**Let's start creating dash application**

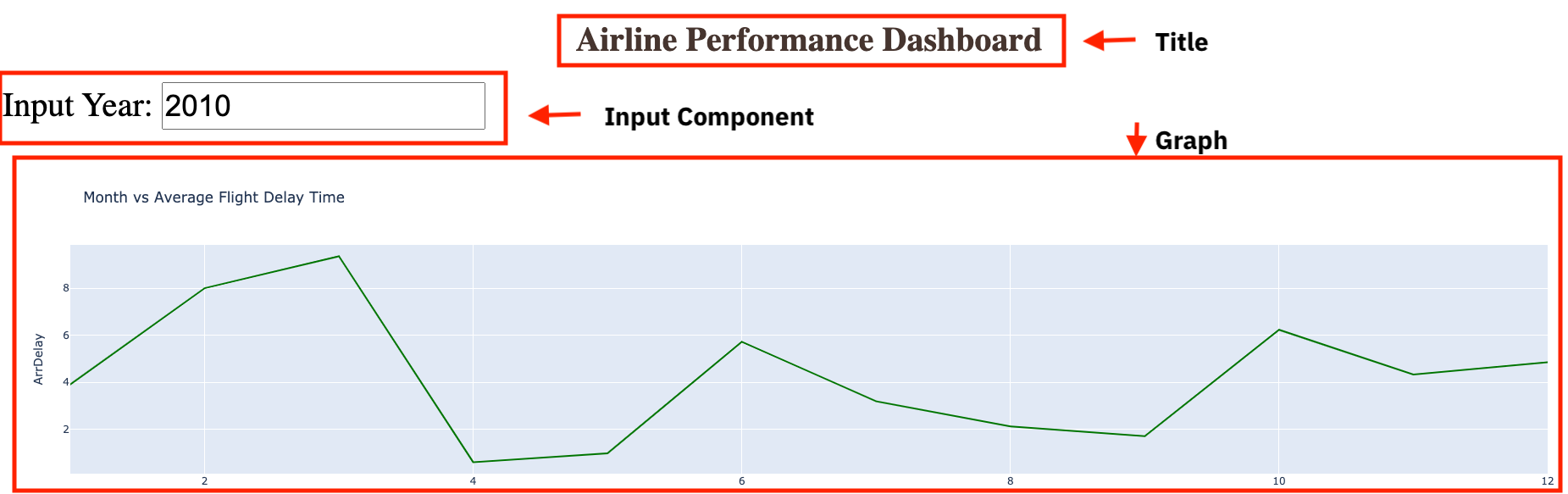
**Theme**

Extract average monthly arrival delay time and see how it changes over the year. Year range is from 2010 to 2020.

**Expected Output**

Below is the expected result from the lab. Our dashboard application consists of three components:

* Title of the application
* Component to enter input year
* Chart conveying the average monthly arrival delay

