

CASCILIB  
(CAstagliola's SCIlab LIBrary)

Philippe CASTAGLIOLA

July 25, 2012

# Contents

<b>1</b>	<b>CONFIDENCE INTERVALS</b>	<b>6</b>
1.1	intbinomial – binomial confidence interval . . . . .	6
1.2	intexponential – exponential confidence interval . . . . .	6
1.3	intnormalm – normal confidence interval for $\mu$ . . . . .	7
1.4	intnormals – normal confidence interval for $\sigma$ . . . . .	7
1.5	intpoisson – Poisson confidence interval . . . . .	7
<b>2</b>	<b>CUMULATIVE DISTRIBUTION FUNCTIONS</b>	<b>8</b>
2.1	cdfbeta – beta type 1 cdf . . . . .	8
2.2	cdfbeta2 – beta type 2 cdf . . . . .	8
2.3	cdfbinomial – binomial cdf . . . . .	9
2.4	cdfchi2 – $\chi^2$ (central and non-central) cdf . . . . .	9
2.5	cdfcv – sample coefficient of variation cdf . . . . .	9
2.6	cdfdphase – discrete Phase-Type cdf . . . . .	10
2.7	cdfexponential – exponential cdf . . . . .	10
2.8	cdffisher – Fisher (central and non-central) cdf . . . . .	11
2.9	cdffoldednormal – folded normal cdf . . . . .	11
2.10	cdfgamma – gamma cdf . . . . .	11
2.11	cdfgev – generalized Extreme Value cdf . . . . .	12
2.12	cdfhypergeometric – hypergeometric cdf . . . . .	12
2.13	cdfjohnson – Johnson’s cdf . . . . .	12
2.14	cdflognormal – lognormal cdf . . . . .	13
2.15	cdfmedian – normal sample median cdf . . . . .	13
2.16	cdfnormal – normal cdf . . . . .	14
2.17	cdfpareto – Pareto cdf . . . . .	14
2.18	cdfpascal – Pascal cdf . . . . .	14
2.19	cdfpoisson – Poisson cdf . . . . .	15
2.20	cdfrange – normal range cdf . . . . .	15
2.21	cdfstandev – normal sample standard-deviation cdf . . . . .	16
2.22	cdfstudent – Student (central and non-central) cdf . . . . .	16
2.23	cdfweibull – Weibull cdf . . . . .	16
<b>3</b>	<b>DESIGN OF EXPERIMENTS</b>	<b>17</b>
3.1	boxbehnken – Box-Behnken designs . . . . .	17
3.2	boxcoxlinear – Box-Cox linearity transformation . . . . .	17
3.3	centralcomposite – central composite designs . . . . .	18
3.4	coded2natural – coded to natural variables . . . . .	18
3.5	doxpan – design expansion . . . . .	19
3.6	doxptim – design optimisation . . . . .	19
3.7	equiradial – equiradial designs . . . . .	19
3.8	factorial2 – two levels full and fractional factorial designs . . . . .	20
3.9	mulreg – multilinear regression analysis . . . . .	20
3.10	mulregdisp – multilinear regression analysis results display . . . . .	22
3.11	mulregplot – multilinear regression analysis results plot . . . . .	22
3.12	natural2coded – natural to coded variables . . . . .	22
3.13	plackettburman – Plackett-Burman designs . . . . .	23
3.14	simpdex – simplex designs . . . . .	23
<b>4</b>	<b>ESTIMATION</b>	<b>23</b>
4.1	fitbeta – beta type 1 parameters estimation . . . . .	23
4.2	fitgamma – gamma parameters estimation . . . . .	24
4.3	fitgev – generalized Extreme Value parameters estimation . . . . .	24
4.4	fitjohnson – Johnson parameters estimation . . . . .	24
4.5	fitlognormal – lognormal parameters estimation . . . . .	25
4.6	fitweibull – Weibull parameters estimation . . . . .	25

<b>5</b>	<b>INVERSE CUMULATIVE DISTRIBUTION FUNCTIONS</b>	<b>26</b>
5.1	idfbeta – beta type 1 idf . . . . .	26
5.2	idfbeta2 – beta type 2 idf . . . . .	26
5.3	idfchi2 – $\chi^2$ (central and non-central) idf . . . . .	27
5.4	idfcv – sample coefficient of variation idf . . . . .	27
5.5	idfexponential – exponential idf . . . . .	27
5.6	idffisher – Fisher (central and non-central) idf . . . . .	28
5.7	idfgamma – gamma idf . . . . .	28
5.8	idfgev – generalized Extreme Value idf . . . . .	29
5.9	idfjohnson – Johnson’s idf . . . . .	29
5.10	idflognormal – lognormal idf . . . . .	30
5.11	idfmedian – normal sample median idf . . . . .	30
5.12	idfnormal – normal idf . . . . .	30
5.13	idfpareto – Pareto idf . . . . .	31
5.14	idfstandev – normal sample standard-deviation idf . . . . .	31
5.15	idfstudent – Student (central and non-central) idf . . . . .	32
5.16	idfweibull – Weibull idf . . . . .	32
<b>6</b>	<b>MISCELANEOUS</b>	<b>33</b>
6.1	allcombination – matrix element combinations . . . . .	33
6.2	allpermutation – matrix element permutations . . . . .	33
6.3	arrangement – number $A_n^p$ of arrangements . . . . .	33
6.4	combination – number $C_n^p$ of combinations . . . . .	34
6.5	confhyper – confluent hypergeometric function . . . . .	34
6.6	depth – non parametric multivariate depth . . . . .	34
6.7	hausdorff – Hausdorff (median) distance between polylines . . . . .	35
6.8	lowess – LOcally WEighted Scatterplot Smoothing . . . . .	36
6.9	momdphase – first moments of a Discrete Phase-Type distribution . . . . .	36
6.10	nearestneighbors – find the $k$ nearest neighbors . . . . .	38
6.11	neldermead – Nelder Mead’s downhill simplex nonlinear optimization algorithm . . . . .	39
6.12	savitzkygolay – Savitzky-Golay smoothing filter . . . . .	40
6.13	simplex – simplex computation . . . . .	40
6.14	simplexolve – solve a system of non-linear equations . . . . .	41
6.15	torczon – Torczon’s multidirectional nonlinear optimization algorithm . . . . .	42
6.16	vandercorput – Van der Corput’s sequence . . . . .	43
<b>7</b>	<b>PLOTS</b>	<b>43</b>
7.1	boxplot – Box plot . . . . .	43
7.2	qplot – quantile plot . . . . .	44
7.3	qqplot – quantile-quantile plot . . . . .	45
<b>8</b>	<b>PROBABILITY DENSITY FUNCTIONS</b>	<b>45</b>
8.1	pdfbeta – beta type 1 pdf . . . . .	45
8.2	pdfbeta2 – beta type 2 pdf . . . . .	47
8.3	pdfbinomial – binomial pdf . . . . .	48
8.4	pdfchi2 – $\chi^2$ (central and non-central) pdf . . . . .	49
8.5	pdfcp – $C_P$ pdf . . . . .	50
8.6	pdfcpk – $C_{PK}$ pdf . . . . .	51
8.7	pdfcpm – $C_{PM}$ pdf . . . . .	52
8.8	pdfcpmk – $C_{PMK}$ pdf . . . . .	53
8.9	pdfcpuv – Vännman’s $C_p(u, v)$ pdf . . . . .	54
8.10	pdfcv – sample coefficient of variation pdf . . . . .	54
8.11	pdfdphase – discrete phase-type pdf . . . . .	55
8.12	pdfexponential – exponential pdf . . . . .	56
8.13	pdffisher – Fisher (central and non-central) pdf . . . . .	57
8.14	pdffoldednormal – folded normal pdf . . . . .	58
8.15	pdfgamma – gamma pdf . . . . .	58
8.16	pdfgev – generalized Extreme Value pdf . . . . .	59
8.17	pdfhypergeometric – hypergeometric pdf . . . . .	60

8.18	<code>pdfkernel</code> – kernel smoothed pdf . . . . .	61
8.19	<code>pdfjohnson</code> – Johnson’s pdf . . . . .	62
8.20	<code>pdflognormal</code> – lognormal pdf . . . . .	62
8.21	<code>pdfmedian</code> – normal sample median pdf . . . . .	65
8.22	<code>pdfmultinormal</code> – multinormal pdf . . . . .	65
8.23	<code>pdfnormal</code> – normal pdf . . . . .	66
8.24	<code>pdfpareto</code> – Pareto pdf . . . . .	67
8.25	<code>pdfpascal</code> – Pascal pdf . . . . .	68
8.26	<code>pdfpoisson</code> – Poisson pdf . . . . .	68
8.27	<code>pdfrnge</code> – normal range pdf . . . . .	69
8.28	<code>pdfstandev</code> – normal sample standard-deviation pdf . . . . .	70
8.29	<code>pdfstudent</code> – Student pdf . . . . .	71
8.30	<code>pdfweibull</code> – Weibull pdf . . . . .	72
<b>9</b>	<b>QUADRATURES</b>	<b>72</b>
9.1	<code>quadhermite</code> – Gauss-Hermite quadrature . . . . .	72
9.2	<code>quadlaguerre</code> – Gauss-Laguerre quadrature . . . . .	73
9.3	<code>quadlegendre</code> – Gauss-Legendre quadrature . . . . .	73
9.4	<code>quadsimpson</code> – Simpson quadrature . . . . .	74
<b>10</b>	<b>RANDOM NUMBER GENERATORS</b>	<b>74</b>
10.1	<code>rndbeta</code> – beta type 1 random number generator . . . . .	74
10.2	<code>rndbeta2</code> – beta type 2 random number generator . . . . .	75
10.3	<code>rndbinomial</code> – binomial random number generator . . . . .	76
10.4	<code>rndexponential</code> – exponential random number generator . . . . .	77
10.5	<code>rndfoldednormal</code> – folded normal random number generator . . . . .	78
10.6	<code>rndgamma</code> – gamma random number generator . . . . .	79
10.7	<code>rndgev</code> – generalized Extreme Value random number generator . . . . .	80
10.8	<code>rndjohnson</code> – Johnson’s random number generator . . . . .	81
10.9	<code>rndlognormal</code> – lognormal random number generator . . . . .	82
10.10	<code>rndmultinormal</code> – multinormal random number generator . . . . .	82
10.11	<code>rndnormal</code> – normal random number generator . . . . .	84
10.12	<code>rndpareto</code> – Pareto random number generator . . . . .	84
10.13	<code>rndpascal</code> – Pascal random number generator . . . . .	85
10.14	<code>rndpoisson</code> – Poisson random number generator . . . . .	86
10.15	<code>rndstandev</code> – normal sample standard-deviation random number generator . . . . .	87
10.16	<code>rndweibull</code> – Weibull random number generator . . . . .	88
<b>11</b>	<b>SAMPLE STATISTICS</b>	<b>89</b>
11.1	<code>autocorrelation</code> – autocorrelation coefficient . . . . .	89
11.2	<code>bootstrap</code> – bootstrap sampling . . . . .	89
11.3	<code>correlation</code> – correlation matrix . . . . .	90
11.4	<code>crosscorrelation</code> – crosscorrelation coefficient . . . . .	90
11.5	<code>kurtosis</code> – kurtosis coefficient . . . . .	90
11.6	<code>quantile</code> – quantile . . . . .	91
11.7	<code>rng</code> – range . . . . .	91
11.8	<code>skewness</code> – skewness coefficient . . . . .	92
11.9	<code>standev</code> – standard deviation . . . . .	92
11.10	<code>totalmedian</code> – total median coefficients . . . . .	92
11.11	<code>varcovar</code> – variance-covariance matrix . . . . .	93
<b>12</b>	<b>STATISTICAL PROCESS CONTROL</b>	<b>93</b>
12.1	<code>arlmean</code> – <i>ARL</i> of the mean control chart . . . . .	93
12.2	<code>arlmeanRR</code> – <i>ARL</i> of the Run Rules mean control chart . . . . .	94
12.3	<code>arlmedian</code> – <i>ARL</i> of the median control chart . . . . .	94
12.4	<code>arlmedianRR</code> – <i>ARL</i> of the Run Rules median control chart . . . . .	94
12.5	<code>arlrnge</code> – <i>ARL</i> of the range control chart . . . . .	95
12.6	<code>arlstandev</code> – <i>ARL</i> of the standard-deviation control chart . . . . .	95
12.7	<code>arlstandevRR</code> – <i>ARL</i> of the Run Rules standard-deviation control chart . . . . .	96
12.8	<code>cp</code> – capability index $C_P$ estimation and confidence interval . . . . .	96

12.9	cpk – capability index $C_{PK}$ estimation and confidence interval . . . . .	97
12.10	krnge – range coefficients $K_R(n)$ . . . . .	97
12.11	kstandev – standard-deviation coefficients $K_S(n, r)$ . . . . .	98
12.12	mcpshahriari – Shahriari’s multivariate capability index $C_P$ . . . . .	98
12.13	mcptaam – Taam’s multivariate capability index $C_P$ . . . . .	98
<b>13</b>	<b>TESTS</b>	<b>99</b>
13.1	andersondarling – Anderson-Darling’s normality test . . . . .	99
13.2	ansaribradley – Ansari-Bradley’s test . . . . .	99
13.3	bartlett – Bartlett’s test . . . . .	100
13.4	grubbs – Grubbs test . . . . .	100
13.5	kendall – Kendall’s test . . . . .	100
13.6	levene – Levene’s test . . . . .	101
13.7	mardia – Mardia’s test . . . . .	101
13.8	spearman – Spearman’s test . . . . .	101
13.9	tstbinomial1 – binomial one sample $p$ test . . . . .	102
13.10	tstbinomial2 – binomial two samples $p$ test . . . . .	102
13.11	tstexponential – exponential $\lambda$ test . . . . .	103
13.12	tstnormalm1 – normal one sample $\mu$ test . . . . .	103
13.13	tstnormalm2 – normal two samples $\mu$ test . . . . .	104
13.14	tstnormals1 – normal one sample $\sigma$ test . . . . .	104
13.15	tstnormals2 – normal two samples $\sigma$ test . . . . .	105
13.16	tstsku – normal skewness and kurtosis test . . . . .	105
13.17	waldwolfowitz – Wald-Wolfowitz’s run test . . . . .	105
13.18	wilcoxon1 – Wilcoxon’s one sample (paired) test . . . . .	106
13.19	wilcoxon2 – Wilcoxon’s two samples test . . . . .	106

# 1 CONFIDENCE INTERVALS

## 1.1 intbinomial – binomial confidence interval

### Calling Sequence

```
[inter,p]=intbinomial(x,n,side=,level=)
```

### Parameters

- **x** : number  $x$  of successes observed. An integer in  $\{0, \dots, n\}$ .
- **n** : number  $n$  of binomial trials. An integer  $\geq 1$ .
- **side** : side of the confidence interval. Must be "lower", "upper" or "both". Default is "both".
- **level** : confidence level  $1 - \alpha \in (0.5, 1)$ . Default is 0.95.
- **inter** : binomial confidence interval :
  - if **side**="lower", **inter** is the *lower* confidence bound.
  - if **side**="upper", **inter** is the *upper* confidence bound.
  - if **side**="both", **inter** is the bilateral confidence interval.
- **p** : estimation of the probability of success.

### Description

Compute the confidence interval (lower, upper or both) for the probability of success  $p$  based on  $n$  binomial trials. `[inter,p]=intbinomial(x,n)` is equivalent to `[inter,p]=intbinomial(x,n,"both",0.95)`.

### Examples

```
[inter,p]=intbinomial(75,100)
[inter,p]=intbinomial(75,100,"upper")
[inter,p]=intbinomial(75,100,level=0.99)
```

### See Also

tstbinomial1, tstbinomial2

## 1.2 intexponential – exponential confidence interval

### Calling Sequence

```
[inter,lam]=intexponential(X,side=,level=)
```

### Parameters

- **X** : real matrix **X** containing exponential data.
- **side** : side of the confidence interval. Must be "lower", "upper" or "both". Default is "both".
- **level** : confidence level  $1 - \alpha \in (0.5, 1)$ . Default is 0.95.
- **inter** : exponential confidence interval :
  - if **side**="lower", **inter** is the *lower* confidence bound.
  - if **side**="upper", **inter** is the *upper* confidence bound.
  - if **side**="both", **inter** is the bilateral confidence interval.
- **lam** : estimation of parameter  $\lambda$  of the exponential distribution.

### Description

Compute the confidence interval (lower, upper or both) for the parameter  $\lambda$  of the exponential distribution. `[inter,lam]=intexponential(X)` is equivalent to `[inter,lam]=intexponential(X,"both",0.95)`.

### Examples

```
X=rndexponential(100,3);
[inter,lam]=intexponential(X)
[inter,lam]=intexponential(X,"upper")
[inter,lam]=intexponential(X,level=0.99)
```

### See Also

tstexponential

### 1.3 intnormalm – normal confidence interval for $\mu$

#### Calling Sequence

```
[inter,mu]=intnormalm(X,side=,level=)
```

#### Parameters

- **X** : real matrix **X** containing normal data.
- **side** : side of the confidence interval. Must be "lower", "upper" or "both". Default is "both".
- **level** : confidence level  $1 - \alpha \in (0.5, 1)$ . Default is 0.95.
- **inter** : normal confidence interval for  $\mu$ :
  - if **side**="lower", **inter** is the *lower* confidence bound.
  - if **side**="upper", **inter** is the *upper* confidence bound.
  - if **side**="both", **inter** is the bilateral confidence interval.
- **mu** : estimation of parameter  $\mu$  of the normal distribution.

#### Description

Compute the confidence interval (lower, upper or both) for the parameter  $\mu$  of the normal distribution. `[inter,mu]=intnormalm(X)` is equivalent to `[inter,mu]=intnormalm(X,"both",0.95)`.

#### Examples

```
X=rndnormal(100,3);  
[inter,mu]=intnormalm(X)  
[inter,mu]=intnormalm(X,"upper")  
[inter,mu]=intnormalm(X,level=0.99)
```

#### See Also

tstnormalm1

### 1.4 intnormals – normal confidence interval for $\sigma$

#### Calling Sequence

```
[inter,sigma]=intnormals(X,side=,level=)
```

#### Parameters

- **X** : real matrix **X** containing normal data.
- **side** : side of the confidence interval. Must be "lower", "upper" or "both". Default is "both".
- **level** : confidence level  $1 - \alpha \in (0.5, 1)$ . Default is 0.95.
- **inter** : normal confidence interval for  $\sigma$ :
  - if **side**="lower", **inter** is the *lower* confidence bound.
  - if **side**="upper", **inter** is the *upper* confidence bound.
  - if **side**="both", **inter** is the bilateral confidence interval.
- **sigma** : estimation of parameter  $\sigma$  of the normal distribution.

#### Description

Compute the confidence interval (lower, upper or both) for the parameter  $\sigma$  of the normal distribution. `[inter,sigma]=intnormals(X)` is equivalent to `[inter,sigma]=intnormals(X,"both",0.95)`.

#### Examples

```
X=rndnormal(100,sigma=0.5);  
[inter,sigma]=intnormals(X)  
[inter,sigma]=intnormals(X,"upper")  
[inter,sigma]=intnormals(X,level=0.99)
```

#### See Also

tstnormals1

### 1.5 intpoisson – Poisson confidence interval

#### Calling Sequence

```
[inter,lam]=intpoisson(X,side=,level=)
```

### Parameters

- **X** : real matrix **X** containing Poisson data.
- **side** : side of the confidence interval. Must be "lower", "upper" or "both". Default is "both".
- **level** : confidence level  $1 - \alpha \in (0.5, 1)$ . Default is 0.95.
- **inter** : Poisson confidence interval :
  - if **side**="lower", **inter** is the *lower* confidence bound.
  - if **side**="upper", **inter** is the *upper* confidence bound.
  - if **side**="both", **inter** is the bilateral confidence interval.
- **lam** : estimation of parameter  $\lambda$  of the Poisson distribution.

### Description

Compute the confidence interval (lower, upper or both) for the parameter  $\lambda$  of the Poisson distribution. `[inter, lam]=intpoisson(X)` is equivalent to `[inter, lam]=intpoisson(X, "both", 0.95)`.

### Examples

```
X=rndpoisson(100,3);  
[inter, lam]=intpoisson(X)  
[inter, lam]=intpoisson(X, "upper")  
[inter, lam]=intpoisson(X, level=0.99)
```

## 2 CUMULATIVE DISTRIBUTION FUNCTIONS

### 2.1 cdfbeta – beta type 1 cdf

#### Calling Sequence

```
Y=cdfbeta(X,a,b,c=,d=)
```

#### Parameters

- **X, Y** : real matrices **X** and **Y**.
- **a** : parameter  $a > 0$  of the beta type 1 distribution.
- **b** : parameter  $b > 0$  of the beta type 1 distribution.
- **c** : parameter  $c$  of the beta type 1 distribution. Default is 0.
- **d** : parameter  $d > 0$  of the beta type 1 distribution. Default is 1.

#### Description

Compute in matrix **Y** the cdf of the beta type 1 distribution for each entry  $X_{i,j}$  of matrix **X**. The beta type 1 distribution is defined on  $[c, c + d]$ . `cdfbeta(X,a,b)` is equivalent to `cdfbeta(X,a,b,0,1)`.

#### Examples

- If  $X$  is a beta type 1 ( $a = 2, b = 5, c = 0, d = 1$ ) random variable, compute  $P(X \leq 0.5)$ .  
`cdfbeta(0.5,2,5)`
- If  $X$  is a beta type 1 ( $a = 5, b = 2, c = -0.5, d = 2.5$ ) random variable, compute  $P(0.5 < X \leq 1.5)$ .  
`cdfbeta(1.5,5,2,-0.5,2.5)-cdfbeta(0.5,5,2,-0.5,2.5)`

#### See Also

```
fitbeta, idfbeta, pdfbeta, rndbeta
```

### 2.2 cdfbeta2 – beta type 2 cdf

#### Calling Sequence

```
Y=cdfbeta2(X,a,b,c=,d=)
```

#### Parameters

- **X, Y** : real matrices **X** and **Y**.
- **a** : parameter  $a > 0$  of the beta type 2 distribution.
- **b** : parameter  $b > 0$  of the beta type 2 distribution.
- **c** : parameter  $c$  of the beta type 2 distribution. Default is 0.
- **d** : parameter  $d > 0$  of the beta type 2 distribution. Default is 1.

#### Description

Compute in matrix **Y** the cdf of the beta type 2 distribution for each entry  $X_{i,j}$  of matrix **X**. The beta type 2 distribution is defined on  $[c, +\infty)$ . `cdfbeta2(X,a,b)` is equivalent to `cdfbeta2(X,a,b,0,1)`.



### Examples

- If  $X$  is a beta type 2 ( $a = 2, b = 5, c = 0, d = 1$ ) random variable, compute  $P(X \leq 0.5)$ .  
`cdfbeta2(0.5,2,5)`
- If  $X$  is a beta type 2 ( $a = 5, b = 2, c = -0.5, d = 0.1$ ) random variable, compute  $P(0 < X \leq 0.5)$ .  
`cdfbeta2(0.5,5,2,-0.5,0.1)-cdfbeta2(0,5,2,-0.5,0.1)`

### See Also

`idfbeta2, pdfbeta2, rndbeta2`

## 2.3 cdfbinomial – binomial cdf

### Calling Sequence

`Y=cdfbinomial(X,n,p)`

### Parameters

- $\mathbf{X}, \mathbf{Y}$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .
- $n$  : parameter  $n$  of the binomial distribution. Must be an integer  $\geq 1$ .
- $p$  : parameter  $p \in [0, 1]$  of the binomial distribution.

### Description

Compute in matrix  $\mathbf{Y}$  the cdf of the binomial distribution for each entry  $X_{i,j}$  of matrix  $\mathbf{X}$ .

### Examples

- If  $X$  is a binomial ( $n = 20, p = 0.2$ ) random variable, compute  $P(X \leq 4)$ .  
`cdfbinomial(4,20,0.2)`
- If  $X$  is a binomial ( $n = 20, p = 0.5$ ) random variable, compute  $P(X \geq 7)$ .  
`1-cdfbinomial(7-1,20,0.5)`

### See Also

`pdfbinomial, rndbinomial`

## 2.4 cdfchi2 – $\chi^2$ (central and non-central) cdf

### Calling Sequence

`Y=cdfchi2(X,n)`  
`Y=cdfchi2(X,n,nc)`

### Parameters

- $\mathbf{X}, \mathbf{Y}$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .
- $n$  : parameter  $n$  of the  $\chi^2$  distribution. Must be an integer  $\geq 1$ .
- $nc$  : noncentrality parameter. Must be  $\geq 0$ . Default is 0.

### Description

Compute in matrix  $\mathbf{Y}$  the cdf of the  $\chi^2$  distribution for each entry  $X_{i,j}$  of matrix  $\mathbf{X}$ . `cdfchi2(X,n)` is equivalent to `cdfchi2(X,n,0)`.

### Examples

- If  $X$  is a  $\chi^2$  ( $n = 2$ ) random variable, compute  $P(X \leq 3)$ .  
`cdfchi2(3,2)`
- If  $X$  is a  $\chi^2$  ( $n = 4, nc = 1$ ) random variable, compute  $P(2 < X \leq 6)$ .  
`cdfchi2(6,4,1)-cdfchi2(2,4,1)`

### See Also

`idfchi2, pdfchi2`

## 2.5 cdfcv – sample coefficient of variation cdf

### Calling Sequence

`Y=cdfcv(X,n,cv)`

### Parameters

- $\mathbf{X}, \mathbf{Y}$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .

- **n** : sample size. Must be an integer  $\geq 1$ .
- **cv** : coefficient of variation  $\gamma = \sigma/\mu$ . Must be  $\geq 0$ .

#### Description

Compute in matrix **Y** the cdf of the sample coefficient of variation  $\hat{\gamma}$  for each entry  $X_{i,j}$  of matrix **X**. If  $X_1, \dots, X_n$  are  $n$  normal  $(\mu, \sigma)$  random variables, the sample coefficient of variation  $\hat{\gamma} = S/\bar{X}$ , where  $S$  is the sample standard-deviation and  $\bar{X}$  the sample mean.

#### Example

If  $\hat{\gamma}$  is the sample coefficient of variation corresponding to a sample of  $n = 7$  normal  $(\mu = 10, \sigma = 8)$  random variables, compute  $P(\hat{\gamma} \leq 1)$

```
cdfcv(1,7,8/10)
```

#### See Also

```
idfcv, pdfcv
```

## 2.6 cdfdphase – discrete Phase-Type cdf

#### Calling Sequence

```
Y=cdfdphase(X,Q,q)
```

#### Parameters

- **X, Y** : real matrices **X** and **Y**.
- **Q** : square matrix **Q** of transient probabilities.
- **q** : vector **q** of initial transient probabilities.

#### Description

Compute in matrix **Y** the cdf of the Discrete Phase-Type  $(\mathbf{Q}, \mathbf{q})$  distribution for each entry  $X_{i,j}$  of matrix **X**. The Discrete Phase-Type distribution is defined on  $\{1, 2, 3, \dots\}$ .

#### Examples

- If  $X$  is a Discrete Phase-Type  $\left(\mathbf{Q} = \begin{pmatrix} 0.6 & 0.3 \\ 0.2 & 0.5 \end{pmatrix}, \mathbf{q} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}\right)$  random variable, compute  $P(X \leq 7)$ .  

```
cdfdphase(7, [0.6,0.3;0.2,0.5], [1;0])
```
- If  $X$  is a Discrete Phase-Type  $\left(\mathbf{Q} = \begin{pmatrix} 0.5 & 0.2 \\ 0.1 & 0.8 \end{pmatrix}, \mathbf{q} = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}\right)$  random variable, compute  $P(X \geq 6)$ .  

```
1-cdfdphase(6-1, [0.5,0.2;0.1,0.8], [0.5;0.5])
```

#### See Also

```
momdphase, pdfdphase
```

## 2.7 cdfexponential – exponential cdf

#### Calling Sequence

```
Y=cdfexponential(X,lam)
```

#### Parameters

- **X, Y** : real matrices **X** and **Y**.
- **lam** : parameter  $\lambda > 0$  of the exponential distribution.

#### Description

Compute in matrix **Y** the cdf of the exponential distribution for each entry  $X_{i,j}$  of matrix **X**.

#### Examples

- If  $X$  is an exponential  $(\lambda = 0.5)$  random variable, compute  $P(X \leq 3)$ .  

```
cdfexponential(3,0.5)
```
- If  $X$  is an exponential  $(\lambda = 2)$  random variable, compute  $P(0.5 < X \leq 1.5)$ .  

```
cdfexponential(1.5,2)-cdfexponential(0.5,2)
```

#### See Also

```
idfexponential, pdfexponential, rndexponential
```

## 2.8 cdffisher – Fisher (central and non-central) cdf

### Calling Sequence

```
Y=cdffisher(X,m,n)
Y=cdffisher(X,m,n,nc)
```

### Parameters

- $\mathbf{X}, \mathbf{Y}$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .
- $n, m$  : parameters  $m$  and  $n$  of the Fisher distribution. Must be integers  $\geq 1$ .
- $nc$  : noncentrality parameter. Must be  $\geq 0$ . Default is 0.

### Description

Compute in matrix  $\mathbf{Y}$  the cdf of the Fisher  $(m, n)$  distribution for each entry  $X_{i,j}$  of matrix  $\mathbf{X}$ . `cdffisher(X,m,n)` is equivalent to `cdffisher(X,m,n,0)`.

### Examples

- If  $X$  is a Fisher  $(m = 2, n = 3)$  random variable, compute  $P(X \leq 4)$ .  
`cdffisher(4,2,3)`
- If  $X$  is a Fisher  $(m = 11, n = 9, nc = 4)$  random variable, compute  $P(1 < X \leq 3)$ .  
`cdffisher(3,11,9,4)-cdffisher(1,11,9,4)`

### See Also

`idffisher`, `pdffisher`

## 2.9 cdffoldednormal – folded normal cdf

### Calling Sequence

```
Y=cdffoldednormal(X,mu=,sigma=,c=)
```

### Parameters

- $\mathbf{X}, \mathbf{Y}$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .
- $\mu$  : parameter  $\mu$  of the folded normal distribution. Default is 0.
- $\sigma$  : parameter  $\sigma > 0$  of the folded normal distribution. Default is 1.
- $c$  : parameter  $c$  of the folded normal distribution. Default is 0.

### Description

Compute in matrix  $\mathbf{Y}$  the cdf of the folded normal distribution for each entry  $X_{i,j}$  of matrix  $\mathbf{X}$ . The folded normal  $(\mu, \sigma, c)$  distribution is defined on  $[c, +\infty)$ . `cdffoldednormal(X)` is equivalent to `cdffoldednormal(X,0,1,0)`.

### Examples

- If  $X$  is a folded normal  $(\mu = 0, \sigma = 1, c = 0)$  random variable, compute  $P(X \leq 2)$ .  
`cdffoldednormal(2)`
- If  $X$  is a folded normal  $(\mu = 2, \sigma = 1.5, c = 1)$  random variable, compute  $P(3 < X \leq 5)$ .  
`cdffoldednormal(5,2,1.5,1)-cdffoldednormal(3,2,1.5,1)`

### See Also

`idffoldednormal`, `pdffoldednormal`, `rndfoldednormal`

## 2.10 cdffgamma – gamma cdf

### Calling Sequence

```
Y=cdffgamma(X,a,b=,c=,d=)
```

### Parameters

- $\mathbf{X}, \mathbf{Y}$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .
- $a$  : parameter  $a > 0$  of the gamma distribution.
- $b$  : parameter  $b > 0$  of the gamma distribution. Default is 1.
- $c$  : parameter  $c$  of the gamma distribution. Default is 0.
- $d$  : parameter  $d \neq 0$  of the gamma distribution. Default is 1.

### Description

Compute in matrix  $\mathbf{Y}$  the cdf of the gamma  $(a, b, c, d)$  distribution for each entry  $X_{i,j}$  of matrix  $\mathbf{X}$ . The gamma  $(a, b, c, d)$  distribution is defined on  $[c, +\infty[$ . `cdffgamma(X,a)` is equivalent to `cdffgamma(X,a,1,0,1)`.

### Examples

- If  $X$  is a gamma ( $a = 2, b = 1, c = 0, d = 1$ ) random variable, compute  $P(X \leq 3)$ .  
`cdfgamma(3,2)`
- If  $X$  is a gamma ( $a = 5, b = 0.7, c = 2, d = 1.5$ ) random variable, compute  $P(4 < X \leq 5)$ .  
`cdfgamma(5,5,0.7,2,1.5)-cdfgamma(4,5,0.7,2,1.5)`

### See Also

`fitgamma, idfgamma, pdfgamma, rndgamma`

## 2.11 cdfgev – generalized Extreme Value cdf

### Calling Sequence

`Y=cdfgev(X,a,b=,c=)`

### Parameters

- $\mathbf{X}, \mathbf{Y}$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .
- $a$  : parameter  $a$  of the GEV distribution.
- $b$  : parameter  $b > 0$  of the GEV distribution. Default is 1.
- $c$  : parameter  $c$  of the GEV distribution. Default is 0.

### Description

Compute in matrix  $\mathbf{Y}$  the cdf of the GEV ( $a, b, c$ ) distribution for each entry  $X_{i,j}$  of matrix  $\mathbf{X}$ . `cdfgev(X,a)` is equivalent to `cdfgev(X,a,1,0)`.

### Examples

- If  $X$  is a GEV ( $a = 0.5, b = 1, c = 0$ ) random variable, compute  $P(X \leq 3)$ .  
`cdfgev(3,0.5)`
- If  $X$  is a GEV ( $a = -0.5, b = 1, c = 5$ ) random variable, compute  $P(4 < X \leq 6)$ .  
`cdfgev(6,-0.5,c=5)-cdfgev(4,-0.5,c=5)`

### See Also

`fitgev, idfgev, pdfgev, rndgev`

## 2.12 cdfhypergeometric – hypergeometric cdf

### Calling Sequence

`Y=cdfhypergeometric(X,n,p,N)`

### Parameters

- $\mathbf{X}, \mathbf{Y}$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .
- $n$  : parameter  $n$  of the hypergeometric distribution. Must be an integer in  $\{1, \dots, N\}$ .
- $p$  : parameter  $p \in [0, 1]$  of the hypergeometric distribution.
- $N$  : parameter  $N$  of the hypergeometric distribution. Must be an integer  $\geq 1$ .

### Description

Compute in matrix  $\mathbf{Y}$  the cdf of the hypergeometric ( $n, p, N$ ) distribution for each entry  $X_{i,j}$  of matrix  $\mathbf{X}$ .

### Examples

- If  $X$  is a hypergeometric ( $n = 20, p = 0.3, N = 100$ ) random variable, compute  $P(X \leq 5)$ .  
`cdfhypergeometric(5,20,0.3,100)`
- If  $X$  is a hypergeometric ( $n = 20, p = 0.1, N = 200$ ) random variable, compute  $P(X \geq 3)$ .  
`1-cdfhypergeometric(3-1,20,0.1,200)`

### See Also

`pdfhypergeometric`

## 2.13 cdfjohnson – Johnson's cdf

### Calling Sequence

`Y=cdfjohnson(X,s,a,b,c,d)`

### Parameters

- $\mathbf{X}, \mathbf{Y}$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .
- $s$  : Johnson's system of distribution. Must be "B" for the Johnson's  $S_B$  (bounded) system of distributions or "U" for the Johnson's  $S_U$  (unbounded) system of distributions.
- $a$  : parameter  $a$  of the Johnson's distribution.
- $b$  : parameter  $b > 0$  of the Johnson's distribution.
- $c$  : parameter  $c$  of the Johnson's distribution.
- $d$  : parameter  $d > 0$  of the Johnson's distribution.

### Description

Compute in matrix  $\mathbf{Y}$  the cdf of the Johnson's distribution for each entry  $X_{i,j}$  of matrix  $\mathbf{X}$ .

### Examples

- If  $X$  is a Johnson's bounded ( $a = 4, b = 3, c = 1, d = 5$ ) random variable, compute  $P(X \leq 2.5)$ .  
`cdfjohnson(2.5,"B",4,3,1,5)`
- If  $X$  is a Johnson's unbounded ( $a = 3, b = 4, c = 5, d = 2$ ) random variable, compute  $P(X \geq 3.5)$ .  
`1-cdfjohnson(3.5,"U",3,4,5,2)`

### See Also

`fitjohnson`, `idfjohnson`, `pdfjohnson`, `rndjohnson`

## 2.14 cdflognormal – lognormal cdf

### Calling Sequence

`Y=cdflognormal(X,a=,b=,c=)`

### Parameters

- $\mathbf{X}, \mathbf{Y}$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .
- $a$  : parameter  $a$  of the lognormal distribution. Default is 0.
- $b$  : parameter  $b > 0$  of the lognormal distribution. Default is 1.
- $c$  : parameter  $c$  of the lognormal distribution. Default is 0.

### Description

Compute in matrix  $\mathbf{Y}$  the cdf of the lognormal distribution for each entry  $X_{i,j}$  of matrix  $\mathbf{X}$ . The lognormal distribution is defined on  $[c, +\infty)$ . `cdflognormal(x)` is equivalent to `cdflognormal(x,0,1,0)`.

### Examples

- If  $X$  is a lognormal ( $a = 0.5, b = 2, c = 0$ ) random variable, compute  $P(X \leq 1.5)$ .  
`cdflognormal(1.5,0.5,2)`
- If  $X$  is a lognormal ( $a = -0.5, b = 1, c = 0.5$ ) random variable, compute  $P(2 < X \leq 4)$ .  
`cdflognormal(4,-0.5,c=0.5)-cdflognormal(2,-0.5,c=0.5)`

### See Also

`fitlognormal`, `idflognormal`, `pdflognormal`, `rndlognormal`

## 2.15 cdfmedian – normal sample median cdf

### Calling Sequence

`Y=cdfmedian(X,n,mu=,sigma=)`

### Parameters

- $\mathbf{X}, \mathbf{Y}$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .
- $n$  : parameter  $n$  of the normal  $(\mu, \sigma)$  sample median distribution. Must be an odd integer  $\geq 1$ .
- $\mu$  : parameter  $\mu$  (mean) of the normal distribution. Default is 0.
- $\sigma$  : parameter  $\sigma > 0$  (standard-deviation) of the normal distribution. Default is 1.

### Description

Compute in matrix  $\mathbf{Y}$  the cdf of the normal  $(\mu, \sigma)$  sample median distribution for each entry  $X_{i,j}$  of matrix  $\mathbf{X}$ .

### Examples

- If  $X$  is a normal sample median ( $n = 3, \mu = 0, \sigma = 1$ ) random variable, compute  $P(X \leq -1)$ .

- `cdfmedian(-1,3)`
- If  $X$  is a normal sample median ( $n = 5, \mu = 1, \sigma = 0.4$ ) random variable, compute  $P(0.9 < X \leq 1.2)$ .  
`cdfmedian(1.2,5,1,0.4)-cdfmedian(0.9,5,1,0.4)`

See Also

`idfmedian, pdfmedian`

## 2.16 cdfnormal – normal cdf

Calling Sequence

`Y=cdfnormal(X,mu=,sigma=)`

Parameters

- $\mathbf{X}, \mathbf{Y}$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .
- $\mu$  : parameter  $\mu$  (mean) of the normal distribution. Default is 0.
- $\sigma$  : parameter  $\sigma > 0$  (standard-deviation) of the normal distribution. Default is 1.

Description

Compute in matrix  $\mathbf{Y}$  the cdf of the normal distribution for each entry  $X_{i,j}$  of matrix  $\mathbf{X}$ . `cdfnormal(X)` is equivalent to `cdfnormal(X,0,1)`.

Examples

- If  $X$  is a normal ( $\mu = 0, \sigma = 1$ ) random variable, compute  $P(X \leq -2)$ .  
`cdfnormal(-2)`
- If  $X$  is a normal ( $\mu = 3, \sigma = 2$ ) random variable, compute  $P(1 < X \leq 5)$ .  
`cdfnormal(5,3,2)-cdfnormal(1,3,2)`

See Also

`idfnormal, pdfnormal, rndnormal`

## 2.17 cdfpareto – Pareto cdf

Calling Sequence

`Y=cdfpareto(X,a,b=,c=)`

Parameters

- $\mathbf{X}, \mathbf{Y}$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .
- $a$  : parameter  $a$  of the Pareto distribution.
- $b$  : parameter  $b > 0$  of the Pareto distribution. Default is 1.
- $c$  : parameter  $c$  of the Pareto distribution. Default is 0.

Description

Compute in matrix  $\mathbf{Y}$  the cdf of the Pareto distribution for each entry  $X_{i,j}$  of matrix  $\mathbf{X}$ . The Pareto distribution is defined on

- $[c, +\infty)$  if  $a \geq 0$ ,
- $[c, c - b/a]$  if  $a < 0$ .

`cdfpareto(X,a)` is equivalent to `cdfpareto(X,a,1,0)`.

Examples

- If  $X$  is a Pareto ( $a = 0.5, b = 1, c = 0$ ) random variable, compute  $P(X \leq 1.5)$ .  
`cdfpareto(1.5,0.5)`
- If  $X$  is a Pareto ( $a = -2, b = 6, c = 1$ ) random variable, compute  $P(2 < X \leq 3)$ .  
`cdfpareto(3,-2,6,1)-cdfpareto(2,-2,6,1)`

See Also

`idfpareto, pdfpareto, rndpareto`

## 2.18 cdfpascal – Pascal cdf

Calling Sequence

`Y=cdfpascal(X,n,p)`

**Parameters**

- **X, Y** : real matrices **X** and **Y**
- **n** : parameter  $n$  of the Pascal distribution. Must be an integer  $\geq 1$ .
- **p** : parameter  $p \in (0, 1]$  of the Pascal distribution.

**Description**

Compute in matrix **Y** the cdf of the Pascal distribution for each entry  $X_{i,j}$  of matrix **X**.

