上側 100α% 点計算ライブラリ
Mathematica 版

杉浦 洋

2018/04/05

1　基本的な使い方

ライブラリ・プログラムはノートブック `SincStatistics.nb` として提供される．開発は Mathematica ver.11.1.1 上で行った．旧い version で正常に動作するかどうかは，未調査である．

ダウンロードした zip ファイルを解凍すると，ノートブック・ファイル `SincStatistics.nb` ができる．これは，表題セルと関数セルの 2 つのセルからなる．2 番目の関数セルを実行すれば，それに含まれる全てのライブラリ関数が使用可能となる．主なライブラリ関数については次節で解説する．

2　ライブラリの内容

『多重比較法の理論と数値計算』の第 2 章から第 5 章の統計解析に用いられる分布の上側 100α% 点を求める Mathematica 関数を作成した．それを用いて，閉検定手順で用いられる分布の，上側 100$\alpha(M,\ell)$% 点 ($k \geqq M \geqq 2,\ 2 \leqq \ell \leqq M, \ell \neq M-1$) を要素とする T1 型 2 次元配列 (表 1)，上側 100α% 点 ($1 \leqq \ell \leqq k-1$) を要素とする T2 型 1 次元配列 (表 2)，そして上側 100α% 点 ($2 \leqq \ell \leqq k$) を要素とする T3 型 1 次元配列 (表 3) を出力する Mathematica 関数を作成した．ただし，$\alpha(M,\ell) \equiv 1-(1-\alpha)^{(\ell/M)}$ である．配列は表関数として実現する．表関数とは整数引数の関数である．また，出力された表関数を定められた表形式でコンピュータ・ディスプレイに出力する簡単な関数を作成した．

計算に階層確率 (Level Probability) を用いる分布関数がある．ライブラリには階層確率を計算する関数が附属している．

表 1　T1 型 2 次元配列 : { 上側 100$\alpha(M,\ell)$% 点 }　($k=8$ のとき)

$M \setminus \ell$	2	3	4	5	6	7	8
8	5.144	5.781	6.046	6.172	6.233	◇	6.256
7	4.902	5.519	5.770	5.887	◇	5.956	
6	4.626	5.219	5.455	◇	5.604		
5	4.303	4.868	◇	5.177			
4	3.914	◇	4.637				
3	◇	3.905					
2	2.756						

表 2　T2 型 1 次元配列 : { 上側 $100\alpha\%$ 点 }　($k = 8$ のとき)

$k\backslash\ \ell$	7	6	5	4	3	2	1
8	6.256	6.241	6.233	6.172	6.046	5.781	5.144

表 3　T3 型 1 次元配列 : { 上側 $100\alpha\%$ 点 }　($k = 8$ のとき)

$k\backslash\ \ell$	8	7	6	5	4	3	2
8	6.256	6.241	6.233	6.172	6.046	5.781	5.144

2.1　上側 $100\alpha\%$ 点を求める Mathematica 関数

以下の表 4 に上側 $100\alpha\%$ 点を求める Mathematica 関数を示す.

表 4　上側 $100\alpha\%$ 点を計算する Mathematica 関数

ページ	上側 $100\alpha\%$ 点	Mathematica 関数
33	$ta(k, m; \alpha)$	`taFun[k,m,α]`
34	$a(k; \alpha)$	`aFun[k,α]`
78	$tb_1(k, \boldsymbol{n}; \alpha)$	`tb1Fun[k,ns,α]`
78	$tb_2(k, \boldsymbol{n}; \alpha)$	`tb2Fun[k,ns,α]`
81	$b_1(k, \boldsymbol{\lambda}; \alpha)$	`b1Fun[k,λs,α]`
81	$b_2(k, \boldsymbol{\lambda}; \alpha)$	`b2Fun[k,λs,α]`
126	$\bar{b}^2(k, \boldsymbol{\lambda}, m; \alpha)$	`bbar2Fun[k,λs,m,α]`
129	$\bar{c}^2(k, \boldsymbol{\lambda}; \alpha)$	`cbar2Fun[k,λs,α]`
133	$td_1(k, m; \alpha)$	`td1Fun[k,m,α]`
135	$d_1(k; \alpha)$	`d1Fun[k,α]`
156	$td_2^*(k, m; \alpha)$	`td2Fun[k,m,α]`
158	$d_2^*(k; \alpha)$	`d2Fun[k,α]`

この表で「ページ」は,『多重比較法の理論と数値計算』において，その関数が定義されているページを示す.「上側 $100\alpha\%$ 点」の母数 $\boldsymbol{n}$, $\boldsymbol{\lambda}$ は

$$\boldsymbol{n} = (n_1, n_2, \cdots, n_k),$$

$$\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \cdots, \lambda_k) \equiv (n_1/n, n_2/n, \cdots, n_k/n),\ n = \sum_{i=1}^{k} n_i$$

である.

