Package: mos (via r-universe)

May 12, 2026

Type Package

Title Simulation and Moment Computation for Order Statistics

Version 0.1.3

Description Provides a comprehensive set of tools for working with order statistics, including functions for simulating order statistics, censored samples (Type I and Type II), and record values from various continuous distributions. Additionally, it offers functions to compute moments (mean, variance, skewness, kurtosis) of order statistics for several continuous distributions. These tools assist researchers and statisticians in understanding and analyzing the properties of order statistics and related data. The methods and algorithms implemented in this package are based on several published works, including Ahsanullah et al (2013, ISBN:9789491216831), Arnold and Balakrishnan (2012, ISBN:1461236444), Harter and Balakrishnan (1996, ISBN:9780849394522), Balakrishnan and Sandhu (1995) <doi:10.1080/00031305.1995.10476150>, Genç (2012) <doi:10.1007/s00362-010-0320-y>, Makouei et al (2021) <doi:10.1016/j.cam.2021.113386> and Nagaraja (2013) <doi:10.1016/j.spl.2013.06.028>.

License GPL-3

Encoding UTF-8

Imports stats, hypergeo2

Suggests knitr, rmarkdown, moments

RoxygenNote 7.3.2

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Reyhaneh Arafeh [aut, cre], Mahdi Salehi [aut] (ORCID: <https://orcid.org/0000-0002-0620-0340>)

Maintainer Reyhaneh Arafeh <reyhane.arafe@gmail.com>

Repository https://cran.r-universe.dev

Date/Publication 2025-06-16 11:31:49 UTC

RemoteUrl https://github.com/cran/mos

RemoteRef HEAD

RemoteSha e2de8ec7ef8bcf286c305674cc88861af1fc9ce0

Contents

`kurtOS` *Kurtosis of Order Statistics*

Description

This function computes the kurtosis of order statistics for a given distribution.

Usage

```
kurtOS(r, n, dist = c("unif", "exp", "weibull", "tri"), ...)
```

Arguments

`r`	rank(s) of the desired order statistic(s) (e.g., 1 for the smallest order statistic).
`n`	sample size from which the order statistic is derived.
`dist`	a character string specifying the name of a distribution. Supported values are: • `"unif"`: Uniform distribution • `"exp"`: Exponential distribution • `"weibull"`: Weibull distribution • `"tri"`: Triangular distribution
`...`	further arguments to be passed to `dist`.

Details

The kurtosis of the rth order statistic is calculated using the formula:

$$\text{kurtosis}(X_{r:n}) = \text{E}(\frac{X_{r:n} - \mu_{r:n}}{\sigma_{r:n}})^4$$

where $\mu_{r:n}$ and $\sigma_{r:n}$ are the mean and standard deviation of the rth order statistic, respectively.

Value

The kurtosis of the rth order statistic.

See Also

varOS, skewOS

Examples

```
# Compute the kurtosis of the 3rd order statistic from a sample of size 10
kurtOS(r = 3, n = 10, dist = "unif")
```

---

mo_beta *Moments of Order Statistics from the Beta Distribution (Simulated)*

Description

This function computes the moments of order statistics from the beta distribution using simulation.

Usage

```
mo_beta(r, n, k = 1, a, b, rep = 1e+05, seed = 42)
```

Arguments

r	rank of the desired order statistic (e.g., 1 for the smallest order statistic).
n	sample size from which the order statistic is derived.
k	order of the moment to compute (default is 1).
a, b	non-negative parameters of the beta distribution.
rep	number of simulations (default is 1e5).
seed	optional seed for random number generation to ensure reproducibility (default is 42).

Details

This function estimates the kth moment of the rth order statistic in a sample of size n drawn from a beta distribution with specified shape parameters. The estimation is done via Monte Carlo simulation using the formula:

$$\mathrm{E}[X^k] \approx \frac{1}{\mathrm{rep}} \sum_{i=1}^{\mathrm{rep}} X_i^k,$$

where X_i are the simulated order statistics from the beta distribution.

The function relies on the `ros()` function to generate order statistics.

Value

The estimated kth moment of the rth order statistic from a beta distribution.

