S-Lang Statistics Module Reference

John E. Davis, davis@space.mit.edu Nov 6, 2007

Contents

1 Introduction to the Statistics Module **5**

2 Using the Module **7**

3 Statistics Module Function Reference **9**

3.1 Statistical Tests 9

3.1.1 **ks_test** 9

3.1.2 **ks_test2** 10

3.1.3 **kuiper_test** 10

3.1.4 **kuiper_test2** 11

3.1.5 **chisqr_test** 11

3.1.6 **mw_test** 11

3.1.7 **f_test2** 12

3.1.8 **t_test** 13

3.1.9 **t_test2** 13

3.1.10 **welch_t_test2** 14

3.1.11 **z_test** 14

3.1.12 **kendall_tau** 14

3.1.13 **pearson_r** 15

3.1.14 **spearman_r** 15

3.2 Cumulative Distribution Functions 16

3.2.1 **chisqr_cdf** 16

3.2.2 **poisson_cdf** 16

3.2.3 **smirnov_cdf** 17

3.2.4 **normal_cdf** 17

3.2.5 **mann_whitney_cdf** 17

3.2.6 **kim_jennrich_cdf** 18

3.2.7 **f_cdf** 18
3.3 Miscellaneous Statistical Functions 19
3.3.1 **median** 19
3.3.2 **median_nc** 19
3.3.3 **mean** 20
3.3.4 **stddev** 20
3.3.5 **skewness** 21
3.3.6 **kurtosis** 21
3.3.7 **binomial** 21

Chapter 1

Introduction to the Statistics Module

The **S-lang** statistics module contains a number of well-known statistical tests and functions for the statistical analysis of data. Wherever possible routines that compute the exact "p-values" have been implemented, e.g., for the Kolmogorov-Smirnov tests.

It is assumed that the user of the module is familiar with statistics and significance-testing. This document describes how to use the module and makes no attempt to describe the inherent strengths or weaknesses of the statistical tests implemented by the module. The user is referred to the statistical literature for information about the latter.

Chapter 2

Using the Module

To use the module in a **S-lang** script, it must first be loaded into the interpreter. The standard way to do this is to load it using the `require` function, e.g.,

```
require ("stats");
```

To load it into a specific namespace, e.g., "S", use

```
require ("stats", "S");
```

Most of the `stats` module's functions provide a brief usage message when called without arguments, e.g.,

```
slsh> chisqr_test;
Usage: p=chisqr_test(X,Y,...,Z [,&T])
```

More detailed help is available using the help function:

```
slsh> help chisqr_test
chisqr_test

 SYNOPSIS
  Apply the Chi-square test to a two or more datasets

 USAGE
  prob = chisqr_test (X_1, X_2, ..., X_N [,&t])

 DESCRIPTION
 This function applies the Chi-square test to the N datasets
    .
    .
```

To illustrate the use of the module, consider the task of comparing gaussian-distributed random numbers to a uniform distribution of numbers. In the following, the `ran_gaussian` function from the GNU Scientific Library module will be used to generate the gaussian distributed random numbers.

First, start by loading the `stats` and `gslrand` modules into slsh :

```
slsh> require ("gslrand");
slsh> require ("stats");
```

Now generate 10 random numbers with a variance of 1.0 using the `ran_gaussian` and assign the resulting array to the variable `g`:

```
slsh> g = ran_gaussian (1.0, 10);
```

Similarly, assign `u` to a uniformly distributed range of 10 numbers from -3 to 3:

```
slsh> u = [-3:3:#10];
```

These two datasets may be compared using the `stats` module's two-sample non-parametric tests. First the Kolmogorov-Smirnov test may be applied using `ks_test2`:

```
slsh> ks_test2 (g,u);
0.78693
```

This shows a p-value of about 0.79, which indicates that there is no significant difference between these distributions. Similary, the Kuiper and Mann-Whitney-Wilcoxon tests yield p-values of 0.46, and 0.97, respectively:

```
slsh> mw_test (g,u);
0.970512
slsh> kuiper_test2 (g,u);
0.462481
```

Instead of 10 points per dataset, perform the tests using 100 points:

```
slsh> g = ran_gaussian (1.0, 100);
slsh> u = [-3:3:#100];
slsh> ks_test2 (g,u);
0.00613403
slsh> mw_test (g,u);
0.741508
slsh> kuiper_test2 (g,u);
1.38757e-06
```

As this example shows, both the Kolmogorov-Smirnov and Kuiper tests found significant differences between the data sets, whereas the Mann-Whitney-Wilcoxon test failed to find a significant difference. The fact that the Mann-Whitney-Wilcoxon test failed to find a difference is that the test assumes that the underlying distributions have the same shape but may differ in location. Clearly the distributions represented by `g` and `u` violate this assumption.

