

Record of Installation Julia Pluto notebook

In any Julia REPL > 1.5 as detailed in ID's:

35 Raspberry Pi 4 Raspberry Pi OS 32 bit

36 Raspberry Pi 400 Raspberry Pi OS 32 bit

38 Raspberry Pi 400 Manjaro OS

39 M1 Macbook Pro macOS Big Sur x86 emulation (should be same on Intel Macs)

40 Raspberry Pi 4 Raspberry Pi OS 64 bit

41 M1 Macbook Pro macOS Big Sur ARM native (Julia 1.8-DEV at time of writing)

42 M1 Macbook Pro Parallels Ubuntu 20.04 (should be same on other Linux)

Start environment

julia (see Julia REPL and check version > 1.5)

] enters package manager

add Pluto

status

Delete exits package manager

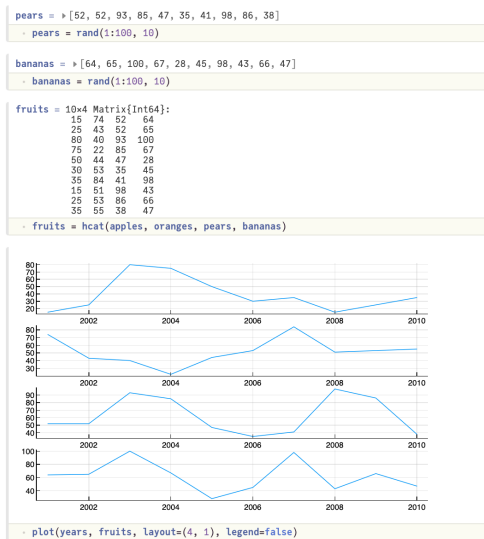
In Julia REPL:

Pluto.run(host="0.0.0.0")

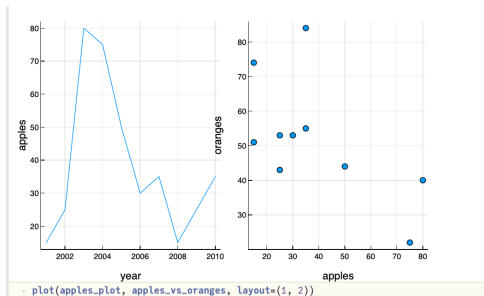
Note the secret the instance runs under

In browser of any device (iPad, your phone?):

Visit YOURIPADDRESS:1234/?secret=THESECRETNOTED



This works when we have a nice output matrix. For more complicated options, named plots are usually a good solution. When we have created a few plots, we can put them together in a single plot.



The Pluto way of plotting

For the most part, plotting in Pluto is not different from anywhere else. However, there are a few things to keep in mind.

The `plot!()` function alters an existing plot. In reactive programming, you are not supposed to alter the value of a variable you defined in a different cell. I strongly recommend that you only use `plot!()` to alter plots you initialised in the same cell.

For the sake of demonstration, here is what happens if you use `plot!()` in its own cell. Let's start by making a new plot.

```
new_plot =
```