Effective Programming Practices for Economists

Software engineering

Reusing test functions

Janoś Gabler and Hans-Martin von Gaudecker

Careful with "or"-style conditions

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

- Could solve by a careful check of message coming with ValueError
- Much clearer: Run once for each element of `["-77", "typo"]`

Reusing test functions

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

- first argument is a string with the test function input
- second argument is an iterable
- test function will be run once for each element of the iterable
- could have more than one argument to test function
  - including expected output
  - see documentation for syntax

One test function, two tests

Countercheck fails as it is supposed to