exercise1

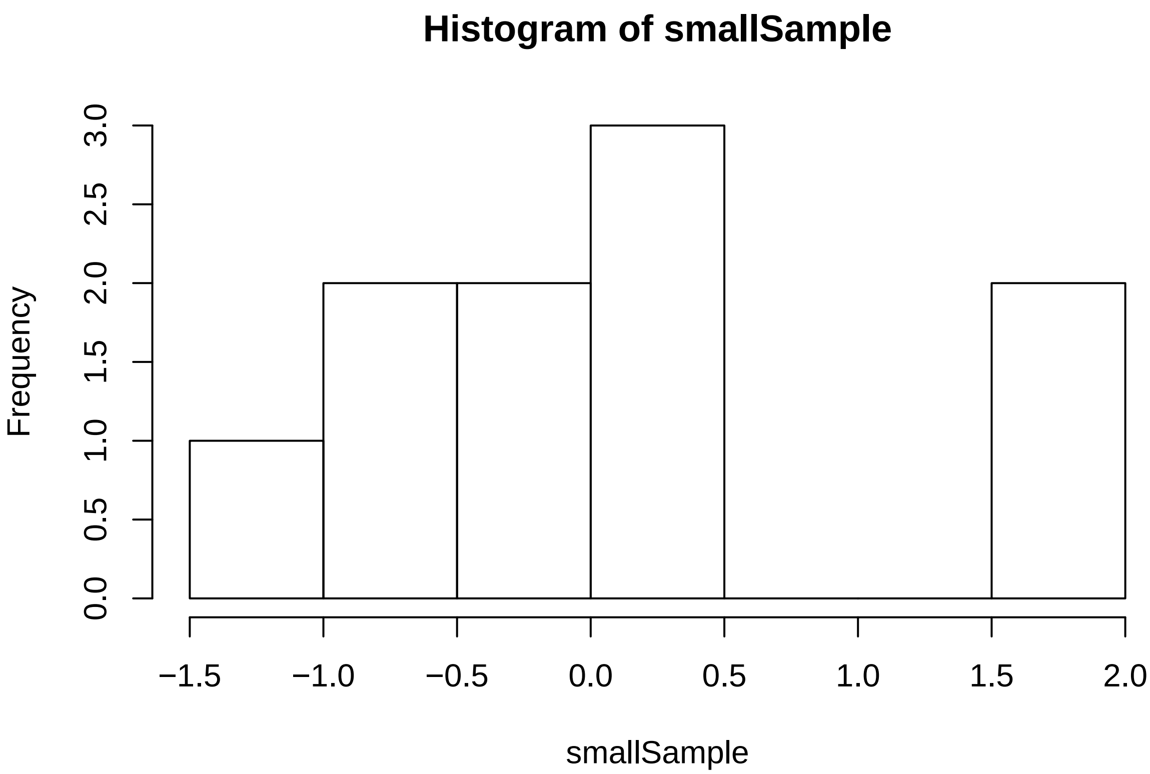
Alexande Van Roijen

January 9, 2019

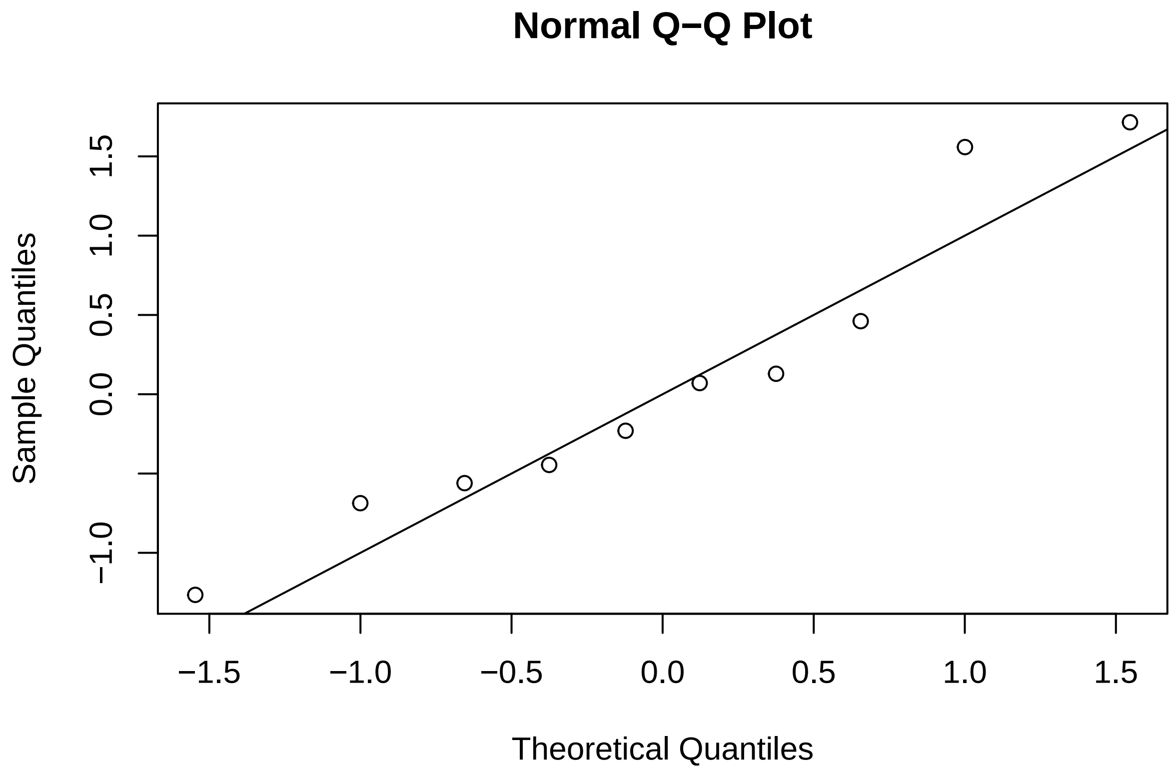
R Markdown

Problem 1

```r
set.seed(123)
smallSample = (rnorm(10,0,1))
hist(smallSample)
```

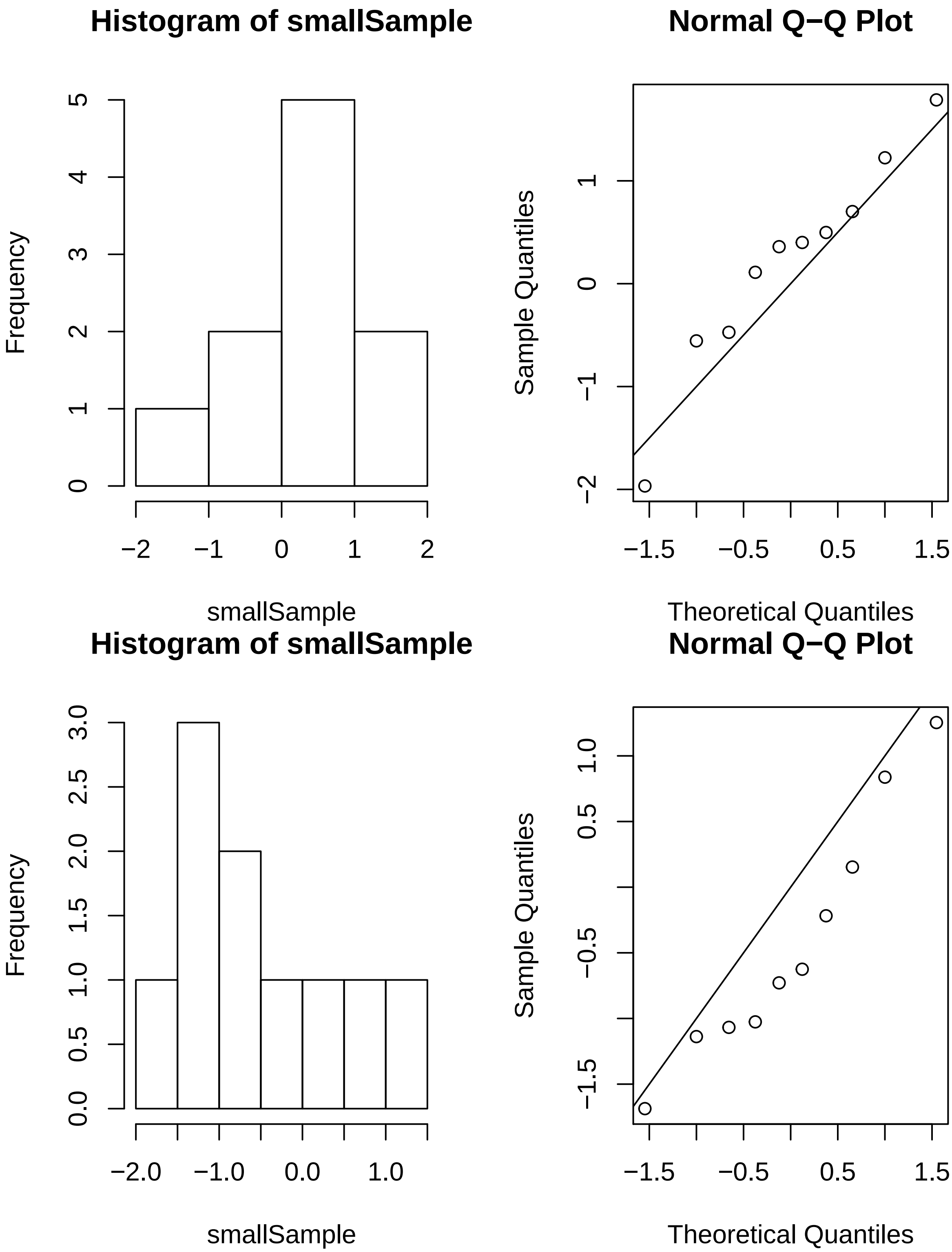



```r
qqnorm(smallSample)
abline(0,1)
```

