```r
# exercise1.r

set.seed(1)
# create the population
p = 0.30
N = 1000
population = rbinom(N,1,p)
table(population)
```

```
## population
##   0   1
## 696 304
```

```r
# collect a sample
n = 50
# row numbers
id = sample(1:N,n)
# row values (0 or 1)
obs = population[id]

# CI from the sample
alpha = 0.05
z_alpha = qnorm(1-alpha/2)
z_alpha
```

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

```r
# find phat and its standard deviation
table(obs)
```

```
## obs
##  0  1
## 30 20
```

```r
phat = 20/50

sdev = sqrt(phat*(1-phat)/n)
sdev
```

```
## [1] 0.06928203
```

```r
# the confidence interval
lb = phat - z_alpha*sdev
ub = phat + z_alpha*sdev
c(lb,ub)
```

```
## [1] 0.2642097 0.5357903
```

```r
# 0.2642097 0.5357903
# this CI covers p = 0.30

# using binom.test
binom.test(20,50,0.30)
```

```
##
##  Exact binomial test
##
## data:  20 and 50
## number of successes = 20, number of trials = 50, p-value = 0.125
## alternative hypothesis: true probability of success is not equal to 0.3
## 95 percent confidence interval:
```

```
##  0.2640784 0.5482060
## sample estimates:
## probability of success
##                    0.4
```

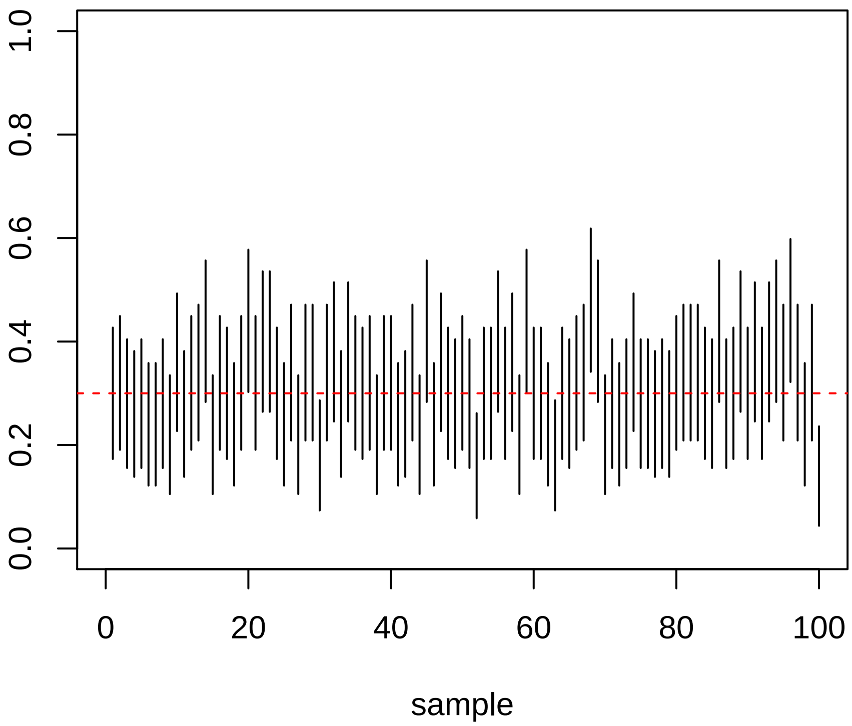
```r
binom.test(20,50,0.30)$conf.int
```

```
## [1] 0.2640784 0.5482060
## attr(,"conf.level")
## [1] 0.95
```

```r
# 0.2640784 0.5482060

# confidence intervals found by hand and by using binom.test agree

# Many confidence intervals
M = 100
lb = rep(0,M)
ub = rep(0,M)
phat = rep(0,M)

for(i in 1:M)
{
  id = sample(1:N,n)
  obs = population[id]
  ob_successes = table(obs)[2]
  phat[i] = ob_successes/n
  sdev = sqrt(phat[i]*(1-phat[i])/n)
  lb[i] = phat[i] - z_alpha*sdev
  ub[i] = phat[i] + z_alpha*sdev
}

i = 1:M
d1 = data.frame(i,lb,ub)
head(d1)
```

```
##   i        lb        ub
## 1 1 0.1729798 0.4270202
## 2 2 0.1907018 0.4492982
## 3 3 0.1555461 0.4044539
## 4 4 0.1384190 0.3815810
## 5 5 0.1555461 0.4044539
## 6 6 0.1216208 0.3583792
```

```r
# plot CIs
plot(ub~i,xlim=c(0,M),ylim=c(0,1),ylab='',xlab='sample',type = 'n')
segments(i,lb,i,ub)
abline(h=0.30,col='red',lty=2)
```




```r
# show CIs that fail to cover p=0.3
color = rep('black',M)
# rows of CIs that fail to cover p
idx = NULL

for(i in 1:M)
{
  if(0.3 < lb[i] | 0.3 > ub[i])
  {
    color[i] = 'red'
    idx = c(idx,i)
  }
}

table(color)
```

```
## color
## black   red
##    92     8
```

```r
# there are 8 red colored CIs

# CIs that failed are in rows
idx
```

