Homework1

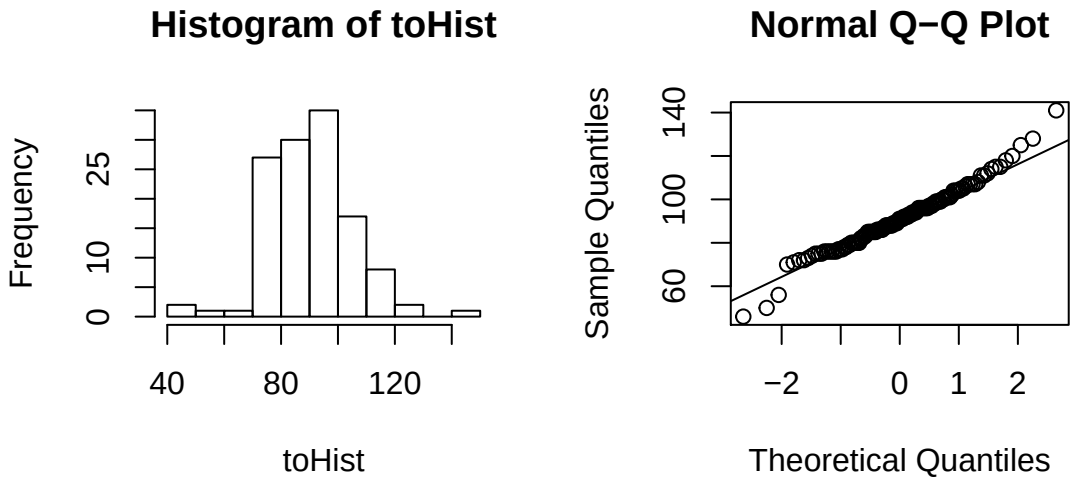
Alexander Van Roijen

January 10, 2019

Problem 1

Important note: " You might wonder why this is necessary. Isn't the coverage probability always (1-??) = 0.95? No, that is only true when the population is normally distributed (which is never true in practice) or the sample sizes are large enough that you can invoke the Central Limit Theorem. "

```
par(mfrow=c(1,2))
toHist = iqData$IQ
hist(toHist)
qqnorm(toHist)
qqline(toHist)
```



Looking at these graphs, we see it looks decently normal, with perhaps some concern for the heavy-ish tails

Problem 2

```
iqs = iqData$IQ
print(mean(iqs))
```

```
## [1] 91.08065
```

```
print(sqrt(var(iqs)))
```

```
## [1] 14.40393
```

It appears that the sample SD is not too far off from the true SD, but the average is 10% off from what is expected to be the true average IQ. There are a few explanations as to why this may be, but lets consider these two possibilities.
1) We have a biased sampling, with a bias towards students who preform worse on the iq test
2) We have too small a sample size, and thus do not accurately reflect the true population.

Problem 3

```
SET = 15/sqrt(n)
negD = -1.96 * SET
posD = 1.96 * SET
lowerB = mean(iqs)+negD
upperB = mean(iqs)+posD
print(lowerB)
```

```
## [1] 88.44045
```

```
print(upperB)
```

```
## [1] 93.72084
```

As we can see, 100 is not within this 95% confidence interval on the sample mean using a standard error calculated using the theoretical SD. This is significant, as even if we consider the possibiity of a unlucky sampling, the variance and our standard error show that we are 95% confident that the true mean is not within this interval. Since 100 is outside of this interval, we can be concerned that perhaps there is something wrong with the children we have sampled outside of random chance, or perhaps something is wrong with our sampling.

Problem 4

The equation for the 95% confidence interval is $\bar{x} \pm se * 1.96$
This means our width is $2 * se * 1.96$, which must satisfy ≤ 30
Pluggin in $se = \frac{SD}{\sqrt{n}}$ and solving for n we get $n \geq (\frac{2*SD*1.96}{30})^2 => n \geq 1.96^2$
This implies n must only be greater than or equal to 4 in order to achieve this desired width at a minimum.

