

5.22 モンテカルロシミュレーション 相関と多変量分布 ver.0.1

分散共分散行列がわかっているときの相関する 2 系列の標準正規乱数については、すでに Kernel の基礎(2)のところで Bivariate のケースとして扱った。基本的なことであるが、これを一般化してもう少し話を進めよう。

プログラム

```
new; cls;
n=3000;
Sig={1 0.2 0.3,
      0.2 2 0.6,
      0.3 0.6 3};
Z=rndn(n,3);
X=Z*chol(Sig);      /* (n x 3) x (3 x 3) */
print vcx(X);
```

画面表示

| | | |
|------------|------------|------------|
| 0.99104409 | 0.19521450 | 0.30468144 |
| 0.19521450 | 1.9860461 | 0.59426691 |
| 0.30468144 | 0.59426691 | 2.9520589 |

ソフトウェアや Cholesky 分解の値を前からかけるか後ろからかけるかによって、転置の記号がつくのかつかないのかが決まるので注意が必要。上では、 $n \times 3$ のデータに後ろから「行列の平方根」である分散共分散行列の Cholesky 分解の値をかけている。これはちょうど 1 系列の標準正規乱数から分散の平方根の をかけると、一般の正規乱数に変換されることと同じ要領である。ここでは のように分散共分散行列となっているところが違う。異なるソフトウェアを用いていたたり、前からかける公式を用いていたたりして、設定したの値と計算結果の 3 系列の変数の分散共分散行列結果があまりにも違う場合には、たいていの場合、転置の問題を考えると解決をする。上では、3 系列の変数であるが、一般的な k 系列のケースも を $k \times k$ にして、 Z を 3 列ではなくて k 列分作成しておくことで同様にプログラムができる。なお、 の対角（左上から右下への）行列はそれぞれの系列の分散の値であり、その他のオフダイアゴナルの値は共分散の値である。例えば、1 行 2 列目の値は第 1 系列と第 2 系列の変数の共分散の値である。また、それは 2 行 1 列目の値に一致していなければならない。つまり、この分散共分散行列は対称行列となっている。

さらに、それぞれの系列の平均が上では 0 であったわけだが、これを何か違う平均値にするのには、GAUSS では行列にベクトルを足すとベクトルが行列と同じ配列に自動複製されて同じものが足し合わせられることから、 X が $n \times 3$ で出てきているのなら、これに 1×3 の行ベクトルで与えられる平均を足し合わせれば一般的な多変量正規乱数となる。

プログラム

```
new; cls;
n=3000;
mu={1 3 -2};
Sig={1 0.2 0.3,
      0.2 2 0.6,
      0.3 0.6 3 };
X=rndmnorm(n,mu,Sig);
print meanc(X);
print vcx(X);

proc rndmnorm(n,mu,Sig);
  local Z,X;
  Z=rndn(n,rows(Sig));
  X=mu+Z*chol(Sig);
  retp(X);
endp;
```

画面表示

```
1.0058823
3.0418584
-2.0450070

0.95363234    0.20623632    0.31073076
0.20623632    1.9499005    0.60060296
0.31073076    0.60060296    3.0543620
```

上のプログラムは procedure 形式にしてある。X=mu+Z*chol(Sig);というところは、単変量の標準正規乱数から平均 μ 標準偏差 の正規乱数への変換とほとんど同じもので、これはその行列版であるということを理解しよう。なお、Light 版等ではワークスペースの関係上あまり多くの乱数を作成できないが、正式版で n を大きくしてより多くの乱数を作成すれば平均および分散共分散行列は設定値に近づく。下記がこの乱数生成の要領である。

$$\begin{array}{ccccc} X & = & \mu & + & Z * \text{chol}(\quad) & \text{ただし、Z は標準正規乱数} \\ (n \times 3) & & (1 \times 3) & & (n \times 3) (3 \times 3) & \\ & & : & & & \\ & & \text{複製} & & & \end{array}$$

2 変量変数の相関係数

2 変量変数の相関をシミュレーションするには、上で行なった Cholesky 分解をかける方法でも当然のことながら発生できる。その場合には、

$$\rho = \frac{Cov(X1, X2)}{\sqrt{Var(X1)}\sqrt{Var(X2)}}$$

を計算して、2 × 2 の分散共分散 行列の対角行列成分が分散でオフダイアグナル成分が共分散であることを考慮に入れて、適当に行列の成分を変化させて作り出したい相関係数

を定めてシミュレーションすればよいだろう。ここでは、そうではなくて、2 変量変数に限定した方法をとろう。それは、X 1 と X 2 の系列をあらかじめ標準正規乱数で同じ数分だけ作成しておく。新しい相関係数 の関係にある 2 変量変数 U と V は