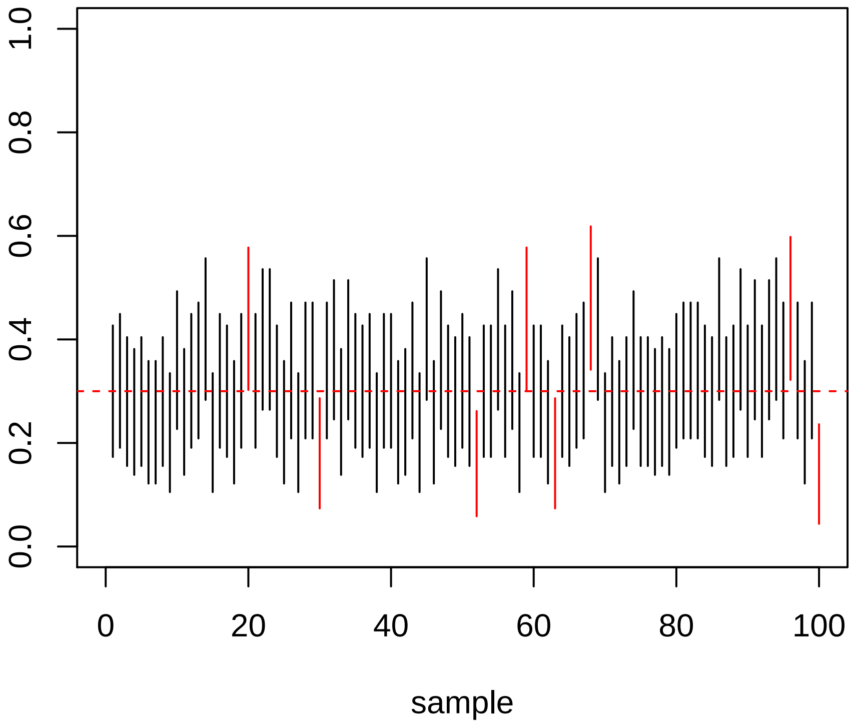
```
## [1]  20  30  52  59  63  68  96 100
```

```r
# see the CIs that failed
d1[idx,]
```

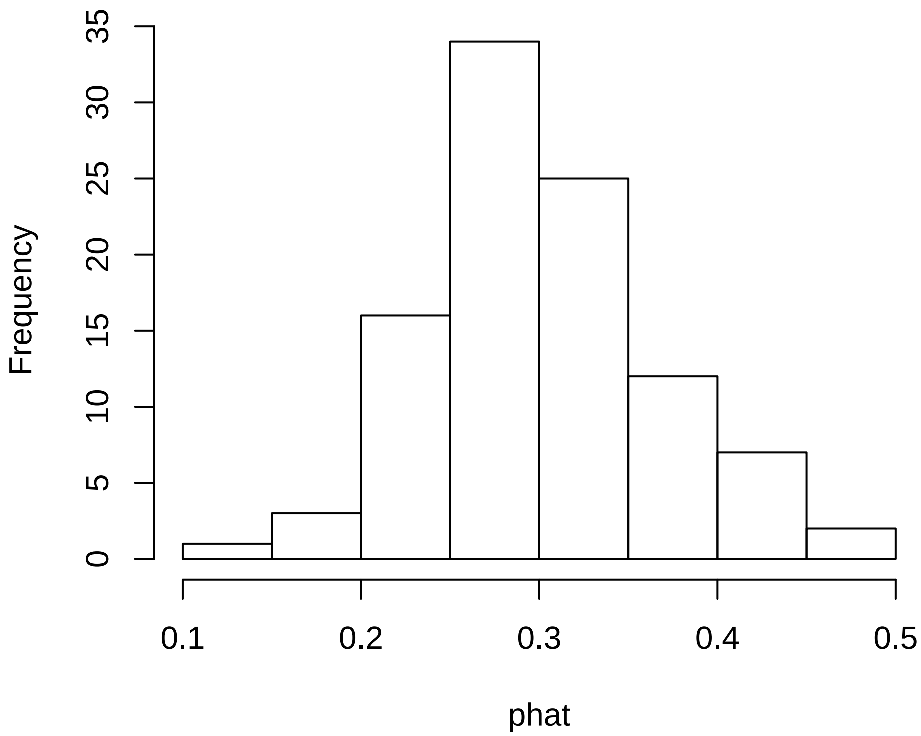
```
##       i         lb        ub
## 20   20 0.30241109 0.5775889
## 30   30 0.07351063 0.2864894
## 52   52 0.05838385 0.2616161
## 59   59 0.30241109 0.5775889
## 63   63 0.07351063 0.2864894
```

```
## 68   68 0.34152053 0.6184795
## 96   96 0.32185382 0.5981462
## 100 100 0.04382187 0.2361781
```

```r
# plot CIs with those that failed
i = 1:M
plot(ub~i,xlim=c(0,M),ylim=c(0,1),ylab='',xlab='sample',type = 'n')
segments(i,lb,i,ub,col=color)
abline(h=0.30,col='red',lty=2)
```