**Examples**

- If  $X$  is a Pascal ( $n = 2, p = 0.3$ ) random variable, compute  $P(X \leq 10)$ .  
`cdfpascal(10,2,0.3)`
- If  $X$  is a Pascal ( $n = 7, p = 0.5$ ) random variable, compute  $P(X \geq 15)$ .  
`1-cdfpascal(15-1,7,0.5)`

**See Also**

`pdfpascal, rndpascal`

**2.19 cdfpoisson – Poisson cdf****Calling Sequence**

`Y=cdfpoisson(X,lam)`

**Parameters**

- **X, Y** : real matrices **X** and **Y**.
- **lam** : parameter  $\lambda > 0$  of the Poisson distribution.

**Description**

Compute in matrix **Y** the cdf of the Poisson distribution for each entry  $X_{i,j}$  of matrix **X**.

**Examples**

- If  $X$  is a Poisson ( $\lambda = 0.8$ ) random variable, compute  $P(X \leq 2)$ .  
`cdfpoisson(2,0.8)`
- If  $X$  is a Poisson ( $\lambda = 3$ ) random variable, compute  $P(X \geq 4)$ .  
`1-cdfpoisson(4-1,3)`

**See Also**

`pdfpoisson, rndpoisson`

**2.20 cdfrnge – normal range cdf****Calling Sequence**

`Y=cdfrnge(X,n)`  
`Y=cdfrnge(X,n,sigma)`

**Parameters**

- **X, Y** : real matrices **X** and **Y**.
- **n** : parameter  $n$  of the normal range distribution. Must be an integer  $\geq 2$ .
- **sigma** : parameter  $\sigma > 0$  (standard-deviation) of the normal distribution. Default is 1.

**Description**

Compute in matrix **Y** the cdf of the normal range distribution for each entry  $X_{i,j}$  of matrix **X**. `cdfrnge(X,n)` is equivalent to `cdfrnge(X,n,1)`.

**Examples**

- If  $R$  is a normal range ( $n = 3, \sigma = 1$ ) random variable, compute  $P(R \leq 2)$ .  
`cdfrnge(2,3)`
- If  $R$  is a normal range ( $n = 5, \sigma = 1.5$ ) random variable, compute  $P(R \geq 3)$ .  
`1-cdfrnge(3,5,1.5)`

**See Also**

`pdfrng`

## 2.21 cdfstandev – normal sample standard-deviation cdf

### Calling Sequence

```
Y=cdfstandev(X,n)
Y=cdfstandev(X,n,sigma)
```

### Parameters

- $\mathbf{X}, \mathbf{Y}$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .
- $n$  : parameter  $n$  of the normal sample standard-deviation distribution. Must be an integer  $\geq 2$ .
- $\sigma$  : parameter  $\sigma > 0$  (standard-deviation) of the normal distribution. Default is 1.

### Description

Compute in matrix  $\mathbf{Y}$  the cdf of the normal sample standard-deviation distribution for each entry  $X_{i,j}$  of matrix  $\mathbf{X}$ . `cdfstandev(X,n)` is equivalent to `cdfstandev(X,n,1)`.

### Examples

- If  $S$  is a normal sample standard-deviation ( $n = 3, \sigma = 1$ ) random variable, compute  $P(S \leq 2)$ .  
`cdfstandev(2,3)`
- If  $S$  is a normal sample standard-deviation ( $n = 9, \sigma = 3.5$ ) random variable, compute  $P(S \geq 3)$ .  
`1-cdfstandev(3,9,3.5)`

### See Also

`idfstandev`, `pdfstandev`, `rndstandev`

## 2.22 cdfstudent – Student (central and non-central) cdf

### Calling Sequence

```
Y=cdfstudent(X,n)
Y=cdfstudent(X,n,nc)
```

### Parameters

- $\mathbf{X}, \mathbf{Y}$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .
- $n$  : parameter  $n$  of the Student distribution. Must be an integer  $\geq 1$ .
- $nc$  : noncentrality parameter. Must be  $\geq 0$ . Default is 0.

### Description

Compute in matrix  $\mathbf{Y}$  the cdf of the Student distribution for each entry  $X_{i,j}$  of matrix  $\mathbf{X}$ . `cdfstudent(X,n)` is equivalent to `cdfstudent(X,n,0)`.

### Examples

- If  $X$  is a Student ( $n = 2$ ) random variable, compute  $P(X \leq 3)$ .  
`cdfstudent(3,2)`
- If  $X$  is a student ( $n = 20, nc = 3$ ) random variable, compute  $P(-2 < X \leq 2)$ .  
`cdfstudent(2,20,3)-cdfstudent(-2,20,3)`

### See Also

`idfstudent`, `pdfstudent`

## 2.23 cdfweibull – Weibull cdf

### Calling Sequence

```
Y=cdfweibull(X,a,b=,c=)
```

### Parameters

- $\mathbf{X}, \mathbf{Y}$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .
- $a$  : parameter  $a > 0$  of the Weibull distribution.
- $b$  : parameter  $b > 0$  of the Weibull distribution. Default is 1.
- $c$  : parameter  $c$  of the Weibull distribution. Default is 0.

### Description

Compute in matrix  $\mathbf{Y}$  the cdf of the Weibull distribution for each entry  $X_{i,j}$  of matrix  $\mathbf{X}$ . The Weibull distribution is defined on  $[c, +\infty)$ . `cdfweibull(X,a)` is equivalent to `cdfweibull(X,a,1,0)`.



### Examples

- If  $X$  is a Weibull ( $a = 2, b = 1, c = 0$ ) random variable, compute  $P(X \leq 1.5)$ .  
`cdfweibull(1.5,2)`
- If  $X$  is a Weibull ( $a = 5, b = 3, c = 2$ ) random variable, compute  $P(4 < X \leq 6)$ .  
`cdfweibull(6,5,3,2)-cdfweibull(4,5,3,2)`

### See Also

`fitweibull, idfweibull, pdfweibull, rndweibull`

## 3 DESIGN OF EXPERIMENTS

### 3.1 boxbehnken – Box-Behnken designs

#### Calling Sequence

```
Z=boxbehnken(k)
Z=boxbehnken(k,n0)
```

#### Parameters

- **k** : number of factors. Must be an integer in  $\{2, \dots, 10\}$ .
- **n0** : number  $n_0$  of center points. Default is 1.
- **Z** : real matrix **Z**.

#### Description

Compute in matrix **Z** the Box-Behnken design for  $k$  factors plus  $n_0$  center points.

#### Examples

```
Z1=boxbehnken(3)
Z2=boxbehnken(5,3)
```

### See Also

`centralcomposite, equiradial, factorial2, plackettburman`

### 3.2 boxcoxlinear – Box-Cox linearity transformation

#### Calling Sequence

```
[lam,rmax]=boxcoxlinear(x,y)
```

#### Parameters

- **x,y** : real vectors **x** and **y** of the same size. Entries  $x_i$  of vector **x** must be  $> 0$ .
- **lam** : parameter  $\lambda$  of the Box-Cox transformation.
- **rmax** : coefficient  $r$  maximizing the correlation between **z** and **y** where  $z_i = (x_i^\lambda - 1)/\lambda$ .

#### Description

Compute the parameter  $\lambda$  of the Box-Cox transformation  $z_i = (x_i^\lambda - 1)/\lambda$  maximizing the correlation  $r$  between **z** and **y**.

#### Examples (see Figure 1)

```
x=linspace(1,5,10)';
y=exp(x)+rndnormal(10,0,3);
xset("window",0);xbasc(0)
plot2d(x,y,-9);
xtitle("", "xi", "yi");xselect()
[lam,rmax]=boxcoxlinear(x,y); [lam,rmax]
xt=(x^lam-1)/lam;
xset("window",1);xbasc(1)
plot2d(xt,y,-9)
xtitle("", "xi", "zi");xselect()
```

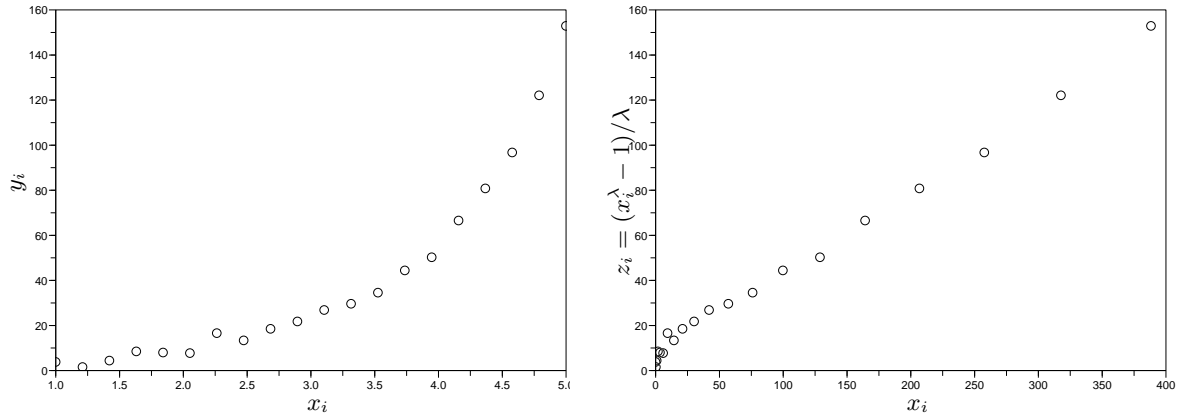


Figure 1: Example of function `boxcoxlinear`

### 3.3 centralcomposite – central composite designs

#### Calling Sequence

```
Z=centralcomposite(k,delta=,n0=)
```

#### Parameters

- **Z** : real matrix **Z**.
- **k** : number of factors. Must be an integer in  $\{2, \dots, 11\}$ .
- **delta** : axial point distance  $\delta$  from the origin. Must be  $> 0$ . Default is  $\delta = (n_F)^{(1/4)}$  where  $n_F$  is the number of experiment in the full or fractional factorial part of the central composite design.
- **n0** : number  $n_0$  of center points. Default is 0.

#### Description

Compute in matrix **Z** the central composite design for  $k$  factors where the axial points are located at a distance  $\delta$  from the origin plus  $n_0$  center points.

#### Examples

```
Z1=centralcomposite(3,n0=2)
Z2=centralcomposite(5,1)
```

#### See Also

`boxbehnken`, `equiradial`, `factorial2`, `plackettburman`, `simpdex`

### 3.4 coded2natural – coded to natural variables

#### Calling Sequence

```
Y=coded2natural(Z,LU)
```

#### Parameters

- **Z, Y** : real matrices **Z** and **Y**.
- **LU** : two columns real matrix. The 1st column contains the lower bounds for each natural variable and the 2nd column contains the upper bounds for each natural variable.

#### Description

Compute in matrix **Y** the natural values corresponding to the coded values in matrix **Z**.

#### Examples

```
Z=centralcomposite(3)
Y=coded2natural(Z,[10,20;100,200;50,80])
```

#### See Also

`natural2coded`

### 3.5 doxpand – design expansion

#### Calling Sequence

```
X=doxpand(Z)
X=doxpand(Z,ex)
```

#### Parameters

- $\mathbf{X}, \mathbf{Z}$  : real matrices  $\mathbf{X}$  and  $\mathbf{Z}$ .
- $\mathbf{ex}$  : expansion type. Must be "x" for a “linear” expansion, "x+xx" for a “linear+interaction” expansion and "x+xx+x2" for a “linear+interaction+quadratic” expansion. Default is "x+xx".

#### Description

Expand matrix  $\mathbf{Z}$  in matrix  $\mathbf{X}$  according to the expansion type  $\mathbf{ex}$ . In any case, a leftmost column of “1” is added.

#### Examples

```
Z=centralcomposite(3)
doxpand(Z)
doxpand(Z,"x+xx")
doxpand(Z,"x+xx+x2")
```

### 3.6 doxptim – design optimisation

#### Calling Sequence

```
[xopt,yopt]=doxptim(a)
[xopt,yopt]=doxptim(a,opt)
```

#### Parameters

- $\mathbf{a}$  : vector  $\mathbf{a}$  of regression coefficients. Size of  $\mathbf{a}$  must be  $p = 1 + k(k + 1)/2$  (linear+interaction) or  $p = 1 + k(k + 3)/2$  (linear+interaction+quadratic).
- $\mathbf{opt}$  : must be "min" or "max". Default is "max".
- $\mathbf{xopt}$  : optimal design point  $\mathbf{x}^*$  maximizing (or minimizing) the response  $y$  in the hypercube  $[-1, +1]^k$ .
- $\mathbf{yopt}$  : optimal value  $y^*$  of the response when  $\mathbf{x} = \mathbf{x}^*$ .

#### Description

Compute the optimal design point  $\mathbf{x}^*$  maximizing (if  $\mathbf{opt} = \text{"max"}$ ) or minimizing (if  $\mathbf{opt} = \text{"min"}$ ) the response  $y$  in the hypercube  $[-1, +1]^k$ . If the size of the regression vector  $\mathbf{a}$  is

- $p = 1 + k(k + 1)/2$  then a “linear+interaction” model is assumed.
- $p = 1 + k(k + 3)/2$  then a “linear+interaction+quadratic” model is assumed.

$[\mathbf{xopt}, \mathbf{yopt}] = \text{doxptim}(\mathbf{a})$  is equivalent to  $[\mathbf{xopt}, \mathbf{yopt}] = \text{doxptim}(\mathbf{a}, \text{"max"})$ .

#### Examples

```
a=[80;2;3;-1;-1.5;-2];
[xopt,yopt]=doxptim(a)
[xopt,yopt]=doxptim(a,"min")
```

### 3.7 equiradial – equiradial designs

#### Calling Sequence

```
Z=equiradial(n)
Z=equiradial(n,n0)
```

#### Parameters

- $\mathbf{Z}$  : real matrix  $\mathbf{Z}$ .
- $\mathbf{n}$  : number of the equiradial vertices. Must be an integer  $\geq 3$ .
- $\mathbf{n0}$  : number of center points. Default is 0.

#### Description

Compute in matrix  $\mathbf{Z}$  the Equiradial design for  $n$  vertices plus  $n_0$  center points.

#### Examples

```
Z1=equiradial(7)
Z2=equiradial(15,3)
```

See Also

boxbehnken, centralcomposite, factorial2, plackettburman, simplex

### 3.8 factorial2 – two levels full and fractional factorial designs

Calling Sequence

```
Z=factorial2(k,gen=,n0=)
```

Parameters

- **Z** : real matrix **Z**.
- **k** : number of factors. Must be an integer  $\geq 1$ .
- **gen** : vector **g** of generators. Default is [].
- **n0** : number of center points. Default is 0.

Description

Compute in matrix **Z** the two levels (full or fractional) factorial design for  $k$  factors using generators in vector **g** plus  $n_0$  center points. A list of usefull generators are in Table 1. **factorial2(k)** is equivalent to **factorial2(k, [], 0)**.

Examples

```
Z1=factorial2(3,n0=2)
Z2=factorial2(5,["+AB","-AC"])
```

See Also

boxbehnken, centralcomposite, equiradial, plackettburman, simplex

### 3.9 mulreg – multilinear regression analysis

Calling Sequence

```
res=mulreg(X,y)
```

Parameters

- **X** : real matrix **X** of size  $(n, p)$ .
- **y** : real column vector **y** of size  $(n, 1)$ .
- **res** : list containing the results of the multilinear regression analysis :
  - **res(1)** : vector  $(m, n, p)$  where  $m$  is the number of identical rows in matrix **X**, i.e. the number of repeated experiments. If all experiments are different,  $m = n$ .
  - **res(2)** : vector of estimated regression coefficients  $\hat{\mathbf{a}}$  of size  $(p, 1)$ .
  - **res(3)** : vector of estimated responses  $\hat{\mathbf{y}}$  of size  $(n, 1)$ .
  - **res(4)** : vector of residuals **e** of size  $(n, 1)$ .
  - **res(5)** : Sum Square of Regression *SSR*.
  - **res(6)** : Sum Square of Error *SSE*.
  - **res(7)** : coefficient of determination  $R^2$ .
  - **res(8)** : Mean Square of Regression *MSR*.
  - **res(9)** : Mean Square of Error *MSE*.
  - **res(10)** : adjusted coefficient of determination  $R_a^2$ .
  - **res(11)** : vector of studentized residuals **r** of size  $(n, 1)$ . Usefull for detecting outliers.
  - **res(12)** : vector of Cook's distance **d** of size  $(n, 1)$ . Usefull for detecting influence points.
  - **res(13)** : vector of 95% lower confidence interval bounds  $\hat{\mathbf{a}}_{\text{inf}}$  for regression coefficients of size  $(p, 1)$ .
  - **res(14)** : vector of 95% upper confidence interval bounds  $\hat{\mathbf{a}}_{\text{sup}}$  for regression coefficients of size  $(p, 1)$ .
  - **res(15)** : vector of  $p$ -values for regression coefficients of size  $(p, 1)$ .
  - **res(16)** : Sum Square of Pure Error *SSPE*.
  - **res(17)** : Sum Square Lack Of Fit *SSLOF*.
  - **res(18)** : Mean Square of Pure Error *MSPE*.
  - **res(19)** : Mean Square Lack Of Fit *MSLOF*.
  - **res(20)** :  $p$ -value for Lack of Fit.

k	Fraction	# experiments	gen
3	$2_{III}^{3-1}$	4	" +AB"
4	$2_{IV}^{4-1}$	8	" +ABC"
5	$2_V^{5-1}$	16	" +ABCD"
	$2_{III}^{5-2}$	8	[" +AB", " +AC"]
6	$2_{VI}^{6-1}$	32	" +ABCDE"
	$2_{IV}^{6-2}$	16	[" +ABC", " +BCD"]
	$2_{III}^{6-3}$	8	[" +AB", " +AC", " +BC"]
7	$2_{VII}^{7-1}$	64	" +ABCDEF"
	$2_{IV}^{7-2}$	32	[" +ABCD", " +ABDE"]
	$2_{IV}^{7-3}$	16	[" +ABC", " +BCD", " +ACD"]
	$2_{III}^{7-4}$	8	[" +AB", " +AC", " +BC", " +ABC"]
8	$2_V^{8-2}$	64	[" +ABCD", " +ABEF"]
	$2_{IV}^{8-3}$	32	[" +ABC", " +ABD", " +BCDE"]
	$2_{IV}^{8-4}$	16	[" +BCD", " +ACD", " +ABC", " +ABD"]
9	$2_{VI}^{9-2}$	128	[" +ACDFG", " +BCEFG"]
	$2_{IV}^{9-3}$	64	[" +ABCD", " +ACEF", " +CDEF"]
	$2_{IV}^{9-4}$	32	[" +BCDE", " +ACDE", " +ABDE", " +ABCE"]
10	$2_V^{10-3}$	128	[" +ABCG", " +ACDE", " +ACDF"]
	$2_{IV}^{10-4}$	64	[" +BCDF", " +ACDF", " +ABDE", " +ABCE"]
	$2_{IV}^{10-5}$	32	[" +ABCD", " +ABCE", " +ABDE", " +ACDE", " +BCDE"]
	$2_{III}^{10-6}$	16	[" +ABC", " +BCD", " +ACD", " +ABD", " +ABCD", " +AB"]
11	$2_V^{11-4}$	128	[" +ABCG", " +BCDE", " +ACDF", " +ABCDEFG"]
	$2_{IV}^{11-5}$	64	[" +CDE", " +ABCD", " +ABF", " +BDEF", " +ADEF"]
	$2_{IV}^{11-6}$	32	[" +ABC", " +BCD", " +CDE", " +ACD", " +ADE", " +BDE"]
	$2_{III}^{11-7}$	16	[" +ABC", " +BCD", " +ACD", " +ABD", " +ABCD", " +AB", " +AC"]

Table 1: Possible generators for `factorial2`

## Description

Perform a multilinear regression analysis and store the various results in list **res**.

## Example

```
Z=factorial2(2,n0=1)
Z=Z.*ones(2,1)
n=size(Z,"r")
y=round((230+Z(:,1)*18+7*Z(:,1).*Z(:,2)+rdnormal(n,0,0.3))*10)/10
X=doxpad(Z,"x+xx")
res=mulreg(X,y);
res(2)
//
mulregdisp(res)
mulregplot(res)
```

## See Also

`mulregdisp`, `mulregplot`

## 3.10 mulregdisp – multilinear regression analysis results display

### Calling Sequence

```
mulregdisp(res)
```

### Parameters

- **res** : list containing the results of the multilinear regression analysis (see `mulreg`).

### Description

Display all available results after the use of function `mulreg`.

**Example** (see `mulreg`).

### See Also

`mulreg`, `mulregplot`

## 3.11 mulregplot – multilinear regression analysis results plot

### Calling Sequence

```
mulregplot(res)
```

### Parameters

- **res** : list containing the results of the multilinear regression analysis (see `mulreg`).

### Description

Plot vector **y** and vector  $\hat{\mathbf{y}}$  after the use of function `mulreg`.

**Example** (see `mulreg`).

### See Also

`mulreg`, `mulregdisp`

## 3.12 natural2coded – natural to coded variables

### Calling Sequence

```
Z=natural2coded(Y,LU)
```

### Parameters

- **Z, Y** : real matrices **Z** and **Y**.
- **LU** : two columns real matrix. The 1st column contains the lower bounds for each natural variable and the 2nd column contains the upper bounds for each natural variable.

### Description

Compute in matrix **Z** the coded values corresponding to the natural values in matrix **Y**.

## Examples

```
Y=[15,100,80;20,150,50;10,200,65]
Z=natural2coded(Y,[10,20;100,200;50,80])
```

## See Also

`coded2natural`

## 3.13 plackettburman – Plackett-Burman designs

### Calling Sequence

```
Z=plackettburman(k)
Z=plackettburman(k,n0)
```

### Parameters

- $Z$  : real matrix  $\mathbf{Z}$ .
- $k$  : number of factors. Must be an integer in  $\{3, 7, \dots, 47\}$ .
- $n_0$  : number of center points. Default is 0.

### Description

Compute in matrix  $\mathbf{Z}$  the Plackett-Burman design for  $k$  factors plus  $n_0$  center points.

### Examples

```
Z1=plackettburman(7)
Z2=plackettburman(15,3)
```

## See Also

`boxbehnken`, `centralcomposite`, `equiradial`, `factorial2`, `simpdex`

## 3.14 simpdex – simplex designs

### Calling Sequence

```
Z=simpdex(k)
Z=simpdex(k,n0)
```

### Parameters

- $Z$  : real matrix  $\mathbf{Z}$ .
- $k$  : number of factors. Must be an integer  $\geq 1$ .
- $n_0$  : number of center points. Default is 0.

### Description

Compute in matrix  $\mathbf{Z}$  the Simplex design for  $k$  factors plus  $n_0$  center points.

### Examples

```
Z1=simpdex(7)
Z2=simpdex(15,3)
```

## See Also

`boxbehnken`, `centralcomposite`, `equiradial`, `factorial2`, `plackettburman`

## 4 ESTIMATION

### 4.1 fitbeta – beta type 1 parameters estimation

#### Calling Sequence

```
[a,b,c,d]=fitbeta(X)
```

#### Parameters

- $X$  : real matrix  $\mathbf{X}$ .
- $a$  : parameter  $a > 0$  of the beta type 1 distribution.
- $b$  : parameter  $b > 0$  of the beta type 1 distribution.

- $c$  : parameter  $c$  of the beta type 1 distribution.
- $d$  : parameter  $d > 0$  of the beta type 1 distribution.

#### Description

Compute the Maximum Likelihood estimates for parameters  $(a, b, c, d)$  of the beta type 1 distribution.

#### Example

```
X=rndbeta(n,5,2,-0.5,2.5);
[a,b,c,d]=fitbeta(X);
mprintf("a = %g    b = %g    c = %g    d = %g\n",a,b,c,d)
```

#### See Also

cdfbeta, idfbeta, pdfbeta, rndbeta

### 4.2 fitgamma – gamma parameters estimation

#### Calling Sequence

```
[a,b,c]=fitgamma(X)
```

#### Parameters

- $X$  : real matrix  $\mathbf{X}$ .
- $a$  : parameter  $a > 0$  of the gamma distribution.
- $b$  : parameter  $b > 0$  of the gamma distribution.
- $c$  : parameter  $c$  of the gamma distribution.

#### Description

Compute Maximum Likelihood estimates for parameters  $(a, b, c)$  of the gamma distribution.

#### Examples

```
X=rndgamma(1000,2,3,4);
[a,b,c]=fitgamma(X);
mprintf("a = %g    b = %g    c = %g\n",a,b,c)
```

#### See Also

cdfgamma, idfgamma, pdfgamma, rndgamma

### 4.3 fitgev – generalized Extreme Value parameters estimation

#### Calling Sequence

```
[a,b,c]=fitgev(X)
```

#### Parameters

- $X$  : real matrix  $\mathbf{X}$ .
- $a$  : parameter  $a$  of the GEV distribution.
- $b$  : parameter  $b > 0$  of the GEV distribution.
- $c$  : parameter  $c$  of the GEV distribution.

#### Description

Compute Maximum Likelihood estimates for parameters  $(a, b, c)$  of the GEV distribution.

#### Examples

```
X=rndgev(1000,0.5,2,5);
[a,b,c]=fitgev(X);
mprintf("a = %g    b = %g    c = %g\n",a,b,c)
```

### 4.4 fitjohnson – Johnson parameters estimation

#### Calling Sequence

```
[s,a,b,c,d]=fitjohnson(X)
```

#### Parameters



- $\mathbf{X}$  : real matrix  $\mathbf{X}$ .
- $s$  : Johnson's system of distribution. Must be "B" for the Johnson's  $S_B$  (bounded) system of distributions or "U" for the Johnson's  $S_U$  (unbounded) system of distributions.
- $a$  : parameter  $a$  of the Johnson distribution.
- $b$  : parameter  $b > 0$  of the Johnson distribution.
- $c$  : parameter  $c$  of the Johnson distribution.
- $d$  : parameter  $d > 0$  of the Johnson distribution.

#### Description

Compute Maximum Likelihood estimates for parameters  $(a, b, c, d)$  of the Johnson distribution.

#### Examples

```
Xb=rndjohnson(1000,"B",4,3,1,5);
[s,a,b,c,d]=fitjohnson(Xb);
mprintf("s = %s      a = %g      b = %g      c = %g      d = %g\n",s,a,b,c,d)
//
Xu=rndjohnson(1000,"U",3,4,5,2);
[s,a,b,c,d]=fitjohnson(Xu);
mprintf("s = %s      a = %g      b = %g      c = %g      d = %g\n",s,a,b,c,d)
```

#### See Also

cdfjohnson, idfjohnson, pdfjohnson, rndjohnson

### 4.5 fitlognormal – lognormal parameters estimation

#### Calling Sequence

```
[a,b,c]=fitlognormal(X)
```

#### Parameters

- $\mathbf{X}$  : real matrix  $\mathbf{X}$ .
- $a$  : parameter  $a$  of the lognormal distribution.
- $b$  : parameter  $b > 0$  of the lognormal distribution.
- $c$  : parameter  $c$  of the lognormal distribution.

#### Description

Compute Maximum Likelihood estimates for parameters  $(a, b, c)$  of the lognormal distribution.

#### Examples

```
X=rndlognormal(1000,0.5,2);
[a,b,c]=fitlognormal(X);
mprintf("a = %g      b = %g      c = %g\n",a,b,c)
```

#### See Also

cdflognormal, idflognormal, pdflognormal, rndlognormal

### 4.6 fitweibull – Weibull parameters estimation

#### Calling Sequence

```
[a,b,c]=fitweibull(X)
```

#### Parameters

- $\mathbf{X}$  : real matrix  $\mathbf{X}$ .
- $a$  : parameter  $a > 0$  of the Weibull distribution.
- $b$  : parameter  $b > 0$  of the Weibull distribution.
- $c$  : parameter  $c$  of the Weibull distribution.

#### Description

Compute Maximum Likelihood estimates for parameters  $(a, b, c)$  of the Weibull distribution.

#### Examples

```

X=rndweibull(1000,5,3,2);
[a,b,c]=fitweibull(X);
mprintf("a = %g      b = %g      c = %g\n",a,b,c)

```

See Also

cdfweibull, idfweibull, pdfweibull, rndweibull

## 5 INVERSE CUMULATIVE DISTRIBUTION FUNCTIONS

### 5.1 idfbeta – beta type 1 idf

Calling Sequence

```
X=idfbeta(Y,a,b,c=,d=)
```

Parameters

- **X**, **Y** : real matrices **X** and **Y**. Each entry  $Y_{i,j}$  of **Y** must be in  $]0, 1[$ .
- **a** : parameter  $a > 0$  of the beta type 1 distribution.
- **b** : parameter  $b > 0$  of the beta type 1 distribution.
- **c** : parameter  $c$  of the beta type 1 distribution. Default is 0.
- **d** : parameter  $d > 0$  of the beta type 1 distribution. Default is 1.

Description

Compute in **X** the idf of the beta type 1 distribution for each entry  $Y_{i,j}$  of matrix **Y**. The beta type 1 distribution is defined on  $[c, c + d]$ . `idfbeta(Y,a,b)` is equivalent to `idfbeta(Y,a,b,0,1)`.

Examples

- If  $X$  is a beta type 1 ( $a = 2, b = 5, c = 0, d = 1$ ) random variable, compute  $x$  such that  $P(X \leq x) = 0.1$ .  
`x=idfbeta(0.1,2,5)`  
`cdfbeta(x,2,5)`
- If  $X$  is a beta type 1 ( $a = 5, b = 2, c = -0.5, d = 2.5$ ) random variable, compute  $x$  such that  $P(X \geq x) = 0.2$ .  
`x=idfbeta(1-0.2,5,2,-0.5,2.5)`  
`1-cdfbeta(x,5,2,-0.5,2.5)`

See Also

cdfbeta, fitbeta, pdfbeta, rndbeta

### 5.2 idfbeta2 – beta type 2 idf

Calling Sequence

```
X=idfbeta2(Y,a,b,c=,d=)
```

Parameters

- **X**, **Y** : real matrices **X** and **Y**. Each entry  $Y_{i,j}$  of **Y** must be in  $]0, 1[$ .
- **a** : parameter  $a > 0$  of the beta type 2 distribution.
- **b** : parameter  $b > 0$  of the beta type 2 distribution.
- **c** : parameter  $c$  of the beta type 2 distribution. Default is 0.
- **d** : parameter  $d > 0$  of the beta type 2 distribution. Default is 1.

Description

Compute in **X** the idf of the beta type 2 distribution for each entry  $Y_{i,j}$  of matrix **Y**. The beta type 2 distribution is defined on  $[c, +\infty)$ . `idfbeta2(Y,a,b)` is equivalent to `idfbeta2(Y,a,b,0,1)`.

Examples

- If  $X$  is a beta type 2 ( $a = 2, b = 5, c = 0, d = 1$ ) random variable, compute  $x$  such that  $P(X \leq x) = 0.1$ .  
`x=idfbeta2(0.1,2,5)`  
`cdfbeta2(x,2,5)`
- If  $X$  is a beta type 2 ( $a = 5, b = 2, c = -0.5, d = 0.1$ ) random variable, compute  $x$  such that  $P(X \geq x) = 0.2$ .  
`x=idfbeta2(1-0.2,5,2,-0.5,0.1)`  
`1-cdfbeta2(x,5,2,-0.5,0.1)`

See Also

`cdfbeta2`, `pdfbeta2`, `rndbeta2`

### 5.3 `idfchi2` – $\chi^2$ (central and non-central) idf

Calling Sequence

```
X=idfchi2(Y,n)
X=idfchi2(Y,n,nc)
```

Parameters

- **X, Y** : real matrices **X** and **Y**. Each entry  $Y_{i,j}$  of **Y** must be in  $(0, 1)$ .
- **n** : parameter  $n$  of the  $\chi^2$  distribution. Must be an integer  $\geq 1$ .
- **nc** : noncentrality parameter. Must be  $\geq 0$ . Default is 0.

Description

Compute in matrix **X** the idf of the  $\chi^2$  distribution for each entry  $Y_{i,j}$  of matrix **Y**. `idfchi2(Y,n)` is equivalent to `idfchi2(Y,n,0)`.

Examples

- If  $X$  is a  $\chi^2$  ( $n = 2$ ) random variable, compute  $x$  such that  $P(X \leq x) = 0.1$ .  

```
x=idfchi2(0.1,2)
cdfchi2(x,2)
```
- If  $X$  is a  $\chi^2$  ( $n = 4, nc = 1$ ) random variable, compute  $x$  such that  $P(X \geq x) = 0.2$ .  

```
x=idfchi2(1-0.2,4,1)
1-cdfchi2(x,4,1)
```

See Also

`cdfchi2`, `pdfchi2`

### 5.4 `idfcv` – sample coefficient of variation idf

Calling Sequence

```
X=idfcv(Y,n,cv)
```

Parameters

- **X, Y** : real matrices **X** and **Y**. Each entry  $Y_{i,j}$  of **Y** must be in  $(0, 1)$ .
- **n** : sample size. Must be an integer  $\geq 1$ .
- **cv** : coefficient of variation  $\gamma = \sigma/\mu$ . Must be  $\geq 0$ .

Description

Compute in matrix **X** the idf of the sample coefficient of variation  $\hat{\gamma}$  for each entry  $Y_{i,j}$  of matrix **Y**. If  $X_1, \dots, X_n$  are  $n$  normal  $(\mu, \sigma)$  random variables, the sample coefficient of variation  $\hat{\gamma} = S/\bar{X}$ , where  $S$  is the sample standard-deviation and  $\bar{X}$  the sample mean.

Example

If  $\hat{\gamma}$  is the sample coefficient of variation corresponding to a sample of  $n = 7$  normal  $(\mu = 10, \sigma = 8)$  random variables, compute  $x$  such that  $P(\hat{\gamma} \leq x) = 0.95$

```
x=idfcv(0.95,7,8/10)
```

See Also

`cdfcv`, `pdfcv`

### 5.5 `idfexponential` – exponential idf

Calling Sequence

```
X=idfexponential(Y,lam)
```

Parameters

- **X, Y** : real matrices **X** and **Y**. Each entry  $Y_{i,j}$  of **Y** must be in  $]0, 1[$ .
- **lam** : parameter  $\lambda > 0$  of the exponential distribution.

## Description

Compute in matrix **X** the idf of the exponential distribution for each entry  $Y_{i,j}$  of matrix **Y**.

### Examples

- If  $X$  is an exponential ( $\lambda = 0.5$ ) random variable, compute  $x$  such that  $P(X \leq x) = 0.1$ .  
`x=idfexponential(0.1,0.5)`  
`cdfexponential(x,0.5)`
- If  $X$  is an exponential ( $\lambda = 2$ ) random variable, compute  $x$  such that  $P(X \geq x) = 0.2$ .  
`x=idfexponential(1-0.2,2)`  
`1-cdfexponential(x,2)`

### See Also

`cdfexponential`, `pdfexponential`, `rndexponential`

## 5.6 idffisher – Fisher (central and non-central) idf

### Calling Sequence

```
X=idffisher(Y,m,n)
X=idffisher(Y,m,n,nc)
```

### Parameters

- **X**, **Y** : real matrices **X** and **Y**. Each entry  $Y_{i,j}$  of **Y** must be in  $]0, 1[$ .
- **n**, **m** : parameters  $m$  and  $n$  of the Fisher distribution. Must be integers  $\geq 1$ .
- **nc** : noncentrality parameter. Must be  $\geq 0$ . Default is 0.

### Description

Compute in matrix **X** the idf of the Fisher ( $m, n$ ) distribution for each entry  $Y_{i,j}$  of matrix **Y**. `idffisher(Y,m,n)` is equivalent to `idffisher(Y,m,n,0)`.

### Examples

- If  $X$  is a Fisher ( $m = 2, n = 3$ ) random variable, compute  $x$  such that  $P(X \leq x) = 0.1$ .  
`x=idffisher(0.1,2,3)`  
`cdffisher(x,2,3)`
- If  $X$  is a Fisher ( $m = 11, n = 9, nc = 4$ ) random variable, compute  $x$  such that  $P(X \geq x) = 0.2$ .  
`x=idffisher(1-0.2,11,9,4)`  
`1-cdffisher(x,11,9,4)`

### See Also

`cdffisher`, `pdffisher`

## 5.7 idfgamma – gamma idf

### Calling Sequence

```
X=idfgamma(Y,a,b=,c=,d=)
```

### Parameters

- **X**, **Y** : real matrices **X** and **Y**. Each entry  $Y_{i,j}$  of **Y** must be in  $]0, 1[$ .
- **a** : parameter  $a > 0$  of the gamma distribution.
- **b** : parameter  $b > 0$  of the gamma distribution. Default is 1.
- **c** : parameter  $c$  of the gamma distribution. Default is 0.
- **d** : parameter  $d \neq 0$  of the gamma distribution. Default is 1.

### Description

Compute in **X** the idf of the gamma ( $a, b, c, d$ ) distribution for each entry  $Y_{i,j}$  of matrix **Y**. The gamma ( $a, b, c, d$ ) distribution is defined on  $[c, +\infty[$ . `idfgamma(Y,a)` is equivalent to `idfgamma(Y,a,1,0,1)`.

### Examples

- If  $X$  is a gamma ( $a = 2, b = 1, c = 0, d = 1$ ) random variable, compute  $x$  such that  $P(X \leq x) = 0.1$ .  
`x=idfgamma(0.1,2)`  
`cdfgamma(x,2)`
- If  $X$  is a gamma ( $a = 5, b = 0.7, c = 2, d = 1.5$ ) random variable, compute  $x$  such that  $P(X \geq x) = 0.2$ .

```
x=idfgamma(1-0.2,5,0.7,2,1.5)
1-cdfgamma(x,5,0.7,2,1.5)
```

See Also

`cdfgamma`, `fitgamma`, `pdfgamma`, `rndgamma`

## 5.8 idfgev – generalized Extreme Value idf

Calling Sequence

```
X=idfgev(Y,a,b=,c=)
```

Parameters

- **X, Y** : real matrices **X** and **Y**. Each entry  $Y_{i,j}$  of **Y** must be in  $]0, 1[$ .
- **a** : parameter  $a$  of the GEV distribution.
- **b** : parameter  $b > 0$  of the GEV distribution. Default is 1.
- **c** : parameter  $c$  of the GEV distribution. Default is 0.

Description

Compute in matrix **X** the idf of the GEV  $(a, b, c)$  distribution for each entry  $Y_{i,j}$  of matrix **Y**. `idfgev(y,a)` is equivalent to `idfgev(y,a,1,0)`.

Examples

- If  $X$  is a GEV  $(a = 0.5, b = 1, c = 0)$  random variable, compute  $x$  such that  $P(X \leq x) = 0.1$ .  

```
x=idfgev(0.1,0.5)
cdfgev(x,0.5)
```
- If  $X$  is a GEV  $(a = -0.5, b = 1, c = 5)$  random variable, compute  $x$  such that  $P(X \geq x) = 0.2$ .  

```
x=idfgev(1-0.2,-0.5,c=5)
1-cdfgev(x,-0.5,c=5)
```

See Also

`cdfgev`, `fitgev`, `pdfgev`, `rndgev`

## 5.9 idfjohnson – Johnson's idf

Calling Sequence

```
X=idfjohnson(Y,s,a,b,c,d)
```

Parameters

- **X, Y** : real matrices **X** and **Y**. Each entry  $Y_{i,j}$  of **Y** must be in  $]0, 1[$ .
- **s** : Johnson's system of distribution. Must be "B" for the Johnson's  $S_B$  (bounded) system of distributions or "U" for the Johnson's  $S_U$  (unbounded) system of distributions.
- **a** : parameter  $a$  of the Johnson's distribution.
- **b** : parameter  $b > 0$  of the Johnson's distribution.
- **c** : parameter  $c$  of the Johnson's distribution.
- **d** : parameter  $d > 0$  of the Johnson's distribution.

Description

Compute in matrix **X** the idf of the Johnson's distribution for each entry  $Y_{i,j}$  of matrix **Y**.

Examples

- If  $X$  is a Johnson's bounded  $(a = 4, b = 3, c = 1, d = 5)$  random variable, compute  $x$  such that  $P(X \leq x) = 0.1$ .  

```
x=idfjohnson(0.1,"B",4,3,1,5)
cdfjohnson(x,"B",4,3,1,5)
```
- If  $X$  is a Johnson's unbounded  $(a = 3, b = 4, c = 5, d = 2)$  random variable, compute  $x$  such that  $P(X \geq x) = 0.2$ .  

```
x=idfjohnson(1-0.2,"U",3,4,5,2)
1-cdfjohnson(x,"U",3,4,5,2)
```

See Also

`cdfjohnson`, `fitjohnson`, `pdfjohnson`, `rndjohnson`

## 5.10 idflognormal – lognormal idf

### Calling Sequence

`X=idflognormal(Y,a=,b=,c=)`

### Parameters

- **X,Y** : real matrices **X** and **Y**. Each entry  $Y_{i,j}$  of **Y** must be in  $]0, 1[$ .
- **a** : parameter  $a$  of the lognormal distribution. Default is 0.
- **b** : parameter  $b > 0$  of the lognormal distribution. Default is 1.
- **c** : parameter  $c$  of the lognormal distribution. Default is 0.

### Description

Compute in **X** the idf of the lognormal distribution for each entry  $Y_{i,j}$  of matrix **Y**. The lognormal distribution is defined on  $[c, +\infty)$ . `idflognormal(y)` is equivalent to `idflognormal(y,0,1,0)`.

### Examples

- If  $X$  is a lognormal ( $a = 0.5, b = 2, c = 0$ ) random variable, compute  $x$  such that  $P(X \leq x) = 0.1$ .  
`x=idflognormal(0.1,0.5,2)`  
`cdflognormal(x,0.5,2)`
- If  $X$  is a lognormal ( $a = -0.5, b = 1, c = 0.5$ ) random variable, compute  $x$  such that  $P(X \geq x) = 0.2$ .  
`x=idflognormal(1-0.2,-0.5,c=0.5)`  
`1-cdflognormal(x,-0.5,c=0.5)`

### See Also

`cdflognormal`, `fitlognormal`, `pdflognormal`, `rndlognormal`

## 5.11 idfmedian – normal sample median idf

### Calling Sequence

`X=idfmedian(Y,n,mu=,sigma=)`

### Parameters

- **X,Y** : real matrices **X** and **Y**. Each entry  $Y_{i,j}$  of **Y** must be in  $]0, 1[$ .
- **n** : parameter  $n$  of the normal  $(\mu, \sigma)$  sample median distribution. Must be an odd integer  $\geq 1$ .
- **mu** : parameter  $\mu$  (mean) of the normal distribution. Default is 0.
- **sigma** : parameter  $\sigma > 0$  (standard-deviation) of the normal distribution. Default is 1.

