Jによる確率分布関数と乱数関数へのイントロダクション

Masato Shimura
jcd02773@nifty.com

2009年7月31日

目次

はじめに

この稿は J 言語で記述されている確率分布と乱数のスクリプトを集めて整理したものである。

実際の計算を通しての確率と確率過程の応用のためスクリプトの動かし方とそのための例題を記した。貴重な関数群と例題は、引用した著者の関数、出典を記してそのまま紹介したものが多い。何れも文献で公表されているものである。

移住 J の system/package の中の math,stats,finance の項が整理され、強化されれて、addons に移植されている。[*1]

次の E.SHow の関数やフォーラムでの議論も整理され、収録されている。

- Ｊには Hyprergeometric Function(超幾何分布)*H.* が原始関数として組み込まれている。これは K.E.Iverson が Graham Knuth Patashnik(GKP) の名著 Concrete Mathematics の Comparison を著したときに組み込まれたものであるが、用法は難解であった。
- 2001 年に Ewart Shaw は H. の解説と便利な関数群を英国 APL 協会の機関誌 Vector に公表した。
- addons/stats/distribs に収録されているハイエンドな関数は *coclass* _**pdistribs**_ が指定されている。

鈴木のスクリプト 鈴木義一郎「Ｊ言語による統計分析」(1996 森北出版) はスクリプトも付いた名著であるが、絶版となっており、入手が難しい。確率関係のスクリプトを可能な範囲で最新バージョン対応を図り、収録した。

[*1] 双方に入っているが addons の方が新しい。

1 summary

1.1 スクリプトファイル一覧

1.1.1 addons

addons/stats/base → distribution.ijs

addons/stats/base → random.ijs

addons/stats/distribs → normal.ijs

addons/stats/distribs → uniform.ijs

次のようなファイルを作成しておくと一括ロードが出来て便利である。

```
require 'plot numeric trig'
NB. --addons------------------------
require jpath '~addons/stats/base/distribution.ijs'
require jpath '~addons/stats/base/random.ijs'
require jpath '~addons/stats/distribs/normal.ijs'
require jpath '~addons/stats/distribs/uniform.ijs'
```

1.1.2 my_plot_main.ijs

```
NB. ---myprob--------------------------------
DIR_PROB=: jpath '~user/classes/calculus/statistics/prob/'
require DIR_PROB,'prob_foreign_0.ijs'
require DIR_PROB,'prob_foreign_1.ijs'
require DIR_PROB,'my_prob.ijs'
```

用語

pdf	$probability\ dencity\ function$	確率密度関数
pmf	$probability\ mass\ function$	離散的確率関数
cdf	$cumulative\ distribution\ function$	累積分布関数

J6 からメルセンヌ乱数を採用して強化されている。

1.2 分布関数と乱数一覧

1.2.1 サマリー (distrib.ijs は次表)

項目	確率分布	作者	乱数	作者
一様 (乱数)			? ?.	J primitive
			rand01 rand11	addons/stats/base
			randunif	C.Reiter
二項分布	binom bden	G.Suzuki		
	loop_bden	M.Shimura		
	binmpf bincdf	E.Shaw		
	binomialdist binomialprob	addons/stats/base	binomialrand	addons/stats/base
ポアソン分布	pden	G.Suzuki		
	poissonpmf poissoncdf	E.Shaw		
	poissondist poissonprob	addons/stats/base	poissonrand	addons/stats/base
正規分布	nden ndens	G.Suzuki	nrand	G.Suzuki
	stnormal NP NQ Ndist	J.Takeuchi	Rndm_Norm	J.Takeuchi
	n01pdf n01cdf	E.Shaw		
	normalprob normalcdf	addons/stats/base	normalrand	addons/stats/base
	rno	N.Thomson		

χ^2	chisqcdf	addons/stats/base		
t	tcdf	E.Shaw		
F	fcdf	E.Shaw		
Γ			gammarand	addons/stats/base
β			betarand	addons/stats/base
exponential			exponentialrand	addons/stats/base
cauchy			cauchyrand	addons/stats/base
discrete			discreterand	addons/stats/base
complex random			rcom	N.Thomson

1.3　disrtibs の関数一覧

ここはハイエンドクラスで

coclass 'pdistribs'　が指定されており、ロケール

`_pdistribs_`が要求される関数も多い

normal	tomusigma	$N[0,1] \to N[\mu,\sigma]$
	tostd	$N[\mu,\sigma \to N[0,1]$
	runif01	unifiorm random derivarive
	vftxt	numeric vector for txt
	ratpoly	rational polinomial approxiation
	ndx	
erf	erf	
	erfc	
	erfinv	inverse erf
normal	dnorm01	standard normal PDF
	pnorm01	standard normal CDF
	pnorm01_f	pnorm01 ref.Abramovits&Stegum
	qnorm01	inverse pnorm01
	qnorm_ut	upper tail version of qnorm
	rnorm	random deviates from normal distribution
uniform	fromunif01	$U[0,1] \to U[min,max]$
	tounif01	$U[min,max] \to U[0,1]$
	dunif01	PDF for U[0,1]

	qunif01	CDF for U[0,1]
	runif01	random deviates from U[0,1]
	dunif	Uniform probavility dencity function

1.3.1 steps

ここで、*steps* 関数を掲げておく。*steps* はファイル *numeric* に入っているので、

`require 'plot numeric trig'` とロードする。

Syntax: steps 0 1 10 // 0 to 1 with step 10

2　確率と確率分布

		mean deviation
二項分布 $B(n,p)$	$p(x) =_n C_x p^x (1-p)^{n-x}$ 二項係数は次のとおり $\left(\begin{array}{c} n \\ x \end{array}\right) =_n C_x = \frac{n!}{x!(n-x)!} (x = 0, 1, \cdots, n)$	平均 np 分散　$np(1-p)$
ポアソン分布	$np \equiv \mu$ $\frac{\mu^x}{x!} e^{-\mu}$ $p(x) = \frac{\lambda^x}{x!} e^{-\lambda}$	平均　λ 分散　λ
超幾何分布	非復元抽出 $H_{n,m,r}(k) = \frac{{}_m C_k \cdot {}_{n-m} C_{r-k}}{{}_n C_r}$	平均 $\mu = np$ 分散 $np(1-p)\frac{N_1 + N_0 - n}{N_1 + N_0 - 1}$

2.1　二項分布

2.1.1　2 項分布の Script

項目	確率分布	作者	乱数	作者
二項分布	binom bden	G.Suzuki		
	loop_bden	M.Shimura		
	binmpf bincdf	E.Shaw		
	binomialdist binomialprob	addons/stats/base	binomialrand	addons/stats/base

2.1.2　解説

二項分布の式

$p(x) =_n C_x p^x (1-p)^{n-x}$

成功　$p\{S\} = p,$

失敗　$p\{F\} = 1 - p = q$

失敗の確率　$p^x(1-p)^{n-x}$

n 試行回数

k　成功率　(成功回数/実行回数）　　（S の回数,F=n-k）

二項係数　${}_nC_x = \dfrac{n!}{x!(n-x)!}$

ベルヌーイ試行　独立な同一確率事象の繰り返しになっている試行

${}_nC_x$ に $p^x(1-p)^{n-x}$ をかけると

${}_nC_x p^x(1-p)^{n-x}$

■Example **とパラメータ** .

```
   4 bden 0.5   NB. B(4,0.5)
0.0625 0.25 0.375 0.25 0.0625

   4 binom 0.5  NB. B(4,0.5)
0625 0.25 0.375 0.25 0.0625
```

bden は 0.001 以下をまとめている。

```
 (i.5),. 4 0.5 binmpf i.5
0 0.0625
1   0.25
2  0.375
3   0.25
4 0.0625
```

```
 (i.4),.  4 0.5 bincdf i.4
0 0.0625
1 0.3125
2 0.6875
3 0.9375
```

```
   binomialdist 0.5 4
0.0625 0.25 0.375 0.25 0.0625

NB.  0  =  probability of success in one trial
NB.  1  =  number of trials
```

```
   binomialprob 0.25 100 30 40
0.149217

NB.   0  =  probability of success in one trial
NB.   1  =  number of trials
NB.   2  =  minimum number of successes
NB.  {3} =  maximum number of successes

  0.5 binmpf i.10
0.707107 0.353553 _0.0883883 0.0441942 _0.0276214
0.019335 _0.0145012 0.0113938 _0.00925747 0.00771456

 0.2   bincdf i.6
0.956352 1.00417 0.999388 1.00011 0.99998 1
```

■Worked Examples　.

