Package ‘mvnormalTest’
April 28, 2020

Type Package
Title Powerful Tests for Multivariate Normality
Version 1.0.0
Author Yian Zhang [aut, cre],
Ming Zhou [aut],
Yongzhao Shao [aut]
Maintainer Yian Zhang <yz2777@nyu.edu>
Description A simple informative powerful test (mvnTest()) for multivariate normality proposed by
Zhou and Shao (2014) <doi:10.1080/02664763.2013.839637>, which combines kurtosis
with Shapiro-Wilk test that is easy for biomedical researchers to understand and
easy to implement in all dimensions. This package also contains some other multivariate
normality tests including Fattorini’s FA test (faTest()), Mardia’s skewness and kurtosis
test (mardia()), Henze-Zirkler's test (mhz()), Bowman and Shenton's test (msk()),
Royston’s H test (msw()), and Villasenor-Alva and Gonzalez-Estrada’s test (msw()). Empirical
power calculation functions for these tests are also provided. In addition, this package
includes some functions to generate several types of multivariate distributions mentioned in
Zhou and Shao (2014).
License GPL (>= 2)
Encoding UTF-8
Imports stats, nortest, moments, copula
Suggests knitr
Depends R (>= 3.5.0)
Language en-US
RoxygenNote 7.1.0
NeedsCompilation no
Repository CRAN
Date/Publication 2020-04-28 09:10:02 UTC

R topics documented:
copulas ................................................................. 2
Random Generation for the Copula Generated Distributions

Description

Generate univariate or multivariate random sample for the Copula Generated Distributions.

Usage

copulas(n, p, c = "clayton", param, invF, ...)

Arguments

- **n**: number of rows (observations).
- **p**: total number of columns (variables).
- **c**: name of an Archimedean copula, choosing from "clayton" (default), "frank", or "gumbel".
- **param**: number (numeric) specifying the copula parameter.
- **invF**: inverse function (quantile function, e.g. qnorm).
- **...**: optional arguments passed to invF.

Value

univariate ($p = 1$) or multivariate ($p > 1$) random sample.
faTest

References

Examples
set.seed(12345)

## Generate 5X2 random sample matrix from Clayton(0.5, qnorm) ##
copulas(n=50, p=2, c="clayton", param=0.5, invF=qnorm)

## Power calculation against bivariate (p=2) Clayton(0.5, qnorm) distribution ##
## at sample size n=50 at one-sided alpha = 0.05 ##

# Zhou-Shao's test #
power.mvnTest(a=0.05, n=50, p=2, B=100, FUN=copulas, c="clayton", param=0.5, invF=qnorm)

Description
faTest

It computes FA Test proposed by Fattorini (1986). This test would be more rotationally robust than other SWT tests such as Royston (1982) H test and the test proposed by Villasenor-Alva and Gonzalez-Estrada (2009). The p-value of the test statistic is computed based on a simulated null distribution of the statistic.

Usage
faTest(X, B = 1000)

Arguments

X  an n * p data matrix or data frame, where n is number of rows (observations) and p is number of columns (variables) and n > p.

B  number of Monte Carlo simulations for null distribution, default is 1000 (increase B to increase the precision of p-value).

Value
Returns a list with two objects:

mv.test  results of the FA test for multivariate normality, i.e., test statistic, p-value, and multivariate normality summary (YES, if p-value>0.05).

uv.shapiro  a dataframe with p rows detailing univariate Shapiro-Wilk tests. Columns in the dataframe contain test statistics W, p-value, and univariate normality summary (YES, if p-value>0.05).
References


See Also

`power.faTest`, `mvnTest`, `msk`, `mardia`, `msw`, `mhz`

Examples

```r
set.seed(12345)

## Data from gamma distribution ##
X = matrix(rgamma(50*4,shape = 2),50)
faTest(X, B=100)

## load the ubiquitous multivariate iris data ##
## (first 50 observations of columns 1:4) ##
iris.df = iris[1:50, 1:4]
faTest(iris.df, B=100)
```

---

**IMMV**

*Random Generation for Distribution with Independent Marginals*

**Description**

Generate univariate or multivariate random sample for distribution with independent marginals such that $D_1 \otimes D_2$. $D_1 \otimes D_2$ denotes the distribution having independent marginal distributions $D_1$ and $D_2$. This function can generate multivariate random samples only from distribution $D_1$ or from both $D_1$ and $D_2$.

