homework2

Alexander Van Roijen

January 17, 2019

Problem 1

Problem 1.1

```
vals = c(-5, -2, -1, -1, 0, 0, 2, 3, 4, 4, 5, 5, 6, 6, 11)
dft = length(vals)
sampleMean = mean(vals)
sampleSD = sd(vals)
se = sampleSD/sqrt(dft)
widthFactor = qt(0.975,dft-1)
upper = sampleMean+widthFactor*se
lower = sampleMean - widthFactor*se
print(upper)
```

```
## [1] 4.700176
```

```
print(lower)
```

```
## [1] 0.2331574
```

For this hypothesis test, we are checking the following $H_0 : \mu = 0$ if $qt(0.975, 14) = \frac{|\bar{X}-0|}{\frac{s}{\sqrt(15)}}$ where $\bar{X} = sampleMean$ from above. plugging in everything we get

```
print(sampleMean/(se))
```

```
## [1] 2.368682
```

```
print(widthFactor)
```

```
## [1] 2.144787
```

We see this still agrees that we should reject the null that the mean is 0 as our CI does not contain 0 and our tabulated Z is greater than the critical value for a t distribution with a alpha of 0.05 and 14 degrees of freedom. This agrees with our solutions found in problem 2.2. This is also know as a 1-sample t test.

Problem 1.2

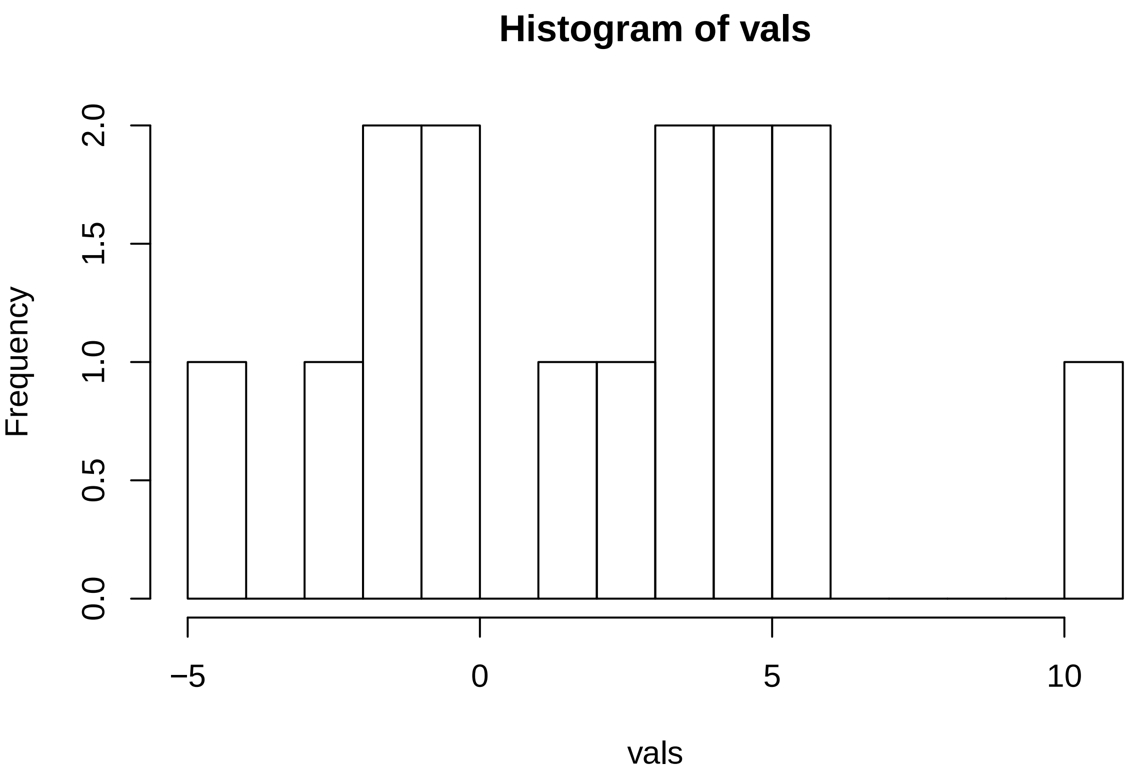
```
numRepl=100
samples = replicate(numRepl,runif(15,min=-7,max=7))
allMeans = apply(samples, MARGIN=2, FUN = mean)
numOff=0
lower = -se*2.14
upper = se*2.14
for(i in 1:numRepl)
{
  if(allMeans[i]<lower || allMeans[i]>upper)
  {
    numOff = numOff+1
  }
}
print(numOff/numRepl)
```

```
## [1] 0.03
```

In this example, I used a uniform distribution. However, in the previous problem I sampled from an incorrect distribution! (it didn't satisfy the variance requirement). Accordingly, I achieved a simulated type I error of 12%, while in this one it is much lower. Consequently ,this uniform distribution is doing a better job at simulating the data sampled. We can say this test is quite more valid as it simulates a type one error much closer to the true data we have. Although, I would say it is still a bit too far off to accurately estimate the distribution for further tests, but its the best we have so far.

Problem 1.3

```
{
hist(vals,breaks=seq(-5,11,1))
ss=15
numIteration=100
allSamples = numeric(numIteration)
allSamples=replicate(numIteration,mean(sample(c(-3,5),size=ss,replace=TRUE)))
SE = sqrt(15/ss)
lower = -1.96*SE
upper = 1.96*SE
numOutside=0
for(i in 1:60)
{
  if(allSamples[i]>upper || allSamples[i]<lower)
  {
    numOutside=numOutside+1
  }
}
print((numOutside/100))
}
```



```
## [1] 0.06
```

Inspired by my last attempt, I decided to change the bounds of my dichotomous sampling. Re-“eye balling” it from the histogram, I chose an average of -3 and 5. As we can see, and as was aforementioned, this was a bit aggressive and does not produce a very good type I error and is likely not a good distribution to emulate the original. However, I will say that I wasn’t too far off.