1	事象Aの起こる確率 $P(A)$ は $\frac{1}{3}$
	$n = 5$, この場合の W_1, W_3
	$B(5, \frac{1}{3})$
	$W_1 = {}_5C_1 \left(\frac{1}{3}\right)\left(\frac{2}{3}\right)^4 = 0.329$
	$W_3 = {}_5C_3 \left(\frac{1}{3}\right)^3 \left(\frac{2}{3}\right)^2 = 0.165$
		`_1 x: 5 binom 1r3`
		
		`0.131687 0.329218 0.329218`
		`0.164609 0.0411523 0.00411523`
		`_1 x:  (5 1r3) binmpf i.6`
		`0.131687 0.329218 0.329218`
		`0.164609 0.0411523 0.00411523`
		
		` (5 1r3) bincdf i.5`
		`NB. cumurative`
		`0.131687 0.460905 0.790123`
		`0.954733 0.995885`

2	コインを 3 回投げた場合の表の出る組み合わせ
	$B(3, 0.5)$
		`(i.4),. 3 binom 0.5`
		`0 0.125`
		`1 0.375`
		`2 0.375`
		`3 0.125`
		
		`(3 0.5) binmpf i.4`
		`0.125 0.375 0.375 0.125`

2	n 秒間に m 歩右へ移動する方法は ${}_nC_m$ とおりある。
	n の値が与えられたときのランダムウオークの方法は等しい可能性を持っているから、n 秒後に m 歩だけ移動している確率は
	$P_n(m) = \frac{{}_nC_m}{2^n}$
	に等しい。
	（コルモロゴフ）
		100 秒間で 50 歩移動する確率
		$P_{100}(50) = \frac{{}_{100}C_{50}}{2^n} = 0.0796$
		`(50!100)%2^100`
		`0.0795892`
		
		`1 x:   (50!100)%2^100`
		`400825339491r5036175149422`

*2 *3

2.1.3　二項分布のグラフ

```
10 binom 0.5
0.000976563 0.00976563 0.0439453 0.117188 0.205078 0.246094
0.205078 0.117188 0.0439453 0.00976563 0.000976563
```

■B(n,p)　n 個数　p 確率

```
'key 1r4 1r3 1r2 2r3 3r4' plot ;("1) ,.50 binom L:0 {@>1r4 1r3 1r2 2r3 3r4
```

```
  a=. 10;20;30;40;50
+--+--+--+--+--+
|10|20|30|40|50|
```

*2 1r3 is $\frac{1}{3}$

*3 x: Extended precision _1 x: は分数を実数で表記

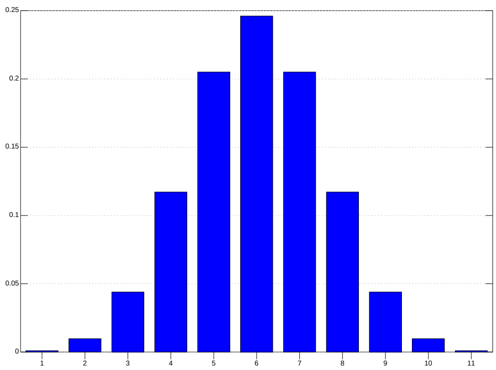

図 1　二項分布 n=10 /p=0.5

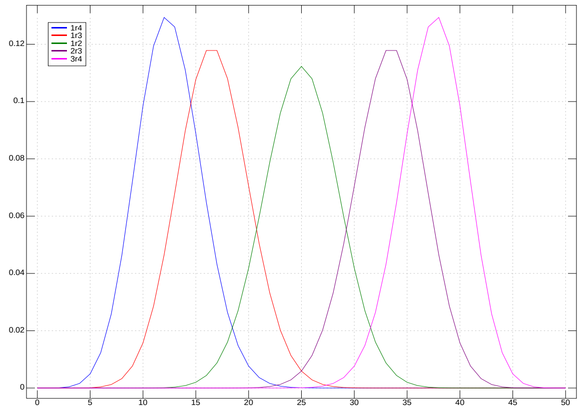

図 2　二項分布 n=10 /p=0.25-0.75

```
+--+--+--+--+--+

 plot ;("1) ,. a binom (L:0)0.5
```

個数が大きくなるほど平べったくなる。

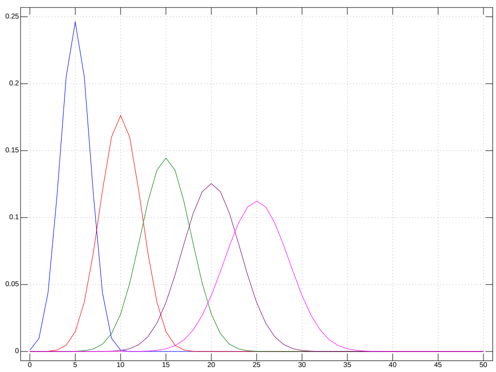

図 3　二項分布 n=10 20 30 40 50 /p=0.5

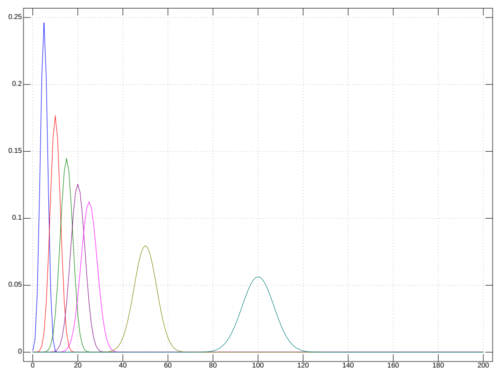

図 4　二項分布 10 20 30 40 50 100 200/P=0.5

2.1.4　Permutation and combination

電子や情報の 0/1、勝負の裏表、政治の当落や YES/NO、犯罪の白黒、勝ち負け...

0/1 で表される世界がある。

この様な結果が AB のどちらかとなる試行を**ベルヌイ試行**という。

ベルヌイ試行は結果の割合を問わない。

二項分布はスキーで言えばボーゲン。パスカルやフェルマーが基礎を築いた。

階乗	${}_nP_n = n!$	`!5` `120`	${}_nP_n = n(n-1)(n-2)\cdots 3\cdot 2\cdot 1 \equiv !$
Permutation 順列	${}_nP_r = \dfrac{n!}{(n-r)!}$	`(!5)%!5-3` `60`	n 個の中から r 個重複せずに取り出す。順序を考慮

combination
組合せ
	${}_nC_r = \dfrac{n!}{(n-r)!r!}$	`3!5`
		`10`
		
		`( !5) % (!5-3)*!3`
		`10`
			n 個の中から r 個をとる組み合わせ。順序は考慮しない。

2.1.5　J と permutiation combination

左引数 x は `i.!x` で全ての組み合わせを採る。A. は辞書式順序/Anagram [*4]

```
 (i.!3)A. i.3
0 1 2
0 2 1
1 0 2
1 2 0
2 0 1
2 1 0
```

2.1.6　Pascal の三角形

でパスカルの三角形が簡単に作れる。! は階乗/factorial

- $(p+q)^3 = p^3 + 3p^2q + 3pq^2 + q^3$
  $(p+q)^3 ={}_3 C_0 q^3 +_3 C_1 pq^2 +_3 C_2 p^2 q +_3 C_3 p^3$

- ニュートンの 2 項式

[*4] 綴り換えゲームの意味もある。

```
   pascal 6
1 0  0  0 0 0
1 1  0  0 0 0
1 2  1  0 0 0
1 3  3  1 0 0
1 4  6  4 1 0
1 5 10 10 5 1
```

$$
\begin{array}{ccccccc}
 & & {}_1C_0 & & {}_1C_1 & & \\
 & & & \backslash / & & & \\
 & {}_2C_0 & & {}_2C_1 & & {}_2C_2 & \\
 & & \backslash / & & \backslash / & & \\
{}_3C_0 & & {}_3C_1 & & {}_3C_2 & & {}_3C_3
\end{array}
$$

