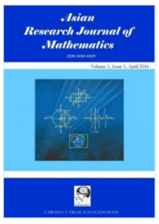


Asian Research Journal of Mathematics

5(2): 1-8, 2017; Article no.ARJOM.34694

ISSN: 2456-477X

Permutation Tests for Nonparametric Statistics Using *R*

Steven T. Garren[1*]

[1] *Department of Mathematics and Statistics, James Madison University, Harrisonburg, VA 22807, USA.*

Author's contribution

The sole author designed, analyzed and interpreted and prepared the manuscript.

Article Information

DOI: 10.9734/ARJOM/2017/34694

Editor(s):

(1) Jaya Bishwal, Department of Mathematics and Statistics, University of North Carolina at Charlotte, USA.

Reviewers:

(1) Diana Bilkova, University of Economics, Czech Republic.

(2) Krasimir Yordzhev, South-West University, Bulgaria.

(3) Jiexiang Li, College of Charleston, USA.

Complete Peer review History: http://www.sciencedomain.org/review-history/19927

Received: 6th June 2017

Accepted: 29th June 2017

Published: 7th July 2017

Original Research Article

Abstract

The *R*-package `jmuOutlier`, which contains functions for performing nonparametric statistical analyses, is introduced. This *R*-package is quite simplistic, and is intended to accompany undergraduate textbooks on nonparametric statistics. The main objectives of this *R*-package include the following: using *R* for running a permutation test on a mean, median, or any other statistic; running a permutation F-test on the difference between location parameters, without using an asymptotic approximation to the F-distribution; running a permutation test on Pearson and Spearman correlation, without using an asymptotic approximation to the normal or t-distribution; running the permutation Siegel-Tukey test in both the forward and reverse directions and the ratio mean deviance test on the difference between scale parameters. Additionally, this *R*-package provides exact power calculations using the binomial test, provides exact confidence intervals on percentiles (based on the binomial test), plots two empirical cumulative distribution functions on the same graph, and produces van der Waerden and exponential scores. The aim is to perform these tests based on either all permutations or a large number of simulated permutations, to obtain p-values without using asymptotic approximations. The greatest contribution of this *R*-package is simplicity, in that an elementary understanding of *R* is sufficient when using this package. We conclude that students can both learn nonparametric statistics and perform analyses using this *R*-package, with just a basic understanding of *R*.

**Corresponding author: E-mail: garrenst@jmu.edu*

Keywords: Nonparametric statistics; permutation tests; power

2010 Mathematics Subject Classification: 62G10, 62G15.

1 Introduction

Functions in *R* [1] written by the author are available online in the *R*-package named `jmuOutlier` through CRAN, so the reader can quickly run nonparametric tests. An undergraduate non-calculus-based textbook based on nonparametric statistics by Higgins [2] is referenced frequently herein, and has a *SAS* companion textbook [3]. The *R*-package `jmuOutlier` is useful for professors who teach nonparametric statistics [2] but prefer to use *R* instead of *SAS*. The functions within `jmuOutlier` are quite easy to use, so that undergraduates who are relatively new to *R* can learn about the statistical field of nonparametric statistics without being delayed by the learning curve of *R*. Since this package `jmuOutlier` also contains functions unrelated to nonparametric statistics (such as `fourier` and `plotVector`), not every function in `jmuOutlier` will be discussed herein, since the purpose of this paper is to discuss nonparametric statistical analysis using *R*. The *R*-functions are available by typing

```
> install.packages( "jmuOutlier" )   # Next, select a mirror site.
> library( jmuOutlier )
```

within *R*.

Several textbooks which discuss nonparametric statistics using *R* are available. Biological data can be analyzed in *R* using nonparametric models [4]. Multivariate nonparametric data analysis also can be performed in *R* [5]. Quite a few *R*-packages for nonparametric statistics are available in the literature. The *R*-package `coin` [6] performs a large array of permutation tests. The *R*-package `nparcomp` [7] performs multiple comparisons and simultaneous confidence intervals for nonparametric statistics. The *R*-package `npmv` [8] analyzes multivariate data using a nonparametric approach. The *R*-package `DPpackage` [9] uses Bayesian semiparametric and nonparametric models. Nonparametric tests for the interaction in two-way factorial designs are performed using *R* [10].

