Probability and Statistics with R

Assignment 2

Submission Nov 16-2022 (Wednesday)

Group Members: Anna Maria KV (MDS202209) Sarvesh Bhandary Sneha KK(MDS202240)

```
knitr::opts_chunk$set(echo=TRUE,warning=FALSE, message = FALSE,
fig.height = 5, fig.width = 10, fig.align = "center")
```

Note: Below I explain how you collaborate on `GitHub` .

1. It will be group assignment.
2. A group would be of size at most 3. If you want to create a group size more than 3, you must take permission.
3. Decide among yourself and one of you create a GitHub repository for Probability Statistics Assignments.
4. In that repository add your group members as collaborator
5. Once you add your collaborator (or group members), create a folder and name it as `Assignment_2`
6. In that folder you should have 2 folders `code` and `report` . And one `README.md` file. Write a brief report in `README.md` file.
7. For each problem, you should create a separate GitHub `issue` . All your discussion should be documented in the `issue` .
8. In the issue mention clearly, which group member is taking ownership of what problem?
9. The other member should `fork` the repository in their GitHub account.
10. Once you have your forked the main repository in your GitHub account - you should clone the repository in you local laptop or just download it as zip.
11. Once you develop the code - you should `commit` the code first in your repository and then `push` it.
12. Finally you make the `pull-request` in the final repository.
13. Once a member make a pull request, the other members have to review the code.
14. While reviewing the code the reviewer may have to download the code and run the code in his or her system and reproduce the result.
15. If the result is reproduced then she or he would accept and merge the code in final repository.
16. At the end you submit the link of the repository in the moodle.
17. The entire process will be evaluated.

Problem 1

Suppose X denote the number of goals scored by home team in premier league. We can assume X is a random variable. Then we have to build the probability distribution to model the probability of number of goals. Since X takes value in $\mathbb{N} = \{0, 1, 2, \cdots\}$, we can consider the geometric progression sequence as possible candidate model, i.e.,

$$S = \{a, ar, ar^2, ar^3, \cdots\}.$$

But we have to be careful and put proper conditions in place and modify S in such a way so that it becomes proper probability distributions.

1. Figure out the necesary conditions and define the probability distribution model using S. We require the summation to be equal to 1. It is $\sum_{n=0} \mathbb{P}(X = n) = 1$ For geometric series to converge, we need $r < 1$ Sum of the series equals $\frac{a}{1-r}$ Therefore $a = 1 - r$ Probabilities should lie between 0 and 1. Hence $0 < r < 1$ and $a = 1 - r$

$\mathbb{P}(X = n) = (1 - r)^n r$, which is the pmf of Geometric(1-r).

Hence X follows Geometric distribution.

2. Check if mean and variance exists for the probability model. Mean and variance exists for geometric distribution. Hence it exists for the given probability model also.

3. Can you find the analytically expression of mean and variance. \begin{itemize}

Mean

$$\begin{aligned} E[X] &= \sum_{n=0} n\mathbb{P}(X=n) \\ &= \sum_{n=0} n(1-r)r^n \\ &= r\sum_{n=0} n(1-r)r^{n-1} \\ &= r\sum_{n=1}(n-1)(1-r)r^{n-1} + r\sum_{n=1}(1-r)r^{n-1} \\ &= rE[X] + r(1-r)(\frac{1}{1-r}) \\ &= rE[X] + r \end{aligned}$$

$$\text{Hence, } E[X] = \frac{r}{1-r}$$

Variance

$$\begin{aligned} E[X^2] &= \sum_{n=0} n^2\mathbb{P}(X=n) \\ &= \sum_{n=0} n^2(1-r)r^n \\ &= r\sum_{n=0} n^2(1-r)r^{n-1} \\ &= r\sum_{n=1}(n-1)^2(1-r)r^{n-1} + \sum_{n=1} 2n(1-r)r^n - r\sum_{n=1}(1-r)r^{n-1} \\ &= rE[X^2] + 2E[X] - r(1-r)\frac{1}{1-r} \\ &= rE[X^2] + \frac{2r}{1-r} - r \end{aligned}$$

$$\text{Hence, } E[X^2] = \frac{r^2+r}{(1-r)^2}. \text{ Therefore,}$$

$$\begin{aligned} Var(X) &= E[X^2] - (E[X])^2 \\ &= \frac{r^2+r}{(1-r)^2} - \frac{r^2}{(1-r)^2} \\ &= \frac{r}{(1-r)^2} \end{aligned}$$

\end{itemize}

4. From historical data we found the following summary statistics Using the summary statistics and your newly defined probability distribution model find the following:

a. What is the probability that home team will score at least one goal?
b. What is the probability that home team will score at least one goal but less than four goal?

We can see that the mean and the variance of the distribution, when equated to $\frac{r}{1-r}$ and $\frac{r}{(1-r)^2}$ yield different values of r, and hence it is not possible for this to happen simultaneously.

5. Suppose on another thought you want to model it with off-the shelf Poisson probability models. Under the assumption that underlying distribution is Poisson probability find the above probabilities, i.e.,

a. What is the probability that home team will score at least one goal?
b. What is the probability that home team will score at least one goal but less than four goal?

6. Which probability model you would prefer over another?

7. Write down the likelihood functions of your newly defined probability models and Poisson models. Clearly mention all the assumptions that you are making.

Mean and variance of Poisson distribution must be same and hence the given values are not possible.

Problem 2 : Simulation Study to Understand Sampling Distribution

Part A Suppose $X_1, X_2, \cdots, X_n \overset{iid}{\sim} Gamma(\alpha, \sigma)$, with pdf as

$$f(x|\alpha,\sigma) = \frac{1}{\sigma^{\alpha}\Gamma(\alpha)} e^{-x/\sigma} x^{\alpha-1}, \quad 0 < x < \infty,$$

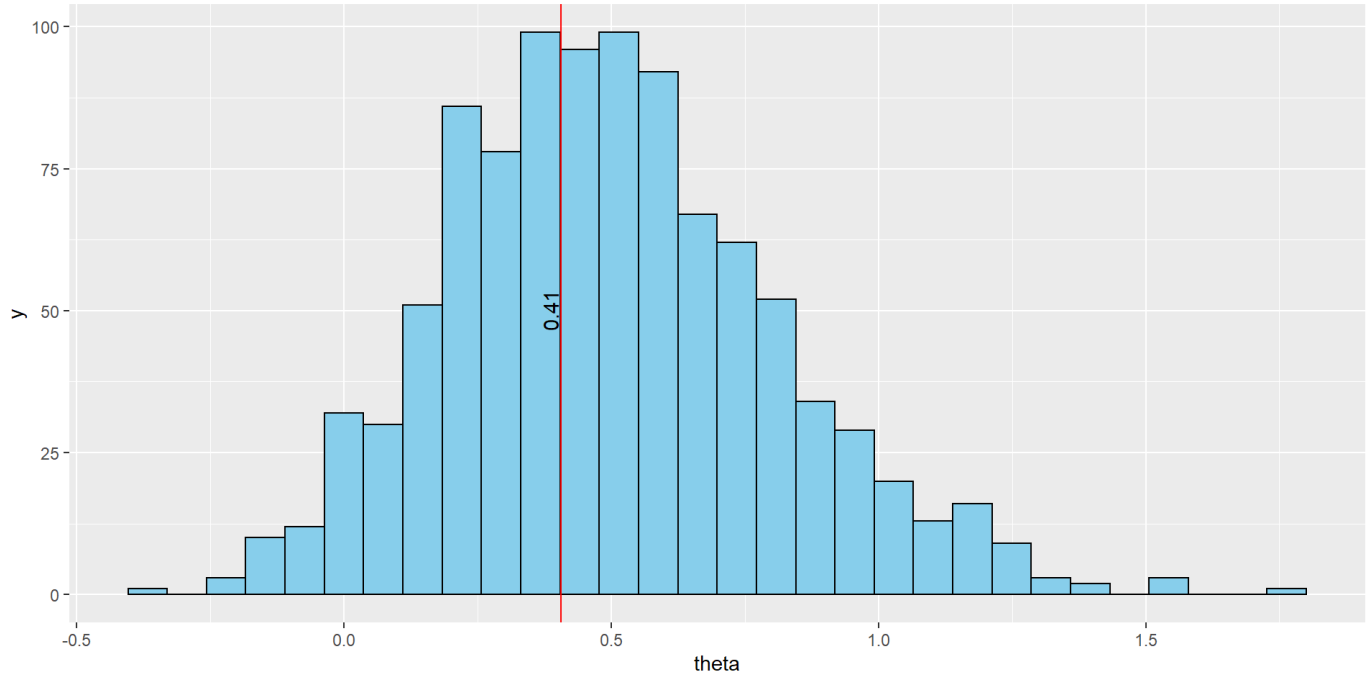
The mean and variance are $E(X) = \alpha\sigma$ and $Var(X) = \alpha\sigma^2$. Note that `shape` = α and `scale` = σ.