$$U = X1$$

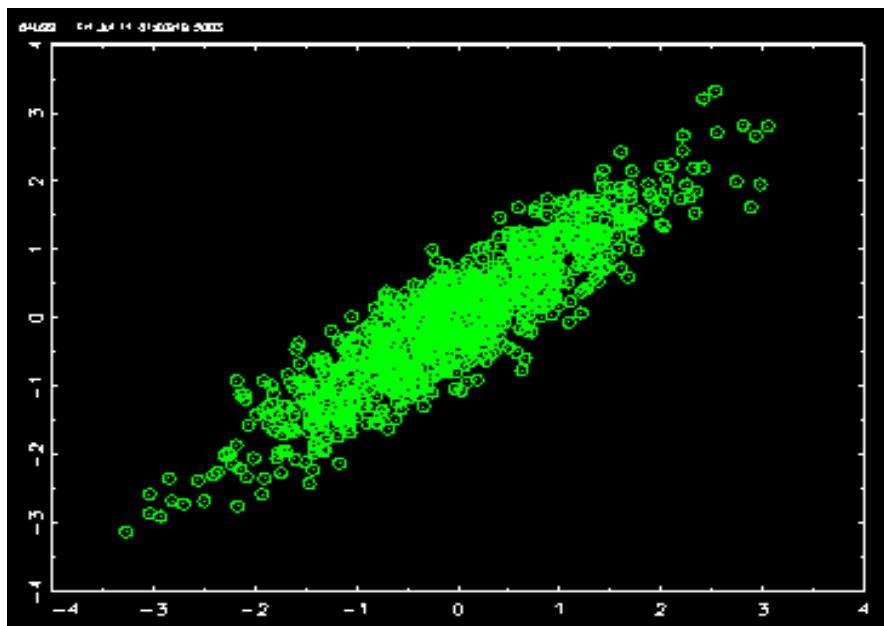
$$V = \rho X1 + \sqrt{1 - \rho^2} X2$$

として発生させればよい。なお、上のケースは U も V もともに 0 を中心とするものだが、0 以外をそれぞれの平均にするには、U および V にそれぞれ何か定数を足せばよい。式の形からもわかるように、U と V はともに $\sigma = 1$ の標準正規乱数になっている。

プログラム

```
new; cls;
rho=0.9;
X1=rndn(1000,1);
X2=rndn(1000,1);
U=X1;
V=rho*X1+sqrt(1-rho^2)*X2;
library pgraph;
graphset;
_plctrl=-1;
xy(U,V);
```

グラフ表示



の値を 0 に近づけると(0,0)のまわりに集中した丸い分布となり、 ρ をマイナスの値とすると右下がりの関係になる。 ρ は - 1 と 1 の間の数であるが、その絶対値が 1 に近いほど分布は横長になる。

ここで、2 変量の場合の ρ に関する変換式

$$U = X1$$

$$V = \rho X1 + \sqrt{1 - \rho^2} X2$$

は何を意味するのであろうか？それは 2×2 の行列でその係数を表現すると

$$\begin{pmatrix} U \\ V \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \rho & \sqrt{1 - \rho^2} \end{pmatrix} \begin{pmatrix} X1 \\ X2 \end{pmatrix} = \text{Chol}(\Sigma)' \begin{pmatrix} X1 \\ X2 \end{pmatrix}$$

というように、これは Σ が対角成分がすべて 1 で標準化された（その他の成分は 1 行 2 列目であれば 1 つ目の対角成分の平方および 2 つ目の対角成分の平方で割れば簡単に分散共分散行列を相関行列に変換できる）分散共分散行列、すなわち相関行列の Cholesky 分解を転置したものである。なお、 $X1$ と $X2$ の単体のベクトルにかけるときは転置した Cholesky 分解を前からかけるが（これは結果を 2×1 にするため）、データ全体を変換して $n \times 2$ の変換されたデータをつくるには、後ろから転置しないそのままのかたちの Cholesky 分解をかける。実際に実験してみると、

プログラム

```
new; cls;
```

```
rho=0.7;
```

```

Sig=1~rho|
    rho~1;
print chol(sig)';
print "          rho=" rho;
print "sqrt(1-rho^2)=" sqrt(1-rho^2);
画面表示
          1.0000000      0.00000000
          0.7000000      0.71414284
          rho=          0.70000000
sqrt(1-rho^2)=          0.71414284

```

相関する異なる分布への変換

上の例は相関係数 ρ のときの U と V の 2 変量正規乱数のシミュレーションした。さらに、これを標準正規分布の CDF を用いて一様乱数にそれぞれ変換できる。一様乱数ができあがれば、逆関数がわかっていればどのような分布にも変換することができる。ここでは、以前に取り上げた単純な三角分布 $f(x) = 2x$ を考えてその逆変換をやってみよう。

