17 Statistical Hypothesis Tests in Python (Cheat Sheet)

by **Jason Brownlee** on August 15, 2018 in **Statistics**

Last Updated on November 28, 2019

Quick-reference guide to the 17 statistical hypothesis tests that you need in applied machine learning, with sample code in Python.

Although there are hundreds of statistical hypothesis tests that you could use, there is only a small subset that you may need to use in a machine learning project.

In this post, you will discover a cheat sheet for the most popular statistical hypothesis tests for a machine learning project with examples using the Python API.

Each statistical test is presented in a consistent way, including:

- The name of the test.
- What the test is checking.
- The key assumptions of the test.
- How the test result is interpreted.
- Python API for using the test.

Note, when it comes to assumptions such as the expected distribution of data or sample size, the results of a given test are likely to degrade gracefully rather than become immediately unusable if an assumption is violated.

Generally, data samples need to be representative of the domain and large enough to expose their distribution to analysis.

In some cases, the data can be corrected to meet the assumptions, such as correcting a nearly normal distribution to be normal by removing outliers, or using a correction to the degrees of freedom in a statistical test when samples have differing variance, to name two examples.

Finally, there may be multiple tests for a given concern, e.g. normality. We cannot get crisp answers to questions with statistics; instead, we get probabilistic answers. As such, we can arrive at different answers to the same question by considering the question in different ways. Hence the need for multiple different tests for some questions we may have about data.

Kick-start your project with my new book Statistics for Machine Learning, including *step-by-step tutorials* and the *Python source code* files for all examples.

Let's get started.

- **Update Nov/2018**: Added a better overview of the tests covered.
- **Update Nov/2019**: Added complete working examples of each test. Add time series tests.

Statistical Hypothesis Tests in Python Cheat Sheet
Photo by davemichuda, some rights reserved.

Tutorial Overview

This tutorial is divided into 5 parts; they are:

1. **Normality Tests**
    1. Shapiro-Wilk Test
    2. D'Agostino's K^2 Test
    3. Anderson-Darling Test
2. **Correlation Tests**
    1. Pearson's Correlation Coefficient
    2. Spearman's Rank Correlation
    3. Kendall's Rank Correlation
    4. Chi-Squared Test
3. **Stationary Tests**
    1. Augmented Dickey-Fuller
    2. Kwiatkowski-Phillips-Schmidt-Shin
4. **Parametric Statistical Hypothesis Tests**
    1. Student's t-test
    2. Paired Student's t-test
    3. Analysis of Variance Test (ANOVA)
    4. Repeated Measures ANOVA Test
5. **Nonparametric Statistical Hypothesis Tests**
    1. Mann-Whitney U Test
    2. Wilcoxon Signed-Rank Test
    3. Kruskal-Wallis H Test
    4. Friedman Test

1. Normality Tests

This section lists statistical tests that you can use to check if your data has a Gaussian distribution.

Shapiro-Wilk Test

Tests whether a data sample has a Gaussian distribution.

Assumptions

- Observations in each sample are independent and identically distributed (iid).

Interpretation

- H0: the sample has a Gaussian distribution.
- H1: the sample does not have a Gaussian distribution.

Python Code

```python
# Example of the Shapiro-Wilk Normality Test
from scipy.stats import shapiro
data = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
stat, p = shapiro(data)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably Gaussian')
else:
    print('Probably not Gaussian')
```

More Information

- A Gentle Introduction to Normality Tests in Python
- scipy.stats.shapiro
- Shapiro-Wilk test on Wikipedia

D'Agostino's K^2 Test

Tests whether a data sample has a Gaussian distribution.

Assumptions

- Observations in each sample are independent and identically distributed (iid).

Interpretation

- H0: the sample has a Gaussian distribution.
- H1: the sample does not have a Gaussian distribution.

Python Code

```python
# Example of the D'Agostino's K^2 Normality Test
from scipy.stats import normaltest
data = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
stat, p = normaltest(data)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably Gaussian')
else:
    print('Probably not Gaussian')
```

More Information

- A Gentle Introduction to Normality Tests in Python
- scipy.stats.normaltest
- D'Agostino's K-squared test on Wikipedia

Anderson-Darling Test

Tests whether a data sample has a Gaussian distribution.

Assumptions

- Observations in each sample are independent and identically distributed (iid).

Interpretation

- H0: the sample has a Gaussian distribution.
- H1: the sample does not have a Gaussian distribution.

