

# **Effective Programming Practices for Economists**

## **Software engineering**

**What to test? How to test it?**

Janoś Gabler and Hans-Martin von Gaudecker

# Test for passing invalid data

```
def test_clean_agreement_scale_invalid_data():  
    with pytest.raises(ValueError):  
        _clean_agreement_scale(pd.Series([-77, "typo"]))
```

- Passing two codes that should not work
- We expect a `ValueError` to be raised
- Test will fail if
  - no error is being raised
  - a different error is being raised

# Always perform the countercheck

```
def test_clean_agreement_scale_invalid_data():  
    with pytest.raises(ValueError):  
        _clean_agreement_scale(pd.Series(["-77", "typo"]))
```

- "-77" is perfectly valid data.
- Still, `ValueError` is raised as soon as one element in the series is invalid. Test passes.
- Tests may pass for other reasons than what you have in mind!

# What to test?

- Only interfaces!
  - Typical input
  - Corner cases
  - "All" exceptions
  - Any bugs that you have encountered. Workflow:
    1. Pin down by finding minimal testcase
    2. Make it part of the test suite
    3. Fix the bug
- Any bug that came up once is likely to come back!

# How to test?

- Granular tests
  - one assert statement per function
  - careful with anything that is not a scalar (make sure test uses "and" conditions, not "or")
- Always perform the countercheck!