600\_02\_Quarto

Tomaz Kastrun

Table of contents

# 1. Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see <https://quarto.org>.

## 1.1 Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

# highlight-style: github  
1 + 1

[1] 2

You can add options to executable code like this

[1] 4

The echo: false option disables the printing of code (only output is displayed).

### 1.1.1 Running Code in R

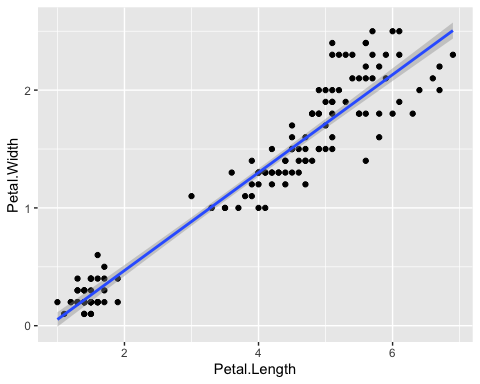
Here with using R Language.

iris <- iris  
library(ggplot2)  
library(tidyverse)

── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
✔ dplyr 1.1.4 ✔ readr 2.1.4  
✔ forcats 1.0.0 ✔ stringr 1.5.0  
✔ lubridate 1.9.2 ✔ tibble 3.2.1  
✔ purrr 1.0.2 ✔ tidyr 1.3.1  
── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
✖ dplyr::filter() masks stats::filter()  
✖ dplyr::lag() masks stats::lag()  
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

iris %>%  
 select ('Petal.Width', 'Petal.Length') %>%  
 ggplot(aes(x=Petal.Length,y=Petal.Width))+  
 geom\_point()+  
 geom\_smooth(method="lm")

`geom\_smooth()` using formula = 'y ~ x'



### 1.1.2 Running Code in Py

And here is to use with Python:

a = 1

And overall it is irrelevant the origin of language.

This is overall job!