### Description

Compute in matrix **X** the idf of the normal  $(\mu, \sigma)$  sample median distribution for each entry  $Y_{i,j}$  of matrix **Y**.

### Examples

- If  $X$  is a normal sample median ( $n = 3, \mu = 0, \sigma = 1$ ) random variable, compute  $x$  such that  $P(X \leq x) = 0.05$ .  
`x=idfmedian(0.05,3)`  
`cdfmedian(x,3)`
- If  $X$  is a normal sample median ( $n = 5, \mu = 1, \sigma = 0.4$ ) random variable, compute  $x$  such that  $P(X \geq x) = 0.01$ .  
`x=idfmedian(1-0.01,5,1,0.4)`  
`1-cdfmedian(x,5,1,0.4)`

### See Also

`cdfmedian`, `pdfmedian`

## 5.12 idfnormal – normal idf

### Calling Sequence

`X=idfnormal(Y,mu=,sigma=)`

### Parameters

- **X,Y** : real matrices **X** and **Y**. Each entry  $Y_{i,j}$  of **Y** must be in  $]0, 1[$ .
- **mu** : parameter  $\mu$  (mean) of the normal distribution. Default is 0.

- **sigma** : parameter  $\sigma > 0$  (standard-deviation) of the normal distribution. Default is 1.

#### Description

Compute in **X** the idf of the normal distribution for each entry  $Y_{i,j}$  of matrix **Y**. `idfnormal(Y)` is equivalent to `idfnormal(Y,0,1)`.

#### Examples

- If  $X$  is a normal ( $\mu = 0, \sigma = 1$ ) random variable, compute  $x$  such that  $P(X \leq x) = 0.05$ .  
`x=idfnormal(0.05)`  
`cdfnormal(x)`
- If  $X$  is a normal ( $\mu = 3, \sigma = 2$ ) random variable, compute  $x$  such that  $P(X \geq x) = 0.01$ .  
`x=idfnormal(1-0.01,3,2)`  
`1-cdfnormal(x,3,2)`

#### See Also

`cdfnormal`, `pdfnormal`, `rndnormal`

### 5.13 idfpareto – Pareto idf

#### Calling Sequence

`X=idfpareto(Y,a,b=,c=)`

#### Parameters

- **X,Y** : real matrices **X** and **Y**.
- **a** : parameter  $a$  of the Pareto distribution.
- **b** : parameter  $b > 0$  of the Pareto distribution. Default is 1.
- **c** : parameter  $c$  of the Pareto distribution. Default is 0.

#### Description

Compute in matrix **X** the idf of the Pareto distribution for each entry  $Y_{i,j}$  of matrix **Y**. The Pareto distribution is defined on

- $[c, +\infty)$  if  $a \geq 0$ ,
- $[c, c - b/a]$  if  $a < 0$ .

`idfpareto(Y,a)` is equivalent to `idfpareto(Y,a,1,0)`.

#### Examples

- If  $X$  is a Pareto ( $a = 0.5, b = 1, c = 0$ ) random variable, compute  $x$  such that  $P(X \leq x) = 0.1$ .  
`x=idfpareto(0.1,0.5)`  
`cdfpareto(x,0.5)`
- If  $X$  is a Pareto ( $a = -2, b = 6, c = 1$ ) random variable, compute  $x$  such that  $P(X \geq x) = 0.2$ .  
`x=idfpareto(1-0.2,-2,6,1)`  
`1-cdfpareto(x,-2,6,1)`

#### See Also

`cdfpareto`, `pdfpareto`, `rndpareto`

### 5.14 idfstandev – normal sample standard-deviation idf

#### Calling Sequence

`X=idfstandev(Y,n)`  
`X=idfstandev(Y,n,sigma)`

#### Parameters

- **X,Y** : real matrices **X** and **Y**. Each entry  $Y_{i,j}$  of **Y** must be in  $]0, 1[$ .
- **n** : parameter  $n$  of the normal sample standard-deviation distribution. Must be an integer  $\geq 2$ .
- **sigma** : parameter  $\sigma > 0$  (standard-deviation) of the normal distribution. Default is 1.

#### Description

Compute in matrix **X** the idf of the normal sample standard-deviation distribution for each entry  $Y_{i,j}$  of matrix **Y**. `idfstandev(Y,n)` is equivalent to `idfstandev(Y,n,1)`.

#### Examples

- If  $S$  is a normal sample standard-deviation ( $n = 3, \sigma = 1$ ) random variable, compute  $s$  such that  $P(S \leq s) = 0.05$ .  
`s=idfstandev(0.05,3)`  
`cdfstandev(s,3)`
- If  $S$  is a normal sample standard-deviation ( $n = 9, \sigma = 3.5$ ) random variable, compute  $s$  such that  $P(S \geq s) = 0.01$ .  
`s=idfstandev(1-0.01,9,3.5)`  
`1-cdfstandev(s,9,3.5)`

See Also

`cdfstandev`, `pdfstandev`, `rndstandev`

## 5.15 idfstudent – Student (central and non-central) idf

Calling Sequence

```
X=idfstudent(Y,n)
X=idfstudent(Y,n,nc)
```

Parameters

- $\mathbf{X}, \mathbf{Y}$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ . Each entry  $Y_{i,j}$  of  $\mathbf{Y}$  must be in  $]0, 1[$ .
- $n$  : parameter  $n$  of the Student distribution. Must be an integer  $\geq 1$ .
- $nc$  : noncentrality parameter. Must be  $\geq 0$ . Default is 0.

Description

Compute in matrix  $\mathbf{X}$  the idf of the Student distribution for each entry  $Y_{i,j}$  of matrix  $\mathbf{Y}$ . `idfstudent(Y,n)` is equivalent to `idfstudent(Y,n,0)`.

Examples

- If  $X$  is a Student ( $n = 2$ ) random variable, compute  $x$  such that  $P(X \leq x) = 0.1$ .  
`x=idfstudent(0.1,2)`  
`cdfstudent(x,2)`
- If  $X$  is a Student ( $n = 20, nc = 3$ ) random variable, compute  $x$  such that  $P(X \geq x) = 0.2$ .  
`x=idfstudent(1-0.2,20,3)`  
`1-cdfstudent(x,20,3)`

See Also

`cdfstudent`, `pdfstudent`

## 5.16 idfweibull – Weibull idf

Calling Sequence

```
X=idfweibull(Y,a,b=,c=)
```

Parameters

- $\mathbf{X}, \mathbf{Y}$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ . Each entry  $Y_{i,j}$  of  $\mathbf{Y}$  must be in  $]0, 1[$ .
- $a$  : parameter  $a > 0$  of the Weibull distribution.
- $b$  : parameter  $b > 0$  of the Weibull distribution. Default is 1.
- $c$  : parameter  $c$  of the Weibull distribution. Default is 0.

Description

Compute in  $\mathbf{X}$  the idf of the Weibull distribution for each entry  $Y_{i,j}$  of matrix  $\mathbf{Y}$ . The Weibull distribution is defined on  $[c, +\infty)$ . `idfweibull(Y,a)` is equivalent to `idfweibull(Y,a,1,0)`.

Examples

- If  $X$  is a Weibull ( $a = 2, b = 1, c = 0$ ) random variable, compute  $x$  such that  $P(X \leq x) = 0.1$ .  
`x=idfweibull(0.1,2)`  
`cdfweibull(x,2)`
- If  $X$  is a Weibull ( $a = 5, b = 3, c = 2$ ) random variable, compute  $x$  such that  $P(X \geq x) = 0.2$ .  
`x=idfweibull(1-0.2,5,3,2)`  
`1-cdfweibull(x,5,3,2)`

See Also

`cdfweibull`, `fitweibull`, `pdfweibull`, `rndweibull`



## 6 MISCELANEOUS

### 6.1 allcombination – matrix element combinations

#### Calling Sequence

```
C=allcombination(p,X)
```

#### Parameters

- $p$  : an integer  $p$  that must satisfy  $1 \leq p \leq n$ .
- $X$  : a real matrix  $X$  of length  $n$ .
- $C$  : a matrix  $C$ .

#### Description

Compute in matrix  $C$  the  $C_n^p$  combinations of  $p$  elements among the  $n$  elements of matrix  $X$ .

#### Examples

```
allcombination(3,5:9)
allcombination(4,[0;2;4;6;8;10])
```

#### See Also

```
allpermutation
```

### 6.2 allpermutation – matrix element permutations

#### Calling Sequence

```
P=allpermutation(X)
```

#### Parameters

- $X$  : a real matrix  $X$  of length  $n$ .
- $P$  : a matrix  $P$ .

#### Description

Compute in matrix  $P$  the  $n!$  permutations of the  $n$  elements of matrix  $X$ .

#### Examples

```
allpermutation(5:7)
allpermutation([2;4;6;8;10])
```

#### See Also

```
allcombination
```

### 6.3 arrangement – number $A_n^p$ of arrangements

#### Calling Sequence

```
A=arrangement(N,P)
```

#### Parameters

- $N$  : matrix  $N$  of integers  $\geq 1$ .
- $P$  : matrix  $P$  of integers of the same size as  $N$ . Each element  $p_{i,j}$  of  $P$  must verify  $0 \leq p_{i,j} \leq n_{i,j}$ .
- $A$  : matrix  $A$  of integers.

#### Description

Compute in matrix  $A$  the arrangements  $A_n^p$  for each elements of matrices  $N$  and  $P$ .

#### Examples

```
A=arrangement(10*ones(1,10+1)',(0:10)');
mprintf("A(10,%2d) = %d\n",[(0:10)',A])
```

#### See Also

```
combination
```

## 6.4 combination – number $C_n^p$ of combinations

### Calling Sequence

`C=combination(N,P)`

### Parameters

- **N** : matrix **N** of integers  $\geq 1$ .
- **P** : matrix **P** of integers of the same size as **N**. Each element  $p_{i,j}$  of **P** must verify  $0 \leq p_{i,j} \leq n_{i,j}$ .
- **C** : matrix **C** of integers.

### Description

Compute in matrix **C** the combinations  $C_n^p$  for each elements of matrices **N** and **P**.

### Examples

```
C=combination(10*ones(1,10+1)',(0:10)');  
mprintf("C(10,%2d) = %d\n",[0:10]',C)
```

### See Also

`arrangement`

## 6.5 confhyper – confluent hypergeometric function

### Calling Sequence

`Y=confhyper(X,a,b)`

### Parameters

- **X** : matrix **X**.
- **a,b** : parameters  $a$  and  $b$ .
- **Y** : matrix **Y**.

### Description

Compute in matrix **Y** the confluent hypergeometric function for each element of matrix **Y**, i.e.

$$Y_{i,j} = {}_1F_1(a, b, X_{i,j}) = 1 + \frac{a}{b} \frac{X_{i,j}}{1!} + \frac{a(a+1)}{b(b+1)} \frac{X_{i,j}^2}{2!} + \frac{a(a+1)(a+2)}{b(b+1)(b+2)} \frac{X_{i,j}^3}{3!} + \dots$$

### Example

```
confhyper([0.1,0.2,0.3,0.4,0.5],2,3)
```

## 6.6 depth - non parametric multivariate depth

### Calling Sequence

```
[d,id]=depth(X,dep)  
[d,id]=depth(X,dep,Y)
```

### Parameters

- **X** : real matrix **X** of size  $(n_X, p)$ .
- **Y** : real matrix **Y** of size  $(n_Y, p)$ . Default is **X**.
- **dep** : multivariate definition of depth in  $\mathbb{R}^p$ . Must be "halfspace", "majority" or "simplicial".
- **d** : column vector **d** of depths.
- **id** : vector of indices corresponding to the depths sorted in descending order.

### Description

Compute in vector **d** the depths for each row vector of matrix **Y** based on row vectors in matrix **X**. The indices corresponding to the depths sorted in descending order are in **id**, i.e. `X(id1(1),:)` is a potential multivariate median. `[d,id]=depth(X,dep)` is equivalent to `[d,id]=depth(X,dep,X)`.

### Examples (see Figure 2)

```

X=rndmultinormal(100,[0,0],[2,1.9;1.9,5]);
[d1,id1]=depth(X,"halfspace");
[d2,id2]=depth(X,"majority");
[d3,id3]=depth(X,"simplicial");
//
xset("window",0);xbasc()
plot2d(X(id1(1:20),1),X(id1(1:20),2),-4)
plot2d(X(id1(21:100),1),X(id1(21:100),2),-5)
xtitle("HALFSPACE MULTIVARIATE DEPTH");xselect()
//
xset("window",1);xbasc()
plot2d(X(id2(1:20),1),X(id2(1:20),2),-4)
plot2d(X(id2(21:100),1),X(id2(21:100),2),-5)
xtitle("MAJORITY MULTIVARIATE DEPTH");xselect()
//
xset("window",2);xbasc()
plot2d(X(id3(1:20),1),X(id3(1:20),2),-4)
plot2d(X(id3(21:100),1),X(id3(21:100),2),-5)
xtitle("SIMPLICIAL MULTIVARIATE DEPTH");xselect()

```

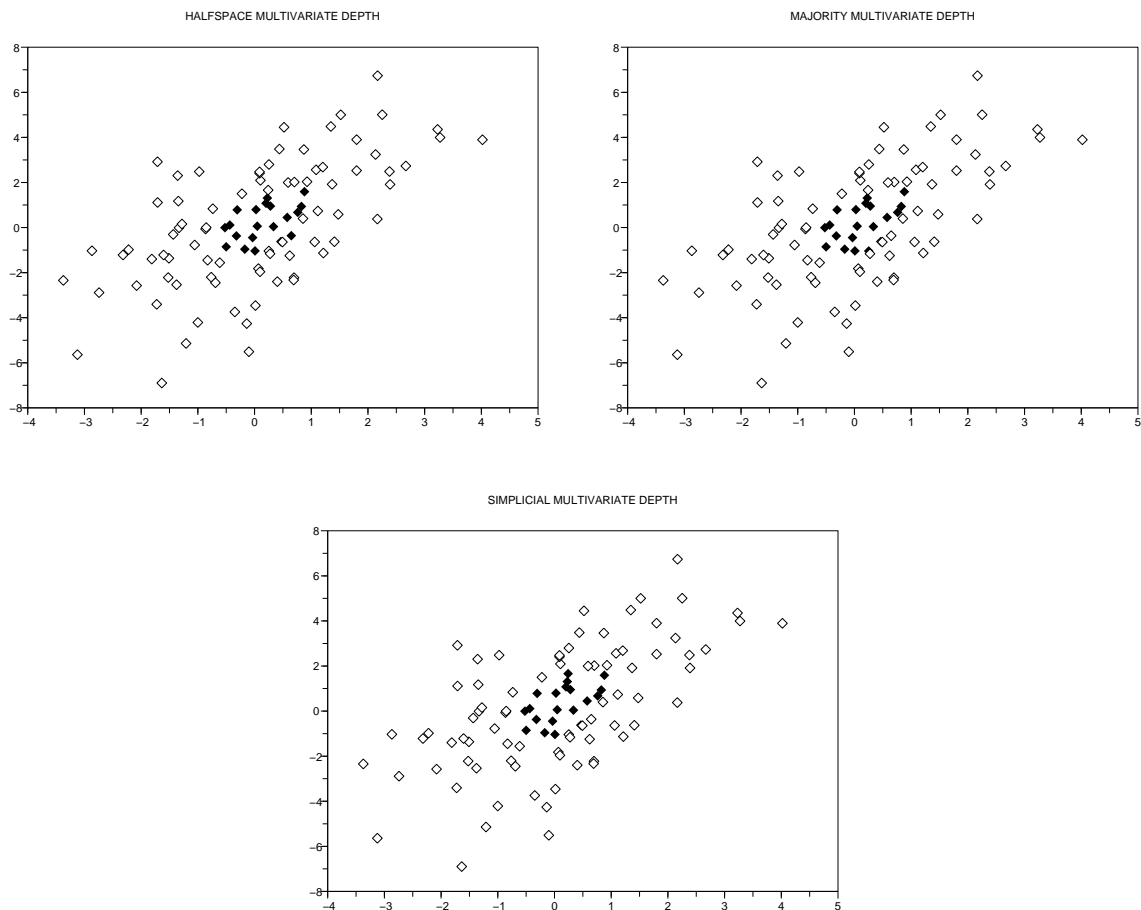


Figure 2: Example of function depth

## 6.7 hausdorff – Hausdorff (median) distance between polylines

### Calling Sequence

```
h=hausdorff(X,Y)
```

### Parameters

- $\mathbf{X}$  : real matrix  $\mathbf{X}$  of size  $(n_X, p)$ .
- $\mathbf{Y}$  : real matrix  $\mathbf{Y}$  of size  $(n_Y, p)$ .
- $h$  : Hausdorff (median) distance between polylines  $\mathbf{X}$  and  $\mathbf{Y}$ .

### Description

Compute the Hausdorff (median) distance between polylines  $\mathbf{X}$  and  $\mathbf{Y}$  using the Euclidean distance.

### Examples

```
X=linspace(-%pi,%pi)';
Y1=sin(X)+rndnormal(100,sigma=0.1);
Y2=sin(X)+rndnormal(100,sigma=0.1);
Y3=sin(2*X)+rndnormal(100,sigma=0.1);
hausdorff([X,Y1],[X,Y2])
hausdorff([X,Y1],[X,Y3])
```

## 6.8 lowess – LOcally WEighted Scatterplot Smoothing

### Calling Sequence

```
Y=lowess(X,X0,Y0,h)
```

### Parameters

- $\mathbf{X}$  : real matrix  $\mathbf{X}$  of size  $(n, p)$ .
- $\mathbf{X}_0$  : real matrix  $\mathbf{X}_0$  of size  $(n_0, p)$ .
- $\mathbf{Y}_0$  : real vector  $\mathbf{Y}_0$  of size  $(n_0, 1)$ .
- $h$  : smooting parameter  $h \in (0, 1]$ .
- $\mathbf{Y}$  : real vector  $\mathbf{Y}$  of size  $(n, 1)$ .

### Description

For each entry (row) of matrix  $\mathbf{X}$ , compute in vector  $\mathbf{Y}$  the LOWESS (linear) regression estimates based on data in matrix  $\mathbf{X}_0$  and vector  $\mathbf{Y}_0$ . The smooting parameter  $h \in (0, 1]$  allows to select the % of the nearest points used for the LOWESS regression (i.e.  $h = 0.2$  means that the 20% nearest points are used). The weight function used for LOWESS regression is the tri-cube weight function.

### Examples (see Figure 3)

```
X0=linspace(-%pi,%pi)';
Y0=sin(X0)+rndnormal(100,0,0.2);
X=linspace(-%pi,%pi)';
Y2=lowess(X,X0,Y0,0.2);
Y4=lowess(X,X0,Y0,0.4);
Y6=lowess(X,X0,Y0,0.6);
Y8=lowess(X,X0,Y0,0.8);
xset("window",0);xbasc(0)
plot2d(X0,Y0,-1);plot2d(X,Y2,2)
xlabel("h=0.2");xselect()
xset("window",1);xbasc(1)
plot2d(X0,Y0,-1);plot2d(X,Y4,2)
xlabel("h=0.4");xselect()
xset("window",2);xbasc(2)
plot2d(X0,Y0,-1);plot2d(X,Y6,2)
xlabel("h=0.6");xselect()
xset("window",3);xbasc(3)
plot2d(X0,Y0,-1);plot2d(X,Y8,2)
xlabel("h=0.8");xselect()
```

## 6.9 momdphase – first moments of a Discrete Phase-Type distribution

### Calling Sequence

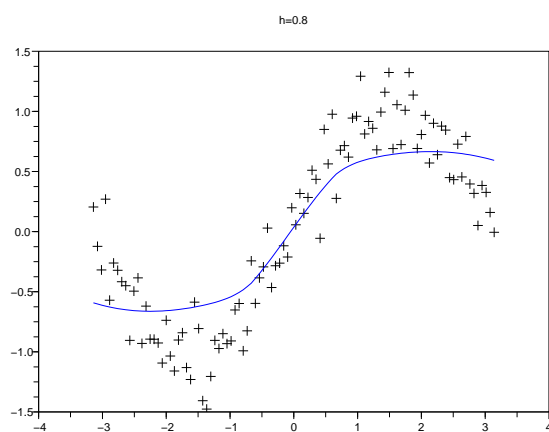
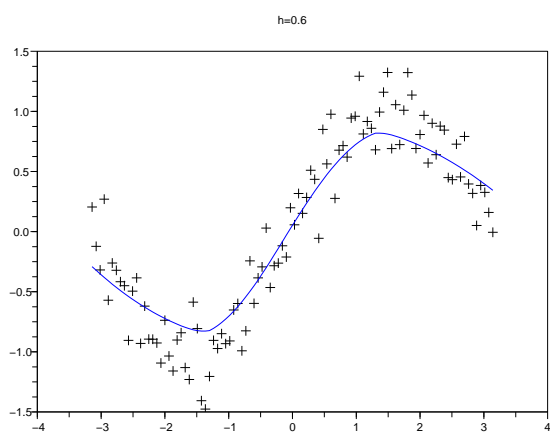
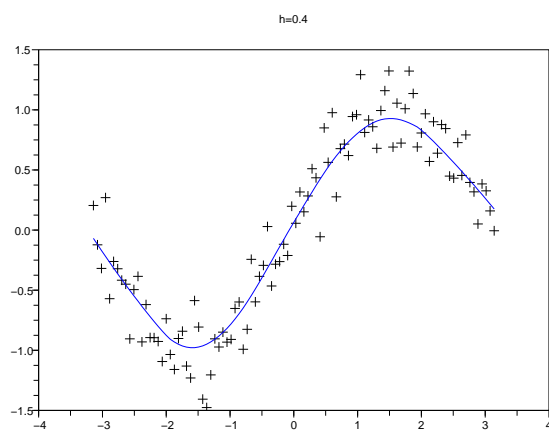
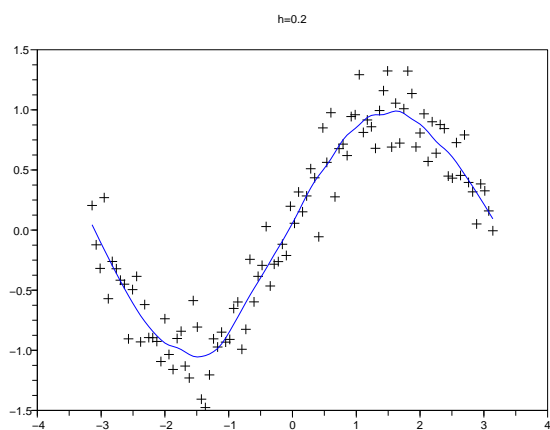


Figure 3: Example of function `lowess`

```
[mu]=momdphase(Q,q)
[mu,sd]=momdphase(Q,q)
[mu,sd,sk]=momdphase(Q,q)
[mu,sd,sk,ku]=momdphase(Q,q)
```

#### Parameters

- **Q** : square matrix **Q** of transient probabilities.
- **q** : vector **q** of initial transient probabilities.
- **mu** : mean  $\mu$  of the Discrete Phase-Type (**Q**, **q**) distribution.
- **sd** : standard-deviation  $\sigma$  of the Discrete Phase-Type (**Q**, **q**) distribution.
- **sk** : skewness coefficient  $\gamma_3$  of the Discrete Phase-Type (**Q**, **q**) distribution.
- **ku** : kurtosis coefficient  $\gamma_4$  of the Discrete Phase-Type (**Q**, **q**) distribution.

#### Description

Compute the mean  $\mu$ , standard-deviation  $\sigma$ , skewness coefficient  $\gamma_3$  and kurtosis coefficient  $\gamma_4$  of the Discrete Phase-Type (**Q**, **q**) distribution. The Discrete Phase-Type distribution is defined on  $\{1, 2, 3, \dots\}$ .

#### Examples

```
Q=[0.6,0.3;0.2,0.5];
q=[1;0];
[mu,sd,sk,ku]=momdphase(Q,q); [mu,sd,sk,ku]
x=(1:1000)';
[sum(x.*pdfdphase(x,Q,q)),...
sqrt(sum(((x-mu).^2).*pdfdphase(x,Q,q))),...
sum((((x-mu)/sd).^3).*pdfdphase(x,Q,q)),...
sum((((x-mu)/sd).^4).*pdfdphase(x,Q,q))-3]
```

#### See Also

cdfdphase, pdfdphase

## 6.10 nearestneighbors – find the $k$ nearest neighbors

#### Calling Sequence

```
i=nearestneighbors(k,x,Y)
i=nearestneighbors(k,x,Y,dis)
```

#### Parameters

- **k** : the number  $k$  of nearest neighbors
- **x** : real row vector **x** of size  $(1, p)$ .
- **Y** : real matrix **Y** of size  $(n, p)$ .
- **dis** : distance used for finding the  $k$  nearest neighbors. Must be "L1", "L2" or "Linf". Default is "L2".
- **i** : indices of the  $k$  nearest neighbors of **x** in **Y**.

#### Description

Find the indices of the  $k$  nearest neighbors of **x** in **Y**. `i=nearestneighbors(k,x,Y)` is equivalent to `i=nearestneighbors(k,x,Y,"L2")`.

#### Examples (see Figure 4)

```
Y=randmultinormal(100,[0,0]);
i=nearestneighbors(7,[0,0],Y)
xset("window",0);xbasc(0)
plot2d(Y(:,1),Y(:,2),-5);plot2d(Y(i,1),Y(i,2),-4)
xlabel("7 nearest neighbors of (0,0), L2 norm")
xselect()
//
i=nearestneighbors(7,[0,0],Y,"L1")
xset("window",1);xbasc(1)
plot2d(Y(:,1),Y(:,2),-5);plot2d(Y(i,1),Y(i,2),-4)
xlabel("7 nearest neighbors of (0,0), L1 norm")
xselect()
```

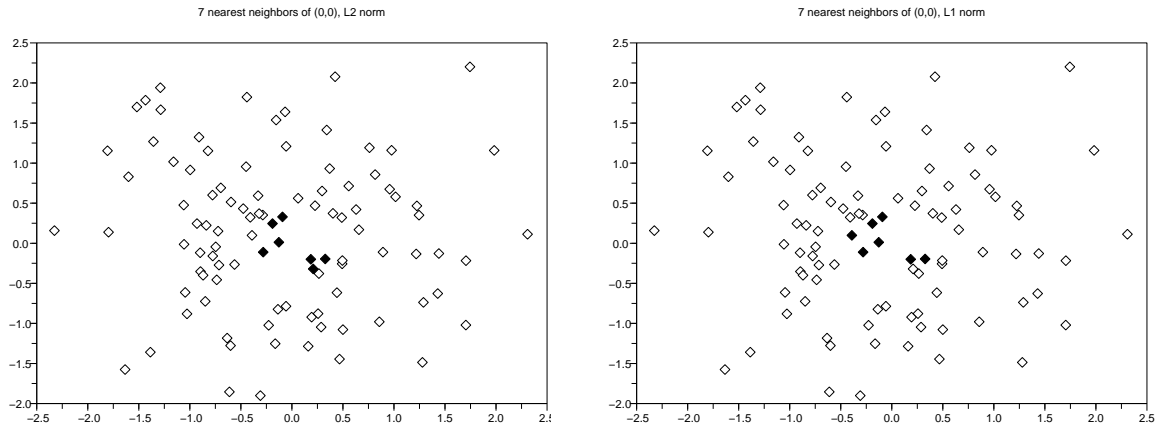


Figure 4: Example of function `nearestneighbors`

## 6.11 `neldermead` – Nelder Mead’s downhill simplex nonlinear optimization algorithm

### Calling Sequence

```
[xopt,fopt]=neldermead(x0,fun,extra=,tol=,opt=)
```

### Parameters

- `x0` : real row vector  $\mathbf{x}_0$  (initial value).
- `fun` : function  $f$  from  $\mathbb{R}^p \rightarrow \mathbb{R}$  to be optimized.
- `extra` : list containing the extra input arguments of function  $f$ . Default is `list()`.
- `tol` : tolerance to be reached. Default is `1e-12`.
- `opt` : optimization flag. Must be "min" for minimization and "max" for maximization. Default is "max".
- `xopt` : real row vector  $\mathbf{x}^*$  that optimizes function  $f$ .
- `fopt` : optimum value for function  $f$  at  $\mathbf{x}^*$ .

### Description

Search the vector  $\mathbf{x}^*$  that optimizes function  $f$  using the Nelder Mead’s downhill simplex nonlinear optimization algorithm.

**Example #1** Maximize function  $f(x_1, x_2) = 80 + 2x_1 + 3x_2 - 1.5x_1^2 - 2x_2^2 - x_1x_2$ .

```
function f=fun1(x)
    f=80+2*x(1)+3*x(2)-1.5*x(1)^2-2*x(2)^2-x(1)*x(2)
endfunction
//
//Maximize
[xmax,fmax]=neldermead([0,0],fun1)
```

**Example #2** Minimize function  $f(x_1, x_2) = -80 - 2x_1 - 3x_2 + 1.5x_1^2 + 2x_2^2 + x_1x_2$  subject to  $|x_1| \leq 1$  and  $|x_2| \leq 1$ .

```
function f=fun2(x,a)
    if or(abs(x)>1)
        f=%inf
    else
        f=a(1)+a(2)*x(1)+a(3)*x(2)+a(4)*x(1)^2+a(5)*x(2)^2+a(6)*x(1)*x(2)
    end
endfunction
//
//Minimize
[xmin,fmin]=neldermead([0,0],fun2,list([-80,-2,-3,1.5,2,1]),tol=1e-8,opt="min")
```

### See Also

`torczon`

## 6.12 savitzkygolay – Savitzky-Golay smoothing filter

### Calling Sequence

```
Y=savitzkygolay(X,p,nL)
Y=savitzkygolay(X,p,nL,nR)
```

### Parameters

- $\mathbf{X}, \mathbf{Y}$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .
- $p$  : degree  $p$  of the polynomial involved in the smoothing procedure.
- $nL$  : number  $n_L$  of points used “to the left” of a data point.
- $nR$  : number  $n_R$  of points used “to the right” of a data point. Default is  $nR=nL$ .

### Description

Compute in matrix  $\mathbf{Y}$  a smoothed version of data in matrix  $\mathbf{X}$  using the Savitzky-Golay smoothing filter. `savitzkygolay(X,p,nL)` is equivalent to `savitzkygolay(X,p,nL,nL)`.

### Examples (see Figure 5)

```
X=linspace(-%pi,%pi)';
Y=sin(X)+rndnormal(100,0,0.1);
Z1=savitzkygolay(Y,2,20,30);
xset("window",0);xbasc(0)
plot2d([X,X],[Y,Z1],[1,5])
xtitle("Savitzky-Golay smoothing filter p=2, nL=20, nR=30")
xselect()
//
Z2=savitzkygolay(Y,4,15);
xset("window",1);xbasc(1)
plot2d([X,X],[Y,Z2],[1,5])
xtitle("Savitzky-Golay smoothing filter p=4, nL=15, nR=15")
xselect()
```

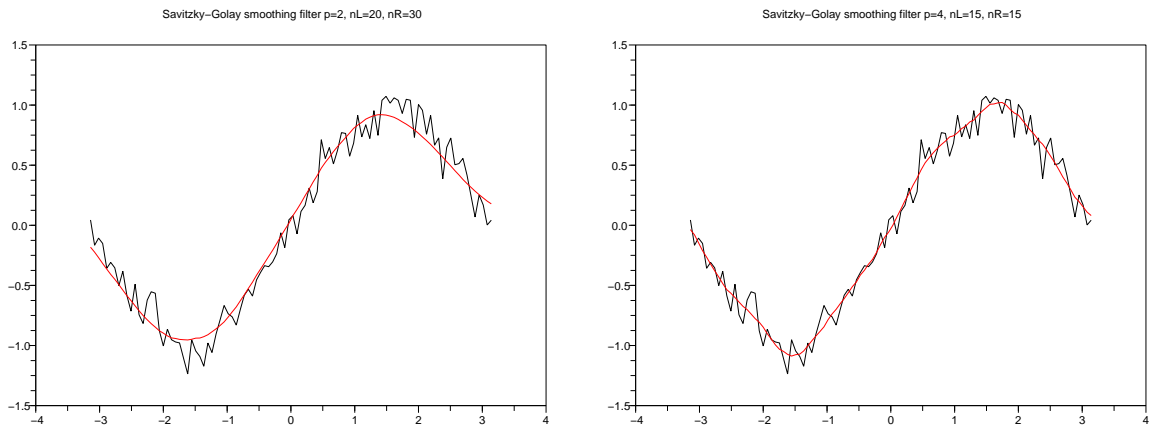


Figure 5: Example of function `savitzkygolay`

## 6.13 simplex – simplex computation

### Calling Sequence

```
X=simplex(x0)
X=simplex(x0,r)
```

### Parameters

- $\mathbf{x}_0$  : centroid  $\mathbf{x}_0$  of the simplex. Must be a row vector of size  $(1, p)$ .
- $r$  : radius  $r > 0$  of the simplex. Default is 1.
- $\mathbf{X}$  : simplex  $\mathbf{X}$  of size  $(p+1, p)$ .



## Description

Compute the simplex  $\mathbf{X}$  of radius  $r$  centered in  $\mathbf{x}_0$ . `simplex(x0)` is equivalent to `simplex(x0,1)`.

**Example** (see Figure 6)

```
X1=simplex([0,0])
X2=simplex([2,3])
X3=simplex([-2,-3],2)
X1($+1,:)=X1(1,:);
X2($+1,:)=X2(1,:);
X3($+1,:)=X3(1,:);
xbasc()
plot2d([X1(:,1),X2(:,1),X3(:,1)], [X1(:,2),X2(:,2),X3(:,2)],...
        [1,2,3],rect=[-5,-5,5,5])
xselect()
```

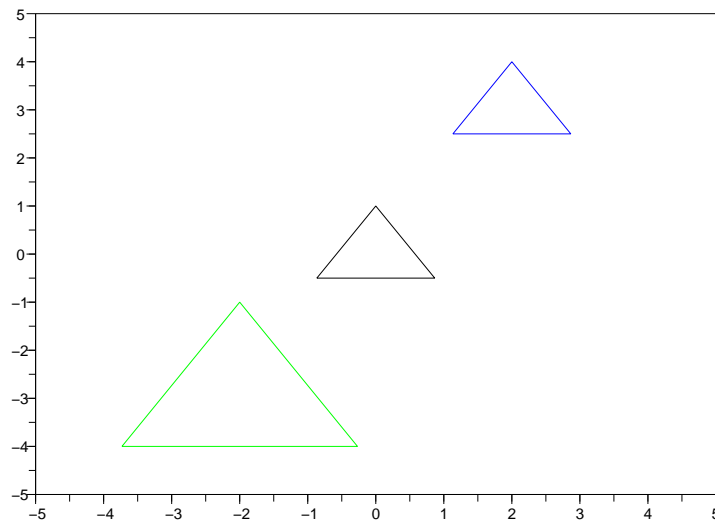


Figure 6: Example of function `simplex`

## 6.14 simplexolve – solve a system of non-linear equations

### Calling Sequence

```
xsol=simplexolve(x0,fun,extra=,tol=)
```

### Parameters

- `x0` : real row vector  $\mathbf{x}_0$  (initial value).
- `fun` : function  $f$  from  $\mathbb{R}^p \rightarrow \mathbb{R}^m$ .
- `extra` : list containing the extra input arguments of function  $f$ . Default is `list()`.
- `tol` : tolerance to be reached. Default is `1e-12`.
- `xsol` : real row vector  $\mathbf{x}^*$  that solves the system of non-linear equations, i.e.  $f(\mathbf{x}^*) = \mathbf{0}$ .

### Description

Search the vector  $\mathbf{x}^*$  that solves the system of non-linear equations  $f(\mathbf{x}^*) = \mathbf{0}$  using a downhill simplex algorithm.

**Example #1** Solve the following system of non-linear equations:

$$f(x_1, x_2) = \begin{cases} (x_1 - x_2)^2 - 1 \\ (x_1 + x_2)^2 - 25 \end{cases}$$

```

function f=fun1(x)
x1=x(1)
x2=x(2)
f(1)=(x1-x2)^2-1
f(2)=(x1+x2)^2-25
endfunction
//
//Solve
xsol=simplexolve([0,0],fun1)

```

**Example #2** Solve the same system of non-linear equations but subject to  $x_1 \geq 0$  and  $x_2 \geq 0$ :

```

function f=fun2(x)
x1=x(1)
x2=x(2)
if (x1<0)|(x2<0)
    f=[%inf,%inf]
else
    f(1)=(x1-x2)^2-1
    f(2)=(x1+x2)^2-25
end
endfunction
//
//Solve
xsol=simplexolve([0,0],fun2,tol=1e-8)

```

## 6.15 torczon – Torczon’s multidirectional nonlinear optimization algorithm

### Calling Sequence

```
[xopt,fopt]=torczon(x0,fun,extra=,tol=,opt=)
```

### Parameters

- **x0** : real row vector  $\mathbf{x}_0$  (initial value).
- **fun** : function  $f$  from  $\mathbb{R}^p \rightarrow \mathbb{R}$  to be optimized.
- **extra** : list containing the extra input arguments of function  $f$ . Default is `list()`.
- **tol** : tolerance to be reached. Default is `1e-12`.
- **opt** : optimization flag. Must be "min" for minimization and "max" for maximization. Default is "max".
- **xopt** : real row vector  $\mathbf{x}^*$  that optimizes function  $f$ .
- **fopt** : optimum value for function  $f$  at  $\mathbf{x}^*$ .

### Description

Search the vector  $\mathbf{x}^*$  that optimizes function  $f$  using the Torczon’s multidirectional nonlinear optimization algorithm.

**Example #1** Maximize function  $f(x_1, x_2) = 80 + 2x_1 + 3x_2 - 1.5x_1^2 - 2x_2^2 - x_1x_2$ .

```

function f=fun1(x)
f=80+2*x(1)+3*x(2)-1.5*x(1)^2-2*x(2)^2-x(1)*x(2)
endfunction
//
//Maximize
[xmax,fmax]=torczon([0,0],fun1)

```

**Example #2** Minimize function  $f(x_1, x_2) = -80 - 2x_1 - 3x_2 + 1.5x_1^2 + 2x_2^2 + x_1x_2$  subject to  $|x_1| \leq 1$  and  $|x_2| \leq 1$ .

```

function f=fun2(x,a)
if or(abs(x)>1)
    f=%inf
else
    f=a(1)+a(2)*x(1)+a(3)*x(2)+a(4)*x(1)^2+a(5)*x(2)^2+a(6)*x(1)*x(2)
end
endfunction

```

```
//
//Minimize
[xmin,fmin]=torczon([0,0],fun2,list([-80,-2,-3,1.5,2,1]),tol=1e-8,opt="min")
```

See Also

neldermead

## 6.16 vandercorput – Van der Corput's sequence

Calling Sequence

```
v=vandercorput(n,b)
```

Parameters

- **v** : a real vector **v**.
- **n** : length  $n$  of the Van der Corput's sequence. The value of  $n$  must be an integer  $\geq 1$ .
- **b** : base  $b$  of the Van der Corput's sequence. The value of  $b$  must be an integer  $\geq 2$ .

Description

Compute in vector **v** the  $n$  first values of the base- $b$  Van der Corput's sequence. Usually  $b$  is a prime number, i.e. 2, 3, 5, 7, 11, ...

Examples (see Figure 7)

```
X=vandercorput(100,2);
Y=vandercorput(100,3);
xbasc();plot2d(X,Y,-5)
xtitle("2D VAN DER CORPUT'S SEQUENCE");xselect()
```

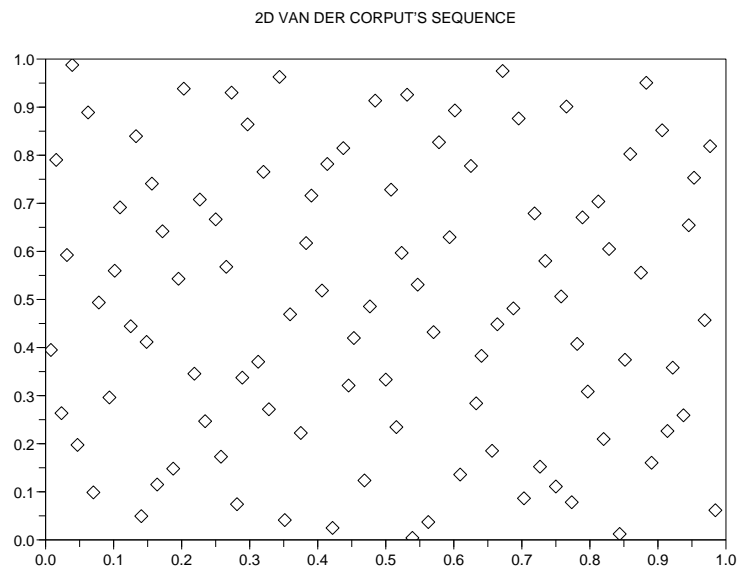


Figure 7: Example of function `vandercorput`

## 7 PLOTS

### 7.1 boxplot – Box plot

Calling Sequence

```
boxplot(X1,X2,...)
```

Parameters

- $X_1, X_2, \dots$  : real matrices  $\mathbf{X}_1, \mathbf{X}_2, \dots$

### Description

Plot a “Box Plot” of data in matrices  $\mathbf{X}_1, \mathbf{X}_2, \dots$

### Examples (see Figure 8)

```
X1=rndnormal(100,5,0.2);
X2=rndnormal(100,5.5,0.3);
X3=rndnormal(100,6,0.1);
// + a strong outlier for X1
X1(101)=6.2;
xbasc();boxplot(X1,X2,X3)
xtitle("BOXPLOT");xselect()
```

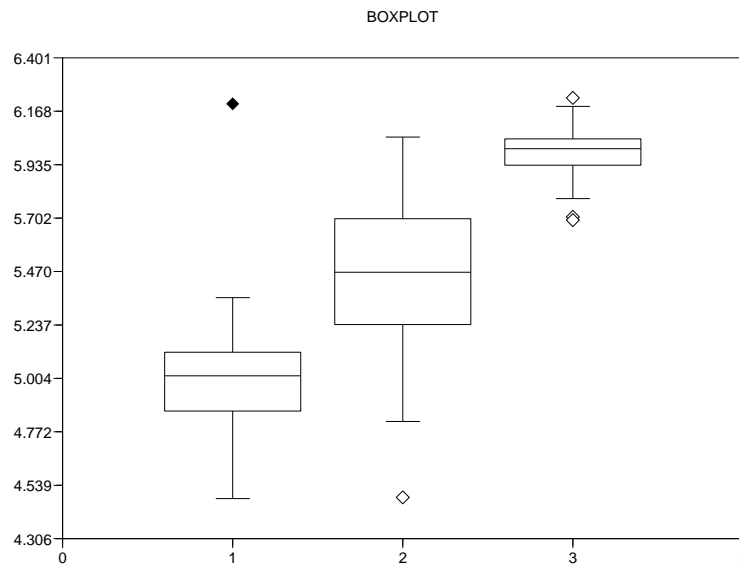


Figure 8: Example of function `boxplot`

## 7.2 qplot – quantile plot

### Calling Sequence

```
qplot(X)
qplot(X,dis)
```

### Parameters

- $X$  : real matrix  $\mathbf{X}$ .
- $dis$  : distribution of the quantile plot. Must be "exponential", "lognormal", "multinormal", "normal" or "weibull". Default is "normal".

