How to Calculate Nonparametric Statistical Hypothesis Tests in Python

by **Jason Brownlee** on May 16, 2018 in **Statistics**

Last Updated on August 8, 2019

In applied machine learning, we often need to determine whether two data samples have the same or different distributions.

We can answer this question using statistical significance tests that can quantify the likelihood that the samples have the same distribution.

If the data does not have the familiar Gaussian distribution, we must resort to nonparametric version of the significance tests. These tests operate in a similar manner, but are distribution free, requiring that real valued data be first transformed into rank data before the test can be performed.

In this tutorial, you will discover nonparametric statistical tests that you can use to determine if data samples were drawn from populations with the same or different distributions.

After completing this tutorial, you will know:

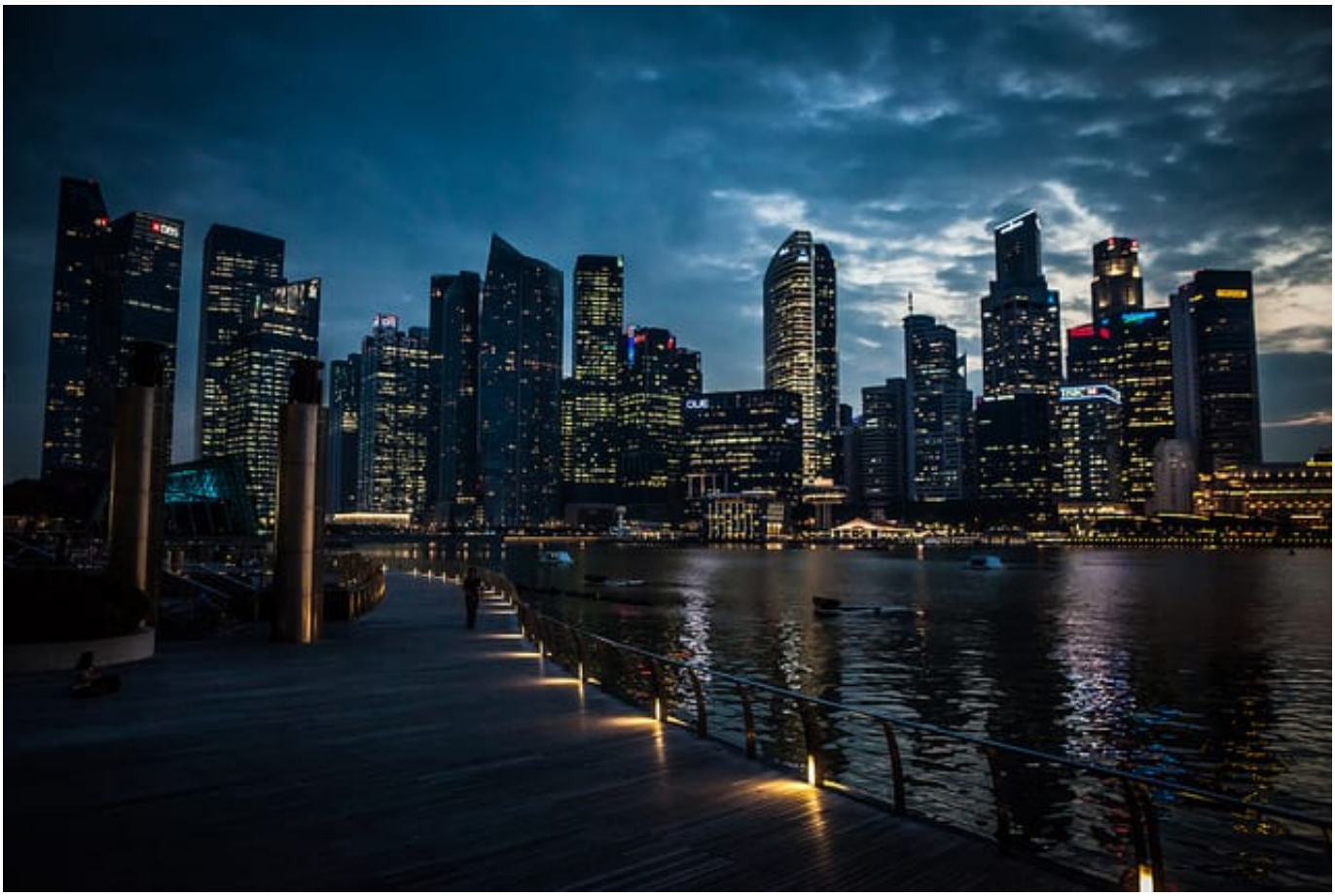
- The Mann-Whitney U test for comparing independent data samples: the nonparametric version of the Student t-test.
- The Wilcoxon signed-rank test for comparing paired data samples: the nonparametric version of the paired Student t-test.
- The Kruskal-Wallis H and Friedman tests for comparing more than two data samples: the

nonparametric version of the ANOVA and repeated measures ANOVA tests.

