Coverage for **test_r.py** : 100%

29 statements | 29 run | 0 missing | 0 excluded

```
'''
--------------------------------------------------------------------------
This script tests the function get_r() in the get_r.py module
--------------------------------------------------------------------------
'''

# Import packages
import pytest
import numpy as np
import get_r as gr


@pytest.mark.parametrize('K',
                         [0.1, 203.5,
                          np.array([90.5, 130.2, 141.7, 140.0, 135.8])])
@pytest.mark.parametrize('L',
                         [0.1, 150.7,
                          np.array([76.5, 82.2, 85.7, 83.0, 79.8])])
@pytest.mark.parametrize('alpha', [0.1, 0.3, 0.7])
@pytest.mark.parametrize('Z', [0.1, 1.0, 10.0])
@pytest.mark.parametrize('delta', [0.0, 0.05, 0.1])
def test_get_r(K, L, alpha, Z, delta):
    '''
    --------------------------------------------------------------------
    This test ensures that the sizes of the output vectors correspond to
    the inputs. It also makes sure that the resulting interest rates are
    greater than delta
    --------------------------------------------------------------------
    '''
    r = gr.get_r(K, L, alpha, Z, delta)
    if np.isscalar(K) and np.isscalar(L):
        assert (np.isscalar(r) and r > -delta and np.isfinite(r))
    elif not np.isscalar(K) and not np.isscalar(L):
        assert (r.shape == K.shape)
        assert (r > -delta).sum() == len(r)


def test_get_rvals():
    alpha = 0.3
    Z = 1.0
    delta = 0.05
    K = 1.0
    L = 2.0
    r = gr.get_r(K, L, alpha, Z, delta)
    assert np.isclose(r, 0.43735143781374125)

    alpha2 = 0.4
    Z2 = 0.5
    delta2 = 0.01
    K2 = np.array([90.5, 130.2, 141.7, 140.0, 135.8])
    L2 = np.array([76.5, 82.2, 85.7, 83.0, 79.8])
    r2 = gr.get_r(K2, L2, alpha2, Z2, delta2)
    assert np.allclose(r2, np.array([0.17081635, 0.1417702, 0.13790967,
                                     0.13615041, 0.13537572]))
```