SASを用いたコピュラに従う擬似乱数の生成

〇矢田 真城 [1]　　浜田 知久馬 [2]

[1] 株式会社 ACRONET データサイエンス本部 生物統計部　[2] 東京理科大学 工学部 経営工学科

Generating pseudo-random numbers from copulas by using SAS system

Shinjo Yada[1]　Chikuma Hamada[2]

[1] Biostatistics Department, ACRONET Corporation

[2] Department of Management Science, Tokyo University of Science

要旨

コピュラ(copula)とは，変数間の依存関係を表す接合関数のことである．複数の変量についてモデル化したい場合，各変量の周辺分布とコピュラとを別々に設定することで，様々な多変量確率分布をあてはめることができる．

医薬品開発において，想定した試験デザインの動作特性を評価するためシミュレーションを行うが，その際，コピュラに従う擬似データを生成しなければならないケースが考えられる．そこで本稿では，代表的なコピュラの乱数生成法について整理し，SAS を用いてこれらの擬似乱数を生成するための具体的な方法を紹介する．

キーワード：擬似乱数，コピュラ，マーシャルオルキン法，COPULA プロシジャ

1. はじめに

医薬品開発において，検討対象となる試験デザインの振舞いをうまく数式で表せない，あるいは数式で表現できてもその解を解析的に求めることが極めて困難な場合，近似的な解を得るためにシミュレーションは威力を発揮する．そのようなシミュレーションでは，大量の乱数を必要とすること，再現性が求められることから，擬似乱数が用いられる[1].

擬似乱数が従う分布は，試験デザインの評価項目に依存し，コピュラによるモデリングが有効な場合がある．SAS では，擬似乱数を生成させるために RAND 関数が提供されている[2]が，コピュラの場合，RAND 関数で指定できない確率分布が必要とされる．そこで本稿では，正規コピュラ(Normal copula)，スチューデント *t* コピュラ(Student’s t copula)，クレイトンコピュラ(Clayton copula)[3]，フランクコピュラ(Frank copula)[4]，ガンベルコピュラ(Gumbel copula)[5]を取りあげ，各コピュラに従う乱数を生成させる方法について述べる．そして，これら 5 種類のコピュラに対し，SAS を用いて乱数を生成するための方法について具体的に説明する．

2. コピュラについて

m 個の確率変数を $X_1, X_2, \ldots, X_m$ とおき，これらの周辺分布関数を $F_1(x_1), F_2(x_2), \ldots, F_m(x_m)$，同時分布関数を $F(x_1, x_2, \ldots, x_m)$ と表すとき，スクラーの定理(Sklar's theorem)として

$$F(x_1, x_2, \ldots, x_m) = C(F_1(x_1), F_2(x_2), \ldots, F_m(x_m)) \tag{2.1}$$

を満たす関数 C が存在することが知られており，この関数 C がコピュラである[6-8]．これは，関数 C が多次元同時分布とその 1 次元周辺関数を結合する役割を担っていることを示している．よって，m 個の確率変数 $X_1, X_2, \ldots, X_m$ についてモデル化したい場合，個々の周辺分布関数 $F_1(x_1), F_2(x_2), \ldots,$ $F_m(x_m)$ と，これら周辺分布関数の依存関係を表す関数 C，即ちコピュラとを別々に特定することで，$X_1, X_2, \ldots, X_m$ の同時分布関数を構築することができる．

(2.1)において，関数 F が連続関数である場合には C は一意に定まり，任意の $u_i = F_i(x_i)$（ここに i=12,…,m で $u_i \in [0,1]$）に対し

$$C(u_1, u_2, \ldots, u_m) = F(F_1^{-1}(u_1), F_2^{-1}(u_2), \ldots, F_n^{-1}(u_m)) \tag{2.2}$$

と与えられる．このとき C は，各周辺分布が区間[0,1]の一様分布である同時分布関数となる．

変量間の依存関係を行列で表現するコピュラとして，正規コピュラ，スチューデント t コピュラが挙げられる．またそれ以外に，変量間の依存関係を 1 種類のパラメータで表現するコピュラがあり，その代表例がアルキメディアンコピュラ(Archimedean copula)[6-8]である．アルキメディアンコピュラとは，生成素(generator)と呼ばれる関数を用いて表現されるコピュラの総称であり，クレイトンコピュラ，フランクコピュラ，ガンベルコピュラが該当する．

3. 乱数の生成

3.1. アルキメディアンコピュラに従う乱数の生成

区間 (0,1] 上で定義され，0 を含む正の実数値をとる単調減少凸関数 ϕ が

$$\phi(1) = 0 \tag{3.1}$$

を満たすとする．このとき

$$C(u_1, u_2) = \phi^{-1}(\phi(u_1) + \phi(u_2)) \quad ここに\ (u_1, u_2) \in (0,1]^2 \tag{3.2}$$

