STAT0020 ICA

1 Compound process models under LDA framework

1.1 Question A

1.1.1 (a)

Since $X_i(t)$ is a gamma distribution, and assuming $X_i \perp\!\!\!\perp X_j \quad \forall i \neq j$, we have that the distribution of the sum of n-losses is:

$$\sum_{i=1}^{n} X_i(t) = Z_n(t) \sim Gamma(\alpha n, \beta)$$

With density function:

$$f_{Z_n}(z) = \frac{\beta^{\alpha n} z^{\alpha n - 1} e^{-\beta z}}{\Gamma(\alpha n)}$$

and distribution function:

$$F_{Z_n}(z) = \int_0^z f_{Z_n}(u)\,du = \frac{\gamma(n\alpha, \beta z)}{\Gamma(n\alpha)}$$

where $\gamma(\cdot,\cdot)$ is the lower incomplete gamma function defined as:

$$\gamma(s,x) = \int_0^x t^{s-1} \exp(-t)\,dt$$

1.1.2 (b)

Distribution and Density in closed form for the compound process LDA model with Poisson frequency with intensity parameter λ and severity model characterised by Gamma distribution driven equation (1):

We know $X_i(t) \sim Gamma(\alpha, \beta), \quad N(t) \sim Poisson(\lambda), \quad Z_n(t) = \sum_{n=1}^{N} X_n,$

Also:

$$F_{Z_N}(z) = \sum_{k=0}^{\infty} p_k F_{Z_N}(z) = \sum_{k=0}^{\infty} P(N=k) \cdot F_{X_i}^{\star(k)}(z)$$

where

$$F_{X_i}^{\star(k)}(z) = \int_0^z F_{X_i}^{\star(k-1)}(z-x) f_X(x)\, dx$$

and

$$F_{X_i}^{\star(0)}(z) = \quad 1 \quad if \quad z \geq 0, \quad 0 \quad if \quad z \leq 0$$

$\therefore$

$$F_{Z_N}(z) = \sum_{k=0}^{\infty} \frac{\lambda^k \exp(-\lambda)}{k!} \cdot F_{X_i}^{\star(k)}(z)$$

Now we know from part a) that for $k > 0$

$$F_{X_i}^{(k)\star}(z) = P[X_1 + \cdots + X_k \leq z]$$

$$= \int_0^z f_{Z_k}(u)\, du = \frac{\gamma(k\alpha, \beta z)}{\Gamma(k\alpha)}$$

$$F_{Z_N}(z) = \exp(-\lambda) + \sum_{k=1}^{\infty} \frac{\lambda^k \exp(-\lambda)}{k!} \cdot \frac{\gamma(k\alpha, \beta z)}{\Gamma(k\alpha)}$$

is the distribution, where $\gamma(k\alpha, \beta z)$ is defined as in part a).

And the density is

$$f_{Z_N}(z) = \sum_{k=1}^{\infty} \frac{\lambda^k \exp(-\lambda)}{k!} \cdot \frac{\beta^{\alpha k} z^{\alpha k - 1} e^{-\beta z}}{\Gamma(\alpha k)}$$

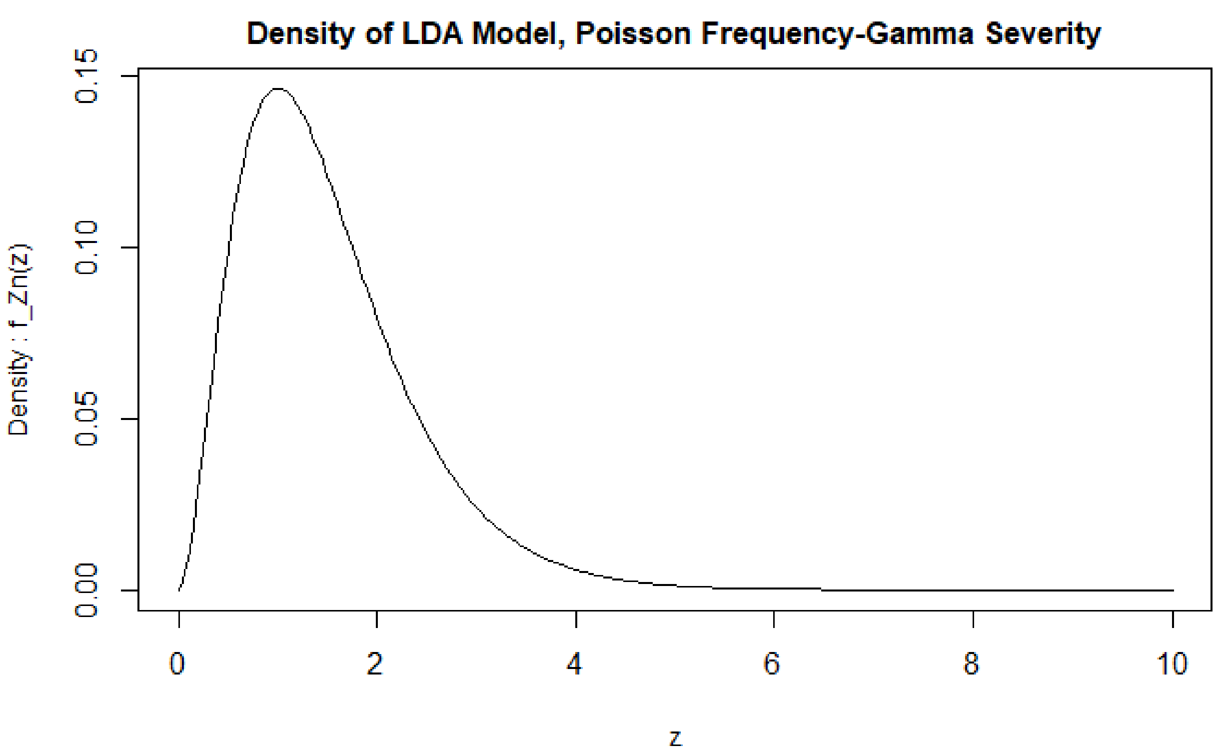
1.1.3 (c)

Hide Code

```
rm(list=ls(all=TRUE))
library(pracma)
```

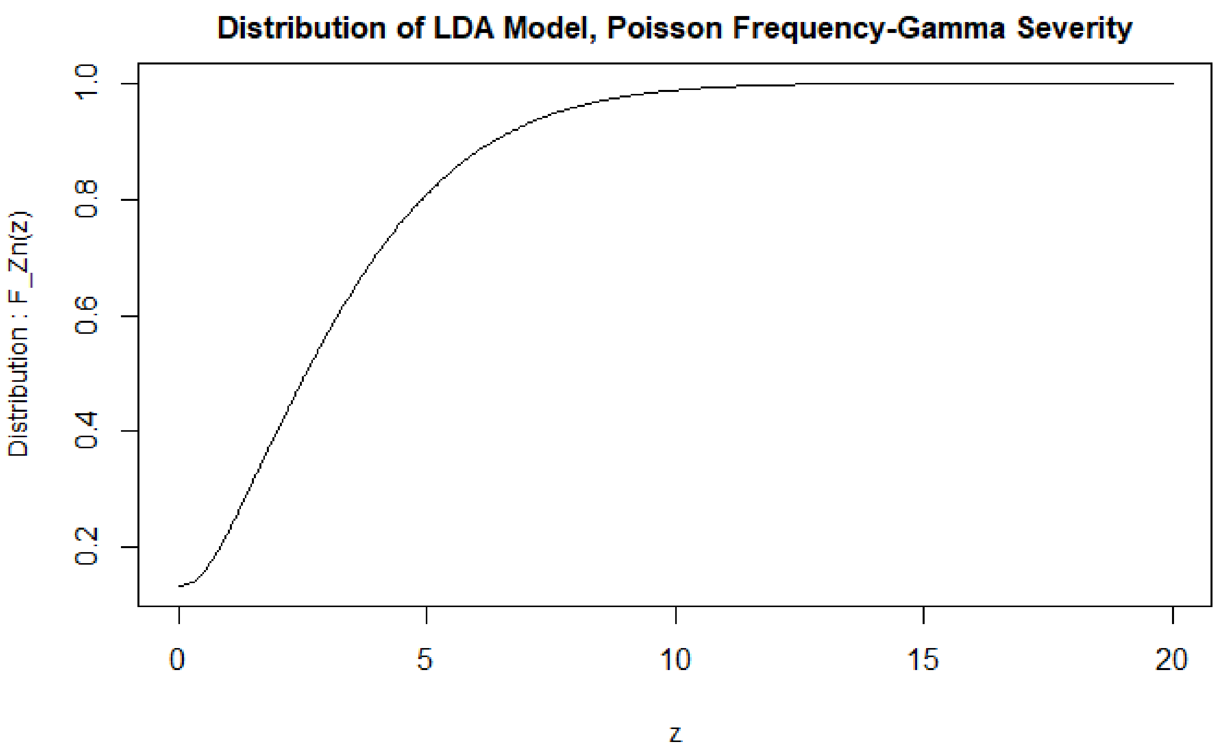
Hide

```
#Setting the parameters
lambda <- 2
bet <- 2
alph <- 3

###PDF

#Defining our density function for Poisson-Gamma distribution
PGdens <- function(z,n,alph,bet,lambda) {partialsum = 0

for(k in 1:n){  #We want a sum, hence we use a for loop
  partialsum = partialsum + dpois(k,lambda)*dgamma(z,alph*k,bet) #Defining the sum using the
 equation we defined

  return(partialsum)
}}

zs1 = seq(0, 10, by = 0.05) #Our range of z values

#Plotting the density function for our parameter values
plot(zs1,PGdens(zs1,200,alph,bet,lambda),main = "Density of LDA Model, Poisson Frequency-Gamm
a Severity",cex.main=1.05 ,xlab = "z",ylab="Density : f_Zn(z)",cex.lab=0.9,"l")
```



Hide

```
####CDF
zs = seq(0, 20, by = 0.05)#Our range of z values

#Defining our distribution function for Poisson-Gamma distribution
PGcdf <- function(z,n,alph,bet,lambda) {partialsum = exp(-lambda)
if(n>0){

  for(k in 1:n){
    partialsum = partialsum +  dpois(k,lambda)* pgamma(z,alph*k,bet)#Defining the sum using t
he equation we defined
  }
}

return(partialsum)
}


plot(zs,PGcdf(zs,50,alph,bet,lambda),main = "Distribution of LDA Model, Poisson Frequency-Gam
ma Severity",cex.main=1.05 ,xlab = "z",ylab="Distribution : F_Zn(z)",cex.lab=0.9,"l")
```

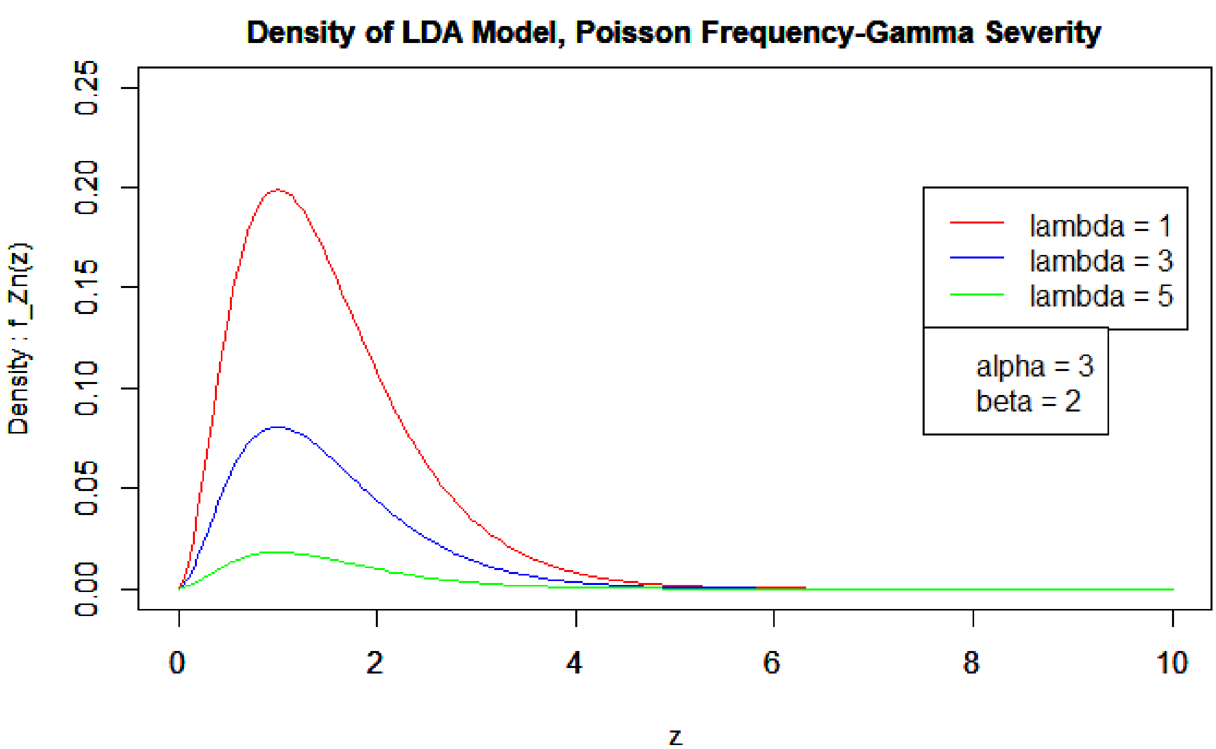


Hide

```
#Testing different values of lambda, alpha and beta

#Lambda tests____________________________________________________
zs1 = seq(0, 10, by = 0.05)
zs2 = seq(0, 20, by = 0.05)

alpha_lambdatest <- 3 #Setting our alpha and beta parameters
beta_lambdatest <- 2

lambdas = c(1,3,5)#We are going to plot the model for lambda = 1, 3 and 5

color = c("red","blue","green")#Defining different colours to see the difference in plots

for(i in 1:3){ #Testing each parameter value for the density function
  pgdens = PGdens(zs1,200,alpha_lambdatest,beta_lambdatest,lambda = lambdas[i])

  #Plotting the density
  plot(zs1,pgdens,main = "Density of LDA Model, Poisson Frequency-Gamma Severity", cex.main=
1.05 ,xlab = "z",ylab="Density : f_Zn(z)",ylim = c(0,0.25),cex.lab=0.9,"l",col = color[i])
  par(new = TRUE)
}
legend(7.5,0.20,legend = c("lambda = 1","lambda = 3","lambda = 5"),col = c("red","blue","gree
n"),lty = 1)
```

Hide

```
legend(7.5,0.13, legend = c("alpha = 3","beta = 2"))
```



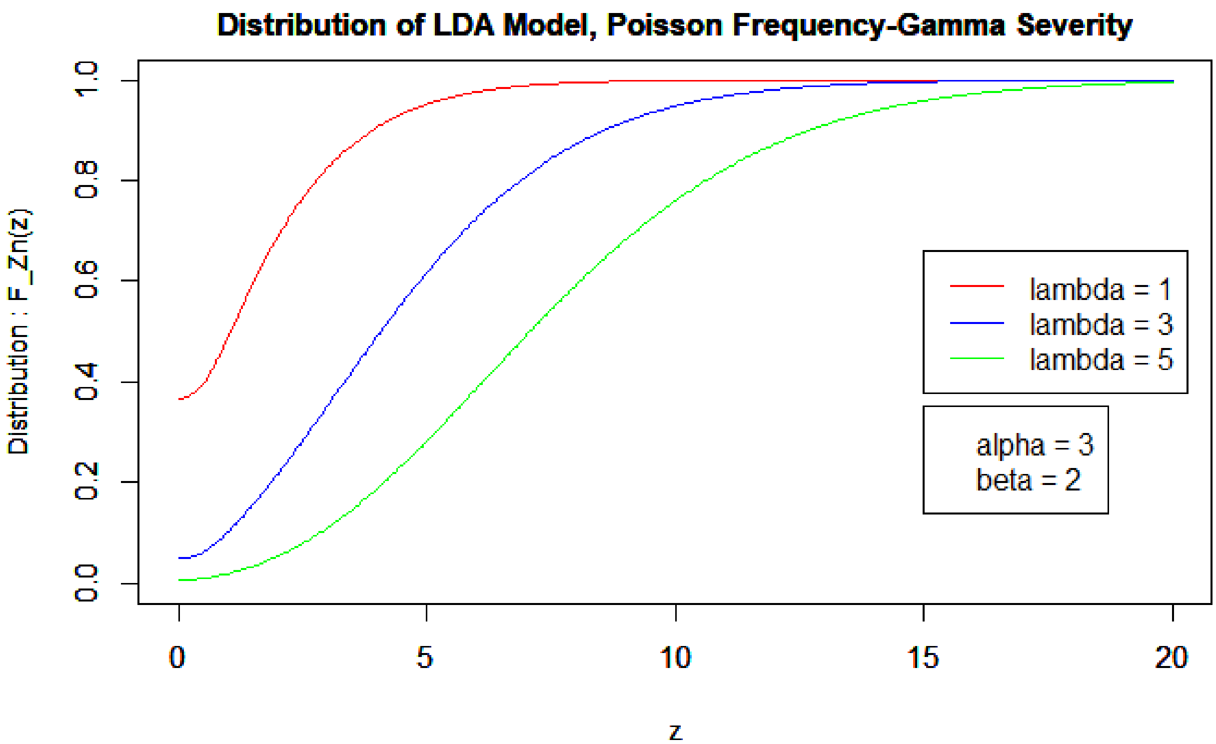
We see that the peak height reduces as lambda increases.