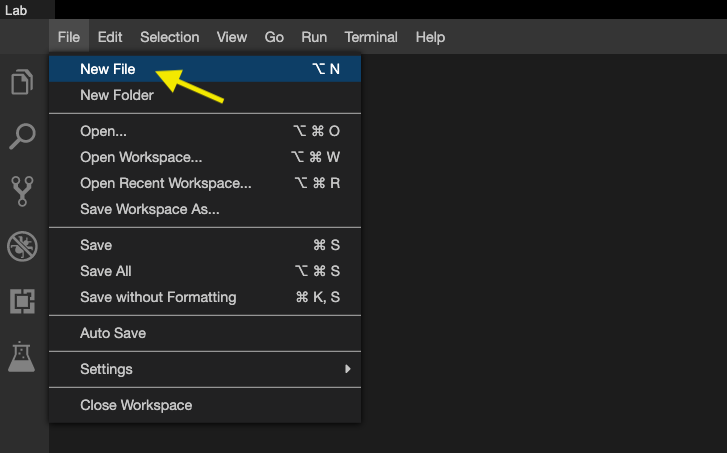


**To do:**

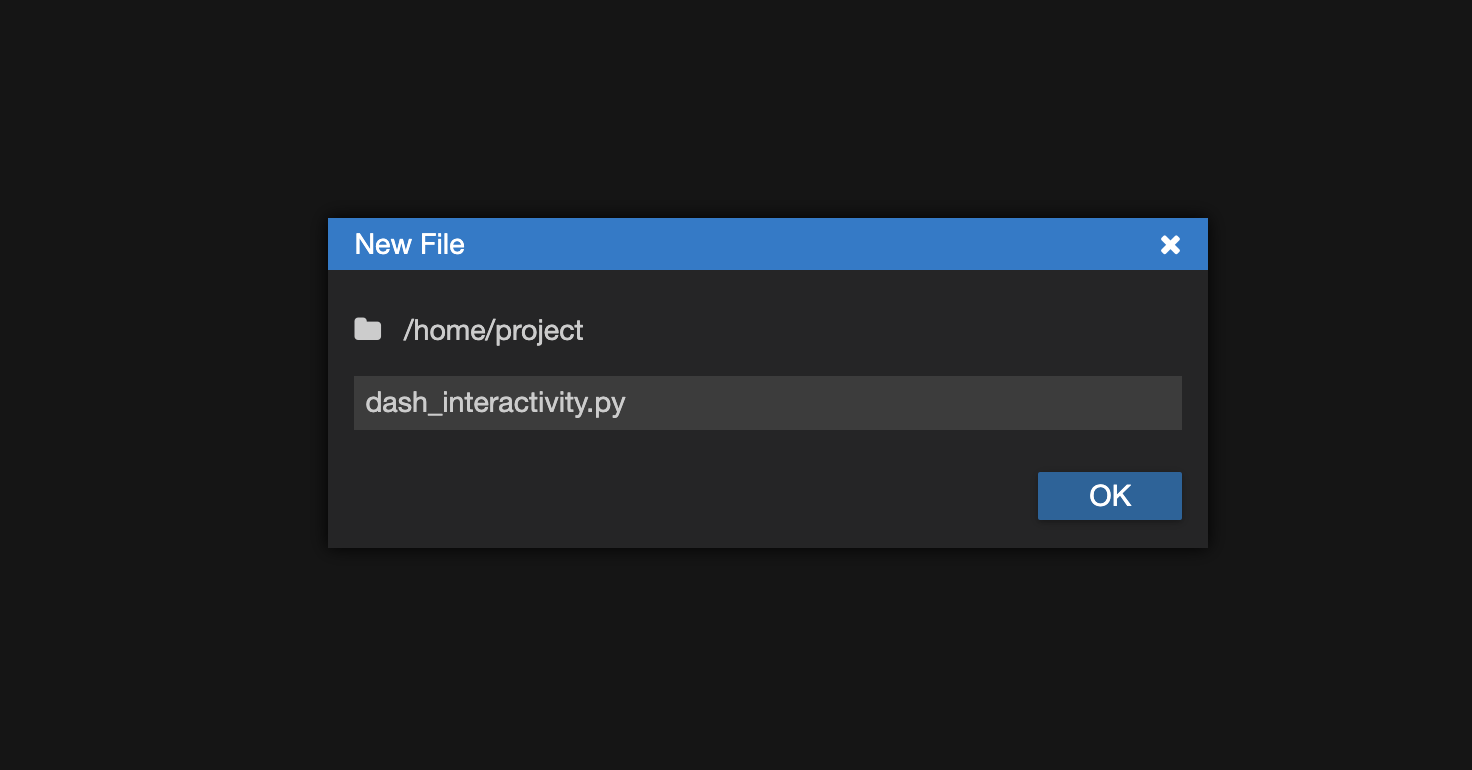
1. Import required libraries and read the dataset
2. Create an application layout
3. Add title to the dashboard application using HTML H1 component
4. Add an input text box using core input component
5. Add the line chart using core graph component
6. Run the app

