Probability and Statistics with R

Assignment 2

Submission Nov 16-2022 (Wednesday)

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(n,shape,scale)
{
  n<<-n
  Negloglike=function(data,theta)
  {

    l=0
    for(i in 1:n)
    {
      l=l+log(dgamma(data[i], theta[1],scale =theta[2]))

    }
    return(-l)
  }

  theta=c(0.1,0.1)


  sim=rgamma(n,shape,scale)
  data=sim
  log(optim(par=theta,Negloglike,data=sim)$par[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)`

```r
rgamma(20,1.5,scale=2.2)
```

```
##  [1] 0.5039623 2.3552935 0.7228502 6.2820617 2.6307185 4.1501565 1.2931251
##  [8] 3.0560797 2.6804478 1.5096613 9.5317328 4.2675976 0.8255186 6.3180853
## [15] 2.8050190 2.5827480 3.7267318 2.3089081 0.9250903 1.9420152
```

```
 (ii) Apply the 'MyMLE' to estimate $\theta$ and append the value in a vector
```

```r
x=MyMLE(20,1.5,2.2)
x
```

```
## [1] 0.4972155
```

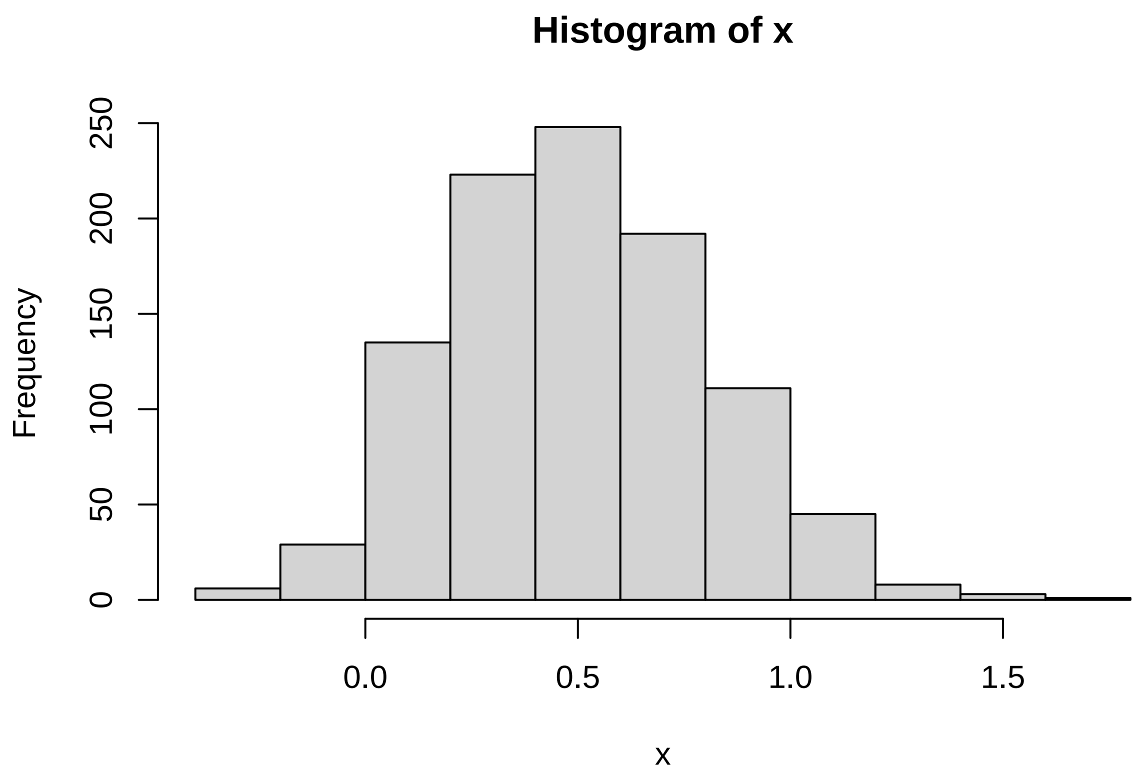
```
 (iii) Repeat the step (i) and (ii) 1000 times
```

```r
for (i in 1:1000){

  x=append(x,MyMLE(20,1.5,2.2))
}
```

