クレジット:
UTokyo Online Education 統計データ解析 Ⅱ 2018 小池祐太

東京大学 THE UNIVERSITY OF TOKYO

CC BY NC ND

統計データ解析 II (平成３０年度)

東京大学大学院数理科学研究科
統計データ解析教育研究グループ

村田　昇 (早稲田大学, 東京大学)
吉田朋広 (東京大学)
小池祐太 (東京大学)

第 4 章 多変量分布のシミュレーション

確率統計学では, 多くの場合において, 観測データをランダムな現象の実現値として捉えることによって解析を進める. この章では, そのようなランダムな現象のモデル化に利用される確率変数・確率分布の概念と, その R 上でのシミュレーション方法を扱う.

4.1. 乱数

乱数 *(random numbers)* とはランダムに生成された数列のことである. もちろん, コンピューターでは完全にランダムに数字を発生されることは不可能なため, それらの乱数は厳密には**擬似乱数** *(pseudo random numbers)* である.[1] 特に, 数値シミュレーションを行う上では, それが再現可能であることが要請されるため, 発生される乱数も再現可能である必要がある. R ではこれを実行するために, 乱数の初期値を指定するための関数 `set.seed()` が用意されている (同一の初期値から生成される乱数は同一のものとなる).

ここでは基本的な乱数として, ランダムサンプリング, 二項乱数および一様乱数を考える. ランダムサンプリングは, その名の通り, 与えられた集合の要素をランダムに抽出することで発生する乱数のことである. 二項乱数は,「確率 p で表がでるコインを n 回投げた際の表が出る回数」に対応する乱数である. 従って p と n によって乱数の発生の仕方が変わるため, それを明示するために「確率 p に対する次数 n の二項乱数」とも呼ぶ. 一様乱数は, ある決まった区間 (a, b) $(a < b)$ に含まれる数字からランダムに発生する乱数のことである.[2] 従って区間 (a, b) によって乱数の発生の仕方が変わるため, それを明示するために「区間 (a, b) 上の一様乱数」とも呼ぶ.

ランダムサンプリングは関数 `sample()` で実行できる. 二項乱数および一様乱数はそれぞれ関数 `rbinom()` および `runif()` で発生させられる.

```
> ## 関数 sample の使い方
> x <- 1:10 # サンプリング対象の集合をベクトルとして定義
> set.seed(123) # 乱数の初期値を指定
> sample(x, 5) # x から 5 つの要素を重複なしでランダムに抽出
[1] 3 8 4 7 6
> sample(x, 10) # x の要素のランダムな並べ替えとなる
 [1]  1  5  8  4  3  9  2  6  7 10
> sample(x, 5, replace = TRUE) # x から 5 つの要素を重複ありでランダムに抽出
[1]  9  3  1  4 10
> sample(1:6, 10, replace = TRUE) #「サイコロを 10 回振って出た目を記録する実験」の再現
 [1] 6 5 4 6 4 5 4 4 2 1
> sample(1:6, 10, prob = 6:1, replace = TRUE) # 出る目の確率に偏りがある場合
 [1] 6 5 3 4 1 2 4 1 2 1
> ## 関数 rbinom の使い方
> rbinom(10, size = 4, prob = 0.5) # 確率 0.5 に対する次数 4 の二項乱数を 10 個発生
```

[1] R では擬似乱数を発生させるための方法として Mersenne ツイスターがデフォルトでは用いられている. `help(Random)` 参照.

[2] (a, b) は a より大きく b より小さい実数全体からなる集合を表す.

```
 [1] 1 2 2 2 1 1 1 2 1 3
> rbinom(20, size = 4, prob = 0.2) # 個数を 20, 確率を 0.2 に変更
 [1] 0 1 1 0 1 0 0 1 2 0 1 0 0 0 1 1 1 1 1 1
> ## 関数 runif の使い方
> runif(5, min = -1, max = 2) # 区間 (-1,1) 上の一様乱数を 5 個発生
[1]  1.2634255  0.8876634  1.1305472 -0.9981257  0.4259497
> runif(5) # 何も指定しないと区間 (0,1) を指定したことになる
[1] 0.2201189 0.3798165 0.6127710 0.3517979 0.1111354
> ## 関数 set.seed について
> set.seed(1) # 乱数の初期値を seed=1 で指定
> runif(5)
[1] 0.2655087 0.3721239 0.5728534 0.9082078 0.2016819
> set.seed(2) # 乱数の初期値を seed=2 で指定
> runif(5) # seed=1 の場合と異なる結果となる
[1] 0.1848823 0.7023740 0.5733263 0.1680519 0.9438393
> set.seed(1) # 乱数の初期値を seed=1 で指定
> runif(5) # 初めの seed=1 の結果と同じ
[1] 0.2655087 0.3721239 0.5728534 0.9082078 0.2016819
```

(sample.r)

R には他にも様々な種類の確率分布に従う乱数が実装されているが, それらについては以下で詳しく説明する.

4.2. 確率変数と確率分布

数学的には, 乱数は**確率変数** *(random variable)* という概念でモデル化される. 確率変数とは, 値がランダムに決定される変数で, すべての実数 $a \leq b$ に対して, その値が区間 $[a, b]$ に含まれる確率があらかじめ定められているような変数のことをいう.[3]

X を確率変数とすると, 定義より X が区間 $[a, b]$ $(a \leq b)$ に含まれる確率が定まるから, その確率を $P(a \leq X \leq b)$ で表す. 特に $a = b$ のとき, $P(a \leq X \leq b)$ は $X = a$ となる確率を表すから, それを $P(X = a)$ で表すことにする.

乱数はランダムに生成された数列であったが, この場合「ランダム」という言葉は以下の 2 種類の意味に使われている:

(i) 数列の個々の数字がランダムに決定されている.
(ii) 数列の値の並び方に規則性がない (個々の数字がとる値が他の数字がとる値に影響しない).

(i) のランダム性は確率変数によってモデル化できる. (ii) のランダム性は, 数学的には**独立性** *(independence)* という概念でモデル化される: 確率変数の列 $X_1, X_2, \ldots, X_n$ が**独立** *(independent)* であるとは, $a_i \leq b_i$ $(i = 1, \ldots, n)$ なる任意の実数 $a_1, b_1, \ldots, a_n, b_n$ に対して,

$$\begin{aligned}(4.1)\quad & P(a_1 \leq X_1 \leq b_1, a_2 \leq X_2 \leq b_2, \ldots, a_n \leq X_n \leq b_n) \\ & \qquad = P(a_1 \leq X_1 \leq b_1)P(a_2 \leq X_2 \leq b_2) \cdots P(a_n \leq X_n \leq b_n)\end{aligned}$$

が成り立つことをいう. ここに, (4.1) の左辺は「X_1 が区間 $[a_1, b_1]$ に値をとり, X_2 が区間 $[a_2, b_2]$ に値をとり, ..., X_n が区間 $[a_n, b_n]$ に値をとる」という事象が起きる確率を表す. 従って, 以下で乱数というときは, 数学的には独立な確率変数列を指すものとする.

一口に「値がランダムに決定される」といっても, 出現しやすい数値や, まったく出現しない数値があるかもしれない. 確率統計学ではこのような値の出現頻度 (確率) を決定する法則が確率変数の背後に存在すると考えて, その法則を**確率分布** *(probability distribution)* または単に**分布**と呼び, 確率分布の数学的モデリングを通じて現象の理

[3] この定義は数学的には厳密性を欠くが, 本講義ではこの定義を採用する.