**Usage**

```r
IMMV(n, p, q = NULL, D1, D2 = NULL, D1.args = list(), D2.args = list())
```
Arguments

n  number of rows (observations).
p  total number of columns (variables).
q  number of columns from distribution $D_1$ if generate multivariate samples from
   independent marginal distribution $D_1$ and $D_2$. Default is NULL, i.e., generating
   samples only from one distribution.

$D_1$  random generation function for 1st distribution (e.g., `rnorm`, `rbeta`).
$D_2$  random generation function for 2nd distribution (e.g., `rnorm`, `rbeta`).

$D_1$.args  a list of optional arguments passed to $D_1$.
$D_2$.args  a list of optional arguments passed to $D_2$.

Value

Returns univariate ($p = 1$) or multivariate ($p > 1$) random sample matrix.

References


Examples

```r
set.seed(12345)

## Generate 5X2 random sample matrix from IMMV(N(0,1),Beta(1,2)) ##
IMMV(n=5, p=2, q=1, D1=rbeta, D1.args=list(shape1=1,shape2=2), D2=rnorm)

## Power calculation against bivariate (p=2) IMMV(Gamma(5,1)) distribution ##
## at sample size n=50 at one-sided alpha = 0.05 ##
# Zhou-Shao's test #
power.mvnTest(a=0.05, n=50, p=2, B=100, FUN=IMMV, D1=rgamma, D1.args=list(shape=5, rate=1))

## Power calculation against bivariate (p=2) IMMV(N(0,1),Beta(1,2)) distribution ##
## at sample size n=50 at one-sided alpha = 0.05 ##
# Zhou-Shao's test #
power.mvnTest(a=0.05, n=50, p=2, B=100, FUN=IMMV, q=1, D1=rbeta, D1.args=list(shape1=1,shape2=2),
D2=rnorm)
```
**mardia**  
*Mardia Test (Skewness and Kurtosis) for Multivariate Normality*

**Description**

It computes Mardia (1970)’s multivariate skewness and kurtosis statistics and their corresponding p-value. Both p-values of skewness and kurtosis statistics should be greater than 0.05 to conclude multivariate normality. The skewness statistic will be adjusted for sample size \(n < 20\).

**Usage**

```r
mardia(X, std = TRUE)
```

**Arguments**

- **X**: an \(n \times p\) numeric matrix or data frame.
- **std**: if TRUE, the data matrix or data frame will be standardized via normalizing the covariance matrix by \(n\).

**Value**

Returns a list with two objects:

- **mv.test**: results of the Mardia test, i.e., test statistic, \(p\)-value, and multivariate normality summary (YES, if both skewness and kurtosis \(p\)-value>0.05).
- **uv.shapiro**: a dataframe with \(p\) rows detailing univariate Shapiro-Wilk tests. Columns in the dataframe contain test statistics \(W\), \(p\)-value, and univariate normality summary (YES, if \(p\)-value>0.05).

**References**


**See Also**

`mvnTest, faTest, msw, msk, mhz, mvn`
Examples

```r
set.seed(12345)

## Data from gamma distribution
X = matrix(rgamma(50*4,shape = 2),50)
mardia(X)

## Data from normal distribution
X = matrix(rnorm(50*4,mean = 2 , sd = 1),50)
mardia(X)

## load the ubiquitous multivariate iris data ##
## (first 50 observations of columns 1:4) ##
iris.df = iris[1:50, 1:4]
mardia(iris.df)
```

---

### Henze-Zirkler Test for Multivariate Normality

**Description**

It computes a multivariate normality test based on a non-negative functional distance which was proposed by Henze and Zirkler (1990). Under the null hypothesis the test statistic is approximately log-normally distributed.

**Usage**

```r
mhz(X)
```

**Arguments**

- `X` an `n * p` numeric matrix or data frame.

**Value**

Returns a list with two objects:

- `mv.test` results of the Henze-Zirkler test, i.e., test statistic, `p`-value, and multivariate normality summary (YES, if `p`-value>0.05).
- `uv.shapiro` a dataframe with `p` rows detailing univariate Shapiro-Wilk tests. Columns in the dataframe contain test statistics `W`, `p`-value, and univariate normality summary (YES, if `p`-value>0.05).
References