**Get the tool ready**

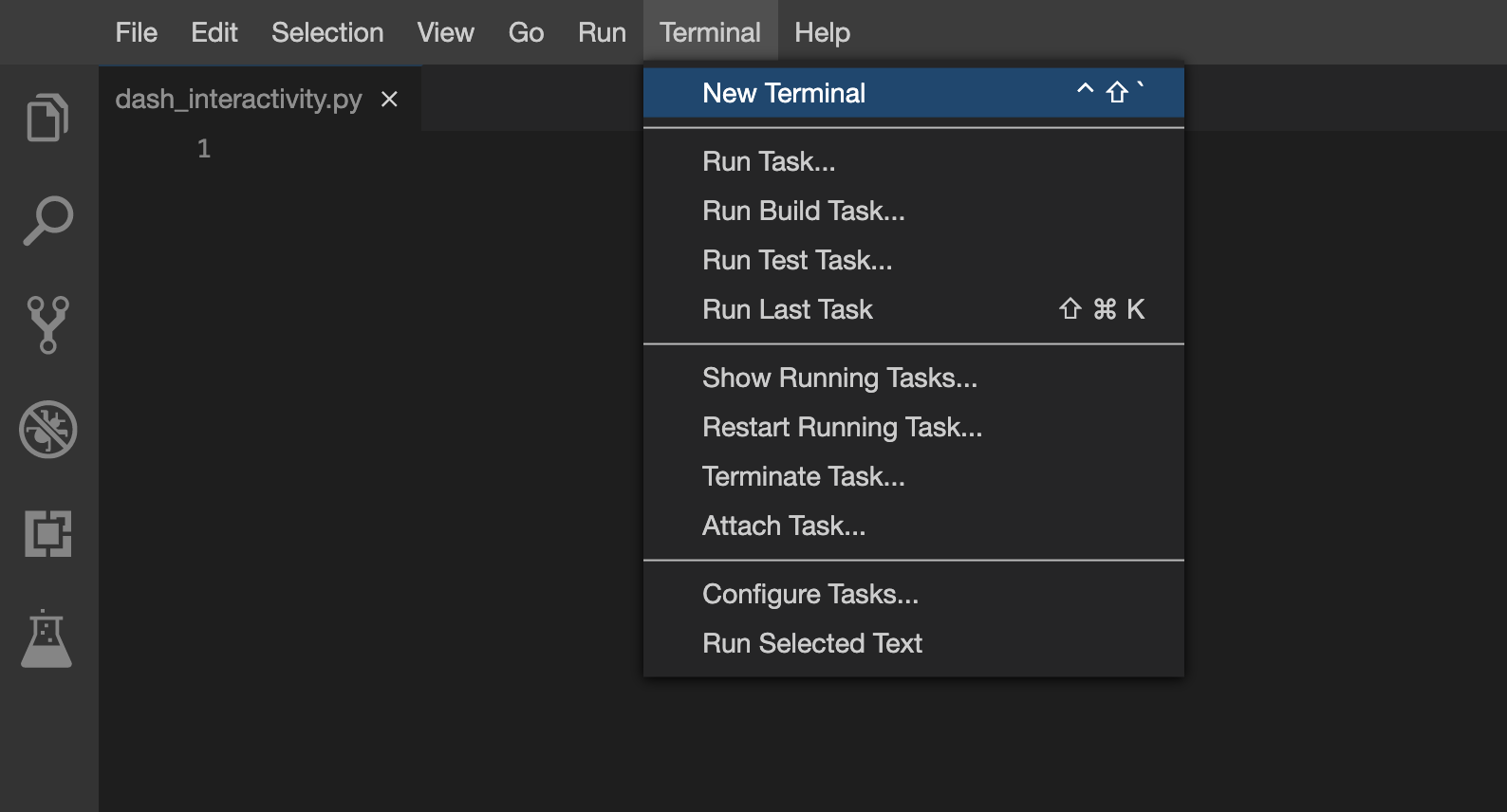
* Create a new python script, by clicking on the menu bar and selecting **File**->**New File**, as in the image below.



