

R の HGM パッケージ (設計と実装)
高山信毅 (神戸大学)

参考文献等 [hgm OpenXM search](#).

Q. 何をやるソフトウェア?

A. パラメータ付きの積分が満す線形微分方程式を数値解析して、積分の値を計算する Holonomic Gradient Method (HGM) のソフトウェア.

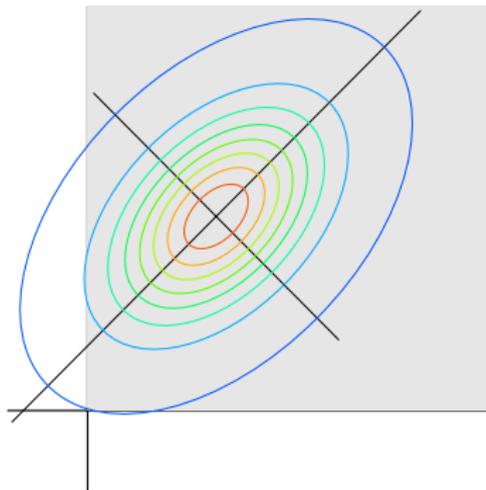
著者: 高山信毅, 小山民雄, 清智也, 小原功任, 中山洋将.

インストールの方法. `install.packages('hgm')`;
Version 1.16 (つい最近 version up)

何が計算できるの？

- ① Wishart 分布に従う正定値対称行列の最大固有値の確率. $\text{matrix } {}_1F_1$ の計算. `hgm.pwishart`
- ② Wishart 分布に従う正定値対称行列二つの比の最大固有値の確率. $\text{matrix } {}_2F_1$ の計算. `hgm.p2wishart`
- ③ A, S を行列とするとき, $\exp(\text{tr}(AS))$ を $A \in SO(3)$ で積分する. Fisher 積分. `hgm.ncso3`

- ① 多変数の正規分布を第一象限で積分する. `hgm.ncorthant`
- ② 平均が原点の多変数の正規分布を超球表面で積分する. `hgm.ncBingham`
- ③ 準備中 (小山): 多変数の正規分布を超球内部で積分する.
- ④ 準備中 (小山): 多変数の正規分布を単体で積分する.



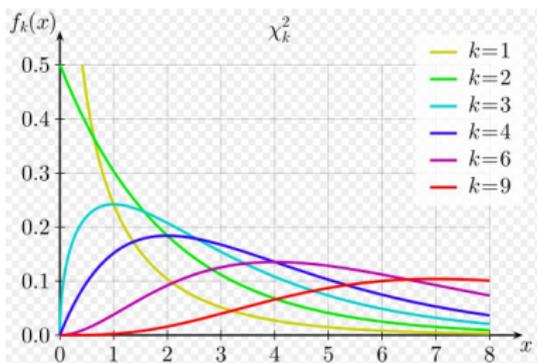
Wishart 分布

X : $n \times m$ 行列 ($n \geq m$) . 行は m 次元の正規分布に従う random variable (vector) (独立, 同分布, 平均は 0, covariance matrix は Σ).
 $m \times m$ 値 random variable (行列) $X^T X$ に対応する確率分布が自由度 n の Wishart 分布. 特に $m = 1, \Sigma = 1$ の時は χ^2 -分布である:

$$H(t)t^{\frac{n}{2}-1} \exp\left(-\frac{t}{2}\right) / Z(k), \quad Z(k) = 2^{n/2} \Gamma(n/2)$$

$H(t)$ は $t > 0$ の特性関数 (Heaviside 関数).

χ^2 -分布密度関数 (Wikipedia より, $k = n$ (自由度))



Wishart 分布の第一固有値の分布

$\lambda_1(W)$ を行列 W の最大固有値とする.

問題: W が Wishart 分布に従うとき

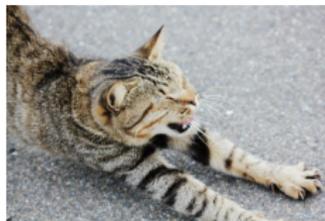
$$P(\lambda_1(W) < q)$$

を計算せよ.

