1 Lecture 1b: Introduction to Jupyter Notebooks, Google Colab and Matplotlib

Data Visualization . 1-DAV-105

Lecture by Brona Brejova

1.1 Jupyter Notebook

- https://jupyter.org/
- Web-based software for interactive work in Python
- Frequently used for data processing and visualization
- A document called **notebook** consists of **cells**
- Each cell contains either text or Python code
- Text may include formatting in Markdown language
- A cell with Python code can be executed and the results display below, including images
- This presentation is created in Jupyter

1.2 Google Colab

- https://colab.research.google.com/
- Stores your notebook on Google drive, executes them on Google servers
- We will use it in this course
- There are also other notebook hosting services
- Alternatively you can install and run Jupyter Notebook software on your computer (e.g. for sensitive data)

1.3 How to use Notebooks

- Intuitive interface with menus, toolbars, context menus (right-click) etc
- It is useful to learn some keyboard shortcuts

When you are not editting a cell:

- use Up and Down arrows to move between cells
- use **Enter** (or double-click) to start editing a cell
- use **Esc** to stop editing a cell
- use **Ctrl-Enter** to run a code in a cell, **Shift-Enter** to run the code and move to the next cell

1.3.1 Example of a code cell

- A cell can include imports, function definitions, commands
- Variables will be visible in other cells
- The results of print are shown below cell when executed
- The last expression is printed below cell when executed

[ ]:
```
x = range(10)
y = [xval * xval for xval in x]
print(list(x))
```

```
print(y)
y
```

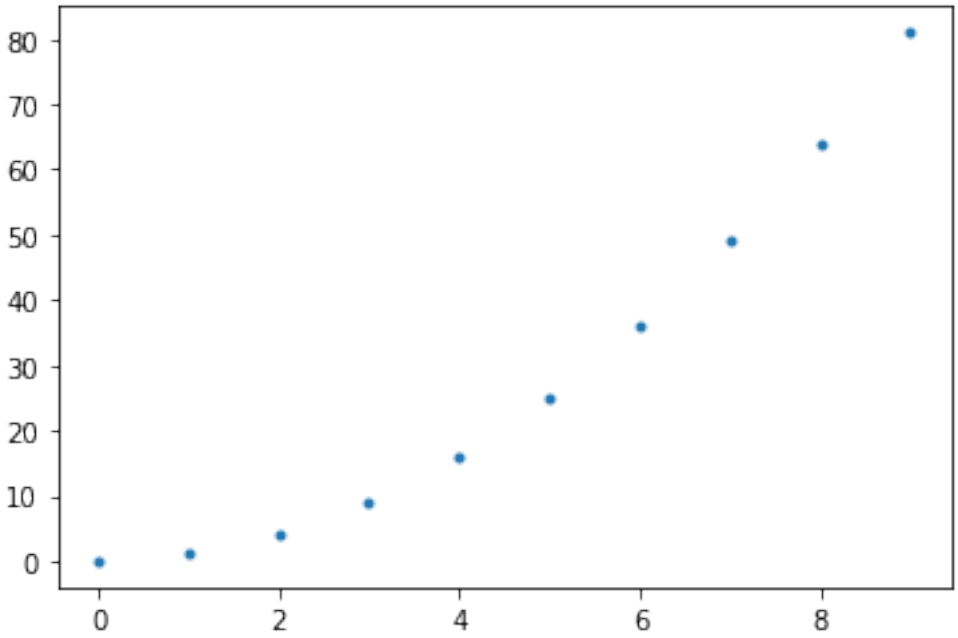
```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

[ ]:
```
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

