ST2137 Tutorial 8

Wei Hao Khoong

6 April 2019

Q1

Inversion Method

Let $U = F(X)$. Then $F(X) = 1 - (\frac{\beta}{X})^{\alpha}$, for $X $. Hence $X = \frac{\beta}{(1-U)^{1-\alpha}}$.

```
rpareto = function(n, a, b){
  u = runif(n)
  x = b/((1-u)^(1/a))
  return(x)
}
```

```
rpareto(30,4,2)
```

```
##  [1] 3.142961 3.102971 4.587648 3.465512 2.242528 2.407634 2.293807
##  [8] 4.170991 3.602366 2.133248 2.339509 2.402116 2.596601 3.966964
## [15] 2.382627 7.010474 2.170537 8.380715 2.702882 4.195204 3.202300
## [22] 2.552545 5.498315 2.205813 2.483572 2.543854 2.027179 2.193499
## [29] 2.502645 2.169274
```

Q2

Part (a) - Generating Samples of Size 1000 From Chi-Square Distributions With 6 & 10 Degrees of Freedom Respectively

```
a = 3
b = 5
n = 10^3
x1 = rchisq(n,a*2)
x2 = rchisq(n,b*2)
```

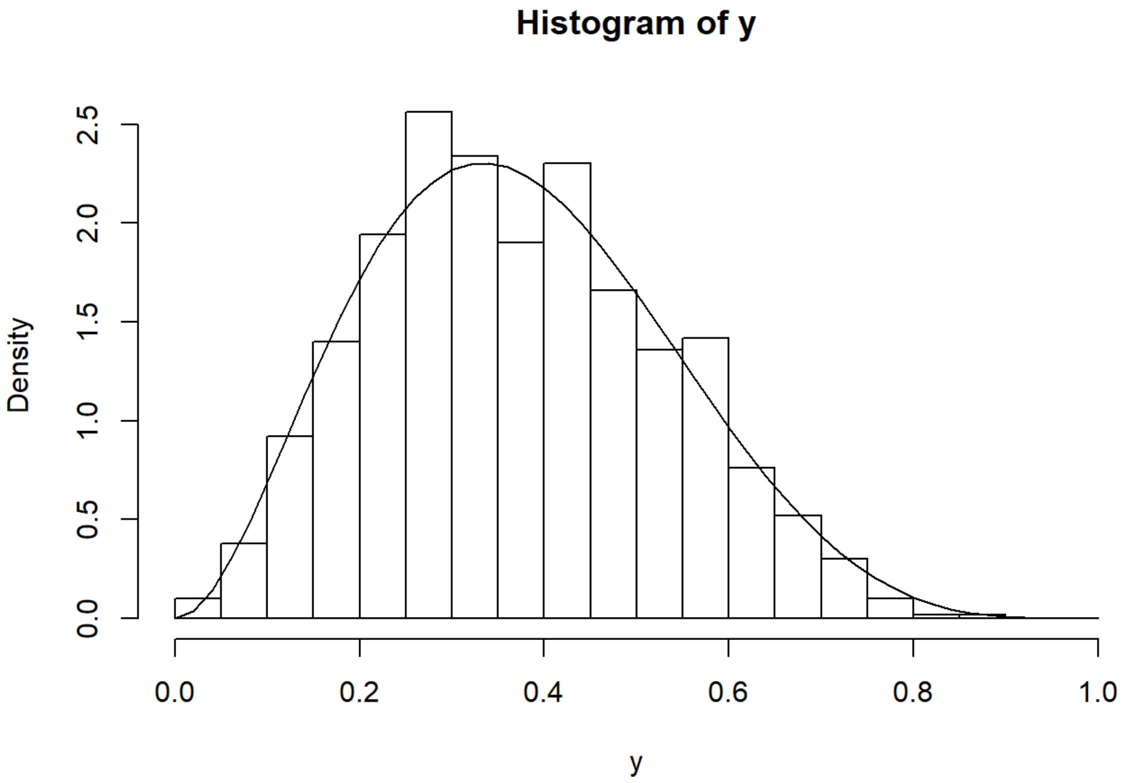
Part (b) - Generating Samples of Size 1000 From $Beta(3,5)$ Distribution Using Two Samples Obtained In (a)

```
y = x1/(x1+x2)
```

Part (c) - Histogram With True Density Superimposed

```
# true density
xpt <- seq(0, 1, 0.02)
ypt <- dbeta(xpt, a, b)

# Plotting Density Function
hist(y, freq = F, breaks = seq(0, 1, 0.05))
lines(xpt, ypt)
```



Q3(a) - $n = 9$

```
realmu=c(-1,-0.5,0.5,1)
power=numeric(length(realmu))
ns=1000

means = numeric(ns)
sds = numeric(ns)
for(mu in realmu){
  for(i in 1:ns){
    X = rnorm(9,mu,2)
    means[i]=mean(X)
    sds[i]=sd(X)
  }
  ts = means/(sds/sqrt(9))
  tcrit = qt(0.975,8)
  power[which(realmu==mu)] = sum(abs(ts) > tcrit)/ns
}

cbind(realmu,power)
```

```
##      realmu power
## [1,]   -1.0 0.262
## [2,]   -0.5 0.094
## [3,]    0.5 0.096
## [4,]    1.0 0.252
```

The power is getting bigger as μ moves away from the value under $H_0 : \mu = 0$.

Q3(b) - $n = 25$

```
for(mu in realmu){
  for(i in 1:ns){
    X = rnorm(25,mu,2)
    means[i]=mean(X)
    sds[i]=sd(X)
  }
  ts = means/(sds/sqrt(25))
  tcrit = qt(0.975,24)
  power[which(realmu==mu)] = sum(abs(ts) > tcrit)/ns
}

cbind(realmu,power)
```

```
##      realmu power
## [1,]   -1.0 0.665
## [2,]   -0.5 0.215
## [3,]    0.5 0.249
## [4,]    1.0 0.663
```

The power is getting bigger with greater magnitude when $n = 25$ as compared to $n = 9$ as μ moves away from the value under $H_0 : \mu = 0$.

Q4(a)-(b) - Bootstrap Estimate Of Bias & Standard Error Of 25% Trimmed Mean

```
t10q1 <- read.table("C:/Users/Wei Hao/Desktop/ST2137/Tutorials/Data/price.txt", header=TRUE)
attach(t10q1)
library(boot)
set.seed(123)
tm25 <- function(x,b){
  return(mean(x[b],0.25))
}
boot.tm25 <- boot(price, statistic=tm25, R=10000)
boot.tm25
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = price, statistic = tm25, R = 10000)
##
##
## Bootstrap Statistics :
##     original   bias    std. error
## t1* 244.0019 0.713921    17.79328
```

The estimate for the bias is 0.7744433 while the standard error is 17.81654.

Q4(c) - 95% Basic Bootstrap Confidence Interval Of 25% Trimmed Mean

```
boot.ci(boot.tm25, type=c("basic","norm","perc"))
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 10000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot.tm25, type = c("basic", "norm", "perc"))
##
## Intervals :
## Level      Normal              Basic              Percentile
## 95%   (208.4, 278.2 )   (206.6, 275.2 )   (212.8, 281.4 )
## Calculations and Intervals on Original Scale
```

The 95% basic bootstrap confidence interval is $(205.5, 275.3)$ while the confidence interval using the bootstrap percentile method is $(212.7, 282.5)$.

Q4(d) - 5% Confidence Interval For The 25% Trimmed Mean Using Bootstrap Percentile Method

```
# without boot package
ns=10000 #number of bootstrap samples
tm25.sim = numeric(ns)

for(i in 1:ns) {
  tm25.sim[i]=mean(price[sample(length(price),replace=TRUE)],0.25)
}
est = mean(price,0.25) # original estimate
biashat = mean(tm25.sim) - est
sdhat = sd(tm25.sim)
alpha = 0.05
low = quantile(tm25.sim,alpha/2)
high = quantile(tm25.sim,1-alpha/2)
```

```
biashat
```

```
## [1] 0.6865054
```

```
sdhat
```

```
## [1] 17.46953
```

```
cat("A",100*(1-alpha),"% bootstrap CI is (",2*est - high,",",2*est-low,")","\n")
```

```
## A 95 % bootstrap CI is ( 207.5861 , 275.1348 )
```

```
cat("Another",100*(1-alpha),"% bootstrap CI is (",low,",",high,")","\n")
```

```
## Another 95 % bootstrap CI is ( 212.869 , 280.4177 )
```