答え (Constantine 1963, James 1964, Muirhead 1970): Wishart 分布の積分で上記の確率は書ける. 積分は matrix hypergeometric function ${}_1F_1$ で表現できて, パラメータについて線形微分方程式系を満たす.

hgm (橋口, 沼田, 高山, 竹村, (2013)) では,

- ① 小さい q (q_0) で上記の値を ${}_1F_1$ の級数展開から計算 (Koev-Edelman 2006).
- ② q_0 での値を微分方程式を数値解析することにより大きい q への**ばす**.



Wishart 分布の第一固有値, 行列引数超幾何系 ${}_1F_1$ (詳細)

自由度 n , $m \times m$ 共分散行列 Σ できまる Wishart 分布に従う $m \times m$ 行列の最大固有値 ℓ_1 が x より小さい確率は C を定数として

$$P[\ell_1 < x] = C \exp\left(-\frac{x}{2} \text{Tr} \Sigma^{-1}\right) x^{\frac{1}{2}nm} {}_1F_1\left(\frac{m+1}{2}; \frac{n+m+1}{2}; \frac{x}{2} \Sigma^{-1}\right) \quad (1)$$

と書けることが知られている. ${}_1F_1$ は行列引数の超幾何関数で次の積分で定義される.

$${}_1F_1(a; c; Y) = G \int_{0 < X < I_m} \exp(\text{Tr} XY) |X|^{a-(m+1)/2} |I_m - X|^{c-a-(m+1)/2} dX, \quad (2)$$

$G = \frac{\Gamma_m(c)}{\Gamma_m(a)\Gamma_m(c-a)}$. なお $0 < X < I_m$ は行列 X および $I_m - X$ が正定値対称行列であることを意味する. $dX = \prod_{i \leq j} dx_{ij}$ は X の上三角成分に渡る積分を意味する. また

$$\Gamma_m(a) = \pi^{\frac{1}{4}m(m-1)} \prod_{i=1}^m \Gamma\left(a - \frac{i-1}{2}\right).$$

Wishart 分布の第一固有値, 行列引数超幾何系 ${}_1F_1$ (詳細)

${}_1F_1(a; c; Y)$ の Y は対角成分が y_1, \dots, y_m である対角行列であるとする. Muirhead (1970) は ${}_1F_1$ が満たす次の微分方程式を導いた.

$$\left[y_i \partial_i^2 + p_i(y) \partial_i - \frac{1}{2} \sum_{j=1, j \neq i}^m \frac{y_j}{y_i - y_j} \partial_j - a \right] F = 0,$$
$$p_i(y) = \left\{ c - \frac{m-1}{2} - y_i + \frac{1}{2} \sum_{j=1, j \neq i}^m \frac{y_i}{y_i - y_j} \right\}$$
$$(i = 1, \dots, m)$$

hgm パッケージではどう計算?

```
library(hgm)
hgm.pwishart(m=3,n=5,beta=c(1,2,3),q=3)
[1] 3.0105152 0.5209449 0.2647107 0.3397387
      0.1702736 0.3881708 0.1959106
[8] 0.2524648 0.1256162
```

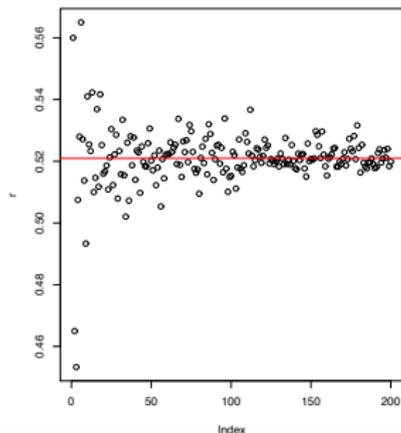
$\text{beta} = \Sigma^{-1}/2.$

なぜ hgm?

- ① そこに関数があるから.
- ② hgm の高精度高速計算によって乱数によるシミュレーションを評価できる.
- ③ 乱数によるシミュレーションでできないことが短時間で出来ることがある.