1.3.2 Example of a cell with a plot

- Uses variables x and y from the previous cell

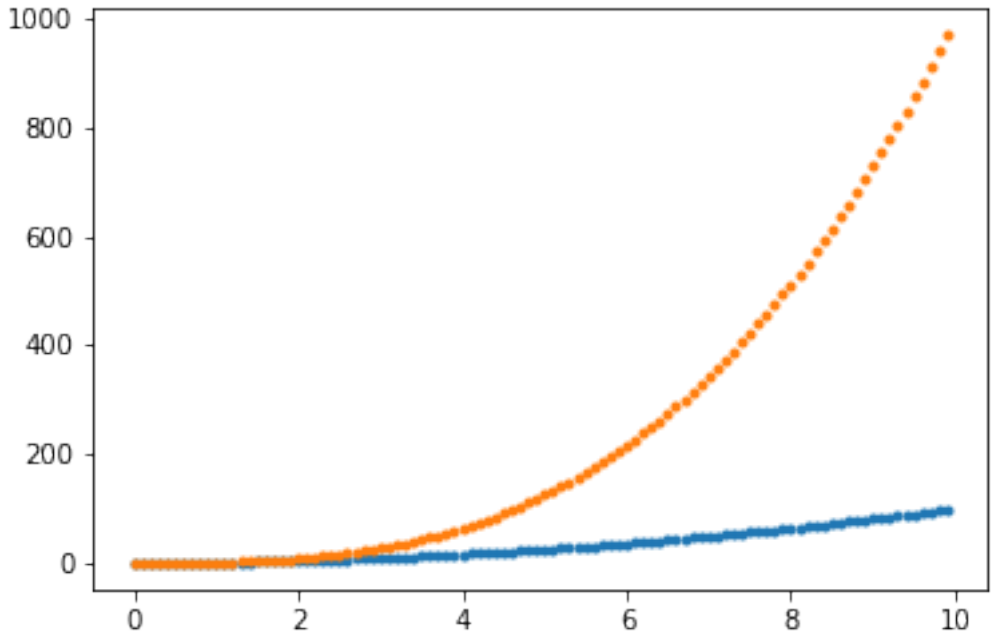
[ ]:
```
import matplotlib.pyplot as plt
# create figure with a single plot (axes)
figure, axes = plt.subplots()
axes.plot(x, y, '.')
figure.show()
```



1.4 Matplotlib library

- https://matplotlib.org/
- A Python library for creating plots
- Axes is Matplotlib name for a single plot
- Let us now plot two functions: quadratic and cubic

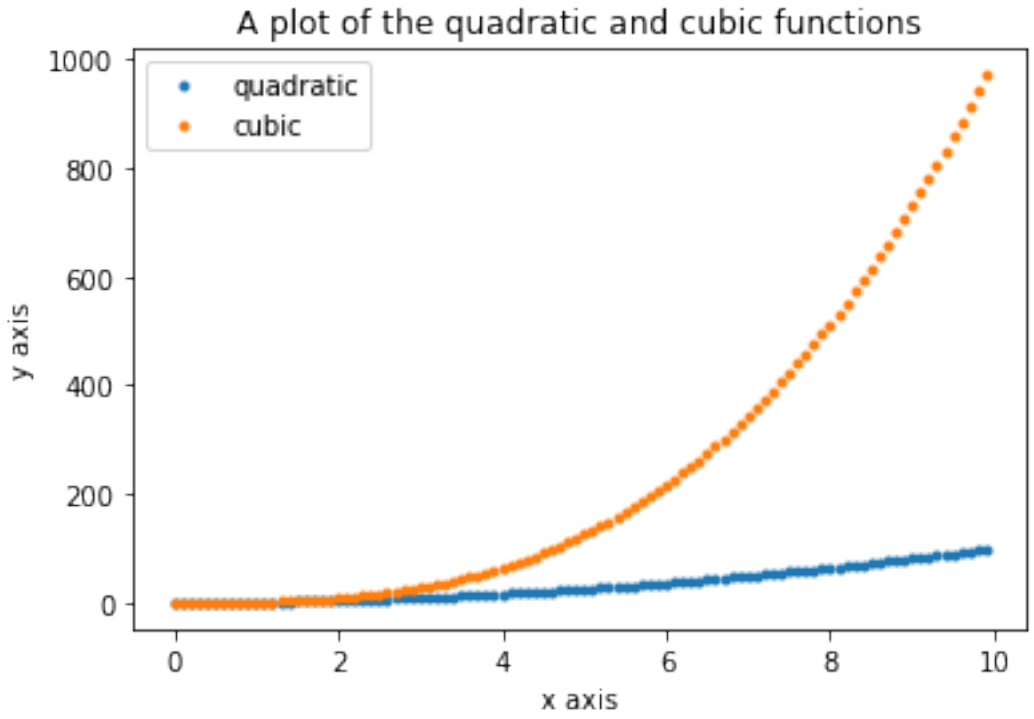
[ ]:
```
# x is values from 0 to 10 with step 0.1
x = [val / 10 for val in range(0, 100)]
# values in y2 are values from x squared
y2 = [xval ** 2 for xval in x]
# values in y3 are values from x to the power of 3
y3 = [xval ** 3 for xval in x]
# plot the quadratic and cubic function in a single plot
figure, axes = plt.subplots()
axes.plot(x, y2, '.')
axes.plot(x, y3, '.')
figure.show()
```