Note

The accuracy of the estimated moment depends on the number of simulations (`rep`). The default value `rep = 1e5` provides a reasonable trade-off between speed and accuracy for most practical cases. For higher order moments or when greater precision is required, users are encouraged to increase `rep` (e.g. `1e6`).

See Also

ros for generating random samples of order statistics.

Examples

```
# Compute the first moment of the 2nd order statistic from Beta(3, 4) with sample size 5
mo_beta(r = 2, n = 5, k = 1, a = 3, b = 4)

# Compute the second moment with 10000 simulations
mo_beta(r = 2, n = 5, k = 2, a = 2, b = 2.5, rep = 1e4)
```

mo_compbeta *Moments of Order Statistics from the Complementary Beta Distribution*

Description

This function computes the moments of order statistics from the complementary beta (CB) distribution. For small values of k and integer b, a closed-form formula is used; otherwise, Monte Carlo simulation is applied.

Usage

```
mo_compbeta(r, n, k = 1, a, b, rep = 1e+05, seed = 42, verbose = TRUE)
```

Arguments

r	rank of the desired order statistic (e.g., 1 for the smallest order statistic).
n	sample size from which the order statistic is derived.
k	order of the moment to compute (default is 1).
a, b	positive parameters of the complementary beta distribution.
rep	number of simulations (used when b is non-integer, default is 1e5).
seed	optional seed for random number generation to ensure reproducibility (used when b is non-integer, default is 42).
verbose	logical; if TRUE, prints a message when Monte Carlo simulation is used.

Details

The computation method varies depending on b and k:

- **For integer** b **and** k = 1, 2: The function calculates the moments using the closed-form expression derived in Makouei et al. (2021):

$$\mathrm{E}[X_{r:n}^{s}] = \frac{1}{B(r, n-r+1)} \sum_{j=0}^{n-r} \binom{n-r}{j} (-1)^j \mathcal{M}^{(s)}(a, b, r+j-1),$$

Here

$$\mathcal{M}^{(s)}(a, b, k) = \frac{1}{k+1} \left[1 - \frac{s}{B(a,b)} \sum_{j=0}^{\infty} \binom{b-1}{j} (-1)^j \mathcal{M}^{(s-1)}(a, b, a+k+j) \right], \quad s \geq 1,$$

with the starting point

$$\mathcal{M}^{(1)}(a, b, k) = \frac{B(a+k+1, b+1)}{aB(a,b)} \cdot {}_3F_2\left(a+b, 1, a+k+1; a+1, a+b+k+2; 1\right),$$

where $B(a,b)$ is the beta function, ${}_3F_2$ is the generalized hypergeometric function, and the upper limit of the summation stops at $j = b-1$ if b is an integer.

- **For non-integer** b **or** k > 2: When b is non-integer or k is greater than 2 the function employs Monte Carlo simulation using the following formula:

$$\mathrm{E}[X^s] \approx \frac{1}{\mathrm{rep}} \sum_{i=1}^{\mathrm{rep}} X_i^s,$$

where X_i are the simulated order statistics obtained from the complementary beta distribution. The method relies on the `ros()` function to generate order statistics.

When `verbose = TRUE`, the function prints a message only if Monte Carlo simulation is used (i.e., when k > 2 or b is non-integer).

Value

The estimated or exact kth moment of the rth order statistic from a complementary beta distribution.

Note

The closed-form formula is only available for small values of k and integer b. Monte Carlo simulation is used otherwise, and results may vary slightly depending on `rep`.

References

Makouei, R., Khamnei, H. J., & Salehi, M. (2021). *Moments of order statistics and k-record values arising from the complementary beta distribution with application.* Journal of Computational and Applied Mathematics, 390, 113386.

See Also

ros for generating random samples of order statistics.

Examples

```
# Exact moment when k = 1
mo_compbeta(r = 2, n = 15, k = 1, a = 0.5, b = 2)

# Simulation when k > 2 or b is non-integer
mo_compbeta(r = 2, n = 15, k = 3, a = 2.5, b = 3.7, rep = 1e4)
```

---

mo_exp *Moments Of Order Statistics from the Exponential Distribution*

Description

This function computes the moments of order statistics from the exponential distribution.

