3.1 The dataset "peabody" in Canvas contains scores on the peabody picture vocabulary text given to a sample of preschool children as part of a study on early childhood eduation

(a) Compute the sample mean of the peabody data, along with the standard error of the sample mean using the usual $S/\sqrt{n}$ formula

sample mean=46.41 SE of sample mean=1.06

(b) Use the R function median to calculate the sample median of the peabody data

median=42

(c) Unlike the sample mean, there is not a simple formula for the standard error of the sample median. Hence the bootstrap provides a useful approach. Use the bootstrap to find the standard error of the sample median for the peabody data. Note that, although there is a base R function of the sample median, you will need to write a function that can be used in the bootstrap function boot. Use 100000 bootstrap replication and use the random seed 35202.

SE=.23

(d) Suppose that we are interested in comparing the sample mean and sample median. Calculate $\bar{Y} - m$ for the peabody data, where $\bar{Y}$ is the sample mean and m is the sample median. Use the bootstrap method to find the standard error of $\bar{Y} - m$; use 100000 bootstrap replications and use the random seed 35202. Is there evidence that the true mean and median differ for the peabody data? Why or why not?

$\bar{Y} - m$=4.41

Standard error=1.04.

Based on the results of the bootstrap method the estimated bias is essentially zero. The sample mean and median is an unbiased estimator of the mean and median of the distribution so it is not surprising the estimated bias is very small. In this case there is evidence that the true mean and median differ from the peabody data since our standard error is above 1 indicating that the distribution is skewed.

In [2]:
```
library("boot")
#data_loc<-'/Users/Alexis/Documents/Spring2020/nonparametrics/data/peabody.csv'
data_loc<-'/Users/aporter1350/Documents/Courses/Spring2020/nonparametrics/data/peabody.csv'
pea<-read.csv(data_loc)
#Get n
n <- dim(pea)[1]
#Compute sample mean
sample_mean=mean(pea$peabody)
#SE of sample mean
sd(pea$peabody/(sqrt(n)))
#calculate sample median
sample_median=median(pea$peabody)
med=function(x, ind){1.253*median(x[ind])/length(ind)}
#bootstrap to calculate SE of sample median
set.seed(35202)
#SE of median
boot(pea$peabody, med, 100000)
#SE for y-m
Ym=function(x,ind){mean(x[ind])-median(x[ind])}
boot(pea$peabody, Ym, 100000)
#true value
sample_mean-sample_median
```

1.06819635491289

```
ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = pea$peabody, statistic = med, R = 1e+05)


Bootstrap Statistics :
     original       bias    std. error
t1* 0.2370541 -0.003186486 0.009008546

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = pea$peabody, statistic = Ym, R = 1e+05)


Bootstrap Statistics :
    original    bias    std. error
t1* 4.414414 0.5643674     1.04655
```

4.41441441441442

3.2 Consider the failure data for the software system used in the previous exercise. Suppose that we are interested in whether or not these data follow an exponential distribution. If Y follows an exponential distribution then the mean and standard deviation of Y are equal. Hence if the data $Y_1, Y_2, .. Y_n$ follow an exponential distribution, the value of the statistic $T = \bar{Y}/S$ should be about 1 where $\bar{Y}$ and S^2 are the sample mean and sample variance, respectively of the data.

(a) Calculate the value of T for the software failure data

T=.64

(b)Using the bootstrap method, find the bias and standard error of T. Use 100000 bootstrap replications and random seed 35203

bias=.02 SE=.06

(c) Based on these results, is there evidence that the distribution of the software failure data is not an exponential distribution? Why or why not?

Based on the value of T and the small bias and SE it would appear that the software data is not an exponential distribution because T $\neq$ 1

In [5]:
```
#data_loc<-'/Users/Alexis/Documents/Spring2020/nonparametrics/data/software.csv'
data_loc<-'/Users/aporter1350/Documents/Courses/Spring2020/nonparametrics/data/software.csv'
software<-read.csv(data_loc, header=FALSE)
colnames(software) <- c("Minutes")
#Compute T
(mean(software$Minutes)/sd(software$Minutes))
#Bootstrap
set.seed(35203)
T=function(x,ind){(mean(x[ind])/sd(x[ind]))}
boot(software$Minutes, T, 100000)
```

0.639943891248242

```
ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = software$Minutes, statistic = T, R = 1e+05)


Bootstrap Statistics :
     original     bias    std. error
t1* 0.6399439 0.01701452   0.0605003
```

3.3 Consider i.i.d. random variables $Y_1, Y_2, \ldots . Y_n$ each distributed according to the distribution with frequency function $\theta(1-\theta)^2, x = 0, 1, 2, \ldots$ where $0 < \theta < 1$; note that this is a geometric distribution. The maximum likleihood estimate of θ is shown to be $1/1 + \bar{Y}$ where $\bar{Y} = 1/n \sum Y_j$

(a) Consider a set of observations

3 1 0 0 0 2 1 2 3 0 0 1

Corresponding to the random variables $Y_1, Y_2, \ldots . Y_n$ described above (for n=12). Based on these observations, find the maximum likelihood estimate of θ

MLE θ=.48

(b) Using the bootstrap method, find the bias and standard error of the maximum likelihood estimator; use 100000 bootstrap replications and set the random seed to 35204.

