PyStat Documentation

Release 0.1.1a

Tirthankar Chakravarty

July 29, 2014

CONTENTS

CHAPTER

ONE

INTRODUCTION

This package contains commonly used statistical routines implemented in Python. The functionality is organized in 2 broad categories:

- Estimation routines
- Testing routines

The classical, mostly parametric tests implemented in this package are taken from the

CHAPTER

TWO

STATISTICAL TESTS

tests.tests.**test_correlation**(*series1*, *series2*, *correlation*, *test_type='z'*)

Test population correlation is equal to a given value if the correlation passed is zero, then a t-test is called else, a z-test is called. Actually it is possible to also compute the z-test when the hypothesised population correlation is zero.

Param series1, series2: the pandas series whose correlation has to be computed

Returns the test statistic, the p-value, the degrees of freedom, and the type of the test

tests.tests.**test_mean**(*series*, *mean*, *variance=None*, *type='two-sided'*)

Test that the mean of the series is equal to given number.

Param series: a pandas Series

Param mean: the hypothesised true mean of the population

Param variance: if supplied, the known variance of the population

Returns a dict with the statistic and the p-value

This module implements two tests of population mean – one where the population variance is assumed to be known, and the other where the population variance is assumed to be unknown.

^ Test: Z-test for population mean (variance known)

Data: The data is assumed to be a random (IID) sample from a normal population [1].

Hypothesis:

$$H_0 : \mu = \mu_0$$
$$H_1 : \mu \neq \mu_0$$

```
import pandas as pd
import numpy as np
# import unittest as ut

from PyStatTests.tests import tests

#================================================
# test 1: Z-test for a population mean (variance known)
# test 7: t-test for population mean (variance unknown)
#================================================

print("================================================================")
```

[1] Note that the distribution of the statistic might be robust to other kinds of populations and sampling schemes but we discuss only the cases in [kanji98].

```
    print("Tests of population mean under the null")
    print("================================================================")
    # generate the univariate series
    series1 = pd.Series(np.random.randn(10))
    help(tests.test_means)
    # test 1: null  DGP
    print(tests.test_mean(series1, mean = 0, variance = 1))
    # test 7: null DGP
    print(tests.test_mean(series1, mean = 0))

    print("================================================================")
    print("Tests of population mean under the alternative")
    print("================================================================")
    # test 1: alternative DGP
    print(tests.test_mean(series1, mean = 2, variance = 1))
    # test 7: alternative DGP
    print(tests.test_mean(series1, mean = 2))
```

tests.tests.**test_means**(*series1*, *series2*, *variance1=None*, *variance2=None*, *var_equal=False*, *var_unknown=False*, *type='two-sided'*)

Test whether two population means are different

Param series1, series 2: two pandas series from the two populations to be compared

Param variance1, variance2: if supplied, the known variances of the populations

Param var_equal: flag for if the two variances are equal

Param var_unknown: flag for whether the variances are unknown

Param type: whether to computer a two-sided, upper or lower tailed test

Returns a dict with the statistic, the p-value and the degrees of freedom of the test

tests.tests.**test_proportion**(*series*, *proportion*)

Test whether the population proportion is equal to a given value

Param series: sample of data from a population

Param proportion: hypothesised proportion of the binomial population

Returns a dict containing the test statistic and the p-value of the test

tests.tests.**test_proportions**(*series1*, *series2*)

Test the significance of the difference between two proportions

Param series1, series2: pandas series from which the proportion is to be computed

Returns a dict with the statistic and the p-value

tests.tests.**test_variance**(*series*, *variance*)

Test whether the population variance is equal to a given value

Param series: sample of data from a population

Param variance: hypothesised variance of the population

Returns a dict containing the statistic, p-value and the degrees of freedom of the test

CHAPTER

THREE

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

CHAPTER

FOUR

BIBLIOGRAPHY

[kanji98] Kanji, Gopal K. 100 statistical tests. Sage, 2006.

t

T