HypothesisTests.jl Documentation

Release 0.1

HypothesisTests.jl Contributors

October 04, 2016

Contents:

CHAPTER 1

Methods

The following main concepts are implemented by most of the Parametric tests and Nonparametric tests:

1.1 Confidence Interval

confint (*test::HypothesisTest*, *alpha=0.05; tail=:both*)

Compute a confidence interval C with coverage 1-`alpha`.

If `tail` is `:both` (default), then a two-sided confidence interval is returned. If `tail` is `:left` or `:right`, then a one-sided confidence interval is returned

Note: Most of the implemented confidence intervals are *strongly consistent*, that is, the confidence interval with coverage 1-`alpha` does not contain the test statistic under h_0 if and only if the corresponding test rejects the null hypothesis $h_0 : \theta = \theta_0$:

$$C(x, 1\alpha) = \{\theta : p_\theta(x) > \alpha\},$$

where p_θ is the *p-value* of the corresponding test.

1.1.1 Confidence Interval for Binomial Proportions

confint (*test::BinomialTest*, *alpha=0.05; tail=:both*, *method=:clopper_pearson*)

Compute a confidence interval with coverage 1-`alpha` for a binomial proportion using one of the following methods. Possible values for `method` are:

- Clopper-Pearson interval `:clopper_pearson` (default): This interval is based on the binomial distribution. The empirical coverage is never less than the nominal coverage of 1-`alpha`; it is usually too conservative.
- Wald interval `:wald` (normal approximation interval): This interval relies on the standard approximation of the actual binomial distribution by a normal distribution. Coverage can be erratically poor for success probabilities close to zero or one.
- Wilson score interval `:wilson`: This interval relies on a normal approximation. In contrast to `:wald` the standard deviation is not approximated by an empirical estimate resulting in good empirical coverages even for small numbers of draws and extreme success probabilities.
- Jeffreys interval `:jeffrey`: Bayesian confidence interval obtained by using a non-informative Jeffreys prior. The interval is very similar to the Wilson interval.

- Agresti Coull interval `:agresti_coull`: Simplified version of the Wilson interval; they are centered around the same value. The Agresti Coull interval has higher or equal coverage.

References:

- Brown, L.D., Cai, T.T., and DasGupta, A. Interval estimation for a binomial proportion. Statistical Science, 16(2):101–117, 2001.

1.1.2 Confidence Interval for Multinomial Proportions

`confint` (*test::PowerDivergenceTest*, *alpha=0.05; tail=:both*, *method=:sison_glaz*)

Compute a confidence interval with coverage 1-`alpha` for multinomial proportions using one of the following methods. Possible values for `method` are:

- Sison, Glaz intervals `:sison_glaz` (default):
- Bootstrap intervals `:bootstrap` :
- Quesenberry, Hurst intervals `:quesenberry_hurst` :
- Gold intervals `:gold` (Asymptotic simultaneous intervals):

References:

- Agresti, Alan. Categorical Data Analysis, 3rd Edition. Wiley, 2013.
- Sison, C.P and Glaz, J. Simultaneous confidence intervals and sample size determination for multinomial proportions. Journal of the American Statistical Association, 90:366-369, 1995.
- Quesensberry, C.P. and Hurst, D.C. Large Sample Simultaneous Confidence Intervals for Multinational Proportions. Technometrics, 6:191-195, 1964.
- Gold, R. Z. Tests Auxiliary to χ^2 Tests in a Markov Chain. Annals of Mathematical Statistics, 30:56-74, 1963.

1.1.3 Confidence Interval for Fisher exact test

`confint` (*x::FisherExactTest*, *alpha::Float64=0.05; tail=:both*, *method=:central*)

Compute a confidence interval with coverage 1-`alpha` by inverting the `:central` p-value.

References:

- Gibbons, J.D, Pratt, J.W. P-values: Interpretation and Methodology American Statistican, 29(1):20-25, 1975.
- Fay, M.P. Supplementary material to Confidence intervals that match Fisher's exact or Blaker's exact tests. Biostatistics, 0(0):1-13, 2009.

