Package ‘reliaR’

February 20, 2015

Type Package
Title Package for some probability distributions.
Version 0.01
Date 2010-10-28
Author Vijay Kumar <vkgkp@rediffmail.com> and Uwe Ligges
<ligges@statistik.tu-dortmund.de>
Maintainer Vijay Kumar <vkgkp@rediffmail.com>
Description A collection of utilities for some reliability models/probability distributions.
License GPL-2
LazyLoad yes
Repository CRAN
Date/Publication 2011-01-03 18:56:29
NeedsCompilation no

R topics documented:

abic.burrX ......................................................... 4
abic.chen ...................................................... 5
abic.exp.ext .................................................... 6
abic.exp.power .................................................. 7
abic.expo.logistic .............................................. 8
abic.expo.weibull .............................................. 9
abic.flex.weibull .............................................. 10
abic.gen.exp ................................................... 11
abic.gompertz ................................................... 12
abic.gp.weibull ................................................ 13
abic.gumbel ..................................................... 14
abic.inv.genexp ............................................... 15
abic.lfr ......................................................... 16
abic.log.gamma ................................................ 17
abic.logis.exp ................................................. 18
R topics documented:

abic.logis.rayleigh ................................. 19
abic.loglog ......................................... 20
abic.moe ............................................. 21
abic.moew ............................................ 22
abic.weibull.ext ................................... 23
bearings ............................................... 24
BurrX .................................................. 25
BurrXsurvival ........................................ 26
Chen .................................................... 28
Chensurvival .......................................... 30
conductors ............................................ 31
dataset2 ............................................... 32
EPsurvival ............................................. 33
ExpExt .................................................. 34
ExpExtsurvival ....................................... 35
ExpoLogistic .......................................... 37
ExpoLogisticsurvival ............................... 38
ExpoWeibull .......................................... 39
ExpoWeibullsurvival ............................... 41
ExpPower .............................................. 42
FlexWeibull .......................................... 43
FlexWeibullsurvival ............................... 45
GenExp ................................................ 46
GenExpsurvival ........................................ 47
Gompertz .............................................. 49
Gompertzsurvival ..................................... 50
GPWeibull ............................................ 51
GPWeibullsurvival ................................... 53
Gumbel ............................................... 54
Gumbelsurvival ....................................... 55
InvGenExp ............................................. 57
InvGenExpEsurvival ................................. 58
ks.burrX .............................................. 59
ks.chen ............................................... 61
ks.exp.ext ........................................... 62
ks.exp.power ........................................ 63
ks.expo.logistic .................................... 64
ks.expo.weibull ..................................... 65
ks.flex.weibull ...................................... 67
ks.gen.exp .......................................... 68
ks.gompertz ......................................... 69
ks.gp.weibull ........................................ 70
ks.gumbel .......................................... 71
ks.inv.genexp ....................................... 72
ks.lfr ................................................. 74
ks.log.gamma ........................................ 75
ks.logis.exp ......................................... 76
ks.logis.rayleigh ................................... 77
<table>
<thead>
<tr>
<th>R topics documented:</th>
</tr>
</thead>
<tbody>
<tr>
<td>ks.loglog</td>
</tr>
<tr>
<td>ks.moe</td>
</tr>
<tr>
<td>ks.moew</td>
</tr>
<tr>
<td>ks.weibull.ext</td>
</tr>
<tr>
<td>LFR</td>
</tr>
<tr>
<td>LFRsurvival</td>
</tr>
<tr>
<td>Loggamma</td>
</tr>
<tr>
<td>Loggamma survival</td>
</tr>
<tr>
<td>LogisExp</td>
</tr>
<tr>
<td>LogisExp-survival</td>
</tr>
<tr>
<td>LogisRayleigh</td>
</tr>
<tr>
<td>LogisRayleigh-survival</td>
</tr>
<tr>
<td>Loglog</td>
</tr>
<tr>
<td>Loglog-survival</td>
</tr>
<tr>
<td>MOEE</td>
</tr>
<tr>
<td>MOEE-survival</td>
</tr>
<tr>
<td>MOEW</td>
</tr>
<tr>
<td>MOEW-survival</td>
</tr>
<tr>
<td>pp.burrX</td>
</tr>
<tr>
<td>pp.chen</td>
</tr>
<tr>
<td>pp.exp.ext</td>
</tr>
<tr>
<td>pp.exp.power</td>
</tr>
<tr>
<td>pp.exp-logistic</td>
</tr>
<tr>
<td>pp.exp-weibull</td>
</tr>
<tr>
<td>pp.flex-weibull</td>
</tr>
<tr>
<td>pp.gen.exp</td>
</tr>
<tr>
<td>pp.gompertz</td>
</tr>
<tr>
<td>pp.gp-weibull</td>
</tr>
<tr>
<td>pp.gumbel</td>
</tr>
<tr>
<td>pp.inv-geneexp</td>
</tr>
<tr>
<td>pp.lfr</td>
</tr>
<tr>
<td>pp.log-gamma</td>
</tr>
<tr>
<td>pp.logis.exp</td>
</tr>
<tr>
<td>pp.logis-rayleigh</td>
</tr>
<tr>
<td>pp.loglog</td>
</tr>
<tr>
<td>pp.moe</td>
</tr>
<tr>
<td>pp.moew</td>
</tr>
<tr>
<td>pp.weibull.ext</td>
</tr>
<tr>
<td>qq.burrX</td>
</tr>
<tr>
<td>qq.chen</td>
</tr>
<tr>
<td>qq.exp.ext</td>
</tr>
<tr>
<td>qq.exp.power</td>
</tr>
<tr>
<td>qq.exp-logistic</td>
</tr>
<tr>
<td>qq.exp-weibull</td>
</tr>
<tr>
<td>qq.flex-weibull</td>
</tr>
<tr>
<td>qq.gen.exp</td>
</tr>
<tr>
<td>qq.gompertz</td>
</tr>
<tr>
<td>qq.gp-weibull</td>
</tr>
</tbody>
</table>
abic.burrX

Akaike information criterion (AIC) and Bayesian information criterion (BIC) for BurrX distribution

Description

The function abic.burrX() gives the loglikelihood, AIC and BIC values assuming an BurrX distribution with parameters alpha and lambda.

Usage

abic.burrX(x, alpha.est, lambda.est)

Arguments

x vector of observations
alpha.est estimate of the parameter alpha
lambda.est estimate of the parameter lambda

Value

The function abic.burrX() gives the loglikelihood, AIC and BIC values.
References


See Also

pp.burrX for PP plot and qq.burrX for QQ plot

Examples

```r
## Load data sets
data(bearings)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.1989515, lambda.est = 0.0130847

## Values of AIC, BIC and LogLik for the data(bearings)
abic.burrX(bearings, 1.1989515, 0.0130847)
```

abic.chen

Akaike information criterion (AIC) and Bayesian information criterion (BIC) for a sample from Chen distribution

Description

The function abic.chen() gives the loglikelihood, AIC and BIC values assuming Chen distribution with parameters beta and lambda. The function is based on the invariance property of the MLE.

Usage

abic.chen(x, beta.est, lambda.est)

Arguments

- `x` : vector of observations
- `beta.est` : estimate of the parameter beta
- `lambda.est` : estimate of the parameter lambda
Value
The function abic.chen() gives the loglikelihood, AIC and BIC values.

References

See Also
ppo.chen for PP plot and qq.chen for QQ plot

Examples
```r
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of beta & lambda for the data(sys2)
## beta.est = 0.262282404, lambda.est = 0.007282371

## Values of AIC, BIC and LogLik for the data(sys2)
abic.chen(sys2, 0.262282404, 0.007282371)
```

abic.exp.ext

Akaike information criterion (AIC) and Bayesian information criterion (BIC) for Exponential Extension(EE) distribution

Description
The function abic.exp.ext() gives the loglikelihood, AIC and BIC values assuming an Exponential Extension(EE) distribution with parameters alpha and lambda.

Usage
abic.exp.ext(x, alpha.est, lambda.est)

Arguments
- **x**: vector of observations
- **alpha.est**: estimate of the parameter alpha
- **lambda.est**: estimate of the parameter lambda
abic.exp.power

Value

The functionabic.exp.ext() gives the loglikelihood, AIC and BIC values.

References


See Also

pp.exp.ext for PP plot and qq.exp.ext for QQ plot

Examples

```r
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.0126e+01, lambda.est = 1.5848e-04

## Values of AIC, BIC and LogLik for the data(sys2)
abic.exp.ext(sys2, 1.0126e+01, 1.5848e-04)
```

abic.exp.power

Akaike information criterion (AIC) and Bayesian information criterion (BIC) for a sample from Exponential Power(EP) distribution

Description

The functionabic.exp.power() gives the loglikelihood, AIC and BIC values assuming Chen distribution with parameters alpha and lambda. The function is based on the invariance property of the MLE.

Usage

abic.exp.power(x, alpha.est, lambda.est)
Arguments

x          vector of observations
alpha.est  estimate of the parameter alpha
lambda.est estimate of the parameter lambda

Value

The function abic.exp.power() gives the loglikelihood, AIC and BIC values.

References


See Also

pp.exp.power for PP plot and qq.exp.power for QQ plot

Examples

```r
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.905868898, lambda.est = 0.001531423

## Values of AIC, BIC and LogLik for the data(sys2)
abic.exp.power(sys2, 0.905868898, 0.001531423)
```

abic.exp.logistic

*Akaike information criterion (AIC) and Bayesian information criterion (BIC) for Exponentiated Logistic(EL) distribution*

Description

The function abic.exp.logistic() gives the loglikelihood, AIC and BIC values assuming an Exponentiated Logistic(EL) distribution with parameters alpha and beta.

Usage

abic.exp.logistic(x, alpha.est, beta.est)
Arguments

- **x**: vector of observations
- **alpha.est**: estimate of the parameter alpha
- **beta.est**: estimate of the parameter beta

Value

The function `abic.expo.logistic()` gives the loglikelihood, AIC and BIC values.

References


See Also

- `pp.expo.logistic` for PP plot and `qq.expo.logistic` for QQ plot

Examples

```r
## Load data sets
data(dataset2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(dataset2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 5.31302, beta.est = 139.04515

## Values of AIC, BIC and LogLik for the data(dataset2)
abic.expo.logistic(dataset2, 5.31302, 139.04515)
```

abic.expo.weibull

**Akaike information criterion (AIC) and Bayesian information criterion (BIC) for Exponentiated Weibull(EW) distribution**

Description

The function `abic.expo.weibull()` gives the loglikelihood, AIC and BIC values assuming an Exponentiated Weibull(EW) distribution with parameters alpha and theta.

Usage

abic.expo.weibull(x, alpha.est, theta.est)
Arguments

- \( x \) vector of observations
- \( \text{alpha.est} \) estimate of the parameter alpha
- \( \text{theta.est} \) estimate of the parameter theta

Value

The function \( \text{abic expo weibull()} \) gives the loglikelihood, AIC and BIC values.

References


See Also

- \( \text{pp expo weibull} \) for PP plot and \( \text{qq expo weibull} \) for QQ plot

Examples

```r
## Load data sets
data(stress)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(stress)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est = 1.026465, theta.est = 7.824943

## Values of AIC, BIC and LogLik for the data(stress)
abic.expo.weibull(stress, 1.026465, 7.824943)
```

Description

The function \( \text{abic flex weibull()} \) gives the loglikelihood, AIC and BIC values assuming an flexible Weibull(FW) distribution with parameters alpha and beta.

Usage

\( \text{abic flex weibull}(x, \text{alpha.est}, \text{beta.est}) \)
**abic.gen.exp**

Arguments

- **x**: vector of observations
- **alpha.est**: estimate of the parameter alpha
- **beta.est**: estimate of the parameter beta

Value

The function `abic.flex.weibull()` gives the loglikelihood, AIC and BIC values.

References


See Also

- `pp.flex.weibull` for PP plot and `qq.flex.weibull` for QQ plot

Examples

```r
## Load data sets
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(repairtimes)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 0.07077507, beta.est = 1.13181535

## Values of AIC, BIC and LogLik for the data(repairtimes)
abic.flex.weibull(repairtimes, 0.07077507, 1.13181535)
```

---

abic.gen.exp *Akaike information criterion (AIC) and Bayesian information criterion (BIC) for a sample from Generalized Exponential distribution*

Description

The function `abic.gen.exp()` gives the loglikelihood, AIC and BIC values assuming an Generalized Exponential distribution with parameters alpha and lambda. The function is based on the invariance property of the MLE.
Usage
abic.gen.exp(x, alpha.est, lambda.est)

Arguments
x vector of observations
alpha.est estimate of the parameter alpha
lambda.est estimate of the parameter lambda

Value
The function abic.gen.exp() gives the loglikelihood, AIC and BIC values.

References

See Also
pp.gen.exp for PP plot and qq.gen.exp for QQ plot

Examples
## Load data set
data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 5.28321139, lambda.est = 0.03229609
abic.gen.exp(bearings, 5.28321139, 0.03229609)

abic.gompertz Akaike information criterion (AIC) and Bayesian information criterion (BIC) for Gompertz distribution

Description
The function abic.gompertz() gives the loglikelihood, AIC and BIC values assuming a Gompertz distribution with parameters alpha and theta.

Usage
abic.gompertz(x, alpha.est, theta.est)

Arguments
x vector of observations
alpha.est estimate of the parameter alpha
theta.est estimate of the parameter theta
abic.gp.weibull

Value

The function `abic.gompertz()` gives the loglikelihood, AIC and BIC values.

References


See Also

`pp.gompertz` for PP plot and `qq.gompertz` for QQ plot

Examples

```r
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(sys2)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est = 0.00121307, theta.est = 0.00173329

## Values of AIC, BIC and LogLik for the data(sys2)
abic.gompertz(sys2, 0.00121307, 0.00173329)
```

abic.gp.weibull

Akaike information criterion (AIC) and Bayesian information criterion (BIC) for generalized power Weibull(GPW) distribution

Description

The function `abic gp.weibull()` gives the loglikelihood, AIC and BIC values assuming an generalized power Weibull(GPW) distribution with parameters alpha and theta.

Usage

```r
abic.gp.weibull(x, alpha.est, theta.est)
```

Arguments

- `x` vector of observations
- `alpha.est` estimate of the parameter alpha
- `theta.est` estimate of the parameter theta
Value

The function `abic.gp.weibull()` gives the loglikelihood, AIC and BIC values.

References


See Also

`pp.gp.weibull` for PP plot and `qq.gp.weibull` for QQ plot

Examples

```r
## Load data sets
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(repairtimes)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est = 1.566093, theta.est = 0.355321
#
## Values of AIC, BIC and LogLik for the data(repairtimes)
abic.gp.weibull(repairtimes, 1.566093, 0.355321)
```

abic.gumbel

*Akaike information criterion (AIC) and Bayesian information criterion (BIC) for Gumbel distribution*

Description

The function `abic.gumbel()` gives the loglikelihood, AIC and BIC values assuming an Gumbel distribution with parameters mu and sigma.

Usage

`abic.gumbel(x, mu.est, sigma.est)`

Arguments

- `x`: vector of observations
- `mu.est`: estimate of the parameter mu
- `sigma.est`: estimate of the parameter sigma
The function `abic.gumbel()` gives the loglikelihood, AIC and BIC values.

**References**


**See Also**

`pp.gumbel` for PP plot and `qq.gumbel` for QQ plot

**Examples**

```r
## Load data sets
data(dataset2)
## Maximum Likelihood(ML) Estimates of mu & sigma for the data(dataset2)
## Estimates of mu & sigma using 'maxLik' package
## mu.est = 212.157, sigma.est = 151.768

## Values of AIC, BIC and LogLik for the data(dataset2)
abic.gumbel(dataset2, 212.157, 151.768)
```

**Description**

The function `abic.inv.genexp()` gives the loglikelihood, AIC and BIC values assuming an Inverse Generalized Exponential(IGE) distribution with parameters alpha and lambda.

**Usage**

```r
abic.inv.genexp(x, alpha.est, lambda.est)
```

**Arguments**

- `x` : vector of observations
- `alpha.est` : estimate of the parameter alpha
- `lambda.est` : estimate of the parameter lambda

**abic.inv.genexp**

*Akaike information criterion (AIC) and Bayesian information criterion (BIC) for Inverse Generalized Exponential(IGE) distribution*
The function `abic.inv.genexp()` gives the loglikelihood, AIC and BIC values.

References


See Also

`pp.inv.genexp` for PP plot and `qq.inv.genexp` for QQ plot

Examples

```r
## Load data sets
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(repairtimes)
## Estimates of alpha & lambda using 'maxLik' package
alpha.est = 1.097807, lambda.est = 1.206889

## Values of AIC, BIC and LogLik for the data(repairtimes)
abic.inv.genexp(repairtimes, 1.097807, 1.206889)
```

abic.lfr

*Akaike information criterion (AIC) and Bayesian information criterion (BIC) for linear failure rate(LFR) distribution*

Description

The function `abic.lfr()` gives the loglikelihood, AIC and BIC values assuming a linear failure rate(LFR) distribution with parameters alpha and beta.

Usage

`abic.lfr(x, alpha.est, beta.est)`

Arguments

- `x` vector of observations
- `alpha.est` estimate of the parameter alpha
- `beta.est` estimate of the parameter beta
**abic.log.gamma**

Value

The function `abic.lfr()` gives the loglikelihood, AIC and BIC values.

References


See Also

`pp.lfr` for PP plot and `qq.lfr` for QQ plot

Examples

```r
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(sys2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 1.77773e-03,  beta.est = 2.77764e-06

## Values of AIC, BIC and LogLik for the data(sys2)
abic.lfr(sys2, 1.777673e-03, 2.777640e-06)
```

**abic.log.gamma**

Akaike information criterion (AIC) and Bayesian information criterion (BIC) for log-gamma(LG) distribution

Description

The function `abic.log.gamma()` gives the loglikelihood, AIC and BIC values assuming an log-gamma(LG) distribution with parameters alpha and lambda.

Usage

`abic.log.gamma(x, alpha.est, lambda.est)`

Arguments

- `x` : vector of observations
- `alpha.est` : estimate of the parameter alpha
- `lambda.est` : estimate of the parameter lambda
Value

The function `abic.log.gamma()` gives the loglikelihood, AIC and BIC values.

References


See Also

`pp.log.gamma` for PP plot and `qq.log.gamma` for QQ plot

Examples

```r
## Load data sets
data(conductors)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(conductors)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 0.0088741, lambda.est = 0.6059935

## Values of AIC, BIC and LogLik for the data(conductors)
abic.log.gamma(conductors, 0.0088741, 0.6059935)
```

abic.logis.exp	Akaike information criterion (AIC) and Bayesian information criterion (BIC) for Logistic-Exponential(LE) distribution

Description

The function `abic.logis.exp()` gives the loglikelihood, AIC and BIC values assuming an Logistic-Exponential(LE) distribution with parameters alpha and lambda.

Usage

`abic.logis.exp(x, alpha.est, lambda.est)`

Arguments

x	vector of observations
alpha.est	estimate of the parameter alpha
lambda.est	estimate of the parameter lambda
abic.logis.rayleigh

Value
The function abic.logis.exp() gives the loglikelihood, AIC and BIC values.

References

See Also
pp.logis.exp for PP plot and qq.logis.exp for QQ plot

Examples

```r
## Load data sets
data(bearings)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 2.36754, lambda.est = 0.01059

## Values of AIC, BIC and LogLik for the data(bearings)
abic.logis.exp(bearings, 2.36754, 0.01059)
```

abic.logis.rayleigh Akaike information criterion (AIC) and Bayesian information criterion (BIC) for Logistic-Rayleigh(LR) distribution

Description
The function abic.logis.rayleigh() gives the loglikelihood, AIC and BIC values assuming an Logistic-Rayleigh(LR) distribution with parameters alpha and lambda.

Usage
abic.logis.rayleigh(x, alpha.est, lambda.est)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>vector of observations</td>
</tr>
<tr>
<td>alpha.est</td>
<td>estimate of the parameter alpha</td>
</tr>
<tr>
<td>lambda.est</td>
<td>estimate of the parameter lambda</td>
</tr>
</tbody>
</table>
The function `abic.logis.rayleigh()` gives the loglikelihood, AIC and BIC values.

References


See Also

`pp.logis.rayleigh` for PP plot and `qq.logis.rayleigh` for QQ plot

Examples

```r
## Load data sets
data(stress)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(stress)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.4779388, lambda.est = 0.2141343

## Values of AIC, BIC and LogLik for the data(stress)
abic.logis.rayleigh(stress, 1.4779388, 0.2141343)
```

abic.loglog

Akaike information criterion (AIC) and Bayesian/ Schwartz information criterion (BIC)/(SIC) for a sample from Loglog distribution

Description

The function `abic.loglog()` gives the loglikelihood, AIC and BIC values assuming Loglog distribution with parameters alpha and lambda. The function is based on the invariance property of the MLE.

