Assignment 2

Sumer Vaid

```r
library("dplyr")
```

```
## Warning: package 'dplyr' was built under R version 3.2.5

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library("ggplot2")
```

```
## Warning: package 'ggplot2' was built under R version 3.2.5
```

```r
library("ggfortify")
```

```
## Warning: package 'ggfortify' was built under R version 3.2.5
```

```r
library("bbmle")
```

```
## Warning: package 'bbmle' was built under R version 3.2.5

## Loading required package: stats4

##
## Attaching package: 'bbmle'

## The following object is masked from 'package:dplyr':
##
##     slice
```

```r
library("stats4")
library("lmtest")
```

```
## Warning: package 'lmtest' was built under R version 3.2.5

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 3.2.5

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library("extRemes")
```

```
## Warning: package 'extRemes' was built under R version 3.2.5

## Loading required package: Lmoments
```

```
## Loading required package: distillery

## Warning: package 'distillery' was built under R version 3.2.5

## Loading required package: car

## Warning: package 'car' was built under R version 3.2.5

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode

##
## Attaching package: 'extRemes'

## The following objects are masked from 'package:stats':
##
##     qqnorm, qqplot
```

```r
data<-data.frame(read.table("incomes.txt"))
```

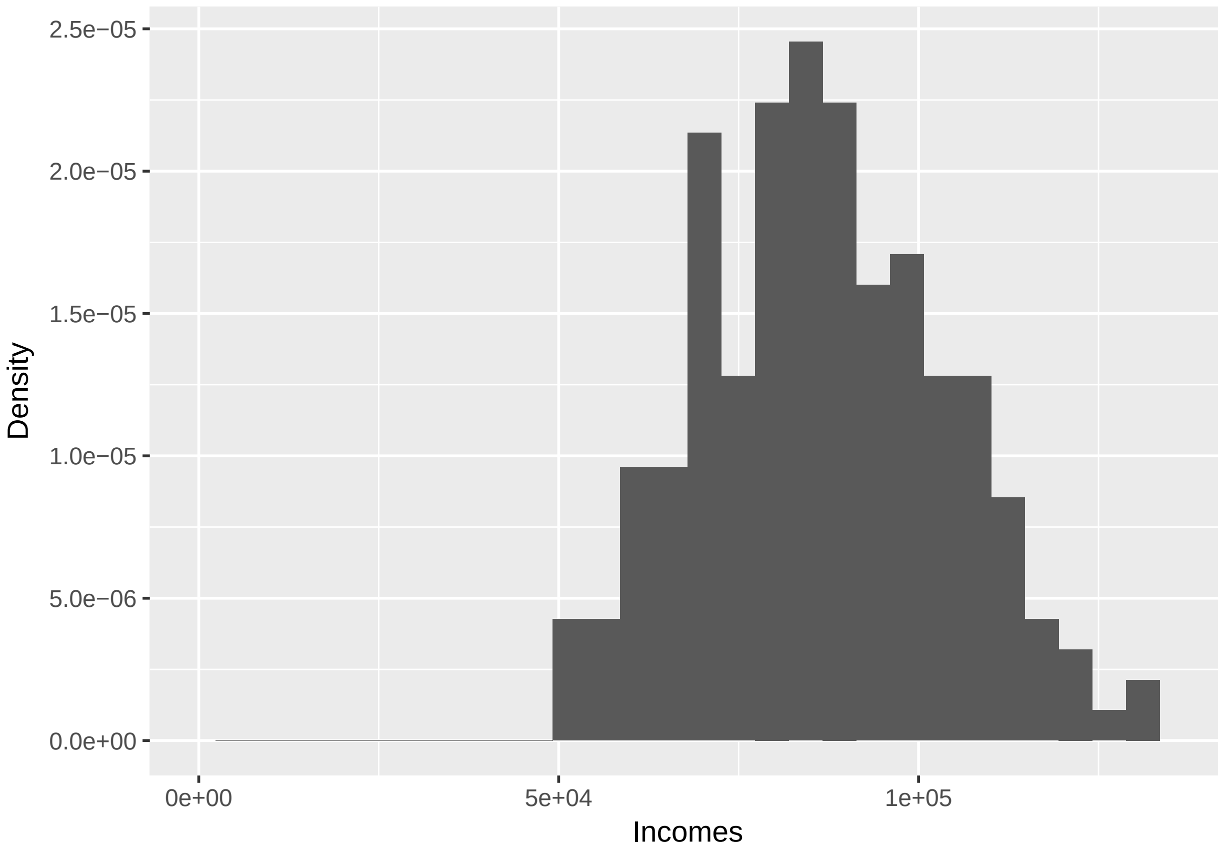
First, we define the function that takes in the parameters and returns the log likelihood value below:

```r
ll_pdf<-function(vector){
  pdf_vals=dlnorm(data$V1, meanlog = vector[1], sdlog = vector[2])
  log_pdf_vals=log(pdf_vals)
  log.lik.val=sum(log_pdf_vals)
  return(log.lik.val)
}
```