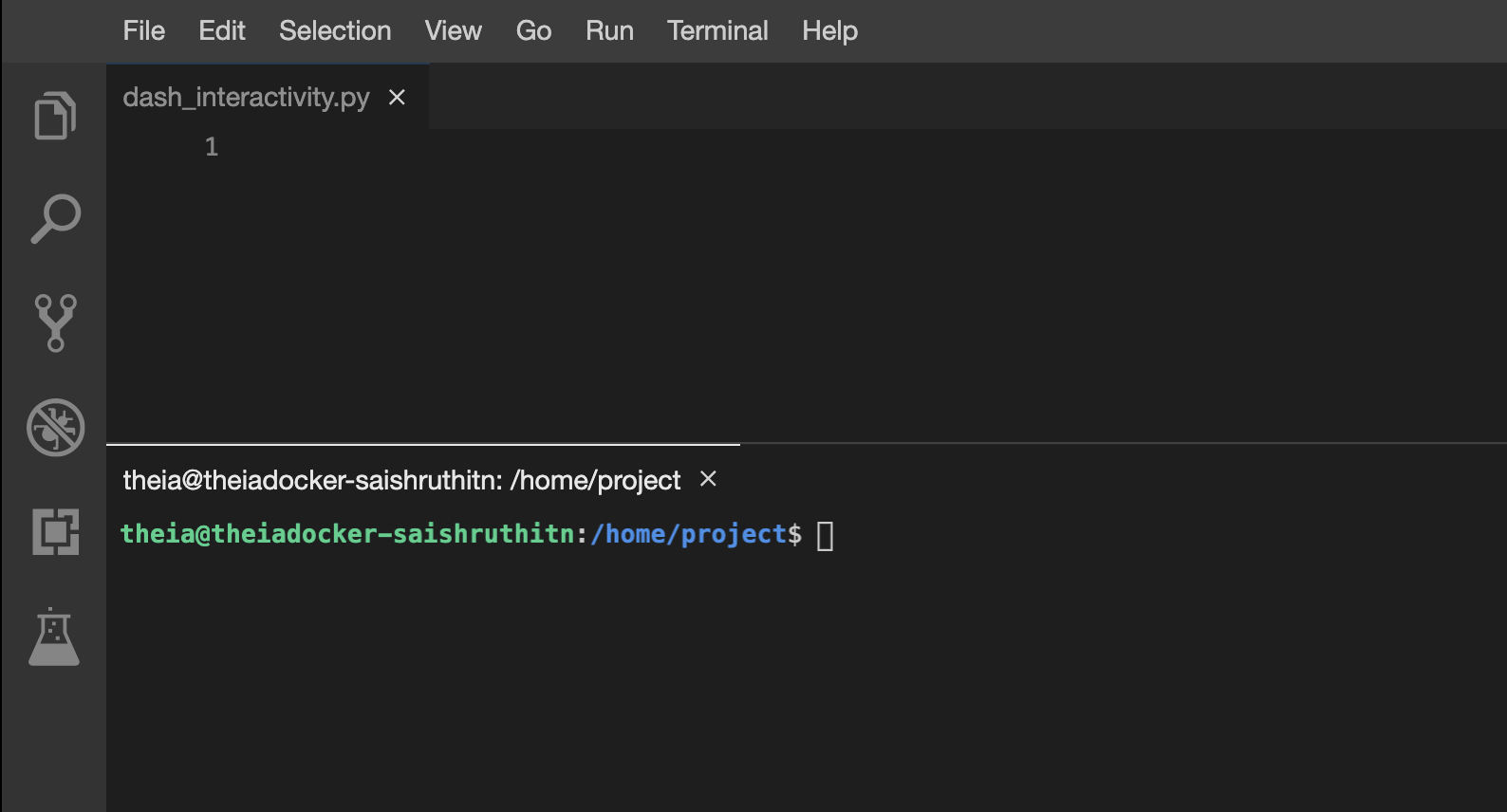
* Provide the file name as dash\_interactivity.py



* Open a new terminal, by clicking on the menu bar and selecting **Terminal**->**New Terminal**, as in the image below.

