RjpWiki アーカイブス

【確率分布、乱数関数一覧 (06.06.25) 】

1　疑似乱数

1.1　疑似乱数発生関連関数

.Random.seed は整数ベクトルで、R の乱数生成機構に対する乱数生成器 (RNG) の状態を含んでいる。これは保存し、元に戻すことができるが、ユーザが変更すべきではない。RNGkind() は使用中の RNG の種類を確認したり、設定したりするためのより分かりやすいインタフェイスである。RNGversion() は以前のバージョンの R に於けるように乱数発生器を設定するために使うことができる (再現性のために設けられている)。set.seed() は乱数種を指定するための推薦できる方法である。

```
.Random.seed <- c(rng.kind, n1, n2, ...)
save.seed <- .Random.seed
RNGkind(kind = NULL, normal.kind = NULL)
RNGversion(vstr)
set.seed(seed, kind = NULL)
```

1.2　ユーザ定義の疑似乱数発生器

関数 RNGkind() はユーザがコードした一様疑似乱数と正規疑似乱数生成器の使用を許す。以下にその詳細を述べる。

ユーザー指定の一様 RNG は動的にロードされたコンパイル済みコードのエントリ点から呼び出される。ユーザはエントリ点 user unif rand を提供する必要があり、これは引数はなく、倍精度実数へのポインタを返す。以下の例は一般的なパターンを示す。

オプションとして、ユーザは RNGkind (もしくは set.seed) が呼び出されたとき引数 unsigned int で呼び出されるエントリ点 user unif init を提供することができ、ユーザの RNG コードを初期化するために使うことが想定されている。2 個の引数は「乱数種」を設定するのに使われることを想定している。それは set.seed への seed 引数であるか、もし RNGkind が呼び出されるならば本質的にランダムな乱数種である。

もしこれらの関数だけが提供されると、生成器に対する如何なる情報も .Random.seed に記録されない。オプションとして、引数無しに呼び出されると、乱数種の数と乱数種の整数配列へのポインタを返す関数 user unif nseed と user unif seedloc を与えることができる。GetRNGstate と PutRNGstate への呼び出しは、この配列を.Random.seed へ、そして .Random.seed からコピーする。ユーザ定義の正規 RNG は単一のエントリ点 user norm rand で指定され、これは引数を持たず、倍精度実数へのポインタを返す。

2 連続分布

2.1 一様分布

これらの関数は区間 min から max 上の一様分布に関する情報を与える。dunif() は密度関数、punif() は分布関数、qunif() はクォンタイル関数、そして runif() は乱数を与える。

```
dunif(x, min=0, max=1, log = FALSE)
punif(q, min=0, max=1, lower.tail = TRUE, log.p = FALSE)
qunif(p, min=0, max=1, lower.tail = TRUE, log.p = FALSE)
runif(n, min=0, max=1)
```

2.2 正規分布

dnorm() は平均 mean、標準偏差 sd の正規分布の密度関数、pnorm() は分布関数、qnorm() はクォンタイル関数、そして rnorm() は乱数を与える。

```
dnorm(x, mean=0, sd=1, log = FALSE)
pnorm(q, mean=0, sd=1, lower.tail = TRUE, log.p = FALSE)
qnorm(p, mean=0, sd=1, lower.tail = TRUE, log.p = FALSE)
rnorm(n, mean=0, sd=1)
```

2.3 対数正規分布

dlnorm は対数正規分布の密度関数、plnorm は分布関数、qlnorm はクォンタイル関数、そして rlnorm は乱数を与える。対数値の平均は meanlog、標準偏差は sdlog となる。

```
dlnorm(x, meanlog = 0, sdlog = 1, log = FALSE)
plnorm(q, meanlog = 0, sdlog = 1, lower.tail = TRUE, log.p = FALSE)
qlnorm(p, meanlog = 0, sdlog = 1, lower.tail = TRUE, log.p = FALSE)
rlnorm(n, meanlog = 0, sdlog = 1)
```

2.4 ガンマ分布

dgamma() はパラメータ shape と scale のガンマ分布の密度関数、pgamma() は分布関数、qgamma() はクォンタイル関数、そして rgamma() は乱数を与える。

```
dgamma(x, shape, rate = 1, scale = 1/rate, log = FALSE)
pgamma(q, shape, rate = 1, scale = 1/rate,
       lower.tail = TRUE, log.p = FALSE)
qgamma(p, shape, rate = 1, scale = 1/rate,
       lower.tail = TRUE, log.p = FALSE)
rgamma(n, shape, rate = 1, scale = 1/rate)
```