Mathematica 関数の入力引数と母数の対応は

$$\texttt{k} = k,\quad \texttt{l} = \ell,\quad \texttt{m} = m,\quad \texttt{ns} = \boldsymbol{n},\quad \alpha = \alpha,\quad \lambda\texttt{s} = \boldsymbol{\lambda} \tag{1}$$

である. m は通常 $m = n - k$ として用いられるが，ここでは独立した引数である.

Mathematica 関数の出力は倍精度の上側 $100\alpha\%$ 点である．例えば，母数 $k=5,\ m=100$ に対する，表 4 先頭のテューキー・クレーマーの上側 1% 点 $t=ta(5,100;0.01)$ は

```
k=5; m=100; α=0.01
t = taFun[k,m,α];
```

で計算できる．

Mathematica 関数の入力引数 `k`,`m`,`al` には以下の制限を設ける．`k`,`l` の制限は計算速度向上のためである．また，α の制限は精度保証のためである．この制限内で，Mathematica 関数の出力は小数点以下 6 桁まで正しいことが保証される．

$$
\begin{aligned}
2 \leqq \mathtt{k} \leqq 20, &\quad 2 \leqq \mathtt{l} \leqq 20, \\
\mathtt{m} \geqq 1 &\ \text{または}\ \mathtt{m}=\infty, \\
10^{-4} \leqq \alpha &< \frac{1}{2}
\end{aligned} \tag{2}
$$

入力引数 $\mathtt{m}=\infty$ とすると，対応する漸近分布に関する計算値が出力される．すなわち，$\mathtt{m}=\infty$ のとき

$$
\begin{aligned}
\mathtt{taFun}[\mathtt{k},\mathtt{m},\alpha] &= ta(k,\infty;\alpha) &&= a(t,k), \\
\mathtt{tb1Fun}[\mathtt{k},\mathtt{ns},\mathtt{m},\alpha] &= tb_1(k,\boldsymbol{n},\infty;\alpha) &&= b_1(k,\boldsymbol{\lambda};\alpha), \\
\mathtt{tb2Fun}[\mathtt{k},\mathtt{ns},\mathtt{m},\alpha] &= tb_2(k,\boldsymbol{n},\infty;\alpha) &&= b_2(k,\boldsymbol{\lambda};\alpha), \\
\mathtt{bbar2Fun}[\mathtt{k},\lambda\mathtt{s},\mathtt{m},\alpha] &= \bar{b}^2(k,\boldsymbol{\lambda},\infty;\alpha) &&= \bar{c}^2(k,\boldsymbol{\lambda};\alpha), \\
\mathtt{td1}[\mathtt{k},\mathtt{m},\alpha] &= td_1(k,\infty;\alpha) &&= d_1(k;\alpha), \\
\mathtt{td2}[\mathtt{k},\mathtt{m},\alpha] &= td_2^*(k,\infty;\alpha) &&= d_2^*(k;\alpha)
\end{aligned} \tag{3}
$$

である．最左辺の Mathematica 関数と最右辺に対応する Mathematica 関数とは，出力が一致する．

表 4 の 7,8 番目の Mathematica 関数 `bbar2Fun`，`cbar2Fun` は内部で階層確率を計算するため，2^{k+1} 個の倍精度実数を格納するためのワーキングメモリー約 16×2^k バイトを消費する．それに見合うメモリ容量を確保する必要がある．また，階層確率の計算には時間が掛かる．関数 `bbar2Fun`，`cbar2Fun` を繰り返し計算する際には，一度計算した階層確率を使える限り使い回して，再計算を抑止する必要がある．そのアルゴリズムについては，4.2 節をご覧頂きたい．

2.2 上側 $100\alpha(M,\ell)\%$ 点の 2 次元配列を出力する Mathematica 関数

以下の表 5 に，閉検定手順で用いられる分布の上側 $100\alpha(M,\ell)\%$ 点 $(k\geqq M\geqq 2,\ 2\leqq \ell\leqq M, \ell\neq M-1)$ を要素とする T1 型 2 次元配列 (表 1) を作る Mathematica 関数を示す．ただし，$\alpha(M,\ell)\equiv 1-(1-\alpha)^{(\ell/M)}$ である．

