Assignment_4–MEM_MLE.R

lenovo

Fri Oct 09 23:44:49 2015

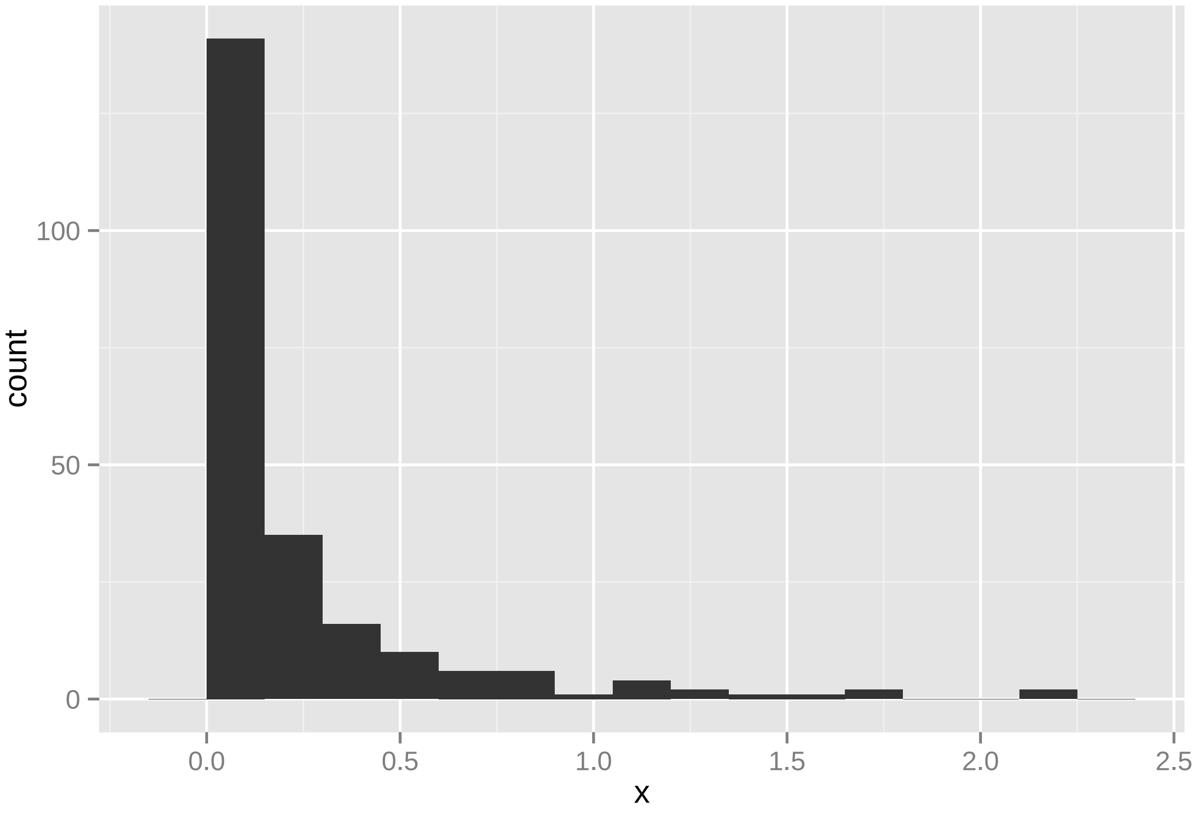
```
library(ggplot2)
data <- read.csv("illinois rain 1960-1964.csv",header=FALSE)
data1 <- unlist(data)
head(data1)
```

```
##   V11   V12   V13   V14   V15   V16
## 0.020 0.001 0.080 1.720 0.490 0.020
```

```
tail(data1)
```

```
## V1214 V1215 V1216 V1217 V1218 V1219
## 0.170 0.090 1.040 0.003 0.030    NA
```

```
data1 <- data.frame(data1[1:227])
colnames(data1) <- "x"
#histogram of illinois data
qplot(x, data=data1, geom = "histogram",binwidth=.15)
```



```
##  Now using the MGF for gamma or be simply looking it up
##  use the following facts about the gamma function
##
##  firt moment = m1 = (alpha/lambda)

##  second moment = m2 = m1^2 + (m1/lambda)

##  from with you get equations for alpha and labda in terms of the moments

##  lambda = m1 / (m2 - m1^2)     note that (m2 - m1^2) = variance(x)

##  alpha = (m1^2)/(m2 - mx^2)

##  Now use the sample statistics X-bar and S-squared to estimate lambda and alpha

## lambda-hat = X-bar/S-squre

## alpha-hat =  (X-bar)^2 / S-square

##  So here are the calculations:

mean(data1$x)
```

```
## [1] 0.2243921
```

```
var(data1$x)
```

```
## [1] 0.1338252
```

```
alpha <- mean(data1$x)^2/var(data1$x)
alpha
```

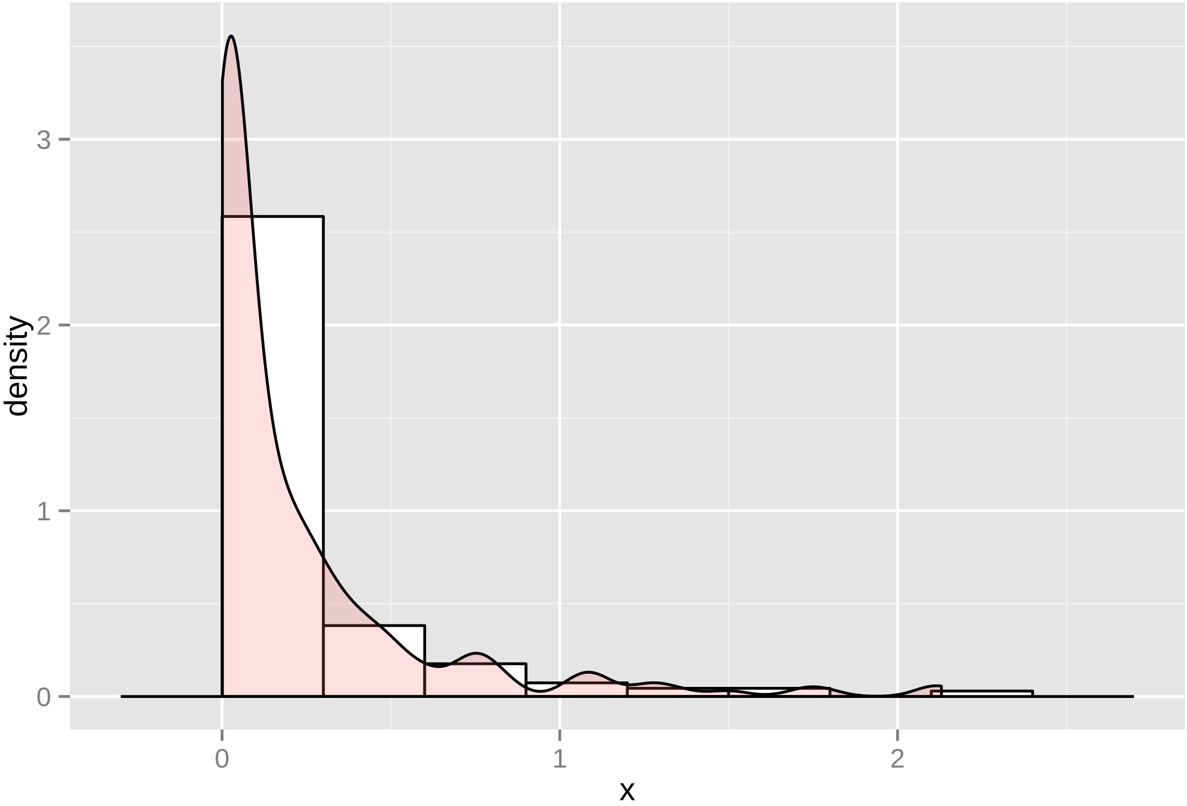
```
## [1] 0.3762506
```

```
lambda <- mean(data1$x)/var(data1$x)
lambda
```

```
## [1] 1.676755
```

