S520 Problem Set 7 Solutions

Arturo Valdivia

Due on 02/27/2023

1.

Using R, we get:

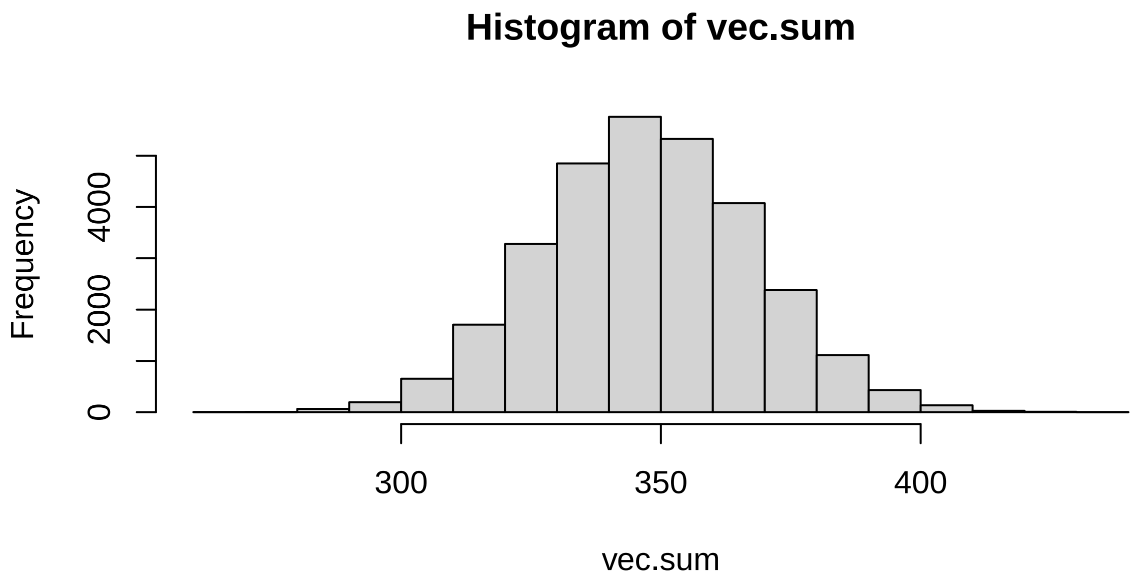
```r
# 1a
urn = c(3,3,3,4,4,7,7,7,10,10)
```

```r
# 1b
set.seed(520)
y = sum(sample(x = urn, size = 60, replace = T))
x.bar = mean(sample(x = urn, size = 60, replace = T))
c(y, x.bar)
```

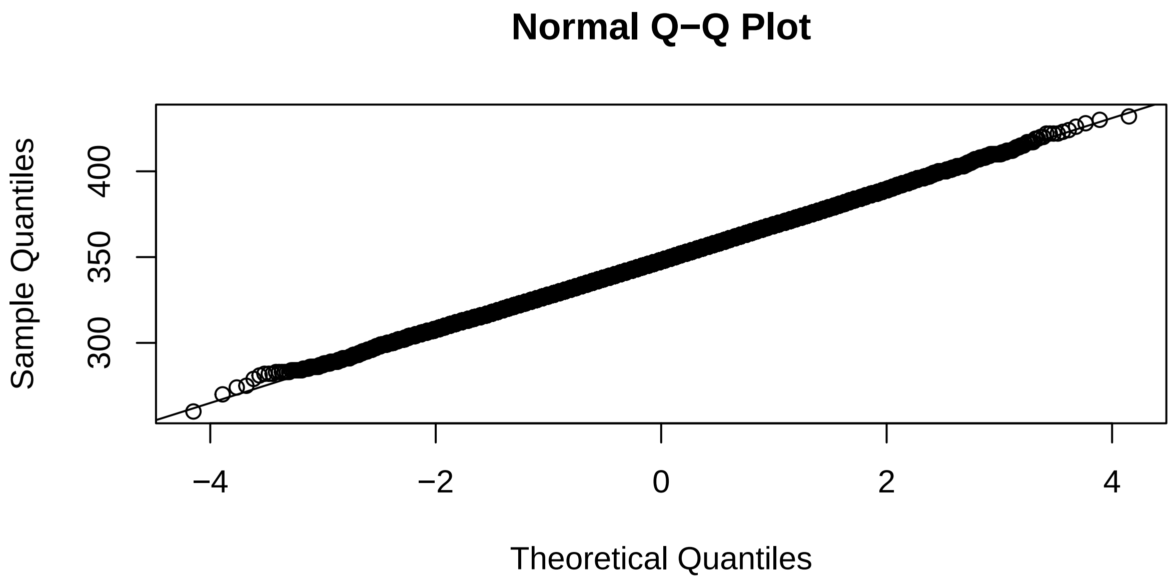
```
## [1] 336.000000   5.666667
```

```r
# 1c
vec.sum = replicate(n = 30000, expr = sum(sample(urn, 60, replace = T)))
```

