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.

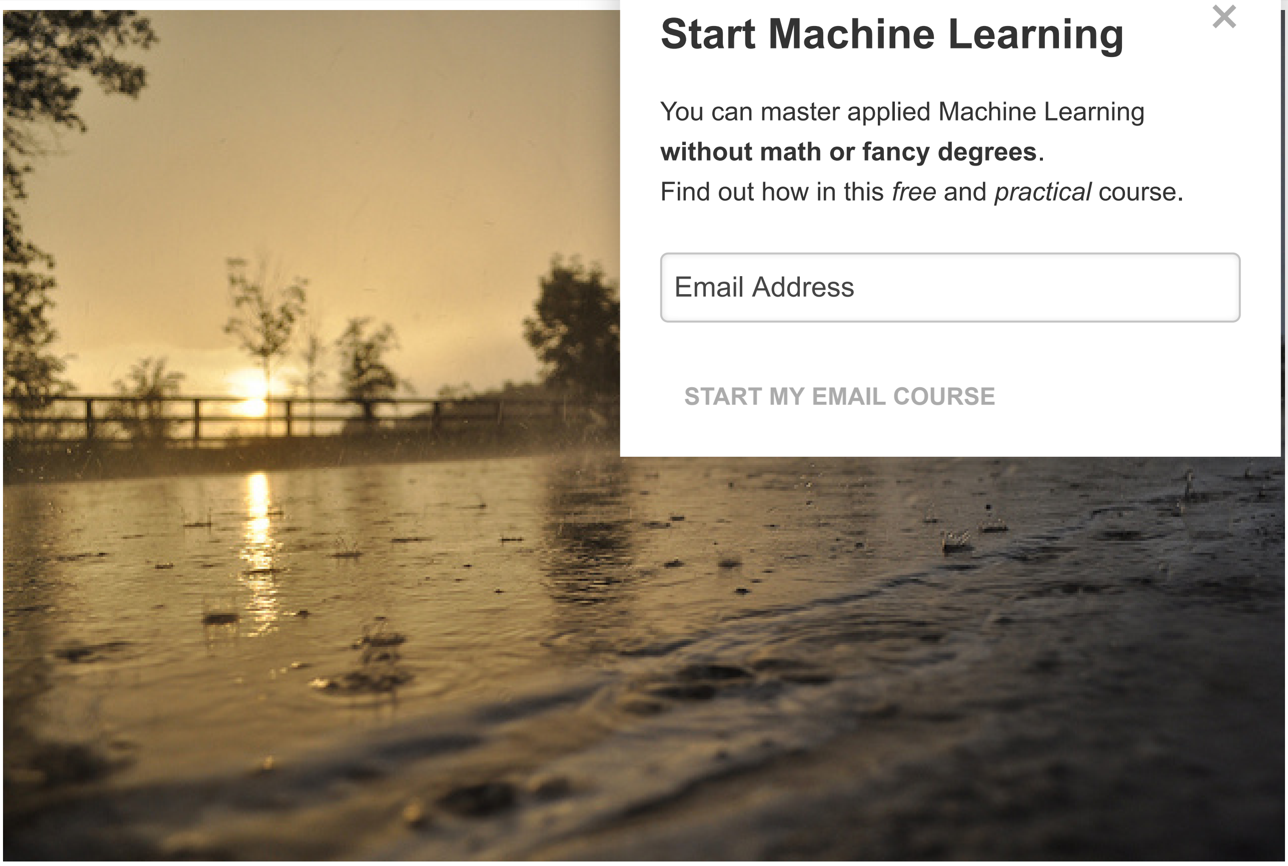
In some cases, the data can be corrected to meet the assumptions, such as correcting a nearly normal distribution to be normal by removing outliers, or using a correction to the degrees of freedom in a statistical test when samples have differing variance, to name two examples.

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.

Discover statistical hypothesis testing, resampling methods, estimation statistics and nonparametric methods in my new book, with 29 step-by-step tutorials and full source code.

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

1. Normality Tests

This section lists statistical tests that you can use t

Shapiro-Wilk Test

Tests whether a data sample has a Gaussian distri

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

```
# 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
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 Py
- 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

```
# 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 Py
- 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
- Observations in each sample are normally dis
- Observations in each sample have the same v

Interpretation

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

Python Code


```
# Example of the Pearson's Correlation tes
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

```
# 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 Correla
- scipy.stats.spearmanr
- Spearman's rank correlation coefficient on Wi

Kendall's Rank Correlation

Tests whether two samples have a monotonic rela

Assumptions

- Observations in each sample are independen
- Observations in each sample can be ranked.

