Modules

Thomas Schwarz

Modules

- Modules are the way for Python to extend functionality
  - Modules contain definitions of constants, functions, …
  - To use a module, a module needs to be imported
- Modules are:
  - Provided to us as part of python: math, random
  - Provided to us as maintained packages that we need to install: numpy (numerical python), matplotlib (visualization) …
  - Provided by us in order to structure our code and to reuse parts of it in other projects

How to import a module

- If we want to use a module, we need to tell Python
  - Use the import statement
    - `import my_module`
      - Any component `fun` is now available as `my_module.fun`
    - `from my_module import fun`
      - Only component `fun` is now available as `fun`

How to import a module

- How to import a module
  - `from my_module import *`
  - Everything in `my_module` is now available without prefix
    - This is dangerous because you can now have naming conflicts (two different things are named the same, and the latter named wins)
- Use an abbreviation:
  - `import my_module as mm`
  - Everything in `my_module` is now available as `mm.fun`

How to install a module

- Open up a command window / terminal
- Use pip, the python installer
    - To install matplotlib, say
        - pip3.10 install matplotlib
        - wait a few seconds.
            - Using the version number of Python allows you to install for that version
            - pip will install for the latest installed versions
- Python now has virtual environments to manage different and possibly conflicting sets of installed modules

Built-in Modules

- The math module
  - Contains constants: `math.e, math.pi`
  - Contains functions:
    - Trigonometry: `math.sin, math.cos, math.tan, …`
    - Exponentials: `math.exp, math.log, math.sqrt`

Built-in Modules

- Random module:
  - Useful for simulations
- Example: When we throw two dice, how often is the same face
  - Use randint(1,6) to simulate the throwing of a die
    - First argument is first possible value
    - Last argument is last possible value
- Count how often this happens

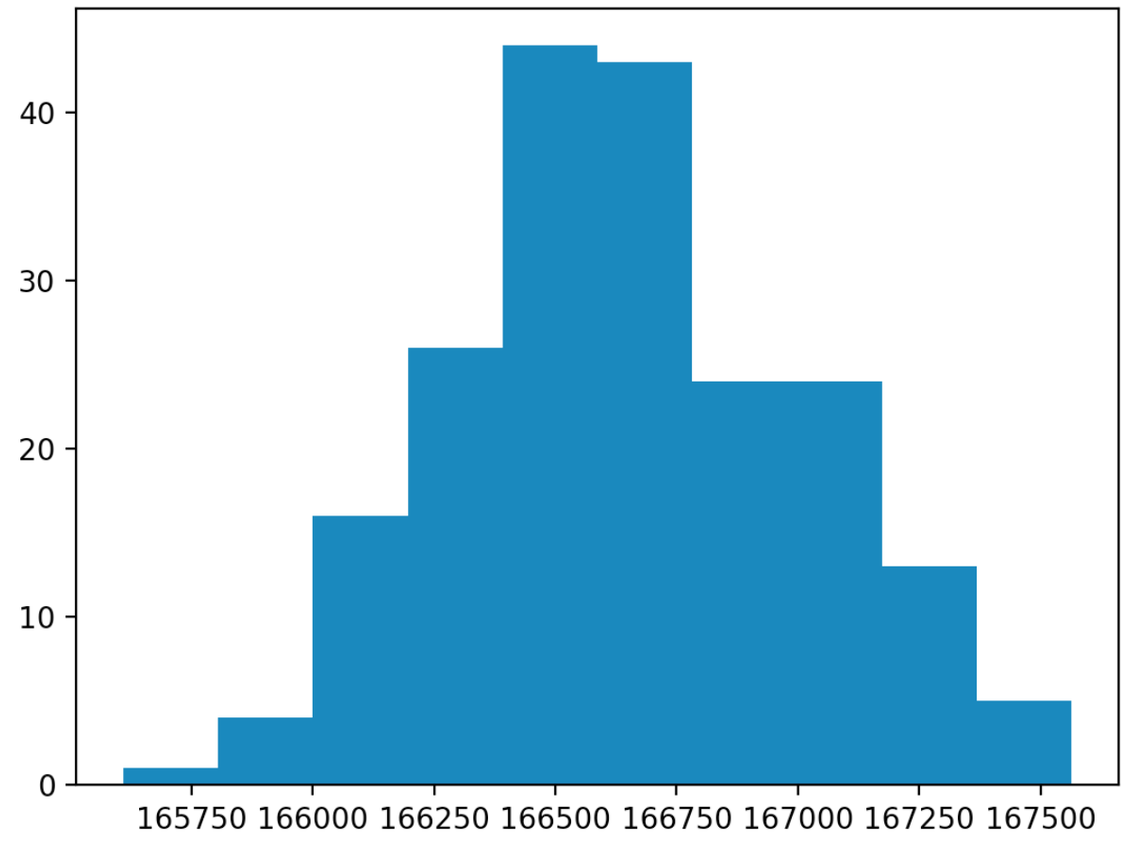
Built in Modules

```
import random as rd

def try_out(trials):
    count = 0
    for _ in range(trials):
        x = rd.randint(1,6)
        y = rd.randint(1,6)
        if x==y:
            count += 1
return count
```

Built-in Modules

- Repeat 200 times for 1,000,000 trials and create a histogram



Creating Modules

- Write a Python file
  - containing constants and functions
- You just created a module that you can import