として表現されるコピュラを，2 次元アルキメディアンコピュラと呼び，ϕ を C の生成素という．3 次元以上の場合，(3.1)及び $\lim_{t \to 0} \phi(t) = \infty$ を満たし ϕ^{-1} が完全単調(completely monotone)であるとき

$$C(u_1, u_2, \ldots, u_m) = \phi^{-1}(\phi(u_1) + \phi(u_2) + \ldots + \phi(u_m)), \quad (u_1, u_2, \ldots, u_m) \in (0,1]^m \tag{3.3}$$

として定義される．

アルキメディアンコピュラにおいて，よく用いられる乱数生成法の 1 つがマーシャルオルキン法(Marshall and Olkin method)[9]であり，マーシャルオルキン法を用いた乱数生成法は以下のようになる．手順 1 の $\zeta(\cdot)$ はラプラス変換である．乱数を生成させたいアルキメディアンコピュラに対し，生成素 ϕ の逆関数 $\phi^{-1}(\cdot)$ に対応するような確率分布 $F(\eta)$ を特定し，その確率分布に従う乱数を発生させる必要がある．

手順 1　C の生成素に対し，以下を満たす確率分布 $F(\eta)$ に従う乱数 η を生成させる.

$$\zeta(t)=\int e^{-t\eta}dF(\eta)=\phi^{-1}(t)$$

手順 2　区間[0, 1]の一様分布からの乱数 $x_1,x_2,..,x_m$ を生成させる.

手順 3　$u_i=\zeta(-\eta^{-1}\log(x_i))$ （i=1,2,…,m）を算出する.

3.2. 正規コピュラに従う乱数の生成[10]

平均ベクトル $\mathbf{0}$，分散が全て 1 となる分散共分散行列 $\boldsymbol{\Sigma}$ をもつ多変量正規分布が同時分布の場合，周辺分布は標準正規分布となる．従って，正規コピュラから 1 組の乱数 $u_1, u_2, \ldots, u_{\mathrm{m}}$ を生成するための手順は以下のようになる.

手順 1　所与の相関係数から，分散全て 1 の分散共分散行列を作成する.

手順 2　平均ベクトル $\mathbf{0}$，手順 1 で作成した分散共分散行列の多変量正規分布に従う乱数 $z_1, z_2,\ldots, z_m$ を生成させる.

手順 3　$u_i=\Phi(z_i)$　(i=1,2,…,m) を計算する．ここに $\Phi(\cdot)$ は標準正規分布の分布関数である.

3.3. スチューデント t コピュラに従う乱数の生成[10]

自由度 ν，相関行列 $\boldsymbol{\Sigma}$ ($m\times m$)をもつ多変量 t 分布が同時分布の場合，周辺分布は自由度 ν の t 分布となる．従って，スチューデント t コピュラから 1 組の乱数 $u_1,u_2,\ldots,u_m$ を生成するための手順は以下のようになる.

なお，自由度 ν，相関行列 $\boldsymbol{\Sigma}$ の多変量 t 分布に従う乱数 $x_1,x_2,\cdots,x_m$ は，平均ベクトル $\mathbf{0}$，分散共分散行列 $\boldsymbol{\Sigma}$ の多変量正規分布に従う m 個の乱数 $z_1,z_2,\cdots,z_m$ と，自由度 ν のカイ二乗分布に従う乱数 s とを用いて導出できる.

手順 1　自由度 ν，相関行列 $\boldsymbol{\Sigma}$ ($m\times m$)の多変量 t 分布に従う乱数 $x_1,x_2,\ldots,x_m$ を生成させる.

手順 2　$u_i=t_\nu(x_i)$　(i=1,2,…,m) を計算する．ここに $t_\nu(\cdot)$ は自由度 ν の t 分布の分布関数である.

4. SAS による擬似乱数の生成

コピュラを用いた例として，Yuan and Yin(2009)で用いられた 2 変量生存時間データを参考にした以下のモデルを考える.

T を毒性発現までの時間を表す非負の確率変数とし，T の生存関数としてパラメータ λ_T の指数分布

$$S_T(t_T)=\exp(-\lambda_T t_T) \tag{4.1}$$

を考える．効果発現までの時間 E についても，生存関数としてパラメータ λ_E の指数分布

$$S_E(t_E)=\exp(-\lambda_E t_E) \tag{4.2}$$

を仮定する．更に，毒性発現までの時間 T と効果発現までの時間 E との相関構造として，クレイトンコピュラ(4.3)を想定する.

$$S(t_T, t_E) = \{S_T(t_T)^{-\theta} + S_E(t_E)^{-\theta} - 1\}^{-1/\theta} \tag{4.3}$$

以上に示したモデルに従う生存時間(t_T, t_E)の擬似データは，以下 2 つのステップにより得ることができる.