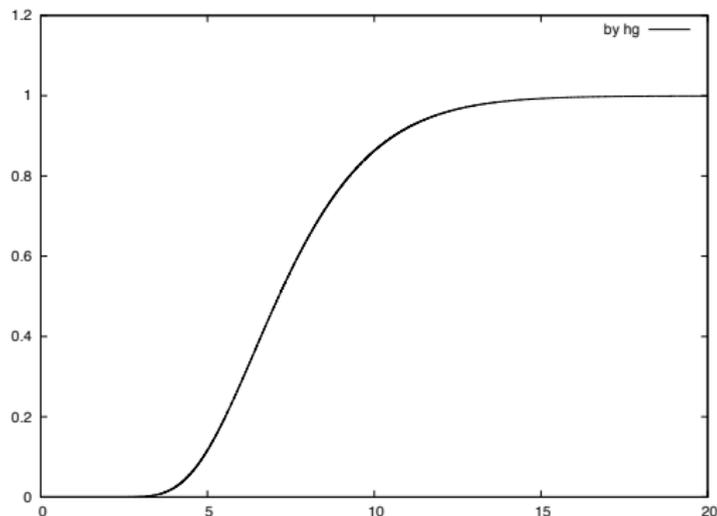
シミュレーションとの比較

```
rpwishart  
<-function(lambda=3,sigma=diag(c(1/2,1/4,1/6)),n=5,try=100) {  
  count <- 0;  
  r<-rWishart(try,df=n,Sigma=sigma);  
  for (k in seq(1,try)) {  
    ell1 <- max(eigen(r[, ,k])$values);  
    if (ell1 < lambda) count<-count+1; }  
  return(count/try) }  
}
```



$P(\lambda_1(W) < 3)$ (最大固有値が
3 以下である頻度/総試行回数)
の値. 試行 20,000 回までのシ
ミュレーション値.

累積分布関数 $P[l_1 < x]$, $m = 10, n = 12$,
 $\Sigma^{-1}/2 = \text{diag}(1, 2, \dots, 10)$



出典: 橋口, 沼田, 高山, 竹村, Holonomic gradient method for the distribution function of the largest root of a Wishart matrix, *Journal of Multivariate Analysis*, 117, (2013) 296-312.

Q. 自由度を大きくすると default 値ではうまく行かない. どうすればいいの?

```
plot(hgm.pwishart(m=3,beta=c(1,2,3),n=100,q0=0.3,q=10,  
err=c(1e-30,1e-10),autoplot=1))
```

```
%%Warning, abserr seems not to be small enough, abserr=1e-30, y[0]=3.43846e-30.  
Increasing the starting point (q0 or X0g(standalone case)) may  
or making abserr (err[1] or abserror(standalone case)) smaller will help,  
e.g., err=c(1e-20,1e-10)  
Error: x0 < x must hold.  
スタート点 q0 での値が小さすぎる. q0 が自動的に増えるのだが q=10 を越した.  
hgm.pwishart(n = 100, q0 = 0.3, q = 10, err = c(1e-30, 1e-10), エラー:  
mh_exit(-30) is called.
```

q を増やす. (q0 が自動で増えた場合.) 参考 verbose=1

```
plot(hgm.pwishart(n=100,q0=0.3,q=20,err=c(1e-30,1e-10),autoplot=1))
```

```
%%Warning, abserr seems not to be small enough, abserr=1e-30, y[0]=3.43846e-30.  
Increasing the starting point (q0 or X0g(standalone case)) may  
or making abserr (err[1] or abserror(standalone case)) smaller will help,  
e.g., err=c(1e-20,1e-10)
```

err を減らす

```
> plot(hgm.pwishart(n=100,q0=0.3,q=20,err=c(1e-40,1e-10),autoplot=1))  
> plot(hgm.pwishart(n=100,q0=0.3,q=200,err=c(1e-40,1e-10),autoplot=1))
```

Q. 自由度を大きくすると default 値ではうまく行かない. どうすればいいの? その2

q0 と err を上手に調整.

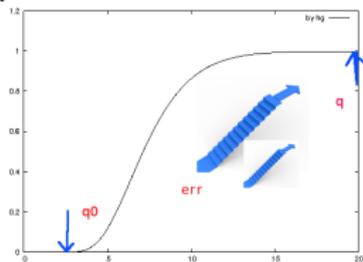
```
> plot(hgm.pwishart(m=5,n=20,beta=c(1,2,3,4,5),autoplot=1,q=100))  
%%Warning, abserr seems not to be small enough, abserr=1e-40,  
y[0]=1.4645e-53.
```

初期値が $1e-53$ なので, abs err を小さくしてみる.

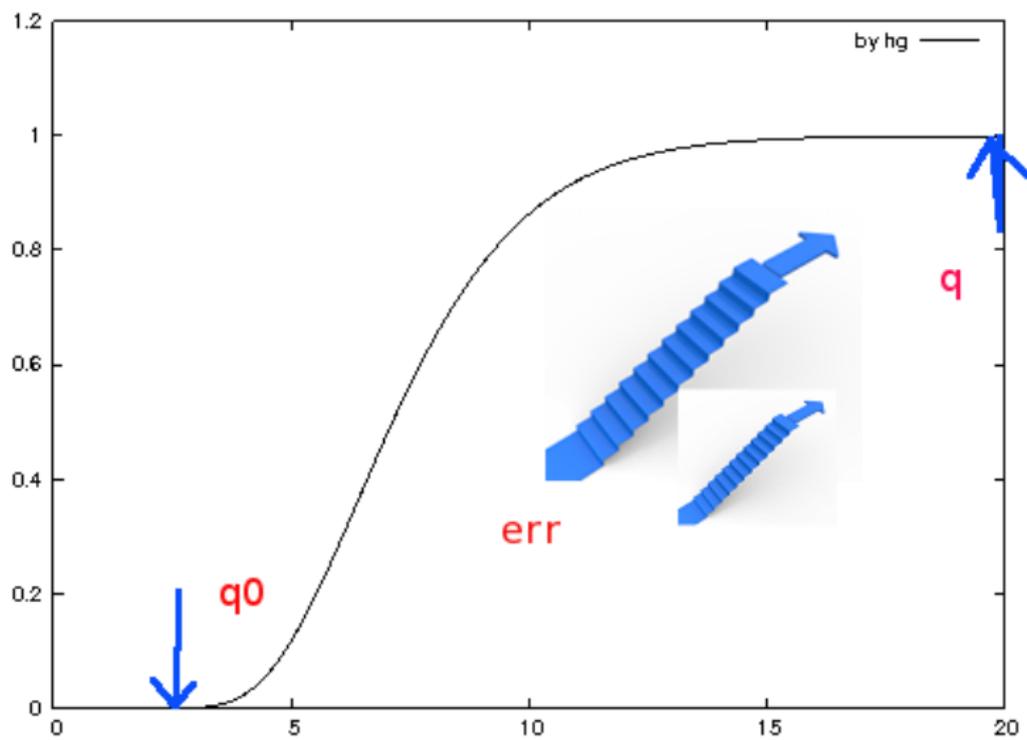
```
plot(hgm.pwishart(m=5,n=20,beta=c(1,2,3,4,5,  
err=c(1e-70,1e-20),autoplot=1,q=100))
```

これは時間がかかる. q0 を増やし, err をすこし大きく.

```
plot(hgm.pwishart(m=5,n=20,q0=0.8,  
beta=c(1,2,3,4,5),  
err=c(1e-30,1e-10),autoplot=1,q=100))
```



q_0 , err , q の調整 (よりよい自動化は今後の課題)



どうしてこうした対処が必要? HGM の原理を知る必要あり (詳細)

hgm の原理を χ^2 分布で説明.

$$T_n(x) = \begin{cases} x^{\frac{n}{2}-1} e^{-\frac{x}{2}} / N_T(n) & x > 0 \\ 0 & x \leq 0 \end{cases}$$