Usage

```
mo_exp(r, n, k = 1, mu = 0, sigma = 1)
```

Arguments

r	rank(s) of the desired order statistic(s) (e.g., 1 for the smallest order statistic).
n	sample size from which the order statistic is derived.
k	order of the moment to compute (default is 1).
mu	location parameter of the exponential distribution (default is 0).
sigma	scale parameter of the exponential distribution (default is 1).

Details

The function calculates the kth moment using the following relationship:

$$\mathrm{E}[X_{r:n}^k] = \frac{n!}{(r-1)!(n-r)!} \cdot \sum_{j=0}^{r-1} (-1)^j \binom{r-1}{j} \frac{\Gamma(k+1)}{(n-r+j+1)^{k+1}}.$$

For non-standard exponential distributions with μ and σ parameters, the transformation $X^* = \mu + \sigma X$ is used.

Value

The kth moment of the rth order statistic from an exponential distribution.

References

Ahsanullah, M., Nevzorov, V. B., & Shakil, M. (2013). *An introduction to order statistics* (Vol. 8). Paris: Atlantis Press.

Examples

```
# First moment (mean) of the 2nd order statistic from a sample of size 5
mo_exp(2, 5, k = 1, mu = 0, sigma = 1)

# Second moment of the 3rd order statistic from an exponential distribution
# with mu = 2 and sigma = 3
mo_exp(3, 7, k = 2, mu = 2, sigma = 3)
```

---

mo_gamma *Moments of Order Statistics from the Gamma Distribution (Simulated)*

Description

This function computes the moments of order statistics from the gamma distribution using simulation.

Usage

```
mo_gamma(r, n, k = 1, shape, rate, rep = 1e+05, seed = 42)
```

Arguments

r	rank of the desired order statistic (e.g., 1 for the smallest order statistic).
n	sample size from which the order statistic is derived.
k	order of the moment to compute (default is 1).
shape	shape parameter of the gamma distribution.
rate	rate parameter of the gamma distribution.
rep	number of simulations (default is 1e5).
seed	optional seed for random number generation to ensure reproducibility (default is 42).

Details

This function estimates the kth moment of the rth order statistic in a sample of size n drawn from a gamma distribution with specified shape and rate parameters. The estimation is done via Monte Carlo simulation using the formula:

$$\mathrm{E}[X^k] \approx \frac{1}{\mathrm{rep}} \sum_{i=1}^{\mathrm{rep}} X_i^k,$$

where X_i are the simulated order statistics from the gamma distribution.

The function relies on the `ros()` function to generate order statistics.

Value

The estimated kth moment of the rth order statistic from a gamma distribution.

Note

The accuracy of the estimated moment depends on the number of simulations (`rep`). The default value `rep = 1e5` provides a reasonable trade-off between speed and accuracy for most practical cases. For higher order moments or when greater precision is required, users are encouraged to increase `rep` (e.g. `1e6`).

See Also

`ros`

Examples

```
# Compute the first moment (mean) of the 3rd order statistic from a sample of size 10
mo_gamma(r = 3, n = 10, shape = 2, rate = 1, k = 1)

# Compute the second moment with 10000 simulations
mo_gamma(r = 2, n = 10, shape = 2, rate = 0.5, k = 2, rep = 1e4)
```

mo_kumar *Moments of Order Statistics from the Kumaraswamy Distribution (Simulated)*

Description

This function computes the moments of order statistics from the kumaraswamy distribution using simulation.

Usage

```
mo_kumar(r, n, k = 1, a, b, rep = 1e+05, seed = 42)
```

Arguments

r	rank of the desired order statistic (e.g., 1 for the smallest order statistic).
n	sample size from which the order statistic is derived.
k	order of the moment to compute (default is 1).
a, b	positive parameters of the kumaraswamy distribution.
rep	number of simulations (default is 1e5).
seed	optional seed for random number generation to ensure reproducibility (default is 42).

Details

This function estimates the kth moment of the rth order statistic in a sample of size n drawn from a kumaraswamy distribution with specified shape parameters. The estimation is done via Monte Carlo simulation using the formula:

$$\mathrm{E}[X^k] \approx \frac{1}{\mathrm{rep}} \sum_{i=1}^{\mathrm{rep}} X_i^k,$$

where X_i are the simulated order statistics from the kumaraswamy distribution.

