STAT 321 Final

Charles Hwang

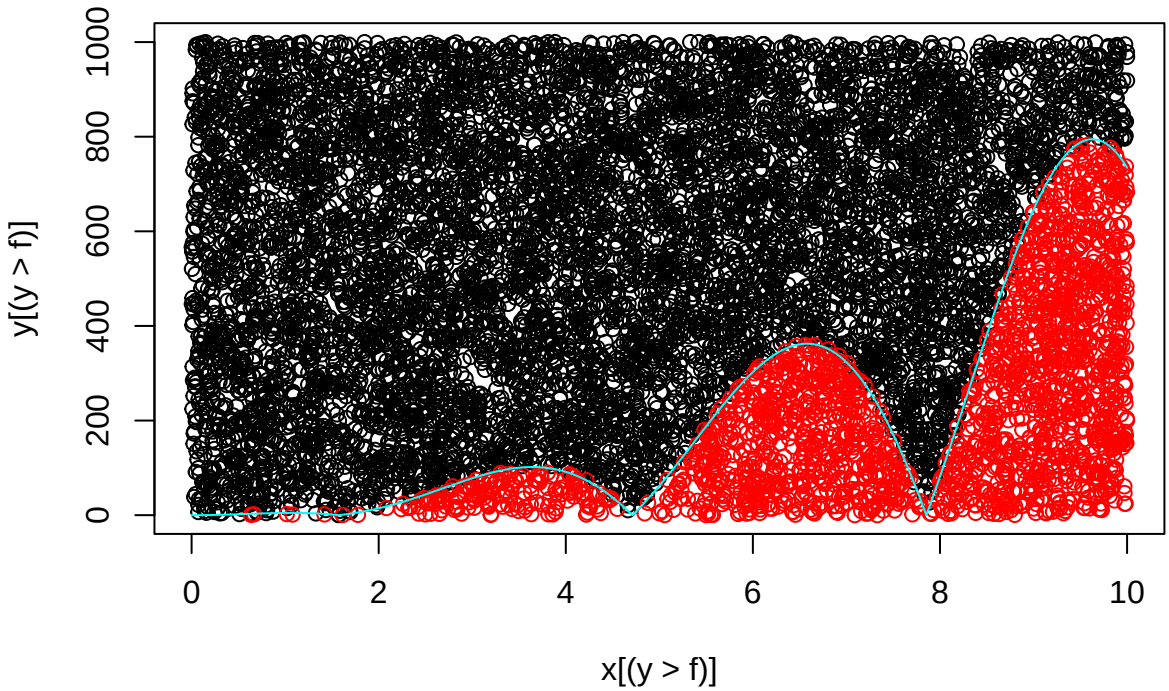
Professor Matthews

STAT 321-001

10 December 2019

Problem 1

```
rm(list=ls())
library(stats)
set.seed(1012)
x <- runif(10000,0,10)    # Checking function on Desmos, y is between 0 and 1000
y <- runif(10000,0,1000) # Choosing n = Area = 10*1000 = 10000
f <- 8.75166*abs(x^2*cos(x))
fx <- function(x) 8.75166*abs(x^2*cos(x))
plot(x[(y>f)],y[(y>f)])
points(x[(y<=f)],y[(y<=f)],col="red")
curve(fx,0,10,n=10000,col="cyan",add=TRUE)
```




```
length(y[(y<=f)])
```

```
## [1] 2015
```

```r
integrate(function(x) 8.75166*abs(x^2*cos(x)),0,10)$value
```

```
## [1] 2020.002
```

```r
cat("MarginOfError:",integrate(function(x) 8.75166*abs(x^2*cos(x)),0,10)$value-length(y[(y<=f)]),"\nPer
```

```
## MarginOfError: 5.00241
## Percent Error: 0.2476437
```