配列は，ここでは表関数として実現される．表関数は整数引数の関数のことである．

表 5　上側 $100\alpha(M,\ell)\%$ 点の T1 型 2 次元配列を出力する Mathematica 関数

節番号	表関数値 $[M,\ell]$	Mathematica 関数
2.1.3	$ta(\ell,m;\alpha(M,\ell))$	`taTab[k,m,α]`
2.1.5	$a(\ell;\alpha(M,\ell))$	`aTab[k,α]`
2.1.3	$F_m^{\ell-1}(\alpha(M,\ell))$	`FTab[k,m,α]`
2.3.3	$\chi^2_{\ell-1}(\alpha(M,\ell))$	`chi2Tab[k,α]`
5.3.3	$td_1(\ell,m;\alpha(M,\ell))$	`td1Tab[k,m,α]`
5.3.3	$d_1(\ell;\alpha(M,\ell))$	`d1Tab[k,α]`
5.3.4	$td_2^*(\ell,m;\alpha(M,\ell))$	`td2Tab[k,m,α]`
5.4.3	$d_2^*(\ell;\alpha(M,\ell))$	`d2Tab[k,α]`
5.6.1	$\bar{b}_1^{2*}(\ell,m;\alpha(M,\ell))$	`bbarTab[k,m,α]`
5.6.1	$\bar{c}_1^{2*}(\ell;\alpha(M,\ell))$	`cbarTab[k,α]`
5.6.1	$t(m;\alpha(M,\ell))$	`tTab[k,m,α]`
5.6.1	$z(\alpha(M,\ell))$	`zTab[k,α]`

この表で「節番号」は，『多重比較法の理論と数値計算』において，その統計量を用いる閉検定手順を解説している節の番号である．Mathematica 関数の入力引数と母数の対応は (1) に従う．出力は T1 型の 2 変数表関数である．

たとえば，表 5の $ta(\ell,m;\alpha(M,\ell))$ について，母数 $k=5,\ m=100$ に対する水準 1% の閉検定手順に用いられる T1 型 2 次元配列を，表関数 `tab` として実現するプログラムは，

```
k = 5; m = 100; α = 0.01;
tab = taTab[k,m,α];
```

である．表関数 `tab` の出力は

$$\texttt{tab}[M,\ell] = ta(\ell,m;\alpha(M,\ell)) \quad (2 \leqq M \leqq k, 2 \leqq \ell \leqq M, \ell \neq M-1)$$

である．

入力引数 $\mathtt{m}=\infty$ とすると，対応する漸近分布に関する表関数が作られる．すなわち，$\mathtt{m}=\infty$ のとき

$$\begin{aligned} \texttt{taTab}[\mathtt{k},\mathtt{m},\alpha] &= \texttt{aTab}[\mathtt{k},\alpha] \\ \texttt{td1}[\mathtt{k},\mathtt{m},\alpha] &= \texttt{d1}[\mathtt{k},\alpha], \\ \texttt{td2}[\mathtt{k},\mathtt{m},\alpha] &= \texttt{d2}[\mathtt{k},\alpha], \\ \texttt{bbar2Fun}[\mathtt{k},\mathtt{m},\alpha] &= \texttt{cbar2Fun}[\mathtt{k},\alpha] \end{aligned} \tag{4}$$

である．

Mathematica 関数の入力引数 `k,m,al` は (2) で制限される．この制限内で Mathematica 関数の出力は小数点以下 6 桁まで正しいことが保証される．

2.3　上側 $100\alpha\%$ 点の 1 次元配列を出力する Mathematica 関数

以下の表 6 に，閉検定手順で用いられる分布の，上側 $100\alpha\%$ 点 $(1 \leqq \ell \leqq k-1)$ を要素とする T2 型 1 次元配列 (表 2)，及び 上側 $100\alpha\%$ 点 $(2 \leqq \ell \leqq k)$ を要素とする T3 型 1 次元配列 (表 3) を出力する Mathematica 関数を示す．

