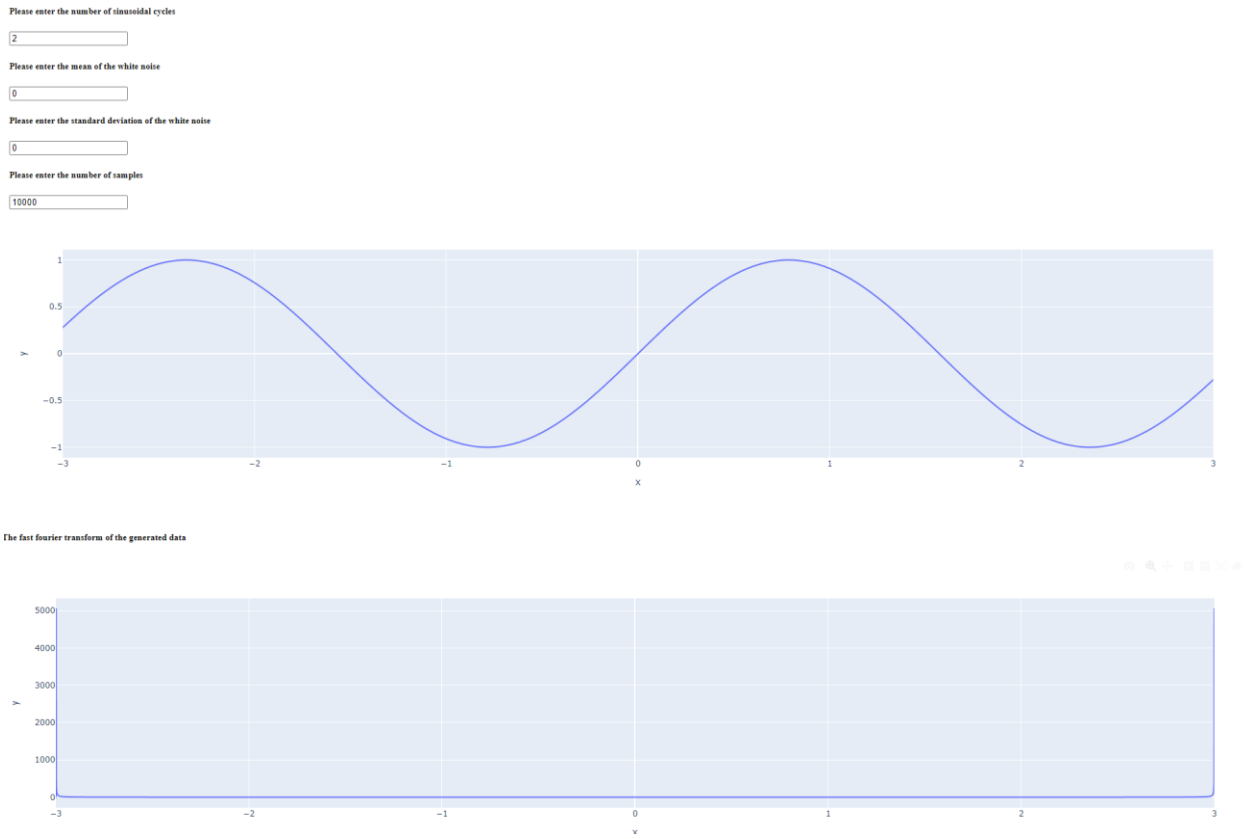

Fig 2: Output for Question 2

3. Using Dash create a drop-down menu with the items listed below. Once one of the items is selected, then a message should display that the selected item inside the dropdown menu is_____. The default must be 'Introduction'. Then deploy the created App through GCP and provide the working web address into your report.

Answer:
GCP link was not generated due to service unavailability.

Fig 3: Output for Question 3

CONCLUSION

The primary objective of this lab is to setup callbacks between input and output components. The first app helps in displaying the value typed on an input textbox. This app was deployed on GCP, and the link was provided in the previous section. The second app helps in simulation of sine wave and FFT. The third app helps in displaying the option selected on the dropdown menu. The last two apps weren’t deployed on GCP due to service unavailability.