```
# Homework #1-----------------------------------------------------------------------
# 1.Make a plot ths superimposes the gamma density with the alpha and lambda as above
# on the histogram of the data.

ggplot(data1,aes(x))+
  geom_histogram(aes(y=..density..),binwidth=.3,colour="black",fill="white")+
  geom_density(alpha=.2,fill="#FF6666")
```



```
# Homework #2----------------------------------------------------------------------
# 1.bootstrap -- samples (n=227) from gamma(alpha, lambda)
# to find the variance for the estimates of alpha and lambda

B <- 1000
tBoot.lambda <- rep(0,B)
tBoot.alpha <- rep(0,B)
for(i in 1:B){
  x.s1 <- sample(data1$x, length(data1$x), replace = TRUE)
  tBoot.lambda[i] <- mean(x.s1)/var(x.s1)
  tBoot.alpha[i] <- (mean(x.s1))^2/var(x.s1)
}
#get variance of estimated almbda and alpha
v.lambda = var(tBoot.lambda)
v.lambda
```

```
## [1] 0.07134382
```

```
v.alpha = var(tBoot.alpha)
v.alpha
```

```
## [1] 0.002292904
```

```r
#get standard error of bootstrap parametrics
se.lambda = sqrt(v.lambda)
se.lambda
```

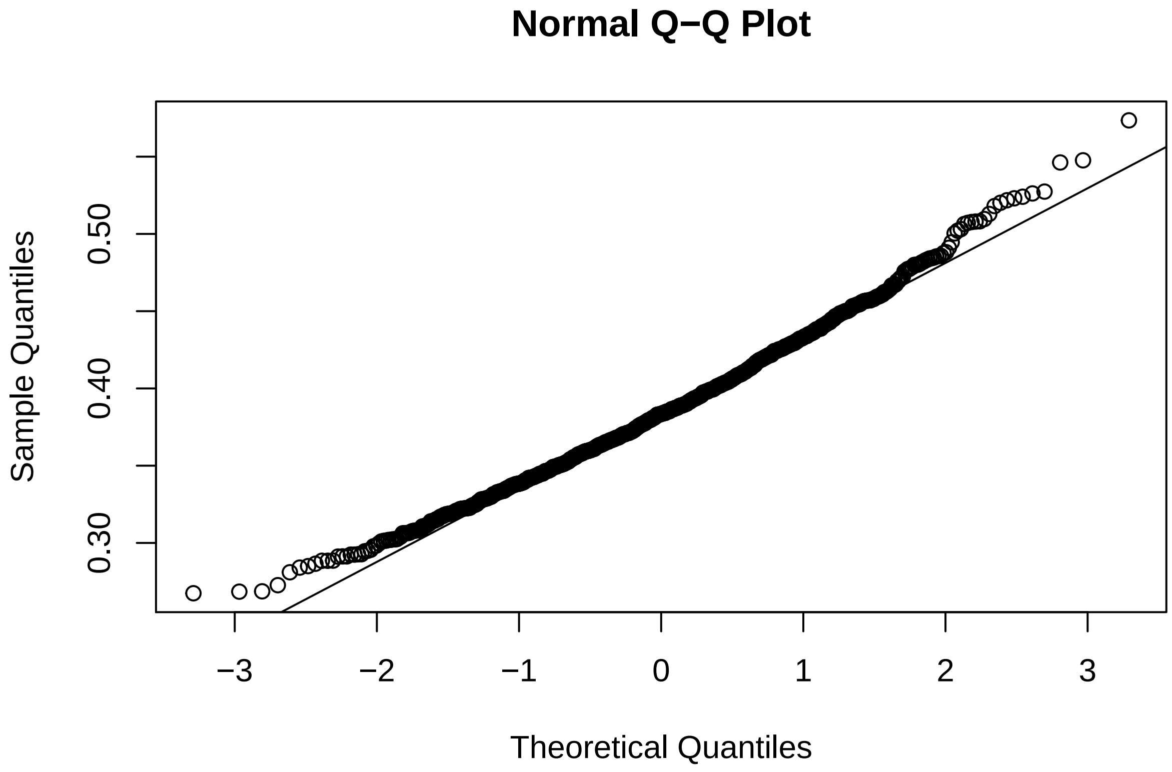
```
## [1] 0.2671026
```

```r
se.alpha = sqrt(v.alpha)
se.alpha
```

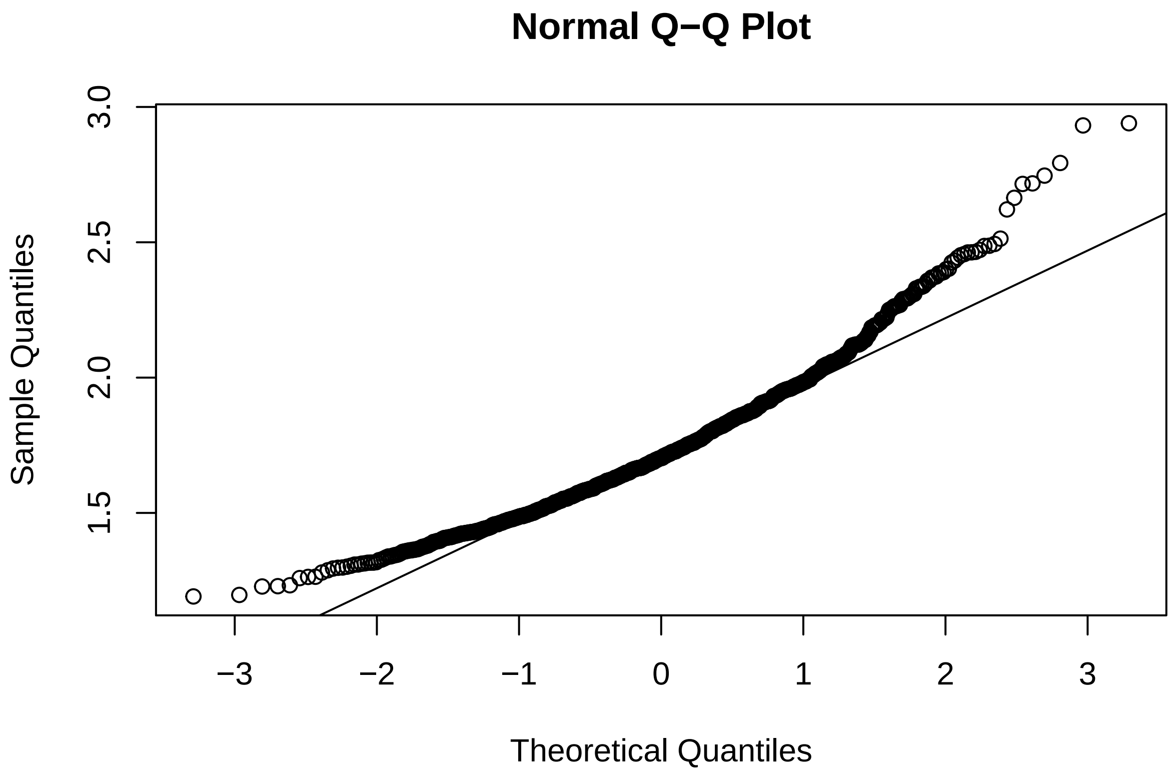
```
## [1] 0.04788427
```

```r
# 2.State confidence for your estimates(all three methods of bootstrap confidence interval)
#get 95% confidence interval of lambda and alpha