The function relies on the `ros()` function to generate order statistics.

Value

The estimated kth moment of the rth order statistic from a kumaraswamy distribution.

Note

The accuracy of the estimated moment depends on the number of simulations (`rep`). The default value `rep = 1e5` provides a reasonable trade-off between speed and accuracy for most practical cases. For higher order moments or when greater precision is required, users are encouraged to increase `rep` (e.g. `1e6`).

See Also

ros for generating random samples of order statistics.

Examples

```
# Compute the 2nd moment of the 3rd order statistic from Kumaraswamy(2, 3) with sample size 10
mo_kumar(r = 3, n = 10, k = 2, a = 2, b = 3)

# Compute the first moment with 10000 simulations
mo_kumar(r = 2, n = 5, k = 1, a = 2, b = 2.5, rep = 1e4)
```

`mo_norm` *Moments of Order Statistics from the Normal Distribution (Simulated)*

Description

This function computes the moments of order statistics from the normal distribution using simulation.

Usage

```
mo_norm(r, n, k = 1, mean = 0, sd = 1, rep = 1e+05, seed = 42)
```

Arguments

`r`	rank of the desired order statistic (e.g., `1` for the smallest order statistic).
`n`	sample size from which the order statistic is derived.
`k`	order of the moment to compute (default is `1`).
`mean`	mean of the normal distribution (default is `0`).
`sd`	standard deviation of the normal distribution (default is `1`).
`rep`	number of simulations (default is `1e5`).
`seed`	optional seed for random number generation to ensure reproducibility (default is `42`).

Details

This function estimates the kth moment of the rth order statistic in a sample of size n drawn from a normal distribution with the specified mean and standard deviation. The estimation is done via Monte Carlo simulation using the formula:

$$\mathrm{E}[X^k] \approx \frac{1}{\mathrm{rep}} \sum_{i=1}^{\mathrm{rep}} X_i^k,$$

where X_i are the simulated order statistics obtained from the normal distribution.

The function relies on the `ros()` function to generate order statistics.

Value

The estimated kth moment of the rth order statistic from a normal distribution.

Note

The accuracy of the estimated moment depends on the number of simulations (`rep`). The default value `rep = 1e5` provides a reasonable trade-off between speed and accuracy for most practical cases. For higher order moments or when greater precision is required, users are encouraged to increase `rep` (e.g. `1e6`).

See Also

`ros`

Examples

```
# Compute the first moment (mean) of the 3rd order statistic from a sample of size 10
mo_norm(r = 3, n = 10, k = 1, mean = 0, sd = 1)

# Compute the second moment of the 2nd order statistic with 1 million simulations
mo_norm(r = 2, n = 10, k = 2, rep = 1e6)
```

---

mo_pareto *Moments of Order Statistics from the Pareto Distribution (Simulated)*

Description

This function computes the kth moment of order statistics from the pareto distribution using simulation.

Usage

```
mo_pareto(r, n, k = 1, scale, shape, rep = 1e+05, seed = 42)
```

Arguments

`r`	rank of the desired order statistic (e.g., 1 for the smallest order statistic).
`n`	sample size from which the order statistic is derived.
`k`	order of the moment to compute (default is 1).
`scale, shape`	non-negative parameters of the pareto distribution.
`rep`	Number of simulations (default is `1e5`).
`seed`	Optional seed for random number generation to ensure reproducibility (default is 42).

Details

This function estimates the kth moment of the rth order statistic in a sample of size n drawn from a pareto distribution with specified scale and shape parameters. The estimation is done via Monte Carlo simulation using the formula:

$$\mathrm{E}[X^k] \approx \frac{1}{\text{rep}} \sum_{i=1}^{\text{rep}} X_i^k,$$

where X_i are the simulated order statistics from the pareto distribution.

The function relies on the `ros()` function to generate order statistics.

Value

The estimated kth moment of the rth order statistic from a pareto distribution.