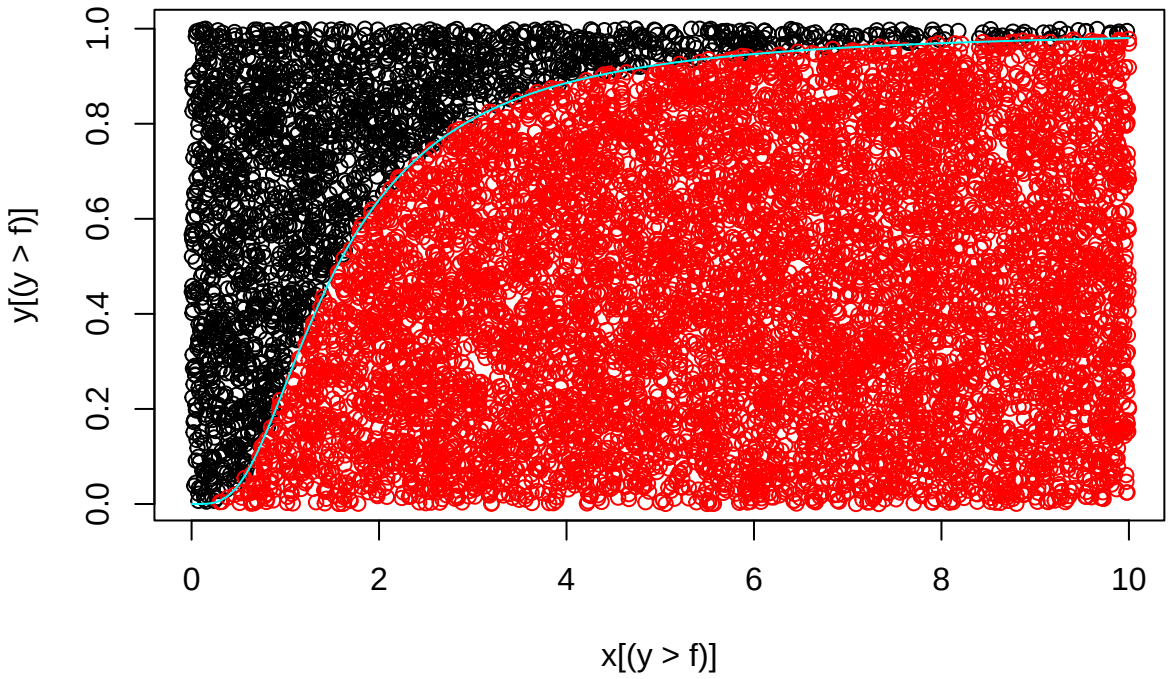
Problem 2

Problem 2a

```r
rm(list=ls())
set.seed(1012)
f <- function(x,a=2,b=1,p=2) {
  b*(x^(-1/p)-1)^(-1/a) # Inverse: b*(x^(-1/p)-1)^(-1/a)
  }
f(runif(1000)) # Output suppressed
```

Problem 2b

```r
set.seed(1012)
x <- runif(10000,0,10)
y <- runif(10000,0,1)
a <- 2
b <- 1
p <- 2
f <- (1+(x/b)^-a)^-p
fx <- function(x,a=2,b=1,p=2) {
  (1+(x/b)^-a)^-p
  }
plot(x[(y>f)],y[(y>f)])
points(x[(y<=f)],y[(y<=f)],col="red")
curve(fx,0,10,n=10000,col="cyan",add=TRUE)
```




```
e <- .05 # Choosing arbitrary e
mean(y[(abs(y-f) <= e)])
```

```
## [1] 0.7772834
```

```
mean(fx(x))
```

```
## [1] 0.7831661
```

```
cat("    Mean - MarginOfError:",mean(fx(x))-mean(y[(abs(y-f) <= e)]),"\n    Mean - Percent Error:",100*
```

```
##     Mean - MarginOfError: 0.005882698
##     Mean - Percent Error: 0.7511431
```

```
d <- .01 # Choosing arbitrary d
max(y[(abs(x-5) <= d)])
```

```
## [1] 0.9113819
```

```
median(fx(x))
```

```
## [1] 0.923069
```

```
cat("  Median - MarginOfError:",median(fx(x))-max(y[(abs(x-5) <= d)]),"\n  Median - Percent Error:",100
```

```
##   Median - MarginOfError: 0.01168715
##   Median - Percent Error: 1.266118
```

```
var(y)
```

```
## [1] 0.0831433
```

```
var(fx(x))
```

```
## [1] 0.08144914
```