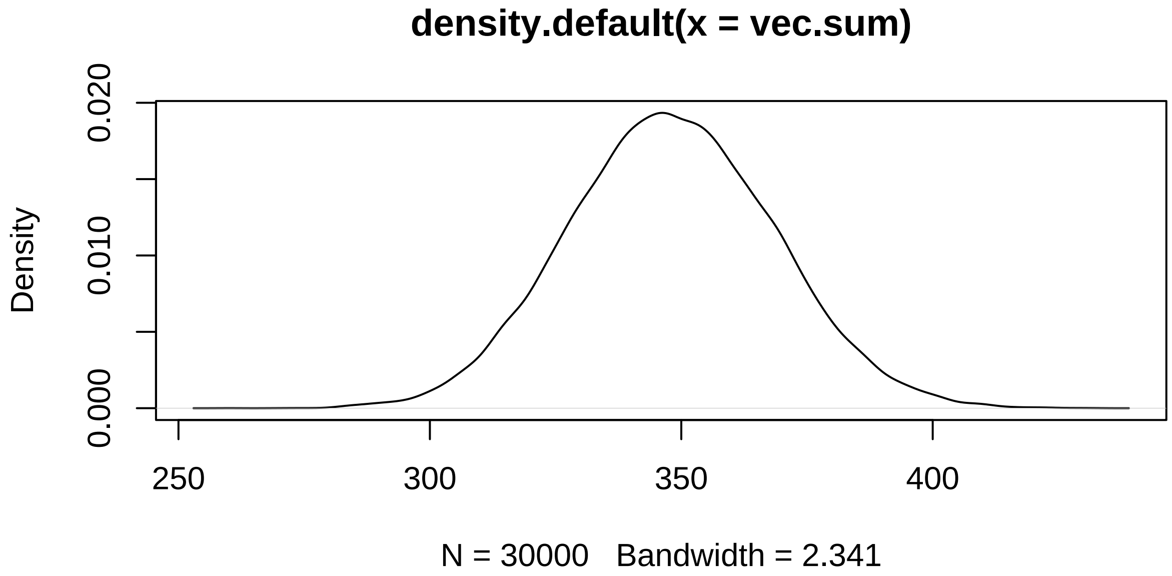
```r
# 1d
hist(vec.sum)
```



```
qqnorm(vec.sum)
qqline(vec.sum)
```



