ภาคผนวก

ภาคผนวก ก

โปรแกรมที่ใช้ในการวิจัย

ในการวิจัยครั้งนี้ ใช้โปรแกรมสำเร็จรูปทางสถิติ โปรแกรม R version 2.9.1 ในการจำลองข้อมูล คำนวณค่าขีดจำกัดล่างและขีดจำกัดบน และเกณฑ์ที่ใช้ในการวิจัย

A comparison on confidence intervals estimation methods for the poisson population mean.

```
        function (n,M,alpha,alpha1,beta1,alpha2,beta2,alpha3,beta3,alpha4,beta4)
{       ###Written by Rungtawan , 15/06/09
        ###n is a sample size.
        ###M is a number of simulation runs.
        ###alpha is a significance level.
        ###alpha1,beta1,alpha2,beta2,alpha3,beta3,alpha4 and beta4 are parameter of
        ###gamma distribution

                L.WALD=rep(0,M)
                U.WALD=rep(0,M)
                L.SCORE=rep(0,M)
                U.SCORE=rep(0,M)
                L.ADJUST=rep(0,M)
                U.ADJUST=rep(0,M)
                L.BAYE1=rep(0,M)
                U.BAYE1=rep(0,M)
                L.BAYE2=rep(0,M)
                U.BAYE2=rep(0,M)
                L.BAYE3=rep(0,M)
                U.BAYE3=rep(0,M)
```

```
L.BAYE4=rep(0,M)
U.BAYE4=rep(0,M)
length.WALD=rep(0,M)
length.SCORE=rep(0,M)
length.ADJUST=rep(0,M)
length.BAYE1=rep(0,M)
length.BAYE2=rep(0,M)
length.BAYE3=rep(0,M)
length.BAYE4=rep(0,M)
lambda<-c(0.2,0.4,0.6,0.8,1)
y1=rep(0,5)
y2=rep(0,5)
y3=rep(0,5)
y4=rep(0,5)
y5=rep(0,5)
y6=rep(0,5)
y7=rep(0,5)
for(k in 1:length(lambda))
{
        count1=rep(0,M)
        count2=rep(0,M)
        count3=rep(0,M)
        count4=rep(0,M)
        count5=rep(0,M)
        count6=rep(0,M)
        count7=rep(0,M)
```

```
for(i in 1:M)
{       x=rpois(n,lambda[k])
        z=qnorm(1-alpha/2)
######## compute upper and lower bound
        lambda_hat=mean(x)
        L.WALD[i]=lambda_hat-z*sqrt(lambda_hat/n)
        U.WALD[i]=lambda_hat+z*sqrt(lambda_hat/n)
        L.SCORE[i]=(lambda_hat+(z^2/(2*n)))-
z*sqrt(lambda_hat/n+(z^2/(4*n*n)))
        U.SCORE[i]=(lambda_hat+(z^2/(2*n)))+
z*sqrt(lambda_hat/n+(z^2/(4*n*n)))
        lambda_adjust=(sum(x)+(z^2)/2)/n
        L.ADJUST[i]=lambda_adjust-z*sqrt(lambda_adjust/n)
        U.ADJUST[i]=lambda_adjust+z*sqrt(lambda_adjust/n)
        df1=2*(sum(x)+alpha1)
        L.BAYE1[i]=beta1*qchisq(alpha/2,df1)/(2*(beta1*n+1))
        U.BAYE1[i]=beta1*qchisq(1-alpha/2,df1)/(2*(beta1*n+1))
        df2=2*(sum(x)+alpha2)
        L.BAYE2[i]=beta2*qchisq(alpha/2,df2)/(2*(beta2*n+1))
        U.BAYE2[i]=beta2*qchisq(1-alpha/2,df2)/(2*(beta2*n+1))
        df3=2*(sum(x)+alpha3)
        L.BAYE3[i]=beta3*qchisq(alpha/2,df3)/(2*(beta3*n+1))
        U.BAYE3[i]=beta3*qchisq(1-alpha/2,df3)/(2*(beta3*n+1))
        df4=2*(sum(x)+alpha4)
        L.BAYE4[i]=beta4*qchisq(alpha/2,df4)/(2*(beta4*n+1))
        U.BAYE4[i]=beta4*qchisq(1-alpha/2,df4)/(2*(beta4*n+1))
```

```
######## compute coverage probability and average length
