รายการอ้างอิง

ภาษาไทย

เคอร์นิกแฮน,คับบลิว,ไบรอัน; และริดชี,เอ็ม,เดนนิส. ภาษาและการโปรแกรม C. เรียบเรียงโดย วิทยา วัชระวิทยากุล. กรุงเทพมหานคร: บริษัท ซีเอ็ดยูเคชั่น จำกัด, 2534.

มนตรี พจนารถลาวัณย์. การเขียนโปรแกรมคอมพิวเตอร์ด้วยเทอร์โบซี. กรุงเทพมหานคร: บริษัท ซีเอ็ดยูเคชั่น จำกัด, 2535.

ภาษาอังกฤษ

Casella, G., and E.I. George. Explaining the Gibbs sampler. The American Statistician. 46, 1992, 167-174.

Castledine, B.J. A Bayesian analysis of mulitple-recapture sampling for a closed population. Biometrika. 67, 1981, 197-210.

Darroch, J.N. The multiple-recapture census I. Biometrika. 45, 1958, 343-359.

Gelfand, A.E., F.M. Smith, and Tai-Ming Lee. Bayesian analysis of constrained parameter and trunoated data problems using Gibbs sampling. Journal of the American statistical Association. 87, 1992, 523-532.

Law, A.M., and W.D. Kelton. Simulation Modeling & Analysis. New York: McGraw-Hill International Inc, 1991.

Ritter, C., and M.a. Tanner. Facilitating the Gibbs sampler: the Gibbs stopper and the Griddy-Gibbs sampler. Journal of the American Statistical Association. 87, 1992, 861-868.

Seber, G.A.F. A note on the multiple-recapture census. Biometrika. 52, 1965, 249-259.

_____. and R. Felton. Tag loss and the Petersen mark-recapture experiment. Biometrika. 68, 1981, 211-219.

Smith, P.J. Bayesian analysis for a multiple capture-recapture model. Biometrika. 78, 1991, 399-407.

Zeger, S.L., and M.R. Karim. Generalised linear models with random effects; a Gibbs sampling approach. Journal of the American Statistical Association. 86, 1991, 79-86.

ภาคผนวก

<u>วิธีการจำลองข้อมูลให้มีการแจกแจงต่าง ๆ มีดังนี้</u>

ก. การจำลองข้อมูล X ให้มีการแจกแจงแบบ Uniform(a,b) มีขั้นตอนดังนี้

1. สุ่มค่า u มาหนึ่งค่า โดยที่ $0 <= u <= 1$
2. ให้ $X = a + (b-a)u$

ข. การจำลองข้อมูล X ให้มีการแจกแจงแบบ Gamma(α ,1) มีขั้นตอนดังนี้

1. คำนวณค่า $b = (2.71282818 + \alpha)/2.71282818$
2. จำลองค่า U_1 ให้มีการแจกแจงแบบ U(0,1)
3. คำนวณ $P = bU_1$ ถ้า $P > 1$ ให้ข้ามไปทำขั้นตอนที่ 5
4. คำนวณ $Y = P^{1/\alpha}$ และจำลองค่า U_2 ให้มีการแจกแจงแบบ U(0,1)

   ถ้า $U_2 <= e^{-Y}$ ให้ $X = Y$

   สำหรับกรณีอื่น ๆ ให้เริ่มต้นทำในข้อ 2
5. คำนวณ $Y = -\ln[(b - P)/\alpha]$ และจำลองค่า U_2 ให้มีการแจกแจงแบบ U(0,1)