解を試みる. 確率分布の定義をもう少し正確に述べると, 確率変数 X に対して, 各区間 $[a,b]$ $(a \leq b)$ と, X が区間 $[a,b]$ に含まれる確率 $P(a \leq X \leq b)$ との対応を示したものを, X の確率分布または単に分布という.[4] また, このとき X はこの分布に**従う**という.

4.3. 離散分布

取りうる値が有限個, もしくは可算無限個 (例えば整数値のみとる場合) であるような確率変数は**離散型** *(discrete)* であるといい, 対応する確率分布を**離散分布**と呼ぶ. 離散分布は, その分布に従う確率変数 X が取りうる値 x のそれぞれに対して, $X = x$ となる確率 $P(X = x)$ を対応させる関数 $f(x) = P(X = x)$ を考えることで完全に決定される. この関数 f を**確率質量関数** *(probability mass function)*, あるいは単に**確率関数** *(probability function)* と呼ぶ.

4.3.1. 二項分布. n を正の整数, p を 0 以上 1 以下の実数とする. 取りうる値が $0, 1, \ldots, n$ であり, 確率関数が

$$f(x) = \binom{n}{x} p^x (1-p)^{n-x}, \qquad x = 0, 1, \ldots, n$$

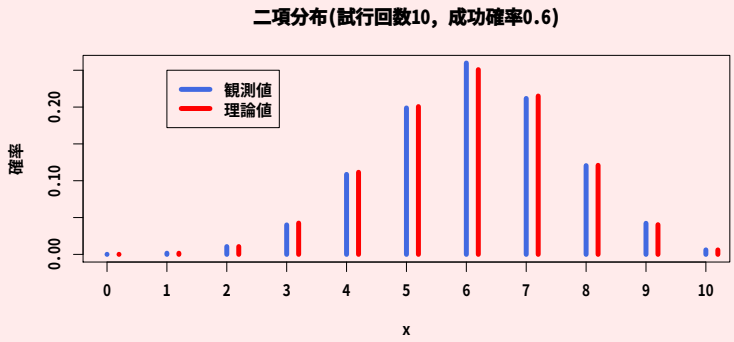
で与えられる離散分布を, 試行回数 n, 成功確率 p の**二項分布**と呼ぶ. 特に, 試行回数 1 の二項分布を *Bernoulli* **分布** *(Bernoulli distribution)* と呼ぶ.

例えば, 表が出る確率が p のコインを n 回投げたときに表が出る回数は試行回数 n, 成功確率 p の二項分布に従う.

二項分布に従う乱数の発生には関数 `rbinom()` を用いる. なお, 原則として, ある確率分布に従う乱数を生成するための R の関数の命名規則は,「`r` + その乱数が従う分布の名前の省略形」となっている (一部例外がある). また, 離散分布の場合, その確率関数を計算するための関数が, 同じ省略形の文頭に `d` をつけることで得られる. 例えば, 二項分布の確率関数は関数 `dbinom()` で計算できる.

```
> set.seed(123) # 乱数の初期値を指定
> rbinom(10, size = 1, prob = 0.5) # Bernoulli 分布のシミュレーション
 [1] 0 1 0 1 1 0 1 1 1 0
> rbinom(10, size = 1, prob = 0.2) # 成功確率を小さくしてみる
 [1] 1 0 0 0 0 1 0 0 0 1
> rbinom(20, size = 5, prob = 0.6) # 20 個の二項分布のシミュレーション
 [1] 2 2 3 0 3 2 3 3 4 4 1 2 2 2 5 3 2 4 4 4
> ## 統計的性質の確認
> m <- 10
> p <- 0.6
> x <- rbinom(10000, size = m, prob = p)
> mean(x) # 10 * 0.6 = 6 に近い (大数の法則)
[1] 6.0167
> (A <- table(x)/10000) # 出現確率ごとの表 (度数分布表) を作成
x
     0      1      2      3      4      5      6      7      8      9     10
0.0001 0.0016 0.0106 0.0400 0.1086 0.1988 0.2598 0.2117 0.1205 0.0422 0.0061
> plot(A, type ="h", lwd = 5, col = "royalblue", ylab = "確率",
+      main = paste0("二項分布 (試行回数", m, ", 成功確率", p, ")"))
> lines(0:10 + 0.2, dbinom(0:10, size = m, prob = p),
+       type = "h", col = "red", lwd = 5) # 理論上の出現確率
> legend(1, 0.25, legend = c("観測値", "理論値"),
+        col = c("royalblue", "red"), lwd = 5) # 凡例を作成
```

[4] より現代的な定義を述べるためには測度論の知識が必要となるため, ここでは簡易的な定義を述べた.



(rbinom2.r)

4.3.2. Poisson 分布. λ を正の実数とする. 取りうる値が 0 以上の整数であり, 確率関数が

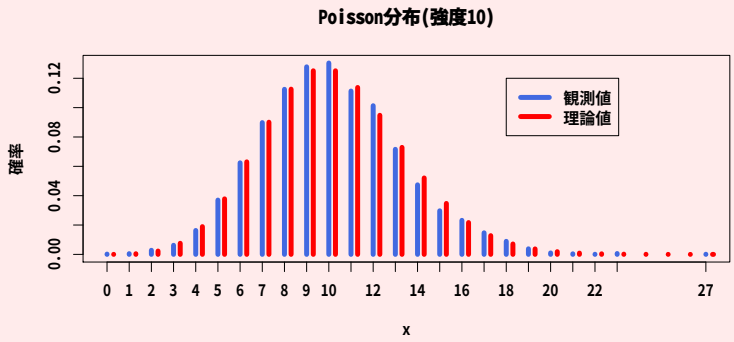
$$f(x) = \frac{\lambda^x}{x!} e^{-\lambda}, \qquad x = 0, 1, \ldots$$

で与えられる離散分布をパラメーター λ の *Poisson* **分布** *(Poisson distribution)* と呼び, 記号 $P_o(\lambda)$ で表す. λ は**強度** *(intensity)* と呼ばれることがある.

放射性物質から一定時間に放射される粒子の数や, 一定期間に起こる交通事故の数などは Poisson 分布に従うことが知られている. また, 前章で観察したように, 発生確率が低い事象が十分長い期間のあいだに起こる回数の分布は Poisson 分布で近似できる.