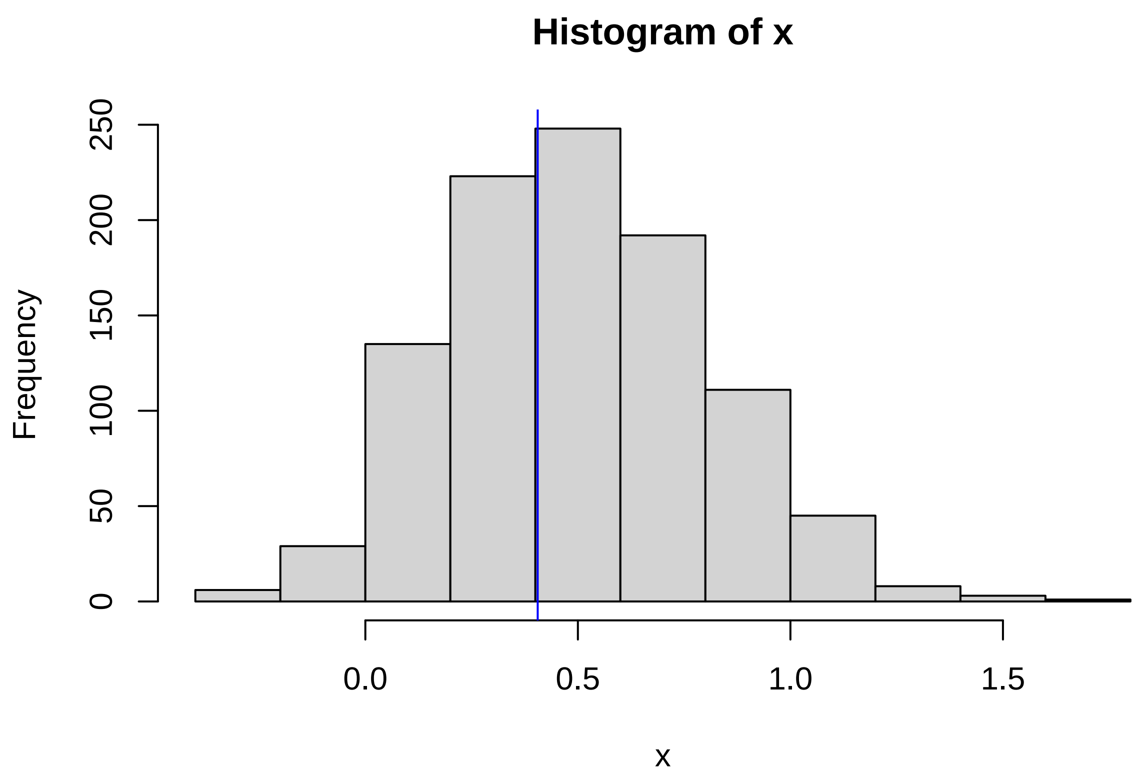
```
 (iv) Draw histogram of the estimated MLEs of $\theta$.
```

```r
hist(x)
```



(v) Draw a vertical line using `abline` function at the true value of θ.

```
hist(x)
abline(v=log(1.5),col="blue")
```




```
(vi) Use ‘quantile‘ function on estimated $\theta$’s to find the 2.5 and 97.5-percentile points.
```

```
y=quantile(x, probs = c(.025, .975))
y
```

```
##        2.5%       97.5%
## -0.03022325  1.11604693
```

3. Choose `n=40`, and `alpha=1.5` and repeat the (2). ##
   (i) Simulate $\{X_1, X_2, \cdots, X_n\}$ from `rgamma(n=20,shape=1.5,scale=2.2)`

```
rgamma(40,1.5,scale=2.2)
```

```
##  [1]  1.95308082  0.71059678 10.52767906  6.28336265  0.06759737  4.74689929
##  [7]  0.83103342  4.46360122  2.78456692  2.89068628  9.06832777  2.05131565
## [13]  6.61713033  3.84470389  1.16844382  6.03470210  2.01534693  1.28636930
## [19]  0.80732757  7.10531945 10.89276542  2.98047991  3.69206418  3.67129580
## [25]  3.90617117  1.02455331  4.88976468  1.01086076  6.47009268  4.53327739
## [31]  1.69020794  2.62091347  2.23908520  1.51160073  1.28305488  2.27124301
## [37]  0.66245713  1.34217774  1.66778439  2.31748871
```

```
(ii) Apply the ‘MyMLE‘ to estimate $\theta$ and append the value in a vector
```

```r
x=MyMLE(40,1.5,2.2)
x
```

```
## [1] 0.2124265
```