$N_T(n)$ は、上の関数を \mathbf{R} で積分したときに 1 になるための正規化定数で

$$N_T(n) = \int_0^{\infty} x^{\frac{n}{2}-1} e^{-\frac{x}{2}} = 2^{\frac{n}{2}} \Gamma\left(\frac{n}{2}\right)$$

$T_n(x)$ を確率密度関数とする分布は χ^2 分布.

$$\gamma(x) = \int_0^x t^{\frac{n}{2}-1} e^{-\frac{t}{2}} dt$$

数値計算をしたい. Heaviside 関数を用いて

$$\gamma(x) = \int_{-\infty}^{\infty} H(x-t)H(t)t^{\frac{n}{2}-1} e^{-\frac{t}{2}} dt$$

Step 1. $\gamma(x)$ の満たす線形微分方程式を求める.(詳細)

(積分表示を元にして)

$$\begin{aligned}\ell_1 &= (x-t) \left(t\partial_t - \left(\frac{n}{2} - 1 \right) + \frac{1}{2}t \right) \\ \ell_2 &= (x-t)\partial_x\end{aligned}$$

が積分核 f の満たす方程式系.

微分作用素環 $D = K\langle x, t, \partial_x, \partial_t \rangle$ の左イデアル $I = D\ell_1 + D\ell_2$ を考える. I に属する微分作用素は f を零化. (つまり $\ell \in I$ なら $\ell \bullet f = 0$ となる).

$(I + \partial_t D) \cap K\langle x, \partial_x \rangle \ni \ell \neq 0$ とする.

$$\ell = P_1 + \partial_t P_2, \quad (P_1 \in I, P_2 \in D) \quad (3)$$

と書けるので、

$$\begin{aligned}\ell \bullet \int_{-\infty}^{\infty} f(x, t) dt &= \int_{-\infty}^{\infty} P_1 f(x, t) dt + \int_{-\infty}^{\infty} \partial_t P_2 f(x, t) dt \\ &= [P_2 f]_{-\infty}^{\infty} = 0\end{aligned}$$

となつて、 $\ell \bullet \gamma(x) = 0$

ここで ℓ は実は $\ell = 2x\partial_x^2 + (x-n+2)\partial_x$. (グレブナー基底を使う)

```
import("nk_restriction.rr");
L=[(x-t)*(t*dt-(n/2-1)+t/2),
   (x-t)*dx];
G=nk_restriction.integration_ideal(L,[t,x],[dt,dx],[1,0]);
```

Step 2. ℓ を Pfaffian へ. (多変数ではグレブナー基底を使う.)(詳細)

$$\partial_x \begin{pmatrix} \gamma(x) \\ \partial_x \gamma(x) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & -\frac{1}{2x}(x-n+2) \end{pmatrix} \begin{pmatrix} \gamma(x) \\ \partial_x \gamma(x) \end{pmatrix}$$

Step 3. 級数展開、数値積分などを初期値として Step 2 の Pfaffian を Runge-Kutta 法で数値解析して、広い範囲の積分値を求める。(詳細)

(matrix ρF_q では初期値計算に Koev-Edelman 2006)

$\gamma(x)$ の冪級数展開を求めると、

$$\begin{aligned}\int_0^x t^{\frac{n}{2}-1} e^{-\frac{t}{2}} dt &= \sum_{k=0}^{\infty} \int_0^x t^{\frac{n}{2}-1} \frac{1}{k!} \left(-\frac{t}{2}\right)^k dt \\ &= \sum_{k=0}^{\infty} \frac{1}{k!} \left(-\frac{1}{2}\right)^k \int_0^x t^{\frac{n}{2}-1+k} dt \\ &= \sum_{k=0}^{\infty} \frac{1}{k!} \left(-\frac{1}{2}\right)^k \frac{x^{\frac{n}{2}+k}}{\frac{n}{2}+k} \\ &= \frac{n}{2} x^{\frac{n}{2}} {}_1F_1\left(\frac{n}{2}; \frac{n}{2} + 1; -\frac{x}{2}\right)\end{aligned}$$