プログラム

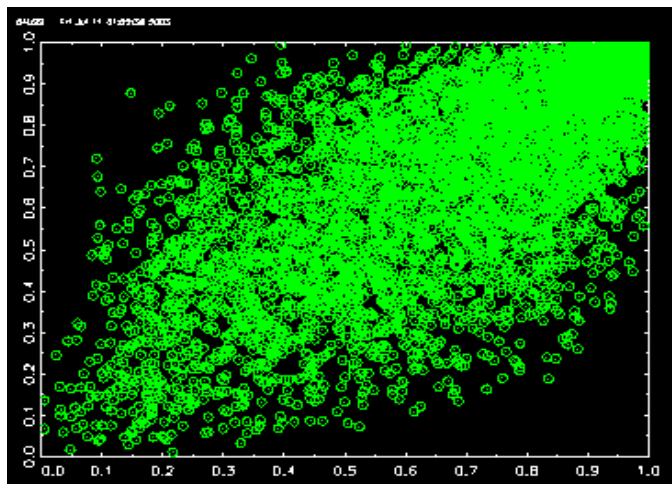
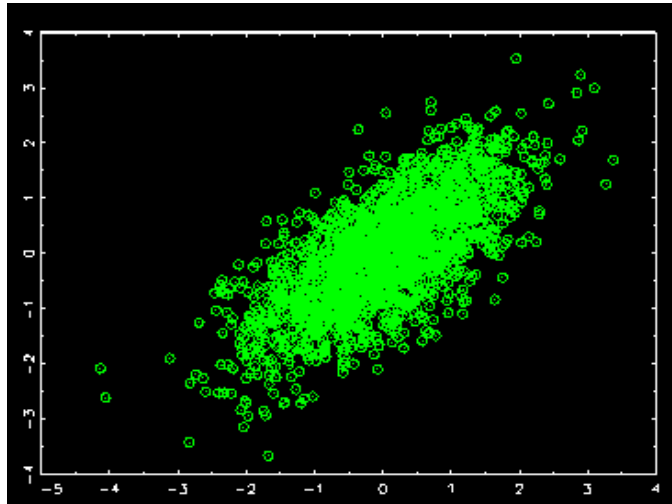
```

new; cls;
n=5000;
rho=0.7;
X1=rndn(n,1);
X2=rndn(n,1);
U=X1;
V=rho*X1+sqrt(1-rho^2)*X2;
library pgraph;
graphset;
pqgwin auto;
_plctrl=-1;
xy(U,V);
call hist(cdfn(U),19);
call hist(cdfn(V),19);
xy(Finv(cdfn(U)),Finv(cdfn(V)));

proc Finv(x);
    retp(sqrt(x)); /* Triangular case */
endp;

```

グラフ表示



上のグラフは ρ が 0.7 のときの 2 変量正規乱数のプロット、下のグラフは一様乱数の変換を経て (グラフは省略) 三角分布を仮定したシミュレーションである。当然のことながら、この場合の三角分布は 1 のところで最大の頻度になるので右上のプロットの密度が高くなっている。このケースは、三角分布を仮定しているが、逆関数がわかっている分布であるならば何でもかまわない。

t 分布を仮定したケースでは以下のようになる。

プログラム

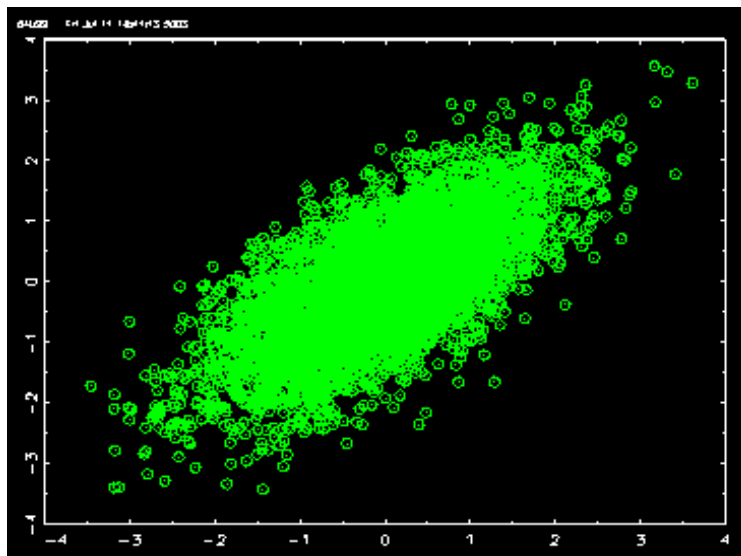
```
new; cls;  
n=5000;  
rho=0.7;  
df=3;  
X1=rndn(n,1);  
X2=rndn(n,1);
```

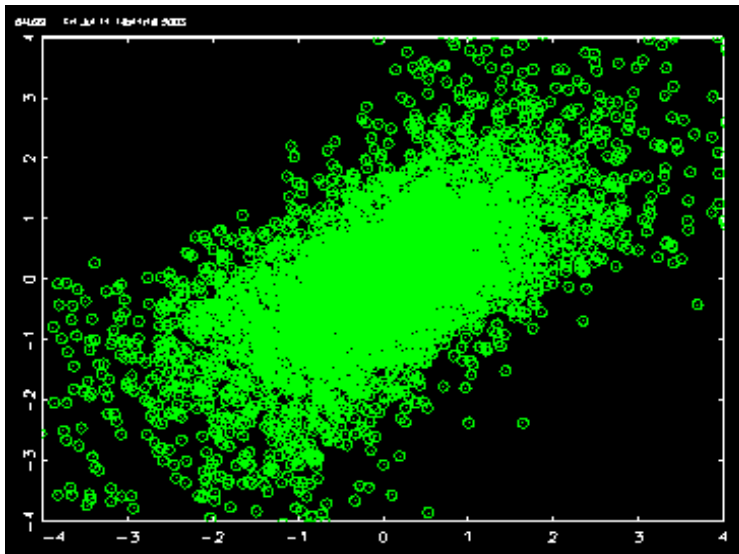
```

U=X1;
V=rho*X1+sqrt(1-rho^2)*X2;
library pgraph;
graphset;
pqgwin auto;
_plctrl=-1;
xy(U,V);
xtics(-4,4,1,0);
ytics(-4,4,1,0);
xy(Finv(cdfn(U),df),Finv(cdfn(V),df));

proc Finv(x,df);
    retp(cdfpci(1-x,df)); /* student-t with degree of freedom df */
endp;
グラフ表示

