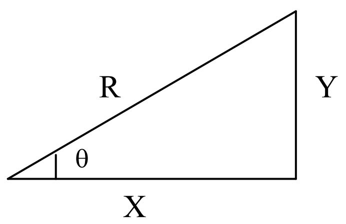
Dr. Halil Yurdugül
yurdugul@hacettepe.edu.tr

Standart Normal Dağılıma Uygun Veri Üretmek için Simülasyon (Benzeşim) Çalışması

Simülasyon çalışmaları, kuramsal dağılıma uygun veriler ile çalışma gerektiğinde başvurulan bir yöntemdir. Bazı durumlarda gerçek veriler yerine fonksiyondan türetilen veriler ile çalışmak tercih edilir. Bu tür çalışmaların avantajı, deneysel tasarımdaki varsayımları sağlamış olmaktır. Örneğin, istatistiksel doğrusal modellerin en önemli varsayımlarından birisi normal dağılıma sahip verilerin varlığıdır. Simülasyon çalışmalarından üretilmiş veriler bu varsayımları sağlar niteliktedir.

Günümüze kadar bir çok fonksiyona ilişkin veri türetme algoritmaları geliştirilmiştir. Ancak, istatistiksel olarak en yaygın kullanılan fonksiyon normal dağılım ya da normal yoğunluk fonksiyonu olduğu için bu tür veriler en çok simüle edilen dağılım biçimidir. Normal dağılıma uygun en önemli simülasyon çalışması "Box-Muller yöntemi" ya da "Box Muller dönüşümü" olarak bilinen yöntemdir.

Bu yöntem basitçe; birim çember ilişkilerinden yola çıkılarak oluşturulur.



X ve Y standart normal dağılıma sahip iki rastlantı değişkeni olsun.

$X, Y \sim N(0,1)$

Diğer taraftan çemberin denklemi: $\mathbf{R^2}=X^2+Y^2$
$\boldsymbol{\theta}=\tan^{-1}(Y/X)$

Burada R^2 (üstel dağılır) ve θ'nın (Tekbiçimli-Uniform dağılır) marjinal dağılımlarının parametreleri sırasıyla;

$R^2 \sim Exp(1/2)$
$\theta \sim Uniform(0,2\pi)$.

R^2 ve θ 'nın bileşik (jointly) dağılım fonksiyonu aşağıdaki gibidir.

$$f_{R,\theta} = \frac{1}{\sqrt{2\pi}} e^{-d/2} . \frac{1}{2} = \frac{1}{2} e^{-d/2} \frac{1}{2\pi}$$

Bu sürecin işletilmesiyle aşağıdaki eşitliklere ulaşılabilir.

$X = R\cos\theta$
$Y = R\sin\theta$

Burada X ve Y standart normal dağılıma sahip bağımsız rastlantı değişkenleridir $X,Y \sim N(0,1)$. Aynı zamanda θ ise uniform dağılıma sahip raslantı değişkenidir. Bilgisayarlarda tek değişkenli normal dağılıma sahip veri üretmek için bu ilişkiler aşağıdaki biçimde kullanılırlar.

$X = \cos(2\pi U_1)\sqrt{-2\log U_2}$

Dr. Halil Yurdugül
yurdugul@hacettepe.edu.tr

$$Y=\mathrm{Sin}(2\pi U_1)\sqrt{-2\log U_2}$$

Simülasyon için kaynak kod:

```
u = RND
v = RND
r = sqr(-2 log(v))
t = 2 * π * u
x = r * cos(t)
y = r * sin(t)
```

C kodu:

```
float x1, x2, w, y1, y2;

do {
x1 = 2.0 * ranf() - 1.0;
x2 = 2.0 * ranf() - 1.0;
w = x1 * x1 + x2 * x2;
} while ( w >= 1.0 );

w = sqrt( (-2.0 * ln( w ) ) / w );
y1 = x1 * w;
y2 = x2 * w;
```

VB Kodu:

```
Function gauss()
    Dim fac As Double, r As Double, V1 As Double, V2 As Double
10      V1 = 2 * Rnd - 1
        V2 = 2 * Rnd - 1
        r = V1 ^ 2 + V2 ^ 2
        If (r >= 1) Then GoTo 10
        fac = Sqr(-2 * Log(r) / r)
        gauss = V2 * fac
End Function
```