1. Write a `function` in `R` which will compute the MLE of $\theta = \log(\alpha)$ using `optim` function in `R` . You can name it `MyMLE`

```
MyMLE=function(obs,st){
  logsum=function(params,x){
    return(sum(+params[1]*log(params[2])+log(gamma(params[1]))+(x/params[2])-(params[1]-1)*log(x)))
  }
  pars=optim(par=st,logsum,x=obs)$par
  return(log(pars[1]))
}
```

2. Choose `n=20` , and `alpha=1.5` and `sigma=2.2`
   i. Simulate $\{X_1, X_2, \cdots, X_n\}$ from `rgamma(n=20,shape=1.5,scale=2.2)`
   ii. Apply the `MyMLE` to estimate θ and append the value in a vector
   iii. Repeat the step (i) and (ii) 1000 times
   iv. Draw histogram of the estimated MLEs of θ.
   v. Draw a vertical line using `abline` function at the true value of θ.
   vi. Use `quantile` function on estimated θ's to find the 2.5 and 97.5-percentile points.

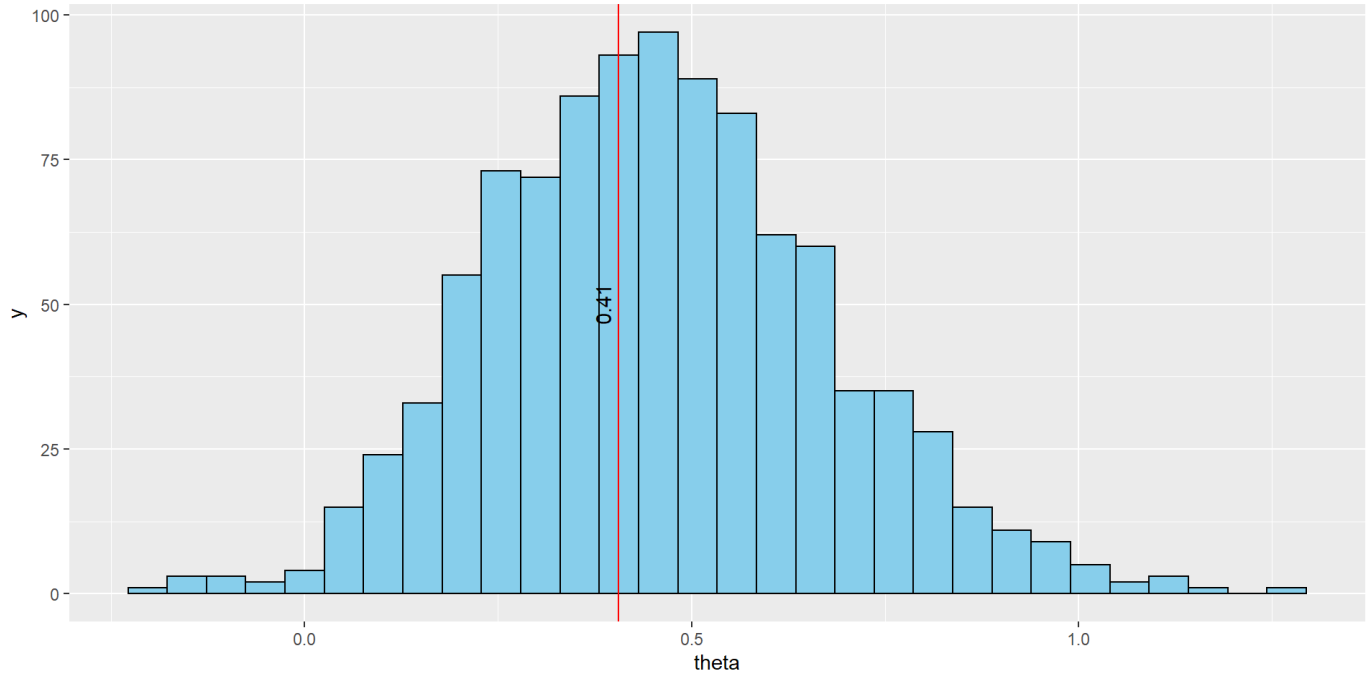
```
simu<-function(n,simu_size,alpha,sigma){
theta=c()
for (i in 1:simu_size){
  obs=rgamma(n=n,shape=alpha,scale=sigma)
  theta=append(theta,MyMLE(obs,c(2,2)))
}
library(ggplot2)
df_theta=as.data.frame(theta)
val=log(1.5)
plot<-ggplot(df_theta)+geom_histogram(aes(x=theta),fill='skyblue',color='black')+
geom_vline(xintercept=val, color='red')+annotate("text",x=val-0.02,y=50,label=as.character(round(va
l,2)),angle=90)
q25=quantile(theta,probs=seq(0,1,0.025))
q75=quantile(theta,probs=seq(0,1,0.975))
print(plot)
cat(cat("The 2.5 th percentile is",q25[2]),sep='\n')
cat(cat("The 97.5th percentile is",q75[2]),sep='\n')
cat(cat("The gap between 2.5 and 97.5 percentile points is ",q75[2]-q25[2]),sep='\n')
return(q75[2]-q25[2])}
q2<-simu(20,1000,1.5,2.2)
```




```
## The 2.5 th percentile is -0.03707148
## The 97.5th percentile is 1.16822
## The gap between 2.5 and 97.5 percentile points is  1.205292
```

3. Choose `n=40` , and `alpha=1.5` and repeat the (2).

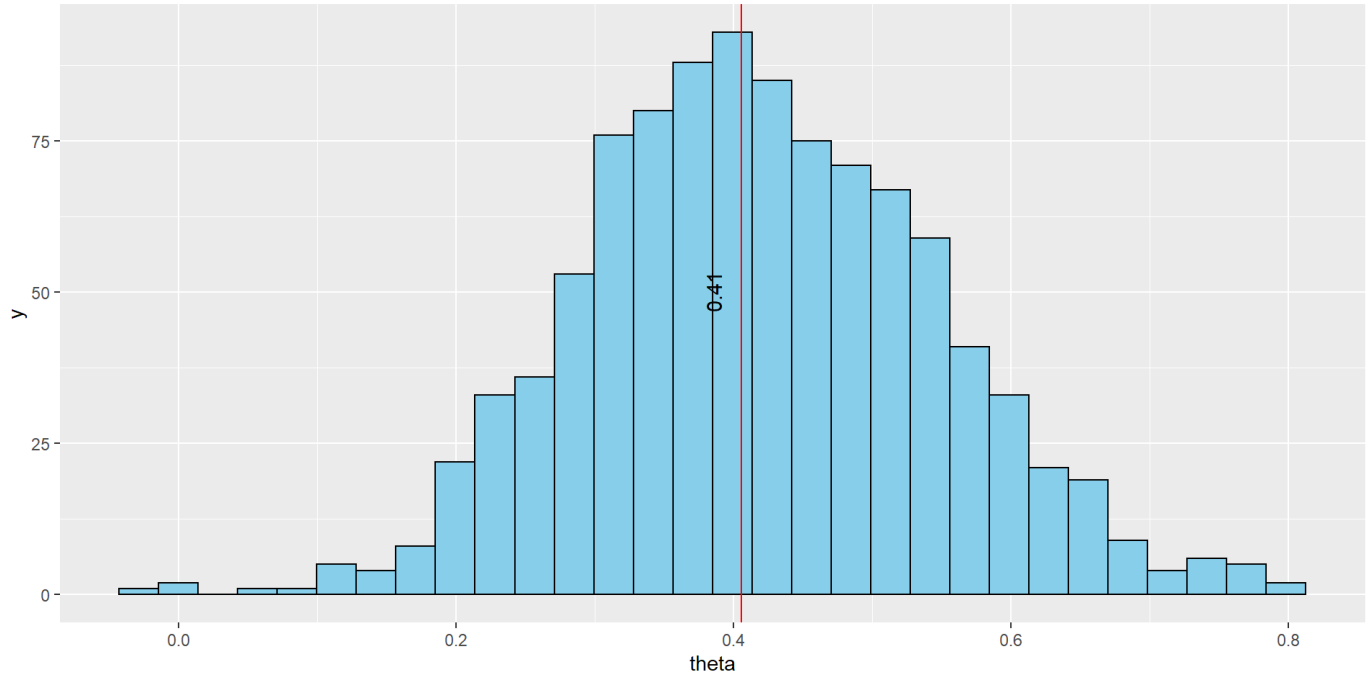
```
q3<-simu(40,1000,1.5,2.2)
```




```
## The 2.5 th percentile is 0.07005775
## The 97.5th percentile is 0.9132483
## The gap between 2.5 and 97.5 percentile points is  0.8431905
```