```





1 番目のグラフは通常的相关係数 $\rho = 0.7$ のときの 2 変量正規分布のプロットで、2 番目のグラフはそれぞれの系列を標準正規分布の CDF によって一様分布に変換したものを、さらに t 分布（ここでは自由度 3）に逆関数法で変換したものである。2 変量的相关関係を保ったまま、t 分布的相关関係になっている。前の三角分布のケースも、この t 分布のケースもどちらもそれぞれの系列でプロットしてみると、三角分布や t 分布の形状になることは言うまでもない。なぜなら、一様乱数から逆関数で変換しているだけだからだ。ただし、この場合相关関係を保持したまま変換がなされていることに注目しよう。上の t 分布のケースでは、GAUSS には t 分布の CDF の逆関数の complement を求める組込み関数は、すでに `cdftci` として存在するので、complement であることを考慮に入れて、`1-x` を第 1 インプットに、自由度 `df` を第 2 インプットに設定するように逆関数を設定してやれば簡単に一様分布から t 分布での逆変換がなされる。なお、t 分布は三角分布とは違って、定義域が $-\infty$ から $+\infty$ までの値を理論上とるのでグラフは間延びしてしまう。そのため、もともとの相关グラフの枠と同じ部分を表示するように `xticks` および `yticks` で -4 から 4 まで 1 ごとの大目盛で小目盛はなしの設定で区切るように、その他の範囲はグラフからカットしている。実際には、もっと広く複雑に広がっている。もともとの相关した 2 変量正規分布のグラフとくらべると、この相关した 2 変量 t 分布を同じ範囲の枠内で見てやると、少なくとも裾の部分で分厚く相关して分布することがわかるであろう。同様のことは、上のプログラムでは、`Finv` の関数のところに異なる分布の逆関数を設定することで簡単にできる。また、2 つの分布は必ずしも同じ分布や同じパラメーター（例えば同じ自由度）である必要はない。その場合も、できあがった 2 変量の系列を独自にプロットすると逆関数が意図した分布にそれぞれ独自にしたがっていることが簡単に確認できる。

相関する対数正規分布

相関する log-normal または multivariate する log-normal にしたがう乱数を生成するのは至って簡単である。あらかじめ正規分布の平均と分散共分散行列（または相関係数）がわかっているならば、それらにもとづいて相関する通常の正規分布を冒頭で扱った方法で作成しておいて、それを exp 変換すればよい。

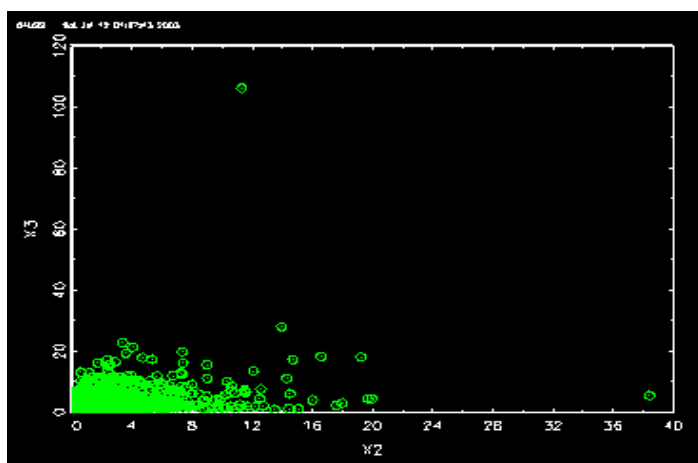
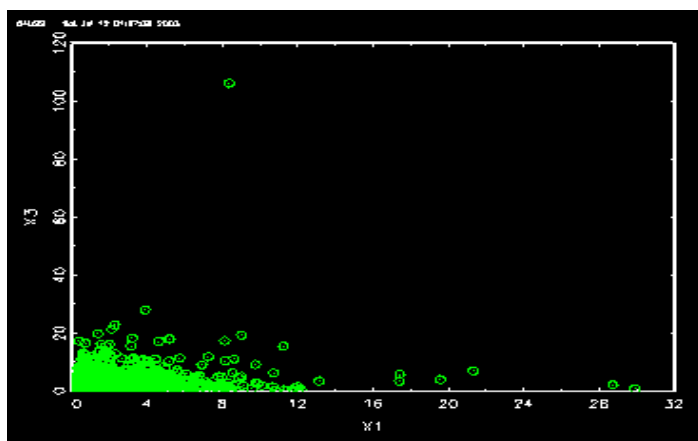
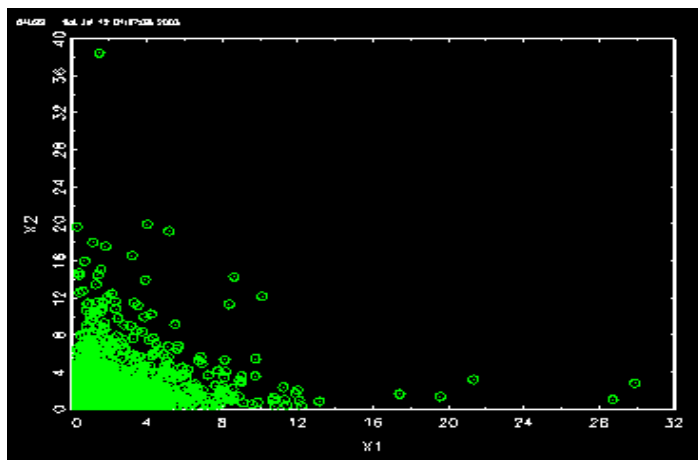
プログラム

```
new; cls;
n=3000;
mu={0 0 0};
Sig={0.9 0.2 0.3,
      0.2 1    0.6,
      0.3 0.6 1.2};
X=rndmlnnorm(n,mu,Sig);
library pgraph;
graphset;
pqgwin auto;
_plctrl=-1;
xlabel("X1"); ylabel("X2");
xy(X[:,1],X[:,2]);
xlabel("X1"); ylabel("X3");
xy(X[:,1],X[:,3]);
xlabel("X2"); ylabel("X3");
xy(X[:,2],X[:,3]);

proc rndmlnnorm(n,mu,Sig);
  local Z,Y,X;
  Z=rndn(n,rows(Sig));
  Y=mu+Z*chol(Sig);
  X=exp(Y);
  retp(X);
endp;
```

