

rain.R

jiayuan

Fri Oct 9 15:19:07 2015

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

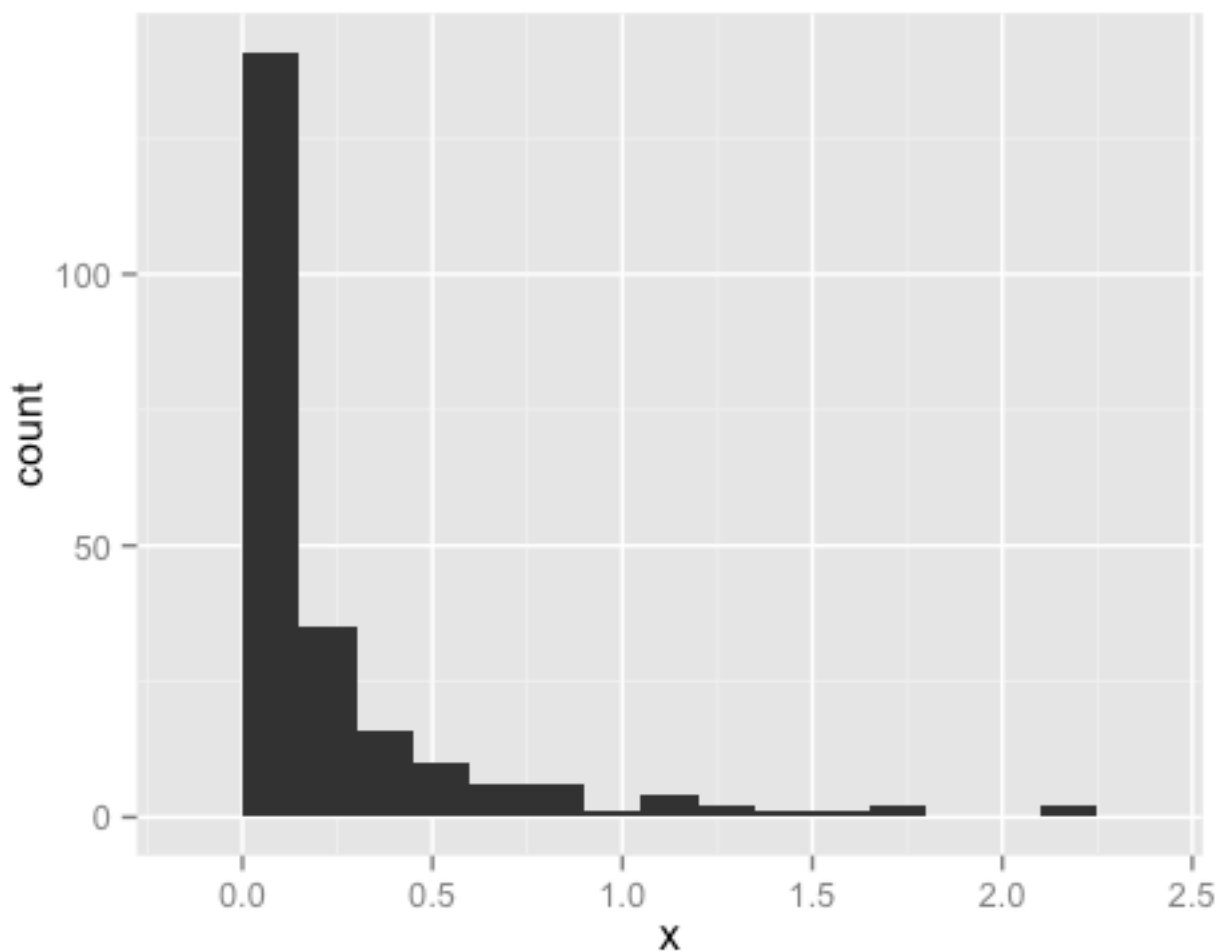
## V11 V12 V13 V14 V15 V16
## 0.020 0.001 0.080 1.720 0.490 0.020

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 gamma or be simply looking it up
## use the following facts about the gamma function
##
## first 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-square
## 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) # 0.376
lambda <- mean(data1$x)/var(data1$x) # 1.677

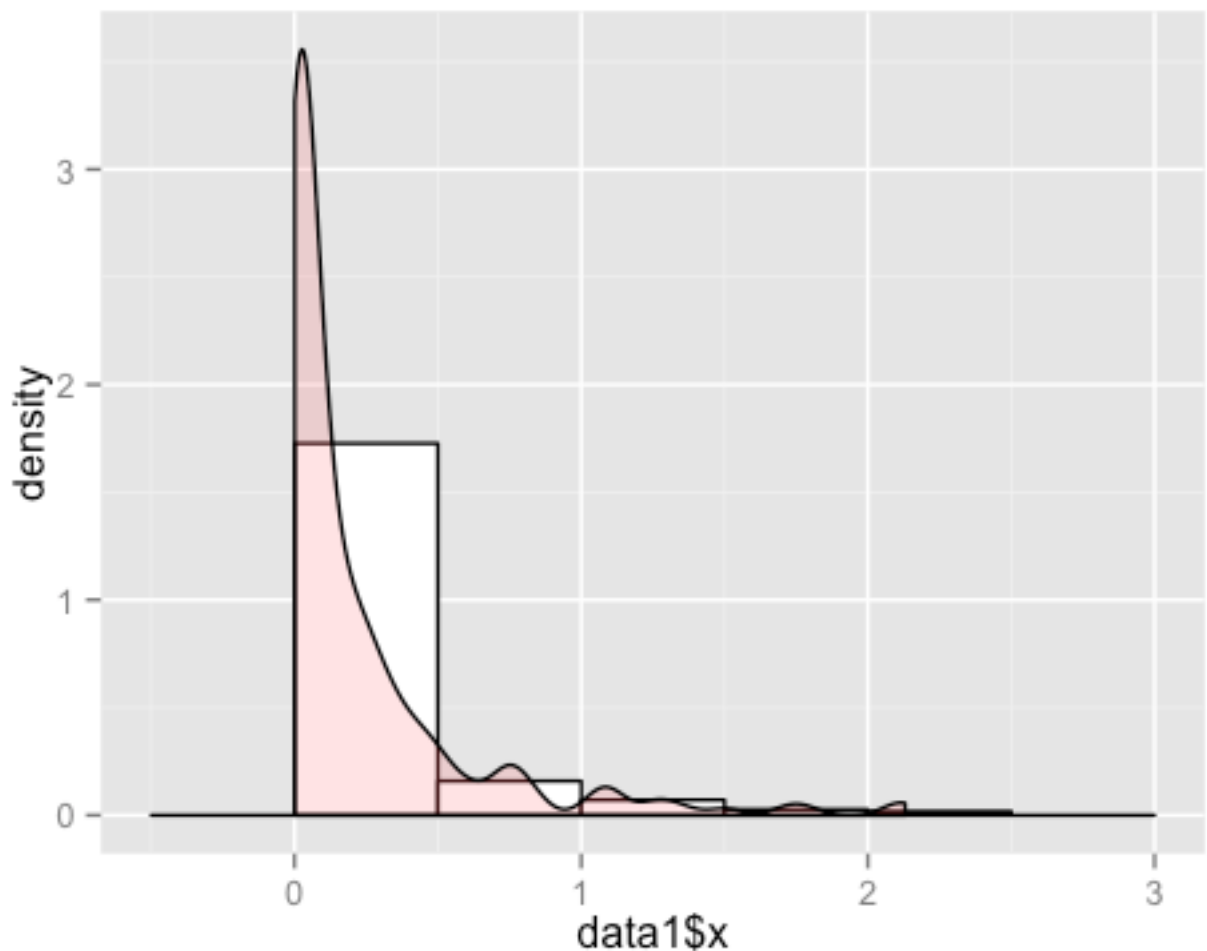
# Homework #1: Now make a plot ths superimposes the gamma density with the alpha and la
```

```

mbda 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

```



```

##
## 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] 1.639427e-02 4.923204e-05 9.256490e-02 6.706518e-02 1.550918e-04
## [6] 7.026665e-02 8.856718e-01 4.122631e-02 1.595571e-03 3.940848e-01
## [11] 2.159636e-02 2.472996e-01 1.674156e-01 1.827778e-01 1.386501e-04
## [16] 1.555116e-01 2.042835e-02 3.854851e-01 4.511481e-01 2.893594e-02
## [21] 2.510354e-04 1.357537e-01 3.510195e-01 8.251238e-02 2.690851e-01
## [26] 3.754204e-02 7.605440e-03 8.675528e-01 1.853597e-04 1.308938e-01
## [31] 4.003111e-04 3.389376e-04 3.581098e-01 3.862916e-01 1.630211e-06
## [36] 9.665731e-02 6.484413e-03 1.022977e-02 4.450054e-02 1.169065e-01
## [41] 1.054746e+00 1.646271e-01 1.524315e-04 5.978809e-01 1.869772e-02
## [46] 2.527131e-01 1.937999e-01 4.131944e-01 6.681168e-02 1.933194e-02
## [51] 1.017120e-01 7.067963e-04 5.577661e-03 2.783929e-03 1.075739e-02
## [56] 6.689259e-03 9.291403e-02 1.210790e+00 2.650500e-01 1.997948e-04
## [61] 1.879767e-01 8.500479e-03 1.295352e-01 5.333991e-02 6.171825e-02
## [66] 1.668321e-03 2.311995e-01 4.471265e-01 2.343563e-03 1.461467e-05
## [71] 5.980965e-01 1.588027e-08 4.829600e-03 1.973625e-03 3.639743e-01
## [76] 9.770380e-02 8.082332e-02 2.068310e-01 4.138841e-04 3.387886e-02
## [81] 5.645456e-02 4.242555e-01 2.289931e-01 4.583477e-02 1.860485e-02
## [86] 1.171652e-03 9.661801e-04 2.865663e-02 4.056093e-01 3.450456e-01
## [91] 7.810107e-01 2.212486e-03 1.817362e-02 8.118254e-01 3.690014e-02
## [96] 2.524717e-03 1.790297e-05 2.380201e-01 1.283508e-03 2.737689e+00
## [101] 5.875510e-01 1.477087e-01 1.406369e-01 2.102148e-01 8.726717e-02
## [106] 5.971951e-03 7.930696e-02 2.458451e-02 4.055686e-02 5.446204e-01
## [111] 1.322382e-04 3.285335e-02 9.702507e-03 4.820374e-01 2.743762e-02
## [116] 1.766116e-01 1.031060e-02 3.328418e-02 1.262724e-01 8.784556e-02
## [121] 8.297732e-02 1.184878e+00 1.507712e-01 3.603137e-01 8.091807e-02
## [126] 8.051061e-02 3.225980e-04 7.428479e-03 4.209448e-01 2.075289e-01
## [131] 1.348936e-02 4.035579e-01 3.931314e-01 2.970388e-01 6.669386e-01
## [136] 2.345437e-02 9.014066e-01 3.817986e-01 2.752373e-01 2.546764e-01
## [141] 1.874700e-01 1.311387e-01 2.834407e-04 2.806906e-02 3.294843e-01
## [146] 1.668112e-03 1.464583e-06 1.036349e-02 6.639402e-01 1.567759e-03
## [151] 3.070776e-04 6.385966e-01 1.911597e-01 3.739401e-02 1.220197e+00
## [156] 7.583761e-02 1.838914e-02 5.155551e-01 4.311688e-04 5.196643e-03
## [161] 7.646333e-05 1.514323e-03 2.989866e-02 1.051129e+00 2.224292e-01
## [166] 3.652177e-05 6.965808e-02 1.224655e-01 2.223358e-01 2.112172e-01
## [171] 1.258398e-02 2.523854e-04 2.225700e-02 1.910400e-01 1.649848e+00
## [176] 2.020349e-02 1.697759e-01 3.933313e-02 3.743484e-01 1.042507e+00
## [181] 4.439496e-05 6.110984e-02 6.371865e-03 6.473334e-01 1.107216e-02
## [186] 5.072382e-02 3.758099e-02 2.997655e-01 1.052165e-02 1.218356e-01
## [191] 1.295609e-02 2.952234e-02 1.391077e+00 1.354416e-05 3.758166e-02
## [196] 1.244594e-01 2.775228e-01 1.153842e-02 1.054604e-01 2.477454e-02
## [201] 3.272097e-08 6.570556e-03 1.754891e-01 4.933016e-03 1.499366e-01
## [206] 1.173698e-02 8.873582e-01 5.473135e-03 7.955292e-01 5.642808e-01
## [211] 8.767368e-02 6.904446e-02 9.827599e-03 8.459622e-03 9.113958e-01
## [216] 1.681631e-01 1.129084e-02 6.006914e-01 3.121117e-01 7.597097e-02
## [221] 3.454048e-05 3.350061e-02 1.255283e-01 7.724296e-01 2.583937e-02
## [226] 1.324527e-02 8.399678e-04
```

```
mu.hat<-mean(x)
sigma2.hat<-var(x)
t.alpha<-mu.hat^2/sigma2.hat
t.alpha
```

```
## [1] 0.3750887
```

```
t.lambda<-mu.hat/sigma2.hat
t.lambda
```

```

## [1] 1.813556

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] 0.004438629

var(tboot.lambda)

## [1] 0.1482477

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

## [1] 0.06662304

se.lambda

## [1] 0.3850294

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)))

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, .025)))

cat("Method    95% Interval\n")

## Method    95% Interval

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

## Percentile  ( 0.2754058 ,    0.5249979 )

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

## Pivotal  ( 0.2251795 ,    0.4747717 )

cat("Method    95% Interval\n")

## Method    95% Interval

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

## Percentile  ( 1.295234 ,    2.738746 )

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

## Pivotal  ( 0.8883659 ,    2.331878 )

```

```
#####
## 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$x
n <- length(data1$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(x1))-theta[2]*sum(x1))}
max.likelihood <- nlminb(start=c(.3762, 1.6767), obj = minus.likelihood)
max.likelihood$par #0.4407914 1.9643791

## [1] 0.4407914 1.9643791

# 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=alpha, rate = lambda)
x

## [1] 1.123218e-03 8.623543e-03 1.211448e-05 2.566566e-03 4.820499e-04
## [6] 1.012231e-01 2.613928e-01 6.975579e-03 4.546736e-03 1.586898e-01
## [11] 3.123449e-01 6.075826e-02 2.082893e-02 6.883695e-01 4.866521e-04
## [16] 3.364883e-02 1.125614e+00 2.209795e-01 4.571497e-01 7.232508e-03
## [21] 6.465334e-01 1.041690e+00 3.827135e-01 4.506556e-02 9.645593e-01
## [26] 3.781964e-01 4.255471e-01 6.691175e-02 3.861511e-02 6.340193e-02
## [31] 7.405088e-02 2.849427e-02 1.068946e-01 7.168035e-03 3.765277e-01
## [36] 8.411046e-01 3.514033e-01 5.550095e-01 1.522266e-02 3.297363e-01
## [41] 5.583317e-02 2.740958e-07 6.102581e-02 9.492104e-04 1.097135e-01
## [46] 6.137425e-02 1.495099e-01 3.970698e-01 7.667044e-03 1.536638e-01
## [51] 1.468429e-02 3.475621e-01 3.700504e-02 1.862580e-03 7.876792e-02
## [56] 1.188286e-04 4.457826e-01 3.615261e-01 6.721439e-03 2.246372e-01
```

```
## [61] 1.908372e-02 1.746113e-02 9.423595e-04 6.112317e-02 2.339778e-01
## [66] 2.926173e-02 8.565216e-04 3.802780e-01 1.986077e-02 7.414478e-02
## [71] 1.229517e-01 1.366666e-03 4.629016e-04 3.071196e-01 6.144004e-03
## [76] 8.624421e-03 1.961949e-01 6.453813e-01 1.220957e+00 3.693175e-01
## [81] 8.273769e-02 2.871189e-01 2.279864e-01 3.758845e-01 2.417371e-01
## [86] 3.161632e-04 8.332463e-02 1.054142e+00 6.199104e-03 7.889043e-02
## [91] 4.353224e-03 2.706568e-02 2.263281e-01 7.541246e-02 8.725314e-02
## [96] 7.459667e-03 3.359551e-02 2.181032e-02 5.692365e-01 6.181189e-02
## [101] 4.483376e-08 8.315902e-02 4.915722e-02 9.869016e-02 3.268713e-01
## [106] 1.132465e+00 1.497294e-01 2.071243e-03 3.347683e-01 7.699790e-02
## [111] 1.795446e+00 1.116713e-03 4.194077e-01 3.579860e-01 6.717797e-02
## [116] 4.872762e-01 5.024440e-01 1.635917e-07 8.476849e-01 9.389861e-10
## [121] 4.535103e-02 1.461967e+00 2.929469e-04 2.199374e-01 6.652120e-02
## [126] 2.418419e-04 7.136333e-04 5.149558e-01 1.428472e+00 6.828473e-05
## [131] 1.415849e+00 1.665849e-01 3.375721e-01 5.817140e-01 2.298766e-01
## [136] 7.311580e-02 9.405542e-03 3.995754e-02 3.724229e-02 5.083838e-03
## [141] 1.165604e-02 8.405478e-02 4.429846e-02 9.223050e-03 3.668019e-01
## [146] 3.705767e-02 5.617714e-01 3.204324e-01 3.375629e-02 4.896949e-03
## [151] 5.515808e-01 9.406514e-01 3.589457e-02 7.937316e-02 9.497017e-01
## [156] 3.706386e-02 1.486663e-02 1.172232e+00 4.915000e-01 7.071635e-01
## [161] 4.821936e-01 4.089738e-01 2.042460e-01 2.642803e-01 1.046991e-01
## [166] 8.894088e-02 8.183651e-02 5.328752e-02 2.037262e-01 1.144375e-04
## [171] 2.577516e-02 2.793385e-03 7.154105e-02 2.213728e-01 4.759158e-01
## [176] 1.611973e-01 1.506590e-01 1.652454e-01 2.735582e-01 1.213540e-02
## [181] 1.495131e-03 1.245247e-01 2.854072e-03 1.939277e-02 3.510502e-01
## [186] 3.416523e-02 1.808204e-02 3.780056e-04 8.122901e-01 3.290763e-04
## [191] 6.166418e-01 5.720898e-04 6.106560e-01 2.852132e-01 2.973805e-01
## [196] 2.240075e-02 7.488249e-02 1.525748e-01 4.777757e-01 4.517058e-02
## [201] 3.834121e-05 1.391544e-01 2.319561e-02 7.766773e-01 2.278272e-04
## [206] 1.083960e-02 1.012190e-02 1.879631e-02 2.818042e-03 2.247814e-01
## [211] 4.015875e-02 5.266465e-02 4.919350e-03 1.516687e-01 1.857685e-01
## [216] 1.115428e-03 7.796378e-04 3.155983e-01 1.269986e-01 2.867528e-04
## [221] 2.228539e-03 2.177750e-03 2.500017e-02 1.270019e+00 5.095561e-05
## [226] 5.076418e-01 5.763998e-01
```

```
minus.likelihood <- function(theta){
  -(n*theta[1]*log(theta[2])-n*lgamma(theta[1])+
    (theta[1]-1)*sum(log(x1))-theta[2]*sum(x1))
}
max.likelihood <- nlminb(start=c(.3762, 1.6767), obj = minus.likelihood)
para<-max.likelihood$par
t.alpha<-para[1]
t.alpha

## [1] 0.4407914

t.lambda<-para[2]
t.lambda

## [1] 1.964379

B <- 1000
tboot.alpha <- rep(0,B)
tboot.lambda <- rep(0,B)
for(i in 1:B){
  x.s <- sample(x, n, replace=TRUE)
```

```

minus.likelihood <- function(theta) {
  -(n*theta[1]*log(theta[2])-n*lgamma(theta[1])+
    (theta[1]-1)*sum(log(x.s))-theta[2]*sum(x.s))}
alpha1<-mean(x.s)^2/var(x.s)
lambda1<-mean(x.s)/var(x.s)
max.likelihood <- nlminb(start=c(alpha1, lambda1), obj = minus.likelihood)
tboot.alpha[i]<-max.likelihood$par[1]
tboot.lambda[i]<-max.likelihood$par[2]
}

var(tboot.alpha)

## [1] 4.604771

var(tboot.lambda)

## [1] 0.03173062

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

## [1] 2.145873

se.lambda

## [1] 0.1781309

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)))

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, .025)))

cat("Method    95% Interval\n")

## Method    95% Interval

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

## Percentile  ( -11.12563 ,    -2.740968 )

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

## Pivotal  ( 3.622551 ,    12.00721 )

cat("Method    95% Interval\n")

## Method    95% Interval

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

## Percentile  ( 2.568986e-15 ,    2.993527e-13 )

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

## Pivotal  ( 3.928758 ,    3.928758 )

```