2.5 ベータ分布

dbeta はパラメータ shape1, shape2 (そしてオプションの非心度パラメータ ncp) を持つベータ分布の密度関数、pbeta は分布関数、qbeta はクォンタイル関数、そして rbeta は乱数を与える。

```
dbeta(x, shape1, shape2, ncp=0, log = FALSE)
pbeta(q, shape1, shape2, ncp=0, lower.tail = TRUE, log.p = FALSE)
qbeta(p, shape1, shape2,        lower.tail = TRUE, log.p = FALSE)
rbeta(n, shape1, shape2)
```

2.6 カイ自乗分布

dchisq() は自由度 df、そしてオプションの非心パラメータ ncf を持つカイ自乗分布の密度関数、pchisq() は分布関数、qchisq() はクォンタイル関数、そして rchisq() は乱数を与える。

```
dchisq(x, df, ncp=0, log = FALSE)
pchisq(q, df, ncp=0, lower.tail = TRUE, log.p = FALSE)
qchisq(p, df, ncp=0, lower.tail = TRUE, log.p = FALSE)
rchisq(n, df, ncp=0)
```

2.7 t 分布

dt() は自由度 df (そして非心度パラメータ ncp の) t 分布の密度関数、pt() は分布関数、qt() はクォンタイル関数、そして rt() は乱数を与える。

```
dt(x, df, ncp=0, log = FALSE)
pt(q, df, ncp=0, lower.tail = TRUE, log.p = FALSE)
qt(p, df,        lower.tail = TRUE, log.p = FALSE)
rt(n, df)
```

2.8 F 分布

df() は自由度 df1, df2 (そしてオプションの非心度パラメータ ncp) を持つ F 分布の密度関数、pf() は分布関数、qf() はクォンタイル関数、そして rf() は乱数を与える。

```
df(x, df1, df2, log = FALSE)
pf(q, df1, df2, ncp=0, lower.tail = TRUE, log.p = FALSE)
qf(p, df1, df2,        lower.tail = TRUE, log.p = FALSE)
rf(n, df1, df2)
```

2.9 コーシー分布

dcauchy() は位置パラメータ location、スケールパラメータ scale のコーシー分布の密度関数、pcauchy() は分布関数、qcauchy() はクォンタイル関数、そして rcauchy() は乱数を与える。

```
dcauchy(x, location = 0, scale = 1, log = FALSE)
pcauchy(q, location = 0, scale = 1, lower.tail = TRUE, log.p = FALSE)
qcauchy(p, location = 0, scale = 1, lower.tail = TRUE, log.p = FALSE)
rcauchy(n, location = 0, scale = 1)
```

2.10 指数分布

dexp() は割合 rate (つまり平均が 1/rate) の指数分布の密度関数、pexp() は分布関数、qexp() はクォンタイル関数、そして rexp() は乱数を与える。

```
dexp(x, rate = 1, log = FALSE)
pexp(q, rate = 1, lower.tail = TRUE, log.p = FALSE)
qexp(p, rate = 1, lower.tail = TRUE, log.p = FALSE)
rexp(n, rate = 1)
```

2.11 ロジスティック分布

dlogis() は位置パラメータ location とスケールパラメータ scale のロジスティック分布の密度関数、plogis() は分布関数、qlogis() はクォンタイル関数、そして rlogis() は乱数を与える。

```
dlogis(x, location = 0, scale = 1, log = FALSE)
plogis(q, location = 0, scale = 1, lower.tail = TRUE, log.p = FALSE)
qlogis(p, location = 0, scale = 1, lower.tail = TRUE, log.p = FALSE)
rlogis(n, location = 0, scale = 1)
```

2.12 ワイブル分布

dweibull() はパラメータ shape と scale のワイブル分布の密度関数、pweibull() は分布関数、qweibull() はクォンタイル関数、そして rweibull() は乱数を与える。

```
dweibull(x, shape, scale = 1, log = FALSE)
pweibull(q, shape, scale = 1, lower.tail = TRUE, log.p = FALSE)
qweibull(p, shape, scale = 1, lower.tail = TRUE, log.p = FALSE)
rweibull(n, shape, scale = 1)
```

2.13 スチューデント化範囲 (Studentized range) 分布

スチューデント化範囲 (Studentized range) 分布とは、R を n 個の標準正規標本の範囲、s^2 をそれと独立な自由度 df のカイ自乗分布に従う確率変数とすると、R/s の確率分布である。

```
ptukey(q, nmeans, df, nranges = 1, lower.tail = TRUE, log.p = FALSE)
qtukey(p, nmeans, df, nranges = 1, lower.tail = TRUE, log.p = FALSE)
```

