```
import unittest
import ROOT
from RegressionTest import RegressionTest, AggregateRegressionTests

class MyUnitTest(unittest.TestCase):
    def test_regression(self):
        root_file = ROOT.TFile('test_mon.root', 'RECREATE')
        h1 = ROOT.TH1D("test1", "test1", 100, 0, 20)
        h2 = ROOT.TH1D("test2", "test2", 100, 0, 20)
        h3 = ROOT.TH1D("test3", "test3", 100, 0, 20)
        r  = ROOT.TRandom3(7765234724234);
        for i in range(10000):
            h1.Fill(r.Gaus(10))
            h2.Fill(r.Poisson(10))
            h3.Fill(r.Gaus(12))
        h1.Write()
        h2.Write()
        h3.Write()
        root_file.Close()

        print RegressionTest('test_mon.root', 'test_ref.root')

        self.assertTrue(AggregateRegressionTests('test_mon.root',
                                                 'test_ref.root'),
                        'Regression tests failed.')

if __name__ == '__main__':
    unittest.main()
```

```
  {  'test1': {'Chi2Test':           TestResult(status='Pass', pval=0.514611805089563,
                                                warning=0.1, failure=0.05),
               'KolmogorovTest': TestResult(status='Pass', pval=0.984136074222734 7,
                                                warning=0.1, failure=0.05)
               },
     'test2': {'Chi2Test':           TestResult(status='Pass', pval=0.834108157976536 8,
                                                warning=0.1, failure=0.05),
               'KolmogorovTest': TestResult(status='Pass', pval=0.912647181089197 4,
                                                warning=0.1, failure=0.05)
               },
     'test3': {'Chi2Test':           TestResult(status='Failure', pval=0.0,
                                                warning=0.1, failure=0.05),
               'KolmogorovTest': TestResult(status='Failure', pval=0.0,
                                                warning=0.1, failure=0.05)
               }
  }
F
======================================================================
FAIL: test_regression (__main__.MyUnitTest)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "MyUnitTest.py", line 31, in test_regression
    'Regression tests failed.')
AssertionError: Regression tests failed.

----------------------------------------------------------------------
Ran 1 test in 0.496s

FAILED (failures=1)
```

```
RegressionTest( monitored_filename,
                reference_filename,
                config              = {},
                default_config   = [Chi2Test(0.1,0.05), KolmogorovTest(0.1,0.05)],
                type_to_extract = ROOT.TH1,
                fail_on_missing = False)
```

---

monitored_filename: type(str) name of the file to be checked against a reference file

---

reference_filename: type(str) name of the reference file against which the monitored is checked

---

config: { <pattern> : [<test>] }

<pattern> = type(str) representing the regex pattern to match to extracted objects name
<test>   = type(Test) representing the tests to perform

---

default_config: [<test>]

Configuration used by default unless overridden by config
<test>   = type(Test) representing the tests to perform

---

type_to_extract: Give the object type that you wish to extract and perform tests on

---

fail_on_missing: type(bool) determines if exception (hence failure) raised if reference file defines an object which has no corresponding object in the monitored file. By default it is just skipped

---

Test = namedTuple(<name>, <test_func>, <pVal_Warning>, pVal_Failure)

<name> = type(str) a name for the test
<test_fun> = a callable object taking monitored and reference objects and returning a pVal
<pVal_Warning> = type(double) the pVal below which the test returns a warning
<pVal_Failure> = type(double) the pVal below which the test returns failure

---

Special Cases:

Chi2Test(<pVal_Warning>, <pVal_Failure>) = Test("Chi2Test",
X2,
<pVal_Warning>,
<pVal_Failure>)

