+

Pandas PyTesting

*with **Python** 3.6*

Group 23

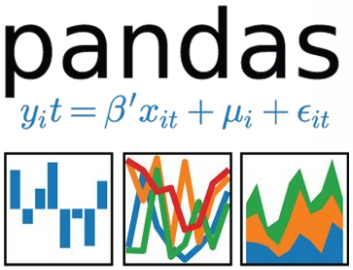
Shu-Pei Huang
Adam Ross

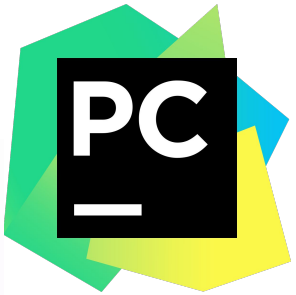
Li Ju
Vladislav Shapar

Yin-Hsiang Liao
Liang-Yan Yu

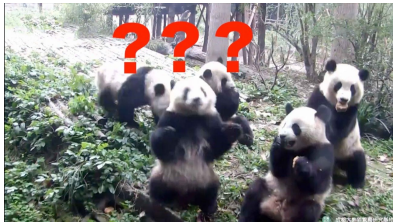
Summary of Tools

- Technical:
  - **Python** v3.6: *theoretical programming language*
  - **Pandas** v0.25.3: *Python library - get to that next*
  - **PyTest** v5.3.1: *full-featured Python testing program*
  - **Coverage.py** v4.5.4: *measures code coverage of Python programs*
  - **Numpy** v1.17.4*: scientific computing package*
  - **PyCharm** v2019.2.5: *IDE for Python*
  - **Ubuntu** v18.04 : *Linux command terminal*
- Teamwork:
  - **Slack**: *cloud-based communication*
  - **Overleaf**: *online LaTex report writing*
  - **GitHub**: *git version control*
  - **Google slides**: *presentation design*





Why **Pandas** *v0.25.3*



- Pandas is an open source library providing all sufficient tools for data mining and preparation as well as for data analysis and modeling.

- It has a huge collection of different classes and functions, which describe either mathematical or algorithmic ideas.

- Many people use this library and rely on it

pandas

$$y_i t = \beta' x_{it} + \mu_i + \epsilon_{it}$$

Testing tools: **PyTest** *v5.3.1*

- Minimal prerequisites
  - Python & CLI
- Supports UnitTest, pandas.test() and more
- Stores values inside test cases
  - can specifically inform of which value failed and which value asserted
- Simplicity
  - shorter code with special routines; easy to write, understand; < errors
  - however, special routines compromise compatibility with other frameworks
- Fixtures & plugins
  - apply multiple param combinations w/o rewriting methods; parallel testing

```
a@DESKTOP-IAIPB4P:/mnt/c/Users/Adam/Desktop/Mjukvarutestning$ pytest -rA pandas_series_test.py
============================================================================================================ test session starts =============================================================================================================
platform linux -- Python 3.6.9, pytest-5.3.1, py-1.8.0, pluggy-0.13.1
rootdir: /mnt/c/Users/Adam/Desktop/Mjukvarutestning
collected 4 items

pandas_series_test.py ....                                                                                                                                                                                                          [100%]

================================================================================================================ PASSES ==================================================================================================================
============================================================================================================ short test summary info =============================================================================================================
PASSED pandas_series_test.py::PandasSeriesTest::test_four_divmod_Series_range_10_reversed
PASSED pandas_series_test.py::PandasSeriesTest::test_one_divmod_one
PASSED pandas_series_test.py::PandasSeriesTest::test_three_divmod_Series_range_10
PASSED pandas_series_test.py::PandasSeriesTest::test_two_divmod_two
============================================================================================================ 4 passed in 0.56s ===========================================================================================================
```

Testing tools: **Coverage** *v4.5.4*

- Monitors and measures code coverage of Python programs
- Analyzes source to identify effectiveness of test code in execution

```
a@DESKTOP-IAIPB4P:/mnt/c/Users/Adam/Desktop/Mjukvarutestning$ coverage run pandas_series_test.py
....
----------------------------------------------------------------------
Ran 4 tests in 0.012s

OK
a@DESKTOP-IAIPB4P:/mnt/c/Users/Adam/Desktop/Mjukvarutestning$ coverage report
Name                                                                                                    Stmts   Miss  Cover
---------------------------------------------------------------------------------------------------------------------
/home/a/.local/lib/python3.6/site-packages/_pytest/__init__.py                                              5      2    60%
/home/a/.local/lib/python3.6/site-packages/_pytest/_code/__init__.py                                        9      0   100%
...
/home/a/.local/lib/python3.6/site-packages/zipp.py                                                         75     36    52%
pandas_series_test.py                                                                                      68     11    84%
/usr/lib/python3/dist-packages/attr/__init__.py                                                            19      0   100%
/usr/lib/python3/dist-packages/attr/_compat.py                                                             54     30    44%
/usr/lib/python3/dist-packages/attr/_config.py                                                              9      4    56%
/usr/lib/python3/dist-packages/attr/_funcs.py                                                              68     58    15%
/usr/lib/python3/dist-packages/attr/_make.py                                                              530    158    70%
/usr/lib/python3/dist-packages/attr/converters.py                                                           7      1    86%
/usr/lib/python3/dist-packages/attr/exceptions.py                                                           8      0   100%
/usr/lib/python3/dist-packages/attr/filters.py                                                             15      9    40%
/usr/lib/python3/dist-packages/attr/validators.py                                                          46     17    63%
/usr/lib/python3/dist-packages/six.py                                                                     450    210    53%
---------------------------------------------------------------------------------------------------------------------
TOTAL                                                                                                   82176  62861    24%
```