ランダムウオークとパスカルの 3 角形
(4) 歩で原点に戻る方法は 6 通り、(6) 歩では 20 通り、(8) 歩では 70 通りある。

```
  pascal2 9
1 1  1  1  1  1  1 1 1
1 2  3  4  5  6  7 8 0
1 3  6 10 15 21 28 0 0
1 4 10 20 35 56  0 0 0
1 5 15 35 70  0  0 0 0
1 6 21 56  0  0  0 0 0
1 7 28  0  0  0  0 0 0
1 8  0  0  0  0  0 0 0
1 0  0  0  0  0  0 0 0
```

1	1	1	1	1	1	1	1	1
1	**(2)**	3	(4)	5	(6)	7	(8)	
1	3	**6**	10	15	21	28		
1	4	10	**20**	35	56			
1	5	15	35	**70**				
1	6	21	56					
1	7	28						
1	8							
1								

$$(a+b)^n = a^n +_n C_1 a^{n-1} b + \cdots +_n C_m a^{n-m} b^m + \cdots + b^n$$

2.1.7　Script

```
NB. Kormogorov P13
pascal=: 3 : 0
tmp=:}."1 |: (<:i.y)!/i. y
tmp,.(0,|.{.tmp)
)

pascal2=: 3 : '(i.y)|."0 1 |: pascal y'
```

2.2　ポアソン分布

ポアソン分布	pden	G.Suzuki	pden μ
	poissonpmf poissoncdf	E.Shaw E.Shaw	μ $poissonpmf\ i.x$ μ $poissoncdf\ i.x$
	poissondist poissonprob	addon addon	

pdf probability　dencity function 確率密度関数
pmf　probability mass function 離散的確率関数
cdf は　cumulative distribution function 累積分布関数

起こることの極稀な事象（ｐが小さい）に対する多数回試行（ｎが大きい）場合は二項分布はポアソン分布で近似できる。

２項分布で n が大きく p が小さい極限の場合。

$np \equiv \mu$　は有限とする。

$W_x = \dfrac{\mu^x}{x!} e^{-\mu}$

2.2.1　J Script

```
pden=: 3 : 0
p=.(^-y )*(y ^k)%!k=.i.5 +>.+: y
p,1-+/p=.(+/p>1e_3){.p
)
```

0.001 以下は丸めてある

■Worked Example　issue: 一石

不良率が 0.1% の LED
100 個入りの袋の中に不良品が少なくとも一個は行っている確率。
$\mu = 0.001 \times 100 = 0.1$

$P_\mu(r) = \dfrac{\mu^r}{r!} e^{-\mu}$

不良品がない：　$P_{0.1}(0) = \dfrac{0.1^0}{0!} e^{-0.1} = \dfrac{1}{1} \times 0.904837 = 0.904837$

不良品が 1 個含まれる：　$P_{0.1}(1) = \dfrac{0.1^1}{1!} e^{-0.1} = \dfrac{0.1}{1} \times 0.904837 = 0.0904837$

不良品が 2 個含まれる：　$P_{0.2}(1) = \dfrac{0.1^2}{2!} e^{-0.1} = \dfrac{0.01}{2} \times 0.904837 = 0.00452419$

```
     pden 0.1
0.904837 0.0904837 0.00452419 0.000154653


   0.1 poissonpmf i.5
0.904837 0.0904837 0.00452419 0.000150806 3.77016e_6


   0.1 poissoncdf i.5
0.904837 0.995321 0.999845 0.999996 1

    poissondist 0.1 5
0.904837 0.0904837 0.00452419 0.000150806 3.77016e_6 7.54031e_8


NB.   0 = mean of distribution
NB.   1 = maximum value to show

    poissonprob 0.1 2
0.00467884


    poissonprob 0.1 1
0.0951626


NB.  0   =  mean of distribution
NB.  1   =  minimum number of successes
NB.  {2} =  maximum number of successes
```

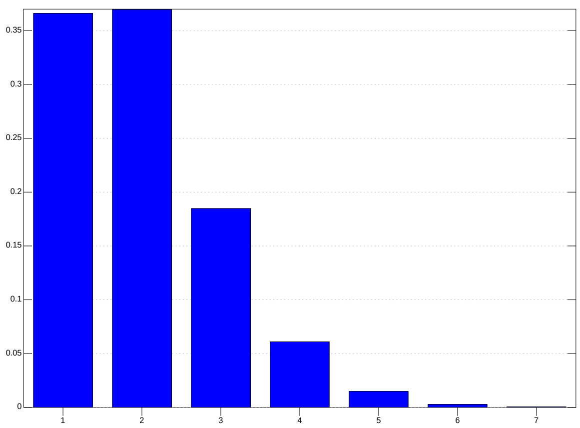

図 5　B(100,0.01)

2 項分布とポアソン分布の比較（1% ポイント）

```
(pden 1),.  100 bden 0.01
   0.367879     0.366032
   0.367879      0.36973
    0.18394     0.184865
  0.0613132    0.0609992
  0.0153283    0.0149417
 0.00306566   0.00289779
0.000594185 0.000534534
```

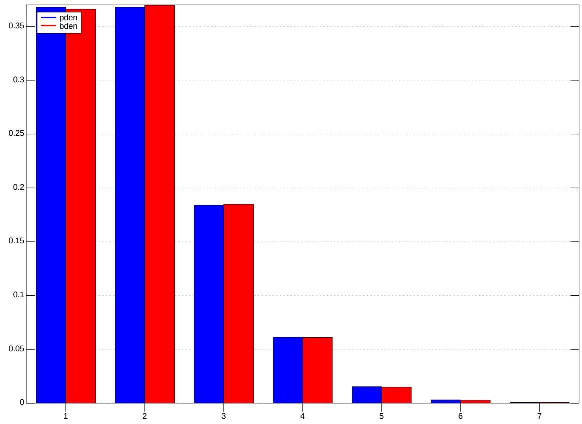

図 6　poisson binominal

2.2.2　Worked Examples

<table>
<tr><td>単位時間あたりの原子核崩壊のモデルは、二項分布の極限であるポアソン分布で近似できることが知られている。</td><td>$p = \frac{(\frac{1}{N})}{(\frac{1}{\lambda})} = \frac{\lambda}{N}$
$B(N,p) = B(N, \frac{\lambda}{N})$
$N \rightarrow \infty$
$B(N,p) = B(N, \frac{\lambda}{N}) \rightarrow p(\lambda)$</td></tr>
<tr><td>一年間の一日当たり交通事故の死亡者数の割合。
（乱数で作成したサンプル）
$\begin{array}{c|cccccc} fatal & 0 & 1 & 2 & 3 & 4 & 5 \\ day & 114 & 149 & 64 & 28 & 3 & 0 \end{array}$
$\mu = \frac{1}{365}\sum x \times 度数 = \frac{404}{365} = 1.1068$</td><td><code>100 bden 0.011068 NB.</code> 分数と差異あり

<code>0.328581 0.367744 0.203729 0.0744837</code>
<code>0.0202151 0.0043439 0.000902413</code>

<code>(i.7),. _1 x: pden 404r365</code>

<code>0 0.330599</code>
<code>1 0.365923</code>
<code>2 0.202511</code>
<code>3 0.0747164</code>
<code>4 0.0206749</code>
<code>5 0.00457681</code>
<code>6 0.000998831</code></td></tr>
</table>

<table>
<tr><td>クラス（５０人）に Morzwalt と誕生日 (1756/1/27) が同じ人がいる確率</td><td>

```
(i.4),. (50 bden 1%365),.pden 50%365

    B(50 1/365)  p(50/365)
-------------------------
0    0.871818    0.871982
1    0.119755     0.11945
2  0.00806045  0.00818148
3 0.000366038 0.000386736
```

</td></tr>
</table>

2.3　指数分布とポアソン分布

シュミレーションでよく用いるランダム到着の確率事象は、正規回数の側で見れば、ポアソン分布であり、時間感覚で見れば、指数分布になる。 [**?**]

分子の運動も指数分布となる。指数分布は無記憶性でありシュミレーションに活用される。

$f(x) = \lambda e^{-\lambda x}$

$E(x) = \dfrac{1}{\lambda}$

$V(x) = \dfrac{1}{\lambda^2}$

J の $/addons/stats/base/random.ijs$ に $exponentialrand$ が含まれている。$exponential(0.1, N)$ に固定されており、N のみが指定できる。

```
0.001 plot_count0  1r59* exponentialrand 3000

   'stick' plot 10 hist_count exponentialrand 3000
```

