



Performance Analysis

Instructor: Christopher Werner



Measuring performance

- Correction before optimisation
- Before optimising, understand where time is being spent
 - 90% of time can be spent on 10% of code
- Simple tools to use:
 - `time`, `timeit`
 - `cProfile`
- Profiling tools: TAU, Intel Vtune, etc.



- Python's time module can be used to find the time spent on specific lines of the program

```
import time
import numpy as np

t0 = time.time()

arr = np.arange(100000)
dif = np.zeros(99999, int)
for i in range(1, len(arr)):
    dif[i-1] = arr[i] - arr[i-1]

t1 = time.time()

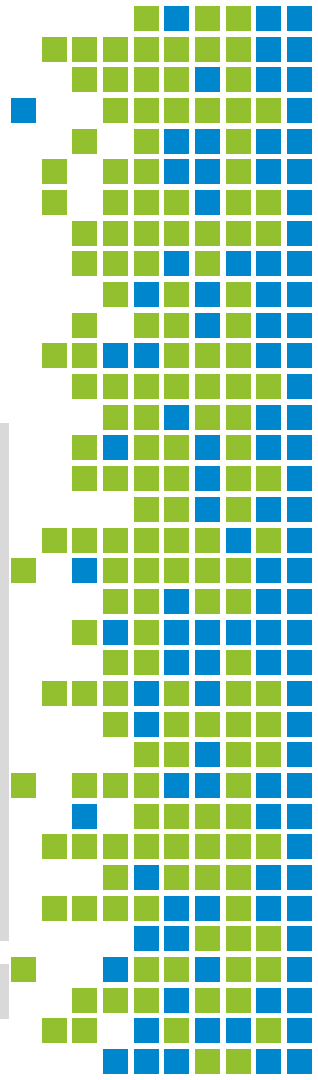
print('Time taken in for loop: ', t1-t0)
```

```
t0 = time.time()

arr = np.arange(100000)
dif = arr[1:] - arr[:-1]

t1 = time.time()

print('Time taken for
vectorised option: ', t1-t0)
```



- Similar to time, with easy timing of small bits of Python code
- Avoids execution time omission as seen in time module

```
### Lots of imports  
  
def func1():  
    . . .  
def func_time(x, y):  
    # do something
```

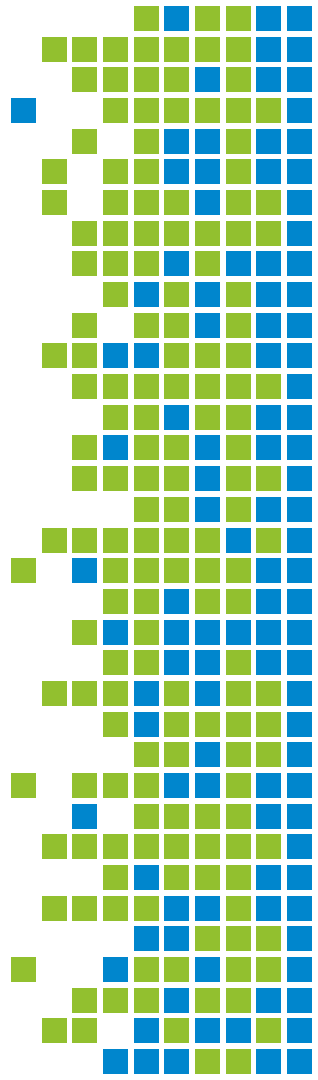
IPython:

```
%timeit func_time(x, y)
```

```
#time_func.py  
  
import timeit  
  
s = """  
# imports needed for func_time  
  
def func_time(x, y):  
    # do something  
    """  
  
t=timeit.Timer(stmt="func_time(x, y)",  
               setup=setup)  
print(t,timeit(5))
```

Terminal:

```
python -m time_func.py
```



- cProfile provides an API for profiling a python program
 - A profile is a set of statistics
 - Returns the time spent in different parts of the program

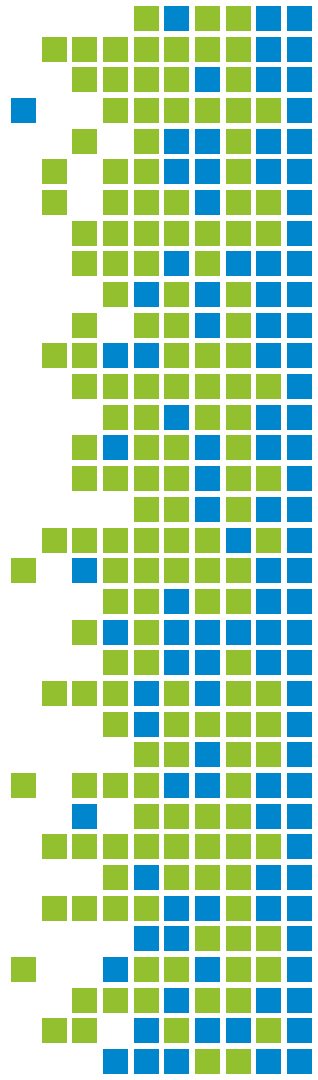
- IPython:

```
import cProfile
cProfile.run('func(arg1, arg2)', 'func.prof')
```

- Terminal:

```
python -m cProfile -o myprofile.prof myprog.py
```

Investigating cProfile with pstats



- Prints the execution time of selected functions
- Can sort by function name, time, cumulative time etc
- Python module interface and interactive browser

```
# mystats.py
from pstats import Stats
P = Stats('myprofile.prof')
p.strip_dirs()
p.sort_stats('time').print_stats(10)
```

```
$ python -m pstats myprofile.prof
Welcome to the profile statistics
% strip
% sort time
% stats 5
```