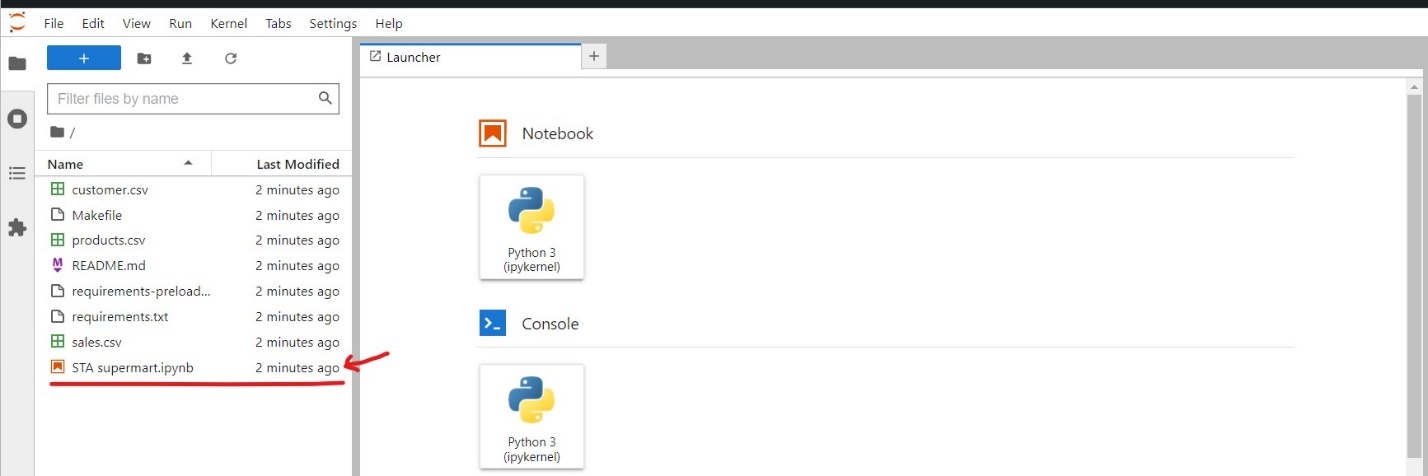
1. **State-wise Sales Analysis: Bar Chart Visualization**

Your task is to find out the total sales of each state by creating a horizontal bar chart of state vs. sales to gain insights into the sales performance across different states. The bars should be sorted in a way so that the state with maximum sales value is on the top and the state with lowest sales value is at the bottom.

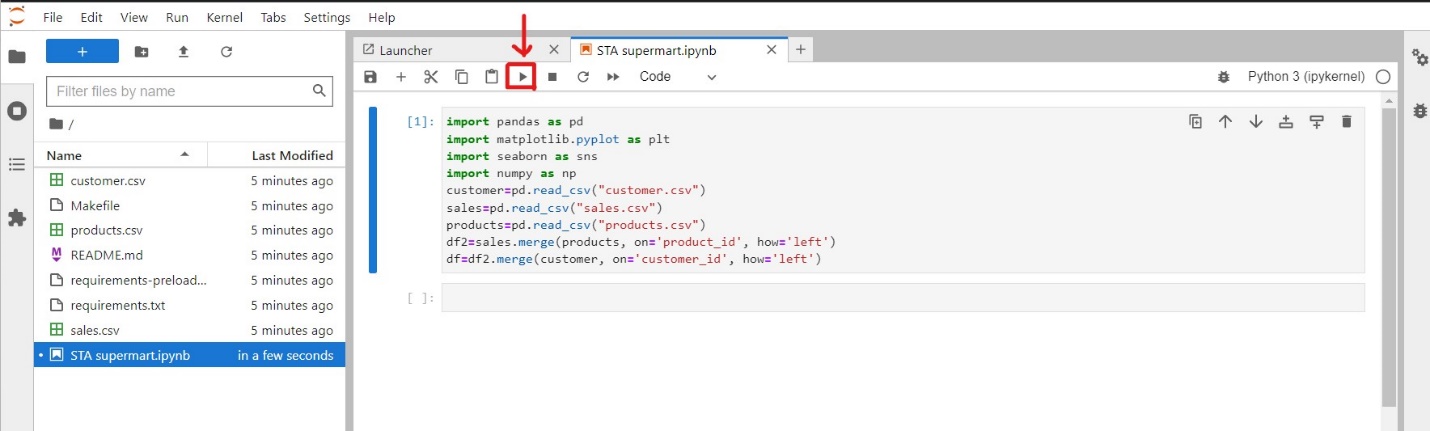
**Example approach\_FirstQuestion**

1: First, we need to setup the environment where we can run our Python code. For that, launch the workspace.

2: Open the Jupyter notebook titled "STA supermart.ipynb"



3: Run the first cell to import the libraries and to get the dataset from the available csv files into a variable called 'df'.



4: Create a new DataFrame called 'bar' by selecting the 'state' and 'sales' columns from the original DataFrame df.

bar = df.loc[:, ['state', 'sales']]

5: Group the 'bar' DataFrame by the 'state' column and calculate the sum of the 'sales' column for each state.

bar\_df = bar.groupby('state').sum().reset\_index()

6: Sort the values in bar\_df so that the bars in the bar chart are also sorted as per sales.

bar\_df = bar\_df.sort\_values(by=['sales'])

7: Set a larger size for the plot as there are a lot of states and a smaller plot will look cluttered.

plt.figure(figsize=(9,9))

8: In the same code block, use plt.barh to create a horizontal bar chart using the 'state' column as the x-axis values and the 'sales' column as the y-axis values.

plt.barh(bar\_df['state'], bar\_df['sales'])

9: In the same code block, add a title to the chart using plt.title('State vs Sales').

plt.title('State vs Sales')

10: In the same code block, add labels for the x-axis and y-axis using plt.xlabel('State') and plt.ylabel('Sales'), respectively.

1. plt.xlabel('Sales')
2. plt.ylabel('State')

11: In the same code block, display the chart using plt.show().

plt.show()

You output should look like this:

