# 附錄 B Fortran 95 程式

在此提供本文使用比較 q-B、AVS、Gibbs sampler 三種方法的 Fortran 95 程式。

```
!利用d矩陣判斷貝氏法的Normalized Constant
subroutine calculate(c,I,J,a,b,constant_a,constant_b1,constant_b2)
implicit none
integer k1,k2,m,I,J,n
real constant_a,constant_b1,constant_b2 !準備傳回去的結果
real a(I),b(I,J) !傳入的參數
integer c(I,J) !導入的參數
real c_a(I),c_b1(I,J),c_b2(I) !內部使用
c_a=1.0
c_b2=1.0
c_b1=1.0

do k1=1,I
   if (sum(c(k1,:)) /= 0) then
      do m=1,sum(c(k1,:))
         c_a(k1)=c_a(k1)*(a(k1)+real(m-1))
         c_b2(k1)=c_b2(k1)*(sum(b(k1,:))+real(m-1))
      end do
   end if
   constant_a=constant_a*c_a(k1)
   constant_b2=constant_b2*c_b2(k1)
   do k2=1,J
      if (c(k1,k2)/=0) then
         do m=1,c(k1,k2)
            c_b1(k1,k2)=c_b1(k1,k2)*(b(k1,k2)+real(m-1))
         end do
      end if
   constant_b1=constant_b1*c_b1(k1,k2)
   end do
end do
constant_b1=constant_b1/constant_b2
return
end subroutine calculate


!計算貝氏法的後驗均數
subroutine Bayes(ii,jj,a,b,n,r)
integer ii,jj,i,j,m,n,L,t
INTEGER(KIND=4) k
REAL t1,t2,time_Bayes
REAL constant_a,constant_b2,constant_b1
REAL Coef_A,sum_CoefA
integer r(n),index_k(n),c(ii,jj)
REAL a(ii),b(ii,jj)
REAL*8 expec(ii),eex(ii),std(ii),var(ii),cov(ii,ii)
REAL Coef_B,sum_CoefB
REAL Coef_C,sum_CoefC
REAL Coef_D,sum_CoefD
REAL Coef_E,sum_CoefE
real etime,ta(2)
```

```
time_Bayes=etime(ta)
!計算期望值
do m=1,ii
   sum_CoefA=0.0
   sum_CoefB=0.0
   sum_CoefC=0.0
   sum_CoefD=0.0
   sum_CoefE=0.0
   !將迴圈單一化
   do k=1,ii**n
      L=k-1+(ii**n-1)/(ii-1)
      do i=1,n
         SELECT case(MOD(L,ii))
            case (0)
               index_k(i)=ii
               L=INT(L/ii)-1
            case default
               index_k(i)=MOD(L,ii)
               L=INT(L/ii)
        end select
      end do
      c=0
      do i=1,n
         c(index_k(i),r(i))=c(index_k(i),r(i))+1
      end do
      constant_a=1.0
      constant_b2=1.0
      constant_b1=1.0
      call calculate(c,ii,jj,a,b,constant_a,constant_b1,constant_b2)
      Coef_A=constant_a*constant_b1
      sum_CoefA=sum_CoefA+Coef_A
      Coef_B=Coef_A*REAL(COUNT(index_k==m))
      sum_CoefB=sum_CoefB+Coef_B
      Coef_C=Coef_A*REAL(COUNT(index_k==m))*REAL(COUNT(index_k==m))
      sum_CoefC=sum_CoefC+Coef_C
   end do
      expec(m)=(a(m)+sum_CoefB/sum_CoefA)/(SUM(a)+REAL(n))            !後驗均數
      eex(m)=(a(m)*(a(m)+1)+((2*a(m)+1)*sum_CoefB+sum_CoefC)/sum_CoefA)/((SUM(a)+REAL(n))*(SUM(a)+REAL(n)+1))     !
後驗二階動差
      var(m)=eex(m)-(expec(m)*expec(m))                               !後驗變異數
      std(m)=sqrt(var(m))                                             !後驗標準差

   do t=1,ii
      Coef_D=Coef_A*REAL(COUNT(index_k==t))
      sum_CoefD=sum_CoefD+Coef_D
      Coef_E=Coef_A*REAL(COUNT(index_k==t))*REAL(COUNT(index_k==m))
      sum_CoefE=sum_CoefE+Coef_E
   end do
end do

do m=1,ii
    do t=m,ii
        cov(m,t)=
(a(m)*a(t)+(a(m)*sum_CoefD+a(t)*sum_CoefB+sum_CoefE)/sum_CoefA)/((SUM(a)+REAL(n))*(SUM(a)+REAL(n)+1))
    end do
end do

do t=2,ii
    do m=1,t-1
        cov(t,m)=cov(m,t)
    end do
end do

!後驗共變異數
do m=1,ii
do t=1,ii
   cov(m,t)=cov(m,t)-expec(m)*expec(t)
end do
end do

time_Bayes=etime(ta)-time_Bayes
WRITE(*,*)
write(*,*)" Bayes method "
write(*,"(a12,100F8.4)") " theta= ",expec
write(*,"(a12,100F8.4)") " eex =  ",eex
write(*,"(a12,100F8.4)") " std =  ",std
write(*,"(a12,100F8.4)") " var =  ",var
write(*,"(a12,200f8.4)")"cov=",(cov(1,j),j=1,ii)
do i=2,ii
   write(*,"(a12,200f8.4)")"  ",(cov(i,j),j=1,ii)
end do
write(*,*)" CPU-SEC : Bayes method約",time_Bayes,"秒"
WRITE(*,*) " "
end subroutine Bayes


!利用AVS計算後驗估計值
subroutine averagevariancesum(ii,jj,a,b,nn,r)
integer iseed
integer::ii,jj,nn
integer::i,j,yn,ans,k,m
integer rn(jj),r(nn),total(jj),delta(ii,jj)
real aa(ii),bbb(ii,jj)
real a(ii),b(ii,jj),b_row(jj),brs(ii),bb(ii,jj),bb_row(jj),bbrs(ii)
real*8 p(ii,jj,ii),pp(ii,jj),q(ii,jj,ii),qq(ii,jj),pq(ii,jj),vpq(jj),vvpq(ii)
real*8 AC(ii),AAS(ii),E(ii),EE(ii),Var(ii),V(ii)
real*8 as,VV,alpha_plus
real*8 pha(ii),alpha(ii),ex(ii),avar(ii),ast(ii)
real*8 eta(ii,jj),beta(ii,jj),eta_row(jj),beta_plus(ii),G(ii,jj),w_avs(jj)
real*8 bvar(ii,jj),bbvar(ii,jj),bst(ii,jj)
real etime,time_avs,ta(2)
aa=a
bbb=b
time_avs=etime(ta)
do k=1,nn
    do i=1,ii
       do j=1,jj
          b_row(j)=b(i,j)
       end do
       brs(i)=sum(b_row)
    end do
    do i=1,ii
       AC(i)=a(i)*b(i,r(k))/brs(i)
    end do
    do i=1,ii
       AAS(i)=AC(i)/sum(AC)
    end do

   !在疊代過程中計算得到每一個觀察值後的真實後驗分配theta的估計值
    as=sum(a)
    do i=1,ii
     E(i)=(a(i)+AAS(i))/(as+1)                                     !真實後驗分配theta的期望值
     EE(i)=((a(i)*(a(i)+1))+(2*(a(i)+1)*AAS(i)))/((as+1)*(as+2))   !真實後驗分配theta的二階動差
     Var(i)=EE(i)-(E(i)*E(i))                                      !真實後驗分配theta的變異數
    end do
    do i=1,ii
    V(i)=E(i)*Var(i)
    end do
    VV=sum(V)   !真實後驗分配theta的Average variance sum

!計算簡單的Dirichlet分配中的參數alpha
 do i=1,ii
  pha(i)=E(i)*E(i)*(1-E(i))
 end do
 alpha_plus=(sum(pha)/VV)-1  !alpha+

 do i=1,ii
 alpha(i)=E(i)*alpha_plus   !alpha
 end do

 do i=1,ii
 a(i)=alpha(i)
 end do

 !在疊代過程中計算得到每一個觀察值後的真實後驗分配lambda的估計值
  do m=1,ii
     do i=1,ii
      do j=1,jj
         if ((i==m) .AND. (j==r(k))) then
          delta(i,j)=1
          else
              delta(i,j)=0
          end if
      end do
     end do

       do i=1,ii
            do j=1,jj
              bb(i,j)=b(i,j)+delta(i,j)
              bb_row(j)=bb(i,j)
            end do
            bbrs(i)=sum(bb_row)
            do j=1,jj
              p(i,j,m)=AAS(m)*(bb(i,j)/bbrs(i))
              q(i,j,m)=AAS(m)*((bb(i,j)*(bb(i,j)+1))/(bbrs(i)*(bbrs(i)+1)))
            end do
        end do
   end do

   do m=2,ii
     do i=1,ii
      do j=1,jj
      p(i,j,m)=p(i,j,m)+p(i,j,m-1)
      q(i,j,m)=q(i,j,m)+q(i,j,m-1)
      end do
     end do
   end do

   do m=1,ii
     do i=1,ii
      do j=1,jj
      pp(i,j)=p(i,j,m)  !真實後驗分配lambda的期望值
      qq(i,j)=q(i,j,m)  !真實後驗分配lambda的二階動差
      end do
     end do
   end do

   do i=1,ii
    do j=1,jj
    pq(i,j)=qq(i,j)-(pp(i,j)*pp(i,j))  !真實後驗分配lambda的變異數
    vpq(j)=pp(i,j)*pq(i,j)
    end do
    vvpq(i)=sum(vpq)  !真實後驗分配lambda的Average variance sum
   end do

!計算簡單的Dirichlet分配中的參數beta
  do i=1,ii
    do j=1,jj
        eta(i,j)=pp(i,j)*pp(i,j)*(1-pp(i,j))
        eta_row(j)=eta(i,j)
    end do
    beta_plus(i)=(sum(eta_row)/vvpq(i))-1  !beta(i+)
  end do

  do i=1,ii
   do j=1,jj
        beta(i,j)=pp(i,j)*beta_plus(i)   !beta(ij)
   end do
  end do

  do i=1,ii
     do j=1,jj
      b(i,j)=beta(i,j)
     end do
  end do
end do

do i=1,ii
ex(i)=a(i)/sum(a)                                           !得到所有資料後的後驗近似均數
avar(i)=(a(i)*(sum(a)-a(i)))/(sum(a)*sum(a)*(sum(a)+1))     !得到所有資料後的後驗近似變異數
```

```
ast(i)=sqrt(avar(i))                                              !得到所有資料後的後驗近似標準差
end do

do i=1,ii
 do j=1,jj
  b_row(j)=b(i,j)
 end do
 brs(i)=sum(b_row)
end do

 do i=1,ii
  do j=1,jj
  G(i,j)=b(i,j)/brs(i)                                             !lambda的一階動差
  bvar(i,j)=(b(i,j)*(brs(i)-b(i,j)))/(brs(i)*brs(i)*(brs(i)+1))   !lambda的變異數
  bst(i,j)=sqrt(bvar(i,j))                                          !lambda的標準差
  end do
 end do

w_avs=matmul(ex,G)
time_avs=etime(ta)-time_avs
WRITE(*,*)

write(*,*)"AVS解"
write(*,"(a12,100f8.4)") " w_avs= ",w_avs
write(*,"(a12,100f8.4)") " theta= ",ex
write(*,"(a12,100f8.4)") " std =  ",ast
write(*,"(a12,100f8.4)") " var =  ",avar
write(*,*)"CPU-SEC : Average variance sum約",time_avs,"秒"
write(*,*)
a=aa
b=bbb
end subroutine averagevariancesum



!利用準貝氏法計算後驗估計值
subroutine quasiBayes(ii,jj,a,b,nn,r)
integer::ii,jj,nn
integer::i,j,k
real a(ii),b(ii,jj),b_row(jj)
real aa(ii),bb(ii,jj)
real*8 u(ii)
real*8 G(ii,jj),G_row(jj)
real*8 d(ii),dd(ii)
real*8 w_quasi(jj)
integer r(nn),rn(jj)
integer delta(ii,jj),total(jj)
real*8 v(ii),s(ii),ss(ii,jj),var(ii,jj),q(jj)
real::etime,ta(2),time_quasi
aa=a
bb=b
time_quasi=etime(ta)
do i=1,II
   u(i)=a(i)/sum(a)
   do k=1,JJ
```

```
      G_row(k)=b(i,k)
   end do
   do j=1,JJ
      G(i,j)=b(i,j)/sum(G_row)
   end do
end do

!貝第一步：用u, G 來計算d(i)
do k=1,nn
   do i=1,II
      dd(i)=G(i,r(k))*u(i)
   end do
   do i=1,II
      d(i)=dd(i)/sum(dd)
   end do

!貝第二步：計算出新的 a,b
   do i=1,II
      do j=1,JJ
         if (j==r(k)) then
          delta(i,j)=1
          else
              delta(i,j)=0
          end if
      end do
   end do
   do i=1,II
      a(i)=a(i)+d(i)
      do j=1,JJ
         b(i,j)=b(i,j)+d(i)*delta(i,j)
      end do
   end do

   !貝第三步：計算出新的 u,G
   do i=1,II
      u(i)=a(i)/sum(a)
      do j=1,JJ
         b_row(j)=b(i,j)
      end do
      do j=1,JJ
         G(i,j)=b(i,j)/sum(b_row)
      end do
   end do
   !回到貝第一步
end do

w_quasi=matmul(u,G)

!計算theta的後驗變異數及後驗標準差
do i=1,ii
   v(i)=(a(i)*(sum(a)-a(i)))/(sum(a)*sum(a)*(sum(a)+1))
   s(i)=sqrt(v(i))
end do

!計算lambda的後驗變異數及後驗標準差
```

```
do i=1,ii
 do j=1,jj
  b_row(j)=b(i,j)
 end do
 q(i)=sum(b_row)
end do

do i=1,ii
 do j=1,jj
  var(i,j)=(b(i,j)*(q(i)-b(i,j)))/(q(i)*q(i)*(q(i)+1))
  ss(i,j)=sqrt(var(i,j))
  end do
end do

t2=cpsec()
time_quasi=etime(ta)-time_quasi

WRITE(*,*)
write(*,*)"quasi_Bayes解"
write(*,"(a12,100f8.4)") "w_quasi=",w_quasi
write(*,"(a12,100f8.4)") " theta= ",u
write(*,"(a12,100f8.4)") " std =  ",s
write(*,"(a12,100f8.4)") " var =  ",v
write(*,*)"CPU-SEC : quasi-Bayes method 約",time_quasi,"秒"
write(*,*)
a=aa
b=bb
end subroutine quasiBayes



!利用吉氏取樣器計算後驗估計值
subroutine Gibbs(ii,jj,a,b,nn,r,mm,kk)
integer::ii,jj,nn,mm,kk
integer::i,j,k,n,m,ans
real a(ii),b(ii,jj),aa(ii),bb(ii,jj),b_row(jj)
real*8 u(ii),uu(ii),u_k(kk,ii),u_out(mm,ii)
real*8 G(ii,jj),GG(ii,jj),G_row(jj),G_k(kk,ii,jj),G_out(mm,ii,jj)
real*8 d(ii),dd(ii)
real yy(nn),p(ii),pp(ii,jj)
integer r(nn),y(nn),x(ii),z(ii,jj),total(jj)
real*8 theta(ii),lambda(ii,jj),sd(ii),std(ii),var(ii),c(ii,ii),cov(ii,ii)
real*8 w_gibbs(jj)
integer delta(ii,jj)
real*8 l(ii)
integer::iseed1,iseed2,iseed3
real etime,ta(2),time_Gibbs
aa=a
bb=b
!Gibbs sequence3方程式 !第一步：訂出起始值
!given y
time_Gibbs=etime(ta)
a=aa
b=bb
do i=1,II
```

```
   u(i)=a(i)/sum(a)
   do k=1,JJ
      G_row(k)=b(i,k)
   end do
   do j=1,JJ
      G(i,j)=b(i,j)/sum(G_row)
   end do
end do

do m=1,mm
   iseed1=123**m
   call rnset(iseed1)
   call RNUN(nn,yy)

   do k=1,nn
       do i=1,ii
          l(i)=u(i)*G(i,r(k))
      end do
       do i=2,ii
          l(i)=l(i)+l(i-1)
       end do
       do i=1,ii
          if (yy(k)<(l(i)/l(ii))) then
                 y(k)=i
                 exit
           end if
       end do
   end do

   !第二步：用y,r造出x,z，從這裡開始做20次
   do k=1,kk
   x=0;z=0
     !compute x(i)
      do i=1,ii
         do n=1,nn
            if (y(n)==i) then
                 x(i)=x(i)+1
             end if
         end do
      end do

      !compute[(zij)]
      do n=1,nn
         do i=1,II
             do j=1,JJ
            if (y(n)==i .and. r(n)==j) then
                     z(i,j)=z(i,j)+1
                  end if
            end do
          end do
      end do

      !compute a+x,b+z
      a=aa;b=bb
      do i=1,ii
         a(i)=a(i)+x(i)
         do j=1,jj
            b(i,j)=b(i,j)+z(i,j)
         end do
      end do

      !第三步：用a+x,b+z去造出u,G
      !generate u,G
      iseed2=456*k*m
      !Monte Carlo 第一步
      call rnset(iseed2)
      call rnun(ii,p)      !generate p(3)，再以p去反推u
      call rnun(ii*jj,pp)  !generate pp(3*7)，再以pp去反推G
      do i=1,ii
         uu(i)=chiin(p(i),2*a(i)) !再把p當成機率反推chi-square的反函數值
         do j=1,jj
         if (b(i,j)==0) then
             GG(i,j)=0
         else
                GG(i,j)=chiin(pp(i,j),2*b(i,j))
            end if
         end do
      end do

      !Monte Carlo第二步
      do i=1,ii
         u_k(k,i)=uu(i)/sum(uu)
         do j=1,jj
            G_row(j)=GG(i,j)
         end do
         do j=1,jj
            G_k(k,i,j)=GG(i,j)/sum(G_row)
         end do
      end do

      !第四步：用u,G去生成新的y,再回到第四步
      !generate y
      iseed3=789*k*m
      call rnset(iseed3)
      call rnun(nn,yy)
      do n=1,nn
         do i=1,ii
            dd(i)=G_k(k,i,r(n))*u_k(k,i)
         end do
         do i=2,ii
            dd(i)=dd(i)+dd(i-1)
         end do
         do i=1,ii
            if (yy(n)<(dd(i)/dd(ii))) then
                y(n)=i
                   exit
             end if
         end do
      end do
```

```
        if (k==kk) then
           do i=1,ii
               u_out(m,i)=u_k(k,i)
               do j=1,jj
                  G_out(m,i,j)=G_k(k,i,j)
               end do
           end do
        end if
   end do !做完 k 次以後再從第一步開始
end do !完成 m 次囉！

theta=0;lambda=0;sd=0
do i=1,ii
   do m=1,mm
        theta(i)=theta(i)+u_out(m,i)
   end do
   theta(i)=theta(i)/mm                       !theta的後驗近似均數
   do m=1,mm
    sd(i)=sd(i)+u_out(m,i)*u_out(m,i)
   end do
   var(i)=(sd(i)-(mm*theta(i)*theta(i)))/(mm-1)      !theta的後驗近似變異數
   std(i)=sqrt(var(i))                               !theta的後驗近似標準差

   do j=1,jj
      do m=1,mm
           lambda(i,j)=lambda(i,j)+G_out(m,i,j)
        end do
        lambda(i,j)=lambda(i,j)/mm            !lambda的後驗近似均數
   end do
end do
w_gibbs=matmul(theta,lambda)

c=0
do i=1,ii
 do j=1,ii
  do m=1,mm
  c(i,j)=c(i,j)+u_out(m,i)*u_out(m,j)
  end do
 end do
 !write(*,"(a10,200f10.5)")"c=",(c(i,j),j=1,ii)
end do

do i=1,ii
 do j=1,ii
 cov(i,j)=(c(i,j)-mm*theta(i)*theta(j))/(mm-1)    !theta的後驗近似共變異數
 end do
end do

write(*,*) "Gibbs sampler解"
write(*,"(a12,100f8.4)") "w_gibbs=",(w_gibbs(j),j=1,jj)
write(*,"(a12,100f8.4)") " theta= ",theta
write(*,"(a12,100f8.4)") " std =  ",std
write(*,"(a12,100f8.4)") " var =  ",var
write(*,"(a12,200f8.4)")"cov=",(cov(1,j),j=1,II)
do i=2,ii
```

```
   write(*,"(a12,200f8.4)")"  ",(cov(i,j),j=1,II)
end do

time_Gibbs=etime(ta)-time_Gibbs
write(*,*)" CPU-SEC：Gibbs sampler約",time_Gibbs,"秒"
a=aa
b=bb
end subroutine Gibbs



!輸入所需要的參數及資料
program comparision
include 'link_f90_dll.h'
integer i,j,n,ans,mm,kk,mmp
real etime,ta(2),tot
REAL*8, allocatable::rea_theta(:),rea_lambda(:,:),w_rea(:)
REAL, allocatable::a(:),b(:,:)
REAL*8, allocatable::expec(:),theta(:),u(:),ex(:)
REAL*8, ALLOCATABLE::rel_Gibbs(:),rel_quasi(:),rel_avs(:)
REAL*8, ALLOCATABLE::Gib(:),Qua(:),Avs(:)
REAL*8, ALLOCATABLE::w_gibbs(:),w_quasi(:),w_avs(:)
integer,allocatable::rn(:),r(:),total(:)


10  write(*,*) "輸入：類別數(I):"
    read(*,*) ii
    write(*,*)"類別數 I=",ii
    write(*,*)"確定嗎？(Yes=1/No=2)"
    read(*,*)ans
    if(ans/=1) then
        goto 10
     end if

20   write(*,*) "輸入：報告共分多少種類(J)"
     read(*,*) jj
     write(*,*)"報告種類 J=",jj
     write(*,*)"確定嗎？(Yes=1/No=2)"
     read(*,*)ans
     if(ans/=1) then
        goto 20
     end if

     allocate(a(ii));allocate(b(ii,jj))
     ALLOCATE(w_Gibbs(ii))
     ALLOCATE(w_quasi(ii))
     ALLOCATE(w_avs(ii))
     ALLOCATE(expec(ii))
     ALLOCATE(theta(ii))
     ALLOCATE(u(ii))
     ALLOCATE(ex(ii))
     ALLOCATE(w_rea(jj))
     allocate(total(jj));allocate(rn(jj))

30       write(*,*)" 輸入：Gibbs 取起始值的次數 (m)"
```

```
      read(*,*)mm
      write(*,*)"Gibbs 取起始值的次數 (m)=",mm
    write(*,*)"確定嗎？(Yes=1/No=2)"
      read(*,*)ans
      if(ans/=1) then
         goto 30
      end if

40    write(*,*)" 輸入：每次疊代的次數 (k)"
      read(*,*)kk
      write(*,*)"每次疊代的次數 (k)=",kk
    write(*,*)"確定嗎？(Yes=1/No=2)"
      read(*,*)ans
      if(ans/=1) then
         goto 40
      end if

50    write(*,*)"輸入：樣本總數 (n)"
    read(*,*) nn

60    write(*,*)"輸入：每種報告的個數（其和爲總樣本數)"
    read(*,*)(rn(j),j=1,jj)
      do j=2,jj
         rn(j)=rn(j)+rn(j-1)
      end do
      if(nn/=rn(jj)) then
        write(*,*)"每種報告個數的和不等於總樣本數，請重新輸入。"
        goto 50
      end if

  allocate(r(nn))
  !決定樣本報告種類
  do k=1,nn
     do j=1,jj
        if (k<=rn(j)) then
          r(k)=j
          !write(*,*)"k=",k,"r(k)=",r(k)
           exit
         end if
      end do
   end do
  ! write(*,"(a15,100i5)")"r=",r

70   write(*,*)"輸入： prior a:"
   read(*,*)(a(i),i=1,ii)
   write(*,"(a7,100f6.2)")"a=",a
     write(*,*)"確定嗎？ (Yes=1/No=2)"
     read(*,*)ans
      if(ans/=1) then
         goto 70
      end if

80   do i=1,ii
       write(*,"(a20,i3)")"輸入：prior bi*,i=",i
       read(*,*)(b(i,j),j=1,jj)
     end do

    write(*,*)"b="
    do i=1,ii
         write(*,"(a7,2000f6.2)")"  ",(b(i,j),j=1,jj)
    end do
      write(*,*)"確定嗎？ (Yes=1/No=2)"
      read(*,*)ans
      if(ans/=1) then
         goto 80
      end if

    total=0
    do j=1,jj
        do k=1,nn
          if (r(k)==j) then
             total(j)=total(j)+1
          end if
        end do
    end do
    write(*,"(a15,200i6)")"各報告個數爲",total
    do j=2,jj
        total(j)=total(j)+total(j-1)
    end do
    write(*,"(a15,2000i6)")"累計後得",total
   WRITE(*,*) "類別數 I= ",ii
   write(*,*) "報告種類 J=",jj
   write(*,"(a8,100f8.2)")"prior a= ",a
   WRITE(*,*) "prior b="
   do i=1,ii
      write(*,"(a7,2000f8.2)")"  ",(b(i,j),j=1,jj)
   end do
   WRITE(*,*) "總樣本數 n=", nn
   write(*,*)"Gibbs 取起始値的次數 (m)=",mm
   write(*,*)"每次疊代的次數 (k)=",kk
   write(*,*)"所有的資料都輸入正確嗎？(Yes=1/No=2)"
   read(*,*)ans
   if(ans/=1) then
      WRITE(*,*) "所有資料重新輸入"
      deallocate(a);deallocate(b)
      deALLOCATE(w_Gibbs);deALLOCATE(w_quasi);deALLOCATE(w_avs);deALLOCATE(expec)
      deALLOCATE(theta);deALLOCATE(u);deALLOCATE(ex);deALLOCATE(w_rea)
      deallocate(total);deallocate(rn);deallocate(r)
      goto 10
   end if
WRITE(*,*)"---------------------"
tot=etime(ta)
      call Bayes(ii,jj,a,b,nn,r)
      call Gibbs(ii,jj,a,b,nn,r,mm,kk)
      call quasiBayes(ii,jj,a,b,nn,r)
      call averagevariancesum(ii,jj,a,b,nn,r)
tot=etime(ta)-tot
WRITE(*,*) "--------------------------------------"
WRITE(*,*) "此程式所花總時間爲 ",tot, "秒"
end
```