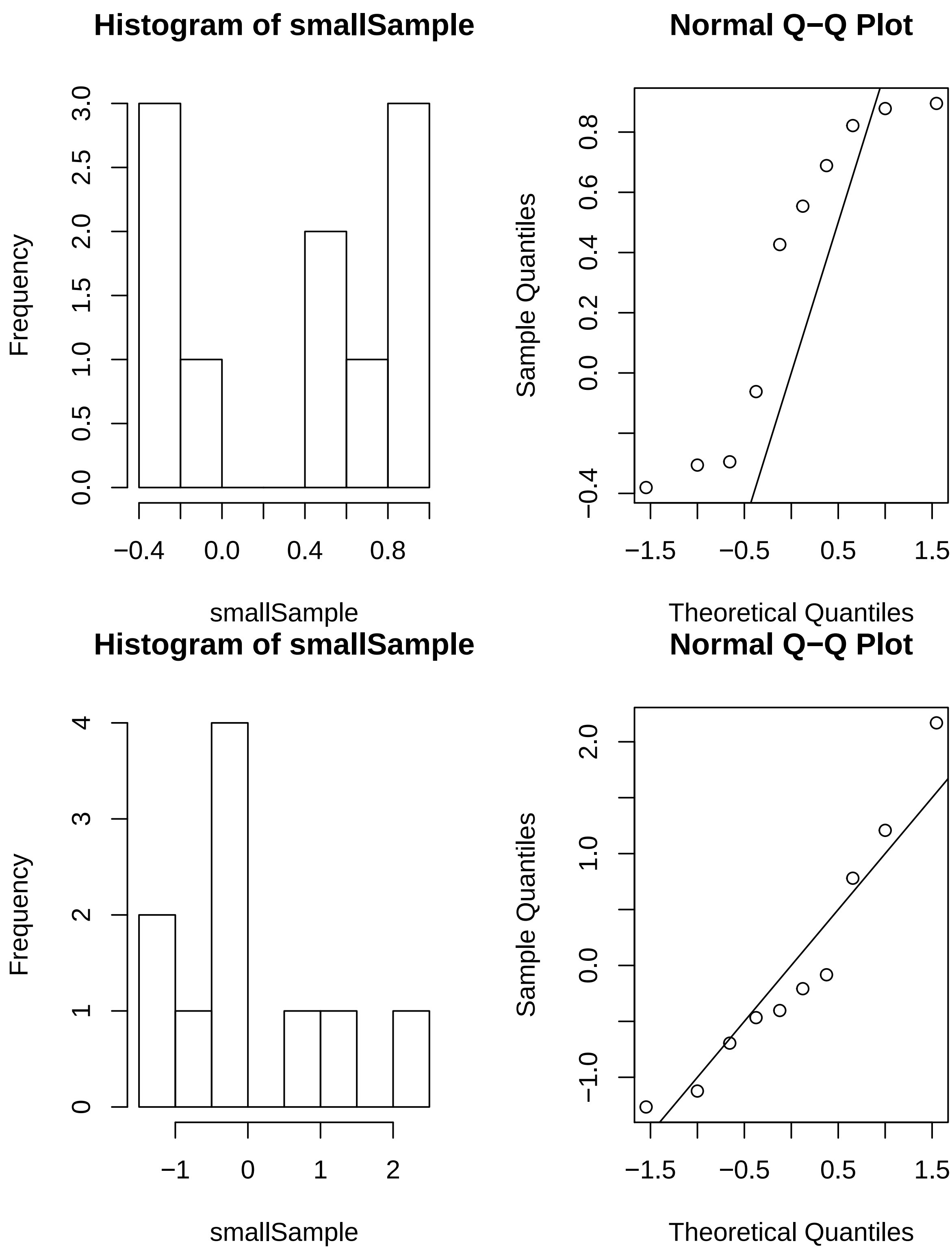


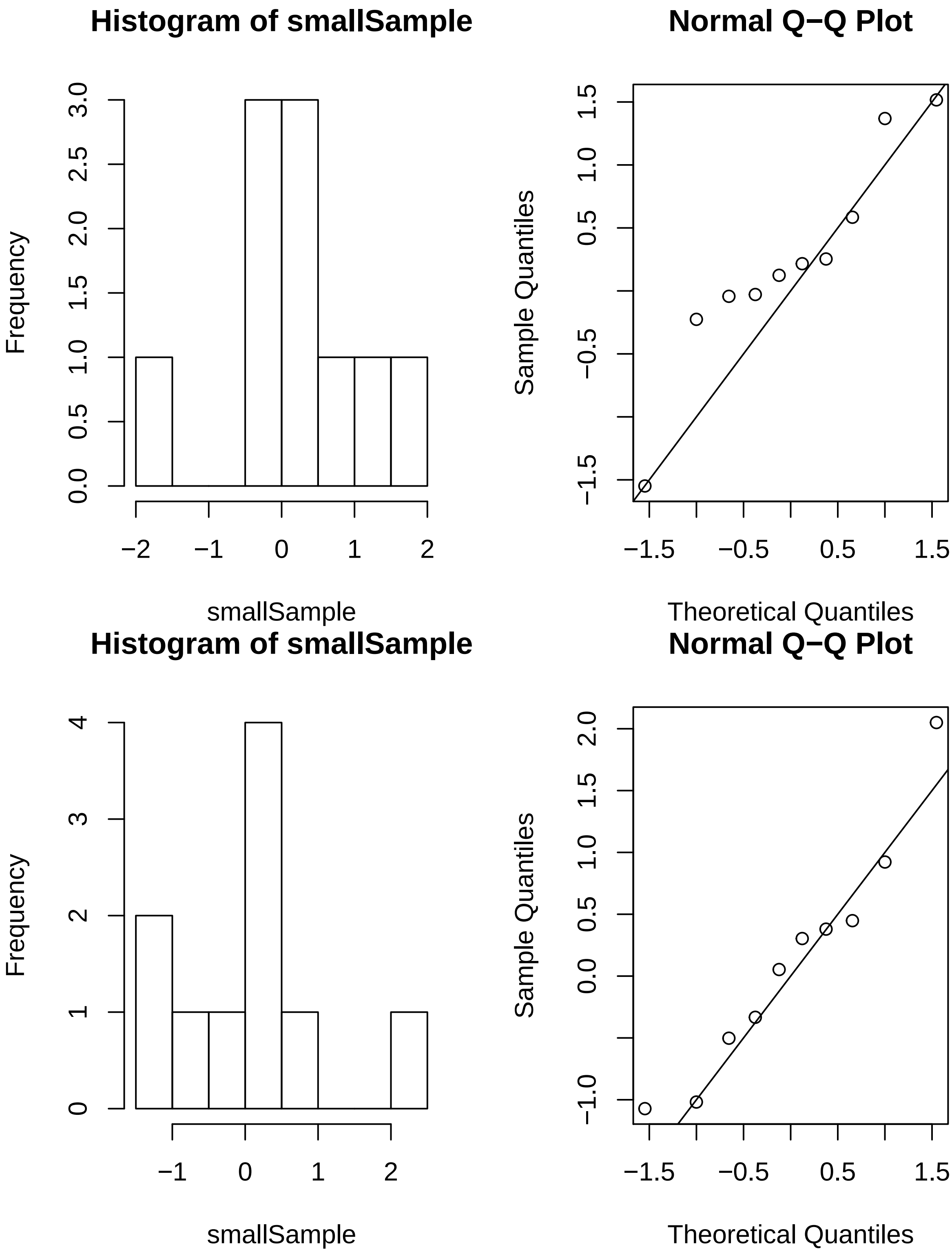
surprisingly, this does look rather normal! or at least the qqplot says so. The histogram does not look normal, more uniform if anything.

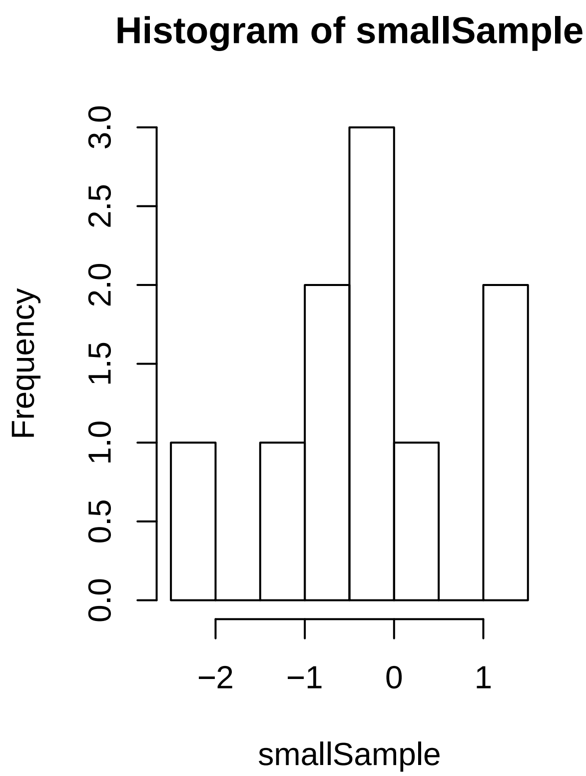
```
for(i in 1:20){
  smallSample = (rnorm(10,0,1))
  par(mfrow=c(1,2))
  hist(smallSample)
  qqnorm(smallSample)
  abline(0,1)
}
```

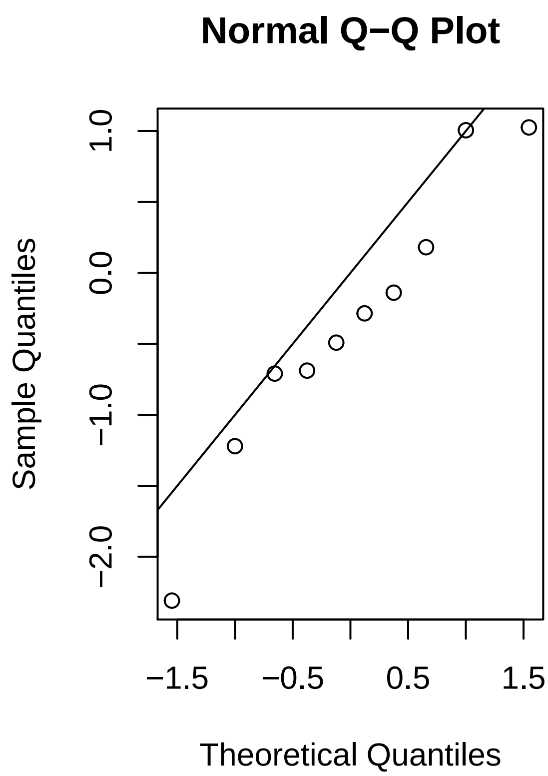


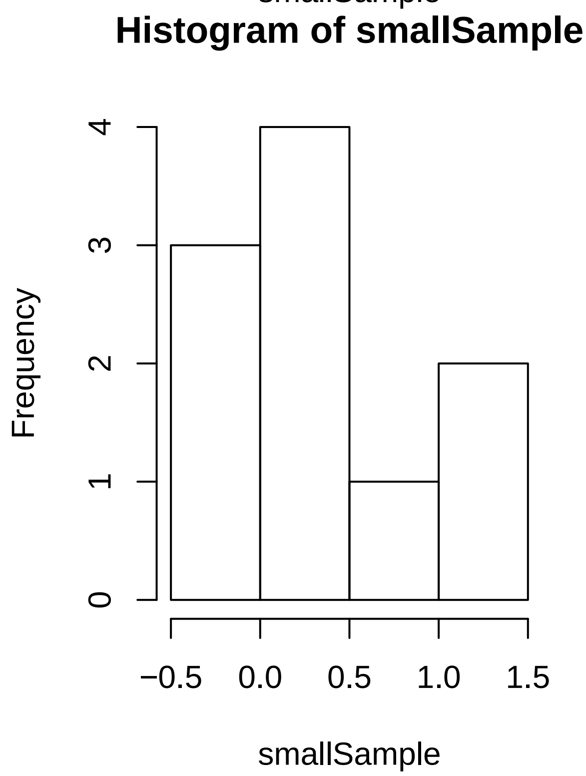
Histogram of smallSample
Frequency
0
1
2
3
4
5
−2
−1
0
1
2
smallSample
Normal Q−Q Plot
Sample Quantiles
−2
−1
0
1
−1.5
−0.5
0.5
1.5
Theoretical Quantiles
Histogram of smallSample
Frequency
0.0
0.5
1.0
1.5
2.0
2.5
3.0
−2.0
−1.0
0.0
1.0
smallSample
Normal Q−Q Plot
Sample Quantiles
−1.5
−0.5
0.5
1.0
−1.5
−0.5
0.5
1.5
Theoretical Quantiles

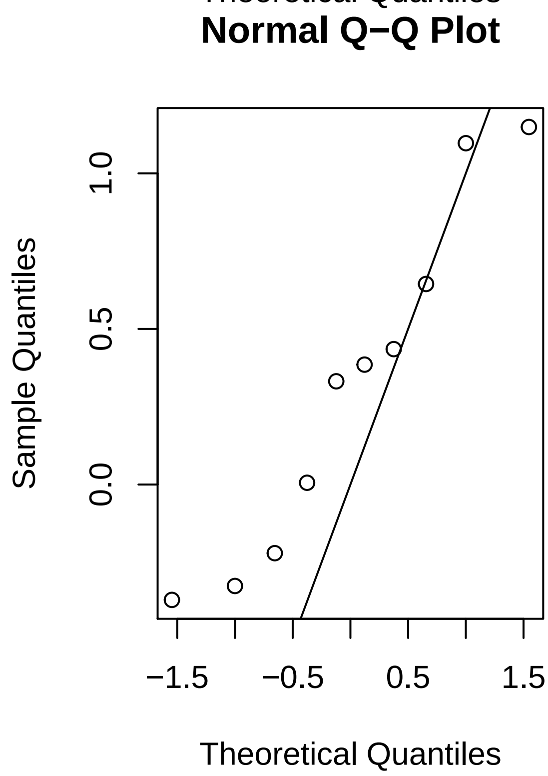

Histogram of smallSample
Frequency
0.0
0.5
1.0
1.5
2.0
2.5
3.0
−0.4
0.0
0.4
0.8
smallSample
Normal Q−Q Plot
Sample Quantiles
−0.4
0.0
0.2
0.4
0.6
0.8
−1.5
−0.5
0.5
1.5
Theoretical Quantiles
Histogram of smallSample
Frequency
0
1
2
3
4
−1
0
1
2
smallSample
Normal Q−Q Plot
Sample Quantiles
−1.0
0.0
1.0
2.0
−1.5
−0.5
0.5
1.5
Theoretical Quantiles

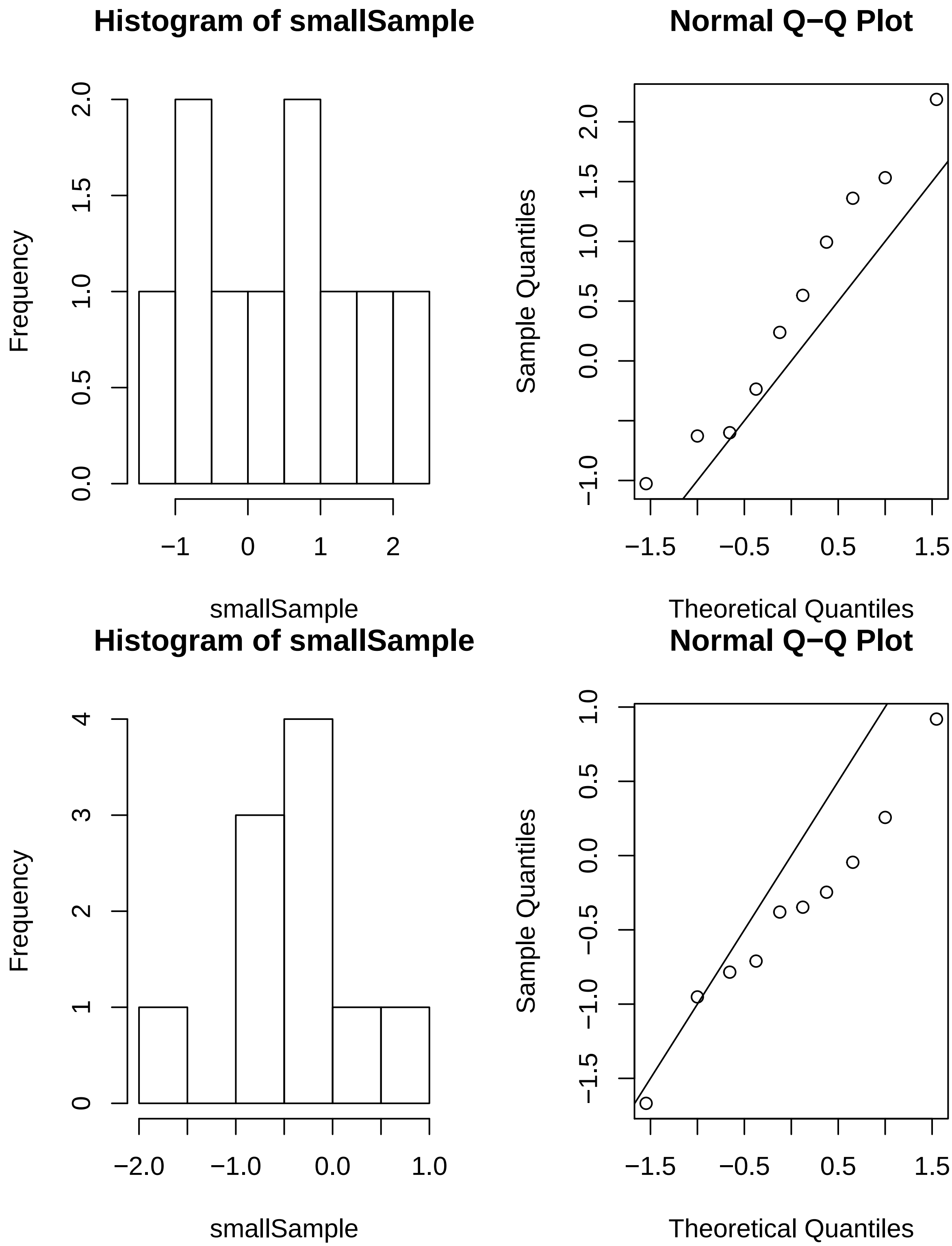

Histogram of smallSample
Frequency
0.0 0.5 1.0 1.5 2.0 2.5 3.0
−2 −1 0 1 2
smallSample
Normal Q−Q Plot
Sample Quantiles
−1.5 −0.5 0.5 1.0 1.5
−1.5 −0.5 0.5 1.5
Theoretical Quantiles
Histogram of smallSample
Frequency
0 1 2 3 4
−1 0 1 2
smallSample
Normal Q−Q Plot
Sample Quantiles
−1.0 0.0 0.5 1.0 1.5 2.0
−1.5 −0.5 0.5 1.5
Theoretical Quantiles