### Description

Plot a “quantile plot” of data in matrix  $\mathbf{X}$  corresponding to the selected distribution. `qplot(X)` is equivalent to `qplot(X,"normal")`. If `dis="multinormal"`,  $\mathbf{X}$  must be a  $(n,p)$  matrix.

### Examples (see Figure 9)

```
X=rndnormal(100,3,0.1);
xset("window",0)
xbasc()
qplot(X);xtitle("NORMAL QPLOT");xselect()
//
xset("window",1)
```

```

xbasc()
qqplot(X,"exponential");xtitle("EXPONENTIAL QPLOT");xselect()
//
xset("window",2)
xbasc()
qqplot(X,"lognormal");xtitle("LOGNORMAL QPLOT");xselect()
//
xset("window",3)
xbasc()
qqplot(X,"weibull");xtitle("WEIBULL QPLOT");xselect()
//
t=%pi/6;
R=[cos(t),-sin(t);sin(t),cos(t)];
V=diag([0.1,0.4]);
sigma=R*V*R';
mu=[5,5];
X=rndmultinormal(100,mu,sigma);
xset("window",4)
xbasc()
qqplot(X,"multinormal");xtitle("MULTINORMAL QPLOT");xselect()

```

See Also

qqplot

### 7.3 qqplot – quantile-quantile plot

Calling Sequence

```
qqplot(X,Y)
```

Parameters

–  $X, Y$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .

Description

Plot a “quantile-quantile plot” of data in matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .

Examples (see Figure 10)

```

X=rndnormal(100,3,0.1);
Y=rndnormal(100,3,0.1);
xset("window",0);xbasc(0)
qqplot(X,Y)
xtitle("QQPLOT X and Y");xselect()
//
Z=rndlognormal(100,2,3);
xset("window",1);xbasc(1)
qqplot(X,Z)
xtitle("QQPLOT X and Z");xselect()

```

See Also

qplot

## 8 PROBABILITY DENSITY FUNCTIONS

### 8.1 pdfbeta – beta type 1 pdf

Calling Sequence

```
Y=pdfbeta(X,a,b,c=,d=)
```

Parameters

–  $X, Y$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .

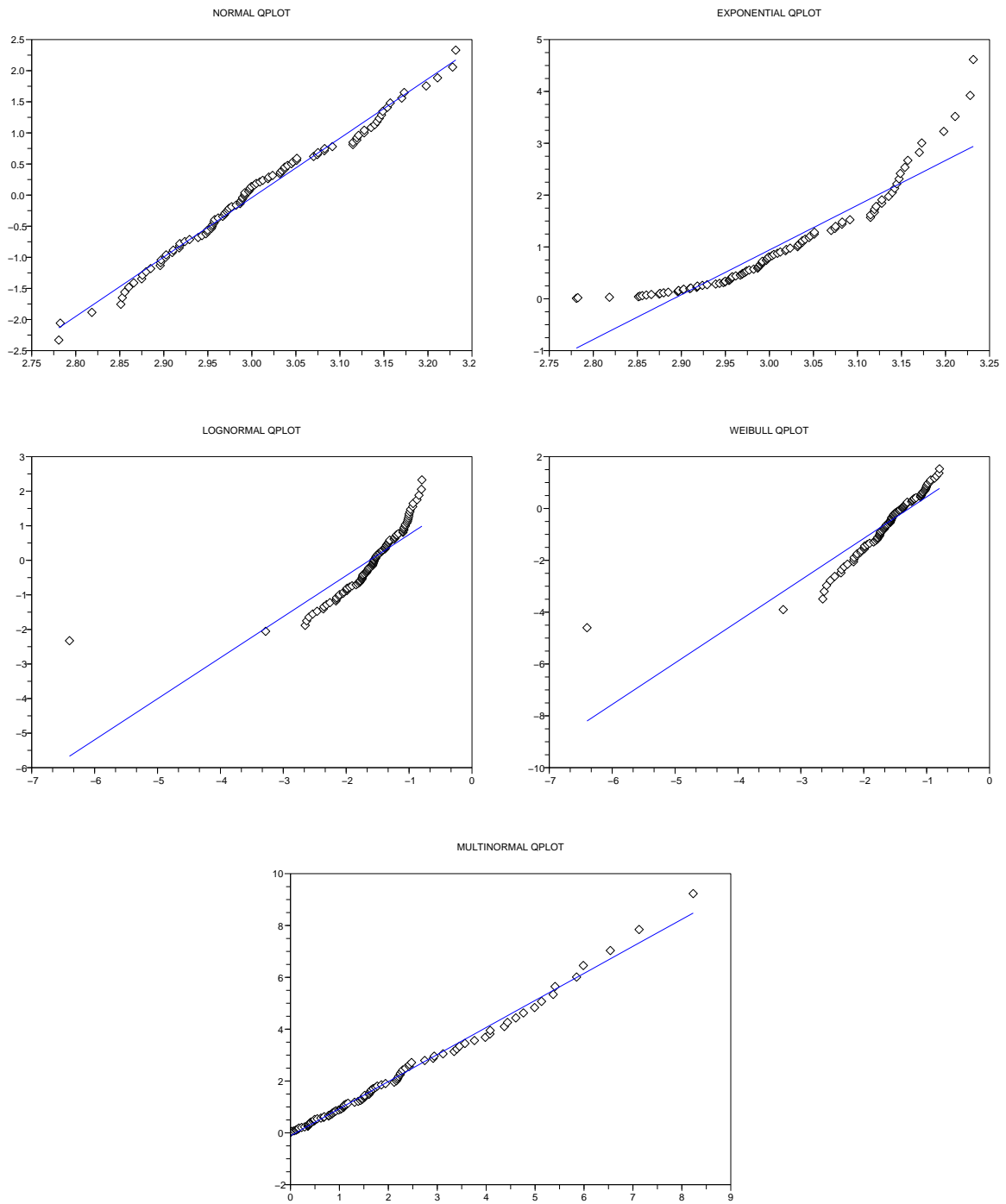


Figure 9: Example of function `qplot`

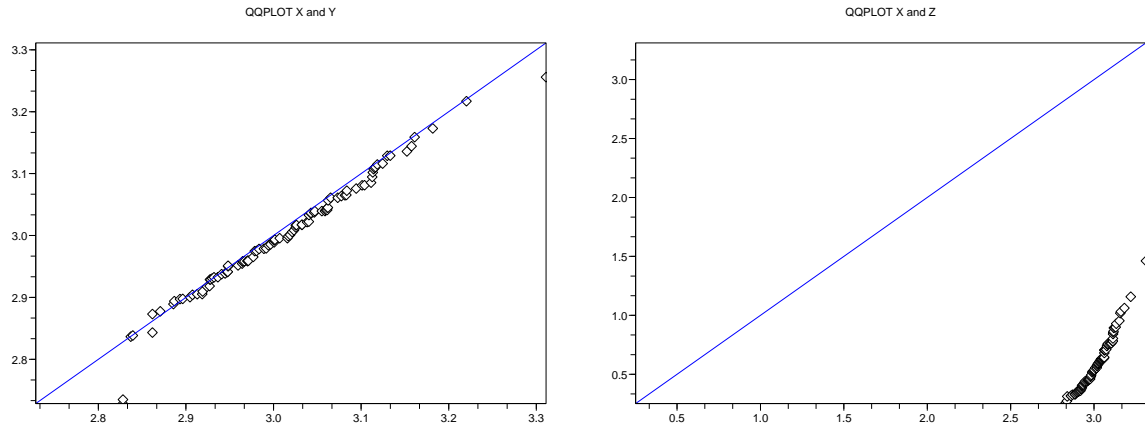


Figure 10: Example of function `qqplot`

- `a` : parameter  $a > 0$  of the beta type 1 distribution.
- `b` : parameter  $b > 0$  of the beta type 1 distribution.
- `c` : parameter  $c$  of the beta type 1 distribution. Default is 0.
- `d` : parameter  $d > 0$  of the beta type 1 distribution. Default is 1.

### Description

Compute in matrix  $\mathbf{Y}$  the pdf of the beta type 1 distribution for each entry  $X_{i,j}$  of matrix  $\mathbf{X}$ . The beta type 1 distribution is defined on  $[c, c + d]$ . `pdfbeta(X,a,b)` is equivalent to `pdfbeta(X,a,b,0,1)`.

**Examples** (see Figure 11)

```
X=linspace(-1,2,300)';
Y1=pdfbeta(X,2,5);
xset("window",0);xbasc(0);plot2d(X,Y1,1,rect=[-1,0,2,2.5])
xtitle("PDF OF THE BETA TYPE 1 DISTRIBUTION a=2, b=5, c=0, d=1")
xselect()
//
Y2=pdfbeta(X,5,2,-0.5,2.5);
xset("window",1);xbasc(1);plot2d(X,Y2,1,rect=[-1,0,2,2.5])
xtitle("PDF OF THE BETA TYPE 1 DISTRIBUTION a=5, b=2, c=-0.5, d=2.5")
xselect()
```

**See Also**

`cdfbeta`, `fitbeta`, `idfbeta`, `rndbeta`

## 8.2 pdfbeta2 – beta type 2 pdf

### Calling Sequence

```
Y=pdfbeta2(X,a,b,c=,d=)
```

### Parameters

- $\mathbf{X}, \mathbf{Y}$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .
- `a` : parameter  $a > 0$  of the beta type 2 distribution.
- `b` : parameter  $b > 0$  of the beta type 2 distribution.
- `c` : parameter  $c$  of the beta type 2 distribution. Default is 0.
- `d` : parameter  $d > 0$  of the beta type 2 distribution. Default is 1.

### Description

Compute in matrix  $\mathbf{Y}$  the pdf of the beta type 2 distribution for each entry  $X_{i,j}$  of matrix  $\mathbf{X}$ . The beta type 2 distribution is defined on  $[c, +\infty)$ . `pdfbeta2(X,a,b)` is equivalent to `pdfbeta2(X,a,b,0,1)`.

**Examples** (see Figure 12)

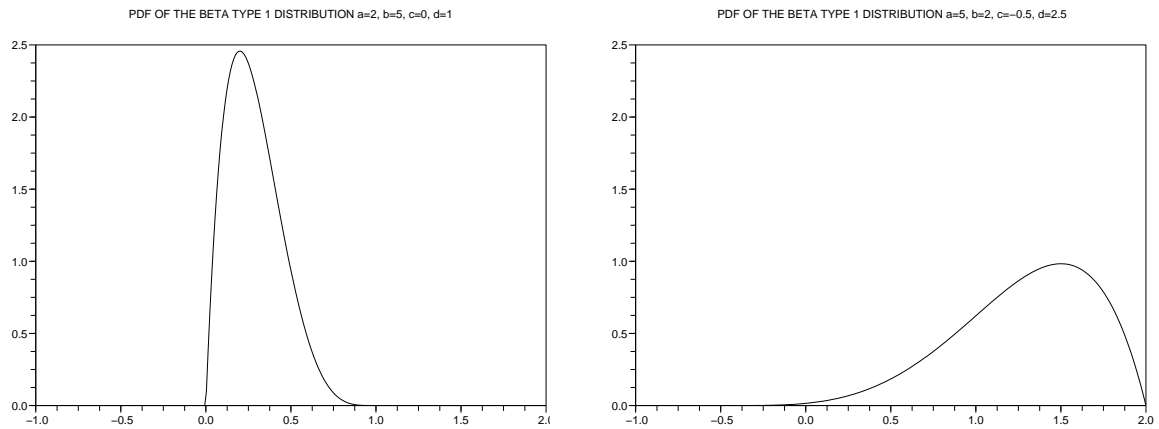


Figure 11: Example of function `pdfbeta`

```
X=linspace(-1,2,300)';
Y1=pdfbeta2(X,2,5);
xset("window",0);xbasc(0);plot2d(X,Y1,1,rect=[-1,0,2,3])
xtitle("PDF OF THE BETA TYPE 2 DISTRIBUTION a=2, b=5, c=0, d=1")
xselect()
//
Y2=pdfbeta2(X,5,2,-0.5,0.1);
xset("window",1);xbasc(1);plot2d(X,Y2,1,rect=[-1,0,2,3])
xtitle("PDF OF THE BETA TYPE 2 DISTRIBUTION a=5, b=2, c=-0.5, d=0.1")
xselect()
```

See Also

`cdfbeta2`, `idfbeta2`, `rndbeta2`

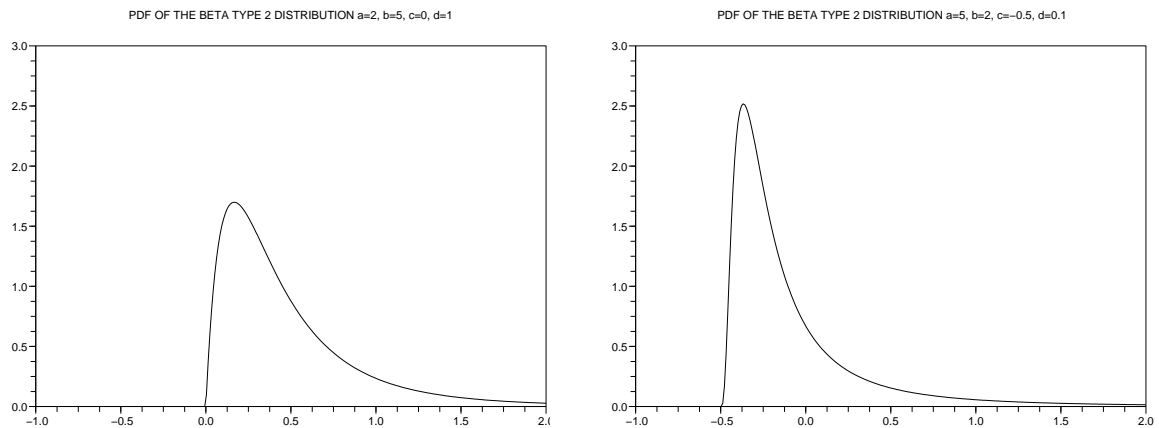


Figure 12: Example of function `pdfbeta2`

### 8.3 `pdfbinomial` – binomial pdf

Calling Sequence

```
Y=pdfbinomial(X,n,p)
```

Parameters

–  $\mathbf{X}, \mathbf{Y}$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .



- **n** : parameter  $n$  of the binomial distribution. Must be an integer  $\geq 1$ .
- **p** : parameter  $p \in [0, 1]$  of the binomial distribution.

### Description

Compute in matrix **Y** the pdf of the binomial distribution for each entry  $X_{i,j}$  of matrix **X**.

**Examples** (see Figure 13)

```
X=(-1:20)';
Y1=pdfbinomial(X,20,0.2);
xset("window",0);xbasc(0);plot2d3(X,Y1,1,rect=[-1,0,20,0.25],nax=[0,22,0,11])
xtitle("PDF OF THE BINOMIAL DISTRIBUTION n=20, p=0.2");xselect()
//
Y2=pdfbinomial(X,20,0.5);
xset("window",1);xbasc(1);plot2d3(X,Y2,1,rect=[-1,0,20,0.25],nax=[0,22,0,11])
xtitle("PDF OF THE BINOMIAL DISTRIBUTION n=20, p=0.5");xselect()
```

**See Also**

`cdfbinomial`, `rndbinomial`

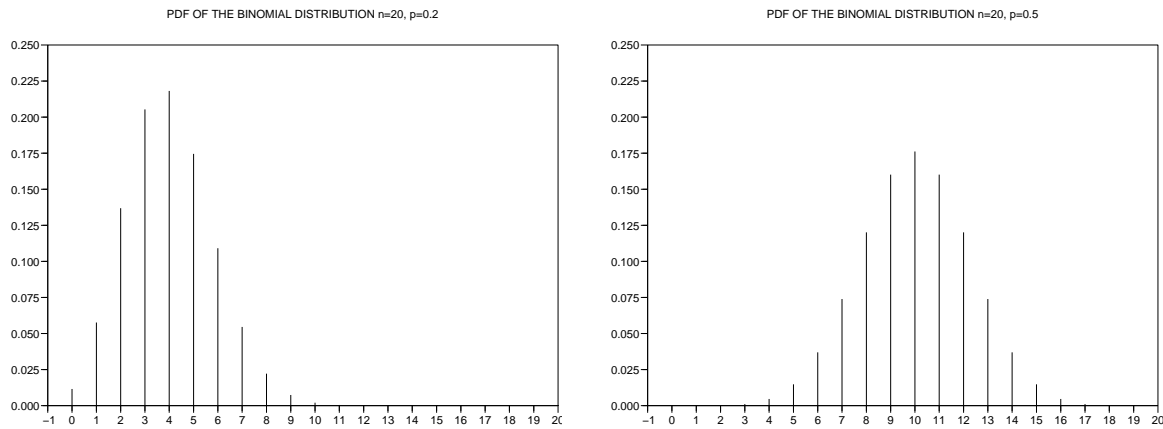


Figure 13: Example of function `pdfbinomial`

## 8.4 pdfchi2 – $\chi^2$ (central and non-central) pdf

**Calling Sequence**

```
Y=pdfchi2(X,n)
Y=pdfchi2(X,n,nc)
```

**Parameters**

- **X, Y** : real matrices **X** and **Y**.
- **n** : parameter  $n$  of the  $\chi^2$  distribution. Must be an integer  $\geq 1$ .
- **nc** : noncentrality parameter. Must be  $\geq 0$ . Default is 0.

**Description**

Compute in matrix **Y** the pdf of the  $\chi^2$  distribution for each entry  $X_{i,j}$  of matrix **X**. `pdfchi2(X,n)` is equivalent to `pdfchi2(X,n,0)`.

**Examples** (see Figure 14)

```
X=linspace(-1,10,300)';
Y1=pdfchi2(X,2);
xset("window",0);xbasc(0);plot2d(X,Y1,1,rect=[-1,0,10,0.5],nax=[0,12,0,11])
xtitle("PDF OF THE CHI-2 DISTRIBUTION n=2");xselect()
//
Y2=pdfchi2(X,4,1);
```

```
xset("window",1);xbasc(1);plot2d(X,Y2,1,rect=[-1,0,10,0.5],nax=[0,12,0,11])
xtitle("PDF OF THE CHI-2 DISTRIBUTION n=4, nc=1");xselect()
```

See Also

`cdfchi2`, `idfchi2`

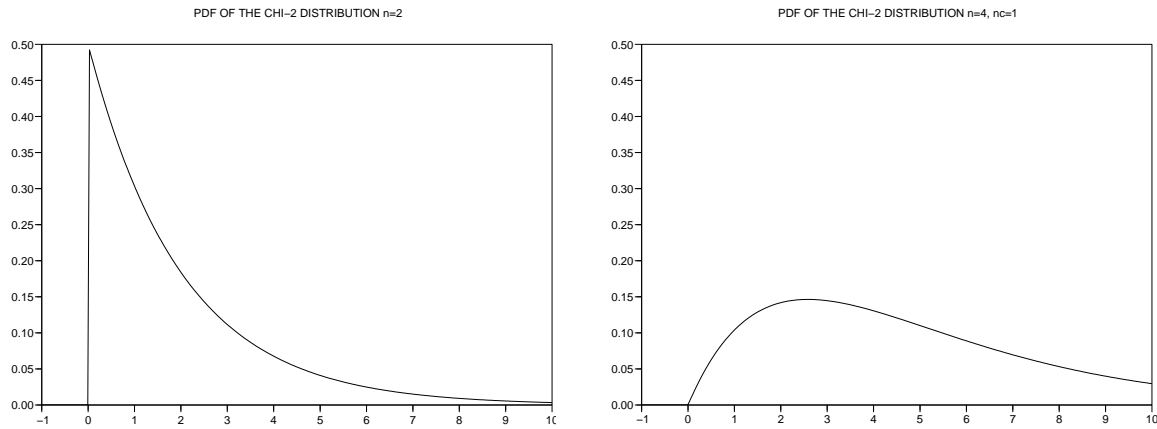


Figure 14: Example of function `pdfchi2`

## 8.5 pdfcp – $C_P$ pdf

### Calling Sequence

```
Y=pdfcp(X,sit,n)
```

### Parameters

- **X, Y** : real matrices **X** and **Y**.
- **sit** : parameter  $\sigma_t = \sigma/(U - T) > 0$  of the  $C_P$  distribution.
- **n** : parameter  $n$  of the  $C_P$  distribution. Must be an integer  $\geq 2$ .

### Description

Compute in matrix **Y** the pdf of the  $C_P$  distribution for each entry  $X_{i,j}$  of matrix **X**.

### Examples (see Figure 15)

```
X=linspace(0,3,300)';
L=4.5;
U=5.5;
T=(L+U)/2;
sigma=0.1;
sit=sigma/(U-T);
Cp=(U-L)/(6*sigma);
Y1=pdfcp(X,sit,20);
xset("window",0);xbasc(0)
plot2d(X,Y1,1,rect=[0,0,3,3],nax=[4,7,4,7])
xtitle("PDF OF Cp DISTRIBUTION sit=0.2, n=20 (E(Cp)=1.67)");xselect()
//
sigma=0.15;
sit=sigma/(U-T);
Cp=(U-L)/(6*sigma);
Y2=pdfcp(X,sit,30);
xset("window",1);xbasc(1)
plot2d(X,Y2,1,rect=[0,0,3,3],nax=[4,7,4,7])
xtitle("PDF OF Cp DISTRIBUTION sit=0.3, n=30 (E(Cp)=1.11)");xselect()
```

See Also

pdfcpk, pdfcpm, pdfcpmk, pdfcpuv

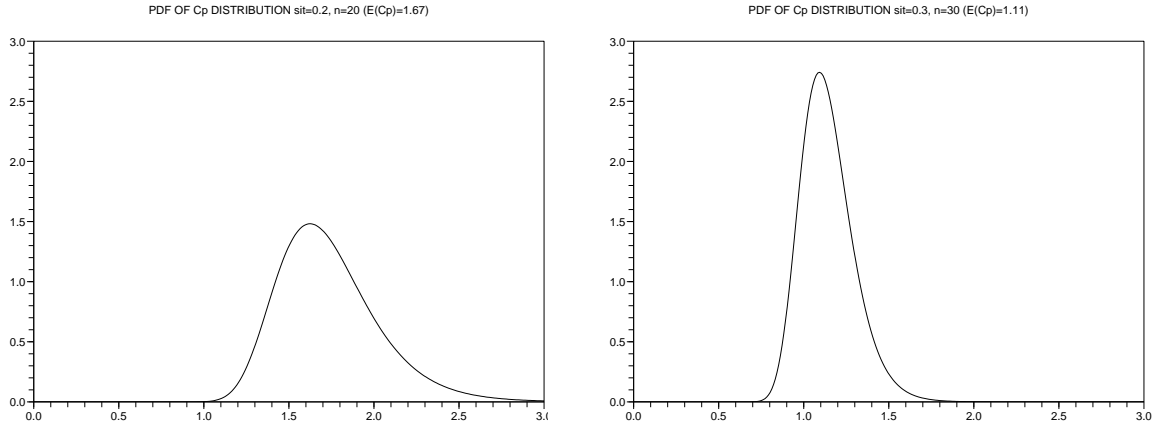


Figure 15: Example of function pdfcp

## 8.6 pdfcpk – $C_{PK}$ pdf

### Calling Sequence

`Y=pdfcpk(X,mu_t,sit,n)`

### Parameters

- `X,Y` : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .
- `mu_t` : parameter  $\mu_t = (\mu - T)/(U - T)$  of the  $C_{PK}$  distribution.
- `sit` : parameter  $\sigma_t = \sigma/(U - T) > 0$  of the  $C_{PK}$  distribution.
- `n` : parameter  $n$  of the  $C_{PK}$  distribution. Must be an integer  $\geq 2$ .

### Description

Compute in matrix  $\mathbf{Y}$  the pdf of the  $C_{PK}$  distribution for each entry  $X_{i,j}$  of matrix  $\mathbf{X}$ .

**Examples** (see Figure 16)

```
X=linspace(0,3,300)';
L=4.5;
U=5.5;
T=(L+U)/2;
sigma=0.1;
sit=sigma/(U-T);
mu=5.1;
mu_t=(mu-T)/(U-T);
Cpk=min(U-mu,mu-L)/(3*sigma);
Y1=pdfcpk(X,mu_t,sit,20);
xset("window",0);xbase(0)
plot2d(X,Y1,1,rect=[0,0,3,3],nax=[4,7,4,7])
xtitle("PDF OF Cpk DISTRIBUTION mu_t=0.2, sit=0.2, n=20 (E(Cpk)=1.33)")
xselect()
//
mu=5.2;
mu_t=(mu-T)/(U-T);
Cpk=min(U-mu,mu-L)/(3*sigma);
Y2=pdfcpk(X,mu_t,sit,30);
xset("window",1);xbase(1)
plot2d(X,Y2,1,rect=[0,0,3,3],nax=[4,7,4,7])
xtitle("PDF OF Cpk DISTRIBUTION mu_t=0.4, sit=0.2, n=30 (E(Cpk)=1)")
xselect()
```

See Also

pdfcp, pdfcpm, pdfcpmk, pdfcpuv

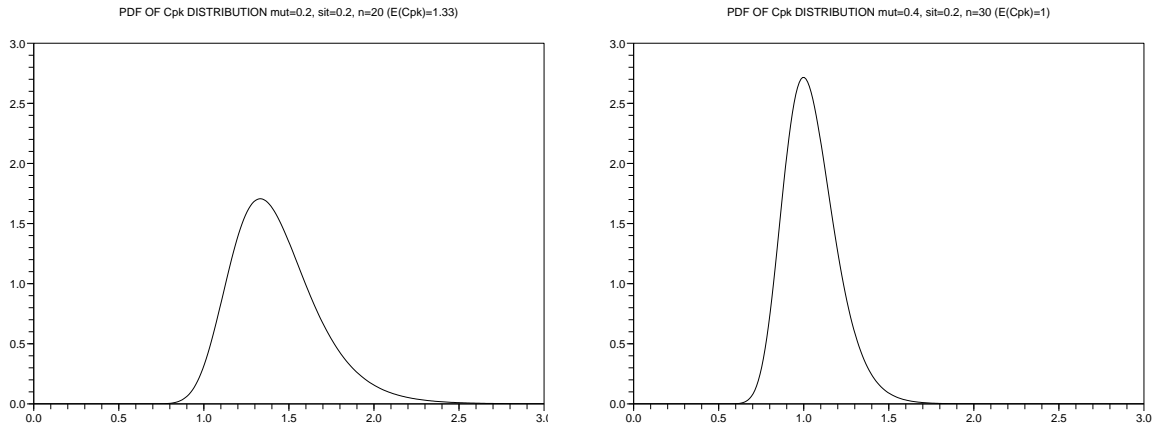


Figure 16: Example of function pdfcpk

## 8.7 pdfcpm – $C_{PM}$ pdf

### Calling Sequence

`Y=pdfcpm(X,mu,t,sit,n)`

### Parameters

- `X,Y` : real matrices **X** and **Y**.
- `mut` : parameter  $\mu_t = (\mu - T)/(U - T)$  of the  $C_{PM}$  distribution.
- `sit` : parameter  $\sigma_t = \sigma/(U - T) > 0$  of the  $C_{PM}$  distribution.
- `n` : parameter  $n$  of the  $C_{PM}$  distribution. Must be an integer  $\geq 2$ .

### Description

Compute in matrix **Y** the pdf of the  $C_{PM}$  distribution for each entry  $X_{i,j}$  of matrix **X**.

### Examples (see Figure 17)

```
X=linspace(0,3,300)';
L=4.5;
U=5.5;
T=(L+U)/2;
sigma=0.1;
sit=sigma/(U-T);
mu=5.1;
mut=(mu-T)/(U-T);
Cpm=(U-L)/(6*sqrt(sigma^2+(mu-T)^2));
Y1=pdfcpm(X,mu,t,sit,20);
xset("window",0);xbasc(0)
plot2d(X,Y1,1,rect=[0,0,3,3],nax=[4,7,4,7])
xtitle("PDF OF Cpm DISTRIBUTION mut=0.2, sit=0.2, n=20 (E(Cpm)=1.18)")
xselect()
//
mu=5.02;
mut=(mu-T)/(U-T);
Cpm=(U-L)/(6*sqrt(sigma^2+(mu-T)^2));
Y2=pdfcpm(X,mu,t,sit,30);
xset("window",1);xbasc(1)
plot2d(X,Y2,1,rect=[0,0,3,3],nax=[4,7,4,7])
xtitle("PDF OF Cpm DISTRIBUTION mut=0.04, sit=0.2, n=30 (E(Cpm)=1.63)")
xselect()
```

See Also

pdfcp, pdfcpk, pdfcpmk, pdfcpuv

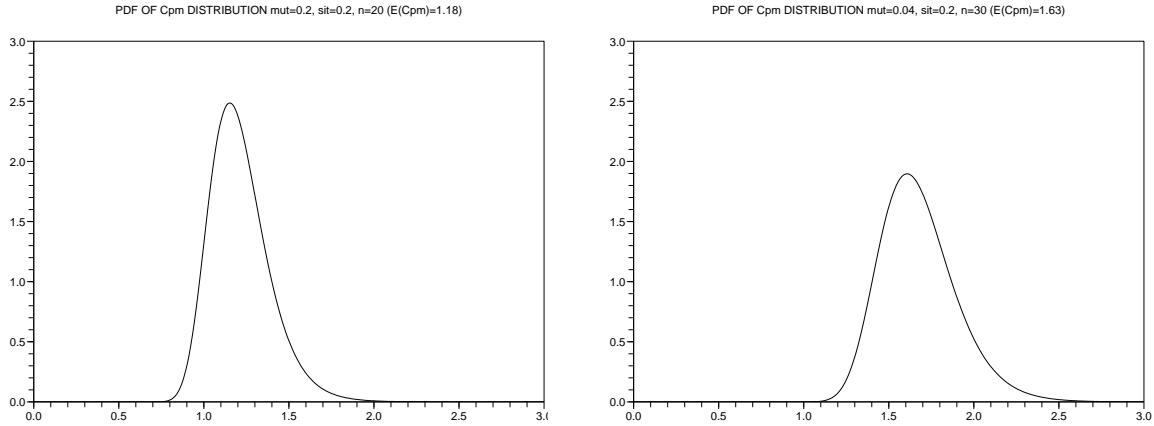


Figure 17: Example of function pdfcpm

## 8.8 pdfcpmk – $C_{PMK}$ pdf

### Calling Sequence

$Y = \text{pdfcpmk}(X, \text{mut}, \text{sit}, n)$

### Parameters

- $X, Y$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .
- $\text{mut}$  : parameter  $\mu_t = (\mu - T)/(U - T)$  of the  $C_{PMK}$  distribution.
- $\text{sit}$  : parameter  $\sigma_t = \sigma/(U - T) > 0$  of the  $C_{PMK}$  distribution.
- $n$  : parameter  $n$  of the  $C_{PMK}$  distribution. Must be an integer  $\geq 2$ .

### Description

Compute in matrix  $\mathbf{Y}$  the pdf of the  $C_{PMK}$  distribution for each entry  $X_{i,j}$  of matrix  $\mathbf{X}$ .

### Examples (see Figure 18)

```
X=linspace(0,3,300)';
L=4.5;
U=5.5;
T=(L+U)/2;
sigma=0.1;
sit=sigma/(U-T);
mu=5.1;
mut=(mu-T)/(U-T);
Cpmk=min(U-mu,mu-L)/(3*sqrt(sigma^2+(mu-T)^2));
Y1=pdfcpmk(X,mut,sit,20);
xset("window",0);xbasc(0)
plot2d(X,Y1,1,rect=[0,0,3,3],nax=[4,7,4,7])
xtitle("PDF OF Cpmk DISTRIBUTION mut=0.2, sit=0.2, n=20 (E(Cpmk)=0.94)")
xselect()
//
mu=5.02;
mut=(mu-T)/(U-T);
Cpmk=min(U-mu,mu-L)/(3*sqrt(sigma^2+(mu-T)^2));
Y2=pdfcpmk(X,mut,sit,30);
xset("window",1);xbasc(1)
plot2d(X,Y2,1,rect=[0,0,3,3],nax=[4,7,4,7])
xtitle("PDF OF Cpmk DISTRIBUTION mut=0.04, sit=0.2, n=30 (E(Cpmk)=1.57)")
xselect()
```

See Also

pdfcp, pdfcpk, pdfcpm, pdfcpuv

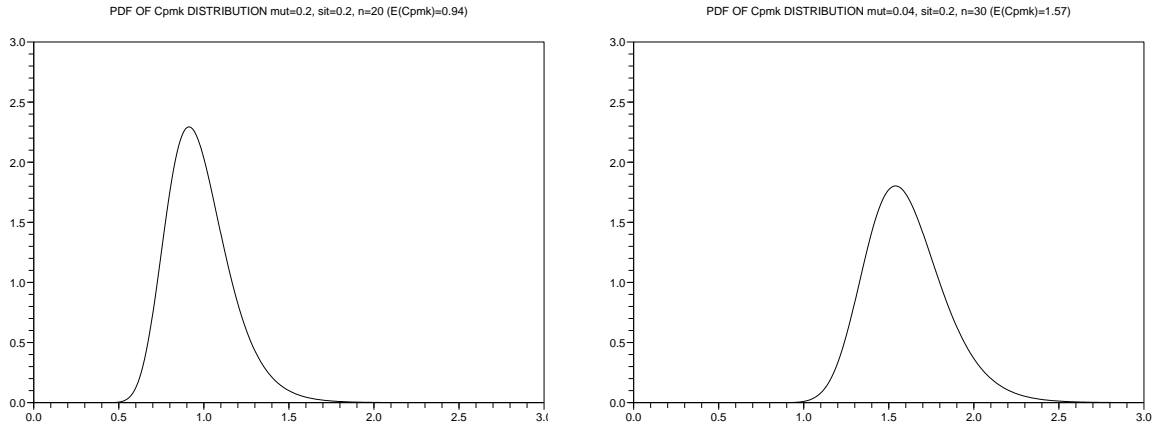


Figure 18: Example of function `pdfcpmk`

## 8.9 pdfcpuv – Vännman's $C_p(u, v)$ pdf

Calling Sequence

`Y=pdfcpuv(X,u,v,mu,t,sit,n)`

Parameters

- `X, Y` : real matrices **X** and **Y**.
- `u, v` : parameters ( $u \geq 0, v \geq 0$ ) of the Vännman's  $C_p(u, v)$  distribution.  $(u, v) \neq (0, 0)$ .
- `mu` : parameter  $\mu_t = (\mu - T)/(U - T)$  of the Vännman's  $C_p(u, v)$  distribution.
- `sit` : parameter  $\sigma_t = \sigma/(U - T) > 0$  of the Vännman's  $C_p(u, v)$  distribution.
- `n` : parameter  $n$  of the Vännman's  $C_p(u, v)$  distribution. Must be an integer  $\geq 2$ .

Description

Compute in matrix **Y** the pdf of the Vännman's  $C_p(u, v)$  distribution for each entry  $X_{i,j}$  of matrix **X**.

Examples (see Figure 19)

```
X=linspace(0,3,300)';
Y1=pdfcpuv(X,0,1,0.2,0.3,20);
xset("window",0);xbasc(0)
plot2d(X,Y1,1,rect=[0,0,3,3],nax=[4,7,4,7])
xtitle("PDF OF THE VANNMANN''s Cp(u,v) DISTRIBUTION u=0, v=1, mu=0.2, sit=0.3, n=20")
xselect()
//
Y2=pdfcpuv(X,1,0,0.3,0.15,40);
xset("window",1);xbasc(1)
plot2d(X,Y2,1,rect=[0,0,3,3],nax=[4,7,4,7])
xtitle("PDF OF THE VANNMANN''s Cp(u,v) DISTRIBUTION u=1, v=0, mu=0.3, sit=0.15, n=40")
xselect()
```

See Also

pdfcp, pdfcpk, pdfcpm, pdfcpmk

## 8.10 pdfcv – sample coefficient of variation pdf

Calling Sequence

`Y=pdfcv(X,n,cv)`

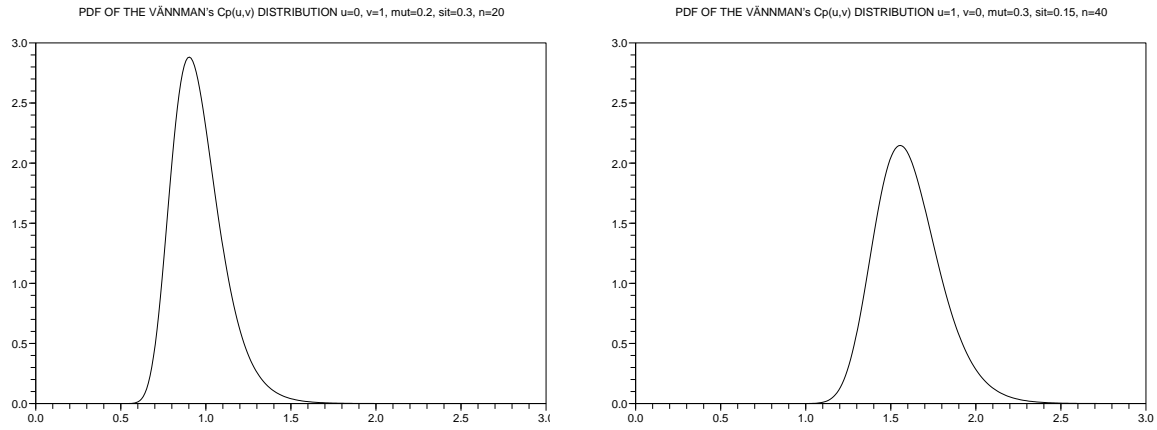


Figure 19: Example of function `pdfcpuv`

### Parameters

- $\mathbf{X}, \mathbf{Y}$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .
- $n$  : sample size. Must be an integer  $\geq 1$ .
- $cv$  : coefficient of variation  $\gamma = \sigma/\mu$ . Must be  $\geq 0$ .

### Description

Compute in matrix  $\mathbf{Y}$  the pdf of the sample coefficient of variation  $\hat{\gamma}$  for each entry  $X_{i,j}$  of matrix  $\mathbf{X}$ . If  $X_1, \dots, X_n$  are  $n$  normal  $(\mu, \sigma)$  random variables, the sample coefficient of variation  $\hat{\gamma} = S/\bar{X}$ , where  $S$  is the sample standard-deviation and  $\bar{X}$  the sample mean.

**Example** (see Figure 20)

```
X=linspace(-0.1,3,300)';
Y=pdfcv(X,7,0.8);
xset("window",0);xbasc(0);plot2d(X,Y,1)
xtitle("PDF OF THE COEFFICIENT OF VARIATION n=7, cv=0.8");xselect()
```

See Also

`cdfcv`, `idfcv`

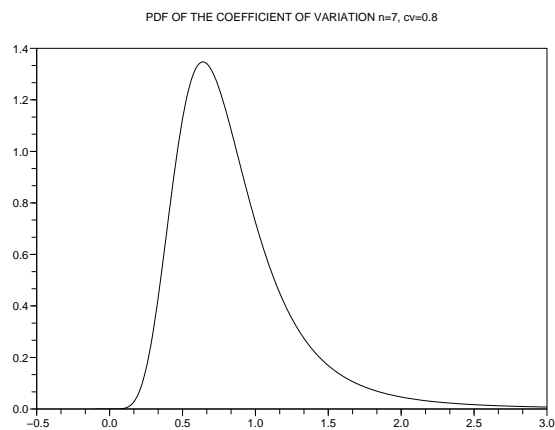


Figure 20: Example of function `pdfcv`

## 8.11 pdfdphase – discrete phase-type pdf

Calling Sequence

```
Y=pdfdphase(X,Q,q)
```

#### Parameters

- $\mathbf{X}, \mathbf{Y}$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .
- $\mathbf{Q}$  : square matrix  $\mathbf{Q}$  of transient probabilities.
- $\mathbf{q}$  : vector  $\mathbf{q}$  of initial transient probabilities.