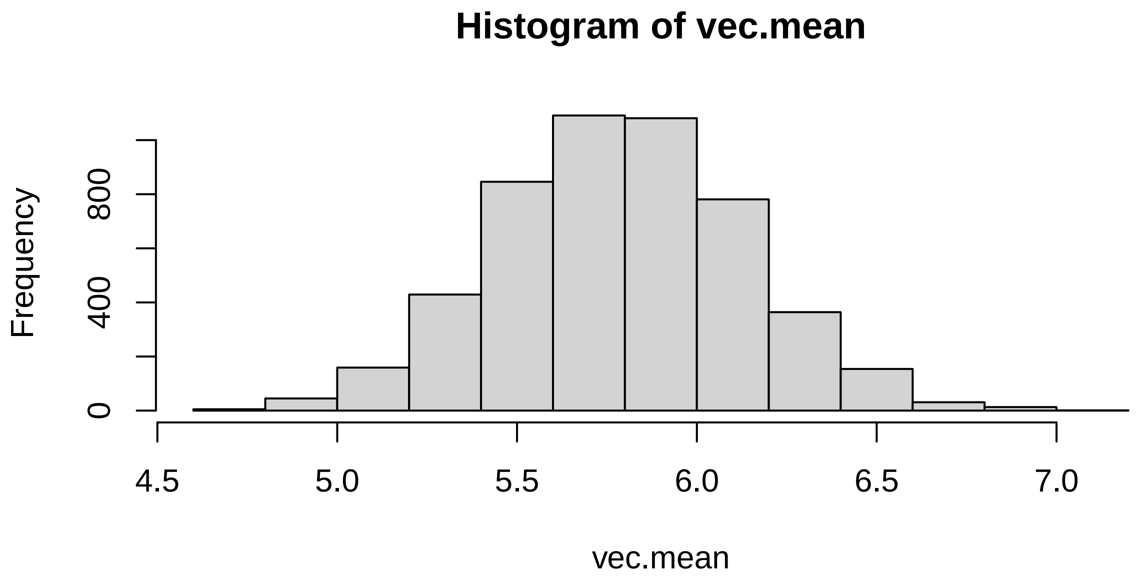
```
plot(density(vec.sum))
```



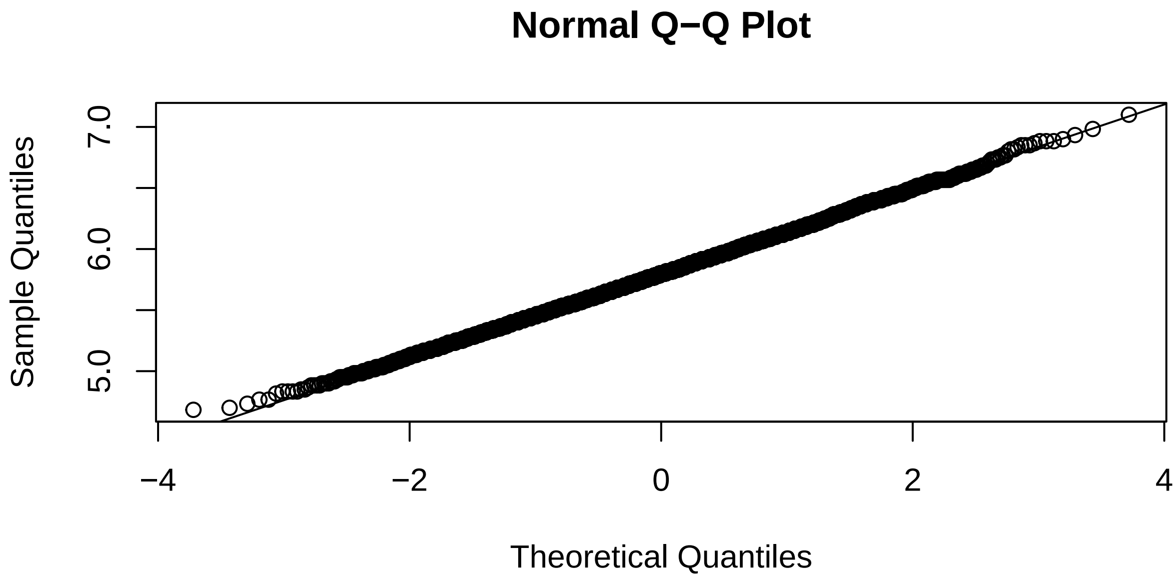
Based on the plots, the sample of 30000 sums does seem to come from a normal distribution. This is not surprising, as the sample size is large enough for the CLT to have an effect in the sum.

```
#1e
vec.mean=replicate(5000,mean(sample(urn,60,replace = T)))
```