1.2 p-value

`pvalue` (*test::HypothesisTest; tail=:both*)

Compute the p-value for a given significance test.

If `tail` is `:both` (default), then the p-value for the two-sided test is returned. If `tail` is `:left` or `:right`, then a one-sided test is performed.

1.2.1 p-value for Fisher exact test

function pvalue(x::FisherExactTest; tail=:both, method=:central)

Compute the p-value for a given significance test. The one-sided p-values are based on Fisher's non-central hypergeometric distribution $f_\omega(i)$ with odd-ratio ω:

$$p_\omega^{(\text{left})} = \sum_{i \le a} f_\omega(i)$$
$$p_\omega^{(\text{right})} = \sum_{i \ge a} f_\omega(i)$$

For `tail=:both`, possible values for `method` are:

- Central interval `:central` (default): This p-value is two times the minimum of the one-sided p-values.
- Minimum likelihood interval `:minlike`: This p-value is computed by summing all tables with the same marginals that are equally or less probable:

$$p_\omega = \sum_{f_\omega(i) \le f_\omega(a)} f_\omega(i)$$

Note: Since the p-value is not necessarily unimodal, the corresponding confidence region might not be an interval.

References:

- Gibbons, J.D, Pratt, J.W. P-values: Interpretation and Methodology American Statistican, 29(1):20-25, 1975.
- Fay, M.P. Supplementary material to Confidence intervals that match Fisher's exact or Blaker's exact tests. Biostatistics, 0(0):1-13, 2009.

CHAPTER 2

Parametric tests

2.1 T-test

OneSampleTTest (*v::AbstractVector{T<:Real}, mu0::Real=0*)

Perform a one sample t-test of the null hypothesis that the data in vector `v` comes from a distribution with mean `mu0` against the alternative hypothesis that the distribution does not have mean `mu0`.

Implements: *pvalue*, confint

OneSampleTTest (*xbar::Real, stdev::Real, n::Int, mu0::Real=0*)

Perform a one sample t-test of the null hypothesis that `n` values with mean `xbar` and sample standard deviation `stdev` come from a distribution with `mu0` against the alternative hypothesis that the distribution does not have mean `mu0`.

Implements: *pvalue*, confint

OneSampleTTest (*x::AbstractVector{T<:Real}, y::AbstractVector{T<:Real}, mu0::Real=0*)

Perform a paired sample t-test of the null hypothesis that the differences between pairs of values in vectors `x` and `y` come from a distribution with `mu0` against the alternative hypothesis that the distribution does not have mean `mu0`.

Implements: *pvalue*, confint

EqualVarianceTTest (*x::AbstractVector{T<:Real}, y::AbstractVector{T<:Real}*)

Perform a two-sample t-test of the null hypothesis that `x` and `y` come from a distributions with the same mean and equal variances against the alternative hypothesis that the distributions have different means and but equal variances.

Implements: *pvalue*, confint

UnequalVarianceTTest (*x::AbstractVector{T<:Real}, y::AbstractVector{T<:Real}*)

Perform an unequal variance two-sample t-test of the null hypothesis that `x` and `y` come from a distributions with the same mean against the alternative hypothesis that the distributions have different means.

This test is also known as sometimes known as Welch's t-test. It differs from the equal variance t-test in that it computes the number of degrees of freedom of the test using the Welch-Satterthwaite equation:

$$\nu_{\chi'} \approx \frac{\left(\sum_{i=1}^n k_i s_i^2\right)^2}{\sum_{i=1}^n \frac{(k_i s_i^2)^2}{\nu_i}}$$

Implements: *pvalue*, confint

2.2 Power Divergence Test

PowerDivergenceTest (*x [,y] [, lambda] [,theta0]*)

If `x` is a matrix with one row or column, or if `x` is a vector and `y` is not given, then a goodness-of-fit test is performed (`x` is treated as a one- dimensional contingency table. The entries of `x` must be non-negative integers. In this case, the hypothesis tested is whether the population probabilities equal those in `theta0`, or are all equal if `theta0` is not given.