Bias=0.01 SE=.08

In [7]:
```
obs=c(3,1,0,0,0,2,1,2,3,0,0,1)
set.seed(35204)
1/(1+mean(obs))
theta=function(x,ind){1/(1+mean(x[ind]))}
boot(obs, theta, 100000)
```

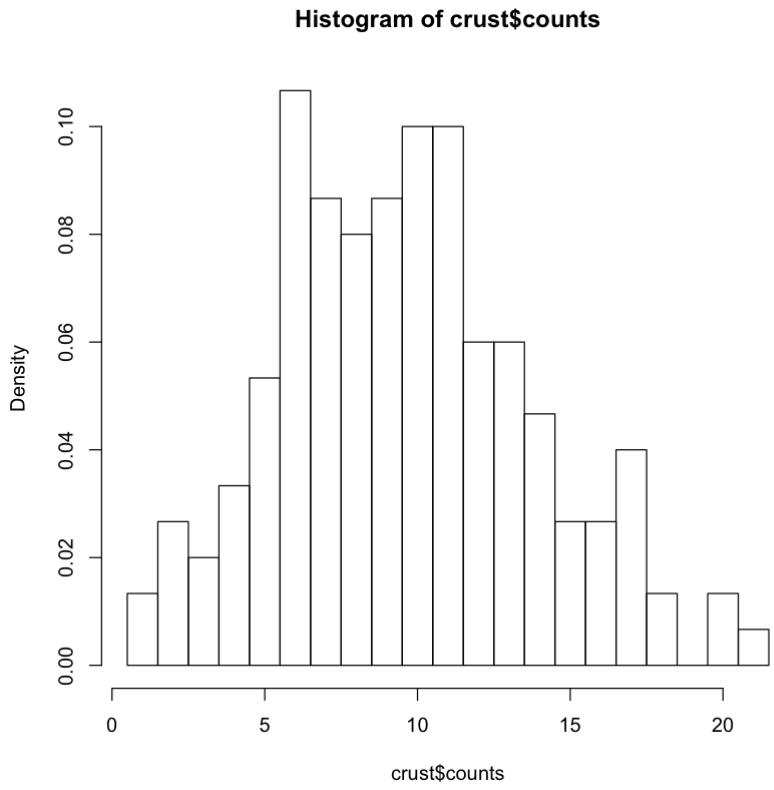
0.48

```
ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = obs, statistic = theta, R = 1e+05)


Bootstrap Statistics :
    original     bias    std. error
t1*     0.48 0.01220491  0.07976956
```

3.4 The dataset "crustaceans" contains the number of a certain type of crustacean found in a sample of marine plankton, for 150 such samples. Construct a histogram of these data. Choose the number of cells subjectively, with the goal of constructing histogram that provides useful information regarding the distribution of crustacean counts in samples of marine plankton

In [42]:
```
#data_loc<-'/Users/Alexis/Documents/Spring2020/nonparametrics/data/crustaceans.csv'
data_loc<-'/Users/aporter1350/Documents/Courses/Spring2020/nonparametrics/data/crustaceans.csv'
crust<-read.csv(data_loc, header=FALSE)
colnames(crust) <- c("counts")
#Determine breaks
table(crust$counts)/length(crust$counts)
hist(crust$counts, breaks=(0:21)+.5, freq=F)
```

```
          1           2           3           4           5           6 
0.013333333 0.026666667 0.020000000 0.033333333 0.053333333 0.106666667 
          7           8           9          10          11          12 
0.086666667 0.080000000 0.086666667 0.100000000 0.100000000 0.060000000 
         13          14          15          16          17          18 
0.060000000 0.046666667 0.026666667 0.026666667 0.040000000 0.013333333 
         20          21 
0.013333333 0.006666667 
```



3.5 Consider the failure data for software system

(a) Construct a histogram for the failure data using the default number of bins. Note that to use the default number of bins, simply omit the argument breaks when using the function hist

(b) Construct a histogram for the failure data using 5 bins

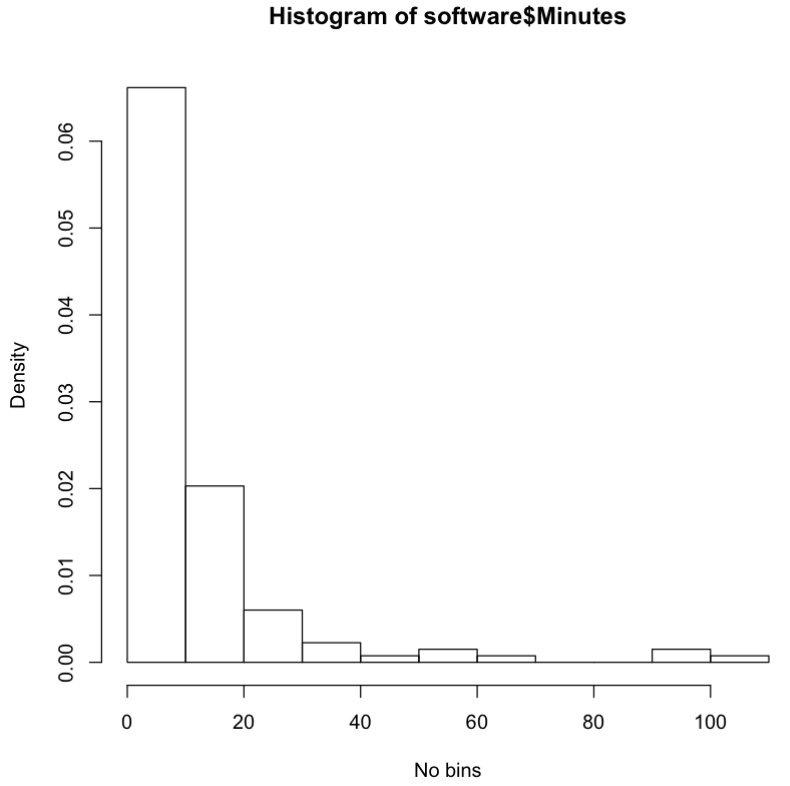
(c) Construct a histogram for the failure data using 15 bins