   ถ้า $U_2 <= Y^{-1}$ ให้ $X = Y$

   สำหรับกรณีอื่น ๆ ให้เริ่มต้นทำในข้อ 2

ค. การจำลองข้อมูล X ให้มีการแจกแจงแบบ Geo(p) มีขั้นตอนดังนี้

1. สุ่มค่า u มาหนึ่งค่า โดยที่ $0 <= u <= 1$
2. ให้ $X = \ln(u)/\ln(1-p)$

```
 /**********   PROGRAM CALCULATE GIBBS SAMPLER    **********/
#include<stdlib.h>
#include<stdio.h>
#include<conio.h>
#include<math.h>
/**********   MAIN PROGRAM    **********/
main()
{
 long double  alp1[4],alp2[4],s[101],q[101],V[4],W[101],Nr[21];
 long double  Ns,Nh=0.0,ape=0.0;
 long double  Beta();
 long double  NB();
 int          a,b,r,n[4],p1,p2,p3,k,m,l,o=1;
 char         choice='y';
 FILE         *fp;
 srand(32767);
 fp = fopen("b:outg.lis","w");
 while ((choice == 'y') || (choice == 'Y'))
 {
  printf("\n Enter Ns,n1,n2,n3,a,b,r\n");
  scanf("%Lf %d %d %d %d %d %d",&Ns,&n1,&n2,&n3,&a,&b,&r);
  l = 1;
  W[1] = Ns;
  while (l <= 20)
  {
   m = 1;
   while (m <= 100)
   {
    k = 1;
    while (k <= 3)
```

```
        {
          alp1[k] = n[k]+a;
          alp2[k] = W[m]-n[k]+b;
          k = k+1;
        }
        s[m] = (long double)r-1;
        q[m] = (long double)1-((long double)(1-V[1])*(1-V[2])*(1-V[3]));
        W[m+1] = NB(s[m],q[m]);
        m = m+1;
      }
      W[1] = W[100];
      Nr[l] = W[100];
      l = l+1;
    }
    Nh = (long double)Nh/20;
    ape = (long double)fabs((Ns - Nh)*100/Ns);
    fprintf(fp," Nh = %.2Lf      ape = %.2Lf \n",Nh,ape);
    printf("Do you want to continue? (Y=continue,N=exit)");
    choice = getch();
  }
}
/**********     Funotion Beta     **********/
long double  Beta(long double aaa,long double bbb)
{
  long double   b1,b2,SX,X1,X2;
  long double   Gam();
  b1 = ((long double)2.718281828+aaa)/2.718281828;
  b2 = ((long double)2.718281828+bbb)/2.718281828;
  X1 = Gam('N',aaa,b1);
  X2 = Gam('N',bbb,b2);
```

```
 SX = (long double)X1/(X1+X2);
 return(SX);
}
/**********    Funotion Gamma    **********/
long double  Gam(char ans,long double a,long double b)
{
 long double  P,X,Y,u1,u2,u3;
 long double  Uni();
 while (ans == 'N')
 {
  u1 = Uni(0,1);
  P = b*u1;
  if (P > 1)
  {
   Y = (-1)*(log((long double)(b-P)/a));
   u2 = Uni(0,1);
   if (log(u2) <= (a-1)*log(Y))
   {
    X = Y;
    ans = 'Y';
   }
   else
   {
    ans = 'N';
   }
 }
 else
 {
  Y  = pow(P,((long double)1/a));
  u2 = Uni(0,1);
```

```
      u3 = (long double)exp((-1)*Y);
      if (u2 <= u3)
      {
        X = Y;
        ans = 'Y';
      }
      else
      {
        ans = 'N';
      }
    }
  }
  return(X);
}
/**********     Funotion NB()     **********/
long double  NB(long double rr,long double t)
{
  long double   f,Uni();
  int           G[20000],H=0;
  int           j=1;
  while (j <= rr)
  {
    f = Uni(0,1);
    G[j] = ceil(log((long double)f)/log((1-t)));
    H = H+G[j];
    j = j+1;
  }
  H = ceil(H);
  return(H);
}
```

```