<Step1> 想定したコピュラからの擬似乱数を生成する

<Step2> Step1 で生成された擬似乱数を変換する

Step1 において，想定したコピュラから擬似乱数を生成するにあたり，まず DATA ステップを用いた方法を説明する．次に，Ver.9.3 より SAS/ETS に搭載された COPULA プロシジャを用いた方法について紹介する.

4.1. DATA ステップを用いた擬似乱数の生成

Step1 では，クレイトンコピュラ

$$C(u_1, u_2) = (u_1^{-\theta} + u_2^{-\theta} - 1)^{-1/\theta} \tag{4.4}$$

に従う擬似乱数(u_1,u_2)を生成させる．クレイトンコピュラでは，生成素の逆関数はパラメータ θ^{-1} のガンマ分布に従う確率変数のラプラス変換に一致することから，マーシャルオルキン法を用いて，(4.4)で表されるクレイトンコピュラに従う擬似乱数を 1 組生成させるためのアルゴリズムは，以下のようになる.

手順 1　パラメータ $1/\theta$ のガンマ分布に従う擬似乱数 Z を 1 つ生成する.

手順 2　手順 1 で生成した擬似乱数 Z と独立に, 区間[0,1]の一様分布に従う擬似乱数 X_1, X_2 を生成させる.

手順 3　$U_i = (1 - \ln(X_i)/Z)^{-1/\theta}$ （i=1,2）を算出する.

プログラム 4.1 は，上記の手順 1 から手順 3 までを用いて 2 次元クレイトンコピュラに従う擬似乱数を生成させるための SAS マクロプログラムである. マクロ引数 theta にクレイトンコピュラのパラメータ $\theta(>1)$ を，マクロ引数 ndraws に生成する擬似乱数の個数を，マクロ引数 outuniform に生成した擬似乱数を保存するためのデータセット名を，それぞれ指定する.

プログラム 4.1　DATA ステップを用いたクレイトンコピュラの擬似乱数生成

```
%macro mclayton(theta,ndraws,outuniform);
  data &outuniform.;run;
  %do m_i=1 %to &ndraws.;
    data m_wrk ;
     length m_r1 m_r2 m_u 8.; format m_r1 m_r2 m_u best.;
     m_r1=rand('gamma',1/&theta.) ;
     do m_j=1 to 2;
       m_r2=rand('uniform') ;
       m_u =(1-log(m_r2)/m_r1)**((-1)/&theta.);
       output;
     end;
    run;
```

```
        proc transpose data=m_wrk out=m_wrk prefix=m_u;var m_u;run;
        data m_wrk ;set m_wrk;if n(m_u1,m_u2)=2 then SimNo=&m_i.;run;
        data &outuniform.;set &outuniform. m_wrk(keep=SimNo m_u1 m_u2);
          if SimNo^=.;
        run;
        proc datasets library=work nowarn nolist nodetails; delete m_wrk; run; quit;
    %end;
%mend mclayton;
```

Step2 では，Step1 において得られた擬似乱数(u_1,u_2) を用いて($S_T^{-1}(u_1)$, $S_E^{-1}(u_2)$)と変換することにより，同時分布関数(4.3)に従う擬似乱数を算出する．(4.1), (4.2)より

$$t_T = S_T^{-1}(u_1) = -\ln u_1 / \lambda_T \tag{4.5}$$

$$t_E = S_E^{-1}(u_2) = -\ln u_2 / \lambda_E \tag{4.6}$$

であるから，プログラム 4.1 により得られた擬似乱数(u_1,u_2)を(4.5), (4.6)へ代入することにより，(4.1)から(4.3)に示したモデルに従う擬似乱数を得ることができる. プログラム 4.2 は, (4.1), (4.2)においてパラメータを λ_T =1, λ_E =1 としたときの SAS プログラムである．SAS データセット randata に t1,t2 という変数名でデータが入力される．なお，プログラム 4.2 では，各症例の観察期間 O は区間[0, ω]の一様分布に従うものとし，i 番目のイベント発現までの時間(T_i,E_i)に対し O_i<T_i または O_i<E_i なら打ち切りとする[12]ように組んでいる．