See Also

*power.mhz*, *mvnTest*, *faTest*, *msw*, *msk*, *mardia*, *mvn*

Examples

```r
set.seed(12345)

## Data from gamma distribution
X = matrix(rgamma(50*4,shape = 2),50)
mhz(X)

## Data from normal distribution
X = matrix(rnorm(50*4,mean = 2 , sd = 1),50)
mhz(X)

## load the ubiquitous multivariate iris data ##
## (first 50 observations of columns 1:4) ##
iris.df = iris[1:50, 1:4]
mhz(iris.df)
```

**msk**

*Bowman and Shenton Test for Multivariate Normality*

Description

It computes Bowman and Shenton (1975)'s test statistic (MSK) and its corresponding p-value for multivariate normality. The statistic is calculated based on a combination of multivariate skewness (MS) and kurtosis (MK) such that $MSK = MS + |MK|^2$. For formulas of MS and MK, please refer to Mardia (1970). The corresponding p-value of the statistic is computed based on a simulated null distribution of MSK. The skewness statistic (MS) will be adjusted for sample size $n < 20$.

Usage

`msk(X, B = 1000)`

Arguments

- `X`: an $n \times p$ numeric matrix or data frame.
- `B`: number of Monte Carlo simulations for null distribution, default is 1000 (increase B to increase the precision of p-value).
Value

Returns a list with two objects:

mv.test results of the Bowman and Shenton test, i.e., test statistic, p-value, and multivariate normality summary (YES, if p-value>0.05).

uv.shapiro a dataframe with p rows detailing univariate Shapiro-Wilk tests. Columns in the dataframe contain test statistics W, p-value, and univariate normality summary (YES, if p-value>0.05).

References

Bowman, K. O., & Shenton, L. R. (1975). Omnibus test contours for departures from normality based on $\sqrt{b_1}$ and $b_2$. *Biometrika*, 62(2), 243-250.


See Also

power.msk, mvnTest, faTest, msw, mardia, mhz, mvn

Examples

```r
set.seed(12345)

## Data from gamma distribution
X = matrix(rgamma(50*4, shape = 2), 50)
msk(X, B=100)

## load the ubiquitous multivariate iris data ##
## (first 50 observations of columns 1:4) ##
iris.df = iris[1:50, 1:4]
msk(iris.df, B=100)
```

---

### msw

**Shapiro-Wilk Type (SWT) Tests for Multivariate Normality**

**Description**

The SWT-based tests for multivariate normality including Royston’s H test and the test proposed by Villasenor-Alva and Gonzalez-Estrada (2009).
Usage

msw(X)

Arguments

X an n * p numeric matrix or data frame, the number of n must be between 3 and 5000, n>p.

Value

Returns a list with two objects:

mv.test a result table of multivariate normality tests, including the name of the test, test statistic, p-value, and multivariate normality summary (Yes, if p-value>0.05). Note that the test results of Royston will not be reported if n > 2000 or n < 3 and the test results of Villasenor-Alva and Gonzalez-Estrada (VAGE) will not be reported if n > 5000 or n < 12.

uv.shapiro a dataframe with p rows detailing univariate Shapiro-Wilk tests. Columns in the dataframe contain test statistics W, p-value, and univariate normality summary (YES, if p-value>0.05). If the number of variable is p = 1, only univariate Shapiro-wilk’s test result will be produced.

References


See Also

power.mswR, power.mswV, mvnTest, faTest, msk, mardia, mhz, mvn, shapiro.test

Examples

set.seed(12345)

## Data from gamma distribution
X = matrix(rgamma(50*4,shape = 2),50)
msw(X)

## Data from normal distribution
X = matrix(rnorm(50*4,mean = 2, sd = 1),50)
msw(X)

## load the ubiquitous multivariate iris data ##
## MVNMIX

**Random Generation for the Normal Mixture Distribution**

### Description

Generate univariate or multivariate random sample for the normal mixture distribution with density $\lambda N(0, \sum_1) + (1 - \lambda) N(bl, \sum_2)$, where $l$ is the column vector with all elements being 1, $\sum_i = (1 - \rho_i)I + \rho_i ll^T$ for $i = 1, 2$. $\rho$ has to satisfy $\rho > -1/(p-1)$ in order to make the covariance matrix meaningful.

### Usage

```r
MVNMIX(n, p, lambda, mu2, rho1 = 0, rho2 = 0)
```

### Arguments

- `n`: number of rows (observations).
- `p`: total number of columns (variables).
- `lambda`: weight parameter to allocate the proportions of the mixture, $0 < \lambda < 1$.
- `mu2`: is $bl$ of $N(bl, \sum_2)$.
- `rho1`: parameter in $\sum_1$.
- `rho2`: parameter in $\sum_2$.

