Testing

# **Testing And Data Science**

* **TEST DRIVEN DEVELOPMENT:** a development process where you write tests for tasks before you even write the code to implement those tasks.
* **UNIT TEST:** a type of test that covers a “unit” of code, usually a single function, independently from the rest of the program.

### **Resources:**

* Four Ways Data Science Goes Wrong and How Test Driven Data Analysis Can Help: [Blog Post](https://www.predictiveanalyticsworld.com/patimes/four-ways-data-science-goes-wrong-and-how-test-driven-data-analysis-can-help/6947/)
* Ned Batchelder: Getting Started Testing: [Slide Deck](https://speakerdeck.com/pycon2014/getting-started-testing-by-ned-batchelder) and [Presentation Video](https://www.youtube.com/watch?v=FxSsnHeWQBY)

#### **Unit Test Advantages and Disadvantages**

The advantage of unit tests is that they are isolated from the rest of your program, and thus, no dependencies are involved. They don't require access to databases, APIs, or other external sources of information. However, passing unit tests isn’t always enough to prove that our program is working successfully. To show that all the parts of our program work with each other properly, communicating and transferring data between them correctly, we use integration tests. In this lesson, we'll focus on unit tests; however, when you start building larger programs, you will want to use integration tests as well.

You can read about integration testing and how integration tests relate to unit tests [here](https://www.fullstackpython.com/integration-testing.html). That article contains other very useful links as well.

# **Unit Testing Tools**

To install pytest, run pip install -U pytest in your terminal. You can see more information on getting started [here](https://docs.pytest.org/en/latest/getting-started.html).

* Create a test file starting with test\_
* Define unit test functions that start with test\_ inside the test file
* Enter pytest into your terminal in the directory of your test file and it will detect these tests for you!

test\_ is the default - if you wish to change this, you can learn how to in this [pytest configuration](https://docs.pytest.org/en/latest/customize.html)

In the test output, periods represent successful unit tests and F's represent failed unit tests. Since all you see is what test functions failed, it's wise to have only one assert statement per test. Otherwise, you wouldn't know exactly how many tests failed, and which tests failed.

Your tests won't be stopped by failed assert statements, but it will stop if you have syntax errors.

Test driven development for data science is relatively new and has a lot of experimentation and breakthroughs appearing, which you can learn more about in the resources below.

* [Data Science TDD](https://www.linkedin.com/pulse/data-science-test-driven-development-sam-savage/)
  + Test for model acc in different level (A,B,C,etc...)
* [TDD for Data Science](http://engineering.pivotal.io/post/test-driven-development-for-data-science/)
* [TDD is Essential for Good Data Science Here's Why](https://medium.com/@karijdempsey/test-driven-development-is-essential-for-good-data-science-heres-why-db7975a03a44)
* [Testing Your Code](http://docs.python-guide.org/en/latest/writing/tests/) (general python TDD)