注意: 自由度 n が大きくなってくると、単純に Runge-Kutta 法を適用するだけでは、精度が得られなくなってくる。このような場合には、 $\gamma(x) = e^{-\frac{x}{2}} x^{\frac{n}{2}} h(x)$ と置いて、 $h(x)$ に対する方程式 $\theta(\theta + \frac{n}{2}) - \frac{x}{2}(\theta + 1)$ を数値解析するなどの工夫が必要になる。

hgm.p2wishart. 自由度 n_i , $m \times m$ 共分散行列 Σ_i できまる Wishart 分布に従う $m \times m$ 行列 W_i .

$$\begin{aligned}
 & P(\ell_1(W_1 W_2^{-1}) \leq x) \\
 = & \frac{\Gamma_m\left(\frac{m+1}{2}\right)\Gamma_m\left(\frac{n_1+n_2}{2}\right)}{\Gamma_m\left(\frac{n_1+m+1}{2}\right)\Gamma_m\left(\frac{n_2}{2}\right)} x^{mn_1/2} |\Sigma_2 \Sigma_1^{-1}|^{n_1/2} \\
 & \cdot {}_2F_1\left(\frac{n_1}{2}, \frac{n_1+n_2}{2}; \frac{n_1+m+1}{2}; -x \Sigma_1^{-1} \Sigma_2\right).
 \end{aligned}$$

$$\Gamma_m(a) = \pi^{\frac{1}{4}m(m-1)} \prod_{i=1}^m \Gamma\left(a - \frac{i-1}{2}\right).$$

Y.Chikuse, Asymptotic expansions for the joint and marginal distributions of the latent roots of $S_1 S_2^{-1}$, Annals of the Institute of Statistical Mathematics 29 (1977), 221-233

Wishart 行列の比の第一固有値, 行列引数超幾何系 ${}_2F_1$

${}_2F_1(a, b; c; Y)$ の Y が対角成分が y_1, \dots, y_m である対角行列であるとする. Muirhead (1970) は ${}_2F_1$ が満たす次の微分方程式 $g_i \bullet {}_2F_1 = 0$ を導いた.

$$g_i = \partial_i^2 + [p(x_i) + \sum_{j \neq i} q_2(x_i, x_j)] \partial_i - \sum_{j \neq i} q(x_i, x_j) \partial_j - r(x_i), \quad i = 1, \dots, m,$$

ここで

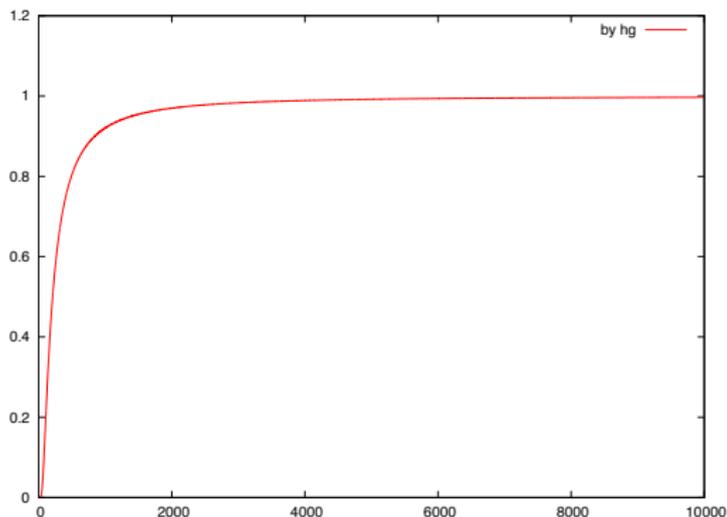
$$p(x_i) = \frac{c - (m-1)/2 - (a+b+1 - (m-1)/2)x_i}{x_i(1-x_i)},$$

$$q_2(x_i, x_j) = \frac{1}{2(x_i - x_j)},$$

$$q(x_i, x_j) = \frac{x_j(1-x_j)}{2x_i(1-x_i)(x_i - x_j)},$$

$$r(x_i) = \frac{ab}{x_i(1-x_i)}.$$

累積分布関数 $P[\ell_1 < x]$, $m = 10, n_1 = 11, n_2 = 12,$
 $\Sigma_2^{-1}\Sigma_1 = \text{diag}(1, 2, \dots, 10)$