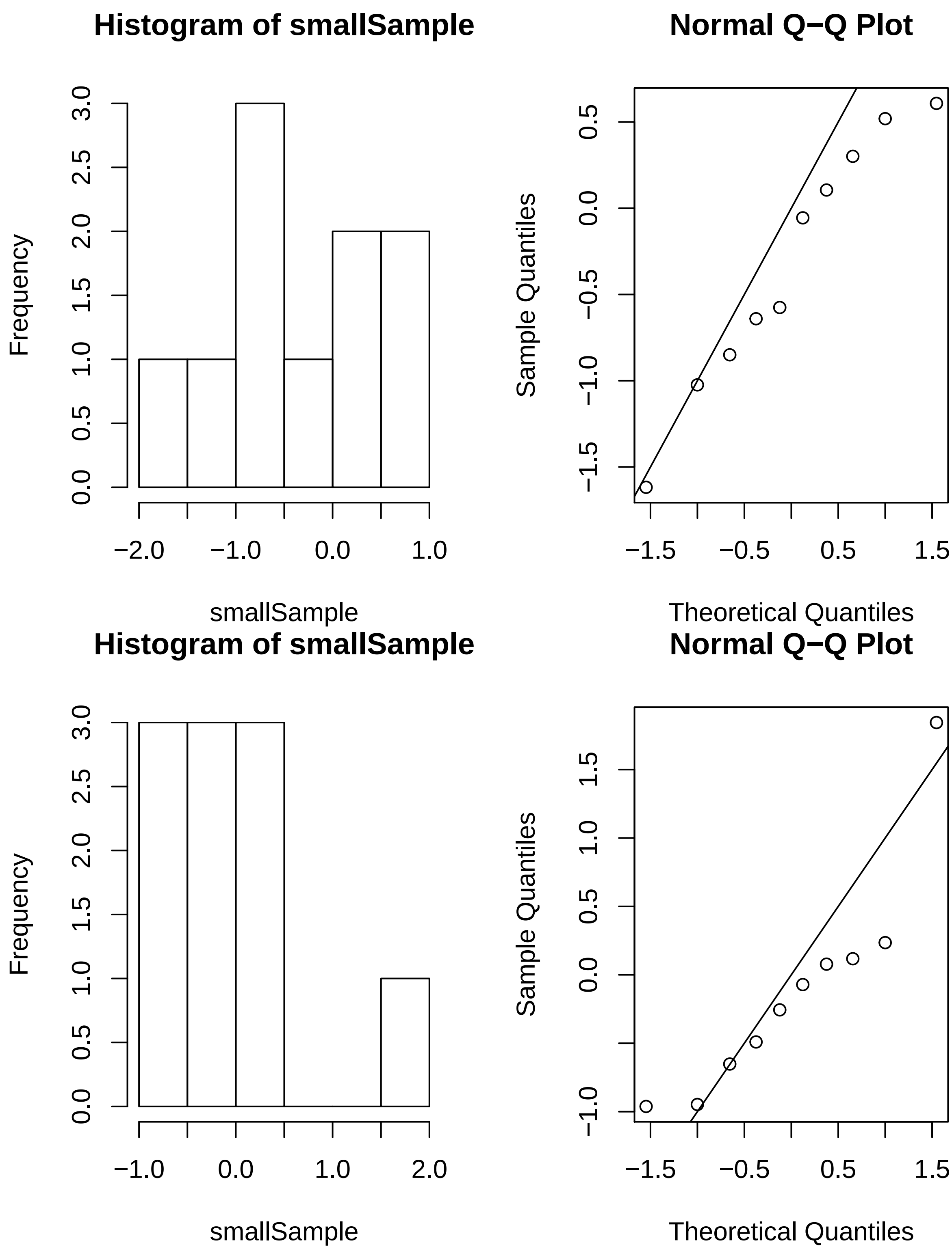

Histogram of smallSample
Frequency
0.0
0.5
1.0
1.5
2.0
2.5
3.0
−2
−1
0
1
smallSample

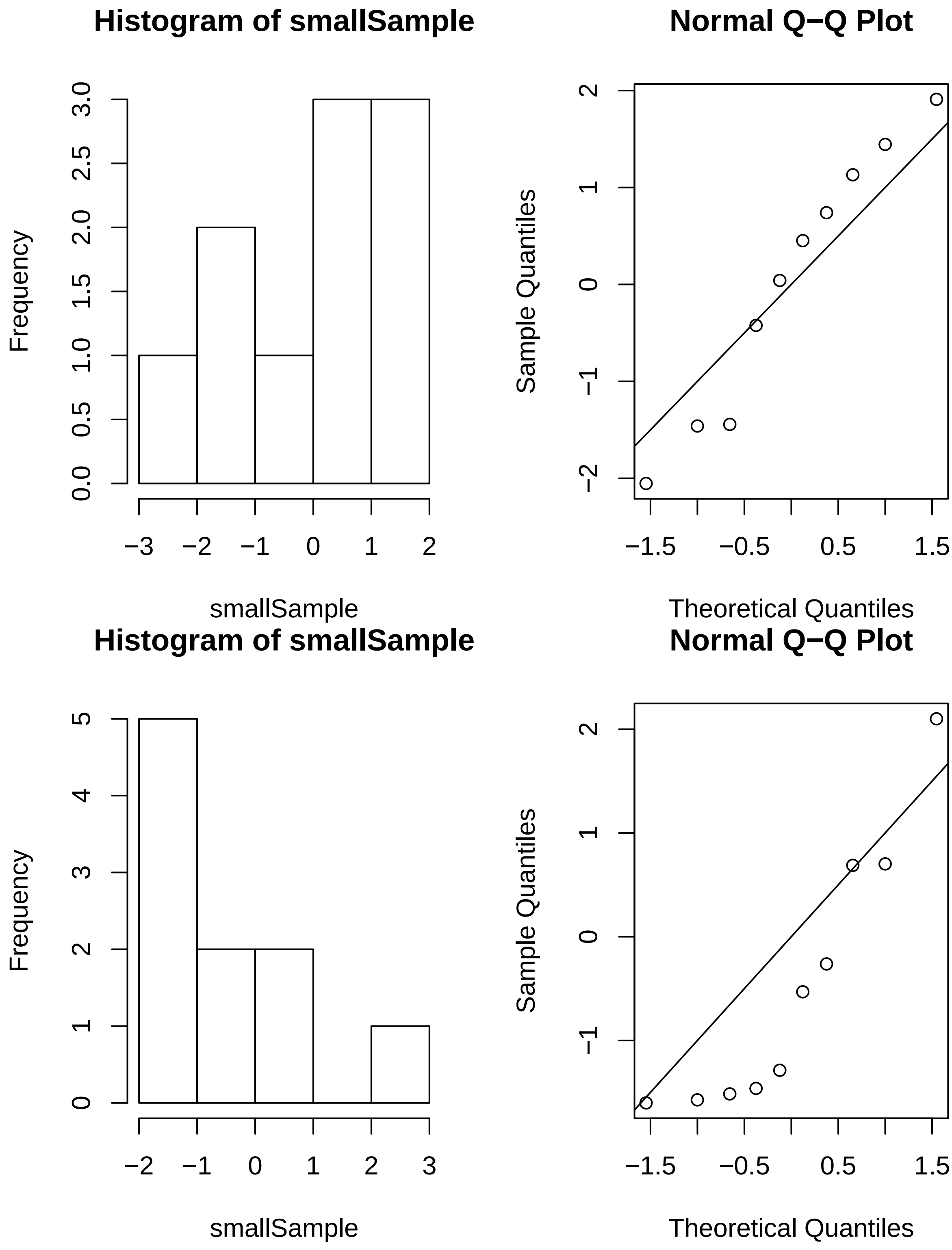

Normal Q−Q Plot
Sample Quantiles
−2.0
−1.0
0.0
1.0
−1.5
−0.5
0.5
1.5
Theoretical Quantiles

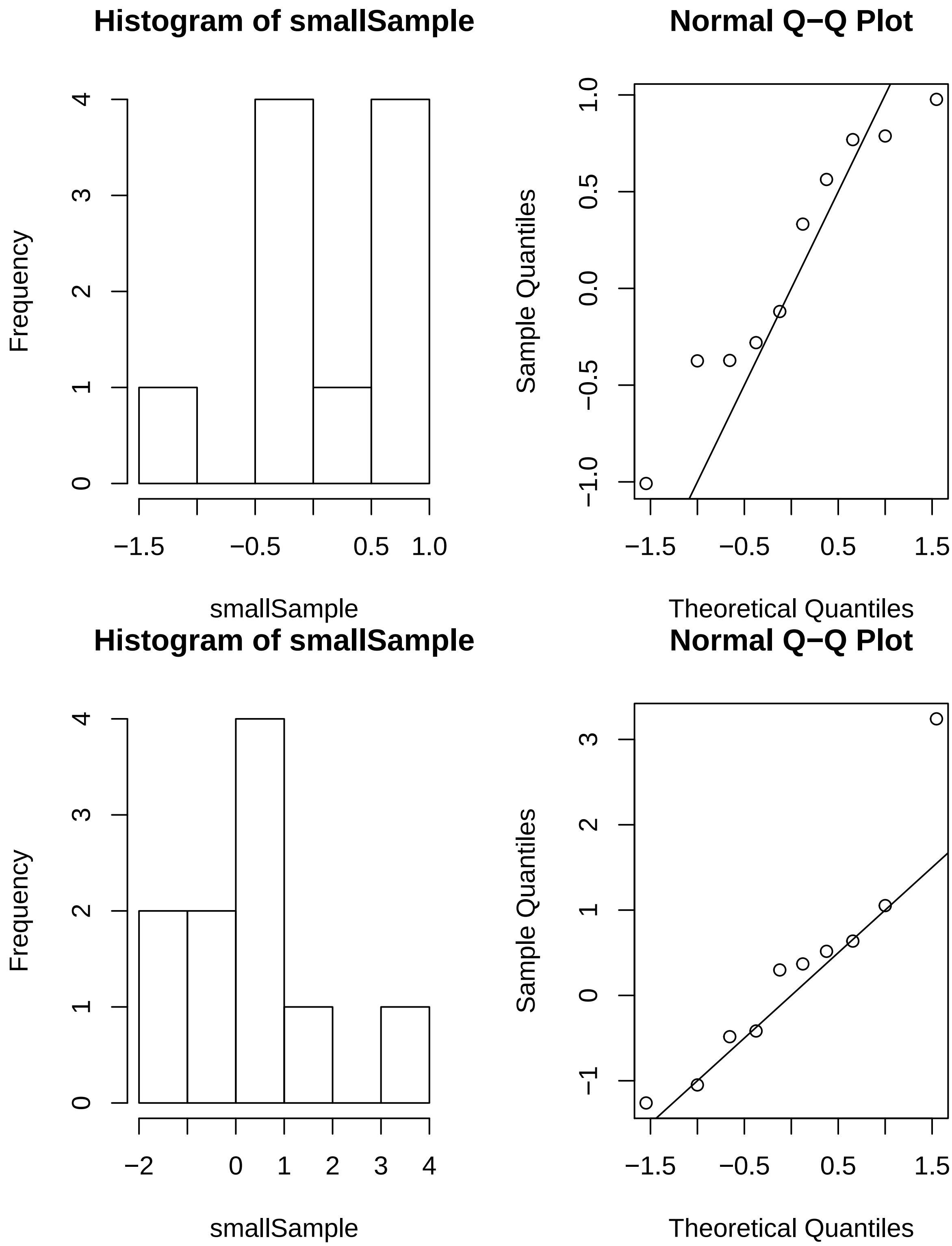

Histogram of smallSample
Frequency
0
1
2
3
4
−0.5
0.0
0.5
1.0
1.5
smallSample

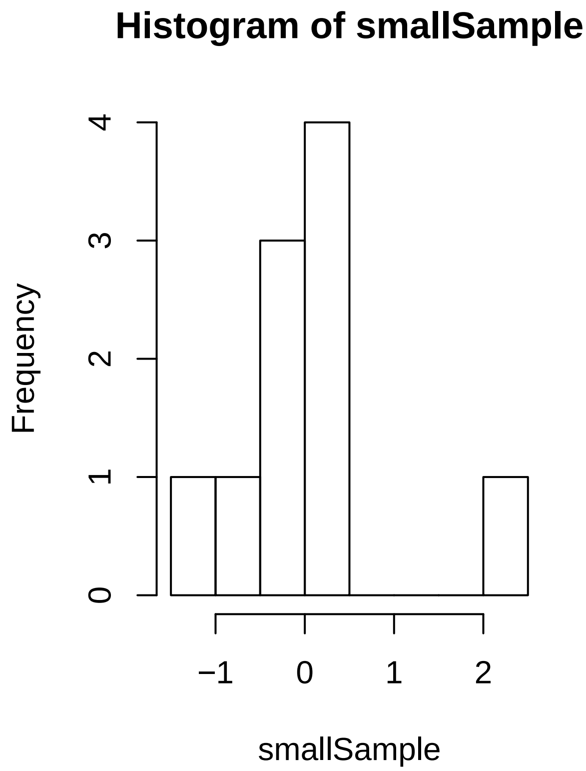

Normal Q−Q Plot
Sample Quantiles
0.0
0.5
1.0
−1.5
−0.5
0.5
1.5
Theoretical Quantiles