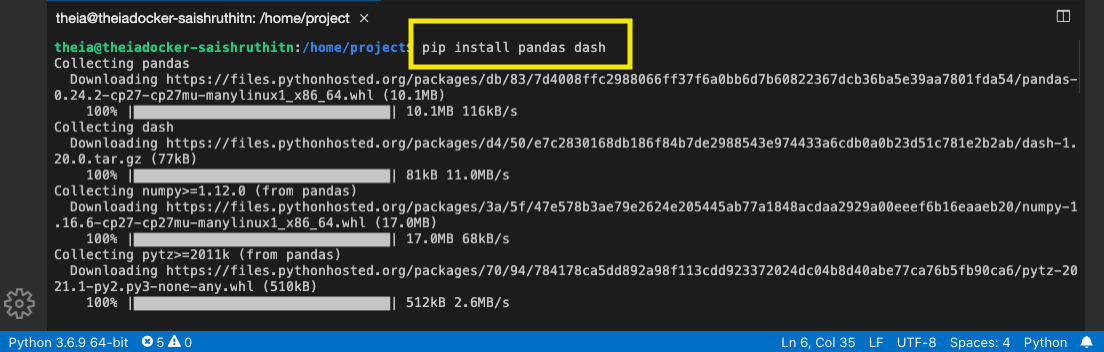


* Now, you have script and terminal ready to start the lab.



* Install python packages required to run the application. Copy and paste the below command to the terminal.

**pip** install pandas dash



**TASK 1 - Read the data**

Let's start with

* Importing necessary libraries
* Reading the data

Copy the below code to the dash\_interactivity.py script and review the code.

# Import required libraries

**import** pandas **as** pd

**import** plotly.graph\_objects **as** go

**import** dash

**import** dash\_html\_components **as** html

**import** dash\_core\_components **as** dcc

**from** dash.dependencies **import** Input, Output

# Read the airline data into pandas dataframe

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

encoding = "ISO-8859-1",

dtype={'Div1Airport': str, 'Div1TailNum': str,

'Div2Airport': str, 'Div2TailNum': str})

**TASK 2 - Create dash application and get the layout skeleton**

Next, we create a skeleton for our dash application. Our dashboard application layout has three components as seen before:

* Title of the application
* Component to enter input year inside a layout division
* Chart conveying the average monthly arrival delay inside a layout division

Mapping to the respective Dash HTML tags:

* Title added using html.H1() tag
* Layout division added using html.Div() and input component added using dcc.Input() tag inside the layout division.
* Layout division added using html.Div() and chart added using dcc.Graph() tag inside the layout division.

Copy the below code to the dash\_interactivity.py script and review the structure.

*NOTE*: Copy below the current code

# Create a dash application

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

# Get the layout of the application and adjust it.

# Create an outer division using html.Div and add title to the dashboard using html.H1 component

# Add a html.Div and core input text component

# Finally, add graph component.

app.layout = html.Div(children=[html.H1(),

html.Div(["Input Year", dcc.Input(),],

style={}),

html.Br(),

html.Br(),

html.Div(),

])

**TASK 3 - Update layout components**

**Application title**

* Heading reference: **[Plotly H1 HTML Component](https://dash.plotly.com/dash-html-components/h1?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkDV0101ENSkillsNetwork20297740-2021-01-01" \t "_blank)**
* Title as Airline Performance Dashboard
* Use style parameter and make the title center aligned, with color code #503D36, and font-size as 40. Check More about HTML section [**here**](https://dash.plotly.com/layout?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkDV0101ENSkillsNetwork20297740-2021-01-01).

**Input component**

* Update **[dcc.Input](https://dash.plotly.com/dash-core-components/input?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkDV0101ENSkillsNetwork20297740-2021-01-01" \t "_blank)** component id as input-year, default value as 2010, and type as number. Use style parameter and assign height of the input box to be 50px and font-size to be 35.
* Use style parameter and assign font-size as 40 for the whole division.