Wishart matrices の比の固有値. ${}_2F_1$ の場合.

```
library(hgm)
hgm.p2wishart(m=3,n1=5,n2=10,beta=c(1,2,4),q=4)
[1] 4.0240291 0.5779336 13.1008398 7.2411721 163.8297748
[6] 4.3388397 98.0285307 54.0157128 1217.7748159
```

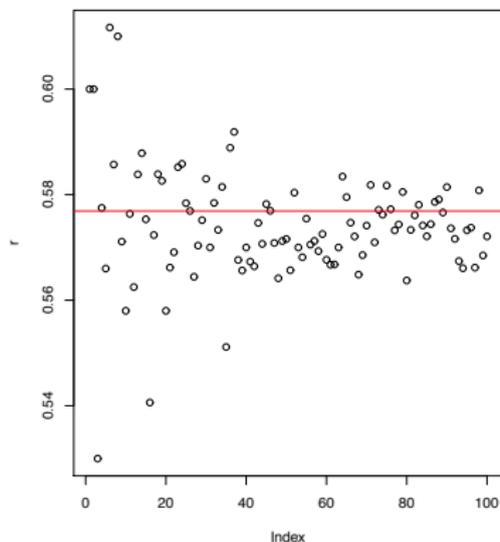


Figure: Wishart matrices の比, 10,000 回までのシミュレーション

hgm/R (R 言語), hgm/src (C 言語), hgm/man (マニュアル), DESCRIPTION, NAMESPACE(百聞は一見にしかず) ソースコード, ほとんどが C.

- ① gsl より借用してきた adaptive Runge-Kutta の code. t- で始まる. wc で 1873 行
- ② matrix ${}_pF_q$ 関連. jack-n.c, code-n*.c wc で 5085 行
- ③ 象限積分. hgm_ko_orthant.c wc で 782 行
- ④ Fisher 積分. so3_nc.c wc で 2233 行 (テーブルデータも含む)
- ⑤ Bingham 分布関連. se_bingham.R wc で 100 行
- ⑥ R と C 部分とのインタフェース. wc で 303 行

CRAN への commit のための覚え書き

```
R CMD check hgm で warning をすべてつぶす.  
R CMD INSTALL hgm  
R CMD REMOVE hgm
```

- ① C 言語. R-exts.pdf が基本文献.
- ② valgrind で memory leak がないか調べる。

```
R CMD check hgm --use-valgrind
```

- ③ R_CheckUserInterrupt(); を呼ぶ. (R/src/sfile.c の mh_check_intr(n))

野呂, $P(\ell_1(W) < x)$. covariance matrix の固有値に重複度がある場合.

```
import("n_wishartd.rr");
n_wishartd.message(1);
P
=n_wishartd.prob_by_hgm(10,100,[9,1],[1/100,1],100|eps=10^(-6));
// m, n, multi, eigen, P(L_1<100)
...
0.516246912217004
266.4sec + gc : 8.277sec(276.8sec)
```

M.Noro. System of partial differential equations for the hypergeometric function ${}_1F_1$ of a matrix argument on diagonal regions. In ISSAC'16 Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation, 381-388, 2016.

分割表の条件付き確率の hgm による正確計算.
後藤, 橘, 高山 (gtt):

```
import("gtt_ekn.rr");  
gtt_ekn.expectation([[4,14,9],[11,6,10]],  
                    //row sum, column sum,  
                    [[1,1/2,1/3],[1,1/5,1/7],[1,1,1]]);  
// prob
```

Q. なぜ asir? A. (1) 数式処理を多用するため. (2) modular method の活用.