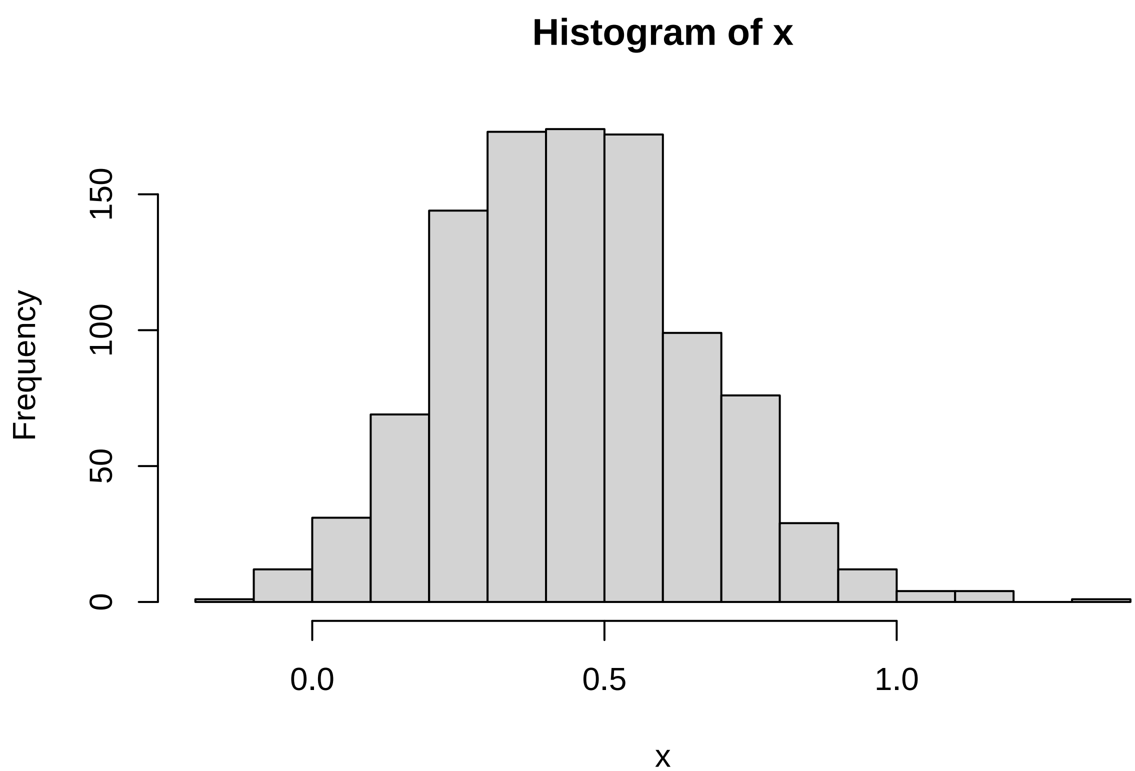
```
(iii) Repeat the step (i) and (ii) 1000 times
```

```r
for (i in 1:1000){

  x=append(x,MyMLE(40,1.5,2.2))
}
```