Problem 1.4

```
vals = c(-5, -2, -1, -1, 0, 0, 2, 3, 4, 4, 5, 5, 6, 6)
dft = length(vals)
sampleMean = mean(vals)
sampleSD = sd(vals)
se = sampleSD/sqrt(dft)
widthFactor = qt(0.975,dft-1)
upper = sampleMean+widthFactor*se
lower = sampleMean - widthFactor*se
print(paste("Upper CI Bound: ",upper,""))
```

```
## [1] "Upper CI Bound:  3.81654597625001 "
```

```
print(paste("Lower CI Bound: ",lower,""))
```

```
## [1] "Lower CI Bound:  -0.102260261964292 "
```

```
print(paste("T value: ",sampleMean/se,""))
```

```
## [1] "T value:  2.04762010431697 "
```

```
print(paste("Crit value: ",widthFactor,""))
```

```
## [1] "Crit value:  2.16036865646279 "
```

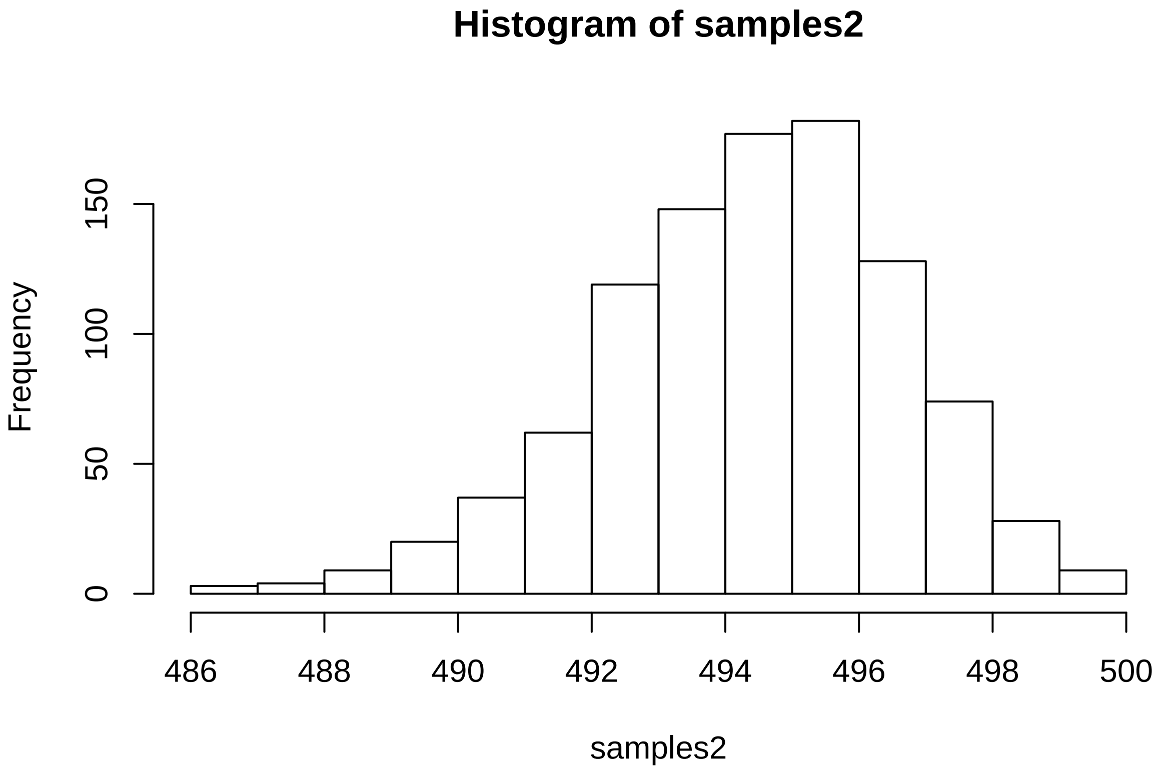
Clearly, if we assume the 11 was in fact an error, we see that we can no longer reject the null hypothesis, both due to removing a high value away from zero and reducing our degrees of freedom by 1 once more. Thus extending the range of our confidence interval to include 0, and our t value falling below the critical value.

