**Credits : Google/ Analytics Vidhya/ Machine learning mastery blog/ several other courses+blogs + Udacity + Udemy**

**IID : independent and identically distributed**

**Coin Tosses**

**1 coin -> 10 toss**

**PS : Coin gets damaged after each toss**

**Picking a blue ball from a pool of 5 blue and 3 black balls : ⅝ ,**

**Bl Bl Bl Bl Bl Bck Blck Blck**

**If the first ball was blue - 4B +3 Bck->Prob of blue - 4/7**

**If the first ball was black- 5B +2 Bck->Prob of blue- 5/7**

**⅝**

**If the first ball is blue - Bl Bl Bl Bl Bck Blck Blck - 4/7**

**If the first ball is black - Bl Bl Bl Bl Bl Blck Blck - 5/7**

**If the first ball was blue -> 4/7**

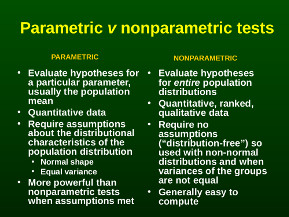
**If the first ball is not blue-> 5/7**

**With replacement (⅝) - Independent**

**Without replacement - Non Independent**

**A collection of random variables is independent and identically distributed if each random variable has the same probability distribution as the others and all are mutually independent.**

**Parametric vs Non Parametric**

****

**Normality Tests**

* 1. Shapiro-Wilk Test
  2. D’Agostino’s K^2 Test
  3. Anderson-Darling Test

1. **Correlation Tests**
   1. Pearson’s Correlation Coefficient
   2. Spearman’s Rank Correlation
   3. Kendall’s Rank Correlation
   4. Chi-Squared Test
2. **Stationary Tests**
   1. Augmented Dickey-Fuller
   2. Kwiatkowski-Phillips-Schmidt-Shin
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 Tests**
   1. Mann-Whitney U Test
   2. Wilcoxon Signed-Rank Test
   3. Kruskal-Wallis H Test
   4. Friedman Test

**What is hypothesis testing ?**

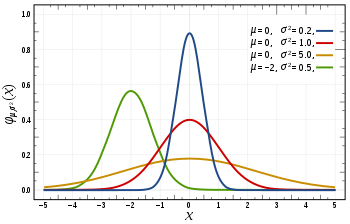
* Statistical method that is used in making statistical decisions using experimental data.
* an assumption that we make about the population parameter.

**Why do we use it ?**

essential procedure in statistics.

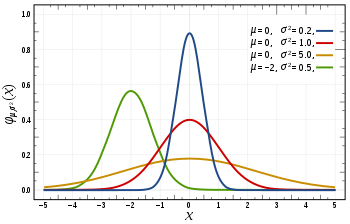
A **hypothesis test** evaluates two mutually exclusive statements about a population to determine which statement is best supported by the sample data. When **we** say that a finding is statistically significant, it’s thanks to a **hypothesis test**.

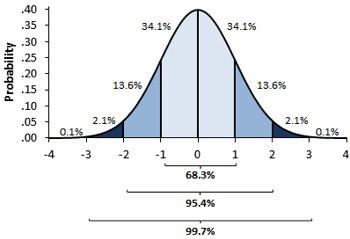
**3**. **what are basic of hypothesis ?**

****

Normal Curve images with different mean and variance

[normalisation](https://en.wikipedia.org/wiki/Normalization_(statistics)) & [standard normalisation](https://stats.stackexchange.com/questions/10289/whats-the-difference-between-normalization-and-standardization). all our hypothesis is revolve around basic of these 2 terms. let’s see these.



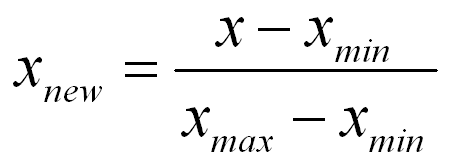


Standardised Normal curve image and separation on data in percentage in each section.