1.4.1 Setting labels and titles in Matplotlib

[ ]:
```
# the same plot as before, but name the two sets of points by label
figure, axes = plt.subplots()
axes.plot(x, y2, '.', label="quadratic")
axes.plot(x, y3, '.', label="cubic")
# add titles for axes, usually use more descriptive titles
axes.set_xlabel("x axis")
axes.set_ylabel("y axis")
# legend (which plot is which function), uses the labels set in axes.plot
axes.legend()
# a title of the whole plot
axes.set_title("A plot of the quadratic and cubic functions")
```

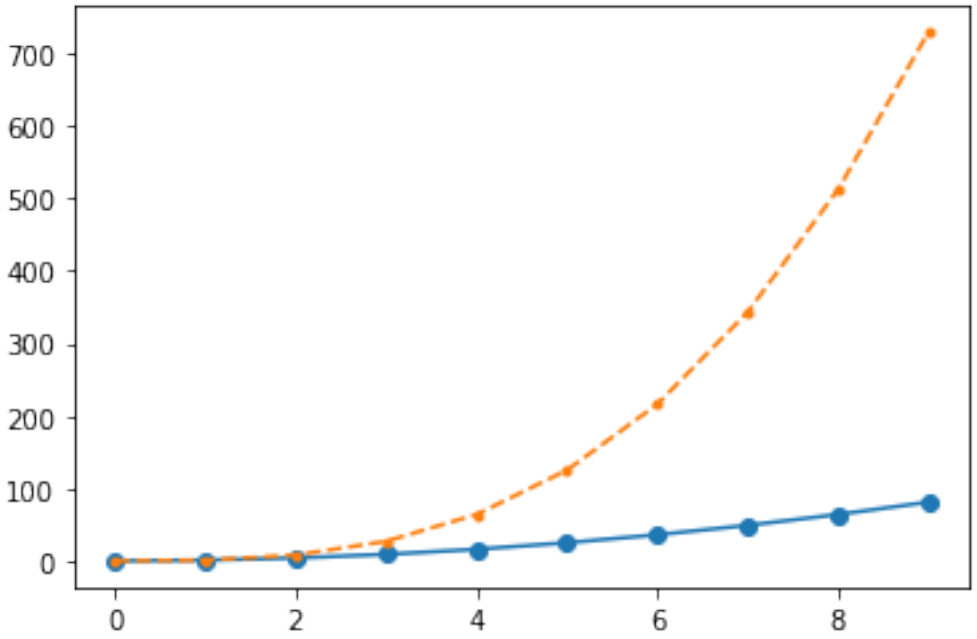
```
figure.show()
```



1.4.2 Setting lines/markers/colors

- In the `axes.plot` command, `'.'` represents formatting, in this case a small dot
- The formatting string has three optional parts: marker, line, color
- Examples: `'or'` red circle, `'-g'` green solid line, `'--'` dashed line
- See more in documentation

