

Learning Objectives

Students will be able to...

- **Write and execute python code in a Jupyter Notebook**
- **Write Markdown-formatted text in a Jupyter Notebook**
- **Create a matplotlib visualization in a Jupyter Notebook**

definition

Assumptions

- Learners are comfortable reading and importing CSV data sets, extracting relevant data into data frames, and printing that data to the console.
- Learners are comfortable using ggplot2 to create visualization charts.

Jupyter Notebook

In order to easily see our matplotlib plots, we are going to use Jupyter Notebooks. A Jupyter notebook has been created and opened for you on the left.

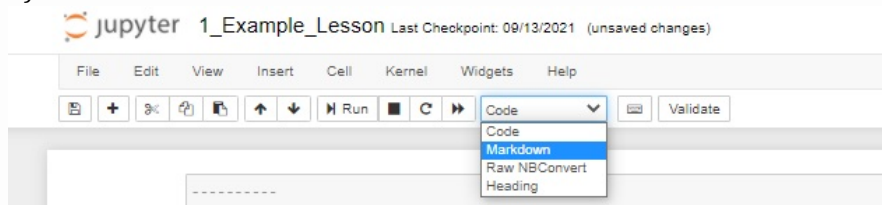
What is Jupyter?

Jupyter is a web application that:

- Runs code
- Embeds visualizations
- Allows Markdown-formatted explanatory text

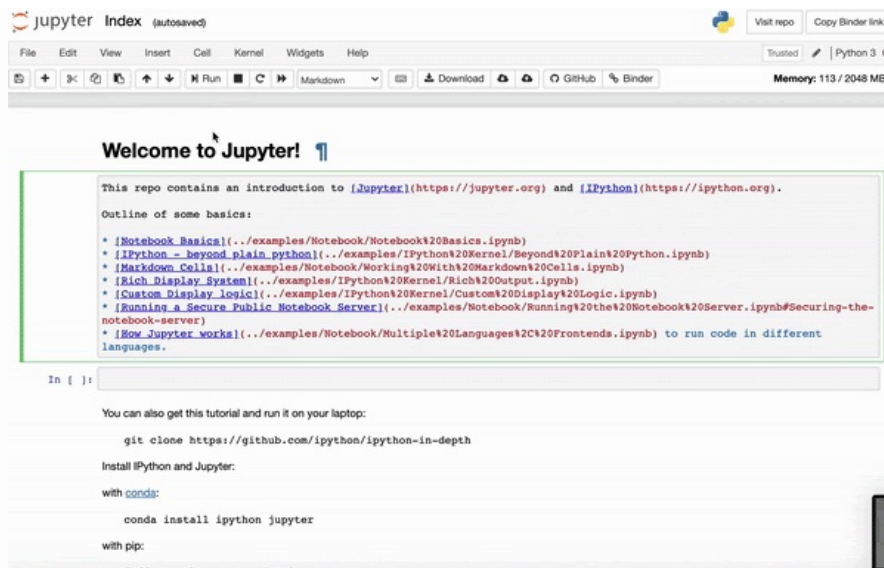
A Jupyter Notebook has the file extension `.ipynb` which stands for **interactive python notebook**.

Everything on the left is editable and runnable. Cells can contain either Python code or Markdown-formatted text.



This means you can combine explanatory text with python code and the resulting matplotlib visualizations in order to better convey our findings.

In a Jupyter notebook, you can run each cell by clicking the Run button in the top toolbar.



challenge

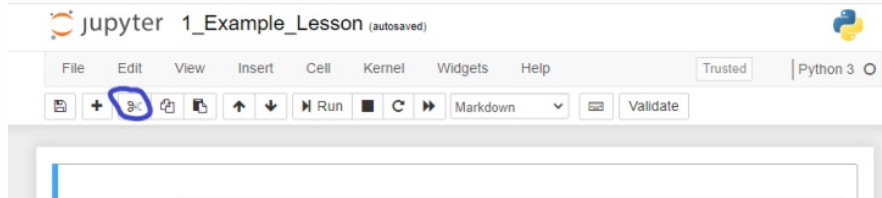
Try it out:

- Click on the last cell (`print(numbers)`), and then press the Run button
- Click on the first code cell (starting with `numbers = [1, 2, 3, 4]`) and then re-run the last code cell

Jupyter notebooks only run a cell, not the entire notebook. This means that you need to ensure you run any code that the current cell depends on **before** attempting to run the current cell.