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.

- **Updated May/2018**: Improved language around reject vs fail to reject of statistical tests.



Introduction to Nonparametric Statistical Significance Tests in Python
Photo by Jirka Matousek, some rights reserved.

Tutorial Overview

This tutorial is divided into 6 parts; they are:

1. Nonparametric Statistical Significance Tests
2. Test Data
3. Mann-Whitney U Test
4. Wilcoxon Signed-Rank Test

5. Kruskal-Wallis H Test
6. Friedman Test

Nonparametric Statistical Significance Tests

Nonparametric statistics are those methods that do not assume a specific distribution to the data.

Often, they refer to statistical methods that do not assume a Gaussian distribution. They were developed for use with ordinal or interval data, but in practice can also be used with a ranking of real-valued observations in a data sample rather than on the observation values themselves.

A common question about two or more datasets is whether they are different. Specifically, whether the difference between their central tendency (e.g. mean or median) is statistically significant.

This question can be answered for data samples that do not have a Gaussian distribution by using nonparametric statistical significance tests. The null hypothesis of these tests is often the assumption that both samples were drawn from a population with the same distribution, and therefore the same population parameters, such as mean or median.

If after calculating the significance test on two or more samples the null hypothesis is rejected, it indicates that there is evidence to suggest that samples were drawn from different populations, and in turn the difference between sample estimates of population parameters, such as means or medians may be significant.

These tests are often used on samples of model skill scores in order to confirm that the difference in skill between machine learning models is significant.

In general, each test calculates a test statistic, that must be interpreted with some background in statistics and a deeper knowledge of the statistical test itself. Tests also return a p-value that can be used to interpret the result of the test. The p-value can be thought of as the probability of observing the two data samples given the base assumption (null hypothesis) that the two samples were drawn from a population with the same distribution.

The p-value can be interpreted in the context of a chosen significance level called alpha. A common value for alpha is 5% or 0.05. If the p-value is below the significance level, then the test says there is enough evidence to reject the null hypothesis and that the samples were likely drawn from populations with differing distributions.

- **p <= alpha**: reject H0, different distribution.
- **p > alpha**: fail to reject H0, same distribution.

Test Dataset

Before we look at specific nonparametric significance tests, let's first define a test dataset that we can use to demonstrate each test.

We will generate two samples drawn from different distributions. We will draw the samples from Gaussian distributions for simplicity, although, as noted, the tests we review in this tutorial are for data samples where we do not know or assume any specific distribution.

We will use the randn() NumPy function to generate a sample of 100 Gaussian random numbers in each sample with a mean of 0 and a standard deviation of 1. Observations in the first sample are scaled to have a mean of 50 and a standard deviation of 5. Observations in the second sample are scaled to have a mean of 51 and a standard deviation of 5.

We expect the statistical tests to discover that the samples were drawn from differing distributions, although the small sample size of 100 observations per sample will add some noise to this decision.

The complete code example is listed below.

```
# generate gaussian data samples
from numpy.random import seed
from numpy.random import randn
from numpy import mean
from numpy import std
# seed the random number generator
seed(1)
# generate two sets of univariate observations
data1 = 5 * randn(100) + 50
data2 = 5 * randn(100) + 51
```

```
# summarize
print('data1: mean=%.3f stdv=%.3f' % (mean(data1), std(data1)))
print('data2: mean=%.3f stdv=%.3f' % (mean(data2), std(data2)))
```

Running the example generates the data samples, then calculates and prints the mean and standard deviation for each sample, confirming their different distribution.

```
data1: mean=50.303 stdv=4.426
data2: mean=51.764 stdv=4.660
```

Mann-Whitney U Test

The Mann-Whitney U test is a nonparametric statistical significance test for determining whether two independent samples were drawn from a population with the same distribution.