Poisson 分布に従う乱数の発生には関数 `rpois()` を用いる.

```
> set.seed(12345) # 乱数の初期値を指定
> rpois(10, lambda = 1) # 強度 1 の Poisson 分布に従う乱数を 10 個発生
 [1] 1 2 2 2 1 0 0 1 1 4
> rpois(20, lambda = 10) # 強度 10 の Poisson 分布に従う乱数を 20 個発生
 [1]  4 11  7  8 11  9 10  9 11 13 10 12 14  7  4 15  8 11 11  9
> ## 統計的性質の確認
> lambda <- 10
> x <- rpois(10000, lambda = lambda)
> mean(x) # lambda=10 に近い (大数の法則)
[1] 10.0125
> (A <- table(x)/10000) # 出現確率ごとの表 (度数分布表) を作成
x
     0      1      2      3      4      5      6      7      8      9     10
0.0002 0.0005 0.0028 0.0062 0.0163 0.0370 0.0624 0.0898 0.1125 0.1278 0.1304
    11     12     13     14     15     16     17     18     19     20     21
0.1113 0.1013 0.0716 0.0474 0.0297 0.0232 0.0147 0.0089 0.0038 0.0010 0.0004
    22     23     27
0.0001 0.0006 0.0001
> plot(A, type ="h", lwd = 5, col = "royalblue", ylab = "確率",
+      main = paste0("Poisson 分布 (強度", lambda, ")"))
> lines(min(x):max(x) + 0.3, dpois(min(x):max(x), lambda = lambda),
+       type = "h", col = "red", lwd = 5) # 理論上の出現確率
> legend(18, 0.12, legend = c("観測値", "理論値"),
+        col = c("royalblue", "red"), lwd = 5) # 凡例を作成
```



(rpois2.r)

4.4. 連続分布

実際のデータでは, 取りうる値が任意の実数またはある範囲の実数である場合, もしくは取りうる値のパターンが数多いため近似的にすべての実数値またはある範囲の実数値をとりうると考えられる場合が頻繁にある. 具体例としては, 株価, 気温, 風速, 液体の体積などがある. このようなデータのモデル化には, しばしば連続分布に従う確率変数が用いられる. さらに, 以下で見るように, 離散分布に従うデータであっても, サンプル数が非常に大きい状況ではその分布はしばしば連続分布で近似できる. このように, 離散的なデータの解析であったとしても, 連続分布を考えることは理論上重要となる.

一般に, 確率変数 X が**連続型** *(continuous)* であるとは, 非負の値をとる実数直線上の関数 f があって, $a \leq b$ なるすべての実数 a, b に対して

$$P(a \leq X \leq b) = \int_a^b f(x)dx$$

が成り立つことをいい, 対応する確率分布を**連続分布**と呼ぶ. また, 関数 f をこの確率分布の**確率密度関数** *(probability density function)*, あるいは単に**密度** *(density)* と呼ぶ.

4.4.1. 正規分布.

μ を実数, σ を正の実数とするとき, 確率密度関数が

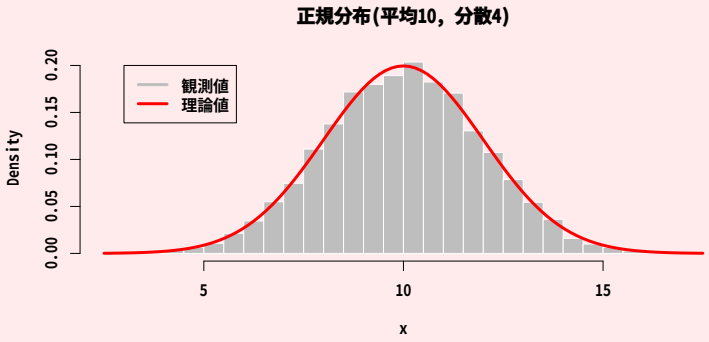
$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

で与えられる分布を平均 μ, 分散 σ^2 の**正規分布** *(normal distribution)* または **Gauss 分布**と呼び, 記号 $N(\mu, \sigma^2)$ で表す. 特に, 平均 0, 分散 1 の正規分布を**標準正規分布** *(standard normal distribution)* と呼ぶ. なお, σ のことを標準偏差と呼ぶ. 物理実験等の観測誤差の分布はしばしば正規分布でモデル化される.

ここで,「平均」,「分散」,「標準偏差」という言葉は, データから計算される平均, 分散, 標準偏差とは意味合いが異なることに注意する必要がある. 両者を区別するために, 後者の文頭に「標本」という言葉をつける場合がある. 適当な仮定のもとで, データ数が大きくなるにつれて, 後者の意味での平均, 分散, 標準偏差はそれぞれ前者の意味での値に近づいていくことが知られている (大数の法則).

正規分布に従う乱数の発生には関数 `rnorm()` を用いる. なお, 連続分布の場合, 分布の省略形の文頭に `d` をつけることで, 確率密度関数を計算するための関数が得られる. 例えば, 正規分布の確率密度関数は関数 `dnorm()` で計算できる.

```
> set.seed(20) # 乱数の初期値を指定
> rnorm(10) # 標準正規乱数を 10 個発生
 [1]  1.1626853 -0.5859245  1.7854650 -1.3325937 -0.4465668  0.5696061
 [7] -2.8897176 -0.8690183 -0.4617027 -0.5555409
> ## 統計的性質の確認
> mu <- 10
> sigma <- 2
> x <- rnorm(10000, mean = mu, sd = sigma)
> mean(x) # mu=10 に近い (大数の法則)
[1] 9.986371
> hist(x, freq = FALSE, breaks = 25, col = "gray", border = "white",
+      main = paste0("正規分布 (平均", mu, ", 分散", sigma^2, ")")) # ヒストグラム (密度表示)
> curve(dnorm(x, mean = mu, sd = sigma), add = TRUE,
+       col = "red", lwd = 3) # 理論上の確率密度関数
> legend(3, 0.2, legend = c("観測値", "理論値"),
+        col = c("gray", "red"), lwd = 3) # 凡例を作成
```

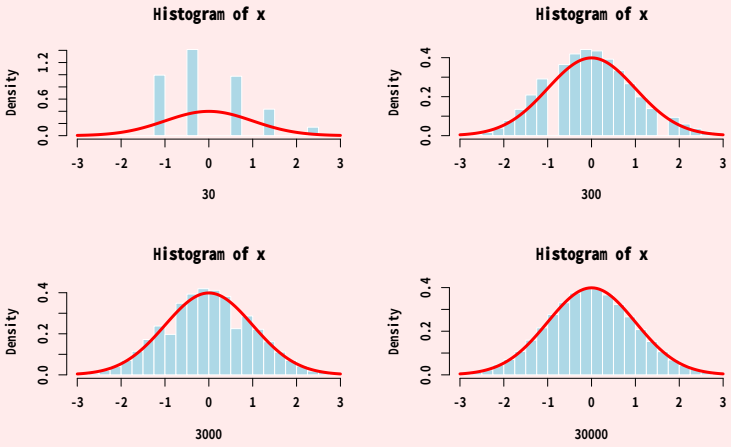


(rnorm2.r)

正規分布は離散分布の極限としても現れる．Y を試行回数 n, 成功確率 p の二項分布に従う確率変数とすると, n が十分大きいとき, $(Y-np)/\sqrt{np(1-p)}$ の分布は標準正規分布で近似できる.[5]

```
> # 二項分布の極限：離散分布から連続分布へ
> set.seed(123)
> op <- par(mfrow=c(2,2))
> p <- 1/(7*pi)
> for(i in 1:4){
+   n <- 3*10^i
+   x <- (rbinom(1000000,n,prob=p)-n*p)/sqrt(n*p*(1-p))
+   hist(x,breaks = c(-Inf,seq(-3,3,0.25),Inf),freq = FALSE,
+        xlim=c(-3,3),xlab=n,col="lightblue",border = "white")
+   curve(dnorm(x, mean = 0, sd = 1), add = TRUE,
+         col = "red", lwd = 3) # 理論上の確率密度関数
+ }
> par(op)
```

[5] **de Moivre-Laplace の定理**として知られている．**中心極限定理** *(central limit theorem)* はその一般化である．



(rbinom-normal2.r)