Markdown Cells

Delete all the current cells before the next task. To do this, click on the scissors in Jupyter's toolbar until you empty the Jupyter Notebook:



Markdown basics

Markdown is a common mark up language, which means it has syntax or symbols that indicate formatting.

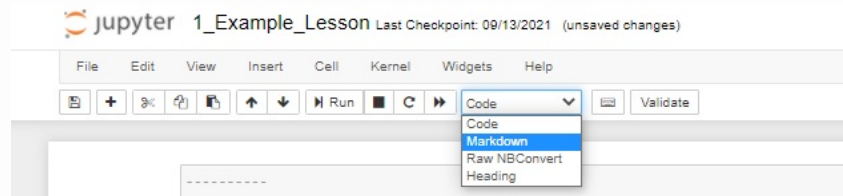
Here is an overview of some basic Markdown Syntax:

Element	Markdown Syntax
Heading	# H1 ## H2 ### H3
Bold	**bold text**
Italic	<i>*italicized text*</i>
Blockquote	> blockquote
Ordered List	1. First item 2. Second item 3. Third item
Unordered List	- First item - Second item - Third item
Code	`code`

You can [see more Markdown syntax using an online cheatsheet](#).

Write some Markdown

1. Set the top cell to **Markdown** using the drop-down in the Jupyter Notebook toolbar:



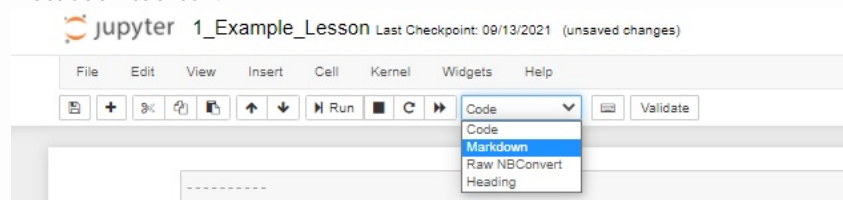
2. Copy the following text:

```
## My first Jupyter Notebook      This notebook:      1. Creates a  
dataset      1. Generates a `matplotlib` plot
```
3. Run your Markdown cell to see the formatted text

Code Cells

Let's declare a couple of data sets to use for visualizations:

1. If a new cell did not appear below your Markdown cell, you can use the + button in the Jupyter Notebook toolbar to create a new cell.
2. Make sure the new cell is set to **Code** in the drop-down of the Jupyter Notebook toolbar:



3. Copy the following python code:

```
x = [10,20,30,40]    y = [5,20,12,3]    print(x)    print(y)
```
4. Run your code cell to ensure the variables have been declared

Your notebook should now look like this:

My first Jupyter Notebook

This notebook:

1. Creates a dataset
2. Generates a `matplotlib` plot

```
In [1]: x = [10,20,30,40]
        y = [5,20,12,3]
        print(x)
        print(y)

[10, 20, 30, 40]
[5, 20, 12, 3]
```

Notice that the output of code cells persist in the Jupyter Notebook.

Visualizations with Matplotlib

Let's create our first visualizations using Matplotlib.

info

Matplotlib

Matplotlib is a plotting library that we can use with Python. We will be using the following command in each Jupyter Notebook:

```
import matplotlib
```

For efficiency, we will tend to give specify which part of the matplotlib library to import. For example:

```
import matplotlib.pyplot as plt
```

The most versatile command in the matplotlib library is the `plot()` function. Try it out by copying the following code into a Code cell in your Jupyter Notebook and run it:

```
import matplotlib.pyplot as plt
plt.plot(y)
plt.show()
```

Please Note: If you remove the `plt.show()` Jupyter would still show our plot. Feel free to try it in the next cell.

In this course, please use the `plt.show()` function in case you work in another environment which does not instantly show you the plot.

In the plot above, we provided a single list `plot()` command. Matplotlib assumes it is a sequence of y values and generated the x values for us. We can input our own x-values as an additional argument. Try running the following code in your cell:

```
import matplotlib.pyplot as plt  
plt.plot(x,y)  
plt.show()
```