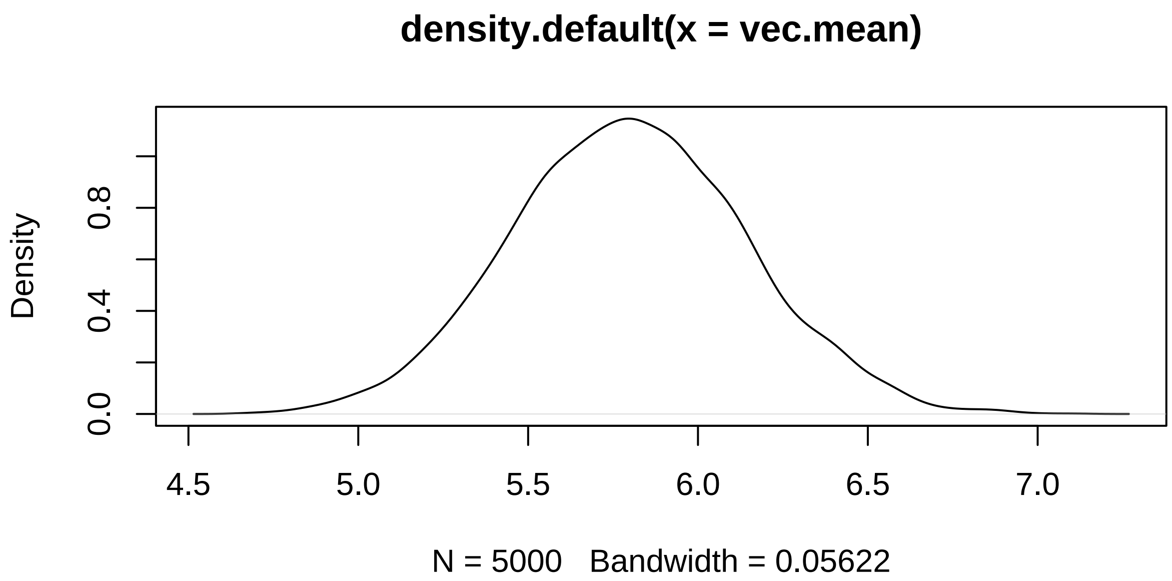
```
#1f
hist(vec.mean)
```



```
qqnorm(vec.mean)
qqline(vec.mean)
```



```
plot(density(vec.mean))
```



Based on the plots, the sample of 5000 sums does seem to come from a normal distribution. This is not surprising, as the sample size is large enough for the CLT to have an effect in the sample mean.

2.

a.

$$EX = \sum_{x \in X(S)} xf(x) = 1 \cdot 0.6 + 2 \cdot 0.1 + 3 \cdot 0.2 + 6 \cdot 0.1 = 2$$

```
x=c(1,2,3,6)
fx=c(0.6,0.1,0.2,0.1)
EX=sum(x*fx)
EX
```

```
## [1] 2
```

b.

$$VarX = \sum_{x \in X(S)} (x - EX)^2 f(x) = (1-2)^2 \cdot 0.6 + (2-2)^2 \cdot 0.1 + (3-2)^2 \cdot 0.2 + (6-2)^2 \cdot 0.1 = 2.4$$

```
var=sum((x-EX)^2*fx)
var
```

```
## [1] 2.4
```

c. $P(1.8 < X \leq 2.1) = P(X = 2) = 0.1$

d. Based on results shown in class, it is enough to show that $E\bar{X}_{80} = EX_1 = \mu = 2$ and $Var\bar{X}_{80} = \sigma^2/80 = 2.4/80 = 0.03$, but we can also redo the work in the context of this problem:

$$E\bar{X}_{80} = E\left(\sum_{i=1}^{80} \frac{1}{80} X_i\right) = \frac{1}{80}\left(\sum_{i=1}^{80} EX_i\right) = \frac{1}{80}\sum_i 2 = \frac{1}{80} 80 \cdot 2 = 2$$

and

$$Var\bar{X}_{80} = Var\left(\sum_{i=1}^{80} \frac{1}{80} X_i\right) = \left(\frac{1}{80}\right)^2 \left(\sum_{i=1}^{80} VarX_i\right) = \frac{1}{80^2}\sum_i 2.4 = \frac{1}{80^2} 80 \cdot 2.4 = \frac{2.4}{80} = 0.03$$

```
2.4/80
```

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

e. Since the probability distribution of the urn doesn't seem to be too extreme in any way, a sample of size 80 is large enough for the CLT to make $\bar{X}_{80} \dot\sim Normal\,(2, 0.03)$ that is, $\bar{X}$ follows approximately a normal distribution with mean 2 and variance 0.03. So $P(1.8 < \bar{X} \leq 2.1) = P(\bar{X} \leq 2.1) - P(\bar{X} \leq 1.8) \approx 0.59$

```
pnorm(2.1, 2, sqrt(0.03))-pnorm(1.8, 2, sqrt(0.03))
```

```
## [1] 0.594042
```

f.

```
x = c(1,2,3,6)
fx = c(0.6, 0.1, 0.2, 0.1)
xbar.vec = replicate(40000, mean(sample(x, 80, replace = T, prob = fx)))
mean(xbar.vec <=2.1 & xbar.vec >1.8)
```

```
## [1] 0.60595
```

```
sqrt(0.03)
```

```
## [1] 0.1732051
```

The probabilities in parts e and f are both close to 0.60.

3.

a.

```
urn = c(1,1,1,2,2,5,10,10,10,10)
urn.model = function(x = urn,n=40){sum(sample(x,n,replace=T))}
vec.y = replicate(10^5,urn.model(x=urn, n=40))
mean(vec.y > 170.5 & vec.y < 199.5)
```

```
## [1] 0.3017
```

b. We need to obtain the expected value and variance for the sum of 40 tickets. This is simply the expected value and the variance for selecting one ticket from the urn, multiply by 40, as shown in the following R code.

```
mu.urn = mean(urn)
var.urn = mean(urn^2) - mean(urn)^2
mu.y = 40*mu.urn
var.y = 40*var.urn
c(mu.y,var.y)
```

```
## [1] 208.0 662.4
```

Mean and standard deviation are not correct

```
se = sqrt(var.y)
pnorm(199.5, 208,se) - pnorm(170.5,208,se)
```

```
## [1] 0.2980481
```

c. If the number of simulations is large enough, eventually, because of the WLLN, method in part a will be more accurate. However, given the structure of the urn (no extreme numbers and no extreme associated probabilities) the normal approximation is quite good here and likely comparable to a simulation with a large number of replications.

4.

a. We have $EX_i = 351$ and $VarX_i = 1$ as the weight mean and variance of the ith Coke can, respectively. We don't know the distribution of weights, but assuming a random sample of 40 Coke cans is large enough for the CLT, we get

$$\bar{X}_{40} \dot{\sim} \mathcal{N}\left(351, \frac{1}{40}\right)$$

b. As in part a, we get:

$$\bar{Y}_{42} \dot{\sim} \mathcal{N}\left(350, \frac{1}{42}\right).$$

c. We cannot find $P(X_1 > 351.5)$ because we don't know the probability distribution of X_1 (in particular, we don't know whether it was drawn from an normal distribution).

d. Assuming a random sample of 40 Coke cans is large enough, the sample mean is approximately normally distributed and $P(\bar{X}_{40} > 351.5)$ can be calculated as follows:

```
1 - pnorm(q = 351.5, mean = 351, sd = sqrt(1/40))
```

```
## [1] 0.0007827011
```

e. We are asked to find $P(\bar{X}_{40} > \bar{Y}_{42})$ or alternatively $P(\bar{X}_{40} - \bar{Y}_{42} > 0)$. Observe this is simply the difference of two normally distributed random variables. So,

$$\bar{X}_{40} - \bar{Y}_{42} \dot{\sim} \mathcal{N}\left(351 - 350, \frac{1}{40} + \frac{1}{42}\right)$$

and if F is the CDF of $\bar{X}_{40} - \bar{Y}_{42}$ then

$$P(\bar{X}_{40} - \bar{Y}_{42} > 0) = 1 - P(\bar{X}_{40} - \bar{Y}_{42} \leq 0) = 1 - F(0)$$

In R we get

```
1 - pnorm(q = 0, mean = 351 - 350, sqrt(1/40 + 1/42))
```

```
## [1] 0.999997
```

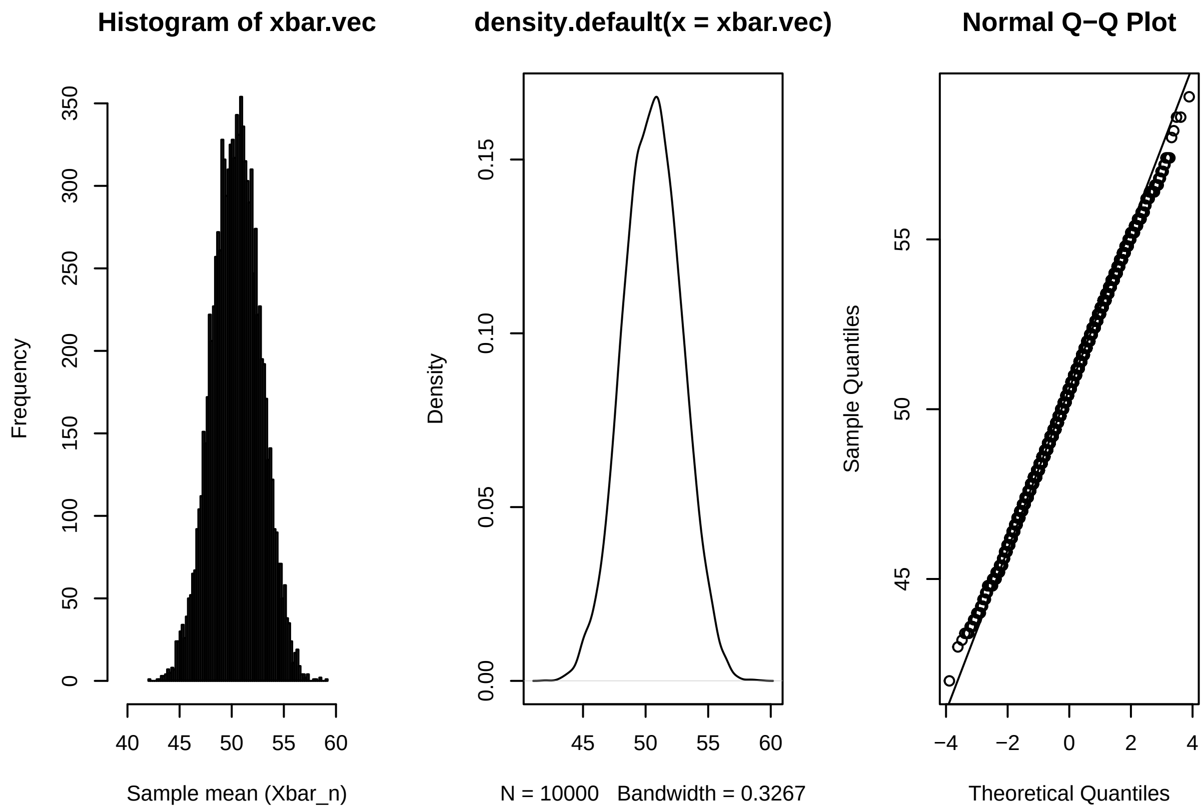
It is almost certain that the average weight of Coke cans is greater than the average weight of Pepsi cans, proving once and for all that Coke is better than Pepsi. :)

5.

```
clt = function(x, n, N = 10^4){
  xbar.vec = replicate(N, mean(sample(x, n, replace = T)))
  op = par(mfrow = c(1,3))
  hist(xbar.vec, breaks = 100,
       xlim = c(min(x), max(x)),
       xlab = paste("Sample mean (Xbar_n)"))
  plot(density(xbar.vec))
  qqnorm(xbar.vec);qqline(xbar.vec)
  par(op)
}
```

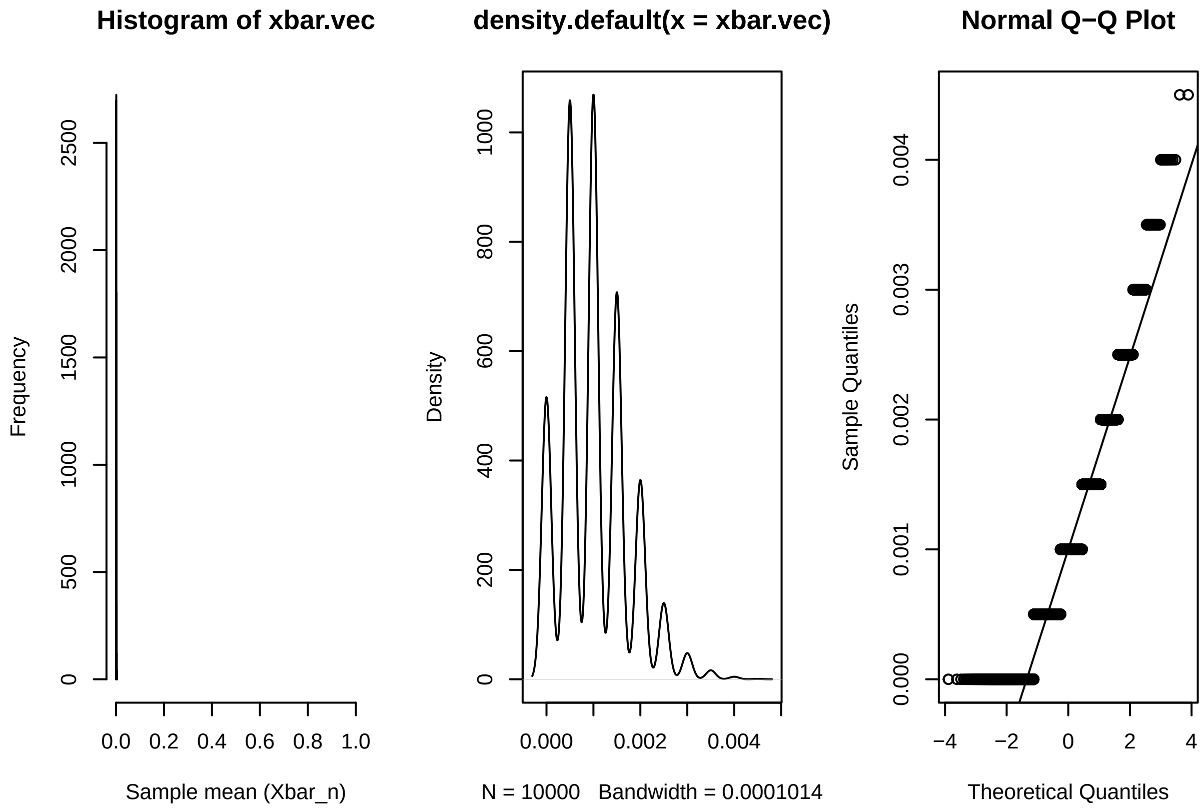
a. Many distributions could be used here. In particular, observe that if the probability distribution of the population is normal, the sample mean of a random sample of any size from that distribution will also be normal. For this example, we use a binomial with $p = 0.5$

```
x = rbinom(100,100,0.5)
clt(x,5)
```

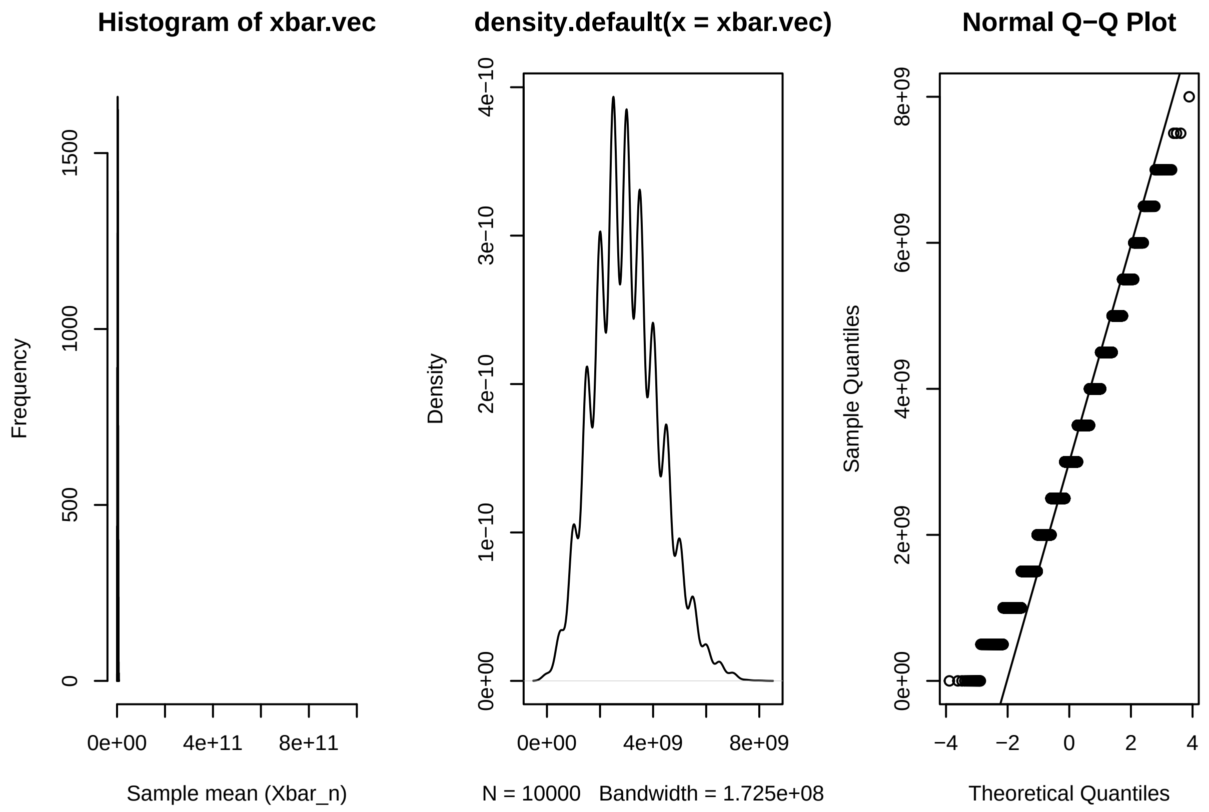


b. What we need is a distribution with some extreme values. Simple distributions such as a Bernoulli trial with a very small probability, say $p = 0.001$ would do the job. Below we also present another that comes from the union of two normals (one with a much higher mean than the other):

```
x1 = c(rep(0, 999),1)
clt(x1,n = 2000)
```



```
x2 = c(rnorm(9970),rnorm(30, mean = 10^12, sd = 0.1))
clt(x2,n = 2000)
```



Additional questions:

6.

```
n=20
mu = 5
sd = 30/60
var = sd^2
mu.y = n*mu
se = sqrt(n*var)
# to find P(Xbar > 105)
1 - pnorm(105,mu.y,se)
```

```
## [1] 0.01267366
```

7.

a.

```r
urn = c(1,1,1,1,2,5,5,10,10,10)
urn.model = replicate(100000,sum(sample(urn,40,replace=T)))
mean(urn.model > 170.5 & urn.model < 199.5)
```

```
## [1] 0.44874
```

b.

```r
mu = mean(urn) * 40
var = mean(urn^2 * 40) - mean(urn)^2 * 40
c(mu,var)
```

```
## [1] 184.0 585.6
```

The mean and standard deviation used is correct

```r
se = sqrt(var)
pnorm(199.5, 184,se) - pnorm(170.5,184,se)
```

```
## [1] 0.4506155
```

The calculation is correct and normal approximation is used which is correct.

c.

If the number of simulations is large enough, eventually, because of the WLLN, method in part a will be more accurate. However, given the structure of the urn (no extreme numbers and no extreme associated probabilities) the normal approximation is quite good here and likely comparable to a simulation with a large number of replications.

8.

Since a sample of size 1 million is taken, the standard deviation of the sample mean will be $\sigma/\sqrt{1000000} = 0.001 \cdot \sigma$. Hence, the vector proposed should have a large variance (and standard deviation) to account for this. Observe that if $\sigma = 50$ then $\sigma/\sqrt{1000000} = 0.05$ which is the size of ϵ used here. If you recall that about 68% of the observations are within one standard deviation from the mean, we then need at least $\sigma = 50$ to get the desired result, perhaps a little larger to make sure we don't get too many observations near the mean by chance. While many creative vectors can be proposed, here we create a random vector from a normal distribution with $\sigma = 100$:

```r
wlln = function (x, repl, n, epsilon){
          xbar.vec = replicate(repl, mean(sample(x, n, replace = T)))
          lb = mean(x) - epsilon # lower bound
          ub = mean(x) + epsilon
          prob = mean(xbar.vec >= lb & xbar.vec <= ub)
          data.frame(n = n, probability = round(prob,2))
        }

X = rnorm(10^4, 0, 100)
res1 = wlln(x = X, repl = 100, n = 1000000, epsilon = 0.05)
res1
```

```
##       n probability
## 1 1e+06        0.44
```