Interpretation

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

Python Code

```
# 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

```
# 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
- 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

```
# 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

```
# Example of the Kwiatkowski-Phillips-Schmi
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

```
# 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 ar

Assumptions

- Observations in each sample are independen
- Observations in each sample are normally dis
- Observations in each sample have the same v
- 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

```
# 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

```
# Example of the Analysis of Variance Test
from scipy.stats import f_oneway
data1 = [0.873, 2.817, 0.121, -0.945, -0.0
data2 = [1.142, -0.432, -0.938, -0.729, -0
data3 = [-0.208, 0.696, 0.928, -1.148, -0.
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 distribution
```


More Information

- How to Calculate Parametric Statistical Hypot
- 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 equa
- H1: the distributions of both samples are not e

Python Code

```
# Example of the Mann-Whitney U Test
from scipy.stats import mannwhitneyu
data1 = [0.873, 2.817, 0.121, -0.945, -0.0
data2 = [1.142, -0.432, -0.938, -0.729, -0
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

```
# Example of the Wilcoxon Signed-Rank Test
from scipy.stats import wilcoxon
data1 = [0.873, 2.817, 0.121, -0.945, -0.0
```

```
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 inde

Assumptions

- Observations in each sample are independen
- Observations in each sample can be ranked.

Interpretation

- H0: the distributions of all samples are equal.
- H1: the distributions of one or more samples a

Python Code

```
# 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 distribution
```


More Information

- How to Calculate Nonparametric Statistical Hy
- 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.
View all posts by Jason Brownlee →

‹ How to Reduce Variance in a Final Machine Learning Model

A Gentle Introduction to SARIMA for Time Series Forecasting in Python ›

47 Responses to *17 Statistical Hypothesis Tests in Python (Cheat Sheet)*

Jonathan dunne August 17, 2018 at 7:17 am # REPLY

hi, the list looks good. a few omissions. fishers exact test and Bernards test (potentially more power than a fishers exact test)

one note on the anderson darling test. the use of p values to determine GoF has been discouraged in some fields .

Jason Brownlee August 17, 2018 at 7:43 am # REPLY

Excellent note, thanks Jonathan.

Indeed, I think it was a journal of psychology that has adopted “estimation statistics” instead of hypothesis tests in reporting results.

Hitesh August 17, 2018 at 3:19 pm # REPLY

Very Very Good and Useful Article

Jason Brownlee August 18, 2018 at 5:32

Thanks, I’m happy to hear that.

Barrie August 17, 2018 at 9:38 pm #

Hi, thanks for this nice overview.

Some of these tests, like friedmanchisquare, expec
same over time. But in practice this is not allways t

Lets say there are 4 observations on a group of 100 people, but the size of the response from this group changes over time with n1=100, n2=95, n3=98, n4=60 respondants.
n4 is smaller because some external factor like bad weather.
What would be your advice on how to tackle this different ‘respondants’ sizes over time?

Jason Brownlee August 18, 2018 at 5:36 am # REPLY

Good question.

Perhaps check the literature for corrections to the degrees of freedom for this situation?

Fredrik August 21, 2018 at 5:44 am # REPLY

Shouldn’t it say that Pearson correlation measures the linear relationship between variables? I would say that monotonic suggests, a not necessarily linear, “increasing” or “decreasing” relationship.

Jason Brownlee August 21, 2018 at 6:23 am # REPLY

Right, Pearson is a linear relationship, nonparametric methods like Spearmans are monotonic relationships.

Thanks, fixed.

Fredrik August 23, 2018 at 8:59 pm #

REPLY

No problem. Thank you for a great blog! It has introduced me to so many interesting and useful topics.

Jason Brownlee August 24, 2018 at 6:07 am #

REPLY

Happy to hear that!

Anthony The Koala August 22, 2018 at 2:47

Two points/questions on testing for norma
(1) In the Shapiro/Wilk, D'Agostino and Anderson/D
data is likely to be normally distributed? Or put it ar
indicate that the data may be gaussian?

(2) What about using graphical means such as a hi
normal plots https://www.itl.nist.gov/div898/handbo
then with the statistical tests described in (1), you can assess that the data may well come from a gaussian distribution.

Thank you,
Anthony of Sydney

Jason Brownlee August 22, 2018 at 6:15 am #

REPLY

More on what normality tests to use here (graphical and otherwise):
https://machinelearningmastery.com/a-gentle-introduction-to-normality-tests-in-python/

Tej Yadav August 26, 2018 at 4:07 pm #

REPLY

Wow.. this is what I was looking for. Ready made thing for ready reference.