プログラム 4.2　生存時間の擬似乱数を算出する SAS プログラム

```
data randata; set unifdata ;
    length lambda_T lambda_E t_1 t_2 8.;
    format lambda_T lambda_E t_1 t_2 best.;
    lambda_T=1; lambda_E=1;
    array unidata(2) m_u1 m_u2;
    array stime(2) t_1 t_2;
    array parm(2) lambda_T lambda_E ;
    do i=1 to 2; stime(i)=round((-1)*(log(1-unidata(i)))/parm(i),1e-12) ;end;
run;
data randata; set randata;
  length w 8.;format w best.; w=7;
  if rand('uniform')*w<t_1 | rand('uniform')*w<t_2 then censor=0; else censor=1;
run;
```

ここでは，2 つの生存時間を表す確率変数 T と E の依存構造として，クレイトンコピュラ(4.4)を想定したが，フランクコピュラ，ガンベルコピュラは，(3.2)を用いて統一的に表現でき，よってこれらのコピュラに

従う擬似乱数は，クレイトンコピュラと同様の手順で生成することができる．ただし以下に示すとおり，生成アルゴリズムはやや複雑になる．

・フランクコピュラの擬似乱数生成

フランクコピュラ

$$C(u_1,u_2,...,u_m)=\phi^{-1}(\phi(u_1)+\phi(u_2)+\cdots+\phi(u_m))$$

$$=-\frac{1}{\theta}\log\left[1+\frac{\prod_{i=1}^{m}(\exp(-\theta u_i)-1)}{(e^{-\theta}-1)^{m-1}}\right] \tag{4.7}$$

において生成素の逆関数は，パラメータ $\beta=1-e^{-\theta}$ の対数級数分布に従う確率変数のラプラス変換に一致する．よってフランクコピュラの乱数を生成するためには，パラメータ $\beta=1-e^{-\theta}$ の対数級数分布に従う乱数が必要となる．しかし，SAS の RAND 関数では対数級数分布に従う擬似乱数は生成させることができない．逆関数法を用いて擬似乱数を生成させる方法もあるが，ここでは，区間[0,1]の一様分布に従う確率変数 U とパラメータ 1 の指数分布に従う確率変数 V が互いに独立であるとき，

$$X=\left\lfloor-\frac{V}{\ln(1-(1-\beta)^U)}\right\rfloor+1 \quad \text{（ここに}\lfloor\cdot\rfloor\text{は・を超えない最大の整数値）} \tag{4.8}$$

によって定義される確率変数 X が対数級数分布に従うことを用いた[13]．この性質を用いれば，区間[0,1]の一様乱数 U とパラメータ 1 の指数分布に従う乱数 V を互いに独立に生成させ，(4.8)に代入することで，対数級数分布に従う乱数を得ることができる．従って，マーシャルオルキン法を用いて，2 次元フランクコピュラに従う擬似乱数を 1 組生成させるためのアルゴリズムは，以下のようになる．

手順 1　以下の手順にて対数級数分布に従う擬似乱数 Z を生成する．

手順 1-1　区間[0,1]の一様乱数 U とパラメータ 1 の指数分布に従う擬似乱数 V とを互いに独立に生成する．

手順 1-2　$Z=\left\lfloor-\frac{V}{\ln(1-(e^{-\theta})^U)}\right\rfloor+1$ を算出する．

手順 2　手順 1 で生成した擬似乱数 Z とは独立に，区間[0,1]の一様分布に従う擬似乱数 X_1,X_2 を生成させる．

手順 3　$U_i=-\log\{1+\exp(Z^{-1}\ln(X_i))(e^{-\theta}-1)\}/\theta$ （i=1,2）を算出する．

・ガンベルコピュラの擬似乱数生成

ガンベルコピュラ

$$C(u_1,u_2,...,u_m)=\phi^{-1}(\phi(u_1)+\phi(u_2)+\cdots+\phi(u_m))$$

$$=\exp\left\{-\left(\sum_{i=1}^{m}(-\log u_i)^{\theta}\right)^{1/\theta}\right\} \tag{4.9}$$

において，生成素の逆関数は，特性指数(characteristic exponent)が θ^{-1}，歪度パラメータが 1，尺度パラメータ

が$[\cos(\pi/2\theta)]^{\theta}$，位置パラメータが 0 の安定分布 $St(\theta^{-1},1,\ [\cos(\pi/2\theta)]^{\theta},0)$に従う確率変数のラプラス変換に一致する．従って，マーシャルオルキン法を用いて，2 次元ガンベルコピュラに従う擬似乱数を 1 組生成させるためのアルゴリズムは，以下のようになる．