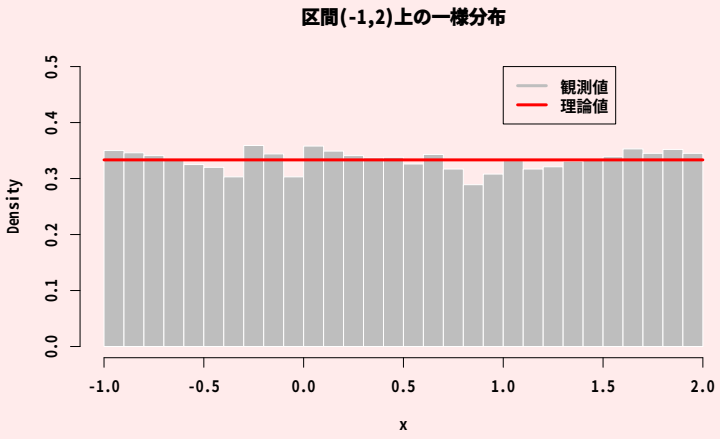
4.4.2. 一様分布. $a < b$ とする. 確率密度関数が

$$f(x) = \begin{cases} \frac{1}{b-a} & a < x < b\text{のとき}, \\ 0 & \text{上記以外のとき} \end{cases}$$

で与えられる連続分布を区間 (a, b) 上の**一様分布** *(uniform distribution)* と呼び, 記号 $U(a, b)$ で表す.

一様分布に従う乱数の発生には関数 `runif()` を用いる.

```
> set.seed(1) # 乱数の初期値を指定
> runif(10) # 区間 (0,1) 上の一様乱数を 10 個発生
 [1] 0.26550866 0.37212390 0.57285336 0.90820779 0.20168193 0.89838968
 [7] 0.94467527 0.66079779 0.62911404 0.06178627
> ## 統計的性質の確認
> a <- -1
> b <- 2
> x <- runif(10000, min = a, max = b)
> mean(x) # (a+b)/2=0.5 に近い (大数の法則)
[1] 0.5001657
> hist(x, freq = FALSE, breaks = 25, col = "gray",
+      border = "white", ylim = c(0, 0.5),
+      main = paste0("区間 (", a, ",", b, ") 上の一様分布")) # ヒストグラム (密度表示)
> curve(dunif(x, min = a, max = b), add = TRUE,
+       col = "red", lwd = 3) # 理論上の確率密度関数
> legend(1, 0.5, legend = c("観測値", "理論値"),
+        col = c("gray", "red"), lwd = 3) # 凡例を作成
```



(runif2.r)

4.4.3. ガンマ分布. ν, α を正の実数とする. 確率密度関数が

$$f(x) = \frac{1}{\Gamma(\nu)}\alpha^{\nu}x^{\nu-1}e^{-\alpha x} \quad (x > 0), \qquad f(x) = 0 \quad (x \leq 0)$$

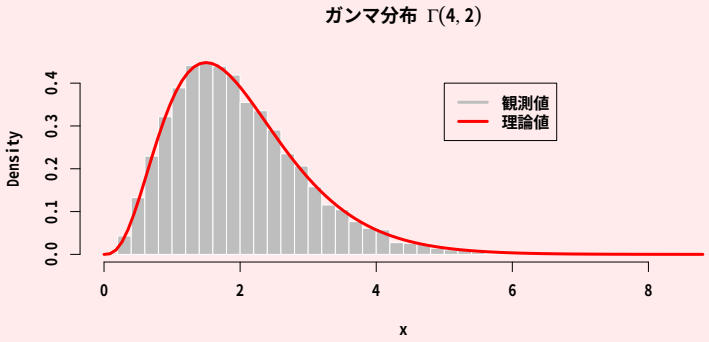
で与えられる連続分布をパラメータ ν, α の**ガンマ分布** *(gamma distribution)* と呼び, 記号 $\Gamma(\nu, \alpha)$ や $G(\alpha, \nu)$ で表す. ただし, $\Gamma(\nu)$ は**ガンマ関数** *(gamma function)*

$$\Gamma(\nu) = \int_0^{\infty} x^{\nu-1}e^{-x}dx$$

を表す. ν, α はそれぞれ**形状パラメーター** *(shape)*, **レート** *(rate)* と呼ばれることがある. 体重の分布はガンマ分布に従うといわれている.

ガンマ分布に従う乱数の発生には関数 `rgamma()` を用いる.

```
> set.seed(123) # 乱数の初期値を指定
> # ガンマ分布に従う乱数
> rgamma(10,shape = 3, rate = 1) # ガンマ分布に従う乱数を 10 個発生
 [1] 1.6923434 4.7360299 0.5422275 2.7086007 5.9471178 3.2818834 0.8998575
 [8] 0.5148113 4.8100373 3.1012821
> ## 統計的性質
> nu <- 4
> alpha <- 2
> x <- rgamma(10000, shape = nu, rate = alpha)  # ガンマ乱数を 10000 個発生
> mean(x) # nu/alpha=2 に近い (大数の法則)
[1] 1.980431
> hist(x, freq = FALSE, breaks = 50, col = "gray", border = "white",
+      main = bquote(paste("ガンマ分布 ", Gamma(.(nu), .(alpha))))) # ヒストグラム (密度表示)
> curve(dgamma(x, shape = nu, rate = alpha), add = TRUE,
+       col = "red", lwd = 3) # 理論上の確率密度関数
> legend(5, 0.4, legend = c("観測値", "理論値"),
+        col = c("gray", "red"), lwd = 3) # 凡例を作成
```



(rgamma2.r)

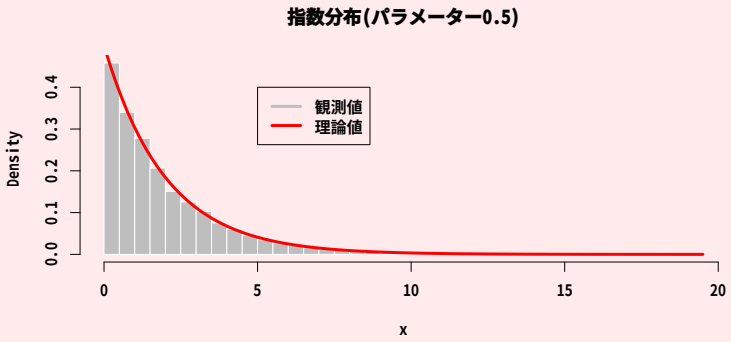
上の実行例におけるタイトルの作成では, 文字列・数式・R オブジェクトを組み合わせた文字列を作成するために関数 bquote() を利用している. 表現 .() は数式と R オブジェクトを区別するために使われている.

ガンマ分布はいくつかの応用上重要な確率分布を特殊な場合として含む. 正の実数 λ に対して, $\Gamma(1,\lambda)$ をパラメータ λ の**指数分布** *(exponential distribution)* と呼び, 記号 $\mathrm{Exp}(\lambda)$ で表す. λ は**レート**と呼ばれることがある. また, 正の実数 k に対して, $\Gamma(k/2,1/2)$ を自由度 k の χ^2 **分布**と呼び, 記号 $\chi^2(k)$ で表す.[6]

χ^2 分布および指数分布はガンマ分布の特殊な場合であるから関数 rgamma() によって乱数を発生させられるが, 便宜のためそれぞれ専用の乱数発生関数 rchisq() および rexp() が用意されている.

