A. The Central Limit Theorem

Question1a

```
set.seed(seed = 55)
binom.mean <- function(n, rp){
  v.mean <- rep( NA, 500)
  for(i in 1:rp){
    binom.v <- rbinom(n, size = 1, p = 0.15)
    v.mean[i] <- mean(binom.v)
  }
   v.mean
}
binom.10 <- binom.mean(10, 500)
head(binom.10)
```

```
## [1] 0.0 0.2 0.1 0.3 0.4 0.4
```

```
mean(binom.10)
```

```
## [1] 0.1546
```

Question1b

```
binom.20 <- binom.mean(20, 500)
binom.30 <- binom.mean(30, 500)
binom.40 <- binom.mean(40, 500)
binom.50 <- binom.mean(50, 500)
```

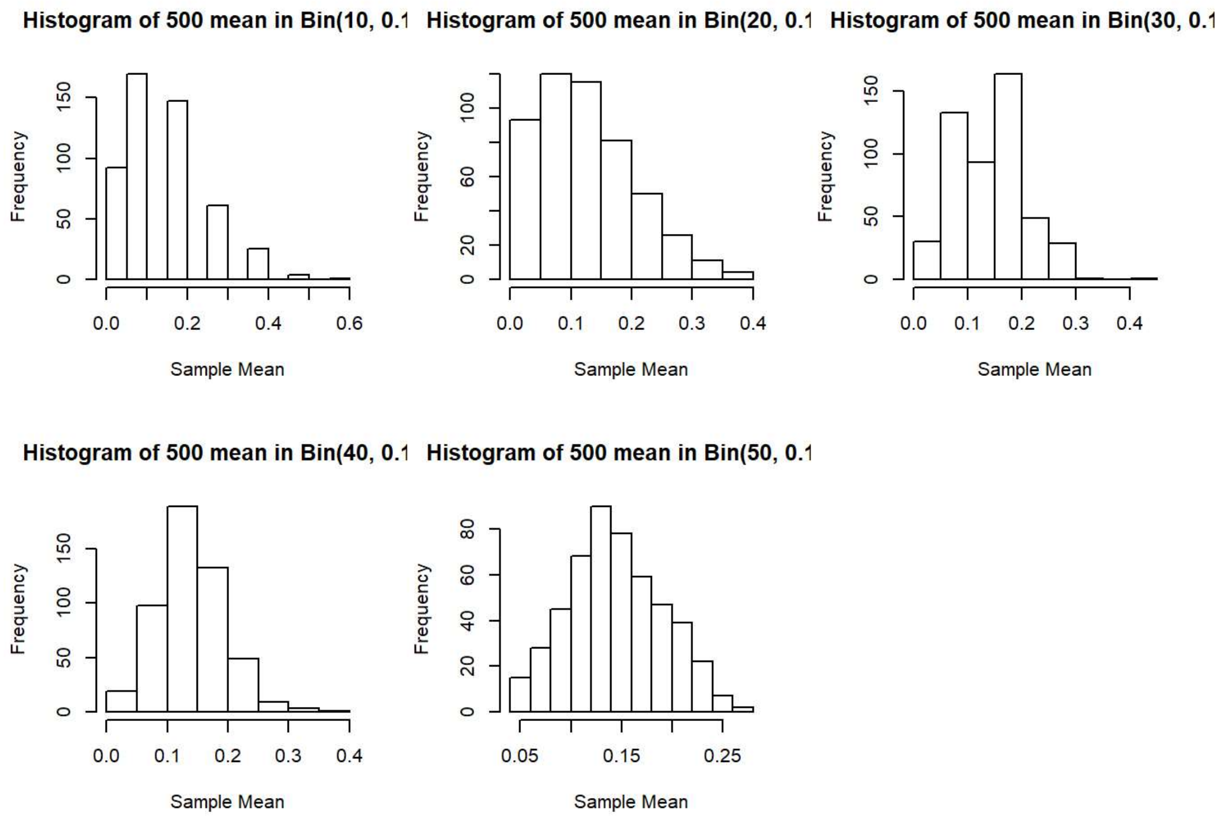
Question1c

```
binom.sample= list(binom.10, binom.20, binom.30, binom.40, binom.50)
binom.mu <- sapply(binom.sample, mean)
binom.sigma <- sapply(binom.sample, sd)
rbind(binom.mu, binom.sigma)
```

```
##                  [,1]       [,2]       [,3]     [,4]       [,5]
## binom.mu    0.1546000 0.15010000 0.15000000 0.151400 0.15412000
## binom.sigma 0.1139509 0.08263563 0.06658311 0.054649 0.04727256
```