Problem 5

```
SET = sqrt(var(iqs))/sqrt(n)
negD = -1.96 * SET
posD = 1.96 * SET
lowerB = mean(iqs)+negD
upperB = mean(iqs)+posD
print(lowerB)
```

```
## [1] 88.54536
```

```
print(upperB)
```

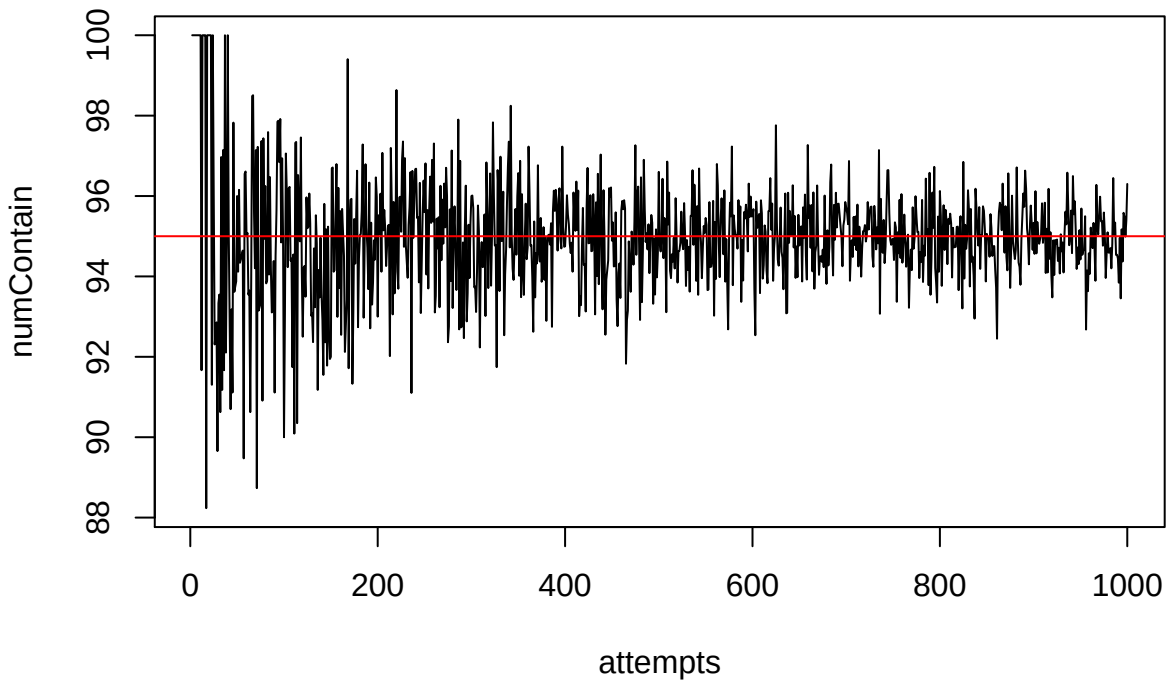
```
## [1] 93.61593
```

We can see that the confidence interval is slightly smaller in this section, as the variance and according standard deviation are smaller in this sample than what is expected to be the case.

Problem 6

```
numSamples=400
attempts =c(2:1000)
numAttempts = length(attempts)
numContain=numeric(numAttempts)
```

```
for(i in 1:numAttempts)
{
  samples = replicate(attempts[i],rnorm(n,100,15))
  allSEs = apply(samples,FUN=confInterval,MARGIN=2,doTrueVar = T)
  res = getResults(allSEs,100)
  percentBelow = mean(res)*100
  numContain[i] = (percentBelow)
}
plot(x=attempts,y=numContain,type="l")
abline(h=95,col='red')
```




```
samples = replicate(numSamples,rnorm(n,100,15))
allSEs = apply(samples,FUN=confInterval,MARGIN=2,doTrueVar = TRUE)
#print(allSEs)
res = getResults(allSEs,100)
allSEs=lapply(allSEs, "[[", 1)
percentOfTrue = sapply(allSEs,FUN=withinInterval,100)
print(mean(res)*100)
```

```
## [1] 93.75
```

After many simulations, it appears that the variance of coverage probability is satisfactorily reduced around 400 iterations. Clearly, we could continue going with more and more simulations, however, I believe 400 is more than generous. This indicates we do not have too much to worry as the 95% coverage probability seems to be a converging point for our simulations. Further there is no indication that the coverage probability is not around .95.