Not: Her bir standart normal dağılım değişkeninin karesi, 1 serbestlik derecesi ile ki-kare dağılımı gösterdiğinden dolayı, ($Z_1^2 \sim \chi^2_{(1)}$) bu ilişkiye dayalı olarak istenilen serbestlik derecesine sahip ki-kare dağılım verisi elde edilebilir.

$$Z_1^2 + Z_2^2 + \ldots\ldots + Z_n^2 \sim \chi^2_{(n)}$$

Aynı şekilde, n=2m olarak alındığında;

$$-2\log(u_1)\ -2\log(u_2)-\ldots\ldots\ -2\log(u_m) \sim \mathrm{Exp}(2)$$

Dr. Halil Yurdugül
yurdugul@hacettepe.edu.tr

Aynı şekilde, n serbestlik dereceli *t* dağılım verisi elde edilmek istendiğinde;

$$t=n^{1/2}Z_{n+1}/(Z_1^2+Z_2^2+.....+Z_n^2)^{1/2}$$

Aynı şekilde, n ve m serbestlik dereceli *F* dağılım verisi elde edilmek istendiğinde;

$$F=(n/m)(Z_1^2+Z_2^2+.....+Z_m^2)/(Z_{m+1}^2+Z_{m+2}^2+.....+Z_{m+n}^2)$$

İlişkili Verilerin Üretimi

X ve Y standart normal dağılıma sahiptir ve her ikisi de bağımsızdır ($\rho=0$). Ancak bazı durumlarda bağımsızlık ortadan kaldırılabilir. Bir diğer ifadeyle belirli bir r değerine sahip iki rastlantı değişkenine sahip veriler üretebilmek için aşağıdaki eşitlikten yararlanılabilir.

X ve X_1 yukarıda anlatıldığı gibi 2 bağımsız rastlantı değişkenidir X, X_1~N(0,1).

$$Y=\rho X+X_1(1-\rho^2)^{1/2}$$

Bu durumda ρ, X ve Y arasındaki korelasyonu göstermektedir. Üretilen yeni Y ise Y~N(0,1) dağılımına sahip yeni rastlantı değişkenidir. X ve Y bağımsız değil, aralarındaki bağımlılık miktarı ise r ile ifade edilebilir. Bu şekilde, önceden belirli bir korelasyona sahip normal dağılımlı veriler üretmek olanaklıdır.

```
Dim Z1(), Dim Z2()
    Rho=0.5
    Randomize Timer
    U = Rnd
    V = Rnd
    r = Sqr(-2 * Log(V))
    t = 2 * 3.14 * U
    X = r * Cos(t)
    Y = r * Sin(t)

    Z1() = r * Sin(t)
    Z2() = Rho * Y + Sqr(1-Rho) * x
```

Yukarı kodda üretilen Z1 ve Z2 N(0,1) dağılıma sahiptir ve üretilen veri kümeleri arasındaki ilişki miktarı 0.5'tir. Bu şekilde, istenilen korelasyon değerine sahip normal dağılıma uygun veriler üretmek olanaklıdır.

Dr. Halil Yurdugül
yurdugul@hacettepe.edu.tr

C Dilinde yazılmış program

```
#include <math.h>

extern float ranf();

float box_muller(float ort, float std_sapma)
{
        float x1, x2, w, y1;
        static float y2;
        static int use_last = 0;

        if (use_last)
        {
                y1 = y2;
                use_last = 0;
        }
        else
        {
                do {
                        x1 = 2.0 * ranf() - 1.0;
                        x2 = 2.0 * ranf() - 1.0;
                        w = x1 * x1 + x2 * x2;
                } while ( w >= 1.0 );

                w = sqrt( (-2.0 * log( w ) ) / w );
                y1 = x1 * w;
                y2 = x2 * w;
                use_last = 1;
        }

        return(ort + y1 * std_sapma );
}
```

Dr. Halil Yurdugül
yurdugul@hacettepe.edu.tr

ADA Dilinde yazılmış program

```
with RANDOM, Generic_Elementary_Functions;

function Box_Muller(Ortalama : in float; Std_Sapma : in float) return float
is
    x1, x2, w, y1 : float;

    package float_funcs is new Generic_Elementary_Functions( float );
    use float_funcs;

    begin
             loop
               x1 := Random.Uniform(-1.0, 1.0);
               x2 := Random.Uniform(-1.0, 1.0);

               w  := x1 * x1 + x2 * x2;

               exit when w < 1.0;
             end loop;

             w := sqrt( ( -2.0 * log( w ) ) / w );

             y1 := x1 * w;

      return Ortalama + y1 * Std_Sapma;

    end Box_Muller;
```