4. Choose `n=100` , and `alpha=1.5` and repeat the (2).

```
q4<-simu(100,1000,1.5,2.2)
```




```
## The 2.5 th percentile is 0.1885243
## The 97.5th percentile is 0.6718747
## The gap between 2.5 and 97.5 percentile points is  0.4833504
```

5. Check if the gap between 2.5 and 97.5-percentile points are shrinking as sample size `n` is increasing?

```
cat("The values for n=20,40,100 are",q2,",",q3,",",q4,"respectively. As we can see,the gap between
 2.5 and 97.5 percentile points are shrinking as sample size is increasing")
```

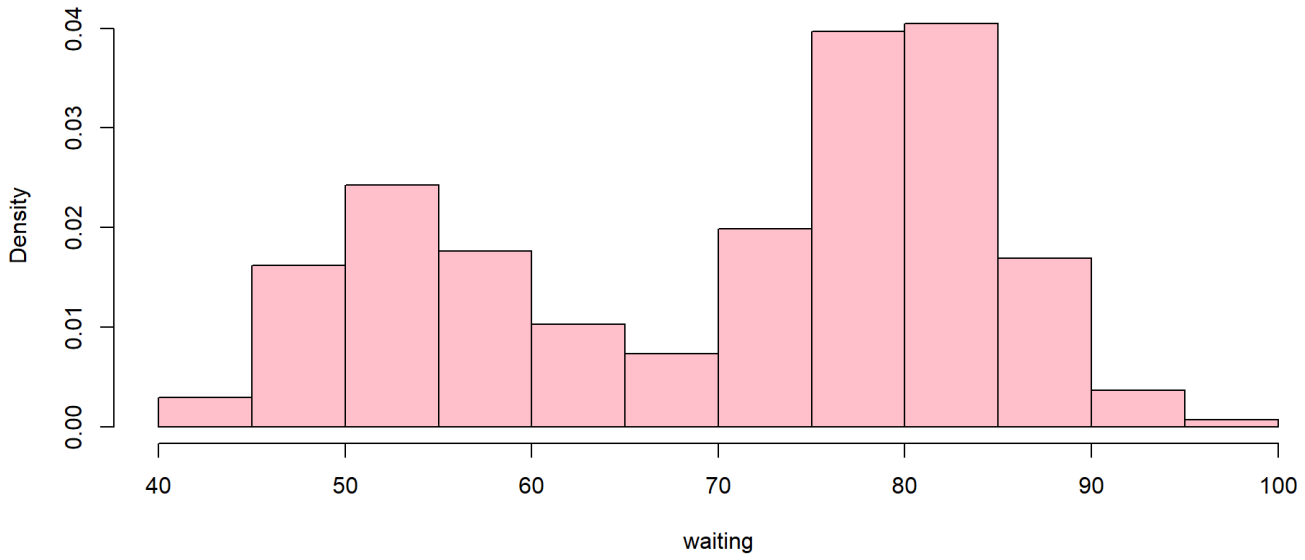
```
## The values for n=20,40,100 are 1.205292 , 0.8431905 , 0.4833504 respectively. As we can see,the
gap between 2.5 and 97.5 percentile points are shrinking as sample size is increasing
```

Hint: Perhaps you should think of writing a single `function` where you will provide the values of `n`, `sim_size`, `alpha` and `sigma`; and it will return the desired output.

Problem 3: Analysis of `faithful` datasets.

Consider the `faithful` datasets:

```
attach(faithful)
hist(faithful$waiting,xlab = 'waiting',probability = T,col='pink',main='')
```



Fit following three models using MLE method and calculate **Akaike information criterion** (aka., AIC) for each fitted model. Based on AIC decides which model is the best model? Based on the best model calculate the following probability

$$\mathbb{P}(60 < \texttt{waiting} < 70)$$

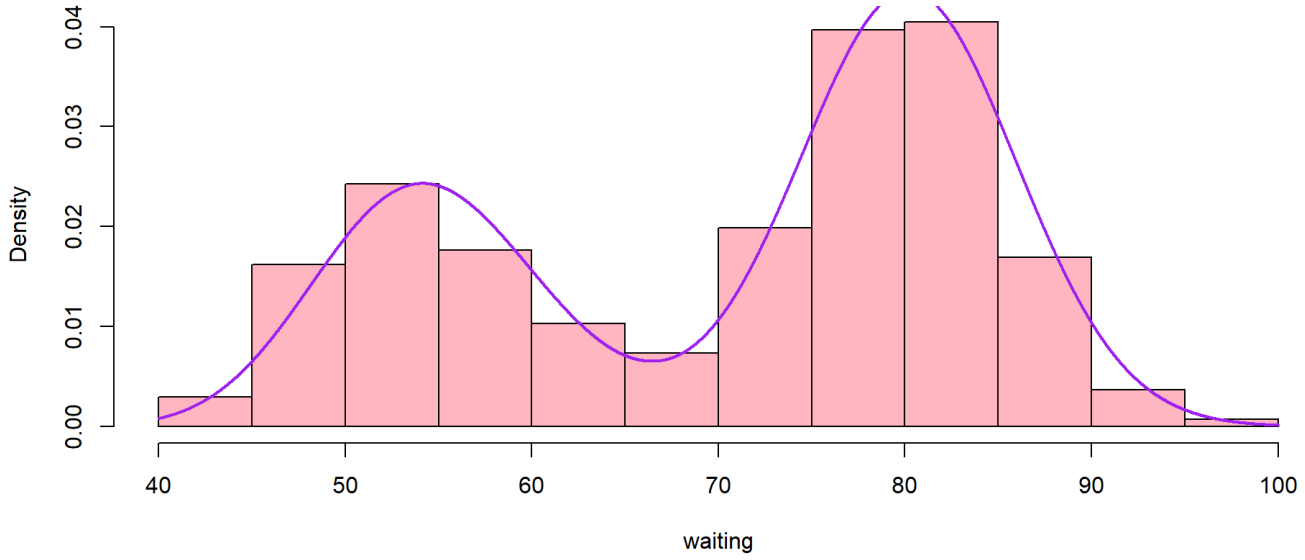
i. **Model 1**:

$$f(x) = p * Gamma(x|\alpha, \sigma_1) + (1-p)N(x|\mu, \sigma_2^2), \;\; 0 < p < 1$$

```
x = faithful$waiting
MLE1 = function(x, st){
nll1 = function(x, par){
  return (sum(-log(par[1]*dgamma(x,par[2],par[3]) + (1-par[1])*dnorm(x,par[4],par[5]))))
}
pars = optim(par = st, nll1, x=x, lower = c(0,0,0,-Inf,0), upper=c(1,Inf,Inf,Inf,Inf), method = 'L-BFGS-B')$par
cat(cat('The optimal values of p, alpha, sigma_1, mu. sigma_2 is', pars),sep='\n')
cat('AIC =', 2*nll1(x, pars) + 2*length(pars))
return (pars)
}
pars = MLE1(x,c(0.5,55,1,85,1))
```

```
## The optimal values of p, alpha, sigma_1, mu. sigma_2 is 0.3652577 82.78556 1.510993 80.16598 5.810134
## AIC = 2076.18
```

```
x = seq(40,100,length.out=10000)
f=function(x, par){
  return (par[1]*dgamma(x,par[2],par[3]) + (1-par[1])*dnorm(x,par[4],par[5]))
}
hist(faithful$waiting,xlab = 'waiting',probability = T,col='lightpink',main='')
lines(x,f(x,pars), col = 'purple',lwd=2)
```