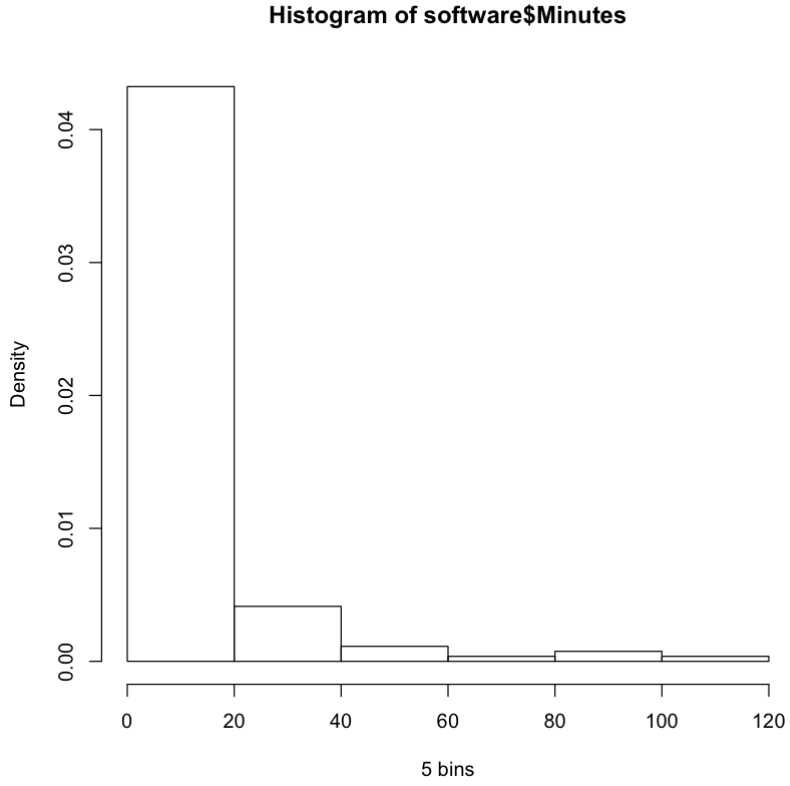
(d) Construct a histogram for the failure data using 20 bins

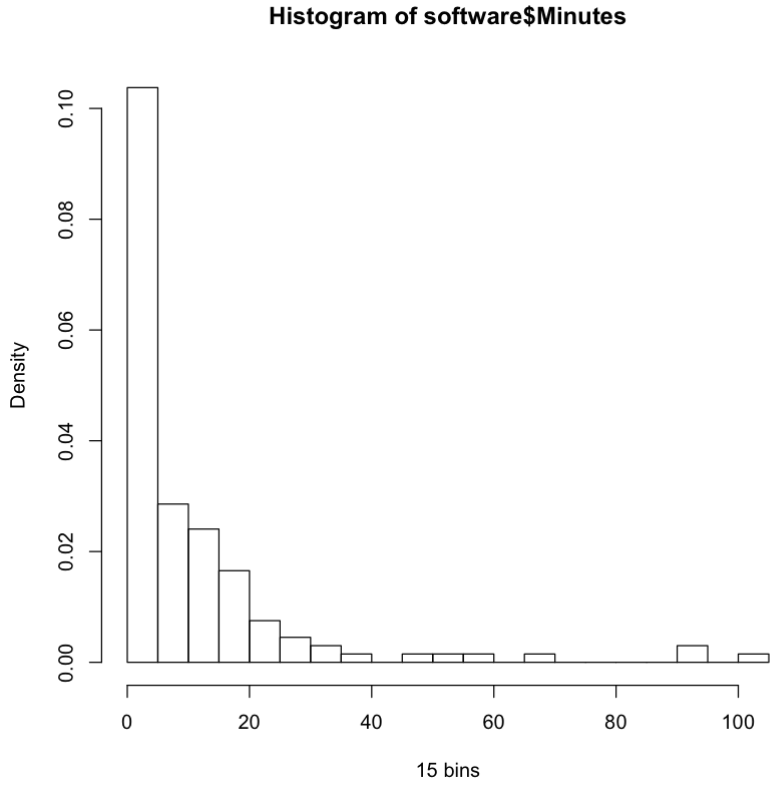
(e) Which of the three histograms constructed appears to be most useful for describing the distribution of the failure data?