Functions within `jmuOutlier` are based on nonparametric approaches. The motivation for our *R*-package `jmuOutlier` is to provide simple methods in *R* for testing hypotheses for unequal location parameters, hypotheses for unequal scale parameters, and hypotheses for a nonzero correlation coefficient (both Pearson and Spearman). Therefore, nonparametric statistical analysis can be performed in *R* by using `jmuOutlier`, without requiring an advanced knowledge of *R*. Calculations of both power and percentiles based on the nonparametric binomial test also can be performed using functions within `jmuOutlier`. Furthermore, a function within `jmuOutlier` allows plotting two empirical cumulative distribution functions on the same graph, as a neat way to illustrate the nonparametric Kolmogorov-Smirnov test. Finally, permutation tests can be performed quite easily on van der Waerden and exponential scores, using functions within `jmuOutlier`.

We discuss the *R*-functions in `jmuOutlier` which are useful for performing permutation tests in section 2. We introduce additional *R*-functions in `jmuOutlier` which are useful in the field of nonparametric statistics in section 3, and we end with a brief conclusion in section 4.

2 Permutation Tests

The functions `perm.test`, `perm.f.test`, `perm.cor.test`, `siegel.test`, and `rmd.test` are useful for nonparametric statistics, and are available in the `jmuOutlier` package.

2.1 The *R*-function perm.test

Prior to introducing the more commonly-known nonparametric tests, such as the Wilcoxon rank-sum and Wilcoxon signed-rank tests, Higgins [2] explains the concept of a two-sample permutation test in section 2.1 and a one-sample permutation test in section 4.1 of his textbook. The function `perm.test` in `jmuOutlier` performs both the one-sample and two-sample permutation tests, and uses roughly the same format as `t.test`, which of course performs one-sample and two-sample t-tests.

For the two-sample case, `perm.test` computes a p-value based on either all permutations or a large number of permutations of the difference in sample means (the default) or the difference in other statistics (such as sample medians or trimmed sample means). The total number of permutations possible is $(m + n)!/(m!\ n!)$ for sample sizes m and n. A plot of these simulated permutations also may be produced. Higgins [2] shows that a test based on the difference in sample means is equivalent to a test based on the sample sum of just one of the two samples.

For the one-sample case, `perm.test` produces permutations by multiplying observations by $+1$ or -1, so the total number of permutations possible is 2^n, for a sample of size n. If 2^n permutations require too much computing time, then a large number of permutations are simulated when approximating a p-value. Again, the default test statistic is the sample mean, but may be chosen to be something else, such as the sample median.

Examples for the one-sample and two-sample tests are available by typing

```
> x = rnorm( 10, 0.5 ) ;   list( x ) ;   perm.test( x, stat=median )
> y = rnorm( 13, 1 ) ;   list( y ) ;   perm.test( x, y )
```

or by typing

```
> example( perm.test )
```

in *R*.

2.2 The *R*-function perm.f.test

The function `perm.f.test` performs a permutation F-test, based on the one-way analysis of variance F-test statistic. The p-value based on the standard asymptotic method (i.e., using the F-distribution) and the p-value based on permuting treatments are produced. The data may consist of a vector `x` of treatments and a vector `y` of responses or may consist of a matrix of those two column vectors. This p-value is based on a large number of permutations, so the p-value is approximated.

The example

```
> perm.f.test( c( 14,6,5,2,54,7,9,15,11,13,12 ), rep( c("I","II","III"), c(4,4,3) ) )
```

which is also available by typing

```
> example( perm.f.test )
```

produces the permuted p-value of approximately 0.40 based on 20,000 simulations, and produces the asymptotic p-value of 0.38 based on the F-distribution with 2 and 8 degrees of freedom.

2.3 The *R*-function perm.cor.test

The function `perm.cor.test` performs a permutation correlation test, using either the commonly-used Pearson correlation or the Spearman correlation based on ranks. Like `perm.f.test`, the data may consist of a vector `x` of treatments and a vector `y` of responses or may consist of a matrix of those two column vectors. For each permutation, the `x`-data are held fixed, while the `y`-data are randomly permuted. Theoretically, the number of permutations possible is thus $n!$. However, an exact p-value based on all $n!$ permutations is not calculated. Rather, the p-value is based on a large number of randomly sampled permutations, so the p-value calculated is only approximate, just as when using the `perm.f.test` function.