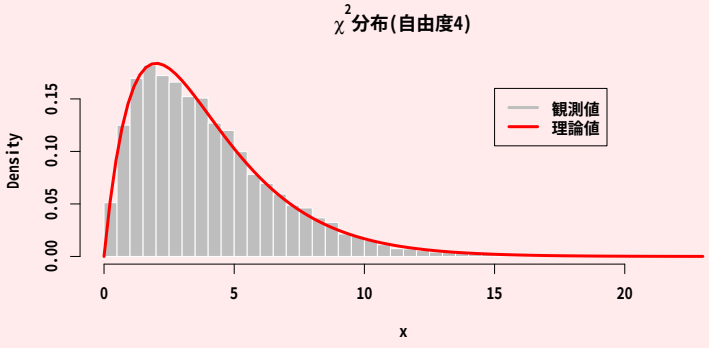
```
> set.seed(20) # 乱数の初期値を指定
> rexp(10) # レート 1 の指数分布に従う乱数を 10 個発生
 [1] 0.19336251 0.05832739 0.06330693 2.21143320 1.00352299 1.17344535
 [7] 0.43105511 0.51559271 6.37169900 0.98173630
> ## 統計的性質の確認
> lambda <- 0.5
> x <- rexp(10000, rate = lambda) # レート 0.5 の指数乱数を 10000 個発生
> mean(x) # 1/lambda = 2 に近い (大数の法則)
[1] 1.962623
> hist(x, freq = FALSE, breaks = 50, col = "gray", border = "white",
+      main = paste0("指数分布 (パラメーター", lambda, ")")) # ヒストグラム (密度表示)
> curve(dexp(x, lambda), add = TRUE, col = "red", lwd = 3) # 理論上の確率密度関数
> legend(5, 0.4, legend = c("観測値", "理論値"),
+        col = c("gray", "red"), lwd = 3) # 凡例を作成
```

[6] χ^2 は「カイ二乗」と読む.



(rexp.r)

```
> set.seed(20) # 乱数の初期値を指定
> rchisq(10, df = 1) # 自由度 1 のカイ二乗分布に従う乱数を 10 個発生
 [1] 2.47564812 0.38394375 1.60988258 0.29093644 0.67851954 0.01357661
 [7] 1.27772421 0.56221273 0.63248955 0.18637919
> ## 統計的性質の確認
> k <- 4 # 自由度
> x <- rchisq(10000, df = k) # 自由度 4 のカイ二乗乱数を 10000 個発生
> mean(x) # k = 4 に近い (大数の法則)
[1] 4.01317
> hist(x, freq = FALSE, breaks = 50, col = "gray", border = "white",
+      main = bquote(paste(chi^2, "分布 (自由度", .(k), ")"))) # ヒストグラム (密度表示)
> curve(dchisq(x, k), add = TRUE, col = "red", lwd = 3) # 理論上の確率密度関数
> legend(15, 0.16, legend = c("観測値", "理論値"),
+        col = c("gray", "red"), lwd = 3) # 凡例を作成
```

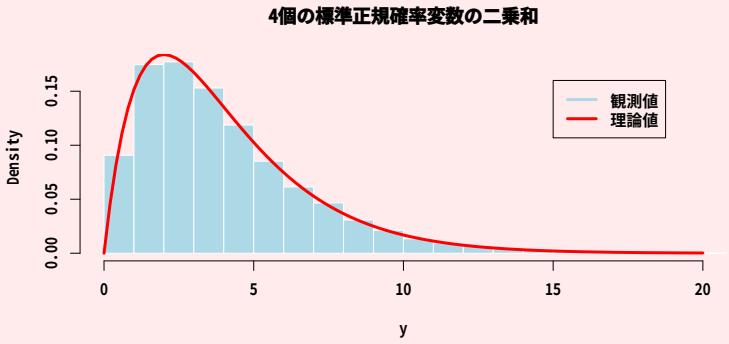


(rchisq.r)

標準正規分布に従う k 個の独立な確率変数の二乗和は自由度 k の χ^2 分布に従うことが知られている．この事実は推定や検定の理論において重要な役割を果たす．

```
> ## 標準正規確率変数の二乗和
> set.seed(123) # 乱数の初期値を指定
> n <- 30000
> k <- 4
> y <- colSums(matrix(rnorm(n*k, 0,1)^2,k,n))
> # 標準正規分布に従う乱数を nk 個発生し，k 個の 2 乗和を n 個作る．
```

```
> hist(y, freq = FALSE, breaks = 40, col = "lightblue", xlim = c(0,20),
+      border = "white",
+      main = paste0(k, "個の標準正規確率変数の二乗和")) # ヒストグラム (密度表示)
> curve(dchisq(x, df = k), add = TRUE, xlim=c(0,20),
+       col = "red", lwd = 3) # 理論上の確率密度関数
> legend(15, 0.16, legend = c("観測値", "理論値"),
+        col = c("lightblue", "red"), lwd = 3) # 凡例を作成
```



(rgamma-chi2.r)

4.4.4. t **分布.** Y, Z を 2 つの確率変数とし, Y が自由度 n のカイ 2 乗分布に従い, Z が標準正規分布に従い, かつ Y, Z が独立であるとする. このとき, 確率変数

$$\frac{Z}{\sqrt{Y/n}}$$

の分布を自由度 n の *(Student* **の**$)t$ **分布**と呼び, 記号 $t(n)$ で表す.[7] t 分布は連続型であることが知られており, その確率密度関数は

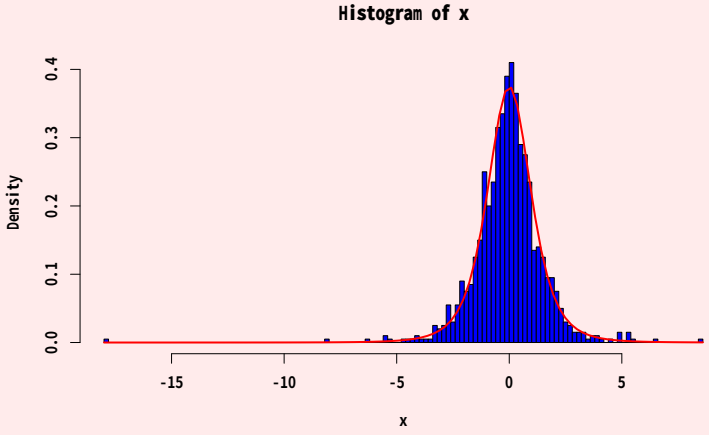
$$f(x) = \frac{1}{\sqrt{\pi n}} \frac{\Gamma((n+1)/2)}{\Gamma(n/2)} \left(1 + \frac{x^2}{n}\right)^{-(n+1)/2}$$

で与えられる.