Hide

```
for(i in 1:3){#Testing each parameter value for the distribution function
  pgcdf = PGcdf(zs2,20,alpha_lambdatest,beta_lambdatest,lambda = lambdas[i])

  #Plotting the distribution
  plot(zs2,pgcdf,main = "Distribution of LDA Model, Poisson Frequency-Gamma Severity", cex.ma
in=1.05 ,xlab = "z",ylab="Distribution : F_Zn(z)",ylim = c(0,1),cex.lab=0.9,"l",col = color
[i])
  par(new = TRUE)
}
legend(15,0.66,legend = c("lambda = 1","lambda = 3","lambda = 5"),col = c("red","blue","gree
n"),lty = 1)
```

Hide

```
legend(15,0.35, legend = c("alpha = 3","beta = 2"))
```

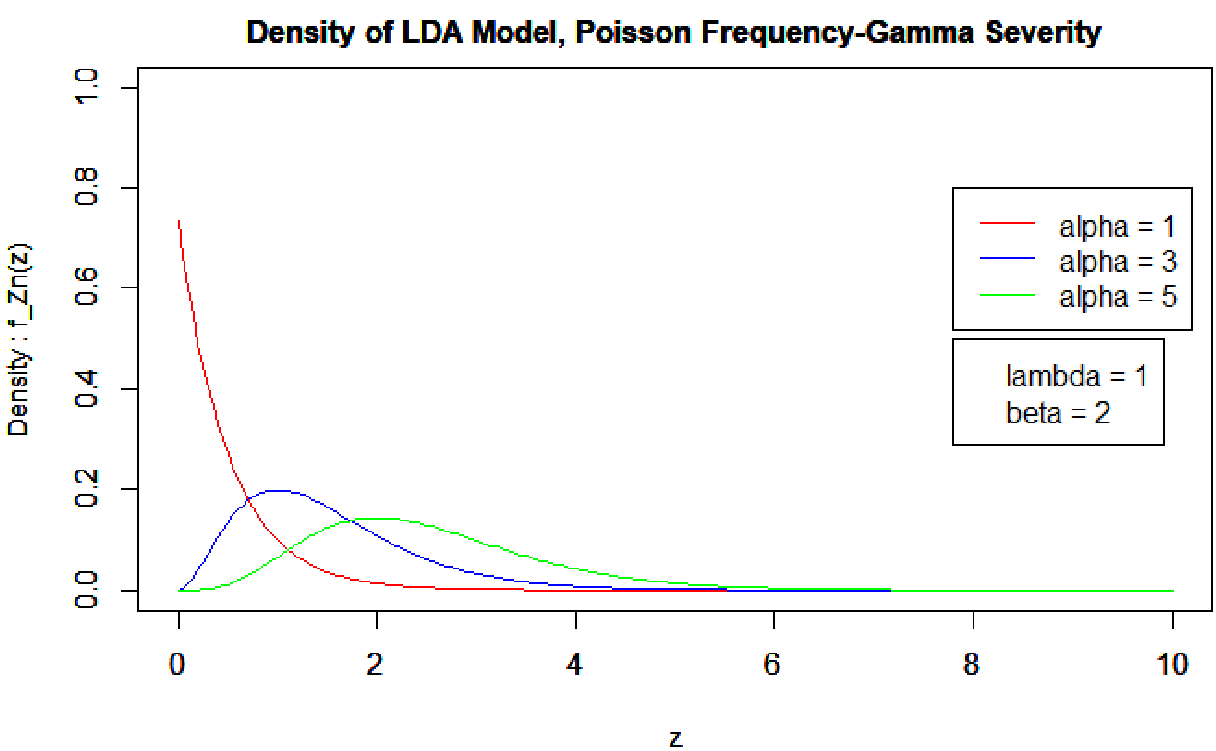


Hide

```
#Alpha tests____________________________________________________
zs1 = seq(0, 10, by = 0.05)
zs2 = seq(0, 20, by = 0.05)

beta_alphatest <- 2 #Setting our beta and lambda parameters
lambda_alphatest <- 1

alphas = c(1,3,5)#We are going to plot the model for alphas = 1, 3 and 5
color = c("red","blue","green")

for(i in 1:3){#Testing each parameter value for the density function
  pgdens = PGdens(zs1,200,alph = alphas[i],beta_alphatest,lambda_alphatest)

  #Plotting the density
  plot(zs1,pgdens,main = "Density of LDA Model, Poisson Frequency-Gamma Severity", cex.main=
1.05 ,xlab = "z",ylab="Density : f_Zn(z)",ylim = c(0,1),cex.lab=0.9,"l",col = color[i])
  par(new = TRUE)
}
legend(7.8,0.80,legend = c("alpha = 1","alpha = 3","alpha = 5"),col = c("red","blue","green"
),lty = 1)
```

Hide

```
legend(7.8,0.5, legend = c("lambda = 1","beta = 2"))
```



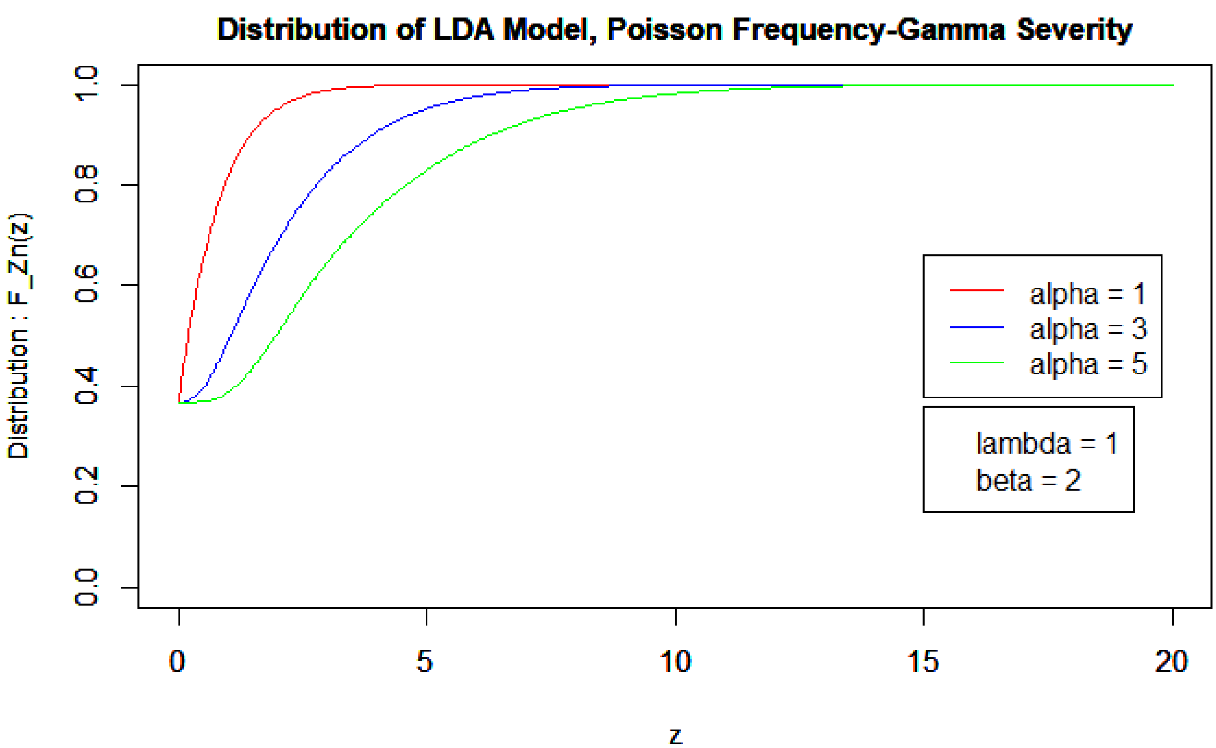
We see that that the peak position shifts to the right as alpha increases.

Hide

```
for(i in 1:3){#Testing each parameter value for the density function
  pgcdf = PGcdf(zs2,20,alph = alphas[i],beta_alphatest,lambda_alphatest)

  #Plotting the distribution
  plot(zs2,pgcdf,main = "Distribution of LDA Model, Poisson Frequency-Gamma Severity", cex.ma
in=1.05 ,xlab = "z",ylab="Distribution : F_Zn(z)",ylim = c(0,1),cex.lab=0.9,"l",col = color
[i])
  par(new = TRUE)
}
legend(15,0.66,legend = c("alpha = 1","alpha = 3","alpha = 5"),col = c("red","blue","green"),
lty = 1)
```

Hide

```
legend(15,0.36, legend = c("lambda = 1","beta = 2"))
```

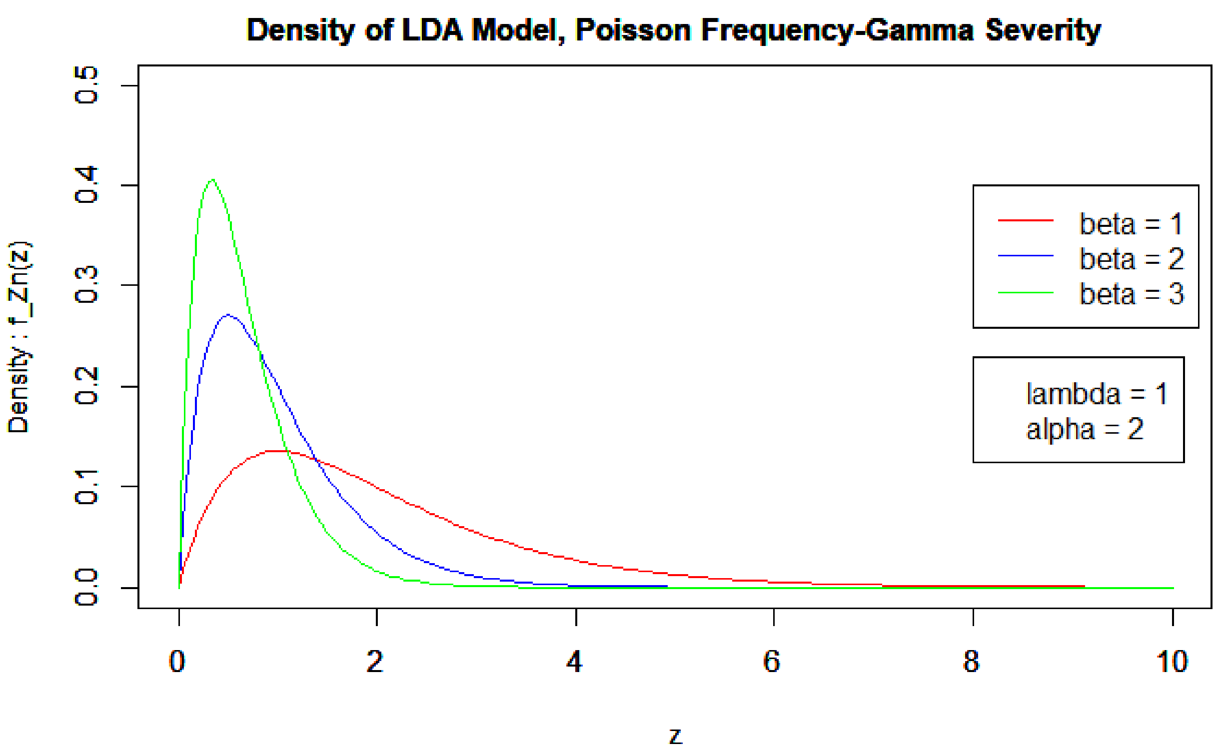


Hide

```
#Beta tests_____________________________________________________
zs1 = seq(0, 10, by = 0.05)
zs2 = seq(0, 20, by = 0.05)
alpha_betatest <- 2 #Setting our alpha and lambda parameters
lambda_betatest <- 1
betas = c(1,2,3)#We are going to plot the model for betas = 1,2 and 3
color = c("red","blue","green")

for(i in 1:3){#Testing each parameter value for the density function
  pgdens = PGdens(zs1,200,alpha_betatest,bet = betas[i],lambda_alphatest)

  #Plotting the density
  plot(zs1,pgdens,main = "Density of LDA Model, Poisson Frequency-Gamma Severity", cex.main=
1.05 ,xlab = "z",ylab="Density : f_Zn(z)",ylim = c(0,0.5),cex.lab=0.9,"l",col = color[i])
  par(new = TRUE)
}
legend(8,0.4,legend = c("beta = 1","beta = 2","beta = 3"),col = c("red","blue","green"),lty =
1)
```

Hide

```
legend(8,0.23, legend = c("lambda = 1","alpha = 2"))
```



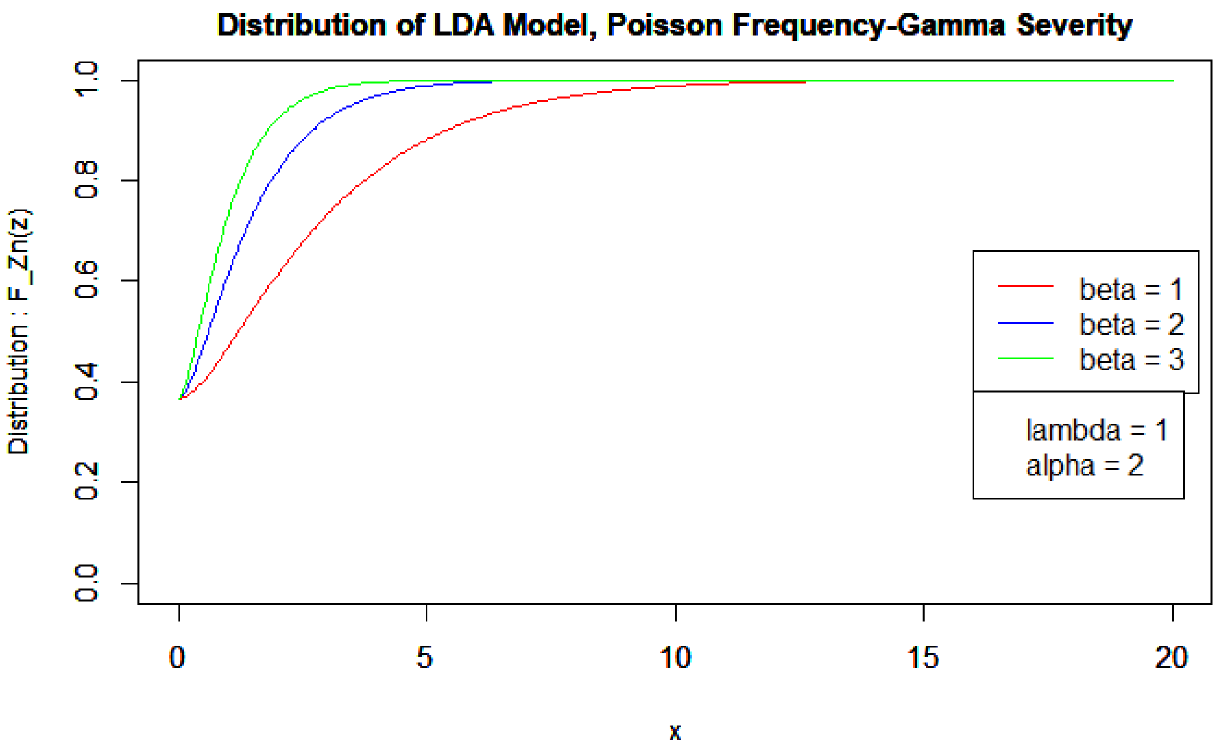
We can see heavier tails as beta gets smaller.