手順 1　安定分布 $St(\theta^{-1},1,[\cos(\pi/2\theta)]^{\theta},0)$に従う擬似乱数 Z を生成する．

手順 2　手順 1 で生成した擬似乱数 Z とは独立に，区間[0,1]の一様分布に従う擬似乱数 X_1,X_2 を生成させる．

手順 3　$U_i = \exp(-(-\ln(X_i)/Z)^{1/\theta})$ （i=1,2）を算出する．

これを SAS で実装する場合，問題となるのは安定分布に従う擬似乱数の生成である．SAS の RAND 関数では（一部の特殊な場合を除いて）安定分布に従う擬似乱数を生成することができない．そこで，Weron のアルゴリズム[14]により標準安定分布 $St(\alpha,\beta,1,0)$に従う擬似乱数を生成させ，それを用いて任意のパラメータに対応する擬似乱数を作り出すことにした．

Step1　区間$[-\pi/2,\pi/2]$の一様乱数 U を生成する．

Step2　一様乱数 U とは独立にパラメータ 1 の指数分布に従う擬似乱数 V を生成する．

Step3　標準安定分布 $St(\alpha,\beta,1,0)$に従う擬似乱数 X を算出する．

1) $\alpha \neq 1$のとき

$$X = S_{\alpha,\beta}\frac{\sin(\alpha(U+B_{\alpha,\beta}))}{(\cos(U))^{\frac{1}{\alpha}}}\left(\frac{\cos(U-\alpha(U+B_{\alpha,\beta}))}{V}\right)^{\frac{1-\alpha}{\alpha}}$$

ここに

$$S_{\alpha,\beta} = \left(1+\beta^2\tan^2\left(\frac{\pi\alpha}{2}\right)\right)^{\frac{1}{2\alpha}},\ B_{\alpha,\beta} = \frac{\arctan\left(\beta\tan\left(\frac{\pi\alpha}{2}\right)\right)}{\alpha}$$

2) $\alpha = 1$のとき

$$X = \frac{2}{\pi}\left[\left(\frac{\pi}{2}+\beta U\right)\tan U - \beta\log\left(\frac{\frac{\pi}{2}V\cos U}{\frac{\pi}{2}+\beta U}\right)\right]$$

上記アルゴリズムにより得られた標準安定分布 $St(\alpha,\beta,1,0)$に従う擬似乱数 X に対し

$$Y = \begin{cases} \sigma X + \mu & (\alpha \neq 1) \\ \sigma X + 2\beta\sigma\log\sigma/\pi + \mu & (\alpha = 1) \end{cases} \tag{4.10}$$

とすれば，特性指数 α ($0<\alpha\le 2$)，歪度パラメータ β ($-1\le\beta\le 1$)，尺度パラメータ σ，位置パラメータ μ の安定分布 $St(\alpha,\beta,\sigma,\mu)$に従う擬似乱数となる[15]．

付録 A に，DATA ステップを用いて，クレイトンコピュラ，ガンベルコピュラ，フランクコピュラ，正規コピュラ，スチューデント t コピュラの擬似乱数を生成するプログラムを示した．少し補足すると，正規コ

ピュラ及びスチューデント t コピュラの場合，3.2 節，3.3 節に示したように，多変量正規分布の乱数が必要となる．多変量正規分布に従う乱数の生成は，分散共分散行列を Cholesky 分解することに基づいて行うことができる[5]ため，付録 A では MIXED プロシジャを用いて Cholesky 分解を行い，多変量正規分布に従う擬似乱数を生成させている[16].

4.2. COPULA プロシジャを用いた擬似乱数の生成

SAS では，Ver.9.3 より SAS/ETS に COPULA プロシジャが Experimental として搭載され，正規コピュラ，スチューデント t コピュラ，クレイトンコピュラ，フランクコピュラ，ガンベルコピュラに従う擬似乱数の生成が可能となった[10]．(4.4)で示されるクレイトンコピュラに従う擬似乱数を，COPULA プロシジャを用いて生成するための SAS プログラムを以下に示す.

プログラム 4.3　COPULA プロシジャを用いたクレイトンコピュラの乱数生成

```
proc copula;
   var U1-U2 ;
   define COP clayton (theta=8) ;
   simulate COP/ ndraws=1000 seed=0724 outuniform=unifdata;
run;
```

ユーザーが指定したコピュラに従う擬似乱数を生成することが目的であるので，PROC COPULA ステートメントにおいて，DATA=の指定は不要である．生成する擬似乱数の変数名は，VAR ステートメントに記載する．DEFINE ステートメントにおいて，クレイトンコピュラを表すキーワードである CLAYTON と指定し，あわせてパラメータ値をオプション THETA=で指定する．また，SIMULATE ステートメントにて使用するために，このコピュラの名前を“COP”とした．生成させる擬似乱数に関する指定は SIMULATE ステートメントで行う．SIMULATE ステートメントにおいて，DEFINE ステートメントにおいて定義したコピュラの名前を記述した後，オプション NDRAWS=で生成する擬似乱数の個数を，オプション SEED=で擬似乱数を生成するためのシードを，それぞれ指定する．生成した擬似乱数を保存するための SAS データセット名は，オプション OUTUNIFORM=で記述する.