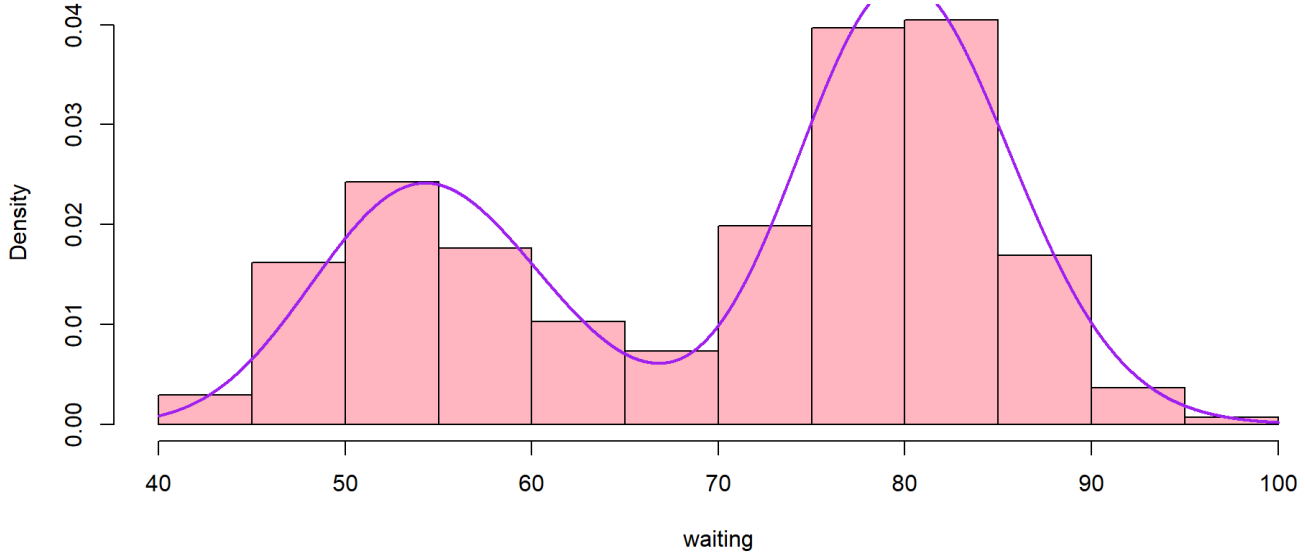
ii. **Model 2**:

$$f(x) = p * Gamma(x|\alpha_1, \sigma_1) + (1-p)Gamma(x|\alpha_2, \sigma_2), \;\; 0 < p < 1$$

```
x = faithful$waiting
MLE2 = function(x, st){
nll2 = function(x, par){
  return (sum(-log(par[1]*dgamma(x,par[2],par[3]) + (1-par[1])*dgamma(x,par[4],par[5]))))
}
pars = optim(par = st, nll2, x=x, lower = c(0,0,0,0,0), upper=c(1,Inf,Inf,Inf,Inf), method = 'L-BFG
S-B')$par
cat(cat('The optimal values of p, alpha, sigma_1, mu. sigma_2 is', pars),sep='\n')
cat('AIC =', 2*nll2(x, pars) + 2*length(pars))
return (pars)
}
pars = MLE2(x,c(0.5,55,1,85,1))
```

```
## The optimal values of p, alpha, sigma_1, mu. sigma_2 is 0.3709333 79.66891 1.449311 199.7652 2.4
88091
## AIC = 2076.116
```

```
x = seq(40,100,length.out=10000)
f=function(x, par){
  return (par[1]*dgamma(x,par[2],par[3]) + (1-par[1])*dgamma(x,par[4],par[5]))
}
hist(faithful$waiting,xlab = 'waiting',probability = T,col='lightpink',main='')
lines(x,f(x,pars), col = 'purple',lwd=2)
```



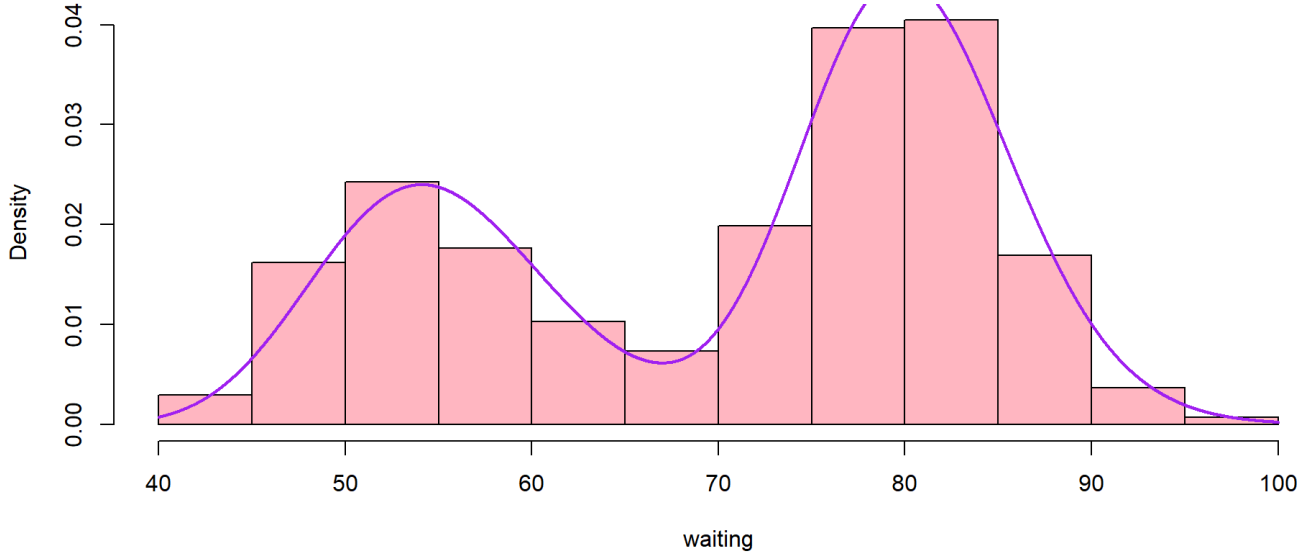
iii. **Model 3**:

$$f(x) = p * logNormal(x|\mu_1, \sigma_1^2) + (1-p)logNormal(x|\mu_1, \sigma_1^2), \;\; 0 < p < 1$$

```
x = faithful$waiting
MLE3 = function(x, st){
nll3 = function(x, par){
  return (sum(-log(par[1]*dlnorm(x,par[2],par[3]) + (1-par[1])*dlnorm(x,par[4],par[5]))))
}
pars = optim(par = st, nll3, x=x, lower = c(0,-Inf,0,-Inf,0), upper=c(1,Inf,Inf,Inf,Inf), method =
'L-BFGS-B')$par
cat(cat('The optimal values of p, alpha, sigma_1, mu. sigma_2 is', pars),sep='\n')
cat('AIC =', 2*nll3(x, pars) + 2*length(pars))
return (pars)
}
pars = MLE3(x,c(0.5,4,1,4.4,1))
```

```
## The optimal values of p, alpha, sigma_1, mu. sigma_2 is 0.6238638 4.384306 0.06972831 4.003846
0.1148513
## AIC = 2075.42
```

```
x = seq(40,100,length.out=10000)
f=function(x, par){
  return (par[1]*dlnorm(x,par[2],par[3]) + (1-par[1])*dlnorm(x,par[4],par[5]))
}
hist(faithful$waiting,xlab = 'waiting',probability = T,col='lightpink',main='')
lines(x,f(x,pars), col = 'purple',lwd=2)
```