Note

The accuracy of the estimated moment depends on the number of simulations (`rep`). The default value `rep = 1e5` provides a reasonable trade-off between speed and accuracy for most practical cases. For higher order moments or when greater precision is required, users are encouraged to increase `rep` (e.g. `1e6`).

See Also

`ros`

Examples

```
# Compute the first moment (mean) of the 3rd order statistic from a sample of size 10
mo_pareto(r = 3, n = 10, scale = 2, shape = 3, k = 1)

# Compute the second moment with 1 million simulations
mo_pareto(r = 2, n = 10, scale = 1, shape = 2, k = 2, rep = 1e6)
```

---

mo_t *Moments of Order Statistics from the Student's t-Distribution (Simulated)*

Description

This function computes the moments of order statistics from the student's t-distribution using simulation.

Usage

```
mo_t(r, n, k = 1, df, rep = 1e+05, seed = 42)
```

Arguments

r	rank of the desired order statistic (e.g., 1 for the smallest order statistic).
n	sample size from which the order statistic is derived.
k	order of the moment to compute (default is 1).
df	degrees of freedom for the student's t-distribution.
rep	number of simulations (default is 1e5).
seed	optional seed for random number generation to ensure reproducibility (default is 42).

Details

This function estimates the kth moment of the rth order statistic in a sample of size n drawn from a student's t-distribution with the specified degrees of freedom (df). The estimation is done via Monte Carlo simulation using the formula:

$$\mathrm{E}[X^k] \approx \frac{1}{\mathrm{rep}} \sum_{i=1}^{\mathrm{rep}} X_i^k,$$

where X_i are the simulated order statistics obtained from the student's t-distribution.

The function relies on the ros() function to generate order statistics.

Value

The estimated kth moment of the rth order statistic from a student's t-distribution.

Note

The accuracy of the estimated moment depends on the number of simulations (rep). The default value rep = 1e5 provides a reasonable trade-off between speed and accuracy for most practical cases. For higher order moments or when greater precision is required, users are encouraged to increase rep (e.g. 1e6).

See Also

ros

Examples

```
# Compute the first moment (mean) of the 3rd order statistic from a sample of size 10
mo_t(r = 3, n = 10, df = 5, k = 1)

# Compute the second moment of the 2nd order statistic with 10000 simulations
mo_t(r = 2, n = 10, df = 10, k = 2, rep = 1e4)
```

`mo_topple` *Moments of Order Statistics from the Topp-Leone Distribution*

Description

This function computes the moments of order statistic from the topp-leone distribution, based on the formula presented in Genç, A. İ. (2012).

Usage

```
mo_topple(r, n, k = 1, a, b = 1)
```

Arguments

`r`	rank(s) of the desired order statistic(s) (e.g., 1 for the smallest order statistic).
`n`	sample size from which the order statistic is derived.
`k`	order of the moment to compute (default is 1).
`a`	shape parameter of the topp-leone distribution (`a > 0`).
`b`	scale parameter of the topp-leone distribution (default is 1, `b > 0`).

Details

This function implements the exact formula for moments of order statistics from the topp-leone distribution as provided in Genç, A. İ. (2012):

$$\mathrm{E}[X_{r:n}^k] = \frac{n!(ab^k)}{(r-1)!(n-r)!} \sum_{j=0}^{n-r} \binom{n-r}{j} (-1)^j 2^{k+2a(r+j)} \left[B_{1/2}(k+a(r+j), a(r+j)) - 2B_{1/2}(k+a(r+j)+1, a(r$$

Here, $B_x(.,.)$ is the incomplete Beta function.

Value

The kth moment of the rth order statistic from a topp-leone distribution.

References

Genç, A. İ. (2012). *Moments of order statistics of Topp–Leone distribution.* Statistical Papers, 53, 117-131.

Examples

```
# Compute the first moment of the first order statistic for n=5, a=2, b=1
mo_topple(1, 5, 1, 2)

# Compute the second moment of the second order statistic for n=10, a=1.5, b=2
mo_topple(2, 10, 2, 1.5, 2)
```

mo_tri *Moments of Order Statistics from the Symmetric Triangular Distribution*

Description

This function computes the moments of order statistics from the symmetric triangular distribution.