Problem 2

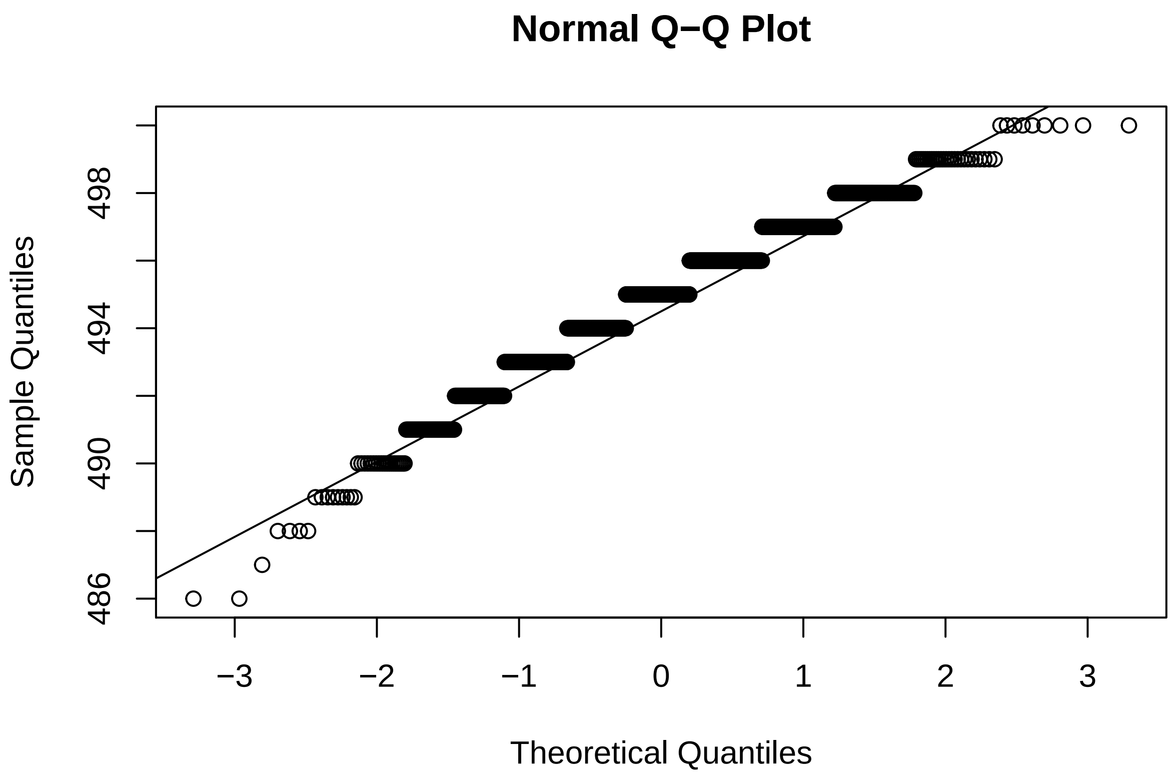
Problem 2.1

I initially thought this could have been done using the CLT. Looking at the plots below, there is certainly skew in the first histogram representing the distribution of proportions sampled. However, the z values approximated from these values look quite normal with a good distribution of those values. Note this only applies to the 500 level. For these first few problems, there will be some mention of this normal approximation. We will see how this wont apply in the case of only 100 samples.

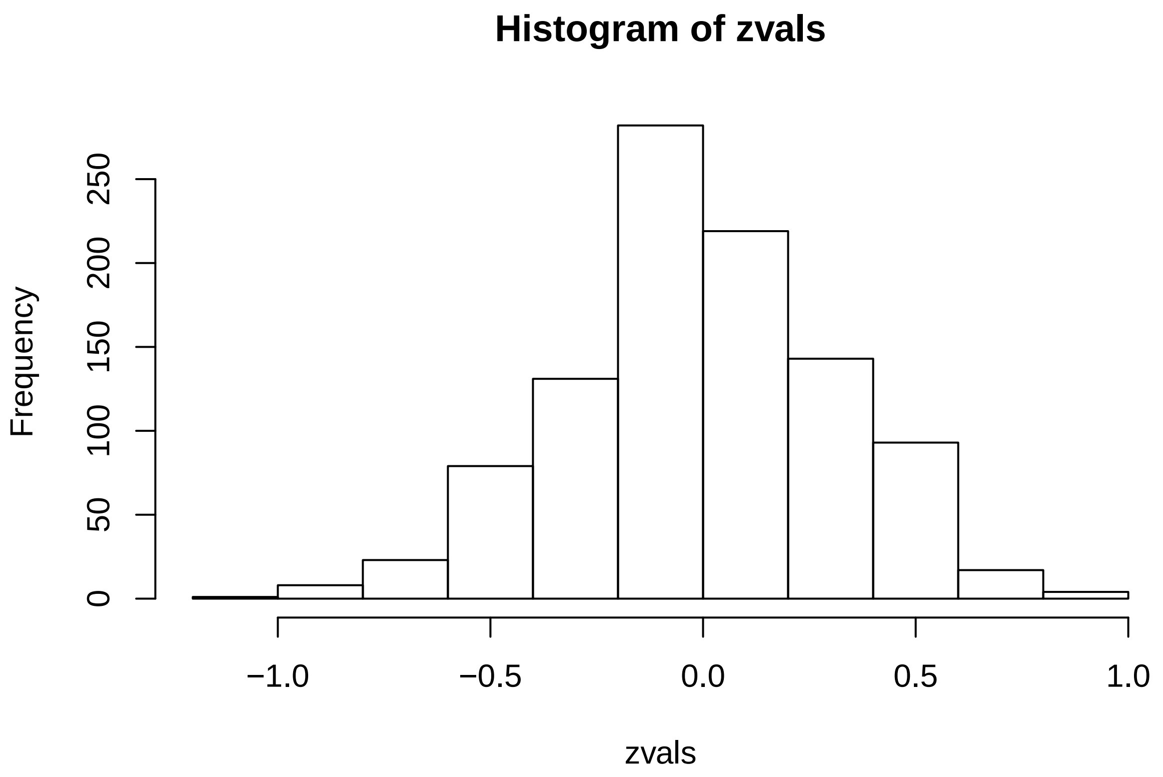
```
numSimu = 1000
numSample = 500
critP = 494/numSample
p_knot = .99
samples=replicate(numSimu,rbinom(10,numSample,p_knot))
samples2=rbinom(numSimu,numSample,p_knot)
hist(samples2)
```



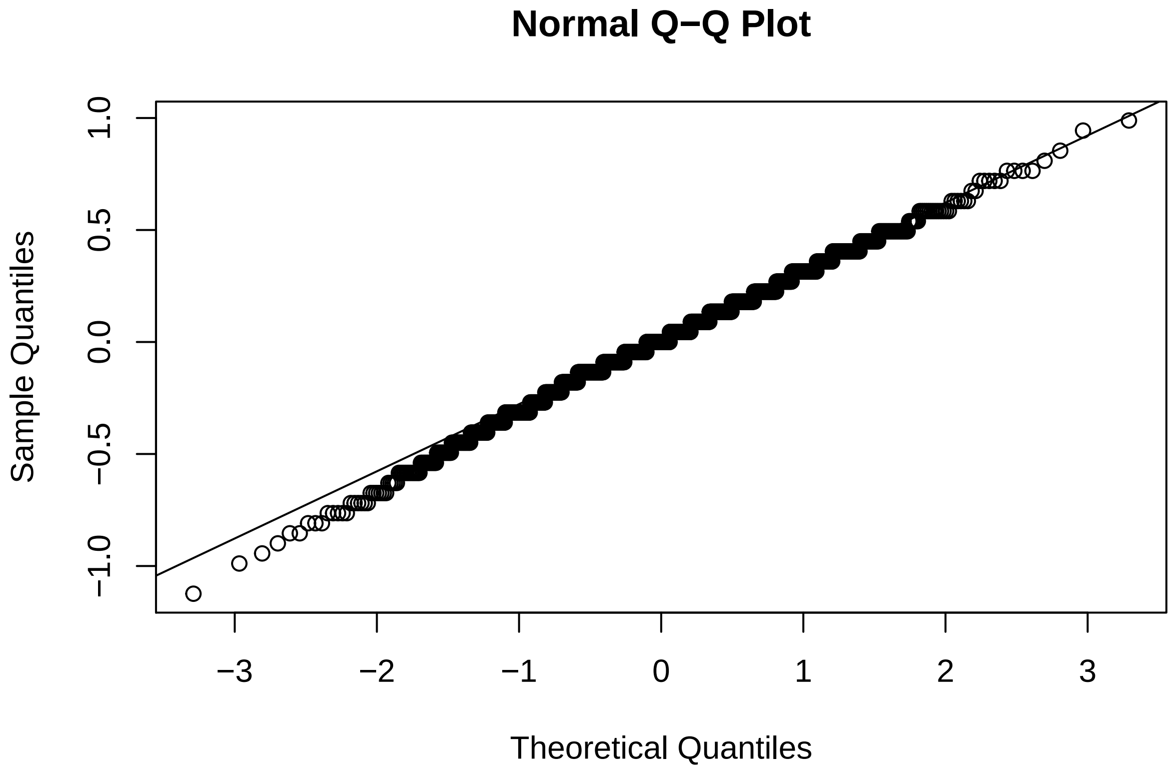
```
qqnorm(samples2)
qqline(samples2)
```



```
allMeans = apply(samples, MARGIN=2, FUN = mean)
zvals = numeric(numSimu)
trueSD = sqrt((p_knot*(1-p_knot)/numSample))
for (i in 1:numSimu)
{
  zvals[i] = (((allMeans[i]/numSample)-p_knot)/trueSD)
}
hist(zvals)
```



```r
qqnorm(zvals)
qqline(zvals)
```



```r
print(sum(zvals<0))
```

```
## [1] 459
```

```r
print(sum(zvals>0))
```

```
## [1] 476
```

If we were to assume the normal can be used to approximate our proportions from the binomial distribution, we would use the following to solve for our significance level: $\frac{|\hat{p}-.99|}{\frac{\sigma}{\sqrt{n}}} > C$ where C is our critical Value. If we were to solve for X, we would have $n * |\bar{X}| < 494$. We notice this is a one sided test. solving for $\bar{X}$ we get $\hat{p} = .988$. We plug this in to get our SE $\sqrt{\frac{\hat{p}(1-\hat{p})}{n}} = 0.00444$. With all this calculated, and plugging back in, we get the following critical value

```r
numSample=500
trueSD = sqrt((p_knot*(1-p_knot)/numSample))
#trueSD = sqrt((.988*(1-.988)/numSample))
altZ=(.988-p_knot)/(sqrt(p_knot*(1-p_knot)/numSample))
ZVAL = (critP-p_knot)/trueSD
print(ZVAL)
```

```
## [1] -0.4494666
```

```r
print(pnorm(ZVAL))
```

```
## [1] 0.3265476
```

The results show a pretty terrible significance level of about 33%! However we will use the simpler results from the density of the binomial distribution we know this data comes from to determine the true significance level.

```r
print(sum(dbinom(c(0:494),numSample,p_knot)))
```

```
## [1] 0.3840379
```

Thus we can see that we have an approximate 38% probability of achieving a type I error, where we reject the Null hypothesis when it is in fact true. This is somewhat close to the normal approximation that was derived earlier, but still quite different. However, both results for the type one error are not very good.

Problem 2.2

Now we want type I error very close to 0.05, so, if it were normal as I assumed prior, $P(\frac{|\hat{p}-.99|}{\frac{\sigma}{\sqrt{n}}} < -1.644854)$

```r
cval = qnorm(0.05)
res = (cval * trueSD)+.99
print(res*numSample)
```

```
## [1] 491.3404
```

Under this assumption of a normal distribution, we would say if one were to miss 9 or more from the 500 samples, we would reject the null hypothesis when we shouldn't as close to 5% of the time as we possibly can. Now we will look at the results using the binomial distribution to determine the rejection region for this significance.

```r
for( i in 1:500)
{
  if(sum(dbinom(c(0:i),numSample,p_knot))>0.05)
  {
    print(i-1)
    break
  }
}
```

```
## [1] 490
```

```
print(sum(dbinom(c(0:489),numSample,p_knot)))
```

```
## [1] 0.01324357
```

```
print(sum(dbinom(c(0:490),numSample,p_knot)))
```

```
## [1] 0.03110211
```

```
print(sum(dbinom(c(0:491),numSample,p_knot)))
```

```
## [1] 0.06711016
```

Accordingly, we see that we need to see more misses in our test in order to achieve this type I error. More precisely, we need to miss 490 out of 500 samples in order to have this confidence in rejecting the null hypothesis and only being wrong as close as possible to 5% of the time. This is slightly different, but is significantly different than the normal approximation shown above.

Problem 2.3

If we got 490 out of 500 samples correctly identified, we would reject this hypothesis. We can calculate the p value below

```
print(sum(dbinom(c(0:490),numSample,p_knot)))
```

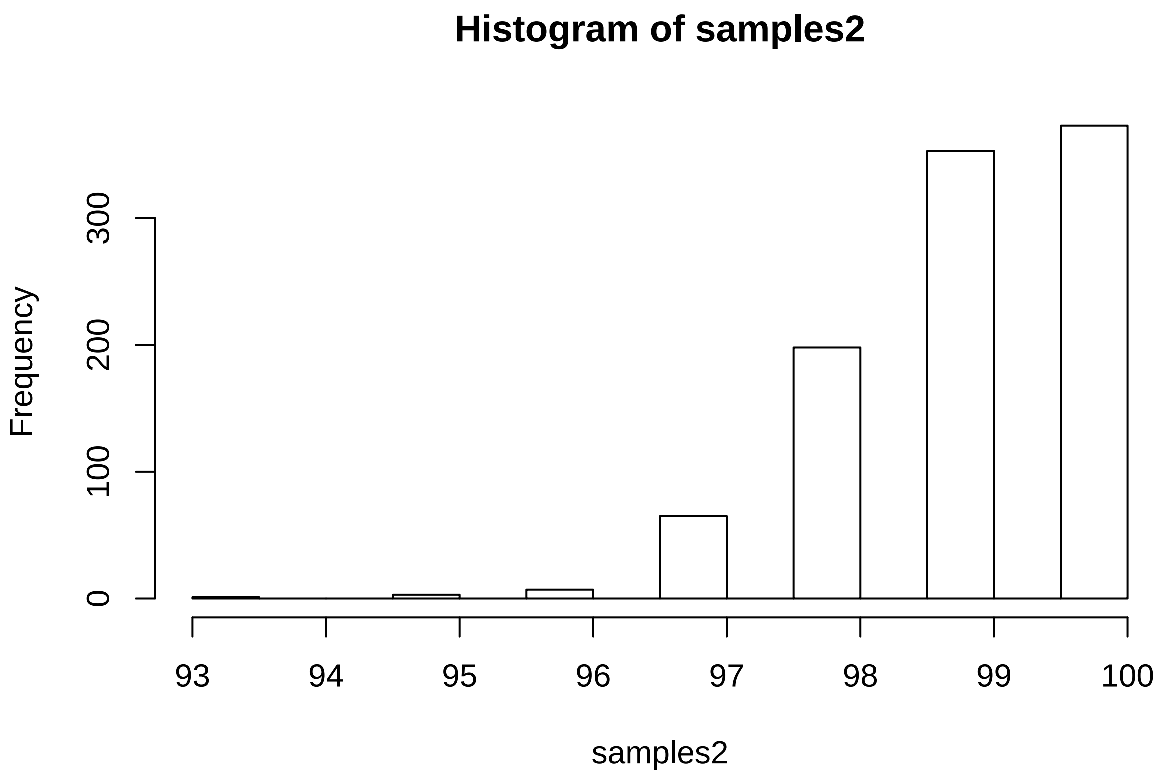
```
## [1] 0.03110211
```

As we can see, and what was assumed before, we set the rejection region to include this exact value. Thus, we reject 490 out of 500 samples. We would have also rejected the null hypothesis if we were to follow the results from the normal distribution assumed prior.

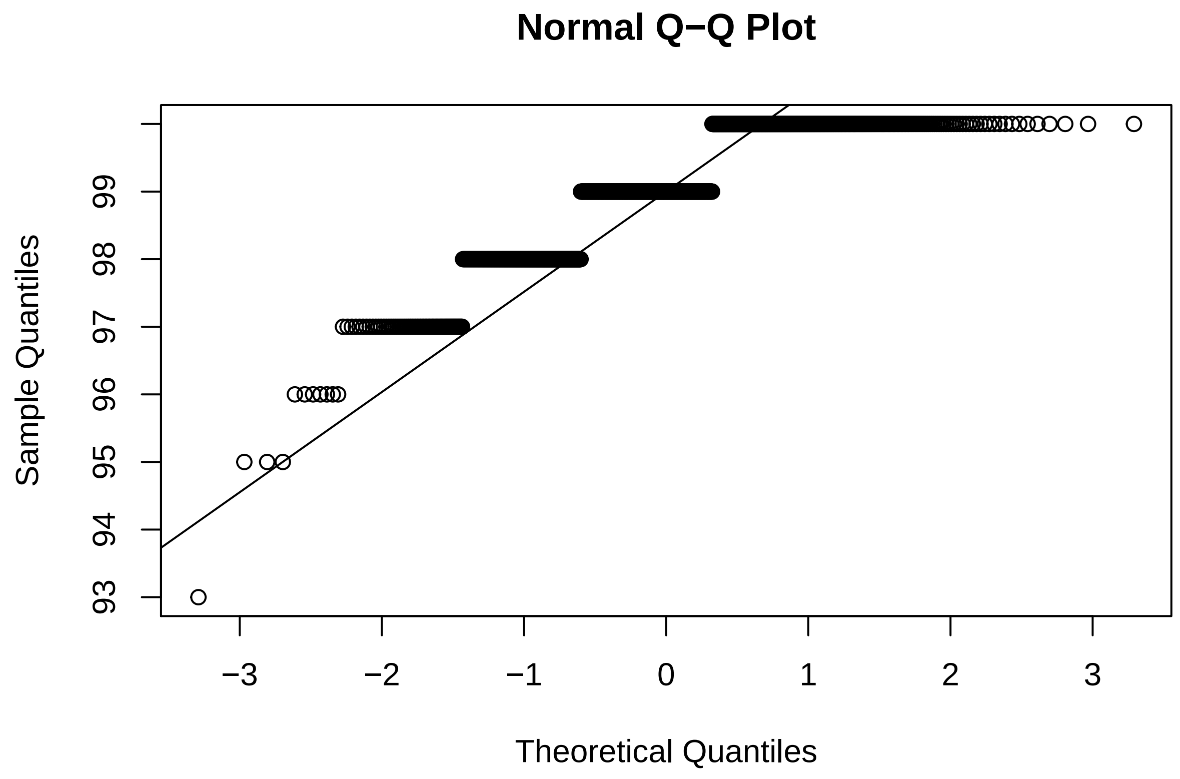
Another note on our sample size.

By decreasing the sample size, as shown below, we will have an even more skewed distribution, showing that we can with even greater confidence not assume an approximate normal distribution, unlike the case of 500 samples which is much closer to a normal distribution.

```
numSimu = 1000
numSample = 100
critP = 98/numSample
p_knot = .99
samples=replicate(numSimu,rbinom(10,numSample,p_knot))
samples2=rbinom(numSimu,numSample,p_knot)
hist(samples2)
```

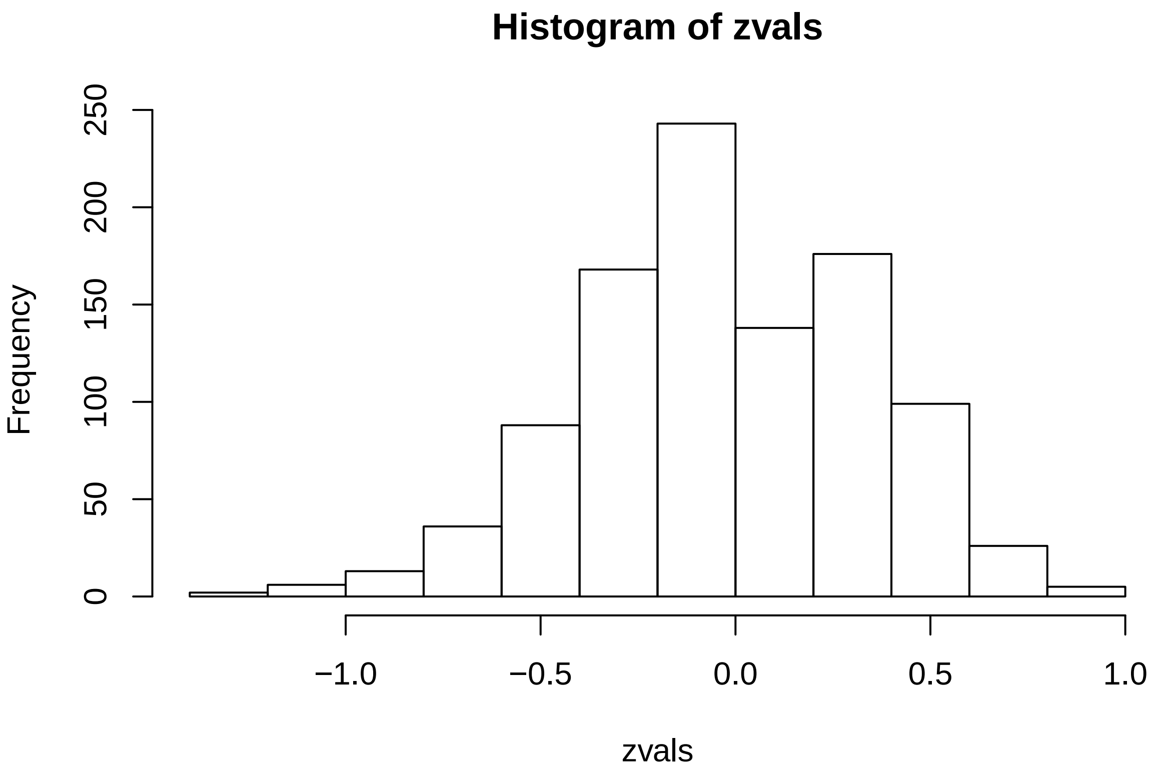


```
qqnorm(samples2)
qqline(samples2)
```

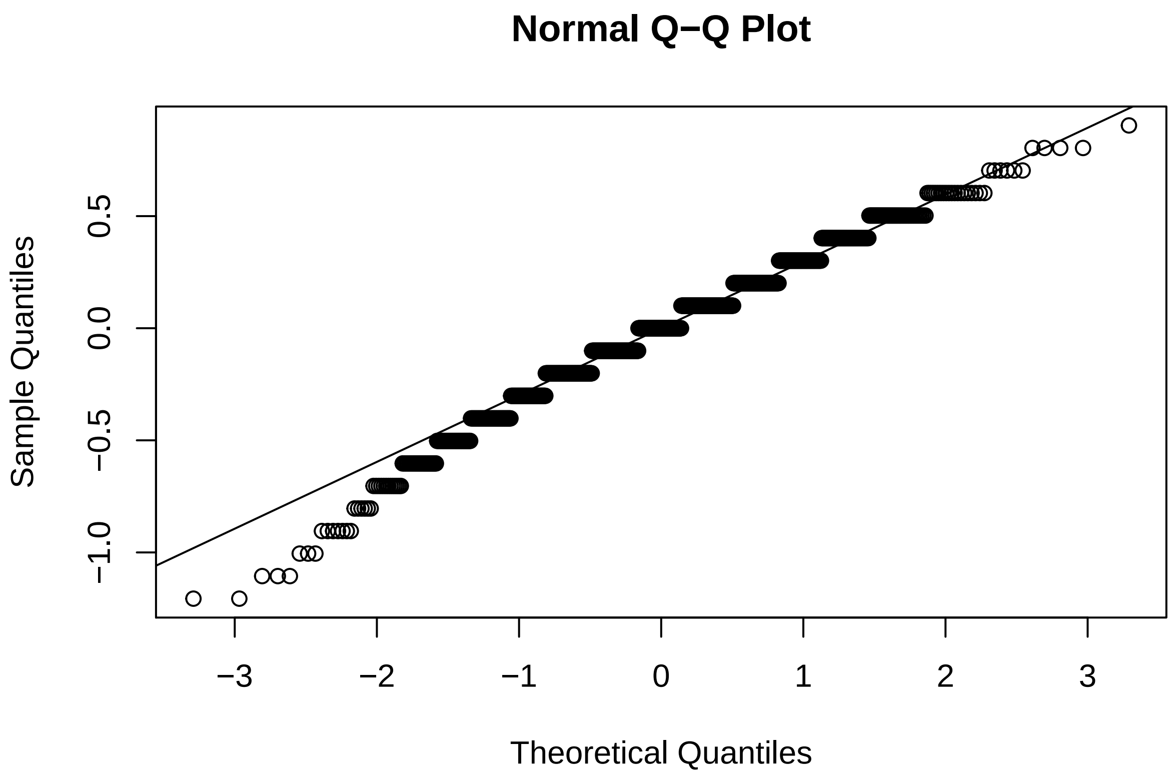


```
allMeans = apply(samples, MARGIN=2, FUN = mean)
zvals = numeric(numSimu)
```

```r
trueSD = sqrt((p_knot*(1-p_knot)/numSample))
for (i in 1:numSimu)
{
  zvals[i] = (((allMeans[i]/numSample)-p_knot)/trueSD)
}
hist(zvals)
```



```r
qqnorm(zvals)
qqline(zvals)
```



```
print(sum(zvals<0))
```

```
## [1] 436
```

```
print(sum(zvals>0))
```

```
## [1] 444
```

Problem 2.4

With a reduced sample size, we would expect to see a wider interval for our rejection region as our standard error will increase

```
numSample2=100
for( i in 1:numSample2)
{
  if(sum(dbinom(c(0:i),numSample2,p_knot))>0.05)
  {
    print(i-1)
    break
  }
}
```

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

```
print(sum(dbinom(c(0:95),numSample2,p_knot)))
```

```
## [1] 0.003432322
```

```
print(sum(dbinom(c(0:96),numSample2,p_knot)))
```

```
## [1] 0.01837404
```

```
print(sum(dbinom(c(0:97),numSample2,p_knot)))
```

```
## [1] 0.0793732
```

As expected, we must drop 4 units, to 96, in order to have a number of samples such that the significance level is not greater than 0.05. Thus any value less than or equal to 96 would be rejected in this case. Comparing the change, 99 to 96 is about a 3 % decrease to meet the criteria, meanwhile, the change from 495 to 490 is only a 1% change.

Problem 2.5

The p value represents the probability that we would see a value of equal or greater in value than what we have already realized. We already calculated the values above, but to restate; the calculation will be done below

```
numSample = 100
numSample2 = 500
truep = .99
samplep = 0.98
sdtop = truep*(1-truep)
sd1 = sqrt(sdtop/numSample)
sd2 = sqrt(sdtop/numSample2)
testDiffs = samplep-truep
zval1 = testDiffs/sd1
zval2 = testDiffs/sd2
#print(pnorm(zval1))
#print(pnorm(zval2))
print(sum(dbinom(c(0:490),numSample2,truep)))
```

```
## [1] 0.03110211
```

```
print(sum(dbinom(c(0:98),numSample,truep)))
```

```
## [1] 0.264238
```

As we expected, the reduced sample size is having a huge impact on our p values. Despite the sample proportions representing the same ratio, the p values are much different. In the case of 500 samples, we are below our significance level and would reject the null hypothesis. However, with the case of only 100 trials, we are not even close to the significance level and would fail to reject the null hypothesis. In general, with the reduced number of samples, the probability of seeing a more extreme value with less samples for the same proportion is much more likely due to the discrete nature of our distribution.

Problem 3

```
iqData = read.csv('/home/bdvr/Documents/GitHub/Data557/Week1/Homework/iq.csv')
n = length(iqData$IQ)
trueSD = 15
iqs = iqData$IQ
```

We are solving $H_0 : \mu = 100$ if $qt(0.975, n) = \frac{|\bar{X}-100|}{\frac{s}{\sqrt{(n)}}}$ for $\bar{X}$

Problem 3.1

```r
trueMean=100
twoSides = qt(0.975,n-1)
dataSD = sd(iqs)
print(dataSD)
```

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

```r
print(twoSides)
```

```
## [1] 1.979439
```

Plugging in our previous value and solving for $\bar{X}$ we get $\bar{X} > 1.97928 * \frac{14.40393}{\sqrt{124}} + 100$ and $\bar{X} < -1.97928 * \frac{14.40393}{\sqrt{124}} + 100$ we get

```r
lower = trueMean- ((twoSides*dataSD)/sqrt(n))
upper = trueMean+ ((twoSides*dataSD)/sqrt(n))
print(lower)
```

```
## [1] 97.43957
```

```r
print(upper)
```

```
## [1] 102.5604
```

```r
sampleMean = mean(iqs)
print(sampleMean)
```

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

```r
zval = (sampleMean - trueMean) / (dataSD/sqrt(n))
print(zval)
```

```
## [1] -6.895462
```

```r
print(2*pt(zval,n-1))
```

```
## [1] 2.486475e-10
```

We see that our mean of iqs is outside of the rejection region at the 0.05 significance level and thus say we have enough evidence to reject the null hypothesis. Accordingly, our pvalue is also very very small, which makes sense as our mean is very far away from the rejection region. the probability of seeing a more extreme value is quite small.

Problem 3.2

WE are now going to assume that our sample mean is the null hypothesis in a way, and establishing our confidence interval around it and see what is the 2 sided 95% confidence interval that our true mean falls within the region.

```r
lower = sampleMean- ((twoSides*dataSD)/sqrt(n))
upper = sampleMean+ ((twoSides*dataSD)/sqrt(n))
print(lower)
```

```
## [1] 88.52022
```

```r
print(upper)
```

```
## [1] 93.64107
```

As we can see, under the 95% CI, we believe that the true mean should fall within this range, however, the mean of 100 does not fall within this range, which agrees with out previous conclusion.

Problem 3.3

By increasing the requirement of our confidence interval and similarly decreasing the significance level, we expect to see a wider range of our acceptance region and our confidence interval, as we are requiring the probability of us rejecting the null hypothesis when it is true to be even smaller. Similarly, for the confidence interval, it will be larger as we want to expand the chance of us finding the true mean within the interval.

```
twoSides = qt(0.995,n-1)
lower = trueMean- ((twoSides*dataSD)/sqrt(n))
upper = trueMean+ ((twoSides*dataSD)/sqrt(n))
print(paste("Lower rejection region bound:" ,lower))
```

```
## [1] "Lower rejection region bound: 96.6156688598188"
```

```
print(paste("upper rejection region bound:" ,upper))
```

```
## [1] "upper rejection region bound: 103.384331140181"
```

```
sampleMean = mean(iqs)
zval = (sampleMean - trueMean) / (dataSD/sqrt(n))
print(paste("P value:" ,2*pt(zval,n-1)))
```

```
## [1] "P value: 2.48647459656314e-10"
```

```
lower = sampleMean- ((twoSides*dataSD)/sqrt(n))
upper = sampleMean+ ((twoSides*dataSD)/sqrt(n))
print(paste("Lower confidence interval bound:" ,lower))
```

```
## [1] "Lower confidence interval bound: 87.6963140211091"
```

```
print(paste("Upper confidence interval bound:" ,upper))
```

```
## [1] "Upper confidence interval bound: 94.4649763014715"
```

Looking at the output above, as expected, the regions have expanded, but not enough to prevent us from rejecting the null hypothesis, suggesting a pvalue that is greater than the significance level. Similarly, we have a wider confidence interval with a 99% chance of including the true mean, but 100 is still not within this interval.

Problem 3.4

```
n=124
twoSides = qt(0.975,n-1)
lower = trueMean- ((twoSides*dataSD)/sqrt(n))
upper = trueMean+ ((twoSides*dataSD)/sqrt(n))
numSimulations = 500
allConclusions = numeric(500)
tempMean=0
for(i in 1:numSimulations)
{
  tempMean = mean(rnorm(n,100,15))
  #print(tempMean)
  if((tempMean < lower) | (tempMean>upper))
  {
    allConclusions[i]=1
  }
  else
  {
    allConclusions[i]=0
  }
}
print(mean(allConclusions))
```

```
## [1] 0.062
```

As expected, we see that the percentage of means that fall outside of our rejection region matches very closely with the significance level defined.

Problem 3.5

fix this 2

```
twoSides = qt(0.975,n-1)
lower = trueMean- ((twoSides*dataSD)/sqrt(n))
upper = trueMean+ ((twoSides*dataSD)/sqrt(n))
numSimulations = 500
allConclusions = numeric(500)
for(i in 1:numSimulations)
{
  tempMean = mean(rnorm(n,95,15))
  if(tempMean > lower & tempMean<upper)
  {
    allConclusions[i]=1
  }
  else
  {
    allConclusions[i]=0
  }
}
print(1-mean(allConclusions))
```

```
## [1] 0.962
```

We have a power of ~96-97% in the case of an alternative hypothesis of mean 95. This makes sense with our previous findings of the confidence interval and rejection region including only as low as 96.6, which means many of these means will fall outside of this region, as to be expected.

Problem 3.6

```

twoSides = qt(0.975,n-1)
lower = trueMean- ((twoSides*dataSD)/sqrt(n))
upper = trueMean+ ((twoSides*dataSD)/sqrt(n))
numSimulations = 500
allConclusions = numeric(500)
altMean = 90
power=1
```

```r
while(power >=0.9)
{
  for(i in 1:numSimulations)
  {
    tempMean = mean(rnorm(n,altMean,15))
    if(tempMean > lower & tempMean<upper)
    {
      allConclusions[i]=1
    }
    else
    {
      allConclusions[i]=0
    }
  }
  power=(1-mean(allConclusions))
  if(power<.99)
  {
    print(paste("Alternative Mean: ",altMean))
    print(paste("Power: ",power))
  }
  altMean=altMean+0.1
}
```

```
## [1] "Alternative Mean:  94.1999999999998"
## [1] "Power:  0.988"
## [1] "Alternative Mean:  94.3999999999997"
## [1] "Power:  0.982"
## [1] "Alternative Mean:  94.5999999999997"
## [1] "Power:  0.972"
## [1] "Alternative Mean:  94.6999999999997"
## [1] "Power:  0.964"
## [1] "Alternative Mean:  94.7999999999997"
## [1] "Power:  0.976"
## [1] "Alternative Mean:  94.8999999999997"
## [1] "Power:  0.976"
## [1] "Alternative Mean:  94.9999999999997"
## [1] "Power:  0.976"
## [1] "Alternative Mean:  95.0999999999997"
## [1] "Power:  0.96"
## [1] "Alternative Mean:  95.1999999999997"
## [1] "Power:  0.95"
## [1] "Alternative Mean:  95.2999999999997"
## [1] "Power:  0.94"
## [1] "Alternative Mean:  95.3999999999997"
## [1] "Power:  0.93"
## [1] "Alternative Mean:  95.4999999999997"
## [1] "Power:  0.9"
## [1] "Alternative Mean:  95.5999999999997"
## [1] "Power:  0.906"
## [1] "Alternative Mean:  95.6999999999997"
## [1] "Power:  0.884"
```

It appears that the closest whole number we can approximate as an alternative mean with power of at least .9 would be 95. If we allow for decimal places, we can say a mean of about 95.7 is the largest alternative mean we can achieve with a power of at least .9. However, as aforementioned, we cant and thus stick with our value of 95 as the largest number for our alternative hypothesis with a power of at least .9