Assignment 7

Ellen Hsieh

1. Unit Testing in Python

(a)

At first, the code provided in the problem 1 cannot pass all the tests in py.test due to the issue of the boundary of the range, which would only run through all the number until n**.5 but not included. Therefore, the range of the test should be from 2 to n**.5+1 to make sure that n**.5 is also included in the range.

```
#smallest_factor.py(revised)

def smallest_factor(n):
    """Return the smallest prime factor of the positive integer n."""
    if n == 1: return 1
    for i in range(2, int(n**.5) + 1):
        if n % i == 0: return i
    return n
```

The test code for the smallest_factor included randomly chose numbers and their correct answers, also, with a number that is a power of certain prime, which is used to test the correctness of the boundary of the range.

```
#test_smallest_factor.py
import smallest_factor as sf

def test_smallest_factor():

    assert sf.smallest_factor(100) == 2, "Incorrect answer"
    assert sf.smallest_factor(3) == 3, "Incorrect answer"
    assert sf.smallest_factor(27) == 3, "Incorrect answer"
    assert sf.smallest_factor(169) == 13, "Incorrect answer"
```

```
============================= test session starts ==============================
platform darwin -- Python 3.6.6, pytest-4.0.0, py-1.7.0, pluggy-0.8.0
rootdir: /Users/ellenhsieh/Documents/UChicago/2018 Fall/Perspecitves on Computatio
nal Analysis/persp-analysis_A18/Assignments/A7/unit test/smallest factor, inifile:
plugins: remotedata-0.3.1, openfiles-0.3.0, cov-2.6.0
collected 1 item

test_smallest_factor.py .                                                 [100%]

---------- coverage: platform darwin, python 3.6.6-final-0 -----------
Name                      Stmts   Miss  Cover
---------------------------------------------
smallest_factor.py            5      0   100%
test_smallest_factor.py       6      0   100%
---------------------------------------------
TOTAL                        11      0   100%


=========================== 1 passed in 0.03 seconds ===========================
```

(b)

In order to cover all the code written in the month_length, the test code test the input, the randomly chose months, and the special case, February, both in leap year or not in leap year.

```python
#test_month_length.py
import month_length as ml

def test_month_length():

    assert ml.month_length("jkisld") == None, "Not a correct input for month"
    assert ml.month_length("September") == 30, "Unexpected days for September"
    assert ml.month_length("March") == 31, "Unexpected days for March"
    assert ml.month_length("February") == 28, "Unexpected days for February not in leap year"
    assert ml.month_length("February", leap_year = True) == 29, "Unexpected days for February in leap year"
```

```
============================= test session starts ==============================
platform darwin -- Python 3.6.6, pytest-4.0.0, py-1.7.0, pluggy-0.8.0
rootdir: /Users/ellenhsieh/Documents/UChicago/2018 Fall/Perspecitves on Computatio
nal Analysis/persp-analysis_A18/Assignments/A7/unit test/month length, inifile:
plugins: remotedata-0.3.1, openfiles-0.3.0, cov-2.6.0
collected 1 item

test_month_length.py .                                                    [100%]

---------- coverage: platform darwin, python 3.6.6-final-0 -----------
Name                   Stmts   Miss  Cover
------------------------------------------
month_length.py           10      0   100%
test_month_length.py       7      0   100%
------------------------------------------
TOTAL                     17      0   100%


=========================== 1 passed in 0.03 seconds ===========================
```

(c)

The test code for operate.py would test the type of the input, the denominator for division that should not be zero, and the input of the operator. Also, the test provides the correct answers for some operations with those specific operators to test operate.py.

```python
#test_operate.py
import pytest
import operate as op

def test_operate():

    with pytest.raises(TypeError) as excinfo:
        op.operate(1, 2, 34)
    assert excinfo.value.args[0] == "oper must be a string"
    assert op.operate(3, 4, '+') == 7, "Unexpected value for operator +"
    assert op.operate(10, 1, '-') == 9, "Unexpected value for operator -"
    assert op.operate(5, 6, '*') == 30, "Unexpected value for operator *"
    assert op.operate(9, 3, '/') == 3, "Unexpected value for operator /"
    with pytest.raises(ZeroDivisionError) as excinfo:
        op.operate(5, 0, '/')
    assert excinfo.value.args[0] == "division by zero is undefined"
    with pytest.raises(ValueError) as excinfo:
        op.operate(7, 8, '**')
    assert excinfo.value.args[0] == "oper must be one of '+', '/', '-', or '*'"
```

```
============================== test session starts ===============================
platform darwin -- Python 3.6.6, pytest-4.0.0, py-1.7.0, pluggy-0.8.0
rootdir: /Users/ellenhsieh/Documents/UChicago/2018 Fall/Perspecitves on Computatio
nal Analysis/persp-analysis_A18/Assignments/A7/unit test/operate, inifile:
plugins: remotedata-0.3.1, openfiles-0.3.0, cov-2.6.0
collected 1 item

test_operate.py .                                                          [100%]

---------- coverage: platform darwin, python 3.6.6-final-0 -----------
Name                Stmts   Miss  Cover
---------------------------------------
operate.py             14      0   100%
test_operate.py        16      0   100%
---------------------------------------
TOTAL                  30      0   100%


============================ 1 passed in 0.04 seconds ============================
```

