Study Guide: Data Visualization with Python

Afshine Amidi and Shervine Amidi

August 21, 2020

General structure

❏ **Overview** – The general structure of the code that is used to plot figures is as follows:
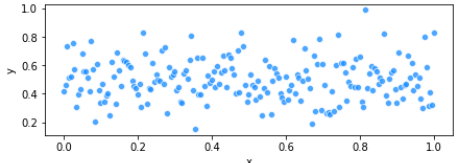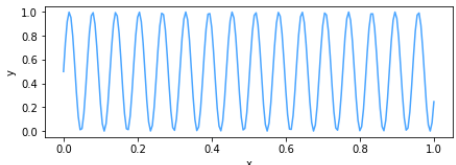
Python
```python
# Plot
f, ax = plt.subplots(...)
ax = sns...

# Legend
plt.title()
plt.xlabel()
plt.ylabel()
```
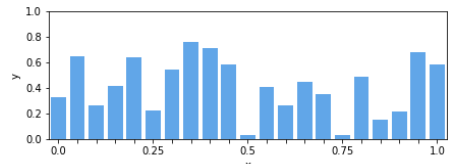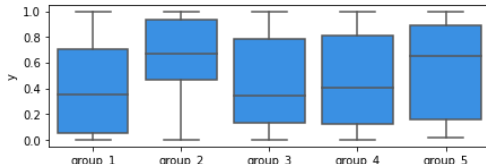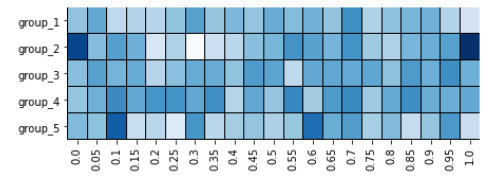
We note that the `plt.subplots()` command enables to specify the figure size.

❏ **Basic plots** – The main basic plots are summarized in the table below:

Type	Command	Illustration

Scatter plot	`sns.scatterplot(`
	`    x, y, params`
	`)`	

Line plot	`sns.lineplot(`
	`    x, y, params`
	`)`	

Bar chart	`sns.barplot(`
	`    x, y, params`
	`)`	
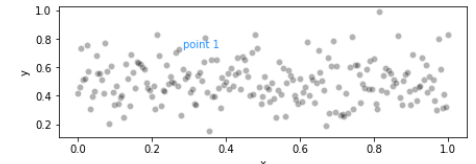
Type	Command	Illustration

Box plot	`sns.boxplot(`
	`    x, y, params`
	`)`	

Heatmap	`sns.heatmap(`
	`    data, params`
	`)`	

where the meaning of parameters are summarized in the table below:
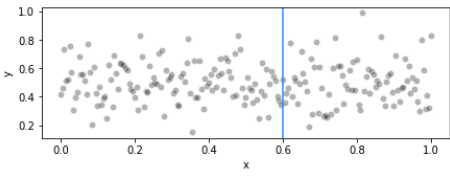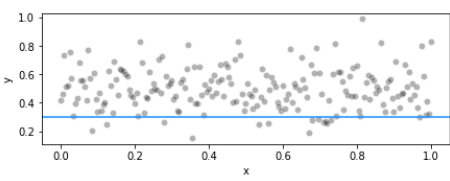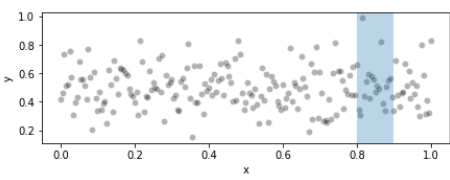
Command	Description	Use case
`hue`	Color of a line / point / border	`'red'`
`fill`	Color of an area	`'red'`
`size`	Size of a line / point	`4`
`linetype`	Shape of a line	`'dashed'`
`alpha`	Transparency, between 0 and 1	`0.3`

Advanced features

❏ **Text annotation** – Plots can have text annotations with the following commands:

Type	Command	Illustration

Text	`ax.text(`
	`    x, y, s, color`
	`)`	

❏ **Additional elements** – We can add objects on the plot with the following commands:

Type	Command	Illustration
Line	```ax.axvline(x, ymin, ymax, color, linewidth, linestyle)```	
	```ax.axhline(y, xmin, xmax, color, linewidth, linestyle)```	
Rectangle	```ax.axvspan(xmin, xmax, ymin, ymax, color, fill, alpha)```	

Last touch

❑ **Legend** – The title of legends can be customized to the plot with the commands summarized below:

Element	Command
Title / subtitle of the plot	```ax.set_title('text', loc, pad)``` ```plt.suptitle('text', x, y, size, ha)```
Title of the x / y axis	```ax.set_xlabel('text') / ax.set_ylabel('text')```
Title of the size / color	```ax.get_legend_handles_labels()```
Caption of the plot	```ax.text('text', x, y, fontsize)```

This results in the following plot:
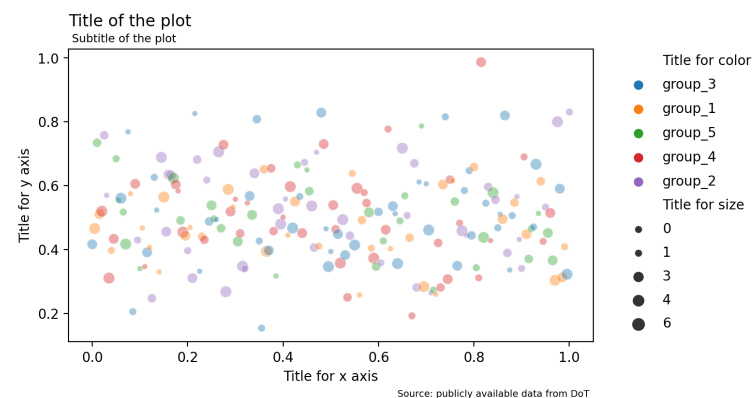
❑ **Double axes** – A plot can have more than one axis with the `plt.twinx()` command. It is done as follows:

Python
```
ax2 = plt.twinx()
```

❑ **Figure saving** – There are two main steps to save a plot:

- Specifying the width and height of the plot when declaring the figure:

  Python
  ```
  f, ax = plt.subplots(1, figsize=(width, height))
  ```

- Saving the figure itself:

  Python
  ```
  f.savefig(fname)
  ```