        if((lambda[k]>=L.WALD[i])&(lambda[k]<=U.WALD[i]))
            {count1[i]=1}
        if((lambda[k]>=L.SCORE[i])&(lambda[k]<=U.SCORE[i]))
            {count2[i]=1}
        if((lambda[k]>=L.ADJUST[i])&(lambda[k]<=U.ADJUST[i]))
            {count3[i]=1}
        if((lambda[k]>=L.BAYE1[i])&(lambda[k]<=U.BAYE1[i]))
            {count4[i]=1}
        if((lambda[k]>=L.BAYE2[i])&(lambda[k]<=U.BAYE2[i]))
            {count5[i]=1}
        if((lambda[k]>=L.BAYE3[i])&(lambda[k]<=U.BAYE3[i]))
            {count6[i]=1}
        if((lambda[k]>=L.BAYE4[i])&(lambda[k]<=U.BAYE4[i]))
            {count7[i]=1}
        length.WALD[i]=U.WALD[i]-L.WALD[i]
        length.SCORE[i]=U.SCORE[i]-L.SCORE[i]
        length.ADJUST[i]=U.ADJUST[i]-L.ADJUST[i]
        length.BAYE1[i]=U.BAYE1[i]-L.BAYE1[i]
        length.BAYE2[i]=U.BAYE2[i]-L.BAYE2[i]
        length.BAYE3[i]=U.BAYE3[i]-L.BAYE3[i]
        length.BAYE4[i]=U.BAYE4[i]-L.BAYE4[i]
    }
    y1[k]=mean(count1)
    y2[k]=mean(count2)
```

```
y3[k]=mean(count3)
y4[k]=mean(count4)
y5[k]=mean(count5)
y6[k]=mean(count6)
y7[k]=mean(count7)
cat("\n"," n = ",n,"  Lambda = ",lambda[k],"\n")
cat(" Baye1=(",alpha1,",",beta1,")       Baye2=(",alpha2,",",beta2,")
Baye3=(",alpha3,",",beta3,")       Baye4=(",alpha4,  "," ,beta4,")"  ,"\n")
cat(" Coverage Prob Wald    = ", mean(count1),"  Length Wald    = " ,
mean(length.WALD), "\n")
cat(" Coverage Prob Score   = ", mean(count2),"  Length Score   = " ,
mean(length.SCORE), "\n")
cat(" Coverage Prob Adjust  = ", mean(count3),"  Length Adjust  = " ,
mean(length.ADJUST), "\n")
cat(" Coverage Prob Baye1   = ", mean(count4),"  Length Baye1   = " ,
mean(length.BAYE1), "\n")
cat(" Coverage Prob Baye2   = ", mean(count5),"  Length Baye2   = " ,
mean(length.BAYE2), "\n")
cat(" Coverage Prob Baye3   = ", mean(count6),"  Length Baye3   = " ,
mean(length.BAYE3), "\n")
cat(" Coverage Prob Baye4   = ", mean(count7),"  Length Baye4   = " ,
mean(length.BAYE4), "\n")
}
main_title=paste("n = ",n)
plot(lambda,y1,type="o",lwd=2,col="black",main=main_title,ylab="Covera
ge Probability : CP",xlim=c(0.2,1.2),ylim=c(0.98,0.995),pch=20,lty=1)
lines(lambda,y2,type="o",lwd=2,col="red",pch=20,lty=1)
lines(lambda,y3,type="o",lwd=2,col="green",pch=20,lty=1)
```

```
lines(lambda,y4,type="o",lwd=2,col="blue",pch=20,lty=1)
lines(lambda,y5,type="o",lwd=2,col="dark red",pch=20,lty=1)
lines(lambda,y6,type="o",lwd=2,col="yellow",pch=20,lty=1)
lines(lambda,y7,type="o",lwd=2,col="pink",pch=20,lty=1)
legend("bottomright",c("Wald","Score","Adjust","Baye(1,1)","Baye(1,2)","Ba
ye(2,1)","Baye(2,2)"),
lty=c(1,1,1,1,1,1,1),col=c("black","red","green","blue","dark
red","yellow","pink"),lwd=2,cex=0.93)
}
```

A comparison on confidence intervals estimation methods for the geometric population mean.

```
function (n,M,alpha,alpha1,beta1,alpha2,beta2,alpha3,beta3,alpha4,beta4)

{       ###Written by Rungtawan , 17/06/09
        ###n is a sample size.
        ###M is a number of simulation runs.
        ###alpha is a significance level.
        ###alpha1,beta1,alpha2,beta2,alpha3,beta3,alpha4 and beta4 are parameter of
        ###beta distribution

                L.WALD=rep(0,M)
                U.WALD=rep(0,M)
                L.SCORE=rep(0,M)
                U.SCORE=rep(0,M)
                L.ADJUST=rep(0,M)
                U.ADJUST=rep(0,M)
                L.BAYE1=rep(0,M)
                U.BAYE1=rep(0,M)
                L.BAYE2=rep(0,M)
                U.BAYE2=rep(0,M)
                L.BAYE3=rep(0,M)
                U.BAYE3=rep(0,M)
                L.BAYE4=rep(0,M)
                U.BAYE4=rep(0,M)
                length.WALD=rep(0,M)
                length.SCORE=rep(0,M)
                length.ADJUST=rep(0,M)
                length.BAYE1=rep(0,M)
                length.BAYE2=rep(0,M)
```

```
length.BAYE3=rep(0,M)
length.BAYE4=rep(0,M)

p<-c(0.1,0.3,0.5,0.7,0.9)
y1=rep(0,5)
y2=rep(0,5)
y3=rep(0,5)
y4=rep(0,5)
y5=rep(0,5)
y6=rep(0,5)
y7=rep(0,5)

for(k in 1:length(p))
{
        count1=rep(0,M)
        count2=rep(0,M)
        count3=rep(0,M)
        count4=rep(0,M)
        count5=rep(0,M)
        count6=rep(0,M)
        count7=rep(0,M)

        for(i in 1:M)
        {       y=rgeom(n,p[k])
                x=y+1
                z=qnorm(1-alpha/2)
######## compute upper and lower bound
                p_hat=1/mean(x)
                L.WALD[i]=(1/p_hat)-z*sqrt((1-p_hat)/(n*p_hat^2))
                U.WALD[i]=(1/p_hat)+z*sqrt((1-p_hat)/(n*p_hat^2))
```

```
L.SCORE[i]=((n/p_hat)-((z^2)/2)-z*sqrt((z^2)/4-
(n/p_hat)*(1-1/p_hat)))/(n-z^2)
U.SCORE[i]=((n/p_hat)-((z^2)/2)+z*sqrt((z^2)/4-
(n/p_hat)*(1-1/p_hat)))/(n-z^2)

p_adjust=(n-z^2)/(sum(x)-(z^2)/2)
L.ADJUST[i]=1/p_adjust-z*sqrt((1-p_adjust)/((n-
z^2)*(p_adjust^2)))
U.ADJUST[i]=1/p_adjust+z*sqrt((1-p_adjust)/((n-
z^2)*(p_adjust^2)))

a1=2*(beta1+sum(x)-n)
b1=2*(alpha1+n)
L.BAYE1[i]=(2*(alpha1+n)+2*(beta1+sum(x)-
n)*qf(alpha/2,a1,b1))/(2*(alpha1+n))
U.BAYE1[i]=(2*(alpha1+n)+2*(beta1+sum(x)-n)*qf(1-
alpha/2,a1,b1))/(2*(alpha1+n))

a2=2*(beta2+sum(x)-n)
b2=2*(alpha2+n)
L.BAYE2[i]=(2*(alpha2+n)+2*(beta2+sum(x)-
n)*qf(alpha/2,a2,b2))/(2*(alpha2+n))
U.BAYE2[i]=(2*(alpha2+n)+2*(beta2+sum(x)-n)*qf(1-
alpha/2,a2,b2))/(2*(alpha2+n))

a3=2*(beta3+sum(x)-n)
b3=2*(alpha3+n)
L.BAYE3[i]=(2*(alpha3+n)+2*(beta3+sum(x)-