The test was named for Henry Mann and Donald Whitney, although it is sometimes called the Wilcoxon-Mann-Whitney test, also named for Frank Wilcoxon, who also developed a variation of the test.

> *The two samples are combined and rank ordered together. The strategy is to determine if the values from the two samples are randomly mixed in the rank ordering or if they are clustered at opposite ends when combined. A random rank order would mean that the two samples are not different, while a cluster of one sample values would indicate a difference between them.*

— Page 58, Nonparametric Statistics for Non-Statisticians: A Step-by-Step Approach, 2009.

The default assumption or null hypothesis is that there is no difference between the distributions of the data samples. Rejection of this hypothesis suggests that there is likely some difference between the samples. More specifically, the test determines whether it is equally likely that any randomly selected observation from one sample will be greater or less than a sample in the other distribution. If violated, it suggests differing distributions.

- **Fail to Reject H0**: Sample distributions are equal.
- **Reject H0**: Sample distributions are not equal.

For the test to be effective, it requires at least 20 observations in each data sample.

We can implement the Mann-Whitney U test in Python using the mannwhitneyu() SciPy function. The functions takes as arguments the two data samples. It returns the test statistic and the p-value.

The example below demonstrates the Mann-Whitney U test on the test dataset.

```
# Mann-Whitney U test
from numpy.random import seed
from numpy.random import randn
from scipy.stats import mannwhitneyu
# seed the random number generator
seed(1)
# generate two independent samples
data1 = 5 * randn(100) + 50
data2 = 5 * randn(100) + 51
# compare samples
stat, p = mannwhitneyu(data1, data2)
print('Statistics=%.3f, p=%.3f' % (stat, p))
# interpret
alpha = 0.05
if p > alpha:
    print('Same distribution (fail to reject H0)')
else:
    print('Different distribution (reject H0)')
```

Running the example calculates the test on the datasets and prints the statistic and p-value.

The p-value strongly suggests that the sample distributions are different, as is expected.

```
Statistics=4025.000, p=0.009
Different distribution (reject H0)
```

Wilcoxon Signed-Rank Test

In some cases, the data samples may be paired.

There are many reasons why this may be the case, for example, the samples are related or matched in some way or represent two measurements of the same technique. More specifically, each sample is independent, but comes from the same population.

Examples of paired samples in machine learning might be the same algorithm evaluated on different datasets or different algorithms evaluated on exactly the same training and test data.

The samples are not independent, therefore the Mann-Whitney U test cannot be used. Instead, the Wilcoxon signed-rank test is used, also called the Wilcoxon T test, named for Frank Wilcoxon. It is the equivalent of the paired Student T-test, but for ranked data instead of real valued data with a Gaussian distribution.

> *The Wilcoxon signed ranks test is a nonparametric statistical procedure for comparing two samples that are paired, or related. The parametric equivalent to the Wilcoxon signed ranks*

> *test goes by names such as the Student's t-test, t-test for matched pairs, t-test for paired samples, or t-test for dependent samples.*

— Pages 38-39, Nonparametric Statistics for Non-Statisticians: A Step-by-Step Approach, 2009.

The default assumption for the test, the null hypothesis, is that the two samples have the same distribution.

- **Fail to Reject H0**: Sample distributions are equal.
- **Reject H0**: Sample distributions are not equal.

For the test to be effective, it requires at least 20 observations in each data sample.

The Wilcoxon signed-rank test can be implemented in Python using the wilcoxon() SciPy function. The function takes the two samples as arguments and returns the calculated statistic and p-value.

The complete example is below, demonstrating the calculation of the Wilcoxon signed-rank test on the test problem. The two samples are technically not paired, but we can pretend they are for the sake of demonstrating the calculation of this significance test.

```
# Wilcoxon signed-rank test
from numpy.random import seed
from numpy.random import randn
from scipy.stats import wilcoxon
# seed the random number generator
seed(1)
# generate two independent samples
data1 = 5 * randn(100) + 50
data2 = 5 * randn(100) + 51
# compare samples
stat, p = wilcoxon(data1, data2)
print('Statistics=%.3f, p=%.3f' % (stat, p))
# interpret
alpha = 0.05
if p > alpha:
    print('Same distribution (fail to reject H0)')
else:
    print('Different distribution (reject H0)')
```

Running the example calculates and prints the statistic and prints the result.