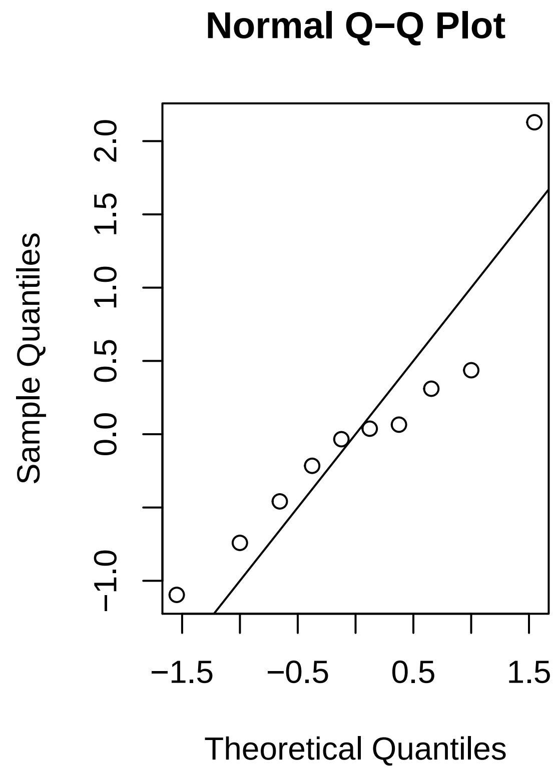

Histogram of smallSample
Frequency
smallSample
Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles
Histogram of smallSample
Frequency
smallSample
Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles

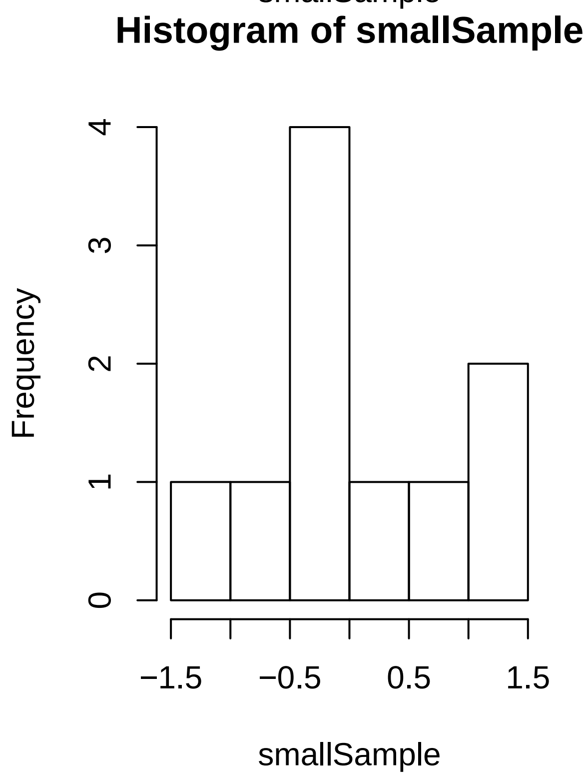

Histogram of smallSample
Frequency
smallSample
Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles
Histogram of smallSample
Frequency
smallSample
Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles

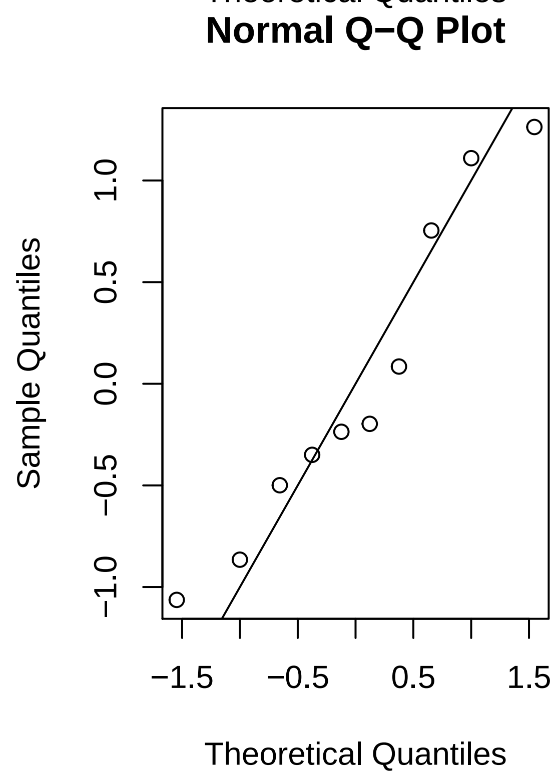

Histogram of smallSample
Frequency
0.0
0.5
1.0
1.5
2.0
2.5
3.0
−3
−2
−1
0
1
2
smallSample
Normal Q−Q Plot
Sample Quantiles
−2
−1
0
1
2
−1.5
−0.5
0.5
1.5
Theoretical Quantiles
Histogram of smallSample
Frequency
0
1
2
3
4
5
−2
−1
0
1
2
3
smallSample
Normal Q−Q Plot
Sample Quantiles
−1
0
1
2
−1.5
−0.5
0.5
1.5
Theoretical Quantiles

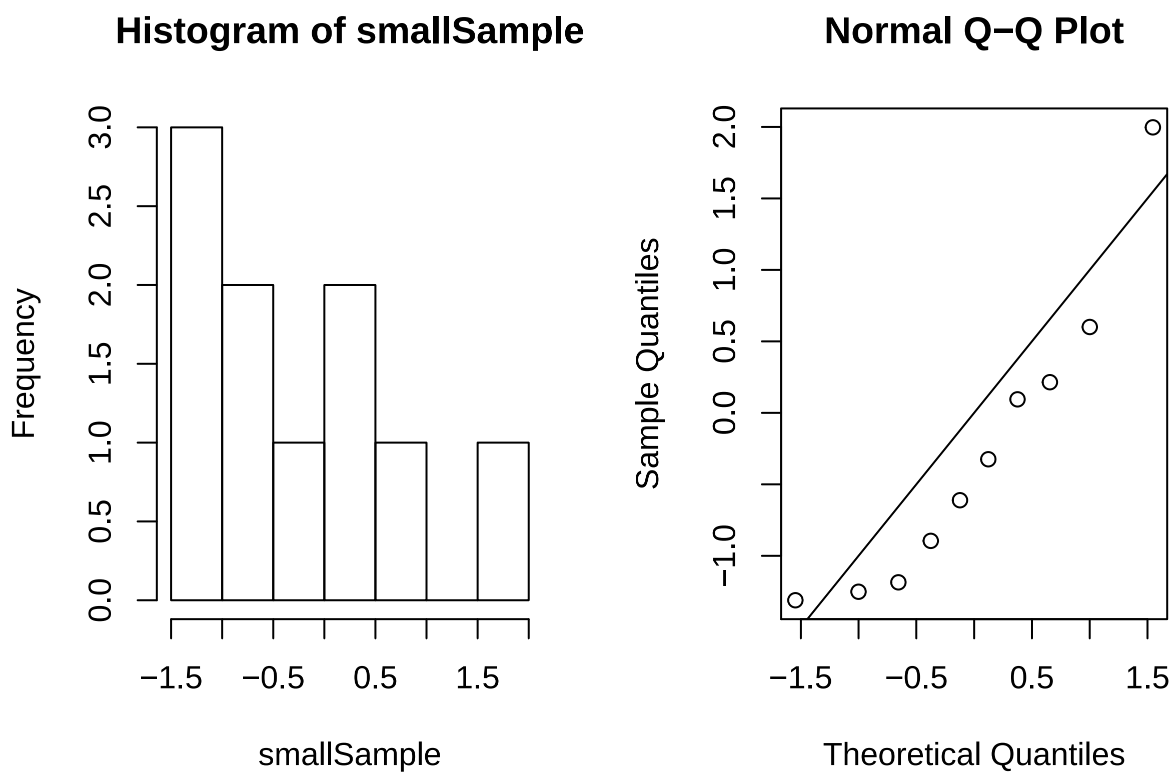

Histogram of smallSample
Frequency
0
1
2
3
4
−1.5
−0.5
0.5
1.0
smallSample
Normal Q−Q Plot
Sample Quantiles
−1.0
−0.5
0.0
0.5
1.0
−1.5
−0.5
0.5
1.5
Theoretical Quantiles
Histogram of smallSample
Frequency
0
1
2
3
4
−2
0
1
2
3
4
smallSample
Normal Q−Q Plot
Sample Quantiles
−1
0
1
2
3
−1.5
−0.5
0.5
1.5
Theoretical Quantiles

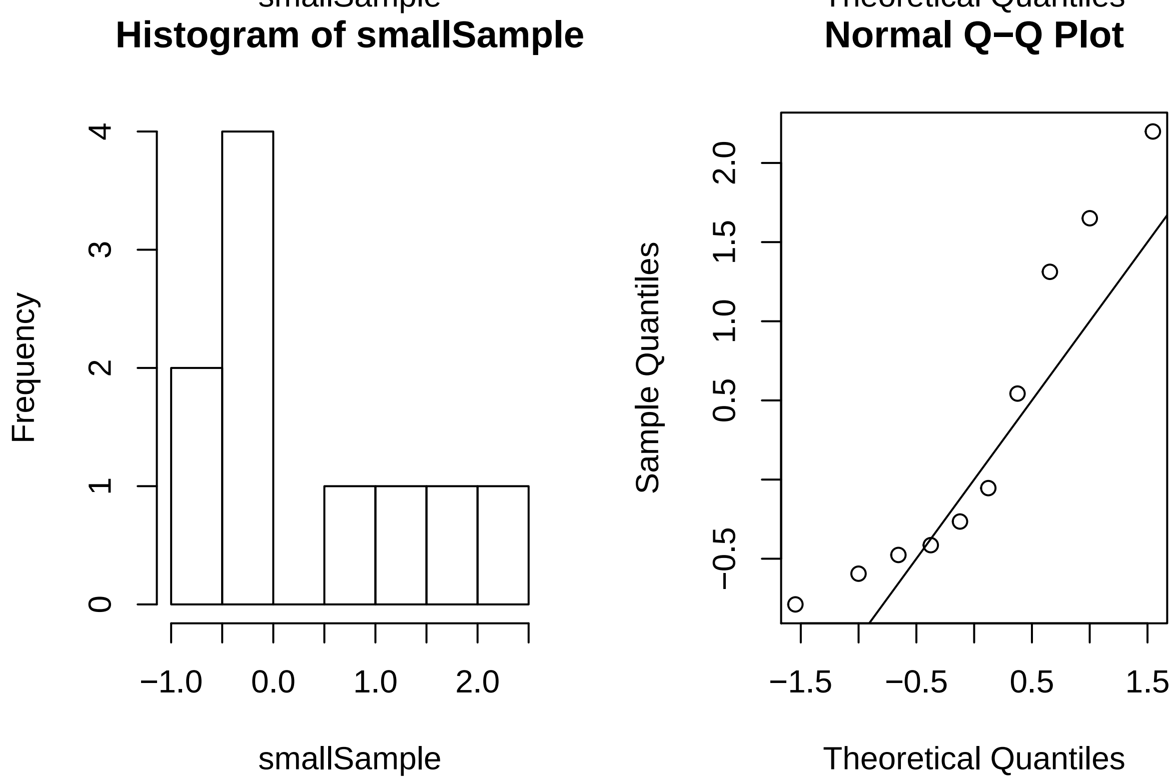

Histogram of smallSample
Frequency
0
1
2
3
4
−1
0
1
2
smallSample


Normal Q−Q Plot
Sample Quantiles
−1.0
0.0
0.5
1.0
1.5
2.0
−1.5
−0.5
0.5
1.5
Theoretical Quantiles


Histogram of smallSample
Frequency
0
1
2
3
4
−1.5
−0.5
0.5
1.5
smallSample


Normal Q−Q Plot
Sample Quantiles
−1.0
−0.5
0.0
0.5
1.0
−1.5
−0.5
0.5
1.5
Theoretical Quantiles



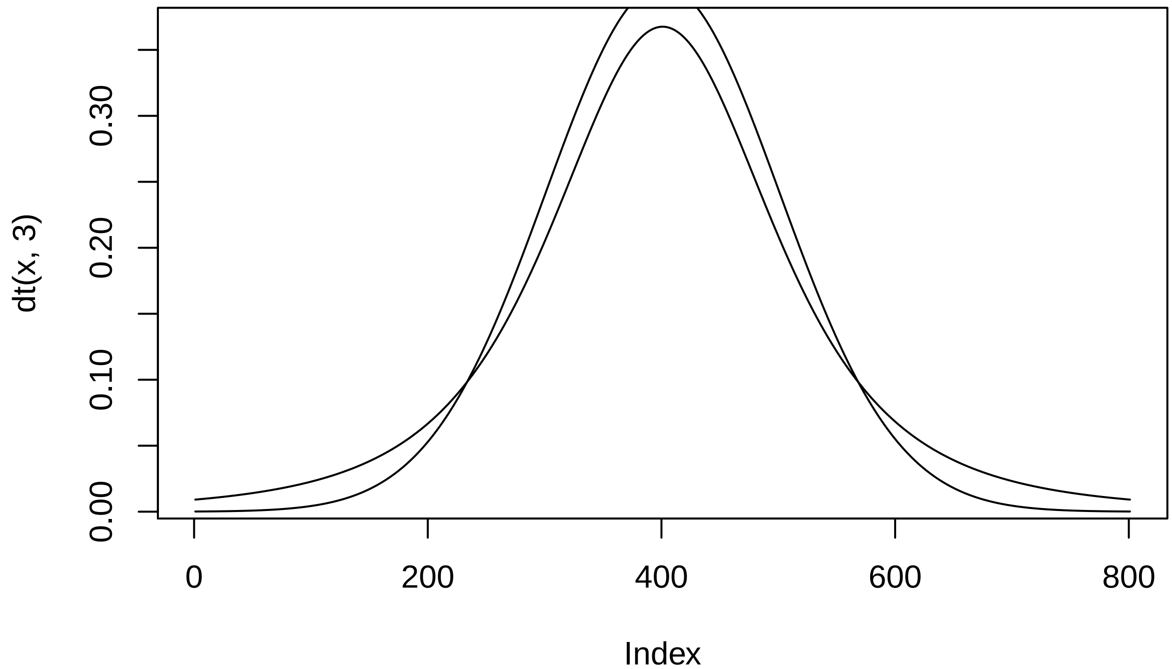


After generating 20 seperate instances, we see we were a bit lucky with the qqplot and it doesnt really come up to snuff.