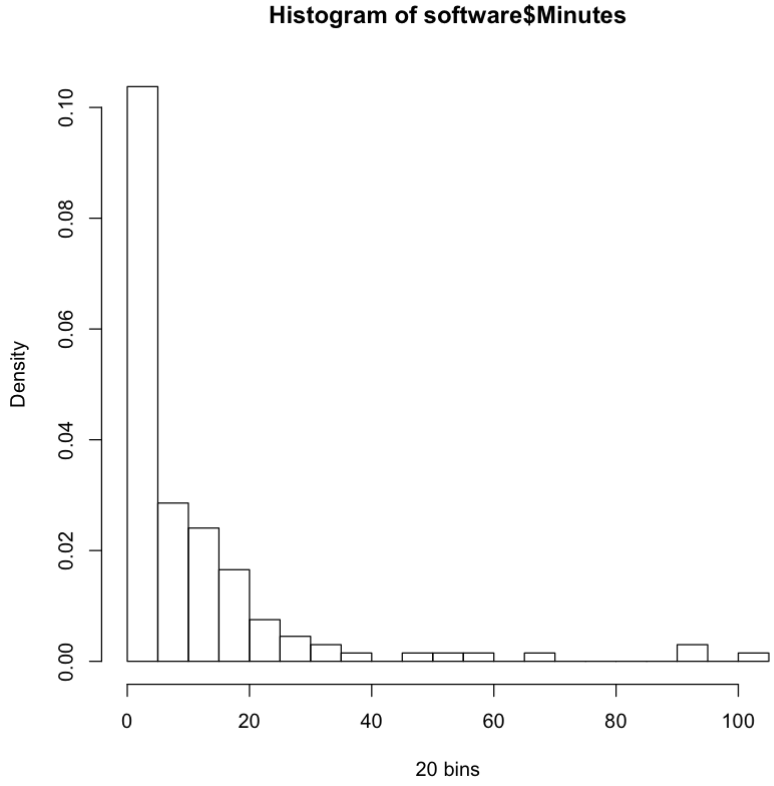
Based on the plots 15 bins appears to be the most informative of the skew and frequency of the dataset

In [81]:
```
library('patchwork')
a=hist(software$Minutes,freq=F, xlab='No bins')
b=hist(software$Minutes,breaks=5,freq=F, xlab='5 bins')
c=hist(software$Minutes,breaks=15,freq=F, xlab='15 bins')
d=hist(software$Minutes,breaks=20,freq=F, xlab='20 bins')
```


Histogram of software$Minutes
Density
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0
20
40
60
80
100
No bins


Histogram of software$Minutes
Density
0.00
0.01
0.02
0.03
0.04
0
20
40
60
80
100
120
5 bins


Histogram of software$Minutes
Density
0.00
0.02
0.04
0.06
0.08
0.10
0
20
40
60
80
100
15 bins


Histogram of software$Minutes
Density
0.00
0.02
0.04
0.06
0.08
0.10
0
20
40
60
80
100
20 bins

3.6 The purpose of this exercise is to study the relationship between the number of bins, the accuracy of a histogram estimator, and the sample size

(a) Plot the density function of the chi-squared distribution with 3 degrees of freedom. Evaluate the density function at the values in seq(0,12,.001)

(b) Generate a sample of size 100 of random variates from a chi squared distribution with 3 df. Use the random seed 35206

(c) Construct a histogram of these data with breaks taken to be 5 and save the results to a varialbe called hout

(d) Let mid denot the component mid of the histogram result from part (a) and let den denote the density estimate given in component density. Then den contains a vector of estimates of the density function of the exponential distribution with rate parameter 1 evaluated at the values in mid. Therefore, the values in den are estimates of the values in dchisq(mid, df=3). Hence mean((den-dchisq(mid, df=3))^2)^.5 is the square root of the average squared error of the histogram density estimate, where the average is over the values in mid. Compute this value for the histogram estimate computed in part (b)

.02

(e) Repeat parts (c) and (d) twice, with the value of breaks taken to 10 and 20, respectively

(f) Based on these results, which value of breaks produces the most accurate density estimate?

Based on the bias variance tradeoff we would want a small h but not too small based on all of these bins I would select the 5 bins to have a small bias because it is the closest to zero.

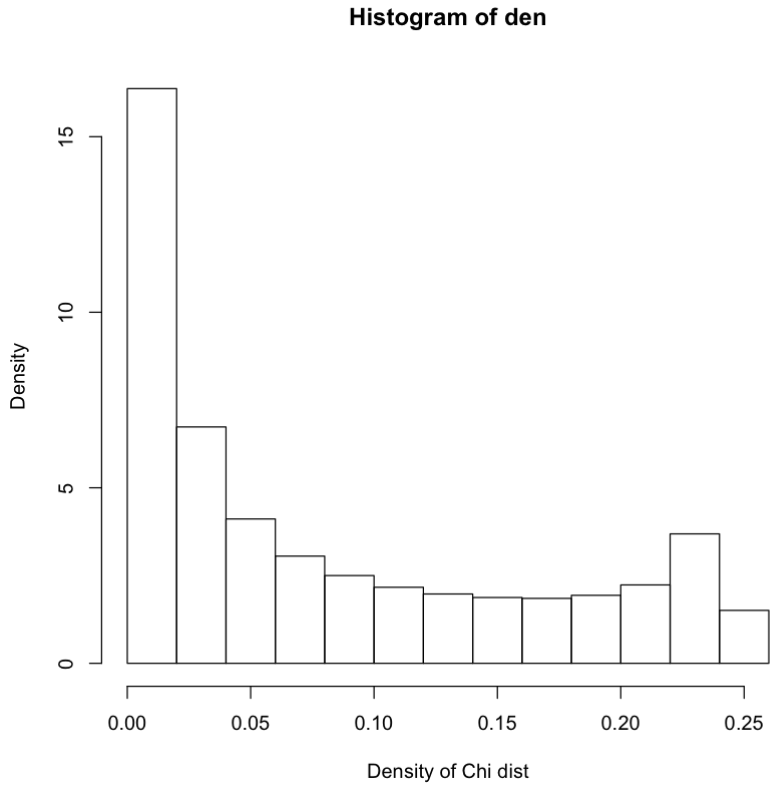
(g) Repeat parts (b)-(f) for a sample size of 500, use the same random seed 35206

(h) Repeat parts (b)-(f) for a sample size of 1000, use the same random seed 35206

(i) Based on these results, what do you conclude about the relationship between the number of bins, the accuracy of a histogram estimator, and the sample size. Does this conclusion agree with the theoretical results given in the notes?