Hide

```
for(i in 1:3){#Testing each parameter value for the distribution function
  pgcdf = PGcdf(zs2,20,alpha_betatest,bet = betas[i],lambda_alphatest)

  #Plotting the distribution
  plot(zs2,pgcdf,main = "Distribution of LDA Model, Poisson Frequency-Gamma Severity", cex.ma
in=1.05 ,xlab = "x",ylab="Distribution : F_Zn(z)",ylim = c(0,1),cex.lab=0.9,"l",col = color
[i])
  par(new = TRUE)
}
legend(16,0.66,legend = c("beta = 1","beta = 2","beta = 3"),col = c("red","blue","green"),lty
= 1)
```

Hide

```
legend(16,0.38, legend = c("lambda = 1","alpha = 2"))
```



1.1.4 (d)

To calculate the compound loss distribution via the Monte Carlo method, follow the steps:

(i) Simulate the annual number of events N from the Poisson frequency distribution;

(ii) Simulate independent severities $X_1, \ldots, X_N$ from the Gamma severity distribution, where N is from step (i);

(iii) Aggregate the losses to obtain a realization of Z_N given by $Z_n = X_1 + ... + X_n$. Repeat these steps K times to get $Z_1, \ldots, Z_K$ independent samples of Z from the Poisson-Gamma compound distribution

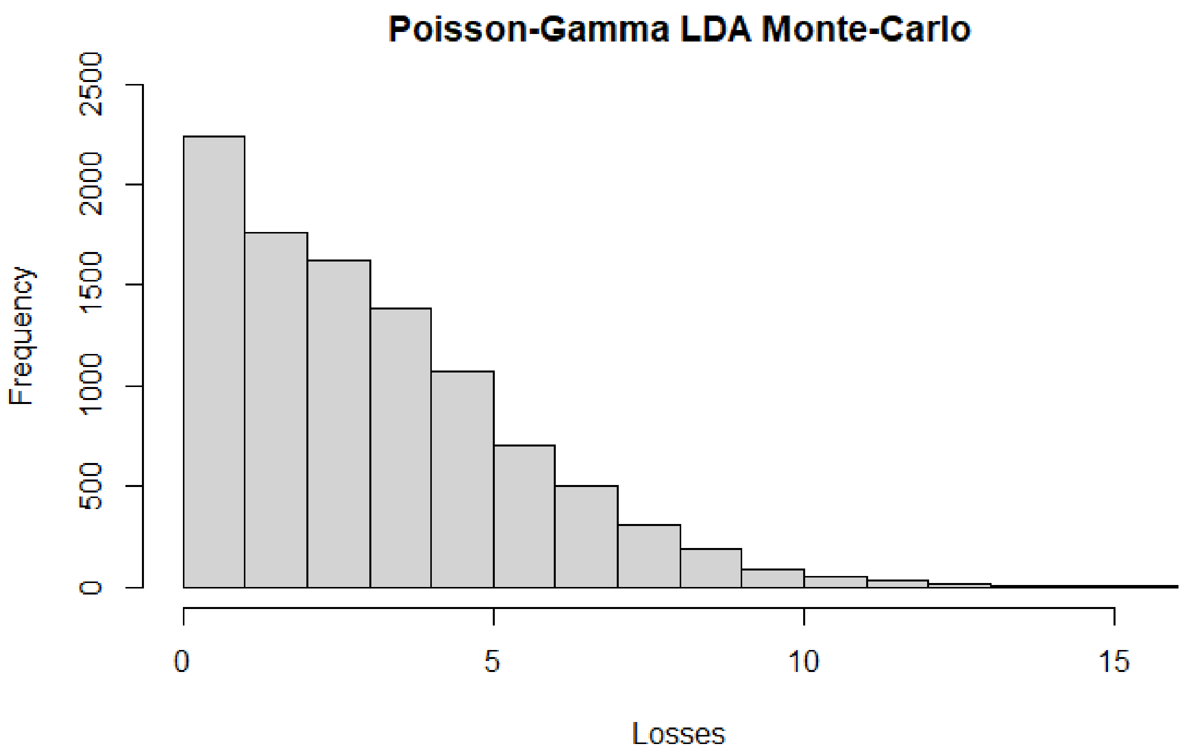
1.1.5 (e)

Hide Code

```
#PART E-------------------------
#Monte Carlo Simulation_______________________________________________________________
lambda <- 2 # Deining intial parameters
bet <- 2
alph <- 3

poisson_gamma_montecarlo <- function(lambda,alpha,beta,nsim){

  annual_loss <- rep(NA,nsim)

  for (i in 1:nsim){ #We want to simulate for n yea so we use a for loop

    no_of_losses <- rpois(1,lambda) #Smpling the number of losses from the poisson distributi
on

    if (no_of_losses == 0){ #If the number of losses in that year is 0 then we want a zero va
lue

      annual_loss[i] <- 0

    } else{

      individual_losses = rgamma(no_of_losses,alpha, beta) #Sampling the individual losses th
at year from the Gamma distribution

      agg_losses = sum(individual_losses) #Aggregating the losses
      annual_loss[i] <- agg_losses #Assigning the loss for that year as the aggregate loss

    }

  }


  return(annual_loss)

}

monty = poisson_gamma_montecarlo(lambda = lambda,alpha = alph,beta = bet, nsim = 10000)
```

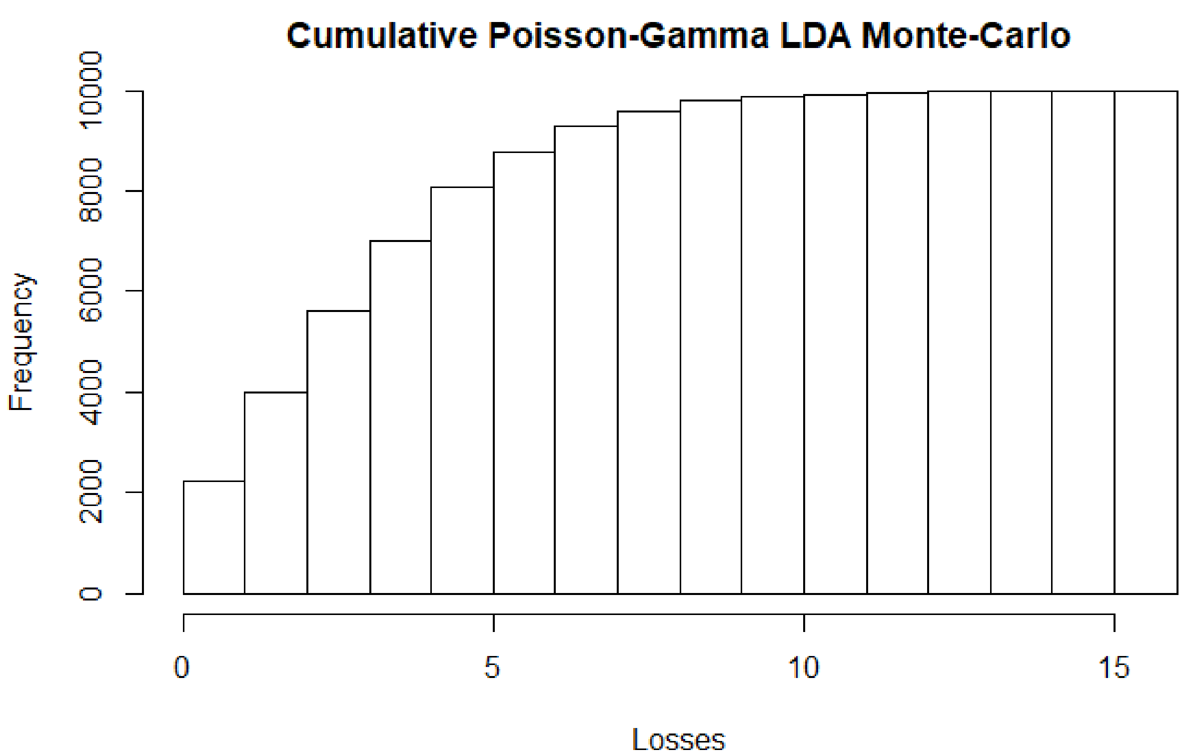
Hide

```
my_hist <- hist(monty,ylim = c(0,2500),xlab='Losses', main='Poisson-Gamma LDA Monte-Carlo') #
Plotting the histogram and Storing histogram info
```



Hide

```
my_hist$counts <- cumsum(my_hist$counts)    # Change histogram counts
plot(my_hist,xlab='Losses',main='Cumulative Poisson-Gamma LDA Monte-Carlo')
```



Hide

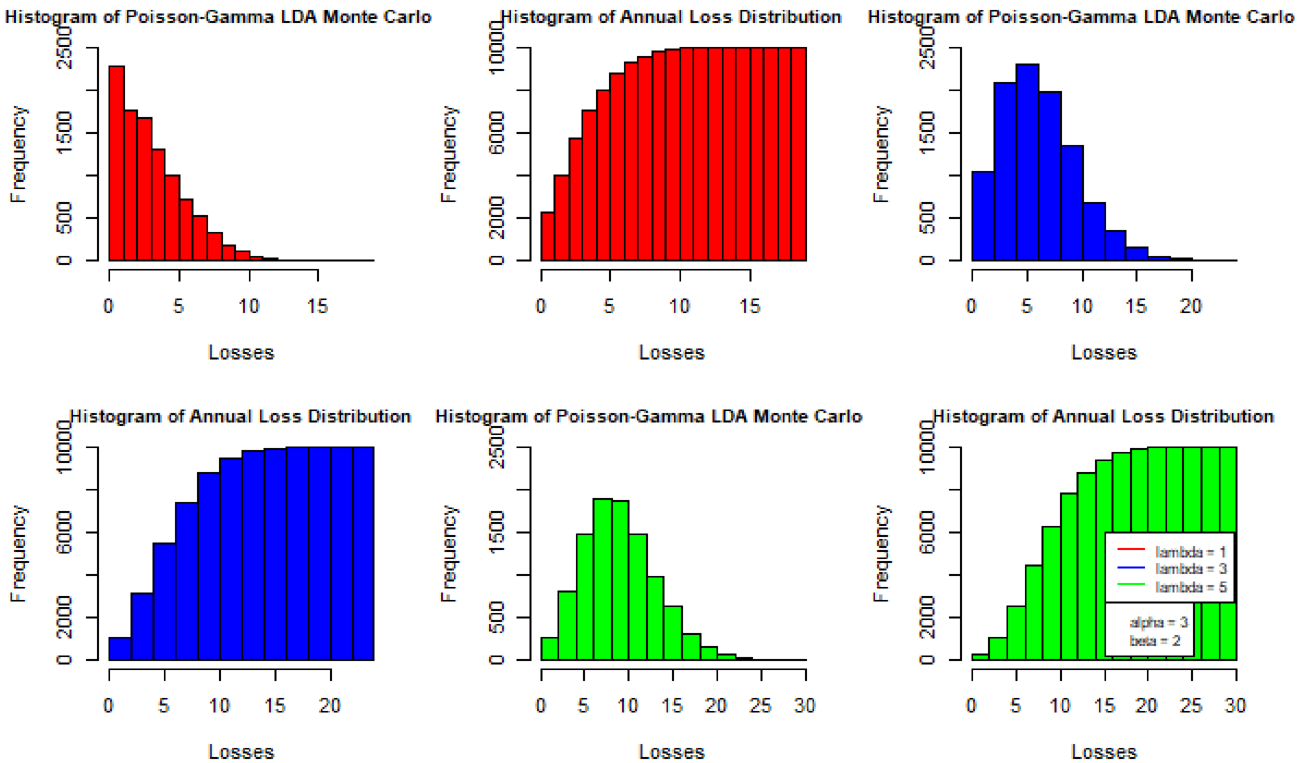
```
#Montecarlo Simulations with Testing
zs1 = seq(0, 10, by = 0.05)
zs2 = seq(0, 20, by = 0.05)

alpha_lambdatest <- 3 #Setting consistent parameters for alpha , beta and lambda for each tes
beta_lambdatest <- 2

alpha_betatest <- 2
lambda_betatest <- 5

beta_alphatest <- 2
lambda_alphatest <- 5

alphas = c(1,3,5)#We are going to plot the model for alpha = 1, 3 and 5
betas = c(1,2,3)#We are going to plot the model for lambda = 1, 2 and 3
lambdas = c(2,4,6)#We are going to plot the model for lambda = 2, 4 and 6

color = c("red","blue","green")
```

Hide

```
#Lambda tests____________________________________________________

par(mfrow=c(2,3))

for(i in 1:3){#Testing the Monte carlo distribution for values of lambda
  monty_test = poisson_gamma_montecarlo(lambda = lambdas[i],alpha = alpha_lambdatest,beta = b
eta_lambdatest, nsim = 10000)
  par(new=FALSE)
  cuml_hist <- hist(monty_test,col = color[i],main = "Histogram of Poisson-Gamma LDA Monte Ca
rlo Sim.",cex.main = 0.94,ylim = c(0,2500),xlab='Losses')


  cuml_hist$counts <- cumsum(cuml_hist$counts)
  plot(cuml_hist,col=color[i],main = "Histogram of Annual Loss Distribution",cex.main = 0.94,
xlab='Losses')
}
legend(15,6000,legend = c("lambda = 1","lambda = 3","lambda = 5"),col = c("red","blue","gree
n"),lty = 1,cex = 0.75)
```

Hide

```
legend(15,2700, legend = c("alpha = 3","beta = 2"),cex = 0.75)
```

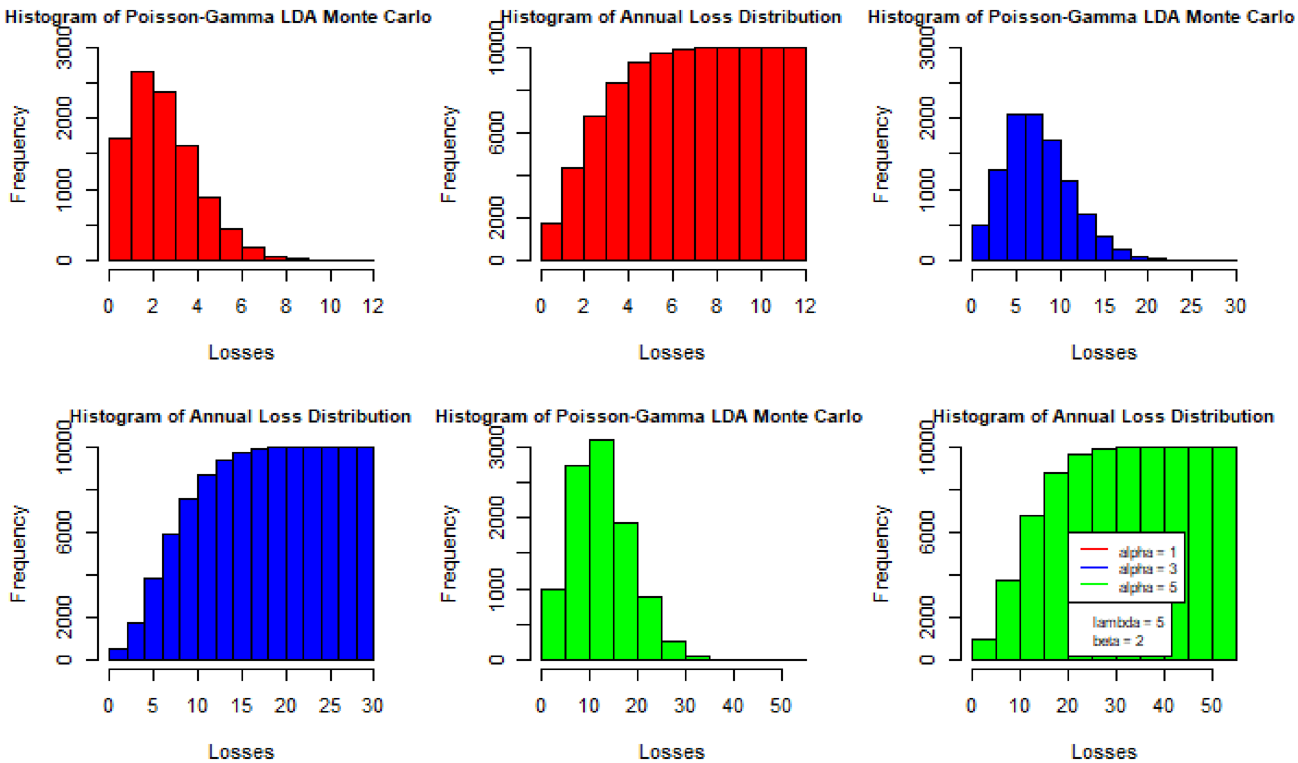


Hide

```
#Alpha tests_____________________________________________________
par(mfrow=c(2,3))
for(i in 1:3){#Testing the Monte carlo distribution for values of alphas
  monty_test = poisson_gamma_montecarlo(lambda_alphatest,alph = alphas[i],beta_alphatest, nsi
m = 10000)
  par(new=FALSE)

  cuml_hist <- hist(monty_test,col = color[i],main = "Histogram of Poisson-Gamma LDA Monte Ca
rlo Sim.",cex.main = 0.94,ylim = c(0,3000), xlab='Losses')


  cuml_hist$counts <- cumsum(cuml_hist$counts)
  plot(cuml_hist,col=color[i],main = "Histogram of Annual Loss Distribution",cex.main = 0.94,
xlab='Losses')
}

legend(20,6000,legend = c("alpha = 1","alpha = 3","alpha = 5"),col = c("red","blue","green"),
lty = 1,cex = 0.75)
```