### Value

Returns univariate ($p = 1$) or multivariate ($p > 1$) random sample matrix.

### References


### Examples

```r
set.seed(12345)

## Generate 5X2 random sample matrix from MVNMIX(0.5,4,0,0) ##
MVNMIX(n=5, p=2, lambda=0.5, mu2=4, rho1=0, rho2=0)

## Power calculation against bivariate (p=2) MVNMIX(0.5,4,0,0) distribution ##
## at sample size n=50 at one-sided alpha = 0.05 ##
### Description

A simple and powerful test for multivariate normality with a combination of multivariate kurtosis (MK) and Shapiro-Wilk which was proposed by Zhou and Shao (2014). The p-value of the test statistic ($T_n$) is computed based on a simulated null distribution of $T_n$. Details see Zhou and Shao (2014).

### Usage

```r
mvnTest(X, B = 1000, pct = c(0.01, 0.99))
```

### Arguments

- **X**: an $n \times p$ data matrix or data frame, where $n$ is number of rows (observations) and $p$ is number of columns (variables) and $n > p$.
- **B**: number of Monte Carlo simulations for null distribution, default is 1000 (increase B to increase the precision of p-value).
- **pct**: percentiles of MK to get $c_1$ and $c_2$ described in the reference paper, default is (0.01, 0.99).

### Value

Returns a list with two objects:

- **mv.test** results of the Zhou-Shao’s test for multivariate normality, i.e., test statistic $T_n$, p-value (under H0, i.e. multivariate normal, that $T_n$ is at least as extreme as the observed value), and multivariate normality summary (YES, if p-value>0.05).
- **uv.shapiro** a dataframe with $p$ rows detailing univariate Shapiro-Wilk tests. Columns in the dataframe contain test statistics $W$, p-value, and univariate normality summary (YES, if p-value>0.05).

### References


power.faTest

See Also

power.mvnTest, msk, mardia.msw.faTest.mhz

Examples

set.seed(12345)

## Data from gamma distribution ##
X = matrix(rgamma(50*4,shape = 2),50)
mvnTest(X, B=100)

## load the ubiquitous multivariate iris data ##
## (first 50 observations of columns 1:4) ##
iris.df = iris[1:50, 1:4]
mvnTest(iris.df, B=100)

---

power.faTest  

Power Calculation using the Fattorini’s FA Test Statistic

Description

Empirical power calculation using the Fattorini’s FA Test Statistic.

Usage

power.faTest(a, n, p, B = 1000, FUN, ...)

Arguments

- **a**: significance level (α).
- **n**: number of rows (observations).
- **p**: number of columns (variables), \( n > p \).
- **B**: number of Monte Carlo simulations, default is 1000 (can increase B to increase the precision).
- **FUN**: self-defined function for generate multivariate distribution. See example.
- **...**: optional arguments passed to FUN.

Value

Returns a numeric value of the estimated empirical power (value between 0 and 1).

References

Examples

```
set.seed(12345)

## Power calculation against bivariate (p=2) independent Beta(1, 1) distribution ##
## at sample size n=50 at one-sided alpha = 0.05 ##

power.mhz(a = 0.05, n = 50, p = 2, B = 100, FUN=IMMV, D1=runif)
```

-----

power.mhz  

*Power Calculation using the Henze-Zirkler Test Statistic*

Description

Empirical power calculation using the Henze-Zirkler Test Statistic.

Usage

```
power.mhz(a, n, p, B = 1000, FUN, ...)```

Arguments

- **a**  
  significance level ($\alpha$).
- **n**  
  number of rows (observations).
- **p**  
  number of columns (variables), $n > p$.
- **B**  
  number of Monte Carlo simulations, default is 1000 (can increase $B$ to increase the precision).
- **FUN**  
  self-defined function for generate multivariate distribution. See example.
- **...**  
  optional arguments passed to FUN.

Value

Returns a numeric value of the estimated empirical power (value between 0 and 1).

References


Examples

```
set.seed(12345)

## Power calculation against bivariate (p=2) independent Beta(1, 1) distribution ##
## at sample size n=50 at one-sided alpha = 0.05 ##

power.mhz(a = 0.05, n = 50, p = 2, B = 100, FUN=IMMV, D1=runif)
```
Power Calculation using the Bowman and Shenton Test Statistic

Description

Empirical power calculation using Bowman and Shenton Test Statistic.