Problem 2

```
x = seq(-4,4,.01)
plot(dt(x,3),type='l')
```

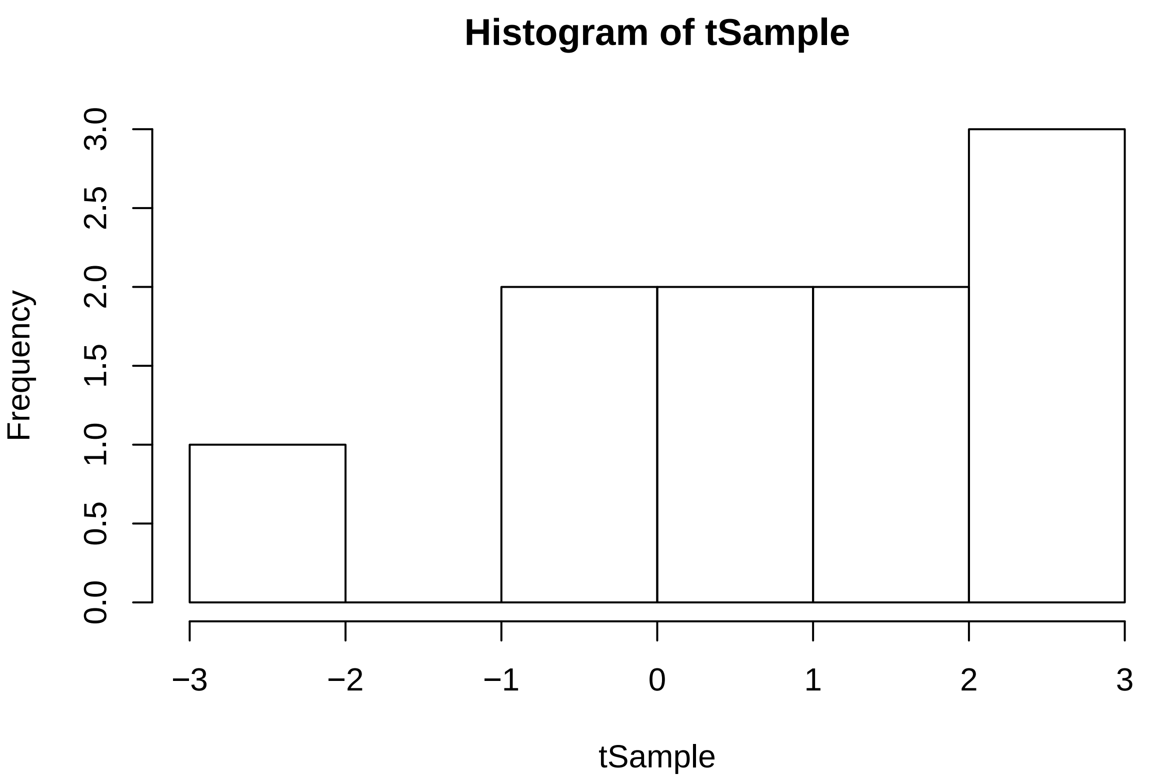
```
lines(dnorm(x,0,1),type='l')
```



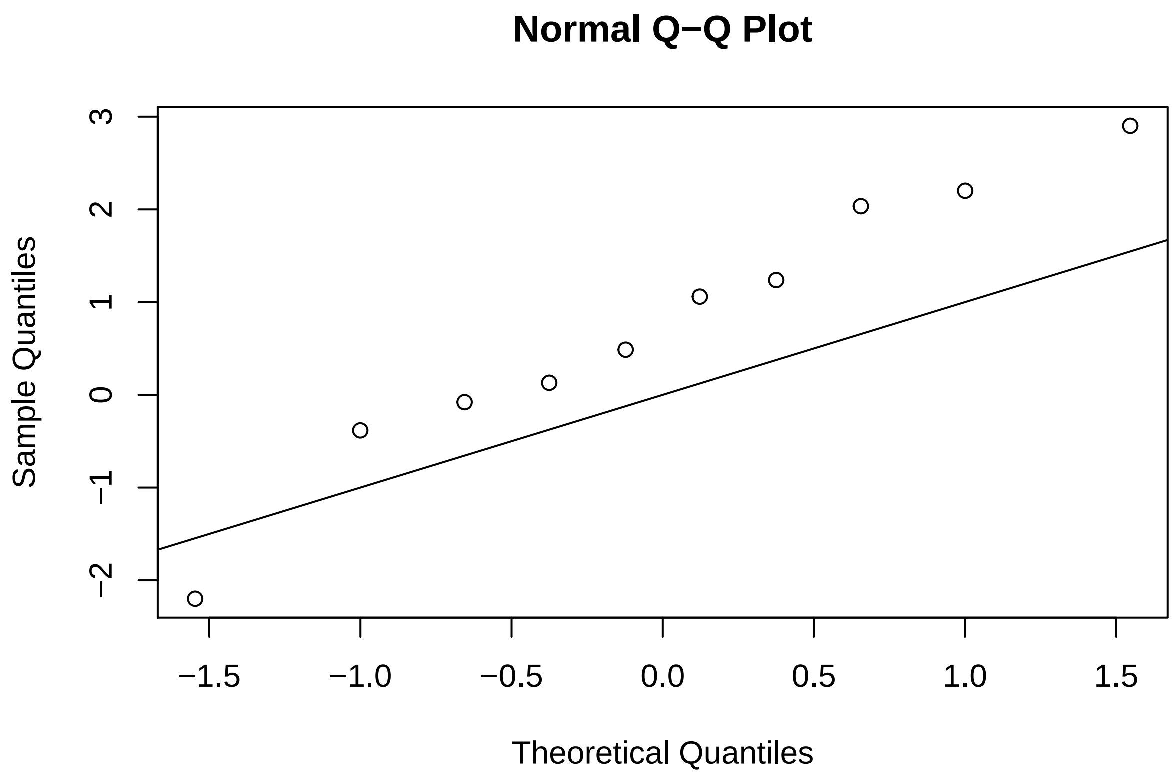
as we can see, there is a heavier tail and smaller and flatter peak for the t distribution

Problem 3

```
tSample = rt(10,3)
hist(tSample)
```



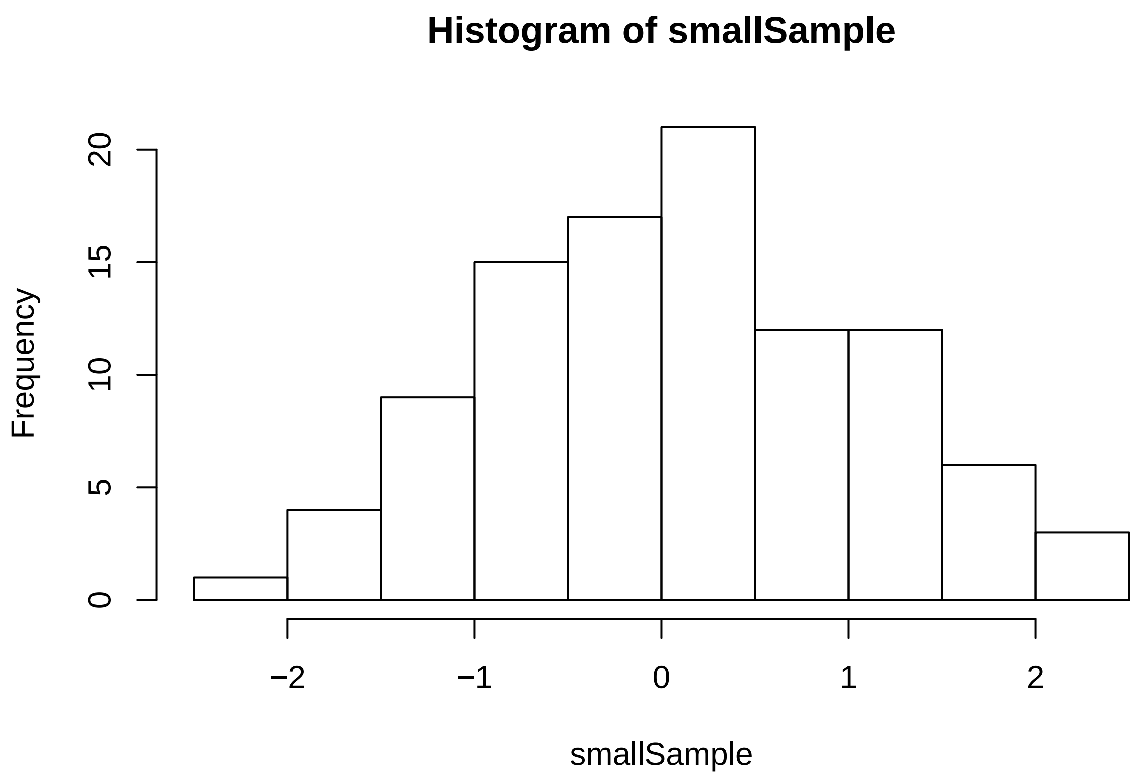
```r
qqnorm(tSample)
abline(0,1)
```



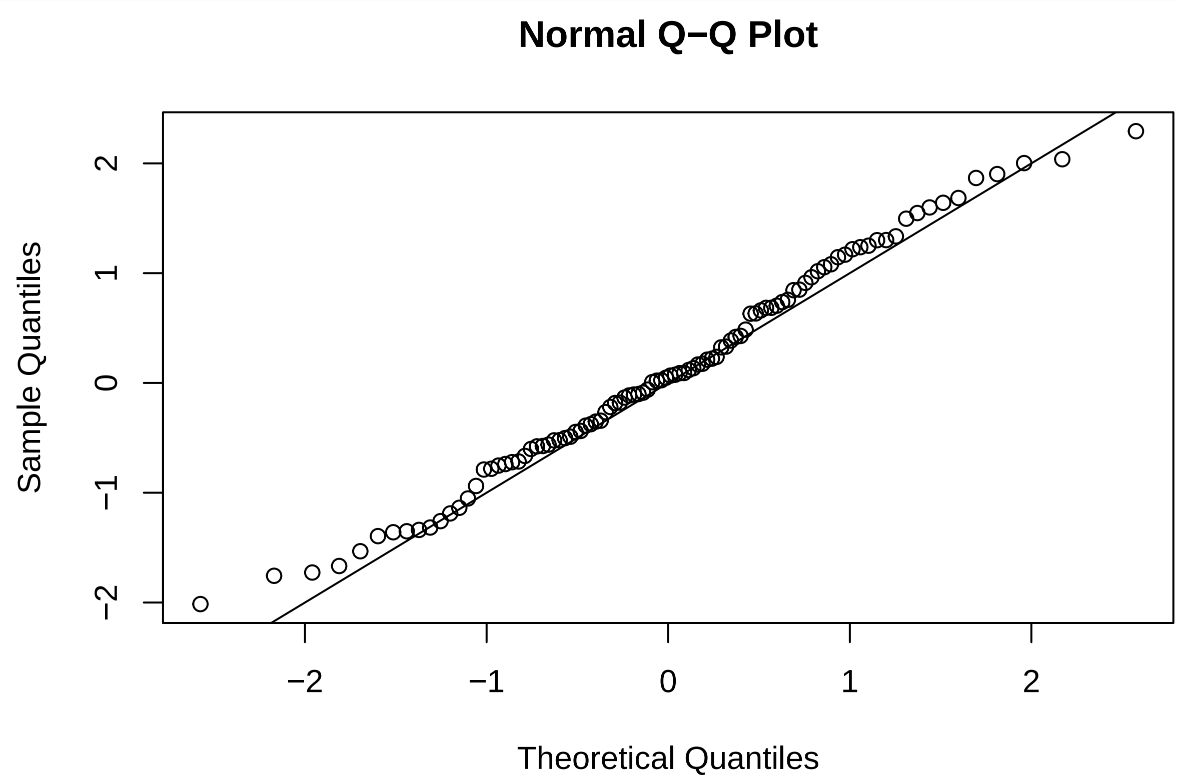
Surpirisngly, it looks pretty normal via the qqplot, but not in the histogram, just like before