#### Description

Compute in matrix  $\mathbf{Y}$  the pdf of the Discrete Phase-Type ( $\mathbf{Q}, \mathbf{q}$ ) distribution for each entry  $X_{i,j}$  of matrix  $\mathbf{X}$ . The Discrete Phase-Type distribution is defined on  $\{1, 2, 3, \dots\}$ .

**Examples** (see Figure 21)

```
X=(0:20)';
Y1=pdfdphase(X,[0.6,0.3;0.2,0.5],[1;0]);
xset("window",0);xbas(0)
plot2d3(X,Y1,1,rect=[0,0,20,0.2],nax=[0,21,0,11])
xtitle("PDF OF THE DPHASE DISTRIBUTION Q=[0.6,0.3;0.2,0.5], q=[1,0]")
xselect()
//
Y2=pdfdphase(X,[0.5,0.2;0.1,0.8],[0.5,0.5]);
xset("window",1);xbas(1)
plot2d3(X,Y2,1,rect=[0,0,20,0.2],nax=[0,21,0,11])
xtitle("PDF OF THE DPHASE DISTRIBUTION Q=[0.5,0.2;0.1,0.8], q=[0.5;0.5]")
xselect()
```

#### See Also

`cdfdphase`, `momdphase`

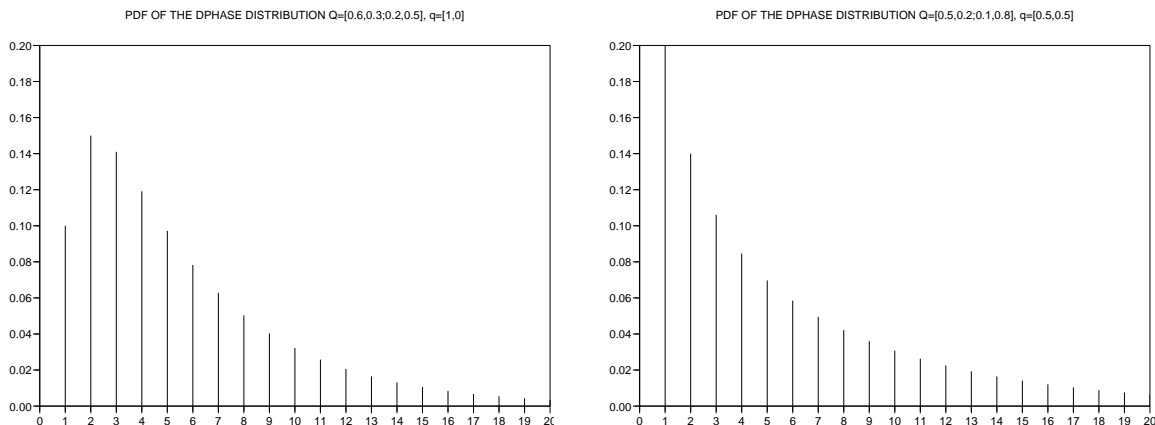


Figure 21: Example of function `pdfdphase`

## 8.12 pdfexponential – exponential pdf

#### Calling Sequence

```
Y=pdfexponential(X,lam)
```

#### Parameters

- $\mathbf{X}, \mathbf{Y}$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .
- `lam` : parameter  $\lambda > 0$  of the exponential distribution.

#### Description

Compute in matrix  $\mathbf{Y}$  the pdf of the exponential ( $\lambda$ ) distribution for each entry  $X_{i,j}$  of matrix  $\mathbf{X}$ .

**Examples** (see Figure 22)



```

X=linspace(-1,6,300)';
Y1=pdfexponential(X,0.5);
xset("window",0);xbasc(0);plot2d(X,Y1,1,rect=[-1,0,6,2],nax=[0,8,1,11])
xlabel("PDF OF THE EXPONENTIAL DISTRIBUTION lam=0.5");xselect()
//
Y2=pdfexponential(X,2);
xset("window",1);xbasc(1);plot2d(X,Y2,1,rect=[-1,0,6,2],nax=[0,8,1,11])
xlabel("PDF OF THE EXPONENTIAL DISTRIBUTION lam=2");xselect()

```

See Also

`cdfexponential`, `idfexponential`, `rndexponential`

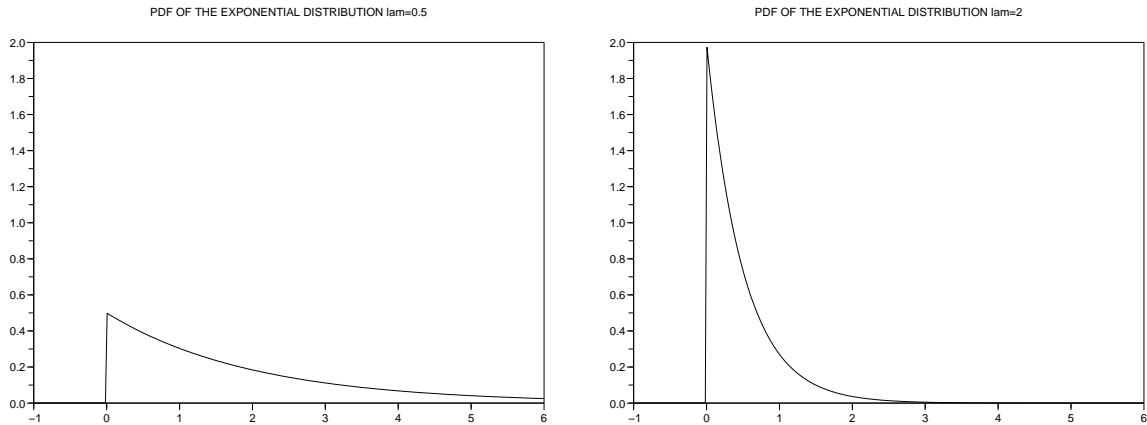


Figure 22: Example of function `pdfexponential`

### 8.13 `pdffisher` – Fisher (central and non-central) pdf

Calling Sequence

```

Y=pdffisher(X,m,n)
Y=pdffisher(X,m,n,nc)

```

Parameters

- $\mathbf{X}, \mathbf{Y}$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .
- $n, m$  : parameters  $m$  and  $n$  of the Fisher distribution. Must be integers  $\geq 1$ .
- $nc$  : noncentrality parameter. Must be  $\geq 0$ . Default is 0.

Description

Compute in matrix  $\mathbf{Y}$  the pdf of the Fisher  $(m, n)$  distribution for each entry  $X_{i,j}$  of matrix  $\mathbf{X}$ . `cdffisher(X,m,n)` is equivalent to `cdffisher(X,m,n,0)`.

Examples (see Figure 23)

```

X=linspace(-1,7,300)';
Y1=pdffisher(X,2,3);
xset("window",0);xbasc(0);plot2d(X,Y1,1,rect=[-1,0,7,1])
xlabel("PDF OF THE FISHER DISTRIBUTION m=2, n=3");xselect()
//
Y2=pdffisher(X,11,9,4);
xset("window",1);xbasc(1);plot2d(X,Y2,1,rect=[-1,0,7,1])
xlabel("PDF OF THE FISHER DISTRIBUTION m=11, n=9, nc=4");xselect()

```

See Also

`cdffisher`, `idffisher`

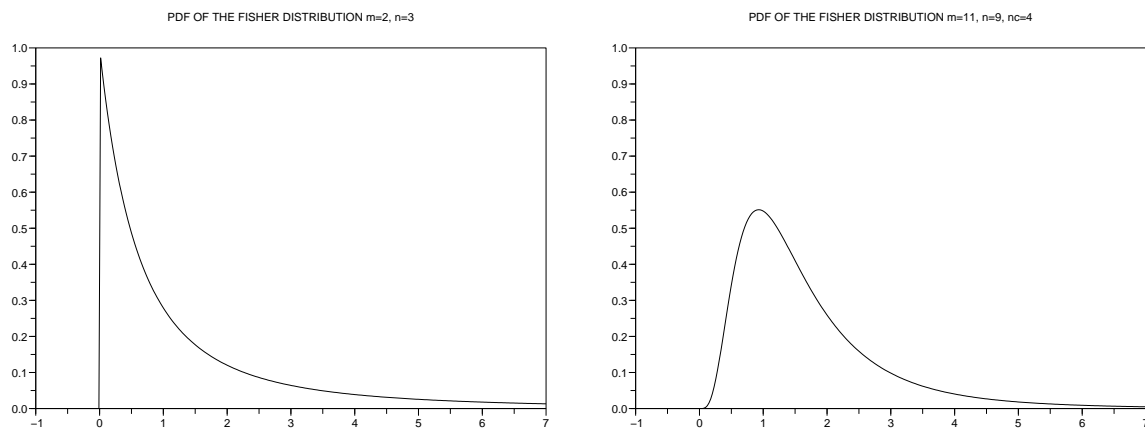


Figure 23: Example of function `pdffisher`

## 8.14 `pdffoldednormal` – folded normal pdf

### Calling Sequence

`Y=pdffoldednormal(X,mu=,sigma=,c=)`

### Parameters

- `X,Y` : real matrices **X** and **Y**.
- `mu` : parameter  $\mu$  of the folded normal distribution. Default is 0.
- `sigma` : parameter  $\sigma > 0$  of the folded normal distribution. Default is 1.
- `c` : parameter  $c$  of the folded normal distribution. Default is 0.

### Description

Compute in matrix **Y** the pdf of the folded normal distribution for each entry  $X_{i,j}$  of matrix **X**. The folded normal  $(\mu, \sigma, c)$  distribution is defined on  $[c, +\infty)$ . `pdffoldednormal(X)` is equivalent to `pdffoldednormal(X,0,1,0)`.

### Examples (see Figure 24)

```
X=linspace(-1,7,300)';
Y1=pdffoldednormal(X);
xset("window",0);xbasc(0);plot2d(X,Y1,1,rect=[-1,0,7,0.8])
xtitle("PDF OF THE FOLDED NORMAL DISTRIBUTION mu=0, sigma=1, c=0");xselect()
//
Y2=pdffoldednormal(X,2,1.5,1);
xset("window",1);xbasc(1);plot2d(X,Y2,1,rect=[-1,0,7,0.8])
xtitle("PDF OF THE FOLDED NORMAL DISTRIBUTION mu=2, sigma=1.5, c=1");xselect()
```

### See Also

`cdffoldednormal`, `idffoldednormal`, `rndfoldednormal`

## 8.15 `pdfgamma` – gamma pdf

### Calling Sequence

`Y=pdfgamma(X,a,b=,c=,d=)`

### Parameters

- `X,Y` : real matrices **X** and **Y**.
- `a` : parameter  $a > 0$  of the gamma distribution.
- `b` : parameter  $b > 0$  of the gamma distribution. Default is 1.
- `c` : parameter  $c$  of the gamma distribution. Default is 0.
- `d` : parameter  $d \neq 0$  of the gamma distribution. Default is 1.

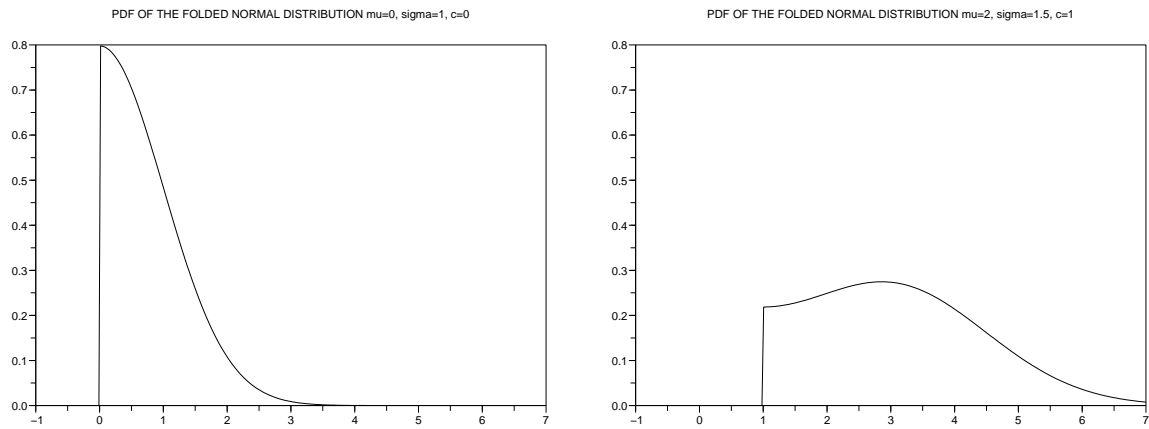


Figure 24: Example of function `pdffoldednormal`

## Description

Compute in matrix  $\mathbf{Y}$  the pdf of the gamma  $(a, b, c, d)$  distribution for each entry  $X_{i,j}$  of matrix  $\mathbf{X}$ . The gamma  $(a, b, c, d)$  distribution is defined on  $[c, +\infty[$ . `pdfgamma(X,a)` is equivalent to `pdfgamma(X,a,1,0,1)`.

**Examples** (see Figure 25)

```
X=linspace(-1,10,300)';
Y1=pdfgamma(X,2);
xset("window",0);xbasc(0);plot2d(X,Y1,1,rect=[-1,0,10,0.7],nax=[1,12,1,8])
xtitle("PDF OF THE GAMMA DISTRIBUTION a=2, b=1, c=0, d=1");xselect()
//
Y2=pdfgamma(X,5,0.7,2,1.5);
xset("window",1);xbasc(1);plot2d(X,Y2,1,rect=[-1,0,10,0.7],nax=[1,12,1,8])
xtitle("PDF OF THE GAMMA DISTRIBUTION a=5, b=0.7, c=2, d=1.5");xselect()
```

**See Also**

`cdfgamma`, `fitgamma`, `idfgamma`, `rndgamma`

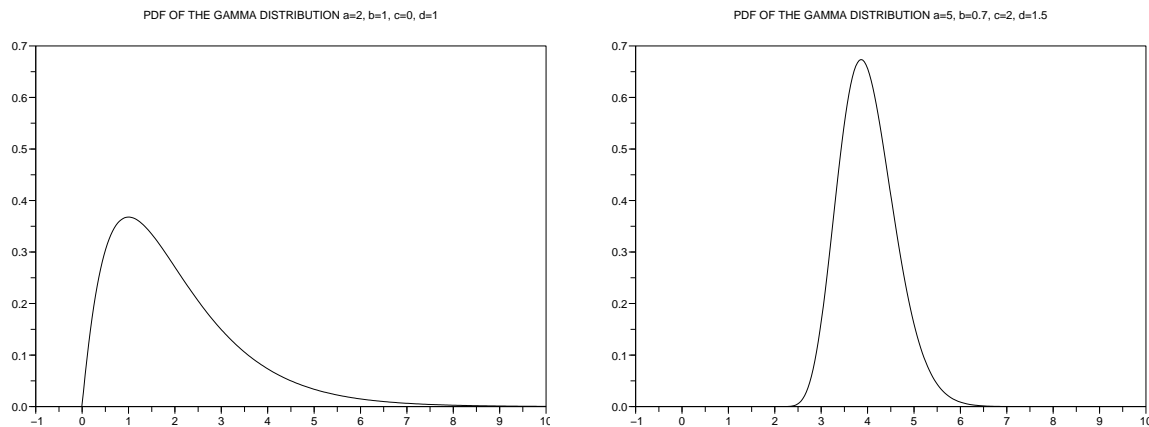


Figure 25: Example of function `pdfgamma`

## 8.16 pdfgev – generalized Extreme Value pdf

**Calling Sequence**

`Y=pdfgev(X,a,b,c=)`

#### Parameters

- `X,Y` : real matrices **X** and **Y**.
- `a` : parameter  $a$  of the GEV distribution.
- `b` : parameter  $b > 0$  of the GEV distribution. Default is 1.
- `c` : parameter  $c$  of the GEV distribution. Default is 0.

#### Description

Compute in matrix **Y** the pdf of the GEV  $(a, b, c)$  distribution for each entry  $X_{i,j}$  of matrix **X**. `pdfgev(X,a)` is equivalent to `pdfgev(X,a,1,0)`.

#### Examples (see Figure 26)

```
X=linspace(-1,7,300)';
Y1=pdfgev(X,0.5);
xset("window",0);xbasc(0);plot2d(X,Y1,1)
xtitle("PDF OF THE GEV DISTRIBUTION a=0.5, b=1, c=0");xselect()
//
Y2=pdfgev(X,-0.5,c=5);
xset("window",1);xbasc(1);plot2d(X,Y2,1)
xtitle("PDF OF THE GEV DISTRIBUTION a=-0.5, b=1, c=5");xselect()
```

#### See Also

`cdfgev`, `fitgev`, `idfgev`, `rndgev`

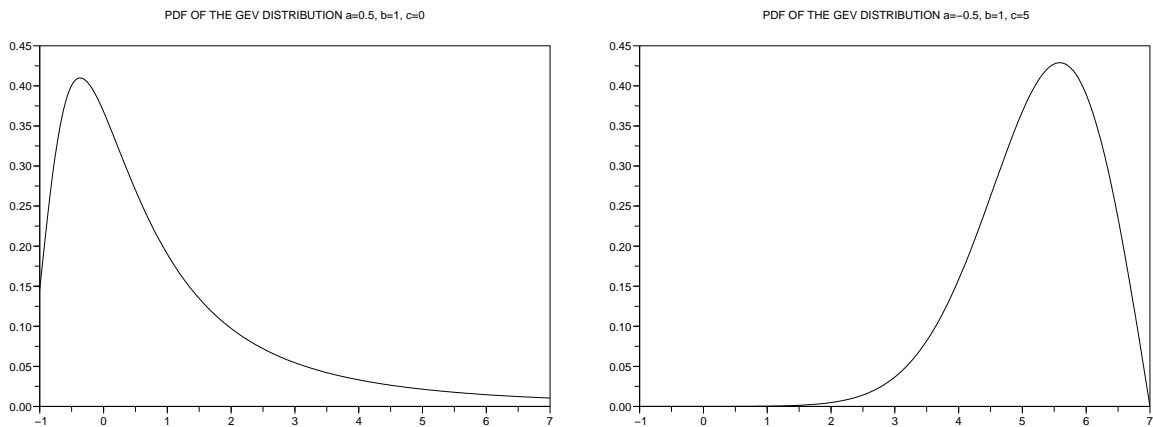


Figure 26: Example of function `pdfgev`

## 8.17 pdfhypergeometric – hypergeometric pdf

#### Calling Sequence

`Y=pdfhypergeometric(X,n,p,N)`

#### Parameters

- `X,Y` : real matrices **X** and **Y**.
- `n` : parameter  $n$  of the hypergeometric distribution. Must be an integer in  $\{1, \dots, N\}$ .
- `p` : parameter  $p \in [0, 1]$  of the hypergeometric distribution.
- `N` : parameter  $N$  of the hypergeometric distribution. Must be an integer  $\geq 1$ .

#### Description

Compute in matrix **Y** the pdf of the hypergeometric  $(n, p, N)$  distribution for each entry  $X_{i,j}$  of matrix **X**.

#### Examples (see Figure 27)

```

X=(-1:20)';
Y1=pdfhypergeometric(X,20,0.3,100);
xset("window",0);xbasc(0);plot2d3(X,Y1,1,rect=[-1,0,20,0.35],nax=[0,22,0,11])
xtitle("PDF OF THE HYPERGEOMETRIC DISTRIBUTION n=20, p=0.3, N=100");xselect()
//
Y2=pdfhypergeometric(X,20,0.1,200);
xset("window",1);xbasc(1);plot2d3(X,Y2,1,rect=[-1,0,20,0.35],nax=[0,22,0,11])
xtitle("PDF OF THE HYPERGEOMETRIC DISTRIBUTION n=20, p=0.1, N=200");xselect()

```

See Also

`cdfhypergeometric`

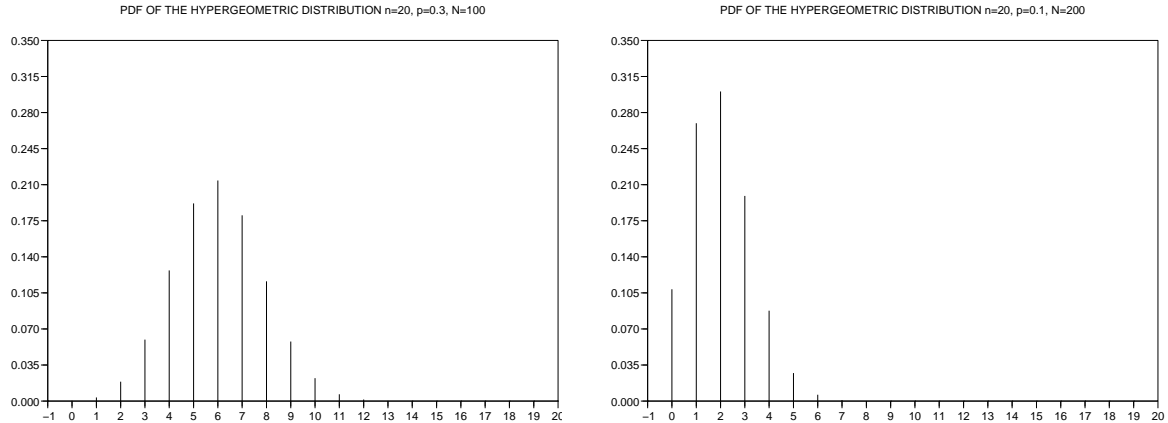


Figure 27: Example of function `pdfhypergeometric`

## 8.18 pdfkernel – kernel smoothed pdf

### Calling Sequence

```

Y=pdfkernel(X,Z,h)
Y=pdfkernel(X,Z,h,ker)

```

### Parameters

- **X, Y, Z** : real matrices **X**, **Y** and **Z**
- **h** : width  $h \geq 0$  of the kernel.
- **ker** : type of kernel. Must be "uniform", "triangular", "epanechnikov", "biweight", "triweight" or "normal". Default is "epanechnikov".

### Description

Compute in matrix **Y** the kernel smoothed pdf for each entry  $X_{i,j}$  of matrix **X** based on data in matrix **Z**. `pdfkernel(X,Z,h)` is equivalent to `pdfkernel(X,Z,h,"epanechnikov")`.

### Examples (see Figure 28)

```

Z1=rndnormal(200,20,0.1);
Z2=rndnormal(200,20.4,0.15);
Z=[Z1;Z2];
X=linspace(19.5,21)';
Y=(pdfnormal(X,20,0.1)+pdfnormal(X,20.4,0.15))/2;
//
Yu=pdfkernel(X,Z,0.05,"uniform");
xset("window",0);xbasc();plot2d([X,X],[Yu,Y],[2,1])
xtitle("UNIFORM KERNEL");xselect()
//
Yt=pdfkernel(X,Z,0.05,"triangular");

```

```

xset("window",1);xasc();plot2d([X,X],[Yt,Y],[2,1])
xtitle("TRIANGULAR KERNEL");xselect()
//
Ye=pdfkernel(X,Z,0.05,"epanechnikov");
xset("window",2);xasc();plot2d([X,X],[Ye,Y],[2,1])
xtitle("EPANECHNIKOV KERNEL");xselect()
//
Y2=pdfkernel(X,Z,0.05,"biweight");
xset("window",3);xasc();plot2d([X,X],[Y2,Y],[2,1])
xtitle("BIWEIGHT KERNEL");xselect()
//
Y3=pdfkernel(X,Z,0.05,"triweight");
xset("window",4);xasc();plot2d([X,X],[Y3,Y],[2,1])
xtitle("TRIWEIGHT KERNEL");xselect()
//
Yn=pdfkernel(X,Z,0.05,"normal");
xset("window",5);xasc();plot2d([X,X],[Yn,Y],[2,1])
xtitle("NORMAL KERNEL");xselect()

```

## 8.19 pdfjohnson – Johnson’s pdf

### Calling Sequence

```
Y=pdfjohnson(X,s,a,b,c,d)
```

### Parameters

- $\mathbf{X}, \mathbf{Y}$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .
- $s$  : Johnson’s system of distribution. Must be "B" for the Johnson’s  $S_B$  (bounded) system of distributions or "U" for the Johnson’s  $S_U$  (unbounded) system of distributions.
- $a$  : parameter  $a$  of the Johnson’s distribution.
- $b$  : parameter  $b > 0$  of the Johnson’s distribution.
- $c$  : parameter  $c$  of the Johnson’s distribution.
- $d$  : parameter  $d > 0$  of the Johnson’s distribution.

### Description

Compute in matrix  $\mathbf{Y}$  the pdf of the Johnson’s distribution for each entry  $X_{i,j}$  of matrix  $\mathbf{X}$ .

### Examples (see Figure 29)

```

X=linspace(0,6,300)';
Yb=pdfjohnson(X,"B",4,3,1,5);
xset("window",0);xasc(0);plot2d(X,Yb,1,rect=[0,0,6,1.5])
xtitle("PDF OF THE JOHNSON’S BOUNDED DISTRIBUTION a=4, b=3, c=1, d=5")
xselect()
//
Yu=pdfjohnson(X,"U",3,4,5,2);
xset("window",1);xasc(1);plot2d(X,Yu,1,rect=[0,0,6,1.5])
xtitle("PDF OF THE JOHNSON’S UNBOUNDED DISTRIBUTION a=3, b=4, c=5, d=2")
xselect()

```

### See Also

```
cdfjohnson, fitjohnson, idfjohnson, rndjohnson
```

## 8.20 pdflognormal – lognormal pdf

### Calling Sequence

```
Y=pdflognormal(X,a=,b=,c=)
```

### Parameters

- $\mathbf{X}, \mathbf{Y}$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .
- $a$  : parameter  $a$  of the lognormal distribution. Default is 0.

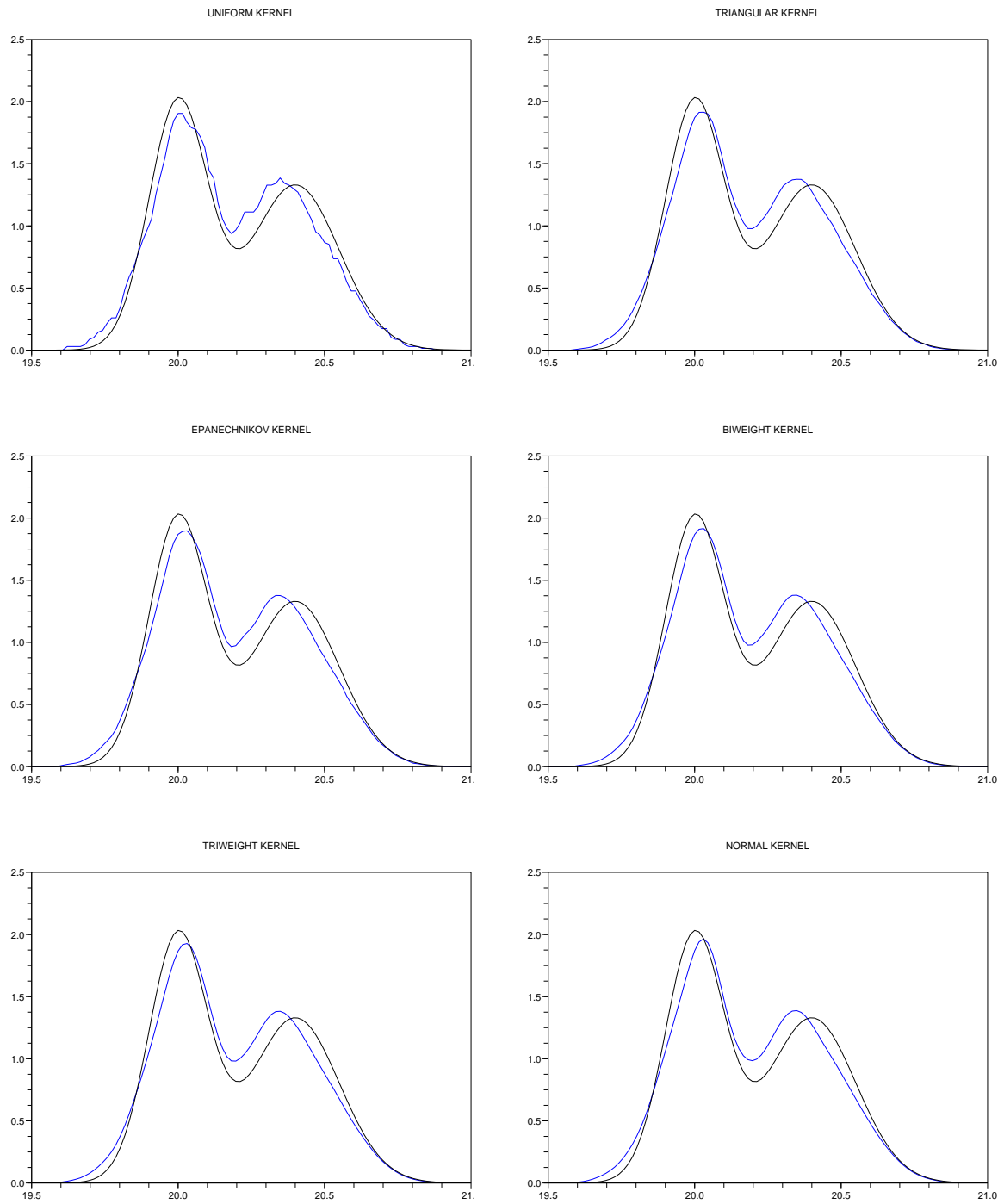


Figure 28: Example of function `pdfkernel`

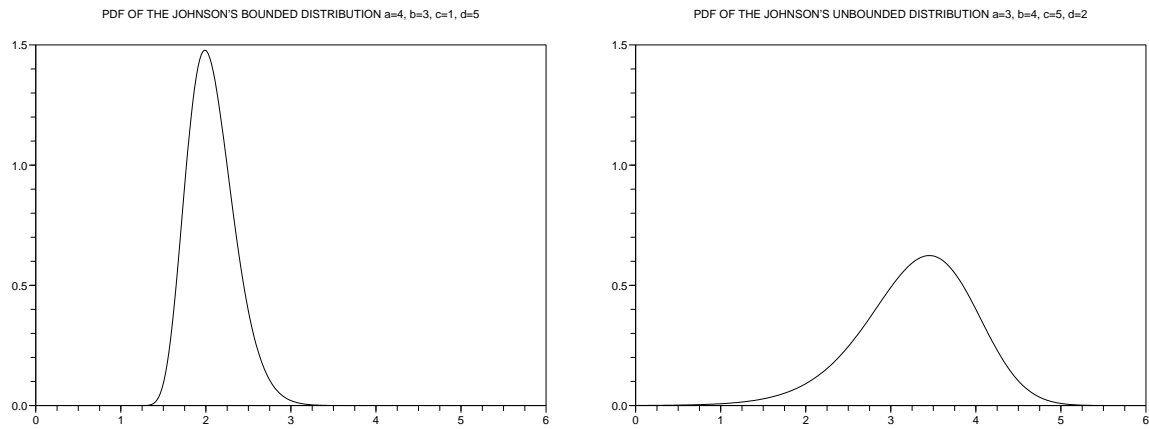


Figure 29: Example of function `pdfjohnson`

- `b` : parameter  $b > 0$  of the lognormal distribution. Default is 1.
- `c` : parameter  $c$  of the lognormal distribution. Default is 0.

### Description

Compute in matrix  $\mathbf{Y}$  the pdf of the lognormal distribution for each entry  $X_{i,j}$  of matrix  $\mathbf{X}$ . The lognormal distribution is defined on  $[c, +\infty)$ . `pdflognormal(x)` is equivalent to `pdflognormal(x,0,1,0)`.

### Examples (see Figure 30)

```
X=linspace(-1,7,300)';
Y1=pdflognormal(X,0.5,2);
xset("window",0);xbasc(0)
plot2d(X,Y1,1,rect=[-1,0,7,1.2],nax=[1,9,1,7])
xtitle("PDF OF THE LOGNORMAL DISTRIBUTION a=0.5, b=2, c=0");xselect()
//
Y2=pdflognormal(X,-0.5,c=0.5);
xset("window",1);xbasc(1)
plot2d(X,Y2,1,rect=[-1,0,7,1.2],nax=[1,9,1,7])
xtitle("PDF OF THE LOGNORMAL DISTRIBUTION a=-0.5, b=1, c=0.5");xselect()
```

### See Also

`cdflognormal`, `fitlognormal`, `idflognormal`, `rndlognormal`

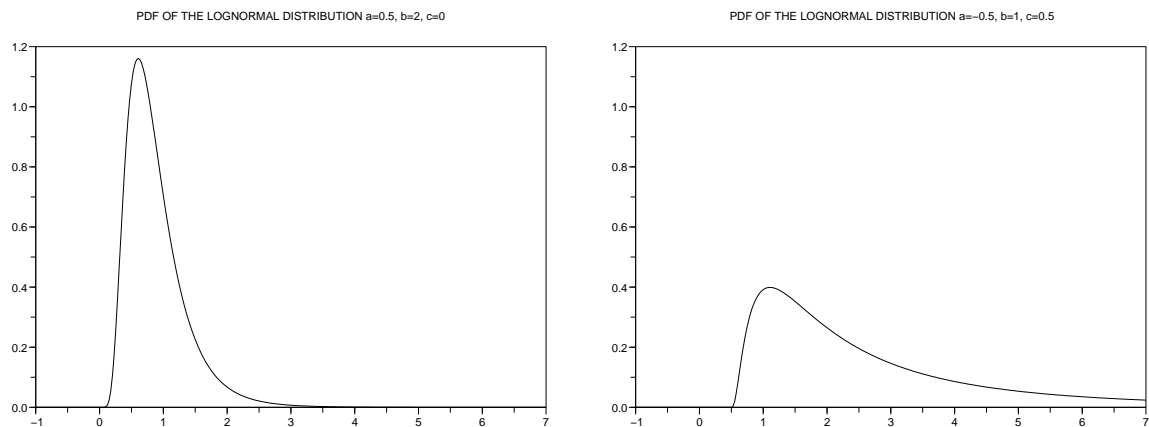


Figure 30: Example of function `pdflognormal`



## 8.21 pdfmedian – normal sample median pdf

### Calling Sequence

```
Y=pdfmedian(X,n,mu=,sigma=)
```

### Parameters

- $\mathbf{X}, \mathbf{Y}$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .
- $n$  : parameter  $n$  of the normal  $(\mu, \sigma)$  sample median distribution. Must be an odd integer  $\geq 1$ .
- $\mu$  : parameter  $\mu$  (mean) of the normal distribution. Default is 0.
- $\sigma$  : parameter  $\sigma > 0$  (standard-deviation) of the normal distribution. Default is 1.

### Description

Compute in matrix  $\mathbf{Y}$  the pdf of the normal  $(\mu, \sigma)$  sample median distribution for each entry  $X_{i,j}$  of matrix  $\mathbf{X}$ .

### Examples (see Figure 31)

```
X=linspace(-3,3,300)';
Y1=pdfmedian(X,3);
xbasec()
xset("window",0);xbasec(0);
plot2d(X,Y1,1,rect=[-3,0,3,2],nax=[1,7,1,11])
xtitle("PDF OF THE NORMAL SAMPLE MEDIAN n=3, mu=0, sigma=1");xselect()
//
Y2=pdfmedian(X,5,1,0.4);
xset("window",1);xbasec(1);plot2d(X,Y2,1)
xtitle("PDF OF THE NORMAL SAMPLE MEDIAN n=5, mu=1, sigma=0.4")
xselect()
```

### See Also

cdfmedian, idfmedian

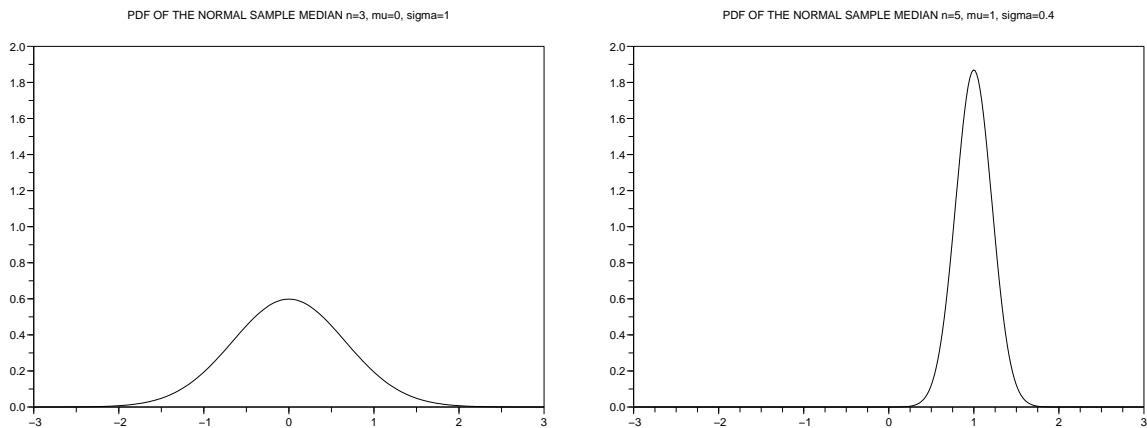


Figure 31: Example of function pdfmedian

## 8.22 pdfmultinormal – multinormal pdf

### Calling Sequence

```
y=pdfmultinormal(X,mu)
y=pdfmultinormal(X,mu,sigma)
```

### Parameters

- $\mathbf{X}$  : real matrix  $\mathbf{X}$  of size  $(n, p)$ .
- $\mu$  : mean vector  $\boldsymbol{\mu}$  of the multinormal distribution. Must be a  $(1, p)$  row vector.

- **sigma** : variance-covariance matrix  $\Sigma$  of the multinormal distribution. Must be a  $(p, p)$  definite positive matrix or  $(1, p)$  row vector  $(\sigma_1^2, \dots, \sigma_p^2)$  where  $\sigma_1^2, \dots, \sigma_p^2$  are the diagonal elements (variance) of matrix  $\Sigma$ . Default is **eye(p,p)**.
- **y** : column vector **y** of size  $(n, 1)$ .

### Description

Compute in vector **y** the pdf of the multinormal  $(\mu, \Sigma)$  distribution for each row  $X_{i, \cdot}$  of matrix **X**. **pdfmultinormal(X,mu)** is equivalent to **pdfmultinormal(X,mu,eye(p,p))**.

### Examples (see Figure 32)

```
X1=linspace(-4,4,50)';
Z1=ones(50,1).*X1;
X2=linspace(-4,4,50)';
Z2=X2.*ones(50,1);
Y1=pdfmultinormal([Z1,Z2],[0,0],[0.7,0.8]);
Y1=matrix(Y1,50,50);
xset("window",0);xbasc(0);plot3d(X1,X2,Y1,alpha=88,theta=56);xselect()
//
Y2=pdfmultinormal([Z1,Z2],[-1,2],[2,0.3;0.3,1]);
Y2=matrix(Y2,50,50);
xset("window",1);xbasc(1);plot3d(X1,X2,Y2,alpha=88,theta=56);xselect()
```

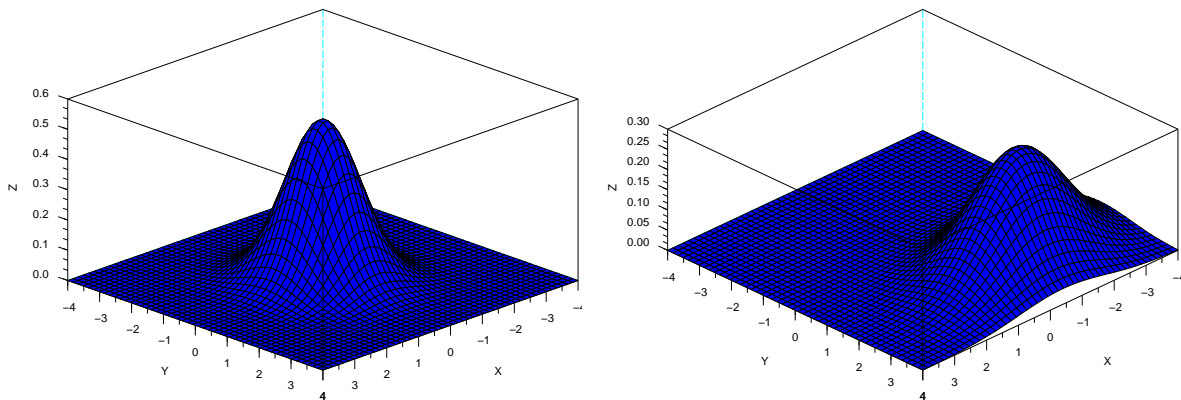


Figure 32: Example of function **pdfmultinormal**

## 8.23 pdfnormal – normal pdf

### Calling Sequence

```
Y=pdfnormal(X,mu=,sigma=)
```

### Parameters

- **X,Y** : real matrices **X** and **Y**.
- **mu** : parameter  $\mu$  (mean) of the normal distribution. Default is 0.
- **sigma** : parameter  $\sigma > 0$  (standard-deviation) of the normal distribution. Default is 1.

### Description

Compute in matrix **Y** the pdf of the normal distribution for each entry  $X_{i,j}$  of matrix **X**. **pdfnormal(X)** is equivalent to **pdfnormal(X,0,1)**.