If `x` is a matrix with at least two rows and columns, it is taken as a two-dimensional contigency table: the entries of `x` must be non-negative integers. Otherwise, `x` and `y` must be vectors of the same length. The contigency table is calculated using `counts` from `Statsbase`. Then the power divergence test is performed of the null hypothesis that the joint distribution of the cell counts in a 2-dimensional contingency table is the product of the row and column marginals.

The power divergence test is given by

$$\frac{2}{\lambda(\lambda+1)}\sum_{i=1}^{I}\sum_{j=1}^{J} n_{ij}\left[(n_{ij}/\hat{n}_{ij})^{\lambda}-1\right]$$

where n_{ij} is the cell count in the i th row and j th column and λ is a real number. Note that when $\lambda = 1$, this is equal to Pearson's chi-squared statistic, as :math'lambda to 0', it converges to the likelihood ratio test statistic, as $\lambda \to -1$ it converges to the minimum discrimination information statistic (Gokhale and Kullback 1978), for $\lambda = -2$ it equals Neyman modified chi-squared (Neyman 1949), and for $\lambda = -1/2$ it equals the Freeman-Tukey statistic (Freeman and Tukey 1950). Under regulairty conditions, their asymptotic distributions are identical (see Drost et. al. 1989). The chis-squared null approximation works best for λ near $2/3$.

Implements: *pvalue*, confint

References:

- Agresti, Alan. Categorical Data Analysis, 3rd Edition. Wiley, 2013.

ChisqTest (*x [,y] [,theta0]*)

Convenience function for power divergence test with $\lambda = 1$.

MultinomialLRT (*x [,y] [,theta0]*)

Convenience function for power divergence test with $\lambda = 0$.

Nonparametric tests

In contrast to Parametric tests, these tests do not involve specific probability distributions. In general, they are less powerful than their parametric counterparts. The package contains the following nonparametric tests:

3.1 Binomial test

`BinomialTest` (*x::Integer*, *n::Integer*, *p::Real=0.5*)

Perform a binomial test of the null hypothesis that the distribution from which `x` successes were encountered in `n` draws has success probability `p` against the alternative hypothesis that the success probability is not equal to `p`.

Computed confidence intervals are Clopper-Pearson intervals.

Implements: *pvalue*, *confint*

`BinomialTest` (*x::AbstractVector{Bool}*, *p::Real=0.5*)

Perform a binomial test of the null hypothesis that the distribution from which `x` was drawn has success probability `p` against the alternative hypothesis that the success probability is not equal to `p`.

Implements: *pvalue*, *confint*

3.2 Fisher exact test

`FisherExactTest` (*a::Integer*, *b::Integer*, *c::Integer*, *d::Integer*)

Perform Fisher's exact test of the null hypothesis that the success probabilities `a/c` and `b/d` are equal, that is the odds ratio `(a/c) / (b/d)` is one, against the alternative hypothesis that they are not equal.

The contingency table is structured as:

	X1	X1
Y1	a	b
Y2	c	d

Note: The print output contains the conditional maximum likelihood estimate of the odd-ratio rather than the sample odds ratio; it maximizes the likelihood given by Fisher's non-central hypergeometric distribution.

Implements: *pvalue*, *confint*

References:

- Fay, M.P. Supplementary material to Confidence intervals that match Fisher's exact or Blaker's exact tests. Biostatistics, 0(0): 1-13, 2009.

3.3 Kolmogorov–Smirnov test

The null hypothesis of the Kolmogorov–Smirnov test is that a dataset comes from a certain distribution; the reference distribution can be specified explicitely (one-sample test) or by an empirical sample (two-sample test). The alternative hypothesis is that the cumulative distributions of the sample is different (`tail=:both`; default), smaller (`tail=:left`), or larger (``tail:=right`) than the reference cumulative distribution. The exact test is based on the exact distribution of the differences whereas the approximate test is derived from its asymptotic distribution.