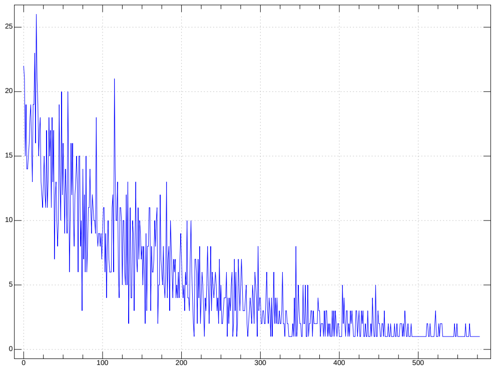

図 7　指数分布

2.4　正規分布

項目	確率分布	作者	乱数	作者
正規分布	nden ndens	G.Suzuki	nrand	G.Suzuki
	stnormal NP NQ Ndist	J.Takeuchi	Rndm_Norm	J.Takeuchi
	n01pdf n01cdf	E.Shaw		
	normalprob normalcdf	addons/stats/base	normalrand	addons/stats/base
	rno	N.Thomson		

ド・モアブルやラプラスはｎの値だけが大きい 2 項分布の確率の求め方から, 今日正規分布といわれる確率密度関数を見出した。
ガウスは, 偶然誤差の研究から, 正規分布を特徴づけた。

観測値の個数がある程度大きい場合は,（2 項分布などの諸分布は,）正規分布で近似できる。中心極限定理と呼ばれる命題である。

著作による発表はラプラスの方が早く、後からガウスが研究ノートを示して、私の方が早くから使っていたと主張した。

正規分布	$\frac{1}{\sqrt{2\pi}\sigma}e^{-(x-\mu)^2/2\sigma^2}$

```
plot n01pdf steps _5 5 1000
```

$\frac{1}{\sqrt{2\pi}\sigma}e^{-(x-\mu)^2/2\sigma^2}$

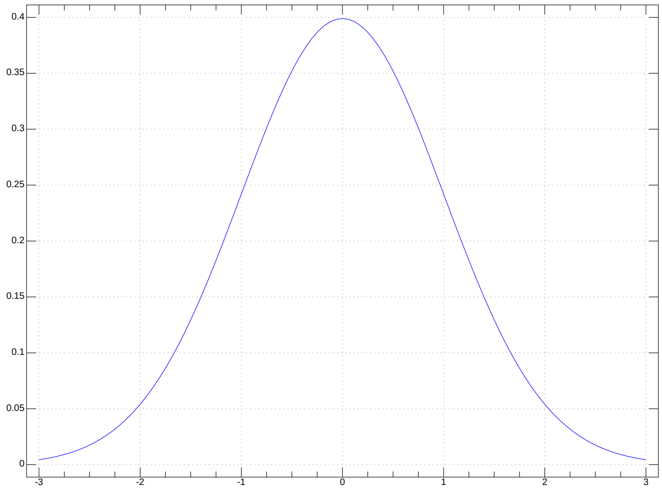

図 8　plot n01pdf steps _5 5 1000

2.4.1　J　Script

```
ndens=: 3 : 0
(^--:*:y )%%:o.2
)
```

2.4.2　実行例

標準正規分布の確率密度関数の-2 -1 -0.5 0 0.5 1 2 での値。

標準正規分布の確率密度関数の-2 -1 -0.5 0 0.5 1 2 での値。

```
   ndens _2 _1 _0.5 0 0.5 1 2
0.053991 0.241971 0.352065 0.398942 0.352065 0.241971 0.053991

  n01pdf _2 _1 _0.5 0 0.5 1 2
0.053991 0.241971 0.352065 0.398942 0.352065 0.241971 0.053991
```

左引数で与えた平均と分散を持つ正規分布の確率密度

```
  0 1 nden 0 0.5 1 2
0.398942 0.352065 0.241971 0.053991

  1 4 nden 0 0.5 1 2
0.176033 0.193334 0.199471 0.176033
```

```
   ndens 0 0.5 1 2
0.398942 0.352065 0.241971 0.053991

   n01cdf _15 _8 _7 0 1.96 7
3.67097e_51 6.22096e_16 1.27981e_12 0.5 0.975002 1

   normalcdf _15 _8 _7 0 1.96 7
_3.33067e_16 1.38778e_15 1.27998e_12 0.5 0.975002 1
```

2.4.3 正規分布の累積分布テーブル

教科書に載っている正規分布のテーブルには３のタイプがある。何れのテーブルも 簡単に作成できる。

z	a	b	c
.0	.5000	.0000	.5000
1.0	.8413	.3413	.15866
2.0	.9772	.4772	.022750
3.0	.9987	.4987	.0013499

```
   n01cdf 0 1 2 3
0.5 0.841345 0.97725 0.99865

   normalcdf 0 1 2 3
0.5 0.841345 0.97725 0.99865

 0.5-~  n01cdf 0 1 2 3
0 0.341345 0.47725 0.49865

  1- n01cdf 0 1 2 3
0.5 0.158655 0.0227501 0.0013499


  a=. (<7.2":100%~i.10),:< 0 10 20{7.4 ": 30 10 $  n01cdf steps 0 3 300
  b=.(<'z'),:<0.0,1.0,:2.0
  b,.a
+-+-----------------------------------------------------------------+
|z|   0.00    0.01    0.02    0.03    0.04    0.05    0.06    0.07    0.08    0.09|
```

```
+-+-----------------------------------------------------------------------+
|0| 0.5000 0.5040 0.5080 0.5120 0.5160 0.5199 0.5239 0.5279 0.5319 0.5359|
|1| 0.8413 0.8438 0.8461 0.8485 0.8508 0.8531 0.8554 0.8577 0.8599 0.8621|
|2| 0.9772 0.9778 0.9783 0.9788 0.9793 0.9798 0.9803 0.9808 0.9812 0.9817|
+-+-----------------------------------------------------------------------+
```

2.5　超幾何分布と誤差関数 erf

超幾何分布は二項分布の親である。

幾何分布	事象 E の起こる確率が p であるとき、何回か試行して r 回目にその事象が出る確率	$G_p(r) = q^{r-1}p, \ \ (q = 1 - p)$ 復元抽出　${}_nC_r$
超幾何分布	非復元抽出である。	非復元抽出 $H_{n,m,r}(k) = \dfrac{{}_mC_k \cdot {}_{n-m}C_{r-k}}{{}_nC_r}$

検体抽出の場合、ｎが１０００個などのオーダーになると、復元しなくとも大勢に影響が無くなる。

赤玉　$p = \dfrac{m}{n}$, 白玉　$1 - p = \dfrac{n-m}{n}$ の確率を

超幾何分布 $H_{n,m,r}(k) = \dfrac{{}_mC_k \cdot {}_{n-m}C_{r-k}}{{}_nC_r}$ に代入して計算すると

${}_rC_k p^k q^{r-k}$ と２項分布になる。

従って、超幾何分布を用いて、色々な２項分布、正規分布の近似ができ、多くの確率分布の累積分布関数 (CDFs) が作り出せる。級数展開により、積分の計算を容易にしている。

2.5.1　erf

接続詞 $H.$ を用いた erf と正規分布の定義 (E.Show による）

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

$$\frac{d}{dx} erf(x) = \frac{2}{\sqrt{\pi}} e^{-x^2}$$

(E.Show)

```
erf=: (*&(%:4p_1) % ^@:*:) * [: 1 H. 1.5 *:
n01cdf=: [: -: 1: + [: erf %&(%:2)   NB. CDF of N(0,1)
```

標準正規分布関数との関係

$\Phi(x) = \frac{1}{2}\left(1 + erf\left(\frac{x}{\sqrt{2}}\right)\right) = \frac{1}{2}erfc\left(-\frac{x}{\sqrt{2}}\right)$

```
tmp,. erf  tmp=.(i.11)%10
  0        0
0.1 0.112463
0.2 0.222703
0.3 0.328627
0.4 0.428392
0.5   0.5205
0.6 0.603856
0.7 0.677801
0.8 0.742101
0.9 0.796908
  1 0.842701
```

```
plot _3 3;' erf'
```