3 離散分布

3.1 無作為抽出とランダムな置換

sample() は x の要素から指定したサイズの標本を復元、もしくは非復元抽出する。

```
sample(x, size, replace = FALSE, prob = NULL)
```

3.2 二項分布

dbinom はパラメータ size と prob の二項分布の確率関数、pbinom() は分布関数、qbinom() はクォンタイル関数、そして rbinom() は乱数を与える。

```
dbinom(x, size, prob, log = FALSE)
pbinom(q, size, prob, lower.tail = TRUE, log.p = FALSE)
qbinom(p, size, prob, lower.tail = TRUE, log.p = FALSE)
rbinom(n, size, prob)
```

3.3 負の二項分布

dnbinom() はパラメータ size と prob の負の二項分布の密度関数、pnbinom() は分布関数、qnbinom() はクォンタイル関数、そして rnbinom() は乱数を与える。

```
dnbinom(x, size, prob, mu, log = FALSE)
pnbinom(q, size, prob, mu, lower.tail = TRUE, log.p = FALSE)
qnbinom(p, size, prob, mu, lower.tail = TRUE, log.p = FALSE)
rnbinom(n, size, prob, mu)
```

3.4 ポアッソン分布

dpois89 はパラメータ lambda のポアッソン分布の (対数) 確率関数、ppois() は (対数) 分布関数、qpois() はクォンタイル関数、そして rpois() は乱数を計算する。

```
dpois(x, lambda, log = FALSE)
ppois(q, lambda, lower.tail = TRUE, log.p = FALSE)
qpois(p, lambda, lower.tail = TRUE, log.p = FALSE)
rpois(n, lambda)
```

3.5 幾何分布

dgeom() はパラメータ prob の幾何分布の密度関数、pgeom() は分布関数、qgeom() はクォンタイル関数、そして rgeom() は乱数を与える。

```
dgeom(x, prob, log = FALSE)
pgeom(q, prob, lower.tail = TRUE, log.p = FALSE)
qgeom(p, prob, lower.tail = TRUE, log.p = FALSE)
rgeom(n, prob)
```

3.6 超幾何分布

dhyper() は超幾何分布の密度関数、phyper() は分布関数、qhyper() はクォンタイル関数、そして rhyper() は乱数を与える。

```
dhyper(x, m, n, k, log = FALSE)
phyper(q, m, n, k, lower.tail = TRUE, log.p = FALSE)
qhyper(p, m, n, k, lower.tail = TRUE, log.p = FALSE)
rhyper(nn, m, n, k)
```

3.7 多項分布

多項分布に従う乱数ベクトルを発生し、確率関数を計算する。

```
rmultinom(n, size, prob)
dmultinom(x, size = NULL, prob, log = FALSE)
```

3.8 Wilcoxon のランク和統計量分布

dwilcox() はサイズがそれぞれ m, n の標本から得られた Wilcoxon のランク和統計量の密度関数、pwilcox() は分布関数、qwilcox() はクォンタイル関数、そして rwilcox() は乱数を与える。

```
dwilcox(x, m, n, log = FALSE)
pwilcox(q, m, n, lower.tail = TRUE, log.p = FALSE)
qwilcox(p, m, n, lower.tail = TRUE, log.p = FALSE)
rwilcox(nn, m, n)
```

3.9 Wilcoxon 符号付きランク統計量分布

サイズ n の標本に対する Wilcoxon 符号付きランク統計量の分布関数に関する情報を得る。dsignrank() は確率関数、psignrank() は分布関数、qsignrank() はクォンタイル関数、そして rsignrank() は乱数を与える。

```
dsignrank(x, n, log = FALSE)
psignrank(q, n, lower.tail = TRUE, log.p = FALSE)
qsignrank(p, n, lower.tail = TRUE, log.p = FALSE)
rsignrank(nn, n)
```

4 その他

4.1 ランダムな 2 次元配列

Patefield のアルゴリズムを用い周辺和を与えたランダムな 2 次元配列（2 次元分割表）を生成する。

```
r2dtable(n, r, c)
```

4.2 一致確率

一般化された「誕生日のパラドックス」問題への近似解を与える。pbirthday() は一致確率、qbirthday() は指定された一致確率に必要な観測数を計算する。

```
qbirthday(prob = 0.5, classes = 365, coincident = 2)
pbirthday(n, classes = 365, coincident = 2)
```

4.3 randu

VAX の FORTRAN 関数 RANDU (VMS 1.5) からとられた引き続く乱数の 400 個の三つ組