Examples involving both the Pearson correlation and the Spearman correlation are available by typing

```
> x = c( 4, 6, 8, 11 ) ;   y = c( 19, 44, 15, 13 )
> perm.cor.test( x, y, "less", "pearson" )
> perm.cor.test( x, y, "less", "spearman" )
```

or by typing

```
> example( perm.cor.test )
```

to produce p-values of approximately 0.29 when using Pearson correlation and 0.17 when using Spearman correlation.

2.4 The *R*-functions siegel.test and rmd.test

The function `siegel.test` performs the Siegel-Tukey test, by calling function `perm.test`, in both the forward and reverse directions, depending on the value of `reverse` within the `siegel.test` function. This test is similar to the Ansari-Bradley test, which may be performed using the `ansari.test` function available in the standard `stats` *R*-package. The function `rmd.test` performs the ratio mean deviance (RMD) test. Both `siegel.test` and `rmd.test` produce p-values based on either all permutations or a large number of simulated permutations, and are thoroughly explained by Higgins [2].

Examples using `siegel.test` and `rmd.test` are available by typing

```
> siegel.test( c(13, 34, 2, 19, 49, 63), c(17, 29, 22) )
> siegel.test( c(13, 34, 2, 19, 49, 63), c(17, 29, 22), reverse=TRUE )
> rmd.test( c(13, 34, 2, 19, 49, 63), c(17, 29, 22) )
> rmd.test( c(13, 34, 2, 19, 49, 63), c(17, 29, 22), "greater" )
```

or from the `example( siegel.test )` and `example( rmd.test )` commands, to produce exact p-values of 0.17, 0.036, 0.012, and 0.012 (again!), respectively.

3 Additional Useful *R*-Functions

The *R*-functions `power.binom.test`, `quantileCI`, `plotEcdf`, and `score`, along with functions pertaining to the Laplace (double-exponential) distribution, are useful when working in the field of nonparametric statistics. These functions are available in the *R*-package `jmuOutlier`.

3.1 *R*-functions involving the Laplace distribution

Higgins [2] frequently references the normal, uniform, exponential, Laplace (or double exponential), and Cauchy distributions. Since the Laplace distribution is not defined within the standard *R*-packages (e.g., `stats` or `base`), the functions `dlaplace`, `plaplace`, `qlaplace`, and `rlaplace` are in fact included in `jmuOutlier`, to provide the density, the distribution function, the quantile function, and random deviates, respectively, of the Laplace distribution. These four functions use a format similar to some other distribution-related *R*-functions, such as `dnorm`, `pnorm`, `qnorm`, and `rnorm`, in that the arguments for the Laplace *R*-functions are in terms of the mean and standard deviation. Another *R*-package, `rmutil` [11], defines the Laplace distribution in terms of the mean and dispersion parameters, but `jmuOutlier` defines the Laplace distribution in terms of the mean and the standard deviation in order to keep the programming simple.

3.2 The *R*-function power.binom.test

The function `power.binom.test` computes power calculations based on the binomial test. Note that the binomial test on a population median is based on determining the number of observations greater (or less) than the null median. The function `power.binom.test` is somewhat similar in format to `power.t.test` in the *R*-package `stats`, although the former requires the entire specification of the alternative distribution, whereas the latter simply requires knowledge of the alternative mean and standard deviation and uses the asymptotic normality of the sample mean due to the Central Limit Theorem.

As an example, suppose that we test the null hypothesis that the population median is 55 against a right-sided alternative at level $\alpha = 0.05$ and sample size 30. Using

```
> power.binom.test( 30, 0.05, "greater", 55, pnorm, 55.7, 2.5 )
> power.binom.test( 30, 0.05, "greater", 55, plaplace, 55.7, 2.5 )
```

we see that the powers of the binomial test under the alternative distributions of Normal($\mu = 55.7,\ \sigma = 2.5$) and Laplace($\mu = 55.7,\ \sigma = 2.5$) are 0.33 and 0.57, respectively.