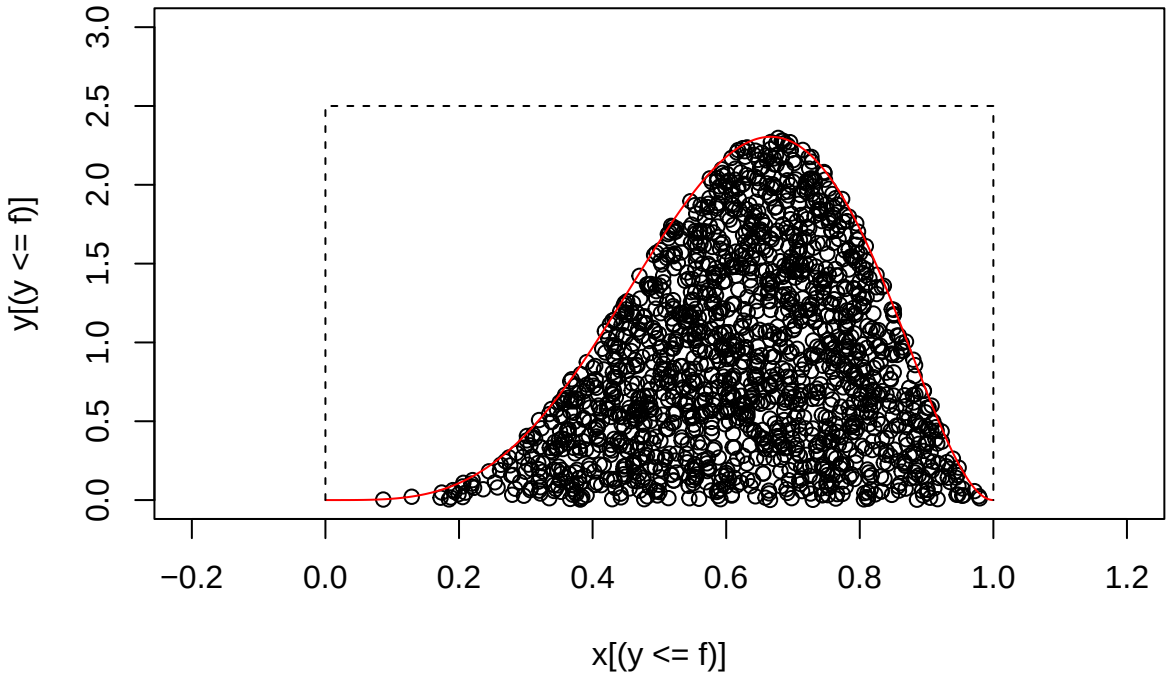
```
cat("Variance - MarginOfError:",var(y)-var(fx(x)),"\nVariance - Percent Error:",100*(var(y)-var(fx(x))),
```

```
## Variance - MarginOfError: 0.001694162
```

```
## Variance - Percent Error: 2.037641
```

