Lesson title

Author name

Introduction

Welcome to this course

Objectives

- □ Create a webpage
- □ Learn how to use metadata
- □ Add a link in the navigation bar
- □ Writing in markdown

0.1 Metadata

This file's metadata is displayed at the top between the "—".

0.1.1 Basic setup

```
title: "Title of this page"
description: "Description of this page"
author: "Author name"
format: html                                # because it's a webpage
from: markdown+emoji                        # you use markdown language
```

0.1.2 Advanced

0.1.2.1 Add a table of contents

With the following setup, all titles, i.e. lines beginning with `#` will be added to the table of contents.

```
format:
  html:
    toc: true                               # Add a Table Of Contents
    toc-location: left
    toc-expand: 3                           # 1 : only main titles ... 5 : highly detailed
```

If you want to remove a title from the table of contents, use the `unlisted` class :

```
# My title {.unlisted}
```

0.1.3 Self-numbering

If you use this setup `number-sections: true`, each title will be numbered.
If you don't want to number a part, use class `unnumbered` :

```
# Introduction {.unnumbered}
```

0.1.4 Export pdf

The format of a web page is html.
But you can also export in pdf format.

```
format:
  html:
    ...
  pdf: default
```

When you will render the quarto project, it will create a file `lesson.pdf` in addition of `lesson.html`.
On the webpage, a link will appear under the table of contents to download the pdf.

> Note
>
> To perform this you have to install first tinytex : `quarto install tinytex`

0.2 Add a link in the navigation bar

The navigation bar is setup in the file `_quarto.yml`. Look for `navbar` key, and then add a link to this page

0.3 Markdown tips

0.3.1 Unordered List

```
- first
- second
- third
- [ ] checkbox
```

0.4 Quarto tips

0.4.1 Tabulations

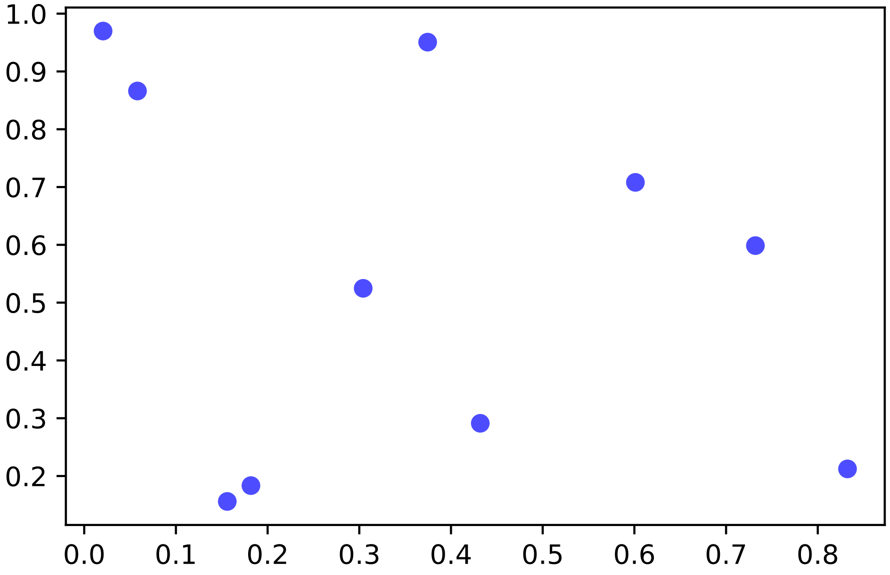
0.4.1.1 Code

Listing 1 `file.py`

```python
import numpy as np
import matplotlib.pyplot as plt

np.random.seed(42)
pairs = np.random.rand(10, 2)

plt.scatter(pairs[:, 0], pairs[:, 1], color='blue', alpha=0.7)
plt.show()
```

0.4.1.2 Plot

```
import numpy as np
import matplotlib.pyplot as plt

np.random.seed(42)
pairs = np.random.rand(10, 2)

plt.scatter(pairs[:, 0], pairs[:, 1], color='blue', alpha=0.7)
plt.show()
```



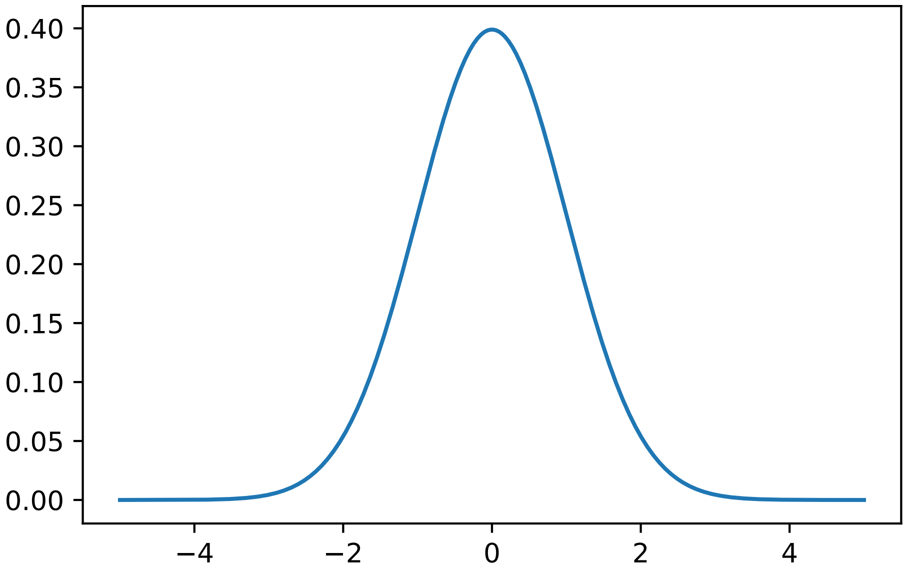
0.4.1.3 Data

```
import numpy as np
import matplotlib.pyplot as plt

np.random.seed(42)
pairs = np.random.rand(10, 2)
print(pairs)
```

```
[[0.37454012 0.95071431]
 [0.73199394 0.59865848]
 [0.15601864 0.15599452]
 [0.05808361 0.86617615]
 [0.60111501 0.70807258]
 [0.02058449 0.96990985]
 [0.83244264 0.21233911]
 [0.18182497 0.18340451]
 [0.30424224 0.52475643]
 [0.43194502 0.29122914]]
```

0.4.2 Code

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

x = np.linspace(-5, 5, 1000)

pdf = norm.pdf(x, 0, 1)
plt.plot(x, pdf, label='Normal Distribution')
plt.show()
```



0.4.3 Callout Blocks

ℹ Note

Note that there are five types of callouts, including: `note`, `tip`, `warning`, `caution`, and `important`.

💡 Show

Hidden text