Usage

\[
power.msk(a, n, p, B = 1000, FUN, \ldots)
\]

Arguments

- \(a\) significance level (\(\alpha\)).
- \(n\) number of rows (observations).
- \(p\) number of columns (variables), \(n > p\).
- \(B\) number of Monte Carlo simulations, default is 1000 (can increase \(B\) to increase the precision).
- \(FUN\) self-defined function for generate multivariate distribution. See example.
- \(\ldots\) optional arguments passed to \(FUN\).

Value

Returns a numeric value of the estimated empirical power (value between 0 and 1).

References

Bowman, K. O., & Shenton, L. R. (1975). Omnibus test contours for departures from normality based on \(\sqrt{b_1}\) and \(b_2\). Biometrika, 62(2), 243-250.

Examples

```
set.seed(12345)

## Power calculation against bivariate (p=2) independent Beta(1, 1) distribution ##
## at sample size n=50 at one-sided alpha = 0.05 ##

power.msk(a = 0.05, n = 50, p = 2, B = 100, FUN=IMMV, D1=runif)
```
Description

Empirical power calculation using Royston test statistic.

Usage

\texttt{power.mswR(a, n, p, B = 1000, FUN, \ldots)}

Arguments

\begin{itemize}
\item \texttt{a} \quad \text{significance level (} \alpha \text{).}
\item \texttt{n} \quad \text{number of rows (observations).}
\item \texttt{p} \quad \text{number of columns (variables), } n > p.
\item \texttt{B} \quad \text{number of Monte Carlo simulations, default is 1000 (can increase B to increase the precision).}
\item \texttt{FUN} \quad \text{self-defined function for generate multivariate distribution. See example.}
\item \ldots \quad \text{optional arguments passed to FUN.}
\end{itemize}

Value

Returns a numeric value of the estimated empirical power (value between 0 and 1).

References


Examples

\begin{verbatim}
set.seed(12345)

## Power calculation against bivariate (p=2) independent Beta(1, 1) distribution ##
## at sample size n=50 at one-sided alpha = 0.05 ##

power.mswR(a = 0.05, n = 50, p = 2, B = 100, FUN=IMMV, D1=runif)
\end{verbatim}
Power Calculation using the SWT-based Villasenor-Alva and Gonzalez-Estrada (VAGE) Test Statistic

Description

Empirical power calculation using VAGE test statistic.

Usage

\[
\text{power.mswV}(a, n, p, B = 1000, \text{FUN}, \ldots)
\]

Arguments

- \(a\)  significance level (\(\alpha\)).
- \(n\)   number of rows (observations).
- \(p\)   number of columns (variables), \(n > p\).
- \(B\)   number of Monte Carlo simulations, default is 1000 (can increase \(B\) to increase the precision).
- \(\text{FUN}\)  self-defined function for generate multivariate distribution. See example.
- \(\ldots\)  optional arguments passed to \(\text{FUN}\).

Value

Returns a numeric value of the estimated empirical power (value between 0 and 1).

References


Examples

```r
set.seed(12345)

## Power calculation against bivariate (p=2) independent Beta(1, 1) distribution ##
## at sample size n=50 at one-sided alpha = 0.05 ##

power.mswV(a = 0.05, n = 50, p = 2, B = 100, FUN=IMMV, D1=runif)
```
power.mvnTest

Power Calculation using the Zhou-Shao’s Multivariate Normality Test Statistic ($T_n$)

Description

Empirical power calculation using the Zhou-Shao’s multivariate normality test Statistic $T_n$.

Usage

```r
power.mvnTest(a, n, p, B = 1000, pct = c(0.01, 0.99), FUN, ...)
```

Arguments

- **a**: significance level ($\alpha$).
- **n**: number of rows (observations).
- **p**: number of columns (variables), $n > p$.
- **B**: number of Monte Carlo simulations, default is 1000 (can increase B to increase the precision).
- **pct**: percentiles of MK to get $c_1$ and $c_2$ described in the reference paper, default is (0.01, 0.99).
- **FUN**: self-defined function for generate multivariate distribution. See example.
- **...**: optional arguments passed to FUN.

Value

Returns a numeric value of the estimated empirical power (value between 0 and 1).

References


Examples

```r
set.seed(12345)

## Power calculation against bivariate (p=2) independent Beta(1, 1) distribution ##
## at sample size n=50 for Tn at one-sided alpha = 0.05 ##

power.mvnTest(a = 0.05, n = 50, p = 2, B = 100, pct = c(0.01, 0.99), FUN=IMMV, D1=runif)
```
Description

Empirical power calculation using univariate Shapiro-Wilk test statistic.