n)*qf(alpha/2,a3,b3))/(2*(alpha3+n))
U.BAYE3[i]=(2*(alpha3+n)+2*(beta3+sum(x)-n)*qf(1-
alpha/2,a3,b3))/(2*(alpha3+n))
```

```
a4=2*(beta4+sum(x)-n)
b4=2*(alpha4+n)
L.BAYE4[i]=(2*(alpha4+n)+2*(beta4+sum(x)-
n)*qf(alpha/2,a4,b4))/(2*(alpha4+n))
U.BAYE4[i]=(2*(alpha4+n)+2*(beta4+sum(x)-n)*qf(1-
alpha/2,a4,b4))/(2*(alpha4+n))
######## compute coverage probability and average length
if((1/p[k]>=L.WALD[i])&(1/p[k]<=U.WALD[i]))
    {count1[i]=1}
if((1/p[k]>=L.SCORE[i])&(1/p[k]<=U.SCORE[i]))
    {count2[i]=1}
if((1/p[k]>=L.ADJUST[i])&(1/p[k]<=U.ADJUST[i]))
    {count3[i]=1}
if((1/p[k]>=L.BAYE1[i])&(1/p[k]<=U.BAYE1[i]))
    {count4[i]=1}
if((1/p[k]>=L.BAYE2[i])&(1/p[k]<=U.BAYE2[i]))
    {count5[i]=1}
if((1/p[k]>=L.BAYE3[i])&(1/p[k]<=U.BAYE3[i]))
    {count6[i]=1}
if((1/p[k]>=L.BAYE4[i])&(1/p[k]<=U.BAYE4[i]))
    {count7[i]=1}
length.WALD[i]=U.WALD[i]-L.WALD[i]
length.SCORE[i]=U.SCORE[i]-L.SCORE[i]
length.ADJUST[i]=U.ADJUST[i]-L.ADJUST[i]
length.BAYE1[i]=U.BAYE1[i]-L.BAYE1[i]
```

```
                length.BAYE2[i]=U.BAYE2[i]-L.BAYE2[i]
                length.BAYE3[i]=U.BAYE3[i]-L.BAYE3[i]
                length.BAYE4[i]=U.BAYE4[i]-L.BAYE4[i]
            }
            y1[k]=mean(count1)
            y2[k]=mean(count2)
            y3[k]=mean(count3)
            y4[k]=mean(count4)
            y5[k]=mean(count5)
            y6[k]=mean(count6)
            y7[k]=mean(count7)
        cat("\n"," n = ",n,"   E(X) = 1/p  = ",1/p[k],"\n")
        cat(" Baye1=(",alpha1,",",beta1,")       Baye2=(",alpha2,",",beta2,")
Baye3=(",alpha3,",",beta3,")       Baye4=(",alpha4,  "," ,beta4,")"  ,"\n")
        cat(" Coverage Prob Wald    = ", mean(count1),"  Length Wald    = " ,
mean(length.WALD), "\n")
        cat(" Coverage Prob Score   = ", mean(count2),"  Length Score   = " ,
mean(length.SCORE), "\n")
        cat(" Coverage Prob Adjust  = ", mean(count3),"  Length Adjust  = " ,
mean(length.ADJUST), "\n")
        cat(" Coverage Prob Baye1   = ", mean(count4),"  Length Baye1   = " ,
mean(length.BAYE1), "\n")
        cat(" Coverage Prob Baye2   = ", mean(count5),"  Length Baye2   = " ,
mean(length.BAYE2), "\n")
        cat(" Coverage Prob Baye3   = ", mean(count6),"  Length Baye3   = " ,
mean(length.BAYE3), "\n")
        cat(" Coverage Prob Baye4   = ", mean(count7),"  Length Baye4   = " ,
mean(length.BAYE4), "\n")
        }
```

```
main_title=paste("n = ",n)
plot(p,y1,type="o",lwd=2,col="black",main=main_title,ylab="Coverage
Probability : CP",xlim=c(0.1,1.15),ylim=c(0.8,1),pch=20,lty=1)