The AIC values we got are: 2076.18, 2076.116, 2075.42, so clearly since the AIC value for the third model is lower, the log likelihood for third model will be the highest, and hence model 3 is the best.

$$P(60 < waiting < 70) = P(X < 70) - P(X < 60)$$

where X has pdf in Model 3

```
pars[1]*(plnorm(70,pars[2],pars[3])-plnorm(60,pars[2],pars[3]))+(1-pars[1])*(plnorm(70,pars[4],pars[5])-plnorm(60,pars[4],pars[5]))
```

```
## [1] 0.09080635
```

Problem 4: Modelling Insurance Claims

Consider the `Insurance` datasets in the `MASS` package. The data given in data frame `Insurance` consist of the numbers of policyholders of an insurance company who were exposed to risk, and the numbers of car insurance claims made by those policyholders in the third quarter of 1973.

This data frame contains the following columns:

`District` (factor): district of residence of policyholder (1 to 4): 4 is major cities.

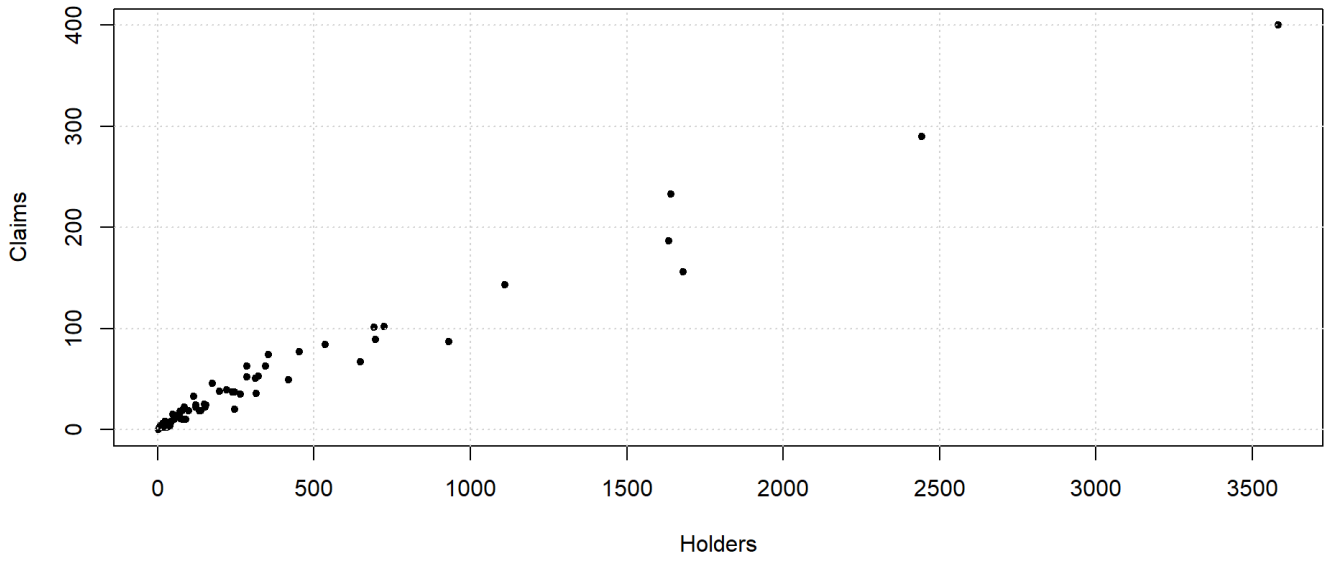
`Group` (an ordered factor): group of car with levels <1 litre, 1–1.5 litre, 1.5–2 litre, >2 litre.

`Age` (an ordered factor): the age of the insured in 4 groups labelled <25, 25–29, 30–35, >35.

`Holders` : numbers of policyholders.

`Claims` : numbers of claims

```
library(MASS)
attach(Insurance)
data = Insurance[,c('Holders','Claims')]
plot(Insurance$Holders,Insurance$Claims
     ,xlab = 'Holders',ylab='Claims',pch=20)
grid()
```



Note: If you use built-in function like `lm` or any packages then no points will be awarded.

Part A: We want to predict the `Claims` as function of `Holders` . So we want to fit the following models:

$$\texttt{Claims}_i = \beta_0 + \beta_1\, \texttt{Holders}_i + \varepsilon_i, \quad i = 1, 2, \cdots, n$$

Assume : $\varepsilon_i \sim N(0, \sigma^2)$. Note that $\beta_0, \beta_1 \in \mathbb{R}$ and $\sigma \in \mathbb{R}^+$.

The above model can alse be re-expressed as,

$$\texttt{Claims}_i \sim N(\mu_i, \sigma^2), \quad where$$

$$\mu_i = \beta_0 + \beta_1\, \texttt{Holders}_i + \varepsilon_i, \quad i = 1, 2, \cdots, n$$

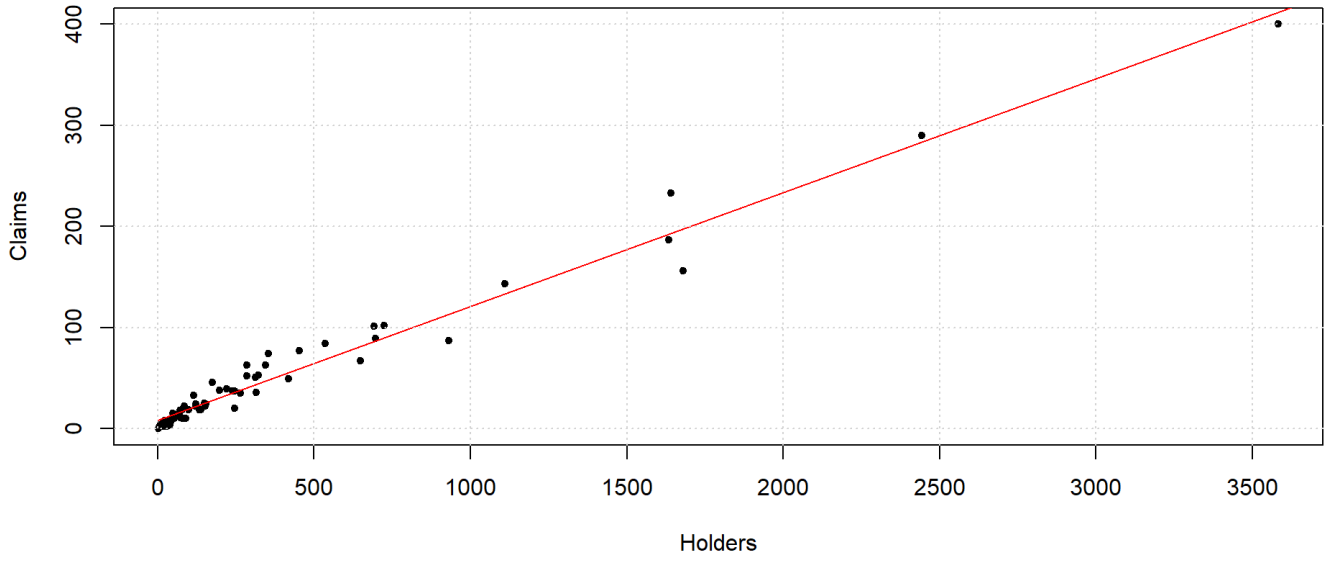
i. Clearly write down the negative-log-likelihood function in `R` . Then use `optim` function to estimate MLE of $\theta = (\beta_0, \beta_1, \sigma)$