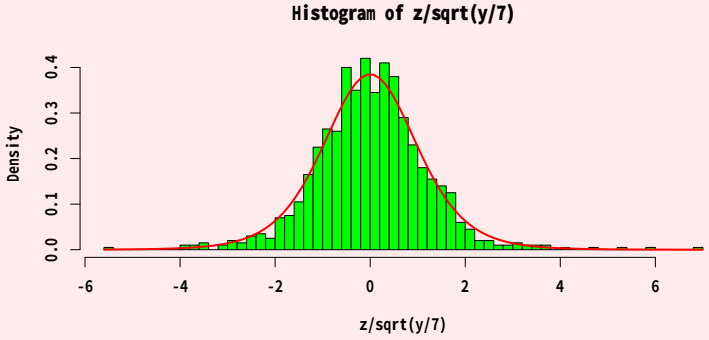
t 分布に従う乱数の発生には関数 `rt()` を用いる.

```
> set.seed(123) # 乱数の初期値を指定
> rt(10, df = 4) # 自由度 4 の t 分布に従う乱数を 10 個発生
 [1] -0.7143044 -1.2954198 -0.1321482 -2.0299890  1.6187180  2.3848622
 [7]  0.4810578  0.5147932 -0.6945907 -2.3176275
> x <- rt(1000, df = 4)
> hist(x, freq = FALSE, breaks = 100, col = "blue") # ヒストグラム (密度表示)
> curve(dt(x, df = 4), add = TRUE,
+       col = "red", lwd = 2) # 理論上の確率密度関数
> ## 0 から大きく離れた値が現れている (裾が重い)
```

[7] Student は t 分布を導入した統計学者 Gosset のペンネームである.




```
> ### 正規分布とカイ 2 乗分布を利用して$t$分布を生成
> y <- rchisq(1000, df = 7) # 自由度 7 のカイ 2 乗分布に従う乱数
> z <- rnorm(1000) # 標準正規乱数
> hist(z/sqrt(y/7), freq = FALSE, breaks = 50, col = "green") # ヒストグラム (密度表示)
> curve(dt(x, df = 7), add = TRUE,
+       col = "red", lwd = 2) # 理論上の確率密度関数
```



(rt.r)

4.4.5. F **分布.** Y_1, Y_2 を 2 つの確率変数とし, Y_1 が自由度 m のカイ 2 乗分布に従い, Y_2 が自由度 n のカイ 2 乗分布に従い, かつ Y_1, Y_2 が独立であるとする. このとき, 確率変数

$$\frac{Y_1/m}{Y_2/n}$$

の分布を自由度 m, n の F **分布**と呼び, 記号 $F(m,n)$ で表す. F 分布は連続型であることが知られており, その確率密度関数は

$$f(x) = \frac{(m/n)^{m/2}}{B(m/2, n/2)} \frac{x^{m/2-1}}{(1+mx/n)^{(m+n)/2}}$$

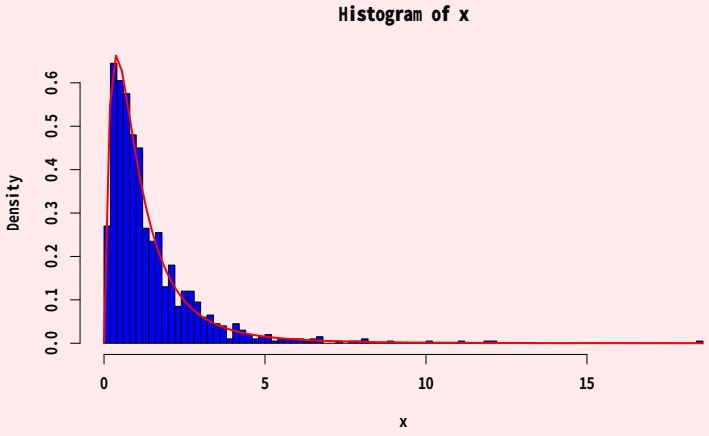
で与えられる. ただし, $p, q > 0$ に対して, $B(p,q)$ は**ベータ関数** *(beta function)*

$$B(p,q) = \int_0^1 x^{p-1}(1-x)^{q-1} dx$$

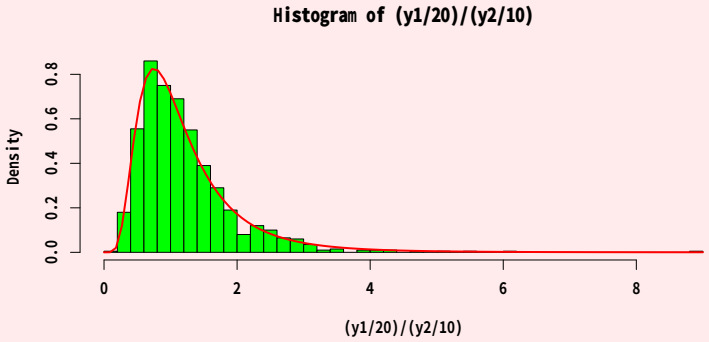
を表す.

F 分布に従う乱数の発生には関数 `rf()` を用いる.

```
> set.seed(123) # 乱数の初期値を指定
> rf(10, df1 = 4, df2 = 7) # 自由度 4,7 の F 分布に従う乱数を 10 個発生
 [1] 0.28826915 0.07826748 1.97011660 0.81384301 1.61509656 1.11251962
 [7] 0.27709589 1.61571001 0.64807256 0.31063495
> x <- rf(1000, df1 = 4, df2 = 7)
> hist(x, freq = FALSE, breaks = 100, col = "blue") # ヒストグラム (密度表示)
> curve(df(x, df1 = 4, df2 = 7), add = TRUE,
+       col = "red", lwd = 2) # 理論上の確率密度関数
```




```
> ### カイ 2 乗分布を利用して$F$分布を生成
> y1 <- rchisq(1000, df = 20) # 自由度 20 のカイ 2 乗分布に従う乱数
> y2 <- rchisq(1000, df = 10) # 自由度 10 のカイ 2 乗分布に従う乱数
> hist((y1/20)/(y2/10), freq = FALSE, breaks = 50, col = "green") # ヒストグラム (密度表示)
> curve(df(x, df1 = 20, df2 = 10), add = TRUE,
+       col = "red", lwd = 2) # 理論上の確率密度関数
```



(rf.r)

4.5. 多次元確率変数と多変量分布

本講義では多変量データを扱うので, 確率変数の多次元版を考える必要がある. 値がランダムに決定される d 次元ベクトルで, 各座標が確率変数であるようなものを d **次元確率変数** *(d-dimensional random variable)* または d **次元確率ベクトル** *(d-dimensional random vector)* と呼ぶ.[8]

1 次元の場合の多次元化として, 多次元確率変数の分布を以下のようにして定義する. d 次元確率変数 $X = (X_1, X_2, \ldots, X_d)^\top$ に対して, d 次元長方形 $\{(x_1, \ldots, x_d) :$

[8] 以下特に断らない限り, ベクトルは列ベクトルとみなす.

$a_i \le x_i \le b_i\ (i = 1, \ldots, d)\}$ $(a_i \le b_i,\ i = 1, \ldots, d)$ と, X がこの d 次元長方形に含まれる確率

$$P(a_1 \le X_1 \le b_1, a_2 \le X_2 \le b_2, \ldots, a_d \le X_d \le b_d)$$

との対応を示したものを, X の (d **変量**) **確率分布**または単に (d **変量**) **分布**といい, X はこの分布に**従う**という.

4.5.1. 多項分布. k を正整数とする. 1 回の試行で起こり得る k 個の排反な事象 $E_1, ..., E_k$ があり，どれかは必ず起こるとする．E_i の起こる確率が p_i であるとする．この試行を独立に n 回繰り返し，E_i が起こった回数を X_i とする．このようにして定義される k 次元確率変数 $X = (X_1, \ldots, X_k)^\top$ の分布を試行回数 n, 確率 $p_1, \ldots, p_k$ の k **項分布**と呼ぶ. 総称して**多項分布** *(multinomial distribution)* と呼ぶ. なお, 各 $i = 1, \ldots, k$ に対して, 確率変数 X_i の分布は試行回数 n, 成功確率 p_i の二項分布であることが確認できる. この意味で, 多項分布は二項分布の多変量版であると考えられる.