lines(p,y2,type="o",lwd=2,col="red",pch=20,lty=1)
lines(p,y3,type="o",lwd=2,col="green",pch=20,lty=1)
lines(p,y4,type="o",lwd=2,col="blue",pch=20,lty=1)
lines(p,y5,type="o",lwd=2,col="dark red",pch=20,lty=1)
lines(p,y6,type="o",lwd=2,col="yellow",pch=20,lty=1)
lines(p,y7,type="o",lwd=2,col="pink",pch=20,lty=1)
legend("bottomright",c("Wald","Score","Adjust","Baye(0.5,2)","Baye(1,1)","B
aye(1,2)","Baye(2,1)"),
lty=c(1,1,1,1,1,1,1),col=c("black","red","green","blue","dark
red","yellow","pink"),lwd=2,cex=0.93)

}
```

A comparison on confidence intervals estimation methods for the exponential population mean.

```
function (n,M,alpha,alpha1,beta1,alpha2,beta2,alpha3,beta3,alpha4,beta4)

{       ###Written by Rungtawan , 20/06/09
        ###n is a sample size.
        ###M is a number of simulation runs.
        ###alpha is a significance level.
        ###alpha1,beta1,alpha2,beta2,alpha3,beta3,alpha4 and beta4 are parameter of
        ###gamma distribution

            beta_hat=rep(0,M)
            z=rep(0,M)
            beta_adjust=rep(0,M)
            L.WALD=rep(0,M)
            U.WALD=rep(0,M)
            L.SCORE=rep(0,M)
            U.SCORE=rep(0,M)
            L.ADJUST=rep(0,M)
            U.ADJUST=rep(0,M)
            L.BAYE1=rep(0,M)
            U.BAYE1=rep(0,M)
            L.BAYE2=rep(0,M)
            U.BAYE2=rep(0,M)
            L.BAYE3=rep(0,M)
            U.BAYE3=rep(0,M)
            L.BAYE4=rep(0,M)
            U.BAYE4=rep(0,M)
            L.B_SCORE=rep(0,M)
            U.B_SCORE=rep(0,M)
```

```
L.B_BAYE1=rep(0,M)
U.B_BAYE1=rep(0,M)
L.B_BAYE2=rep(0,M)
U.B_BAYE2=rep(0,M)
L.B_BAYE3=rep(0,M)
U.B_BAYE3=rep(0,M)
L.B_BAYE4=rep(0,M)
U.B_BAYE4=rep(0,M)
length.WALD=rep(0,M)
length.SCORE=rep(0,M)
length.ADJUST=rep(0,M)
length.BAYE1=rep(0,M)
length.BAYE2=rep(0,M)
length.BAYE3=rep(0,M)
length.BAYE4=rep(0,M)
length.B_SCORE=rep(0,M)
length.B_BAYE1=rep(0,M)
length.B_BAYE2=rep(0,M)
length.B_BAYE3=rep(0,M)
length.B_BAYE4=rep(0,M)

beta<-c(1,5,10,20,30)
y1=rep(0,5)
y2=rep(0,5)
y3=rep(0,5)
y4=rep(0,5)
y5=rep(0,5)
y6=rep(0,5)
y7=rep(0,5)
y8=rep(0,5)
```

```
    y9=rep(0,5)
    y10=rep(0,5)
    y11=rep(0,5)
    y12=rep(0,5)

    for(k in 1:length(beta))
    {
        count1=rep(0,M)
        count2=rep(0,M)
        count3=rep(0,M)
        count4=rep(0,M)
        count5=rep(0,M)
        count6=rep(0,M)
        count7=rep(0,M)
        count8=rep(0,M)
        count9=rep(0,M)
        count10=rep(0,M)
        count11=rep(0,M)
        count12=rep(0,M)

        for(i in 1:M)