配列は，ここでは表関数として実現される．表関数は整数引数の関数のことである．

表 6　上側 $100\alpha\%$ 点の T2 型，T3 型の 1 次元配列を出力する Mathematica 関数

節番号	表関数値 $[\ell]$	Mathematica 関数
3.3.1	$tb_1^*(\ell,m,n_2/n_1;\alpha)$	`tb1Tab[k,n1,n2,α]`
3.3.1	$tb_2^*(\ell,m,n_2/n_1;\alpha)$	`tb2Tab[k,n1,n2,α]`
3.3.2	$b_1^*(\ell,m,n_2/n_1;\alpha)$	`b1Tab[k,n1,n2,α]`
3.3.2	$b_2^*(\ell,m,n_2/n_1;\alpha)$	`b2Tab[k,n1,n2,α]`
節番号	表関数値 $[\ell]$	Mathematica 関数
5.5.1	$td_3(\ell,m,1;\alpha)$	`td3Tab[k,m,α]`
5.5.1	$d_3(\ell,1;\alpha)$	`d3Tab[k,α]`
5.6.2	$\bar{b}_3^2(\ell,\boldsymbol{\lambda},m;\alpha)$	`bbar32Tab[k,λs,m,α]`
5.6.2	$\bar{c}_3^2(\ell,\boldsymbol{\lambda};\alpha)$	`cbar32Tab[k,λs,α]`

この表で「節番号」は，『多重比較法の理論と数値計算』において，その統計量を用いる閉検定手順を解説している節の番号である．Mathematica 関数の入力引数と母数の対応は (1) に従う．出力は T1 型，あるいは T2 型の 1 変数表関数である．

たとえば，表 6 の $tb_1^*(\ell,m,n_2/n_1;\alpha)$ について，母数 $k=10$, $n_1=23$, $n_2=45$ に対する水準 1% の逐次棄却型検定に用いられる T2 型 1 次元配列を表関数 `tab` として実現するプログラムは，

```
k = 10; n1 = 23; n2 = 45; α = 0.01;
tab = tb1Tab[k,n1,n2,α];
```

である．表関数 `tab` の出力は

$$\texttt{tab}[\ell] = tb_1^*(\ell,m,n_2/n_1;\alpha) \quad (1 \leqq \ell \leqq k-1)$$

である．$tb_1^*(\ell,m,n_2/n_1;\alpha(M,\ell))$, $td_2^*(\ell,m,n_2/n_1;\alpha(M,\ell))$ は $m=n_1+(k-1)n_2-k$ として計算される．したがって，対応する Mathematica 関数 `tb1Tab`，`tb2Tab` は入力引数 `m` を持たない．

また，$td_3(\ell,m,1;\alpha)$ について，母数 $k=10$, $m=230$ に対する水準 5% のウイリアムス法に用いられる T3 型 1 次元配列を表関数 `tab` として実現するプログラムは，

```
k=10; m=230; α=0.05;
tab = td3Tab(k,m,α);
```

である．表関数 `tab` の出力は

$$\texttt{tab}[\ell] = td_3(\ell,m,1;\alpha) \quad (2 \leqq \ell \leqq k)$$

である．この関数は現在，$n_2/n_1=1$ にしか対応出来ない．

入力引数 $\texttt{m}=\infty$ とすると，対応する漸近分布に関する配列が出力される．すなわち，$\texttt{m}=-1$ のとき

$$\begin{aligned} \texttt{td3Tab}[\texttt{k},\texttt{m},\alpha] &= \texttt{d3Tab}[\texttt{k},\alpha], \\ \texttt{bbar32Tab}[\texttt{k},\lambda\texttt{s},\texttt{m},\alpha] &= \texttt{cbar32Tab}[\texttt{k},\lambda\texttt{s},\alpha] \end{aligned} \tag{5}$$

である．

Mathematica 関数の入力引数 $\texttt{k},\texttt{m},\texttt{al}$ は (2) で制限される．この制限内で Mathematica 関数の出力は小数点以下 6 桁まで正しいことが保証される．

表 6 の 7,8 番目の Mathematica 関数 `bbar32Tab`，`cbar32Tab` は内部で階層確率を計算するため，2^{k+1} 個の倍精度実数を格納するためのワーキングメモリー約 16×2^k バイトを消費する．それに見合うメモリ容量を確保する必要がある．また，階層確率の計算には時間が掛かる．関数 `bbar32Tab`，`cbar32Tab` を計算する際には，一度計算した階層確率を使える限り使い回して，再計算を抑止する必要がある．そのアルゴリズムについては，4.2 節をご覧頂きたい．

3 表出力プログラム

表関数生成関数が出力した表関数 `tab` とそのサイズ `k` 入力し，指定された形式の表をコンピュータ・ディスプレイに出力する．表の型，T1, T2, T3 に合わせて，次の 3 種類から選択する．

`PrintTabT1[k,tab]`,

`PrintTabT2[k,tab]`,

`PrintTabT3[k,tab]`.