The p-value is interpreted strongly suggesting that the samples are drawn from different distributions.

```
Statistics=1886.000, p=0.028
Different distribution (reject H0)
```

Kruskal-Wallis H Test

When working with significance tests, such as Mann-Whitney U and the Wilcoxon signed-rank tests, comparisons between data samples must be performed pair-wise.

This can be inefficient if you have many data samples and you are only interested in whether two or more samples have a different distribution.

The Kruskal-Wallis test is a nonparametric version of the one-way analysis of variance test or ANOVA for short. It is named for the developers of the method, William Kruskal and Wilson Wallis. This test can be used to determine whether more than two independent samples have a different distribution. It can be thought of as the generalization of the Mann-Whitney U test.

The default assumption or the null hypothesis is that all data samples were drawn from the same distribution. Specifically, that the population medians of all groups are equal. A rejection of the null hypothesis indicates that there is enough evidence to suggest that one or more samples dominate another sample, but the test does not indicate which samples or by how much.

> *When the Kruskal-Wallis H-test leads to significant results, then at least one of the samples is different from the other samples. However, the test does not identify where the difference(s) occur. Moreover, it does not identify how many differences occur. To identify the particular differences between sample pairs, a researcher might use sample contrasts, or post hoc tests, to analyze the specific sample pairs for significant difference(s). The Mann-Whitney U-test is a useful method for performing sample contrasts between individual sample sets.*

— Page 100, Nonparametric Statistics for Non-Statisticians: A Step-by-Step Approach, 2009.

- **Fail to Reject H0**: All sample distributions are equal.
- **Reject H0**: One or more sample distributions are not equal.

Each data sample must be independent, have 5 or more observations, and the data samples can differ in size.

We can update the test problem to have 3 data samples, instead of 2, two of which have the same sample mean. Given that one sample differs, we would expect the test to discover the difference and reject the null hypothesis.

```
# generate three independent samples
data1 = 5 * randn(100) + 50
data2 = 5 * randn(100) + 50
```

```
data3 = 5 * randn(100) + 52
```

The Kruskal-Wallis H-test can be implemented in Python using the kruskal() SciPy function. It takes two or more data samples as arguments and returns the test statistic and p-value as the result.

The complete example is listed below.

```
# Kruskal-Wallis H-test
from numpy.random import seed
from numpy.random import randn
from scipy.stats import kruskal
# seed the random number generator
seed(1)
# generate three independent samples
data1 = 5 * randn(100) + 50
data2 = 5 * randn(100) + 50
data3 = 5 * randn(100) + 52
# compare samples
stat, p = kruskal(data1, data2, data3)
print('Statistics=%.3f, p=%.3f' % (stat, p))
# interpret
alpha = 0.05
if p > alpha:
    print('Same distributions (fail to reject H0)')
else:
    print('Different distributions (reject H0)')
```

Running the example calculates the test and prints the results.

The p-value is interpreted, correctly rejecting the null hypothesis that all samples have the same distribution.

```
Statistics=6.051, p=0.049
Different distributions (reject H0)
```

Friedman Test

As in the previous example, we may have more than two different samples and an interest in whether all samples have the same distribution or not.

If the samples are paired in some way, such as repeated measures, then the Kruskal-Wallis H test would not be appropriate. Instead, the Friedman test can be used, named for Milton Friedman.

The Friedman test is the nonparametric version of the repeated measures analysis of variance test, or repeated measures ANOVA. The test can be thought of as a generalization of the Kruskal-Wallis H Test to more than two samples.

The default assumption, or null hypothesis, is that the multiple paired samples have the same distribution. A rejection of the null hypothesis indicates that one or more of the paired samples has a different distribution.

- **Fail to Reject H0**: Paired sample distributions are equal.
- **Reject H0**: Paired sample distributions are not equal.

The test assumes two or more paired data samples with 10 or more samples per group.

> *The Friedman test is a nonparametric statistical procedure for comparing more than two samples that are related. The parametric equivalent to this test is the repeated measures analysis of variance (ANOVA). When the Friedman test leads to significant results, at least one of the samples is different from the other samples.*

— Pages 79-80, Nonparametric Statistics for Non-Statisticians: A Step-by-Step Approach, 2009.

We can implement the Friedman test in Python using the friedmanchisquare() SciPy function. This function takes as arguments the data samples to compare and returns the calculated statistic and p-value.