As a similar example, suppose that we test the null hypothesis that the population mean (rather than median) is 55 against a right-sided alternative at level $\alpha = 0.05$ and sample size 30. The power of the *t*-test is 0.44 exactly under the alternative Normal($\mu = 55.7,\ \sigma = 2.5$) distribution, but is 0.44 only approximately under the alternative Laplace($\mu = 55.7,\ \sigma = 2.5$) distribution due to the Central Limit Theorem. The command

```
> power.t.test( 30, 0.7, 2.5, type="one.sample", alternative="one.sided")
```

verifies this power calculation. These examples show that the *t*-test is *more* powerful than the binomial test under this Normal alternative, but *less* powerful under this Laplace alternative.

3.3 The *R*-function quantileCI

The *R*-function `quantileCI` produces exact confidence intervals for any percentile. For example, a 95% confidence interval on a population median $\widetilde{\mu}$ consists of all real values of $\widetilde{\mu}_0$ such that $H_0 :\ \widetilde{\mu} = \widetilde{\mu}_0$ is not rejected in favor of $H_a :\ \widetilde{\mu} \neq \widetilde{\mu}_0$ at significance level $\alpha = 0.05$, based on the binomial test described on section 3.2. Note, however, that the *p*-value as a function of $\widetilde{\mu}_0$ has jumps only where $\widetilde{\mu}_0$ is a data value. Therefore, $\widetilde{\mu}_0$ needs to be assigned to only the actual data values and $\pm\infty$ rather than all real values when performing the binomial test, so the endpoints of the confidence interval must be values of the actual data set or $\pm\infty$.

Using the data set `Nile` based on *Flow of the River Nile* in the `datasets` package, 90% confidence intervals on the 25th, 50th, and 75th percentiles are constructed via:

```
> require( datasets )
> quantileCI( as.numeric( Nile ), c( 0.25, 0.5, 0.75 ), 0.9 )
```

to produce the output

```
 lower upper
0.25 759 824
0.5 846 935
0.75 986 1110
```

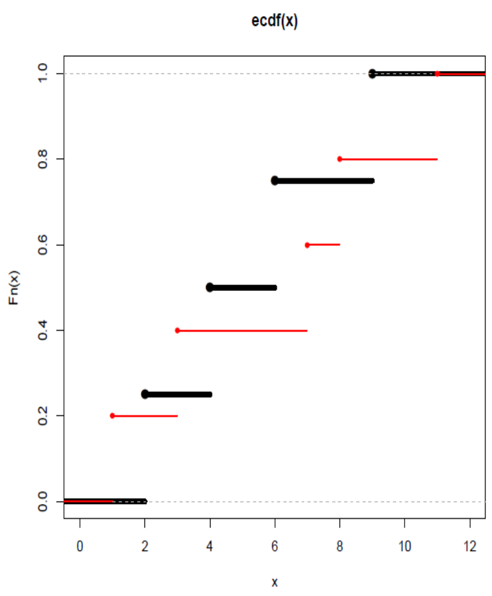
3.4 The *R*-function plotEcdf

The *R*-function `plot.ecdf` in the `stats` *R*-package can plot only one empirical cumulative distribution function at a time. However, the *R*-function `plotEcdf` in the `jmuOutlier` *R*-package can plot one or two empirical cumulative distributions functions on the same graph. This is useful in the field of nonparametric statistics, in that the Kolmogorov-Smirnov test statistic may be illustrated, since this test statistic is based on the maximum difference in the two empirical cumulative distribution functions.

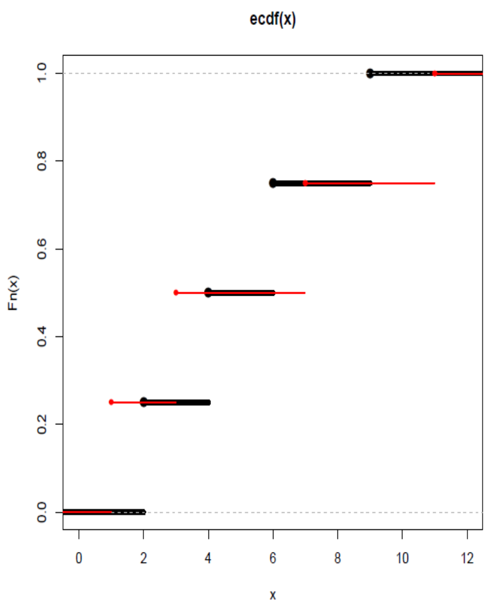
Empirical cumulative distribution functions using `plotEcdf` and different data sets are produced via

```
> plotEcdf( c(2,4,9,6), c(1,7,11,3,8) )
> plotEcdf( c(2,4,9,6), c(1,7,11,3) )
```

and the graphs are shown in Figs. 1a and 1b, respectively.