Usage

`abic.loglog(x, alpha.est, lambda.est)`
abic.moee

Arguments

- `x`  
  vector of observations

- `alpha.est`  
  estimate of the parameter alpha

- `lambda.est`  
  estimate of the parameter lambda

Value

The function `abic.loglog()` gives the loglikelihood, AIC and BIC values.

References


See Also

- `qq.loglog` for QQ plot and `ks.loglog` function

Examples

```r
## Load data set
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.9058689 lambda.est = 1.0028228

## Values of AIC, BIC and LogLik for the data(sys2)
abic.loglog(sys2, 0.9058689, 1.0028228)
```

abic.moee  

Akaize information criterion (AIC) and Bayesian information criterion (BIC) for the Marshall-Olkin Extended Exponential(MOEE) distribution

Description

The function `abic.moee()` gives the loglikelihood, AIC and BIC values assuming an MOEE distribution with parameters alpha and lambda.

Usage

```r
abic.moee(x, alpha.est, lambda.est)
```
Arguments

- **x**: vector of observations
- **alpha.est**: estimate of the parameter alpha
- **lambda.est**: estimate of the parameter lambda

Value

The function `abic.moew()` gives the loglikelihood, AIC and BIC values.

References


See Also

`pp.moee` for PP plot and `qq.moee` for QQ plot

Examples

```r
# Load data set
data(stress)
# Estimates of alpha & lambda using 'maxLik' package
# alpha.est = 75.67982, lambda.est = 1.67576
abic.moew(stress, 75.67982, 1.67576)
```

abic.moew  
*Akaike information criterion (AIC) and Bayesian information criterion (BIC) for the Marshall-Olkin Extended Weibull(MOEW) distribution*

Description

The function `abic.moew()` gives the loglikelihood, AIC and BIC values assuming an MOEW distribution with parameters alpha and lambda.

Usage

```r
abic.moew(x, alpha.est, lambda.est)
```

Arguments

- **x**: vector of observations
- **alpha.est**: estimate of the parameter alpha
- **lambda.est**: estimate of the parameter lambda
Value

The function \texttt{abic.moe}\texttt{w}() gives the loglikelihood, AIC and BIC values.

References


See Also

\texttt{pp.moe}\texttt{w} for PP plot and \texttt{qq.moe}\texttt{w} for QQ plot

Examples

```r
## Load data set
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.3035937, lambda.est = 279.2177754

## Values of AIC, BIC and LogLik for the data(sys2)
abic.moe\texttt{w}(sys2, 0.3035937, 279.2177754)
```

---

abic.weibull.ext \textit{Akaike information criterion (AIC) and Bayesian information criterion (BIC) for Weibull Extension(WE) distribution}

Description

The function \texttt{abic.weibull.ext}() gives the loglikelihood, AIC and BIC values assuming a Weibull Extension(WE) distribution with parameters alpha and beta.

Usage

abic.weibull.ext(x, alpha.est, beta.est)

Arguments

- \texttt{x} vector of observations
- \texttt{alpha.est} estimate of the parameter alpha
- \texttt{beta.est} estimate of the parameter beta
Value

The function abic.weibull.ext() gives the loglikelihood, AIC and BIC values.

References


See Also

*pp.weibull.ext* for PP plot and *qq.weibull.ext* for QQ plot

Examples

```r
## Load data sets
data(bearings)

## Maximum Likelihood(ML) Estimates of alpha & beta for the data(bearings)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 0.00019114, beta.est = 0.14696242

## Values of AIC, BIC and LogLik for the data(bearings)
abic.weibull.ext(bearings, 0.00019114, 0.14696242)
```

---

**bearings**

**bearings**

Description

Several data sets related to life test are available in the reliaR package, which have been taken from the literature.

Usage

```r
data(bearings)
```

Format

A vector containing 23 observations.
Details

The data given here arose in tests on endurance of deep groove ball bearings. The data are the number of million revolutions before failure for each of the 23 ball bearings in the life test.

References


Examples

```r
## Load data sets
data(bearings)
## Histogram for bearings
hist(bearings)
```

---

**BurrX**

*The BurrX (Generalized Rayleigh) distribution*

---

Description

Density, distribution function, quantile function and random generation for the BurrX distribution with shape parameter alpha and scale parameter lambda.

Usage

```r
dburrX(x, alpha, lambda, log = FALSE)
pburrX(q, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
qburrX(p, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
rburrX(n, alpha, lambda)
```

Arguments

- `x, q`: vector of quantiles.
- `p`: vector of probabilities.
- `n`: number of observations. If `length(n) > 1`, the length is taken to be the number required.
- `alpha`: shape parameter.
- `lambda`: scale parameter.
- `log, log.p`: logical; if TRUE, probabilities `p` are given as log(p).
- `lower.tail`: logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$. 
Details

The BurrX distribution has density

\[ f(x; \alpha, \lambda) = 2\alpha \lambda^2 x e^{-\lambda x^2} \left\{ 1 - e^{-(\lambda x)^2} \right\}^{\alpha - 1}; (\alpha, \lambda) > 0, x > 0. \]

where \( \alpha \) and \( \lambda \) are the shape and scale parameters, respectively.

Value

dburrX gives the density, pburrX gives the distribution function, qburrX gives the quantile function, and rburrX generates random deviates.

References


See Also

.random.seed about random number; sburrX for BurrX survival / hazard etc. functions

Examples

```r
## Load data sets
data(bearings)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.1989515, lambda.est = 0.0130847

dburrX(bearings, 1.1989515, 0.0130847, log = FALSE)
pburrX(bearings, 1.1989515, 0.0130847, lower.tail = TRUE, log.p = FALSE)
qburrX(0.25, 1.1989515, 0.0130847, lower.tail=TRUE, log.p = FALSE)
rburrX(30, 1.1989515, 0.0130847)
```

BurrXsurvival

Survival related functions for the BurrX distribution

Description

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the BurrX distribution with shape parameter alpha and scale parameter lambda.
Usage

crf.burrX(x, t = 0, alpha, lambda)
hburrX(x, alpha, lambda)
hra.burrX(x, alpha, lambda)
sburrX(x, alpha, lambda)

Arguments

x vector of quantiles.
alpha shape parameter.
lambda scale parameter.
t age component.

Details

The hazard function is defined by

\[ h(x) = \frac{f(x)}{1 - F(x)}, \quad t > 0, 0 < F(x) < 1, \]

where \( f(\cdot) \) and \( F(\cdot) \) are the pdf and cdf, respectively. The behavior of \( h(x) \) allows one to characterize the aging of the units. For example, if the failure rate is increasing (IFR class), then the units age with time. If \( h(x) \) is decreasing (DFR class), then the units improve in performance with time. Finally, if \( h(x) \) is constant, then the lifetime distribution is necessarily exponential.

There are two more aging indicators which are the following:

The failure rate average (FRA) of \( X \) is given by

\[ FRA(x) = \frac{H(x)}{x} = \frac{\int_0^x h(x) \, dx}{x}, \quad x > 0, \]

where \( H(x) \) is the cumulative hazard function. An analysis for \( FRA(x) \) on \( x \) permits to obtain the IFRA and DFRA classes.

The survival/reliability function (s.f.) and the conditional survival of \( X \) are defined by

\[ R(x) = 1 - F(x) \quad \text{and} \quad R(x|t) = \frac{R(x + t)}{R(x)}, \quad x > 0, t > 0, R(\cdot) > 0, \]

respectively, where \( F(\cdot) \) is the cdf of \( X \). Similarly to \( h(x) \) and \( FRA(x) \), the distribution of \( X \) belongs to the new better than used (NBU), exponential, or new worse than used (NWU) classes, when \( R(x|t) < R(x) \), \( R(x|t) = R(x) \), or \( R(x|t) > R(x) \), respectively.

Value

crf.burrX gives the conditional reliability function (crf), hburrX gives the hazard function, hra.burrX gives the hazard rate average (HRA) function, and sburrX gives the survival function for the BurrX distribution.
Chen

The Chen distribution

Description

Density, distribution function, quantile function and random generation for the Chen distribution with shape parameter beta and scale parameter lambda.

References


See Also
dburrX for other BurrX distribution related functions;

Examples

```r
# load data set
data(bearings)

# Maximum Likelihood(ML) Estimates of alpha & lambda for the data(bearings)
# Estimates of alpha & lambda using 'maxLik' package
# alpha.est = 1.1989515, lambda.est = 0.0130847

# Reliability indicators for data(bearings):

# Reliability function
sburrX(bearings, 1.1989515, 0.0130847)

# Hazard function
hburrX(bearings, 1.1989515, 0.0130847)

# hazard rate average(hra)
hra.burrX(bearings, 1.1989515, 0.0130847)

# Conditional reliability function (age component=0)
crf.burrX(bearings, 0.00, 1.1989515, 0.0130847)

# Conditional reliability function (age component=3.0)
crf.burrX(bearings, 3.0, 1.1989515, 0.0130847)
```
Usage

dchen(x, beta, lambda, log = FALSE)
pchen(q, beta, lambda, lower.tail = TRUE, log.p = FALSE)
qchen(p, beta, lambda, lower.tail = TRUE, log.p = FALSE)
rchen(n, beta, lambda)

Arguments

- x, q: vector of quantiles.
- p: vector of probabilities.
- n: number of observations. If `length(n) > 1`, the length is taken to be the number required.
- beta: shape parameter.
- lambda: scale parameter.
- log, log.p: logical; if TRUE, probabilities p are given as log(p).
- lower.tail: logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$.

Details

The Chen distribution has density

$$f(x; \lambda, \beta) = \lambda \beta x^{\beta-1} \exp \left( x^{\beta} \right) \exp \left[ \lambda \left( 1 - \exp \left( x^{\beta} \right) \right) \right] ; \ (\lambda, \beta) > 0, \ x > 0,$$

where $\beta$ and $\lambda$ are the shape and scale parameters, respectively.

Value

dchen gives the density, pchen gives the distribution function, qchen gives the quantile function, and rchen generates random deviates.

References


See Also

`.Random.seed` about random number; `schen` for Chen survival / hazard etc. functions
Examples

```r
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of beta & lambda for the data(sys2)
## beta.est = 0.262282404, lambda.est = 0.007282371
dchen(sys2, 0.262282404, 0.007282371, log = FALSE)
pchen(sys2, 0.262282404, 0.007282371, lower.tail = TRUE,
log.p = FALSE)
qchen(0.25, 0.262282404, 0.007282371, lower.tail = TRUE, log.p = FALSE)
rchen(10, 0.262282404, 0.007282371)
```

Chensurvival  
Survival related functions for the Chen distribution

Description

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the Chen distribution with shape parameter beta and scale parameter lambda.

Usage

```r
crf.chen(x, t = 0, beta, lambda)
hchen(x, beta, lambda)
hra.chen(x, beta, lambda)
schen(x, beta, lambda)
```

Arguments

- `x` vector of quantiles.
- `beta` shape parameter.
- `lambda` scale parameter.
- `t` age component.

Value

- `crf.chen` gives the conditional reliability function (crf), `hchen` gives the hazard function, `hra.chen` gives the hazard rate average (HRA) function, and `schen` gives the survival function for the Chen distribution.

References


See Also
dchen for other Chen distribution related functions

Examples

```r
## Maximum Likelihood(ML) Estimates of beta & lambda
## beta.est = 0.262282404, lambda.est = 0.007282371
## Load data sets
data(sys2)

## Reliability indicators:

## Reliability function
schen(sys2, 0.262282404, 0.007282371)

## Hazard function
hchen(sys2, 0.262282404, 0.007282371)

## hazard rate average(hra)
hra.chen(sys2, 0.262282404, 0.007282371)

## Conditional reliability function (age component=0)
crf.chen(sys2, 0.00, 0.262282404, 0.007282371)

## Conditional reliability function (age component=3.0)
crf.chen(sys2, 3.0, 0.262282404, 0.007282371)
```

---

<table>
<thead>
<tr>
<th>conductors</th>
<th>Accelerated life test data</th>
</tr>
</thead>
</table>

**Description**

Several data sets related to life test are available in the reliaR package, which have been taken from the literature.

**Usage**

data(conductors)

**Format**

A vector containing 59 observations.

**Details**

The data is obtained from Lawless(2003, pp. 267) and it represents the failure times of 59 conductors from an accelerated life test. Failure times are in hours, and there are no censored observations.
References


Examples

```r
## Load data sets
data(controller)
## Histogram for controllers
hist(controller)
```

---

dataset2  

**Controller Dataset**

Description

Several data sets related to life test are available in the reliaR package, which have been taken from the literature.

Usage

```r
data(dataset2)
```

Format

A vector containing 111 observations.

Details

The data is obtained from Lyu(1996) and is given in chapter 11 as DATASET2. The data set contains 36 months of defect-discovery times for a release of Controller Software consisting of about 500,000 lines of code installed on over 100,000 controllers.

References


Examples

```r
## Load data sets
data(dataset2)
## Histogram for dataset2
hist(dataset2)
```
**EPsurvival**

Survival related functions for the Exponential Power (EP) distribution

---

**Description**

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the Exponential Power distribution with shape parameter `alpha` and scale parameter `lambda`.

**Usage**

```r
crf.exp.power(x, t = 0, alpha, lambda)
hexp.power(x, alpha, lambda)
hra.exp.power(x, alpha, lambda)
sexp.power(x, alpha, lambda)
```

**Arguments**

- `x`: vector of quantiles.
- `alpha`: tilt parameter.
- `lambda`: scale parameter.
- `t`: age component.

**Value**

- `crf.exp.power` gives the conditional reliability function (crf), `hexp.power` gives the hazard function, `hra.exp.power` gives the hazard rate average (HRA) function, and `sexp.power` gives the survival function for the Exponential Power distribution.

**References**


**See Also**

- `dexp.power` for other Exponential Power distribution related functions
Examples

```r
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.905868898, lambda.est = 0.001531423

## Reliability indicators:
## Reliability function
s.exp.power(sys2, 0.905868898, 0.001531423)

## Hazard function
h.exp.power(sys2, 0.905868898, 0.001531423)

## Hazard rate average(hra)
hra.exp.power(sys2, 0.905868898, 0.001531423)

## Conditional reliability function (age component=0)
crf.exp.power(sys2, 0.00, 0.905868898, 0.001531423)

## Conditional reliability function (age component=3.0)
crf.exp.power(sys2, 3.0, 0.905868898, 0.001531423)
```

**ExpExt**

*The Exponential Extension (EE) distribution*

**Description**

Density, distribution function, quantile function and random generation for the Exponential Extension (EE) distribution with shape parameter alpha and scale parameter lambda.

**Usage**

```r
dexp.ext(x, alpha, lambda, log = FALSE)
pexp.ext(q, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
qexp.ext(p, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
rexp.ext(n, alpha, lambda)
```

**Arguments**

- `x, q` vector of quantiles.
- `p` vector of probabilities.
- `n` number of observations. If `length(n) > 1`, the length is taken to be the number required.
- `alpha` shape parameter.
- `lambda` scale parameter.
- `log, log.p` logical; if TRUE, probabilities `p` are given as `log(p)`.
- `lower.tail` logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$. 
Details

The Exponential Extension (EE) distribution has density

\[ f(x) = \alpha \lambda (1 + \lambda x)^{\alpha - 1} \exp \left\{ 1 - (1 + \lambda x)^{\alpha} \right\}; \quad x \geq 0, \alpha > 0, \lambda > 0. \]

where \( \alpha \) and \( \lambda \) are the shape and scale parameters, respectively.

Value

dexp.ext gives the density, pexp.ext gives the distribution function, qexp.ext gives the quantile function, and rexp.ext generates random deviates.

References


See Also

.random.seed about random number; sexp.ext for ExpExt survival / hazard etc. functions

Examples

```r
## Load data sets
data(sys2)
## Maximum Likelihood (ML) Estimates of alpha & lambda for the data(sys2)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.0126e+01, lambda.est = 1.5848e-04
dexp.ext(sys2, 1.012556e+01, 1.5848e-04, log = FALSE)
pexp.ext(sys2, 1.012556e+01, 1.5848e-04, lower.tail = TRUE, log.p = FALSE)
qexp.ext(0.25, 1.012556e+01, 1.5848e-04, lower.tail=TRUE, log.p = FALSE)
rexp.ext(30, 1.012556e+01, 1.5848e-04)
```

Description

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the Exponential Extension (EE) distribution with shape parameter alpha and scale parameter lambda.

Usage

```r
crf.exp.ext(x, t = 0, alpha, lambda)
hexp.ext(x, alpha, lambda)
hra.exp.ext(x, alpha, lambda)
sexp.ext(x, alpha, lambda)
```
Arguments

- **x**: vector of quantiles.
- **alpha**: shape parameter.
- **lambda**: scale parameter.
- **t**: age component.

Value

crf.exp.ext gives the conditional reliability function (crf), hexp.ext gives the hazard function, hra.exp.ext gives the hazard rate average (HRA) function, and sexp.ext gives the survival function for the Exponential Extension(EE) distribution.

References


See Also

dexp.ext for other Exponential Extension(EE) distribution related functions;

Examples

```r
## load data set
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.0126e+01, lambda.est = 1.5848e-04

## Reliability indicators for data(sys2):

## Reliability function
sexp.ext(sys2, 1.0126e+01, 1.5848e-04)

## Hazard function
hexp.ext(sys2, 1.0126e+01, 1.5848e-04)

## hazard rate average(hra)
hra.exp.ext(sys2, 1.0126e+01, 1.5848e-04)

## Conditional reliability function (age component=0)
crf.exp.ext(sys2, 0.00, 1.0126e+01, 1.5848e-04)

## Conditional reliability function (age component=3.0)
crf.exp.ext(sys2, 3.0, 1.0126e+01, 1.5848e-04)
```
Description

Density, distribution function, quantile function and random generation for the Exponentiated Logistic(EL) distribution with shape parameter \( \alpha \) and scale parameter \( \beta \).

Usage

\[
\begin{align*}
\text{dexpo.logistic}(x, \alpha, \beta, \log = \text{FALSE}) \\
\text{pexpo.logistic}(q, \alpha, \beta, \text{lower.tail = TRUE, log.p = FALSE}) \\
\text{qexpo.logistic}(p, \alpha, \beta, \text{lower.tail = TRUE, log.p = FALSE}) \\
\text{rexpo.logistic}(n, \alpha, \beta)
\end{align*}
\]

Arguments

- \( x, q \) vector of quantiles.
- \( p \) vector of probabilities.
- \( n \) number of observations. If \( \text{length}(n) > 1 \), the length is taken to be the number required.
- \( \alpha \) shape parameter.
- \( \beta \) scale parameter.
- \( \log, \log.p \) logical; if \( \text{TRUE} \), probabilities \( p \) are given as \( \log(p) \).
- \( \text{lower.tail} \) logical; if \( \text{TRUE} \) (default), probabilities are \( P[X \leq x] \) otherwise, \( P[X > x] \).

Details

The Exponentiated Logistic(EL) distribution has density

\[
f(x; \alpha, \beta) = \frac{\alpha}{\beta} \exp\left(-\frac{x}{\beta}\right) \left\{1 + \exp\left(-\frac{x}{\beta}\right)\right\}^{-(\alpha+1)} \quad ; \quad (\alpha, \beta) > 0, \ x > 0
\]

where \( \alpha \) and \( \beta \) are the shape and scale parameters, respectively.

Value

\text{dexpo.logistic} gives the density, \text{pexpo.logistic} gives the distribution function, \text{qexpo.logistic} gives the quantile function, and \text{rexpo.logistic} generates random deviates.

References


ExpoLogisticsurvival

See Also

.Random.seed about random number; expo.logistic for Exponentiated Logistic(EL) survival / hazard etc. functions

Examples

## Load data sets
data(dataset2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(dataset2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 5.31302, beta.est = 139.04515
dexpo.logistic(dataset2, 5.31302, 139.04515, log = FALSE)
pexpo.logistic(dataset2, 5.31302, 139.04515, lower.tail = TRUE, log.p = FALSE)
qexpo.logistic(0.25, 5.31302, 139.04515, lower.tail=TRUE, log.p = FALSE)
rexpologisticsurvival (30, 5.31302, 139.04515)

ExpoLogisticsurvival  Survival related functions for the Exponentiated Logistic(EL) distribution

Description

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the Exponentiated Logistic(EL) distribution with shape parameter alpha and scale parameter beta.

Usage

crf.expo.logistic(x, t = 0, alpha, beta)
hexpo.logistic(x, alpha, beta)
hra.expo.logistic(x, alpha, beta)
sexpo.logistic(x, alpha, beta)

Arguments

x vector of quantiles.
alpha shape parameter.
beta scale parameter.
t age component.

Value

crf.expo.logistic gives the conditional reliability function (crf), hexpo.logistic gives the hazard function, hra.expo.logistic gives the hazard rate average (HRA) function, and sexpo.logistic gives the survival function for the Exponentiated Logistic(EL) distribution.
The Exponentiated Weibull (EW) distribution

Description

Density, distribution function, quantile function and random generation for the Exponentiated Weibull (EW) distribution with shape parameters alpha and theta.