        {       x=rexp(n,1/beta[k])
                z=qnorm(1-alpha/2)

######## compute upper and lower bound

                beta_hat=mean(x)
                L.WALD[i]=beta_hat-z*sqrt((beta_hat^2)/n)
                U.WALD[i]=beta_hat+z*sqrt((beta_hat^2)/n)

                L.SCORE[i]=(n*beta_hat-z*sqrt(n*beta_hat^2))/(n-z^2)
                U.SCORE[i]=(n*beta_hat+z*sqrt(n*beta_hat^2))/(n-z^2)
```

```
beta_adjust=sum(x)/(n-(z^2))
L.ADJUST[i]=beta_adjust-z*sqrt(beta_adjust^2/(n-z^2))
U.ADJUST[i]=beta_adjust+z*sqrt(beta_adjust^2/(n-z^2))

df1=2*(alpha1+n)
L.BAYE1[i]=2*(beta1*sum(x)+1)/(beta1*qchisq(1-alpha/2,df1))
U.BAYE1[i]=2*(beta1*sum(x)+1)/(beta1*qchisq(alpha/2,df1))

df2=2*(alpha2+n)
L.BAYE2[i]=2*(beta2*sum(x)+1)/(beta2*qchisq(1-alpha/2,df2))
U.BAYE2[i]=2*(beta2*sum(x)+1)/(beta2*qchisq(alpha/2,df2))

df3=2*(alpha3+n)
L.BAYE3[i]=2*(beta3*sum(x)+1)/(beta3*qchisq(1-alpha/2,df3))
U.BAYE3[i]=2*(beta3*sum(x)+1)/(beta3*qchisq(alpha/2,df3))

df4=2*(alpha4+n)
L.BAYE4[i]=2*(beta4*sum(x)+1)/(beta4*qchisq(1-alpha/2,df4))
U.BAYE4[i]=2*(beta4*sum(x)+1)/(beta4*qchisq(alpha/2,df4))

######## compute coverage probability and average length

if((beta[k]>=L.WALD[i])&(beta[k]<=U.WALD[i]))
    {count1[i]=1}

if((beta[k]>=L.SCORE[i])&(beta[k]<=U.SCORE[i]))
    {count2[i]=1}

if((beta[k]>=L.ADJUST[i])&(beta[k]<=U.ADJUST[i]))
    {count3[i]=1}

if((beta[k]>=L.BAYE1[i])&(beta[k]<=U.BAYE1[i]))
    {count4[i]=1}
```

```
        if((beta[k]>=L.BAYE2[i])&(beta[k]<=U.BAYE2[i]))
                {count5[i]=1}

        if((beta[k]>=L.BAYE3[i])&(beta[k]<=U.BAYE3[i]))
                {count6[i]=1}

        if((beta[k]>=L.BAYE4[i])&(beta[k]<=U.BAYE4[i]))
                {count7[i]=1}

        length.WALD[i]=U.WALD[i]-L.WALD[i]
        length.SCORE[i]=U.SCORE[i]-L.SCORE[i]
        length.ADJUST[i]=U.ADJUST[i]-L.ADJUST[i]
        length.BAYE1[i]=U.BAYE1[i]-L.BAYE1[i]
        length.BAYE2[i]=U.BAYE2[i]-L.BAYE2[i]
        length.BAYE3[i]=U.BAYE3[i]-L.BAYE3[i]
        length.BAYE4[i]=U.BAYE4[i]-L.BAYE4[i]

#---Bootstrap Estimation --------------#
        L.b_score=rep(0,200); U.b_score=rep(0,200)
        L.b_baye1=rep(0,200); U.b_baye1=rep(0,200)
        L.b_baye2=rep(0,200); U.b_baye2=rep(0,200)
        L.b_baye3=rep(0,200); U.b_baye3=rep(0,200)
        L.b_baye4=rep(0,200); U.b_baye4=rep(0,200)

        for(j in 1:200)
        {
######## compute upper and lower bound  of bootstrap

                xBoots=sample(x,n,replace=TRUE)
                beta_hat_boot=mean(xBoots)
```

```