Problem 7

```
numSamples = 400
samples = replicate(numSamples,rnorm(n,100,15))
allSEs = apply(samples,FUN=confInterval,MARGIN=2,doTrueVar = FALSE)
```

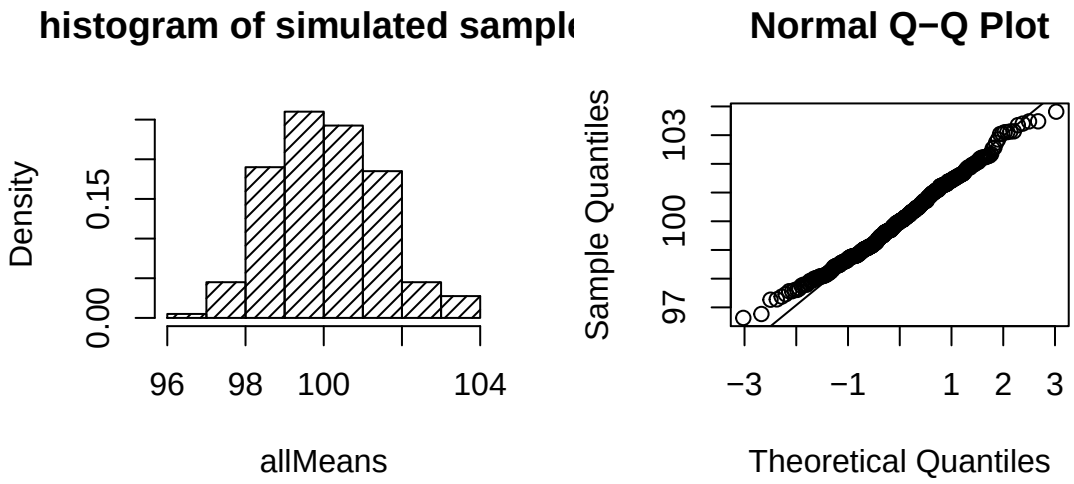
```r
#print(allSEs)
res = getResults(allSEs,100)
#allSEs=lapply(allSEs, "[[", 1)
#print(allSEs[,1]$lower)
#percentOfTrue = sapply(allSEs,FUN=withinInterval,100)
print(mean(res)*100)
```

```
## [1] 95.5
```

Similar to before, we see a similar coverage probability, which is to be expected as the variance between our sample and the true variance arent too off from each other.

Problem 8

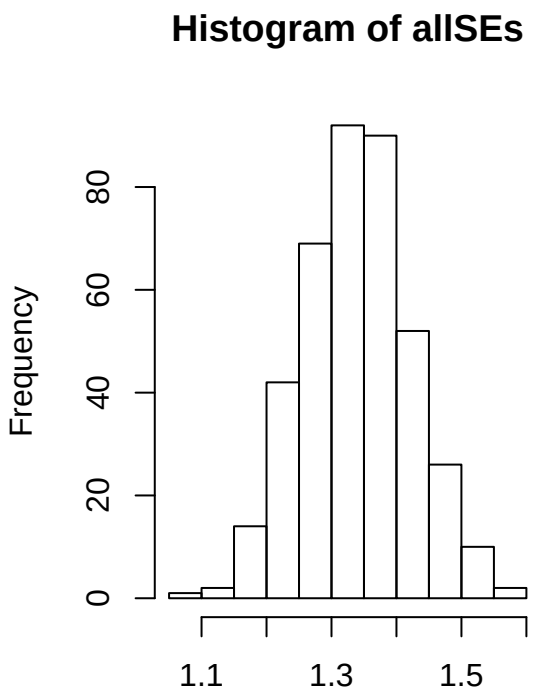
```r
par(mfrow=c(1,2))
allMeans = apply(samples,FUN=mean,MARGIN=2)
hist(allMeans,density=20,prob=T,main="histogram of simulated samples")
qqnorm(allMeans)
qqline(allMeans)
```