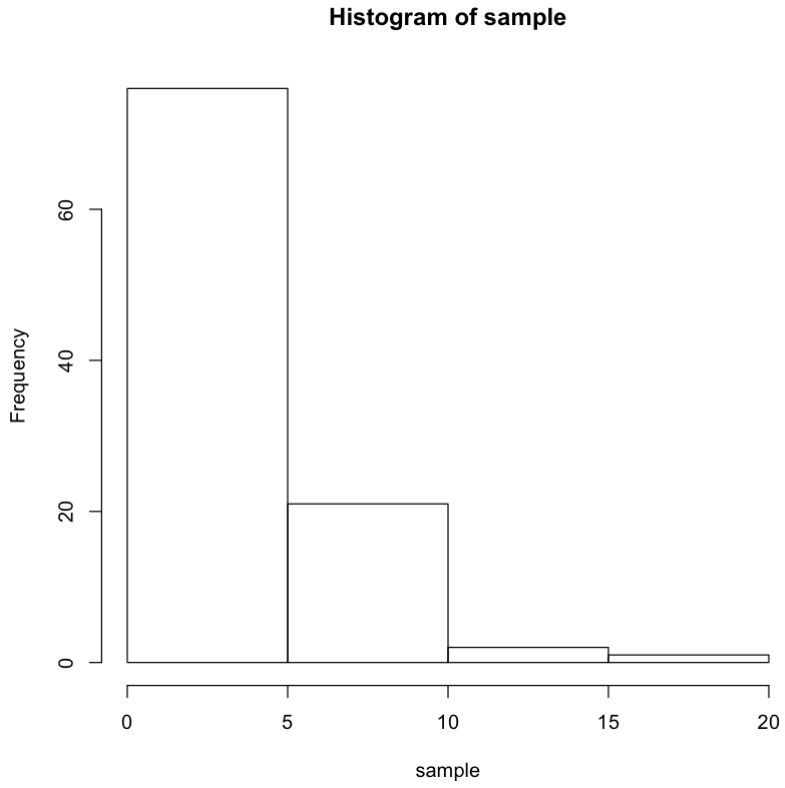
Yes based on the idea that as sample size is large, h will be smaller. As n increased the number of bins should increase to account for the variance.

In [98]:
```
y<-seq(0,12,.001)
den=dchisq(y,df=3)
hist(den,freq=F, xlab='Density of Chi dist')
```



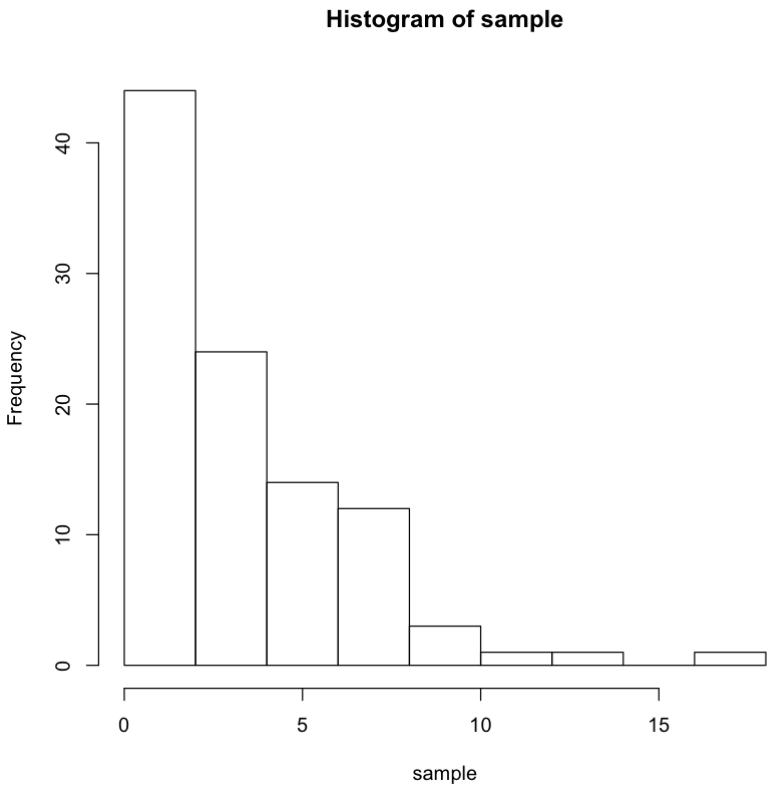
In [102]:
```
#Generate sample
set.seed(35206)
sample=rchisq(100, df=3)
hout=hist(sample, breaks=5)
#Compute sqrt of avg sq error of hist
h=mean((hout$density-dchisq(hout$mid, df=3))^2)^.5

#h*100
```