1st first you can see there are different normal curve all those normal curve can have different mean’s and variances where as in 2nd image if you notice the graph is properly distributed and **mean =0 and variance =1 always**. concept of z-score comes in picture when we use **standardised normal data.**

**Normal Distribution -**

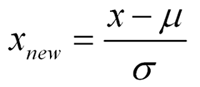
A variable is said to be normally distributed or have a **normal distribution** if **its distribution** has the shape of a **normal curve** — a special bell-shaped **curve**. … The graph of a **normal distribution** is called the **normal curve**, which has all of the following **properties**: 1. The mean, median, and mode are equal.



Normal distribution formula

**Standardised Normal Distribution —**

A standard normal distribution is a normal distribution with mean 0 and standard deviation 1



Standard Normal Distribution

**Which are important parameter of hypothesis testing ?**

**Null hypothesis :-** In inferential statistics, the null hypothesis is a general statement or default position that there is no relationship between two measured phenomena, or no association among groups

In other words it is a basic assumption or made based on domain or problem knowledge.

Example : a company production is = 50 unit/per day etc.

Alternative hypothesis :-

The alternativehypothesis is the hypothesis used in **hypothesis** testing that is contrary to the null hypothesis. It is usually taken to be that the observations are the result of a real effect (with some amount of chance variation superposed)Example : a company production is !=50 unit/per day etc.

**Level of significance:** Refers to the degree of significance in which we accept or reject the null-hypothesis. 100% accuracy is not possible for accepting or rejecting a hypothesis, so we therefore select a level of significance that is usually 5%.

This is normally denoted with alpha(maths symbol ) and generally it is 0.05 or 5% , which means your output should be 95% confident to give similar kind of result in each sample.

**Type I error:** When we reject the null hypothesis, although that hypothesis was true. Type I error is denoted by alpha. In hypothesis testing, the normal curve that shows the critical region is called the alpha region

**Type II errors:** When we accept the null hypothesis but it is false. Type II errors are denoted by beta. In Hypothesis testing, the normal curve that shows the acceptance region is called the beta region.

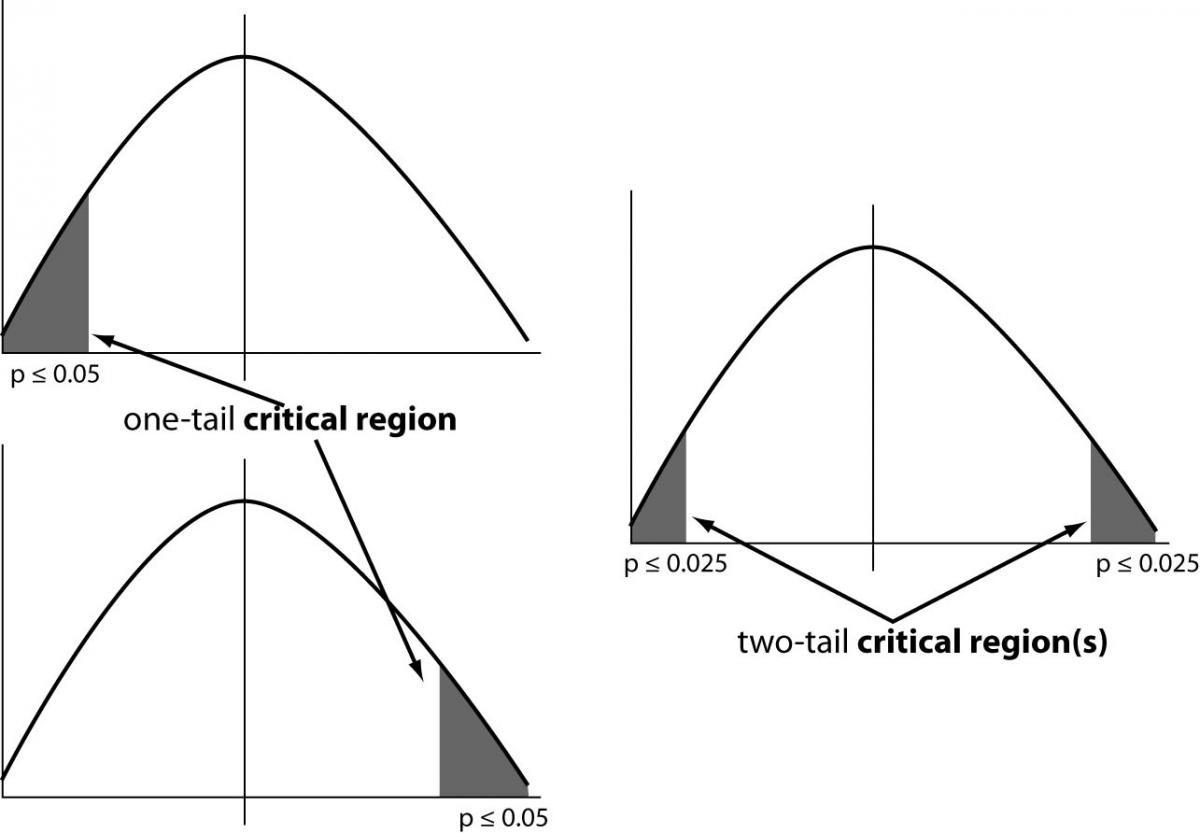
| **Prediction yes, Reality Yes** | **Prediction : No, Reality : Yes -> False Negative** |
| --- | --- |
| **Prediction : Yes , Reality : No -> False Postives** | **Prediction No, Reality No** |