Prototype Testing

- Tests pandas.Series() support of the inbuilt divmod()
- divmod() is applied elementwise to input Series()
- Tests the input: pandas.Series(numpy.arange(10))
- Minimum test pass requirements for any Series():
  - Each of the quotient and remainder outputs must also be Series() of equal size and data type
  - Outputs must be in range(0, modulus), otherwise outputs are 0 if either modulus or dividend is 0
- Top test:
  - Divide, modulo input by Series(np.arange(10))
  - Expected quotients: [0 1 1 1 1 1 1 1 1 1]
  - Expected remainders: [0 0 0 0 0 0 0 0 0 0]
- Bottom test:
  - Divide, modulo input by Series(np.arange(9, -1, -1))
  - Expected quotients: [0 0 0 0 0 1 2 3 8 0]
  - Expected remainders: [0 1 2 3 4 1 0 1 0 0]

```
def test_three_divmod_Series_range_10(self):
    """..."""
    div, rem = self.ser.divmod(self.ser)

    # The input, and both output Series() must have equal dtype (int64/32):
    self.assertEqual(self.ser.dtype, self.d_type)
    self.assertEqual(div.dtype and rem.dtype, self.d_type)
    # The input, and both output Series() must have equal data sizes/len:
    self.assertTrue(self.ser.size, self.length)
    self.assertEqual(div.size, self.ser.size)
    self.assertEqual(rem.size, self.ser.size)
    # The quotient Series() must equal Series([0 1 1 1 1 1 1 1 1 1]):
    # asarray([0] + [1] * 9) = [0 1 1 1 1 1 1 1 1 1]
    assert_series_equal(div, Series(asarray([0] + [1] * 9)))
    # The rem Series() must be equal to a Series() of 10 zero values:
    # asarray([0] * 10) = [0 0 0 0 0 0 0 0 0 0]
    assert_series_equal(rem, Series(asarray([0] * 10)))

def test_four_divmod_Series_range_10_reversed(self):
    """..."""
    div, rem = self.ser.divmod(Series(arange(9, -1, -1)))

    # The input, and both output Series() must have equal dtype (int64/32):
    self.assertEqual(self.ser.dtype, self.d_type)
    self.assertEqual(div.dtype and rem.dtype, self.d_type)
    # The input, and both output Series() must have equal data sizes/len:
    self.assertTrue(self.ser.size, self.length)
    self.assertEqual(div.size, self.ser.size)
    self.assertEqual(rem.size, self.ser.size)
    # The quotient Series() must equal Series([0 0 0 0 0 1 2 3 8 0]):
    # asarray([0] * 5 + [1 2 3 8 0] = [0 0 0 0 0 1 2 3 8 0]
    assert_series_equal(div, Series(asarray([0] * 5 + [1, 2, 3, 8, 0])))
    # The rem Series() must be equal Series([0 1 2 3 4 1 0 1 0 0]):
    # asarray([0, 1, 2, 3, 4, 1, 0, 1, 0, 0]) = [0 1 2 3 4 1 0 1 0 0]
    assert_series_equal(rem, Series(asarray([0, 1, 2, 3, 4, 1, 0, 1, 0, 0])))
```

Future Plans

- **Testing**:
  - plan multiple test suites with justified motivations
    - White-box testing of specific pandas library code internal structures and workings
    - Black-box testing of the pandas library API for functionality
  - implement testing
  - prove validity of testing
- **Statement and path coverage of testing:**
  - learn and improve implementing coverage.py in testing
  - supply it with control flow graphing
- **Report; write at least:**
  - library description
  - testing strategy
  - control flow graph
  - thorough test case documentation

More Specific Testing Plans

- Shu-Pei Huang
  - Pandas.Series.Count()
- Li Ju
  - Pandas.Series.dropna(), Pandas.DataFrame.dropna(): conduct more test cases for specific coverage
  - More functions/methods to handle NaNs, outliers and other unexpected values in data
- Yin-Hsiang Liao
  -
- Adam Ross
  - Pandas.Series() support for boolean reductions; any of: .empty(); .any(); .all(); and .bool()
  - Pandas Time Series methods (generally, potentially)
- Vladislav Shapar
  -
- Liang-Yan Yu
  - Series.round(), rounding numbers in a series

Challenges

- **Experienced significant instability PyTesting with Python v3.8**
  - changed the Python version to 3.6
- **Time**
  - not having enough free between ourselves for the project so far
- **No group member having any prior experience with either:**
  - pandas
  - PyTest, nor
  - coverage.py