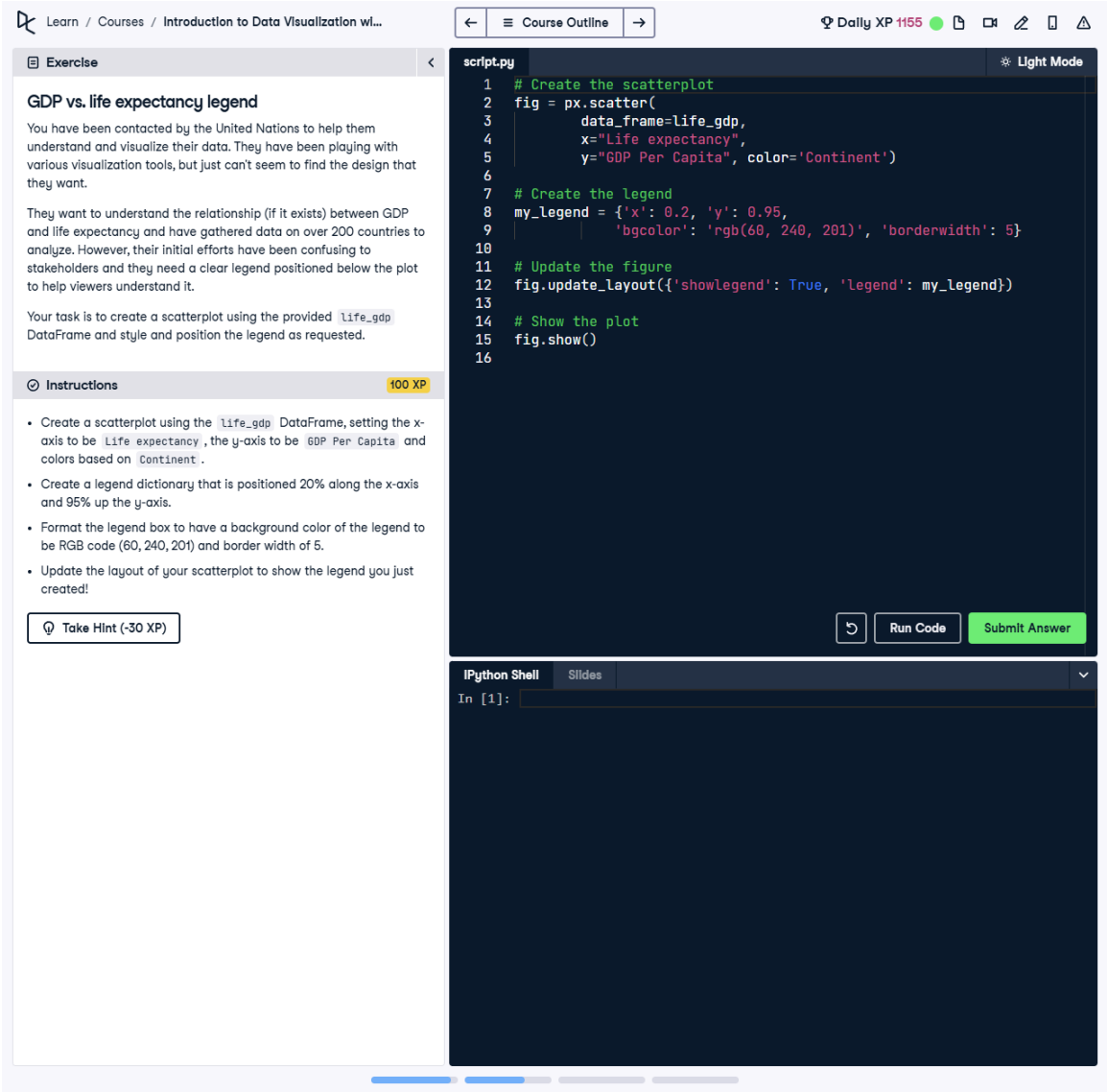
GDP vs. Life Expectancy Legend (Updated)



Question:

You have been contacted by the United Nations to help them understand and visualize their data. They have been playing with various visualization tools, but just can’t seem to find the design that they want.

They want to understand the relationship (if it exists) between GDP and life expectancy and have gathered data on over 200 countries to analyze. However, their initial efforts have been confusing to stakeholders and they need a clear legend positioned below the plot to help viewers understand it.

Your task is to create a scatterplot using the provided `life_gdp` DataFrame

and style and position the legend as requested.

Instructions:
1. Create a scatterplot using the `life_gdp` DataFrame, setting the x-axis to be `Life Expectancy`, the y-axis to be `GDP Per Capita`, and colors based on `Continent`.
2. Create a legend dictionary that is positioned 20% along the x-axis and 95% up the y-axis.
3. Format the legend box to have a background color of the legend to be RGB code (60, 240, 201) and a border width of 5.
4. Update the layout of your scatterplot to show the legend you just created.
5. Display the plot using `fig.show()`.

Answer:

```
# Create the scatterplot
fig = px.scatter(
    data_frame=life_gdp,
    x='Life expectancy',
    y='GDP Per Capita',
    color='Continent'
)

# Create the legend
my_legend = {
    'x': 0.2,
    'y': 0.95,
    'bgcolor': 'rgb(60, 240, 201)',
    'borderwidth': 5
}

# Update the figure
fig.update_layout({
    'showlegend': True,
    'legend': my_legend
})

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

Explanation of the Answer:

The scatterplot is created using the `px.scatter()` function, with `Life expectancy` on the x-axis, `GDP Per Capita` on the y-axis, and `Continent` as the color variable. The legend is positioned using the `my_legend` dictionary, specifying its `x` and `y` coordinates, background color

(`bgcolor`), and border width (`borderwidth`). The layout is updated using `fig.update_layout()` to apply the legend customizations. Finally, the plot is displayed using `fig.show()`.