Usage

```
mo_tri(r, n, k = 1)
```

Arguments

r	rank(s) of the desired order statistic(s) (e.g., 1 for the smallest order statistic).
n	sample size from which the order statistic is derived.
k	order of the moment to compute (default is 1).

Details

The function implements the following relationship from Nagaraja (2013) for the symmetric triangular distribution:

$$\mathrm{E}[X_{r:n}^k] = \frac{n!}{(r-1)!(n-r)!}\left\{(\frac{1}{2})^{k/2}B\left(\frac{1}{2};\frac{k}{2}+r,n-r+1\right)+\sum_{j=0}^{k}(-1)^j\binom{k}{j}(\frac{1}{2})^{j/2}B\left(\frac{1}{2};\frac{j}{2}+n-r+1,r\right)\right\}.$$

Here, $B(x;a,b)$ is the incomplete Beta function.

Value

The kth moment of the rth order statistic from a symmetric triangular distribution.

References

Nagaraja, H. N. (2013). *Moments of order statistics and L-moments for the symmetric triangular distribution.* Statistics & Probability Letters, 83(10), 2357-2363.

Examples

```
# Compute the 2nd moment of the 3rd order statistic for n=5
mo_tri(3, 5, 2)
```

mo_unif *Moments of Order Statistics from the Uniform Distribution*

Description

This function computes the moments of order statistics for the uniform distribution based on the relationship described by Arnold and Balakrishnan (2012).

Usage

```
mo_unif(r, n, k = 1, a = 0, b = 1)
```

Arguments

r	rank(s) of the desired order statistic(s) (e.g., 1 for the smallest order statistic).
n	sample size from which the order statistic is derived.
k	order of the moment to compute (default is 1).
a, b	lower and upper limits of the distribution. Must be finite.

Details

The function calculates the kth moment based on the formula:

$$\mathrm{E}[U_{r,n}^k] = \frac{B(k+r, n-r+1)}{B(r, n-r+1)},$$

where $B(a, b)$ is the complete beta function. When $a \neq 0$ or $b \neq 1$, the transformation $U^* = a + (b-a)U$ is used.

Value

The kth moment of the rth order statistic from a uniform distribution.

References

Arnold, B. C., & Balakrishnan, N. (2012). *Relations, bounds and approximations for order statistics* (Vol. 53). Springer Science & Business Media.

Examples

```
# Example 1: First moment (mean) of the 2nd order statistic from a sample of size 5
mo_unif(2, 5, k = 1, a = 0, b = 1)

# Example 2: Second moment of the 3rd order statistic from a uniform distribution on [2, 5]
mo_unif(3, 7, k = 2, a = 2, b = 5)
```

`mo_weibull` *Moments of Order Statistics from the Weibull Distribution*

Description

This function computes the moments of order statistics from the weibull distribution.

Usage

```
mo_weibull(r, n, k = 1, shape, scale = 1)
```

Arguments

`r`	rank(s) of the desired order statistic(s) (e.g., 1 for the smallest order statistic).
`n`	sample size from which the order statistic is derived.
`k`	order of the moment to compute (default is 1).
`shape`	shape parameter of the weibull distribution.
`scale`	scale parameter of the weibull distribution (default is 1).

Details

The function calculates the kth moment using the formula:

$$\mathrm{E}[X_{r:n}^k] = \frac{n!}{(r-1)!(n-r)!}\Gamma\left(1+\frac{k}{\text{shape}}\right)\sum_{j=0}^{r-1}(-1)^j\binom{r-1}{j}\frac{1}{(n-r+1+j)^{1+\frac{k}{\text{shape}}}}$$

For non-standard weibull distributions (`scale` not equal to 1), the relationship $\mathrm{E}[Z_{r:n}^k] = \text{scale}^k\mathrm{E}[X_{r:n}^k]$ is used.

Value

The kth moment of the rth order statistic from a weibull distribution.

References

Harter, H. L., & Balakrishnan, N. (1996). *CRC handbook of tables for the use of order statistics in estimation.* CRC press.

