

# Pytesting

## Running tests with pytest

Luke Herbert

January 21, 2025

## Presentation Overview

- 1 Setting Up pytest
- 2 Running Tests
- 3 Writing Tests
- 4 Continuous Integration
- 5 Debugging Tests
- 6 Coverage Analysis

# Contents

- 1 Setting Up pytest
- 2 Running Tests
- 3 Writing Tests
- 4 Continuous Integration
- 5 Debugging Tests
- 6 Coverage Analysis

## Initial Setup

### Install Required Packages

```
1 pip install pytest==8.0.0
2 pip install pytest-cov==4.1.0
3 pip install hypothesis==6.98.0
4 pip install pytest-snapshot==0.9.0
5
```

### Project Configuration

Create `pytest.ini` at project root:

```
1 [pytest]
2 testpaths = tests
3 python_files = test_*.py
4 addopts = -v --cov=streamlit_app
5
```

# Test Configuration

## confest.py Setup

```
1 import pytest
2 from streamlit_app import SmokeModel_MECC_Model
3
4 @pytest.fixture
5 def base_model():
6     params = {
7         'N_people': 50,
8         'N_service': 1,
9         'visit_prob': 0.1,
10        'seed': 42
11    }
12    return SmokeModel_MECC_Model(**params)
13
```

# Contents

- 1 Setting Up pytest
- 2 Running Tests**
- 3 Writing Tests
- 4 Continuous Integration
- 5 Debugging Tests
- 6 Coverage Analysis

# Basic Test Commands

## Common Usage

```
1 # Run all tests with coverage
2 pytest --cov=streamlit_app
3
4 # Run specific test file
5 pytest tests/unit/test_parameters.py -v
6
7 # Run tests matching pattern
8 pytest -k "test_parameter" -v
9
10 # Generate HTML coverage report
11 pytest --cov=streamlit_app --cov-report=html
12
```

# Test Organization

```
1 tests/
2 |-- conftest.py           # Shared fixtures
3 |-- unit/
4 |   |-- test_parameters.py
5 |   |-- test_smoke_model_initialization.py
6 |   |-- test_smoke_model_quit_mechanics.py
7 |   `-- test_smoke_model_relapse.py
8 `-- integration/
9     |-- test_model_integration.py
10    `-- test_service_scaling.py
```



# Contents

- 1 Setting Up pytest
- 2 Running Tests
- 3 Writing Tests**
- 4 Continuous Integration
- 5 Debugging Tests
- 6 Coverage Analysis

## Test Structure Example

```
1 def test_smoke_model_initialization(base_model):
2     """Test model initialization with default params"""
3     # Check initial state
4     assert base_model.N_people == 50
5     assert base_model.visit_prob == 0.1
6
7     # Verify agent creation
8     agents = [a for a in base_model.schedule.agents]
9     assert len(agents) == 51 # 50 people + 1 service
10
11    # Check random seed
12    assert base_model.random.getstate() is not None
13
```

## Testing Model Behavior

```
1 def test_service_interaction(base_model):  
2     """Test interaction between people and services"""  
3     # Force visit probability to ensure contact  
4     base_model.visit_prob = 1.0  
5  
6     # Run one step  
7     base_model.step()  
8  
9     # Check data collection  
10    data =  
11    base_model.datacollector.get_model_vars_dataframe()  
12    assert "Total Contacts" in data.columns  
13    assert data["Total Contacts"].iloc[-1] > 0
```

# Contents

- 1 Setting Up pytest
- 2 Running Tests
- 3 Writing Tests
- 4 Continuous Integration**
- 5 Debugging Tests
- 6 Coverage Analysis

# Continuous Testing

## Using pytest-watch

```
1 # Install pytest-watch
2 pip install pytest-watch
3
4 # Run with continuous monitoring
5 ptw --onpass "echo 'Success!'" --onfail "echo 'Failed!'"
6
7 # Watch specific tests
8 ptw tests/unit/ -- -v
9
```

# Pre-commit Hooks

## Pre-commit Configuration

Create `.pre-commit-config.yaml`:

```
1 repos:
2 - repo: local
3   hooks:
4     - id: pytest
5       name: pytest
6       entry: pytest
7       language: system
8       types: [python]
9       pass_filenames: false
10
```

# Contents

- 1 Setting Up pytest
- 2 Running Tests
- 3 Writing Tests
- 4 Continuous Integration
- 5 Debugging Tests**
- 6 Coverage Analysis

# Debugging with PDB

## Using Python Debugger

```
1 def test_complex_behavior(base_model):  
2     # Add breakpoint for debugging  
3     import pdb; pdb.set_trace()  
4  
5     # Or use pytest.set_trace()  
6     import pytest; pytest.set_trace()  
7
```



# Debug Command Options

## Running Tests with Debug Mode

```
1 # Print output during test run
2 pytest -s
3
4 # Show local variables on failure
5 pytest -l
6
7 # Enter PDB on first failure
8 pytest --pdb
9
```

# Contents

- 1 Setting Up pytest
- 2 Running Tests
- 3 Writing Tests
- 4 Continuous Integration
- 5 Debugging Tests
- 6 Coverage Analysis**

## Coverage Reports

### Generating Coverage Data

```
1 # Basic coverage
2 pytest --cov=streamlit_app
3
4 # Detailed terminal report
5 pytest --cov=streamlit_app --cov-report=term-missing
6
7 # HTML report
8 pytest --cov=streamlit_app --cov-report=html
9
10 # XML for CI systems
11 pytest --cov=streamlit_app --cov-report=xml
12
```

# Coverage Settings

## Coverage Configuration

In `.coveragerc`:

```
1 [run]
2 source = streamlit_app
3 omit = tests/*
4         setup.py
5
```