ii. Calculate **Bayesian Information Criterion** (BIC) for the model.

```
data = Insurance[,c('Holders','Claims')]
n=nrow(data)
Norm = function(p0){
  NLL = function(x,par){
    mean = with(x, par[1]+par[2]*Holders)
     f1 = function(x,mean,sigma){
       return ((exp(-0.5*((x-mean)^2/(sigma^2))))/(sigma*sqrt(2*pi)))
     }
     LL = c()
     for (i in 1:n){
       LL = c(LL, f1(x[i,'Claims'],mean[i],par[3]))
     }
     return (-sum(log(LL)))
  }
 fit=optim(par=p0,NLL,x=data)
  cat(cat("MLE=",fit$par),sep="\n")
  BIC = 3*log(n)+2*fit$value
  cat("BIC =",BIC)
  return(fit$par)
}
pars = Norm(c(0,0.1,100))
```

```
## MLE= 8.121281 0.1126417 11.86943
## BIC = 510.7587
```

```
plot(Insurance$Holders,Insurance$Claims ,xlab = 'Holders',ylab='Claims',pch=20)
x = seq(0,4000,length.out=10000)
lines(x,pars[1]+pars[2]*x,col='red')
grid()
```



Part B: Now we want to fit the same model with change in distribution:

$$\texttt{Claims}_i = \beta_0 + \beta_1 \, \texttt{Holders}_i + \varepsilon_i, \quad i = 1, 2, \cdots, n$$

Assume : $\varepsilon_i \sim Laplace(0, \sigma^2)$. Note that $\beta_0, \beta_1 \in \mathbb{R}$ and $\sigma \in \mathbb{R}^+$.

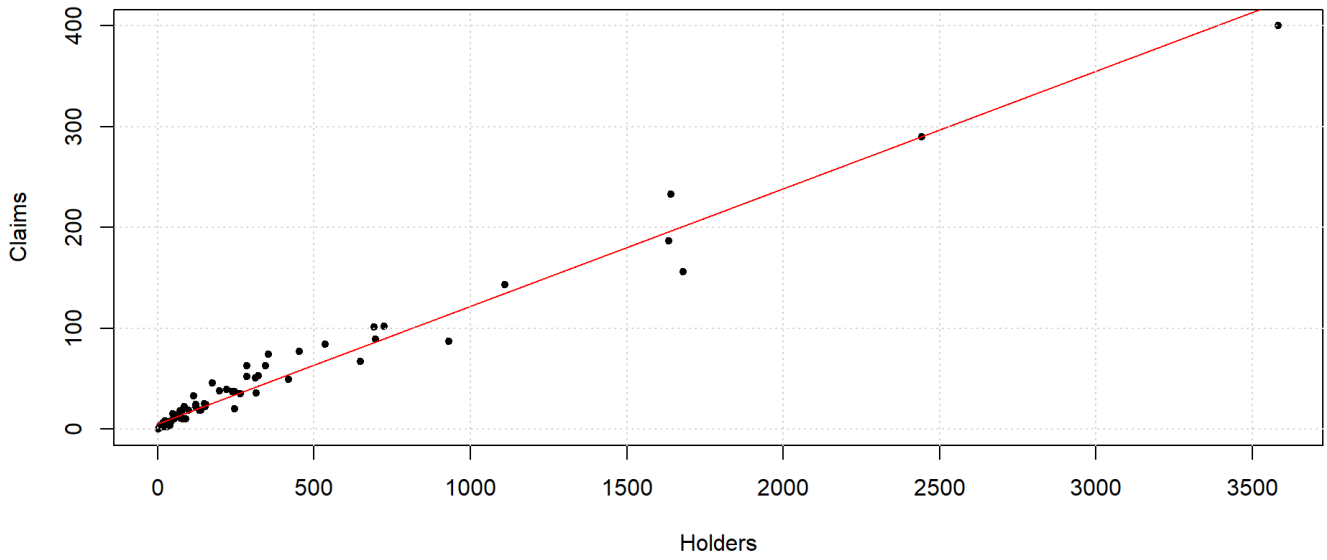
i. Clearly write down the negative-log-likelihood function in `R`. Then use `optim` function to estimate MLE of $\theta = (\beta_0, \beta_1, \sigma)$

ii. Calculate **Bayesian Information Criterion** (BIC) for the model.

```
n=nrow(data)
Lapl = function(p0){
  NLL = function(x,par){
    mean = with(x, par[1]+par[2]*Holders)
     f2 = function(x,mean,sigma){
      return ((exp(-(abs(x-mean)/(sigma))))/(2*sigma))
    }
     LL = c()
     for (i in 1:n){
       LL = c(LL, f2(x[i,'Claims'],mean[i],par[3]))
     }
     return (-sum(log(LL)))
  }
 fit=optim(par=p0,NLL,x=data)
  cat(cat("MLE=",fit$par),sep="\n")
  BIC = 3*log(n)+2*fit$value
  cat("BIC =",BIC)
  return(fit$par)
}
pars = Lapl(c(0,0.1,100))
```

```
## MLE= 5.084407 0.1166253 8.213428
## BIC = 498.6869
```

```
plot(Insurance$Holders,Insurance$Claims ,xlab = 'Holders',ylab='Claims',pch=20)
x = seq(0,4000,length.out=10000)
lines(x,pars[1]+pars[2]*x,col='red')
grid()
```



Part C: We want to fit the following models:

$$\texttt{Claims}_i \sim LogNormal(\mu_i, \sigma^2), where$$

$$\mu_i = \beta_0 + \beta_1 \log(\texttt{Holders}_i), \;\; i = 1, 2, \ldots, n$$

Note that $\beta_0, \beta_1 \in \mathbb{R}$ and $\sigma \in \mathbb{R}^+$.

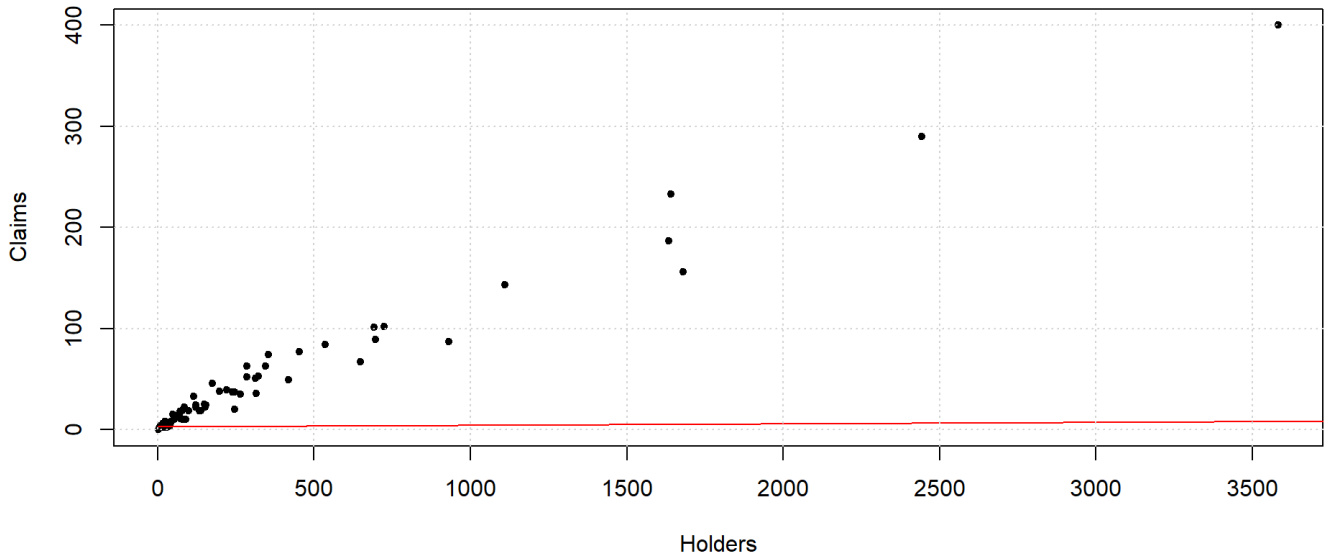
i. Clearly write down the negative-log-likelihood function in `R`. Then use `optim` function to estimate MLE of $\theta = (\alpha, \beta, \sigma)$