Now we plot the requisite histogram with 30 bins below:(the default is 30 bins)

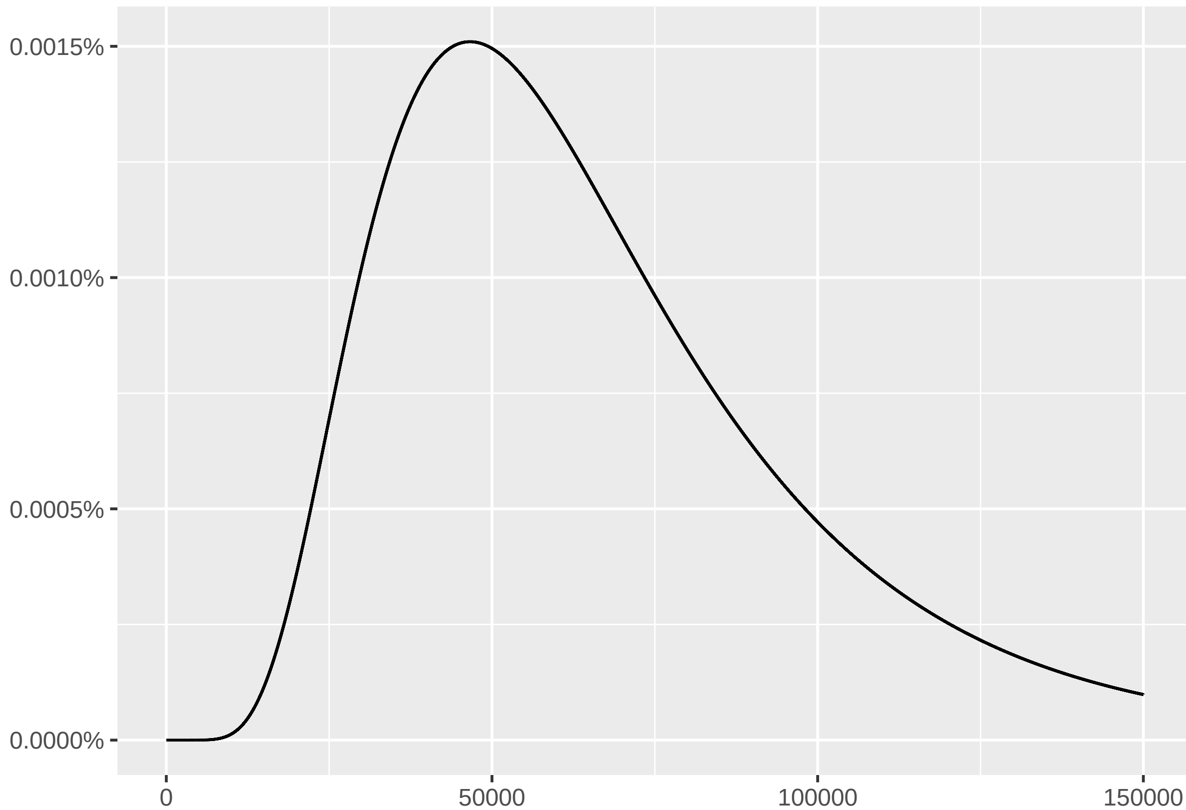
```r
Incomes_Hist=ggplot(data, aes(x=V1))+geom_histogram(aes(y=..density..))+xlab("Incomes")+
  xlim(0, max(data$V1))+ylab("Density")
print(Incomes_Hist)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Now we plot the lognormal pdf with the provided parameters:

```
lnpdf<-ggdistribution(dlnorm, seq(0, 150000), mean = 11, sd = 0.5)
print(lnpdf)
```




```
#plotting the density distribution with requisite parameters
```

Below, we input the parameters into the defined function to yield the log likelihood.

```
vector<-c(11,0.5) #creating a vector of mean and SD  to input into pdf function
ll_pdf(vector) #inputting the mean and SD values into the custom function
```

```
## [1] -2385.857
```

Below, we use the minimizer to find the set of parameters that maximize the log likelihood value:

```
fit1<-optim(c(11,0.5), fn=ll_pdf, method="BFGS", control=list(fnscale=-1), hessian = TRUE)
```

```
## Warning in dlnorm(data$V1, meanlog = vector[1], sdlog = vector[2]): NaNs
## produced

## Warning in dlnorm(data$V1, meanlog = vector[1], sdlog = vector[2]): NaNs
## produced

## Warning in dlnorm(data$V1, meanlog = vector[1], sdlog = vector[2]): NaNs
## produced

## Warning in dlnorm(data$V1, meanlog = vector[1], sdlog = vector[2]): NaNs
## produced

## Warning in dlnorm(data$V1, meanlog = vector[1], sdlog = vector[2]): NaNs
## produced

## Warning in dlnorm(data$V1, meanlog = vector[1], sdlog = vector[2]): NaNs
## produced
```

```
## Warning in dlnorm(data$V1, meanlog = vector[1], sdlog = vector[2]): NaNs
## produced

## Warning in dlnorm(data$V1, meanlog = vector[1], sdlog = vector[2]): NaNs
## produced

## Warning in dlnorm(data$V1, meanlog = vector[1], sdlog = vector[2]): NaNs
## produced

## Warning in dlnorm(data$V1, meanlog = vector[1], sdlog = vector[2]): NaNs
## produced

## Warning in dlnorm(data$V1, meanlog = vector[1], sdlog = vector[2]): NaNs
## produced

## Warning in dlnorm(data$V1, meanlog = vector[1], sdlog = vector[2]): NaNs
## produced
```

```
inv_cov<-fit1$hessian
covariance_matrix<-solve(inv_cov)*matrix(c(1,-1, 1,-1), nrow=2, ncol=2)
print(fit1)
```

