# Entropy-Based Inference using R and the np Package: A Primer

**Jeffrey S. Racine**
McMaster University

**Abstract**

We describe new functionality in the R (R Development Core Team (2010)) package **np** (Hayfield and Racine (2008)) by providing a brief overview of each approach and its implementation followed by brief illustrative examples. A simple demonstration outlining how practitioners can implement their own entropy functions is also provided.

*Keywords*: nonparametric, semiparametric, kernel smoothing, categorical data..

## 1. Overview

In versions 0.30-4 through 0.30-7 of the **np** package (Hayfield and Racine (2008)) we introduced a range of new functions that may be of interest to those conducting applied research. The five new functions are summarized in Table 1 below.

| Function | Description | Reference |
|---|---|---|
| `npdeneqtest` | Nonparametric Test for Equality of Densities | Li, Maasoumi, and Racine (2009) |
| `npdeptest` | Nonparametric Entropy Test for Pairwise Dependence | Maasoumi and Racine (2002) |
| `npsdeptest` | Nonparametric Entropy Test for Serial Nonlinear Dependence | Granger, Maasoumi, and Racine (2004) |
| `npsymtest` | Nonparametric Entropy Test for Asymmetry | Racine and Maasoumi (2007), Maasoumi and Racine (2009) |
| `npunitest` | Nonparametric Entropy Test for Univariate Density Equality | Maasoumi and Racine (2002) |

Table 1: New **np** functions introduced in versions 0.30-4 through 0.30-7.

In what follows we briefly describe each new function listed in Table 1 and provide illustrative examples of their use. Be aware that many of these functions rely on numerical integration and can be computationally demanding. Though we provide moment-based versions of the most computationally demanding functions, we advocate the use of the integral-based versions which are the default. In the last section of this overview we describe in detail how the user can implement their own custom entropy functions with minimal effort using the **np** package.

## 2. Testing Equality of Multivariate Densities

Li *et al.* (2009) proposed a nonparametric test for equality of multivariate densities comprised

of continuous and categorical data. This test can be accessed via the `npdeneqtest` function. Let $X$ and $Y$ be multivariate vectors of dimension $q + r$ where $q$ denotes the number of continuous variables and $r$ the number of discrete/categorical variables. A test statistic can be constructed based on the integrated squared density difference given by $I = \int [f(x) - g(x)]^2 dx = \int [f(x)dF(x) + g(x)dG(x) - f(x)dG(x) - g(x)dF(x)]$, where $F(\cdot)$ and $G(\cdot)$ are the cumulative distribution functions for $X$ and $Y$, respectively, and where $\int dx = \sum_{x^d \in \mathbb{S}^d} \int dx^c$. Replacing the first occurrences of $f(\cdot)$ and $g(\cdot)$ by their leave-one-out kernel estimates, and replacing $F(\cdot)$ and $G(\cdot)$ by their empirical distribution functions, we obtain the following test statistic,

$$I_n = \frac{1}{n_1(n_1-1)} \sum_{i=1}^{n_1} \sum_{j \neq i}^{n_1} K_{\gamma,x_i,x_j} + \frac{1}{n_2(n_2-1)} \sum_{i=1}^{n_2} \sum_{j \neq i}^{n_2} K_{\gamma,y_i,y_j} - \frac{2}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=i}^{n_2} K_{\gamma,x_i,y_j}.$$

Li *et al.* (2009) demonstrate that, under the null of equality,

$$T_n = (n_1 n_2 h_1 \dots h_q)^{1/2} I_n / \sigma_n \to N(0,1) \text{ in distribution,}$$

where

$$\sigma_n^2 = 2(n_1 n_2 h_1 \dots h_q) \left[ \frac{1}{n_1^2(n_1-1)^2} \sum_{i=1}^{n_1} \sum_{j \neq i}^{n_1} (K_{\gamma,x_i,x_j})^2 + \frac{1}{n_2^2(n_2-1)^2} \sum_{i=1}^{n_2} \sum_{j \neq i}^{n_2} (K_{\gamma,y_i,y_j})^2 + \frac{2}{n_1^2 n_2^2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} (K_{\gamma,x_i,y_j})^2 \right],$$

which is a consistent estimator of $\sigma_0^2 = 2[\delta^{-1} + \delta + 2][E[f(X_i)]][\int W^2(v)dv]$, the asymptotic variance of $(n_1 n_2 h_1 \dots h_q)^{1/2} I_n$, where $\delta = \lim_{\min\{n_1,n_2\} \to \infty} (n_1/n_2)$. Under the alternative the statistic diverges to $+\infty$, so the test is one-sided rejecting when the test statistic is sufficiently large.

