mem_to_max_like_1_.R

jiayuan

Mon Oct 5 22:23:51 2015

```
##  This file summarizes the presentation demonstrating the use of
##  method of moment estimations as entry points for maximum likelihood estimation
##  when calculation of MLE's cannot be performed analytically.
##
##  The data used for this example is from Rainfall Evaluation Studies carried out
##  in the 1960's at the Univeristy of Illinois by Floyd Huff.
##
##  The data are rain gauge readings taken after rainstorms in the years 1960 through 1964
##  in selected areas in central and souther Illinois.
##
##  The accumulated rainfall readings are assumed to be gamma distributed.
##
##  In the classroom presentation, the MLE estimation for the parameeters of the gamma
##  distribution are shown to yield a nonlinear equation that cannot be solved explicitly.
##  This is a well-known result.  So, the MLE estimation is carried out using
##  a numerical estimation that is seeded with initial estimates obtained by using
##  the method of moments.
##
##  This famous case provides you excellent practice for the method of moments
##  estimation and maximum likelihood methods we hav discussed -- and it shows you
##  an iterative method for estimating MLE's when analysis is unavailable.
##


##  Start with the data -- be sure to examine the variables as you go through this
##  script, so that you know what has been done and how.

library(ggplot2)
setwd("/Users/jiayuan/Documents/MA681/hw4")
data <- read.csv("illinois rain 1960-1964.csv",header=F)
data1 <- unlist(data)
head(data1)
```

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

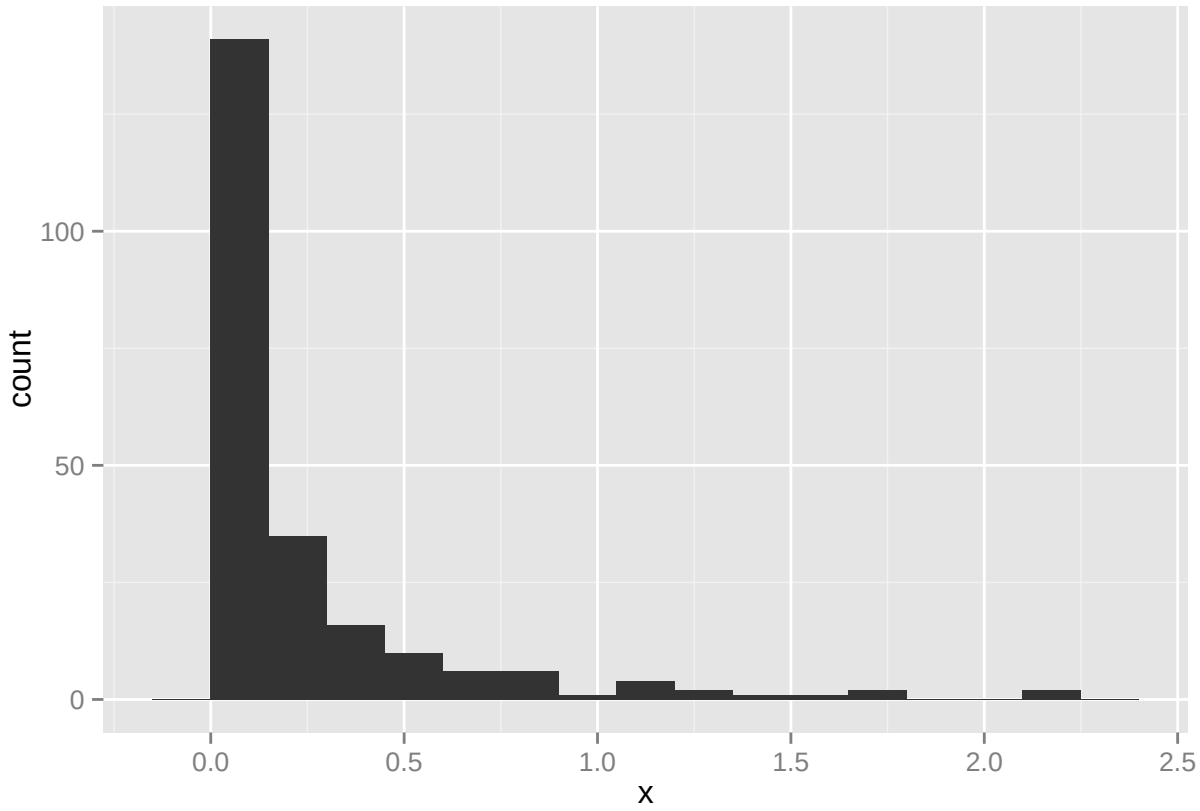
```
tail(data1)
```

```
## 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"
```

```
qplot(x, data=data1, geom = "histogram",binwidth=.15)
```




```
##  Now using the MGF for gamme 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
```

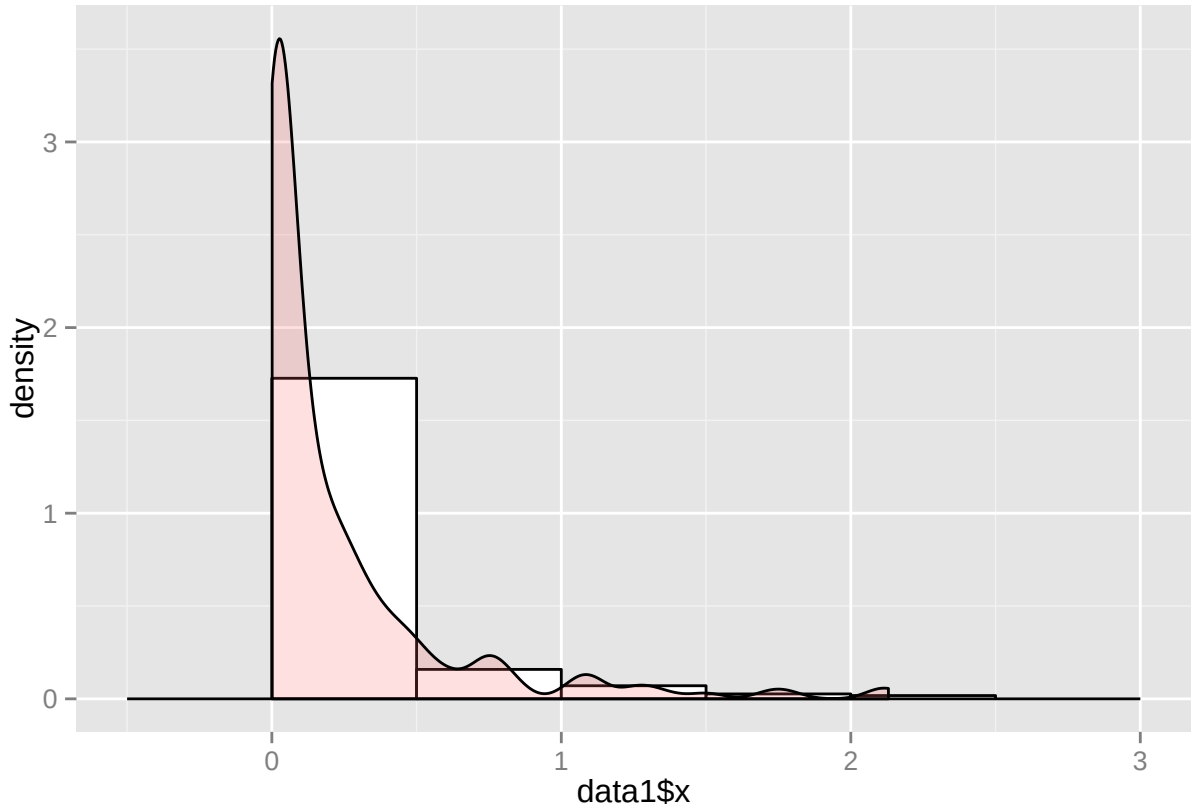
```r
var(data1$x)
```

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

```r
alpha <- mean(data1$x)^2/var(data1$x)  # 0.376
lambda <- mean(data1$x)/var(data1$x)   # 1.677

# Homework #1:  Now make a plot ths superimposes the gamma density with the alpha and lambda as above
# on the histogram of the data.
#
# see http://www.cookbook-r.com/Graphs/Plotting_distributions_(ggplot2)/
# for instructions on how to plot

## Of course, you now want to know how close these estimates are so ...
ggplot(data1, aes(x=data1$x)) +
  geom_histogram(aes(y=..density..),      # Histogram with density instead of count on y-axis
                 binwidth=.5,
                 colour="black", fill="white") +
  geom_density(alpha=.2, fill="#FF6666")  # Overlay with transparent density plot
```