This significance test can be demonstrated on the same variation of the test dataset as was used in the previous section. Namely three samples, two with the same population mean and one with a slightly different mean. Although the samples are not paired, we expect the test to discover that not all of the samples have the same distribution.

The complete code example is listed below.

```
# Friedman test
from numpy.random import seed
from numpy.random import randn
from scipy.stats import friedmanchisquare
# seed the random number generator
seed(1)
# generate three independent samples
data1 = 5 * randn(100) + 50
data2 = 5 * randn(100) + 50
data3 = 5 * randn(100) + 52
# compare samples
stat, p = friedmanchisquare(data1, data2, data3)
print('Statistics=%.3f, p=%.3f' % (stat, p))
# interpret
alpha = 0.05
if p > alpha:
    print('Same distributions (fail to reject H0)')
else:
```

```
    print('Different distributions (reject H0)')
```

Running the example calculates the test on the three data samples and prints the test statistic and p-value.

The interpretation of the p-value correctly indicates that at least one sample has a different distribution.

```
Statistics=9.360, p=0.009
Different distributions (reject H0)
```

Extensions

This section lists some ideas for extending the tutorial that you may wish to explore.

- Update all examples to operate on data samples that have the same distribution.
- Create a flowchart for choosing each of the statistical significance tests given the requirements and behavior of each test.
- Consider 3 cases of comparing data samples in a machine learning project, assume a non-Gaussian distribution for the samples, and suggest the type of test that could be used in each case.

If you explore any of these extensions, I'd love to know.

Further Reading

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

Books

- Nonparametric Statistics for Non-Statisticians: A Step-by-Step Approach, 2009.

API

- numpy.random.seed() API
- numpy.random.randn() API
- scipy.stats.mannwhitneyu() API
- scipy.stats.wilcoxon() API
- scipy.stats.kruskal() API
- scipy.stats.friedmanchisquare() API

Articles

- Nonparametric statistics on Wikipedia
- Paired difference test on Wikipedia
- Mann-Whitney U test on Wikipedia
- Wilcoxon signed-rank test on Wikipedia
- Kruskal-Wallis one-way analysis of variance on Wikipedia
- Friedman test on Wikipedia

Summary

In this tutorial, you discovered nonparametric statistical tests that you can use to determine if data samples were drawn from populations with the same or different distributions.

Specifically, you learned:

- The Mann-Whitney U test for comparing independent data samples: the nonparametric version of the Student t-test.
- The Wilcoxon signed-rank test for comparing paired data samples: the nonparametric version of the paired Student t-test.
- The Kruskal-Wallis H and Friedman tests for comparing more than two data samples: the nonparametric version of the ANOVA and repeated measures ANOVA tests.

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

More On This Topic

A Gentle Introduction to Statistical Hypothesis Testing

17 Statistical Hypothesis Tests in Python (Cheat Sheet)

What is a Hypothesis in Machine Learning?

Statistical Significance Tests for Comparing Machine…

A Gentle Introduction to Nonparametric Statistics

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

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 →

‹ A Gentle Introduction to Statistical Hypothesis Testing

How to Calculate Parametric Statistical Hypothesis Tests in Python ›

55 Responses to *How to Calculate Nonparametric Statistical Hypothesis Tests in Python*

Anirban May 17, 2018 at 4:36 pm #

REPLY

Thanks for this great article, one query, how do we determine if a sample is following Normal Distribution. In each of these we are comparing two distributions, but how to know if any one of them is Normal dist pls.

Jason Brownlee May 18, 2018 at 6:20 am #

REPLY

Great question, here are a list of methods:
https://machinelearningmastery.com/a-gentle-introduction-to-normality-tests-in-python/

Hawthorne May 20, 2018 at 10:25 am #

REPLY

I stopped reading after “Accept H0”.

Jason Brownlee May 21, 2018 at 6:24 am #

REPLY

Thanks, I’ll rework the language.

Yoav July 14, 2018 at 12:49 am #

REPLY

Thanks for the great article. For right skewed data(like ARPU), would you recommend running u test on the data or running a lot of sampling simulations and running t test on the standard error of the mean (which will have normal distribution). The method described in detail here: http://blog.analytics-toolkit.com/2017/statistical-significance-non-binomial-metrics-revenue-time-site-pages-session-aov-rpu/

Jason Brownlee July 14, 2018 at 6:19 am #

REPLY

Perhaps try it and see?