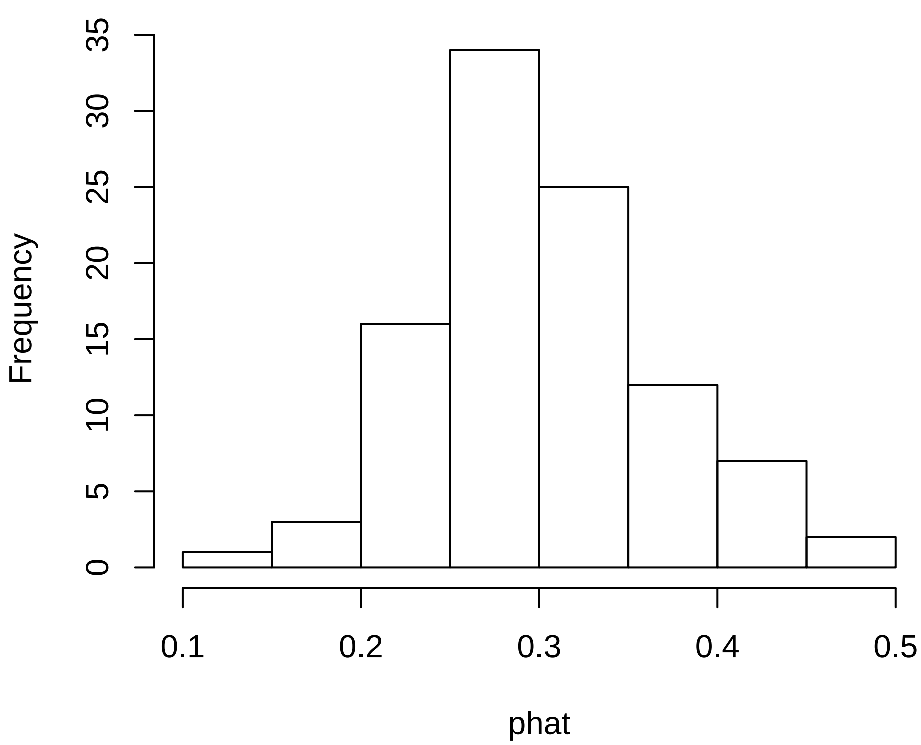



```r
# histogram of phat values
hist(phat,main='')
```




```r
# to center the histogram
h1 = hist(phat)
```

Histogram of phat



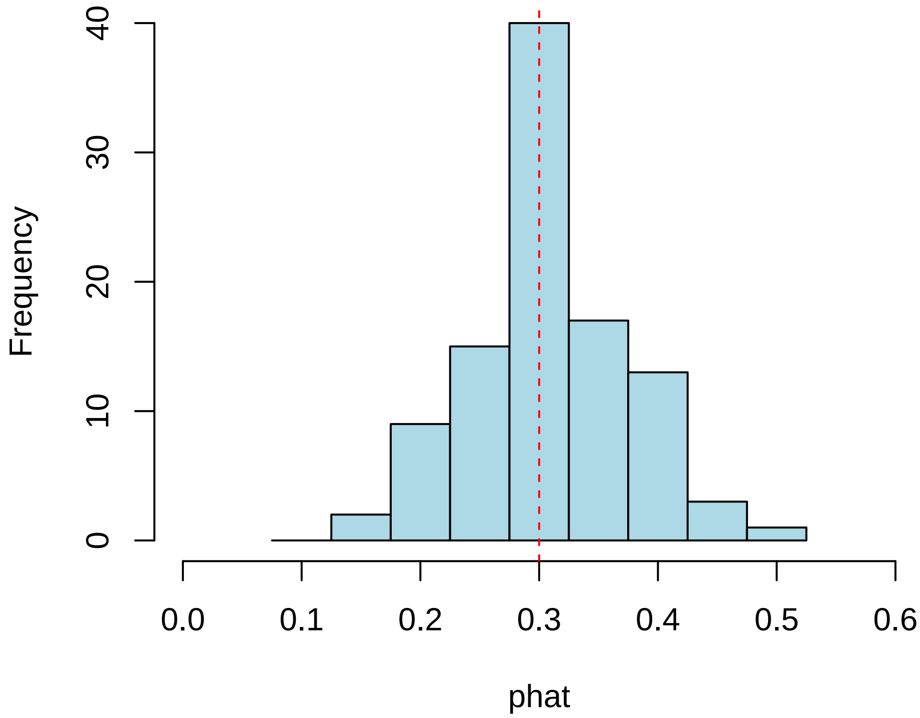

```r
h1$breaks
```

```
## [1] 0.10 0.15 0.20 0.25 0.30 0.35 0.40 0.45 0.50
```

```r
# change the breaks
aux = seq(0.075,0.525,0.05)
aux
```

```
##  [1] 0.075 0.125 0.175 0.225 0.275 0.325 0.375 0.425 0.475 0.525
```

```r
hist(phat,breaks = aux, col='lightblue', xlim = c(0,0.6),main='')
abline(v=0.3,col='red',lty=2)
```




```r
summary(phat)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.140   0.260   0.300   0.308   0.340   0.480
```