Python Code

```python
# Example of the Anderson-Darling Normality Test
from scipy.stats import anderson
data = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
result = anderson(data)
print('stat=%.3f' % (result.statistic))
for i in range(len(result.critical_values)):
    sl, cv = result.significance_level[i], result.critical_values[i]
    if result.statistic < cv:
        print('Probably Gaussian at the %.1f%% level' % (sl))
    else:
        print('Probably not Gaussian at the %.1f%% level' % (sl))
```

More Information

- A Gentle Introduction to Normality Tests in Python
- scipy.stats.anderson
- Anderson-Darling test on Wikipedia

2. Correlation Tests

This section lists statistical tests that you can use to check if two samples are related.

Pearson's Correlation Coefficient

Tests whether two samples have a linear relationship.

Assumptions

- Observations in each sample are independent and identically distributed (iid).
- Observations in each sample are normally distributed.
- Observations in each sample have the same variance.

Interpretation

- H0: the two samples are independent.
- H1: there is a dependency between the samples.

Python Code

```python
# Example of the Pearson's Correlation test
from scipy.stats import pearsonr
data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
data2 = [0.353, 3.517, 0.125, -7.545, -0.555, -1.536, 3.350, -1.578, -3.537, -1.579]
stat, p = pearsonr(data1, data2)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably independent')
else:
    print('Probably dependent')
```

More Information

- How to Calculate Correlation Between Variables in Python
- scipy.stats.pearsonr
- Pearson's correlation coefficient on Wikipedia

Spearman's Rank Correlation

Tests whether two samples have a monotonic relationship.

Assumptions

- Observations in each sample are independent and identically distributed (iid).
- Observations in each sample can be ranked.

Interpretation

- H0: the two samples are independent.
- H1: there is a dependency between the samples.

Python Code

```python
# Example of the Spearman's Rank Correlation Test
from scipy.stats import spearmanr
data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
data2 = [0.353, 3.517, 0.125, -7.545, -0.555, -1.536, 3.350, -1.578, -3.537, -1.579]
stat, p = spearmanr(data1, data2)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably independent')
else:
    print('Probably dependent')
```

More Information

- How to Calculate Nonparametric Rank Correlation in Python
- scipy.stats.spearmanr
- Spearman's rank correlation coefficient on Wikipedia

Kendall's Rank Correlation

Tests whether two samples have a monotonic relationship.

Assumptions

- Observations in each sample are independent and identically distributed (iid).
- Observations in each sample can be ranked.

Interpretation

- H0: the two samples are independent.
- H1: there is a dependency between the samples.

Python Code

```python
# Example of the Kendall's Rank Correlation Test
from scipy.stats import kendalltau
data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
data2 = [0.353, 3.517, 0.125, -7.545, -0.555, -1.536, 3.350, -1.578, -3.537, -1.579]
stat, p = kendalltau(data1, data2)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably independent')
else:
    print('Probably dependent')
```

More Information

- How to Calculate Nonparametric Rank Correlation in Python
- scipy.stats.kendalltau
- Kendall rank correlation coefficient on Wikipedia

Chi-Squared Test

Tests whether two categorical variables are related or independent.

Assumptions

- Observations used in the calculation of the contingency table are independent.

- 25 or more examples in each cell of the contingency table.

Interpretation

- H0: the two samples are independent.
- H1: there is a dependency between the samples.

Python Code

```python
# Example of the Chi-Squared Test
from scipy.stats import chi2_contingency
table = [[10, 20, 30],[6,  9,  17]]
stat, p, dof, expected = chi2_contingency(table)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably independent')
else:
    print('Probably dependent')
```

More Information

- A Gentle Introduction to the Chi-Squared Test for Machine Learning
- scipy.stats.chi2_contingency
- Chi-Squared test on Wikipedia

3. Stationary Tests

This section lists statistical tests that you can use to check if a time series is stationary or not.

Augmented Dickey-Fuller Unit Root Test

Tests whether a time series has a unit root, e.g. has a trend or more generally is autoregressive.

Assumptions

- Observations in are temporally ordered.

Interpretation

- H0: a unit root is present (series is non-stationary).
- H1: a unit root is not present (series is stationary).

Python Code

```python
# Example of the Augmented Dickey-Fuller unit root test
from statsmodels.tsa.stattools import adfuller
data = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
stat, p, lags, obs, crit, t = adfuller(data)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably not Stationary')
else:
    print('Probably Stationary')
```

More Information

- How to Check if Time Series Data is Stationary with Python
- statsmodels.tsa.stattools.adfuller API.
- Augmented Dickey–Fuller test, Wikipedia.