プログラムでは、procedure の中の最後に exp(Y)によって multivariate normal のXをまとめて変換していることに注目しよう。このプログラムは、対数正規のパラメータからは乱数を生成できないが、今まで扱ってきた正規乱数のパラメーターによる相関をそのまま変換するプログラムと同じである。一様乱数にするステップを踏む必要はない。

グラフ表示



それぞれの対数正規分布は正の定義域に 0 近くにピークがあって、徐々に広く減っていくので、2 変量の log-normal dependence を表示すると左下の部分に集中する。その散らばりは、もとの正規乱数の相関の度合いや平均の値による。

ポートフォリオの分散化

もう一度 multivariate normal のケースに戻って、その最大の応用例であるポートフォリオの分散化を考えよう。いま、同ウエイトの資産の数が 1 から n まで増えるものとする。それぞれの資産の個数に対する分散がどのように変化するのか、共分散が 0 の相関していないケースと、共分散が 0.3 で共通の相関しているケースをそれぞれ別々に考える。ただし、どちらも、それぞれの資産の分散は 1 であるものとして 100 回分の平均を見るものとする。

プログラム

```
new; cls;
n=20;
times=100;
s2=0.3;
call diversification(n,times,s2);

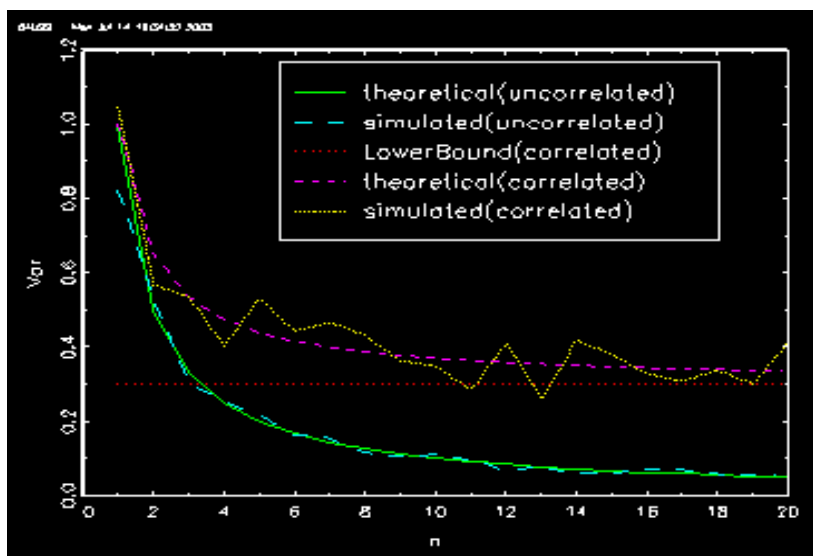
proc diversification(n,times,s2);
    local var0,var1,theovar0,theovar1,i,rbar,j,cov,s;
/* uncorrelated case */
    var0=zeros(n,1);
    theovar0=zeros(n,1);
    i=1;
    do while i<=n;
        rbar=zeros(times,1);
        j=1;
        do while j<=times;
            rbar[j]=sumc(rndn(i,1))/i;
            j=j+1;
        endo;
        var0[i]=vcx(rbar);
        theovar0[i]=1/i;
        i=i+1;
    endo;
/* correlated case */
    var1=zeros(n,1);
    theovar1=zeros(n,1);
    i=1;
    do while i<=n;
        rbar=zeros(times,1);
```

```

j=1;
do while j<=times;
    cov=s2*ones(i,i);
    cov=diagrv(cov,ones(i,1));
    s=chol(cov)';
    rbar[j]=sumc(s*rndn(i,1))/i;
    j=j+1;
enddo;
var1[i]=vcx(rbar);
theo0var1[i]=(1-s2)/i+s2;
i=i+1;
enddo;
/* graph */
library pgraph;
graphset;
xlabel("n");
ylabel("Var");
_plegctl={2 7 3 4};
_plegstr="theoretical(uncorrelated)¥000simulated(uncorrelated)
¥000LowerBound(correlated)¥000theoretical(correlated)¥000simulated(correlated)";
xy(seqa(1,1,n),theo0var0~var0~s2*ones(n,1)~theo0var1~var1);
retp(var0~var1);
endp;