The test that uses critical values taken from the asymptotic distribution displays finite-sample size distortions, so the `npdeneqtest` function employs bootstrap resampling to obtain the finite-sample distribution of the statistic (this provides a test having correct size). The bootstrap resamples are obtained by resampling from the empirical distribution of the pooled data (i.e. are drawn from a common multivariate distribution under the null). Bandwidths are obtained via likelihood cross-validation by default.

The following code snippet provides two simple examples using a mix of continuous and discrete data. Note that the two samples must be data frames with identically named variables (e.g., the variables 'wages' and 'experience' must be common to both data frames while sample A could be that for females, sample B for males).

```
R> set.seed(1234)
R> n <- 250
R> ## Distributions are equal
R>
R> sample.A <- data.frame(a=rnorm(n),b=factor(rbinom(n,2,.5)))
R> sample.B <- data.frame(a=rnorm(n),b=factor(rbinom(n,2,.5)))
R> npdeneqtest(sample.A,sample.B,boot.num=99)
```

```
Consistent Density Equality Test
99 Bootstrap Replications

Test Statistic 'Tn': 0.0809        P Value: 0.2
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Fail to reject the null of equality at the 10% level
```

```
R> ## Distributions are unequal
R>
R> sample.A <- data.frame(a=rnorm(n),b=factor(rbinom(n,2,.5)))
R> sample.B <- data.frame(a=rnorm(n,sd=10),b=factor(rbinom(n,2,.25)))
R> npdeneqtest(sample.A,sample.B,boot.num=99)
```

```
Consistent Density Equality Test
99 Bootstrap Replications

Test Statistic 'Tn': 50.4        P Value: <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Null of equality is rejected at the 0.1% level
```

## 3. Testing Equality of Univariate Densities

Maasoumi and Racine (2002) consider a metric entropy useful for testing for equality of densities for two univariate random variables $X$ and $Y$. The function `npunitest` computes the nonparametric metric entropy (normalized Hellinger of Granger *et al.* (2004)) for testing the null of equality of two univariate density (or probability) functions. For continuous variables we construct

$$\begin{aligned} S_\rho &= \frac{1}{2}\int \left(f_1^{1/2} - f_2^{1/2}\right)^2 dx \\ &= \frac{1}{2}\int \left(1 - \frac{f_2^{1/2}}{f_1^{1/2}}\right)^2 dF_1(x), \end{aligned}$$

where $f_1 = f(x)$ and $f_2 = f(y)$ are the marginal densities of the random variables $X$ and $Y$. The second expression is in a moment form which is often replaced with a sample average, especially for theoretical developments. When $X$ and $Y$ are discrete/categorical, we replace integration with the sum over all possible outcomes. The unknown density/probability functions are replaced with nonparametric kernel estimates.

The bootstrap is conducted by resampling with replacement from the pooled empirical distribution of $X$ and $Y$ ($X$ only for the moment version). Default bandwidths are of the plug-in variety ('bw.SJ' for continuous variables and direct plug-in for discrete variables).

The following code snippet provides three simple examples for both continuous and discrete data.

```
R> set.seed(1234)
R> n <- 1000
R> ## Compute the statistic only, different distributions
R>
R> x <- rchisq(n,df=10)
R> y <- rnorm(n,sd=10000,mean=-10000)
R> npunitest(x,y,bootstrap=FALSE)

Consistent Univariate Density Difference Metric Entropy

Metric Entropy 'Srho': 0.977

R> ## Data drawn from same continuous distribution
R>
R> x <- rnorm(n)
R> y <- rnorm(n)
R> npunitest(x,y,boot.num=99)

Consistent Univariate Entropy Density Equality Test
99 Bootstrap Replications

Test Statistic 'Srho': 0.0016        P Value: 0.5
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Fail to reject the null of equality at the 10% level

R> ## Data drawn from different continuous distributions having the
R> ## same mean and variance
R>
R> x <- rchisq(n,df=5)
R> y <- rnorm(n,mean=5,sd=sqrt(10))
R> npunitest(x,y,boot.num=99)

Consistent Univariate Entropy Density Equality Test
99 Bootstrap Replications

Test Statistic 'Srho': 0.0373        P Value: <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Null of equality is rejected at the 0.1% level

R> ## Data drawn from different discrete distributions
R>
R> x <- factor(rbinom(n,2,.5))
R> y <- factor(rbinom(n,2,.1))
R> npunitest(x,y,boot.num=99)
```

```
Consistent Univariate Entropy Density Equality Test
99 Bootstrap Replications

Test Statistic 'Srho': 0.222        P Value: <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Null of equality is rejected at the 0.1% level
```

## 4. Testing Univariate Asymmetry

Consider a (strictly) stationary series $\{Y_t\}_{t=1}^T$. Let $\mu_y$ denote a measure of central tendency, say $\mu_y = E[Y_t]$, let $f(y)$ denote the density function of the random variable $Y_t$, let $\tilde{Y}_t = -Y_t + 2\mu_y$ denote a rotation of $Y_t$ about its mean, and let $f(\tilde{y})$ denote the density function of the random variable $\tilde{Y}_t$. Note that if $\mu_y = 0$ then $\tilde{Y}_t = -Y_t$, though in general this will not be so.