いまの場合, k 次元確率変数 X は有限個の値しかとりえないため, 1 次元の場合と同様その分布は X のとりうる値 x のそれぞれに対して X が値 x をとる確率を対応させる関数 $f(x)$ を与えることで完全に決定される. この関数 f を X(の分布) の**確率 (質量) 関数**と呼ぶ. 試行回数 n, 確率 $p_1, \ldots, p_k$ の k 項分布の確率関数は, $0 \le x_i \le n$ $(i = 1, \ldots, k)$, $\sum_{i=1}^{k} x_i = n$ なる整数の組 $(x_1, \ldots, x_k)$ に対して定義され,

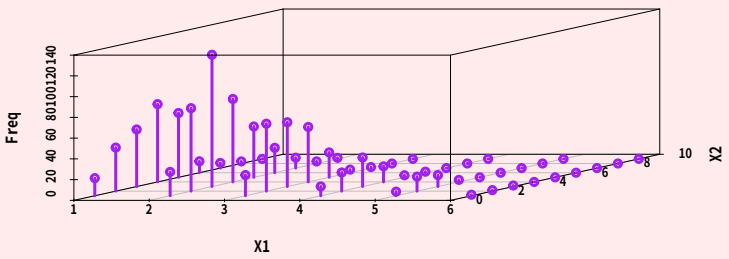
$$f(x_1, \ldots, x_k) = \frac{n!}{x_1! \cdots x_k!} p_1^{x_1} \cdots p_k^{x_k}$$

で与えられる.

1 次元の場合と同様にして, d 次元確率変数の列の独立性が定義される (区間を d 次元長方形に置き換えればよい). 1 次元の場合と同様に, 独立な多次元確率変数列を (多変量の) 乱数と呼ぶことにする.

1 変量分布の場合と異なり, 多変量分布に従う乱数の生成は容易でないことが多く, R にデフォルトで実装されている関数もほとんどない. 多項分布は数少ない例外であり, 関数 `rmultinom()` によって多項分布に従う乱数を生成できる.

```
> set.seed(123) # 乱数の初期値の設定
> rmultinom(10, size = 12, prob = c(0.1,0.2,0.7)) # (3,10) 行列
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]    1    1    3    1    1    3    2    0    0     1
[2,]    4    4    0    4    2    2    2    5    0     5
[3,]    7    7    9    7    9    7    8    7   12     6
> ## 可視化: scatterplot3d パッケージを利用
> install.packages("scatterplot3d")
> library(scatterplot3d)
> x <- rmultinom(1000, size = 12, prob = c(0.1,0.2,0.7))
> (A <- table(x[1, ], x[2, ])) # (X1,X2) の出現頻度の表 (クロス表) を作成
    0   1   2   3   4   5   6   7   8
0  17  42  55  75  62  11   5   2   0
1  23  80 127  80  49  24  10   2   1
2  20  65  62  53  24   3   1   0   0
3   9  18  28  15   2   1   0   0   0
4   4  14  11   2   0   0   0   0   0
5   1   1   1   0   0   0   0   0   0
> scatterplot3d(as.data.frame(A), type = "h", lwd = 3, xlab = "X1",
+ ylab = "X2", color = "purple")
```



(rmultinom.r)

4.5.2. 多変量正規分布. 1 変量の場合と同様に, 多変量の場合も連続分布が定義される. d 次元確率変数 X(の分布) が**連続型**であるとは, ある d 個の変数をもつ非負値関数 $f(x_1,\dots,x_d)$ が存在して, $a_i \le b_i$ $(i=1,2,\dots,d)$ なる任意の実数 $a_1,b_1,a_2,b_2,\dots,a_d,b_d$ に対して,

$$P(a_1 \le X_1 \le b_1, a_2 \le X_2 \le b_2, \dots, a_d \le X_d \le b_d) = \int_{a_1}^{b_1}\int_{a_2}^{b_2}\cdots\int_{a_d}^{b_d} f(x_1,x_2,\dots,x_d)dx_1dx_2\cdots dx_d$$

が成り立つことをいう. 1 変量の場合と同様 X の分布は関数 f によって完全に決定されることが知られており, この関数 f を X(の分布) の **(確率) 密度 (関数)** と呼ぶ.

多変量連続分布の最も重要な例は多変量正規分布である: $\boldsymbol{\mu}$ を d 次元ベクトル, Σ を d 次正定値対称行列[9] とするとき, 確率密度関数が

$$f(\boldsymbol{x}) = \frac{1}{(2\pi)^{d/2}\sqrt{\det\Sigma}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

で与えられる連続型 d 変量分布を, 平均ベクトル $\boldsymbol{\mu}$, 共分散行列 Σ の d **変量正規分布** *(d-dimensional normal distribution)* と呼ぶ.[10] ここで, d 次元ベクトル $\boldsymbol{x}$ は列ベクトルとみなしている. 特に, 平均ベクトルが零ベクトルで共分散行列が単位行列の d 変量正規分布を d **変量標準正規分布** *(d-dimensional standard normal distribution)* と呼ぶ.

R にはデフォルトでは多変量正規分布に従う乱数を生成するための関数は用意されていないため, 自作する必要がある (後述するように, パッケージをインストールする方法もある). 多変量正規分布のシミュレーションを行うためには, 次の命題が有用である:

命題 4.1. (a) $X_1,\dots,X_d$ を標準正規分布に従う独立な d 個の確率変数とする. このとき, d 次元確率変数 $X=(X_1,\dots,X_d)^\top$ は d 変量標準正規分布に従う.

(b) X を平均ベクトル $\boldsymbol{\mu}$, 共分散行列 Σ の d 変量正規分布に従う d 変量確率変数とする. このとき, A が d 次正則行列, $\boldsymbol{b}$ が d 次元列ベクトルならば, d 次元確率変数 $AX+\boldsymbol{b}$ は平均ベクトル $A\boldsymbol{\mu}+\boldsymbol{b}$, 共分散行列 $A\Sigma A^\top$ の d 変量正規分布に従う.

命題 4.1 より, d 次元ベクトル $\boldsymbol{\mu}$ および d 次元正定値対称行列 Σ が与えられたとき, 以下の手順によって平均ベクトル $\boldsymbol{\mu}$, 共分散行列 Σ の d 変量正規分布に従う d 次元確率変数 X を生成できる:

---

[9] d 次対称行列 Σ が**正定値** *(positive definite)* であるとは, Σ の固有値がすべて正であることをいう.

[10] Σ の行列式は Σ の固有値の積で与えられるから, $\det\Sigma>0$ であり, 特に Σ は正則である.

(1) d 個の標準正規乱数 $Z_1, \ldots, Z_d$ を生成し, $Z = (Z_1, \ldots, Z_d)^\top$ とおく. 命題 4.1(a) より Z は d 変量標準正規分布に従う.
(2) d 次正方行列 A で $\Sigma = AA^\top$ を満たすものを計算し, $X = \boldsymbol{\mu} + AZ$ とおく. 命題 4.1(b) より X は平均ベクトル $\boldsymbol{\mu}$, 共分散行列 Σ の d 変量正規分布に従う.