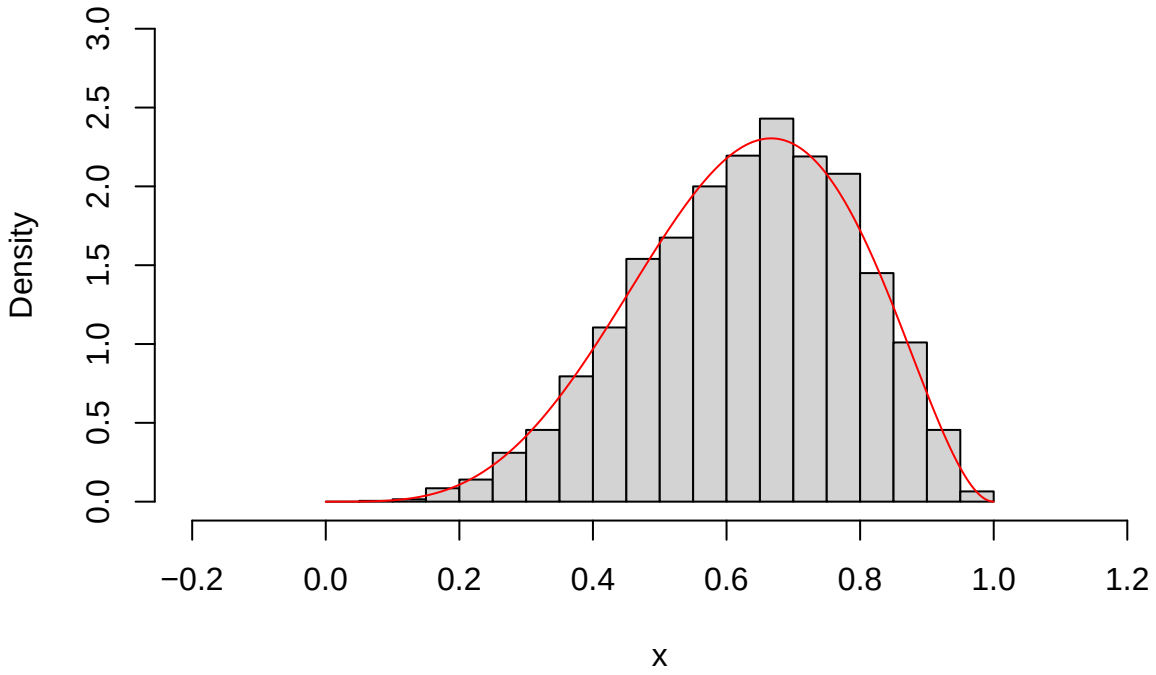
Problem 3

```
rm(list=ls())
set.seed(1012)
x <- runif(4000,0,1)
y <- runif(4000,0,2.5)
a <- 5
b <- 3
f <- x^(a-1)*(1-x)^(b-1)/(beta(a,b))
fx <- function(x,a=5,b=3) {
  x^(a-1)*(1-x)^(b-1)/(beta(a,b))
  }
plot(x[(y<=f)],y[(y<=f)],xlim=c(-.2,1.2),ylim=c(0,3))
curve(fx,0,1,n=10000,col="red",add=TRUE)
lines(c(0,0,1,1),c(0,2.5,2.5,0),lty=2)
```



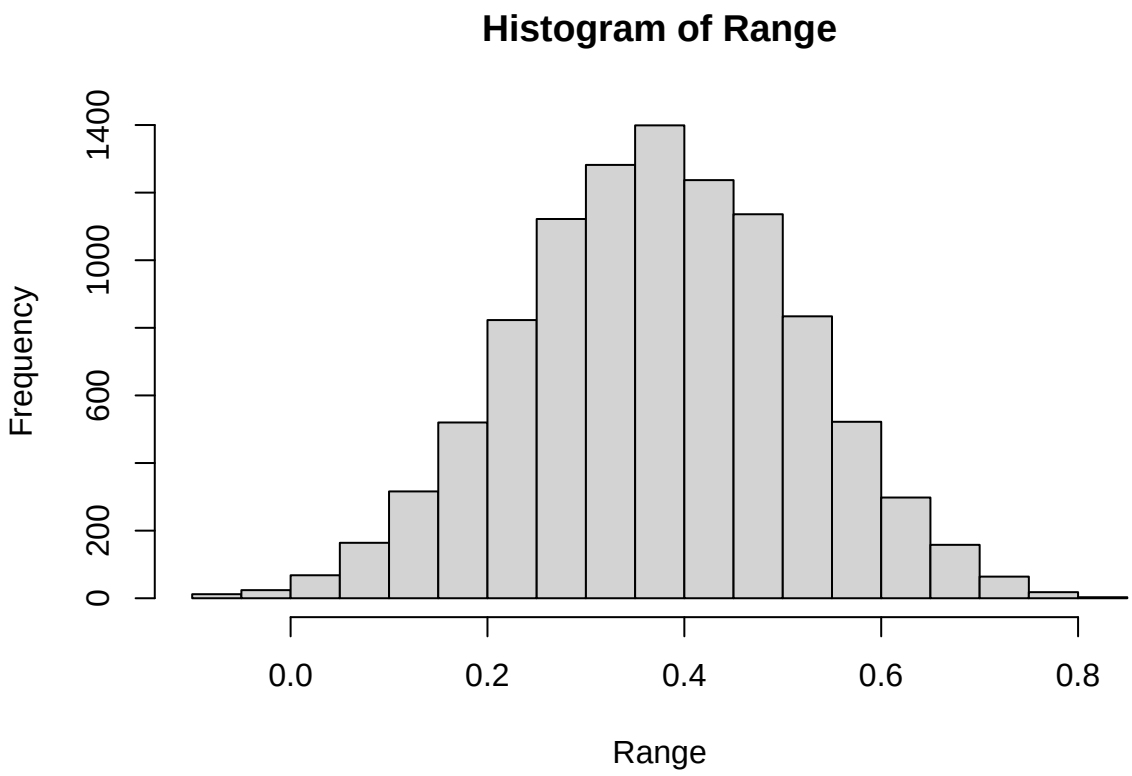

```
r <- function(fx,a,b,k) { # Same data displayed as a histogram
  while (TRUE) {
    x <- runif(1,a,b)
    y <- runif(1,0,k)
    if (y < fx(x)) return(x)
  }
}
fx <- function(x,a=5,b=3) {
  x^(a-1)*(1-x)^(b-1)/(beta(a,b))
  }
n <- 4000
x <- rep(0,n)
for(i in 1:n) {
  x[i] <- r(fx,0,1,2.5)
```

```
}
hist(x,breaks=seq(0,1,by=.05),freq=FALSE,xlim=c(-.2,1.2),ylim=c(0,3),main="")
curve(fx,0,1,n=1000,col="red",add=TRUE)
```

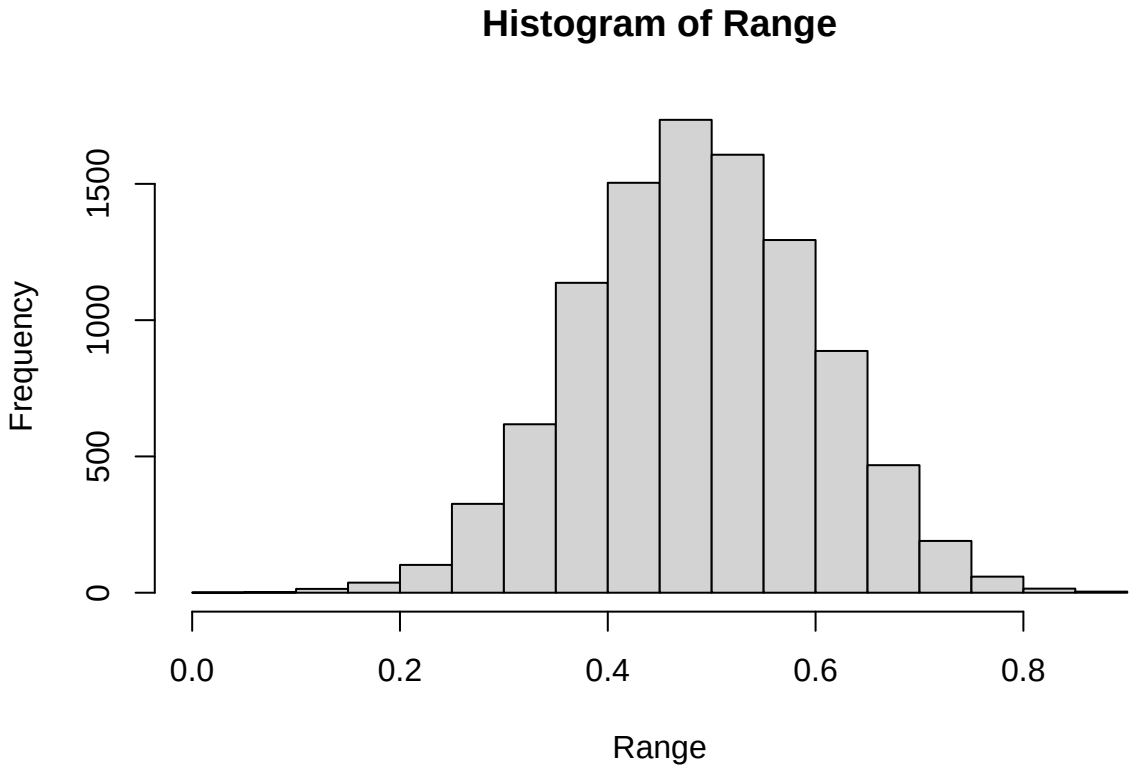


Problem 4

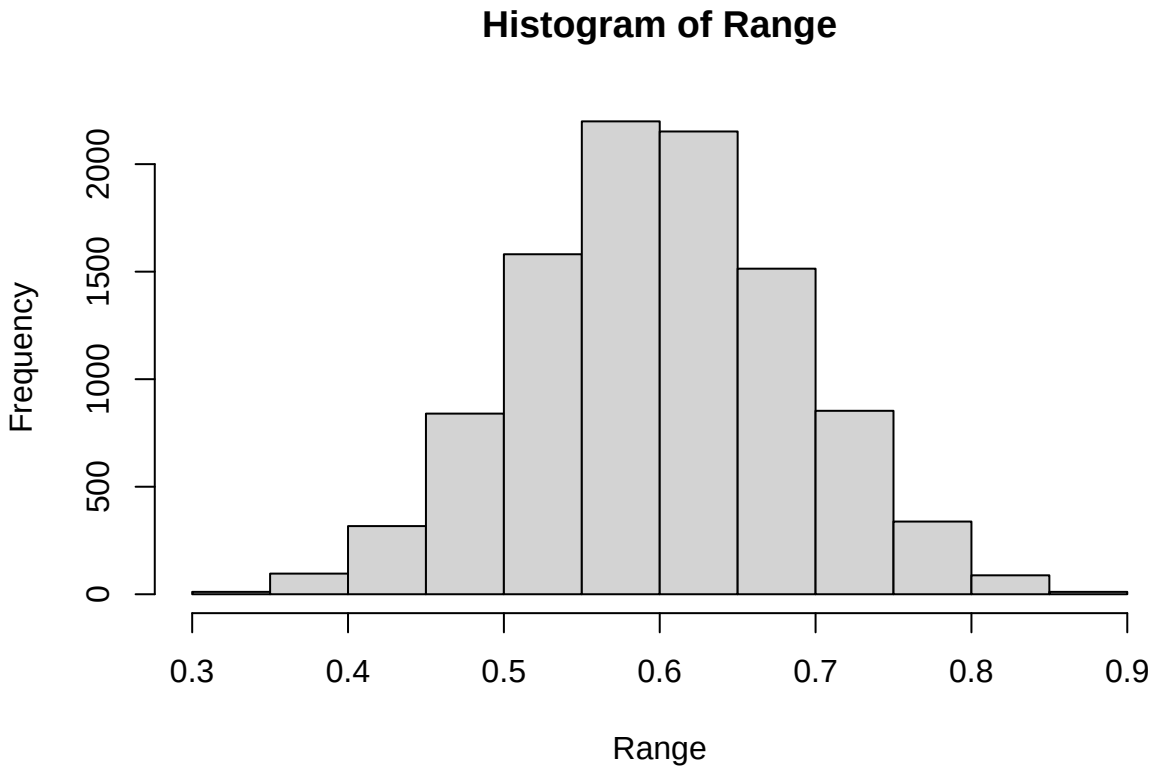
```
set.seed(1012)
a <- 3
b <- 5
r <- function(m){
  Range <- rep(0,10000)
  for (i in 1:10000) {
    Range[i] <- max(rbeta(m,3,5))-min(rbeta(m,3,5))
    }
  hist(Range)
}
r(5)
```


Histogram of Range
Frequency
0
200
600
1000
1400
0.0
0.2
0.4
0.6
0.8
Range

```
r(10)
```


Histogram of Range
Frequency
0
500
1000
1500
0.0
0.2
0.4
0.6
0.8
Range

```
r(25)
```



Problem 5

```
set.seed(1012)
m <- function(m){
  r <- rep(0,10000)
  for (i in 1:10000) {
    r[i] <- max(rbeta(m,3,5))-min(rbeta(m,3,5))
    }
  mean(r)
}
m(5)
```

```
## [1] 0.373581
```

```
m(10)
```

```
## [1] 0.4868689
```

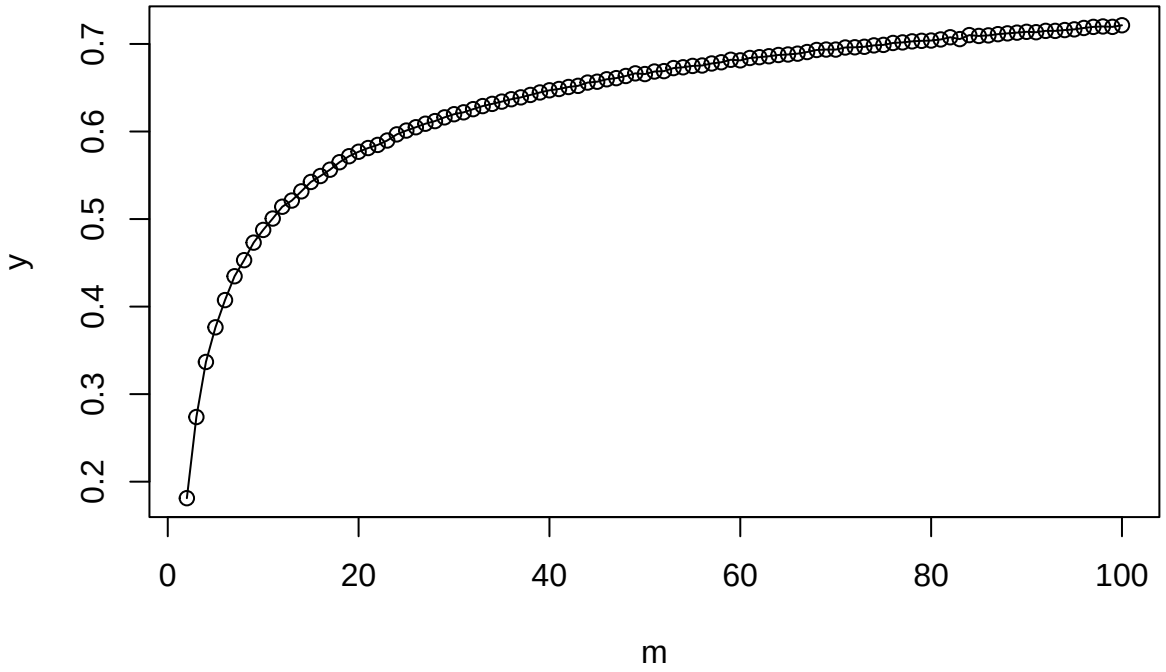
```
m(25)
```

```
## [1] 0.5997959
```

Problem 6

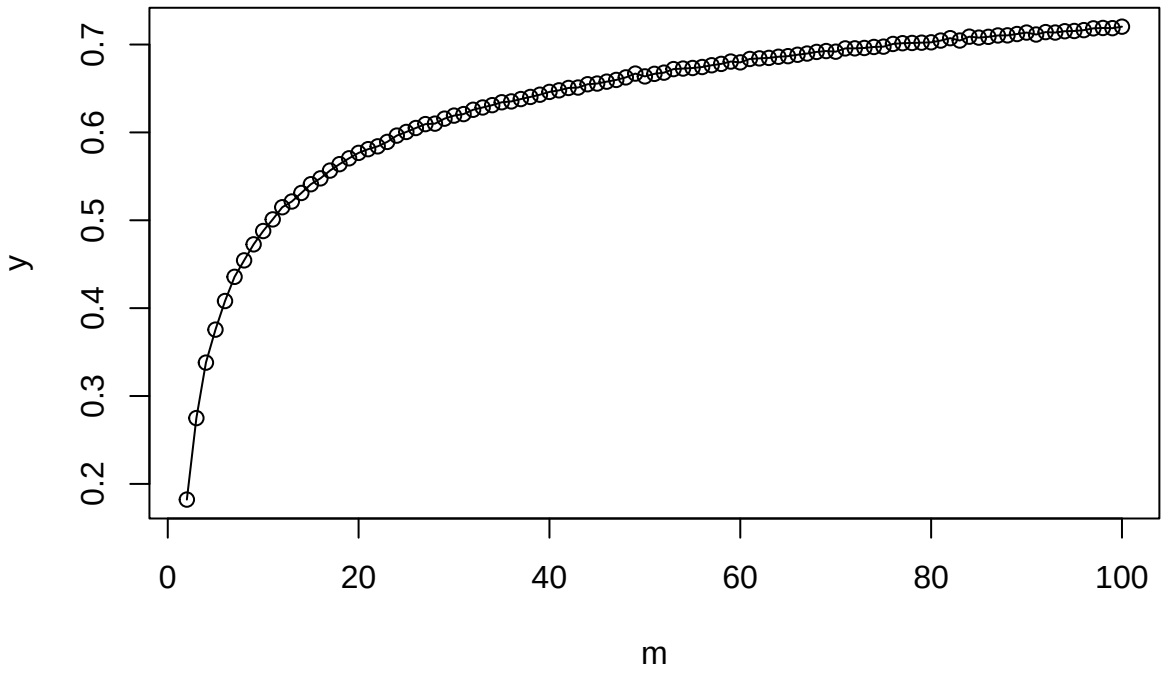
```
set.seed(1012)
nm <- function(ml,mu){
  r <- matrix(c(rep(0,(mu-ml+1)*10000)),nrow=mu-ml+1,ncol=10000)
  y <- rep(0,mu-ml+1)
  for (m in ml:mu) {
    for (i in 1:10000) {
```

```
      r[m-ml+1,i] <- max(rbeta(m,3,5))-min(rbeta(m,3,5))
      y[m-ml+1] <- mean(r[m-ml+1,])
      }
    }
  plot(c(ml:mu),y,type="o",xlab="m")
}
nm(2,100)
```



Problem 7

```
set.seed(1012)
nm <- function(ml,mu){
  r <- matrix(c(rep(0,(mu-ml+1)*10000)),nrow=mu-ml+1,ncol=10000)
  y <- rep(0,mu-ml+1)
  for (m in ml:mu) {
    for (i in 1:10000) {
      r[m-ml+1,i] <- max(rbeta(m,3,5))-min(rbeta(m,3,5))
      y[m-ml+1] <- median(r[m-ml+1,])
      }
    }
  plot(c(ml:mu),y,type="o",xlab="m")
}
nm(2,100)
```


y
0.2
0.3
0.4
0.5
0.6
0.7
0
20
40
60
80
100
m