Question1d

```
par(mfrow=c(2,3))
hist(binom.10, xlab = "Sample Mean", ylab = "Frequency", main = "Histogram of 500 mean in Bin(10, 0.15)")
hist(binom.20, xlab = "Sample Mean", ylab = "Frequency", main = "Histogram of 500 mean in Bin(20, 0.15)")
hist(binom.30, xlab = "Sample Mean", ylab = "Frequency", main = "Histogram of 500 mean in Bin(30, 0.15)")
hist(binom.40, xlab = "Sample Mean", ylab = "Frequency", main = "Histogram of 500 mean in Bin(40, 0.15)")
hist(binom.50, xlab = "Sample Mean", ylab = "Frequency", main = "Histogram of 500 mean in Bin(50, 0.15)")
```



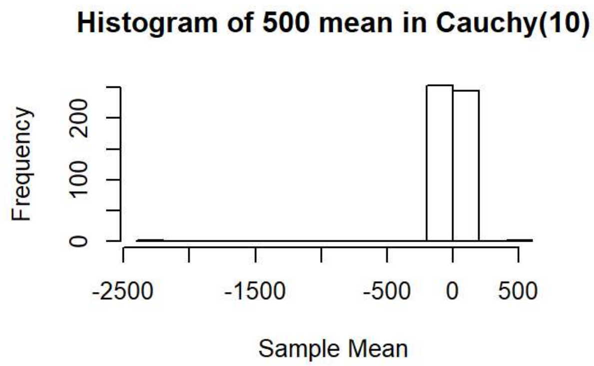
Question1e

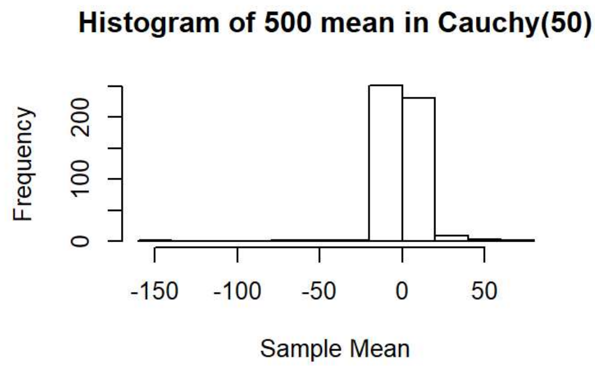
As we increase the sample size, the distribution of the sample mean gets more and more normal. The sample size n=50 begins to look like normal.

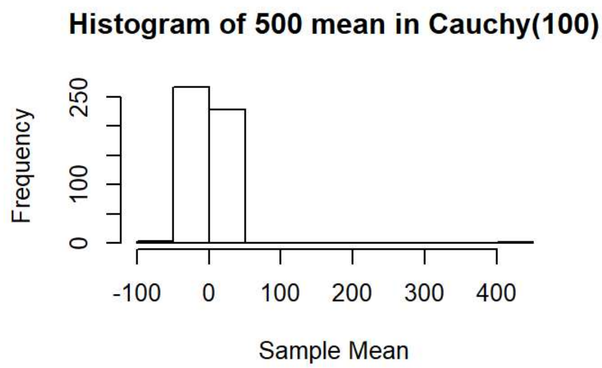
B. The CLT and the Cauchy Distribution

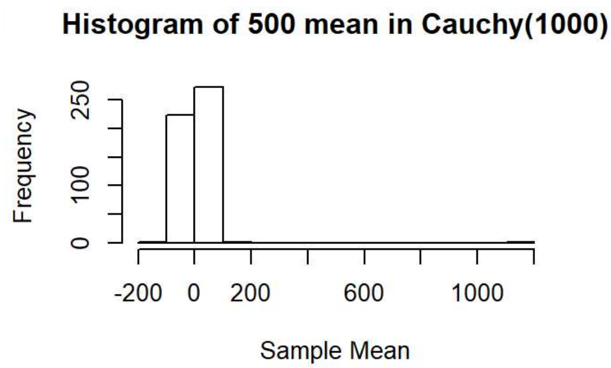
```
set.seed(seed = 55)
cauchy.mean <- function(n,rp){
  v.mean <- rep( NA, 500)
  for(i in 1:rp){
    cauchy.v <- rcauchy(n)
    v.mean[i] <- mean(cauchy.v)
  }
   v.mean
}

cauchy.10 <- cauchy.mean(10, 500)
cauchy.50 <- cauchy.mean(50, 500)
cauchy.100 <- cauchy.mean(100, 500)
cauchy.1000 <- cauchy.mean(1000, 500)
par(mfrow=c(2, 2))
hist(cauchy.10, xlab = "Sample Mean", ylab = "Frequency", main = "Histogram of 500 mean in Cauchy(10)")
hist(cauchy.50, xlab = "Sample Mean", ylab = "Frequency", main = "Histogram of 500 mean in Cauchy(50)")
hist(cauchy.100, xlab = "Sample Mean", ylab = "Frequency", main = "Histogram of 500 mean in Cauchy(100)")
hist(cauchy.1000, xlab = "Sample Mean", ylab = "Frequency", main = "Histogram of 500 mean in Cauchy(1000)")
```









No matter how large of the sample size, the sample means of Cauchy distribution are not normal at all. The CLT does not apply to Cauchy.