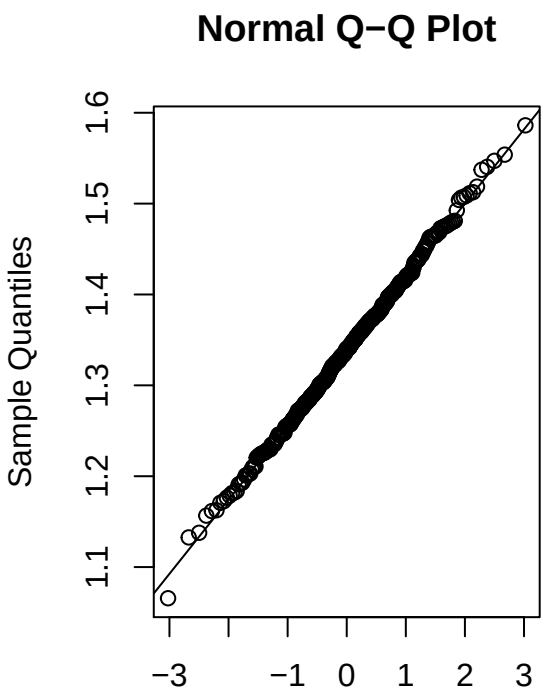


It appears that the means are quite normally distributed, wth a mean around 100, as to be expected. I expect this as we sampled from a normal distribution with mean 100 and SD 15, and thus expected to get these types of means as a result.

Problem 9

```r
par(mfrow=c(1,2))
allSEs = calcSEs(samples,n)
hist(allSEs)
qqnorm(allSEs)
qqline(allSEs)
```






```
allMeans = apply(samples,FUN=mean,MARGIN=2)
print(sd(allMeans))
```

```
## [1] 1.343453
```

```
print(mean(allSEs))
```

```
## [1] 1.338555
```

From each sample, we have a calculated standard error, looking at the mean of those standard errors, we expect that given our sample size from the population, each sampled mean should vary from each other by about 1.33 points. or at least, that is what most of the samples indicated. the standard deviation amongst the means of the samples is also indicative of this fact, representing from our samples, what is actually the case. Since they are close, we can see that our standard errors are quite indicative of the true variance between samples, which is good, as we can place trust in using them instead of the true sd, which is often unavailable.

Problem 10

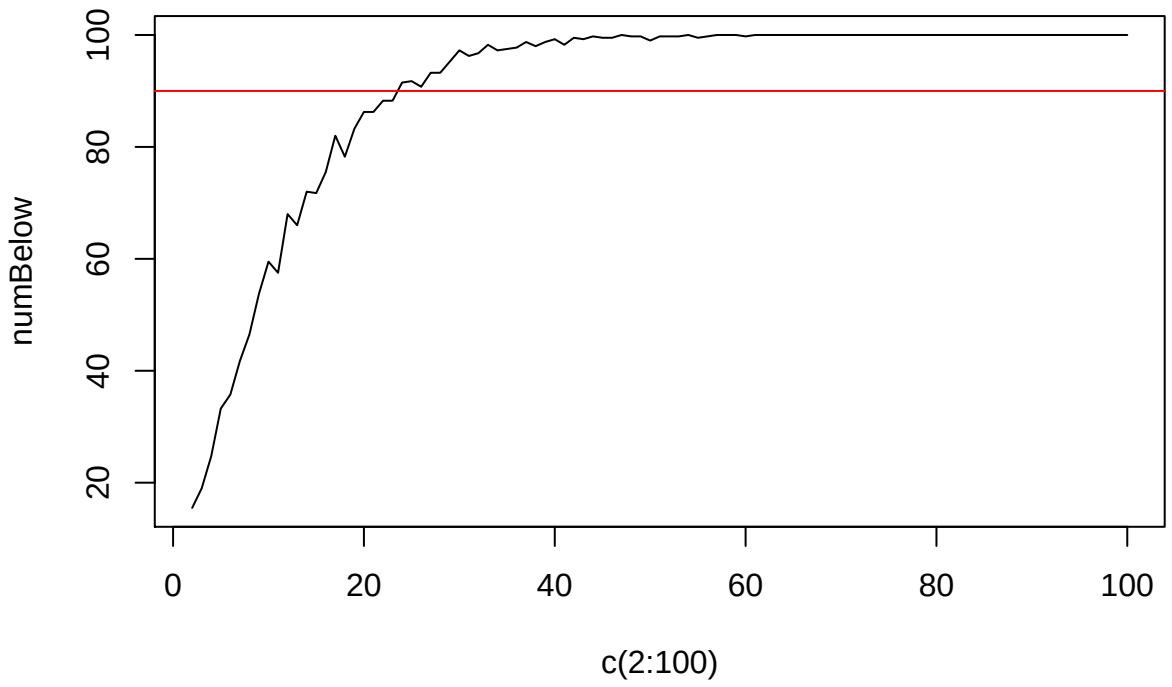
```
numSamples=400
iqSamples = replicate(numSamples,rnorm(n,90,15))
allSEs = apply(iqSamples,FUN=confInterval,MARGIN=2,doTrueVar = T)
res = getAltResults(allSEs,100)
percentBelow = mean(res)*100
print(percentBelow)
```

