15 Statistical Hypothesis Tests in Python (Cheat Sheet)

by Jason Brownlee on August 15, 2018 in Statistical Methods

Quick-reference guide to the 15 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 o[...] freedom in a statistical test when samples have

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.

Let's get started.

- **Update Nov/2018**: Added a better overview of the tests covered.

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

Tutorial Overview

This tutorial is divided into four 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. **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
4. **Nonparametric Statistical Hypothesis Te**
    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

```
from scipy.stats import shapiro
data1 = ....
stat, p = shapiro(data)
```

More Information

- 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

```
from scipy.stats import normaltest
data1 = ....
stat, p = normaltest(data)
```

More Information

- 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

```
from scipy.stats import anderson
data1 = ....
result = anderson(data)
```

More Information

- 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

```
from scipy.stats import pearsonr
data1, data2 = ...
corr, p = pearsonr(data1, data2)
```

More Information

- 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

```
from scipy.stats import spearmanr
data1, data2 = ...
corr, p = spearmanr(data1, data2)
```

More Information

- 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

```
from scipy.stats import kendalltau
data1, data2 = ...
corr, p = kendalltau(data1, data2)
```

More Information

- 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

```
from scipy.stats import chi2_contingency
table = ...
stat, p, dof, expected = chi2_contingency(table)
```

More Information

- scipy.stats.chi2_contingency
- Chi-Squared test on Wikipedia

3. 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

```
from scipy.stats import ttest_ind
data1, data2 = ...
stat, p = ttest_ind(data1, data2)
```

More Information

- 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

```
from scipy.stats import ttest_rel
data1, data2 = ...
stat, p = ttest_rel(data1, data2)
```

More Information

- 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

```
from scipy.stats import f_oneway
data1, data2, ... = ...
stat, p = f_oneway(data1, data2, ...)
```

More Information

- 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

- Analysis of variance on Wikipedia

4. 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

```
from scipy.stats import mannwhitneyu
data1, data2 = ...
stat, p = mannwhitneyu(data1, data2)
```

More Information

- 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

```
from scipy.stats import wilcoxon
data1, data2 = ...
stat, p = wilcoxon(data1, data2)
```

More Information

- 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

```
from scipy.stats import kruskal
data1, data2, ... = ...
stat, p = kruskal(data1, data2, ...)
```

More Information

- 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

```
from scipy.stats import friedmanchisquare
data1, data2, ... = ...
stat, p = friedmanchisquare(data1, data2, ...)
```

More Information

- 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 ›

20 Responses to *15 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 am #

REPLY

Thanks, I'm happy to hear that.

Barrie August 17, 2018 at 9:38 pm #

REPLY

Hi, thanks for this nice overview.

Some of these tests, like friedmanchisquare, expect that the quantity of events is the group to remain the same over time. But in practice this is not allways the case.

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 am #

REPLY

Two points/questions on testing for normality of data:
(1) In the Shapiro/Wilk, D'Agostino and Anderson/Darling tests, do you use all three to be sure that your data is likely to be normally distributed? Or put it another way, what if only one or two of the three test indicate that the data may be gaussian?

(2) What about using graphical means such as a histogram of the data – is it symmetrical? What about normal plots https://www.itl.nist.gov/div898/handbook/eda/section3/normprpl.htm if the line is straight, 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 #

REPLY

Hi Jason, Statsmodels is another module that has got lots to offer but very little info on how to go about it on the web. The documentation is not as comprehensive either compared to scipy. Have you written anything on Statsmodels ? A similar article would be of great help.

Jason Brownlee November 9, 2018 at 5:22 am #

REPLY

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.

Leave a Reply

Name (required)

Email (will not be published) (required)

Website

SUBMIT COMMENT

Welcome to Machine Learning Mastery!

Hi, I'm **Jason Brownlee**, PhD.
I write tutorials to help developers (*like you*) get results with machine learning.

Read More

POPULAR

How to Develop LSTM Models for Multi-Step Time Series Forecasting of Household Power Consumption
OCTOBER 10, 2018

11 Classical Time Series Forecasting Methods in Python (Cheat Sheet)
AUGUST 6, 2018

How to Develop LSTM Models for Time Series Forecasting
NOVEMBER 14, 2018

A Gentle Introduction to LSTM Autoencoders
NOVEMBER 5, 2018

A Gentle Introduction to k-fold Cross-Validation
MAY 23, 2018

How to Develop RNN Models for Human Activity Recognition Time Series Classification
SEPTEMBER 24, 2018

When to Use MLP, CNN, and RNN Neural Networks
JULY 23, 2018

Top beginner tutorials:

- How to Install Python for Machine Learning
- Your First Machine Learning Project in Python
- Your First Neural Network in Python
- Your First Classifier in Weka
- Your First Time Series Forecasting Project

Top project tutorials:

- Photo Captioning Project
- Neural Machine Translation Project
- Sentiment Analysis Project
- Power Forecasting Project