Hide

```
legend(20,2700, legend = c("lambda = 5","beta = 2"),cex = 0.75)
```

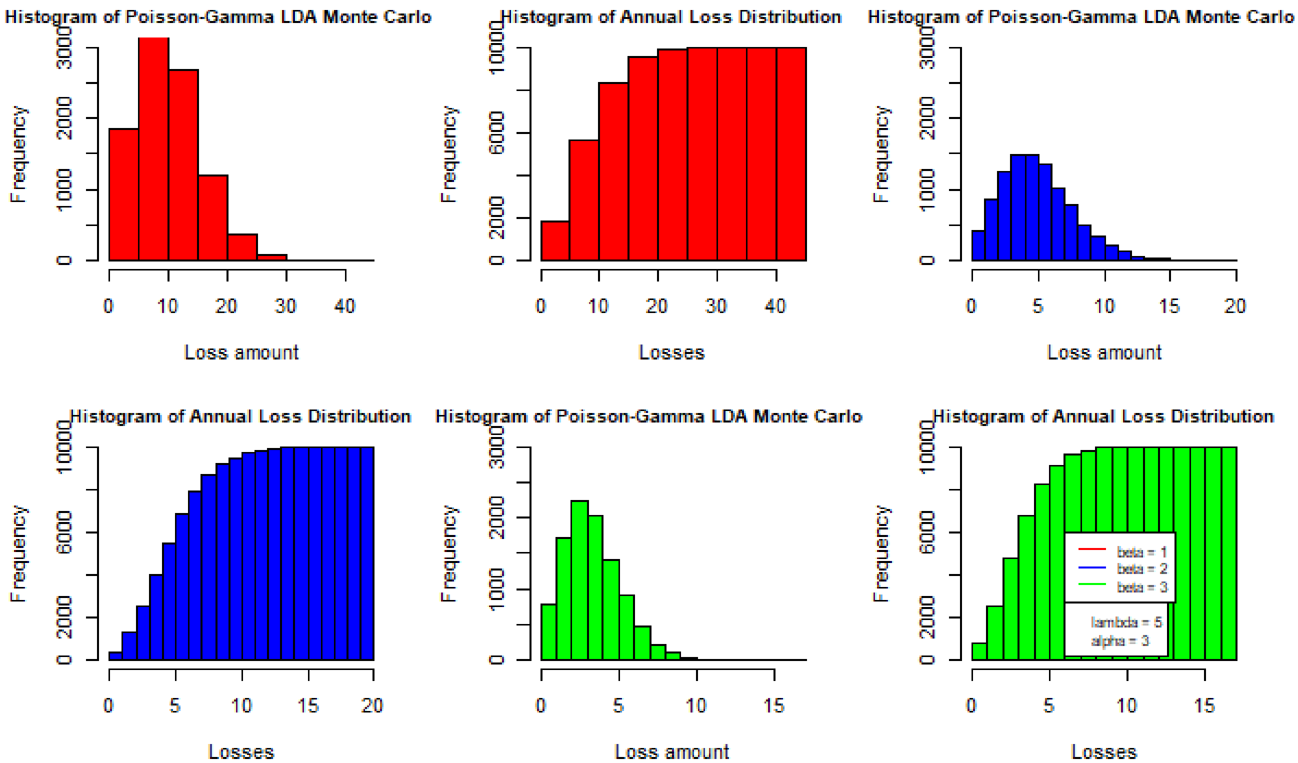


Hide

```
#Beta tests________________________________________________________________
par(mfrow=c(2,3))
for(i in 1:3){#Testing the Monte carlo distribution for values of beta
  monty_test = poisson_gamma_montecarlo(lambda_betatest,alpha_betatest,beta = betas[i], nsim = 10000)
  par(new=FALSE)

  cuml_hist <- hist(monty_test,col = color[i],main = "Histogram of Poisson-Gamma LDA Monte Carlo Sim.",cex.main = 0.94,xlab = "Loss amount",ylab = "Frequency",ylim = c(0,3000))

  cuml_hist$counts <- cumsum(cuml_hist$counts)
  plot(cuml_hist,col=color[i],main = "Histogram of Annual Loss Distribution",cex.main = 0.94,xlab='Losses')
}
legend(6,6000,legend = c("beta = 1","beta = 2","beta = 3"),col = c("red","blue","green"),lty = 1,cex = 0.75)
```

Hide

```
legend(6,2700, legend = c("lambda = 5","alpha = 3"),cex = 0.75)
```



We can see that the Monte-Carlo simulations produce very similar graph shapes to their respective density and distribution graphs.

1.2 Question B

1.2.1 (a)

So we have a severity model for losses in a given business unit risk type given by

$$X_i(t) \sim S_\alpha(\beta, \gamma, \delta; 0)$$

where $\alpha = 0.5$ and $\beta = 1$. The distribution of the linear combination given by $N = n$, is

$$Z_n = \sum_{i=1}^{n} X_i \sim S(\alpha, \tilde{\beta_n}, \tilde{\gamma_n}, \tilde{\delta_n}; 0)$$

Where,

$$\tilde{\gamma_n}^{0.5} = \sum_{i=1}^{n} \gamma_i^{0.5}, \tilde{\beta_n} = \frac{\sum_{i=1}^{n} \gamma_i^{0.5}}{\sum_{i=1}^{n} \gamma_i^{0.5}} = 1$$

and,

$$\begin{aligned} \tilde{\delta_n} &= \sum_{i=1}^{n} \delta_i + tan(\frac{\pi\alpha}{2})(\tilde{\beta_n}\tilde{\gamma_n} - \sum_{i=1}^{n} \beta_i\gamma_i) \\ &= \sum_{i=1}^{n} \delta_i + tan(\frac{0.5\pi}{2})(\tilde{\gamma_n} - \sum_{i=1}^{n} \gamma_i) \end{aligned}$$

So our distribution is,

$$f_{Z_n}(z) = \sqrt{\frac{\tilde{\gamma_n}}{2\pi}} \frac{1}{(z-\delta)^{\frac{3}{2}}} \exp\left(-\frac{\tilde{\gamma_n}}{2(z-\tilde{\delta_n})}\right)$$

$$F_{Z_n}(z) = erfc\left(\sqrt{\frac{\tilde{\gamma_n}}{2(z-\tilde{\delta_n})}}\right)$$

for

$$z \in [\tilde{\delta_n}, \infty)$$

Where

$$erfc(x) = 1 - erf(x) = 1 - \frac{2}{\pi}\int_0^x e^{-t^2} dt$$

1.2.2 (b)

The density is:

$$f_{Z_N}(z) = \sum_{n=1}^{\infty} P_n \left[\sqrt{\frac{\tilde{\gamma_n}}{2\pi}} \frac{1}{(z-\tilde{\delta_n})^{3/2}} \exp{(-\frac{\tilde{\gamma_n}}{2(z-\tilde{\delta_n})})} \cdot I_{[\tilde{\delta_n}<z<\infty]}\right]$$

Where

$$P_n = \exp(-\lambda)\frac{\lambda^n}{n!}, \tilde{\gamma_n}^{0.5} = \sum_{i=1}^{n} \gamma_i^{0.5} = n|\gamma|^{0.5}, \tilde{B_n} = 1$$

$$\tilde{\delta_n} = \sum_{i=1}^{n} \delta_i + \tan(\frac{\pi}{4}) \left(\tilde{\gamma_n} - \sum_{j=1}^{n} \gamma_j \right) = n\delta + \tan(\frac{\pi}{4}) \left(n^2|\gamma| - n\gamma \right)$$

The distribution is,

$$F_{Z_N}(z) = \sum_{n=1}^{\infty} P_n erfc\left(\sqrt{\frac{\tilde{\gamma_n}}{2(z - \tilde{\delta}_n)}} \right) \cdot I_{[\tilde{\delta_n} < z < \infty]} + \exp(-\lambda)$$

Where

$$F_Z(0) = P(N_t = 0) = exp(-\lambda)$$

for $N = 0$

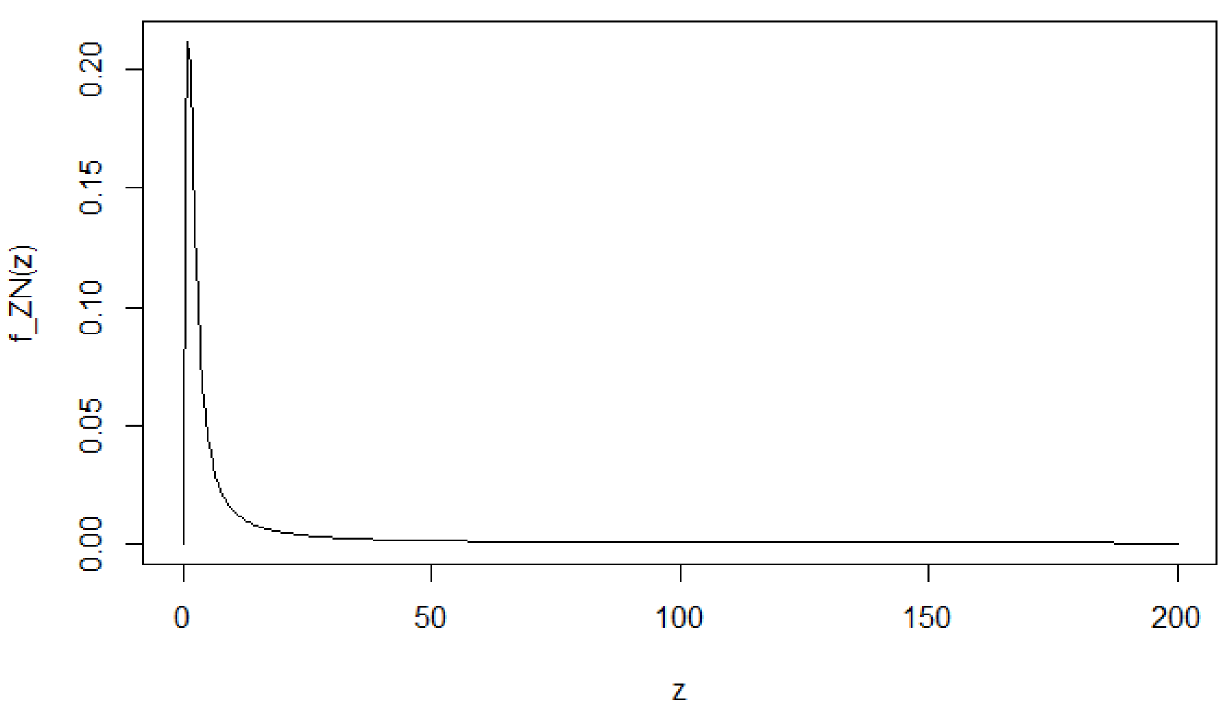
1.2.3 (c)

Hide Code

```
rm(list=ls(all=TRUE))
library(pracma)
library(stats)
library(rmutil)
```

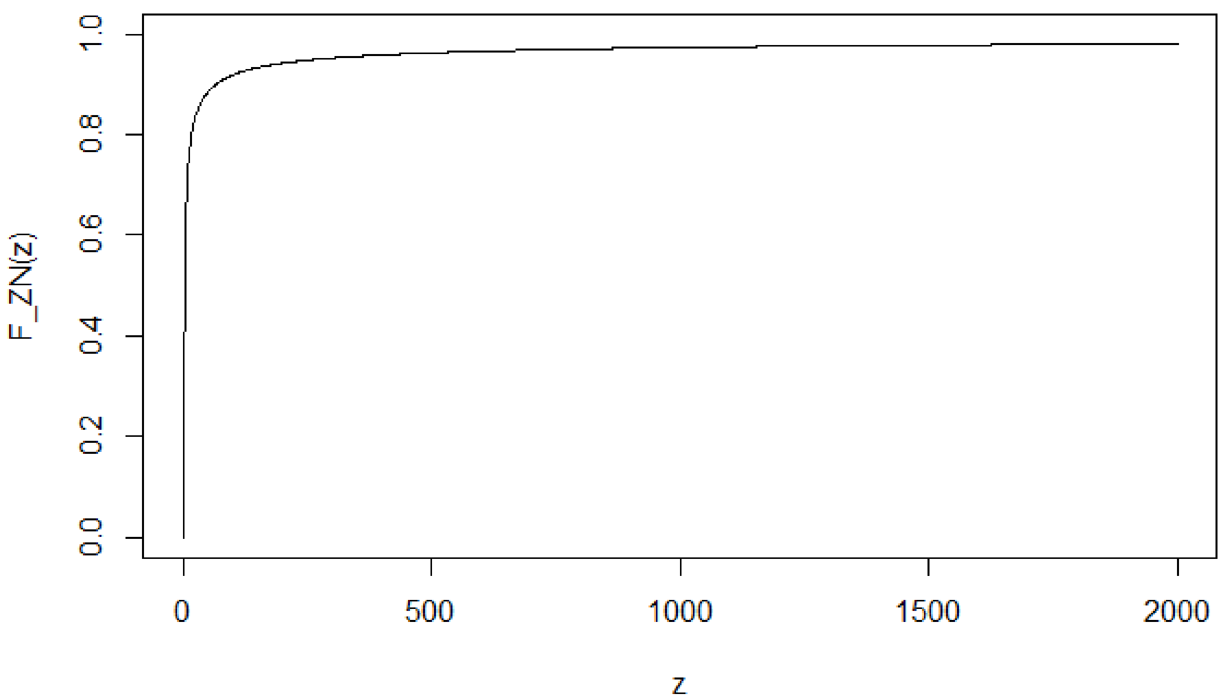
Hide

```
lambda <- 10 #rate of Poisson frequency model
gamm <- 0.01 #scale of Levy severity model
delt <- 0 #Location parameter of Levy severity model

delt_tilde = function(n,delt,gamm) {n*delt + (n^2 *abs(gamm) - n*gamm)} #Under convolution ne
w delta

#Density of Poisson-Levy LDA Model-----------
dens <- function(z,n,lambda,delta,gamma) {partialsum = 0
for(k in 1:n){
  if(delt_tilde(k,delta,gamma)<z){
    partialsum = partialsum + dpois(k,lambda)*dlevy(z,delt_tilde(k,delta,gamma),s=(k^2)*gamm
a)}
}
return(partialsum)
}

#Plotting density of Poisson-Levy LDA Model
zs1 = seq(0, 200, by = 0.5)

dens_vals_list <- c(1:length(zs1))

for(i in 1:length(zs1)){
  dens_vals_list[i] <- dens(zs1[i],n=20,lambda = 10, delta = 0, gamma= 0.01)}

plot(zs1,dens_vals_list,'l', xlab = 'z', ylab= 'f_ZN(z)')
```