Kwiatkowski-Phillips-Schmidt-Shin

Tests whether a time series is trend stationary or not.

Assumptions

- Observations in are temporally ordered.

Interpretation

- H0: the time series is not trend-stationary.
- H1: the time series is trend-stationary.

Python Code

```python
# Example of the Kwiatkowski-Phillips-Schmidt-Shin test
from statsmodels.tsa.stattools import kpss
data = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
stat, p, lags, crit = kpss(data)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably not Stationary')
else:
    print('Probably Stationary')
```

More Information

- statsmodels.tsa.stattools.kpss API.
- KPSS test, Wikipedia.

4. Parametric Statistical Hypothesis Tests

This section lists statistical tests that you can use to compare data samples.

Student’s t-test

Tests whether the means of two independent samples are significantly different.

Assumptions

- Observations in each sample are independent and identically distributed (iid).
- Observations in each sample are normally distributed.
- Observations in each sample have the same variance.

Interpretation

- H0: the means of the samples are equal.
- H1: the means of the samples are unequal.

Python Code

```python
# Example of the Student's t-test
from scipy.stats import ttest_ind
data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
data2 = [1.142, -0.432, -0.938, -0.729, -0.846, -0.157, 0.500, 1.183, -1.075, -0.169]
stat, p = ttest_ind(data1, data2)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably the same distribution')
else:
    print('Probably different distributions')
```

More Information

- How to Calculate Parametric Statistical Hypothesis Tests in Python
- scipy.stats.ttest_ind
- Student's t-test on Wikipedia

Paired Student's t-test

Tests whether the means of two paired samples are significantly different.

Assumptions

- Observations in each sample are independent and identically distributed (iid).
- Observations in each sample are normally distributed.
- Observations in each sample have the same variance.
- Observations across each sample are paired.

Interpretation

- H0: the means of the samples are equal.
- H1: the means of the samples are unequal.

Python Code

```python
# Example of the Paired Student's t-test
from scipy.stats import ttest_rel
data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
data2 = [1.142, -0.432, -0.938, -0.729, -0.846, -0.157, 0.500, 1.183, -1.075, -0.169]
stat, p = ttest_rel(data1, data2)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably the same distribution')
else:
    print('Probably different distributions')
```

More Information

- How to Calculate Parametric Statistical Hypothesis Tests in Python
- scipy.stats.ttest_rel
- Student's t-test on Wikipedia

Analysis of Variance Test (ANOVA)

Tests whether the means of two or more independent samples are significantly different.

Assumptions

- Observations in each sample are independent and identically distributed (iid).
- Observations in each sample are normally distributed.
- Observations in each sample have the same variance.

Interpretation

- H0: the means of the samples are equal.
- H1: one or more of the means of the samples are unequal.

Python Code

```python
# Example of the Analysis of Variance Test
from scipy.stats import f_oneway
data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
data2 = [1.142, -0.432, -0.938, -0.729, -0.846, -0.157, 0.500, 1.183, -1.075, -0.169]
data3 = [-0.208, 0.696, 0.928, -1.148, -0.213, 0.229, 0.137, 0.269, -0.870, -1.204]
stat, p = f_oneway(data1, data2, data3)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably the same distribution')
else:
    print('Probably different distributions')
```

More Information

- How to Calculate Parametric Statistical Hypothesis Tests in Python
- scipy.stats.f_oneway
- Analysis of variance on Wikipedia

Repeated Measures ANOVA Test

Tests whether the means of two or more paired samples are significantly different.

Assumptions

- Observations in each sample are independent and identically distributed (iid).
- Observations in each sample are normally distributed.
- Observations in each sample have the same variance.
- Observations across each sample are paired.

Interpretation

- H0: the means of the samples are equal.
- H1: one or more of the means of the samples are unequal.

Python Code

Currently not supported in Python.

More Information

- How to Calculate Parametric Statistical Hypothesis Tests in Python
- Analysis of variance on Wikipedia

5. Nonparametric Statistical Hypothesis Tests

Mann-Whitney U Test

Tests whether the distributions of two independent samples are equal or not.

Assumptions

- Observations in each sample are independent and identically distributed (iid).
- Observations in each sample can be ranked.