例えば，サイズ $k = 10$ の T1 型表関数 tab に対し，`PrintTabT1[10,tab]` で次の表示が得られる．

M\l	2	3	4	5	6	7	8	9	10
10	6.599	8.364	9.804	11.113	12.349	13.536	14.689	◇	16.919
9	6.412	8.155	9.576	10.868	12.087	13.259	◇	15.507	
8	6.205	7.921	9.320	10.592	11.793	◇	14.067		
7	5.970	7.657	9.031	10.279	◇	12.592			
6	5.701	7.352	8.696	◇	11.070				
5	5.385	6.993	◇	9.488					
4	5.002	◇	7.815						
3	◇	5.991							
2	3.841								

4 階層確率の計算

階層確率の定義については，『多重比較法の理論と数値計算』の 5.2 節をご覧頂きたい．また，アルゴリズムについては，7.4 節に詳しい解説がある．

4.1 階層確立計算関数

与えられた整数 $k \geqq 1$ と重みベクトル $\boldsymbol{w} = (w_1, w_2, \cdots, w_k) \in \mathbb{R}_+^k$ に対し，階層確率の Mathematica の表関数

$$P[L,m,s] = P(L,m,(w_s, w_{s+1}, \cdots, w_{s+m-1})),$$
$$1 \leqq L \leqq m \leqq k,\ 1 \leqq s \leqq k-m+1$$

を定義する関数

`MakeTableP(k,ws)`

を用意した．整数 k $=k \geqq 1$ と正数のリスト ws=$\{w_1, w_2, \cdots, w_k\}$ を入力し，

```
P = MakeTableP[k,ws];
```

により，Mathematica 関数 P が定義される，表関数 P の出力は

$$\texttt{P[L,m,s]=} P(L, m, (w_s, w_{s+1}, \cdots, w_{s+m-1})), \quad 1 \leqq L \leqq m \leqq k,\ 1 \leqq s \leqq k - m + 1$$

である．制約 $1 \leqq$ L $\leqq$ m $\leqq$ k, $1 \leqq$ s $\leqq$ k $-$ m $+ 1$ を満たさない引数 L, m, s に対しては，未評価の式 P[L,m,s] が出力される．

計算のために要素数 2^{k+1} の倍精度作業配列 Q が生成されるので，それに見合うメモリ容量 (16×2^k バイト) が必要である．また，計算時間も 2^{k+1} に比例する．

$k = 10$ の実行時間 s_{10} 秒を測定しておけば，任意の k における実行時間 $\cong 2^{k-10}s_{10}$ 秒が予想できる．筆者のパソコンでは (CPU 3.2GHz) では $s_{10} = 3$ 程度であった．

4.2 表関数 P の再利用

サイズ k の大きい表関数 P を作るためには，多大のメモリと計算時間を消費する．一度作った表関数 P は可能な限り再利用すべきである．

そのために，表関数を使用する関数 bbar2Fun，cbar2Fun，bbar32Tab，cbar32Tab などでは，次の様な制御を行う．最初にこれらの関数が呼ばれるとき，MakeTableP[k,λs] を

```
LevelProbability`P = MakeTableP[k,λs];
LevelProbability`k = k;
LevelProbability`λs =λs;
```

のように使う．以下，簡単のため 3 式の左辺を P0，k0，λs0 と略記する．

直接表関数 P を作らず，等価な関数を P0 として作る．また作ったときのパラメタを k0，λs0 に記録する．P は常に P0 をコピーして使う．

上記の関数 bbar2Fun，cbar2Fun，bbar32Tab，cbar32Tab などが呼ばれるのが 2 度目以降のとき，k $\leq$ k0 かつ λs = Take[λs0,{k}] なら P0 を再計算しない．そうでないときは，P0 を新しいパラメタで再計算し，k0，λs0 を更新する．ここで，Take[λs0,{k}] はリスト λs0 の最初の k 個の要素からなる部分リストである．

SincStatistics.nb の関数セルを実行すると，k0 = 0 にリセットされる．これにより，bbar2Fun，cbar2Fun，bbar32Tab，cbar32Tab などが初回に呼ばれたときには，必ず表関数 P0 が作られる．また，二度目以降に呼ばれたとき，パラメタ k，λs が再利用可能な P を生成すると判断されたときは，その再計算が抑止される．

＜注意＞ 正常なパラメタで MakeTableP が呼ばれないとき，このシステムが働かなくなる恐れがある．例えば，k に自然数を代入せずにシンボルのまま MakeTableP[k,λs] を呼ぶと，k0 はシンボルにセットされてしまい，以後この制御システム内の IF 関数が評価不能になり，制御が失われる．

そのときは，SincStatistics.nb の関数セルを実行する．あるいは，

```
LevelProbability`k = k;
```

を実行して，この制御システムをリセットする．