1.65491365252845

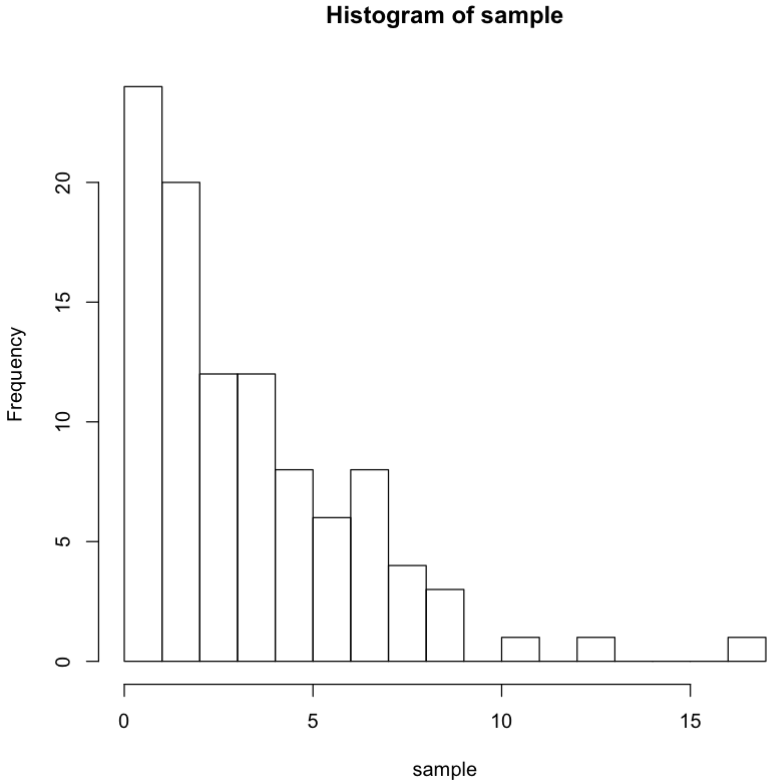


In [104]:
```
#10 bins
ten=hist(sample, breaks=10)
#Compute sqrt of avg sq error of hist
hten=mean((ten$density-dchisq(ten$mid, df=3))^2)^.5

#hten*100
#20 bins
twenty=hist(sample, breaks=20)
#Compute sqrt of avg sq error of hist
htwen=mean((twenty$density-dchisq(twenty$mid, df=3))^2)^.5

#htwen*100
```

