

Data Visualization with Python

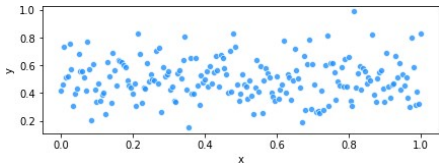
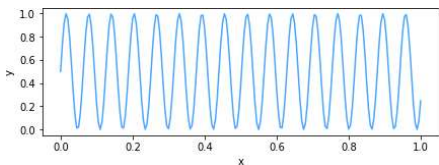
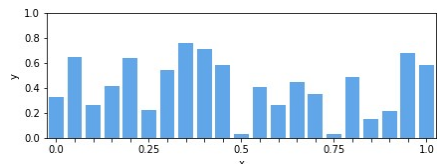
General structure

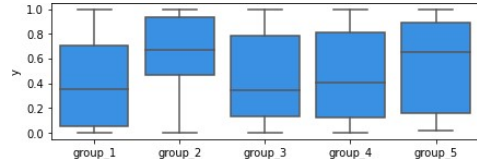
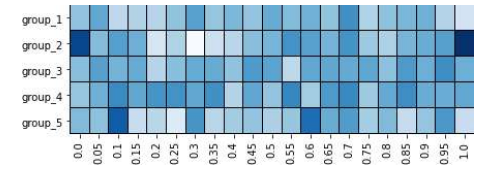
□ **Overview** – The general structure of the code that is used to plot figures is as follows:

Python

```
# Plot
f, ax=plt.subplots(...) ax=sns...
# Legend plt.title()
plt.xlabel() plt.ylabel()
```

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

Type	Command	Illustration
Scatter plot	<code>sns.scatterplot(x, y, params)</code>	
Line plot	<code>sns.lineplot(x, y, params)</code>	
Bar chart	<code>sns.barplot(x, y, params)</code>	

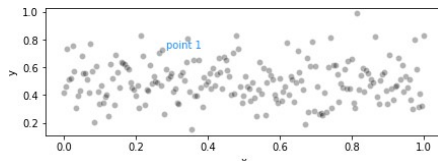
Type	Command	Illustration
Box plot	<code>sns.boxplot(x, y, params)</code>	
Heatmap	<code>sns.heatmap(data, params)</code>	

where the meaning of parameters are summarized in the table below:

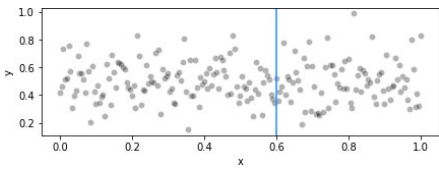
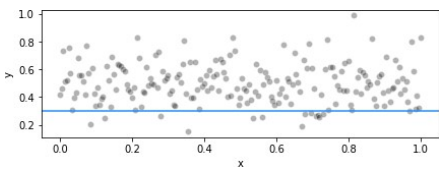
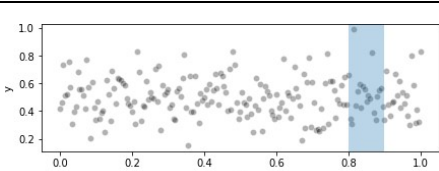
Command	Description	Use case
hue	Color of a line / point / border	'red'
fill	Color of an area	'red'
size	Size of a line / point	4
linetype	Shape of a line	'dashed'
alpha	Transparency, between 0 and 1	0.3

Advanced features

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

Type	Command	Illustration
Text	<code>ax.text(x, y, s, color)</code>	

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

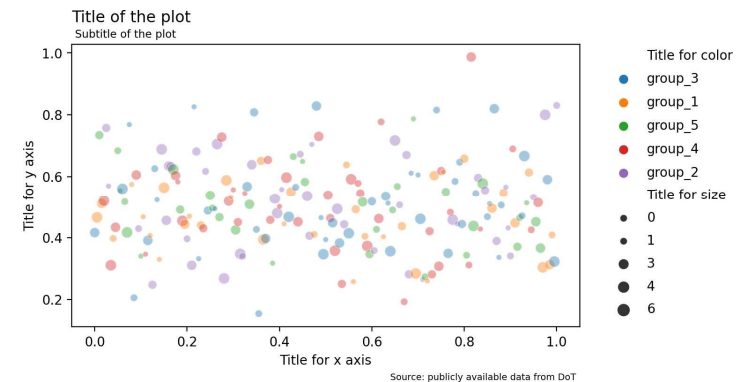
Type	Command	Illustration
Line	<pre>ax.axvline(x, ymin, ymax, color, linewidth, linestyle)</pre>	
	<pre>ax.axhline(y, xmin, xmax, color, linewidth, linestyle)</pre>	
Rectangle	<pre>ax.axvspan(xmin, xmax, ymin, ymax, color, fill, alpha)</pre>	

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	<code>ax.set_title('text', loc, pad)</code>
	<code>plt.suptitle('text', x, y, size, ha)</code>
Title of the <i>x</i> / <i>y</i> axis	<code>ax.set_xlabel('text') / ax.set_ylabel('text')</code>
Title of the size / color	<code>ax.get_legend_handles_labels()</code>
Caption of the plot	<code>ax.text('text', x, y, fontsize)</code>

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)
```