Interpretation

- H0: the distributions of both samples are equal.
- H1: the distributions of both samples are not equal.

Python Code

```python
# Example of the Mann-Whitney U Test
from scipy.stats import mannwhitneyu
data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
data2 = [1.142, -0.432, -0.938, -0.729, -0.846, -0.157, 0.500, 1.183, -1.075, -0.169]
stat, p = mannwhitneyu(data1, data2)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably the same distribution')
else:
    print('Probably different distributions')
```

More Information

- How to Calculate Nonparametric Statistical Hypothesis Tests in Python
- scipy.stats.mannwhitneyu
- Mann-Whitney U test on Wikipedia

Wilcoxon Signed-Rank Test

Tests whether the distributions of two paired samples are equal or not.

Assumptions

- Observations in each sample are independent and identically distributed (iid).
- Observations in each sample can be ranked.
- Observations across each sample are paired.

Interpretation

- H0: the distributions of both samples are equal.
- H1: the distributions of both samples are not equal.

Python Code

```python
# Example of the Wilcoxon Signed-Rank Test
from scipy.stats import wilcoxon
data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
data2 = [1.142, -0.432, -0.938, -0.729, -0.846, -0.157, 0.500, 1.183, -1.075, -0.169]
stat, p = wilcoxon(data1, data2)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably the same distribution')
else:
    print('Probably different distributions')
```

More Information

- How to Calculate Nonparametric Statistical Hypothesis Tests in Python
- scipy.stats.wilcoxon
- Wilcoxon signed-rank test on Wikipedia

Kruskal-Wallis H Test

Tests whether the distributions of two or more independent samples are equal or not.

Assumptions

- Observations in each sample are independent and identically distributed (iid).
- Observations in each sample can be ranked.

Interpretation

- H0: the distributions of all samples are equal.
- H1: the distributions of one or more samples are not equal.

Python Code

```python
# Example of the Kruskal-Wallis H Test
from scipy.stats import kruskal
data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
data2 = [1.142, -0.432, -0.938, -0.729, -0.846, -0.157, 0.500, 1.183, -1.075, -0.169]
```

```
stat, p = kruskal(data1, data2)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably the same distribution')
else:
    print('Probably different distributions')
```

More Information

- How to Calculate Nonparametric Statistical Hypothesis Tests in Python
- scipy.stats.kruskal
- Kruskal-Wallis one-way analysis of variance on Wikipedia

Friedman Test

Tests whether the distributions of two or more paired samples are equal or not.

Assumptions

- Observations in each sample are independent and identically distributed (iid).
- Observations in each sample can be ranked.
- Observations across each sample are paired.

Interpretation

- H0: the distributions of all samples are equal.
- H1: the distributions of one or more samples are not equal.

Python Code

```
# Example of the Friedman Test
from scipy.stats import friedmanchisquare
data1 = [0.873, 2.817, 0.121, -0.945, -0.055, -1.436, 0.360, -1.478, -1.637, -1.869]
data2 = [1.142, -0.432, -0.938, -0.729, -0.846, -0.157, 0.500, 1.183, -1.075, -0.169]
data3 = [-0.208, 0.696, 0.928, -1.148, -0.213, 0.229, 0.137, 0.269, -0.870, -1.204]
stat, p = friedmanchisquare(data1, data2, data3)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably the same distribution')
else:
    print('Probably different distributions')
```

More Information

- How to Calculate Nonparametric Statistical Hypothesis Tests in Python
- scipy.stats.friedmanchisquare
- Friedman test on Wikipedia

Further Reading

This section provides more resources on the topic if you are looking to go deeper.

- A Gentle Introduction to Normality Tests in Python
- How to Use Correlation to Understand the Relationship Between Variables
- How to Use Parametric Statistical Significance Tests in Python
- A Gentle Introduction to Statistical Hypothesis Tests

Summary

In this tutorial, you discovered the key statistical hypothesis tests that you may need to use in a machine learning project.

Specifically, you learned:

- The types of tests to use in different circumstances, such as normality checking, relationships between variables, and differences between samples.
- The key assumptions for each test and how to interpret the test result.
- How to implement the test using the Python API.

Do you have any questions?
Ask your questions in the comments below and I will do my best to answer.

Did I miss an important statistical test or key assumption for one of the listed tests?
Let me know in the comments below.

About Jason Brownlee

Jason Brownlee, PhD is a machine learning specialist who teaches developers how to get results with modern machine learning methods via hands-on tutorials.