### Examples (see Figure 33)

```
X=linspace(-4,8,200)';
Y1=pdfnormal(X);
xset("window",0);xbasc(0);plot2d(X,Y1,1,rect=[-4,0,8,0.4],nax=[0,13,1,9])
xtitle("PDF OF THE NORMAL DISTRIBUTION mu=0, sigma=1");xselect()
```

```
//
Y2=pdfnormal(X,3,2);
xset("window",1);xbasc(1);plot2d(X,Y2,1,rect=[-4,0,8,0.4],nax=[0,13,1,9])
xtitle("PDF OF THE NORMAL DISTRIBUTION mu=3, sigma=2");xselect()
```

See Also

`cdfnormal`, `idfnormal`, `rndnormal`

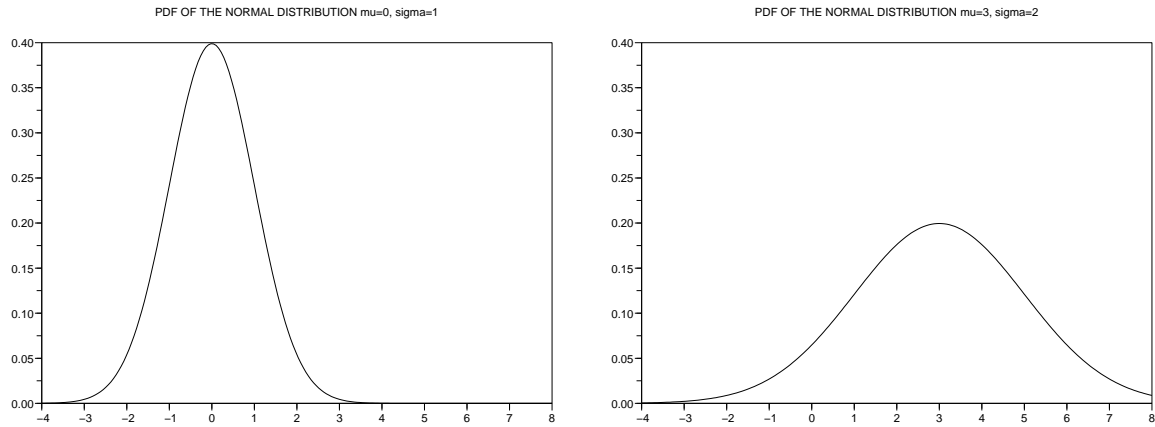


Figure 33: Example of function `pdfnormal`

## 8.24 `pdfpareto` – Pareto pdf

Calling Sequence

```
Y=pdfpareto(X,a,b=,c=)
```

Parameters

- **X**, **Y** : real matrices **X** and **Y**.
- **a** : parameter  $a$  of the Pareto distribution.
- **b** : parameter  $b > 0$  of the Pareto distribution. Default is 1.
- **c** : parameter  $c$  of the Pareto distribution. Default is 0.

Description

Compute in matrix **Y** the pdf of the Pareto distribution for each entry  $X_{i,j}$  of matrix **X**. The Pareto distribution is defined on

- $[c, +\infty)$  if  $a \geq 0$ ,
- $[c, c - b/a]$  if  $a < 0$ .

`pdfpareto(X,a)` is equivalent to `pdfpareto(X,a,1,0)`.

Examples (see Figure 34)

```
X=linspace(-1,7,300)';
Y1=pdfpareto(X,0.5);
xset("window",0);xbasc(0)
plot2d(X,Y1,1,rect=[-1,0,7,2],nax=[1,9,1,11])
xtitle("PDF OF THE PARETO DISTRIBUTION a=0.5, b=1, c=0");xselect()
//
Y2=pdfpareto(X,-2,6,1);
xset("window",1);xbasc(1)
plot2d(X,Y2,1)
xtitle("PDF OF THE PARETO DISTRIBUTION a=-2, b=6, c=1");xselect()
```

See Also

`cdfpareto`, `idfpareto`, `rndpareto`

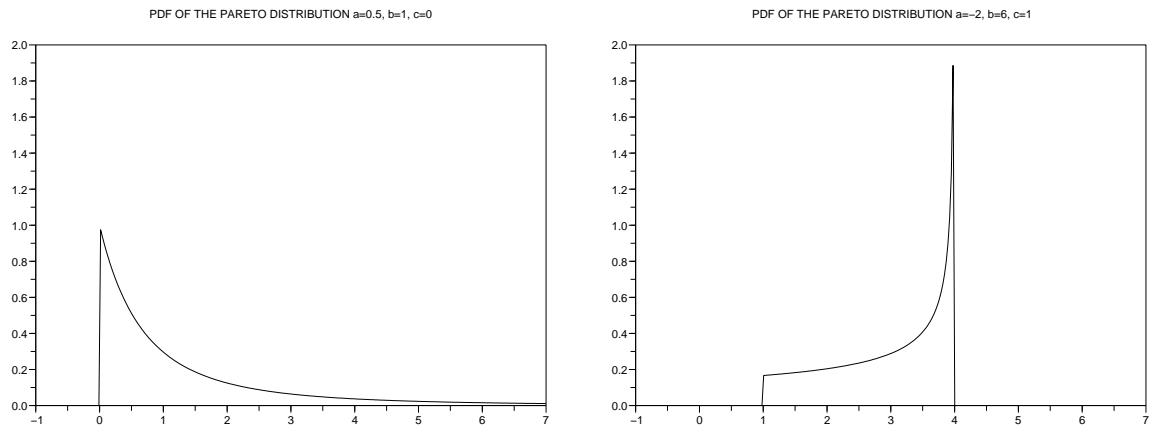


Figure 34: Example of function `pdfpareto`

## 8.25 `pdfpascal` – Pascal pdf

### Calling Sequence

`Y=pdfpascal(X,n,p)`

### Parameters

- `X,Y` : real matrices **X** and **Y**
- `n` : parameter  $n$  of the Pascal distribution. Must be an integer  $\geq 1$ .
- `p` : parameter  $p \in (0, 1]$  of the Pascal distribution.

### Description

Compute in matrix **Y** the pdf of the Pascal distribution for each entry  $X_{i,j}$  of matrix **X**.

### Examples (see Figure 35)

```
X=(0:30)';
Y1=pdfpascal(X,2,0.3);
xset("window",0);xbasc(0);plot2d3(X,Y1,1,rect=[0,0,30,0.14],nax=[4,7,1,8])
xtitle("PDF OF THE PASCAL DISTRIBUTION n=2, p=0.3");xselect()
//
Y2=pdfpascal(X,7,0.5);
xset("window",1);xbasc(1);plot2d3(X,Y2,1,rect=[0,0,30,0.14],nax=[4,7,1,8])
xtitle("PDF OF THE PASCAL DISTRIBUTION n=7, p=0.5");xselect()
```

### See Also

`cdfpascal`, `rndpascal`

## 8.26 `pdfpoisson` – Poisson pdf

### Calling Sequence

`Y=pdfpoisson(X,lam)`

### Parameters

- `X,Y` : real matrices **X** and **Y**.
- `lam` : parameter  $\lambda > 0$  of the Poisson distribution.

### Description

Compute in matrix **Y** the pdf of the Poisson distribution for each entry  $X_{i,j}$  of matrix **X**.

### Examples (see Figure 36)

```
X=(-1:15)';
Y1=pdfpoisson(X,0.8);
```

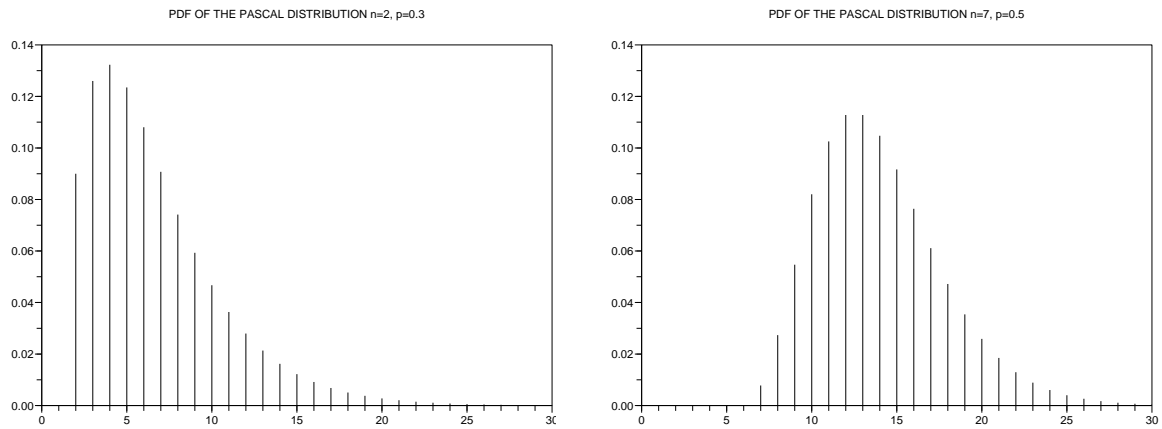


Figure 35: Example of function `pdfpascal`

```
xset("window",0);xbasc(0)
plot2d3(X,Y1,1,rect=[-1,0,12,0.5],nax=[0,14,0,11])
xtitle("PDF OF THE POISSON DISTRIBUTION lam=0.8");xselect()
//
Y2=pdfpoisson(X,3);
xset("window",1);xbasc(1)
plot2d3(X,Y2,1,rect=[-1,0,12,0.5],nax=[0,14,0,11])
xtitle("PDF OF THE POISSON DISTRIBUTION lam=3");xselect()
```

See Also

`cdfpoisson`, `rndpoisson`

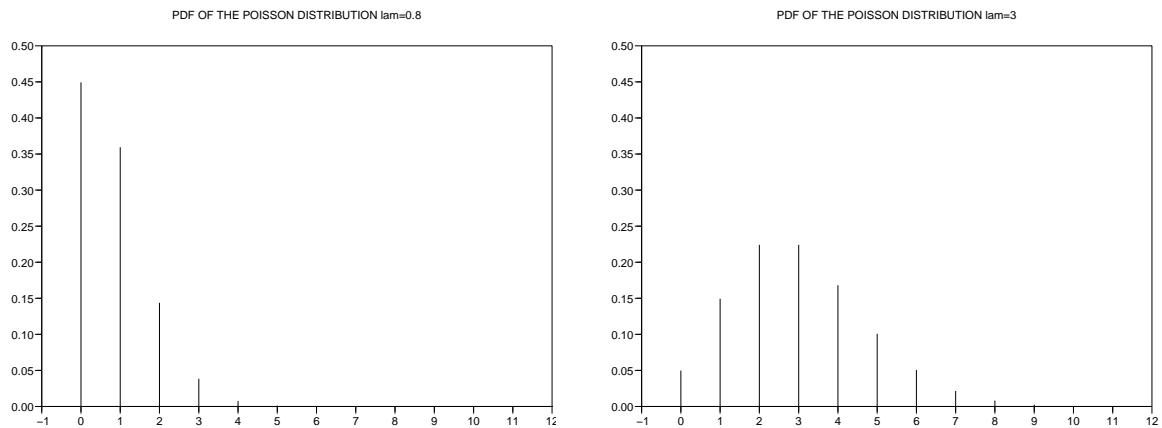


Figure 36: Example of function `pdfpoisson`

## 8.27 `pdfrng` – normal range pdf

### Calling Sequence

```
Y=pdfrng(X,n)
Y=pdfrng(X,n,sigma)
```

### Parameters

- **X**, **Y** : real matrices **X** and **Y**.
- **n** : parameter  $n$  of the normal range distribution. Must be an integer  $\geq 2$ .

- **sigma** : parameter  $\sigma > 0$  (standard-deviation) of the normal distribution. Default is 1.

### Description

Compute in matrix **Y** the pdf of the range distribution for each entry  $X_{i,j}$  of matrix **X**. **pdfrng**(**X**,**n**) is equivalent to **pdfrng**(**X**,**n**,1).

### Examples (see Figure 37)

```
X=linspace(0,7,300)';
Y1=pdfrng(X,3);
xset("window",0);xbasc(0)
plot2d(X,Y1,1,rect=[0,0,7,0.5],nax=[1,8,0,11])
xtitle("PDF OF THE NORMAL RANGE n=3, sigma=1");xselect()
//
Y2=pdfrng(X,5,1.5);
xset("window",1);xbasc(1)
plot2d(X,Y2,1,rect=[0,0,7,0.5],nax=[1,8,0,11])
xtitle("PDF OF THE NORMAL RANGE n=5, sigma=1.5");xselect()
```

### See Also

**cdfrng**

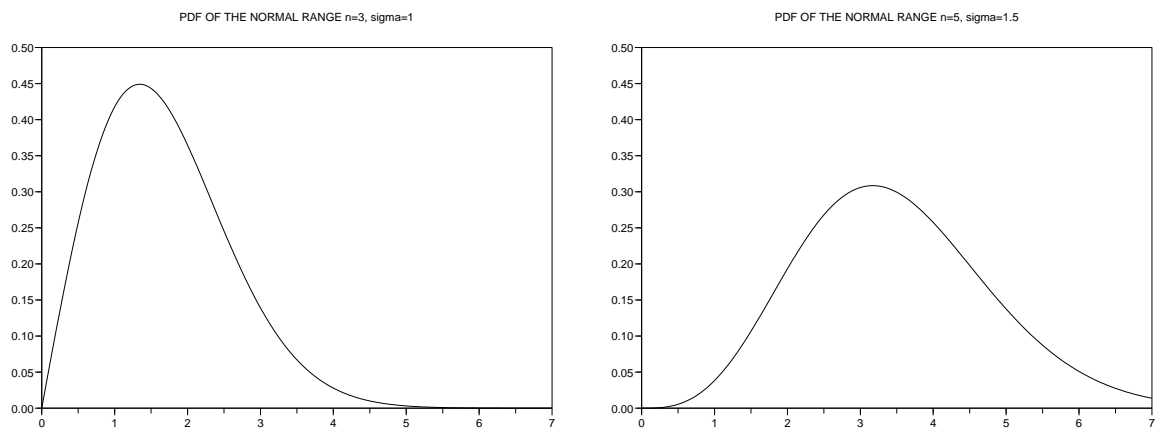


Figure 37: Example of function **pdfrng**

## 8.28 pdfstandev – normal sample standard-deviation pdf

### Calling Sequence

```
Y=pdfstandev(X,n)
Y=pdfstandev(X,n,sigma)
```

### Parameters

- **X**,**Y** : real matrices **X** and **Y**.
- **n** : parameter  $n$  of the normal sample standard-deviation distribution. Must be an integer  $\geq 2$ .
- **sigma** : parameter  $\sigma > 0$  (standard-deviation) of the normal distribution. Default is 1.

### Description

Compute in matrix **Y** the pdf of the normal sample standard-deviation distribution for each entry  $X_{i,j}$  of matrix **X**. **pdfstandev**(**X**,**n**) is equivalent to **pdfstandev**(**X**,**n**,1).

### Examples (see Figure 38)

```
X=linspace(0,7,300)';
Y1=pdfstandev(X,3);
xset("window",0);xbasc(0)
plot2d(X,Y1,1,rect=[0,0,7,0.9],nax=[1,8,1,10])
```

```

xtitle("PDF OF THE NORMAL SAMPLE STANDARD-DEVIATION n=3, sigma=1");xselect()
//
Y2=pdfstandev(X,9,3.5);
xset("window",1);xbasc(1)
plot2d(X,Y2,1,rect=[0,0,7,0.9],nax=[1,8,1,10])
xtitle("PDF OF THE NORMAL SAMPLE STANDARD-DEVIATION n=9, sigma=3.5");xselect()

```

See Also

`cdfstandev`, `idfstandev`, `rndstandev`

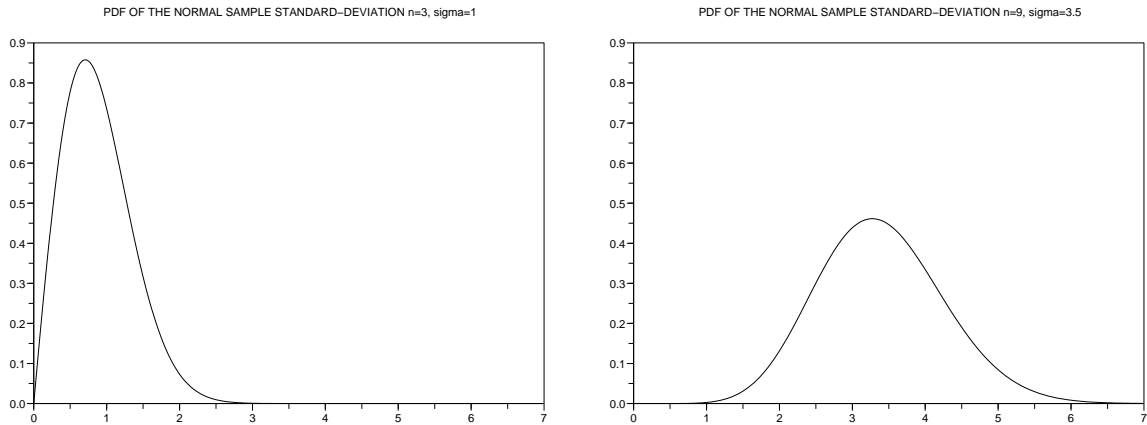


Figure 38: Example of function `pdfstandev`

## 8.29 pdfstudent – Student pdf

### Calling Sequence

```

Y=pdfstudent(X,n)
Y=pdfstudent(X,n,nc)

```

### Parameters

- **X**, **Y** : real matrices **X** and **Y**.
- **n** : parameter  $n$  of the Student distribution. Must be an integer  $\geq 1$ .
- **nc** : noncentrality parameter. Must be  $\geq 0$ . Default is 0.

### Description

Compute in matrix **Y** the pdf of the Student distribution for each entry  $X_{i,j}$  of matrix **X**. `pdfstudent(X,n)` is equivalent to `pdfstudent(X,n,0)`.

### Examples (see Figure 39)

```

X=linspace(-4,9,300)';
Y1=pdfstudent(X,2);
xset("window",0);xbasc(0);plot2d(X,Y1,1)
xtitle("PDF OF THE STUDENT DISTRIBUTION n=2, nc=0");xselect()
//
Y2=pdfstudent(X,20,3);
xset("window",1);xbasc(1);plot2d(X,Y2,1)
xtitle("PDF OF THE STUDENT DISTRIBUTION n=20, nc=3");xselect()

```

See Also

`cdfstudent`, `idfstudent`

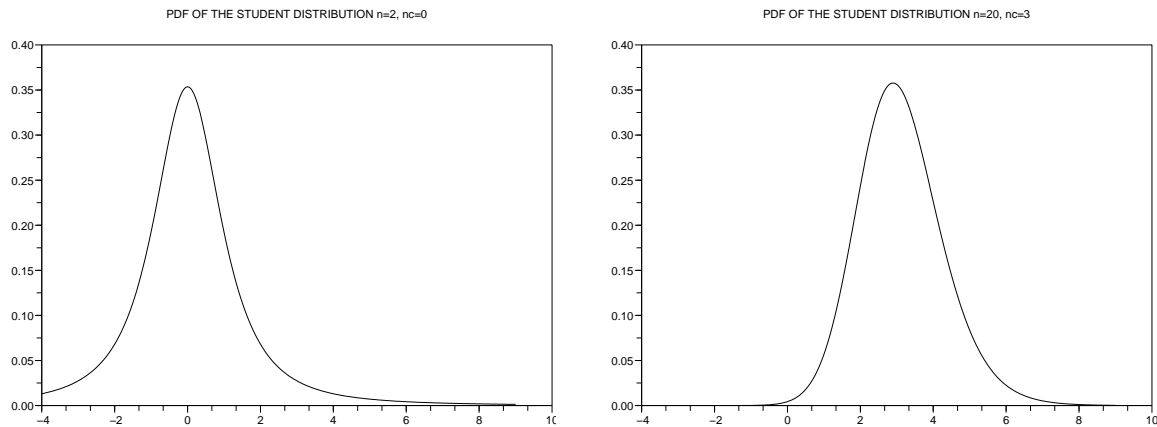


Figure 39: Example of function `pdfstudent`

### 8.30 pdfweibull – Weibull pdf

#### Calling Sequence

`Y=pdfweibull(X,a,b=,c=)`

#### Parameters

- `X,Y` : real matrices **X** and **Y**.
- `a` : parameter  $a > 0$  of the Weibull distribution.
- `b` : parameter  $b > 0$  of the Weibull distribution. Default is 1.
- `c` : parameter  $c$  of the Weibull distribution. Default is 0.

#### Description

Compute in matrix **Y** the pdf of the Weibull distribution for each entry  $X_{i,j}$  of matrix **X**. The Weibull distribution is defined on  $[c, +\infty)$ . `pdfweibull(x,a)` is equivalent to `pdfweibull(x,a,1,0)`.

#### Examples (see Figure 40)

```
X=linspace(-1,7,300)';
Y1=pdfweibull(X,2);
xset("window",0);xbasc(0)
plot2d(X,Y1,1,rect=[-1,0,7,0.9])
xtitle("PDF OF THE WEIBULL DISTRIBUTION a=2, b=1, c=0");xselect()
//
Y2=pdfweibull(X,5,3,2);
xset("window",1);xbasc(1)
plot2d(X,Y2,1,rect=[-1,0,7,0.9])
xtitle("PDF OF THE WEIBULL DISTRIBUTION a=5, b=3, c=2");xselect()
```

#### See Also

`cdfweibull`, `fitweibull`, `idfweibull`, `rndweibull`

## 9 QUADRATURES

### 9.1 quadhermite – Gauss-Hermite quadrature

#### Calling Sequence

`[x,w]=quadhermite(n)`

#### Parameters

- `n` : number  $n$  of Gauss-Hermite quadrature points. Must be 7, 15, 31, 63, 127, 255 or 511.
- `x` : abscissas of the  $n$ -points Gauss-Hermite quadrature.
- `w` : weights of the  $n$ -points Gauss-Hermite quadrature.



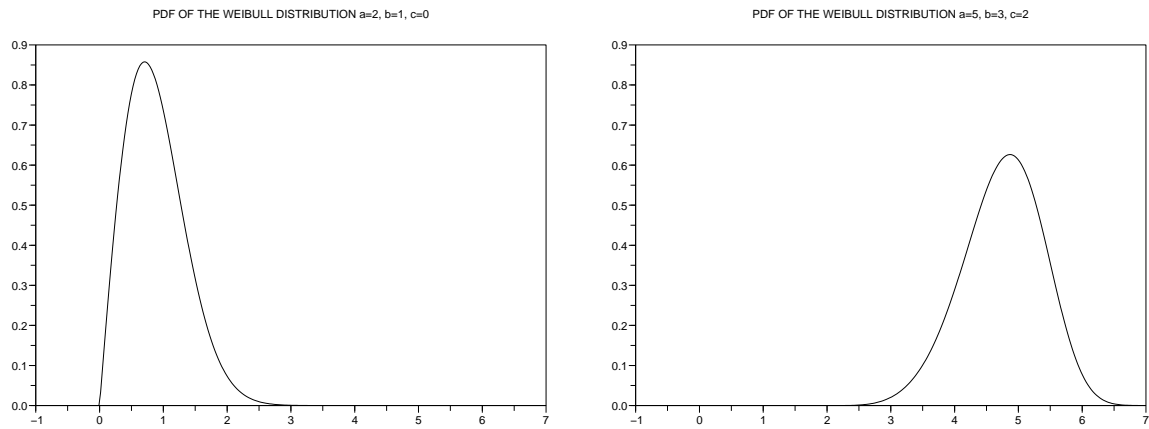


Figure 40: Example of function `pdfweibull`

## Description

Compute the abscissas and weights for the  $n$ -points Gauss-Hermite quadrature.

## Examples

```
[x,w]=quadhermite(31);
y=pdfnormal(x);
sum(y.*w)
```

## See Also

`quadlaguerre`, `quadlegendre`, `quadsimpson`

## 9.2 quadlaguerre – Gauss-Laguerre quadrature

### Calling Sequence

```
[x,w]=quadlaguerre(n)
[x,w]=quadlaguerre(n,a)
```

### Parameters

- **n** : number  $n$  of Gauss-Laguerre quadrature points. Must be 7, 15, 31, 63, 127.
- **a** : left bound  $a$  of the Gauss-Laguerre quadrature. Default is 0.
- **x** : abscissas of the  $n$ -points Gauss-Laguerre quadrature.
- **w** : weights of the  $n$ -points Gauss-Laguerre quadrature.

## Description

Compute the abscissas and weights for the  $n$ -points Gauss-Laguerre quadrature.

## Examples

```
[x,w]=quadlaguerre(15);
y=pdfgamma(x,3);
sum(y.*w)
[x,w]=quadlaguerre(15,2);
y=pdfgamma(x,3);
[sum(y.*w),1-cdfgamma(2,3)]
```

## See Also

`quadhermite`, `quadlegendre`, `quadsimpson`

## 9.3 quadlegendre – Gauss-Legendre quadrature

### Calling Sequence

```
[x,w]=quadlegendre(n,a,b=)
```

#### Parameters

- **n** : number  $n$  of Gauss-Legendre quadrature points. Must be 7, 15, 31, 63, 127, 255 or 511.
- **a** : left bound  $a$  of the Gauss-Legendre quadrature. Default is -1.
- **b** : right bound  $b$  of the Gauss-Legendre quadrature. Default is +1.
- **x** : abscissas of the  $n$ -points Gauss-Legendre quadrature.
- **w** : weights of the  $n$ -points Gauss-Legendre quadrature.

#### Description

Compute the abscissas and weights for the  $n$ -points Gauss-Legendre quadrature.

#### Examples

```
[x,w]=quadlegendre(15,2,3);  
y=exp(-x);  
sum(y.*w)  
[x,w]=quadlegendre(31,0,%pi/2);  
y=cos(x);  
sum(y.*w)
```

#### See Also

quadhermite, quadlaguerre, quadsimpson

## 9.4 quadsimpson – Simpson quadrature

#### Calling Sequence

```
[x,w]=quadsimpson(n,a,b)
```

#### Parameters

- **n** : number  $n$  of Simpson quadrature points. Must be an odd integer  $\geq 3$ .
- **a** : left bound  $a$  of the Simpson quadrature.
- **b** : right bound  $b$  of the Simpson quadrature.
- **x** : abscissas of the  $n$ -points Simpson quadrature.
- **w** : weights of the  $n$ -points Simpson quadrature.

#### Description

Compute the abscissas and weights for the  $n$ -points Simpson quadrature.

#### Examples

```
[x,w]=quadsimpson(101,2,3);  
y=exp(-x);  
sum(y.*w)  
[x,w]=quadsimpson(201,0,%pi/2);  
y=cos(x);  
sum(y.*w)
```

#### See Also

quadhermite, quadlaguerre, quadlegendre

## 10 RANDOM NUMBER GENERATORS

### 10.1 rndbeta – beta type 1 random number generator

#### Calling Sequence

```
X=rndbeta(row,a,b,c=,d=)  
X=rndbeta([row,col],a,b,c=,d=)
```

#### Parameters

- **X** : real matrix **X**.
- **row** : number  $r$  of rows of matrix **X**. Must be an integer  $\geq 1$ .

- `col` : number  $c$  of columns of matrix  $\mathbf{X}$ . Must be an integer  $\geq 1$ . Default is 1.
- `a` : parameter  $a > 0$  of the beta type 1 distribution.
- `b` : parameter  $b > 0$  of the beta type 1 distribution.
- `c` : parameter  $c$  of the beta type 1 distribution. Default is 0.
- `d` : parameter  $d > 0$  of the beta type 1 distribution. Default is 1.

### Description

Generate a  $(r, c)$  matrix  $\mathbf{X}$  of beta type 1 random numbers. The beta type 1 distribution is defined on  $[c, c + d]$ . `rndbeta(row,a,b)` is equivalent to `rndbeta([row,1],a,b,0,1)`.

### Examples (see Figure 41)

```
X=linspace(-1,2,300)';
X1=rndbeta(900,2,5);
Y1=pdfbeta(X,2,5);
xset("window",0);xbasc(0);plot2d(X,Y1,5);histplot(30,X1,1,rect=[-1,0,2,2.5])
xtitle("RANDOM NUMBERS FOR THE BETA TYPE 1 DISTRIBUTION a=2, b=5, c=0, d=1")
xselect()
//
X2=rndbeta(900,5,2,-0.5,2.5);
Y2=pdfbeta(X,5,2,-0.5,2.5);
xset("window",1);xbasc(1);plot2d(X,Y2,5);histplot(30,X2,1,rect=[-1,0,2,2.5])
xtitle("RANDOM NUMBERS FOR THE BETA TYPE 1 DISTRIBUTION a=5, b=2, c=-0.5, d=2.5")
xselect()
```

### See Also

`cdfbeta`, `fitbeta`, `idfbeta`, `pdfbeta`

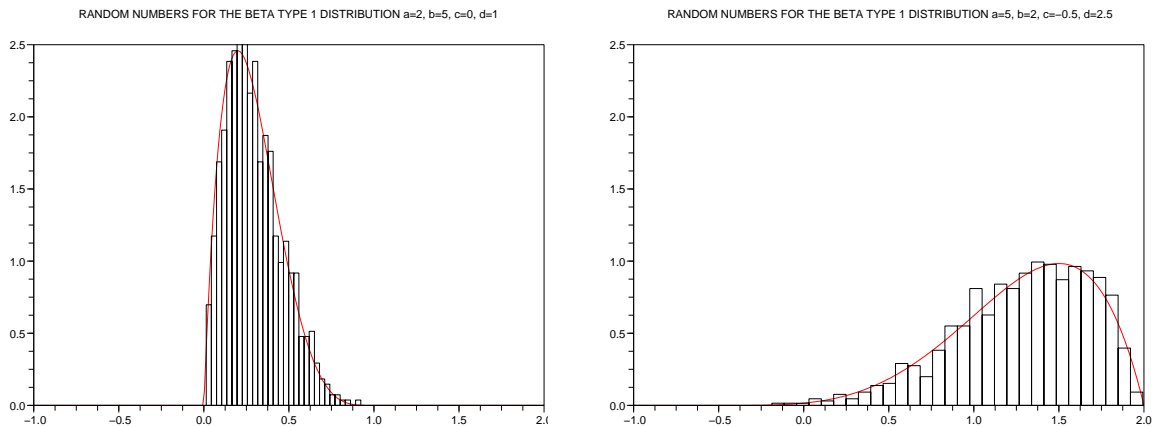


Figure 41: Example of function `rndbeta`

## 10.2 rndbeta2 – beta type 2 random number generator

### Calling Sequence

```
X=rndbeta2(row,a,b,c=,d=)
X=rndbeta2([row,col],a,b,c=,d=)
```

### Parameters

- `X` : real matrix  $\mathbf{X}$ .
- `row` : number  $r$  of rows of matrix  $\mathbf{X}$ . Must be an integer  $\geq 1$ .
- `col` : number  $c$  of columns of matrix  $\mathbf{X}$ . Must be an integer  $\geq 1$ . Default is 1.
- `a` : parameter  $a > 0$  of the beta type 2 distribution.
- `b` : parameter  $b > 0$  of the beta type 2 distribution.
- `c` : parameter  $c$  of the beta type 2 distribution. Default is 0.

- $d$  : parameter  $d > 0$  of the beta type 2 distribution. Default is 1.

### Description

Generate a  $(r, c)$  matrix  $\mathbf{X}$  of beta type 2 random numbers. The beta type 2 distribution is defined on  $[c, +\infty)$ . `rndbeta2(row,a,b)` is equivalent to `rndbeta2([row,1],a,b,0,1)`.

### Examples (see Figure 42)

```
X=linspace(-1,10,300)';
X1=rndbeta2(900,2,5);
Y1=pdfbeta2(X,2,5);
xset("window",0);xbasc(0);plot2d(X,Y1,5);histplot(30,X1,1,rect=[-2,0,10,3])
xtitle("RANDOM NUMBERS FOR THE BETA TYPE 2 DISTRIBUTION a=2, b=5, c=0, d=1")
xselect()
//
X2=rndbeta2(900,5,2,-0.5,0.1);
Y2=pdfbeta2(X,5,2,-0.5,0.1);
xset("window",1);xbasc(1);plot2d(X,Y2,5);histplot(30,X2,1,rect=[-2,0,10,3])
xtitle("RANDOM NUMBERS FOR THE BETA TYPE 2 DISTRIBUTION a=5, b=2, c=-0.5, d=0.1")
xselect()
```

### See Also

`cdfbeta2`, `idfbeta2`, `pdfbeta2`

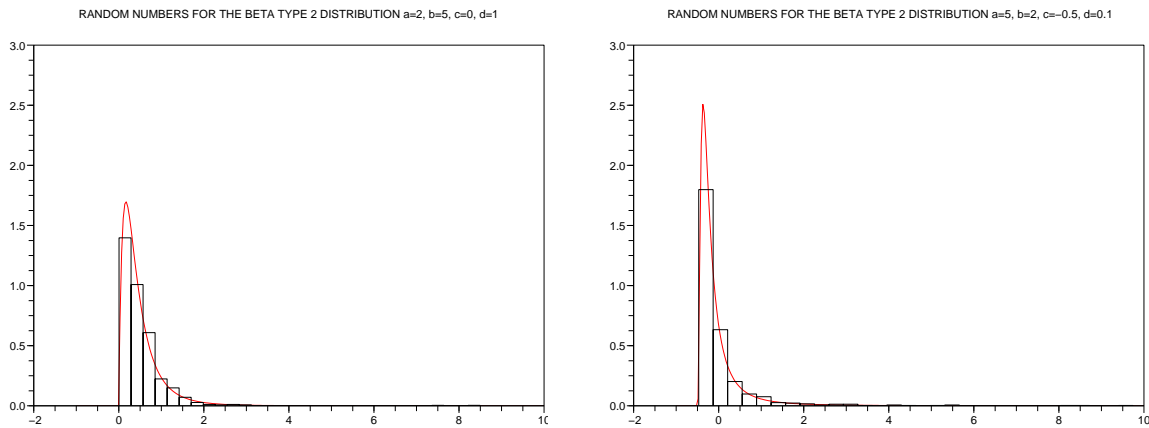


Figure 42: Example of function `rndbeta2`

## 10.3 rndbinomial – binomial random number generator

### Calling Sequence

```
X=rndbinomial(row,n,p)
X=rndbinomial([row,col],n,p)
```

### Parameters

- $\mathbf{X}$  : real matrix  $\mathbf{X}$ .
- `row` : number  $r$  of rows of matrix  $\mathbf{X}$ . Must be an integer  $\geq 1$ .
- `col` : number  $c$  of columns of matrix  $\mathbf{X}$ . Must be an integer  $\geq 1$ . Default is 1.
- $n$  : parameter  $n$  of the binomial distribution. Must be an integer  $\geq 1$ .
- $p$  : parameter  $p \in [0, 1]$  of the binomial distribution.

### Description

Generate a  $(r, c)$  matrix  $\mathbf{X}$  of binomial random numbers. `rndbinomial(row,n,p)` is equivalent to `rndbinomial([row,1],n,p)`.

### Examples (see Figure 43)

```

X=(-1:20)';
X1=rndbinomial(900,20,0.2);
for z=-1:20, Y1(z+2)=length(find(X1==z))/900; end
Z1=pdfbinomial(X,20,0.2);
xset("window",0);xasc(0)
plot2d(X,Y1,-2);plot2d3(X,Z1,5,rect=[-1,0,20,0.25],nax=[0,22,0,11])
xtitle("RANDOM NUMBERS FOR THE BINOMIAL DISTRIBUTION n=20, p=0.2")
xselect()
//
X2=rndbinomial(900,20,0.5);
for z=-1:20, Y2(z+2)=length(find(X2==z))/900; end
Z2=pdfbinomial(X,20,0.5);
xset("window",1);xasc(1)
plot2d(X,Y2,-2);plot2d3(X,Z2,5,rect=[-1,0,20,0.25],nax=[0,22,0,11])
xtitle("RANDOM NUMBERS FOR THE BINOMIAL DISTRIBUTION n=20, p=0.5")
xselect()

```

See Also

cdfbinomial, pdfbinomial

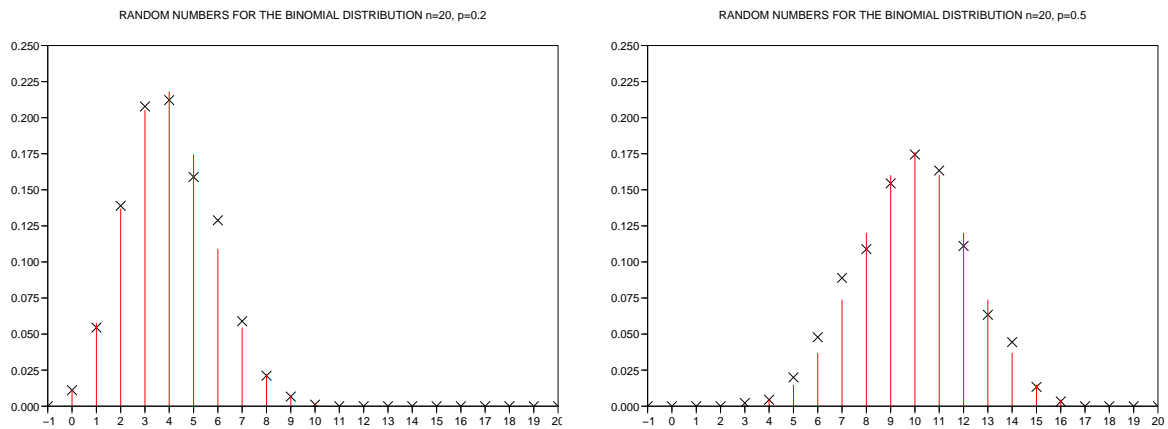


Figure 43: Example of function `rndbinomial`

## 10.4 `rndexponential` – exponential random number generator

### Calling Sequence

```

X=rndexponential(row,lam)
X=rndexponential([row,col],lam)

```

### Parameters

- **X** : real matrix **X**.
- **row** : number  $r$  of rows of matrix **X**. Must be an integer  $\geq 1$ .
- **col** : number  $c$  of columns of matrix **X**. Must be an integer  $\geq 1$ . Default is 1.
- **lam** : parameter  $\lambda > 0$  of the exponential distribution.

### Description

Generate a  $(r, c)$  matrix **X** of exponential random numbers. `rndexponential(row,lam)` is equivalent to `rndexponential([row,1],lam)`.

### Examples (see Figure 44)

```

X=linspace(-1,6,300)';
X1=rndexponential(900,0.5);
Y1=pdfexponential(X,0.5);

```

```

xset("window",0);xbasc(0);plot2d(X,Y1,5);histplot(30,X1,1,rect=[-1,0,6,2])
xtitle("RANDOM NUMBERS FOR THE EXPONENTIAL DISTRIBUTION lam=0.5")
xselect()
//
X2=rndexponential(900,2);
Y2=pdfexponential(X,2);
xset("window",1);xbasc(1);plot2d(X,Y2,5);histplot(30,X2,1,rect=[-1,0,6,2])
xtitle("RANDOM NUMBERS FOR THE EXPONENTIAL DISTRIBUTION lam=2")
xselect()

```

See Also

`cdfexponential`, `idfexponential`, `pdfexponential`

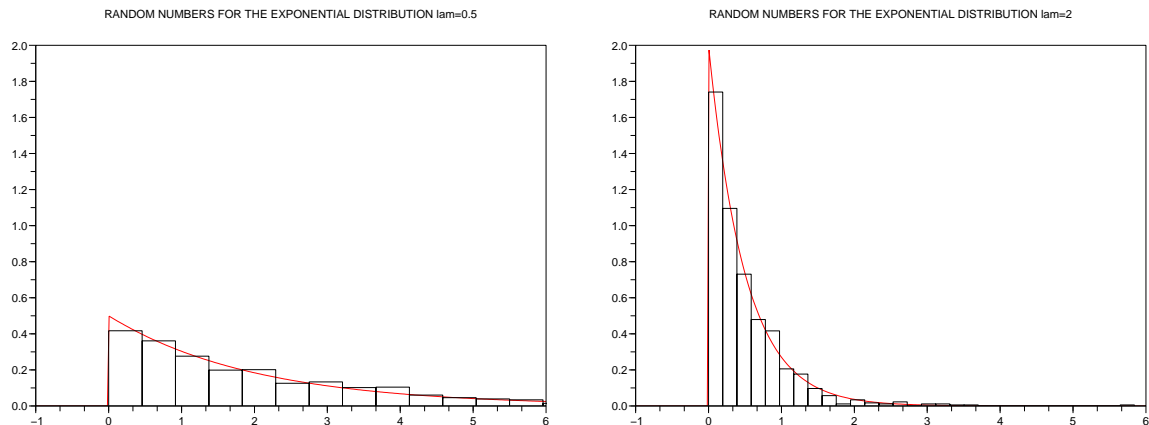


Figure 44: Example of function `rndexponential`

## 10.5 `rndfoldednormal` – folded normal random number generator

### Calling Sequence

```

X=rndfoldednormal(row,mu=sigma,c=)
X=rndfoldednormal([row,col],mu=sigma,c=)

```

### Parameters

- **X** : real matrix **X**.
- **row** : number  $r$  of rows of matrix **X**. Must be an integer  $\geq 1$ .
- **col** : number  $c$  of columns of matrix **X**. Must be an integer  $\geq 1$ . Default is 1.
- **mu** : parameter  $\mu$  (mean) of the folded normal distribution. Default is 0.
- **sigma** : parameter  $\sigma > 0$  (standard-deviation) of the folded normal distribution. Default is 1.
- **c** : parameter  $c$  of the folded normal distribution. Default is 0.

### Description

Generate a  $(r, c)$  matrix **X** of folded normal random numbers. `rndfoldednormal(row)` is equivalent to `rndfoldednormal([row,1],0,1,0)`.