                L.b_score[j]=(n*beta_hat_boot-
z*sqrt(n*beta_hat_boot^2))/(n-z^2)
                U.b_score[j]=(n*beta_hat_boot+
z*sqrt(n*beta_hat_boot^2))/(n-z^2)

                df1=2*(alpha1+n)
                L.b_baye1[j]=2*(beta1*sum(xBoots)+1)/
(beta1*qchisq(1-alpha/2,df1))
                U.b_baye1[j]=2*(beta1*sum(xBoots)+1)/
(beta1*qchisq(alpha/2,df1))

                df2=2*(alpha2+n)
                L.b_baye2[j]=2*(beta2*sum(xBoots)+1)/
(beta2*qchisq(1-alpha/2,df2))
                U.b_baye2[j]=2*(beta2*sum(xBoots)+1)/
(beta2*qchisq(alpha/2,df2))

                df3=2*(alpha3+n)
                L.b_baye3[j]=2*(beta3*sum(xBoots)+1)/
(beta3*qchisq(1-alpha/2,df3))
                U.b_baye3[j]=2*(beta3*sum(xBoots)+1)/
(beta3*qchisq(alpha/2,df3))

                df4=2*(alpha4+n)
                L.b_baye4[j]=2*(beta4*sum(xBoots)+1)/
(beta4*qchisq(1-alpha/2,df4))
                U.b_baye4[j]=2*(beta4*sum(xBoots)+1)/
(beta4*qchisq(alpha/2,df4))

            }
```

```
######## compute coverage probability and average length of bootstrap
            L.B_SCORE[i]=mean(L.b_score);
U.B_SCORE[i]=mean(U.b_score)
            L.B_BAYE1[i]=mean(L.b_baye1);
U.B_BAYE1[i]=mean(U.b_baye1)
            L.B_BAYE2[i]=mean(L.b_baye2);
U.B_BAYE2[i]=mean(U.b_baye2)
            L.B_BAYE3[i]=mean(L.b_baye3);
U.B_BAYE3[i]=mean(U.b_baye3)
            L.B_BAYE4[i]=mean(L.b_baye4);
U.B_BAYE4[i]=mean(U.b_baye4)
            if((beta[k]>=L.B_SCORE[i])&(beta[k]<=U.B_SCORE[i]))
                {count8[i]=1}

            if((beta[k]>=L.B_BAYE1[i])&(beta[k]<=U.B_BAYE1[i]))
                {count9[i]=1}

            if((beta[k]>=L.B_BAYE2[i])&(beta[k]<=U.B_BAYE2[i]))
                {count10[i]=1}

            if((beta[k]>=L.B_BAYE3[i])&(beta[k]<=U.B_BAYE3[i]))
                {count11[i]=1}

            if((beta[k]>=L.B_BAYE4[i])&(beta[k]<=U.B_BAYE4[i]))
                {count12[i]=1}

            length.B_SCORE[i]=U.B_SCORE[i]-L.B_SCORE[i]
            length.B_BAYE1[i]=U.B_BAYE1[i]-L.B_BAYE1[i]
            length.B_BAYE2[i]=U.B_BAYE2[i]-L.B_BAYE2[i]
            length.B_BAYE3[i]=U.B_BAYE3[i]-L.B_BAYE3[i]
            length.B_BAYE4[i]=U.B_BAYE4[i]-L.B_BAYE4[i]
    }
```

```
y1[k]=mean(count1)
y2[k]=mean(count2)
y3[k]=mean(count3)
y4[k]=mean(count4)
y5[k]=mean(count5)
y6[k]=mean(count6)
y7[k]=mean(count7)
y8[k]=mean(count8)
y9[k]=mean(count9)
y10[k]=mean(count10)
y11[k]=mean(count11)
y12[k]=mean(count12)
cat("\n"," n = ",n,"  Beta = ",beta[k],"\n")
cat(" Baye1=(",alpha1,",",beta1,")       Baye2=(",alpha2,",",beta2,")
