Package ‘OwenQ’

January 8, 2020

Type Package

Title Owen Q-Function

Version 1.0.3

Date 2019-12-21

Author Stéphane Laurent

Maintainer Stéphane Laurent <laurent_step@outlook.fr>

Description Evaluates the Owen Q-function for an integer value of the degrees of freedom, by applying Owen’s algorithm (1965) <doi:10.1093/biomet/52.3-4.437>

It is useful for the calculation of the power of equivalence tests.

License GPL-3

Imports Rcpp (>= 0.12.10), stats

LinkingTo Rcpp, BH, RcppNumerical, RcppEigen

Suggests testthat, knitr, rmarkdown, mvtnorm

Encoding UTF-8

LazyData true

SystemRequirements C++11

RoxygenNote 7.0.2

VignetteBuilder knitr

URL https://github.com/stla/OwenQ

BugReports https://github.com/stla/OwenQ/issues

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-01-08 09:20:07 UTC
Description

Evaluates the first Owen Q-function (integral from 0 to $R$) for an integer value of the degrees of freedom.

Usage

OwenQ1(nu, t, delta, R, algo = 2)

Arguments

nu integer greater than 1, the number of degrees of freedom
t number, positive or negative, possibly infinite
delta vector of finite numbers, with the same length as R
R (upper bound of the integral) vector of finite positive numbers, with the same length as delta
algo the algorithm, 1 or 2

Value

A vector of numbers between 0 and 1, the values of the integral from 0 to $R$.

Note

When the number of degrees of freedom is odd, the procedure resorts to the Owen T-function (OwenT).

References

OwenQ2

Examples

# As R goes to Inf, OwenQ1(nu, t, delta, R) goes to pt(t, nu, delta):
OwenQ1(nu=5, t=3, delta=2, R=100)
pt(q=3, df=5, ncp=2)

<table>
<thead>
<tr>
<th>OwenQ2</th>
<th>Second Owen Q-function</th>
</tr>
</thead>
</table>

Description

Evaluates the second Owen Q-function (integral from R to ∞) for an integer value of the degrees of freedom.

Usage

OwenQ2(nu, t, delta, R, algo = 2)

Arguments

nu integer greater than 1, the number of degrees of freedom

 t number, positive or negative, possibly infinite

delta vector of finite numbers, with the same length as R

 R (lower bound of the integral) vector of finite positive numbers, with the same length as delta

algo the algorithm used, 1 or 2

Value

A vector of numbers between 0 and 1, the values of the integral from R to ∞.

Note

When the number of degrees of freedom is odd, the procedure resorts to the Owen T-function (OwenT).

References


Examples

# OwenQ1(nu, t, delta, R) + OwenQ2(nu, t, delta, R) equals pt(t, nu, delta):
OwenQ1(nu=5, t=3, delta=2, R=1) + OwenQ2(nu=5, t=3, delta=2, R=1)
pt(q=3, df=5, ncp=2)
OwenT  \textit{Owen T-function}

\textbf{Description}
Evaluates the Owen T-function.

\textbf{Usage}
OwenT(h, a)

\textbf{Arguments}
\begin{itemize}
  \item \texttt{h} numeric scalar
  \item \texttt{a} numeric scalar
\end{itemize}

\textbf{Details}
This is a port of the function \texttt{owens_t} of the \texttt{boost} collection of C++ libraries.

\textbf{Value}
A number between 0 and 0.25.

\textbf{References}

\textbf{Examples}
\begin{verbatim}
  integrate(function(x) pnorm(1+2*x)^2*dnorm(x), lower=-Inf, upper=Inf)
  pnorm(1/sqrt(5)) - 2*OwenT(1/sqrt(5), 1/3)
\end{verbatim}

\textbf{powen}  \textit{Owen distribution functions when }\delta_1 > \delta_2\textbf{ when }\delta_1 > \delta_2\textbf{ and the number of degrees of freedom is integer.}

\begin{itemize}
  \item \texttt{powen1} evaluates $P(T_1 \leq t_1, T_2 \leq t_2)$ (Owen’s equality 8)
  \item \texttt{powen2} evaluates $P(T_1 \leq t_1, T_2 > t_2)$ (Owen’s equality 9)
  \item \texttt{powen3} evaluates $P(T_1 > t_1, T_2 > t_2)$ (Owen’s equality 10)
  \item \texttt{powen4} evaluates $P(T_1 > t_1, T_2 \leq t_2)$ (Owen’s equality 11)
\end{itemize}
Usage

powen1(nu, t1, t2, delta1, delta2, algo = 2)
powen2(nu, t1, t2, delta1, delta2, algo = 2)
powen3(nu, t1, t2, delta1, delta2, algo = 2)
powen4(nu, t1, t2, delta1, delta2, algo = 2)

Arguments

nu integer greater than 1, the number of degrees of freedom; infinite allowed
t1, t2 two numbers, positive or negative, possible infinite
delta1, delta2 two vectors of possibly infinite numbers with the same length, the noncentrality
er parameters; must satisfy delta1>delta2
algo the algorithm used, 1 or 2

Value

A vector of numbers between 0 and 1, possibly containing some NaN.