プログラム 4.3 より，クレイトンコピュラに従う擬似乱数が生成されれば，あとは DATA ステップでの生成と同様に，得られた擬似乱数(u_1,u_2)を用いて($S_T^{-1}(u_1)$, $S_E^{-1}(u_2)$)と変換することにより，同時分布関数(4.3)に従うデータを算出すればよい．なお，2 つの生存時間を表す確率変数 T と E の依存構造として，正規コピュラ，スチューデント t コピュラ，ガンベルコピュラ，フランクコピュラを考えるのであれば，プログラム 4.3 において，DEFINE ステートメントの CLAYTON の代わりに各コピュラを表すキーワード及びパラメータを指定する．ただし，正規コピュラ，スチューデント t コピュラにおいては，分散共分散行列の各成分をもつ SAS データセットを予め用意した上で，DEFINE ステートメントのオプション CORR=にて指定する必要がある. プログラム 4.4 に, ケンドールの τ を 0.8 としたときの正規コピュラに従う擬似乱数を生成するための SAS プログラムを示した．2 変量正規コピュラにおいて，ケンドールの τ と相関係数 ρ とには

$$\tau = (2/\pi)\arcsin\rho \tag{4.11}$$

との関係が成り立つことを用い，DATA ステップで対応する分散共分散行列を用意した上で，COPULA プロシジャにより擬似乱数を生成している．

プログラム 4.4　COPULA プロシジャを用いた正規コピュラの擬似乱数生成

```
data corr;
   length U1 U2 rho pi 8.; format U1 U2 rho pi best.;
   keep U1 U2;
   pi=constant('pi'); rho=sin(0.8*pi/2);
   U1=1;    U2=rho;output;
   U1=rho; U2=1 ; output;
run;
proc copula;
   var U1-U2 ;
   define COP NORMAL(cor=corr) ;
   simulate COP/ ndraws=1000 seed=0724 outuniform=unifdata;
run;
```

5. おわりに

本稿では，SAS を用いてコピュラに従う乱数を生成させる方法について紹介した．コピュラを取り扱うプロシジャとして，SAS Ver.9.3 より SAS/ETS に COPULA プロシジャが搭載され，計 5 種類のコピュラに従う乱数を簡単に生成させることができる．また，当該プロシジャを使用できない環境下で乱数を生成させる方法として，DATA ステップを用いたマクロプログラムを示した．

COPULA プロシジャでは，解析対象となるデータにコピュラをあてはめたときのパラメータを推定することも可能である．現段階では Experimental の扱いであるが，完全にサポートされれば更に利用価値が高まるものと期待される．コピュラを用いたシミュレーション実験あるいはデータ解析を行う際に，本稿がその一助になれば望外の喜びである．

参考文献

[1] 岩崎学(2005). 統計的データ解析のための数値計算法入門. 朝倉書店.

[2] 魚住龍史，浜田知久馬(2013). RAND 関数による擬似乱数の生成. SAS ユーザー総会論文集 2013, 325-333.

[3] Clayton,D.G. (1978). A model for association in bivariate life tables and its application in epidemiological studies of familial tendency in chronic disease incidence. Biometrika 65, 141–151.

[4] Frank,M.J. (1979). On the simultaneous associativity of F(x,y) and x+y-F(x,y). Aequationes Mathematicae 19,

194–226.

[5] Hougaard, P. (1986). A class of multivariate failure time distributions. Biometrika 73, 671–678.

[6] Nelson, R.B. (2006). An Introduction to Copulas. New York: Springer.

[7] 塚原英敦(2012).「 接合分布関数(コピュラ)の理論と応用」. 北川源四郎, 竹村彰通(編). 21 世紀の統計科学 <Vol.III> 数理・計算の統計科学. 東京大学出版会増補 HP 版. http://park.itc.u-tokyo.ac.jp/atstat/jss75shunen/Vol3.pdf(最終閲覧日: 2014 年 6 月 1 日).

[8] 戸坂凡展, 吉羽要直(2005) .「 コピュラの金融実務での具体的な活用方法の解説」. 日本銀行金融研究所. http://www.imes.boj.or.jp/japanese/kinyu/2005/kk24-b2-3.pdf(最終閲覧日: 2014 年 6 月 1 日).

