

Growth Locations Dropdown - Hint and Updated Solution

AI-Generated Hint

Have you considered how to set the visibility for each state in the dropdown buttons? Remember, you need to specify which traces should be visible for each button option.

```
```python
dropdown_buttons = [
 {'label': "ALL", 'method': "update", 'args': [{"visible": [True, True, True]},
 {"title": "ALL"}]},
 {'label': "NSW", 'method': "update", 'args': [{"visible": [True, False,
False]}, {"title": "NSW"}]},
 {'label': "QLD", 'method': "update", 'args': [{"visible": [False, True,
False]}, {"title": "QLD"}]},
 {'label': "VIC", 'method': "update", 'args': [{"visible": [False, False, True]},
 {"title": "VIC"}]},
]
```
```

If this hint doesn't help out, take a look at the solution.

Updated Question

The Australian Government is looking to understand which Local Government Areas (LGAs) have had recent strong population growth to assist in planning infrastructure projects.

They have provided you with some data on the top 5 LGAs by the percentage population increase (from 2018 to 2019) and have asked if you can visualize it. However, they want to be able to select a certain state or see everything at once.

In this exercise, you are tasked with creating a bar chart of this data with a drop-down menu to switch between different states and see all states at once.

Use the dropdown buttons to explicitly set the visibility for `ALL`, `NSW`, `QLD`, or `VIC`, and update the chart title dynamically.

Updated Question Explanation

This task requires creating a bar chart using `plotly.graph_objects` for population growth data. Dropdown buttons must be added to toggle the

visibility of traces for 'ALL', 'NSW', 'QLD', and 'VIC', and dynamically update the chart title.

Updated Solution

Here is the updated solution:

```
```python
import plotly.graph_objects as go
import pandas as pd

Assume pop_growth DataFrame is already loaded with columns:
'State', 'Local Government Area', 'Change %'

Create the basic figure
fig = go.Figure()

Loop through the states
states = ['NSW', 'QLD', 'VIC']
for state in states:
 # Subset the DataFrame
 df = pop_growth[pop_growth.State == state]

 # Add a trace for each state subset
 fig.add_trace(go.Bar(
 x=df['Local Government Area'],
 y=df['Change %'],
 name=state
))

Define dropdown buttons with explicit visibility
dropdown_buttons = [
 {
 'label': "ALL",
 'method': "update",
 'args': [{"visible": [True, True, True]}, {"title": "Population Growth: ALL
States"}]
 },
 {
 'label': "NSW",
 'method': "update",
 'args': [{"visible": [True, False, False]}, {"title": "Population Growth:
NSW"}]
 },
 {
```

```

 'label': "QLD",
 'method': "update",
 'args': [{"visible": [False, True, False]}, {"title": "Population Growth:
QLD"}]
 },
 {
 'label': "VIC",
 'method': "update",
 'args': [{"visible": [False, False, True]}, {"title": "Population Growth:
VIC"}]
 }
]

Add the dropdown menu to the layout
fig.update_layout(
 updatemenus=[{
 'buttons': dropdown_buttons,
 'direction': 'down',
 'showactive': True
 }],
 title="Population Growth: ALL States"
)

Show the plot
fig.show()
```

```

Updated Solution Explanation

1. The `pop_growth` DataFrame is iterated over to filter data for each state ('NSW', 'QLD', 'VIC').
2. For each state, a bar chart trace is added to the figure, displaying the population growth percentage for its LGAs.
3. Dropdown buttons are created to explicitly set the visibility for:
 - 'ALL' states combined.
 - Individual states ('NSW', 'QLD', 'VIC').
4. Each button updates the `visible` property of the traces and dynamically updates the chart title.
5. The layout is updated with the dropdown menu for user interactivity.