Chapter 3

Statistics Module Function Reference

3.1 Statistical Tests

3.1.1 ks_test

Synopsis

One sample Kolmogorov test

Usage

```
p = ks_test (CDF [,&D])
```

Description

This function applies the Kolmogorov test to the data represented by `CDF` and returns the p-value representing the probability that the data values are "consistent" with the underlying distribution function. If the optional parameter is passed to the function, then it must be a reference to a variable that, upon return, will be set to the value of the Kolmogorov statistic..

The `CDF` array that is passed to this function must be computed from the assumed probability distribution function. For example, if the data are constrained to lie between 0 and 1, and the null hypothesis is that they follow a uniform distribution, then the CDF will be equal to the data. In the data are assumed to be normally (Gaussian) distributed, then the `normal_cdf` function can be used to compute the CDF.

Example

Suppose that X is an array of values obtained from repeated measurements of some quantity. The values are are assumed to follow a normal distribution with a mean of 20 and a standard deviation of 3. The `ks_test` may be used to test this hypothesis using:

```
pval = ks_test (normal_cdf(X, 20, 3));
```

See Also

3.1.2 (ks_test2), 3.1.3 (kuiper_test), 3.1.8 (t_test), 3.1.11 (z_test)

3.1.2 ks_test2

Synopsis

Two-Sample Kolmogorov-Smirnov test

Usage

```
prob = ks_test2 (X, Y [,&d])
```

Description

This function applies the 2-sample Kolmogorov-Smirnov test to two datasets `X` and `Y` and returns p-value for the null hypothesis that they share the same underlying distribution. If the optional parameter is passed to the function, then it must be a reference to a variable that, upon return, will be set to the value of the statistic.

Notes

If `length(X)*length(Y)<=10000`, the `kim_jennrich_cdf` function will be used to compute the exact probability. Otherwise an asymptotic form will be used.

See Also

3.1.1 (ks_test), 3.1.3 (kuiper_test), 3.2.6 (kim_jennrich_cdf)

3.1.3 kuiper_test

Synopsis

Perform a 1-sample Kuiper test

Usage

```
pval = kuiper_test (CDF [,&D])
```

Description

This function applies the Kuiper test to the data represented by `CDF` and returns the p-value representing the probability that the data values are "consistent" with the underlying distribution function. If the optional parameter is passed to the function, then it must be a reference to a variable that, upon return, will be set to the value of the Kuiper statistic.

The `CDF` array that is passed to this function must be computed from the assumed probability distribution function. For example, if the data are constrained to lie between 0 and 1, and the null hypothesis is that they follow a uniform distribution, then the CDF will be equal to the data. In the data are assumed to be normally (Gaussian) distributed, then the `normal_cdf` function can be used to compute the CDF.

Example

Suppose that X is an array of values obtained from repeated measurements of some quantity. The values are are assumed to follow a normal distribution with a mean of 20 and a standard deviation of 3. The `ks_test` may be used to test this hypothesis using:

```
pval = kuiper_test (normal_cdf(X, 20, 3));
```

See Also

3.1.4 (kuiper_test2), 3.1.1 (ks_test), 3.1.8 (t_test)

3.1.4 kuiper_test2

Synopsis

Perform a 2-sample Kuiper test

Usage

```
pval = kuiper_test2 (X, Y [,&D])
```

Description

This function applies the 2-sample Kuiper test to two datasets `X` and `Y` and returns p-value for the null hypothesis that they share the same underlying distribution. If the optional parameter is passed to the function, then it must be a reference to a variable that, upon return, will be set to the value of the Kuiper statistic.

Notes

The p-value is computed from an asymotic formula suggested by Stephens, M.A., Journal of the American Statistical Association, Vol 69, No 347, 1974, pp 730-737.

See Also

3.1.2 (ks_test2), 3.1.3 (kuiper_test)

3.1.5 chisqr_test

Synopsis

Apply the Chi-square test to a two or more datasets

Usage

```
prob = chisqr_test (X_1[], X_2[], ..., X_N [,&t])
```

Description

This function applies the Chi-square test to the N datasets `X_1`, `X_2`, ..., `X_N`, and returns the probability that each of the datasets were drawn from the same underlying distribution. Each of the arrays `X_k` must be the same length. If the last parameter is a reference to a variable, then upon return the variable will be set to the value of the statistic.