Normal.lambda <- c(lambda-2*se.lambda,lambda+2*se.lambda)
Normal.alpha <- c(alpha-2*se.alpha,alpha+2*se.alpha)
Percentile.lambda <- c(quantile(tBoot.lambda,0.025),quantile(tBoot.lambda,0.975))
Percentile.alpha <- c(quantile(tBoot.alpha,0.025),quantile(tBoot.alpha,0.975))
Pivotal.lambda <- c(2*lambda-quantile(tBoot.lambda,0.95),2*lambda-quantile(tBoot.lambda,0.05))
Pivotal.alpha <- c(2*alpha-quantile(tBoot.alpha,0.95),2*alpha-quantile(tBoot.alpha,0.05))

# 3.State why you picked the estimator you used for the confidence interval.
# Only when lambda and alpha are normal distribution, the normal CI is reasonable
# So check whether they are normal using qqplot and ks.test
qqnorm(tBoot.alpha)
qqline(tBoot.alpha)
```



```
qqnorm(tBoot.lambda)
qqline(tBoot.lambda)
```



```
ks.test(tBoot.alpha,"pnorm")
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  tBoot.alpha
## D = 0.60681, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
ks.test(tBoot.lambda,"pnorm")
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  tBoot.lambda
## D = 0.89225, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
#According to the qqplot and ks.test, I find the distributions of lamda and alpha
#are different from normal distribution respectively.