C. Estimating Hospital Budget

Part1

```
ProcedureCost <- read.csv("C:/Users/Goodgolden5/Desktop/BIOS6611-Alexander Kaizer/ProcedureCost.csv")
Group1 <- ProcedureCost[ProcedureCost$Procedure == 1, ]
Group1.Zero <- Group1[Group1$Cost == 0, ]
Group1.NZero <- Group1[Group1$Cost != 0, ]
Group2 <- ProcedureCost[ProcedureCost$Procedure == 2, ]
Group2.Zero <- Group2[Group2$Cost == 0, ]
Group2.NZero <- Group2[Group2$Cost != 0, ]
Group1.r <- cbind(length(Group1.Zero$Cost), length(Group1.NZero$Cost))
Group2.r <- cbind(length(Group2.Zero$Cost), length(Group2.NZero$Cost))
Cost.matrix <- rbind(Group1.r, Group2.r)
Cost.matrix
```

```
##      [,1] [,2]
## [1,]   48   72
## [2,]   15   65
```

Part2

```
p1 <- Cost.matrix[1, 2]/sum(Cost.matrix[1, ]); p1
```

```
## [1] 0.6
```

```
p2 <- Cost.matrix[2, 2]/sum(Cost.matrix[2, ]); p2
```

```
## [1] 0.8125
```

```
m1 <- mean(Group1.NZero$Cost); m1
```

```
## [1] 2.155417
```

```
m2 <- mean(Group2.NZero$Cost); m2
```

```
## [1] 1.085077
```

```
v1 <- var(Group1.NZero$Cost); v1
```

```
## [1] 1.262825
```

```
v2 <- var(Group2.NZero$Cost); v2
```

```
## [1] 1.58376
```

Part3

If the random variable R and Z are independent:

$$E[Y_i] = E[R_i Z_i] = Pr(R = i) * E[Z_i] = m_i p_i$$

$$Var[Y_i] = Var[R_i Z_i] = E[(R_i Z_i)^2] - E^2[R_i Z_i]$$

$$= E[R_i^2 Z_i^2] - E^2[R_i] E^2[Z_i] = E[R_i^2] E[Z_i^2] - E^2[R_i] E^2[Z_i]$$

$$= (p_i^2 + p_i(1 - p_i))(v_i + m_i^2) - p_i^2 m_i^2 = p_i v_i + p_i m_i^2 - p_i^2 m_i^2$$

$$Var[Y_i] = p_i v_i + p_i m_i^2 - p_i^2 m_i^2$$

Part4

It is definitely the qnorm will be applied. We have to recalculate the total sample mean and variance by sum up the each subset.

$$E[Y] = E[RZ] = (n_1 * E[Y_1] + n_2 * E[Y_2])/(n_1 + n_2)$$

$$Var[Y] = (n_1 * Var[Y_1] + n_2 * Var[Y_2])/(n_1 + n_2)$$

$$\sigma = \sqrt{Var[Y]}$$

```
n1 <- 120; n2 <- 200
e1 <- m1 * p1; e1
```

```
## [1] 1.29325
```

```
e2 <- m2 * p2; e2
```

```
## [1] 0.881625
```

```
e <- (n1 * e1 + n2 * e2); e
```

```
## [1] 331.515
```

```
var1 <- p1*v1 + p1*m1^2 - p1^2 * m1^2; var1
```

```
## [1] 1.872692
```

```
var2 <- p2*v2 + p2*m2^2 - p2^2 * m2^2; var2
```

```
## [1] 1.466173
```

```
var <- (n1 * var1 + n2 * var2); var
```

```
## [1] 517.9577
```

```
sigma <- sqrt(var); sigma
```

```
## [1] 22.75868
```

```
set.seed( seed = 555)
qnorm( 0.8, mean = e, sd = sigma)
```

```
## [1] 350.6692
```