See Also

3.2.1 (chisqr_cdf), 3.1.2 (ks_test2), 3.1.6 (mw_test)

3.1.6 mw_test

Synopsis

Apply the Two-sample Wilcoxon-Mann-Whitney test

Usage

```
p = mw_test(X, Y [,&w])
```

Description

This function performs a Wilcoxon-Mann-Whitney test and returns the p-value for the null hypothesis that there is no difference between the distributions represented by the datasets `X` and `Y`.

If a third argument is given, it must be a reference to a variable whose value upon return will be to to the rank-sum of `X`.

Qualifiers

The function makes use of the following qualifiers:

```
side=">"  :    H0: P(X<Y) >= 1/2    (right-tail)
side="<"  :    H0: P(X<Y) <= 1/2    (left-tail)
```

The default null hypothesis is that `P(X<Y)=1/2`.

Notes

There are a number of definitions of this test. While the exact definition of the statistic varies, the p-values are the same.

If `length(X)`$<$50, `length(Y)` < 50, and ties are not present, then the exact p-value is computed using the `mann_whitney_cdf` function. Otherwise a normal distribution is used.

This test is often referred to as the non-parametric generalization of the Student t-test.

See Also

3.2.5 (mann_whitney_cdf), 3.1.2 (ks_test2), 3.1.5 (chisqr_test), 3.1.8 (t_test)

3.1.7 f_test2

Synopsis

Apply the Two-sample F test

Usage

```
p = f_test2 (X, Y [,&F]
```

Description

This function computes the two-sample F statistic and its p-value for the data in the `X` and `Y` arrays. This test is used to compare the variances of two normally-distributed data sets, with the null hypothesis that the variances are equal. The return value is the p-value, which is computed using the module's `f_cdf` function.

Qualifiers

The function makes use of the following qualifiers:

```
side=">"  :    H0: Var[X] >= Var[Y]  (right-tail)
side="<"  :    H0: Var[X] <= Var[Y]  (left-tail)
```

See Also

3.2.7 (f_cdf), 3.1.2 (ks_test2), 3.1.5 (chisqr_test)

3.1.8 t_test

Synopsis

Perform a Student t-test

Usage

```
pval = t_test (X, mu [,&t])
```

Description

This function computes Student's t-statistic and returns the p-value that the data X are consistent with a Gaussian distribution with a mean of `mu`. If the optional parameter is passed to the function, then it must be a reference to a variable that, upon return, will be set to the value of the statistic.

Qualifiers

The following qualifiers may be used to specify a 1-sided test:

```
side="<"      Perform a left-tailed test
side=">"      Perform a right-tailed test
```

Notes

Strictly speaking, this test should only be used if the variance of the data are equal to that of the assumed parent distribution. Use the Mann-Whitney-Wilcoxon (`mw_test`) if the underlying distribution is non-normal.

See Also

3.1.6 (mw_test), 3.1.9 (t_test2)

3.1.9 t_test2

Synopsis

Perform a 2-sample Student t-test

Usage

```
pval = t_test2 (X, Y [,&t])
```

Description

This function compares two data sets `X` and `Y` using the Student t-statistic. It is assumed that the the parent populations are normally distributed with equal variance, but with possibly different means. The test is one that looks for differences in the means.

Notes

The `welch_t_test2` function may be used if it is not known that the parent populations have the same variance.

See Also

3.1.9 (t_test2), 3.1.10 (welch_t_test2), 3.1.6 (mw_test)

3.1.10 welch_t_test2

Synopsis

Perform Welch's t-test

Usage

```
pval = welch_t_test2 (X, Y [,&t])
```

Description

This function applies Welch's t-test to the 2 datasets `X` and `Y` and returns the p-value that the underlying populations have the same mean. The parent populations are assumed to be normally distributed, but need not have the same variance. If the optional parameter is passed to the function, then it must be a reference to a variable that, upon return, will be set to the value of the statistic.

Qualifiers

The following qualifiers may be used to specify a 1-sided test:

```
side="<"      Perform a left-tailed test
side=">"      Perform a right-tailed test
```

See Also

3.1.9 (t_test2)

3.1.11 z_test

Synopsis

Perform a Z test

Usage

```
pval = z_test (X, mu, sigma [,&z])
```

Description

This function applies a Z test to the data `X` and returns the p-value that the data are consistent with a normally-distributed parent population with a mean of `mu` and a standard-deviation of `sigma`. If the optional parameter is passed to the function, then it must be a reference to a variable that, upon return, will be set to the value of the Z statistic.