**One tailed test :-** A test of a statistical hypothesis , where the region of rejection is on only **one** side of the sampling distribution , is called a **one**-**tailed test**.

Example :- a college has ≥ 4000 student or data science ≤ 80% org adopted.

**Two-tailed test :-** A **two**-**tailed test** is a statistical **test** in which the critical area of a distribution is **two**-**sided** and testswhether a sample is greater than or less than a certain range of values. If the sample being testedfalls into either of the critical areas, the alternative hypothesis is accepted instead of the null hypothesis.

Example : a college = 4000 student or data science = 80% org adopted



one and two-tailed images

**P-value :-** The **P value**, or calculated probability, is the probability of finding the observed, or more extreme, results when the null hypothesis (H 0) of a study question is true — the **definition** of ‘extreme’ depends on how the hypothesis is being tested.

If your P value is less than the chosen significance level then you reject the null hypothesis i.e. accept that your sample gives reasonable evidence to support the alternative hypothesis. It does NOT imply a “meaningful” or “important” difference; that is for you to decide when considering the real-world relevance of your result.

Example : you have a coin and you don’t know whether that is fair or tricky so let’s decide **null** and **alternate hypothesis**

Tests

## **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

| 1  2  3  4  5  6  7  8  9 | # 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') |
| --- | --- |

### **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

| 1  2  3  4  5  6  7  8  9 | # 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](https://machinelearningmastery.com/a-gentle-introduction-to-normality-tests-in-python/)
* [scipy.stats.normaltest](https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.normaltest.html)
* [D’Agostino’s K-squared test on Wikipedia](https://en.wikipedia.org/wiki/D%27Agostino%27s_K-squared_test)

### **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

| 1  2  3  4  5  6  7  8  9  10  11 | # 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](https://machinelearningmastery.com/a-gentle-introduction-to-normality-tests-in-python/)
* [scipy.stats.anderson](https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.anderson.html)
* [Anderson-Darling test on Wikipedia](https://en.wikipedia.org/wiki/Anderson%E2%80%93Darling_test)

## **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. (> 0.7 , positively correlated, <-0.7, negatively correlated, [-0.7 : 0.7] - not strongly correlated

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

| 1  2  3  4  5  6  7  8  9  10 | # 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](https://machinelearningmastery.com/how-to-use-correlation-to-understand-the-relationship-between-variables/)
* [scipy.stats.pearsonr](https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.pearsonr.html)
* [Pearson’s correlation coefficient on Wikipedia](https://en.wikipedia.org/wiki/Pearson_correlation_coefficient)

### **Spearman’s Rank Correlation**

a = [1,2,3,4,5,6]

b = [1,4,9,16,25,36]

a and b have not so high pearson’s correlation coefficient but a high spearman correlation coefficient

Pearson’s correlation coefficient between Asq and b= 1

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

| 1  2  3  4  5  6  7  8  9  10 | # 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](https://machinelearningmastery.com/how-to-calculate-nonparametric-rank-correlation-in-python/)
* [scipy.stats.spearmanr](https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.spearmanr.html)
* [Spearman’s rank correlation coefficient on Wikipedia](https://en.wikipedia.org/wiki/Spearman%27s_rank_correlation_coefficient)