[9] Marshall,A.W. and Olkin,I. (1988). Families of Multivariate Distributions. Journal of the American Statistical Association 83, 834–841.

[10] SAS Institute Inc. (2011). SAS/ETS 9.3 User's Guide. The COPULA Procedure. Cary, NC: SAS Institute Inc.

[11] Ying,G. and Yuan,Y. (2009). Bayesian dose finding by jointly modeling toxicity and efficacy as time-to-event outcomes. Journal of the Royal Statistical Society 58, 719-736.

[12] Schemper,M.,Kaider,A.,Wakounig,S. and Heinze,G.. (2013). Estimating the correlation of bivariate failure times under censoring. Statistics in Medicine 32, 4781-4790.

[13] 四辻哲章(2010). 計算機シミュレーションのための確率分布乱数生成法. プレアデス出版.

[14] Weron,R. (1996). On the Chambers-Mallows-Stuck method for simulating skewed stable random variables. Statistics and Probability Letters 28, 165-171.

[15] Clizek,P.,Hardle,W.K. and Weron,R. (2005). Statistical Tools for Finance and Insurance. Springer.

[16] SAS CUSTOMER SUPPORT/TECHNICAL SUPPORT. 多次元正規分布に従う乱数列を生成する方法. http://www.sas.com/offices/asiapacific/japan/service/technical/faq/list/body/stat034.html(最終閲覧日: 2013 年 10 月 19 日).