1.66203724971844

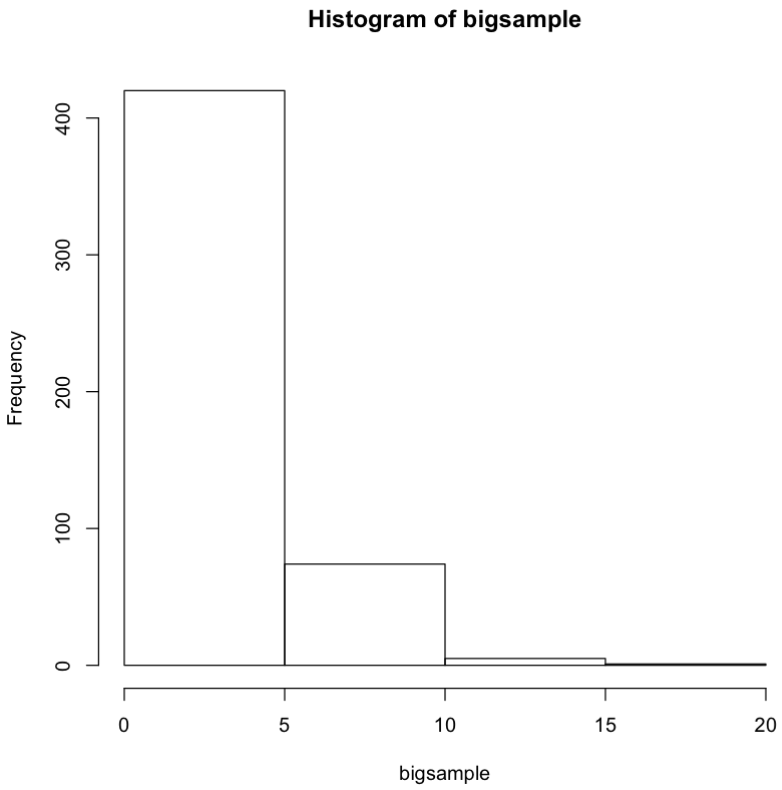


2.10710459535054

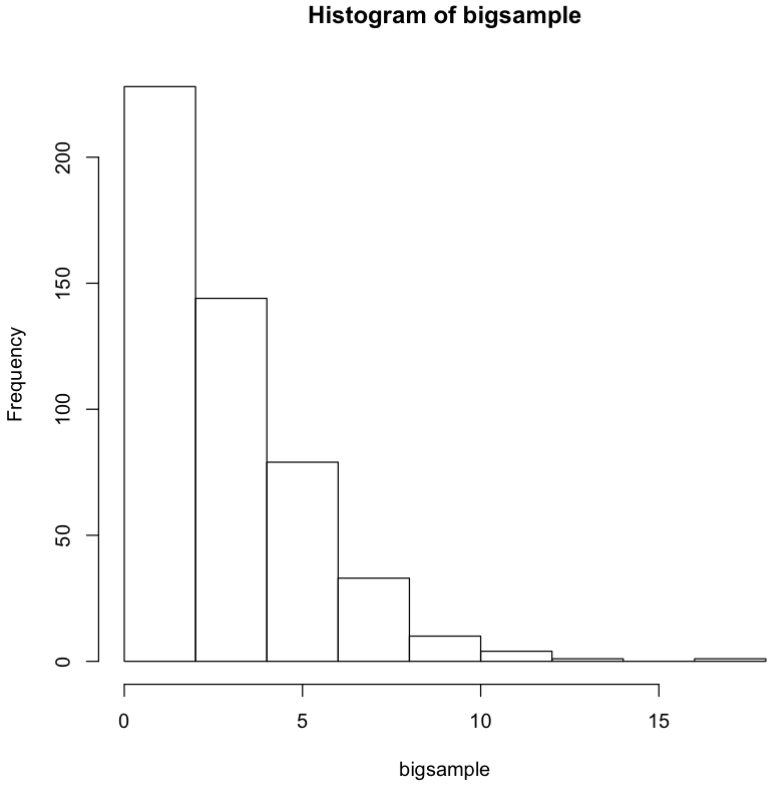


In [105]:
```
#Generate sample
set.seed(35206)
bigsample=rchisq(500, df=3)
hout=hist(bigsample, breaks=5)
#Compute sqrt of avg sq error of hist
h=mean((hout$density-dchisq(hout$mid, df=3))^2)^.5
h*100
#10 bins
ten=hist(bigsample, breaks=10)
#Compute sqrt of avg sq error of hist
hten=mean((ten$density-dchisq(ten$mid, df=3))^2)^.5

hten*100
#20 bins
twenty=hist(bigsample, breaks=20)
#Compute sqrt of avg sq error of hist
htwen=mean((twenty$density-dchisq(twenty$mid, df=3))^2)^.5

htwen*100
```