```r
##
## homework # 2
## bootstrap -- samples (n=227) from gamma(alpha, lambda)
## to find the variance for the estimates of alpha and lambda
## state confidence for your estimates.  State why you picked
## the estimator you used for the confidence interval.
```

```
##

n<-227
x<-rgamma(n, shape=alpha, rate = lambda)
x
```

```
##   [1] 1.868998e-02 5.583982e-05 2.019245e-02 3.599832e-06 2.301168e-01
##   [6] 1.792122e-03 1.563377e-01 8.140655e-05 1.257638e-01 4.184654e-01
##  [11] 4.641941e-01 4.258303e-02 5.521535e-01 2.647963e-01 5.297374e-01
##  [16] 6.773416e-02 1.949391e-01 2.088204e-01 2.051589e-02 7.845532e-05
##  [21] 2.716998e-01 6.753466e-02 1.764814e-02 2.723396e-02 1.209074e-01
##  [26] 5.100112e-04 1.206931e-01 3.095652e-03 2.678480e-01 4.829613e-01
##  [31] 4.374958e-01 4.878158e-02 2.866237e-01 3.552702e-02 1.419602e-02
##  [36] 6.904865e-01 2.445875e-01 6.398321e-01 2.522223e-02 3.255857e-02
##  [41] 5.464708e-06 7.102322e-05 1.344569e-01 3.722802e-01 2.041533e-02
##  [46] 1.533934e-02 6.279060e-03 3.694738e-05 8.828874e-02 3.155330e-02
##  [51] 8.118379e-01 6.190647e-01 4.608757e-01 2.376493e-01 5.359394e-03
##  [56] 6.535589e-03 1.517929e-01 4.621346e-06 3.857484e-02 1.079653e-01
##  [61] 6.390844e-03 1.286438e-01 4.540335e-02 6.746674e-02 7.687799e-02
##  [66] 4.880352e-01 5.796232e-03 5.024345e-01 1.133017e-01 4.740285e-03
##  [71] 2.468844e-03 1.422642e-03 1.095789e-01 2.642885e-02 1.925140e-01
##  [76] 1.608113e-02 4.278055e-03 6.236853e-02 4.806509e-01 6.167961e-07
##  [81] 5.853045e-03 5.723528e-02 1.523581e-02 9.929616e-02 1.105898e-01
##  [86] 1.726233e-04 8.136452e-01 1.270878e-02 9.570706e-03 5.470549e-02
##  [91] 1.625521e-01 9.051510e-03 1.912798e-03 6.205272e-02 2.211858e-04
##  [96] 5.965131e-06 4.631461e-03 1.002523e-01 8.426241e-02 6.495868e-03
## [101] 4.020996e-04 5.406842e-01 4.388874e-04 1.263847e-01 2.954529e-01
## [106] 1.243523e-02 2.516577e-01 2.235946e-01 9.398120e-01 4.614840e-02
## [111] 4.516364e-04 4.164457e-01 4.092049e-05 3.585159e-01 2.591646e-03
## [116] 3.950141e-01 9.270686e-01 2.316257e-01 1.066623e-02 2.224942e-03
## [121] 3.067316e-02 3.441488e-03 1.412237e-02 9.661920e-03 2.592211e-01
## [126] 9.657892e-02 9.619188e-03 7.480337e-02 7.713578e-01 4.669843e-01
## [131] 1.030485e-01 3.973274e-01 3.161942e-01 2.642458e-02 1.099508e-01
## [136] 6.092716e-02 9.993185e-02 3.873970e-01 6.034370e-01 3.105656e-03
## [141] 1.086105e-02 3.790621e-02 1.214782e-01 1.519974e-01 3.286201e-01
## [146] 9.886561e-02 1.098813e+00 3.107920e-03 8.578076e-01 2.595416e-02
## [151] 2.207822e-03 6.131913e-02 6.803741e-02 1.033981e-02 2.820250e-02
## [156] 2.657830e-02 2.408119e-01 5.568536e-05 1.436224e-01 2.419261e-02
## [161] 9.572892e-03 1.659461e-01 3.009243e-01 3.281279e-02 1.357022e-01
## [166] 9.432915e-02 1.909488e-03 8.080197e-03 4.105839e-02 1.430741e-01
## [171] 4.322805e-01 7.174027e-02 5.580861e-02 3.012570e-02 1.417652e-02
## [176] 3.838551e-03 4.118306e-04 5.469218e-02 8.711896e-01 2.555011e-01
## [181] 2.686349e-01 5.255158e-01 1.573098e+00 7.029272e-03 3.373786e-02
## [186] 1.669477e-01 6.099999e-01 5.512624e-01 7.853928e-02 2.782849e-01
## [191] 2.008209e-04 1.996621e-01 6.675964e-01 1.106569e-02 7.604630e-02
## [196] 2.785193e-04 3.365072e-02 3.111456e-01 3.342294e-01 4.589420e-01
## [201] 1.854376e-01 1.432594e-01 1.104066e-01 2.460899e-03 1.549517e-02
## [206] 1.272837e-03 2.002857e-04 4.612806e-01 2.160885e-02 1.417048e-02
## [211] 2.902544e-01 8.794282e-02 2.058976e+00 1.126119e+00 9.577013e-04
## [216] 9.533894e-01 8.043226e-03 1.588364e-03 2.430593e-01 5.467790e-02
## [221] 3.470512e-02 3.072332e-01 4.406292e-02 1.821346e+00 5.892518e-01
## [226] 3.166391e-03 7.794364e-01
```