### **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

| 1  2  3  4  5  6  7  8  9  10 | # 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](https://machinelearningmastery.com/how-to-calculate-nonparametric-rank-correlation-in-python/)
* [scipy.stats.kendalltau](https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.kendalltau.html)
* [Kendall rank correlation coefficient on Wikipedia](https://en.wikipedia.org/wiki/Kendall_rank_correlation_coefficient)

### **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

| 1  2  3  4  5  6  7  8  9 | # 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](https://machinelearningmastery.com/chi-squared-test-for-machine-learning/)
* [scipy.stats.chi2\_contingency](https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.chi2_contingency.html)
* [Chi-Squared test on Wikipedia](https://en.wikipedia.org/wiki/Chi-squared_test)

## **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

| 1  2  3  4  5  6  7  8  9 | # 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](https://machinelearningmastery.com/time-series-data-stationary-python/)
* [statsmodels.tsa.stattools.adfuller API](https://www.statsmodels.org/dev/generated/statsmodels.tsa.stattools.adfuller.html).
* [Augmented Dickey–Fuller test, Wikipedia](https://en.wikipedia.org/wiki/Augmented_Dickey%E2%80%93Fuller_test).

### **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 trend-stationary.
* H1: the time series is not trend-stationary.

Python Code

| 1  2  3  4  5  6  7  8  9 | # 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 Stationary')  else:  print('Probably not Stationary') |
| --- | --- |

More Information

* [statsmodels.tsa.stattools.kpss API](https://www.statsmodels.org/stable/generated/statsmodels.tsa.stattools.kpss.html#statsmodels.tsa.stattools.kpss).
* [KPSS test, Wikipedia](https://en.wikipedia.org/wiki/KPSS_test).

## **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

| 1  2  3  4  5  6  7  8  9  10 | # 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](https://machinelearningmastery.com/parametric-statistical-significance-tests-in-python/)
* [scipy.stats.ttest\_ind](https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_ind.html)
* [Student’s t-test on Wikipedia](https://en.wikipedia.org/wiki/Student%27s_t-test)

### **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

| 1  2  3  4  5  6  7  8  9  10 | # 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](https://machinelearningmastery.com/parametric-statistical-significance-tests-in-python/)
* [scipy.stats.ttest\_rel](https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_rel.html)
* [Student’s t-test on Wikipedia](https://en.wikipedia.org/wiki/Student%27s_t-test)

### **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

| 1  2  3  4  5  6  7  8  9  10  11 | # 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](https://machinelearningmastery.com/parametric-statistical-significance-tests-in-python/)
* [scipy.stats.f\_oneway](https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.f_oneway.html)
* [Analysis of variance on Wikipedia](https://en.wikipedia.org/wiki/Analysis_of_variance)

### **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](https://machinelearningmastery.com/parametric-statistical-significance-tests-in-python/)
* [Analysis of variance on Wikipedia](https://en.wikipedia.org/wiki/Analysis_of_variance)

## **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

| 1  2  3  4  5  6  7  8  9  10 | # 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](https://machinelearningmastery.com/nonparametric-statistical-significance-tests-in-python/)
* [scipy.stats.mannwhitneyu](https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.mannwhitneyu.html)
* [Mann-Whitney U test on Wikipedia](https://en.wikipedia.org/wiki/Mann%E2%80%93Whitney_U_test)

### **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

| 1  2  3  4  5  6  7  8  9  10 | # 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](https://machinelearningmastery.com/nonparametric-statistical-significance-tests-in-python/)
* [scipy.stats.wilcoxon](https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.wilcoxon.html)
* [Wilcoxon signed-rank test on Wikipedia](https://en.wikipedia.org/wiki/Wilcoxon_signed-rank_test)

### **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

| 1  2  3  4  5  6  7  8  9  10 | # 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](https://machinelearningmastery.com/nonparametric-statistical-significance-tests-in-python/)
* [scipy.stats.kruskal](https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.kruskal.html)
* [Kruskal-Wallis one-way analysis of variance on Wikipedia](https://en.wikipedia.org/wiki/Kruskal%E2%80%93Wallis_one-way_analysis_of_variance)

### **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

| 1  2  3  4  5  6  7  8  9  10  11 | # 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