Problem 4

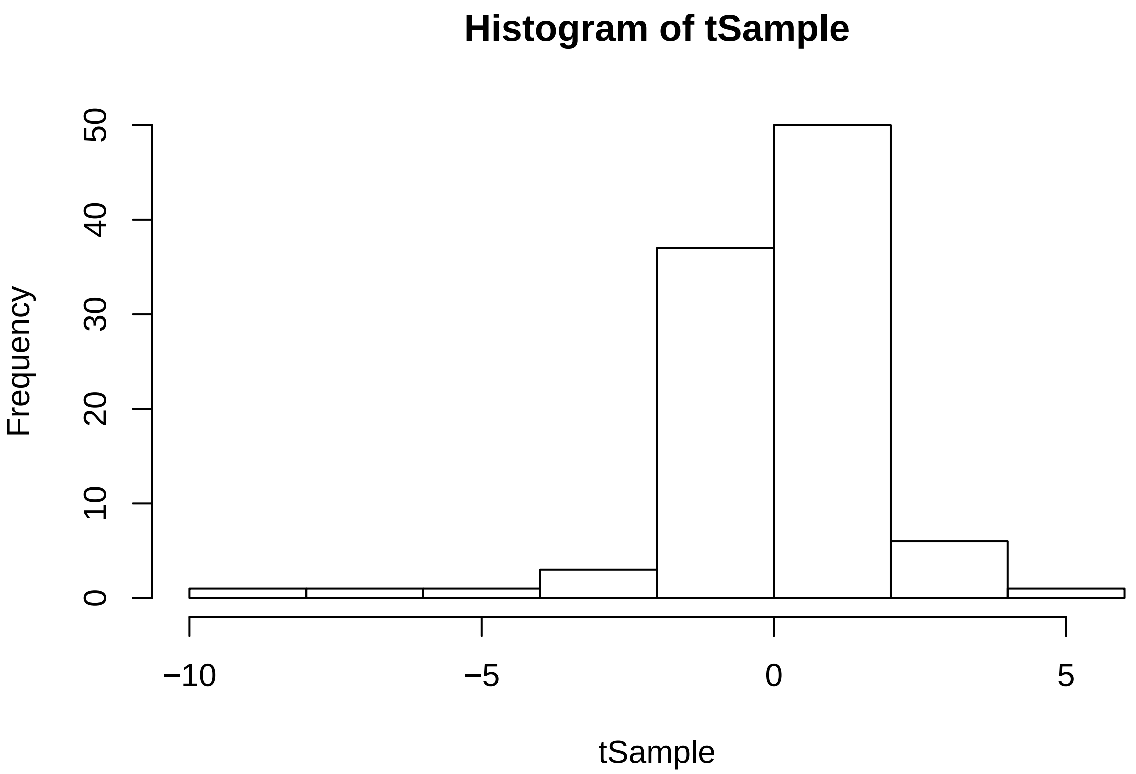
```r
n=100
smallSample = (rnorm(n,0,1))
tSample = rt(n,3)
hist(smallSample)
```



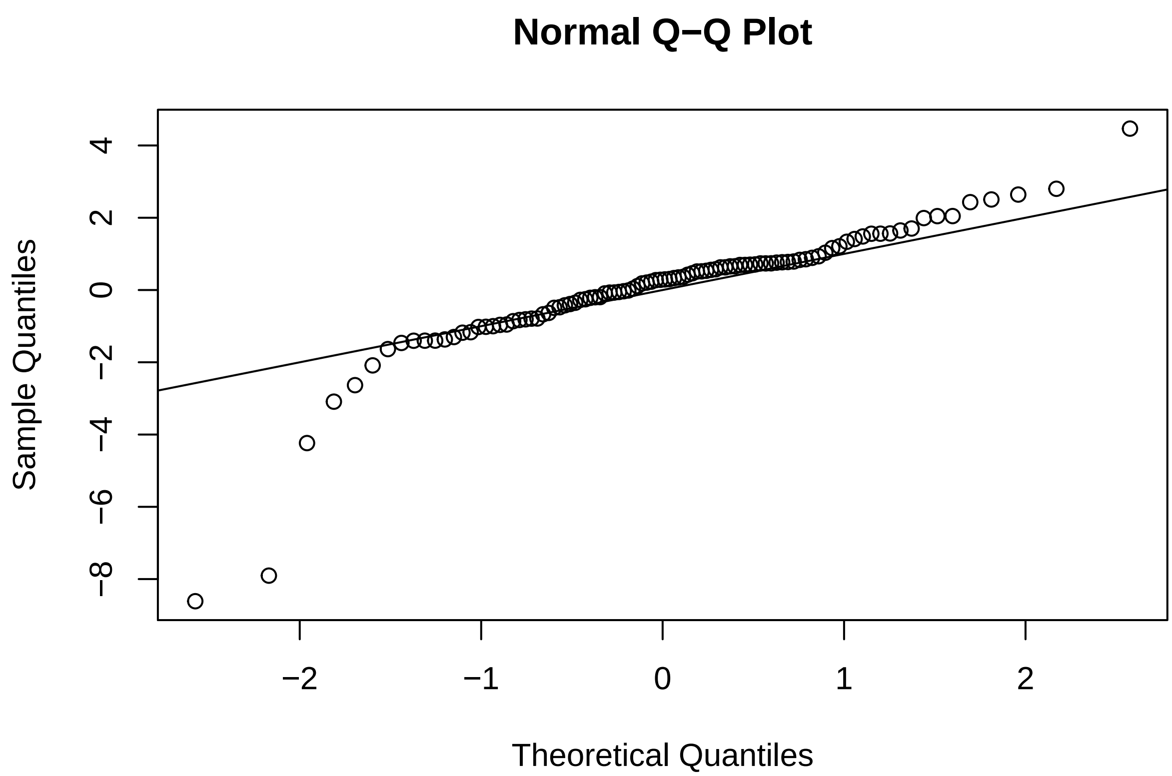

```r
qqnorm(smallSample)
abline(0,1)
```



```r
hist(tSample)
```



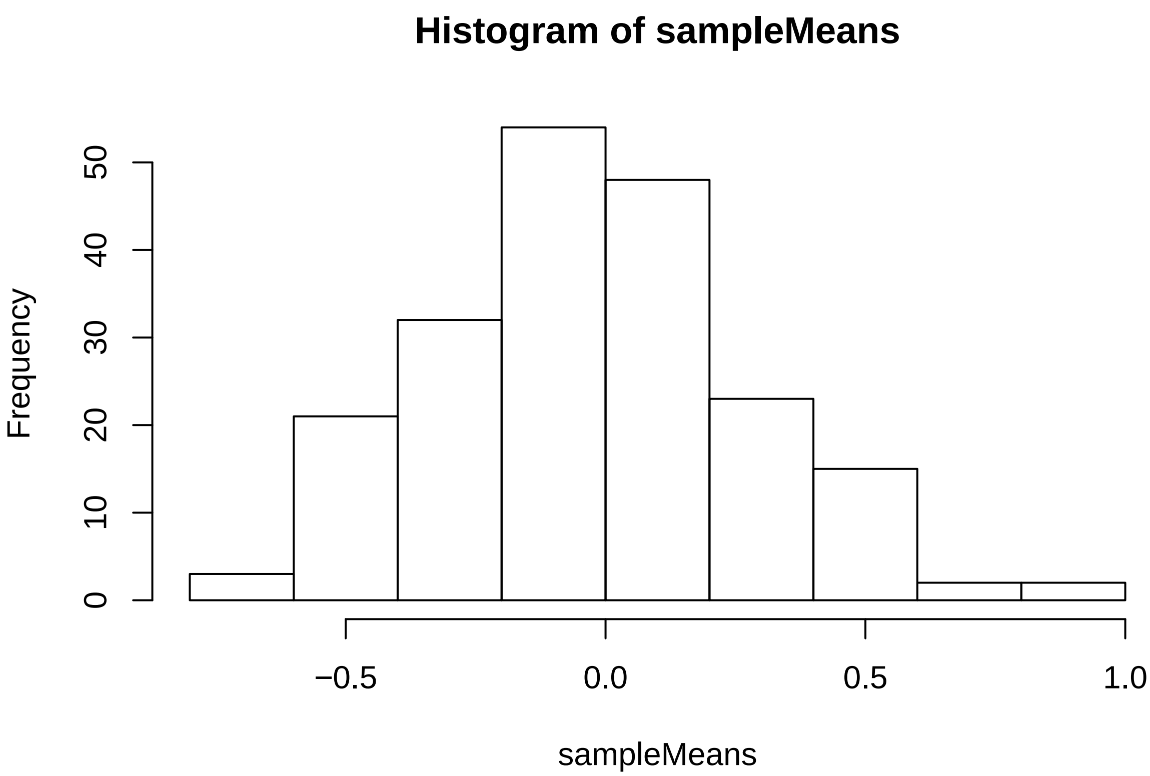

```r
qqnorm(tSample)
abline(0,1)
```



We seem to have found a normal looking noramlly generated distrib, but not very true to the actual curve is the t distributed sample.

Problem 5

```
sampleMeans = numeric(200)
sampleSize = 10
for(i in 1:200)
{
  sampleMeans[i]=mean(rnorm(sampleSize))
}
hist(sampleMeans)
```




```
print(mean(sampleMeans))
```

```
## [1] -0.02552663
```

```
print((var(sampleMeans)))
```

```
## [1] 0.09160695
```

```
print(sqrt(var(sampleMeans)))
```

```
## [1] 0.3026664
```

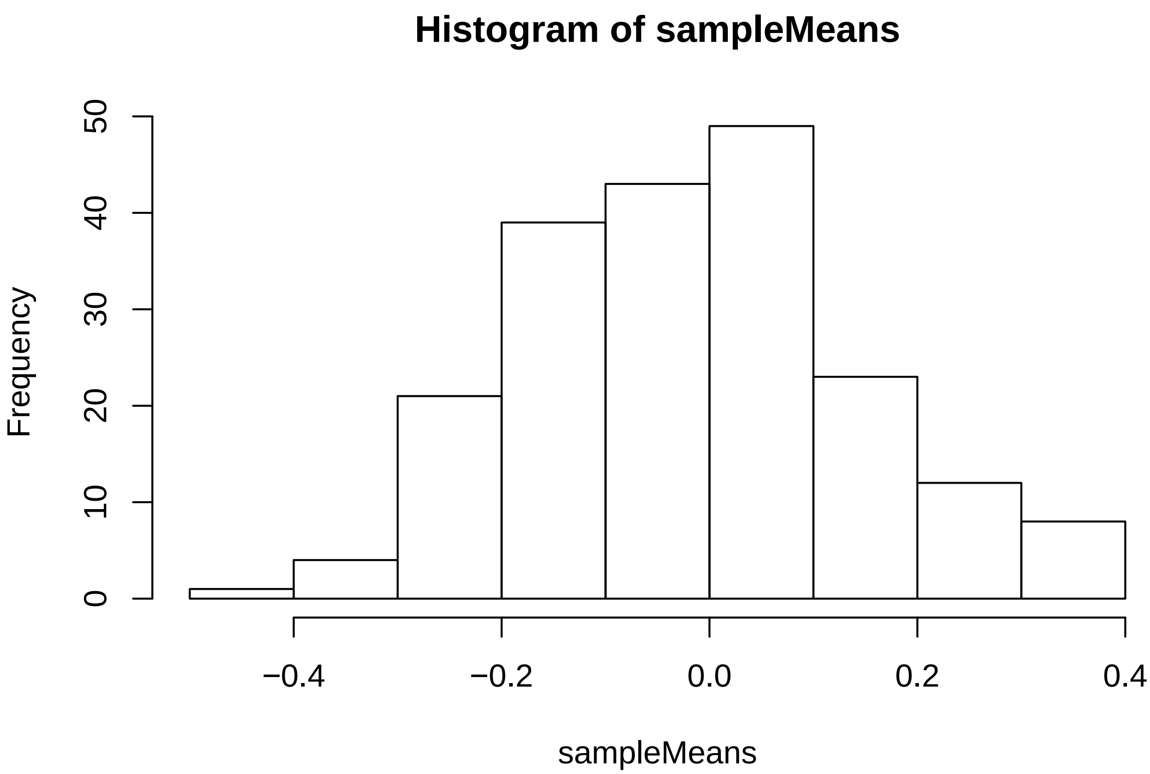
```
print(1/sqrt(sampleSize))
```

```
## [1] 0.3162278
```

Problem 6

```
sampleMeans = numeric(200)
sampleSize = 40
for(i in 1:200)
{
```

```
  sampleMeans[i]=mean(rnorm(sampleSize))
}
hist(sampleMeans)
```




```
print(mean(sampleMeans))
```

```
## [1] -0.01851286
```

```
print((var(sampleMeans)))
```

```
## [1] 0.02508546
```

```
print(sqrt(var(sampleMeans)))
```

```
## [1] 0.1583839
```

```
print(1/sqrt(sampleSize))
```

```
## [1] 0.1581139
```

Problem 7

```
print(1/.01)
```

```
## [1] 100
```

by dividing the true variance by the standard error we desire, we see that 100 sample size is needed theoretically to achieve it.