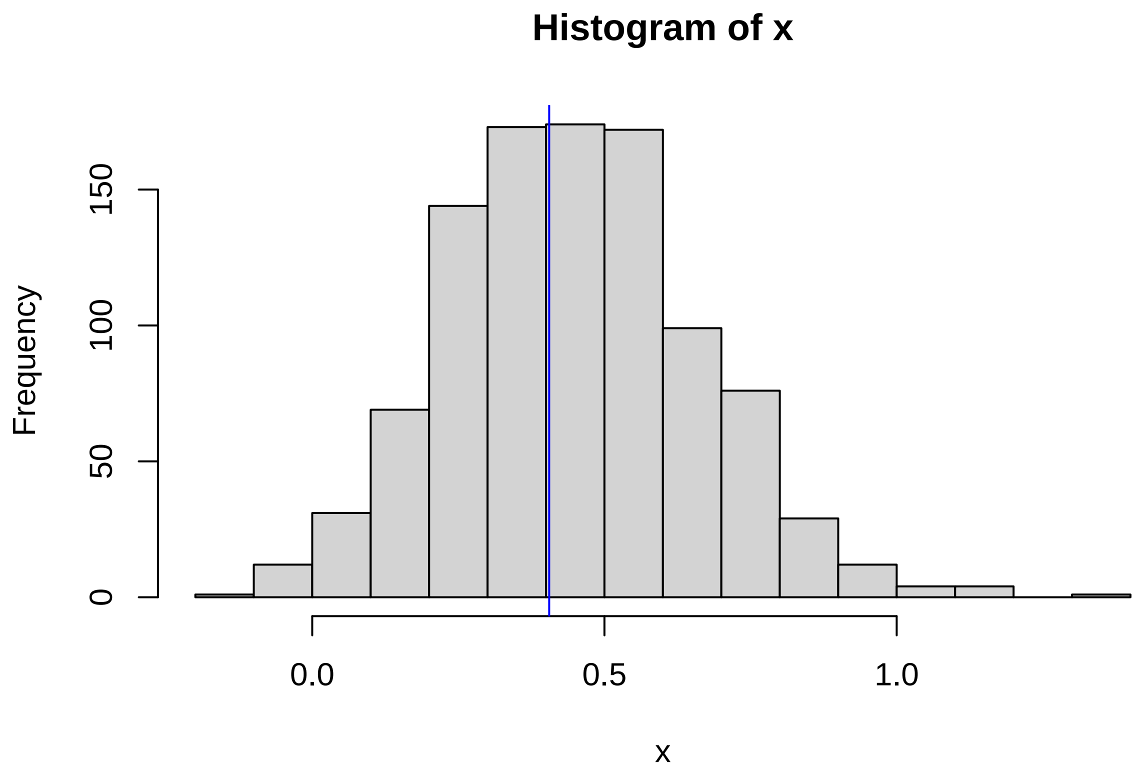
```
(iv) Draw histogram of the estimated MLEs of $\theta$.
```

```r
hist(x)
```



(v) Draw a vertical line using `abline` function at the true value of θ.

```r
hist(x)
abline(v=log(1.5),col="blue")
```




```
(vi) Use `quantile` function on estimated $\theta$'s to find the 2.5 and 97.5-percentile points.
```

```
y=quantile(x, probs = c(.025, .975))
y
```

```
##       2.5%      97.5%
## 0.06568476 0.87839723
```

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

   (i) Simulate $\{X_1, X_2, \cdots, X_n\}$ from `rgamma(n=20,shape=1.5,scale=2.2)`

```
rgamma(100,1.5,scale=2.2)
```

```
##   [1]  0.47904230  3.18628003  0.19746407  0.37177769  0.05669380  0.14950903
##   [7]  4.92295211  1.18732364  2.24504643  0.95439672  1.05533074  6.18637053
##  [13]  1.28631386  1.27850945  6.64782668  2.12446403  0.39125715  3.93416838
##  [19]  0.97822981  2.58142805  2.93101591  5.44391821  0.10654847  2.31991116
##  [25]  3.04636180  0.57166155  2.54378655  1.78426594  1.79696467  8.36589139
##  [31]  3.31675958  5.48766525  8.78646836  0.07694896  2.84497344  0.37261990
##  [37]  3.77664476  2.10356243  4.91401083  1.39610026  2.55437032  1.77393217
##  [43]  0.06734837  0.42161565  2.06216734  1.07763757  5.51264772  1.76578592
```

```
## [49] 15.73131643  4.56018571  3.14129454  4.81432059  5.70955258  2.62995442
## [55]  3.03459788  4.64480789  6.33681591  0.48657577  1.93844675  2.47572727
## [61]  6.40782568  0.89294412  1.19543241  2.83215104  3.83774094  2.12386936
## [67]  1.43928846  2.48669575  2.30868085  0.70426904  0.77429093  3.29571796
## [73]  6.99073734  1.01436370  8.82054487  4.29967126  0.42057551 10.92911407
## [79]  4.18028782  0.51986332  7.03205540  3.17022156  1.74964501  0.09058812
## [85]  2.86315472  4.69137490  3.43084648  4.21713740  8.74369162  4.65439632
## [91]  5.05535755  2.02570756  0.34078353  5.63798573  0.20781502  1.77848155
## [97]  1.72168007  0.98477116  1.19242564  1.42339914
```

```
 (ii) Apply the ‘MyMLE‘ to estimate $\theta$ and append the value in a vector
```

```
x=MyMLE(100,1.5,2.2)
x
```

```
## [1] 0.3281667
```