### Examples (see Figure 45)

```

X=linspace(-1,7,300)';
X1=rndfoldednormal(900);
Y1=pdffoldednormal(X);
xset("window",0);xbasc(0)
plot2d(X,Y1,5);histplot(30,X1,1,rect=[-1,0,7,0.8])
xtitle("RANDOM NUMBERS FOR THE FOLDED NORMAL DISTRIBUTION mu=0, sigma=1, c=0")
xselect()
//

```

```

X2=rndfoldednormal(900,2,1.5,1);
Y2=pdf foldednormal(X,2,1.5,1);
xset("window",1);xbasc(1)
plot2d(X,Y2,5);histplot(30,X2,1,rect=[-1,0,7,0.8])
xtitle("RANDOM NUMBERS FOR THE FOLDEDNORMAL DISTRIBUTION mu=2, sigma=1.5, c=1")
xselect()

```

See Also

`cdf foldednormal`, `idf foldednormal`, `pdf foldednormal`

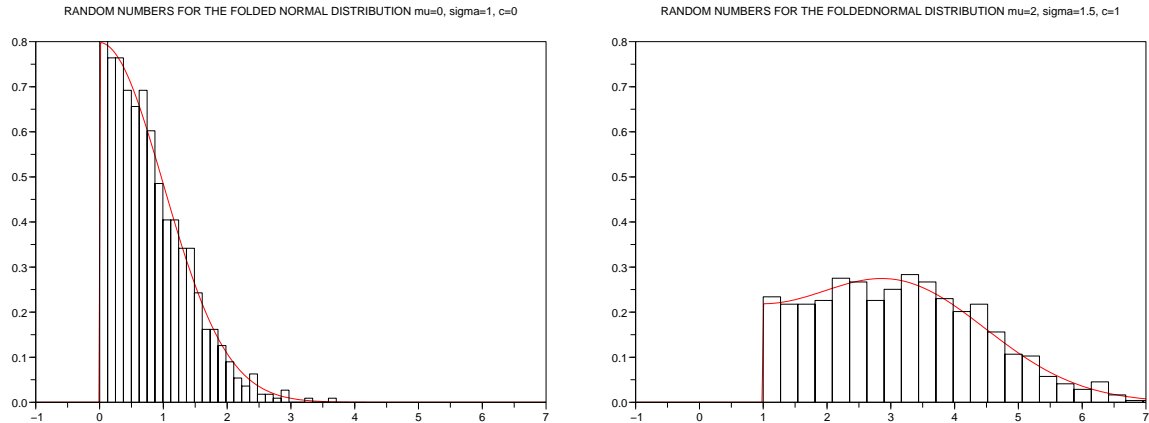


Figure 45: Example of function `rndfoldednormal`

## 10.6 `rndgamma` – gamma random number generator

### Calling Sequence

```

X=rndgamma(row,a,b=,c=,d=)
X=rndgamma([row,col],a,b=,c=,d=)

```

### Parameters

- **X** : real matrix **X**.
- **row** : number  $r$  of rows of matrix **X**. Must be an integer  $\geq 1$ .
- **col** : number  $c$  of columns of matrix **X**. Must be an integer  $\geq 1$ . Default is 1.
- **a** : parameter  $a > 0$  of the gamma distribution.
- **b** : parameter  $b > 0$  of the gamma distribution. Default is 1.
- **c** : parameter  $c$  of the gamma distribution. Default is 0.
- **d** : parameter  $d \neq 0$  of the gamma distribution. Default is 1.

### Description

Generate a  $(r, c)$  matrix **X** of gamma  $(a, b, c, d)$  random numbers. The gamma  $(a, b, c, d)$  distribution is defined on  $[c, +\infty[$ . `rndgamma(row,a)` is equivalent to `rndgamma([row,1],a,1,0)`.

### Examples (see Figure 46)

```

X=linspace(-1,10,300)';
X1=rndgamma(900,2);
Y1=pdfgamma(X,2);
xset("window",0);xbasc(0);plot2d(X,Y1,5)
histplot(30,X1,1,rect=[-1,0,10,0.7],nax=[1,12,1,8])
xtitle("RANDOM NUMBERS FOR THE GAMMA DISTRIBUTION a=2, b=1, c=0, d=1")
xselect()
//
X2=rndgamma(900,5,0.7,2,1.5);
Y2=pdfgamma(X,5,0.7,2,1.5);

```

```
xset("window",1);xbasc(1);plot2d(X,Y2,5)
histplot(30,X2,1,rect=[-1,0,10,0.7],nax=[1,12,1,8])
xtitle("RANDOM NUMBERS FOR THE GAMMA DISTRIBUTION a=5, b=0.7, c=2, d=1.5")
xselect()
```

See Also

cdfgamma, fitgamma, idfgamma, pdfgamma

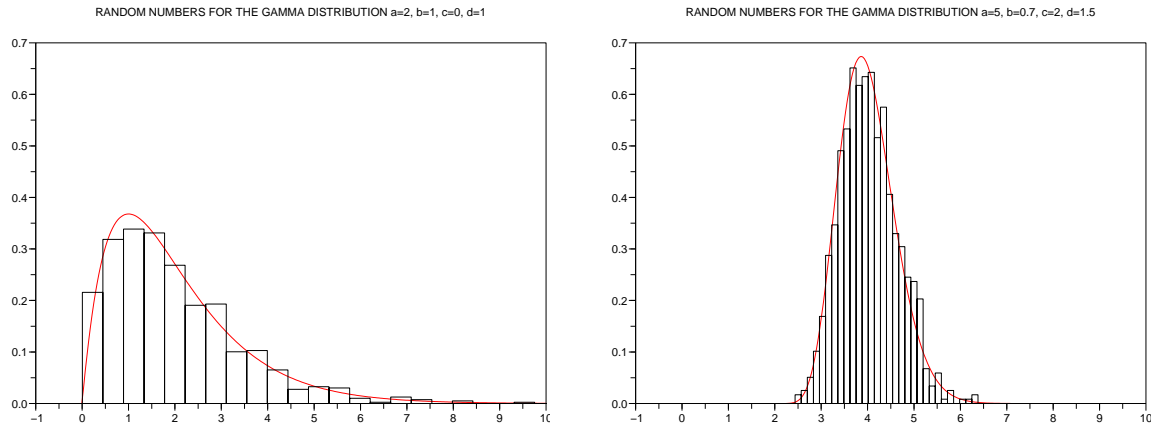


Figure 46: Example of function `rndgamma`

## 10.7 `rndgev` – generalized Extreme Value random number generator

### Calling Sequence

```
X=rndgev(row,a,b,c=)
X=rndgev([row,col],a,b,c=)
```

### Parameters

- **X** : real matrices **X**.
- **row** : number  $r$  of rows of matrix **X**. Must be an integer  $\geq 1$ .
- **col** : number  $c$  of columns of matrix **X**. Must be an integer  $\geq 1$ . Default is 1.
- **a** : parameter  $a$  of the GEV distribution.
- **b** : parameter  $b > 0$  of the GEV distribution. Default is 1.
- **c** : parameter  $c$  of the GEV distribution. Default is 0.

### Description

Generate a  $(r, c)$  matrix **X** of GEV  $(a, b, c)$  random numbers. `rndgev(row,a)` is equivalent to `rndgev([row,1],a,1,0)`.

### Examples (see Figure 47)

```
X=linspace(-1,7,300)';
X1=rndgev(900,0.5);
Y1=pdfgev(X,0.5);
xset("window",0);xbasc(0);plot2d(X,Y1,5)
histplot(30,X1,1)
xtitle("RANDOM NUMBERS FOR THE GEV DISTRIBUTION a=0.5, b=1, c=0")
xselect()
//
X2=rndgev(900,-0.5,c=5);
Y2=pdfgev(X,-0.5,c=5);
xset("window",1);xbasc(1);plot2d(X,Y2,5)
histplot(30,X2,1)
xtitle("RANDOM NUMBERS FOR THE GEV DISTRIBUTION a=-0.5, b=1, c=5")
xselect()
```



See Also

`cdfgev`, `fitgev`, `idfgev`, `pdfgev`

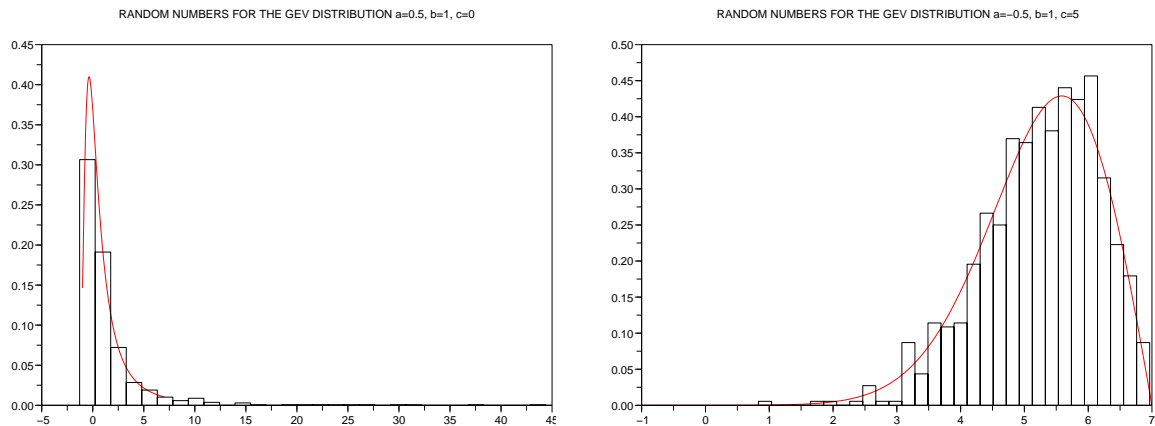


Figure 47: Example of function `rndgev`

## 10.8 `rndjohnson` – Johnson’s random number generator

### Calling Sequence

```
X=rndjohnson(row,s,a,b,c,d)
x=rndjohnson([row,col],s,a,b,c,d)
```

### Parameters

- **X** : real matrix **X**.
- **row** : number  $r$  of rows of matrix **X**. Must be an integer  $\geq 1$ .
- **col** : number  $c$  of columns of matrix **X**. Must be an integer  $\geq 1$ . Default is 1.
- **s** : Johnson’s system of distribution. Must be "B" for the Johnson’s  $S_B$  (bounded) system of distributions or "U" for the Johnson’s  $S_U$  (unbounded) system of distributions.
- **a** : parameter  $a$  of the Johnson’s distribution.
- **b** : parameter  $b > 0$  of the Johnson’s distribution.
- **c** : parameter  $c$  of the Johnson’s distribution.
- **d** : parameter  $d > 0$  of the Johnson’s distribution.

### Description

Generate a  $(r, c)$  matrix **X** of Johnson’s random numbers. `rndjohnson(row,s,a,b,c,d)` is equivalent to `rndjohnson([row,1],s,a,b,c,d)`.

### Examples (see Figure 48)

```
X=linspace(0,6,300)';
X1=rndjohnson(900,"B",4,3,1,5);
Y1=pdfjohnson(X,"B",4,3,1,5);
xset("window",0);xbasc(0);plot2d(X,Y1,5);histplot(30,X1,1,rect=[0,0,6,1.5])
xtitle("RANDOM NUMBERS FOR THE JOHNSON'S BOUNDED DISTRIBUTION a=4, b=3, c=1, d=5")
xselect()
//
X2=rndjohnson(900,"U",3,4,5,2);
Y2=pdfjohnson(X,"U",3,4,5,2);
xset("window",1);xbasc(1);plot2d(X,Y2,5);histplot(30,X2,1,rect=[0,0,6,1.5])
xtitle("RANDOM NUMBERS FOR THE JOHNSON'S UNBOUNDED DISTRIBUTION a=3, b=4, c=5, d=2")
xselect()
```

See Also

`cdfjohnson`, `fitjohnson`, `idfjohnson`, `pdfjohnson`

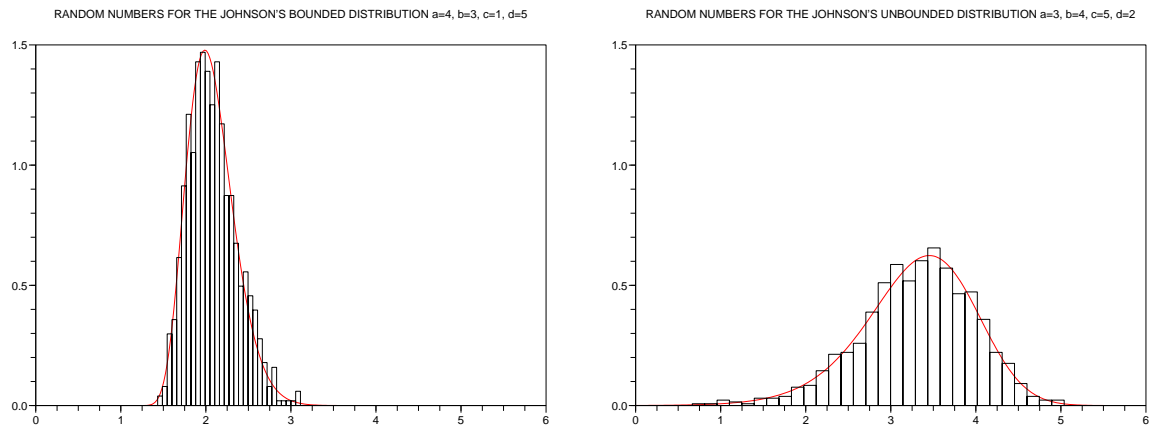


Figure 48: Example of function `rndjohnson`

## 10.9 `rndlognormal` – lognormal random number generator

### Calling Sequence

```
X=rndlognormal(row,a=b,c=)
X=rndlognormal([row,col],a=b,c=)
```

### Parameters

- **X** : real matrix **X**.
- **row** : number  $r$  of rows of matrix **X**. Must be an integer  $\geq 1$ .
- **col** : number  $c$  of columns of matrix **X**. Must be an integer  $\geq 1$ . Default is 1.
- **a** : parameter  $a$  of the lognormal distribution. Default is 0.
- **b** : parameter  $b > 0$  of the lognormal distribution. Default is 1.
- **c** : parameter  $c$  of the lognormal distribution. Default is 0.

### Description

Generate a  $(r, c)$  matrix **X** of lognormal random numbers. The lognormal distribution is defined on  $[c, +\infty)$ . `rndlognormal(row)` is equivalent to `rndlognormal([row,1],0,1,0)`.

### Examples (see Figure 49)

```
X=linspace(-1,7,300)';
X1=rndlognormal(900,0.5,2);
Y1=pdflognormal(X,0.5,2);
xset("window",0);xbasc(0);plot2d(X,Y1,5)
histplot(30,X1,1,rect=[-1,0,7,1.2],nax=[1,9,1,7])
xtitle("RANDOM NUMBERS FOR THE LOGNORMAL DISTRIBUTION a=0.5, b=2, c=0")
xselect()
//
X2=rndlognormal(900,-0.5,c=0.5);
Y2=pdflognormal(X,-0.5,c=0.5);
xset("window",1);xbasc(1);plot2d(X,Y2,5)
histplot(30,X2,1,rect=[-1,0,7,1.2],nax=[1,9,1,7])
xtitle("RANDOM NUMBERS FOR THE LOGNORMAL DISTRIBUTION a=-0.5, b=1, c=0.5")
xselect()
```

### See Also

`cdflognormal`, `fitlognormal`, `idflognormal`, `pdflognormal`

## 10.10 `rndmultinormal` – multinormal random number generator

### Calling Sequence

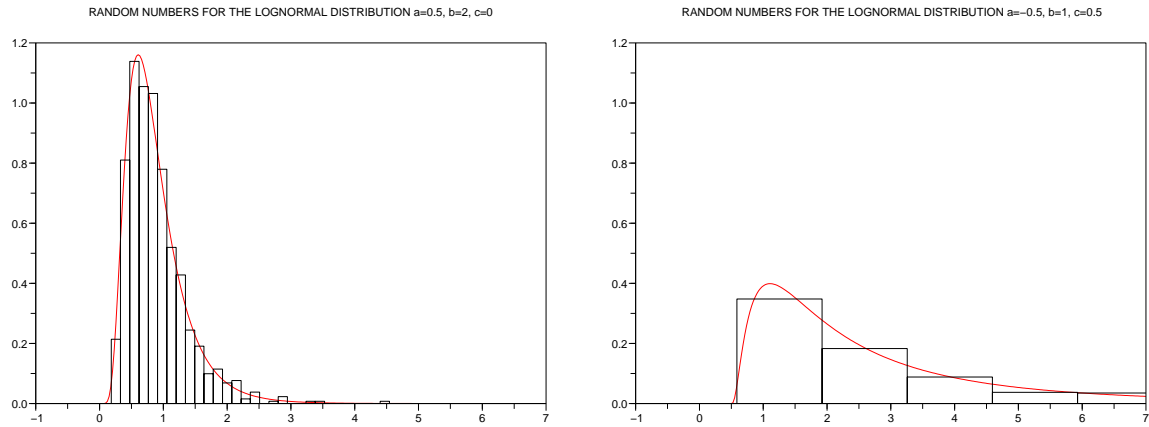


Figure 49: Example of function `rndlognormal`

```
X=rndmultinormal(n,mu)
X=rndmultinormal(n,mu,sigma)
```

#### Parameters

- **X** : real matrix **X**.
- **n** : number of multinormal row random vectors. Must be an integer  $\geq 1$ .
- **mu** : mean vector  $\boldsymbol{\mu}$  of the multinormal distribution. Must be a  $(1, p)$  row vector.
- **sigma** : variance-covariance matrix  $\boldsymbol{\Sigma}$  of the multinormal distribution. Must be a  $(p, p)$  definite positive matrix or  $(1, p)$  row vector  $(\sigma_1^2, \dots, \sigma_p^2)$  where  $\sigma_1^2, \dots, \sigma_p^2$  are the diagonal elements (variance) of matrix  $\boldsymbol{\Sigma}$ . Default is `eye(p,p)`.

#### Description

Generate a  $(n, p)$  matrix **X** of  $n$  multinormal row random vectors. `rndmultinormal(n,mu)` is equivalent to `rndmultinormal(n,mu,eye(p,p))`.

#### Examples (see Figure 50)

```
X1=rndmultinormal(100,[5,0]);
X2=rndmultinormal(100,[0,5],[0.7,0.8]);
X3=rndmultinormal(100,[6,6],[0.7,0.01;0.01,0.8]);
xbasc();
plot2d([X1(:,1),X2(:,1),X3(:,1)], [X2(:,1),X2(:,2),X3(:,2)], [-1,-2,-3])
xselect()
```

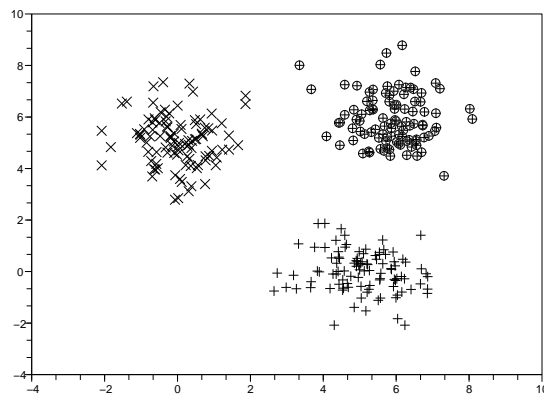


Figure 50: Example of function `rndmultinormal`

## 10.11 rndnormal – normal random number generator

### Calling Sequence

```
X=rndnormal(row,mu,sigma=)
X=rndnormal([row,col],mu,sigma=)
```

### Parameters

- **X** : real matrix **X**.
- **row** : number  $r$  of rows of matrix **X**. Must be an integer  $\geq 1$ .
- **col** : number  $c$  of columns of matrix **X**. Must be an integer  $\geq 1$ . Default is 1.
- **mu** : parameter  $\mu$  (mean) of the normal distribution. Default is 0.
- **sigma** : parameter  $\sigma > 0$  (standard-deviation) of the normal distribution. Default is 1.

### Description

Generate a  $(r, c)$  matrix **X** of normal random numbers. `rndnormal(row)` is equivalent to `rndnormal([row,1],0,1)`.

### Examples (see Figure 51)

```
X=linspace(-4,8,200)';
X1=rndnormal(900);
Y1=pdfnormal(X);
xset("window",0);xbasc(0)
plot2d(X,Y1,5);histplot(30,X1,1,rect=[-4,0,8,0.4],nax=[0,13,1,9])
xtitle("RANDOM NUMBERS FOR THE NORMAL DISTRIBUTION mu=0, sigma=1")
xselect()
//
X2=rndnormal(900,3,2);
Y2=pdfnormal(X,3,2);
xset("window",1);xbasc(1)
plot2d(X,Y2,5);histplot(30,X2,1,rect=[-4,0,8,0.4],nax=[0,13,1,9])
xtitle("RANDOM NUMBERS FOR THE NORMAL DISTRIBUTION mu=3, sigma=2")
xselect()
```

### See Also

`cdfnormal`, `idfnormal`, `pdfnormal`

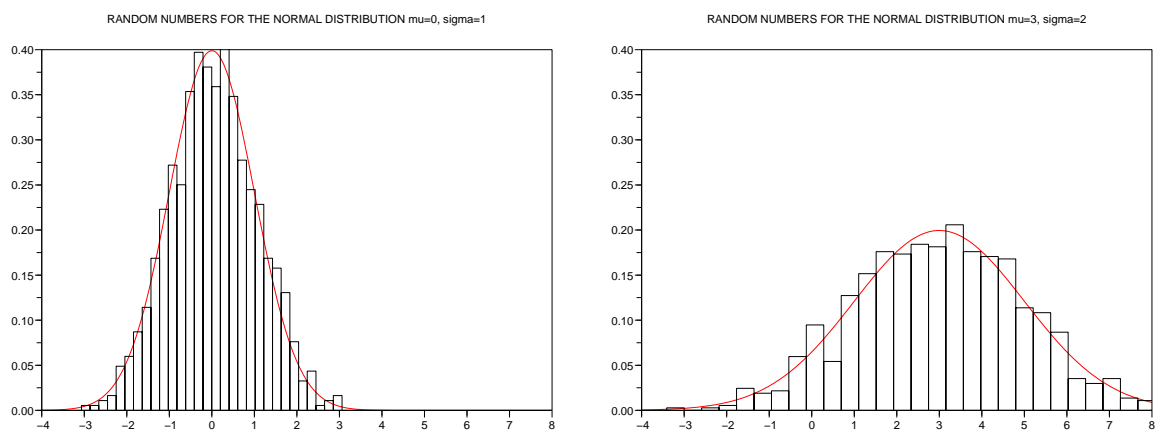


Figure 51: Example of function `rndnormal`

## 10.12 rndpareto – Pareto random number generator

### Calling Sequence

```
X=rndpareto(row,a,b,c=)
X=rndpareto([row,col],a,b,c=)
```

### Parameters

- **X** : real matrix **X**.
- **row** : number  $r$  of rows of matrix **X**. Must be an integer  $\geq 1$ .
- **col** : number  $c$  of columns of matrix **X**. Must be an integer  $\geq 1$ . Default is 1.
- **a** : parameter  $a$  of the Pareto distribution.
- **b** : parameter  $b > 0$  of the Pareto distribution. Default is 1.
- **c** : parameter  $c$  of the Pareto distribution. Default is 0.

### Description

Generate a  $(r, c)$  matrix **X** of Pareto random numbers. The Pareto distribution is defined on

- $[c, +\infty)$  if  $a \geq 0$ ,
- $[c, c - b/a]$  if  $a < 0$ .

`rndpareto(row,a)` is equivalent to `rndpareto(row,1,a,1,0)`.

### Examples (see Figure 52)

```
X=linspace(-1,7,300)';
X1=rndpareto(900,0.5);
Y1=pdfpareto(X,0.5);
xset("window",0);xbasc(0);plot2d(X,Y1,5)
histplot(300,X1,1,rect=[-1,0,7,2],nax=[1,9,1,11])
xtitle("RANDOM NUMBERS FOR THE PARETO DISTRIBUTION a=0.5, b=1, c=0")
xselect()
//
X2=rndpareto(900,-2,6,1);
Y2=pdfpareto(X,-2,6,1);
xset("window",1);xbasc(1);plot2d(X,Y2,5)
histplot(30,X2,1,rect=[-1,0,7,2],nax=[1,9,1,11])
xtitle("RANDOM NUMBERS FOR THE PARETO DISTRIBUTION a=-2, b=6, c=1")
xselect()
```

### See Also

`cdfpareto`, `fitpareto`, `idfpareto`, `pdfpareto`

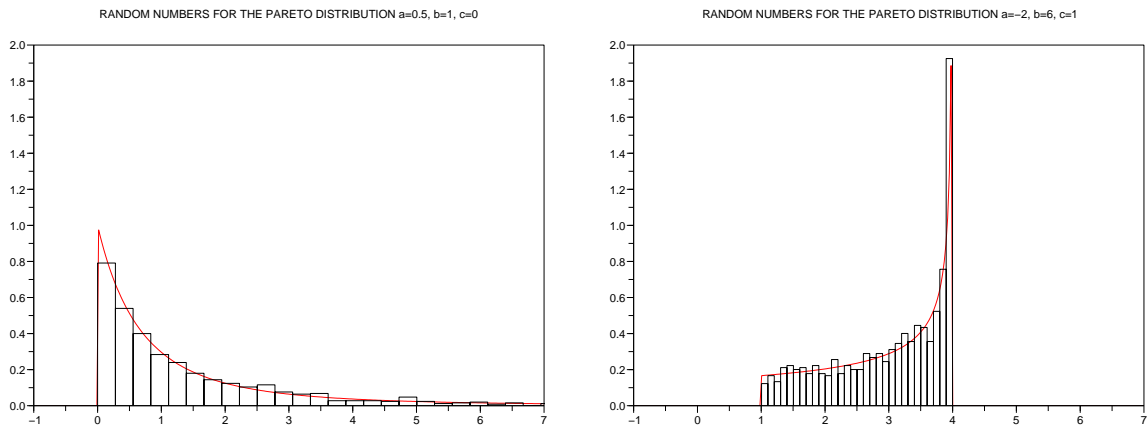


Figure 52: Example of function `rndpareto`

## 10.13 `rndpascal` – Pascal random number generator

### Calling Sequence

```
X=rndpascal(row,n,p)
X=rndpascal([row,col],n,p)
```

### Parameters

- **X** : real matrix **X**.
- **row** : number  $r$  of rows of matrix **X**. Must be an integer  $\geq 1$ .
- **col** : number  $c$  of columns of matrix **X**. Must be an integer  $\geq 1$ . Default is 1.
- **n** : parameter  $n$  of the Pascal distribution. Must be an integer  $\geq 1$ .
- **p** : parameter  $p \in (0, 1]$  of the Pascal distribution.

### Description

Generate a  $(r, c)$  matrix **X** of Pascal random numbers. `rndpascal(row,n,p)` is equivalent to `rndpascal([row,1],n,p)`.

### Examples (see Figure 53)

```
X=(0:30)';
X1=rndpascal(900,2,0.3);
for z=0:30, Y1(z+1)=length(find(X1==z))/900; end
Z1=pdfpascal(X,2,0.3);
xset("window",0);xbasc(0)
plot2d(X,Y1,-2);plot2d3(X,Z1,5,rect=[0,0,30,0.14],nax=[4,7,1,8])
xtitle("RANDOM NUMBERS FOR THE PASCAL DISTRIBUTION n=2, p=0.3")
xselect()
//
X2=rndpascal(900,7,0.5);
for z=0:30, Y2(z+1)=length(find(X2==z))/900; end
Z2=pdfpascal(X,7,0.5);
xset("window",1);xbasc(1)
plot2d(X,Y2,-2);plot2d3(X,Z2,5,rect=[0,0,30,0.14],nax=[4,7,1,8])
xtitle("RANDOM NUMBERS FOR THE PASCAL DISTRIBUTION n=7, p=0.5")
xselect()
```

### See Also

`cdfpascal`, `pdfpascal`

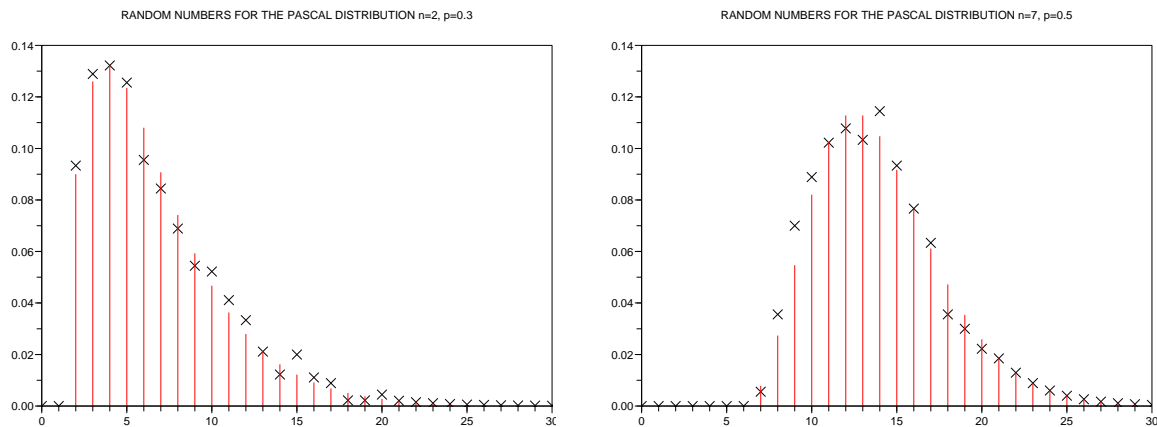


Figure 53: Example of function `rndpascal`

## 10.14 `rndpoisson` – Poisson random number generator

### Calling Sequence

```
X=rndpoisson(row,lam)
X=rndpoisson([row,col],lam)
```

### Parameters

- **X** : real matrix **X**.
- **row** : number  $r$  of rows of matrix **X**. Must be an integer  $\geq 1$ .
- **col** : number  $c$  of columns of matrix **X**. Must be an integer  $\geq 1$ . Default is 1.

- `lam` : parameter  $\lambda > 0$  of the Poisson distribution.

### Description

Generate a  $(r, c)$  matrix **X** of Poisson random numbers. `rndpoisson(row, lam)` is equivalent to `rndpoisson([row, 1], lam)`.

### Examples (see Figure 54)

```
X=(-1:12)';
X1=rndpoisson(900,0.8);
for z=-1:12, Y1(z+2)=length(find(X1==z))/900; end
Z1=pdfpoisson(X,0.8);
xset("window",0);xbasc(0)
plot2d(X,Y1,-2,rect=[-1,0,12,0.5],nax=[0,14,0,11]);plot2d3(X,Z1,5)
xtitle("RANDOM NUMBERS FOR THE POISSON DISTRIBUTION lam=0.8");xselect()
//
X2=rndpoisson(900,3);
for z=-1:12, Y2(z+2)=length(find(X2==z))/900; end
Z2=pdfpoisson(X,3);
xset("window",1);xbasc(1)
plot2d(X,Y2,-2,rect=[-1,0,12,0.5],nax=[0,14,0,11]);plot2d3(X,Z2,5)
xtitle("RANDOM NUMBERS FOR THE POISSON DISTRIBUTION lam=3");xselect()
```

### See Also

`cdfpoisson`, `pdfpoisson`

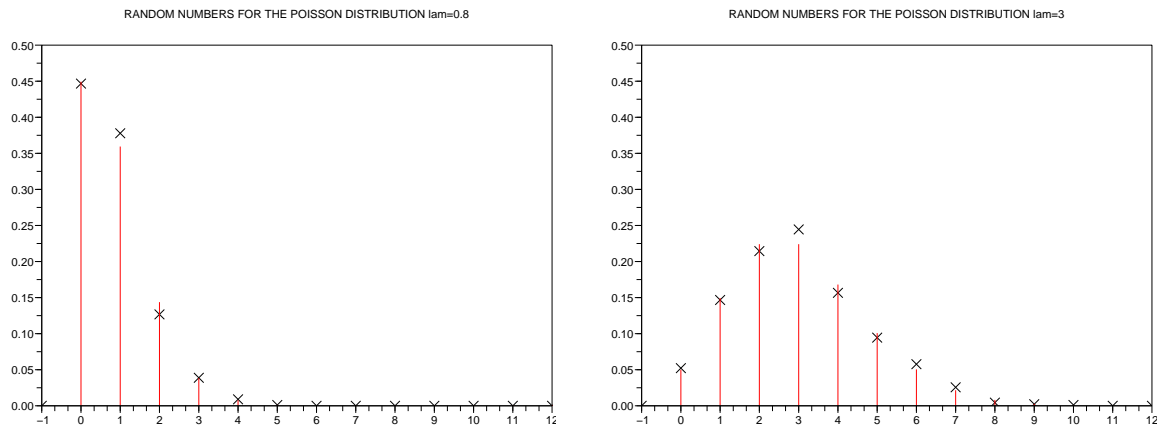


Figure 54: Example of function `rndpoisson`

## 10.15 `rndstandev` – normal sample standard-deviation random number generator

### Calling Sequence

```
X=rndstandev(row,n,sigma=)
X=rndstandev([row,col],n,sigma=)
```

### Parameters

- **X** : real matrix **X**.
- **row** : number  $r$  of rows of matrix **X**. Must be an integer  $\geq 1$ .
- **col** : number  $c$  of columns of matrix **X**. Must be an integer  $\geq 1$ . Default is 1.
- **n** : parameter  $n$  of the normal sample standard-deviation distribution. Must be an integer  $\geq 2$ .
- **sigma** : parameter  $\sigma > 0$  (standard-deviation) of the normal distribution. Default is 1.

### Description

Generate a  $(r, c)$  matrix **X** of normal sample standard-deviation random numbers. The normal sample standard deviation  $(n, \sigma)$  distribution is defined on  $[0, +\infty[$ . `rndstandev(row, n)` is equivalent to `rndstandev([row, 1], n, 1)`.

**Examples** (see Figure 55)

```
X=linspace(0,7,300)';
X1=rdstandev(900,3);
Y1=pdfstandev(X,3);
xset("window",0);xbasc(0);plot2d(X,Y1,5)
histplot(30,X1,1,rect=[0,0,7,0.9],nax=[1,8,1,10])
xtitle("RANDOM NUMBERS FOR THE NORMAL SAMPLE STANDARD-DEVIATION n=3, sigma=1")
xselect()
//
X2=rdstandev(900,9,3.5);
Y2=pdfstandev(X,9,3.5);
xset("window",1);xbasc(1);plot2d(X,Y2,5)
histplot(30,X2,1,rect=[0,0,7,0.9],nax=[1,8,1,10])
xtitle("RANDOM NUMBERS FOR THE NORMAL SAMPLE STANDARD-DEVIATION n=9, sigma=3.5")
xselect()
```

**See Also**

`cdfstandev`, `idfstandev`, `pdfstandev`

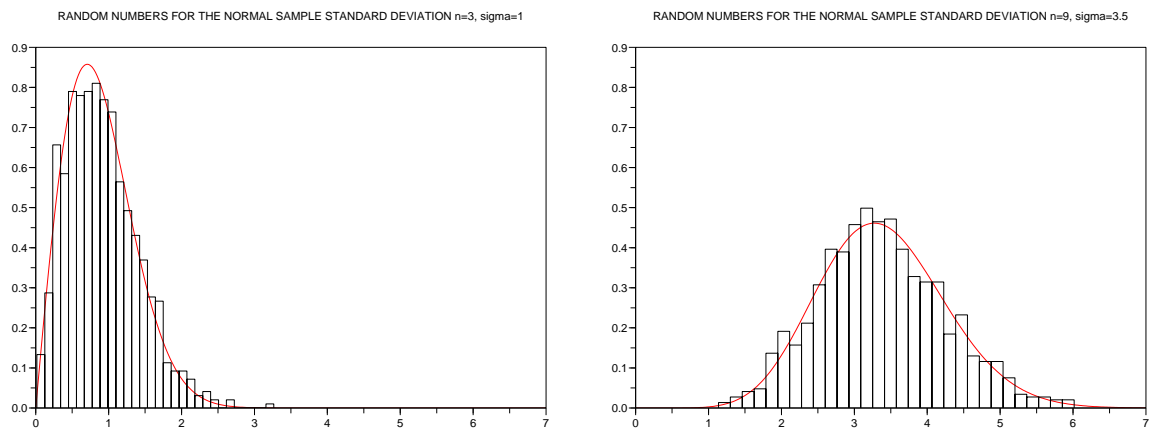


Figure 55: Example of function `rdstandev`

## 10.16 `rndweibull` – Weibull random number generator

### Calling Sequence

```
X=rndweibull(row,a,b,c=)
X=rndweibull([row,col],a,b,c=)
```

### Parameters

- **X** : real matrix **X**.
- **row** : number  $r$  of rows of matrix **X**. Must be an integer  $\geq 1$ .
- **col** : number  $c$  of columns of matrix **X**. Must be an integer  $\geq 1$ . Default is 1.
- **a** : parameter  $a > 0$  of the Weibull distribution.
- **b** : parameter  $b > 0$  of the Weibull distribution. Default is 1.
- **c** : parameter  $c$  of the Weibull distribution. Default is 0.

### Description

Generate a  $(r, c)$  matrix **X** of Weibull random numbers. The Weibull distribution is defined on  $[c, +\infty)$ . `rndweibull(row,a)` is equivalent to `rndweibull([row,1],a,1,0)`.

**Examples** (see Figure 56)

```
X=linspace(-1,7,300)';
X1=rndweibull(900,2);
```



```

Y1=pdfweibull(X,2);
xset("window",0);xbasc(0);plot2d(X,Y1,5)
histplot(30,X1,1,rect=[-1,0,7,0.9])
xtitle("RANDOM NUMBERS FOR THE WEIBULL DISTRIBUTION a=2, b=1, c=0")
xselect()
//
X2=rndweibull(900,5,3,2);
Y2=pdfweibull(X,5,3,2);
xset("window",1);xbasc(1);plot2d(X,Y2,5)
histplot(30,X2,1,rect=[-1,0,7,0.9])
xtitle("RANDOM NUMBERS FOR THE WEIBULL DISTRIBUTION a=5, b=3, c=2")
xselect()

```

See Also

cdfweibull, fitweibull, idfweibull, pdfweibull

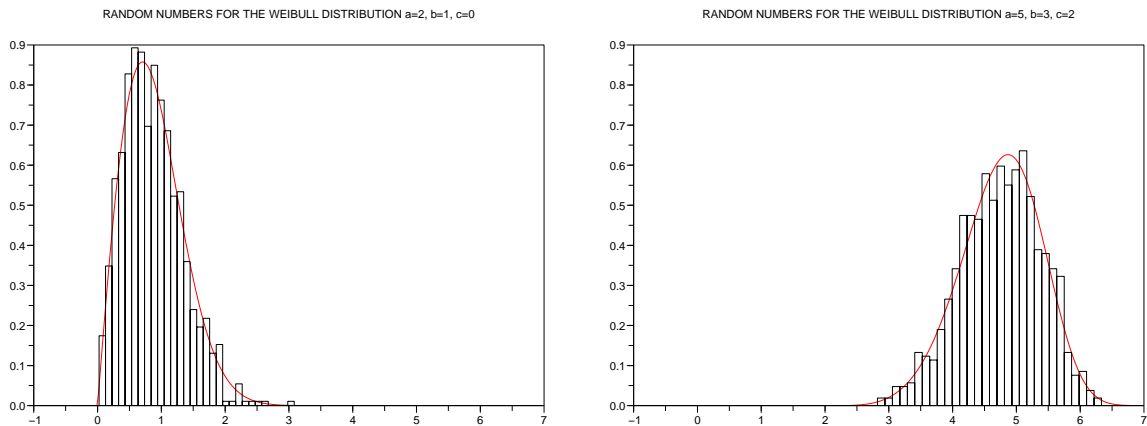


Figure 56: Example of function `rndweibull`

## 11 SAMPLE STATISTICS

### 11.1 autocorrelation – autocorrelation coefficient

Calling Sequence

```
c=autocorrelation(X,lag)
```

Parameters

- **X** : real matrix **X**.
- **lag** : integer vector  $\ell$  of lags. Default is 0.
- **c** : vector of autocorrelation coefficients **c**.

Description

Compute the autocorrelation coefficients  $c_i$  for each lag  $\ell_i$  of  $\ell$ .

Examples

```

X=rndnormal(1000);
for i=2:1000, X(i)=0.5*X(i-1)+X(i); end
autocorrelation(X)
autocorrelation(X,[-1,0,1,2])

```

### 11.2 bootstrap – bootstrap sampling

Calling Sequence

```
Y=bootstrap(row,X)
Y=bootstrap([row,col],X)
```

#### Parameters

- $\mathbf{X}, \mathbf{Y}$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .
- `row` : number  $r$  of rows of matrix  $\mathbf{Y}$ . Must be an integer  $\geq 1$ .
- `col` : number  $c$  of columns of matrix  $\mathbf{Y}$ . Must be an integer  $\geq 1$ . Default is 1.

#### Description

Generate a  $(r, c)$  matrix  $\mathbf{Y}$  by “bootstrapping” the matrix  $\mathbf{X}$ . `bootstrap(row,X)` is equivalent to `bootstrap([row,1],X)`

#### Examples

```
X=rndnormal(100,5,0.1);
Y=bootstrap([1000,7],X);
standev(mean(Y,"c"))
```

### 11.3 correlation – correlation matrix

#### Calling Sequence

```
R=correlation(X)
```

#### Parameters

- $\mathbf{X}$  : real matrix  $\mathbf{X}$  of size  $(n, p)$ .
- $\mathbf{R}$  : correlation matrix  $\mathbf{R}$ .

#### Description

Compute the  $(p, p)$  correlation matrix  $\mathbf{R}$ .

#### Examples

```
X=rndmultinormal(100,[4,5],[2,0.5;0.5,5]);
correlation(X)
```

### 11.4 crosscorrelation – crosscorrelation coefficient

#### Calling Sequence

```
c=crosscorrelation(X,Y,lag)
```

#### Parameters

- $\mathbf{X}, \mathbf{Y}$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$  of the same size.
- `lag` : integer vector  $\ell$  of lags. Default is 0.
- `c` : vector of crosscorrelation coefficients  $\mathbf{c}$ .

#### Description

Compute the crosscorrelation coefficients  $c_i$  for each lag  $\ell_i$  of  $\ell$ .

#### Examples

```
X=rndnormal(1000);
Y=[0;X(1:999)]+rndnormal(1000,0.01);
crosscorrelation(X,Y,[-1,0,1,2])
```

### 11.5 kurtosis – kurtosis coefficient

#### Calling Sequence

```
ku=kurtosis(X)
ku=kurtosis(X,cr)
```

#### Parameters

- $\mathbf{X}$  : real matrix  $\mathbf{X}$ .
- `cr` : column or row option. Must be "c" or "r".
- `ku` : kurtosis coefficient(s) of data in matrix  $\mathbf{X}$ .

## Description

If no option **cr** provided, compute the kurtosis coefficient of data in matrix **X**. If **cr**="c", **ku** is a column vector containing the kurtosis coefficients of each row of **X**. If **cr**="r", **ku** is a row vector containing the kurtosis coefficients of each column of **X**.