Part5

So, the Gamma distributed simulation is very near to the expected budget we got from Part2, and Part4.

```
set.seed( seed = 555 )
Z1 <- 1:10000
Z2 <- 1:10000
Z <- 1:10000
for( i in 1:10000){
  Z1[i] <- sum(rgamma( n1, shape = e1^2/var1, scale = var1/e1))
  Z2[i] <- sum(rgamma( n2, shape = e2^2/var2, scale = var2/e2))
  Z[i] = Z1[i] + Z2[i]
}

par(mfrow=c(2,2))
hist(Z1)
hist(Z2)
hist(Z)
quantile(Z, )
```

```
##       0%      25%      50%      75%     100%
## 247.8470 315.8833 331.0828 346.1652 428.6142
```

```
quantile(Z, 0.8)
```

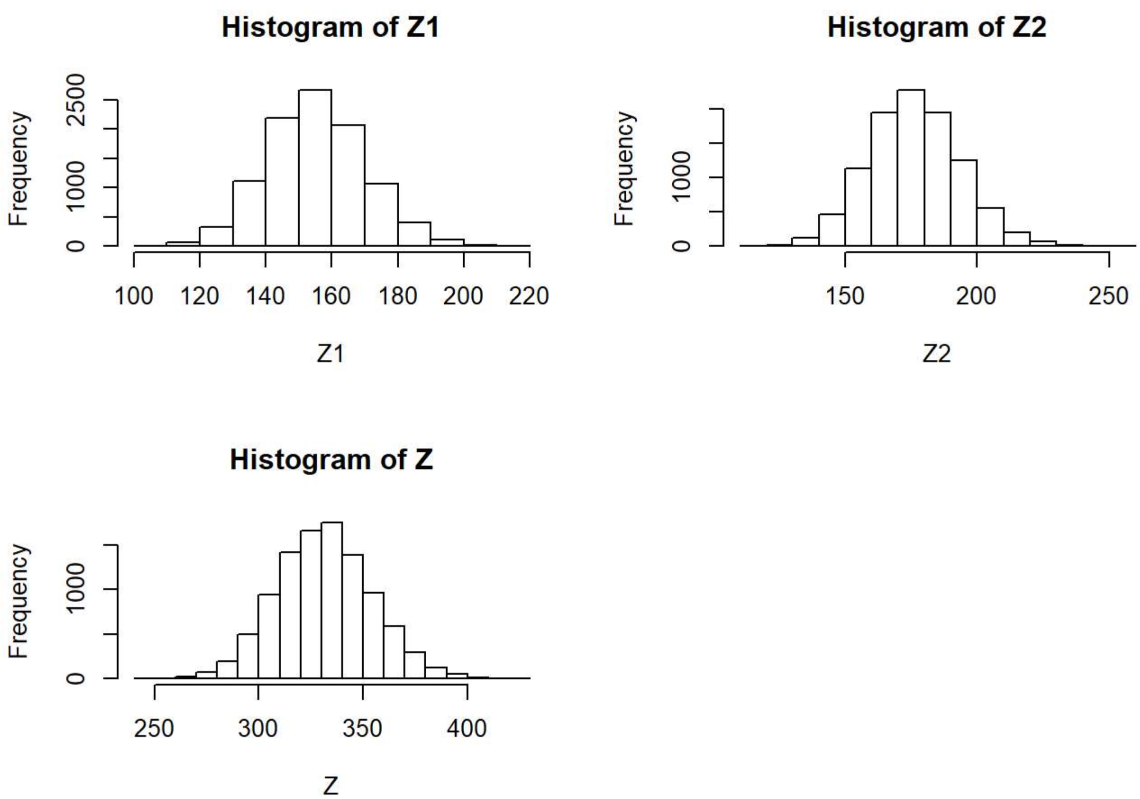
```
##      80%
## 350.2712
```

```
## This one is just for recheck
qgamma(0.8, shape = e^2/var, scale = var/e)
```

```
## [1] 350.5016
```

```
qnorm( 0.8, mean = e, sd = sigma)
```

```
## [1] 350.6692
```



Part6

a. I would assum the simulation sample is large enough (in this case, n1=120, n2=200, or n=320).

I really have no idea what the question is asking…, because I do not know what will happen if sample size is very small. especially the n1 = 120 has a more accurate approximation than n2 = 200.

```
e1*n1; mean(Z1); (e1*n1 - mean(Z1))/(e1*n1)
```

```
## [1] 155.19
```

```
## [1] 155.3252
```

```
## [1] -0.0008709957
```

```
e2*n2; mean(Z2); (e2*n2 - mean(Z2)) / (e2*n2)
```

```
## [1] 176.325
```

```
## [1] 176.1399
```

```
## [1] 0.001049792
```

```
e; mean(Z); (e - mean(Z)) / e
```

```
## [1] 331.515
```

```
## [1] 331.4651
```

```
## [1] 0.0001506259
```

b. I would assume that the level of accuracy is around 0.001 for the approximation via simulation.