Hide

```
#Distribution of Poisson-Levy LDA Model-----------
cdf <- function(z,n,lambda,delta,gamma) {partialsum = exp(-lambda)
for(k in 1:n){
  if(delt_tilde(k,delta,gamma)<z){
    partialsum = partialsum + dpois(k,lambda)*plevy(z,delt_tilde(k,delta,gamma),s=(k^2)*gamm
a)}
}
return(partialsum)
}

#Plotting distribution of Poisson-Levy LDA Model
zs2 = seq(0, 2000, by = 1)

dens_vals_list <- c(1:length(zs1))
cdf_vals_list <- c(1:length(zs2))

for(i in 1:length(zs2)){
  cdf_vals_list[i] <- cdf(zs2[i],n=20,lambda = 10, delta = 0, gamma= 0.01)}

plot(zs2,cdf_vals_list,'l',ylim=c(0,1), xlab = 'z', ylab= 'F_ZN(z)')
```

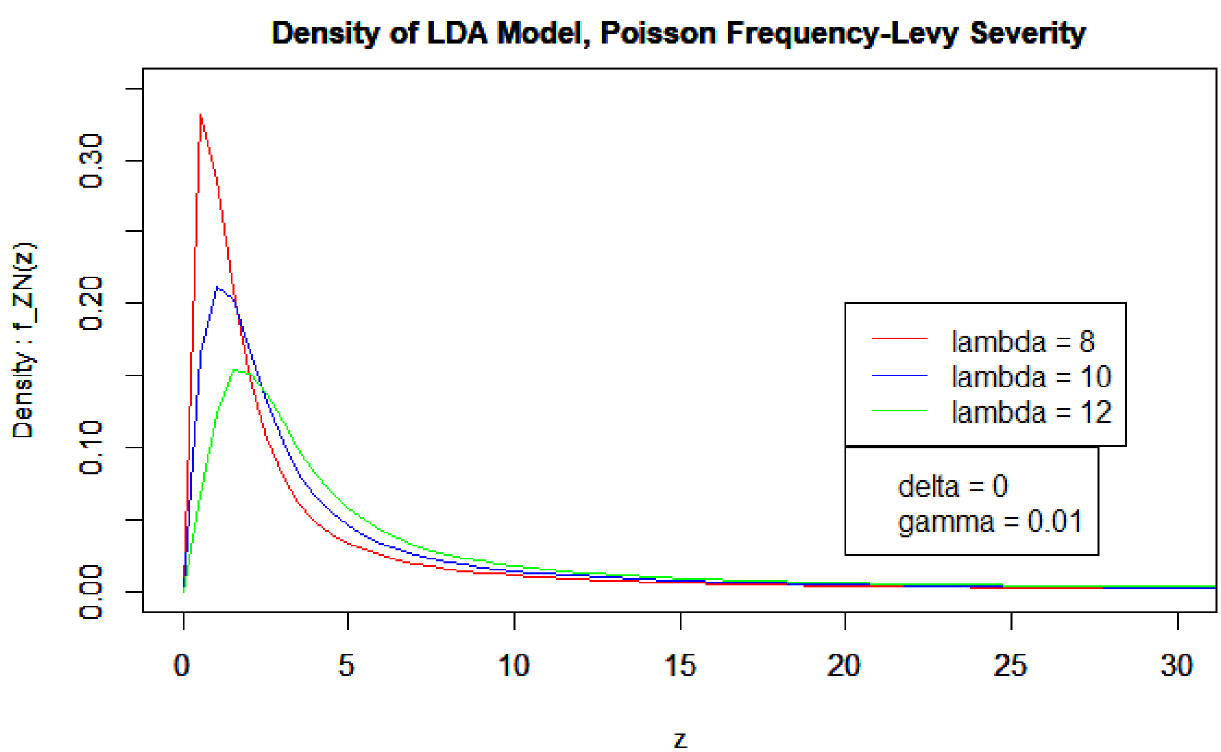


Hide

```
#Testing different values of lambda, alpha and beta

#Lambda tests_____________________________________________________

zs1 = seq(0, 100, by = 0.5)
zs2 = seq(0, 200, by = 1)
dens_vals_list <- c(1:length(zs1))
cdf_vals_list <- c(1:length(zs2))

delta_lambdatest <- 0 #Setting our delta and gamma parameters
gamma_lambdatest <- 0.01

lambdas = c(8,10,12)#We are going to plot the model for lambda = 8, 10 and 12

color = c("red","blue","green")#Defining different colours to see the difference in plots

for(i in 1:3){ #Testing each parameter value for the density function

  for(j in 1:length(zs1)){
    dens_vals_list[j] <- dens(zs1[j],n=20,lambda = lambdas[i], delta = delta_lambdatest, gamm
a= gamma_lambdatest)}

  plot(zs1,dens_vals_list,main = "Density of LDA Model, Poisson Frequency-Levy Severity", ce
x.main=1.05 ,xlab = "z",ylab="Density : f_ZN(z)",xlim = c(0,30),ylim = c(0,0.35),cex.lab=0.9,
"l",col = color[i])

  par(new = TRUE)
}
legend(20,0.20,legend = c("lambda = 8","lambda = 10","lambda = 12"),col = c("red","blue","gre
en"),lty = 1)
```

Hide

```
legend(20,0.1, legend = c("delta = 0","gamma = 0.01"))
```



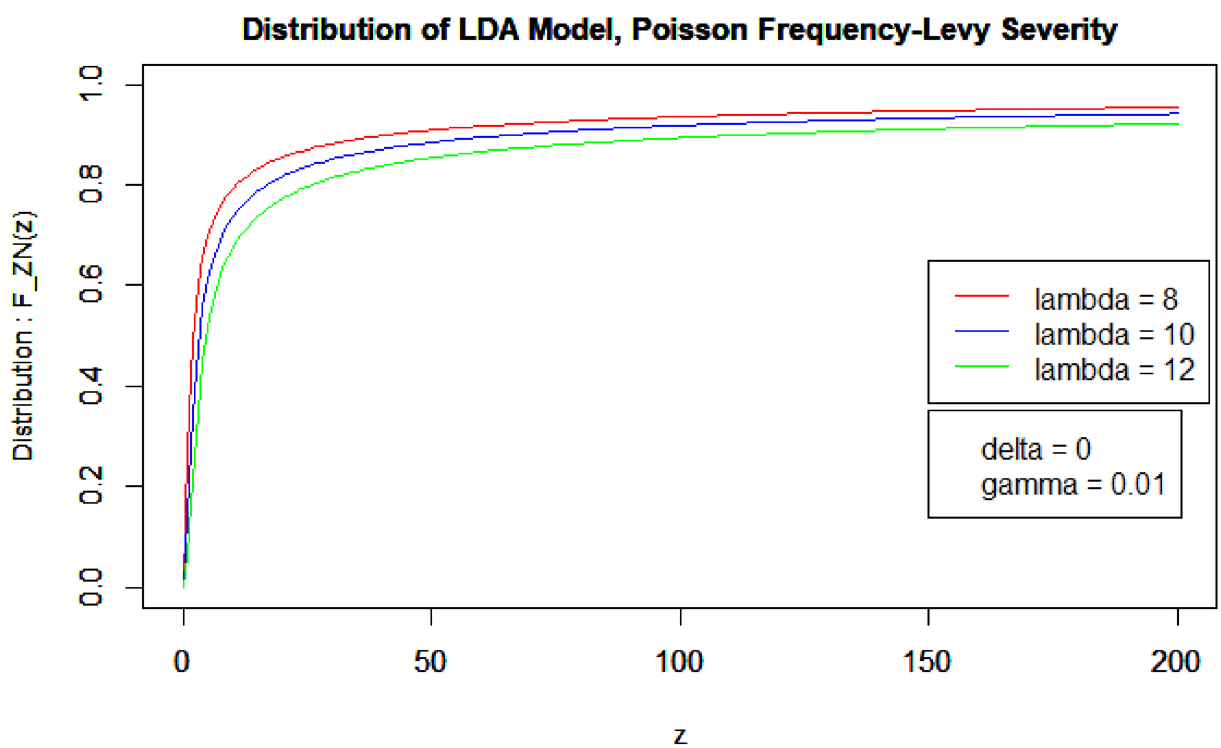
We can see as we increase lambda the peak height decreases.

Hide

```
for(i in 1:3){ #Testing each parameter value for the distribution function

  for(j in 1:length(zs2)){
    cdf_vals_list[j] <- cdf(zs2[j],n=20,lambda = lambdas[i], delta = delta_lambdatest, gamma=
gamma_lambdatest)}

  plot(zs2,cdf_vals_list,main = "Distribution of LDA Model, Poisson Frequency-Levy Severity",
cex.main=1.05 ,xlab = "z",ylab="Distribution : F_ZN(z)",xlim = c(0,200),ylim = c(0,1),cex.lab
=0.9,"l",col = color[i])

  par(new = TRUE)
}
legend(150,0.65,legend = c("lambda = 8","lambda = 10","lambda = 12"),col = c("red","blue","gr
een"),lty = 1)
```

Hide

```
legend(150,0.35, legend = c("delta = 0","gamma = 0.01"))
```

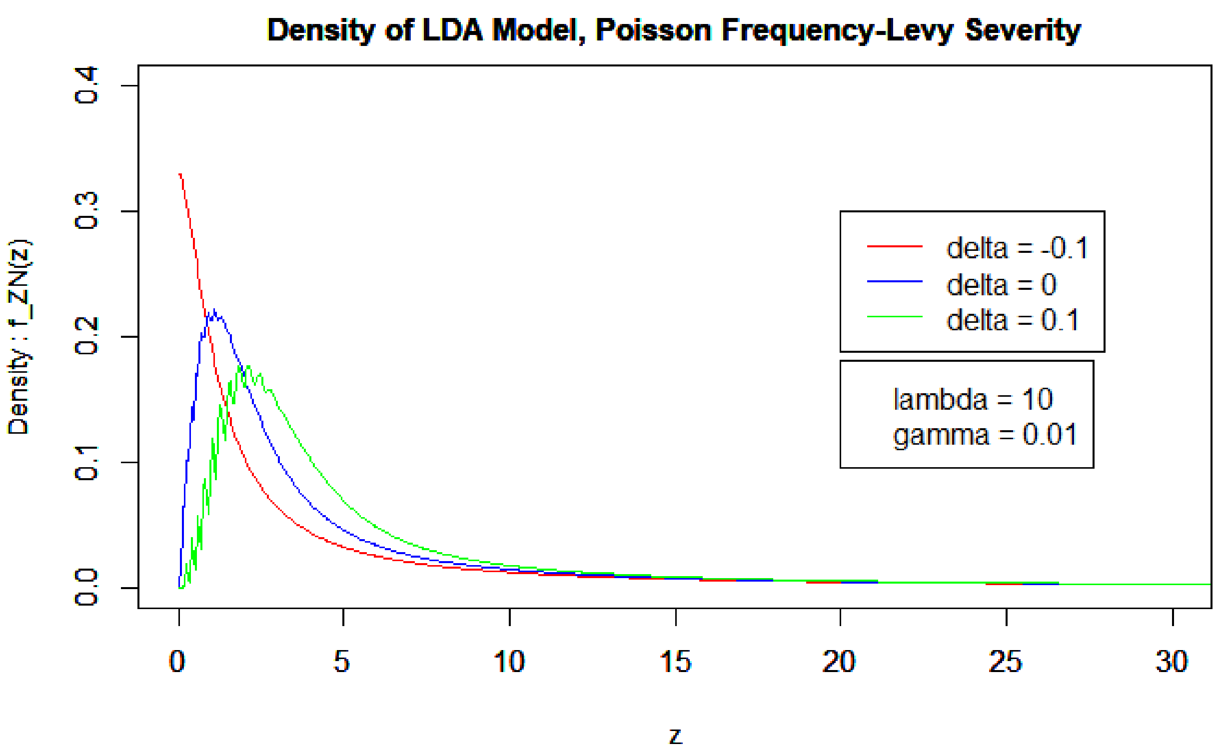


Hide

```
#Delta tests_____________________________________________________
zs1 = seq(0, 100, by = 0.05)
zs2 = seq(0, 200, by = 0.05)

lambda_deltatest <- 10 #Setting our gamma and lambda parameters
gamma_deltatest <- 0.01

deltas = c(-0.1,0,0.1)#We are going to plot the model for delta = -0.1, 0 and 0.1
color = c("red","blue","green")

for(i in 1:3){ #Testing each parameter value for the density function

  for(j in 1:length(zs1)){
    dens_vals_list[j] <- dens(zs1[j],n=20,lambda = lambda_deltatest, delta = deltas[i], gamma
= gamma_deltatest)}

  plot(zs1,dens_vals_list,main = "Density of LDA Model, Poisson Frequency-Levy Severity", ce
x.main=1.05 ,xlab = "z",ylab="Density : f_ZN(z)",xlim = c(0,30),ylim = c(0,0.40),cex.lab=0.9,
"l",col = color[i])

  par(new = TRUE)
}
legend(20,0.3,legend = c("delta = -0.1","delta = 0","delta = 0.1"),col = c("red","blue","gree
n"),lty = 1)
```

Hide

```
legend(20,0.18, legend = c("lambda = 10","gamma = 0.01"))
```



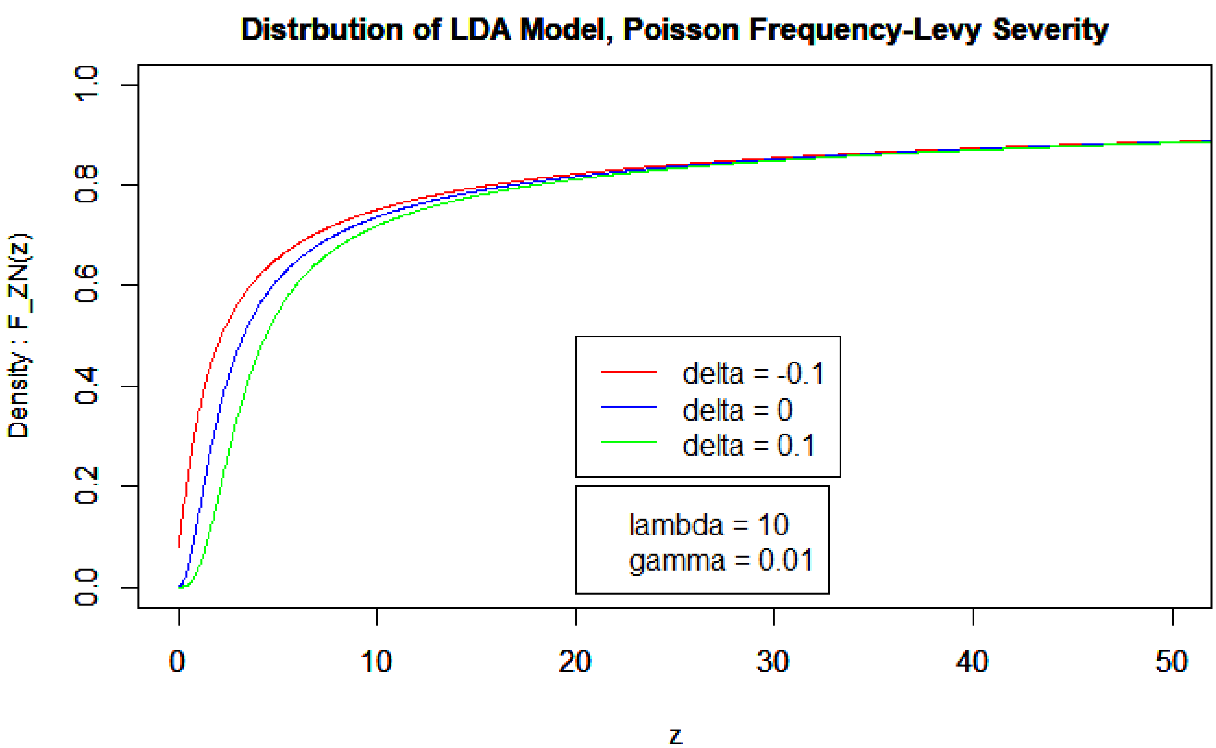
We can see as we increase delta the peak location shifts to the right.