Georgi Georgiev January 18, 2019 at 12:38 am #

REPLY

Hi Yoav,

The simulations in the blog post you reference were simply to illustrate the point of the Central Limit Theorem (proven a long time ago) which holds very well even for relatively small sample sizes (30-40 observations per group) according to sims I've done. You can apply a t-test to each particular case without any simulations whatsoever.

Furthermore, a non-parametric test like the Mahn-W Rank test will only evaluate the same thing as a t-test (difference in mean or median) only when the t-test assumptions hold, otherwise it is a test for stochastic difference and it harder to interpret and communicate.

Best,
Georgi

Ardeshir June 23, 2019 at 12:22 am #

REPLY

Hi Jason, thanks for the enlightening post. I was thinking whether I can apply the Wilcoxon test for two sets of data, each is the measurement of the same variable, but one is through an instrument and the other is through a model. They are time series data and paired hourly.

Jason Brownlee June 23, 2019 at 5:36 am #

REPLY

The possible auto correlation across time intervals might make it a problem.

Perhaps check with a statistician/

AkT September 14, 2018 at 9:29 pm #

REPLY

You great helper, You!

Man, I don't know how you do it – and just create all these pertinent topics that address my multifaceted data science needs. But thank you Jason Brownlee. Thank you!

REPLY

Jason Brownlee September 15, 2018 at 6:07 am #

Thanks, I'm happy they help!

REPLY

Marius October 25, 2018 at 12:02 am #

Interesting article!
I am running a comparison of two different machine learning methods on the same dataset for a number of 30 seeds. So I get 30 AUC for each of the methods.
Is the The Wilcoxon signed-rank test a good method to compare these two methods?

REPLY

Jason Brownlee October 25, 2018 at 7:57 am #

Hmmm, maybe a Student's t test:
https://machinelearningmastery.com/statistical-significance-tests-for-comparing-machine-learning-algorithms/

If its critical to the project, check in with a statistician.

REPLY

Manuel November 26, 2018 at 8:48 am #

Why do you using only p-value? The stat value does,nt matter? There is no t-table like for comparing?

REPLY

Jason Brownlee November 26, 2018 at 2:00 pm #

The p-value is an interpretation of the critical value (stat value).

Manuel Gonçalves November 29, 2018 at 2:03 am #

REPLY

After carefully documentation reading, I use r2py and port the stats.wilcox.test() from R to Python due to approximation errors and normal approximations. The scipy version only returns the two-sided values and does not implement the less and greater alternative hypothesis. There are also problems with the small number of tests (less than 20) that is the 10-fold cross-validation case.

Jason Brownlee November 29, 2018 at 7:43 am #

REPLY

Thanks for sharing.

Pietro February 7, 2019 at 1:22 am #

REPLY

I find the following extremely misleading:

Fail to Reject H0: Sample distributions are equal.

These tests never conclude that the distributions are equal, but simply that we cannot say for sure that they are not equal. Failure to reject could be motivated simply by low power, which is to say: your distributions may be different, but with the amount of data you have, we cannot be sure.

Jason Brownlee February 7, 2019 at 6:41 am #

REPLY

Yes, you are correct. I was aiming to provide a simplified interpretation for common usage.

Instead, we could say “sample distributions are likely drawn from the same population”.

Raz March 7, 2019 at 10:03 pm #

REPLY

Hi Jason,

Great tutorials. I have few questions as i don’t have a statistics background:

1. I am using classifiers performance scores (such as accuracy) as sample distribution from each of 10-folds (CV test scores). Am i doing it correctly?

2. Is a non-parametric test most suitable in my case?

Jason Brownlee March 8, 2019 at 7:49 am #

REPLY

Great question!

I recommend reading this post on this exact topic:
https://machinelearningmastery.com/statistical-significance-tests-for-comparing-machine-learning-algorithms/

Davide July 2, 2019 at 6:28 am #

REPLY

Hi Jason,

thanks for the amazing tutorial.
I was wondering how to do a post-hoc Bonferroni correction (after performing Kruskal-Wallis test) in order to get which groups are statistically different.
I already did it in Matlab using the following function:
[results,means] = multcompare(stats,'CType','bonferroni')

where 'stats' is the value taken from Kruskal-Wallis test.

Thank you in advance.