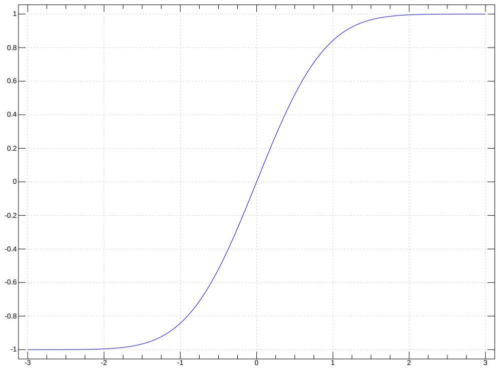

図 9　erf

- 誤差関数 $(erf(x))$ はガウス分布関数を 0 から x まで積分したものとして定義される。
- $erfc(x)$ 相補誤差関数
  $erfc(x) = \frac{2}{\sqrt{\pi}}\int_0^{\infty} e^{-t^2}dt = 1 - erf(x)$
- 不完全ガンマ関数を用いた定義
  $erf(x) = 1 - \frac{\Gamma\left(\frac{1}{2}, x^2\right)}{\sqrt{\pi}}$
- Φ の逆関数はプロビット関数である
  $probit(p) = \Phi^{-1}(p) = \sqrt{2}erf^{-1}(2p-1) = -\sqrt{2}erfc^{-1}(2p)$
  *5
- 複素誤差関数
  $w(x) = e^{-x^2}erfc(-ix)$
- erf のロケール

```
 erfc_pdistribs_
erfinv_pdistribs_
```

*5 MATLAB の erf はアルゴンヌ国立研究所の W.J.Cody 作成の Fortran プログラムを変換したものと注記されている

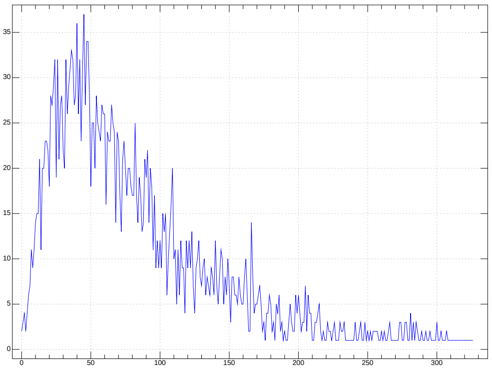

図 10　対数正規分布

2.6　対数正規分布

$$lnN(\mu, \sigma^2, N) = \frac{1}{x\sigma\sqrt{2\pi}} exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right)$$

対数を用いるので確率変数 x の範囲は $(0<x<\infty)$ で負の値にはならない。
正規分布 $N(\mu, \sigma^2)$ を打ち出し、$s = \sigma t + \mu$ を求めて e^s で作成する。

```
plot 0 hist_count 3000  rand_normal_ln 4.345 0.48
```

2.6.1　Script

```
NB. ===========Norman thomson========================
rnd=: (%˜?)@(1e9&(#˜)) NB. uniformed random number (0 1)
run=: +‘(*rnd)/
rn=:-:-+/@run@:(0 1&,)NB. Normal distribution
NB. Usage: rno a b c//a(mean) b(standard deviation c(number)
NB. e.g.  nro 2 5 100 is N(2 5 100)
rno=: 3 : ’({. y )+(1{y )*({: y )(rn &>@#)12’
NB. ----------------------------------------

 rand_normal_ln=: 4 : 0
NB. y is lnN( 4.345 0.48)
NB. y is N (rno 0 1 N)
’MU VAR ’=: y
```

```
NML=: rno ; 0 1, x
1x1^ MU + (%: VAR)* NML
)
NB. grouping for histgram
hist_count=: 3 : 0
NB. usage:  plot (100) hist_count rno 4.3 0.48 10000
NB. x is pitch(times)// y is random number
; # L:0 (~: <. tmp)<;.1 tmp=. /:~ y
:
; # L:0 (~: <.&.(x &*) tmp)<;.1 tmp=. /:~ y
)
```

2.7 χ^2

```
    5 chisqcdf 0.831211 4.35146 11.0705 20.515
0.025 0.5 0.95 0.999
```

x 自由度
y （表の）値
出力は自由度 x での α の位
0.025 0.5 0.95 0.999 is 0.975 0.5 0.050 0.001

```
  NB. c chisqcdf y      chi-squared on n d.f.
chisqcdf=: incgam&-:
```

2.8 t-value と p-value

2.8.1 t-value

Guiness brewery の化学、数学の技師であった William Sealey Gosett(1876-1937) が少量のサンプルで出来るビールの品質管理の研究を Student のペンネームで発表したものが t-distribution t-分布である。

t-value においても、χ^2 と同じく自由度と t 値 を示して、α の位をえる。表に置き換える場合には端数はラウンドする。

```
   5 tcdf 0.727 1.476 2.015 2.575 3.365 4.032
```

```
0.750088 0.900015 0.949997 0.975134 0.990001 0.994999
```

0.750088 0.900015 0.949997 0.975134 0.990001 0.994999
is 　0.25 0.10 0.05 0.025 0.01 0.005

```
NB. (n1,n2)fcdf y    F on (n1,n2) d.f.
fcdf=: -:@[ incbet ({. % +/)@(* ,:&1)
NB. n tcdf y         t on n d.f.
tcdf=: [: -:@>: *@] * 1&,@[ fcdf *:@]
```

2.8.2　p-value

最近の経済学などでは、t 値と併せて、p 値が記載されることが多い。

p value(i.e. probability value) は t-値から求められる。

p-value は帰無仮説のもとで、検定統計量の値を超える確率を示し、帰無仮説を棄却する最少の有意水準を示す。

D.Gjarati [?] p value is defined as the lowest significance level of at which a null hypothesis can rejected

例えば、自由度 (df) ８の t-value が 5.86 であった場合、t-table の α の値は 0.0010 では 5.041 である。　$tcdf$ で求めた値は 5.041 よりも大きく α は 0.001 より小さくなる。ｔ値から計算すると、ｐ値は 0.000189233 でエラーの確率は概ね 0.02 %、$\frac{2}{10000}$ であるといえる。

```
   8 tcdf 5.86
0.999811

  1- 8 tcdf 5.86
0.000189233   NB. p-value
```

2.9　F-value

Ｆ値は x に双方の自由度を、y にＦ値を入れて、％の台を求める。自由度３と１０でＦ値は 50% 5% 1% 0.1% である。

```
   3 10 fcdf 0.84508 3.70826 6.55231 12.5527
0.5 0.95 0.99 0.999
```

```
  3 10 fcdf 1.60
0.749374   NB.this is 0.25
```

2.10 Γ *and* *β*

```
     gamma
!&<:
   gamma 1
1
   gamma 1r2
1.77245
   %: 1p1
1.77245
```

2.10.1 Γ

```
   gamma  1.5 5 6 7
0.886227 24 120 720

   6 ig0 0 5 10 20 30
0 46.0847 111.95 119.991 120

   6 incgam 0 5 10 20 30
0 0.384039 0.932914 0.999928 1
```

2.10.2 Beta

```
   4 beta  2 3 4 1.5
0.05 0.0166667 0.00714286 0.101587

  4 4 2 4   beta  2 3 4 1.5
0.05 0.0166667 0.05 0.101587

   4 2 ib0 0 0.1 0.6 0.7 1
0 2.3e_5 0.016848 0.026411 0.05
```

```
   4 2 incbet 0 0.1 0.6 0.7 1
0 0.00046 0.33696 0.52822 1
```

2.10.3　others

```
   5 0.2 bincdf i.6
0.32768 0.73728 0.94208 0.99328 0.99968 _.

  5 chisqcdf 0.831211 4.35146 11.0705 20.515
0.025 0.5 0.95 0.999

  3 10 fcdf 0.84508 3.70826 6.55231 12.5527
0.5 0.95 0.99 0.999

  2.3 poissoncdf i.5
0.100259 0.330854 0.596039 0.799347 0.916249
```

登場する歴史上の数学者

フランス人の作曲家は、クープラン、ラモーから少々異端なベルリオーズまでの間が薄いが、印象派の絵画とともに、ドビッシー、ラベル、サテェーがでてきた。色彩豊かな音楽である。