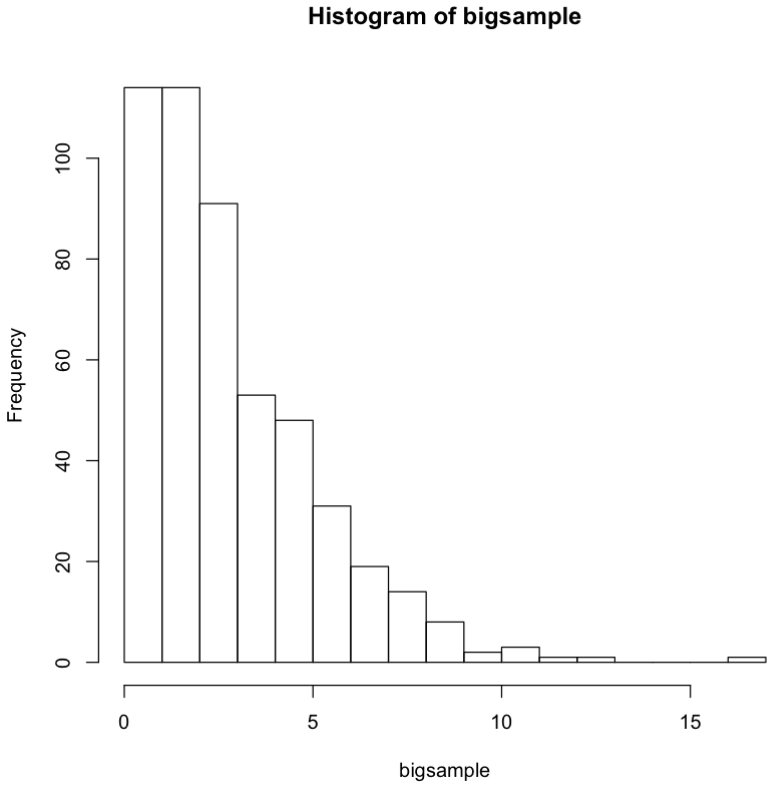
0.666436127354777



0.622545782007763

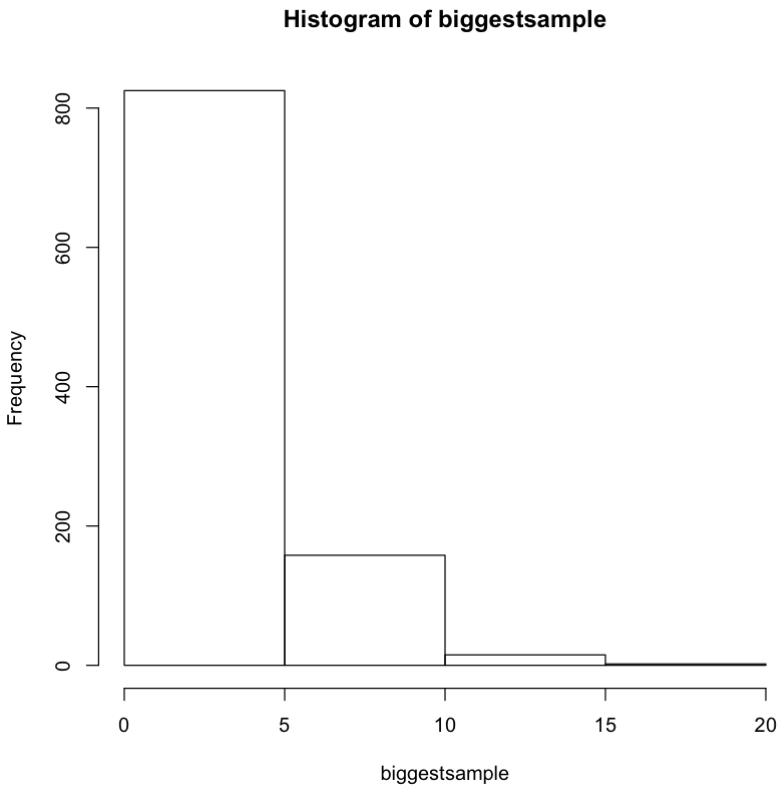


0.667008767458126

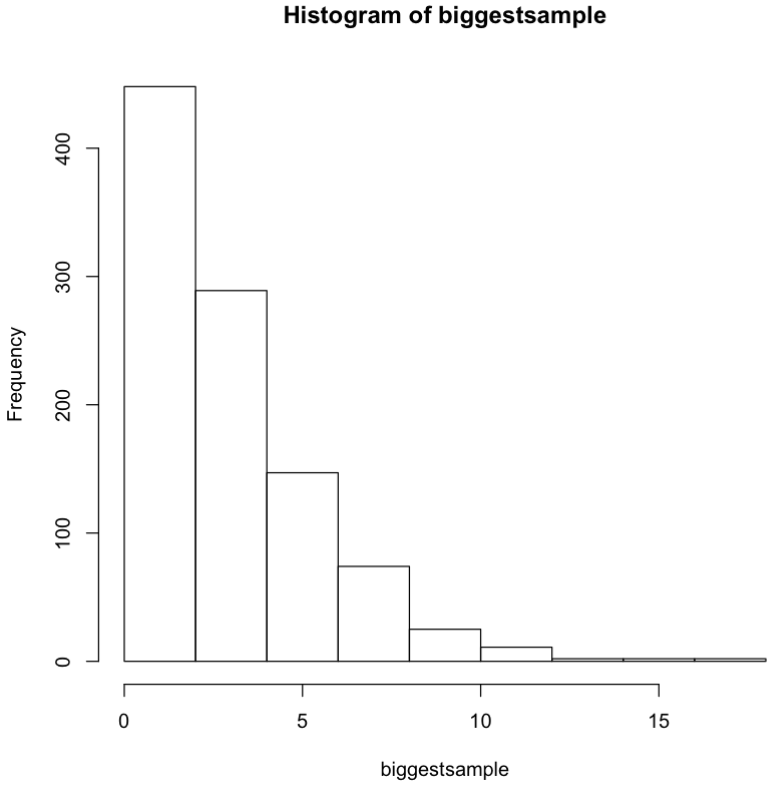

Histogram of bigsample
Frequency
0
20
40
60
80
100
0
5
10
15
bigsample

In [106]:
```
#Generate sample
set.seed(35206)
biggestsample=rchisq(1000, df=3)
hout=hist(biggestsample, breaks=5)
#Compute sqrt of avg sq error of hist
h=mean((hout$density-dchisq(hout$mid, df=3))^2)^.5
h*100
#10 bins
ten=hist(biggestsample, breaks=10)
#Compute sqrt of avg sq error of hist
hten=mean((ten$density-dchisq(ten$mid, df=3))^2)^.5

hten*100
#20 bins
twenty=hist(biggestsample, breaks=20)
#Compute sqrt of avg sq error of hist
htwen=mean((twenty$density-dchisq(twenty$mid, df=3))^2)^.5

htwen*100
```

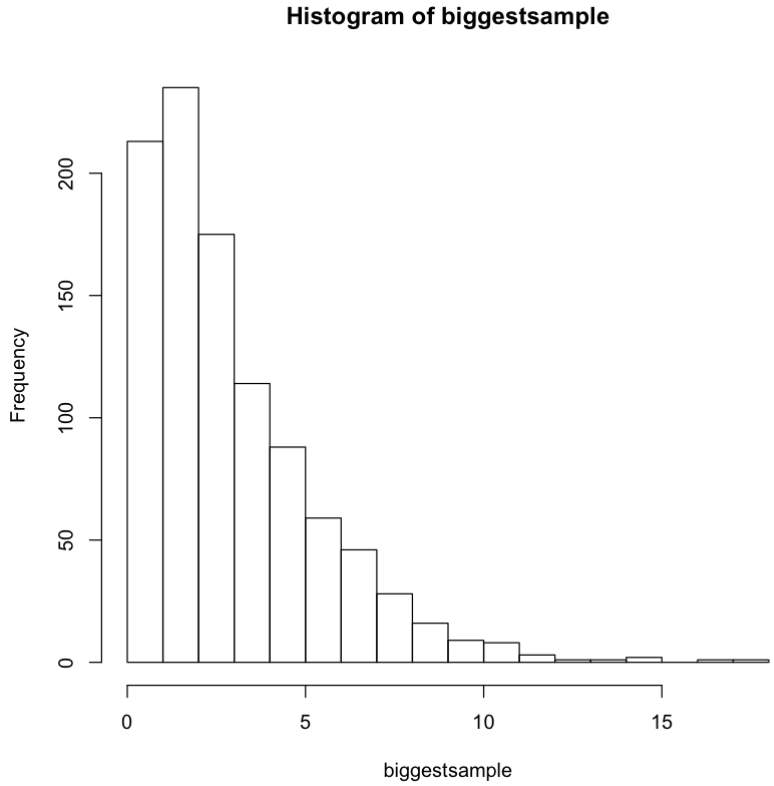
0.839890446526323



0.703538856424166



0.473732555208228


Histogram of biggestsample
Frequency
0
50
100
150
200
0
5
10
15
biggestsample

In [ ]: