

SUPPLEMENTAL MATERIALS OF PAPER

Inference for partially observed competing risks model for Kumaraswamy distribution under generalized progressive hybrid censoring

Amulya Kumar Mahto^a, Chandrakant Lodhi^b, Yogesh Mani Tripathi^a and Liang Wang^c

^aDepartment of Mathematics, Indian Institute of Technology, Patna-801103, India; ^bneurIOT Technologies LPP, CoWorks, Golf, Course, Gurugram, India; ^cSchool of Mathematics, Yunnan Normal University, Kunming 650500, P.R. China.

ARTICLE HISTORY

Compiled January 26, 2021

A. Proof of article results

The proof of Theorem 1

After taking derivatives of log-likelihood function $l(\alpha, \lambda_1, \lambda_2)$ with respect to λ_1 and λ_2 and equating to zero, we get

$$\frac{e_0}{\lambda_0} + \frac{e_1}{\lambda_1} = \psi(\alpha) \quad \text{and} \quad \frac{e_0}{\lambda_0} + \frac{e_2}{\lambda_2} = \psi(\alpha), \quad (1)$$

which implies that $\frac{e_1}{\lambda_1} = \frac{e_2}{\lambda_2}$.

Using above expression in Equation (1), we have the following

$$\lambda_1 = \frac{e_1}{\psi(\alpha)} \left[\frac{e_0 + e_1 + e_2}{e_1 + e_2} \right] \quad \text{and} \quad \lambda_2 = \frac{e_2}{\psi(\alpha)} \left[\frac{e_0 + e_1 + e_2}{e_1 + e_2} \right]$$

which are rewritten as

$$\lambda_j = \frac{e_j}{\psi(\alpha)} \left[\frac{e_0 + e_1 + e_2}{e_1 + e_2} \right], \quad j = 1, 2.$$

We further show that the MLEs of λ_1 and λ_2 maximize log-likelihood function $l(\alpha, \lambda_1, \lambda_2)$ for given α . Let $H(\lambda_1, \lambda_2)$ be the Hessian matrix of $l(\alpha, \lambda_1, \lambda_2)$ of (λ_1, λ_2) . Then

$$H_{jj}(\lambda_1, \lambda_2) = \frac{\partial^2 l}{\partial \lambda_j^2} = -\frac{e_0}{\lambda_0^2} - \frac{e_1}{\lambda_j^2}, \quad j = 1, 2 \quad \text{and} \quad H_{12}(\lambda_1, \lambda_2) = \frac{\partial^2 l}{\partial \lambda_1 \partial \lambda_2} = -\frac{e_0}{\lambda_0^2}$$

and determinant of the Hessian matrix is given by

$$\begin{aligned} D(\hat{\lambda}_1, \hat{\lambda}_2) &= H_{11}(\hat{\lambda}_1, \hat{\lambda}_2)H_{22}(\hat{\lambda}_1, \hat{\lambda}_2) - (H_{12}(\hat{\lambda}_1, \hat{\lambda}_2))^2 \\ &= \left(\frac{e_0}{\lambda_0^2} + \frac{e_1}{\lambda_1^2}\right) \left(\frac{e_0}{\lambda_0^2} + \frac{e_2}{\lambda_2^2}\right) - \left(\frac{e_0}{\lambda_0^2}\right)^2 \\ &= \frac{e_0 e_1}{\lambda_0^2 \lambda_1^2} + \frac{e_0 e_2}{\lambda_0^2 \lambda_2^2} + \frac{e_1 e_2}{\lambda_1^2 \lambda_2^2} > 0. \end{aligned}$$

So $(\hat{\lambda}_1, \hat{\lambda}_2)$ is the local maximum of $l(\alpha, \lambda_1, \lambda_2)$ for given α . Also there is no singular point of $l(\alpha, \lambda_1, \lambda_2)$ and it has a single critical point, thus $\hat{\lambda}_1$ and λ_2 are the absolute maximum of $l(\alpha, \lambda_1, \lambda_2)$.

The proof of Theorem 2

The profile log-likelihood function of α with $\lambda_j = \lambda_j(\alpha)$ and using $\sum_{j=0}^2 e_j = d^*$ in (6) is given by

$$l_p(\alpha) \propto d^* \log \alpha - d^* \log \psi(\alpha) + \sum_{i=1}^{d^*} \log \left(\frac{y_{i:m:n}^{\alpha-1}}{1 - y_{i:m:n}^\alpha} \right). \quad (2)$$

By equating the first derivative of (2) with respect to α to zero, we obtain

$$\frac{1}{\alpha} + \frac{1}{d^*} \sum_{i=1}^{d^*} \left(\frac{\ln y_{i:m:n}}{1 - y_{i:m:n}^\alpha} \right) = \frac{\psi'(\alpha)}{\psi(\alpha)} \quad (3)$$

where $\psi'(\alpha)$ is derivative of $\psi(\alpha)$ with respect to α and is given by the expression

$$\psi'(\alpha) = \begin{cases} \sum_{i=1}^k (1 + R_i) \{y_{i:m:n}^\alpha \ln y_{i:m:n} / (1 - y_{i:m:n}^\alpha)\}, & \text{case I,} \\ \sum_{i=1}^d (1 + R_i) \{y_{i:m:n}^\alpha \ln y_{i:m:n} / (1 - y_{i:m:n}^\alpha)\} + R_{d+1}^* \{T^\alpha \ln T / (1 - T^\alpha)\}, & \text{case II,} \\ \sum_{i=1}^m (1 + R_i) \{y_{i:m:n}^\alpha \ln y_{i:m:n} / (1 - y_{i:m:n}^\alpha)\}, & \text{case III.} \end{cases}$$

We verify uniqueness and existence of MLE of α . Consider

$$H_1(\alpha) = \frac{1}{\alpha} + \frac{1}{d^*} \sum_{i=1}^{d^*} \frac{\ln y_{i:m:n}}{1 - y_{i:m:n}^\alpha} \quad \text{and} \quad H_2(\alpha) = \frac{\psi'(\alpha)}{\psi(\alpha)}.$$

Note that $H_1(\alpha)$ is increasing in α and we also see that

$$\lim_{\alpha \rightarrow 0} H_1(\alpha) = +\infty \quad \text{and} \quad \lim_{\alpha \rightarrow +\infty} H_1(\alpha) = \frac{1}{d^*} \sum_{i=1}^{d^*} \ln y_{i:m:n}.$$

Now for $H_2(\alpha)$, on differentiating w.r.t. α , we have

$$\frac{dH_2(\alpha)}{d\alpha} = \frac{\psi(\alpha)\psi''(\alpha) - (\psi'(\alpha))^2}{[\psi(\alpha)]^2}$$

The numerator of the above expression is given by

$$\left\{ \begin{array}{l} \left(-\sum_{i=1}^k (1+R_i) \ln(1-y_{i:m:n}^\alpha) \right) \left(\sum_{i=1}^k (1+R_i) y_{i:m:n}^\alpha \left(\frac{\ln y_{i:m:n}}{1-y_{i:m:n}^\alpha} \right)^2 \right) \\ - \left(\sum_{i=1}^k (1+R_i) \frac{y_{i:m:n}^\alpha \ln y_{i:m:n}}{1-y_{i:m:n}^\alpha} \right)^2, \quad \text{case I,} \\ \left(-\sum_{i=1}^d (1+R_i) \ln(1-y_{i:m:n}^\alpha) - R_{d+1}^* \ln(1-T^\alpha) \right) \left(\sum_{i=1}^d (1+R_i) y_{i:m:n}^\alpha \left(\frac{\ln y_{i:m:n}}{1-y_{i:m:n}^\alpha} \right)^2 \right) \\ + R_{d+1}^* T^\alpha \left(\frac{\ln T}{1-T^\alpha} \right)^2 - \left(\sum_{i=1}^d (1+R_i) \frac{y_{i:m:n}^\alpha \ln y_{i:m:n}}{1-y_{i:m:n}^\alpha} \right) + R_{d+1}^* \frac{T^\alpha \ln T}{1-T^\alpha} \right)^2, \quad \text{case II,} \\ \left(-\sum_{i=1}^m (1+R_i) \ln(1-y_{i:m:n}^\alpha) \right) \left(\sum_{i=1}^m (1+R_i) y_{i:m:n}^\alpha \left(\frac{\ln y_{i:m:n}}{1-y_{i:m:n}^\alpha} \right)^2 \right) \\ - \left(\sum_{i=1}^m (1+R_i) \frac{y_{i:m:n}^\alpha \ln y_{i:m:n}}{1-y_{i:m:n}^\alpha} \right)^2, \quad \text{case III.} \end{array} \right.$$

Using the Cauchy-Schwarz inequality, we see that $\frac{dH_2(\alpha)}{d\alpha} > 0$ and thus $H_2(\alpha)$ is increasing in α . Further, we also obtain

$$\lim_{\alpha \rightarrow 0} H_2(\alpha) = -\infty$$

and

$$\lim_{\alpha \rightarrow +\infty} H_2 = \begin{cases} \ln y_{k:m:n}, & \text{case I,} \\ \ln T, & \text{case II,} \\ \ln y_{m:m:n}, & \text{case III.} \end{cases}$$

We also have the following

$$\lim_{\alpha \rightarrow 0} \frac{H_1(\alpha)}{H_2(\alpha)} = \begin{cases} \frac{1}{d^*} \sum_{i=1}^{d^*} \frac{\ln y_{i:m:n}}{\ln y_{k:m:n}}, & \text{case I,} \\ \frac{1}{d^*} \sum_{i=1}^{d^*} \frac{\ln y_{i:m:n}}{\ln T}, & \text{case II,} \\ \frac{1}{d^*} \sum_{i=1}^{d^*} \frac{\ln y_{i:m:n}}{\ln y_{m:m:n}}, & \text{case III.} \end{cases} < 1.$$

This leads to the conclusion that curves $H_1(\alpha)$ and $H_2(\alpha)$ have a unique intersection point. Thus MLE of α exists and is unique also.

The proof of Theorem 4

When $e_1 < e_2$, the MLEs of λ_j , $j = 1, 2$ as obtained in Theorem 1 satisfy the order restriction $\tilde{\lambda}_1 < \tilde{\lambda}_2$ for given α . So for given α , MLEs of λ_j under the order restriction is given by

$$\tilde{\lambda}_j = \frac{e_j}{\psi(\alpha)} \left[\frac{e_0 + e_1 + e_2}{e_1 + e_2} \right], \quad j = 1, 2, \quad \text{when } e_1 < e_2.$$

If $e_1 \geq e_2$, MLEs obtained in Theorem 1 do not hold order restriction condition. In this case, the likelihood function $l(\alpha, \lambda_1, \lambda_2; \text{data})$ is maximized on $\lambda_1 = \lambda_2 = \lambda$ under the restriction $\lambda_1 \leq \lambda_2$. Therefore, associated MLEs $\tilde{\lambda}_1$ and $\tilde{\lambda}_2$ can be obtained from

$\max_{\lambda>0} \{\}$, that is,

$$\max_{\lambda} \left\{ \log C^* + d^* \ln \lambda + e_0 \log(2\lambda) + (e_1 + e_2) \ln \lambda - 2\lambda \psi(\alpha) + \sum_{i=1}^{d^*} \ln \left(\frac{y_i^{\alpha-1}}{1 - y_i^\alpha} \right) \right\}.$$

Taking the derivative of above equation with respect to λ , we obtain MLEs of λ_1 and λ_2 for given α as

$$\tilde{\lambda}_1 = \tilde{\lambda}_2 = \frac{e_0 + e_1 + e_2}{2\psi(\alpha)}, \quad \text{when } e_1 \geq e_2.$$

B. Program Code of the Illustrative Example

```
##Real Data I (Partially Observed Competing Risk(Kumaraswamy Distribution))##
rm(list=ls(all=TRUE));
rm(list=ls(all=TRUE));
rm(list=ls(all=TRUE));
library(nleqslv);
# times <- 1000;
# a1 <- 1.5;
# bt1 <- 0.7;
# bt2 <- 0.9;
xsamp <- c(0.0222, 0.0333, 0.0333, 0.0778, 0.0889, 0.1000, 0.1222, 0.1444, 0.1556,
           0.1556, 0.1667, 0.1778, 0.1778, 0.2000, 0.2000, 0.2111, 0.2222, 0.2667, 0.2889,
           0.3000, 0.3111, 0.3111, 0.3333, 0.3778, 0.4000);
n <- 37; m <- 30; k=25;
R <- c(rep(0,(m-1)),(n-m));
T=0.3;
los <- 0.05;
xstart <- 1;
MLE_Estimation <- function(ings, ysp, dst, rr, rrstr, tt){ # MLE for nu using nleqslv
  lik_eqn <- function(nu){
    lkeq <- (dst/nu)+(sum((log(ysp))/(1-(ysp^nu))))-(dst*((sum((rr+1)*(((ysp^nu)*
    log(ysp))/(1-(ysp^nu))))+(rrstr*((tt^nu)*log(tt))/((1-(tt^nu))))))/
    (-sum((rr+1)*log(1-(ysp^nu)))-(rrstr*log(1-(tt^nu))));
    return(lkeq);
  }
  mlnu <- nleqslv(ings, lik_eqn);
  #cat(mlnu$message);
  return(mlnu$x);
}

Bayes_Estimation <- function(xsp, dst, rr, rrstr, tt){
  #a1 = 6; b1 = 4; a0 = 3.2; b0 = 2; c1 <- (7/9); c2 <- 1;
  a1 = 0.00001; b1 = 2; a0 = 0.00001; b0 = 2; c1 <- 0.00001; c2 <- 2;
  dens_of_nu <- function(vnu, xsm, ds, rst, ri, TTT){
    # Density of nu from posterior density
    s1=0;s2=0;s3= s4= 0;
    for(jk in 1:length(xsm)){
```

```

    s1 = s1 + ((ri[jk]+1)*log(1-(xsm[jk]^vnu)))
    s2 = s2 + ((ri[jk]+1)*(((xsm[jk]^(vnu))*log(xsm[jk]))/(1-(xsm[jk]^(vnu)))));
    s3 = s3 + ((ri[jk]+1)*((xsm[jk]^(vnu))*(log(xsm[jk]))/(1-(xsm[jk]^(vnu))))^(2))
    s4 = s4 + log((xsm[jk]^(vnu-1))/(1-(xsm[jk]^vnu)))
  }
  W_nu <- (-s1 - rst*log(1-(TTT^vnu)));
  W_nuD <- (s2 + (rst*(((TTT^(vnu))*log(TTT))/(1-(TTT^(vnu))))));
  W_nuDD <- (s3 + (rst*(((TTT^(vnu))*((log(TTT))/(1-(TTT^vnu))))^(2))))
  #fnval <- (((vnu^(a1 + ds - 1))/((b0 + W_nu)^(a0 + ds)))*exp(-(b1-sum(log(xsm)))*
  vnu))/exp(-vnu);
  fnval <- (((vnu^(a1 + ds - 1))/((b0 + W_nu)^(a0 + ds)))*exp(-(b1*vnu)+(s4)))
  #/exp(-vnu);
  #fDDVal <- (sum(((xsm^vnu)*((log(xsm^vnu))^2))/((1-(xsm^vnu))^2)))/(a1+ds-1)
  return(fnval);
  #-((a1+ds-1)/(vnu^2))-((a0+ds)*(((b0+W_nu)*(W_nuDD))-(W_nuD^2))/((b0+W_nu)^2))
}
dens_of_nu_mean <- function(vnu, xsm, ds, rst, ri, TTT,cnst){
  # Density of nu from posterior density
  s1=0;s2=0;s3=s4=0;
  for(jk in 1:length(xsm)){
    s1 = s1 + ((ri[jk]+1)*log(1-(xsm[jk]^vnu)))
    s2 = s2 + ((ri[jk]+1)*(((xsm[jk]^(vnu))*log(xsm[jk]))/(1-(xsm[jk]^(vnu)))));
    s3 = s3 + ((ri[jk]+1)*((xsm[jk]^(vnu))*(log(xsm[jk]))/(1-(xsm[jk]^(vnu))))^(2))
    s4 = s4 + log((xsm[jk]^(vnu-1))/(1-(xsm[jk]^vnu)))
  }
  W_nu <- (-s1 - rst*log(1-(TTT^vnu)));
  W_nuD <- (s2 + (rst*(((TTT^(vnu))*log(TTT))/(1-(TTT^(vnu))))));
  W_nuDD <- (s3 + (rst*(((TTT^(vnu))*((log(TTT))/(1-(TTT^vnu))))^(2))))
  #fnval <- (((vnu^(a1 + ds - 1))/((b0 + W_nu)^(a0 + ds)))*
  exp(-(b1-sum(log(xsm))*vnu))/exp(-vnu));
  fnval <- (1/cnst)*vnu*(((vnu^(a1 + ds - 1))/((b0 + W_nu)^(a0 +
  ds)))*exp(-(b1*vnu)+(s4))) #/exp(-vnu);
  #fDDVal <- (sum(((xsm^vnu)*((log(xsm^vnu))^2))/((1-(xsm^vnu))^2)))/(a1+ds-1)
  return(fnval);
  #-((a1+ds-1)/(vnu^2))-((a0+ds)*(((b0+W_nu)*(W_nuDD))-(W_nuD^2))/((b0+W_nu)^2))
}
dens_of_nu_mean_2 <- function(vnu, xsm, ds, rst, ri, TTT,cnst){
  # Density of nu from posterior density
  s1=0;s2=0;s3=s4=0;
  for(jk in 1:length(xsm)){
    s1 = s1 + ((ri[jk]+1)*log(1-(xsm[jk]^vnu)))
    s2 = s2 + ((ri[jk]+1)*(((xsm[jk]^(vnu))*log(xsm[jk]))/(1-(xsm[jk]^(vnu)))));
    s3 = s3 + ((ri[jk]+1)*((xsm[jk]^(vnu))*(log(xsm[jk]))/(1-(xsm[jk]^(vnu))))^(2))
    s4 = s4 + log((xsm[jk]^(vnu-1))/(1-(xsm[jk]^vnu)))
  }
  W_nu <- (-s1 - rst*log(1-(TTT^vnu)));
  W_nuD <- (s2 + (rst*(((TTT^(vnu))*log(TTT))/(1-(TTT^(vnu))))));
  W_nuDD <- (s3 + (rst*(((TTT^(vnu))*((log(TTT))/(1-(TTT^vnu))))^(2))))
  #fnval <- (((vnu^(a1 + ds - 1))/((b0 + W_nu)^(a0 + ds)))*
  exp(-(b1-sum(log(xsm))*vnu))/exp(-vnu));

```

```

fnval <- (1/cnst)*(vnu^2)*(((vnu^(a1 + ds - 1))/((b0 + W_nu)^(a0 + ds)))*
exp(-(b1*vnu)+(s4))) #/exp(-vnu);
#fDDVal <- (sum(((xsm^vnu)*((log(xsm^vnu))^(2)))/((1-(xsm^vnu))^(2))))/(a1+ds-1)
return(fnval);
#-((a1+ds-1)/(vnu^2))-((a0+ds)*(((b0+W_nu)*(W_nuDD))-(W_nuD^2))/((b0+W_nu)^(2)))
}
const <- integrate(dens_of_nu, 0, Inf, xsm=xsp, ds=dst,
rst=rrstr, ri=rr, TTT=tt)$value;
mn <- integrate(dens_of_nu_mean, 0, Inf, xsm=xsp, ds=dst,
rst=rrstr, ri=rr, TTT=tt, cnst = const)$value; # Calculating first moment
mn2 <- integrate(dens_of_nu_mean_2, 0, Inf, xsm=xsp, ds=dst,
rst=rrstr, ri=rr, TTT=tt, cnst = const)$value; # Calculating first moment
#cat("\n The mean and variance", c(const,mn,mn2))
vr <- (mn2-((mn)^2)); # Calculating variance of using density of nu
a5 <- (mn^2)/vr; b5 <- mn/vr;
# Calculating parameters for the approximating gamma distribution
#cat("\n mn and vr =",a5/b5);
if(a5<0 || b5 <0){
  a5 <- 1.5;
  b5 <- 1;
  count = 1;
}else{
  count = 0;
}
#cat("\n a5 and b5 =",c(a5,b5));
baynuu <- numeric(0);
baybtt1 <- numeric(0);
baybtt2 <- numeric(0);
for(ii in 1:4000){
  baynuu[ii] <- rgamma(1, a5, b5);
  #print(baynuu[ii]);
  w_nu <- -sum((rr+1)*log(1-(xsp^baynuu[ii])))- rrstr*log(1-(tt^baynuu[ii]));
  # Calculating the function W(nu)
  bt0 <- rgamma(1, a0+dst,b0+w_nu);
  # Generating random beta0 = (beta1 + beta2) from gamma distribution
  btbybt0 <- rbeta(1, c1+e1, c2+e2); # Generating random beta
  baybtt1[ii] <- bt0*btbybt0; # Generated values of beta1
  baybtt2[ii] <- bt0-baybtt1[ii]; # Generated values of beta1
}
# BAYNUU <- baynuu[(1001:4000)];
# BAYBTT1 <- baybtt1[(1001:4000)];
# BAYBTT2 <- baybtt2[(1001:4000)];
nu_nw <- mean((baynuu[(1001:4000)]));
#cat("\n Mean value of nu", nu_nw)
bt1_nw <- mean((baybtt1[(1001:4000)]));
bt2_nw <- mean((baybtt2[(1001:4000)]));
bynums <- (nu_nw - 11)^2;
bybt1ms <- (bt1_nw - 11)^2;
bybt2ms <- (bt2_nw - 11)^2;
baynuusort <- sort((baynuu[(1001:4000)]));

```

```

baybtt1sort <- sort((baybtt1[(1001:4000)]));
baybtt2sort <- sort((baybtt2[(1001:4000)]));
Bayes_low_nu <- Bayes_upp_nu <- Bayes_len_nu <- numeric(0);
Bayes_low_bt1 <- Bayes_upp_bt1 <- Bayes_len_bt1 <- numeric(0);
Bayes_low_bt2 <- Bayes_upp_bt2 <- Bayes_len_bt2 <- numeric(0);
for(ir in 1:(length(baynuusort)*los)){
  Bayes_low_nu[ir] <- (baynuusort[ir])
  Bayes_upp_nu[ir] <- (baynuusort[ir+((length(baynuusort)*(1-los)))]))
  Bayes_len_nu[ir] <- (Bayes_upp_nu[ir]- Bayes_low_nu[ir])
  Bayes_low_bt1[ir] <- (baybtt1sort[ir])
  Bayes_upp_bt1[ir] <- (baybtt1sort[ir+((length(baynuusort)*(1-los)))]))
  Bayes_len_bt1[ir] <- (Bayes_upp_bt1[ir]- Bayes_low_bt1[ir]);
  Bayes_low_bt2[ir] <- (baybtt2sort[ir])
  Bayes_upp_bt2[ir] <- (baybtt2sort[ir+((length(baynuusort)*(1-los)))]))
  Bayes_len_bt2[ir] <- (Bayes_upp_bt2[ir]- Bayes_low_bt2[ir]);
}
#print(baynuusort)
id1 <- which(Bayes_len_nu==min(Bayes_len_nu));
id2 <- which(Bayes_len_bt1==min(Bayes_len_bt1));
id3 <- which(Bayes_len_bt2==min(Bayes_len_bt2));
HPD_low_nu <-Bayes_low_nu[id1];
HPD_upp_nu <-Bayes_upp_nu[id1];
HPD_len_nu <-(HPD_upp_nu-HPD_low_nu)
HPD_low_bt1 <-Bayes_low_bt1[id2];
HPD_upp_bt1 <-Bayes_upp_bt1[id2];
HPD_len_bt1 <-(HPD_upp_bt1-HPD_low_bt1);
HPD_low_bt2 <-Bayes_low_bt2[id3];
HPD_upp_bt2 <-Bayes_upp_bt2[id3];
HPD_len_bt2 <-(HPD_upp_bt2-HPD_low_bt2);

Sym_low_nu <-mean(Bayes_low_nu);
Sym_upp_nu <-mean(Bayes_upp_nu);
Sym_len_nu <-mean(Bayes_len_nu);
Sym_low_bt1 <-mean(Bayes_low_bt1);
Sym_upp_bt1 <-mean(Bayes_upp_bt1);
Sym_len_bt1 <-mean(Bayes_len_bt1);
Sym_low_bt2 <-mean(Bayes_low_bt2);
Sym_upp_bt2 <-mean(Bayes_upp_bt2);
Sym_len_bt2 <-mean(Bayes_len_bt2);

#print(Nu_Sam)

return(c(nu_nw, bt1_nw, bt2_nw, bynums, bybt1ms, bybt2ms,
Sym_low_nu, Sym_upp_nu, Sym_len_nu, Sym_low_bt1, Sym_upp_bt1,
Sym_len_bt1, Sym_low_bt2, Sym_upp_bt2, Sym_len_bt2, HPD_low_nu,
HPD_upp_nu, HPD_len_nu, HPD_low_bt1, HPD_upp_bt1, HPD_len_bt1,
HPD_low_bt2, HPD_upp_bt2, HPD_len_bt2, count)); #Return for parameter estimates

#return(list(BAYNUU,BAYBTT1,BAYBTT2)) #Return for trace plot
}

```

```

#####Ordered Bayes #####
0Bayes_Estimation <- function(oxsp, odst, oe0, oe1, oe2, oe3, orr, orrstr, ott){
  #oa1 = 6; ob1 = 4; oa0 = 3.2; ob0 = 2; oc1 <- (7/9); oc2 <- 1;
  oa1 = 0.00001; ob1 = 2; oa0 = 0.00001; ob0 = 2; oc1 <- 0.00001; oc2 <- 2;
  odens_of_nu <- function(ovnu, oxsm, ods, orst, ori, oTTT){
    # Density of nu from posterior density
    os1=0;os2=0;os3= os4= 0;
    for(ojk in 1:length(oxsm)){
      os1 = os1 + ((ori[ojk]+1)*log(1-(oxsm[ojk]^ovnu)))
      os2 = os2 + ((ori[ojk]+1)*(((oxsm[ojk]^(ovnu))*log(oxsm[ojk]))/
        (1-(oxsm[ojk]^(ovnu)))));
      os3 = os3 + ((ori[ojk]+1)*((oxsm[ojk]^(ovnu))*(log(oxsm[ojk])/
        (1-(oxsm[ojk]^(ovnu))))^(2)));
      os4 = os4 + log((oxsm[ojk]^(ovnu-1))/(1-(oxsm[ojk]^ovnu)))
    }
    oW_nu <- (-os1 - orst*log(1-(oTTT^ovnu)));
    oW_nuD <- (os2 + (orst*(((oTTT^(ovnu))*log(oTTT))/(1-(oTTT^(ovnu))))));
    oW_nuDD <- (os3 + (orst*(((oTTT^(ovnu))*((log(oTTT))/(1-(oTTT^ovnu)))^(2))))));
    ofnval <- (((ovnu^(oa1 + ods - 1))/(ob0 + oW_nu)^(oa0 + oe0 +(2*oe3))))*
      exp(-(ob1*ovnu)+(os4));
    return(ofnval);
  }
  odens_of_nu_mean <- function(ovnu, oxsm, ods, orst, ori, oTTT,ocnst){
    # Density of nu from posterior density
    os1=0;os2=0;os3= os4= 0;
    for(ojk in 1:length(oxsm)){
      os1 = os1 + ((ori[ojk]+1)*log(1-(oxsm[ojk]^ovnu)))
      os2 = os2 + ((ori[ojk]+1)*(((oxsm[ojk]^(ovnu))*log(oxsm[ojk]))/
        (1-(oxsm[ojk]^(ovnu)))));
      os3 = os3 + ((ori[ojk]+1)*((oxsm[ojk]^(ovnu))*(log(oxsm[ojk])/
        (1-(oxsm[ojk]^(ovnu))))^(2)));
      os4 = os4 + log((oxsm[ojk]^(ovnu-1))/(1-(oxsm[ojk]^ovnu)))
    }
    oW_nu <- (-os1 - orst*log(1-(oTTT^ovnu)));
    oW_nuD <- (os2 + (orst*(((oTTT^(ovnu))*log(oTTT))/(1-(oTTT^(ovnu))))));
    oW_nuDD <- (os3 + (orst*(((oTTT^(ovnu))*((log(oTTT))/(1-(oTTT^ovnu)))^(2))))));
    ofnval <- (1/ocnst)*ovnu*(((ovnu^(oa1 + ods - 1))/
      ((ob0 + oW_nu)^(oa0 + oe0 +(2*oe3))))*exp(-(ob1*ovnu)+(os4));
    return(ofnval);
  }
  odens_of_nu_mean_2 <- function(ovnu, oxsm, ods, orst, ori, oTTT,ocnst){
    # Density of nu from posterior density
    os1=0;os2=0;os3 = os4= 0;
    for(ojk in 1:length(oxsm)){
      os1 = os1 + ((ori[ojk]+1)*log(1-(oxsm[ojk]^ovnu)))
      os2 = os2 + ((ori[ojk]+1)*(((oxsm[ojk]^(ovnu))*log(oxsm[ojk]))/
        (1-(oxsm[ojk]^(ovnu)))));
      os3 = os3 + ((ori[ojk]+1)*((oxsm[ojk]^(ovnu))*(log(oxsm[ojk])/
        (1-(oxsm[ojk]^(ovnu))))^(2)));
      os4 = os4 + log((oxsm[ojk]^(ovnu-1))/(1-(oxsm[ojk]^ovnu)))
    }
  }
}

```



```

}
oW_nu <- (-os1 - orst*log(1-(oTTT^ovnu)));
oW_nuD <- (os2 + (orst*((oTTT^(ovnu))*log(oTTT)/(1-(oTTT^(ovnu))))));
oW_nuDD <- (os3 + (orst*((oTTT^(ovnu))*((log(oTTT))/(1-(oTTT^ovnu)))^(2)))));
ofnval <- (1/ocnst)*(ovnu^2)*((ovnu^(oa1 + ods - 1))/
((ob0 + oW_nu)^(oa0 + oe0 + (2*oe3))))*exp(-(ob1*ovnu)+(os4));
return(ofnval);
}
oNN <- 1000;
oconst <- integrate(odens_of_nu, 0, Inf, oxsm=oxsp, ods=odst, orst=orrstr,
ori=orr, oTTT=ott)$value;
omn <- integrate(odens_of_nu_mean, 0, Inf, oxsm=oxsp, ods=odst, orst=orrstr,
ori=orr, oTTT=ott, ocnst = oconst)$value; # Calculating first moment
omn2 <- integrate(odens_of_nu_mean_2, 0, Inf, oxsm=oxsp, ods=odst, orst=orrstr,
ori=orr, oTTT=ott, ocnst = oconst)$value; # Calculating first moment
#cat("\n The mean and variance", c(const,mn,mn2))
ovr <- (omn2-((omn)^2)); # Calculating variance of using density of nu
oa5 <- (omn^2)/ovr; ob5 <- omn/ovr;
# Calculating parameters for the approximating gamma distribution
#cat("\n mn and vr =",a5/b5);
if(oa5<0 || ob5 <0){
  oa5 <- 2;
  ob5 <- 2;
  ocount = 1;
}else{
  ocount = 0;
}
obaynuuu <- numeric(0);
obaybttt1 <- numeric(0);
obaybttt2 <- numeric(0);
obaynuu <- numeric(0);
obaybtt1 <- numeric(0);
obaybtt2 <- numeric(0);
ohfn <- numeric(0);
for(oii in 1:oNN){
  obaynuuu[oii] <- rgamma(1, oa5, ob5);
  ow_nu <- -sum((orr+1)*log(1-(oxsp^obaynuuu[oii])))-
  orrstr*log(1-(ott^obaynuuu[oii])); # Calculating the function W(nu)
  obt0 <- rgamma(1, oa0 + oe0 + 2*oe3, ob0+ow_nu);
  # Generating random beta0 = (beta1 + beta2) from gamma distribution
  obtbybt0 <- rbeta(1, oc1+oe3, oc2+oe3); # Generating random beta
  obaybttt1[oii] <- obt0*obtbybt0; # Generated values of beta1
  obaybttt2[oii] <- obt0-obaybttt1[oii]; # Generated values of beta1
  obaybttt1[oii] <- min(obaybttt1[oii], obaybttt2[oii]);
  obaybttt2[oii] <- max(obaybttt1[oii], obaybttt2[oii]);
  ol5 <- (1/oconst)*odens_of_nu(obaynuuu[oii], oxsp, odst, orrstr, orr, ott);
  ol6 <- dgamma(obaynuuu[oii], oa5, ob5);
  ohfn[oii] <- ((obaybttt1[oii]^(oe1-oe3))*(obaybttt2[oii]^(oe2-oe3)))*(ol5/ol6);
  obaynuu[oii] <- obaynuuu[oii]*ohfn[oii];
  obaybtt1[oii] <- obaybttt1[oii]*ohfn[oii];
}

```

```

    obaybtt2[oii] <- obaybttt2[oii]*ohfn[oii];
}
ow1 <- ohfn/sum(ohfn);
onu_nw <- sum(obaynuu)/sum(ohfn);
obt1_nw <- sum(obaybtt1)/sum(ohfn);
obt2_nw <- sum(obaybtt2)/sum(ohfn);
obynums <- (onu_nw - 11)^2;
obybt1ms <- (obt1_nw - 11)^2;
obybt2ms <- (obt2_nw - 11)^2;
obaynuusort <- sort((obaynuuu));
obaybtt1sort <- sort((obaybttt1));
obaybtt2sort <- sort((obaybttt2));
#####
owisortnu <- ow1[order(obaynuuu)];
owisortbt1 <- ow1[order(obaybttt1)];
owisortbt2 <- ow1[order(obaybttt2)];
onuj1 <- numeric(0);
onuj2 <- numeric(0);
onui <- 1;
obt1j1 <- numeric(0);
obt1j2 <- numeric(0);
obt1i <- 1;
obt2j1 <- numeric(0);
obt2j2 <- numeric(0);
obt2i <- 1;
for(oj1 in 1:(oNN-1)){

    for(oj2 in oj1:(oNN-1)){
        olwnuw1 <- sum(owisortnu[oj1:oj2]);
        oupnuw1 <- sum(owisortnu[oj1:(oj2+1)]);
        if(olwnuw1 <= (1-los) && oupnuw1 > (1-los)){onuj1[onui] = oj1;
        onuj2[onui]=oj2; onui <- onui+1; break;}
    }
}
for(oj1 in 1:(oNN-1)){
    for(oj2 in oj1:(oNN-1)){
        olwbt1w1 <- sum(owisortbt1[oj1:oj2]);
        oupbt1w1 <- sum(owisortbt1[oj1:(oj2+1)]);
        if(olwbt1w1 <= (1-los) && oupbt1w1 > (1-los)){obt1j1[obt1i] = oj1;
        obt1j2[obt1i]=oj2; obt1i <- obt1i + 1; break;}
    }
}
for(oj1 in 1:(oNN-1)){
    for(oj2 in oj1:(oNN-1)){
        olwbt2w1 <- sum(owisortbt2[oj1:oj2]);
        oupbt2w1 <- sum(owisortbt2[oj1:(oj2+1)]);
        if(olwbt2w1 <= (1-los) && oupbt2w1 > (1-los)){obt2j1[obt2i] = oj1;
        obt2j2[obt2i]=oj2; obt2i <- obt2i +1; break;}
    }
}
}

```

```

ocri_nu_lw <- obaynuusort[onuj1]; ocri_nu_up <- obaynuusort[onuj2];
ocri_bt1_lw <- obaybtt1sort[obt1j1]; ocri_bt1_up <- obaybtt1sort[obt1j2];
ocri_bt2_lw <- obaybtt2sort[obt2j1]; ocri_bt2_up <- obaybtt2sort[obt2j2];
odiffcri_nu <- ocri_nu_up - ocri_nu_lw;
odiffcri_bt1 <- ocri_bt1_up - ocri_bt1_lw;
odiffcri_bt2 <- ocri_bt2_up - ocri_bt2_lw;
oind_min_nu <- which(odiffcri_nu==min(odiffcri_nu));
oind_min_bt1 <- which(odiffcri_bt1==min(odiffcri_bt1));
oind_min_bt2 <- which(odiffcri_bt2==min(odiffcri_bt2));
oHPD_low_nu <- ocri_nu_lw[oind_min_nu];
oHPD_upp_nu <- ocri_nu_up[oind_min_nu];
oHPD_len_nu <- (oHPD_upp_nu-oHPD_low_nu)
oHPD_low_bt1 <- ocri_bt1_lw[oind_min_bt1];
oHPD_upp_bt1 <- ocri_bt1_up[oind_min_bt1];
oHPD_len_bt1 <-(oHPD_upp_bt1-oHPD_low_bt1);
oHPD_low_bt2 <- ocri_bt2_lw[oind_min_bt2];
oHPD_upp_bt2 <- ocri_bt2_up[oind_min_bt2];
oHPD_len_bt2 <-(oHPD_upp_bt2-oHPD_low_bt2);

oSym_low_nu <-mean(ocri_nu_lw);
oSym_upp_nu <-mean(ocri_nu_up);
oSym_len_nu <-mean(odiffcri_nu);
oSym_low_bt1 <-mean(ocri_bt1_lw);
oSym_upp_bt1 <-mean(ocri_bt1_up);
oSym_len_bt1 <-mean(odiffcri_bt1);
oSym_low_bt2 <-mean(ocri_bt2_lw);
oSym_upp_bt2 <-mean(ocri_bt2_up);
oSym_len_bt2 <-mean(odiffcri_bt2);

#print(Nu_Sam)
return(c(onu_nw, obt1_nw, obt2_nw, obynums, obybt1ms, obybt2ms,
oSym_low_nu, oSym_upp_nu, oSym_len_nu, oSym_low_bt1, oSym_upp_bt1,
oSym_len_bt1, oSym_low_bt2, oSym_upp_bt2,oSym_len_bt2,oHPD_low_nu,
oHPD_upp_nu, oHPD_len_nu, oHPD_low_bt1, oHPD_upp_bt1, oHPD_len_bt1,
oHPD_low_bt2, oHPD_upp_bt2,oHPD_len_bt2, ocount ));
}
dstr <- k; xsmp <- xsamp[1:dstr]; r <- c(R[1:(dstr-1)],(n-sum(R[1:(dstr-1)]+1)-1));
rstr <- 0;
e0 <- 3;
e1 <- 11;
e2 <- 11;
mlenu <- MLE_Estimation(xstart, xsmp, dstr, r, rstr, T);
mlebt1 <- ((e1)/(-sum((1+r)*log(1-(xsmp^(mlenu))))))*((e0+e1+e2)/(e1+e2));
mlebt2 <- ((e2)/(-sum((1+r)*log(1-(xsmp^(mlenu))))))*((e0+e1+e2)/(e1+e2));
if(e1 < e2){
  mleobt1 <- mlebt1;
  mleobt2 <- mlebt2;
}else{
  mleobt1 <- mleobt2 <- (dstr/(2*sum((r+1)*log(1+(mlenu*xsmp)))));
}
}

```

```

obs_fish_matrix <- function(RRR, rsstr, xx, pr, TT){
  nuu <- pr[1];
  bet1 <- pr[2];
  bet2 <- pr[3];
  ofm <- matrix(0, nrow=3, ncol=3);
  ofm[1,1] <- (e0/((bet1+bet2)^2)) + ((e1)/(bet1^2));
  ofm[1,2] <- ofm[2,1] <- (e0/((bet1+bet2)^2));
  ofm[1,3] <- ofm[3,1] <- ofm[2,3] <- ofm[3,2] <-
  sum((RRR+1)*((xx^nuu)*log(xx))/(1-(xx^nuu))) +
  (rsstr*(((TT^nuu)*log(TT))/(1-(TT^nuu))))
  ofm[2,2] <- (e0/((bet1+bet2)^2)) + ((e2)/(bet2^2));
  ofm[3,3] <- (dstr/(nuu^2)) +
  ((bet1+bet2)*(sum((RRR+1)*(xx^nuu)*((log(xx))/(1-(xx^nuu)))^2)) +
  (rsstr*(TT^nuu)*((log(TT))/(1-(TT^nuu)))^2)))
  + (sum(((xx^nuu)*((log(xx))^2))/(1-(xx^nuu)^2)));
  return(ofm);
}
var_cov <- solve(obs_fish_matrix(r, rstr, xsmp, c(mlenu,mlebt1,mlebt2), T));
var_cov_or <- solve(obs_fish_matrix(r, rstr, xsmp, c(mlenu,mleobt1,mleobt2), T));
print(var_cov);
confid<-function(acon,v,ad){
  #print(v)
  fcon<-1-(ad/2);
  zad<-qnorm(fcon,0,1);
  ac1<-acon-((zad)*(sqrt(v)));
  ac2<-acon+((zad)*(sqrt(v)));
  asconf<-c(ac1,ac2);
  asconf
}
#print(var_cov);
mlen <- mlenu;
mleb1 <- mlebt1;
mleb2 <- mlebt2;
mleon <- mlenu;
mleob1 <- mleobt1;
mleob2 <- mleobt2;
if(var_cov[1,1] < 0 ){var_cov[1,1] <- 0.2}
if(var_cov[2,2] < 0 ){var_cov[2,2] <- 0.2}
if(var_cov[3,3] < 0 ){var_cov[3,3] <- 0.2}
CI_nuu <- confid(mlen,(var_cov[3,3]), los);
CI_bt1 <- confid(mleb1,(var_cov[1,1]), los);
CI_bt2 <- confid(mleb2,(var_cov[2,2]), los);
CI_nuu_width <- CI_nuu[2]-CI_nuu[1];
CI_bt1_width <- CI_bt1[2]-CI_bt1[1];
CI_bt2_width <- CI_bt2[2]-CI_bt2[1];
### Classical OR
CI_nuu_or <- confid(mleon,abs(var_cov_or[3,3]), los);
CI_bt1_or <- confid(mleob1,abs(var_cov_or[1,1]), los);
CI_bt2_or <- confid(mleob2,abs(var_cov_or[2,2]), los);
CI_nuu_or_width <- CI_nuu_or[2]-CI_nuu_or[1];

```

```

CI_bt1_or_width <- CI_bt1_or[2]-CI_bt1_or[1];
CI_bt2_or_width <- CI_bt2_or[2]-CI_bt2_or[1];
Bayes <- Bayes_Estimation(xsamp, dstr, r, rstr, T);
OBayes <- OBayes_Estimation(xsamp, dstr, e0, e1, e2, min(e1, e2), r, rstr, T);
cat(OBayes)
nubayes <- Bayes[1];
bt1bayes <- Bayes[2];
bt2bayes <- Bayes[3];
numsbayes <- Bayes[4];
bt1msbayes <- Bayes[5];
bt2msbayes <- Bayes[6];
CRI_low_nu <- Bayes[7];
CRI_upp_nu <- Bayes[8];
CRI_len_nu <- Bayes[9];
CRI_low_bt1 <- Bayes[10];
CRI_upp_bt1 <- Bayes[11];
CRI_len_bt1 <- Bayes[12];
CRI_low_bt2 <- Bayes[13];
CRI_upp_bt2 <- Bayes[14];
CRI_len_bt2 <- Bayes[15];
HCRI_low_nu <- Bayes[16];
HCRI_upp_nu <- Bayes[17];
HCRI_len_nu <- Bayes[18];
HCRI_low_bt1 <- Bayes[19];
HCRI_upp_bt1 <- Bayes[20];
HCRI_len_bt1 <- Bayes[21];
HCRI_low_bt2 <- Bayes[22];
HCRI_upp_bt2 <- Bayes[23];
HCRI_len_bt2 <- Bayes[24];
countNew <- Bayes[25];
# NUSamp <- Bayes[[1]];
# BT1Samp <- Bayes[[2]];
# BT2Samp <- Bayes[[3]];
# df1 <- cbind(NUSamp, BT1Samp, BT2Samp)
# write.csv(df1, "C:\\Users\\amuly\\OneDrive\\PhD_Full_Folder\\PhD 4th year\\
GPHC_Kumaraswamy\\KumPrograms\\MCMC_Samples_For_Trace\\MyData1.csv", row.names =
FALSE)

```

```

onubayes <- OBayes[1];
obt1bayes <- OBayes[2];
obt2bayes <- OBayes[3];
onnumsbayes <- OBayes[4];
obt1msbayes <- OBayes[5];
obt2msbayes <- OBayes[6];
oCRI_low_nu <- OBayes[7];
oCRI_upp_nu <- OBayes[8];
oCRI_len_nu <- OBayes[9];
oCRI_low_bt1 <- OBayes[10];
oCRI_upp_bt1 <- OBayes[11];

```

```

oCRI_len_bt1 <- OBayes[12];
oCRI_low_bt2 <- OBayes[13];
oCRI_upp_bt2 <- OBayes[14];
oCRI_len_bt2 <- OBayes[15];
oHCRI_low_nu <- OBayes[16];
oHCRI_upp_nu <- OBayes[17];
oHCRI_len_nu <- OBayes[18];
oHCRI_low_bt1 <- OBayes[19];
oHCRI_upp_bt1 <- OBayes[20];
oHCRI_len_bt1 <- OBayes[21];
oHCRI_low_bt2 <- OBayes[22];
oHCRI_upp_bt2 <- OBayes[23];
oHCRI_len_bt2 <- OBayes[24];
##### Classical Estimation Without Order Restriction
# Average Estimate
mean(mlen); mean(mleon);
mean(mleb1); mean(mleob1);
mean(mleb2); mean(mleob2);
#Bayes Average Estimates
mean(nubayes);mean(onubayes);
mean(bt1bayes);mean(obt1bayes);
mean(bt2bayes);mean(obt2bayes);
# Asymptotic confidence intervals
CI_nuu;
CI_bt1;
CI_bt2;
#.....Credible and HPD intervals.....
mean(CRI_low_nu);mean(CRI_upp_nu);
mean(CRI_low_bt1);mean(CRI_upp_bt1);
mean(CRI_low_bt2);mean(CRI_upp_bt2);
mean(HCRI_low_nu);mean(HCRI_upp_nu);
mean(HCRI_low_bt1);mean(HCRI_upp_bt1);
mean(HCRI_low_bt2);mean(HCRI_upp_bt2);
#####
mean(CI_nuu_width); mean(CRI_len_nu); mean(HCRI_len_nu);
mean(CI_nuu_or_width); mean(oCRI_len_nu); mean(oHCRI_len_nu);

mean(CI_bt1_width); mean(CRI_len_bt1); mean(HCRI_len_bt1);
mean(CI_bt1_or_width); mean(oCRI_len_bt1); mean(oHCRI_len_bt1);

mean(CI_bt2_width); mean(CRI_len_bt2); mean(HCRI_len_bt2);
mean(CI_bt2_or_width); mean(oCRI_len_bt2); mean(oHCRI_len_bt2);

###Real Data II (Partially Observed Competing Risk(Kumaraswamy Distribution))###
rm(list=ls(all=TRUE));
rm(list=ls(all=TRUE));
rm(list=ls(all=TRUE));
library(nleqslv);
xsamp <- c(0.0222, 0.0333, 0.0778, 0.0889, 0.1000, 0.1222, 0.1444, 0.1556, 0.1667,

```

```

0.1778, 0.1778, 0.2000, 0.2000, 0.2111, 0.2222, 0.2667, 0.2889, 0.3000, 0.3111,
0.3333, 0.3778, 0.4000, 0.4333, 0.4333, 0.4444, 0.4667, 0.5333, 0.5444);
n <- 37; m <- 30; k=26;
R <- c((n-m), rep(0,(m-1)));
T=0.6;
los <- 0.05;
xstart <- 1;
MLE_Estimation <- function(ings, ysp, dst, rr, rrstr, tt){
  # MLE for nu using nleqslv Package
  lik_eqn <- function(nu){
    lkeq <-
      (dst/nu)+(sum((log(ysp))/(1-(ysp^nu))))-(dst*((sum((rr+1)*((ysp^nu)*log(ysp))/
      (1-(ysp^nu))))+(rrstr*((tt^nu)*log(tt))/((1-(tt^nu)))))/(-sum((rr+1)*
      log(1-(ysp^nu)))-(rrstr*log(1-(tt^nu))));
    return(lkeq);
  }
  mlnu <- nleqslv(ings, lik_eqn);
  return(mlnu$x);
}

Bayes_Estimation <- function(xsp, dst, rr, rrstr, tt){
  a1 = 0.00001; b1 = 2; a0 = 0.00001; b0 = 2; c1 <- 0.00001; c2 <- 2;
  dens_of_nu <- function(vnu, xsm, ds, rst, ri, TTT){
    # Density of nu from posterior density
    s1=0;s2=0;s3= s4= 0;
    for(jk in 1:length(xsm)){
      s1 = s1 + ((ri[jk]+1)*log(1-(xsm[jk]^vnu)))
      s2 = s2 + ((ri[jk]+1)*(((xsm[jk]^(vnu))*log(xsm[jk]))/(1-(xsm[jk]^(vnu)))));
      s3 = s3 + ((ri[jk]+1)*((xsm[jk]^(vnu))*(log(xsm[jk]))/(1-(xsm[jk]^(vnu))))^(2));
      s4 = s4 + log((xsm[jk]^(vnu-1))/(1-(xsm[jk]^vnu)))
    }
    W_nu <- (-s1 - rst*log(1-(TTT^vnu)));
    W_nuD <- (s2 + (rst*((TTT^vnu)*log(TTT))/(1-(TTT^vnu))));
    W_nuDD <- (s3 + (rst*((TTT^vnu)*((log(TTT))/(1-(TTT^vnu)))^(2))))
    #fnval <- (((vnu^(a1 + ds - 1))/((b0 + W_nu)^(a0 +
    ds)))*exp(-(b1-sum(log(xsm))*vnu))/exp(-vnu));
    fnval <- (((vnu^(a1 + ds - 1))/((b0 + W_nu)^(a0 + ds)))*exp(-(b1*vnu)+(s4)))
    #/exp(-vnu);
    #fDDVal <- (sum(((xsm^vnu)*((log(xsm^vnu))^2))/((1-(xsm^vnu))^2)))/(a1+ds-1)
    return(fnval);
    #-((a1+ds-1)/(vnu^2))-((a0+ds)*(((b0+W_nu)*(W_nuDD))-(W_nuD^2))/((b0+W_nu)^2))
  }
  dens_of_nu_mean <- function(vnu, xsm, ds, rst, ri, TTT,cnst){
    # Density of nu from posterior density
    s1=0;s2=0;s3= s4= 0;
    for(jk in 1:length(xsm)){
      s1 = s1 + ((ri[jk]+1)*log(1-(xsm[jk]^vnu)))
      s2 = s2 + ((ri[jk]+1)*(((xsm[jk]^(vnu))*log(xsm[jk]))/(1-(xsm[jk]^(vnu)))));
      s3 = s3 + ((ri[jk]+1)*((xsm[jk]^(vnu))*(log(xsm[jk]))/(1-(xsm[jk]^(vnu))))^(2));
      s4 = s4 + log((xsm[jk]^(vnu-1))/(1-(xsm[jk]^vnu)))
    }
  }
}

```

```

}
W_nu <- (-s1 - rst*log(1-(TTT^vnu)));
W_nuD <- (s2 + (rst*(((TTT^(vnu))*log(TTT))/(1-(TTT^(vnu))))));
W_nuDD <- (s3 + (rst*(((TTT^(vnu))*((log(TTT))/(1-(TTT^vnu)))^(2))))))
#fnval <- (((vnu^(a1 + ds - 1))/((b0 + W_nu)^(a0 +
ds))) * exp(-(b1-sum(log(xsm)) * vnu)) / exp(-vnu));
fnval <- (1/cnst) * vnu * (((vnu^(a1 + ds - 1))/((b0 + W_nu)^(a0 +
ds))) * exp(-(b1*vnu)+(s4))) #/exp(-vnu);
#fDDVal <- (sum(((xsm^vnu)*((log(xsm^vnu))^2))/((1-(xsm^vnu))^2)))/(a1+ds-1)
return(fnval);
#-((a1+ds-1)/(vnu^2))-((a0+ds)*(((b0+W_nu)*(W_nuDD))-(W_nuD^2))/((b0+W_nu)^2))
}
dens_of_nu_mean_2 <- function(vnu, xsm, ds, rst, ri, TTT, cnst){
# Density of nu from posterior density
s1=0;s2=0;s3= s4= 0;
for(jk in 1:length(xsm)){
s1 = s1 + ((ri[jk]+1)*log(1-(xsm[jk]^vnu)))
s2 = s2 + ((ri[jk]+1)*(((xsm[jk]^(vnu))*log(xsm[jk]))/(1-(xsm[jk]^(vnu)))));
s3 = s3 + ((ri[jk]+1)*((xsm[jk]^(vnu))*log(xsm[jk])/(1-(xsm[jk]^(vnu))))^(2))
s4 = s4 + log((xsm[jk]^(vnu-1))/(1-(xsm[jk]^vnu)))
}
W_nu <- (-s1 - rst*log(1-(TTT^vnu)));
W_nuD <- (s2 + (rst*(((TTT^(vnu))*log(TTT))/(1-(TTT^(vnu))))));
W_nuDD <- (s3 + (rst*(((TTT^(vnu))*((log(TTT))/(1-(TTT^vnu)))^(2))))))
#fnval <- (((vnu^(a1 + ds - 1))/((b0 + W_nu)^(a0 +
ds))) * exp(-(b1-sum(log(xsm)) * vnu)) / exp(-vnu));
fnval <- (1/cnst) * (vnu^2) * (((vnu^(a1 + ds - 1))/((b0 + W_nu)^(a0 +
ds))) * exp(-(b1*vnu)+(s4))) #/exp(-vnu);
#fDDVal <- (sum(((xsm^vnu)*((log(xsm^vnu))^2))/((1-(xsm^vnu))^2)))/(a1+ds-1)
return(fnval);
#-((a1+ds-1)/(vnu^2))-((a0+ds)*(((b0+W_nu)*(W_nuDD))-(W_nuD^2))/((b0+W_nu)^2))
}
const <- integrate(dens_of_nu, 0, Inf, xsm=xsp, ds=dst, rst=rrstr, ri=rr,
TTT=tt)$value;
mn <- integrate(dens_of_nu_mean, 0, Inf, xsm=xsp, ds=dst, rst=rrstr, ri=rr,
TTT=tt, cnst = const)$value; # Calculating first moment
mn2 <- integrate(dens_of_nu_mean_2, 0, Inf, xsm=xsp, ds=dst, rst=rrstr, ri=rr,
TTT=tt, cnst = const)$value; # Calculating first moment
#cat("\n The mean and variance", c(const,mn,mn2))
vr <- (mn2-((mn)^2)); # Calculating variance of using density of nu
a5 <- (mn^2)/vr; b5 <- mn/vr;
# Calculating parameters for the approximating gamma distribution
#cat("\n mn and vr =",a5/b5);
if(a5<0 || b5 <0){
a5 <- 1.5;
b5 <- 1;
count = 1;
}else{
count = 0;
}
}

```



```

#cat("\n a5 and b5 =",c(a5,b5));
baynuu <- numeric(0);
baybtt1 <- numeric(0);
baybtt2 <- numeric(0);
for(ii in 1:4000){
  baynuu[ii] <- rgamma(1, a5, b5);
  #print(baynuu[ii]);
  w_nu <- -sum((rr+1)*log(1-(xsp^baynuu[ii])))- rrstr*log(1-(tt^baynuu[ii]));
  # Calculating the function W(nu)
  bt0 <- rgamma(1, a0+dst,b0+w_nu);
  # Generating random beta0 = (beta1 + beta2) from gamma distribution
  btbybt0 <- rbeta(1, c1+e1, c2+e2); # Generating random beta
  baybtt1[ii] <- bt0*btbybt0; # Generated values of beta1
  baybtt2[ii] <- bt0-baybtt1[ii]; # Generated values of beta1
}
BAYNUU <- baynuu[(1001:4000)];
BAYBTT1 <- baybtt1[(1001:4000)];
BAYBTT2 <- baybtt2[(1001:4000)];
nu_nw <- mean((baynuu[(1001:4000)]));
bt1_nw <- mean((baybtt1[(1001:4000)]));
bt2_nw <- mean((baybtt2[(1001:4000)]));
bynums <- (nu_nw - 11)^2;
bybt1ms <- (bt1_nw - 11)^2;
bybt2ms <- (bt2_nw - 11)^2;
baynuusort <- sort((baynuu[(1001:4000)]));
baybtt1sort <- sort((baybtt1[(1001:4000)]));
baybtt2sort <- sort((baybtt2[(1001:4000)]));
Bayes_low_nu <- Bayes_upp_nu <- Bayes_len_nu <- numeric(0);
Bayes_low_bt1 <- Bayes_upp_bt1 <- Bayes_len_bt1 <- numeric(0);
Bayes_low_bt2 <- Bayes_upp_bt2 <- Bayes_len_bt2 <- numeric(0);
for(ir in 1:(length(baynuusort)*los)){
  Bayes_low_nu[ir] <- (baynuusort[ir])
  Bayes_upp_nu[ir] <- (baynuusort[ir+((length(baynuusort)*(1-los)))]))
  Bayes_len_nu[ir] <- (Bayes_upp_nu[ir]- Bayes_low_nu[ir])
  Bayes_low_bt1[ir] <- (baybtt1sort[ir])
  Bayes_upp_bt1[ir] <- (baybtt1sort[ir+((length(baynuusort)*(1-los)))]))
  Bayes_len_bt1[ir] <- (Bayes_upp_bt1[ir]- Bayes_low_bt1[ir]);
  Bayes_low_bt2[ir] <- (baybtt2sort[ir])
  Bayes_upp_bt2[ir] <- (baybtt2sort[ir+((length(baynuusort)*(1-los)))]))
  Bayes_len_bt2[ir] <- (Bayes_upp_bt2[ir]- Bayes_low_bt2[ir]);
}
#print(baynuusort)
id1 <- which(Bayes_len_nu==min(Bayes_len_nu));
id2 <- which(Bayes_len_bt1==min(Bayes_len_bt1));
id3 <- which(Bayes_len_bt2==min(Bayes_len_bt2));
HPD_low_nu <-Bayes_low_nu[id1];
HPD_upp_nu <-Bayes_upp_nu[id1];
HPD_len_nu <-(HPD_upp_nu-HPD_low_nu)
HPD_low_bt1 <-Bayes_low_bt1[id2];
HPD_upp_bt1 <-Bayes_upp_bt1[id2];

```

```

HPD_len_bt1 <-(HPD_upp_bt1-HPD_low_bt1);
HPD_low_bt2 <-Bayes_low_bt2[id3];
HPD_upp_bt2 <-Bayes_upp_bt2[id3];
HPD_len_bt2 <-(HPD_upp_bt2-HPD_low_bt2);

Sym_low_nu <-mean(Bayes_low_nu);
Sym_upp_nu <-mean(Bayes_upp_nu);
Sym_len_nu <-mean(Bayes_len_nu);
Sym_low_bt1 <-mean(Bayes_low_bt1);
Sym_upp_bt1 <-mean(Bayes_upp_bt1);
Sym_len_bt1 <-mean(Bayes_len_bt1);
Sym_low_bt2 <-mean(Bayes_low_bt2);
Sym_upp_bt2 <-mean(Bayes_upp_bt2);
Sym_len_bt2 <-mean(Bayes_len_bt2);

#print(Nu_Sam)
return(c(nu_nw, bt1_nw, bt2_nw, bynums, bybt1ms, bybt2ms, Sym_low_nu,
Sym_upp_nu, Sym_len_nu, Sym_low_bt1, Sym_upp_bt1, Sym_len_bt1,
Sym_low_bt2, Sym_upp_bt2, Sym_len_bt2, HPD_low_nu, HPD_upp_nu,
HPD_len_nu, HPD_low_bt1, HPD_upp_bt1, HPD_len_bt1, HPD_low_bt2,
HPD_upp_bt2, HPD_len_bt2, count ));
#return(list(BAYNUU,BAYBTT1,BAYBTT2)) #Return for trace plot
}
#####Ordered Bayes #####
OBayes_Estimation <- function(oxsp, odst, oe0, oe1, oe2, oe3, orr, orrstr, ott){
  #oa1 = 6; ob1 = 4; oa0 = 3.2; ob0 = 2; oc1 <- (7/9); oc2 <- 1;
  oa1 = 0.00001; ob1 = 2; oa0 = 0.00001; ob0 = 2; oc1 <- 0.00001; oc2 <- 2;
  odens_of_nu <- function(ovnu, oxsm, ods, orst, ori, oTTT){
    # Density of nu from posterior density
    os1=0;os2=0;os3= os4= 0;
    for(ojk in 1:length(oxsm)){
      os1 = os1 + ((ori[ojk]+1)*log(1-(oxsm[ojk]^ovnu)))
      os2 = os2 +
        ((ori[ojk]+1)*(((oxsm[ojk]^(ovnu))*log(oxsm[ojk]))/(1-(oxsm[ojk]^(ovnu)))));
      os3 = os3 +
        ((ori[ojk]+1)*((oxsm[ojk]^(ovnu))*(log(oxsm[ojk]))/(1-(oxsm[ojk]^(ovnu))))^(2));
      os4 = os4 + log((oxsm[ojk]^(ovnu-1))/(1-(oxsm[ojk]^ovnu)))
    }
    oW_nu <- (-os1 - orst*log(1-(oTTT^ovnu)));
    oW_nuD <- (os2 + (orst*(((oTTT^(ovnu))*log(oTTT))/(1-(oTTT^(ovnu))))));
    oW_nuDd <- (os3 + (orst*(((oTTT^(ovnu))*((log(oTTT))/(1-(oTTT^ovnu))))^(2))));
    ofnval <- (((ovnu^(oa1 + ods - 1)))/((ob0 + oW_nu)^(oa0 + oe0+
    (2*oe3))))*exp(-(ob1*ovnu)+(os4));
    return(ofnval);
  }
}
odens_of_nu_mean <- function(ovnu, oxsm, ods, orst, ori, oTTT,ocnst){
  # Density of nu from posterior density
  os1=0;os2=0;os3= os4= 0;
  for(ojk in 1:length(oxsm)){
    os1 = os1 + ((ori[ojk]+1)*log(1-(oxsm[ojk]^ovnu)))
  }
}

```

```

    os2 = os2 +
      ((ori[ojk]+1)*(((oxsm[ojk]^(ovnu))*log(oxsm[ojk]))/(1-(oxsm[ojk]^(ovnu)))));
    os3 = os3 +
      ((ori[ojk]+1)*((oxsm[ojk]^(ovnu))*(log(oxsm[ojk]))/(1-(oxsm[ojk]^(ovnu))))^(2));
    os4 = os4 + log((oxsm[ojk]^(ovnu-1))/(1-(oxsm[ojk]^(ovnu))))
  }
  oW_nu <- (-os1 - orst*log(1-(oTTT^ovnu)));
  oW_nuD <- (os2 + (orst*(((oTTT^(ovnu))*log(oTTT))/(1-(oTTT^(ovnu))))));
  oW_nuDD <- (os3 + (orst*(((oTTT^(ovnu))*((log(oTTT))/(1-(oTTT^ovnu)))^(2))))));
  ofnval <- (1/ocnst)*ovnu*(((ovnu^(oa1 + ods - 1))/((ob0 + oW_nu)^(oa0 + oe0+
    (2*oe3)))))*exp(-(ob1*ovnu)+(os4));
  return(ofnval);
}

odens_of_nu_mean_2 <- function(ovnu, oxsm, ods, orst, ori, oTTT,ocnst){
  # Density of nu from posterior density
  os1=0;os2=0;os3 = os4= 0;
  for(ojk in 1:length(oxsm)){
    os1 = os1 + ((ori[ojk]+1)*log(1-(oxsm[ojk]^(ovnu))))
    os2 = os2 +
      ((ori[ojk]+1)*(((oxsm[ojk]^(ovnu))*log(oxsm[ojk]))/(1-(oxsm[ojk]^(ovnu)))));
    os3 = os3 +
      ((ori[ojk]+1)*((oxsm[ojk]^(ovnu))*(log(oxsm[ojk]))/(1-(oxsm[ojk]^(ovnu))))^(2));
    os4 = os4 + log((oxsm[ojk]^(ovnu-1))/(1-(oxsm[ojk]^(ovnu))))
  }
  oW_nu <- (-os1 - orst*log(1-(oTTT^ovnu)));
  oW_nuD <- (os2 + (orst*(((oTTT^(ovnu))*log(oTTT))/(1-(oTTT^(ovnu))))));
  oW_nuDD <- (os3 + (orst*(((oTTT^(ovnu))*((log(oTTT))/(1-(oTTT^ovnu)))^(2))))));
  ofnval <- (1/ocnst)*(ovnu^2)*(((ovnu^(oa1 + ods - 1))/
    ((ob0 + oW_nu)^(oa0 + oe0 + (2*oe3)))))*exp(-(ob1*ovnu)+(os4));
  return(ofnval);
}

oNN <- 1000;
oconst <- integrate(odens_of_nu, 0, Inf, oxsm=oxsp, ods=odst,
  orst=orrstr, ori=orr, oTTT=ott)$value;
omn <- integrate(odens_of_nu_mean, 0, Inf, oxsm=oxsp, ods=odst,
  orst=orrstr, ori=orr, oTTT=ott, ocnst = oconst)$value; # Calculating first moment
omn2 <- integrate(odens_of_nu_mean_2, 0, Inf, oxsm=oxsp, ods=odst,
  orst=orrstr, ori=orr, oTTT=ott, ocnst = oconst)$value; # Calculating first moment
#cat("\n The mean and variance", c(const,mn,mn2))
ovr <- (omn2-((omn)^2)); # Calculating variance of using density of nu
oa5 <- (omn^2)/ovr; ob5 <- omn/ovr;
# Calculating parameters for the approximating gamma distribution
#cat("\n mn and vr =",a5/b5);
if(oa5<0 || ob5 <0){
  oa5 <- 2;
  ob5 <- 2;
  ocount = 1;
}else{
  ocount = 0;
}
}

```

```

obaynuuu <- numeric(0);
obaybttt1 <- numeric(0);
obaybttt2 <- numeric(0);
obaynuu <- numeric(0);
obaybtt1 <- numeric(0);
obaybtt2 <- numeric(0);
ohfn <- numeric(0);
for(oii in 1:oNN){
  obaynuuu[oii] <- rgamma(1, oa5, ob5);
  ow_nu <- -sum((orr+1)*log(1-(oxsp^obaynuuu[oii]))) - orrstr*log(1-(ott^obaynuuu[oii]));
  # Calculating the function W(nu)
  obt0 <- rgamma(1, oa0 + oe0 + 2*oe3, ob0+ow_nu);
  # Generating random beta0 = (beta1 + beta2) from gamma distribution
  obtbybt0 <- rbeta(1, oc1+oe3, oc2+oe3); # Generating random beta
  obaybttt1[oii] <- obt0*obtbybt0; # Generated values of beta1
  obaybttt2[oii] <- obt0-obaybttt1[oii]; # Generated values of beta1
  obaybttt1[oii] <- min(obaybttt1[oii], obaybttt2[oii]);
  obaybttt2[oii] <- max(obaybttt1[oii], obaybttt2[oii]);
  ol5 <- (1/oconst)*odens_of_nu(obaynuuu[oii], oxsp, odst, orrstr, orr, ott);
  ol6 <- dgamma(obaynuuu[oii], oa5, ob5);
  ohfn[oii] <- ((obaybttt1[oii]^(oe1-oe3))*(obaybttt2[oii]^(oe2-oe3)))*(ol5/ol6);
  obaynuu[oii] <- obaynuuu[oii]*ohfn[oii];
  obaybtt1[oii] <- obaybttt1[oii]*ohfn[oii];
  obaybtt2[oii] <- obaybttt2[oii]*ohfn[oii];
}
ow1 <- ohfn/sum(ohfn);
onu_nw <- sum(obaynuu)/sum(ohfn);
obt1_nw <- sum(obaybtt1)/sum(ohfn);
obt2_nw <- sum(obaybtt2)/sum(ohfn);
obynums <- (onu_nw - 11)^2;
obybt1ms <- (obt1_nw - 11)^2;
obybt2ms <- (obt2_nw - 11)^2;
obaynuusort <- sort((obaynuuu));
obaybtt1sort <- sort((obaybttt1));
obaybtt2sort <- sort((obaybttt2));
#####
owisortnu <- ow1[order(obaynuuu)];
owisortbt1 <- ow1[order(obaybttt1)];
owisortbt2 <- ow1[order(obaybttt2)];
onuj1 <- numeric(0);
onuj2 <- numeric(0);
onui <- 1;
obt1j1 <- numeric(0);
obt1j2 <- numeric(0);
obt1i <- 1;
obt2j1 <- numeric(0);
obt2j2 <- numeric(0);
obt2i <- 1;
for(oj1 in 1:(oNN-1)){

```

```

    for(oj2 in oj1:(oNN-1)){
      olwnuw1 <- sum(owisortnu[oj1:oj2]);
      oupnuw1 <- sum(owisortnu[oj1:(oj2+1)]);
      if(olwnuw1 <= (1-los) && oupnuw1 > (1-los)){onuj1[onui] =
        oj1; onuj2[onui]=oj2; onui <- onui+1; break;}
    }
  }
for(oj1 in 1:(oNN-1)){
  for(oj2 in oj1:(oNN-1)){
    olwbt1w1 <- sum(owisortbt1[oj1:oj2]);
    oupbt1w1 <- sum(owisortbt1[oj1:(oj2+1)]);
    if(olwbt1w1 <= (1-los) && oupbt1w1 > (1-los)){obt1j1[obt1i] =
      oj1; obt1j2[obt1i]=oj2; obt1i <- obt1i + 1; break;}
  }
}
for(oj1 in 1:(oNN-1)){
  for(oj2 in oj1:(oNN-1)){
    olwbt2w1 <- sum(owisortbt2[oj1:oj2]);
    oupbt2w1 <- sum(owisortbt2[oj1:(oj2+1)]);
    if(olwbt2w1 <= (1-los) && oupbt2w1 > (1-los)){obt2j1[obt2i] =
      oj1; obt2j2[obt2i]=oj2; obt2i <- obt2i +1; break;}
  }
}
ocri_nu_lw <- obaynuusort[onuj1]; ocri_nu_up <- obaynuusort[onuj2];
ocri_bt1_lw <- obaybtt1sort[obt1j1]; ocri_bt1_up <- obaybtt1sort[obt1j2];
ocri_bt2_lw <- obaybtt2sort[obt2j1]; ocri_bt2_up <- obaybtt2sort[obt2j2];
odiffcri_nu <- ocri_nu_up - ocri_nu_lw;
odiffcri_bt1 <- ocri_bt1_up - ocri_bt1_lw;
odiffcri_bt2 <- ocri_bt2_up - ocri_bt2_lw;
oind_min_nu <- which(odiffcri_nu==min(odiffcri_nu));
oind_min_bt1 <- which(odiffcri_bt1==min(odiffcri_bt1));
oind_min_bt2 <- which(odiffcri_bt2==min(odiffcri_bt2));
oHPD_low_nu <- ocri_nu_lw[oind_min_nu];
oHPD_upp_nu <- ocri_nu_up[oind_min_nu];
oHPD_len_nu <- (oHPD_upp_nu-oHPD_low_nu)
oHPD_low_bt1 <- ocri_bt1_lw[oind_min_bt1];
oHPD_upp_bt1 <- ocri_bt1_up[oind_min_bt1];
oHPD_len_bt1 <-(oHPD_upp_bt1-oHPD_low_bt1);
oHPD_low_bt2 <- ocri_bt2_lw[oind_min_bt2];
oHPD_upp_bt2 <- ocri_bt2_up[oind_min_bt2];
oHPD_len_bt2 <-(oHPD_upp_bt2-oHPD_low_bt2);

oSym_low_nu <-mean(ocri_nu_lw);
oSym_upp_nu <-mean(ocri_nu_up);
oSym_len_nu <-mean(odiffcri_nu);
oSym_low_bt1 <-mean(ocri_bt1_lw);
oSym_upp_bt1 <-mean(ocri_bt1_up);
oSym_len_bt1 <-mean(odiffcri_bt1);
oSym_low_bt2 <-mean(ocri_bt2_lw);
oSym_upp_bt2 <-mean(ocri_bt2_up);

```

```

oSym_len_bt2 <-mean(odiffcri_bt2);

#print(Nu_Sam)
return(c(onu_nw, obt1_nw, obt2_nw, obynums, obybt1ms, obybt2ms,
oSym_low_nu, oSym_upp_nu, oSym_len_nu, oSym_low_bt1, oSym_upp_bt1,
oSym_len_bt1, oSym_low_bt2, oSym_upp_bt2, oSym_len_bt2, oHPD_low_nu,
oHPD_upp_nu, oHPD_len_nu, oHPD_low_bt1, oHPD_upp_bt1, oHPD_len_bt1,
oHPD_low_bt2, oHPD_upp_bt2,oHPD_len_bt2, ocount ));
}
#dstr <- k; xsamp <- xsamp[1:dstr]; r <- c(R[1:(dstr-1)],(n-sum(R[1:(dstr-1)]+1)-1));
rstr <- 0;
dstr <- length(which(xsamp < T)); xsamp <- xsamp[1:dstr]; r <- R[1:dstr];
rstr <- (n-sum(R[1:dstr]+1));
e0 <- 3;
e1 <- 13;
e2 <- 12;
mlenu <- MLE_Estimation(xstart, xsamp, dstr, r, rstr, T);
mlebt1 <- ((e1)/(-sum((1+r)*log(1-(xsamp^(mlenu))))-
(rstr*log(1-(T^mlenu)))))*((e0+e1+e2)/(e1+e2));
mlebt2 <- ((e2)/(-sum((1+r)*log(1-(xsamp^(mlenu))))-
(rstr*log(1-(T^mlenu)))))*((e0+e1+e2)/(e1+e2));
if(e1 < e2){
  mleobt1 <- mlebt1;
  mleobt2 <- mlebt2;
}else{
  mleobt1 <- mleobt2 <- (dstr/(2*(sum((r+1)*log(1+(mlenu*xsmp))) +
(rstr*log(1+(mlenu*T))))));
}
}
obs_fish_matrix <- function(RRR, rsstr, xx, pr, TT){
  nuu <- pr[1];
  bet1 <- pr[2];
  bet2 <- pr[3];
  ofm <- matrix(0, nrow=3, ncol=3);
  ofm[1,1] <- (e0/((bet1+bet2)^2)) + ((e1)/(bet1^2));
  ofm[1,2] <- ofm[2,1] <- (e0/((bet1+bet2)^2));
  ofm[1,3] <- ofm[3,1] <- ofm[2,3] <- ofm[3,2] <-
sum((RRR+1)*((xx^nuu)*log(xx))/(1-(xx^nuu))) +
(rsstr*(((TT^nuu)*log(TT))/(1-(TT^nuu))))
  ofm[2,2] <- (e0/((bet1+bet2)^2)) + ((e2)/(bet2^2));
  ofm[3,3] <- (dstr/(nuu^2)) +
((bet1+bet2)*(sum((RRR+1)*(xx^nuu)*((log(xx))/(1-(xx^nuu)))^2))) +
(rsstr*(TT^nuu)*((log(TT))/(1-(TT^nuu)))^2))) +
(sum(((xx^nuu)*((log(xx))^2))/((1-(xx^nuu))^2)));
  return(ofm);
}
var_cov <- solve(obs_fish_matrix(r, rstr, xsamp, c(mlenu,mlebt1,mlebt2), T));
var_cov_or <- solve(obs_fish_matrix(r, rstr, xsamp, c(mlenu,mleobt1,mleobt2), T));
print(var_cov);
confid<-function(acon,v,ad){
  #print(v)

```

```

    fcon<-1-(ad/2);
    zad<-qnorm(fcon,0,1);
    ac1<-acon-((zad)*(sqrt(v)));
    ac2<-acon+((zad)*(sqrt(v)));
    asconf<-c(ac1,ac2);
    asconf
}
#print(var_cov);
mlen <- mlenu;
mleb1 <- mlebt1;
mleb2 <- mlebt2;
mleon <- mlenu;
mleob1 <- mleobt1;
mleob2 <- mleobt2;
if(var_cov[1,1] <0 ){var_cov[1,1] <- 0.2}
if(var_cov[2,2] <0 ){var_cov[2,2] <- 0.2}
if(var_cov[3,3] <0 ){var_cov[3,3] <- 0.2}
CI_nuu <- confid(mlen,(var_cov[3,3]), los);
CI_bt1 <- confid(mleb1,(var_cov[1,1]), los);
CI_bt2 <- confid(mleb2,(var_cov[2,2]), los);
CI_nuu_width <- CI_nuu[2]-CI_nuu[1];
CI_bt1_width <- CI_bt1[2]-CI_bt1[1];
CI_bt2_width <- CI_bt2[2]-CI_bt2[1];
### Classical OR
CI_nuu_or <- confid(mleon,abs(var_cov_or[3,3]), los);
CI_bt1_or <- confid(mleob1,abs(var_cov_or[1,1]), los);
CI_bt2_or <- confid(mleob2,abs(var_cov_or[2,2]), los);
CI_nuu_or_width <- CI_nuu_or[2]-CI_nuu_or[1];
CI_bt1_or_width <- CI_bt1_or[2]-CI_bt1_or[1];
CI_bt2_or_width <- CI_bt2_or[2]-CI_bt2_or[1];
Bayes <- Bayes_Estimation(xsmpl, dstr, r, rstr, T);
OBayes <- OBayes_Estimation(xsmpl, dstr, e0, e1, e2, min(e1, e2), r, rstr, T);
#cat(OBayes)
nubayes <- Bayes[1];
bt1bayes <- Bayes[2];
bt2bayes <- Bayes[3];
numsbayes <- Bayes[4];
bt1msbayes <- Bayes[5];
bt2msbayes <- Bayes[6];
CRI_low_nu <- Bayes[7];
CRI_upp_nu <- Bayes[8];
CRI_len_nu <- Bayes[9];
CRI_low_bt1 <- Bayes[10];
CRI_upp_bt1 <- Bayes[11];
CRI_len_bt1 <- Bayes[12];
CRI_low_bt2 <- Bayes[13];
CRI_upp_bt2 <- Bayes[14];
CRI_len_bt2 <- Bayes[15];
HCRI_low_nu <- Bayes[16];
HCRI_upp_nu <- Bayes[17];

```

```

HCRI_len_nu <- Bayes[18];
HCRI_low_bt1 <- Bayes[19];
HCRI_upp_bt1 <- Bayes[20];
HCRI_len_bt1 <- Bayes[21];
HCRI_low_bt2 <- Bayes[22];
HCRI_upp_bt2 <- Bayes[23];
HCRI_len_bt2 <- Bayes[24];
countNew <- Bayes[25];
# NUSamp <- Bayes[[1]];
# BT1Samp <- Bayes[[2]];
# BT2Samp <- Bayes[[3]];
# df1 <- cbind(NUSamp, BT1Samp, BT2Samp)
# write.csv(df1,"C:\\Users\\amuly\\OneDrive\\PhD_Full_Folder\\PhD 4th year\\
GPHC_Kumaraswamy\\KumPrograms\\MCMC_Samples_For_Trace\\
MyData2_1.csv", row.names = FALSE)

onubayes <- OBayes[1];
obt1bayes <- OBayes[2];
obt2bayes <- OBayes[3];
onumsbayes <- OBayes[4];
obt1msbayes <- OBayes[5];
obt2msbayes <- OBayes[6];
oCRI_low_nu <- OBayes[7];
oCRI_upp_nu <- OBayes[8];
oCRI_len_nu <- OBayes[9];
oCRI_low_bt1 <- OBayes[10];
oCRI_upp_bt1 <- OBayes[11];
oCRI_len_bt1 <- OBayes[12];
oCRI_low_bt2 <- OBayes[13];
oCRI_upp_bt2 <- OBayes[14];
oCRI_len_bt2 <- OBayes[15];
oHCRI_low_nu <- OBayes[16];
oHCRI_upp_nu <- OBayes[17];
oHCRI_len_nu <- OBayes[18];
oHCRI_low_bt1 <- OBayes[19];
oHCRI_upp_bt1 <- OBayes[20];
oHCRI_len_bt1 <- OBayes[21];
oHCRI_low_bt2 <- OBayes[22];
oHCRI_upp_bt2 <- OBayes[23];
oHCRI_len_bt2 <- OBayes[24];
##### Classical Estimation Without Order Restriction
# Average Estimate
mean(mlen); mean(mleon);
mean(mleb1); mean(mleob1);
mean(mleb2); mean(mleob2);
#Bayes Average Estimates
mean(nubayes);mean(onubayes);
mean(bt1bayes);mean(obt1bayes);
mean(bt2bayes);mean(obt2bayes);
# Asymptotic confidence intervals

```



```

CI_nuu;
CI_bt1;
CI_bt2;
#.....Credible and HPD intervals.....
mean(CRI_low_nu);mean(CRI_upp_nu);
mean(CRI_low_bt1);mean(CRI_upp_bt1);
mean(CRI_low_bt2);mean(CRI_upp_bt2);
mean(HCRI_low_nu);mean(HCRI_upp_nu);
mean(HCRI_low_bt1);mean(HCRI_upp_bt1);
mean(HCRI_low_bt2);mean(HCRI_upp_bt2);
#####
mean(CI_nuu_width); mean(CRI_len_nu); mean(HCRI_len_nu);
mean(CI_nuu_or_width); mean(oCRI_len_nu); mean(oHCRI_len_nu);

mean(CI_bt1_width); mean(CRI_len_bt1); mean(HCRI_len_bt1);
mean(CI_bt1_or_width); mean(oCRI_len_bt1); mean(oHCRI_len_bt1);

mean(CI_bt2_width); mean(CRI_len_bt2); mean(HCRI_len_bt2);
mean(CI_bt2_or_width); mean(oCRI_len_bt2); mean(oHCRI_len_bt2);

### Real Data III (Partially Observed Competing Risk (Kumaraswamy Distribution)) ###
rm(list=ls(all=TRUE));
rm(list=ls(all=TRUE));
rm(list=ls(all=TRUE));
library(nleqslv);
xsamp <- c(0.0222, 0.0333, 0.0778, 0.0889, 0.1000, 0.1222, 0.1444, 0.1556, 0.1667,
0.1778, 0.1778, 0.2000, 0.2000, 0.2111, 0.2222, 0.2667, 0.2889, 0.3000, 0.3111,
0.3333, 0.3778, 0.4000, 0.4333, 0.4444, 0.4667, 0.5333, 0.5444, 0.6889, 0.7111,
0.7333);
n <- 37; m <- 30; k=28;
R <- c(rep(1,(n-m)), rep(0,(2*m-n)));
T=0.8;
los <- 0.05;
xstart <- 1;
MLE_Estimation <- function(ings, ysp, dst, rr, rrstr, tt){
  # MLE for nu using nleqslv Package
  lik_eqn <- function(nu){
    lkeq <- (dst/nu)+(sum((log(ysp))/(1-(ysp^nu)))-(dst*((sum((rr+1)*((ysp^nu)*
log(ysp))/(1-(ysp^nu)))+(rrstr*((tt^nu)*log(tt))/((1-(tt^nu)))))/
(-sum((rr+1)*log(1-(ysp^nu)))-(rrstr*log(1-(tt^nu))))));
    return(lkeq);
  }
  mlnu <- nleqslv(ings, lik_eqn);
  return(mlnu$x);
}

Bayes_Estimation <- function(xsp, dst, rr, rrstr, tt){
  #a1 = 6; b1 = 4; a0 = 3.2; b0 = 2; c1 <- (7/9); c2 <- 1;
  a1 = 0.00001; b1 = 2; a0 = 0.00001; b0 = 2; c1 <- 0.00001; c2 <- 2;

```

```

dens_of_nu <- function(vnu, xsm, ds, rst, ri, TTT){
  # Density of nu from posterior density
  s1=0;s2=0;s3= s4= 0;
  for(jk in 1:length(xsm)){
    s1 = s1 + ((ri[jk]+1)*log(1-(xsm[jk]^vnu)))
    s2 = s2 + ((ri[jk]+1)*(((xsm[jk]^(vnu))*log(xsm[jk]))/(1-(xsm[jk]^(vnu)))));
    s3 = s3 + ((ri[jk]+1)*((xsm[jk]^(vnu))*(log(xsm[jk]))/(1-(xsm[jk]^(vnu))))^(2))
    s4 = s4 + log((xsm[jk]^(vnu-1))/(1-(xsm[jk]^vnu)))
  }
  W_nu <- (-s1 - rst*log(1-(TTT^vnu)));
  W_nuD <- (s2 + (rst*(((TTT^(vnu))*log(TTT))/(1-(TTT^(vnu))))));
  W_nuDD <- (s3 + (rst*(((TTT^(vnu))*((log(TTT))/(1-(TTT^vnu))))^(2))))
  #fnval <- (((vnu^(a1 + ds - 1))/(b0 + W_nu)^(a0 +
  ds))) * exp(-(b1-sum(log(xsm))*vnu)/exp(-vnu));
  fnval <- (((vnu^(a1 + ds - 1))/(b0 + W_nu)^(a0 + ds))) *
  exp(-(b1*vnu)+(s4)) #/exp(-vnu);
  #fDDVal <- (sum(((xsm^vnu)*((log(xsm^vnu))^2))/((1-(xsm^vnu))^2)))/(a1+ds-1)
  return(fnval);
  #-((a1+ds-1)/(vnu^2))-((a0+ds)*(((b0+W_nu)*(W_nuDD))-(W_nuD^2))/((b0+W_nu)^2))
}

dens_of_nu_mean <- function(vnu, xsm, ds, rst, ri, TTT,cnst){
  # Density of nu from posterior density
  s1=0;s2=0;s3= s4= 0;
  for(jk in 1:length(xsm)){
    s1 = s1 + ((ri[jk]+1)*log(1-(xsm[jk]^vnu)))
    s2 = s2 + ((ri[jk]+1)*(((xsm[jk]^(vnu))*log(xsm[jk]))/(1-(xsm[jk]^(vnu)))));
    s3 = s3 + ((ri[jk]+1)*((xsm[jk]^(vnu))*(log(xsm[jk]))/(1-(xsm[jk]^(vnu))))^(2))
    s4 = s4 + log((xsm[jk]^(vnu-1))/(1-(xsm[jk]^vnu)))
  }
  W_nu <- (-s1 - rst*log(1-(TTT^vnu)));
  W_nuD <- (s2 + (rst*(((TTT^(vnu))*log(TTT))/(1-(TTT^(vnu))))));
  W_nuDD <- (s3 + (rst*(((TTT^(vnu))*((log(TTT))/(1-(TTT^vnu))))^(2))))
  #fnval <- (((vnu^(a1 + ds - 1))/(b0 + W_nu)^(a0 +
  ds))) * exp(-(b1-sum(log(xsm))*vnu)/exp(-vnu));
  fnval <- (1/cnst)*vnu*(((vnu^(a1 + ds - 1))/(b0 + W_nu)^(a0 +
  ds))) * exp(-(b1*vnu)+(s4)) #/exp(-vnu);
  #fDDVal <- (sum(((xsm^vnu)*((log(xsm^vnu))^2))/((1-(xsm^vnu))^2)))/(a1+ds-1)
  return(fnval);
  #-((a1+ds-1)/(vnu^2))-((a0+ds)*(((b0+W_nu)*(W_nuDD))-(W_nuD^2))/((b0+W_nu)^2))
}

dens_of_nu_mean_2 <- function(vnu, xsm, ds, rst, ri, TTT,cnst){
  # Density of nu from posterior density
  s1=0;s2=0;s3= s4= 0;
  for(jk in 1:length(xsm)){
    s1 = s1 + ((ri[jk]+1)*log(1-(xsm[jk]^vnu)))
    s2 = s2 + ((ri[jk]+1)*(((xsm[jk]^(vnu))*log(xsm[jk]))/(1-(xsm[jk]^(vnu)))));
    s3 = s3 + ((ri[jk]+1)*((xsm[jk]^(vnu))*(log(xsm[jk]))/(1-(xsm[jk]^(vnu))))^(2))
    s4 = s4 + log((xsm[jk]^(vnu-1))/(1-(xsm[jk]^vnu)))
  }
  W_nu <- (-s1 - rst*log(1-(TTT^vnu)));

```

```

W_nuD <- (s2 + (rst*(((TTT^(vnu))*log(TTT))/(1-(TTT^(vnu))))));
W_nuDD <- (s3 + (rst*(((TTT^(vnu))*(((log(TTT))/(1-(TTT^vnu)))^(2))))));
#fnval <- (((vnu^(a1 + ds - 1))/((b0 + W_nu)^(a0 +
ds))) * exp(-(b1-sum(log(xsm))*vnu))/exp(-vnu));
fnval <- (1/cnst)*(vnu^2)*(((vnu^(a1 + ds - 1))/((b0 + W_nu)^(a0 +
ds))) * exp(-(b1*vnu)+(s4))) #/exp(-vnu);
#fDDVal <- (sum(((xsm^vnu)*((log(xsm^vnu))^(2)))/((1-(xsm^vnu))^(2))))/(a1+ds-1)
return(fnval);
#-((a1+ds-1)/(vnu^(2)))-((a0+ds)*(((b0+W_nu)*(W_nuDD))-(W_nuD^2))/((b0+W_nu)^(2)))
}
const <- integrate(dens_of_nu, 0, Inf, xsm=xsp, ds=dst,
rst=rrstr, ri=rr, TTT=tt)$value;
mn <- integrate(dens_of_nu_mean, 0, Inf, xsm=xsp, ds=dst,
rst=rrstr, ri=rr, TTT=tt, cnst = const)$value; # Calculating first moment
mn2 <- integrate(dens_of_nu_mean_2, 0, Inf, xsm=xsp, ds=dst,
rst=rrstr, ri=rr, TTT=tt, cnst = const)$value; # Calculating first moment
#cat("\n The mean and variance", c(const,mn,mn2))
vr <- (mn2-((mn)^2)); # Calculating variance of using density of nu
a5 <- (mn^2)/vr; b5 <- mn/vr;
# Calculating parameters for the approximating gamma distribution
#cat("\n mn and vr =",a5/b5);
if(a5<0 || b5 <0){
  a5 <- 1.5;
  b5 <- 1;
  count = 1;
}else{
  count = 0;
}
#cat("\n a5 and b5 =",c(a5,b5));
baynuu <- numeric(0);
baybtt1 <- numeric(0);
baybtt2 <- numeric(0);
for(ii in 1:4000){
  baynuu[ii] <- rgamma(1, a5, b5);
  #print(baynuu[ii]);
  w_nu <- -sum((rr+1)*log(1-(xsp^baynuu[ii])))- rrstr*log(1-(tt^baynuu[ii]));
  # Calculating the function W(nu)
  bt0 <- rgamma(1, a0+dst,b0+w_nu);
  # Generating random beta0 = (beta1 + beta2) from gamma distribution
  btbybt0 <- rbeta(1, c1+e1, c2+e2); # Generating random beta
  baybtt1[ii] <- bt0*btbybt0; # Generated values of beta1
  baybtt2[ii] <- bt0-baybtt1[ii]; # Generated values of beta1
}
BAYNUU <- baynuu[(1001:4000)];
BAYBTT1 <- baybtt1[(1001:4000)];
BAYBTT2 <- baybtt2[(1001:4000)];
nu_nw <- mean((baynuu[(1001:4000)]));
bt1_nw <- mean((baybtt1[(1001:4000)]));
bt2_nw <- mean((baybtt2[(1001:4000)]));
bynums <- (nu_nw - 11)^2;

```

```

bybt1ms <- (bt1_nw - 11)^2;
bybt2ms <- (bt2_nw - 11)^2;
baynuusort <- sort((baynuu[(1001:4000)]));
baybtt1sort <- sort((baybtt1[(1001:4000)]));
baybtt2sort <- sort((baybtt2[(1001:4000)]));
Bayes_low_nu <- Bayes_upp_nu <- Bayes_len_nu <- numeric(0);
Bayes_low_bt1 <- Bayes_upp_bt1 <- Bayes_len_bt1 <- numeric(0);
Bayes_low_bt2 <- Bayes_upp_bt2 <- Bayes_len_bt2 <- numeric(0);
for(ir in 1:(length(baynuusort)*los)){
  Bayes_low_nu[ir] <- (baynuusort[ir])
  Bayes_upp_nu[ir] <- (baynuusort[ir+((length(baynuusort)*(1-los))])
  Bayes_len_nu[ir] <- (Bayes_upp_nu[ir]- Bayes_low_nu[ir])
  Bayes_low_bt1[ir] <- (baybtt1sort[ir])
  Bayes_upp_bt1[ir] <- (baybtt1sort[ir+((length(baynuusort)*(1-los))])
  Bayes_len_bt1[ir] <- (Bayes_upp_bt1[ir]- Bayes_low_bt1[ir]);
  Bayes_low_bt2[ir] <- (baybtt2sort[ir])
  Bayes_upp_bt2[ir] <- (baybtt2sort[ir+((length(baynuusort)*(1-los))])
  Bayes_len_bt2[ir] <- (Bayes_upp_bt2[ir]- Bayes_low_bt2[ir]);
}
#print(baynuusort)
id1 <- which(Bayes_len_nu==min(Bayes_len_nu));
id2 <- which(Bayes_len_bt1==min(Bayes_len_bt1));
id3 <- which(Bayes_len_bt2==min(Bayes_len_bt2));
HPD_low_nu <-Bayes_low_nu[id1];
HPD_upp_nu <-Bayes_upp_nu[id1];
HPD_len_nu <-(HPD_upp_nu-HPD_low_nu)
HPD_low_bt1 <-Bayes_low_bt1[id2];
HPD_upp_bt1 <-Bayes_upp_bt1[id2];
HPD_len_bt1 <-(HPD_upp_bt1-HPD_low_bt1);
HPD_low_bt2 <-Bayes_low_bt2[id3];
HPD_upp_bt2 <-Bayes_upp_bt2[id3];
HPD_len_bt2 <-(HPD_upp_bt2-HPD_low_bt2);

Sym_low_nu <-mean(Bayes_low_nu);
Sym_upp_nu <-mean(Bayes_upp_nu);
Sym_len_nu <-mean(Bayes_len_nu);
Sym_low_bt1 <-mean(Bayes_low_bt1);
Sym_upp_bt1 <-mean(Bayes_upp_bt1);
Sym_len_bt1 <-mean(Bayes_len_bt1);
Sym_low_bt2 <-mean(Bayes_low_bt2);
Sym_upp_bt2 <-mean(Bayes_upp_bt2);
Sym_len_bt2 <-mean(Bayes_len_bt2);

#print(Nu_Sam)
return(c(nu_nw, bt1_nw, bt2_nw, bynums, bybt1ms, bybt2ms,
Sym_low_nu, Sym_upp_nu, Sym_len_nu, Sym_low_bt1, Sym_upp_bt1,
Sym_len_bt1, Sym_low_bt2, Sym_upp_bt2, Sym_len_bt2, HPD_low_nu,
HPD_upp_nu, HPD_len_nu, HPD_low_bt1, HPD_upp_bt1, HPD_len_bt1,
HPD_low_bt2, HPD_upp_bt2, HPD_len_bt2, count ));

```

```

#return(list(BAYNUU,BAYBTT1,BAYBTT2)) #Return for trace plot
}
#####Ordered Bayes #####
0Bayes_Estimation <- function(oxsp, odst, oe0, oe1, oe2, oe3, orr, orrstr, ott){
  #oa1 = 6; ob1 = 4; oa0 = 3.2; ob0 = 2; oc1 <- (7/9); oc2 <- 1;
  oa1 = 0.00001; ob1 = 2; oa0 = 0.00001; ob0 = 2; oc1 <- 0.00001; oc2 <- 2;
  odens_of_nu <- function(ovnu, oxsm, ods, orst, ori, oTTT){
    # Density of nu from posterior density
    os1=0;os2=0;os3= os4= 0;
    for(ojk in 1:length(oxsm)){
      os1 = os1 + ((ori[ojk]+1)*log(1-(oxsm[ojk]^ovnu)))
      os2 = os2 + ((ori[ojk]+1)*(((oxsm[ojk]^(ovnu))*log(oxsm[ojk]))/
(1-(oxsm[ojk]^(ovnu)))));
      os3 = os3 + ((ori[ojk]+1)*((oxsm[ojk]^(ovnu))*(log(oxsm[ojk])/
(1-(oxsm[ojk]^(ovnu))))^(2))));
      os4 = os4 + log((oxsm[ojk]^(ovnu-1))/(1-(oxsm[ojk]^ovnu)))
    }
    oW_nu <- (-os1 - orst*log(1-(oTTT^ovnu)));
    oW_nuD <- (os2 + (orst*(((oTTT^(ovnu))*log(oTTT))/(1-(oTTT^(ovnu))))));
    oW_nuDD <- (os3 + (orst*(((oTTT^(ovnu))*((log(oTTT))/(1-(oTTT^ovnu)))^(2))))));
    ofnval <- (((ovnu^(oa1 + ods - 1))/(ob0 + oW_nu)^(oa0 + oe0 +(2*oe3))))*
exp(-(ob1*ovnu)+(os4)));
    return(ofnval);
  }
}
odens_of_nu_mean <- function(ovnu, oxsm, ods, orst, ori, oTTT,ocnst){
  # Density of nu from posterior density
  os1=0;os2=0;os3= os4= 0;
  for(ojk in 1:length(oxsm)){
    os1 = os1 + ((ori[ojk]+1)*log(1-(oxsm[ojk]^ovnu)))
    os2 = os2 + ((ori[ojk]+1)*(((oxsm[ojk]^(ovnu))*log(oxsm[ojk]))/
(1-(oxsm[ojk]^(ovnu)))));
    os3 = os3 + ((ori[ojk]+1)*((oxsm[ojk]^(ovnu))*(log(oxsm[ojk])/
(1-(oxsm[ojk]^(ovnu))))^(2))));
    os4 = os4 + log((oxsm[ojk]^(ovnu-1))/(1-(oxsm[ojk]^ovnu)))
  }
  oW_nu <- (-os1 - orst*log(1-(oTTT^ovnu)));
  oW_nuD <- (os2 + (orst*(((oTTT^(ovnu))*log(oTTT))/(1-(oTTT^(ovnu))))));
  oW_nuDD <- (os3 + (orst*(((oTTT^(ovnu))*((log(oTTT))/(1-(oTTT^ovnu)))^(2))))));
  ofnval <- (1/ocnst)*ovnu*(((ovnu^(oa1 + ods - 1))/(ob0 + oW_nu)^(oa0 + oe0 +
(2*oe3))))*exp(-(ob1*ovnu)+(os4)));
  return(ofnval);
}
}
odens_of_nu_mean_2 <- function(ovnu, oxsm, ods, orst, ori, oTTT,ocnst){
  # Density of nu from posterior density
  os1=0;os2=0;os3 = os4= 0;
  for(ojk in 1:length(oxsm)){
    os1 = os1 + ((ori[ojk]+1)*log(1-(oxsm[ojk]^ovnu)))
    os2 = os2 + ((ori[ojk]+1)*(((oxsm[ojk]^(ovnu))*log(oxsm[ojk]))/
(1-(oxsm[ojk]^(ovnu)))));
    os3 = os3 + ((ori[ojk]+1)*((oxsm[ojk]^(ovnu))*(log(oxsm[ojk])/

```

```

      (1-(oxsm[ojk]^(ovnu))))^(2));
      os4 = os4 + log((oxsm[ojk]^(ovnu-1))/(1-(oxsm[ojk]^ovnu)))
    }
    oW_nu <- (-os1 - orst*log(1-(oTTT^ovnu)));
    oW_nuD <- (os2 + (orst*((oTTT^(ovnu))*log(oTTT))/(1-(oTTT^(ovnu)))));
    oW_nuDD <- (os3 + (orst*((oTTT^(ovnu))*((log(oTTT))/(1-(oTTT^ovnu)))^2))))
    ofnval <- (1/ocnst)*(ovnu^2)*((ovnu^(oa1 + ods - 1))/
      ((ob0 + oW_nu)^(oa0 + oe0 + (2*oe3))))*exp(-(ob1*ovnu)+(os4));
    return(ofnval);
  }
  oNN <- 1000;
  oconst <- integrate(odens_of_nu, 0, Inf, oxsm=oxsp, ods=odst, orst=orrstr,
ori=orr, oTTT=ott)$value;
  omn <- integrate(odens_of_nu_mean, 0, Inf, oxsm=oxsp, ods=odst, orst=orrstr,
ori=orr, oTTT=ott, ocnst = oconst)$value; # Calculating first moment
  omn2 <- integrate(odens_of_nu_mean_2, 0, Inf, oxsm=oxsp, ods=odst, orst=orrstr,
ori=orr, oTTT=ott, ocnst = oconst)$value; # Calculating first moment
  #cat("\n The mean and variance", c(const,mn,mn2))
  ovr <- (omn2-((omn)^2)); # Calculating variance of using density of nu
  oa5 <- (omn^2)/ovr; ob5 <- omn/ovr;
  # Calculating parameters for the approximating gamma distribution
  #cat("\n mn and vr =",a5/b5);
  if(oa5<0 || ob5 <0){
    oa5 <- 2;
    ob5 <- 2;
    ocount = 1;
  }else{
    ocount = 0;
  }
  obaynuuu <- numeric(0);
  obaybttt1 <- numeric(0);
  obaybttt2 <- numeric(0);
  obaynuu <- numeric(0);
  obaybtt1 <- numeric(0);
  obaybtt2 <- numeric(0);
  ohfn <- numeric(0);
  for(oii in 1:oNN){
    obaynuuu[oii] <- rgamma(1, oa5, ob5);
    ow_nu <- -sum((orr+1)*log(1-(oxsp^obaynuuu[oii])))-
    orrstr*log(1-(ott^obaynuuu[oii])); # Calculating the function W(nu)
    obt0 <- rgamma(1, oa0 + oe0 + 2*oe3, ob0+ow_nu);
    # Generating random beta0 = (beta1 + beta2) from gamma distribution
    obtbybt0 <- rbeta(1, oc1+oe3, oc2+oe3); # Generating random beta
    obaybttt1[oii] <- obt0*obtbybt0; # Generated values of beta1
    obaybttt2[oii] <- obt0-obaybttt1[oii]; # Generated values of beta1
    obaybttt1[oii] <- min(obaybttt1[oii], obaybttt2[oii]);
    obaybttt2[oii] <- max(obaybttt1[oii], obaybttt2[oii]);
    ol5 <- (1/oconst)*odens_of_nu(obaynuuu[oii], oxsp, odst, orrstr, orr, ott);
    ol6 <- dgamma(obaynuuu[oii], oa5, ob5);
    ohfn[oii] <- ((obaybttt1[oii]^(oe1-oe3))*(obaybttt2[oii]^(oe2-oe3)))*(ol5/ol6);
  }

```

```

    obaynuu[oii] <- obaynuuu[oii]*ohfn[oii];
    obaybtt1[oii] <- obaybttt1[oii]*ohfn[oii];
    obaybtt2[oii] <- obaybttt2[oii]*ohfn[oii];
}
ow1 <- ohfn/sum(ohfn);
onu_nw <- sum(obaynuu)/sum(ohfn);
obt1_nw <- sum(obaybtt1)/sum(ohfn);
obt2_nw <- sum(obaybtt2)/sum(ohfn);
obynums <- (onu_nw - 11)^2;
obybt1ms <- (obt1_nw - 11)^2;
obybt2ms <- (obt2_nw - 11)^2;
obaynuusort <- sort((obaynuuu));
obaybtt1sort <- sort((obaybttt1));
obaybtt2sort <- sort((obaybttt2));
#####
owisortnu <- ow1[order(obaynuuu)];
owisortbt1 <- ow1[order(obaybttt1)];
owisortbt2 <- ow1[order(obaybttt2)];
onuj1 <- numeric(0);
onuj2 <- numeric(0);
onui <- 1;
obt1j1 <- numeric(0);
obt1j2 <- numeric(0);
obt1i <- 1;
obt2j1 <- numeric(0);
obt2j2 <- numeric(0);
obt2i <- 1;
for(oj1 in 1:(oNN-1)){

    for(oj2 in oj1:(oNN-1)){
        olwnuw1 <- sum(owisortnu[oj1:oj2]);
        oupnuw1 <- sum(owisortnu[oj1:(oj2+1)]);
        if(olwnuw1 <= (1-los) && oupnuw1 > (1-los)){onuj1[onui] =
            oj1; onuj2[onui]=oj2; onui <- onui+1; break;}
    }
}
for(oj1 in 1:(oNN-1)){
    for(oj2 in oj1:(oNN-1)){
        olwbt1w1 <- sum(owisortbt1[oj1:oj2]);
        oupbt1w1 <- sum(owisortbt1[oj1:(oj2+1)]);
        if(olwbt1w1 <= (1-los) && oupbt1w1 > (1-los)){obt1j1[obt1i] =
            oj1; obt1j2[obt1i]=oj2; obt1i <- obt1i + 1; break;}
    }
}
for(oj1 in 1:(oNN-1)){
    for(oj2 in oj1:(oNN-1)){
        olwbt2w1 <- sum(owisortbt2[oj1:oj2]);
        oupbt2w1 <- sum(owisortbt2[oj1:(oj2+1)]);
        if(olwbt2w1 <= (1-los) && oupbt2w1 > (1-los)){obt2j1[obt2i] =
            oj1; obt2j2[obt2i]=oj2; obt2i <- obt2i + 1; break;}
    }
}

```

```

    }
}
ocri_nu_lw <- obaynuusort[onuj1]; ocri_nu_up <- obaynuusort[onuj2];
ocri_bt1_lw <- obaybtt1sort[obt1j1]; ocri_bt1_up <- obaybtt1sort[obt1j2];
ocri_bt2_lw <- obaybtt2sort[obt2j1]; ocri_bt2_up <- obaybtt2sort[obt2j2];
odiffcri_nu <- ocri_nu_up - ocri_nu_lw;
odiffcri_bt1 <- ocri_bt1_up - ocri_bt1_lw;
odiffcri_bt2 <- ocri_bt2_up - ocri_bt2_lw;
oind_min_nu <- which(odiffcri_nu==min(odiffcri_nu));
oind_min_bt1 <- which(odiffcri_bt1==min(odiffcri_bt1));
oind_min_bt2 <- which(odiffcri_bt2==min(odiffcri_bt2));
oHPD_low_nu <- ocri_nu_lw[oind_min_nu];
oHPD_upp_nu <- ocri_nu_up[oind_min_nu];
oHPD_len_nu <- (oHPD_upp_nu-oHPD_low_nu)
oHPD_low_bt1 <- ocri_bt1_lw[oind_min_bt1];
oHPD_upp_bt1 <- ocri_bt1_up[oind_min_bt1];
oHPD_len_bt1 <-(oHPD_upp_bt1-oHPD_low_bt1);
oHPD_low_bt2 <- ocri_bt2_lw[oind_min_bt2];
oHPD_upp_bt2 <- ocri_bt2_up[oind_min_bt2];
oHPD_len_bt2 <-(oHPD_upp_bt2-oHPD_low_bt2);

oSym_low_nu <-mean(ocri_nu_lw);
oSym_upp_nu <-mean(ocri_nu_up);
oSym_len_nu <-mean(odiffcri_nu);
oSym_low_bt1 <-mean(ocri_bt1_lw);
oSym_upp_bt1 <-mean(ocri_bt1_up);
oSym_len_bt1 <-mean(odiffcri_bt1);
oSym_low_bt2 <-mean(ocri_bt2_lw);
oSym_upp_bt2 <-mean(ocri_bt2_up);
oSym_len_bt2 <-mean(odiffcri_bt2);

#print(Nu_Sam)
return(c(onu_nw, obt1_nw, obt2_nw, obynums, obybt1ms, obybt2ms, oSym_low_nu,
oSym_upp_nu, oSym_len_nu, oSym_low_bt1, oSym_upp_bt1, oSym_len_bt1, oSym_low_bt2,
oSym_upp_bt2,oSym_len_bt2, oHPD_low_nu, oHPD_upp_nu, oHPD_len_nu, oHPD_low_bt1,
oHPD_upp_bt1, oHPD_len_bt1, oHPD_low_bt2, oHPD_upp_bt2,oHPD_len_bt2, ocount ));
}
#dstr <- k; xsmp <- xsamp[1:dstr]; r <- c(R[1:(dstr-1)],(n-sum(R[1:(dstr-1)]+1)-1));
rstr <- 0;
dstr <- m; xsmp <- xsamp[1:dstr]; r <- R; rstr <- 0;
e0 <- 3;
e1 <- 14;
e2 <- 13;
mlenu <- MLE_Estimation(xstart, xsmp, dstr, r, rstr, T);
mlebt1 <- ((e1)/(-sum((1+r)*log(1-(xsmp^(mlenu))))))*((e0+e1+e2)/(e1+e2));
mlebt2 <- ((e2)/(-sum((1+r)*log(1-(xsmp^(mlenu))))))*((e0+e1+e2)/(e1+e2));
if(e1 < e2){
  mleobt1 <- mlebt1;
  mleobt2 <- mlebt2;
}else{

```



```

mleobt1 <- mleobt2 <- (dstr/(2*sum((r+1)*log(1+(mlenu*xsmp)))));
}
obs_fish_matrix <- function(RRR, rsstr, xx, pr, TT){
  nuu <- pr[1];
  bet1 <- pr[2];
  bet2 <- pr[3];
  ofm <- matrix(0, nrow=3, ncol=3);
  ofm[1,1] <- (e0/((bet1+bet2)^2)) + ((e1)/(bet1^2));
  ofm[1,2] <- ofm[2,1] <- (e0/((bet1+bet2)^2));
  ofm[1,3] <- ofm[3,1] <- ofm[2,3] <- ofm[3,2] <-
  sum((RRR+1)*((xx^nuu)*log(xx))/(1-(xx^nuu))) +
  (rsstr*(((TT^nuu)*log(TT))/(1-(TT^nuu))))
  ofm[2,2] <- (e0/((bet1+bet2)^2)) + ((e2)/(bet2^2));
  ofm[3,3] <- (dstr/(nuu^2)) + ((bet1+bet2)*(sum((RRR+1)*(xx^nuu)*
  (((log(xx))/(1-(xx^nuu)))^2))) + (rsstr*(TT^nuu)*(((log(TT))/
  (1-(TT^nuu)))^2)))) + (sum(((xx^nuu)*((log(xx))^2)))/((1-(xx^nuu))^2));
  return(ofm);
}
var_cov <- solve(obs_fish_matrix(r, rstr, xsmp, c(mlenu,mlebt1,mlebt2), T));
var_cov_or <- solve(obs_fish_matrix(r, rstr, xsmp, c(mlenu,mleobt1,mleobt2), T));
print(var_cov);
confid<-function(acon,v,ad){
  #print(v)
  fcon<-1-(ad/2);
  zad<-qnorm(fcon,0,1);
  ac1<-acon-((zad)*(sqrt(v)));
  ac2<-acon+((zad)*(sqrt(v)));
  asconf<-c(ac1,ac2);
  asconf
}
#print(var_cov);
mlen <- mlenu;
mleb1 <- mlebt1;
mleb2 <- mlebt2;
mleon <- mlenu;
mleob1 <- mleobt1;
mleob2 <- mleobt2;
if(var_cov[1,1] <0 ){var_cov[1,1] <- 0.2}
if(var_cov[2,2] <0 ){var_cov[2,2] <- 0.2}
if(var_cov[3,3] <0 ){var_cov[3,3] <- 0.2}
CI_nuu <- confid(mlen,(var_cov[3,3]), los);
CI_bt1 <- confid(mleb1,(var_cov[1,1]), los);
CI_bt2 <- confid(mleb2,(var_cov[2,2]), los);
CI_nuu_width <- CI_nuu[2]-CI_nuu[1];
CI_bt1_width <- CI_bt1[2]-CI_bt1[1];
CI_bt2_width <- CI_bt2[2]-CI_bt2[1];
### Classical OR
CI_nuu_or <- confid(mleon,abs(var_cov_or[3,3]), los);
CI_bt1_or <- confid(mleob1,abs(var_cov_or[1,1]), los);
CI_bt2_or <- confid(mleob2,abs(var_cov_or[2,2]), los);

```

```

CI_nuu_or_width <- CI_nuu_or[2]-CI_nuu_or[1];
CI_bt1_or_width <- CI_bt1_or[2]-CI_bt1_or[1];
CI_bt2_or_width <- CI_bt2_or[2]-CI_bt2_or[1];
Bayes <- Bayes_Estimation(xsmpl, dstr, r, rstr, T);
OBayes <- OBayes_Estimation(xsmpl, dstr, e0, e1, e2, min(e1, e2), r, rstr, T);
#cat(OBayes)
nubayes <- Bayes[1];
bt1bayes <- Bayes[2];
bt2bayes <- Bayes[3];
numsbayes <- Bayes[4];
bt1msbayes <- Bayes[5];
bt2msbayes <- Bayes[6];
CRI_low_nu <- Bayes[7];
CRI_upp_nu <- Bayes[8];
CRI_len_nu <- Bayes[9];
CRI_low_bt1 <- Bayes[10];
CRI_upp_bt1 <- Bayes[11];
CRI_len_bt1 <- Bayes[12];
CRI_low_bt2 <- Bayes[13];
CRI_upp_bt2 <- Bayes[14];
CRI_len_bt2 <- Bayes[15];
HCRI_low_nu <- Bayes[16];
HCRI_upp_nu <- Bayes[17];
HCRI_len_nu <- Bayes[18];
HCRI_low_bt1 <- Bayes[19];
HCRI_upp_bt1 <- Bayes[20];
HCRI_len_bt1 <- Bayes[21];
HCRI_low_bt2 <- Bayes[22];
HCRI_upp_bt2 <- Bayes[23];
HCRI_len_bt2 <- Bayes[24];
countNew <- Bayes[25];
# NUSamp <- Bayes[[1]];
# BT1Samp <- Bayes[[2]];
# BT2Samp <- Bayes[[3]];
# df1 <- cbind(NUSamp, BT1Samp, BT2Samp)
# write.csv(df1,"C:\\Users\\amuly\\OneDrive\\PhD_Full_Folder\\
PhD 4th year\\GPHC_Kumaraswamy\\KumPrograms\\MCMC_Samples_For_Trace\\
MyData3_4.csv", row.names = FALSE)

onubayes <- OBayes[1];
obt1bayes <- OBayes[2];
obt2bayes <- OBayes[3];
onnumsbayes <- OBayes[4];
obt1msbayes <- OBayes[5];
obt2msbayes <- OBayes[6];
oCRI_low_nu <- OBayes[7];
oCRI_upp_nu <- OBayes[8];
oCRI_len_nu <- OBayes[9];
oCRI_low_bt1 <- OBayes[10];
oCRI_upp_bt1 <- OBayes[11];

```

```

oCRI_len_bt1 <- OBayes[12];
oCRI_low_bt2 <- OBayes[13];
oCRI_upp_bt2 <- OBayes[14];
oCRI_len_bt2 <- OBayes[15];
oHCRI_low_nu <- OBayes[16];
oHCRI_upp_nu <- OBayes[17];
oHCRI_len_nu <- OBayes[18];
oHCRI_low_bt1 <- OBayes[19];
oHCRI_upp_bt1 <- OBayes[20];
oHCRI_len_bt1 <- OBayes[21];
oHCRI_low_bt2 <- OBayes[22];
oHCRI_upp_bt2 <- OBayes[23];
oHCRI_len_bt2 <- OBayes[24];
##### Classical Estimation Without Order Restriction
# Average Estimate
mean(mlen); mean(mleon);
mean(mleb1); mean(mleob1);
mean(mleb2); mean(mleob2);
#Bayes Average Estimates
mean(nubayes); mean(onubayes);
mean(bt1bayes); mean(obl1bayes);
mean(bt2bayes); mean(obl2bayes);
#.....Credible and HPD intervals.....
mean(CRI_low_nu); mean(CRI_upp_nu);
mean(CRI_low_bt1); mean(CRI_upp_bt1);
mean(CRI_low_bt2); mean(CRI_upp_bt2);
mean(HCRI_low_nu); mean(HCRI_upp_nu);
mean(HCRI_low_bt1); mean(HCRI_upp_bt1);
mean(HCRI_low_bt2); mean(HCRI_upp_bt2);
#####
mean(CI_nuu_width); mean(CRI_len_nu); mean(HCRI_len_nu);
mean(CI_nuu_or_width); mean(oCRI_len_nu); mean(oHCRI_len_nu);

mean(CI_bt1_width); mean(CRI_len_bt1); mean(HCRI_len_bt1);
mean(CI_bt1_or_width); mean(oCRI_len_bt1); mean(oHCRI_len_bt1);

mean(CI_bt2_width); mean(CRI_len_bt2); mean(HCRI_len_bt2);
mean(CI_bt2_or_width); mean(oCRI_len_bt2); mean(oHCRI_len_bt2);

```