We say a series is *symmetric about the mean* (median, mode) if $f(y) \equiv f(\tilde{y})$ almost surely. Tests for asymmetry about the mean therefore naturally involve testing the following null:

$$H_0 : f(y) = f(\tilde{y}) \text{ almost everywhere (a.e.)}$$

against the alternative:

$$H_1 : f(y) \neq f(\tilde{y}) \text{ on a set with positive measure.}$$

The function `npsymtest` computes the nonparametric metric entropy (normalized Hellinger of Granger *et al.* (2004)) outlined in Maasoumi and Racine (2009) for testing the null of symmetry using the densities/probabilities of the data and the rotated data, $f(y)$ and $f(\tilde{y})$, respectively, and in Racine and Maasoumi (2007). $Y$ must be univariate and can be a time series, continuous, or even categorical valued so long as the outcomes are not character strings.

For bootstrapping the null distribution of the statistic, 'iid' conducts simple random resampling, while 'geom' conducts stationary bootstrapping using automatic block length selection via the 'b.star' function in the 'np' package (Politis and Romano (1994), Politis and White (2004), Patton, Politis, and White (2009)). Bootstrapping is conducted by resampling from the empirical distribution of the pooled data and rotated data. Default bandwidths are of the plug-in variety ('bw.SJ' for continuous variables and direct plug-in for discrete variables).

For continuous variables we use

$$\begin{aligned} S_\rho &= \frac{1}{2}\int \left(f_1^{1/2} - f_2^{1/2}\right)^2 dx \\ &= \frac{1}{2}\int \left(1 - \frac{f_2^{1/2}}{f_1^{1/2}}\right)^2 dF_1(x), \end{aligned}$$

where $f_1$ and $f_2$ are the marginal densities of the data and rotated data, respectively. The second expression is in a moment form which is often replaced with a sample average, especially for theoretical developments. When $Y$ is discrete/categorical, we replace integration with the sum over all possible outcomes.

The following code snippet provides two simple examples for both continuous and discrete data.

```
R> set.seed(1234)
R> n <- 100
R> ## Asymmetric discrete probability distribution
R>
R> x <- factor(rbinom(n,2,.8))
R> npsymtest(x,boot.num=99)

Consistent Entropy Asymmetry Test
99 Bootstrap Replications

Test Statistic 'Srho': 0.522        P Value: <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Null of symmetry is rejected at the 0.1% level

R> ## Symmetric continuous distribution
R>
R> y <- rnorm(n)
R> npsymtest(y,boot.num=99)

Consistent Entropy Asymmetry Test
99 Bootstrap Replications

Test Statistic 'Srho': 0.00726        P Value: 0.2
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Fail to reject the null of symmetry at the 10% level
```

## 5. Testing Nonlinear Pairwise Independence

Maasoumi and Racine (2002) consider a metric entropy useful for testing for pairwise independence of two random variables $X$ and $Y$. The function `npdeptest` computes the nonparametric metric entropy (normalized Hellinger of Granger *et al.* (2004)) for testing the null of pairwise independence of two univariate density (or probability) functions. For continuous variables we construct

$$\begin{aligned} S_\rho &= \frac{1}{2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left(f_1^{1/2} - f_2^{1/2}\right)^2 dx\, dy \\ &= \frac{1}{2}\int\int\left(1 - \frac{f_2^{1/2}}{f_1^{1/2}}\right)^2 dF_1(x,y), \end{aligned}$$

where $f_1 = f(x_i, y_i)$ is the joint density and $f_2 = g(x_i) \times h(y_i)$ is the product of the marginal densities of the random variables $X_i$ and $Y_i$. The unknown density/probability functions are replaced with nonparametric kernel estimates.

The bootstrap distribution is obtained by resampling with replacement from the empirical distribution of $X$ delivering $\{X_i, Y_i\}$ pairs under the null generated as $\{X_i^*, Y_i\}$ where $X^*$ is the bootstrap resample (i.e. we 'shuffle' $X$ leaving $Y$ unchanged thereby breaking any pairwise dependence to generate resamples under the null). Bandwidths are obtained via likelihood cross-validation by default for the marginal and joint densities.