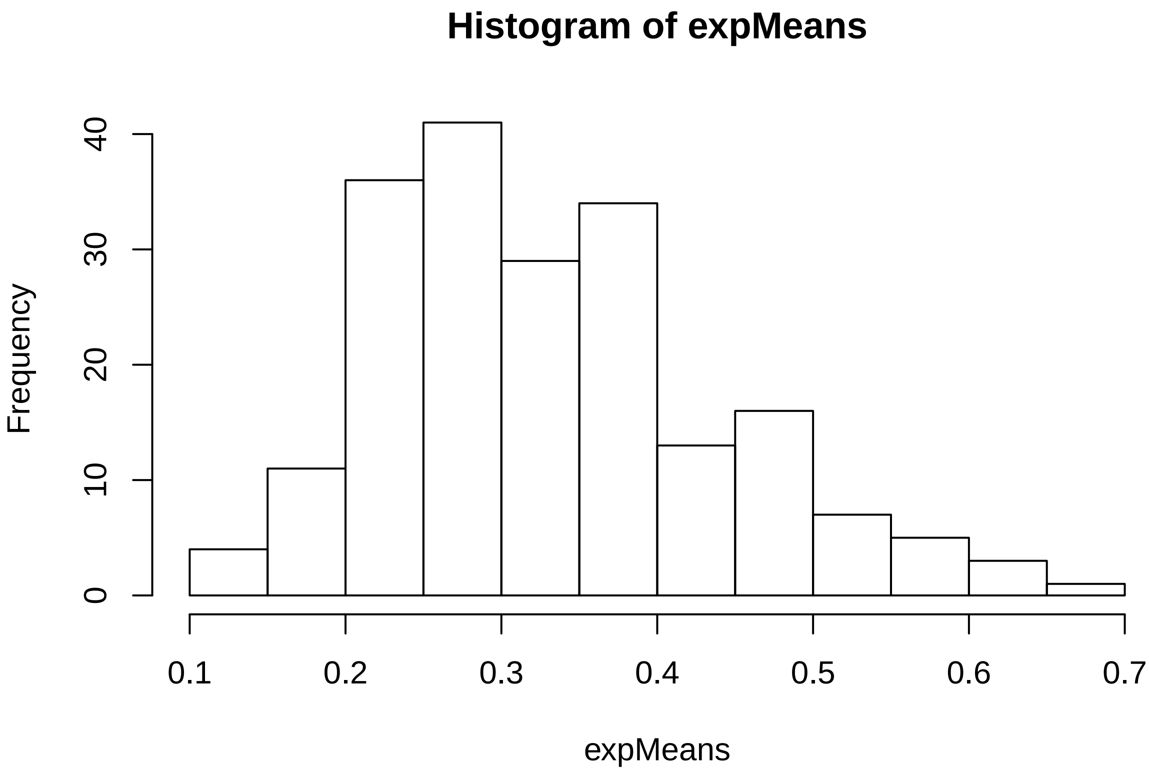
Problem 8

```
bernoulSample = rbinom(10,5,0.1)
print((.1*.9)/(0.01**2))
```

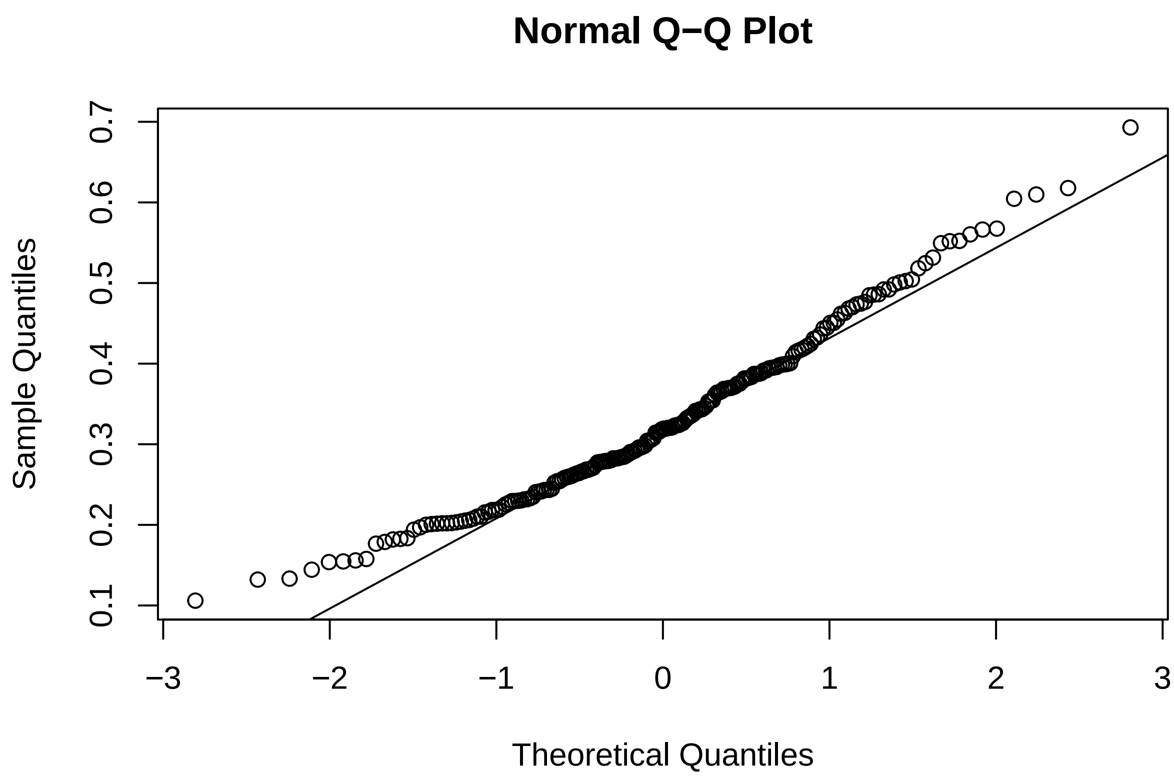
```
## [1] 900
```

In the above, we knew the standard error, which is the SD of the sample proportion, is 0.01, and then knew the true variance follows p*q. We simple invert, square, and solve for n Problem 9

```
expMeans = numeric(200)
for(i in 1:200)
{
  expMeans[i] = mean(rexp(10,3))
}
hist(expMeans)
```




```
qqnorm(expMeans)
qqline(expMeans)
```



problem 10

```
print(mean(expMeans))
```

```
## [1] 0.3304988
```

```
print(var(expMeans))
```

```
## [1] 0.01217654
```

```
print(sqrt(var(expMeans)))
```

```
## [1] 0.1103474
```

```
print("versus theoreticals")
```

```
## [1] "versus theoreticals"
```

```
print((1/3))
```

```
## [1] 0.3333333
```

```
print((1/9)/10)
```

```
## [1] 0.01111111
```

```
print(sqrt((1/9)/10))
```

```
## [1] 0.1054093
```

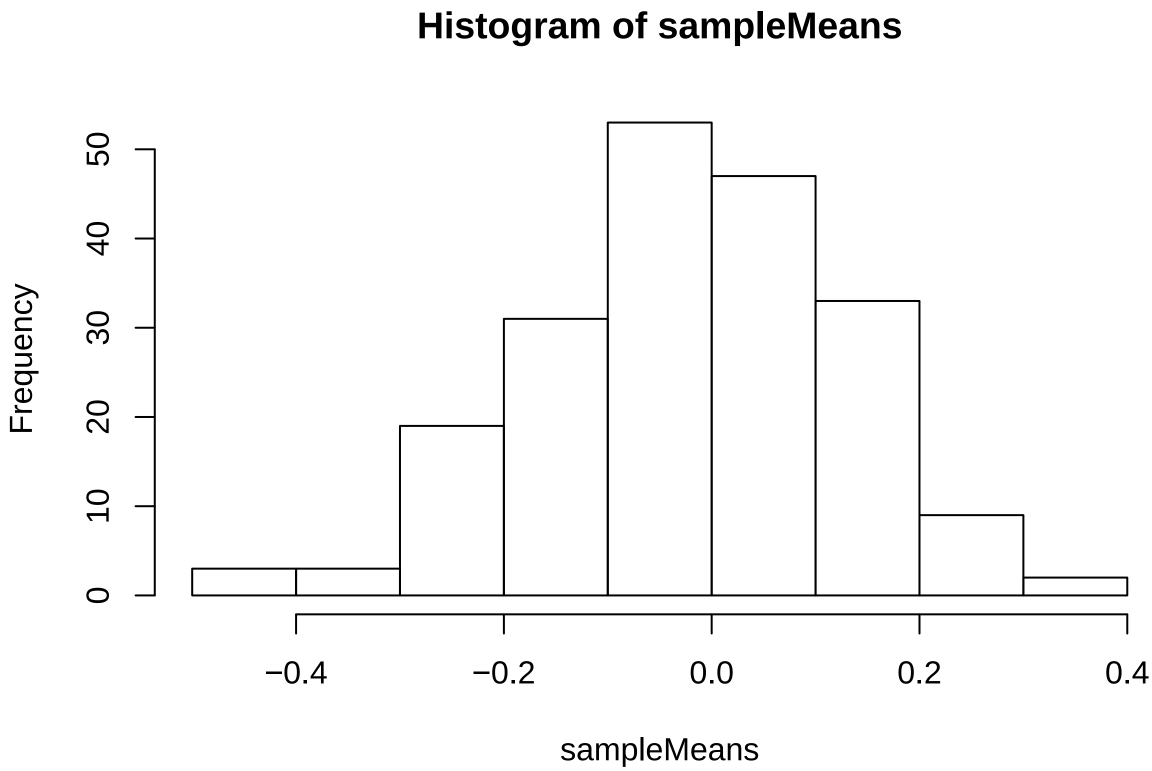
They are pretty close!

problem 11

```
print("problem number 5 repeated")
```

```
## [1] "problem number 5 repeated"
```

```
sampleMeans = numeric(200)
sampleSize = 40
for(i in 1:200)
{
  sampleMeans[i]=mean(rnorm(sampleSize))
}
hist(sampleMeans)
```




```
print(mean(sampleMeans))
```

```
## [1] -0.02022599
```

```
print((var(sampleMeans)))
```

```
## [1] 0.02224429
```

```
print("versus theoreticals")
```

```
## [1] "versus theoreticals"
```

```
print(sqrt(var(sampleMeans)))
```

```
## [1] 0.1491452
```

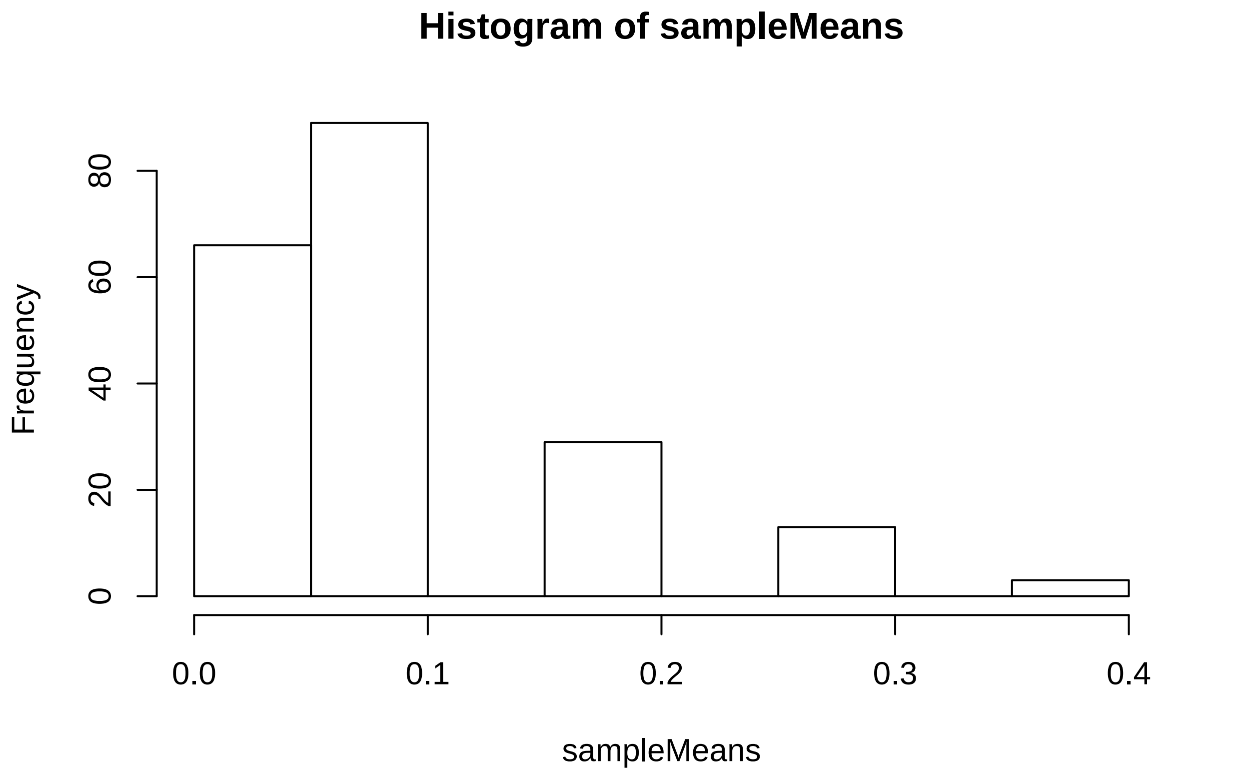
```
print(1/sqrt(sampleSize))
```

```
## [1] 0.1581139
```

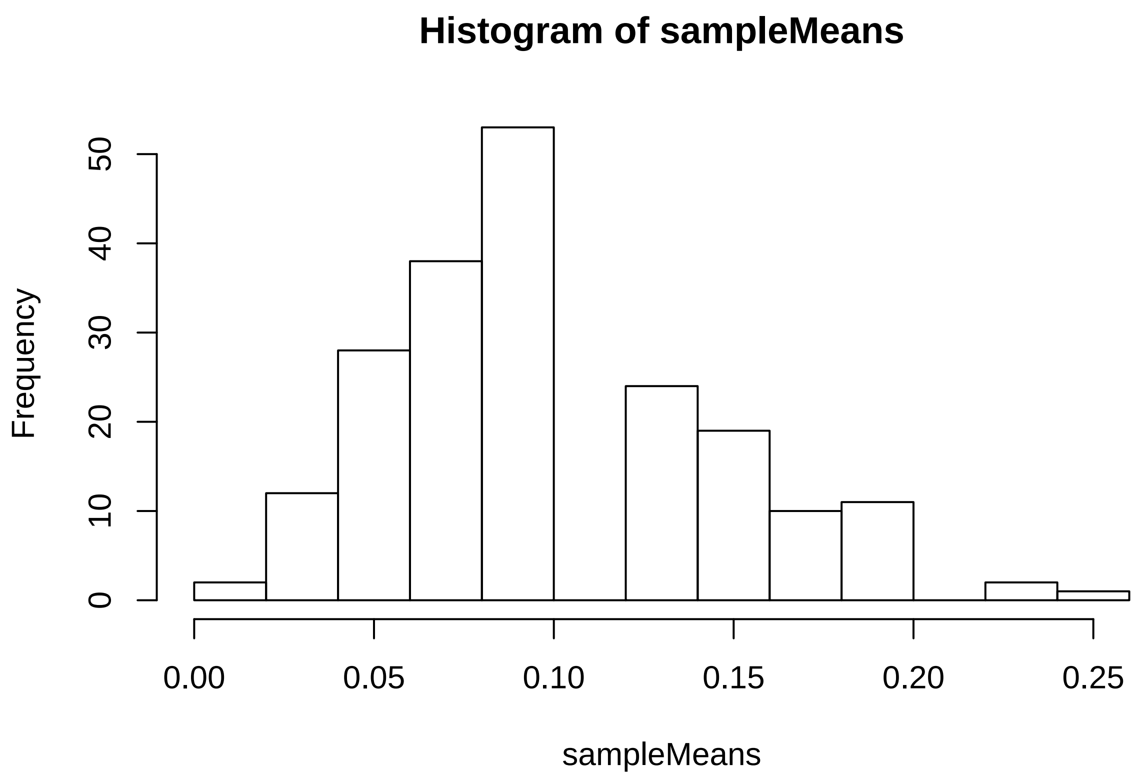
problem 12

```
sampleMeans = numeric(200)
sampleSize = 10
```