Usage

dexpo.weibull(x, alpha, theta, log = FALSE)
pexpo.weibull(q, alpha, theta, lower.tail = TRUE, log.p = FALSE)
qexpo.weibull(p, alpha, theta, lower.tail = TRUE, log.p = FALSE)
rexpo.weibull(n, alpha, theta)
ExpoWeibull

Arguments

- `x, q` vector of quantiles.
- `p` vector of probabilities.
- `n` number of observations. If length(n) > 1, the length is taken to be the number required.
- `alpha` shape parameter.
- `theta` shape parameter.
- `log, log.p` logical; if TRUE, probabilities p are given as \( \log(p) \).
- `lower.tail` logical; if TRUE (default), probabilities are \( P[X \leq x] \) otherwise, \( P[X > x] \).

Details

The Exponentiated Weibull (EW) distribution has density

\[
f(x; \alpha, \theta) = \alpha \theta x^{\alpha - 1} e^{-x^\alpha} \{1 - \exp(-x^\alpha)\}^{\theta - 1}; \quad (\alpha, \theta) > 0, x > 0
\]

where \( \alpha \) and \( \theta \) are the shape and scale parameters, respectively.

Value

dexpo.weibull gives the density, pexpo.weibull gives the distribution function, qexpo.weibull gives the quantile function, and rexpo.weibull generates random deviates.

References


See Also

- `.Random.seed` about random number; `sexpo.weibull` for Exponentiated Weibull(EW) survival / hazard etc. functions

Examples

```r
## Load data sets
data(stress)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(stress)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est = 1.026465, theta.est = 7.824943
dexpo.weibull(stress, 1.026465, 7.824943, log = FALSE)
pexpo.weibull(stress, 1.026465, 7.824943, lower.tail = TRUE, log.p = FALSE)
qexpo.weibull(0.25, 1.026465, 7.824943, lower.tail=TRUE, log.p = FALSE)
rexp.weibull(30, 1.026465, 7.824943)
```
ExpoWeibullsurvival

Survival related functions for the Exponentiated Weibull(EW) distribution

Description

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the Exponentiated Weibull(EW) distribution with shape parameters alpha and theta.

Usage

```r
crf.expo.weibull(x, t = 0, alpha, theta)
hexpo.weibull(x, alpha, theta)
hra.expo.weibull(x, alpha, theta)
sexpo.weibull(x, alpha, theta)
```

Arguments

- `x`: vector of quantiles.
- `alpha`: shape parameter.
- `theta`: shape parameter.
- `t`: age component.

Value

`crf.expo.weibull` gives the conditional reliability function (crf), `hexpo.weibull` gives the hazard function, `hra.expo.weibull` gives the hazard rate average (HRA) function, and `sexpo.weibull` gives the survival function for the Exponentiated Weibull(EW) distribution.

References


See Also

dexpo.weibull for other Exponentiated Weibull(EW) distribution related functions;
Examples

```r
## load data set
data(stress)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(stress)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est = 1.026465, theta.est = 7.824943

## Reliability indicators for data(stress):

## Reliability function
se.xpo.weibull(stress, 1.026465, 7.824943)

## Hazard function
he.xpo.weibull(stress, 1.026465, 7.824943)

## hazard rate average(hra)
hra.xpo.weibull(stress, 1.026465, 7.824943)

## Conditional reliability function (age component=0)
crf.xpo.weibull(stress, 0.00, 1.026465, 7.824943)

## Conditional reliability function (age component=3.0)
crf.xpo.weibull(stress, 3.0, 1.026465, 7.824943)
```

ExpPower  

*The Exponential Power distribution*

Description

Density, distribution function, quantile function and random generation for the Exponential Power distribution with shape parameter alpha and scale parameter lambda.

Usage

```r
dexp.power(x, alpha, lambda, log = FALSE)
pexp.power(q, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
qexp.power(p, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
rexp.power(n, alpha, lambda)
```

Arguments

- `x, q` vector of quantiles.
- `p` vector of probabilities.
- `n` number of observations. If length(n) > 1, the length is taken to be the number required.
- `alpha` shape parameter.
- `lambda` scale parameter.
- `log, log.p` logical; if TRUE, probabilities p are given as log(p).
- `lower.tail` logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$. 
Details

The probability density function of exponential power distribution is

\[ f(x; \alpha, \lambda) = \alpha \lambda^\alpha x^{\alpha - 1} e^{(\lambda x)^\alpha} \exp \left\{ 1 - e^{(\lambda x)^\alpha} \right\}; \quad (\alpha, \lambda) > 0, x > 0. \]

where \( \alpha \) and \( \lambda \) are the shape and scale parameters, respectively.

Value

\texttt{dexp.power} gives the density, \texttt{pexp.power} gives the distribution function, \texttt{qexp.power} gives the quantile function, and \texttt{rexp.power} generates random deviates.

References


See Also

\texttt{.Random.seed} about random number; \texttt{sexp.power} for Exponential Power distribution survival / hazard etc. functions;

Examples

```r
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.905868898, lambda.est = 0.001531423
dexp.power(sys2, 0.905868898, 0.001531423, log = FALSE)
pexp.power(sys2, 0.905868898, 0.001531423, lower.tail = TRUE, log.p = FALSE)
qexp.power(0.25, 0.905868898, 0.001531423, lower.tail = TRUE, log.p = FALSE)
rexp.power(30, 0.905868898, 0.001531423)
```

---

**FlexWeibull**

*The flexible Weibull(FW) distribution*

Description

Density, distribution function, quantile function and random generation for the flexible Weibull(FW) distribution with parameters \( \alpha \) and \( \beta \).
Usage

dflex.weibull(x, alpha, beta, log = FALSE)
pflex.weibull(q, alpha, beta, lower.tail = TRUE, log.p = FALSE)
qflex.weibull(p, alpha, beta, lower.tail = TRUE, log.p = FALSE)
rflex.weibull(n, alpha, beta)

Arguments

x, q vector of quantiles.
p vector of probabilities.
n number of observations. If length(n) > 1, the length is taken to be the number required.
alpha parameter.
beta parameter.
log, log.p logical; if TRUE, probabilities p are given as log(p).
lower.tail logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$.

Details

The flexible Weibull (FW) distribution has density

$f(x) = \left( \alpha + \beta \frac{x}{x^2} \right) \exp \left( \alpha x - \beta \frac{x}{x^2} \right) \exp \left( - \exp \left( \alpha x - \beta \frac{x}{x^2} \right) \right); \quad x \geq 0, \alpha > 0, \beta > 0.$

where $\alpha$ and $\beta$ are the shape and scale parameters, respectively.

Value

dflex.weibull gives the density, pflex.weibull gives the distribution function, qflex.weibull gives the quantile function, and rflex.weibull generates random deviates.

References


See Also

.random.seed about random number; sflex.weibull for flexible Weibull(FW) survival / hazard etc. functions

Examples

## Load data sets
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(repairtimes)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 0.07077507, beta.est = 1.13181535
**Survival related functions for the flexible Weibull(FW) distribution**

**Description**

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the flexible Weibull(FW) distribution with parameters alpha and beta.

**Usage**

```r
crf.flex.weibull(x, t = 0, alpha, beta)
hflex.weibull(x, alpha, beta)
hra.flex.weibull(x, alpha, beta)
sflex.weibull(x, alpha, beta)
```

**Arguments**

- `x`: vector of quantiles.
- `alpha`: parameter.
- `beta`: parameter.
- `t`: age component.

**Value**

- `crf.flex.weibull` gives the conditional reliability function (crf), `hflex.weibull` gives the hazard function, `hra.flex.weibull` gives the hazard rate average (HRA) function, and `sflex.weibull` gives the survival function for the flexible Weibull(FW) distribution.

**References**


**See Also**

- `dflex.weibull` for other flexible Weibull(FW) distribution related functions;
Examples

```r
## load data set
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(repairtimes)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 0.07077507, beta.est = 1.13181535

## Reliability indicators for data(repairtimes):

## Reliability function
sflex.weibull(repairtimes, 0.07077507, 1.13181535)

## Hazard function
hflex.weibull(repairtimes, 0.07077507, 1.13181535)

## hazard rate average(hra)
hra.flex.weibull(repairtimes, 0.07077507, 1.13181535)

## Conditional reliability function (age component=0)
crf.flex.weibull(repairtimes, 0.00, 0.07077507, 1.13181535)

## Conditional reliability function (age component=3.0)
crf.flex.weibull(repairtimes, 3.0, 0.07077507, 1.13181535)
```

---

The Generalized Exponential (GE) distribution

Description

Density, distribution function, quantile function and random generation for the Generalized Exponential (GE) distribution with shape parameter \( \alpha \) and scale parameter \( \lambda \).

Usage

```r
dgen.exp(x, alpha, lambda, log = FALSE)
pgen.exp(q, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
qgen.exp(p, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
rgen.exp(n, alpha, lambda)
```

Arguments

- `x, q` vector of quantiles.
- `p` vector of probabilities.
- `n` number of observations. If `length(n) > 1`, the length is taken to be the number required.
- `alpha` shape parameter.
- `lambda` scale parameter.
- `log, log.p` logical; if TRUE, probabilities p are given as \( \log(p) \).
- `lower.tail` logical; if TRUE (default), probabilities are \( P[x \leq x] \) otherwise, \( P[x > x] \).
The generalized exponential distribution has density
\[
f(x; \alpha, \lambda) = \alpha \lambda x e^{-\lambda x} \left\{1 - e^{-\lambda x}\right\}^{\alpha-1}; \quad (\alpha, \lambda) > 0, x > 0.
\]
where \(\alpha\) and \(\lambda\) are the shape and scale parameters, respectively.

\section*{Value}

dgen.exp gives the density, pgen.exp gives the distribution function, qgen.exp gives the quantile function, and rgen.exp generates random deviates.

\section*{References}


\section*{See Also}

.random.seed about random number; sgen.exp for GE survival / hazard etc. functions

\section*{Examples}

```r
## Load data set
data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 5.28321139, lambda.est = 0.03229609

dgen.exp(bearings, 5.28321139, 0.03229609, log = FALSE)
pgen.exp(bearings, 5.28321139, 0.03229609, lower.tail = TRUE, log.p = FALSE)
qgen.exp(0.25, 5.28321139, 0.03229609, lower.tail = TRUE, log.p = FALSE)
rgen.exp(10, 5.28321139, 0.03229609)
```

\section*{Description}

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the Generalized Exponential (GE) distribution with shape parameter \(\alpha\) and scale parameter \(\lambda\).
Usage

```r
crf.gen.exp(x, t = 0, alpha, lambda)
hgen.exp(x, alpha, lambda)
hra.gen.exp(x, alpha, lambda)
sgen.exp(x, alpha, lambda)
```

Arguments

- `x`: vector of quantiles.
- `alpha`: shape parameter.
- `lambda`: scale parameter.
- `t`: age component.

Value

`crf.gen.exp` gives the conditional reliability function (crf), `hgen.exp` gives the hazard function, `hra.gen.exp` gives the hazard rate average (HRA) function, and `sgen.exp` gives the survival function for the GE distribution.

References


See Also

- `dgen.exp` for other GE distribution related functions;

Examples

```r
## load data set
data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 5.28321139, lambda.est = 0.03229609
sgen.exp(bearings, 5.28321139, 0.03229609)
hgen.exp(bearings, 5.28321139, 0.03229609)
hra.gen.exp(bearings, 5.28321139, 0.03229609)
crf.gen.exp(bearings, 20.0, 5.28321139, 0.03229609)
```
The Gompertz distribution

Description
Density, distribution function, quantile function and random generation for the Gompertz distribution with shape parameter alpha and scale parameter theta.

Usage

dgompertz(x, alpha, theta, log = FALSE)
pgomertz(q, alpha, theta, lower.tail = TRUE, log.p = FALSE)
qgompertz(p, alpha, theta, lower.tail = TRUE, log.p = FALSE)
rgompertz(n, alpha, theta)

Arguments

x, q  vector of quantiles.

p  vector of probabilities.

n  number of observations. If length(n) > 1, the length is taken to be the number required.

alpha  shape parameter.

theta  scale parameter.

log, log.p  logical; if TRUE, probabilities p are given as log(p).

lower.tail  logical; if TRUE (default), probabilities are P[X ≤ x] otherwise, P[X > x].

Details
The Gompertz distribution has density

\[ f(x) = \theta e^{\alpha x} \exp \left( \frac{\theta}{\alpha} (1 - e^{\alpha x}) \right) ; x \geq 0, \theta > 0, -\infty < \alpha < \infty. \]

where \( \alpha \) and \( \theta \) are the shape and scale parameters, respectively.

Value
dgompertz gives the density, pgomertz gives the distribution function, qgompertz gives the quantile function, and rgompertz generates random deviates.

References
Gompertzsurvival

See Also

.Random.seed about random number; sgompertz for Gompertz survival/hazard etc. functions

Examples

```r
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(sys2)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est = 0.00121307, theta.est = 0.00173329

dgompertz(sys2, 0.00121307, 0.00173329, log = FALSE)
pgompertz(sys2, 0.00121307, 0.00173329, lower.tail = TRUE, log.p = FALSE)
qgompertz(0.25, 0.00121307, 0.00173329, lower.tail = TRUE, log.p = FALSE)
rgompertz(0.5, 0.00121307, 0.00173329)
```

Gompertzsurvival Survival related functions for the Gompertz distribution

Description

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the Gompertz distribution with shape parameter alpha and scale parameter theta.

Usage

```r
crf.gompertz(x, t = 0, alpha, theta)
hgompertz(x, alpha, theta)
hra.gompertz(x, alpha, theta)
sgompertz(x, alpha, theta)
```

Arguments

- `x`: vector of quantiles.
- `alpha`: shape parameter.
- `theta`: scale parameter.
- `t`: age component.

Value

- `crf.gompertz` gives the conditional reliability function (crf), `hgompertz` gives the hazard function, `hra.gompertz` gives the hazard rate average (HRA) function, and `sgompertz` gives the survival function for the Gompertz distribution.

References

The generalized power Weibull (GPW) distribution

Description

Density, distribution function, quantile function and random generation for the generalized power Weibull (GPW) distribution with shape parameters alpha and theta.

Usage

dgp.weibull(x, alpha, theta, log = FALSE)
pgp.weibull(q, alpha, theta, lower.tail = TRUE, log.p = FALSE)
qgp.weibull(p, alpha, theta, lower.tail = TRUE, log.p = FALSE)
rgp.weibull(n, alpha, theta)

Arguments

x, q
vector of quantiles.

p
vector of probabilities.
Number of observations. If length(n) > 1, the length is taken to be the number required.

Alpha shape parameter.

Theta shape parameter.

Logical; if TRUE, probabilities p are given as log(p).

Logical; if TRUE (default), probabilities are \( P[X \leq x] \) otherwise, \( P[X > x] \).

Details

The generalized power Weibull (GPW) distribution has density

\[
f(x) = \alpha \theta x^{\alpha - 1} (1 + x^\alpha)^{\theta - 1} \exp \left\{ 1 - (1 + x^\alpha)^{\theta} \right\}; \quad x \geq 0, \quad \alpha > 0, \quad \theta > 0.
\]

where \( \alpha \) and \( \theta \) are the shape and scale parameters, respectively.

Value

dgp.weibull gives the density, pgp.weibull gives the distribution function, qgp.weibull gives the quantile function, and rgp.weibull generates random deviates.

References


See Also

.Random.seed about random number; sgp.weibull for generalized power Weibull(GPW) survival / hazard etc. functions

Examples

```r
## Load data sets
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(repairtimes)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est = 1.566093, theta.est = 0.355321

dgp.weibull(repairtimes, 1.566093, 0.355321, log = FALSE)
pgp.weibull(repairtimes, 1.566093, 0.355321, lower.tail = TRUE, log.p = FALSE)
qgp.weibull(0.25, 1.566093, 0.355321, lower.tail=TRUE, log.p = FALSE)
rgp.weibull(30, 1.566093, 0.355321)
```
Description

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the generalized power Weibull(GPW) distribution with shape parameters alpha and theta.

Usage

crf.gp.weibull(x, t = 0, alpha, theta)
hgp.weibull(x, alpha, theta)
hra.gp.weibull(x, alpha, theta)
sgp.weibull(x, alpha, theta)

Arguments

x vector of quantiles.
alpha shape parameter.
theta shape parameter.
t age component.

Value

crf.gp.weibull gives the conditional reliability function (crf), hgp.weibull gives the hazard function, hra.gp.weibull gives the hazard rate average (HRA) function, and sgp.weibull gives the survival function for the generalized power Weibull(GPW) distribution.

References


See Also

dgp.weibull for other generalized power Weibull(GPW) distribution related functions;
Examples

```
## load data set
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(repairtimes)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est = 1.566093, theta.est = 0.355321

## Reliability indicators for data(repairtimes):

## Reliability function
sgp.weibull(repairtimes, 1.566093, 0.355321)

## Hazard function
hgp.weibull(repairtimes, 1.566093, 0.355321)

## hazard rate average(hra)
hra.gp.weibull(repairtimes, 1.566093, 0.355321)

## Conditional reliability function (age component=0)
crf.gp.weibull(repairtimes, 0.00, 1.566093, 0.355321)

## Conditional reliability function (age component=3.0)
crf.gp.weibull(repairtimes, 3.0, 1.566093, 0.355321)
```

---

**Gumbel**

*The Gumbel distribution*

**Description**

Density, distribution function, quantile function and random generation for the Gumbel distribution with location parameter \( \mu \) and scale parameter \( \sigma \).

**Usage**

- `dgumbel(x, mu, sigma, log = FALSE)`
- `pgumbel(q, mu, sigma, lower.tail = TRUE, log.p = FALSE)`
- `qgumbel(p, mu, sigma, lower.tail = TRUE, log.p = FALSE)`
- `rgumbel(n, mu, sigma)`

**Arguments**

- `x, q` vector of quantiles.
- `p` vector of probabilities.
- `n` number of observations. If \( \text{length}(n) > 1 \), the length is taken to be the number required.
- `mu` location parameter.
- `sigma` scale parameter.
- `log, log.p` logical; if TRUE, probabilities \( p \) are given as \( \log(p) \).
- `lower.tail` logical; if TRUE (default), probabilities are \( P[X \leq x] \) otherwise, \( P[X > x] \).
**Details**

The Gumbel distribution has density

\[
f(x) = \frac{1}{\sigma} \exp \left\{ -\frac{x - \mu}{\sigma} \right\} \exp \left[ -\exp \left\{ -\frac{x - \mu}{\sigma} \right\} \right]; \quad -\infty < x < \infty, \sigma > 0.
\]

where \(\mu\) and \(\sigma\) are the shape and scale parameters, respectively.

**Value**

dgumbel gives the density, pgumbel gives the distribution function, qgumbel gives the quantile function, and rgumbel generates random deviates.

**References**


**See Also**

.Random.seed about random number; sgumbel for Gumbel survival / hazard etc. functions

**Examples**

```r
## Load data sets
data(dataset2)
## Maximum Likelihood(ML) Estimates of mu & sigma for the data(dataset2)
## Estimates of mu & sigma using 'maxLik' package
## mu.est = 212.157, sigma.est = 151.768
dgumbel(dataset2, 212.157, 151.768, log = FALSE)
pgumbel(dataset2, 212.157, 151.768, lower.tail = TRUE, log.p = FALSE)
qgumbel(0.25, 212.157, 151.768, lower.tail=TRUE, log.p = FALSE)
rgumbel(30, 212.157, 151.768)
```

---

**Gumbelsurvival**  
*Survival related functions for the Gumbel distribution*

**Description**

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the Gumbel distribution with location parameter \(\mu\) and scale parameter \(\sigma\).

**Usage**

```r
crf.gumbel(x, t = 0, mu, sigma)
hgumbel(x, mu, sigma)
hra.gumbel(x, mu, sigma)
sgumbel(x, mu, sigma)
```
Arguments

- **x**: vector of quantiles.
- **mu**: location parameter.
- **sigma**: scale parameter.
- **t**: age component.

Value

crf.gumbel gives the conditional reliability function (crf), hgumbel gives the hazard function, hra.gumbel gives the hazard rate average (HRA) function, and sgumbel gives the survival function for the Gumbel distribution.

References


See Also

dgumbel for other Gumbel distribution related functions;

Examples

```r
## load data set
data(dataset2)
## Maximum Likelihood(ML) Estimates of mu & sigma for the data(dataset2)
## Estimates of mu & sigma using 'maxLik' package
## mu.est = 212.157, sigma.est = 151.768

## Reliability indicators for data(dataset2):

## Reliability function
sgumbel(dataset2, 212.157, 151.768)

## Hazard function
hgumbel(dataset2, 212.157, 151.768)

## hazard rate average(hra)
hra.gumbel(dataset2, 212.157, 151.768)

## Conditional reliability function (age component=0)
crf.gumbel(dataset2, 0, 00, 212.157, 151.768)

## Conditional reliability function (age component=3.0)
crf.gumbel(dataset2, 3.0, 212.157, 151.768)
```
The Inverse Generalized Exponential (IGE) distribution

Description
Density, distribution function, quantile function and random generation for the Inverse Generalized Exponential (IGE) distribution with shape parameter \( \alpha \) and scale parameter \( \lambda \).