```r
mu.hat<-mean(x)
sigma2.hat<-var(x)
t.alpha<-mu.hat^2/sigma2.hat
t.lambda<-mu.hat/sigma2.hat
B <- 1000
tboot.alpha <- rep(0,B)
tboot.lambda <- rep(0,B)
for(i in 1:B){
  x.s <- sample(x, n, replace=TRUE)
  tboot.alpha[i]<-mean(x.s)^2/var(x.s)
  tboot.lambda[i]<-mean(x.s)/var(x.s)
}
var(tboot.alpha)
```

```
## [1] 0.003899207
```

```r
var(tboot.lambda)
```

```
## [1] 0.1519053
```

```r
se.alpha <- sqrt(var(tboot.alpha))
se.lambda <- sqrt(var(tboot.lambda))
se.alpha
```

```
## [1] 0.06244363
```

```r
se.lambda
```

```
## [1] 0.3897504
```

```r
qnorm(.025,0,1) #-1.96
```

```
## [1] -1.959964
```

```r
qnorm(.975,0,1) #1.96
```

```
## [1] 1.959964
```

```r
Normal.alpha <- c(t.alpha - 1.96*se.alpha, t.alpha + 1.96*se.alpha)
Percentile.alpha <- c(quantile(tboot.alpha,.025),quantile(tboot.alpha,.975))
pivotal.alpha <- c((2*t.alpha - quantile(tboot.alpha, .975)),(2*t.alpha - quantile(tboot.alpha, .025)))

Normal.lambda <- c(t.lambda - 1.96*se.lambda, t.lambda + 1.96*se.lambda)
Percentile.lambda <- c(quantile(tboot.lambda,.025),quantile(tboot.lambda,.975))
pivotal.lambda <- c((2*t.lambda - quantile(tboot.lambda, .975)),(2*t.lambda - quantile(tboot.lambda, .0

cat("Method       95% Interval\n")
```

```
## Method       95% Interval
```

```
cat("Normal      (", Normal.alpha[1], ",    ",Normal.alpha[2], ")\n")
```

```
## Normal      ( 0.2933807 ,    0.5381598 )
```

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

```
## Pivotal     ( 0.3217764 ,      0.5578549 )
```

```
cat("Percentile  (", pivotal.alpha[1], ",    ", pivotal.alpha[2], ") \n")
```

```
## Percentile  ( 0.2736856 ,     0.5097641 )
```

```
cat("Method       95% Interval\n")
```

```
## Method       95% Interval
```

```
cat("Normal      (", Normal.lambda[1], ",    ",Normal.lambda[2], ")\n")
```

```
## Normal      ( 1.359767 ,    2.887588 )
```

```
cat("Pivotal     (", Percentile.lambda[1], ",     ", Percentile.lambda[2], ") \n")
```

```
## Pivotal     ( 1.604805 ,      3.072845 )
```

```
cat("Percentile  (", pivotal.lambda[1], ",    ", pivotal.lambda[2], ") \n")
```

```
## Percentile  ( 1.17451 ,     2.64255 )
```

```
####################
##  to get read for the max likelihood estimation
##  tryout the function we're going to use

# try nlminb -- notice how it takes a function to optimize (minimize)
# examine the result

func <- function(y){(y[1]-3)^2 + (y[2]+1)^2}
min.func <- nlminb(start=c(1,1), obj= func)
min.func$par
```

```
## [1]  3 -1
```

```
# comes up with the obvious answer


# now lets use it to get max likelihood
x1 <- data1$x
```

```r
n <- length(data1$x)
# remember we know how to MINIMIZE so
# setup theta <- c(alpha,lambda)
# and

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 #0.4407914 1.9643791
```

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