```

画面表示



上のように、明らかに資産間に正の共分散が存在するケースでは、実現できる分散には下限がある。それに対して、共分散が0のケース（この場合は相関係数が0と考えても同じことである）では、実現できる分散には下限はほとんどない。nを増やせば限りなくゼロに近づく。プログラムでは、正の共分散の存在するケースでは、最初に資産の数の $i \times i$ の共通する共分散 s^2 の行列を作っておいてから、その対角成分を1の列ベクトルに置き換えている。その Cholesky 分解を計算したものを毎回 s として、 i 個の標準正規乱数に前からかけ合わせている。それを平均したものを考え、それを $times$ 回繰り返してその個数の資産に対する分散のシミュレートした値としている。これを $i = 1$ から例えば 20 個分まで繰り返している。なお、無相関のケースの理論的な分散値と資産の個数 n との関係は

$$r = \frac{1}{n} \sum_{i=1}^n r_i \text{ の平均に対して}$$

$$\text{var}(r) = \frac{1}{n^2} \sum_{i=1}^n \sigma^2 = \frac{\sigma^2}{n}$$

となる。上のケースでは $\sigma^2 = 1$ となっているから、分散の理論値は $1/n$ である。他方、共分散が 0.3 の場合の理論的な分散値と資産の個数 n との関係は

$$\begin{aligned} \text{var}(r) &= E \left[\sum_{i=1}^n \frac{1}{n} (r_i - \bar{r}) \right]^2 = \frac{1}{n^2} E \left[\left(\sum_{i=1}^n (r_i - \bar{r}) \right) \left(\sum_{j=1}^n (r_j - \bar{r}) \right) \right] = \frac{1}{n^2} \sum_{i,j} \sigma_{ij} \\ &= \frac{1}{n^2} (n\sigma^2 + 0.3(n^2 - n)\sigma^2) = \frac{(1-0.3)\sigma^2}{n} + 0.3\sigma^2 \end{aligned}$$

となる。ここでも同様に $\sigma^2 = 1$ となっているから、分散の理論値は $(1-0.3)/n + 0.3$ となる。したがって、資産の数 n が大きくなれば、無相関のケースの分散はゼロに近づいていくものの、そうでない相関するケースでは共通する共分散の値 0.3 という下限に近づいていくだけである。このシミュレーションは資産の分散投資の根拠となる理論を説明したものであり、特にその場合、十分な分散化には無相関な資産どうしを選んでくる必要があることをこの例は示している。

相関する2つの幾何ブラウン運動

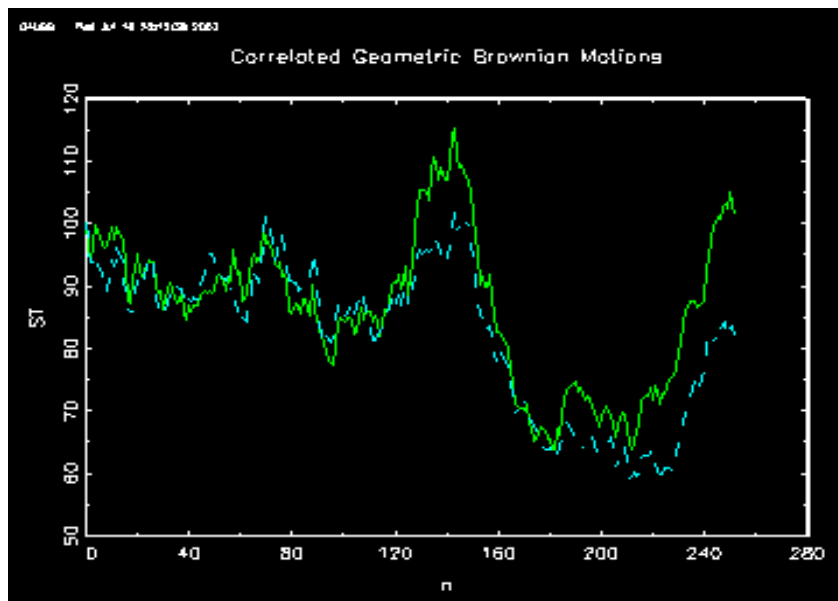
ここでは標準的な幾何ブラウン運動の系列の作り方を応用して、その2つの系列が互いに相関係数 ρ で相関しているケースを考えよう。

プログラム

```
new; cls;
S01=100; S02=100;
mu1=0.10; mu2=0.10;
sig1=0.40; sig2=0.40;
rho=0.7;
delt=1/252;
n=252;
call corrgrbr(S01,S02,mu1,mu2,sig1,sig2,rho,delt,n);

proc corrgrbr(S01,S02,mu1,mu2,sig1,sig2,rho,delt,n);
    local Sig,e,v,St1,St2,i;
    Sig=sig1^2~sig1*sig2*rho |
        sig1*sig2*rho~sig2^2;
    e=rndn(n,2);
    v=e*chol(Sig);
    St1=S01 | zeros(n,1);
    St2=S02 | zeros(n,1);
    i=1;
    do while i<=n;
        St1[i+1]=exp((mu1-(sig1^2)/2)*delt+v[i,1]*sqrt(delt))*St1[i];
        St2[i+1]=exp((mu2-(sig2^2)/2)*delt+v[i,2]*sqrt(delt))*St2[i];
        i=i+1;
    endo;
/* graph */
    library pgraph;
    graphset;
    title("Correlated Geometric Brownian Motions");
    xlabel("n");
    ylabel("St");
    xy(seqa(0,1,n+1),St1~St2);
    retp(St1~St2);
endp;
```

グラフ表示



なお、上のグラフは実行することにより乱数の出具合で変化する。なお、この作成方法は、

$$\ln S(t_{k+1}) - \ln S(t_k) = \mu \Delta t + \sigma \varepsilon(t_k) \sqrt{\Delta t}$$

$$S(t_{k+1}) = e^{\mu \Delta t + \sigma \varepsilon(t_k) \sqrt{\Delta t}} S(t_k)$$

ただし、 $\varepsilon = \mu - \sigma^2/2$ の1変量の幾何ブラウン運動の通常の作成方法では $(t_k) \sim N(0,1)$ となる標準正規乱数を設定するのに対して、 Σ を含む行列で表された分散共分散行列

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_1 \sigma_2 \rho \\ \sigma_1 \sigma_2 \rho & \sigma_2^2 \end{pmatrix}$$

を Cholesky 分解したものを2列の標準正規乱数にかけてやることによって、上の公式にある ε も含んだ部分の (t_k) を逐次に計算して、それぞれの系列の μ と σ について通常どおりの幾何ブラウン運動のシミュレーションを行なっている。分散共分散行列 Σ を標準化して対角成分を1にするとオフダイアゴナルは ρ になることでも、これらの2変量は ρ という相関係数で相関するようにシミュレーションされていることがわかるであろう。

上では2変量だけの場合であるが、これを共通の初期値 S_0 、共通の μ 、共通の σ_i についてポートフォリオの分散化のところで行なったのと同じテクニックを使って、対角成分は σ_i^2 として、その他の成分を $\sigma_i \sigma_j \rho_{ij}$ とすることによって、共通の相関係数 ρ_{ij} に対する多変量の幾何ブラウン運動を作成してみよう。

プログラム

```
new; cls;
vol=35;
S0=100;
```

```

mu=0.10;
sigi=0.40;
rho=0.7;
delt=1/252;
n=252;
call mgbr(vol,S0,mu,sigi,rho,delt,n);

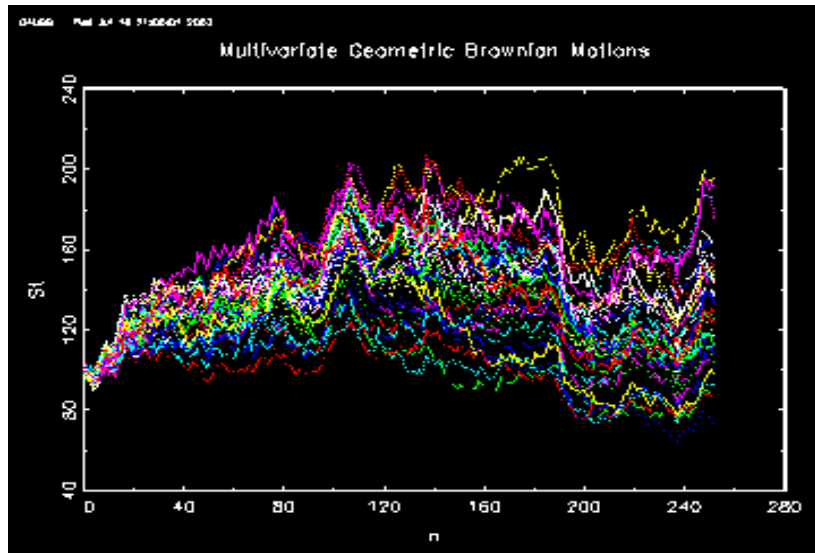
proc mgbr(vol,S0,mu,sigi,rho,delt,n);
    local Sig,e,v,St,i,j;
    Sig=sigi*sigi*rho*ones(vol,vol);
    Sig=diagrv(Sig,sigi^2*ones(vol,1));
    e=rndn(n,vol);
    v=e*chol(Sig);
    St=S0*ones(1,vol) | zeros(n,vol);
    i=1;
    do while i<=n;
        j=1;
        do while j<=vol;
            St[i+1,j]=exp((mu-(sigi^2)/2)*delt+v[i,j]*sqrt(delt))*St[i,j];
            j=j+1;
        endo;
        i=i+1;
    endo;
/* graph */
    library pgraph;
    graphset;
    title("Multivariate Geometric Brownian Motions");
    xlabel("n");
    ylabel("St");
    xy(seqa(0,1,n+1),St);
    retp(St);
endp;