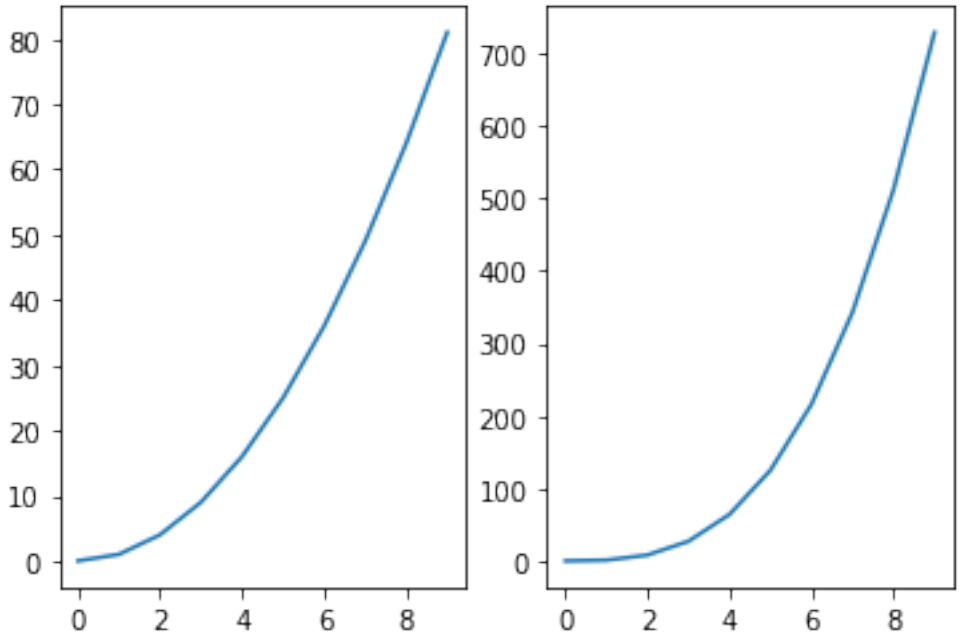
[ ]:
```
x = range(0, 10)
y2 = [xval ** 2 for xval in x]
y3 = [xval ** 3 for xval in x]
figure, axes = plt.subplots()
axes.plot(x, y2, 'o-', x, y3, '.--')
figure.show()
```



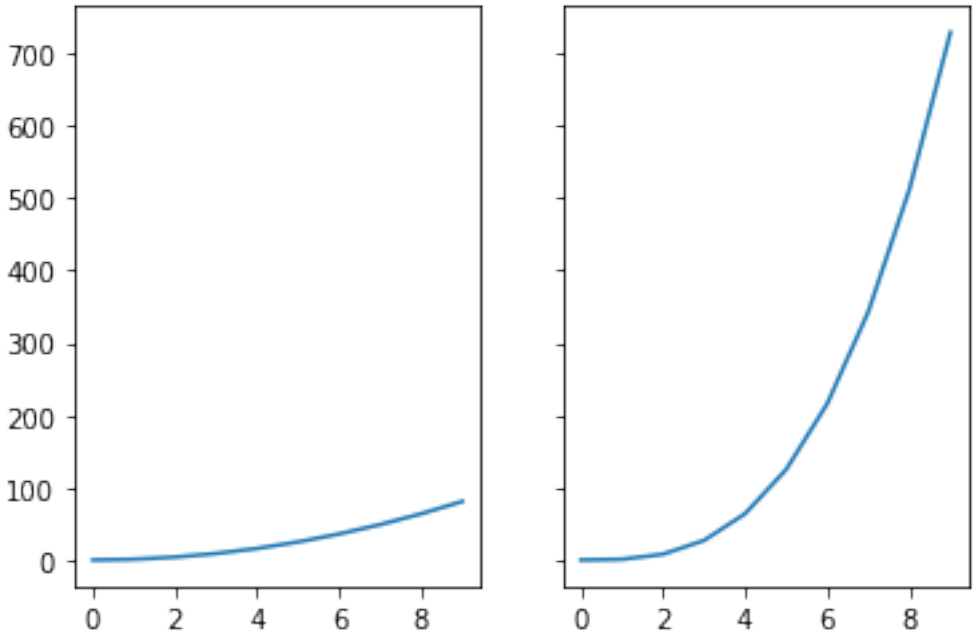
1.4.3 Multiple plots per image

- plt.subplots can take as arguments the number of rows and the number of columns and creates multiple subplots per image
- Note that each plot has different y-axis, which is not ideal because we do not immediately see that the cubic function grows much faster than the quadratic

[ ]:
```
figure, axes = plt.subplots(1, 2)
axes[0].plot(x, y2)
axes[1].plot(x, y3)
figure.show()
```



[ ]:
```
# fixing the problem with different y-axis
figure, axes = plt.subplots(1, 2, sharey=True)
axes[0].plot(x, y2)
axes[1].plot(x, y3)
figure.show()
```



1.5 Additional resources

- Python Data Science Handbook by Jake VanderPlas, O'Reilly 2016 https://jakevdp.github.io/PythonDataScienceHandbook/
- Google Colab website https://colab.research.google.com/ and introductory video https://www.youtube.com/watch?v=inN8seMm7UI
- Matplotlib tutorials https://matplotlib.org/stable/tutorials/index.html