Blaise Pascal	1623-1662	Heinrich Schütz	1585-1672
Abraham de Moivre	1667-1754	Johann Sebastian Bach	1685-1750
Pierre Simon,marquis Laplace	1749-1827	Franz Peter Schubert	1797-1827
Carl Friedrich Gauss	1777-1855	Ludwig van Beethoven	1770-1827
Simeon denis Poisson	1781-1840	Fernando Sor	1778-1839
August Louis Cauchy	1789-1857	Felix Mendelssohn Bartholdy	1809-1847

3　乱数

　使用するファイル

random.ijs　addons/stats/base/random.ijs
normal.ijs　addons/stats/base/normal.ijs
unoform.ijs　addons/stats/base/uniform.ijs

3.1　乱数の Count

　分布の状況をカウントする。図で確認すると分布の特徴が捉えられる。刻み (例 0.4) は任意である。

count0 plot_count0 (M.SHimura)	0.4 count0 1000 rnd 0 1 0.4 plot_count0 1000 rnd 0 1	x 0.3-0.5 is useful

```
  0.3  plot_count0  normalrand  1000
```

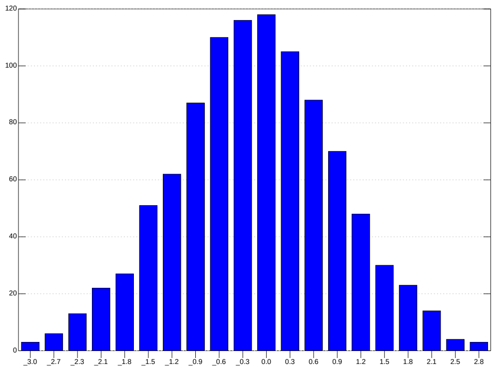

図 11　plot_count0 normalramd 1000

3.2　一様乱数

3.2.1　非重複一様乱数

```
   10?20
7 4 0 12 18 8 3 2 14 5

  rand01 5
```

```
0.0338078 0.207486 0.218838 0.0078528 0.982013

   rand11 5
0.380354 0.616439 0.497451 0.655164 _0.663936

   normalrand 5
1.77641 _0.590654 1.83539 0.545246 0.09242
```

3.2.2　C.Reiter の一様乱数

randunif (array of psuedo-random real numbers)

単項では 0,1 間の乱数を y で指定する個数 (行列でも指定できる) 2 項では x で指定する区間の乱数を生成する。

```
randunif=: (?%<:)@:($&2147483647) : ( {. @[+({:-{.)@[*$:@])

    randunif 6
0.229462 0.447347 0.787654 0.355192 0.399463 0.461342
```

2 項　x　で指定する間の乱数

```
  _1 2 randunif 2 3

   1.5792  0.54126 0.955131
_0.108855 0.471075 0.363747
```

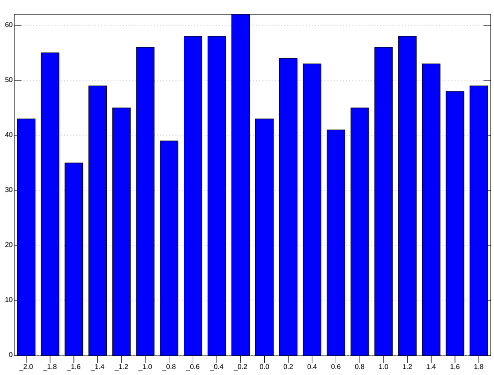

図 12　A Random walk

■Script　.

```
randunif=: (?%<:)@:($&2147483647) : ( {. @[+({:-{.)@[*$:@])
```

3.2.3　重複乱数

一様乱数は、同じパターンが再現されてしまうので、重複乱数が必要となることがある。

重複乱数は次により出すことができる。

```
   ?20#20
0 18 12 16 17 0 3 9 19 15 14 8 18 10 2 8 6 5 19 5

  ? 5 20 $ 20
 1 13 10  0 16  8  7 18  6 10 17 19  9  3  3 14 16  7  8  4
 6  1  1 15  2  8 15 14 14  4  1  2  1 18 18  5 15 18  4 14
15  0  7 11  7  5  6  9  9  7 17  9  3  5  1  4 18  2 13  0
 2 10 17 11 12 19 13  6  9  5  2  8 16  6  2  4 19 10  2  9
 8  3  3  0  1  4 12 18 17  9  7  8  8 19 11  4 16  8 16  4
```

3.3　正規乱数

3.3.1　簡便法

Ｊのプリミティブがサポートする乱数は非重複一様乱数である。これを元に正規乱数を作成する「一様乱数を１２個加えて６を引く」簡便法がある。$x_i (i = 1, 2, ..., 12)$ を $1 > x_i > 0$ であるような一様乱数としたとき、

$$z = (x_1 + x_2 + ... + x_{12}) - 6$$

z は平均０、標準偏差１の正規分布に従う乱数の近似値を与えるというものである。

鈴木の *rnd* がこの簡便法である。

```
   rnd=:4 : 0
NB. x  rnd y  (ex. 100 rnd 0 1)
TMP0=: (? (x ,12)$999)%1000
TMP1=:(0{y )+ (1{y ) * _6 + +/"1 TMP0
)
```

```
0.4  plot_count0  1000 rnd 0 1
```

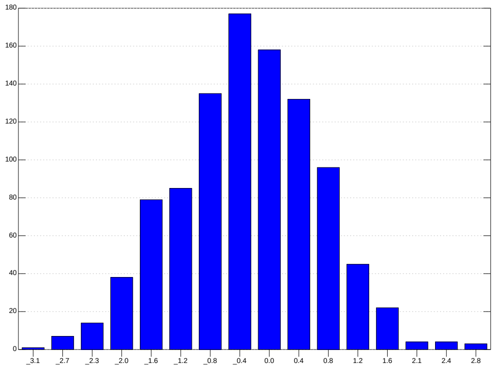

図 13 plot_count0 1000 rnd 0 1

3.3.2 Box-Muller 法

- 洗練された正規乱数発生法にボックス・ミュラー法がある。$x_1.x_2$ を区間 $0 \le x < 1$ の一様乱数とすると

$$y_1 = \sqrt{-2lnx_i} \cdot cos(2\pi x_2)$$

$$y_2 = \sqrt{-2lnx_i} \cdot sin(2\pi x_2)$$

  で計算される y_1, y_2 は標準正規分布に従う。二組の結果の内、どちらか片方を用いればよい。
- J の addons の *normalrand* はボックス · ミュラー法を採用している。
  `normalrand=:3 : '(2 o. +: o. rand01 y) * %: - +: ^. rand01 y'`
- 以下はかって作成したボックス · ミュラー法である。

```
5j2 ":  10 bm 100
 3.03 1.14 1.33 1.38 0.28 1.47 0.60 0.22 0.49 0.44
 0.04 0.57 0.57 0.84 0.23 1.06 0.68 1.93 0.55 0.25
```

J の一様乱数を 0<x<1 とする。

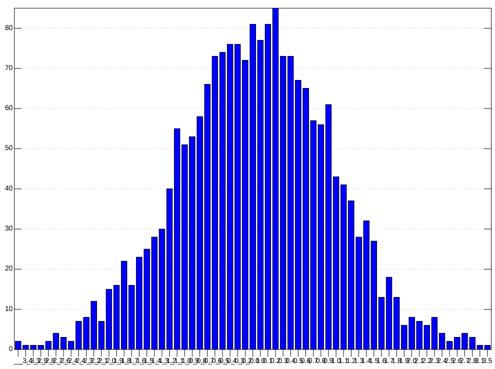

図 14　Box Muller

```
bm=: 4 : 0
NB. Cumurative random number
NB. Box-Muller
NB. Usage x  bm y (ex  10 bm 100)
BM0=:y %~ (x ?y),: (x ?y)
BM1=: 1&o. o. 0{BM0
BM2=: %: (_2)* ^. 1{BM0
BM_A=: BM1 * BM2
BM3=: 1&o. o. 1{BM0
BM4=: %: (_2)* ^. 0{BM0
BM_B=: BM3 * BM4
BM_A,:BM_B
)
```