`ExactOneSampleKSTest{T<:Real}(x::AbstractVector{T}, d::UnivariateDistribution)`

Perform a one sample Kolmogorov–Smirnov-test of the null hypothesis that the data in vector `x` comes from the distribution `d` against the alternative hypothesis that the sample is not drawn from `d`.

Implements: *pvalue*

`ApproximateOneSampleKSTest{T<:Real}(x::AbstractVector{T}, d::UnivariateDistribution)`

Perform an asymptotic one sample Kolmogorov–Smirnov-test of the null hypothesis that the data in vector `x` comes from the distribution `d` against the alternative hypothesis that the sample is not drawn from `d`.

Implements: *pvalue*

`ApproximateTwoSampleKSTest{T<:Real, S<:Real}(x::AbstractVector{T}, y::AbstractVector{S})`

Perform an asymptotic two sample Kolmogorov–Smirnov-test of the null hypothesis that `x` and `y` are drawn from the same distribution against the alternative hypothesis that the distribution comes from different distributions.

Implements: *pvalue*

References:

- Approximation of one-sided test: http://www.encyclopediaofmath.org/index.php/Kolmogorov-Smirnov_test

3.4 Kruskal-Wallis rank sum test

`KruskalWallisTest{T<:Real}(groups::AbstractVector{T}...)`

Perform Kruskal Wallis rank sum test of the null hypothesis that the location parameters of the distribution of the n observations are the same in each of the `groups` $\mathcal{G}$ against the alternative hypothesis that they differ in at least one.

The Kruskal-Wallis test is an extension of the *Mann-Whitney U test* to more than two groups.

The p-value is computed using a chi-square approximation to the distribution of the test statistic $H_c = \frac{H}{C}$:

$$H = \frac{12}{n(n+1)} \sum_{g \in \mathcal{G}} \frac{R_g^2}{n_g} - 3(n+1)$$

$$C = 1 - \frac{1}{n^3 - n} \sum_{t \in \mathcal{T}} (t^3 - t),$$

where $\mathcal{T}$ is the set of the counts of tied values at each tied position, n_g is the number of observations and R_g is the rank sum in group g. See references for further details.

Implements: *pvalue*

References:

- Meyer, J.P, Seaman, M.A., Expanded tables of critical values for the Kruskal-Wallis H statistic. Paper presented at the annual meeting of the American Educational Research Association, San Francisco, April 2006.

3.5 Mann Whitney U test

`MannWhitneyUTest` (*x::AbstractVector{T<:Real}*, *y::AbstractVector{T<:Real}*)

Perform a Mann-Whitney U test of the null hypothesis that the probability that an observation drawn from the same population as `x` is greater than an observation drawn from the same population as `y` is equal to the probability that an observation drawn from the same population as `y` is greater than an observation drawn from the same population as `x` against the alternative hypothesis that these probabilities are not equal.

The Mann-Whitney U test is sometimes known as the Wilcoxon rank sum test.

When there are no tied ranks and 50 samples, or tied ranks and 10 samples, `MannWhitneyUTest` performs an exact Mann-Whitney U test. In all other cases, `MannWhitneyUTest` performs an approximate Mann-Whitney U test. Behavior may be further controlled by using `ExactMannWhitneyUTest` or `ApproximateMannWhitneyUTest` directly. See below for further algorithmic details.

Implements: *pvalue*

`ExactMannWhitneyUTest` (*x::AbstractVector{T<:Real}*, *y::AbstractVector{T<:Real}*)

Perform an exact Mann-Whitney U test.

When there are no tied ranks, the exact p-value is computed using the `pwilcox` function from libRmath. In the presence of tied ranks, a p-value is computed by exhaustive enumeration of permutations, which can be very slow for even moderately sized data sets.

Implements: *pvalue*

`ApproximateMannWhitneyUTest` (*x::AbstractVector{T<:Real}*, *y::AbstractVector{T<:Real}*)

Perform an approximate Mann-Whitney U test.