See Also

3.1.8 (t_test), 3.1.6 (mw_test)

3.1.12 kendall_tau

Synopsis

Kendall's tau Correlation Test

Usage

```
pval = kendall_tau (x, y [,&tau]
```

Description

This function computes Kendall's tau statistic for the paired data values (x,y). It returns the p-value associated with the statistic.

Notes

The current version of this function uses an asymptotic formula based upon the normal distribution to compute the p-value.

Qualifiers

The following qualifiers may be used to specify a 1-sided test:

```
side="<"       Perform a left-tailed test
side=">"       Perform a right-tailed test
```

See Also

3.1.14 (spearman_r), 3.1.13 (pearson_r)

3.1.13 pearson_r

Synopsis

Compute Pearson's Correlation Coefficient

Usage

```
pval = pearson_r (X, Y [,&r])
```

Description

This function computes Pearson's r correlation coefficient of the two datasets `X` and `Y`. It returns the the p-value that `x` and `y` are mutually independent. If the optional parameter is passed to the function, then it must be a reference to a variable that, upon return, will be set to the value of the correlation coefficient.

Qualifiers

The following qualifiers may be used to specify a 1-sided test:

```
side="<"       Perform a left-tailed test
side=">"       Perform a right-tailed test
```

See Also

3.1.12 (kendall_tau), 3.1.14 (spearman_r)

3.1.14 spearman_r

Synopsis

Spearman's Rank Correlation test

Usage

```
pval = spearman_r(x, y [,&r])
```

Description

This function computes the Spearman rank correlation coefficient (r) and returns the p-value that `x` and `y` are mutually independent. If the optional parameter is passed to the function, then it must be a reference to a variable that, upon return, will be set to the value of the correlation coefficient.

Qualifiers

The following qualifiers may be used to specify a 1-sided test:

```
side="<"      Perform a left-tailed test
side=">"      Perform a right-tailed test
```

See Also

3.1.12 (kendall_tau), 3.1.13 (pearson_r)

3.2 Cumulative Distribution Functions

3.2.1 chisqr_cdf

Synopsis

Compute the Chisqr CDF

Usage

```
cdf = chisqr_cdf (Int_Type n, Double_Type d)
```

Description

This function returns the probability that a random number distributed according to the chi-squared distribution for `n` degrees of freedom will be less than the non-negative value `d`.

Notes

The importance of this distribution arises from the fact that if `n` independent random variables `X_1,...X_n` are distributed according to a gaussian distribution with a mean of 0 and a variance of 1, then the sum

```
X_1^2 + X_2^2 + ... + X_n^2
```

follows the chi-squared distribution with `n` degrees of freedom.

See Also

3.1.5 (chisqr_test), 3.2.2 (poisson_cdf)

3.2.2 poisson_cdf

Synopsis

Compute the Poisson CDF

Usage

```
cdf = poisson_cdf (Double_Type m, Int_Type k)
```

Description

This function computes the CDF for the Poisson probability distribution parameterized by the value m.

See Also

3.2.1 (chisqr_cdf)

3.2.3 smirnov_cdf

Synopsis

Compute the Kolmogorov CDF using Smirnov's asymptotic form

Usage

```
cdf = smirnov_cdf (x)
```

Description

This function computes the CDF for the Kolmogorov distribution using Smirnov's asymptotic form. In particular, the implementation is based upon equation 1.4 from W. Feller, "On the Kolmogorov-Smirnov limit theorems for empirical distributions", Annals of Math. Stat, Vol 19 (1948), pp. 177-190.

See Also

3.1.1 (ks_test), 3.1.2 (ks_test2), 3.2.4 (normal_cdf)

3.2.4 normal_cdf

Synopsis

Compute the CDF for the Normal distribution

Usage

```
cdf = normal_cdf (x)
```

Description

This function computes the CDF (integrated probability) for the normal distribution.

See Also

3.2.3 (smirnov_cdf), 3.2.5 (mann_whitney_cdf), 3.2.2 (poisson_cdf)

3.2.5 mann_whitney_cdf

Synopsis

Compute the Mann-Whitney CDF

Usage

```
cdf = mann_whitney_cdf (Int_Type m, Int_Type n, Int_Type s)
```

Description

This function computes the exact CDF P(X<=s) for the Mann-Whitney distribution. It is used by the `mw_test` function to compute p-values for small values of `m` and `n`.

See Also

3.1.6 (mw_test), 3.1.1 (ks_test), 3.2.4 (normal_cdf)

3.2.6 kim_jennrich_cdf

Synopsis

Compute the 2-sample KS CDF using the Kim-Jennrich Algorithm

Usage

```
p = kim_jennrich (UInt_Type m, UInt_Type n, UInt_Type c)
```

Description

This function returns the exact two-sample Kolmogorov-Smirnov probability that that `D_mn <= c/(mn)`, where `D_mn` is the two-sample Kolmogorov-Smirnov statistic computed from samples of sizes `m` and `n`.