連絡先

E-mail: s-yada@acronet.jp

付録 A　DATA ステップによるコピュラの乱数生成プログラム

```
/* クレイト ンコピュラに従う 乱数生成マクロ
 dim…………コピュラの次元
 theta...............コピュラのパラメータ
 seed…………シード
 ndraws.............生成し たい乱数の個数
 outuniform.....出力用データ セット */
%macro mclaycop(dim,theta,seed,ndraws,outuniform);
  data &outuniform.; keep m_u: SimNo;
   array col{&dim.}; array m_u{&dim.}; call streaminit(&seed.);
   do SimNo=1 to &ndraws.;
     m_r1=rand('gamma',1/&theta.) ;
     do m_j=1 to dim(m_u);
      col{m_j}=rand('uniform') ;
      m_u{m_j}=(1-log(col{m_j})/m_r1)**((-1)/&theta.);
     end;
     output;
    end;
  run;
%mend mclaycop;

/* フランクコピュラに従う 乱数生成マクロ
 dim…………コピュラの次元
 theta...............コピュラのパラメータ
 seed…………シード
 ndraws.............生成し たい乱数の個数
 outuniform.....出力用データ セット */
%macro mfrankcop(dim,theta,seed,ndraws,outuniform);
  data &outuniform.; keep m_u:;
   array col{&dim.}; array m_u{&dim.}; call streaminit(&seed.);
   do SimNo=1 to &ndraws.;
     m_su=rand('uniform') ;
     m_sv=rand('exponential') ;
     m_r1=int((-1)*m_sv/(log(1-(exp((-1)*&theta.))**m_su)))+1;
     do m_j=1 to dim(m_u);
       col{m_j}=rand('uniform');
```

```
        m_u{m_j}=( log( 1+ exp(log(col{m_j})/m_r1)*(exp((-1)*&theta.)-1) ) )*((-1)/&theta.);
      end;
      output;
    end;
  run;
%mend mfrankcop;

/* ガンベルコピュラに従う 乱数生成マクロ
  dim…………コピュラの次元
  theta...............コピュラのパラメータ
  seed…………シード
  ndraws.............生成したい乱数の個数
  outuniform.....出力用データセット */
%macro mgumbelcop(dim,theta,seed,ndraws,outuniform);
  data &outuniform.; keep m_u:;
    array col{&dim.}; array m_u{&dim.};
    call streaminit(&seed.);
    do SimNo=1 to &ndraws.;
      length m_xu m_v pi m_x m_s m_b m_r1 8.; format m_xu m_v pi m_x m_s m_b m_r1 best.;
      pi=constant('pi');
      m_gamma=(cos(pi/(2*&theta.)))**&theta.; m_alpha=round(1/&theta.,1e-12); m_beta =1;
      if    m_alpha^=1 then do;
        m_xu=rand('uniform')*pi-pi/2;
        m_v =rand('exponential');
        m_b =atan(m_beta*tan(pi*m_alpha/2));
        m_s =(1+(m_beta*tan(pi*m_alpha/2))**2)**(1/(2*m_alpha));
        m_x =m_s*(sin(m_alpha*m_xu+m_b)/(cos(m_xu)**(1/m_alpha)))
        *( (cos(m_xu-m_alpha*m_xu-m_b)/m_v)**((1-m_alpha)/m_alpha) );
        m_r1=round(m_gamma*m_x,1e-12);
      end;
      else if m_alpha=1 then do;
        m_xu=rand('uniform')*pi-pi/2;
        m_v =rand('exponential');
        m_x =(2/pi)*((pi/2+m_beta*m_xu)*tan(m_xu)-m_beta*log(((pi/2)*m_v*cos(m_xu))/(pi/2+m_beta*m_xu)));
        m_r1=round(m_gamma*m_x+(2/pi)*m_beta*m_gamma*log(m_gamma),1e-12);
      end;
```

```
        do m_j=1 to dim(m_u);
          col{m_j}=rand('uniform') ;
          m_u{m_j}=exp( (-1)*(-log(col{m_j})/m_r1)**(1/&theta.) );
        end;
        output;
      end;
    run;
%mend mgumbelcop;

/*  正規コピュラに従う乱数生成マクロ
  dim................コピュラの次元
  cor...................相関行列(SAS データセット名).  変数名は colX でないとエラーになる.
  seed................シード
  ndraws............生成したい乱数の個数
  outuniform.....出力用データセット */
%macro mnormcop(dim,cor,seed,ndraws,outuniform);
    data m_wrk;
      call streaminit(&seed.);
      _TYPE_="SCORE"; _MODEL_="col"; mean=1; array col{&dim.};
      do num=1 to &ndraws.;do i=1 to dim(col);col{i}=rand("NORMAL");end;output;end;
      drop i;
    run;
    data m_cov ;set &cor.;length mean 8.;format mean best.;mean=0;row=_n_;run;
    ods listing close;
    ods output CHOLG=m_cholesky;
    proc mixed data=m_cov;
      class row mean; parms /noiter;model mean=; random row*mean/type=UN gdata=m_cov GC;
    run;
    ods listing;
    proc score data=m_cholesky score=m_wrk out=m_mnorm(keep=col num); by num;var mean col:; run;
    proc transpose data=m_mnorm OUT=m_mnorm(DROP=_NAME_);by num;run;
    data &outuniform.;set m_mnorm;
      array col{&dim.} ; array m_u{&dim.};
      do i=1 to &dim.;do i=1 to dim(m_u);m_u{i}=CDF('NORMAL',col{i});end;end;
      drop i col:;
    run;
```

```
  proc datasets library=work nowarn nolist nodetails; delete m_wrk m_cholesky m_cov m_mnorm; run; quit;
%mend mnormcop;

/* t コピュラに従う 乱数生成マクロ
 dim.................コピュラの次元
 cor....................相関行列(SAS データ セット 名). 変数名は colX でないとエラーになる.
 df.....................t コピュラの自由度
 seed.................シード
 ndraws............生成し たい乱数の個数
 outuniform.....出力用データ セット */
 %macro mtcop(dim,cor,df,seed,ndraws,outuniform);
   data m_wrk;
     call streaminit(&seed.);
     _TYPE_="SCORE"; _MODEL_="col"; mean=1; array col{&dim.};
     do num=1 to &ndraws.;do i=1 to dim(col);col{i}=rand("NORMAL");end;output;end;
     drop i;
   run;
   data m_cov ;set &cor.;length mean 8.;format mean best.;mean=0;row=_n_;run;
   ods listing close; ods output CHOLG=m_cholesky;
   proc mixed data=m_cov;
    class row mean; parms /noiter;model mean=; random row*mean/type=UN gdata=m_cov GC;
   run; ods listing;
   proc score data=m_cholesky score=m_wrk out=m_mnorm(keep=col num); by num;var mean col:; run;
   proc transpose data=m_mnorm OUT=m_mnorm(DROP=_NAME_);by num;run;
   data m_wrk ;do num=1 to &ndraws.;output;end;run;
   data m_wrk;set m_wrk;length m_s 8.;format m_s best.; call streaminit(&seed.+num);m_s=rand('chisquare',&df.);run;
   data &outuniform.;
    merge m_mnorm m_wrk;by num;
    array col{&dim.} ; array m_u{&dim.};
    do i=1 to &dim.;do i=1 to dim(m_u);m_u{i}=CDF('T',col{i}*sqrt(&df./m_s),&df.);end;end;
    drop i col:;
   run;
   proc datasets library=work nowarn nolist nodetails; delete m_wrk m_cholesky m_cov m_mnorm; run; quit;
 %mend mtcop;
```