Usage
\[
\begin{align*}
dinv.genexp & : x, \alpha, \lambda, \text{log} = \text{FALSE} \\
pinv.genexp & : q, \alpha, \lambda, \text{lower.tail} = \text{TRUE}, \text{log.p} = \text{FALSE} \\
qinv.genexp & : p, \alpha, \lambda, \text{lower.tail} = \text{TRUE}, \text{log.p} = \text{FALSE} \\
rinv.genexp & : n, \alpha, \lambda
\end{align*}
\]

Arguments
- \( x, q \) : vector of quantiles.
- \( p \) : vector of probabilities.
- \( n \) : number of observations. If \( \text{length}(n) > 1 \), the length is taken to be the number required.
- \( \alpha \) : shape parameter.
- \( \lambda \) : scale parameter.
- \( \text{log}, \text{log.p} \) : logical; if \( \text{TRUE} \), probabilities \( p \) are given as \( \log(p) \).
- \( \text{lower.tail} \) : logical; if \( \text{TRUE} \) (default), probabilities are \( P[X \leq x] \) otherwise, \( P[X > x] \).

Details
The Inverse Generalized Exponential (IGE) distribution has density
\[
f(x; \alpha, \lambda) = \frac{\alpha}{x^2} e^{-\lambda/x} \left( 1 - e^{-\lambda/x} \right)^{\alpha-1}; \quad (\alpha, \lambda) > 0, x > 0
\]
where \( \alpha \) and \( \lambda \) are the shape and scale parameters, respectively.

Value
- \( \text{dinv.genexp} \) gives the density,
- \( \text{pinv.genexp} \) gives the distribution function,
- \( \text{qinv.genexp} \) gives the quantile function,
- \( \text{rinv.genexp} \) generates random deviates.

References

See Also

.Random.seed about random number; .sinv.genexp for Inverse Generalized Exponential(IGE) survival/hazard etc. functions

Examples

## Load data sets
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(repairtimes)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.097807, lambda.est = 1.206889
dinv.genexp(repairtimes, 1.097807, 1.206889, log = FALSE)
pinv.genexp(repairtimes, 1.097807, 1.206889, lower.tail = TRUE, log.p = FALSE)
qinv.genexp(0.25, 1.097807, 1.206889, lower.tail=TRUE, log.p = FALSE)
rinv.genexp(30, 1.097807, 1.206889)

---

Description

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the Inverse Generalized Exponential(IGE) distribution with shape parameter alpha and scale parameter lambda.

Usage

```r
crf.inv.genexp(x, t = 0, alpha, lambda)
hinv.genexp(x, alpha, lambda)
hra.inv.genexp(x, alpha, lambda)
sinv.genexp(x, alpha, lambda)
```

Arguments

- `x`: vector of quantiles.
- `alpha`: shape parameter.
- `lambda`: scale parameter.
- `t`: age component.

Value

`crf.inv.genexp` gives the conditional reliability function (crf), `hinv.genexp` gives the hazard function, `hra.inv.genexp` gives the hazard rate average (HRA) function, and `sinv.genexp` gives the survival function for the Inverse Generalized Exponential(IGE) distribution.
References


See Also

dinv.genexp for other Inverse Generalized Exponential(IGE) distribution related functions;

Examples

```r
## load data set
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(repairtimes)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.097807, lambda.est = 1.206889

## Reliability indicators for data(repairtimes):

## Reliability function
sinv.genexp(repairtimes, 1.097807, 1.206889)

## Hazard function
hinv.genexp(repairtimes, 1.097807, 1.206889)

## hazard rate average(hra)
hra.inv.genexp(repairtimes, 1.097807, 1.206889)

## Conditional reliability function (age component=0)
crf.inv.genexp(repairtimes, 0.00, 1.097807, 1.206889)

## Conditional reliability function (age component=3.0)
crf.inv.genexp(repairtimes, 3.0, 1.097807, 1.206889)
```

---

`ks.burrX` is a function that performs a Test of Kolmogorov-Smirnov for the BurrX distribution.

**Description**

The function `ks.burrX()` gives the values for the KS test assuming a BurrX with shape parameter alpha and scale parameter lambda. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

**Usage**

```r
ks.burrX(x, alpha.est, lambda.est,
         alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
```
Arguments

- **x**: vector of observations.
- **alpha.est**: estimate of the parameter alpha.
- **lambda.est**: estimate of the parameter lambda.
- **alternative**: indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
- **plot**: Logical; if TRUE, the cdf plot is provided.
- **...**: additional arguments to be passed to the underlying plot function.

Details

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

Value

The function `ks.burrx()` carries out the KS test for the BurrX.

References


See Also

- `pp.burrx` for PP plot and `qq.burrx` for QQ plot

Examples

```r
## Load data sets
data(bearings)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.1989515, lambda.est = 0.0130847

ks.burrx(bearings, 1.1989515, 0.0130847, alternative = "two.sided", plot = TRUE)
```
ks.chen

Test of Kolmogorov-Smirnov for the Chen distribution

Description

The function ks.chen() gives the values for the KS test assuming the Chen distribution with shape parameter beta and scale parameter lambda. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

Usage

ks.chen(x, beta.est, lambda.est, alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)

Arguments

x vector of observations.
beta.est estimate of the parameter beta
lambda.est estimate of the parameter lambda
alternative indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
plot Logical; if TRUE, the cdf plot is provided.
... additional arguments to be passed to the underlying plot function.

Details

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

Value

The function ks.chen() carries out the KS test for the Chen.

References


See Also

pp.chen for PP plot and qq.chen for QQ plot
Examples

```r
## Load data sets
data(sys2)
## Estimates of beta & lambda using 'maxLik' package
## beta.est = 0.262282404, lambda.est = 0.007282371

ks.chen(sys2, 0.262282404, 0.007282371, alternative = "two.sided", plot = TRUE)
```

---

**ks.exp.ext**

**Test of Kolmogorov-Smirnov for the Exponential Extension(EE) distribution**

Description

The function `ks.exp.ext()` gives the values for the KS test assuming a Exponential Extension(EE) with shape parameter alpha and scale parameter lambda. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

Usage

```r
ks.exp.ext(x, alpha.est, lambda.est, alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
```

Arguments

- `x` vector of observations.
- `alpha.est` estimate of the parameter alpha
- `lambda.est` estimate of the parameter lambda
- `alternative` indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
- `plot` Logical; if TRUE, the cdf plot is provided.
- `...` additional arguments to be passed to the underlying plot function.

Details

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

Value

The function `ks.exp.ext()` carries out the KS test for the Exponential Extension(EE)

References

ks.exp.power

See Also

pp.exp.ext for PP plot and qq.exp.ext for QQ plot

Examples

```r
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.0126e+01, lambda.est = 1.5848e-04

ks.exp.ext(sys2, 1.0126e+01, 1.5848e-04, alternative = "two.sided", plot = TRUE)
```

Description

The function `ks.exp.power()` gives the values for the KS test assuming an Exponential Power distribution with shape parameter alpha and scale parameter lambda. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

Usage

```r
ks.exp.power(x, alpha.est, lambda.est, alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)```

Arguments

- `x`: vector of observations.
- `alpha.est`: estimate of the parameter alpha
- `lambda.est`: estimate of the parameter lambda
- `alternative`: indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
- `plot`: Logical; if TRUE, the cdf plot is provided.
- `...`: additional arguments to be passed to the underlying plot function.

Details

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.
The function `ks.exp.power()` carries out the KS test for the EP.

**References**


**See Also**

`pp.exp.power` for PP plot and `qq.exp.power` for QQ plot

**Examples**

```r
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.905868898, lambda.est = 0.001531423

ks.exp.power(sys2, 0.905868898, 0.001531423, alternative = "two.sided", plot = TRUE)
```

---

**ks.expo.logistic**

*Test of Kolmogorov-Smirnov for the Exponentiated Logistic (EL) distribution*

The function `ks.expo.logistic()` gives the values for the KS test assuming a Exponentiated Logistic(EL) with shape parameter alpha and scale parameter beta. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

**Usage**

```r
ks.expo.logistic(x, alpha.est, beta.est, alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)```

**Arguments**

- `x` vector of observations.
- `alpha.est` estimate of the parameter alpha
- `beta.est` estimate of the parameter beta
- `alternative` indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
- `plot` Logical; if TRUE, the cdf plot is provided.
- `...` additional arguments to be passed to the underlying plot function.
Details

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

Value

The function ks.expo.logistic() carries out the KS test for the Exponentiated Logistic(EL)

References


See Also

pp.expo.logistic for PP plot and qq.expo.logistic for QQ plot

Examples

```r
## Load data sets
data(dataset2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(dataset2)
## Estimates of alpha & beta using 'maxlik' package
## alpha.est = 5.31302, beta.est = 139.04515
kappa.expo.logistic(dataset2, 5.31302, 139.04515, alternative = "two.sided", plot = TRUE)
```

---

**ks.expo.weibull**  
Test of Kolmogorov-Smirnov for the Exponentiated Weibull(EW) distribution

Description

The function ks.expo.weibull() gives the values for the KS test assuming a Exponentiated Weibull(EW) with shape parameter alpha and scale parameter theta. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

Usage

```r
ks.expo.weibull(x, alpha.est, theta.est,
alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
```
Arguments

- **x**: vector of observations.
- **alpha.est**: estimate of the parameter alpha
- **theta.est**: estimate of the parameter theta
- **alternative**: indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
- **plot**: Logical; if TRUE, the cdf plot is provided.
- **...**: additional arguments to be passed to the underlying plot function.

Details

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

Value

The function `ks.expo.weibull()` carries out the KS test for the Exponentiated Weibull(EW)

References


See Also

- `pp.expo.weibull` for PP plot and `qq.expo.weibull` for QQ plot

Examples

```r
## Load data sets
data(stress)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(stress)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est = 1.026465, theta.est = 7.824943

ks.expo.weibull(stress, 1.026465, 7.824943, alternative = "two.sided", plot = TRUE)
```
ks.flex.weibull  

Test of Kolmogorov-Smirnov for the flexible Weibull(FW) distribution

Description

The function ks.flex.weibull() gives the values for the KS test assuming a flexible Weibull(FW) with shape parameter alpha and scale parameter beta. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

Usage

ks.flex.weibull(x, alpha.est, beta.est, alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)

Arguments

x vector of observations.
alpha.est estimate of the parameter alpha
beta.est estimate of the parameter beta
alternative indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
plot Logical; if TRUE, the cdf plot is provided.
... additional arguments to be passed to the underlying plot function.

Details

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

Value

The function ks.flex.weibull() carries out the KS test for the flexible Weibull(FW)

References


See Also

pp.flex.weibull for PP plot and qq.flex.weibull for QQ plot
Examples

```r
## Load data sets
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(repairtimes)
## Estimates of alpha & beta using 'maxlik' package
## alpha.est = 0.07077507, beta.est = 1.13181535

ks.flex.weibull(repairtimes, 0.07077507, 1.13181535,
alternative = "two.sided", plot = TRUE)
```

---

### ks.gen.exp

**Test of Kolmogorov-Smirnov for the Generalized Exponential(GE) distribution**

**Description**

The function `ks.gen.exp()` gives the values for the KS test assuming an GE with shape parameter alpha and scale parameter lambda. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

**Usage**

```r
ks.gen.exp(x, alpha.est, lambda.est,
alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
```

**Arguments**

- `x`: vector of observations.
- `alpha.est`: estimate of the parameter alpha
- `lambda.est`: estimate of the parameter lambda
- `alternative`: indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
- `plot`: Logical; if TRUE, the cdf plot is provided.
- `...`: additional arguments to be passed to the underlying plot function.

**Details**

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

**Value**

The function `ks.gen.exp()` carries out the KS test for the GE.
ks.gompertz 69

References

See Also
pp.gen.exp for PP plot and qq.gen.exp for QQ plot

Examples

## Load data sets
data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 5.28321139, lambda.est = 0.03229609
ks.gen.exp(bearings, 5.28321139, 0.03229609, alternative = "two.sided", plot = TRUE)

ks.gompertz Test of Kolmogorov-Smirnov for the Gompertz distribution

Description
The function ks.gompertz() gives the values for the KS test assuming a Gompertz with shape parameter alpha and scale parameter theta. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

Usage
ks.gompertz(x, alpha.est, theta.est, alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)

Arguments
x vector of observations.
alpha.est estimate of the parameter alpha
theta.est estimate of the parameter theta
alternative indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
plot Logical; if TRUE, the cdf plot is provided.
... additional arguments to be passed to the underlying plot function.

Details
The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.
Value

The function `ks.gompertz()` carries out the KS test for the Gompertz

References


See Also

`pp.gompertz` for PP plot and `qq.gompertz` for QQ plot

Examples

```r
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(sys2)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est = 0.00121307, theta.est = 0.00173329
ks.gompertz(sys2, 0.00121307, 0.00173329, alternative = "two.sided", plot = TRUE)
```

---

### ks.gp.weibull

Test of Kolmogorov-Smirnov for the generalized power Weibull(GPW)

distribution

Description

The function `ks.gp.weibull()` gives the values for the KS test assuming a generalized power Weibull(GPW) with shape parameter alpha and scale parameter theta. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

Usage

```r
ks.gp.weibull(x, alpha.est, theta.est,
              alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
```

Arguments

- `x` : vector of observations.
- `alpha.est` : estimate of the parameter alpha
- `theta.est` : estimate of the parameter theta
- `alternative` : indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
- `plot` : Logical; if TRUE, the cdf plot is provided.
- `...` : additional arguments to be passed to the underlying plot function.
ks.gumbel

Details

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

Value

The function ks.gp.weibull() carries out the KS test for the generalized power Weibull(GPW)

References


See Also

pp.gp.weibull for PP plot and qq.gp.weibull for QQ plot

Examples

```r
### Load data sets
data(repairtimes)
### Maximum Likelihood(ML) Estimates of alpha & theta for the data(repairtimes)
### Estimates of alpha & theta using 'maxLik' package
### alpha.est = 1.566093, theta.est = 0.355321
ks.gp.weibull(repairtimes, 1.566093, 0.355321, alternative = "two.sided", plot = TRUE)
```

ks.gumbel

Test of Kolmogorov-Smirnov for the Gumbel distribution

Description

The function ks.gumbel() gives the values for the KS test assuming a Gumbel with shape parameter mu and scale parameter sigma. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

Usage

ks.gumbel(x, mu.est, sigma.est,
          alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
Arguments

- **x**: vector of observations.
- **mu.est**: estimate of the parameter mu.
- **sigma.est**: estimate of the parameter sigma.
- **alternative**: indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
- **plot**: Logical; if TRUE, the cdf plot is provided.
- **...**: additional arguments to be passed to the underlying plot function.

Details

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

Value

The function `ks.gumbel()` carries out the KS test for the Gumbel.

References


See Also

- `pp.gumbel` for PP plot and `qq.gumbel` for QQ plot

Examples

```r
## Load data sets
data(dataset2)
## Maximum Likelihood(ML) Estimates of mu & sigma for the data(dataset2)
## Estimates of mu & sigma using 'maxLik' package
## mu.est = 212.157, sigma.est = 151.768
ks.gumbel(dataset2, 212.157, 151.768, alternative = "two.sided", plot = TRUE)
```

Test of Kolmogorov-Smirnov for the Inverse Generalized Exponential(IGE) distribution

Description

The function `ks.inv.genexp()` gives the values for the KS test assuming a Inverse Generalized Exponential(IGE) with shape parameter alpha and scale parameter lambda. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.
ks.inv.genexp

Usage

ks.inv.genexp(x, alpha.est, lambda.est, alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)

Arguments

x vector of observations.
alpha.est estimate of the parameter alpha
lambda.est estimate of the parameter lambda
alternative indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
plot Logical; if TRUE, the cdf plot is provided.
... additional arguments to be passed to the underlying plot function.

Details

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

Value

The function ks.inv.genexp() carries out the KS test for the Inverse Generalized Exponential(IGE)

References


See Also

pp.inv.genexp for PP plot and qq.inv.genexp for QQ plot

Examples

## Load data sets
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(repairtimes)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.097807, lambda.est = 1.206889

ks.inv.genexp(repairtimes, 1.097807, 1.206889, alternative = "two.sided", plot = TRUE)
**ks.lfr**

Test of Kolmogorov-Smirnov for the linear failure rate (LFR) distribution

---

**Description**

The function `ks.lfr()` gives the values for the KS test assuming a linear failure rate (LFR) with shape parameter alpha and scale parameter beta. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

**Usage**

```r
ks.lfr(x, alpha.est, beta.est,
       alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
```

**Arguments**

- `x` : vector of observations.
- `alpha.est` : estimate of the parameter alpha
- `beta.est` : estimate of the parameter beta
- `alternative` : indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
- `plot` : Logical; if TRUE, the cdf plot is provided.
- `...` : additional arguments to be passed to the underlying plot function.

**Details**

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

**Value**

The function `ks.lfr()` carries out the KS test for the linear failure rate (LFR).

**References**


**See Also**

`pp.lfr` for PP plot and `qq.lfr` for QQ plot
Examples

```r
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(sys2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 1.77773e-03, beta.est = 2.77764e-06
ks.lfr(sys2, 1.777673e-03, 2.77764e-06, alternative = "two.sided", plot = TRUE)
```

ks.log.gamma Test of Kolmogorov-Smirnov for the log-gamma(LG) distribution

Description

The function `ks.log.gamma()` gives the values for the KS test assuming a log-gamma(LG) with shape parameter alpha and scale parameter lambda. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

Usage

```r
ks.log.gamma(x, alpha.est, lambda.est, alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
```

Arguments

- `x`: vector of observations.
- `alpha.est`: estimate of the parameter alpha
- `lambda.est`: estimate of the parameter lambda
- `alternative`: indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
- `plot`: Logical; if TRUE, the cdf plot is provided.
- `...`: additional arguments to be passed to the underlying plot function.

Details

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

Value

The function `ks.log.gamma()` carries out the KS test for the log-gamma(LG)

References


See Also

*pp.log.gamma* for PP plot and *qq.log.gamma* for QQ plot

Examples

```r
## Load data sets
data(conductors)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(conductors)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 0.0088741, lambda.est = 0.6059935

ks.log.gamma(conductors, 0.0088741, 0.6059935, alternative = "two.sided", plot = TRUE)
```

---

### Description

The function `ks.logis.exp()` gives the values for the KS test assuming a Logistic-Exponential(LE) with shape parameter alpha and scale parameter lambda. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

### Usage

```r
ks.logis.exp(x, alpha.est, lambda.est, alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
```

### Arguments

- **x**: vector of observations.
- **alpha.est**: estimate of the parameter alpha
- **lambda.est**: estimate of the parameter lambda
- **alternative**: indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
- **plot**: Logical; if TRUE, the cdf plot is provided.
- **...**: additional arguments to be passed to the underlying plot function.

### Details

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

### Value

The function `ks.logis.exp()` carries out the KS test for the Logistic-Exponential(LE)
ks.logis.rayleigh

References


See Also

pp.logis.exp for PP plot and qq.logis.exp for QQ plot

Examples

```r
## Load data sets
data(bearings)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 2.36754, lambda.est = 0.01059

ks.logis.exp(bearings, 2.36754, 0.01059, alternative = "two.sided", plot = TRUE)
```

ks.logis.rayleigh

 **Test of Kolmogorov-Smirnov for the Logistic-Rayleigh(LR) distribution**

Description

The function `ks.logis.rayleigh()` gives the values for the KS test assuming a Logistic-Rayleigh(LR) with shape parameter alpha and scale parameter lambda. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

Usage

```r
ks.logis.rayleigh(x, alpha.est, lambda.est,
                   alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
```

Arguments

- `x`: vector of observations.
- `alpha.est`: estimate of the parameter alpha
- `lambda.est`: estimate of the parameter lambda
- `alternative`: indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
- `plot`: Logical; if TRUE, the cdf plot is provided.
- `...`: additional arguments to be passed to the underlying plot function.
Details

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

Value

The function `ks.logis.rayleigh()` carries out the KS test for the Logistic-Rayleigh(LR)

References


See Also

`pp.logis.rayleigh` for PP plot and `qq.logis.rayleigh` for QQ plot

Examples

```r
## Load data sets
data(stress)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(stress)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.4779388, lambda.est = 0.2141343

ks.logis.rayleigh(stress, 1.4779388, 0.2141343,
                  alternative = "two.sided", plot = TRUE)
```

---

**ks.loglog**

*Test of Kolmogorov-Smirnov for the Loglog distribution*

Description

The function `ks.loglog()` gives the values for the KS test assuming the Loglog distribution with shape parameter alpha and scale parameter lambda. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

Usage

```r
ks.loglog(x, alpha.est, lambda.est,
          alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
```
Arguments

- `x` vector of observations.
- `alpha.est` estimate of the parameter alpha
- `lambda.est` estimate of the parameter lambda
- `alternative` indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
- `plot` Logical; if TRUE, the cdf plot is provided.
- `...` additional arguments to be passed to the underlying plot function.