APPENDIX

Homework1_Question1.py

```
import dash
from dash import html
from dash import dcc
from dash.dependencies import Input, Output
import plotly.express as px
import pandas as pd

# 1. Using Dash in write a program that creates an input field and displays
the entered data as a text
# on the line bellow. You need to create a callback function for the
exercise. Then deploy the
# created App through GCP and provide the working world web address into your
report.

my_app=dash.Dash('My_App')

my_app.layout=html.Div([html.H3('Change the value in the textbox to see
callbacks in action'),
                        html.P('Input:',style={'display':'inline-
block','margin-right':10}),

dcc.Input(id='input1',type='text',placeholder='',style={'display':'inline-
block'}),
                        html.Br(),html.Br(),
                        html.Div(id='my_out')
                        ])

@my_app.callback(Output(component_id='my_out',component_property='children'),
                 [Input(component_id='input1',component_property='value')])
def update_text(input):
    return f'The output value is {input}'

my_app.run_server(port=8100, host='0.0.0.0')
```

Homework4_question2.py

```
import dash
from dash import html
from dash import dcc
from dash.dependencies import Input, Output
import plotly.express as px
import pandas as pd
from scipy.fft import fft
import numpy as np

# 2. Using Dash create an app that user can input the followings:
# # a. Number of cycles of sinusoidal.
# # b. Mean of the white noise.
# # c. Standard deviation of the white noise.
```

```
# # d. Number of samples.
# # Then generates the data accordingly ( f(x) = sin(x) + noise ). Plot the
function f(x) and the Fast Fourier
# # Transform (FFT) of the generated data. The range of the x axis is -pi to
pi. For tr FFT, you can use :
# # from scipy.fft import fft

# # Then deploy the created App through GCP and provide the working web
address into your report.
my_app=dash.Dash('My_App')

my_app.layout=html.Div(children=[
    html.H5('Please enter the number of sinusoidal cycles'),
    dcc.Input(id='numbers',type='number'),
    html.H5('Please enter the mean of the white noise'),
    dcc.Input(id='mean',type='number'),
    html.H5('Please enter the standard deviation of the white noise'),
    dcc.Input(id='std',type='number'),
    html.H5('Please enter the number of samples'),
    dcc.Input(id='samples',type='number'),
    dcc.Graph(id='graph_sin'),
    html.H5('The fast fourier transform of the generated data'),
    dcc.Graph(id='graph_fft'),

])
@my_app.callback(Output(component_id='graph_sin',component_property='figure')
,
                 Output(component_id='graph_fft',component_property='figure'),
                  [Input(component_id='numbers',component_property='value'),
                   Input(component_id='mean',component_property='value'),
                   Input(component_id='std',component_property='value'),
                   Input(component_id='samples',component_property='value')])
def update_sin(numbers,mean,std,samples):

    noise=np.random.normal(mean,std,size=samples)
    x=np.linspace(-3,3,samples)
    y=np.sin(numbers*x)+noise
    fft_v=abs(fft(y))
    fig=px.line(x=x,y=y)
    fig2 = px.line(x=x, y=fft_v)
    return fig,fig2

my_app.run_server(port=8101, host='0.0.0.0')
```

Homework4_question3.py

```
import dash
from dash import html
from dash import dcc
from dash.dependencies import Input, Output
import plotly.express as px
import pandas as pd
```

```
# 3. Using Dash create a drop-down menu with the items listed below. Once one
of the items is
# selected, then a message should display that the selected item inside the
dropdown menu
# is_____. The default must be 'Introduction'. Then deploy the created App
through GCP and
# provide the working web address into your report.
# a. Introduction
# b. Panda package
# c. Seaborn package
# d. Matplotlib Package
# e. Principal Component Analysis
# f. Outlier Detection
# g. Interactive Visualization
# h. Web-based App using Dash
# i. Tableau

my_app=dash.Dash('My_App')

my_app.layout=html.Div([html.H1('Complex Data Visualization'),
                        dcc.Dropdown(id='drop1',options=[
                            {'label':'Introduction','value':'Introduction'},
                            {'label':'Panda Package','value':'Panda
Package'},
                            {'label':'Seaborn Package','value':'Seaborn
Package'},
                            {'label':'Matplotlib Package','value':'Matplotlib
Package'},
                            {'label':'Principal Component
Analysis','value':'Principal Component Analysis'},
                            {'label':'Outlier Detection','value':'Outlier
Detection'},
                            {'label':'Interactive
Visualization','value':'Interactive Visualization'},
                            {'label':'Web-based App using Dash','value':'Web-
based App using Dash'},
                            {'label':'Tableau','value':'Tableau'}

                        ],value='Introduction'),
                        html.Div(id='my_out')])

@my_app.callback(
    Output(component_id='my_out',component_property='children'),
    [Input(component_id='drop1',component_property='value')]
)

def update_drop(input):
    return f'The selected item inside the dropdown menu is {input}'

my_app.run_server(port=8100, host='0.0.0.0')
```

REFERENCES

- https://dash.plotly.com/dash-core-components/input
- https://dash.plotly.com/dash-core-components/dropdown
- https://dash.plotly.com/dash-core-components/graph