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

Considering the size of our sample, it would take almost 8 standard errors to get up to 100 on the upper bound assuming a mean around 90. Basically, a lot would have to go wrong from this sampling to have an upper bound above 100. Consequently, we can safely say that sampling from a normal distribution with mean 90 and SD 15 will rarely ever produce a mean around 100.

Problem 11

```
numBelow=numeric(99)
for(i in 2:100)
{
  numSamples=400
  iqSamples = replicate(numSamples,rnorm(i,90,15))
  allSEs = apply(iqSamples,FUN=confInterval,MARGIN=2,doTrueVar = T)
  res = getAltResults(allSEs,100)
  percentBelow = mean(res)*100
  numBelow[i-1] = (percentBelow)
}
plot(x=c(2:100),y=numBelow,type="l")
abline(h=90,col='red')
```



We can study this, and we see that a sample size of about 25 or so will be satisfactory for a 90% below rate.

Problem 12

```
count = sum(100>iqs)
estProp = count/n
estSE = sqrt(estProp*(1-estProp))/sqrt(n)
print(estProp-1.96*estSE)
```

```
## [1] 0.6826859
```

```
print(estProp+1.96*estSE)
```

```
## [1] 0.8334431
```

we see that our confidence interval does not inclue .5, which is siginifigant, as that means the proportion of students above and below 100, which would be expected to be 50/50 for a normal centered around 100, is

likely not the case for this sample of students.

Problem 13

Bernoulli sampling

```
numSamples=400
props = replicate(numSamples,rbinom(n,1,.5))
allSEs = calcPropSEs(props,n,numSamples)
allProps = apply(props, MARGIN=2, FUN = mean)
numPass = testIntervals(allProps,allSEs,numSamples,0.5)
print((numPass/numSamples)*100)
```