ii. Calculate **Bayesian Information Criterion** (BIC) for the model.

```
data1<-data[data$Claims !=0,]
row.names(data1)<-NULL
n=nrow(data1)
Lognorm = function(p0){
  NLL = function(x,par){
    mean = with(x, par[1]+par[2]*Holders)
     f3 = function(x,mean,sigma){
       return ((exp(-0.5*((log(x)-mean)^2/(sigma^2))))/(x*sigma*sqrt(2*pi)))
     }
     LL = c()
     for (i in 1:n){
       LL = c(LL, f3(x[i,'Claims'],mean[i],par[3]))
     }
     return (-sum(log(LL)))
  }
 fit=optim(par=p0,NLL,x=data1)
  cat(cat("MLE=",fit$par),sep="\n")
  BIC = 3*log(n)+2*fit$value
  cat("BIC =",BIC)
  return(fit$par)
}
pars = Lognorm(c(0,0.0001,0.3))
```

```
## MLE= 2.639779 0.001472126 0.8229641
## BIC = 567.9726
```

```
plot(Insurance$Holders,Insurance$Claims ,xlab = 'Holders',ylab='Claims',pch=20)
x = seq(0,4000,length.out=10000)
lines(x,pars[1]+pars[2]*x,col='red')
grid()
```



Part D: We want to fit the following models:

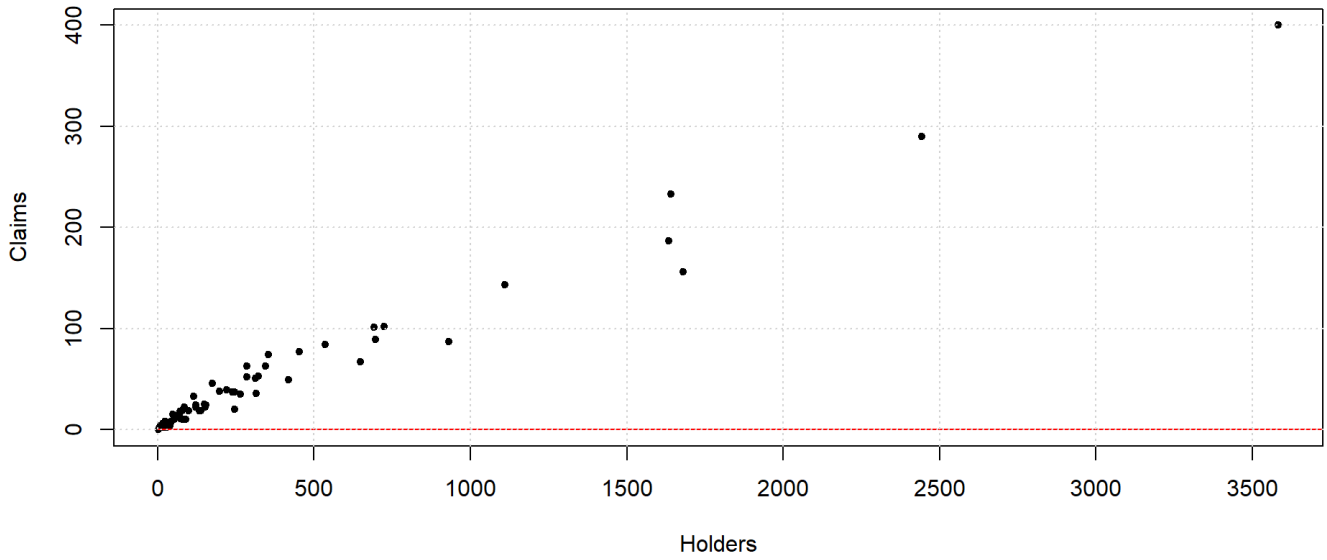
$$\texttt{Claims}_i \sim Gamma(\alpha_i, \sigma), where$$

$$log(\alpha_i) = \beta_0 + \beta_1 \log(\texttt{Holders}_i), \;\; i = 1, 2, \ldots, n$$

```
n=nrow(data)
Gamma = function(p0){
  NLL = function(x,par){
    alpha = with(x, exp(par[1]+par[2]*log(Holders)))
     f4 = function(x,alpha,sigma){
      return ((exp(-x/sigma)*x^(alpha-1))/((sigma^alpha)*gamma(alpha)))
    }
     LL = c()
     for (i in 1:n){
       LL = c(LL, f4(x[i,'Claims'],(alpha[i]),par[3]))
     }
     return (-sum(log(LL)))
  }
 fit=optim(par=p0,NLL,x=data)
  cat(cat("MLE=",fit$par),sep="\n")
  BIC = 3*log(n)+2*fit$value
  cat("BIC =",BIC)
  return(fit$par)
}
pars = Gamma(c(0,0,100))
```

```
## MLE= 0 0 100
## BIC = 664.9584
```

```
plot(Insurance$Holders,Insurance$Claims ,xlab = 'Holders',ylab='Claims',pch=20)
x = seq(0,4000,length.out=10000)
lines(x,pars[1]+pars[2]*x,col='red')
grid()
```



iii. Compare the BIC of all three models BIC for Laplace < Normal < Log Normal < Gamma, so Laplace is the best fit for this data.

Problem 5: Computational Finance - Modelling Stock prices

Following piece of code download the prices of TCS since 2007

```
library(quantmod)
getSymbols('TCS.NS')
```

```
## [1] "TCS.NS"
```

```
tail(TCS.NS)
```

```
##            TCS.NS.Open TCS.NS.High TCS.NS.Low TCS.NS.Close TCS.NS.Volume
## 2022-11-04      3217.0     3220.05    3166.15       3217.40       1464013
## 2022-11-07      3229.0     3242.80    3195.10       3233.70       1474498
## 2022-11-09      3249.8     3249.80    3201.65       3216.05       1162267
## 2022-11-10      3170.0     3225.00    3170.00       3205.65       1573092
## 2022-11-11      3269.6     3341.60    3255.05       3315.95       3265394
## 2022-11-14      3324.0     3349.00    3309.00       3335.50       1342074
##            TCS.NS.Adjusted
## 2022-11-04         3217.40
## 2022-11-07         3233.70
## 2022-11-09         3216.05
## 2022-11-10         3205.65
## 2022-11-11         3315.95
## 2022-11-14         3335.50
```

Plot the adjusted close prices of TCS

```
plot(TCS.NS$TCS.NS.Adjusted)
```

Download the data of market index Nifty50. The Nifty 50 index indicates how the over all market has done over the similar period.

```
getSymbols('^NSEI')
```