Usage

\[ \text{power.usw}(a, n, p = 1, B = 1000, \text{FUN}, \ldots) \]

Arguments

- \( a \): significance level (\( \alpha \)).
- \( n \): number of rows (observations).
- \( p \): \( p=1 \) for univariate.
- \( B \): number of Monte Carlo simulations, default is 1000 (can increase \( B \) to increase the precision).
- \( \text{FUN} \): self-defined function for generate multivariate distribution. See example.
- \( \ldots \): optional arguments passed to \( \text{FUN} \).

Value

Returns a numeric value of the estimated empirical power (value between 0 and 1).

References


Examples

```r
set.seed(12345)

## Power calculation against univariate (p=1) independent Beta(1, 1) distribution ##
## at sample size n=50 at one-sided alpha = 0.05 ##

power.usw(a = 0.05, n = 50, p = 1, B = 100, FUN=IMMV, D1=runif)
```
PSII

Random Generation for the Spherically Symmetric Pearson Type II Distribution

Description

Generate univariate or multivariate random sample for the spherically symmetric Pearson type II distribution.

Usage

PSII(n, p, s)

Arguments

n
number of rows (observations).

p
number of columns (variables).

s
shape parameter, \( s > -1 \).

Value

Returns univariate \((p = 1)\) or multivariate \((p > 1)\) random sample matrix.

References


Examples

```r
set.seed(12345)

## Generate 5X2 random sample matrix from PSII(s=1) ##
PSII(n=5, p=2, s=1)

## Power calculation against bivariate (p=2) PSII(s=1) distribution ##
## at sample size n=50 at one-sided alpha = 0.05 ##
# Zhou-Shao's test #
power.mvnTest(a = 0.05, n = 50, p = 2, B = 100, FUN = PSII, s = 1)
```
Random Generation for the Spherically Symmetric Pearson Type VII Distribution

Description
Generate univariate or multivariate random sample for the spherically symmetric Pearson type VII distribution.

Usage
PSVII(n, p, s)

Arguments
- n: number of rows (observations).
- p: number of columns (variables).
- s: shape parameter, $s > p/2$.

Value
Returns univariate ($p = 1$) or multivariate ($p > 1$) random sample matrix.

References

Examples
```
set.seed(12345)

## Generate 5X2 random sample matrix from PSVII(s=3) ##
PSVII(n=5, p=2, s=3)

## Power calculation against bivariate (p=2) PSVII(s=3) distribution ##
## at sample size n=50 at one-sided alpha = 0.05 ##

# Zhou-Shao's test #
power.mvnTest(a = 0.05, n = 50, p = 2, B = 100, FUN = PSVII, s = 3)
```
Description

Generate univariate or multivariate random sample for general spherically symmetric distributions.

Usage

SPH(n, p, D, ...)

Arguments

- **n**: number of rows (observations).
- **p**: number of columns (variables).
- **D**: random generation functions for some distributions (e.g., `rgamma`, `rbeta`).
- **...**: optional arguments passed to `D`.

Value

Returns univariate \((p = 1)\) or multivariate \((p > 1)\) random sample matrix.

References


Examples

```r
set.seed(12345)

## Generate 5X2 random sample matrix from SPH(Beta(1,1)) ##
SPH(n=5, p=2, D=rbeta, shape1=1, shape2=1)

## Power calculation against bivariate (p=2) SPH(Beta(1,1)) distribution ##
## at sample size n=50 at one-sided alpha = 0.05 ##
# Zhou-Shao's test #
power.mvnTest(a=0.05, n=50, p=2, B=100, FUN=SPH, D=rbeta, shape1=1, shape2=1)
```
Index

copulas, 2
faTest, 3, 6, 8–10, 13
IMMV, 4
mardia, 4, 6, 8–10, 13
mhz, 4, 6, 7, 9, 10, 13
msk, 4, 6, 8, 10, 13
msw, 4, 6, 8, 9, 9, 13
mvn, 6, 8–10
MVNMIX, 11
mvnTest, 4, 6, 8–10, 12
power.faTest, 4, 13
power.mhz, 8, 14
power.msk, 9, 15
power.mswR, 10, 16
power.mswV, 10, 17
power.mvnTest, 13, 18
power.usw, 19
PSII, 20
PSVII, 21

shapiro.test, 10
SPH, 22