Code Quality

The practices and the tools

What is Code Quality?

- It can mean different things to different people.
- To business, it may mean just time to market and accuracy in the process
- To a data scientist, quality of code = quality of models deployed.
- To a developer, it is
  - modularity (manual)
  - correctness (tests)
  - performance (profiling)
  - readability (stylizing/documentation)

Modularity

- In a bid to reduce complexity and to avoid ambiguity, each function must do only one task.
- Separate concerns as much as possible.

```
def add(a,b):
    return a+b

def sub(a,b):
    return a-b
```

Question of debate - 1 or many?

- There is really no 1 tool to rule them all.
- Or is there? ;)

Correctness

- What is McCabe complexity?
- Also known as cyclomatic complexity, it is the number of paths the code can take. The number goes higher than 1 if the number of control flow statements are higher than 0.
- To see what this number is -

```
pip install mccabe
python -m mccabe --min 1 pyq_mccabe.py
```

How to test?

- There are different phases of testing - unit, integration etc.
- The most popular tool is pytest.
- Coverage needs to be ensured.

Performance

- Why profile?
  - Not all operations are equally expensive.
  - In long running scripts that are not very easily discernable with the eye, it is useful to use a line profiler to tell you which operation is the costliest in terms of time so you can try to optimize it better.

- What can you use to profile time?

```
pip install line-profiler
kernprof -l pyq1.py
python -m line_profiler pyq1.py.lprof
```

Memory?

- That is important is some sectors and can be considered a measure of quality.
- How do you handle different data types?
- It can be measured using a simple decorator as well. Illustrated in pyq_mem_profile.py

Readability

- While everyone's tolerance and abilities differ, it's good to have a standard set of practices.
- In the absence of such a standard on a company-wide level, PEP-8 serves as a good reference.
- There are companies that have their own style guides - Google for e.g.
- To facilitate this, we have pylint, black

py-linting

- [R]efactor for “good practice” metric violation
- [C]onvention for coding standard violation
- [W]arning for stylistic problems, or minor programming issues
- [E]rror for important programming issues (i.e. most probably a bug)
- [F]atal for errors which prevented further processing

Expectation vs Reality

-

```
------------------------------------
Your code has been rated at 10.00/10
```

- Reality?

For Django Projects

- pylint-django

```
pip install pylint-django
pylint --load-plugins pylint_django .
```

`pre-commit`

Installation and usage

- This is a git-hook. It means, you use it within a git repository and "hook" into git's config folders.
- `pip install pre-commit`
- `pre-commit install`
- Set up a .pre-commit-config.yaml file with a config perhaps like -

```
repos:
-   repo: https://github.com/pre-commit/pre-commit-hooks
    rev: v2.3.0
    hooks:
    -   id: check-yaml
    -   id: end-of-file-fixer
    -   id: trailing-whitespace
-   repo: https://github.com/psf/black
    rev: 19.3b0
    hooks:
    -   id: black
```

- `pre-commit run --all-files`
- This performs all the operations in the Config file for you.
- What else can you do with it?

Coming back to our question of debate - 1 or many?

- Is there really 1 tool to rule them all?
- Not really.
- Prospector comes close.
- But you will have to run pytest and coverage anyway.

Other libraries to check

- Bandit - to find common security issues
- MyPy, Pyre – Static type checker - https://pyre-check.org/
- Sphinx – Documentation creator
- Isort – sorts imports
- Coverage
- Wemake
- Deepsource (:))
- Sonarqube? Codeclimate?

Resources

- https://github.com/pre-commit/pre-commit
- https://github.com/PyCQA
- https://github.com/pyutils/line_profiler
- https://pythonspeed.com/articles/pylint/
- https://github.com/wemake-services/wemake-python-styleguide
- Talk by James Powell
- Code on Github

Thank you

- Abhiram R
- https://abhiramr.com
- https://github.com/abhiramr
- @abhicantdraw on Twitter