Hide

```
for(i in 1:3){ #Testing each parameter value for the distribution function

  for(j in 1:length(zs2)){
    cdf_vals_list[j] <- cdf(zs2[j],n=20,lambda = lambda_deltatest, delta = deltas[i], gamma=
 gamma_deltatest)}

  plot(zs2,cdf_vals_list,main = "Distrbution of LDA Model, Poisson Frequency-Levy Severity",
 cex.main=1.05 ,xlab = "z",ylab="Density : F_ZN(z)",xlim = c(0,50),ylim = c(0,1),cex.lab=0.9,
"l",col = color[i])

  par(new = TRUE)
}
legend(20,0.50,legend = c("delta = -0.1","delta = 0","delta = 0.1"),col = c("red","blue","gre
en"),lty = 1)
```

Hide

```
legend(20,0.20, legend = c("lambda = 10","gamma = 0.01"))
```

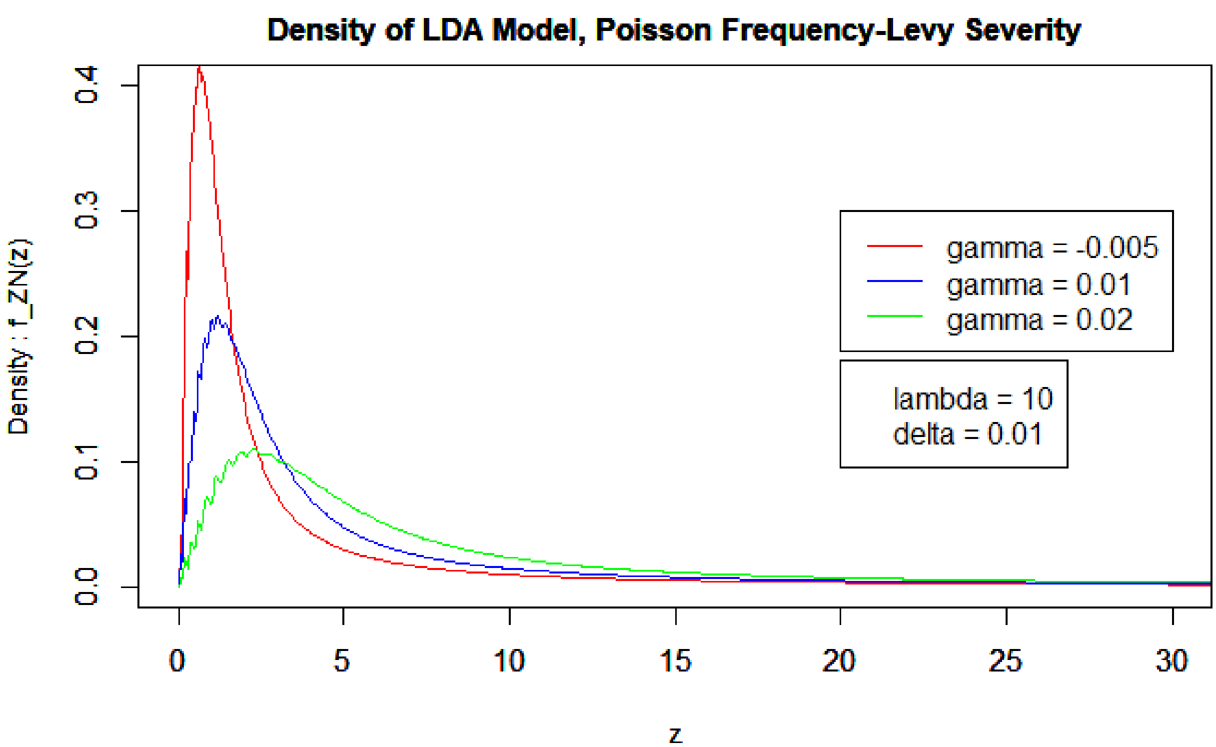


Hide

```
#Gamma tests_____________________________________________________
zs1 = seq(0, 100, by = 0.05)
zs2 = seq(0, 200, by = 0.05)

lambda_gammatest <- 10 #Setting our delta and lambda parameters
delta_gammatest <- 0.01
gammas = c(0.005,0.01,0.02)#We are going to plot the model for gamma = 0.005,0.01 and 0.02
color = c("red","blue","green")

for(i in 1:3){ #Testing each parameter value for the density function

  for(j in 1:length(zs1)){
    dens_vals_list[j] <- dens(zs1[j],n=20,lambda = lambda_gammatest, delta = delta_gammatest,
gamma = gammas[i])}

  plot(zs1,dens_vals_list,main = "Density of LDA Model, Poisson Frequency-Levy Severity", ce
x.main=1.05 ,xlab = "z",ylab="Density : f_ZN(z)",xlim = c(0,30),ylim = c(0,0.4),cex.lab=0.9,
"l",col = color[i])

  par(new = TRUE)
}
legend(20,0.30,legend = c("gamma = -0.005","gamma = 0.01","gamma = 0.02"),col = c("red","blu
e","green"),lty = 1)
```

Hide

```
legend(20,0.18, legend = c("lambda = 10","delta = 0.01"))
```



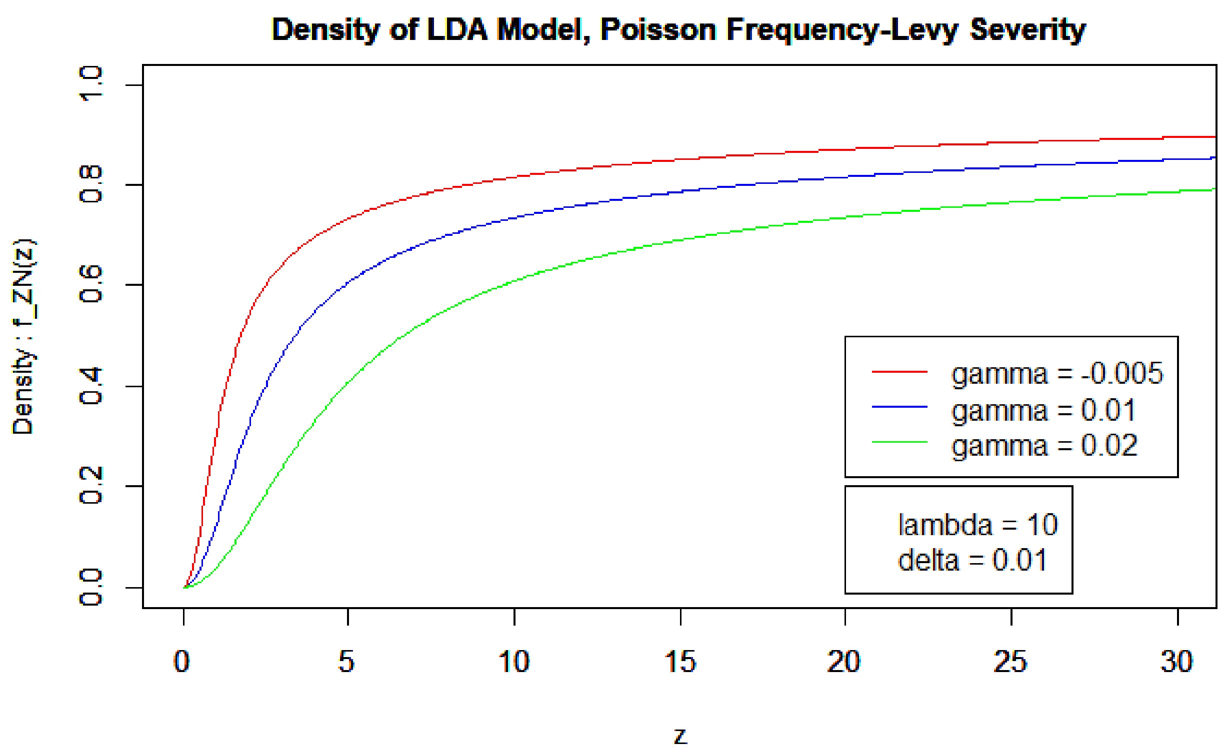
We can see as we increase gamma the distributions tail gets heavier.

Hide

```
for(i in 1:3){ #Testing each parameter value for the distribution function

  for(j in 1:length(zs2)){
    cdf_vals_list[j] <- cdf(zs2[j],n=20,lambda = lambda_gammatest, delta = delta_gammatest, g
amma = gammas[i])}

  plot(zs2,cdf_vals_list,main = "Density of LDA Model, Poisson Frequency-Levy Severity", cex.
main=1.05 ,xlab = "z",ylab="Density : f_ZN(z)",xlim = c(0,30),ylim = c(0,1),cex.lab=0.9,"l",c
ol = color[i])

  par(new = TRUE)
}
legend(20,0.50,legend = c("gamma = -0.005","gamma = 0.01","gamma = 0.02"),col = c("red","blu
e","green"),lty = 1)
```

Hide

```
legend(20,0.2, legend = c("lambda = 10","delta = 0.01"))
```



1.2.4 (d)

To calculate the compound loss distribution via the Monte Carlo method, follow the steps:

(i) Simulate the annual number of events N from the Poisson frequency distribution;

(ii) Simulate independent severities $X_1, \ldots, X_N$ from the Levy severity distribution, where N is from step (i);

(iii) Aggregate the losses to obtain a realization of Z_N given by $Z_n = X_1 + ... + X_n$. Repeat these steps K times to get $Z_1, \ldots, Z_K$ independent samples of Z from the Poisson-Levy compound distribution

1.2.5 (e)

Q1B Part e)

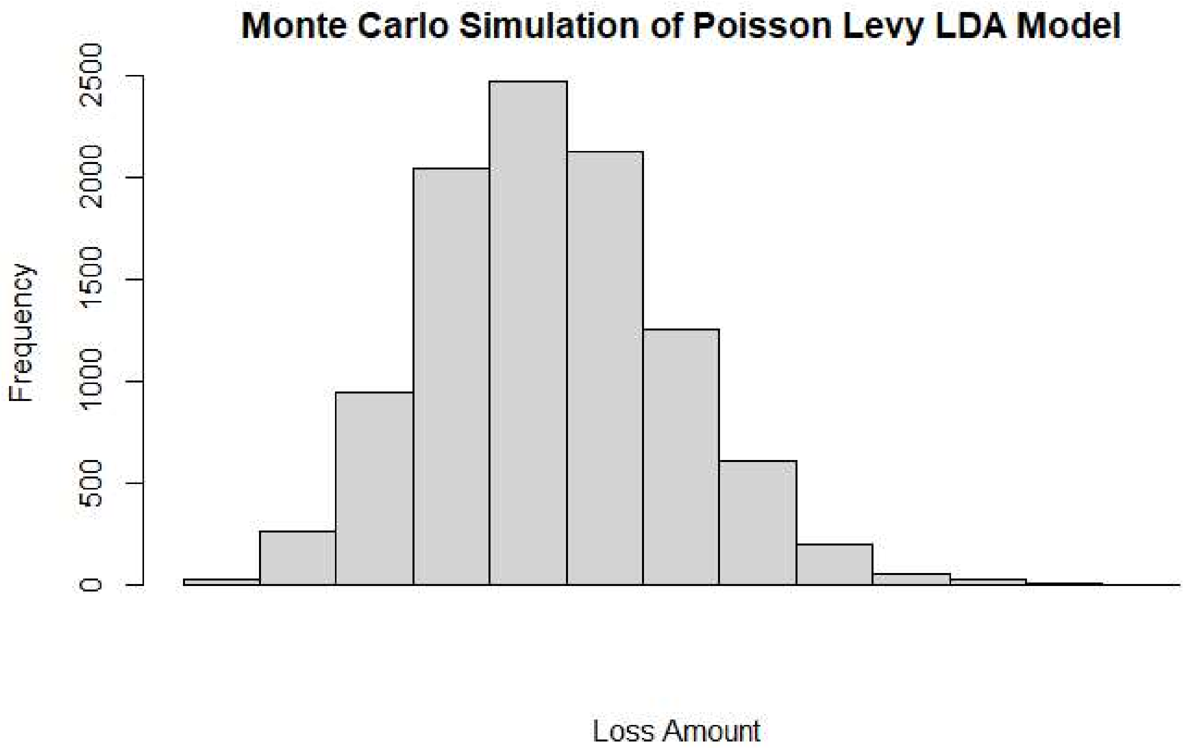
Code

Hide

```
library(pracma)
library(stats)
library(rmutil)
```

Hide

```
#Monte Carlo Simulation_______________________________________________________________
lambda <- 10 #rate of Poisson frequency model
gamm <- 0.01 #scale of Levy severity model
delt <- 0 #Location parameter of Levy severity model

poisson_levy_montecarlo <- function(lambda,gamma,delta,nsim){

  annual_loss <- rep(NA,nsim)

  for (i in 1:nsim){ #We will run nsim = 10,000 simulations

    no_of_losses <- rpois(1,lambda) #N drawn from poisson distribution

    if (no_of_losses == 0){

      annual_loss[i] <- 0

    } else{

      individual_losses = rlevy(no_of_losses,gamma, delta)#severity X drawn from levy distrib
ution

      agg_losses = sum(individual_losses)
      annual_loss[i] <- agg_losses

    }

  }


  return(annual_loss)

}
levy = poisson_levy_montecarlo(lambda = lambda,gamma = gamm, delta = delt, nsim = 10000)
#Histogram for simulations is plot
hist(levy, main = "Monte Carlo Simulation of Poisson Levy LDA Model" ,xaxt='n' , xlab = "Loss
Amount ")
```



Hide

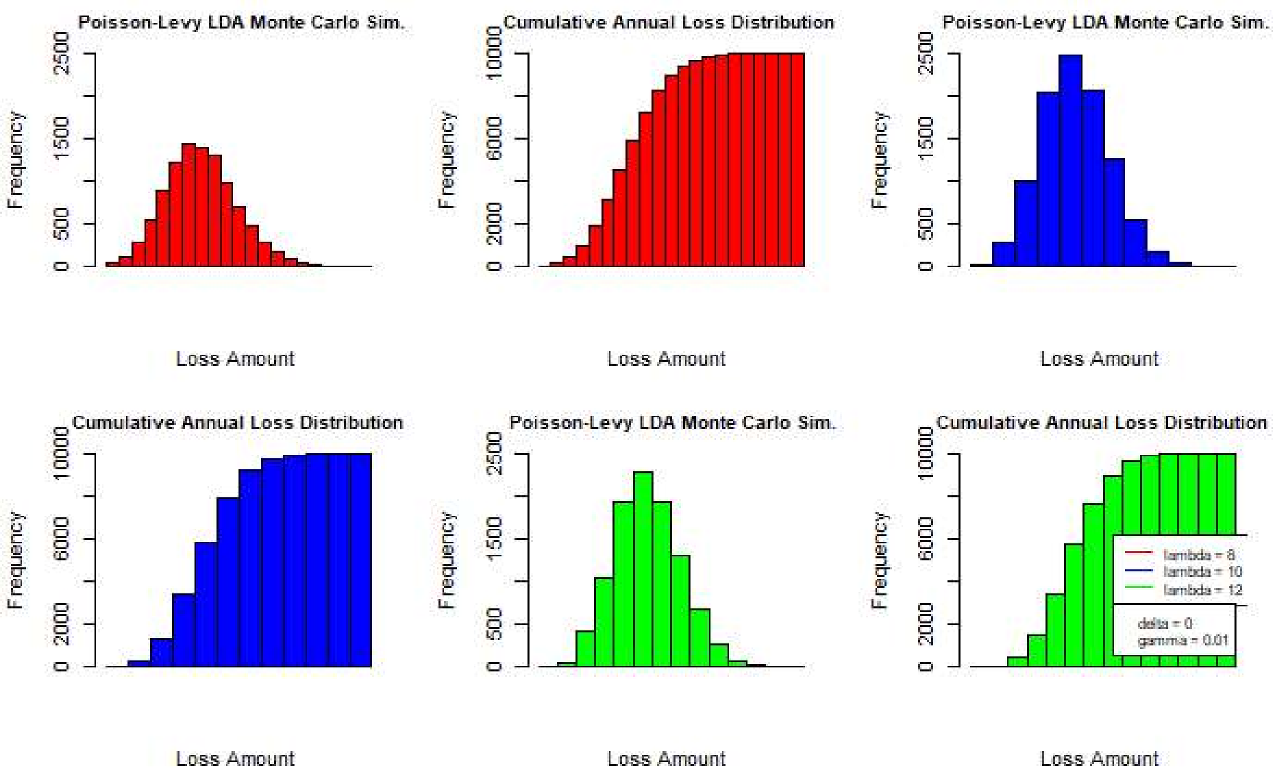
```
#Monte-carlo Simulations for parameter testing-----------------------
zs1 = seq(0, 100, by = 0.05)
zs2 = seq(0, 200, by = 0.05)

#Setting parameters for gamma, delta and lambda for each test
delta_lambdatest <- 0
gamma_lambdatest <- 0.01

lambda_gammatest <- 10
delta_gammatest <- 0

lambdas = c(8,10,12)
deltas = c(0,0.05,0.1)
gammas = c(0.005,0.01,0.02)

color = c("red","blue","green")
```

