Student name:

Student ID:

SIT225: Data Capture Technologies

Q1. Perform week 6 activities mentioned in the unit site and produce outputs.

I completed all the week 6 activities from the SIT225 unit site, including exploring Plotly Dash, creating dashboards, using interactive controls such as radio buttons, and generating visualizations with both DataTables and graphs. I modified the Hello World example, connected to a dataset, explored different aggregation functions, and changed the graph type to a line graph as required. Screenshots of each output have been captured to demonstrate my work.

Q2. Paste Plotly Dash Python code and explain the program steps, how plots were generated,

and interaction was handled. Provide insight on each plot w.r.t. into the data segments.

from dash import Dash, html, dash\_table, dcc, callback, Output, Input

import pandas as pd

import plotly.express as px

# Load dataset

df = pd.read\_csv('https://raw.githubusercontent.com/plotly/datasets/master/gapminder2007.csv')

# Initialize Dash app

app = Dash()

# App layout

app.layout = [

html.Div(children='My First App with Data, Graph, and Controls'),

html.Hr(),

dcc.RadioItems(options=['pop', 'lifeExp', 'gdpPercap'], value='lifeExp', id='controls-and-radio-item'),

dash\_table.DataTable(data=df.to\_dict('records'), page\_size=6),

dcc.Graph(figure={}, id='controls-and-graph')

]

# Callback for interaction

@callback(

Output(component\_id='controls-and-graph', component\_property='figure'),

Input(component\_id='controls-and-radio-item', component\_property='value')

)

def update\_graph(col\_chosen):

fig = px.line(df, x='continent', y=col\_chosen, title=f'Line Graph of {col\_chosen}')

return fig

# Run the app

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

app.run(debug=True, jupyter\_mode="tab")  
  
The program begins by loading the gapminder2007.csv dataset using pandas, which allows us to work with the data in Python. A Dash app is then initialized with Dash(), and the layout is defined to include several components: html.Div displays the title of the app, dcc.RadioItems provides interactive controls for selecting a variable to visualize (population, life expectancy, or GDP per capita), dash\_table.DataTable presents the dataset in a paginated table, and dcc.Graph displays the selected graph. The callback function captures the input from the radio buttons (col\_chosen), generates a line graph using Plotly Express, and updates the graph component dynamically whenever the user selects a different option. The resulting line graphs show aggregated data per continent, making it easy for users to compare trends in population, life expectancy, or GDP per capita across continents.

Q3. Create a video in Panopto/CloudDeakin showing your program execution, graph output, user

interaction and share the video link here.

<https://drive.google.com/file/d/1OPZbHsZUmnBokzoFrBpMSOx0X1C4kbtR/view?usp=sharing>

Q4. Create a subdirectory ‘week-6’ under directory ‘SIT225\_<YYYY>T2’ in your drive where you

copy the Python script file, Arduino sketch file if any, data file and the generated graphs. Commit

and push changes to GitHub. Include the link to your repository here with a GitHub page

screenshot of weekly folder content. A tutor may try to access your GitHub link, if necessary. Give

access to your tutor by adding the tutor’s email address as a collaborator of your private

repository.

# Activity 6.1: Plotly data dashboard

Plotly Dash apps give a point-&-click interface to models written in Python, vastly expanding the notion of what's possible in a traditional "dashboard". With Dash apps, data scientists and engineers put complex Python analytics in the hands of business decision-makers and operators. In this activity, you will learn basic building blocks of Plotly to create Dash apps.

## Hardware Required

* No hardware is required.

## Software Required

Plotly library and Dash module

Python 3

## Steps

|  |  |
| --- | --- |
| **Step** | **Action** |
| 1 | Install Plotly and dash using the command below in the command line.  $ pip install plotly dash  You can download Jupyter Notebook from here (<https://github.com/deakin-deep-dreamer/sit225/blob/main/week_6/plotly_explore.ipynb> ) and run all the cells. The Notebook contains multiple sections such as Hello World which follows a sample code in a following cell. If you run the Hello world cell it will show Plotly Dash web page. The cell also includes a Question (#\*\*\* Question) which you will need to carry out to get a modified output. You will need to capture the output and share the screenshot in the following steps. |
| 2 | Question: **Hello World** cell has a question - add another html.Div to show your name, and re-run the cell for output. You will need to update the code, run the cell, capture the screenshot of the output and paste it here.  Answer: <Your answer>  A screenshot of a computer  AI-generated content may be incorrect. |
| 3 | Question: **Connecting to Data** cell has a question - change page size and observe the change in widget controls such as, total number of pages. You will need to update the code, run the cell, capture the screenshot of the output and paste it here.  Answer: <Your answer>  A screenshot of a computer code  AI-generated content may be incorrect.  A screenshot of a computer  AI-generated content may be incorrect. |
| 4 | Question: **Visualising data** cell has a question - explore another histfunc other than 'avg' used above and observe behaviour. You will need to update the code, run the cell, capture the screenshot of the output and paste it here.  Answer: <Your answer>A screenshot of a graph  AI-generated content may be incorrect. |
| 5 | Question: **Controls and Callbacks** cell has a question - use line graphs instead of histogram. You will need to update the code, run the cell, capture the screenshot of the output and paste it here.  Answer: <Your answer> |
| 6 | Question: Now you have learned how to use Plotly Dash for visualising your data, describe how you will be using this tool for your desired sensor monitoring dashboard with a number of sensors including DHT22 or accelerometer data.  Answer: <Your answer>  I would use Plotly Dash to build a real-time sensor monitoring dashboard for multiple sensors, such as a **BMP280 pressure sensor** for measuring air pressure and temperature, and a **gyroscope** for detecting orientation or movement. The dashboard would display live line graphs for each sensor, showing trends over time. Interactive controls like dropdowns or radio buttons would allow selecting specific sensors, time periods, or locations. Callbacks in Dash would update the graphs automatically as new sensor data is received. This would make it easier to monitor environmental conditions, analyse patterns, and detect anomalies efficiently. |
| 7 | Question: Convert the Notebook to PDF and merge with this activity sheet PDF. You will need this merged PDF to combine with this week’s OnTrack task for submission.  Answer: <Your answer> |