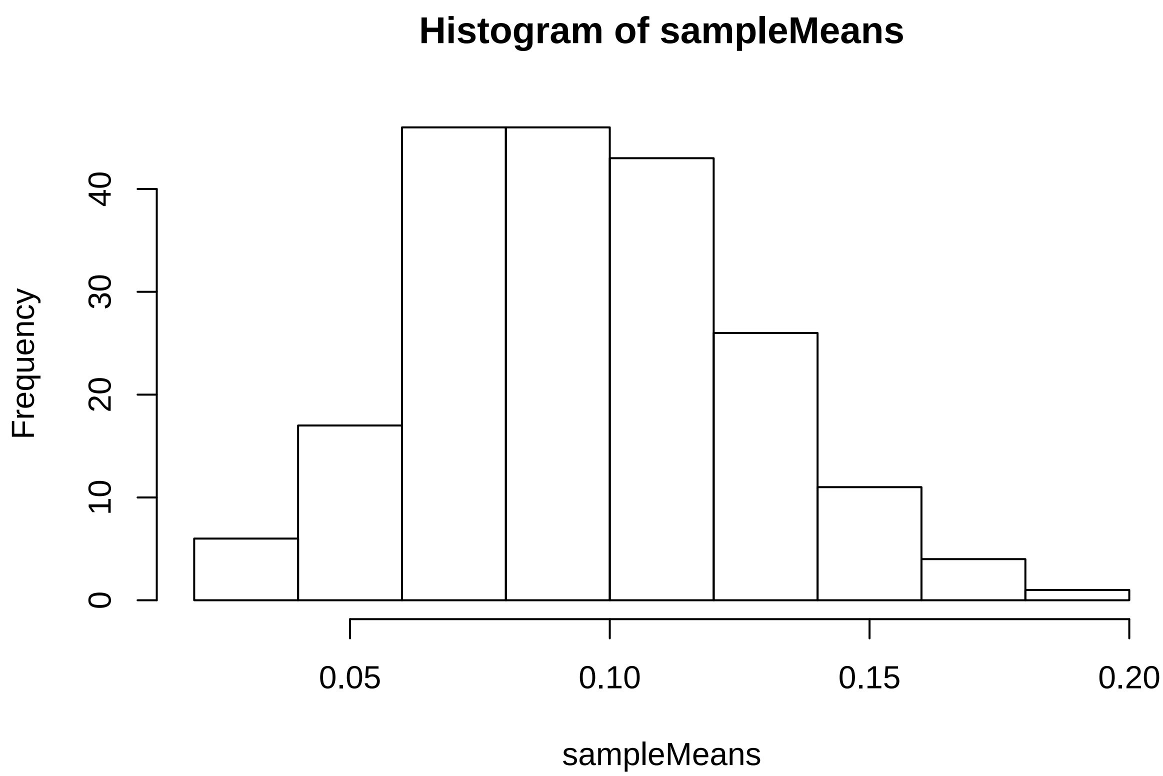
```r
for(i in 1:200)
{
  sampleMeans[i]=mean(rbinom(sampleSize,1,0.1))
}
hist(sampleMeans)
```




```r
sampleMeans = numeric(200)
sampleSize = 40
for(i in 1:200)
{
  sampleMeans[i]=mean(rbinom(n=sampleSize,size=1,prob=0.1))
}
hist(sampleMeans)
```

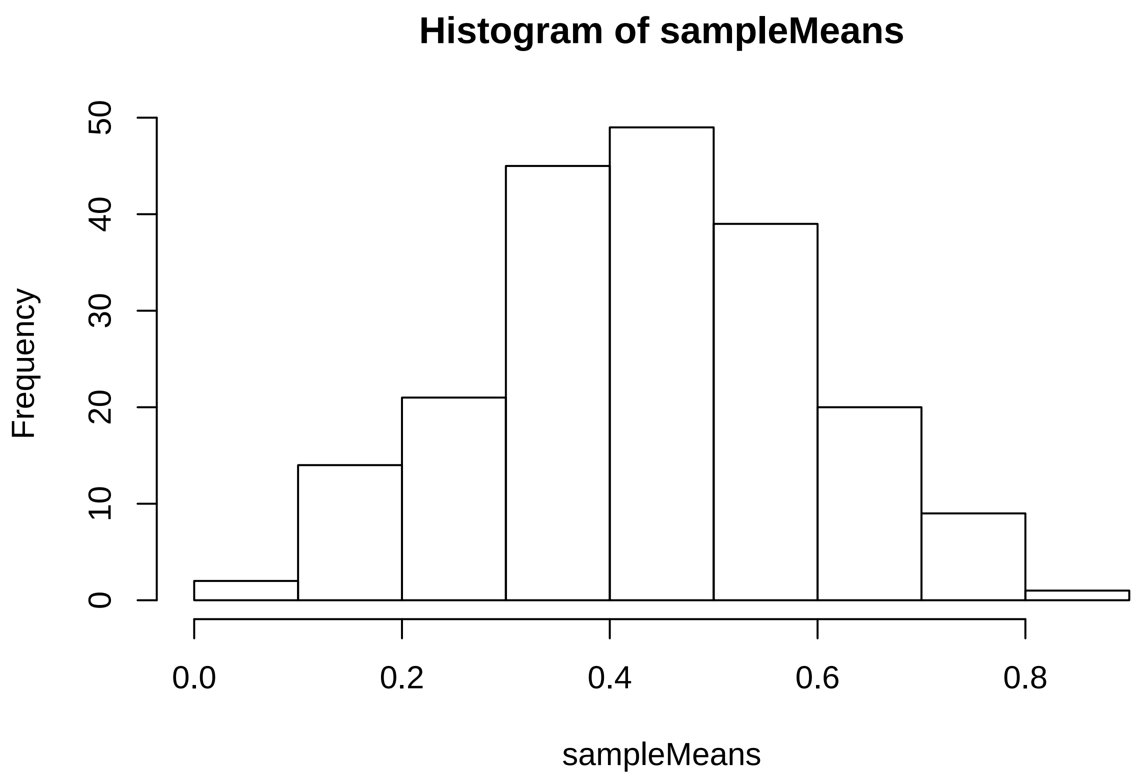



```
sampleMeans = numeric(200)
sampleSize = 100
for(i in 1:200)
{
  sampleMeans[i]=mean(rbinom(n=sampleSize,size=1,prob=0.1))
}
hist(sampleMeans)
```

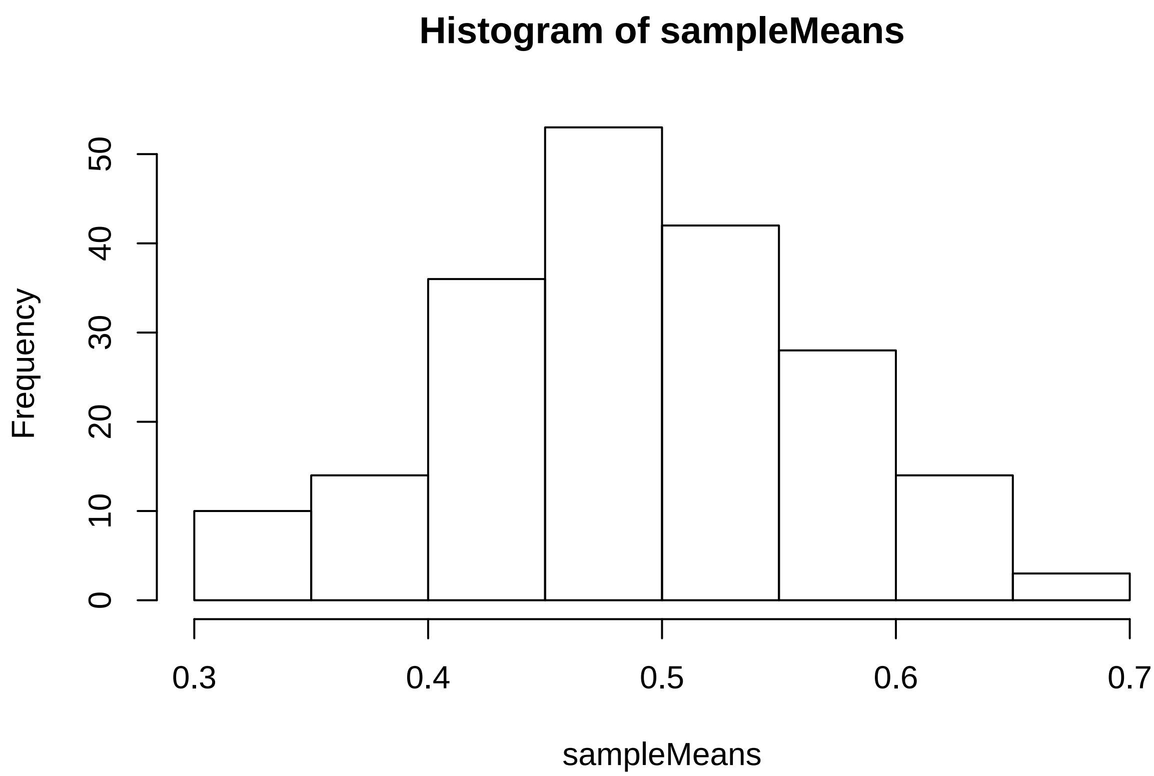


Problem 13

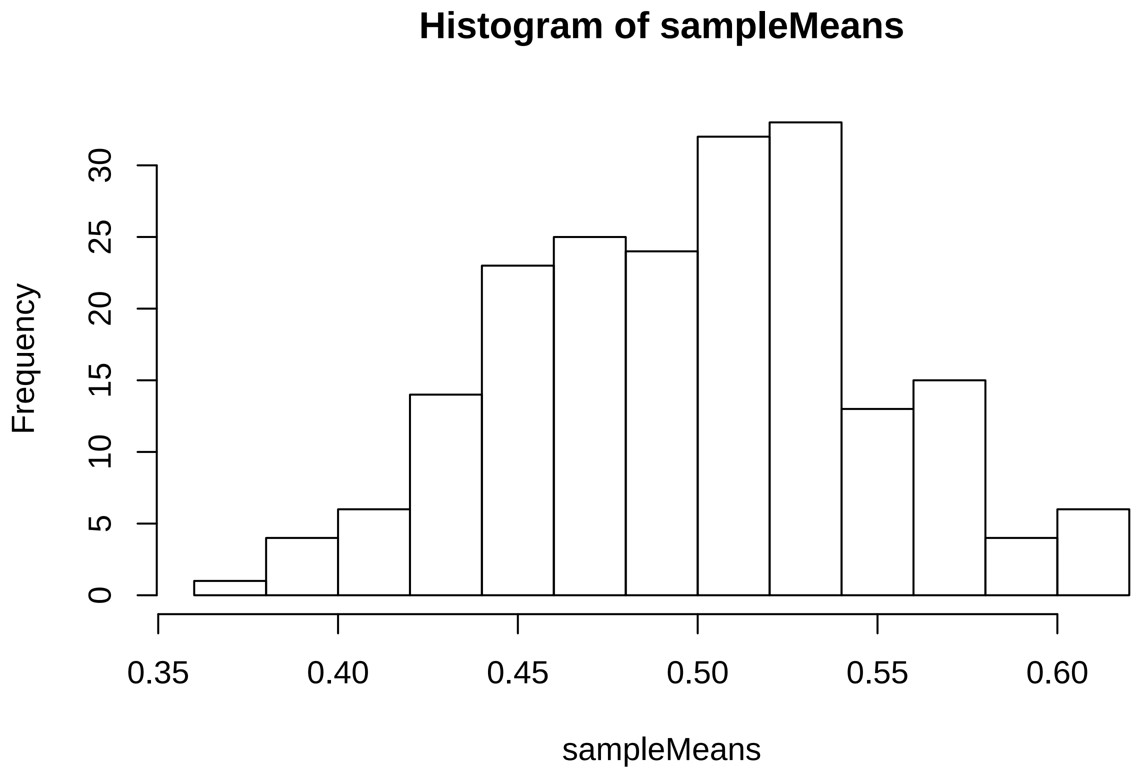
```
sampleMeans = numeric(200)
sampleSize = 10
for(i in 1:200)
{
  sampleMeans[i]=mean(rbinom(sampleSize,1,0.5))
}
hist(sampleMeans)
```




```r
sampleMeans = numeric(200)
sampleSize = 40
for(i in 1:200)
{
  sampleMeans[i]=mean(rbinom(n=sampleSize,size=1,prob=0.5))
}
hist(sampleMeans)
```




```
sampleMeans = numeric(200)
sampleSize = 100
for(i in 1:200)
{
  sampleMeans[i]=mean(rbinom(n=sampleSize,size=1,prob=0.5))
}
hist(sampleMeans)
```



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.