```

上のプログラムでは、まず $\text{vol} \times \text{vol}$ のディメンションの $i \ i$ からなる を作っておいて、その対角成分に i^2 を上書きすることで、分散共分散行列を作成してその Cholesky 分解したものを標準正規乱数にかけている。外側の i ループはそれぞれの 1 から n 期までの幾何

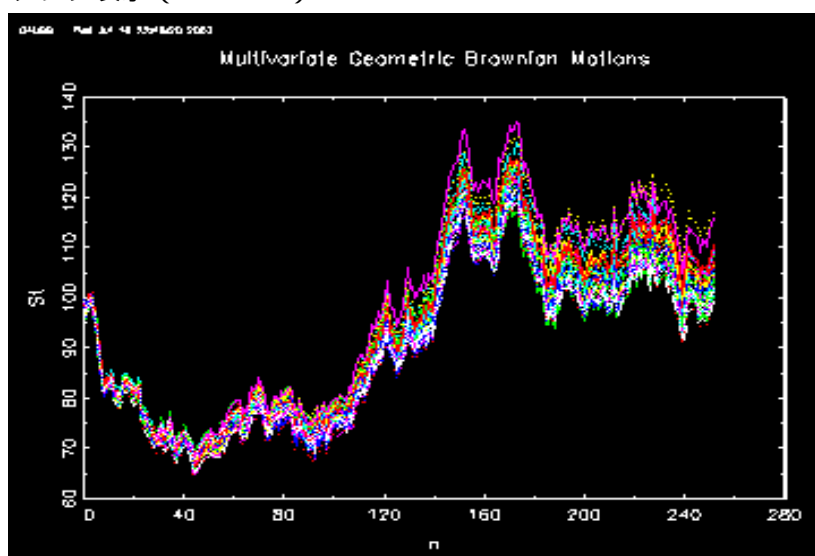
ブラウン運動生成の過程であり、内側の j ループは 1 から vol までの第 j 番目の系列についてのものである。残念ながら、配列展開が膨大になるため、あまり大きい vol の数に対してはプログラムを動かすことはメモリ上の制約がある。

グラフ表示 ($\rho = 0.7$ のケース)

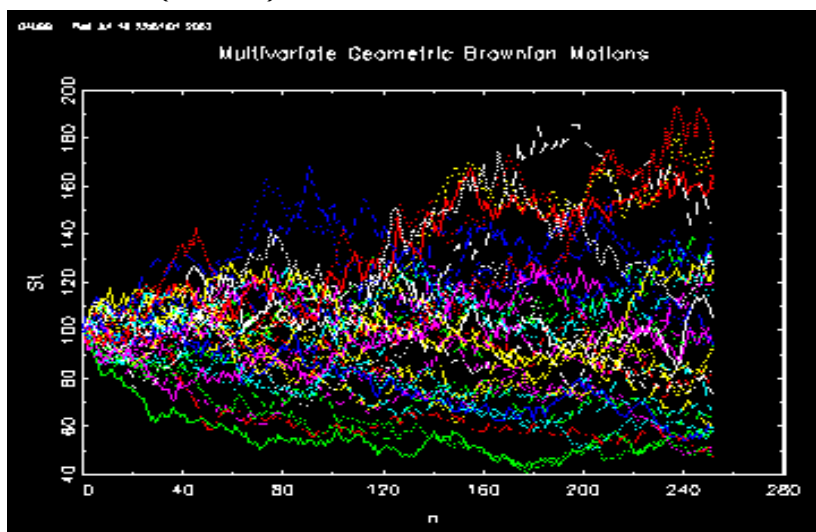


なお、上のグラフはそれぞれ 1 回限りの幾何ブラウン運動が互いに多重相関しているものであって、何度も繰り返した平均をとる作業ではないので注意されたい。当然のことながら、乱数の出具合によってグラフは変化する。また、共通の相関係数 ρ を 1 に近づけるとこれよりも相関関係ははっきりと表れる。その反対に相関係数 ρ を 0 に近づけると全くばらばらな振舞いになってくる。

グラフ表示 ($\rho = 0.99$)



グラフ表示 ($\rho = 0$)



最後の $\rho = 0$ の無相関のケースは、通常の幾何ブラウン運動のシミュレーションの vol 回数分と同じことである。これをさらに数千回から数万回繰り返してその平均をとるとそれぞれの時間 n (ただし、例えばこの場合 $\Delta t = 1/252$) に対する推定値となる。なお、上の 3 つのグラフは共通の乱数について描かれているものではないこと、また、 St の縦軸の縮尺は GAUSS のシステムが自動的に決めていて一定ではないことに注意しよう。

あえて、それらに対処するようにプログラムしていないが、e の乱数発生させる直前に `rndseed 数字;` という乱数の出具合を一定化させるコマンドと、xy グラフの直前に `ytics(0,250,20,0);` という y 軸の範囲と目盛の間隔を決めるコマンドを挿入すれば、 ρ をかえてプログラムを実行しても共通の乱数の出具合に対する比較ができるはずである。なお、乱数シードの数は任意の自然数、ytics の数字の意味は、0 から 250 までの範囲を 20 という大きな目盛で区切り小目盛はなしという意味である。