2. Test driven development

```
def get_r(K, L, alpha, Z, delta):
    '''
    This function generate the interest rate or vector of interest rates
    '''

    r = alpha * Z * (L / K) ** (1 - alpha) - delta

    return r
```

```
============================= test session starts ==============================
platform darwin -- Python 3.6.6, pytest-4.0.0, py-1.7.0, pluggy-0.8.0
rootdir: /Users/ellenhsieh/Documents/UChicago/2018 Fall/Perspecitves on Computatio
nal Analysis/persp-analysis_A18/Assignments/A7, inifile:
plugins: remotedata-0.3.1, openfiles-0.3.0, cov-2.6.0
collected 244 items

test_r.py ............................................................. [ 26%]
........................................................................ [ 56%]
........................................................................ [ 86%]
................................                                         [100%]

---------- coverage: platform darwin, python 3.6.6-final-0 -----------
Name          Stmts   Miss  Cover
---------------------------------
get_r.py          3      0   100%
test_r.py        29      0   100%
---------------------------------
TOTAL            32      0   100%


========================== 244 passed in 1.11 seconds ==========================
```

3. Watts (2014)

Rational choice theory is a framework for understanding and modeling social and economic behavior, in which the individual or collective actions are explained by the intentions, beliefs, circumstances and so on. (Watts, Duncan J. 2014, p.314). As the former Nobel laureate John Harsnayi(1969, p.514) has argued that the rational choice theory can explain a wide variety of empirical facts in terms of a small number of theoretical assumptions. Others have also claimed that this theory provides strong causal inferences in an analytical way, which can yield testable predictions. (Watts, Duncan J. 2014, p.320)

However, some people still question this theory and criticize this approach. First, some people point out that rational choice theory relies on implausible or empirically invalid assumption about the preferences, knowledge and so on. (Watts, Duncan J. 2014, p.320) What's more, they also mention that the predictions yielded by rational choice theory are at odds with empirical evidence. Second, the utility maximization in the theory is criticized due to its weak assumptions in rationality. Third, even some of the rational choice theorists, who regard the backward-looking behavior as permissible (Macy 1993; Macy and Flache 2002), has challenged the concept of rational choice theory that the actions should be forward looking and purposive. Finally, those criticisms have triggered the development of the rational choice theory, therefore, it became more prone to the analyst rather than to follow certain specific rules. (Watts, Duncan J. 2014, p.321)

Commonsense theories of action explain how human predict the behavior of others based on their commonsense rationalization. Nevertheless, the main pitfall for using those theories is that commonsense notion is not necessarily universal. As the Boudon (1988, p.1) mentioned in his study that "a theory can easily be perceived as true when it is false, or as more valid than it deserves to be, if it includes beside its explicit statements implicit unnoticed commonsensical statements, which, although valid in everyday life, are not of universal validity". The social scientists often use mental simulation to explain and construct the theories, which implies that the conflation of empathetic and causal explanation might not be able to explicitly acknowledged as the authors point out (Watts, Duncan J. 2014, p.327). That is to say, theorizing social phenomena via mental simulation such as using commonsense theories of action can undermine the validity of the results.

Since understanding the social behavior in an empathetic sense is not a reliable way for sociological explanations, the authors point out some proposed solution to this issue. First, the most straightforward approach is experimental methods (Harrison and List 2004; Gerber and Green 2012) such as field experiments which have been long regarded as gold standard in medicine, or natural experiments and quasi experiments which occur randomly, and laboratory experiment which has been commonly used in psychology. Another alternative is the counterfactual model of causal inference applying non-experimental data (Rubin 1974; Morgan and Winship 2007). To be more specific, the researchers can analyze the observational data such as online archival records by using some computational methods. Besides the two methods mentioned above, evaluating the explanations for their ability to predict can also be another way to solve the problem.

Even though the authors argue that sociologists rely on common sense more than they realize (Watts, Duncan J. 2014, p.314) and criticize the rational choice theory and other theoretical theories based on rationalizable action, the theoretical models are still critical in social science. Without any theoretical models, it would be really hard for social scientists to simplify their findings and present to other people. Perhaps, researchers can illustrate their observation with the data they collected, however, it would be difficult for people to understand and be convinced in a short time. What's more, social behavior is hard to predict and explain due to the fact that all individual is unique and different from others. Thus, without any assumptions, it would be hard to say that the prediction can be appropriately yielded and reflect the reality. In conclusion, the paper indeed reasonably relates the causality to prediction, however, the theoretical models are also beneficial to relate causal inference and prediction with their simplicity and specific assumptions about mechanism.

Reference

Watts, Duncan J., "Common Sense and Sociological Explanations," American Journal of Sociology, September 2014, 120 (2), 313–351.