```r
# Homework # 3
# Justify the minus.likelihood fuction used above.  Note the use of "lgamma."
#
# once you have solutions you believe,
# bootstrap to get standard errors for alpha and lambda
# and produce an extimated confidence interval
#
# 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.

n<-227
x<-rgamma(n, shape=max.likelihood$par[1], rate = max.likelihood$par[2])
x
```

```
##   [1] 7.559414e-02 5.819406e-01 3.174927e-01 1.417432e-01 5.998539e-02
##   [6] 6.289493e-03 1.256272e-05 1.903829e-01 3.139768e-01 3.288606e-02
##  [11] 1.153558e-01 2.738425e-01 2.931273e-03 1.061569e-02 7.190827e-01
##  [16] 6.698410e-02 5.366663e-02 1.787844e+00 1.891745e-01 3.126898e-01
##  [21] 6.354475e-01 7.051725e-02 1.269920e-02 1.179281e-01 8.973504e-02
##  [26] 5.160871e-01 9.370751e-02 2.137377e-02 2.489450e-02 3.592702e-01
##  [31] 6.565198e-02 5.020373e-01 3.381239e-02 8.410303e-01 3.249940e-01
##  [36] 9.322344e-03 1.337790e-03 9.973093e-01 2.132815e-01 1.343985e-03
##  [41] 3.323126e-03 8.164285e-01 5.171931e-04 9.575030e-01 4.291145e-02
##  [46] 1.829567e-03 2.298101e-01 4.382344e-01 4.518175e-02 8.921169e-01
##  [51] 6.116464e-02 3.065712e-01 1.357932e-01 4.465584e-01 5.738778e-04
##  [56] 4.318819e-02 3.276364e-02 7.116097e-03 3.795093e-01 1.758619e-02
##  [61] 5.043745e-02 9.189630e-01 1.373308e-01 1.274771e-01 1.695988e-01
##  [66] 1.888580e-01 6.193706e-01 1.699254e-05 3.049728e-02 6.334104e-02
##  [71] 1.522063e-01 6.375865e-01 5.153196e-01 3.191015e-04 3.922741e-02
##  [76] 4.150159e-01 1.774238e-01 1.143229e-02 3.146740e-04 6.819569e-03
##  [81] 1.010685e-02 1.157266e-02 1.305457e-03 9.110907e-01 3.303546e-01
##  [86] 1.240834e-02 2.578322e-01 2.382978e-02 1.728882e-01 2.137363e-02
##  [91] 9.296074e-01 4.585170e-02 2.369103e-02 1.607414e-02 8.257290e-03
##  [96] 2.111114e-01 6.350928e-01 2.256178e-01 1.500491e-02 1.075943e-01
## [101] 7.947516e-01 4.649536e-02 2.426204e-01 1.727335e-01 2.162671e+00
## [106] 2.664704e-01 1.011957e+00 1.676470e-02 5.022345e-01 2.279841e-02
## [111] 3.627009e-03 2.597793e-02 1.832017e-01 7.239520e-03 2.434597e-01
## [116] 1.291945e-02 1.279871e-02 4.210651e-02 3.974178e-05 1.725870e-01
## [121] 1.447223e-01 1.061566e-02 4.317295e-02 9.448973e-02 2.650803e-05
```

```
## [126] 2.045093e-01 5.798600e-03 7.543526e-02 2.375080e-01 1.410989e-01
## [131] 1.504992e+00 7.368456e-02 1.228522e-04 8.157786e-03 5.301659e-02
## [136] 2.711096e-01 4.323566e-02 1.606089e-01 4.087849e-01 8.930391e-01
## [141] 7.217826e-01 4.061717e-03 9.948493e-02 5.119845e-01 4.306008e-01
## [146] 2.868858e-02 4.672543e-03 1.727612e-01 8.916613e-02 4.858899e-03
## [151] 1.471960e-01 3.998285e-01 7.940685e-02 1.205432e-02 7.992797e-04
## [156] 2.690995e-01 5.092570e-02 2.128593e-01 1.948786e-01 1.524663e-02
## [161] 1.360677e-02 1.864259e-02 6.143234e-01 1.451787e-03 6.224488e-03
## [166] 2.300045e-01 1.645663e-05 1.600130e-02 1.067693e+00 6.053706e-02
## [171] 3.784137e-03 1.322312e-01 2.848427e-02 