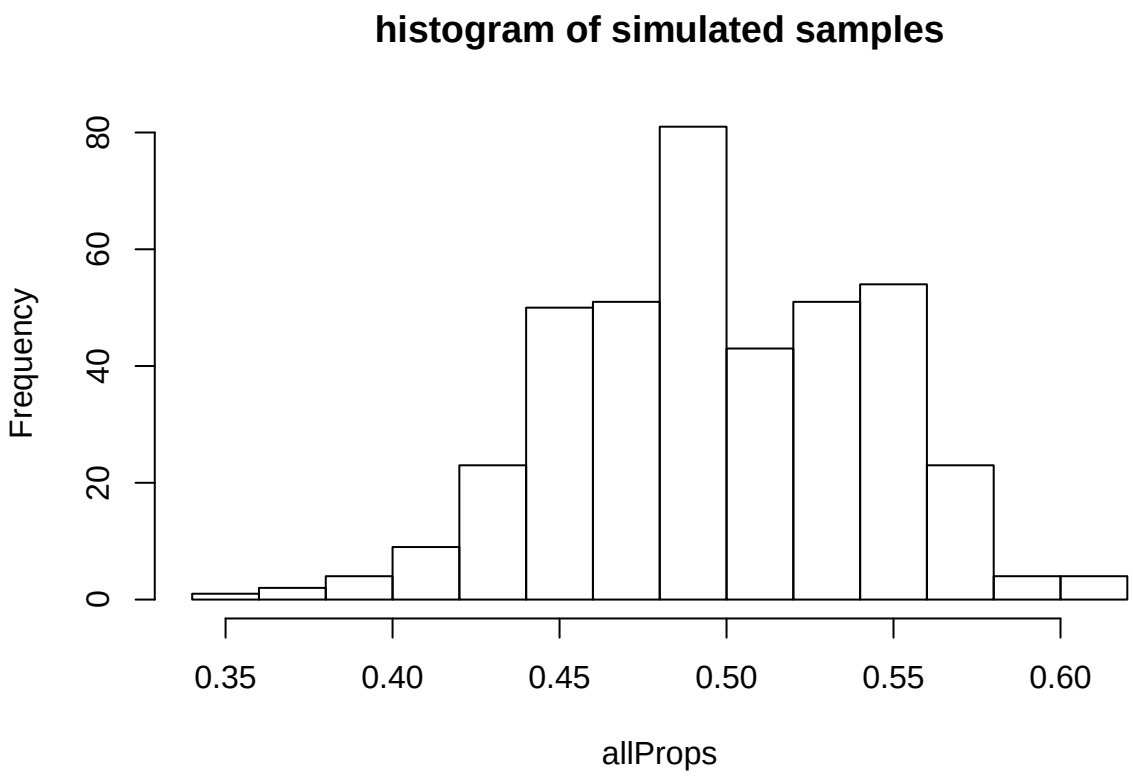
```
## [1] 96
```

```
#trueSEs = apply(props,margin=2,FUN=sd)
#trueSEs = calcPropSEs(props,n)
```

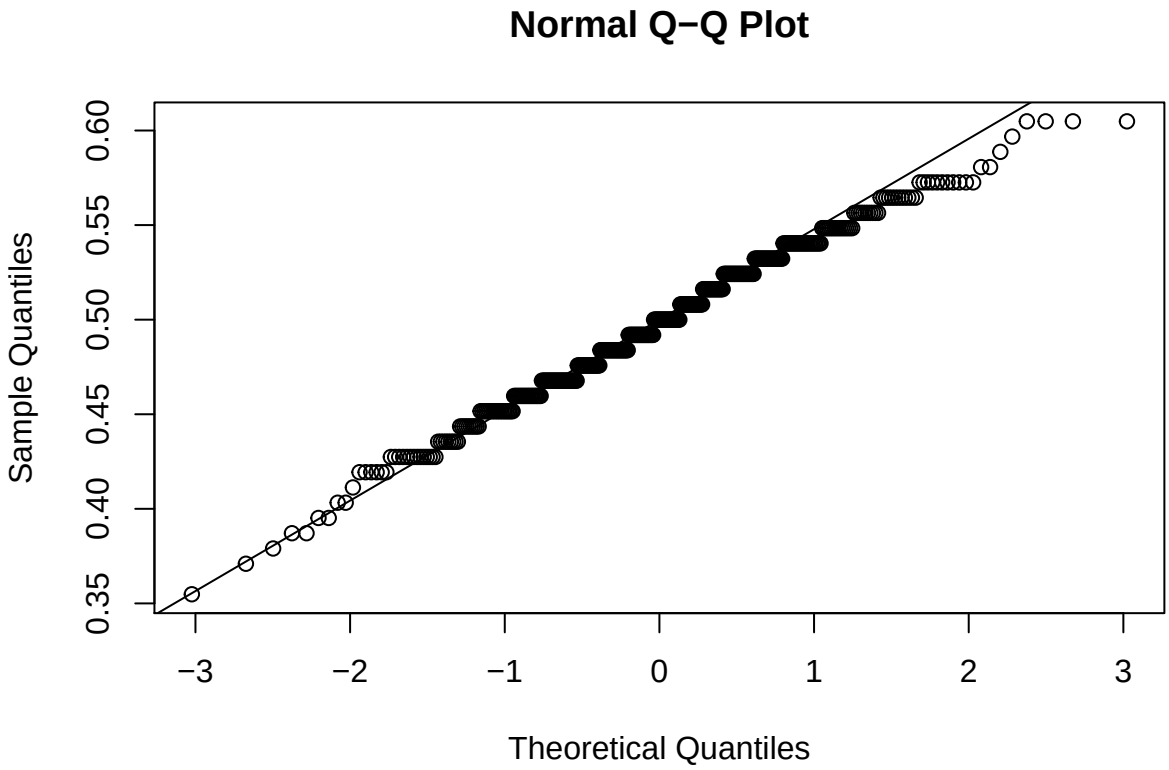
As we can see, we do just about achieve that 95% confidence interval, as we would expect.

Problem 14

```
hist(allProps,main="histogram of simulated samples")
```



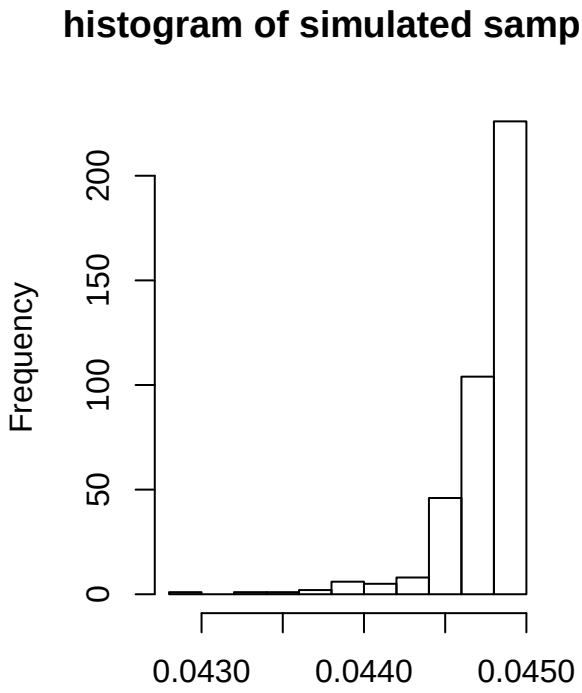

```
qqnorm(allProps)
qqline(allProps)
```

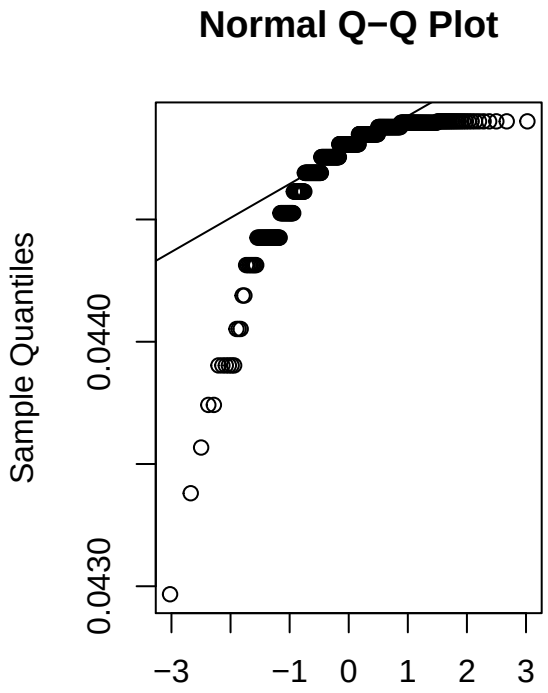


I was expecting a normal distribution, and it almost looks quite normal, however, there are some disturbances in the expected flow of the histograms supposed bell shape curvature. The qqplot does however display a rather indicative straight line of a normal distribution.

Problem 15

```
par(mfrow=c(1,2))
hist(allSEs,main="histogram of simulated samples")
qqnorm(allSEs)
qqline(allSEs)
```






```
print(mean(allSEs))
```

```
## [1] 0.04472372
```

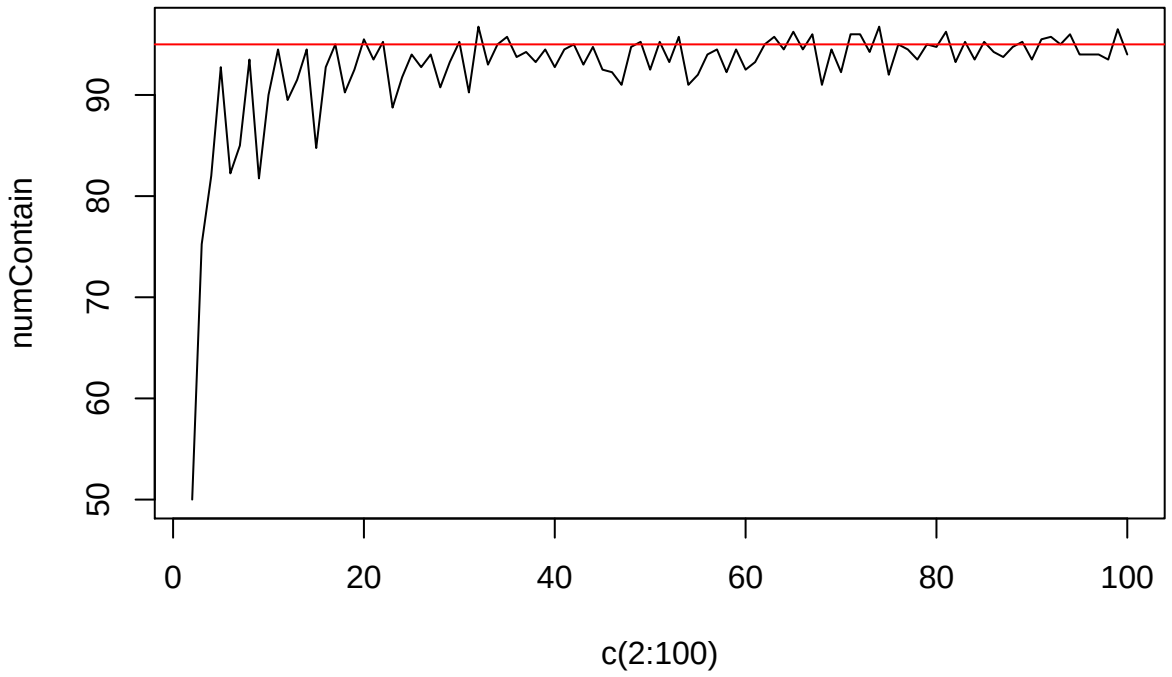
```
print(sd(allProps))
```

```
## [1] 0.04434258
```

Well, this is not what I was expecting! This is a heavily skewed distribution, and clearly not even close to normal. Reminds me of a log scale(at least the qqplot does). The histogram more resembles some exponential growth with a boundary condition. However, the standard deviation of all the simulated proportions and the mean of the estimated SEs are quite close. Which is important, for the same reasons as listed in problem 9

Problem 16

```
numContain=numeric(99)
for(i in 2:100)
{
  numSamples=400
  props = replicate(numSamples,rbinom(i,1,.5))
  allSEs = calcPropSEs(props,i,numSamples)
  allProps = apply(props, MARGIN=2, FUN = mean)
  numPass = testIntervals(allProps,allSEs,numSamples,0.5)
  percentBelow = (numPass/numSamples)*100
  numContain[i-1] = (percentBelow)
}
plot(x=c(2:100),y=numContain,type="l")
abline(h=95,col='red')
```



It appears that a sample size of size 40 satisifies this requirement

Problem 17

Well we understand that the formula to calculate the standard error for a sample proportion is $\frac{\sqrt{\hat{p}*(1-\hat{p})}}{\sqrt{n}} = SE$.

In order for our width, which is $2*1.96*SE$, to have a width of .1 or less, we must satisfy $1.96*\frac{\sqrt{\hat{p}*(1-\hat{p})}}{\sqrt{n}} \leq 0.1$. Solving for n, we get the following $n \geq (10*1.96*2*\sqrt{\hat{p}*(1-\hat{p})})^2$. If $\hat{p} = .5$ then $n \geq 385$, as n approaches 0 or 1, this number only decreases, thus, we are garunteed this theoretically at 385.

NOTE: I apologize for any typos. there is no spell check in R markdown apparently.