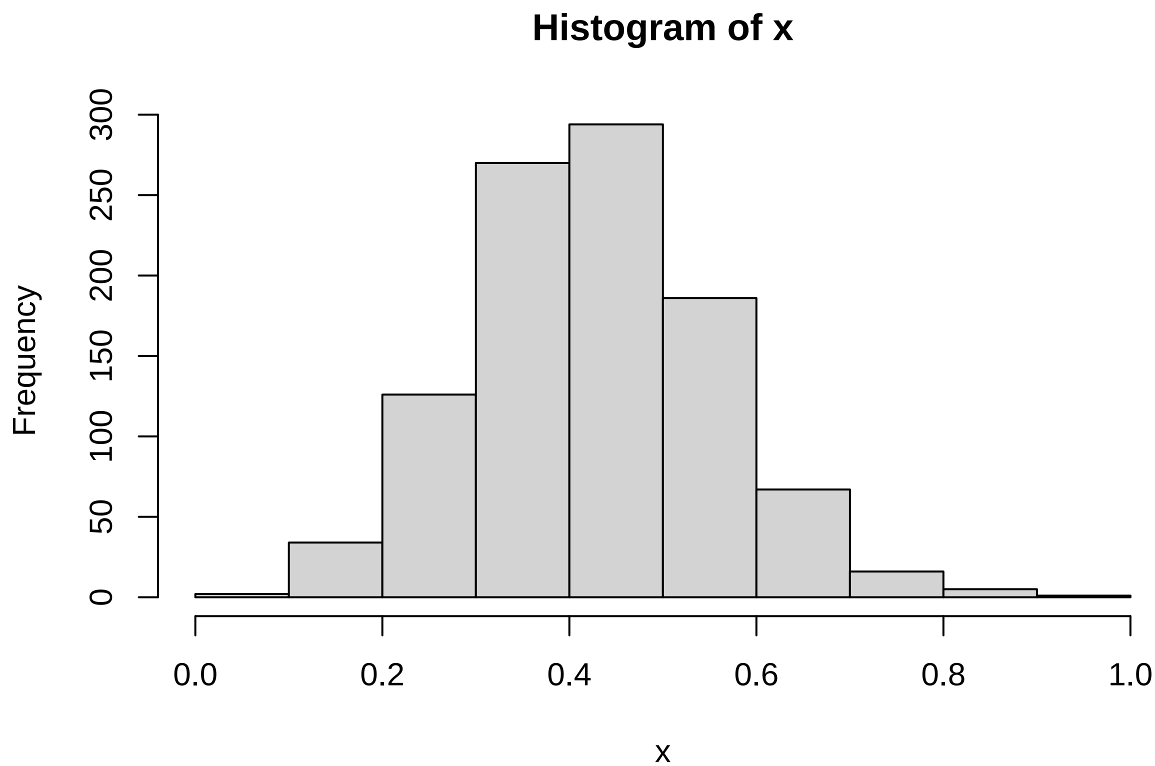
```
 (iii) Repeat the step (i) and (ii) 1000 times
```

```
for (i in 1:1000){

  x=append(x,MyMLE(100,1.5,2.2))
}
```