Hide

```
#Lambda tests______________________________________________________________
par(mfrow=c(2,3))

for(i in 1:3){#Testing the Monte-Carlo distribution for values of lambda
  monty_test = poisson_levy_montecarlo(lambda = lambdas[i],gamma = gamma_lambdatest, delta =
 delta_lambdatest, nsim = 10000)
  par(new=FALSE)
  cuml_hist <- hist(monty_test,col = color[i],xlab = "Loss Amount ",main = " Poisson-Levy LDA
Monte Carlo Sim.",xaxt='n',cex.main = 0.94,ylim = c(0,2500))

  cuml_hist$counts <- cumsum(cuml_hist$counts)
  plot(cuml_hist,col=color[i],xlab = "Loss Amount ",main = "Cumulative Annual Loss Distributi
on",xaxt='n',cex.main = 0.94)
}
legend(0.15,6200,legend = c("lambda = 8","lambda = 10","lambda = 12"),col = c("red","blue","g
reen"),lty = 1,cex = 0.75)
```

Hide

```
legend(0.15,3000,legend = c("delta = 0","gamma = 0.01"),cex = 0.75)
```

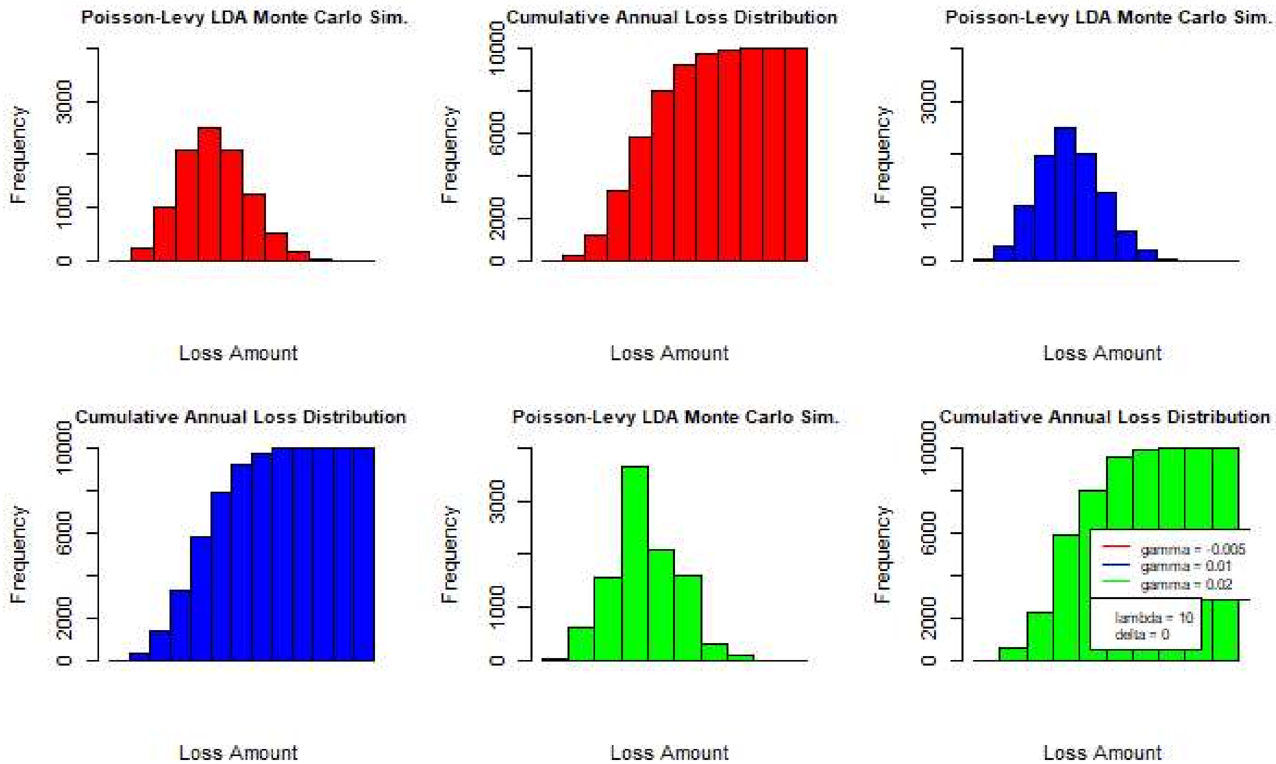


Hide

```
#Gamma tests_____________________________________________________
par(mfrow=c(2,3))
for(i in 1:3){#Testing the Monte carlo distribution for values of gamma
  monty_test = poisson_levy_montecarlo(lambda = lambda_gammatest,gamma = gammas[i], delta = delta_gammatest, nsim = 10000)
  par(new=FALSE)
  cuml_hist <- hist(monty_test,col = color[i],xaxt='n',xlab = "Loss Amount ",main = " Poisson-Levy LDA Monte Carlo Sim.",cex.main = 0.94,ylim = c(0,4000))


  cuml_hist$counts <- cumsum(cuml_hist$counts)
  plot(cuml_hist,col=color[i],xlab = "Loss Amount ", main = "Cumulative Annual Loss Distribution",xaxt='n',cex.main = 0.94)
}
legend(0.22,6200,legend = c("gamma = -0.005","gamma = 0.01","gamma = 0.02"),col = c("red","blue","green"),lty = 1,cex = 0.75)
```

Hide

```
legend(0.22,3000, legend = c("lambda = 10","delta = 0"),cex = 0.75)
```



We can see the densities and cumulative distribution follow a similar shape to the histograms.

2 Heavy tail loss models and quantiles

2.1 Question A

2.1.1 (a)

The hazard function is defined to be the ratio of the density of the survival functions:

$$h(x) = \frac{f(x)}{S(x)} = \frac{f(x)}{1 - F(x)}$$

It can also be shown that:

$$h(x) = -\frac{S'(x)}{S(x)} = -\frac{d(x)}{dx}$$

Where $S(x) = \bar{F}(x) = P(X > x)$

Hazard rate functions reveal information about the tail of the distribution:

- Distributions with decreasing hazard rate functions have heavy tails.
- Distributions with increasing hazard rate functions have lighter tails.

Comparison between distributions can be made on the basis of the rate of increase or decrease of the hazard functions. Therefore a heavy tailed severity model will have a decreasing hazard rate functions.

We choose the Pareto Type 2 (Lomax) distribution as our two parameter severity model. So we have $X \sim P2(\alpha, \theta)$ which has pdf,

$$f(x) = \frac{\alpha\theta^\alpha}{(x+\theta)^{\alpha+1}}$$

and,

$$\bar{F}(x) = 1 - F(X) = 1 - (1 - \frac{\theta^\alpha}{(x+\theta)^\alpha}) = \frac{\theta^\alpha}{(x+\theta)^\alpha}$$

Hence so we use the hazard function

$$h(x) = \frac{f(x)}{S(x)} = \frac{\frac{\alpha\theta^\alpha}{(x+\theta)^{\alpha+1}}}{\frac{\theta^\alpha}{(x+\theta)^\alpha}} = \frac{\alpha}{x+\theta}$$

and $\lim_{x\to\infty} h(x) \to 0$ so therefore the hazard function is decreasing. So therefore

2.1.2 (b)

$$Z = \sum_{n=0}^{N} X_n$$

$$N \sim G(n), X_n \sim F(x)$$

Under collective risk model X_i are i.i.d and independent of N. So we need to prove that:

$$\mathbb{E}[Z] = \mathbb{E}[N]\mathbb{E}[X](1)$$

$$Var(Z) = \mathbb{E}[N]Var(X) + Var(N)\mathbb{E}[X^2].(2)$$

Proof of (1) :

$$\mathbb{E}[Z] = \mathbb{E}[\sum_{n=0}^{N} X_n]$$

$$= \sum_{k=0}^{\infty} P(N = k)\mathbb{E}(\sum_{n=0}^{N} X_n | N = k)$$

$$= \sum_{k=0}^{\infty} P(N = k)K\mathbb{E}[X]$$

But since N is independent of X

$$\mathbb{E}[Z] = \mathbb{E}[N]\mathbb{E}[X]$$

Proof of (2) :
So we first prove that

$$E(Z|N) = N\mathbb{E}[X]$$

so for fixed value of k of N:

$$\mathbb{E}(\sum_{n=0}^{N} | N = k) = k\mathbb{E}[X_n]$$

since it's true for all n we have proved:

$$Var(Z|N) = NVar(X)$$

for fixed value of k of N:

$$Var(\sum_{n=0}^{N} X_n | N = k) = Var(\sum_{n=0}^{k} X_n) = kVar(X)$$

Hence,

$$Var(Z) = \mathbb{E}_N[Var(Z|N)] + Var_N(\mathbb{E}[Z|N])$$

$$= \mathbb{E}_N[NVar(X)] + Var_N(N\mathbb{E}[X])$$

$$= \mathbb{E}[N]Var(X) + Var(N)\mathbb{E}[X]^2(3)$$

But we know that

$$Var(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2$$

$$\implies \mathbb{E}[X]^2 = \mathbb{E}[X^2] - Var(X)$$

Substituting this into (3)

$$Var(Z) = \mathbb{E}[N](\mathbb{E}[X^2] - \mathbb{E}[X]^2) + Var(N)(\mathbb{E}[X^2] - Var(X))$$

Hence we can write the variance as either:

$$Var(Z) = \mathbb{E}[X^2]\mathbb{E}[N] + \mathbb{E}[X^2](Var(N) - \mathbb{E}[N])$$

or

$$Var(Z) = \mathbb{E}[X^2]Var(N) + Var(X)(\mathbb{E}[N] - Var(N))$$

Therefore we can see that the follwing bound holds:

$$Var(Z) \leq \mathbb{E}[X^2]\max(\mathbb{E}[N], Var(N))$$

2.1.3 (c)

Demonstrate that for the Gamma distribution all positive moments exist but for the Pareto distribution they do not:

Let $X \sim Gamma(\alpha, \beta)$

$$\mathbb{E}[X^k] = \int_0^\infty x^k \frac{x^{\alpha-1}e^{-x/\beta}}{\Gamma(\alpha)\beta^\alpha}dx$$

Letting y $= \frac{x}{\beta}$

$$= \int_0^\infty (y\beta)^k \frac{(y\beta)^{\alpha-1}e^{-y}}{\Gamma(\alpha)\beta^\alpha}dy\beta$$

$$= \beta^k \int_0^\infty \frac{y^{k+\alpha-1}e^{-y}}{\Gamma(\alpha)}dy$$

$$= \frac{\beta^k\Gamma(k+\alpha)}{\Gamma(\alpha)}\int_0^\infty \frac{y^{k+\alpha-1}e^{-y}}{\Gamma(k+\alpha)}dy$$

$$= \frac{\Gamma(k+\alpha)\beta^k}{\Gamma(\alpha)} < \infty$$

$$\forall k > 0$$

Hence all positive moments exist for Gamma distribution.

Now we will prove that not all positive moments for the Pareto distribution exist.

Let $Y \sim Pareto(\alpha, \beta)$

$$\mathbb{E}[Y^k] = \int_0^\infty y^k \frac{\alpha\beta^\alpha}{(y+\beta)^{\alpha+1}}dy$$

Letting x $= y + \beta$

$$= \int_{\beta}^{\infty} (x-\beta)^k \frac{\alpha\beta^\alpha}{x^{\alpha+1}} dx$$

$$= \alpha\beta^\alpha \int_{\beta}^{\infty} \sum_{j=0}^{k} \binom{k}{j} x^j (-\beta)^{k-j} x^{-\alpha-1} dx$$

$$= \alpha\beta^\alpha \int_{\beta}^{\infty} \sum_{j=0}^{k} \binom{k}{j} x^{j-\alpha-1} (-\beta)^{k-j} dx$$

We an see this integral exists for $j-\alpha-1<-1$, which is equivalent to $k<\alpha$, therefore not all moments exist.

3 Risk measure calculations and asymptotic approximations

3.1 Question A

3.1.1 (a)

VaR is defined by:

$VaR_\alpha = F^{-1}(\alpha) = \inf\{l \in \mathbb{R} : P(L>l) \leq 1-\alpha\} = inf\{l \in \mathbb{R} : F_L(l) \geq \alpha\} = \sup\{x : F(x) < \alpha\}$

That is, VaR is the minimum threshold exceeded by X with probability at most $1-\alpha$.

3.1.2 (b)

The expected shortfall of a random variable $x \sim F(x)$ at the α -th probability level $ES_\alpha[X]$ is

$$ES_\alpha[X] = \frac{1}{1-\alpha} \int_{\alpha}^{1} VaR_p[X]$$

which is the "arithmetic average" of the VaRs of X from α to 1
In the case of continuous distributions, it can be shown that $ES[X]$ is just expected loss given that the loss exceeds $VaR[X]$

$$ES_\alpha[X] = \mathbb{E}[X|X \geq VaR_\alpha[X]] = \mathbb{E}[X|X > VaR_\alpha[X]]$$

which is the conditional expected loss given that the loss exceeds $VaR_\alpha[X]$.