Thanks for sharing Jason.

Jason Brownlee August 27, 2018 at 6:10 am #

REPLY

I'm happy it helps!

Nithin November 7, 2018 at 11:23 pm # REPLY

Thanks a lot, Jason! You're the best. I've been scouring the internet for a piece on practical implementation of Inferential statistics in Machine Learning for some time now!
Lots of articles with the same theory stuff going over and over again but none like this.

Jason Brownlee November 8, 2018 at 6:08 am # REPLY

Thanks, I'm glad it helped.

Nithin November 8, 2018 at 11:12 pm #

Hi Jason, Statsmodels is another how to go about it on the web. The docume scipy. Have you written anything on Statsm

Jason Brownlee November 9,

Yes, I have many tutorials showing how to use statsmodels for time series:
https://machinelearningmastery.com/start-here/#timeseries

and statsmodels for general statistics:
https://machinelearningmastery.com/start-here/#statistical_methods

Thomas March 29, 2019 at 10:02 pm # REPLY

Hey Jason, thank you for your awesome blog. Gave me some good introductions into unfamiliar topics!

If your seeking for completeness on easy appliable hypothesis tests like those, I suggest to add the Kolmogorov-Smirnov test which is not that different from the Shapiro-Wilk.

– https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ks_2samp.html
– https://www.researchgate.net/post/Whats_the_difference_between_Kolmogorov-Smirnov_test_and_Shapiro-Wilk_test

Jason Brownlee March 30, 2019 at 6:27 am # REPLY

Thanks for the suggestion Thomas.

Paresh April 16, 2019 at 5:17 pm # REPLY

Which methods fits for classification or regression data sets? Which statistical tests are good for Semi-supervised/ un-supervised data sets?

Jason Brownlee April 17, 2019 at 6:55 am # REPLY

This post will help:
https://machinelearningmastery.com/statistical-significance-tests-for-comparing-machine-learning-algorithms/

Luc May 1, 2019 at 10:01 pm #

Hello,
Thank you very much for your blog !

I'm wondering how to check that "observations in e
test to check that ?

Jason Brownlee May 2, 2019 at 8:03 am # REPLY

Great question.

You can calculate the mean and standard deviation for each interval.

You can also plot the series and visually look for increasing variance.

João Antônio Martins June 2, 2019 at 4:39 am # REPLY

Is there a test similar to the friedman test? which has the same characteristics "whether the distributions of two or more paired samples are equal or not".

Jason Brownlee June 2, 2019 at 6:45 am # REPLY

Yes, the paired student's t-test.

MIAO June 27, 2019 at 3:37 pm # REPLY

HI, Jason, Thank you for your nice blog. I have one question. I have two samples with different size (one is 102, the other is 2482), as well as the method is appropriate? Thank you.

Jason Brownlee June 28, 2019 at 5:57 am # REPLY

That is a very big difference.

The test depends on the nature of the question you're trying to answer.

Adrian Olszewski February 27, 2020 at 11:32 pm # REPLY

Practically ALL assumptions and ALL
recommend this, as if a student repeats that or
one's likely to fail it. I am messaged regularly b
problems after repeating texts from the interne
situations.

1. Assumptions of the paired t-test are totally w
2. Anova is not a test, but OK, let's pretend I di
that, you may be really surprised doing the pos
3. interpretation of the RM-ANOVA is wrong
4. Mann-Whitney is described imprecisely.
5. Paired Wilcoxon has wrong interpretation.
6. Normality tests – all is wrong. What "each sample" – in normality test? and it doesn't tell if it's Gaussian! It says of the data is approximated by the normal distribution acceptably well at this sample size. In a minute I can give you examples drawn from log-normal or Weibull reported as "Gaussian" .

It's worth noting there are over 270 tests, 50 in constant, everyday use, varying across industries and areas of specialization. Type "100 statistical tests PDF" into Google or find the handbook of parametric and non-parametric methods by Sheskin (also available in PDF), to get some rough idea about them. The more you know, the less you are limited. Each of those tests has its weaknesses and strengthens you should know before the use. Always pay attention to the null hypothesis and the assumptions. Jason Brownlee

Jason Brownlee February 28, 2020 at 6:09 am # REPLY

Thanks for your feedback Adrian.

Mr.T March 1, 2020 at 9:08 am # REPLY

You sir, are patronizing.

I am an early stage learner of all of this, and Jason's posts have been incredibly helpful in
helping me construct a semantic tree of all
knowledge pieces would be scattered.