Examples

```
# Example 1: Standard weibull distribution (shape = 2, scale = 1)
mo_weibull(r = 2, n = 5, k = 1, shape = 2)

# Example 2: Non-standard weibull distribution (shape = 2, scale = 3)
mo_weibull(r = 3, n = 6, k = 2, shape = 2, scale = 3)
```

`rcens` *Generate Censored Samples (Type I or Type II)*

Description

This function generates censored samples from a specified distribution, using Type I (time-based) or Type II (failure-based) censoring schemes.

Usage

```
rcens(n, r = NULL, dist, type = c("I", "II"), cens.time = NULL, ...)
```

Arguments

`n`	total number of items in the sample
`r`	number of uncensored observations (only for Type II censoring).
`dist`	a character string specifying the name of the distribution (e.g., `"norm"` for the normal distribution, `"exp"` for the exponential distribution).
`type`	type of censoring: `"I"` for Type I (time-based) or `"II"` for Type II (failure-based).
`cens.time`	censoring time for Type I censoring.
`...`	further arguments to be passed to `dist`.

Details

This function implements two types of censoring schemes:

1. **Type I censoring:** Observations are censored if they exceed a specified `cens.time`. The function returns all observations less than `cens.time`.
2. **Type II censoring:** The smallest `r` observations are returned, simulating a situation where the experiment stops after `r` failures.

Value

A numeric vector of censored samples.

See Also

`rpcens2`

Examples

```
# Type I censoring: Exponential distribution with rate = 1, censored at time 2
rcens(n = 10, dist = "exp", type = "I", cens.time = 2, rate = 1)

# Type II censoring: Normal distribution, smallest 5 values
rcens(n = 10, r = 5, dist = "norm", type = "II", mean = 0, sd = 1)
```

rkrec *Generate Upper and Lower k-Records from Continuous Distributions*

Description

This function generates k-records (upper or lower) from a specified continuous distribution.

Usage

```
rkrec(size, k, record = c("upper", "lower"), dist, ...)
```

Arguments

size	number of k-records to generate.
k	the rank of the record to generate (k-record).
record	the type of record to generate: "upper" for upper k-records, "lower" for lower k-records. Default is "upper".
dist	a character string specifying the name of the continuous distribution (e.g., "norm", "exp", "gamma").
...	further arguments to be passed to dist.

Details

Note: Setting k = 1 generates standard (1-)records.

Value

A numeric vector of size size, representing the simulated k-records.

See Also

ros

Examples

```
# Generate 5 upper 2-records from the normal distribution
rkrec(size = 5, k = 2, record = "upper", dist = "norm", mean = 0, sd = 1)

# Generate 5 lower 3-records from the exponential distribution
rkrec(size = 5, k = 3, record = "lower", dist = "exp", rate = 1)
```

ros *Generate Random Data from Order Statistics*

Description

This function generates random data from the order statistics of a specified distribution. The user can specify a known distribution in R or provide a custom quantile function.

Usage

```
ros(size, r, n, dist = NULL, qf = NULL, ...)
```

Arguments

`size`	number of observations.
`r`	rank(s) of the desired order statistic(s) (e.g., 1 for the smallest order statistic).
`n`	sample size from which the order statistic is derived.
`dist`	a character string specifying the name of a known distribution in R (e.g. `'norm'`, `'exp'`). Default is `NULL`.
`qf`	a custom quantile function, either as a name (string) or directly as a function. Default is `NULL`.
`...`	further arguments to be passed to `dist` or `qf`.

Details

The `ros` function generates random data from order statistics using two approaches:

1. **Using a Known Distribution:** When `dist` is provided, random data is generated from a known distribution in R.
2. **Using a Custom Quantile Function:** When `qf` is provided, `ros` applies the user-provided quantile function to generate random data.

Value

A numeric vector or matrix containing the generated random data from the specified order statistics. If a single rank is provided (i.e., scalar `r`), a numeric vector of size `size` is returned. If multiple ranks are provided (i.e., vector `r`), a matrix is returned with `size` rows and `length(r)` columns, where each row corresponds to a simulation and each column to an order statistic.