Examples include, (a) a measure/test of "fit", for in-sample values of a variable $y$ and its fitted values, $\hat{y}$, and (b) a measure of "predictability" for a variable $y$ and its predicted values $\hat{y}$ (from a user implemented model).

The following code snippet provides a simple example using the actual and fitted values from a regression model. Note that we strongly advocate the use of the integration (default) version of the statistic in applied settings but use the summation (i.e. moment) version below purely by way of demonstration as it is computationally faster.

```
R> set.seed(123)
R> ## Test/measure lack of fit between y and its fitted value from a
R> ## regression model when x is relevant.
R> n <- 100
R> x <- rnorm(n)
R> y <- 1 + x + rnorm(n)
R> model <- lm(y~x)
R> y.fit <- fitted(model)
R> npdeptest(y,y.fit,boot.num=99,method="summation")

Consistent Metric Entropy Test for Dependence
99 Bootstrap Replications

Test Statistic 'Srho': 0.028        P Value: <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Null of independence is rejected at the 0.1% level
```

## 6. Testing Nonlinear Serial Independence

Granger *et al.* (2004) consider a metric entropy useful for testing for nonlinear serial independence in a univariate random variable $Y$. The function `npsdeptest` computes the nonparametric metric entropy (normalized Hellinger) for testing the null of nonlinear serial independence of such a series. We construct

$$\begin{aligned} S_\rho &= \frac{1}{2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left(f_1^{1/2} - f_2^{1/2}\right)^2 dx\, dy \\ &= \frac{1}{2}\int\int\left(1 - \frac{f_2^{1/2}}{f_1^{1/2}}\right)^2 dF_1(x,y), \end{aligned}$$

where $f_1 = f(y_t, y_{t-k})$ is the joint density and $f_2 = g(y_t) \times h(y_{t-k})$ is the product of the marginal densities of the random variables $Y_t$ and $Y_{t-k}$. The second expression is in a moment

form which is often replaced with a sample average, especially for theoretical developments. The unknown density/probability functions are replaced with nonparametric kernel estimates.

The bootstrap distribution is obtained by resampling with replacement from the empirical distribution of $Y_t$ delivering $Y_t^*$ under the null of nonlinear serial independence. Bandwidths are obtained via likelihood cross-validation by default for the marginal and joint densities.

The following code snippet provides a simple example for a continuous time series. Note that we strongly advocate the use of the integration (default) version of the statistic in applied settings but use the summation (i.e. moment) version below purely by way of demonstration as it is computationally faster.

```
R> set.seed(123)
R> ## A function to create a time series
R> ar.series <- function(phi,epsilon) {
+   n <- length(epsilon)
+   series <- numeric(n)
+   series[1] <- epsilon[1]/(1-phi)
+   for(i in 2:n) {
+     series[i] <- phi*series[i-1] + epsilon[i]
+   }
+   return(series)
+ }
R> n <- 100
R> ## Stationary persistent time-series
R> yt <- ar.series(0.95,rnorm(n))
R> npsdeptest(yt,lag.num=2,boot.num=99,method="summation")

Consistent Metric Entropy Test for Nonlinear Dependence
99 Bootstrap Replications, 2 Lags

Test Statistic 'Srho[1]': 0.102        P Value: <2e-16 ***
Test Statistic 'Srho[2]': 0.0775        P Value: <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Null of independence is rejected at lag 1 at the 0.1% level
Null of independence is rejected at lag 2 at the 0.1% level
```

## 7. Rolling Your Own Entropy Function

The functions outlined above allow for a range of entropy-based tests where the underlying distributions are modelled nonparametrically. However, practitioners may wish to construct their own entropy measures that rely on nonparametric estimates of the underlying distributions. By way of example we consider a metric entropy that measures distance (Hellinger) of an unknown univariate density from a specific parametric density. The parametric density for this example will be Gaussian and the unknown density will be estimated via kernel methods.