The algorithm used is that of Kim and Jennrich. The run-time scales as m*n. As such, it is recommended that asymptotic form given by the `smirnov_cdf` function be used for large values of m*n.

Notes

For more information about the Kim-Jennrich algorithm, see: Kim, P.J., and R.I. Jennrich (1973), Tables of the exact sampling distribution of the two sample Kolmogorov-Smirnov criterion Dmn(m<n), in Selected Tables in Mathematical Statistics, Volume 1, (edited by H. L. Harter and D.B. Owen), American Mathematical Society, Providence, Rhode Island.

See Also

3.2.3 (smirnov_cdf), 3.1.2 (ks_test2)

3.2.7 f_cdf

Synopsis

Compute the CDF for the F distribution

Usage

```
cdf = f_cdf (t, nu1, nu2)
```

Description

This function computes the CDF for the distribution and returns its value.

See Also

3.1.7 (f_test2)

3.3 Miscellaneous Statistical Functions

3.3.1 median

Synopsis

Compute the median of an array of values

Usage

```
m = median (a [,i])
```

Description

This function computes the median of an array of values. The median is defined to be the value such that half of the the array values will be less than or equal to the median value and the other half greater than or equal to the median value. If the array has an even number of values, then the median value will be the smallest value that is greater than or equal to half the values in the array.

If called with a second argument, then the optional argument specifies the dimension of the array over which the median is to be taken. In this case, an array of one less dimension than the input array will be returned.

Notes

This function makes a copy of the input array and then partially sorts the copy. For large arrays, it may be undesirable to allocate a separate copy. If memory use is to be minimized, the `median_nc` function should be used.

See Also

3.3.2 (median_nc), 3.3.3 (mean)

3.3.2 median_nc

Synopsis

Compute the median of an array

Usage

```
m = median_nc (a [,i])
```

Description

This function computes the median of an array. Unlike the `median` function, it does not make a temporary copy of the array and, as such, is more memory efficient at the expense increased run-time. See the `median` function for more information.

See Also

3.3.1 (median), 3.3.3 (mean)

3.3.3 mean

Synopsis

Compute the mean of the values in an array

Usage

```
m = mean (a [,i])
```

Description

This function computes the arithmetic mean of the values in an array. The optional parameter `i` may be used to specify the dimension over which the mean it to be take. The default is to compute the mean of all the elements.

Example

Suppose that `a` is a two-dimensional MxN array. Then

```
m = mean (a);
```

will assign the mean of all the elements of `a` to `m`. In contrast,

```
m0 = mean(a,0);
m1 = mean(a,1);
```

will assign the N element array to `m0`, and an array of M elements to `m1`. Here, the jth element of `m0` is given by `mean(a[*,j])`, and the jth element of `m1` is given by `mean(a[j,*])`.

See Also

3.3.4 (stddev), 3.3.1 (median), 3.3.6 (kurtosis), 3.3.5 (skewness)

3.3.4 stddev

Synopsis

Compute the standard deviation of an array of values

Usage

```
s = stddev (a [,i])
```

Description

This function computes the standard deviation of the values in the specified array. The optional parameter `i` may be used to specify the dimension over which the standard-deviation it to be taken. The default is to compute the standard deviation of all the elements.

See Also

3.3.3 (mean), 3.3.1 (median), 3.3.6 (kurtosis), 3.3.5 (skewness)

3.3.5 skewness

Synopsis

Compute the skewness of an array of values

Usage

```
s = skewness (a)
```

Description

This function computes the so-called skewness of the array `a`.

See Also

3.3.3 (mean), 3.3.4 (stddev), 3.3.6 (kurtosis)

3.3.6 kurtosis

Synopsis

Compute the kurtosis of an array of values

Usage

```
s = kurtosis (a)
```

Description

This function computes the so-called kurtosis of the array `a`.

Notes

This function is defined such that the kurtosis of the normal distribution is 0, and is also known as the "excess-kurtosis".

See Also

3.3.3 (mean), 3.3.4 (stddev), 3.3.5 (skewness)

3.3.7 binomial

Synopsis

Compute binomial coefficients

Usage

```
c = binomial (n [,m])
```

Description

This function computes the binomial coefficients (n m) where (n m) is given by n!/(m!(n-m)!). If `m` is not provided, then an array of coefficients for m=0 to `n` will be returned.