Note

When the number of degrees of freedom is odd, the procedure resorts to the Owen T-function
(OwenT).

References


See Also

Use psbt for general values of delta1 and delta2.

Examples

nu=5; t1=2; t2=1; delta1=3; delta2=2
# Wolfram integration gives 0.1394458271284726
( p1 <- powen1(nu, t1, t2, delta1, delta2) )
# Wolfram integration gives 0.0353568969628651
( p2 <- powen2(nu, t1, t2, delta1, delta2) )
# Wolfram integration gives 0.806507459306199
( p3 <- powen3(nu, t1, t2, delta1, delta2) )
# Wolfram integration gives 0.018689824158
( p4 <- powen4(nu, t1, t2, delta1, delta2) )
# the sum should be 1
p1+p2+p3+p4
Description

Evaluates the Owen cumulative distribution function for an integer number of degrees of freedom.

- psbt1 evaluates $P(T_1 \leq t_1, T_2 \leq t_2)$
- psbt2 evaluates $P(T_1 \leq t_1, T_2 > t_2)$
- psbt3 evaluates $P(T_1 > t_1, T_2 > t_2)$
- psbt4 evaluates $P(T_1 > t_1, T_2 \leq t_2)$

Usage

psbt1(nu, t1, t2, delta1, delta2, algo = 2)
psbt2(nu, t1, t2, delta1, delta2, algo = 2)
psbt3(nu, t1, t2, delta1, delta2, algo = 2)
psbt4(nu, t1, t2, delta1, delta2, algo = 2)

Arguments

nu integer greater than 1, the number of degrees of freedom; infinite allowed

$\text{t1, t2}$ two numbers, positive or negative, possibly infinite

$\text{delta1, delta2}$ two vectors of possibly infinite numbers with the same length, the noncentrality parameters

algo the algorithm used, 1 or 2

Value

A vector of numbers between 0 and 1, possibly containing some NaN.

Note

When the number of degrees of freedom is odd, the procedure resorts to the Owen T-function ($\text{OwenT}$).

References


See Also

It is better to use powen if delta1>delta2.
Examples

\[
\begin{align*}
&\text{nu}=5; \ t_1=1; \ t_2=2; \delta_1=2; \ \delta_2=3 \\
&(p_1 \leftarrow \text{psbt1}(\text{nu}, t_1, t_2, \delta_1, \delta_2)) \\
&(p_2 \leftarrow \text{psbt2}(\text{nu}, t_1, t_2, \delta_1, \delta_2)) \\
&(p_3 \leftarrow \text{psbt3}(\text{nu}, t_1, t_2, \delta_1, \delta_2)) \\
&(p_4 \leftarrow \text{psbt4}(\text{nu}, t_1, t_2, \delta_1, \delta_2)) \\
&\# \text{ the sum should be 1} \\
&p_1+p_2+p_3+p_4
\end{align*}
\]

\[
\begin{array}{ll}
\text{ptOwen} & \text{Student CDF with integer number of degrees of freedom} \\
\end{array}
\]

Description

Cumulative distribution function of the noncentral Student distribution with an integer number of degrees of freedom.

Usage

\[
\text{ptOwen}(q, \text{nu}, \text{delta} = 0)
\]

Arguments

- \(q\) \quad \text{quantile, a finite number}
- \(\text{nu}\) \quad \text{integer greater than 1, the number of degrees of freedom; possibly infinite}
- \(\text{delta}\) \quad \text{numeric vector of noncentrality parameters; possibly infinite}

Value

Numeric vector, the CDF evaluated at \(q\).

Note

The results are theoretically exact when the number of degrees of freedom is even. When odd, the procedure resorts to the Owen T-function.

References


Examples

\[
\begin{align*}
&\text{ptOwen}(2, 3) - \text{pt}(2, 3) \\
&\text{ptOwen}(2, 3, \text{delta}=1) - \text{pt}(2, 3, \text{ncp}=1)
\end{align*}
\]
Description

Evaluation of the second Owen distribution function in a special case (see details).

Usage

spowen2(nu, t, delta, algo = 2)

Arguments

- nu: positive integer, possibly infinite
- t: positive number
- delta: vector of positive numbers
- algo: the algorithm used, 1 or 2

Details

The value of spowen2(nu, t, delta) is the same as the value of powen2(nu, t, -t, delta, -delta), but it is evaluated more efficiently.

Value

A vector of numbers between 0 and 1.

See Also

powen2

Examples

spowen2(4, 1, 2) == powen2(4, 1, -1, 2, -2)
Index

OwenQ1, 2
OwenQ2, 3
OwenT, 2, 3, 4, 5, 6
powen, 4, 6
powen1 (powen), 4
powen2, 8
powen2 (powen), 4
powen3 (powen), 4
powen4 (powen), 4
psbt, 5, 6
psbt1 (psbt), 6
psbt2 (psbt), 6
psbt3 (psbt), 6
psbt4 (psbt), 6
ptOwen, 7
spowen2, 8