KolmogorovTest(<pVal_Warning>, <pVal_Failure>) = Test("KolmogorovTest",
KS,
<pVal_Warning>,
<pVal_Failure>)

```
def KS(mon, ref):
    return ref.KolmogorovTest(mon)

def X2(mon, ref):
    return ref.Chi2Test(mon)
```

```
  {  'test1': {'Chi2Test':           TestResult(status='Pass', pval=0.514611805089563,
                                                warning=0.1, failure=0.05),
               'KolmogorovTest': TestResult(status='Pass', pval=0.984136074222734 7,
                                                warning=0.1, failure=0.05)
               },
     'test2': {'Chi2Test':           TestResult(status='Pass', pval=0.834108157976536 8,
                                                warning=0.1, failure=0.05),
               'KolmogorovTest': TestResult(status='Pass', pval=0.912647181089197 4,
                                                warning=0.1, failure=0.05)
               },
     'test3': {'Chi2Test':           TestResult(status='Failure', pval=0.0,
                                                warning=0.1, failure=0.05),
               'KolmogorovTest': TestResult(status='Failure', pval=0.0,
                                                warning=0.1, failure=0.05)
               }
  }
F
======================================================================
FAIL: test_regression (__main__.MyUnitTest)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "MyUnitTest.py", line 31, in test_regression
    'Regression tests failed.')
AssertionError: Regression tests failed.

----------------------------------------------------------------------
Ran 1 test in 0.496s

FAILED (failures=1)
```

```
import unittest
import ROOT
from RegressionTest import RegressionTest, AggregateRegressionTests

class MyUnitTest(unittest.TestCase):
    def test_regression(self):
        root_file = ROOT.TFile('test_mon.root', 'RECREATE')
        h1 = ROOT.TH1D("test1", "test1", 100, 0, 20)
        h2 = ROOT.TH1D("test2", "test2", 100, 0, 20)
        h3 = ROOT.TH1D("test3", "test3", 100, 0, 20)
        r  = ROOT.TRandom3(7765234724234);
        for i in range(10000):
            h1.Fill(r.Gaus(10))
            h2.Fill(r.Poisson(10))
            h3.Fill(r.Gaus(12))
        h1.Write()
        h2.Write()
        h3.Write()
        root_file.Close()

        print RegressionTest('test_mon.root', 'test_ref.root', {'test3': [ ]})

        self.assertTrue(AggregateRegressionTests('test_mon.root', 'test_ref.root', {'test3': [ ]}),
                        'Regression tests failed.')

if __name__ == '__main__':
    unittest.main()
```

```
{  'test1': {'Chi2Test':           TestResult(status='Pass', pval=0.514611880508956 3,
                                              warning=0.1, failure=0.05),
             'KolmogorovTest': TestResult(status='Pass', pval=0.984136074222734 7,
                                              warning=0.1, failure=0.05)
            },
   'test2': {'Chi2Test':           TestResult(status='Pass', pval=0.834108157976536 8,
                                              warning=0.1, failure=0.05),
             'KolmogorovTest': TestResult(status='Pass', pval=0.912647181089197 4,
                                              warning=0.1, failure=0.05)
            },
   'test3': {}
}
```

```
.
----------------------------------------------------------------------
Ran 1 test in 0.414s

OK
```

```
import unittest
import ROOT
from RegressionTest import RegressionTest, AggregateRegressionTests, Chi2Test

class MyUnitTest(unittest.TestCase):
    def test_regression(self):
        root_file = ROOT.TFile('test_mon.root', 'RECREATE')
        h1 = ROOT.TH1D("test1", "test1", 100, 0, 20)
        h2 = ROOT.TH1D("test2", "test2", 100, 0, 20)
        h3 = ROOT.TH1D("test3", "test3", 100, 0, 20)
        r  = ROOT.TRandom3(7765234724234);
        for i in range(10000):
            h1.Fill(r.Gaus(10))
            h2.Fill(r.Poisson(10))
            h3.Fill(r.Gaus(12))
        h1.Write()
        h2.Write()
        h3.Write()
        root_file.Close()

        print RegressionTest('test_mon.root', 'test_ref.root', {'test3': [ Chi2Test(0.0, 0.0 ) ]})

        self.assertTrue(AggregateRegressionTests('test_mon.root', 'test_ref.root',
                                                 {'test3': [ Chi2Test(0.0, 0.0 ) ]} ),
                        'Regression tests failed.')

if __name__ == '__main__':
    unittest.main()
```

```
{  'test1': {'Chi2Test':           TestResult(status='Pass', pval=0.514611880508956 3,
                                              warning=0.1, failure=0.05),
             'KolmogorovTest': TestResult(status='Pass', pval=0.984136074222734 7,
                                              warning=0.1, failure=0.05)
            },
   'test2': {'Chi2Test':           TestResult(status='Pass', pval=0.834108157976536 8,
                                              warning=0.1, failure=0.05),
             'KolmogorovTest': TestResult(status='Pass', pval=0.912647181089197 4,
                                              warning=0.1, failure=0.05)
            },
   'test3': {'Chi2Test':           TestResult(status='Pass', pval=0.0,
                                              warning=0.0, failure=0.0)
            }
}
```

```
.
----------------------------------------------------------------------
Ran 1 test in 0.414s

OK
```

```
import unittest
import ROOT
from RegressionTest import RegressionTest, AggregateRegressionTests, Test

def myTest(mon, ref):
        ## do stuff here with the two objects (whatever they are)
        return 0.9

class MyUnitTest(unittest.TestCase):
    def test_regression(self):
        root_file = ROOT.TFile('test_mon.root', 'RECREATE')
        h1 = ROOT.TH1D("test1", "test1", 100, 0, 20)
        h2 = ROOT.TH1D("test2", "test2", 100, 0, 20)
        h3 = ROOT.TH1D("test3", "test3", 100, 0, 20)
        r  = ROOT.TRandom3(7765234724234);
        for i in range(10000):
            h1.Fill(r.Gaus(10))
            h2.Fill(r.Poisson(10))
            h3.Fill(r.Gaus(12))
        h1.Write()
        h2.Write()
        h3.Write()
        root_file.Close()

        print RegressionTest('test_mon.root', 'test_ref.root', {'test3': [ Test("MyTest", myTest,0.5, 0.1 ) ]})

        self.assertTrue(AggregateRegressionTests('test_mon.root', 'test_ref.root',
                                                 {'test3': [ Test("MyTest", myTest,0.5, 0.1 ) ]} ),
                        'Regression tests failed.')

if __name__ == '__main__':
    unittest.main()
```

```
{  'test1': {'Chi2Test':           TestResult(status='Pass', pval=0.514611880508956 3,
                                              warning=0.1, failure=0.05),
             'KolmogorovTest': TestResult(status='Pass', pval=0.984136074222734 7,
                                              warning=0.1, failure=0.05)
             },
   'test2': {'Chi2Test':           TestResult(status='Pass', pval=0.834108157976536 8,
                                              warning=0.1, failure=0.05),
             'KolmogorovTest': TestResult(status='Pass', pval=0.912647181089197 4,
                                              warning=0.1, failure=0.05)
             },
   'test3':  {'MyTest':             TestResult(status='Pass', pval=0.9,
                                              warning=0.5, failure=0.1)
             }
}
```

```
.
----------------------------------------------------------------------
Ran 1 test in 0.414s

OK
```