Jason Brownlee July 2, 2019 at 7:40 am #

REPLY

Not sure off hand, sorry. Perhaps check the API to see if it is offered, or perhaps try implement it yourself in Python?
https://en.wikipedia.org/wiki/Bonferroni_correction

Magnus February 29, 2020 at 12:38 am #

REPLY

Hi Jason,

Thank you for a nice post. Are there any algorithms that can divide an existing (e.g. skewed) distribution into 2-3 distributions that use e.g. the Mann-Whitney U Test to verify that they are from the same

distribution?
I am thinking e.g. when dividing data sets for training neural networks, when we want the training, validation and test sets to be similar.

Jason Brownlee February 29, 2020 at 7:15 am #

REPLY

I don't know about dividing a sample and then using a test on it. Why not use the test on it directly.

The Dude April 19, 2020 at 6:19 am #

REPLY

Thanks for the super clear explanation,
Just a small comment, I think the test is actually used for small samples (<20) when you have a larger # of points, it just behaves like normal dist with known parameters (see wikipedia for details), so I think you should perhaps remove that requirement from this wonderful text.
cheers!

Jason Brownlee April 19, 2020 at 6:46 am #

REPLY

Thanks for sharing!

Himanshu May 27, 2020 at 5:10 am #

REPLY

Hi Jason,
A very informative article, helped me learn a lot!
Is there a way to calculate the confidence interval for the Wilcoxon signed-rank test in python? I could see an easy R implementation but nothing in python. Any pointers?

Thanks

Jason Brownlee May 27, 2020 at 8:03 am #

REPLY

Thanks!

Sorry, I don't have an example.

Himanshu May 31, 2020 at 2:16 am #

REPLY

No worries Jason. It led me to dig deeper to look for a solution and eventually I found one.

Since nothing out of the box was available in python, I followed the research and created a small function. I decided to write an article around the solution so that others could use it if required and I have mentioned you in credits as this post was one of reasons I started thinking about it.

Your expert opinions on the solution would be great to have, in case you get the chance to have a look at it, anytime in future : https://towardsdatascience.com/prepare-dinner-save-the-day-by-calculating-confidence-interval-of-non-parametric-statistical-29d031d079d0

Thanks

Jason Brownlee May 31, 2020 at 6:29 am #

REPLY

Well done!

jessie June 30, 2020 at 4:43 pm #

REPLY

While i contain the categorical data, would i convert it to numerical data and use Mann-Whitney U Test ?

Jason Brownlee July 1, 2020 at 5:51 am #

REPLY

That does not sound appropriate, consider a chi squared test for categorical data: https://machinelearningmastery.com/chi-squared-test-for-machine-learning/

jedi July 25, 2020 at 10:07 am #

REPLY

What is it that we have to do if we are having a pair of samples(A and B) with two dimensional data('x' rows and 'y' columns)?
I am trying to train an ML model with the sample A and then test the model with a different sample B. I am trying to understand how statistically different are the two samples and see if model is actually good or it needs re-training.

Jason Brownlee July 26, 2020 at 6:11 am #

REPLY

Good question, and tough to answer.

One approach might be to look at each pair of variables and see if they differ between the samples, or each pair of input variable with target and if they differ between samples. The latter sounds appropriate.

Mahmoud Sabri August 24, 2020 at 9:09 pm #

REPLY

Hello,

Thanks a lot for your informative blog!
Can I use Python wilcoxon for "unequal N" ?

Regards,

– Mahmoud Sabri

Jason Brownlee August 25, 2020 at 6:41 am #

REPLY

Thanks.

Wilcoxon is a paired test. Off the cuff I would say no. Perhaps you can dig into the literature to confirm.

Ron B. September 9, 2020 at 7:36 am #

REPLY

Hi Jason, thanks as always for the great blog posts! Do you have any guidance on how to calculate an effect size for these tests (especially Kruskal-Wallis) in python? Thanks!

Jason Brownlee September 9, 2020 at 8:10 am #

REPLY

This might help as a first step:
https://machinelearningmastery.com/effect-size-measures-in-python/

Raz September 25, 2020 at 4:01 am #

REPLY

Hi Jason, thank you very much for the great tutorial. I need your guidance regarding a small confusion. I am testing the performance of 5 different classifiers on 10 datasets (by increasing the size). I have a table of accuracy values for classifiers (rows) and datasets (columns). I want to use the Friedman test to check if there is significant difference among the performance of the classifiers. Should i pass the rows (5 samples) or columns (10 samples) to the friedmanchisquare function?