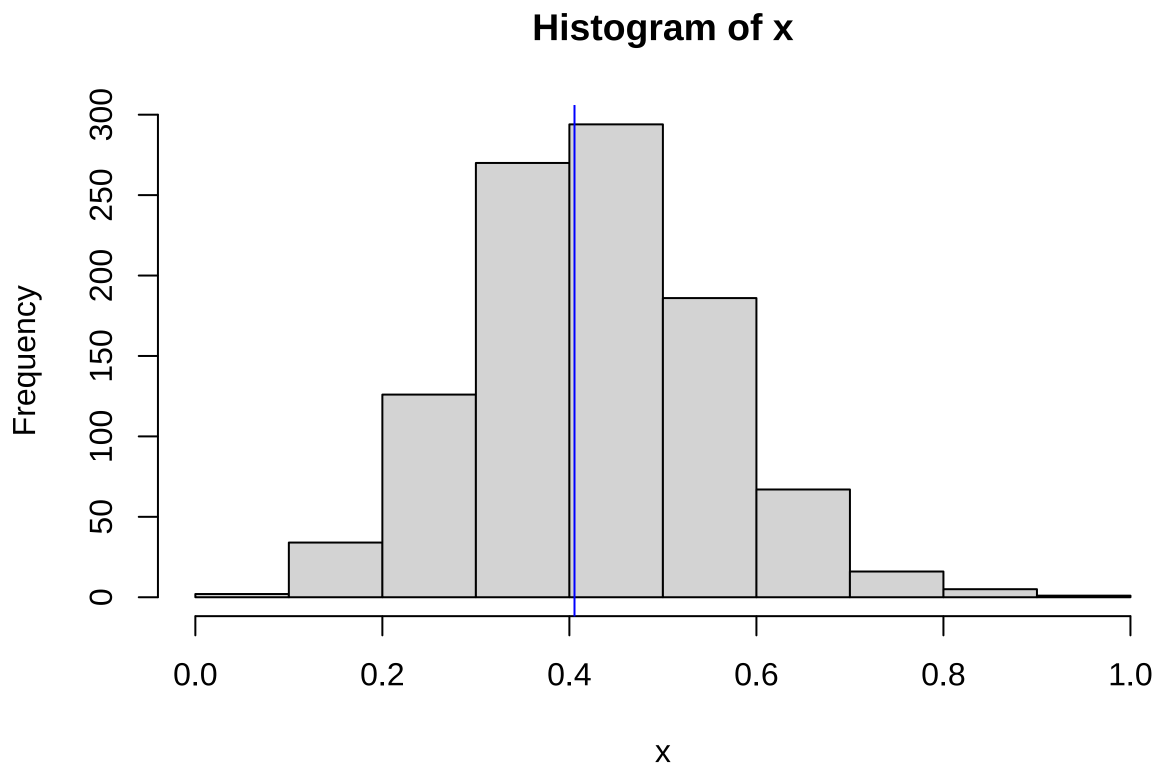
```
 (iv) Draw histogram of the estimated MLEs of $\theta$.
```

```
hist(x)
```



(v) Draw a vertical line using `abline` function at the true value of θ.

```
hist(x)
abline(v=log(1.5),col="blue")
```




```
(vi) Use `quantile` function on estimated $\theta$'s to find the 2.5 and 97.5-percentile points.
```

```r
y=quantile(x, probs = c(.025, .975))
y
```

```
##      2.5%     97.5%
## 0.1852827 0.6890047
```

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

```r
#Yes, It does.
```

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 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.

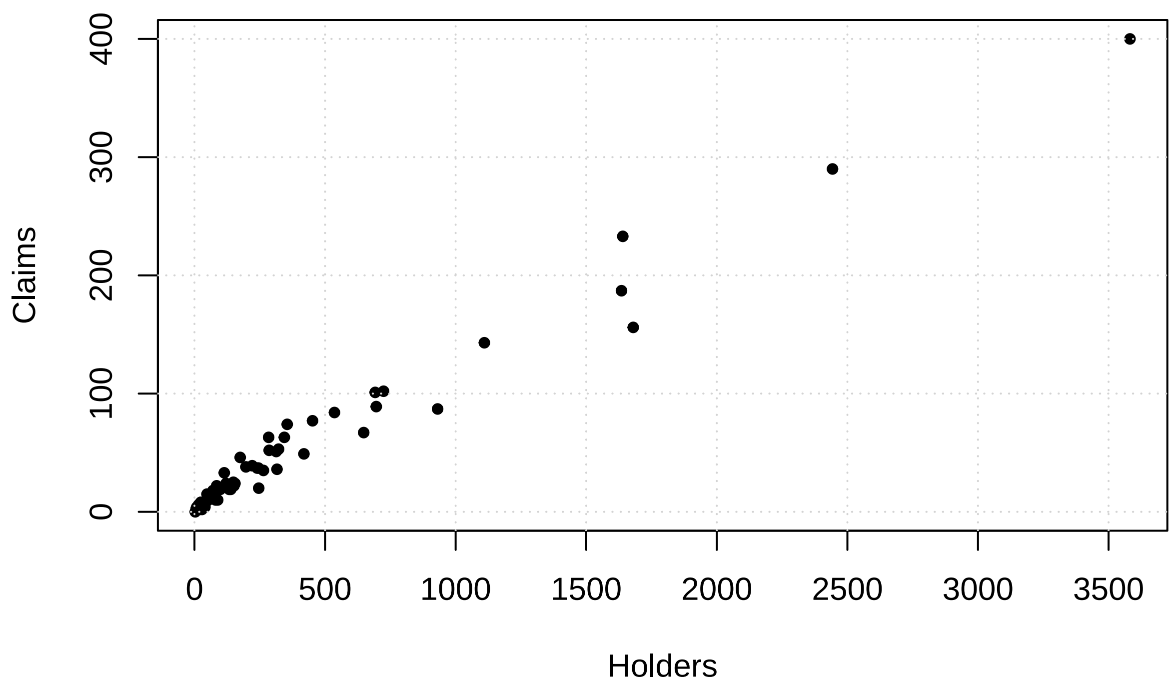
`Group` (an ordered factor): group of car with levels <1 litre, 1–1.5 litre, 1.5–2 litre, >2 litre.