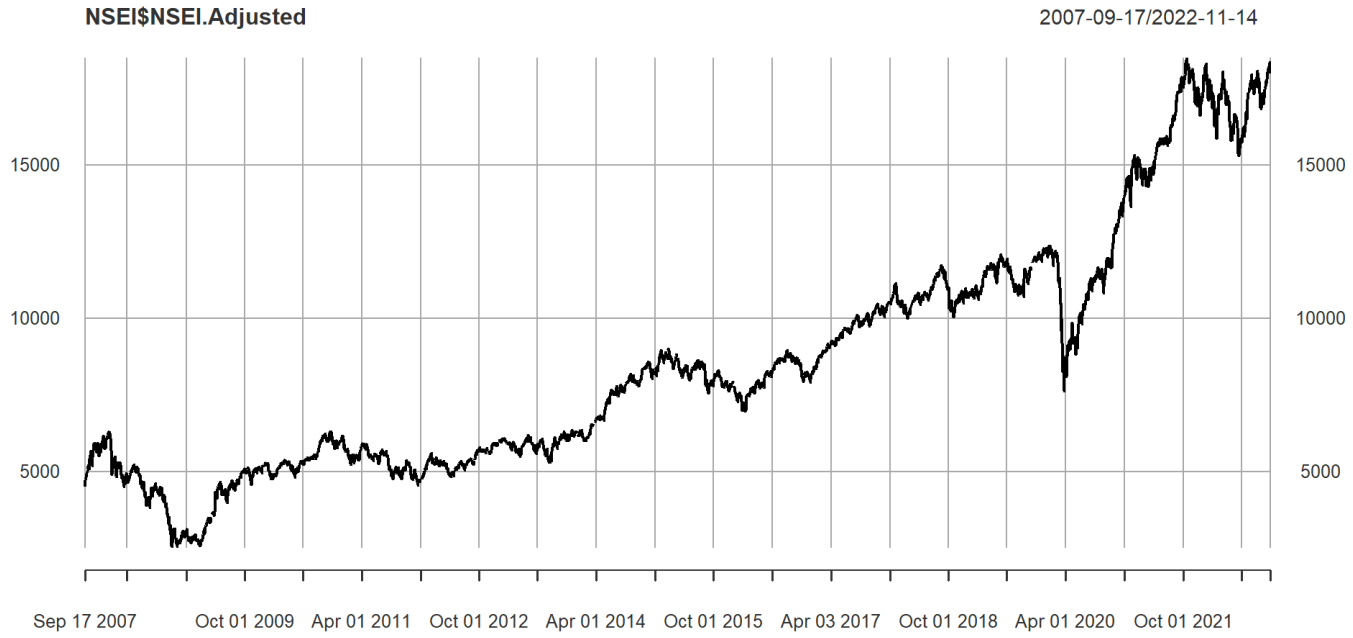
```
## [1] "^NSEI"
```

```
tail(NSEI)
```

```
##            NSEI.Open NSEI.High NSEI.Low NSEI.Close NSEI.Volume NSEI.Adjusted
## 2022-11-04  18053.40  18135.10 18017.15   18117.15      267900      18117.15
## 2022-11-07  18211.75  18255.50 18064.75   18202.80      314800      18202.80
## 2022-11-09  18288.25  18296.40 18117.50   18157.00      307200      18157.00
## 2022-11-10  18044.35  18103.10 17969.40   18028.20      256500      18028.20
## 2022-11-11  18272.35  18362.30 18259.35   18349.70      378500      18349.70
## 2022-11-14  18376.40  18399.45 18311.40   18329.15      301400      18329.15
```

Plot the adjusted close value of Nifty50

```
plot(NSEI$NSEI.Adjusted)
```



Log-Return

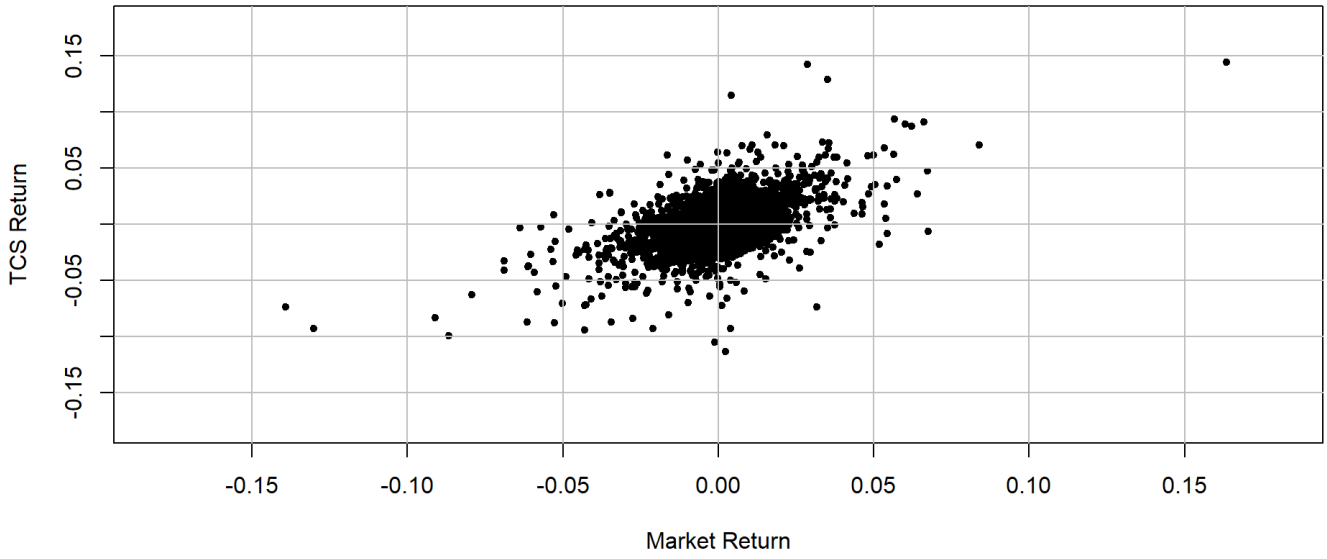
We calculate the daily log-return, where log-return is defined as

$$r_t = \log(P_t) - \log(P_{t-1}) = \Delta \log(P_t),$$

where P_t is the closing price of the stock on t^{th} day.

```
TCS_rt = diff(log(TCS.NS$TCS.NS.Adjusted))
Nifty_rt = diff(log(NSEI$NSEI.Adjusted))
retrn = cbind.xts(TCS_rt,Nifty_rt)
retrn = na.omit(data.frame(retrn))

plot(retrn$NSEI.Adjusted,retrn$TCS.NS.Adjusted
     ,pch=20
     ,xlab='Market Return'
     ,ylab='TCS Return'
     ,xlim=c(-0.18,0.18)
     ,ylim=c(-0.18,0.18))
grid(col='grey',lty=1)
```



- Consider the following model:

$$r_t^{TCS} = \alpha + \beta r_t^{Nifty} + \varepsilon,$$

where $\mathbb{E}(\varepsilon) = 0$ and $\mathbb{V}ar(\varepsilon) = \sigma^2$.

1. Estimate the parameters of the models $\theta = (\alpha, \beta, \sigma)$ using the method of moments type plug-in estimator discussed in the class.

```
E_tcs=mean(retrn$TCS.NS.Adjusted)
E_nif=mean(retrn$NSEI.Adjusted)
Var_tcs=var(retrn$TCS.NS.Adjusted)
Var_nif=var(retrn$NSEI.Adjusted)
Cov=cov(retrn$TCS.NS.Adjusted,retrn$NSEI.Adjusted)
Beta_mom=Cov/Var_nif
Alpha_mom=E_tcs-Beta_mom*E_nif
Sigma_mom=sqrt(Var_tcs-((Beta_mom)^2*Var_nif))
cat("Method of moments plug in estimator of theta is: (",Alpha_mom,",",Beta_mom,",",Sigma_mom,")")
```

```
## Method of moments plug in estimator of theta is: ( 0.0004628221 , 0.7436845 , 0.01618466 )
```

2. Estimate the parameters using the `lm` built-in function of `R`. Note that `lm` using the OLS method.

```
lmpar=summary(lm(TCS.NS.Adjusted~NSEI.Adjusted,data=retrn))
Beta_lm=lmpar$coef[2,1]
Alpha_lm=lmpar$coef[1,1]
Sigma_lm=lmpar$sigma
cat("Estimator of theta using lm function is: (",Alpha_lm,",",Beta_lm,",",Sigma_lm,")")
```

```
## Estimator of theta using lm function is: ( 0.0004628221 , 0.7436845 , 0.01618686 )
```

3. Fill-up the following table

Parameters	Method of Moments	OLS
α	0.0004611203	0.0004611203
β	0.7436965	0.7436965

Parameters	Method of Moments	OLS
σ	0.01618653	0.01618873

4. If the current value of Nifty is 18000 and it goes up to 18200. The current value of TCS is Rs. 3200/-. How much you can expect TCS price to go up?

```
P_TCS=3200*exp(Alpha_mom+Beta_mom*log(18200/18000))
cat("We can expect the TCS price to go up to",P_TCS)
```

```
## We can expect the TCS price to go up to 3227.898
```