Jason Brownlee September 25, 2020 at 6:39 am #

REPLY

The sample of scores for each algorithm on one dataset would be provided to the function.

Jo December 18, 2020 at 11:21 pm #

REPLY

Hello,
Thanks for this great overview. I am performing a Friedman Test.
I get a p-value of 0.000 and ‘Different distribution (reject H0)’
I would like to know, if I can access the complete p-value.

Thanks for your help.

Jason Brownlee December 19, 2020 at 6:18 am #

REPLY

Yes, it is returned from the function directly.

Michelle February 2, 2021 at 8:48 am #

REPLY

Hi Jason, thanks for the article, if there are two paired non-parametric samples, with different sizes, which test could be applied to compare the mean please? it is exactly the same case as the wilcoxon signed rank test, but wilcoxon signed rank test can be only applied to compare two samples datasets with the same size.

Jason Brownlee February 2, 2021 at 1:18 pm #

REPLY

If the samples are not paired, consider the Mann-Whitney U.

Michelle February 2, 2021 at 9:43 pm #

REPLY

Hi Jason, thanks, that is the problem, because the samples are paired, dependent. I also don't find a test, that can be applied to such a scenario… Thank you anyway.

Jason Brownlee February 3, 2021 at 6:19 am #

REPLY

If the samples are paired, you will have the same number in each sample and can use "Wilcoxon Signed-Rank Test".

Jo February 11, 2021 at 7:10 pm #

REPLY

I understand that mannwhitneyu takes in large sample sizes only. Hence, does it already use a normal distribution to approximate the test statistic?

Jason Brownlee February 12, 2021 at 5:45 am #

REPLY

No it does not, it is a rank based method.

Josh Brown February 22, 2021 at 5:06 am #

REPLY

Great article! Question on interpreting the data. Let's say you run a Wilcoxon signed-rank test for paired data and the Null Hypothesis of the distributions A and B being from the same population is rejected. Is it valid to say that since the distributions A&B are different (per the test), that the distribution with the lower mean (say A) is part of a population that has an overall lower mean than the population distribution B belongs to? If that is not valid, how would you set up a test to validate that?

Jason Brownlee February 22, 2021 at 7:37 am #

REPLY

Thanks!

If the null is reject, the samples are probably from different populations, e.g. have different distributions.

We can't say much about the mean of the samples/distributions, as they are rank tests.

curious April 5, 2021 at 12:46 pm #

REPLY

Always liked your articles and have been following your site for years and I know many people do.

Just a question – this condition seems right but I am not sure why it is not generating the correct answer.

if p > alpha:
print('Same distributions (fail to reject H0)')
else:
print('Different distributions (reject H0)')

For example – the p-value is definitely < than 0.05, then this means we cannot reject Ho. But the result will print reject H0.

Statistics=9.360, p=0.009
Different distributions (reject H0)

Jason Brownlee April 6, 2021 at 5:15 am #

REPLY

Thanks!

Perhaps there is a bug/typo in your code, confirm your p and alpha values prior to the condition.

Afek October 26, 2021 at 1:28 am #

REPLY

Thanks, what can we do when we are dealing with two extremely skewed distributions (let's say it's an A/B test) and more than 95 of the observations in each group is zero and the other are right tailed (like salaries or purchase amount. Here, when I simulate results it's really rare to reject the null hypothesis when the means of each sample are different but not so far from each other (Man-Whitney test)

Adrian Tam October 27, 2021 at 2:32 am #

REPLY

I am not sure this is correct but when you say extremely skewed, it sounds to me that it is not Gaussian. Most statistics would like Gaussian, or at least approximate to it. So one idea might be to transform the data before applying the test. Take log, for example, is a common trick.

Leave a Reply

Name (required)

Email (will not be published) (required)

SUBMIT COMMENT

Welcome!
I'm *Jason Brownlee* PhD
and I **help developers** get results with **machine learning**.
Read more

Never miss a tutorial:

Picked for you:


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


A Gentle Introduction to k-fold Cross-Validation


How to Calculate Bootstrap Confidence Intervals For Machine Learning Results in Python


Statistical Significance Tests for Comparing Machine Learning Algorithms


A Gentle Introduction to Normality Tests in Python

Loving the Tutorials?

The Statistics for Machine Learning EBook is
where you'll find the ***Really Good*** stuff.