2.013748e-03 3.202159e-01
## [176] 2.110239e-02 6.965824e-05 2.588673e-01 7.004365e-03 2.411396e-01
## [181] 3.792301e-02 3.049821e-01 5.807199e-02 8.540761e-01 8.536104e-04
## [186] 2.044310e-03 1.873483e-01 6.246239e-02 4.562041e-02 3.379469e-04
## [191] 7.879883e-07 8.645299e-02 1.205415e-01 1.758142e-01 2.025856e-01
## [196] 2.029038e-03 7.633430e-03 1.218792e-01 9.738644e-03 3.084663e-01
## [201] 3.513677e-01 1.962536e-01 1.055278e-03 3.381141e-01 8.144111e-02
## [206] 6.963903e-01 1.962972e-01 2.349799e-02 3.545085e-04 1.319205e-01
## [211] 3.140254e-01 2.721285e-01 1.482365e-01 1.485589e-01 1.480982e-01
## [216] 8.107761e-01 2.435260e-04 1.704461e-01 1.938883e-02 3.288345e-01
## [221] 1.704507e-02 4.908978e-01 9.086208e-02 1.306065e-01 4.784317e-01
## [226] 4.845160e-02 1.732012e+00
```

```
mu.hat<-mean(x)
sigma2.hat<-var(x)
t.alpha<-mu.hat^2/sigma2.hat
t.lambda<-mu.hat/sigma2.hat
B <- 1000
tboot.alpha <- rep(0,B)
tboot.lambda <- rep(0,B)
for(i in 1:B){
  x.s <- sample(x, n, replace=TRUE)
  tboot.alpha[i]<-mean(x.s)^2/var(x.s)
  tboot.lambda[i]<-mean(x.s)/var(x.s)
}
var(tboot.alpha)
```

```
## [1] 0.004027666
```

```
var(tboot.lambda)
```

```
## [1] 0.1112272
```

```
se.alpha <- sqrt(var(tboot.alpha))
se.lambda <- sqrt(var(tboot.lambda))
se.alpha
```

```
## [1] 0.0634639
```

```
se.lambda
```

```
## [1] 0.3335074
```

```r
qnorm(.025,0,1) #-1.96
```

```
## [1] -1.959964
```

```r
qnorm(.975,0,1) #1.96
```

```
## [1] 1.959964
```

```r
Normal.alpha <- c(t.alpha - 1.96*se.alpha, t.alpha + 1.96*se.alpha)
Percentile.alpha <- c(quantile(tboot.alpha,.025),quantile(tboot.alpha,.975))
pivotal.alpha <- c((2*t.alpha - quantile(tboot.alpha, .975)),(2*t.alpha - quantile(tboot.alpha, .025)))

Normal.lambda <- c(t.lambda - 1.96*se.lambda, t.lambda + 1.96*se.lambda)
Percentile.lambda <- c(quantile(tboot.lambda,.025),quantile(tboot.lambda,.975))
pivotal.lambda <- c((2*t.lambda - quantile(tboot.lambda, .975)),(2*t.lambda - quantile(tboot.lambda, .0

cat("Method       95% Interval\n")
```

```
## Method       95% Interval
```

```r
cat("Normal      (", Normal.alpha[1], ",    ",Normal.alpha[2], ")\n")
```

```
## Normal      ( 0.3297133 ,    0.5784918 )
```

```r
cat("Pivotal     (", Percentile.alpha[1], ",     ", Percentile.alpha[2], ") \n")
```

```
## Pivotal     ( 0.3580246 ,      0.605507 )
```

```r
cat("Percentile  (", pivotal.alpha[1], ",     ", pivotal.alpha[2], ") \n")
```

```
## Percentile  ( 0.3026981 ,     0.5501806 )
```

```r
cat("Method       95% Interval\n")
```

```
## Method       95% Interval
```

```r
cat("Normal      (", Normal.lambda[1], ",    ",Normal.lambda[2], ")\n")
```

```
## Normal      ( 1.406257 ,    2.713606 )
```

```r
cat("Pivotal     (", Percentile.lambda[1], ",     ", Percentile.lambda[2], ") \n")
```

```
## Pivotal     ( 1.597498 ,      2.881862 )
```

```r
cat("Percentile  (", pivotal.lambda[1], ",    ", pivotal.lambda[2], ") \n")
```

```
## Percentile  ( 1.238 ,     2.522364 )
```