Details

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

Value

The function `ks.loglog()` carries out the KS test for the Loglog.

References


See Also

`pp.loglog` for PP plot and `qq.loglog` for QQ plot

Examples

```r
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.9058689 lambda.est = 1.0028228

ks.loglog(sys2, 0.9058689, 1.0028228, alternative = "two.sided", plot = TRUE)
```

**ks.moee**

*Test of Kolmogorov-Smirnov for the Marshall-Olkin Extended Exponential(MOEE) distribution*

Description

The function `ks.moee()` gives the values for the KS test assuming an GE with tilt parameter alpha and scale parameter lambda. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.
Usage

ks.moee(x, alpha.est, lambda.est,
    alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)

Arguments

  x  vector of observations.
  alpha.est  estimate of the parameter alpha
  lambda.est  estimate of the parameter lambda
  alternative  indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
  plot  Logical; if TRUE, the cdf plot is provided.
  ...  additional arguments to be passed to the underlying plot function.

Details

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

Value

The function ks.moee() carries out the KS test for the MOEE

References


See Also

pp.moee for PP plot and qq.moee for QQ plot

Examples

```r
## Load dataset
data(stress)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 75.67982, lambda.est = 1.67576

ks.moee(stress, 75.67982, 1.67576, alternative = "two.sided", plot = TRUE)
```
ks.moew  

Test of Kolmogorov-Smirnov for the Marshall-Olkin Extended Exponential (MOEW) distribution

Description

The function ks.moew() gives the values for the KS test assuming a MOEW with shape parameter alpha and tilt parameter lambda. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

Usage

ks.moew(x, alpha.est, lambda.est, alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)

Arguments

x  
vector of observations.

alpha.est  
estimate of the parameter alpha

lambda.est  
estimate of the parameter lambda

alternative  
indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".

plot  
Logical; if TRUE, the cdf plot is provided.

...  
additional arguments to be passed to the underlying plot function.

Details

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

Value

The function ks.moew() carries out the KS test for the MOEW

References


See Also

pp.moew for PP plot and qq.moew for QQ plot
Examples

```r
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.3035937, lambda.est = 279.2177754

ks.moew(sys2, 0.3035937, 279.2177754, alternative = "two.sided", plot = TRUE)
```

### Description

The function `ks.weibull.ext()` gives the values for the KS test assuming a Weibull Extension(WE) with shape parameter alpha and scale parameter beta. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

### Usage

```r
ks.weibull.ext(x, alpha.est, beta.est,
    alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
```

### Arguments

- **x**: vector of observations.
- **alpha.est**: estimate of the parameter alpha
- **beta.est**: estimate of the parameter beta
- **alternative**: indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
- **plot**: Logical; if TRUE, the cdf plot is provided.
- **...**: additional arguments to be passed to the underlying plot function.

### Details

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

### Value

The function `ks.weibull.ext()` carries out the KS test for the Weibull Extension(WE)

### References


See Also

pp.weibull.ext for PP plot and qq.weibull.ext for QQ plot

Examples

```r
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(sys2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 0.00019114, beta.est = 0.14696242

ks.weibull.ext(sys2, 0.00019114, 0.14696242, alternative = "two.sided", plot = TRUE)
```

LFR

The linear failure rate(LFR) distribution

Description

Density, distribution function, quantile function and random generation for the linear failure rate(LFR) distribution with parameters alpha and beta.

Usage

```r
dlfr(x, alpha, beta, log = FALSE)
plfr(q, alpha, beta, lower.tail = TRUE, log.p = FALSE)
qlfr(p, alpha, beta, lower.tail = TRUE, log.p = FALSE)
rlfr(n, alpha, beta)
```

Arguments

- `x, q` vector of quantiles.
- `p` vector of probabilities.
- `n` number of observations. If length(n) > 1, the length is taken to be the number required.
- `alpha` parameter.
- `beta` parameter.
- `log, log.p` logical; if TRUE, probabilities p are given as log(p).
- `lower.tail` logical; if TRUE (default), probabilities are P[X ≤ x] otherwise, P[X > x].

Details

The linear failure rate(LFR) distribution has density

\[
f(x) = (\alpha + \beta x) \exp\left\{-\left(\alpha x + \frac{\beta x^2}{2}\right)\right\}; \quad x ≥ 0, \alpha > 0, \beta > 0.
\]

where \(\alpha\) and \(\beta\) are the shape and scale parameters, respectively.
Value

dlf is the density, pf gives the distribution function, qfl gives the quantile function, and rlf generates random deviates.

References


See Also

.Random.seed about random number; slfr for linear failure rate(LFR) survival / hazard etc.

Examples

```r
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(sys2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 1.77773e-03,  beta.est = 2.77764e-06

dlf(sys2, 1.777673e-03, 2.777640e-06, log = FALSE)
plf(sys2, 1.777673e-03, 2.777640e-06, lower.tail = TRUE, log.p = FALSE)
qflr(0.25, 1.777673e-03, 2.777640e-06, lower.tail=TRUE, log.p = FALSE)
rlfr(30, 1.777673e-03, 2.777640e-06)
```

LFRsurvival

*Survival related functions for the linear failure rate(LFR) distribution*

Description

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the linear failure rate(LFR) distribution with parameters alpha and beta.

Usage

```r
crf.lfr(x, t = 0, alpha, beta)
hlfr(x, alpha, beta)
hra.lfr(x, alpha, beta)
slfr(x, alpha, beta)
```
Arguments

- **x**: vector of quantiles.
- **alpha**: parameter.
- **beta**: parameter.
- **t**: age component.

Value

crf.lfr gives the conditional reliability function (crf), hlfr gives the hazard function, hra.lfr gives the hazard rate average (HRA) function, and slfr gives the survival function for the linear failure rate (LFR) distribution.

References


See Also
dlfr for other linear failure rate (LFR) distribution related functions;

Examples

```r
## load data set
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(sys2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 1.77773e-03, beta.est = 2.77764e-06

## Reliability indicators for data(sys2):

## Reliability function
slfr(sys2, 1.777673e-03, 2.777640e-06)

## Hazard function
hlfr(sys2, 1.777673e-03, 2.777640e-06)

## hazard rate average(hra)
hra.lfr(sys2, 1.777673e-03, 2.777640e-06)

## Conditional reliability function (age component=0)
crf.lfr(sys2, 0.00, 1.777673e-03, 2.777640e-06)

## Conditional reliability function (age component=3.0)
crf.lfr(sys2, 3.0, 1.777673e-03, 2.777640e-06)
```
Loggamma

The log-gamma(LG) distribution

Description
Density, distribution function, quantile function and random generation for the log-gamma(LG) distribution with parameters alpha and lambda.

Usage

dlog.gamma(x, alpha, lambda, log = FALSE)
plog.gamma(q, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
qlog.gamma(p, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
rlog.gamma(n, alpha, lambda)

Arguments
x, q vector of quantiles.
p vector of probabilities.
n number of observations. If length(n) > 1, the length is taken to be the number required.
alpha parameter.
lambda parameter.
log, log.p logical; if TRUE, probabilities p are given as log(p).
lower.tail logical; if TRUE (default), probabilities are \( P[X \leq x] \) otherwise, \( P[X > x] \).

Details
The log-gamma(LG) distribution has density
\[
f(x; \alpha, \lambda) = \alpha \lambda \exp \{ \lambda x \} \exp \{ -\alpha \exp \lambda x \}; \ (\alpha, \lambda) > 0, x > 0
\]
where \( \alpha \) and \( \lambda \) are the parameters, respectively.

Value
dlog.gamma gives the density, plog.gamma gives the distribution function, qlog.gamma gives the quantile function, and rlog.gamma generates random deviates.

References
See Also

.random.seed about random number; 

Examples

## Load data sets
data(conductors)

## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(conductors)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 0.0088741, lambda.est = 0.6059935
dlog.gamma(conductors, 0.0088741, 0.6059935, log = FALSE)
plog.gamma(conductors, 0.0088741, 0.6059935, lower.tail = TRUE, log.p = FALSE)
qlog.gamma(0.25, 0.0088741, 0.6059935, lower.tail = TRUE, log.p = FALSE)
rlog.gamma(30, 0.0088741, 0.6059935)

Loggammasurvival

Survival related functions for the log-gamma(LG) distribution

Description

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the log-gamma(LG) distribution with shape parameters alpha and lambda.

Usage

```
crf.log.gamma(x, t = 0, alpha, lambda)
hlog.gamma(x, alpha, lambda)
hra.log.gamma(x, alpha, lambda)
slog.gamma(x, alpha, lambda)
```

Arguments

- **x**: vector of quantiles.
- **alpha**: parameter.
- **lambda**: parameter.
- **t**: age component.

Value

- **crf.log.gamma** gives the conditional reliability function (crf),
- **hlog.gamma** gives the hazard function,
- **hra.log.gamma** gives the hazard rate average (HRA) function,
- **slog.gamma** gives the survival function for the log-gamma(LG) distribution.
LogisExp

The Logistic-Exponential(LE) distribution

Description

Density, distribution function, quantile function and random generation for the Logistic-Exponential(LE) distribution with shape parameter \( \alpha \) and scale parameter \( \lambda \).

Usage

dlogis.exp(x, alpha, lambda, log = FALSE)
plogis.exp(q, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
qlogis.exp(p, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
rlogis.exp(n, alpha, lambda)

References


See Also
dlog.gamma for other log-gamma(LG) distribution related functions;

Examples

```r
## load data set
data(conductors)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(conductors)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 0.0088741, lambda.est = 0.6059935

## Reliability indicators for data(conductors):

## Reliability function
slog.gamma(conductors, 0.0088741, 0.6059935)

## Hazard function
hlog.gamma(conductors, 0.0088741, 0.6059935)

## hazard rate average(hra)
hra.log.gamma(conductors, 0.0088741, 0.6059935)

## Conditional reliability function (age component=0)
crf.log.gamma(conductors, 0.00, 0.0088741, 0.6059935)

## Conditional reliability function (age component=3.0)
crf.log.gamma(conductors, 3.0, 0.0088741, 0.6059935)
```
Arguments

- `x`, `q` vector of quantiles.
- `p` vector of probabilities.
- `n` number of observations. If `length(n) > 1`, the length is taken to be the number required.
- `alpha` shape parameter.
- `lambda` scale parameter.
- `log`, `log.p` logical; if TRUE, probabilities `p` are given as log(p).
- `lower.tail` logical; if TRUE (default), probabilities are \( P[X \leq x] \) otherwise, \( P[X > x] \).

Details

The Logistic-Exponential(LE) distribution has density

\[
f(x) = \frac{\lambda \alpha e^{\lambda x} (e^{\lambda x} - 1)^{\alpha - 1}}{(1 + (e^{\lambda x} - 1)^\alpha)^2}; \ x \geq 0, \ \alpha > 0, \ \lambda > 0.
\]

where \( \alpha \) and \( \lambda \) are the shape and scale parameters, respectively.

Value

dlogis.exp gives the density, plogis.exp gives the distribution function, qlogis.exp gives the quantile function, and rlogis.exp generates random deviates.

References


See Also

.Random.seed about random number; slogis.exp for ExpExt survival / hazard etc. functions

Examples

```r
## Load data sets
data(bearings)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 2.36754, lambda.est = 0.01059
dlogis.exp(bearings, 2.36754, 0.01059, log = FALSE)
plogis.exp(bearings, 2.36754, 0.01059, lower.tail = TRUE, log.p = FALSE)
qlogis.exp(0.25, 2.36754, 0.01059, lower.tail=TRUE, log.p = FALSE)
rlogis.exp(30, 2.36754, 0.01059)
```
LogisExpsurvival  

Survival related functions for the Logistic-Exponential(LE) distribution

Description

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the Logistic-Exponential(LE) distribution with shape parameter \( \alpha \) and scale parameter \( \lambda \).

Usage

- \texttt{crf.logis.exp(x, t = 0, alpha, lambda)}
- \texttt{hlogis.exp(x, alpha, lambda)}
- \texttt{hra.logis.exp(x, alpha, lambda)}
- \texttt{slogis.exp(x, alpha, lambda)}

Arguments

- \texttt{x} vector of quantiles.
- \texttt{alpha} shape parameter.
- \texttt{lambda} scale parameter.
- \texttt{t} age component.

Value

- \texttt{crf.logis.exp} gives the conditional reliability function (crf), \texttt{hlogis.exp} gives the hazard function, \texttt{hra.logis.exp} gives the hazard rate average (HRA) function, and \texttt{slogis.exp} gives the survival function for the Logistic-Exponential(LE) distribution.

References


See Also

- \texttt{dlogis.exp} for other Logistic-Exponential(LE) distribution related functions;

Examples

```r
## load data set
data(bearings)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 2.36754, lambda.est = 0.01059

## Reliability indicators for data(bearings):
```
LogisRayleigh

The Logistic-Rayleigh (LR) distribution

Description

Density, distribution function, quantile function and random generation for the Logistic-Rayleigh (LR) distribution with shape parameter $\alpha$ and scale parameter $\lambda$.

Usage

dlogis.rayleigh(x, alpha, lambda, log = FALSE)
plogis.rayleigh(q, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
qlogis.rayleigh(p, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
rlogis.rayleigh(n, alpha, lambda)

Arguments

- **x, q** vector of quantiles.
- **p** vector of probabilities.
- **n** number of observations. If length(n) > 1, the length is taken to be the number required.
- **alpha** shape parameter.
- **lambda** scale parameter.
- **log, log.p** logical; if TRUE, probabilities $p$ are given as log($p$).
- **lower.tail** logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$.

Details

The cumulative distribution function (cdf) of Logistic-Rayleigh (LR) is given by

$$F(x) = 1 - \frac{1}{1 + \left( e^{(\lambda x^2/2)} - 1 \right)^{\alpha}}; \quad x \geq 0, \alpha > 0, \lambda > 0.$$ 

where $\alpha$ and $\lambda$ are the shape and scale parameters, respectively.
Value

dlogis.rayleigh gives the density, plogis.rayleigh gives the distribution function, qlogis.rayleigh gives the quantile function, and rlogis.rayleigh generates random deviates.

References


See Also

.Random.seed about random number; slogis.rayleigh for ExpExt survival / hazard etc. functions

Examples

```r
## Load data sets
data(stress)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(stress)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.4779388, lambda.est = 0.2141343
dlogis.rayleigh(stress, 1.4779388, 0.2141343, log = FALSE)
plogis.rayleigh(stress, 1.4779388, 0.2141343, lower.tail = TRUE, log.p = FALSE)
qlogis.rayleigh(0.25, 1.4779388, 0.2141343, lower.tail=TRUE, log.p = FALSE)
rlogis.rayleigh(30, 1.4779388, 0.2141343)
```

Description

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the Logistic-Rayleigh(LR) distribution with shape parameter alpha and scale parameter lambda.

Usage

```r
crf.logis.rayleigh(x, t = 0, alpha, lambda)
hlogis.rayleigh(x, alpha, lambda)
hra.logis.rayleigh(x, alpha, lambda)
slogis.rayleigh(x, alpha, lambda)
```

Arguments

- `x` vector of quantiles.
- `alpha` shape parameter.
- `lambda` scale parameter.
- `t` age component.
Loglog

Value

crf.logis.rayleigh gives the conditional reliability function (crf), hlogis.rayleigh gives the hazard function, hra.logis.rayleigh gives the hazard rate average (HRA) function, and slogis.rayleigh gives the survival function for the Logistic-Rayleigh(LR) distribution.

References


See Also
dlogis.rayleigh for other Logistic-Rayleigh(LR) distribution related functions;

Examples

```r
## load data set
data(stress)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(stress)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.4779388, lambda.est = 0.2141343

## Reliability indicators for data(stress):

## Reliability function
slogis.rayleigh(stress, 1.4779388, 0.2141343)

## Hazard function
hlogis.rayleigh(stress, 1.4779388, 0.2141343)

## hazard rate average(hra)
hra.logis.rayleigh(stress, 1.4779388, 0.2141343)

## Conditional reliability function (age component=0)
crf.logis.rayleigh(stress, 0.00, 1.4779388, 0.2141343)

## Conditional reliability function (age component=3.0)
crf.logis.rayleigh(stress, 3.0, 1.4779388, 0.2141343)
```

Description

Density, distribution function, quantile function and random generation for the Loglog distribution with shape parameter alpha and scale parameter lambda.
Usage

dloglog(x, alpha, lambda, log = FALSE)
ploglog(q, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
qloglog(p, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
rloglog(n, alpha, lambda)

Arguments

- x, q: vector of quantiles.
- p: vector of probabilities.
- n: number of observations. If length(n) > 1, the length is taken to be the number required.
- alpha: shape parameter.
- lambda: scale parameter.
- log, log.p: logical; if TRUE, probabilities p are given as log(p).
- lower.tail: logical; if TRUE (default), probabilities are \( P[X \leq x] \) otherwise, \( P[X > x] \).

Details

The loglog(Pham) distribution has density

\[
f(x) = \alpha \ln(\lambda) x^{\alpha - 1} \lambda x^\alpha \exp\left\{1 - \lambda x^\alpha\right\}; \quad x > 0, \lambda > 0, \alpha > 0
\]

where \( \alpha \) and \( \lambda \) are the shape and scale parameters, respectively. (Pham, 2002)

Value

dloglog gives the density, ploglog gives the distribution function, qloglog gives the quantile function, and rloglog generates random deviates.

References


See Also

.random.seed about random number; sloglog for Loglog survival / hazard etc. functions;

Examples

data(sys2)

# Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
# alpha.est = 0.9058689 lambda.est = 1.0028228

dloglog(sys2, 0.9058689, 1.0028228, log = FALSE)
### Description

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the Loglog distribution with shape parameter \( \alpha \) and scale parameter \( \lambda \).

### Usage

- `crf.loglog(x, t = 0, alpha, lambda)`
- `hloglog(x, alpha, lambda)`
- `hra.loglog(x, alpha, lambda)`
- `sloglog(x, alpha, lambda)`

### Arguments

- **x**: vector of quantiles.
- **alpha**: shape parameter.
- **lambda**: scale parameter.
- **t**: age component.

### Value

- `crf.loglog` gives the conditional reliability function (crf), `hloglog` gives the hazard function, `hra.loglog` gives the hazard rate average (HRA) function, and `sloglog` gives the survival function for the Loglog distribution.

### References


### See Also

- `dloglog` for other Loglog(Pham) distribution related functions;
Examples

```r
## load data set
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.9058689 lambda.est = 1.0028228

## Reliability indicators for data(sys2):

## Reliability function
sloglog(sys2, 0.9058689, 1.0028228)

## Hazard function
hloglog(sys2, 0.9058689, 1.0028228)

## hazard rate average(hra)
hra.loglog(sys2, 0.9058689, 1.0028228)

## Conditional reliability function (age component=0)
crf.loglog(sys2, 0.00, 0.9058689, 1.0028228)

## Conditional reliability function (age component=3.0)
crf.loglog(sys2, 3.0, 0.9058689, 1.0028228)
```

---

MOEE

The Marshall-Olkin Extended Exponential (MOEE) distribution

Description

Density, distribution function, quantile function and random generation for the Marshall-Olkin Extended Exponential (MOEE) distribution with tilt parameter alpha and scale parameter lambda.

Usage

```r
dmoee(x, alpha, lambda, log = FALSE)
epmoe(q, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
qmoee(p, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
rmoee(n, alpha, lambda)
```

Arguments

- `x, q` vector of quantiles.
- `p` vector of probabilities.
- `n` number of observations. If length(n) > 1, the length is taken to be the number required.
- `alpha` tilt parameter.
- `lambda` scale parameter.
- `log, log.p` logical; if TRUE, probabilities p are given as log(p).
- `lower.tail` logical; if TRUE (default), probabilities are \( P[X \leq x] \) otherwise, \( P[X > x] \).
Details

The Marshall-Olkin extended exponential (MOEE) distribution has density

\[ f(x; \alpha, \lambda) = \frac{\alpha \lambda e^{-\lambda x}}{\left(1 - (1 - \alpha) e^{-\lambda x}\right)^2}; \quad x > 0, \lambda > 0, \alpha > 0 \]

where \( \alpha \) and \( \lambda \) are the tilt and scale parameters, respectively.

Value

dmoee gives the density, pmoee gives the distribution function, qmoee gives the quantile function, and rmoee generates random deviates.

References


See Also

.Random.seed about random number; smoee for MOEE survival / hazard etc. functions

Examples

```r
## Load data sets
data(stress)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 75.67982, lambda.est = 1.67576
dmoee(stress, 75.67982, 1.67576, log = FALSE)
pmoee(stress, 75.67982, 1.67576, lower.tail = TRUE, log.p = FALSE)
qmoee(0.25, 0.4, 2.0, lower.tail = TRUE, log.p = FALSE)
rmoee(10, 75.67982, 1.67576)
```

MOEEsurvival

Survival related functions for the Marshall-Olkin Extended Exponential (MOEE) distribution

Description

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the Marshall-Olkin Extended Exponential (MOEE) distribution with tilt parameter \( \alpha \) and scale parameter \( \lambda \).
MOEEsurvival

Usage

```r
 crf.moee(x, t = 0, alpha, lambda)
 hmoee(x, alpha, lambda)
 hra.moee(x, alpha, lambda)
 smoee(x, alpha, lambda)
```

Arguments

- `x` vector of quantiles.
- `alpha` tilt parameter.
- `lambda` scale parameter.
- `t` age component.

Value

crf.moee gives the conditional reliability function (crf), hmoee gives the hazard function, hra.moee gives the hazard rate average (HRA) function, and smoee gives the survival function for the MOEE distribution.

References


See Also

dmoee for other MOEE distribution related functions;

Examples

```r
## Load data sets
data(stress)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 75.67982, lambda.est = 1.67576
smoee(stress, 75.67982, 1.67576)
hmoee(stress, 75.67982, 1.67576)
hra.moee(stress, 75.67982, 1.67576)
crf.moee(stress, 3.00, 75.67982, 1.67576)
```
MOEW

The Marshall-Olkin Extended Weibull (MOEW) distribution

Description

Density, distribution function, quantile function and random generation for the Marshall-Olkin Extended Weibull (MOEW) distribution with tilt parameter alpha and scale parameter lambda.

Usage

dmoew(x, alpha, lambda, log = FALSE)

pmoew(q, alpha, lambda, lower.tail = TRUE, log.p = FALSE)

qmoew(p, alpha, lambda, lower.tail = TRUE, log.p = FALSE)

rmoew(n, alpha, lambda)

Arguments

x, q vector of quantiles.
p vector of probabilities.
n number of observations. If length(n) > 1, the length is taken to be the number required.
alpha shape parameter.
lambda tilt parameter.
log, log.p logical; if TRUE, probabilities p are given as log(p).
lower.tail logical; if TRUE (default), probabilities are \( P[X \leq x] \) otherwise, \( P[X > x] \).

Details

The Marshall-Olkin extended Weibull (MOEW) distribution has density

\[
f(x) = \frac{\lambda \alpha x^{\alpha-1} \exp(-x^{\alpha})}{\{1 - (1 - \lambda) \exp(-x^{\alpha})\}^{2}}; \quad x > 0, \lambda > 0, \alpha > 0
\]

where \( \alpha \) and \( \lambda \) are the tilt and scale parameters, respectively.

Value

dmoew gives the density, pmoew gives the distribution function, qmoew gives the quantile function, and rmoew generates random deviates.

References


MOEWsurvival

Survival related functions for the Marshall-Olkin Extended Weibull (MOEW) distribution

Description

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the Marshall-Olkin Extended Weibull (MOEW) distribution with tilt parameter alpha and scale parameter lambda.

Usage

```r
crf.moew(x, t = 0, alpha, lambda)
hmoew(x, alpha, lambda)
hra.moew(x, alpha, lambda)
smoew(x, alpha, lambda)
```

Arguments

- `x` vector of quantiles.
- `alpha` tilt parameter.
- `lambda` scale parameter.
- `t` age component.

Value

`crf.moew` gives the conditional reliability function (crf), `hmoew` gives the hazard function, `hra.moew` gives the hazard rate average (HRA) function, and `smoew` gives the survival function for the MOEW distribution.

Examples

```r
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.3035937, lambda.est = 279.2177754
dmoew(sys2, 0.3035937, 279.2177754, log = FALSE)
substr(sys2, 0.3035937, 279.2177754, lower.tail = TRUE, log.p = FALSE)
qmoew(0.25, 0.3035937, 279.2177754, lower.tail = TRUE, log.p = FALSE)
rmoew(50, 0.3035937, 279.2177754)
```
pp.burrX

References


See Also
dmoew for other MOEW distribution related functions;

Examples

```R
## load data set
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.3035937, lambda.est = 279.2177754
## Reliability indicators for data(sys2):

## Reliability function
smoew(sys2, 0.3035937, 279.2177754)

## Hazard function
hmoew(sys2, 0.3035937, 279.2177754)

## hazard rate average(hra)
hra.moew(sys2, 0.3035937, 279.2177754)

## Conditional reliability function (age component=0)
crf.moew(sys2, 0.00, 0.3035937, 279.2177754)

## Conditional reliability function (age component=3.0)
crf.moew(sys2, 3.0, 0.3035937, 279.2177754)
```

Description

The function `pp.burrX()` produces a PP plot for the BurrX based on their MLE or any other estimate. Also, a reference line can be sketched.

Usage

```R
pp.burrX(x, alpha.est, lambda.est, main = "", line = FALSE, ...)
```
Arguments

- `x` vector of observations
- `alpha.est` estimate of the parameter alpha
- `lambda.est` estimate of the parameter lambda
- `main` the title for the plot.
- `line` logical; if TRUE, a 45 degree line is sketched.
- `...` additional arguments to be passed to the underlying plot function.

Value

The function `pp.burrX()` carries out a PP plot for the BurrX.

References


See Also

- `qq.burrX` for QQ plot and `ks.burrX` function

Examples

```r
## Load data sets
data(bearings)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.1989515, lambda.est = 0.0130847

pp.burrX(bearings, 1.1989515, 0.0130847, main = "", line = TRUE)
```

Description

The function `pp.chen()` produces a PP plot for the Chen based on their MLE or any other estimator. Also, a reference line can be sketched.

Usage

```r
pp.chen(x, beta.est, lambda.est, main = "", line = TRUE, ...)
```
Arguments

- `x`: vector of observations
- `beta.est`: estimate of the parameter beta
- `lambda.est`: estimate of the parameter lambda
- `main`: the title for the plot.
- `line`: logical; if TRUE, a 45 degree line is sketched.
- `...`: additional arguments to be passed to the underlying plot function.

Value

The function `pp.chen()` carries out a PP plot for the Chen.

References


See Also

- `qq.chen` for QQ plot and `ks.chen` function;

Examples

```r
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of beta & lambda for the data(sys2)
## beta.est = 0.262282404, lambda.est = 0.007282371

pp.chen(sys2, 0.262282404, 0.007282371, line = TRUE)
```

Description

The function `pp.exp.ext()` produces a PP plot for the Exponential Extension(EE) based on their MLE or any other estimate. Also, a reference line can be sketched.

Usage

```r
pp.exp.ext(x, alpha.est, lambda.est, main = "", line = FALSE, ...)
```
Arguments

- `x`: vector of observations
- `alpha.est`: estimate of the parameter alpha
- `lambda.est`: estimate of the parameter lambda
- `main`: the title for the plot.
- `line`: logical; if TRUE, a 45 degree line is sketched.
- `...`: additional arguments to be passed to the underlying plot function.

Value

The function `pp.exp.ext()` carries out a PP plot for the Exponential Extension (EE).

References


See Also

`qq.exp.ext` for QQ plot and `ks.exp.ext` function;

Examples

```r
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.0126e+01, lambda.est = 1.5848e-04

pp.exp.ext(sys2, 1.0126e+01, 1.5848e-04, main = " ", line = TRUE)
```

Description

The function `pp.exp.power()` produces a PP plot for the Exponential Power distribution based on their MLE or any other estimator. Also, a reference line can be sketched.

Usage

```r
pp.exp.power(x, alpha.est, lambda.est, main = " ", line = FALSE, ...)
```
**Arguments**

- `x`: vector of observations
- `alpha.est`: estimate of the parameter alpha
- `lambda.est`: estimate of the parameter lambda
- `main`: the title for the plot.
- `line`: logical; if TRUE, a 45 degree line is sketched.
- `...`: additional arguments to be passed to the underlying plot function.

**Value**

The function `pp.exp.power()` carries out a PP plot for the Exponential Power distribution.

**References**


**See Also**

`qq.exp.power` for QQ plot and `ks.exp.power` function;

**Examples**

```r
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.905868898, lambda.est = 0.001531423

pp.exp.power(sys2, 0.905868898, 0.001531423, main = '', line = TRUE)
```

**Description**

The function `pp.exp.logistic()` produces a PP plot for the Exponentiated Logistic(EL) distribution on their MLE or any other estimate. Also, a reference line can be sketched.

**Usage**

```r
pp.exp.logistic(x, alpha.est, beta.est, main = "", line = FALSE, ...)
```
Arguments

- `x`: vector of observations
- `alpha.est`: estimate of the parameter alpha
- `beta.est`: estimate of the parameter beta
- `main`: the title for the plot.
- `line`: logical; if TRUE, a 45 degree line is sketched.
- `...`: additional arguments to be passed to the underlying plot function.

Value

The function `pp.expo.logistic()` carries out a PP plot for the Exponentiated Logistic(EL).

References


See Also

- `qq.expo.logistic` for QQ plot
- `ks.expo.logistic` function;

Examples

```r
## Load data sets
data(dataset2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(dataset2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 5.31302, beta.est = 139.04515

pp.expo.logistic(dataset2, 5.31302, 139.04515, main = "", line = TRUE)
```

---

`pp.expo.weibull` *Probability versus Probability (PP) plot for the Exponentiated Weibull(EW) distribution*

Description

The function `pp.expo.weibull()` produces a PP plot for the Exponentiated Weibull(EW) based on their MLE or any other estimate. Also, a reference line can be sketched.

Usage

```r
pp.expo.weibull(x, alpha.est, theta.est, main = "", line = FALSE, ...)
```
pp.flex.weibull

Arguments

- **x** vector of observations
- **alpha.est** estimate of the parameter alpha
- **theta.est** estimate of the parameter theta
- **main** the title for the plot.
- **line** logical; if TRUE, a 45 degree line is sketched.
- **...** additional arguments to be passed to the underlying plot function.

Value

The function `pp.expo.weibull()` carries out a PP plot for the Exponentiated Weibull (EW).

References


See Also

- `qq.expo.weibull` for QQ plot and `ks.expo.weibull` function;

Examples

```r
## Load data sets
data(stress)
## Maximum Likelihood (ML) Estimates of alpha & theta for the data(stress)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est = 1.026465, theta.est = 7.824943

pp.expo.weibull(stress, 1.026465, 7.824943, main = "", line = TRUE)
```

Description

The function `pp.flex.weibull()` produces a PP plot for the flexible Weibull (FW) based on their MLE or any other estimate. Also, a reference line can be sketched.

Usage

```r
pp.flex.weibull(x, alpha.est, beta.est, main = "", line = FALSE, ...)
```
pp.gen.exp

Probability versus Probability (PP) plot for the Generalized Exponential (GE) distribution

Description

The function pp.gen.exp() produces a PP plot for the GE based on their MLE or any other estimator. Also, a reference line can be sketched.

Usage

pp.gen.exp(x, alpha.est, lambda.est, main = " ", line = FALSE, ...)
Arguments

- **x**: vector of observations
- **alpha.est**: estimate of the parameter alpha
- **theta.est**: estimate of the parameter theta
- **main**: the title for the plot.
- **line**: logical; if TRUE, a 45 degree line is sketched.
- **...**: additional arguments to be passed to the underlying plot function.

Value

The function `pp.gen.exp()` carries out a PP plot for the GE.

See Also

- `qq.gen.exp` for QQ plot and `ks.gen.exp` functions;

Examples

```r
## Load dataset
data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 5.28321139, lambda.est = 0.03229609

pp.gen.exp(bearings, 5.28321139, 0.03229609, line = TRUE)
```

Description

The function `pp.gompertz()` produces a PP plot for the Gompertz based on their MLE or any other estimate. Also, a reference line can be sketched.

Usage

```r
pp.gompertz(x, alpha.est, theta.est, main = "", line = FALSE, ...)
```

Arguments

- **x**: vector of observations
- **alpha.est**: estimate of the parameter alpha
- **theta.est**: estimate of the parameter theta
- **main**: the title for the plot.
- **line**: logical; if TRUE, a 45 degree line is sketched.
- **...**: additional arguments to be passed to the underlying plot function.
Value

The function pp.gompertz() carries out a PP plot for the Gompertz.

References


See Also

qq.gompertz for QQ plot and ks.gompertz function;

Examples

## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(sys2)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est = 0.00121307, theta.est = 0.00173329

pp.gompertz(sys2, 0.00121307, 0.00173329, main = " ", line = TRUE)

pp.gp.weibull

Probability versus Probability (PP) plot for the generalized power Weibull(GPW) distribution

Description

The function pp.gp.weibull() produces a PP plot for the generalized power Weibull(GPW) based on their MLE or any other estimate. Also, a reference line can be sketched.

Usage

pp.gp.weibull(x, alpha.est, theta.est, main = " ", line = FALSE, ...)

Arguments

x vector of observations
alpha.est estimate of the parameter alpha
theta.est estimate of the parameter theta
main the title for the plot.
line logical; if TRUE, a 45 degree line is sketched.
... additional arguments to be passed to the underlying plot function.

Value

The function pp.gp.weibull() carries out a PP plot for the generalized power Weibull(GPW).
References


See Also

`qq.gp.weibull` for QQ plot and `ks.gp.weibull` function;

Examples

```r
## Load data sets
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(repairtimes)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est = 1.566093, theta.est = 0.355321

pp.gp.weibull(repairtimes, 1.566093, 0.355321, main = '', line = TRUE)
```

---

**Description**

The function `pp.gumbel()` produces a PP plot for the Gumbel based on their MLE or any other estimate. Also, a reference line can be sketched.

**Usage**

```r
pp.gumbel(x, mu.est, sigma.est, main = '', line = FALSE, ...)
```

**Arguments**

- `x` vector of observations
- `mu.est` estimate of the parameter mu
- `sigma.est` estimate of the parameter sigma
- `main` the title for the plot.
- `line` logical; if TRUE, a 45 degree line is sketched.
- `...` additional arguments to be passed to the underlying plot function.

**Value**

The function `pp.gumbel()` carries out a PP plot for the Gumbel.
References


See Also

`qq.gumbel` for QQ plot and `ks.gumbel` function;

Examples

```r
## Load data sets
data(dataset2)
## Maximum Likelihood(ML) Estimates of mu & sigma for the data(dataset2)
## Estimates of mu & sigma using 'maxLik' package
## mu.est = 212.157, sigma.est = 151.768

pp.gumbel(dataset2, 212.157, 151.768, main = " ", line = TRUE)
```

- **pp.inv.genexp**
  - *Probability versus Probability (PP) plot for the Inverse Generalized Exponential(IGE) distribution*

Description

The function `pp.inv.genexp()` produces a PP plot for the Inverse Generalized Exponential(IGE) based on their MLE or any other estimate. Also, a reference line can be sketched.

Usage

```r
pp.inv.genexp(x, alpha.est, lambda.est, main = " ", line = FALSE, ...)
```

Arguments

- `x` vector of observations
- `alpha.est` estimate of the parameter alpha
- `lambda.est` estimate of the parameter lambda
- `main` the title for the plot.
- `line` logical; if TRUE, a 45 degree line is sketched.
- `...` additional arguments to be passed to the underlying plot function.

Value

The function `pp.inv.genexp()` carries out a PP plot for the Inverse Generalized Exponential(IGE).
pp.lfr

References


See Also

`qq-inv.genexp` for QQ plot and `ks-inv.genexp` function;

Examples

```r
## Load data sets
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(repairtimes)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.097807, lambda.est = 1.206889

pp.inv.genexp(repairtimes, 1.097807, 1.206889, main = "", line = TRUE)
```

pp.lfr

*Probability versus Probability (PP) plot for the linear failure rate(LFR) distribution*

Description

The function `pp.lfr(x)` produces a PP plot for the linear failure rate(LFR) based on their MLE or any other estimate. Also, a reference line can be sketched.

Usage

`pp.lfr(x, alpha.est, beta.est, main = "", line = FALSE, ...)`

Arguments

- `x` vector of observations
- `alpha.est` estimate of the parameter alpha
- `beta.est` estimate of the parameter beta
- `main` the title for the plot.
- `line` logical; if TRUE, a 45 degree line is sketched.
- `...` additional arguments to be passed to the underlying plot function.

Value

The function `pp.lfr()` carries out a PP plot for the linear failure rate(LFR).
References


See Also

`qq.lfr` for QQ plot and `ks.lfr` function;

Examples

```r
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(sys2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 1.7773e-03,  beta.est = 2.77764e-06

pp.lfr(sys2, 1.777673e-03, 2.77764e-06, main = " ", line = TRUE)
```

---

**Description**

The function `pp.log.gamma()` produces a PP plot for the log-gamma(LG) based on their MLE or any other estimate. Also, a reference line can be sketched.

**Usage**

```r
pp.log.gamma(x, alpha.est, lambda.est, main = " ", line = FALSE, ...)
```

**Arguments**

- `x` vector of observations
- `alpha.est` estimate of the parameter alpha
- `lambda.est` estimate of the parameter lambda
- `main` the title for the plot.
- `line` logical; if TRUE, a 45 degree line is sketched.
- `...` additional arguments to be passed to the underlying plot function.
**Value**

The function `pp.log.gamma()` carries out a PP plot for the log-gamma (LG).

**References**


**See Also**

`qq.log.gamma` for QQ plot and `ks.log.gamma` function;

**Examples**

```r
## Load data sets
data(conductors)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(conductors)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 0.0088741, lambda.est = 0.6059935

pp.log.gamma(conductors, 0.0088741, 0.6059935, main = "", line = TRUE)
```

---

**pp.logis.exp**  
Probability versus Probability (PP) plot for the Logistic-Exponential (LE) distribution

**Description**

The function `pp.logis.exp()` produces a PP plot for the Logistic-Exponential (LE) based on their MLE or any other estimate. Also, a reference line can be sketched.

