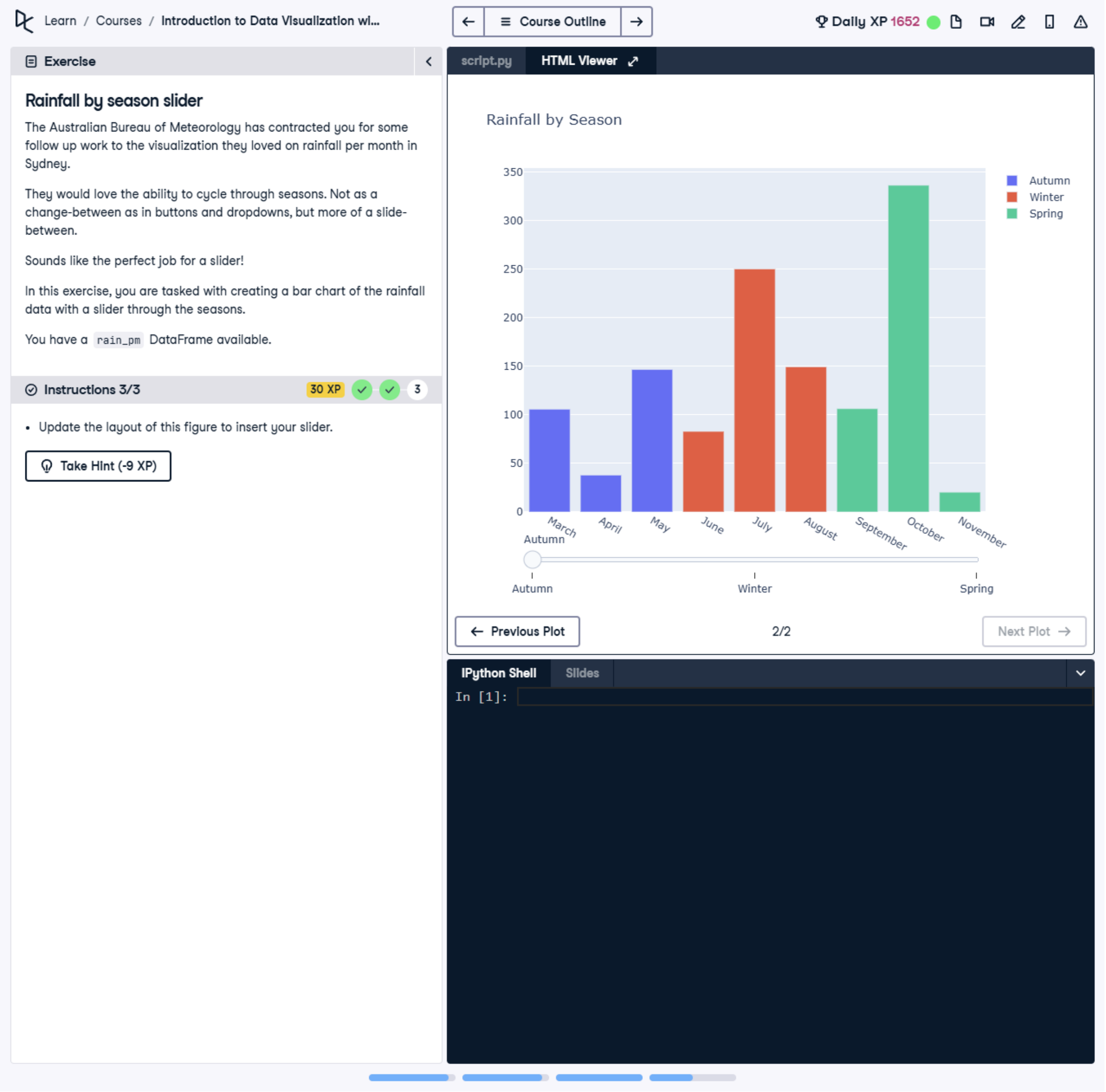
Rainfall by Season Slider - Final Solution



Rainfall by season slider

The Australian Bureau of Meteorology has contracted you for some follow up work to the visualization they loved on rainfall per month in Sydney.

They would love the ability to cycle through seasons. Not as a change-between as in buttons and dropdowns, but more of a slide-between.

Sounds like the perfect job for a slider!

In this exercise, you are tasked with creating a bar chart of the rainfall data with a slider through the seasons.

You have a `rain_pm` DataFrame available.

Instructions 3/3

- Update the layout of this figure to insert your slider.

```
In [1]:
```

Final Question

The Australian Bureau of Meteorology has contracted you for some follow-up work to the visualization they loved on rainfall per month in Sydney.

They would love the ability to cycle through seasons. Not as a change-between as in buttons and dropdowns, but more of a slide-between.

Sounds like the perfect job for a slider!

In this exercise, you are tasked with creating a bar chart of the rainfall data with a slider through the seasons. Ensure that the slider is correctly integrated into the layout and functions smoothly.

Final Question Explanation

This task involves creating a bar chart using `plotly.graph_objects` with a slider to dynamically transition through seasons ('Autumn', 'Winter', 'Spring') while displaying rainfall data. The slider should work seamlessly to update the traces for each season.

Final Solution

```
import plotly.graph_objects as go

# Create the basic figure
fig = go.Figure()

# Loop through the seasons
for season in ['Autumn', 'Winter', 'Spring']:
    # Subset the DataFrame
    df = rain_pm[rain_pm.Season == season]

    # Add a trace for each season
    fig.add_trace(go.Bar(
        x=df['Month'],
        y=df['Total Rainfall'],
        name=season
    ))

# Create the slider
sliders = [
    {
        'steps': [
            {
                'method': 'update',
                'label': 'Autumn',
                'args': [{'visible': [True, False, False]}, {'title': 'Rainfall in
Autumn'}]
            },
            {
                'method': 'update',
                'label': 'Winter',
                'args': [{'visible': [False, True, False]}, {'title': 'Rainfall in
Winter'}]
            },
            {
                'method': 'update',
                'label': 'Spring',
                'args': [{'visible': [False, False, True]}, {'title': 'Rainfall in
Spring'}]
            }
        ]
    }
]

# Add the slider to the layout
fig.update_layout(
    sliders=sliders,
    title='Rainfall by Season'
)

# Show the plot
fig.show()
```

Final Solution Explanation

1. The `rain_pm` DataFrame contains rainfall data split by seasons ('Autumn', 'Winter', 'Spring') and months.
2. A bar chart is created for each season using `go.Bar`.
3. A slider is defined with `steps`, where each step updates the visibility of the traces for a specific season.
4. The `args` parameter in each step dynamically updates the `visible` traces and the chart title.
5. The layout is updated to include the slider, enabling smooth transitions between seasons with appropriately labeled steps.