In the following code snippet we construct a simple function in R that accomplishes this. You could of course use nonparametric methods for both distributions or parametric methods for both. The function `npudens` computes unconditional kernel density estimates, and `fitted` computes the fitted density for the sample realizations. The function `dnorm` provides the normal density function. The `integrate` function performs univariate numerical integration.

```
R> Srho <- function(x,y,...) {
+   ## First write a function to compute the integrand (this is fed to
+   ## the `integrate' function). This function's first argument is the
+   ## point at which the two densities are computed (the remaining
+   ## arguments are the data vectors).
+   integrand <- function(t,x,y) {
+     ## First, nonparametrically estimate the density of the x data
+     ## using a plug-in bandwidth and evaluate the density at the point
+     ## `t'.
+     f.x <- fitted(npudens(tdat=x,edat=t,bws=bw.SJ(x),...))
+     ## Next, estimate the parametric density of the data y using the
+     ## Gaussian distribution and evaluate the density at the point
+     ## `t'.
+     f.y <- dnorm(t,mean=mean(y),sd=sd(y))
+     ## Compute and return the integrand evaluated at the point `t'.
+     return(0.5*(sqrt(f.x)-sqrt(f.y))**2)
+   }
+   ## Feed the integrand function to integrate() and return the value
+   ## of the integral.
+   return(integrate(integrand,-Inf,Inf,x=x,y=y)$value)
+ }
R> set.seed(123)
R> n <- 1000
R> ## Data drawn from the same distribution
R> x <- rnorm(n)
R> y <- rnorm(n)
R> Srho(x,y)

[1] 0.000871

R> ## Data drawn from different distributions
R> y <- rnorm(n,sd=100)
R> Srho(x,y)

[1] 0.858
```

Should the reader be interested in multivariate measures, multivariate numerical integration is available in the R packages **cubature** and **R2Cuba** (Johnson and Narasimhan (2009), Hahn, Bouvier, and Kiêu (2010)).

## 8. Summary

The **np** package (Hayfield and Racine (2008)) contains new functionality introduced in versions 0.30-4 through 0.30-7 that implements a range of entropy-based inferential procedures. We hope you find these to be easy to use. Please report any issues encountered to racinej@mcmaster.ca. Any suggestions for improvements to this document would be greatly appreciated. As always, if you use these functions for your work we would sincerely appreciate your citing both the relevant published research papers and the **np** package (Hayfield and Racine (2008)).

## Acknowledgments

We would like to gratefully acknowledge support from the Natural Sciences and Engineering Research Council of Canada (https://nserc-crsng.canada.ca/), the Social Sciences and Humanities Research Council of Canada (https://sshrc-crsh.canada.ca/), and the Shared Hierarchical Academic Research Computing Network (https://www.sharcnet.ca/).

## References

Granger C, Maasoumi E, Racine JS (2004). "A Dependence Metric for Possibly Nonlinear Time Series." *Journal of Time Series Analysis*, **25**(5), 649–669.

Hahn T, Bouvier A, Kiêu K (2010). *R2Cuba: Multidimensional Numerical Integration*. R package version 1.0-4, URL https://CRAN.R-project.org/package=R2Cuba.

Hayfield T, Racine JS (2008). "Nonparametric Econometrics: The np Package." *Journal of Statistical Software*, **27**(5), 1–32. doi:10.18637/jss.v027.i05. URL https://www.jstatsoft.org/index.php/jss/article/view/v027i05.

Johnson SG, Narasimhan B (2009). *cubature: Adaptive multivariate integration over hypercubes*. R package version 1.0, URL https://CRAN.R-project.org/package=cubature.

Li Q, Maasoumi E, Racine JS (2009). "A Nonparametric Test for Equality of Distributions with Mixed Categorical and Continuous Data." *Journal of Econometrics*, **148**, 186–200.

Maasoumi E, Racine JS (2002). "Entropy and Predictability of Stock Market Returns." *Journal of Econometrics*, **107**(2), 291–312.

Maasoumi E, Racine JS (2009). "A robust entropy-based test of asymmetry for discrete and continuous processes." *Econometric Reviews*, **28**, 246–261.

Patton A, Politis DN, White H (2009). "CORRECTION TO "Automatic block-length selection for the dependent bootstrap" by D. Politis and H. White." *Econometric Reviews*, **28**(4), 372–375.

Politis DN, Romano JP (1994). "Limit theorems for weakly dependent Hilbert space valued random variables with applications to the stationary bootstrap." *Statistica Sinica*, **4**, 461–476.

Politis DN, White H (2004). "Automatic block-length selection for the dependent bootstrap." *Econometric Reviews*, **23**, 53–70.

R Development Core Team (2010). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL https://www.R-project.org.

Racine J, Maasoumi E (2007). "A Versatile and Robust Metric Entropy Test of Time Reversibility and Dependence." *Journal of Econometrics*, **138**, 547–567.

## Affiliation:

Jeffrey S. Racine
Department of Economics
McMaster University
Hamilton, Ontario, Canada, L8S 4L8
E-mail: racinej@mcmaster.ca
URL: https://experts.mcmaster.ca/people/racinej