3.3.3　任意の平均と標準偏差を指定した正規乱数

- *tomusigma*

  addons/stats/distribs/normal.ijs に *tomusignma* が入った。

  `0.4 plot_count0 5 1.2 tomusigma  normalrand 1000`

- *tostd*

  N[mu,sigma] から N[0,1] に逆変換する *tostd* もある。

  `0 1 tostd  5 1.2 tomusigma  normalrand 1000`

- *Takeuchi*

  `Rndm_Norm 100 5  1.2` NB. number mu sigma

  詳細は竹内「バリアオプション」シリーズ参照

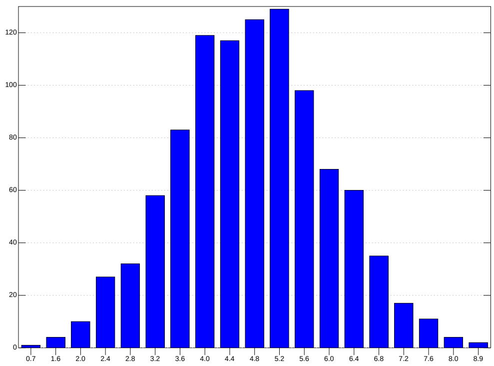

図 15　tomusigma normal

```
NB.*tomusigma v Converts from N[0,1] to N[mu,sigma]
NB. returns: rescaled numeric array adjusted by mean mu &
NB.          stddev sigma
NB. y is: numeric array
NB. x is: 2-item numeric list
NB.     0{x desired mean adjustment for array
NB.     1{x desired stddev adjustment for array
tomusigma=: 4 : 0
  'mu sigma'=. x
  mu + sigma*y
)
```

3.4 対数正規乱数

N.Thomson の正規乱数の Script

N.Thomson の著書に、任意の平均と標準偏差を与えて、その構成での正規分布を求める関数が紹介されている。これを用いて、平均 0、標準偏差 1 の正規分布を求め、上の式により対数正規分布に変換する。*6

	到着時間	サービス時間	対数サービス時間
合計	2952	4889	217.448
平均	59.54	97.78	4.34897
分散	4019.41	5895.53	0.43661

到着時間

$$\frac{F(t+dt)-F(t)}{1-F(t)} = \frac{e^{\lambda t}-e^{\lambda(t+dt)}}{e^{-\lambda t}} = 1-e^{\lambda dt}$$

現時点から dt 時間の間に次の客が到着する確率は、前の客の到着とは関係なく、常に $1-e^{\lambda dt}$ になる。

$$\frac{1}{\lambda} = 59.5$$

```
   1%59.5
0.016806
```

客の**到着時間**は、$\lambda = 0.0168$ の指数分布に従う。

サービス時間

*6 正規分布ではマイナスが現れ複素数がでる。

対数分布で、確率変数 $Y = lnX$、平均 μ、分散 σ^2 の正規分布に従うとき、サービス待ち時間の平均と分散は次のようになる。

対数分布	$f(t) = \frac{1}{\sqrt{2\pi}\sigma x} e^{\left(-\frac{(lnX-\mu)^2}{2\sigma^2}\right)}$	
平均	$e^{\mu+\frac{\sigma^2}{2}} (= e^{2\mu+\sigma^2})$	97.78
分散	$e^{2\mu+2\sigma^2} - e^{2\mu+\sigma^2}$	5895.53

$$\begin{cases} e^{\mu+\frac{\sigma^2}{2}} & = 97.78 \\ e^{2\mu+2\sigma^2} - e^{2\mu+\sigma^2} & = 5895.53 \end{cases}$$

$e^{2\mu+2\sigma^2} = 58.95 + e^{2\mu+\sigma^2}$ 平均

両辺に対数をとると

$$\begin{cases} 2\mu + \sigma^2 & = 9.16544 \\ 2\mu + 2\sigma^2 & = 9.64578 \end{cases}$$

この連立方程式を解くと、対数正規分布の平均と分散が求まる。生データがない場合は、この式により求める。

$a = 2 * ln97.78 = 2\mu + \sigma^2$

```
  2* ^. 97.78
  9.16544
```

$e^{2\mu+2\sigma^2} = e^{2\mu+2\sigma^2} - e^{2\mu+\sigma^2} + e^{2\mu+\sigma^2} = 5895.78 + e^a$

```
     5895.53 +  1x1 ^ 2* ^. 97.78
15456.5
  ^. 5895.53 +  1x1 ^ 2* ^. 97.78
9.64578
```

連立方程式をクラメール法で解く

```
  dat=. 2 3 $ 2 1 9.16544 2 2 9.64578
    dat
2 1 9.16544
2 2 9.64578


    cr=: %.}:"1
    cr dat
```

```
          1 _1.11022e_15 4.34255
1.77636e_15            1 0.48034
```

客の到着時間は、$\lambda = \dfrac{1}{59.5} = 0.0168$ の指数分布に従う。

客のサービス時間は平均　4.34255　分散 0.48034 の対数正規分布に従う。

```
33 plot_count0  ^ 4.3490 + (%: 0.4366)* rno 0 1 10000

   pd 'eps /temp/normal_exp.eps'
```

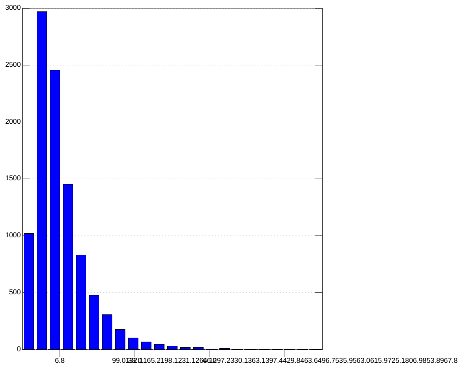

図 16　exponential normal

4　種々の乱数

4.1　2 項乱数

binomial random numbers

```
   binomialrand 0.4 20
1 1 1 1 1 0 1 1 1 0 0
1 1 1 0 1 1 0 0 0
```

NB. random numbers 0 and 1 in a binomial distribution
NB. 0 = probability of success in one trial
NB. 1 = number of trials

```
binomialrand=: 3 : 0
'p n'=. y
r=. 2147483647
s=. <:p*r
s>?n#<:r
)
```

4.2　正規乱数

normal random numbers

Box Muller Method

互いに独立な一様乱数　U_1, U_2

$Z_1 = \sqrt{-2logU_1}cos(2\pi U_2)$

$Z_2 = \sqrt{-2logU_1}sin(2\pi U_2)$

0.1 plot_count0 normalrand 10000

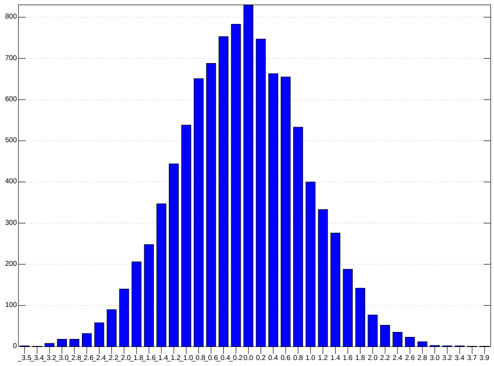

図 17　normalrand

```
normalrand=: 3 : 0
(2 o. +: o. rand01 y) * %: - +: ^. rand01 y
)
```

Z_1 is cos

NB. with mean=0, standard deviation=1.

NB. y = number of trials.