Baye3=(",alpha3,",",beta3,")       Baye4=(",alpha4,  "," ,beta4,")"  ,"\n")
cat(" Coverage Prob Wald    = ", mean(count1),"  Length Wald   = " ,
mean(length.WALD), "\n")
cat(" Coverage Prob Score   = ", mean(count2),"  Length Score  = " ,
mean(length.SCORE), "\n")
cat(" Coverage Prob Adjust  = ", mean(count3),"  Length Adjust = " ,
mean(length.ADJUST), "\n")
cat(" Coverage Prob Baye1   = ", mean(count4),"  Length Baye1  = " ,
mean(length.BAYE1), "\n")
cat(" Coverage Prob Baye2   = ", mean(count5),"  Length Baye2  = " ,
mean(length.BAYE2), "\n")
cat(" Coverage Prob Baye3   = ", mean(count6),"  Length Baye3  = " ,
mean(length.BAYE3), "\n")
cat(" Coverage Prob Baye4   = ", mean(count7),"  Length Baye4  = " ,
mean(length.BAYE4), "\n")
```

```
	cat(" Coverage Prob Bootstrap_Score   = ", mean(count8), "  Length 
Bootstrap_Score  = " , mean(length.B_SCORE), "\n")
	cat(" Coverage Prob Bootstrap_Baye1   = ", mean(count9), "  Length 
Bootstrap_Baye1  = " , mean(length.B_BAYE1), "\n")
	cat(" Coverage Prob Bootstrap_Baye2   = ", mean(count10),"  Length 
Bootstrap_Baye2  = " , mean(length.B_BAYE2), "\n")
	cat(" Coverage Prob Bootstrap_Baye3   = ", mean(count11),"  Length 
Bootstrap_Baye3  = " , mean(length.B_BAYE3), "\n")
	cat(" Coverage Prob Bootstrap_Baye4   = ", mean(count12),"  Length 
Bootstrap_Baye4  = " , mean(length.B_BAYE4), "\n")
	}
	main_title=paste("n = ",n)
	plot(beta,y1,type="o",lwd=2,col="black",main=main_title,ylab="Coverage 
Probability : CP",xlim=c(1,37),ylim=c(0.885,0.92),pch=20,lty=1)
	lines(beta,y2,type="o",lwd=2,col="red",pch=20,lty=1)
	lines(beta,y3,type="o",lwd=2,col="green",pch=20,lty=1)
	lines(beta,y4,type="o",lwd=2,col="blue",pch=20,lty=1)
	lines(beta,y5,type="o",lwd=2,col="dark red",pch=20,lty=1)
	lines(beta,y6,type="o",lwd=2,col="yellow",pch=20,lty=1)
	lines(beta,y7,type="o",lwd=2,col="pink",pch=20,lty=1)
	lines(beta,y8,type="o",lwd=2,col="purple",pch=20,lty=1)
	lines(beta,y9,type="o",lwd=2,col="brown",pch=20,lty=1)
	lines(beta,y10,type="o",lwd=2,col="dark blue",pch=20,lty=1)
	lines(beta,y11,type="o",lwd=2,col="orange",pch=20,lty=1)
	lines(beta,y12,type="o",lwd=2,col="gray",pch=20,lty=1)

	legend("bottomright",c("Wald","Score","Adjust","Baye(0.5,2)","Baye(0.5,3)","Baye
(1,1)","Baye(1,2)","B_Score","B_Baye(0.5,2)","B_Baye(0.5,3)","B_Baye(1,1)","B_Baye(1,2)"),lty
=c(1,1,1,1,1,1,1,1,1,1,1,1),col=c("black","red","green","blue","dark 
red","yellow","pink","purple","brown","dark blue","orange","gray"),lwd=2,cex=0.76)
}
```