**Usage**

```r
pp.logis.exp(x, alpha.est, lambda.est, main = "", line = FALSE, ...)"n```

**Arguments**

- `x`: vector of observations
- `alpha.est`: estimate of the parameter alpha
- `lambda.est`: estimate of the parameter lambda
- `main`: the title for the plot.
- `line`: logical; if TRUE, a 45 degree line is sketched.
- `...`: additional arguments to be passed to the underlying plot function.
Value

The function `pp.logis.exp()` carries out a PP plot for the Logistic-Exponential(LE).

References


See Also

`qq.logis.exp` for QQ plot and `ks.logis.exp` function;

Examples

```r
## Load data sets
data(bearings)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 2.36754, lambda.est = 0.01059

pp.logis.exp(bearings, 2.36754, 0.01059, main = " ", line = TRUE)
```

---

**Description**

The function `pp.logis.rayleigh()` produces a PP plot for the Logistic-Rayleigh(LR) based on their MLE or any other estimate. Also, a reference line can be sketched.

**Usage**

```r
pp.logis.rayleigh(x, alpha.est, lambda.est, main = " ", line = FALSE, ...)
```

**Arguments**

- `x` vector of observations
- `alpha.est` estimate of the parameter alpha
- `lambda.est` estimate of the parameter lambda
- `main` the title for the plot.
- `line` logical; if TRUE, a 45 degree line is sketched.
- `...` additional arguments to be passed to the underlying plot function.

**Value**

The function `pp.logis.rayleigh()` carries out a PP plot for the Logistic-Rayleigh(LR).
References


See Also

qq.logis.rayleigh for QQ plot and ks.logis.rayleigh function;

Examples

## Load data sets
data(stress)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(stress)
## Estimates of alpha & lambda using 'maxlik' package
## alpha.est = 1.4779388, lambda.est = 0.2141343

pp.logis.rayleigh(stress, 1.4779388, 0.2141343, main = " ", line = TRUE)

Description

The function pp.loglog() produces a PP plot for the Loglog based on their MLE or any other estimator. Also, a reference line can be sketched.

Usage

pp.loglog(x, alpha.est, lambda.est, main = " ", line = FALSE, ...)

Arguments

x vector of observations
alpha.est estimate of the parameter alpha
lambda.est estimate of the parameter lambda
main the title for the plot.
line logical; if TRUE, a 45 degree line is sketched.
... additional arguments to be passed to the underlying plot function.

Value

The function pp.loglog() carries out a PP plot for the Loglog.
References


See Also

qq.loglog for QQ plot and ks.loglog function;

Examples

```r
## Load data sets.
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.9058689 lambda.est = 1.0028228

pp.loglog(sys2, 0.9058689, 1.0028228, line = TRUE)
```

---

**pp.moee**

*Probability versus Probability (PP) plot for the Marshall-Olkin Extended Exponential (MOEE) distribution*

Description

The function `pp.moee()` produces a PP plot for the MOEE based on their MLE or any other estimate. Also, a reference line can be sketched.

Usage

```r
pp.moee(x, alpha.est, lambda.est, main = "", line = FALSE, ...)
```

Arguments

- `x` vector of observations
- `alpha.est` estimate of the parameter alpha
- `lambda.est` estimate of the parameter lambda
- `main` the title for the plot.
- `line` logical; if TRUE, a 45 degree line is sketched.
- `...` additional arguments to be passed to the underlying plot function.

Value

The function `pp.moee()` carries out a PP plot for the MOEE.
References


See Also

`qq.moee` for QQ plot and `ks.moee` functions

Examples

```r
## Load dataset
data(stress)

## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 75.67982, lambda.est = 1.67576

pp.moee(stress, 75.67982, 1.67576, main = "", line = TRUE)
```

Description

The function `pp.moew` produces a PP plot for the MOEW based on their MLE or any other estimate. Also, a reference line can be sketched.

Usage

```r
pp.moew(x, alpha.est, lambda.est, main = "", line = FALSE, ...)
```

Arguments

- `x`: vector of observations
- `alpha.est`: estimate of the parameter alpha
- `lambda.est`: estimate of the parameter lambda
- `main`: the title for the plot.
- `line`: logical; if TRUE, a 45 degree line is sketched.
- `...`: additional arguments to be passed to the underlying plot function.

Value

The function `pp.moew` carries out a PP plot for the MOEW.
References


See Also

`qq.moew` for QQ plot and `ks.moew` function;

Examples

```r
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.3035937, lambda.est = 279.2177754

pp.moew(sys2, 0.3035937, 279.2177754, main = " ", line = TRUE)
```

Description

The function `pp.weibull.ext()` produces a PP plot for the Weibull Extension(WE) based on their MLE or any other estimate. Also, a reference line can be sketched.

Usage

```r
pp.weibull.ext(x, alpha.est, beta.est, main = " ", line = FALSE, ...)
```

Arguments

- `x` vector of observations
- `alpha.est` estimate of the parameter alpha
- `beta.est` estimate of the parameter beta
- `main` the title for the plot.
- `line` logical; if TRUE, a 45 degree line is sketched.
- `...` additional arguments to be passed to the underlying plot function.

Value

The function `pp.weibull.ext()` carries out a PP plot for the Weibull Extension(WE).
References


See Also

qq.weibull.ext for QQ plot and ks.weibull.ext function;

Examples

## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(sys2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 0.00019114, beta.est = 0.14696242

pp.weibull.ext(sys2, 0.00019114, 0.14696242, main = " ", line = TRUE)

qq.burrX

Quantile versus quantile (QQ) plot for the BurrX distribution

Description

The function qq.burrX() produces a QQ plot for the BurrX based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

Usage

qq.burrX(x, alpha.est, lambda.est, main = " ", line.qt = FALSE, ...)

Arguments

x vector of observations
alpha.est estimate of the parameter alpha
lambda.est estimate of the parameter lambda
main the title for the plot
line.qt logical; if TRUE, a line going by the first and third quartile is sketched.
...

Value

The function qq.burrX() carries out a QQ plot for the BurrX.
References


See Also

`pp.burrX` for PP plot and `ks.burrX` function

Examples

```r
## Load data sets
data(bearings)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.1989515, lambda.est = 0.0130847
qq.burr(bearings, 1.1989515, 0.0130847, main = " ", line.qt = FALSE)

qq.chen

Quantile versus quantile (QQ) plot for the Chen distribution

Description

The function `qq.chen()` produces a QQ plot for the Chen based on their MLE or any other estimator. Also, a line going through the first and the third quartile can be sketched.

Usage

`qq.chen(x, beta.est, lambda.est, main = " ", line.qt = FALSE, ...)`

Arguments

- `x`: vector of observations
- `beta.est`: estimate of the parameter beta
- `lambda.est`: estimate of the parameter lambda
- `main`: the title for the plot
- `line.qt`: logical; if TRUE, a line going by the first and third quartile is sketched.
- `...`: additional arguments to be passed to the underlying plot function.

Value

The function `qq.chen()` carries out a QQ plot for the Chen
References


See Also

`pp.chen` for PP plot and `ks.chen` function;

Examples

```r
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of beta & lambda for the data(sys2)
## beta.est = 0.262282404, lambda.est = 0.007282371

qq.chen(sys2, 0.262282404, 0.007282371, line.qt = FALSE)
```

```
qq.exp.ext        Quantile versus quantile (QQ) plot for the Exponential Extension(EE) distribution

Description

The function `qq.exp.ext()` produces a QQ plot for the ExpExt based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

Usage

`qq.exp.ext(x, alpha.est, lambda.est, main = " ", line.qt = FALSE, ...)`

Arguments

- `x` vector of observations
- `alpha.est` estimate of the parameter alpha
- `lambda.est` estimate of the parameter lambda
- `main` the title for the plot
- `line.qt` logical; if TRUE, a line going by the first and third quartile is sketched.
- `...` additional arguments to be passed to the underlying plot function.

Value

The function `qq.exp.ext()` carries out a QQ plot for the Exponential Extension.
References


See Also

pp.exp.ext for PP plot and ks.exp.ext function;

Examples

## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.0126e+01, lambda.est = 1.5848e-04

qq.exp.ext(sys2, 1.0126e+01, 1.5848e-04, main = " ", line.qt = FALSE)

qq.exp.power Quantile versus quantile (QQ) plot for the Exponential Power distribution

Description

The function `qq.exp.power()` produces a QQ plot for the Exponential Power distribution based on their MLE or any other estimator. Also, a line going through the first and the third quartile can be sketched.

Usage

`qq.exp.power(x, alpha.est, lambda.est, main = " ", line.qt = FALSE, ...)`

Arguments

- `x` vector of observations
- `alpha.est` estimate of the parameter alpha
- `lambda.est` estimate of the parameter lambda
- `main` the title for the plot
- `line.qt` logical; if TRUE, a line going by the first and third quartile is sketched.
- `...` additional arguments to be passed to the underlying plot function.

Value

The function `qq.exp.power()` carries out a QQ plot for the Exponential Power distribution.
qq.expo.logistic

References


See Also

pp.exp.power for PP plot and ks.exp.power function;

Examples

## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.905868898, lambda.est = 0.001531423

qq.exp.power(sys2, 0.905868898, 0.001531423, line.qt = FALSE)

qq.expo.logistic Quantile versus quantile (QQ) plot for the Exponentiated Logistic(EL) distribution

Description

The function qq.expo.logistic() produces a QQ plot for the Exponentiated Logistic(EL) based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

Usage

qq.expo.logistic(x, alpha.est, beta.est, main = " ", line.qt = FALSE, ...)

Arguments

x vector of observations
alpha.est estimate of the parameter alpha
beta.est estimate of the parameter beta
main the title for the plot
line.qt logical; if TRUE, a line going by the first and third quartile is sketched.
...
additional arguments to be passed to the underlying plot function.

Value

The function qq.expo.logistic() carries out a QQ plot for the Exponentiated Logistic(EL).
qq.expo.weibull

References


See Also

pp.expo.logistic for PP plot and ks.expo.logistic function;

Examples

## Load data sets
data(dataset2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(dataset2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 5.31302, beta.est = 139.04515

qq.expo.logistic(dataset2, 5.31302, 139.04515, main = "", line.qt = FALSE)

---

qq.expo.weibull Quantile versus quantile (QQ) plot for the Exponentiated Weibull(EW) distribution

Description

The function qq.expo.weibull() produces a QQ plot for the Exponentiated Weibull(EW) based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

Usage

qq.expo.weibull(x, alpha.est, theta.est, main = "", line.qt = FALSE, ...)

Arguments

x vector of observations
alpha.est estimate of the parameter alpha
theta.est estimate of the parameter theta
main the title for the plot
line.qt logical; if TRUE, a line going by the first and third quartile is sketched.
... additional arguments to be passed to the underlying plot function.

Value

The function qq.expo.weibull() carries out a QQ plot for the Exponentiated Weibull(EW).
References


See Also

`pp.expo.weibull` for PP plot and `ks.expo.weibull` function;

Examples

```r
## Load data sets
data(stress)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(stress)
## Estimates of alpha & theta using ’maxLik’ package
## alpha.est =1.026465, theta.est = 7.824943

qq.expo.weibull(stress, 1.026465, 7.824943, main = "", line.qt = FALSE)
```

**qq.flex.weibull** *Quantile versus quantile (QQ) plot for the flexible Weibull(FW) distribution*

Description

The function `qq.flex.weibull()` produces a QQ plot for the flexible Weibull(FW) based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

Usage

`qq.flex.weibull(x, alpha.est, beta.est, main = "", line.qt = FALSE, ...)

Arguments

- `x`: vector of observations
- `alpha.est`: estimate of the parameter alpha
- `beta.est`: estimate of the parameter beta
- `main`: the title for the plot
- `line.qt`: logical; if TRUE, a line going by the first and third quartile is sketched.
- `...`: additional arguments to be passed to the underlying plot function.

Value

The function `qq.flex.weibull()` carries out a QQ plot for the flexible Weibull(FW).
References


See Also

pp.flex.weibull for PP plot and ks.flex.weibull function;

Examples

## Load data sets
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(repairtimes)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 0.07077507, beta.est = 1.13181535

qq.flex.weibull(repairtimes, 0.07077507, 1.13181535, main = " ", line.qt = FALSE)

qq.gen.exp

Quantile versus quantile (QQ) plot for the Generalized Exponential(GE) distribution

Description

The function qq.gen.exp() produces a QQ plot for the GE based on their MLE or any other estimator. Also, a line going through the first and the third quartile can be sketched.

Usage

qq.gen.exp(x, alpha.est, lambda.est, main = " ", line.qt = FALSE, ...)

Arguments

x vector of observations
alpha.est estimate of the parameter alpha
lambda.est estimate of the parameter lambda
main the title for the plot
line.qt logical; if TRUE, a line going by the first and third quartile is sketched.
... additional arguments to be passed to the underlying plot function.

Value

The function qq.gen.exp() carries out a QQ plot for the GE
References


See Also

`pp.gen.exp` for PP plot and `ks.gen.exp` function

Examples

```r
## Load data
data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 5.28321139, lambda.est = 0.03229609

qq.gen.exp(bearings, 5.28321139, 0.03229609, line.qt = FALSE)
```

---

**qq.gompertz**

*Quantile versus quantile (QQ) plot for the Gompertz distribution*

Description

The function `qq.gompertz()` produces a QQ plot for the Gompertz based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

Usage

`qq.gompertz(x, alpha.est, theta.est, main = "", line.qt = FALSE, ...)`

Arguments

- `x` vector of observations
- `alpha.est` estimate of the parameter alpha
- `theta.est` estimate of the parameter theta
- `main` the title for the plot
- `line.qt` logical; if TRUE, a line going by the first and third quartile is sketched.
- `...` additional arguments to be passed to the underlying plot function.

Value

The function `qq.gompertz()` carries out a QQ plot for the Gompertz.
References


See Also

pp.gompertz for PP plot and ks.gompertz function;

Examples

```r
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(sys2)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est = 0.00121307, theta.est = 0.00173329

qq.gompertz(sys2, 0.00121307, 0.00173329, main = "", line.qt = FALSE)
```

---

**qq.gp.weibull**

Quantile versus quantile (QQ) plot for the generalized power Weibull(GPW) distribution

Description

The function `qq.gp.weibull()` produces a QQ plot for the generalized power Weibull(GPW) based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

Usage

```r
qq.gp.weibull(x, alpha.est, theta.est, main = "", line.qt = FALSE, ...)
```

Arguments

- **x**: vector of observations
- **alpha.est**: estimate of the parameter alpha
- **theta.est**: estimate of the parameter theta
- **main**: the title for the plot
- **line.qt**: logical; if TRUE, a line going by the first and third quartile is sketched.
- **...**: additional arguments to be passed to the underlying plot function.

Value

The function `qq.gp.weibull()` carries out a QQ plot for the generalized power Weibull(GPW).
References


See Also

`pp.gp.weibull` for PP plot and `ks.gp.weibull` function;

Examples

```r
## Load data sets
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(repairtimes)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est = 1.566093, theta.est = 0.355321

qq.gp.weibull(repairtimes, 1.566093, 0.355321, main = "", line.qt = FALSE)
```

---

**qq.gumbel**  
*Quantile versus quantile (QQ) plot for the Gumbel distribution*

Description

The function qq.gumbel() produces a QQ plot for the Gumbel based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

Usage

```r
qq.gumbel(x, mu.est, sigma.est, main = "", line.qt = FALSE, ...)
```

Arguments

- **x**: vector of observations  
- **mu.est**: estimate of the parameter mu  
- **sigma.est**: estimate of the parameter sigma  
- **main**: the title for the plot  
- **line.qt**: logical; if TRUE, a line going by the first and third quartile is sketched.  
- **...**: additional arguments to be passed to the underlying plot function.

Value

The function qq.gumbel() carries out a QQ plot for the Gumbel.
**References**


**See Also**

`pp.gumbel` for PP plot and `ks.gumbel` function;

**Examples**

```r
## Load data sets
data(dataset2)
## Maximum Likelihood(ML) Estimates of mu & sigma for the data(dataset2)
## Estimates of mu & sigma using 'maxLik' package
## mu.est = 212.157, sigma.est = 151.768

qq.gumbel(dataset2, 212.157, 151.768, main = " ", line.qt = FALSE)
```

---

### Description

The function `qq.inv.genexp()` produces a QQ plot for the Inverse Generalized Exponential(IGE) based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

### Usage

```r
qq.inv.genexp(x, alpha.est, lambda.est, main = " ", line.qt = FALSE, ...)
```

### Arguments

- `x`: vector of observations
- `alpha.est`: estimate of the parameter alpha
- `lambda.est`: estimate of the parameter lambda
- `main`: the title for the plot
- `line.qt`: logical; if TRUE, a line going by the first and third quartile is sketched.
- `...`: additional arguments to be passed to the underlying plot function.

### Value

The function `qq.inv.genexp()` carries out a QQ plot for the Exponential Extension.
References


See Also

pp.inv.genexp for PP plot and ks.inv.genexp function;

Examples

```r
## Load data sets
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(repairtimes)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.097807, lambda.est = 1.206889

qq.inv.genexp(repairtimes, 1.097807, 1.206889, main = "", line.qt = FALSE)
```

```r

qq.lfr

Quantile versus quantile (QQ) plot for the linear failure rate(LFR) distribution

Description

The function qq.lfr() produces a QQ plot for the linear failure rate(LFR) based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

Usage

qq.lfr(x, alpha.est, beta.est, main = "", line.qt = FALSE, ...)

Arguments

x
vector of observations

alpha.est
estimate of the parameter alpha

beta.est
estimate of the parameter beta

main
the title for the plot

line.qt
logical; if TRUE, a line going by the first and third quartile is sketched.

...
additional arguments to be passed to the underlying plot function.

Value

The function qq.lfr() carries out a QQ plot for the linear failure rate(LFR).
References


See Also

`pp.lfr` for PP plot and `ks.lfr` function;

Examples

```r
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(sys2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 1.77773e-03,  beta.est = 2.77764e-06

qq.lfr(sys2, 1.777673e-03, 2.777640e-06, main = " ", line.qt = FALSE)
```

---

**Description**

The function `qq.log.gamma()` produces a QQ plot for the ExpExt based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

**Usage**

`qq.log.gamma(x, alpha.est, lambda.est, main = " ", line.qt = FALSE, ...)`

**Arguments**

- `x` vector of observations
- `alpha.est` estimate of the parameter alpha
- `lambda.est` estimate of the parameter lambda
- `main` the title for the plot
- `line.qt` logical; if TRUE, a line going by the first and third quartile is sketched.
- `...` additional arguments to be passed to the underlying plot function.
qq.logis.exp

Value

The function qq.log.gamma() carries out a QQ plot for the log-gamma(LG).

References


See Also

pp.log.gamma for PP plot and ks.log.gamma function;

Examples

```r
## Load data sets
data(conductors)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(conductors)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 0.0088741, lambda.est = 0.6059935

qq.log.gamma(conductors, 0.0088741, 0.6059935, main = " ", line.qt = FALSE)
```

```
qq.logis.exp

Quantile versus quantile (QQ) plot for the Logistic-Exponential(LE) distribution
```

Description

The function qq.logis.exp() produces a QQ plot for the ExpExt based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

Usage

```r
qq.logis.exp(x, alpha.est, lambda.est, main = " ", line.qt = FALSE, ...)
```

Arguments

- `x` vector of observations
- `alpha.est` estimate of the parameter alpha
- `lambda.est` estimate of the parameter lambda
- `main` the title for the plot
- `line.qt` logical; if TRUE, a line going by the first and third quartile is sketched.
- `...` additional arguments to be passed to the underlying plot function.
Value

The function `qq.logis.exp()` carries out a QQ plot for the Exponential Extension.

References


See Also

`pp.logis.exp` for PP plot and `ks.logis.exp` function;

Examples

```r
## Load data sets
data(bearings)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 2.36754, lambda.est = 0.01059

qq.logis.exp(bearings, 2.36754, 0.01059, main = " ", line.qt = FALSE)
```

Description

The function `qq.logis.rayleigh()` produces a QQ plot for the ExpExt based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

Usage

```r
qq.logis.rayleigh(x, alpha.est, lambda.est, main = " ", line.qt = FALSE, ...)
```

Arguments

- `x` vector of observations
- `alpha.est` estimate of the parameter alpha
- `lambda.est` estimate of the parameter lambda
- `main` the title for the plot
- `line.qt` logical; if TRUE, a line going by the first and third quartile is sketched.
- `...` additional arguments to be passed to the underlying plot function.

Value

The function `qq.logis.rayleigh()` carries out a QQ plot for the Exponential Extension.
References


See Also

`pp.logis.rayleigh` for PP plot and `ks.logis.rayleigh` function;

Examples

```r
## Load data sets
data(stress)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(stress)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.4779388, lambda.est = 0.2141343

qq.logis.rayleigh(stress, 1.4779388, 0.2141343, main = "", line.qt = FALSE)
```

qq.loglog

Quantile versus quantile (QQ) plot for the Loglog distribution

Description

The function `qq.loglog()` produces a QQ plot for the Loglog based on their MLE or any other estimator. Also, a line going through the first and the third quartile can be sketched.

Usage

```r
qq.loglog(x, alpha.est, lambda.est, main = "", line.qt = FALSE, ...)
```

Arguments

- `x` vector of observations
- `alpha.est` estimate of the parameter alpha
- `lambda.est` estimate of the parameter lambda
- `main` the title for the plot
- `line.qt` logical; if TRUE, a line going by the first and third quartile is sketched.
- `...` additional arguments to be passed to the underlying plot function.

Value

The function `qq.loglog()` carries out a QQ plot for the Loglog
References


See Also

*pp.loglog* for PP plot and *ks.loglog* function;

Examples

```r
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.9058689 lambda.est = 1.0028228

qq.loglog(sys2, 0.9058689, 1.0028228, line.qt = FALSE)
```

---

**qq.moee**

Quantile versus quantile (QQ) plot for the Marshall-Olkin Extended Exponential (MOEE) distribution

Description

The function `qq.moee()` produces a QQ plot for the MOEE based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

Usage

```r
qq.moee(x, alpha.est, lambda.est, main = "", line qt = FALSE, ...)
```

Arguments

- `x`: vector of observations
- `alpha.est`: estimate of the parameter alpha
- `lambda.est`: estimate of the parameter lambda
- `main`: the title for the plot
- `line.qt`: logical; if TRUE, a line going by the first and third quartile is sketched.
- `...`: additional arguments to be passed to the underlying plot function.

Value

The function `qq.moee()` carries out a QQ plot for the MOEE.
References


See Also

`pp.moee` for PP plot and `ks.moee` function

Examples

```r
## Load dataset
data(stress)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 75.67982, lambda.est = 1.67576

qq.moew(stress, 75.67982, 1.67576, main = '', line.qt = FALSE)
```

---

**Description**

The function `qq.moew()` produces a QQ plot for the MOEW based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

**Usage**

`qq.moew(x, alpha.est, lambda.est, main = "", line.qt = FALSE, ...)`

**Arguments**

- `x` vector of observations
- `alpha.est` estimate of the parameter alpha
- `lambda.est` estimate of the parameter lambda
- `main` the title for the plot
- `line.qt` logical; if TRUE, a line going by the first and third quartile is sketched.
- `...` additional arguments to be passed to the underlying plot function.

**Value**

The function `qq.moew()` carries out a QQ plot for the MOEW.
References


See Also

*pp.moew* for PP plot and *ks.moew* function;

Examples

```r
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.3035937, lambda.est = 279.2177754

qq.moew(sys2, 0.3035937, 279.2177754, main = " ", line.qt = FALSE)
```

---

**qq.weibull.ext**

Quantile versus quantile (QQ) plot for the Weibull Extension(WE) distribution

Description

The function `qq.weibull.ext()` produces a QQ plot for the Weibull Extension(WE) based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

Usage

```r
qq.weibull.ext(x, alpha.est, beta.est, main = " ", line.qt = FALSE, ...)
```

Arguments

- `x` vector of observations
- `alpha.est` estimate of the parameter alpha
- `beta.est` estimate of the parameter beta
- `main` the title for the plot
- `line.qt` logical; if TRUE, a line going by the first and third quartile is sketched.
- `...` additional arguments to be passed to the underlying plot function.