4.3 ポアソン乱数

poisson *random numbers*

$$f(x) = \frac{\lambda^x e^{-\lambda}}{x!}$$

```
poissonrand=: 3 : 0
'm n'=. y
roll=. -@^.@rand01
r=. b=. m >: t=. roll n
i=. i.n
while. #i=. b#i do.
  b=. m >: t=. (b#t) + roll #i
  r=. (b + i{r) i } r
end.
r
)
```

0.1 plot_count0 poissonrand 3 10000

NB. y has 2 elements:

NB. 0 = mean of distribution (=variance)

NB. 1 = number of trials

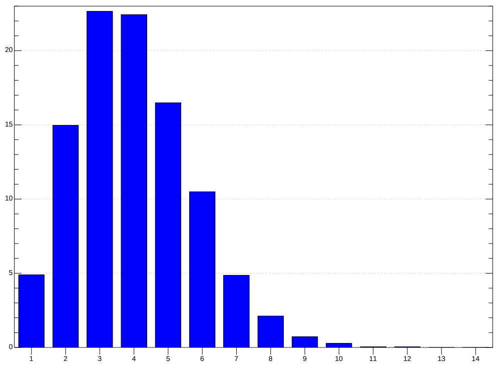

図 18 *poisson*

4.4 対数正規乱数

exponential *random numbers*

$f(x) = \lambda e^{-\lambda x}$

0.1 plot_count0 exponentialrand 10000

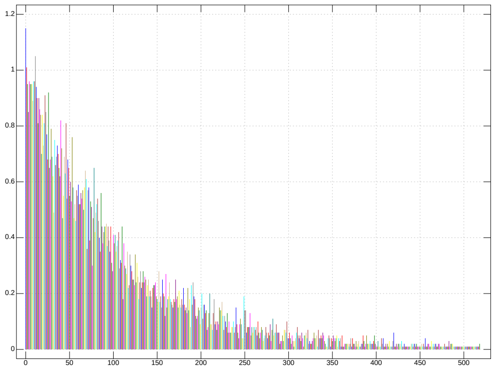

図 19 *e*

```
exponentialrand=: 3 : 0
-^.rand01 y
)
```

$f(x) = e^{-\lambda x}$

NB. random numbers in an exponential distribution

NB. with mean=1. $F(x) = 1-^{-x}$

NB. y = number of trials

4.5　ガンマ乱数

Γ *random numbers*

$f(x) = \dfrac{1}{\beta^{\alpha}\Gamma(\alpha)} x^{\alpha} e(-\dfrac{x}{\beta})$

gammarand(p,n)

p is $\lambda(=\alpha)$, β はなし）

p=1 is exponential rand /指数分布になる

独立な指数分布に従う確率変数の n 個の和である。$\alpha = 1$ の場合は平均 β の指数分布 $(\lambda = \dfrac{1}{\beta})$

α が正の整数の場合はアーラン分布である。

$a = \dfrac{m}{2}, \beta = 2$ とすると自由度 m のχ^2 分布

0.1 plot_count0 gammarand 2 10000

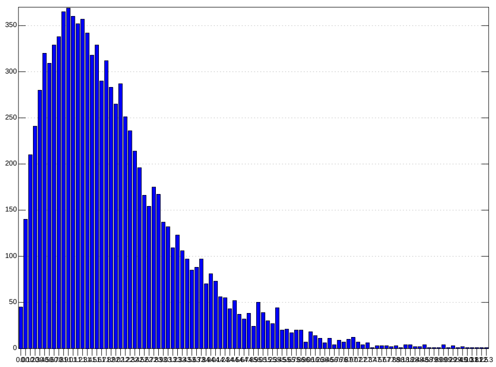

図 20　$\Gamma(\lambda = 2)$

NB. random numbers in a gamma distribution

NB. y has 2 elements p,n

NB. 0 = power parameter

NB. 1 = number of trials

NB. if p=1 this is the exponential distribution

```
gammarand=: 3 : 0
'p n'=. y
r=. n#0
k=. p-i=. <.p
if. k do.
  r=. betarand k,(-.k),n
  r=. r * -^.rand01 n
end.
if. i do.
  r-^.*/rand01 i,n
end.
)
```

0.1 plot_count0 gammarand 3 10000

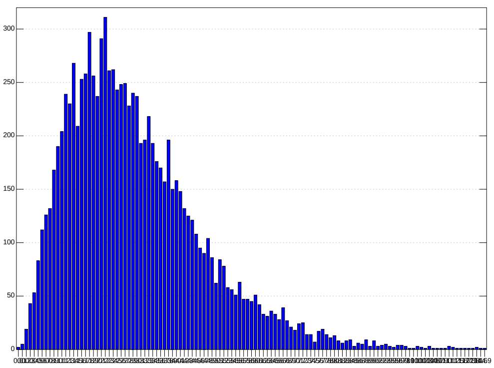

図 21　$\Gamma(\lambda = 3)$

4.6　ベータ乱数

β *randomnumbers*

$$f(x) = \frac{1}{B(a,b)} x^{a-1}(1-x)^{b-1}$$

$$where, B(a,b) = \int_0^1 t^{a-1}(1-t)^{b-1} dt$$

betarand(p1,p2,n)

p1 is λ_1, p2 is λ_2

NB. random numbers in a beta distribution

NB. y has 3 elements p,q,n

NB. 0 = power parameter

NB. 1 = power parameter

NB. 2 = number of trials

```
betarand=: 3 : 0
'p q n'=. y
if. (1>p) *. 1>q do.
  b=. n#1
  r=. n#0
  whilst. 1 e. b do.
    m=. +/b
    x=. (rand01 m)^%p
    y=. x+(rand01 m)^%q
    t=. 1>:y
    z=. (t#x)%t#y
    i=. t#b#i.#b
    b=. 0 i } b
    r=. (z+i{r) i } r
  end.
else.
  s%(gammarand q,n)+s=. gammarand p,n
end.
)
```

0.01 plot_count0 betarand 2 3 10000

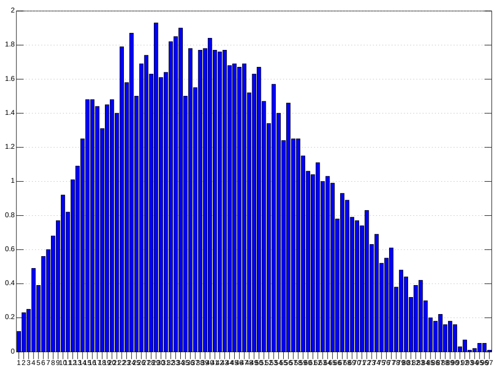

図 22　$\beta(\lambda_1 = 2,\ \lambda_2 = 3)$

plot 0.01 plot_count0 betarand 4 6 10000

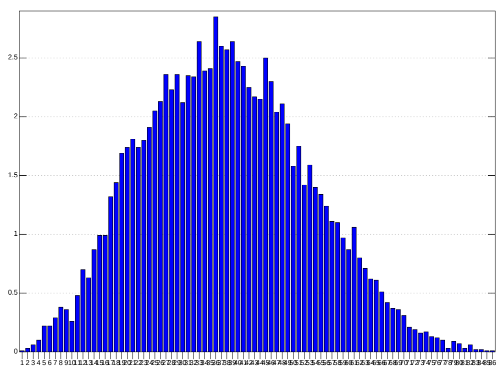

図 23　$\beta(\lambda_1 = 4,\ \lambda_2 = 6)$

4.7　コーシー乱数

'stick' plot ; 0.01 count0 cauchyrand 10000

cauchy random numbers

```
NB. random numbers in a cauchy distribution
NB. with F(x)=0.5+(arctan x)%o.1
NB. y  =  number of trials

cauchyrand=: 3 : 0
3 o. o. _0.5+rand01 y
)
```

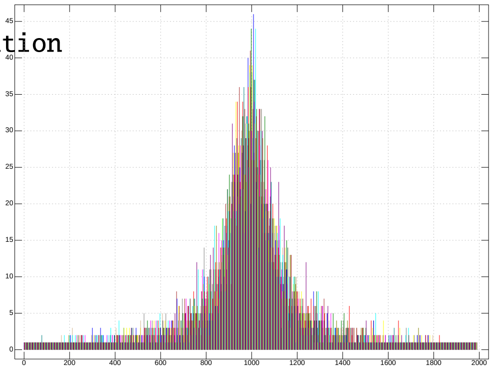

図 24　*cauchy*

```
 discreterand
 ((i.4);0.1 0.3 0.4 0.2);10

1 1 3 1 1 2 2 2 3 2
```

NB. random numbers in a discrete distribution

NB. y has two elements

NB. 0 = 2-row matrix: 0 = discrete values

NB. 1 = number of trials

5　References

コルモゴロフ · ジュルベンコ · プロホロフ（丸山 · 馬場訳）「コルモゴロフの確率論入門」　森北出版　2003

Cliford A. Reiter :Fractal Visualization and J (2000)

小西　貞則「ブートストラップ法」（村上・田村編「パソコンによるデータ解析」朝倉書店　１９８８）

東北大学統計グループ「これだけは知っておこう！統計学」有斐閣　2002

西川利男　浜田節雄「実用ＡＰＬ」日刊工業新聞社　1986

蓑谷千凰彦「金融データの統計分析」東洋経済新報社 (2001)

和達　三樹　十川　清「キーポイント確率・統計」岩波書店　1993