Navigating JupyterLab

# Start Anaconda Navigator & launch JupyterLab which runs in your browser.

1. Set the project name to Lab 1
2. Use the Settings menu to determine if the theme for JupyterLab is set to Dark or Light. Then, change the theme to see if you like it better. If not, change it back.
3. Open the Polling Notebook and run the first three cells 3.
4. Use the Kernel menu to restart the kernel and clear all outputs.
5. Click in the first cell that contains code and run the cell by clicking on the Run button in the toolbar.
   * This code uses the Pandas **read\_csv()** method to import polling data from the FiveThirtyEight website into a **DataFrame** named polls.
6. With the cursor in the next cell, press Shift+Enter to run the statement in that cell and see how it displays the data in the polls DataFrame.
7. Next, change the code to: **polls.head()** Then, run the cell and note that the head() method displays just the first five rows in the DataFrame.
8. Run the **sort\_values()** method in the third cell. Note that this method has one positional parameter (state) and one keyword parameter.
9. Fix syntax and runtime errors 8. Still in the third cell, delete the e in state to make it stat, and delete the right parenthesis at the end of the **sort\_values()** method.

polls.rename(columns={'state':'stat'},inplace=True)

polls.head()

1. Run that cell and note that the error message says that a syntax error occurred. Fix that by replacing the right parentheses and run the cell again.
2. This time, there’s a runtime error because there’s no column named “stat”. Fix that and run the cell again. Now, the sort should work.

# Use Markdown language, Tab completion, and tooltips

1. Add a new cell after the one for the **sort\_values()** method.
   * Use Markdown language to create a subheading that says: “Use Tab completion and tooltips”.
2. Add a cell after the one you just created and enter:
   * polls and press the Tab key.
3. Enter a set of parentheses after the **sort\_values** method name, and with the cursor in the parentheses, press **Shift+Tab** to display the tooltip. Note that the signature lists one positional parameter (by) and five keyword parameters. Then, scroll through the tooltip to see all that it offers.

A screenshot of a computer

Description automatically generated

1. Finish the **sort\_value()** method so it looks like this: **sort\_values('state', ascending=False)**
   * Next, run the statement to see how it displays the results.
   * Then, chain a **head()** method to the **sort\_values()** method, and run it to see the results.
2. Try the tooltip feature for the **sort\_values()** method that you just coded, and see that it still works. But try it for the **head()** method, and you’ll see that it only works for the first method in a chain.
3. Run the rest of the cells 16 and run the rest of the cells in the Notebook.
   * To do that, press Shift+Enter for each cell or click on the Run button in the toolbar. (If the first cell for plotting doesn’t display a plot when you run it, run it a second time.)
4. Review the code in each of the cells that you’ve just run. Note the use of a dictionary in the **rename()** method and the use of lists in some of the other statements. Note too that the **plot()** method in the last cell of this Notebook is chained to the **query()** method.

# Use two Magic Commands and the Python type() function

1. Add the **%%time** Magic Command as the first line in the cell for the chained **plot()** method. Then, run the cell to see how long it takes.
2. In the new cell at the end of the Notebook, run the **%whos** command and note the data types for all the variables that have been created by this Notebook.

A screenshot of a computer

Description automatically generated

1. In the next cell, run the Python **type()** function for the polls DataFrame, which is another way to identify the data type for a variable.

# Start a new Notebook, use a split screen, and copy code into it

1. Use File/New Launcher to open a new tab in JupyterLab and click on the first Python 3 icon to start a new Notebook.
2. Right-click on the tab for the new file, which will say “Untitled.ipynb”, select Rename in the popup menu, and change the name to Lab 1-1.
3. Select the first four cells in the Lab 1 Notebook (the two heading cells and the two code cells), right-click on them, and select Copy Cells from the popup menu. Then, go to the tab for the new Notebook, right-click on the empty first cell, and select Paste Cells Below.
4. After you paste the cells into the new Notebook, the first cell in the notebook will be empty. To delete that cell, right-click on it and select Delete Cells from the popup menu.
5. Split the tabs for the two Notebooks vertically by dragging Lab1-1 to the left side of Lab 1. Then, copy the next two or three cells from the first Notebook to the new Notebook.
6. Restore the split screens to a single screen.

# Experiment on your own

1. If you have the time and interest, experiment with some of the other ways that JupyterLab provides for getting the results you want. Otherwise, close both Notebooks