## Examples

```
X=rndnormal([30,7]);
kurtosis(X)
kurtosis(X,"c")
kurtosis(X,"r")
```

## See Also

skewness

## 11.6 quantile – quantile

### Calling Sequence

```
q=quantile(X,alpha)
q=quantile(X,alpha,cr)
```

### Parameters

- **X** : real matrix **X**.
- **alpha** : quantile level  $\alpha$ . Must be in  $]0, 1[$ .
- **cr** : column or row option. Must be "c" or "r".
- **q** :  $\alpha$ -level quantile of data in matrix **X**.

### Description

If no option **cr** provided, compute the  $\alpha$ -level quantile of data in matrix **X**. If **cr**="c", **q** is a column vector containing the  $\alpha$ -level quantiles of each row of **X**. If **cr**="r", **q** is a row vector containing the  $\alpha$ -level quantiles of each column of **X**.

## Examples

```
X=rndnormal([100,7]);
quantile(X,0.25)
quantile(X,0.25,"c")
quantile(X,0.75,"r")
```

## 11.7 rng – range

### Calling Sequence

```
rg=rng(X)
rg=rng(X,cr)
```

### Parameters

- **X** : real matrix **X**.
- **cr** : column or row option. Must be "c" or "r".
- **rg** : range of data in matrix **X**.

### Description

If no option **cr** provided, compute the range (i.e.  $\max X_{i,j} - \min X_{i,j}$ ) of data in matrix **X**. If **cr**="c", **rg** is a column vector containing the range of each row of **X**. If **cr**="r", **rg** is a row vector containing the range of each column of **X**.

## Examples

```
X=rndnormal([30,7]);
rng(X)
rng(X,"c")
rng(X,"r")
```

## See Also

standev

## 11.8 skewness – skewness coefficient

### Calling Sequence

```
sk=skewness(X)
sk=skewness(X,cr)
```

### Parameters

- **X** : real matrix **X**.
- **cr** : column or row option. Must be "c" or "r".
- **sk** : skewness coefficient(s) of data in matrix **X**.

### Description

If no option **cr** provided, compute the skewness coefficient of data in matrix **X**. If **cr**="c", **sk** is a column vector containing the skewness coefficients of each row of **X**. If **cr**="r", **sk** is a row vector containing the skewness coefficients of each column of **X**.

### Examples

```
X=rndnormal([30,7]);
skewness(X)
skewness(X,"c")
skewness(X,"r")
```

### See Also

kurtosis

## 11.9 standev – standard deviation

### Calling Sequence

```
sd=standev(X)
sd=standev(X,cr)
```

### Parameters

- **X** : real matrix **X**.
- **cr** : column or row option. Must be "c" or "r".
- **sd** : standard deviation of data in matrix **X**.

### Description

If no option **cr** provided, compute the standard deviation of data in matrix **X**. If **cr**="c", **sd** is a column vector containing the standard deviation of each row of **X**. If **cr**="r", **sd** is a row vector containing the standard deviation of each column of **X**.

### Examples

```
X=rndnormal([30,7]);
standev(X)
standev(X,"c")
standev(X,"r")
```

### See Also

range

## 11.10 totalmedian – total median coefficients

### Calling Sequence

```
a=totalmedian(n)
```

### Parameters

- **n** : sample size  $n \geq 1$ .
- **a** : Total Median row vector of coefficients  $a_i$ .

## Description

Compute the Total Median coefficients  $a_i$ . If  $X_1, \dots, X_n$  is a sample of size  $n$  and  $X_{(1)}, \dots, X_{(n)}$  is the corresponding ordered sample, then the Total Median  $\tilde{X}_T$  is defined as

$$\tilde{X}_T = \sum_{i=1}^n a_i X_{(i)}$$

## Examples

```
n=7;
a=totalmedian(n)
X=rndnormal(n);
Y=sort(X);
Tmed=sum(a.*Y)
```

## 11.11 varcovar – variance-covariance matrix

### Calling Sequence

```
V=varcovar(X)
```

### Parameters

- $\mathbf{X}$  : real  $(n, p)$  matrix  $\mathbf{X}$ .
- $\mathbf{V}$  : symmetric  $(p, p)$  variance-covariance matrix  $\mathbf{V}$ .

### Description

Compute the  $(p, p)$  variance-covariance matrix  $\mathbf{V}$ .

### Examples

```
X=rndmultinormal(100,[4,5],[2,0.5;0.5,5]);
varcovar(X)
```

### See Also

correlation

## 12 STATISTICAL PROCESS CONTROL

### 12.1 arlmean – ARL of the mean control chart

#### Calling Sequence

```
arl=arlmean(tau,n,K,side)
```

#### Parameters

- $\mathbf{tau}$  : real matrix  $\boldsymbol{\tau}$  containing shifts in position  $\tau = |\mu_0 - \mu_1|/\sigma_0 \geq 0$  where  $(\mu_0, \sigma_0)$  are the nominal mean and nominal standard-deviation and where  $\mu_1$  is the non nominal mean.
- $n$  : sample size  $n$ .
- $K$  : control constant  $K \geq 0$ .
- $side$  : must be "1sided" (one-sided control limit) or "2sided" (two-sided control limits). Default is "2sided".

#### Description

Compute the  $ARL$  of the mean control chart for all shifts in position  $\tau$  in matrix  $\boldsymbol{\tau}$ . The control limits of the mean control chart are  $LCL = \mu_0 - K\sigma_0$  and  $UCL = \mu_0 + K\sigma_0$ . `arlmean(tau,n,K)` is equivalent to `arlmean(tau,n,K,"2sided")`.

#### Examples

```
tau=(0:0.1:2)';
n=5;
K=1.3416;
[tau,arlmean(tau,n,K)]
//
K=1.2442;
[tau,arlmean(tau,n,K,"1sided")]
```

## 12.2 arlmeanRR – ARL of the Run Rules mean control chart

### Calling Sequence

```
arl=arlmeanRR(tau,n,K,RR)
```

### Parameters

- **tau** : real matrix  $\tau$  containing shifts in position  $\tau = |\mu_0 - \mu_1|/\sigma_0 \geq 0$  where  $(\mu_0, \sigma_0)$  are the nominal mean and nominal standard-deviation and where  $\mu_1$  is the non nominal mean.
- **n** : sample size  $n$ .
- **K** : control constant  $K \geq 0$ .
- **RR** : Run Rules. Must be "2/3" for "2-out-3" rule, "3/4" for "3-out-4" rule or "4/5" for "4-out-5" rule. Default is "2/3".

### Description

Compute the *ARL* of the Run Rules mean control chart for all shifts in position  $\tau$  in matrix  $\tau$ . The control limits of the Run Rules mean control chart are  $LCL = \mu_0 - K\sigma_0$  and  $UCL = \mu_0 + K\sigma_0$ . `arlmeanRR(tau,n,K)` is equivalent to `arlmeanRR(tau,n,K,"2/3")`.

### Examples

```
tau=(0:0.1:2)';  
n=5;  
K=0.8628;  
[tau,arlmeanRR(tau,n,K)]  
//  
K=0.4666;  
[tau,arlmeanRR(tau,n,K,"4/5")]
```

## 12.3 arlmedian – ARL of the median control chart

### Calling Sequence

```
arl=arlmedian(tau,n,K,side)
```

### Parameters

- **tau** : real matrix  $\tau$  containing shifts in position  $\tau = |\mu_0 - \mu_1|/\sigma_0 \geq 0$  where  $(\mu_0, \sigma_0)$  are the nominal mean and nominal standard-deviation and where  $\mu_1$  is the non nominal mean.
- **n** : sample size  $n$ .
- **K** : control constant  $K \geq 0$ .
- **side** : must be "1sided" (one-sided control limit) or "2sided" (two-sided control limits). Default is "2sided". `arlmedian(tau,n,K)` is equivalent to `arlmedian(tau,n,K,"2sided")`.

### Description

Compute the *ARL* of the median control chart for all shifts in position  $\tau$  in matrix  $\tau$ . The control limits of the median control chart are  $LCL = \mu_0 - K\sigma_0$  and  $UCL = \mu_0 + K\sigma_0$ .

### Example

```
tau=(0:0.1:2)';  
n=5;  
K=1.6193;  
[tau,arlmedian(tau,n,K)]  
//  
K=1.4994;  
[tau,arlmedian(tau,n,K,"1sided")]
```

## 12.4 arlmedianRR – ARL of the Run Rules median control chart

### Calling Sequence

```
arl=arlmedianRR(tau,n,K,RR)
```

### Parameters

- **tau** : real matrix  $\tau$  containing shifts in position  $\tau = |\mu_0 - \mu_1|/\sigma_0 \geq 0$  where  $(\mu_0, \sigma_0)$  are the nominal mean and nominal standard-deviation and where  $\mu_1$  is the non nominal mean.

- **n** : sample size  $n$ .
- **K** : control constant  $K \geq 0$ .
- **RR** : Run Rules. Must be "2/3" for "2-out-3" rule, "3/4" for "3-out-4" rule or "4/5" for "4-out-5" rule. Default is "2/3".

### Description

Compute the *ARL* of the Run Rules median control chart for all shifts in position  $\tau$  in matrix  $\tau$ . The control limits of the Run Rules median control chart are  $LCL = \mu_0 - K\sigma_0$  and  $UCL = \mu_0 + K\sigma_0$ . `arlmedianRR(tau,n,K)` is equivalent to `arlmedianRR(tau,n,K,"2/3")`.

### Example

```
tau=(0:0.1:2)';
n=5;
K=1.0344;
[tau,arlmedianRR(tau,n,K)]
//
K=0.5573;
[tau,arlmedianRR(tau,n,K,"4/5")]
```

## 12.5 arlrnge – *ARL* of the range control chart

### Calling Sequence

```
arl=arlrng(tau,n,KL,KU)
```

### Parameters

- **tau** : real matrix  $\tau$  containing shifts in dispersion  $\tau = \sigma_1/\sigma_0 > 0$  where  $\sigma_0$  is the nominal standard-deviation and where  $\sigma_1$  is the non nominal standard-deviation.
- **n** : sample size  $n$ .
- **KL** : lower control constant  $K_L \geq 0$ .
- **KU** : upper control constant  $K_U \geq K_L$ .

### Description

Compute the *ARL* of the range control chart for all shifts in dispersion  $\tau$  in matrix  $\tau$ . The control limits of the range control chart are  $LCL = K_L\sigma_0$  and  $UCL = K_U\sigma_0$ .

### Example

```
tau=[0.5:0.1:0.9,0.95,1,1.05,1.1:0.1:2]';
n=5;
KL=0.3965;
KU=5.3774;
[tau,arlrng(tau,n,KL,KU)]
```

## 12.6 arlstandev – *ARL* of the standard-deviation control chart

### Calling Sequence

```
arl=arlstandev(tau,n,KL,KU)
```

### Parameters

- **tau** : real matrix  $\tau$  containing shifts in dispersion  $\tau = \sigma_1/\sigma_0 > 0$  where  $\sigma_0$  is the nominal standard-deviation and where  $\sigma_1$  is the non nominal standard-deviation.
- **n** : sample size  $n$ .
- **KL** : lower control constant  $K_L \geq 0$ .
- **KU** : upper control constant  $K_U \geq K_L$ .

### Description

Compute the *ARL* of the standard-deviation control chart for all shifts in dispersion  $\tau$  in matrix  $\tau$ . The control limits of the standard-deviation control chart are  $LCL = K_L\sigma_0$  and  $UCL = K_U\sigma_0$ .

### Example

```
tau=[0.5:0.1:0.9,0.95,1,1.05,1.1:0.1:2]';
n=5;
KL=0.1626;
KU=2.1095;
[tau,arlstandev(tau,n,KL,KU)]
```

## 12.7 arlstandevRR – ARL of the Run Rules standard-deviation control chart

### Calling Sequence

```
arl=arlstandevRR(tau,n,KL,KU,RR)
```

### Parameters

- **tau** : real matrix  $\tau$  containing shifts in dispersion  $\tau = \sigma_1/\sigma_0 > 0$  where  $\sigma_0$  is the nominal standard-deviation and where  $\sigma_1$  is the non nominal standard-deviation.
- **n** : sample size  $n$ .
- **KL** : lower control constant  $K_L \geq 0$ .
- **KU** : upper control constant  $K_U \geq K_L$ .
- **RR** : Run Rules. Must be "2/3" for "2-out-3" rule, "3/4" for "3-out-4" rule or "4/5" for "4-out-5" rule. Default is "2/3".

### Description

Compute the *ARL* of the run rules standard-deviation control chart for all shifts in dispersion  $\tau$  in matrix  $\tau$ . The control limits of the run rules standard-deviation control chart are  $LCL = K_L\sigma_0$  and  $UCL = K_U\sigma_0$ . `arlstandevRR(tau,n,KL,KU)` is equivalent to `arlstandevRR(tau,n,KL,KU,"2/3")`.

### Example

```
tau=[0.5:0.1:0.9,0.95,1,1.05,1.1:0.1:2]';
n=5;
KL=0.3548;
KU=1.6564;
[tau,arlstandevRR(tau,n,KL,KU)]
//
KL=0.5825;
KU=1.3012;
[tau,arlstandevRR(tau,n,KL,KU,"4/5")]
```

## 12.8 cp – capability index $C_P$ estimation and confidence interval

### Calling Sequence

```
Cp=cp(X,L,U,side=,level=)
[Cp,inter]=cp(X,L,U,side=,level=)
```

### Parameters

- **X** : real matrix  $\mathbf{X}$ .
- **L** : lower specification limit  $L$ .
- **U** : upper specification limit  $U$ .
- **side** : side of the confidence interval. Must be "lower", "upper" or "both". Default is "both".
- **level** : confidence level  $1 - \alpha \in ]0.5, 1[$ . Default is 0.95.
- **Cp** : estimation of capability index  $C_P$ .
- **inter** : capability index  $C_P$  confidence interval :
  - if **side**="lower", **inter** is the *lower* confidence bound.
  - if **side**="upper", **inter** is the *upper* confidence bound.
  - if **side**="both", **inter** is the bilateral confidence interval.

### Description

Estimate the capability index  $C_P$  and compute the corresponding confidence interval (lower, upper or both). `[Cp,inter]=cp(x,L,U)` is equivalent to `[Cp,inter]=cp(x,L,U,"both",0.95)`.

### Examples



```

X=rndnormal(100,20,0.1);
L=19.5
U=20.5
Cp=cp(X,L,U)
[Cp,inter]=cp(X,L,U)
[Cp,inter]=cp(X,L,U,"upper")
[Cp,inter]=cp(X,L,U,level=0.99)

```

## 12.9 cpk – capability index $C_{PK}$ estimation and confidence interval

### Calling Sequence

```

Cpk=cpk(X,L,U,side=,level=)
[Cpk,inter]=cpk(X,L,U,side=,level=)

```

### Parameters

- **X** : real matrix **X**.
- **L** : lower specification limit  $L$ .
- **U** : upper specification limit  $U$ .
- **side** : side of the confidence interval. Must be "lower", "upper" or "both". Default is "both".
- **level** : confidence level  $1 - \alpha \in ]0.5, 1[$ . Default is 0.95.
- **Cpk** : estimation of capability index  $C_{PK}$ .
- **inter** : capability index  $C_{PK}$  confidence interval :
  - if **side**="lower", **inter** is the *lower* confidence bound.
  - if **side**="upper", **inter** is the *upper* confidence bound.
  - if **side**="both", **inter** is the bilateral confidence interval.

### Description

Estimate the capability index  $C_{PK}$  and compute the corresponding confidence interval (lower, upper or both).  $[Cpk,inter]=cpk(X,L,U)$  is equivalent to  $[Cpk,inter]=cpk(X,L,U,"both",0.95)$ .

### Examples

```

X=rndnormal(100,20,0.1);
L=19.5
U=20.5
Cpk=cpk(X,L,U)
[Cpk,inter]=cpk(X,L,U)
[Cpk,inter]=cpk(X,L,U,"upper")
[Cpk,inter]=cpk(X,L,U,level=0.99)

```

## 12.10 krnge – range coefficients $K_R(n)$

### Calling Sequence

```

K=krnge(N)

```

### Parameters

- **N** : matrix **N** of integers  $N_{i,j} \geq 2$ .
- **K** : real matrix **K** of coefficients  $K_{i,j} = K_R(N_{i,j})$ .

### Description

Compute in matrix **K** the coefficients  $K_{i,j} = K_R(N_{i,j})$  for each entry of matrix **N**. If data in vector **x** follow a normal  $(\mu, \sigma)$  distribution, then  $\text{rng}(\mathbf{x})/\text{krnge}(\text{length}(\mathbf{x}))$  is an unbiased estimator for  $\sigma$ .

### Examples

```

m=10000;n=5;
x=rndnormal([m,n],5,0.1);
sunbiased=rng(x,"c")/krnge(n);
mean(sunbiased)

```

### See Also

kstandev

## 12.11 kstandev – standard-deviation coefficients $K_S(n, r)$

### Calling Sequence

```
K=kstandev(N)
K=kstandev(N,r)
```

### Parameters

- **N** : matrix **N** of integers  $N_{i,j} \geq \max(2, 2 - r)$ .
- **r** : power  $r$  of standard-deviation  $\sigma$ . Default is 1.
- **K** : real matrix **K** of coefficients  $K_{i,j} = K_S(N_{i,j}, r)$ .

### Description

Compute in matrix **K** the coefficients  $K_{i,j} = K_S(N_{i,j}, r)$  for each entry of matrix **N**. If data in vector **x** follow a normal  $(\mu, \sigma)$  distribution, then  $(\text{standev}(\mathbf{x}))^r / \text{kstandev}(\text{length}(\mathbf{x}), r)$  is an unbiased estimator for  $\sigma^r$ . `kstandev(n)` is equivalent to `kstandev(n, 1)`.

### Examples

```
m=10000;n=5;
x=rndnormal([m,n],5,0.1);
sbiased=standev(x,"c");
sunbiased=sbiased/kstandev(n);
mean([sbiased,sunbiased],"r")
//
kstandev((3:20)',-1)
```

### See Also

krnge

## 12.12 mcpshahriari – Shahriari’s multivariate capability index $C_P$

### Calling Sequence

```
Cp=mcpshahriari(X,L,U)
```

### Parameters

- **X** : real matrix **X** of size  $(n, p)$ .
- **L** : vector of lower specification limits  $L$ .
- **U** : vector of upper specification limit  $U$ .
- **Cp** : estimation of Shahriari’s multivariate capability index  $C_P$ .

### Description

Compute the Shahriari’s multivariate capability index  $C_P$  based on the paper “Multivariate Process Capability Vector”, H. Shahriari, N.F. Hubele and F.P. Lawrence, Proceedings of the 4th Industrial Engineering Research Conference, Institute of Industrial Engineers, pp. 304–309, 1995.

### Examples

```
X=rndmultinormal(100,[20,40],[0.1,0.05;0.05,0.15]);
L=[19.5,39.5];
U=[20.5,40.5];
Cp=mcpshahriari(X,L,U)
```

## 12.13 mcptaam – Taam’s multivariate capability index $C_P$

### Calling Sequence

```
Cp=mcptaam(X,L,U)
```

### Parameters

- **X** : real matrix **X** of size  $(n, p)$ .
- **L** : vector of lower specification limits  $L$ .
- **U** : vector of upper specification limit  $U$ .
- **Cp** : estimation of Taam’s multivariate capability index  $C_P$ .

## Description

Compute the Taam's multivariate capability index  $C_P$  based on the paper "A Note on Multivariate Capability Indices", W. Taam, P. Subbaiah and J.W. Liddy, Journal of Applied Statistics 20, pp. 339–351, 1993.

## Examples

```
X=rmultinormal(100,[20,40],[0.1,0.05;0.05,0.15]);
L=[19.5,39.5]
U=[20.5,40.5]
Cp=mcptaam(X,L,U)
```

# 13 TESTS

## 13.1 andersondarling – Anderson-Darling's normality test

### Calling Sequence

```
pv=andersondarling(X)
```

### Parameters

- $X$  : real matrix  $\mathbf{X}$ .
- $pv$  :  $p$ -value of the Anderson-Darling test.

### Description

Compute the  $p$ -value of the Anderson-Darling normality test. If the  $p$ -value  $pv$  is  $< 0.05$  then the hypothesis " $H_0$ : the data follow a normal distribution" is rejected.

## Examples

```
Xn=rmvnormal(100,5,0.1);
andersondarling(Xn)
Xln=rmvlognormal(100,2,3,4);
andersondarling(Xln)
```

## See Also

tstsku

## 13.2 ansaribradley – Ansari-Bradley's test

### Calling Sequence

```
pv=ansaribradley(X,Y)
pv=ansaribradley(X,Y,t)
```

### Parameters

- $X, Y$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .
- $t$  : type of test. Must be "<", ">" or "~". Default is "~".
- $pv$  :  $p$ -value of the Ansari-Bradley test.

### Description

Compute the  $p$ -value (normal approximation) of the Ansari-Bradley test. Ties are not taken into account.

- if  $t="<"$  the  $p$ -value  $pv$  corresponds to the test  $H_0 : \sigma_X = \sigma_Y$  versus  $H_1 : \sigma_X < \sigma_Y$ ,
- if  $t=">"$  the  $p$ -value  $pv$  corresponds to the test  $H_0 : \sigma_X = \sigma_Y$  versus  $H_1 : \sigma_X > \sigma_Y$ ,
- if  $t="~"$  the  $p$ -value  $pv$  corresponds to the test  $H_0 : \sigma_X = \sigma_Y$  versus  $H_1 : \sigma_X \neq \sigma_Y$ .

If the  $p$ -value  $pv$  is  $< 0.05$  then the hypothesis " $H_0 : \sigma_X = \sigma_Y$ " is rejected.

## Examples

```
X=rmvnormal(90,5,0.1);
Y=rmvnormal(110,6,0.15);
ansaribradley(X,Y,"<")
ansaribradley(X,Y,">")
ansaribradley(X,Y)
```

### 13.3 bartlett – Bartlett’s test

#### Calling Sequence

```
pv=bartlett(X1,X2,...)
```

#### Parameters

- $X_1, X_2, \dots$  : real matrices  $\mathbf{X}_1, \mathbf{X}_2, \dots$
- $pv$  :  $p$ -value of the Bartlett test.

#### Description

Compute the  $p$ -value of the Bartlett test. If the  $p$ -value  $pv$  is  $< 0.05$  then the hypothesis “ $H_0 : \sigma_{X_1} = \sigma_{X_2} = \dots$ ” is rejected.

#### Examples

```
X1=rndnormal(100,5,0.1);
X2=rndnormal(100,6,0.1);
X3=rndnormal(100,4,0.1);
X4=rndnormal(100,5,0.15);
bartlett(X1,X2,X3)
bartlett(X1,X2,X4)
```

#### See Also

levene

### 13.4 grubbs – Grubbs test

#### Calling Sequence

```
[pv,i]=grubbs(X)
```

#### Parameters

- $X$  : real matrix  $\mathbf{X}$ .
- $pv$  :  $p$ -value of the Grubbs test.
- $i$  : index of the potential outlier.

#### Description

Compute the  $p$ -value of the Grubbs test. If the  $p$ -value  $pv$  is  $< 0.05$  then the hypothesis “ $H_0$  : the sample does not contain any outlier” is rejected and  $X_i$  is assumed to be a potential outlier. Data in  $\mathbf{X}$  are supposed to be normally distributed.

#### Examples

```
X=[rndnormal(30,20,0.1);20.7];
[pv,i]=grubbs(X)
```

### 13.5 kendall – Kendall’s test

#### Calling Sequence

```
pv=kendall(X,Y)
```

#### Parameters

- $X, Y$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$  of the same sizes.
- $pv$  :  $p$ -value of the Kendall’s test.

#### Description

Compute the  $p$ -value of the Kendall’s test. If the  $p$ -value  $pv$  is  $< 0.05$  then the hypothesis “ $H_0 : \mathbf{X}$  and  $\mathbf{Y}$  are uncorrelated” is rejected.

#### Examples

```
X1=rndmultinormal(100,[0,0]);
kendall(X1(:,1),X1(:,2))
X2=rndmultinormal(100,[0,0],[2,1.9;1.9,5]);
kendall(X2(:,1),X2(:,2))
```

#### See Also

spearman

### 13.6 levene – Levene’s test

#### Calling Sequence

```
pv=levene(X1,X2,...)
```

#### Parameters

- $X_1, X_2, \dots$  : real matrices  $\mathbf{X}_1, \mathbf{X}_2, \dots$
- $pv$  :  $p$ -value of the Levene’s test.

#### Description

Compute the  $p$ -value of the Levene’s test. If the  $p$ -value  $pv$  is  $< 0.05$  then the hypothesis “ $H_0 : \sigma_{X_1} = \sigma_{X_2} = \dots$ ” is rejected.

#### Examples

```
X1=rndnormal(100,5,0.1);
X2=rndnormal(100,6,0.1);
X3=rndnormal(100,4,0.1);
X4=rndnormal(100,5,0.15);
levene(X1,X2,X3)
levene(X1,X2,X4)
```

#### See Also

bartlett

### 13.7 mardia – Mardia’s test

#### Calling Sequence

```
[pvsk,pvku]=mardia(X)
```

#### Parameters

- $X$  : real  $(n, p)$  matrix  $\mathbf{X}$ .
- $pvsk$  :  $p$ -value of Mardia’s normal multivariate skewness test.
- $pvku$  :  $p$ -value of Mardia’s normal multivariate kurtosis test.

#### Description

Compute the  $p$ -values of the Mardia’s normal multivariate skewness and kurtosis test. If the  $p$ -values  $pvsk$  or  $pvku$  are  $< 0.05$  then the hypothesis “ $H_0$  : the data follow a multinormal distribution” is rejected.

#### Examples

```
X1=rdmultinormal(100,[4,5],[2,0.5;0.5,5]);
[pvsk,pvku]=mardia(X1)
X2=[rndlognormal(100,2,3,4);rndlognormal(100,2,3,4)];
[pvsk,pvku]=mardia(X2)
```

### 13.8 spearman – Spearman’s test

#### Calling Sequence

```
pv=spearman(X,Y)
```

#### Parameters

- $X, Y$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$  of the same sizes.
- $pv$  :  $p$ -value of the Spearman’s test.

#### Description

Compute the  $p$ -value of the Spearman’s test. If the  $p$ -value  $pv$  is  $< 0.05$  then the hypothesis “ $H_0$  :  $\mathbf{X}$  and  $\mathbf{Y}$  are uncorrelated” is rejected.

#### Examples

```
X1=rndmultinormal(100,[0,0]);
spearman(X1(:,1),X1(:,2))
X2=rndmultinormal(100,[0,0],[2,1.9;1.9,5]);
spearman(X2(:,1),X2(:,2))
```

See Also

kendall

### 13.9 tstbinomial1 – binomial one sample $p$ test

Calling Sequence

```
pv=tstbinomial1(x,n,p0)
pv=tstbinomial1(x,n,p0,t)
```

Parameters

- $x$  : number  $x$  of successes observed. An integer in  $\{0, \dots, n\}$ .
- $n$  : number  $n$  of binomial trials. An integer  $\geq 1$ .
- $p_0$  : parameter  $p_0$  of the binomial distribution.
- $t$  : type of test. Must be "<", ">" or "~". Default is "~".
- $pv$  :  $p$ -value of the binomial one sample  $p$  test.

Description

Compute the  $p$ -value of the binomial one sample  $p$  test.

- if  $t=<$  the  $p$ -value  $pv$  corresponds to the test  $H_0 : p = p_0$  versus  $H_1 : p < p_0$ ,
- if  $t=>$  the  $p$ -value  $pv$  corresponds to the test  $H_0 : p = p_0$  versus  $H_1 : p > p_0$ ,
- if  $t=\sim$  the  $p$ -value  $pv$  corresponds to the test  $H_0 : p = p_0$  versus  $H_1 : p \neq p_0$ .

If the  $p$ -value  $pv$  is  $\geq 0.05$  then the hypothesis " $H_0 : p = p_0$ " is accepted, otherwise this hypothesis is rejected.

Examples

```
tstbinomial1(18,100,0.2,<)
tstbinomial1(18,100,0.1,>)
tstbinomial1(18,100,0.15)
```

See Also

intbinomial, tstbinomial2

### 13.10 tstbinomial2 – binomial two samples $p$ test

Calling Sequence

```
pv=tstbinomial2(x,nx,y,ny)
pv=tstbinomial2(x,nx,y,ny,t)
```

Parameters

- $x$  : number  $x$  of successes observed in the 1st sample. An integer in  $\{0, \dots, n_X\}$ .
- $nx$  : number  $n_X$  of binomial trials in the 1st sample. An integer  $\geq 1$ .
- $y$  : number  $y$  of successes observed in the 2nd sample. An integer in  $\{0, \dots, n_Y\}$ .
- $ny$  : number  $n_Y$  of binomial trials in the 2nd sample. An integer  $\geq 1$ .
- $t$  : type of test. Must be "<", ">" or "~". Default is "~".
- $pv$  :  $p$ -value of the binomial two samples  $p$  test.

Description

Compute the  $p$ -value of the binomial two samples  $p$  test.

- if  $t=<$  the  $p$ -value  $pv$  corresponds to the test  $H_0 : p_X = p_Y$  versus  $H_1 : p_X < p_Y$ ,
- if  $t=>$  the  $p$ -value  $pv$  corresponds to the test  $H_0 : p_X = p_Y$  versus  $H_1 : p_X > p_Y$ ,
- if  $t=\sim$  the  $p$ -value  $pv$  corresponds to the test  $H_0 : p_X = p_Y$  versus  $H_1 : p_X \neq p_Y$ .

If the  $p$ -value  $pv$  is  $\geq 0.05$  then the hypothesis " $H_0 : p_X = p_Y$ " is accepted, otherwise this hypothesis is rejected.

Examples

```
tstbinomial2(18,100,23,100,<)
tstbinomial2(42,100,50,200,>)
tstbinomial2(18,100,34,200)
```

See Also

tstbinomial1

### 13.11 tstexponential – exponential $\lambda$ test

#### Calling Sequence

```
pv=tstexponential(X,lam0)
pv=tstexponential(X,lam0,t)
```

#### Parameters

- $X$  : real matrix  $\mathbf{X}$ .
- $\text{lam0}$  : parameter  $\lambda_0$  of the exponential distribution.
- $t$  : type of test. Must be "<", ">" or "~". Default is "~".
- $pv$  :  $p$ -value of the exponential  $\lambda$  test.

#### Description

Compute the  $p$ -value of the exponential  $\lambda$  test.

- if  $t=<$  the  $p$ -value  $pv$  corresponds to the test  $H_0 : \lambda = \lambda_0$  versus  $H_1 : \lambda < \lambda_0$ ,
- if  $t=>$  the  $p$ -value  $pv$  corresponds to the test  $H_0 : \lambda = \lambda_0$  versus  $H_1 : \lambda > \lambda_0$ ,
- if  $t=\sim$  the  $p$ -value  $pv$  corresponds to the test  $H_0 : \lambda = \lambda_0$  versus  $H_1 : \lambda \neq \lambda_0$ .

If the  $p$ -value  $pv$  is  $\geq 0.05$  then the hypothesis " $H_0 : \lambda = \lambda_0$ " is accepted, otherwise this hypothesis is rejected.

#### Examples

```
X=rndexponential(100,5);
tstexponential(X,6,"<")
tstexponential(X,4,">")
tstexponential(X,5)
```

#### See Also

intexponential

### 13.12 tstnormalm1 – normal one sample $\mu$ test

#### Calling Sequence

```
pv=tstnormalm1(X,mu0)
pv=tstnormalm1(X,mu0,t)
```

#### Parameters

- $X$  : real matrix  $\mathbf{X}$ .
- $\text{mu0}$  : parameter  $\mu_0$  of the normal distribution.
- $t$  : type of test. Must be "<", ">" or "~". Default is "~".
- $pv$  :  $p$ -value of the normal one sample  $\mu$  test.

#### Description

Compute the  $p$ -value of the normal one sample  $\mu$  test.

- if  $t=<$  the  $p$ -value  $pv$  corresponds to the test  $H_0 : \mu = \mu_0$  versus  $H_1 : \mu < \mu_0$ ,
- if  $t=>$  the  $p$ -value  $pv$  corresponds to the test  $H_0 : \mu = \mu_0$  versus  $H_1 : \mu > \mu_0$ ,
- if  $t=\sim$  the  $p$ -value  $pv$  corresponds to the test  $H_0 : \mu = \mu_0$  versus  $H_1 : \mu \neq \mu_0$ .

If the  $p$ -value  $pv$  is  $\geq 0.05$  then the hypothesis " $H_0 : \mu = \mu_0$ " is accepted, otherwise this hypothesis is rejected.

#### Examples

```
X=rndnormal(100,5,0.1);
tstnormalm1(X,6,"<")
tstnormalm1(X,4,">")
tstnormalm1(X,5)
```

#### See Also

tstnormalm2, tstnormals1, tstnormals2

### 13.13 `tstnormalm2` – normal two samples $\mu$ test

#### Calling Sequence

```
pv=tstnormalm2(X,Y)
pv=tstnormalm2(X,Y,t)
```

#### Parameters

- `X,Y` : real matrices **X** and **Y**.
- `t` : type of test. Must be "<", ">" or "~". Default is "~".
- `pv` :  $p$ -value of the normal two samples  $\mu$  test.

#### Description

Compute the  $p$ -value of the normal two samples  $\mu$  test.

- if `t`="<" the  $p$ -value `pv` corresponds to the test  $H_0 : \mu_X = \mu_Y$  versus  $H_1 : \mu_X < \mu_Y$ ,
- if `t`=">" the  $p$ -value `pv` corresponds to the test  $H_0 : \mu_X = \mu_Y$  versus  $H_1 : \mu_X > \mu_Y$ ,
- if `t`="~" the  $p$ -value `pv` corresponds to the test  $H_0 : \mu_X = \mu_Y$  versus  $H_1 : \mu_X \neq \mu_Y$ .

If the  $p$ -value `pv` is  $\geq 0.05$  then the hypothesis " $H_0 : \mu_X = \mu_Y$ " is accepted, otherwise this hypothesis is rejected.

#### Examples

```
X=rndnormal(90,5,0.1);
Y=rndnormal(110,5.1,0.1);
tstnormalm2(X,Y,"<")
tstnormalm2(X,Y,">")
tstnormalm2(X,Y)
```

#### See Also

`tstnormalm1`, `tstnormals1`, `tstnormals2`

### 13.14 `tstnormals1` – normal one sample $\sigma$ test

#### Calling Sequence

```
pv=tstnormals1(X,sig0)
pv=tstnormals1(X,sig0,t)
```

#### Parameters

- `X` : real matrix **X**.
- `sig0` : parameter  $\sigma_0$  of the normal distribution.
- `t` : type of test. Must be "<", ">" or "~". Default is "~".
- `pv` :  $p$ -value of the normal one sample  $\sigma$  test.

#### Description

Compute the  $p$ -value of the normal one sample  $\sigma$  test.

- if `t`="<" the  $p$ -value `pv` corresponds to the test  $H_0 : \sigma = \sigma_0$  versus  $H_1 : \sigma < \sigma_0$ ,
- if `t`=">" the  $p$ -value `pv` corresponds to the test  $H_0 : \sigma = \sigma_0$  versus  $H_1 : \sigma > \sigma_0$ ,
- if `t`="~" the  $p$ -value `pv` corresponds to the test  $H_0 : \sigma = \sigma_0$  versus  $H_1 : \sigma \neq \sigma_0$ .

If the  $p$ -value `pv` is  $\geq 0.05$  then the hypothesis " $H_0 : \sigma = \sigma_0$ " is accepted, otherwise this hypothesis is rejected.

#### Examples

```
X=rndnormal(100,5,0.1);
tstnormals1(X,0.15,"<")
tstnormals1(X,0.05,">")
tstnormals1(X,0.1)
```

#### See Also

`tstnormalm1`, `tstnormalm2`, `tstnormals2`



### 13.15 `tstnormals2` – normal two samples $\sigma$ test

#### Calling Sequence

```
pv=tstnormals2(X,Y)
pv=tstnormals2(X,Y,t)
```

#### Parameters

- $X, Y$  : real matrices  $\mathbf{X}$  and  $\mathbf{Y}$ .
- $t$  : type of test. Must be "<", ">" or "~". Default is "~".
- $pv$  :  $p$ -value of the normal two samples  $\sigma$  test.

#### Description

Compute the  $p$ -value of the normal two samples  $\sigma$  test.

- if  $t=<$  the  $p$ -value  $pv$  corresponds to the test  $H_0 : \sigma_X = \sigma_Y$  versus  $H_1 : \sigma_X < \sigma_Y$ ,
- if  $t=>$  the  $p$ -value  $pv$  corresponds to the test  $H_0 : \sigma_X = \sigma_Y$  versus  $H_1 : \sigma_X > \sigma_Y$ ,
- if  $t=\sim$  the  $p$ -value  $pv$  corresponds to the test  $H_0 : \sigma_X = \sigma_Y$  versus  $H_1 : \sigma_X \neq \sigma_Y$ .

If the  $p$ -value  $pv$  is  $\geq 0.05$  then the hypothesis " $H_0 : \sigma_X = \sigma_Y$ " is accepted, otherwise this hypothesis is rejected.

#### Examples

```
X=rndnormal(90,5,0.1);
Y=rndnormal(110,5,0.15);
tstnormals2(X,Y,"<")
tstnormals2(X,Y,">")
tstnormals2(X,Y)
```

#### See Also

`tstnormalm1`, `tstnormalm2`, `tstnormals1`

### 13.16 `tstsku` – normal skewness and kurtosis test

#### Calling Sequence

```
[pvsk,pvku,pvsku]=tstsku(X)
```

#### Parameters

- $X$  : real matrix  $\mathbf{X}$ .
- $pvsk$  :  $p$ -value of the normal skewness test.
- $pvku$  :  $p$ -value of the normal kurtosis test.
- $pvsku$  :  $p$ -value of the joint normal skewness/kurtosis test.

#### Description

Compute the  $p$ -values of the normal skewness, kurtosis and joint skewness/kurtosis test:

- If the  $p$ -values  $pvsk$  or  $pvku$  are  $< 0.05$  then the hypothesis " $H_0$  : the data follow a normal distribution" is rejected.
- If the  $p$ -value  $pvsku$  is  $< 0.05$  then the hypothesis " $H_0$  : the data follow a normal distribution" is rejected.

#### Examples

```
Xn=rndnormal(100,5,0.1);
[pvsk,pvku,pvsku]=tstsku(Xn); [pvsk,pvku,pvsku]
Xln=rndlognormal(100,2,3,4);
[pvsk,pvku,pvsku]=tstsku(Xln); [pvsk,pvku,pvsku]
```

#### See Also

`andersondarling`

### 13.17 `waldwolfowitz` – Wald-Wolfowitz's run test

#### Calling Sequence

```
pv=waldwolfowitz(X)
```

**Parameters**

- **X** : matrix **X** of  $\{0,1\}$ .
- **pv** :  $p$ -value of the Wald-Wolfowitz's run test.

**Description**

Compute the  $p$ -value of the Wald-Wolfowitz's run test (normal approximation). If the  $p$ -value **pv** is  $< 0.05$  then the hypothesis " $H_0$  : the sample is random" is rejected.

**Examples**

```
X1=rndbinomial(40,1,0.5);
waldwolfowitz(X1)
X2=[ones(1,10),zeros(1,10),ones(1,10),zeros(1,10)];
waldwolfowitz(X2)
```

**13.18 wilcoxon1 – Wilcoxon's one sample (paired) test****Calling Sequence**

```
pv=wilcoxon1(X,Y)
pv=wilcoxon1(X,Y,t)
```

**Parameters**

- **X,Y** : real matrices **X** and **Y** of the same size.
- **t** : type of test. Must be "<", ">" or "~". Default is "~".
- **pv** :  $p$ -value of the Wilcoxon's one sample test. Ties are not taken into account.

**Description**

Compute the  $p$ -value of the Wilcoxon's one sample test.

- if **t**="<" the  $p$ -value **pv** corresponds to the test  $H_0 : X = Y$  versus  $H_1 : X < Y$ ,
- if **t**=">" the  $p$ -value **pv** corresponds to the test  $H_0 : X = Y$  versus  $H_1 : X > Y$ ,
- if **t**="~" the  $p$ -value **pv** corresponds to the test  $H_0 : X = Y$  versus  $H_1 : X \neq Y$ .

If the  $p$ -value **pv** is  $< 0.05$  then the hypothesis " $H_0 : X = Y$ " is rejected.

**Examples**

```
X=rndnormal(100,5,0.1);
Y=rndnormal(100,5.1,0.1);
wilcoxon1(X,Y,"<")
wilcoxon1(X,Y,">")
wilcoxon1(X,Y)
```

**See Also**

wilcoxon2

**13.19 wilcoxon2 – Wilcoxon's two samples test****Calling Sequence**

```
pv=wilcoxon2(X,Y)
pv=wilcoxon2(X,Y,t)
```

**Parameters**

- **X,Y** : real matrices **X** and **Y**.
- **t** : type of test. Must be "<", ">" or "~". Default is "~".
- **pv** :  $p$ -value of the Wilcoxon's two samples test. Ties are not taken into account.

**Description**

Compute the  $p$ -value of the Wilcoxon's two samples test.

- if **t**="<" the  $p$ -value **pv** corresponds to the test  $H_0 : X = Y$  versus  $H_1 : X < Y$ ,
- if **t**=">" the  $p$ -value **pv** corresponds to the test  $H_0 : X = Y$  versus  $H_1 : X > Y$ ,
- if **t**="~" the  $p$ -value **pv** corresponds to the test  $H_0 : X = Y$  versus  $H_1 : X \neq Y$ .

If the  $p$ -value **pv** is  $< 0.05$  then the hypothesis " $H_0 : X = Y$ " is rejected.

**Examples**

```
X=rndnormal(90,5,0.1);  
Y=rndnormal(110,5.1,0.1);  
wilcoxon2(X,Y,"<")  
wilcoxon2(X,Y,">")  
wilcoxon2(X,Y)
```

#### See Also

wilcoxon1