#So I choose to use pivotal confidence interval not normal confidence interval
cat("Pivotal CI of lamda (",Pivotal.lambda[1],",",Pivotal.lambda[2], ")\n")
```

```
## Pivotal CI of lamda ( 1.09033 , 1.973736 )
```

```
cat("Pivotal CI of alpha (",Pivotal.alpha[1],",",Pivotal.alpha[2], ")\n")
```

```
## Pivotal CI of alpha ( 0.2856305 , 0.4404579 )
```

```
# Homework #3-----------------------------------------------------------------
# 3.1 Justify the minus.likelihood fuction used above.  Note the use of "lgamma."
x1 <- data1$x

n <- length(data1$x)

minus.likelihood <- function(theta) {-(n*theta[1]*log(theta[2])-n*lgamma(theta[1])+(theta[1]-1)*sum(log

max.likelihood <- nlminb(start=c(.3762, 1.6767), obj = minus.likelihood)

max.likelihood$par
```

```
## [1] 0.4407914 1.9643791
```

```
alpha1 <- max.likelihood$par[1]
alpha1
```

```
## [1] 0.4407914
```

```
lambda1 <- max.likelihood$par[2]
lambda1
```

```
## [1] 1.964379
```

```
#lgamma return to the natural logarithm of the absulute value of the gamma function G(alpha)
#minus.likehood is the minus log-likelihood function, so we get the minimum of minus.likehood by
#nlminb function whose value is just the MLE of alpha and lambda


# 3.2 bootstrap to get standard errors for alpha and lambda
# and produce an extimated confidence interval
options(warn=-1)
MEM.alpha <- NULL
MEM.lambda <- NULL
MLE.alpha <- rep(0,B)
MLE.lambda <- rep(0,B)
for(i in 1:B){
  x.s2 <- sample(x1,length(x1),replace=TRUE)
  min.likelihood <- function(theta)
  {-(length(x.s2)*theta[1]*log(theta[2])-length(x.s2)*lgamma(theta[1])+
       (theta[1]-1)*sum(log(x.s2))-theta[2]*sum(x.s2))}
  MEM.alpha <- (mean(x.s2))^2/var(x.s2)
  MEM.lambda <- mean(x.s2)/var(x.s2)
  max.likelihood <- nlminb(start=c(MEM.alpha,MEM.lambda), obj = min.likelihood)
  MLE.alpha[i] <- max.likelihood$par[1]
  MLE.lambda[i] <- max.likelihood$par[2]
}

se.MLE.alpha <- sqrt(var(MLE.alpha))
se.MLE.alpha
```

```
## [1] 1.914439
```

```
se.MLE.lambda <- sqrt(var(MLE.lambda))
se.MLE.lambda
```

```
## [1] 0.5649215
```

```
#Pivotal confidence interval
Pivotal.lambda1 <- c(2*lambda1-quantile(MLE.lambda,0.95),2*lambda1-quantile(MLE.lambda,0.05))
Pivotal.alpha1 <- c(2*alpha1-quantile(MLE.alpha,0.95),2*alpha1-quantile(MLE.alpha,0.05))

cat("Pivotal CI of lamda (",Pivotal.lambda1[1],",",Pivotal.lambda1[2], ")\n")
```

```
## Pivotal CI of lamda ( 1.467049 , 3.928758 )
```

```
cat("Pivotal CI of alpha (",Pivotal.alpha1[1],",",Pivotal.alpha1[2], ")\n")
```

```
## Pivotal CI of alpha ( 0.3946239 , 5.790476 )
```

```
# 3.3 Use this case to build a an illustrated guide to this kind of estimation.
# The homework assignments will be part of this guide, but go beyond that to
# make a resource for yourself.

#For MEM
#Given a sample is a parametric distribution, it's easy to get its MEM using MGF for the distribution;
#And then bootstraping can assist us to find variance and confidence interval of parametrics
#For MLE
#First I need to get the likehood or log-likehood when the likehood function is hard to get derivative.
#Then nlminb function can assist me to get the MLE, which is the par result. But the obj argument in nl
#should be the minus likehood or log-likehood
```