上の手順のうち, (1) における標準正規乱数の生成は関数 `rnorm()` によって実行できる. 手順 (2) における行列 A の計算にはいくつか方法があるが, ここでは固有値分解を用いる方法を説明する.[11] Σ は対称行列だから, ある d 次正則行列 V によって対角化できる: $V^{-1}\Sigma V = \Lambda$. ここに, Λ は Σ の固有値を対角成分とする対角行列である. さらに, V を直交行列, すなわち $V^{-1} = V^\top$ となるようにとることができることが知られている. いま, Σ は正定値であったから, Λ の対角成分 $\lambda_1, \ldots, \lambda_d$ はすべて正である. 従って, $\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_d}$ を対角成分とする対角行列 D を考えることができる. このとき $A := VDV^\top$ とおくと,

$$AA^\top = VDV^\top(VDV^\top)^\top = VDV^\top VDV^\top = VD^2V^\top = V\Lambda V^\top = \Sigma$$

となるので, この行列 A が求めるべきものである.

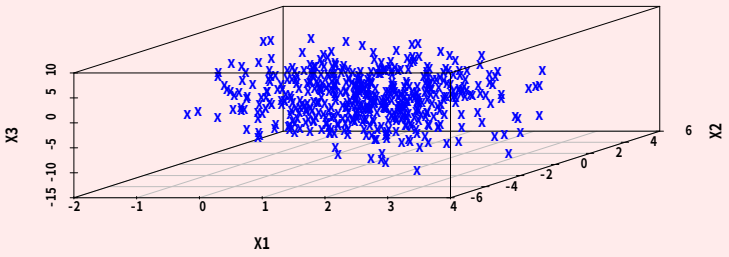
なお, 多変量正規分布に従う乱数を発生させるための関数を実装しているパッケージはいくつか存在する. パッケージ `MASS` には関数 `mvrnorm()` が, パッケージ `mvtnorm` には関数 `rmvnorm()` がそれぞれ多変量正規分布に従う乱数を発生させるための関数として実装されている.

```
> set.seed(123) # 乱数の初期値の設定
> ### 3 変量正規乱数のシミュレーション
> n <- 500 # 生成する乱数の個数
> mu <- c(1, 0, -1) # 平均ベクトル
> (Sigma <- matrix(c(1, -0.4, -2.1,
+                    -0.4, 4, 3.6,
+                    -2.1, 3.6, 9), nrow = 3)) # 共分散行列
     [,1] [,2] [,3]
[1,]  1.0 -0.4 -2.1
[2,] -0.4  4.0  3.6
[3,] -2.1  3.6  9.0
> ## 固有値分解による方法
> z <- rnorm(3 * n) # 標準正規乱数の生成
> z <- matrix(z, 3, n) # 計算しやすくするために行列に変換
> r <- eigen(Sigma)
> A <- r$vectors %*% diag(sqrt(r$values)) %*% solve(r$vectors)
> x <- mu + A %*% z
> x <- t(x) # 標準的なデータ形式に変換
> colMeans(x) # mu に近い
[1]  1.00393933  0.04316403 -0.93453934
> cov(x) # Sigma に近い
           [,1]       [,2]      [,3]
[1,]  0.9288979 -0.2662726 -1.978813
[2,] -0.2662726  3.9361528  3.346740
[3,] -1.9788132  3.3467398  8.857162
> cov2cor(Sigma) # 理論上の相関行列
     [,1] [,2] [,3]
[1,]  1.0 -0.2 -0.7
[2,] -0.2  1.0  0.6
[3,] -0.7  0.6  1.0
> cor(x) # 理論上のものに近い
```

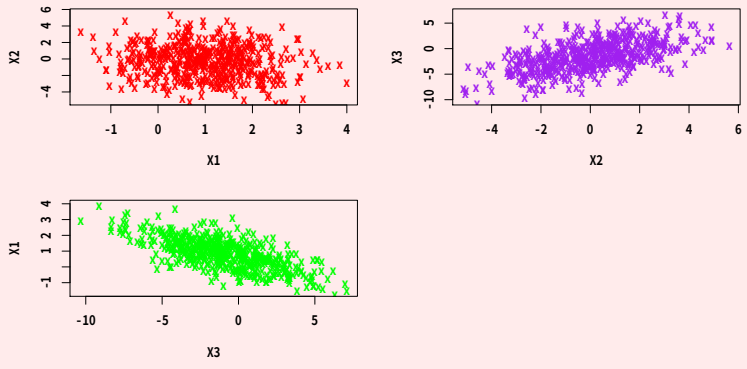
[11] 別の方法としては, 例えば 2.7.3 節で説明したコレスキー分解を使う方法があり, より直接的である.

```
           [,1]       [,2]       [,3]
[1,]  1.0000000 -0.1392536 -0.6898799
[2,] -0.1392536  1.0000000  0.5668115
[3,] -0.6898799  0.5668115  1.0000000
```

```
> scatterplot3d(x, pch = "x", color = "blue", xlab = "X1",
+               ylab = "X2", zlab = "X3") # 3次元散布図
```




```
> op <- par(mfrow = c(2,2),    # 描画領域を 2x2 に分割
+           mar = c(5,5,1,1)) # 余白を調整
> plot(x[ ,1:2],  pch = "x", xlab = "X1", ylab = "X2",
+      col = "red")
> plot(x[ ,2:3],  pch = "x", xlab = "X2", ylab = "X3",
+      col = "purple")
> plot(x[ ,c(3,1)],  pch = "x", xlab = "X3", ylab = "X1",
+      col = "green")
> par(op)
```



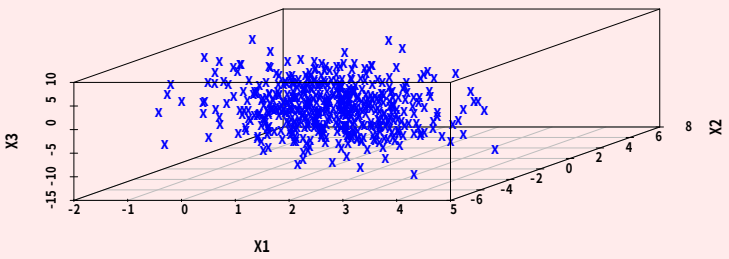

```
> ## パッケージ MASS の利用
> library(MASS)
> x <- mvrnorm(n, mu = mu, Sigma = Sigma)
> colMeans(x) # mu に近い
```

```
[1]  1.083652293 -0.005249756 -1.204166624
```

```
> cov(x) # Sigma に近い
```

```
           [,1]       [,2]      [,3]
[1,]  1.0795627 -0.5349028 -2.271364
[2,] -0.5349028  3.8673094  3.922672
[3,] -2.2713637  3.9226722  9.402201
```

```
> scatterplot3d(x, pch = "x", color = "blue", xlab = "X1", ylab = "X2", zlab = "X3")
```



(rmvnorm2.r)

4.6. その他

R にはここで紹介した以外にも数多くの確率分布を発生させる乱数が実装されている. 詳細は `help(Distributions)` を参照してほしい.

4.7. 参考文献

1. 福島正俊著「確率論 (第 5 版)」, 裳華房 (2006 年).
2. U. リゲス著, 石田基広訳「R の基礎とプログラミング技法」, 丸善出版 (2012 年).
3. 竹村彰通著「統計 (第 2 版)」, 共立出版 (2007 年).
4. 東京大学教養学部統計学教室編「統計学入門」, 東京大学出版会 (1991 年).
5. 吉田朋広著「数理統計学」, 朝倉書店 (2006 年).