I am not certain about the accuracy as you have pointed out, but your lack of constructiveness in your comment is concerning. You do not provide what you believe is the correct interpretation.

I truly hate to see a comment like this. Keep up the good work Jason!

Jason Brownlee March 2, 2020 at 6:10 am # REPLY

Thanks for your support!

MIAO June 28, 2019 at 5:50 pm #

Thank you. Jason. The problem I process
patient group and 2482 features for healthy group,
features of two groups to test if the feature is appro
which method is right for this case. Could you give

Jason Brownlee June 29, 2019 at 6:37 a

Sounds like you want a classification (

MIAO July 1, 2019 at 10:52 am # REPLY

Yeah, I think you are right. I will use SVM to classify the features. Thank you.

Veetee August 6, 2019 at 1:04 am # REPLY

Hi Jason, thanks for the very useful post. Is there a variant of Friedman's test for only two sets of measurements? I have an experiment in which two conditions were tested on the same people. I expect a semi-constant change between the two conditions, such that the ranks within blocks are expected to stay very similar.

Jason Brownlee August 6, 2019 at 6:40 am # REPLY

Yes: Wilcoxon Signed-Rank Test

wishy September 6, 2019 at 10:09 pm # REPLY

Dear Sir,

I have one question if we take subset of the huge data,and according to the Central limit theorem the 'samples averages follow normal distribution'.So in that case is it should we consider Nonparametric Statistical Hypothesis Tests or parametric Statistical Hypothesis Tests

Jason Brownlee September 7, 2019 at 5:29 am # REPLY

I don't follow your question sorry, please you can restate it?

Generally nonparametric stats use ranking instead of gaussians.

gopal jamnal September 28, 2019 at 10:43 pm

What is A-B testing, and how it can be use

testing?

Jason Brownlee September 29, 2019 at

More on a/b testing:
https://en.wikipedia.org/wiki/A/B_testing

It is not related to machine learning.

Instead, in machine learning, we will evaluate the performance of different machine learning algorithms, and compare the samples of performance estimates to see if the difference in performance between algorithms is significant or not.

Does that help?

More here:
https://machinelearningmastery.com/statistical-significance-tests-for-comparing-machine-learning-algorithms/

Peiran November 14, 2019 at 8:57 am # REPLY

You can't imagine how happy I am to find a cheat sheet like this! Thank you for the links too.

Jason Brownlee November 14, 2019 at 1:43 pm # REPLY

Thanks, I'm happy it helps!

Chris Winsor December 3, 2019 at 2:23 pm #

Hi Jason –

Thank you for helping to bring the theory of statistics to everyday application !

I'm wishing you had included an example of a t-test for equivalence. This is slightly different from the standard t-test and there are many applications – for example – demonstrating version 2.0 of the ml algorithm matches version 1.0. That is actually super important for customers that don't want to re-validate their instruments, or manufacturers that would need to answer why/if those versions perform the same as one-another.

I observe a library at
http://www.statsmodels.org/0.9.0/generated/statsmodels.stats.weightstats.ttost_paired.html#statsmodels.stats.weightstats.ttost_paired
but it doesn't explain how to establish reasonable low and high limits.

Anyway thank you for the examples !

Jason Brownlee December 4, 2019 at 5:

Great suggestion, thanks Chris!

makis January 29, 2020 at 4:58 am #

Hi Jason,

Great article.

If I want to compare the Gender across 2 groups, is chi-square test a good choice?
I want to test for signiicant differences similarly to a t-test for a numerical variable.

Jason Brownlee January 29, 2020 at 6:48 am # REPLY

It depends on the data, perhaps explore whether it is appropriate with a prototype?

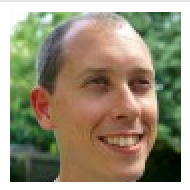
Leave a Reply

Name (required)

Email (will not be published) (required)

Website

SUBMIT COMMENT



Welcome!
My name is *Jason Brownlee* PhD, and I
Read more

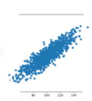
Never miss a tutorial:

Picked for you:

A Gentle Introduction to k-fold Cross-Validation

A Gentle Introduction to Normality Tests in Python

Statistical Significance Tests for Comparing Machine Learning Algorithms



How to Calculate Correlation Between Variables in Python

Statistics for Machine Learning (7-Day Mini-Course)

Loving the Tutorials?

The Statistics for Machine Learning EBook is where I keep the ***Really Good*** stuff.

Start Machine Learning