Value

The function `qq.weibull.ext()` carries out a QQ plot for the Weibull Extension(WE).
reactorpump

Description

Several data sets related to life test are available in the reliaR package, which have been taken from the literature.

Usage

data(reactorpump)

Format

A vector containing 23 observations.

Details

The data is based on total time on test plot analysis for mechanical components of the RSG-GAS reactor. The data are the time between failures of secondary reactor pumps.

References


Description

Several data sets related to life test are available in the reliaR package, which have been taken from the literature.

Usage

data(repairtimes)

Format

A vector containing 46 observations.

Details

repairtimes correspond to maintenance data on active repair times (in hours) for an airborne communications transceiver.

References


Examples

```r
## Load data sets
data(repairtimes)
## Histogram for repairtimes
hist(repairtimes)
```
**Description**

Several data sets related to life test are available in the reliaR package, which have been taken from the literature.

**Usage**

```r
data(stress)
```

**Format**

A vector containing 100 observations.

**Details**

The data is obtained from Nichols and Padgett (2006) and it represents the breaking stress of carbon fibres (in Gba).

**References**


**Examples**

```r
## Load data sets
data(stress)
## Histogram for stress
hist(stress)
```

---

**sys2**  
Software Reliability Dataset

**Description**

Several data sets related to life test are available in the reliaR package, which have been taken from the literature.

**Usage**

```r
data(sys2)
```
WeibullExt

Format

A vector containing 86 observations.

Details

The data is obtained from DACS Software Reliability Dataset, Lyu (1996). The data represents the time-between-failures (time unit in milliseconds) of a software. The data given here is transformed from time-between-failures to failure times.

References


Examples

```r
## Load data sets
data(sys2)
## Histogram for sys2
hist(sys2)
```

---

The Weibull Extension(WE) distribution

Description

Density, distribution function, quantile function and random generation for the Weibull Extension(WE) distribution with shape parameter alpha and scale parameter beta.

Usage

```r
dweibull.ext(x, alpha, beta, log = FALSE)
pweibull.ext(q, alpha, beta, lower.tail = TRUE, log.p = FALSE)
qweibull.ext(p, alpha, beta, lower.tail = TRUE, log.p = FALSE)
rweibull.ext(n, alpha, beta)
```

Arguments

- `x, q` vector of quantiles.
- `p` vector of probabilities.
- `n` number of observations. If `length(n) > 1`, the length is taken to be the number required.
- `alpha` shape parameter.
- `beta` scale parameter.
- `log, log.p` logical; if TRUE, probabilities p are given as log(p).
- `lower.tail` logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$.
WeibullExt

Details

The Weibull Extension(WE) distribution has density

\[
f(x; \alpha, \beta) = \beta \left( \frac{x}{\alpha} \right)^{\beta - 1} \exp \left( \frac{x}{\alpha} \right)^{\beta} \exp \left\{ -\alpha \left( \exp \left( \frac{x}{\alpha} \right)^{\beta} - 1 \right) \right\}; \quad (\alpha, c\beta) > 0, x > 0
\]

where \( \alpha \) and \( \beta \) are the shape and scale parameters, respectively.

Value

\text{dweibull.ext} \text{ gives the density, pweibull.ext} \text{ gives the distribution function, qweibull.ext} \text{ gives the quantile function, and rweibull.ext} \text{ generates random deviates.}

References


See Also

\texttt{Random.seed} about random number; \texttt{sweibull.ext} for Weibull Extension(WE) survival / hazard etc. functions

Examples

```r
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(sys2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 0.00019114, beta.est = 0.14696242

dweibull.ext(sys2, 0.00019114, 0.14696242, log = FALSE)
pweibull.ext(sys2, 0.00019114, 0.14696242, lower.tail = TRUE, log.p = FALSE)
qweibull.ext(0.25, 0.00019114, 0.14696242, lower.tail=TRUE, log.p = FALSE)
rweibull.ext(30, 0.00019114, 0.14696242)
```
Description

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the Weibull Extension(WE) distribution with shape parameter alpha and scale parameter beta.

Usage

\[
\begin{align*}
\text{crf.weibull.ext}(x, t = 0, \alpha, \beta) \\
\text{hweibull.ext}(x, \alpha, \beta) \\
\text{hra.weibull.ext}(x, \alpha, \beta) \\
\text{sweibull.ext}(x, \alpha, \beta)
\end{align*}
\]

Arguments

- **x**: vector of quantiles.
- **alpha**: shape parameter.
- **beta**: scale parameter.
- **t**: age component.

Value

- crf.weibull.ext gives the conditional reliability function (crf), hweibull.ext gives the hazard function, hra.weibull.ext gives the hazard rate average (HRA) function, and sweibull.ext gives the survival function for the Weibull Extension(WE) distribution.

References


See Also

dweibull.ext for other c distribution related functions;

Examples

```
## load data set
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(sys2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 0.00019114, beta.est = 0.14696242
```
## Reliability indicators for data(sys2):

## Reliability function
sweibull.ext(sys2, 0.00019114, 0.14696242)

## Hazard function
hweibull.ext(sys2, 0.00019114, 0.14696242)

## Hazard rate average(hra)
hra.weibull.ext(sys2, 0.00019114, 0.14696242)

## Conditional reliability function (age component=0)
crf.weibull.ext(sys2, 0.00, 0.00019114, 0.14696242)

## Conditional reliability function (age component=3.0)
crf.weibull.ext(sys2, 3.0, 0.00019114, 0.14696242)
Index

+Topic datasets
  bearings, 24
  conductors, 31
  dataset2, 32
  reactorpump, 141
  repairtimes, 142
  stress, 143
  sys2, 143
+Topic distribution
  BurrX, 25
  Chen, 28
  ExpoExt, 34
  ExpoLogistic, 37
  ExpoWeibull, 39
  ExpPower, 42
  FlexWeibull, 43
  GenExp, 46
  Gompertz, 49
  GPWeibull, 51
  Gumbel, 54
  InvGenExp, 57
  LFR, 83
  Loggamma, 86
  LogisExp, 88
  LogisRayleigh, 91
  Loglog, 93
  MOEE, 96
  MOEW, 99
  WeibullExt, 144
+Topic hplot
  pp.burrX, 101
  pp.chen, 102
  pp.exp.ext, 103
  pp.exp.power, 104
  pp.expo.logistic, 105
  pp.expo.weibull, 106
  pp.flex.weibull, 107
  pp.gen.exp, 108
  pp.gompertz, 109
  pp.gp.weibull, 110
  pp.gumbel, 111
  pp.inv.genexp, 112
  pp.lfr, 113
  pp.log.gamma, 114
  pp.logis.exp, 115
  pp.logis.rayleigh, 116
  pp.loglog, 117
  pp.moee, 118
  pp.moew, 119
  pp.weibull.ext, 120
  qq.burrX, 121
  qq.chen, 122
  qq.exp.ext, 123
  qq.exp.power, 124
  qq.expo.logistic, 125
  qq.expo.weibull, 126
  qq.flex.weibull, 127
  qq.gen.exp, 128
  qq.gompertz, 129
  qq.gp.weibull, 130
  qq.gumbel, 131
  qq.inv.genexp, 132
  qq.lfr, 133
  qq.log.gamma, 134
  qq.logis.exp, 135
  qq.logis.rayleigh, 136
  qq.loglog, 137
  qq.moee, 138
  qq.moew, 139
  qq.weibull.ext, 140
+Topic htest
  ks.burrX, 59
  ks.chen, 61
  ks.exp.ext, 62
  ks.exp.power, 63
  ks.expo.logistic, 64
  ks.expo.weibull, 65
  ks.flex.weibull, 67
INDEX

ks.gen.exp, 68
ks.gompertz, 69
ks.gp.weibull, 70
ks.gumbel, 71
ks.inv.genexp, 72
ks.lfr, 74
ks.log.gamma, 75
ks.logis.exp, 76
ks.logis.rayleigh, 77
ks.loglog, 78
ks.moee, 79
ks.moew, 81
ks.weibull.ext, 82

*Topic models
abic.burrx, 4
abic.chen, 5
abic.exp.ext, 6
abic.exp.power, 7
abic.expo.logistic, 8
abic.expo.weibull, 9
abic.flex.weibull, 10
abic.gen.exp, 11
abic.gompertz, 12
abic.gp.weibull, 13
abic.gumbel, 14
abic.inv.genexp, 15
abic.lfr, 16
abic.log.gamma, 17
abic.logis.exp, 18
abic.logis.rayleigh, 19
abic.loglog, 20
abic.moee, 21
abic.moew, 22
abic.weibull.ext, 23

*Topic survival
BurrXsurvival, 26
Chensurvival, 30
EPsurvival, 33
ExpExtsurvival, 35
ExpoLogisticsurvival, 38
ExpoWeibullSurvival, 41
FlexWeibullSurvival, 45
GenExpSurvival, 47
GompertzSurvival, 50
GPWeibullSurvival, 53
GumbelSurvival, 55
InvGenExpESurvival, 58
LFRSurvival, 84
LoggammaSurvival, 87
LogisExpsurvival, 90
LogisRayleighSurvival, 92
LoglogSurvival, 95
MOEEsurvival, 97
MOEWsurvival, 100
WeibullExtsurvival, 146
.Random.seed, 26, 29, 35, 38, 40, 43, 44, 47.
  50, 52, 55, 58, 84, 87, 89, 92, 94, 97, 100, 145
abic.burrX, 4
abic.chen, 5
abic.exp.ext, 6
abic.exp.power, 7
abic.expo.logistic, 8
abic.expo.weibull, 9
abic.flex.weibull, 10
abic.gen.exp, 11
abic.gompertz, 12
abic.gp.weibull, 13
abic.gumbel, 14
abic.inv.genexp, 15
abic.lfr, 16
abic.log.gamma, 17
abic.logis.exp, 18
abic.logis.rayleigh, 19
abic.loglog, 20
abic.moee, 21
abic.moew, 22
abic.weibull.ext, 23
bearings, 24
BurrX, 25
BurrXSurvival, 26
Chen, 28
Chensurvival, 30
conductors, 31
crf.burrX (BurrXsurvival), 26
crf.chen (Chensurvival), 30
crf.exp.ext (ExpExtsurvival), 35
crf.exp.power (EPsurvival), 33
crf.expo.logistic
  (ExpoLogisticsurvival), 38
crf.expo.weibull (ExpoWeibullSurvival), 41
crf.flex.weibull (FlexWeibullSurvival), 45
crf.gen.exp (GenExpsurvival), 47
crf.gompertz (Gompertzsurvival), 50
crf.gp.weibull (GPWeibullsurvival), 53
crf.gumbel (Gumbelsurvival), 55
crf.inv.genexp (InvGenExpSurvival), 58
crf.lfr (LFRsurvival), 84
crf.log.gamma (LoggammaSurvival), 87
crf.logis.exp (LogisExpSurvival), 90
crf.logis.rayleigh (LogisRayleighSurvival), 92
crf.loglog (LoglogSurvival), 95
dexp.ext (ExpextSurvival), 35
dexp.power (ExpPower), 33
dexp.power (ExpPowerSurvival), 37
dexpo.logistic (ExpoLogisticSurvival), 38
dexpo.logistic (ExpoLogisticSurvival), 37
dexpo.weibull (ExpoweibullSurvival), 41
dexpo.weibull (ExpoweibullSurvival), 40
dflex.weibull (FlexWeibullSurvival), 45
dflex.weibull (FlexWeibullSurvival), 44
dgen.exp (GenExpSurvival), 48
dgen.exp (GenExpSurvival), 47
dgompertz (GompertzSurvival), 51
dgompertz (GompertzSurvival), 49
dgp.weibull (GPWeibullSurvival), 53
dgumbel (GumbelSurvival), 56
dgumbel (GumbelSurvival), 54
dinv.genexp (InvGenExpSurvival), 57
dinv.gamma, 85
dinv.gamma (InvGammaSurvival), 83
dlog.gamma (LogGammaSurvival), 86
dlogis.exp (LogisExpSurvival), 90
dlogis.exp (LogisExpSurvival), 88
dlogis.rayleigh (LogisRayleighSurvival), 91
dlogis.rayleigh (LogisRayleighSurvival), 90
dloglog (LoglogSurvival), 95
INDEX

hmoee (MOEEsurvival), 97
hmoew (MOEWe survival), 100
hra.burrX (BurrXsurvival), 26
hra.chen (Chensurvival), 30
hra.exp.ext (ExpExtsurvival), 35
hra.exp.power (EPsurvival), 33
hra.expologic (ExpoLogisticsurvival), 38
hra.expow.eiweibull (ExpoWeibullsurvival), 41
hra.fle.xe.weibull (FlexWeibullsurvival), 45
hra.gen.exp (GenExpsurvival), 47
hra.gompertz (Gompertzsurvival), 50
hra.gp.weibull (GPWeibullsurvival), 53
hra.gumbel (Gumbelurvival), 55
hra.inv.genexp (InvGenExpurvival), 58
hra.lfr (LFRsurvival), 84
hra.log.gamma (Loggamma, 86)
hra.logis.exp (LogisExpurvival), 90
hra.logis.rayleigh (LogisRayleighurvival), 92
hra.logis.rayleigh (LogisRayleighurvival), 92
hra.loglog (Loglogsurvival), 95
hra.moee (MOEEsurvival), 97
hra.moew (MOEWe survival), 100
hra.weibull.ext (WeibullExtsurvival), 146
hra.weibull.ext (WeibullExtsurvival), 146
InvGenExp, 57
InvGenExpurvival, 58
InvGenExpurvival (InvGenExpurvival), 58
ks.burrX, 59, 102, 122
ks.chen, 61, 103, 123
ks.exp.ext, 62, 104, 124
ks.exp.power, 63, 105, 125
ks.expo.logistic, 64, 106, 126
ks.expo.weibull, 65, 107, 127
ks.flex.weibull, 67, 108, 128
ks.gen.exp, 68, 109, 129
ks.gompertz, 69, 110, 130
ks.gp.weibull, 70, 111, 131
ks.gumbel, 71, 112, 132
ks.inv.genexp, 72, 113, 133
ks.lfr, 74, 114, 134
ks.log.gamma, 75, 115, 135
ks.logis.exp, 76, 116, 136
ks.logis.rayleigh, 77, 117, 137
ks.loglog, 21, 78, 118, 138
ks.moee, 79, 119, 139
ks.moew, 81, 120, 140
ks.weibull.ext, 82, 121, 141
LFR, 83
LFRsurvival, 84
Loggamma, 86
Loggamma, 86
LogisExp, 88
LogisExpurvival, 90
LogisRayleigh, 91
LogisRayleighurvival, 92
Loglog, 93
Loglogsurvival, 95
MOEE, 96
MOEEsurvival, 97
MOEWe survival, 100
pburrX (BurrX), 25
pchen (Chen), 28
pexp.ext (ExpExt), 34
pexp.power (ExpPower), 42
pexpo.logistic (ExpoLogistic), 37
pexpo.weibull (ExpoWeibull), 39
pflex.weibull (FlexWeibull), 43
pgen.exp (GenExp), 46
pgompertz (Gompertz), 49
pgp.weibull (GPWeibull), 51
pgumbel (Gumbel), 54
pinv.genexp (InvGenExp), 57
plfr (LFR), 83
plog.gamma (Loggamma), 86
plogis.exp (LogisExp), 88
plogis.rayleigh (LogisRayleigh), 91
ploglog (Loglog), 93
pmoee (MOEE), 96
pmoew (MOEWe), 99
pp.burrX, 5, 60, 101, 122
pp.chen, 6, 61, 102, 123
pp.exp.ext, 7, 63, 103, 124
pp.exp.power, 8, 64, 104, 125
pp.expo.logistic, 9, 65, 105, 126
pp.expo.logistic, 9, 65, 105, 126
pp.exp.weibull, 10, 66, 106, 127
pp.flex.weibull, 11, 67, 107, 128
pp.gen.exp, 12, 69, 108, 129
pp.gompertz, 13, 70, 109, 130
pp.gp.weibull, 14, 71, 110, 131
pp.gumbel, 15, 72, 111, 132
pp.inv.genexp, 16, 73, 112, 133
pp.lfr, 17, 74, 113, 134
pp.log.gamma, 18, 76, 114, 135
pp.logis.exp, 19, 77, 115, 136
pp.logis.rayleigh, 20, 78, 116, 137
pp.loglog, 79, 117, 138
pp.moee, 22, 80, 118, 139
pp.moew, 23, 81, 119, 140
pp.weibull.ext, 24, 83, 120, 141
qweibull.ext (WeibullExt), 144
qburrX (BurrX), 25
qchen (Chen), 28
qexp.ext (ExExt), 34
qexp.power (ExpPower), 42
qexpo.logistic (ExpoLogistic), 37
qexpo.weibull (ExpoWeibull), 39
qflflexweibull (FlexWeibull), 43
qgen.exp (GenExp), 46
qgompertz (Gompertz), 49
qp.weibull (GPWeibull), 51
qgumbel (Gumbel), 54
qinv.genexp (InvGenExp), 57
qlfr (LFR), 83
qllog.gamma (Loggamma), 86
qlogis.exp (LogExp), 88
qlogis.rayleigh (LogisRayleigh), 91
qloglog (Loglog), 93
qmoee (MOEE), 96
qmoew (MOEW), 99
qq.burrx, 5, 60, 102, 121
qq.chen, 6, 61, 103, 122
qq.exp, 7, 63, 104, 123
qq.exp.power, 8, 64, 105, 124
qqexpo.logistic, 9, 65, 106, 125
qq.expo.weibull, 10, 66, 107, 126
qq.flex.weibull, 11, 67, 108, 127
qq.gen.exp, 12, 69, 109, 128
qq.gompertz, 13, 70, 110, 129
qq.gp.weibull, 14, 71, 111, 130
qq.gumbel, 15, 72, 112, 131
qq.inv.genexp, 16, 73, 113, 132
qq.lfr, 17, 74, 114, 133
qq.log.gamma, 18, 76, 115, 134
qq.logis.exp, 19, 77, 116, 135
qq.logis.rayleigh, 20, 78, 117, 136
qq.loglog, 21, 79, 118, 137
qq.moee, 22, 80, 119, 138
qq.moew, 23, 81, 120, 139
qq.weibull.ext, 24, 83, 121, 140
qweibull.ext (WeibullExt), 144
rburrX (BurrX), 25
rchen (Chen), 28
reactorpump, 141
repairtimes, 142
r.exp.ext (ExpExt), 34
r.exp.power (ExpPower), 42
rexpo.logistic (ExpoLogistic), 37
rexpo.weibull (ExpoWeibull), 39
rflex.weibull (FlexWeibull), 43
rgen.exp (GenExp), 46
rgompertz (Gompertz), 49
rgp.weibull (GPWeibull), 51
rgumbel (Gumbel), 54
rinv.genexp (InvGenExp), 57
rlfr (LFR), 83
rlog.gamma (Loggamma), 86
rlogis.exp (LogisExp), 88
rlogis.rayleigh (LogisRayleigh), 91
rloglog (Loglog), 93
rmoe (MOEE), 96
rmoew (MOEW), 99
rweibull.ext (WeibullExt), 144
sburrX, 26
sburrX (BurrXsurvival), 26
schen, 29
schen (Chensurvival), 30
sexp.ext, 35
sexp.ext (ExExtSurvival), 35
sexp.power, 43
sexp.power (Epsurvival), 33
sexpo.logistic, 38
sexpo.logistic (ExpoLogisticsurvival), 38
sexpo.weibull, 40
sexpo.weibull (ExpoWeibullsurvival), 41
sfexpo.weibull, 44
sfexpo.weibull (FlexWeibullsurvival), 45
sgen.exp, 47
sgen.exp (GenExpsurvival), 47
sgompertz, 50
sgompertz (Gompertzsurvival), 50
sgp.weibull, 52
sgp.weibull (GPWeibullsurvival), 53
sgumbel, 55
sgumbel (Gumbelsurvival), 55
sinv.genexp, 58
sinv.genexp (InvGenExpEsurvival), 58
slfr, 84
slfr (LFRsurvival), 84
slog.gamma, 87
slog.gamma (Loggammasurvival), 87
slogis.exp, 89
slogis.exp (LogisExpsurvival), 90
slogis.rayleigh, 92
slogis.rayleigh (LogisRayleighsurvival), 92
sloglog, 94
sloglog (Loglogsurvival), 95
smoee, 97
smoee (MOEEsurvival), 97
smoew, 100
smoew (MOEWsurvival), 100
stress, 143
sweibull.ext, 145
sweibull.ext (WeibullExtsurvival), 146
sys2, 143

WeibullExt, 144
WeibullExtsurvival, 146