```
## $par
## [1] 11.3590229  0.2081824
##
## $value
## [1] -2241.719
##
## $counts
## function gradient
##       51       10
##
## $convergence
## [1] 0
##
## $message
## NULL
##
## $hessian
##               [,1]          [,2]
## [1,] -4.614683e+03 -4.399453e-03
## [2,] -4.399453e-03 -9.230605e+03
```

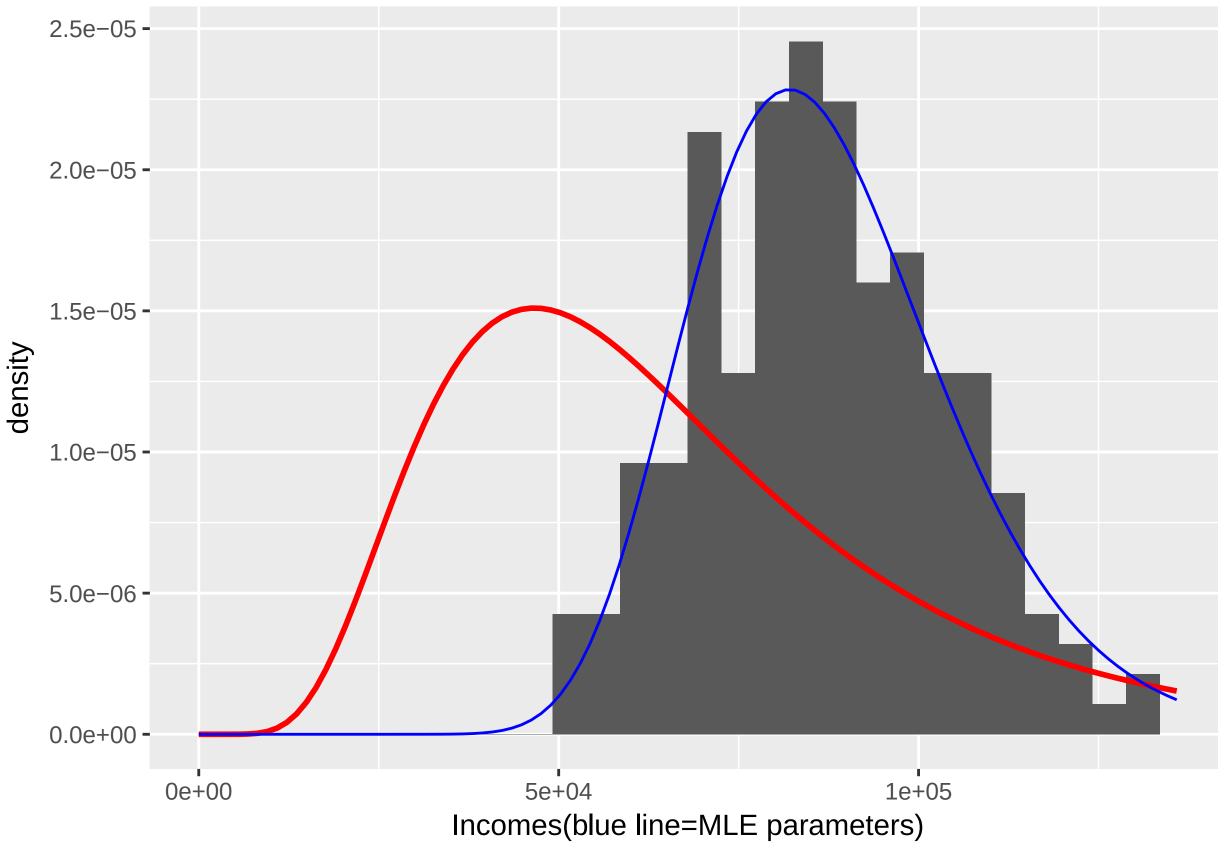
```
vector2<-c(11.3590229, 0.2081824)
ll_pdf(vector2) #calculate log likelihood value for most optimal parameters
```

```
## [1] -2241.719
```

Now we plot the updated histogram overlayed with both the probability density functions.

```
Incomes_Hist_lines=ggplot(data,aes(x=V1))+geom_histogram(aes(y=..density..))+xlab("Incomes(blue line=ML
print(Incomes_Hist_lines)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Below, we calculate the likelihood ratio test for the two models and find the probability values for the requisite incomes.

```
MLEvals<-c(11.35, 0.208)
model1<-ll_pdf(vector)
model2<-ll_pdf(MLEvals)
lr.test(-2241.9076275, -2385.856997)
```

```
##
##  Likelihood-ratio Test
##
## data:  -2241.9076275-2385.856997
## Likelihood-ratio = 287.9, chi-square critical value = 3.8415,
## alpha = 0.0500, Degrees of Freedom = 1.0000, p-value < 2.2e-16
## alternative hypothesis: greater
```

```
plnorm(100000, mean=11.359, sd=0.2081824, lower.tail = FALSE)
```

```
## [1] 0.2298388
```

```
plnorm(75000, mean=11.359, sd=0.2081824, lower.tail=TRUE)
```

```
## [1] 0.2602752
```