The p-value is computed using a normal approximation to the distribution of the Mann-Whitney U statistic:

$$\mu = \frac{n_x n_y}{2}$$
$$\sigma = \frac{n_x n_y}{12}\left(n_x + n_y + 1 - \frac{a}{(n_x+n_y)(n_x+n_y-1)}\right)$$
$$a = \sum_{t\in\mathcal{T}} t^3 - t$$

where $\mathcal{T}$ is the set of the counts of tied values at each tied position.

Implements: *pvalue*

3.6 Sign test

`SignTest` (*x::AbstractVector{T<:Real}*, *median::Real=0*)

Perform a sign test of the null hypothesis that the distribution from which `x` was drawn has median `median` against the alternative hypothesis that the median is not equal to `median`.

Implements: *pvalue*, confint

`SignTest` (*x::AbstractVector{T<:Real}*, *y::AbstractVector{T<:Real}*, *median::Real=0*)

Perform a sign test of the null hypothesis that the distribution of `x - y` has median `median` against the alternative hypothesis that the median is not equal to `median`.

Implements: *pvalue*, confint

3.7 Wilcoxon signed rank test

SignedRankTest (*x::AbstractVector{T<:Real}*, *y::AbstractVector{T<:Real}*)

Perform a Wilcoxon signed rank test of the null hypothesis that the distribution of the difference `x - y` has zero median against the alternative hypothesis that the median is non-zero.

When there are no tied ranks and 50 samples, or tied ranks and 15 samples, `SignedRankTest` performs an exact signed rank test. In all other cases, `SignedRankTest` performs an approximate signed rank test. Behavior may be further controlled by using `ExactSignedRankTest` or `ApproximateSignedRankTest` directly. See below for further algorithmic details.

Implements: *pvalue*

SignedRankTest (*x::AbstractVector{T<:Real}*)

Perform a Wilcoxon signed rank test of the null hypothesis that the distribution from which `` `x` `` is drawn has zero median against the alternative hypothesis that the median is non-zero.

Implements: *pvalue*

ExactSignedRankTest (*x::AbstractVector{T<:Real}*[, *y::AbstractVector{T<:Real}*])

Perform an exact signed rank U test.

When there are no tied ranks, the exact p-value is computed using the `psignrank` function from libRmath. In the presence of tied ranks, a p-value is computed by exhaustive enumeration of permutations, which can be very slow for even moderately sized data sets.

Implements: *pvalue*

ApproximateSignedRank (*x::AbstractVector{T<:Real}*[, *y::AbstractVector{T<:Real}*])

Perform an approximate signed rank U test.

The p-value is computed using a normal approximation to the distribution of the signed rank statistic:

$$\begin{aligned}
\mu &= \frac{n(n+1)}{4} \\
\sigma &= \frac{n(n+1)(2*n+1)}{24} - \frac{a}{48} \\
a &= \sum_{t\in\mathcal{T}} t^3 - t
\end{aligned}$$

where $\mathcal{T}$ is the set of the counts of tied values at each tied position.

Implements: *pvalue*

3.8 Anderson–Darling test

The null hypothesis of the Anderson–Darling test is that a dataset comes from a certain distribution; the reference distribution can be specified explicitely (one-sample test). K-sample Anderson–Darling tests are available for testing whether several samples are coming from a single population drawn from the distribution function which does not have to be specified.

OneSampleADTest{T<:Real}(x::AbstractVector{T}, d::UnivariateDistribution)

Perform a one sample Anderson–Darling test of the null hypothesis that the data in vector `x` comes from the distribution `d` against the alternative hypothesis that the sample is not drawn from `d`.

Implements: *pvalue*

KSampleADTest{T<:Real}(xs::AbstractVector{T}...; modified=true)

Perform an k-sample Anderson–Darling test of the null hypothesis that the data in vectors `xs` comes from the same distribution against the alternative hypothesis that the samples comes from different distributions.

`modified` paramater enables a modified test calculation for samples whose observations do not all coincide.

Implements: *pvalue*

References:

- k-Sample Anderson-Darling Tests, F. W. Scholz and M. A. Stephens, Journal of the American Statistical Association, Vol. 82, No. 399. (Sep., 1987), pp. 918-924.

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

Index