/**********     Function Uniform     **********/
long double Uni(aa,bb)
{
  long double  rn,xx=0;
  while (xx == 0)
  {
    rn = (long double)random(32767)/32767;
    xx = (long double)aa+(bb-aa)*rn;
  }
  return(xx);
}
/**********     END OF PROGRAM     **********/

/**********     PROGRAM CALCULATE PETERSEN AND METHOD 3 **********/
#include<stdlib.h>
#include<stdio.h>
#include<conio.h>
#include<math.h>
/**********     MAIN PROGRAM     **********/
main()
{
  long double   N,n1,n2p,n2,n3,m1,m2,m3,p1,p2,p3;
  long double   i,j,k,l,Nhp,Nh1,Nh2,Wh1,Wh2,Nh;
  long double   apep,ape,mape,nmape,mmape,mapep;
  char          ans='y',cans;
  FILE          *fp;
  fp = fopen("b:outp3.lis","w");
  while ((ans == 'y') || (ans == 'Y'))
  {
    printf("\nEnter N, p1, p2, p3 \n");
```

```
scanf("%Lf %Lf %Lf %Lf"),&N, &p1, &p2, P3);
fprintf(fp,"N = %.2Lf p1 = %.2Lf p2 = %.2Lf p3 = %.2Lf\n",N,p1,p2,p3);
i := 1.0;
WHILE (i >= 0.50)
{
 fprintf(fp,"n2 = %L.2f n1",i);
 fprintf(fp,"n1/N            PETERSEN":24);
 fprintf(fp,"     MODEL(II)");
 mmape = 0.0;
 mapep = 0.0;
 j = 0.01;
 WHILE (j <= 0.1)
 {
  k = 1.0;
  nmape = 0.0;
  n1 = ceil(N*j);
  n2p = ceil(n1*i);
  m1 = ceil((n1*n2p*p1)/(N*100));
  Nhp = (long double)(n1+1)*(n2p+1)/(m1+1) - 1;
  apep = fabs((long double)(N - Nhp)*100/N);
  fprintf(fp,"%Lg (%)            %L.5f",j*100,apep);
  WHILE (k >= 0.2)
  {
   mape =  0.0;
   l = 0.1;
   WHILE (l < k)
   {
    n2 = ceil(n2p*l);
    n3 = ceil(n2p*k - n2);
    m1 = ceil((long double)n1*n2*p1/(N*100));
```

```
            m2 = ceil(((long double)n1+n2-m1)*n3*p2/(N*100));
            m3 = ceil((long double)m1*n3*p3/(N*100));
            Nh1 = (long double)(n1+1)*(n2+1)/(m1+1) - 1;
            Nh2 = (long double)(n1+n2-m1+1)*(n3+1)/(m2+1) - 1;
            Wh1 = (long double)(m1+1)/(m1+m2-m3+1);
            Wh2 = (long double)(m2+1)/(m1+m2-m3+1);
            Nh = wh1*Nh1 + wh2*Nh2;
            ape = fabs((long double)(N - Nh)*100/N);
            mape = mape + ape;
            l = l + 0.1;
          }
          nmape = nmape + mape;
          k = k - 0.1;
        }
        nmape = (long double)mape/45;
        fprintf(fp,"                %.5lf\n",nmape;
        mmape = mmape + nmape;
        mapep = mapep + apep;
        j = j + 0.01;
      }
      mmape = (long double)mmape/10;
      mapep = (long double)mapep/10;
      fprintf(fout,'   MAPE      %.5Lf          %.5Lf\n",mapep,mmape);
      i = i - 0.25;
    }
    printf("Do you want to continue? Y = continue, N = exit");
    printf("  Enter your choice \n");
    ans = getch();
  }
}
```

ประวัติผู้เขียน

นางสาววิไลลักษณ์ เฮ้ง เกิดเมื่อวันที่ 22 มิถุนายน พ.ศ. 2512 สำเร็จการศึกษาระดับปริญญาตรีที่มหาวิทยาลัยเกษตรศาสตร์ คณะวิทยาศาสตร์ สาขาวิชาสถิติ ปีการศึกษา 2534 และเข้าศึกษาต่อที่จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2535