3.2 Question B

3.2.1 (a)

We know the first order approximation is :

$$\bar{F}_z(x) = \mathbb{E}[N]\bar{F}_x(x)(1 + o(1))$$

as $x \to \infty$
For frequency $N \sim Poi(\lambda)$ we have:

$$\mathbb{E}[N] = \lambda$$

For severity $X \sim LogNormal(\mu, \sigma)$, we have the tail function:

$$\bar{F}(x; \mu, \sigma) = 1 - F(x) = 1 - \frac{1}{2}\left[1 + erf\left(\frac{\ln(x) - \mu}{\sigma\sqrt{2}}\right)\right]$$

$$\bar{F}(x; \mu, \sigma) = \frac{1}{2} - \frac{1}{2}erf\left(\frac{\ln(x) - \mu}{\sigma\sqrt{2}}\right)$$

Hence the first order asymptotic approximation for the Poisson-Lognormal model tail function is:

$$\bar{F}_z(x) = \lambda\left(\frac{1}{2} - \frac{1}{2}erf\left(\frac{\ln(x) - \mu}{\sigma\sqrt{2}}\right)\right)$$

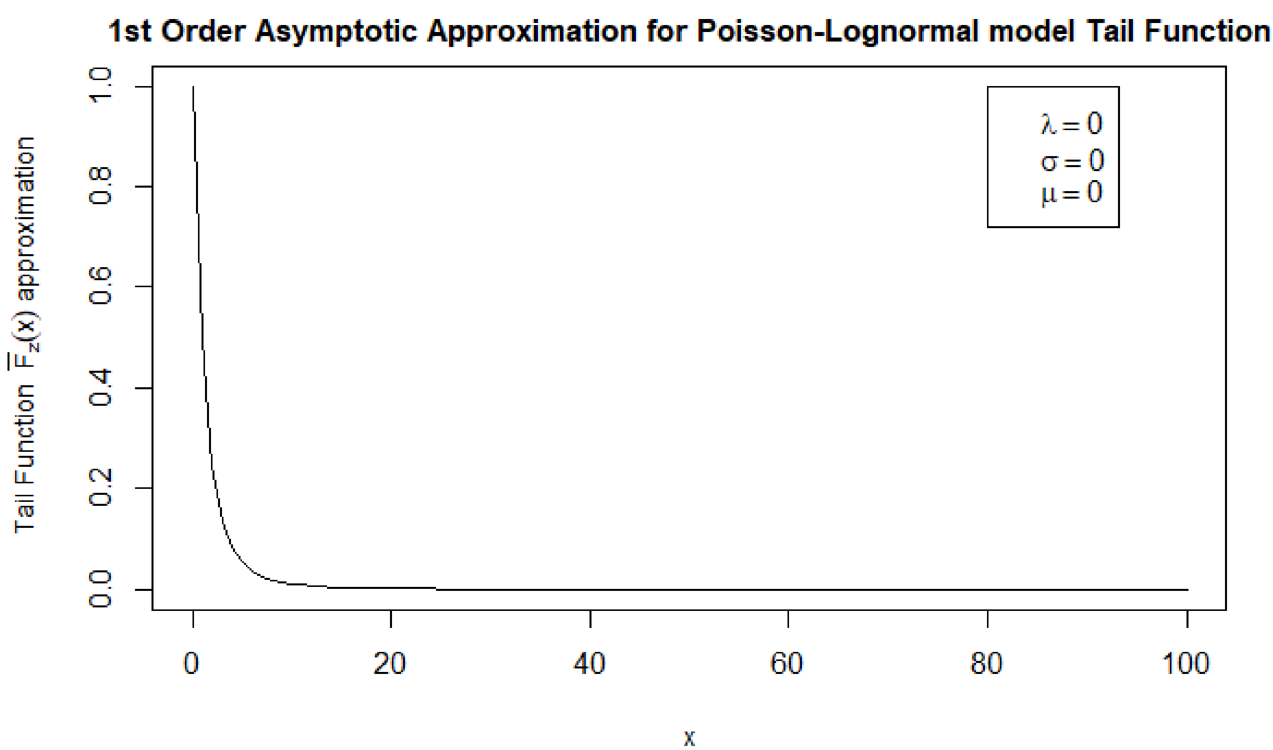
3.2.2 (b)

Hide Code

```
library(pracma)
library(latex2exp)
```

Hide

```
#Defining variables used in First order asymptotic approximation for the Poisson-Lognormal mo
del tail function
mu <- 0
sigma <- 1
lambda <- 1


PL_LDA = function(x) {
  lambda * (1/2 - 1/2*erf((log(x)-mu)/(sigma*sqrt(2))))#This is the equation found from part
 a)
}

x_s = 0:100
plot(x_s,PL_LDA(x_s),main = "1st Order Asymptotic Approximation for Poisson-Lognormal model T
ail Function", cex.main=1.05 ,xlab = "x",ylab=TeX(r'(Tail Function  $\bar{F}_z(x)$ approximat
ion)'),cex.lab=0.9,"l",ylim = c(0,1)) #We plot the tail function from x values going from 1-1
00.

legend(80,1,legend = c(TeX(r'($\lambda = 0)'),TeX(r'($\sigma = 0)'),TeX(r'($\mu = 0)')))
```

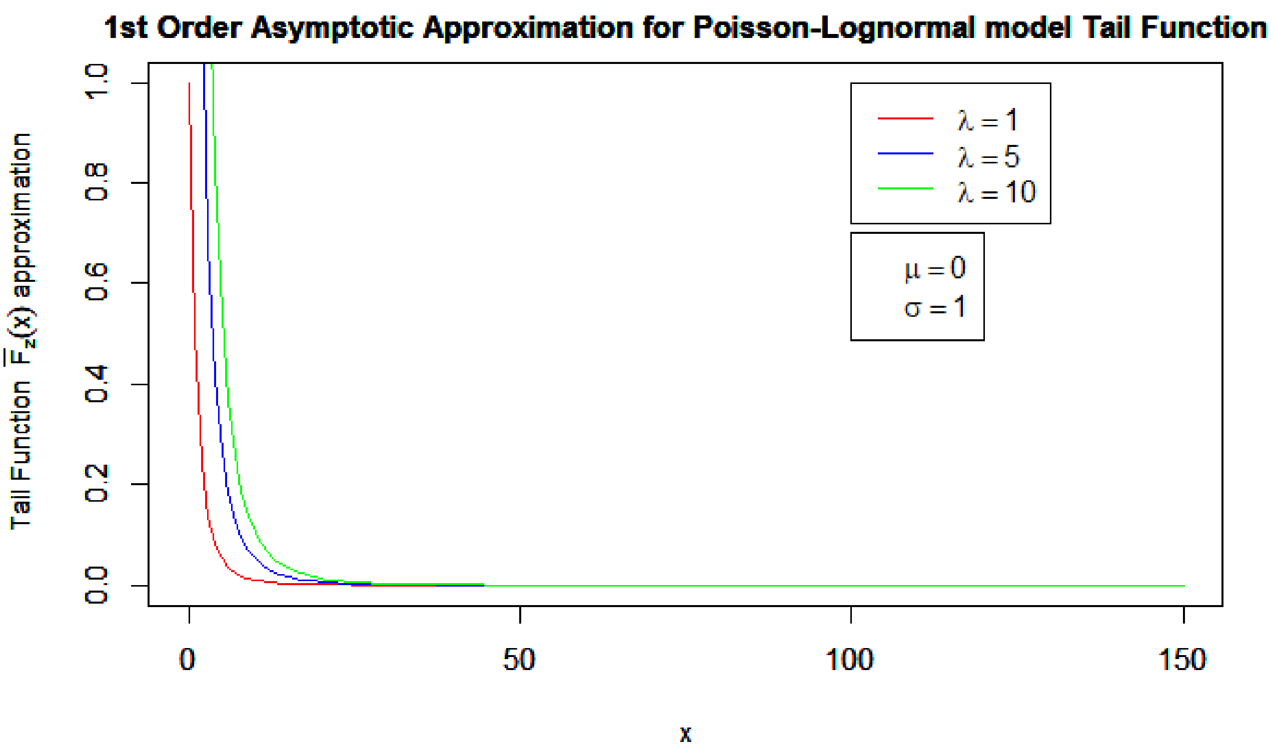


Hide

```
#Lambda tests_____________________________________________________
x_s = 0:150
sigma_lambdatest <- 1 #We are going to plot the model for lambda = 1, 5 and 10
mu_lambdatest <- 0
lambdas = c(1,5,10)
color = c("red","blue","green")

for(i in 1:3){
  PL = function(x) {
    lambdas[i] * (1/2 - 1/2*erf((log(x)-mu_lambdatest)/(sigma_lambdatest*sqrt(2))))
  }

  plot(x_s,PL(x_s),main = "1st Order Asymptotic Approximation for Poisson-Lognormal model Tai
l Function", cex.main=1.05 ,xlab = "x",ylab=TeX(r'(Tail Function  $\bar{F}_z(x)$ approximatio
n)'),cex.lab=0.9,"l",ylim = c(0,1),col = color[i])
  par(new = TRUE)
}
legend(100,1,legend = c(TeX(r'($\lambda = 1)'),TeX(r'($\lambda = 5)'),TeX(r'($\lambda = 10)'
)),col = c("red","blue","green"),lty = 1)
```

Hide

```
legend(100,0.7, legend = c(TeX(r'($\mu = 0)'),TeX(r'($\sigma = 1)')))
```



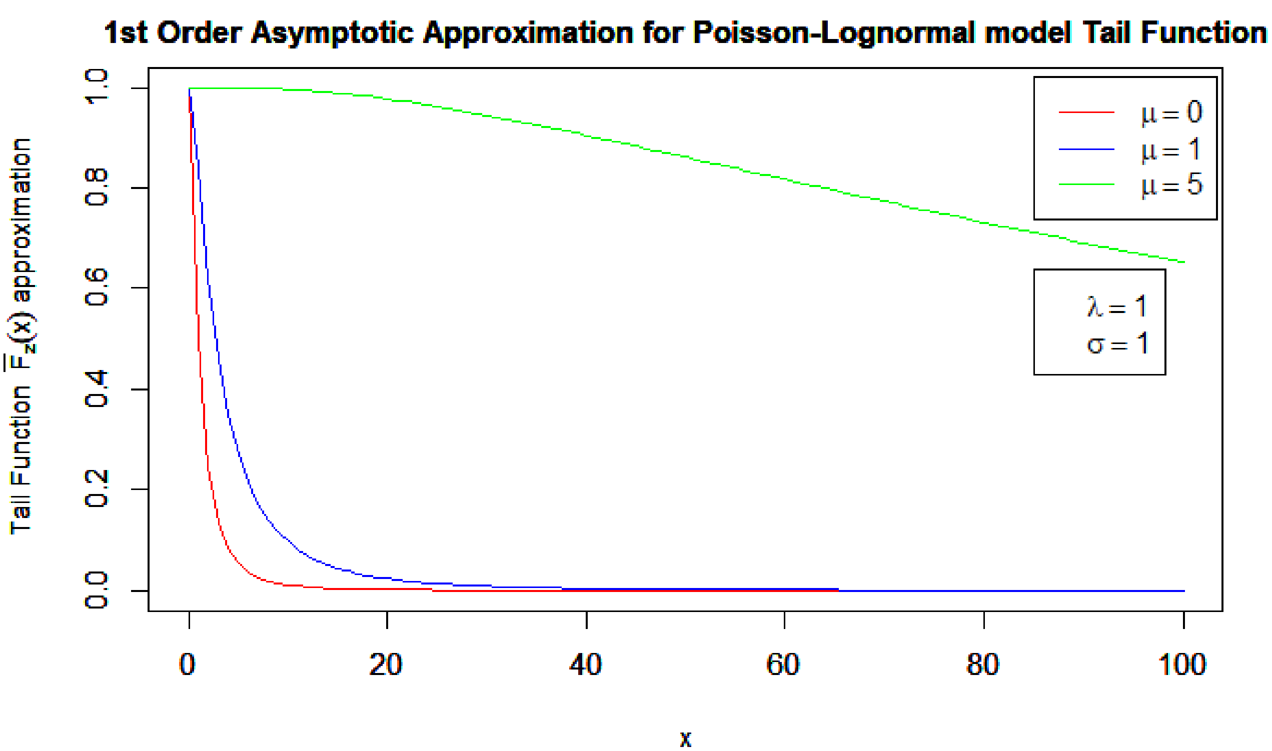
We can see for different values of λ that the tail does not vary much, which suggests that the tail behaviour is more dependant on the severity.

Hide

```
#Mu tests___________________________________________________________
x_s = 0:100
lambda_mutest <- 1
sigma_mutest <- 1
mus = c(0,1,5) #We are going to plot the model for mu = 0, 1, and 5
color = c("red","blue","green")

for(i in 1:3){
  PL = function(x) {
    lambda_mutest * (1/2 - 1/2*erf((log(x)-mus[i])/(sigma_mutest*sqrt(2))))
  }
  plot(x_s,PL(x_s),"l",main = "1st Order Asymptotic Approximation for Poisson-Lognormal model
Tail Function", cex.main=1.05 ,xlab = "x",ylab=TeX(r'(Tail Function  $\bar{F}_z(x)$ approxima
tion)'),cex.lab=0.9,ylim = c(0,1),col = color[i])
  par(new = TRUE)
}
legend(85,1.02,legend = c(TeX(r'($\mu = 0)'),TeX(r'($\mu = 1)'),TeX(r'($\mu = 5)')),col = c(
"red","blue","green"),lty = 1)
```

Hide

```
legend(85,0.64, legend = c(TeX(r'($\lambda = 1)'),TeX(r'($\sigma = 1)')))
```



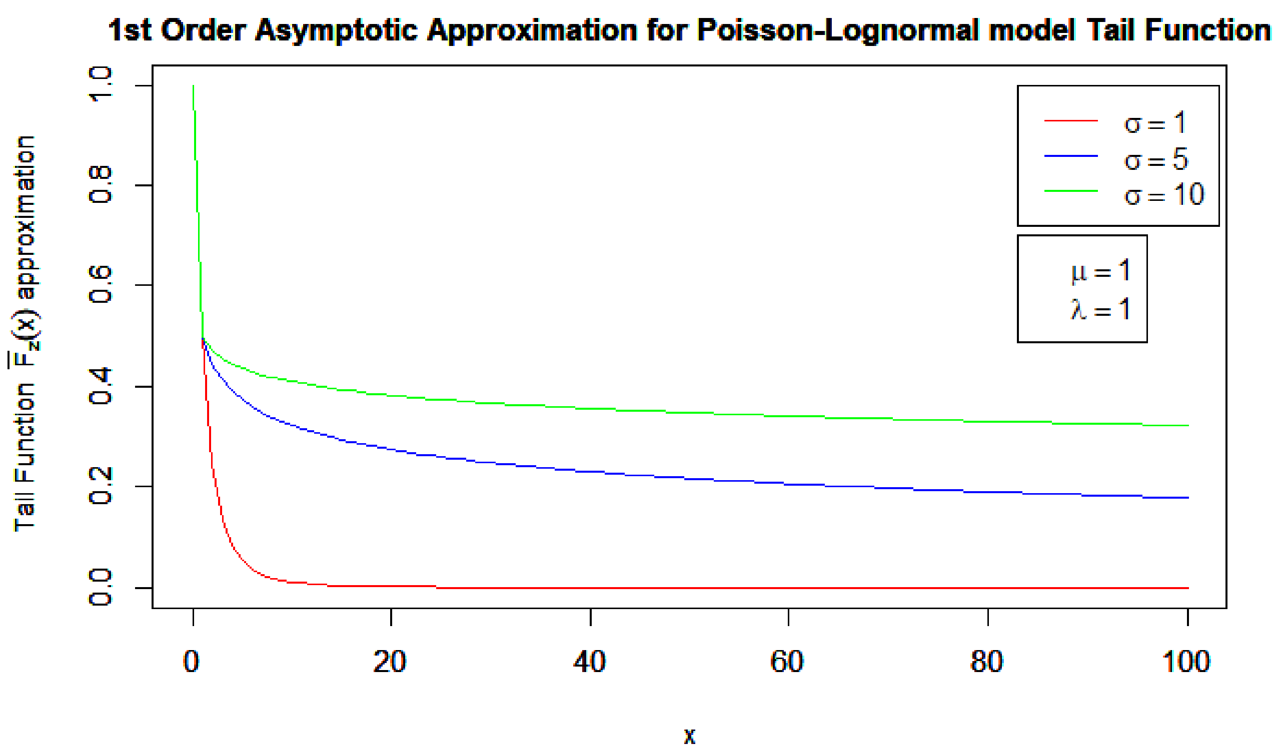
As suggested earlier, we can see tail behaviour does vary when we change μ from the Lognormal severity distribution, as for larger values of μ, we get heavier tails than smaller values. This makes sense as the mean (e^{μ}) and median ($e^{\mu+\frac{\sigma^2}{2}}$) increase as μ increases.

Hide

```
#Sigma tests_______________________________________________________
lambda_sigmatest <- 1
mu_sigmatest <- 0
sigmas = c(1,5,10) #We are going to plot the model for sigma = 1, 5 and 10
color = c("red","blue","green")

for(i in 1:3){
  PL = function(x) {
    lambda_sigmatest * (1/2 - 1/2*erf((log(x)-mu_sigmatest)/(sigmas[i]*sqrt(2))))
  }
  plot(x_s,PL(x_s),main = "1st Order Asymptotic Approximation for Poisson-Lognormal model Tail Function", cex.main=1.05 ,xlab = "x",ylab=TeX(r'(Tail Function  $\bar{F}_z(x)$ approximation)'),cex.lab=0.9,ylim = c(0,1),"l",col = color[i])
  par(new = TRUE)
}
legend(83,1,legend = c(TeX(r'($\sigma = 1)'),TeX(r'($\sigma = 5)'),TeX(r'($\sigma = 10)')),col = c("red","blue","green"),lty = 1)
```

Hide

```
legend(83,0.7, legend = c(TeX(r'($\mu = 1)'),TeX(r'($\lambda = 1)')))
```



We can also see we get heavier tails for larger values of σ than smaller values, which makes sense given that the median of the Lognormal severity ($e^{\mu+\frac{\sigma^2}{2}}$) increases as σ increases.