**Output component**

* Add dcc.Graph() component to the second division.
* Update **[dcc.Graph](https://dash.plotly.com/dash-core-components/graph?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkDV0101ENSkillsNetwork20297740-2021-01-01" \t "_blank)** component id as line-plot.

**TASK 4 - Add the application callback function**

The core idea of this application is to get year as user input and update the dashboard in real-time. We will be using callback function for the same.

Steps:

* Define the callback decorator
* Define the callback function that uses the input provided to perform the computation
* Create graph and return it as an output
* Run the application

Copy the below code to the dash\_interactivity.py script and review the structure.

*NOTE*: Copy below the current code

# add callback decorator

@app.callback(Output(),

Input())

# Add computation to callback function and return graph

**def** **get\_graph**(entered\_year):

# Select data based on the entered year

df = airline\_data[airline\_data['Year']==int(entered\_year)]

# Group the data by Month and compute average over arrival delay time.

line\_data = df.groupby('Month')['ArrDelay'].mean().reset\_index()

#

fig = go.Figure(data=)

fig.update\_layout()

**return** fig

# Run the app

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

app.run\_server()

**TASK 5 - Update the callback function**

**Callback decorator**

* Refer examples provided [**here**](https://dash.plotly.com/basic-callbacks?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkDV0101ENSkillsNetwork20297740-2021-01-01)
* Update output component id parameter with the id provided in the dcc.Graph() component and component property as figure.
* Update input component id parameter with the id provided in the dcc.Input() component and component property as value.

**Callback function**

* Update data parameter of the go.Figure() with the scatter plot. Refer [**here**](https://plotly.com/python/line-and-scatter/?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkDV0101ENSkillsNetwork20297740-2021-01-01#scatter-and-line-plot-with-goscatter). Sample syntax below:

**go**.Scatter(x='----', y='----', mode='-----', marker='----)

* Update x as line\_data['Month'], y as line\_data['ArrDelay'], mode as lines, and marker as dict(color='green').
* Update fig.update\_layout with title, xaxis*title, and yaxis*title parameters.
  + Title as Month vs Average Flight Delay Time
  + xaxis\_title as Month
  + yaxis\_title as ArrDelay Refer the update layout function [**here**](https://plotly.com/python/line-and-scatter/?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkDV0101ENSkillsNetwork20297740-2021-01-01#style-scatter-plots).

# Import required libraries

import pandas as pd

import plotly.graph\_objects as go

import dash

import dash\_html\_components as html

import dash\_core\_components as dcc

from dash.dependencies import Input, Output

# Read the airline data into pandas dataframe

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

encoding = "ISO-8859-1",

dtype={'Div1Airport': str, 'Div1TailNum': str,

'Div2Airport': str, 'Div2TailNum': str})

# Create a dash application

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

app.layout = html.Div(children=[ html.H1('Airline Performance Dashboard',

style={'textAlign': 'center', 'color': '#503D36',

'font-size': 40}),

html.Div(["Input Year: ", dcc.Input(id='input-year', value='2010',

type='number', style={'height':'50px', 'font-size': 35}),],

style={'font-size': 40}),

html.Br(),

html.Br(),

html.Div(dcc.Graph(id='line-plot')),

])

# add callback decorator

@app.callback( Output(component\_id='line-plot', component\_property='figure'),

Input(component\_id='input-year', component\_property='value'))

# Add computation to callback function and return graph

def get\_graph(entered\_year):

# Select 2019 data

df = airline\_data[airline\_data['Year']==int(entered\_year)]

# Group the data by Month and compute average over arrival delay time.

line\_data = df.groupby('Month')['ArrDelay'].mean().reset\_index()

fig = go.Figure(data=go.Scatter(x=line\_data['Month'], y=line\_data['ArrDelay'], mode='lines', marker=dict(color='green')))

fig.update\_layout(title='Month vs Average Flight Delay Time', xaxis\_title='Month', yaxis\_title='ArrDelay')

return fig

# Run the app

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

app.run\_server()