`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)
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 $\theta = (\beta_0, \beta_1, \sigma)$

```
library(SciViews)
library(MASS)
```

```
library(jmuOutlier)
Holders=Insurance$Holders
Claims=Insurance$Claims
data=data.frame(cbind(Claims,Holders))
data=data[-61,]
n=length(Holders)-1

y=data[,1]
x=data[,2]
```

```
Negloglike=function(data,theta)
{
  l=0
  for(i in 1:n)
  {
    l=l+log(dnorm(y[i], theta[1]+theta[2]*x[i],theta[3]))

  }
  return(-l)
}

theta=c(0.1,0.1,50)
fit=optim(theta,Negloglike,data=data)
##Estimated value of theta is:

c(fit$par[1],fit$par[2],fit$par[3])
```

```
## [1]  8.3084803  0.1125138 11.9133879
```

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

```
BIC_A=ln(n)*(length(fit$par))+2*fit$value
#BIC value is:
BIC_A
```

```
## [1] 503.405
```

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 $\theta = (\beta_0, \beta_1, \sigma)$

```
Negloglike=function(data,theta)
{
  l=0
  for(i in 1:n)
  {
    l=l+log(dlaplace(y[i], theta[1]+theta[2]*x[i],theta[3]))

  }
  return(-l)
}

theta=c(0.1,0.1,50)
fit=optim(theta,Negloglike,data=data)
##Estimated value of theta is:

c(fit$par[1],fit$par[2],fit$par[3])
```

```
## [1]  5.2021496  0.1165771 11.6746589
```

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

```
BIC_B=ln(n)*(length(fit$par))+2*fit$value
#BIC value is:
BIC_B
```

```
## [1] 491.7071
```

Part C: We want to fit the following models:

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

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

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 $\theta = (\alpha, \beta, \sigma)$

```
Negloglike=function(data,theta)
{
  l=0
  for(i in 1:n)
  {
    l=l+log(dlnorm(y[i], theta[1]+theta[2]*log(x[i]),theta[3]))

  }
  return(-l)
}
```

```r
theta=c(0.1,0.1,1)
fit=optim(theta,Negloglike,data=data)
##Estimated value of theta is:

c(fit$par[1],fit$par[2],fit$par[3])
```

```
## [1] -1.0243551  0.8479072  0.3293700
```

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

```r
BIC_C=ln(n)*(length(fit$par))+2*fit$value
#BIC value is:
BIC_C
```

```
## [1] 452.6034
```

Part D: We want to fit the following models:

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

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

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

```r
e=2.718281828459045
Negloglike=function(data,theta)
{
  l=0
  for(i in 1:n)
  {
    l=l+log(dgamma(y[i], e^(theta[1]+theta[2]*log(x[i])),theta[3]))

  }
  return(-l)
}

theta=c(0.1,0.1,0.1)
fit=optim(theta,Negloglike,data=data)

##Estimated value of theta is:

c(fit$par[1],fit$par[2],fit$par[3])
```

```
## [1] -1.6430902  0.8371016  0.4858613
```

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

```r
BIC_D=ln(n)*(length(fit$par))+2*fit$value
#BIC value is:
BIC_D
```

```
## [1] 437.3382
```

(iii) Compare the BIC of all three models

```r
c(BIC_A,BIC_B,BIC_C,BIC_D)
```

```
## [1] 503.4050 491.7071 452.6034 437.3382
```