Examples

```
# Example 1: Generate from the normal distribution
ros(5, 3, 15, "norm", mean = 4, sd = 2)

# Example 2: Using a custom quantile function for the Pareto distribution
ros(3, 2, 10, qf = function(p, scale, shape) scale * (1 - p)^(-1 / shape), scale = 3, shape = 2)
```

```
# Example 3: Generate multiple order statistics from the uniform distribution
# In this example, first through 5th order statistics are generated from a sample size of 5.
ros(3, 1:5, 5, dist = "unif")
```

---

rpcens2 *Generate Progressive Type-II Censored Samples*

Description

This function generates progressive Type-II censored samples based on the algorithm provided by Balakrishnan and Sandhu (1995).

Usage

```
rpcens2(n, R, dist, ...)
```

Arguments

`n`	total number of items in the sample.
`R`	a vector of non-negative integers representing the number of items censored at each stage of the experiment. The length of `R` determines the number of failure stages (`m`), and the sum of `R` plus `m` must equal `n`.
`dist`	a character string specifying the name of the distribution (e.g., `"norm"` for the normal distribution, `"exp"` for the exponential distribution).
`...`	further arguments to be passed to `dist`.

Details

This function implements the algorithm described by Balakrishnan and Sandhu (1995) to simulate progressive Type-II censored samples. Progressive Type-II censoring is a common scheme in reliability and life-testing experiments, where items are progressively removed (censored) during the testing process.

Value

A numeric vector of size `m`, representing the failure times of the observed items.

References

Balakrishnan, N., & Sandhu, R. A. (1995). *A simple simulational algorithm for generating progressive Type-II censored samples.* The American Statistician, 49(2), 229-230.

See Also

`rcens`

Examples

```
# Generate a progressive Type-II censored sample from the normal distribution
n <- 10
R <- c(2, 1, 2, 0, 0)
rpcens2(n, R, dist = "norm", mean = 0, sd = 1)

# Generate a progressive Type-II censored sample from the exponential distribution
rpcens2(n = 10, R = c(2, 2, 1, 0, 0), dist = "exp", rate = 1)
```

---

skewOS *Skewness of Order Statistics*

Description

This function computes the skewness of the order statistics for a given distribution.

Usage

```
skewOS(r, n, dist = c("unif", "exp", "weibull", "tri"), ...)
```

Arguments

`r`	rank(s) of the desired order statistic(s) (e.g., 1 for the smallest order statistic).
`n`	sample size from which the order statistic is derived.
`dist`	a character string specifying the name of a distribution. Supported values are: • `"unif"`: Uniform distribution • `"exp"`: Exponential distribution • `"weibull"`: Weibull distribution • `"tri"`: Triangular distribution
`...`	further arguments to be passed to `dist`.

Details

The skewness of the rth order statistic is calculated using the formula:

$$\text{Skewness}(X_{r:n}) = \text{E}(\frac{X_{r:n} - \mu_{r:n}}{\sigma_{r:n}})^3$$

where $\mu_{r:n}$ and $\sigma_{r:n}$ are the mean and standard deviation of the rth order statistic, respectively.

Value

The skewness of the rth order statistic.

See Also

varOS, kurtOS

Examples

```
# Skewness of the 3rd order statistic for a uniform distribution
skewOS(3, 10, "unif")
```

---

varOS *Variance of Order Statistics*

Description

This function computes the variance of order statistics for a given distribution.

Usage

```
varOS(r, n, dist = c("unif", "exp", "weibull", "tri", "topple"), ...)
```

Arguments

r	rank(s) of the desired order statistic(s) (e.g., 1 for the smallest order statistic).
n	sample size from which the order statistic is derived.
dist	a character string specifying the name of a distribution. Supported values are: • "unif": Uniform distribution • "exp": Exponential distribution • "weibull": Weibull distribution • "tri": Triangular distribution • "topple": Topp-Leon distribution
...	further arguments to be passed to dist.

Details

This function computes the variance of the rth order statistic ($X_{r:n}$) for a given sample size (n) and distribution (dist). The variance is calculated using:

$$\text{Var}(X_{r:n}) = \text{E}(X_{r:n}^2) - (\text{E}(X_{r:n}))^2$$

Value

The variance of the rth order statistic.

Examples

```
# Variance of the 3rd order statistic in a sample of size 10 from a uniform distribution
varOS(r = 3, n = 10, dist = "unif")
```

Index