(a) Data sets {2,4,9,6} and {1,7,11,3,8}



(b) Data sets {2,4,9,6} and {1,7,11,3}

Fig. 1. Using the `plotEcdf` function, with two data sets per graph

3.5 The *R*-function score

The *R*-function `score` determines the van der Waerden scores and the exponential scores (which are statistically equivalent to the Savage scores), as defined in Higgins [2]. The commands

```
> round( score( 12 ), digits=2 )
[1] -1.43 -1.02 -0.74 -0.50 -0.29 -0.10 0.10 0.29 0.50 0.74 1.02 1.43
> round( score( 12, expon=TRUE ), digits=2 )
[1] 0.08 0.17 0.27 0.39 0.51 0.65 0.82 1.02 1.27 1.60 2.10 3.10
```

produce the van der Waerden and exponential scores, respectively, based on twelve observations. These above scores match the scores listed in Table 2.7.1 of Higgins [2].

4 Conclusion

The *R*-package `jmuOutlier` allows students to successfully use *R* in a non-calculus-based nonparametric statistics course, even if the students have no prior experience using *R*. The functions within `jmuOutlier` are intended to be easy-to-use, so that students can learn nonparametric statistics at a rudimentary level without feeling the burden of trying to understand a plethora of details about *R*. The textbook by Higgins [2] and the *R*-package `jmuOutlier` provide an excellent approach for undergraduate students to learn nonparametric statistics using *R*.

The functions within `jmuOutlier` allow rudimentary use in *R* for testing for inequality of location parameters, inequality of scale parameters, and nonzero correlation, based on nonparametric statistics. The originality of these functions is the simplicity when using *R*. Additional functions within `jmuOutlier` allow computing power and percentiles based on the nonparametric binomial test, and also allow computing score functions and plotting two empirical cumulative distribution functions on the same graph. Writing simplistic nonparametric *R*-functions in other fields of statistics, such as multivariate analysis, blocked designs, multifactor experiments, and censored-data analysis, would be a future research goal.

Acknowledgements

The author is very thankful to three anonymous reviewers and an editor, whose helpful suggestions added greatly to the quality of this paper.

Competing Interests

Author has declared that no competing interests exist.

References

[1] R Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria; 2014.
Available: http://www.R-project.org/

[2] Higgins JJ. Introduction to modern nonparametric statistics. Cengage Learning, Boston; 2003.

[3] Richter SJ, Higgins JJ. SAS companion to nonparametric statistics. Cengage Learning, Boston; 2005.

[4] MacFarland TW, Yates JM. Introduction to nonparametric statistics for the biological sciences using R. Springer, New York; 2016.

[5] Oja H. Multivariate nonparametric methods with R: An approach based on spatial signs and ranks. Springer, New York; 2010.

[6] Hothorn T, Hornik K, van de Wiel MA, Zeileis A. Implementing a class of permutation tests: the coin package. Journal of Statistical Software. 2008;28(8):1-23.

[7] Konietschke F, Placzek M, Schaarschmidt F, Hothorn LA. Nparcomp: An R software package for nonparametric multiple comparisons and simultaneous confidence intervals. Journal of Statistical Software. 2015;64(9):1-17.

[8] Ellis AR, Burchett WW, Solomon WH, Bathke AC. Nonparametric inference for multivariate data: The R package npmv. Journal of Statistical Software. 2017;76(4):1-18.

[9] Jara A, Hanson TE, Quintana FA, Müller P, Rosner GL. DPpackage: Bayesian semi- and nonparametric modeling in R. Journal of Statistical Software. 2011;40(5):1-30.

[10] Feys J. Nonparametric tests for the interaction in two-way factorial designs using R. The R Journal. 2016;8(1):367-378.

[11] Swihart B, Lindsey J. Rmutil: Utilities for nonlinear regression and repeated measurements models. R package version 1.1.0; 2017. Accessed 5 June 2017.
Available: https://CRAN.R-project.org/package=rmutil

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
http://sciencedomain.org/review-history/19927